ETSI TR 103 536 V1.1.2 (2019-12)

SmartM2M;
Strategic/technical approach on how to achieve interoperability/interworking of existing standardized IoT Platforms

Reference
RTR/SmartM2M-103536v112

Keywords
interoperability, IoT, IoT platforms, oneM2M,
SAREF, semantic

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

Contents

Intellectual Property Rights 8
Foreword 8
Modal verbs terminology 8
Introduction 8
1 Scope 10
1.1 Context for the present document 10
1.2 Scope of the present document 10
2 References 11
2.1 Normative references 11
2.2 Informative references 11
3 Definition of terms, symbols and abbreviations 15
3.1 Terms 15
3.2 Symbols 16
3.3 Abbreviations 16
4 Platforms Interoperability in the context of IoT 18
4.1 A global approach to IoT Systems 18
4.1.1 Major characteristics of IoT systems 18
4.1.2 The need for an "IoT-centric" view 19
4.1.2.1 Introduction 19
4.1.2.2 Roles 19
4.1.2.3 Reference Architecture(s) 19
4.1.2.4 Guidelines 20
4.2 Main objectives of the present document 20
4.3 Purpose and target group 20
4.4 Content of the present document 20
5 The IoT Platforms Landscape 21
5.1 A framework for IoT Platforms 21
5.1.1 Expectations and definition 21
5.1.2 Challenges 22
5.1.2.1 Flexibility, versatility 22
5.1.2.2 Semantic Interoperability 22
5.1.2.3 Flexible deployment models 23
5.1.2.4 Open and efficient implementations 23
5.1.2.5 Non-functional properties 23
5.1.2.6 Security 23
5.1.2.7 Privacy and data confidentiality 24
5.1.2.7 Integration with legacy 24
5.2 An IoT Platforms Landscape 24
5.2.1 Fragmentation and lack of maturity 24
5.2.2 A typology of platforms 24
5.2.2.1 Main dimensions for platform analysis 24
5.2.2.2 Scope and breadth 25
5.2.2.3 Openness 25
5.2.2.4 Origin and governance 26
5.2.2.5 Ecosystem 28
5.2.2.6 Maturity 28
5.2.2.7 A classification of Platforms 29
5.2.3 Finding a way in the jungle of platforms 29
5.2.3.1 Introduction 29
5.2.3.2 Platforms identified by UNIFY-IoT and the IoT-EPI 30
5.2.3.3 Platforms in the IoT Large Scale Pilots 30
5.2.3.4 Emerging approaches: Marketplaces and APIs 32
5.3 Standardized IoT Platforms 33

5.3.1 Characterization of Standardized IoT Platforms 33
5.3.2 oneM2M 33
5.3.2.1 Scope 33
5.3.2.2 Architecture 34
5.3.2.3 Interoperability and other aspects 35
5.3.3 The OCF Platform 36
5.3.3.1 The Ecosystem 36
5.3.3.2 The Interoperability 36
5.3.3.3 The Architecture 36
5.3.4 The Apache Platform 37
5.3.4.1 The Ecosystem 37
5.3.4.2 Some elements of the platform 38
5.3.5 Point solutions and the challenge of integration 39
5.3.5.1 Fitting point solutions in global platforms 39
5.3.5.2 Stand-alone or cloud-based solutions: two examples 39
5.3.5.3 The role of integration 40

6 Dealing with Interoperability 40
6.1 Strategic Approaches to Interoperability 40
6.2 Technical Approaches to Interoperability 41
6.2.1 A program for evolution 41
6.2.2 The Internet of Things (IoT): The basic objectives of IoT platforms 42
6.2.3 The WoT: a step towards interoperability of IoT platforms 42
6.2.4 The SWoT: The foundations for semantic interoperability of IoT platforms 42
6.3 Interoperability Frameworks 42
6.3.1 The AIOTI Reference Framework 42
6.3.2 Other Interoperability Frameworks 43
6.3.3 Interoperability examples of use-cases 44
6.4 The challenge of IoT Deployment 44
6.4.1 Key technologies and design requirements 44
6.4.2 Interoperability in Smart Cities 45
6.5 Criteria for Interoperability 45

7 The case of Industrial IoT 47
7.1 The challenges of Industrial IoT 47
7.1.1 The role of Industrial IoT in Smart Manufacturing 47
7.1.1.1 Smart Manufacturing 47
7.1.1.2 Industrial IoT 48
7.1.2 IIoT: a major segment of the IoT with significant specificities 49
7.1.2.1 A major business segment 49
7.1.2.2 Differences with traditional Operational Technology (OT) 49
7.1.2.3 Differences with consumer IoT 49
7.1.3 Expected Benefits of IIoT 50
7.1.4 Challenges and barriers to, and strategies for the adoption of IIoT 52
7.1.4.1 The current situation: A Progressive Adoption 52
7.1.4.2 On the importance of legacy: Greenfield vs Brownfield 52
7.1.4.3 Technical barriers to adoption 52
7.1.4.4 Strategic choices and their impact on platforms 53
7.2 Using Standardized Platforms in IIoT 54
7.2.1 Technical aspects 54
7.2.2 Connectivity 54
7.2.2.1 The importance of legacy 54
7.2.2.2 Greenfield: starting from scratch 54
7.2.2.3 Brownfield: integrating (with) legacy 55
7.2.3 Interoperability and the role of Semantics 56
7.2.4 IoT Virtualization and the role of Cloud 57
7.2.4.1 IoT Virtualization 57
7.2.4.2 Virtualization in the context of IIoT 58
7.2.5 Data Management and Analysis 58
7.2.6 Business Processes and Enterprise view 59
7.2.6.1 The need for Vertical Integration 59
7.2.6.2 The Impact of IIoT 60

7.2.7 Software Development 61
7.3 Platform adoption: proprietary or open/standardized 62
7.3.1 Proprietary platforms 62
7.3.1.1 Benefits and limits of proprietary platforms 62
7.3.1.2 Issues in coupling proprietary platforms and open/standardized platforms 62
7.3.2 A review of IIoT Platforms 63
7.3.2.1 Introduction 63
7.3.2.2 Standardized Platforms 63
7.3.2.3 Open Source Platforms 63
7.3.2.4 Industry Groups Platforms 63
7.3.2.5 Proprietary Platforms 65
7.3.3 Conclusions 66

8 Conclusions 66
8.1 Lessons learned 66
8.2 Guidelines and Recommendations 67
8.2.1 Introduction 67
8.2.2 Strategy Recommendations 68
8.2.3 Technical Recommendations 70
8.2.4 Recommendations to oneM2M 70

Annex A: IoT Platforms identified by UNIFY-IoT and IoT-EPI 72

A.1 The platforms identified by UNIFY-IoT 72

A.2 The platforms in the IoT-EPI projects 72

History 74

List of figures

Figure 1: AIOTI 3-Layer Functional Model....22
Figure 2: The Three IoT Software Stacks....23
Figure 3: Functional components of ACTIVAGE IoT platforms....31
Figure 4: The platforms across the AUTOPILOT Use Cases....32
Figure 5: oneM2M high level architecture....34
Figure 6: oneM2M functional architecture....35
Figure 7: Building Blocks of OCF architecture....37
Figure 8: The example of the Apache Hadoop ecosystem....37
Figure 9: AIOTI HLA Functional Model....43
Figure 10: Synthetic view of interoperability dimensions....46
Figure 11: Manufacturing Pyramid....48
Figure 12: Cyber-Physical Production Systems....48
Figure 13: The potential of Cloud-Native Infrastructures....57
Figure 14: An HLA for IoT Virtualization....58
Figure 15: OPC-UA multiple queries support....61
Figure 16: OPC-UA support for Information Models....64
Figure 17: OPC UA Companion Specifications - The example of EUROMAP....65
Figure 18: Risk of double work and approaches in the Companion Specifications....65
Figure 19: oneM2M OPC-UA Interworking and Functional Architecture with IPE....71
Figure A.1: UNIFY-IoT: Leading IoT Platforms selected for in-depth analysis....72

List of tables

Table 1: A classification of platforms 29
Table 2: Examples of Apache Software Components 38
Table 3: Expected benefits of Industrial IoT 50
Table 4: IIoT Platform selection scenarios 53
Table 5: Scenarios for Control Systems modifications 55
Table 6: Functional Level of Activities 59
Table A.1: Platforms used by the IoT EPI Projects 73

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 Report (TR) has been produced by ETSI Technical Committee Smart Machine-to-Machine communications (SmartM2M).

Modal verbs terminology

In the present document "**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

The initial project of Machine-to-Machine (M2M) communications was addressing the possibility for a device to interact with other devices (point-to-point or via gateways). This project has been handled at the very start by a variety of specialized (often sector-specific) platforms and solutions. Soon, it has been clear that this approach was bearing a strong risk of fragmentation with great difficulty in ensuring interoperability of such platforms when required. The Standard Development Organizations (SDOs) and Standard Setting Organizations (SSOs) have started to address the question of the M2M communications and have developed a number of approaches focusing on interoperability, in particular at the network level. Amongst the standards developed, some have addressed the possibility to serve as a basis for the development of platforms that could use these standards to deal with interoperability in a generic manner, across a variety of business sectors, with a variety of possible implementations. Such "standardized platforms" are relying on reference architectures, interoperability stacks addressing different layers, generic protocol adaptors, etc.

Gradually, the focus of the industry has shifted to the design and development of IoT systems with the purpose to offer full-fledge systems dealing with a vast number of devices (with various computing and interaction capabilities) and potentially integrating these devices into larger systems implementing often complex business processes. This has been enabled by the emergence of IoT devices with higher computing capacity and the possibility of producing massive amounts of data that will be collected, transformed, stored and managed by larger (non IoT specific) information systems which transform it into qualitative information to trigger useful actions.

This incorporation of IoT with Big Data is one new challenge for IoT platforms, a significant one but not the only one. Another example is the use of Virtualization technologies coming from Cloud Computing that wants to get the benefits of Cloud in terms of flexibility and cost effectiveness. In the case of Big Data or Virtualization, the role of standards is challenged by new approaches based on the usage of Open Source Software (OSS) components. The "standardized IoT platforms" will have to address the challenges and probably not all of the existing ones will be able to make it.

An important business sector for the validation of the approach of generic standardized platforms is Industrial IoT. The need for the Industry to have a holistic approach to the use of Information Technologies to foster innovation and competitiveness has been addressed by a variety of initiatives coming from business sectors (such as Industrie 4.0 in Germany and similar national initiatives) or from the European Commission (such as Digitizing European Industry - DEI). The approaches taken will have to combine the benefits of existing technology solutions (including established standards) with the flexibility offered by new approaches such as Big Data, Virtualization, or Semantic Interoperability.

Two main challenges have to be addressed by IoT standardization (organizations) and by the "standardized" platforms (an example is oneM2M, see ETSI TS 118 101 [i.13] that some of these organizations are developing:

- The "advanced technology" challenge posed by e.g. the incorporation of Big Data or Virtualization.
- The "business sector" challenge with the question of which level of genericity can be provided in support of the development of large IoT systems for Smart Cities, Intelligent Transport or Industrial IoT.
- The "standards" challenge posed by the role of emerging approaches such as Open Source.

The example of Industrial IoT is addressed in detail, based on considerations and questions such as the following:

- Considering that Industrial IoT is a business sector in which the Return on Investment (RoI) of IoT is expected to be positive in the short/medium period, how is it possible to foster the adoption of IoT standards and standardized IoT platforms in this particular sector.
- The adoption of standards and platforms for interoperability should benefit not only to the technology providers but, first and foremost, to those who purchase and use these solutions, in particular the SMEs who do not always have the technical knowledge and the leverage available to large businesses.

The present document addresses these questions first by carefully outlining the nature, the role of IoT platforms and proposing elements for the identification of the most relevant ones. It also addresses detailed examples such as Industrial IoT to outline the challenges posed to generic IoT platforms.

1 Scope

1.1 Context for the present document

The design, development and deployment of - potentially large - IoT systems require to address a number of topics - such as security, interoperability or privacy - that are related and should be treated in a concerted manner. In this context, several Technical Reports have been developed that each address a specific facet of IoT systems.

In order to provide a global a coherent view of all the topics addressed, a common approach has been outlined across the Technical Reports concerned with the objective to ensure that the requirements and specificities of the IoT systems are properly addressed and that the overall results are coherent and complementary.

The present document has been built with this common approach also applied in all of the other documents listed below:

- ETSI TR 103 533 [i.1].
- ETSI TR 103 534 [i.2].
- ETSI TR 103 535 [i.3].
- ETSI TR 103 537 [i.4].
- ETSI TR 103 591 [i.5].

1.2 Scope of the present document

The present document is addressing the issues related to the interoperability and interworking of IoT platforms, in particular standardized IoT platforms, and how the way they are handled can foster their adoption by the IoT community. The following points are discussed:

- What is a platform and what are the relevant ones for IoT?
- What are the main requirements of Interoperability and Interworking?
- How these requirements are taken into account by typical platforms.
- How those elements are taken into account in specific sectors such as Industrial IoT.
- Which recommendations can be made for an effective selection and usage?

2 References

2.1 Normative references

Normative references are not applicable in the present document.

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] ETSI TR 103 533 (V1.1.1): "SmartM2M; Security; Standards Landscape and best practices".

[i.2] ETSI TR 103 534 (Parts 1 and 2) (V1.1.1): "SmartM2M; Teaching material".

[i.3] ETSI TR 103 535 (V1.1.1): "SmartM2M; Guidelines for using semantic interoperability in the industry".

[i.4] ETSI TR 103 537 (V1.1.1): "SmartM2M; PlugtestsTM preparation on Semantic Interoperability".

[i.5] ETSI TR 103 591 (V1.1.1): "SmartM2M; Privacy study report; Standards Landscape and best practices".

[i.6] White Paper: "IoT Platforms Interoperability Approaches", IoT-EPI Platform Interoperability Task Force, 2017.

[i.7] AIOTI: "IoT LSP Standards Framework Concepts", Release 2.8, White Paper, 2017.

[i.8] AIOTI: "High Level Architecture (HLA)", Release 4.0, June 2018.

[i.9] "Semantic Interoperability", Release 2.0, AIOTI, 2015.

NOTE: Two new AIOTI Joint White Papers on Semantic Interoperability have been issued by AIOTI on 22 October 2019. See https://aioti.eu/aioti-iso-iec-jtc1-etsi-onem2m-and-w3c-collaborate-on-two-joint-white-papers-on-semantic-interoperability-targeting-developers-and-standardization-engineers/.

[i.10] UNIFY-IoT Deliverable D03.01: "Report on IoT platform activities", 2017.

NOTE: Available at http://www.internet-of-things-research.eu/pdf/D03_01_WP03_H2020_UNIFY-IoT_Final.pdf.

[i.11] UNIFY-IoT Deliverable D03.02: "Analysis on IoT Platforms Adoption Activities", 2017.

NOTE: Available at http://www.internet-of-things-research.eu/pdf/D03_02_WP03_H2020_UNIFY-IoT_Final.pdf.

[i.12] UNIFY-IoT Deliverable D05.01:"Interoperable IoT Platforms Standards Framework", 2017.

[i.13] ETSI TS 118 101 (V2.10.0): "Functional Architecture (oneM2M TS-0001 version 2.10.0 Release 2)".

[i.14] ETSI TS 118 102 (V2.7.1): "oneM2M Requirements (oneM2M TS-0002 version 2.7.1 Release 2)".

[i.15] oneM2M TS-0012 (2018): "Base Ontology".

[i.16] oneM2M TS-0023 (2018): "Home Appliances Information Model and Mapping".

[i.17] ETSI TS 118 121 (V2.0.0): "oneM2M; oneM2M and AllJoyn® Interworking (oneM2M TS-0021 version 2.0.0 Release 2)".

[i.18] oneM2M TS-0014 (2017): "LWM2M Interworking".

[i.19] oneM2M TS-0024 (2017): "OIC Interworking".

[i.20] oneM2M TS-0033 (2017): "Interworking Framework".

[i.21] ETSI TR 103 527 (V1.1.1):"SmartM2M; Virtualized IoT Architectures with Cloud Back-ends".

[i.22] ETSI TR 103 528 (V1.1.1): "SmartM2M; Landscape for open source and standards for cloud native software applicable for a Virtualized IoT service layer".

[i.23] ETSI TR 103 529 (V1.1.1): "SmartM2M; IoT over Cloud back-ends: A Proof of Concept".

[i.24] ETSI TS 103 378 (V1.1.1): "Smart Body Area Networks (SmartBAN) Unified data representation formats, semantic and open data model".

[i.25] ACTIVAGE Deliverable D3.1: "Report on IoT European Platforms". .

NOTE: Available at https://www.activageproject.eu/docs/downloads/activage_public_deliverables/ACTIVAGE_D3.1_M3_Report%20on%20IoT%20European%20Platforms_V1.0.pdf.

[i.26] T. Berners-Lee, J. Hendler and O. Lassila: "The Semantic Web" Scientific American, 2001, vol. 284, no 5, p. 28-37.

[i.27] Recommendation ITU-T Y.2063: "Framework of the web of things".

NOTE: Available at https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-Y.2063-201207-I!!PDF-E&type=items.

[i.28] Recommendation ITU-T Y.2060: "Overview of the web of things".

[i.29] EUROMAP 83: "OPC UA interfaces for plastics and rubber machinery - General Type definitions".

[i.30] G. Hatzivasilis, I. Askoxylakis, G. Alexandris, G. Spanoudakis, et al.: "The Interoperability of Things: Interoperable solutions as an enabler for IoT and Web 3.0", Conference: IEEE International Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks (CAMAD) 2018, Barcelona, Spain. Project: SEMIoTICS, September 2018 DOI:10.1109/CAMAD.2018.8514952.

[i.31] Linked Open Vocabularies for Internet of Things (LOV4IoT).

NOTE: Available at http://lov4iot.appspot.com/?p=ontologies.

[i.32] ETSI TS 103 264 (V1.1.1): "SmartM2M; Smart Appliances; Reference Ontology and oneM2M Mapping".

NOTE: Available at http://www.etsi.org/deliver/etsi_ts/103200_103299/103264/01.01.01_60/ts_103264v010101p.pdf.

[i.33] European Research Cluster on the Internet of Things: "Internet of Things. IoT Semantic Interoperability: Research Challenges, Best Practices, Recommendations and Next Steps", March 2015.

NOTE: Available at http://www.internet-of-things-research.eu/pdf/IERC_Position_Paper_IoT_Semantic_Interoperability_Final.pdf.

[i.34] Sarogini Grace Pease, Paul P. Conway and Andrew A. West: "Hybrid ToF and RSSI real-time semantic tracking with an adaptive industrial internet of things architecture", Journal of Network and Computer Applications, 99(August 2016): 98-109, 2017. ISSN 10958592. doi:10.1016/j.jnca.2017.10.010.

[i.35] Paul Murdock et al.: "Semantic Interoperability for the Web of Things", 2016.

NOTE: See note in [i.9].

[i.36] ETSI GS CIM 009: "Context Information Management (CIM); NGSI-LD API".

[i.37] Mahdi Ben Alaya, Khalil Drira, Ghada Gharbi: "Semantic-aware IoT platforms", IEEE AIMS2017. Honolulu Jul 2017.

[i.38] ETSI TS 102 690 (V1.2.1): "Machine-to-Machine communications (M2M); Functional architecture".

[i.39] Web of Things Working Group.

NOTE: Available at https://www.w3.org/WoT/WG.

[i.40] European Innovation Partnership for Smart Cities & Communities, EIP-SCC: "6-Nations Smart Cities Forum Smart Cities National Market Blueprint", Version 3, March 2016.

NOTE: Available at https://eu-smartcities.eu/sites/default/files/2018-09/6-Nations SC BLUEPRINT v3.pdf.

[i.41] AIOTI: "Smart City LSP Recommendations Report", AIOTI WG08 - Smart Cities, 2015.

NOTE: Available at https://aioti.eu/wp-content/uploads/2017/03/AIOTIWG08Report2015-Smart-Cities.pdf.

[i.42] Thomas Casey, Ville Valovirta, Immo Heino, Janne Porkka, Ville Kotovirta, Sampsa Ruutu: "Interoperability Environment for Smart Cities (InterCity) - Report of Phase 2 -Smart City Interoperability Environment Concept", September 2016.

NOTE: Available at https://www.vtt.fi/sites/InterCity/en/Documents/InterCity_Report_Phase_2_FINAL.pdf.

[i.43] Omer Ozdemir, José Manuel Cantera, Martino Maggio, Nicola Muratore, Francesco Arigliano, Eunah Kim, Luis Muñoz, Ignacio Elicegui Maestro, Andrea Gaglione and Angelo Capossele: "Reference Architecture for IoT Enabled Smart Cities SynchroniCity: Delivering an IoT enabled Digital Single Market for Europe and Beyond".

[i.44] The British Standards Institution, PAS 182:2014: "Smart city concept model - Guide to establishing a model for data interoperability".

[i.45] Eurostat: "Urban Europe - statistics on cities, towns and suburbs - executive summary".

NOTE: Available at https://ec.europa.eu/eurostat/statistics-explained/index.php/Urban_Europe_-_statistics_on_cities,_towns_and_suburbs_-_executive_summary#People_and_life_in_cities.

[i.46] IEC 62264 (Parts 1 to 6): "Enterprise-control system integration".

[i.47] Industrial Internet Consortium: "A Practical Way to Get Started in Manufacturing IIoT: Cultivate a "Green Patch" in Your Brownfield".

NOTE: Available at https://www.iiconsortium.org/pdf/2017-11-14-Cultivate_a_green_patch_in_brownfield_whitepaper.pdf.

[i.48] IoT eclipse.org: "The Three Software Stacks Required for IoT Architectures" White paper.

NOTE: Available at https://iot.eclipse.org/resources/white-papers/Eclipse%20IoT%20White%20Paper%20-%20The%20Three%20Software%20Stacks%20Required%20for%20IoT%20Architectures.pdf.

[i.49] IoT-O.

NOTE: Available at: https://www.irit.fr/recherches/MELODI/ontologies/IoT-O.html.

[i.50] Bain & Company: "Choosing the Right Platform for the Industrial IoT", 2018.

NOTE: Available at https://www.bain.com/insights/choosing-the-right-platform-for-the-industrial-iot/.

[i.51] "The Forrester Wave: Industrial IoT Software Platforms", Forrester, Q3 2018.

[i.52] IEC 61360: "Common Data Dictionary".

[i.53] EUROMAP 77: "OPC UA interfaces for plastics and rubber machinery - Data exchange between injection moulding machines and MES".

[i.54] Website of AVNU Alliance.

NOTE: Available at https://avnu.org.

[i.55] Website of BIG IoT.

NOTE: Available at http://big-iot.eu.

[i.56] Website of DATEX II.

NOTE: Available at https://datex2.eu.

[i.57] "VDMA supports developing OPC UA CS", presentation by Andreas Faath.

NOTE: Available at https://www.automaatioseura.fi/site/assets/files/1824/03_opc_finland_vdma_andreas_faath.pdf.

[i.58] "VDMA Overview of activities and companion specs", presentation by Andreas Faath.

NOTE: Available at https://jp.opcfoundation.org/wp-content/uploads/sites/2/2018/12/8_Faath_VDMA_OPCUA.pdf.

[i.59] "Wanted: A Plug-In Architecture for Hadoop Development", article by Alex Woodie, DATANAMI, May 6, 2015.

NOTE: Available at https://www.datanami.com/2015/05/06/wanted-a-plug-in-architecture-for-hadoop-development/.

[i.60] "Role of CPS in Manufacturing", presentation by Prof. Marco Taisch at Workshop "Platforms for connected Factories of the Future", 2015-10-05.

NOTE: Available at http://ec.europa.eu/information_society/newsroom/image/document/2015-44/6_taisch_11943.pdf.

[i.61] "AUTOPILOT The role of IoT interoperability in Smart Mobility", presentation by Mahdi Ben Alaya at oneM2M TP#37 Industry Day, 2018-09-14.

NOTE: Available at ftp://ftp.onem2m.org/Meetings/TP/2018%20meetings/20180914_Industry%20Day_TP37/Industry_Day-2018-0004-AUTOPILOT-The_role_of_IoT_interop_in_Smart_Mobility.PDF.

[i.62] "OPC Unified Architecture Interoperability for Industrie 4.0 and the Internet of Things", Version 10, November 2019, published by OPC Foundation.

NOTE: Available at https://opcfoundation.org/wp-content/uploads/2017/11/OPC-UA-Interoperability-For-Industrie4-and-IoT-EN.pdf.

[i.63] oneM2M TS-0001-V3.15.1: "Functional Architecture".

NOTE: Available at http://www.onem2m.org/images/files/deliverables/Release3/TS-0001-Functional_Architecture-V3_15_1.pdf.

[i.64] ETSI TR 118 518 V2.0.0 (2016-09) "oneM2M; Industrial Domain Enablement (oneM2M TR-0018 version 2.0.0 Release 2)".

NOTE: Available at https://www.etsi.org/deliver/etsi_tr/118500_118599/118518/02.00.00_60/tr_118518v020000p.pdf.

[i.65] oneM2M TR-0018-V4.0.0: "Industrial Domain Enablement".

NOTE: Available from http://www.onem2m.org/technical/published-drafts/release-4 at http://member.onem2m.org/Application/documentapp/downloadLatestRevision/default.aspx?docID=29334.

[i.66] "Server Side Public License (SSPL)", Copyright © 2018 MongoDB, Inc.

NOTE: Available from https://www.mongodb.com/licensing/server-side-public-license.

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

AVNU Alliance: community creating an interoperable ecosystem servicing the precise timing and low latency requirements of diverse applications using open standards through certification [i.54]

copyleft: practice of offering people the right to freely distribute copies and modified versions of a work with the stipulation that the same rights be preserved in derivative works created later

cyber security (or cybersecurity): collection of tools, policies, security concepts, security safeguards, guidelines, risk management approaches, actions, training, best practices, assurance and technologies that can be used to protect the cyber environment and organization and user's assets

DATEX: European standard for the exchange of traffic information and traffic data [i.56]

IoT LSP: Internet of Things Large-Scale Projects, funded under the IoT European Large-Scale Pilots (LSP) Programme

NOTE: The EU-funded IoT Large-Scale Pilots Programme (LSP) comprises a total of seven innovation consortia (5 LSPs and 2 Communication Support Actions), working hand in hand to foster the uptake of Internet of Things (IoT) in industrial sectors in Europe and beyond within the European IoT Pilot working group.

information models: representation of concepts and relationships, constraints, rules, and operations to specify data semantics for a chosen domain of discourse

oneM2M: Partnership Project formed in 2012 and consisting of eight of the world's preeminent standards development organizations (SDOs) notably: ARIB (Japan), ATIS (United States), CCSA (China), ETSI (Europe), TIA (USA), TSDSI (India), TTA (Korea) and TTC (Japan) together with GlobalPlatform

open source license: type of license for computer software and other products that allows the source code, blueprint or design to be used, modified and/or shared under defined terms and conditions

NOTE: Examples of popular Open Source licenses are: Apache License 2.0, GNU General Public License (GPL) or Eclipse Public License.

Open Source Software (OSS): computer software that is available in source code form

NOTE: The source code and certain other rights normally reserved for copyright holders are provided under an Open Source license that permits users to study, change, improve and at times also to distribute the software.

SERCOS: user organization for the development of standards, an officially recognized partner of the Industrial Electrotechnical Commission (IEC)

NOTE: See https://www.sercos.org/organization/.

source code: any collection of computer instructions written using some human-readable computer language, usually as text

standard: output from an SSO

Standards Setting Organization (SSO): any entity whose primary activities are developing, coordinating, promulgating, revising, amending, reissuing, interpreting or otherwise maintaining standards that address the interests of a wide base of users outside the standards development organization

3.2 Symbols

Void.

3.3 Abbreviations

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

3GPP	3rd Generation Partnership Project
AAA	Authentication, Authorization and Accounting
AAS	Asset Administration Shell
ADN	Application Dedicated Node
AE	Application Entity
AGILE	Adaptive Gateways for dIverse muLtiple Environments IoT Project
AGPL	Affero General Public License
AI	Artificial Intelligence
AIDC	Automatic Identification and Data Capture
AIOTI	Alliance for the Internet of Things Innovation
AM	Application Master
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
ARIB	Association of Radio Industries and Businesses
ASN	Application Service Node
ATIS	Alliance for Telecommunications Industry Solutions
AWS	Amazon Web Services
B2B	Business-to-Business
BAN	Body Area Network
BBF	Broadband Forum
BIG IoT	Bridging the Interoperability Gap of the Internet of Things (EU H2020 Project)

NOTE: See http://big-iot.eu [i.55].

BSI	British Standards Institution
CAN	Controller Area Network
CAPEX	Capital Expenditure
CCSA	China Communications Standards Association
CDD	Common Data Dictionary
CIM	Context Information Management
CIP	Common Industrial Protocol
CoAP	Constrained Application Protocol
CPPS	Cyber-Physical Production Systems
CPS	Cloud Service Provider
CPU	Central Processing Unit
CRUDN	Create, Retrieve, Update, Delete, Notify
CSA	Coordination and Support Action
CSE	Common Services Entity
CSF	Common Services Function
CSP	Cloud Service Provider
DCS	Distributed Control Systems
DDS	Data Distribution Service
DEI	Digitizing European Industry
EC	European Commission
EIF	European Interoperability Framework
EIP-SCC	European Innovation Partnership on Smart Cities and Communities
ERP	Enterprise Resource Planning
ETSI	European Telecommunications Standards Institute

FDIS	Final Draft International Standard
GDPR	General Data Protection Regulation
HDFS	Hadoop Distributed File System
HLA	High Level Architecture
HTTP	Hypertext Transfer Protocol
HVAC	Heating, Ventilation and Air Conditioning
IaaS	Infrastructure as a Service
IACS	Industrial Automation and Control Systems
ICT	Information and Communication Technology
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IERC	European Research Cluster on the Internet of Things
IETF	Internet Engineering Task Force
IG	Industry Group
IIoT	Industrial IoT
IN	Infrastructure Node
IoT	Internet of Things
IoT-EPI	IoT-European Platforms Initiative
IP	Internet Protocol
IPC	Industrial PC
IPR	Intellectual Property Rights
IRI	International Resource Identifier
ISA	International Society of Automation
ISG	Industry Specification Group
ISO	International Organization for Standardization
IT	Information Technology
ITS	Intelligent Transport Systems
ITU	International Telecommunication Union
ITU-T	ITU Telecom sector
JDBC	Java Database Connectivity
JSON	JavaScript Object Notation
KaaS	Kubernetes as a Service
KPI	Key Performance Indicator
LAN	Local Area Network
LOV	Linked Open Vocabularies
LOV4IoT	Linked Open Vocabularies for Internet of Things
LSP	Large Scale Pilot
M2M	Machine-to-Machine
MES	Manufacturing Execution System
MESA	Manufacturing Enterprise Solutions Association
MN	Middle Node
MOM	Manufacturing Operations Management
MQTT	Message Queuing Telemetry Transport
OAG	Open Applications Group
OAGIS	OAG Integration Specification
OASC	Open and Agile Smart Cities
OCF	Open Connectivity Foundation
OCP	Open Core Protocol
ODVA	Open DeviceNet Vendor Association
OIC	Open Interconnect Consortium
OLE	Object Linking and Embedding
OMA	Open Mobile Alliance
OP	OPerations
OPC	Open Platform Communications
OPC DA	OPC Data Access
OPC UA	OPC Unified Architecture
OS	Operating System
OSS	Open Source Software
OT	Operational Technology
OWL	Web Ontology Language
PaaS	Platform as a Service
PAN	Personal Area Network

PAS	Publicly Available Specification
PI4.0	Platform Industry 4.0
PID	Proportional, Integral and Derivative
PLC	Programmable Logic Controller
PoC	Proof-of-Concepts
PSS	Product-Service Systems
QoS	Quality of Service
RAMI	Reference Architecture Model (for Industrie 4.0)
RDF	Resource Description Framework
REST	REpresentational State Transfer
RM	Resource Manager
RoI	Return on Investment
RS	Recommended Standard
SaaS	Software as a Service
SAREF	Smart Applications REFerence Ontology
SCADA	Supervisory Control and Data Acquisition
SDK	Software Development Kit
SDO	Standard Development Organization
SDT	Smart Device Template
SEMI	Semiconductor Equipment and Materials International
SLA	Service Level Agreement
SME	Small and Medium Enterprise
SQL	Structured Query Language
SSO	Standards Setting Organization
SSPL	Server Side Public License
SW	SoftWare
SWoT	Semantic Web of Things
TC	Technical Committee
TCP/IP	Transmission Control Protocol/Internet Protocol
TIA	Telecommunications Industry Association
TIM	Telecom Italia Mobile
TRL	Technology Readiness Level
TSDSI	Telecommunications Standards Development Society, India
TTA	Telecommunications Technology Association
TTC	Telecommunication Technology Committee
UML	Unified Modeling Language
URI	Unique Resource Identifier
VDMA	The German Mechanical Engineering Industry Association
W3C	World Wide Web Consortium
WAN	Wide Area Network
WG	Work Group
WoT	Web of Things
WSN	Wireless Sensor Network
XML	eXtensible Markup Language

4 Platforms Interoperability in the context of IoT

4.1 A global approach to IoT Systems

4.1.1 Major characteristics of IoT systems

IoT systems are often seen as an extension to existing systems needed because of the (potentially massive) addition of networked devices. However, this approach does not take stock of a set of essential characteristics of IoT systems that push for an alternative approach where the IoT system is at the centre of attention of those who want to make them happen. This advocates for an "IoT-centric" view.

Most of the above-mentioned essential characteristics may be found in other ICT-based systems. However, the main difference with IoT systems is that they all have to be dealt with simultaneously. The most essential ones are:

- **Stakeholders.** There is a large variety of potential stakeholders with a wide range of roles that shape the way each of them can be considered in the IoT system. Moreover, none of them can be ignored.
- **Privacy.** In the case of IoT systems that deal with critical data in critical applications (e.g. e-Health, Intelligent Transport, Food, Industrial systems), privacy becomes a make or break property.
- **Interoperability.** There are very strong interoperability requirements because of the need to provide seamless interoperability across many different systems, sub-systems, devices, etc.
- **Security.** As an essential enabling property for Trust, security is a key feature of all IoT systems and needs to be dealt with in a global manner. One key challenge is that it is involving a variety of users in a variety of Use Cases.
- **Technologies.** By nature, all IoT systems have to integrate potentially very diverse technologies, very often for the same purpose (with a risk of overlap). The balance between proprietary and standardized solutions has to be carefully managed, with a lot of potential implications on the choice of the supporting platforms.
- **Deployment.** A key aspect of IoT systems is that they emerge at the very same time where Cloud Computing and Edge Computing have become mainstream technologies. All IoT systems have to deal with the need to support both Cloud-based and Edge-based deployments with the associated challenges of management of data, etc.
- **Legacy.** Many IoT systems have to deal with legacy (e.g. existing connectivity, back-end ERP systems). The challenge is to deal with these requirements without compromising the "IoT centric" approach.

4.1.2 The need for an "IoT-centric" view

4.1.2.1 Introduction

In support of an "IoT-centric" approach, some elements have been used in the present document in order to:

- Support the analysis of the requirements, Use Cases and technology choices (in particular related to interoperability).
- Ensure that the target audience can benefit from recommendations adapted to their needs.

4.1.2.2 Roles

A drawback of many current approaches to system development is a focus on the technical solutions, which may lead to suboptimal or even ineffective systems. In the case of IoT systems, a very large variety of potential stakeholders are involved, each coming with specific - and potentially conflicting - requirements and expectations. Their elicitation requires that the precise definition of roles that can be related to in the analysis of the requirements, of the Use Cases, etc.

Examples of such roles to be characterized and analysed are: System Designer, System Developer, System Deployer, End-user, Device Manufacturer. Some of these roles are specifically addressed in the present document.

4.1.2.3 Reference Architecture(s)

In order to better achieve interoperability, many elements (e.g. vocabularies, definitions, models) have to be defined, agreed and shared by the IoT stakeholders. This can ensure a common understanding across them of the concepts used for the IoT system definition. They also are a preamble to standardization. Moreover, the need to be able to deal with a great variety of IoT systems architectures, it is also necessary to adopt Reference Architectures, in particular Functional Architectures. The AIOTI High-Level Architecture [i.8] will be referred to in the present document.

4.1.2.4 Guidelines

The very large span of requirements, Use Cases and roles within an IoT system make it difficult to provide prototypical solutions applicable to all of the various issues addressed. The approach taken in the present document is to outline some solutions but also to provide guidelines on how they can be used depending on the target audience. Such guidelines are associated to the relevant roles and provide support for the decision-making.

4.2 Main objectives of the present document

A very large number of IoT platforms have been developed with the initial purpose of ensuring that a device could interact with other devices or equipment, providing connectivity from point-to-point to more universal. In the first place, Standard Development Organizations (SDOs) and Standard Setting Organizations (SSOs) have worked on standards focusing on interoperability, initially at the network level and gradually addressing the upper layers. Many of the resulting standards have been used as a basis for the development of platforms. Some of these platforms have been developed upfront with the objective to deal with interoperability as generically as possible, across several business sectors, allowing a variety of competing implementations. Such "standardized platforms" are relying on standardized components such as reference architectures, interoperability stacks, generic protocol adaptors, etc.

IoT is maturing fast, in particular with the emergence of more capable IoT producing massive amounts of data. This has transformed the IoT systems and created new challenges linked to the collection, transformation, storage and management of this data in larger (and more and more non IoT specific) information systems. This evolution comes with new major challenges for IoT platforms: the support to Big Data, the use Virtualization technologies and deployment models and the usage of Open Source Software (OSS) components that challenge the "traditional" role of standards. It is quite likely that only a part of the existing platforms will be able to successfully go through these changes.

The analysis of these new challenges for "standardized IoT platforms" is the main objective of the present document.

4.3 Purpose and target group

The present document addresses the topic of interoperability in the context of IoT platforms, in particular how far interoperability is supported by standards, and how this support is embedded in a platform that can be used across a variety of business sectors and Use Cases. The underlying question is the definition and the identification of "standardized platforms" that can be in the same time flexible, open, multi-purpose and rely on a set of well-defined standards. Ultimately, the question posed to these "standardized platforms" is how they can evolve better to serve the needs of a large number of IoT systems and can be chosen by the initiators of IoT systems and adopted by a large community of designers and developers.

The target group for the present document is the community of people that design, develop, implement and validate IoT systems, that have to make use of a limited number of technical platforms, and that have to make sure that the platforms they chose/use are supported by standards and offer the greatest possible support of interoperability at all levels.

4.4 Content of the present document

Clause 5 is addressing the very fragmented IoT platforms landscape. It outlines some requirements that should be met by the main IoT platforms in order to expand their capabilities and attractiveness to IoT systems designers and developers. The impact of two major evolutions, namely Big Data and Virtualization, on these platforms is analysed. The overall objective is to better characterize what are the properties that a "standardized IoT platform" should embed in order to become a major reference to the developers of IoT solutions in various business sectors.

Clause 6 is addressing a special attention to interworking, across all layers of the interoperability stack (from technical to organizational). It analyses the technical approaches in support of interoperability and outlines some criteria for best support of interoperability within and between platforms. Based on these criteria, a list of "candidate platforms" is established and an evaluation of the actual support of these criteria by the identified platforms is made.

Clause 7 is presenting Industrial IoT (IIoT) as a typical case study of the many challenges that are posed to standardized platforms. Beyond the identification of major requirements, it addresses some challenges such as the role of legacy and its impact on candidate platforms. Based on these requirements, a list of potential platforms is provided. Some of them are analysed in order to evaluate their coverage and what should be done to overcome potential limitations.

Clause 8 is presenting some lessons learned from the above analysis, in particular from the in-depth analysis of the case of Industrial IoT. Based on these lessons, some recommendations are made towards the IoT community regarding standardization, convergence of platforms, interoperability support frameworks, etc. In particular, some recommendations are made to oneM2M.

Annex A is presenting the platforms that have been selected by the UNIFY-IoT project and the IoT-EPI in their analysis of the IoT platform landscape. These platforms are addressed in clause 5.

5 The IoT Platforms Landscape

5.1 A framework for IoT Platforms

5.1.1 Expectations and definition

The Internet of Things (IoT) has emerged with the intention to support the massive deployment of a very large range of devices within new or existing systems that would allow the development of associated services. In this perspective, an IoT platform cannot be seen only as an execution environment specialized in IoT devices. Indeed, IoT systems have to deal with a (potentially very) large number of (potentially very) heterogeneous IoT devices with very fast evolving underlying technologies. Secondly, the main expectation of IoT is that it will allow for the fast and cost-effective development and deployment of new applications and services.

The expectations for an IoT platform are high: it is supposed to mask the heterogeneity of devices, to handle and simplify communication, to support (end-to-end) data flows, and to provide generic services to the applications built on top of it. IoT applications will see an IoT platform as a framework that will connect devices, gateways and machines, applications, and users; and will potentially span the entire value chain of an end-to-end IoT system.

Beyond the provision of a very large number of point solutions in the early days of IoT, the notion of IoT platform has emerged as a key building block to better support the development of IoT systems, in particular as a "mediation" between the needs of the IoT devices and those of the applications and services supported by corresponding architectural layers.

The definition of the IoT European Platforms Initiative (IoT-EPI) is adequately reflecting the "mediation" approach in its White Paper [i.6]:

- An IoT Platform can be defined as an intelligent layer that connects the things to the network and abstract applications from the things with the goal to enable the development of services. […] An IoT platform facilitates communication, data flow, device management, and the functionality of applications. The goal is to build IoT applications within an IoT platform framework.

The above definition can be put in perspective with the AIOTI Work Group 03 High-Level Functional Model [i.8] which is composed of three layers as depicted in Figure 1:

- The Application layer that contains the communications and interface methods used in process-to-process communications.
- The IoT layer that groups IoT specific functions, such as data storage and sharing, and exposes those to the application layer via interfaces commonly referred to as Application Programming Interfaces (APIs). The IoT Layer makes use of the Network layer's services.
- The Network layer that provides services which can be grouped into data plane services, providing short- and long-range connectivity and data forwarding between entities, and control plane services such as location, device triggering, QoS or determinism.

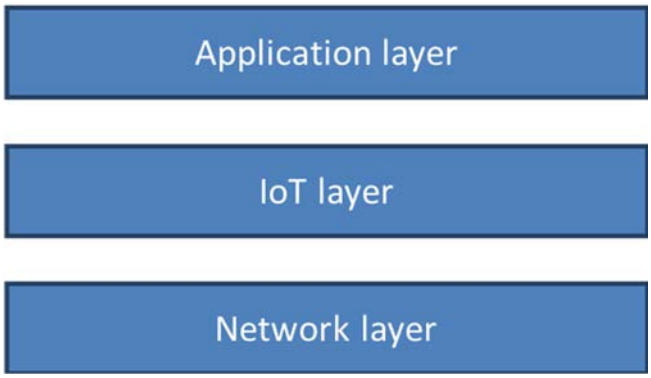


Figure 1: AIOTI 3-Layer Functional Model

An IoT system has to span the entire value chain from devices to applications but the "mediation" provided by the IoT platform will be concentrated in the IoT layer. Consequently, the IoT layer will be providing different kind of services (e.g. connectivity, processing of data, cloud and edge deployment) that can themselves be structured into sublayers.

5.1.2 Challenges

5.1.2.1 Flexibility, versatility

The emergence of the IoT platforms comes with a number of challenges regarding the current approach to the developers of IoT systems and more generally to the IoT community. IoT platforms are more and more seen as a way to handle two concurrent trends.

One the one hand, IoT systems are facing very complex and pressing requirements (e.g. number of components, volume of data, latency or reliability, security and privacy) that tend to drastically increase their complexity.

On the other hand, some new approaches are emerging and are being validated (e.g. layered architectures; new interoperability frameworks; marketplaces; Semantic Interoperability; virtualization) that are likely to reduce systems complexity; to provide more flexible applications development; and to open the development of systems to a larger number of stakeholders with new value propositions.

The possibility to address concurrently address the above requirements will put a lot of stress on the IoT platforms in terms of architectures, technological choices, requirement coverage or development costs. It is quite likely that only a limited number of IoT platforms will emerge as main contenders on the market.

5.1.2.2 Semantic Interoperability

The focus of interoperability has been initially on technical interoperability (basic connectivity, network interoperability) and syntactic interoperability (Common Information Models with static information based on a pre-defined syntax). This was reflected in the work of standardization with many great achievements.

However, as soon as the requirement on the information exchanged become more complex (e.g. systems from different sectors), static information is no longer sufficient, and the need arise for basing the exchange of information on its meaning (independently of underlying protocols). This is the role of Semantic Interoperability: making sure that the meaning of semantics can be understandable and processed by machines, and most common way to achieve this is by using an ontology which is "an explicit specification of a [shared] conceptualization" [i.6]. The AIOTI has defined a framework for IoT Semantic Interoperability [i.9] with the following conclusion:

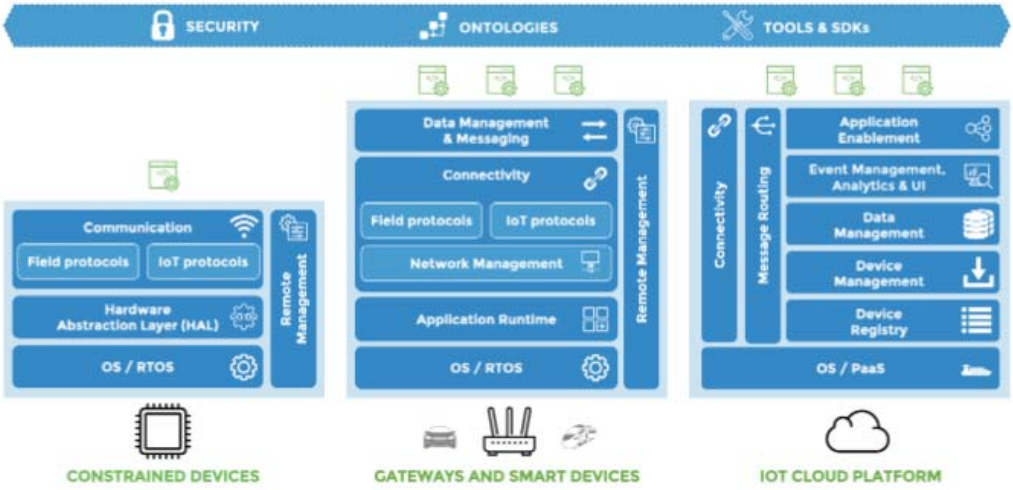
- "*[…] semantic approaches will expose many firms and individual engineers to new interoperability architectures and will require changes in tools, technologies, and thought processes.*"

Two companion documents are addressing this issue:

- ETSI TR 103 535 [i.3]
- ETSI TR 103 537 [i.4]

5.1.2.3 Flexible deployment models

In the case of IoT systems, the coexistence of different kinds of architectures and associated deployment models will make interoperability a more difficult target. A typical IoT system will encompass a variety of elements, devices, gateways and (potentially cloud-based) platforms. The implementation of such elements will be constrained by their capabilities (e.g. computing power) and will rely on a more or less complex IoT stack. The model proposed by www.eclipse.org/ in Figure 2 illustrates three typical stacks: for constrained devices; for gateways and smart devices; and for cloud platforms.



Source: eclipse.org [i.48].

Figure 2: The Three IoT Software Stacks

These stacks offer a growing level of functionality as the capabilities of the elements that run the stack is growing. They all use cross-stack functionality related to security (in particular for authentication, encryption, and authorization); development tools and Software Development Kits SDKs (supporting the different hardware and software platforms involved); and ontologies.

5.1.2.4 Open and efficient implementations.

Openness is a clear requirement for IoT platforms that are based around a combination of layers, interoperability solutions, APIs, etc. thus allowing the assembly of the most appropriate solution for every part of the IoT system in a flexible manner. Such open and flexible architectures have to be supported by a large catalogue of effective, easily available and possibly certified components available through marketplaces. A number of these components will be provided by the Open Source Software (OSS) communities.

5.1.2.5 Non-functional properties

Most of the new IoT systems will have to be extremely effective from a number of non-functional properties such as latency, reliability, near real-time handling of massive quantities of data, etc. The introduction of techniques such as Cloud or Edge computing, microservices-based architectures will have to be supported by IoT platforms and standards.

5.1.2.6 Security

Security is a key enabler for trusted IoT systems. From this standpoint, there is no silver-bullet solution for security in the current (IoT) systems. The current approach in most of the IoT platforms is to provide a (standards-based) support for solutions at each layer of the IoT system HLA, (e.g. in order to support authentication and authorization). But, ensuring global, cross-layer is a complex task. It can be debated whether or not IoT is a distinct security problem compared to other ICT systems. In any case, in IoT, there may be more uncertainties to be resolved than in conventional centrally managed security systems due to complex structure of the systems noted above (e.g. heterogeneity and large number of - often not well protected - devices).

Two companion documents are addressing this issue:

- ETSI TR 103 533 [i.1]
- ETSI TR 103 534 [i.2]

5.1.2.7 Privacy and data confidentiality

Privacy is universally pointed out as a key enabler for trusted IoT systems. It has long been considered as the "poor cousin" in the industry, until a repeated number of failures due to the non-acceptance of systems for privacy reasons has brought the issue on top of the requirements list. The coming into force of the General Data Protection Regulation (GDPR) has been an important motivating factor for the (IoT) community to embrace the issue. GDPR forces a reconsideration of the current approaches for personal data (and identification) protection and related security. From a process-centric approach, the systems supporting GDPR should switch to a more user-centric and data-centric approach.

Two companion documents are addressing this issue:

- ETSI TR 103 591 [i.5]
- ETSI TR 103 534 [i.2]

5.1.2.7 Integration with legacy

Only a limited number of IoT systems can be fully seen as greenfield, built from scratch. Most of them have (or will have) to incorporate existing (and sometimes long existing) elements. The introduction of new techniques within the entire system may not be possible, often for cost reasons and also because of difficulties related to old or unmaintained technologies. The potential coexistence of old and new parts (the latter based on those new approaches, e.g. interoperability patterns or Semantic Interoperability) will require, sometimes complex, adaptations.

5.2 An IoT Platforms Landscape

5.2.1 Fragmentation and lack of maturity

In the past several years, as IoT was gradually taking-off, a number of platforms have started to emerge, of different types depending on a number of factors such as the nature of the devices, the scope and breadth of the application(s), the business sector, etc. As a result, there are literally hundreds of IoT platforms available for the development of whole or parts of IoT systems.

These platforms have very different coverage (from point solutions to complete Enterprise systems), different maturity and development status or user adoption level. Moreover, they can also have very different support to interoperability and to the standards in support of it (from fully proprietary solutions to open standardized platforms).

Hence, the most obvious aspect of the current platform landscape is its great fragmentation. Additionally, this plethora is also the signal of a certain lack of maturity with a lot of solutions developed as an ad-hoc point answer to a very specific question. Many of these solutions have not reached a level of maturity (e.g. TRL 9) that will guaranty that they are long-term solutions. There may be multiple reasons for this, amongst which unstable business models that cannot support steady development costs or an insufficient ecosystem of developers.

5.2.2 A typology of platforms

5.2.2.1 Main dimensions for platform analysis

The main difficulty for the choice of platform(s) will be to carefully analyse them with appropriate selection criteria. Several dimensions have to be considered amongst which:

- Scope and breadth: which kind of business sector and solution will the platform support?
- Openness: how is a platform going to comply with openness criteria such as those that define the work of standardization or open source communities?

- Origin and governance: which entity is in charge of the definition of the platform and its evolution?
- Ecosystem: has the platform attracted a number of partners that can participate to the extension of its footprint?
- Maturity: how far can the platform support the implementation of effective and efficient implementations?

Each of these criteria is discussed in the following clauses and a classification scheme, based on these criteria, is proposed in clause 5.2.2.7.

5.2.2.2 Scope and breadth

Some of the existing IoT platforms may address a specific problem or a limited technical environment, offering a point solution addressing a part of the IoT stacks. On the other hand, some platforms can be very general purpose ("generic") and integrate the IoT system in a larger (Enterprise) system. The chances are that a "generic" platform (or component) will benefit from a larger business potential, of a larger ecosystem of developers than a "specific" one.

The potential advantage of platforms with large breadth over point solutions has an important consequence on the integrability (i.e. the ease of adding new functional components) of the IoT system. For the development of an IoT system, the resources should be used devoted to the development of the services rather than to the integration (possibly repeated across different network settings) with protocols, data models, APIs, clouds, servers, etc. Platforms with large breadth will address these integration issues in a more systematic way (e.g. with generic protocol adaptors).

Similarly, generic platforms will have a potential advantage over (sector-) specific ones for at least two reasons. Firstly, they offer solutions that can be applied in the same manner across different "verticals" (e.g. the generic protocol adaptor mentioned above, or the way to integrate functionality from Cloud Service Providers) and will not generate excessive integration effort. In addition, they allow to better prepare the future integration of function and sub-systems coming from another sector (with the example of Smart Cities that have to deal with the need to gradually integrate different sub-systems).

5.2.2.3 Openness

Openness criteria

There is no single definition widely adopted for openness since the criteria are on the one hand, differing between the Standards community and the Open Source community. Within the standards community, the view may also differ from one SDO/SSO to another one (e.g. ETSI, ITU-T or IEEE). However, some criteria are generally seen as mandatory for the "open" label to be granted to the outcome of the process.

Amongst them are the following ones (that apply specifically to standards but also to Open Source to a very large extent):

- The process should be collaborative and follow a transparent consensus-driven approach that is open (reasonably) to all interested parties.
- The process should be balanced (reasonably) and should prevent from the domination by any interest group.
- The process should take into account comments by interested parties and answer to them.
- The outcome of the process displays a quality and a level of detail that is enough to allow for the development of a variety of competing implementations of interoperable products or services; when applicable, the standardized interfaces are controlled by the organization (SDO/SSO, Open Source project) in charge.
- The outcome of the process is publicly available at a reasonable price.
- The outcome of the process is maintained and supported over a long period of time.

Additional considerations can be done regarding the management of Intellectual Property Rights (IPRs) but they are not relevant in the context of the present document.

Openness, Standards and Open Source

As pointed above, there is a certain similarity between Standards and Open Source regarding openness criteria. The adherence to this criterion will strongly influence the classification of platforms (developed in clause 5.2.2.7) by clearly separating the "open" platforms from those which are not, in particular the vast majority of "proprietary" platforms.

5.2.2.4 Origin and governance

The provision of functionalities and solutions can have different origins and come from different actors with a very different set of objectives (e.g. regarding monetization). Each of the models developed below has some characteristics (participants ecosystem, governance, pricing and licensing, etc.) and different benefits.

Standards-based.

A standardized platform is referring to the development by an SDO or SSO of (paper-based) specifications (with additional interoperability support such as plugtests). A standardized platform will typically encompass the description of a Reference Architecture (with potentially several models, the most frequently used being the Functional Model), a set of supported protocols, a set of interfaces (in particular Application Programming Interfaces), etc. The main advantages of a standardized platform are that it allows for multiple implementations, offers controlled interfaces, provable and proven interoperability, and maintenance over time with transparent control over the evolution of the features.

Beyond the general description above, there are some differences whether the platforms emanate from SDOs, SSOs as described below:

- SDO originated platforms. These solutions are typically joint efforts from very different players, ranging from big industry to small-medium size organizations, often including universities and consumer associations, and government entities. These standards are developed openly with clear and fair IPR rules, and typically are not controlled by any specific company or group of companies. The resulting standards may be officially recognized - and even mandated - by governmental entities. This approach has been extremely successful in different environments, the example of mobile telecommunications being the most famous one. In the IoT domain, oneM2M is the most prominent example.

- SSO originated platforms. The Standard Setting Organizations have a well-established tradition in setting standards (e.g. IETF). Their standards often become relevant because of global market recognition. However, their standards will not necessarily get the official recognition by government entities (unlike SDOs). In the IoT domain, only parts of such platforms exist.

Open Source-based.

An open source platform is referring to the development by an Open Source community or ecosystem of a set of (source code-based) software components. An open source platform will typically encompass the provision of source code and documentation attached to a configuration and version management framework.

The main expected advantages of an Open Source platform are that it is likely to offer controlled interfaces, some support for proof of interoperability, maintenance over time with transparent control over the evolution of the features.

An important aspect when using open source is the licensing. Open Source licenses are organized in two main categories: copy-left and business-friendly licenses:

- Under a copyleft license, users have to copy, distribute, and update the software under the same license as the original software. Copyleft clause is assumed to have an automatic effect that can lead to "contamination" (that any software combined with copyleft licensed software could somehow be transformed to be licensed under a copyleft license). Copyleft supporters are concerned with ensuring that their work remains available to everyone.

- By contrast, business-friendly licenses do not restrict the licenses under which these acts can be done and do not cause any "contamination" effects. Business-friendly license supporters believe the licensing restrictions mean that the copyleft is not a free license and their alternative encourages the use of free software.

Another aspect of licenses is that they can change over time, thus potentially inducing significant consequences on the technical strategy of those who have adopted the associated product.

A recent example is that of an Open Source Data Base Management System:

- Until recently, it was released under GNU AGPL License v3.0, and the drivers under Apache License v2.0: this approach was "business friendly" towards application programs using the database, while remaining copyleft for changes to the database engine itself (a much less common occurrence).

- Since October 2018, it is licensed under a new, expressly devised, license. This new license requires explicitly that anybody who wants to offer it as a service needs to either get a commercial license or open source the service to give back the community; this requirement is extended to all the software that is used to make it "*available as a service, including, without limitation, management software, user interfaces, application program interfaces, automation software, monitoring software, backup software, storage software and hosting software, all such that a user could run an instance of the service*" [i.66].

- Many are questioning now whether this license can still be considered an Open Source license. Several major Linux distributions, in the meantime, have removed it from their Linux distributions, while a big Company launched a database designed to be compatible with existing Open Source Data Base Management System applications and tools.

Licensing is a key aspect, not to be underestimated, for the choice and usage of any software product and solution that is based on open source components.

This is even more important since, in emerging technology markets, the adoption of Open Source approaches is increasingly the rule, not the exception.

Industry Group-based.

An Industry Group-based platform refers to the development by an Industry Group (IG) of (paper-based) specifications. The specification can come-up with a reference implementation developed by some of the IG members (that may be proposed as Open Source sometimes during the platform development).

Such solutions are typically organized around the solutions of a leading company or a set of leading companies, in an attempt to enlarge the ecosystem and reach market recognition. These specifications are often (but not always) developed openly with clear and usually fair IPR rules, even though they tend to be associated with specific technology players. In most of the cases these groups develop around specific protocols, API and data models, and they limit the specification to IoT components and in some cases frameworks for platform development specifying platform solutions (e.g. OCF).

Proprietary.

A proprietary platform is referring to the provision by a company (ranging from SME to Multinational Corporations) of a set of (executable) software components and documentation provided under the form of standalone software or as Software as a Service (SaaS) over a public, private or hybrid cloud.

Beyond the general description above, there are some differences whether the addition of some support to openness:

- Closed specifications. They are fully integrated and developed in house, and rely on a closed ecosystem of providers, developers and integrators. It is very common to find specialized solutions for specialized components of the IoT systems, and sometimes in some vertical IoT business sectors. This is also the case of large providers of more traditional solutions, that are now being positioned in the market in a way to take advantage of the current interest in IIoT technology. An example is the case of a proprietary middleware product for which, due to lack of proper documentation, some OEMs willing to extend and customize some functionality had to resort to the use of communication protocols sniffing in order to understand inner system working.

- Open Specifications. Such solutions are based on a proprietary solution (that may include components based on existing standards or adapted from existing standards) initially developed by a single organization (often a very large corporation) and further opened to an ecosystem of other companies and players. The intention is to create a market or to accelerate its consolidation, and to steer the market towards their solution. This approach tends to confer a major role to the developer of the proprietary solution, thus making the market dependent upon it. The success of such an approach is depending on the ability to quickly become a de facto standard for the market.

5.2.2.5 Ecosystem

The available platforms may have different sorts of ecosystems with a variety of situation regarding, in particular, the number of entities and individuals involved. In considering the ecosystem associated to a platform, the following elements have to be considered:

- Nature of the supporting group of entities in the addressed market. This question is applicable to standards-based, Open Source-based or industry-based platforms. In all cases, a number of entities (very often companies) have decided to work together. Important elements to consider are in particular the relative weight of the associated entities with respect to those operating in the domain, and the degree of openness to a relevant set of competing entities.

- Number of developers of the standard, the Open Source components, etc. The number of individuals that develop the solution (e.g. standard or Open Source) will have a direct impact on the richness of the result, on the possibility to promote it towards a large audience and to see it adopted by a large number of practitioners and become a de facto reference (not forgetting that it may also lead to over specification).

- Ability to address different sectors. The platforms that can be used beyond their sector of origin (this is true also for the platforms originating from the ICT industry) have the possibility to support a larger set of business cases (when IoT has to span across several sectors like, for instance, in Smart Cities), and to reduce the overall development costs (by no doing the same integration tasks several times).

5.2.2.6 Maturity

The available platforms may have different development status, technical readiness levels and user adoption level. In the quickly evolving landscape of IoT, it is possible to outline three main phases regarding the maturation of available platforms. Those phases are not fully clear-cut and sometimes may overlap, but they can illustrate different levels of maturity reached by the IoT community:

- Phase 1: **Burgeoning.** In the early times of the emergence of IoT as a potentially huge new market, a lot of experiments and prototypes have been launched with, in most case, an ad-hoc selection of platforms (or platform elements). Given the large number of early deployments, their often-limited scope and the focus on functional rather than non-functional requirements, a huge number of candidate platforms have been used. Many of the platforms identified by Unify-IoT (in clauses 5.2.3.2 and A.1) have been active during this phase. Only part of Phase 1 platforms has made their way in the list of platforms in Phase 2 and are still alive and relevant in Phase 3.

- Phase 2: **Positioning and specialization.** After the early phase of platforms usage, the requirements put on them have started to become more complex and stringent. In particular, these new requirements have started to address non-functional properties related to scalability, reliability, etc. The platforms identified by the IoT-EPI (in clauses 5.2.3.3 and A.2) have been actively used during this phase, some have been further developed or only subject to minor adjustments (e.g. interoperability-related). Only part of Phase 2 platforms have made their way in the list of platforms in Phase 3.

- Phase 3: **Consolidation.** With the consolidation of the market comes the consolidation of the IoT platforms. The number of platforms that will have the credibility to provide solutions for large and complex IoT systems development - potentially across different business sectors - is starting to drastically reduce in each of the categories identified in clause 5.2.2.4 (origin and governance). Notably, it is yet unclear how the proprietary platforms are shaping, with very different list of top players depending on the evaluations (see [i.51]). In particular, the fact that a platform is backed by a very large (multinational) corporation is not necessarily a guaranty for being successfully part of the consolidation (the example of the General Electric platform, GE Predix™, can be seen as an illustration of a change in strategy leading to a divertissement due to a reconsideration of the company expectations in the IoT market).

NOTE: Mention of Predix™ trademark in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with this trademark.

The consolidation will not only impact the number of platforms that will remain on the market. In particular, due to the nature of IoT services which require strong interoperability and the integration of information coming from different domains, the core solutions offered by the winning platforms will have to be common to different vertical business domains, thus reducing the space for specific vertical domain solutions dedicated to a specialized component of a common IoT system.

5.2.2.7 A classification of Platforms

Based in particular on the "origin and governance" criterion, a classification is proposed in Table 1.

Table 1: A classification of platforms

Type	Advantages	Drawbacks
SDO-based	• No dominant stakeholder • Open Source implementation availability • No dependency from a single company • Formal testing suites available • Global certification program available • Suitable for all the IoT services in the different region of the world • Strongly focused on interoperability • Strongly focused on integration of existing technologies • Global standardization • Competition on the platform is suitable for the users who reduce the associated costs	• A standard platform makes the platform a commodity • Competition on the platform is not suitable for the providers, who prefer to invest and focus on the IoT services
SSO-based	• There is usually an ecosystem of stakeholders representing the whole chain • Open Source solution often available, especially on device and gateway side • Some have certification programs • Some have global presence, even in vertical sectors	• Few of them are focusing on platform interoperability, while more are focused on protocol and devices, so integration effort if expected to be still predominant • There will be still a certain dependency from a specific ecosystem
Open Source-based	• No dominant stakeholder • Proven high TRL (e.g. TRL-9)	• Cover only parts of requirements • Limited focus on interoperability validation
Industry Group-based	• Usually reflect the needs of vertical sections of the industry • Usually well thought and helpful for the implementation of some interoperability interfaces • Sometimes no alternatives, either because of extremely widespread acceptance or because they are mandated by regulations in specific areas	• Cover only parts of manufacturers requirements • Need to be used in conjunction with other interoperability standards • May allow for specific extensions by individual manufacturers

The focus of the present document is on standardized and open platforms. Some examples of such platforms are described in more detail in clause 5.3.

5.2.3 Finding a way in the jungle of platforms

5.2.3.1 Introduction

The choice of platform(s) by IoT system designers, developers and validators may be very complex with potentially hundreds of IoT platforms available for the development of IoT systems, ranging from point solutions managing a part of the IoT stacks up to very general purpose that integrate the IoT system as a component in a larger system (e.g. Enterprise systems).

5.2.3.2 Platforms identified by UNIFY-IoT and the IoT-EPI

Guidelines for the identification and selection of IoT platforms may be useful to the IoT community. Two activities have been undertaken within the IoT Research and Innovation community, in particular by the UNIFY-IoT Coordination and Support Action (CSA) and the IoT European Platform Initiative (IoT-EPI) projects:

- The analysis of general technical literature done in UNIFY-IoT has defined criteria for the selection of platforms and led to the identification of 23 platforms IoT (see [i.10], [i.11] and [i.12]).
- The analysis of the platforms used in the IoT-EPI projects (see [i.6]) has led to the identification of 34 platforms.

More details on the corresponding platforms can be found in annex A.

5.2.3.3 Platforms in the IoT Large Scale Pilots

The five EU-funded IoT Large Scale Pilots (ACTIVAGE, AUTOPILOT, IoF2020, MONICA, SynchroniCity) are addressing the deployment of a (potentially large) number of Use Cases, in most cases across several pilot sites, in different domains (eHealth, Intelligent Transport Systems, Agriculture and Food, Entertainment, Smart Cities). They have to deal with different local situations, in particular with respect to the integration to the legacy systems.

In the context of the IoT Large Scale Pilots (LSPs) Use Cases, the issues related to the choice and usage of the platforms can vary from one LSP to another for a variety of reasons, such as:

- The requirements on cross-site interoperability may be more or less stringent.
- The support of connectivity or applications can induce specific choices.
- Local interworking with legacy applications can require pre-defined choices or the use of specific adaptors.

Two examples LSPs have been selected, in particular because of the relatively different nature of the criteria for platform choice:

- ACTIVAGE, a European Multi Centric Large-Scale Pilot on Smart Living Environments, building the first European IoT ecosystem across 9 Deployment Sites in seven European countries, reusing and scaling up underlying open and proprietary IoT platforms, technologies and standards, and integrating new interfaces needed to provide interoperability across these heterogeneous platforms.
- AUTOPILOT, a Large-Scale Pilot that develops new services on top of IoT to involve autonomous driving vehicles, like autonomous car sharing, automated parking, or enhanced digital dynamic maps to allow fully autonomous driving. IoT enabled autonomous driving cars are tested, in real conditions, at four Large-Scale Pilot sites.

ACTIVAGE

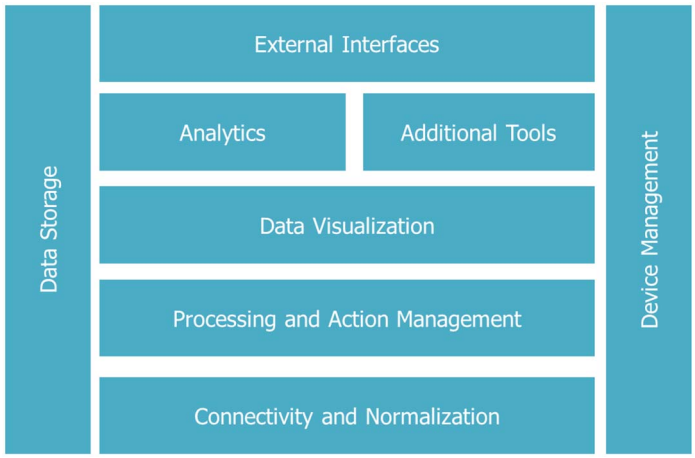
The IoT platforms provide the ability to create application and services in a structured environment, formed by proven existing functional components that are usually common and repeated across many IoT applications and services as shown in Figure 3.

This IoT platform functional components offer the following services:

- Connectivity & normalization.
- Device management.
- Data Storage.
- Processing & action management.
- Analytics.
- Visualization.

They are complemented by the following elements:

- Additional tools that allow IoT developers prototype, test and market the IoT Use Case.
- External interfaces that integrate with the wider IT-ecosystem via built-in Application Programming Interfaces (API), Software Development Kits (SDK), and gateways.



Source: Activage deliverable 3.1 [i.25].

Figure 3: Functional components of ACTIVAGE IoT platforms

For the ACTIVAGE project, the platform support is particularly focusing on the application layers, offering means to transform the information received from the devices and sensors into meaningful knowledge. Thus, the selection is based on platforms residing on the right or top part of Figure 3.

A second criterion is the availability of the platforms, since Open Source solutions are preferred to proprietary solutions. Thus, platforms such as Microsoft Azure™, IBM Watson™ or Oracle Integration Cloud™ have not been preferred.

NOTE: Mention of Microsoft Azure™, IBM Watson™ and Oracle Integration Cloud™ trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with this trademarks.

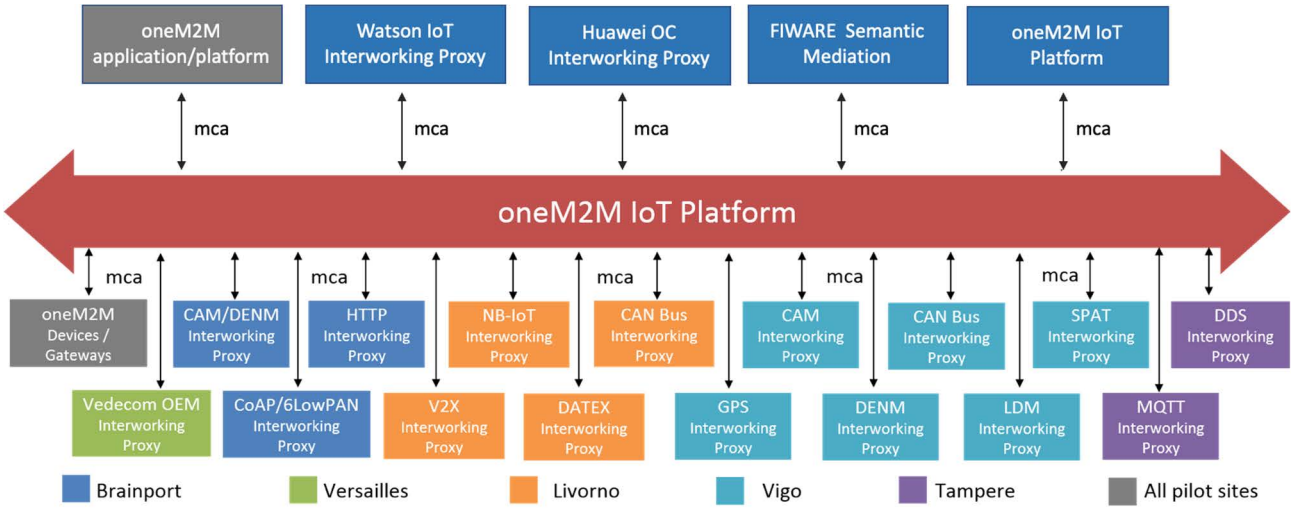
As a result, the following platforms, comprising a comprehensive set of Open Source, application-oriented IoT platforms, covering a relatively wide range of functionalities, are selected to be used in the ACTIVAGE project:

- FIWARE, an Open Source software platform that provides enhanced OpenStack-based cloud hosting capabilities plus a rich library of components bringing a number of added-value functions offered "as a Service".
- IoTivity, an Open Source software framework enabling seamless device-to-device connectivity.
- OpenIoT, an open source middleware for getting information from sensor clouds.
- Seniorsome, a proprietary platform to help seniors and dementia patients to stay at home longer safely.
- sensiNact, an Open Source software platform enabling the collection, processing and redistribution of any data relevant to improving the quality of life of urban citizens.
- Sofia2, an IoT and Big Data platform supporting the creation of new business models.
- universAAL, an Open Source platform provides the framework for communication, connectivity and compatibility between otherwise disparate products, services and devices.

AUTOPILOT

Figure 4 shows the platforms and technologies used for the implementation of the Use Cases for AUTOPILOT. For this LSP, the requirement of interoperability across pilot sites has led to the choice of oneM2M as a unifying IoT service platform. On top of this, the particularities of each pilot site (such as the nature of the application deployed or the legacy platforms) may lead to the choice of site-specific platforms and technologies that are interconnected via the oneM2M platform that serve as an interoperability backbone.

The AUTOPILOT IoT service platform is therefore a federation of several IoT platforms, allowing it to be open and flexible. An open oneM2M standard IoT platform, referred to as the "oneM2M interoperability platform", interconnects the pilot site specific proprietary IoT platforms provided by the project partners. These proprietary IoT platforms collect data from connected devices and exchange IoT data and events with the interoperability platform through oneM2M interworking proxies.



Source: AUTOPILOT [i.61].

Figure 4: The platforms across the AUTOPILOT Use Cases

Currently, several IoT platforms have been deployed for the pilot sites:

- FIWARE IoT platform, used in the Dutch pilot site.
- HUAWEI OceanConnect IoT platform, used in the Dutch pilot site.
- SENSINOV oneM2M platform, used in the Finnish, French and Dutch pilot sites.
- TIM oneM2M IoT platform, used in the Italian pilot site.
- Watson IoT Platform used in the Dutch and Spanish pilot sites.

5.2.3.4 Emerging approaches: Marketplaces and APIs

Over time, the IoT systems have matured and evolved towards layered, potentially cloud-based or edge-enabled architectures. In this new world of IoT systems, designers and developers expect that the myriad of devices that are deployed and connected to the network can seamlessly interoperate with a large range of platform services (e.g. data analytics, monitoring, visualization, etc.) and end-user/end-customers applications.

In the emerging model of market places, the actors involved in the provision of the IoT service are seen as consumers and providers within an application market. An IoT marketplace is a new platform to extend the "traditional" IoT platforms with brokerage concepts supporting automated discovery, trading and even pricing. Within an IoT marketplace platform, the IoT device owners will have the possibility to selectively grant access and trade their data with many potential vendors.

This is in general supported by:

- A number of publicly available Application Programming Interfaces (APIs) hiding the actual way the underlying provision of the service from the consumer of the service is performed. The implementation of the service can change without impacting the rest of the system and the evolution of the APIs can be mastered via the APIs publication mechanism.
- An approach based on microservices where any service (whichever its size and scope) can be published and consumed. This lean software development approach provides system flexibility and guaranties a faster adaptation to support emerging standards (or Open Source Software components provided by the OSS communities) without impacting the whole system architecture.

5.3 Standardized IoT Platforms

5.3.1 Characterization of Standardized IoT Platforms

A standardized platform is referring to the development of a set of components that support the development of a variety of (potentially competing) implementations. The components of the platform have been produced through a transparent and open development process in which all IoT stakeholders can participate.

These platforms can be provided, for example, by:

- An SDO or SSO that develop paper-based specifications (with additional interoperability support such as plugtests). An example is that of oneM2M described below.
- An Open Source Software ecosystem that develop software components with openly available source-code (with additional interoperability support such as Application Programming Interfaces). An example is that of Apache® described below.

NOTE: Mention of Apache® trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with this trademarks.

A standardized platform will typically encompass the provision of a number of elements such as:

- The description of a Reference Architecture (with potentially several models, the most frequently used being the Functional Model).
- A set of supported protocols.
- A set of interfaces or Reference Points (in particular Application Programming Interfaces).
- Etc.

The main advantages of a standardized platform are that it allows for multiple implementations, offers controlled interfaces, provable and proven interoperability, maintenance over time with transparent control over the evolution of the features.

5.3.2 oneM2M

5.3.2.1 Scope

oneM2M is an alliance of regional telecom SDOs (ETSI, TIA, ATIS, ARIB, TTC, TTA, CCSA and TSDSI), associated with industry forums such as GlobalPlatform, that operates similarly as 3GPP but with IoT as its focus. The purpose and goal of oneM2M is to develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software and relied upon to connect the myriad of devices in the field with M2M application servers worldwide.

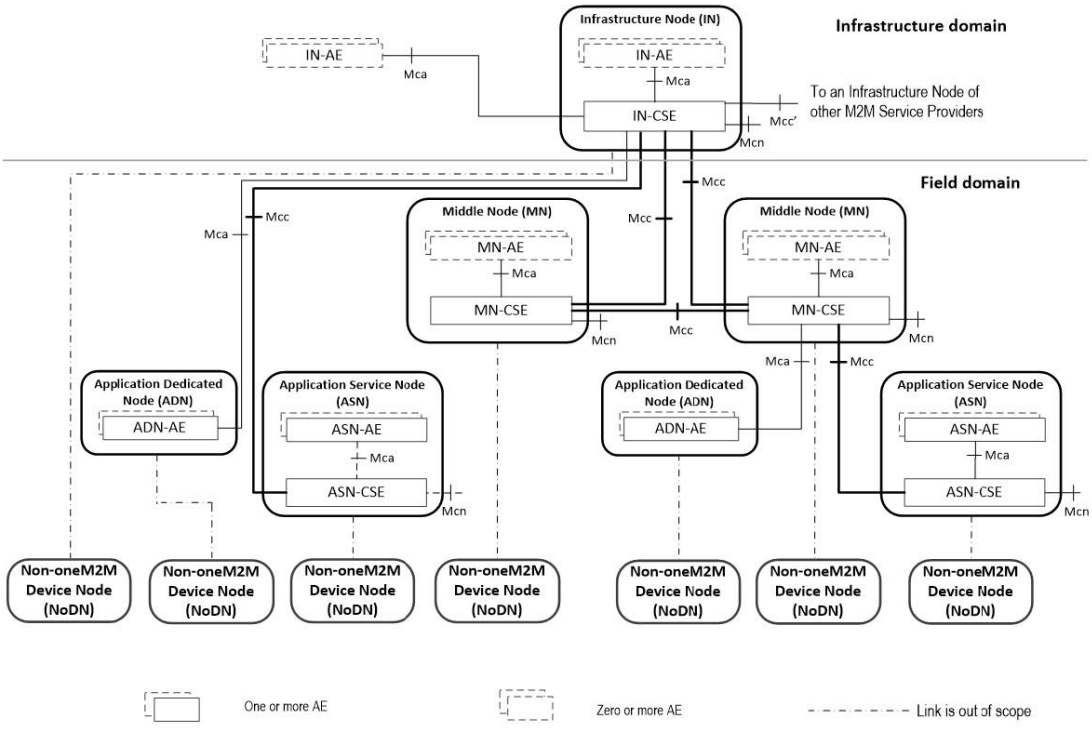
oneM2M is an M2M services platform built upon devices, gateways, and servers. It allows end-to-end communication between data source and applications. oneM2M is network centric. It allows interoperability between devices and applications through the use of uniform interfaces and APIs. oneM2M reaches to achieve interoperability through different standardization efforts.

5.3.2.2 Architecture

ETSI TS 118 101 [i.13] and ETSI TS 118 102 [i.14] define the oneM2M architecture and support the deployment of IoT infrastructures, using service platforms that provide multi-domain support and interoperability with a middleware offering e.g. identification and naming of devices and applications.

As shown in Figure 5, oneM2M architecture is composed of four functional entities called nodes, Application Dedicated Node (ADN), Application Service Node (ASN), Middle Node (MN) and Infrastructure Node (IN). Each node offers a Common Services Entity (CSE), an Application Entity (AE), or both. An AE provides application logic such as remote blood sugar monitoring.

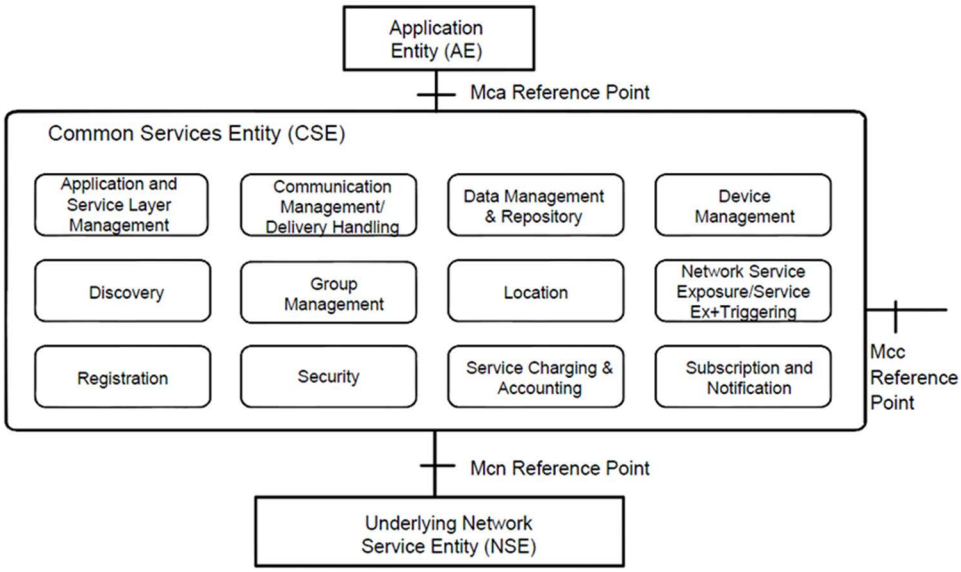
oneM2M specified 4 reference points called Mca, Mcc, Mcn and Mcc'. Mca enables AEs to use the services provided by the CSE. Mcc enables communications between CSEs belonging to the same service provider. Mcn enables communication between a CSE and the service entities in the underlying networks. The Mcc' interface enables communications between CSE belonging to different service providers.



Source: oneM2M TS-0001 [i.63].

Figure 5: oneM2M high level architecture

As shown in Figure 6, each CSE exposes a set of Common Services Functions (CSFs) that can be used by applications and other CSEs such as Application Enablement, Device interworking, Remote device management, Subscription and notification, Security, Group broadcasting, Device triggering, location, etc.



Source: oneM2M TS-0001 [i.63].

Figure 6: oneM2M functional architecture

5.3.2.3 Interoperability and other aspects

oneM2M standard enables heterogeneous devices and applications to understand exchanged data in a similar way, implying a precise and unambiguous meaning of the exchanged information. Several approaches dealing with the structuring of the data exchange and the codification of the data are supported so the interacting entities can exchange and interpret the data:

1) Pure ontology-based solution (RDF/OWL serialization format): oneM2M base ontology extended with a domain-specific ontology e.g. SAREF. For more details, see oneM2M TS-0012 [i.15].

2) Common vocabulary (basic serialization format XML or JSON): Smart Device Template (SDT) for the home domain. For more details, see oneM2M TS-0023 [i.16].

3) Resources specializations: oneM2M FlexContainer resources specialized with a technology-specific data model. For more detail, see ETSI TS 118 121 [i.17].

4) Blackbox resources: Basic oneM2M resources (Container, ContentInstance and Group) extended with an external domain-specific data model. The contentInstances resources are considered as black boxes and could contain any domain-specific data model. For more details, see oneM2M TS-0014 [i.18] and oneM2M TS-0024 [i.19].

A work item called "oneM2M WI-0056 Evolution of Proximal IoT Interworking" has been defined to provide an harmonization of the work done for interworking between oneM2M and specific proximal IoT technologies, such as AllJoyn, OMA LWM2M, and OCF (formerly known as OIC). The idea is enabling interworking with external "proximal" IoT technologies without the need for a oneM2M application to be aware of the details of device specific technology. For more details, see oneM2M TS-0033 [i.20].

oneM2M has been designed to simplify computable logic, inferencing, knowledge discovery, and data federation between a myriad of applications and devices.

5.3.3 The OCF Platform

5.3.3.1 The Ecosystem

IoTivity is an Linux Foundation collaborative Open Source project that implements the specifications of the Open Connectivity Foundation (OCF) in an IoT platform. The OCF succeeded to the Open Interconnect Consortium (OIC), which was launched in 2014 and which was in competition with the AllSeen alliance project. Since the merge of the OIC and AllSeen alliance in 2016, the two initially different Open Source projects IoTivity and Alljoyn are collaborating to implement the future versions of the OCF specification in a unique platform.

5.3.3.2 The Interoperability

IoTivity is aimed to provide the interoperability support for different OSs, including Linux, Arduino, Android, Tizen, and uniform APIs for different programming languages including C, C++ and Java as well as the interoperability support for multiple connectivity types (e.g. Wi-Fi, Ethernet, Bluetooth low energy, Thread, Z-Wave) and extensibility mechanisms to support proprietary protocols. A Resource Model is specified allowing interoperability to be defined independently of the transport protocols. Besides the general mechanisms of the CRUDN (Create, Retrieve, Update, Delete, Notify) model, the Resource Model introduces a set of primary concepts, namely: Entity, Resources, Uniform Resource Identifiers (URI), Resource Types, Properties, Representations, Interfaces, Collections and Links. OCF has also created "oneiota", a Web-based tool (https://www.oneiota.org/) for IoT data model sharing between IoT application developers.

5.3.3.3 The Architecture

The architecture adopts the RESTful model where physical world entities (bulb, human body, house appliances, etc.) and IoT objects (end-user devices, sensors, actuators, etc.) are considered as resources addressed by a Unique Resource Identifier (URI) following the client/server (CoAP for Home applications, or DDS for industrial IoT) or publish/subscribe (MQTT) models, OCP APIs and non-OCF protocols bindings. The architecture building blocks are organized in horizontal layers: Application Profiles (including, Data Models), Framework (i.e. Core Services) and Communication, with security as a transverse layer as depicted in Figure 7:

- The Application Profiles provide market segment specific data models and functionalities.
- The Framework provides the core functionality.
- The communication layer, actually composed of three sublayers:
  - Transport that provides end-to-end flow transport with specific QoS constraints.
  - Networking that provides functionality for data exchange between devices.
  - L2 connectivity that provides functionality to establish physical and data link layer connections (e.g. Bluetooth) to the network.

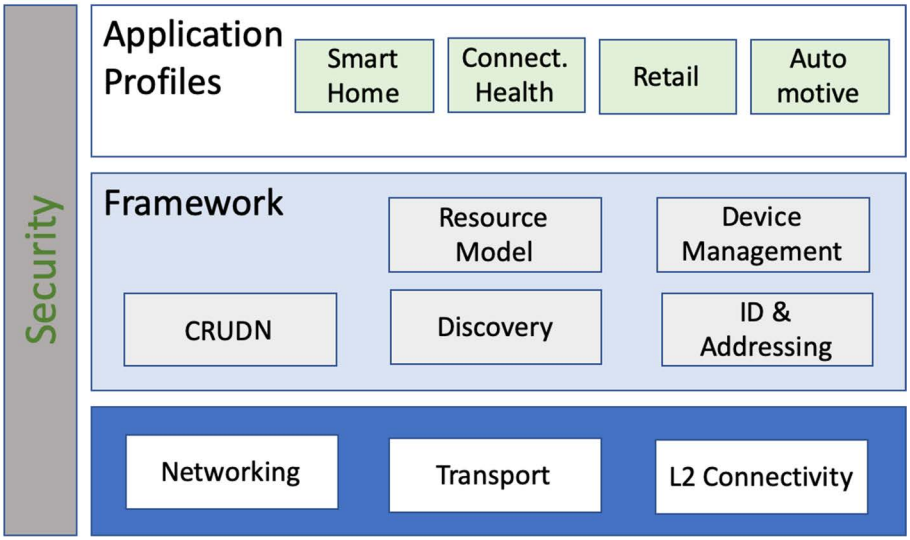


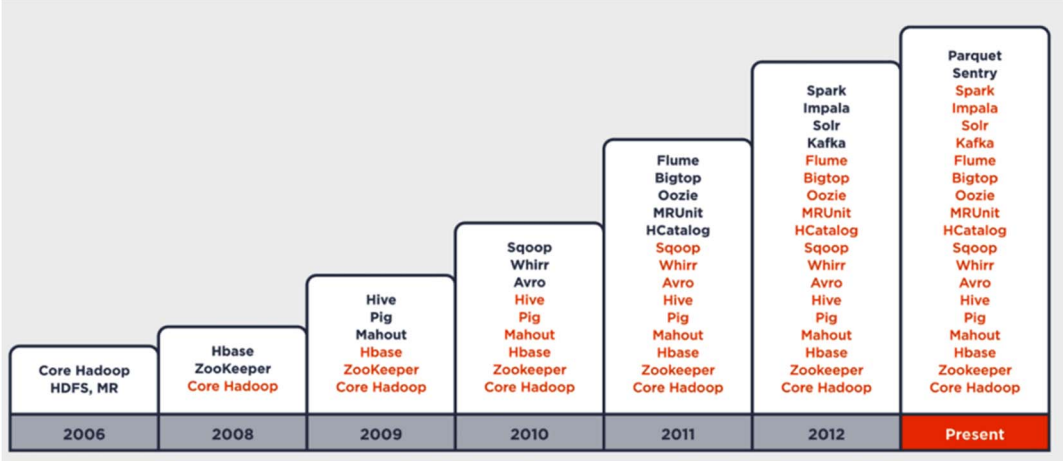
Figure 7: Building Blocks of OCF architecture

5.3.4 The Apache Platform

5.3.4.1 The Ecosystem

The Open Source Software (OSS) communities are involved in the development of a large number of projects that aim at developing, evolving and maintaining components that apply to a large number of contexts. In a number of cases, the OSS components are developed in competition with the development of other OSS communities. However, some communities become more successful, in particular because they adopt governance structures and ways of working that attract a large number of highly skilled developers and provide long-term commitment that are key for the success of their components.

Such OSS ecosystems are bearing large similarities in the way they operate with the Standards Developing Organizations (SDOs) or Standards Setting Organizations (SSOs): open participation, transparent feature selection process, strict peer review, public Application Programming Interfaces (APIs), etc. When they become large and stable enough, they can be viewed not just as developers of OSS components, but as providers of an "OSS platform" with a catalogue of components that provide solutions across a large of the functional domains. The example of the Apache ecosystem and its maturation over time (as illustrated in Figure 8) is typical. From an initial component (that turned out to be critical in the development of data analytics), the ecosystem has expanded to a larger set that can reasonably be termed as an OSS platform.



Source: [i.59].

Figure 8: The example of the Apache Hadoop ecosystem

5.3.4.2 Some elements of the platform

Some examples of components of the Apache platform are presented in the Table 2 below. All the components listed have a Technology Readiness Level (TRL) at TRL-9. It should be noted that several components can be used in the same functional domain, depending on the needs (functionality, interoperability).

Table 2: Examples of Apache Software Components

Component	Scope	Functionality
ZooKeeper	Orchestration	ZooKeeper (https://zookeeper.apache.org/) is an Open Source server which enables highly reliable distributed coordination. It is essentially a distributed hierarchical key-value store, which is used to provide a distributed configuration service, synchronization service, and naming registry for large distributed systems.
Mesos	Orchestration	Mesos (http://mesos.apache.org/) abstracts CPU, memory, storage, and other compute resources away from machines (physical or virtual), enabling fault-tolerant and elastic distributed systems to easily be built and run effectively. The Mesos kernel runs on every machine and provides applications (e.g. Hadoop, Spark, Kafka, Elasticsearch) with APIs for resource management and scheduling across entire datacentre and cloud environments.
Yarn	Orchestration	Yarn (https://yarnpkg.com/lang/en/) splits up the functionalities of resource management and job scheduling/monitoring into separate daemons: a global Resource Manager (RM) and per-application Application Master (AM).
Kafka	Communication	Kafka (https://kafka.apache.org/) is a distributed streaming platform that follows a publish/subscribe architecture to manage data streams. It is used for two broad classes of application: Building real-time streaming data pipelines that reliably get data between systems or applications; and Building real-time streaming applications that transform or react to the streams of data.
Flume	Communication	Flume (https://flume.apache.org/) is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving substantial amounts of log data. It uses a simple extensible data model that allows for online analytic application.
Redis	Communication	Redis (https://redis.io/) is an in-memory data structure store, used as a database, cache and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs and geospatial indexes with radius queries. Redis has built-in replication and provides high availability via Redis Sentinel and automatic partitioning with Redis Cluster.
Flink	Computation	Flink (https://flink.apache.org/) is a stream processing framework for distributed, high-performing, always-available, and accurate data streaming applications.
Spark	Computation	Spark (https://spark.apache.org/) is a fast, in-memory data processing engine with elegant and expressive development APIs to allow data workers to efficiently execute streaming, machine learning or SQL workloads that require fast iterative access to datasets.
Storm	Computation	Storm (http://storm.apache.org/) is a distributed real-time computation system. Storm makes it easy to reliably process unbounded streams of data, doing for real-time processing what Hadoop does for batch processing.
Hadoop	Computation	Hadoop (https://hadoop.apache.org/) is a software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs.
Cassandra	Storage	Cassandra (http://cassandra.apache.org/) is a database that manages fast massive amounts of data. It is scalable and highly available with linear scalability and proven fault-tolerance on commodity hardware or cloud infrastructure. Cassandra supports replicating across multiple datacentres.
Hive	Storage	The Apache Hive data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. Structure can be projected onto data already in storage. A command line tool and JDBC driver are provided to connect users to Hive.

Component	Scope	Functionality
HBase	Storage	Apache HBase (https://hbase.apache.org/) is the Hadoop database, a distributed, scalable, big data store. The goal of the project is the hosting of very large atop clusters of commodity hardware. Apache HBase is an Open Source, distributed, versioned, non-relational database. Apache HBase provides Google™ Bigtable-like capabilities on top of Hadoop and HDFS.
Solr	Search Engine	Apache Solr (http://lucene.apache.org/solr/) is an Open Source Enterprise search platform, written in Java, from the Apache Lucene project. Providing distributed search and index replication, Solr is designed for scalability and fault tolerance. It has REST-like HTTP/XML and JSON APIs that make it usable from most popular programming languages.
Lucene	Search Engine	Apache Lucene (https://lucene.apache.org/core/) is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform.

More on the use of OSS components in the context of IoT Virtualization can be found in three Technical Reports:

- ETSI TR 103 527 [i.21];
- ETSI TR 103 528 [i.22];
- ETSI TR 103 529 [i.23].

5.3.5 Point solutions and the challenge of integration

5.3.5.1 Fitting point solutions in global platforms

Some of the platforms presented in clauses 5.3.2, 5.3.3 and 5.3.4 are meant to cover a very large span of functionalities and provide an integrated approach in destination of the designers and developers of IoT systems.

However, a growing number of open (i.e. non-proprietary) partial solutions, that do not by themselves constitute a full-fledged platform, but offer "packaged" solutions that can be further integrated into an existing (and potentially already selected and deployed) platform, start to appear. Many of them are supported by Open Source Software (OSS) communities and provide proven (sometimes TRL-9) solutions to emerging pressing functional requirements.

These new, often very innovative, solutions come with different usage models with two major ones: the stand-alone model (still the most frequent) and the service-based model (more and more frequent with the development of virtualization and the provision of new cloud-based services). In both cases, the integration of these innovative solutions will come with a cost related to the integration of the solution (discussed in clause 5.3.5.3).

5.3.5.2 Stand-alone or cloud-based solutions: two examples

Two examples of point solutions originating from Open Source communities are provided below. The examples are chosen to reflect how an innovative open point solution can be made available to cover new requirements. An important aspect to be considered with the use of such solutions is that they can provided with different deployment models, i.e. as a stand-alone element that have to be directly integrated in the standardized platform or as a cloud-based service (the latter being a more and more frequent case, given the progress in the usage of cloud within IoT systems.

The Elastic stack

The Elastic Stack is an Open Source integrated solution that is integrating the components coming from three Open Source projects: Elasticsearch, Logstash, and Kibana:

- Elasticsearch is a search and analytics engine.
- Logstash is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms it, and then sends it to a "stash" like Elasticsearch.
- Kibana lets users visualize data with charts and graphs in Elasticsearch.

The Elastic stack can be provided as a stand-alone software as well as a hosted (cloud-based) service.

Kubernetes as a service

Kubernetes (https://kubernetes.io/) is an Open Source system for automating deployment, scaling, and management of containerized applications. It is a recognized and effective way to package microservices-based applications that use containers. It groups the containers that make up an application into logical units for easy management and discovery. It orchestrates computing, networking, and storage infrastructure on behalf of user workloads. This enables portability across infrastructure providers.

Kubernetes as a Service (KaaS) is a way to package Kubernetes that offer simple APIs to manage the underlying mechanisms (the so-called pods) in a transparent manner, with the joint benefits of Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).

Kubernetes can be provided as a stand-alone software as well as a hosted (cloud-based) service which is provided by a (rapidly growing) number of cloud service providers.

5.3.5.3 The role of integration

The constant innovation in IoT is concerning different aspects of the IoT systems. Any global needs to integrate very diverse new IoT components such as protocols, APIs, SW frameworks, etc. The very high-level differentiation of services in IoT is transforming each IoT service development into an integration exercise with an associated effort.

The question is how much of this effort is dedicated to the service development and the combination of the service components, versus the development and integration of the platform components and of the communication chain.

The integration of a specific new sensor or a new actuator with a device in order to build a new physical object can be taken as an example. This integration requires not only the creation of the new physical object but also the development of the control software and of the applications in order to use it in combination with other components.

The challenge for integration is that the effort should not be dedicated (as it is very often the case today) to integration with protocols, APIs, clouds, servers (including the related data structures) with the risk of having to redo it later in case the same solution needs to be used for a different customer with a different mix of technical choices (in particular regarding the semantics of the data structures).

The related defocusing of resources from the IoT product/service development innovation effort towards increased efforts on integration is largely due to the fragmentation of the solutions and the ecosystems. The challenge of the adoption of standardized platforms is to demonstrate the effectiveness of its approach and reduce the appeal of specific solutions which are often (closed or open) proprietary and create a high dependency from the customers and a lock-in situation.

6 Dealing with Interoperability

6.1 Strategic Approaches to Interoperability

Interoperability has been addressed since the early days of Information and Communication Technology (ICT), first, with basic interoperability standards (in particular protocols) and then maturing to tackle more complex issues such as the development of Information Models in support of information exchange between independent systems. The role of standardization in interoperability solutions is key, both by allowing a formalized support to various implementations, and by ensuring a consolidation of the options available so that the interoperability technical landscape is not a jungle of competing solutions.

Interoperability technical frameworks, platforms as well as standards are systematically refined and expanded in the IoT community. The definition of a framework for IoT standards is a constantly evolving target with new challenges and new solutions emerging constantly.

These standardized IoT interoperability frameworks need to be challenged by in-depth consideration of usage scenarios. In particular, their application and adoption in complex business sectors such as the Industry sector, which involve interoperability with a wide range of standardized and non-standardized "Things" (e.g. PLCs and other types of controllers and automation assets) will be carefully analysed.

Four interoperability levels are identified by IERC AC4 [i.33] and adopted by standardization initiatives (ETSI, EIF) and research activities (e.g. see [i.30]): Technical, Syntactical, Semantic and Organizational interoperability:

- The **Technical Interoperability** concerns heterogeneous software and hardware (e.g. communication protocol heterogeneity).
- The **Syntactical Interoperability** concerns data formats (e.g. JSON or XML). Syntactical Interoperability is also an issue for combining and reusing ontologies or semantic datasets developed with different software dealing with different syntaxes (e.g. RDF/XML, N3).
- The **Semantic interoperability** addresses the meaning of content and concerns the human rather than machine interpretation of the content [i.19]. One can consider that Semantic Interoperability can be extended to cover machine interpretation. It may concern in general:

  1) ontology heterogeneity (e.g. ontology designed by different persons differ in the structure);
  2) terms used to describe (e.g. "t", "temp" and "temperature" are different terms to describe temperature); and
  3) the meaning of data exchanged according to the context (e.g. body temperature differs from room temperature).

  This is important to later interpret IoT data and build smarter and interoperable semantic-based IoT applications. IERC AC4 [i.33] underlines the need to be agreed on common vocabularies to describe data.
- The **Organizational Interoperability** concerns the heterogeneity of the digital infrastructures of different service providers. The European Interoperability Framework (EIF) gives specific guidance on how to set up interoperable digital public services. EIF provides a multilayer model that distinguishes between technical, semantic and organizational interoperability.

Being able to exchange messages between IoT platforms with correct content is a step toward end-to-end interoperability and exchanging messages which content is understood from end-to-end is of higher usefulness for interoperability. For this purpose, the communication and protocol-level standardization initiatives are to be strengthened by full Semantic Interoperability for IoT platforms. The semantic Web of Things constitutes one of the common founding directions towards this objective of end-to-end Semantic Interoperability in the different initiatives that brings the semantic web principles into the Internet of Things technology.

The solutions for IoT platforms can be implemented as a continuity and an extension of the Web of Things providing rich high-level description of the IoT platforms components as Web resources and allowing dynamic discovery and requesting of the IoT remote resources. Different architectures can however be implemented depending on the criticality and the QoS properties of the associated domain. The requirements related to privacy and reliability of data collection and access services are constituting a central focus for IoT and e-health applications while performance and real-time properties are of the utmost importance for the IIoT and the related industrial applications. The IoT sensors can be connected individually as Web resources or interconnected and associated to a Web-enabled database. The collected data are then semantically enriched by metadata annotation and exposed for interoperable exchange between different IoT platforms and applications.

Both the IoT platforms and the Semantic Web principles aim to enable Machine-to-Machine data production and consumption. Hence the founding principles of the Semantic Web, that is to say linked data principles, formal vocabularies and deductions mechanisms are good candidates for end-to-end Semantic Interoperability implementation in IoT platforms. Ontologies allow to describe the objects of the IoT platforms as well as the environment in which they are deployed and the data information they produce as sensors or consume as actuators.

6.2 Technical Approaches to Interoperability

6.2.1 A program for evolution

One of the main objectives of the standardization activities for the last decade was to enable the transition from the Internet of Things (IoT) to the so-called "Web of Things" (WoT). The ultimate objective for the next decade is to enable the transition from the Web of Things (WoT) to the so-called "Semantic Web of Things" (SWoT). The key reference concepts underlying these evolutions are developed below.

6.2.2 The Internet of Things (IoT): The basic objectives of IoT platforms

The IoT is defined by ITU-T as "a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) Things based on existing and evolving interoperable information and communication technologies" [i.28]. The same ITU report defines a Thing as "an object of the physical world (physical things) or the information world (virtual Things), which is capable of being identified and integrated into communication networks". Objects can be physical electronic devices such as sensors, actuators, smartphones, but also software services as well as biological living mobile or immobile (animals, plants) or inert (land parcels) entities, or "artificial" mobile or immobile entities (cars, buildings) of the real world on which (or from which) remote actors act (or collect data).

6.2.3 The WoT: a step towards interoperability of IoT platforms

In order to bridge the technical interoperability gap between IoT platforms and their users at the service level, the use of Web principles and technologies has been proposed by the different initiatives and projects for designing and standardizing IoT platforms. The REST architectures are used for interconnecting the platforms clients and the HTTP transport protocol is adopted for carrying the data requests and responses.

The Web of Things (WoT) was introduced in the International Conference on Semantic Computing in 2008 [i.26]. It was adopted later by standardization organisms (W3C mainly) for identifying the principles for Web-based interconnection and control of physical and virtual objects. The WoT is defined by ITU-T [i.27] as "*A way to realize the IoT where (physical and virtual) things are connected and controlled through the World Wide Web*". On top of the heterogeneous IoT communication networks, the WoT provides a unified access to both data and things identified with International Resource Identifier (IRI) through Web protocols. Various communication technologies are deployed for the interconnection of IoT networks, sometimes on top of dedicated hardware and vendor-specific. Application developer cannot tackle this heterogeneity. The Web-based interoperability technologies have been proposed by standardization organisms for tackling this heterogeneity. In particular, the WoT is standardized by the W3C working group [i.39], following the definition of the ITU-T. Objects of the WoT are identified with an IRI, but the target device might be unable to communicate over HTTP. In this case, a "proxy" is necessary to map HTTP to an ad-hoc IoT protocol allowing the object to be addressed as a Web resource. Applications communicate with Web servers, but these servers are usually not directly connected to IoT devices: dedicated gateways are deployed to ensure the communication.

The reference IoT platforms architectures consider a multi-tier architecture where applications connect to remote cloud servers that collect data from (or forward action requests to) remote sensors (actuators) that may be connected via gateways or directly when they are IP-enabled and when they implement the standardized API.

6.2.4 The SWoT: The foundations for semantic interoperability of IoT platforms

The extension of the WoT architecture, aiming to allow the end-to-end interoperability, constitutes the ultimate objective of interoperability and machine-level communication and understanding of exchanged data. It is called "semantic interoperability". Relying on Web technologies to expose devices and IoT data to applications brings interoperability at the technical level: applications can access representations of the devices over protocols they understand, such as HTTP. However, the notion of interoperability is richer than just being able to communicate over the same protocol. It can be declined at multiple levels, each one extending the previous as defined by IERC AC4 [i.33].

6.3 Interoperability Frameworks

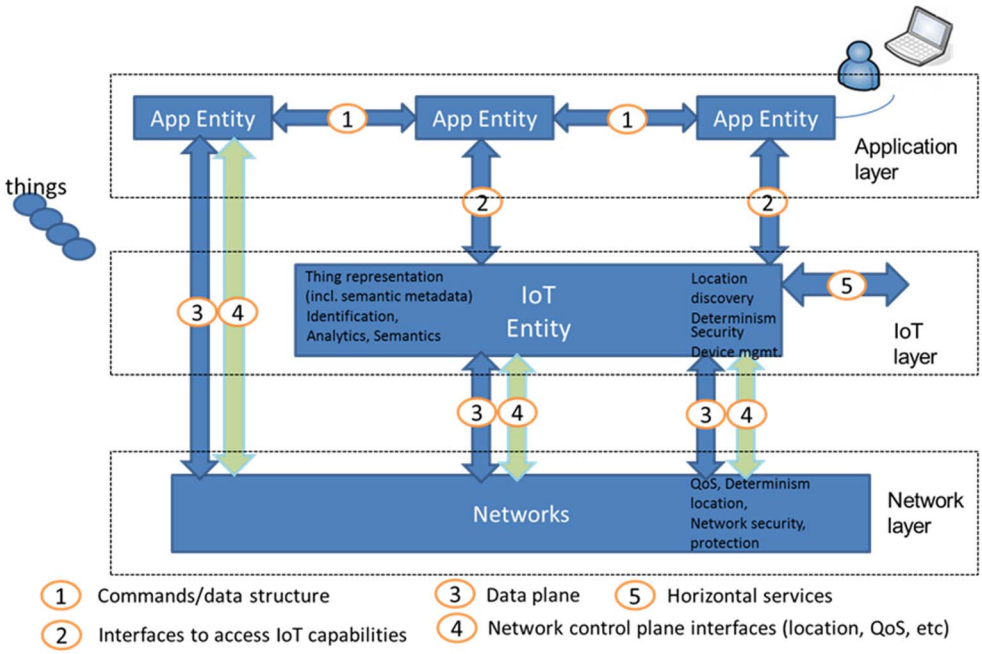
6.3.1 The AIOTI Reference Framework

One major effort is done within the AIOTI - in particular in AIOTI Work Group 03 on IoT Standardization - in order to address all the new challenges and provide a consistent framework over time. The AIOTI Working Group 03 has now published several releases of its "IoT LSP Standards Framework Concepts" report [i.7]. Though initially targeted as a set of concepts for the IoT Large-Scale Pilots (LSPs) that have started at the beginning of 2017, these framework concepts are applicable to the IoT community at large (e.g. in the Industrial IoT industry segment).

The "framework concepts" are used for providing the main elements of the AIOTI WG03 shared standardization recommendations: the "IoT Mappings" and the High-Level Architecture (HLA) [i.8].

Three layered categories of entities are distinguished according to the AIOTI reference architecture (see Figure 9):

- The Application layer contains entities that implement IoT application logic. An App Entity can reside in devices, gateways or servers.
- The IoT layer contains entities that expose IoT functions to App Entities via the access interface or to other IoT entities via the horizontal service interface. Typical examples of IoT functions include data storage, data sharing, subscription and notification, firmware upgrade of a device, access right management, location, analytics, semantic discovery, etc. An IoT Entity makes use of the underlying Networks' data plane interfaces to send or receive data via the data plane interface. Additionally, the network control plane interface could be used to access control plane network services such as location or device triggering.
- The Network layer: may be realized via different network technologies (PAN, LAN, WAN, etc.) and consist of different interconnected administrative network domains with best effort data forwarding or a premium service with QoS guarantees including deterministic guarantees.



Source: AIOTI [i.8].

Figure 9: AIOTI HLA Functional Model

6.3.2 Other Interoperability Frameworks

The European Interoperability Framework (EIF) defines an interoperability framework as a set of standards and guidelines that describes the way in which organizations have agreed, or should agree, to interact with each other.

In the particular context of IoT, different interoperability frameworks have been defined and have been recently unified by the oneM2M Partnership Project that defines an interoperability framework at the service level with different interworking proxies. The oneM2M system interoperability framework is composed of four functional entities:

- the Application Dedicated Node (ADN);
- the Application Service Node (ASN);
- the Middle Node (MN); and
- the Infrastructure Node (IN).

Each node contains a Common Services Entity (CSE), an Application Entity (AE), or both. An AE provides application logic, such as remote power monitoring, for end-to-end M2M solutions. A CSE comprises a set of service functions called Common Services Functions (CSFs) that can be used by applications and other CSEs. CSFs include registration, security, application, service, data and device management, etc.

The ITU interoperability framework [i.27] built on top of a WoT conceptual model considers mash-up services connected to message brokers that implements also the equivalent of the interworking proxies of the oneM2M framework.

Almost all standardization initiatives have not efficiently tackled the issue of end-to-end interoperability, i.e. considering both communication and data interoperability. Thanks to communication interoperability, M2M system entities already benefit from services such as discovery, monitoring, management, etc. Although such an interoperability framework can be sufficient for the design and implementation of specific M2M systems, autonomic management using automated reasoning based on a knowledge-oriented service platform cannot be achieved.

For example, using a service platform built upon the oneM2M framework, an application can seamlessly discover new devices plugged into the system. This application can subscribe to the new device events and will receive them as soon as they are triggered, even if the routing path implies crossing multiple entities and using heterogeneous communication protocols or network technologies at any segment of the communication path. This has been made possible thanks to the interoperability at the communication level. Now that device events have been successfully reported, the application does not have any means to understand the reports' content without prior conventions (data formats, encapsulation, and semantics) set up between the application and the device application developers.

6.3.3 Interoperability examples of use-cases

Ontologies have been proven to be beneficial for intelligent information integration, information retrieval, and knowledge management. They enable the indexing of resources' content using semantic annotations that can result in the representation of explicit knowledge that cannot be assessed and managed because of their mess. Ontologies are very popular and useful to overcome challenges of end-to-end semantic interoperability for IoT platforms because they provide an efficient way of cleverly structuring a domain, making use of semantic hierarchical and property/value relationships based on a vocabulary of concepts/instances [i.26].

To overcome this gap, different ontologies have been defined by standardization initiatives including the oneM2M base ontology and the ETSI SAREF ontology [i.32], and by research initiatives such as IoT-O [i.49] (available at https://www.irit.fr/recherches/MELODI/ontologies/IoT-O.html).

An ontology for IoT represents a variety of concepts such as platform, deployment system, thing, device, node, service, sensor, actuator, sensing and actuating capabilities, observation, operation, time, unit, kind, and their relationships. And it allows users and applications to discover, monitor, and control sensors and actuators offering particular services and having particular properties with a high degree of automation.

In an Industrial IoT (IIoT) Use Case, [i.34] proposes an asset-tracking system where an application ontology is used to represent, in near-real time, the location of assets in a smart factory. Observations are combined with background knowledge in order to explicitly represent the position of an asset, producing high-value information from lower level observations.

6.4 The challenge of IoT Deployment

6.4.1 Key technologies and design requirements

The potential benefits of combining the strengths of IoT and Cloud Computing industries in a new value proposition are now clearly visible to the IoT industry. On the one hand, virtualization is expected to provide technical benefits such as more flexibility on assigning IoT virtualized objects and functions to physical resources. Moreover, virtualization should bring as well financial benefits (e.g. greater CAPEX efficiency) or operational benefits (e.g. improvement of automation and operating procedures) altogether resulting in boosted service innovation.

Requirements for IoT ontology modularization are commonly driven by use-cases in which only parts of an existing ontology are needed, or in which constrained devices are unable to perform inference and reasoning on a full ontology [i.35].

Modularization requires the partitioning of ontologies into independent sub-modules [i.37]. Sub-modules are self-contained knowledge components that:

- Are loosely coupled.
- Define their own set of core concepts and relations.

- Are reusable.
- Are linked to other module(s) with explicit relationship(s).

Good examples of modular ontologies are the Smart BANs (Body Area Networks) and MyOntoSens ontologies [i.24]. Within MyOntoSens, a Wireless Sensor Network (WSN) module is formed of clusters (Cluster module; BAN module for Smart BANs) that are composed of nodes (Node Module). A node is used for process (Process Module) and takes measurements (Measurements Module). The 'Measurements Module' is sufficiently light to be instantiated and stored within sensors, while the Process and Measurements modules full instantiation and inference/reasoning can actually only be performed within a more powerful edge node. The full BAN ontology needs to be instantiated, inferred and processed on powerful remote cloud servers.

6.4.2 Interoperability in Smart Cities

The support of ICT technologies in general, and IoT more particularly, for Smart Cities is of most importance for Europe knowing that almost three quarters of the European population lived in a urban area in 2015 [i.45]. The interoperability is identified as a key challenge in different standardization initiatives, research studies, and European projects on Smart Cities, and as a requirement to avoid Smart City fragmentation and vendor lock-in (e.g. [i.42] and [i.43] projects) and to provide open and horizontal interoperable platforms across Smart Cities facilities and Smart City sectors. Different kinds of platforms and technologies are to be tackled including IoT platforms, Big Data platforms, citizen-centric services, as well as management-oriented tools for fleet management, decision making or crisis management.

The SynchroniCity project [i.43] adopted the BSI model [i.44] for defining the Upper level ontology for Smart Cities. AIOTI identified several requirements for LSP projects to implement interoperability [i.41]. The list includes: the deployment of communication infrastructure (e.g. capillary networks), the use of Open Source software and hardware as well as data interoperability. Open and Agile Smart Cities (OASC) principles as defined by AIOTI include the definition of open interfaces and APIs [i.41]. The European Innovation Partnership on Smart Cities and Communities (EIP-SCC) identifies the different initiatives related to smart cities and in particular those related to interoperability. The EIP-SCC report [i.40] highlights the multi-city, multi-culture, multi-partner and scale-free properties for interoperable protocols in smart cities addressed by the "City Protocol" initiative. Several other initiatives and projects addressed the smart cities interoperability requirements. The list includes: ESPRESSO, BIG-IOT, OrganiCity, Triangulum, and symbIoTe.

6.5 Criteria for Interoperability

Semantic interoperability is achieved when interacting systems attribute the same meaning to an exchanged piece of data, ensuring consistency of the data across systems regardless of individual data format [i.35]. This consistency of meaning can be derived from pre-existing standards or agreements on the format and meaning of data or it can be derived, in a dynamic way, using shared vocabularies either in a schema form and/or in an ontology-driven approach. These two complementary approaches are considered as the main categories that are driving the interoperability design criteria in standardization, industrial and academic initiatives. They can be referred to as *"data-model based semantic interoperability"* and *"ontology based semantic interoperability"*, respectively. In this context, it can be pointed out that for high performance and real-time requirements in application domain of the category of IIoT, the *"data-model based semantic interoperability"* can constitute the most appropriate interoperability solution. A generic data model specific to each vertical domain can be designed and adopted. This is for example the approach proposed by the OCF for the IoTivity platform and oneM2M vertical extensions. Other Use-case scenarios without real-time constraints can require more elaborated run-time interoperability solutions with advanced reasoning capabilities allowing vocabulary alignment and rule-based inference. The Medical IoT platforms and the associated e-Health vertical scenarios can require and take benefit from such solutions. Reference ontologies can be found under the category "LOV4IoT-Health" of the Linked Open Vocabularies for Internet of Things (LOV4IoT) [i.31] provided by the LOV initiative (http://lov4iot.appspot.com/?p=ontologies). Smart Home IoT platforms and related services constitute also an appropriate application domain for the deployment of elaborated *"ontology based semantic interoperability"* solutions. Standardization initiatives of oneM2M and ETSI SAREF are providing advanced solutions for triggering such directions.

To achieve communication interoperability, it is required to standardize a service layer with open interfaces which is already done in many standards such as ETSI TS 102 690 [i.38] and ETSI TS 118 101 [i.13]. However, the system architecture design that will bring life to such horizontal solution presents, in itself, many challenges [i.37]. The system architecture should be flexible enough to be deployed in different kind of machines. Given that the platform cannot support from scratch all existing technologies and protocols, the architecture should be modular, highly extensible and support service dynamic discovery. In addition, achieving communication interoperability is not sufficient to solve the semantic gap between objects required for inferencing, knowledge discovery, and data federation. In absence of interoperability, each IoT application or device should personalize the payload by using its own vocabulary, so all interacting applications should agree beforehand on a specific terminology before establishing any communication. This implies necessarily a strong coupling between applications, which is in contradiction with the horizontality criterion that constitutes a commonly-agreed requirement for IoT frameworks. An interoperable platform makes it easy to interconnect heterogeneous devices and applications in one system and makes it possible to connect various IoT systems together.

Semantic Interoperability may require additional functions for content transformation operations such as enrichment, abstraction, aggregation and presentation. It may also require additional functions of awareness and heterogeneity management for IoT entities, which includes: selection, abstraction, composition, discovery, configuration, and exposition. Depending on the associated non-functional requirements (such as privacy and performances) these functions may require powerful cloud resources to be executed or may, alternatively, be performed on the edge devices, or, even, locally on the sensors that produce and consume the data.

Figure 10 summarizes the different interoperability criteria by a two-dimensions classification considering the horizontal interoperability (with three levels H1, H2, H3 each of them being an extension of its predecessor) and the vertical interoperability (with two sub-categories, each composed of two levels: V1, V2 and V3, V4). The trade-off between the different levels by putting the focus on a given requirement gives rise to a choice with horizontal and vertical interoperability criteria. Moreover, it can be considered for example that the ultimate "organizational interoperability" identified in the EIF classification can be reached by considering solutions implementing both levels V4 and H3. Examples of candidate platforms could be fulfilled by platforms that implement the different oneM2M specifications including interoperability with OCF-IoTivity and SAREF ontology for smart home applications.

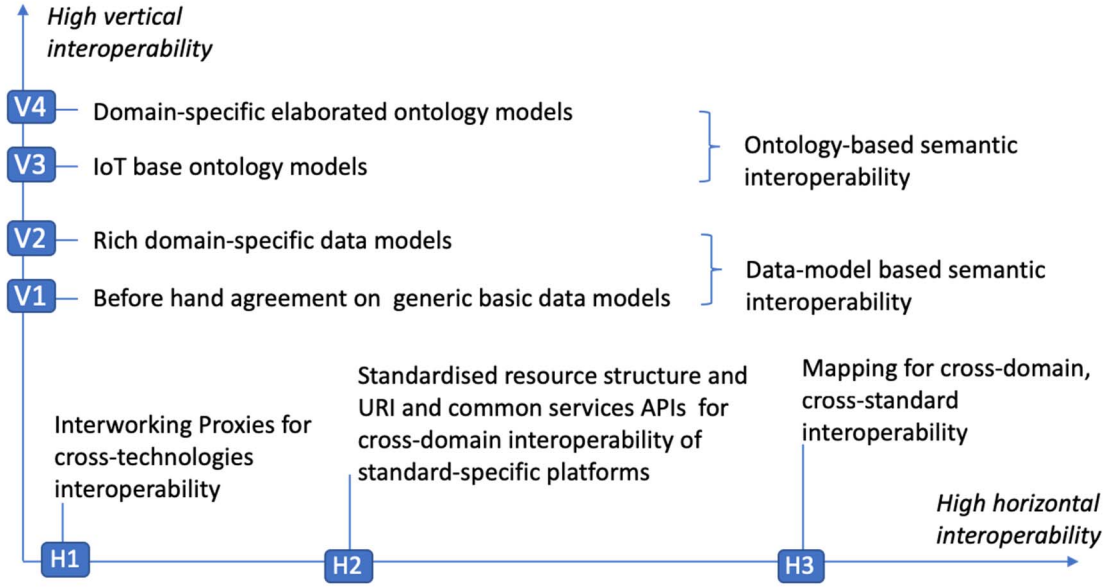


Figure 10: Synthetic view of interoperability dimensions

For example, the H2020 AUTOPILOT project presented in Figure 4 supports the following levels of horizontal and vertical interoperability:

- H1: The system architecture of AUTOPILOT is composed of several interworking proxies for interworking with heterogeneous technologies from the ITS domain such as CAN, 6LowPan, etc.
- H2: AUTOPILOT platform is based on oneM2M standard which offers a unified resource structure and common service APIs paving the way to interoperability with specific platform such as Fiware context broker, Watson IoT and Ocean.

- V1: In AUTOPILOT devices and applications could exchange data in a seamless way however a beforehand agreement still required to understand the exchanged data.
- V2: AUTOPILOT offers a rich data model inspired from DATEX and SENSORIS that serves as a common vocabulary between the interacting entities.

AUTOPILOT does not support the H3 level since the current architecture and data model remains specific to ITS domain and is not extended and validated with other domains. In addition, AUTOPILOT does not support V3 and V4 interoperability levels since the data model does not rely on ontologies for horizontal interoperability.

7 The case of Industrial IoT

7.1 The challenges of Industrial IoT

7.1.1 The role of Industrial IoT in Smart Manufacturing

7.1.1.1 Smart Manufacturing

Smart Manufacturing is a central concept in the current digital transformation of the industry. It is referring to the systematic creation and usage of data and information in Enterprises throughout the whole production life-cycle. The expected outcome is more agile and flexible manufacturing processes that enable the optimization of resources usage, that allow a quick response to the change in demand and that limit the negative environmental impact.

It is deemed so important that initiatives referring to it are promoted in many countries. A prominent one (and a reference for similar national initiatives) is Industrie 4.0, a project launched in 2011 and publicly backed and steered by the German government, that considers it as vital for the future of the country with the intent of ensuring technological leadership. Similar initiatives in other parts of the word go by different name: China 2025 (China), Industrie du Futur (France), Industria 4.0 (Italy, Brazil, Mexico, etc.), Industria Connectada 4.0 (Spain), Manufacturing USA (United States). India, Japan, Korea, Spain, Sweden, or the UK have country-specific efforts as well.

The main challenge of Smart Manufacturing is to enable a massive adoption and integration of new technologies such as IoT or Cloud Computing in order to provide much more flexibility, adaptability and security. This evolution will require achieving a transition from the current manufacturing paradigm (the so-called "Manufacturing Pyramid") towards the era of Cyber-Physical Production Systems (CPPS).

The current model based on the current "Manufacturing Pyramid" approach is dominant and has proven to be very effective in relatively closed environments. It is a layered model (as described in IEC 62264 [i.46]) where the different layers of the pyramid are quite strictly hierarchically separated and the communication between the bottom layer of IoT devices and the upper layer of the production system at-large are complex and the supporting data models often very specialized.

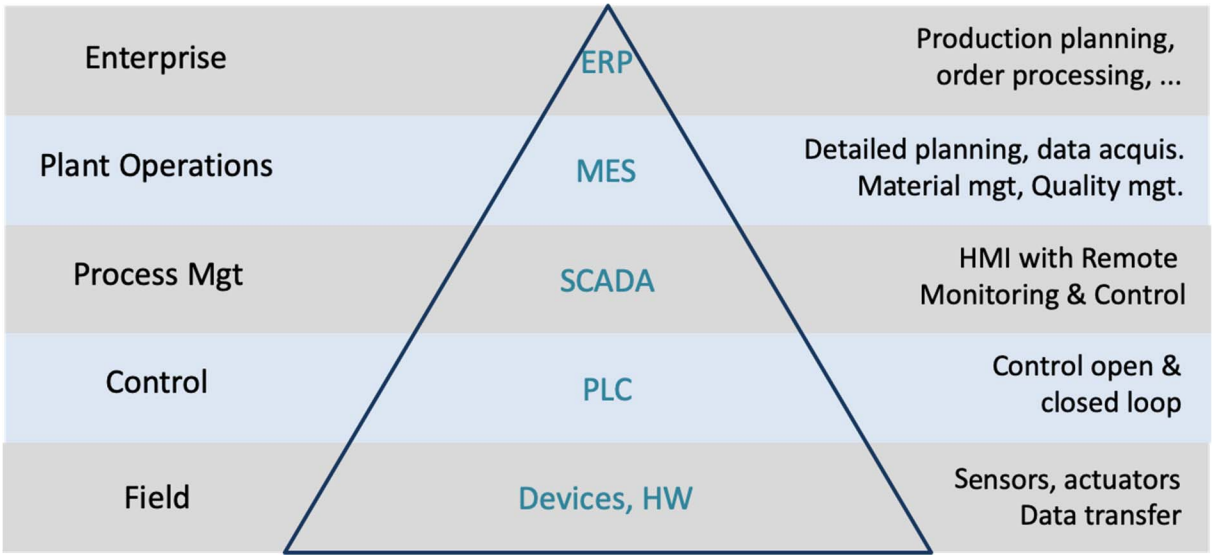
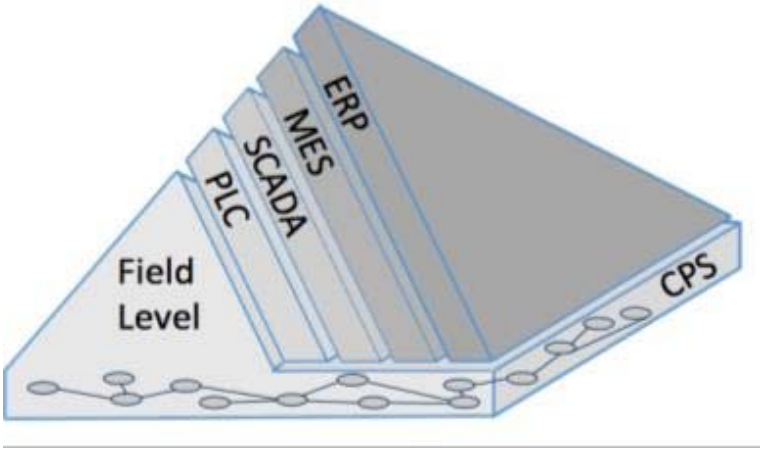


Figure 11: Manufacturing Pyramid

The main challenge posed to this model is that the management of data is complex with data models difficult to adapt rapidly and limiting interoperability. This lack of flexibility makes the collection, analysis and decision-making based on the massive amount of data produced by IoT components a daunting task.

With the "Cyber-Physical Production System" (CPPS) approach, it is expected that the field level (e.g. the factory, the robots, the sensors) will be connected with a wider range of applications and services - making use of the vast quantities of data available to plan, monitor, re-tool and maintain, optimize supply chains, etc. - together with being ensured a higher level of trust and security from a redefined security architecture.



Source: [i.60].

Figure 12: Cyber-Physical Production Systems

The expected benefits are relating to greater operational efficiency as well as the possibility to deploy a large set of new application and services.

7.1.1.2 Industrial IoT

A key enabling technology of Smart Manufacturing is the Industrial Internet of Things (IIoT). IIoT is a subcategory of the IoT, in which the Things that are connected are industrial devices: sensors, actuators, automated machines and equipment, robots, etc.

IoT is a core element of Smart Manufacturing enabling important new capabilities in factories. In particular:

- It provides the communications backbone that allows data to flow within the factory.

- It can help providing business processes (e.g. supply chain) getting real-time information on products, materials and equipment, and thereby improve their efficiency.
- It can provide fine-grain information on energy consumption, and improve the overall energy efficiency.

7.1.2 IIoT: a major segment of the IoT with significant specificities

7.1.2.1 A major business segment

The market of IoT is clearly booming and will continue according to analysts. According to Bain Research (see [i.50]), it is expected to grow from 195 B€ in 2015 to more 412 B€ in 2020. More importantly, the consumer segment will only represent 30 % of the global market with 70 % for the Business-to-Business (B2B) segment. In the B2B segment, Industrial IoT is expected to be the largest one with 75 B€ (around 25 %).

Consequently, considering the huge amount of investment to be made to reach this expected level of revenue, the question of the choice of an IIoT platform is important. The investment in platforms themselves is still hard to measure but at least the largest two categories for investment will be Data Service and Analytics (around 20 %) and System Integration (around 20 %).

Consequently, IIoT is a tough challenge to the IoT platforms and solutions developed so far, at least because the development of IIoT will require:

- The massive and effective integration of data analytics in the field.
- The optimization of the integration effort due to the introduction of Things, devices and networks.
- The optimization of the integration with legacy embedded systems.

7.1.2.2 Differences with traditional Operational Technology (OT)

A few differences may be noted between IIoT and the traditional OT technologies, based on Programmable Logic Controllers (PLCs), Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), etc.:

- More effective data collection capability, from the point of view of costs, speed and scalability.
- Ability to federate heterogenous data sources, including IT data bases, thus helping to reduce silos fragmentation.
- Ability to communicate across factory and Enterprise boundaries.
- Offering single point of access for analytics to all federated data.
- Better, more flexible and suitable for self-consumption tools for data visualization are expected by users.

7.1.2.3 Differences with consumer IoT

A significant part of the accumulated experience on the use of IoT platforms comes from their usage in the consumer segment, though a growing feedback comes from industrial sectors as the adoption of IoT is taking-up. Some important differences with respect to consumer IoT have to be taken into account in the choice of a platform. This includes:

- Lower number of end nodes.
- Higher frequency of data acquisition.
- Higher volume of data managed.
- Need to ensure contextual consistence among data, both spatially and temporally.

7.1.3 Expected Benefits of IIoT

There are many expected benefits from IoT in general. Table 3 is focusing on those that are mostly expected for Industrial IoT.

Table 3: Expected benefits of Industrial IoT

Type	Description	IIoT specificities
Data Analytics	Data analytics solutions are widely available for collecting, manipulating, transforming and analysing data. Many solutions are available in the general IT domain and become largely available for the IoT.	A common application of data analytics is the use of KPI for evaluating the operational efficiency of production plants, the efficiency in the use of energy and other factors that are relevant for company success.
Operations Optimization	Data collected from sensors and other sources is processed in order to determine optimal settings of production equipment, according to desired criteria (increase efficiency, quality etc.), reducing the dependence upon human intervention. According to various sources, this is where lays the greatest potential for value creation. Operation optimization appears to be particularly favoured by large organizations.	Though this is partly possible with today's technologies (e.g. closed-loop feedback controllers) what is new is what can be done using a mix of modern technologies: • Cheap sensors can be placed in many parts of plants and production lines. • The communication infrastructure is now available to collect large amounts of data, scaling to levels that were not attainable in the past. • Optimal settings can be determined using AI technologies, and this allows multivariate optimization without the need for an explicit mathematical model of the production facilities; setting adjustments can be applied continuously in order to compensate also for drifts and changes in the environment.
Predictive Maintenance	In a similar way to Operations Optimization, the combination of large amounts of data, both live and historical, together with AI processing can be used to anticipate possible failures. This has the potential to prevent breakdowns and may help reducing maintenance costs. Predictive maintenance usually involves the creation, with AI based techniques, of dedicated models that can sense impending equipment failures and call for appropriate action.	In order to collect the data needed to build the model, in first instance, and afterwards run a monitoring application, IIoT technology is usually needed, for several reasons, among which are: the data that need to be acquired are rarely part of the control strategy in place, so new, dedicated sensors (e.g. vibration or sound sensors) have to be installed. The amount of data and speed of collection involved often exceed the data processing capability of typical supervisory (SCADA) systems. Since a dedicated data collection infrastructure is put in place, it can be deployed without fear of interference with control strategies already in place. Cheaper sensors and communications networks can be used: this can be done because potential sensor failures do not immediately impact on plant control. The data collection system can sense sensor failures, so that they can be fixed quickly. Data collected this way may be unsuitable for transmission over the Internet these cases it is processed by edge devices. Filtered features and reduced amounts of data can be transmitted to central facilities, where inputs from different pieces of equipment, possibly distantly located from each other can be compared and processed further: this approach enables manufacturers of IIoT enabled equipment to get better insight.

Type	Description	IIoT specificities
Manufacturing Execution (Systems)	A seamless integration of shop floor with higher levels of the company is increasingly needed. In particular, the linking of ERP production planning with operations on the shop floor via MES functionalities is becoming a requirement, that the dominant companies in certain manufacturing supply chains impose to smaller suppliers downstream.	Implementing MES does not strictly imply the use of IIoT technologies. However, especially in SMEs, the two often go hand in hand. This happens because, once an IIoT infrastructure linking machines to edge devices and higher computational resources is put in place, it is often found that the same infrastructure can be exploited also to support the deployment of lightweight MES solutions.
Supply Chain Integration	With IoT, real-time information may become available so that products and supplies can be better tracked. IIoT technology already is beginning to make a difference in areas such as asset tracking or Fleet Management (see below).	The communication between companies along the supply chain involves better knowledge of the status of shipped goods, but also allows the exchange of production related information directly from the shop floor of suppliers. In fact, the pressure from larger companies controlling the chain is a driving force for small suppliers to put in place IIoT based MES solutions.
Asset Tracking	Shipped goods can be tracked for location and also for environmental conditions. The latter is especially relevant in order to determine whether sensitive goods (e.g. pharmaceuticals, food) have been properly handled throughout transportation.	With IIoT, it is possible to gather insights about the condition of products while they are still in transit, thanks to Internet-connected sensors and other IoT devices: this knowledge may help managers take timely decisions during transportation.
Fleet Management	It is now possible to install IIoT devices that allow monitoring in real time not only the location of transportation means (e.g. trucks) but also many other parameters (speed, fuel consumption etc.).	This information, together with information coming from other sources (e.g. traffic monitoring, weather prediction) forms the basis for intelligent management of fleets.
Incorporating IoT capabilities into products	There is a significant market opportunity for companies looking to build IIoT capabilities into their physical products. This can be considered a form of "external use" of IIoT (as opposed to "internal use", by companies that use IIoT for ameliorating their own internal production process). The design of such products is often done in partnership with companies that specialize in software development.	The most common kind of solution in this business approach involves the use of remote monitoring to provide end users with information on equipment performance. Other Use Cases are IIoT solutions for maintenance and service.
Servitization	More and more manufacturing companies tends (and try) to bundle their product-based solutions with integrated services into Product-Service Systems (PSS). The integration of early PSS offers with data analytics is creating new, and potentially disruptive, offering in existing value chains. Servitization is a business approach that can be enabled through incorporating IIoT capabilities into products, leveraging a number of the technical approaches outlined above: • Analytics • Optimization • Remote maintenance • Predictive maintenance Companies that follow this approach improve the strategic value of their relationship with customers via more closely tying the customers' success to the individual equipment performance.	This opens a path towards selling not "just" pieces of equipment, but directly the value that the equipment brings to the end customer (e.g. in terms of units of goods processed). This approach becomes possible by means of remote monitoring, that allows sharing information on throughput and asset utilization between equipment manufacturers and their customers. In many cases, maintenance of the equipment is not left to the customer but is retained and directly managed by the manufacturer. The business model based on the idea of servitization ties the customer's success to equipment performance, can strengthen the relationship between the two parties and may help shielding that relationship from competition.

7.1.4 Challenges and barriers to, and strategies for the adoption of IIoT

7.1.4.1 The current situation: A Progressive Adoption

Although steadily increasing, the adoption of IIoT in the industry is slow. A large part of the investment and efforts has been so far on proof-on-concepts and limited implementations. This can be attributed to several reasons, like:

- Industry is often slow in adopting new technologies and concepts.
- There is a strong need to leave existing control equipment in place, and integrate it into the IIoT infrastructure that is being deployed.
- There are still a number of technical challenges to be solved such as those outlined in clause 7.1.4.3.
- In the case of heavily regulated sectors (e.g. Pharmaceuticals), it is unlikely that companies consider ditching the kind of OP equipment that is traditionally used to control their core manufacturing process. They do implement IIoT applications, but those applications are either related to auxiliary equipment (e.g. utilities) or, when core equipment is involved, the do not directly influence the main process and are limited to ancillary, albeit useful, functionality (e.g. predictive maintenance).
- Non-technical, human related factors have to be considered as well, since the adoption of new technology paradigms also change the way people work.

7.1.4.2 On the importance of legacy: Greenfield vs Brownfield

When a new factory, or at least a new plant, is built from scratch (the "Greenfield" scenario), it is much easier to design and implement an IIoT ecosystem because it can be taken into account since the beginning of the design phase.

However, an IIoT project is much more commonly started to enhance, complement or change in some way existing plants and production lines (the "Brownfield" scenario). One of the drivers for the adoption of IoT is to make increase the operational efficiency of the existing brownfield assets of manufacturers. Contrary to greenfield, brownfield requires integration with legacy systems which becomes a major challenge when the existing assets have been deployed several decades before and cannot integrate modern technology, in particular software (see [i.47]).

When the basic design of existing equipment and control systems cannot be changed, this makes the implementation of the IIoT ecosystem much more difficult. The "Greenfield" scenario calls for the need to connect to a variety of different devices that are already in place, with different connectivity capabilities, which in turn poses interoperability challenges (that will be discussed in greater details in following clauses).

7.1.4.3 Technical barriers to adoption

Several technical roadblocks are still major factors in the slow adoption of IoT by industry, amongst which:

- Security. The lack of security of basic IoT devices (sensors, video cameras, etc.) is very often pointed out as a roadblock for IoT. In the case of IIoT, a very important vector for the progression of IoT in the Enterprise is the network which has now enormous capacity to transport the vast amount of data that IoT produces. However, the network brings new challenges in terms of security, in particular with respect to Authentication (and AAA in general), Identity Management, global use of encryption, etc.
- Data protection and integrity. Companies generate greater and greater quantities of sensitive data which may become liabilities in case of theft (by competitors or malevolent actors). On top of the security measures that have to be upscaled, companies face the need to define and implement data protection and integrity policies for which they often do not have a clear strategy and the internal skills to define and undertake.
- Complex standards landscape. There is a number of standards available in Smart Manufacturing. However, the standards landscape is also complex with potentially difficult choices. On the one hand, there is a form of fragmentation in certain aspects, for example with a multiplicity of protocols to deal with the vast range of devices brought by the growing variety of IoT devices. On the other hand, many IIoT systems depend on standards related to Information Models that may prove hard to evolve, for instance to take into account more dynamic models such as those provided by Semantic interoperability [i.3].

7.1.4.4 Strategic choices and their impact on platforms

In such an evolving environment, it is necessary to understand which strategy to apply. Such a strategy depends upon multiple factors, many of which are not technical, such as:

- The specific business case.
- The specific market and product.
- The capacity of investment in the short and medium term.
- The product timeframe and its evolution.
- The in-house ability to perform the integration versus using (and depending upon) a system integrator.
- The position of the company in the value chain.

These strategy choices have a direct impact of the choice of the IoT platform. In particular, different options are available that are discussed in Table 4. The scenarios identified (that may not be as clear-cut as they appear and be potentially chosen concurrently by the same actor) are presented is the order of growing relative independence towards the main actors in the sector where the company facing the platform choice is operating.

Table 4: IIoT Platform selection scenarios

Scenario	Description	Pros and Cons
Internal development	This is often the solution taken by incumbents that want to be able to integrate the latest technologies within their legacy solutions. The result is a proprietary platform that can become a semi-open platform by offering open components (e.g. APIs) that be used to enlarge its ecosystem.	• Mostly for (very) large companies • Supports incremental innovation • Allows for a coherent approach towards the customers • May become the "de facto" reference in a sector and create an ecosystem of developers and integrators
Integration with an ecosystem	When a significant (or de facto) platform provider wants to enlarge the breadth of its platform to new use cases (and even to new adjacent sectors), it may be interesting for a company in this new sector to enter in the incumbent ecosystem with the objective to contribute to the definition of the platform along the lines of its own strategy.	• A possible approach for SMEs • Possibility to leverage the strength of the platform provider to promote its solutions against its competitors (provided this strategy is decided and implemented quickly enough) • Difficult to maintain a differentiation in the longer term
Point solutions coupled with cloud service provider(s)	Some companies may have a basis of internal competence in some sector with a specialized skill set without have the resources (financial and/or human) to build a full-fledge platform. The approach taken is to plug the company point solution on the infrastructure (IaaS, PaaS and SaaS) of a cloud Service Provider (CSP).	• A possible approach for SMEs • Supports the use of Open Source Software components • Dependency towards the CSP and limited choice for evolution • Difficult to generate a differentiation in the longer term
Standardized approach	With this approach, the choice of a reference (technical) architecture is key with a definition of the layered model chosen, the choice of an information and interoperability strategy and of the reference points and supported APIs. Different parts of the platforms can be served by a combination of some of the above scenarios.	• A possible approach for SMEs • Supports the use of Open Source Software components • Limits (but does not suppress) the dependency towards the de facto platforms or CSP platforms chosen

7.2 Using Standardized Platforms in IIoT

7.2.1 Technical aspects

IIoT is bringing new challenges to IoT by creating requirements that cannot always be provided as such by current generic IoT solutions. Some aspects are concerning in particular:

- Connectivity. Apart from the case of Greenfield projects where the adaptation to the state-of-the-art technologies is possible, the Brownfield projects face a number of hurdles relative to the existence of old equipment with limited software capabilities.
- Interoperability. The current approach to interoperability embedded in the "Manufacturing Pyramid" does not allow in general to benefit from the latest approach to interoperability at higher levels, in particular Semantic Interoperability.
- Virtualization. The virtualization of IoT has the potential to allow for new deployments models (e.g. off-premises) and to open to a new range of innovative cloud-based services. Reaping the benefits will require overcoming barriers to adoption such as security and data protection.
- Data Management and Analysis. This area is the one where most benefits are expected. However, some technical challenges remain like, for instance, the possibility to define edge-based solutions that allow for the local treatment of the massive amount of data produced by the "Things" on the field.
- Business Process Integration. The current challenge regarding solutions for connecting the major business processes with the IoT devices and the network is that these solutions need move from point solutions to more generic ones that will require less integration effort. This aspect is closely linked to the issues of interoperability.
- Support to development. The variety of elements to be integrated in an IoT system is also very challenging for the development tool chain that will have to support (and simplify) the integration of various devices and protocols, software components (e.g. OSS ones), a large range of APIs, etc.

The remainder of clause 7.2 is analysing some of these aspects with, in particular, the objective to outline some elements that are specific to IIoT (and not applicable in general to IoT systems).

7.2.2 Connectivity

7.2.2.1 The importance of legacy

Whereas in "greenfield" projects, the requirements can be specified since the beginning of the design, the most frequent and difficult case to handle, as pointed out in clause 7.2.2.3, is when legacy (sometimes quite old) needs to be integrated.

7.2.2.2 Greenfield: starting from scratch

In principle the easiest case is that of a "greenfield" project, where the needed requirements can be specified since the beginning of the design:

- Modern physical data communication channels and protocols can be selected, with common interoperability in mind.
- Equipment that comes with its own embedded controller may be selected by taking into account also its ability to expose and exchange data according to desired characteristics (physical communication channels, protocols, speed, etc.).

- In the cases where PLCs are used, that need to be custom programmed, it is possible to ensure that the model configuration of the selected PLCs allow the exchange of the anticipated amount of data, and that their programming is done in ways that allows a proper data exchange. The programming aspect may appear strange at first glance, but it not uncommon that PLC programmers handle data exchange as just an afterthought, or use data exchange techniques that may work in simple cases (e.g. sampling of slow changing continuous measurements) but are not adequate in more complex cases (e.g. time stamping data samples, or ensure that data has been properly received before overwriting its value, etc.).

7.2.2.3 Brownfield: integrating (with) legacy

Brownfield projects present a different, potentially larger, set of challenges. These can include:

- Having to deal with legacy unfamiliar communications channels and protocols.
- Requiring the use of OS versions that may no longer be supported.

Generally, there is little room for modifications of control systems that are already in place. In this case, it is necessary to do a survey of existing systems in order to determine the best approach case by case.

Table 5 addresses the most common situations. One should remember that more than one of those situations may occur at the same time in a given plant.

Table 5: Scenarios for Control Systems modifications

Scenario	Description
No data exchange possible	In some cases, the controller of a given piece of equipment is not capable of exchanging data. This may be due to several reasons, the most obvious (but by no means the only one) being the lack of a communication channel. It is sometimes still possible to obtain information about the operations of the controlled equipment, although this involves connecting extra I/O devices to selected points in the wiring strips. A few examples are: • Most operating equipment have, by regulation, a sort of semaphore with lights that show the operating status: halted, running, alarm, etc. By wiring the lamps to a data collection device one can get a crude representation of machine operativity, useful to determine simple efficiency KPIs. • Connecting meters to the electrical power lines that feed a machine and taking some measures (e.g. incoming current) may give insights into the operating cycles of the machine. • Utility plants that run on gas (e.g. building heating facilities) usually have a meter, installed by the gas utility, that is not accessible to the user. Since it is very desirable to know the gas consumption in real time, many ingenious ways have been devised in order to obtain a reading without actually connecting to the meter itself. This happens frequently now, however in perspective some new regulation seems to be upcoming that will make a data channel dedicated to user reading mandatory.
Simple Legacy Controller	Simple controllers (e.g. stand-alone PID controllers) or legacy PLCs often provide a built-in communication interface. Most often, the communication interface is not based on Ethernet but uses some legacy serial standard (RS-232, RS-422, RS-485, etc.) or a fieldbus instead. The communication protocols involved are usually rather low-level and, especially in the case of serial communication lines, may be non-standard ones. Integrating these controllers into an IIoT system requires: • For the physical aspect, the ability to exchange data over non-Ethernet communication channels. • For the logical aspect, the implementation of software interfaces in order to handle the data exchange. This usually calls for the use of edge devices that are equipped with the required physical communication channel support and allow for the installation of the needed software support. These devices act also as gateways between the controllers and the higher levels of the IIoT system.

Scenario	Description
Reprogrammable controller	In some cases, the PLCs or other type of controller are equipped with communication interfaces, possibly even using a (legacy) standard protocol over Ethernet, but in order to make data exchange possible some programming have to be done. As an example of a minimal intervention, consider the case of Siemens S7 PLCs. To make data exchange possible, the absolute minimum intervention requires: • Creating a dedicated Data Block, a data area set apart for communications purposes. • Defining a structure within said Data Block, analogous to the structure of a record in some programming language; • Connecting each field within the structure to an actual internal variable, which is used by the sequential control program. In more complex cases, the above actions are not enough and require special programming on the IoT side as well. This happens because the legacy communication protocols available on these devices are very low level and are not message-oriented. PLC programmers tend to use them as transport protocols, where data layout and exchange sequences are designed by the PLC programmer on case-by-case basis, resulting in the creation of custom protocols. Special handling from the IoT side is therefore also required to cope with these custom protocols.
Controller with Special Interface	It may be impossible to exchange data with the controller itself, but the manufacturer of the controlled equipment may provide an additional, usually optional, specialized interface to the factory information system. This is often the case for high-valued assets, and usually involves costs both for the interface itself and manpower required for its installation and configuration by specialized personnel. However, these costs are usually small compared to the cost of the equipment/machine. This is one of the best scenarios because it minimizes risk while at the same time offering good chances for interoperability with a standards-based interface. For recent equipment, it is reasonable to expect an OPC UA interface or OPC DA for less recent ones. Other alternatives, however, may be encountered.

7.2.3 Interoperability and the role of Semantics

One important objective of Smart Manufacturing is to improve interoperability in the upper layers of the stack (e.g. information and business).

Without semantics, a single piece of data does not convey any relevant meaning to a person or a client software application. However, when that piece of data is paired with some semantic context, the data inherits significantly more meaning. The data can then be more completely interpreted by a client software application without human intervention.

This means in particular that approaches like Semantic Interoperability might be adequate, but it appears that there are several aspects of resistance to adoption of this kind of technology:

- The market of interoperability solutions for non-trivial cases is dominated by a few solutions (all proprietary ones) offering their own semantic support.
- In addition, a relevant trend is observed: members of vertical industry domains define interfacing standards that revolve around the definition of specific "ontologies" (or equivalent).

Examples of this trend are:

- EUROMAP for the plastic and rubber machinery manufacturers (see clause 7.3.2.4).
- MTConnect for Computer Numerical Control equipment. MTConnect is a protocol designed for the exchange of data between shop floor equipment and software applications used for monitoring and data analysis (available under royalty-free licensing terms).

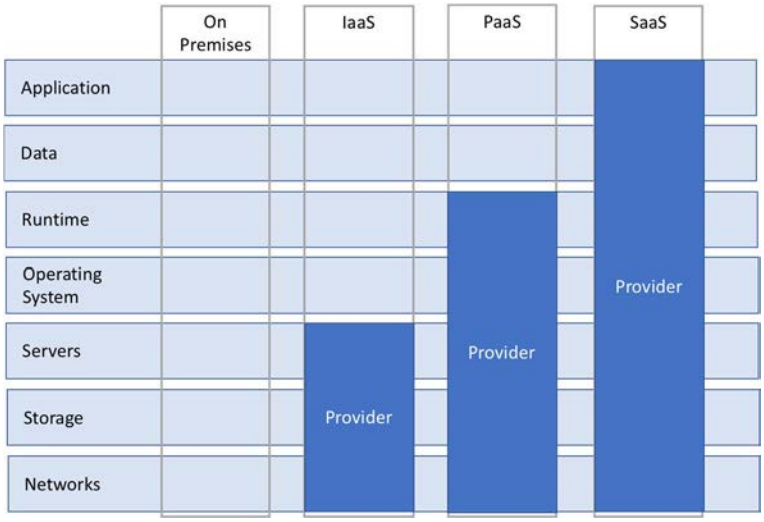
The Semantic Interoperability aspects are further discussed in the companion ETSI TR 103 535 [i.3]. In particular, the Technical Report includes a number of guidelines in support of the strategic decision regarding Semantic Interoperability as well as their concrete implementation in the context of industry.

7.2.4 IoT Virtualization and the role of Cloud

7.2.4.1 IoT Virtualization

The challenge of IoT Virtualization is to ensure that innovative services permitted by new cloud-based models (Platform-as-a-Service, Infrastructure-as-a-Service; Software-as-a-Service; etc.) are available to IIoT systems with the expected functional and non-functional support (e.g. low latency fault-tolerance, horizontal scalability, cost-optimization, or geo-optimization) together with Service Level Agreements (SLAs), and security.

The possibility to develop and deploy "cloud-native" applications above "cloud-native" infrastructures, goes together with the possibility to choose the most appropriate level of support from one (or more) Cloud Service Provider.



Source: ETSI TR 103 527 [i.21].

Figure 13: The potential of Cloud-Native Infrastructures

IoT Virtualization will provide technical benefits (e.g. flexibility on assigning IoT virtualized functions and objects to physical resource but also operational benefits (e.g. improved of automation and operating procedures) and financial benefits (e.g. CAPEX efficiency). As a result, it is expected to boost innovation, in particular "servitization" (as described in clause 7.1.3).

IoT Virtualization will rely on layered architectures that can structure the functionality such as the one provided in the ETSI TR 103 527 [i.21].

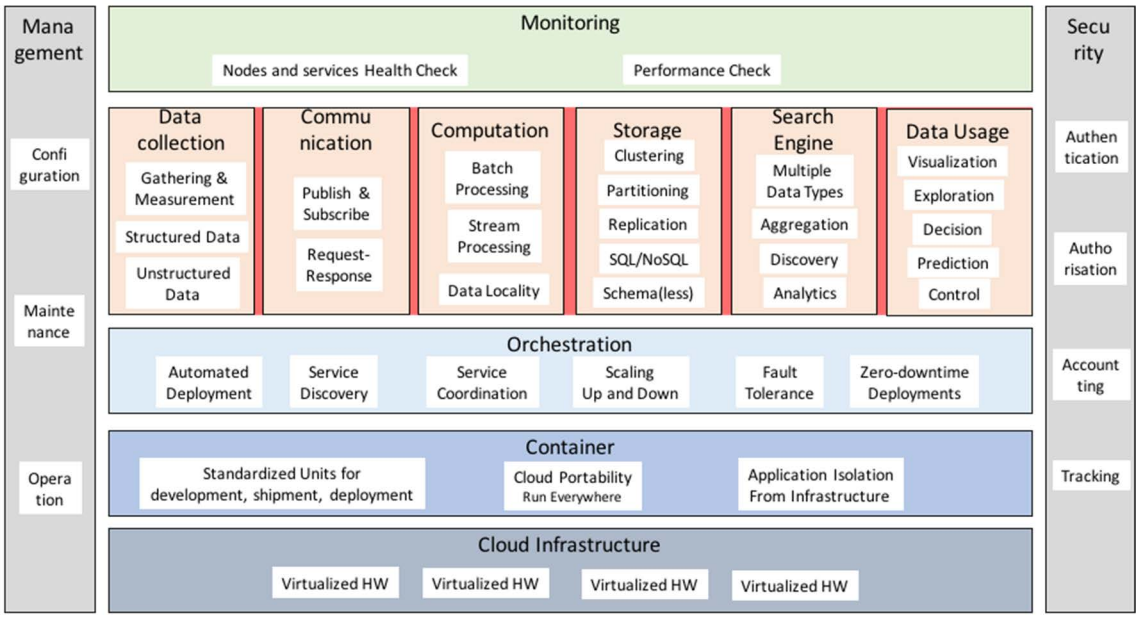


Figure 14: An HLA for IoT Virtualization

An important aspect of the above architecture (and similar ones) is that it offers several very rich functional layers with different benefits:

- The container layer provides support for the development, deployment upgrade and scaling of independent microservices. Such a structuration makes it possible, for instance, to benefit from cloud-based services such as KaaS (Kubernetes-as-a-Service) described in clause 5.3.5.2.
- The orchestration layer will support the deployment and concurrent usage of services.
- The "Common Services" layer (i.e. Data Collection, Communication, etc.) provides, in particular, the support for the very large number of (sometimes competing) components and solutions in support of data analytics.

More can be found on the ETSI TR 103 527 [i.21].

7.2.4.2 Virtualization in the context of IIoT

Recent research shows that nearly 60 % of IIoT technology is currently being deployed and hosted on-premises. This approach is often preferred when there is a strong concern for security and control over data in general. It also supposes that the company has the resources for procuring and maintaining the needed infrastructure.

Some typical cases for using a public (or hybrid) cloud infrastructure instead, are:

- When there is the need to integrate data pertaining to multiple sites/factories.
- When there is the need to use special computing capabilities, possibly of variable scale, and/or special algorithms that may not conveniently fit into available on-premises resources.
- When the user does not find it practical to own and manage the computing resources that are needed for an on-premises solution.

7.2.5 Data Management and Analysis

IoT data analytics refers to the usage of data analysis tools and procedures to extract value from the huge volumes of data generated by IoT devices. It is expected that IIoT will be most benefiting from the potential of IoT analytics: it becomes possible for Enterprises to collect and analyse data from all sorts of system components: sensors on shop floors, smart meters, weather stations, trucks, etc.

With the gradual uptake of IIoT, the focus of attention has been primarily on connectivity (with associated issues, see clause 7.2.2). The focus is gradually shifting to data analytics and the associated promise of creating new value from installing a global chain for the management and analysis of data. To some extent, a large part of the data that starts to be addressed on a large scale has existed (though in a reduced form) for a long time but was locked in incompatible and siloed plant floor systems. Only now that the data starts to be fully accessible, it can become part of a systematic effort for analytics with the objective to foster predictive maintenance, optimize energy efficiency of plant floor assets, or to respond to critical events (e.g. component failures, supply shortages) to minimize the production loss.

A very large of new data analytics offerings specifically aimed at IIoT and manufacturing use cases has flourished, be it as global offering for verticals (e.g. energy, oil and gas sector) as well as more specialized offerings applying analytics models and machine learning to specific problems (e.g. wind turbine efficiency). At the same time, very large industrial automation companies start to offer a core platform for the whole data management and analytics value chain. They develop an ecosystem of small start-ups that complement their basic offering with vertical or sector-based data analytics and positioning themselves as an open integration platform for IIoT analytics.

Given the very large business potential of IIoT (see clause 7.1.2.1), the Cloud Service Providers (CSP) are using their experience in IoT to address IIoT with the same kind of integrated approach. As an example, the new offering of Amazon Web Services ™ (AWS™), AWS IoT SiteWise™, is a managed service that collects, structures, and searches IoT data from industrial facility devices and uses it to analyse equipment and process performance data. This includes:

- AWS IoT Events™: a managed IoT service that detect and respond to changes indicated by IoT sensors and applications, such as malfunctioning, and automatically trigger actions or alerts.
- AWS IoT Things Graph™: a service which connects different devices and cloud services (e.g. humidity sensors with sprinklers and weather data services to create an agricultural application) with a visual drag-and-drop interface.
- AWS IoT Greengrass Connectors™: gives developers the ability to connect third-party applications (like ServiceNow for service management), on-premises software (like Splunk for log analytics), and AWS services like Amazon Kinesis for data ingest via common cloud Application Programming Interfaces.

NOTE: Mention of these AWS™ trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with these trademark.

A prerequisite to a proper and efficient use of data analytics in the Enterprise is the definition of a data analytics strategy, involving in particular the definition of a data management framework including a data architecture, i.e. a set of models, policies, rules and standards which defines which data is collected, how it is stored, arranged, and made available in a database system. Some layers have to be specified including Metadata Management, Data Quality, Data Governance, Data Integration, or Analytics & Data Privacy. The effective usage of data based on the data management framework requires as a first step to consolidate the Enterprise data and make it available to the OT experts as well as the IT experts.

7.2.6 Business Processes and Enterprise view

7.2.6.1 The need for Vertical Integration

The traditionally accepted view for vertical integration within manufacturing Enterprises is the partitioning into 5 levels, defined according to IEC 62264 [i.46].

Table 6: Functional Level of Activities

Level	Function	Software Perspective
4	Functions involved in the business-related activities needed to manage a manufacturing organization	Business logistics systems (e.g. ERP)
3	Functions involved in managing the work flows to produce the desired end-products	Manufacturing operations systems (e.g. MES, MOM)
2	Functions involved in monitoring and controlling of the physical process	Control systems (e.g. PLCs, DCSs)
1	Functions involved in sensing and manipulating the physical process	Intelligent devices (e.g. sensors, actuators)
0	Actual physical process	-

Relatively few manufacturing companies have a seamlessly integrated view of operations from shop floor up to the corporate level. It is common to find that the shop floor/factory level is isolated with respect to the other levels of the company, the missing link being level 3.

This happens both for SMEs and larger companies as well:

- SMEs have shied away from using conventional MES/MOM software systems because of the high cost and complexity of such solutions.
- Many large Enterprises also tend to consider factories as remote, not integrated, black boxes: this approach has been reinforced by years of globalization.

As a result, a very large part of the data that can be generated at the lower (1, 2) levels is not currently used to generate actionable insight. This gap needs to be bridged in order to exploit the full potential of smart factories.

7.2.6.2 The Impact of IIoT

The availability of IIoT technologies is gradually changing the above scenario. This comes in connection with:

- Lower costs of many of the components.
- Availability of cloud technologies.

SMEs are starting to use IIoT solutions to connect ERP scheduled activities to the shop floor, enabling the transfer of work programs and setup to machines according to the production plan: this is often done in non-conventional ways that take advantage of the IIoT infrastructure that is being put in place for this and other purposes as well.

A common requirement for smart factories is that all relevant business system should be connected to each other by way of some unifying service or software. Even within the relatively limited amount of data available from traditional control systems, companies that have tried connecting directly level 4 to level 2, i.e. ERP level with SCADA systems and supervisory control in general, in many cases have found that ERP-level software systems and databases are easily overwhelmed by production process data.

In order to accommodate the greater amount of information made available by IIoT-grade devices and systems at levels 1 and 2, a cloud-centric infrastructure approach is now (more and more) commonly used. With such an infrastructure, it is possible to build a platform that enables overall communications with the following advantages:

- It can accommodate information coming not only from sensors and control systems but also from other data sources, both internal (e.g. company databases) and external (e.g. the Internet).
- It can accommodate, at its edge, proxy software adaptors for interoperability with legacy systems.
- It allows a point of view which takes into account the context from which data originates, taking into consideration also surrounding information. As an example, consider an alarm message coming from a piece of equipment: knowing also the status of machinery upstream and downstream according to process flow can give better insight on what is actually happening.

This holistic point of view is not in contrast with, but rather complements, the modern trend that tends to incorporate a data server within each individual relevant machine or piece of equipment.

To clarify this concept, consider the following figure, excerpted from OPC UA official documentation, which is intended to show the capability of OPC UA servers to serve queries from multiple clients.

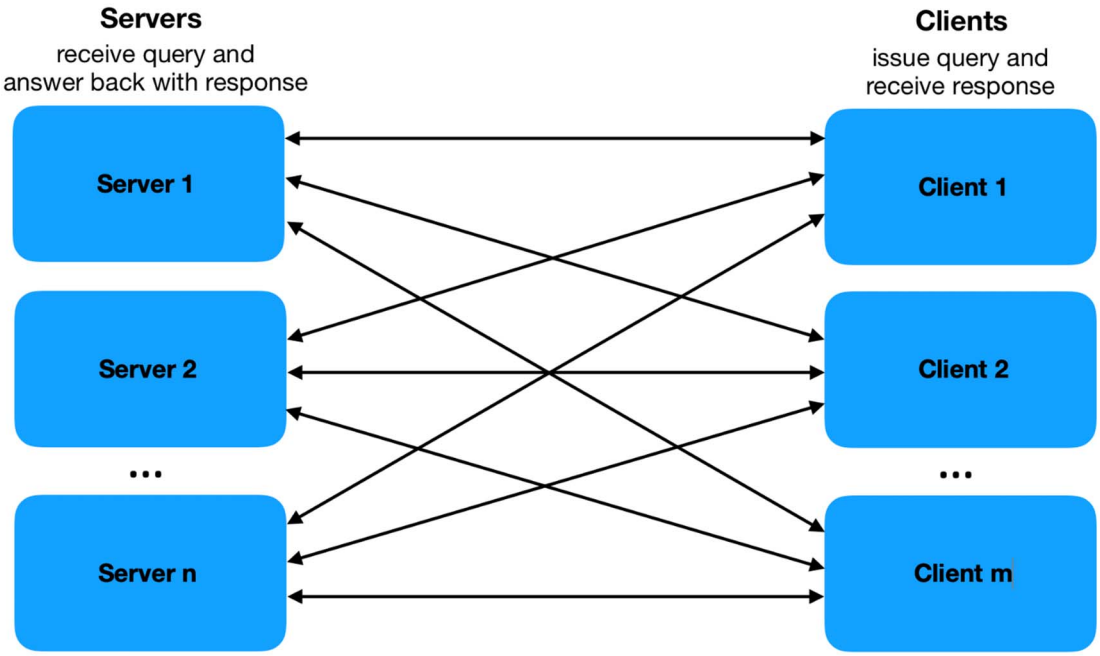


Figure 15: OPC-UA multiple queries support

It should be noted that it is only in very simple cases that business applications can advantageously connect directly to individual machines. The need to "know" the context, and therefore the plant/factory layout and interconnections is real, and it seems reasonable that this kind of knowledge should not be embedded into each one of the business applications but would be better placed into a common platform that acts a common server for all data analysis and processing needs.

7.2.7 Software Development

The development of IIoT systems will deal with a variety of devices, gateways, services using standardized protocols, information models, APIs, etc. Integration will play a key role and the dimension of software development tools and toolkits should not be overlooked.

The criteria developed in clause 5.2.2 on IoT platforms will also apply to the IIoT platforms and the Software Development Kits (SDK) that will support the work of designers, developers, testers or integrators: scope and breadth, openness, support of standards, ecosystem and, ultimately, maturity.

The requirements for the IIoT SDKs are those of IoT plus a number of specific ones:

- Scalability. The SDKs designed for use in the Industrial Internet of Things (IIoT) and Industrie 4.0, which means to allow users to interconnect industrial software systems independently from the hardware platform or operating system, and to operate in small embedded environments as well as in large server-based applications.
- Easy integration with the supply chain and other business processes. This means the ability to quickly adapt to a vast range of APIs and development models (UML but also REST, etc.), to create not only code but also business rules, scripts, etc.
- Management of the legacy code. The requirement of easy integration also applies to existing and legacy code.
- Built-in support for security-by-design. The dimension of security is critical in IIoT systems and many of the solutions available in IT also apply to IoT and IIoT (see the companion ETSI TR 103 533 [i.1]).
- Built-in support for safety. Design and tooling support of the standards that apply to industrial systems operating in safety-critical environments (e.g. IEC 16508) also need to be supported.
- Efficiency. The SKDs should support and leverage the most modern software development techniques and methodologies (e.g. microservices, agile methods, DevOps). In particular the SDKs should support the easy integration of OSS components.

Examples of what will be expected from these IIoT SDKs are:

- Fast development and integration of adaptors for a large range of protocols.
- Support for the creation of IIoT edge communication and data management gateways.
- Connection to the servers of Cloud Service Providers for storage, processing, analysis, and decision-making on the data produced by the connected devices.

The IIoT SDKs are not, in the vast majority of cases, built by the companies that develop the platforms. They are developed by specialized companies proposing a global integrated offer (in the case of proprietary SDKs) or by Open Source projects delivering parts of SDKs to be integrated in the development tool chain.

7.3 Platform adoption: proprietary or open/standardized

7.3.1 Proprietary platforms

7.3.1.1 Benefits and limits of proprietary platforms

Commercial IoT platforms tend to provide users with what is sometimes called "full experience", i.e. they offer services that go from connectivity up to data visualization, analytics, processing and rule-based actions (and more). Plus, dedicated APIs are provided that enable access from third party applications and systems.

This is done so that users should be able to use the platform from the start for all the most common tasks. All the provided functionalities are well integrated and optimized to work together.

The trade-off is that the various parts that form the system are tightly coupled with its internals, so that attempting to use them with other platforms may prove infeasible or, at the very best, extremely impractical.

Some issues are examined in the following clause.

7.3.1.2 Issues in coupling proprietary platforms and open/standardized platforms

Data Ingestion and Communication

This is where most platforms appear, at first glance, more standards-oriented in that they support ingestion through standard protocols (most used are MQTT and AMQP).

However, the payload formats are not identical to each other: this has the effect that a device sending data needs to be aware of the type of platform it is connected to, in order to properly format its messages.

Companies that develop solutions that can run on more than one platform tend to add a server-side application that will handle the messages after they have been dispatched and normalize them before storing the relevant information into the platform's database.

It is noted that OPC UA interoperability is typically not native to these platforms. Usually some sort of edge functionality is needed, that acts as a proxy and translates between OPC UA and whatever is the preferred ingestion mechanism for the platform at hand.

Embedded Functionalities

Much of the appeal of the various platforms lies in the availability of integrated applications, which allow users to quickly bring available data to fruition. Those application are, by design, tightly coupled with the internals of the platform itself and usually cannot be used directly with other data sources.

Companies that need to do so typically end up having to copy data from the external source to the internal database, where integrated applications can find and process them. Apart from being wasteful of resources, and possibly a source of additional costs, duplication of data can hardly be considered a good practice.

APIs

Each platform exposes APIs to enable access from external actors. The exposed APIs are peculiar to each platform, so external applications should be designed specifically for accessing one platform specifically.

Companies that develop applications that are intended to be used interchangeably with more than one platform typically resort to developing an intermediate layer that adapts to the specifics of each platforms and presents unified interface of its own to applications. The "portable" applications are then designed to use only this third, custom interface for data access: this approach makes the application even less portable, because now an additional dependency has been added.

The above considerations also apply to Open Source platforms, like ThingsBoard etc.

In principle, since source code is available, it is conceivable to take one of the embedded applications (e.g. a data visualization one) and adapt it so that in can be used on another platform.

However, since its design is intimately tied to the internals of the platform, the amount of work needed would be staggering.

7.3.2 A review of IIoT Platforms

7.3.2.1 Introduction

A general classification of IoT platforms has been done in clause 5.2.2.7. In the remainder of this clause, all the categories identified will be evaluated in the context of IIoT with the objective of identifying if IIoT platforms exist in each category and if they are global or point solutions.

7.3.2.2 Standardized Platforms

There are a lot of standards relevant for IIoT, but the number of platforms based on them is relatively limited, with one global solution developed by SDOs and several partial solutions developed by SSOs. More specifically:

- Platforms from SDOs. There is only one relevant example of standardized IoT platform, the oneM2M standard developed by the oneM2M Partnership Project. Despite its reference to M2M, oneM2M is a full IoT solution, offering a very complete service platform layer (supporting cloud-based deployments), a number of Reference Points, integrated advanced security data sharing capabilities, and strong integration support (e.g. protocol adaptors) that allow the coexistence of new and legacy technologies to be deployed on field. It is the only full IOT platform currently available in this category with no others currently in sight.

- Platforms from SSOs. Currently there are a lot of standards emanating from SSOs that are relevant for IIoT components in terms of connectivity protocols, API and data models (e.g. OMA light weight M2M). They are providing specifications supporting interoperability that can be used by solution and application developers. However, it is currently not possible to identify specifications in support of global platform in this category.

7.3.2.3 Open Source Platforms

The role of Open Source Software in IIoT platforms will be limited to the provision of components to be integrated in larger solutions. Examples of such integrated solutions are those related to Stand-alone or cloud-based solutions such as Kubernetes as a Service (described in clause 5.3.5.2), or data analytics components (described in clause 7.2.5).

It is currently not possible to identify specifications in support of global platform in this category.

7.3.2.4 Industry Groups Platforms

Examples of Industry Groups specifications

Some global Industry Groups specifications are available in support of IIoT. Examples of these are:

- The specification such as MTConnect, a protocol designed for the exchange of data between shop floor equipment and software applications used for monitoring and data analysis.

- The interoperability standards from the Semiconductor Equipment and Materials International (SEMI).

- The ETSI Industry Specification Group for cross-cutting Context Information Management (ISG CIM) has released its main specification ETSI GS CIM 009 [i.36] for NGSI-LD API, particularly targeting Smart City applications and government services. NGSI-LD leverages the experiences of the developer community associated with a large number of FIWARE NGSIv2 projects, along with the Linked Data communities.

- GS1 standards ensure key processes (e.g. supply chain) run smoothly in some large industries. They provide a common language to identify, capture and share supply chain data- ensuring important information is accessible, accurate and easy to understand.

Many of these specifications are still under development. This is particularly true for the ones that are built upon OPC UA: in most cases, only the common foundation basis has been standardized, while the work continues on specific details and variants.

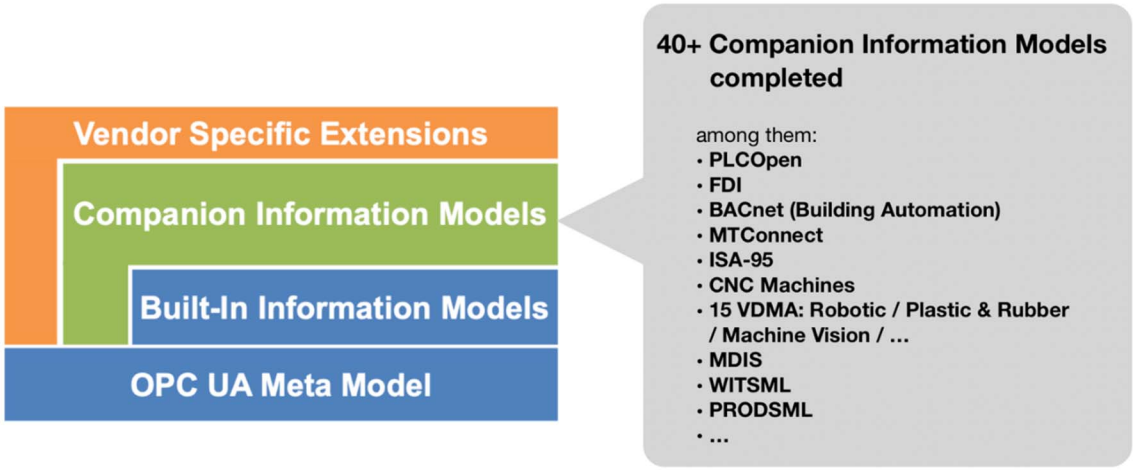
NOTE: Even if some SDOs are providing expertise and support to Industry Groups (for example, the Industry Specification Groups (ISGs) hosted by ETSI), the resulting specifications remain industry specifications.

The case of OPC-UA

OPC UA (Unified Architecture) is a standard for horizontal communication from Machine to Machine (M2M) and for vertical communication. It is promoted as the foundation for digitalization in the context of Industrie 4.0.

OPC UA provides a framework that can be used to represent complex information as objects. There is one overall OPC UA information model, which describes all basic types. This information model is incorporated into every OPC UA server and can be used by developers as a foundation for the representation of their own specific data model.

Many Industry Standards are being developed under the umbrella of OPC UA. This is enabled by the fact that OPC UA supports the notion of "Companion Industry Standards Information Models" for vertical standardization, plus vendor specific extensibility as depicted in Figure 16.



Derived from a picture available in [i.62].

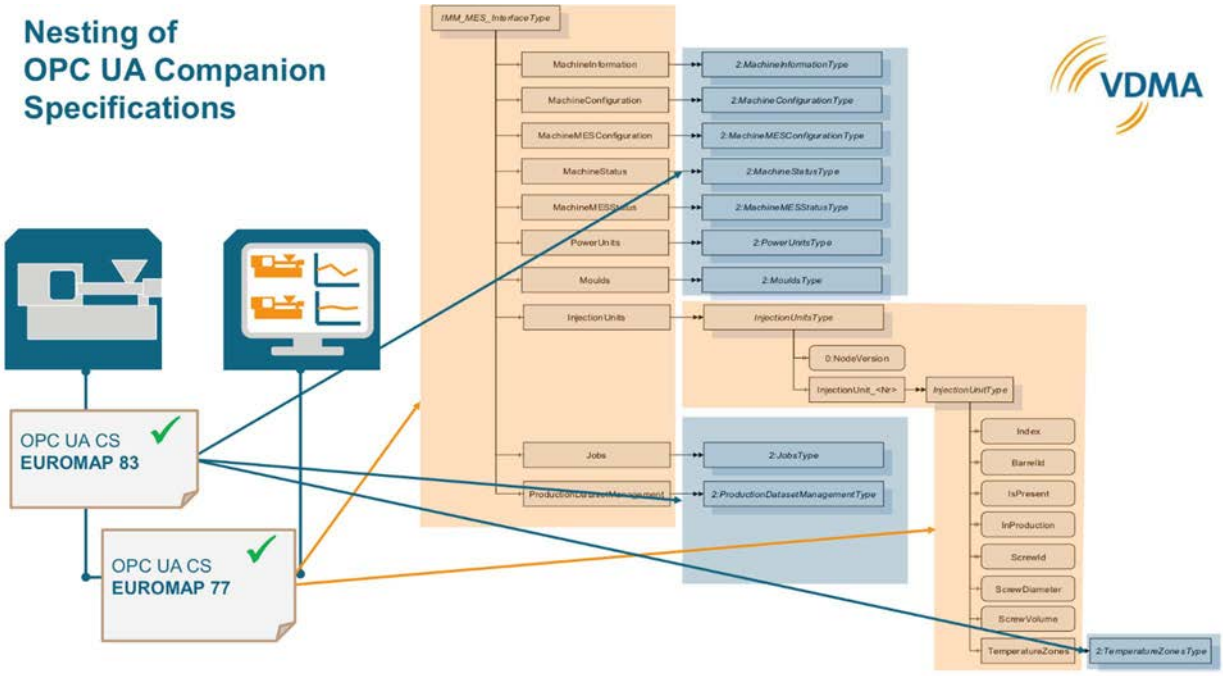
Figure 16: OPC-UA support for Information Models

In general, a Companion Specification is articulated in several Standards, often nested or otherwise related to each other. Figure 17 refers to types defined by two Companion Specifications, EUROMAP 83 [i.29] and EUROMAP 77 [i.53], published by EUROMAP, the European umbrella association of plastics and rubber machinery manufacturers (which provides technical recommendations for this industry sector and defines, amongst others, mechanical and electrical interfaces between the machines).

EUROMAP publishes two Companion Specifications that are by handled by VDMA in coordination with OPC Foundation (see: https://opcua.vdma.org/en/):

- EUROMAP 83 specifies General Type definitions. This is the basis for all other EUROMAP interfaces based on OPC UA.

- EUROMAP 77 specifies the Data exchange between injection moulding machines and Manufacturing Execution System (MES).



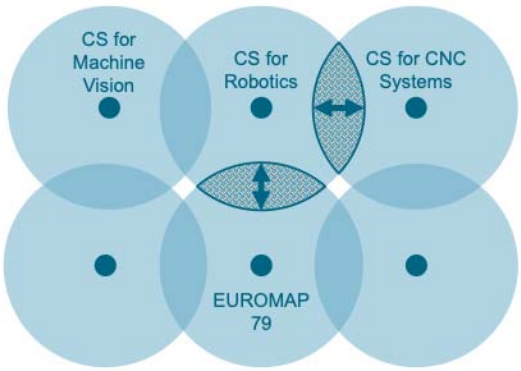
Source: [i.57].

Figure 17: OPC UA Companion Specifications - The example of EUROMAP

The benefits associated with Companion Standards are obvious:

- Without OPC UA Companion Spec, every device provides its own modelling: this involves extra effort for engineering.
- With OPC UA Companion Specifications, each device (even from different manufacturers) provides the same (base) modelling: this reduces engineering efforts.

However, the proliferation of Companion Specifications is not without drawbacks, since it can lead to double work and competition between industry standardization bodies:



Source: [i.58].

Figure 18: Risk of double work and approaches in the Companion Specifications

7.3.2.5 Proprietary Platforms

This is not in the scope of the present document.

7.3.3 Conclusions

As already pointed out in clause 7.1.4.1, the adoption of IIoT in the industry is slow with a large part of the investment and efforts has been so far on proof-on-concepts and limited implementations. However, despite a number of very strong challenges (e.g. security, safety, integration of legacy), the uptake of IIoT has started and will generate the largest business segment of IoT.

In order to address the design, development, integration and deployment of IIoT platforms, only a few mature platforms are available. Many of them are proprietary ones, but even in this category, the requirements for more openness is fostering the use of standardized elements such as standardized protocols, data models, APIs, etc.

The only global standardized platform available is oneM2M.

8 Conclusions

8.1 Lessons learned

A landscape still very fragmented and immature

Though consolidation is actively going on, the current platform landscape is still very fragmented. A plethora of available platforms indicates a certain lack of maturity with many solutions developed as an ad-hoc point answer to a very specific question and have not reached a level of maturity (e.g. TRL 9) that will guaranty that they are long-term solutions. There are multiple reasons for this, amongst which technical challenges still to address (e.g. security) or unstable business models that cannot support steady development costs or an insufficient ecosystem of developers. Consequently, the choice of platform(s) for IoT system design, development and validation may be very complex.

Proprietary platforms are not a panacea

The main value proposition of commercial (proprietary) IoT platforms is to provide users with "full experience" with a service offering ranging from connectivity up to data visualization, analytics, processing and rule-based actions (and more). These platforms can originate from specific sectors (e.g. Industrial IoT) or from large Cloud Service Providers. A growing trend for these platforms is to offer more mechanisms in support of openness such as dedicated APIs that enable access from third party applications and systems. All the provided functionalities are well integrated and optimized to work together. The trade-off is that the various parts that form the system are tightly coupled with its internals, so that attempting to use them with other platforms may prove infeasible or, at the very best, extremely impractical.

Open platform adoption in the Enterprise is (even more) complex

The requirements on platforms in the Enterprise are even more stringent than the ones for the Consumer segment. This is clearly visible in Industrial IoT (IIoT). This is in part due to the need to concurrently address the creation of new solutions (built on the deployment of data analytics supported by efficient networks) and the coexistence of legacy solutions based traditional Operation Technologies (based on PLCs, SCADA, etc.). But IIoT also has to deal with the deployment of systems with built-in safety and secure-by-design.

The key role of integration

The uptake of IoT is a supported by a constant innovation concerning all aspects of the IoT systems, in particular the creation of new services. Such services require the efficient and fast integration of very diverse new IoT components such as protocols, APIs, SW frameworks, etc. The quest for differentiation of services is transforming each IoT service development into an integration exercise with an associated effort. This effort should be dedicated to the service development and the combination of the service components rather than to the development and integration of the platform components and of the communication chain.

Different scenarios are available for platform availability

There are several ways to make a platform available to properly cover the needs of IoT system development with different levels of openness (and reduced dependency on proprietary solutions. The most relevant ones are:

1) internal development, a solution often taken by (large) incumbents;

2) integration with an ecosystem, when a significant (or de facto) platform provider wants to enlarge the breadth of its platform to new use cases and offers support for new players to integrate its ecosystem); or

3) point solutions of a specialized company coupled with cloud service provider(s).

The choice between different options is based on strategy considerations as much as technical ones.

A growing role for standardized solutions

The "Standardized approach" (SDO originated) to platforms is gaining momentum. The approach is relying on the choice of a reference (technical) architecture with a layered model, an information and interoperability strategy, a selection of Reference Points and APIs. The resulting platform can be a combination of platforms supporting one or more of the above scenarios. These solutions are based on standards developed openly with clear and fair IPR rules, and typically are not controlled by any specific company or group of companies. In the IoT domain, oneM2M is the most prominent example.

Semantic Interoperability is a key issue and a key enabler to open platform adoption

Semantic Interoperability can be designed and implemented following different approaches and techniques providing different levels of end-to-end interworking and understanding for IoT platforms. Protocol and service level interoperability is being achieved by SDOs specifications for horizontal services and APIs, such as the oneM2M standard. An important progress is being made for rich advanced data interoperability models such as ETSI SAREF and oneM2M base ontologies. Nevertheless, the wide scope of vertical applications domains prevents the design of a unique standard data models for cross-domain Semantic Interoperability. Metamodels such as oneM2M base ontology can be considered as a meta-model that may be extended for vertical domain ontology design and implementation. Simpler metamodels can be made using data structuration techniques such as JSON and XML. Moreover, some verticals can have specific constraints such as real-time or near real-time responsiveness for Industrial IoT that may prevent from adopting efficient ontology-based interoperability models. For cross-domain applications, Semantic Interoperability may require content adaptation by removing or adding details in addition to structure adaptation.

Many issues related to platform adoption are cultural

The adoption of and integration of IoT in the Enterprise is not just related to the resolution of technical problems. Non-technical, human related factors have to be considered as well, since the adoption of new technology paradigms also change the way people work. This means that organizations (i.e. companies, professional or industry associations, etc.) have first to evaluate, choose and apply innovations to achieve their organizational objectives. Once this is done, they should propose educational programs (possibly delivered as a service by professional associations) for designers and developers who do not have enough understanding or knowledge of the new technologies required (e.g. security, semantic interoperability) and make sure that they actively participate to the associated training programs.

8.2 Guidelines and Recommendations

8.2.1 Introduction

Taking into account the lessons learned and the main issues identified so far in the present document (in particular in the analysis of the Industrial IoT), the following recommendations are regarding how to improve the adoption of standardized IoT platforms by a larger community, as well as how to improve the usage (including the learning curve) of the open platforms identified.

The following recommendations are targeting (potentially overlapping) sets of stakeholders in the IoT community:

- Strategy recommendations for strategists in the organizations that have to address the choice of IoT platforms.
- Technical recommendations for stakeholders (designers, developers, device manufacturers, etc.) that have to deal with the concrete implementation of the chosen IoT platform(s) in the overall development chain of the organization.
- Recommendations to the oneM2M community regarding different approaches (technical, strategic, cultural, etc.) that support a greater impact and adoption of the oneM2M platform in the overall IoT community and, in particular, in the Enterprise business (as opposed to Consumer business).

8.2.2 Strategy Recommendations

Carefully approach the platform choices

The choice of the IoT platform has a lot of implications for the organization (e.g. a company) that has to make it. Such a choice has to take into account a number of important criteria that have to be carefully balanced before any strategy decision is taken. Amongst these criteria, important ones are: the criticality of the IoT system with respect to the strategic objectives of the organization, its investment capacity on the short and long-term, the technical skills available in-house and the need for (re-)training of the technical teams, the possibility to avoid vendor lock-in, and the potential need to align with the other legacy platform(s) already in use in the organization.

In support of these strategic choices, a variety of scenarios have to be considered. Table 4 in clause 7.1.4.4 analyses four main scenarios with associated pros and cons, and what is said in the context of Industrial IoT is valid also for organizations in other sectors:

- A first scenario that can be considered often is the internal development which is in general more adapted to large companies who have the financial and technical resources for the development of the platform and the creation of the ecosystem around it. Becoming a "de facto" platform is often reserved to large incumbents.

- For the companies that do not have the resources (or simply the objective) to develop their own platform, an associated scenario is to join the ecosystem of a "de facto" reference platform in one sector with the objective of providing incremental solutions and/or to expand the incumbent platform to a new, untapped sector. Such a scenario is a possible approach for innovative SMEs, keeping in mind that sustaining a differentiation in the long-term is difficult and costly.

- Another approach is offered by the fast maturation of Cloud Computing and the emergence of very strong cross-sector Cloud Service Providers (CSP). The main characteristic of this approach is to focus on the development of specific solutions that can be deployed on one or more of the main CPS platforms (with an IaaS, PaaS or SaaS model). It is well adapted to the organizations that have a good set of innovative point solutions (for one or more sectors) and limited resources for the creation of the overall platform and ecosystem, such as innovative SMEs or Open Source components integrators.

- A promising approach (largely addressed in the present document) is to opt for Standardized Platforms with the main objectives of globally addressing the interoperability issues (e.g. by allowing the federation of different platform though standardized middleware) and of limiting the dependency towards the "de facto" or CSP platforms. Such Standardized Platforms have one global definition and several possible implementations, emanating from industry or Open Source communities and competing for excellence.

When a final choice is made for the IoT platform, it may correspond to one of above scenarios as well as to a combination of several of them. In any case, the possibility to evolve and not to be constrained for a very long period of time by early choices is one key objective.

Avoid the dominant platform when possible

The upside of the dominant (and often proprietary) platforms is that the functionalities provided are well integrated and optimized to work together. The downside is that the various parts of the platform are tightly coupled (and more and more over time with the extension of the breadth of available functionality), so that making their usage with other platforms may prove impractical to achieve. A greater alignment on open platforms (in particular global standardized platforms and open source point solutions) is becoming not only feasible, but also sometimes a better choice to support differentiation in the longer term.

Consider standardized and open platforms seriously

Beside the many proprietary platforms currently available on the market, a range of platforms termed as "standardized" and "open" have also been developed, in standardization organizations (i.e. SDOs, SSOs and Industry Groups), in Open Source communities and also in R&D projects (such as those presented in annex A). They have been analysed in detail in clause 5.2.2. Besides specific pros and cons, all these organizations have in common to develop solutions that focus on interoperability and the integration of multiple technologies. They operate in a transparent manner in order to prevent the control by a dominant stakeholder and to offer solutions that have a global, worldwide applicability. With this approach, the development of the platform is simplified and open to multiple, interoperable implementations that allow their adopters to focus on the development of IoT services.

A large number of the contenders in the open platforms landscape may (and probably will) not become mature enough to satisfy the requirement of industry-grade IoT systems: in many cases, they have emerged from early prototype situations with very specific legacy constraints and are not going to evolve towards generic solutions. In fact, many R&D projects have not been undertaken with the objective to become platforms, but rather to investigate different aspects of interoperability that can further be integrated into "standardized" platforms.

On the other hand, more and more open solutions have been designed with generic interoperability requirements and offer the possibility to be chosen for the development of industry-grade systems, be it for the whole of them (with the very significant case of oneM2M) or for only parts (with the growing number of solutions emerging from the Open Source communities).

Prepare for massive innovation and disruptive changes in the value chains

The introduction of IoT technologies and the resulting development of new offerings from incumbents or new entrants are creating enormous changes in the established value chains. The emergence of new entrants is largely facilitated, in particular with the massive virtualization of IoT currently taking place and allowing for the creation of new ecosystems based on the emergence of new, potentially dominant, platforms proposed by Cloud Service Providers to a large number of new specialized start-ups. Hence, the introduction of IoT has to be considered as a major strategic challenge and treated as a priority by the organization impacted. Though this has been already the case for some organizations (in particular the large ones), a large number of them still have to articulate clear strategies beyond the development of initial prototypes or Proof-of-Concepts.

This recommendation is in particular valid for the SMEs. The choice of platforms with a large degree of openness will limit platforms vendor-lock, reduce investment in platform integration, and ultimately ensure that the resources of the company are put on sustainable differentiation in a controlled ecosystem.

This recommendation is also valid for the Industrial IoT. Servitization is an essential expected benefit of IIoT and it comes with great changes in the way the value chains are currently organized in various sectors of the industry.

Clearly outline the scope of the IoT (sub-) system and its integration

The introduction of the IoT system needs to be clarified upfront from the point of view of the organization's technical strategy. This is in particular true regarding the place of the IoT system in the overall organization offering (e.g. as a sub-system integrated in legacy versus as a new central system to which remaining legacy elements are integrated). Once this initial approach is followed and completed, it will be easier to understand the implication of the introduction of new technologies (such as Semantic Interoperability) and make the best trade-off between the expected substantial efficiency and performance improvement and the limitations due to the adoption of technologies that may be still at an early stage of the maturity evolution and the difficulty of working together with more mature existing technologies such as data management or data mining.

Advertise and operationalize the decisions made and the resulting successes

Even though strategic decisions to adopt new technologies (e.g. data analytics or semantic interoperability) are made, further efforts are necessary to clarify their impact and ease their diffusion and full adoption in the organization. A staged model of technology diffusion (consisting of initiation, adoption and acceptance, adaptation, routinization, and infusion) should be followed. An increased investment budget for extending systems based on the chosen technologies may offer resulting effective and sustainable services that demonstrate positive results. The strategy decisions have to be taken and made clear to the entire (technical) organization and the resulting successes to be advertised (and even rewarded explicitly).

Train the teams

Once they have made the choices regarding the selection of platforms, organizations need to provide educational programs for designers, developers, integrators and deployment teams who may not have enough understanding or knowledge of the new technologies required and make sure that they participate in the training programs. The efforts of the organization to communicate with its engineers and train them is essential to overcome the knowledge gap and can align the technical capability of the organization with the needs of customers.

8.2.3 Technical Recommendations

Enough standards to start with

A large number of standards are available. It is currently possible to use them on many aspects of the IoT system development. Examples are:

1) the integration of devices using communication protocols for which a very large set of protocol adaptation solutions exist (e.g. with oneM2M);

2) the integration of new and more dynamic information models such as the ones promoted by Semantic Interoperability (e.g. with SAREF [i.15]);

3) the development of secure-by-design solutions for which all the required standards exist (see ETSI TR 103 533 [i.1]).

In summary, there is no reason to wait for the standards gaps to be filled.

Start Small on IIoT projects

Manufacturers typically begin their approach to IIoT by first starting a pilot, or PoC (Proof-of-Concepts) project. It is usually either a small plant or manufacturing line that needs to be (re)built from scratch, or an existing one that is been retrofitted with IIoT. In any case, it is not a toy demonstration project: it is a fully operational facility, intended to carry on profitable production.

The small size of the project allows for teams to be involved more directly (they have other facilities to attend to as well), so that they can experience and understand the complexity of the undertaking, thus building internal expertise that will be most valuable when the new approach gets extended to other parts of the factory. A gradual approach will ease the learning curve while reducing possible deployment issues of the newly introduced hardware and functionalities.

Agree on a trade-off for implementable Semantic Interoperability

The choice of the level of interoperability to be adopted and the technique to be implemented can be constrained by the computation and the communication capacities. A trade-off between the richness of the model and the constrains of its implementation should lead to the choice of the appropriate approach and technique to be adopted. The choices can range from simple JSON data models for HVAC sensors values exchange by smart building IoT platforms, to elaborated domain ontologies for remote assistance and automated diagnosis using wearable sensors for internet of medical things platforms. Modularity of the design may help easier transformation and evolving of the model within the different level of interoperability and for the different modelling techniques.

Insert the new technologies in the overall development process

Very often, when it comes to developing larger scale IoT systems, many organizations prefer to start the project with proof of concepts (PoC) limited in terms of technologies, data sources and scope. During the PoC phase, the need for upfront integration of critical technologies such as security or semantic interoperability is not necessarily well addressed, and their future integration becomes much costlier and sometimes extremely difficult to integrate properly. For this reason, new technologies for IoT should be inserted at an early stage in the development process to ease subsequent large-scale deployments of IoT (sub-)systems.

8.2.4 Recommendations to oneM2M

Profiling for IIoT

oneM2M has successfully evolved from a set of standards applying to Machine-to-Machine communications into a standardized platform that address a great variety of IoT systems. This platform is a unique interoperability framework that can be embedded in a variety of hardware and software and supports a wide range of applications and services. When dealing with Industrial IoT, oneM2M should make sure that additional elements required for a larger adoption of oneM2M as the standardized platform of choice be developed (if needed), explained and marketed. These additional elements can be technical (e.g. in support of IoT Virtualization and, in particular, service orchestration, as already recommended in ETSI TR 103 527 [i.21]) as well as typical Use Cases description or training material.

End-to-end Semantic Interoperability in oneM2M-enabled IoT platforms:

The oneM2M Partnership Project is actively developing horizontal interoperability models. In addition to the advances in horizontal and semantic interoperability models and metamodels of oneM2M, possible models for domain-specific vertical interoperability can be elaborated or recommended from other standardization organisms and coalitions. This would enable the development of end-to-end semantic interoperable standardized IoT platforms, preventing the vendor lock-in and enabling the deployment of IoT-enabled industrial and societal applications.

Interworking between oneM2M and open industrial platforms

Given the wide array of available hardware platforms and operating systems, many efforts have been made by industry to develop an open industrial platform with maximum level of independence for interoperability across the Enterprise.

One important example is OPC UA, an IEC standard communication interface, with a central role in Platform Industry 4.0 (PI4.0). It is gaining support in the manufacturing industry, in particular from the manufacturers of devices and equipment. Many in the PI4.0 community believe that OPC UA is sufficient to enable all types of M2M data exchanges that are needed in the factory, and perhaps across factories as well.

However, some in the oneM2M community think that, though OPC UA is very well fit as an interface exposed by devices and machines and allowing good interoperability with them, a different kind of platform would be much better suited for data integration at higher levels.

oneM2M Industrial Domain Enablement has been published since ETSI TR 118 518 [i.64] and interworking with OPC-UA is under study in oneM2M Release 4 as depicted in Figure 19 from oneM2M TR-0018 [i.65].

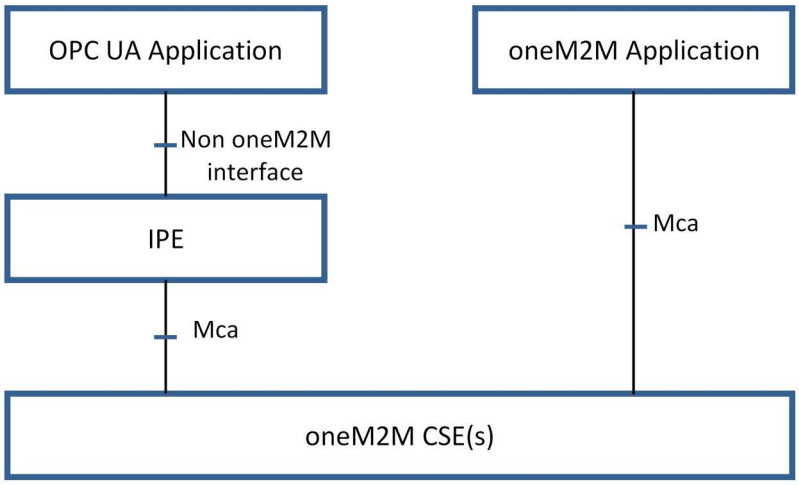


Figure 19: oneM2M OPC-UA Interworking and Functional Architecture with IPE

Such different kind of platform platform would be oneM2M connected to the OPC UA interfaces present on the shop floor. A key enabling factor for this platform would be the ability to reach interoperability also to the semantic level, between the oneM2M ontology (possibly augmented with SAREF) and the AAS (Asset Administration Shell), which makes reference to the IEC 61360 [i.52].

Some initial approaches at verifying the feasibility and usefulness of this kind of interoperability are currently attempted. An example is the Eclipse BaSyx 4.0 (spun-off from Basys 4.0, a R&D funded project where oneM2M was used to interrogate and interact with AAS attributes/properties/parameters).

Regarding standardization, two views coexist within the community of companies backing up PI4.0 does not share the same view on communication and interoperability. On the one hand, some consider that all standardization should be carried on within IEC whereas, on the other hand, another one is open to cooperation with other standardization organizations. A first joint meeting (held on February 2019), has convened the IEC and the oneM2M communities in order to share the basic elements of their respective platforms. As a follow-up, the oneM2M experts gather information about the AAS models already defined by VDMA in PI4.0 and analyse how to best integrate them into oneM2M.

It is important that open standards are used in the context of Industry 4.0 Use Cases. The opportunity created by getting the communities around ISO/IEC standards on one side, to cooperate with telecommunication standards (oneM2M via ETSI) on the other side needs to be further explored actively.

Annex A:
IoT Platforms identified by UNIFY-IoT and IoT-EPI

A.1 The platforms identified by UNIFY-IoT

UNIFY-IoT Work Package 3 has produced two deliverables (see [i.10] and [i.11]) with the purpose of identifying a list of "leading IoT platforms" that are seen as having more relevance to the IoT community (industry, research, etc.) as a whole:

- Deliverable D03.01 [i.10] "*provides an overview of IoT platforms followed by a systematic analysis and concise description of the platforms and their features. The purpose is to analyse the IoT platforms both commercial and Open Source, while mapping the IoT Use Cases and applications around the platforms and presenting the factors that are relevant for the adoption of the platform.*".
- Deliverable D03.02 [i.11] "*aims to support IoT platforms ecosystems in understanding their key success factors and their barriers for adoption. Platform adoption is considered from the point of views of both, IoT developers, who build services on top of IoT platforms and end users of these services.*".

This analysis undertaken in Deliverable D03.01 has led to the identification of 24 platforms that are shown in Figure A.1.

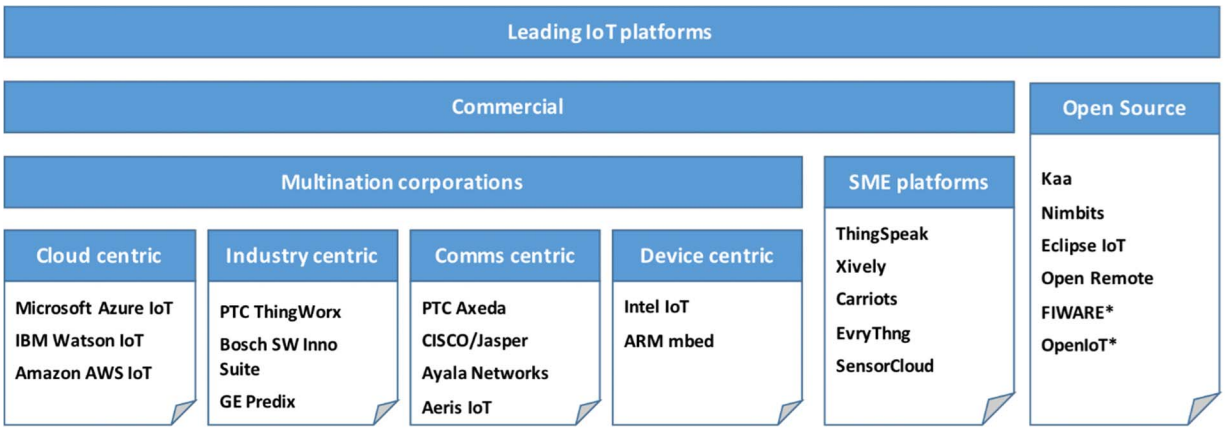


Figure A.1: UNIFY-IoT: Leading IoT Platforms selected for in-depth analysis

A.2 The platforms in the IoT-EPI projects

The 8 IoT-EPI projects (AGILE, bIoTope, BIG IoT, Inter-IoT, symbIoTe, TagItSmart! And VICINITY) are developing various interoperability solutions addressing different layers in the IoT architecture; and offering mechanisms for providing interoperability between different IoT platforms (see [i.6]).

The IoT-EPI projects are in general embedding several platforms. The Table A.1 below is listing the platforms used by the various. It can be noted that some of them are used across several IoT-EPI projects **(highlighted in bold)**.

In total, 34 different platforms are used by the 8 IoT-EPI projects referenced.

Table A.1: Platforms used by the IoT EPI Projects

Project	IoT Platforms
AGILE	Eclipse IoT, **NodeRED**, Resin.io.
bIoTope	DIALOG, eAir web, **FIWARE**, Mist, **NodeRED**, O-MI/O-DF Reference Implementation, **Open IoT**, Warp 10
BIG IoT	BEZIRK, Bitcarrier/Sensefield/FastPrk, **Open IoT**, Smart City Platform, Smart Data Platform, Traffic Information Center, Wubby
INTER-IoT	AWS, Azure, e-Care Tilab, Eclipse OM2M, **FIWARE**, I3WSN, **NodeRED**, **Open IoT**, SEAMS, Unical BodyCloud, UniversAAL
symbIoTe	KIOLA, MoBaaS, nAssist, Navigo Digitale IoT, **Open IoT**, Symphony
TagItSmart	Evrythng, **FIWARE**, RunMyProcess, SocIoTal
VICINITY	IoTivity, LinkSmart

History

Document history		
V1.1.1	December 2019	Publication
V1.1.2	December 2019	Publication