Software Architecture for Cyber-Physical Systems

Lotfi ben Othmane

A cyber-physical system is a system that augments the capabilities of physical objects through computation and communication.

Recall - Software Architecture

The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them.

L.Bass, P.Clements, R.Kazman, Software Architecture in Practice (2nd edition), Addison-Wesley 2003

CPS Components and Connectors

- Cyber family
  - Cyber components – data store, computation, IO interface
  - Cyber connectors – call-return, publish describe
- Physical family
  - Physical components – sensing, acting
  - Physical connectors: power flow

Example of Problems of Interest for CPS

1. Messages are small
2. Devices are diverse
3. Communication is Intermittent
4. Messages are frequent
5. Processing capabilities are limited
6. Size of data is important
7. Actions are time-critical
8. Number of devices is important

Example of Solutions

1. Messages are small → buffer messages
2. Devices are diverse → create adaptors
3. Communication is Intermittent →buffer messages + support communication errors
4. Messages are frequent → event-based system
5. Processing capabilities are limited → split the computation and use remote services
6. Size of data is important → use data lakes
7. Actions are time-critical → use real-time protocols
8. Number of devices is important → use edge devices

Modifiability Patterns

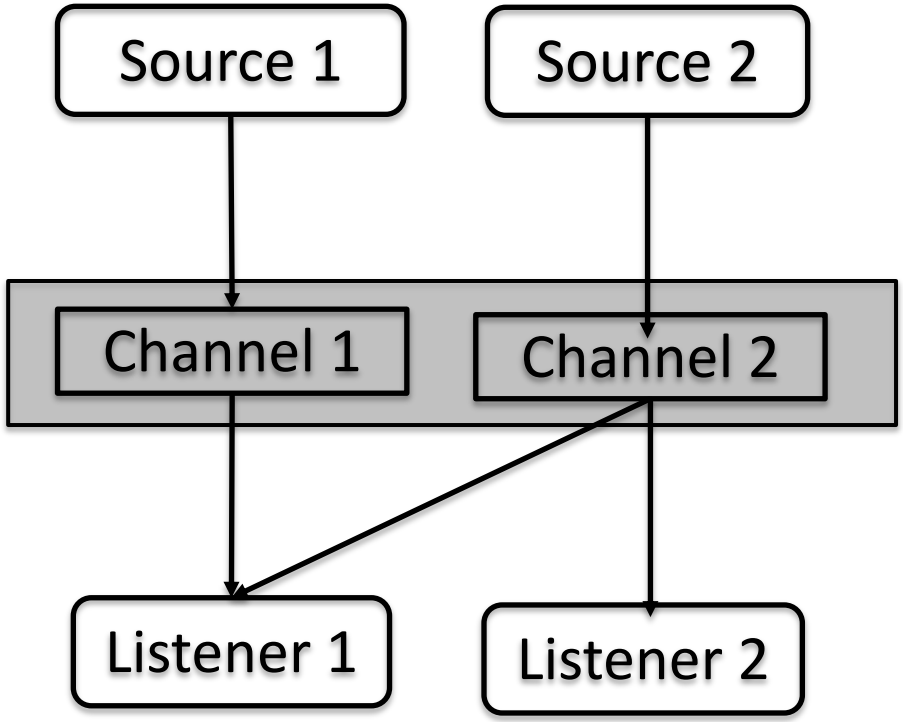
How to improve the modifiability of a system?

We have seen for example:

1. Microservices
2. Plug-ins architecture
3. Increase coherence and reduce coupling

Publish-Subscribe Pattern

Problem: Generation and consumption of events is asynchronous. You need to extend the capabilities that use the events.



Publish-Subscribe Pattern

The pattern could be used for process monitoring, software development environment, and sensor data collection software

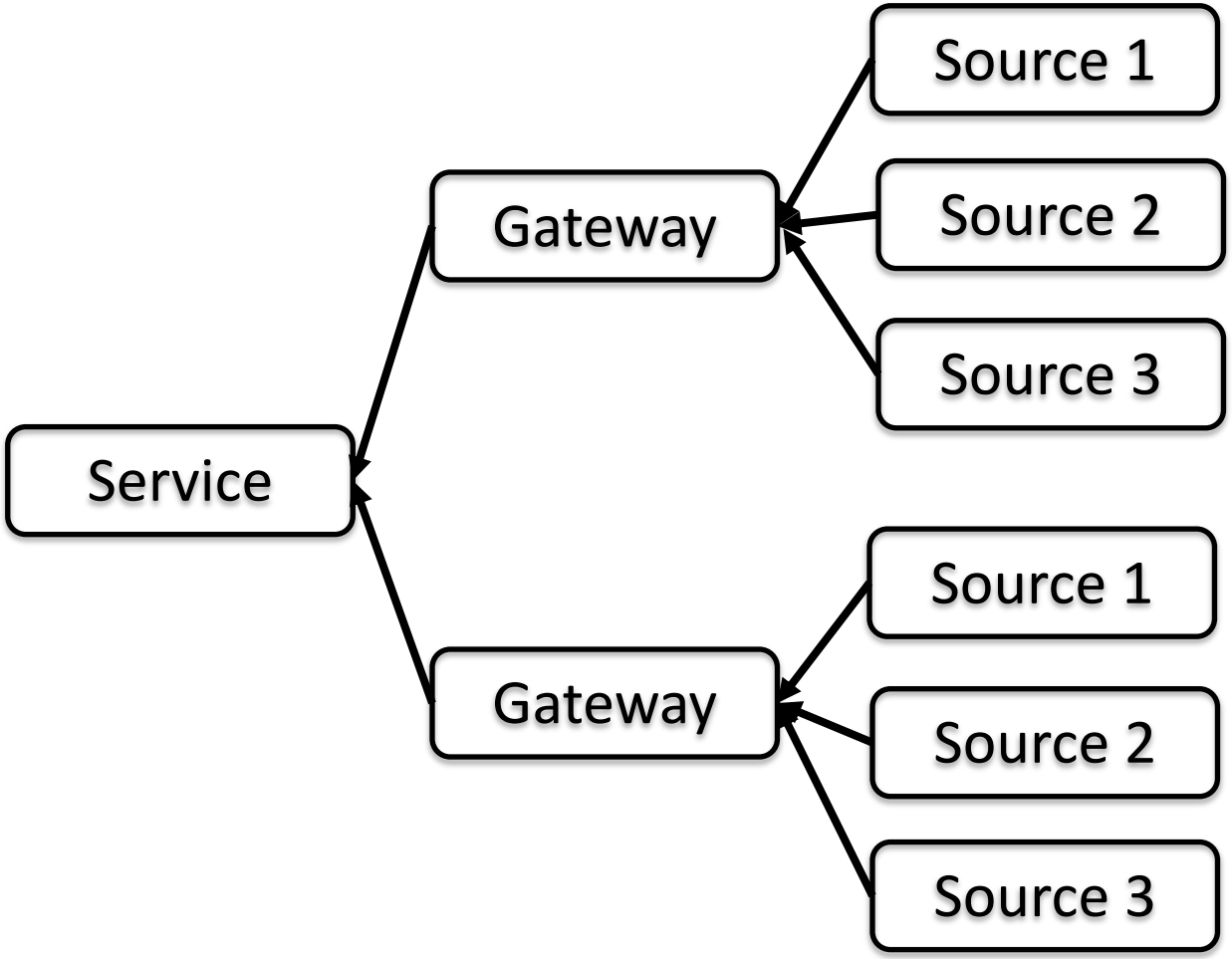
How to improve the performance and reliability of a given system?

Gateway Pattern

Problem: Reduce the communication cost and frequency

Solution: Use gateway to fuse the data

Trade-off: Data are old.

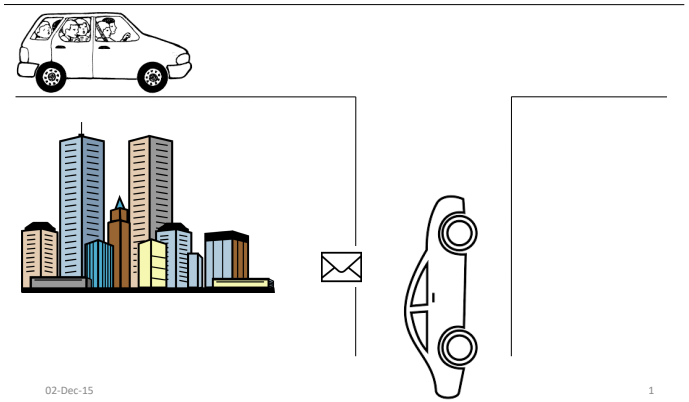


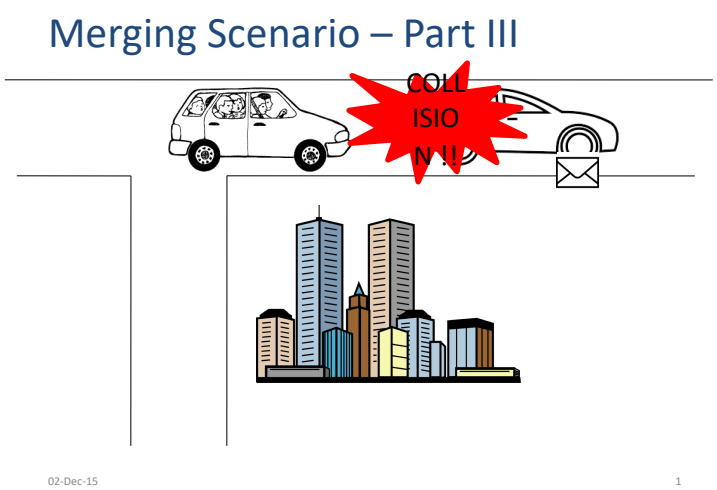
Buffering Pattern

Problem: The network connection is not reliable and/or the communication cost is high.

Solution: Buffer the data and send them when there is connection.

Trade-off: Messages are not on time.



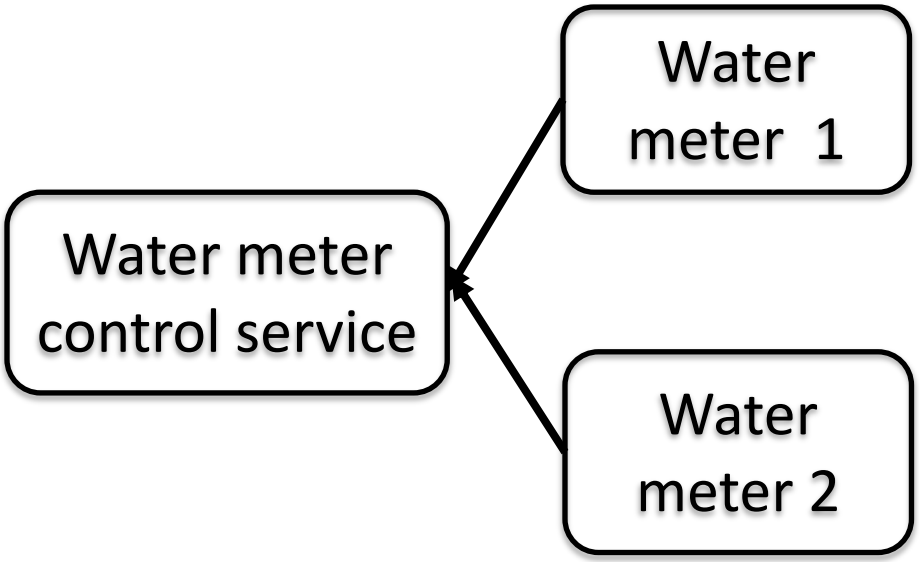


Shadow Copy Pattern

Problem: The communication is not stable and the device gets intermittently offline.

Solution: Components communicate with persistent virtual representation of the device that is synchronized when the device reconnect

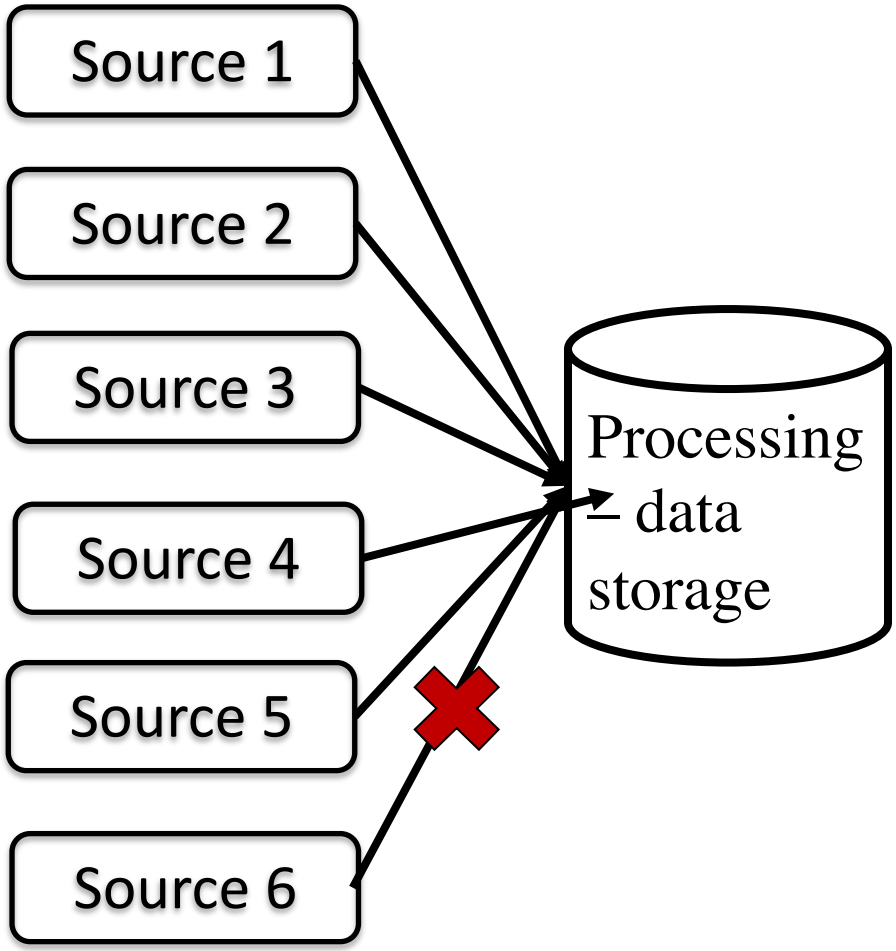
Tradeoff: what trade-off do we have?



The water meter connects every hour to save energy.

Queue Management Pattern

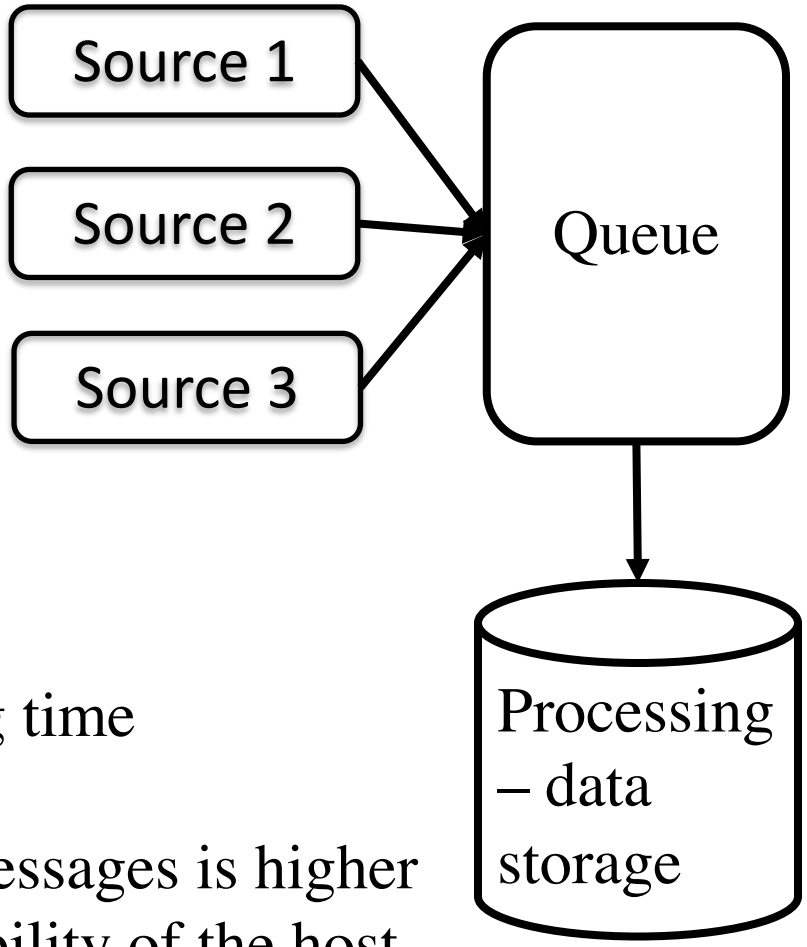
Problem: Sporadic heavy load on a service can cause performance and reliability issues, e.g., loss of data.



Processing time > arrival time

Queue Management Pattern

Solution: Introduce a queue between the task and service

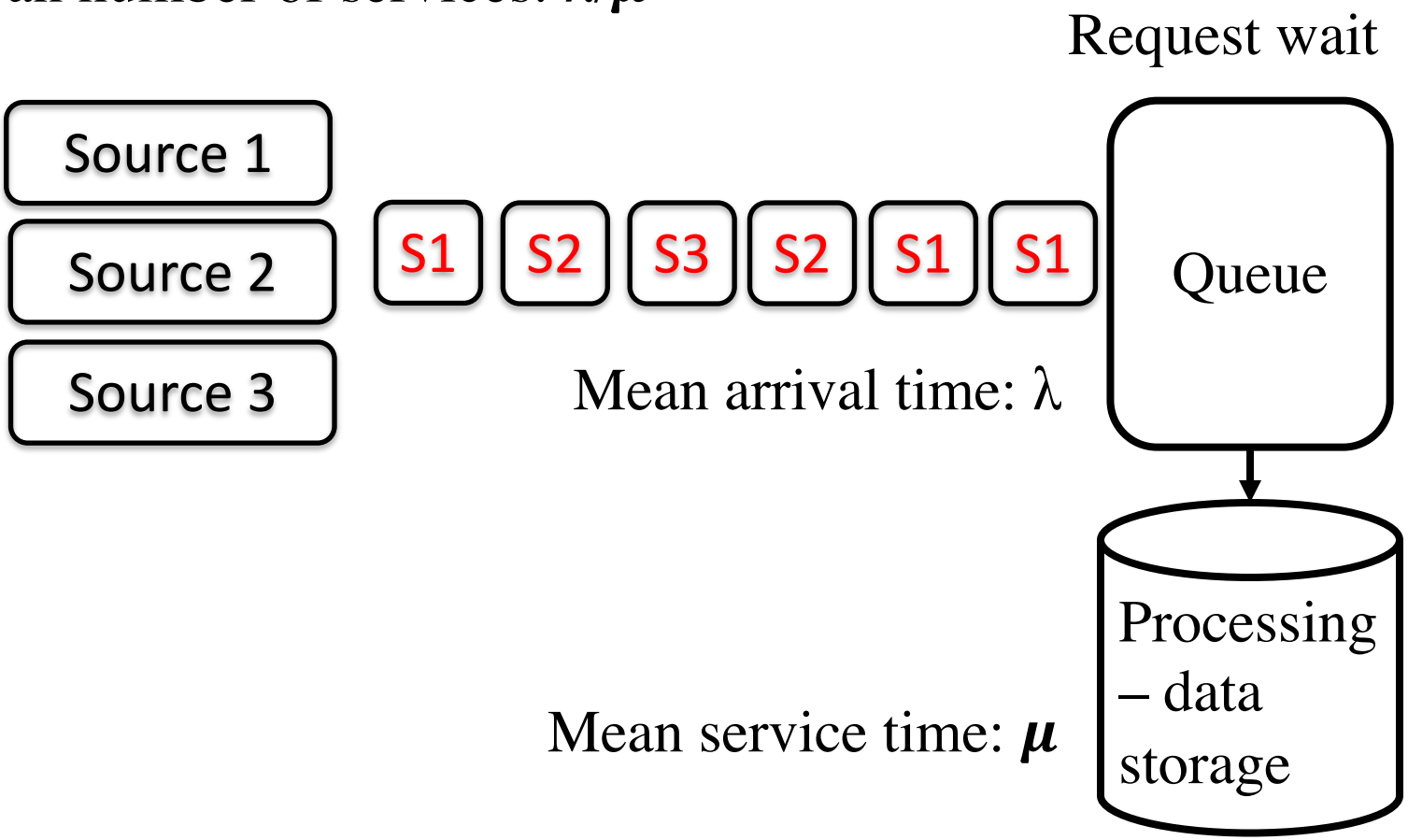


Arrival time < processing time

Frequency or received messages is higher than the processing capability of the host.

Queue Management Pattern

Mean number of services: $\lambda/\boldsymbol{\mu}$



Practice

Scenario: You are working on a system for collecting data from 1000 vehicles. Each vehicle sends 50 messages of 64 bytes per min. Only a subset of the vehicles are in the road at the same time.

Each host can process at maximum 1000 messages in 1 min. A queue processes 10000 message per min.

- How many processing hosts do you need?
- How many queues do you need?

Queue Management Pattern - RabbitMQ

https://www.rabbitmq.com/getstarted.html

RabbitMQ by Pivotal

Features | Get Started | Support | Community | Docs | Blog

RabbitMQ Tutorials

These tutorials cover the basics of creating messaging applications using RabbitMQ. You need to have the RabbitMQ server installed to go through the tutorials, please see the installation guide. Code examples of these tutorials are open source, as is RabbitMQ website.

1 "Hello World!"

The simplest thing that does *something*

- Python
- Java
- Ruby
- PHP
- C#
- JavaScript
- Go
- Elixir

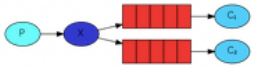
2 Work queues

Distributing tasks among workers (the competing consumers pattern)

- Python
- Java
- Ruby
- PHP
- C#
- JavaScript

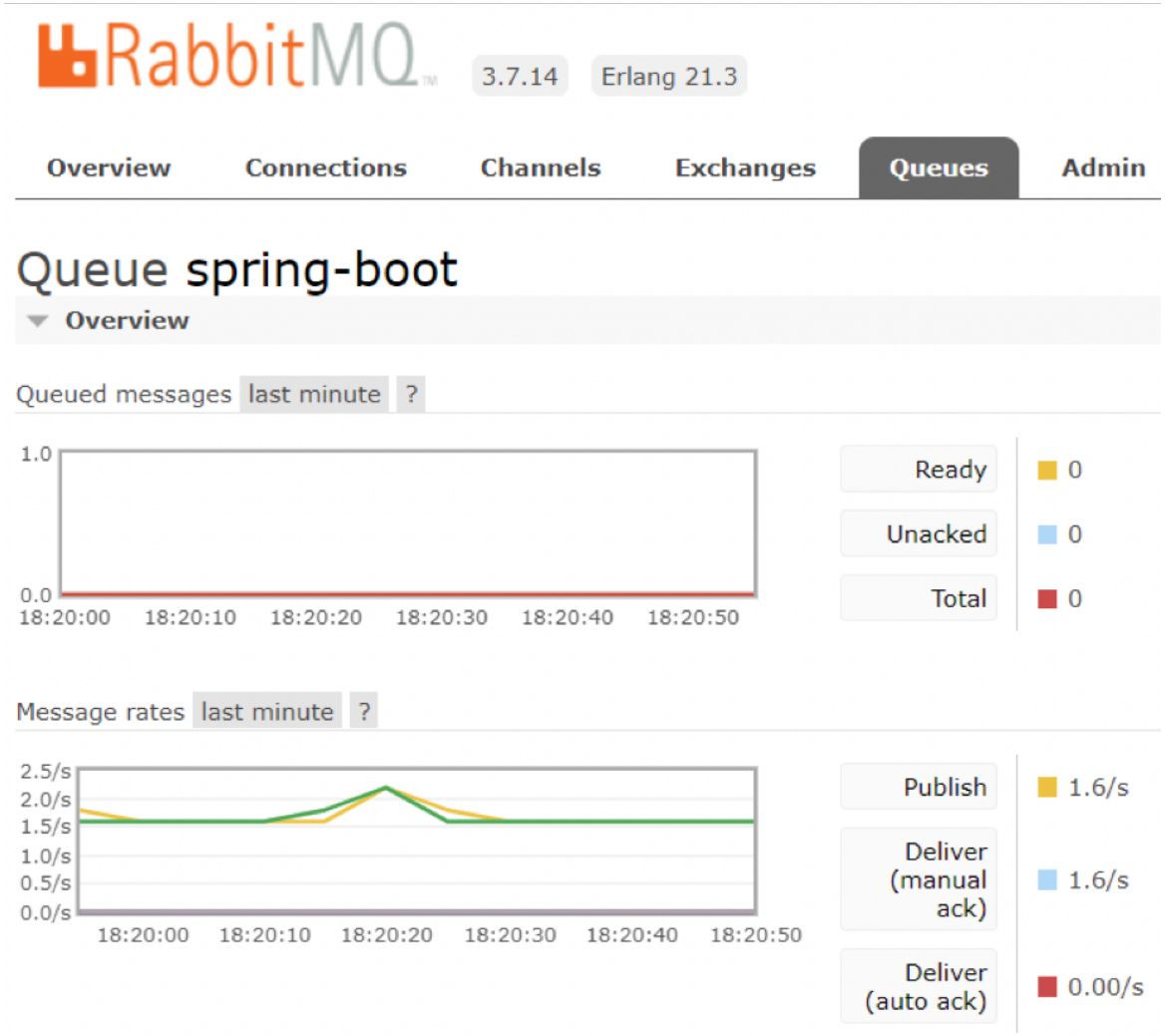
3 Publish/Subscribe

Sending messages to many consumers at once

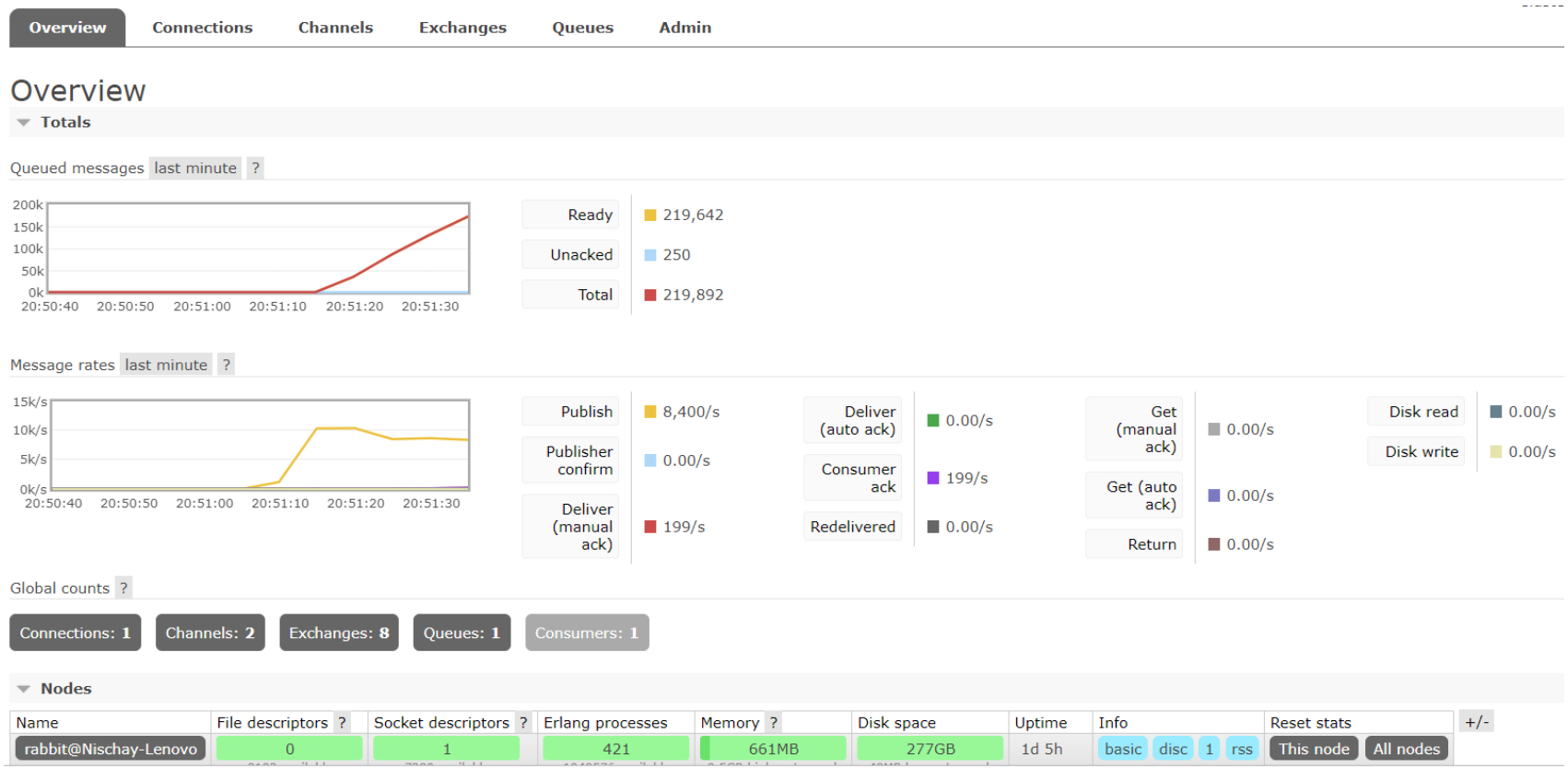


- Python
- Java
- Ruby
- PHP
- C#
- JavaScript
- Go
- Elixir

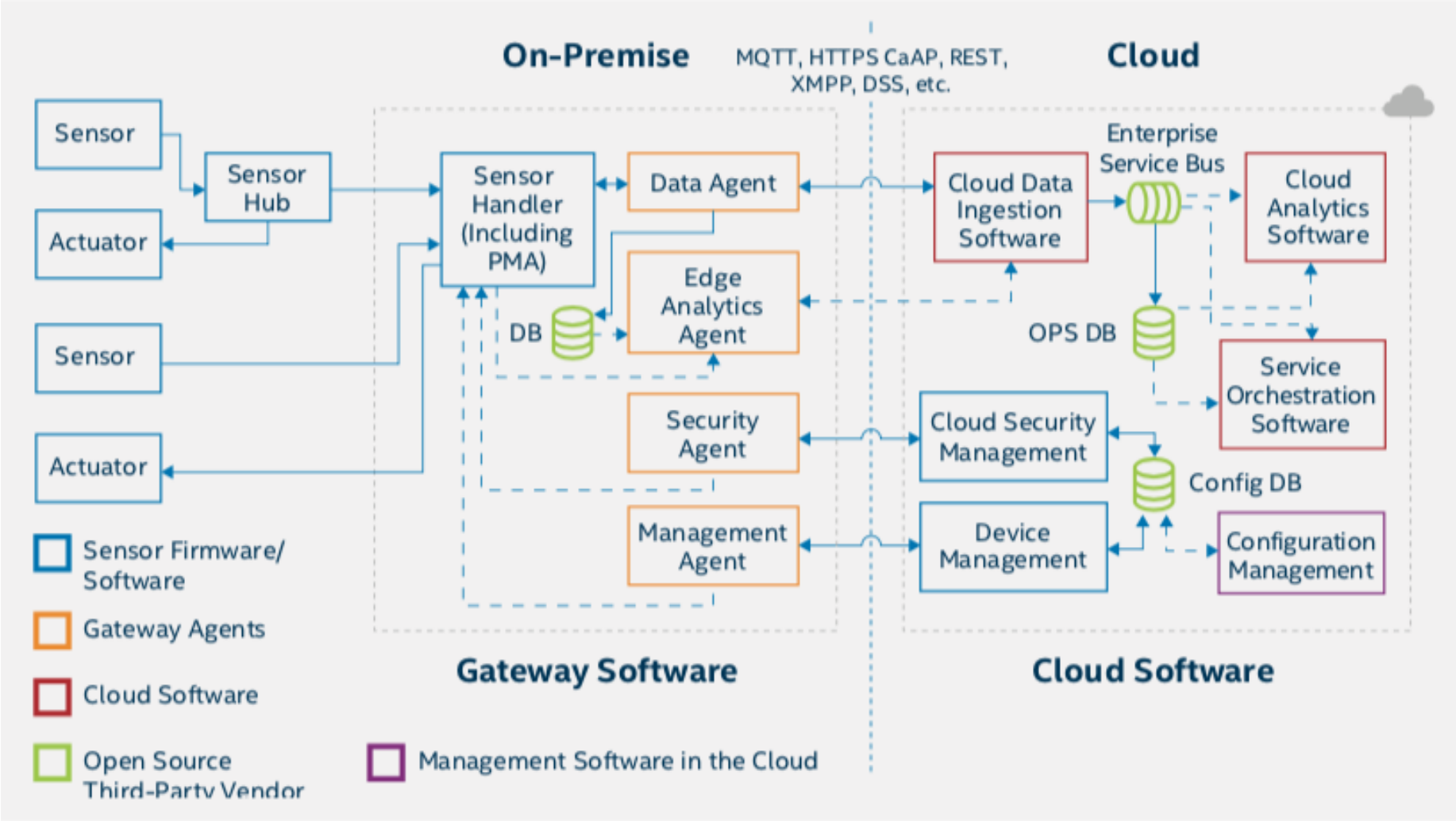
Queue Management Pattern - RabbitMQ



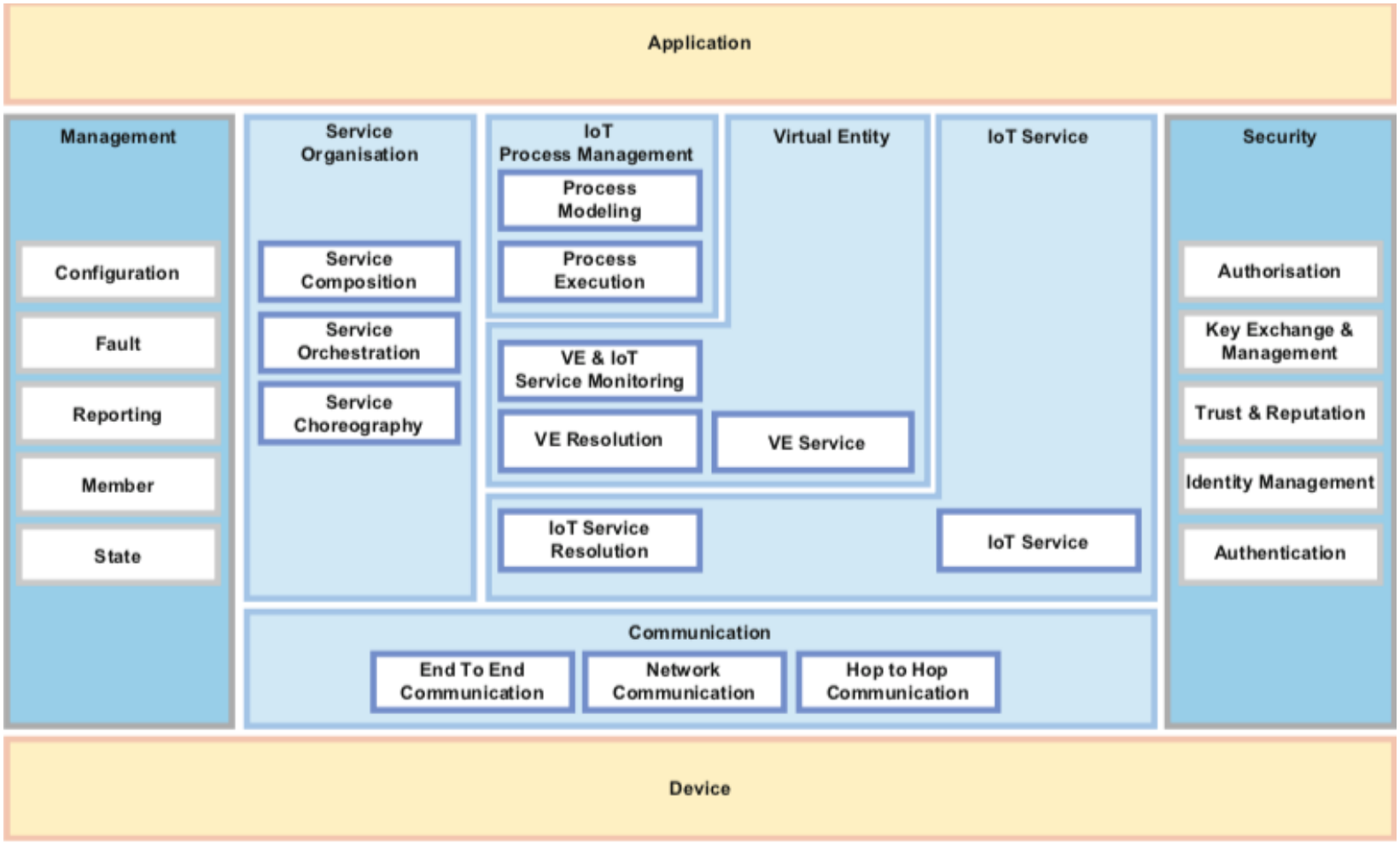
Queue Management Pattern - RabbitMQ



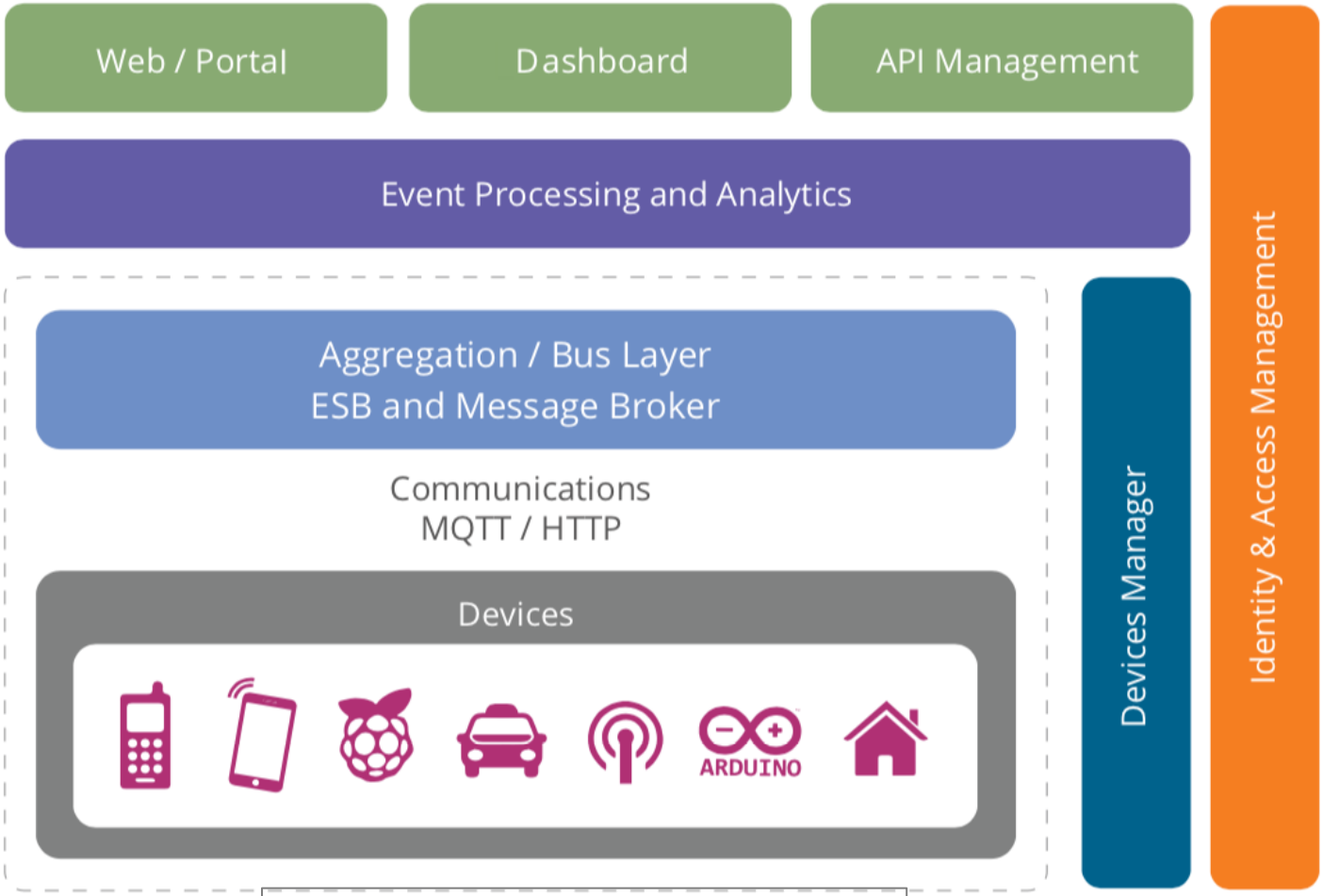
Reference Architecture for IOT-Intel



Reference Architecture for IOT-Research



Reference Architecture for IOT-WS02



Case of WSO2.com

Practice

Select a reference architecture for your project among the provided ones:

2- WSO2 (Figure 2)

3- Group (Figure 8.2)

4- Intel (Figure 4 + table 2)

Justify your choice

Criteria

1. How does it address interoperability?
2. How does it address extensibility?
3. How does it address service orchestration?

SE CPS Main Research Challenges

1. Modeling dimension
2. Faults and conflicts
3. Testing and verification
4. Collaboration
5. Human interface

Self-Check

1. What is a CPS? Give an example.
2. Enumerate three architecture patterns commonly used to address CPS architecture requirements?
3. How can we select a reference architecture? Give an example of reference architecture for CPS.

Thank you

Questions?