ETSI TS 103 195-2 V1.1.1 (2018-05)

Autonomic network engineering for the self-managing Future Internet (AFI); Generic Autonomic Network Architecture; Part 2: An Architectural Reference Model for Autonomic Networking, Cognitive Networking and Self-Management

Reference

DTS/NTECH-AFI-0015-2-GS02

Keywords

autonomic networking, self-management, self-organization, artificial intelligence

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Network Technologies (NTECH).

The present document is part 2 of a multi-part deliverable. Full details of the entire series can be found in part 1 [i.70].

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

As discussed in ETSI White Paper No.16 [3], the industry consensus is that in the digital services ecosystem, networks evolve to so-called "smart networks of the future", which are characterized by the need to be operated based on principles of dynamically adaptive Automated and Autonomic Management and Control (AMC) of networks and services (a.k.a autonomics). AMC replaces the increasingly complex and error-prone manual and static management and optimization of networks and services. Networks become smart, intelligent and self-managing or self-driving in some of their operations and behaviours, thanks to the AMC (autonomics) paradigm. The present document presents the Generic Autonomic Networking Architecture (GANA) Reference Model for Autonomic Networking, Cognitive Networking and Self-Management for Networks and Services - a model for implementing the AMC paradigm. GANA defines so-called Autonomic Functions (AFs) as autonomous and autonomic decision-making elements (DEs) for network management and control that can be instrumented at four basic complementary abstraction levels for self-management within network nodes or elements/functions and in the outer management and control realm. The main goal of the GANA reference model is prescribing design and operational principles for Decision Elements (DEs) as the drivers for cognitive, self-managing and self-adaptive network behaviours that enable to achieve OPEX reduction and other benefits "Artificial Intelligence/Cognition in AMC (autonomics)" bring to Network Operators and End User Customers, and to Enterprise Networks as well, such as:

- Dynamic and Analytics-Driven Service Fulfilment and Closed-Loop (Autonomic) Service Assurance.
- Predictive, Proactive and Advanced Customer Experience.

Autonomics algorithms in the scope of GANA are meant to be implemented by the so-called GANA Decision-making-Elements (DEs) that can be designed to operate at four basic complementary hierarchical levels for self-management behaviours within network nodes and in the outer management and control realm. Autonomics algorithms include Cognitive algorithms for Artificial Intelligence (AI)-such as Machine Learning (ML), Deep Learning (DL), Computational Intelligence, and other algorithms that can be employed in DEs' closed-loop operations.

NOTE: The ETSI White Paper No.16 [3] describes the two categories that determine the actors or players the GANA model is addressing, namely: Suppliers (vendors) of GANA Functional Blocks (FBs); and Provider of assets required by the developers of GANA Functional Blocks (FBs).

1 Scope

The scope of the present document is to provide the definition of the Generic Autonomic Network Architecture (GANA) as an architectural reference model for autonomic networking, cognitive networking and self-management that addresses the requirements defined in ETSI TS 103 194 [1] - a compilation of example requirements which reflect real-world problems that benefit from the application of automated management, autonomic management and self-management principles for networks and services delivered by the network to applications. The objective of the present document is to describe the GANA reference model with its associated Functional Blocks (FBs) and their associated reference points that can be instantiated onto target currently existing, emerging or future reference network architectures (including their management and control architectures) to create autonomics-enabled reference network architectures and their associated management and control architectures. The present document builds on the ETSI GS AFI 002 [2] specification by extracting key concepts of the GANA model and adding additional aspects that were not covered in ETSI GS AFI 002 [2] and also providing pointers on where to find details on the integration of the GANA model with reference models for other emerging complementary networking paradigms other than autonomics, namely:

- SON (Self-Organizing Networks).
- SDN (Software-Defined Networking).
- NFV (Network Functions Virtualisation).
- E2E Orchestration.
- Network Analytics.
- Big-Data Analytics for Autonomic Management and Control (AMC) of networks and services; and
- Closed-Loop Service Assurance.

This means is recommended that the present document is used together with ETSI GS AFI 002 [2], which contains valuable complementary details. The other goal is to describe how the human network operator could govern end to end autonomic networks and their management and control architectures.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 103 194: "Network Technologies (NTECH); Autonomic network engineering for the self-managing Future Internet (AFI); Scenarios, Use Cases and Requirements for Autonomic/Self-Managing Future Internet".

[2] ETSI GS AFI 002: "Autonomic network engineering for the self-managing Future Internet (AFI); Generic Autonomic Network Architecture (An Architectural Reference Model for Autonomic Networking, Cognitive Networking and Self-Management)".

[3] ETSI White Paper no. 16: "GANA - Generic Automatic Networking Architecture".

NOTE: Available at http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp16_gana_Ed1_20161011.pdf.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] R. Chaparadza, S. Papavassiliou, T. Kastrinogiannis, M. Vigoureux , E. Dotaro, A. Davy, K. Quinn, M. Wodczak, A. Toth, A. Liakopoulos, M. Wilson: "Creating a viable Evolution Path towards Self-Managing Future Internet via a Standardizable Reference Model for Autonomic Network Engineering". Published in the book by the Future Internet Assembly (FIA) in Europe: Towards the future internet - A European research perspective. Amsterdam: IOS Press, 2009, pp. 136-147.

[i.2] Ranganai Chaparadza, Tayeb Ben Meriem, John Strassner, Steven Wright, Joel Halpern: "Industry Harmonization for Unified Standards on Autonomic Management & Control of Networks and Services, SDN, NFV, E2E Orchestration, and Software-oriented enablers for 5G", Report from the Joint SDOs/Fora Workshop hosted by TMForum during TMForum Live 2015 Event Nice, France (June 4th, 2015).

NOTE: Available at https://www.tmforumlive.org/wp-content/uploads/2015/06/Report-on-Joint-SDOs-Industry-Harmonization-for-Unified-Standards-on-AMC_SDN_NFV_E2E-Orchestration_ver3.01.pdf.

[i.3] ETSI TS 103 371: "Network Technologies (NTECH); Autonomic network engineering for the self-managing Future Internet (AFI); Proofs of Concept Framework".

[i.4] ETSI TR 103 404: "Network Technologies (NTECH); Autonomic network engineering for the self-managing Future Internet (AFI); Autonomicity and Self-Management in the Backhaul and Core network parts of the 3GPP Architecture".

[i.5] ETSI EG 203 341: "Core Network and Interoperability Testing (INT); Approaches for Testing Adaptive Networks".

[i.6] ETSI TR 103 495: "Network Technologies (NTECH); Autonomic network engineering for the self-managing Future Internet (AFI); Autonomicity and Self-Management in Wireless Ad-hoc/Mesh Networks: Autonomicity-enabled Ad-hoc and Mesh Network Architectures".

[i.7] ETSI NTECH AFI Proof of Concept (PoC): "5G Network Slices Autonomic Management & E2E Orchestration, with Closed-Loop (Autonomic) Service Assurance for the Slices: IoT use case"; Orange proposal PoC description V01 Tayeb Ben Meriem June 27th 2016.

[i.8] Gilles Privat, Wenbin Li: "Unlocking IoT/WoT APIs with linked data & semantic referencing"; 29th March 2016, Orange APIs Standardisation Seminar.

[i.9] Appledore Research Group White Paper: "Closed Loop Automation and the new role of Assurance", 9/1/2016.

[i.10] John C Strassner, Nazim Agoulmine, and Elyes Lehtihet. FOCALE: "A Novel Autonomic Networking Architecture"; In LAACS, Campo Grande, MS, Brazil, 2006.

[i.11] Albert Greenberg, et al.: "A clean slate 4D approach to network control and management"; In ACM SIGCOMM Computer Communication Review Homepage archive Volume 35 Issue 5, October 2005, pages 41-54.

[i.12] IBM White paper: "An architectural blueprint for autonomic computing"; June 2005.

[i.13] Autonomic Networking Integrated Model and Approach (anima).

NOTE: Available at https://datatracker.ietf.org/wg/anima/charter/.

[i.14] R. Chaparadza, et al.: "SDN Enablers in the ETSI AFI GANA Reference Model for Autonomic Management & Control (emerging standard), and Virtualisation Impact"; In the proceedings of the 5th IEEE™ MENS Workshop at IEEE Globecom 2013, December, Atlanta, Georgia, USA.

[i.15] R. Chaparadza, Tayeb Ben Meriem, Benoit Radier, Szymon Szott, Michal Wodczak, Arun Prakash, Jianguo Ding, Said Soulhi, Andrej Mihailovic: "Implementation Guide for the ETSI AFI GANA Model: a Standardized Reference Model for Autonomic Networking, Cognitive Networking and Self-Management"; In the proceedings of the 5th IEEE™ MENS Workshop at IEEE Globecom 2013, December, Atlanta, Georgia, USA.

[i.16] GENI (Global Environment for Network Innovations) Initiative.

NOTE: Available at http://www.geni.net/.

[i.17] Hitesh Ballani and Paul Francis: "CONMan: A Step towards Network Manageability"; In SIGCOMM '07: Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, pages 205-216, New York, NY, USA, 2007. ACM.

[i.18] Ranganai Chaparadza; Michal Wodczak; Tayeb Ben Meriem; Paolo De Lutiis; Nikolay Tcholtchev; Laurent Ciavaglia: "Standardization of resilience & survivability, and autonomic fault-management, in evolving and future networks: An ongoing initiative recently launched in ETSI"; In proceedings of 2013 9th International Conference on the Design of Reliable Communication Networks (DRCN 2013); ISBN 9781479900497; 4-7 March 2013, Budapest, Hungary.

[i.19] John C Strassner, Nazim Agoulmine, and Elyes Lehtihet: "FOCALE: A Novel Autonomic Networking Architecture"; In LAACS, Campo Grande, MS, Brazil, 2006.

[i.20] David D. Clark, Craig Partridge, and J. Christopher Ramming: "A knowledge plane for the Internet"; In SIGCOMM, pages 3-10, 2003.

[i.21] Michal Wodczak, Tayeb Ben Meriem, Benoit Radier, Ranganai Chaparadza, Kevin Quinn, Jesse Kielthy, Brian A. Lee, Laurent Ciavaglia, Kostas Tsagkaris, Szymon Szott, Anastasios Zafeiropoulos, Athanassios Liakopoulos, Apostolos Kousaridas, Maurice Duault: "Standardizing a reference model and autonomic network architectures for the self-managing future internet"; IEEE Network 25(6): 50-56 (2011).

[i.22] Edited by Nazim Agoulmine: "Autonomic Network Management Principles: From Concepts to Applications"; Book published by Elsevier in 2011: ISBN 978-0-12-382190-4.

[i.23] Nikolay Tcholtchev, Ranganai Chaparadza, and Arun Prakash: "Addressing Stability of Control-Loops in the Context of the GANA Architecture: Synchronization of Actions and Policies"; In IWSOS "09: Proceedings of the 4th IFIP TC 6 International Workshop on Self-Organizing Systems, pages 262-268, Berlin, Heidelberg, 2009. Springer.

[i.24] DTR/NTECH-AFI-0020-GANA-impact: "Network Technologies (NTECH); Autonomic network engineering for the self-managing Future Internet (AFI); Study of impact of selected emerging technologies on the AFI GANA model and on its instantiations".

NOTE: Available at https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=42950.

[i.25] IETF Draft: "IP based Generic Control Protocol (IGCP)".

NOTE: Available at https://datatracker.ietf.org/doc/draft-chaparadza-intarea-igcp/.

[i.26] draft-ietf-anima-grasp-08: "A Generic Autonomic Signaling Protocol (GRASP)".

NOTE: Available at https://tools.ietf.org/html/draft-ietf-anima-grasp-08.

[i.27] ETSI TR 103 404: "Network Technologies (NTECH); Autonomic network engineering for the self-managing Future Internet (AFI); Autonomicity and Self-Management in the Backhaul and Core network parts of the 3GPP Architecture".

NOTE: Available at http://www.etsi.org/deliver/etsi_tr/103400_103499/103404/01.01.01_60/tr_103404v010101p.pdf.

[i.28] ETSI TR 103 473: "Network Technologies (NTECH); Autonomic network engineering for the self-managing Future Internet (AFI); Autonomicity and Self-Management in the Broadband Forum (BBF) Architectures".

[i.29] Szymon Szott, et al.: "Enabling autonomicity in wireless mesh networks with the ETSI AFI GANA reference model", Wiley InterScience: DOI: 10.1002/nem: international journal of network management.

[i.30] Tobias Bandh: PhD Dissertation, April 2013: "Coordination of autonomic function execution in Self-Organizing Networks", Network Architectures and Services Department of Computer Science, Technische Universität München (TUM), Germany.

[i.31] Soguy Mak karé Gueye, Noël De Palma, Eric Rutten, Alain Tchana and Daniel Hagimont: "Discrete control for ensuring consistency between multiple autonomic managers", Journal of Cloud Computing: Advances, Systems and Applications 2013, 2:16.

[i.32] Gabor Retvari, Felician Nemeth, Ranganai Chaparadza and Robert Szabo: "OSPF for Implementing Self-adaptive Routing in Autonomic Networks: a Case Study"; In proceedings of the 4th IEEE International Workshop on Modelling Autonomic Communication Environments (MACE 2009), October 26-27 2009, Venice, Italy.

[i.33] A. Atlas, J. Halpern, et al.: "An Architecture for the Interface to the Routing System: draft-ietf-i2rs-architecture-13", February 20, 2016.

[i.34] Krzysztof Cabaj, Krzysztof Szczypiorski and Sheila Becker: "Towards Self-defending Mechanisms Using Data Mining in the EFIPSANS Framework"; N.-T. Nguyen et al. (Eds.): Adv. in Multimed. and Netw. Inf. Syst. Technol., AISC 80, pp. 143-151.

[i.35] Albert Mestres et al.: "Knowledge-Defined Networking"; In ACM SIGCOMM Computer Communication Review Volume 47 Issue 3, July 2017.

[i.36] Alexej Starschenko, Nikolay Tcholtchev, Arun Prakash, Ina Schieferdecker, Ranganai Chaparadza: "Auto-Configuration of OSPFv3 Routing in FixedIPv6 Networks"; In 2015 7th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT): DOI: 10.1109/ICUMT.2015.7382427.

[i.37] gRPC protocol.

NOTE: Available at https://grpc.io/about/.

[i.38] Chaparadza R., Petre R., Prakash A., Németh F., Kukliński S., Starschenko A. (2011): "IPv6 and Extended IPv6 (IPv6++) Features That Enable Autonomic Network Setup and Operation"; In: Szabó R., Zhu H., Imre S., Chaparadza R. (eds) Access Networks. AccessNets 2010. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 63. Springer, Berlin, Heidelberg.

[i.39] White Paper by ASSIA: "Software Defined Access Networking (SDAN): Taking NFV and SDN to the Edge".

[i.40] Heavy Reading & SaS White Paper: "Advanced Predictive Analytics: Optimize your Network & Transform Customer Experience".

[i.41] Ren Quinn, et al.: KnowNet: "Towards a knowledge plane for enterprise network management"; In proceedings of NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium: 10.1109/NOMS.2016.7502819.

[i.42] Bruno Cornaglia, Vodafone Group Services: "Fixed Access Network Virtualisation", Broadband Forum Contribution bbf2014.084.00, February 14th, 2014.

[i.43] Jay S. Bayne: "Creating Rational Organizations: Theory of Enterprise Command and Control"; Publisher: Meta Command Systems, Inc. (2002): ISBN-10: 097885960X: ISBN-13: 978-0978859602.

[i.44] Wendong Wang, Ye Tian, Xiangyang Gong, Qinglei Qi, Yannan Hu: "Software defined autonomic QoS model for future Internet", Elsevier Journal of Systems and Software: Volume 110, December 2015, pages 122-135.

[i.45] Haiyan Zhang, Mincheng Zhao, Wendong Wang, Xiangyang Gong, Xirong Que: "A novel autonomic architecture for QoS management in wired network", GLOBECOM Workshops (GC Wkshps), 2010 IEEE: 10.1109/GLOCOMW.2010.5700376.

[i.46] A. Prakash, et al.: "Auto-Discovery and Auto-Configuration of Routers in Autonomic Networks": DOI: 10.1007/978-3-642-20931-4-24: Conference: Access Networks - 5th International ICST Conference on Access Networks, AccessNets 2010 and First ICST International Workshop on Autonomic Networking and Self-Management in Access Networks, SELFMAGICNETS 2010, Budapest, Hungary, November 3-5, 2010.

[i.47] V. Kaldanis, et al.: "Methodology Towards Integrating Scenarios and Testbeds for Demonstrating Autonomic/self-managing Networks and Behaviors required in Future Networks", SELFMAGICNETS'10: Proc of the International Workshop on Autonomic Networking and Self-Management. ICST ACCESSNETS'10, Budapest, Hungary, November 2010.

[i.48] G. Aristomenopoulos, T Kastrinogiannis, Z Li, M Wilson, M González Juan, A Lozano-López Jose, Y Li, V Kaldanis, S Papavassiliou: "Autonomic Mobility and Resource Management over an Integrated Wireless Environment - A GANA oriented approach", Proc of the 2nd International workshop on Management of Emerging Networks and Services, MENS 2010, Miami, Florida, December 2010.

[i.49] G. Aristomenopoulos, T. Kastrinogiannis, Z. Li and S. Papavassiliou: "An Autonomic QoS-centric Architecture for Integrated Heterogeneous Wireless Networks", Springer Mobile Networks and Applications (MONET), vol. 16, Issue 4, pp. 490-504, 2011.

[i.50] A. Liakopoulos, A. Zafeiropoulos, C. Marinos, M. Grammatikou, N. Tcholtchev and P. Gouvas: "Applying distributed monitoring techniques in autonomic networks", In Proceedings of the 2nd International workshop on Management of Emerging Networks and Services, MENS 2010, Miami, Florida, December 2010.

[i.51] Marinos C., Argyropoulos C., Grammatikou M., Maglaris V. (2011): "An Autonomic Monitoring Framework for QoS Management in Multi-service Networks"; In: Pentikousis K., Agüero R., García-Arranz M., Papavassiliou S. (eds) Mobile Networks and Management. MONAMI 2010. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 68. Springer, Berlin, Heidelberg.

[i.52] Cong-Vinh Phan: "Formal and Practical Aspects of Autonomic Computing and Networking: Specification, Development and Verification", IGI Global ISBN: 1609608461.

[i.53] Yuhong Li, Wei Deng, Baowen Liu, Yan Shi, Zhen Cao, Hui Deng: "Autonomicity in Distributed Mobility Management", GLOBECOM workshop MENS, Dec. 5-9, 2011, Houston, Texas, USA.

[i.54] R. Chaparadza: "On-Demand Monitoring: a Monitoring Paradigm for Traffic Flows in Multi-Service Self-Managing Networks", PhD Dissertation, 2011: Faculty of Electrical Engineering and Computer Science (German: Fakultät Elektrotechnik und Informatik der TU Berlin, FR5-1: Fakultät IV) of The Technical University of Berlin (TUB), Germany.

[i.55] Lei Shi: "On monitoring and fault management of next generation networks", PhD Dissertation, Georg-August-Universität Göttingen Fakultät für Mathematik und Informatik, Göttingen 2010.

[i.56] Anastasios Zafeiropoulos, et al.: "Monitoring within an Autonomic Network: AGANA based Network Monitoring Framework", Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops: 2nd Workshop on Monitoring, Adaptation and Beyond (MONA+).

[i.57] Resource Description Framework (RDF).

NOTE: Available at https://www.w3.org/RDF/.

[i.58] JSON for Linking Data.

NOTE: Available at https://json-ld.org/.

[i.59] Arun Prakash, Zoltan Theisz et al.: "Formal Methods for Modeling, Refining and Verifying Autonomic Components of Computer Networks", Springer Transactions on Computational Science (TCS) - Advances in Autonomic Computing: Formal Engineering Methods for Nature-Inspired Computing Systems, 2012.

[i.60] Jeffrey O. Kephart, Rajarshi Das: "Achieving Self-Management via Utility Functions"; In Journal IEEE Internet Computing archive, Volume 11 Issue 1, January 2007, Pages 40-48: DOI: 10.1109/MIC.2007.2.

[i.61] Thomas Magedanz, F.Schreiner and S.Wahle: "Service-oriented testbed infrastructures and cross-domain federation for Future Internet research"; In Proc. Integrated Network Management-Workshops, 2009. IM '09. IFIP/IEEE International Symposium pp.101-106, 1-5 June 2009.

[i.62] Raouf Boutaba, Jin Xiao, and Qi Zhang: "Autonomic Computing and Networking", chapter "Toward Autonomic Networks": Knowledge Management and Self-Stabilization, pages 239-260. 2009.

[i.63] A. Mihailovic, I. Chochliouros, A. Kousaridas, G. Nguengang, C. Polychronopoulos, J. Borgel, M. Israel, V. Conan, M. Belesioti, E. Sfakianakis, G. Agapiou, H. Aghvami and N. Alonistioti: "Architectural Principles for Synergy of Self-management and Future Internet Evolution", Proceedings of ICT Mobile Summit, June 2009.

[i.64] Verizon White Paper: Version 1.0, February 2016: "Verizon Network Infrastructure Planning: SDN-NFV Reference Architecture".

[i.65] OODA Loop.

NOTE: Available at http://www.artofmanliness.com/2014/09/15/ooda-loop/.

[i.66] Stephen Quirolgico, Kevin Mills and Doug Montgomery: "Deriving Knowledge for the Knowledge Plane". Draft from National Institute of Standards and Technology Advanced Network Technologies Division Gaithersburg, June 2003. MD 20899-8920.

[i.67] OMG (version 2.4.1): "Meta Object Facility (MOF)" Core Specification.

NOTE: Available at http://www.omg.org/spec/MOF/2.4.1.

[i.68] IETF RFC 3444: "On the Difference between Information Models and Data Models".

[i.69] Teemu Koponen, Martin Casado, Natasha Gude, Jeremy Stribling, Leon Poutievski, Min Zhu, Rajiv Ramanathan, Yuichiro Iwata, Hiroaki Inoue, Takayuki Hama and Scott Shenker: "Onix: A Distributed Control Platform for Large-scale Production Networks". 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI '10), October 4-6, 2010, Vancouver, BC, Canada.

[i.70] ETSI TR 103 195-1: "Autonomic network engineering for the self-managing Future Internet (AFI); Generic Autonomic Network Architecture; Part 1: Business drivers for autonomic networking".

NOTE: Available at https://portal.etsi.org/webapp/workProgram/Report_WorkItem.asp?wki_id=50969.

[i.71] ETSI TR 103 195-3: "Autonomic network engineering for the self-managing Future Internet (AFI); Generic Autonomic Network Architecture; Part 3: Guidelines to instantiating the GANA onto a target implementation-oriented reference architecture".

NOTE: Available at https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=50971.

[i.72] Zoltan Theisz, Gabor Batori, Domonkos Asztalos: "Formal Logic Based Configuration Modeling and Verification for Dynamic Component Systems"; In proceedings of MOPAS 2011: The Second International Conference on Models and Ontology-based Design of Protocols, Architectures and Services, April 17-22, 2011, Budapest, Hungary.

[i.73] Piotr Rygielski, Steffen Zschaler, Samuel Kounev: "A Meta-Model for Performance Modeling of Dynamic Virtualised Network Infrastructures"; In proceedings of the 4th ACM/SPEC International Conference on Performance Engineering, pages 327-330: Prague, Czech Republic - April 21 - 24, 2013: DOI:10.1145/2479871.2479918.

[i.74] Smirnov, M., Tiemann, J., Chaparadza, R.; Rebahi, Y., Papavassiliou, S. et al.: "Demystifying self-awareness of autonomic systems"; In ICT-MobileSummit 2009 Conference Proceedings: 10-12 June 2009, Santander, Spain; ISBN: 978-1-905824-12-0; 9 pp.

[i.75] Tele Management Forum (TMForum) ZOOM Project Standard IG1139: "Business Rationale and Technical Overview for Orchestration and Autonomic Control Loops", Release 16.0.1, published by TMF in 2016.

[i.76] Francisco da Silva: "The Impact of Artificial Intelligence and Autonomous Functions on the ETSI work-programme"; ETSI Presentation: document No. GSC-21-008: Presented ad GSC-21; 26-27 September 2017: Hosted by IEEE Standards Association.

[i.77] Fi-Ware System: European Commission (EC)-funded Future Internet Core Platform.

NOTE: Available at https://forge.fiware.org/plugins/mediawiki/wiki/fiware/index.php/.

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETSI TS 103 194 [1] and the following apply:

autonomically: by virtue of employing a control-loop (including feedback control-loop(s))

Autonomic Behaviour (AB): behaviour or action that may consist of a set of sub-behaviours or sub-actions triggered and performed by a Decision-making-Element (DE) in an attempt to achieve the goal defined by how the Decision-making-Element manages a Managed Entity (or Entities)-ME(s) under its control in a Control-Loop Structure

NOTE 1: An AB is an observable and a verifiable (testable to some extent) behaviour on interfaces of an autonomic manager element (i.e. a Decision Element). In GANA, an autonomic behaviour is considered as a behaviour of a DE, triggered as a result of reception of information from its information suppliers such as its associated Managed Entity (or Entities) in an attempt to regulate or (re-)configure the behaviour of the Managed Entity (/Entities), OR starts as a behaviour spontaneously triggered by the DE. A behaviour triggered spontaneously by a DE is simply a spontaneous transition in the Finite-State-Machine describing the overall behaviours of the DE.

NOTE 2: Example of an autonomic behaviour is: self-description and self-advertisement, self-healing, self-configuration, all triggered by a DE or multiple collaborating DEs. Therefore, it is important to note that an autonomic behaviour is bound to a DE, and possibly (though not necessarily) to information supply parts of the Control-Loop implemented by the DE together with the Managed Entity (or Entities) under the control of the DE.

NOTE 3: Autonomic Behaviour (in management and control planes) can also be viewed as a process which understands how desired Managed Entity (ME) element's behaviours are learned, influenced or changed, and how, in turn, these affect other elements, groups and network.

autonomic manager element: functional entity that drives a control-loop meant to configure and adapt (i.e. regulate) the behaviour or state of a Managed Entity (i.e. a resource)

NOTE 1: An Autonomic Manager Element can configure and adapt a managed resource like a protocol module or some other type of a Managed Entity (ME) such as a component, by processing sensory information from the managed resource and from other types of primarily required information sources and reacting to observed conditions by effecting a change in the behaviour of the managed resource to achieve some goal.

NOTE 2: In GANA, and Autonomic Manager Element is synonymously represented by a Decision-making-Element (DE).

autonomic systems with cognitive capabilities: systems which determine their behaviour in a reactive or proactive manner based on continuous observation of their environment of operation (external stimuli), as well as the goals they are required to fulfil, their principles of operation, capabilities, experience and knowledge they continue to build over time, and are characterized by presence of a control-loop structure (or structures) at the heart of their operation

NOTE: In the case of telecommunications networks, this definition means that a cognitive system has the ability to dynamically select the network's configuration, through self-management functionality that reaches optimal decisions, taking into account the context of operation (environment requirements and characteristics), goals and policies (corresponding to principles of operation), profiles (corresponding to capabilities i.e. functional features supported), and cognitive algorithms such as machine learning and deep learning (for managing and exploiting knowledge and experience).

cognition: learning, analysing and reasoning capability used to effect advanced adaptation of behaviour or state of an entity for which the capability is being employed

NOTE 1: Cognition is also considered as a combination of learning and reasoning as defined by Clark, Partridge, Ramming and Wroclawski in [i.20].

NOTE 2: A cognitive process may modify the Meta-Space and the elements of its instances (Models) during a learning process, by adding new "concepts" to the Meta-Space OR manipulating existing instances (Models) of the Meta-Space that are already known (i.e. are in the Knowledge Base) OR adding new instances (Models). The Meta-Space and instances constitute a knowledge base of the cognitive process.

Complex Event Processing (CEP): data processing discipline which correlates data from multiple sources to identify patterns of events

Context Aware Engine (CAE): functional entity designed and assigned to generate context with CEP of its execution and assigned environment

Data Model (DM): definition and format of data, including data-types and values, used for the purposes of storing or communicating the data from one entity to another

NOTE: A Data Model models data. The definition of Data Model adopted in the present document is the same as known in IETF, TMF, ITU-T, where there are a number of Data Models defined such as SNMP's SMI definitions of MIB modules, YANG Models, SID (Shared Information Data) for TMF, IRP (Integration Reference Point) for 3GPP, CMIP's Management Objects definitions, and other types of Data Models. IETF RFC 3444 [i.68] gives information On the Difference between Information Models and Data Models.

Decision Element (DE): functional entity designed and assigned to autonomically manage and control some Managed Entities (MEs)

NOTE 1: In GANA, Decision-Making-Elements (DMEs) [i.1] are also referred in short as Decision Elements (DEs) that fulfil the role of Autonomic Manager Elements (components).

NOTE 2: An ME can be a protocol or a mechanism implemented by some functional entity. MEs and their associated configurable parameters are assigned to be managed and controlled by a concrete DE such that an ME Parameter is mapped to one DE.

NOTE 3: A DE is designed and assigned to autonomically manage and control some Managed Entities (MEs). A Decision Element (DE) is an Autonomic Manager Element that implements the logic that drives a control-loop over the "management interfaces" of its assigned Managed Entities (MEs). Therefore, self-* functionalities are the functionalities implemented by the Decision Element(s).

decision plane: real made up of autonomic manager elements called Decision-Elements (DEs), and makes all decisions driving a node's behaviour (including the behaviour of all managed entities of the node) and network-wide control, including reachability, load balancing, access control, security, and network interface(s) configuration

NOTE: Replacing today's Management Plane, the decision plane operates in real time on a network-wide view of the topology, the traffic, events, context and context changes, network objectives/goals/policies, and the capabilities and resource limitations of the nodes and devices of a network of some scope (**Definition adopted but with modification, from the 4D architecture** [i.11]). The **GANA Decision Plane** encapsulates today's Vertical **Management Plane** and replaces it in the long term, and adds the Horizontal view of the Decision Plane to allow distributed DE-to-DE interactions for network-intrinsic management (for those aspects requiring (or are best addressed by) network-intrinsic management i.e. distributed control-loops).

federation: interconnection or interoperation of two or more independent administrative domains for the creation of a richer environment and for the increased multilateral benefits of the users of the individual domains

NOTE: Federation can be seen as:

- a model for the establishment of a large scale and diverse infrastructure for communication technologies, services, and applications [i.61];
- an agreement between different GANA Decision Elements (DEs) belonging to different domains. Such an agreement may concern the negotiated way the peer DEs should configure the behaviour of their assigned Managed Entities (MEs) e.g. protocols to fulfil the required network behaviour, and such DE-to-DE negotiations may be governed by some policies (the reader is referred to the Network Governance clause of the present document).

GANA node: network element (physical or virtualised) in which GANA level-2 and Level-3 DEs software has been instantiated or is being introduced to run and autonomically manage and control Managed Entities (MEs) such as networking protocols, stacks and mechanisms of the network element

Generic Autonomic Network Architecture (**GANA):** conceptual architectural reference model for autonomic network engineering, and cognition and self-management capabilities within a network and its management and control architectures

NOTE 1: The purpose of the GANA reference model is to serve as a "blueprint model" that prescribes design and operational principles of autonomic decision-making manager elements (called GANA DEs) responsible for autonomic and cognitive management and control of resources and parameters (e.g. individual protocols, stacks and mechanisms) as Managed Entities (MEs) in network elements (physical or virtual).

NOTE 2: GANA is a functional architecture and not an implementation architecture [i.1], [2] and this means it needs to be instantiated and implemented in a target implementation oriented reference network architecture and its associated management and control architecture.

holistic: property of completeness in capturing all the key architectural abstractions that should be considered in addressing the big picture of an engineering problem space

NOTE 1: The holistic nature of the GANA Reference Model as a model for autonomic networking, cognitive networking and self-management of networks and services stems from the following aspects: in autonomic computing and networking models developed in the past, the abstraction levels at which to place control-loops and distinctions between "fast-control loops" and "slow control-loops", have often not been fully defined as some models were limited to assuming and considering only control-loops outside of Network Elements (NEs), yet a holistic model is one that defines all the viable interworking abstractions at which control-loops can be designed.

NOTE 2: Some models (reviewed and now unified within a single model in the GANA Reference Model defined in the present document), initially introduced autonomics in the outer loops while assuming Network Elements (NEs) to be managed "dumb" devices, due to the assumption that was taken, of limited processing power and resources that could be available in Network Elements. A more complete model is now one that also further defines lower levels of abstractions (and nesting) at which control-loops may be introduced into the Network Elements themselves since the limitation of resources in Network Elements does not necessarily hold any longer. Meaning that, where distributed control-loops encompassing multiple network elements could be considered while at the same time accommodating interworking outer "logically centralized" loops, the GANA Reference Model presents a "holistic" view of the different possibilities as a "hybrid model". The model should combine "centralization of some Decision-Making Capabilities" while allowing also some "distributed Decision-Making Capabilities" of the network as well as some degree of self-management to be exercised by individual network elements at the lower abstraction levels of self-management. Four basic levels of abstractions for introducing control-loops are defined in the GANA Reference Model.

NOTE 3: A more complete model is now one that also further defines lower levels of abstractions (and nesting) at which control-loops may be introduced into the Network Elements themselves since the limitation of resources in Network Elements does not necessarily hold any longer. Meaning that, where distributed control-loops encompassing multiple network elements could be considered while at the same time accommodating interworking outer "logically centralized" loops, the GANA Reference Model presents a "holistic" view of the different possibilities as a "hybrid model". The model should combine "centralization of some Decision-Making Capabilities" while allowing also some "distributed Decision-Making Capabilities" of the network as well as some degree of self-management to be exercised by individual network elements at the lower abstraction levels of self-management.

information: processed data or a model that could be a result of processing data and representing it by following and respecting a Data Model, or the Model can be one that is created by humans

NOTE 1 : E.g. an Information Model, a System-Model, etc.

NOTE 2: [i.66] simplifies things (than in other approaches) by simply using one term "Information" to inclusively describe both Data and Information, hence the so-called "Information-layer".

Information Model (IM): definition of concepts representing entities that require management information and data to be defined, as well as the relationships, constraints, rules, operations, and structure in relation to the concepts

NOTE 1: The definition of Information Model adopted in the present document is the same as known in IETF, TMF, DMTF, where there are a number of Information Models defined such SID and CIM. IETF RFC 3444 [i.68] gives information On the Difference between Information Models and Data Models.

NOTE 2: The concepts and their relationships, defined by the GANA Reference Model for Autonomic Networking, Cognition and Self-Management presented in the present document, should be used to define a GANA-Metal Model (a formal model) and to extend/evolve the existing Information Models such as SID, CIM, DEN-ng, etc.

knowledge: information correlated according to a Model that is theoretically considered as a valid instance of a Meta-Space that defines the correlation among the abstract concepts of the Information Elements, or a mathematical model that establishes the correlations of inputs and outputs of a system

NOTE 1: Why "theoretically"? Because it may be difficult to achieve an ideal case/situation whereby all the elements required of correlated information are instantiated, due to the fact that some data or information from which to derive Knowledge, may be corrupted, incomplete or invalid. Therefore, an attempt to derive knowledge may result in Partial or Incomplete knowledge. Therefore, as such, Knowledge can be represented by a Model and its associated Meta-Space of which the Model is an instance. The Model is completely a valid one if it conforms to the Meta-Space that describes valid instances of the Meta-Space. The Meta-Space can be a Meta-Model or could be some Ontology and associated Schemas and together with the associated models, is provided as input to the cognitive process that claims to be able to operate on the knowledge and do something with it e.g. make some decision(s). Cognitive processes may modify the Meta-Space and the elements of its instances (Models) during a learning process, by adding new "concepts" to the Meta-Space OR manipulating existing instances (Models) of the Meta-Space that are already known (i.e. are in the Knowledge Base) OR adding new instances (Models).

NOTE 2: A mathematical model that establishes the correlations of inputs and outputs of a system may be built using cognitive algorithms, whereby algorithms such as Machine learning and Deep Learning are applied in order to build the mathematical model or to improve one.

knowledge field: sort of gravitational field emitted by each autonomic component such as a GANA DE, such that by following the local shape of the fields emitted by autonomic components, components that are able to read into the fields could perform decision-making and actuation processes, whereby the word "field" is interpreted as derived from the discipline of Physics

NOTE 1: A field emitted by an autonomic component such as a GANA DE could represent the current amount of available resources or capabilities of Managed Enties (MEs) under its scope of management and control as well as its own capabilities.

NOTE 2: Autonomic components i.e. DEs in the GANA Model could sense the knowledge field and act accordingly to specified behavioural patterns or plans.

NOTE 3: Decision-making and actuation processes that could be performed by components in relation to a knowledge field emitted by an autonomic component include the discovery of available resources or capabilities and what to do as a result.

knowledge plane: construct that exhibits cognitive capabilities and behaviours in performing the management and control of networks and services, and operates on network-wide views (including knowledge continuously gathered about the state and behaviours of network elements) to dynamically and autonomically program (configure) network resources, services and their configurable parameters- as governed by the business and technical objectives required of the network

NOTE 1: The Knowledge Plane adaptively and autonomically makes changes to the network and services compositions and configurations in order to meet the objectives (e.g. business objectives) desired for the whole network in the best optimal way.

NOTE 2: The network and services compositions the Knowledge Plane can instantiate or change include topologies and connectivity.

NOTE 3: The cognitive capabilities and behaviours of the Knowledge Plane can be achieved through Artificial Intelligence Algorithms.

NOTE 4: This definition of the Knowledge Plane is tailored to the GANA Knowledge Plane and is adopted from the definition in [i.20] by Clark, Partridge, Ramming and Wroclawski, and refined accordingly for the GANA autonomic network.

NOTE 5: The GANA Knowledge Plane consists of the following Functional Blocks: GANA Network Level Decision Elements (DEs), Model-Based Translation Services (MBTS) Functional Block, and the Overlay Network for Information eXchange (ONIX) system.

NOTE 6: The Knowledge Plane builds and maintains high-level models of what the network is supposed to do, in order to provide services (e.g. recommendations) and advice to other elements of the network, as it gathers, aggregates and acts upon information about network behaviour and operation.

learning: ability that enables the system to gradually obtain knowledge on how to handle complex situations as they emerge during the system operation

NOTE: This can increase the speed of the decision making process, and also the degree of certainty w.r.t. the quality of the decisions.

Managed Entity (ME): physical or logical resource that can be managed by an Autonomic Manager Element (i.e. a Decision Element) in terms of its orchestration, configuration and re-configuration through parameter settings

Meta-Model (MM): abstraction scope/level that is a collection of concepts (i.e. abstract representation models), their relationships and constraints among each other as they define a "domain" in form of a formal description

NOTE: For example, a car can be considered as a domain (i.e. a composite model), that consists of constituent interrelated concepts such as a wheel, an engine, etc. The domain being defined could also be a "language" e.g. a high-level language such as UML, ITU-T SDL, ITU-T MSC, or C, C++, as being defined by the particular Meta-Model. A correct instance of the Meta-Model is governed by the instantiation of the individual constituent concepts belonging to the domain in their completeness and having valid and well-formed complete links between the instantiated constituent elements. The domain, also viewed as a model, is called the "meta-level" of abstraction while different types of instances (e.g. different cars in the case of the "car domain"), are called instances or simply models. (Refer to the OMG's MDA architecture [i.67] that defines four layers of models, the lower being an instance of the intermediate upper model, starting from M0 Models, through to M1, M2 and M3 levels of abstractions (also called Models)).

Meta-Space (MS): realm that captures concepts, their relationships and constraints amongst the concepts, and is expressible formally as a model such as a Meta-Model or Ontology

NOTE: A Meta-Space can be represented as a Meta-Model or could be some Ontology and associated Schemas, and together with the associated models, it is provided as input to the cognitive process that claims to be able to operate on the knowledge and do something with it e.g. make some decision(s).

Network Governance Interface (NGI): interface to the autonomic network that is exposed to human operators and automated-management tools to enable them to provide inputs that govern the operation of the autonomic network

NOTE 1: The Network Governance Interface is an interface through which the human operator supplies to the autonomic network, e.g. by use of automated management tools, inputs such as Network Goals/Objectives, Application Intents, Service Definitions, Service Chains Definitions, Policies, Profiles and Configuration Data that the autonomic network uses to self-configure and reconfigure in case of changes to the initial input.

NOTE 2: In GANA, the Network Governance Interface is supposed to be exposed to the human by the GANA Knowledge Plane.

ontology: definition of concepts, their relationships and constraints

NOTE: In contrast to a Meta-Model, which also defines concepts, their relationships and constraints, an ontology includes both the "meta-aspects" i.e. concepts, relationships and constraints, and also their actual instances if any have been created.

overlay: logical network that runs on top of another network

EXAMPLE: Peer-to-peer networks are overlay networks on the Internet. They use their own addressing system for determining how files are distributed and accessed, which provides a layer on top of the Internet's IP addressing.

reference model: blueprint specification that defines the concepts, abstractions, concept-relationships and principles of a "domain" e.g. autonomic networking domain, as well as the Functional Blocks (FBs) and Reference Points/Interfaces between FBs, in such a generic way that implementation-oriented details are left out (because there may be diverse ways of implementing the prescribed concepts)

reference point(s): "logical interface" between at least two Functional Blocks (which include the traditionally so-called Function Blocks), that indicates that the associated Functional Blocks (FBs) communicate with each other, as peers, towards some goal

NOTE: This term comes from sources such as ITU-T SG15 and was defined for SDH Functional Architecture-whereby Atomic Function could be mapped to FB in this GANA specification. The information (messages and data) communicated by the associated FBs is called "Characteristic Information" i.e. it is information that characterizes what is communicated between the FBs. Since a Reference Point defines a logical interface in a way that is agnostic to the actual physical means/channel by which the communicating FBs are connected to facilitate their communication, a protocol (usually) or some other type of mechanism can be designed to be the "vehicle" that conveys the "characteristic information" between the FBs.

self-healing: having own ability to detect and recover from potential problems and continue to function smoothly [i.63]

self-management: intrinsic ability to manage own operation without human intervention

NOTE 1: Self-management includes the functionality required for exhibiting self-* features such as self-configuration, self-optimization, self-healing, self-protection, self-awareness, and self-organization.

NOTE 2: The term self-management is also interpreted as a general term (i.e. it is the "big-picture") describing all properties of a system falling under the umbrella of autonomic and cognition-based operations for performing the relevant operational aspects of the system. Hence, under the umbrella of self-management there are distinct methods of self-management with specific realizations and purposes in the system. The six distinct self-management methods are the following and depicted on Figure 6 of ETSI GS AFI 002 [2]. For more details, refer to [i.63] from which the definitions below were adopted:

1) Self-optimization.
2) Self-configuration.
3) Self-healing.
4) Self-protection.
5) Self-awareness.
6) Self-organization.

NOTE 3: [i.74] elaborates the self-awareness concept deeper.

self-monitoring: having ability to observe own internal state

EXAMPLE: Observation of internal state can be quality-of-service metrics such as reliability, precision, rapidity, or throughput.

self-optimization: having ability to detect suboptimal behaviours and optimize itself to improve its execution [i.63]

self-organizing: having ability to organize itself with minimum manual intervention [i.62]

3.2 Symbols

For the purposes of the present document, the following symbols apply:

DE2DE_I	DE Peer DE Interface of a DE
DE2ME_E_I	DE-to-ME Effector sub Interface of the DE-ME(s) Interface (DeMe)
DeM	Reference Point between an EMS/NMS and NEs
DeMe	Vertical Reference Point between a Node Main DE or Function level DE and its Managed Entity (ME) (in the fundamental networking resources layer of the GANA node)
E_I	Effector Sub interface of the Management Interface of an ME or a DE
FFuDe	Federation Reference point Between Function Level DEs belonging to different KPs
FMM	Federation Reference Point between F-MBTSs belonging to different KPs
FNeDe	Federation Reference point Between Network Level DEs belonging to different KPs
FNoDe	Federation Reference point Between Node Level DEs (Node-Main-DEs) belonging to different KPs
FOO	Federation Reference Point between ONIX systems belonging to different KPs
FuDe	Horizontal Reference Point between Function level DEs
FUNC_LEVEL_FWD_M_DE	Function-Level-Forwarding-And-Data Plane Management-DE
FUNC_LEVEL_GCP_DE	Function-Level-Generalized Control Plane DE
FUNC_LEVEL_MOM_DE	Function-Level-Mobility Management-DE
FUNC_LEVEL_MON_DE	Function-Level Monitoring Management-DE
FUNC_LEVEL_QoS_M_DE	Function-Level Quality of Service Management-DE
FUNC_LEVEL_RM_DE	Function-Level-Routing Management-DE
FUNC_Level_RT_M DE	Function Level Routing Management Decision Element

FUNC_LEVEL_SM_DE	Function Level Services Management-DE
GCP_M_ DE	Generalized Control Plane Management DE
GNSIR_I	General Non-Sensory Information Retrieval sub Interface of the Management Interface of an ME or a DE
GOs	Vertical Reference Point between OSS/BSS/EM/NM and Network level DEs through G-MBTS
ME2DE_GNSIR_I	ME-to-DE General Non-Sensory Information Retrieval sub-Interface of the DE-ME(s) Interface (DeMe)
ME2DE_SIR_I	ME to DE Sensory Information Retrieval sub-Interface of the DE-ME(s) Interface (DeMe)
NeDe	Horizontal Reference Point between Network Level DEs
NeI	Knowledge Plane Reference Point between Network level DE and ONIX
NeM	Knowledge Plane Reference Point between Network Element and KP through AMC-MBTS (M-MBTS)
NeMe	Vertical Reference Point between a Network Level DE and a Node Main DE
NoDe	Horizontal Reference Point between Node Main DEs (of some GANA node)
NoI	Knowledge Plane Reference Point between Node Main DE and ONIX
NoMe	Vertical Reference Point between a Node Main DE and a Function level DE
NS_I	Non-Sensory Information Retrieval Sub Management Rfp of a Management Interface of an ME or DE between a ME and a DE for non-sensor
NET_LEVEL_E2E_Service _M_DE	Network Level End-to-End Service and Applications Management
NET_LEVEL_AC_DE	Network-Level-Auto-Configuration
NET_LEVEL_FM_DE	Network-Level-Fault Management-DE
NET_LEVEL_FWD_DE	Network-Level-Data Plane and Forwarding Management-DE
NET_LEVEL_MOM_DE	Network-Level-Mobility Management-DE
NET_LEVEL_MON_DE	Network-Level-Monitoring Management-DE
NET_LEVEL_QoS_&_QoE_M_DE	Network-Level-Quality of Service and Experience Management-DE
NET_LEVEL_QoS_M_DE	Network-Level-Quality of Service Management DE
NET_LEVEL_R&S_DE	Network-Level-Resilience and Survivability-DE
NET_LEVEL_RM_DE	Network-Level-Routing Management-DE
NET_LEVEL_SEC_M_DE	Network-Level-Security Management-DE
NODE_LEVEL_AC_DE	Node-Level-Auto-Configuration and auto-discovery DE
NODE_LEVEL_FM_DE	Node-Level - Fault Management -DE
NODE_LEVEL_R&S_DE	Node-Level-Resilience and Survivability-DE
NODE_LEVEL_SEC_M_DE	Node-Level-Security Management-DE
OsDe	Vertical Reference Point between OSS/BSS and Network level DEs
OsI	Knowledge Plane Reference Point between OSS/BSS and ONIX
Other_Int_I	Other Interaction interface of a DE
Rfp_GANA-Level2&3-AccessToProtocolsAndMechanisms	Vertical Reference Point between a Node Main DE or Function level DE and its Managed Entity (ME) (in the fundamental networking resources layer of the GANA node)
Rfp_GANA-Level2(&3)-AccessToProtocolsAndMechanisms	Vertical Reference Point between a Node Main DE or Function level DE and its Managed Entity (ME) (in the fundamental networking resources layer of the GANA node)
Rfp_GANA-Level2-AccessToProtocolsAndMechanisms	Vertical Reference Point between a Node Main DE or Function level DE and its Managed Entity (ME) (in the fundamental networking resources layer of the GANA node)
Rfp_NMDE-to-Level2-DE	Vertical Reference Point between a Node Main DE and a Function level DE
S_I	Sensory sub interface of the Management Interface of an ME or a DE
SP-I	Service Providing Interface of an ME through which the ME provides a to an entity such as an upper layer entity of the protocol stack of which an ME that is a protocol entity in the stack

SR_I	Service Requesting Interface of an ME through which the ME requests for service provided by an entity such as a lower layer entity of the protocol stack of which an ME that is a protocol entity in the stack

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

3GPP	3rdGeneration Partnership Project
AB	Autonomic Behaviour
AF	Autonomic Function
AFI	Autonomic network engineering for the self-managing Future Internet
AI	Artificial Intelligence
AMC	Autonomic Management and Control
AMC-MBTS	Autonomic Management and Control MBTS - the MBTS linking the Knowledge Plane and Network Infrastructure NEs
API	Application Programming Interface
Auto-Config	Auto Configuration
BBF	BroadBand Forum
BFD	Bi-directional Forwarding Detection
BGP	Boarder Gateway Protocol
BNG	Broadband Network Gateway
BSS	Business Support System
CAE	Context Aware Engine
CEM	Customer Experience Management
CEP	Complex Event Processing
CIM	Common Information Model
CLI	Command Line Interface
CM	Cognition Module
CMIP	Common Management Information Protocol
Config-Data	Configuration Data
COPS	Common Open Policy Service protocol
CPE	Customer Premises Equipment
C-SON	Centralized Self Organizing Network
DE	Decision making Element; or simply Decision Element
DE-2-DE	DE to DE
DHCP	Dynamic Host Configuration Protocol
DHT	Distributed Hash Table
DL	Deep Learning
DME	Decision-Making-Element
E2E	End to End
EM	Element Manager
EMS	Element Management System
FB	Functional Block
F-MBTS	Federation MBTS
GANA	Generic Autonomic Networking Architecture
GANAPROF	GANA Network Profile
GENI	Global Environment for Network Innovations
G-MBTS	GANA Governance Interface MBTS
GME	Generic Modeling Environment
GMPLS	Generic Multi-Protocol Label Switching
GRASP	Generic Autonomic Signalling Protocol
GUI	Graphical User Interface
HRP	Hierarchical Reference Point
I/K	Information/Knowledge
IBN	Intent Based Networking
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGCP	ICMPv6 based Generic Control Protocol
IP	Internet Protocol
IPFIX	IP Flow Information eXport

IRP	Integration Reference Point
ISIS	Intermediate System to Intermediate System
ITU	International Telecommunications Union
K application	Knowledge plane application (as a domain of cognitive functionality)
KB	Knowledge Base
KP DE	Knowledge Plane Network level DE
KP	Knowledge Plane
KPI	Key Performance Indicator
LDAP	Lightweight Directory Access Protocol
MANO	Management and Orchestration
MAPE-K	Monitor, Analyse, Plan, Execute Control-Loop empowered by Knowledge
MB	MegaByte
MBTS	Model-Based Translation Service
MDA	Model Driven Architecture
ME	Managed Entity
MIB	Management Information Base
ML	Machine Learning
MO	Managed Object
MOF	Meta-Object Facility
MPLS	Multi-Protocol Label Switching
MSC	Message Sequence Chart
NDP	IPv6 Neighbour Discovery Protocol
NE	Network Element
Net-Level-DE	Network Level DE
NETCONF	Network Configuration Protocol
NETPROF	GANA Network Profile
NFV	Network Function Virtualisation
NFVO	Network Function Virtualisation Orchestrator
NIC	Network Interface Cards
NM	Network Manager
NMS	Network Management System
Node Config	Node configuration files
NODECONF	GANA Node Configuration with vendor specific options
NODEConfig	GANA Node Configuration
NODEPROF	GANA Node Profile
OMG	Object Management Group
ONF	Open Networking Foundation
ONIX	Overlay Network Information eXchange
OODA	Observe, Orient, Decide, and Act
OPEX	OPerational EXpenditure
OS	Operating System
OSI	Open Systems Interconnection model
OSPF	Open Shortest Path First
OSS	Operations Support System
OTT	Over The Top
PNF	Physical Network Function
PoC	Proof of Concept
QoE	Quality of Experience
QoS	Quality of Service
RACI	Responsible, Accountable, Consulted, Informed responsibility
RAN	Radio Access Network
RAT	Representation, Acquisition and Translation
RDF	Resource Description Framework
Rfp (or rfp)	Reference Point
RIF	Rule Interchange Format
RIP	Routing Information Protocol
RQ	Requirement
RSVP	Resource ReSerVation Protocol
RTCP	Real Time Control Protocol
SDAN	Software Defined Access Network
SDL	Specification and Description Language
SDN	Software Defined Networking

SEND Protocol	SEcure Neighbor Discovery Protocol
SFN	Service Function Node
SID	Shared Information Data
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SON	Self Organizing Network
SPML	Service Provisioning Markup Language
TC	Technical Committee
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol (internet model)
TMF	TeleManagement Forum
UML	Unified Modeling Language
VIM	Virtual Infrastructure Manager
VLAN	Virtual Local Area Network
VNF	Virtual Network Function
VNFM	Virtual Network Function Manager
VRP	Vertical Reference Point
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol
YANG	Yet Another Next Generation

4 Introduction and Overview of the GANA Reference Model

4.1 Snapshot of the GANA Reference Model

4.1.1 Overview

Generic Autonomic Networking Architecture (GANA) reference model specifies the concepts and principles defining the domain of autonomic communication, autonomic networking, autonomic and cognitive management and control-all as part of the "big-picture" of Self-Management. Figure 1 presents the GANA abstraction levels for self-management functionality at which interworking hierarchical/nested control-loops and their associated DEs can be designed. Figure 1 defines the key GANA Functional Blocks (FBs) for enabling and implementing autonomics in target implementation-oriented architectures, as described in the clauses below. A table of a summary of all GANA FB Reference Points (Rfps) and characteristic information descriptions is given in clause 13 of ETSI GS AFI 002 [2], and additional Reference Points (Rfps) have been added in the further elaboration of the GANA model as described in the present document. All the Reference Points (Rfps) are further described in the present document in clauses that follow later in the present document. As such, ETSI GS AFI 002 [2] shall be considered as being complementary to the present document.

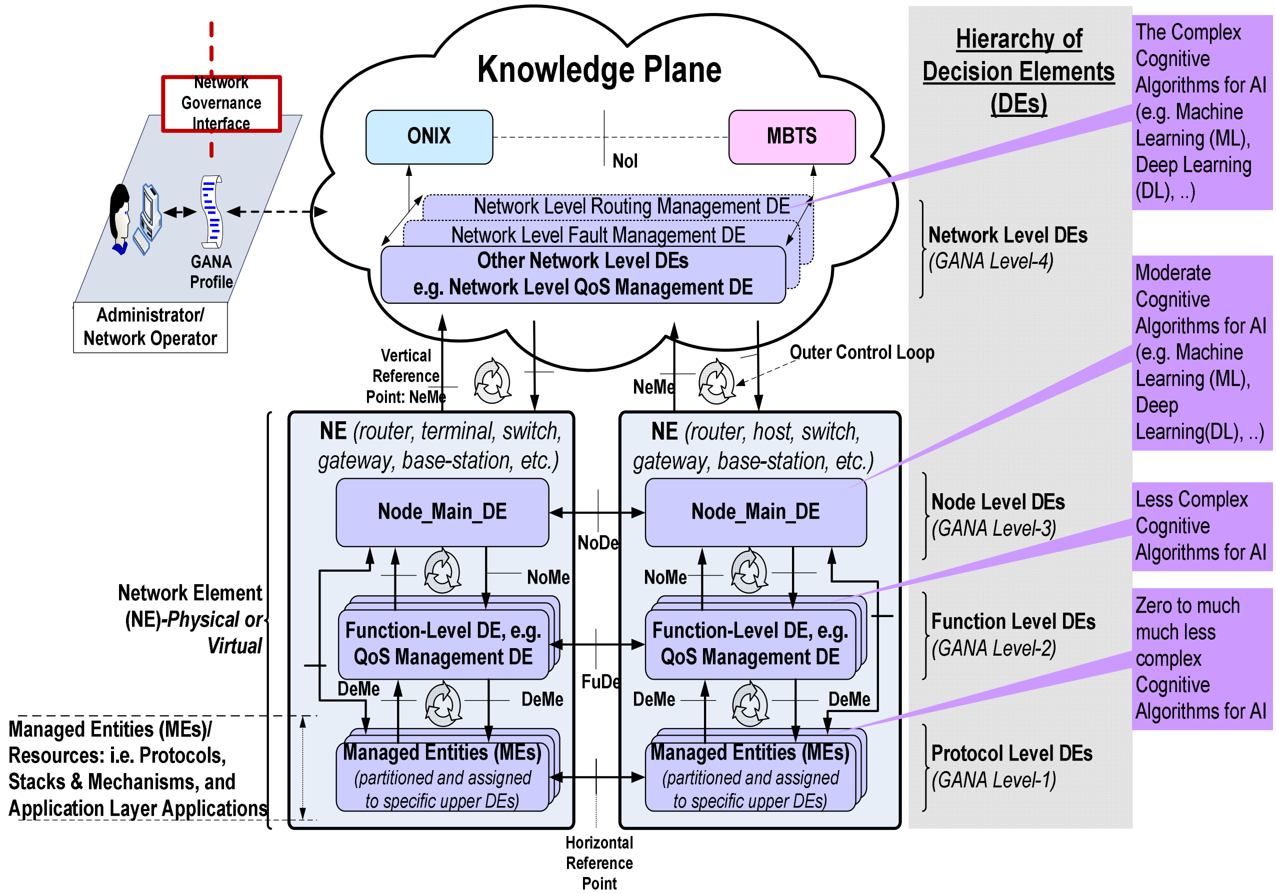


Figure 1: Snapshot of the GANA Reference Model

The GANA is a Reference Model for Autonomic Networking, Cognitive Networking and Self-Management. The aspects of autonomic networking, cognitive networking and self-management of networks and services are covered in GANA collectively as the AMC (Autonomic Management and Control) paradigm. AMC is about Decision-making-Elements (DEs) as autonomic functions (logics that dynamically configure their associated managed entities in respective closed control-loops) with cognition introduced in the management plane as well as in the control plane (whether these planes are distributed or centralized).

Cognition (learning, analysing and reasoning used to effect advanced adaptation) in DEs, enhances DE logic and enables DEs to manage and handle even the unforeseen situations and events detected in the network.

Control (in "AMC") refers to the control-loop logic kernel of the DE, capable of dynamically adapting network resources and parameters or services in response to changes in network goals/policies, context changes and challenges in the network environment that affect service availability, reliability and quality.

DEs realize self-* features of a functionality or system (self-configuration, self-optimization, etc.) as a result of the decision-making behaviour of a DE that performs dynamic and adaptive management and control of its associated Managed Entities (MEs) and their configurable and controllable parameters. Such a DE can be embedded in a Network Element (NE) or higher at a specific layer of the outer overall network and services management and control architecture-thereby creating AMC architecture (composed of nested and hierarchically stacked DEs that can also collaborate horizontally across management and control planes). An NE may be physical or virtualised (such as in the case of the NFV (Network Functions Virtualisation) paradigm). Network functionality such as routing, forwarding, mobility management, etc. can be made autonomic by embedding a DE. DEs (as software components) are meant to empower the networks and the management and control planes to realize self-* properties:

- auto-discovery of information/resources/capabilities/services;
- self-configuration;
- self-protection;

- self-diagnosis;
- self-repair/heal;
- self-optimization;
- self-organization behaviours;
- as well as self-awareness.

The GANA reference model [3] and [2], as AMC architectural model, defines generic Functional Blocks (FBs) and their associated reference points and characteristic information (messages conveyed through those reference points). All of which are specific to enabling autonomics, cognition and self-management in a target architecture when instantiated onto an implementation-oriented reference architecture such as BBF (Broadband Forum) architecture, NGN/IMS architecture, or 3GPP architecture, or even on any future network architecture.

NOTE 1: ETSI is performing a number of GANA instantiations onto various target standardized reference network architectures to enable autonomics algorithms implementers to design and implement DEs in those architectures in standardized way. Example GANA instantiations are GANA instantiation onto the 3GPP Backhaul and Core network architectures (ETSI TR 103 404 [i.27]), GANA instantiation onto the Ad-Hoc/Mesh network architecture (ETSI TR 103 495 [i.6] and [i.29]).

NOTE 2: There is ongoing work in ETSI on GANA instantiation onto the BBF architecture scenarios (ETSI TR 103 473 [i.28] is work in progress).

Therefore, distinctively performed GANA instantiations enable to create various types of autonomics-enabled reference architectures, e.g. autonomics-enabled BBF architecture, etc.

NOTE 3: Autonomics is also synonymously referred to as autonomicity in literature.

The GANA reference model can also be applied in designing future network architectures that shall exhibit self-management capabilities from the outset of their design. GANA is a generic model in the sense that it defines and separates generic concepts and associated architectural principles for the domain of autonomic networking, cognitive networking and self-management from implementation strategies, details and methods that can be used to implement them. Hence, it is not constrained by any implementation-oriented architecture and to the extent possible avoids inheriting the limitations of today's technology-specific network architectures.

4.1.2 Concept of GANA DE (autonomic Decision-making-Element)

The GANA model serves as a blueprint model that defines and prescribes the design and operational principles of autonomic Decision-making Elements (as "autonomic manager" components) called DEs. DEs are also called "Autonomic Functions" (AFs) and are responsible for autonomic management and adaptive control of systems and network resources, parameters, and services. Autonomic behaviours of a DE (also called autonomic manager) include secure auto-discovery of the following items: network objectives and policies specified by the human operator, other DEs it requires to collaborate with, and capabilities of the DE's assigned Managed Entities (MEs)-i.e. the information that gets available at run-time. Then after auto-discovery, a DE performs the self-* operations on its assigned MEs (by-design) by orchestrating (launching and/or configuring) the MEs when required, and adaptively (re-)programming the MEs via the effectors of their management interfaces. Orchestration means launching (at run-time) an entity (e.g. an ME) if no instance exists yet that can provide a desired service and then configuring the instance (the newly launched or an already existing one) such that it is ready to provide a service of which the entity is designed to provide. A DE is designed to perform one or multiple self-* operations such as self-configuration, self-diagnosing, self-healing, self-repair, self-optimization, self-protection, etc. Some specialized DEs may be designed to perform certain self-* operations on a macroscopic level that takes into account wider perspectives needed to complement the same self-* operations intrinsically performed by DEs on the microscopic level. Therefore, in general, a DE is said to realize or implement self-* features. What drives a DE to perform its operations in collaboration with other DEs whenever required are various input information and changes that drive its algorithms (e.g. machine learning, deep learning, computational science and other types of decision-making algorithms or Artificial Intelligence (AI) algorithms), such as changes in the operational state of its ME(s), changes in the governing input policies, context changes, and challenges (e.g. faults, errors, failures) detected in the operation of the MEs and the underlying network substrate. The collective and collaborative DEs' self-* operations on orchestrating and adaptively programming (configuring) the various MEs in the network so as to achieve a global network objective w.r.t desired instantaneous operational state of the MEs amount to the notion of self-organization of the network and its management and control operations.

Regarding cognitive algorithms that can be designed for DE logic, Figure 1 illustrates the complexity (degree) of cognition as a property in DEs that should be expected at particular levels in GANA, showing that cognition in individual protocols (GANA Level 1) may vary from zero (none) to much less complex cognitive algorithms, while higher up the DEs Hierarchy cognitive algorithms become more complex as expected by the widening scope of inputs data/information the higher level DEs process and use in decision-making. This subject is discussed in more detail in the clauses that follow later in the present document (e.g. in the clause 4.7 on "A Consolidated Characterization of the GANA Knowledge Plane, and Cognitive Algorithms for Artificial Intelligence (AI) in Autonomic Functions (DEs)").

As discussed in the ETSI White Paper No.16 [3], DE algorithms, just as in the case with SON Functions algorithms, may not be standardized as they should provide the means for DE vendor differentiation so as to facilitate for DE vendor differentiation-as there is a need to promote continuous innovation in autonomics algorithms for various types of DEs.

As discussed in ETSI GS AFI 002 [2], approaches such as FOCALE principles [i.10], IBM's MAPE-K control-loop model [i.12], and other principles validated in various research projects on autonomics can be applied to designing a DE's internal structure and logic and associated control-loops. Such principles applicable to designing DE internal structure and logic include the **O**bserve, **O**rient, **D**ecide, and **A**ct (OODA) control-loop model [i.65] and [i.75].

NOTE: [i.75] discusses efforts on integration of GANA with the other emerging complementary networking paradigms of SDN, NFV, Big-Data Analytics, and E2E Service Orchestration.

The GANA fuses a number of leading autonomics efforts/models, including FOCALE [i.10], IBM-MAPE-K [i.12], 4D architecture [i.11], Knowledge Plane for the Internet [i.20], and other models, as a unified holistic reference model for AMC. This subject, on how concepts from the various models are unified and fused together (accommodated) in GANA is discussed in the present document and complemented by more details in ETSI GS AFI 002 [2].

4.1.3 Concept of GANA MEs (Managed Entities)

GANA adopts the concept of a Managed Entity (ME) to mean a managed resource, instead of a managed element- a term used in traditional network management terminology and normally intended to mean only a physical Network Element (NE) and not some functional entity within a node/device such as a protocol module or a component (e.g. a monitoring component).

MEs can vary: they can be fundamental MEs at the bottom of the management hierarchy (at the fundamental resources layer, see Figure 1) such as individual protocols or stacks, OSI layer 7 or TCP/IP application layer and other types of resources or managed mechanisms hosted in a network node (NE) or in the network in general. MEs can also be composite MEs such as whole NEs themselves (i.e. MEs that embed sub-MEs). It is noted that an NE can be physical or virtual. Managed Entities (MEs) at the bottom level (the fundamental resources layer in GANA): at the bottom of the management andcontrol hierarchy (see Figure 1) are the fundamental MEs that are hosted in a network node (NE) and can be employed, orchestrated, configured and dynamically adapted to achieve some goals. These form the fundamental resources layer MEs. The fundamental MEs at the resources layer often have their management interface or Managed Objects (MOs) specified using data modelling structures such as MIBs (Management Information Bases). Such definitions of MOs can be part of the basis upon which DEs that dynamically infer state of the MEs and perform dynamic ME parameter configuration, can be designed to operate. In addition to the MOs other views/info that are external to the MEs are also relevant to be used by the DEs' logic/algorithms (as discussed in ETSI White Paper No.16 [3]).

NOTE: As illustrated on Figure 1 there are certain MEs that should be directly orchestrated and managed and controlled by the Node-Main-DE as their resultant configuration and behavior governs the node's behavior on a global level. An NE is represented by a concept called GANA Node in the GANA reference model. The GANA Node defines the place-holders for NE internal control-loops. GANA node level enables designing and embedding "node-local" self-management algorithms and in-network self-management through distributed algorithms for DE-to-DE interactions horizontally across a network of some scope.

The GANA Node shall be governed by the GANA Knowledge Plane.

4.1.4 Why call the behaviours of DEs self-* features?

The behaviours of DEs are called self-* features because humans are relieved from having to perform the traditionally manual management-oriented tasks, and software, i.e. the DEs, automate the tasks and dynamically perform the tasks based on human specified networking goals and policies, context or state changes, and events detected in network nodes and the network.

A DE should receive as input network goals or governance policies specified by the human operator, and also auto-discover other DEs the DE requires for DE to DE collaboration, and the capabilities of its assigned managed entities (MEs) before the DE starts performing the self-* operations it is designed to perform. Such DE behaviours should be performed by individual DEs embedded in NEs (for self-management objectives and operations that should be driven by the need for local reactions within the NE). DEs operating on the macroscopic level outside of NEs should perform similar behaviours while at the same time are expected to control the lower level DEs in NEs via policy control as elaborated in more detail later in the clauses of the present document.

***Horizontal ("Distributed") DE-to-DE collaboration*:** Some DE algorithms may require the collaboration of a DE within an NE with other DEs along an end-to-end (E2E) path in the network, though not necessarily involving hop-by-hop NEs, for a self-* operation (e.g. self-optimization) that may require the collaboration of distributed DEs along a path in the network.

***Vertical ("Hierarchical") DE-to-DE collaboration*:** Another possibility is that actions of an NE's DEs may also need to be synchronized by higher level autonomic behaviours coordinated by upper layer DEs (outside the NE) at the GANA Knowledge Plane (KP) level (KP concept is defined below).

4.1.5 Concept of Knowledge Plane (KP) in the GANA Model

The GANA Knowledge Plane (GANA KP) enables advanced management andcontrol intelligence at the Element Management (EM), Network Management (NM) and Operation and Support System (OSS) levels by interworking with them or enhancing and evolving the intelligence of the systems at these levels by way of replaceable and (re)-loadable autonomics modules (DEs) that can be loaded at specific abstraction levels of management and control operations (more details in ETSI White Paper No.16 [3], and in ETSI GS AFI 002 [2]).

The GANA KP concept is inherited from the Knowledge Plane concept defined in [i.20] as a pervasive system within the network that builds and maintains high-level models of what the network is supposed to do, in order to provide services and advice to other elements of the network. In order to consolidate and articulate all the key design and operational principles and enabler paradigms for the Knowledge Plane such as autonomics and cognitive capabilities an elaborate definition of the Knowledge Plane provided in the "Definitions" clause of the present document is tailored to the GANA Knowledge Plane concept. As illustrated in Figure 1, GANA KP's DEs should be complementarily designed to collaborate with DEs at lower autonomics-abstracting layers of the GANA model.

GANA Knowledge Plane Functional Blocks (FBs): The **GANA KP** (defined earlier) consists of the following FBs, namely:

- **Network Level DEs** whose scope of input is network wide. They are designed to operate the outer closed control loops on the basis of network wide views or state as input to the DEs' algorithms and logics for autonomic management and control (AMC).
- **ONIX (Overlay Network for Information eXchange)** (distributed scalable overlay system of federated information servers). The ONIX is useful for enabling auto-discovery of information/resources of an autonomic network via "publish/subscribe/queryandfind" protocols. DEs can make use of ONIX to discover information and entities (e.g. other DEs) in the network to enhance their decision making capability. More details on ONIX are given in the present document and are complemented by ETSI GS AFI 002 [2]. The ONIX itself does not have network management and control decision logic (as DEs are the ones that exhibit decision logic for AMC).

- **MBTS (Model-Based Translation Service)** which is an intermediation layer between the GANA KP DEs and the NEs (physical or virtual) for translating technology specific and/or vendors' specific raw data onto a common data model for use by network level DEs, based on an accepted and shared information/data model. KP DEs can be programmed to communicate commands and process NE responses in a language that is agnostic to vendor specific management protocols and technology specific management protocols that can be used to manage NEs. The MBTS translates DE commands and NE responses to the appropriate data model and communication methods understood on either side. More details are given in the present document and are complemented by ETSI GS AFI 002 [2]. The concept of MBTS (Model Based Translation Layer) in FOCALE [i.10] uses a common model that is also accommodated and unified together with other models into the GANA Model. The value the MBTS brings to network programmability is that it enables KP DEs designers to design DEs to talk a language that is agnostic to vendor specific management protocols and technology specific management protocols that can be used to manage NEs.

A consolidated characterization of the GANA Knowledge Plane is provided later in the present document.

4.1.6 The value in having Decision-Elements/Engines (DEs) forming a Decision Plane Hierarchy for AMC

GANA DE design is based on the need for hierarchical abstraction levels for self-management (autonomic) functionality. The hierarchy of DEs is meant for implementing AMC of MEs, at various levels of abstractions of self-management functionality, with DEs viewing their assigned lower DEs in the management and control hierarchy as MEs of some sort (since upper DEs manage and control lower level DEs, using policy control for example). The GANA defines four levels of self-management functionality, i.e. levels of abstractions at which DEs and their autonomics control-loops can be designed. The levels are described in more detail in the corresponding clauses that follow in the present document. Lower GANA level DEs are viewed as MEs by upper GANA level DEs, which inductively control lower level DEs through policies.

NOTE 1: It does not mean that in order to implement Autonomic Functions (AFs), meaning DEs, every hierarchical level in GANA has to be implemented in the target architecture, due to the fact that autonomics introduced at a particular level has its own value as described and illustrated in the GANA model.

Because in incremental implementation of autonomics in a network architecture, one particular GANA level or multiple GANA levels and associated DEs may be collectively considered at a time. A full implementation of interworking autonomics at multiple levels (especially Level-2 to Level-4) may simply emerge over time. More discussions on this subject are provided later in the clause on the Implementation Guide for GANA in the ETSI White Paper No.16 [3].

NOTE 2: There is ongoing work in ETSI on elaborations of the Implementation Guide for GANA and the resultant work is expected to be published in ETSI TR 103 195-3 [i.71].

The value of autonomic behaviour driven by a particular DE instantiated in a particular environment is in the DE's behaviour in monitoring the events/views exposed by its assigned MEs and their execution state, reasoning about the exposed views together with any other input from the environment required by the DE's algorithms. Then deciding on whether to (re)-configure their MEs and their parameters so as to achieve certain objectives of local scope to the DE or requiring collaboration with other DEs. DEs may collaborate horizontally or vertically in the decision plane (see Figure 1). This value of autonomics will always continue to be the subject of research and innovation. In some network environments there may be certain network and policy control dynamics that require certain DEs and associated control-loops (for purpose of collaboration) to be instrumented.

NOTE 3: Communication methods (e.g. protocols) for DE-to-DE communications horizontally or vertically in the Decision Plane, outside of NEs, would need to be developed in SDOs such as IETF [i.13], if existing protocols of today cannot be applied for DE-to-DE communications outside of NEs.

ETSI GS AFI 002 [2] provides guidelines on principles that can be applied for designing DE logic, including external and internal structural models of a DE, interfaces and primitives that should be supported and implemented on DE interfaces. The principles are also descripted in the present document.

Therefore, in general, self-manageability of a network and services using GANA is achieved through instrumenting the NEs and an outer realm called the GANA KP with DEs that collaboratively work together in realizing self-* features in nodes and a network as a whole. Nodes' DEs may form peers along a path within the fundamental E2E transport/data-plane architecture (see Figure 1 and other figures that follow later in the present document as they provide more details on the subject).

In short, GANA design principles for autonomic management and control of networks and services involve the following primary principles:

- Abstractions for self-management at various levels of the global decomposition of the space constituted by networked systems and their associated management and control architectures.
- Hierarchical and nesting of control-loops operating at the various abstraction levels for self-management functionalities/capabilities.
- Time-scaling in the operations of the control-loops for autonomics.
- Provision of methods and techniques for ensuring convergence and stability of the various levels of autonomics in achieving desirable aggregate global objectives (from "micro-autonomics" at lower levels, i.e. within network nodes, to "macro-autonomics" in the outer higher levels of the abstractions space).

Such principles embraced by the GANA have been well studied and validated in implementations of complex large scale systems as demonstrated by the maturity of research on autonomic and autonomous systems organizations and federations on large scale.

EXAMPLE: [i.43] describes in a broader picture the Theory of Enterprise Command and Control, covering various principles such as those employed in GANA, such as Hierarchical Control Structures, hierarchical and nesting of autonomics control-loops (with inclusion of what can be an inspiration from the autonomic nervous system of humans), cooperation and federation of command and control, cooperative and collaborative structures and organization decomposition levels. Various design principles described in [i.43] and many other sources can be considered in implementing GANA autonomics in various GANA instantiation and implementation cases.

4.1.7 GANA Network Governance Interface for Governing the Autonomic Network

The network governance interface is an interface through which humans (through the support of automation tools) generate GANA network profiles (GANA profiles) to be provided as input to the autonomic network through the GANA KP. With the help of automated management tools, the human operator (administrator) creates a GANA profile in form of a data structure that contains the networking objectives/goals, policies and certain types of configuration data that govern the autonomic network. The network profile is then used by the autonomic network to generate lower level policies and configuration data to configure the DEs and the network.

4.1.8 GANA as a holistic unifying model for the well-established models for AMC

GANA accommodates and unifies concepts from the well-established models for autonomics. The ETSI Group fused a number of leading autonomics efforts/models, including FOCALE [i.10], IBM MAPE-K [i.12], 4D architecture [i.11], Knowledge Plane for the Internet [i.20], GENI (Global Environment for Network Innovations) [i.16], CONMan (Complexity Oblivious Network Management) [i.17] and other models, and developed the GANA, as a unified reference model for AMC. This subject, on how concepts from the various models are unified and fused together (accommodated) in GANA is discussed in clause 9.12 of ETSI GS AFI 002 [2].

4.1.9 Summary of the GANA Abstraction Levels for Self-Management (Autonomic) Functionality

The GANA reference model defines a hierarchy of DEs. It is organized in four basic hierarchical abstraction levels of self-management (autonomic) functionality (presented below in a bottom-up approach):

- Protocol level DE; (lowest level of abstraction for autonomics in GANA).
- Function level DE.
- Node level DE.
- Network level DE (highest level).

Each of the GANA Abstraction Levels for Self-Management (autonomic) Functionality is described in the clauses that follow.

4.2 Relationships between GANA and other complementary networking paradigms: SON, SDN, NFV, E2E Orchestration, Network Analytics, Big-Data, and other

The GANA as a reference model for the AMC paradigm is complementary to other emerging networking paradigms as described below:

- SON (Self-Organizing Networks): The GANA model shares common principles with the Hybrid-SON (Self-Organizing Network) architectural model, as both models enable developers of algorithms to combine and interwork centralized and distributed management and control solutions for networks and services. More details on how the Hybrid SON Model conforms to the GANA Model are in ETSI White Paper No.16 [3].

- SDN (Software Defined Networking): [i.14], [i.15] and ETSI White Paper No.16 [3] provide details on how the GANA integrates with SDN. [i.2] and [i.75] also provided additional insights on this subject.

- NFV (Network Functions Virtualisation): [i.15] and ETSI White Paper No.16 [3] provide details on how the GANA integrates with NFV and NFV layers (including MANO). [i.2] and [i.75] also provided additional insights on this subject.

- E2E Service and Resources Orchestration: ETSI White Paper No.16 [3] provides details on how the GANA integrates with E2E Service Orchestration and Resources Orchestration [i.2] and [i.75] also provided additional insights on this subject.

- Network Analytics and Big-Data Analytics for AMC: ETSI White Paper No.16 [3] provides details on how the GANA integrates with Big-Data Analytics for AMC, and how the GANA Knowledge plays a role in complementing Network Analytics performed on data collectors or management and control systems. In the present document further details are provided on how the GANA Knowledge Plane should fulfill the combined role of Network Analytics Driven Service Orchestration and Network Analytics Driven Closed-Loop (Autonomic) Service Assurance [i.2] and [i.75] also provided additional insights on this subject.

- Closed-Loop (Autonomic) Service Assurance: [i.9] provides an outlook into how the GANA fits into the big picture on closed-loop automation and the new role of assurance, as well as the evolution of service assurance towards closed-loop (autonomic) service assurance. The present document provides details on how the GANA helps achieve Closed-Loop (Autonomic) Service Assurance.

- White Box Networking: ETSI White Paper No.16 [3] provides details on the fact that GANA DEs software cans be designed in such a way that such software can be loaded into white box devices to empower them with autonomics. DEs may easily be "loaded" (or even replaced after) into an NE (physical or virtual), and this implies the enrichment of the notion of "Software-Driven or Software-Empowered Networks.

NOTE: ETSI is working on a document that is expected to provide more insights on GANA integration with SDN and NFV within the context of BroadBand Forum (BBF) architectures, namely work in progress in ETSI TR 103 473 [i.28].

4.3 GANA Abstraction Levels for Self-Management (Autonomics), Functional Blocks (FBs) and Reference Points; GANA Instantiations Guide and Examples/Cases

This clause presents an overview on GANA Abstraction Levels for Self-Management (Autonomics), Functional Blocks (FBs) and Reference Points in GANA, Network Governance in GANA, Federated AMC in GANA, and GANA Instantiations cases onto various types of implementation-oriented reference network architectures.

Figure 2 depicts a global view of key Functional Blocks (FBs) and associated Reference Points (Rfps) of the GANA Reference Model. These FBs are specific to enabling autonomics, cognition, and self-management in a target architecture, when instantiated onto an implementation-oriented reference architecture and its management and control architecture such as those defined by Standardization Organizations (SDOs/Fora) (e.g. 3GPP, BBF, ETSI, ITU-T, TMF, IEEE, ONF and other SDOs/Fora). GANA reference model shall also be applied in designing future network architectures that exhibit self-management capabilities from the dawn (outset) of their design.

In general, self-manageability in GANA is achieved through instrumenting (equipping) Network Elements (NEs) with autonomic Decisions making Elements (DEs) that collaboratively work together and automate network operations by implementing control loops for self-adaptive network management and control.DE software logic drives closed control-loop over its MEs. Such control loops operate using information/knowledge regarding contexts, detected events and the state of MEs. GANA defines the concept of "ownership" meaning a given DE shall manage only their associated MEs and drives its control loop based on a continuous learning cycle. DEs may form "peers" along a path within the fundamental end to end transport network architectures (the data plane in general), and this also applies to DEs in the outer control plane of the network (e.g. DEs in the GANA Knowledge Plane) and in the service layer. The DE-to-DE peers need not necessarily be hop-by-hop neighbours (i.e. being resident in on-link neighbouring nodes) but the peer relationships may relate to e.g. border-relationships management in a heterogeneous network or may relate to some DEs in certain NEs along an end to end path.

At the same time, the DEs continuously receive local "views" of their MEs, such as detected events the DE may need to act upon. DEs may also be designed to operate on other information/knowledge from other required or potential information suppliers of DEs, such as databases and the environment in which the device hosting the DE is operating. DEs may also be designed to subscribe to receive information published by other entities into the Overlay Network Information eXchange (ONIX) system of Information Servers. All the various information/knowledge retrieved from the MEs and other information/data sources is collectively used by DEs to change the behaviour or configuration state of their associated ME(s) in order to achieve and maintain the goals known by DEs as their targets.

The human network operator should govern the autonomic network using what is called a GANA Profile (GANAPROF), which encapsulates the goals and objectives of each DE, policies and other configuration data as described later in the present document. The GANA governance interface (described later in the present document) should allow the human operator to control each DE (particularly the higher-level DEs in the Knowledge Plane) and to initiate DE orchestrations and configure them to operate either in "open-loop" or "closed-loop" mode. GANA operational principles advocate for the human operator to be equipped with tools-chains (e.g. OSS/BSS) for governing the autonomic network in such a way that input to the autonomic network, e.g. policies and goals, is "conflict-free". This is described in the clauses on Governance Interface and GANA Operations Procedures for the Human Network Operator.

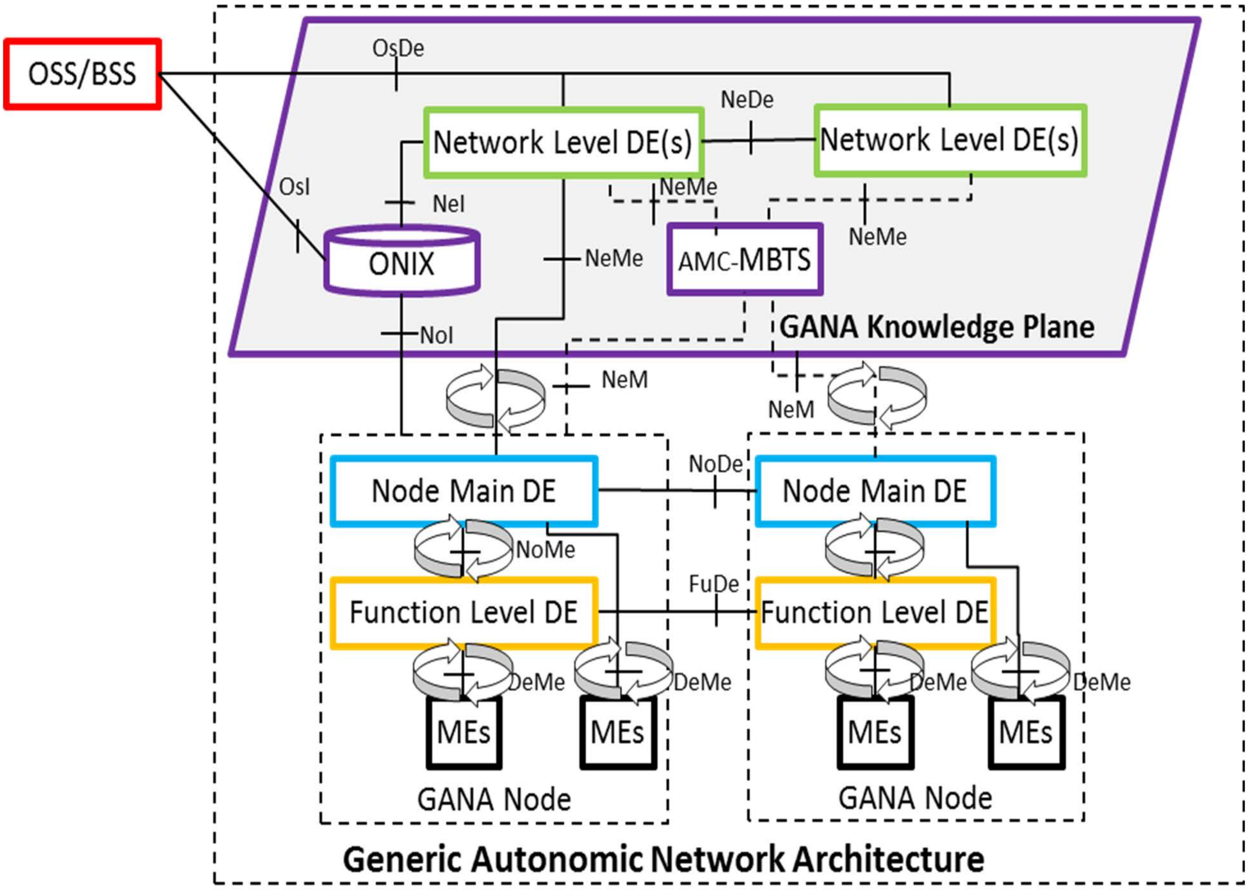


Figure 2: Global view of the GANA Reference Model

The place-holders for internal control-loops in a GANA Node, an autonomic NE, are depicted in Figure 2 and Figure 3. The GANA node level enables designing and embedding "node-local" self-management algorithms.

NOTE 1: While I/K Repository is a generalized Information Repository, a Knowledge Base (KB) Repository can be considered as part and parcel of the I/K Repository, and Figure 3 illustrates that DEs of the node may use a shared I/K Repository.

Each DE in an NE manages its associated MEs and lower level DEs through a control loop. Lower-level DEs are therefore considered as Managed Entities (MEs). Each DE realizes some specific control-loop(s) with own algorithms, logic, and therefore, represents an "Autonomic Activity" or an "Autonomic Function" (AF). The GANA Node shall be governed by the GANA Knowledge Plane DEs.

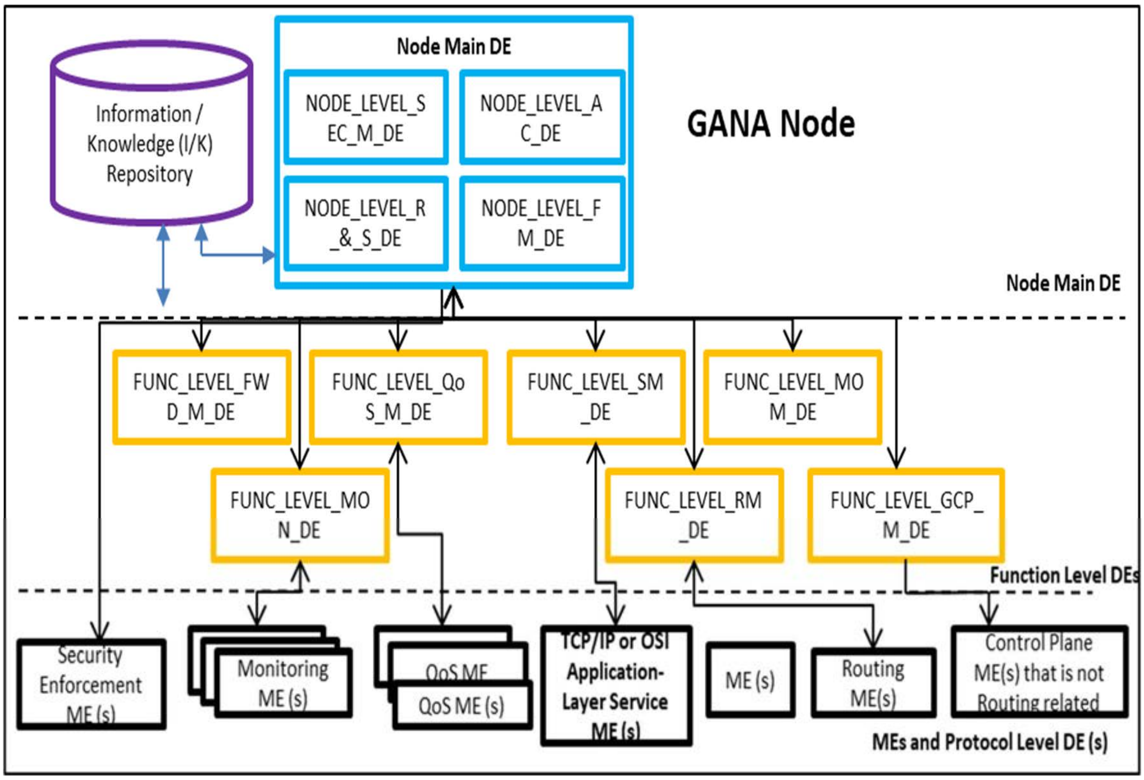


Figure 3: GANA Node Structure

NOTE 2: Insights on GANA Implementation Guide that implementers need to consider: Possible implementation approaches that can be pursued for DEs in the GANA node are as follows: GANA Node DEs may be implemented as standalone run-time entities or merged to run as single process at run-time. Quoting ETSI White Paper No.16 [3]: "*It does not mean that in order to implement Autonomic Functions (AFs), meaning DEs, every hierarchical level in GANA has to be implemented in the target architecture. Because in incremental implementation of autonomics in a network architecture, one particular GANA level or multiple GANA levels and associated DEs may be collectively considered at a time. A full implementation of interworking autonomics at multiple levels (especially Level-2 to Level-4) may simply emerge over time*". More discussions on this subject are provided in the clause on the Implementation Guide for GANA in ETSI White Paper No.16 [3].

NOTE 3: There is ongoing work in ETSI on elaborations of the Implementation Guide for GANA and the resultant work is expected to be published in ETSI TR 103 195-3 [i.71].

4.4 GANA Abstraction Levels for Self-Management (Autonomics) Functionality: GANA's four basic abstraction levels for Hierarchical Control-Loops

4.4.1 The Four Basic Levels of Abstractions for Designing Self-Management Functionality and associated Hierarchical (Nested) Control-Loops

GANA defines four basic hierarchical control loops levels of abstractions at which autonomicity by associated Decision Elements (DEs) may be introduced within the network, namely:

- Protocol-Level.
- Function-Level.
- Node-Level and Network-Level (in the realm called the GANA Knowledge Plane).

However, the three GANA levels, namely the Protocol-Level; Function-Level; Node-Level and Network-Level, as these abstraction levels involve introducing management and control software in an overlay manner outside and above protocol stacks without needing any changes to existing networking protocols. Annex A of the present document provides a summary of the four levels of abstractions for autonomics defined in the GANA, even though the level-2, level-3 and level-4 are recommended to be the main focus for designing DEs and autonomics control loops as argued in ETSI White Paper No.16 [3]. The abstractions levels for self-management (autonomics) functionality further described below pertain to how "micro-level" (low-level) autonomics and its interworking with "macro-level" (higher-level) autonomics can be introduced in a network architecture and its associated management and control architecture.

4.4.2 GANA Level-1 DE (the lowest), called a "Protocol" Level DE

Protocol Level DE relates to any Managed Entity (ME) such as protocol, network service enabler, software components, and other fundamental mechanisms that may exhibit intrinsic control-loops and associated inbuilt DE Logic, as it is the case for most of today's protocols such as the routing protocol OSPF (Open Shortest Path First). OSPF may be considered as an example of the instantiation of a Protocol-Level-DE. Since the Protocol-Level-DE involves embedding an intrinsic control loop within an individual protocol, it may not be to introduce such "intelligence" into individual protocols. Avoiding "protocol-intrinsic control-loops" is recommended since they may complicate network manageability and may create undesired emergent behavior in complex protocol interaction scenarios as known today (refer to ETSI White Paper No.16 [3] for further discussion on this subject). Therefore, GANA hierarchy covers only the three other levels which should collaboratively work together.

4.4.3 GANA Level-2 DE, called a "Function"-Level DE

This Function level DE manages entities in the networking resources layer of a Network Element (NE), including those that embed Protocol-level DE logics. The Function Level DEs relates to a DE for collective autonomic management and control of a group of protocols and mechanisms that are abstracted (viewed like a bundle) by a "networking function" or a "management/control function". For example, a "routing" function abstracts all the routing protocols (e.g. OSPF, BG, RIP) and mechanisms supported on the NE that can realize the routing function. A "mobility management" function collectively abstracts all the mobility enabling protocols and mechanisms supported on the NE that can realize the "mobility management" function. Similarly, for the "QoS management" function, etc.

The following DEs are the examples of the DEs identified as necessary to be introduced at this level in GANA:

- "routing management-DE";
- "forwarding management-DE";
- "Quality of Service management-DE";
- "mobility management-DE";
- "monitoring management-DE";
- "service and application management-DE".

Currently these six Function Level DEs are defined in the GANA Reference Model as fundamental DEs that can be instantiated to run in NEs depending on the types of MEs of an NE. The control-loop is external to the individual protocols and mechanisms subscribed to the "Function" (by virtue of abstraction).

DEs that can be instantiated into a node (NE) at this level are determined by the "networking functions" required to be operationally supported by the node (NE) and realized by specific Managed Entities (MEs) such as protocols and mechanisms or tools associated with the networking functions. More details on this subject are found in ETSI White Paper No.16 [3].

NOTE: There is ongoing work in ETSI on elaborations of the Implementation Guide for GANA and the resultant work is expected to be published in ETSI TR 103 195-3 [i.71].

4.4.4 GANA Level-3 DE -called the Node-Main-DE

The Node-Main-DE orchestrates and manages DEs on the Function Level DE and also manages MEs in the networking resources layer that are not to be managed by the Function-level-DE but required to be managed on the node-level as described in ETSI GS AFI 002 [2] and in ETSI White Paper No.16 [3]. The Node-Main-DE relates to a DE for autonomic management and control of those aspects that cover and restrict the behavior of the node as a whole, as well as the orchestration and policing of the "Function Level-DE".

The following four DEs have been identified at this Node level DE:

- Security Management DE.
- Fault Management DE.
- Auto Configuration and Auto-Discovery DE.
- Resilience and Survivability DE.

All four Node Main DEs level shall be present in any GANA Node, an autonomic NE.

4.4.5 GANA Level-4 DE (the highest)-called a "Network" Level-DE

The Network Level DE relates to a DE for autonomic management and control of those aspects that cover network-wide views and the management andcontrol of the GANA Nodes which embed lower levels DEs (node level DE, function level DE and MEs at the protocol level DE). The Control-Loops at this level complement lower level control loops by operating on slower timescale (i.e. they are slower control-loops in contrast to lower level control-loops (the faster control-loops)).Therefore Network Level DEs shall govern all the DEs of the GANA Nodes by policy-control, just as is done in the application of autonomics in the 3GPP radio access network (RAN) using the Hybrid SON Model that combines Centralized Self Organization Network (C-SON) and the Distributed SON (D-SON) in the NodeB (or eNodeB).The ETSI White Paper No.16 [3] and a GANA instantiation onto the 3GPP Backhaul and Core Network architectures (ETSI TR 103 404 [i.4]) discuss this subject to some depth.

What should be noted is that the Network-Level DEs shall embed cognition functions (Observing, Comparing, Learning, Analysing functions) and a knowledge database (KB) or simply Knowledge Repository (KR). Cognitions functions may be added in lower level-DEs if cognition properties are required for DEs in GANA Node. However, cognition requires a large amount of process and storage not always available on Network Elements (NEs) and they are recommended for slow control loop as in the Network level-DE. KB may be used by DEs to store facts about their associated MEs and an inference engine that can reason about those facts and use rules to deduce new facts or highlight inconsistencies. An ontology model may be developed for each DEs and should be linked to a composite ontology which is used to describe the network state. Those ontologies may be linked with others ontology domains which may infer with the autonomic network such as the GANA profile used to govern the autonomic network.

4.4.6 Implementation Guide for GANA DEs

The ETSI White Paper No.16 [3] provide insights on possible approaches to implementing the GANA and GANA DEs.

NOTE 1: There is on-going work in ETSI on elaborations of the Implementation Guide for GANA and the resultant work is expected to be published in ETSI TR 103 195-3 [i.71]. Figure 4 illustrate the point that some DEs (GANA Level 2, Level-3, and Level-4 DEs as "overlay software" inserted to run on top of the resources layer) can be implemented as a singly merged run-time entity (e.g. a process) or that individual DEs can be implemented to run as standalone processes (or threads for example) at run-time.

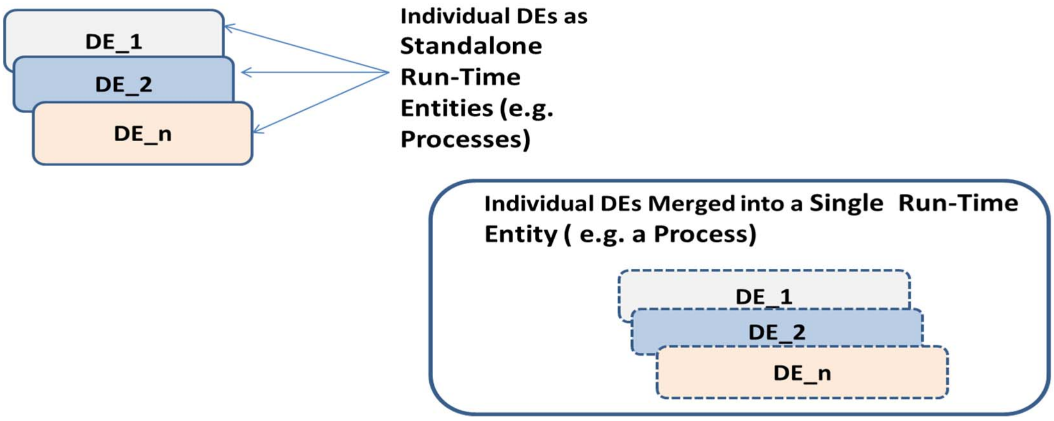


Figure 4: Illustration of possible approach to implementing GANA Levels 2, 3 and 4 DEs at run-time

NOTE 2: GANA Level-2, Level-3, and Level-4 DEs can be viewed as "overlay software" modules that can be easily inserted (loaded) into NEs without needing to make changes to Managed Entities (MEs) such as protocols and stacks the DEs should adaptively manage and control. As such the three GANA Levels of Hierarchical Control-Loops (GANA levels 2-4) demonstrate how Autonomics can be introduced in architectures without impact to stacks/resources layer of an NE. Such DEs may easily be "loaded" (or even replaced after) into an NE (physical or virtual), and this implies the enrichment of the notion of "Software-Driven" or "Software-Empowered" Networks, especially considering the growing trend on hardware/software disaggregation (splitting). As discussed in ETSI GS AFI 002 [2] (particularly in clauses such as clause 9.6.2 of ETSI GS AFI 002 [2]), run-time interpretable (e.g. by the aid of an interpreter of some sort) and executable software code, or run-time executable behavioural models should be considered in designing and implementing (and deploying) GANA Level-2 DEs and GANA Level-3 DEs in relation to the notion of loadability of DEs into an NE to enable it to be autonomic-thanks to the DEs.

4.5 GANA's Network Governance Interface

On the Network Governance Interface, the human operator/network-administrator can manage the operation of the whole autonomic network by authoring, validating and submitting High-Level Network Objectives and some Configuration-Data and Models, all encapsulated together in the form of a data structure called the GANA Network Profile (NETPROF) - a concept defined in ETSI GS AFI 002 [2].

NOTE: The concept of a Network Profile defined in ETSI GS AFI 002 [2] and also encapsulates the concept of Node Profile, is there to simply provide guidance on how to pack configuration data that could be generated by Automated Management Tool Chains, and so the structure of a Network Profile can be customized and adapted to fit the needs of a particular automated management environment for networks and services.

The Governance MBTS (G-MBTS) function can be used to translate the proprietary operator network profile into the NETPROF together with configuration data for DEs and NEs that can be generated by a collection of Automated Management Tool Chains at the disposal of the human network operator.

The human operator first, edits various types of items described later in more detail in the clause on GANA Operations Procedures for the Human Network Operator and the Annex on Operations Guide for GANA Empowered AMC and Autonomic Networks, such as High Level Business Goals/Objectives for the Network, Service Profiles andSLAs, Network Service Definitions/Graphs and Policies, Application Intents (see the definition of Intent Based Networking (IBN) in [i.64] for insights on this paradigm that is yet to be fully developed and mature) and Profiles/Policies and other types of configuration data, then validates them and resolves any potential conflicts in the configuration data and policies using appropriate tools before the input is supplied to Knowledge Plane and the network infrastructure elements (NEs). Second, he encapsulates them within a NETPROF as input to the DEs and their control-loops to govern the whole network goals. The autonomic network exposes views and report of the network behavior to be used by the Human operator in this process.

The GANA NETPROF is used to govern the autonomic network. NETPROF contains goals (objectives) and policies for all DEs on the different levels of the GANA hierarchy. The Node Configuration Files (NODECONF) are generated by Automated Management Tool Chains as skeletons and augmented and completed by Network-Level-DEs, these can influence the behavior of the underlying DEs in the GANA Nodes (NEs) by setting or modifying DE policies inside of the Node Profiles and some config data in Node Configurations files or models. A NODECONF (Node Configuration Files) may contain a part that contains the NODEPROF needed by a GANA Node and vendor specific configuration options. The NODECONF may be used to govern any vendors' nodes types with the same roles or may be used to govern a specific GANA NODE. An MBTS may be used to translate the NODECONF to proprietary vendor's node configuration and communications mechanisms. Besides, NETPROF may be applied by Knowledge Plane DEs to coordinate and orchestrate DEs decisions by adaptive policy control when multiple DEs are involved in a whole decision making process. The NETPROF should also be contain reporting information that should be generated by the Autonomics Software (the DEs) as aggregate reports of how the autonomics DEs are striving to manage and control the network and automatically fixing detected problems as well as recommendations the DEs produce for the human operator to address. The reports may be embedded in a Network Profile by inserting the report information or may be produced as standalone files that can be consumed by the human operator and inspected. The human operator can inspect the network behaviors through the aggregate generated by the Knowledge Plane DEs and containing reports by DEs in NEs as may be necessary. The reports may be used by the Human operator to control the autonomic network. More details are provided in clauses in the present document that cover the following aspects: GANA Operations Procedures for the Human Network Operator, GANA Policy Based Governance, and the Annex on Operations Guide for GANA Empowered AMC and Autonomic Networks.

4.6 Reference Points (Rfps) in GANA

In addition to defining the abovementioned GANA FBs, the GANA Reference Model also specifies Reference Points (Rfps) that shall be used between FBs of the GANA reference model. These Rfps between DEs may be either Horizontal Reference Points (HRPs) (i.e. between DEs belonging to same levels) or Vertical Reference Points (VRP) (i.e. between DEs belonging to a different level) depicted in Figure 2. HRPs "in-network" DE-to-DE communication and interaction of associated distributed Control-Loops may simply be viewed as an enhancement to the traditional distributed control plane with distributed control that enable NEs to (re)-negotiate some configuration aspects and to adaptively control the behaviour of MEs i.e. to collaboratively self-optimized end to end service delivered to end users. Clause 6 in the present document defines the Rfps between GANA Functional Blocks (FBs) as well as their interactions with other types of Functional Blocks in the network and its management and control architectures, by adopting the Reference Points defined in clause 13 of ETSI GS AFI 002 [2] and extending them with additional Reference Points (Rfps) necessary.

In particular, in order to introduce autonomicity in any network architecture, an instantiation needs to be carried out concerning the FBs and Rfps prescribed by the GANA Reference Model onto target architecture, e.g. the wireless mesh network architecture [i.6], the 3GPP network architecture [i.4] or the future internet architecture [i.2] and [i.7]. As illustrated in the example GANA instantiations ([i.4] and [i.6]), an instantiation of GANA onto target architecture includes the instantiation of the required Rfps and shall also provide implementation details for the instantiated Rfps-meaning the complete details on the communication means (methods) and data models concerning the characteristic information communicated by the Functional Blocks (FBs) bound by the specific Rfp under consideration, as each instantiation needs to be defined according to the requirements and uses cases for autonomics to be realized in the target architecture [1], [3] and [i.71]. Implementers of GANA autonomics onto a target network architecture and its associated management and control architecture shall consider the insights provided by the requirements and use cases described in [1] in order to identify and consider implementing the requirements and uses cases captured in [1] that apply to the target network scenario under consideration, and to understand how GANA is meant to complement traditional network management systems and its impact in the long term of network and services management. Those aspects of GANA and impacts on traditional management systems described in [1] are complemented by the further elaborations on this subject that are provided in the present document.

NOTE: There is ongoing work in ETSI on GANA instantiation onto the BBF architecture scenarios (ETSI TR 103 473 [i.28]).

4.7 A Consolidated Characterization of the GANA Knowledge Plane, and Cognitive Algorithms for Artificial Intelligence (AI) in Autonomic Functions (DEs)

1) The Knowledge Plane (KP) is an integral part of Management and Control Systems that provides for the space to implement complex network analytics functions performed by Decision-making-Elements (DEs) that run as software in the Knowledge Plane and drive self-* operations such as self-adaptation, self-optimization objectives for the network and services by programmatically (re)-configuring Managed Entities (MEs) in the network infrastructure through various means possible: e.g. through the NorthBound Interfaces available at the OSS, Service Orchestrator, Domain Orchestrator, SDN controller, EMS/NMS, NFV Orchestrator, etc. In such interactions the Knowledge Plane DEs may employ the services of an MBTS (Model-Based Translation) set of software libraries as discussed in ETSI White Paper No.16 [3], as the MBTS enables management-protocol agnostic, control-protocol agnostic, and vendor-agnostic management and control of physical or virtual Network Elements (NEs) by the GANA Knowledge Plane. An *Intent-based language* (Intent Based Networking paradigm-see the definition of Intent Based Networking (IBN) in [i.64] for insights on this paradigm that is yet to be fully developed and mature) for programming the network through the SDN controller that may provide such a northbound intent-based interface can also be applied by the GANA Knowledge Plane DEs in (re)-programming the network.

2) Various types of GANA Knowledge Plane's Decision-Elements (DEs) can be designed to perform autonomic management and control operations (the self-* operations) on various types of Managed Entities (MEs) in the network infrastructure' network elements (virtualised (VNFs) or physical (PNFs)). DEs are representative of cognitive management and control "domains" of reasoning regarding specific management and control aspects. Example DEs that can be designed in the Knowledge Plane are:

   - Fault-Management-DE.
   - Resilience_&_Survivability-DE.
   - Auto-Configuration and auto-Discovery-DE.
   - QoS_&_QoE_&_Performance-Management-DE.
   - Security-Management-DE.
   - Monitoring-DE.
   - Routing-Management-DE.
   - Etc.
   - A DE *monitors* Managed Entities (MEs) assigned to it by design, uses the monitoring data and any other input data from other data/information or policy sources in the environment to *analyse and compare the state of the MEs against the desired state* that is adaptively computed from certain objectives meant to be enforced by the DE (which may change any time based on e.g. context and policy changes, network conditions, manifestations of faults in the network, etc.), and then *creates a plan of actions or strategies to dynamically change the state and operations of the MEs*, and then *executes the actions/plans* to effect changes on the MEs. Such DE operations are performed reactively and/or proactively to meet desired objectives regarding the state or behaviours of MEs. Cognitive algorithms of a DE e.g. Machine Learning, Deep Learning, Computational Intelligence and other types of AI (Artificial Intelligence) algorithms, etc., drive the operations of the DE. ETSI GS AFI 002 [2] provides insights as to which levels in GANA should the DEs at that level need to have a higher degree of cognitive properties (cognition), as DEs at the Level-1 need to implement fast control-loops possibly with little or zero cognitive properties (cognition); moving higher up the GANA levels-at DEs at Level-2, control-loops are fast in contrast to control-loops at Level-4 (outside of a GANA Node/NE) but slower than those at Level-1, but a certain degree of cognitive properties can be introduced in the Level-2 DEs; moving higher up the GANA levels-at DEs at Level-3, control-loops are fast in contrast to control-loops at Level-4 (outside of a GANA Node/NE) but slower than those at Level-2 to some extent, but a much higher degree of cognitive properties than GANA Level-2 and Level-1 can be introduced in the Level-3 DEs; moving higher up the GANA levels-at DEs at Level-4, control-loops become slower in contrast to control-loops at GANA Levels in the GANA Node/NE level, but the highest degree of cognitive properties than GANA Level-3, Level-2 and Level-1 collectively, can be introduced in the Level-4 DEs.

So, in general, control-loops within the NE are fast control-loops (in terms of timescale of reactions to changes) while those outside, at the network-level (GANA Knowledge Plane level), are slower than those at NE level but with much very high degree of cognitive properties and wider (network-wide) views as scope on which the corresponding Network Level DEs operate on. Therefore, regarding cognition, Network Level DEs "shall" exhibit cognitive behaviour; Node-Level DEs "should" exhibit cognitive behaviour; and Function-Level DEs "may" exhibit cognitive behaviour. The degree of cognition and complexity of DE cognitive algorithms decreases from Network-Level through to the lower GANA Levels.

ETSI GS AFI 002 [2] provides more insights on such design principles for DEs. GANA, as described in ETSI White Paper No.16 [3] and ETSI GS AFI 002 [2] and in the present document, provides guidelines on how to design DEs such that they coordinate their operations to avoid conflicts.

3) The GANA Knowledge Plane may perform the functions of traditional management and control systems usually performed by traditional OSS, NMS/EMS if such a scenario is commercially viable such that no such traditional management systems are required anymore, otherwise the Knowledge Plane should interwork with such traditional systems in driving the re-configuration (e.g. to realize Self-Optimization objective or Self-Healing Objective) of the network and services as may be deduced and deemed necessary by DE algorithms during the operation of the network. Another implementation option could involve DEs implemented to run as loadable modules of an OSS, otherwise the Knowledge Plane can integrate with the OSS via an OSS Northbound Interface.

4) The GANA Knowledge Plane should fulfil the combined role of Network Analytics Driven Service Orchestration and Network Analytics Driven Closed-Loop (Autonomic) Service Assurance:

- Network Analytics Driven Service Orchestration should be performed by the Knowledge Plane DEs in response to network or resource capacity demands and resilience targets/objectives.

- Network Analytics Driven Closed-Loop (Autonomic) Service Assurance should be performed by the Knowledge Plane DEs with the target of improving customer experience. Autonomic (Closed-Loop) Service Assurance involves the Knowledge Plane as an Analytics Platform equipped with engines (DEs) that collects and analyses data from various data sources such as traditional Service Assurance Platforms (e.g. Performance management systems), network service functions/nodes, SDN Controllers, etc., and detect any service degradations and SLA violations. The Analytics Platform then closes the loop by communicating monitor results to Orchestrators and triggering remediation and corrective operations via a combination of Service Orchestrators, SDN Controllers, and Service Functions/Nodes such as CPE (Customer Premises Equipment), Access Node, BNG in Broadband Forum (BBF) architectures, and other types of service function nodes of other types of architectures. The Knowledge Plane DEs should be able to communicate to a Service Orchestrator Results obtained from Monitoring a Service such as SLA violations and generate Recommendations (actionable insights) on how the problems can be solved (humans could make use of the generated Recommendations, e.g. making use of the Recommendations to perform the actions if the Knowledge Plane DEs are configured to operate in an "Open-Loop" Mode). At the same time in a "Closed-Loop" mode, the DEs should go further on their own accord to trigger operations on the Service Orchestrators (which include orchestrator types like the NFV Orchestrator) in a "Closed-Loop" (autonomic) service assurance goal based on what the DEs determine to trigger on an orchestrator or any other management and control system such as an SDN controller, so as to realize Self-Healing of the Service(s)-thanks to autonomics of the Knowledge Plane operations. While Service Assurance should now evolve towards "Closed-Loop" (Autonomic) Service Assurance, rather that the Service Assurance Function computing Recommendations as actionable insights and operate in a an open loop as discussed in Heavy Reading and SaS White Paper [i.40] and in [i.9], the GANA Knowledge Plane is meant to be an implementation of a Service Assurance Function that is *autonomic* in its operation, acting in a Closed-Loop fashion that drives Self-* behaviours (performed on the Managed Entities (MEs) of the network) such as *Self-Healing, Self-Organizing, Self-Optimizing, Self-Protection, Self-Repair, etc., and exhibiting Self-Awareness.*

  - Offer insights that help the Operator to create and launch new types of services that could be offered to customers based on the Recommendations that the Analytics performed by DEs in the Knowledge Plane can produce with respect to the types of services (e.g. connectivity services) that can be provisioned over the capacity deduced to be available without compromising QoE (Quality of Experience) of end users currently served by the network. The Recommendations should be based on converged and aggregate analytics that are collectively correlated by the various DEs in the Knowledge Plane over historical usage trends of the E2E network capacity and other information such as performance trending data, etc.

5) The GANA Knowledge Plane (KP) is meant to provide the realm in which Decision-making-Elements (DEs), as software, can be designed and implemented to perform and realize the following functions:

- Network Analytics should be performed in the Knowledge Plane (KP) using various types of algorithms for reactive and predictive analytics, including Machine Learning, Deep Learning, Computational Intelligence and others types of cognitive algorithms such as Artificial Intelligence (AI) algorithms, in augmenting any network analytics performed at systems and platforms such as Data Collectors that feed generated knowledge into the Knowledge Plane. Some cognitive algorithms may also run on the ONIX Information Servers for further correlation of information stored in the ONIX, to maintain an updated view on knowledge pertaining to current state of the network and also knowledge pertaining to historical network state and decisions performed by DEs as historical traces. The analytics performed by KP DEs drive maintenance operations and as well as even marketing campaigns (such value is also discussed in ASSIA SDAN White Paper [i.39]). Sources such as Knowledge-Defined Networking [i.35] and KnowNet [i.41], and many other sources in literature provide insights on data-sources for network analytics that can be performed on analytics platforms such as Data Collectors or at some management systems to generate some knowledge that can be supplied to the Knowledge Plane. The Knowledge Plane DEs can augment the knowledge by consolidating such knowledge and further performing aggregate analytics of information and knowledge from various input sources on a more global level. Cognitive algorithms discussed in Knowledge-Defined Networking [i.35] and many other sources on Knowledge Plane related topics, including results from research projects on autonomics and cognitive network management, and also some real implementations already achieved to some degree in the industry in the areas of Service Assurance and Big-Data Analytics Driven network management can also be applied in implementing the GANA Knowledge Plane and its interfaces described in ETSI White Paper No.16 [3].

- Complex Event Processing (CEP) techniques are employed by DEs in the Knowledge Plane, as discussed in ETSI White Paper No.16 [3].

- Data collected in the network by probes in the NEs or probes specially instrumented to collect data such as traffic captured through means such as link tapping, should be made available to the Knowledge Plane DEs for the purpose of enabling their algorithms to perform optimization and diagnostics.

6) The ONIX Information Servers and services may be employed for realizing a **Real-Time Inventory**.

7) The GANA KP should be aware and make use of the following items:

- SLAs, for use in determining SLA Violations and autonomically programming the network in order to guarantee and sustain acceptable levels of QoS and QoE for user traffic.

- Application policy/Profile Profiles.

- Network Service Designs/graphs.

- Other characteristics of the GANA Knowledge Plane: KP Multi-Tenancy may be required for some networking scenarios that require multi-tenancy in the management and control software responsible for network portions, network slices and administrative domains in the End-to-End network architecture. For example, a GANA instantiation onto the BBF architectures [i.28] discusses the impact of some SDAN (Software-Defined Access Network) Scenarios on the nature, multiplicity and responsibilities desired of the GANA Knowledge Plane (KP) instances that should be operated in the various scenarios, and also the relationships of the specific Knowledge Plane instances with the Network Elements (NEs) that should be under the control of a specific Knowledge Plane instance. The work in [i.28] considers scenarios taken from [i.39], [i.42] and [i.5]. In implementing a KP, either a Standalone Logically-Centralized KP can be implemented, especially for fixed/mobile/wired/wireless networks, while an Overlay KP may be deployed in a distributed manner inside network nodes like in for some types of mesh and ad-hoc networks as discussed in ETSI TR 103 495 [i.6] and [i.29].

4.8 Complex Event Processing (CEP), Context-Awareness, Data Analytics and Cognition in the GANA Knowledge Plane

4.8.1 Complex Event Processing (CEP) as part of a Data Analytics Capability of DE or of a Data Analytics Module employed by multiple DEs

Regarding the use of *Complex Event Processing (CEP)* paradigm [i.77] and its role in relation to the interworking of Fast Control-Loops and Slow control loops: There is a need for coupling CEP capabilities that should be operating in the GANA Knowledge Plane (KP) with policies applied by the KP DEs. CEP in the KP could be realized in two ways: CEP happens in each DE in the GANA Knowledge Plane or a single instance of a Data Analytics Module with CEP capabilities that feeds all the DEs with events that the DEs shall act upon could also be used. To facilitate for the [i.28] interworking of fast control loops and slow control loops in CEP, autonomic network nodes should push up into the GANA knowledge Plane DEs aggregate events concerning local DEs' actions and other aggregate events/views inferred from monitoring data. GANA nodes should also relay synchronization requests from lower level DEs that require that a DE on the network level coordinates the approval of a tentative action the lower level DE intends to execute.

4.8.2 Context-Awareness, Context Aware Engine (CAE) as part of a Data Analytics Capability of DE or of a Data Analytics Module employed by multiple DEs

A Context Aware Engine (CAE) is as an enabler of the Decision making Element (DEs) responsible for generating context from an event in GANA Nodes or in the KP or in the ONIX. A CAE feeds the DEs with context that DE(s) should act upon, and also feeds the Cognition Module (CM) of cognitive DEs if a CM is not implemented as part and parcel of a general Data Analytics Module that exhibits CEP, CAE and CM features/capabilities. CAE may use *Complex Event Processing (CEP)* paradigm to process an event, therefore CEP module and CAE module should be implemented as one and the same module that is an integral part of a Data Analytics Module. It should be feasible to implement the CM module of a DE into a single Data Analytics module combining CEP, CAE and CM features/capabilities.

GANA nodes should push up into the CAE module(s) employed by the GANA KP DEs aggregate events concerning GANA node-local DEs' actions and other aggregate events/views inferred from the GANA node's MEs raw state information. GANA nodes should also relay synchronization requests from lower level DEs that require context from CAE module(s) employed by the GANA KP DEs to be considered by the KP DEs (on the network level) in coordinating the approval of a tentative action on the lower level DE in the GANA Node that the lower level DE intends to execute.

CAE (as a part of a Data Analytics Module) in the GANA KP could be implemented in two ways: a CAE capability can be implemented into each DE in the GANA KP or a single instance of a Data Analytics Module with CAE capabilities that feeds all the DEs in the GANA KPs within an autonomic network environment with context information. Like for standalone single shared Data Analytics Module with CEP capabilities, a single shared instance of a Data Analytics Module with CAE capabilities could possibly be implemented as part of the MBTS (particularly the AMC-MBTS).

CAE gathers *raw information* based on information publish/subscribe mechanisms supported by multiple information sources, which include GANA Nodes, the Overlay Network Information eXchange (ONIX) system(s), entities such as Data Collectors used to gather network and service monitoring data, and the MBTS, which translates raw data/information received from GANA nodes into knowledge that is streamed to KP DEs and also registered (stored) into the ONIX. The translation services of the MBTS imply that the MBTS shall be designed to cater for various methods for Information Registration, Acquisition and Translation between the KP Domain (information and (meta)-data models that support GANA specifics) and the network elements (NEs) domain (management and control related information and data models). The MBTS may also export to raw information to a CAE if it is not implemented as part of the MBTS. The raw information gathered by the CAE is then converted to *facts* and then ultimately converted to meaningful knowledge (*context*) by the *CAE*. The created context is then consumed by the DEs which subscribe to it in order to act upon it. While the context is exposed to KP DEs, these can also be a source of information which is utilized by CAE when generating context and it could be exposed to governance applications (OSS/BSS) to supervise the Context generation.

The CAE (if implemented within a DE) should also take into account the peering possibility with CAEs in peer DEs within the same KP or with CAEs in other KP. Through ONIX instances of different KPs context between DEs may be shared. A CAE can facilitate the interworking of Fast Control-Loops (implemented by GANA Nodes DEs) and the slow control loops (implemented by KP DEs) within GANA KP, or the interworking of different autonomic network environments (peer KPs) as described in the present document (in the clause(s) on Federated GANA domains of management and control).

There are two key tradeoffs which need to be considered in CAE design. First is the temporal aspect that dictates the time scale at which the CAE can influence the decision of the DEs. For instance, handover execution by the mobility management network function shall be performed at a very short time scale (in milliseconds) regardless of any other situation of the network (e.g. Network traffic steering optimization). Meanwhile the determination of best options for mobility anchoring point selection may be performed over longer periods of time (e.g. in seconds or minutes) according to the situation of the network. The CAE can handle both these time scaling requirements but the focus here is to support the operation of DEs in a longer time scale as the fast control loop should be handled locally by the GANA nodes themselves through Node Main level DEs and Function Level DEs.

The second aspect is related to the performance of the CAE in generating context. Having CAE in the GANA KP with a global view of the system (network-wide perspective) can allow for global optimizations using the KP's slow control loops realized by the KP DEs. This relates to the validity or expiry of the monitored raw information and facts which constitute the context. There is a high cost on transferring/translating/registering/acquiring the raw information from the various sources to GANA KP CAEs and analysing/reasoning/learning on all the facts generated in the network. To this end, a distributed implementation of CAE could be pursued to enable high performance and timely generation of context. In such a scenario the CAE instances can run in specific regions to locally solve the identification of context and via collaboration amongst each other exchange information to guarantee also the possibility of generating context information sets that are not only regionalized but are covering multiple regions.

The CAE should continuously be evolved according to new sources of information, facts, rules, algorithms, required to generate context to event consumers. There is a need for coupling the management of context that should be generated by a CAE in the GANA KP with GANA Network Profile.

CAE shall be configured with a "configuration profile" (which should be embedded as part of the Global GANA Network Profile or the DE Configuration Profile if the CAE is embedded in a DE), which set facts and rules used to generate context and store it in the CAE "knowledge base". The "profile" may also define the data analytics algorithms that shall be used to generate context (e.g. machine learning algorithms) e.g. the prediction model that may be used to generate context defined by a cognition module of a DE. The "profile" shall also define the kinds of raw information the CAE should subscribe to receive from information sources to generate context or to publish the context to entities that require the context.

The CAE shall expose KPIs pertaining to its own performance concerning its context processing e.g. latency to generate context, precision, deviation of context exposed. It shall also validate that the facts and rules in its "configuration profile" are not conflicting with existing ones and shall update the inferred facts learnt by the CAE inference engine. The "Profile" may use a language such as RDF [i.57] or **json-ld** [i.58] to manage the CAE. The MBTS may use RIF to translate rules from a GANA-agnostic language to a language (information and (meta)-data model(s)) that incorporates GANA concepts and abstractions and is understood by the KP DEs (KP Domain).

NOTE: There are some sources of resourceful concepts and insights in literature that may be considered in implementing. For example, there are some Fi-WARE concepts and insights from [i.77] that can be exploited in the design and implementation of the CEP and CAE capabilities. From the same Fi-WARE framework, there are also concepts and insights that could be applied in implementing certain aspects/capabilities of MBTS, ONIX, Information/Knowledge Repositories as well as certain algorithms that may be employed by GANA DEs (particularly KP DEs).

4.9 Interworking the GANA Knowledge Plane with Traditional Network Management Systems

In the context of a hybrid network management the GANA Knowledge Plane interworks with the legacy management plane (Operation Support System (OSS), Business Support System (BSS), Network Manager (NM) and Equipment Manager (EM)) is depicted in Figure 5. The way the GANA Knowledge Plane interworks with various types of management systems is described in ETSI White Paper No.16 [3] and [i.24] and in **Annex B** on "Requirements for Protocols and APIs for Enabling GANA based Autonomics, Cognitive Networking and Self-Management of Networks and Services in Evolving and Future Networks", as well as in Annex C that provides insights on Operations Guide for GANA Empowered AMC and Autonomic Networks.

NOTE: It can be envisaged that in the long term the Knowledge Plane may take over most of the management and control operations performed by legacy OSSs, EMs and NMs.

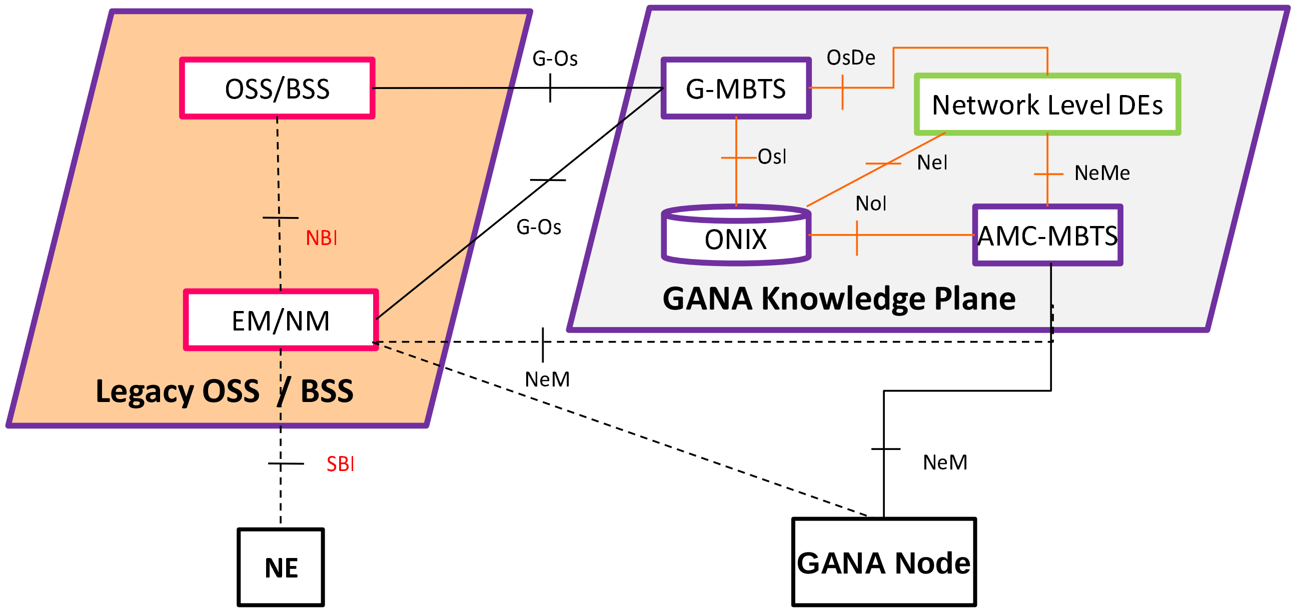


Figure 5: Hybrid network management and control using a combination of legacy EM/NM and the Knowledge Plane (may take over in the long term)

GANA's KP should complement the existing network management plane by the following facilitations (enablers) for adaptive management and control of networks and services:

a) Facilitating information exchange, through publish/subscribe services, an advanced self-awareness of the elements plugged into the network, their capabilities, network resources, configuration-data/profiles/policies, some pointers to information and resources, etc.

b) Establishing the type of autonomic functions (i.e. DEs, their associated control loops and their assignment to specific MEs, as well as parameters they manage and adaptively control) that should be instantiated onto particular Network Elements.

EM/NM functions may also govern some network level DEs and use the same interface than the OSS through the G-Os Rfps. The EM/NM may also be governed by network level DEs through the NeM Rfps. EM/NM may be used by Knowledge Plane DEs in indirectly manage NEs (GANA Nodes) in some cases, e.g. when there is no MBTS employed by the Knowledge Plane DEs or when for some reasons Knowledge Plane DEs have been implemented to manage some NEs through an EM/NM. MBTS functions may be used to map/translate legacy meta-data model associated with legacy protocols to the meta-data model that is specially designed to be employed by the GANA Knowledge Plane DEs as a management and control language that is agnostic to vendor specific management protocols and technology specific management protocols that can be used to manage NEs. This means a legacy network management and control protocol and the data model associated with information/data conveyed by the protocol may translated into the GANA Knowledge Plane DEs domain language and data model through (AMC-)MBTS and vice-versa.

4.10 Architecture for Federated GANA domains of management and control

Regarding the end-to-end transport architecture between different network domains (technical or administrative) the federation management architecture. Figure 6 illustrates the articulation between the components of each domain and the Rfps required to exchange information. Other aspects of federation in management and control of networks are covered in ETSI GS AFI 002 [2] and [i.22].

NOTE 1: In the federation of Autonomic Management and Control (AMC) of Networks and Services, "domain" means either "Administrative Domain", or "Technical or Technology" Domain as described in ETSI GS AFI 002 [2] in the clause on Federation in GANA, and also in ETSI TS 103 194 [1].

The federation architecture is used to identify the relationship between distributed DEs through various network domains which implemented different network standards. Specifying those Rfps, mean identifying operations and primitives that need to be exchanged between the different DEs of the different domains. It requires collaboration between different SDOs to identify the knowledge that need to be shared [i.2]. A federation Broker FB may be used for the federation between different domains. These FBs depicted as an F-MBTS in Figure 6 may be used to translate operations and primitives from one domain to other domains (e.g. data model, language rules, policies language, KB inference). Those Rfps should ensure secure communication channel to exchange information between domains and convey only the kind of knowledge needed to be exchanged between domains [i.4].

The GANA Federation Architecture defines that MEs in different domain should be managed by only DEs in their own domains; it means that only DE and ONIX may exchange information required for the global federation.

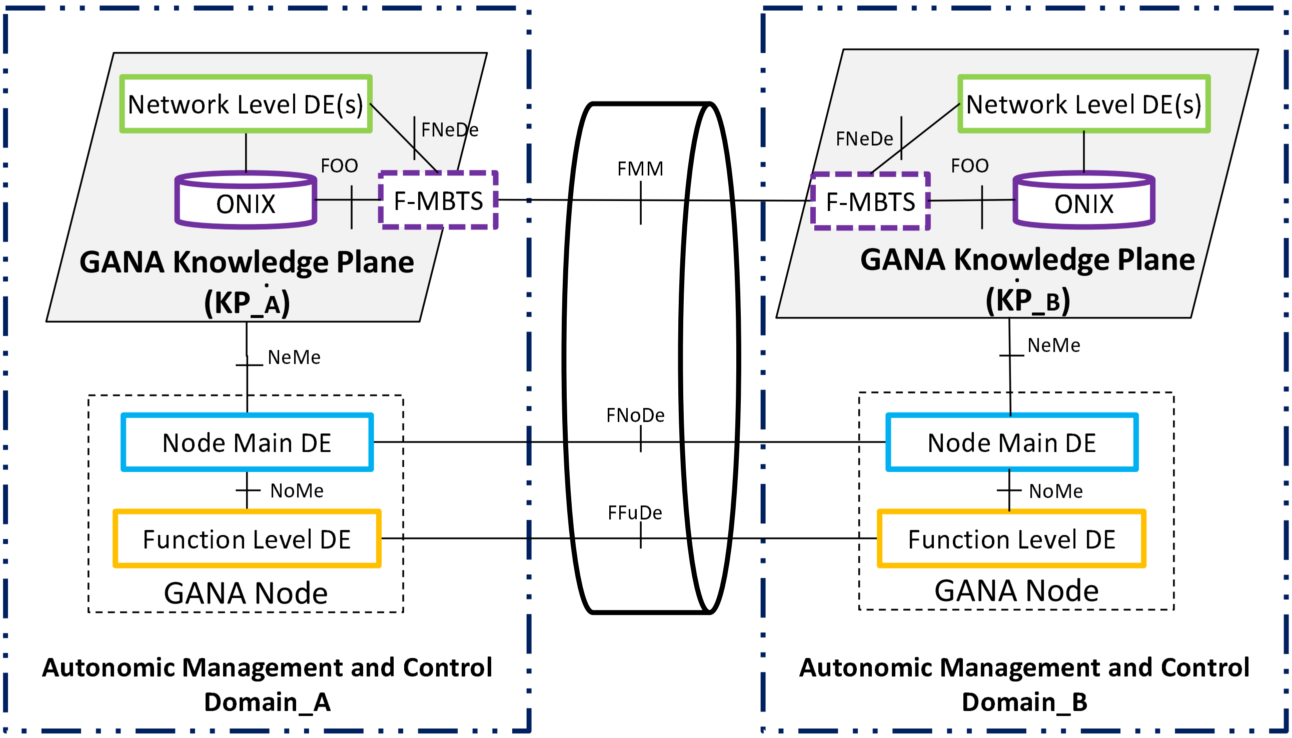


Figure 6: Federated GANA domains of management and control architectures for inter-domain autonomics

NOTE 2: As shown in Figure 6 on "*Federated GANA domains of management and control architectures for inter-domain autonomics*", the F-MBTS (as it plays the role of a "translation/mediation service" between federated domains, may be *Optional*, because there might be federation needs and operations that do not require such a "translation/mediation service" to be put in place.

NOTE 3: A federation Broker (role of "translation/mediation service") may be used for the federation between different domains. Such a federation broker could be one of the ways to implement the reference points FMM, FNoDe, FFuDe depicted in Figure 6, but the reference points could also be implemented using other means such as some communication protocols for direct communication between the Functional Blocks (FBs) on either side of a particular reference point**.** Such a broker should be used to translate Knowledge Plane operations and associated primitives (communication methods) communicated by KP entities from one domain to the other peer domain. The translation should include translation of differing data models, control language semantic rules used in the federation related messages exchanged by the peer domains' entities, differing policy languages for policy control employed in the different domains. The broker should also perform inference of Knowledge shared between the peer domains and translate the knowledge into a form that is usable by the target domain.

NOTE 4: Autonomic Management and Control (AMC) federation across different domains-with possibility to employ a technical broker (F-MBTS) that provides for a "translation/mediation service" between the domains may be driven by the need to federate end-to-end networking and network service oriented capabilities for the benefit network service consumers (e.g. end-user customers of a network operator), but the federation may also be driven by a business driver or model brought about by a Business-to-Business (B2B) service delivery model that brings business benefits to the stakeholders/entities involved in the federation.

4.11 GANA Instantiation Guide and Examples/Cases of GANA Instantiations

The first step in the instantiation of GANA onto an implementation-oriented reference network architecture and its management and control architecture is to define the DEs needed for each function of NE that may be self-managed and their associated types of MEs for specific uses cases. This step should instantiate the GANA FBs and the Reference Points (Rfps) between them that are defined in GANA and are needed in target implementation oriented network architecture, and the process should also identify the need for implementing Rfps that may be required between different domains for enabling end to end self-management of networks and services. The result of the instantiation is an autonomics (autonomicity)-enabled reference network architecture and its management and control architecture, with definitions of the autonomic functions (DEs) that should be implemented in NEs and in the Knowledge Plane, as well as the mappings of DEs to MEs they should autonomically orchestrate (install, launch, etc.) and/or manage and control by applying algorithms that dynamically orchestrate or configure MEs and their configurable and controllable parameter in response to various types of changes such as workload changes, policy changes, manifestations of faults/errors/failures. As described in ETSI White Paper No.16 [3] and ETSI GS AFI 002 [2], vendors and innovators of the DEs shall then use the result of the GANA instantiations onto target implementation oriented architectures to innovate DE algorithms and provide the autonomics required in the architectures. DE algorithms (are not meant to be standardized) provide for DE vendor differentiation. Examples of such instantiations are indicated below. Prioritization of use cases and requirements for autonomics in target architectures onto which GANA has been instantiated may be done by the standardization groups (e.g. ETSI in collaboration with 3GPP, ETSI in collaboration with BBF, ITU, TMF, etc.) producing the GANA instantiation and they may indicate to the industry the next iterative step that enhances the GANA instantiation with newly introduced autonomics uses cases and requirements that should be addressed by DEs and the other instantiated GANA functional blocks. For more details on the process of GANA instantiation, the following documents: ETSI White Paper No.16 [3] and [i.15].

NOTE 1: There is on-going work in ETSI on elaborations of the Implementation Guide for GANA and the resultant work is expected to be published in ETSI TR 103 195-3 [i.71] that complementarily is expected to include the following aspects and other aspects as well:

- The answer to the question of incremental implementation of DEs and the implication of a network that implements only certain DEs and levels in GANA.
- Identifying the policy model and language rules that should be used to govern the autonomic network-the governance of DEs in the implemented in the Knowledge Plane and those implemented at NE level.
- A possible approach that can be pursued in implementing the ONIX federated system of Information Servers.

The following are example GANA instantiations produced in ETSI (other types of GANA instantiations onto other network architectures are expected to be developed and published by ETSI and other Standardization Groups/Fora):

- ETSI TR 103 404 [i.4]: "Autonomicity and Self-Management in the Backhaul and Core network parts of the 3GPP Architecture" (published in October 2016).
- ETSI TR 103 495 [i.6]: "Autonomicity and Self-Management in Wireless Ad-hoc/Mesh Networks" (published in February 2017).

NOTE 2: There is ongoing work in ETSI on GANA instantiation onto the BBF architecture scenarios (ETSI TR 103 473 [i.28]).

The GANA instantiations onto the implementation-oriented reference network and management and control architectures listed above can now be used by different players to implement proof of concept (PoCs) to showcase their autonomics solutions (based on standards) the industry is looking for. For this purpose ETSI launched a PoC framework that can be used by different payers to respond to the call for autonomics PoCs as described in the framework available at http://ntechwiki.etsi.org/. *An example of an autonomics PoC:*

- ***5G Network Slices Autonomic Management*** and ***E2E Orchestration, with Closed-Loop (Autonomic) Service Assurance for the Network Slices: Using IoT Use Cases.***

Such a PoC could be aimed at operationalizing the Unified Architecture for ETSI GANA, SDN, NFV, E2E Service Orchestration, Network Analytics and Big-Data Analytics for AMC, in order to provide guidance to the network operators on how all these complementary paradigms can work together in holistic architecture that integrates them together holistically, and also to obtain guidance on how to make use of E2E orchestration and the complementarity of GANA-based AMC, NFV, SDN, Data Analytics, CEM, legacy network management, and autonomic management and control of network and services.

5 The detailed description of the GANA Reference Model Structure, Core concepts and Principles

5.1 The Core Concepts, and Design and Operational Principles for GANA Functional Blocks

5.1.1 Overview

This clause provides more details on the GANA Core Concepts and Design and Operational Principles of GANA Functional Blocks that complement the descriptions provided in earlier clauses.

5.1.2 Structural Model of a Managed Entity (ME) at the GANA lowest layer (the resources layer), ME Interfaces, and Primitives/Operations for Enabling ME Programmability

5.1.2.1 Overview

Fundamental MEs are constituted by entities belonging at the bottom of the management hierarchy-at the fundamental resources layer where individual protocols or stacks, OSI layer 7 or TCP/IP application layer and other types of resources or managed mechanisms are hosted within a network element (NE) or in some device in the network in general. Some of such MEs may intrinsically embed a Decision Element (logic) in order to drive a control-loop designed into the ME. Such embedded DEs are called GANA Protocol Level DEs-the lowest level GANA DEs. The protocol level DE represents protocols, services, and other fundamental mechanisms; possibly already implemented in the network (e.g. the routing protocol OSPF (Open Shortest Path First) may be considered an example of an ME that embeds Protocol Level DE). ME is generically defined as Managed Entity, not Managed Element, in order to avoid the confusion that arises when one begins to think of an "element" as only meaning a physical or virtual network element (NE) as a "whole" ME may be managed in terms of its orchestration, configuration and re-configuration (re-programmed) through parameters settings.

Parameters of an ME shall be autonomically (re)-programmed (configured) by only one DE and if an algorithm involving the collaboration of multiple DEs require an ME to be (re)-programmed that would happen through the DE that owns the ME. More details on this subject are provided in clause 9.11 in ETSI GS AFI 002 [2].

Clause 9.11.2 of ETSI GS AFI 002 [2] presents a model of a Managed Entity (ME) at GANA's lowest layer. This clause adopts (inherits) the model and provides additional complementary perspectives to those provided in the corresponding clause in ETSI GS AFI 002 [2] for consideration and understanding.

Figure 7 shows a generic model of a Managed Entity (ME) at GANA's lowest level DE. What can be noted is that many types of components and tools integrated into systems today do not support the notion of a "Management Interface" as depicted on the model. To be manageable an ME shall support the notion of a "Management Interface".

NOTE 1: A partial fulfilment of the model may be traced in implementation and management of individual networking protocols.

NOTE 2: Not all managed entities at the resources layer of a network node (NE) may be conforming to the generic model. Additional models of what would be considered as lowest level MEs in the GANA managed networking resources layer may be derived from this generic model.

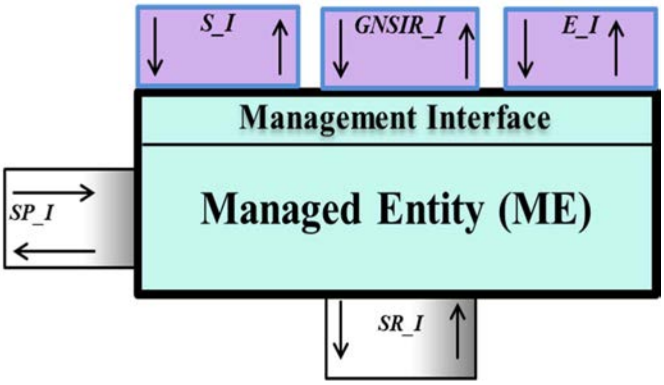


Figure 7: Generic model of a Managed Entity (ME) and its Interfaces

It also adopts the concept of a Managed Entity (ME) to mean a managed resource or an automated task in general, instead of a Managed Element-a term used in traditional network management terminology, in order to be more generic and to avoid the confusion that comes with the use of the term "Managed Element", which is normally associated with only meaning a physical Network Element (NE) and not some functional entity within a node/device such a protocol module or a component such as a monitoring component. The Management Interface of an ME and its sub-interfaces should support primitives as defined for an ME model in clause 9.11 of ETSI GS AFI 002 [2].

NOTE 3: The "management interface" of an ME such as a protocol like IP or Ethernet or other protocols is usually realized using a Management Information Base (MIB) implementation, and accessed by a "manager component" via a "management agent" that implements the corresponding MIB on the device/NE.

The MIB defines Managed Objects (MOs) that a DE designer may consider in designing a DE that dynamically manages the ME via manipulation of MOs of the ME. Clause 9.11 of ETSI GS AFI 002 [2] notes that many types of components, tools integrated into systems (networked devices/nodes) today, do not support the notion of a "Management Interface" as depicted on the model above, rendering such entities to exhibit limited "manageability property" or none of it. To be fully manageable an ME should support the notion of a "Management Interface" and to the extent possible support the primitives for the "sensory part" and "effector part" of its "management interface". The "Management Interfaces" of MEs at the fundamental resources layer in GANA may be collectively implemented using a single Unified API that could get standardized as described by the Reference Point ***Rfp_GANA-Level2-AccessToProtocolsAndMechanisms*** defined in ETSI GS AFI 002 [2] and further described in **Annex B** on "*Requirements for Protocols and APIs for Enabling GANA based Autonomics, Cognitive Networking and Self-Management of Networks and Services in Evolving and Future Networks*".

GANA level 2 and Level 3 DEs may use the same common ***Rfp_GANA-Level2-AccessToProtocolsAndMechanisms***or an API that implements it to dynamically manage and control the MEs at the resources layer of a GANA node.

NOTE 4: Annex C in the present document presents one of the ways in which this ***Rfp_GANA-Level2(&3)-AccessToProtocolsAndMechanisms*** can be implemented, that of using a unified API. Another may the reference point may be implemented is through the traditional approach of using means such as SNMP, by which the GANA Level 3 and Level-2 DEs locally manage and control their MEs through the local SNMP agent and the MIBs implemented for those MEs.

There are ideas that recently emerged on how to design future networking protocols and the management operations/primitives that they shall support in order to support dynamic protocol-stack composition and re-composition as driven by some goals and context of operation and environment in which a NE finds itself. This subject is covered in clause 9.11.3 of ETSI GS AFI 002 [2] where a Managed Entity (ME) that is could be considered as "evolved protocol" or a Future Protocol Model in GANA-at GANA's lowest layer is presented. Additional models of what would be considered as lowest level MEs in GANA are possible, for example other types of ME in a GANA node that are not of the nature (model) of network communication protocols that traditionally are integrated into protocol stacks. DEs at GANA level 3 (the Node-Main-DE) needs to be able to autonomically configure some MEs that are not of the nature (model) of network communication protocols and may be of the nature of physical or software resources. Such MEs could be a Network Interface Cards (NIC) drivers/firmware, some allocated memory or other kinds of Managed Entities (MEs) that could be exposed by an Operating System (OS) through some software libraries that could be leveraged by DEs to dynamically manage and control such types of MEs that may not even be of the nature of networking communication protocols.

The clauses that follow describe Interfaces of an ME Model at GANA's lowest layer (networking resources in an NE).

The generic model of a Managed Entity (ME) at GANA's lowest level MEs like protocols modules, components, etc., that has an "*ServiceProviding_Interface*" (**SP_I**) through which the services offered by the ME are requested for by some "user-entity" (contrast that to the interface with the upper-layer in the OSI Model); "*ServiceRequesting_Interface*" (**SR_I**) through which the ME requests for services provided by another entity (contrast that to the interface with lower-layer in the OSI Model). A Management Interface of the ME is presented with its sub-interfaces and the Management Primitives that ought to be supported (this is not really the case today, but only a partial fulfilment of the model can be traced in implementation and management of individual networking protocols of today).

5.1.2.2 Primitives/Operations for Enabling ME Programmability

Clause 9.11.4 of ETSI GS AF I002 [2] describes aspects of enabling programmability for MEs and DEs. Programmability, as an enabler refers to the provision of "Primitives/Operations" on the "Management Interfaces" of various types of Managed Entities (e.g. protocols, stacks and networking mechanisms) to enable external Decision Logic (e.g. a DE) that realizes autonomic management and control behaviour to "program" (i.e. start, pause, resume, terminate) the operation of a particular Managed Entity (ME) while at the same time supplying as input, parameter values, policies or behavioural specification, etc., required as parameters of the "Primitives/Operations". Apart from "orchestration" of an ME(s), programming also involves configuring an ME to put it into an operational state such that it can deliver the services it is meant to deliver. Network service programming involves "reading network state" (state of MEs) and "writing state" (performing configuration) into the underlying network by (re-)configuration of network resources, functions, and parameters associated with MEs that need to be (re)-configured in NEs.

NOTE 1: MEs of the nature of protocols and stacks may be initialized and orchestrated by some entities like the Operating system (OS) of an NE as discussed in the clauses in the present document that discuss GANA Operations Procedures and Operations Guide for GANA Empowered Autonomic Networks. In such cases then DEs can be designed to focus on the re-configuration of MEs as driven by some objectives that the DE algorithms may want to enforce during self-adaptive and dynamic management and control operations performed by the DEs during the operation of the NEs and the network.

The characterization of ME interfaces provided in the clauses below include desired properties of Managed Entities (MEs) and Decision Elements (DEs) with respect to Primitives/Operations that should be supported on interfaces (particularly on their Management Interfaces), as well as other desired properties that enable programmability. Some of the properties may not be supported in today's management paradigms and architectural principles applied to designing protocols and other types of managed resources, modules/components. That means the desired properties can be applied in designing future network architectures and networking modules such as protocol modules. The behavioural aspects of the "Primitives/Operations" can be further elaborated by implementers of GANA autonomics and contributed to the evolution of the GANA specification. The programmability aspects for MEs described along with ME Management Interfaces in the present document shall be complemented by aspects that are in clause 9.11.4 of ETSI GS AFI 002 [2].

On the interfaces of the ME model, some of the Operations/Primitives that facilitate for Adaptive Control, Programmability, and Policy Control by an upper manager entity (i.e. an upper DE) are presented.

NOTE 2: The ***Primitives/Operations*** on the interfaces of an ME Model that are presented in the subsequent clauses that follow are just ***skeletons*** that serve to guide some implementations**,** and so their further detailing towards real implementations and additions of other primitives that may be desirable is an open subject.

5.1.2.3 S_I sub-interface of the Management Interface: Sensory_Interface

MEs shall expose "Views" in form of sensory information to its associated DE through the "*SensoryInterface*" (S_I) sub interface of its Management Interface.

Description: The Sensory Interface of an ME enables a manager entity (e.g. an upper DE) to retrieve sensory information concerning the ME state and events the ME exposes for management entities to know and act upon as may be necessary (depending on the management objectives of the manager entity that need to take into account the ME state and/or events in its operations or to derive actionable intelligence and collaborate with other manager entities to dynamically make changes to the ME and/or other MEs in collaboration of the other manager entities).

Sensory Information (Views) exposure and retrieval Operations/Primitives:

1) *Get(VariableList)*: This operation/method is called by the upper DE (caller) to get data concerning some variables (Managed Objects) of the ME (callee), and the ME should respond by sending the values of the variables (parameters) listed in the get operation invoked by an entity such as a DE.

2) *Pull(dataSpec of data to be pulled)*: This method/operation is called by the upper DE to get data from the ME, especially in response to a notification by the ME indicating availability of the data. The DE may have had to register a "callback function" with the ME in order to get notified. The ME may send the changes made at the ME on the ME parameters listed in the pull operation.

3) *Push(dataSpec of data to be pushed)*: This upward call operation/method is called on the DE by the ME (caller). The DE may have had to register a "callback function" with the ME that gets called by the push call by the ME. An ME may use this operation to push information (views) to the upper DE to notify it continuously, or on scheduled basis changes made at the ME on the ME parameters listed in the push operation requested by the DE.

4) *Alert(Event e.g. a Security threat or violation description)*: An ME may use this operation to notify on a threat or violations the designer of the ME encoded to be detectable by the ME.

5.1.2.4 GNSIR_I sub-interface of the Management Interface:General_NonSensoryInformationRetrieval_Interface

An ME should expose non-sensory information to its DE through the GNSIR_I sub interface.

NOTE: MEs that are of the nature of networking protocols implemented in today's networking devices (NEs) usually have some associated management interfaces implemented through MIBs or may provide another way for "manager entities" (e.g. DEs) to manage the ME, but their management interface may possibly not be supporting the notion of a *General_NonSensoryInformationRetrieval_Interface*.

Description: General Non-Sensory operations/primitives shall be used on the interface. These operations/primitives shall be used to exchange non-sensory information. Non-sensory Information is for example the information related to Capability Description Model of an ME, Finite State Machine of an ME, State of an ME, Fault-Error-Failure-Alarm-Causality model of an ME.

Non-Sensory Information exposure and retrieval Operations/primitives:

1) *GetCapability-description()*: returns the aggregate capability model description of the ME if retrievable.

2) *GetFinite-State-Machine()*: returns the state transitions of the ME (in case the ME's Finite State Machine is retrievable).

3) *GetVariable(variable list)*: returns the variables list values of some variables the ME may have been designed to maintain and expose upon request.

4) *Show-Potential-State()*: returns the potential state transitions of the ME if designed to be retrievable.

5) *Show-Actual-State()*: returns the actual state of the ME if designed to be retrievable.

6) *List-Fields-And-Values(fields list)*:returns the fields list values supported by the ME and if retrievable from the ME.

7) *GetFault-Error-Failure-Alarm-Causality-Model()*: returns the Fault-Error-Failure-Alarm-Causality model of the ME (if made available by the ME designer such that it is retrievable). The Fault-Error-Failure-Alarm-Causality model is used for autonomic Fault-Management by the FM_DE.

8) *GetFaiulreModesDescription()*: This is required for autonomic Fault-Management if not covered by the implementation of *GetFaultErrorFailureAlarmCausalityModel()*.

5.1.2.5 E_I sub-interface of the Management Interface: Effector Interface

Effector operations/primitives shall be used by a DE owning the ME to enforce the decisions of the DE that need to be applied on the ME by sending messages or invoking commands supported by the effector interface of the ME that (re)-configure the ME.

Commands/Operations/Primitives and parameters that may be available for a DE to call on an ME:

NOTE 1: Some MEs may not have been designed to support invocation of all or some of the primitives directly, but may support them indirectly via management agents that implement the ME's MIBs.

- *Start(Time) (see note 2).*
- *Pause(Time) (see note 2).*
- *Resume(Time) (see note 2).*
- *Terminate(Time) (see note 2).*
- *Enforce_Policy(PolicySpecification).*
- *De_activate_Policy(PolicySpecification): This causes the ME to switch to its "default" behaviour.*
- *Set(Variable List and New Values).*
- *Create(Connector-Type-Specification, Connection Type Specification) (see note 2).*
- *Connect(Connectors identifiers) (see note 2).*
- *Delete Connector(item-identifier) (see note 2).*
- *Test(Connection-identifier) (see note 2).*
- *Set-Filter(Connector-id, Filter specification) (see note 2).*

NOTE 2: Primitives: *Start, Pause, Resume and Terminate*, are relevant for an ME that has the capability to be orchestrated via the DE assigned to manage and control the ME. Primitives related to "connectors, connections setup, testing and filtering messages" are relevant for an ME that has the capability to be made to connect to some entity (or entities) in order to deliver its services (e.g. in cases such as in dynamic protocol stacks composition) that then exchanges messages with the ME-for which the DE assigned to manage and control the ME can dynamically manage the connections setups and operations on the connectors to be performed by the ME. More details on ME types that may support the concept of dynamic connectors and connections management for functional compositions are discussed in ETSI GS AFI 002 [2].

Primitives used by the caller shall be acknowledged by the *callee*, possibly with inclusion of "time" as parameter in the acknowledgement, and possibly with indications (as parameter) of whether a policy injected by the *caller* could be fully enforced or only part of it.

NOTE 3: The autonomic management and control operations the responsible DE can be designed to exercise on an ME on this interface is limited to what is possible to be performed in reality by the DE to effect a change in the operation of the ME in question either by directly impacting the ME or indirectly (e.g. through DE-to-DE interactions) by configuring or triggering the allocation of resources (e.g. system resources) that enable the ME in question to perform better towards the objectives being enforced by the DE. This also depends on where (i.e. Network Element (NE)) in the network the ME is running, and the various types of inputs (e.g. environment or context-related data/information, or various kinds of monitoring data) the DE is designed to take into consideration in its operations on the ME and other MEs under its responsibility.

5.1.2.6 SP_I Interface: Service Providing Interface

Operation/Primitives (with "parameters") that should be supported on the interface: These are defined in the TCP/IP Model or OSI Reference Model for interlayer interactions of protocols, up to the application layer entities that request for services of the layer beneath them. Usually, such an interface relates to MEs that are of the nature of protocols and upper layer applications in particular.

5.1.2.7 SR_I Interface: Service Requesting Interface

Operations/Primitives (with "parameters") that should be supported on the interface: These are defined in the TCP/IP Model or OSI Reference Model for interlayer interactions of protocols, up to the application layer entities that request for services of the layer beneath them. Usually, such an interface relates to MEs that are of the nature of protocols and upper layer applications in particular.

5.1.2.8 Summary of Interfaces of an ME Model

NOTE: The Management Interface of an ME and the sub-interfaces of the Management Interface are enablers for ME Programmability.

Table 1: Summary of Interfaces of an ME Model

Interface	Sub-interface	Full name of the sub-interface	Caller and Callee/Consumer	Primitives	Reference Clause in the present document
Management Interface	S_I	Sensory Interface	The upper DE (owner of the ME) as the caller for Get() and Pull() Primitives, and "this" ME as callee	*Get(), Pull(), Push()*	5.1.2.3
	GNSIR_I	General Non Sensory Information Retrieval Interface	"This" ME as callee for these primitives and upper DE as caller	*GetCapabilityDescription(); GetFiniteStateMachine(); GetVariable(); Show-Potential-State(); Show-Actual-State(); List-Fields-And-Values(); GetFaultErrorFailureAlarmCausalityModel(); GetFailureModesDescription()*	5.1.2.4
	E_I	Effector Interface	"This" ME as callee for these primitives and upper DE (owner of the ME) as caller	*Start(); Pause(); Resume(); Terminate(); Enforce_Policy(); De_activate_Policy(); Set(); Create(); Connect(); Delete(); Test(); Set-Filter()*	5.1.2.5
Other interaction Interfaces	SP_I	Service Providing Interface	Any Entity that can make use of the services the ME provides (e.g. an upper layer entity in a protocol stack of which the ME is a protocol entity)	Standardized Primitives may exist, e.g. primitives defined for protocol entity interactions in the TCP/IP Model and OSI Model. For certain types of MEs, the designer of the ME may have defined primitives that would apply on this sub-interface	5.1.2.6
	SR_I	Service Requesting Interface	Any Entity that can provide a service to the ME (e.g. a lower layer entity in a protocol stack of which the ME is designed to run on top and use the services provided by the stack)	Standardized Primitives may exist, e.g. primitives defined for protocol entity interactions in the TCP/IP Model and OSI Model. For certain types of MEs, the designer of the ME may have defined primitives that would apply on this sub-interface	5.1.2.7

5.1.3 Structural Model of a Decision-making-Element (DE), DE Interfaces, and Primitives/Operations for Enabling DE Programmability

5.1.3.1 Overview

Clause 9.11.1 of ETSI GS AFI 002 [2] presents the DE Model, which shows the Interface Names and what is happening on each interface, as well as the Primitives that should be supported by a particular DE Interface. A Management Interface of the DE is presented with its sub-interfaces and the Management Primitives that should be supported. This clause adopts (inherits) the DE model from ETSI GS AFI 002 [2] and provides additional complementary perspectives to those provided in the corresponding clause in ETSI GS AFI 002 [2] for consideration and understanding. Figure 8 presents the DE Model.

Decision Elements (DEs) in GANA are containers of Algorithms that determine autonomic behaviour i.e. Self-* properties realized by a DE and such Algorithms/Logic per-se cannot be standardized. The algorithms require operation/primitive to be communicated to and from the entities that communicate with a DE. Different types of algorithms require different types of operation/primitive to flow on the Rfps or interfaces of a DE. The DE Model supports the loading of a control strategy that can be expressed as a Run-Time Executable Behavioural Model (specified and provided as input) i.e. an algorithm in some sense, the DE shall interpret and execute at initialization and during the lifecycle of its operation. For example, there may be different types of algorithms that may determine the logic and behaviour of a DE. This means at any time during the operation of a device, DEs may be loaded, disabled and unloaded by the operator. Practically, in terms of the communication between a DE and ME, Command Line Interface (CLI), NETCONF or SNMP, OpenFlow, and other methods could be used or a new type of a common Unified API could emerge and get standardized. For DE-ME communications that are internal to an NE (GANA node) a common Unified API could be provided and used as described by the Reference Point ***Rfp_GANA-Level2-AccessToProtocolsAndMechanisms***defined in ETSI GS AFI 002 [2] and further described in **Annex B** on **"***Requirements for Protocols and APIs for Enabling GANA based Autonomics, Cognitive Networking and Self-Management of Networks and Services in Evolving and Future Networks*".

GANA Level-2 and Level-3 DEs of a GANA Node (NE) shall realize self-management behaviours/algorithms/logic, including node-local Self-Optimization, i.e. some degree of network element intelligence through the NE internal DEs that realize the internal control-loops. In addition to realizing node-local autonomics, the GANA Level-2 and Level-3 DEs shall be governed and policy-controlled by GANA Level-4 (Network-Level) DEs, and may participate in in-network DE-to-DE interactions via the horizontal reference points depending on the distributed algorithm for the DE-to-DE communications innovated by the DE vendors (suppliers/implementers).

A DE shall configure the behaviour of its assigned MEs (MEs owned by the DE).

DE logic and behaviours shall be governed by some policies from upper DEs, and in case of Network-Level DEs human operator injected policies, objectives, and other types of items provided as inputs by the operator to the Knowledge Plane DEs all govern the Knowledge Plane DEs. A DE shall govern its assigned ME(s) as part of its autonomic management and control objectives.

A DE requires at least two types of Interfaces: DE-ME(s)Interface and a Management Interface.

DEs may interact with others MEs not under their responsibilities but as illustrated in clause 9.11.5 of ETSI GS AFI 002 [2] and also later in the present document.

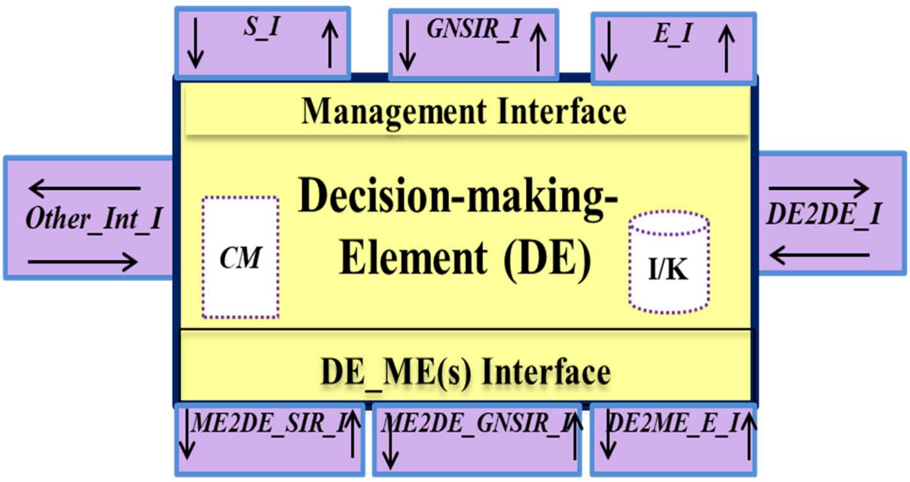


Figure 8: Generic Model of a Decision-making-Element (DE)

The internal modules of DE that may be considered in designing a Decision Element (DE) are depicted on Figure 8. Apart from considering the picture presented in the figure on "*Model of a DE*", the approach that may be taken to designing a DE should consider whether cognitive algorithms (cognition) are required in the DE. A DE may exhibit cognition through the presence of Cognition Module (CM) that implements cognitive algorithm(s). A DE may maintain its own I/K repository for storing some information or knowledge as described later in the clauses that follow. As discussed earlier in the present document, regarding cognition, Network Level DEs "shall" exhibit cognitive behaviour; Node-Level DEs "should" exhibit cognitive behaviour; and Function-Level DEs "may" exhibit cognitive behaviour. The degree of cognition and complexity of DE cognitive algorithms decreases from Network-Level through to the lower GANA Levels. Also refer to clause on characterization of Network-Level DEs.

As discussed in ETSI GS AFI 002 [2], approaches such as FOCALE principles [i.10], IBM's MAPE-K control-loop [i.12], and other principles validated in various research projects on autonomics can be applied to designing a DE's internal structure and logic and associated control-loops.

The Cognition Module (CM) and an I/K repository (or Knowledge Base (KB) Repository) that may be present in DE are described below.

5.1.3.2 Cognition Module (CM) of a DE

This module is meant to implement cognitive algorithms of a DE, and also takes information received from the DE's MEs, other information gathered by the DE and knowledge pertaining to the DE's historical operations over various situations, so as to determine how the DE's cognitive capabilities (intelligence) should be modified to do better in situations encountered in its future operations. The design of the learning module depends very much on the design of the DE module. Depending on the situation, the learning may be continuous (online) or the training may be done beforehand (offline). There may be impacts on the learning mechanisms in the Cognition module for a DE whose behaviour may be modified at run-time. The learning mechanisms may be optional (depends on whether an implementation considers learning as an *essential* part of cognition).

NOTE: As described earlier in the clause "Complex Event Processing (CEP), Context-Awareness, Data Analytics and Cognition in the GANA Knowledge Plane", the Cognition Module should part of Data Analytics capability of a DE that may include other capabilities such as Complex Event Processing (CEP) and Context-Aware Engine, and the presence of the cognition module and CEP and CAE capabilities as part of a Data Analytics capability (module) of DE applies at higher level DEs (going up from the GANA Level-2 up into the GANA Level-4 where the three capabilities are considered necessary to have in the Knowledge Plane).

5.1.3.3 Internal Information/Knowledge (I/K) Repository or DataStorage/Database of a DE

Stores sensory information, non-sensory information, other information received from MEs, DEs or other data/information sources, knowledge received from cognition modules, objectives received from upper DEs and Node Profiles with Configuration Data (including policies, etc.). The Information/knowledge database/storage should make us of an ONIX agent within the GANA Node that publishes some of the information (and information updates) into the ONIX. A DE may be designed to have its own internal information/data repository or use a shared I/K repository with other DEs. Information storage and dissemination depends on the category of information (e.g. consumed outside the DE or the node), and the latency (on-demand, event time or periodically).

NOTE: While I/K Repository is a generalized Information Repository, a Knowledge Base (KB) Repository can be considered as part and parcel of the I/K Repository. The formats of representation of Information or Knowledge stored in the I/K Repository include Ontologies as discussed in the ETSI GS AFI 002 [2] (particularly in reference to internal design and structure of a DE).

5.1.3.4 Primitives/Operations for Enabling DE Programmability

Clause 9.11.4 of ETSI GS AFI 002 [2] describes aspects of enabling programmability for MEs and DEs. As described in the case of *Primitives/Operations for Enabling ME Programmability*, programmability (orchestration and configuration of state) also applies to DEs as "MEs" of some sort that should be orchestrated and/or configured by their upper DEs or special management tools. The characterization of DE interfaces provided in the clauses below include desired properties of Decision Elements (DEs) with respect to Primitives/Operations that should be supported on interfaces (particularly on their Management Interfaces), as well as other desired properties that enable programmability. The behavioural aspects of the "Primitives/Operations" can be further elaborated by implementers of GANA autonomics and contributed to the evolution of the GANA specification. The programmability aspects for DEs described along with DE Management Interfaces in the present document shall be complemented by aspects that are in clause 9.11.4 of ETSI GS AFI 002 [2].

NOTE: The ***Primitives/Operations*** on the interfaces of a DE Model that are presented in the subsequent clauses that follow are just ***skeletons*** that serve to guide implementations**,** and so their further detailing towards real implementations and additions of other primitives that may be desirable is an open subject.

5.1.3.5 An overview on Interfaces of a DE

The DE Model shows the Interface Names and what is happening on each interface, as well as the Primitives that should be supported by a particular DE Interface. A Management Interface of the DE is presented with its sub-interfaces and the Management Primitives that should be supported. The DE Model and Interfaces is adopted/inherited from clause 9.11.1 of ETSI GS AFI 002 [2].Therefore, the characterizing information (descriptions) provided in the present document on this subject are complementing the descriptions provided in clause 9.11.1 of ETSI GS AFI 002 [2]. On the interfaces of the DE model, some of the Operations/Primitives that facilitate for Adaptive Control, Programmability, Policy Control and Loadability of Control Logic by an upper manager entity (i.e. an upper DE) are presented.

NOTE: The characterization of the interfaces of a DE as described in this clause pertains to the generic model of a DE described in Figure 8.

5.1.3.6 E_I sub-interface of the Management Interface: Effector Interface

This sub-interface shall be used by either an upper DE or by some special Tool in the GANA Governance Interface in the case of a Network Level DE (a Knowledge Plane DE) to manage the DE and its operations/behaviours. More details on this subject can be found in ETSI GS AFI 002 [2].

The following primitives should be supported by this sub-interface and are called by the Upper Caller Entity such as the Upper DE or special Tool invoked by the human network operator (more details are found in clause 9.11 of ETSI GS AFI 002 [2]):

1) *Start(Time).*

2) *Pause(Time).*

3) *Resume(Time).*

4) *Terminate(Time).*

5) *Enforce_Policy(PolicySpecification) (see note).*

6) *De_activate_Policy(PolicySpecification): This causes the DE to switch to its "default" behaviour.*

7) *Set(Variable List and New Values).*

8) *Apply_Control_Strategy(Executable Behavioral-Model-Specification, e.g. Extended Finite-State Machine Spec, or State-Chart Spec, etc.) -- Used for providing an executable behavioral model as a control strategy governing a DE's overall behaviour-the DE then executes the executable behavioural model (possibly with the aid of an interpreter of some sort) (see note).*

NOTE: Regarding Policy Control, the *Enforce_Policy(PolicySpecification)* primitive and/or the Apply_Control_Strategy (Executable Behavioral-Model-Specification, e.g. a run-time executable/interpretable Extended Finite-State Machine Specification; or State-Chart Specification that is run-time executable/interpretable; or other types of run-time executable/interpretable behavioural models specifications as parameter(s) of the primitive) should be used. Various types of policies can be employed, e.g. Action Policies, Goal Policies and Utility Function Policies. A comparison of the three types of policies is described in [i.60] and the same reference ([i.60]) discusses why Utility Function Policies can be applied to specifying (by the human operator) high-level objectives for governing autonomic manager components (e.g. GANA DEs-particularly the Knowledge Plane DEs).

5.1.3.7 S_I sub-interface of the Management Interface: Sensory_Interface

A DE shall expose "Views" in form of sensory information to its upper DE through the "*Sensory Interface*" (S_I) sub interface of its Management Interface. The same "Views" may be shared by the DE to another DE(s) via this interface or the DE2DE_I (DE-PeerDE) interface.

The following primitives should be supported by this sub-interface and are called by the Upper DE (more details are found in clause 9.11 of ETSI GS AFI 002 [2]):

1) *Get(Variable List):* This is called by the DE's upper DE to obtain values of variables the DE may have been designed to expose.

2) *Pull(Data Spec of Data to be pulled):* This is called on the DE by its upper DE to obtain data of the DE may have been designed to expose.

3) *Push(Data Spec of the Data to be pushed):* This upward call is called by the DE on its upper DE or can be called on another DE that should receive the data.

5.1.3.8 GNSIR_I sub-interface of the Management Interface: General_NonSensoryInformationRetrieval_Interface

A DE should expose non-sensory information to its upper DE through the "*General_NonSensoryInformationRetrieval_Interface*" (GNSIR_I) sub interface of its Management Interface.

The following primitives should be supported by this sub-interface and are called by the Upper DE (more details are found in clause 9.11 of ETSI GS AFI 002 [2]):

1) *GetCapabilityDescription():* This returns the aggregate capability model description of the DE and its associated MEs.

2) *GetFiniteStateMachine():* This is may be required by an upper DE in order for it to know the state transitions of the DE as an ME.

3) *GetFaultErrorFailureAlarmCausalityModel():* This is required for autonomic Fault-Management if such a model has been embedded by the DE designer to be retrievable by an outer entity such as a Fault-Management-DE.

4) *GetFaiulreModesDescription():* This is required for autonomic Fault-Management if not covered by the implementation of GetFaultErrorFailureAlarmCausalityModel().

5.1.3.9 ME2DE_SIR_I sub-interface of the DE-ME(s) Interface (DeMe): ME-to-DE_SensoryInformationRetrieval_Interface

DE shall retrieve sensory information (i.e. **views** exposed by its ME) of its associated ME through the ME2DE_SIR_I (more details are found in clause 9.11 of ETSI GS AFI 002 [2]).

Description: Sensory Information Retrieval Operation/Primitives shall be used on Sensory Information Retrieval Interface. A DE should retrieve/subscribe to receive sensory information using the following operation/primitives. The ME should respond to the request according to the operations/primitives (more details are found in clause 9.11 of ETSI GS AFI 002 [2]).

Sensory Information Retrieval Operations/Primitives:

1) *Get(VariableList):* A DE may call this operation on an ME to retrieve the values of ME parameters (variables) listed in the get operation.

2) *Pull(dataSpec of data to be pulled):* A DE may use this operation to pull data of interest from an ME. This operation is called on the ME by a DE.

3) *Push(dataSpec of data to be pushed)*: A DE may use this operation to subscribe on demand, continuously, or on scheduled basis to receive data and its updates from the ME, regarding the data indicate in the push operation. This is called on the DE by any Information/Data Supplier of the DE, which could be the ME under its control.

NOTE: By default, the DE may rely on the push operation by the ME in order to minimize the information exchange flows as the ME may invoke a push operation when it has updates to expose as change in "views".

4) *Alert(Security threat or violation description)*: An ME may use this operation to notify on a threat or violations the designer of the ME encoded to be detectable by the ME.

5.1.3.10 ME2DE_GNSIR_I sub-interface of the DE-ME(s) Interface (DeMe): ME-to-DE_General_NonSensoryInformationRetrieval_Interface

A DE shall retrieve non sensory information of its associated ME through the ME2DE_GNSIR_I sub interface (more details are found in clause 9.11 of ETSI GS AFI 002 [2]).

Description: General Non Sensory Information Retrieval operations/primitives shall be used on the interface. These operations/primitives shall be used to retrieve or communicate non sensory information. Non sensory Information is for example the information related to Capability Description Model of an ME, Finite State Machine of an ME, State of an ME, Fault-Error-Failure-Alarm-Causality model of an ME. More details on this subject are found in clause 9.11 of ETSI GS AFI 002 [2].

Non Sensory Information Retrieval Operations/Primitives:

1) *GetCapability-description()*: This is called by a DE in order to return the aggregate capability model description of the associated MEs of the DE.

2) *GetFinite-State-Machine()*: This may be required by a DE in order to know the state transitions of the ME if retrievable.

3) *GetVariable(variable list)*: This is required by a DE in order to retrieve the variable list values of variables the ME is designed to expose.

4) *ShowPotential-State()*: To know the potential state transitions of its associated ME if retrievable.

5) *ShowActual-State()*: This is required by a DE in order to know the actual state of its associated ME if retrievable.

6) *List-Fields-And-Values(fields list)*: This is required by a DE in order to retrieve the fields list values the ME may be designed to expose.

7) *GetFault-Error-Failure-Alarm-Causality-Model():* This is used for autonomic Fault-Management by the FM_DE.

8) *GetFaiulreModesDescription():* This is required for autonomic Fault-Management if not covered by the implementation of GetFaultErrorFailureAlarmCausalityModel().

5.1.3.11 DE2ME_E_Isub-interface of theDE-ME(s) Interface (DeMe): DE-to-ME_Effector_Interface

Descriptions: Effector Operations/Primitives shall be used on the sub-interface. These operations/primitives shall be used by a DE to apply its management and control decision on an ME or a DE under its responsibility.

Commands/Operations/Primitives (parameters): The first four primitives are used to orchestrate, activate/de-activate, and terminate an ME:

1) *Start(Time):* start the ME from the initial state of an ME such that the ME starts operationally executing for a given at a certain time.

2) *Pause(Time):* pause the execution of an ME at a certain time.

3) *Resume(Time):* causes the ME to resume execution at a certain time.

4) *Terminate(Time)*: terminate the ME from operations at a certain time.

5) *Enforce Policy (Policy Specification)*: enforce a policy on the ME using the specification of the policy described (for an ME designed to support this primitive).

6) *De-activate -Policy(Policy specification):* de-enforce the policy set on the ME (for an ME designed to support this primitive). This causes the ME to switch to its default behavior.

7) *Set(Variable list and value):* set values for some management related variables supported by an ME.

8) *Create(Connector-Type-Specification, Connection Type Specification)*: create a connector type and connection specification on the ME (for an ME designed to support this primitive).

9) *Connect(Connectors identifiers):* activate the connection created (for an ME designed to support this primitive).

10) *Delete(item-identifier)*: delete a connector (for an ME designed to support this primitive).

11) *Test(Connection-identifier)*: test a connector(for an ME designed to support this primitive).

12) *Set-Filter(Connector-id, Filter specification):* apply filter to a connector(for an ME designed to support this primitive).

13) *Apply-Control-Strategy(Executable Behavioral Model specification):* used for providing an executable behavioral model as a control strategy governing a DE's overall behavior.

NOTE 1: This primate applies to DEs as "MEs" in particular.

NOTE 2: The autonomic management and control operations a DE can be designed to exercise on an ME on this interface is limited to what is possible to be performed in reality by the DE to effect a change in the operation of the ME in question either by directly impacting the ME or indirectly (e.g. through DE-to-DE interactions) by configuring or triggering the allocation of resources (e.g. system resources) that enable the ME in question to perform better towards the objectives being enforced by the DE. This also depends on where (i.e. Network Element (NE)) in the network the ME is running, and the various types of inputs (e.g. environment or context-related data/information, or various kinds of monitoring data) the DE is designed to take into consideration in its operations on the ME and other MEs under its responsibility.

5.1.3.12 Other_Int_Iinterface of a DE: OtherInteraction_Interface

The *OtherInteraction_Interface* shall be used for communications between a DE and other entities that are not its ME or a DE. For example entities within the environment, such as monitoring tools/components, databases or entities in the underlying substrate of an ME of the DE (e.g. lower layer protocol stack providing services to the ME) that the DE may communicate with for the purposes of obtaining cross layer information as described in ETSI GS AFI 002 [2]) or monitoring data pertaining to the state of some resources whose state (e.g. load condition, etc.) needs to be taken into account by the DE's logic and algorithms in the DE's operations.

Description: A DE designer may exploit Operations/primitives (any interactions methods) made available by entities that the DE needs to interact with in "Other interactions" with an entity that "**is NOT an ME of this DE"** and **"is NOT a PeerDE of this DE on the same level"**. Therefore, the OtherInteraction_Interface of a DE should support primitives/operations that may be used to exchange or retrieve information with some Information/Data supplier or a Service-Providing-Entity e.g. Database.

Example Primitives/operations that should be supported on the interface:

1) *get(variable list):* Retrieve values of some variables available on the entity the DE is made to interact with.

2) *Pull(data spec of Data to be pulled):* Request for some data available on the entity the DE is made to interact with. This operation is expected to be implemented by the entity the DE is made to interact with. An invocation of the Pull(..) operation on the OtherInteraction_Interface of the DE is expected to invoke the Pull(..) operation implemented by the by the entity the DE is made to interact with such that the entity then provides the data the DE is pulling.

3) *Push(DataSpec of Data to be pushed):* This operation/primitive should be implemented by the DE and is called on the DE by the entity the DE is made to interact with when the entity intends to push data (or updates) to the DE at particular time or intervals.

4) *"Service-Request" (…):* This operation/primitive is expected to be implemented by the entity the DE is made to interact with and is invoked by the DE when it requires the service of the service providing entity. The actual nature of the "Service-Request" operation/primitive and the "service" depends on the communication method and "service" supported and implemented by the entity the DE is made to interact with.

5.1.3.13 DE2DE_I Interface of a DE : DE-PeerDE_Interface

As a DE may need to interact with others entities such as MEs or other DEs not directly under its responsibilities, a DE-Peer-DE interface may be required on a DE for interactions with peering DEs.

This interface shall be used by a DE to communicate with another DE at the same Level of Abstraction in GANA (e.g. GANA Level-2 or Level-3).

Descriptions: "DE-PeerDE" Operations/Primitives shall be used by DEs to communicate via the "DE-PeerDE" interface. These operations/primitives shall be used by a DE to exchange information with an ME that is not its own via the DE responsible of the ME. Or to exchange information with a DE at the same decision level within the same control loop in the case the control loop is distributed and the exchanges are transported over the appropriate protocol stack of the NE that enables a remote DE in another NE to receive the messages.

DE-PeerDE Primitives/Operations that should be supported on the Interface: Primitives/Operations are required that enable **Views** to be communicated by a particular Function-Level-DE to other peer Function-level-DEs (e.g. DEs on other nodes/devices), especially concerning events or issues a function of a node e.g. Routing-Function cannot resolve (by performing some action) without jeopardizing network integrity (objectives).**Control Information** exchange also needs to be facilitated between Function-Level-DEs via the DE-2-DE interactions to achieve a "network-intrinsic management and control". Such interactions may include the notion of "compartment formation, policies of operation and compartment management" by DE-2-DE communication in a distributed fashion, and the interactions may include synchronization of actions and policies, negotiations of parameters' values to be configured on their respective MEs as described in Figure 4 of ETSI GS AFI 002 [2] and in the description of the Reference Points (Rfps) concerning DE-to-DE communications.

DEs communicate with each other (within nodes or across nodes) in order to discover services available from their associated MEs by learning the MEs (i.e. capabilities) that have been initialized by DEs and are ready for providing services.

5.1.3.14 Summary of Interfaces of a DE Model

NOTE: The Management Interface of a DE and the sub-interfaces of the Management Interface are enablers for DE Programmability.

Table 2: Summary of Interfaces of a DE Model

Interface	Sub-interface	Full name of the sub-interface	Caller and Callee/Consumer	Primitives	Reference Clause in the present document
Management interface	S_I	Sensory Interface	Upper DE as caller of the *Get() and Pull()* primitives and "this" particular DE as callee; Upper DE as callee of the *Push()* primitive and "this" particular DE as caller	*Get(), Pull(), Push()*	*5.1.3.7*
	GNSIR_I	General Non Sensory Information Retrieval Interface	Upper DE as caller and "this" particular DE as callee for these primitives	*GetCapabilityDescription(); GetFiniteStateMachine(); GetFaultErrorFailureAlarmCausalityModel(); GetFailureModesDescription()*	*5.1.3.8*
	E_I	Effector Interface	Upper DE as caller and "this" particular DE as callee	*Start(); Pause(); Resume(); Terminate(); Enforce_Policy(); De_activate_Policy(); Set(); Apply_Control_Strategy()*	*5.1.3.6*

Interface	Sub-interface	Full name of the sub-interface	Caller and Callee/Consumer	Primitives	Reference Clause in the present document
DE-ME(s) Interface	ME2DE_SIR_I	ME-to-DE_SensoryInformationRetrieval	"This" DE responsible for autonomic management and control of the ME involved as callee for the *Push() and Alert() primitives* and ME as caller; "This" DE responsible for autonomic management and control of the ME involved as caller for the *Get() and Pull() primitives* and ME as callee	*Get(); Pull(); Push(); Alert()*	*5.1.3.9*
	ME2DE_GNSIR_I	ME-to-DE_General_NonSensoryInformationRetrieval Interface	ME (which could be a pure ME or a lower DE as ME of "this" responsible DE) as callee	*GetCapability-description(); GetVariable(); ShowPotential-State(State (); ShowActual-State () GetFaiulreModesDescription()*	*5.1.3.10*
	DE2ME_E_I	DE-to-ME_Effector_Interface	ME (which could be a pure ME or a lower DE as ME of "this" responsible DE)	*Start(); Pause(); Resume(); Terminate(); Enforce Policy(); De-activate -Policy(); Set(); Create(); Connect(); Delete(); Test(); Set-Filter(); Apply-Control-Strategy();*	*5.1.3.11*
Other Interaction Interface	Other_Int_I	Other Interaction_Interface	An entity that is NOT an ME of "this" particular DE and is NOT a DE, as callee mainly (or possibly as caller as well) for the Primitives	*Get(); Pull(); Push(); "Service-Request"()*	*5.1.3.12*
DE-PeerDE interface	DE2DE_I	DE-PeerDE_Interface	Any Peer DE required to interact with "this" particular DE, as caller or callee for the Primitives	*ConveyMessage()* (see note)	*5.1.3.13*
NOTE: Other types of Primitives for DE to DE interactions may be defined.					

5.1.3.15 Assignment of Managed Entities (MEs) and their Configurable and Controllable Parameters to specific Decision Elements (DEs) in GANA ("*1-ME-Param*"-mapped to-"*1-DE*")

5.1.3.15.1 Concept of "ownership" of MEs in GANA

The "Concept of Ownership" of MEs and their configurable and controllable parameters is a feature of the intrinsic stability attributes that shall be considered for autonomic network architecture (as is also defined in GANA). This concept requires that every ME is managed by a single DE, i.e. no two DEs (i.e. control loops) can control the same ME (i.e. functionality, resource, etc.) at any given point of time in the network. This is important from system's stability point of view since it relieves the burden of "conflicts resolution". Specifically, if an ME is controlled by two or more DEs at the same time, then, contrasting, conflicting and at times repetitive policies, objectives and reconfiguration requests, etc., originating from different DEs would lead to an unstable ME or poorly optimized network and thus, to an unstable autonomic network. Through the "Concept of Ownership" of MEs and their configurable and controllable parameters, GANA ensures that this instability is avoided. An ME Parameter is assigned to and owned by exactly one DE, i.e. from an ME Parameter point of view (because a DE may own more than one ME parameter). In modelling terms, e.g. in UML (Unified Modeling Language, standardized by OMG), a "*DE-to-ME_Parameter Relationship*" would have multiplicity of "1" at the DE end point and multiplicity of "n" at the ME_Parameter end point. This clause illustrates how a DE relates in communication and interfaces to its own MEs, to other DEs, and to MEs owned by other DEs. A Table that maps specific types of MEs and their configurable and controllable Parameters to specific types of DEs is provided later in the clause (clause 5.1.3.16) on "GANA DEs Hierarchy and Hierarchical Control-Loops". The subject of approaches and techniques for addressing stability for autonomic components (i.e. DEs) is discussed in ETSI GS AFI 002 [2]. Among the techniques and approaches are approaches for addressing potential stability related problems whilst still at the design time, simulation and validation time for GANA DEs, before employing in addition, approaches and techniques for addressing stability issues at run-time phase for DEs (their interactions and convergence in achieving the objectives and goals of the network and its services). Approaches for addressing potential stability related problems whilst still at the design time, simulation and validation time for GANA DEs include the approach prescribed in [i.59] on "*Formal Methods for Modeling, Refining and Verifying Autonomic Components of Computer Networks*", which also includes the aspect of semantic checking as discussed in [i.60]. Work in [i.72] and [i.73] also provide insights into Meta-Modeling and model-driven approaches to designing and verifying autonomic components of computer and telecommunication networks. Complementarily, approaches and techniques for addressing stability issues at run-time phase for DEs include those described in [i.23] and in [i.30].

Clause 9.11.5 of ETSI GS AFI 002 [2] provides illustrate various types of relationships that can be established and designed between DEs and MEs and the corresponding interfaces that can be used in DE-to-DE communications and in DE-to-ME communications. If a Protocol, Tool, or Component can be used directly by any functional entity (a DE or an ME of some sort) and by a number of functional entities simultaneously without conflicts in the services it offers to the requesting functional entities, then it can be used directly by any functional entity. However, in such a case, a DE assigned to be the manager for the Protocol, Tool, Component, may need to maintain knowledge of the functional entities using it, possibly including statistics, etc. Figure 27, Figure 28 and Figure 29 in ETSI GS AFI 002 [2], illustrate that if direct use of a Protocol, Tool, or Component by multiple entities simultaneously would result in conflicts in the services it offers, then REQUESTS from functional entities intending to use it shall go through the DE acting the role of manager for the Protocol, Tool, or Component, so that REQUESTS may be accepted or rejected, and conflicts resolved by the DE (manager) responsible. If direct use of a Protocol, Tool, or Component by multiple entities simultaneously would result in conflicts in the services it offers, then REQUESTS from functional entities intending to use it, shall go through the DE acting as manager for the Protocol, Tool, or Component, so that REQUESTS may be accepted or rejected, and conflicts resolved by the DE (manager) responsible. DEs need to discover each other.

Figure 9, Figure 10, Figure 11 and Figure 12 illustrate the various DE-to-DE and DE-to-ME interaction scenarios that need to be considered by DE implementers in following the concept of ownership of MEs by DEs. Just for illustration purposes the diagrams relate to MEs that are of protocol and protocol stacks nature.

Illustrations of Vertical Interactions:

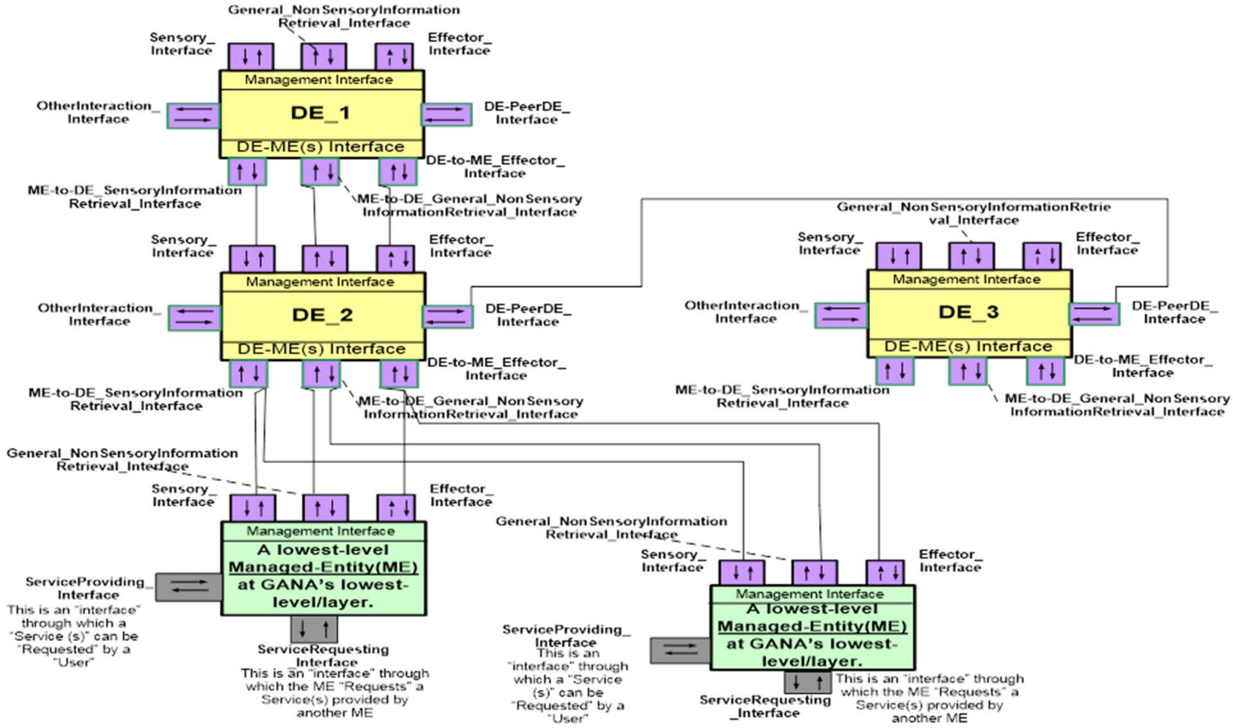


Figure 9: A DE with multiple MEs and interfacing with an Upper and a PeerDE (which could be a sibling DE since a Sibling relation is a special-type of a Peer Relation)

Illustrations of Horizontal Interactions:

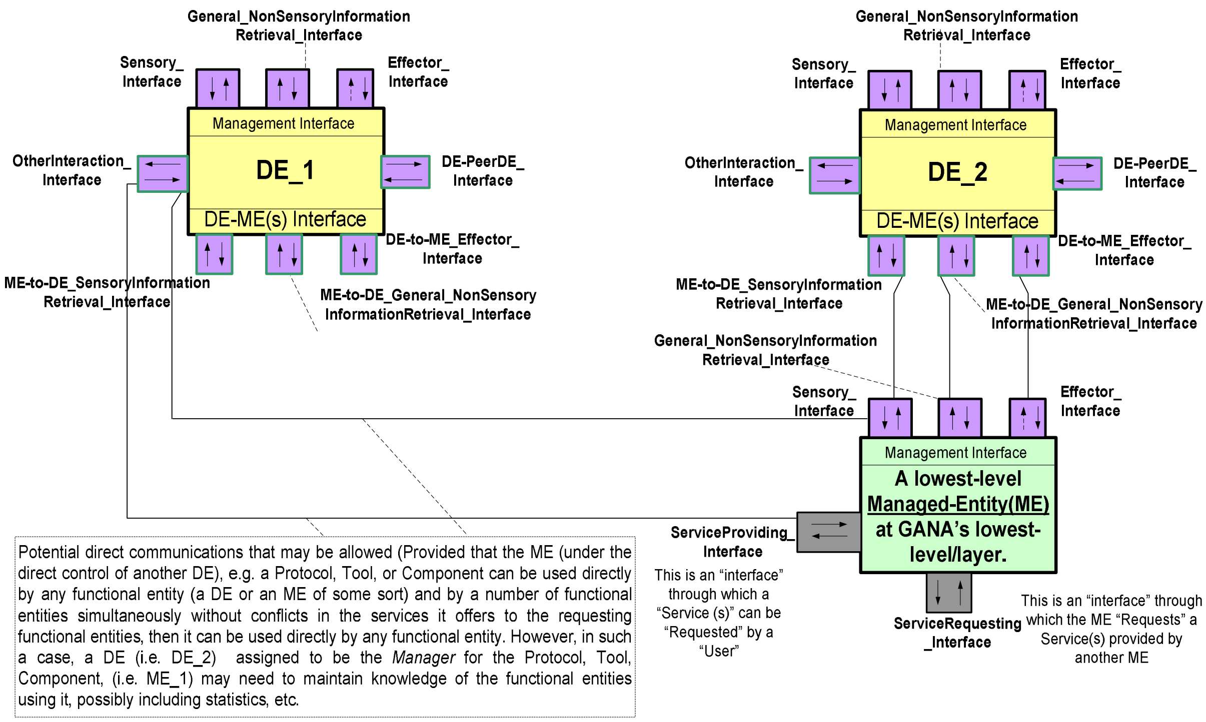


Figure 10: A situation under which a DE is allowed to interact directly with an ME that is not its own

Potential direct communication that may be allowed (Provided that the ME (under the control of another DE) e.g. Tool or component can be used directly by any functional entity (DE / ME /Other) and by functional entities without conflicts in the services if offers to the requesting functional entities, then it can be used directly by any functional entity. However, in such a case, a DE (i.e. DE 2) assigned to be the Manager for the ME (i.e. ME 1) may need to maintain knowledge of the functional entities using it, possibly including statistics, etc.

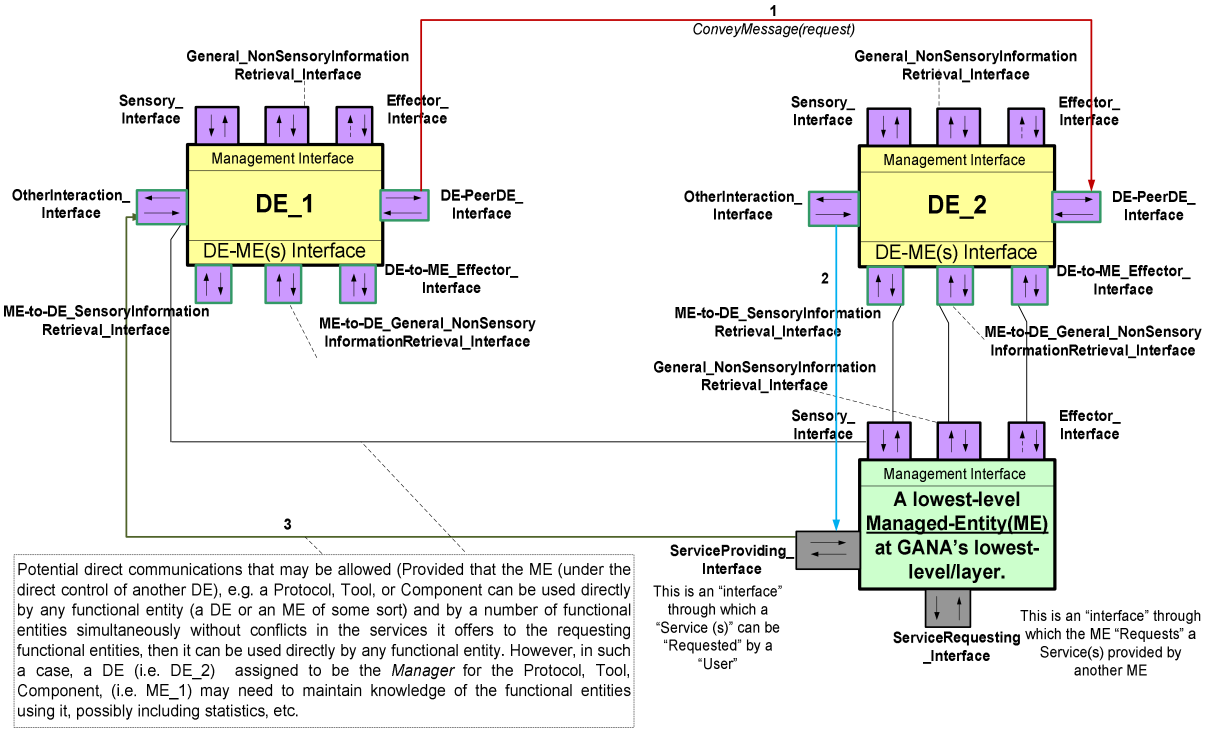


Figure 11: A situation under which the DE may need to communicate with an ME that is not its own, via the DE of the ME

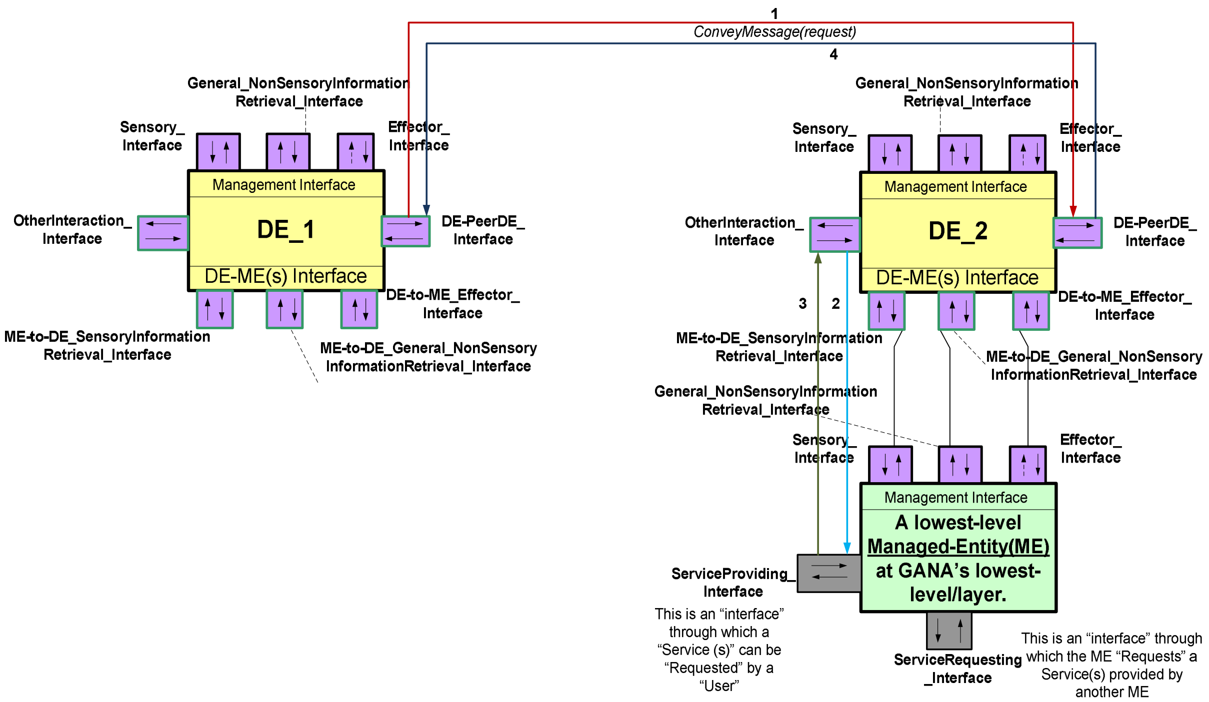


Figure 12: A case in which the DE may need to communicate with an ME that is not its own, via the DE of the ME

5.1.3.16 GANA DEs Hierarchy and Hierarchical Control-Loops, and Mappings of Managed Entities (MEs) and their Configurable and Controllable Parameters to specific DEs ("*1-ME-Param" to* "*1-DE* Mapping")

5.1.3.16.1 An Overview on Mappings of Various Types of Managed Entities (MEs) and their Configurable and Controllable Parameters to Specific DEs

Table 3 (Table 1 in clause 9.11.6 of ETSI GS AFI 002 [2]) presents a mapping of DEs from the Network-Level down to the GANA lowest layer/level Managed Entities (MEs).

NOTE 1: From Left to Right: It means viewing from the Network into a Network Element down to Protocols, Stacks and Mechanisms assigned to be managed by the assigned DEs as Managed Entities (MEs). Configurable and Controllable Parameters of MEs are assigned to specific DEs in a "1-ME-Param" -to- "1-DE" Mapping, i.e. from an ME Parameter point of view (because a DE may own more than one ME parameters). MEs used in this illustration are Protocols and Mechanisms that need to be orchestrated and dynamically managed by their respective DEs.

NOTE 2: The mappings of DEs to specific MEs and their configurable and controllable Parameters get refined and more specific when DEs are being "instantiated" onto a target implementation Reference Architecture (refer to [3] on Implementation/Instantiation Guide for GANA). And, some DEs on the network level that "mirror" specific DEs on Level-2 (Function-Level) may be missing in the Table, and so, the decision on whether it is necessary to have such DEs also on the network-level is left to the decisions that should be taken during any work on instantiations of the GANA Reference Model onto various architecture types and environments to enable autonomicity, cognitive networking and self-management in those architectures. For more insightful information on the extensibilities of functionalities of some various DEs reader may refer to clause 9.11.6 of ETSI GS AFI 002 [2].

NOTE 3: There is on-going work in ETSI on elaborations of the Implementation Guide for GANA and the resultant work is expected to be published in ETSI TR 103 195-3 [i.71].

Table 3: A mapping of DEs from the Network-Level down to the GANA lowest layer/level Managed Entities (MEs) see more details in clause 9.11.6 of ETSI GS AFI 002 [2] and the note below

Network-Level DEs	Node-Level DEs	Function-Level DEs	Protocols and Mechanisms as Managed-Entities (MEs)	Examples of protocols and Mechanisms that are mapped as MEs
	GANA NODE			
NET_LEVEL_SEC_M_DE	NODE_LEVEL_SEC_M_DE		**Security Protocols, Algorithms and Mechanisms**	Certificates/Passwords Algorithms, Hash Algorithms, Encryption Algorithms, Access Control Mechanisms, Trust Mechanisms, Denial of Service (DoS) Detection/Prevention algorithms/mechanisms, Signature based intrusion detection mechanisms, etc.
NET_LEVEL_FM_DE	NODE_LEVEL_FM_DE		Fault Detection Mechanisms, Fault Isolation/Localization/Diagnosis Mechanisms, Fault Removal Mechanisms	Active Probing mechanisms, Bi-Directional Forwarding Detection (BFD protocol) for link failure detection, Self-test/diagnose functions, rebooting, reloading, automated module replacement mechanisms, etc.
NET_LEVEL_RS_DE	NODE_LEVEL_RS_DE		Proactive and Reactive Resilience Mechanisms, Survivability Strategies and Algorithms, Restoration and Protection Mechanisms	Node Resilience mechanisms, and Network Resilience mechanisms, etc.
	NODE_LEVEL_AC_DE		Neighbour Discovery Protocols/Mechanisms and Network Discovery Mechanisms	Neighbour Discovery Protocol (NDP), Secure Neighbour Discovery Protocol (SEND), etc.
NET_LEVEL_RM_DE		FUNC_LEVEL_RM_DE	Routing Protocols and Mechanisms	OSPF, BGP, RIP, ISIS, etc.
NET_LEVEL_FWD_M_DE		FUNC_LEVEL_FWD_M_DE	Layer-3 Forwarding Protocols and Mechanisms, Layer-2.5-Fowarding, Layer-2-Fowarding, Layer-3-Switching, Layer-2-Switching, etc.	IPv4/IPv6 Forwarding Engine, Multi-Protocol Label Switching (MPLS), etc.
NET_LEVEL_QoS_M_DE		FUNC_LEVEL_QoS_M_DE	QoS Protocols and Mechanisms	Packet classifier, Packet Marker, Queue Management, Queue Scheduler, RSVP, etc.
NET_LEVEL_MOM_DE		FUNC_LEVEL_MOM_DE	Mobility Management Protocols and Mechanisms	Mobility Support in Internet Protocol Version 6 (MIPv6), Datagram Congestion Control Protocol, Mobile Stream Control Transmission Protocol, Site Multi-homing by IPv6 Intermediation, Proxy-Mobile-IP, Mobility-Management User-Equipment Managed-Entity, Measurement-Report-Function Managed-Entity, Candidate-Access-Router-Discovery mechanism, Fast Handover Scheme, Policy Control and Charging Rules Function mechanism, etc.
NET_LEVEL_MON_DE	NODE_MAIN_DE	FUNC_LEVEL_MON_DE	Monitoring Protocols, Mechanisms and Tools	IPFIX data collection and dissemination mechanisms, SNMP data collection and dissemination mechanisms, NETFLOW data collection and dissemination mechanisms, Protocol Analysers, Packet Trace creation and dissemination mechanisms. Effective and Available Bandwidth Estimation mechanisms, IPv6 hop-by-hop options for intrinsic monitoring, etc.
		FUNC_LEVEL_SM_DE	Services and Applications	Orchestration of services, service-discovery, interpretation of service and application requirements at run-time and requesting the network layer to behave in a service/application-aware manner, realizing a control-loop over the services/applications as its Managed Entities (MEs), collaboration with other DEs of responsible of autonomic management of the network layer protocols in order to realize collaborative self-adaptation on both the service-layer and the network-layer.

NOTE: There are other DEs that may have not been included in the Table 3 and implementers should take them into account based on their descriptions provided in the present document. Such DEs include Network-Level-Generalized Control Plane-Management-DE (NET-LEVEL-GCP_M_DE), Function-Level-Generalized Control Plane-Management-DE (FUNC-LEVEL-GCP_M_DE), Network Level End-to-End “end-user oriented” Service and Applications Management (NET_LEVEL_E2E_Service_M).

5.1.3.16.2 GANA level 1 DE (the lowest)-called "Protocol" Level DE

Protocol Level DE relates to any Managed Entity (ME) such as protocol, network service enabler, software components, and other fundamental mechanisms that may exhibit intrinsic control-loops and associated DE Logic, as it is the case for most of today's protocols such as the routing protocol, OSPF (Open Shortest Path First). OSPF may be considered as an example of the instantiation of a Protocol-Level-DE. Since the Protocol-Level-DE involves embedding an intrinsic control loop within an individual protocol, it may not be to introduce such "intelligence" into individual protocols. Avoiding "protocol-intrinsic control-loops" is recommended since they may complicate network manageability and may create undesired emergent behavior in complex protocol interaction scenarios as known today. Therefore, GANA hierarchy covers only the three other levels which should collaboratively work together.

5.1.3.16.3 GANA level-2 DE-called "Function" Level DE

5.1.3.16.3.1 A General Characterization of GANA Level 2 DEs (called Function-Level DEs)

A Function Level DE relates to a DE for collective autonomic management and control of a group of protocols and mechanisms that are abstracted (viewed like a bundle) by a particular "networking or a management/control function". MEs of a function-level DE includes networking resources layer entities such as protocols that do not embed decision-logic and associated control-loops (referred to as protocol-level DEs in GANA) and also networking resources layer entities such as protocols that exhibit intrinsic protocol-level DEs that drive protocol-intrinsic control-loops of some protocols, provided that both types of entities are "abstracted" by the same "function".

The following DEs are the only ones identified at this level in GANA:

- "routing management-DE";
- "forwarding management-DE";
- "Quality of Service management-DE";
- "mobility management-DE";
- "monitoring management-DE";
- "service and application management-DE".

Currently these six Function Level DEs are defined in the GANA Reference Model. The control-loop is external to the individual protocols and mechanisms subscribed to the "Function" (by virtue of abstraction).

DEs that can be instantiated into a node (NE) at this level are determined by the "networking functions" required to be operationally supported by the node (NE) and realized by specific Managed Entities (MEs) such as protocols and mechanisms or tools associated with the networking functions. More details on this subject are found in ETSI White Paper No.16 [3].

NOTE: There is ongoing work in ETSI on elaborations of the Implementation Guide for GANA and the resultant work is expected to be published in ETSI TR 103 195-3 [i.71].

5.1.3.16.3.2 Function-Level-Routing Management-DE (FUNC_LEVEL_RM_DE)

Decision Element (DE):

- Function-Level-Routing Management-DE.

Managed Entities (MEs):

- Routing Protocols; and
- Mechanisms.

Example of protocols and mechanisms that are mapped as MEs of the DE:

- OSPF;
- BGP;

- RIP;
- ISIS;
- etc.

Description: the FUNC_LEVEL_RM_DE provides logic, algorithm to ensure the routing of packets in the network in order to optimize the network.

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.32], [i.33], [i.36], [2], [3], [i.46], [i.47], [i.52] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive)(re)-configuration of routing related Managed Entities (MEs) based on reacting to feedback on network and ME(s)state (continuously observed/monitored) and experienced challenges in the network (e.g. link(s) congestion, link and node failures), changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in network performance and QoS for some traffic flows or other objectives (targets).

5.1.3.16.3.3 Function-Level-Forwarding and Data Plane Management-DE (FUNC_LEVEL_FWD_M_DE)

Decision Element (DE):

- Function-Level-Forwarding; and
- DataPlane Management-DE.

Managed Entities (MEs):

- Layer-3 Forwarding Protocols and Mechanisms.
- Layer-2.5-Forwarding.
- Layer-2-Forwarding.
- Layer-3-Switching.
- Layer-2-Switching.
- Layer-1 or 0 related data forwarding mechanisms.

Example of protocols and mechanisms that are mapped as ME of the DE:

- IPv4/IPv6 Forwarding Engine;
- Multi-Protocol Label Switching (MPLS);
- VLAN;
- etc.

Description: the FUNC_LEVEL_FWD_DE provides logic, algorithm that autonomically manages the forwarding protocols and mechanisms of the node in order to optimize the forwarding behaviour of the node so as to meet certain objectives.

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.36], [i.47], [i.46], [i.52] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration of Traffic Flow Forwarding and Switching related Managed Entities (MEs) based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in network performance and QoS for some traffic flows or other objectives (targets).

5.1.3.16.3.4 Function-Level Quality of Service Management-DE(FUNC_LEVEL_QoS_M_DE)

Decisions Element (DE):

- Function-Level Quality of Service Management-DE.

Managed Entities (MEs):

- QoS Protocols; and
- Mechanisms.

Examples of protocols and mechanisms that are mapped as ME of the DE: Packet classifier, Packet Marker, Queue Management, Queue Scheduler, RSVP (Resource ReSerVation Protocol), other protocols and mechanisms for QoS provisioning (e.g. for DiffServ/IntServ models in IP networks).

Description: the FUNC_LEVEL_QoS_M_DE provides logic, algorithms to ensure QoS for services and improve QoE of services within the GANA node and other nodes through the collaboration of their FUNC_LEVEL_QoS_M_DEs.

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.44], [i.45], [i.47], [i.49], [i.51] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration or orchestration of Quality of Service enforcement (or improvement)related Managed Entities (MEs) based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges in the network (e.g. QoS degradation for some traffic flows), changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in network performance and QoS for some traffic flows or other objectives (targets).

5.1.3.16.3.5 Function-Level-Mobility Management-DE(FUNC_LEVEL_MOM_DE)

Decisions Element (DE):

- Function-Level-Mobility Management-DE.

Managed Entities (MEs) of the DE:

- Mobility Management Protocols and Mechanisms.

Example of protocols and mechanisms that are mapped as MEs of the DE: Mobility Support in Internet Protocol Version 6 (MIPv6), Datagram Congestion Control Protocol, Mobile Stream Control Transmission Protocol, Site Multi-homing by IPv6 Intermediation, Proxy-Mobile-IP, Mobility-Management User-Equipment Managed-Entity, Measurement-Report-Function Managed-Entity, Candidate-Access-Router-Discovery mechanism, Fast Handover Scheme, Policy Control and Charging Rules Function mechanism, etc.

Description: the FUNC_LEVEL_MOM_DE provides logic, algorithm(s) to manage mobility related MEs hosted by NE, for example, MEs that help drive handover or handoff of devices/nodes between different networks and technologies and also ensure service continuity of applications flows.

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.47], [i.48], [i.53] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration or orchestration of Mobility enablement (or adaptation)related Managed Entities (MEs) based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges in the network (e.g. QoS degradation for some traffic flows, network connectivity errors and failures), changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in network performance and QoS for some traffic flows or other objectives (targets).

5.1.3.16.3.6 Function-Level-Monitoring Management-DE (FUNC_LEVEL_MON_DE)

Decision Element (DE):

- Function-Level-Monitoring Management-DE.

Managed Entities (MEs) of the DE:

- Monitoring Protocols;
- Mechanisms and Tools (e.g. for traffic monitoring, context inference and derivation, etc.).

Example of protocols and mechanisms that are mapped as ME of the DE: IPFIX data collection and dissemination mechanisms, SNMP data collection and dissemination mechanisms, NETFLOW data collection and dissemination mechanisms, Protocol Analysers, Packet Trace creation and dissemination mechanisms. Effective and available bandwidth estimation mechanisms, IPv6 hop-by-hop options for intrinsic monitoring, etc.

Description: the FUNC_LEVEL_MON_DE provides logic and algorithms to orchestrate monitoring MEs or to (re)-configure them, and to retrieve information from various potential sources of monitoring data and information in order to intelligently cause dissemination of monitoring data needed by DEs within the node to the local DEs by causing the Monitoring MEs to disseminate the monitoring data required by the local DEs or to register the DEs to receive monitoring data or information of interest to them if available through the Monitoring DE itself. The Monitoring DE also pushes monitoring data to external Data Collectors of the network. The FUNC_LEVEL_MON_DE's logic should include an algorithm to retrieve information (subscribe for information) and process the information indicative about the operations of the GANA node in order to infer and adaptively enforce the monitoring behaviour and monitoring-data flow of the GANA node that is needed by DEs within the node.

NOTE 1: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.47], [i.50], [i.51], [i.54], [i.55], [i.56] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration or orchestration of Monitoring related Managed Entities (MEs) based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges in the network (manifestations of faults, errors, failures, security attacks), changes in human operator policies and operational context, monitoring-data demands from other types of entities (e.g. other autonomic manager components), or workload changes, in order to achieve certain objectives such as the gathering and optimal dissemination of various types of monitoring data to entities that need to act upon the monitoring data in order to effect changes in network behaviours or other objectives (targets).

NOTE 2: A Function-Level-Monitoring Management-DE (FUNC_LEVEL_MON_DE) could be implemented as an integral part of a Probe (e.g. a Passive Monitoring Probe, Active Monitoring Probe or one that combines active probing and passive probing aspects) if the Probe is embedded with an NE and has a scope of being a NE-scoped (node-scoped) Monitoring Probe.

5.1.3.16.3.7 Function-Level-Service Management-DE (FUNC_LEVEL_SM_DE)

Decisions element (DE):

- Function-Level-Service Management-DE.

Managed Entities (MEs):

- Services and Applications layer entities, e.g. TCP/IP or OSI Application-Layer Service ME.

Example of protocols and mechanisms that are mapped as ME: Orchestration of services, service-discovery, interpretation of service and application requirements at run-time and requesting the network layer to behave in a service/application-aware manner, realizing a control-loop over the services/applications as its Managed Entities (MEs), collaboration with other DEs of responsible of autonomic management of the network layer protocols in order to realize collaborative self-adaptation on both the application service-layer and the network-layer. This is meant for application/service designed or hosted to operate on the application layer of the TCP/ IP internet model (e.g. RTCP, SIP, DHCP, etc.) or on the OSI application layer.

Description: the FUNC_LEVEL_SM_DE provides logic, algorithms to autonomically manage TCP/IP Application Layer or OSI Application Layer types of applications hosted within a node.

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [2] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration of TCP/IP or OSI Application-Layer Service Components related Managed Entities (MEs) based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (e.g. service or server overload situation, service faults/errors/failures) in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in service performance (for the service/application hosted by the node) and QoS for some traffic flows or other objectives (targets).

5.1.3.16.3.8 Function-Level-Generalized Control Plane Management-DE (FUNC_LEVEL_GCP_DE)

Decisions Element (DE):

- Function-Level-Generalized Control Plane Management-DE.

Managed Entities (MEs):

- Manages control plane protocols and mechanisms that are not abstracted and covered by other DEs that autonomically manage and control certain aspects of the control plane, namely Routing-Management-DE;
- Mobility-Management-DE;
- QoS-Management-DE;
- Monitoring-DE.

Example of protocols and mechanisms that are mapped as ME of the DE: FUNC_LEVEL_GCP_DE autonomically manages control plane protocols and mechanisms. Due to the need for "further specializations" of the Control-Plane, there is a special type of such a DE, called the Function-Level Routing-Management-DE (FUNC_Level_RT_M DE) that dynamically (autonomically) manages routing protocols and mechanisms of the node and needs to complementarily work together with a counterpart on the Network-Level (Network-Level-Routing-Management-DE). This means other types of specialized FUNC_LEVEL_GCP_DE may be introduced, else designers may simply use the FUNC_LEVEL_GCP_DE to autonomically manage control plane protocols and mechanisms such as GMPLS and other protocols that are not abstracted and covered by the DEs that autonomically manage and control certain aspects of the control plane, namely Routing-Management-DE, Mobility-Management-DE, QoS-Management-DE, Monitoring-DE.

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE. There are various sources that can be considered, such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration of Generalized Control Plane related Managed Entities (MEs) based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in network performance and QoS for some traffic flows or other objectives (targets).

5.1.3.16.4 GANA Level-3 DE-called Node-Main-DE

5.1.3.16.4.1 A General Characterization of GANA Level 3 DEs (collectively called the Node-Main-DE (consisting of the sub-DEs))

The Node-Main-DE orchestrates and manages DEs on the Function Level DE and may manage MEs not managed by the Function-level-DE. The Node-Main-DE relates to a DE for autonomic management and control of those aspects that cover and restrict the behavior of the node as a whole, as well as the orchestration and policing of the "Function Level-DE".

The following four DEs have been identified as sub-DEs constituting the Node-Main-DE:

- *Security Management DE.*
- *Fault Management DE.*
- *Auto configuration and auto-discovery DE.*
- *Resilience and Survivability DE.*

All the four sub-DEs of the Node Main DE shall be present in any GANA Node, an autonomic NE.

5.1.3.16.4.2 Node-Level-Security Management-DE (NODE_LEVEL_SEC_M_DE)

Decision Element (DE):

- Node-Level-Security Management-DE (NODE_LEVEL_SEC_M_DE).

Protocols and Mechanisms as Managed-Entities (MEs):

- Security Protocols, Algorithms and Mechanisms that can be applied on the node level.

Examples of protocols and Mechanisms that are mapped as MEs of the DE: Certificates/Passwords Algorithms, Hash Algorithms, Encryption Algorithms, Access Control Mechanisms, Trust Mechanisms, Denial of Service (DoS) Detection/Prevention algorithms/mechanisms, Signature based intrusion detection mechanisms, etc.

Description: the NODE_LEVEL_SEC_M_DE is used to manage any security issues in the Node. NODE_LEVEL_SEC_M_DE may also be used to secure and authenticate interactions between DEs and MEs within the same domain or to secure interaction with other MEs and DEs within different domain.

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.34] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration or orchestration of Security enforcement (or improvement)related Managed Entities (MEs) based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (e.g. security threats) in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in node security or security for some traffic flows or other objectives (targets).

5.1.3.16.4.3 Node-Level-Fault Management-DE (NODE_LEVEL_FM_DE)

Decision Element (DE):

- Node-Level - Fault Management-DE (NODE_LEVEL_FM_DE).

Protocols and Mechanisms as Managed-Entities (MEs):

- Fault Detection Mechanisms;
- Fault Isolation/Localization/Diagnosis Mechanisms;
- Fault Removal Mechanisms.

Examples of protocols and Mechanisms that are mapped as MEs of the DE: Active and Passive Probing mechanisms, Bi-Directional Forwarding Detection (BFD protocol) for link failure detection, Self-test/diagnose functions, rebooting, reloading, automated module replacement mechanisms, etc.

Description: The NODE_LEVEL_FM_DE is for autonomic fault management within the GANA Node by employing appropriate Fault Detection Mechanisms, Fault Isolation/Localization/Diagnosis Mechanisms, and Fault Removal Mechanisms that enable to repair the node's components and the node as a whole. NODE_LEVEL_FM_DEs of various GANA nodes may collaborate in achieving distributed autonomic fault-management that requires the collaboration of various nodes. The Node-Level-Fault Management-DE (NODE_LEVEL_FM_DE) interworks with the Node-Level-Resilience&Survivability-DE (NODE_LEVEL_R&S_DE) as described in [i.18] and in ETSI GS AFI 002 [2].

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.18], [i.47], [i.55] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration or orchestration of Fault-management (fault-detection, fault-diagnosis and fault-removal(repair)) related Managed Entities (MEs) based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (e.g. fault/error/failure detection) in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as node self-diagnosis and self-repair (self-healing) or other objectives (targets).

5.1.3.16.4.4 Node-Level-Resilience&Survivability-DE (NODE_LEVEL_R&S_DE)

Decision Element (DE):

- Node-Level-Resilience &Survivability-DE (NODE_LEVEL_R&S_DE).

Protocols and Mechanisms as Managed-Entities (MEs):

- Proactive and Reactive Resilience Mechanisms;
- Survivability Strategies and Algorithms;
- Restoration and Protection Mechanisms (e.g. for traffic flows).

Examples of protocols and Mechanisms that are mapped as MEs of the DE:

- Node Resilience mechanisms; and
- Network Resilience mechanisms.

Description: the NODE_LEVEL_RS_DE provides logic, algorithms to ensure the resilience and survivability of the node(system) and the network (of some scope) in collaboration with other peer NODE_LEVEL_RS_DEs in other GANA nodes. The Node-Level-Resilience &Survivability-DE (NODE_LEVEL_R&S_DE) interworks with the Node-Level-Fault Management-DE (NODE_LEVEL_FM_DE) as described in [i.18] and in ETSI GS AFI 002 [2].

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.18], [i.47], [i.55] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration or orchestration of "Resilience &Survivability strategies" related Managed Entities (MEs) based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (e.g. fault/error/failure detection) in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in node resilience or resilience and survivability for some traffic flows or other objectives (targets).

5.1.3.16.4.5 Node-Level-Auto-Configuration and auto-discovery DE (NODE_LEVEL_AC_DE)

Decision Element:

- Node-Level-Auto-Configuration and auto-Discovery-DE (NODE_LEVEL_AC_DE).

Protocols and Mechanisms as Managed-Entities (MEs):

- Function-Level-DEs and the other sub-DEs of the Node-Main-DE, which may all need to be orchestrated and initialized and configured with their corresponding Profiles by the NODE_LEVEL_AC_DE.

NOTE 1: More details on this subject are found in the clauses of the present document that discuss aspects pertaining to governance of the autonomic network (e.g. the clause "GANA Operations Procedures for the Human Network Operator").

- Neighbour Discovery Protocols/Mechanisms and Network Discovery Mechanisms.

Examples of protocols and Mechanisms that are mapped as MEs of the DE:

- IPv6 Neighbour Discovery Protocol (NDP);
- IPv6 Secure Neighbour Discovery Protocol (SEND);
- etc.

Description: NODE_LEVEL_AC_DE provides logic, algorithms to ensure plug and play mechanisms of GANA Node. Provide auto discovery mechanisms and auto configuration mechanisms. The NODE_LEVEL_AC_DE may self-adapt the GANA Node according to GANA Profile derived from the KP's Network Level_AC_DE and orchestrate the different DEs within the GANA Node and in collaboration with other DEs outside the GANA node.

NOTE 2: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.36], [i.38], [i.46], [i.47], [2] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration or orchestration of the node's Managed Entities (MEs) that enable the node to bootstrap and initialize all entities that need to be initialized, configured and operationalized-including initialization of the node's agents that enable the node to auto-discover information, network resources, neighbour nodes, network elements (NEs) of interest and the network environment in general. Such auto-configuration operations may be based on environment sensing by the node and/or auto-discovery of network resources and bootstrap information upon the node's initiation on a network link, reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (e.g. fault/error/failure detection) in the network, changes in human operator policies and operational context, or workload changes. All such operations may get performed by the node in order to achieve certain objectives, such as re-configuration of some node's entities in response to context changes or to fulfil node's initialization requirements or other objectives (targets).

5.1.3.16.5 GANA Level-4 DE (the highest level)-called "Network" Level-DE

5.1.3.16.5.1 A General Characterization of GANA Level 4 DEs (called Network-Level DEs)

The Network Level DE relates to a DE for autonomic management and control of those aspects that cover network-wide views and the management andcontrol of the GANA Nodes which embed lower levels DEs (node level DE, function level DE and MEs at the protocol level DE). The Control-Loops at this level complement lower level control loops by operating on slower timescale (i.e. they are slower control-loops in contrast to lower level control-loops (the faster control-loops)). Therefore, they shall govern all the DEs of the GANA Nodes under their responsibility. As is done in the 3GPP Radio Access Network (RAN) with centralized Self Organization Network (C-SON) and the distributed SON (D-SON) in the NodeB/eNodeB. Refer to the ETSI White Paper No.16 [3] on how the Hybrid-SON Model shares the same principles with the GANA model.

What should be noted is that the Network-Level DEs should embed cognition functions (Observing, Comparing, Learning, Analysing functions) and a knowledge base (KB). Cognitions functions may be added in lower level-DEs if cognition properties are required for DEs in GANA Node. However, cognition requires a large amount of process and storage not always available on Network Elements (NEs) and they are recommended for slow control loop as in the Network level-DE. KB may be used by DEs to store rules about facts of their associated MEs and an inference engine that can reason about those facts and use rules to deduce new facts or highlight inconsistencies. An ontology model may be developed for each DEs and should be linked to a composite ontology which is used to describe the network. Those ontologies may be linked with others ontologies domains which may infer with the autonomic network such as the business profile used to govern the autonomic network.

5.1.3.16.5.2 Network-Level-Security Management-DE (NET_LEVEL_SEC_M_DE)

Decision Element (DE):

- Network-Level-Security Management-DE.

DEs, Protocols and Mechanisms as Managed-Entities (MEs):

- NODE_LEVEL_SEC_M_DEs (are policy-controlled by the network-level-DE); and also Security Protocols, Algorithms and Mechanisms that may need to be orchestrated and dynamically managed at the network level by the Network-Level-Security Management-DE rather.

Examples of protocols and Mechanisms that are mapped as MEs of the DE:

- Certificates/Passwords Algorithms, Hash Algorithms, Encryption Algorithms, Access Control Mechanisms, Trust Mechanisms, Denial of Service (DoS) Detection/Prevention algorithms/mechanisms, Signature based intrusion detection mechanisms, etc.

Description: the NET_LEVEL_SEC_M_DE is used to manage any security issues in the network. NET_LEVEL_SEC_M_DE may also be used to secure interactions between DEs and MEs within the same domain or to secure interaction with other MEs and DEs within different domain.

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.34] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration or orchestration of Security enforcement (or improvement) related Managed Entities (MEs) in network nodes (NEs) based on policing the node's own autonomic security management intelligence in form of policy control by the outer logically centralized "autonomic manager" component, and also based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (e.g. security threats) in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in network security as a whole or security for some nodes or traffic flows or other objectives (targets).

5.1.3.16.5.3 Network-Level-Fault Management-DE(NET_LEVEL_FM_DE)

Decision Element:

- Network-Level-Fault Management-DE.

DEs, Protocols and Mechanisms as Managed-Entities (MEs):

- NODE_LEVEL_FM_DEs (are policy-controlled by the network-level-DE); and also Fault Detection Mechanisms, Fault Isolation/Localization/Diagnosis Mechanisms, Fault Removal Mechanisms that may need to be orchestrated and dynamically managed at the network level by the Network-Level-Fault Management-DE rather.

Examples of protocols and Mechanisms that are mapped as MEs of the DE:

- NODE_LEVEL_FM_DEs (are policy-controlled by the network-level-DE); and also Fault Detection Mechanisms, Fault Isolation/Localization/Diagnosis Mechanisms, Fault Removal Mechanisms that may need to be orchestrated and dynamically managed at the network level by the Network-Level-Fault Management-DE rather.

Description: The NET_LEVEL_FM_DE is for autonomic fault management that require to be orchestrated at the network level by logically centralized algorithms of the NET_LEVEL_FM_DE-by employing appropriate Fault Detection Mechanisms, Fault Isolation/Localization/Diagnosis Mechanisms, and Fault Removal Mechanisms that enable to repair network services and/or functionality of a network node. The Network-Level-Fault Management-DE (NODE_LEVEL_FM_DE) interworks with the Network-Level-Resilience_&_Survivability-DE (NODE_LEVEL_R&S_DE) as described in [i.18] and in ETSI GS AFI 002 [2].

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.18], [i.47], [i.55] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration or orchestration of Fault-management (fault-detection, fault-diagnosis and fault-removal (repair))related Managed Entities (MEs) in network nodes (NEs) based on policing the node's own autonomic fault management intelligence in form of policy control by the outer logically centralized "autonomic manager" component, and also based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (e.g. fault/error/failure detection) in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as node-local and/or network-wide self-diagnosis and self-repair (self-healing) or other objectives (targets).

5.1.3.16.5.4 Network-Level-Resilience-And-Survivability-DE: NET_LEVEL_R&S_DE

Decision Element:

- Network-Level-Resilience and Survivability-DE.

DEs, Protocols and Mechanisms as Managed-Entities (MEs):

- NODE_LEVEL_R&S_DEs (are policy-controlled by the network-level-DE); and also Proactive and Reactive Resilience Mechanisms, Survivability Strategies and Algorithms, Restoration and Protection Mechanisms that may need to be orchestrated and dynamically managed at the network level by the Network-Level-Resilience and Survivability-DE rather.

Examples of protocols and Mechanisms that are mapped as MEs of the DE:

- NODE_LEVEL_R&S_DEs (are policy-controlled by the network-level-DE); and Network Resilience and Survivability mechanisms that may need to be orchestrated and dynamically managed at the network level by the Network-Level-Resilience and Survivability-DE rather.

Description: The NET_LEVEL_R&S_DE provides logic and algorithms to ensure the resilience and survivability of the network systems (network nodes/functions).

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.18], [i.47], [i.55] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration or orchestration of "Resilience &Survivability strategies" related Managed Entities (MEs) in network nodes (NEs) based on policing the node's own autonomic resilience and survivability management intelligence in form of policy control by the outer logically centralized "autonomic manager" component, and also based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (e.g. fault/error/failure detection) in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in node-local and/or network-wide resilience or in resilience and survivability for some traffic flows or other objectives (targets).

5.1.3.16.5.5 Network-Level-Data Plane and Forwarding Management-DE: NET_LEVEL_FWD_DE

Decision Element:

- Network-Level-Data Plane and Forwarding Management-DE.

DEs, Protocols and Mechanisms that are Managed-Entities (MEs) of the DE:

- Function-Level-Data Plane and Forwarding Management-DEs (are policy-controlled by the network-level-DE through their respective Node-Main-DEs); and also Layer-3 Forwarding Protocols and Mechanisms, Layer-2.5-Fowarding, Layer-2-Fowarding, Layer-3-Switching, Layer-2-Switchingmechanisms that may need to be orchestrated and dynamically managed at the network level by the Network-Level-Data Plane and Forwarding Management-DE rather.

Examples of protocols and Mechanisms that are mapped as MEs of the DE:

- Protocols such as IPv4/IPv6 Forwarding Engine;
- Multi-Protocol Label Switching (MPLS);
- etc.

Description: the NET_LEVEL_FWD_DE provides logic, algorithms to ensure optimized forwarding of traffic flows in the network and in respect of network operator policies and in order to optimize the network's traffic engineering and forwarding objectives and behaviours.

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.47], [i.46], [i.52] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration of Traffic Flow Forwarding and Switching related Managed Entities (MEs) in network nodes (NEs) based on policing the node's own autonomic forwarding and switching management intelligence in form of policy control by the outer logically centralized "autonomic manager" component, and also based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (e.g. link(s) congestion) in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in network performance and QoS for some traffic flows or other objectives (targets).

5.1.3.16.5.6 Network-Level-Routing Management-DE (NET_LEVEL_RM_DE)

Decision Element (DE):

- Network-Level-Routing Management-DE.

DEs, Protocols and Mechanisms as Managed-Entities (MEs) of the DE:

- Function-Level-Routing Management-DEs (are policy-controlled by the network-level-DE through their respective Node-Main-DEs); and also Routing protocol and mechanisms that may need to be orchestrated and dynamically managed at the network level by the Network-Level-Routing Management-DE rather.

Examples of protocols and Mechanisms that are mapped as MEs of the DE:

- OSPF;
- BGP;
- RIP;
- etc.

Description: the NET_LEVEL_RM_DE provides logic, algorithms to ensure optimized routing of packets and flows in the network and in respect of network operator policies and in order to optimize the network's traffic routing objectives and behaviours.

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.32], [i.33], [i.36], [2], [3], [i.46], [i.47], [i.52] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration routing related Managed Entities (MEs)in network nodes (NEs) based on policing the node's own autonomic routing management intelligence in form of policy control by the outer logically centralized "autonomic manager" component, and also based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (e.g. link(s) congestion, link and node failures) in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in network performance and QoS for some traffic flows or other objectives (targets).

5.1.3.16.5.7 Network-Level-Quality of Service Management DE (NET_LEVEL_QoS_M_DE)

Decision Element:

- Network-Level-Quality of Service Management-DE.

NOTE 1: This DE may be implemented as Network-Level-Quality of Service and Experience Management-DE(NET_LEVEL_QoS_&_QoE_M_DE).

DEs, Protocols and Mechanisms as Managed-Entities (MEs) of the DE:

- Function-Level-Quality of Service Management-DEs (are policy-controlled by the network-level-DE through their respective Node-Main-DEs); and
- also QoS Protocols and Mechanisms that may need to be orchestrated and dynamically managed at the network level by the Network-Level-Quality of Service Management-DE rather.

Examples of protocols and Mechanisms that are mapped as MEs of the DE:

- Packet classifier, Packet Marker, Queue Management, Queue Scheduler, RSVP (Resource ReSerVation Protocol), Policy Control and Charging Rules Function mechanism, etc.

Description: the NET_LEVEL_QoS_M_DE provides logic and algorithms to ensure QoS of services and also improve QoE (Quality of Experience) for services.

NOTE 2: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.44], [i.45], [i.47], [i.49], [i.51] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration or orchestration of Quality of Service enforcement (or improvement) related Managed Entities (MEs) in network nodes (NEs) based on policing the node's own autonomic QoS management intelligence in form of policy control by the outer logically centralized "autonomic manager" component, and also based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (e.g. QoS degradation for some traffic flows) in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in network performance and QoS for some traffic flows or other objectives (targets).

5.1.3.16.5.8 Network-Level-Mobility Management-DE (NET_LEVEL_MOM_DE)

Decision Element:

- Network-Level-Mobility Management-DE.

DEs, Protocols and Mechanisms as Managed-Entities (MEs):

- Function-Level-Mobility Management-DEs (are policy-controlled by the network-level-DE through their respective Node-Main-DEs); and also Mobility Management Protocols and Mechanisms that may need to be orchestrated and dynamically managed at the network level by the Network-Level-Mobility Management-DE rather.

Examples of protocols and Mechanisms that are mapped as MEs of the DE:

- Mobility Support in Internet Protocol Version 6 (MIPv6), Datagram Congestion Control Protocol, Mobile Stream Control Transmission Protocol, Site Multi-homing by IPv6 Intermediation, Proxy-Mobile-IP, Mobility-Management User-Equipment Managed-Entity, Measurement-Report-Function Managed-Entity, Candidate-Access-Router-Discovery mechanism, Fast Handover Scheme, Policy Control and Charging Rules Function mechanism, etc.

Description: The NET_LEVEL_MOM_DE provides logic and algorithms to ensure handover between access networks, network element with service continuity.

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.47], [i.48], [i.53] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration or orchestration of Mobility enablement (or adaptation) related Managed Entities (MEs) in network nodes (NEs) based on policing the node's own autonomic Mobility management intelligence in form of policy control by the outer logically centralized "autonomic manager" component, and also based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (e.g. QoS degradation for some traffic flows, network connectivity errors and failures) in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in network performance and QoS for some traffic flows or other objectives (targets).

5.1.3.16.5.9 Network-Level-Monitoring Management-DE (NET_LEVEL_MON_DE)

Decision Element:

- Network-Level-Monitoring Management-DE.

DEs, Protocols and Mechanisms as Managed-Entities (MEs):

- Function-Level-Monitoring Management-DEs (are policy-controlled by the network-level-DE through their respective Node-Main-DEs); and also Monitoring Protocols, Mechanisms and Tools (e.g. for traffic monitoring, context inference and derivation, etc.) that may need to be orchestrated and dynamically managed at the network level by the Network-Level-Monitoring Management-DE rather.

Examples of protocols and Mechanisms that are mapped as MEs of the DE:

- IPFIX data collection and dissemination mechanisms, SNMP data collection and dissemination mechanisms, NETFLOW data collection and dissemination mechanisms, Protocol Analysers, Packet Trace creation and dissemination mechanisms. Effective and Available Bandwidth Estimation mechanisms, IPv6 Hop-by-Hop Options for network-intrinsic monitoring, etc.

Description: The NET_LEVEL_MON_DE provides logic and algorithms to retrieve monitoring data from various sources, to derive context information, to dynamically orchestrate and regulate monitoring mechanisms and tools of the network and the rate (e.g. sampling rate) at which they create monitoring data and disseminate the data to entities that need the monitoring data (e.g. DEs). The granularity and formats in which monitoring data and/or knowledge presentation is created by monitoring mechanisms and tools and disseminated to data collectors and to entities that directly consume the monitoring data or knowledge are all determined by the Monitoring DEs at the Function-Level and Network-Level collaboratively. The Network-Level Monitoring-DE policies the behaviours of Function-Level-Monitoring Management-DEs that dynamically orchestrate and autonomically manage Monitoring Protocols, Mechanisms and Tools of their respective GANA nodes. In orchestrating and managing the mechanisms and tools for disseminating monitoring data, context and knowledge to other DEs in the Knowledge Plane, the Network-Level-Monitoring Management-DE is supposed to orchestrate and dynamically manage and control the kinds of mechanisms and tools for the dissemination of monitoring data, context or knowledge that complement the ONIX and (amc)-MBTS as information/data/knowledge disseminators. The NET_LEVEL_MON_DE's logic should include an algorithm to retrieve information (subscribe for information) and process the information indicative about the operations of the Knowledge Plane DEs in order to infer and adaptively enforce the monitoring behaviour and monitoring-data flow (or knowledge flow) from NEs, Data Collectors and Probes as may be needed by Knowledge Plane DEs during their operations.

NOTE 1: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.47], [i.50], [i.51], [i.54], [i.55], [i.56] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration or orchestration of Monitoring related Managed Entities (MEs) in network nodes (NEs) based on policing the node's own autonomic Monitoring management intelligence in form of policy control by the outer logically centralized "autonomic manager" component, and also based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (manifestations of faults, errors, failures, security attacks)in the network, changes in human operator policies and operational context, monitoring-data demands from other types of entities (e.g. other autonomic manager components), or workload changes, in order to achieve certain objectives such as the gathering and optimal dissemination of various types of monitoring data to entities that need to act upon the monitoring data in order to effect changes in network behaviours or other objectives (targets).

NOTE 2: A Network-Level-Monitoring Management-DE (NET_LEVEL_MON_DE) could be implemented as an integral part of a Probe (e.g. a Passive Monitoring Probe, Active Monitoring Probe or one that combines active probing and passive probing aspects) if the Probe is external to an NE and has a scope of being a Network-Wide Monitoring Probe.

5.1.3.16.5.10 Network Level End-to-End “end-user oriented” Service and Applications Management (NET_LEVEL_E2E_Service_M)

Decision Element:

- Network Level End-to-End “end-user oriented” Service and Applications Management.

DEs, Protocols and Mechanisms as Managed-Entities (MEs):

- Function-Level-Service Management-DEs.
- TCP/IP Application Layer or OSI Application Layer entities and service chain live cycle management mechanisms.

Examples of protocols and Mechanisms that are mapped as MEs of the DE:

- Service function, application function infrastructure live cycle mechanisms.

Description: the NET_LEVEL_E2E_Service_M_DE provides logic, algorithms to manage end 2 end services delivered to end users. May be used to identify the chain of service functions needed to provide service to the end user. Retrieve information to identify the quality of experience of service delivered to end user.

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [2] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration of TCP/IP or OSI Application-Layer Service Components related Managed Entities (MEs)hosted in network nodes (NEs) such as Servers based on policing the node's or server's own autonomic service management intelligence in form of policy control by the outer logically centralized "autonomic manager" component, and also based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (e.g. service or server overload situation, service faults/errors/failures) in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as improvement in service performance (for the service/application hosted by a node or a group of nodes or servers) and QoS for some traffic flows or other objectives (targets).

5.1.3.16.5.11 Network-Level-AutoConfiguration and_AutoDiscovery-DE (NET_LEVEL_AC_DE)

Decision Element:

- Network-Level-Auto-Configuration_and_Auto-Discovery -DE (NET_LEVEL_AC_DE).

DEs, Protocols and Mechanisms as Managed-Entities (MEs):

- NODE-Level-AutoConfiguration_and_AutoDiscovery-DEs.
- Network Discovery Protocols/Mechanisms and Network Discovery Mechanisms.

Examples of protocols and Mechanisms that are mapped as MEs:

- IPv6 Neighbour Discovery Protocol (NDP), IPv6 Secure Neighbour Discovery Protocol (SEND), Network Topology Discovery protocols and mechanisms, etc.

Description: NETWORK_LEVEL_AC_DE provides logic, algorithm(s) to ensure plug and play mechanisms of GANA Network. Provide auto discovery, bootstrapping mechanisms and auto configuration mechanisms. The NETWORK_LEVEL_AC_DE may self-adapt the GANA Network according to GANA profile defined by an administrator domain and orchestrate the different DEs within the GANA network and in collaboration with other administrative domains.

NOTE: For some insights on how to implement such a DE (e.g. how to design the DE logic and algorithms, etc.), the following sources may contain further insights on how to implement the DE: [i.36], [i.38], [i.46], [i.47], [2] and other various sources such as results from research projects on autonomic networking and autonomic network management that designed and validated algorithms for "autonomic manager" components that are aimed at performing automated configuration and dynamic (adaptive) (re)-configuration or orchestration of a node's (NE's) auto-discovery and auto-configuration enabling Managed Entities (MEs)-based on policing the node's own autonomic auto-discovery and auto-configuration management intelligence in form of policy control and configuration data supply by the outer logically centralized "autonomic manager" component, and also based on reacting to feedback on network and ME(s) state (continuously observed/monitored) and experienced challenges (e.g. network connectivity problems and network services failures that impact various networking functions and their associated management and control operations) in the network, changes in human operator policies and operational context, or workload changes, in order to achieve certain objectives such as a re-configuration of network nodes (NEs) with new profiles and policies and objectives or other objectives (targets). Such operations by a higher level autonomic manager component is meant to complement a node's capability to bootstrap and initialize all its entities that need to be initialized, configured and operationalized-including initialization of the node's agents that enable the node to auto-discover information, network resources, any outer management and control entities that may manage and control the node (by business policy), neighbour nodes, network elements (NEs) of interest and the network environment in general, as described in the case of an node's Node-Level Auto-Configuration_and_Auto-Discovery-DE.

5.1.4 DEs Coordination Function for ensuring Stability in interacting Control-Loops, and the role the Auto-Configuration DE (Auto-Configuration and Auto-Discovery DE) can play in Coordination of DEs

This clause provides guidance on how implementers of DEs can address the need for DEs Coordination Function for ensuring Stability in interacting Control-Loops. The broad problem of addressing stability for autonomic components (e.g. DEs) needs to be addressed by two types of complementary techniques as described in ETSI GS AFI 002 [2], namely "design for stability techniques" and "run-time stability enforcement techniques". Approaches for addressing potential stability related problems on DE interactions whilst still at the design time, simulation and validation time for GANA DEs, i.e. "design for stability techniques", include the approach prescribed in [i.59] and [i.60]. Complementarily the following recommendations are useful to consider towards the goal of addressing the problem of stability of interacting control-loops (autonomic functions/components) at run-time, and should be complemented with approaches prescribed in ETSI GS AFI 002 [2], ETSI White Paper No.16 [3], [i.23] and in [i.30] and those prescribed in the case of Hybrid SON (Self-Organizing Network) model applied to 3GPP RAN as well as other approaches in literature on the subject of coordination of autonomic components/managers:

- The GANA Node's Auto-Config DE should be used to implement the DE Coordination Function for local coordination and arbitration based on the Framework for Stability in GANA (Synchronization of Actions and Policies in GANA) described in ETSI GS AFI 002 [2], [i.23], ETSI White Paper No.16 [3] and [i.23].

- The Network Level Auto-Config DE should be used to implement the DE Coordination Function for KP DEs coordination and arbitration based on the Framework for Stability in GANA (Synchronization of Actions and Policies in GANA as described in [i.23]).

- An action synchronization initiated by a DE in the GANA node whose target is to synchronize with a DE(s) in the Knowledge Plane or in another GANA Node shall be relayed through the Node-Main-DE (particularly the Auto-Configuration DE and Security-Management-DE) if such synchronizations should be known by Node-Main-DE, based on the Framework for Stability in GANA (Synchronization of Actions and Policies in GANA).

- A Knowledge Plane DE can perform Policing of a "mirror-DE" that address the same management andcontrol domain of reasoning in the GANA node, e.g. a Network-Level Security-Management-DE is "mirrored" by Security-Management-DE that operates in a GANA Node; Network-Level Monitoring-DE is "mirrored" by Function-Level Monitoring-DE the operates in a GANA node. The Policing of a lower level (in a node) by a KP DE shall use a special protocol for that and the same protocol may be used for synchronization of actions by a DE intending to synchronize a tentative action to get approval for execution by the KP DE(s)-based on the Framework for Stability in GANA (Synchronization of Actions and Policies in GANA). If multiple DEs and management andcontrol aspects are involved in the synchronization process or Policing process, a Coordination Function may be involved in the process.

- The MBTS may implement the Action Synchronization Protocol employed between the lower level and KP level DEs, else such a protocol may be implemented out of band between the lower and KP DEs. The MBTS may implement the DE Policing Protocol employed a KP level DE(s) to policy a DE(s) in a target GANA node(s) else such a protocol may be implemented out of band between the KP DEs and lower level DEs.

NOTE: The subject of addressing Stability at run-time in autonomics and control-loops interactions is covered in more detail in ETSI GS AFI 002 [2] and in other literature that is widely available on this subject, such as [i.23], [i.30] and [i.31].

5.1.5 Model Based Translation Service (MBTS)

NOTE 1: The description and characterization of the MBTS provided in this clause is an extract from the corresponding MBTS related clauses in ETSI GS AFI 002 [2]. This means that description and characterization of the MBTS features provided in this clause are to be complemented by the MBTS related details provided in ETSI GS AFI 002 [2].

Model Based Translation Service (MBTS) is ***an optional software functionality*** that should be implemented in form of software libraries for the purpose of translating information, operations and commands andresponses from GANA Knowledge Plane's Data Model to a management/control protocol technology-specific or proprietary vendor-specific data models used for communications with an NE(s). Therefore, the GANA Knowledge Plane requires a common unified data model that is agnostic to management and control protocol technologies (such as SNMP, OpenFlow, NETCONF, TR-609, CMIP, COPS, XMPP, etc.) semantic (data) models and vendor-specific data models, for use by Network level DEs to process information from NEs translated to the Knowledge Plane data model and relayed to the DEs by the MBTS. The MBTS may be used to manage different Data Models used by NEs, GANA Node or Human Operator's tools for automated network and service management (e.g. OSS, BSS) as described in Annex C.

NOTE 2: As discussed earlier, not employing an MBTS in communications between Knowledge Plane DEs and the NEs in the network infrastructure makes it invertible that the DEs talk the languages of the various management and control protocols directly with the NEs (which may make the DEs designs and their software programming more complex without a commonly shared MBTS in place).

MBTS "maps" Ontologies or Information Models used to describe node-local behaviour which needs to be shared with the Knowledge Plane. The Translation Services, knowing the source and target representations, translate node-local information received to the appropriate representation for use by a specific cognitive process or the so-called "K-application(s)" i.e. the domains of the Knowledge Plane. Network-Level-DEs form the "domains" of the Knowledge Plane, executing "K-application(s)" specific to a DE.

A Model-Based Translation Service (MBTS) that translates commands (/responses) from (/to) the Network-Level-DEs into (/from) a target command syntax and semantic formulation acceptable to the type of a target node/device. MBTS shall translate data models that are specific to management and control protocols and/or vendor specific proprietary management and control data model to a common data model that can be designed specifically for the GANA Knowledge Plane DEs. Therefore, the value the MBTS brings to network programmability is that it enables KP DEs designers to design DEs to talk a language that is agnostic to vendor specific management protocols and technology specific management protocols that can be used to manage NEs.

The Governance MBTS (G-MBTS) maybe used to translate Governance Profile (as input that conveys network Operator's inputs to the network) from OSS/BSS or EM/NM to the GANA Profile used by Network Level DEs. More details on this aspect are found in later clauses related to GANA Operations Procedures for the Human Network Operator.

The Federation MBTS (F-MBTS) may be used to federate information and knowledge from different Knowledge Planes (KP) in order to share knowledge between ONIX instances (e.g. Knowledge Planes and ONIX instances of different network domain access, core, fixed, mobile, provider, OTT).

When employed by Knowledge Plane DEs, the Management MBTS (AMC-MBTS) should translate information of GANA Nodes (NEs) and store the knowledge into the ONIX while at the same time streaming the knowledge to the Knowledge Plane DEs for their real-time operation on the knowledge. The MBTS should implement a RAT (Representation, Acquisition and Translation) function for derivation of Knowledge from information/data supplied by NEs under the scope of the MBTS. As described in the ETSI White Paper No.16 [3] the knowledge derived by the MBTS should be streamed to KP DEs for them to act upon the knowledge in real-time while at the same time the knowledge should also be stored by the MBTS into the ONIX. The MBTS should employ a Composite Ontology (or Ontologies) in its knowledge generation and dissemination process. For information on how the RAT function for the MBTS may be implemented refer to ETSI GS AFI 002 [2] on this subject (e.g. clause 9.13.6 of ETSI GS AFI 002 [2] on "Possible Approaches to Deriving Knowledge for the Knowledge Plane" as well as Annex D of ETSI GS AFI 002 [2]).

NOTE 3: In case an NE implements a RAT Layer, then the NE should export knowledge generated by its RAT to the AMC MBTS for further relaying to the KP DEs, else the NE can push the Knowledge directly to KP DEs if no MBTS is being used between the KP DEs and the NEs.

NOTE 4: There are other sources of resourceful concepts and insights in literature that may be considered in implementing MBTS capabilities such as context processing and brokerage, and metadata management for the various data models the MBTS should employ in translating between the GANA Knowledge Plane DEs domain and the network infrastructure (NEs) domain. For example, concepts and insights from [i.77] that relate to context event processing and management, context brokering and metadata management for data models can be exploited in the design and implementation of the MBTS.

The ONIX and MBTS may also be used to federate information of different Knowledge Planes (KPs).

5.1.6 Overlay Network Information eXchange (ONIX) System of Information Servers

5.1.6.1 A General Characterization of the ONIX System of Federated Information Servers

NOTE 1: The description and characterization of ONIX provided in this clause is an extract from the corresponding ONIX related clauses in ETSI GS AFI 002 [2]. This means that description and characterization of ONIX features provided in this clause are to be complemented by the ONIX related details provided in ETSI GS AFI 002 [2].

ONIX (Overlay Network for Information eXchange) (distributed scalable overlay system of federated information servers). The ONIX is useful for enabling auto-discovery of information/resources of an autonomic network via "publish/subscribe/query andfind" protocols. DEs can make use of ONIX to discover information and entities (e.g. other DEs) in the network to enhance their decision making capability.

NOTE 2: As pointed out earlier, the ONIX itself does not implement network management and control logic.

NOTE 3: ONIX" is not the same as "onix" system in [i.69].

NOTE 4: An example of an approach to implementing ONIX is given in Annex D of the present document.

NOTE 5: There is on-going work in ETSI on elaborations of the Implementation Guide for GANA and the resultant work is expected to be published in ETSI TR 103 195-3 [i.71], and the work is expected to provide more guidance on how to implement the GANA Functional Blocks (including the ONIX).

The Overlay Network for Information eXchange system (ONIX) shall support information publish &subscribe paradigm and information storage.

The ONIX is also used as the bootstrapping network function and stores the capabilities of the network. DEs subscribe for information they need during their operations (e.g. a GANA Profile) that may have been intended by the human operator to be retrieved by KP DEs from ONIX) and publish their Reports into the ONIX for access by the Network Operator's Governance Tools so that the human operator may view and e the Reports generated by DEs as analysable traces during their operations. The knowledge stored in the distributed KB of DEs or other databases may need to be stored in the ONIX when the ONIX is a preferred dissemination method, otherwise a DE exchanges knowledge and information that is required to be consumed by other DEs directly without latency/overhead by direct DE-to-DE interactions. The ONIX should fulfil the following characteristics:

1) Information/Knowledge acquisition and sharing mechanisms (which include (secure) publish/subscribe, (secure) query/search, data mining and find mechanisms that shall be supported by information/knowledge storage Repositories). This includes services provided by a "distributed Overlay Network Information eXchange system" (ONIX) required in an autonomic network.

2) ONIX system is a peer-to-peer based overlay of Repositories that support publish/subscribe, query and find type of services for information/knowledge such as Capabilities of network elements, Network Configuration Profiles (GANA Profiles), Goals and Policies of the autonomic network, pointers to resources and Data, and other types of information/knowledge.

3) ONIX is a system of Information Sharing Repositories that may be considered as forming a distributed system. Any network system (Database, User equipment, even a router) designed to share information as if it were an Information Sharing system supporting ONIX protocols may join the overlay network (ONIX) and use the protocols of the overlay network to share information and to answer information queries in collaboration with other members of the overlay network (ONIX).

4) ONIX has no decision-making logic. An information query only need to be sent to any of the servers (even if the info is actually stored on another server) and the servers will communicate with each other via an ONIX protocol such as in peer to peer distributed hash table (e.g. Chord, CAN) to find the server having the information, and respond to the query. ONIX Information Repositories can also be used to store and share YANG models as described in Annex D.

5) Knowledge Field: Knowledge is an essential element of cognitive and autonomic control-loops and should be stored in a Knowledge Base (KB) internal to a DE or a KB shared by multiple DEs. The meaning of the word "*Field*" in "Knowledge Field" has to be interpreted as derived from the discipline of Physics. For instance, the knowledge field could look like a sort of gravitational field emitted by each autonomic component i.e. a DE (e.g. representing the current amount of available resources or capabilities) such that by following the local shape of the fields components could perform decision-making and actuation processes (e.g. concerning the discovery of available resources or capabilities and what to do as a result). In other words, autonomic components i.e. DEs in the GANA Model could sense the knowledge field and act accordingly to specified behavioural patterns or plans. Knowledge field models could potentially be implemented, as an overlay network such as the ONIX system, on any middleware providing basic support for data storing, communication and event-notification. What is required is to provide simple storage mechanisms (to store field values), communication protocols and primitives (to efficiently propagate field values among the peers), and basic secure and resilient publish-subscribe mechanisms. The ONIX distributed system is meant to facilitate these kinds of services. The knowledge field could be stored, in a decentralized way, either on DEs in GANA Node, or on distributed specialized nodes (information servers) in charge of generating, handling and distributing the knowledge across the system (ONIX). Data mining (e.g. Gossiping mechanisms) can help achieve an efficient way for distributing and aggregating the information. Reasoning techniques and data correlation features might also be necessary in effective exploitation of the "knowledge field" such as the rules to manage facts of MEs used by DEs. These rules within KB should be managed through a configuration profile that may be included as a part in the global GANA Network Profile.

6) Bootstrapping: ONIX should also be used to store bootstrapping data required by the autonomic network.

7) Fulfils the need for an Information/Knowledge exchange/sharing distributed Overlay Network that is a distributed scalable overlay system of federated information servers-useful for enabling auto-discovery of information/resources of an autonomic network via "publish/subscribe/query and find" protocols of the system (ONIX).

5.1.6.2 Categories of Information that could be stored and shared through ONIX

Implementers of ONIX should consider the following criteria in designing, dimensioning and deploying the ONIX-pertaining to the information that could be stored and shared through ONIX:

1) The type of raw data and information to be stored into ONIX and exchanged/shared with entities which subscribe to receive such information.

2) The type of knowledge, for example numeric knowledge and semantic knowledge that could be stored and shared through ONIX.

3) The size of the information/knowledge:

   - Small sized information/knowledge may be considered the key - value type of information/knowledge. Medium sized information is any type of information with size below 1 MB. Any type of information/knowledge exceeding 1 MB could be considered as large sized information/knowledge.

4) The frequency at which the stored information/knowledge changes over time:

   - Slow changing information is any type of information that changes at intervals longer than 1 hour. Medium changing information is any type of information changing at intervals bigger than 30 seconds and smaller than 1 hour. Fast changing information is any type of information changing at intervals less than 30 seconds.

5) The maximum delay between the data collection into the ONIX or consolidation of data/information within ONIX servers, and information dissemination by ONIX for information consumption by the interested information consumer recipients (directly subscribed or subscribed "on-behalf" by another entity e.g. a DE), i.e. whether the information can be considered time critical or not.

6) The accuracy/ integrity of the information/knowledge in order to be effectively used in the decision making processes of entities such as DEs.

7) The scope of the information/knowledge that define the context in which the information is considered valid.

Based on the above criteria, a general information is defined as any type of information that is small or medium sized, slow or medium changing and not time critical. Examples of such types of information are: capabilities description, policies, network profiles, network inventories, some monitoring data, etc. For general type of information, a general information exchange system (i.e. the ONIX) shall be in place. For special type of information like large sized information or fast changing information or time critical information other specialized systems for information exchange shall be in place within the autonomic network and its associated management and control architecture.

In the following, a distributed system for general type information exchange is described. The information exchange system is a required component for the GANA.

The key requirements and goals that shall be considered when designing such a system (ONIX) are:

- Scalability: The system should operate efficiently in large networks in terms of network nodes, supported services, autonomic entities within a node, number of interconnection links, transported data or control traffic, number of events, storage resources, etc. In addition, the system should address the needs of different administrative domains interconnected together (i.e. *Interaction between Information/Knowledge Sharing Systems(ONIXs') belonging to different administrative domains*). It could be claimed that the system shall be fully distributed, for example, by relying on distributed hash tables (DHT).

- Reliability: The system should be robust against possible failures, e.g. in the network, at the node level, at software, etc. The system shall replicate stored data in order to overcome multiple concurrent failures. In addition, any functionality assigned to a (primary) node providing services to the network nodes should also be replicated to back-up (secondary) node. In addition, the system should be able to remain operational in dynamic environments, e.g. after link or node failures, node mobility, etc.

- Bootstrapping and Topology Formulation Capabilities: The system shall be able to initiate itself without any pre-configured data. System functionality may be further extended or the system performance may be improved when basic administration parameters are set.

- Efficient Data Handling: The system shall be able to manage and update any information stored within it. Data replication shall be easily achieved for any data types stored. Finally, data should have a predefined scope in order to assess the validity of data in different network segments.
- Expressive Queries: The system should support complex (i.e. multi-attribute) queries. Partial queries should be supported (queries that contain only a subset of the attributes originally advertised by resources, considering the other attributes as wildcards). Each query might also have a scope (global, local, n-hops away, etc.).
- Efficient Notification: The system should provide a powerful subscribe mechanism for receiving information of interest. The subscription should support multi-attribute filtering capabilities.
- Standardized ONIX Interfaces (See the related ONIX Rfps defined in the present document that can be implemented as ONIX Interfaces): The system should provide well-designed Interfaces to allow efficient data input and retrieval by/from the autonomic nodes. The control messages of the system should not significantly contribute to network congestion. In addition, it shall provide well-designed communication protocols and primitives to be used by servers that are part of the overlay network, to build and maintain the overlay.
- Security: The system should support capabilities for avoiding or minimizing data poisoning. Authentication and authorization functionality may be needed in various scenarios. The subscription should provide security and integrity of information exchange according to the requester and the requested information.
- Extensibility: The system should be extendable to allow the inclusion of future functionality.
- Minimized Operational Overhead: Since the goal of GANA is to reduce the complexity in network management operations by humans, ONIX System shall be optimized for simplicity of installation and maintenance.
- Knowledge management: A Knowledge Base (KB) is also used to identify the rules used to process events. The entities that generate the rules and store them in the Knowledge Base (e.g. ONIX Information repositories/servers or some shared KB in an NE) shall ensure that there are no conflicting rules within and for the network as a whole in order to avoid conflicted decisions by DEs when they use such information. Some specialized software and algorithms for knowledge management on a Knowledge Base (KB) or by a KB are required. Therefore, information that may be stored in ONIX include e.g. rules to process certain types of data such as events. More details on this subject are provided later in the clause on Knowledge Management.

5.1.6.3 Information Services that should be provided by ONIX

The following services shall be provided by the ONIX:

- Storing and Retrieving Information:
  - Push and Pull models supported.
  - Different classes of information.
  - Add, remove or replace operation supported.
  - Information is described using rules to process data such as those defined in RDF or other mark-up language (e.g. XML, YANG), and the information shall then be shared using an ONIX external protocol as described in Annex D on "The ONIX System and possible ways to implement ONIX". Such ONIX's external protocols could include **json.ld [**i.58**]**).
  - Data mining mechanisms supported on ONIX's Information Servers.
- Query for Information: by supporting a query language capable of expressing complex queries (partial queries, scoped queries, etc.) (e.g. XPath).
- Disseminating the information: Upon request, periodically or event triggered:
  - Normal Subscription.
  - On-behalf Subscription.
  - Publish and Disseminate.

- Security:
    - Authentication.
    - Authorization.
    - Trust.
    - Confidentiality.
    - Integrity.
    - Non-repudiation.
    - Privacy.
    - Tracking of activities taken and originators of each input to the system for accountability and auditing.
- Reliability, fault-tolerance and accessibility.

5.1.6.4 Information Subscription Mechanism

Any authorized entity may query the ONIX system for data or may subscribe to ONIX to retrieve information of interest when the information becomes available in ONIX. A subscription is defined by the requestor, the information requested and the trigger to gather and send the information. For defining what information to receive, a mark-up language can be used. Regarding the triggers for ONIX to send (push, schedule) the information, two types of subscriptions are defined:

1) Timed triggered subscriptions. An entity, e.g. a network element or a DE, should subscribe to receive the information at a specified date and time (which could be immediately) or periodically at certain intervals. In the last case the interested entity should choose to receive the information once, for "n" number of times, or as long it is subscribed.

2) Event triggered subscriptions. An entity, e.g. a network element or a DE, should subscribe to receive the information whenever a certain event occurs. Such an event could be, for example, the fact that a node with a specific set of capabilities joins the network, the fact that some resource's description is updated or a parameter's value exceeds a predefined threshold. Also for this type of subscription the interested entity should choose to receive the information once, "n" number of times, or whenever data gets submitted/updated as long as the information consumer entity is subscribed to receive the information.

5.1.6.5 Bootstrapping information

ONIX should also be used to store bootstrapping data for the autonomic network. Bootstrapping is a self-sustaining process employed by a behaviour-exhibiting entity (e.g. a network element or node) that progresses without (or with limited) external help. In an autonomic networking environment, bootstrapping requires the provision of initial minimal configuration data/information required to newly joined nodes or newly created networks. This should be designed (i.e. facilitated) in such a way as to allow for the successful initialization of the networking functions/mechanisms without any pre-configured data. Network-wide parameter estimation, data forwarding mechanisms, routing protocols, and security policies are some of the processes that need to be activated by the autonomic nodes without human support or pre-configured software. Decentralized bootstrapping, i.e. process instantiation without the need for special-purpose nodes, is also crucial in some autonomic networks since it facilitates the design of functionalities and mechanisms in ad-hoc and resource constrained environments.

Bootstrapping may be realized in multiple phases. Initially, nodes in proximity (or neighbouring nodes) may establish point-to-point communication channels with each other. In a later stage, neighbouring nodes collaborate in order to establish end-to-end communication paths, e.g. by enabling proactive/reactive routing functionality. Service provisioning and fulfilment of any network-wide objectives are realized in later stages, e.g. by provisioning distributed repositories for data, information and knowledge sharing.

Bootstrapping shall be secured within an administrative domain (to avoid bootstrapping unauthorized node/DE/ME).

5.1.6.6 On Federation of ONIX systems, and other characteristics of ONIX

The ONIX system, as a distributed system, may be scoped, owned and policed (i.e. governed) by a particular administrative domain. This implies that different ONIX systems (ONIXs') may exist and so **inter-domain Information/Knowledge sharing** may be required.

NOTE 1: The clauses in the present document that discuss federation of AMC also provide insights on this subject of inter-domain information/knowledge sharing.

NOTE 2: While an aircraft is equipped with the so-called *Black Box* that records historical data/information such as flight data, the ONIX could be viewed as a system that can play a similar kind of role, i.e. serving as some kind of "Black Box" of the network and its management and control operations (by being used to store knowledge on the historical operations of the autonomic network over a certain time), apart from fulfilling its fundamental functions and services.

5.1.7 Knowledge Management on Knowledge Bases (KBs) and the role of ONIX, DEs, MBTS, NEs and Automated Network Management Tools in Knowledge Management

A Knowledge Base (KB) is used for storing knowledge and its evolution over time during the operation of an autonomic function and the Knowledge Plane as a whole. Multiple types of KBs are expected, namely KBs in the ONIX (as Information Servers), KBs in GANA Nodes (shared KB for the node-internal DEs) and micro KBs in DEs as well (internal KBs). Rules to process data amount to knowledge that can be created by tools in the automated management tool chain and also dynamically created by DEs and stored in local KBs, shared KBs in GANA nodes (NEs) and in the ONIX Repositories as KBs, and such rules should evolve with the improvement of cognitive capabilities of DEs. As discussed earlier concerning MBTS (and also in ETSI GS AFI 002 [2]), the MBTS shall implement a RAT (Representation, Acquisition and Translation) layer (discussed in ETSI GS AFI 002 [2]) that generates knowledge from the monitoring data and information it receives from NEs and streams the knowledge to the Knowledge Plane DEs while at the same time storing the knowledge as history into the ONIX. As discussed in ETSI GS AFI 002 [2] and NE may implement a RAT Layer and in such a case it can disseminate knowledge to the Knowledge Plane DEs directly or through the MBTS (if deployed, otherwise the RAT layer would need to be implemented on NEs and/or by the Knowledge Plane DEs).

A KB is also used to identify the rules used to process events. The entities that generate the rules and store them in the Knowledge Base (e.g. ONIX Information repositories/servers or some shared KB in an NE) shall ensure that there are no conflicting rules within and for the network as a whole in order to avoid conflicted decisions by DEs when they use such information. Some specialized software and algorithms for knowledge management on a Knowledge Base (KB) or by a KB are required. Regarding rules to process events, there are events that may be subjected to some rules for processing them and such rules can be defined by the human operator using some tools for automated network management. However, there are rules that should be generated dynamically by Knowledge Plane DEs regarding the processing of certain types of events. For both types of rules and the associated event types there should not be conflicts in how they get applied. Knowledge Plane DEs work with rules generated by humans through some tools that may insert the rules into the GANA Profile, and also work with the rules dynamically generated by Knowledge Plane DEs, in their autonomic management and control of the network and its services. Conflicts that may be managed by a human operator through specialized software (or tool) for knowledge management and policy and goals conflicts resolution shall be applied before the conflict-free rules are inserted into the GANA Profile (see clause 6.6 on "GANA Operations Procedures for the Human Network Operator" on how a GANA Profile evolves from a skeleton to a complete GANA Profile that is finally applied to the network infrastructure). On the other hand (complementarily) KP DEs collaborate to resolve conflicts as well through a Coordinator DE such as the Auto-Configuration-DE, and only conflicts that the DEs cannot resolve through the coordinator DE shall require the human operator to intervene through some tools for automated network management by taking actions based on recommendations the KP DEs shall have to produce in such a case to help the human operator in prioritizing certain rules, goals or policies among those conflicting with each other.

6 Reference Points (Rfps) in the GANA Reference Model

6.1 An Overview of the Types of Reference Points in the GANA Model

In addition to defining the abovementioned FBs, the GANA Reference Model also specifies Reference Points (Rfps) that shall be used between FBs of the GANA reference model. Rfps between DEs may be either horizontal Rfps (HRP) (i.e. between DEs belonging to same levels NeDe, FuDe, NoDe) or vertical Rfps VRP (i.e. between DEs belonging to a different level (NeMe, NoMe, DeMe) depicted in Figure 2. HRPs "in-network" DE-to-DE communication and interaction of associated distributed Control-Loops may simply be viewed as an enhancement to the traditional distributed control plane with distributed control that enable NEs to (re)-negotiate some configuration aspects and to adaptively control the behaviour of MEs i.e. to collaboratively self-optimize end to end service delivered to end users. This clause defines the Rfps between GANA Functional Blocks (FBs) as well as their interactions with other types of Functional Blocks in the network and its management and control architectures, by adopting the Reference Points defined in clause 13 of ETSI GS AFI 002 [2] and extending them with additional Reference Points (Rfps) necessary as described further in this clause 6 of the present document.

In particular, in order to introduce autonomicity in any network architecture, an instantiation needs to be carried out, concerning the FBs and Rfps from the GANA Reference Model onto target architecture, e.g. the wireless mesh network architecture [i.6], the 3GPP network architecture [i.4], BroadBand Forum (BBF) architectures [i.28] or the Future Internet architecture [i.2] and [i.7]. Instantiation of GANA onto target architecture shall identify these Rfps with their operations and primitives as each instantiation need to be defined according to the providers and industries requirements and uses cases [1].

6.2 The basic concepts associated with Reference points in GANA

The GANA Reference Model defines the three core concepts listed below, as enablers for autonomicity, cognitive networking and self-management as well as design and operational principles of autonomic-manager elements (i.e. Decision-Elements (DEs)):

- Functional Blocks (FBs), e.g. the different types of DEs, MBTS, ONIX and other special types of Information-Sharing Components/Systems.
- Reference Points (Rfps).
- Characteristic Information communicated on Reference Points by FBs.

The GANA Model defines a set of basic Reference Points (Rfps) which characterize Information and the actual mechanisms or protocol(s) that shall be employed to convey the information are meant to be further elaborated during the process of the GANA Model instantiation onto a target Implementation-Oriented Reference Architecture. GANA DEs such as QoS-Management-DE, Security-Management-DE, Routing Management-DE, and Forwarding-Management-DE, need to exchange information in order to collaboratively drive the autonomic control of various types of resources (e.g. protocols, stacks and mechanisms). Each of which shall be managed and regulated by a specific DE. As described in the GANA Model, each of such DEs when designed to operate at the "network-level", has a "mirror" DE at the "Function-level" (i.e. Level-2 in GANA) that operates in a faster time scale for those types of adaptive behaviours for which node/device intelligence is dedicated to the ability of a node/device to react based on decisions made at a low level (i.e. "Function-Level"). For example, "Network-Level-Routing-Management-DE" should have a mirror "Function-Level-Routing-Management-DE" that operates within a routing node/device and implements a faster control-loop (though the two mirror DEs and their associated control-loops interwork with each other).

6.3 Summary Table of Reference Points in GANA

Table 4 provides a summary of the Reference Points in GANA, which are further described in the clauses that follow and with complementary details in ETSI GS AFI 002 [2].

NOTE: In addition to the Reference Points summarized in this clause there are other Reference Points (Rfps) that derive from the APIs and protocols necessitated by GANA in cases such as the integration of GANA with frameworks for SDN, NFV, OSS, E2E Service Orchestrator, Big-Data Analytics as described in Annex B on Requirements for Protocols and APIs for Enabling GANA based Autonomics, Cognitive Networking and Self-Management of Networks and Services in Evolving and Future Networks.

Table 4

Reference Point Name (see note 1)	Alias Name of the Rfp	Characteristic Information communicated over the Reference Point	Additional Comments on where the Reference Point is described
Rfp_GANA-Level2&3-AccessToProtocolsAndMechanisms (see note 2) See note 5	DeMe	• **Views:** e.g. event notifications, monitoring data are communicated to Function-level-DEs by their specifically assigned Managed Entities (MEs)-i.e. Protocols, Stacks and Mechanisms (see clause 6.4.2.1 in the present document and clauses 9.11.5 and 9.11.6 of ETSI GS AFI 002 [2] on assignment of DEs to specific types of MEs). • **Commands** are issued by a specific Function-Level-DE e.g. Function-Level-Routing-Management-DE, to its specifically assigned Managed Entities (i.e. protocols and mechanisms such as routing protocols and mechanisms) in order to (re)-configure and regulate the behaviour of the ME(s).	This node/device internal interface is meant to enable the loading of DEs coming from other parties other than the device vendor. The DEs would access and autonomically manage and control the Protocols and Mechanisms of the device. See clause 9.6. of ETSI GS AFI 002 [2] See Figure 19 in ETSI GS AFI 002 [2] See also clause 6.4.2.1 in the present document.
Rfp_FunctionLevelDE-to-FunctionLevelDE	FuDe FFuDe (for federated AMC across different domains)	• **Trust and Authentication** exchange of messages and other types of messages exchanges necessary. • **Domain Type(s)** to which a DE involved is bound need to be exchanged. **Domain Identifier(s)** of DEs hosted by entities belonging to different administrative domains need to be exchanged. • **Views** can be communicated by a particular Function-Level-DE to other peer Function-level-DEs on other nodes/devices, especially concerning events or issues a function of a node e.g. Routing-Function cannot resolve (by performing some action) without jeopardizing network integrity (objectives).	See clauses 9.8 and 11.10 of ETSI GS AFI 002 [2] See Figure 20 and Figure 34 in ETSI GS AFI 002 [2] For Domain Type(s) and Domain Identifier(s) refer to the clause on Federation in GANA and in ETSI GS AFI 002 [2]. See also clauses 6.4.3.1 and 6.4.4.2 in the present document.

Reference Point Name (see note 1)	Alias Name of the Rfp	Characteristic Information communicated over the Reference Point	Additional Comments on where the Reference Point is described
		• **Control Information** exchange between Function-Level-DEs via the DE-2-DE interactions to achieve a "network-intrinsic management and control". Such interactions may include the notion of "compartment formation, policies of operation and compartment management" by DE-2-DE communication in a distributed fashion.	
Rfp_NodeMainDE-to-NodeMainDE	NoDe FNoDe (for federated AMC across different domains)	Similar types of Characteristic Information as in the case of the Reference Point "Rfp_FunctionLevelDE-to-FunctionLevelDE". The difference being the scope for which the Characteristic Information applies i.e. this case applies to the scope of the node/device level than a particular Function-Level (lower level).	See clauses 9.8 and 11.10 in ETSI GS AFI 002 [2] See Figure 20 and Figure 34 in ETSI GS AFI 002 [2]. See also clauses 6.4.3.2 and 6.4.4.3 in the present document.
Rfp_NetworkLevelDE-to-NodeMainDE	NeMe	• **Trust and Authentication** exchange of messages and other types of messages exchanges necessary. • **Domain Type(s)** to which a DE involved is bound need to be exchanged. **Domain Identifier(s)** of DEs hosted by entities belonging to different administrative domains need to be exchanged. • **Views** are communicated to Network-level-DEs, especially concerning events or issues a node/device cannot resolve (by performing some action) without jeopardizing network integrity (objectives). • **Commands** may be issued by a Network-Level-DE to the node or to a Function-Level-DE via the Node-Main-DE.	See clauses 9.8, 9.9 and 9.13.5 in ETSI GS AFI 002 [2] See Figure 21, Figure 22 and Figure 34 in of ETSI GS AFI 002 [2] For Domain Type(s) and Domain Identifier(s) refer to the clause on Federation in GANA in of ETSI GS AFI 002 [2]. See also clause 6.4.2.3 in the present document.

Reference Point Name (see note 1)	Alias Name of the Rfp	Characteristic Information communicated over the Reference Point	Additional Comments on where the Reference Point is described
Rfp_ModelBasedTranslationService-to-NodeMainDE	NeM	• This is a refinement of the Reference Point "Rfp_NetworkLevelDE-to-NodeMainDE" to involve a case whereby Network-Level-DEs communicate with a Node-Main-DE via a Model-Based-Translation Service (MBTS) that translates COMMANDS from Network-Level-DEs and RESPONSES from nodes/devices to a form usable by the targeted entity.	See clauses 9.13.5 and 11.7 in ETSI GS AFI 002 [2] See Figure 64 and Figure 34 in of ETSI GS AFI 002 [2]. See also clause 6.4.2.5 in the present document.
Rfp_AMC-ModelBasedTranslationService-to-ONIX	NoI	• **Trust and Authentication** exchange of messages and other types of messages exchanges necessary. • **Operations/Messages for Storing and Retrieving Information from the ONIX system.** [For example, the MBTS can use the publish/subscribe services of the ONIX that enable Advanced Auto-Discovery of Information and Resources, to retrieve Information about Network Elements/Nodes, such as Capability Description Models of individual nodes/devices, self-advertised/published by an individual node/device upon initialization. Capability Models of a node/device include technological features supported, including management protocols supported and Information about Managed Objects (MOs) of the technologies (e.g. protocols, etc.). Capability Models may include apart from technological features, vendor information].	See clauses 9.13.5 and 11.7 in ETSI GS AFI 002 [2] See Figure 64 and Figure 34 in of ETSI GS AFI 002 [2]. See also clause 6.5.1 NoI Reference Point: GANA Node-Main-DE-to-ONIX Rfp in the present document.

Reference Point Name (see note 1)	Alias Name of the Rfp	Characteristic Information communicated over the Reference Point	Additional Comments on where the Reference Point is described
Rfp_NetworkLevelDE-to-NetworkLevelDE	NeDe FNeDE (for federated AMC across different domains)	• **"Views"** such as **Policy changes** by the human operator; **challenges to the network's operation** from the perspective of a particular DE e.g. detected faults, threats, etc.; "views" communicated from lower-Level DEs in nodes/devices that require Net-Level-DEs to share and act upon if necessary. • **Domain Type(s)** to which a DE involved is bound need to be exchanged. **Domain Identifier(s)** of DEs hosted by entities belonging to different administrative domains need to be exchanged. • **Negotiations and Synchronization of Actions and Policies.**	This Reference Point between Network-Level-DEs is independent of the types of Network-Level-DEs and so should be considered as a common type of Reference Point between any Network-Level-DEs. See clauses 9.9, 9.13.5 and 11.7 in ETSI GS AFI 002 [2] See Figure 22, Figure 34 and Figure 64 in of ETSI GS AFI 002 [2] For Domain Type(s) and Domain Identifier(s) refer to the clause on Federation in GANA in ETSI GS AFI 002 [2]. See also clauses 6.4.3.3 and 6.4.4.4 in the present document.
Rfp_NetworkLevelDE-to-Data_Storage	NeI	• **Trust and Authentication** exchange of messages and other types of messages exchanges necessary. • **Operations/Messages from the DE for retrieval of Data or (Knowledge created out of raw data)** from a storage such as Data Collector that gathers data such as: IPFIX Data, SNMP BulkStats Data, NetFlow Data, Flow Traces, Traffic Matrix, etc., OR Any Data that is not suitable to be stored and shared through the ONIX system. • **Knowledge created out of raw data by Algorithms running on the Data Storage, that operate on raw data and create Knowledge for export to the Knowledge Plane (i.e. to Net-Level-DEs).** • **Data that may need to be communicated by the Storage to the particular DE.**	See clauses 11.7 and 9.13.7 in ETSI GS AFI 002 [2] See Figure 64 and Figure 38 in ETSI GS AFI 002 [2]. See also clause 6.5.2 in the present document.

Reference Point Name (see note 1)	Alias Name of the Rfp	Characteristic Information communicated over the Reference Point	Additional Comments on where the Reference Point is described
Rfp_NetworkLevelDE-to-ONIX-System	NeI	• **Trust and Authentication** exchange of messages and other types of messages exchanges necessary. • **Operations/Messages for Storing and Retrieving Information from the ONIX system.**	See clauses 9.13.4 and 10 in ETSI GS AFI 002 [2] on the Use of Network Profiles, Policies, Objectives, Config-Data, and Capabilities of network elements; and 11.7 in of ETSI GS AFI 002 [2] See Figure 64 of ETSI GS AFI 002 [2]. See also clause 6.5.2 in the present document.
Rfp_NodeMainDE-to-ONIX-System	NoI	• **Trust and Authentication** exchange of messages and other types of messages exchanges necessary. • **Operations/Messages for Storing and Retrieving Information from the ONIX system.**	See clauses 9.13.4 and 10 in ETSI GS AFI 002 [2] on the Use of Network Profiles, Policies, Objectives, Config-Data, and Capabilities of network elements; and 11.7 in ETSI GS AFI 002 [2]. See also the figure in clause 9.13.5 in ETSI GS AFI 002 [2] See Figure 64 and Figure 34 in ETSI GS AFI 002 [2]. See also clause 6.5.1 in the present document.
Rfp_OSS-to-ONIX-System	OsI	• **Trust and Authentication** exchange of messages and other types of messages exchanges necessary. • **Operations/Messages for Storing and Retrieving (mainly) Information from the ONIX system.**	See clauses 11.7 and 11.8 in of ETSI GS AFI 002 [2] See Figure 67 in particular, Figure 64 and Figure 68 in ETSI GS AFI 002 [2]. See also clause 6.5.3 in the present document.
Rfp_OSS-to-Network-Level-DEs	OsDe	• **Trust and Authentication** exchange of messages and other types of messages exchanges necessary. • **Management COMMANDS normally sent to the network by an OSS** through the so-called "network-adapter interface" need to be rather sent directly to the Network-Level-DE (considering that they are ones that take the full responsibility of performing Autonomic Management of the Network), and NOT to the network directly.	This case applies to configurations where OSS systems are integrated to co-exist and interwork harmoniously with Network-Level-DEs in the overall management of the network. The current (today's) OSS-Network Interface would need to be "re-directed" towards Network-Level-DEs (assuming that Network-Level-DEs take full responsibility for network management and control). See clauses 11.7 and 11.8 in ETSI GS AFI 002 [2] See Figure 67 in particular, Figure 64 and Figure 68 in ETSI GS AFI 002 [2].

Reference Point Name (see note 1)	Alias Name of the Rfp	Characteristic Information communicated over the Reference Point	Additional Comments on where the Reference Point is described
			See also clause 6.4.2.4 in the present document.
Rfp_EMS_OR_NMS-to-NodeMainDE	NeM	• **Trust and Authentication** exchange of messages and other types of messages exchanges necessary. • **Management COMMANDS** targeting nodes/devices designed following GANA principles. A Manager in the sense of a traditional EMS/NMS, may create a "Wrapper packet/message that encapsulates a COMMAND" e.g. a SET/WRITE COMMAND on a Variable, and send the packet/message to the Node-Main-DE of a node/device where the Node-Main-DE extracts the COMMAND and relays it to the appropriate Function-Level-DE responsible for autonomically managing and controlling the ME targeted by the COMMAND. The DE then reasons about whether to apply the COMMAND, and if yes, the DE executes the COMMAND directly on the ME's management-interface OR issues the COMMAND via the "loopback interface" to the local Management Agent (on the node/device) for execution if the DE manages and controls the ME indirectly through the Management Agent (see note 3).	This case applies to configurations where today's management systems are integrated to co-exist and interwork harmoniously with Network-Level-DEs in the overall management of the network. See clauses 11.7 and 11.8 in of ETSI GS AFI 002 [2] See Figure 64 and Figure 68, see also Figure 63 in ETSI GS AFI 002 [2]. See also clause 6.4.2.5 in the present document.
Rfp_ONIX-to-ONIX	FOO	• Domain information (including Domain Type(s), Domain Identifier(s)) exchange that may be conveyed through a local ONIX instance to facilitate for federated AMC.	See clause 6.4.4.5 in the present document.

Reference Point Name (see note 1)	Alias Name of the Rfp	Characteristic Information communicated over the Reference Point	Additional Comments on where the Reference Point is described
Rfp_FederationMBTS- to-FederationMBTS	FMM	• Domain information (including Domain Type(s), Domain Identifier(s) exchange between F-MBTS instances that enable two domains to exchange information and control messages for enabling federated AMC across the domains- information and control messages that need to be translated if the domains involved use different data models and information types and formats that are all to be employed in federated AMC.	See clause 6.4.4.6 in the present document.
Rfp_OSS_to_G-MBTS	GoS	• Input to the Knowledge Plane that is generated by an OSS Tool on the Governance interface (refer to OSi, OsDe Rfps) and is input to G-MBTS for translation into the language, data model(s) and data formats employed by the GANA Knowledge Plane DEs.	See clause 6.4.2.6 in the present document.
NOTE 1: "**to**" in the name does not mean unidirectional communication only. In some of the cases communication may be initiated by either party and can be bi-directional. NOTE 2: "GANA-Level2" is also called "Function-Level". NOTE 3: An alternative to this approach is presented in the corresponding clause in ETSI GS AFI 002 [2] where this reference point is defined. NOTE 4: Annex C in the present document presents one of the ways in which this Reference Point can be implemented, that of using a unified API. Another may the reference point may be implemented is through the traditional approach of using means such as SNMP, by which the GANA Level 3 and Level-2 DEs locally manage and control their MEs through the local SNMP agent and the MIBs implemented for those MEs. NOTE 5: Rfp_GANA-Level2-AccessToProtocolsAndMechanisms defined in ETSI GS AFI 002 [2] is synonymous to this Rfp that considers both GANA Level2 and Level3 access to ProtocolsAndMechanisms that are on the Resources Layer of a GANA node (NE) (see note 4).			

6.4 GANA DE-to-DE and DE-to-ME Reference points (Rfps)

6.4.1 Vertical Reference Points and Horizontal Reference Points in GANA, GANA integration with Legacy Management Systems, and Federated AMC Reference Points

Regarding DE reference points, a VRP is a hierarchical reference point between peer DEs in different GANA DE levels and a HRP is a horizontal reference point between DEs within the same GANA DE Level. There are other types of references points defined in GANA that relate to the integration of GANA with legacy management systems, and also reference points that relate to federated AMC across various administrative domains.

6.4.2 Vertical Reference Points in GANA

6.4.2.1 DeMe: reference point between the Node-Main-DE or Function Level DEs and Managed Entities (MEs) in the resources layer of a GANA node (NE)

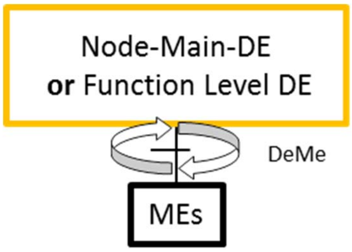


Figure 13: Rfp DeMe: Node-Main-DE_or_Function-Level-DE to Fundamental Managed Entities (MEs)

An ME communicates "**Views**" such as event notifications (event detections) and monitoring data to Function Level DEs or to the Node-Main-DE accordingly to the DEs' specifically assigned Managed Entities (MEs) in the resources layer-i.e. Protocols, Stacks and Mechanisms, etc.

Description: Reference Point between a Level-2 or Level-3 DE and its assigned ME(s) at the resources layer of a GANA node (where e.g. protocols/stacks reside) should use this Rfp as described in the DE-ME(s)interface in the DE Model.

Commands are dynamically (autonomically)issued by a specific Function-Level-DE e.g. Function-Level-Routing-Management-DE, to its specifically assigned Managed Entities (i.e. protocols and mechanisms such as routing protocols and mechanisms) in order to adaptively (re)-configure and regulate the behaviour of the ME(s) to meet certain objectives.

6.4.2.2 NoMe: reference point (Rfp_NMDE-to-Level2-DE) between GANA-Node-Main-DE and a Function-Level DE(s)

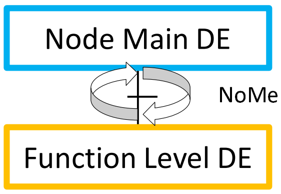


Figure 14: Rfp_NMDE-to-Level2-DE:Reference Point between a Node Main DE and Function level DEs

The **Views** exposed by MEs and function level DEs to the Node-Main-DE on the Rfp (characteristic information exchange): e.g. event notifications (event detections), monitoring data communicated to Node-Main-DE by the lower level DEs and MEs assigned to the Node-Main-DE.

Description: Reference Point between the Node-Main-DE and the lower -level DEs. The GANA function Level DEs should use this Rfp as described in the DE-ME Rfp in the DE Model.

Commands are issued by a specific sub-DE of the Node-Main-DE e.g. Fault-Management-DE (FM-DE), to its specifically assigned MEs and lower-level DEs (i.e. all the DEs in the Node-as the FM-DE shall manage certain faults in the Node) in order to implement the GANA node's self-repair and self-diagnose functionality for DE(s) at the Function Level DEs.

6.4.2.3 NeMe: Vertical Reference Point Network-Level-DE-to-NodeMainDE

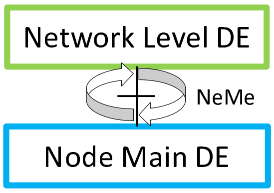


Figure 15: Rfp_NeMe: Reference Point between a Network Level DE and Node Main DEs

Information (e.g. state information) from a Network Element (NE) that is autonomically managed by network level DEs may be conveyed to the DEs through this reference point, e.g. some state information may be conveyed to a particular network level DE by its assigned DEs and MEs in the GANA node. The network level DEs send commands such as policy-control messages and policies to the GANA Node through this reference point.

Description: this reference point may embed security mechanism for Trust and Authentication exchange of messages along with other types of messages exchanged between a GANA Node and the Knowledge Plane. Through the Node's Security DE trust and security related message exchanges happen. Other network level DEs use the reference point to perform policy-control of their mirror DEs in a GANA node through the Node-Main-DE, conveying configuration related data for the node-local DEs through the Auto-Configuration DE of the node (especially if the target DEs are Function-Level-DEs, else if the target is the other sub-DEs of the Node-Main-DE (e.g. FM_DE) the mirror DE on the network level may interact with the target node-local DE directly-provided that bypassing the Auto-Configuration does not result in conflicting configuration state of the GANA node). The Auto-Configuration DE may be used as a proxy in all the interactions between the Knowledge Plane DEs and any node-local DE. Only DEs from the same administrative domain and its assigned ME/DE should use this reference point. Information pertaining to the Domain Type(s) to which a DE involved is bound need to be conveyed by the DE.

NOTE: This Rfp is further specialized(refined) by the reference point Rfp_ModelBasedTranslationService-to-NodeMainDE as described earlier in the summary table on reference points in GANA.

Characteristic Information exchange, such as Message exchanges (e.g. commands issued by DEs): operations/primitives on this Rfp should be based on operation/primitives of the DE-ME interface of the DE Model. The Rfp implementation depends on the management interface used for the GANA Node or the network element as well as the implementation of the AMC-MBTS.

6.4.2.4 OsDe: Vertical Reference Point: OSS/BSS-to-Network-Level-DE

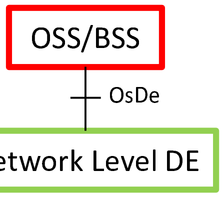


Figure 16: Rpf_OsDe Reference Point between OSS/BSS and Network Level DEs

Views (Characteristic Information) exchange: OSS/BSS may be used to send and update the GANA Profile to DEs in the Knowledge Plane such that the DEs augment the GANA Profile with additional configuration data necessary. Information (e.g. configuration data for KP DEs or Views such as network or node state related information) may be communicated to Network Level-DEs by the OSS through the OsDe Rfp used to govern the target Network level DEs. The KP DEs may expose views (e.g. in form of Reports on autonomic decisions performed by the KP DEs over a certain time) to the OSS/BSS. The GANA Profile used to manage the network NEs through Network Level DEs may also be retrieved from the ONIX via the NeI Rfp. More details on this subject are found later in the clause on GANA Operations Procedures for the Human Network Operator.

Additional insights: Administrators of an administrative domain may need to retrieve and store certain types of information in the ONIX. An administrator associated with a network operations related centermay retrieve and store only information it is authorized to access or store. For example, administrators of the sales managementcenter, the marketing center, the service management center, the customer service management center, the technical management center and the financial analyst center may not be allowed to use the same information and may not be permitted to store the same type and level of information according to their roles/profiles even if they belong to the same administrative domains. As described earlier, the Autonomic network shall be governed through GANA Profile, which describes the objectives set for the autonomic network by administrator of the administrative domain. The OSS/BSS may be used to retrieve information stored by the administrators in the ONIX to ensure that related SLA/QoS needs are well encapsulated in the GANA Profile supplied as input to the KP DEs for the governance and configuration of the autonomic network (as illustrated in the clause on GANA Operations Procedures for the Human Network Operator).

COMMANDS normally sent to the network by an OSS could be rather sent directly to the Network-Level-DEs (considering that they are ones that over the long term take the full responsibility of performing Autonomic Management of the Network). However, another possibility is that Network-Level-DEs (Knowledge Plane) may send commands to (re)-configure Network Elements (NEs) through the OSS.

Commands operations/primitives used for implementing the Rfp should use operation/primitives of the "**Management-interface**" of the DE Model.

6.4.2.5 NeM Vertical Reference Point: Network Element-to-AMC-MBTS

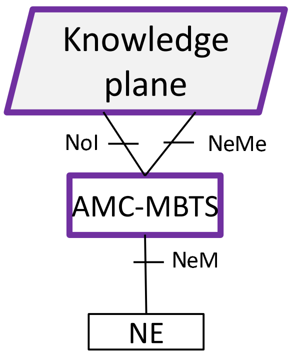


Figure 17: Rfp_NeM Reference Point between Network Element and KP through AMC-MBTS

Views (Characteristic Information) exchange: KP DEs (Network-Level-DEs) retrieve information from an NE and send commands for policy enforcements to Network Element through the AMC-MBTS, which also translates the information retrieved from Network Element to the language understood by the KP and send KP DEs' commands to NEs using the appropriate vendor specific and management and control technology-specific operation and primitives. The same Rfp may be used by an EM/NM. More details on this subject are found in a later clause on GANA Operations Procedures for the Human Network Operator.

NOTE 1: This Rfp (*Rfp_ModelBasedTranslationService-to-NodeMainDE*) is a further specialization (refinement) of the reference point Rfp_Network Level DE-to-Node_Main_DE as described earlier in the summary table on reference points in GANA.

NOTE 2: This is a refinement of the Reference Point NeMe "Rfp_NeMe" to involve a case whereby Network-Level-DEs communicate with a Network element via the AMC-Model-Based-Translation Service (MBTS) that translates by mediation the COMMANDS from Network-Level-DEs to and NE and RESPONSES from nodes/devices and also data/information from the NE to a form usable by the targeted entity on either side of the MBTS. As already described in the clause on MBTS in the present document, the MBTS should implement a RAT (Representation, Acquisition and Translation) function for derivation of Knowledge from information/data supplied by NEs under the scope of the MBTS.

NOTE 3: Implementers should take into consideration the aspects concerning the MBTS's RAT function described in the clause on MBTS in the present document.

6.4.2.6 G-Os: Vertical Reference Point: OSS/BSS-to-Network-Level-DE through G-MBTS

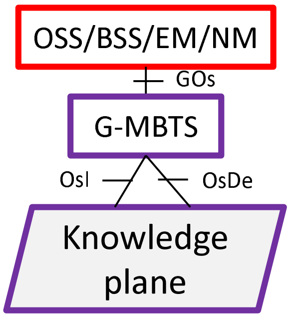


Figure 18: Rfp_GOs: Reference Point between OSS/BSS/EM/NM and G-MBTS

Views (Characteristic Information) exchange: This Rfp is a refinement of OsDe and OSS ONIX Rfps through G-MBTS. KP DEs retrieve the skeleton GANA Profile (includes input policies) through the Governance MBTS (G-MBTS), which translates the information and input data sent from OSS/ EM/NM to the language that KP DEs understand, while at the same time it inserts the translated information/data into the GANA Profile (skeleton) that KP DEs would augment and operate upon as described later in the clause on GANA Operations Procedures for the Human Network Operator. The Governance MBTS shall also provide security and trust mechanisms which are required for the different network operations related centers of interest authorization, and for the different administrative domains.

Treatment of commands issued by DEs: This is a refinement of the Reference Point OsDe to involve a case whereby Network-Level-DEs communicate with a OSS/NM/EM via a Governance Model-Based-Translation Service (MBTS) that translates COMMANDS (Network Profile) from/to OSS/NM/EM and RESPONSES from/to Network-Level-DEs to a form usable by the targeted entity on either side depending on the direction of a COMMAND or RESPONSE.

The Governance MBTS may be used to translate information from the ONIX to the OSS/BSS according to the requested information and the need of the different application and tool in the OSS/BSS. Trust and Authentication exchange of messages and other types of messages exchanges is necessary.

6.4.3 Horizontal Reference Points (Rfps)-to enable to coordinate DE decisions within the same GANA Level

6.4.3.1 FuDe: Horizontal Reference Point: FunctionLevelDE-to-FunctionLevelDE

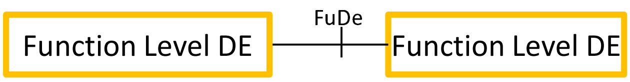


Figure 19: Rfp_FuDe Reference point Between Function level DEs

Views (characteristic information) exchange: Reference point used for interactions between Function Level DEs within the same administrative domain.

Description: information exchange between Function-Level-DEs via the DE-2-DE interactions to achieve a "network-intrinsic management and control". Such interactions may include the notion of "compartment formation, policies of operation and compartment management" by DE-2-DE communication in a distributed fashion.

Peer to peer DE communications may be used as described in the DE Model by a particular Function-Level-DE to other peer Function-level-DEs on other nodes/devices, especially concerning events or issues a function of a node e.g. Routing-Function may not be able to resolve alone (by performing some action) without jeopardizing network integrity (objectives).

Some mechanisms may be embedded within this reference point for realizing **Trust and Authentication** exchange of messages along with the other types of messages exchanges necessary between the DEs. Refer to DE-PeerDE interface in the DE Model for information on the types of messages that may be exchanged by DEs.

6.4.3.2 NoDe: Horizontal Reference Point: NodeMainDE-to-NodeMainDE

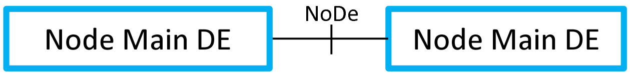


Figure 20: Rfp_NoDe: Reference Point between Node Main DEs

Views (characteristic information) exchange: This Reference point (Rfp) is used for interactions between Node Main DEs and within the same administrative domain.

Description: Similar types of Characteristic Information as in the case of the Reference Point "Rfp_FunctionLevelDE-to-FunctionLevelDE" may be exchanged by the DEs. The difference being the scope for which the Characteristic Information applies i.e. this case applies to the scope of the node/device level than a particular Function-Level (lower level). Refer to DE-PeerDE interface in the DE Model for information on the types of messages that may be exchanged by DEs.

6.4.3.3 NeDe Horizontal Reference Point: NetworkLevelDE-to-NetworkLevelDE

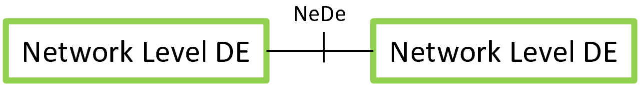


Figure 21: Rfp_NeDe Reference Point between Network level DEs

Views (characteristic information) exchange: Reference Point used for interactions between Network Level DEs within the same administrative domain**.** Similar types of Characteristic Information as in the case of the Reference Point "Rfp_FunctionLevelDE-to-FunctionLevelDE". The difference being the scope for which the Characteristic Information applies, i.e. this case applies to the scope of the node/device level than a particular Function-Level (lower level)."Views" such as Policy changes by the human operator; challenges to the network's operation from the perspective of a particular DE e.g. detected faults, threats, etc.; "views" communicated from lower-Level DEs in nodes/devices that require Net-Level-DEs to share and act upon if necessary. Negotiations and Synchronization of Actions and Policies between different types of DEs which realize a control loop composed of different DEs. Refer to DE-PeerDE interface in the DE Model for information on the types of messages that may be exchanged by DEs.

6.4.4 AMC Federation Reference Points

6.4.4.1 A General Characterization of Reference Points (Rfps) pertaining to Federated AMC across Domains

NOTE: In the federation of Autonomic Management and Control (AMC) of Networks and Services, "domain" in the diagrams in clauses 6.4.4.2 to 6.4.4.6 means either "Administrative Domain", or "Technical or Technology" Domain as described in ETSI GS AFI 002 [2] in the clause on Federation in GANA and also in ETSI TS 103 194 [1].

Two or more Decision Elements (DEs) that intend to form peers across a Reference Point of nature "DE-to-DE interface across multiple network elements/devices" need to discover the following types of information concerning their peers:

- **Domain Type(s):** Whether technical or administrative, of which the DE is member and is willing to share this information (subject to security and trust policies). Regarding the discovery of a "technical" domain to which a DE belongs (or is able to orchestrate and autonomically manage and control), the GANA Model includes the concept of Capability Model and its self-description and publishing by a DE (refer to the clause on Network Governance), through which information about a "technical domain" is encapsulated. A Decision Element should aggregate capabilities of its assigned Managed Entities (MEs) and advertise the descriptive information when required (subject to security policies). For example, a *Function-Level Routing-Management-DE* in a routing device running two routing protocols e.g. OSPF and BGP, would indicate the routing protocols (its Managed Entities) under its control, in its Capability Model description.

- **Domain Identifier(s):** Administrative domains may have Domain Identifiers assigned by the governing authority following some scheme of choice.

Decision Elements (DEs) use such information to verify against security and trust policies, and to behave accordingly in the way they configure their respective Managed Entities (MEs) to fulfil the required network behaviours across domain boundaries. However, the exchange of such information may possibly be restricted to the Node-Main-DEs discovering and exchanging such information on-behalf of the lower level (Function-Level) DEs that then use the information to behave accordingly in the way they provision a service across domain boundaries.

The reference point below refers to reference point between DEs within different administrative domain within the same GANA-Level. Such reference points may embed security and trust mechanisms embedded in the DEs or in separate function or mechanism may be available for that purpose.

A dedicated reference point between administrative domains may be defined when GANA is instantiated in particular reference architecture (e.g. access and core mobile network). In some other case interworking functions may be specified with MBTS functions and broker functions used to specify a distributed peer to peer KPs within different administrative domains.

The first federation Rfps need to be carefully designed as they may be required to fulfil requirements for proactive and fast reaction by the DEs. Characteristic information and operations need to be semantically and formalized in the same way between two different domains.

At the Knowledge Plane level, between ONIX instances belonging to different domains or between DEs belonging to different domains, an MBTS instance may be used to translate the information retrieved within a dedicated domain to information representation and presentation format(s) required by the peer domain.

6.4.4.2 FFuDe: Federation Horizontal Reference Point: FunctionLevelDE-to-FunctionLevelDE

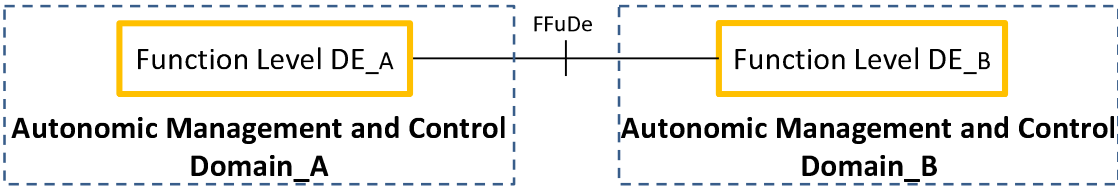


Figure 22: Rfp_FFuDe Federation Reference point between Function level DEs within different KPs

This reference point is used for interactions, cooperation and coordination between Peer to peer Function Level-DEs hosted by systems (nodes)within different administrative domains. The information between peers Function Level-DEs may require security and trust mechanisms within such Rfp (interactions). Same operations/primitives as required of the DE2DE_Iinterfaceof the GANA DE Model may be employed.

The full picture on the variety of characteristic information that may be exchanged on the reference point is a subject "for further study" (FFS).

6.4.4.3 FNoDe: Federation Horizontal Reference Point: NodeMainDE-to-NodeMainDE

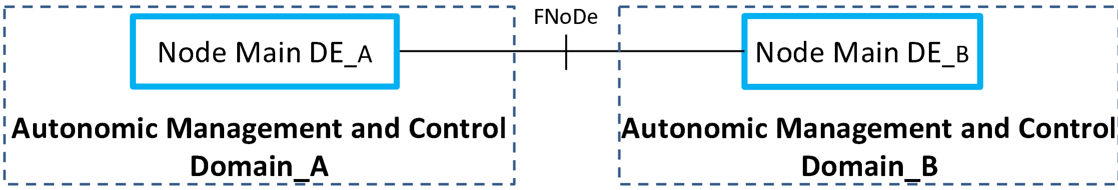


Figure 23: Rfp_FNoDe Federation Reference Point between Node-Main-DEs in different KPs

This reference point is used for interactions, cooperation and coordination between Peer to peer Node Main-DEs hosted by systems (nodes) indifferent administrative domains. The information between peers Node-Main-DEs may require security and trust mechanisms within such an Rfp (interactions). Same operations/primitives as required of the DE2DE_I interface of the GANA DE Model may be employed.

The full picture on the variety of characteristic information that may be exchanged on the reference point is a subject "for further study" (FFS).

6.4.4.4 FNeDe Federation Horizontal Reference Point: NetworkLevelDE-to-NetworkLevelDE

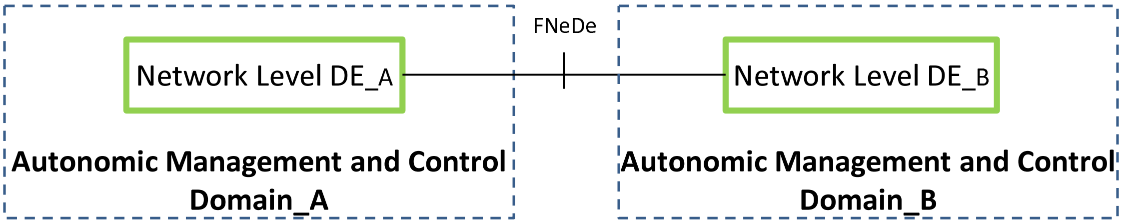


Figure 24: Rfp_FNeDe Federation Reference point between Network level DEs belonging to different KPs

This reference point is used for interactions, cooperation and coordination between Peer to peer Network-Level-DEs in KPs belonging to different domains. The information between peers Network-Level-DEs may require security and trust mechanisms within such Rfp (interactions). Same operations/primitives as required of the DE2DE_Iinterface of the GANA DE Model may be employed.

The full picture on the variety of characteristic information that may be exchanged on the reference point is a subject "for further study" (FFS).

6.4.4.5 FOO Federation Horizontal Reference Point: ONIX-to-ONIX belonging to different KPs

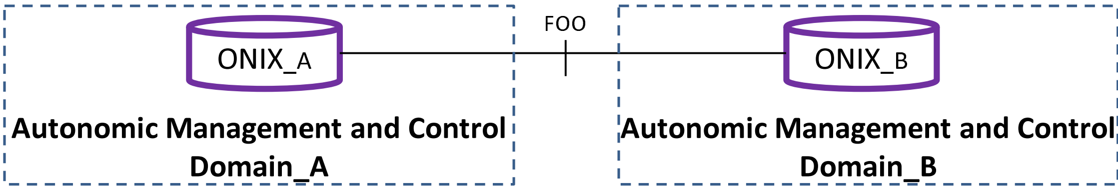


Figure 25: Rfp_FOO Federation reference Point between ONIXs belonging to different KPs

This reference point is used for interactions, cooperation and coordination between Peer to peer ONIX systems in KPs belonging to different domains. The information between peers ONIX systems may require security and trust mechanisms within such Rfp (interactions).

The full picture on the variety of characteristic information that may be exchanged on the reference point is a subject "for further study" (FFS).

6.4.4.6 FMM Federation Horizontal Reference Point: Knowledge Plane-to-Knowledge Plane whereby MBTS services may be required

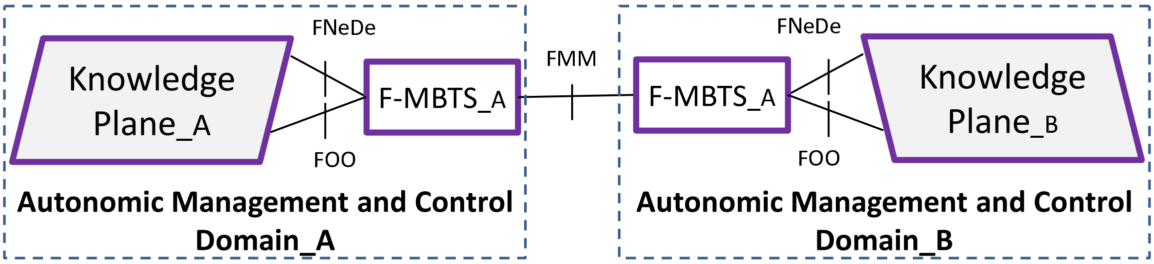


Figure 26: Rfp_FMM Federation reference Point between F-MBTSs belonging to different KPs

This Reference point is used for interactions between Peer to peer MBTS of KPs that belong to different administrative domain of which there is a need to translate data models and information exchange methods between the two KPs/domains involved. The information between the peer F-MBTS service components may require security and trust mechanisms within such Rfp (interactions).

The full picture on the variety of characteristic information that may be exchanged on the reference point is a subject "for further study" (FFS).

6.5 GANA Knowledge Plane (KP) Reference Points (Rfps)-Rfps within a KP belonging to a particular Administrative Domain

6.5.1 NoI Reference Point: GANA Node-Main-DE-to-ONIX Rfp

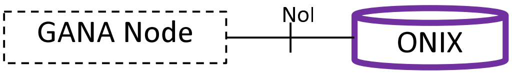


Figure 27: Rfp_NoI Reference Point between Node-Main-DE and ONIX

The Rfp implements Operations/Messages for Storing and Retrieving Information into/from the ONIX system. The Node-Main-DE shall use the publish/subscribe services of the ONIX that enable Advanced Auto-Discovery of Information and Resources, to retrieve Information that may be required by entities in the mode such as DEs. The Node-Main -DE also published node information such as the Node's Capability Description Model as described in ETSI GS AFI 002 [2]. Capability Models of a node/device include technological features supported, including management protocols supported and Information about Managed Objects (MOs) of the technologies (e.g. protocols, etc.). Capability Models may include apart from technological features, vendor information. Trust and Authentication exchange of messages is necessary for usage of ONIX services by the GANA node. Operations and primitives for the "Other interaction Interface" of a DE model should include the operations and primitives required for interactions with entities such as ONIX.

6.5.2 NeI Reference Point: NetworkLevelDE-to-ONIX Rfp

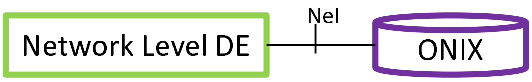


Figure 28: Rfp_NeI Reference Point between Network Level DE and ONIX

Through this Rfp a Network Level DE may retrieve and store information in the ONIX system based on the services of the ONIX and the information stored in the ONIX and information required to be stored into ONIX by KP DEs. The Rfp should implement Operations/Messages required for a Network-Level-DE to retrieve Data or Knowledge created out of raw data, from a storage such as a Data Collector that gathers data such as IPFIX Data, SNMP BulkStats Data, NetFlow Data, Flow Traces, Traffic Matrix, OR any Data, and shares the data/information or derived knowledge with the KP DEs through the ONIX. Any data/information that is suitable to be stored and shared through the ONIX system and is of interest to KP DEs to retrieve or store shall be stored or retrieved through this Rfp.

NOTE: For real-time knowledge streaming to KP DEs, knowledge may be created out of raw data by Cognitive Algorithms that run on a Data Storage (collector) and operate on raw data and create Knowledge for export to the Net-Level-DEs directly (not through ONIX) while storing the knowledge in ONIX as well (refer to the ETSI White Paper No.16 [3] for further details).Trust and Authentication exchange of messages is necessary for accessing the ONIX. ONIX-external protocols implement the Operations/Primitives for Storing and Retrieving Information into/from the ONIX system (refer to Annex D of the present document, on possible approach to implementing ONIX).

6.5.3 OsI Network Governance Reference Point: OSS/BSS to ONIX (Knowledge Plane)

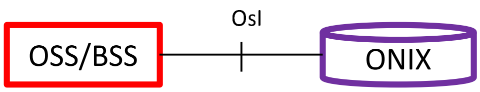


Figure 29: Rfp_OsI Reference Point between OSS/BSS and ONIX

Through this Rfp an OSS/BSS can send and update a GANA Profile to DEs through the ONIX, and also retrieve or store other types of information from/into the ONIX through the G-MBTS, which may be made to interwork with an appropriate tool in the OSS/BSS or with EM or NM functions. This Rfp is complemented by the GOs reference point ("OSS/BSS-to-Network-Level-DE through G-MBTS" Rfp).

6.5.4 NeM Reference Point: Knowledge Plane-to-Network Element through the AMC-MBTS

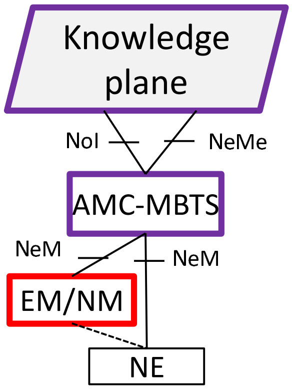


Figure 30: Rfp_NeM Reference Point between Network Element and KP through the AMC-MBTS

The knowledge plane may autonomically manage an NE in legacy infrastructure through an EM/NM.

As described in ETSI GS AFI 002 [2] a manager in the sense of a traditional EM/NM, may create a "Wrapper packet/message that encapsulates a COMMAND" e.g. a SET/WRITE COMMAND on a Variable, and send the packet/message to the Node-Main-DE of a node/device where the Node-Main-DE extracts the COMMAND and relays it to the appropriate Function-Level-DE responsible for autonomically managing and controlling the ME targeted by the COMMAND. The DE then reasons about whether to apply the COMMAND, and if yes, the receiving DE executes the COMMAND directly on the ME's management-interface OR issues the COMMAND via the "loopback interface" to the local Management Agent (on the node/device) for execution if the DE manages and controls the ME indirectly through the Management Agent.

6.6 GANA Operations Procedures for the Human Network Operator

6.6.1 Overall Insights on Operations Procedures that may be applied to Operating a GANA empowered Autonomic Network

This clause provides insights on operations procedures that may be applied to operating a GANA empowered autonomic network as inspired by the complementary aspects of Autonomic Management and Control (AMC) Automated Management described in the ETSI White Paper No.16 [3].

NOTE 1: The insights provided in this clause only serve as guidance as some other approaches may be pursued as well and may complement the approach described in this clause.

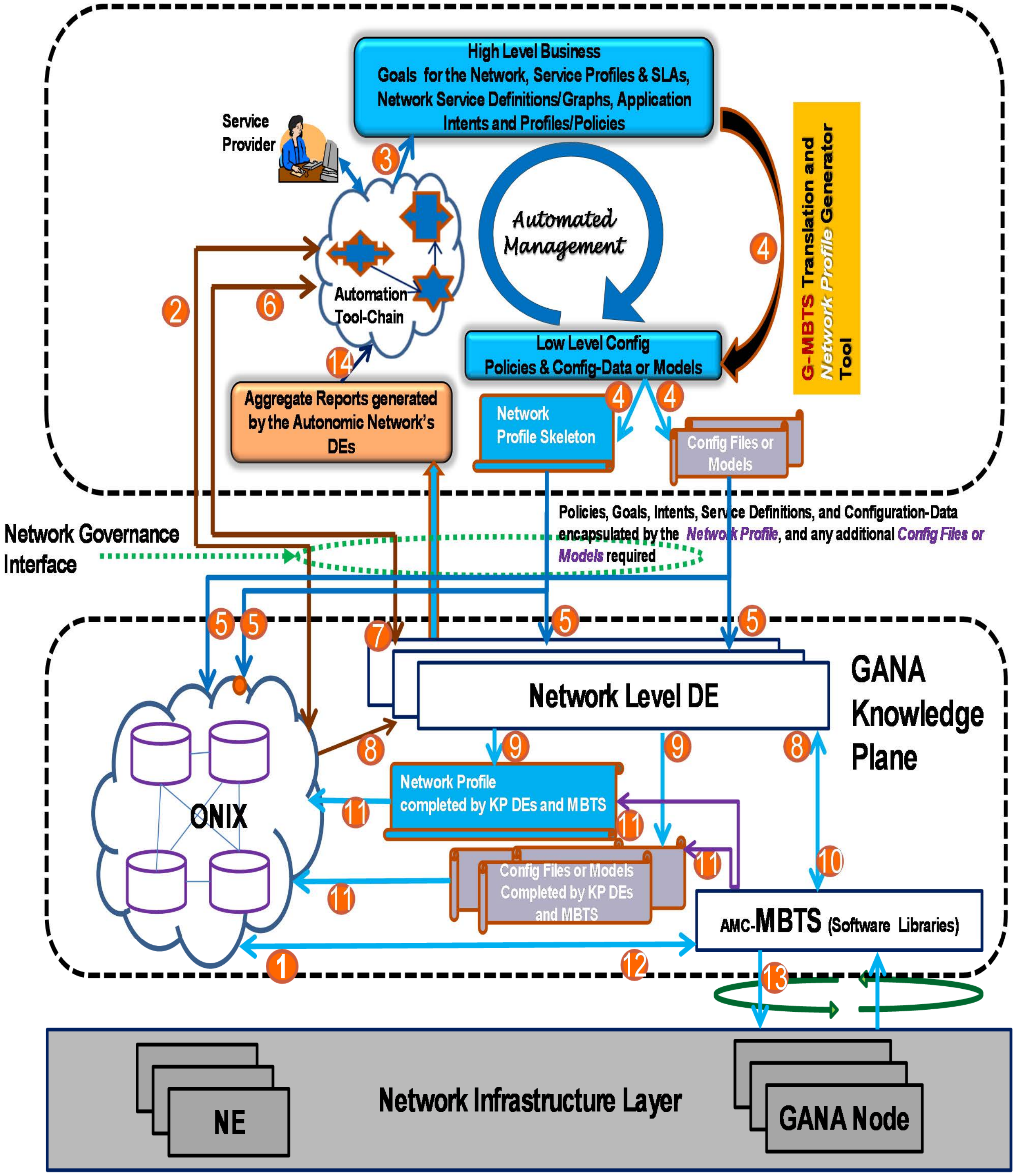


Figure 31: The Interworking of Automated Management and Autonomic Management through the GANA Operations Procedures

The following numbered activities on the diagram in Figure 31 illustrate the Operations Procedures for the Human Network Operator to operate the GANA empowered autonomic network and service management and control functions and the underlying GANA empowered autonomic network elements (NEs) in the network infrastructure.

NOTE 2: These illustrations and guidelines extend those described in the clause on GANA's Network Governance Interface and are further complemented by the further details provided in the Annex C on Operations Guide for GANA Empowered AMC and Autonomic Networks. Also, the aspects described in this guidance on GANA Operations Procedures are to be complemented by the understanding the derives from the interplay of the GANA Knowledge Plane with systems such as OSS, Service Orchestrators and Domain Orchestrators, NFV Orchestrators, EM/NMS, and SDN Controllers, as described in Annex B on "*Requirements for Protocols and APIs for Enabling GANA based Autonomics, Cognitive Networking and Self-Management of Networks and Services in Evolving and Future Networks*"-requirements expected to be further studied, elaborated and further detailed.

NOTE 3: The GANA Operations Procedures provided in this clause should be complemented with insights described earlier in the present document on "Knowledge Management on Knowledge Bases (KBs) and the role of ONIX, DEs and Automated Network Management Tools in Knowledge Management".

Assumptions: The following are aspects that have been left out of consideration in this illustration of how the human operator may operate the GANA empowered autonomic network and service management and control functions and the underlying GANA-empowered autonomic network elements (NEs) in the network infrastructure (the so-called GANA Nodes). Such aspects may require that the Operations Procedures and their implementations be realized differently than illustrated here:

- **Scenario that is out of consideration:** Knowledge Plane Multi-tenancy associated with network slices instantiated by external customers via a Self-Care Portal that creates network slices through the End-to-End Service Orchestrator.

- If anE2E Service Orchestrator is to be used by a Self-Care Portal that triggers service orchestration through invoking the E2E Service Orchestrator, then it shall emulate some of the processes triggered by the human operator that are described in table 5.

Table 5

Activity Number	Description of the Activity
1	NEs (GANA Nodes) that are initiated in the network infrastructure publish their capabilities into ONIX or publish them via the MBTS (AMC-MBTS) and wait to then receive configuration profiles and data for their DEs and MEs. The Capability Models Descriptions published include functional features supported by the GANA node (e.g. protocols and stacks that can be composed or configured), resource capabilities, management and control protocols supported, vendor related information, topological information pertaining to e.g. low level connectivity that may have already been established at NE initialization on a link, neighbors discovered, Network Interface Cards (NICs) and/or other types of connectivity enabling resources, the DEs that are available for orchestration (run-time instantiation), etc.
2	With the aid of Automated Management Tools (e.g. OSS Tools), the human operator retrieves information about NEs (GANA nodes) that initialized in the network infrastructure or are already running and have published Capabilities (Capabilities Description Models) into the ONIX.
3	The Human Operator uses Automated Management Tools (which include Service Fulfilment Tools, OSS/BSS Tools) to define or modify High Level Business Goals for the Network, Service Profiles andSLAs, Network Service Definitions/Graphs and Policies, Application Intents (see the definition of Intent Based Networking (IBN) in [i.64] for insights on this paradigm that is yet to be fully developed and mature) and Profiles/Policies. The Network Service Definitions may be defined based on the NE Capabilities (including resource or capacity available in the NE)compiled from NEs (physical or virtual) of the network infrastructure, else Network Service Definitions defined independently of NE resources and capacity availability need to be matched against NE Capabilities discovered from the ONIX Inventory to map them to specific NE resources and capacity available to verify if they can be instantiated and make modifications as may be necessary. Network Service Definitions include Service Topologies consisting of Service Nodes (Service Functions) that are required to deliver an end to end service required by end user traffic flows. In the automated management activities, End User Service Profiles and SLAs are also mapped to Service Chains defined and validated by the operator, and also Application Intents and Profiles/Policies are also edited and validated against conflicts. Note that Knowledge Plane DEs together with amc-MBTS play a role in dynamic Network Service (re)-Programming as discussed in ETSI White Paper No.16 [3] as well as in Flow (re-)Programming as driven by SLA violations and the need for Dynamic Service Chaining as well, and other dynamics the Knowledge Plane Algorithms for Network Analytics and AMC need to take care of dynamically in driving management and control systems such as E2E Service or Domain Orchestrators, OSS, EMS/NMS, SDN Controllers, NFV Orchestrators, etc. via the NorthBound APIs/Interfaces.

Activity Number	Description of the Activity
4	High Level Business Goals for the Network, Service Profiles andSLAs, Network Service Definitions/Graphs, Application Intents and Profiles/Policies, are all used to generate a Network Profile consisting of information such as technical goals/objectives of the network, network service profiles, network service policies, application intents and profiles, configuration data for network elements (NEs), and other types of information and data that could be necessary to create or generate and store into the inventory in form of configuration files or model descriptions (e.g. YANG models).With the aid of a G-MBTS Translation and Network Profile Generator Tool the human operator generates Low Level Config Policies and Config-Data or Models consisting of Policies, Goals, Intents, Service Definitions, and Configuration-Data encapsulated by a Network Profile Skeleton that may be generated in a format like XML, and any additional Config-Files or Models (e.g. YANG models) required. The Knowledge Plane DEs algorithms shall operate in such a way that they are governed (by the goals and policies) or are aware of such items encapsulated by the Network Profile (i.e. the DEs take into consideration the items in the Network Profile). The Network Profile also contains Node Profiles for individual NEs (GANA nodes)that may be expressed in a format such as XML should be complemented with Node Config Files in some formats accepted by NEs' configuration agents. A Node Profile contains some configuration data and policies for low level DEs (GANA level-2 and level-3 DEs) supported by an NE (GANA node) or are meant to be loaded into an NE by the human operator. The configuration data and policies for low level DEs (GANA level-2 and level-3 DEs) supported by an NE may be initially generated by the Automated Management Tools and augmented later by the Knowledge Plane DEs before they get disseminated to the target NEs. Alternatively configuration data and policies for low level DEs (GANA level-2 and level-3 DEs) supported by an NE or to be loaded into an NE are solely generated by the Knowledge Plane DEs. A NODECONF (Node Configuration Files or Model) may contain a part that contains the Node Profile(contains data for configuring the DEs) needed by a GANA Node and vendor specific configuration options. The NODECONF may be used to govern any vendors' nodes types with the same roles or may be used to govern a specific GANA NODE. The MBTS may be used to translate the NODECONF to proprietary vendor's node configuration and communications mechanisms. See note 9.
5	The human operator then stores the generated items into ONIX while at the same time providing them as input to the Knowledge Plane DEs (Network Level DEs).
6	The human operator then accesses the Knowledge Plane DEs (Network Level DEs) to configure them with any DE specific configuration needed by the DEs if that has not already been done. For example, configuring the DEs to operate in "Closed-Loop" or in "Open-Loop"-in which case the DEs will not program (install changes) in the network by writing state (configurations) into the network NEs or sending commands into management and control systems they influence (e.g. SDN controllers) or Service or Domain Orchestrators, but the DEs would produce "Recommendations" on what could be done to improve network services performance or solve detected problems (manifestations of faults, errors, failures, overload situations, etc.).
7	The human operator then invokes the Knowledge Plane DEs (Network Level DEs) to proceed with creating additional configuration data required to create a complete Network Profile (which needs to be completed by the KP DEs and amc-MBTS) by augmenting the Network Profile Skeleton with configuration data expressed in the language (data model)used by the Knowledge Plane DEs that is agnostic to vendor specific data models used by targeted NEs and being agnostic to the management and control protocols that are then applied by the amc-MBTS to configure NEs. Knowledge Plane DEs (Network Level DEs) may also need to augment Config Files or Models (e.g. YANG models) with additional items or modify items in them due to the fact that Knowledge Plane DEs take into consideration various aspects in computing the optimal configurations in (near)-real-time and powered by reactive and predictive network analytics. The additional configuration items added to the Network Profile and the encapsulated Node Profiles include configuration data and policies for low level DEs (GANA level-2 and level-3 DEs) supported by an NE or to be loaded into an NE (see note 2).
8	Knowledge Plane DEs (Network Level DEs) retrieve any data/information required for creating additional configuration data required to create a complete Network Profile (which needs to be completed by the KP DEs and amc-MBTS)from ONIX if such data/information is available through ONIX. Knowledge Plane DEs (Network Level DEs) may also need to augment Config Files or Models with additional items or modify items in them initially generated by Automated Management Tools (as mentioned earlier).The Knowledge Plane DEs (Network Level DEs) also invoke the amc-MBTS to provide to the DEs the actual status of the targeted NEs and any other network state information that may need to be considered by the DEs before augmenting the Network Profile Skeleton with configuration data expressed in the language (data model) used by the Knowledge Plane DEs that is agnostic to vendor specific data models used by targeted NEs and being agnostic to the management and control protocols that are then applied by the amc-MBTS to configure NEs. The DEs may modify some configuration data initially generated by the Automated Management Tools. Augmenting or modifying some configuration items by the Knowledge Plane DEs is due to the fact that Knowledge Plane DEs take into consideration various aspects in computing the optimal configurations in real-time and powered by reactive and predictive network analytics.

Activity Number	Description of the Activity
9	The Knowledge Plane DEs (Network Level DEs) moves on to augment the Network Profile Skeleton with configuration data expressed in the language (data model) used by the Knowledge Plane DEs that is agnostic to vendor specific data models used by targeted NEs and being agnostic to the management and control protocols that are then applied by the amc-MBTS to configure NEs. The Knowledge Plane DEs (Network Level DEs) also add additional configuration data to some configuration files and/models as necessary. This due to the fact that Knowledge Plane DEs take into consideration various aspects in computing the optimal configurations in real-time and powered by reactive and predictive network analytics (see note 3).
10	The Knowledge Plane DEs (Network Level DEs) moves on to invoke the amc-MBTS to augment the Network Profile and some Config Files or Models with data that is specific to management and control of NEs based on the "model" of the NE that takes into account vendor specific data models for target NEs and specific management and control protocols (e.g. NetConf, SNMP, OpenFlow, etc.) that shall then be applied by the amc-MBTS to configure the NE. The amc-MBTS also adds mappings to the management and control language (data model and primitives) that the Knowledge Plane DEs employ to talk with NEs via the amc-MBTS and is translated between the two domains by the MBTS during NE(s) operations.
11	The amc-MBTS moves on to complete the Network Profile and some Config Files or Models with adding vendor-specific management data and mappings to management and control protocols that are then employed by the amc-MBTS to configure a target NE, and also mappings to the management and control language (data model and primitives) that the Knowledge Plane DEs employ to talk with NEs via the amc-MBTS and is translated between the two domains by the MBTS during NE(s) operations. The amc-MBTS then moves to store the completed Network Profile and some Config Files or Models into ONIX.
12	The amc-MBTS moves on to fetch the stored completed Network Profile (GANA Profile) and some Config Files or Models and/or any other information that may be available in ONIX and is complementary to the completed the Network Profile and some Config Files or Models (e.g. YANG Models), before extracting Node Profiles (NODE PROFs) from the Network Profile and disseminating them to NEs (GANA Nodes) together with any NE (GANA Node) configuration files and data (NODE CONFs) required to configure the target NE(s) using appropriate management and control protocols (see notes 4 and 5).
13	The amc-MBTS extracts Node Profiles from the Network Profile and disseminates them to NEs (GANA Nodes) together with any NE (GANA Node) configuration files and data required to configure the target NE(s) using appropriate management and control protocols in communicating with target NE(s) (see notes 6, 7 and 8).
14	The NETPROF should also be contain reporting information that should be generated by the Autonomics Software (the DEs) as aggregate reports of how the autonomics DEs are striving to manage and control the network and automatically fixing detected problems as well as recommendations the DEs produce for the human operator to address. The Reports may be embedded in a Network Profile by inserting the report information or may be produced as standalone files that can be consumed by the human operator and inspected. The human operator can inspect the network behaviors through the aggregate generated by the Knowledge Plane DEs and containing reports by DEs in NEs as may be necessary. The reports may be used by the Human operator to control the autonomic network.
NOTE 1: Information which sets context for the role a particular NE (GANA Node) plays in the network (based on various factors such as environment in which the NE is being considered to be operating, e.g. rural area/ urban area, very dense area, overloaded/under loaded area, B2B/B2B2C/B2C, etc.) should be included in the GANA Profile (Network Profile) Skeleton generated by the Tools before it is further augmented by KP DEs and MBTS as described in the *Activity Steps 7 through to 12* (described in this table). NOTE 2: The MBTS may be integrated as part of the SouthBound Interface of an SDN controller as described in ETSI White Paper No.16 [3]. NOTE 3: At this stage the Network Profile and some configuration data files and models are to be augmented with NE configuration data by the amc-MBTS.	

Activity Number	Description of the Activity
NOTE 4: This process of disseminating the configuration files to NEs by the MBTS may be administratively disabled by the human operator when the human operator, for some reasons, intends to leverage traditional management and control systems to use the generated Network Profiles and Configuration Files and Models in performing the configuration of NEs. For example, an E2E Service Orchestrator (see details in ETSI White Paper No.16 [3]) or Domain Orchestrator may be made to make use of a Network Profile and/or a Node Profile to drive Network Service(s) Programming via underlying management and control systems such as EMS/NMS, SDN Controller or NFV Orchestrator as discussed in ETSI White Paper No.16 [3] and other literature. As discussed in ETSI White Paper No.16 [3] an E2E Universal Service Orchestrator determines from Service Definitions with attributes that indicate whether a Service Function Node (SFN) is realized by a PNF or VNF, to determine which underlying management system to use to configure the corresponding NE (PNF or VNF). Such "attributes" should be accessible in a Node Profile. Note that an E2E Service Orchestrator or Domain Orchestrator in response to "flow service programming" requests from e.g. service fulfilment operation, makes use of Service Chains definitions associated with flow definitions (e.g. for end user traffic flows) to trigger the appropriate underlying management and control systems to perform Flow Programming (for traffic flows) in which the MBTS and its supported protocols such as OpenFlow may be invoked. Note that Knowledge Plane DEs together with amc-MBTS play a role in dynamic Network Service (re)-Programming as discussed in ETSI White Paper No.16 [3] as well as in Flow (re-)Programming as driven by SLA violations and the need for Dynamic Service Chaining as well. NOTE 5: Annex C provides further details on options on how the Knowledge Plane can be complementarily used together with other types of management and control systems in NEs configurations. In such cases in which the human operator choses to employ other types of management and control systems to configure NEs, MBTS may be constrained to disseminating only Node Profiles containing the configuration data and policies for GANA Level-2 and Level-3 DEs. NOTE 6: As described earlier, in the previous step, this process of disseminating the configuration files to NEs by the MBTS may be administratively disabled by the human operator when the human operator, for some reasons, intends to leverage traditional management and control systems to use the generated Network Profiles and Configuration Files and Models in performing the configuration of NEs. NOTE 7: Annex C provides further details on options on how the Knowledge Plane can be complementarily used together with other types of management and control systems in NEs configurations. In such cases in which the human operator choses to employ other types of management and control systems to configure NEs, MBTS may be constrained to disseminating only Node Profiles containing the configuration data and policies for GANA Level-2 and Level-3 DEs. NOTE 8: NETPROF may be applied by Knowledge Plane DEs to coordinate and orchestrate DEs decisions by adaptive policy control when multiple DEs are involved in a whole decision making process. NOTE 9: ETSI GS AFI 002 [2] contains additional insights regarding the concepts of GANA Network Profile, Node Profile (NODE PROF) and Node Configuration (NODE CONF) data structures (see note 1).	

6.6.2 Policy Based Network Governance in GANA and further insights on use of GANA Network Profiles, Node Profiles and Configuration Files and Models

NOTE: Any suitable frameworks for policy based management and associated information models for policy management and orchestration can be applied for the GANA DEs (including for dynamic policy generation and enforcements by the DEs) and also for policies encapsulated into a GANA Network Profile. Various types of policies can be employed to control DEs at the GANA levels (Level-2 to Level-4), e.g. Action Policies, Goal Policies and Utility Function Policies, and Policy-Continuum in the hierarchical DEs' management and control interactions should be taken into consideration. DEs (particularly GANA Level 3 and Level 4 DEs) should be designed in such a way as to be able to dynamically generate and apply some policies during their operations. As mentioned earlier, a comparison of the three types of policies is described in [i.60] and also the same reference ([i.60]) discusses why Utility Function Policies can be applied to specifying (by the human operator) high-level objectives for governing autonomic manager components (e.g. GANA DEs-particularly the Knowledge Plane DEs).

The ONIX may be used to publish the GANA Network Profiles and Node Profiles to Network Level-DEs and to GANA nodes, respectively, as described earlier in this clause on GANA Operations Procedures for the Human Network Operator. In complementing what has been described earlier, the Net-Level-DEs may retrieve from the ONIX their Profiles (encapsulated by the GANA Profile (GANA PROF))and process the GANAPROF to define the NODE PROFILE (NODEPROF) and may customize the NODEPROFILE with vendor specific options defined as the NODECONF (Node Configuration files). The NODEPROFILE should be sent to the GANA Node, which may then define or augment for itself the NODECONF (Node Configuration files) and publish the information into ONIX for KP DEs to access the node profiles. As described above, Tools such as OSS/BSS tools may be used to edit and disseminate a skeleton GANA Profile either through the ONIX or directly to the Network Level DEs (KP DEs).

Each DEs within the network may subscribe to ONIX to be updated with its profile if the ONIX is being employed for such purposes. The GANA Node should have an ONIX agent within the GANA Node which uses the services of the ONIX in retrieving information such as Node Profiles and in disseminating information that the GANA node may be required to publish into ONIX.

6.6.3 ONIX as Network Inventory

A GANA node may register itself to the ONIX and self-describe its capabilities and the capabilities of its associated MEs and DEs within the GANA Node and possibly with Peers GANA Nodes as well, its connections with other GANA Nodes, and with the KP. The GANA Node may describe its current configuration, if it has default configuration, fault recovery and state of its MEs and DEs within the GANA NODE. ETSI GS AFI 002 [2] contains insights on how a GANA Node should compile its Capabilities and publish them into the ONIX to enable Node and Capabilities Discovery by other nodes and by other entities such as the Knowledge Plane DEs, the OSS, and other network management and control components.

6.6.4 Auto-Configuration of DEs and MEs

A default configuration may be available in the Knowledge Plane for each DE and MEs of a GANA Node. If the GANA node has already a default configuration it may start the next steps in getting operational. The default configuration of the GANA node may be used to create, connect and test the interactions between entities: DE-DE, DE-ME within the Node, between peers GANA Nodes (their DEs and MEs) within the same domain or with its Knowledge Plane. A default policy may be used to set the default configuration of a GANA node. In case of failure the default configuration may be used to restart the GANA node. The default configuration set by provider may be defined by the GANAPROF (i.e. GANA Network Profile). With such procedure any node that registers with the Knowledge Plane may retrieve its default configuration. The KP DEs may set the default NODEPROF (GANA Node Profile) of a GANA Node only if it is authorized and has been authenticated.

7 Concluding Remarks and Perspectives on Further Work

7.1 Main Conclusion Points/Remarks

The present document has presented the GANA Reference Model for Autonomic Networking, Cognitive Networking and Self-Management for communication networks and services. The present document is based on the ETSI GS AFI 002 [2]-a document that complementarily describes the GANA. The difference being that the present document covers a summary of the GANA core concepts and includes some other aspects that are not covered in ETSI GS AFI 002 [2], while it also points the reader to the ETSI GS AFI 002 [2] for certain aspects that are covered in much more detail in the ETSI GS AFI 002 [2].Some of the important aspects the present document has reflected on include pointers to other ETSI documents (including Technical Reports documents on GANA instantiations onto various network architectures and their associated management and control architectures) that address the following aspects of additional value to readers and implementers of GANA autonomics:

- Relationships between GANA and other complementary networking paradigms:
  - SON;
  - SDN;

- NFV;
- E2E Orchestration;
- Network Analytics;
- Big-Data; and
- other paradigms (as described in ETSI White Paper No.16 [3] and other ETSI documents).

NOTE 1: There is on-going work in ETSI on Business drivers for autonomic networking and the resultant work is expected to be published in ETSI TR 103 195-1: "Autonomic network engineering for the self-managing Future Internet (AFI); Generic Autonomic Network Architecture; Part 1: Business drivers for autonomic networking" [i.70].

NOTE 2: There is on-going work in ETSI on elaborations of the Implementation Guide for GANA and the resultant work is expected to be published in ETSI TR 103 195-3 [i.71]: "Autonomic network engineering for the self-managing Future Internet (AFI); Generic Autonomic Network Architecture; Part 3: Guidelines to instantiating the GANA onto a target implementation-oriented reference architecture".

- ETSI TS 103 194 [1]: "Network Technologies (NTECH); Autonomic network engineering for the self-managing Future Internet (AFI); Scenarios, Use Cases and Requirements for Autonomic/Self-Managing Future Internet".

NOTE 3: There is on-going work in ETSI on GANA instantiation onto the BBF architecture scenarios (ETSI TR 103 473 [i.28] on Autonomicity and Self-Management in the Broadband Forum (BBF) Architectures is work in progress), and the work is also expected to provide insights on GANA integration with SDN and NFV, but in BroadBand Forum (BBF) architectures.

- ETSI TR 103 404 [i.4]: Autonomicity and Self-Management in the Backhaul and Core network parts of the 3GPP Architecture (published in October 2016). The document covers GANA instantiation onto the 3GPP Backhaul and Core Network architectures. The GANA instantiation onto the implementation-oriented reference network architecture and its management and control architecture can now be used by different players to implement proof of concept (PoCs) to showcase their autonomics solutions (based on standards) the industry is looking for, and to implement GANA autonomics in the target network architecture.
- ETSI TR 103 495 [i.6]: Autonomicity and Self-Management in Wireless Ad-hoc/Mesh Networks (published in February 2017). The document covers GANA instantiation onto the Ad-hoc/Mesh Network Architectures. The GANA instantiation onto the implementation-oriented reference network architecture and its management and control architecture can now be used by different players to implement proof of concept (PoCs) to showcase their autonomics solutions (based on standards) the industry is looking for, and to implement GANA autonomics in the target network architecture.
- Artificial Intelligence (AI) in GANA: In the present document the subject of Artificial Intelligence (AI) in GANA has been discussed with respect to how AI algorithms play a role in GANA DEs, the MBTS, Data Analytics on various entities in the network and its management and control systems (including Data Collectors), and Cognition Modules and Cognitive Algorithms in general. [i.76] provides insights on GANA in the Big-Picture on *The Impact of Artificial Intelligence and Autonomous Functions on the ETSI work-programme.*
- An ETSI launched PoC framework that can be used by different payers to respond to the call for autonomics PoCs as described in the framework available at http://ntechwiki.etsi.org/. An example of an autonomics PoC that is currently being implemented for showcase is "*5G Network Slices Autonomic Management* and*E2E Orchestration, with Closed-Loop (Autonomic) Service Assurance for the Network Slices: Using IoT Use Cases*".

NOTE 4: A GANA instantiation onto the implementation-oriented reference network architecture and its management and control architecture (e.g. the GANA instantiation produced by ETSI TC NTECH AFI WG listed above) can now be used by different players to implement proof of concept (PoCs) to showcase their autonomics solutions (based on standards) the industry is looking for, and to implement GANA autonomics in the target network architecture.

ETSI White Paper No.16 [3] describes the two categories that determine the actors or players the GANA model is addressing, namely: Suppliers (vendors) of GANA Functional Blocks (FBs); and Provider of assets required by the developers of GANA Functional Blocks (FBs). Therefore, implementers of GANA based AMC shall consider the two categories that determine the actors or players the GANA model is addressing as described in [3] to determine the business model that applies to them accordingly and derive integration requirements that need to be fulfilled by specific players.

7.2 Further Work

Regarding **Further Work**, the following are some of the aspects that should be addressed by the Standards communities by SDOs/Fora and Open Source Communities, in creating work items that should work on these aspects and any other aspects that may be identified in that process:

NOTE: ETSI should create a RACI matrix to help define responsibility demarcations on the various envisaged work items.

- *A Framework for DE-to-DE Coordination for Conflict Avoidance and Stability in Management* and*Control Operations: the required primitives and data model, etc., based on the relevant insights on the subject of synchronization and coordination of DEs (autonomic managers) described in ETSI White Paper No.16 [*3*] and other insights in literature.*

- *The Primitives/Operations on the interfaces of an DE Model that are presented in the present document are just skeletons that require further detailing towards real implementations, and additions of other primitives that may be desirable remains an open subject.*

- *Policy Control Frameworks that can be applied in GANA, within Node's internal DEs and control-loops and in the outer loop driven by Knowledge Plane DEs.*

- *Addressing the Requirements for Protocols and APIs for Enabling GANA based Autonomics, Cognitive Networking and Self-Management of Networks and Services in Evolving and Future Networks outlined in Annex B of the present document.*

- *The Formal Description of the GANA Reference Model in form of a GANA Meta-Model that describes the GANA concepts, relationships and constraints using a modelling language such as UML, MOF, or other modelling languages*. Such work on a GANA Meta-Model had been started as discussed in [i.59] and [i.72] using the GME modelling environment, and could be used as starting basis to complete the work on a GANA Meta-Model.

- *Bridging/Unifying the Emerging Unified Information Model for SDN and NFV being developed by collaborating SDOs/Fora with the GANA Model for AMC through Meta-Modeling.*

- *Methods for Knowledge Representation and Presentation to Knowledge Plane DEs. Knowledge* that can be provided as input to DEs' operations and decision-making logics may vary. It may be Knowledge derived from monitoring data (raw data) concerning events and behaviours observed/monitored in the network infrastructure-knowledge derived using analytics (Big-Data Analytics or Network Analytics). As discussed in ETSI GS AFI 002 [2] a method for Knowledge Representation and Presentation to Knowledge Plane DEs may involve Ontologies for use in representing knowledge derived through Data Analytics and Network Analytics applied to network behaviours and various types of raw data obtained from monitoring components. Such Ontologies need to be developed. Another type of knowledge that may be supplied as input to DEs' operations and decision-making logics is "expert knowledge on how to perform certain management tasks and fix problems" i.e. knowledge that may be known by human experts who may have built this knowledge through experiences in managing and resolving and troubleshooting network problems. Such knowledge may be supplied to DEs as part of their learning to enhance their own intelligence (possibly in offline mode for the DEs before they are activated into operation or possibly such input may be supplied to them while in operation (online). How to design DEs in such a way as to also be able to take into account such type of knowledge as input is an open subject.

- *Standards-anchored Open Source Projects* are now believed to help accelerate the deployment of the associated standards by the industry, and there are already a number of open source projects addressing a number of emerging networking paradigms that are complementary to the GANA autonomics paradigm. This GANA autonomics reference model specification presented in the present document should from now on help trigger the creation of an Open Source Project that should be tailored to GANA autonomics and Artificial Intelligence Algorithms for AMC, and such an open source project could be within any of the already existing projects that could be appropriate to launch an "Open GANA" open source project. Such an open source project can also help accelerate the integration of GANA and complementary technologies of SDN, NFV, E2E Orchestration and Big Data Analytics, SON for RANs (Radio Access Technologies) by developing the APIs and protocols required for the integration of these complementary paradigms as described in ETSI White Paper No.16 [3] and in the Annex B on "Requirements for Protocols and APIs for Enabling GANA based Autonomics, Cognitive Networking and Self-Management of Networks and Services in Evolving and Future Networks". This will also help foster an Open Networking Platform that can serve as a proving ground (validation) for standards underpinning the GANA and these emerging complementary technologies-combined to interwork in the same environment. Such an Open Networking Platform would also serve as an innovation enabler platform.

Annex A (normative): Summary of GANA Abstraction Levels for Self-Management/Autonomics Control-Loops

GANA reference model hierarchical level	Description
GANA Level 1: Protocol level DE (lowest level)	Relates to any Managed Entity (ME) such as a protocol or other fundamental mechanisms that may exhibit intrinsic control-loops (DE logic) and associated DE-as is the case for some of today's protocols such as the routing protocol OSPF (Open Shortest Path First), which can be considered an example of the instantiation of a protocol-level DE (though such autonomic-like feature in OSPF is not cognitive (learning and reasoning) in its operation and by design). The GANA Specification puts forward a recommendation to primarily focus on the three higher GANA levels of hierarchical control-loops (Level2 to Level4) when introducing autonomics in architectures and considers the protocol level DEs as MEs at the resources layer (along with any other fundamental MEs). The GANA hierarchy emphasizes only the three other levels which should collaboratively work together (Refer to ETSI GS AFI 002 [2]).
GANA Level 2: Function level DE	Relates to a DE for collective AMC of a group of protocols and mechanisms that are abstracted (viewed like a "bundle") by a networking or a management/control function. GANA specifies the following six function level DEs: routing management-DE; forwarding management-DE; Quality of Service management-DE; mobility management-DE; monitoring management-DE; service and application management-DE (Refer to ETSI GS AFI 002 [2] for more details on the types of DEs for this level and their associated types of MEs). The control-loop is external to the MEs subscribed to the function (by virtue of abstraction). Multiple DEs at this level are determined by the functions required of the NE.
GANA Level 3: Node level DE	Relates to a DE for AMC of those aspects that cover and restrict the behaviour of the NE as a whole, as well as the orchestration and policing of the function level. GANA Level 3 specifies the following four DEs: Security management DE, fault management DE, auto configuration and discovery DE, resilience and survivability DE. Those four DEs are collectively referred to as the Node-Main-DE. It is because such autonomic management and control functions are the superior ones within a node, as they should operate on the level that globally regulates the node and its composition. (Refer to ETSI GS AFI 002 [2], table 1 for more details on the types of DEs for this GANA level and their associated types of MEs.)
GANA Level 4: Network level DE	Relates to a DE for AMC of those aspects that cover network-wide views and the management and control of lower levels e.g. node/device levels, as well as the policing of the lower levels (e.g. node Levels). Such a DE is designed to operate in a logically centralized manner. The control-loops at this level complement lower level control loops by operating on a slower timescale (i.e. they are slower control-loops in contrast to lower level control-loops (the faster control-loops)). The network level DEs constitute the functional blocks of the Knowledge Plane, together with ONIX (Overlay Network for Information eXchange) and MBTS (Model-Based-Translation Service). (Refer to ETSI GS AFI 002 [2] for more details on the types of DEs for this level and their associated types of managed entities.)

Annex B (normative): Requirements for Protocols and APIs for Enabling GANA based Autonomics, Cognitive Networking and Self-Management of Networks and Services in Evolving and Future Networks

B.1 General Protocol and APIs Requirements Necessitated by the GANA Model as enabler for AMC when instantiated onto particular target network and management architectures

B.1.0 Overview on the Types of Protocol and API Requirements

This clause describes some High-Level Protocol Requirements and APIs (Application Programming Interfaces) Requirements necessitated by GANA Functional Blocks (FBs) - as they serve as enabler for AMC when instantiated onto particular target network and management and control architectures. The protocol requirements derive from the introduction of autonomics (GANA Functional Blocks (FBs) and Reference Points) into network architectures based on GANA instantiations onto the existing network architectures (and their associated management and control architectures) such as the 3GPP Backhaul and Core Network architectures, the BBF architectures, etc.

B.1.1 Protocol Requirements

Remark: A study should be carried out to identify any existing protocols that can be used to implement the individual Requirements captured in the tables below, identify gaps (if any) and to develop new protocols that can address the individual requirements if no existent protocol could be extended to address the particular requirements.

[Protocol Requirement]: A Protocol is required for enabling horizontal (distributed) DE-to-DE collaboration**:** Some DE algorithms may require the collaboration of a DE within an NE (GANA Node) with other DEs along an end-to-end (E2E) path in the network, though not necessarily involving hop-by-hop NEs, for a self-* operation (e.g. self-optimization) that may require the collaboration of distributed DEs along a path in the network in the orchestration and configuration of their associated MEs. The table below breaks this high level requirement into sub-requirements.

Table B.1

Requirement ID	Requirement Name	High-Level Requirement Description	Additional Information/Remarks
RQ_1	**Generic Control Protocol for DE-to-DE Horizontal Control Information Exchange and Synchronization**	A Control Protocol is required for enabling horizontal (distributed) DE-to-DE collaboration. The control protocol should define a set of a variety of selectable control semantics that may be employed by DEs in exchanging control messages and information (e.g. simple one-way or two-way or multi-party control information flow, indications of whether an acknowledgement of information reception is needed or not, solicitations for information or push/pull behaviors (which may be employed by DEs), negotiations for ME parameter value settings, negotiations for ME orchestrations or (re)-configurations, and other control information exchange that may be useful for DE-to-DE collaboration). The reason is that some DE algorithms may require the collaboration of a DE within an NE with other DEs along an end-to-end (E2E) path in the network, though not necessarily involving hop-by-hop NEs, for a self-* operation (e.g. self-optimization) that may require the collaboration of multiple distributed DEs along a path in the network for dynamic orchestration and configuration of their associated MEs. Such a control protocol should be usable for any DE in an NE whose algorithm(s) requires collaboration with other DEs along a path in the network, and even between domains if domains are involved (see Figure 6). In case of domains being involved the control protocol may take the form of a broker as illustrated in Figure 6. The same control protocol shall be applicable in implementing the reference points and their associated characteristic information defined in table 4 of ETSI GS AFI 002 [2] that are listed below: *(1)Rfp_FunctionLevelDE-to-FunctionLevelDE:* • Trust and Authentication exchange of messages and other types of messages exchanges necessary for ensuring secure communication. • Domain Type(s) to which a DE involved is bound need to be exchanged. Domain Identifier(s) of DEs hosted by entities belonging to different administrative domains need to be exchanged. • Views can be communicated by a particular Function-Level-DE to other peer Function-level-DEs on other nodes/devices, especially concerning events or issues a function of a node e.g. Routing-Function (through the Routing-Management-DE) cannot resolve by performing some action without jeopardizing network integrity (objectives). • Control Information exchange between Function-Level-DEs via the DE-2-DE interactions to achieve a "network-intrinsic management and control". Such interactions may include the notion of "compartment formation (formation of a scope for which members can communicate with each other), policies of operation and compartment management" by DE-2-DE communication in a distributed fashion.	See notes 1, 2, 3, 4 and 6

Requirement ID	Requirement Name	High-Level Requirement Description	Additional Information/Remarks
		(2)Rfp_NodeMainDE-to-NodeMainDE: • {Characteristic information communicated: Similar types of Characteristic Information as in the case of the Reference Point "*Rfp_FunctionLevelDE-to-FunctionLevelDE*". The difference being the scope for which the Characteristic Information applies i.e. this case applies to the scope of the node/device level than a particular Function-Level (lower level).} *(3)Rfp_NetworkLevelDE-to-NetworkLevelDE:* • "Views" such as Policy changes by the human operator; challenges to the network's operation from the perspective of a particular DE e.g. detected faults, threats, etc.; "views" communicated from lower-Level DEs in nodes/devices that require Net-Level-DEs to share and act upon if necessary. • Domain Type(s) to which a DE involved is bound need to be exchanged. Domain Identifier(s) of DEs hosted by entities belonging to different administrative domains need to be exchanged. • Negotiations and Synchronization of Actions and Policies (to ensure stable, conflict-free and optimized configuration of the managed entities by collaboratively and selectively scheduling actions that should be executed at a given time). Such interactions, if involving DEs in different administrative domains, shall be subject to policy agreements between the domains involved.	

NOTE 1: The reference points and their associated characteristic information defined in table 4 of the ETSI GS AFI 002 [2], which should be implemented using such a control protocol (see the list below) are referring to the DEs hosted in different network nodes/devices (virtual or physical elements): { *Rfp_FunctionLevelDE-to-FunctionLevelDE*; *Rfp_NodeMainDE-to-NodeMainDE; Rfp_NetworkLevelDE-to-NetworkLevelDE }*.

NOTE 2: The Reference Point between Network-Level-DEs is independent of the types of Network-Level-DEs and so should be considered as a common type of Reference Point between any Network-Level-DEs.

NOTE 3: For more information on the concept of Domain Type(s) and Domain Identifiers, ETSI GS AFI 002 [2] provides details in the clause on Federation. Domain identifiers are identifiers that define a network scope of administrative governance or for enforcement of a policy, and if within that scope there are technology specific qualifiers then the domain identifier may include an administrative or policy determined scope identifier part plus a technology specific part. The identifiers can be uniquely assigned by the network operator's partitioning of the network and its administrative or policy restrained boundaries.

NOTE 4: Where the interactions between the DEs involve different administrative domains, the DE interactions shall be subjected to policy agreements between the domains.

NOTE 5: A study should be carried out to identify any existing protocols that can be used to implement the Requirement, and to develop new protocols that can address the requirement if no existent protocol could be extended to address the requirement.

NOTE 6: Proposals such as IGCP [i.25] and protocols such as GRASP [i.26] need to be reviewed and extended to fulfil the requirements identified in IGCP [i.25], the reference points defined in table 4 of the ETSI GS AFI 002 [2], namely: *Rfp_FunctionLevelDE-to-FunctionLevelDE*; *Rfp_NodeMainDE-to-NodeMainDE; Rfp_NetworkLevelDE-to-NetworkLevelDE*; and also any other desirable features that may be further identified for DE-to-DE communications along a path in the network.

[Protocol Requirement]: A Protocol is required for enabling Vertical (Hierarchical) DE-to-DE collaboration:
Another possibility is that actions of an NE's DEs may also need to be synchronized by higher level autonomic behaviours coordinated by upper level DEs (outside the NE (GANA Node)) at the GANA Knowledge Plane (KP) level.

Table B.2

Requirement ID	Requirement Name	High-Level Requirement Description	Additional Information/ Remarks
RQ_2	**Control Protocol for enabling hierarchical DE-to-DE collaboration in synchronization of actions**	A Control Protocol is required for enabling hierarchical DE-to-DE collaboration in synchronization of actions by which a lower level DE intends to escalate a detected event that is beyond its scope of the set of events it is allowed to act upon without needing an approval by upper DE(s) or when the lower DE requests for approval by the upper DE(s) to go ahead and execute a tentative action the DE has determined to execute in response to a certain event subject to approval by the upper DE(s). Also, an upper DE may retrieve views from lower level DEs via the protocol, and an upper DE may enforce policies to a lower level DE that handles similar management and control aspects via the same protocol.	See notes 1, 2 and 3.
NOTE 1: This requirement is linked to a Framework for ensuring Stability of Control-Loops and associated autonomics operations in GANA as discussed in [3], [2], [i.23] to detail. NOTE 2: A study should be carried out to identify any existing protocols that can be used to implement the Requirement, and to develop new protocols that can address the requirement if no existent protocol could be extended to address the requirement. NOTE 3: The Control protocol required for enabling horizontal (distributed) DE-to-DE collaboration may be the same protocol that may be made to cover this requirement for enabling horizontal (hierarchical) DE-to-DE collaboration in synchronization of actions.			

Table B.3

Requirement ID	Requirement Name	High-Level Requirement Description	Additional Information/ Remarks
RQ_3	**Services commonly offered to *MBTS, a Network-Level-DE, a Node-Main-DE, an OSS*, or any other entity interested in the services offered by the ONIX external protocol**	An **ONIX external protocol** is required for supporting the ONIX services of enabling publish/subscribe paradigm for information, query/find operations for information retrieval by ONIX clients (e.g. DEs, nodes, other entities). The **ONIX external protocol** shall facilitate for the communication of characteristic information that may be exchanged between an MBTS, or a Network-Level-DE, or a GANA Node-Main-DE, or an OSS/BSS that exchanges such information with ONIX via the following Reference Points defined in table 4 of ETSI GS AFI 002 [2], respectively, namely***: Rfp_ModelBasedTranslationService-to-ONIX; Rfp_NetworkLevelDE-to-ONIX-System;Rfp_NodeMainDE-to-ONIX-System****; and* ***Rfp_OSS-to-ONIX-System,*** *all of which convey the following characteristic information:* • **Trust and Authentication** exchange of messages and other types of messages exchanges necessary for ensuring secure communication. • **Operations/Messages for Storing and Retrieving Information from the ONIX system**. *[For example, the MBTS can use the publish/subscribe services of the ONIX that enable Advanced Auto-Discovery of Information and Resources, to retrieve Information about Network Elements/Nodes, such as Capability Description Models of individual nodes/devices, self-advertised/published by an individual node/device upon initialization. Capability Models of a node/device include technological features supported, including management protocols supported and Information about Managed Objects (MOs) of the technologies (e.g. protocols, etc.). Capability Models may include apart from technological features, vendor information].*	See notes 1, 2 and 3.

Requirement ID	Requirement Name	High-Level Requirement Description	Additional Information/ Remarks
NOTE 1: Any entity intending to be discovered or have its capabilities and point of attachment in the network to be discovered by other entities can publish such information into the ONIX but subject to administrative policy. NOTE 2: Any entity (e.g. a DE) interested in subscribing for receiving information when it becomes available in ONIX or to perform query/find operations for information retrieval from ONIX can use such ONIX services but subject to administrative policy. NOTE 3: A study should be carried out to identify any existing protocols that can be used to implement the Requirement, and to develop new protocols that can address the requirement if no existent protocol could be extended to address the requirement.			

Table B.4

Requirement ID	Requirement Name	High-Level Requirement Description	Additional Information/Remarks
RQ_4	**ONIX internal protocol for building a federated ONIX system**	An **ONIX internal protocol** is required for enabling ONIX information servers to work together as a federated single system (from external ONIX services point of view) in responding to a service request for either publishing information into the ONIX by a client, subscribing to receive information from ONIX, querying and retrieving information from ONIX. An external entity requesting for a service only needs to target one of the ONIX servers accessible and the targeted ONIX server responses with required information by coordinating with other ONIX servers using the ONIX internal protocol to fetch for the required information if the information is not stored on the targeted server but may be stored on another server.	More details on the ONIX services and internal coordination of ONIX servers using e.g. DHT (Distributed Hash Tables) may be found in ETSI GS AFI 002 [2]. See note.
NOTE: A study should be carried out to identify any existing protocols that can be used to implement the Requirement, and to develop new protocols that can address the requirement if no existent protocol could be extended to address the requirement.			

Table B.5

Requirement ID	Requirement Name	High-Level Requirement Description	Additional Information/Remarks
RQ_5	**Supporting any network management or control protocol as a plugin to the Southbound Interface of an MBTS**	While the MBTS (Model-Based Translation Service) Libraries that implement the GANA Reference Point "*Rfp_ModelBasedTranslationService-to-NodeMainDE"* that is a refinement of the "Rfp_NetworkLevelDE-to-NodeMainDE" are meant to support multiple network management and control protocols on the MBTS's Southbound Interface as plugins, any newly introduced network management or control protocol shall be supported as a plugin to the Southbound Interface of an MBTS	See note.
NOTE: The Reference Points in question are defined in table 4 of ETSI GS AFI 002 [2].			

B.1.2 API Requirements

The types of APIs (Application Programming Interfaces) being described means programmatic interfaces that should be implemented by the particular entity being referred to-by providing a means for a client entity to invoke an operation and/or provide input through procedures/methods calls on the API.

Remark: A study should be carried out to identify any existing APIs that can be used to implement the individual Requirements captured in the Tables below, and to develop new APIs that can address the individual requirements if no existent API could be extended to address the particular requirements.

Table B.6

Requirement ID	Requirement Name	High-Level Requirement Description	Additional Information/ Remarks
RQ_6	**MBTS Northbound API specially designed to enable a Programming Model for KP DEs that is vendor technology agnostic and network management or control protocol agnostic**	The MBTS (Model-Based Translation Service) Libraries shall provide a Northbound API specially designed to enable a Programming Model for KP DEs that is vendor technology agnostic and network management or control protocol agnostic, as the MBTS would do the mediation and translations between the two domains (Knowledge Plane DEs language domain vis-à-vis network infrastructure management and control language domain). The API shall consist of primitives to be used by Network-Level-DEs to: • (re)-configure network service functions/nodes (NEs) via the MBTS, using Rfp_NetworkLevelDE-to-NodeMainDE reference point and Rfp_ModelBasedTranslationService-to-NodeMainDE reference point; • send dynamically generated policies (via Rfp_NetworkLevelDE-to-NodeMainDE reference point) to a lower level DE(s) (GANA level 3 and level 2 DEs in NEs) that policy the lower level DE(s); • send a command (via Rfp_NetworkLevelDE-to-NodeMainDE reference point) to a lower level DE(s) (GANA level 3 and level 2 DEs in NEs) requesting the lower level DE to orchestrate or change configuration of an ME(s) or its parameters; • send a synchronization message (via Rfp_NetworkLevelDE-to-NodeMainDE reference point) to a lower level DE(s) (GANA level 3 and level 2 DEs in NEs) indicating an approval for the lower level DE to execute a change on an ME(s) in response to synchronization message issued by the lower level DE prior; • receive messages (which include responses to commands issued by DEs prior, or messages conveying information such as aggregate KPIs) coming from GANA nodes (NEs) and being relayed through the MBTS. The API's primitives used by KP DEs shall enable to transfer structured data to be send by the KP DEs to their targets in at least the following forms: XML, YANG.	See notes 1 and 2.
NOTE 1: The interface enables developers of Network Level DEs and their associated algorithms to use a common programming model that is vendor technology agnostic and network management or control protocol agnostic-thereby making it simpler to innovate and program DE operations. NOTE 2: The GANA reference points: Rfp_NetworkLevelDE-to-NodeMainDE; Rfp_ModelBasedTranslationService-to-NodeMainDE; are defined in table 4 of ETSI GS AFI 002 [2].			

Table B.7

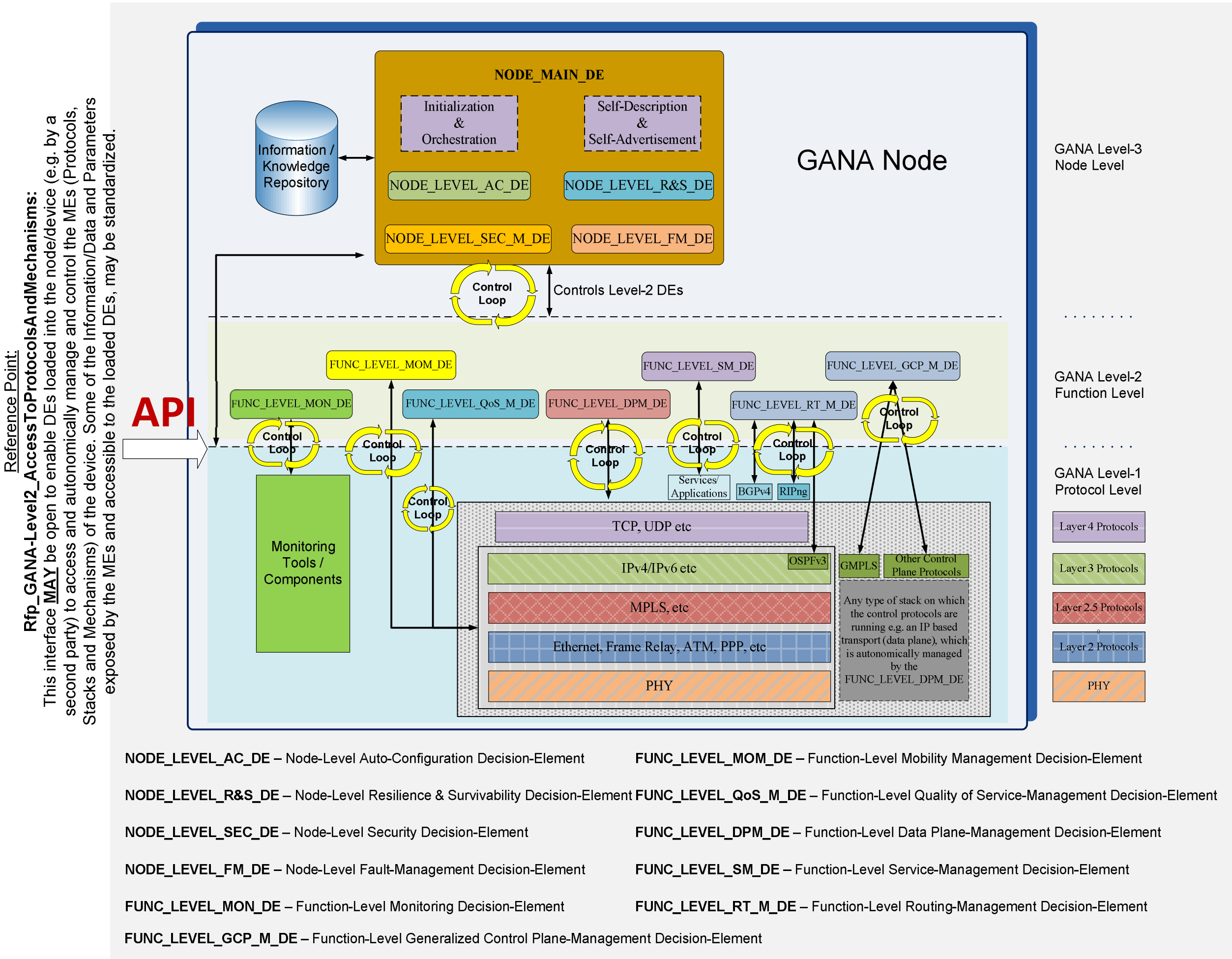
Requirement ID	Requirement Name	High-Level Requirement Description	Additional Information/Remarks
RQ_7	**APIs for implementing the interfaces of a Decision Element (DE) as Autonomic Function(AF)**	Assuming that the interfaces of a Decision Element (DE) Model are implemented as APIs, then a set of APIs should be defined in such a way as to implement the Interfaces of a Decision Element (DE) Model and their associated primitives, all of which are defined in clause 9.11 of ETSI GS AFI 002 [2]. The bi-directional natured of communication on the interfaces of the DE Model shall be realized by the APIs, e.g. using any appropriate means that enable an entity on either side of the API to call methods that should be called on the other party. The further detailing, specification and implementation of the primitives of the management interface of a DE should also be aligned with the way they can be used by the human operator through the GANA Network Governance Interface (refer to Reference Point: *Rfp_GANA Network Governance Interface-IRP*) through which the operator provides input to the autonomic network as well as activating or de-activating DEs through the appropriate primitives they implement on their management interface (see Decision Element (DE) Model in ETSI GS AFI 002 [2]).	The implementation of the southbound interfaces of a DE model meant to be a Network Level DE (in the Knowledge Plane (KP)) should be aligned with the Programming Model for KP DEs and the implementation of the MBTS Northbound API. See note.
NOTE: For DE-to-DE interactions involving DEs hosted in different network nodes a protocol could be designed for that (refer to the requirement concerning a Generic Control Protocol for DE-to-DE Horizontal Control Information Exchange and Synchronization).			

Table B.8

Requirement ID	Requirement Name	High-Level Requirement Description	Additional Information/Remarks
RQ_8	**GANA node internal API for enabling GANA Level-3 and Level-2 DE to access and perform management and control of protocol stacks and other resources as Managed Entities (MEs) at the resource layer**	A GANA node internal API should be defined in such a way as to implement the Reference Point defined in table 4 of ETSI GS AFI 002 [2], namely***: Rfp_GANA-Level2- AccessToProtocolsAndMechanisms,*** *which is meant to enable GANA Level-3 and Level-2 DE to access and perform management & control of protocol stacks and other resources as Managed Entities (MEs) at the resources layer. The API exposes the management interfaces (consisting of sensory parts and effector parts) of the MEs to the DEs*.	See note.
NOTE: The required API is indicated on the diagram in Figure B.1 of the present document. The interface enables innovation in autonomics by enabling DE developers to load and replace DEs of varying decision-making capabilities into nodes.			

Table B.9

Requirement ID	Requirement Name	High-Level Requirement Description	Additional Information/Remarks
RQ_9	**An API for enabling a Network-Level- DE to read and write data from/to a Data Storage**	An API should be defined in such a way as to implement the Reference Point defined in table 4 of ETSI GS AFI 002 [2] , namely: ***Rfp_NetworkLevelDE-to-Data_Storage***	The Data storage may be storage meant for any type of data, even raw data that may either be useful for certain DE algorithms or the storage may be used by DEs to store some data temporarily. Otherwise the ONIX and Monitoring Data Collectors may be used instead.



NOTE: This diagram is same as Figure 19 in ETSI GS AFI 002 [2] that is better readable.

Figure B.1: The API that enables DEs to access and configure protocol stacks and OSI layer 7 or TCP/IP application layer applications and other types of managed resources or mechanisms

B.2 Requirements for APIs Necessitated by the Unified Architecture for ETSI GANA Knowledge Plane, SDN NFV, E2E Orchestration, Big-Data driven analytics for AMC

This clause describes requirements for APIs necessitated by the Unified Architecture for ETSI GANA Knowledge Plane, SDN NFV, E2E Orchestration, Big-Data driven analytics for Service Orchestration and/or for AMC, i.e. the harmonisation architecture that emerged from the Joint SDOs/Fora Industry Harmonisation Initiative on Unified Standards and architectures for these emerging paradigms (as described in [3]). The APIs of focus here are those required by the GANA Knowledge Plane in the integration with the other paradigms, as the Knowledge Plane is meant to be the realm for enabling innovation and implementation of the complex algorithms for analytics-driven autonomics as the enabler for intelligence in management and control of networks and services. The APIs could be developed in SDOs/Fora such as Tele-Management Forum (TMF). The API based interfaces listed below are complemented by those that are protocol based as listed earlier in the present document. The APIs means programmatic interfaces that should be implemented by the particular entity being referred to (should provide a means for a client entity to invoke an operation and/or provide input through procedures/methods calls on the API).

Remark 1: A particular Reference Point in the Unified Architecture for ETSI GANA Knowledge Plane, SDN NFV, E2E Orchestration, Big-Data driven analytics for Service Orchestration and/or for AMC, may be implemented the way traditional interfaces are implemented or as API(s) or Protocol(s). In all the cases there is a need for defining a Data Model and Operations Model. The tables below assume an API based implementation for each of the Reference Points and provide insights into how each of such APIs could look like in terms of the services that the API could provide.

Remark 2: A study should be carried out to identify any existing APIs that can be used to implement the individual Requirements captured in the Tables below, and to develop new APIs that can address the individual requirements if no existent API could be extended to address the particular requirements.

Table B.10

Requirement ID	Requirement Name: Remark: Assuming an API based implementation of a Reference Point, an API name is derived from the Reference Point name by replacing "-IRP" with "-API" and vice-versa).	High Level Description of the API	Additional Information/Remarks
RQ_10	**Rfp_GANA Network Governance Interface-API**	An interface (possibly a GUI (Graphical User Interface) through which the human operator provides input to the autonomic network in form of GANA Network Profile. The GANA Knowledge Plane DEs use the Network Profile in configuring themselves and also lower level DEs use the Node Profile parts of the Network Profile to extract configuration data to apply for themselves and the Managed Entities (MEs) at the resources layer of the GANA node. The Network Profile shall be kept consistent and updated to always reflect the network configuration at any particular point in time.	See notes 1, 2 and 3.

Requirement ID	Requirement Name: Remark: Assuming an API based implementation of a Reference Point, an API name is derived from the Reference Point name by replacing "-IRP" with "-API" and vice-versa).	High Level Description of the API	Additional Information/Remarks
		If for some reason the network protocols/stacks and service functions/nodes are initially configured using orchestration and management systems such as OSS, NMS/EMS, SDN controller, etc., then the Knowledge Plane DEs shall not perform the same configuration unnecessarily but shall update the Network Profile with newly added or modified configuration state data when they dynamically re-configure Managed Entities (MEs) in network service functions/nodes. When the human operator has chosen to perform initial configuration of network protocols/stacks and service functions/nodes using orchestration and management and control systems, and not using the Knowledge Plane, then the governance interface shall disable the KP DEs from performing initial configuration of network protocols/stacks and service functions/nodes. Through the governance interface KP DEs shall be activated or deactivated by the human operator. The interface shall also enable the human operator to ensure that updates to Network Profiles by KP DEs and by Tools the populate Network profiles with configuration data are not resulting in conflicting and or invalid state of the Network Profile data.	
RQ_11	**Rfp_OSS-API-for Integration with the GANA Knowledge Plane DEs**	The API should enable the GANA Knowledge Plane DEs to receive in real-time network and services related KPIs and events that the OSS gathers. The Knowledge Plane (KP) may subscribe to receive such information of interest, at a certain rate, through the API. As such the OSS should share knowledge it may gather about events affecting network and services and KPIs that the OSS can be configured to export into the Knowledge Plane, because DEs in the Knowledge Plane are supposed to be the ones that implement the more complex analytics/autonomics algorithms for self-adaptation and re-programming of network services/resources/parameters. The API should enable GANA Knowledge Plane DEs to send commands into the OSS if the OSS provides a capability of configuring the network via the OSS directly.	See notes 4, 5, 6, 7 and 8.

Requirement ID	Requirement Name: Remark: Assuming an API based implementation of a Reference Point, an API name is derived from the Reference Point name by replacing "-IRP" with "-API" and vice-versa).	High Level Description of the API	Additional Information/Remarks
		For example, commands by DEs aimed at (re)-programming network services. These interactions may be subjected to a translation service to/from the language/data model used by the Knowledge Plane DEs. To avoid conflicts in network resources configuration by enforcing synchronization of network services configuration/programming with the Knowledge Plane, the OSS shall synchronize with the Knowledge Plane's Coordination Function (could be implemented by the Network Level Auto-Configuration DE) whenever a configuration operation has been triggered on the OSS from elsewhere, and the synchronization should happen through the API.	
RQ_12	**Rfp_Universal (E2E) Service Orchestrator-API-for Integration with the GANA Knowledge Plane DEs**	The API should enable the GANA Knowledge Plane DEs to receive in real-time network and services related KPIs and events that the Universal (E2E) Service Orchestrator may know. The Knowledge Plane (KP) may subscribe to receive such information of interest, at a certain rate, through the API. As such the Universal (E2E) Service Orchestrator should share knowledge it may gather about events affecting network and services and KPIs that the Universal (E2E) Service Orchestrator can be configured to export into the Knowledge Plane, because DEs in the Knowledge Plane are supposed to be the ones that implement the more complex analytics/autonomics algorithms for self-adaptation and re-programming of network services/resources/parameters. The API should enable GANA Knowledge Plane DEs to send commands into the Universal (E2E) Service Orchestrator when DEs have deduced that the Universal (E2E) Service Orchestrator needs to perform an operation such as orchestrating a new service node/function or re-orchestrating a service that may have suffered degradation due to manifestations of challenges (e.g. faults/errors/failures/security-threats) in the network's operation.	See notes 9, 10 and 11.

Requirement ID	Requirement Name: Remark: Assuming an API based implementation of a Reference Point, an API name is derived from the Reference Point name by replacing "-IRP" with "-API" and vice-versa).	High Level Description of the API	Additional Information/Remarks
		Also, commands by DEs aimed at (re)-programming network services, if there are commands that need to be issued to the Orchestrator, otherwise commands are issued to the MBTS towards network infrastructure by default. These interactions may be subjected to a translation service to/from the language/data model used by the Knowledge Plane DEs. When the Orchestrator performs an orchestration of a service, the orchestrated service functions/nodes and associated configuration state, and indications of the management function/component instance (e.g. EM, EMS, NMS, SDN-Controller, NFVO, etc.) through which the orchestration happened, shall all be indicated to the Knowledge Plane's DEs (particularly the Network Level Auto-Configuration DE that coordinates the other DEs), through the API. To avoid conflicts in network resources configuration by enforcing synchronization of network services configuration/programming with the Knowledge Plane, the Orchestrator shall synchronize with the Knowledge Plane's Coordination Function (could be implemented by the Network Level Auto-Configuration DE) whenever a configuration operation has been triggered on the Orchestrator from elsewhere, and the synchronization should happen through the API.	
RQ_13	**Rfp_NFV-Orchestrator API-for Integration with the GANA Knowledge Plane DEs**	The API should enable the GANA Knowledge Plane DEs to receive in real-time network and services related KPIs and events that the NFV Orchestrator may gather. The Knowledge Plane (KP) may subscribe to receive such information of interest, at a certain rate, through the API. As such the NFV Orchestrator should share knowledge it may gather about events affecting virtualised network and services and KPIs that the NFV Orchestrator can be configured to export into the Knowledge Plane, because DEs in the Knowledge Plane are supposed to be the ones that implement the more complex analytics/autonomics algorithms for self-adaptation and re-programming of network services/resources/parameters.	If desirable for some reason, KPIs and events gathered by other components of the MANO (e.g. VNFManager, VIM) may also be shared to the Knowledge Plane DEs, especially if such information could be useful for some Knowledge Plane DEs' autonomics algorithms. See notes 12, 13 and 14.

Requirement ID	Requirement Name: Remark: Assuming an API based implementation of a Reference Point, an API name is derived from the Reference Point name by replacing "-IRP" with "-API" and vice-versa).	High Level Description of the API	Additional Information/Remarks
		The API should enable GANA Knowledge Plane DEs to send commands into the NFV Orchestrator when DEs have deduced that the NFV Orchestrator needs to perform an operation such as orchestrating a new service node/function or re-orchestrating a service that may have suffered degradation due to manifestations of challenges (e.g. faults/errors/failures/security-threats) in the network's operation. Also, commands by DEs aimed at (re)-programming network services if there are commands that need to be executed on the Orchestrator, otherwise by default (re)-configuration commands may be issued to the service functions/nodes of target through the MBTS. These interactions may be subjected to a translation service to/from the language/data model used by the Knowledge Plane DEs. When the Orchestrator performs an orchestration of a service, the orchestrated service functions/nodes and associated configuration state, and indications of the management function/component instance (Config-Data Manager, VIM) through which the orchestration happened, shall all be indicated to the Knowledge Plane's DEs (particularly the Network Level Auto-Configuration DE that coordinates the other DEs), through the API. To avoid conflicts in network resources configuration by enforcing synchronization of network services configuration/programming with the Knowledge Plane, the Orchestrator shall synchronize with the Knowledge Plane's Coordination Function (could be implemented by the Network Level Auto-Configuration DE) whenever a configuration operation has been triggered on the Orchestrator from elsewhere, and the synchronization should happen through the API.	
RQ_14	**Rfp_Big-Data-Analytics-App-API-for Integration with the GANA Knowledge Plane DEs**	For a Big-Data-Analytics-Application that cannot be evolved to be implemented and run as a DE in the Knowledge Plane, the application should provide an API that should enable the GANA Knowledge Plane DEs to receive in real-time network and services related KPIs and events that the Big-Data-Analytics-Application gathers.	See notes 15 and 16.

Requirement ID	Requirement Name: Remark: Assuming an API based implementation of a Reference Point, an API name is derived from the Reference Point name by replacing "-IRP" with "-API" and vice-versa).	High Level Description of the API	Additional Information/Remarks
		The Knowledge Plane (KP) may subscribe to receive such information of interest, at a certain rate, through the API. As such the Big-Data-Analytics-Application should share knowledge it may gather about events affecting network and services and KPIs that the Big-Data-Analytics-Application can be configured to export into the Knowledge Plane, because DEs in the Knowledge Plane are supposed to be the ones that implement the more complex analytics/autonomics algorithms for self-adaptation and re-programming of network services/resources/parameters. The API should enable GANA Knowledge Plane DEs to send commands into the Big-Data-Analytics-Application to coordinate its operation.	
RQ_15	**Rfp_SDN Controller NorthBound API-for Integration with the GANA Knowledge Plane**	The API (Northbound API of SDN controller) should enable the GANA Knowledge Plane DEs to receive in real-time network and services related KPIs and events that the SDN Controller gathers. The Knowledge Plane (KP) may subscribe to receive such information of interest, at a certain rate, through the API. As such the SDN Controller should share knowledge it may gather about events affecting network and services and KPIs that the SDN Controller can be configured to export into the Knowledge Plane, because DEs in the Knowledge Plane are supposed to be the ones that implement the more complex analytics/autonomics algorithms for self-adaptation and re-programming of network services/resources/parameters. The API should enable GANA Knowledge Plane DEs to send commands into the SDN Controller if the SDN controller provides a desired capability of configuring the network via the SDN Controller directly. For example, commands by DEs aimed at (re)-programming network services and traffic flows. Through the integration of the MBTS libraries as the southbound interface of the SDN controller, these interactions shall be subjected to a translation service to/from the language/data model used by the Knowledge Plane DEs. The MBTS Libraries shall also be directly accessible for direct interactions between the MBTS and Knowledge Plane DEs.	See notes 17 and 18.

Requirement ID	Requirement Name: Remark: Assuming an API based implementation of a Reference Point, an API name is derived from the Reference Point name by replacing "-IRP" with "-API" and vice-versa).	High Level Description of the API	Additional Information/Remarks
		The MBTS Libraries may also be exposed to the Knowledge Plane DEs via the NorthBound API of the SDN controller as a wrapper API. This particularly applies to the GANA integration with SDN controller as standalone entities, in contrast to the option that involves integration of the Knowledge Plane DEs as loadable modules of the SDN controller. To avoid conflicts in network resources configuration by enforcing synchronization of network services configuration/programming with the Knowledge Plane, the SDN Controller shall synchronize with the Knowledge Plane's Coordination Function (could be implemented by the Network Level Auto-Configuration DE) whenever a configuration operation has been triggered on the SDN Controller from elsewhere, and the synchronization should happen through the API.	
RQ_16	**Rfp_Monitoring Data Collector-API for Integration with the GANA Knowledge Plane**	The API provides streaming services on knowledge supply to the Knowledge Plane DEs in real-time. The knowledge is created by cognitive algorithms that run on the monitoring data collector by performing analytics on monitoring data collected from the network to derive knowledge, represent and present (by streaming) the knowledge in a form in which the DEs can consume the knowledge based on the way and form the MBTS towards network infrastructure also presents knowledge to the Knowledge Plane DEs. The DEs' autonomics algorithms then perform decision-making on what do to adapt the network services/resources/parameters to achieve desired objectives by dynamically policing lower level DEs at lower abstraction levels, down to the DEs at network function/node level, and/or by issuing commands into the functional blocks (e.g. OSS, Orchestrators, SDN Controllers, MBTS, etc.) that then further perform a corrective operation to adapt (orchestrate, mitigate, remediate, re-configure) the network services to achieve objectives determined by the Knowledge Plane DEs or to repair/heal any degraded services.	See notes 19 and 20.
RQ_17	**Rfp_EM-API for Integration with the GANA Knowledge Plane**	The API should be similar to the OSS-API-for Integration with the GANA Knowledge Plane DEs.	See note 21.

Requirement ID	Requirement Name: Remark: Assuming an API based implementation of a Reference Point, an API name is derived from the Reference Point name by replacing "-IRP" with "-API" and vice-versa).	High Level Description of the API	Additional Information/Remarks
RQ_18	**Rfp_EMS-API for Integration with the GANA Knowledge Plane**	The API should be similar to the OSS-API-for Integration with the GANA Knowledge Plane DEs.	See note 22.
RQ_19	**Rfp_NMS-API for Integration with the GANA Knowledge Plane**	The API should be similar to the OSS-API-for Integration with the GANA Knowledge Plane DEs.	See note 23.
RQ_20	**Rfp_VNFM-API for Integration with the GANA Knowledge Plane**	The API should enable the GANA Knowledge Plane DEs to receive in real-time VNFs related KPIs and events that the VNFM (Virtual Network Function Manager) may be configured to export to the Knowledge Plane.	See note 24.
RQ_21	**Rfp_VIM-APIfor Integration with the GANA Knowledge Plane**	The API should enable the GANA Knowledge Plane DEs to receive in real-time Virtual Infrastructure related KPIs and events that the VIM (Virtual Infrastructure Manager) may be configured to export to the Knowledge Plane.	See note 25.

NOTE 1: The orchestration and configuration of DEs with the necessary configuration data/profiles they require should be performed by human operator using appropriate tools on the governance interface and information about target network service functions/nodes that should be configured. Automation of the process can be performed by the Network Level Auto-Configuration DE in the Knowledge Plane, by orchestrating DEs, down to low level DEs required in particular network service function/nodes it is made aware of, and providing the DEs with pointers to configuration data/profiles they require.

NOTE 2: After initial configuration of network services functions/nodes has happened (whether performed using orchestrators, OSS, SDN controllers, NMS/EMS, etc., or through KP DEs) the Knowledge Plane is supposed to then take care of all aspects related to Closed-Loop (Autonomic) Service Assurance, while relying on all data/information/knowledge sources required.

NOTE 3: While it is desirable that in the long term KP DEs should (re)-configure the network service functions/nodes through the MBTS, developers of DEs may need to have the flexibility and freedom (in the interim) to configure network service functions/nodes through the means that may be available in the management and control systems that may be available if MBTS based solution may not yet be available.

NOTE 4: Applies to scenario in which the OSS is a standalone entity and not the scenario in which Knowledge Plane DEs are integrated as modules of the OSS.

NOTE 5: In the evolution of network and services management in the era of network virtualisation and software'rization with NFV/SDN, it is possible that the functionalities of the OSS (as it is known today) may be taken over by E2E Service Orchestrators and the Knowledge Plane.

NOTE 6: Developers of Knowledge Plane DEs may exploit the capabilities available on the OSS API to design their autonomics (DEs) algorithms accordingly. Knowledge Plane DEs may determine during their operations what needs to be retrieved from the OSS and/or what needs to be triggered on the OSS via the API.

NOTE 7: After initial configuration of network services functions/nodes has happened (whether performed using orchestrators, OSS, SDN controllers, NMS/EMS, etc., or through KP DEs) the Knowledge Plane is supposed to then take care of all aspects related to Closed-Loop (Autonomic) Service Assurance, while relying on all data/information/knowledge sources required.

NOTE 8: The Events and KPIs should also be stored by the OSS at a configurable rate of data/information/knowledge store operation and configurable data/info volume into the ONIX as historical traces that shall further be subjected to a configurable data/info retention period in ONIX. DEs in the Knowledge Plane may make use of historical traces or historical contexts in their operations. The data/info/knowledge type stored into ONIX may first undergo filtering before being stored in ONIX.

NOTE 9: After initial configuration of network services functions/nodes has happened (whether performed using orchestrators, OSS, SDN controllers, NMS/EMS, etc., or through KP DEs) the Knowledge Plane is supposed to then take care of all aspects related to Closed-Loop (Autonomic) Service Assurance, while relying on all data/information/knowledge sources required.

Requirement ID	Requirement Name: Remark: Assuming an API based implementation of a Reference Point, an API name is derived from the Reference Point name by replacing "-IRP" with "-API" and vice-versa).	High Level Description of the API	Additional Information/Remarks
NOTE 10: The orchestration and configuration of DEs with the necessary configuration data/profiles they require should be performed by human operator using appropriate tools on the governance interface and information about target network service functions/nodes that should be configured. Automation of the process can be performed by the Network Level Auto-Configuration DE in the Knowledge Plane, by orchestrating DEs, down to low level DEs required in particular network service function/nodes it is made aware of, and providing the DEs with pointers to configuration data/profiles they require. NOTE 11: The Events and KPIs should also be stored by the Service Orchestrator at a configurable rate of data/information/knowledge store operation and configurable data/info volume into the ONIX as historical traces that shall further be subjected to a configurable data/info retention period in ONIX. DEs in the Knowledge Plane may make use of historical traces or historical contexts in their operations. The data/info/knowledge type stored into ONIX may first undergo filtering before being stored in ONIX. NOTE 12: After initial configuration of network services functions/nodes has happened (whether performed using orchestrators, OSS, SDN controllers, NMS/EMS, etc., or through KP DEs) the Knowledge Plane is supposed to then take care of all aspects related to Closed-Loop (Autonomic) Service Assurance, while relying on all data/information/knowledge sources required. NOTE 13: The orchestration and configuration of DEs with the necessary configuration data/profiles they require should be performed by human operator using appropriate tools on the governance interface and information about target network service functions/nodes that should be configured. Automation of the process can be performed by the Network Level Auto-Configuration DE in the Knowledge Plane, by orchestrating DEs, down to low level DEs required in particular network service function/nodes it is made aware of, and providing the DEs with pointers to configuration data/profiles they require. NOTE 14: The Events and KPIs should also be stored by the NFV Orchestrator at a configurable rate of data/information/knowledge store operation and configurable data/info volume into the ONIX as historical traces that shall further be subjected to a configurable data/info retention period in ONIX. DEs in the Knowledge Plane may make use of historical traces or historical contexts in their operations. The data/info/knowledge type stored into ONIX may first undergo filtering before being stored in ONIX. NOTE 15: Big-Data-Analytics Application for management and control of network services and resources should rather be implemented as Knowledge Plane DEs. Big-Data Analytics Application that may have been designed for analytics-driven service orchestration and not necessarily for management and control operations need to expose an API that enables the Knowledge Plane to drive then or to synchronize operations that re-program the network services/resources. NOTE 16: The Events and KPIs should also be stored by the Big-Data Application at a configurable rate of data/information/knowledge store operation and configurable data/info volume into the ONIX as historical traces that shall further be subjected to a configurable data/info retention period in ONIX. DEs in the Knowledge Plane may make use of historical traces or historical contexts in their operations. The data/info/knowledge type stored into ONIX may first undergo filtering before being stored in ONIX. NOTE 17: After initial configuration of network services functions/nodes has happened (whether performed using orchestrators, OSS, SDN controllers, NMS/EMS, etc., or through KP DEs) the Knowledge Plane is supposed to then take care of all aspects related to Closed-Loop (Autonomic) Service Assurance, while relying on all data/information/knowledge sources required. NOTE 18: The Events and KPIs should also be stored by the SDN Controller at a configurable rate of data/information/knowledge store operation and configurable data/info volume into the ONIX as historical traces that shall further be subjected to a configurable data/info retention period in ONIX. DEs in the Knowledge Plane may make use of historical traces or historical contexts in their operations. The data/info/knowledge type stored into ONIX may first undergo filtering before being stored in ONIX. NOTE 19: The Events and KPIs should also be stored by the Monitoring Data Collector at a configurable rate of data/information/knowledge store operation and configurable data/info volume into the ONIX as historical traces that shall further be subjected to a configurable data/info retention period in ONIX. DEs in the Knowledge Plane may make use of historical traces or historical contexts in their operations. The data/info/knowledge type stored into ONIX may first undergo filtering before being stored in ONIX. NOTE 20: The Knowledge supplied by Monitoring Data Collectors to the Knowledge Plane is complemented by Knowledge supplied to the Knowledge Plane DEs by other knowledge sources such as OSS, Orchestrators, SDN controllers and MBTS. NOTE 21: The EM function is an Element Management Function (could be virtualised) in ETSI NFV MANO. NOTE 22: The EMS is a traditional Element Management System that may be targeted for integration with the Knowledge Plane. NOTE 23: The NMS is a traditional Network Management System that may be targeted for integration with the Knowledge Plane.			

Requirement ID	Requirement Name: Remark: Assuming an API based implementation of a Reference Point, an API name is derived from the Reference Point name by replacing "-IRP" with "-API" and vice-versa).	High Level Description of the API	Additional Information/Remarks
NOTE 24: Autonomics introduced and implemented at the ETSI NFV MANO VNFM level by an embedded DE should help limit the events and data that may need to be communicated to the Knowledge Plane DEs. NOTE 25: Autonomics introduced and implemented at the ETSI NFV MANO VIM level by an embedded DE should help limit the events and data that may need to be communicated to the Knowledge Plane DEs.			

Annex C (normative): Operations Guide for GANA-Empowered AMC and Autonomic Networks

C.0 A note on the Availability of the Elaborate Implementation Guide for GANA

NOTE: There is on-going work in ETSI on elaborations of the Implementation Guide for GANA and the resultant work is expected to be published in ETSI TR 103 195-3 [i.71] that is expected to complement this Guide provided in Annex C.

C.1 Automated Management Tool-Chains and their Interface with the GANA Governance Interface

NOTE: This clause provides details that complement insights provided in the clause on GANA Operations Procedures for the Human Network Operator, by providing further details on options on how the GANA Knowledge Plane can be complementarily used together with other types of management and control systems in NEs configurations.

The ETSI White Paper No.16 [3] describes how "Automated Management" and "Autonomic Management and Control (AMC)" are supposed to complement each other and be interworked. The human operator should be equipped with Tool-Chains for Automated Network and Services Management, and such Tool-Chains may include OSS/BSS systems. The Tool-Chains for Automated Network and Services Management interface with GANA Governance, as the human operator uses the Tool-Chains to create inputs to the autonomic management andcontrol software (Knowledge Plane and DEs in the network infrastructure NEs) that should be edited and validated by the human operator, and then consumed by GANA Knowledge Plane and Network Infrastructure Elements (NEs) - (The Producer andConsumer model):

- **Main Tools/Tool-Chains on the Governance Interface (but not limited to):**
  - Editors for various items and definitions, including High-Level Network Objectives/Goals, and Intents for applications, etc.
  - Config Data (Configuration Data) Generators.
  - Policy Generators andPolicy Validators (against conflicts and instantaneous network/resource capabilities): Policy Generators generate Policies from High-Level Network Objectives/Goals and Intents before they undergo Policy Validation.
  - Etc.
- **What kind of Inputs should the Tools in the Tool Chain produce for GANA Knowledge Plane and Network Infrastructure Elements (NEs):**
  - Config Data and Config Data Models should be generated by Tools or Management Systems and/or by the use of the GANA Knowledge Plane DEs, and the Config Data can be encapsulated into the Network Profile or can be used and stored as Config Files or Models.
  - Service chaining descriptions for service chains.
  - Service definitions.
  - Network Service Designs and expressed as service topologies or graphs.
  - Conflict-free Policies for Network Applications and/or Network Behaviours.
  - Goals.

- **Use of GANA Knowledge Plane DEs and MBTS in generating Config Data and Completing the Network Profile with data:**
    - GANA Knowledge Plane DEs may be used for auto-computing and generating some Config Data required for initial network service configuration, encapsulate the Config Data into the Network Profile or generate and emit the Config-Data as Config Files or Models.
    - While Network Profiles may be generated first as Skeletons by automated management tool-chains the GANA Knowledge Plane DEs shall be used for auto-computing and generating any additional Config Data required for initial network service configuration and then encapsulating the Config Data into a complete Network Profile or generating and emitting the additionally required Config Data as Config Files or Models that are then used to configure the network infrastructure elements via the MBTS and are also stored into the ONIX by the Knowledge Plane DEs.
    - The process of auto-generating configuration data, which is either added into the Network Profile or as separate Config Files or Models should take into consideration the management and control protocols that would be used by the MBTS to send configuration commands into the network infrastructure that convey the configuration data to the targeted Network Elements (NEs) of the network infrastructure. The MBTS shall be invoked by DEs in the Knowledge Plane to add to the Network Profile or Config files or models the mappings between commands and data models used by Knowledge Plane DEs and protocol specific commands and data models that are used to configure (program) the Network Elements (NEs) in the network infrastructure. During the operation phase of the Knowledge Plane, the MBTS does the translation between the two domains (that of the Knowledge Plane DEs' language and data models and the domain of the southbound protocols and data models of the (AMC)-MBTS's interfacing with the network infrastructure).
    - Traditional management systems and/or the Auto-Configuration DE in the Knowledge Plane cooperatively act together to drive the initial configuration of network services and resources while thereafter the Knowledge Plane DEs can do a reconfiguration of network services and resources to achieve self-adaptation objectives such as self-optimization, self-healing, etc.
- **Validation of the Inputs against Conflicts and Undesired Effects, and against instantaneous network/resource capabilities:**
    - Goals conflict resolution.
    - Policy conflict resolution.
    - Resolution of bad Service Chaining (Problems such as loops that may occur in certain service chain instances that should be avoided).
- **Consumers of the inputs generated by the Automated Management and Governance Tools:**
    - Autonomic Management andControl Software in the Knowledge Plane, namely DEs.
    - Orchestration Software.
    - Those inputs are Injected into the Software Components that require the inputs which shall have been validated a-priori to avoid conflicts and undesired effects in the consuming components.
- **Reporting by the Autonomic Network:** The autonomic network and its associated GANA Knowledge Plane disseminate to the Automated Management Tool Chains Aggregate Reports generated by the Autonomic Network's DEs. For example such Reports could contain information on the kind of events, challenges or situations in general, that the Network's DEs have met and resolved by their collective actions as well as the situations that could not be resolved over a certain time duration, i.e. information that could be useful for the human to inspect or use in offline analysis to infer on the quality of the Decision-making capabilities of the various DEs.

C.2 Options that may be applied in Configuration of MEs and DEs using Network Profile and Configuration Files or Models and Data

C.2.0 Complementary insights to insights provided in the clause on GANA Operations Procedures for the Human Network Operator

NOTE 1: This clause provides details that complement insights provided in the clause on GANA Operations Procedures for the Human Network Operator, by providing further details on options on how the GANA Knowledge Plane can be complementarily used together with other types of management and control systems in NEs configurations.

There are three options that can be pursued as described below.

NOTE 2: The details provided here should be complementarily used together with details provided on the possible approaches to implementing the ONIX as described in Annex D.

NOTE 3: There is on-going work in ETSI on elaborations of the Implementation Guide for GANA and the resultant work is expected to be published in ETSI TR 103 195-3 [i.71] that is expected to include more insights on how to implement the GANA Functional Blocks (including the ONIX).

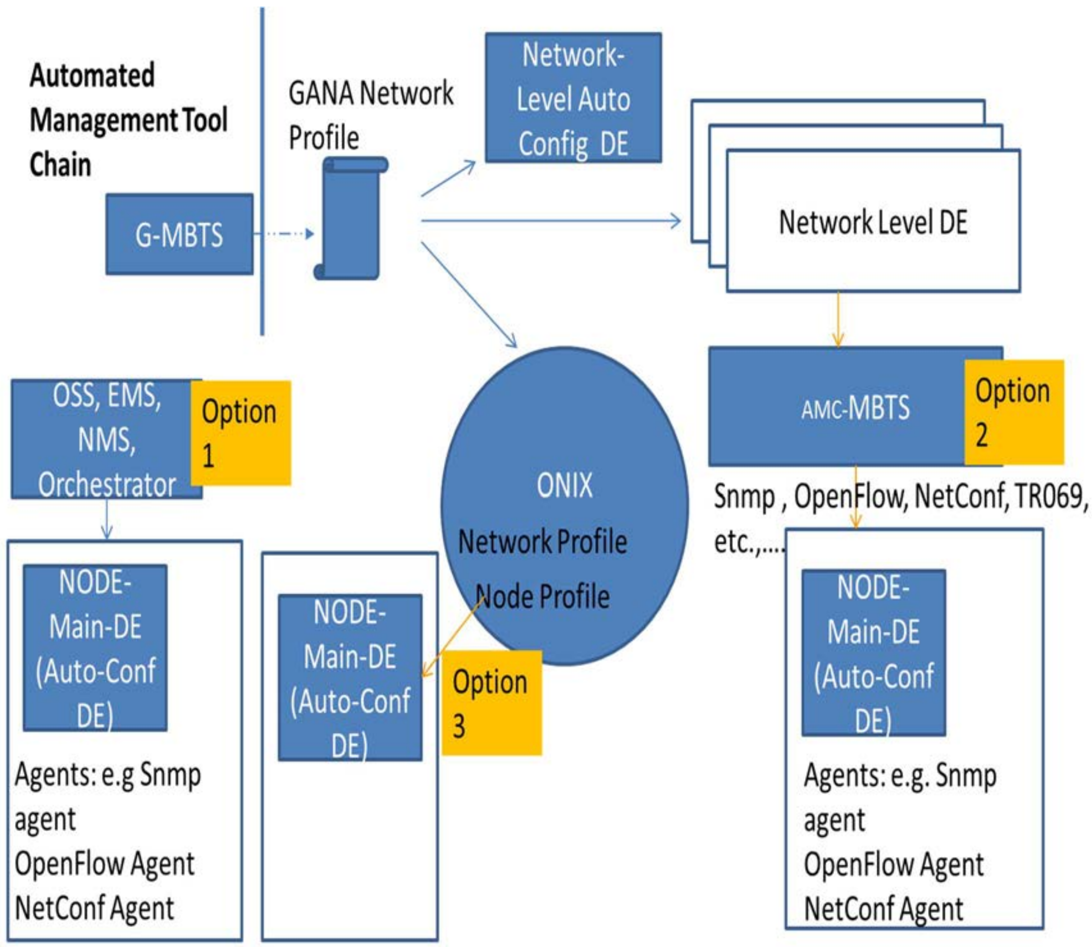


Figure C.1: Options that may be applied in Configuration of MEs and DEs using GANA Network Profile and Configuration Files or Models and Data

C.2.1 Option-1: Network Configuration and Provisioning via Management and Orchestration Systems and interworking with the GANA Knowledge Plane

Network Element (NE)'s Managed Entities (MEs) at the Networking Resources Layer of the GANA model, e.g. individual protocols or stacks may be configured using traditional vendors' management systems such as EMS, NMS, OSS an local management agents available on the NE (e.g. SNMP Agent, OpenFlow Agent, NETCONF Agent, etc.) a possible scenario in network evolution and transitioning to ultimate Knowledge Plane based management and control. In this option the following aspects should be considered:

- Configuration of Network Elements (NEs) and Provisioning of Network Services can be performed using a combination of the following systems:
  - OSS;
  - EMS;
  - NMS;
  - Network Service or Network Resource Orchestrator;
  - SDN Controller;
  - GANA Knowledge Plane Components.

  The GANA Knowledge Plane DEs and the MBTS may be integrated as part of an OSS or an SDN Controller as described in [i.15], [i.24], [3] or the GANA Knowledge Plane DEs and the MBTS (the AMC-MBTS) may be implemented as standalone but coupled run-time entities that can interact with and drive the management and control systems such as an OSS, Network Service or Network Resource Orchestrator, SDN Controller, as described in [i.15], [i.24] and [3].

- Management and Control Systems or Special Tools may provide means for the Network Service Designer to define High Level Business Goals for the Network, Service Profiles andSLAs, Network Service Definitions/Graphs, Application Intents and Profiles/Policies which are then used to generate input to systems that configure the network accordingly.

- Management and Control Systems may provide means for the Network Service Designer to design an End-to-End Network Service Graph/Topology using various types of Service Functions (Physical or Virtual), Service Chaining, and even Network Slicing considerations.

- The Network Service Provider may choose to use management and control systems such as OSS, EMS, NMS, Network Service or Network Resource Orchestrator, and SDN Controller, to perform the initial Configuration of Network Elements (NEs) and Provisioning of Network Services, while administratively restricting GANA Knowledge Plane DEs from participating in this network service initialization process but rather making the GANA Knowledge Plane DEs takeover and focus on Driving Self-Adaptation Objectives such as Self-Optimization, Self-Diagnosis, Self-Healing and Closed-Loop Service Assurance for a network service once instantiated. Network Service definitions, network application intents and profiles and policies and other config-data and models that may be generated by tool chains for automated management as illustrated on Figure 31, can all be used to drive Service, Domain and Resource Orchestration that in turn inductively trigger orchestration at lower layers such as SDN Controller down to the actual configuration that then happens on Network Elements (NEs) in the network infrastructure layer.

- Even if the Network Service Provider has chosen to use management and control systems such as OSS, EMS, NMS, Network Service or Network Resource Orchestrator, and/or SDN Controller, to perform the initial Configuration of Network Elements (NEs) and Provisioning of Network Services, Automated Management Tool Chains should still generated Network Profiles and Config Files or Models that should be complemented and updated by the Knowledge Plane DEs and the MBTS as illustrated in Figure 31. The Network Profile and Node Profiles shall always contain the configuration items that DEs are supposed to use in configuring themselves in order to be operational.

- Knowledge Plane DEs and MBTS may be employed together or separately in aiding the auto-generation of certain types of Config data by Automated Management Tools illustrated on Figure 31. Knowledge Plane DEs may be employed to compute configuration data for the various Managed Entities (MEs) based on the current state of the network and the data models and semantics (language) used by the Knowledge Plane that is agnostic to management or control protocol and vendor specific data models, and if used together with the MBTS the configuration data can be augmented with items and semantics tied to management or control protocol and vendor specific data models. This aspect concerns the MBTS used towards the network infrastructure layer (i.e. the AMC-MBTS).

- When the MBTS (as software libraries) is integrated as southbound interface of an SDN Controller as described in ETSI GANA White Paper No.16 [3] the MBTS still serves to employ semantics tied to management or control protocol and vendor specific data models used towards the network infrastructure layer, and preforms any translation required for the northbound interface of the MBTS.

Implications of this Option on Network Profile, Node Config Files or Models and Data, KP and GANA Node DEs Behaviors:

- Node-Profiles and NodeConfig files or Models should contain the items that are required by DEs in order to self-configure and start their operations in the GANA node.

- The GANA Node's Auto-Config DE is not supposed to configure the GANA Node's Managed Entities (MEs) whose Initial Configuration is performed by the traditional vendors' management systems such as EMS, NMS, OSS, or Service or Domain Orchestrator through means such as SNMP, NETCONF, OpenFlow, etc.

- The GANA Node's Auto-Config DE performs Node Capabilities Compilation and Advertisement to entities allowed to receive Capabilities by an Administrative Policy.

- The GANA Node's Auto-Config DE shall handle the reception of the Node Profile and any Node Config Files that may be relayed by MBTS if not through the ONIX, orchestrate the Level-3 and Level-2 DEs and provide them with the Profiles they need in order to self-configure and start their operations in the GANA node.

- Dynamic (re)-configurations of MEs (Managed Entities) in the Networking Resources Layer (e.g. protocols) that is driven by DE algorithms and objectives that require the collective autonomous operations of DEs in a GANA Node, e.g. for self-optimization and other self-adaptation management and control objectives, may either be done via an API that implements the Reference Point defined in the GANA called "***Rfp_GANA-Level2-AccessToProtocolsAndMechanisms***" (more details can be found in ***Annex B*** *on "Requirements for Protocols and APIs for Enabling GANA based Autonomics, Cognitive Networking and Self-Management of Networks and Services in Evolving and Future Networks"*, or the DEs may configure the MEs through other methods such as SNMP, or Management Agents that in turn can configure the targeted MEs.

- To avoid conflicts in network resources configuration through any means other than the through the Knowledge Plane there is a need to enforce synchronization of network services configuration/programming with the Knowledge Plane as described in **Annex B** *on* "*Requirements for Protocols and APIs for Enabling GANA based Autonomics, Cognitive Networking and Self-Management of Networks and Services in Evolving and Future Networks*".

C.2.2 Option-2: Configurations of NEs happens through the KP DEs and MBTS, with or without being complemented by traditional vendors' management systems

NOTE: Traditional vendors' management systems refer to systems such as EMS, NMS, OSS, Service or Domain Orchestrator.

The Option-2's Implications on Network Profile, Node Config Files or Models and Data, KP and GANA Node DEs Behaviors:

- Driven by the Network-Level Auto-Configuration-DE, Node-Profiles and NodeConfig files are extracted from the Network Profile by the MBTS and are sent to individual GANA Nodes by the MBTs.

- The Network-Level Auto-Configuration-DE may drive all NEs configurations via the MBTS without any need for traditional vendors' management systems such as EMS, NMS, OSS or Service or Domain Orchestrator to drive any configurations on the NEs, while the MBTS is also used to facilitate the dissemination of Node Profiles and Node Configuration Files or Models to GANA Nodes' Auto-Configuration-DEs.
- If for some reasons there is need for using traditional vendors' management systems such as EMS, NMS, OSS or Orchestrator to drive any configurations on the NEs, then the Network-Level Auto-Configuration-DE and other KP DEs and MBTS should cover only those configuration aspects that complement any configurations performed through traditional vendors' management systems such as EMS, NMS, OSS, Service or Domain Orchestrator. The Network-Level Auto-Configuration-DE and other KP DEs should infer and be aware of state (configuration state) installed into NEs by the traditional vendors' management systems such as EMS, NMS, OSS, Service or Domain Orchestrator. The Network Profile used by the KP DEs to drive configurations in this case may initially contain only those items the DEs in the KP and low level DEs in the NEs require for the configurations they should complement to those performed by the other systems. Re-configurations of NEs for the purposes of self-adaptation performed by the KP DEs, e.g. self-optimization and other self-adaptation management and control objectives, should be left to be performed by the KP DEs, which may either use the MBTS, lower level DEs and other means to effect changes on NEs or may drive the traditional vendors' management systems such as EMS, NMS, OSS, Service or Domain Orchestrator, to indirectly effect the desired chances to the configuration state of NEs.
- The GANA Node's Auto-Config DE is not supposed to configure the GANA Node's Managed Entities (MEs) whose Initial Configuration is performed by MBTS through means such as SNMP, NETCONF, OpenFlow, CMIP, COPS, TR069, XMPP, etc.
- The GANA Node's Auto-Config DE performs Node Capabilities Compilation and Advertisement to entities allowed to receive Capabilities by an Administrative Policy.
- The GANA Node's Auto-Config DE shall handle the reception of the Node Profile and any Node Config Files that may be relayed by MBTS if not through the ONIX, orchestrate the Level-3 and Level-2 DEs and provide them with the Profiles they need in order to self-configure and start their operations in the GANA node.
- Dynamic (re)-configurations of MEs (Managed Entities) in the Networking Resources Layer (e.g. protocols) that is driven by DE algorithms and objectives that require the collective autonomous operations of DEs in a GANA Node, e.g. for self-optimization and other self-adaptation management and control objectives, may either be done via an API that implements the Reference Point defined in the GANA called "***Rfp_GANA-Level2-AccessToProtocolsAndMechanisms***", or the DEs may configure the MEs through other methods such as SNMP, or Management Agents that in turn can configure the targeted MEs.
- Details on Requirements pertaining to the Integration of GANA and vendors' management systems such as EMS, NMS, OSS or Service or Domain Orchestrator are found in **Annex B** *on Requirements for Protocols and APIs for Enabling GANA based Autonomics, Cognitive Networking and Self-Management of Networks and Services in Evolving and Future Networks*.

C.2.3 Option-3: Configurations of NEs happens through the KP DEs and MBTS, with ONIX facilitated dissemination of Config-Data to GANA Nodes, with/without being complemented by traditional vendors' management systems

This Option-3 involves Configurations of NEs happens through the KP DEs and MBTS, with ONIX facilitated dissemination of Node Profiles and Node Configuration Files or Models to GANA Nodes' Auto-Configuration-DEs, with/without being complemented by traditional vendors' management systems.

NOTE: Traditional vendors' management systems refer to systems such as EMS, NMS, OSS, Service or Domain Orchestrator.

The Option-3'sImplications on Network Profile, Node Config Files or Models and Data, KP and GANA Node DEs Behaviors:

- Driven by the Network-Level Auto-Configuration-DE, Node-Profiles and NodeConfig files are extracted from the Network Profile by the MBTS and are sent to individual GANA Nodes by the MBTs.
- The Network-Level Auto-Configuration-DE may drive all NEs configurations via the MBTS without any need for traditional vendors' management systems such as EMS, NMS, OSS or Orchestrator to drive any configurations on the NEs, while the ONIX is used to facilitate the dissemination of Node Profiles and Node Configuration Files or Models to GANA Nodes' Auto-Configuration-DEs.
- If for some reasons there is need for using traditional vendors' management systems such as EMS, NMS, OSS or Orchestrator to drive any configurations on the NEs, then the Network-Level Auto-Configuration-DE and other KP DEs and MBTS should cover only those configuration aspects that complement any configurations performed through traditional vendors' management systems such as EMS, NMS, OSS, Service or Domain Orchestrator. The Network-Level Auto-Configuration-DE and other KP DEs should infer and be aware of state (configuration state) installed into NEs by the traditional vendors' management systems such as EMS, NMS, or OSS. The Network Profile used by the KP DEs to drive configurations in this case may initially contain only those items the DEs in the KP and low level DEs in the NEs require for the configurations they should complement to those performed by the other systems. Re-configurations of NEs for the purposes of self-adaptation performed by the KP DEs, e.g. self-optimization and other self-adaptation management and control objectives, should be left to be performed by the KP DEs, which may either use the MBTS, lower level DEs and other means to effect changes on NEs or may drive the traditional vendors' management systems such as EMS, NMS, OSS, Service or Domain Orchestrator, to indirectly effect the desired chances to the configuration state of NEs.
- The GANA Node's Auto-Config DE is not supposed to configure the GANA Node's Managed Entities (MEs) whose Initial Configuration is performed by MBTS through means such as SNMP, NETCONF, OpenFlow, etc.
- The GANA Node's Auto-Config DE performs Node Capabilities Compilation and Advertisement to entities allowed to receive Capabilities by an Administrative Policy.
- The GANA Node's Auto-Config DE shall handle the reception of the Node Profile and any Node Config Files that may be relayed by MBTS if not through the ONIX, orchestrate the Level-3 and Level-2 DEs and provide them with the Profiles they need in order to self-configure and start their operations in the GANA node.
- Dynamic (re)-configurations of MEs (Managed Entities) in the Networking Resources Layer (e.g. protocols) that is driven by DE algorithms and objectives that require the collective autonomous operations of DEs in a GANA Node, e.g. for self-optimization and other self-adaptation management and control objectives, may either be done via an API that implements the Reference Point defined in the GANA called "***Rfp_GANA-Level2-AccessToProtocolsAndMechanisms***", or the DEs may configure the MEs through other methods such as SNMP, or Management Agents that in turn can configure the targeted MEs.
- Details on Requirements pertaining to the Integration of GANA and vendors' management systems such as EMS, NMS, OSS or Service or Domain Orchestrator are found in **Annex B** *on Requirements for Protocols and APIs for Enabling GANA based Autonomics, Cognitive Networking and Self-Management of Networks and Services in Evolving and Future Networks*.

Annex D (normative): The ONIX System and possible ways to implement ONIX

D.1 The ONIX System and possible ways to implement ONIX

ONIX is characterized by:

- **ONIX Internal Protocols** for Federation of Information Servers, e.g. using DHTs (Distributed Hash Tables) and other techniques.
- **ONIX External Protocols** (e.g. DHCPv6++, an IPv6 based protocol that was prototyped in [i.36], [i.38], [i.46], [i.47], gRPC [i.37], or any other protocols that can be used for communicating information in formats that ONIX can be made to support)for supporting the following operations by ONIX users:

  1) Publish Information into ONIX.

  2) Subscribe to receive Information from ONIX, including "on-behalf" subscriptions.

  3) Query and Find Operation to retrieve Info from ONIX.

In terms of Information Servers as members of ONIX, the following types of servers can be distinguished:

- Information Server that stores Info purely in ONIX Native Format such as XML, YANG, and other types that could be supported as ONIX native formats for information communicated to ONIX users (consumers of the Information such as GANA Decision Elements (DEs) or Network Elements (NEs-Physical or Virtual)) or stored by ONIX users intending to store information into the ONIX.
- Traditional Relational Databases that can be made to support ONIX Internal Protocols for Federation of Information Servers, e.g. using DHTs (Distributed Hash Tables) and other techniques, and can convert some data they store into ONIX native formats for information exchange with ONIX users.
- Other Types of Data Storage that can be made to support ONIX Internal Protocols for Federation of Information Servers, e.g. using DHTs (Distributed Hash Tables) and other techniques, and can convert some data they store into ONIX native formats for information exchange with ONIX users.
- Shared Common Repository (R) for State Data (e.g. VNFs (Virtual Network Functions State Data)) that is made to support ONIX Internal Protocols for Federation of Information Servers, e.g. using DHTs (Distributed Hash Tables) and other techniques, and can convert some data they store into ONIX native formats for information exchange with ONIX users. Such a Shared Common Repository can still be directly accessible via a Native Data/Info Access Interface it should still expose. Such a Direct Data Access Interface exposed by the Server may support access methods such as LDAP, SPML, Diameter, API, or other methods.
- Some Information Servers that are members of ONIX and also exposing their Server-Native Interfaces for use by some "Non-ONIX Native" Data/Information Repository User. As such server-native Data/Info Access Interface may be exposed by e.g. a Repository or Database that is also a member of ONIX at the same time.

NOTE 1: Data/Information Servers ("N", "X", R") in Figure D.1 support ONIX Internal Protocols for Federation of Information Servers (i.e. they can be made to operate as ONIX members) and can convert some data they store into ONIX native formats for information exchange with ONIX users.

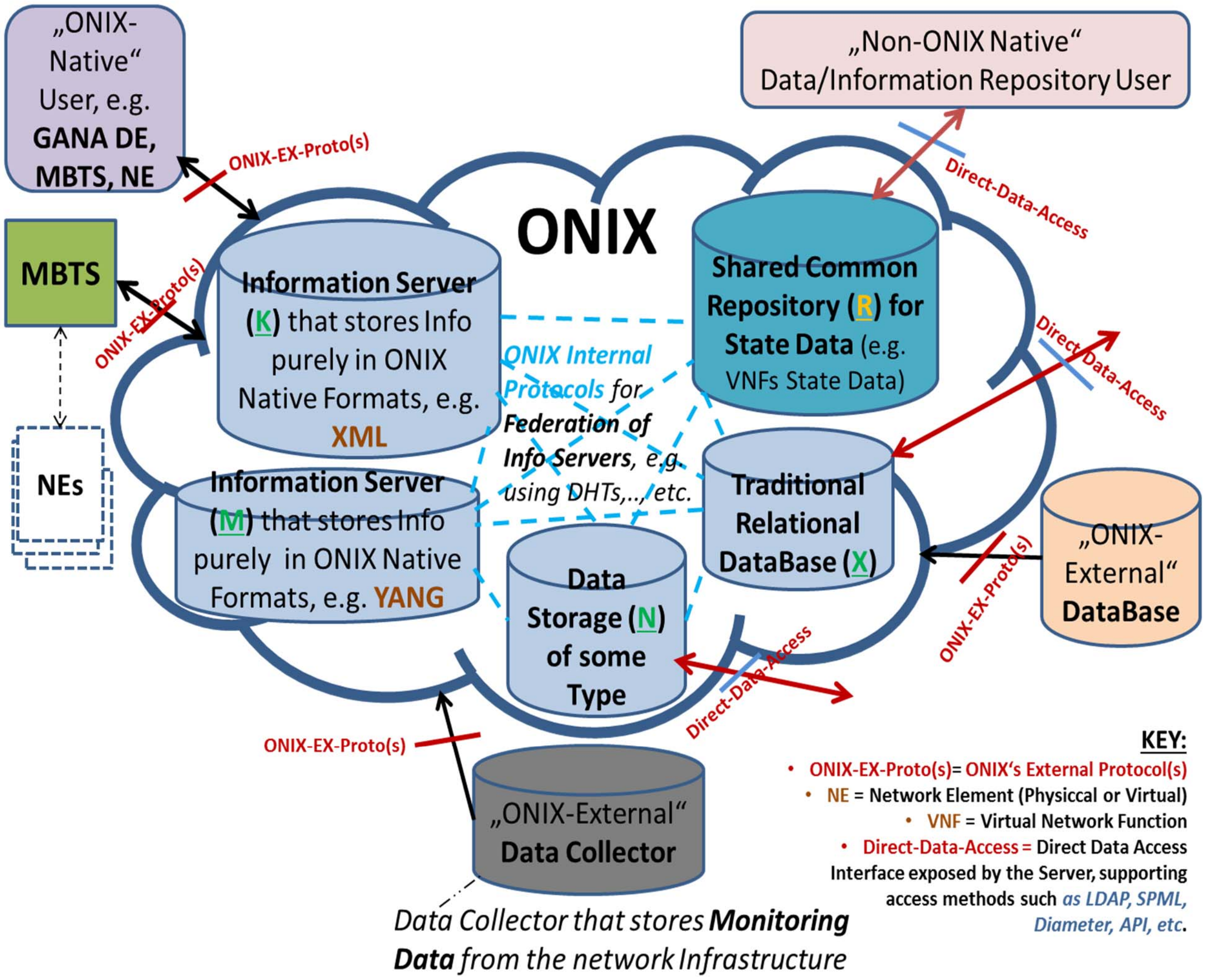


Figure D.1: ONIX Interfaces and Services, Internal and External Protocols

Table D.1 provides illustrations of External Interfaces and Services of ONIX, External Protocols and Internal Protocols of ONIX. Figure D.2 provides an illustration of an example approach used in an ONIX implementation of some limited functionality (in reference to [i.36], [i.38], [i.46] and [i.47]).

Table D.1: ONIX Interfaces and Services, Internal and External Protocols

External Interfaces and Services of ONIX	External Protocols	Internal Protocols of ONIX
ONIX-EX-Proto(s) Interface enables "ONIX native" users such as **GANA DE, MBTS, or a GANA Node (NE)** to use the following services of ONIX: • *Publish Information into ONIX* • *Subscribe to receive Information from ONIX* • *Query and Find Operation to retrieve Info from ONIX*	Various external protocols may be supported by ONIX for providing such services, e.g. extended DHCPv6 (DHCPv6++ [i.36], [i.38], [i.46], [i.47]), gRPC [i.37], or any other protocols that can be used for communicating information in formats that ONIX can be made to support.	Various external protocols may be supported by ONIX for ONIX Internal Protocols for Federation of Information Servers, e.g. using DHTs (Distributed Hash Tables) and other techniques.
ONIX-EX-Proto(s) Interface enables ONIX native users such as **"ONIX-External" Data Collector** that stores Monitoring Data from the network infrastructure to use the following services of ONIX: • *Publish/Update Information into the ONIX via any accessible ONIX server (Cognitive Algorithms operate on raw data on the Data Collector and create Knowledge stored into ONIX and also streamed to DEs in the GANA Knowledge Plane)*	Various external protocols may be supported by ONIX for providing such services, e.g. extended DHCPv6 (DHCPv6++ [i.36], [i.38], [i.46], [i.47]), gRPC [i.37], or any other protocols that can be used for communicating information in formats that ONIX can be made to support.	Various external protocols may be supported by ONIX for ONIX Internal Protocols for Federation of Information Servers, e.g. using DHTs (Distributed Hash Tables) and other techniques.
ONIX-EX-Proto(s) Interface enables ONIX native users such as **"ONIX-External" DataBase** to use the following services of ONIX: • Publish/Update Information into the ONIX via any accessible ONIX server	Various external protocols may be supported by ONIX for providing such services, e.g. extended DHCPv6 (DHCPv6++ [i.36], [i.38], [i.46], [i.47]), gRPC [i.37], or any other protocols that can be used for communicating information in formats that ONIX can be made to support.	Various external protocols may be supported by ONIX for ONIX Internal Protocols for Federation of Information Servers, e.g. using DHTs (Distributed Hash Tables) and other techniques.

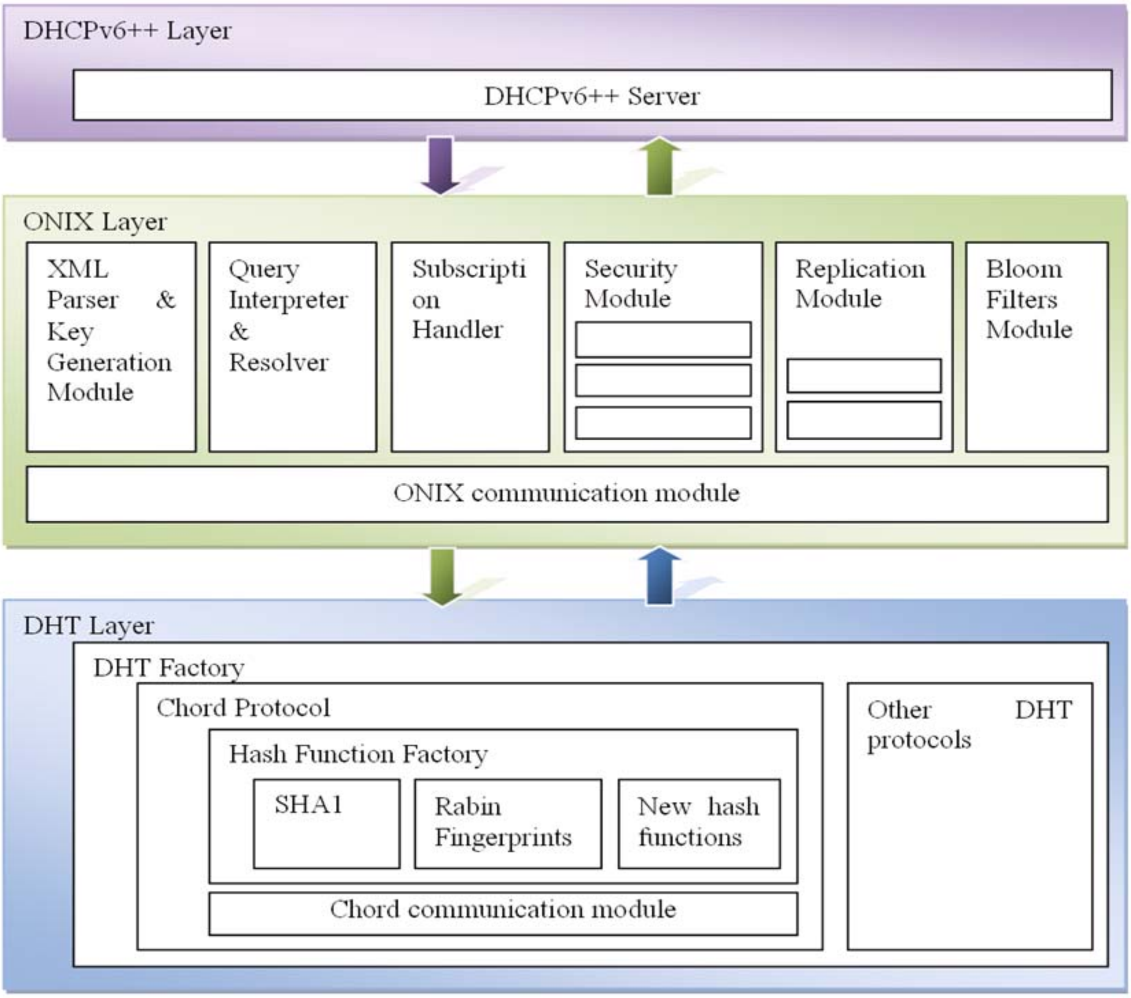


Figure D.2: An illustration of an example approach used in an ONIX implementation of some limited functionality (in reference to [i.36], [i.38], [i.46] and [i.47])

NOTE 2: There are other sources of resourceful concepts and insights in literature that may be considered in implementing ONIX capabilities such as context storage (and possibly processing as well) and brokerage, and metadata management for the various data models the ONIX servers may be designed to use in information stored on some ONIX servers and also for information exchanged between ONIX and its user entities. For example, Fi-WARE concepts and insights from [i.77] that relate to context storage, context brokering and metadata management for data models can be exploited in the design and implementation of the ONIX servers.

Annex E (informative): Bibliography

Examples of some Bibliographical resources on autonomics that may be considered in implementing GANA autonomics:

NOTE: The present document has provided throughout the various clauses and the Annexes some references (pointers to useful bibliographical material) of value to consider in designing and implementing GANA Functional Blocks for autonomics (e.g. DEs). For example, some sources that may contain further insights on how to implement a particular DE have been provided to enable autonomics implementers to obtain some insights on how to implement the DE (e.g. how to design the DE logic and algorithms, etc.). However, there are so many other useful sources in literature and in the current and future research results on autonomics (including the use of Artificial Intelligence (AI) algorithms in designing autonomic management and control components such as DEs) that should be considered by implementers. Therefore, as examples, the following references serve as bibliography that may provide useful insights that complement the present document.

- ETSI TS 103 371: "Network Technologies (NTECH); Autonomic network engineering for the self-managing Future Internet (AFI); Proofs of Concept Framework".
- Gilles Privat, Wenbin Li: "Unlocking IoT/WoT APIs with linked data & semantic referencing"; 29th March 2016, Orange APIs Standardisation Seminar.
- John C Strassner, Nazim Agoulmine, and Elyes Lehtihet: "FOCALE: A Novel Autonomic NetworkingArchitecture"; In LAACS, Campo Grande, MS, Brazil, 2006.
- David D. Clark, Craig Partridge, and J. Christopher Ramming: "A knowledge plane for the Internet"; In SIGCOMM.
- Michal Wodczak, Tayeb Ben Meriem, Benoit Radier, Ranganai Chaparadza, Kevin Quinn, Jesse Kielthy, Brian A. Lee, Laurent Ciavaglia, Kostas Tsagkaris, Szymon Szott, Anastasios Zafeiropoulos, Athanassios Liakopoulos, Apostolos Kousaridas, Maurice Duault: "Standardizing a reference model and autonomic network architectures for the self-managing future internet"; IEEE Network 25(6): 50-56 (2011).
- Deliverable D2 by NGMN Alliance: "Recommended Practices for Multi-vendor SON Deployment", version 1.0, 28 January 2014.
- Prof.P.P.Rewagad, Dipali P.Sapkal: "Management of Future Internet with Autonomic Network Architecture and Model", IOSR Journal of Engineering (IOSRJEN): e-ISSN: 2250-3021, p-ISSN: 2278-8719, www.iosrjen.org: Volume 2, Issue 10 (October 2012), PP 63-70.
- Mehmet Toy: "Self-Managed Networks with Fault Management Hierarchy", Complex Adaptive Systems, Publication 4: Cihan H. Dagli, Editor in Chief: Conference Organized by Missouri University of Science and Technology: 2014- Philadelphia, PA.
- Cornaglia Bruno, Young Gavin, Marchetta Antonio: "Fixed Access Network Sharing", Elsevier publication: 10.1016/j.yofte.2015.07.008.
- Phillip Verbancsics, Douglas S. Lange: "Using Autonomics to Exercise Command and Control of Networks in Degraded Environments", 2013 Conference paper by Space and Naval Warfare Systems Center Pacific San Diego CA, USA (http://www.dtic.mil/get-tr-doc/pdf?AD=ADA587015); In proceeding of 18th ICCRTS (18th International Command and Control Research and Technology Symposium, Alexandria, Virginia, USA, 2013).
- Orange and Huawei Joint White Paper: "Future OSS, Providing the AGILITY to support digital operations transformation of hybrid networks", REV1.0, 05/02/2017.
- Kostas Tsagkaris, Michelle M Wetterwald, Nancy Alonistioti et al.: "Autonomic and Self-Managed 3GPP Core Networks"; In proceedings of the European Conference on Networks and Communications (EUCNC), sponsored by the European Commission and IEEE; 27 to 30 June 2016, Athens, Greece.

- Lav Gupta et al.: "Fault and Performance Management in Multi-Cloud Based NFV using Shallow and Deep Predictive Structures"; In proceedings of 2017 26th International Conference on Computer Communication and Networks (ICCCN): DOI: 10.1109/ICCCN.2017.8038530.
- Smirnov, M., Tiemann, J., Chaparadza, R.; Rebahi, Y., Papavassiliou, S. et al.: "Demystifying self-awareness of autonomic systems"; In ICT-MobileSummit 2009 Conference Proceedings: 10-12 June 2009, Santander, Spain; ISBN: 978-1-905824-12-0; 9 pp.
- Bassem Mokhtar, Mohamed EItoweissy: "Memory-enabled Autonomic Resilient Networking", pp. 132-141, 2011, doi:10.4108/icst.collaboratecom.2011.247113; In proceedings of International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), Orlando, Florida, USA, October 15-18, 2011.

History

Document history		
V1.1.1	May 2018	Publication