Project Acronym:	**VICINITY**
Project Full Title:	**Open virtual neighbourhood network to connect intelligent buildings and smart objects**
Grant Agreement:	**688467**
Project Duration:	**48 months (01/01/2016 - 31/12/2019)**

Deliverable D3.5

Semantic discovery and dynamic configuration services

Work Package:	**WP3 – VICINITY Server Implementation**
Task(s):	**T3.2 – Semantic discovery and dynamic configuration services**
Lead Beneficiary:	**IS**
Due Date:	**30 June 2018 (M30)**
Submission Date:	**29 June 2018 (M30)**
Deliverable Status:	**Final**
Deliverable Type[1]:	**DEM / R**
Dissemination Level[2]:	**PU**
File Name:	**VICINITY-D3.5-Semantic-discovery-and-dynamic-configuration-services-v1.0.pdf**

This project has received funding from the European Union's Horizon 2020 Research and innovation programme under Grant Agreement n°688467

VICINITY Consortium

No	Beneficiary		Country
1.	TU Kaiserslautern (Coordinator)	UNIKL	Germany
2.	ATOS SPAIN SA	ATOS	Spain
3.	Centre for Research and Technology Hellas	CERTH	Greece
4.	Aalborg University	AAU	Denmark
5.	GORENJE GOSPODINJSKI APARATI D.D.	GRN	Slovenia
6.	Hellenic Telecommunications Organization S.A.	OTE	Greece
7.	bAvenir s.r.o.	BVR	Slovakia
8.	Climate Associates Ltd	CAL	United Kingdom
9.	InterSoft A.S.	IS	Slovakia
10.	Universidad Politécnica de Madrid	UPM	Spain
11.	Gnomon Informatics S.A.	GNOMON	Greece
12.	Tiny Mesh AS	TINYM	Norway
13.	HAFENSTROM AS	HITS	Norway
14.	Enercoutim – Associação Empresarial de Energia Solar de Alcoutim	ENERC	Portugal
15.	Municipality of Pylaia-Hortiatis	MPH	Greece

[1] ***Deliverable Type:***

R:	Document, report (excluding the periodic and final reports)
DEM:	Demonstrator, pilot, prototype, plan designs
DEC:	Websites, patents filing, press & media actions, videos, etc.
OTHER:	Software, technical diagram, etc.

[2] ***Dissemination level:***

PU:	Public, fully open, e.g. web
CO:	Confidential, restricted under conditions set out in Model Grant Agreement
CI:	Classified, information as referred to in Commission Decision 2001/844/EC.

Disclaimer

This document reflects only the author's views and the European Union is not liable for any use that may be made of the information contained therein.

Authors List

Leading Author (Editor)				
Surname		First Name	Beneficiary	Contact email
Kostelnik		Peter	IS	peter.kostelnik@intersoft.sk
Co-authors (in alphabetic order)				
No	**Surname**	**First Name**	**Beneficiary**	**Contact email**
1.	Kostelnik	Peter	IS	peter.kostelnik@intersoft.sk
2.	Cimmino	Andrea	UPM	cimmino@fi.upm.es
3.	Serena	Fernando	UPM	fserena@fi.upm.es
4.	Poveda	María	UPM	mpoveda@fi.upm.es
5.	García Castro	Raúl	UPM	rgarcia@fi.upm.es
6.	Vanya	Stefan	BVR	stefan.vanya@bavenir.eu
7.	Almela Miralles	Jorge	BVR	jorge.almela@bavenir.eu

Reviewers List

List of Reviewers (in alphabetic order)				
No	**Surname**	**First Name**	**Beneficiary**	**Contact email**
1.	Hovstø	Asbjørn	HITS	hovsto@online.no
2.	Heinz	Christopher	UNIKL	heinz@cs.uni-kl.de
3.	Almela Miralles	Jorge	BVR	jorge.almela@bavenir.eu

Revision Control

Version	Date	Status	Modifications made by
0.1	4.6.2018	Initial Draft	Peter Kostelnik (IS)
0.2	5.6.2018	Added release notes	Peter Kostelnik (IS)
0.3	6.6.2018	Added BVR and UPM contributions (GitHub)	Stefan Vanya (BVR), Jorge Almela Miralles (BVR), Raúl García Castro (UPM), María Poveda (UPM), Andrea Cimmino (UPM), Fernando Serena (UPM), Peter Kostelnik (IS)
0.4	11.6.2018	Corrections according to reviewer's comments	Asbjørn Hovstø (HITS), Christopher Heinz (UNIKL), Peter Kostelnik (IS)
0.5	19.6.2018	Corrections according to reviewer's comments. Applied actual deliverable template.	Asbjørn Hovstø (HITS), Christopher Heinz (UNIKL),Peter Kostelnik (IS)
0.6	26.6.2018	Corrections according to reviewer's comments.	Asbjørn Hovstø (HITS), Christopher Heinz (UNIKL), Jorge Almela Miralles (BVR), Peter Kostelnik (IS)
0.7	28.6.2018	Quality Check, final Draft Rewieved	Asbjørn Hovstø (HITS), Christopher Heinz (UNIKL), Jorge Almela Miralles (BVR), Peter Kostelnik (IS)
1.0	29.6.2018	Final editing of executive abstract Submission to the EC	Christoph Grimm (UNIKL)

Executive Summary

This deliverable is an important part to reach the milestone MS5 VICINITY server components/services and client infrastructures available, which is the outcome of work packages WP3 - VICINITY server implementation and WP4 - VICINITY Client Infrastructures Implementation (M30). This objective has been reached in time. This deliverable summarizes the outcomes of task T3.2, "Semantic discovery and dynamic configuration services".

Semantic discovery and dynamic configuration platform is the part of VICINITY cloud as shown in see Figure 1. The main purpose of semantic discovery and dynamic configuration platform is to enable the semantic interoperability in VICINITY. The platform serves as the main repository of IoT object descriptions. Descriptors are stored as semantic models, which are linked to the VICINITY ontology. The platform enables to perform the full semantic search for object descriptions, so existing objects providing required functionality can be easily discovered. IoT object descriptions are also used as the core information for auto configuration of agents, through which these objects are available for interaction.

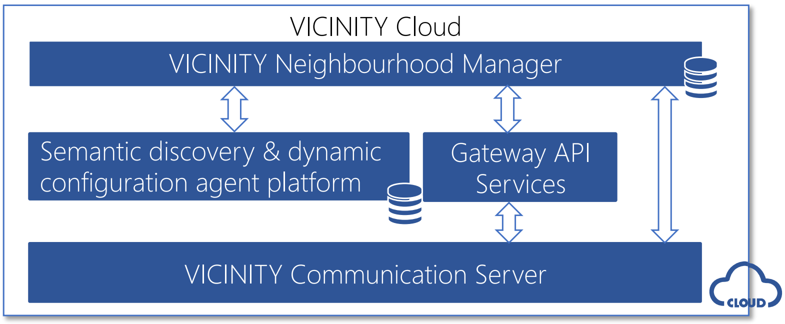
VICINITY Cloud
VICINITY Neighbourhood Manager
Semantic discovery & dynamic configuration agent platform
Gateway API Services
VICINITY Communication Server
CLOUD

Figure 1 VICINITY Cloud

Table of Contents

List of Figures

List of Definitions & Abbreviations

Abbreviation	Definition
EC	European Commission
EU	European Union
GIT	The version control system
GitHub	Software platform managing GIT repositories
IoT	Internet of Things
API	Application programming interface
CRUD	Basic functions of database, Create, Read, Update, Delete
JSON	JavaScript Object notation, serialization formal
JSON-LD	JSON for Linked Data
RDF	Resource Description Framework
N3	Notation3, specific serialization notation for RDF
triple	Atomic semantic data entity in RDF defining subject-predicate-object
triplestore	Semantic database for triples

1. Introduction

This deliverable describes the implementation of the semantic discovery and dynamic configuration platform. The semantic discovery and dynamic configuration platform is realized as two separate components:

1. **Semantic triplestore**. Semantic database used as the physical storage for semantic models (ontology) and semantic IoT descriptions
2. **Semantic discovery and configuration services.** The standalone REST server connected to semantic triplestore. This component provides the API for manipulating IoT object descriptions. This includes the CRUD, SPARQL endpoint to enable rich semantic search; and several transformation modules enabling translation from JSON to semantic model (N3, RDF) and vice versa.

The functionality of semantic discovery and dynamic configuration consist of the cloud-part commonly shared in VICINITY (this component) and the twin client-side component described T4.2 VICINITY Agent and auto discovery platform. Thus, to understand the whole semantic discovery and dynamic configuration, both cloud and client-side components must be checked. The cloud side described in this deliverable serves mostly as the storage and search engine and the client-side implements the full discovery and configuration process. Both components were designed and implemented together.

The outcomes of task T3.2 also contain the update and actualization of VICINITY ontologies (the extension of D2.2 Detailed Specification of the Semantic Model) and design/implementation of VICINITY interoperability approach among the data exchanged by IoT objects.

This document includes only the Release Notes for the semantic discovery and dynamic configuration platform. The task T3.2 formally ends in M30, however, the development of this component still continues in implementing the pilot applications. The actual list of features will be continually extended, driven by specific requirements on the fly, as pilot applications will be developed.

The full actual documentation including the code, history of commits, more detailed description and all technical details is available at GitHub:

https://github.com/vicinityh2020/vicinity-semantic-platform

1.1. Context within VICINITY

The work in deliverable D3.5 is based on requirements specified in WP1, semantic models delivered within WP2 and was designed and implemented with very tight cooperation with the client-side twin delivered by WP4. The relationships are illustrated in Figure 2.

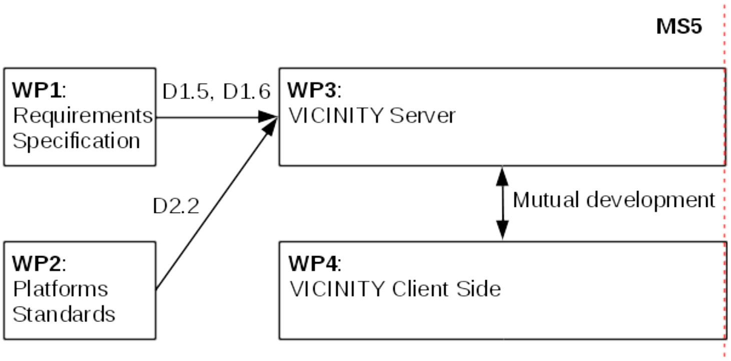

Figure 2 Deliverable context

Design and development of Semantic discovery and dynamic configuration services was driven by architectural design and requirement specification. Definition of VICINITY ontologies was an important input for the design of IoT descriptors. And the task was a coordinated development of auto-discovery and configuration at client-side.

Relation to other deliverables:

D4.2 VICINITY Agent and auto discovery platform describes the client-side twin, which was designed and developed together with the cloud-side component described in this deliverable

- D1.5 VICINITY technical requirements specification summarized the basic requirements identified for this component
- D1.6 VICINITY architectural design describes the position of this component within the VICINITY architecture
- D2.2 Detailed Specification of the Semantic Model describes the VICINITY ontology which is used as the backbone for semantic inference and reasoning
- D4.2 VICINITY Agent and auto discovery platform describes the client-side twin, which was designed and developed together with the cloud-side component described in this deliverable

1.2. Objectives in Work Package WP3 and Task T3.2

The goal of WP3 - VICINITY server implementation is to implement and deploy the VICINITY server components, namely the VICINITY communication Server, VICINITY Neighbourhood Manager and Open Interoperability Gateway API. The part of WP3 objectives is the implementation of Semantic Discovery and Dynamic Configuration Services, which are focused in task T3.2. The goal of this task is to implement and deploy the platform responsible for management of semantic descriptors of IoT objects. IoT objects are reflected semantic models mapped into the VICINITY ontology. The platform also provides the services for rich semantic search and discovery of IoT objects within VICINITY.

2. Release notes

This section contains the overview of Semantic discovery and dynamic configuration platform functionality.

2.1. Current features

This section contains the list of all implemented features according to all relevant requirements defined in D1.5 VICINITY technical requirements specification and according to architecture specification defined in D1.6 VICINITY architectural design.

- Actualization and extensions of VICINITY ontologies
- Model transformations
    - Transformation from VICINITY Common Thing Description format (JSON) into semantic representation using JSON-LD transformation
    - Transformation from semantic model (semantic graph) into VICINITY Common Thing Description format (JSON)
- Model validation (both cloud and client-side use the same model validator)
    - Full syntactic and structural validation of IoT object descriptor
    - Full semantic validation of all semantic annotations included in IoT object descriptor
- Population of IoT object repository
    - Create semantic model of IoT object descriptor
    - Update semantic model of IoT object descriptor
    - Delete semantic model of IoT object descriptor
    - Retrieve semantic model of IoT object descriptor (and transform it to JSON)
- Standard SPARQL endpoint to enable the full semantic search
- Developer tools and utilities
    - Semantic annotations
        - The plain list of all semantic annotations available in ontology separated by annotation type
        - The full annotation tree including the class taxonomy and individuals, separate tree per annotation type
    - Syntactic and semantic validation service enabling developers to validate and debug their IoT descriptors in advance
- Design and implementation of VICINITY Interoperability among the data exchanged by IoT objects

2.2. Planned features

Semantic discovery and dynamic configuration platform will be further maintained and extended with new features, following the requirements from other software components and the pilot site applications. Development will be coordinated using selected issue tracking tools.

- Further extensions to enable more possibilities for semantic search and IoT object discovery
- Implementation of requirements for pilot applications implementation
- Updates for the identified issues

3. Conclusions

This deliverable covered the outcomes of the task T3.2 VICINITY Semantic Discovery and Dynamic Configuration services. The full semantic discovery and configuration process consists of cloud-part and client-part. This deliverable covers the cloud-part, which serves mostly as the storage and search engine. Most of the configuration and discovery logic is implemented at the client-side, which is available in D4.2 VICINITY Agent and auto discovery platform. To fully understand the process, both deliverables should be checked.

The full actual documentation including the code, history of commits, more detailed description and all technical details are available at GitHub:

https://github.com/vicinityh2020/vicinity-semantic-platform