CISTER - Research Center in
Real-Time & Embedded Computing Systems

QoS for High Performance IoT Systems

Students: Renato Ayres; Paulo Barbosa;

Supervisors: Luis Lino Ferreira, Michele Albano, José Silva, Paulo Baltarejo Sousa;

Index

- Motivation
- Arrowhead Framework
- Vision of the Solution/QoS support in Arrowhead
  - QoSManager
  - QoSMonitor
- Pilot Project
- FTT-SE
- Vision of the Solution/Arrowhead with FTT-SE
- Video
- Conclusions

Motivation

- QoS is a central service to guarantee communication robustness. It is vital in certain systems.
- The Arrowhead Framework has a Service Oriented Architecture (SOA) and does not provide QoS support.

Main Objectives

- Developing an architecture that provides QoS support in Arrowhead compliant systems. Capable of working with different communication protocols and QoS requirements.
- Developing a pilot project using FTT-SE protocol.

Objectives in Arrowhead

1. Implement **delay** and **bandwidth** QoS requirements.
2. Verify the feasibility of QoS objectives.
3. Setup devices to ensure the QoS.
4. Monitor, in real time, the performance of services.
5. Detect if a QoS parameter is not being guaranteed anymore, or any other critical event.
6. Integrate it with the Arrowhead Framework.
7. Integrate it with the FTT-SE.

Planning - Development Team Roles

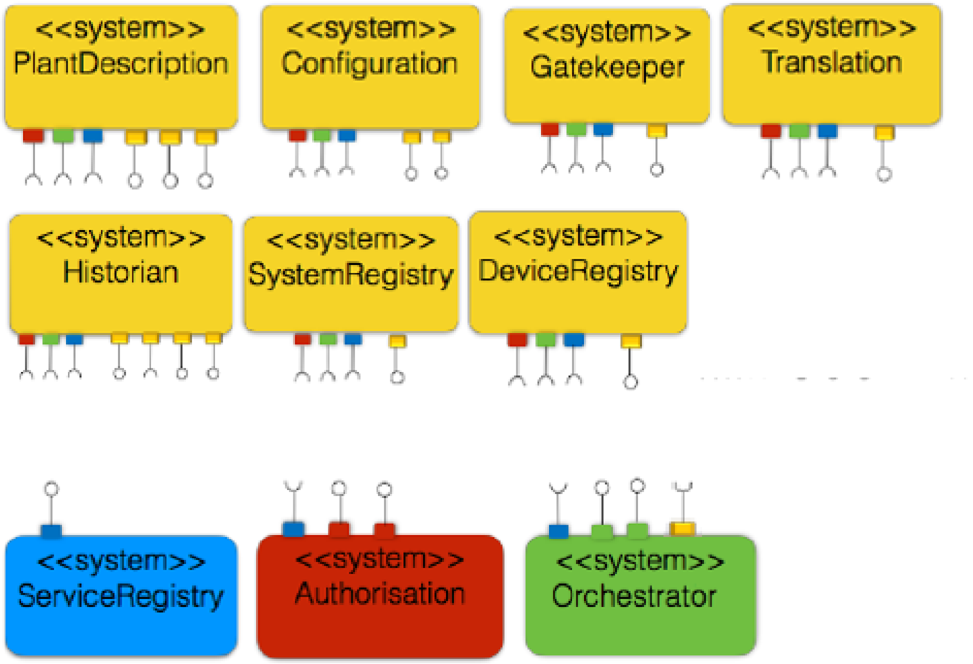
- Paulo Barbosa:
  - **Management** of QoS
  - Pilot Project
- Renato Ayres:
  - **Monitoring** of QoS
  - Pilot Project

ARROWHEAD

Arrowhead Project

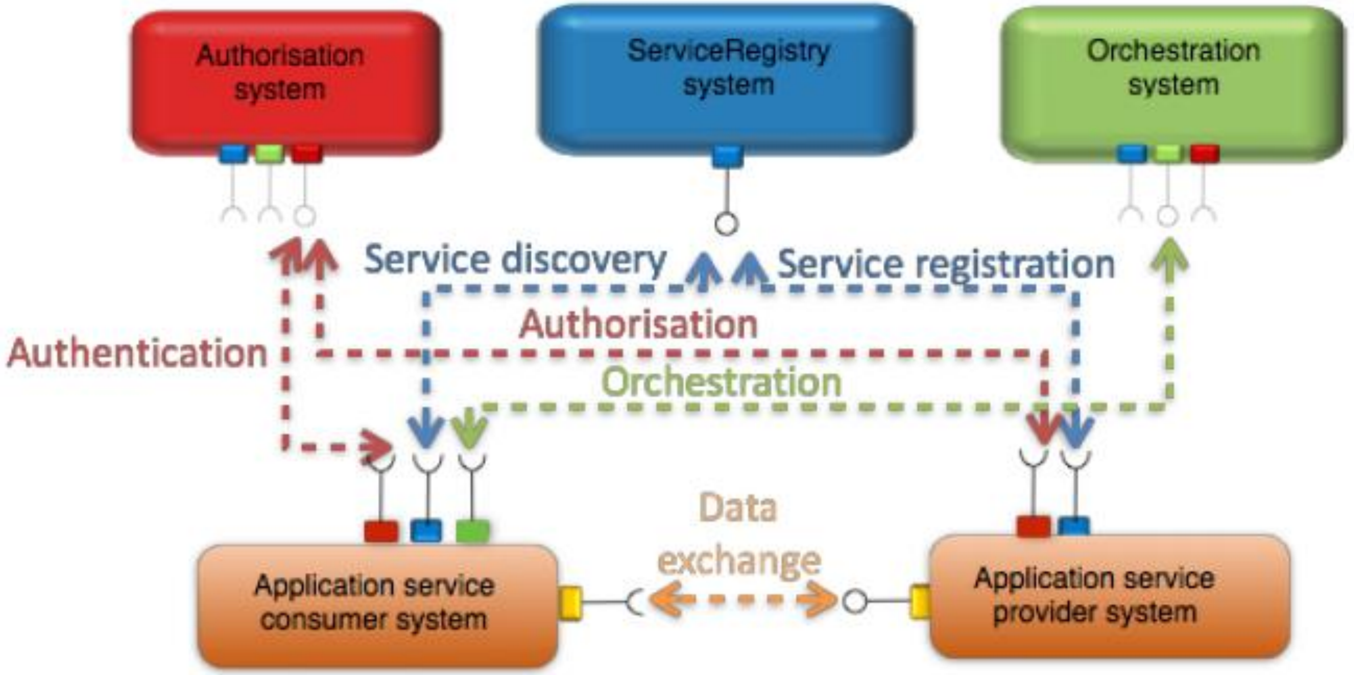
- **Challenge:** Interoperability & Integrability of services provided by almost any device
- **Goal:** Collaborative automation by network embedded devices supported by SOA
- **Applicative Domains:**
  - Electro-mobility
  - Smart buildings, infrastructures and smart cities
  - Industrial production
  - Energy production
  - Energy virtual market

Arrowhead Framework



Arrowhead Systems

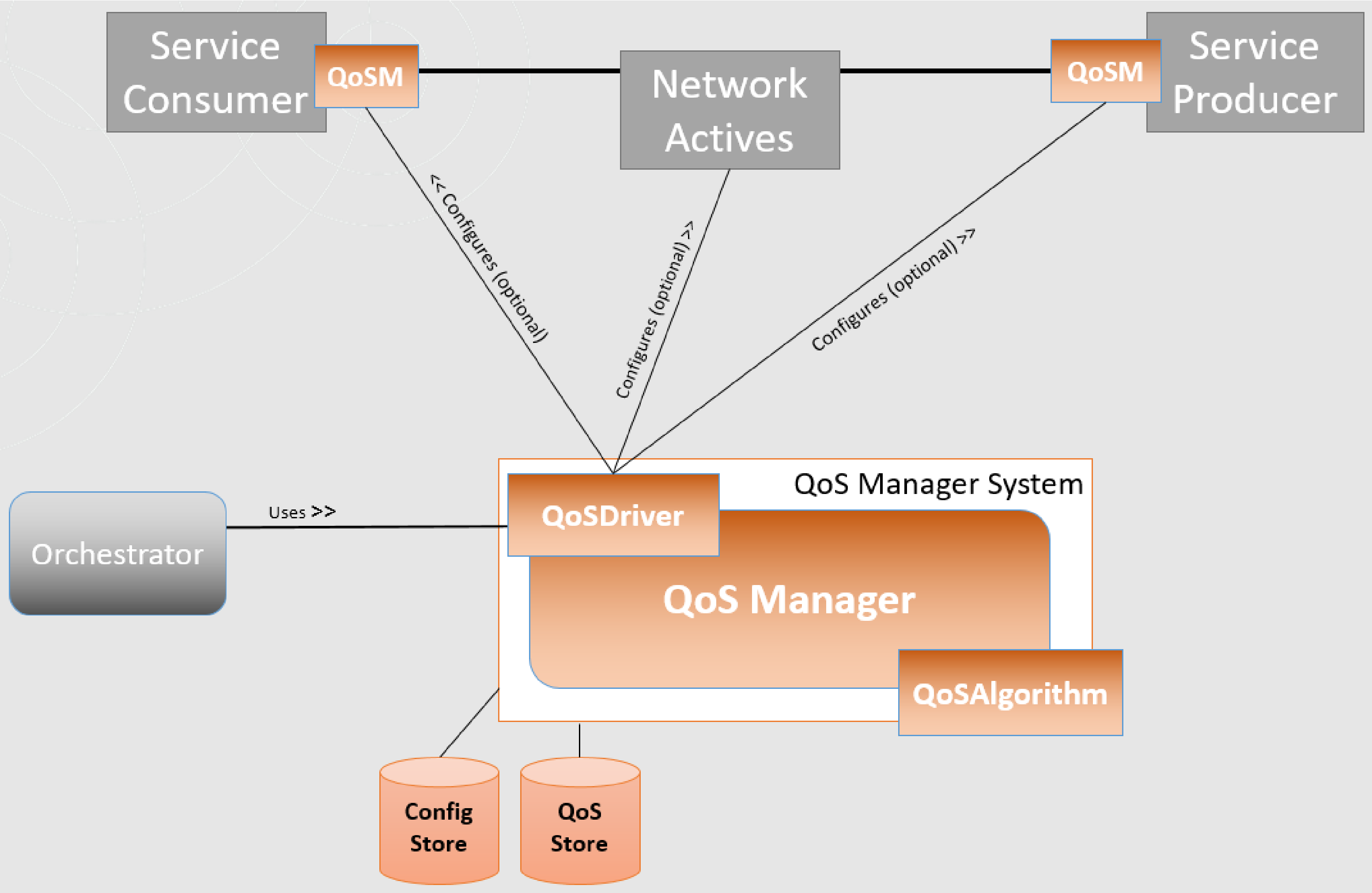
Arrowhead Framework



Vison of the Solution/QoS support in Arrowhead

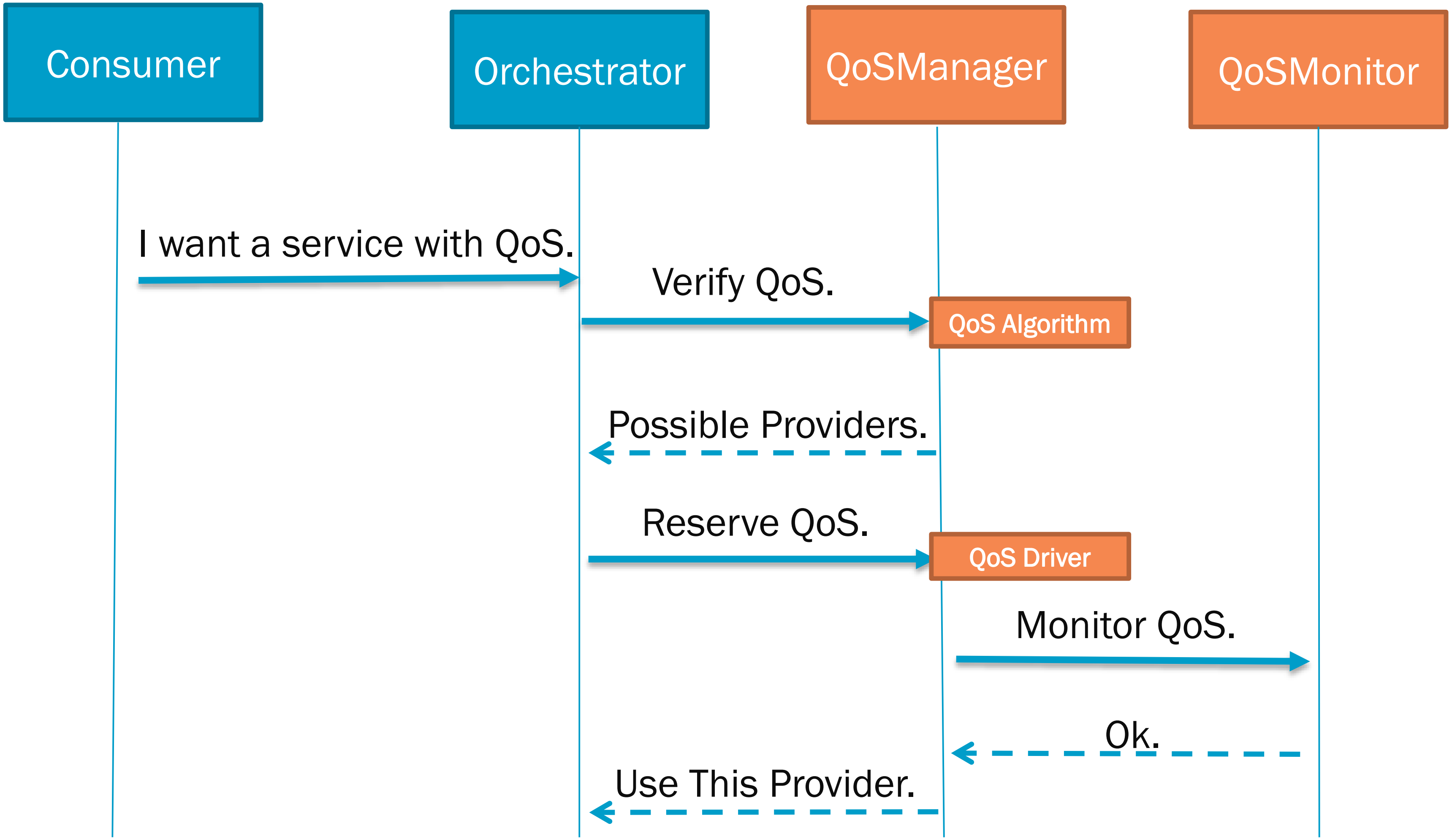
QoSManager

QoSManager



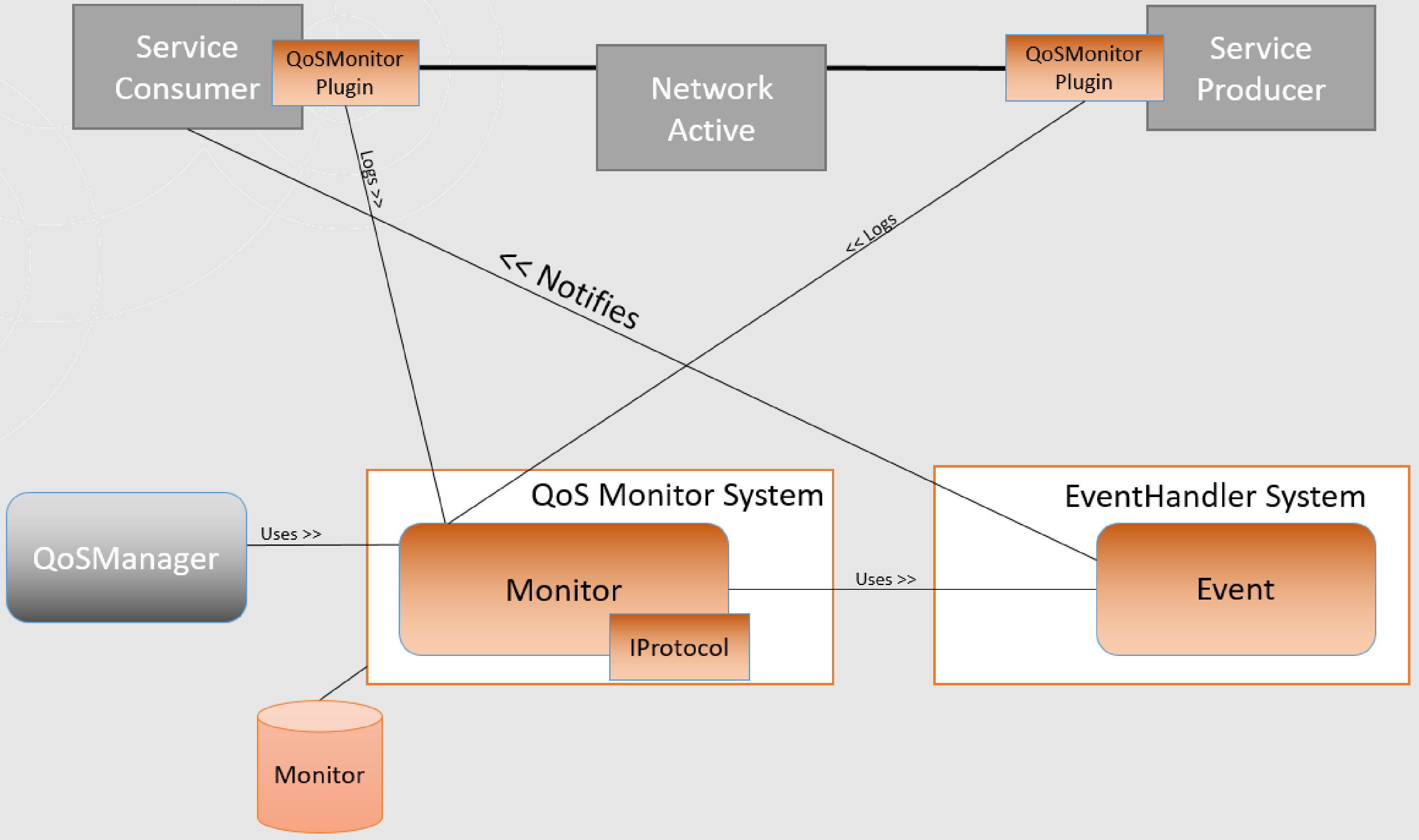
Overview of the QoSManager System

Orchestrator + QoS



QoSMonitor

QoSMonitor



Overview of the QoSMonitor System

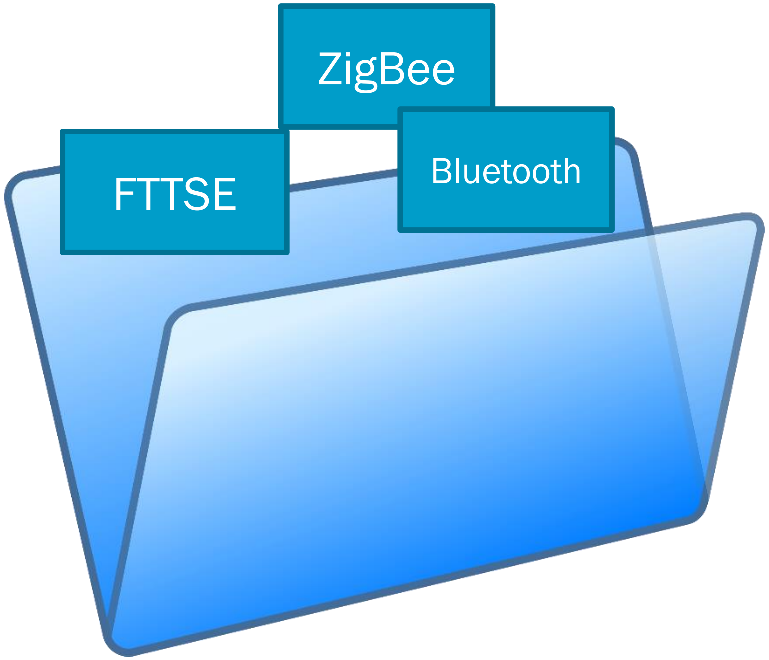
Adaptability to new communication protocols && QoS requirements

Adaptability to new communication protocols

- Usage of **Reflection** pattern to avoid code recompilation each time a new protocol is added.

- Regarding the QoSManager system for each protocol there must be a specific algorithm and driver classes, each one placed at a specific package.

- Regarding the QoSMonitor system for each protocol there must be a specific monitor class.

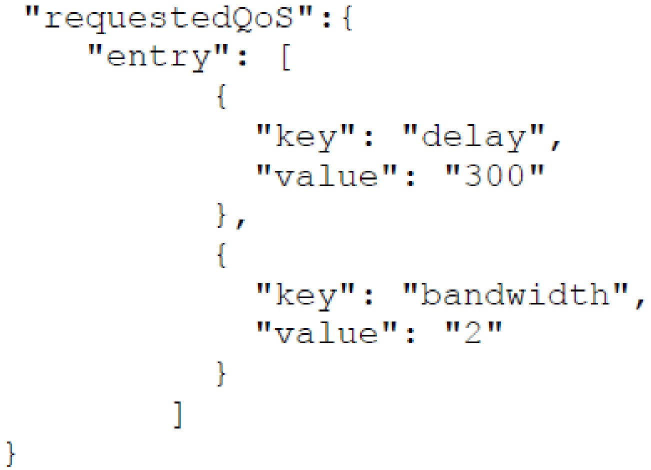


package

Adaptability to new QoS requirements

- The QoS requirements are “opaque” for both QoSManager and QoSMonitor systems. Only the specific protocol classes process the QoS requirements.

- In both systems, the QoS requirements are represented in a Hash Map.



```
"requestedQoS":{
   "entry": [
         {
           "key": "delay",
           "value": "300"
         },
         {
           "key": "bandwidth",
           "value": "2"
         }
       ]
}
```

example of 2 QoS requirements (JSON)
300 ms, 2Bps

Security

Security

- Authorization is a vital part of the Arrowhead Framework. A given system may not be allowed to use a specific service (for example, only the Orchestrator is authorized use the QoSManager).
- Both systems provide a secure HTTP protocol (HTTPS).

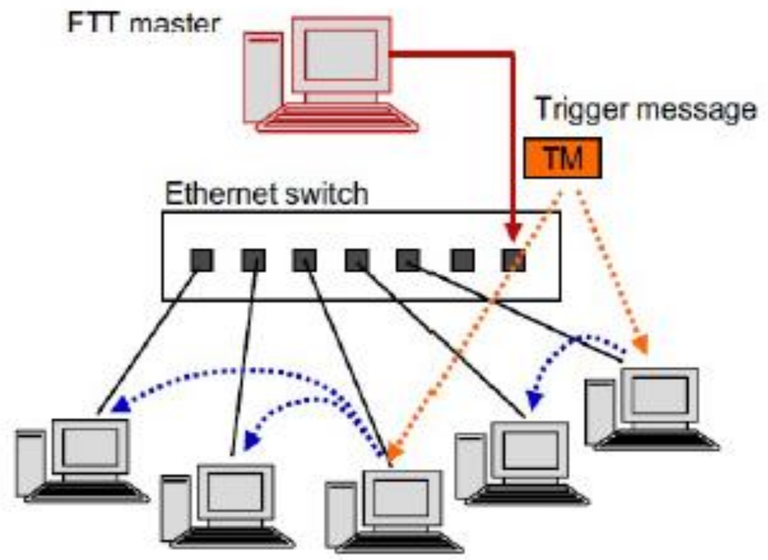
Pilot

Pilot Project

- The pilot project consisted in the integration of the **Arrowhead Framework** with **FTT-SE** communication protocol.
- The team designed and implemented an architecture that could integrate both system-of-systems.
- This model tests if all message streams in the systems will be able to handle their delay and bandwidth requirements.
- **Purpose**: Integrate, Test and Collect data.

Flexible Time Triggered – Switched Ethernet (FTT-SE)

- It is a real-time communication protocol.
- FTT-SE uses switches to reduce non-deterministic behaviour of Ethernet.
- **Master/Slave** Architecture.



- **Goal:** Provide higher bandwidth and conciliating the transition of best-effort traffic and real-time traffic.

Flexible Time Triggered – Switched Ethernet (FTT-SE)

- There are only two types of traffic, **synchronous** and **asynchronous.**

- **Synchronous** traffic is **time-triggered.**
- **Asynchronous** traffic is **event-triggered.**

- Streams have various parameters, of which ID(positive integer), Traffic Type(Sync/ASync), Period(positive integer), Size (bytes) were used.

Flexible Time Triggered – Switched Ethernet (FTT-SE)

- Streams have various parameters:
  - **ID**: Integer identifying a stream.
  - **Traffic Type**: Best-Effort or Synchronous.
  - **Period**: Number of ECs.
  - **Size**: Size of the content to be exchanged.

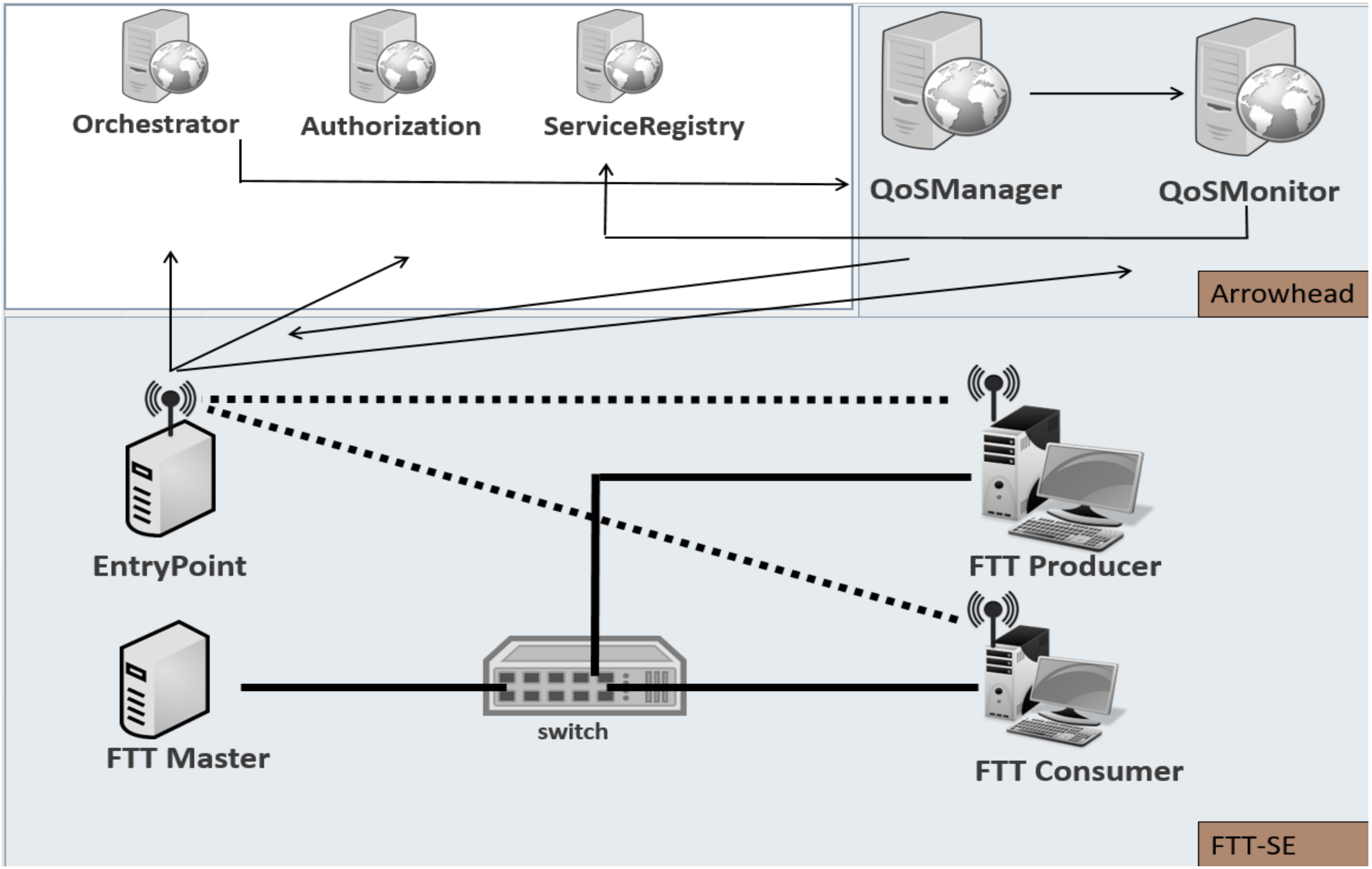
Challenges

- **Address incompatibility**: FTT-SE doesn’t not work with Internet Protocol and uses only MAC addresses to establish communications.
- Since Arrowhead only works with TCP/IP, two possible solutions were proposed:
  - usage of the TunTap technology
  - usage of multiple network interfaces on the nodes.

The EntryPoint

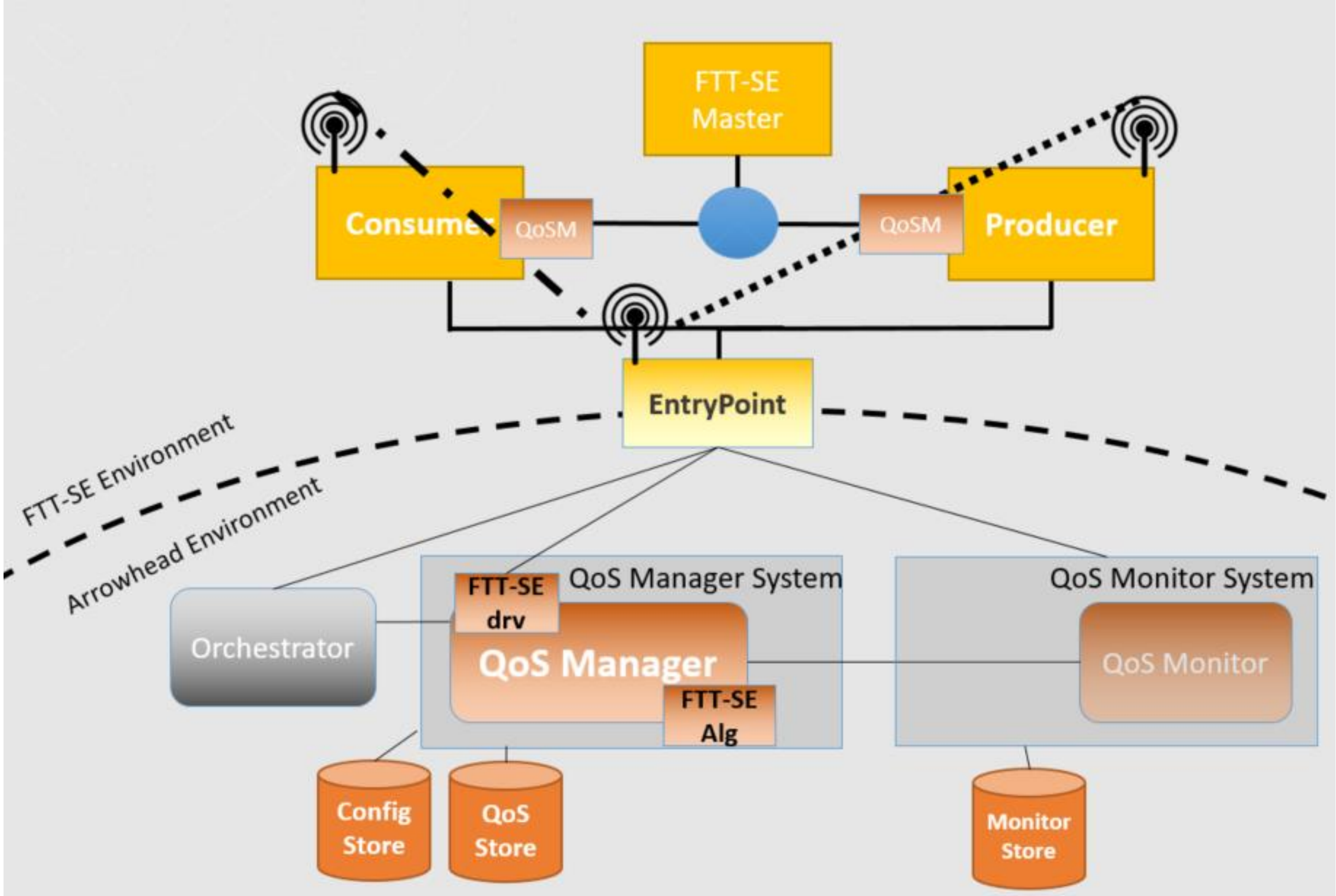
- System developed with the objective of integrating a FTT-SE environment with a REST based approach.
- It acts as a gateway converting HTTP messages payloads into raw socket connections and vice-versa.

Vision of the Solution/Arrowhead with FTT-SE



Overview of the Arrowhead and FTT-SE integration

Vision of the Solution/Arrowhead with FTT-SE



Overview of the Arrowhead and FTT-SE integration

Proof-Of-Concept

Video

Conclusions

- All objectives were accomplished except for the QoSAlgorithm for FTT-SE.
- All Extensibility objectives were accomplished.

Questions?