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 D4.5

VICINITY Client Components Continuous Upgrades

Work Package:	WP4 VICINITY Client Infrastructures Implementation
Task(s):	VICINITY Clients Operation and Continuous Upgrades
Lead Beneficiary:	UNIKL
Due Date:	31. December 2019 (M48)
Submission Date:	18. December 2019 (M48)
Deliverable Status:	Final
Deliverable Type:	R and DEM
Dissemination Level:	PU
File Name:	VICINITY_D4.5_DeliverableVICINITY_Client_Components_Continuous_Upgrades_v1.0.docx

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.

Authors List

Leading Author (Editor)				
Surname		**First Name**	**Beneficiary**	**Contact email**
Heinz		Christopher	UNIKL	heinz@cs.uni-kl.de
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

Reviewers List

List of Reviewers (in alphabetic order)				
No	**Surname**	**First Name**	**Beneficiary**	**Contact email**
1.	Čolić	Nikolaj	GRN	Nikolaj.Colic@gorenje.com

Revision Control

Version	Date	Status	Modifications made by
0.1	14 November 2019	Initial Draft, updated from last submission	Heinz (UNIKL)
0.2	19. November 2019	Inputs by UPM and IS integrated	Heinz (UNIKL), Kostelnik (IS), Poveda-Villalón (UPM)
0.3	25. November 2019	requested inputs have been integrated	Almela Miralles (BVR), Cimmino (UPM), Heinz (UNIKL)
0.5	28. November 2019	Ready for Review	
1.0	03. December 2019	Added Reviewer Comments	Heinz (UNIKL), Čolić (GRN)
1.0	18. December 2019	Submission to EC	Zivkovic (UNIKL)

Executive Summary

This deliverable summarizes the outcomes of task T4.4 VICINITY Clients Operation and Continuous Upgrades of the VICINITY project (Grant Agreement No.: 688467), 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 D4.5 is a part of the WP4 VICINITY Client Infrastructures Implementation. This document is the second and final of two releases under task T4.4. First version of this deliverable was released in M36, the second release is delivered in M48.

From architectural point of view, D4.5 covers actual implementation status of VICINITY Client Infrastructure Components. The VICINITY Client Infrastructure Components consist of VICINITY Gateway Adapters (T4.1), VICINITY Agent and Auto-Discovery platform (T4.2) and VICINITY Security Services (T4.3).

The VICINITY Client Infrastructure Components were updated within VICINITY Project development and integration activities, mostly as part of integration in lab testing and evaluation of pilots as well as due to feedback received from e.g. the Open Call 1 and 2 Projects.

The VICINITY Gateway API has seen two new minor releases during the last reporting period, recently introducing Client-side authentication as well as validation in the Neighbourhood Manager, apart from on-the-fly bug fixes, which were implemented as direct response to reported issues.

The VICINITY Agent and Auto-Discovery platform and the VICINITY adapters ontology both received updates in the form of bug-fixes and implemented feature requests again in direct response to issues raised by external and consortium partners.

This deliverable is accompanying document to all living documentation held in their corresponding GitHub repositories. For each component, links to GitHub documentation will be provided.

Table of Contents

List of Figures

List of Definitions & Abbreviations

Abbreviation	Definition
API	Application Programming Interface
IoT	Internet of Things
JSON	JavaScript Object Notation
P2P	Peer to Peer
SSH	Secure Shell
URI	Uniform Resource Identifier

1. Introduction

This deliverable describes the status of implementation of VICINITY Gateway Adapters (T4.1), VICINITY Agent and Auto-Discovery platform (T4.2) and VICINITY Security Services (T4.3).

1.1. Context within VICINITY

Deliverable D4.4 is the overview of continuous upgrades in all WP4 tasks. Work documented in this deliverable follows the requirements specified in WP1, covers the continuous upgrade of semantic models delivered within WP2 and was designed and implemented with very tight cooperation with the VICINITY core components delivered by WP3.

Important is the mutual influence with work in WP5 (Value-Added Services), WP6 (Integration and Testing) and WP7 (Pilot Deployment and Installations). Output of this deliverable was driven by work in this work packages and is further important input into all three mentioned work packages. Many requirements that arose during the progress in these work packages influenced (and will further influence) the design, implementation and functionality of client infrastructure components. The relationships between work packages are illustrated in Figure 1.

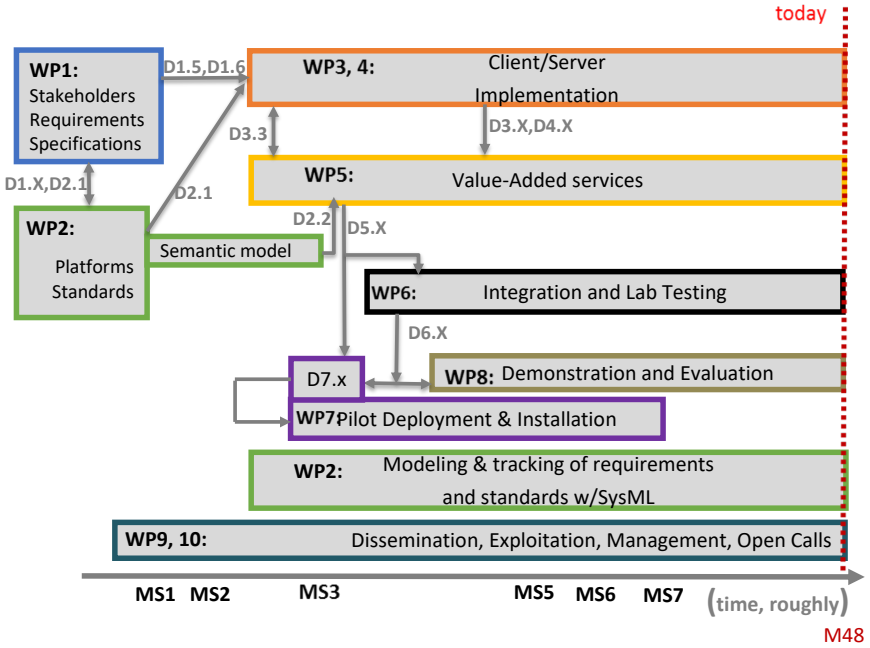

Figure 1: Relationships between work packages

Relation to other deliverables:

- D1.5 VICINITY technical requirements specification summarized the basic requirements identified for VICINITY Client Infrastructure Components
- D1.6 VICINITY architectural design describes the position of VICINITY Client Infrastructure Components within the VICINITY architecture
- D4.1 Set of open sample VICINITY gateway adapters describes the architecture and implementation of Adapters, work in this deliverable is the update of deliverable D4.1
- D4.2 VICINITY Agent and Auto-Discovery Platform describes the core of client node design and implementation.

- D3.4 Open Interoperability, Gateway API describes the VICINITY Gateway implementation, which serves as the proxy to VICINITY cloud and P2P network, updates described in this deliverable were tightly coordinated with updates of Gateway API component
- D2.2 Detailed Specification of the Semantic Model describes the VICINITY ontology, updates of VICINITY ontologies, especially the update of adapters ontology is part of this deliverable
- D3.5 VICINITY semantic discovery and dynamic configuration services describes the VICINITY server-side twin, updates documented in this deliverable was closely coordinated with updates of D3.5
- D3.6 VICINITY core components continuous upgrades were coordinated with updates in this deliverable
- There is mutual influence of work in this deliverable and deliverables D5.2 VICINITY value-added services implementation framework, D6.1 VICINITY integrated prototype, D6.2 VICINITY test-bed deployment, including Validation, Parameterization and Testing and D6.3 Auto-Discovery space deployment validation report.

1.2. Objectives in Work Package WP4 and Task 4.4

The goal of WP4 - VICINITY Client Infrastructures Implementation is to provide an easy-to-use framework for integration of IoT infrastructures. Objectives of work package 4 are as follows:

- Objective 4.1: A framework implemented for designing a general VICINITY Gateway Adapter,
- Objective 4.2: VICINITY Gateway Adapters implemented for specific IoT infrastructures needed in pilot applications and serving as samples at the implementation of further adapters,
- Objective 4.3: VICINITY Auto-Discovery platform implemented and discovering standard IoT assets,
- Objective 4.4: Agents for pilot applications implemented and configured using Auto-Discovery platform,
- Objective 4.5: VICINITY Security Services implemented

The objectives were covered in WP4 tasks:

- T4.1 VICINITY Gateway Adapters covers objectives 4.1 and 4.2. The set of VICINITY Gateway Adapters are implemented, deployed and running, new Gateway Adapters are continually being developed within work in WP5 Value-Added Services Implementation and WP7 On-site Deployment and Pilot Installations
- T4.2 VICINITY Agent and Auto-Discovery platform covers objectives 4.3 and 4.4. Agent and Auto-Discovery Platform is implemented, deployed and running on multiple VICINITY nodes
- T4.3 VICINITY Security Services covers objective 4.5. Security services were successfully integrated into VICINITY Client Infrastructure Components

Objective of task T4.4 is to manage the upgrades of components delivered in all WP4 tasks.

2. VICINITY Client Infrastructure Components Updates

This section contains a summary of the most important improvements covered by this deliverable with links pointing to living documentation on VICINITY GitHub

- VICINITY Gateway Adapters: for each adapter implementation, there exist the separate project on GitHub.
  - All available adapters can be found on the VICINITY GitHub Group at https://github.com/vicinityh2020
  - Additionally, they are listed in the IoT Catalogue at https://www.iot-catalogue.com/
- VICINITY Agent and Auto-Discovery platform is continually updated and all changes are on-the-fly documented on Agent Service Project on GitHub.
  - Full VICINITY Agent and Auto-Discovery platform documentation is available at: https://github.com/vicinityh2020/vicinity-agent
  - All changes are continually tracked per each particular platform release at: https://github.com/vicinityh2020/vicinity-agent/blob/master/docs/AGENT_CHANGELOG.md
- VICINITY Gateway API is continually updated and all changes are on-the-fly documented on respective Project on GitHub:
  https://github.com/vicinityh2020/vicinity-gateway-api
- VICINITY adapter ontology updates are part of this deliverable, the changelog of adapter ontologies is available at:
  https://github.com/mariapoveda/vicinity-ontology-adapters
  As well as updates to the VICINITY semantic services at:
  https://github.com/vicinityh2020/vicinity-semantic-platform

Summary of most important improvements covered by this deliverable:

- VICINITY Gateway Adapters: adapters are continuously developed
- VICINITY Agent and Auto-Discovery platform:
  - added static properties to Things Descriptions (value of static property is returned directly by Agent without any interaction with Thing)
  - added Thing location semantic meta-data into Thing Descriptions
  - added SPARQL endpoint into VICINITY Agent
  - improved Agent interaction interface to manage request timeouts
  - improved Thing Descriptions to fully support semantic interoperability services
  - All bug fixes and continual improvements are tracked in Agent Service Changelog (see link above)
- VICINITY Security Services: Security Features were implemented likewise in components at the user-end and in the Neighbourhood Manager. Respective changes are:
  - VICINITY Gateway API:
    - Authenticate messages sent to NM. Messages carry a token that is signed with a private SSH key in the VICINITY Gateway. Afterwards identity is validated on the NM server by using the SSH public key (This key is uploaded by the System Integrator responsible for each Gateway)
  - Neighbourhood Manager:
    - Implemented middleware that validates the identity of the messages sent by the VICINITY Gateways. Uses SSH public key uploaded by each Gateway owner via NM user interface to validate the token signature.

        - Implemented new recovery password workflow. Randomises the recovery link URI that user receives when trying to recover the password.
        - New feature added to Access Points view. Enables users to upload a public SSH key to the access point identity assigned to their gateway.
- VICINITY Gateway API distributed query client:
    - this java library is continually updated following the requests on GitHub by means of its issue tracker. Most precisely activities related to this library focused on ensuring the completeness and correctness when answering SPARQL queries issued to the Gateway API.
    - VICINITY Gateway API Services: this cloud service is continually updated following the requests on GitHub by means of its issue tracker. Most precisely activities related to this core component focused on its efficiency and effectiveness, but up to the moment they fulfilled the users requirements.
- VICINITY adapters ontology: semantic models are continually updated following the requirements of adapter developers. More precisely the following activities have been carried out:
    - continuous update of the adapter's ontology according to VICINIY needs: add illuminance, carbon dioxide, solar radiation related properties.
    - Fixed bugs and typos
    - Continuous updates of the adapter's ontology according to open call projects and external developments covering fitness (smart watch, smart scale, blood pressure tracker, glucometer) domain, specific battery related properties and water pollution.
    - Figure 2 depicts the issues providing new requirements and reporting bugs that have been implemented or fixed in the latest version of the ontology. As it can be observed, the ontology is supporting a number of different projects indicated by labels attached to each issue title. Such labels include the name of the project for which the issue was defined.

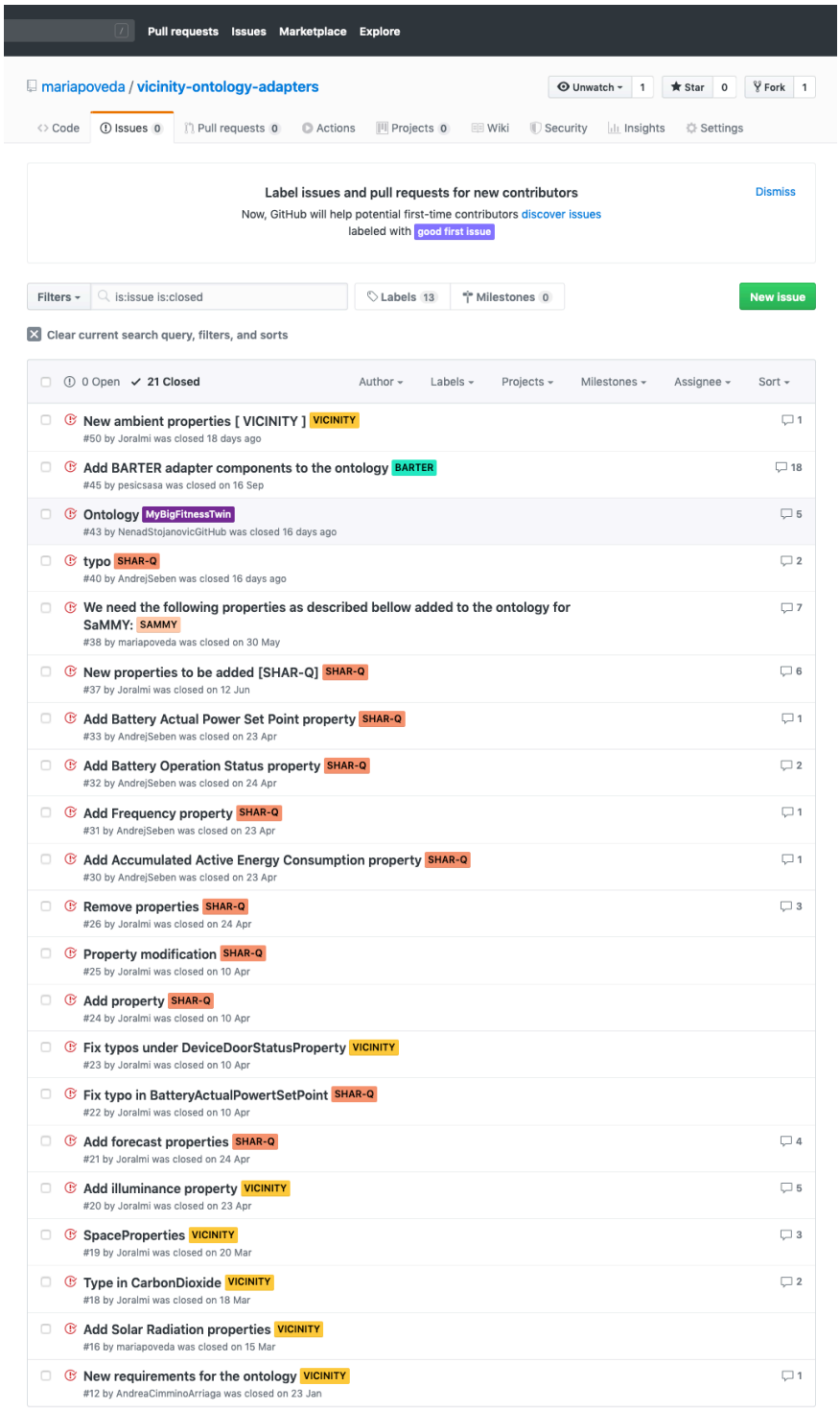

Figure 2: VICINITY adapters ontology issues

Conclusions

This deliverable described the updates of the VICINITY Client Infrastructure components during integration lab testing and evaluation of pilots as well as due to feedback received from e.g. the Open Call 1 and 2 Projects. The Client Infrastructure Components were put under testing during integration which resulted in numerous bug-fixed and new features being implemented as seen necessary for pilot integration and Open Call Projects.

The Client Infrastructure Components were subject to the VICINITY Platform evaluation to evaluate its technical functional and non-functional properties in controlled piloting environment both in the existing pilot sites as well as newly found pilot environments of the Open Call winners. This has led to upgrades of the respective components and as a result, VICINITYs Client Infrastructure is not only tailored to a limited number of use-cases but has now adapted to a variety of applications.