ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Department of
Computer Science
and Engineering

Introduction to W3C Web of Things

Prof. Marco Di Felice,

Luca Sciullo

marco.difelice3@unibo.it,

luca.sciullo@unibo.it

Outline

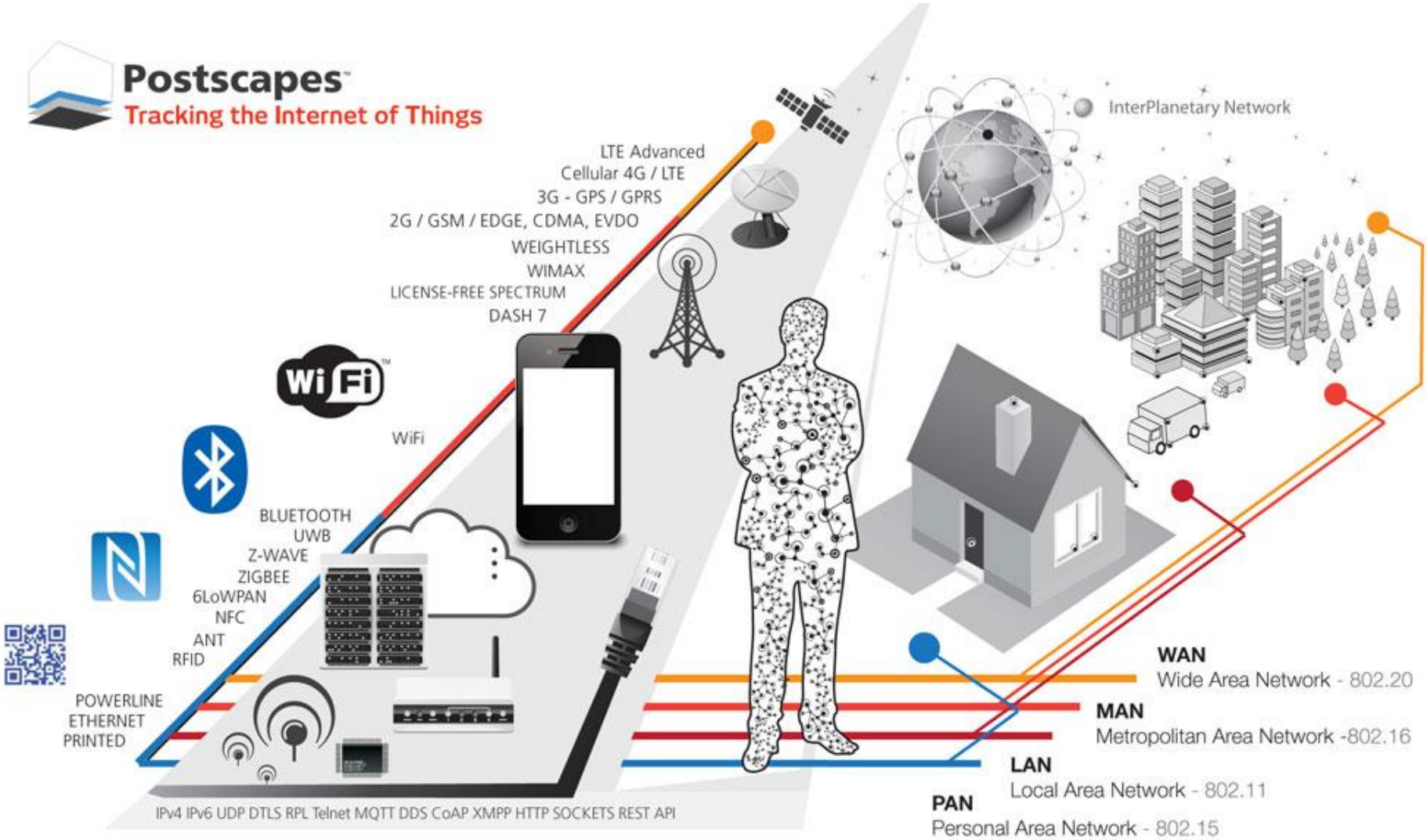
- Introduction
- W3C Web of Things
- WoT Store
- Demo time

Internet of Things

When we talk about the Internet of Things, it's not just putting RFID tags on some dumb thing so we smart people know where that dumb thing is. It's about embedding intelligence so things become smarter and do more than they were proposed to do."

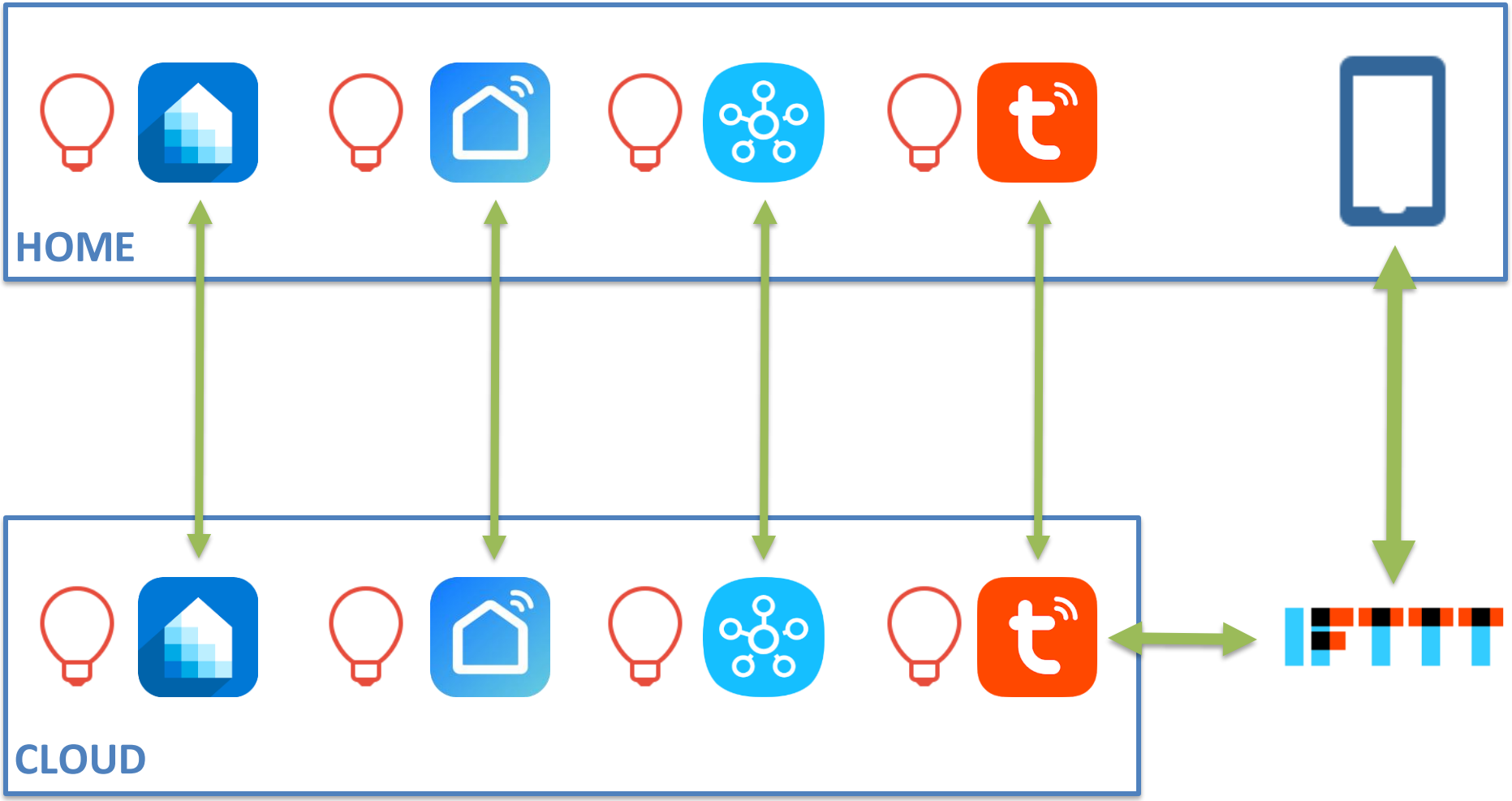
Nicholas Negroponte

Internet of Things: Communication



https://www.postscapes.com/

IoT interoperability: example

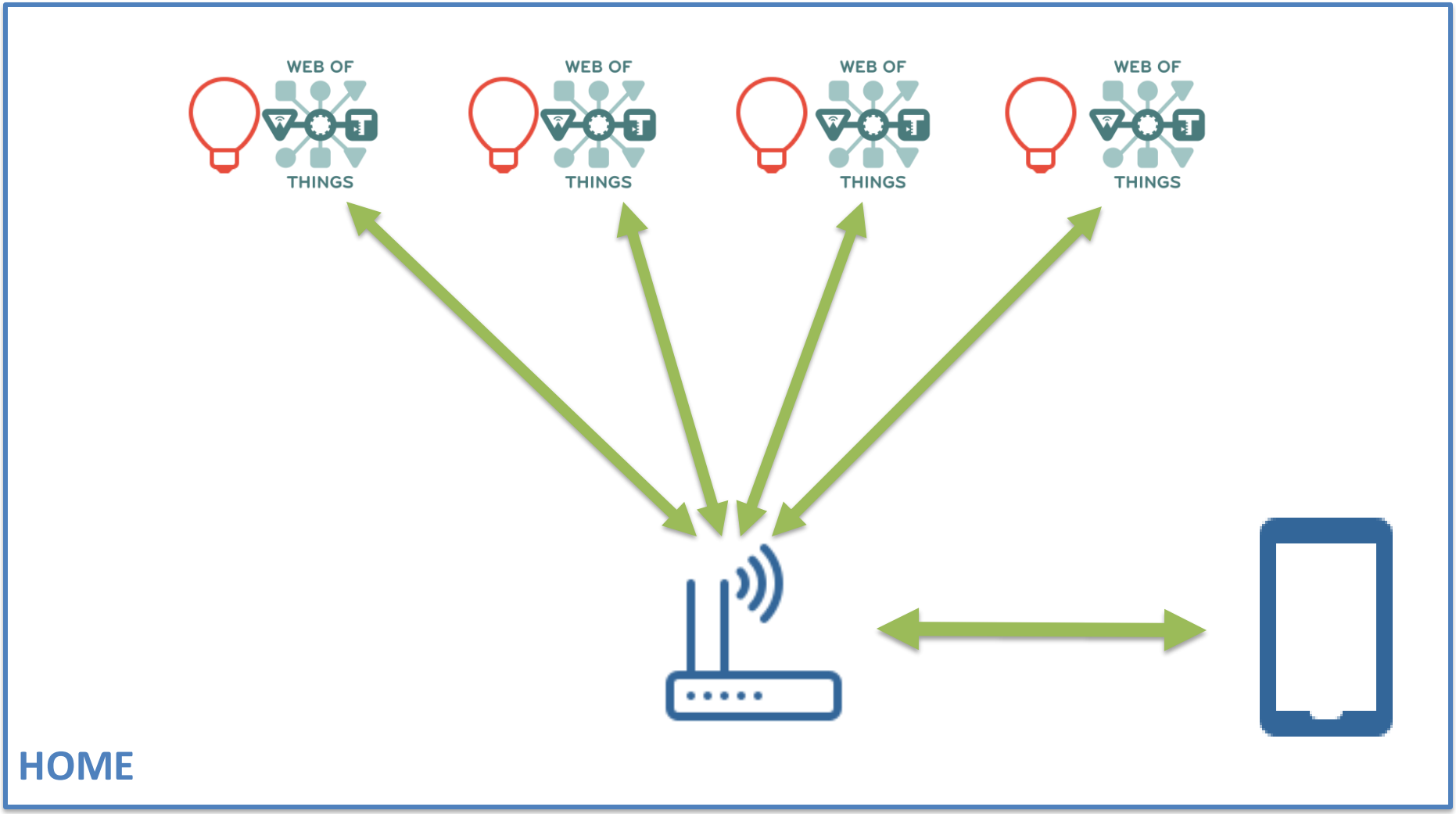


https://icons8.com

From IoT to WoT: open problems

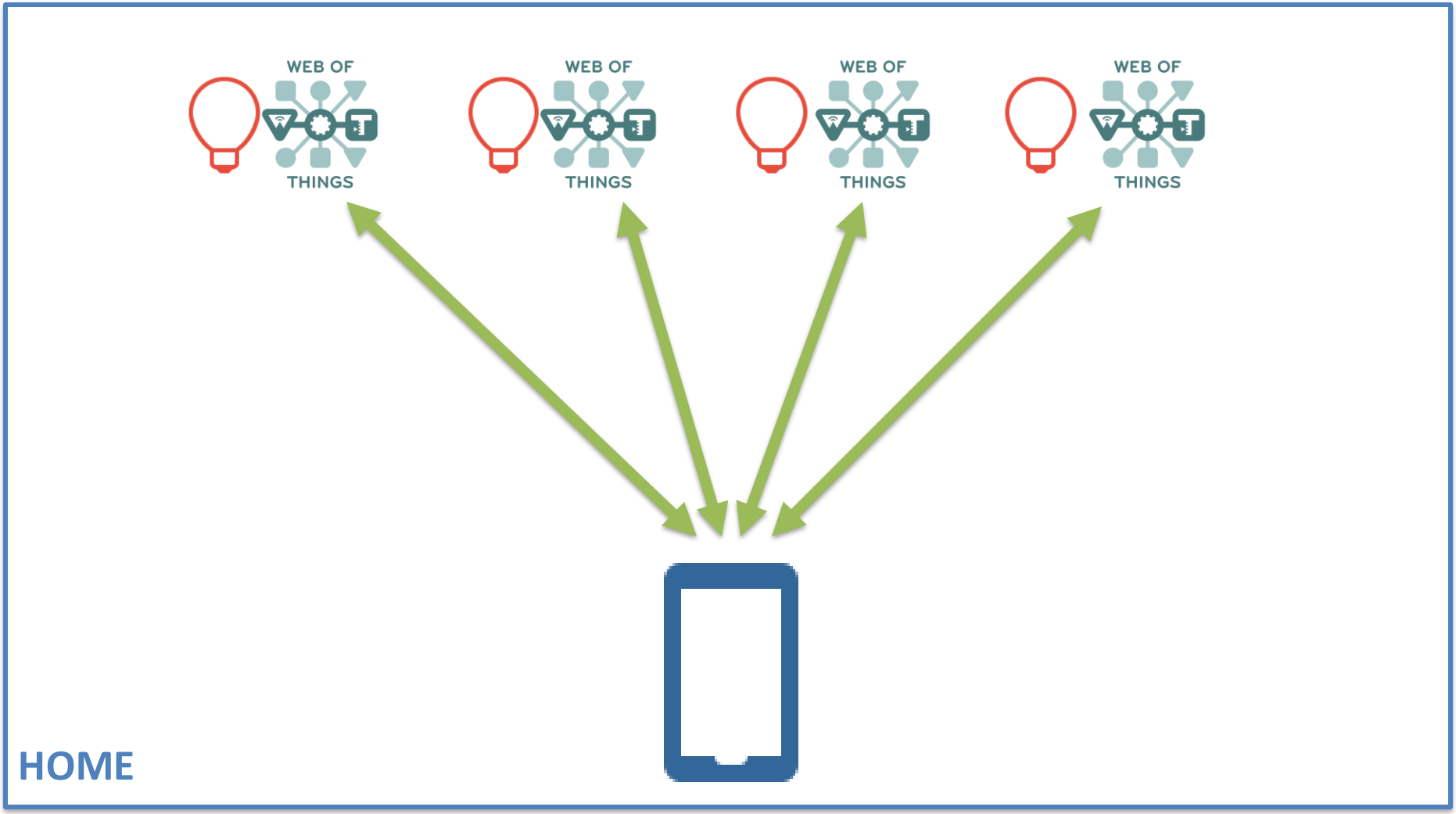
- Data silos
- Privacy
- High Latency
- Dependency from third party
- Thing Discovery

WoT interoperability: example



https://icons8.com

WoT interoperability: example



https://icons8.com

Why do we need Web of Things?

- **Standard way for describing**: things, interfaces, applications, environments, interactions and security requirements
- **Same meaning** when exchanging data
- Easy way for **interpreting data**
- **Lower costs** of development and integration

W3C Web of Things: WoT Working Group

The Web of Things seeks to counter the fragmentation of IoT through standard complementing building blocks (e.g., metadata and APIs) that **enable easy integration** across IoT platforms and application domains.

W3C Thing Definition

An abstraction of a physical or virtual entity whose metadata and interfaces are described by a WoT Thing Description.

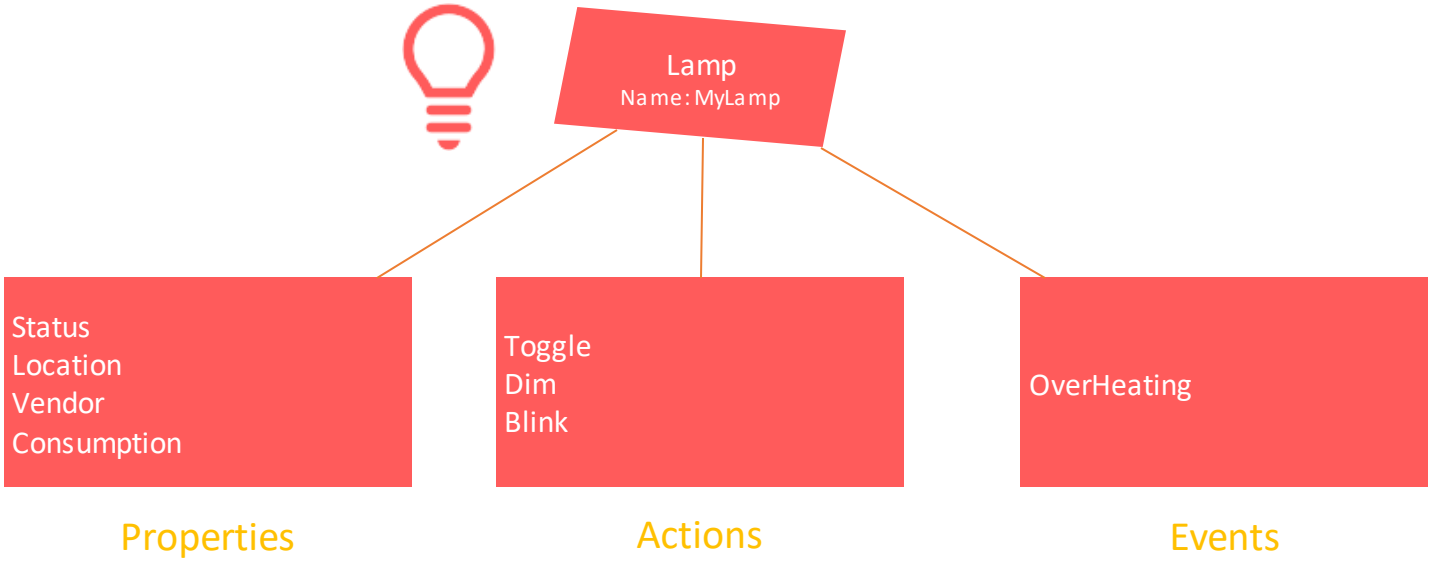
This entity can be:

- an existing device
- a logical component of a device
- a local hardware component
- logical entity (e.g., location)

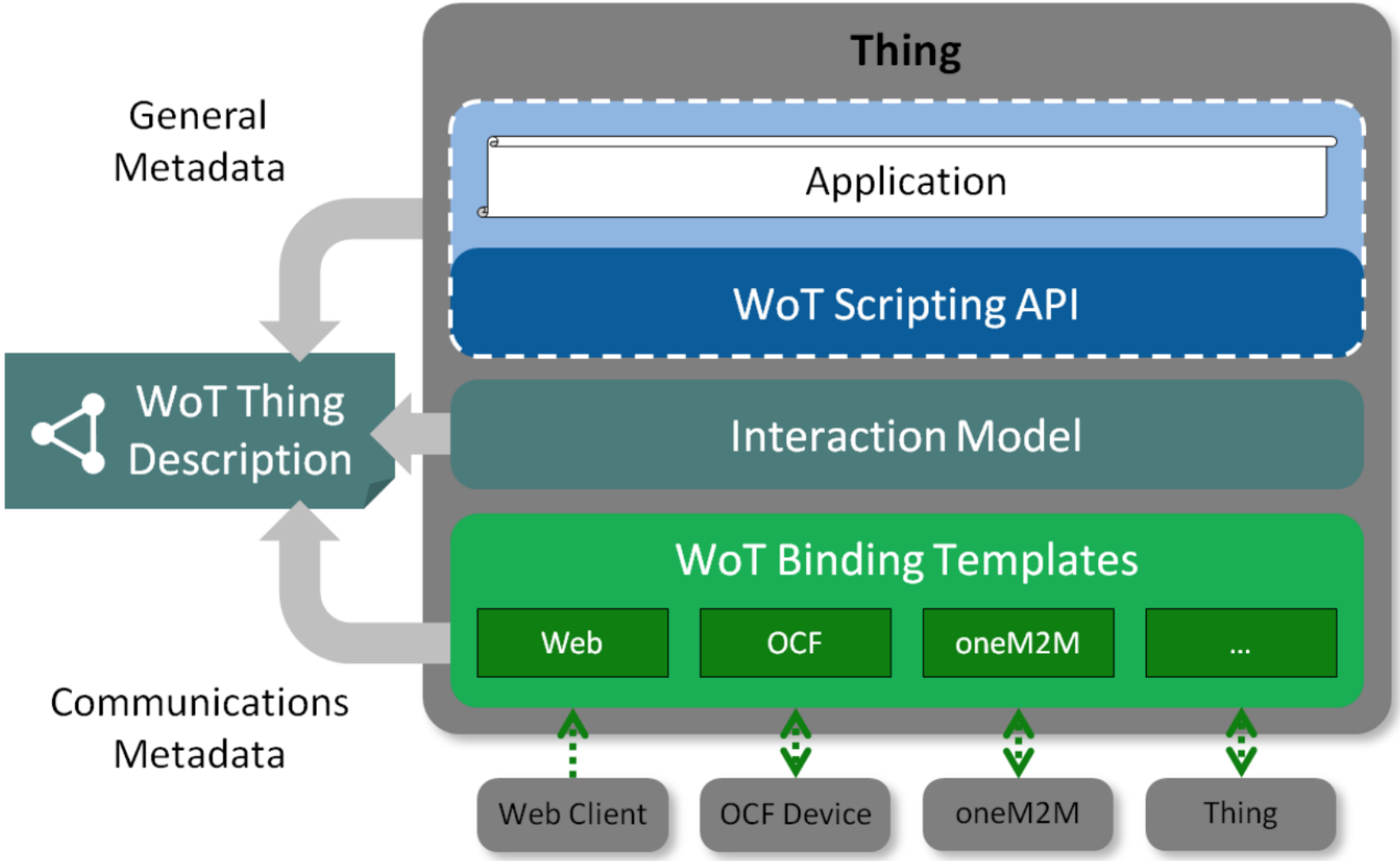
Everything that has a Thing Descriptor is a Thing

Building Blocks: Thing Descriptor

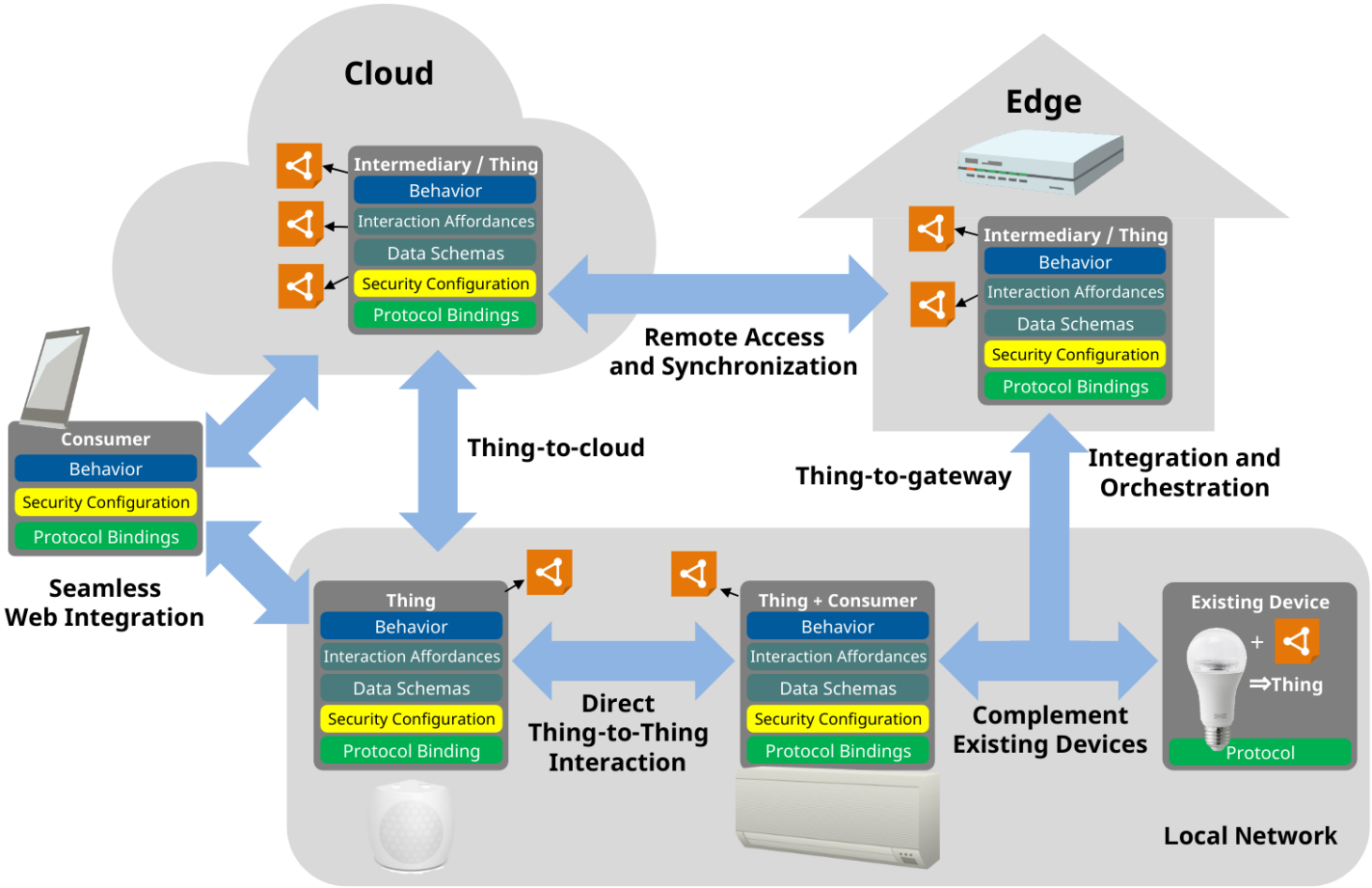
It is the entry point of a thing and it consists of a collection of semantic metadata that describe its **interaction patterns**. It can have **semantic annotations** to make data models machine understandable and an interaction model based on WoT's *Properties, Actions, and Events paradigm*. Its default serialization is **JSON-LD**



Conceptional Architecture of a W3C Thing



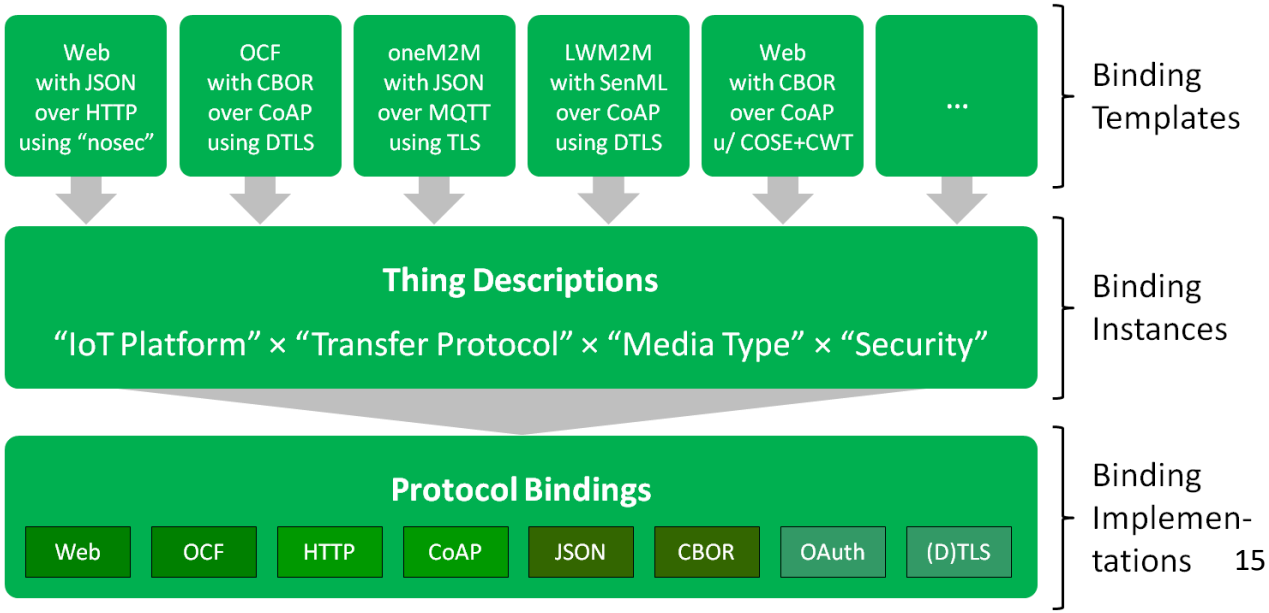
W3C WoT Architecture



Building Blocks: Binding Templates

Problem: enable interactions with a myriad of different IoT Platforms

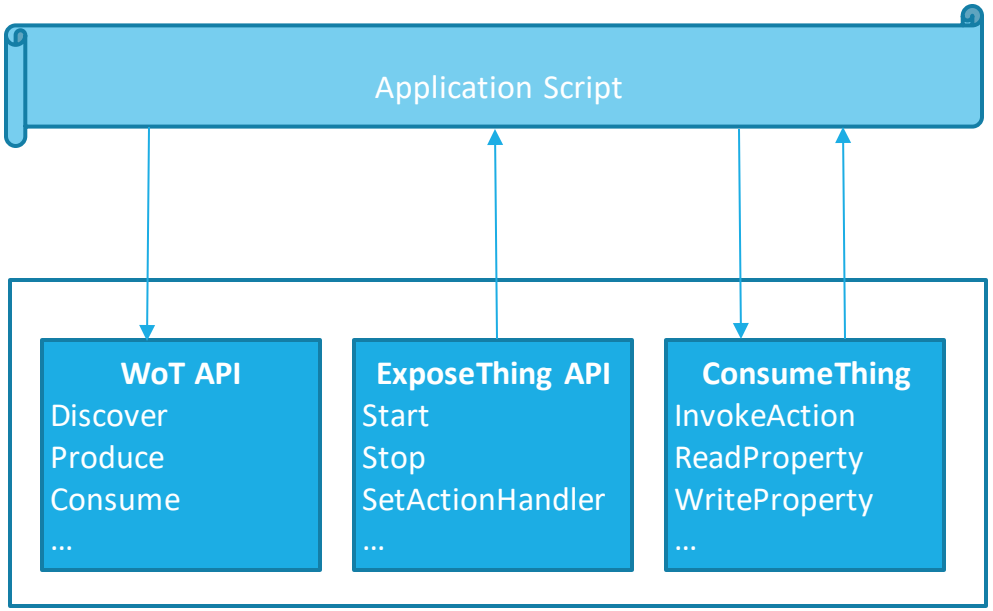
Solution: define multiple vocabularies (**Binding Template**) to describe communication between Things and provide **extension points** in the Thing Descriptor.



Building Blocks: Scripting API

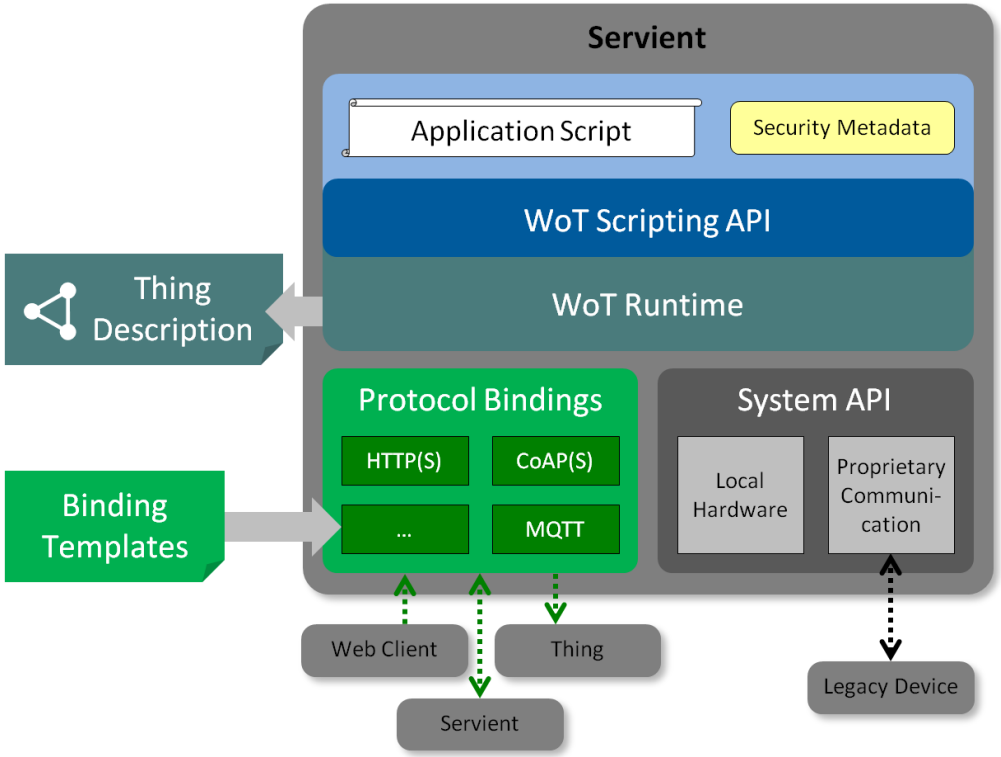
The WoT Scripting API is the runtime system for IoT applications.

- It improves **productivity**
- It reduces the **integration** costs
- It enables **portability** for application modules

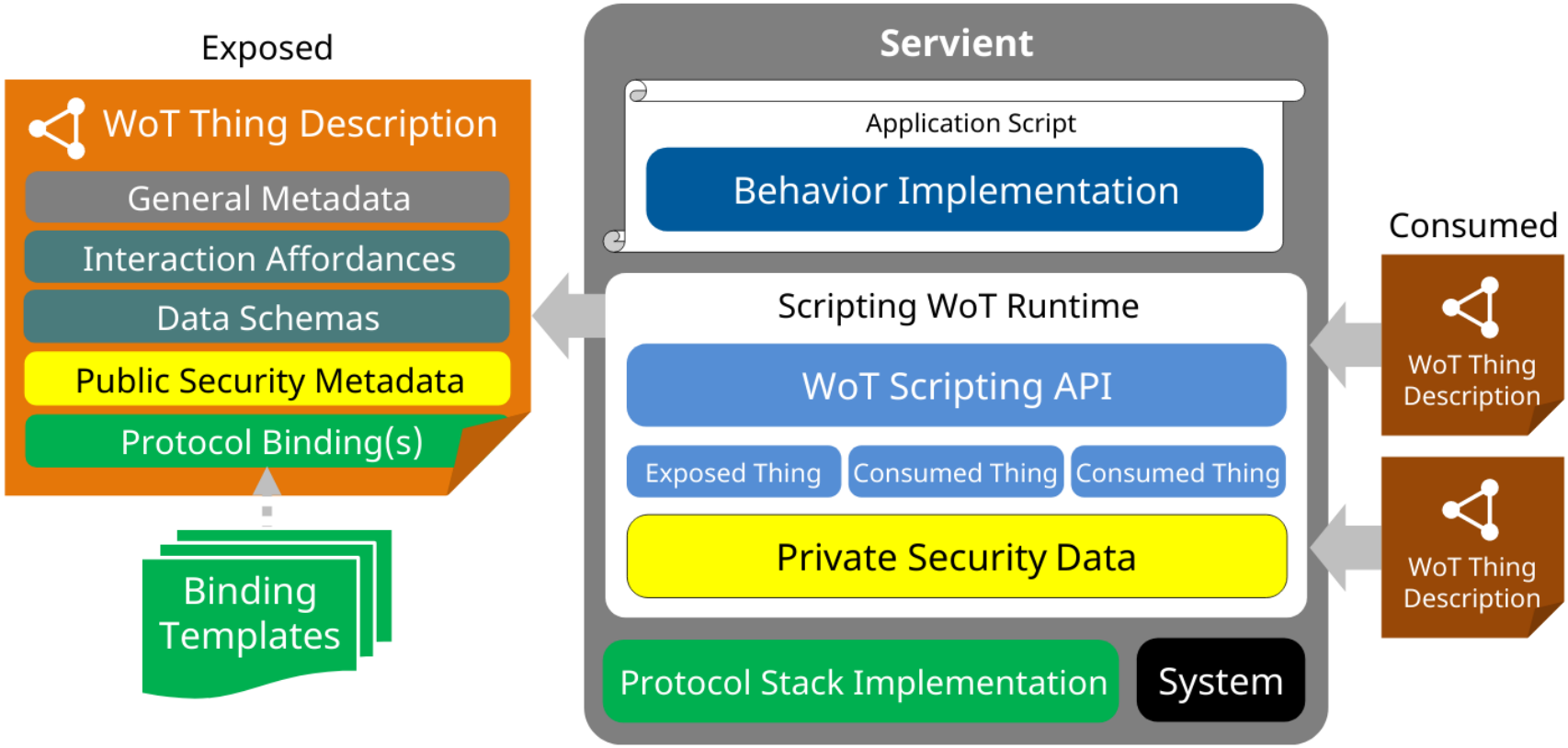


Servient

- **Application**: Thing business logic; implement or using a script or in the firmware
- **WoT Scripting API**: contract between applications and the runtime system (Optional Component)
- **WoT Runtime**: contains Thing and interaction model abstractions. (Optional Component)
- **Protocol Bindings**: implementations of Binding templates, the actual network interface between things
- **System API**: things can access local hardware or system services. (out of scope of WoT standardization)

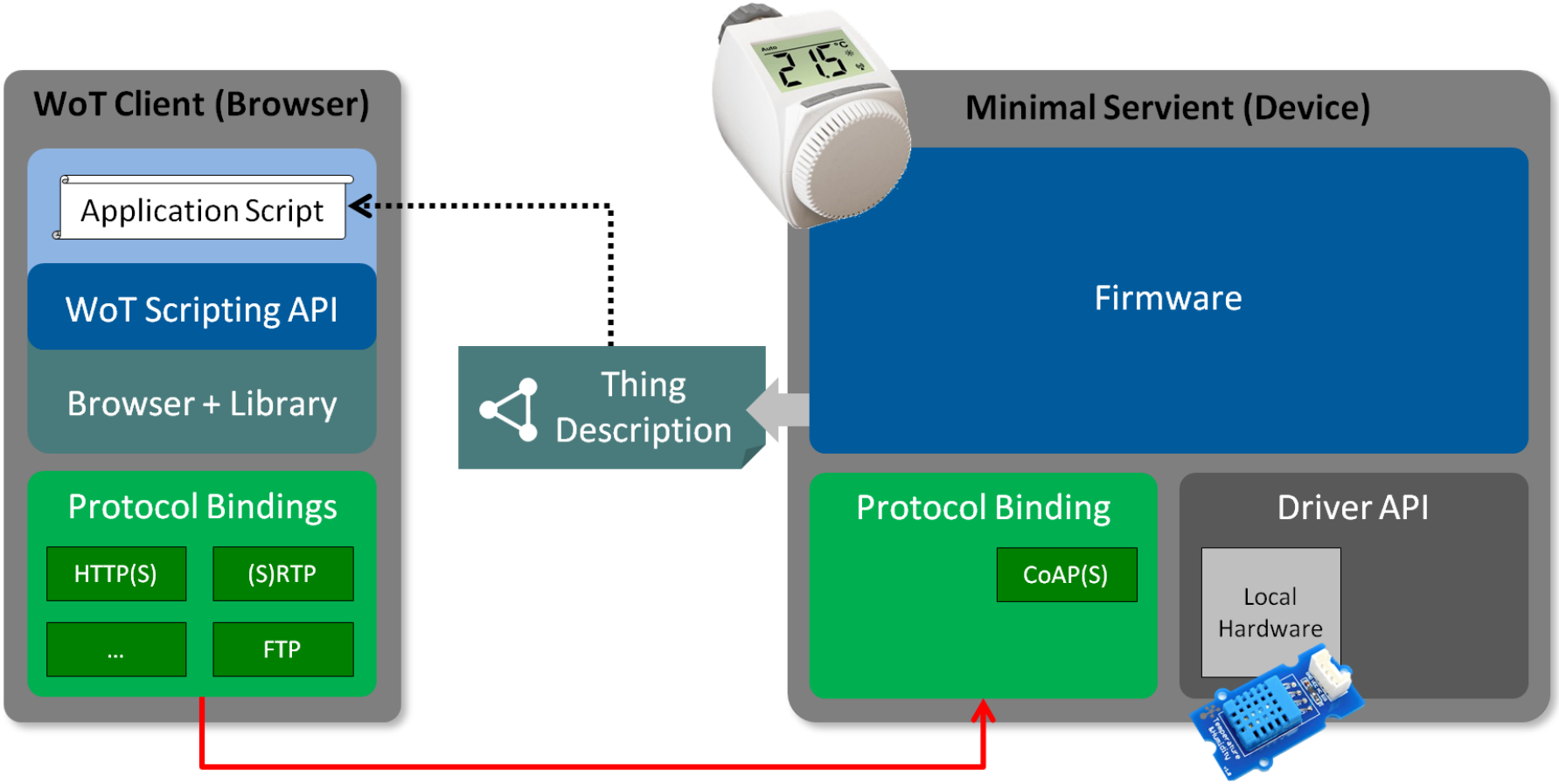


Servient – exposing Web Thing

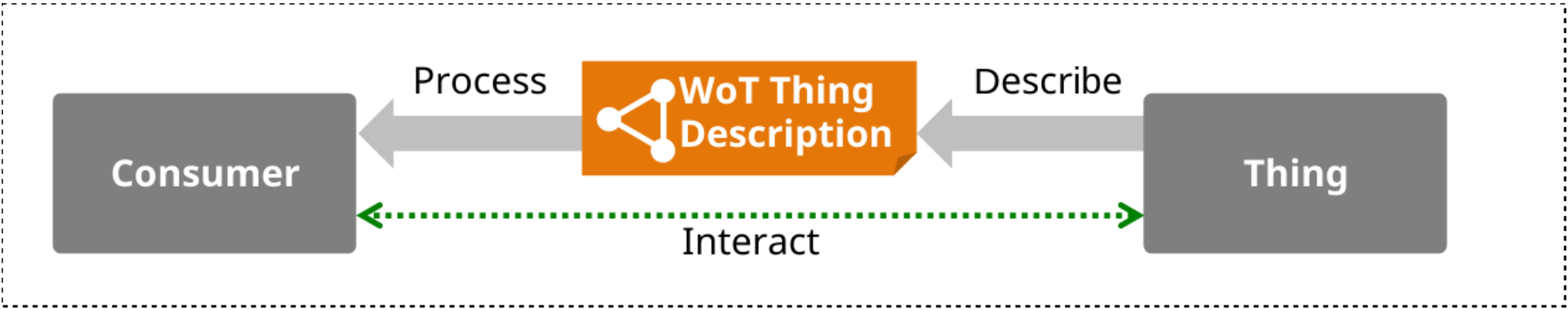


Servient exposes a Thing, making it available to the world

Minimal Servient

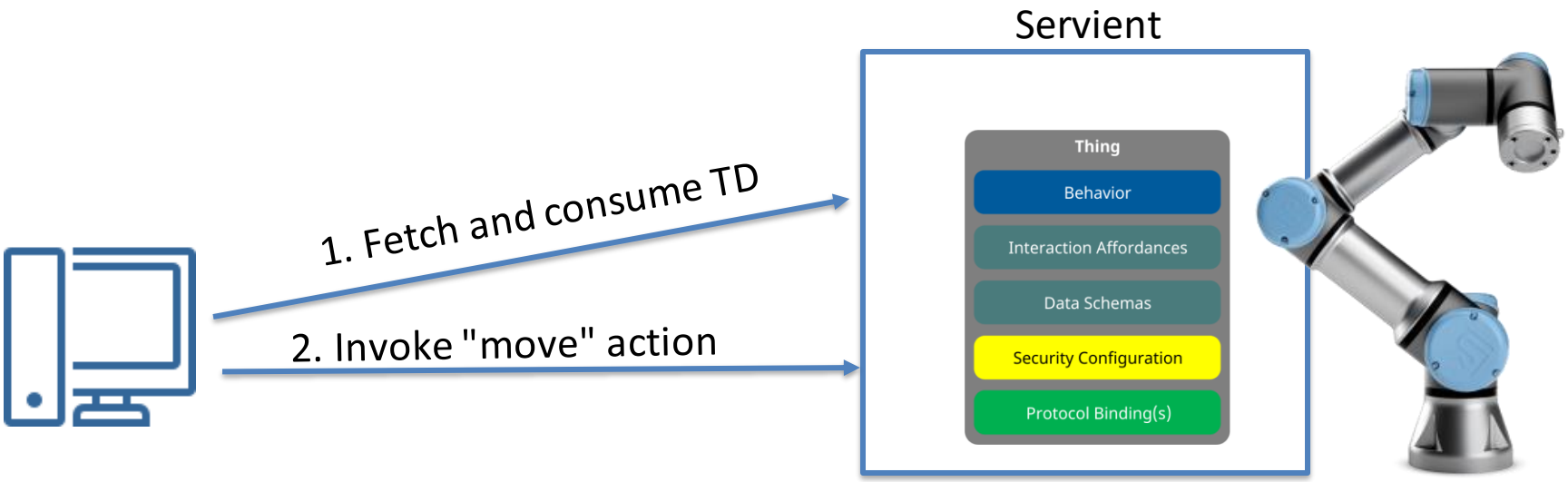


Client – consuming a Thing



Client consumes a Thing and it is now ready to interact with the Thing. It can for instance **read a property, invoke an action or subscribe to an event**.

Minimal Client



Client retrieves the Thing Description in order to understand how to interact with the Thing, then invokes the "move" action to make the arm move

WoT Store

WoT Store is a platform that **enables** the semantic discovery of applications for the W3C WoT, by strictly adhering to the W3C architecture.

Main features:

- Semantic discovery of Thing Applications (TAs)
- Semantic discovery of Mashup Applications (MAs)
- Automatic deploy of TA software on Things

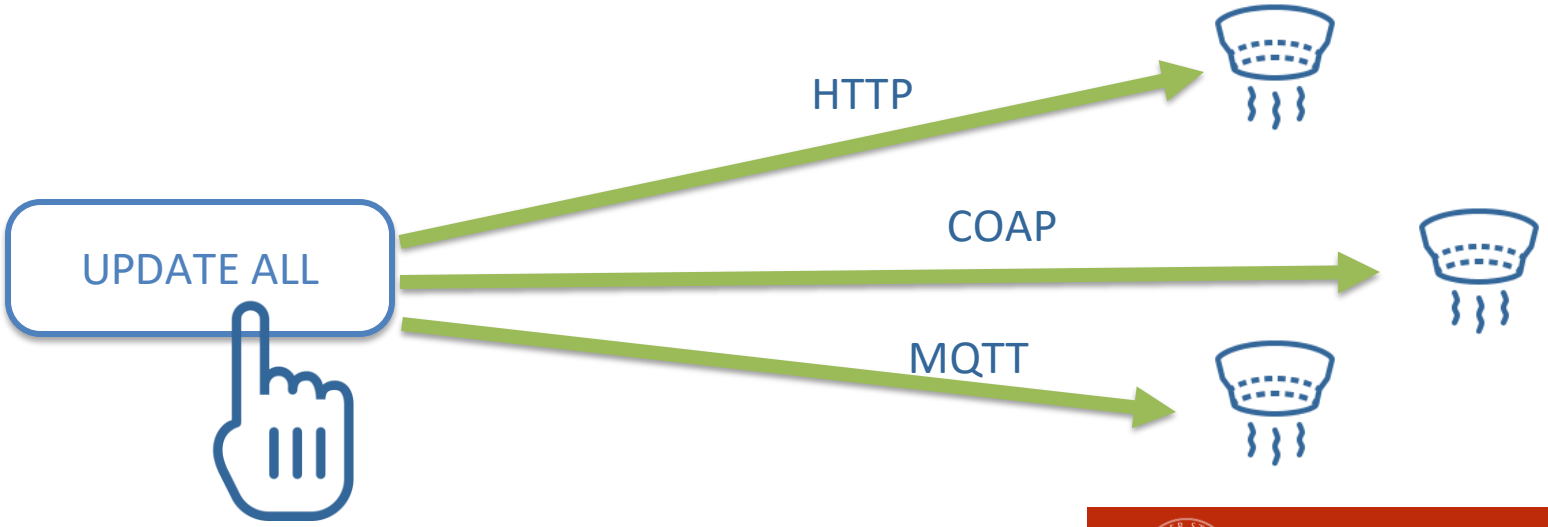
Thing Application (TD) and Mashup Application (MA)

- **Thing application**: it is the source code that implements the *behavior* of a Thing, i.e. the list of properties, actions and events formally defined in its Thing Description.

- **Mashup Applications**: applications producing new outputs or providing new services from a set of existing Things

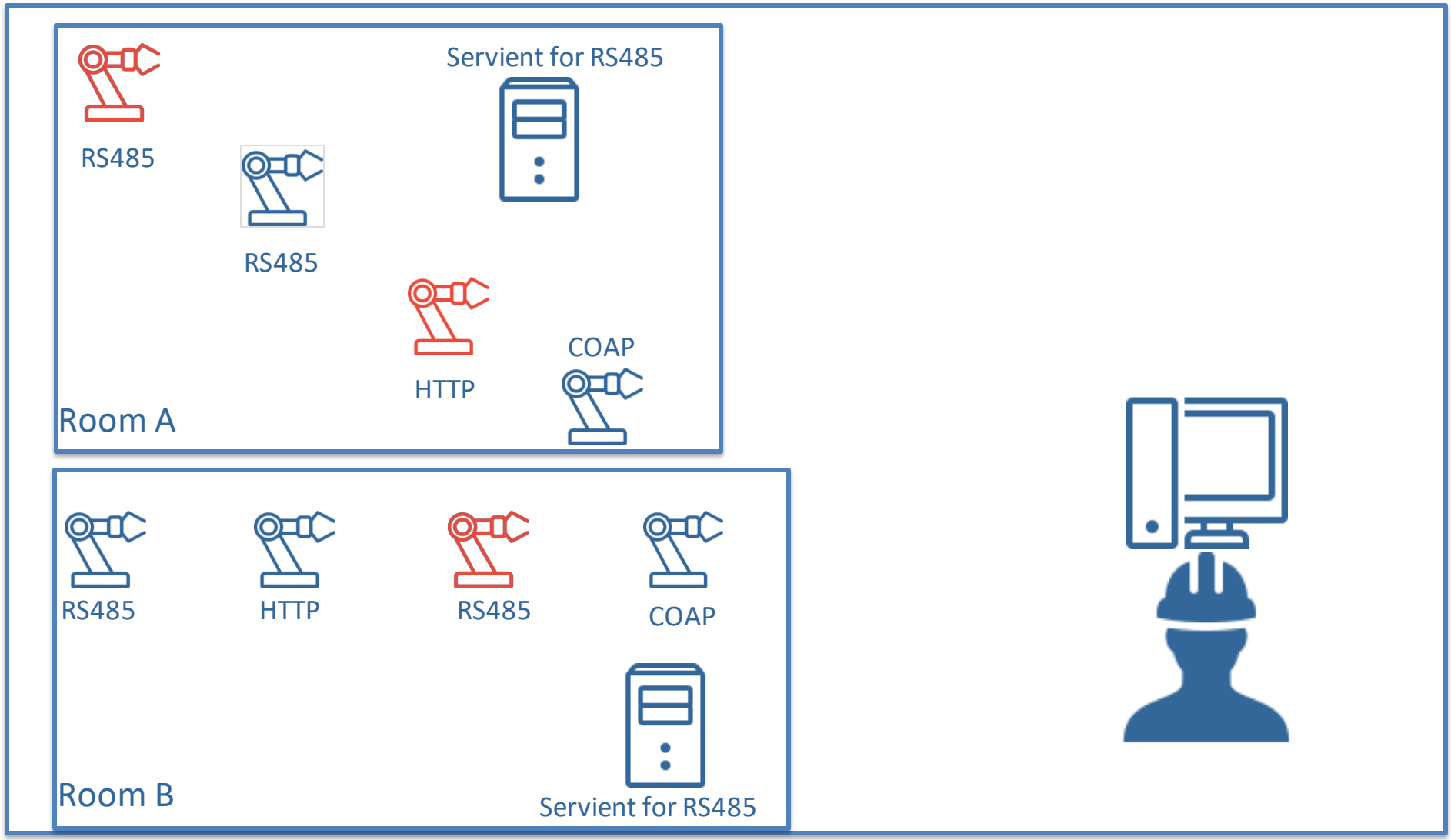
Automatic deploy of TA software on Things

WoT STORE enables the automatic installation and execution of the application code on target Thing(s). This is implemented through an additional Thing search engine, which allows users to issue semantic queries (e.g. indicating the Thing type and capabilities)

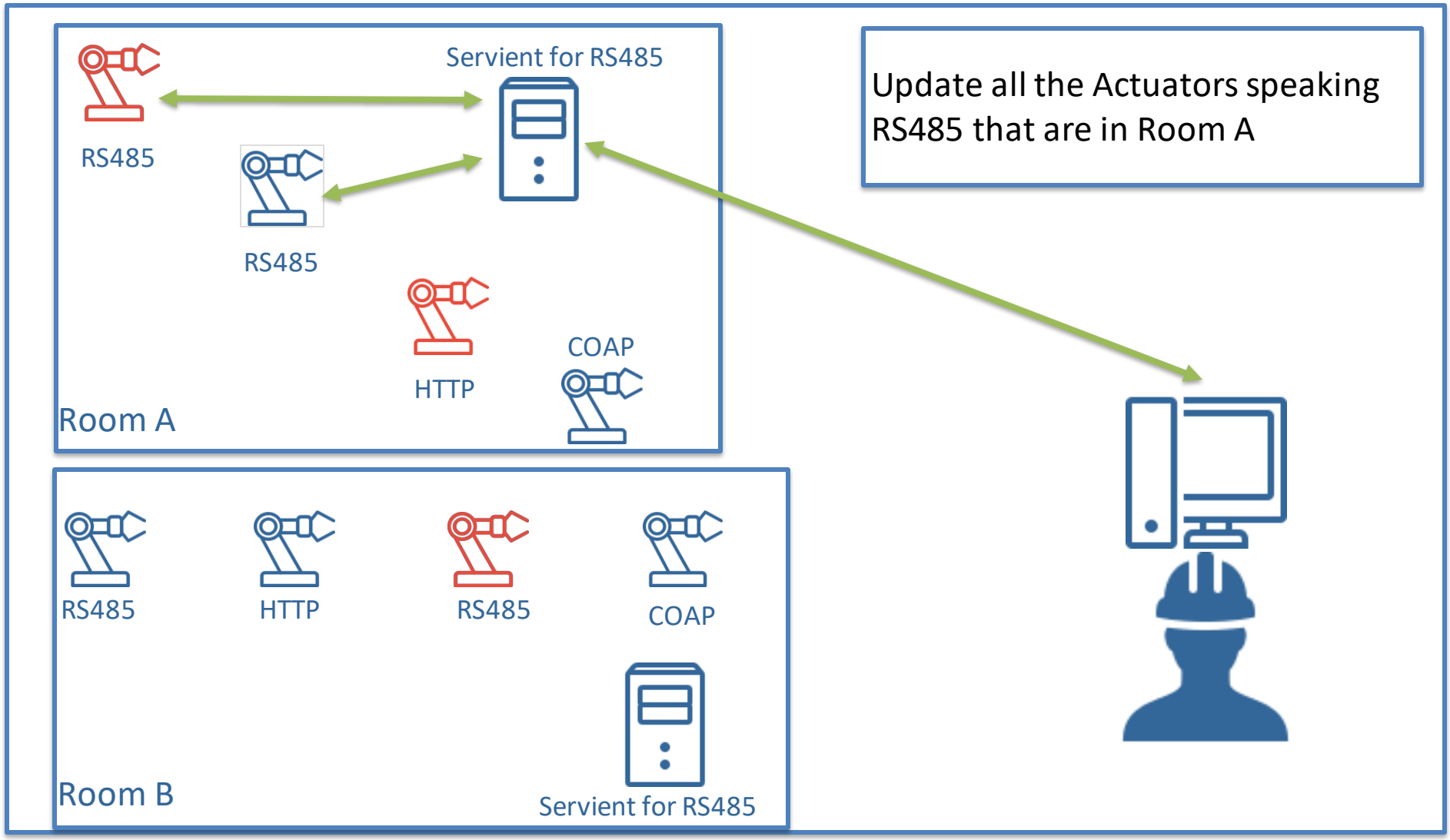


https://icons8.com

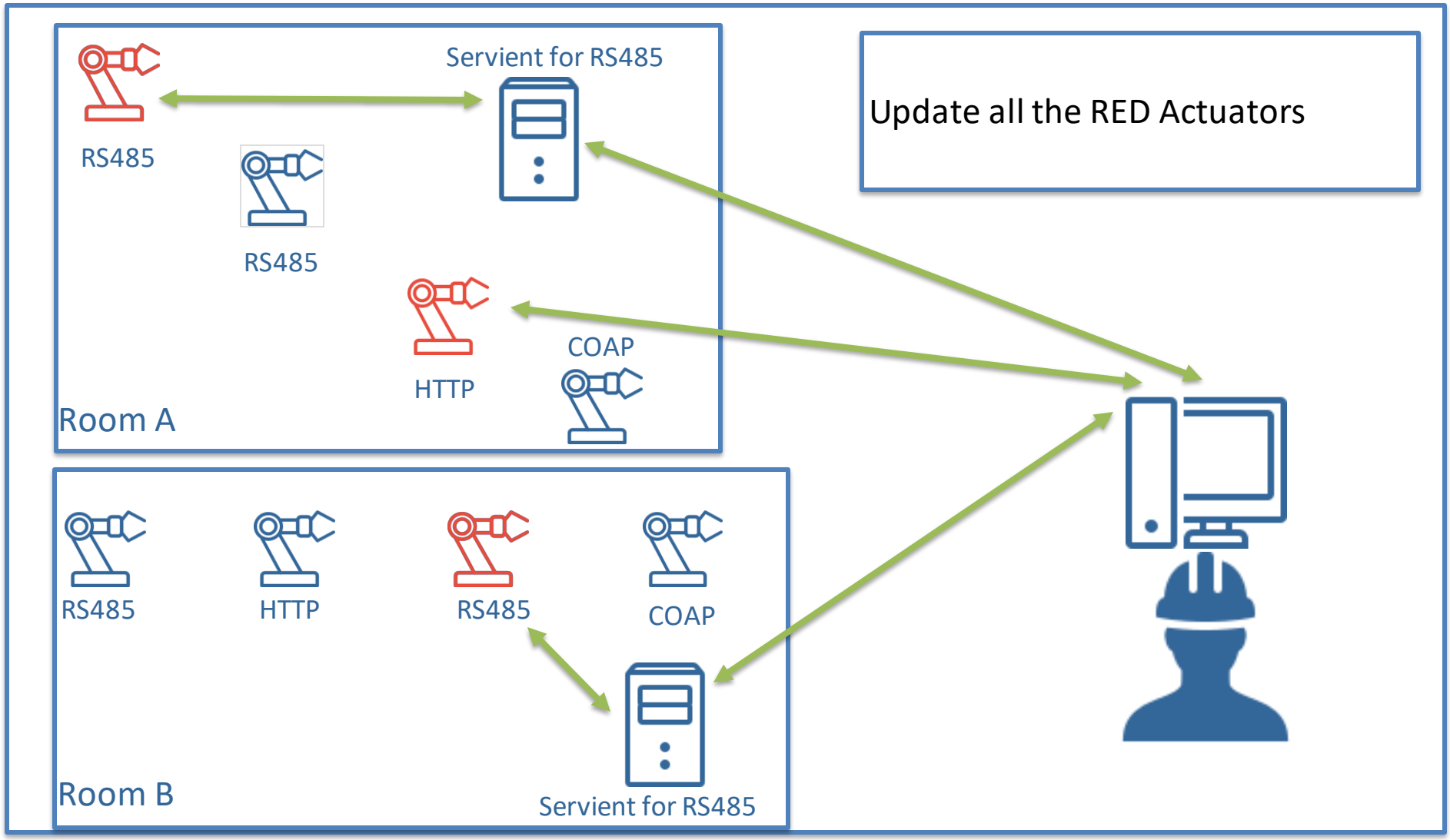
UPDATE-ALL: Industrial use case



UPDATE-ALL: Industrial use case

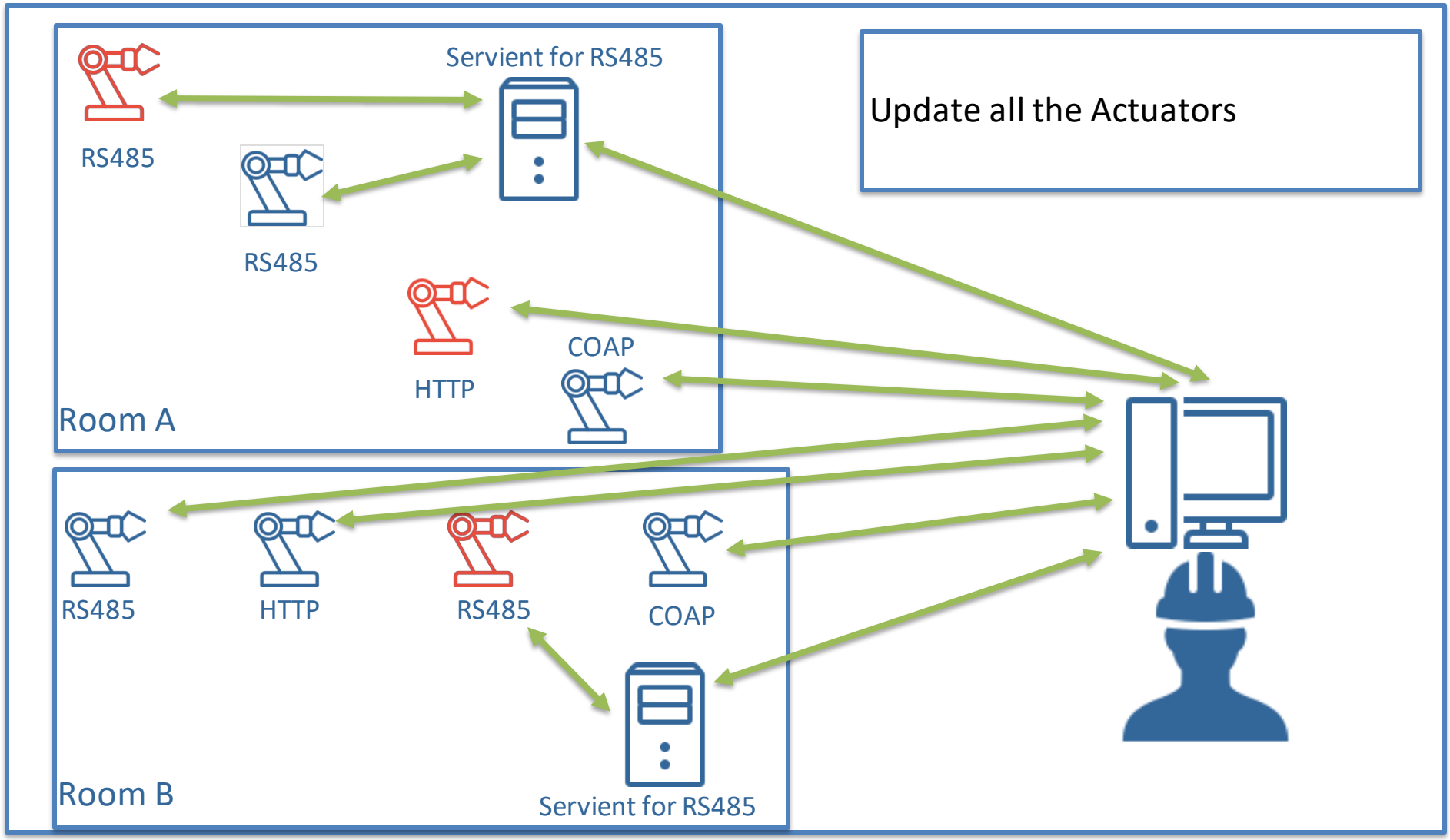


UPDATE-ALL: Industrial use case

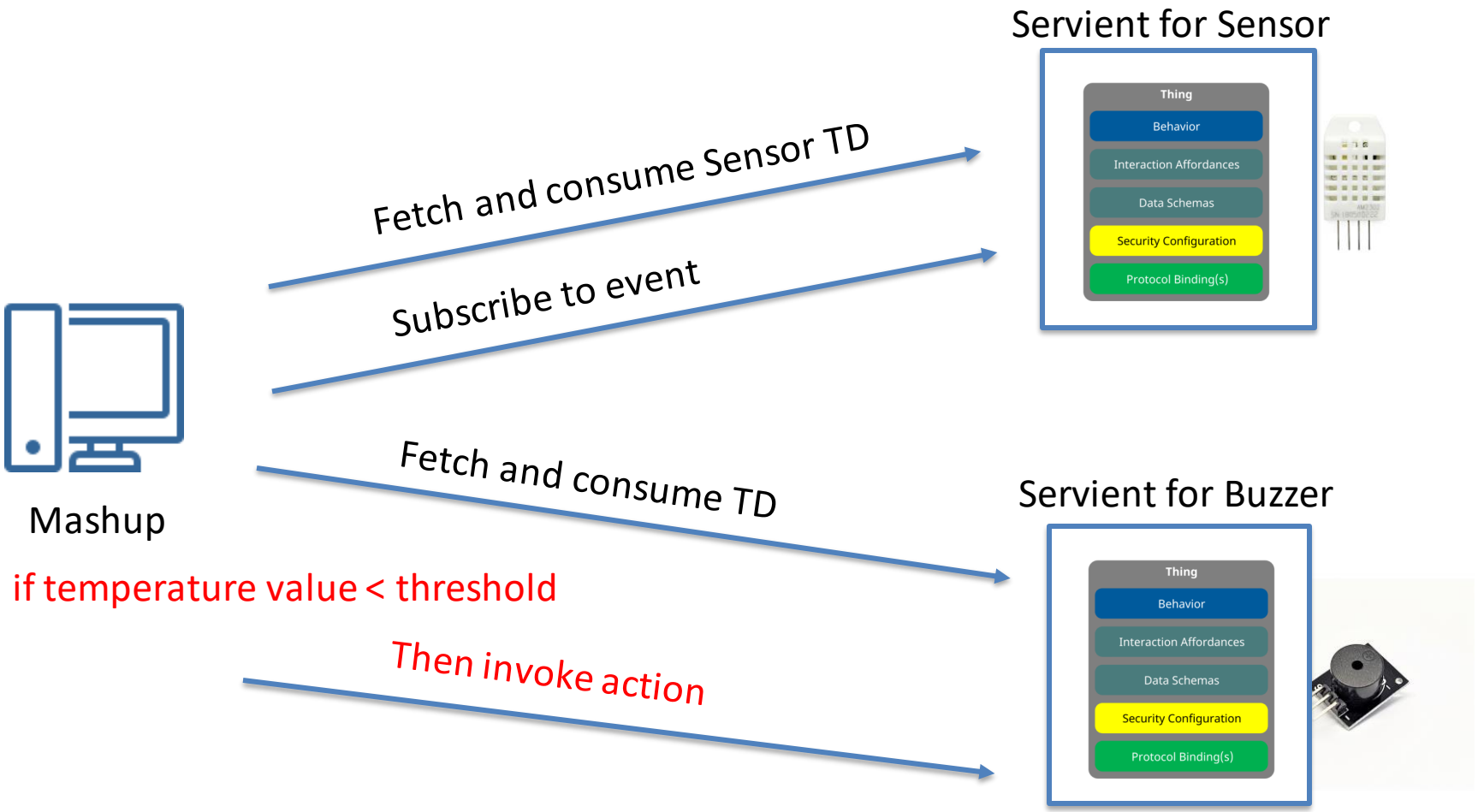


UPDATE-ALL: Industrial use case

FACTORY



Demo



Mashup subscribes to sensor's events. If gets a temperature value below a threshold, then it invokes an action on the buzzer to make it play.

Understand the programming flow with node-wot

1. Install node-wot as a library
2. Instantiate and run a servient
3. Retrieve the Thing Description
4. Consume the Thing
5. Interact with the Thing

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Thank you for the attention!

Prof. Marco Di Felice, Luca Sciullo

marco.difelice3@unibo.it, luca.sciullo@unibo.it