



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 D3.4

Open Interoperability Gateway API, Final Version

Work Package:	WP3 – VICINITY Server Implementation
Task(s):	T3.1- VICINITY core components implementation
Lead Beneficiary:	BVR
Due Date:	30 June 2018 (M30)
Submission Date:	23 July 2018 (M31)
Deliverable Status:	Final
Deliverable Type:	DEM
Dissemination Level:	PU
File Name:	VICINITY_D3.4_Open_Interoperability_Gateway_API_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	ITS	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	
Horniak	Martin	BVR	martin.horniak@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.	Oravec	Viktor	BVR	viktor.oravec@bavenir.eu
4.	Vanya	Stefan	BVR	stefan.vanya@bavenir.eu

Reviewers List

List of Reviewers (in alphabetic order)				
No	Surname	First Name	Beneficiary	Contact email
1.	Bračko	Mihael	GRN	mihael.bracko@gorenje
2.	Sundvor	Mariann	TINYM	mariann@tiny-mesh.com
3.	Koutli	Maria	CERTH	mkoutli@iti.gr

Revision Control

Version	Date	Status	Modifications made by
0.1	2. June 2018 (M30)	Initial Draft	Martin Horniak (BVR)
0.2	8. June 2018	First Draft formatted with contributions received	Jorge Almela (BVR), Andrea Cimmino (UPM), Martin Horniak (BVR), Viktor Oravec (BVR), Stefan Vanya (BVR)
0.3	15. June 2018 (M30)	Deliverable version for final review by partners	Martin Horniak (BVR)
0.4	18. June 2018 (M30)	Final improvements	Viktor Oravec (BVR)
0.5	18. June 2018 (M30)	Deliverable version uploaded for Quality Check	Martin Horniak (BVR)
0.9	20. July 2018 (M30)	Final Draft reviewed (delayed due to finalization of the VICINITY Gateway API)	Viktor Oravec (BVR), Martin Horniak (BVR), Andrea Cimmino (UPM)
1.0	23. July 2018 (M31)	Submission to the EC	Carna Zivkovic (UNIKL)

Executive Summary

The present document is the deliverable D3.4 “Open interoperability Gateway API, 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 D3.1 is a part of the WP3 VICINITY server implementation.

Regarding the architecture of the VICINITY project [2], the VICINITY Open Gateway API provides endpoints for interaction among devices/ services and acts as a gateway to XMPP based P2P network (Figure 1). It also enforces the authorisation rules defined in the VICINITY Neighbourhood database [3] and stored in the Communication Server [4] and encrypts information exchanged within P2P network.

The VICINITY Open Gateway API provides endpoints “locally” close to the connected infrastructure. However, it can be deployed close to IoT infrastructure cloud services as well. Technically, the API is standard Java based service utilizing Ignite Open Fire Spark XMPP engine, which maintains connectivity in P2P network.

The provided endpoints create HTTP based REST API interface which facilitates:

- control plane – device and service registration and configuration messages;
- user plane – measurement, actions, events and semantic distributed query messages.

The facilitated messages needs to be JSON complaint messages.

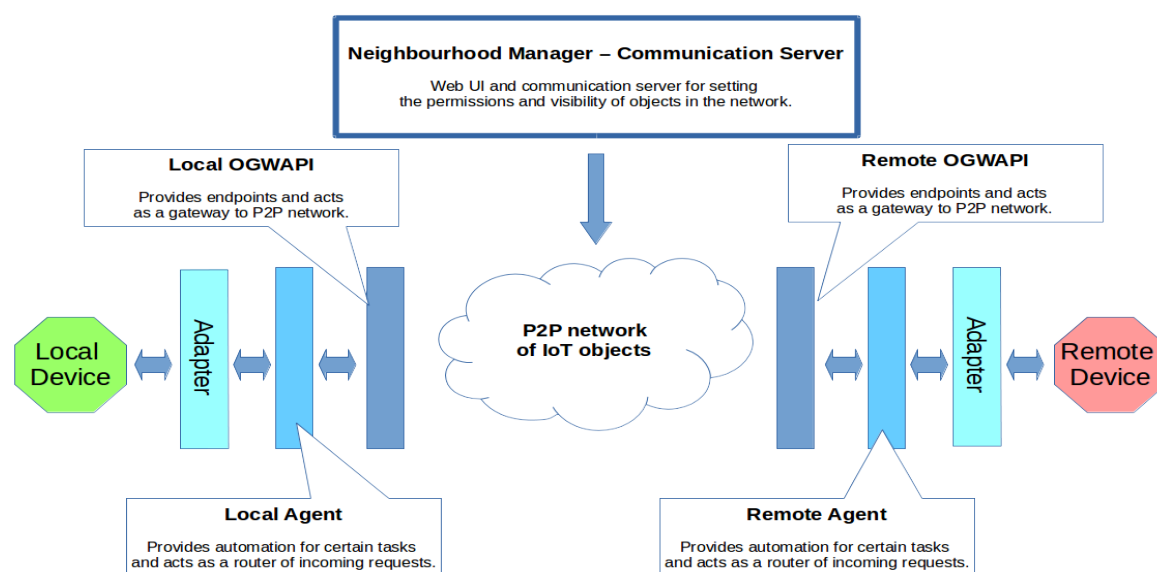


Figure 1 Emplacement of VICINITY Open Gateway API in the system.

Specific feature of the VICINITY Open Gateway API is supporting of custom semantic data queries in the P2P network. This semantic data queries utilizes Distributed Semantic Query Client and VICINITY Open Gateway API Services components which integrated with VICINITY Gateway API and VICINITY Communication Server. The queries client builds and executes distributed data query plan in P2P network based on the result of semantic search in semantic repository. This allows to retrieve measurements from different peers and devices in single query.

Table of Contents

Executive Summary	5
1 Introduction	8
1.1 Context within VICINITY	8
1.2 Objectives in Work Package 3 and Task 3.1	8
2 VICINITY Open Gateway API.....	10
3 Gateway API Services and Distributed Query Client.....	11
4 Conclusions.....	12
5 References	13

List of Figures

Figure 1 Emplacement of VICINITY Open Gateway API in the system.....	5
Figure 2 VICINITY Work package structure	8

List of Definitions & Abbreviations

Abbreviation	Definition
API	Application Programming Interface
DG RTD	Directorate-General for Research and Innovation
EC	European Commission
EU	European Union
P2P	Peer-to-peer
REST	Representational state transfer
WP	Work package
XMPP	Extensible Mark-up and Presence Protocol

1 Introduction

This deliverable describes the operation of the VICINITY Open Gateway API, Gateway API Services and Distributed Query Client the installation, deployment, configuration and integration of the software with IoT infrastructure and value-added services.

1.1 Context within VICINITY

The D3.4 Open Interoperability Gateway API final version is part of WP 3 Server implementation work package (Figure 2). The D3.4 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.

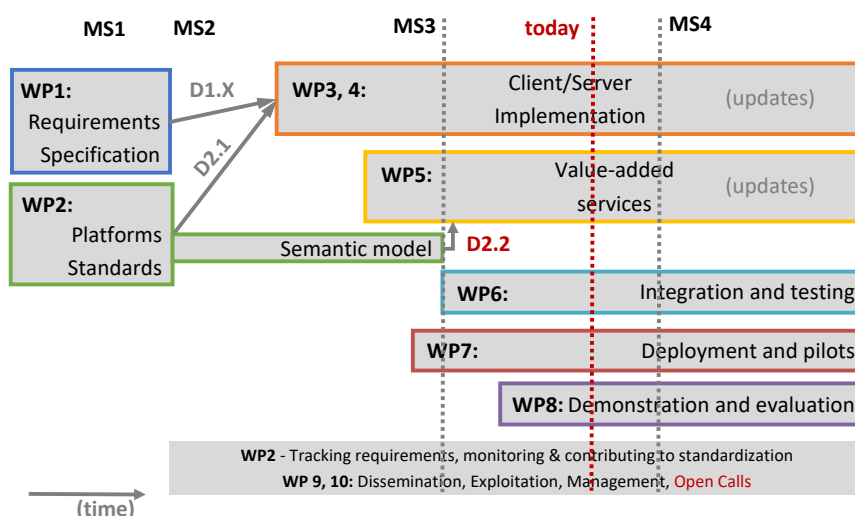


Figure 2 VICINITY Work package structure

The Open Interoperability Gateway API [D3.4 Open Interoperability Gateway API final version] is integrated with VICINITY Neighbourhood Manager [D3.2 Web-based VICINITY Neighbourhood manager], VICINITY Communication server [D3.1 High-available VICINITY server deployment] and Semantic platform [D3.5 Semantic discovery and dynamic configuration services], thus the Open Gateway API documentation assumes prior knowledge about these components as well.

1.2 Objectives in Work Package 3 and Task 3.1

The VICINITY Open Gateway API, Gateway API Services and Distributed Query Client are delivered under Task 3.1 of Work Package 3. The VICINITY Open Gateway API directly addresses objective:

- Objective 3.1 open gateway of semantic interoperability connected to the internet;

and partially addresses objective:

- Objective 3.3 advanced trust, security and privacy assuring mechanism implemented.

To achieve this objective the following key achievements were met:

- The detail design of the VICINITY Open Gateway API;
- The implementation of the services to expose and consume properties;

- The implementation of the services to expose and consume actions;
- The implementation of the services to expose and consume events;
- The implementation of the authentication and authorization mechanism for devices, services and adapters;
- The implementation of the query interface, distributed semantic querying and VICINITY Gateway API Services;
- The implementation of the registry interface to register devices and services;
- The implementation of VICINITY Open Gateway API logging mechanism.

The VICINITY Communication Server installation and configuration guide is described in <https://github.com/vicinityh2020/vicinity-communication-server>.

2 VICINITY Open Gateway API

The VICINITY Open Gateway API, Distributed semantic query client and VICINITY Gateway API service are publicly available in VICINITY H2020 GitHub with configuration, installation documentation including source code changes with D3.4 tag:

- <https://github.com/vicinityh2020/vicinity-gateway-api>;
- <https://github.com/vicinityh2020/vicinity-gateway-api-services>;
- <https://github.com/vicinityh2020/vicinity-open-gateway-api-distributed-query-client>.

The Open Gateway API repository can be downloaded from the following link:

<https://github.com/vicinityh2020/vicinity-gateway-api>

Along with the source code and latest build of the binary file, there are four sources of information that can be found in the repository regarding the software:

- Readme – it can be displayed via GitHub web interface, when user follows the link to the repository or the README.md file can be found in the root directory of downloaded repository. It provides a quick overview about how to get started with the Open Gateway API (<https://github.com/vicinityh2020/vicinity-gateway-api/blob/master/README.md>).
- Javadoc – Can be found in the ‘doc’ directory. It is aimed to help the community of advanced users in understanding how individual classes of the Java project interact and how they provide their functions (<https://github.com/vicinityh2020/vicinity-gateway-api/tree/master/doc>).
- Integration Handbook – It can be found in the ‘doc/Integrator Handbook/’ directory. This comprehensive handbook instructs the potential integrator about details of Open Gateway API behaviour, error propagation, endpoint functionality and sequences of calls that enable him/her to exploit the API functionality. The installation and configuration are described in great length and detail (<https://github.com/vicinityh2020/vicinity-gateway-api/tree/master/doc/Integrator%20handbook>).
- List of implemented endpoints and their functionality – To be found in the ‘doc/Integrator Handbook/’ directory provides quick list of the endpoints and short description of the functionality and parameters that need to be passed. It serves as a quick reference, where the endpoints are listed in ‘denser’ form, than in detailed Integrator Handbook (<https://github.com/vicinityh2020/vicinity-gateway-api/tree/master/doc/Integrator%20handbook>).

3 Gateway API Services and Distributed Query Client

The Gateway API Services and the Distributed Query Client are software applications, the former is a web service and the latter is integrated within the Gateway APIs. Together these components implement the interoperability approach of VICINITY following VICINITY Project Objectives:

- Objective 3.1 Open gateway of semantic interoperability connected to the internet
- Objective 3.3 VICINITY semantic discovery and dynamic configuration features available

The Gateway API Services repository can be downloaded from the following link:

<https://github.com/vicinityh2020/vicinity-gateway-api-services>

The Distributed Query Client library can be downloaded from the following link:

<https://github.com/vicinityh2020/vicinity-open-gateway-api-distributed-query-client>

Along the source codes and latest build of the software, there are other sources of information that can be found in both repositories regarding the software:

- Readme – displayed via the web interface directly when user follows the link to the repository or in the README.md file in the root directory of downloaded repository. This files provide a quick overview about how to get started with the Gateway API Services, as well as its service interface (<https://github.com/vicinityh2020/vicinity-gateway-api-services/blob/master/README.md>), and the Distributed Query Client (<https://github.com/vicinityh2020/vicinity-open-gateway-api-distributed-query-client/blob/master/README.md>).

In addition the Gateway API Services include a wiki page in which additional information regarding its functionality can be found (<https://github.com/vicinityh2020/vicinity-gateway-api-services/wiki>).

The Distributed Query Client includes the following files:

- Javadoc – Can be found in the ‘doc’ directory. It is aimed for the community of advanced users and to help developers in understanding of how individual classes of the Java project interact and how they provide their functions (<https://github.com/vicinityh2020/vicinity-open-gateway-api-distributed-query-client/tree/master/doc>).

4 Conclusions

This deliverable discussed the details of installation, configuration and operation principles of VICINITY Open Gateway API, Distributed Query Client Library and VICINITY Gateway API Services. In the following phase of these components will be used as follows:

VICINITY Open Gateway API will be used as semantic interface towards P2P network for integrated IoT infrastructures and services in pilot sites and VICINITY open call projects;

Distributed Query Client Library and VICINITY Gateway API Services will be used by value-added services to process custom semantic queries on semantic repository and P2P network.

All of the component will be subject of change due to fixing of bugs from integration and evaluation of VICINITY Platform, or potential extension with new relevant features.

5 References

- [1] <http://www.vicinity-h2020.eu>
- [2] D1.6 VICINITY Architectural Design
- [3] D3.2 Web-based VICINITY Neighbourhood Manager
- [4] D3.1 High-available VICINITY server deployment
- [5] D1.5 VICINITY technical requirements specification
- [6] <https://www.w3.org/TR/rdf-sparql-query/>
- [7] <https://www.w3.org/RDF/>