SEMIoTICS

Deliverable D5.1
SEMIoTICS KPIs and Evaluation Methodology

Deliverable release date	31.01.2020 (revised on 21.04.2021)
Authors	1. Juan David Parra, Korbinian Spielvogel, Felix Klement, Henrich C. Pöhls (UP) 2. Darko Anicic, Arne Bröring, Ermin Sakic (SAG), 3. Nikolaos Petroulakis (FORTH), 4. Jordi Serra, Luis Sanabria-Russo, David Pubill, Angelos Antonopoulos and Christos Verikoukis (CTTC) 5. Domenico Presenza (ENG) 6. Mirko Falchetto (ST) 7. Prodromos-Vasileios Mekikis (IQU) 8. Urszula Stawicka (BS)
Responsible person	Felix Klement, Korbinian Spielvogel, Henrich C. Pöhls (UP)
Reviewed by	All
Approved by	PTC Members (Vivek Kulkarni, Nikolaos Petroulakis, Ermin Sakic, Mirko Falchetto, Domenico Presenza, Verikoukis Christos) PCC Members (Vivek Kulkarni, Nikolaos Petroulakis, Verikoukis Christos, Georgios Spanoudakis, Domenico Presenza, Danilo Pau, Joachim Posegga, Darek Dober, Kostas Ramantas, Ulrich Hansen)
Status of the Document	Final
Version	1.0
Dissemination level	Public

Table of Contents

TABLE 1 ACRONYM TABLE

Acronym	Definition
QoS	Quality of Service
SPDI	Security, Privacy, Dependability and Interoperability
IRI	Internationalized Resource Identifiers
RDFS	Resource Description Framework Schema
OWL	Web Ontology Language
WoT	Web of Things
TD	Thing Description
MOT	Monitored Object Types
UOT	Use-case Object Types
MXQ	Monitored Cross-platform Queries
XQ	Cross-Platform Queries
ED	Event Detections
CED	Compliant Event Detections
PMT	Patterns Monitoring Tasks
DPMT	Delegated Patterns Monitoring Tasks
SDN	Software defined networking
NFV	Network function virtualization
NS	Network Services
VNF	Virtualised Network Functions
OSM	Open Source MANO
VNFM	Virtualised Network Functions Manager
PNF	Physical Network Functions
KPI	Key Performance Indicator
BSV	Backend Semantic Validator

1. INTRODUCTION

This deliverable contains the key performance indicators (KPI) by which the SEMIoTICS project will measure the success of the developments taken place in the project. Thus, this deliverable is collecting all the KPIs in one single document to provide a one-stop-shop for all partners inside the project to see the important goals to achieve their current and future work in order to reach the objectives of SEMIoTICS.

In more detail, this deliverable contains and further specifies the KPIs as well as their evaluation methodology including where necessary their baseline, which will be used to assess SEMIoTICS in the different application target areas. The KPIs found in this deliverable have received careful consideration by the involved partners of the consortium and extend and refine those listed as an example during the early project stages. This document now reflects the full set of the project's objectives. Those are aligned with the more technical functional and non-functional requirements of the project at the current state based also on those identified in WP2 for both the SEMIoTICS framework and its usage scenarios. In order to gain a definition of concrete technical, business, and usability criteria that would allow the evaluation of SEMIoTICS's performance with respect to the KPI, this document establishes the baseline performance measures if required. As some KPIs also define the evaluation methodology of the project itself, i.e., the evaluation of dissemination targets, this document also contains those.

It should be noted that the evaluation of the KPIs and the checking of fulfilment of technical requirements is beyond the scope if this deliverable, but it does provide pointers to where within the SEMIOTICS's framework the technical measurements for technology-related KPIs will be taken and what kind of tests will be carried out. With the included task to KPI mapping (Figure 0) one can find the respective task and then consult the task's latest deliverable. With the aggregated list of KPIs provided within, future deliverable of SEMIoTICS will be able to provide a consistent view, using the KPI's identifier (KPI-ID) to allow for easy cross-referencing. The respective deliverables of these tasks will then show how the evaluation of these KPIs due to the results presented in the respective deliverable, e.g. for example for technical functional KPIs by testing the functionality in the lab or by integrating the technical functionality in a demonstrator related to the use case. Thus, this deliverable contains the performance indicators the project has devised to steer towards its objectives and in upcoming deliverables when providing final or interim results of the evaluation, the project will refer to them using the proposed unique identifier.

2. KPI DEFINITION TEMPLATE

This section provides a detailed definition of the KPIs, which includes:

- concrete technical, business, and usability criteria for the evaluation;
- where needed also a description of how to establish the baseline performance measures of the evaluation criteria (e.g., minimum required accuracy of predictions and adaptation response time);
- how SEMIoTICS partners will carry out those tests, e.g. Lab-based experiments, Trial applications;
- and to which usage scenario(s) they relate.

In more detail, the following fields are used to define the specific KPI's and this constitutes the template for the definition to describe each KPI (see 2):

- **KPI-ID**: A unique ID number for the identification of the KPI, which is related to the Objective Number.
- **Goal**: The overall aimed goal which is being attempted to achieve
- **Name**: Short name for the KPI
- **Leader**: A list of all responsible partners that contribute to the evaluation/validation of the described KPI
- **Scope**: Lists the applicable scope of the KPI: if it is applicable in a specific use case or use cases the applicable use case is given; if it is a KPI related to the overall SEMIoTICS framework or platform it is "framework"; if it is a related to the project's goals it is "project". Where appropriate we will also name which of the three use cases this KPI especially applies to, e.g. "UC1, UC2" means this KPI matters not in UC3, but was required for the successful use of the framework in UC1 and UC2.
- **Description**: Detailed description of the KPI
- **Mapping to measurement points**: Description of involved components or the related deliverables, i.e., answer the question where the metric is measured
- **Methodology**: The concrete definition of the method how the KPI is measured; Note, if the KPI is applicable in different use cases, there could be either a generic methodology or a specific one for each use case.
- **Baseline**: The concrete definition of the state without SEMIoTICS solution, known as baseline (if applicable to define the KPI's goal)

Note that the actual evaluation of the KPIs is not in the scope of the deliverable D5.1. The methodology as well as the baseline for each of the KPIs was created with the knowledge of one or more domain experts from the respective use-case or the technical domain. Evaluation of the KPI thus either directly involves domain experts because they must perform the evaluation or they are only indirectly involved as they set the underlying requirements in WP 2 or because the domain expert was involved in the formulation of the evaluation methodology or the baseline (where necessary).

Some KPIs might not need a baseline, simply because one is able to assess the objective. As the template for the definition of the KPI contained a field for baseline it will be filled with "not applicable" or "N/A". Where necessary and possible we also give details on why no baseline is required. This might be due to the fact that there is no baseline necessary for the assessment e.g. when counting defined achievements, or when the baseline is implicitly introduced inside the methodology and we did not replicate it for brevity and readability. Let us stress that a missing baseline does not imply that the KPI can not be measured.

Here it should be noted, that the template provided is only intended for being used in the definition and presentation of the KPIs in this deliverable. While the presentation may be updated or adapted in future deliverables, the intention is that upcoming deliverables of the project retain the unique identifier when providing the results of the evaluation.

TABLE 2: KPI DEFINITION TEMPLATE

KPI-ID	<unique ID>	**Goal**	<aimed goal>
Name	<short name>		
Leader	<name of responsible partner>	**Scope**	<"framework", "project", or the applicable use-case abbreviation>
Description	<description of the KPI>		

Mapping to measurement points	<measurement points mapping>
Methodology	<measurement explanation>
Baseline	<concrete state without SEMIoTICS (i.e. the baseline)>

3. PROJECT-WIDE KPIS

In this section we present the KPIs that correspond to the whole project of SEMIoTICS and explain how they are mapped to each of the five objectives. Note, the KPIs are a different from the technical requirements which have been elicited in D2.2 and D2.3: Those requirements are checked for fulfilment in the respective deliverable, but of course correspond to the functional and non-functional requirements of each scenario. The KPIs are meant to track the project's objectives. For ease of reference the KPI-ID already indicates which objective the KPI belongs in the first level, e.g. KPI with the ID KPI-4.5 belongs to a performance indicator that helps to monitor the project’s performance towards the goals of objective 4. In the following table we provide a

Objective		KPI-ID			Description
					Short
1	SPDI Patterns	KPI-1.1			Number of SPDI Patterns
		KPI-1.2			Pattern Language
2	Semantic Interoperability	KPI-2.1			Semantic descriptions for 6 types of smart objects
		KPI-2.2			Data type mapping and ontology alignment
		KPI-2.3			Semantic interoperability with 3 IoT platforms
3	Monitoring Mechanisms	KPI-3.1	KPI-3.1.1		Generating monitoring strategies in the 3 targeted IoT platforms
			KPI-3.1.2		Fuse results from these monitors
			KPI-3.1.3		Performing predictive monitoring with an average accuracy of 80%
		KPI-3.2			Delivery of a monitoring language
4	Multi-layered Embedded Intelligence	KPI-4.1			Delivery of lightweight ML algorithms
		KPI-4.2			Delivery of mechanisms with adaptation time of 15ms
		KPI-4.3			Delivery of adaptations mechanisms enabling improvement by at least 20%
		KPI-4.4			Detection time of less than 10 ms
		KPI-4.5			Baseline improvement of 20% adaptation time
		KPI-4.6			Development of new security mechanisms/controls
5	IoT-aware Programmable Networks	KPI-5.1			Deployment of a multi-domain SDN orchestrator
		KPI-5.2			Service Function Chaining (SFC) of a minimum 3 VNFs
6	Development of a Reference Prototype	KPI-6.1			Reduce Required Manual Interventions
		KPI-6.2			Leveraging upon FIWARE assets
		KPI-6.3			Delivery of 3 prototypes of IIoT/IoT applications
7	Promote the adoption of EU technology offerings internationally	KPI-7.1			Provision the SEMIoTICS framework and building blocks
		KPI-7.2			Achieve influencer status within major standardization efforts
		KPI-7.3	KPI-7.3.1	KPI-7.3.1.1	Project website
				KPI-7.3.1.2	Push announcements
				KPI-7.3.1.3	Regular Newsletter
				KPI-7.3.1.4	Brochure
				KPI-7.3.1.5	Technical video
			KPI-7.3.2	KPI-7.3.2.1	Journal publications
				KPI-7.3.2.2	Magazine publications
				KPI-7.3.2.3	Conference Publications
				KPI-7.3.2.4	Special Issues
			KPI-7.3.3	KPI-7.3.3.1	Conference organizations
				KPI-7.3.3.2	Workshops
				KPI-7.3.3.3	Summer Schools
			KPI-7.3.4	KPI-7.3.4.1	Exhibition demonstrations
				KPI-7.3.4.2	EU demonstrations
				KPI-7.3.4.3	Conference demonstrations

brief overview of the KPI's mapped to their corresponding project objective. In Appendix A we provide a more comprehensive table mapping the KPIs to tasks.

3.1. Objective 1 – SPDI Patterns

The main scope of Objective 1 is the development of patterns for orchestration of smart objects and IoT platform enablers in IoT applications with guaranteed Security, Privacy, Dependability and Interoperability (SPDI) properties. A more detailed description of this objective defines that the achievement of this objective will be based on developing patterns **(WP4)** defining generic ways for integrating and orchestrating different types of smart objects and components that can guarantee specific SPDI properties, henceforth referred to as SPDI patterns. This guarantee will be based on test evidence and/or formal verification, as appropriate for the type of properties and the smart objects/components orchestrated by the pattern. SPDI patterns should cover both vertical composition of smart objects at different layers in the implementation stack of IoT applications – including sensors/actuators, network, infrastructure, IoT platform and IoT application components – and horizontal composition of smart objects that appear at any of these layers, as necessary. SPDI patterns cover the different and heterogeneous orchestration models required for such applications, including message-driven, event-driven and data-driven models. The development of SPDI patterns will also require the definition of a pattern language **(WP4)** supporting the specification of all facets of patterns. It should also enable the automated application of patterns to realize key capabilities offered by the SEMIoTICS framework, including orchestration, verification and adaptation of smart object compositions at runtime.

3.1.1. DELIVERY OF SPDI PATTERNS

KPI-ID	KPI-1.1	**Goal**	36
Name	Number of SPDI Patterns		
Leader	STS/FORTH	**Scope**	Project
Description	Delivery of verified patterns supporting the composition of IoT applications and smart objects in ways that preserve the basic security properties of Confidentiality, Integrity and Availability, as well as Privacy, Dependability and Interoperability properties for all three usage scenarios. Overall, we will develop 36 patterns covering the six core property types (i.e., Confidentiality (s), Integrity (s) and Availability (s), Privacy (p), Dependability (d) and Interoperability (I)) for three data states (i.e., data-in-transit, data-at-rest and data-in-processing) and two cases of IoT platform connectivity (i.e., within and across IoT platforms).		
Mapping to measurement points	The first set of SPDI patterns are presented in D4.1. The final set of developed SPDI patterns will be documented in deliverable D4.8 – "SEMIoTICS SPDI Patterns (final)", which is due in M28 of the project. This deliverable will provide the final version of the language for specifying SPDI patterns (related to KPI-1.2 below) and all the SPDI patterns developed in SEMIoTICS. These should cover all properties, data states, and use cases of platform connectivity, as defined above.		
Methodology	Since the developed SPDI patterns will be specified in a document format, the verification of the delivered (final) number of developed patterns will be carried out through review of deliverable D4.8 – "SEMIoTICS SPDI Patterns (final)".		
Baseline	N/A (there are no such pattern rules defined prior to the start of the project)		

3.1.2. PATTERN LANGUAGE

KPI-ID	KPI-1.2	**Goal**	True
Name	Machine-processable language for pattern		
Leader	STS/FORTH	**Scope**	Framework
Description	Delivery of a pattern language enabling the specification of machine-processable patterns.		

Mapping to measurement points	The final design and specification of the SEMIoTICS pattern language will be provided in D4.8 – "SEMIoTICS SPDI Patterns (final)", which is due in M28 of the project. This deliverable will also document the approach for deriving machine-processable patterns from the Pattern Language, also documenting them in a machine processable format. The actual components that will process said patterns will be the "Pattern Engine" components (reasoning engines) deployed at all three layers (backend, network, field) of the SEMIoTICS architecture.
Methodology	The verification of delivery of the SEMIoTICS Pattern Language and its usability for defining machine-processable patterns will be achieved through document review of deliverable D4.8 – "SEMIoTICS SPDI Patterns (final)". The verification of the format of the derived machine-processable patterns will be achieved through verification of the correct functionality of a Drools rules reasoning engine (as embedded in Pattern Engines) when processing said derived patterns.
Baseline	N/A (there was no such language defined prior to the start of the project)

3.2. Objective 2 – Semantic Interoperability

The main scope of objective 2 is to focus on the development of semantic interoperability mechanisms for smart objects, networks and IoT platforms. The main description of this objective defines that SPDI patterns will define the necessary conditions for achieving smart object interoperability at the level of data, meta-data and operations available in smart objects. They will also define the transformations that can be applied using generic protocols and annotation schemes based on semantic models to achieve interoperability. SEMIoTICS will develop **(WP3, WP4)**: (i) semantic annotation schemes required for achieving semantic interoperability; (ii) mechanisms to configure smart objects and/or the IoT platforms through which they become available dynamically, and to generate the necessary for semantic interoperability data transformations; and (iii) mechanisms supporting the validation of the semantic interoperability conditions for smart object orchestration. Semantic annotation schemes development will be based on transferring principles of Semantic Web Services to the IoT domain. This requires defining extended descriptions of smart objects, involving not only the smart object interface descriptions but also their pre-/post-conditions, and investigating which parts of these extended descriptions can be produced automatically (e.g. from smart object implementations). Each description element will be defined using concepts from semantic models and be aligned with work on the W3C WoT Thing Description (https://w3c.github.io/wot-thing-description). The annotation of SPDI patterns will be realized using Internationalized Resource Identifiers (IRIs) to link semantic concepts (e.g., included in Resource Description Framework Schema (RDFS) or Web Ontology Language (OWL) ontologies) and smart object descriptions. The development of semantic transformation mechanisms will follow two directions. The first establishes data flows between smart objects by applying patterns with mappings between different data types. The second involves defining interoperability conditions guaranteeing that, whenever two concepts are linked through data flow connections, they can be mapped to each other through ontology alignment methods or suitable transformations. Validation mechanisms for semantic interoperability then ensure that data type mappings exist and that the interoperability conditions of the pattern hold for the particular pattern instantiation.

3.2.1. SEMANTIC DESCRIPTIONS FOR 6 TYPES OF SMART OBJECTS

KPI-ID	KPI-2.1	**Goal**	6
Name	Semantic descriptions for 6 types of smart objects		
Leader	SAG/ENG/ST	**Scope**	UC1, UC2, UC3
Description	Semantic descriptions for all the types of smart objects which are necessary for the usage scenarios.		

Mapping to measurement points	Thing Directory in the backend layer and in field layer. Semantic descriptions for all the types of smart objects will be provided based on W3C Web of Things (WoT) standard. In particular, WoT Thing Description standardized format for describing IoT things will be used. Each sensor, actuator or thing from all SEMIoTICS use cases will be identified and for each smart object one Thing Description (TD) will be provided. We will use iotschema.org to semantically annotate each TD. If we discover that iotschema does not exist for certain smart objects or their parts of, we will provide a proposal for an extension of iotschema.org.
Methodology	This KPI provides insight into semantic descriptions that are provided for all smart objects, which are used in SEMIoTICS use case scenarios. The goal is to enable smart objects to become interoperable. # smart objects from usage scenarios = #Semantic descriptions Use Case 1: Semantic description 1: Temperature Sensor, Camera, Microphone, Acetometer Semantic description 2: Wind Turbine interactions (e.g., start, stop, speed up, speed down etc.), Use Case 2: Semantic description of sensors/actuators on board of the Robotic Rollator: Handlebar, Inertial Measurement Unit (IMU), LiDAR, Range Sensors, Motorised Hun Wheels, Robotic Rollator (as a whole) Use Case 3: Semantic description 1: Environmental sensors (temperature, humidity, pressure), Accelerometer sensor Semantic description 2: IHES Sensing Unit reconfiguration (reset node, retrain sensors, enable/disable local analytics…)
Baseline	N/A (there were no examples for those UC defined in WoT prior to the start of the project)

3.2.2. DATA TYPE MAPPING AND ONTOLOGY ALIGNMENT

KPI-ID	KPI-2.2	**Goal**	True
Name	Data type mapping and ontology alignment		
Leader	FORTH/SAG/ENG	**Scope**	UC1, UC2
Description	Delivery of data type mapping and ontology alignment and transformation techniques that realize semantic interoperability		
Mapping to measurement points	Backend Semantic Validator. KPI satisfaction will be evaluated with a UC specific scenario including data flow which is possible between smart objects and is linked in the composition structure defined by the SPDI patterns in T4.1.		

Methodology	This KPI includes the mechanisms, which generate the necessary for semantic interoperability data transformations and the definition of the mappings between datatypes used in SEMIoTICS, supporting the validation of the semantic interoperability conditions for smart object orchestration, to ensure end-to-end semantic interoperability. KPI measurement will be based on the analysis of: • the definition of data mappings, data transformation techniques and validation mechanisms to ensure end-to-end semantic interoperability in across the SEMIoTICS's layers (field, backend) as detailed in D4.4 • the translation of Recipes into SPDI Patterns and the definition of semantic annotations for the SPDI patterns (D4.11).
Baseline	N/A (there is no such mechanism defined prior to the start of the project)

3.2.3. SEMANTIC INTEROPERABILITY WITH 3 IOT PLATFORMS

KPI-ID	KPI-2.3	**Goal**	3
Name	Semantic interoperability with 3 IoT platforms		
Leader	FORTH/STS	**Scope**	UC1, UC2, UC3
Description	Validated semantic interoperability between the SEMIoTICS framework and IoT platforms		
Mapping to measurement points	The setup includes VMs that host the Backend Semantic Validator (BSV), Thing Directory for the SEMIoTICS framework, Context Broker for the FIWARE framework, MindSphere Asset for the MindSphere framework and OpenHAB for the OpenHAB platform. The BSV is responsible for handling the interoperability issues between the said IoT frameworks by leveraging information available from the components of the respective frameworks.		

Methodology	Semantic interoperability between the SEMIoTICS framework (IoT Gateway, Backend) and IoT platforms needs to be provided. For example, a device that is registered with IoT Gateway should be interoperable with a software artefact by an IoT platform. Use Case 1: For each field device that is registered by IoT Gateway there will be created a semantic description (W3C Thing Description). Further on, there will be created a representation of each device in the IoT platform, i.e., a MindSphere representation for each device in the form of so called MindSphere Asset model will be provided. After this, it can be tested whether certain MindSphere Asset delivers data from a device that is represented by that Asset. Use Case 2: For each field device that is registered by IoT Gateway there will be created a semantic description (W3C Thing Description). Further on, there will be created a representation of each device in the IoT platform, i.e., a context entities based on Orion Context Broker of the FIWARE framework. After this, it can be tested whether certain FIWARE entities deliver data from a device that is represented by that entity Use Case 3: For each field device that is registered by the IoT Gateway there will be created a semantic description (W3C Thing Description). Further on, there will be created a representation of each device in the IoT platform, i.e., Things based on the OpenHAB platform in combination with the respective Items and Bindings. After this, it can be tested whether certain OpenHAB Things deliver data from a device that is represented by that Thing.
Baseline	N/A (there is no such mechanism defined prior to the start of the project)

3.3. Objective 3 – Monitoring Mechanisms

The main scope of objective 3 is focused on the development of dynamically and self-adaptable monitoring mechanisms supporting integrated and predictive monitoring of smart objects of all layers of the IoT implementation stack in a scalable manner. The main description of this objective defines that monitoring the operation of smart objects and IoT applications at runtime is necessary for: (i) ascertaining that conditions, which are necessary for the preservation of the SPDI properties required of them are preserved, and (ii) maintaining an awareness of their operational context that can aid the selection of appropriate adaptation actions for them when the need arises. Existing IoT application enabling platforms offer comprehensive monitoring capabilities. However, these capabilities offer either standard built-in checks or platform specific languages for defining different checks of specific types (e.g., intrusion or performance checks). Hence, checks realizable in one IoT platform are difficult to be ported to other platforms when dynamic adaptations in the smart objects and structures of IoT applications (e.g., addition and departure of smart devices) occur. Furthermore, existing IoT platforms offer limited forms of predictive monitoring. *SEMIoTICS* will develop support for seamless, extensible and adaptive monitoring **(WP4)**. This will be through the development of a *monitoring management layer* for: (a) instantiating the parametric monitoring conditions of SPDI patterns into concrete monitoring conditions regarding the particular smart objects that instantiate a pattern; (b) checking the monitorability of such conditions across different IoT platforms and creating optimal monitoring strategies for this purpose; (c) configuring automatically the monitors of IoT enabling platforms as required for different monitoring strategies; (d) fusing the results of different monitors (possibly in different platforms) as necessary for the checks; and (e) seamlessly adapting the monitoring strategies and monitoring configurations of different IoT enabling platforms following changes in IoT applications and smart objects to enable continuous uninterrupted monitoring. The monitoring management layer of *SEMIoTICS* will be able to support monitoring of smart objects at all the different layers of the IoT implementation stack. Furthermore, *SEMIoTICS* will develop support for predictive monitoring.

3.3.1. DELIVERY OF A MONITORING MANAGEMENT LAYER

KPI-3.1 aims to deliver of a monitoring management layer for: (a) generating monitoring strategies for different checks and configurations of monitors available in the 3 targeted IoT platform, (b) fusing results of these 3 IoT platform monitors, and (c) performing predictive monitoring with an aimed accuracy of 80% on average. Therefore, three different sub-KPIs are required the above KPI.

3.3.1.1. GENERATING MONITORING STRATEGIES IN THE 3 TARGETED IOT PLATFORMS

KPI-ID	KPI-3.1.1	**Goal**	3
Name	Generating monitoring strategies in the 3 targeted IoT platforms		
Leader	ENG	**Scope**	UC1, UC2, UC3
Description	Delivery of a monitoring management layer for generating monitoring strategies for different checks and configurations of monitors available in the targeted IoT platforms (3 platforms above)		
Mapping to measurement points	This KPI aims to measure the ability of the Monitoring Component to consume the events generated by the smart objects used by the Use Cases.		
Methodology	This KPI will be achieved if the measure of the **Monitorable Object Type Coverage (MOC)** of the monitoring component is **greater than 0.6**. We define the **Monitorable Object Type Coverage (MOC)** as: MOC = #MOT / #UOT where: • #UOT : (Use-case Object Types) : number of (smart) object types deployed within use cases • #MOT : (Monitored Object Types) : number of (smart) object types deployed within a use case that can be monitored by the Monitoring Component. The value of XFC will be measured as follows: • at the start of tasks T5.4-T5.6 each use case will define the type of smart objects relevant for that use case. The size of the collection of all smart object types defined by the use cases represents the value of the term UOT. • at the end of tasks T5.4-T5.6 each use case will report how many smart object types were possible to monitor using the monitoring component. The size of the collection of all object types actually monitored by the monitoring component represents the value of the term MOT.		
Baseline	0.6 (This value is the result of a discussion conducted between experts in this field)		

3.3.1.2. FUSE RESULTS FROM THESE MONITORS

KPI-ID	KPI-3.1.2	**Goal**	3
Name	Fuse results from these monitors		
Leader	ENG	**Scope**	Framework applied in UC1, UC2, UC3
Description	Delivery of a monitoring management layer for fusing results of these 3 IoT platform monitors		

Mapping to measurement points	This KPI aims to measure the ability of the Monitoring Component to fuse events generated by different IoT platforms (i.e. to process queries defining patterns of events generated by different IoT platforms).
Methodology	This KPI will be achieved if the measure of the **Cross-platform events fusion capability (XFC)** of the monitoring component is **greater than 0.6**. We define the **Cross-platform events fusion capability (XFC)** as: XCF = #MXQ / #XQ where: • #XQ : (Cross-platform Queries) : total number of **cross-platform queries** defined by all Use Cases. • #MXQ : (Monitored Cross-platform Queries) : total number of cross-platform queries defined by Use Cases that the Monitoring Component is able to process. The value of XFC will be measured as follows: • at the start of tasks T5.4-T5.6 each use case will define the cross-platform queries relevant for that use case. The size of the collection of all cross-platform queries defined by the use cases represents the value of the term XQ. • at the end of tasks T5.4-T5.6 each use case will report how many cross-platform queries were possible to implement as queries for the monitoring component. The size of the collection of all cross-platform queries actually implemented as queries for the monitoring component represents the value of the term MXQ.
Baseline	0.6 (This value is the result of a discussion conducted between experts in this field)

3.3.1.3. PERFORMING PREDICTIVE MONITORING WITH AN AVERAGE ACCURACY OF 80%

KPI-ID	KPI-3.1.3	**Goal**	80%
Name	Performing predictive monitoring with an average accuracy of 80%		
Leader	FORTH/ENG	**Scope**	Framework applied in UC1, UC2, UC3
Description	Delivery of a monitoring management layer for performing predictive monitoring with an aimed accuracy of 80% on average.		
Mapping to measurement points	This KPI aims to measure the ability of the Monitoring Component to perform Monitoring Tasks in compliance with the QoS requested by client applications. This entails the ability to predict possible failures of components needed by a monitoring task and, consequently, to adapt the monitoring infrastructure to deal with the forecasted failures.		

Methodology	This KPI will be achieved if the measure of the **Compliance to QoS (QoSC)** of the monitoring component is **greater than 0.8 in each use case**. We define the **Compliance to QoS (QoSC)** as: QoSC = #CED / # ED where: • #ED : (Event Detections) : number of event patterns (High Level Events) matched by the monitoring component. • #CED : (Compliant Event Detections) : number of event patterns (High Level Events) matched by the monitoring component in compliance with the QoS associated with the query defining that pattern. The value of XFC will be measured as follows: • each use case will use the logs of the monitoring components to count the number of matching detected by the monitoring components. This number represents the value of the term ED. • each use case will use the logs of the monitoring components to count the number of events patterns matched by the monitoring components in compliance of the QoS prescribed by the query defining that pattern. This number represents the value of the term CED.
Baseline	0.8 (This value is the result of a discussion conducted between experts in this field)

3.3.2. DELIVERY OF A MONITORING LANGUAGE

KPI-ID	KPI-3.2	**Goal**	True
Name	Delivery of a monitoring language		
Leader	ENG/STS/FORTH	**Scope**	Framework applied to UC1, UC2, UC3
Description	Delivery of a generic monitoring language capable of defining platform agnostic monitoring conditions (as part of SPDI patterns), correlations of different IoT platform events that are necessary for this, and predictive monitoring checks		
Mapping to measurement points	This KPI aims to measure the ability of the monitoring language to describe the monitoring tasks needed by SPDI Pattern enforcing components (e.g. Pattern Engine). Monitoring language of the Monitoring Component (First version in D4.2, final in D4.9).		

Methodology	This KPI will be achieved if the measure of the **Monitoring Language Expressiveness (MLE)** of the monitoring component is **greater than 0.5**. We define the **Monitoring Language Expressiveness (MLE)** as: MLE = #DPMT / #PMT where: • #PMT : (Patterns Monitoring Tasks) : number of monitoring tasks defined by SPDI patterns. • #DPMT : (Delegated Patterns Monitoring Tasks) : number of monitoring task delegated to Monitoring Component by the SPDI Pattern - aware components (e.g. Pattern Engine). The value of MLE will be measured as follows: • Let UCP denote the set of the SPDI architectural patterns used by at least one use case. • Let MP the set monitoring policies entailed by at least one pattern from UCP. The cardinality of the MP represents the value of term #PMT. • Let DMT the set of monitoring policies from MP that can be translated in a query for the Monitoring Component. The cardinality of DMT is the value of the term #DMPT.
Baseline	0.5 (This value is the result of a discussion conducted between experts in this field)

3.4. Objective 4 – Multi-layered Embedded Intelligence

The main scope of Objective 4 is the development of core mechanisms for multi-layered embedded intelligence, IoT application adaptation, learning and evolution, and end-to-end security, privacy, accountability and user control. The main description of this objective defines that SEMIoTICS will develop specialized, lightweight algorithms for intelligent analysis to enable local semi-autonomous operation, tailored to the resources and constraints of field-level smart objects **(WP4)**. It will also develop mechanisms to fuse local intelligence for enhanced intelligent behaviour at higher layers **(WP4)**. To address adaptation, SEMIoTICS will develop mechanisms supporting horizontal (cross IoT platform) and vertical adaptation (within IoT platform) actions. These will, for example, include changes of communication configurations, changes of smart objects and the compositional structures of IoT applications, and/or compensation actions **(WP4, WP3)**. The operation of SEMIoTICS adaptation mechanisms will be driven by SPDI patterns: patterns will include parametric actions that can be applied to realize specific types of adaptation and that can also be applied in a synthetic manner to generate new smart object orchestrations [6] [16]. The adaptation mechanisms will be in- formed by monitoring and intelligence analytics **(WP4)**, e.g. analysis of information regarding the effectiveness of past adaptations related to different SPDI patterns and IoT applications, and the use them to make more effective selection of adaptation actions in the future (learning and evolution). We will also develop end-to-end authentication and context-aware, distributed authorization mechanisms and tamper-proof, distributed logging of “thing events” **(WP4)**. Logging, along with the monitoring infrastructure and analytics will provide the basis for account- ability. SEMIoTICS will provide APIs to enable IoT applications to establish and adapt, at runtime, user controls **(WP4)**. These controls will be over data production, access, processing, and storage, subject to appropriate user authorization rights. Ensuring security in IoT applications that interconnect with a spectrum of smart devices and sensors with varying and low level computational and energy capabilities requires the development of new security solutions for these objects (e.g., security controllers enabled by appropriate software). The development of security solutions for such objects will be addressed in SEMIoTICS.

3.4.1. DELIVERY OF LIGHTWEIGHT ML ALGORITHMS

KPI-ID	KPI-4.1	**Goal**	3
Name	Delivery of lightweight ML algorithms		

Leader	ST	**Scope**	UC3
Description	Delivery of at least 3 lightweight ML algorithms to enable semi-autonomic behavior on resource-constrained smart devices.		
Mapping to measurement points	Local Embedded Analytics: ST-I deploy for the Local Embedded Analytics in ST-I FW these new algorithms: • An online training for neural model (ESN). • An online predictor using (online) trained neural model (ESN). • A model-free change detection (CDT) test applied to residual signal. • A change-point method to (CPM) validate the detected change. For more detailed information's please see D4.3 (in particular sections 2.3.3 and 3.4.1)		
Methodology	Component release (task 4.3) and integration (task 5.6) of the Local Embedded Analytics component in the SEMIoTICS framework.		
Baseline	3 Machine Learning Algorithms delivered in SEMIoTICS		

3.4.2. DELIVERY OF MECHANISMS WITH ADAPTATION TIME OF 15MS

KPI-ID	KPI-4.2	**Goal**	15ms
Name	Delivery of mechanisms with adaptation time of 15ms		
Leader	ENG/FORTH	**Scope**	All
Description	Delivery of adaptation mechanisms that support proactive and reactive, as well as horizontal and vertical adaptation actions, related to network, smart objects and IoT platforms with an adaptation time of 15ms.		
Mapping to measurement points	Cross-layer pattern engines, monitoring component, embedded intelligence. When a failure has been identified, a list of available adaptations mechanisms is derived from the use of monitoring and embedded intelligence while complying with the restriction of 15ms. The setup includes VMs that host the pattern related components. The pattern rules will act as the adaptation mechanisms that will be enforced. The time will be measured using timestamps that will be attached to appropriate events.		
Methodology	This KPI aims to offer proactive adaptations as well as reactive by means of monitoring and intelligence analytics. In order to accomplish this, adaptation's start of measured mechanism begins from the time that a failure has been detected until the time that it was required to restore the said failure. When a failure has been identified, a list of available adaptations mechanisms is derived from the use of monitoring and embedded intelligence while complying with the restriction of 15ms.		
Baseline	15 ms		

3.4.3. DELIVERY OF ADAPTATIONS MECHANISMS ENABLING IMPROVEMENT BY AT LEAST 20%

KPI-ID	KPI-4.3	**Goal**	20%
Name	Delivery of adaptations mechanisms enabling improvement by at least 20%		

Leader	FORTH	**Scope**	UC2, UC3
Description	Delivery of evolution mechanisms enabling the detection and analysis of the effects of past adaptations the improvement of future adaptations by at least 20%		
Mapping to measurement points	Cross-layer pattern engines, monitoring component, embedded intelligence. The setup includes VMs that host the pattern, monitoring and embedded intelligence related components. The pattern rules will act as the adaptation mechanisms in combination with the embedded intelligence that will be enforced. The time will be measured using timestamps that will be attached to appropriate events.		
Methodology	This KPI aims to improve past adaptations by means of monitoring and intelligence analytics. In order to accomplish this, the existing adaptations will be measured beginning from the time that a failure has been detected until the time that it was required to restore the above failure. The methodology will distinguish two scenarios. • The first scenario is that, after a failure, there is no adaptation; this is the behavior prior to SEMIoTICS. • In the second scenario there is an existing adaptation mechanism, which SEMIoTICS tries to improve upon. In the first scenario any future adaptation mechanism provided by SEMIoTICS will automatically imply 100% improvement given the fact that prior to SEMIoTICS there was no adaptation at all. In the second scenario the improvement of the new mechanisms, will be measured by using appropriate metrics based on the use cases.		
Baseline	Time needed for adapting to a detected failure prior to SEMIoTICS.		

3.4.4. DETECTION TIME OF LESS THAN 10MS

KPI-ID	KPI-4.4	**Goal**	10ms
Name	Detection time of less than 10 ms		
Leader	ST	**Scope**	UC3
Description	Detection time for the respective detection mechanisms should be below 10ms.		
Mapping to measurement points	Component release (task 4.3) and integration (task 5.6) of the Local Embedded Analytics component in the SEMIoTICS framework		
Methodology	ST-I defined the Detection Time as the time in which Local Embedded Analytics in ST-I FW perform all the needed phases to identify if the input data could be a change or not. For this experiment ST-I count Detection Time on the ST Nucleo F401 board using ST-I FW with an ESN model.		
Baseline	10ms		

3.4.5. BASELINE IMPROVEMENT OF 20% ADAPTATION TIME

KPI-ID	KPI-4.5	**Goal**	20%
Name	Baseline improvement of 20% adaptation time		
Leader	ST	**Scope**	UC3

Description	The adaptation response time should bring at least a 20% improvement over the baseline of each domain.
Mapping to measurement points	Component release (task 4.3) and integration (task 5.6) of the Local Embedded Analytics component in the SEMIoTICS framework

Methodology	ST-I defined the Minimum Adaptation Time as the time in which a model can make the inference process on the input data. For this experiment ST-I will use:
	• Neural model 1:
	  ○ type: Autoencoder
	  ○ neurons: 18 (for the encoder)
	  ○ inputs: 50 (this is set as observation window size since autoencoder is a non-recursive neural network)
	• Neural model 2:
	  ○ type: Echo State Network
	  ○ neurons: 35 (for the reservoir)
	  ○ inputs: 3 (tri-axial accelerometer at each instant at the input)
	
	For this experiment ST-I will compare the two models on the ST Nucleo F401 board counting the time from the input data to the output predictions.

Baseline	Neural model 1 (Autoencoder) adaptation time.

3.4.6. DEVELOPMENT OF NEW SECURITY MECHANISMS/CONTROLS

KPI-ID	KPI-4.6	**Goal**	3
Name	Development of new security mechanisms/controls		
Leader	UP/FORTH/STS/CTTC	**Scope**	UC1, UC2
Description	Development of a minimum of 3 new security mechanisms/controls enabling the secure management of smart devices and sensors over programmable industrial networks.		
Mapping to measurement points	Pattern Engine (Backend, Network, Field), Security Managers (Backend, Network, Field), SFC Manager + Security Service Functions. The setup includes VMs that host the pattern, security managers and SFC related components. Also, smart objects which are necessary for the use case scenarios will be part of the setup in order to implement their management and control with additional security mechanisms that were not attainable on their own, due to their limitation to computational and energy resources.		
Methodology	The communication for the management of the smart devices and sensors will be extended with security mechanisms. These extensions will be in the form of forcing the communication to be controlled by the Security Manager as well as creating appropriate paths derived from the Service Function Chains that will be composed by Security Service Functions. This can be checked by providing the implementations well as the results from successful integration of the execution of the mechanisms in at least a lab test or in one of the use cases of the project.		
Baseline	N/A (simply the implemented security controls are being counted)		

3.5. Objective 5 – IoT-aware Programmable Networks

The main scope of Objective 5 is the development of IoT-aware programmable networking capabilities based on adaptation and SDN orchestration. A more detailed description of this objective defines that IoT creates significant challenges for networking and it becomes necessary to achieve significant enhancements in network scalability, adaptability and security. These challenges arise from the plethora of heterogeneous smart objects and devices that will be connected to IoT applications and the sheer volume and heterogeneity of the content of traffic that they create. In recognition of this need, SEMIoTICS has as one of its key objectives the provision of reliable, intelligent, self-managed, robust and context-aware networking for smart objects and IoT applications. To address this objective, we will adopt software defined networking (SDN) and network function virtualization (NFV) technologies (WP3) to construct trusted routing overlays using semantic information on the fly, and to enable the simultaneous usage of smart objects from several applications and service providers. A key element in the development of SDN in SEMIoTICS is that it can be configured by network configuration specifications embedded in SPDI patterns, enabled by the associated SDN Controller modules of the OpenDayLight platform (https://www.opendaylight.org). These configurations reflect end-to-end network optimizations based on current operational conditions (as detected through the SEMIoTICS monitoring mechanisms) and is activated when necessary through the SEMIoTICS framework.

3.5.1. DEPLOYMENT OF A MULTI-DOMAIN SDN ORCHESTRATOR

KPI-ID	KPI-5.1	**Goal**	TRL 5
Name	Deployment of a multi-domain SDN orchestrator		
Leader	FORTH /SAG	**Scope**	UC2
Description	Deployment of a multi-domain SDN orchestrator, capable of operating based on SDN configurations requested by IoT applications, and accessible to the SEMIoTICS framework through an API of TRL 5. The SDN Orchestrator will demonstrate network orchestration in networks.		
Mapping to measurement points	Multi-domain orchestration in Use case 2. The setup includes VMs that host Use case 2 related components. SDN controller in combination with VNFs will guarantee the TRL5 level achievement.		
Methodology	This KPI aims to deploy orchestration in networks. KPI measurement is directly associated with the Use case 2 where NFV network capabilities will be utilized.		
Baseline	Not applicable		

3.5.2. SERVICE FUNCTION CHAINING (SFC) OF A MINIMUM 3 VNFS

KPI-ID	KPI-5.2	**Goal**	3
Name	Service Function Chaining (SFC) of a minimum 3 VNFs		
Leader	FORTH/CTTC	**Scope**	UC2
Description	Service Function Chaining (SFC) of a minimum 3 VNFs for smart object data paths. The setup includes VMs that host the SFC and VNF related components. In addition, pattern related components will be utilized for the instantiation of the SFCs and VNFs in real time.		
Mapping to measurement points	SFC manager + VNFs. The setup includes VMs that host the SFC and VNF related components. In addition, pattern related components will be utilized for the instantiation of the SFCs and VNFs in real time.		

Methodology	This KPI aims to provide enhanced, reactive security services for adapting to events and maintaining the networks' security posture. KPI measurement is based on forwarding the traffic to 3 VNFs (see D3.1, D3.2, D5.5)
Baseline	Not applicable

3.6. Objective 6 – Development of a Reference Prototype

The main scope of this objective is to development of a reference prototype of the SEMIoTICS open architecture, demonstrated and evaluated in both IIoT (renewable energy) and IoT (healthcare), as well as in a horizontal use case bridging the two landscapes (smart sensing), and delivery of the respective open API. A more detailed description of this objective defines that SEMIoTICS will develop and deliver an open source, reference proof-of-concept implementation of the SEMIoTICS framework, integrating the core interoperability, monitoring, intelligence, adaptation, and networking capabilities outlined in Objectives 2-5 **(WP5)**. This infrastructure will provide its functionalities driven by the selection, instantiation, and execution of SEMIoTICS patterns. The infrastructure will be interoperable with (a) the generic open source IoT platform FIWARE, (b) partner IoT platforms and middleware, such as MindSphere, AREAS H-ERP. It will also support generic (e.g. MQTT, http://mqtt.org/) and domain specific protocols (e.g. Healthcare Device Profile), and other standards regarding semantic annotation, security and privacy for IoT applications. The functionalities of the SEMIoTICS framework will be offered through open APIs.
To ensure the technical soundness and industrial applicability of the SEMIoTICS approach and infrastructure, the R&D carried out within SEMIoTICS will be driven by requirements of IoT application scenarios in domains that have already been making use of IoT technology and provide clear scope for further significant uptake of this technology. These domains are renewable energy (**scenario 1**); healthcare (**scenario 2**); and smart sensing (**scenario 3**). The scenarios that we are targeting within these domains have been selected because they use different IoT enabling platforms, types of smart objects, devices and types of networks. They also cover a significant spectrum of different SDPI and performance requirements, thus enabling a comprehensive evaluation of SEMIoTICS framework. This evaluation will focus on investigating: (a) the overall effectiveness of the SEMIoTICS's approach, infrastructure in addressing interoperability, scalability and performance requirements as well as trade-offs between them; and (b) the merit and implications of using SEMIoTICS from a business perspective

3.6.1. REDUCE REQUIRED MANUAL INTERVENTIONS

KPI-ID	KPI-6.1	**Goal**	80%
Name	Reduce manual interventions required for bootstrapping of smart object in each use case domain by at least 80%		
Leader	SAG/CTTC	**Scope**	UC1, UC3
Description	Reduce manual interventions required for bootstrapping of smart object in each use case domain by at least 80%.		
Mapping to measurement points	Recipe Cooker, NFV		

Methodology	This KPI aims to show the effectiveness of our semantic-based bootstrapping approach in comparison to state of the practice. The goal is to demonstrate that our approach makes the process of integration of field devices easier. This includes also the configuration of devices, provision of semantic meta-data of devices, device discovery, and creation of new added-value applications with newly bootstrapped devices. Furthermore, it may also include the orchestration of resources as VNFs (e.g. computation agents, MQTT brokers, databases, etc.), accelerating the service bootstrapping process. #effort_1: Estimate effort of integration of a field device with state of the practice approach. #effort_2: Estimate effort of integration of a field device with our SEMIoTICS plug&play approach. #effort_2 << #effort_1 ("<<" means "significantly smaller than". The percentage of 80 looks too exact amount that is difficult to proof). We will estimate effort_1 and effort_2. Then we will compare the two efforts and produce the percentage that estimates manual reduction during the bootstrapping process. Our estimation will be based on concrete measurements for a device, e.g., a bootstrapped camera. But the produced measurements will be still estimations as we cannot proof that any device can be bootstrapped for the same amount of time. Especially for use case 3, the reduction on manual interventions will be reflected in scaling operations. That is, when a specific VNF is overloaded with tasks, the NFV Orchestrator may automatically instruct the scaling out of such component automatically. Therefore, such operations are expected to eliminate user intervention completely.
Baseline	N/A (the baseline is the state-of-practice as provided within the methodology)

3.6.2. LEVERAGING UPON FIWARE ASSETS

KPI-ID	KPI-6.2	**Goal**	10
Name	Leveraging upon FIWARE assets in developing the SEMIoTICS framework		
Leader	BS/ENG	**Scope**	UC2
Description	Checking the possibility of using ready-made FIWARE Generic Enablers in the SEMIoTICS platform to reduce the time needed in the development process.		
Mapping to measurement points	Analysis will be performed for a generic enablers that are pertinent to SEMIoTICS, of various types: IoT Services Enablement ("Backend Device Management-IDAS"),Data/Context Management("Publish/Subscribe Context Broker-Orion", "BigData Analysis-Cosmos", "FIWARE CKAN Extensions", "Stream-oriented-Kurento"), Security ("Identity Management-KeyRock", "Authorization PDP-AuthZForce", "PEP Proxy-Wilma"), Architecture of Applications/ Services Ecosystem and Delivery Framework ("Application Mashup-Wirecloud", "Business API Ecosystem-Biz", "Data Visualization – Knowage" for Business Intelligence and Big Data Analytics, realized and supported by the SEMIoTICS partner ENG). KPI measurement will be based on the analysis results and further testing of those FIWARE assets which will provide the satisfactory results. Setup, testing and further integration within the SEMIoTICS framework will be performed.		

Methodology	Within this KPI there will be analysis performed of FIWARE assets capabilities and adoption possibilities. The analysis will focus on the technical scope of FIWARE assets, component maturity and verification whether the FIWARE assets are capable to fulfil SEMIoTICS aims and needs. KPI satisfaction will be evaluated with a locally deployed application working following the provided documentation.
Baseline	N/A (the number of analysed assets will be simply counted)

3.6.3. DELIVERY OF 3 PROTOTYPES OF IIOT/IOT APPLICATIONS

KPI-ID	KPI-6.3	**Goal**	3
Name	Delivery of 3 prototypes of IIoT/IoT applications		
Leader	SAG/ENG/IQU	**Scope**	UC1, UC2, UC3
Description	Delivery of 3 prototypes of IIoT/IoT applications that will drive the demonstration of the respective usage scenarios and evaluation of *SEMIoTICS* approach and platform based on the 3 developed applications, covering technological and business aspects.		
Mapping to measurement points	Use Case 1: • App UC1.d: Visualization of field level data in MindSphere • App UC1.a: Oil leakage detection for wind turbine bearings • App UC1.b: Recording of wind turbine inclination over time • App UC1.c: Alarm for detection of unusual noise in turbine Use Case 2: • App UC2: Prototype of the SARA healthcare solution Use Case 3: • App UC3: Smart sensing application		

Methodology	Use Case 1: Overall, 3 applications (apps) will be prototypically developed within Use Case 1. All of these apps will be developed based on the SEMIoTICS framework, utilizing the Recipe Cooker to define the respective app, the Pattern Orchestrator to determine the deployment, and the Pattern Engine to correctly configure the network and switches. • UC1.a will use multiple nodes such as an infrared camera and an AI-based imagery classifier, interlinked via a recipe and deployed on multiple devices networked via SDN. • UC1.b will use an inclinometer inside the wind turbine and the sensed measurements will be uploaded through the SDN to a Cloud platform (Siemens MindSphere) where the data will be recorded and available for analytics. • UC1.c will utilize a microphone installed inside the turbine. The audio stream will be analyzed by a dedicated node to derive a decibel value. A logic will determine whether this value exceeds a certain threshold, which would indicate an incident such as a loosened screw tumbling inside the rotating blade. Use Case 2: The UC2 will deliver a prototype of the SARA healthcare solution taking advantage of the SEMIoTICS technologies. More specifically the SARA prototype will embed SEMIoTICS technologies concerning: • security i.e. Pattern Orchestrator, Pattern Engine, Security Manager, Monitoring. • semantic interoperability i.e. Semantic API & Protocol Biding, GW Semantic Mediator, Thing directory, Backend Semantic Validator. • embedded intelligence i.e. Local embedded intelligence • network management i.e. SDN Controller. Use Case 3: The verification of the UC3 prototype will be done through i) the correct operation of the UC3 smart sensing application at the respective SEMIoTICS platform as defined by the requirements specifications of the UC3 description, and ii) the performance evaluation of the UC3 smart sensing application that ensures the KPI satisfaction of the involved SEMIoTICS components.
Baseline	N/A (the number of prototypes will be counted)

3.7. Objective 7 – Promote the Adoption of EU Technology Offerings Internationally

The main scope of Objective 7 is the promotion of adoption of EU technology offerings international. A more detailed description of this objective defines to achieve this objective is that SEMIoTICS will: (i) Promote EU technologies and IoT platforms (such as FIWARE, MindSphere) in the international IoT landscape, offering novel technologies and tools with increased usability and user acceptance (notably through strengthened security and user control) that will foster industry in- novation and create a path to new, competitive products and services **(WP3, WP4)**; (ii) Promote European influence in standardization and pre-normative activities internationally, aligning with relevant European activities and projects and collaborating with Standardizations bodies **(WP6)** and (iii) Promote European research leadership, broadly and effectively communicating and disseminating the SEMIoTICS results. **(WP6)**.

3.7.1. PROVISION THE SEMIOTICS FRAMEWORK AND BUILDING BLOCKS

KPI-ID	KPI-7.1	**Goal**	TRL 5 or 6

Name	Provision the SEMIoTICS building blocks		
Leader	All (SAG)	**Scope**	Project
Description	This KPI will provide the key technological building blocks at TRL 5 or 6. The SEMIoTICS framework must also be demonstrated as whole or as a partial distribution of the overall framework in appropriate lab Use Cases, corresponding to TRL 4.		
Mapping to measurement points	SEMIoTICS framework as a set of inter-connected components presented the first draft architecture in D2.4 and the final architecture in D2.5. SEMIoTICS building blocks are synonymous with SEMIoTICS components and are defined, refined and implemented in WP3 and WP4. The components and the framework are validated and evaluated and inter-connected for the purpose of Use Case realizations in WP5.		
Methodology	The individual components of the framework will be validated and evaluated in the industrially relevant environment. That is, they will be deployed in, or will interact with devices, that correspond to equipment hosted in operational environment. Thus, the components will be validated for TRL 5 / 6. Semiotics Use Cases 1-3 will demonstrate the interactions among the components in a controlled lab environment, thus directly supporting the claim of TRL 4. This is as discussed in D2.2. We will confirm the reaching of TRL 4 for the overall framework by demonstrating the same framework in Use Case 1-3 scenarios. Note that particular components may be left out in demonstration of individual use cases due to non-dependency on those particular modules. Nevertheless, they all start from the same base SEMIoTICS framework and simply exclude the unused components as per use case goal.		

Baseline	**Component Layer Baseline TRL Target TRL** Backend orchestrator Application Orchestration N/A (new) 6 Backend Semantic Validator Application Orchestration N/A (new) 4 Bootstrapping Manager SDN Orchestration 6 (w/o adaptations) 6 (w/ adaptations) Clustering Manager SDN Orchestration 6 (w/o adaptations) 6 (w/ adaptations) GUI Application Orchestration N/A (new) 6 GW Semantic Mediator Field N/A (new) 5 Local embedded intelligence Field 2 4 Local thing directory Field 6 (w/o adaptations) 6 Monitoring Field N/A (new) 4 Monitoring Application Orchestration N/A (new) 4 NFV Orchestrator NFV Orchestration 6 (w/o adaptations) 6 (w/ adaptations) Path Manager SDN Orchestration 6 (w/o adaptations) 6 (unchanged) Pattern Engine Field N/A (new) 4 Pattern Engine SDN Orchestration N/A (new) 4 Pattern Engine Application Orchestration N/A (new) 4 Pattern Orchestrator Application Orchestration N/A (new) 4 Recipe Cooker Application Orchestration 4 (w/o adaptations) 6 (w/ adaptations) Resource Manager SDN Orchestration 6 (w/o adaptations) 6 (w/o adaptations) Security Manager Field 4 (w/o adaptations) 5 Security Manager SDN Orchestration 6 (w/o adaptations) 6 (unchanged) Security Manager Application Orchestration 4 (w/o adaptations) 5 Semantic API & Protocol Biding Field 4 (w/o adaptations) 5 (w/ adaptations) SFC Manager SDN Orchestration 6 (w/o adaptations) 6 (w/ adaptations) Thing Directory Application Orchestration 6 (w/o adaptations) 6 (unchanged) VIM Connector SDN Orchestration 6 (w/o adaptations) 6 (unchanged) Virtualized Infrastructure Manager NFV Orchestration 6 (w/o adaptations) 6 (w/ adaptations) VNF Manager NFV Orchestration 6 (w/o adaptations) 6 (w/o adaptations) VTN Manager SDN Orchestration 6 (w/o adaptations) 6 (w/ adaptations) Semantic Edge Platform Field 4 (w/o adaptations) 5 (w/ adaptations) Supervisor and Local DB Field N/A (new) 4

3.7.2. ACHIEVE INFLUENCER STATUS WITHIN MAJOR STANDARIZATION EFFORTS

KPI-ID	KPI-7.2	**Goal**	5
Name	Achieve influencer status within major standardization efforts		
Leader	SAG/ENG/FORTH/ST/CTTC	**Scope**	Project
Description	Achieve influencer status within major IIoT/IoT standardization efforts with 5 proactive contributions to the standardization activities of ETSI, AIOTI WGs, IEEE and W3C.		
Mapping to measurement points	It will be verified through ETSI, AIOTI WGs, IEEE and W3C related websites.		

Methodology	This KPI aims to proof that certain requirements from SEMIoTICS project have contributed to IIoT/IoT standardization efforts. # number of concrete requirements from SEMIoTICS project that produced impact in IIoT/IoT standardization efforts. We promised to provide a semantic description for each smart object from all SEMIoTICS use cases. For this purpose, we will use iotschema.org. If this IoT semantic model cannot fulfil all requirements from SEMIoTICS project, then we need to imitate the standardization process. We will document all requirements from this task, which could not be fulfilled before this project and which have been created after the project.
Baseline	N/A

3.7.3. ACHIEVE THE PROJECT'S DISSEMINATION TARGETS

KPI-7.3 aims to achieve the project's dissemination targets as defined in the following table. Based on this table, a number of different sub-KPIs are analyzed bellow.

3.7.3.1. ONLINE DISSEMINATION

KPI-ID	KPI-7.3.1.1	**Goal**	≥1.000 accesses annually ≥100 downloads
Name	Project website		
Leader	FORTH	**Scope**	Project
Description	Web access to deliverables, technical results and presentation materials of SEMIoTICS		
Mapping to measurement points	Google analytics in project website		
Methodology	Google analytics internal method of counting website visits		
Baseline	Not applicable		

KPI-ID	KPI-7.3.1.2	**Goal**	≥50 announcements
Name	Push announcements		
Leader	FORTH	**Scope**	Project
Description	Regular push announcements through social media (Twitter, LinkedIn, ResearchGate)		
Mapping to measurement points	SEMIoTICS Twitter Statistics		
Methodology	Number of tweets		

Baseline	Not applicable

KPI-ID	KPI-7.3.1.3	**Goal**	≥9 newsletters
Name	Regular Newsletter		
Leader	CTTC	**Scope**	Project
Description	Regular quarterly newsletter with the technical activities of SEMIoTICS		
Mapping to measurement points	SEMIoTICS Newsletter announcements. Three newsletters have been already published, while the 4th is under preparation. The newsletters can be found in SEMIoTICS webpage: https://www.semiotics-project.eu/index.php/publications/#newsletters		
Methodology	Each newsletter includes all the project developments since the previous newsletter. In this way, all project advancements are properly disseminated. There are some deviations (between 4-8 months) in the newsletter publications, as the consortium expects to have enough material.		
Baseline	Not applicable		

KPI-ID	KPI-7.3.1.4	**Goal**	≥2.000 hard copies at ≥ 10 events ≥2.000 downloads
Name	Brochure		
Leader	FORTH	**Scope**	Project
Description	High–quality electronic brochure with the technical approach and activities of SEMIoTICS		
Mapping to measurement points	Google analytics in project website.		
Methodology	Hard copies will be made available and distributed at events related to the scope of SEMIoTICS at which SEMIoTICS partners participate. This will result in more visitors to the website in overall, as such some of the impact will be measured inseparable via KPI-7.3.1.1.		
Baseline	Not applicable		

KPI-ID	KPI-7.3.1.5	**Goal**	≥1000 views ≥ 10 event presentations
Name	Technical video		

Leader	All	**Scope**	Project
Description	5 min high-quality video presentations of the technical aspects of SEMIoTICS. Number of times presented 10		
Mapping to measurement points	Google analytics in project website and twitter analytics.		
Methodology	The video will be uploaded in the website, twitter and presented in various events. This will result in more visitors to the website overall, as such some of the impact will be measured inseparable via KPI-7.3.1.1.		
Baseline	Not applicable		

3.7.3.2. SCIENTIFIC PUBLICATIONS

KPI-ID	KPI-7.3.2.1	**Goal**	≥10 publications
Name	Journal publications		
Leader	All	**Scope**	Project
Description	Publications in International referred technical journals in IoT related subjects		
Mapping to measurement points	Uploaded in the EU portal, project website and ResearchGate,		
Methodology	Academic members of the SEMIoTICS consortium will published to international journals in IoT related subjects, the publications or publication attempts (as journals tend to have quite a long time between submission of work and final acceptance for publication) that happen within the project lifetime will be counted. SEMIoTICS partners have done so in the past, therefore proven experience exists. These partners will handle the publications.		
Baseline	Not applicable		

KPI-ID	KPI-7.3.2.2	**Goal**	≥10 publications
Name	Magazine publications		
Leader	All	**Scope**	Project
Description	Publications in International magazines in IoT related subjects		
Mapping to measurement points	Uploaded in the EU portal, project website and ResearchGate.		
Methodology	Academic members of the SEMIoTICS consortium have published to international magazines in IoT related subjects in the past, therefore proven experience exists. These partners will handle the publications. The publications or publication attempts (as magazines might have quite a long time between submission of work		

	and final appearance of the magazine) that happen within the project lifetime will be counted
Baseline	Not applicable

KPI-ID	KPI-7.3.2.3	**Goal**	≥20
Name	Conference publications		
Leader	All	**Scope**	Project
Description	Publications in International referred technical conferences in IoT related subjects		
Mapping to measurement points	Uploaded in the EU portal, project website and ResearchGate,		
Methodology	Members of the SEMIoTICS consortium has published to international technical conferences in IoT related subjects in the past, therefore proven experience exists. These partners will handle the publications. The accepted publications or publication attempts (if the submission time falls within, but the final acceptance time would be after the end of the project) that happen within the project lifetime will be counted.		
Baseline	Not applicable		

KPI-ID	KPI-7.3.2.4	**Goal**	≥4 ≥10 selected papers/ issue
Name	Special issues		
Leader	All	**Scope**	Project
Description	Preparation of special issues in international referred technical journals and magazines		
Mapping to measurement points	Announced in the ComSoc lists, project website and twitter.		
Methodology	Members of the SEMIoTICS consortium has contributed to special issues in international technical journals and magazines in the past, therefore proven experience exists. These partners will handle the preparation of the special issues. The number of special issues announced during the project's lifetime will be counted.		
Baseline	Not applicable		

3.7.3.3. ORGANIZATION OF INTERNATIONAL SCIENTIFIC EVENTS

KPI-ID	KPI-7.3.3.1	**Goal**	≥1 event ≥100 attendees (each)
Name	Conference organizations		
Leader	FORTH	**Scope**	Project
Description	Organization of International conferences in IoT and SEMIoTICS related domains		
Mapping to measurement points	Announced in the ComSoc lists, project website and twitter.		
Methodology	Members of the SEMIoTICS consortium have organized conferences in the past, therefore proven experience exists. These partners will handle the organization of the conferences. The number of announcements of conferences within the lifetime of the project will get counted, even if the actual conference might take place after the end of the project, as the organization also takes place prior to the ability to officially make an announcement.		
Baseline	Not applicable		

KPI-ID	KPI-7.3.3.2	**Goal**	3 workshops ≥50 attendees (each)
Name	Workshops		
Leader	FORTH	**Scope**	Project
Description	Organization of workshops		
Mapping to measurement points	Announced in the ComSoc lists, project website and twitter.		
Methodology	Members of the SEMIoTICS consortium has organized in the past, workshops therefore proven experience exists. These partners will handle the organization of the workshops. The number of announcements of workshops organized within the lifetime of the project will get counted; even if the actual workshop will only take place after the end of the project, as the organization also takes place prior to the ability to officially make an announcement.		
Baseline	Not applicable		

KPI-ID	KPI-7.3.3.3	**Goal**	≥2 events ≥40 attendees (each)
Name	Summer schools		
Leader	FORTH	**Scope**	Project
Description	Organization of an international summer school in IoT with at least 40 participants		

Mapping to measurement points	Announced in the mailing lists, project website and twitter.
Methodology	Members of the SEMIoTICS consortium has organized in the past, summer schools therefore proven experience exists. These partners will handle the organization of the summer schools. The number of announcements of summer schools organized within the lifetime of the project will get counted; even if the actual event will only take place after the end of the project, as a lot of effort for the organization also takes place prior to the ability to officially make an announcement.
Baseline	Not applicable

3.7.3.4. SYSTEM-LEVEL DEMONSTRATIONS

KPI-ID	KPI-7.3.4.1	**Goal**	≥1 demonstration
Name	Exhibition demonstrations		
Leader	CTTC	**Scope**	Project
Description	Demonstrations at major fairs and exhibitions such as MWC, IoTSWC		
Mapping to measurement points	SEMIoTICS was present in several major events, such as MWC 2018 and 2019, SCEWC 2017 and 2018, IoT Week 2018, IoTSWC 2017, 2018 and 2019, demonstrating building blocks of the SEMIoTICS demonstrator.		
Methodology	SEMIoTICS consortium will keep attending major fairs and exhibitions until the end of the project. The number of attendances will get counted.		
Baseline	Not applicable		

KPI-ID	KPI-7.3.4.2	**Goal**	≥2 demonstrations
Name	EU demonstrations		
Leader	CTTC	**Scope**	Project
Description	Demonstrations at major EU events such as EuCNC		
Mapping to measurement points	SEMIoTICS partners have participated in EUCNC 2018 and 2019.		
Methodology	SEMIoTICS consortium will keep attending major EU events until the end of the project. The number of demonstrations given at such events will get counted.		
Baseline	Not applicable		

KPI-ID	KPI-7.3.4.3	**Goal**	≥2 demonstrations
Name	Conference demonstrations		

Leader	SAG	**Scope**	Project
Description	Demonstrations at major conferences such as GLOBECOM, ICC		
Mapping to measurement points	Major fairs and exhibitions		
Methodology	SEMIoTICS consortium will keep attending major conferences until the end of the project. The number of demonstrations given at such events will get counted.		
Baseline	Not applicable		

4. EVALUATION METHODOLOGY

This section explains some more background on the different evaluation methodologies used in the project, i.e. field tests to get baseline measures of performance, lab-experiments, trial applications, cross-evaluation.

4.1. Baseline Performance Measures

This section describes the experiments and how, in absence of field tests the baseline for certain KPIs mentioned in the Description on Action (DoA) is gathered. It can be gathered by running experiments in controlled environments, e.g. in the case of adaptation time we will use failure injection tools (e.g. LFI[1], ChaosMachine[2], TripleAgent[3]) to simulate failures within UC applications at the different levels of the UC applications (i.e. Device, Network and Backend) and use currently available means (i.e. mainly human intervention) to recover from injected failures. Baseline performance measures for adaptation time will be taken using the means of the selected tool.

However, gathering baseline this way requires the acquisition of data over a certain period of time rather than a snapshot of information, or it requires access to critical infrastructure or real-life data, e.g. medical information. Thus, the limited timeframe and resources or privacy-related concerns do not always allow the project to perform such extended data gathering campaigns for all KPIs. More often we do not even need to establish a good baseline from scratch, as the partner's expertise in the vertical use cases allow to establish this baseline or other material, such as industry standards or state of the art exists. Therefore, for these areas the project will rely on the background knowledge of use case owners. This background can be either measures already taken from operations of systems developed using current standard technologies like in the case of wind parks, or, acceptance criteria by potential users, customer or other prototypes already developed without the use of SEMIoTICS technologies like in the case of SARA UC2.

4.2. Controlled Lab-Based Experiments

Next, the methodology, to evaluate the controlled lab-based experiments, is presented. For illustration purposes, the evaluation methodology related to controlled-lab based experiments in NFV is exposed.

In general, the aim of any experiment or test is to validate a target a.k.a. System Under Test (SUT) in [ETSI GS NFV-TST 001][4]. Moreover, the tests to validate a SUT are executed within the context of a test environment. A SUT, in turn, is divided in one or more Functions Under Test (FUT) a.k.a. Devices Under Test (DUT).

In the NFV example, different SUT can be considered within the context of the NFV platform. Namely, examples of SUT in the NFV context are:

- The NFV Infrastructure (NFVI),
- Virtualised Network Functions (VNF),
- Network Services (NS).

[1] https://github.com/dslab-epfl/lfi
[2] https://github.com/KTH/royal-chaos/tree/master/chaosmachine
[3] https://github.com/KTH/royal-chaos/tree/master/tripleagent
[4] [ETSI GS NFV-TST 001] *ETSI "ETSI GS NFV-TST 001: Network Functions Virtualization (NFV); Pre-deployment Testing; Report on Validation of NFV Environments and Services" April 2016.* Retrieved from https://www.etsi.org/deliver/etsi_gs/NFV-TST/001_099/001/01.01.01_60/gs_NFV-TST001v010101p.pdf

The test environment is specified as follows. It consists of the SUT to be tested and reference implementations of the rest of the functional blocks, e.g. the rest of the blocks of the NFV platform. The test environment also contains functional blocks that control the test execution, and which collect the test measurements. This generic description of the test environment is illustrated in Figure 1.

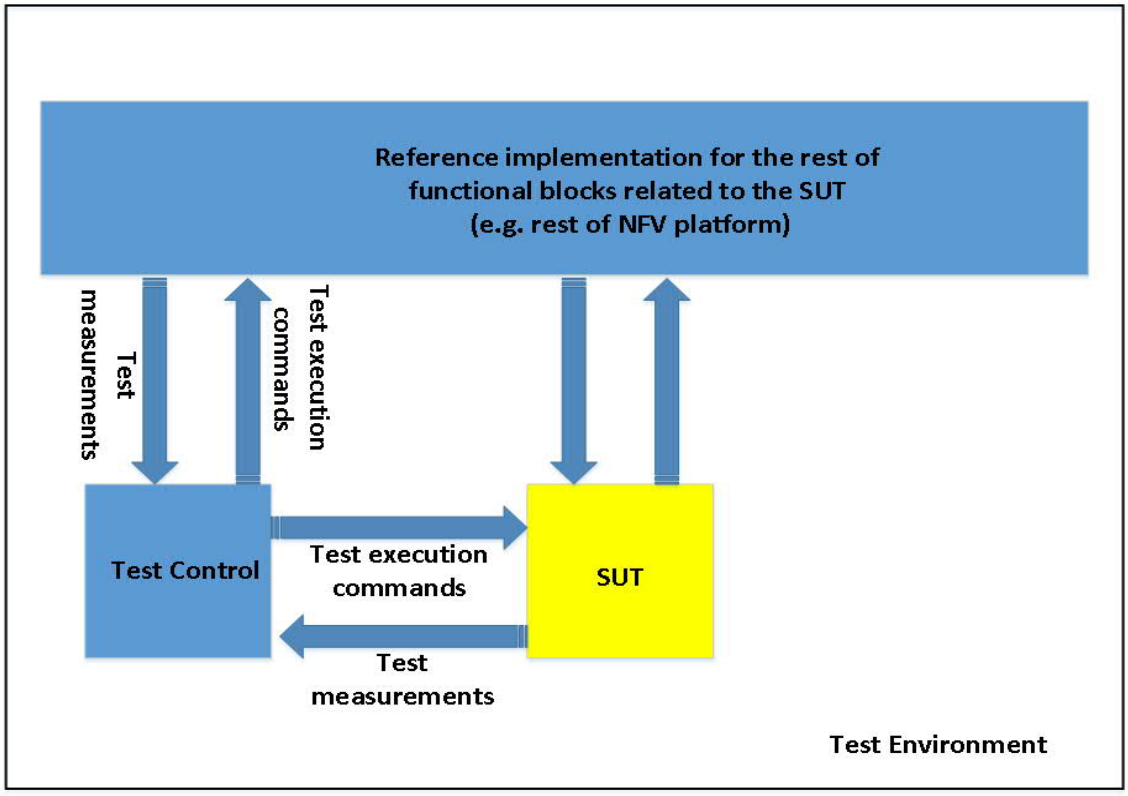

FIGURE 1: GENERIC TEST ENVIRONMENT DESCRIPTION

Figure 2 particularizes the test environment just described above for the particular case where the SUT is the NFVI. As it can be observed in Figure 2 the NFVI consists of several subsystems, which can be considered as FUT, according to the description given above. In order to test the FUT of the NFVI, the methodology is as follows:

- The test control leverages a set of test functions to test each of the FUT within the NFVI.
- The test functions request the NFV MANO[5] to instantiate a set of reference VNFs on top of the NFVI. These VNFs embed the functionality to cover all the aspects needed to test the FUT, e.g. to test the Virtual Compute FUT, one needs a VNF that stresses the virtual compute resources associated to the virtual machine where the VNF is instantiated.
- The test measurements are gathered and monitored by a service of the OpenStack known as telemetry service. Namely, telemetry services rely on Ceilometer and Gnocchi [TelemetryOpenStack][6] to gather NFVI measurements and expose them through API endpoints.
- Finally, the NFV MANO provides the test measurements to the test control functional block. And this one evaluates whether the tests are passed according to the corresponding KPIs.

[5] Note that in SEMIoTICS the NFV MANO is composed of two main blocks. Open Source MANO (OSM) implements the NFV Orchestrator (NFVO) and the VNF Manager (VNFM). The VIM is implemented by OpenStack.
[6] [TelemetryOpenStack] https://wiki.openstack.org/wiki/Telemetry

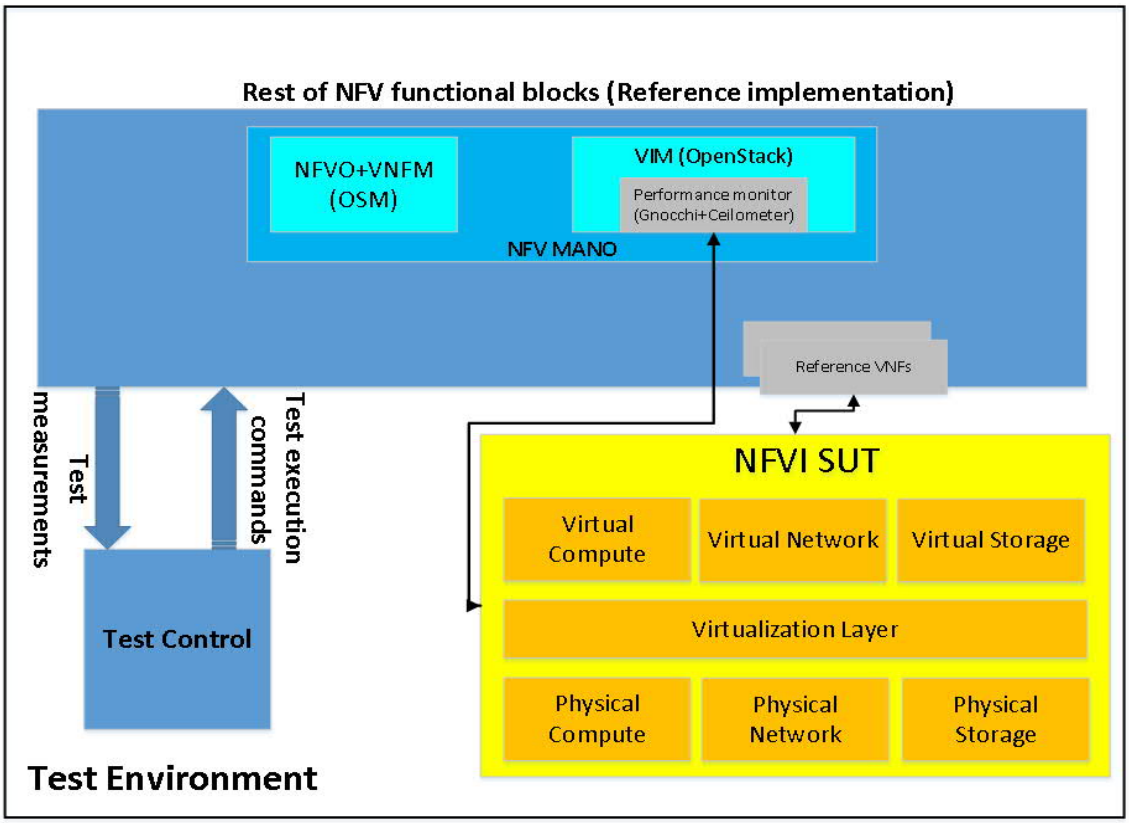

FIGURE 2: TEST ENVIRONMENT WHEN THE SUT IS THE NFV

Another example of a SUT within the NFV context is illustrated in Figure 3. Note that this figure corresponds to the case where the SUT is a NS. As a NS by definition is composed of several VNFs, in turn, each VNF is a FUT. Note that the case where the SUT is one VNF is a particular case of Figure 3. The evaluation methodology is similar than the one explained for the case where the SUT is the NFVI. Namely, the procedure is as follows:

- The test control block leverages a set of test functions, which can be implemented as Physical Network Functions (PNFs) or VNFs to control the tests of the FUT.
    - They enable traffic scenarios towards the NS.
    - They provide the interfaces to expose measurements related to the performance indicators.
- The test control block receives the measurements exposed by the test functions and evaluates whether the KPIs related to the SUT, i.e. the NS, are fulfilled.
- Note that the NS under test, i.e. the SUT, is deployed on top of the NFVI by means of the NFV MANO. That is, it is responsible to allocate the necessary virtual resources to deploy the NS and to manage the NS lifecycle.
- In the case where the test functions are implemented as VNFs, also the NFV MANO is leveraged to instantiate the test functions on top of the NFVI.

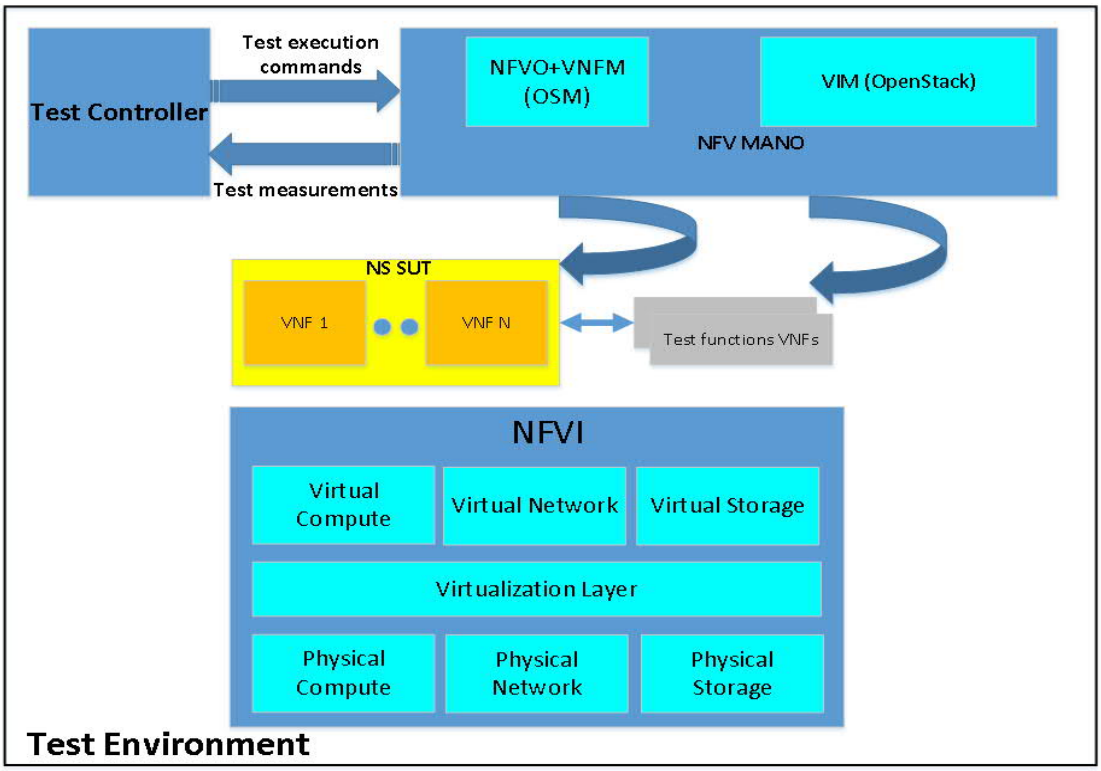

FIGURE 3: TEST ENVIRONMENT WHEN THE SUT IS A NS

Test variables: At this point, it is important to mention that herein we adopt the next common methodology for testing purposes. Namely, the SUT is isolated in order to reduce the number of variables in the test. The rationale is that it is easier to ensure that the performance being measured is that of the SUT itself, without being influenced by the variables related to other devices. Also, the isolation of the SUT paves the way to repeat deterministic configurations and results.

Moreover, in the line of the above paragraph, note that there are two type of parameters that define the configuration of the tests:

- The fixed configuration parameters: these parameters remain constant for all iterations of a test.
- The variable configuration parameters: these can be modified between iterations of a test.

Type of tests: In general, the next type of tests will be taken into account for the evaluation of the SUT:

- Performance verification. The goal is to validate that a set of performance objectives are attainable, when the SUT is under fixed conditions.
- Benchmarking. The aim of this type of tests is to study the maximum performance that a SUT can achieve for a given metric of interest.
- Dimensioning. The aim is to find out the amount of infrastructure resources required to obtain a given performance for a set of metrics of interest.

Test metrics and test environment: As it was mentioned above, the SUT is divided into FUTs. Moreover, in general, each FUT has associated a set of requirements. An important task in the testing methodology is to translate these requirements into proper metrics that allow to evaluate whether a given test is passed or failed. As a consequence, each set of metrics leads to define a test case associated to the FUT. These test cases are executed by the supporting test environment. For instance, bearing in mind the NFV case, recall that the NFVI can be a SUT. Also, note that each VNF application requires its own set of metrics. Thereby, test cases associated to the NFVI arise from investigating whether the NFVI is able to fulfill the metrics and requirements of the VNF applications.

In general, the test metrics are related to the next three subgroups: compute, storage and networking. Moreover, each subgroup can be further split categorized in the next categories:

- Performance/Speed. For instance, processing speed (instructions per second) is an example of a compute performance metric.
- Capacity/Scale. For instance, maximum throughput of a network node.
- Reliability/Availability. The disk mean-time-to-failure is an example of a storage reliability metric.

Consider a given a set of metrics that characterize a given test. Then, in order to validate a test, the supporting test environment executes in general the following steps:

- Configuration and deployment. For instance, in an NFV platform, this step determines the number of virtual machines acting as test functions. Virtual computing and storage resources per virtual machine and the network configuration.
- Test execution.
- Test validation.

Test Structure: It is important to have a structure for all the information related to the tests. For instance, all the information that describes the test, how it is executed and the test conditions. Thereby, for testing purposes, it is very helpful that all the information is condensed in the form of a table. To this end, in Table 3: Test Structure, we present a table structure for the tests.

TABLE 3: TEST STRUCTURE

Test title	
Test identifier	E.g. Test_throughput_1
Metric	E.g. network throughput
Test purpose	E.g. measure the L3 net throughput between two nodes.
Configuration	This is the description of the setup needed to carry out the test.
References	These are the underlying documents that may characterize part of the test.
Applicability	Scenarios where this type of test is applicable.
Pre-test conditions	Here we describe the initial conditions of the test parameters.
Test sequence	Here we describe the sequence of steps needed to carry out the test.
Test verdict	Describes whether the test is passed or not.

4.3. Trial Applications

This section anticipates the definition and scope related to the trial applications within SEMIoTICS project, in relation to the objectives of the overall evaluation methodology defined for the project. The approach, since the beginning, has been to establish, during whole project lifespan, a continuous integration development of all the components at different layers, organized in several incremental macro cycles. For each development cycle a specific evaluation methodology has been identified to ensure at each step that requirements and main milestones were met. The overall evaluation methodology has been defined as well as an incremental set of incremental tests that enables the coherent evolution, in term of implemented functionalities, of all the SEMIoTICS components. Thus, the evaluation has been carried out during project development in an incremental form tool.

From the initial definition and identification of all the KPIs indicators (reported in section 3) a set of controlled lab experiments, mainly to test a component in isolation to verify expected interfaces and behavior are met (see as an example, the NFV infrastructure testing described in section 4.2). Trial applications is a methodology defined in SEMIoTICS to ensure proper integration and interoperability between the component are met and will be widely used during 3rd year of the project during the use cases integration tasks from T5.4, T5.5 and T5.6. The main rational is to incrementally integrate all the components developed in SEMIoTICS, by testing them using incremental mockups trial applications that interact with a subset of components in order to validate part of the functionalities of each use case scenario. In this definition, the three main demonstrators, that will be delivered at the end of the project, will be, to some extent, a trial application that interacts with all the required components in order to implement the vertical scenario. The role of trial applications within SEMIoTICS is shown in following diagram reporting a typical development cycle:

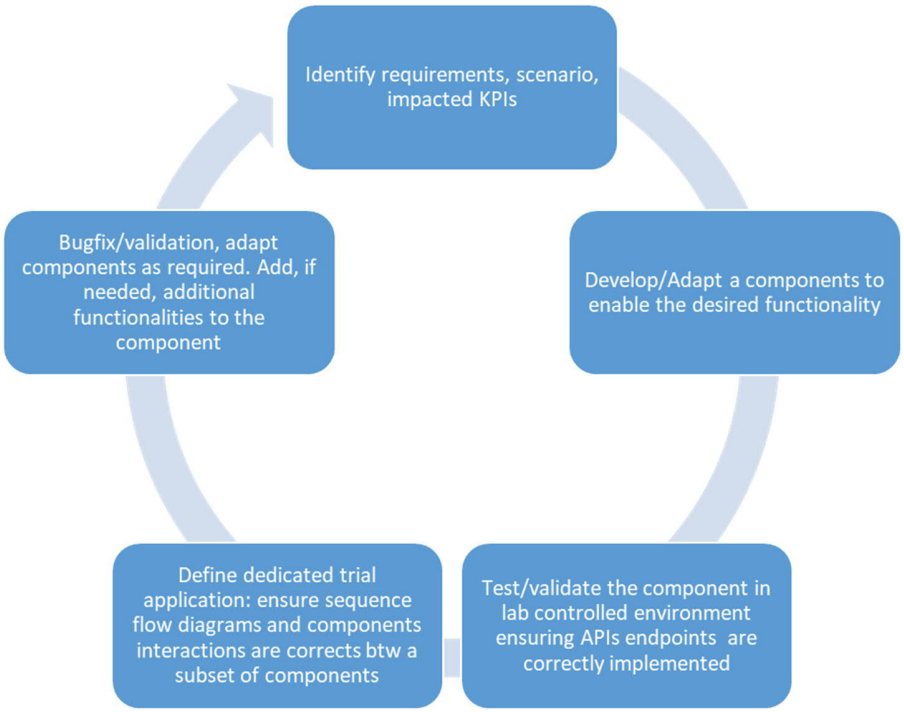

FIGURE 4: COMPONENTS AND TRIAL APPLICATIONS IN SEMIOTICS

4.4. Evaluation and Cross-Validation Methodology and Criteria

This section describes the cross-validation methodology and the criteria to evaluate the SEMIoTICS architectural framework. This evaluation and criteria measure the technical effectiveness, through two different types of tests, lab experiments that will test some components and subsystems in a controlled environment, and trials that will test the architectural framework in the context of three different use cases. Besides the technical criteria the use cases have specific KPI's to measure the specific performance of the application of the SEMIoTICS architecture in real world scenarios, complementary to the evaluation criteria defined later in this section.

4.4.1. DEFINITION OF SEMIOTICS EVALUATION METHODOLOGY

In the scope of the SEMIoTICS project, the evaluation methodology provides the connection between the development of the architectural framework and the lab experiments and use case-based trials to ensure that the architecture provides the expected performance and functionalities.

To perform the evaluation the designers have provided the evaluation criteria based on the critical innovations of the system, that is, the target for the evaluations. These evaluations will be performed by the evaluators, partially through lab experiments performed by a group of partners which have participated in the design and development of the solution, and through the test trials in field trials environments considered in the project.

4.4.1.1. EVALUATION MODEL

The ISO has defined a set of series of Standards dedicated to software product quality and evaluation. ISO/IEC14598 [7] series of standards specify the evaluation methodology for general software product in information technology. ISO/IEC9126 [8] series of standards specify metrics for product quality in software engineering and a simplified process for evaluation. These two series of standards are complementary as shown in Figure 5 below.

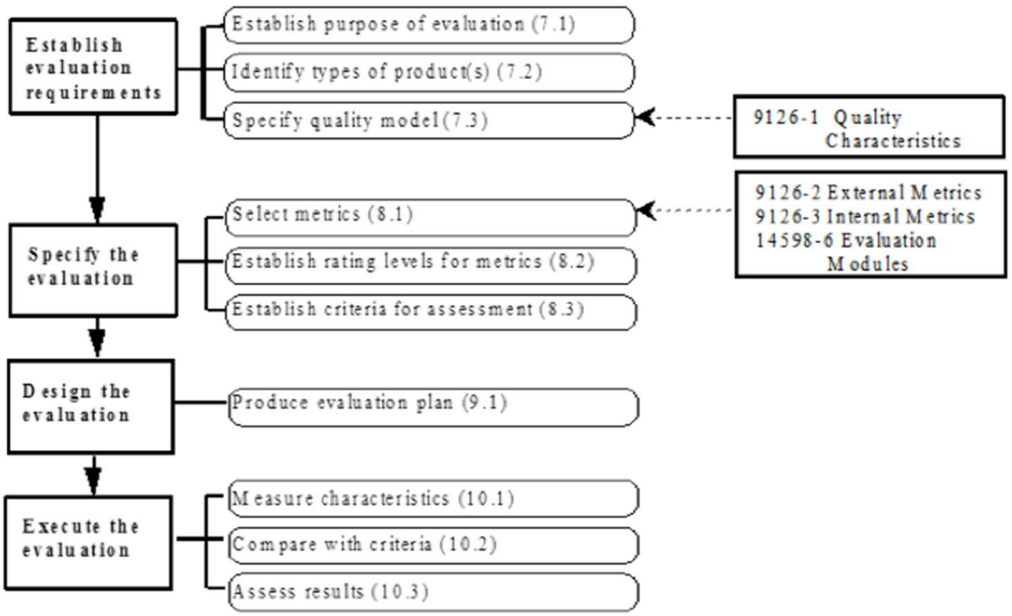

FIGURE 5 EVALUATION PROCESS VIEW ACCORDING TO ISO/IEC 14598-1 [7]

Step 1: Establish evaluation requirements

This step establishes the purpose and the products to evaluate, that in the case of SEMIoTICS is to test the architectural framework through lab experiments and live trial tests based on the use cases. The quality model in this case is based on fulfilling the innovation requirements which is the key differentiator of the SEMIoTICS framework from other IoT existing architectures.

Step 2: Specify the evaluation

This step comprises the activities for the selection of metrics, establishing the rating levels and the criteria for assessment. The quantitative specification and measurement of the software quality requirements can only be made by using metrics which are associated to desired quality characteristics. For each selected metric evaluation rating values are defined for the related scale, where the required level of the attribute to be measured can be expressed. Besides the evaluation criteria, each use case has its related KPI's and performance metrics that will provide an evaluation of the impact of each use case, besides the SEMIoTICS architectural framework. For the use case related KPI's and performance metrics, when applicable, reference measurements should be considered when these measures compare to situations previous to the deployment of the trials, well based on existing statistics, or performing specific measurements previously to the deployment of the trial use case.

Step 3: Design the evaluation

This step defines the evaluation activities and methods. In *SEMIoTICS,* it comprises the in-lab experiments, and the use case trials, where the specific data will be collected to check that the different evaluation criteria meet the evaluation metrics.

[7] ISO, ISO 14598-1. "Information technology - Software product evaluation - Part 1: General overview", April 1999.

[8] SO, ISO 9126-1. "Software engineering - Product quality - Part 1: Quality model", June 2001.

Step 4: Execute the evaluation

The selected metrics are applied to the components or solutions, resulting in values on the scales of the metrics. The measured values are then compared to the criteria established in the specification. In the assessment activity a set of rated values are summarised and a statement of the extent to which the software product meets quality requirements is made.

4.4.1.2. EVALUATION PROCESS

The approach of evaluation process is composed of three steps, one for the evaluation of the in-lab experiments that will assess the performance of architectural components, a second one to evaluate the overall system on a proof of concept field trials approach, and a third one to perform final cross- evaluation of each UC to assess the portability of the system.

Step 1: In-lab experiments evaluation process. In this process it will be performed the experiments defined in section 3 Proof-of-Concept Laboratory experiments, in a controlled environment, assessing the performance of the specified functionalities. This step will be performed in task T5.3:

1. The results will be measured against the evaluation criteria KPIs defined for each experiment and detailed in section 2.
2. Any deviation from the expected results will be assessed to improve the related system modules or to know their limitations.
3. Any improvement will be incorporated in the modules to be integrated in the trials use cases.

Step 2: Field trials evaluation process. In this process the SEMIoTICS architecture will be evaluated with its deployment in different scenarios based on use case descriptions in tasks T5.4 – T5.6:

1. Deployment of the UCs. Including the following activities
    a. Deployment of the integrated components for the use case with the corresponding hardware and software modules.
    b. Collect any deployment issues to provide early input to the other trial phase.
2. Running the trials to gather information defined for each use case, either based on generic or use case specific criteria). Collect any issues during the execution of the trial to provide early input for the second trial phase.
3. Evaluation of measurements through the metrics specified for each criterion and against the specified targets.
4. Exchange of evaluation results and the report of trial deployment and execution issues collected during phase 1 trials, with the support of the technical partners to optimise the deployment of the second phase trials.
5. Perform the second phase of the trials for each use case making the improvements recommended by the issues report of the first phase. Execute previous points 1 to 4.

The deployment and execution issues should be collected also during phase 2 of the trials as will be used also for the final cross-evaluation.

Step 3: Cross-evaluation process. This process will receive the results of the two phases of the trials performed in tasks T5.3 and T5.4 - T5.6 and will analyse the results of each use case deployment to perform a framework cross-evaluation to assess the portability of the SEMIoTICS architectural framework.

1. Collect the evaluation reports and the reports of trial deployment and execution issues from each use case from the two trial phases.
2. Analyse found issues from the collected reports and evaluate if they correspond to:

    a. Specific deployment or execution conditions for one of the given use cases. If the issue was only found during the phase-1 trials inquire if this was not found during phase-2 trials because it was avoided after following the recommendations from the reported issue in phase-1 or because the conditions of the trial in phase-2 are different than those in phase-1

    b. Specific deployment or execution conditions in the trial for both cities for one of the given use cases. If the issue was found in both use cases, determine, if this was because specific conditions found in

both trials or if this issue is independent from those specific conditions, and therefore it will replicate if this scenario is deployed in another use case.,

3. Analyse evaluation results from measurements and look for deviations from expected targets:
    a. For generic evaluation criteria, analyse if these deviations are found in one of the following cases to determine if it is dependent from one specific condition in the UC or if it is inherent to the architectural framework.
    b. For use case specific evaluation criteria analyse if these deviations are found only in one trial or in both to determine if it is dependent from one specific condition or if it is inherent to the architectural framework.
    c. Analyse results, specific for each use case, and look for deviations from expected targets. Analyse if these deviations are found only in one trial or in both to determine if it is dependent from one specific condition or if it is inherent to the architectural framework.
4. Compile the conclusions from the cross-evaluation process to bring out SEMIoTICS's portability to other use cases or the same tested use cases.

4.4.2. CROSS CHECKING METHODOLOGY

A cross checking methodology is required to assess the generality of SEMIoTICS framework and approach. The aim of this subsection is to provide the plan for validating the SEMIoTICS architectural framework and its components with regard to the technical objectives and innovations of the project, which is planned at two levels – through in-lab experiments and use case trials to ensure that the architecture provides the expected performance and functionalities.

The validation of the framework contains lab experiments and trial activities including a set of inter-related tasks where some of them provide feedback to other tasks.

Lab experiments include a proof of concept-controlled experiments to assess the performance of the components defined in **WP2** and developed within **WP3-WP4**. The results of these lab tests will be used to improve the components tested, and the conclusions will be applied in the first phase of the testing. In parallel to the first phase, the lab experiments will continue to improve those components with some performance issues, and the final conclusions and improvements will be provided in M30 just in the middle of the second phase of the use case trials.

Software system integration (T5.2), testing and infrastructure setup (T5.3) tasks start at M13 and will run until M32, performing the implementation in terms of development of specific use case components and integration of the trials for the three use cases. Between M18 and M32 the effective integration of the software components will take place in **T5.2** and will be reported in D5.2 Software system integration (Cycle 1), while between M13 and M32 feedback will be collected in **T5.3** from initial testing of the infrastructure including the experiments and trials to implement necessary revisions in the developments. Report *D5.3 IIoT Infrastructure set-up and testing (cycle 1)* provides the specifications of the hardware and software developed infrastructure for the in-lab experiments and trials.

Trials Cycle 1 run from M24, performing the preparation activities until M30 collecting the information to evaluate the performance of the SEMIoTICS architectural framework for each UC running distributed in the location of the participants of the use cases (UC1 in Munich and Heraklion, UC2 in Rome, Heraklion and Passau, UC3: Milan, Barcelona).

Trials cross reporting, between M30 and M31, include the debriefing activities from the phase 1 trials, compiling the issues fund during the trials the evaluation of the measurements collected during the trials and evaluation results. The results will be reported in *D5.4-5.6 Demonstration and Validation of Use Cases (cycle 2).*

Trials Cycle 2 run from M31, performing the evaluation activities until M36. These evaluation activities are equivalent to those described for trials in cycle 1 but in the location of the relevant field trials (UC1 in Munich, UC2 in Rome and UC3 in Barcelona).

The final step is the **Cross Evaluation** that will collect the results of the two trial phases, analyzing the results of the trials in the three use cases to assess the portability of the SEMIoTICS architectural framework. The conclusions will be reported in *D5.9-D5.11 Demonstration and Validation of Use Cases (cycle 2).*

5. CONCLUSION

This document delivers the comprehensive list of all KPIs of the project. This allows consortium members, as well as readers external to the consortium, to identify how SEMIoTICS will measure results in terms of the project's objectives. The mapping of KPIs to the project's objectives aims to provide a structure to the KPIs and Objectives validation, while verifying complete coverage of the latter.

For all listed KPIs the document also provided their scope, which allows, where appropriate, to immediately identify from which use case(s) this KPI has been elicited during its gathering and where it will be applicable. For all the UC's that are mentioned for each KPI the document then provides a short statement defining what –often very technical– methodology is used to evaluate it. To be even more precise, we aimed to define which components are involved in the specific KPI. Further, we provided in the Appendix how each KPI relates to the different tasks of the project.

It is within those tasks that the fulfilment of the respective requirements is checked and also that the KPI relating to those tasks is checked. Thus, you will find the evaluation in the in different deliverables, e.g. the security-related ones are listed and checked in D4.12. Due to the scope being KPIs you will not find requirements and how they will be tested in the deliverable D5.1. These KPIs foremost map to objective, but as well to different use-cases (with specific technical requirements) and different project-wide tasks like dissemination. Thus, the requirements listed in WP deliverables, especially D2.2 and D2.3, are also checked for fulfilment in the respective technical deliverables, e.g. security-related ones in D4.12, and thus the linkage between fulfilment of requirements and the resulting fulfilment of the specific (i.e. not the project-wide) KPIs is best understood from the respective deliverable. This document thus concentrates on stating the evaluation methodology for the objective-related KPI**.**

The methodology for evaluation, which is described as clearly and concretely as possible, while still being brief and readable, allows measuring the KPI. The project will retain the KPI-ID as a unique identifier when providing the results of the evaluation in upcoming future final deliverables.

APPENDIX A: MAPPING KPIS TO TASKS

Objective		KPI-ID			Description	WP1			WP2			WP3					WP4					WP5					WP6			M01-M18	
					Short	T1.1	T1.2	T1.3	T2.1	T2.2	T2.3	T2.4	T3.1	T3.2	T3.3	T3.4	T3.5	T4.1	T4.2	T4.3	T4.4	T4.5	T4.6	T5.1	T5.2	T5.3	T5.4	T5.5	T5.6	T6.1	T6.2	T6.3	% Per KPI	% Per Obj.
1	SPDI Patterns	KPI-1.1			Number of SPDI Patterns													O					O			O	O	O				60.00%	60.00%
		KPI-1.2			Pattern Language													O					O			O	O	O				60.00%	
2	Semantic Interoperability	KPI-2.1			Semantic descriptions for 6 types of smart objects										O			O		O			O									60.00%	46.67%
		KPI-2.2			Data type mapping and ontology alignment																O		O										40.00%	
		KPI-2.3			Semantic interoperability with 3 IoT platforms				O			O				O				O			O										40.00%	
3	Monitoring Mechanisms	KPI-3.1	KPI-3.1.1		Generating monitoring strategies in the 3 targeted IoT platforms													O	O					O									30.00%	38.33%
			KPI-3.1.2		Fuse results from these monitors														O	O				O									30.00%	
			KPI-3.1.3		Performing predictive monitoring with an average accuracy of 80%													O	O					O									20.00%	
		KPI-3.2			Delivery of a monitoring language										O			O	O					O									50.00%	
4	Multi-layered Embedded Intelligence	KPI-4.1			Delivery of lightweight ML algorithms															O				O					O				50.00%	40.00%
		KPI-4.2			Delivery of mechanisms with adaptation time of 15ms										O			O	O					O	O								30.00%	
		KPI-4.3			Delivery of adaptations mechanisms enabling improvement by at least 20%														O	O				O		O			O				30.00%	
		KPI-4.4			Detection time of less than 10 ms															O				O		O			O				30.00%	
		KPI-4.5			Baseline improvement of 20% adaptation time															O				O		O			O				30.00%	
		KPI-4.6			Development of new security mechanisms/controls											O		O				O		O									70.00%	
5	IoT-aware Programmable Networks	KPI-5.1			Deployment of a multi-domain SDN orchestrator							O	O											O				O					50.00%	60.00%
		KPI-5.2			Service Function Chaining (SFC) of a minimum 3 VNFs								O	O				O						O				O					70.00%	
6	Development of a Reference Prototype	KPI-6.1			Reduce Required Manual Interventions									O	O	O		O						O									40.00%	40.00%
		KPI-6.2			Leveraging upon FIWARE assets				O															O	O								40.00%	
		KPI-6.3			Delivery of 3 prototypes of IIoT/IoT applications					O		O					O						O	O			O	O					40.00%	
7	Promote the adoption of EU technology offerings internationally	KPI-7.1			Provision the SEMIoTICS framework and building blocks							O												O	O								50.00%	46.74%
		KPI-7.2			Achieve influencer status within major standardization efforts																			O							O		30.00%	
		KPI-7.3	KPI-7.3.1	KPI-7.3.1.1	Project website	O																		O						O			80.00%	
				KPI-7.3.1.2	Push announcements																			O						O			70.00%	
				KPI-7.3.1.3	Regular Newsletter																			O						O			30.00%	
				KPI-7.3.1.4	Brochure																			O						O			40.00%	
				KPI-7.3.1.5	Technical video																			O						O			30.00%	
			KPI-7.3.2	KPI-7.3.2.1	Journal publications																			O						O			60.00%	
				KPI-7.3.2.2	Magazine publications																			O						O			15.00%	
				KPI-7.3.2.3	Conference Publications			O																O						O			80.00%	
				KPI-7.3.2.4	Special Issues																			O						O			75.00%	
			KPI-7.3.3	KPI-7.3.3.1	Conference organizations			O																O						O			80.00%	
				KPI-7.3.3.2	Workshops																			O						O			70.00%	
				KPI-7.3.3.3	Summer Schools																			O						O			50.00%	
			KPI-7.3.4	KPI-7.3.4.1	Exhibition demonstrations			O																O							O		70.00%	
				KPI-7.3.4.2	EU demonstrations			O																O							O		80.00%	
				KPI-7.3.4.3	Conference demonstrations			O																O							O		30.00%	