ETSI TR 103 528 V1.1.1 (2018-08)

SmartM2M;
Landscape for open source and standards for cloud native software applicable for a Virtualized IoT service layer

Reference

DTR/SmartM2M-103528

Keywords

cloud, IoT, open source, virtualisation

ETSI

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

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

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

Important notice

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

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

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

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

Copyright Notification

Contents

List of figures

List of tables

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

In addition to interoperability and security that are two recognized key enablers to the development of large IoT systems, a new one is emerging as another key condition of success: virtualization. The deployment of IoT systems will occur not just within closed and secure administrative domains but also over architectures that support the dynamic usage of resources that are provided by virtualization techniques over cloud back-ends.

This new challenge for IoT requires that the elements of an IoT system can work in a fully interoperable, secure and dynamically configurable manner with other elements (devices, gateways, storage, etc.) that are deployed in different operational and contractual conditions. To this extent, the current architectures of IoT will have to be aligned with those that support the deployment of cloud-based systems (private, public, etc.).

Moreover, these architectures will have to support very diverse and often stringent non-functional requirements such as scalability, reliability, fault tolerance, massive data, security. This will require very flexible architectures for the elements (e.g. the application servers) that will support the virtualized IoT services, as well as very efficient and highly modular implementations that will make a massive usage of Open Source components.

These architectures and these implementations form a new approach to IoT systems and the solutions that this STF will investigate will also have to be validated: to this extent, a Proof-of-Concept implementation involving a massive number of virtualized elements will be made.

The present document is one of three Technical Reports addressing this issue:

- ETSI TR 103 527 [i.1]: "Virtualized IoT Architectures with Cloud Back-ends".
- ETSI TR 103 528 (the present document): "Landscape for open source and standards for cloud native software for a Virtualized IoT service layer".
- ETSI TR 103 529 [i.2]: "Virtualized IoT over Cloud back-ends: A Proof of Concept".

1 Scope

The present document:

- Recalls the main elements of the High-Level Architecture (HLA) in support of IoT Virtualization as it is described in ETSI TR 103 527 [i.1] and how Open Source Software (OSS) and Standards can be used in the implementation of virtualized IoT systems.
- Presents, for each of the layers (and sub-layers) of the HLA, several of the OSS components that have been developed by the open source communities.
- Presents on-going developments in standardization that can be used in support of such implementations.

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 103 527: "SmartM2M; Virtualized IoT Architectures with Cloud Back-ends".

[i.2] ETSI TR 103 529: "SmartM2M; IoT over Cloud back-ends: a Proof of Concept".

[i.3] ITU-T News, October 2017: "What is 'cloud-native IoT' and why does it matter?".

NOTE: Available at http://news.itu.int/what-is-cloud-native-iot-why-does-it-matter/.

[i.4] "Cloud Native Infrastructure", Justin Garrison, Kris Nova, O'Reilly Media, 2018.

[i.5] ETSI TR 103 375: "SmartM2M; IoT Standards landscape and future evolutions".

[i.6] ETSI TR 103 376: "SmartM2M IoT LSP use cases and standards gaps".

[i.7] ETSI SR 003 392: "Cloud Standards Coordination Phase 2; Cloud Computing Standards Maturity Assessment; A new snapshot of Cloud Computing Standards".

[i.8] ETSI TS 103 264 (V2.1.1) (03-2017): "SmartM2M; Smart Appliances; Reference Ontology and oneM2M Mapping".

[i.9] White Paper: "IoT Platforms Interoperability Approaches", IoT-EPI Platforms Interoperability Task Force, 2017.

[i.10] NIST SP 1500-1: "NIST Big Data Interoperability Framework: Volume 1, Definitions".

[i.11] NIST SP 1500-2: "NIST Big Data Interoperability Framework: Volume 2, Big Data Taxonomies".

[i.12] NIST SP 1500-3: "NIST Big Data Interoperability Framework: Volume 3, Use Cases and General Requirements".

[i.13] NIST SP 1500-4: "NIST Big Data Interoperability Framework: Volume 4, Security and Privacy".

[i.14] NIST SP 1500-5: "NIST Big Data Interoperability Framework: Volume 5, Architectures White Paper Survey".

[i.15] NIST SP 1500-6: "NIST Big Data Interoperability Framework: Volume 6, Reference Architecture".

[i.16] NIST SP 1500-7: "NIST Big Data Interoperability Framework: Volume 7, Standards Roadmap".

[i.17] Recommendation ITU-T Y.4100 (former Y.2066): "Common requirements of the Internet of things".

[i.18] ISO/IEC DIS 20546: "Information technology -- Big data -- Overview and vocabulary".

[i.19] Recommendation ITU-T Y.4114: "Specific requirements and capabilities of the Internet of things for big data".

[i.20] Recommendation ITU-T Y.2068: "Functional framework and capabilities of the Internet of things".

[i.21] ISO/IEC 20547: "Information technology -- Big data reference architecture -- Part 4: Security and privacy".

3 Definitions and abbreviations

3.1 Definitions

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

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.

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

NOTE: In the present document, SSO is used equally for both Standards Setting Organization or Standards Developing Organizations (SDO).

3.2 Abbreviations

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

AIOTI	Alliance for IoT Innovation
AM	Application Master
API	Application Programming Interface
AWS	Amazon Web Services
BASH	Bourne-again shell (a Unix Shell)
CA	Certificate Authority
CDN	Content Delivery Network
CLI	Command Line Interface
CNCF	Cloud Native Computing Foundation

CPU	Central Processing Unit
CRUD	create, read, update, delete
CSC	Cloud Standards Coordination
CSP	Cloud Service Provider
CSV	Comma-Separated Values
DAG	Directed Acyclic Graph
DC/OS	Datacenter Operating System
DDL	Data Definition Language
DIS	Draft International Standard (in ISO)
DNS	Domain Name Service
DVC	Desktop Cloud Visualization
EBS	Elastic Block Store
EC2	Elastic Compute Cloud
ECR	EC2 Container Registry
ECS	EC2 Container Service
EFS	Elastic File System
ELB	Elastic load Balancing
EMR	Elastic MapReduce
ENA	Elastic Network Adapter
ES	ElasticSearch Service
ETS	Elastic Transcoder Service
EU	European Union
FTP	File Transfer Protocol
GPL	GNU General Public License
GSI	Global Secondary Index
GUI	Graphical User Interface
HDFS	Hadoop Distributed File System
HLA	High-Level Architecture
HPE	Hewlett Packard Enterprise
HSM	Hardware Security Module
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPD	HTTP Daemon
HTTPS	HyperText Transfer Protocol Secure
IaaS	Infrastructure as a Service
IAM	Identity and Access Management
ICMP	Internet Control Message Protocol
IEC	International Electrotechnical Commission
IO	Input/Output
IP	Internet Protocol
ISO	International Organization for Standardization
IT	Information Technology
ITU-T	ITU Telecommunication Standardisation Sector
JAR	Java Archive
JDBC	Java Database Connectivity
JMX	Java Management Extensions
JSON	JavaScript Object Notation
JTC	Joint Technical Committee
JVM	Java Virtual Machine
KMS	Key Management Service
LDAP	Lightweight Directory Access Protocol
LRU	Least Recently Used
MB	Megabyte
NBDIF	NIST Big Data Interoperability Framework
NIST	National Institute of Standards and Technology
OCI	Open Container Image
OLAP	Online Analytical Processing
OSS	Open Source Software
PaaS	Platform as a Service
PHP	Hypertext Preprocessor
RAM	Random Access Memory
RDF	Resource Description Framework

RDS	Relational Database Service
REST	Representational State Transfer
RM	Resource Manager
RPC	Remote Procedure Call
RTT	Round-Trip Time
SaaS	Software as a Service
SAREF	Smart Appliance Reference Ontology
SDO	Standards Development Organization
SES	Simple Email Service
SLA	Service-Level Agreement
SMS	Short Message Service
SNS	Simple Notification Service
SP	Special Publication (of NIST)
SQL	Structured Query Language
SQS	Simple Queue Service
SR	Special Report
SSH	Secure Shell
SSO	Standards Setting Organization
STF	Specialist Task Force
SWF	Simple Workflow
TCP	Transmission Control Protocol
TFS	Team Foundation Server
TLS	Transport Layer Security
TPM	Trusted Platform Module
TRL	Technology Readiness Level
TSV	Tab Separated Values
UDP	User Datagram Protocol
UI	User Interface
URL	Uniform Resource Locator
VM	Virtual Machine
VPC	Virtual Private Cloud
VPN	Virtual Private Network
W3C	World-Wide Web Consortium
WG	Work Group
XML	Extensible Markup Language

4 A Landscape for Open Source and Standards

4.1 Introduction

The IoT industry starts to understand the potential benefits of Cloud-Native Computing for the fast, effective and future-safe development of IoT systems combining the strengths of both IoT and Cloud industries in a new value proposition (see [i.3] for example).

The notion of Cloud-Native Computing is now widely supported by a large set of technologies embedded in Cloud-Native Infrastructures in support of Cloud-Native Applications. A possible definition is: "Cloud native infrastructure is infrastructure that is hidden behind useful abstractions, controlled by APIs, managed by software, and has the purpose of running applications. Running infrastructure with these traits gives rise to a new pattern for managing that infrastructure in a scalable, efficient way" (see [i.4]).

The expectation of Cloud-Native applications is to benefit from offerings from Cloud Service Providers (CSP) that may cover parts or all of the layers of Virtualized application, via Infrastructure as a Service (IaaS), Platform as a Service (PaaS) or Software as a Service (SaaS). Figure 1 presents the possible usages of such offerings in delegating more and more important parts of the underlying layers to a third-party in charge of hiding complexity, resource usage, etc.