ETSI TS 103 942 V1.1.1 (2023-11)

Testing (MTS);
Security Testing;
IoT Security Functional Modules

Reference
DTS/MTS-TST10SecTest_IoTmodule

Keywords
IoT, security, TDL, testing

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 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from:
https://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 prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

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

If you find a security vulnerability in the present document, please report it through our
Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are 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 Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, 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.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS).

Modal verbs terminology

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

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

Executive summary

The present document aims to provide a comprehensive and informative guide for individuals engaged in security testing of Internet of Things (IoT) infrastructures. It covers relevant security testing techniques and offers practical recommendations by defining TDL-TO [2] test objectives applicable across multiple industrial domains.

Introduction

With the rapid rise of interconnected devices in the Internet of Things (IoT), robust security measures have become increasingly significant. Comprehensive security testing of IoT functional modules is imperative to protect sensitive data and prevent potential vulnerabilities. In this regard, the present technical specification intends to support IoT developers and users interested in conducting security testing of IoT functional modules. It offers valuable insights into the testing aspects critical to IoT architectures used across various industrial domains.

The present document covers three foundational areas of testing for IoT architectures:

- Functional Security Testing;
- Static Application Security Testing (SAST); and
- Dynamic Application Security Testing (DAST).

The testing approach presented herein is designed to be versatile and applicable to diverse IoT architectures, irrespective of their specific domain. However, it mainly focuses on the IoTAC System Architecture, which is based on the proposed IoTAC Reference Architecture [i.9]. The IoTAC Reference Architecture builds upon the ISO/IEC 30141 [1] IoT Reference Architecture and addresses known security vulnerabilities.

The present document is structured as follows:

- Clause 4 presents the IoTAC Secure Reference Architecture and explains the key modules and components within the IoTAC System Architecture.
- Clause 5 introduces applicable security testing methods and foundational functional, SAST, and DAST principles. Besides, it provides a well-rounded methodology for transforming functional and SAST test cases into TDL-TO test purposes. This step-by-step methodology ensures practitioners can seamlessly convert their functional and SAST test cases into TDL-TO test purposes, aligning their testing efforts with the structured and formalized approach TDL-TO offers.
- Clause 6 offers concrete examples of intra and inter-component test purposes using the standardized Test Description Language (TDL) defined by ETSI ES 203 119-4 [2].
- Annex A showcases intra and inter-component test objectives as specified within the scope of the IoTAC project and documented in [i.14] and [i.15].
- Annex B outlines the related requirements from [i.15] that are associated with the test objectives.

1 Scope

The scope of the present document is designed to guide users and developers involved in the security testing of IoT systems. While the testing approach described is primarily tailored to the IoTAC System Architecture, it can be adaptable to various IoT domains. The present document covers essential aspects of testing, including Functional Testing, Static Application Security Testing (SAST), and Dynamic Application Security Testing (DAST).

Furthermore, it proposes a methodology for translating functional and SAST test cases into TDL-TO test purposes. The proposed methodology offers a systematic approach, guiding practitioners through analysing functional test case specifications, mapping the relevant information to TDL-TO concepts, and customizing the SAST ruleset to align with TDL-TO descriptions. By adopting this methodology, organizations can ensure consistency and effectiveness in translating functional and security test cases into TDL-TO test purposes, thereby enhancing the efficiency of their testing processes.

The present document goes beyond a theoretical discussion of testing principles by including concrete examples of intra and inter-component Test Purposes (TPs) using TDL-TO [2] as a specification language. It provides tangible applications for developers and users interested in IoT security testing to understand the testing approach better and see how it can be applied in practice.

2 References

2.1 Normative references

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

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

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

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

[1] ISO/IEC 30141:2018: "Internet of Things (IoT) - Reference Architecture".

[2] ETSI ES 203 119-4 (V1.5.1): "Methods for Testing and Specification (MTS); The Test Description Language (TDL); Part 4: Structured Test Objective Specification (Extension)".

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 EN 303 645 (V2.1.1) (2020-06): "CYBER; Cyber Security for Consumer Internet of Things: Baseline Requirements".

[i.2] ETSI ES 203 119-1 (V1.6.1) (2022-05): "Methods for Testing and Specification (MTS); The Test Description Language (TDL); Part 1: Abstract Syntax and Associated Semantics".

[i.3] ETSI 203 119-2 (V1.5.1) (2022-05): "MTS; The Test Description Language (TDL); Part 2: Graphical Syntax".

[i.4] ETSI 203 119-3 (V1.6.1) (2022-05): "MTS; The Test Description Language (TDL); Part 3: Exchange Format".

[i.5] ISO/IEC 19508:2014(E): "Information Technology - Object Management Group Meta Object Facility (MOF) Core".

[i.6] OMG (2012-01): "OMG Object Constrained Language (OCL)", (V2.3.1) (2012-01).

[i.7] ETSI ES 202 553 (V1.2.1) (2009-06): "Methods for Testing and Specification (MTS); TPLan: A notation for expressing Test Purposes".

[i.8] ETSI ES 201 873-1 (V4.10.1) (2018-05): "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language".

[i.9] IoTAC project Deliverable D2.3: "Architecture Design Document", Public Deliverable, February 2022.

[i.10] OWASP: "Static Code Analysis (SCA)".

[i.11] OWASP: "Application Security Verification Standard (ASVS)", March 2019.

[i.12] ETSI TS 103 701 (V1.1.1) (2021-08): "CYBER; Cyber Security for Consumer Internet of Things: Conformance Assessment of Baseline Requirements".

[i.13] IoTAC Project Deliverable D6.2: "Definition of the Development Integration Environment and KPIs", Public, August 2021.

[i.14] IoTAC project Deliverable D6.3: "Integration and Testing of the IoTAC Architecture", Confidential, March 2023.

[i.15] IoTAC project Deliverable D2.2: "Requirements and use-cases specification", Confidential, August 2021.

[i.16] TDL Open Source Project (TOP).

[i.17] OWASP Top Ten 2017: "A3:2017-Sensitive Data Exposure".

[i.18] OWASP Top Ten 2017: "A6:2017-Security Misconfiguration".

[i.19] MITRE, CWE-326: "Inadequate Encryption Strength".

[i.20] MITRE, CWE-327: "Use of a Broken or Risky Cryptographic Algorithm".

[i.21] CWE/SANS Top 25: "Porous Defences".

[i.22] OWASP: "IoT Security Verification Standard (ISVS)", October 2019.

[i.23] OWASP: "Cheat Sheet Series - Password Storage Cheat Sheet".

[i.24] MITRE, CWE-328: "Use of Weak Hash".

[i.25] MITRE, CWE-916: "Use of Password Hash with insufficient effort computation".

[i.26] OWASP Top Ten 2017: "A2:2017 - Broken Authentication".

[i.27] MITRE, CWE-521: "Weak Password Requirements".

[i.28] Sonar Rules, Python Static Code Analysis - Code Smell RSPEC-3516.

[i.29] Sonar Rules, Phyton Static Code Analysis - Code Smell RSPEC-2387.

[i.30] MITRE, CWE-798: "Use of hard-coded credentials".

[i.31] MITRE, CWE-256: "Use of hard-coded password".

[i.32] MITRE, CWE-338: "Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)".

[i.33] MITRE, CWE-330: "Use of Insufficiently Random Values".

[i.34] CERT, MSC02-J: "Generate strong random numbers".

[i.35] CERT, MSC30-C: "Do not use the rand() function for generating pseudorandom numbers".

[i.36] CERT, MSC50-CPP: "Do not use std::rand() for generating pseudorandom numbers".

[i.37] OWASP Top 10-2021.

[i.38] CVE-2019-13466.

[i.39] CVE-2018-15389.

[i.40] CVE-2013-6386.

[i.41] CVE-2006-3419.

[i.42] CVE-2008-4102.

[i.43] Java Design Patterns.

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

black-box testing: testing without an understanding of the system's internal structure

Dynamic Application Testing (DAST): testing methodology that analyses a running application for potential security vulnerabilities during execution

functional security testing: verification of a software's security mechanisms to ensure they operate as expected and safeguard the system

reference architecture: blueprint providing shared terminology and reusable design to guide specific architectural developments

Static Application Testing (SAST): testing methodology that analyses the source code of the application for potential security vulnerabilities without actually executing the application

system under test: real, open system that contains the implementation under test

white-box testing: testing components or systems internally by analysing their internal structures

3.2 Symbols

Void.

3.3 Abbreviations

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

AADRNN	Auto-Associative DRNN
AD	Attack Detection
ADT	Attack Detection Training
AI	Artificial Intelligence

AID	Application ID
APDU	Application Protocol Data Unit
API	Application Programming Interface
AR	Automatic Reconfiguration
ARNN	Adversarial Random Neural Network
ASD	Application and Service Domain
ASIC	Application Specific Integrated Circuit
ASVS	Application Security Verification Standard
BSS	Business Support Systems
CA	Certification Authority
CI	Continuous Integration
CIN	Card Identity Number
CLI	Command Line Interface
CS	Certificate Server
CSR	Certification Signing Request
CWE	Common Weakness Enumeration
DAST	Dynamic Application Security Testing
DB	Data Base
DDoS	Distributed Denial of Service
DoS	Denial of Service Attack
DPE	Data Processing Engine
DR	Data Routing
DRNN	Dense Random Neural Network
FEAM	Front-End Access Management
FPGA	Field Programmable Gate Array
FPGA	Field Programmable Gate Array
FTP	Functional Test Purposes
GP	Get Parameters
GPU	Graphics Processing Unit
HP	Honeypot
HTTP	Hypertext Transfer Protocol
ID	Identifier
IDD	Infected Device Detection
IDE	Integrated Development Environment
IoT	Internet of Things
IP/MAC	Internet Protocol/Medium Access Control
ISO	International Organization for Standardization
ISVS	IoT Security Verification Standard
JSON	JavaScript Object Notation
JWT	JSON Web Token
KPI	Key Performance Indicator
LDAP	Lightweight Directory Access Protocol
LR	Likelihood Ratio
ML	Machine Learning
MOF	Meta-Object Facility
MPPE	Multi-Purpose Processing Engine
MTS	ETSI Technical Committee - Methods for Testing and Specification
N/A	Not Applicable
NWAA	Network Wide Attack Assessment
NWAD	Network Wide Attack Detection
OCL	Object Constrained Language
OMD	Operation and Management Domain
OSS	Operational Support Systems
OTP	One Time Password
OWASP	Open Web Application Security Project
PBKDF2	Password-Based Key Derivation Function 1 and 2
PED	Physical Entities Domain
PHP	Hypertext Preprocessor
PICS	Protocol Implementation Conformance Statement
PMC	Probe Management and Configuration
PR	Probe Registry
PRNG	Pseudorandom Number Generation

RA	Reference Architecture
RAID	Resource and Interchange Domain
RM	Reference Model
RMS	Run-time Monitoring System
RNG	Random Number Generation
RNN	Random Neural Network
SAST	Static Application Security Testing
SCA	Static Code Analysis
SCD	Sensing and Controlling Domain
SDK	Software Development Kit
SG	Security Gateway
SHA	Secure Hash Algorithm
SP	Set Parameters
SQL	Standard Query Language
SSA	Server Secure Application
S-SDLC	Secure Software Development Lifecycle
SSH	Secure Shell Protocol
SSL	Secure Socket Layer
SUT	System Under Test
TC	Technical Committee
TDL	Test Description Language
TDL-TO	TDL Test Objective
TISTQB	International Software Testing Qualifications Board
TLS	Transport Layer Security
TO	Test Objective
TOP	TDL Open Source Project
TP	Test Purpose
TPLan	Test Purpose Language
TTCN-3	Testing and Test Control Notation version 3
UD	User Domain
UML	Unified Modelling Language
VM	Virtual Machine
XF	Exchange Format
XSS	Cross-Site Scripting

4 Specification of the IoT Modules

4.1 IoTAC Secure Reference Architecture

ISO/IEC 30141 [1] provides a comprehensive and flexible framework that organizations can use to design and implement secure IoT systems in various domains. Its international recognition and emphasis on risk management make it a reliable choice for organizations looking to deploy secure IoT solutions. Despite this, ISO/IEC 30141 [1] does not address security aspects sufficiently since it only offers high-level security recommendations and guidelines. The IoTAC project proposes a Secure IoT Reference Architecture based on the ISO/IEC 30141 [1] RA to solve this problem [i.9]. In Figure 1, the extended ISO/IEC 30141 [1] Domain-based Reference Model illustrates the mapping of newly introduced IoTAC components to their corresponding domains.

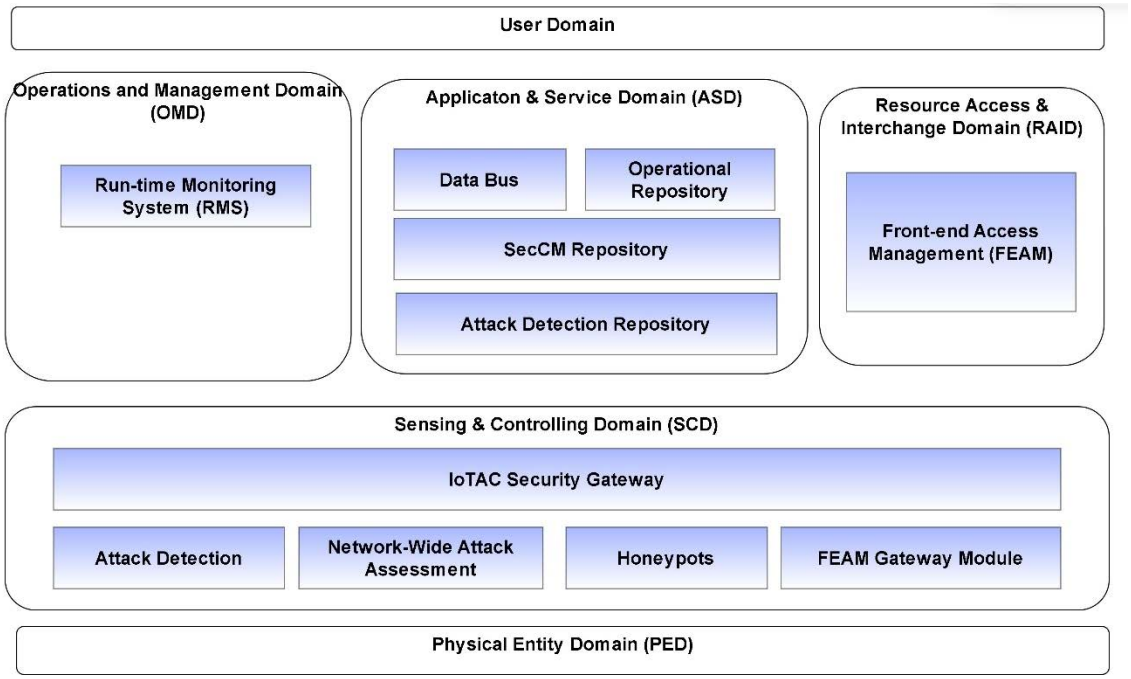

Figure 1: Extended ISO/IEC 30141 [1] Reference Model (RM)

The Physical Entities Domain (PED) defines all physical objects that are part of IoT systems, including sensors, actuators, and devices, as illustrated in Figure 2.

The Sensing and Controlling Domain (SCD) bridges the digital and physical worlds, encompassing sensors that monitor various aspects of PED and manipulating actuators. Additionally, the SCD incorporates IoT gateways, local data stores, and services to facilitate efficient data processing and system control, see ISO/IEC 30141 [1]. The IoTAC Reference Architecture (RA) introduces the following components to the SCD: IoT Security Gateway, AI-based Attack Detection, AI-based Network Wide Attack Assessment (NWAA), Honeypots and FEAM Gateway:

- **The IoT Security Gateway** is a secure entry point for IoT devices in an enterprise network, protecting sensitive data from potential threats. It performs various functions, such as receiving, verifying, and distributing sensor messages and relaying control commands to actuators. Its primary tasks include receiving and scanning messages from sensors and devices. Besides, it logs security events, detects intrusions within the internal network, ensures device cybersecurity, and provides control methods for connected devices. The gateway has robust encryption techniques to safeguard sensitive data and prevent unauthorized access. Additionally, it enforces security policies and controls data flow to minimize attack surfaces, enhancing system security.

- **The AI-based Attack Detection** uses the Dense Random Neural Network (DRNN) model and network metrics derived from the network traffic measurements to ensure IoT security. It detects malicious activity by learning normal communication patterns among IoT devices, detecting deviations, and sending Threat Notification messages through the IoT Security Gateway.

- **The AI-based Network Wide Attack Assessment (NWAA)** begins by conducting a security assessment of each device in the IoT network to provide a comprehensive evaluation of the system's security.

- **The Honeypots** employ advanced anomaly detection algorithms to redirect attackers toward isolated environments and monitor their behaviour, facilitating early identification of potential intrusions and underlying causes of attacks.

- **The FEAM Gateway** is an integral Front-end Access Control Management system component. Its primary function is to serve as an intermediary between the protected device or system and the FEAM Management module. In this capacity, it assumes responsibility for regulating access to the protected system. By providing an additional layer of security, the FEAM Gateway ensures that only authorized users and devices are granted access to the system.

The Resource and Interchange Domain (RAID) includes all the functions required to access the IoT system resources, see ISO/IEC 30141 [1]:

- **The Front-End Access Management (FEAM)** component represents an innovative capability-based access control system that fulfils the requirements of the Zero Trust concept in CWE/SANS Top 25 [i.21]. It relies on using smart cards to store sensitive data, digital signatures and certificates, multi-factor authentication, and fine-grained privileged access management. Additionally, it adheres to the principle of least privilege on a session level. One novel feature of FEAM is the separation, both in time and space, of the delegation of access privileges from authentication and authorization processes.

The Operation and Management Domain (OMD) contains functional components responsible for the overall management of the IoT system. According to the ISO/IEC 30141 [1] RA, the OMD consists of two primary functional components: Operational Support Systems (OSS) and Business Support Systems (BSS). In addition, the IoTAC Secure RA proposes the introduction of an additional RMS component:

- **The Run-time Monitoring System (RMS)** provides a real-time service that collects security-related data from monitored IoT system components or applications and stores it for subsequent processing. The system employs analytics algorithms to analyse the collected data, intending to detect abnormal patterns. The RMS collects and publishes data to the monitoring platform using monitoring probes.

The Application and Service Domain (ASD) represents the collection of functions implementing application and service logic that realizes specific business functionalities for the service providers in the ASD, see ISO/IEC 30141 [1]. Data Bus, Observational Repository, and Attack Detection Repository were identified as essential IoTAC components during the system analysis phase:

- **The Data Bus** is a communication channel that routes all real-time data within IoTAC's platform. The platform supports publish-subscribe functionality, enabling users to push their data or subscribe to receive data that meet their needs. IoTAC's Data Bus facilitates real-time data exchange among various components.

- **The Observational Repository** is a repository that allows the permanent storage of data from the IoTAC platform that is monitored or processed.

- **The Attack Detection Repository** hosts both the offline-trained version of the AD model for parameter storage and the online-trained version for performance evaluation.

- **The User Domain (UD)** includes all users interacting with the IoT system through various interfaces.

Figure 2 illustrates the elaborated IoTAC Domain-based Reference Model indicating the information flow between the components. The IoTAC runtime components produce results aligned with Threat Reporting messaging schemes, as shown in Figure 2. Threat Reports are then published to the Data Bus within the ASD using a publish/subscribe function. By subscribing to these messages, a reporting dashboard or any third-party application can display Threat Reports to end users or facilitate their further processing. More information can be found in the public IoTAC Deliverable D2.3 Architecture Design Document [i.9].

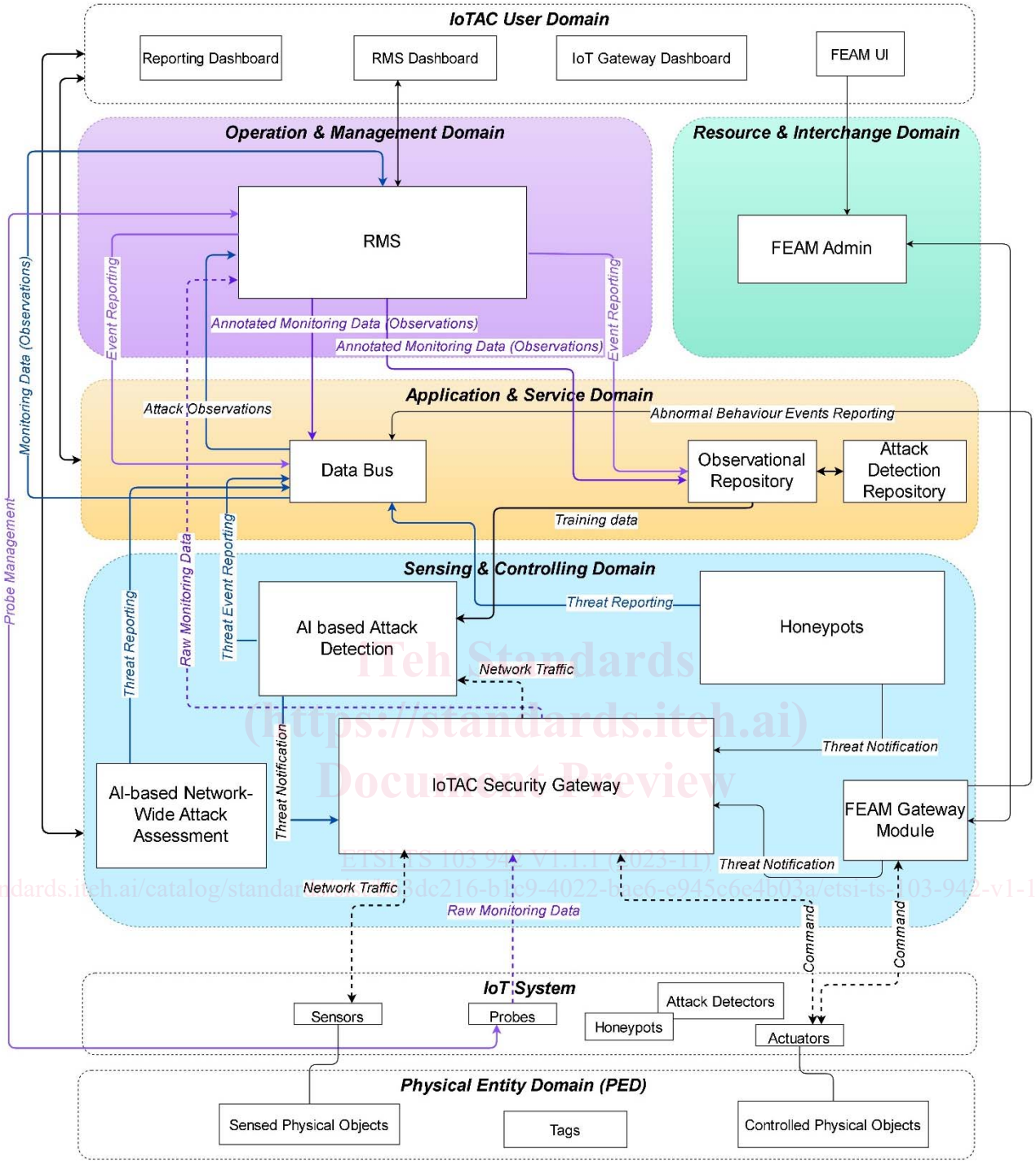

Figure 2: IoTAC Domain-based Reference Model (detailed view) [i.9]