European Light Rail Congress

ELASTIC

A novel software architecture for advanced mobility systems and autonomous transport networks

Elli Kartsakli
elli.kartsakli@bsc.es
Barcelona Supercomputing Center (BSC-CNS)

“The ELASTIC project has received funding from the European Union's Horizon 2020 research and innovation programme under the grant agreement No 825473”

mainspring

ELASTIC

The ELASTIC Project

- ELASTIC: a software architecture for Extreme-scaLe big-data AnalyticS in fog compuTIng eCosystems
  - Under the scope of the H2020 call ICT-12-2018-2020: Big Data technologies and extreme-scale analytics
  - 42 month project (starting Dec 2018); 6 million € budget
  - https://elastic-project.eu/

BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

isep Instituto Superior de Engenharia do Porto

ICE Information Catalyst
Services Data Software

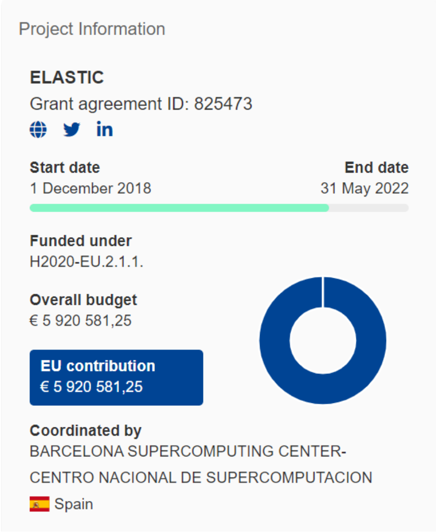
ikerlan

sixsq

THALES

CITTÀ METROPOLITANA DI FIRENZE

GEST



Project Information

ELASTIC
Grant agreement ID: 825473

in

Start date	**End date**
1 December 2018	31 May 2022

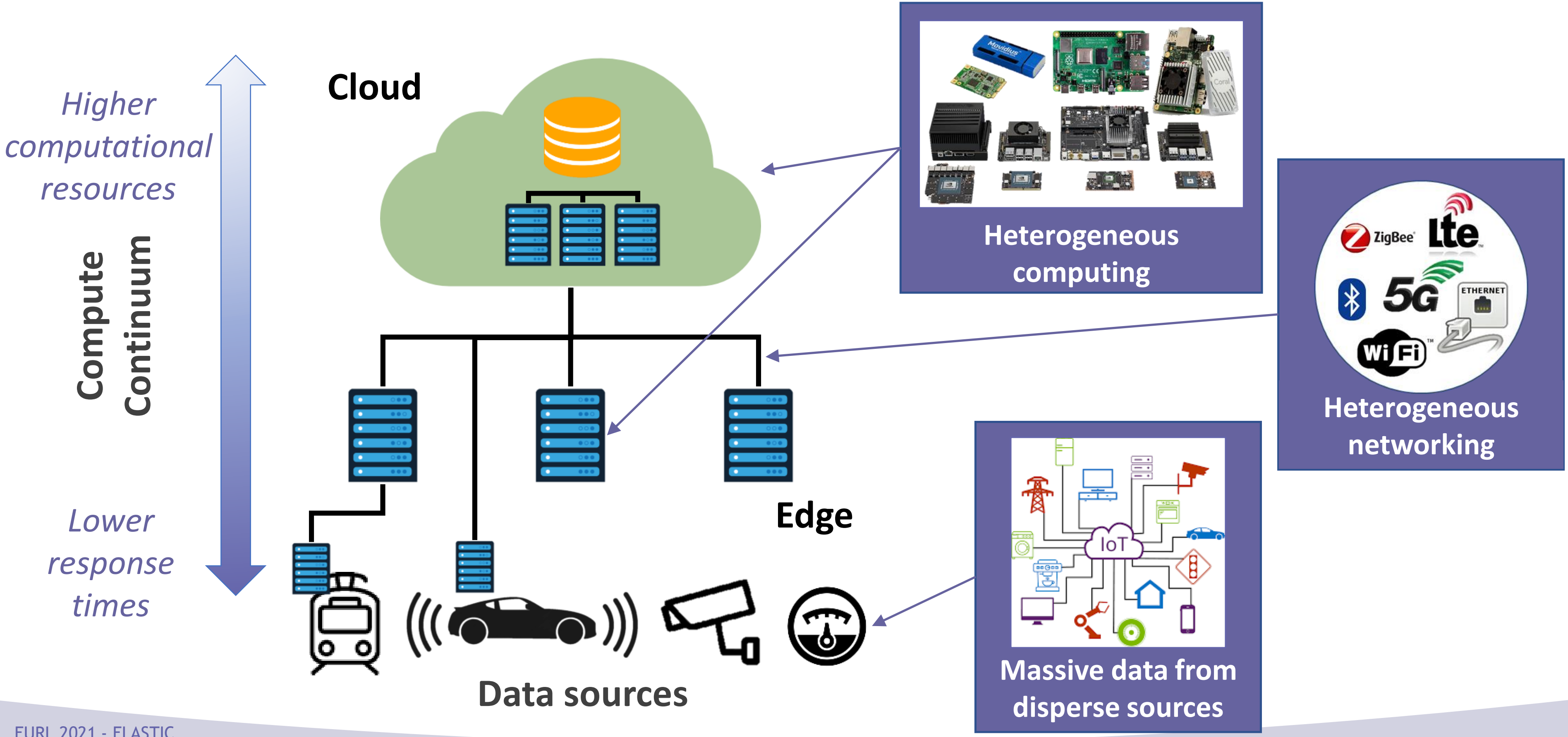
Funded under
H2020-EU.2.1.1.

Overall budget
€ 5 920 581,25

EU contribution
€ 5 920 581,25

Coordinated by
BARCELONA SUPERCOMPUTING CENTER-CENTRO NACIONAL DE SUPERCOMPUTACION
Spain

Challenges in the fog computing ecosystem



Higher computational resources

Compute Continuum

Lower response times

Cloud

Edge

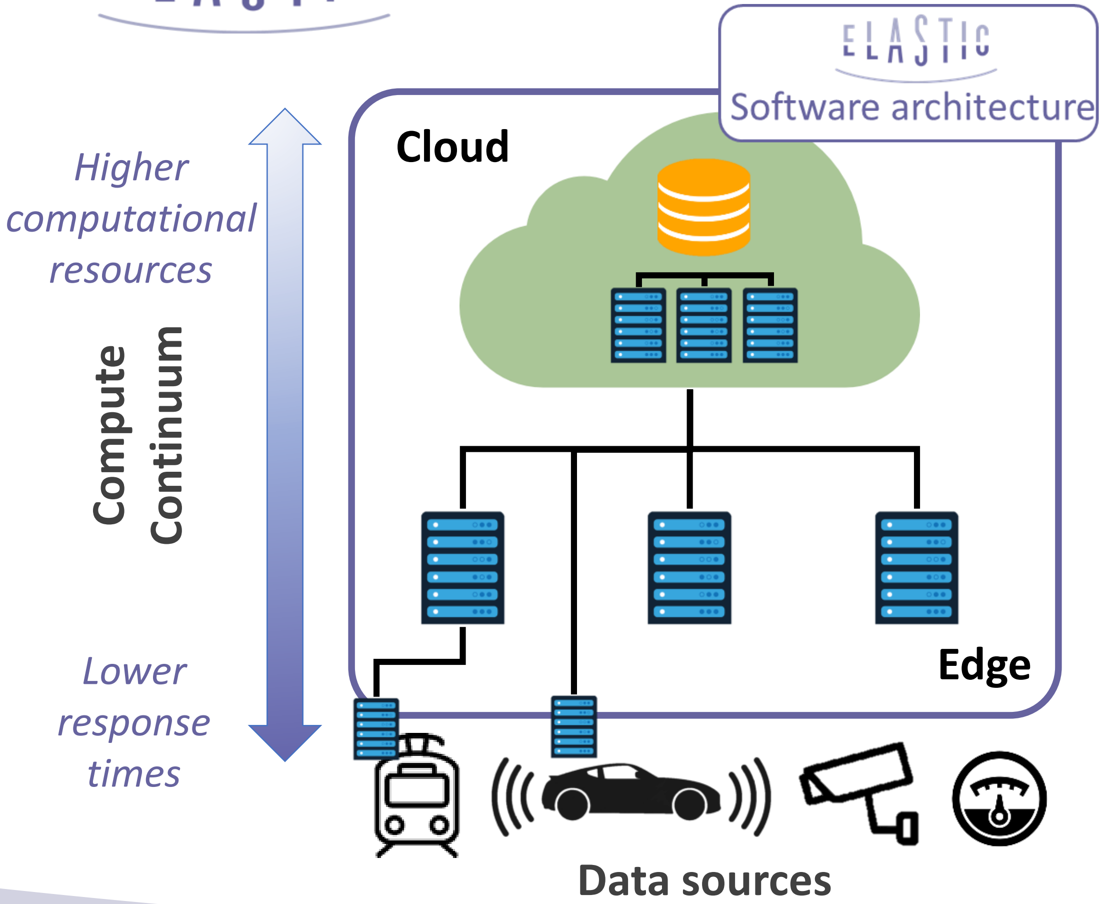
Data sources

Heterogeneous computing

Heterogeneous networking

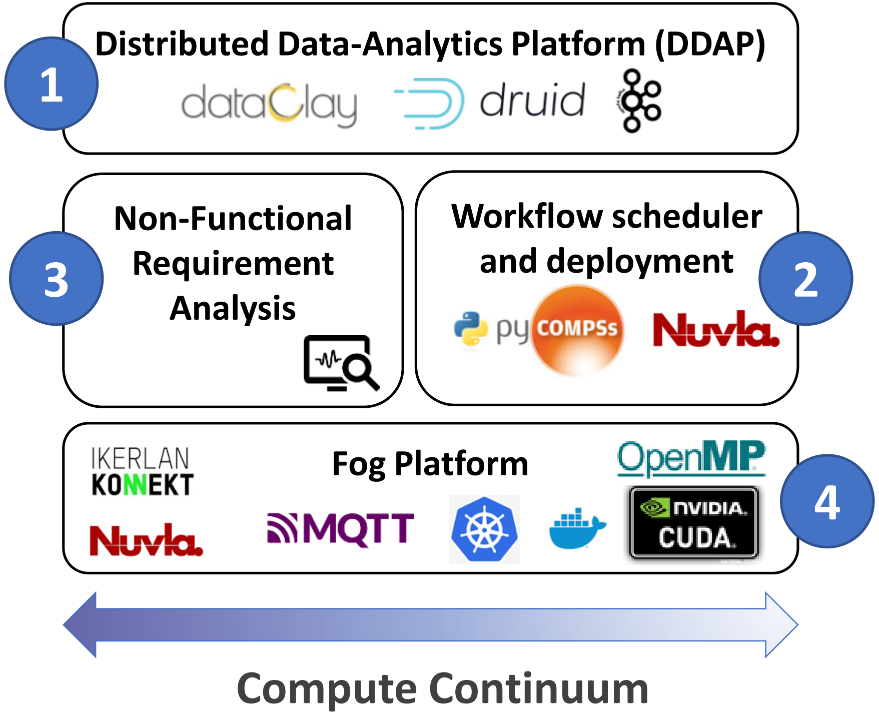
Massive data from disperse sources

The vision of ELASTIC



- ✓ *Facilitate* the development of ***complex data analytics***
- ✓ *Increase* the ***capabilities of the data analytics*** by distributing them across the compute continuum
- ✓ *Fulfill* the ***non-functional properties*** inherited from the application domain
- ✓ *Exploit* ***advanced parallel and energy-efficient*** embedded platforms at the edge

Main Contribution:
The ELASTIC Software Architecture


1
Distributed Data-Analytics Platform (DDAP)
dataClay
druid
3
Non-Functional Requirement Analysis
Workflow scheduler and deployment
2
pyCOMPSs
Nuvla.
IKERLAN KONNEKT
Fog Platform
OpenMP
Nuvla.
MQTT
NVIDIA CUDA
4
Compute Continuum

1. Powerful API for the development of advanced **data-analytics workflows,** supported with a **Distributed Data platform (DDAP)**
2. **Advanced orchestration methods** workflow scheduling and deployment
3. **Non-functional analysis** inherited from the cyber-physical domain
4. Fog-based platforms including
   - **Cloud-based** Container as a Service (Caas)
   - **IoT** cyber-secured communication
   - Advanced highly parallel and energy-efficient **edge** platforms

ELASTIC Smart mobility use case – City of Florence

Towards autonomous vehicles

Towards enhanced maintenance services

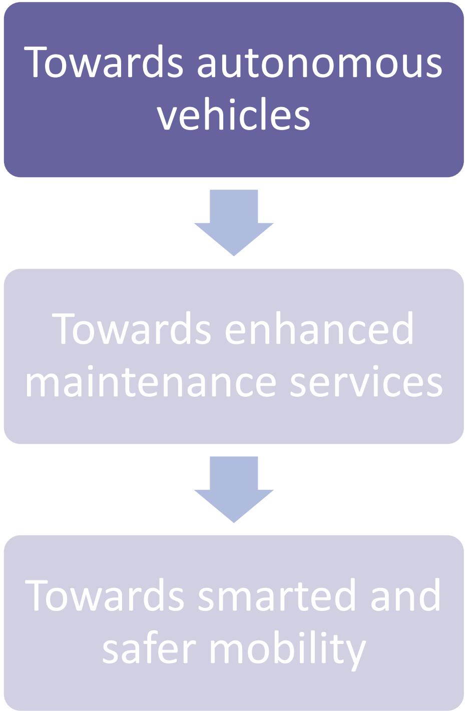
Towards smart and safer mobility

THALES

GEST

CITTÀ METROPOLITANA DI FIRENZE

ELASTIC Smart mobility use case


Towards autonomous vehicles

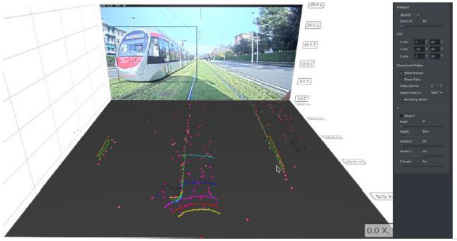
Towards enhanced maintenance services

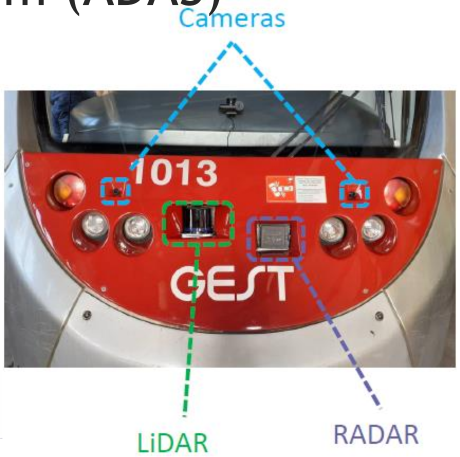
Towards smarted and safer mobility

✓ Next Generation Autonomous Positioning (NGAP)

✓ Advanced Driving Assistant System (ADAS)

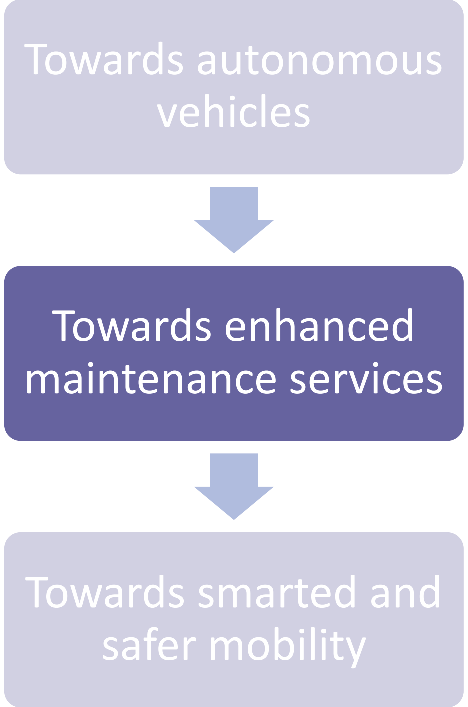
- Sensorized trams
  - Inertial Measurement Unit (IMU)
  - Odometer
  - GPS
  - Lidar
  - Radar
  - Cameras





- Advanced parallel edge processor platforms
- WiFi and LTE connectivity

ELASTIC Smart mobility use case

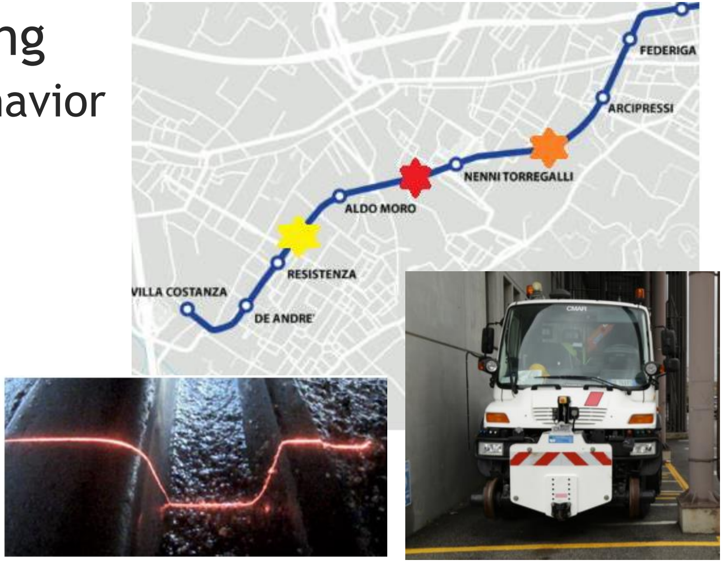

Towards autonomous vehicles

Towards enhanced maintenance services

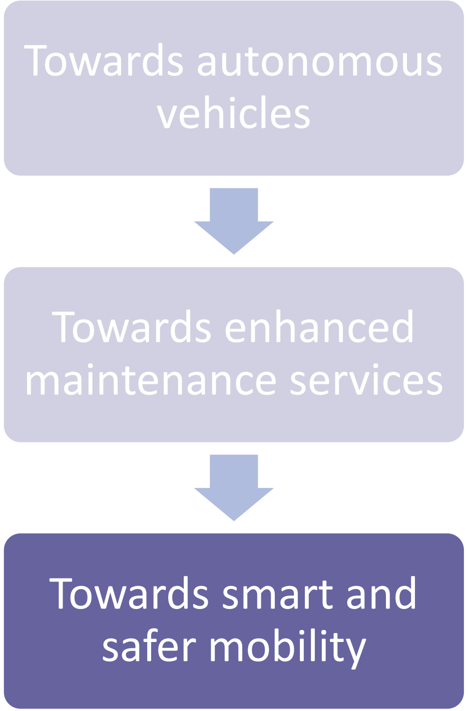
Towards smarted and safer mobility

- ✓ Predictive maintenance
  - Track profiling and automatic detection of track wear
- ✓ Energy efficient driving
  - Profiling of driving behavior

- Sensors employed
  - Laser measuring heads
  - Accelerometers
  - Cameras
  - Odometers
  - Inertial platform
  - Energy extraction unit



ELASTIC Smart mobility use case



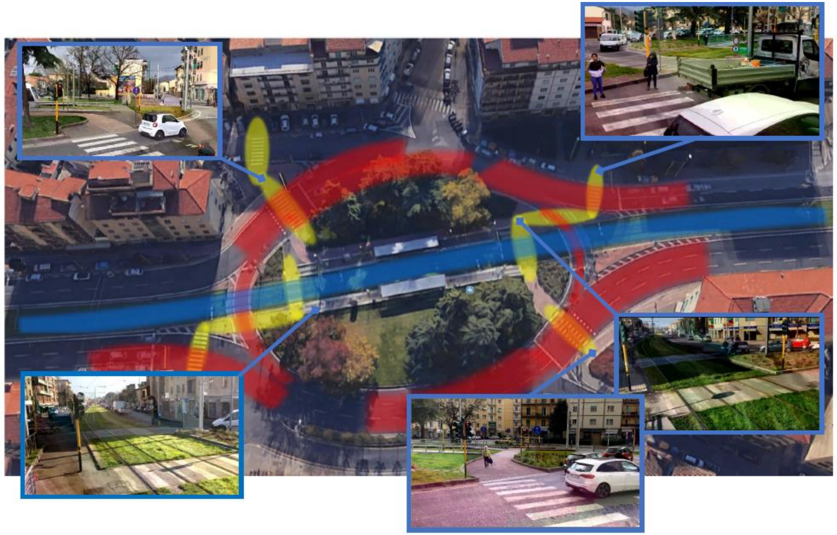
Towards autonomous vehicles

Towards enhanced maintenance services

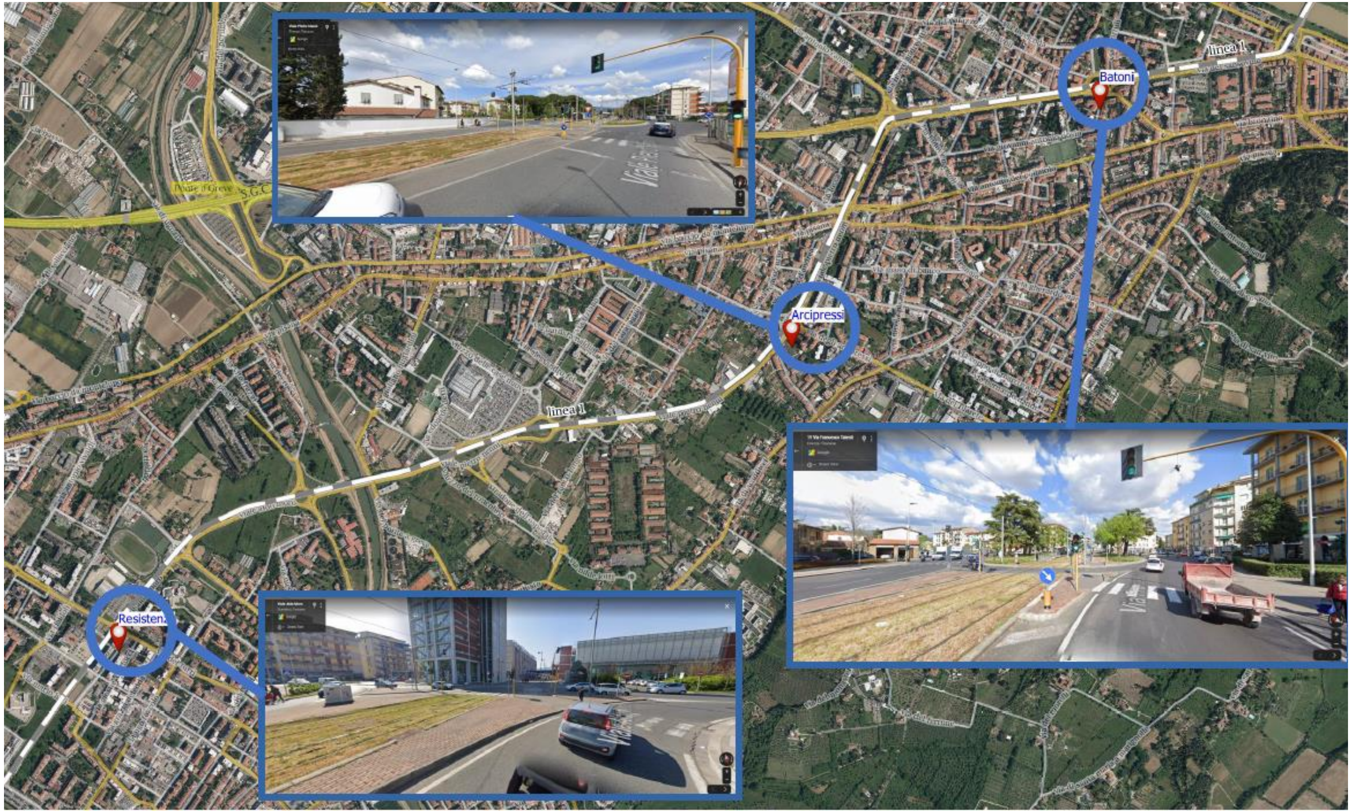
Towards smart and safer mobility

✓ Interaction between the public and the private transport in the City of Florence

- Real-time event detection and hazard alerts
- Offline analytics for traffic management



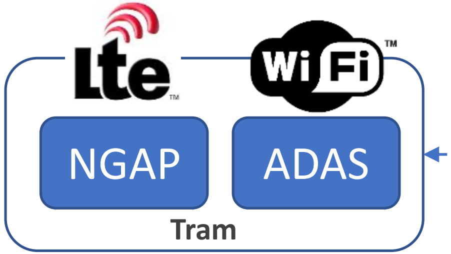
Three pilot intersections

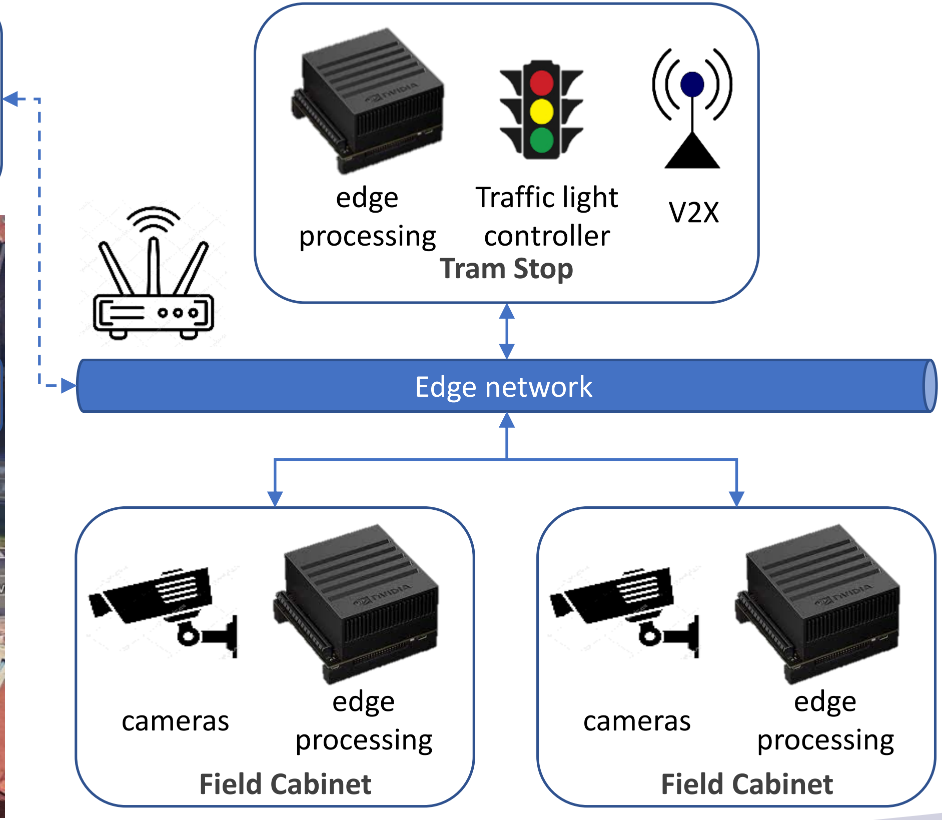


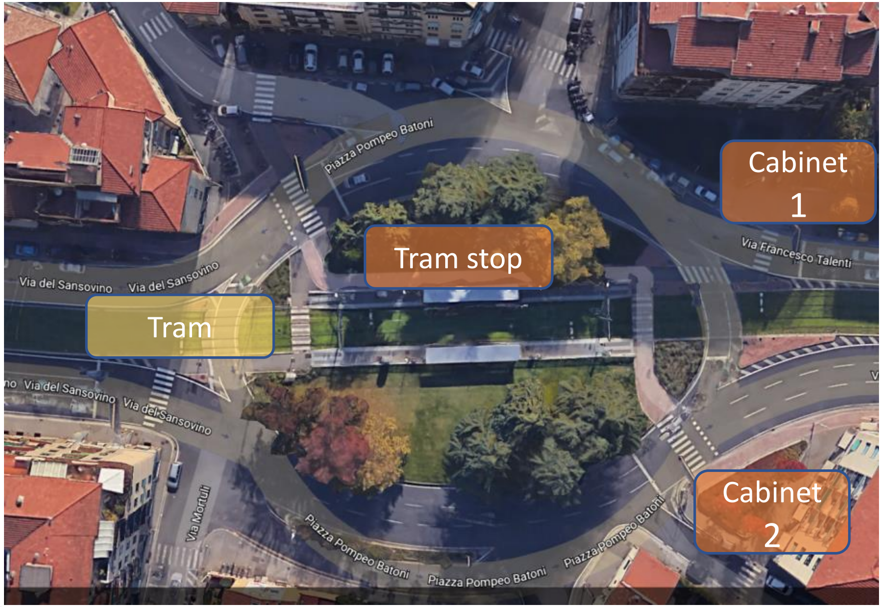
- ✓ 2 cameras per intersection
- ✓ 3 processing edge platforms
- ✓ Traffic light controller
- ✓ V2X infrastructure to alert vehicles

ELASTIC

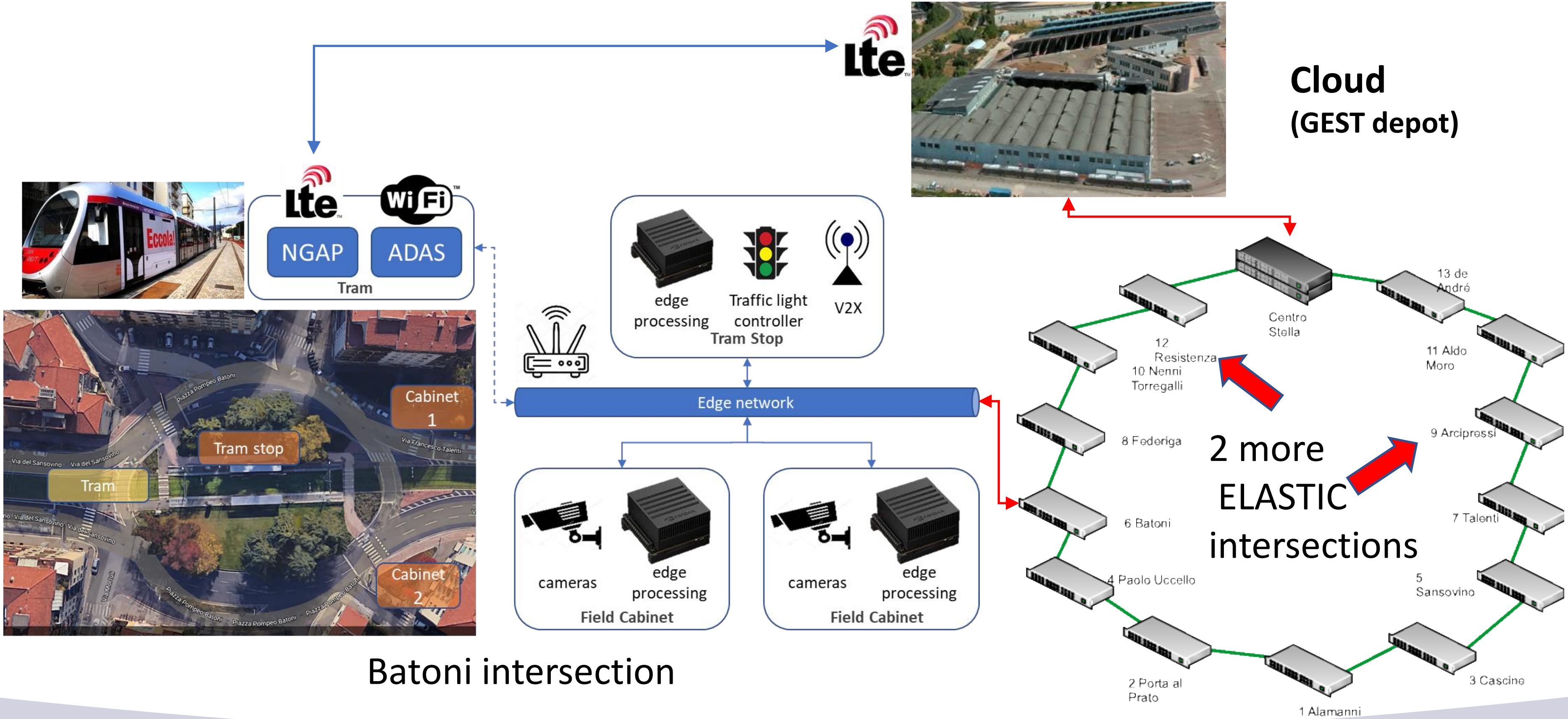
Edge infrastructure







The ELASTIC Compute Continuum



Cloud
(GEST depot)

Tram
NGAP
ADAS

edge processing
Traffic light controller
V2X
Tram Stop

Edge network

cameras
edge processing
Field Cabinet

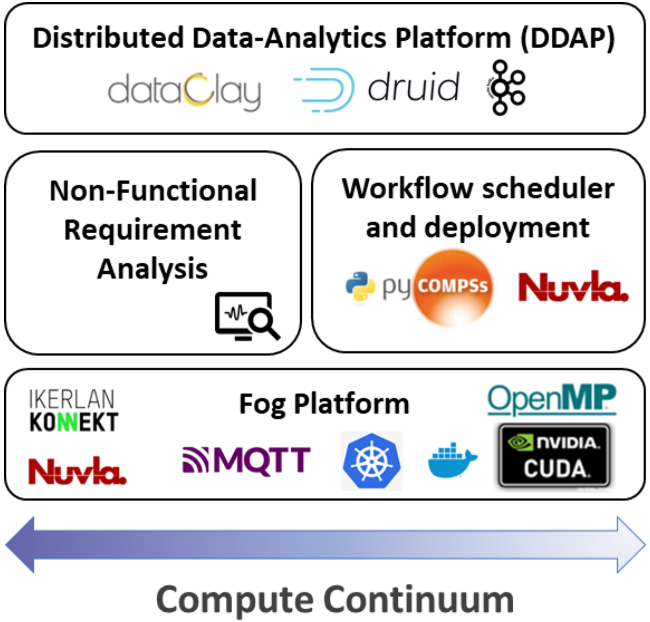
cameras
edge processing
Field Cabinet

Cabinet 1
Tram stop
Tram
Cabinet 2

2 more ELASTIC intersections

Batoni intersection

ELASTIC Software Architecture



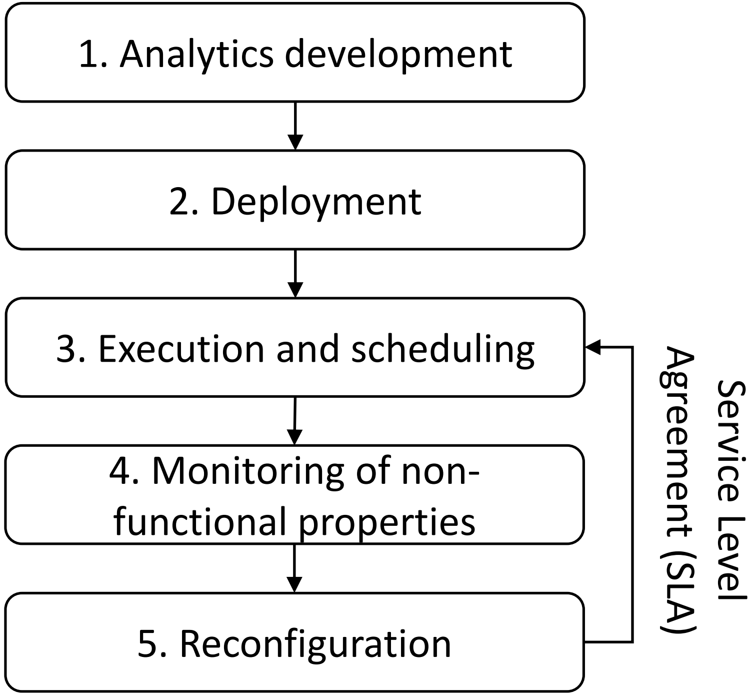
Distributed Data-Analytics Platform (DDAP)

Non-Functional Requirement Analysis

Workflow scheduler and deployment

Fog Platform

Compute Continuum



1. Analytics development
2. Deployment
3. Execution and scheduling
4. Monitoring of non-functional properties
5. Reconfiguration

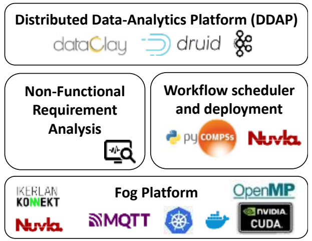
Service Level Agreement (SLA)

ELASTIC

The Data-Analytics Workflow

Data Analytics Methods

1. Sensor fusion (ADAS)
2. Tram position (NGAP)
3. Object detection
4. UTC/Supervisor consolidation
5. Data fusion
6. Data aggregation
7. Dashboard
8. Hazard detection
9. Alert visualization (cars/trams)
10. Electric power consumption
11. Defect Detector

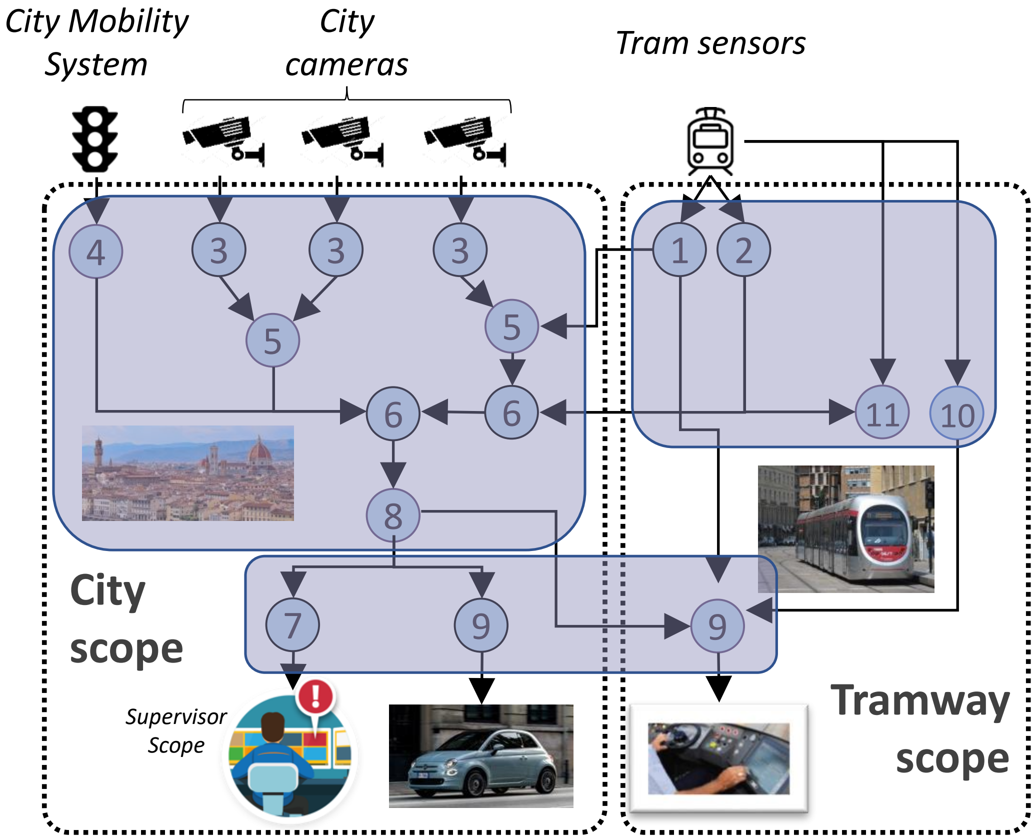


Distributed Data-Analytics Platform (DDAP)

Non-Functional Requirement Analysis

Workflow scheduler and deployment

Fog Platform



City Mobility System

City cameras

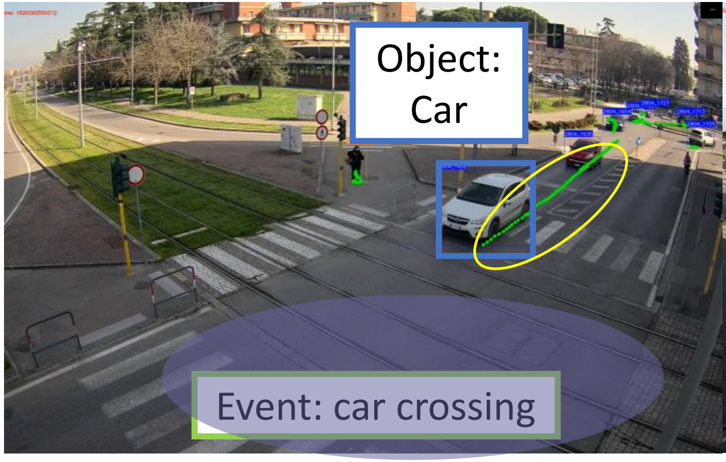
Tram sensors

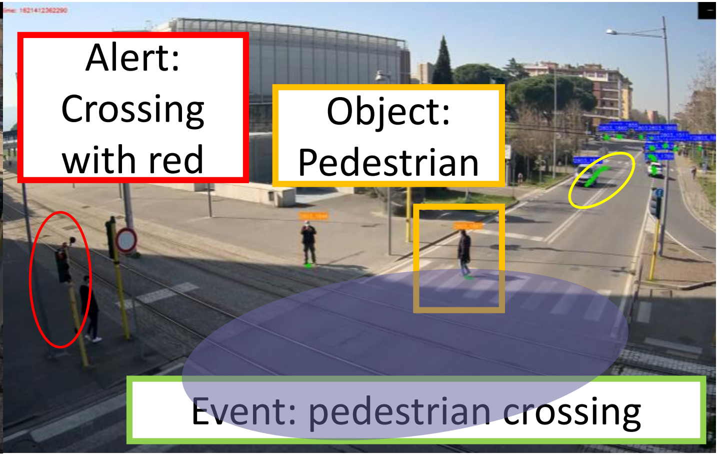
City scope

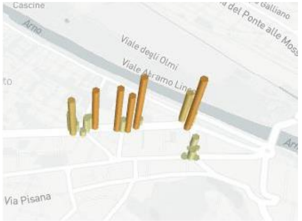
Tramway scope

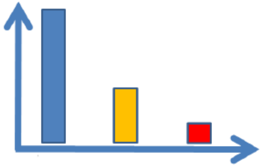
Supervisor Scope

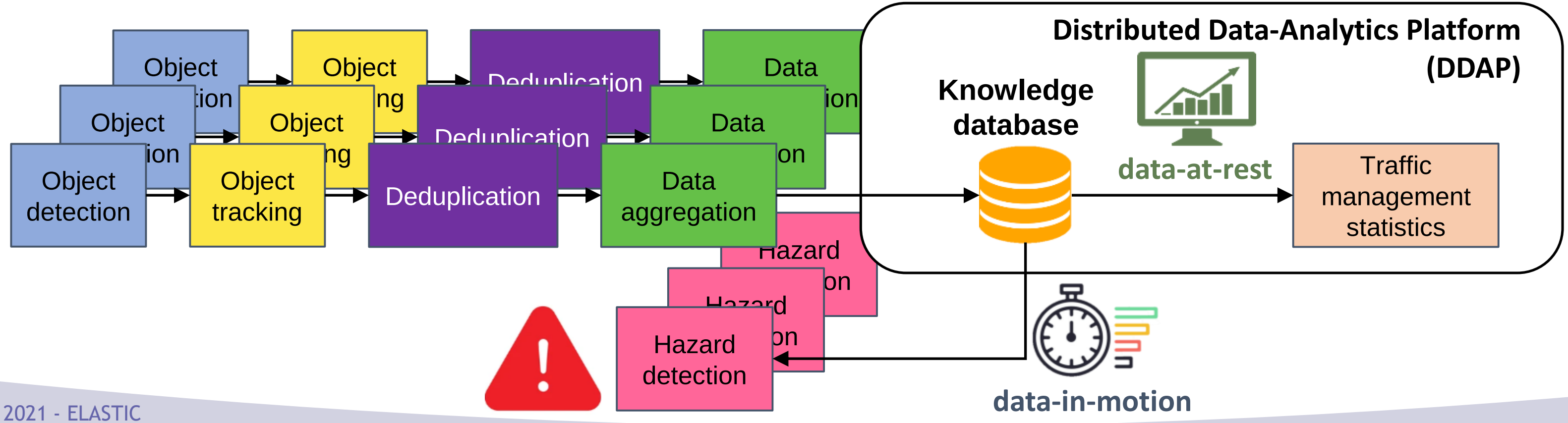
Data analytics: from data to knowledge











Object detection

Object tracking

Deduplication

Data aggregation

Hazard detection

Distributed Data-Analytics Platform (DDAP)

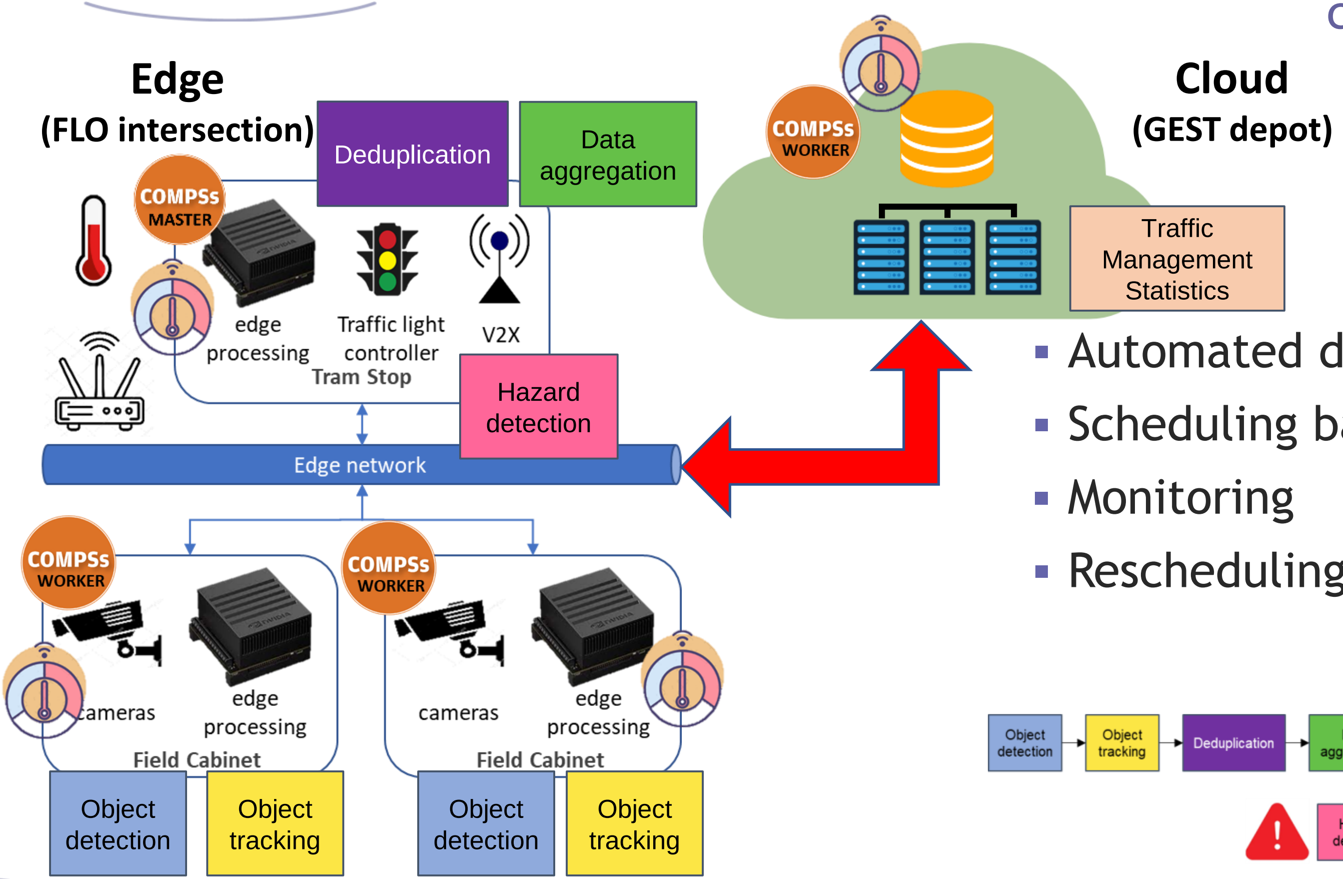
Knowledge database

data-at-rest

Traffic management statistics

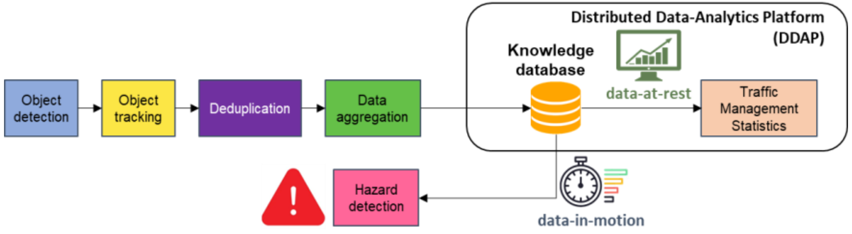
data-in-motion

Deployment and distributed execution across the compute continuum



http://compss.bsc.es

- Automated deployment
- Scheduling based on heuristics
- Monitoring
- Rescheduling on-the-fly



Conclusions

- ELASTIC proposes a **novel software architecture** for the ***development***, ***distribution*** and ***execution*** of
  - ✓ complex analytics workflows over the compute continuum
  - ✓ in a way transparent to programmer and agnostic to infrastructure
  - ✓ while guaranteeing non-functional requirements inherited from the application domain
  - ✓ supporting the reallocation of resources on-the-fly
  - ✓ enhancing the overall ***programmability***, ***portability*** and ***performance***
- The ELASTIC software architecture is being validated in a smart mobility use case involving the Florence tramway network, towards smarter, safer, efficient and autonomous transportation

ELASTIC

A Software Architecture for Extreme-ScaLe
Big-Data AnalyticS in Fog CompuTIng Ecosystems

www.elastic-project.eu

Stay Tuned!

elli.kartsakli@bsc.es

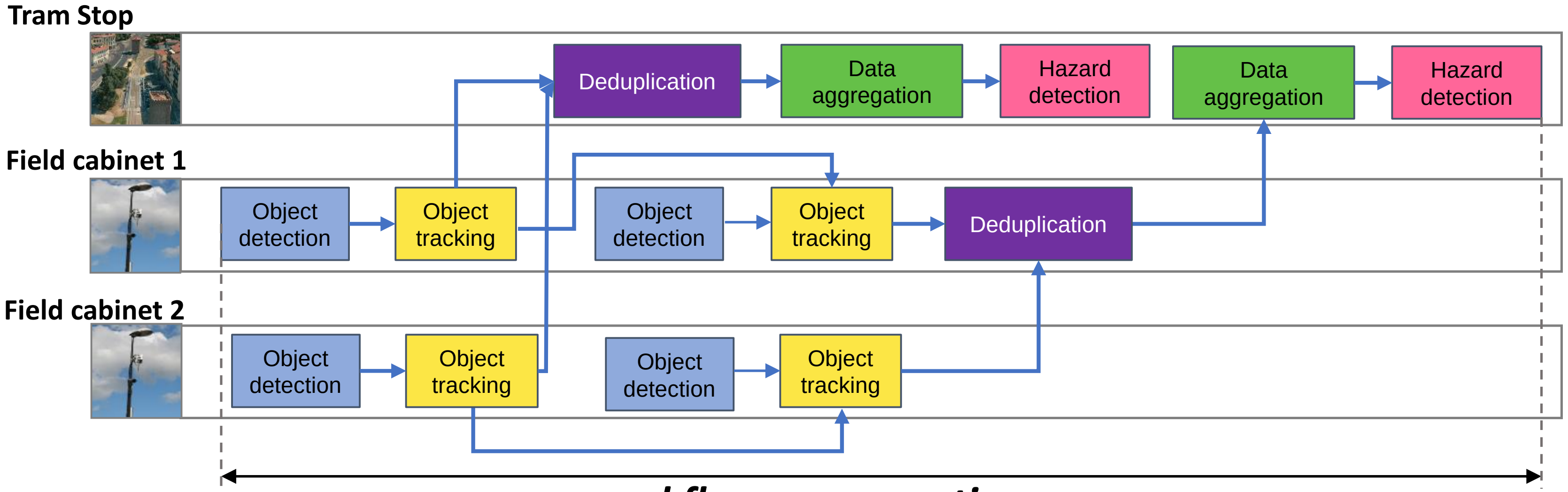
@elastic_EU

www.linkedin.com/company/elastic-project

ELASTIC

Workflow orchestration

COMPSs http://compss.bsc.es



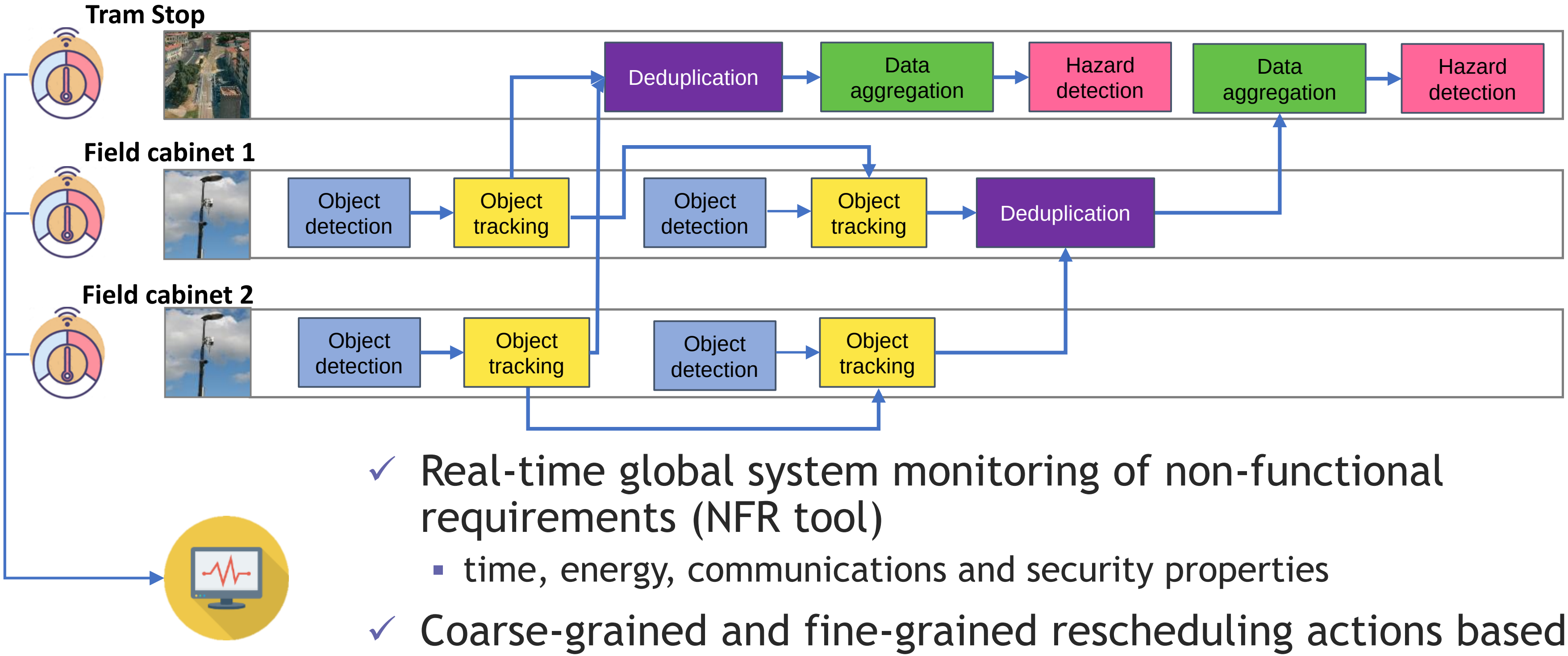
workflow response time

- Scheduling heuristics that minimize the end-to-end response time
- Exploit parallelism of applications and edge platforms

ELASTIC

Real-time monitoring and reconfiguration

COMPSs http://compss.bsc.es



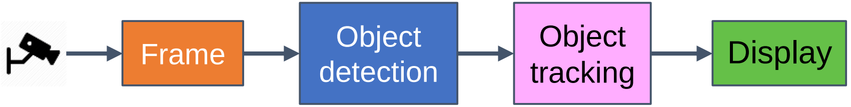
✓ Real-time global system monitoring of non-functional requirements (NFR tool)

- time, energy, communications and security properties

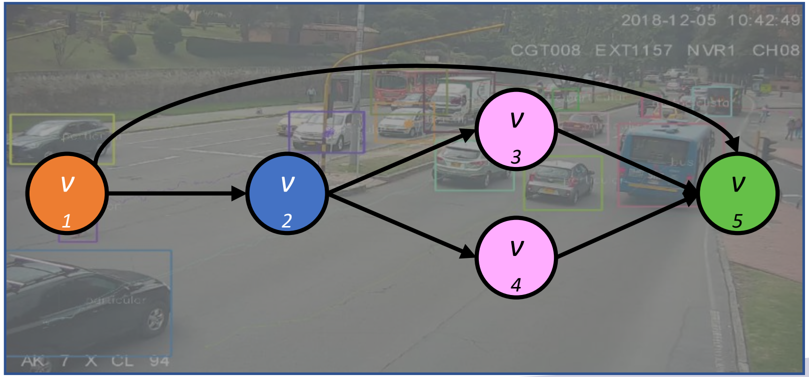
✓ Coarse-grained and fine-grained rescheduling actions based on recommendations and scheduling policies

Application Example –
Task based programming

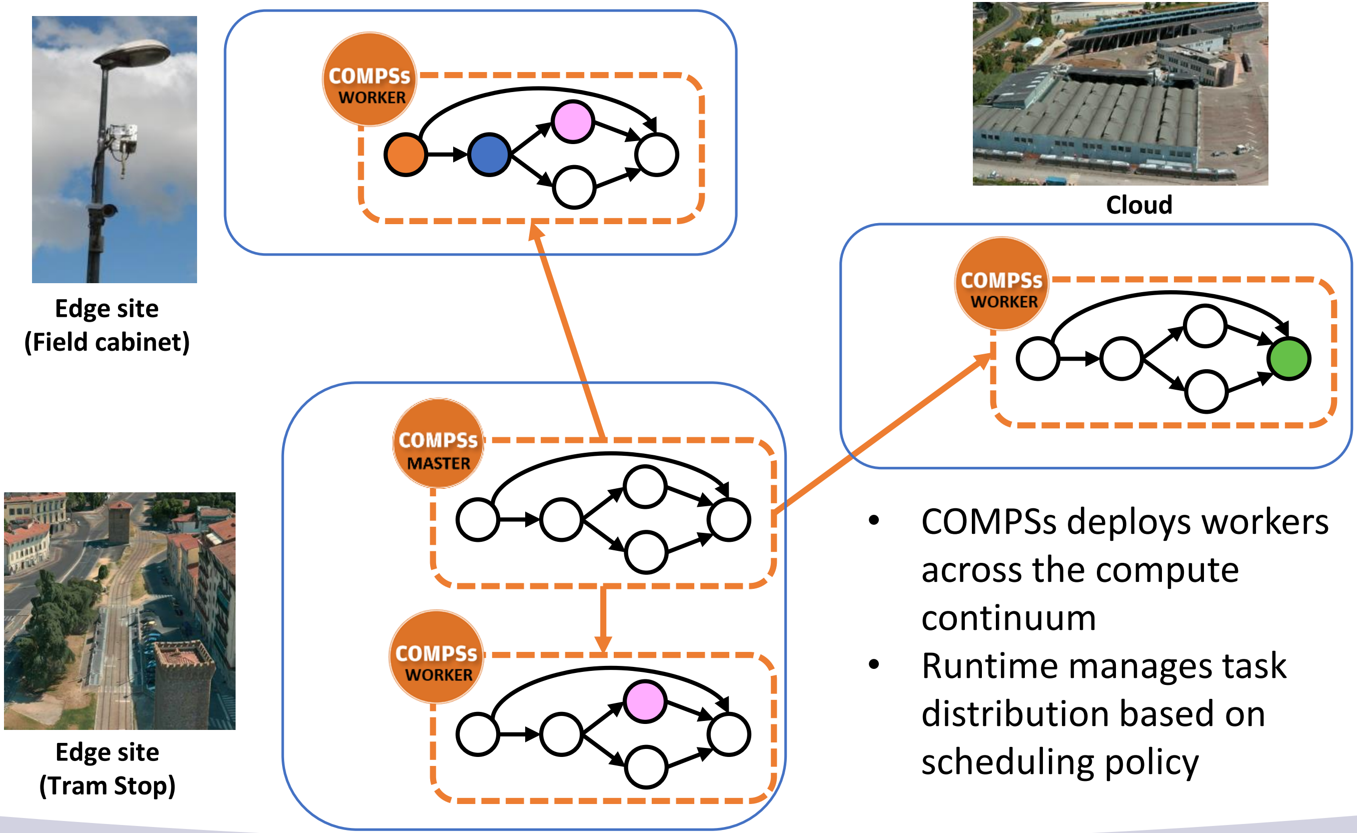
```
@task(returns=numpy.ndarray)
def get_frame():
    return get_next_frame_from_video()

@task(frame=IN, returns=list)
def get_objects_from_frame(frame):
    return yolo.detect(frame)

@task(list_objects=IN)
def tracker(list_objects):
    return track(list_objects)

@task(list_objects=IN, frame = IN)
def collect_and_display(list_objects, frame):
    for obj in list_objects:
        display(obj, frame)

## Main function ##
while True:
    frame = get_frame()
    list_obj = get_objects_from_frame(frame)
    for i in range(len(list_obj)):
        list_obj[i] = tracker (list_obj)
    collect_and_display (list_obj, frame)
```



- Write sequential code
- Annotate tasks to be distributed with @task and identify their dependencies
- ✓ COMPSs will create the task graph and distribute the tasks

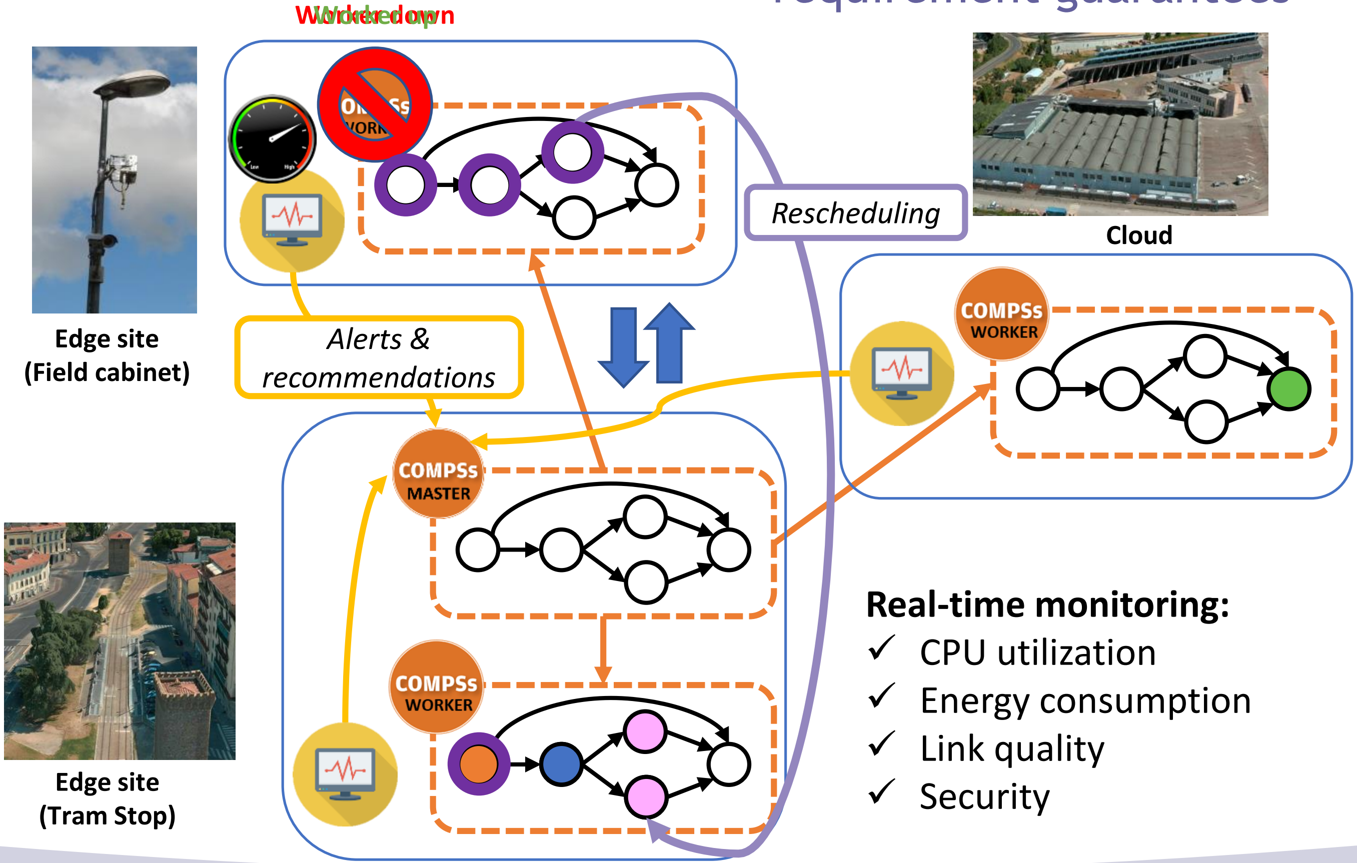


Application Example – Distributed deployment



- COMPSs deploys workers across the compute continuum
- Runtime manages task distribution based on scheduling policy

Application Example – Real-time monitoring for non-functional requirement guarantees



Real-time monitoring:

- ✓ CPU utilization
- ✓ Energy consumption
- ✓ Link quality
- ✓ Security