



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 D3.7

VICINITY core components continuous upgrades, finalversion

Work Package:	WP3 – VICINITY Server Implementation
Task(s):	T3.3 - VICINITY Operation and continuous upgrades of core components
Lead Beneficiary:	BVR
Due Date:	31 December 2019 (M48)
Submission Date:	19 December 2019 (M48)
Deliverable Status:	Final
Deliverable Type:	DEM
Dissemination Level:	PU
File Name:	VICINITY_D3.7_VICINITY_core_components_continuous_upgrades_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.	Enercutim – Associação Empresarial de Energia Solar de Alcoutim	ENERC	Portugal
15.	Municipality of Pylaia-Hortiatis	MPH	Greece

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.

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.

Authors List

Leading Author (Editor)				
Surname	First Name	Beneficiary	Contact email	
Oravec	Viktor	BVR	viktor.oravec@bavenir.eu	
Co-authors (in alphabetic order)				
No	Surname	First Name	Beneficiary	Contact email
1.	Almela Miralles	Jorge	BVR	jorge.almela@bavenir.eu
2.	Cimmino	Andrea	UPM	cimmino@fi.upm.es
3.	Kostelnik	Peter	IS	peter.kostelnik@intersoft.sk
4.	Poveda Villalón	María	UPM	mpoveda@fi.upm.es
5.	Vanya	Stefan	BVR	stefan.vanya@bavenir.eu

Reviewers List

List of Reviewers (in alphabetic order)				
No	Surname	First Name	Beneficiary	Contact email
1.	Koutli	Maria	CERTH	mkoutli@iti.gr
2.	Mach	Marian	IS	marian.mach@tuke.sk
3.	Sundvor	Mariann	TINYM	mariann@tiny-mesh.com

Revision Control

Version	Date	Status	Modifications made by
0.1	25/11/2019	First draft	Jorge Almela (BVR)
0.2	10/12/2019	Version ready for QAR	Jorge Almela (BVR)
0.3	17/12/2019	Version including TINYM and IS QAR recommendations	Jorge Almela (BVR)
0.4	18/12/2019	Version including CERTH QAR recommendations	Jorge Almela (BVR)
1.0	19/12/2019	Final version ready to upload, Submission to the EC	Jorge Almela (BVR), Zivkovic (UNIKL)

Executive Summary

The present document is the deliverable D3.7 “VICINITY core components continuous upgrades, final version” of the VICINITY [1] project, funded by the European Commission’s Directorate-General for Research and Innovation (DG RTD), under its Horizon 2020 Research and Innovation Programme (H2020). The deliverable D3.7 is a part of the WP3 VICINITY server implementation.

D3.7 presents the implementation status of the VICINITY Core components, which were defined in D1.6 VICINITY Architectural Design [2], as part of the VICINITY architecture. The VICINITY Core components consist of the VICINITY Neighbourhood Manager, the VICINITY Communication Server, the VICINITY Semantic discovery and dynamic configuration agent platform, the VICINITY Open Gateway API, the VICINITY Gateway API Services, and the Distributed Query Client including VICINITY Ontology (**Fehler! Verweisquelle konnte nicht gefunden werden.**).

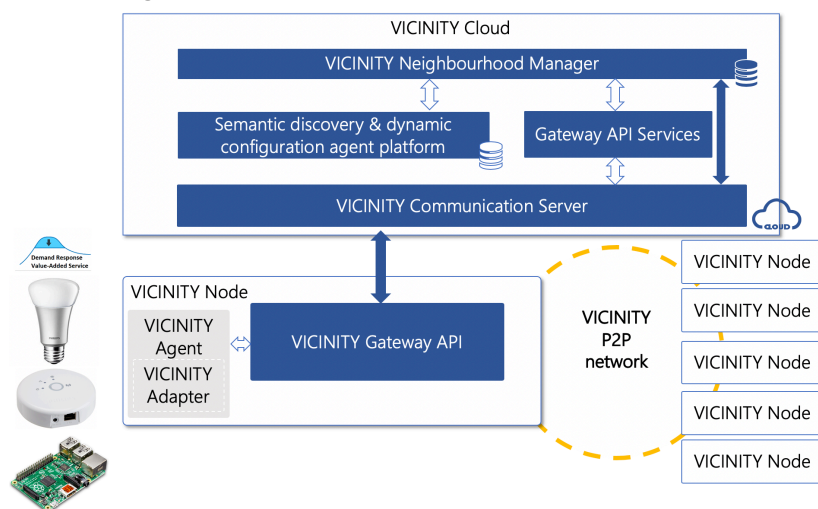


Figure 1 VICINITY Core components

The VICINITY Core components provide interoperability as a service to IoT infrastructures and services connected through VICINITY Agent(s) and VICINITY Adapter(s). These core components enable to register, manage and discover devices and services in the platform, query values from devices based on semantic request and exchange data in P2P network.

This deliverable contains the upgrades to the VICINITY Core components that happened during the evaluation phase of the project. All components received some update over this period of time, as it is described in more detail in section 2, and all these changes can be clustered in three groups:

- New features, these are the completely new functionalities added;
- Operations, these are the changes made to the infrastructure, documentation or code lifecycle management;
- Updates, these are changes on existing features.

D3.7 is a continuation of D3.6 VICINITY core components continuous upgrades, first version [8], deliverable that was covering updates happening during framework integration and lab testing stages.

Table of Contents

Executive Summary.....	5
1 Introduction	9
1.1 Context within VICINITY	9
1.2 Objectives in Work Package 3 and Task 3.3.....	10
2 VICINITY core components.....	11
3 Conclusions.....	13
4 References	14

List of Figures

Figure 1 VICINITY Core components	5
Figure 2 VICINITY Work package structure	9

List of Definitions & Abbreviations

Abbreviation	Definition
API	Application Programming Interface
DG RTD	Directorate-General for Research and Innovation
EC	European Commission
JSON	JavaScript Object Notation
P2P	Peer-to-peer
WP	Work package
NPM	Node JS package manager
Node JS	JavaScript development environment
SSH	Protocol to access remote servers
Jenkins	Open source automation server
CI/CD	Continuous integration and continuous deployment
SPARQL	Semantic query language for databases

1 Introduction

This deliverable describes the status of upgrades after the evaluation phase of the project. The core components covered are the VICINITY Neighbourhood Manager, VICINITY Communication Server, VICINITY Open Gateway API, Gateway API Services and Distributed Query Client. The deliverable itself is divided in the following sections:

- Section 1: this section reflects the deliverable in the context of the VICINITY Project’s deliverables and objectives;
- Section 2: describes current status of implementation of the VICINITY core components comparing to previous version defined by the D3.2, D3.3, D3.5 and D3.6 deliverables;
- Section 3: summarizes the presented results for this last evaluation period of the VICINITY Project.

1.1 Context within VICINITY

The D3.7 VICINITY core components continuous updates final version is part of WP3, Server implementation work package (Figure 2). The D3.7 is derived from the 3 main deliverables D1.5 VICINITY technical requirements specification, D1.6 VICINITY architecture design and D2.1 Analysis of Standardisation Context and Recommendations for Standards Involvement and build ontop of the following deliverables:

- D3.1 High-available VICINITY server deployment,
- D3.2 Web-based VICINITY neighbourhood manager,
- D3.3 Open Interoperability Gateway API,
- D3.5 Semantic discovery and dynamic configuration services.
- D3.6 VICINITY core components continuous updates first version

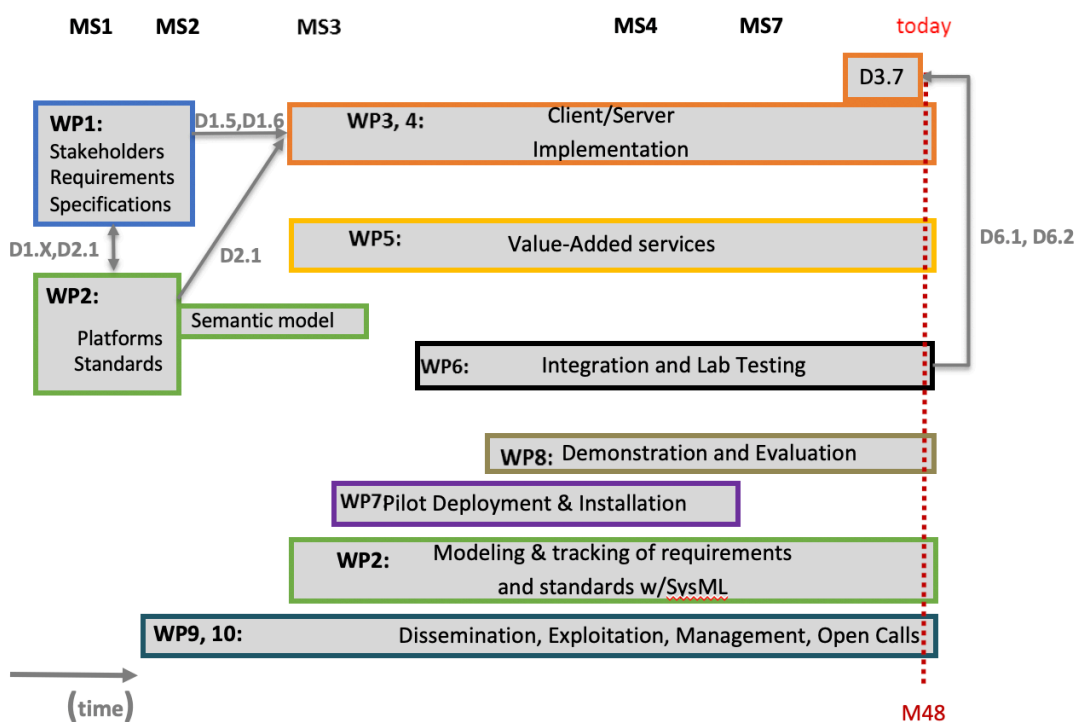


Figure 2 VICINITY Work package structure

This document describes the updates and new functionalities of the VICINITY Core components, for a more detailed description of the functionalities the reader can refer to four of the release notes cited above: D3.1 High-available VICINITY server deployment [4], D3.2 Web-based VICINITY neighbourhood manager [5], D3.3 Open Interoperability Gateway API [6], D3.5 Semantic discovery and dynamic configuration services [7]. Moreover, the D3.7 was influenced by the results of the D6.1 VICINITY Integrated prototype [9] and D6.2 VICINITY test-bed deployment, including Validation, Parametrization and Testing [10].

1.2 Objectives in Work Package 3 and Task 3.3

Objectives of Work Package 3 are as follows:

- Objective 3.1 Open gateway of semantic interoperability connected to the internet;
- Objective 3.2 Web-based Neighbourhood Manager connected to the internet;
- Objective 3.3 VICINITY semantic discovery and dynamic configuration features available;
- Objective 3.4 Operate VICINITY Core components.

These objectives were fulfilled by the deliverables D3.1 – D3.5. The follow-up objective of the task 3.3 is to operate these core components during integration, lab testing and platform evaluation. The components are continuously updated and monitored on the infrastructure level with close to 100% availability.

2 VICINITY core components

The VICINITY core components are publicly available in VICINITY H2020 GitHub with configuration and installation documentation including source code changes with “D3.7” tag and accessible through (<https://vicinityh2020.github.io>):

- VICINITY Neighbourhood Manager User Interface:https://github.com/vicinityh2020/neighbourhood_manager_ui;
- VICINITY Neighbourhood Manager Server:https://github.com/vicinityh2020/neighbourhood_manager_server;
- VICINITY Neighbourhood Manager API: <https://vicinityh2020.github.io/vicinity-neighbourhood-manager-api/>
- VICINITY Communication Server:<https://github.com/vicinityh2020/vicinity-communication-server>
- VICINITY Open Gateway API:<https://github.com/vicinityh2020/vicinity-gateway-api>;
- VICINITY Open Gateway API specification:<https://vicinityh2020.github.io/vicinity-gateway-api/>;
- VICINITY Semantic discovery and dynamic configuration agent platform : <https://github.com/vicinityh2020/vicinity-semantic-platform>
- VICINITY Ontology: <http://vicinity.iot.linkeddata.es/vicinity/>
- Gateway API Services: <https://github.com/vicinityh2020/vicinity-gateway-api-services>
- Distributed Query Client: <https://github.com/vicinityh2020/vicinity-open-gateway-api-distributed-query-client>.

The following functionalities were implemented:

- VICINITY Neighbourhood Manager – User Interface:
 - Design updates:
 - New appearance for Devices and Contracts sections.
 - Add list view to Services, Devices and Organizations sections.
 - Operations:
 - Added Docker file for containerized deployment.
 - Jira service desk integration.
 - Deployment workflow automated with Jenkins CI/CD.
 - New features:
 - Added Counters section with statistics of the messages sent by each gateway.
 - New button in the Access Point section to add an SSH key for Gateway-Neighbourhood Manager integration.
 - Create new landing page with pricing and “How to get started”.
- VICINITY Neighbourhood Manager – Server:
 - Updates:
 - Ontology mapping updated with new properties.
 - Update server connection to Mongo replica set and implement error recovery mechanism.
 - Update password recovery flow to increase security.
 - Operations:
 - Added Docker file for containerized deployment.
 - Contracts module converted into an NPM package¹.
 - Server architecture updated to High Availability
 - Storage upgraded to Replica Set mode in order to achieve High Availability.
 - Deployment workflow automated with Jenkins CI/CD.
 - Added server and application monitoring with Elasticsearch and Kibana.
 - New features:
 - New Access Points, Organisations and Contracts store their identity also in the Semantic Repository.
 - Added feature to auto refresh user tokens when there is a user role change.

¹ <https://www.npmjs.com/package/sharq-contracts>

- Added custom server timeout.
 - Added message counter logic. Now server can process the messages received from the Gateways and generate analysis and aggregations.
- VICINITY Communication Server:
 - Operations:
 - Deployment in High Availability mode of the communication server.
 - Increased limit of concurrent connections.
- VICINITY Open Gateway API:
 - New features:
 - Added data persistence between restarts. Store state of events, actions and logged devices.
 - Implementation of semantic search endpoint.
 - Implement endpoint to receive all visible objects including thing description.
 - Implementation of message counters. Each Gateway API sends logs of all messages sent and received to the platform storage.
 - Operations:
 - Added API swagger documentation.
 - Added Docker file for containerized deployment. The Gateway API also has an official image in the DockerHub public repository.²
- VICINITY Semantic discovery and dynamic configuration platform:
 - Updates:
 - Improved SPARQL endpoint and JSON serialization.
 - Improved automatic generation of semantic models to fully support semantic interoperability services.
 - Continual upgrades to new versions of underlying GraphDB repository.
 - New features:
 - Added support for semantic modelling of Thing static properties.
 - Added support for semantic modelling of Thing location metadata.
- VICINITY Distributed Query Client:
 - Updates:
 - Integration with new versions of Jena library to prevent security errors
 - Core refactoring, improving its reading, and future enhancements
 - Minor bugs fixing
- VICINITY Gateway API Services:
 - Updates:
 - Integration with new versions of Jena library to prevent security errors
 - Core refactoring, improving its reading, and future enhancements
 - Minor bugs fixing
 - Operations:
 - Improve java doc
- VICINITY Ontology: <http://vicinity.iot.linkeddata.es/vicinity/>:
 - Updates:
 - The ontology network repositories have been updated with a new artefact, more precisely, the tests needed for validation, however the ontologies code has not been modified.
 - The adapters ontology has been updated according to VICINITY, open call projects and external projects needs and reported accordingly in D4.4.

² <https://hub.docker.com/r/bavenir/vicinity-gateway-api>

3 Conclusions

This deliverable describes the updates of the VICINITY core components during the evaluation phase of the project. In this period the main focus was on finalizing the last features within controlled pilot environments, and enhancing the infrastructure in terms of availability, security and privacy, these last two covered by the D6.4, VICINITY security and privacy evaluation report [11]. Therefore, the most important updates were on these topics:

- Update the platform infrastructure to offer high availability. To achieve this objective there are now two load balanced server instances and the storage is deployed in replica set mode.
- New additions to the core ontology to meet all the requirements of the pilot sites and Open Calls.
- Adding persistence layer and counters on the Gateway API.
- Completing the semantic interoperability services and related features.

This is the last deliverable about VICINITY core components updates, therefore no significant changes or new features are expected until the end of the project.

4 References

- [1] <http://www.vicinity-h2020.eu>
- [2] D1.6 VICINITY Architectural Design
- [3] D1.5 VICINITY technical requirements specification
- [4] D3.1 High-available VICINITY server deployment
- [5] D3.2 Web-based VICINITY Neighbourhood Manager
- [6] D3.3 Open Interoperability Gateway API
- [7] D3.5 Semantic discovery and dynamic configuration services
- [8] D3.6 VICINITY core components continuous upgrades, first version
- [9] D6.1 VICINITY Integrated prototype
- [10] D6.2 VICINITY test-bed deployment, including Validation, Parametrization and Testing
- [11] D6.4 VICINITY security and privacy evaluation report