Automating Heterogeneous Internet of Things Device Networks from Multiple Brokers with Multiple Data Models

P. Bellini, C. Camerota, P. Nesi

Distributed Systems and Internet Technology Lab DISIT

SNAP4city

Florence, Italy

Agenda

- Introduction
- Goals
- SNAP4city
- Requirements
- Architecture
- Smart Data Model and Rules
- SNAP4city model
- Rules’ structure
- Validation experiments in SNAP4city
- Conclusion

Introduction

- An **IoT platform**:
  - bridges the gap between device sensors and data networks.
  - is a set of components that allows developers to spread out the applications, remotely collect data, secure connectivity, and execute sensor management.

- IoT Broker orchestrates the devices’ networks, which can be **Internal** (managed by the Platform) or **External** (managed by third-party)

Introduction

- A **Data Model** is a declarative framework designed for representing IoT concepts, devices, and workflows.

- We focus on the device's Data Model. In particular, on **FIWARE Smart Model:** an open-source European standard, suitable for FIWARE Orion Broker.

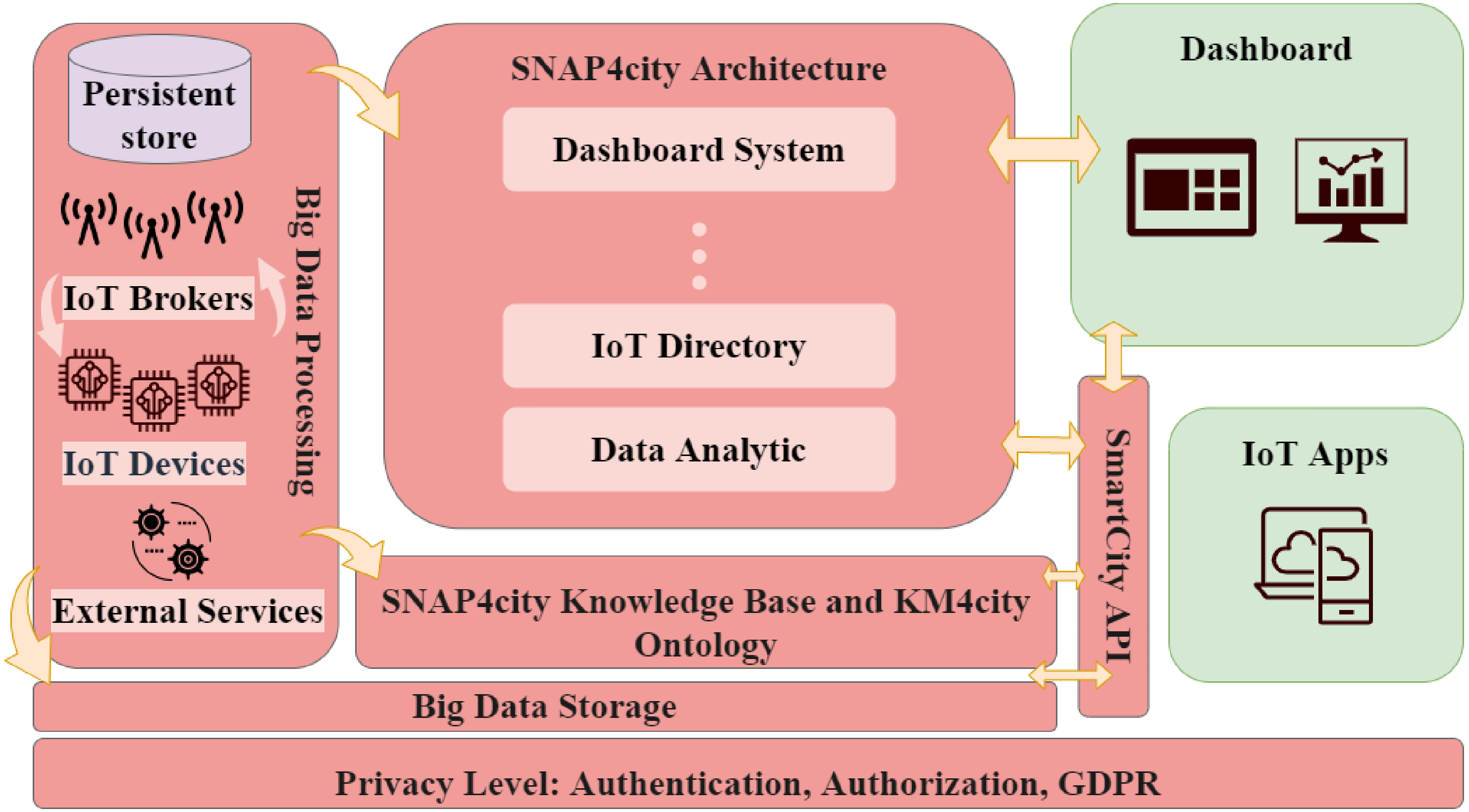
Goals

- Automatic **harvesting** of External Devices
- Automatic **ingestion** of data from External Devices in SNAP4city
- **Automatic rules generation** for FIWARE Data Model
- Rules **editing** and **management**

SNAP4city

- IoT Platform with FIWARE certification.

- Offers an **IoT supporting tool (IoT Directory)** among various features such as Dashboard builder, ETL development, etc….



SNAP4city

- **Knowledge Base**:
  - Implemented by Virtuoso on the basis of Km4City Ontologies
  - Its management allows to index IoT entities and establishes all the relationships.

Entities	# of entities
Internal IoT Broker	20
Internal IoT devices	20.073
External IoT Broker	7
External IoT devices	77
Average Devices for each External Broker	15.400
Average Devices for each Internal Broker	40.936
Distinct value_name	1530

Requirements

- Manage **different kinds** of IoT Brokers, IoT Devices and IoT Edge Devices.
- Connect **External and Internal Brokers**.
- Support **automatic deployment** of Internal IoT Brokers.
- Register **External Brokers**.
- **Discover** IoT Devices on IoT Brokers.

Requirements

- Register, manage and use messages conformant to **any Data Model with any data type**.
- Verify the **correctness** of IoT Messages of IoT Devices.
- **Semantic** Interoperability.
- **Easy management** graphic interface to list and test IoT Brokers, and IoT Devices and query them.
- Manage IoT Device Model and Device Data Type **ownership** and access grant.

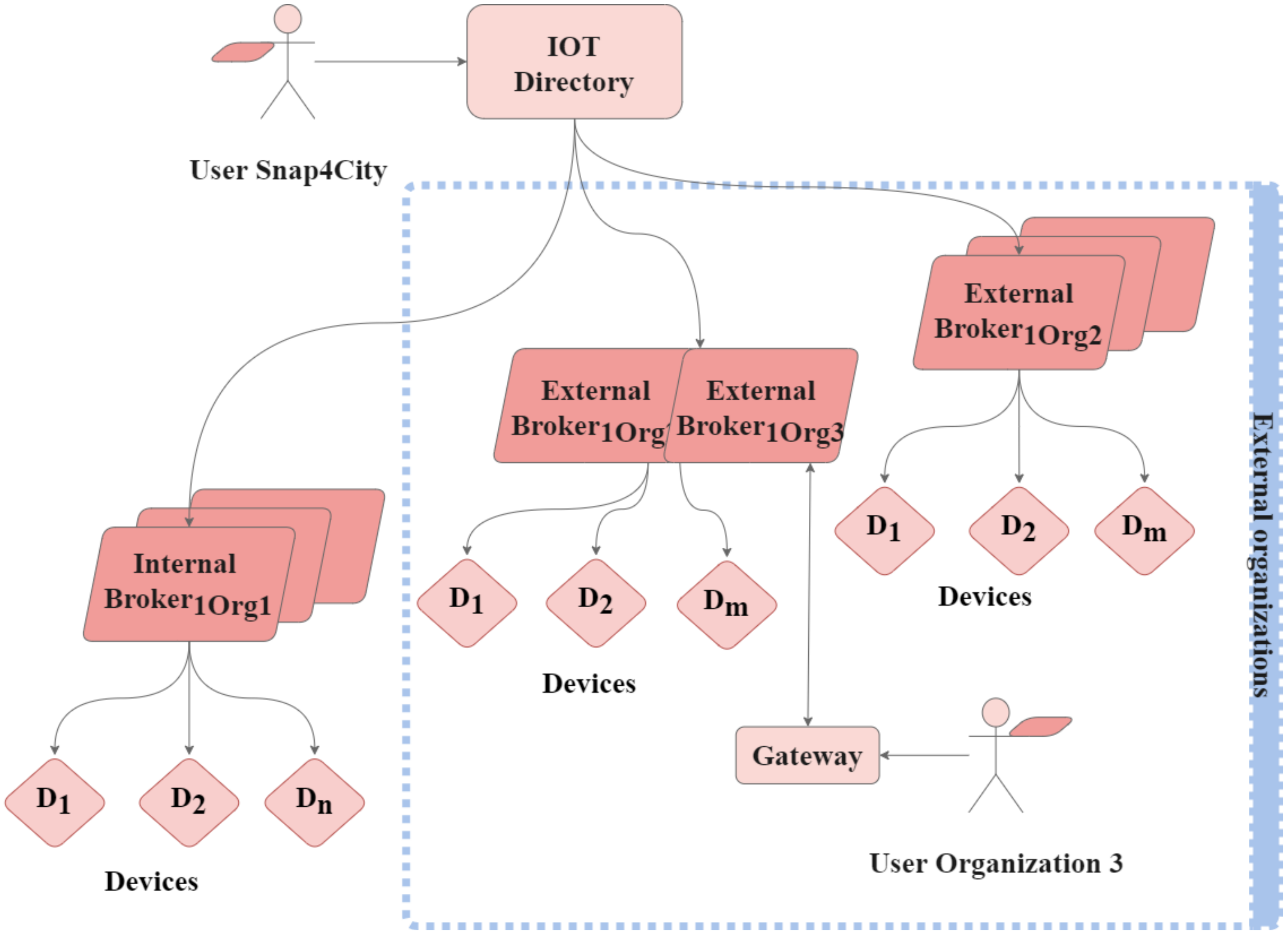
Requirements

Requirement	Snap4City	Google IoT Cloud	Azure IoT	AWS Amazon	IBM Watson	Siemens Mindsphere
Manage different kinds of IoT entities	Y	N	Y	(Y)	Y	Y
Connect External and Internal Brokers	Y	Y	Y	Y	Y	(Y)
Use any Data Model with any data type	Y	Y	(Y)	(Y)	Y	(Y)
Verify the correctness of IoT Messages of IoT Devices	Y	(Y)	(Y)	(Y)	(Y)	(Y)
Semantic Interoperability	Y	Y	Y	Y	Y	(Y)
Automatics deploy of Internal IoT Brokers	Y	N	N	N	N	Y
Register External Brokers	Y	N	N	N	N	N
Discover IoT Devices on IoT Brokers	Y	N	(Y)	N	(Y)	N
Easy management graphic interface to list and test IoT entities	Y	(Y)	(Y)	(Y)	(Y)	(Y)
Manage IoT Device Model and Device Data Type ownership and access grant	Y	Y	(Y)	Y	Y	Y

Architecture

Registration of:

- An Internal IoT Broker
- An External IoT Broker

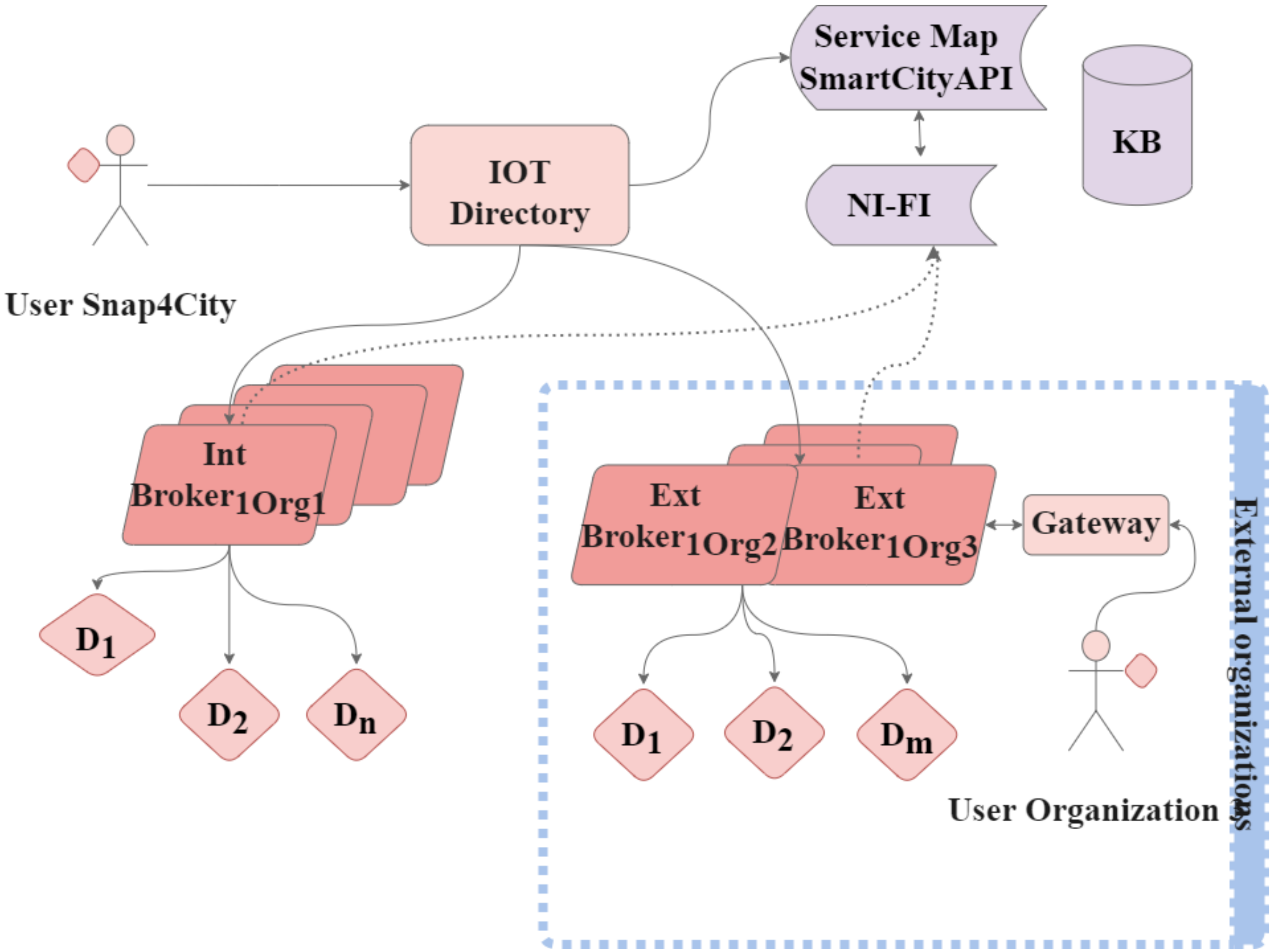


Architecture

Registration of:

- An Internal IoT Device
- An External IoT Device

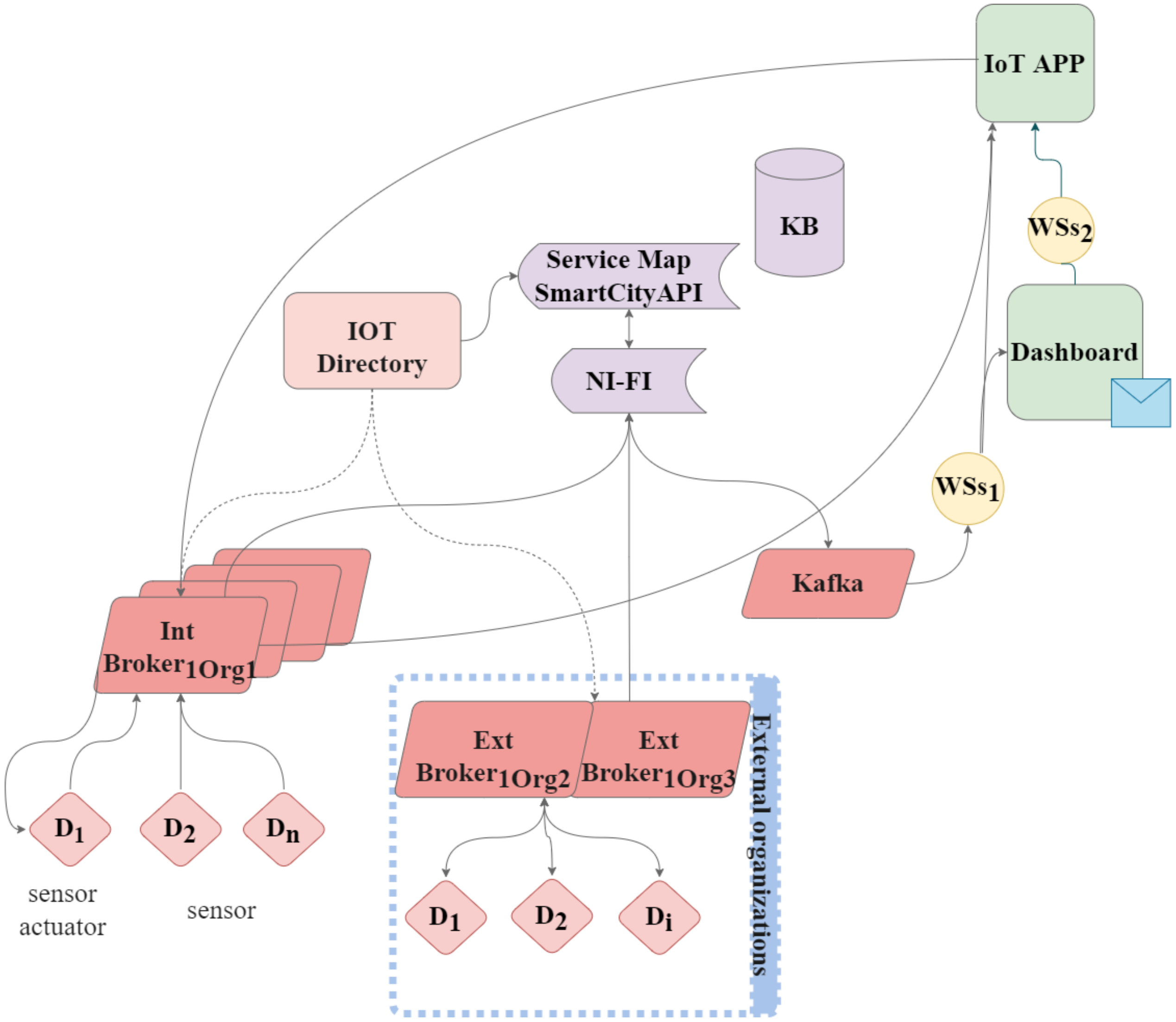
The solid lines indicate the registrations, while the dashed lines indicate the data flow of subscriptions.



Architecture

IoT **Messages exchanged** among entities.

The solid lines are data flows, dashed lines indicate the tests that the user can perform to verify the IoT Devices/Brokers.



Smart Data Model and Rules

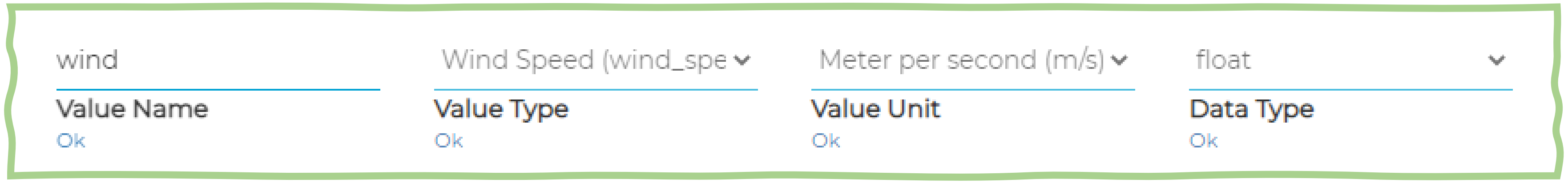
- For successful registration of a Device, its general information and all its attributes **must be compliant** with the standard of the Platform.

- Once a model is known, each Device with the same Smart Data Model could be automatically registered.

- The **rule** is the tool by which the Platform can recognize and fix the compliant error of the model.

SNAP4city model

- Each attribute is defined by:
  - **Value name** – the name of attribute
  - **Value type** – semantic definition of attribute
  - **Value unit** – the attribute’s unit of measurement (if there is)
  - **Data type** – the data type of the attribute



wind	Wind Speed (wind_spe	Meter per second (m/s)	float
Value Name	Value Type	Value Unit	Data Type
Ok	Ok	Ok	Ok

Rules’ structure

The rule is divided into two parts:

- **If statement**, where the conditions are set. A condition defines a subset of devices by operators of equivalence or not.
- **Then statement**, a collection of actions to apply. Each item declares the attribute feature that the user would change and the new value to put in.

Rules in EBNF notation

Rule:= IF <condition list> THEN <action list>

<condition list>: = <c> | <c> AND <condition list>

<c> := <variable> <op> <constant>

<variable> := “device name” | “context broker” | “device type” | “model” | “value name”

<op> := “is equal” | “is not equal” | “is null” | “contains”

<constant> := integer | float | string | list

<action list> := <a>| <a>, <action list>

<a> :=<action variable>: <action constant>

<action variable>:= “Data type” | “Value type” | “Value unit” | “Editable” |
<Coded Healthiness criteria> | <Healthiness value>

<action constant>:= string

Rules example

R:= IF <condition list> THEN <action list>

<condition list>: *= <c_1> AND <c_2>*

<c_1> := “context broker” “is equal” “TestCapelon”

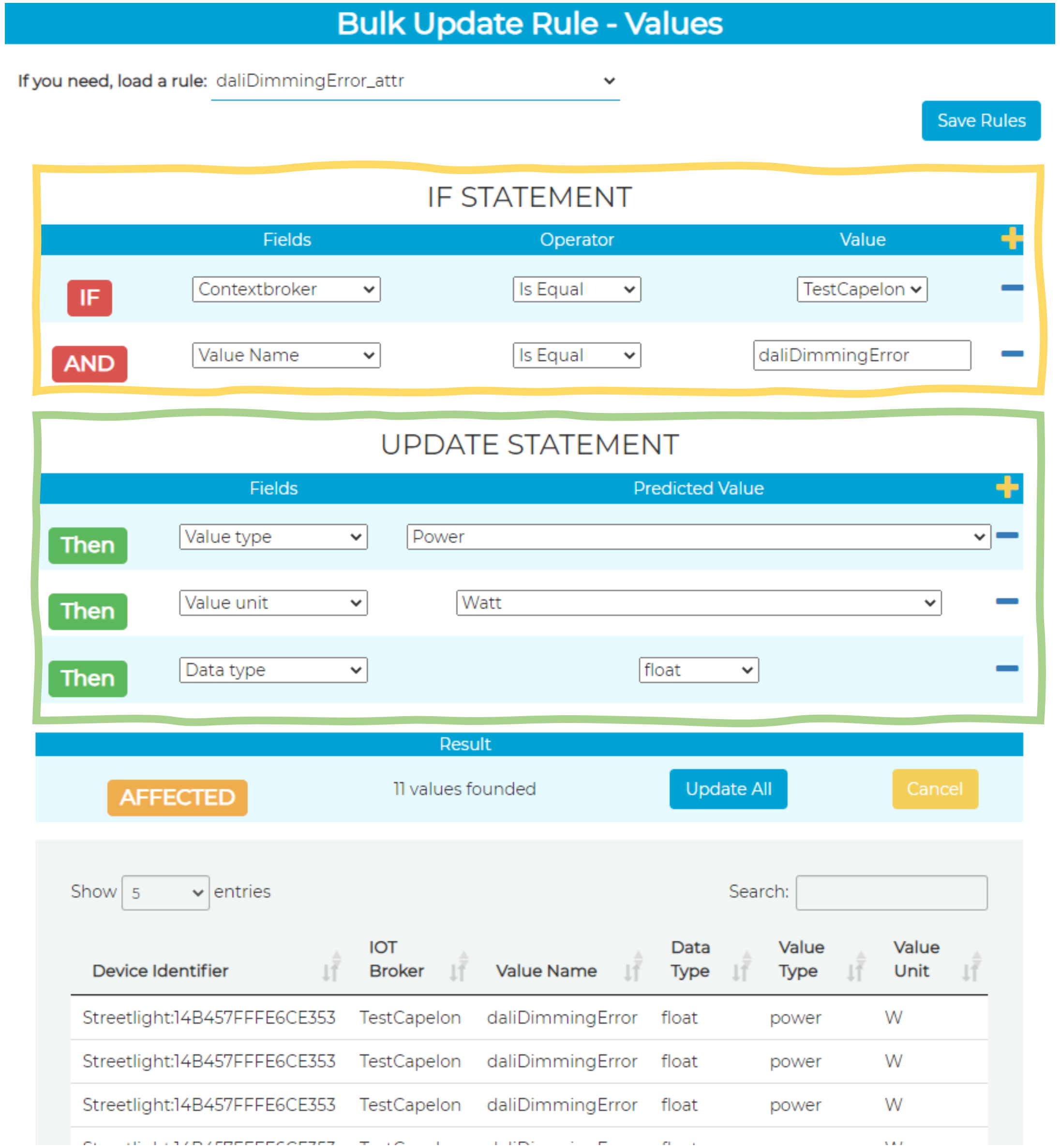
<c_1> := “value name” “is equal” “dailDimminError”

<action list> *:= <a_1>, <a_2>, <a_3>*

<a_1> := “value type”: “Power”

<a_2> := “value unit”: “Watt”

<a_3> := “data type”: “float”



Bulk Update Rule - Values

If you need, load a rule: daliDimmingError_attr

Save Rules

IF STATEMENT

	Fields	Operator	Value
IF	Contextbroker	Is Equal	TestCapelon
AND	Value Name	Is Equal	daliDimmingError

UPDATE STATEMENT

	Fields	Predicted Value
Then	Value type	Power
Then	Value unit	Watt
Then	Data type	float

Result

AFFECTED 11 values founded Update All Cancel

Show 5 entries Search:

Device Identifier	IOT Broker	Value Name	Data Type	Value Type	Value Unit
Streetlight:14B457FFFE6CE353	TestCapelon	daliDimmingError	float	power	W
Streetlight:14B457FFFE6CE353	TestCapelon	daliDimmingError	float	power	W
Streetlight:14B457FFFE6CE353	TestCapelon	daliDimmingError	float	power	W

Validation experiments in SNAP4city

1. Add new device without model and with 10 attributes – (over 10 trial)
   - Average time by human: 1′30″
   - Average time by system: 3″

2. Add new device from model, with 10 attributes – (over 10 trial)
   - Average time by human: 58″
   - Average time by system: 1.035″
   - Average time by IOT APP: 0.623″

Validation experiments in SNAP4city

3. Harvesting of an External multi-tenant broker with 3700 devices – average time: 25′ 50″ (over 10 trial)

4. Automatic ingestion of a harvesting devices with FIWARE streetlight model – average time: 0.0371″ (over 10 trial)

Conclusion

- This paper offers an **analysis and comparison among relevant existing platforms** and **delineates the basic requirements** to achieve the goals.

- SNAP4city achieves all goals and is **full open-source** and **license free**.

- The **IoT Directory** is the proposed solution to manage Internal and External Brokers, perform the automated registration by harvesting, and perform the automated registration and management of Data Models.

Thank you!

Bibliography

- Ghasempour, Alireza. "Internet of things in smart grid: Architecture, applications, services, key technologies, and challenges." Inventions 4.1 (2019): 22.
- Mekki, Kais, et al. "Overview of cellular LPWAN technologies for IoTdeployment: Sigfox, LoRaWAN, and NB-IoT." 2018 ieee international conferenceon pervasive computing and communications workshops (percom workshops).IEEE, 2018.
- Yousefpour, Ashkan, et al. "All one needs to know about fog computing andrelated edge computing paradigms: A complete survey." Journal of SystemsArchitecture 98 (2019): 289-330.
- Adelantado, Ferran, et al. "Understanding the limits of LoRaWAN." IEEECommunications magazine 55.9 (2017): 34-40.
- https://aws.amazon.com/iot
- https://siemens.mindsphere.io/en
- https://cloud.google.com/solutions/io
- https://azure.microsoft.com/en-us/overview/iot
- https://ww.ibm.com/watson
- https://www.snap4city.org
- Cheng B., G. Solmaz, et al.,"FogFlow: Easy Programming of IoT Services Over Cloud and Edges for SmartCities," in IEEE Internet of Things Journal, vol. 5, no. 2, pp. 696-707, April 2018,doi: 10.1109/JIOT.2017.2747214.
- Badii C., P. Bellini, A. Difino, P. Nesi, G. Pantaleo, M. Paolucci, "MicroServicesSuite for Smart City Applications", Sensors, MDPI, 2019. https://doi.org/10.3390/s19214798
- Badii C:, P. Bellini, A. Difino, P. Nesi, "Smart City IoT Platform RespectingGDPR Privacy and Security Aspects", IEEE Access, 2020.10.1109/ACCESS.2020.2968741
- Ammar, Mahmoud, et all. "Internet of Things: Asurvey on the security of IoT frameworks." Journal of Information Security andApplications 38 (2018): 8-27.
- Ray, Partha Pratim. "A survey of IoT cloud platforms." Future Computing andInformatics Journal 1.1-2 (2016): 35-46.
- Bellini P., D. Cenni, et al, "Smart CityControl Room Dashboards: Big Data Infrastructure, from data to decisionsupport", Journal of Visual Languages and Computing, 10.18293/VLSS2018-030
- Bellini P., D. Nesi, et al, "Federation of Smart City Services viaAPIs", Proc of 6th IEEE International Workshop on Sensors and Smart Cities, with IEEE SmartComp, 14-17 Sept. 2020, Bologna, Italy. http://ssc2020.unime.it/