FI-WARE

Developing Internet of Things Applications with FIWARE

César Batista, Pedro Victor Caldas, Everton Cavalcante

Department of Informatics and Applied Mathematics (DIMAp)
Federal University of Rio Grande do Norte (UFRN)
Natal, Brazil

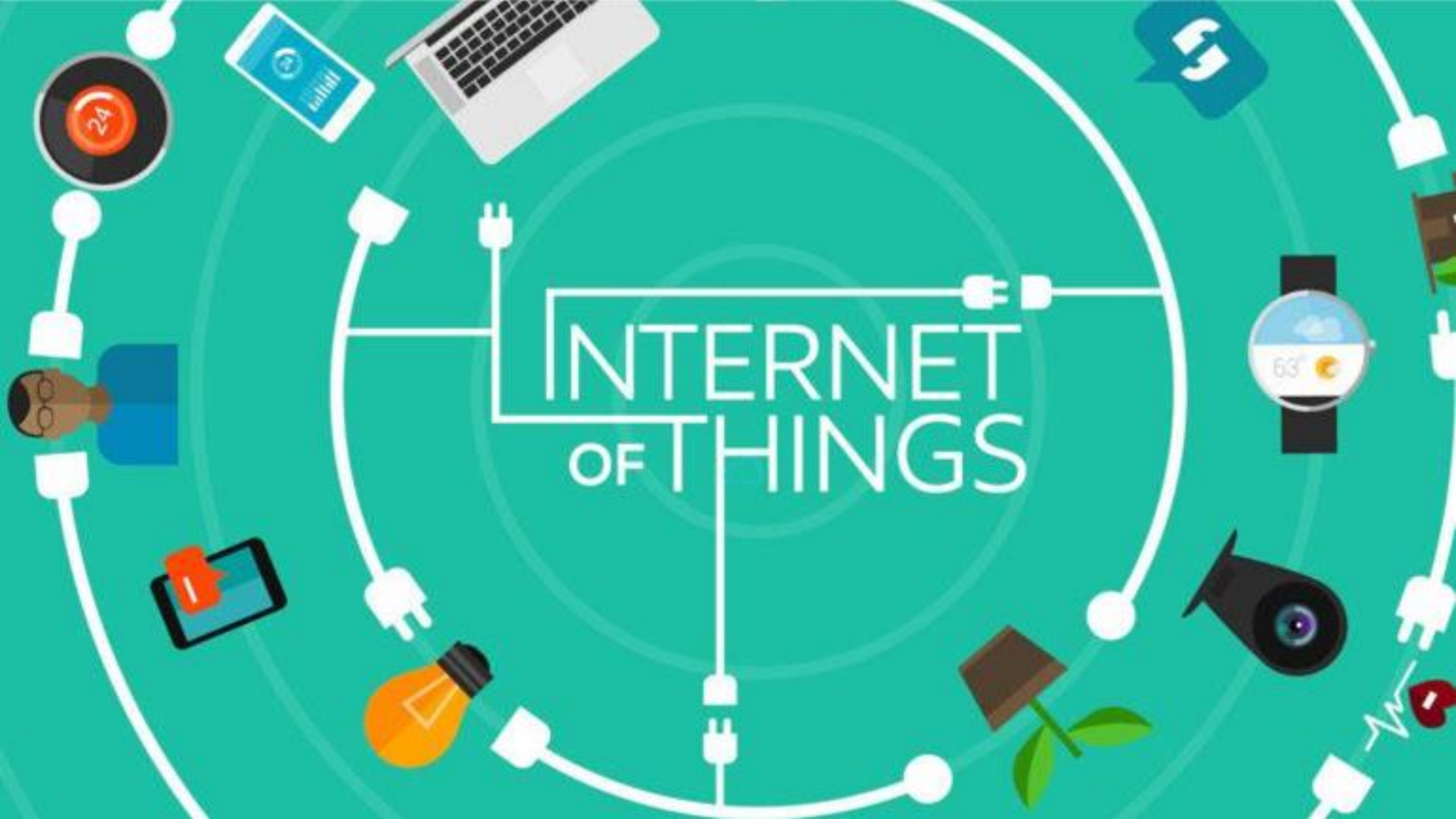
UFRN
UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

DIMAp
UFRN

SMART METROPOLIS

SETE
Software Engineering Team
Research and Innovation

IMD
INSTITUTO
METRÓPOLE DIGITAL


INTERNET
OF THINGS

Internet of Things (IoT)

- A novel computing paradigm relying on the interaction of smart objects (things) with each other and with physical and/or virtual resources through the Internet
- Existence of a diversity of devices typically endowed with sensing and/or actuation capabilities
  - Sensors capture physical phenomena and translate them into data streams to provide information about the environment where they are inserted into
  - Actuators affect the physical realm as a response to various stimuli
- Devices can be identified, controlled, and monitored through the Internet

Internet of Things (IoT)

Key elements in IoT are

- interconnecting devices and systems (and also people)
- securely collecting data at real-time from multiple sources
- providing value-added information and functionalities to end-users and applications
- making reliable decisions through data processing and analysis

IoT applications

The wide dissemination of IoT has the potential of significantly impacting people's lives in several application domains

smart cities

environment

energy

logistics

industry

domotics

retail and services

health

Concerns and challenges in IoT

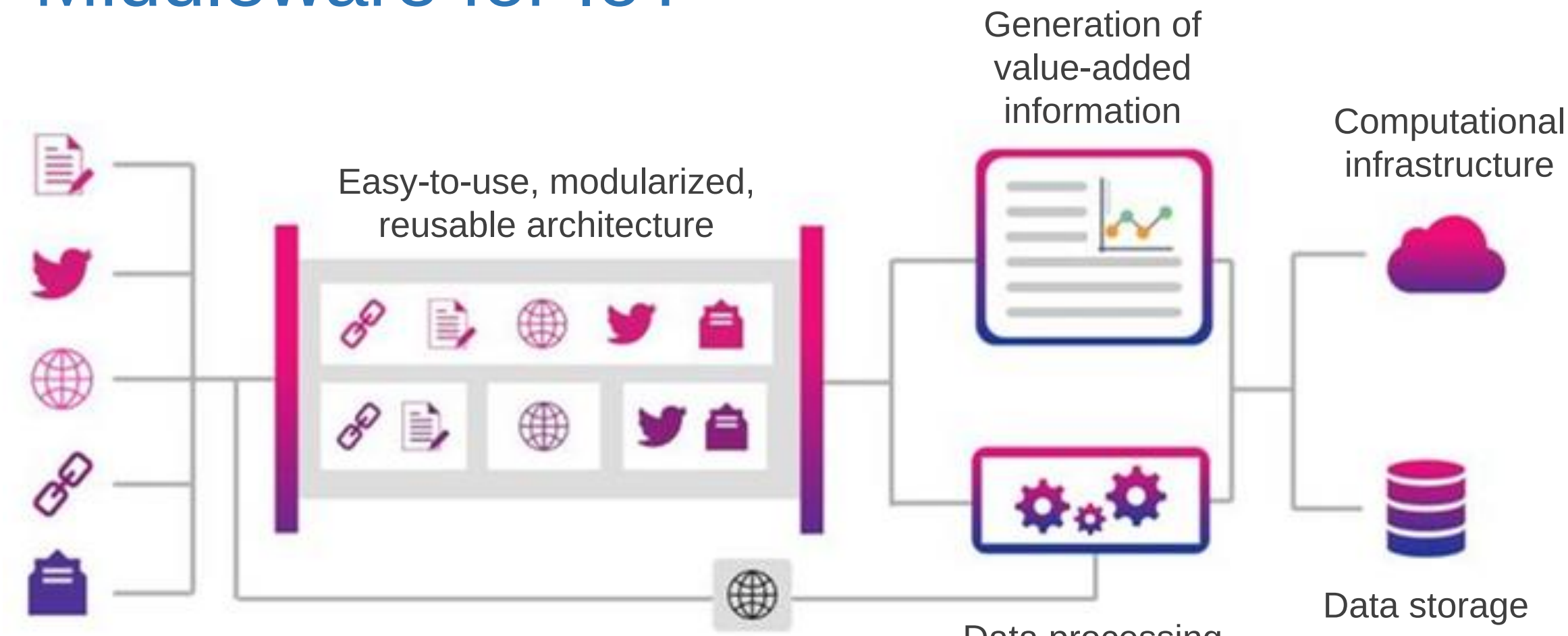
High heterogeneity

due to the diversity of physical devices in terms of

- hardware
- software
- protocols
- data formats

Middleware for IoT

Middleware for IoT

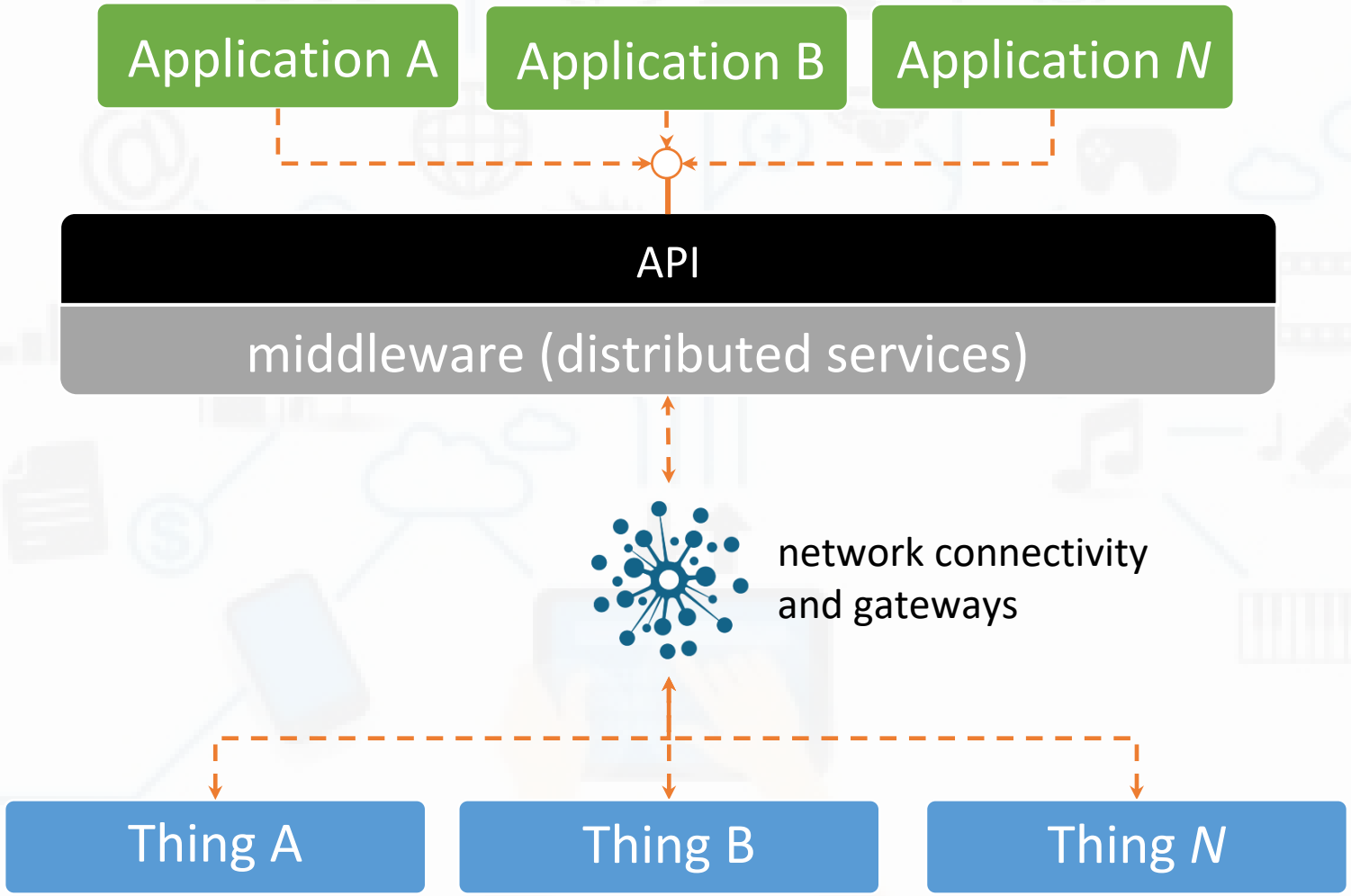


Adapted from Elixir Data - https://www.elixirdata.com/

Middleware for IoT

Promising solution to the high heterogeneity

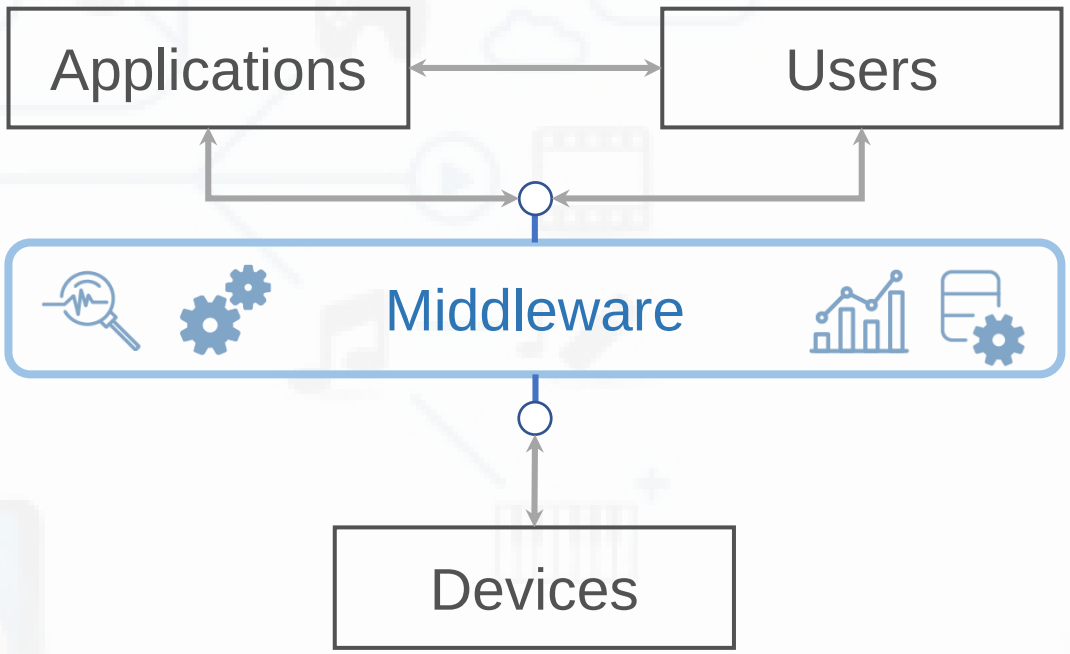
Middleware for IoT



Middleware for IoT

Middleware

- Software layer between physical nodes (sensors, actuators, things) and logical nodes (applications/users), connected through the network
- Provision of high level interfaces and common services to ease the development of applications
- Promotion of transparency in many dimensions, e.g., distribution/location, heterogeneity



Middleware for IoT

In the IoT context, middleware platforms must meet a set of requirements aimed to cope with the needs of applications and users as well as address challenges arisen in this scenario

- Interoperability
- Device discovery and management
- Context-awareness
- Scalability
- Management of large volumes of data
- Security and privacy
- Dynamic adaptation
- Development support

Middleware for IoT

Diversity of existing platforms for IoT

Middleware for IoT

Diversity of existing platforms for IoT

Platform	Interoperability	Discovery and management	Context-awareness	Scalability	Big Data	Security and privacy	Dynamic adaptation	Development support
Kaa	✓	○	○	✓	✓	✓	○	✓
Xively	✓	✓	○	✓	✓	✓	×	✓
Carriots	✓	✓	○	✓	✓	✓	×	✓
LinkSmart	✓	✓	✓	×	✓	✓	×	×
OpenIoT	✓	×	×	✓	✓	✓	×	✓
RestThing	✓	×	○	×	×	×	×	✓
WoT Enabler	✓	×	○	×	×	×	×	✓
S³OIA	✓	✓	✓	×	×	×	✓	×
Ubiware	✓	×	✓	×	×	×	✓	×
WSO2	✓	✓	×	✓	✓	✓	✓	✓
FIWARE	✓	✓	✓	✓	✓	✓	✓	✓

European Commission › Strategy › Digital Single Market › News ›

Digital Single Market

PROJECTS STORY | 30 March 2017

FUTURE INTERNET PPP

FIWARE – a European success story

FIWARE is an open source cloud platform with a collaborative and mature ecosystem of developers, innovation Hubs, accelerators, cities and more than 1000 SMEs and startups.

https://ec.europa.eu/digital-single-market/en/news/fiware-european-success-story

FI-WARE

Open APIs for Open Minds

https://www.fiware.org/

The FIWARE platform provides a rather simple yet powerful set of APIs (Application Programming Interfaces) that ease the development of Smart Applications in multiple vertical sectors. The specifications of these APIs are public and royalty-free. Besides, an open source reference implementation of each of the FIWARE components is publicly available so that multiple FIWARE providers can emerge faster in the market with a low-cost proposition.

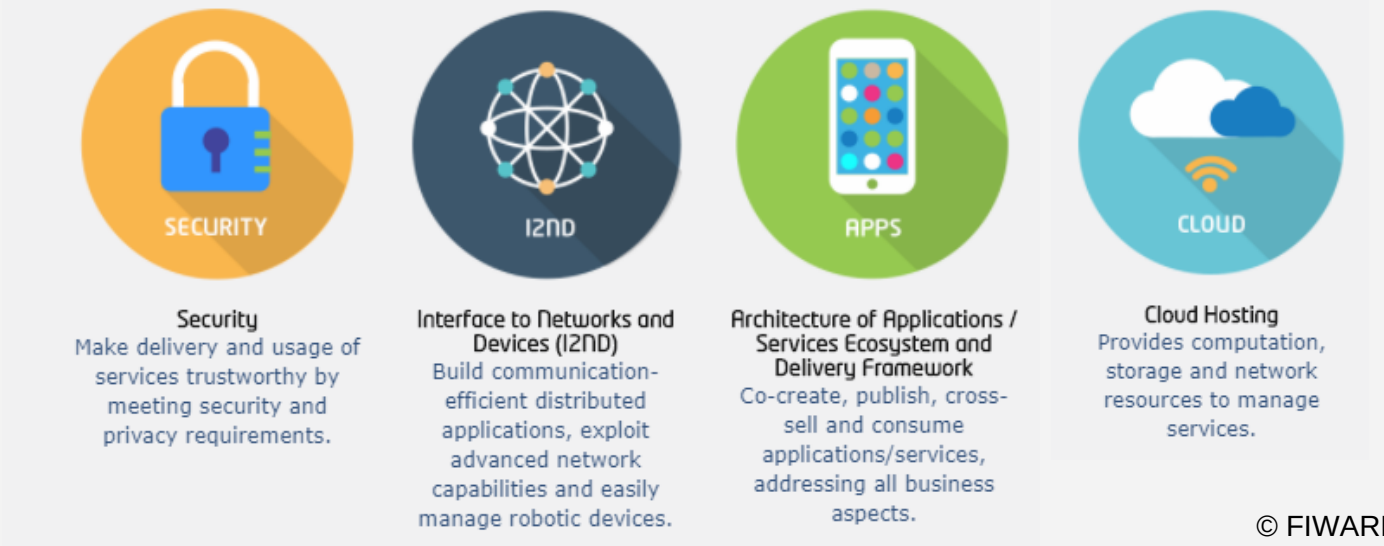
FIWARE

- Future Internet middleware platform developed with involvement of both industry and academia, supported by the European Commission
- Characteristics
  - Component-based, generic nature
  - Open source
  - Set of specifications available through standardized interfaces
  - Flexibility
  - Royalty-free
  - No vendor lock-in
  - Use in many successful cases from different application domains
  - Community support

FIWARE

Generic Enablers (GEs)

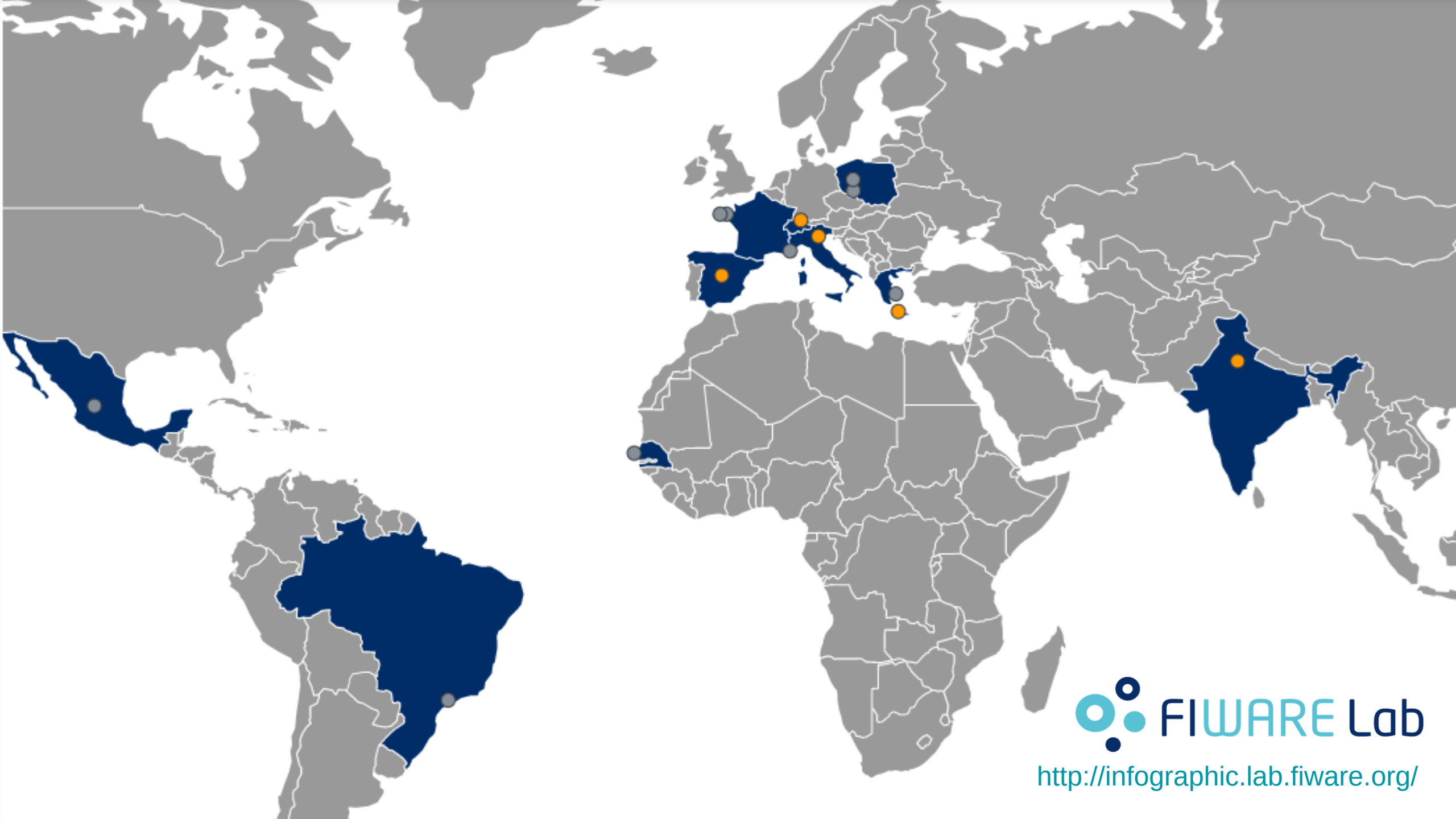
- Generic componentes described by an open specification and API
- Released with a reference implementation
- Available at the FIWARE Catalogue (https://catalogue.fiware.org/)



FIWARE

Domain-Specific Enablers (DSEs)

- Components developed by the community, following the FIWARE NGSI specifications
- Meeting requirements of specific application domains
- Use of existing GEs and/or DSEs


FIWARE Lab
http://infographic.lab.fiware.org/

FIWARE MARKET PLACE

FIWARE
Open APIs for Open Minds

Powered by FIWARE

- Solutions
- Platforms

FIWARE-Ready Technologies

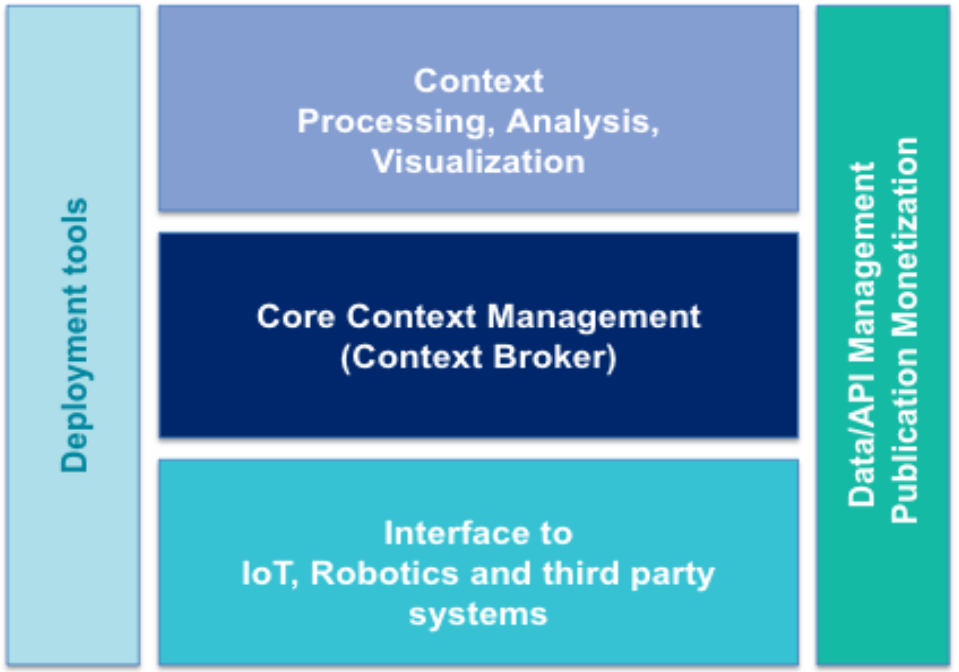
- IoT Devices
- Software Enablers

FIWARE Services

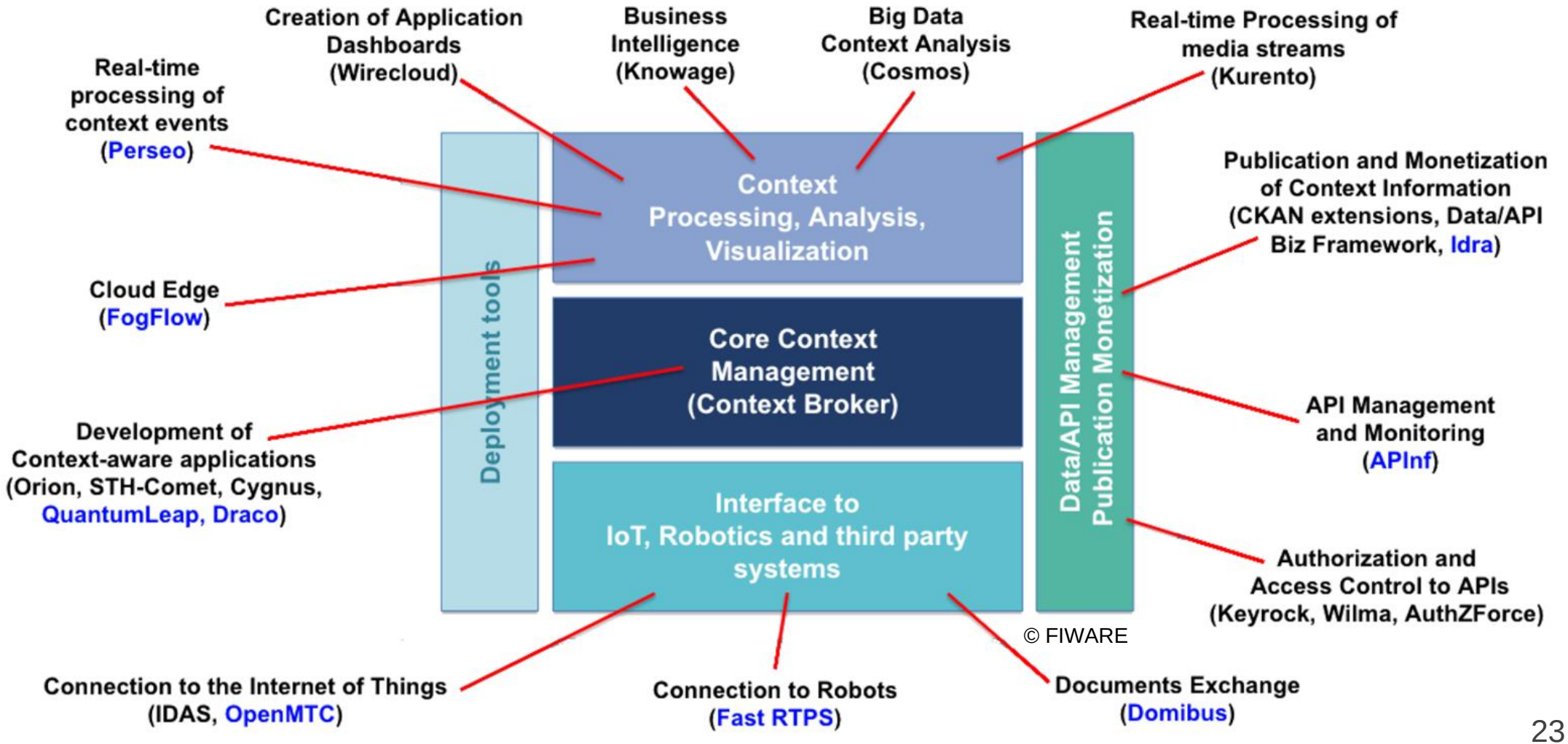
- Training and coaching
- Consultancy and integration services

http://marketplace.fiware.org/

FIWARE Overview



FIWARE Overview

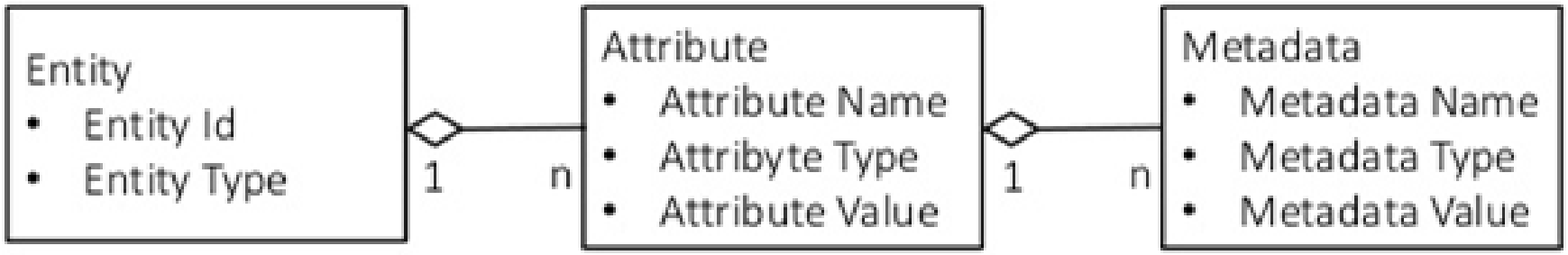


FIWARE NGSI

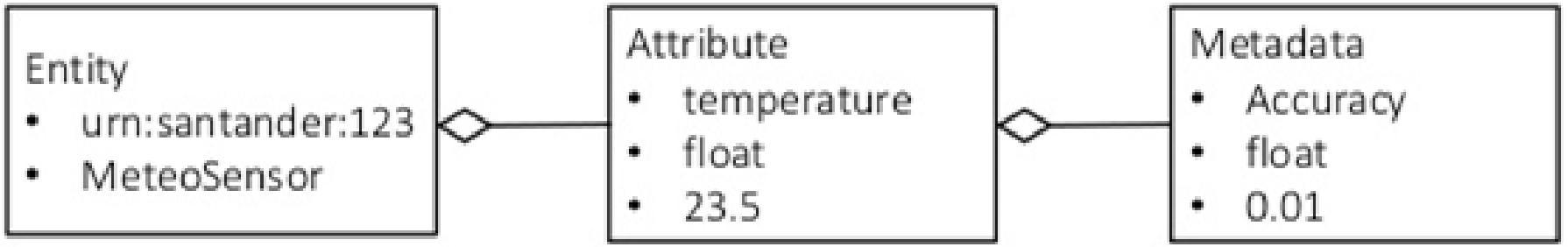
- FIWARE adopts NGSI (Next Generation Service Interface), a RESTful API to allow for interoperability among GEs
- Use of NGSI9 and NGSI10 interfaces defined by OMA NGSI Context Management
- Main elements in the NGSI data model
  - Context entities (or simply entities) representing either physical or logical objects, e.g., a sensor, a person, a room, etc.
  - Attributes as properties of entities
  - Metadata as [optional] data associated to attributes

FIWARE NGSI Data Model

NGSI Metamodel



NGSI data example



FIWARE NGSI Data Model

Normalized format

```
{
    "id": "entityId",
    "type": "entityType",
    "att1": {
        "value": "value1",
        "type": "Text",
        "metadata": {
            "metada1": {
                "value": "metavalue1"
            }
        }
    }
}
```

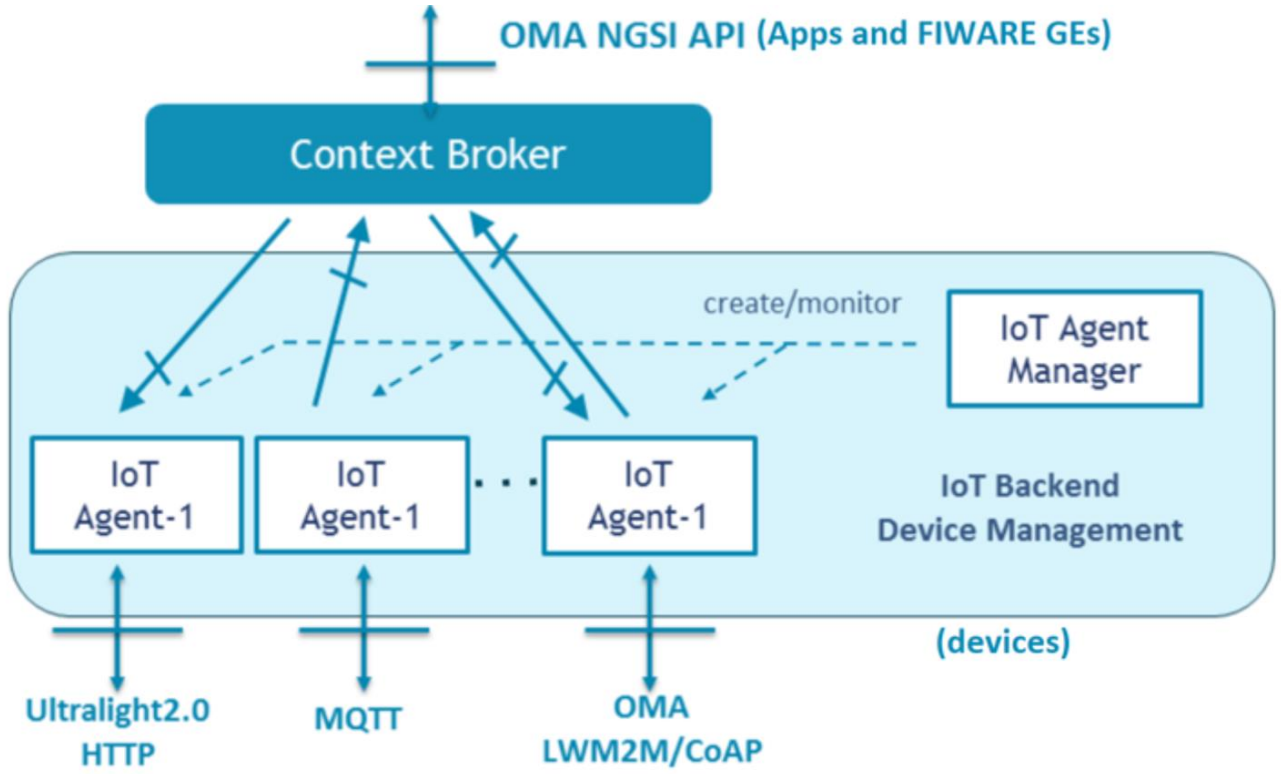
Simplified entity representation

```
{
    "id": "entityId",
    "type": "entityType",
    "att1": "value1"
}
```

FIWARE IoT Architecture

The FIWARE IoT Architecture encompasses

- receiving messages at software level through IP networks
- converting them to the NGSI standard, and
- forwarding them to a broker



FIWARE IoT Architecture

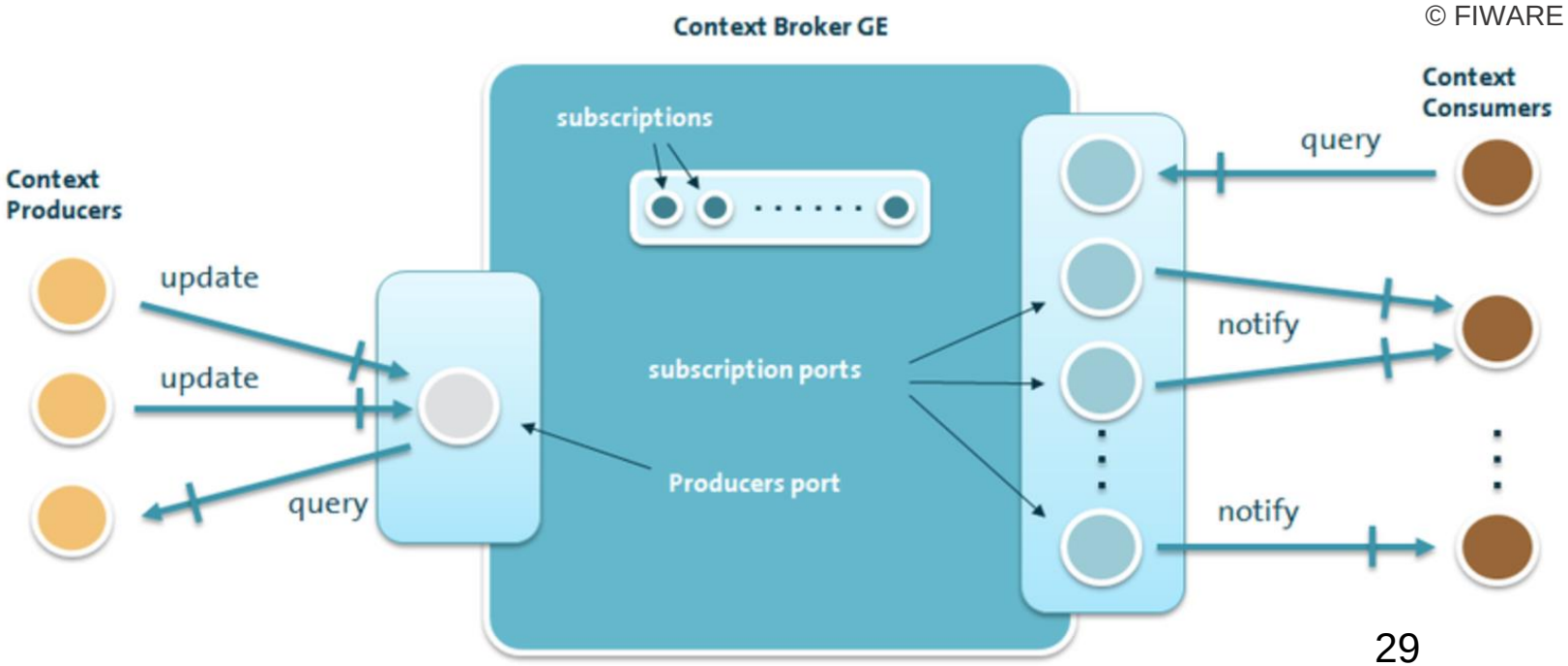
FIWARE IoT Agents ease the interface with devices using the most widely used protocols in IoT, besides allowing developers to build their own IoT Agent

IoT Agent	Bridges...
JSON	HTTP/MQTT messaging (with a JSON payload) and NGSI
LWM2M	Lightweight M2M protocol and NGSI
Ultralight	HTTP/MQTT messaging (with an UltraLight2.0 payload) and NGSI
LoRaWAN	LoRaWAN protocol and NGSI
OPC-UA	OPC Unified Architecture protocol and NGSI

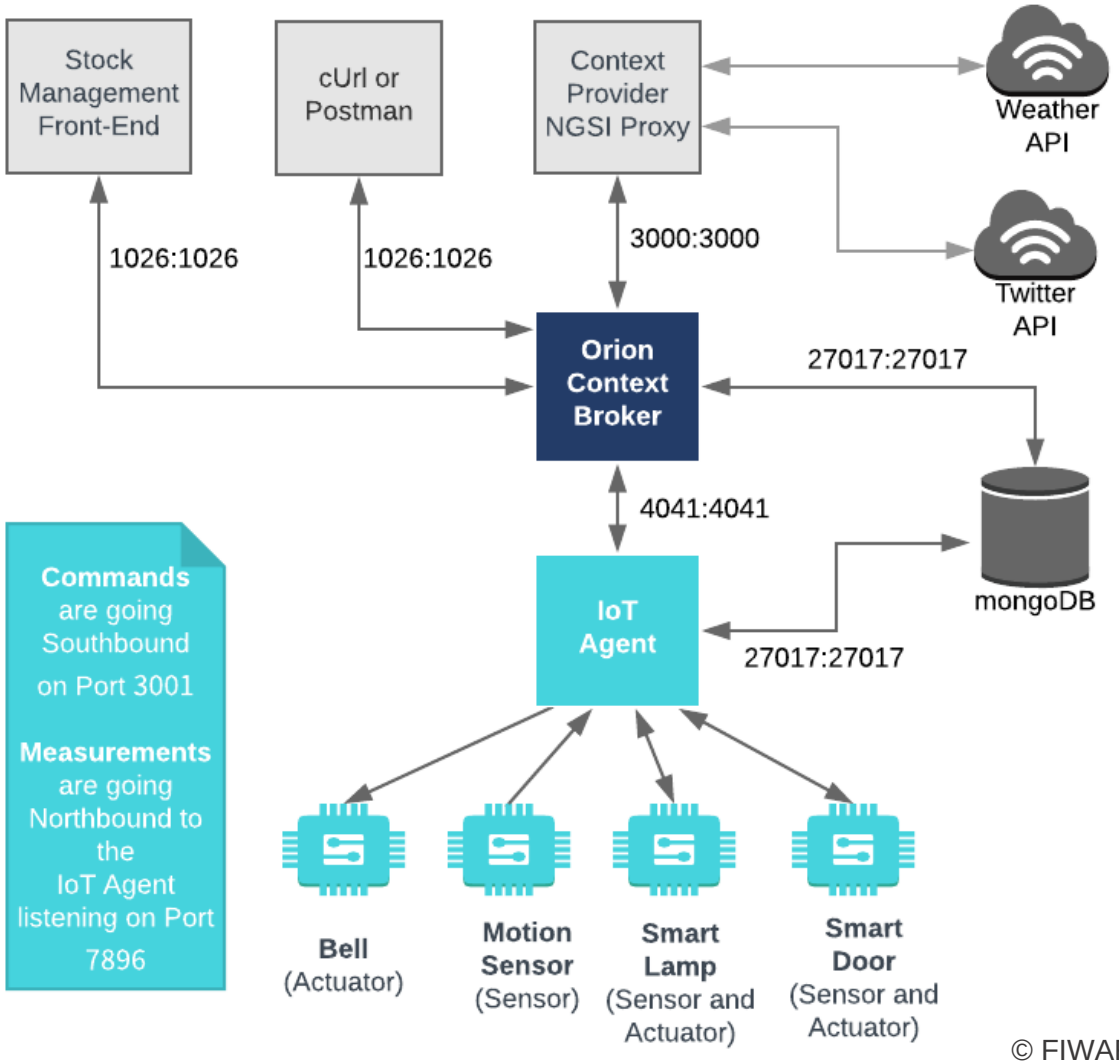
FIWARE

Orion Context Broker (or simply Orion)

- Considered as “the heart of FIWARE”
- Publish-subscribe broker for managing context entities
- Management of entities, updates, queries, and subscriptions



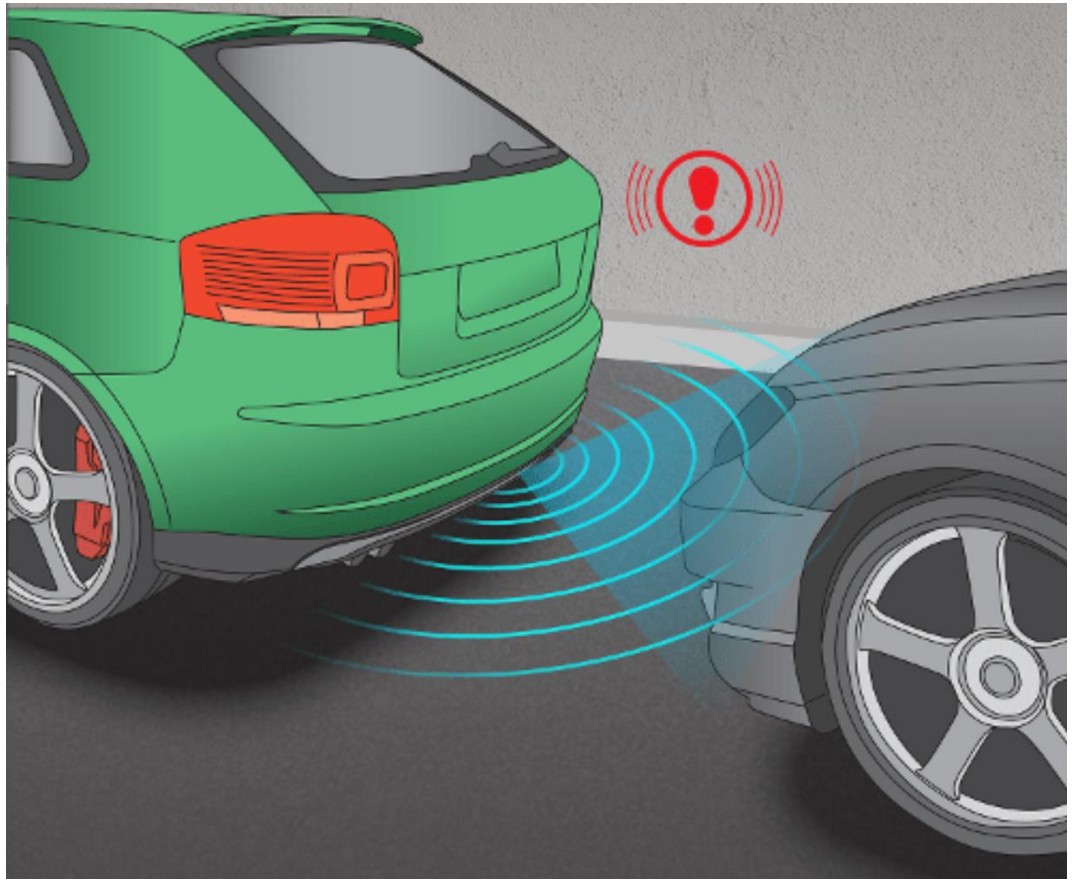
FIWARE



In FIWARE, IoT Agents bridge different protocols to the NGSI standard

Developing an IoT application with FIWARE

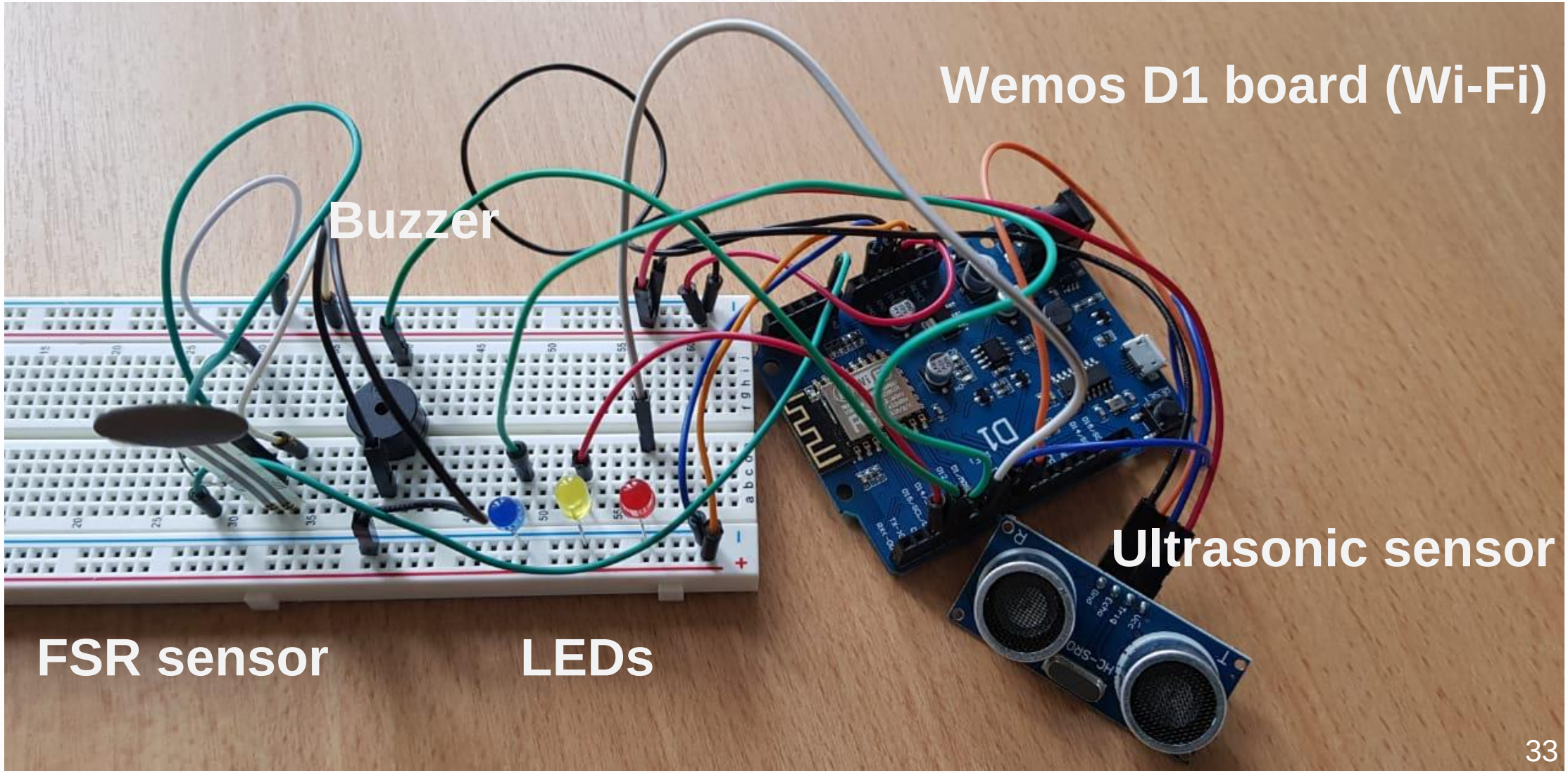
A Parking Assistant



Aim: to warn the driver about the distance between the car and obstacles

- Sound emission according to distance
- Blinking LED
- Collision confirmation

A Parking Assistant: Hardware prototype

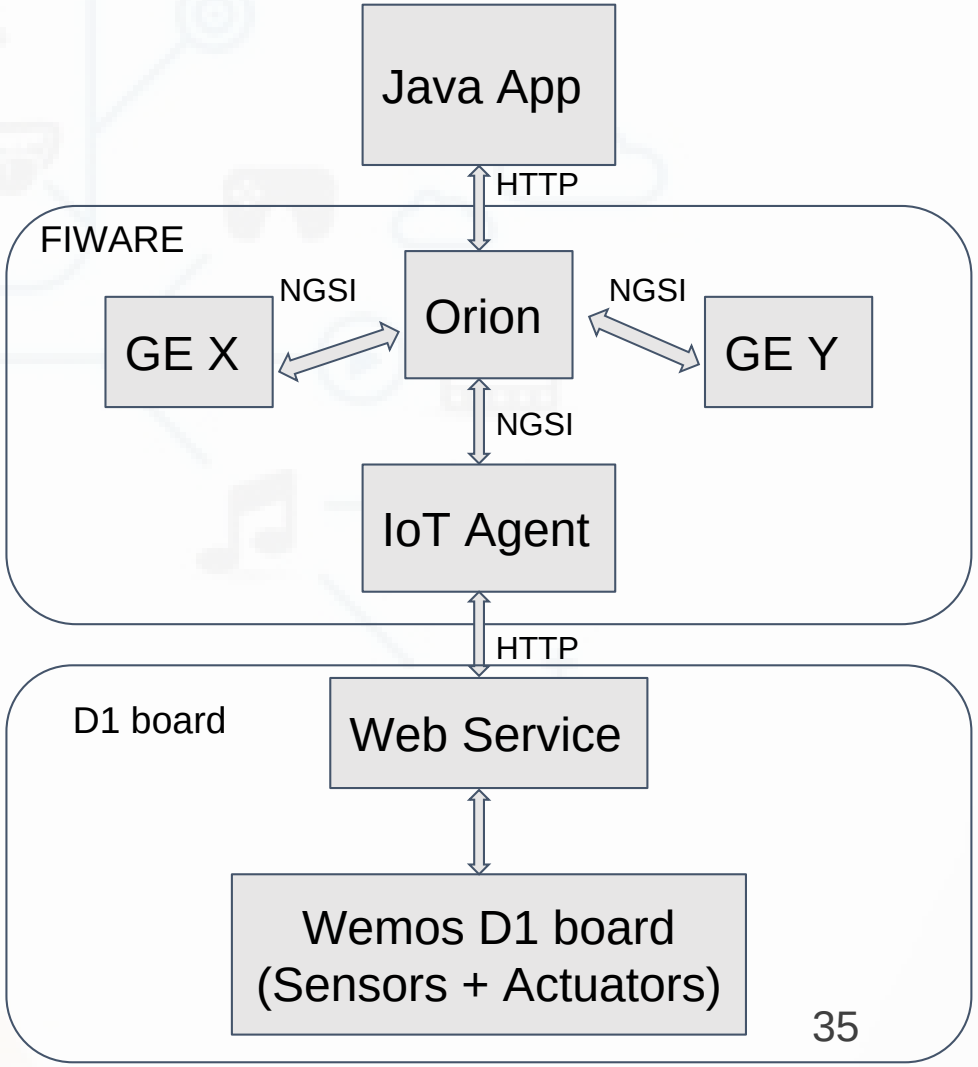


A Parking Assistant: Hardware prototype

Hardware element	Role
Ultrasonic sensor	Measures distance between sensor and obstacles
FSR sensor	Measures mechanical pressing force
Buzzer	Emits sound in different frequencies based on distance measured by the ultrasonic sensor (indicating approximation)
Yellow LED	Activated if obstacle is detected by the ultrasonic sensor (indicating approximation)
Red LED	Activated if FSR sensor is pressed (indicating collision)

A Parking Assistant: Architecture

- Arduino compatible board with Wi-Fi module with deployed sensors and actuators
- FIWARE GEs
  - Orion
  - IoT Agent (Ultralight 2.0)
- A Java application making HTTP requests to Orion



A Parking Assistant: Data flow

1. Sensors deployed at the D1 Wi-Fi board collect data
2. Gathered data are managed by a Web service deployed at the D1 board to enable IP connectivity
3. IoT Agent translates pure HTTP to FIWARE NGSI
4. Orion hosts context entities
5. Application consumes data through HTTP requests to Orion
    a. Measured data are displayed in the application
    b. Blue LED can be activated by the application (indicating actuation)

Developing an IoT application with FIWARE

Steps to follow

1. Assemble hardware
2. Deploy the FIWARE GEs to be used by the application according to its requirements
   - Generic guide available at https://fiware-tutorials.readthedocs.io/

1. Develop the application itself communicating with FIWARE
   - Sending and receiving HTTP requests from/to Orion following the NGSI standard

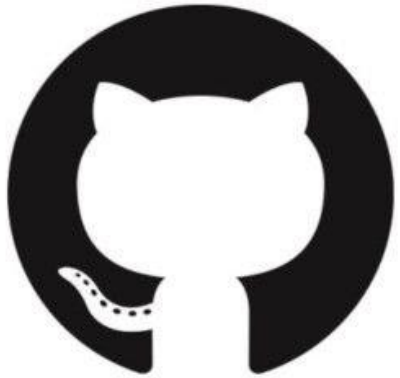
Developing an IoT application with FIWARE

Deploy the FIWARE GEs to be used by the application

1. Install Docker: https://docs.docker.com/install/
2. Start Orion + IoT Agent container
   a. Clone the .yaml file from https://github.com/PedroVictorB/TSP_Tutorial/blob/master/docker-compose.yml
   b. Go to folder and run
      docker-compose `up -d`
   c. Check if Orion and IoT Agent are running with
      `docker ps`
   d. Check if Orion is running, in a browser:
      `http://localhost:1026/v2`

Developing an IoT application with FIWARE

Develop the application itself



GitHub

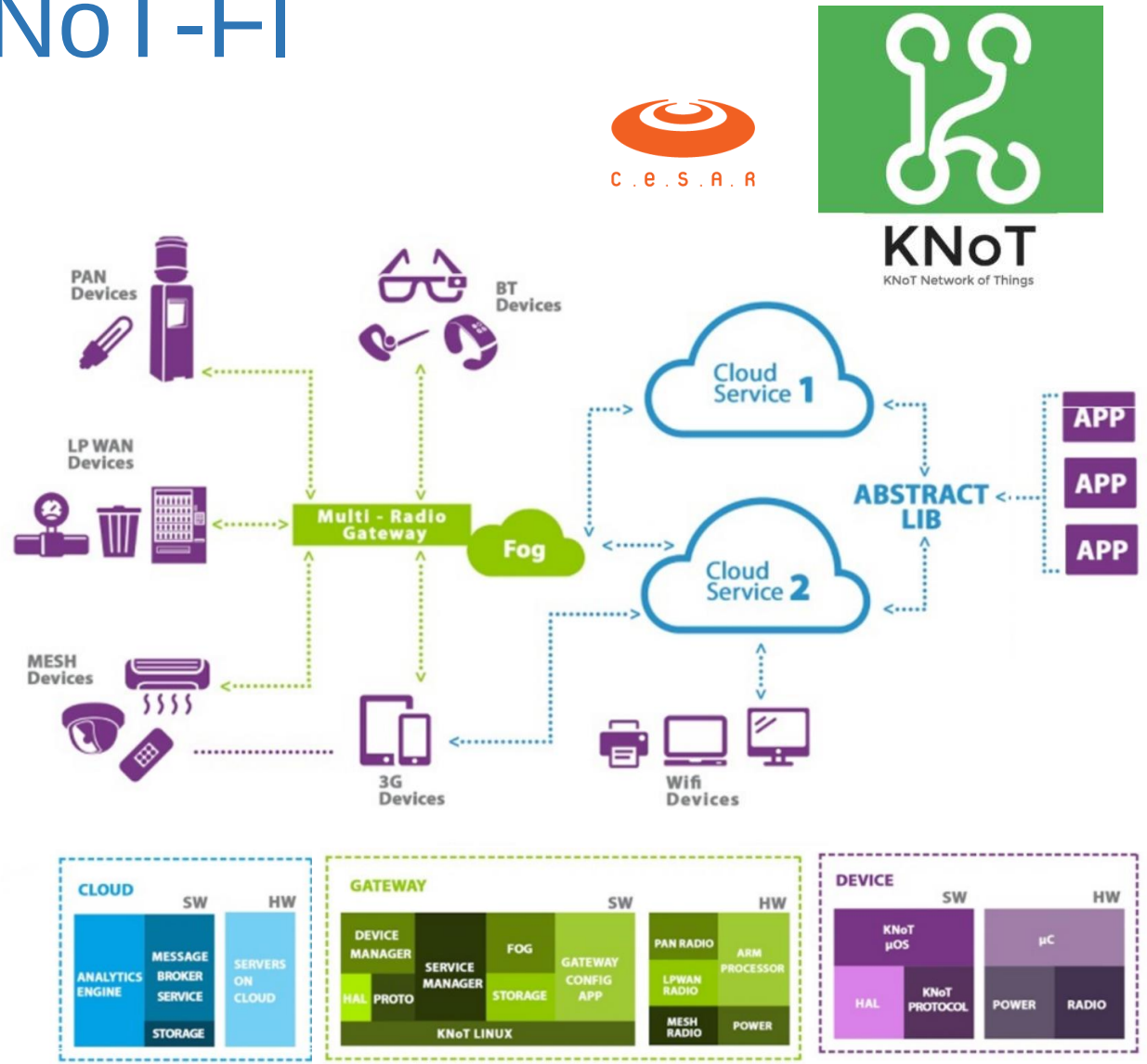
https://github.com/PedroVictorB/TSP_Tutorial


DEMO

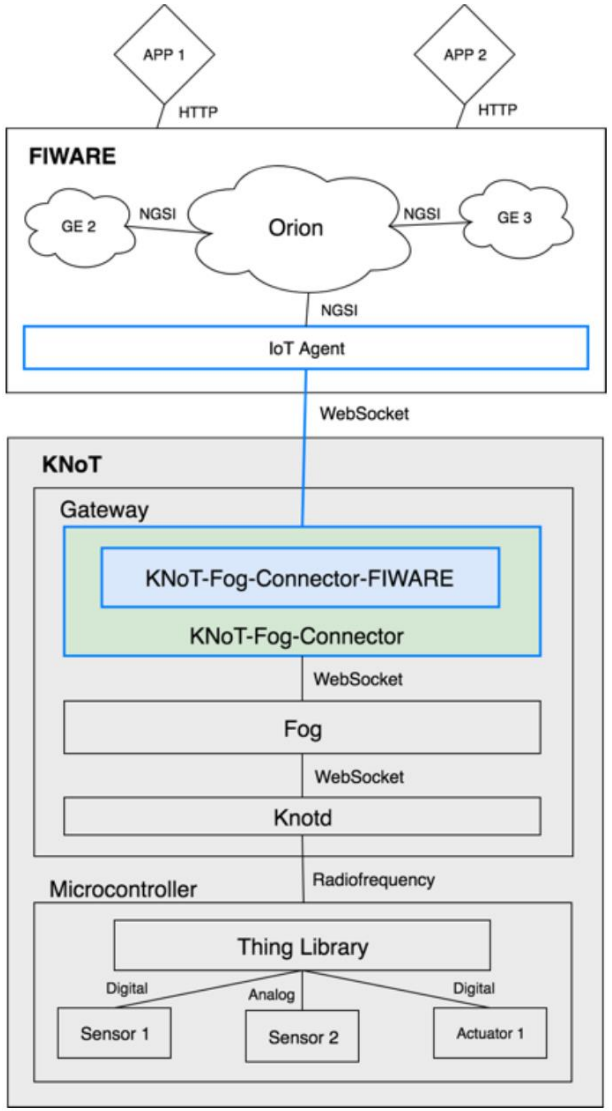
Going a step further: KNoT-FI

KNoT

- Platform developed by the CESAR Research Institute, in Brazil
- Virtualization of physical devices
- Device interoperability and management
- Low-cost hardware and easy-to-deploy software
- Multi-radio gateway to integrate devices without native IP stack



Going a step further: KNoT-FI



KNoT-FI

- Integration of the KNoT (https://www.knot.cesar.org.br/) and FIWARE platforms to support IoT application development
- Allow for the integration of devices with no native Internet connection as provided by KNoT
- Enrich KNoT with capabilities provided by FIWARE


That's all Folks!

FI-WARE

Developing Internet of Things Applications with FIWARE

cesar.perdigao@gmail.com
pedrovictor_caldas@hotmail.com
everton@dimap.ufrn.br

ANOS

UFRN
UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

DIMAp
UFRN

SMART METROPOLIS

SET E
Software Engineering Team
Research and Innovation

IMD
INSTITUTO
METRÓPOLE DIGITAL