



A Software Architecture for Extreme-Scale
Big-Data Analytics in Fog Computing Ecosystems

Non-functional requirements in the ELASTIC architecture

DeCPS 2019 - Workshop on Challenges and new Approaches for Dependable and
Cyber-Physical Systems Engineering

Luís Miguel Pinho



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

14/06/2019

- **ELASTIC:** A Software Architecture for Extreme-ScaLe Big-Data AnalyticS in Fog CompuTing ECosystems
- 3-year H2020 RIA project (Dec-2018, Nov-2021)
- **Website:** <https://elastic-project.eu/>
- **Coordinator:** Eduardo Quiñones, BSC, Spain
- **Partners**

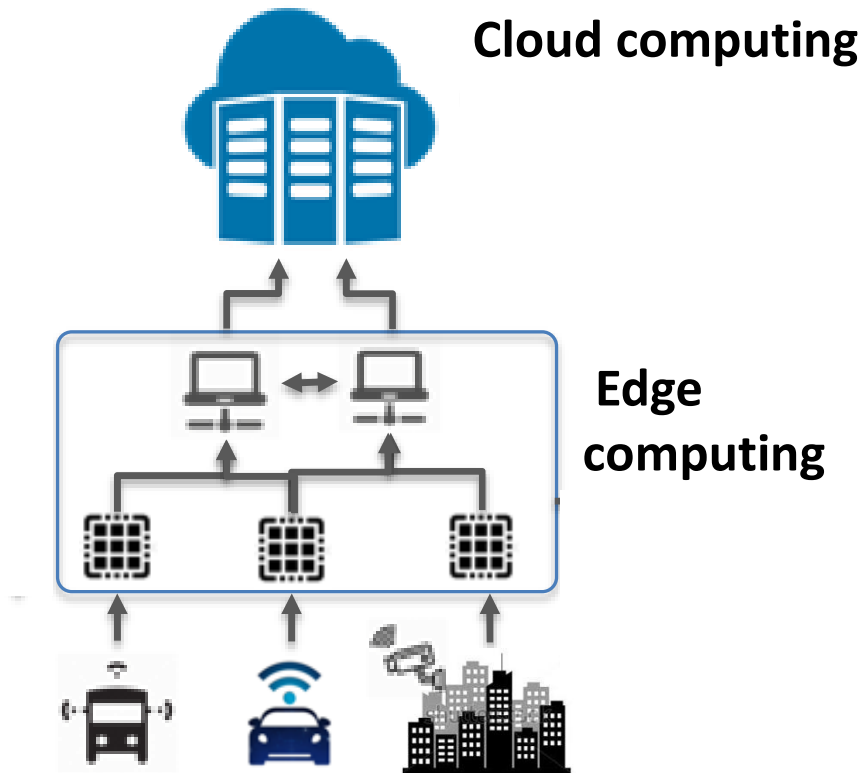


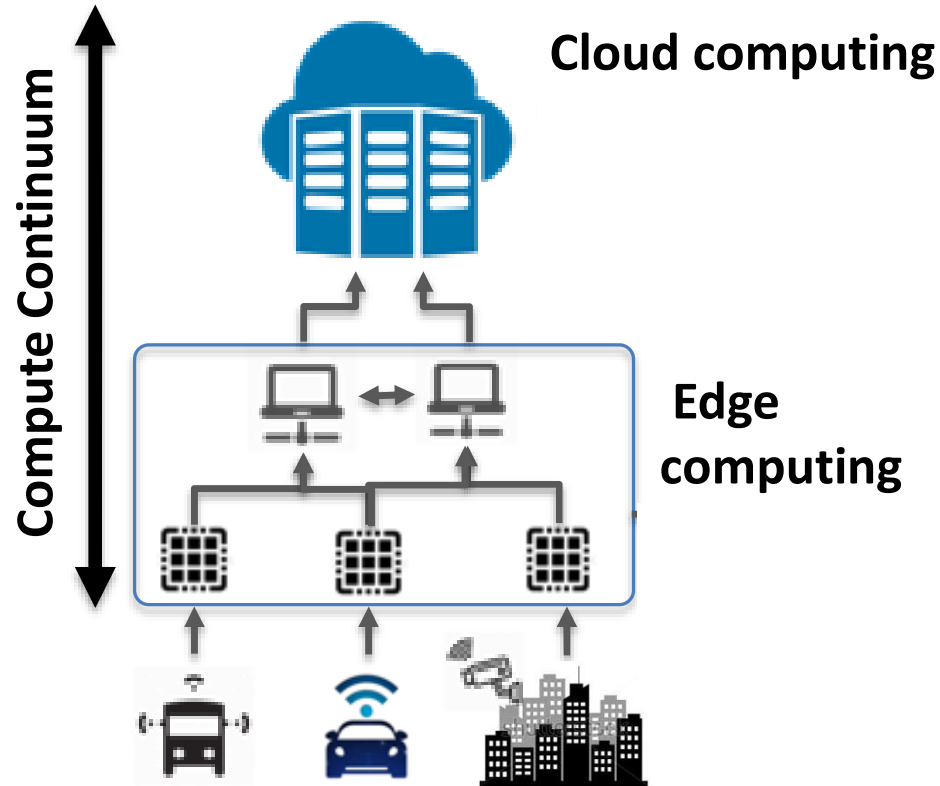
- Motivation
- ELASTIC Concept
- Programming Model
- Use Cases
- Non-Functional Requeriments
 - Time
 - Energy
 - Communication Quality
 - Security

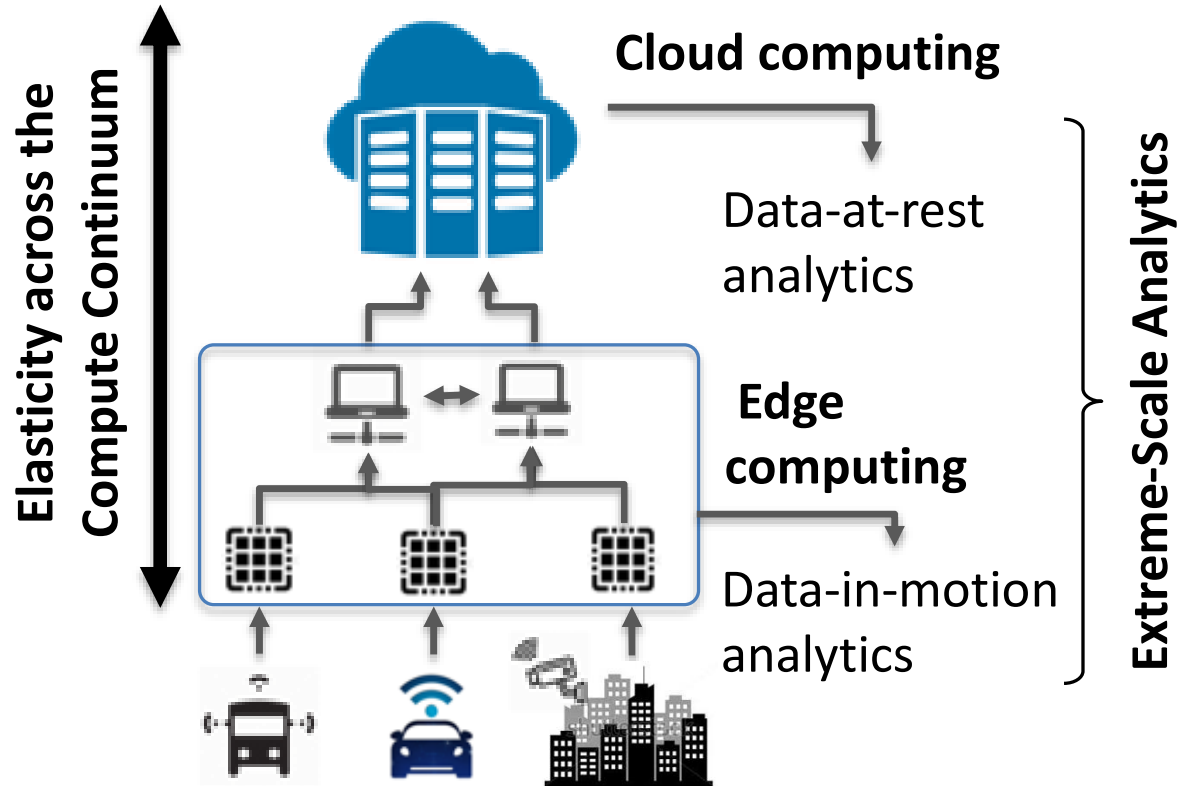
- Extreme-scale analytics are more and more a key enabling application for smart systems
 - process huge amounts of heterogenous data, geographically dispersed, both on the fly and at rest
 - necessity to fulfil non-functional properties inherited from the system (real-time, energy efficiency, communication quality or security)
- Providing the required computing capacity for extreme-scale analytics is of paramount importance
 - dynamically manage resources as needed, guaranteeing required levels of service
 - consider the full architecture of the system, from the Edge devices to cloud infrastructures

- Challenge: fulfil non-functional properties
 - including real-time, energy-efficiency, quality of communications, security
 - need to consider these in a holistic way, as they are interdependent
- Challenge: limits of the existent elasticity concept
 - in which cloud computing resources are orchestrated to provide maximum throughput
 - does not take into account the computing resources located on the edge
 - elasticity mainly focuses on system throughput, without taking into account the non-functional properties
- Need to address these two challenges along the compute continuum, i.e., from the edge to the cloud
 - paramount importance to take full benefit of extreme-scale analytics in smart systems, in industrial and societal environments
 - there are no known end-to-end solutions applied along the complete compute continuum

- ELASTIC will develop a software architecture incorporating a new elasticity concept
 - Efficiently distribute the workloads across the compute continuum, whilst guaranteeing non-functional properties
- The fog paradigm and the extreme-scale analytics promoted by ELASTIC fits with current big data analytics design trends
 - On one hand, the priority may be to provide quick and reactive information, possibly in real-time, based on the flowing stream of data, which typically implies focusing on the most relevant aspects of the stream (data-in-motion)
 - On the other hand, the priority may be to provide thorough and more consistent responses, which typically implies aggregating as much information as possible into larger models (data-at-rest)
 - Despite their capabilities being complementary, both approaches have been historically tackled separately, given their apparently incompatible system requirements.
- The new elasticity concept promoted by ELASTIC will leverage the fog computing paradigm to fully exploit the benefits of both approaches into a single, transparent, extreme-scale analytics solution







- ELASTIC software architecture will take into consideration a number of trade-offs
 - performance, precision/accuracy, non-functional system properties
 - dynamic management of computation
- Edge devices may deliver the time-predictability needed to implement real-time functionalities
 - but do not provide sufficient computational power to run analytics
 - fast and time-predictable, but limited, precision algorithms will be deployed on the edge-side for data-in-motion
- Cloud computing resources provide the computation capabilities to support complex analytics
 - but communication delays may make systems unstable
 - cloud resources will be used to run only accurate but costly models for the long-term refinement and global modelling

- Computation Distribution based on **COMPSs**
 - Software framework developed by BSC for applications targeting distributed infrastructures
- Implements a **task-based programming model** for Python, C, C++, Java
 1. Tasks identifies **units of parallelism** to be scheduled in other computing resources
 2. Defines in/out **data dependencies** among tasks
 3. Defines **constraints** on task scheduling
- Simple linear address space and agnostic of computing platform

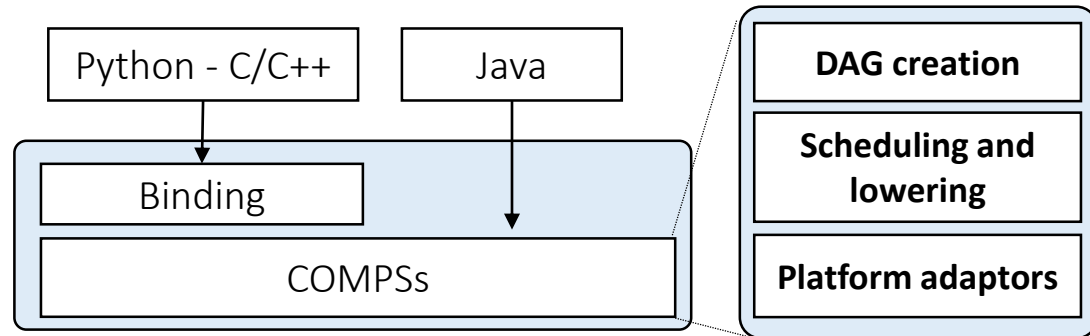
```
def pow(a, b):
    b = a*a
    ...
for i in range(MSIZE):
    pow(A[i], B[i])
```



```
@constraint(ComputingUnits=4)
@task(a=IN,b=OUT)
def pow(a, b):
    b = a*a
    ...
for i in range(MSIZE):
    pow(A[i], B[i])
```

Internal Structure of COMPSs

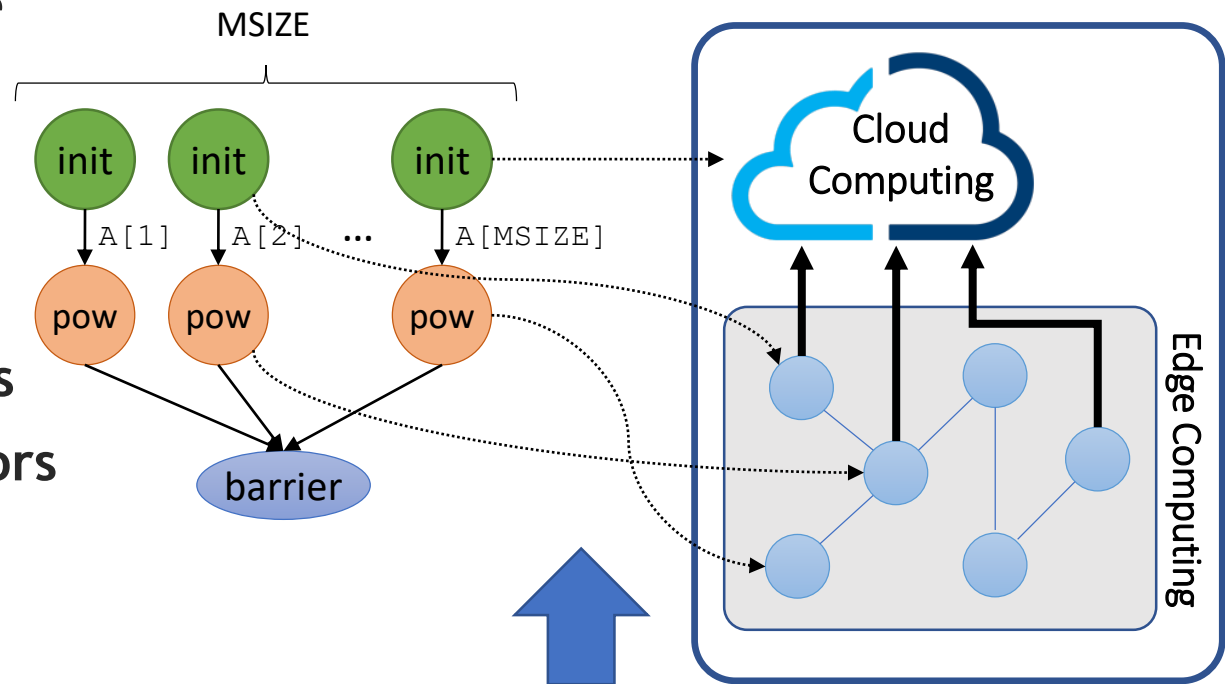
- **DAG creation** identifies the data dependencies and creates the *task dependency graph* (TDG) at run-time
- **Scheduling and lowering** distributes tasks to compute resources and transforms operators to calls to move data
- **Platform adaptor** provides the interface to interact with the computing resources below



```

@constraint (ComputingUnits=4)
@task (a=IN, b=OUT)
def pow(a, b):
    b = a*a
    
```

1. **New scheduling** techniques across the compute continuum
2. **New constraints** considering the distribution of the continuum and **non-functional properties**
3. **New platform adaptors** supporting the fog architecture



Scheduling of big-data workloads along the compute continuum fulfilling **non-functional properties**



- A realistic set of use-cases from the smart mobility domain, applied in a light rail system
 - Next generation autonomous positioning
 - Advanced driving assistant system
 - Interaction between the public and the private transport
 - Predictive maintenance

- The architecture is intended to be generic
 - Requirements from other domains have also been considered
 - Automotive, Avionics, Smart Manufacturing



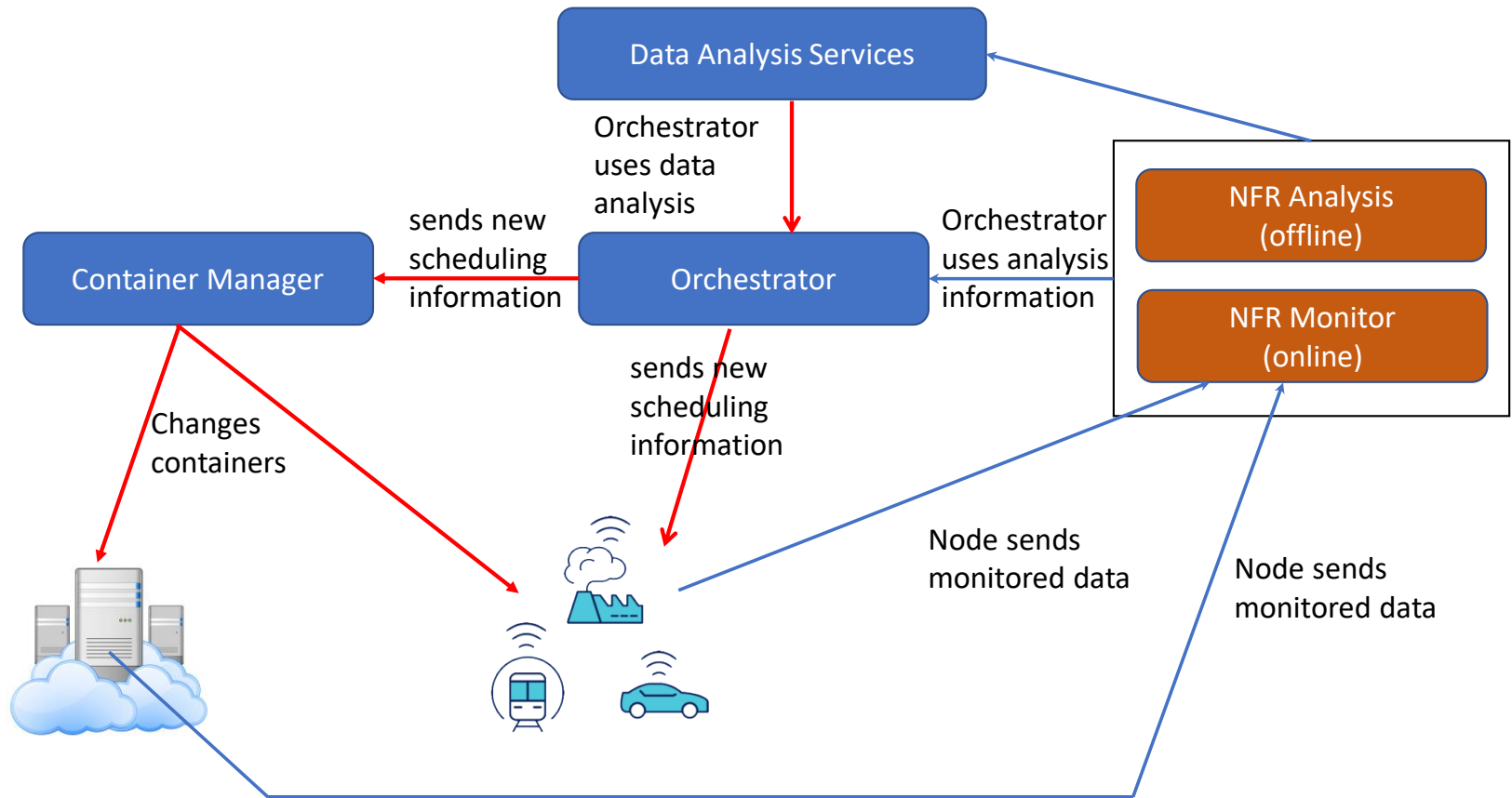
ELASTIC

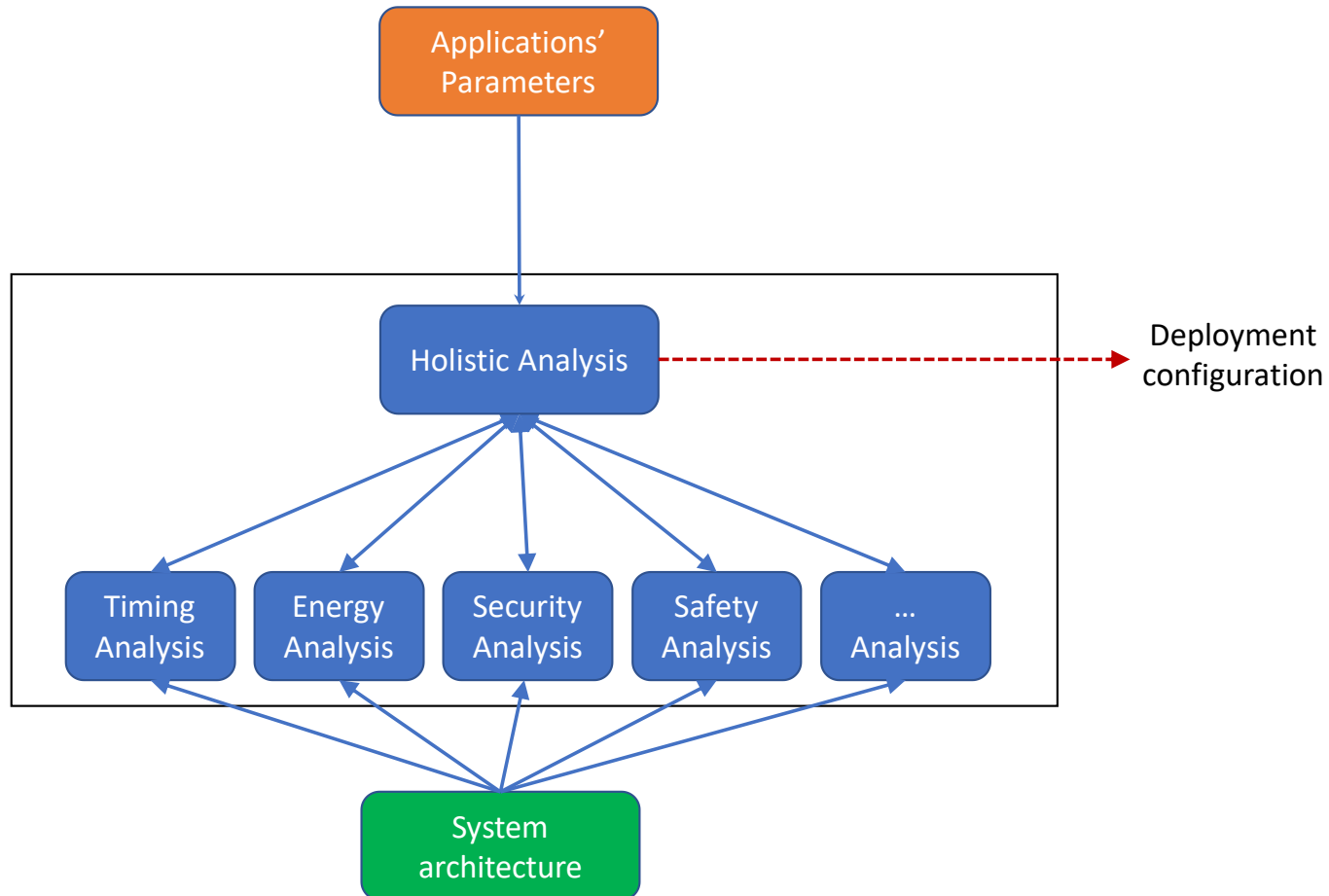
A Software Architecture for Extreme-Scale
Big-Data Analytics in Fog Computing Ecosystems

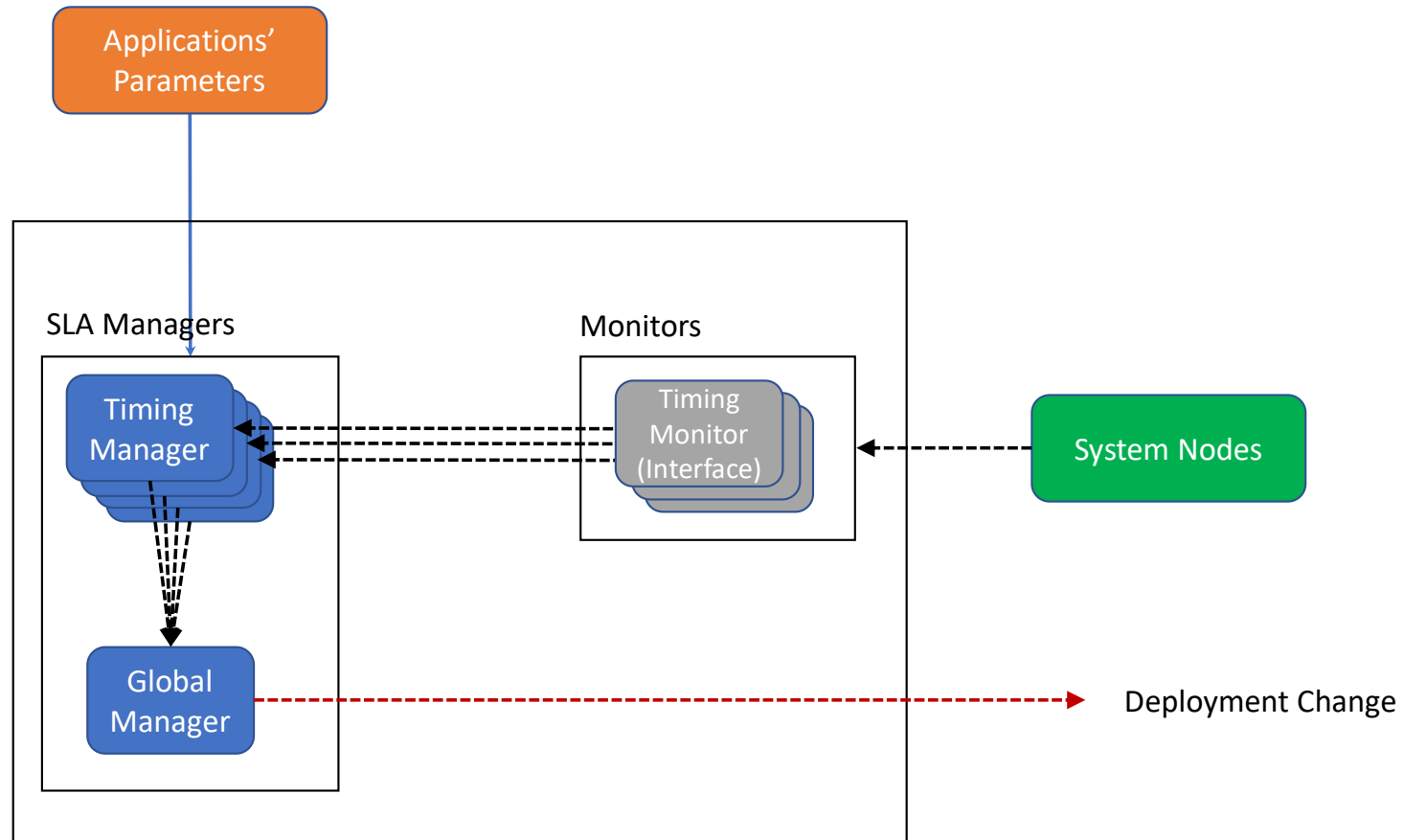
Non-Functional Requeriments

- Real-time computing
 - Real-time data analytics is becoming a main pillar in industrial and societal ecosystems, with the combination of different data sources and prediction models within real-time control loops
 - Unfortunately, the use of remote cloud technologies makes infeasible to provide real-time guarantees due to the large and unpredictable communication costs on cloud environments
- Mobility shows even increased trade-offs and technological difficulties
 - Mobile devices are largely constrained by the access to energy
 - Mobile devices suffer from unstable communication, which may increase random communication delays, unstable data throughput, loss of data and temporal unavailability.
- Security is a continuously growing priority
 - affects data integrity, confidentiality and potentially safety.
 - Strict security policy management may hinder the communication among services and applications, shrinking overall performance and real-time guarantees.

- Interdependency between non-functional properties provides further challenges
 - Processing time and energetic cost of computation is reduced as data analytics is moved to the cloud ...
 - ... but the end-to-end communication delay and the latency of the system increases and becomes unpredictable
 - As computation is moved to the cloud, the required level of security increases to minimise potential attacks
 - ... which may end up affecting the safety assurance levels, hindering the execution and data exchange among edge and cloud resources.
- The ELASTIC architecture must include
 - mechanisms which allow the specification of the required level of non-functional properties
 - the offline analysis of the parameters to determine an appropriate system configuration which enables fulfilment of requirements
 - an online monitoring and analysis capability which is able to trigger configuration changes upon detection of level violations







- Time-related requirements, e.g.
 - Hard and soft real-time control loops
 - Response-time in the order of milliseconds
 - Processing rates in the order of thousand/sec
 - Worst-case execution time estimates and analysis
 - Support to multi-core and many-core
 - Support to mixed-criticality

- Energy-related Requirements, e.g.
 - Energy monitoring capabilities
 - Energy-efficiency at the Edge
 - Speed scaling techniques
 - Multiple modes of operation

- Communication-related Requirements, e.g.
 - Tens of required protocols
 - Dataflow QoS
 - Critical flows
 - Latency and bandwidth requirements
 - End-to-end reliability, integrity
 - Support to priorities

- Security-related requirements, e.g.
 - Authentication
 - Encryption
 - Secure access to data
 - Adherence to security standards
 - Specific GDPR-related requirements

- ELASTIC intends to develop a software architecture for fog ecosystems
 - incorporating a new elasticity concept, efficiently distributing workloads across the compute continuum
- ELASTIC targets smart systems, where non-functional requirements are of paramount importance
 - Time, energy, communication quality, security
- A specific component of the architecture will deal with non-functional properties of applications
 - Guiding the ELASTIC orchestrator and component managers



A Software Architecture for Extreme-Scale
Big-Data Analytics in Fog Computing Ecosystems

Thank you

Contact: LMP@isep.ipp.pt



[@elastic_EU](https://twitter.com/elastic_EU)



www.linkedin.com/company/elastic-project

www.elastic-project.eu