Smart Cities: the Challenge of Interoperable Software Architecture

Guadalupe Ortiz, guadalupe.ortiz@uca.es
Software Engineering Research Group, University of Cádiz

The Sixteenth International Conference on Internet and Web Applications and Services (ICIW 2021)
DataSys 2021Congress

Smart Cities: the Challenge of Interoperable Software Architecture

- **Guadalupe Ortiz** is a permanent Associate Professor in the Department of Computer Science and Engineering, at the University of Cadiz and member of the UCASE Research Group.

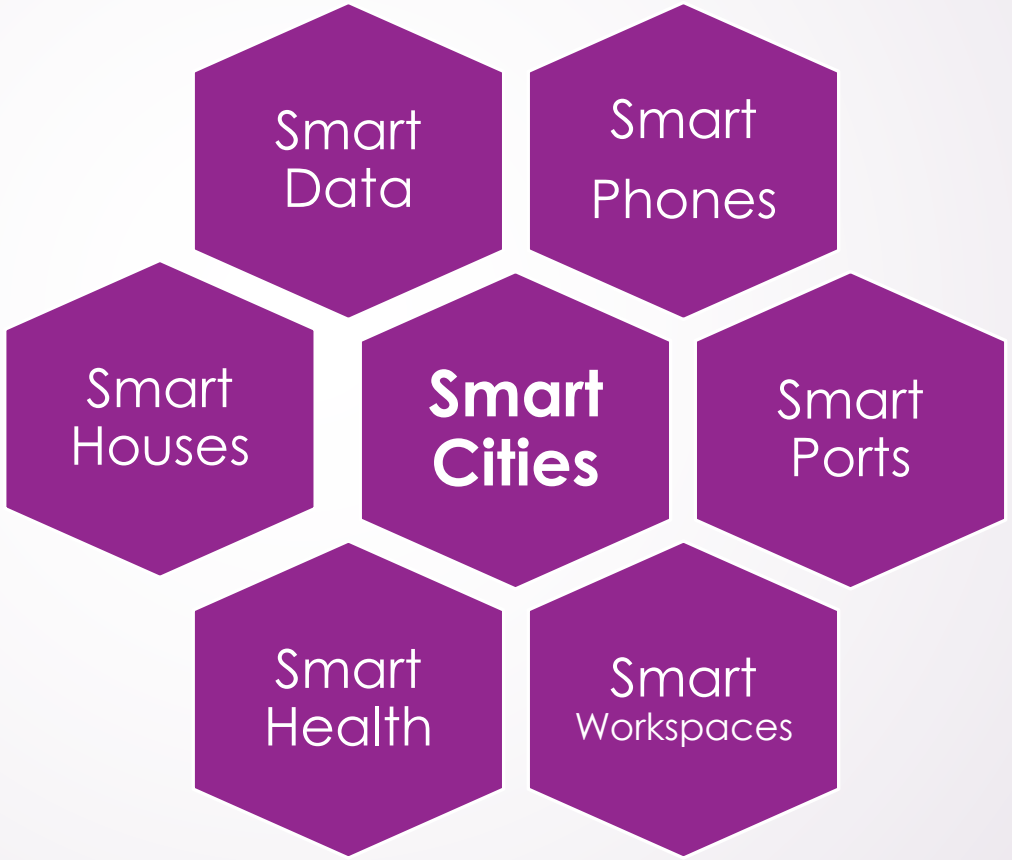
Ongoing research projects:

- Smart Healthy Port: Real-Time Big Data Processing for Intelligent Air Quality Management in Sea Ports
- GANGES: Real-time big data processing for smart water network management
- FAME: Formal Modelling and Advanced Testing Methods. Applications to Medicine and Systems
- RCIS: Network in Service Science and Engineering

- Currently, her research focuses on trending topics such as the integration **of complex-event processing** in **service-oriented architectures** and facilitating **context-awareness** in the scope of **Internet of Things, Smart Cities** and **Ambient Assisted Living.**

Everything is Smart

Everything is Smart



Smart Workspaces

Smart workspaces

- **Human-centered**
- Facilitating smarter living, working and **living experience.**

At smart homes

- Devices, sensors, tools and platforms that learn from the way humans use their homes.
- **Contextualized and personalized experiences**

Smart Health

Smart Health

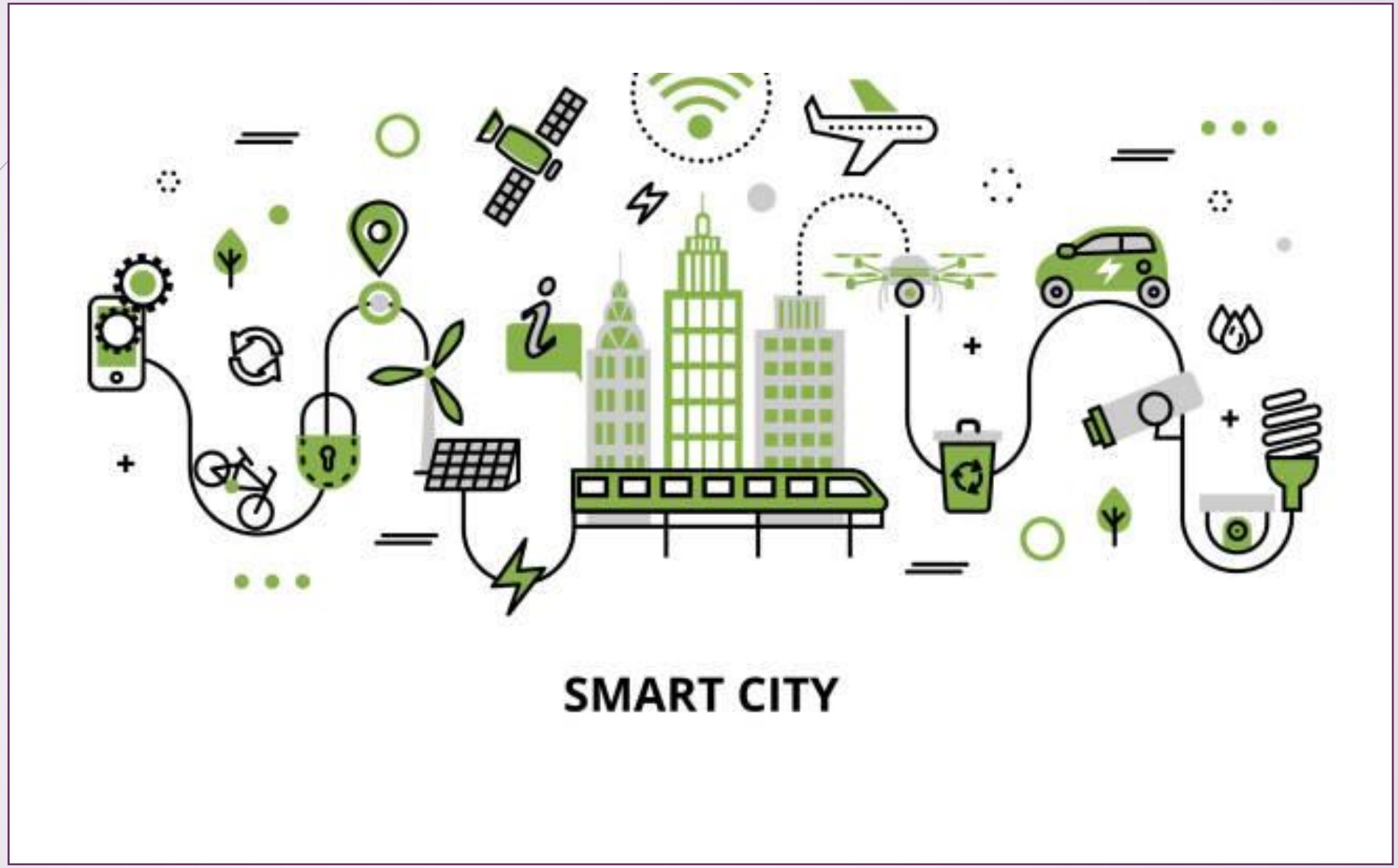
- Use of smart city tools and infrastructures in the service of health

Aim: to improve health and quality of life

- Information and communication technologies
- Contextual patient information and use of mobile devices

Advantages

Simplify processes and streamline procedures → cost reduction
Adapting to the patient's context → better treatment
Prevention of health risks


SMART CITY

Interoperability??

Outline

Concepts

Technologies

Architectures

Case Studies

Conclusion



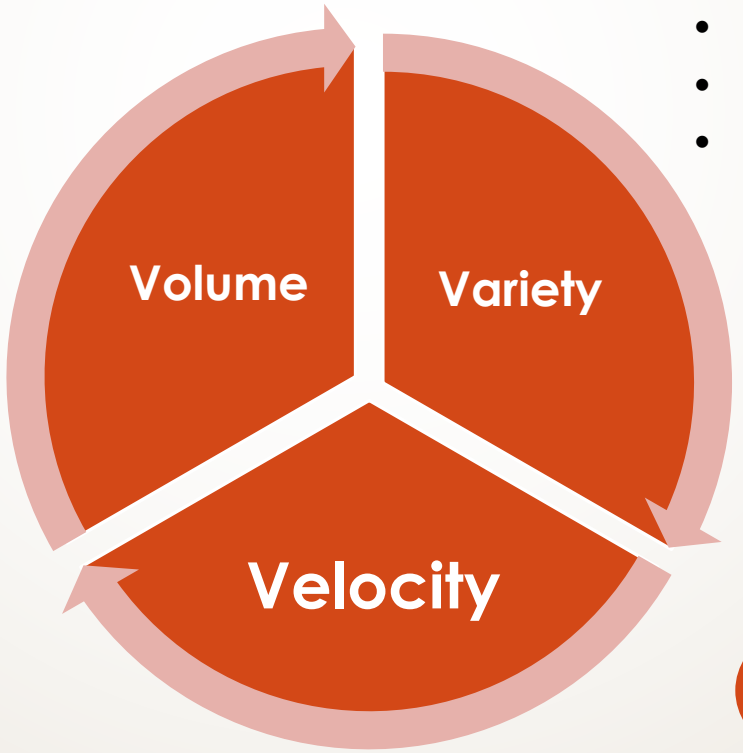


Concepts Technologies Architectures Case Studies Conclusion

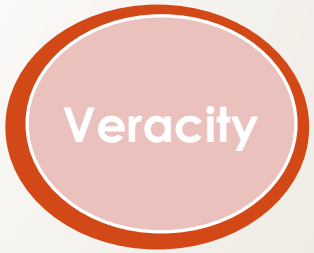
Outline

Big Data

- Internet
- Information Systems
- Cloud
- Social Networks
- Internet of Things



- Storage Format
- Communication protocols
- Data representation



- Social Networks
- Internet of Things
- …Near real-time

Small Data

Why small?

- **Accessible, understandable and actionable** in everyday tasks
- Data mining (batch)

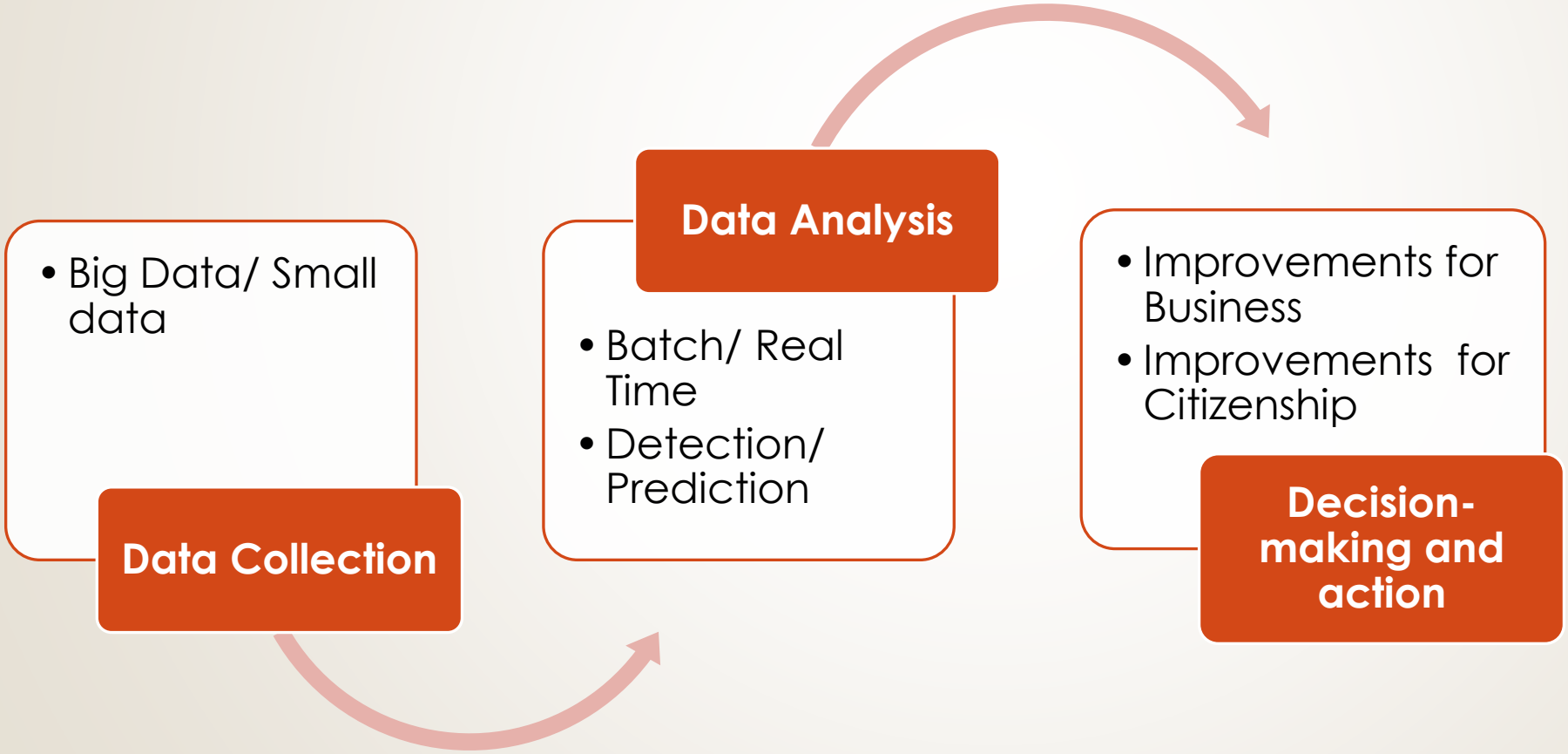
Applications

- Data-driven marketing, CRMs, …

Distinctive features

- **End-user** focused → Context
- **Data democratization** → Collaboration

Smart Data

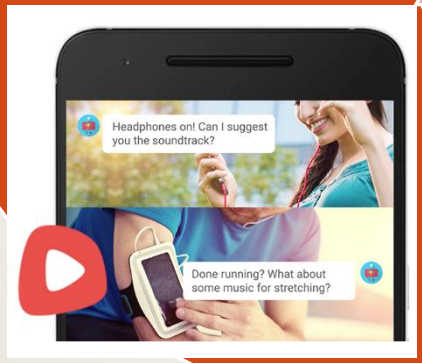

• Big Data/ Small data
Data Collection
Data Analysis
• Batch/ Real Time
• Detection/ Prediction
• Improvements for Business
• Improvements for Citizenship
Decision-making and action

Context and Context Awareness

What is it?

What can we take into account?

Mobile phones do it!



Context type	Example
Time	Current local time
Location	Latitude and longitude
Place	Place, including place type
Activity	Detected user activity (walking, running, biking)
Beacons	Nearby beacons matching the specified namespace
Headphones	Are headphones plugged in?
Weather	Current weather conditions

Google permits programming it

(Collaborative) Internet of Things

IoT

- A network of globally interconnected and uniquely identified things or objects
- IoT platforms

C-IoT

- Individual-Business-Community/Infrastructure
- Multiple domains: health, logistics, energy
- Collaboration at sensor/situation of interest/services level
- Prioritization

Outline

Concepts

Technologies

Architectures

Case Studies

Conclusion

Service Oriented Architecture and RESTful Web Services

Service Oriented Architecture

Services

- A **contract** (user benefits).
- Particular **discoverable functionality** describing what it can do and how to interact with it.

Service Oriented Architecture (SOA)

- Software architecture that defines a **decoupled model** of services to support business process requirements.
- They provide functions that can be **reused** by different clients (they only need to know the service description).

RESTful Web Services

RESTful services

- REST: Representational State transfer
- REST is an **architectural style** for services using **web standards.**

Advantages

- Light
- Human readable
- Easy to build

REST Communications

- Everything can be identified as a **resource** and each resource can be identified by a **URI**.
- A resource can be represented in **multiple formats**, defined by a **media type**.
- Standard **HTTP methods** are used to interact with the resource: mainly GET, POST, PUT and DELETE.
- Communication between the client and the endpoint is stateless.

Event-Driven Architecture and SOA 2.0

Event-Driven Architecture

Events

- A **change** in the **state** of something.
- **Something that occurs (or does not occur).**
- A **detectable** condition.

Event-Driven Architecture (EDA)

- Particular style of **event processing**.
- Architectural style in which one or more components of a software system are activated upon detection of an event and where these components are **decoupled**.
- It is based on the **publish/subscribe** mechanism.

Event-Driven Architecture

Publish/Subscribe systems

- **Heterogeneity**: event generators publish the types of events they offer, and consumers subscribe to them through the interface offered for subscription and receipt of the resulting notifications.
- **Asynchrony**: notifications are sent asynchronously by event-generating publishers to all subscribers who are interested in them (decoupled publishers and subscribers).

SOA versus EDA

Comparative SOA/EDA	SOA	EDA
Interaction	Loosely coupled	Decoupled
Cardinality	One-to-One	Many-to-Many
Paradigm	Request/Response	Event/Notification
Communication	Synchronous	Asynchronous

Event-Driven Service Oriented Architectures (SOA 2.0)

ED-SOA or SOA 2.0

- **Communications** between users, services and applications are **event-driven.**
- Event-driven communication allows a **faster response** to changes in **real time.**
- **Events** in the system **trigger** the launch of **system services**.

Enterprise Service Bus versus Microservices Architectures

Enterprise Service Bus (ESB)

- **Integration** element (multi-protocol and multi-purpose) in SOA.
- It combines web services, messaging, transformation, data routing and enrichment, security policies, among others.

Advantages

- They can **integrate EDA** and **SOA**.
- Ideal for working in **heterogeneous** environments: different technologies and protocols: from the most modern to the most conventional (legacy).
- They **reduce** the total **cost** of management and maintenance.

Enterprise Service Bus

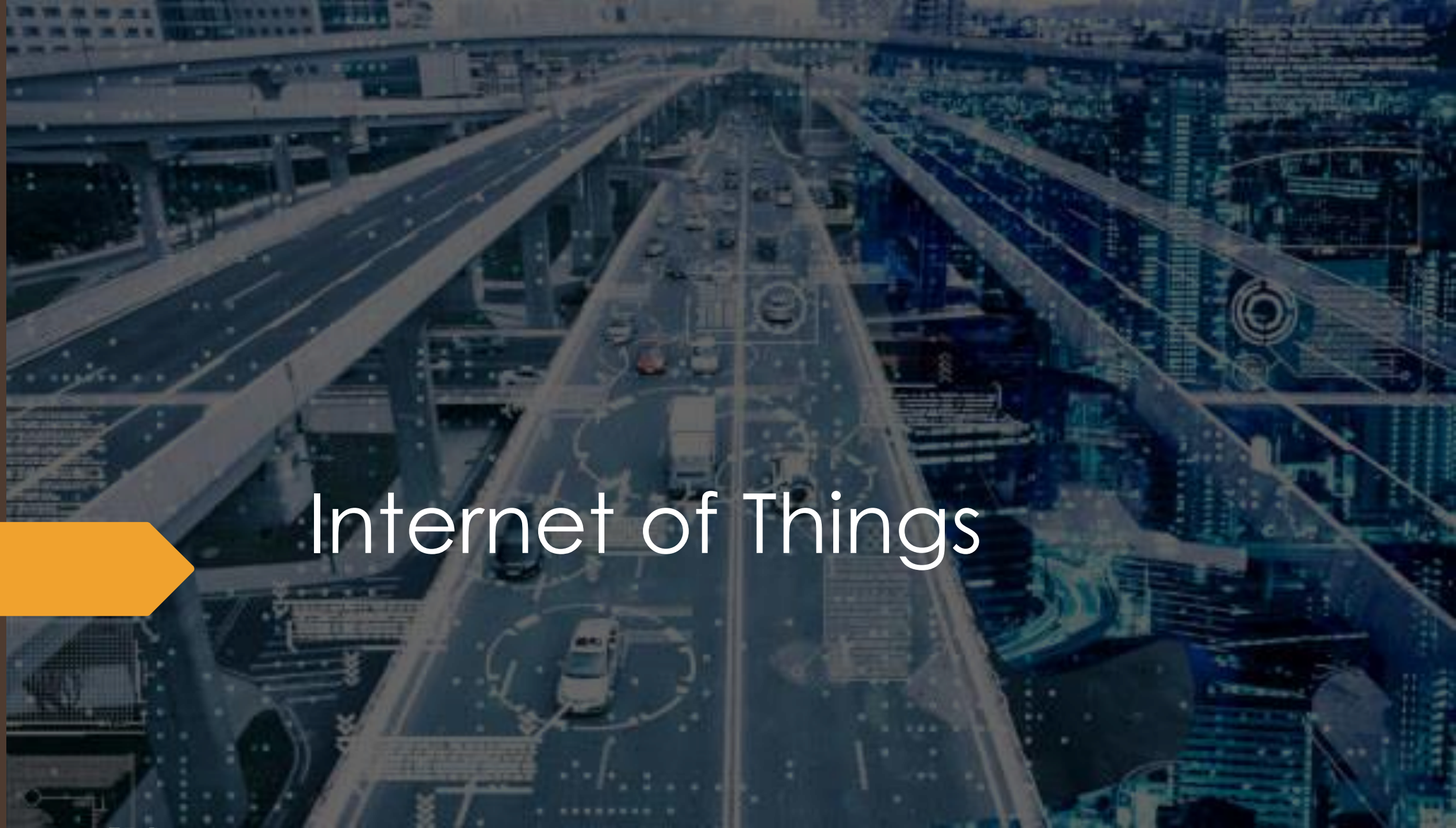
Microservice Architectures

Features

- A single application as a set of **small services**.
- Each service runs in its own process.
- Services communicate with **lightweight mechanisms** (REST API over HTTP).
- **Deployment is independent**.
- There is hardly any centralized management.

Advantages

- ✓ **Scalability, evolution, maintenance**
- Security, consistency, data traffic



Internet of Things

Internet of Things

Features

- It proposes the use of a network of **globally interconnected things** or objects uniquely identified through an address scheme.
- Very present in Smart World
- Several computational layers:
  - Cloud computing
  - Fog computing
  - Edge computing


Complex Event Processing

Complex Event Processing (CEP)

- Technology that allows **processing, analysing and correlating** large quantities of events.
- To detect and respond in **real time** to **critical** or relevant business **situations**.
- **Event patterns** will infer new, more complex events ("**situations**") with greater semantic meaning.

Advantages

- Improved **quality of decisions**.
- **Rapid** response.
- **Prevention** of information **overload**.
- **Reduction** of human **effort**.

Complex Event Processing

Concepts

Technologies

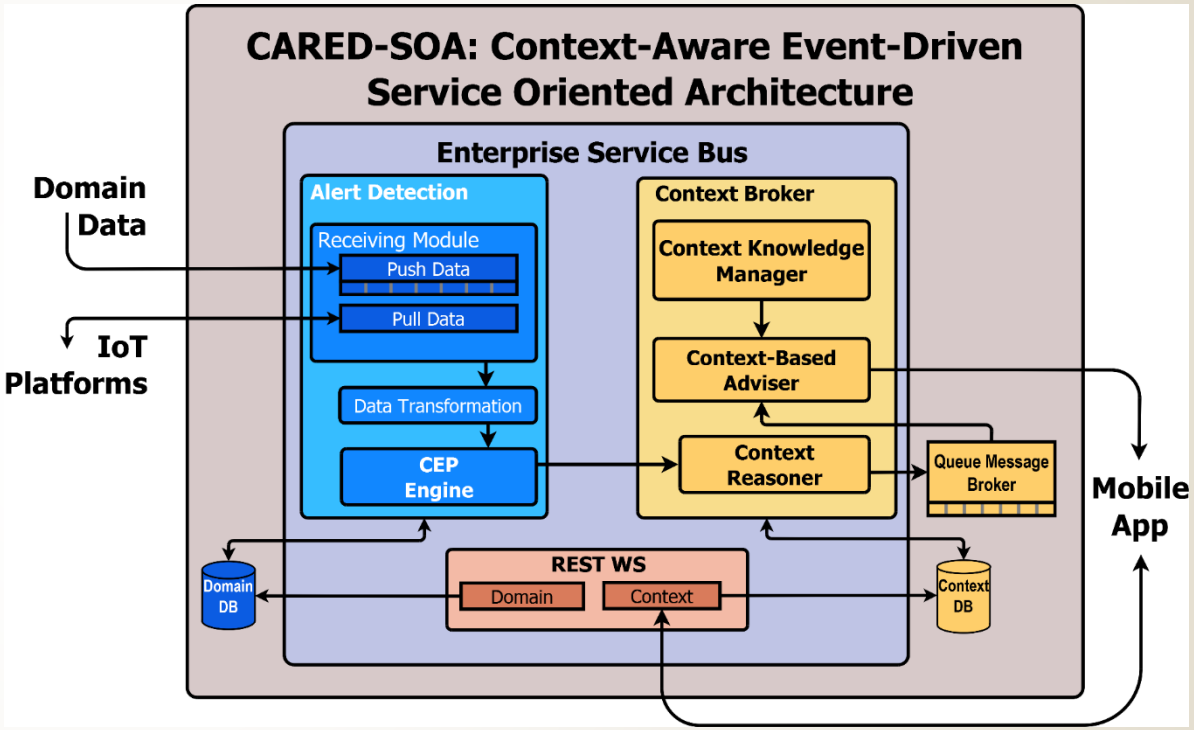
Architectures

Case Studies

Conclusion

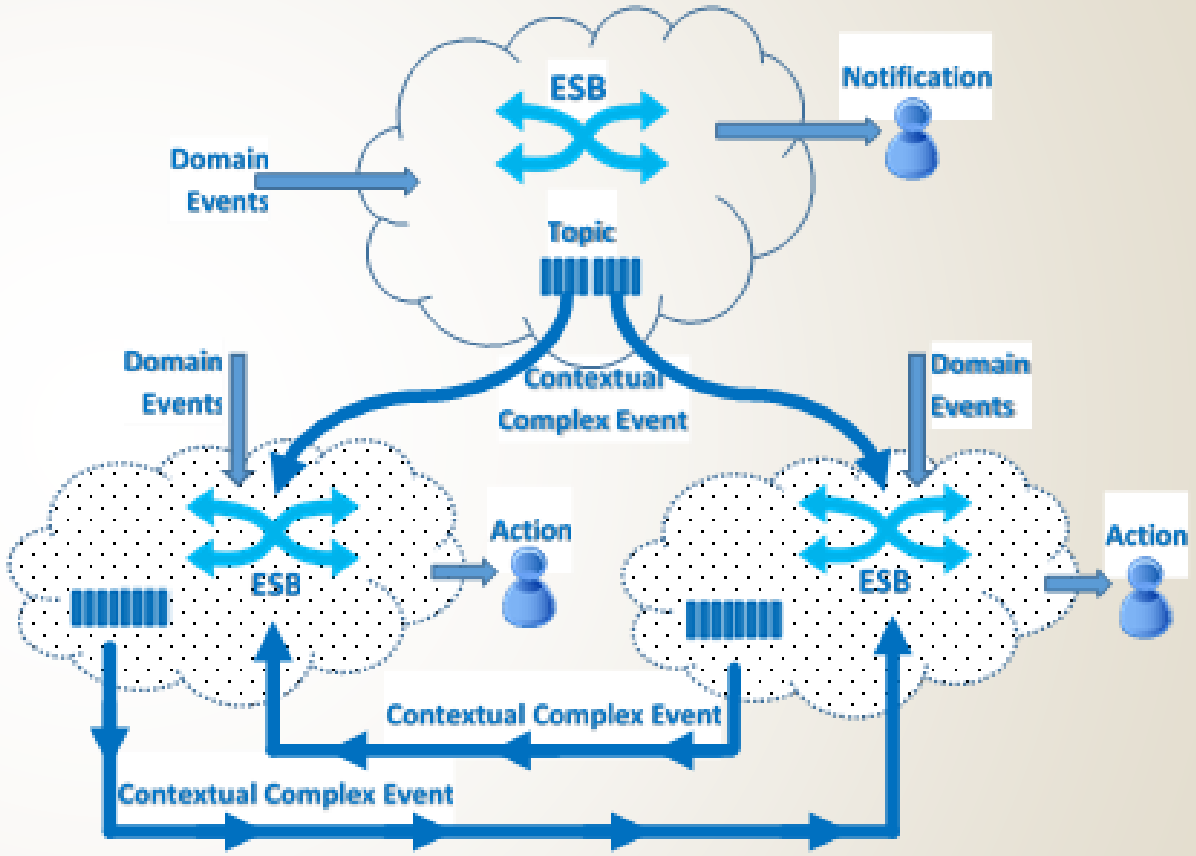
Outline

CARED-SOA Architecture



DOI: 10.1109/ACCESS.2017.2679338

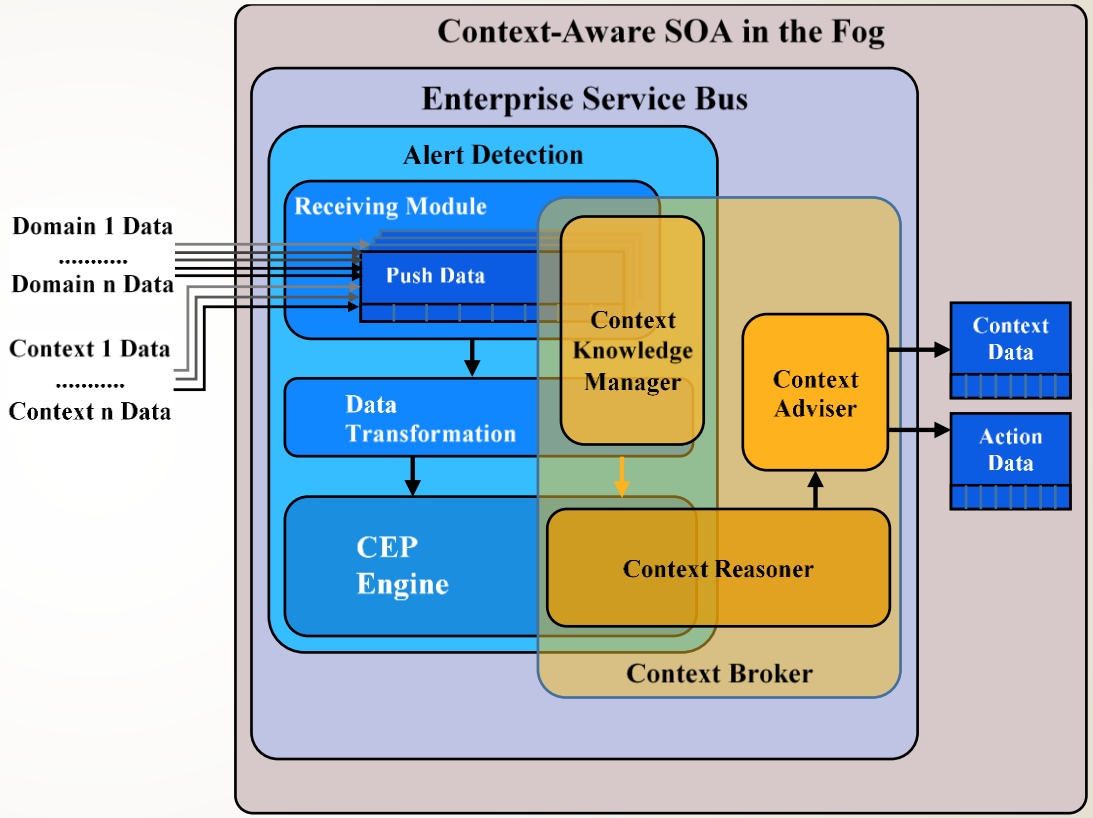
COLLECT Architecture (Overview)



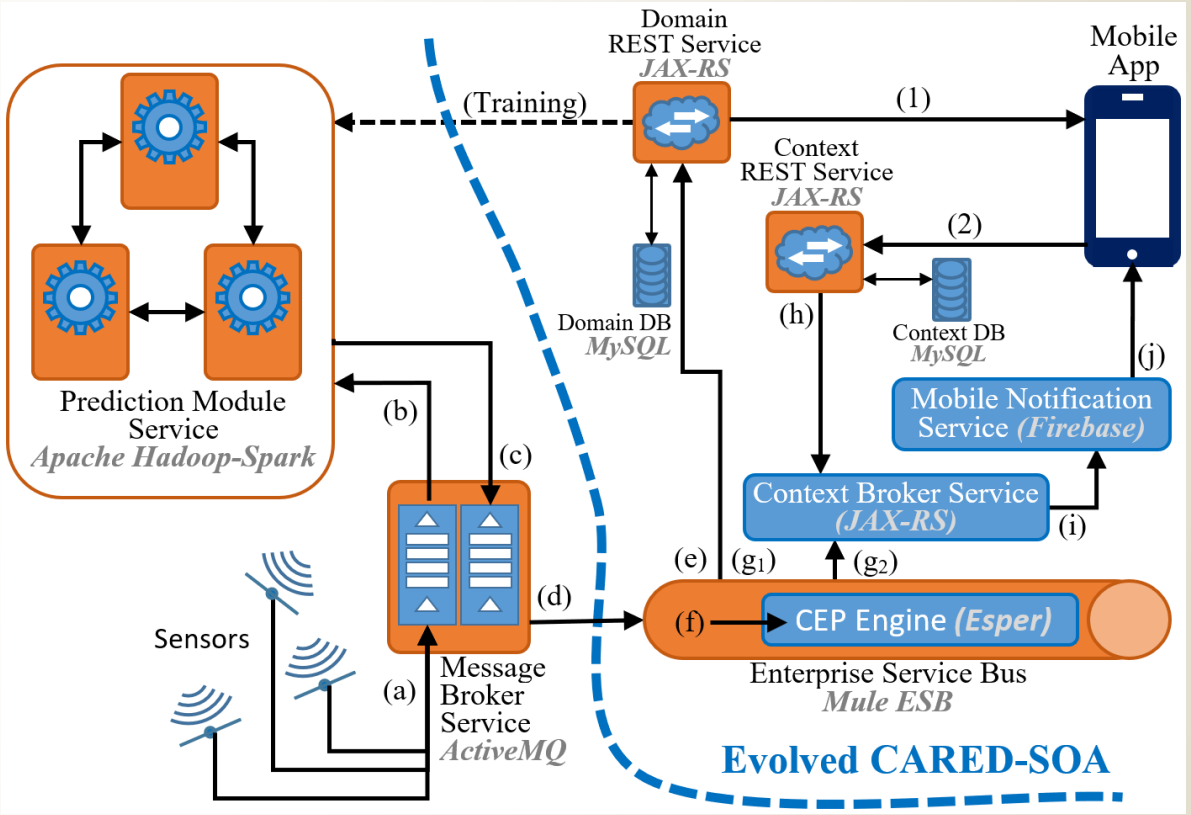
DOI: 10.1016/j.eswa.2017.05.034

COLLECT Architecture (fog node)

DOI: 10.1016/j.eswa.2017.05.034



Predictive CARED-SOA



DOI: 10.1109/ACCESS.2019.2960516


Concepts


Technologies


Architectures


Case Studies


Conclusion

Outline

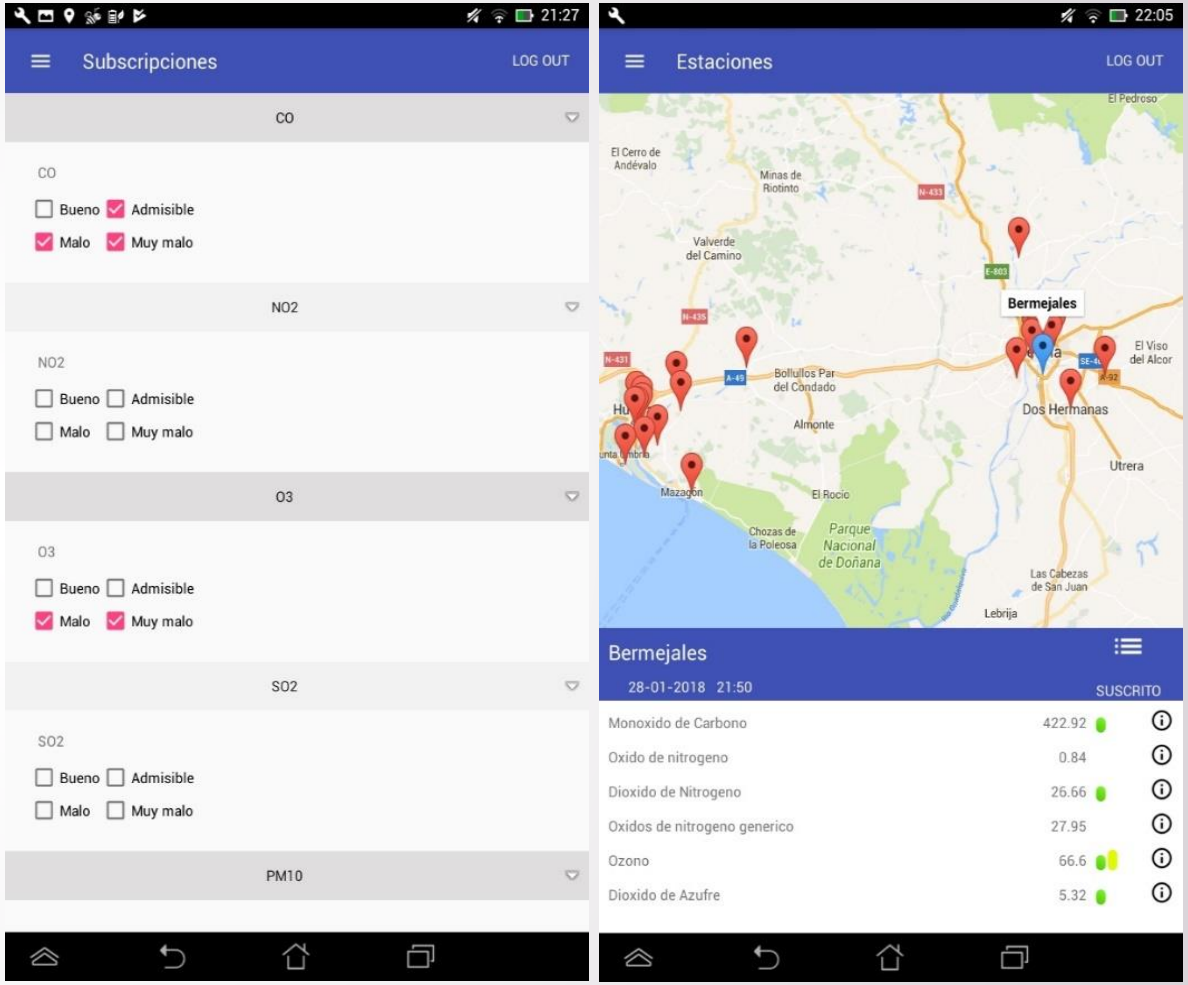
Air4People (Motivation)

DOI: 10.3217/jucs-024-07-0846

Air4People (Architecture)

DOI: 10.3217/jucs-024-07-0846

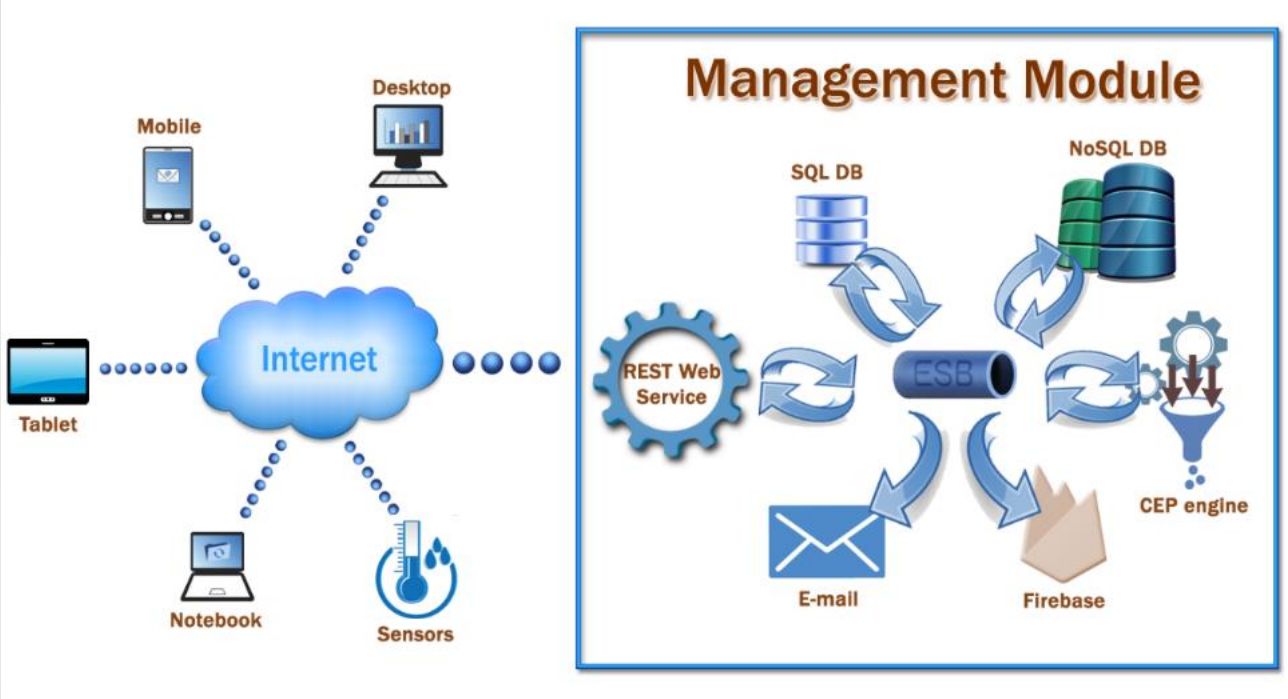
Air4People (Screenshots)



DOI: 10.3217/jucs-024-07-0846

SWAT (Motivation)

DOI: 10.1007/978-3-319-91764-1_18

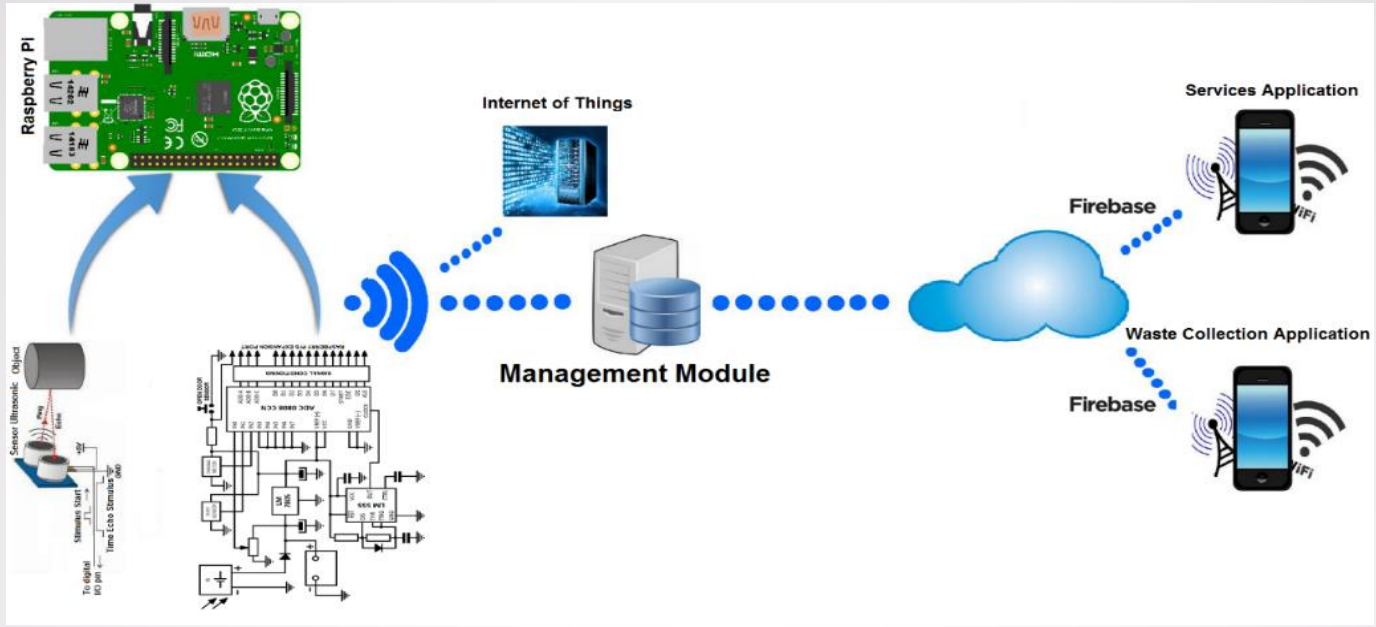


SWAT (Software Architecture)

DOI: 10.1007/978-3-319-91764-1_18

SWAT (Prototype)

DOI: 10.1007/978-3-319-91764-1_18

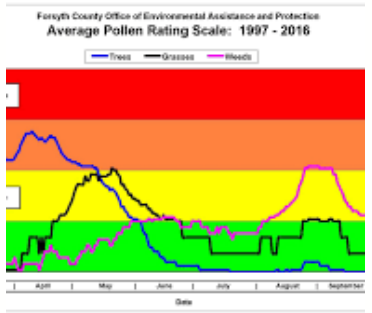


SWAT (Hardware Architecture)

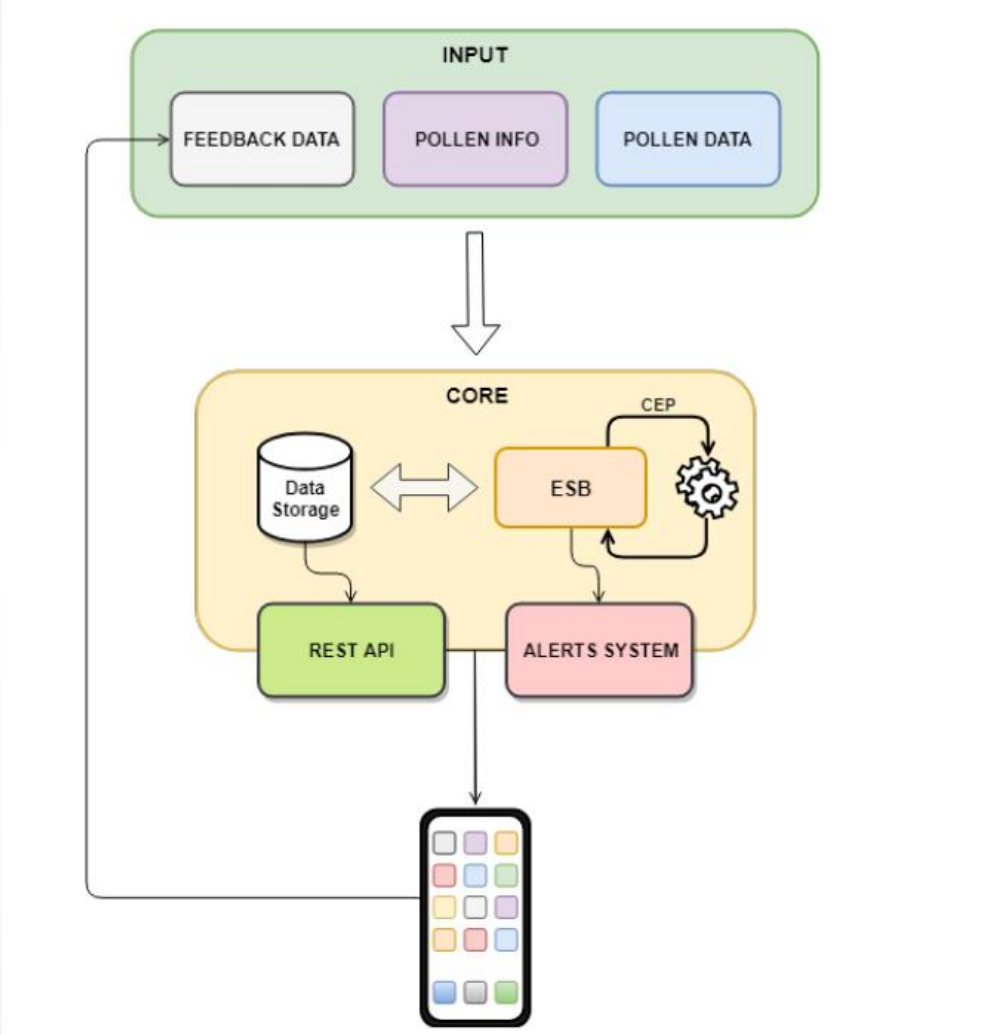
DOI: 10.1007/978-3-319-91764-1_18

AlergiApp (Motivación)

DOI: 10.1007/s11042-021-10759-6



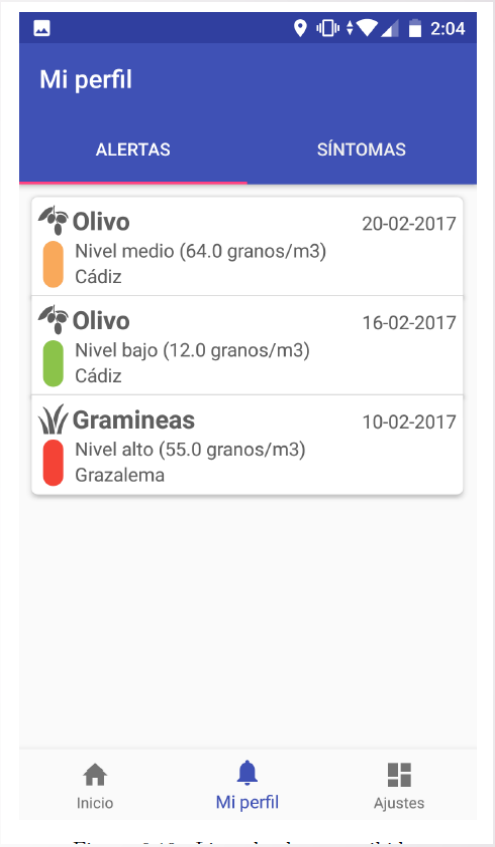
AlergiApp (Architecture)

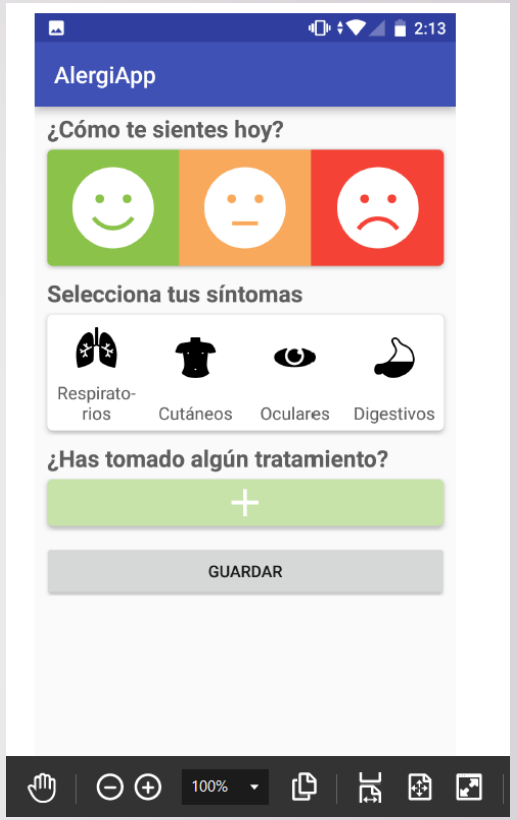


DOI: 10.1007/s11042-021-10759-6

AlergiApp (Screenshots)

DOI: 10.1007/s11042-021-10759-6






Concepts


Technologies


Architectures


Case Studies


Conclusion

Outline

Endless Case Studies and Application Domains

E-Health domain
Sustainability
Energy consumption
Natural resources management
Mobility
Traffic
Pollution
Emergencies
Economy
Governance
Security
Wellness
…

SMART CITIES require INTEROPERABILITY

Technologies

Software Architectures

All parties Collaboration

Smart Cities: the Challenge of Interoperable Software Architecture

- Thank-you very much for your attention
- Guadalupe Ortiz, UCASE Software Engineering Group
- guadalupe.ortiz@uca.es