Microservice-oriented computing for IoT applications development

Stefano Pio Zingaro

Università di Bologna

IoT Applications

some examples…

- Digital Infrastructure*
    - Smart Buildings management systems — e.g. surveillance, environmental quality, monitoring, etc.
    - Smart Mobility system— e.g. smart parking (IoV), smart routing system for public transports, etc.

IoT Applications

some examples…

- Digital Wellbeing a.k.a. eHealth*
    - Smart diagnoses systems — e.g. including data from bio-instrumentation (or user’s smart-devices) in EHR

IoT Applications

some examples…

- Digital Education*
  - Learning analytics systems — i.e. the use of learner-produced data to discover information for advising people's learning

*** identified as key drivers for the smart city in UE H2020**

IoT Applications

some examples…

smart +

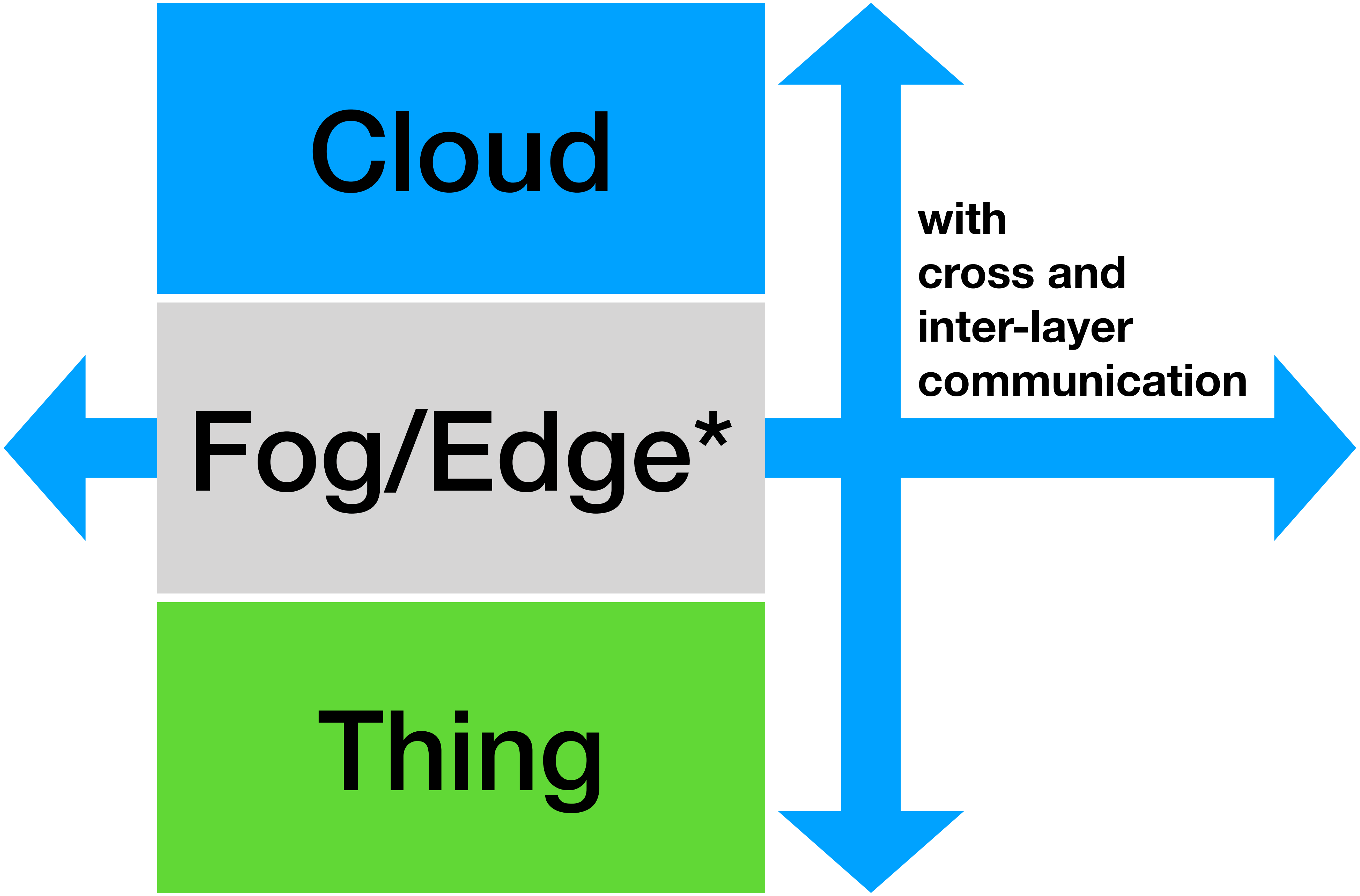
IoT: some context

architectural POV

- distributed systems of heterogeneous platforms
- delocalized topologies
- decomposed application logic (for free)

IoT: some context

architectural POV



Cloud

Fog/Edge*

Thing

with
cross and
inter-layer
communication

*This works focus on the Edge Computing Layer

What you need to dev IoT Apps

developer POV

What you need to dev IoT Apps

- Modularity (updatable)
- Consistency (secure)

↓

Service Oriented

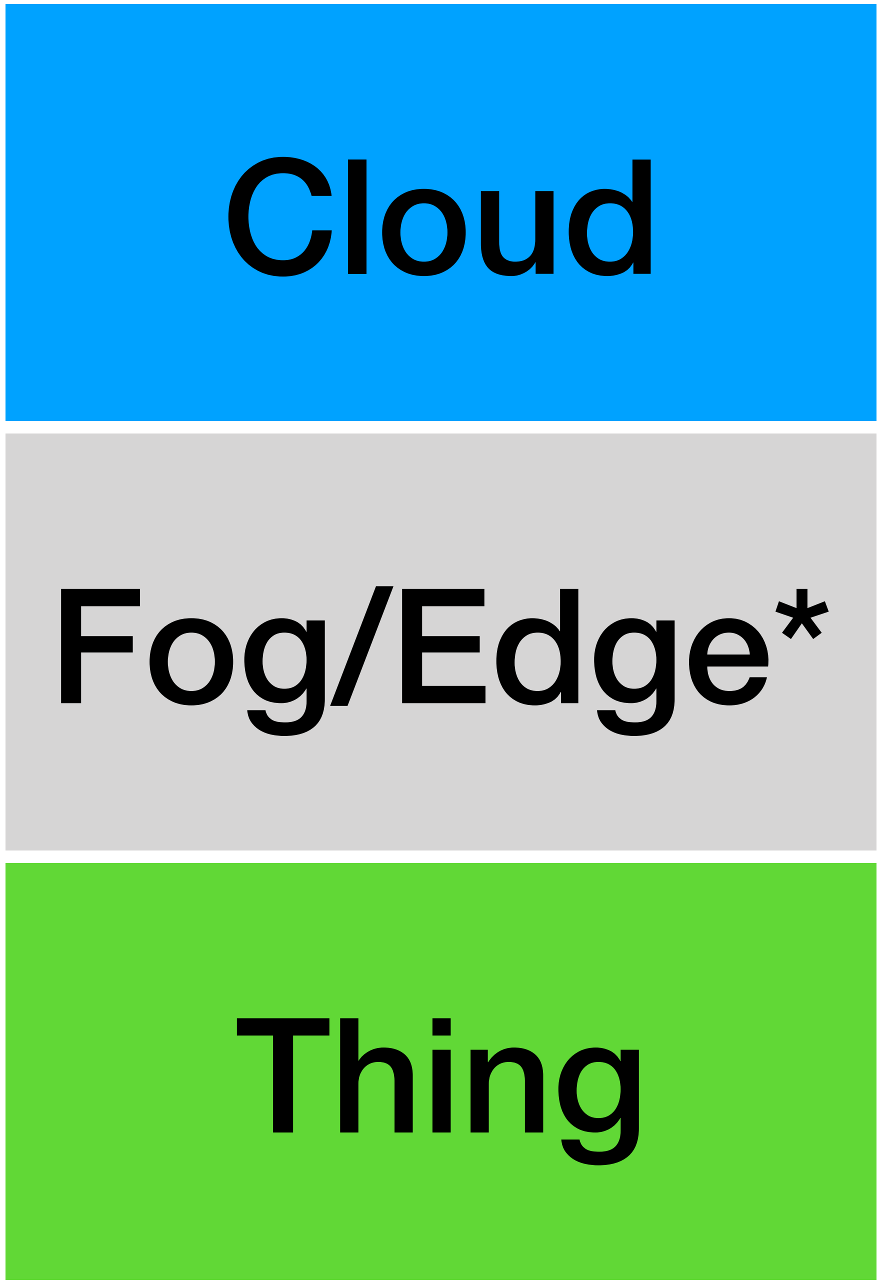
+

- Scalability (e.g. SaaS)
- Agility (QoS compliant)

↓

Microservices

What we need…



Cloud

Fog/Edge*

Thing

*This works focus on the Edge Computing Layer

… what we have

Cloud

**Being web-driven and born from SOA,
reference architecture is cloud-centric**

Things

some definition

- Devices equipped with Sensors/Actuators and (wireless) communication technologies.

Edge Devices

some definition

- A device with computational powers for local data process
  - In compliance with GDPR
  - “no data leaves the building” policy
- Things controllers
- Things collectors

Disclaimer: definitions have not enough consensus to be considered “standards”

Fog nodes

some definition

- Communication middlewares
  - Proxy and API Gateways
  - Brokers

Disclaimer: definitions have not enough consensus to be considered “standards”

Given this context, which solution would help the development of an IoT Application?

some background on me

when people was thinking of me as a
bioinformatician…

I DO COMPUTER SCIENCE
CAN YOU FIX MY LAPTOP?

Computer science deals with programming languages so …

PROGRAMMING LANGUAGE

…we could chose to extend an existing programming language, or to build it from scratch

Jolie

why?

Beacuse it already supports technologies for integration.

Application Protocols	TCP/IP, bluetooth, RMI, unix socket
Mediums	HTTP, HTTPs, SOAP, SOAPs, JSON/RPC, XML/RPC
Data Representation Format	XML, JSON, Binary

Jolie extension

the JIoT project

to support IoT application development addressing the reference architecture we (concretely) integrated into the Jolie (forked) interpreter:

- Message Queuing Telemetry Transport (**MQTT**) — a Publish/Subscribe application protocol

- Constrained Application Protocol (**CoAP**) — a REST-based connection-less lightweight protocol

Remarks

- We used netty.io — an asynchronous event-driven Java library to implement network protocols —> to speed-up the development and increase the performance of the interpreter in modern applications scenarios.
- We extended the medium of Jolie with the support for UDP.
- We provided a end-to-end implementation of the publish/subscribe pattern.

```
main
{
 backHome( );
 toggle@Light( true )
}
```

behaviour

```
inputPort Door {
 Location: "socket://localhost:8001"
 Protocol: http
 OneWay: backHome( undefined )
}
outputPort Light {
 Location: "datagram://localhost:5683"
 Protocol: coap
 OneWay: toggle( bool )
}
```

deployment

```
inputPort Backdoor {
 Location: "socket://localhost:1883"
 Protocol: mqtt {
  .broker = "socket://iot.eclipse.org:1883"
 }
 OneWay: backHome( undefined )
}
```

deployment

discussion

PROs

- easy to program for non-experts
- it becomes easy to emulate lower-level components

CONs

- still need extensions specifically crafted for Jolie

stefanopio.zingaro@unibo.it

References

- SaaS - Sensing as a Service
- JIoT - Jolie for the Internet of Things project