Systèmes embarqués

Architecting IoT Systems, Beyond Functional Correctness

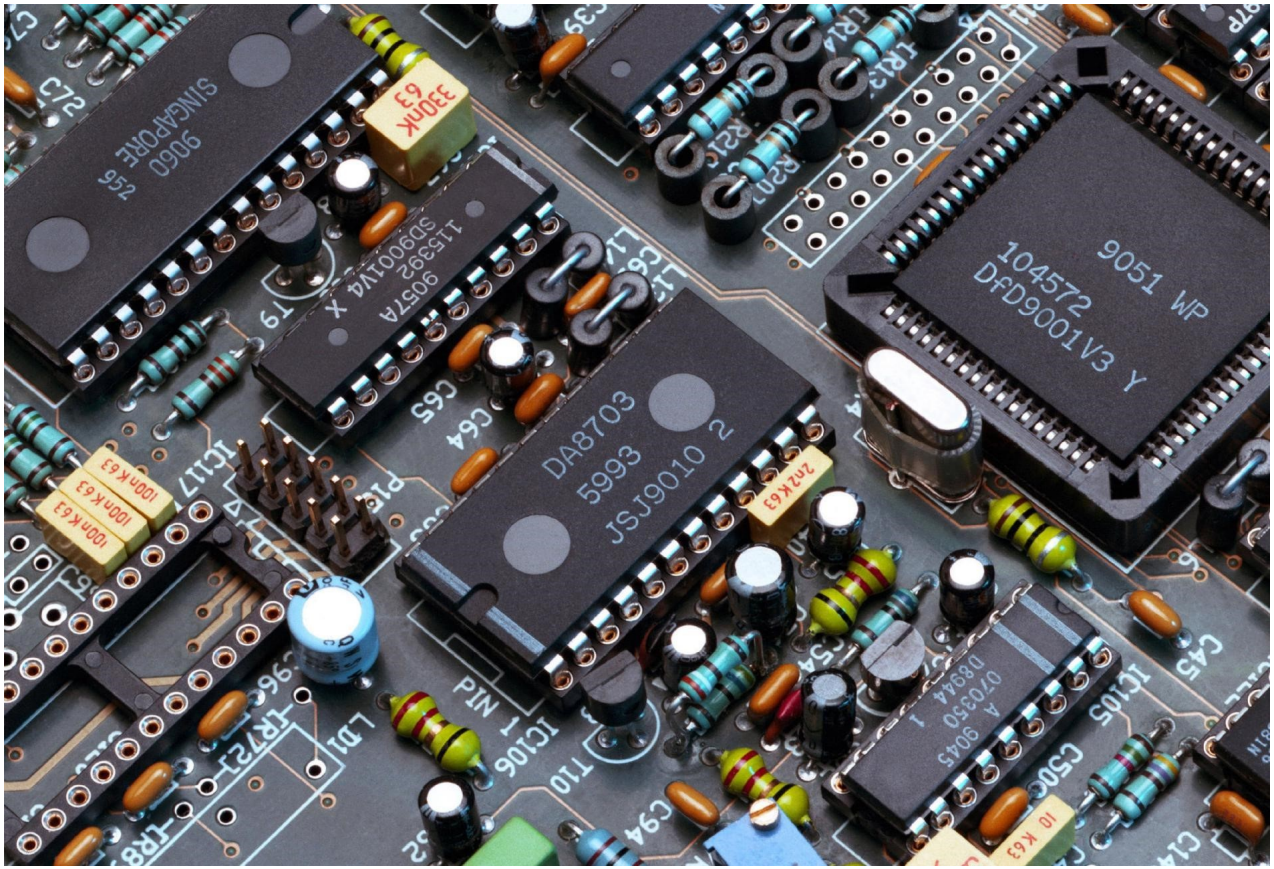
Julien Deantoni

IoT yesterday

- Information gathering for analysis / dash boarding
- challenges
  - « big data »
  - scalability

Pictures shamefully taken from the web

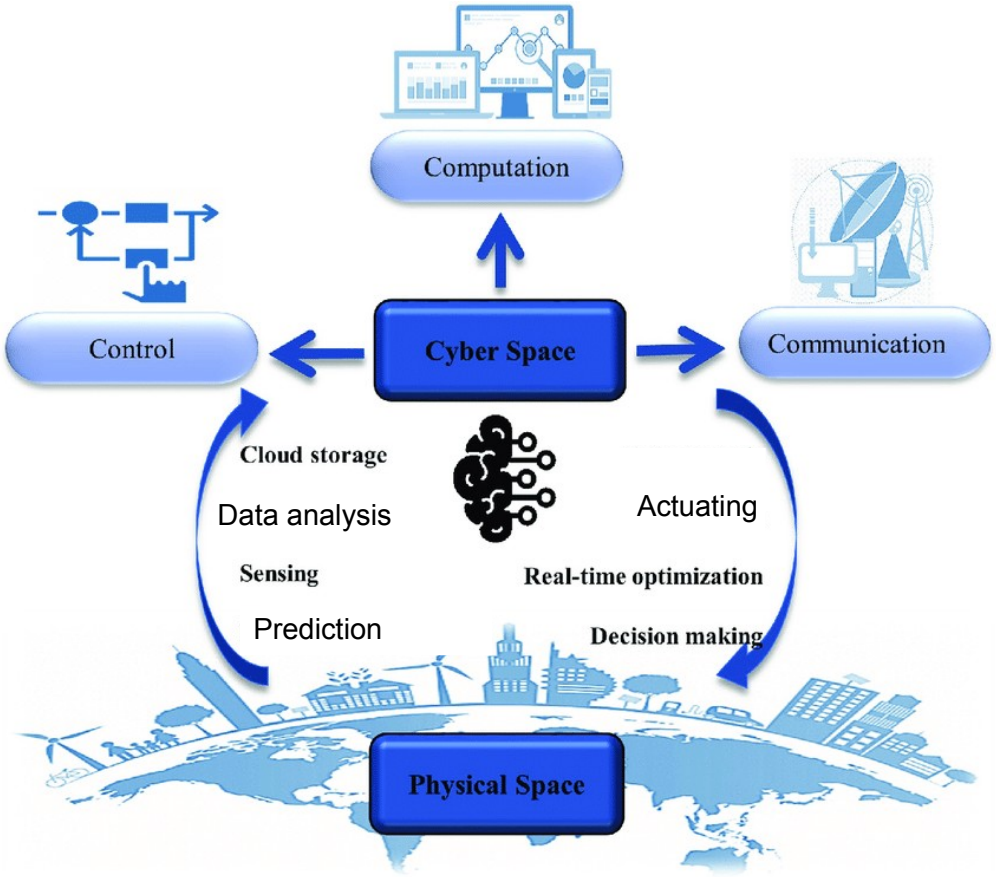
(yesterday?) Embedded systems



- Embedded software with few resources and time constraints
- challenges
  - Adequacy between algorithm and architecture
  - Schedulability
  - (Energetic sobriety)

Pictures shamefully taken from the web

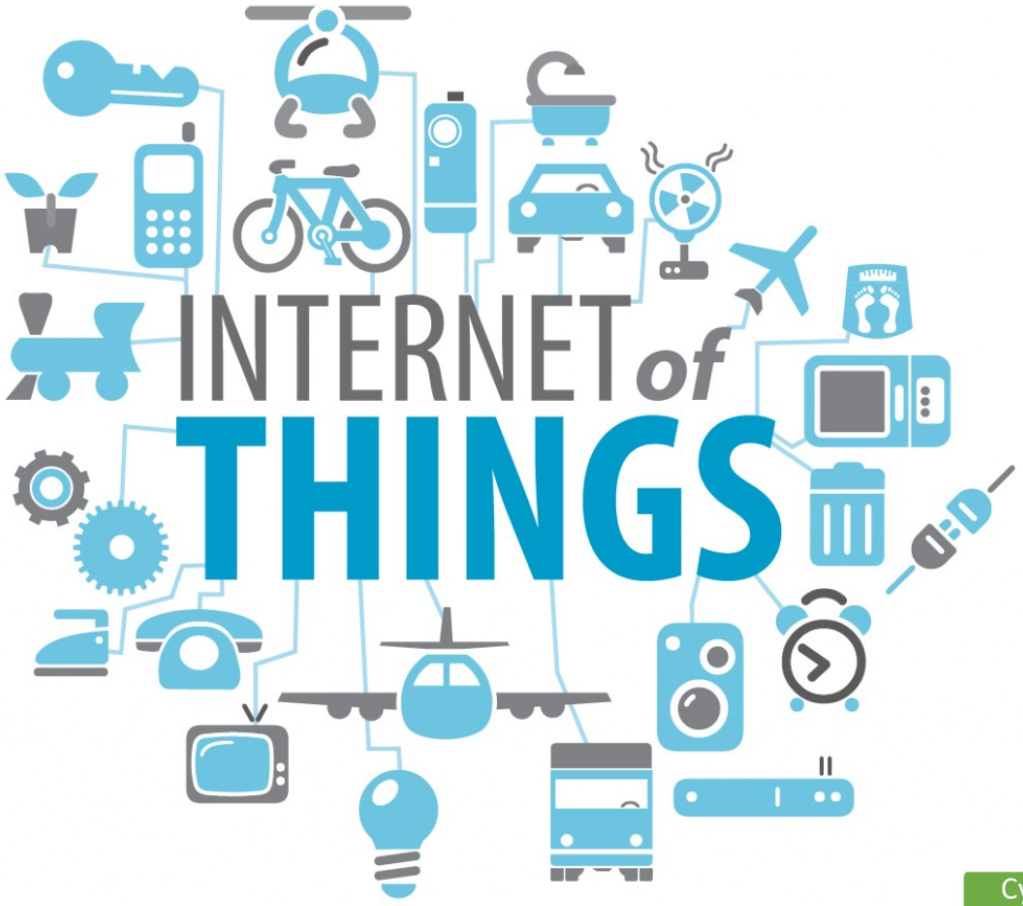
Cyber Physical Systems

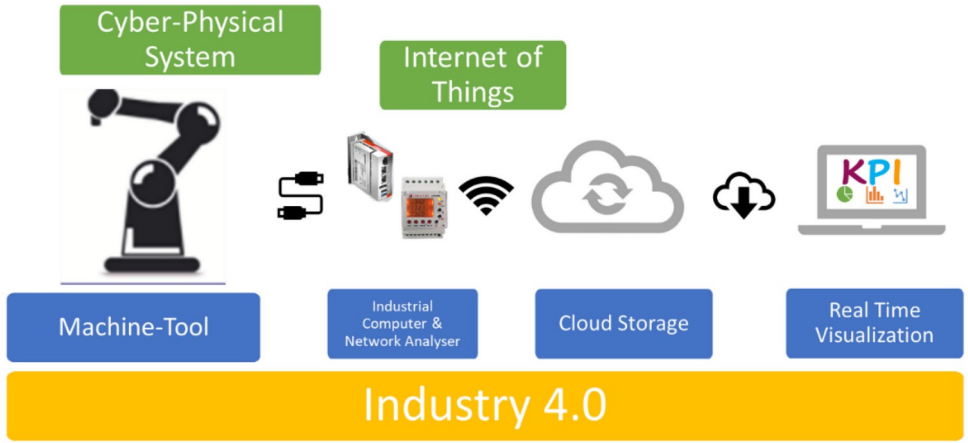


Slightly modified from:
Robust optimal design of FOPID cont roller for five bar linkage robot in a Cy ber-Physical System: A new simulatio n-optimization approach

- Focus on the deep and intense link between the computational and physical elements
- challenges
  - Design with different stakeholders
  - Conception considering intertwined concerns
  - Analysis considering intertwined concerns

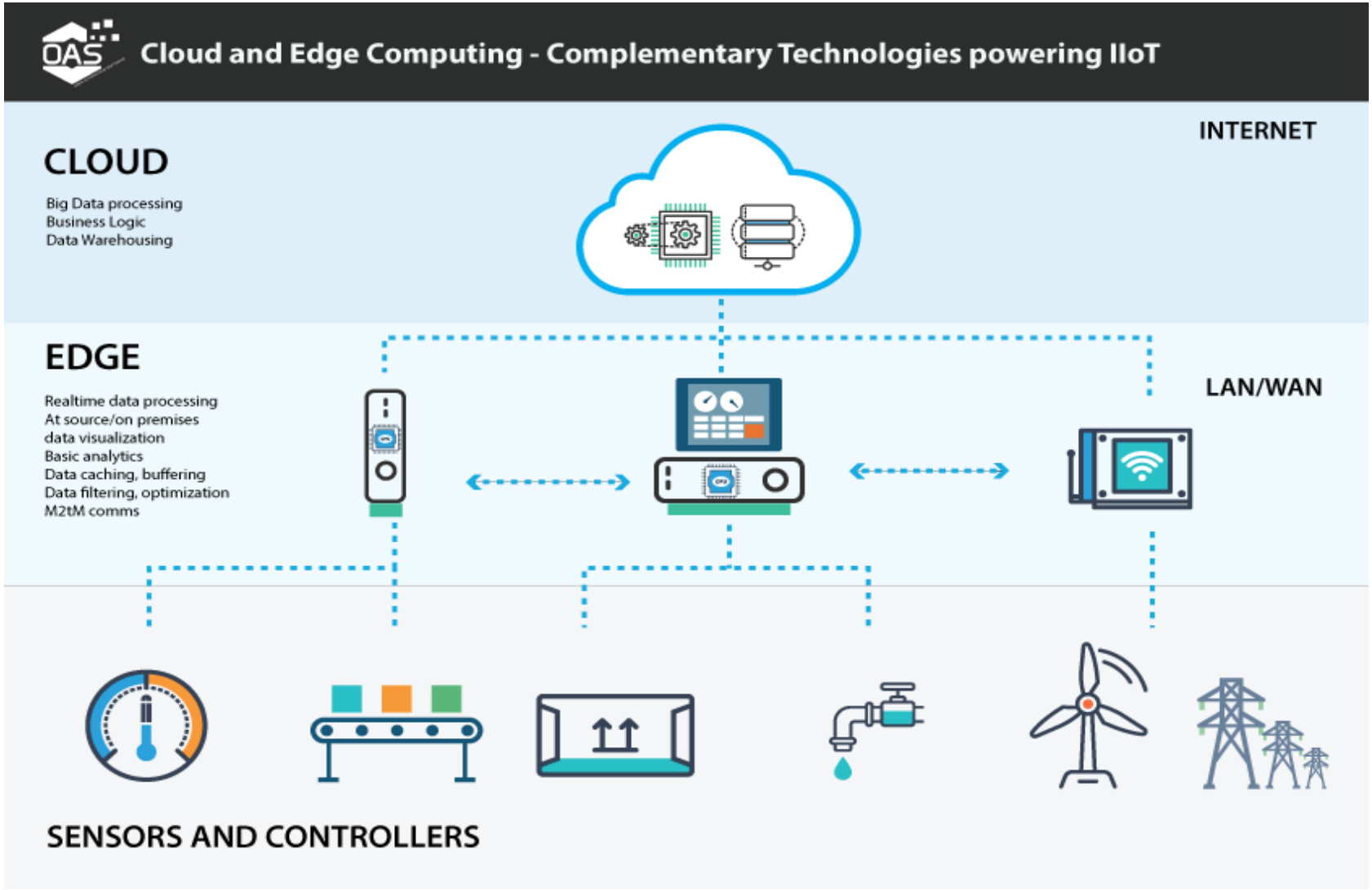
IoT and CPS convergence





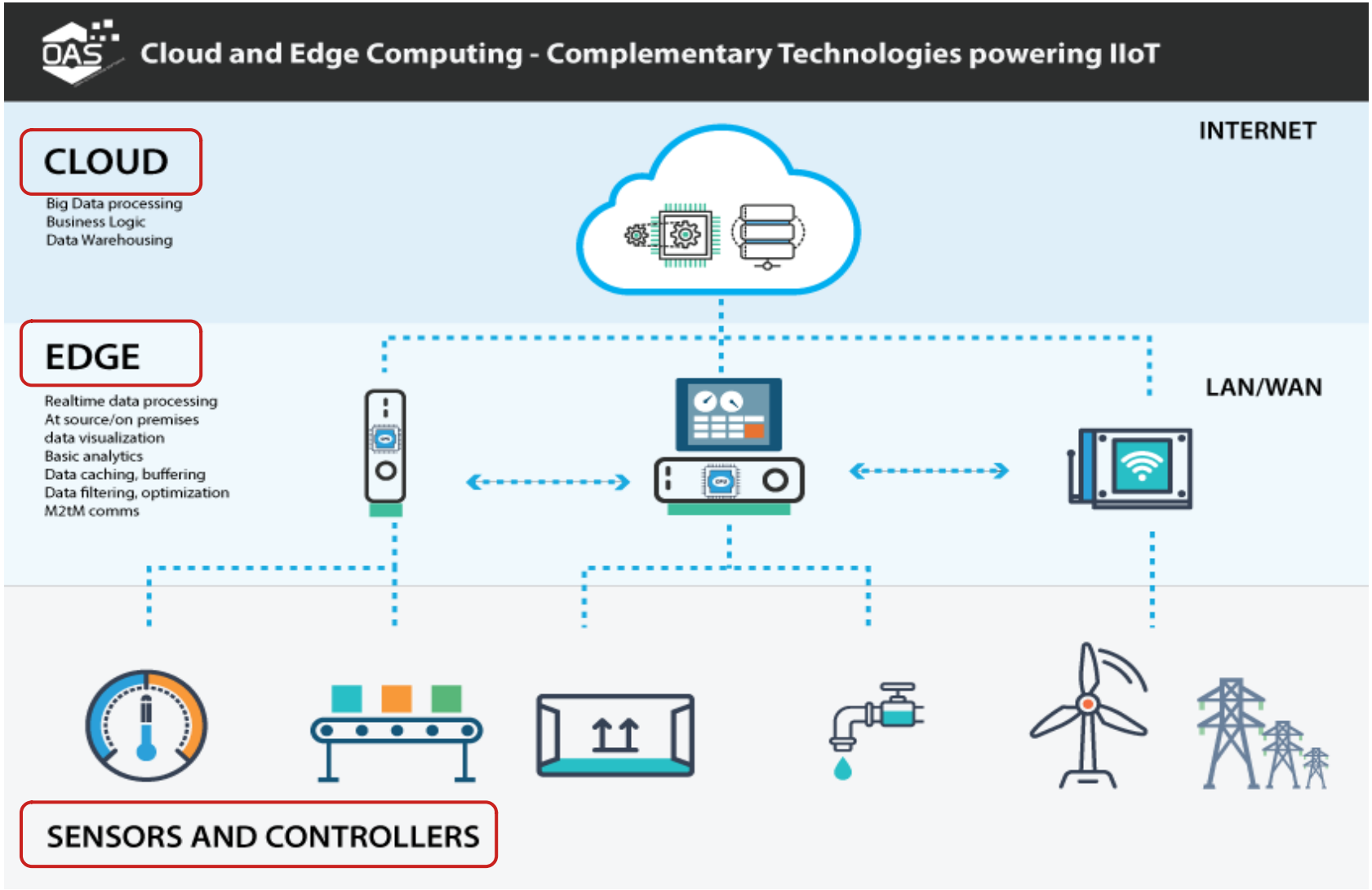
Pictures shamefully taken from the web

IoT and CPS convergence, IoT view



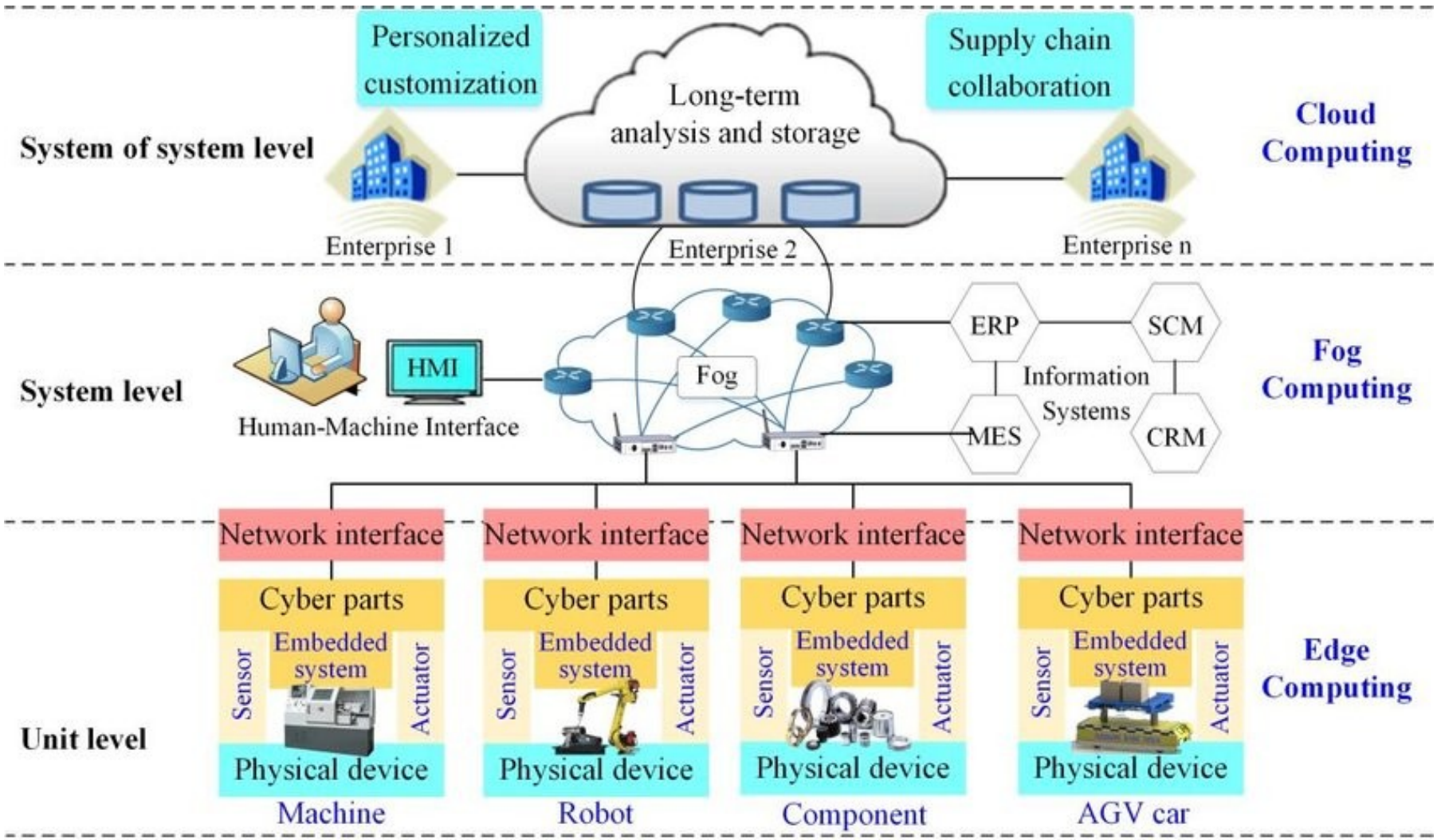
https://antlysis.com/wp-content/uploads/2019/08/edge-v-cloud-computing-graphic.png

IoT and CPS convergence, IoT view



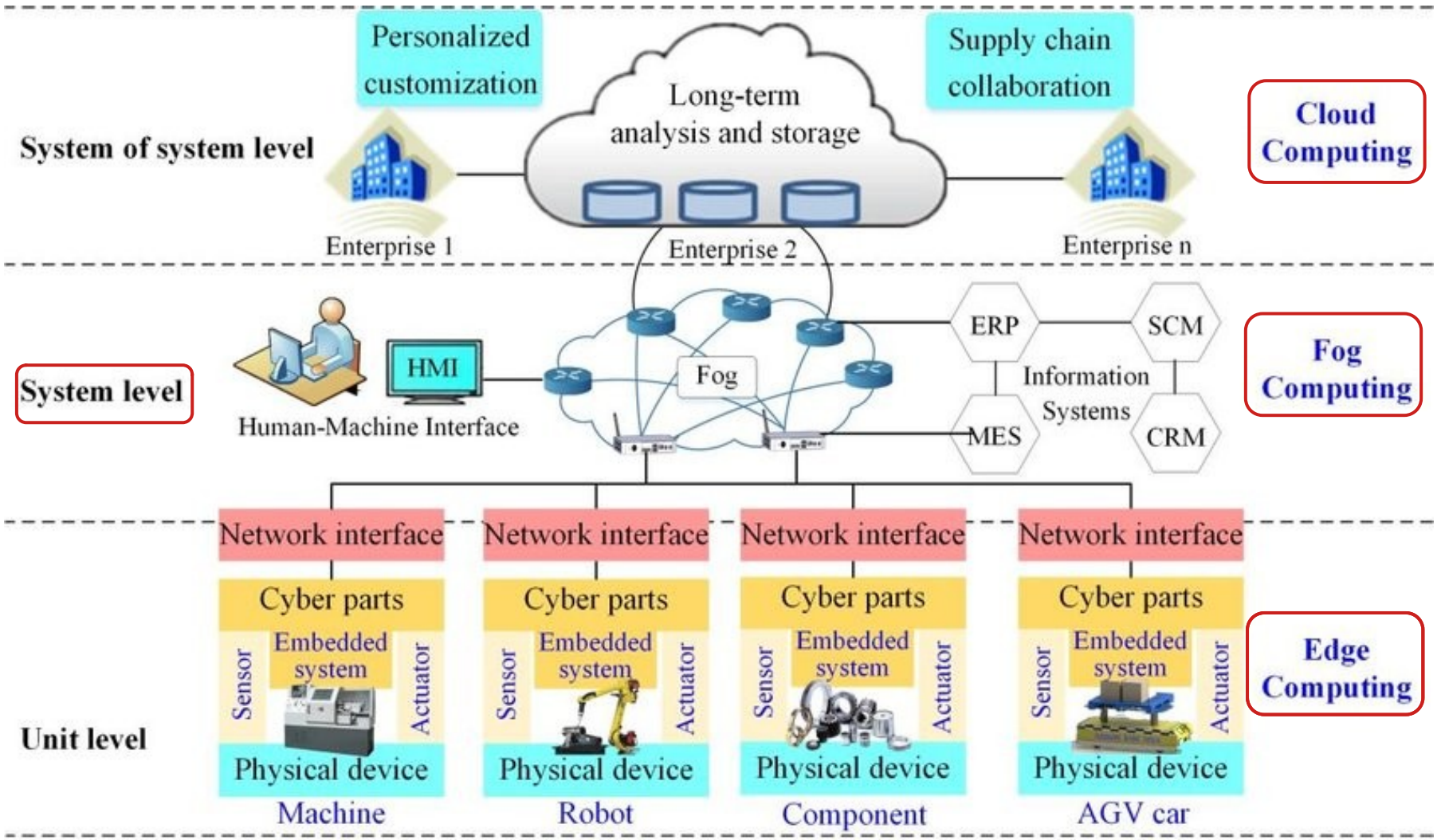
https://antlysis.com/wp-content/uploads/2019/08/edge-v-cloud-computing-graphic.png

IoT and CPS convergence, CPS view (1/3)



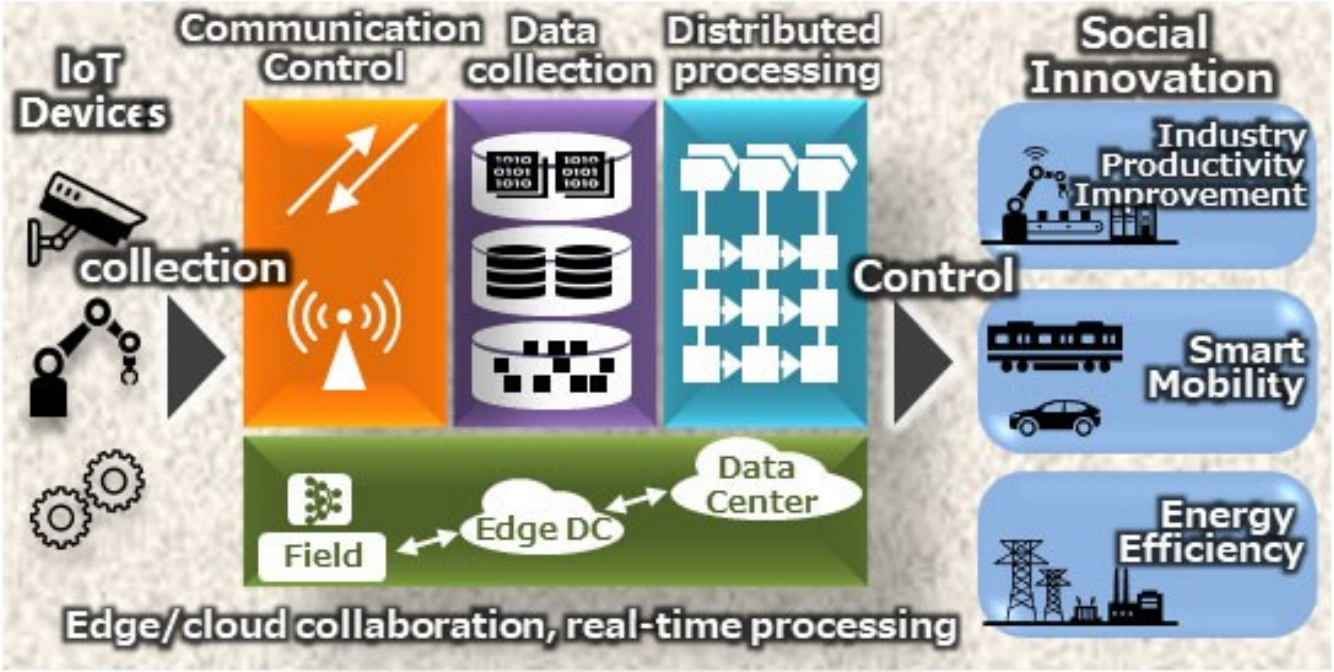
Modeling of Cyber-Physical Systems and Digital Twin Based on Edge Computing, Fo g Computing and Cloud Computing Towards Smart Manufacturing

IoT and CPS convergence, CPS view (1/3)



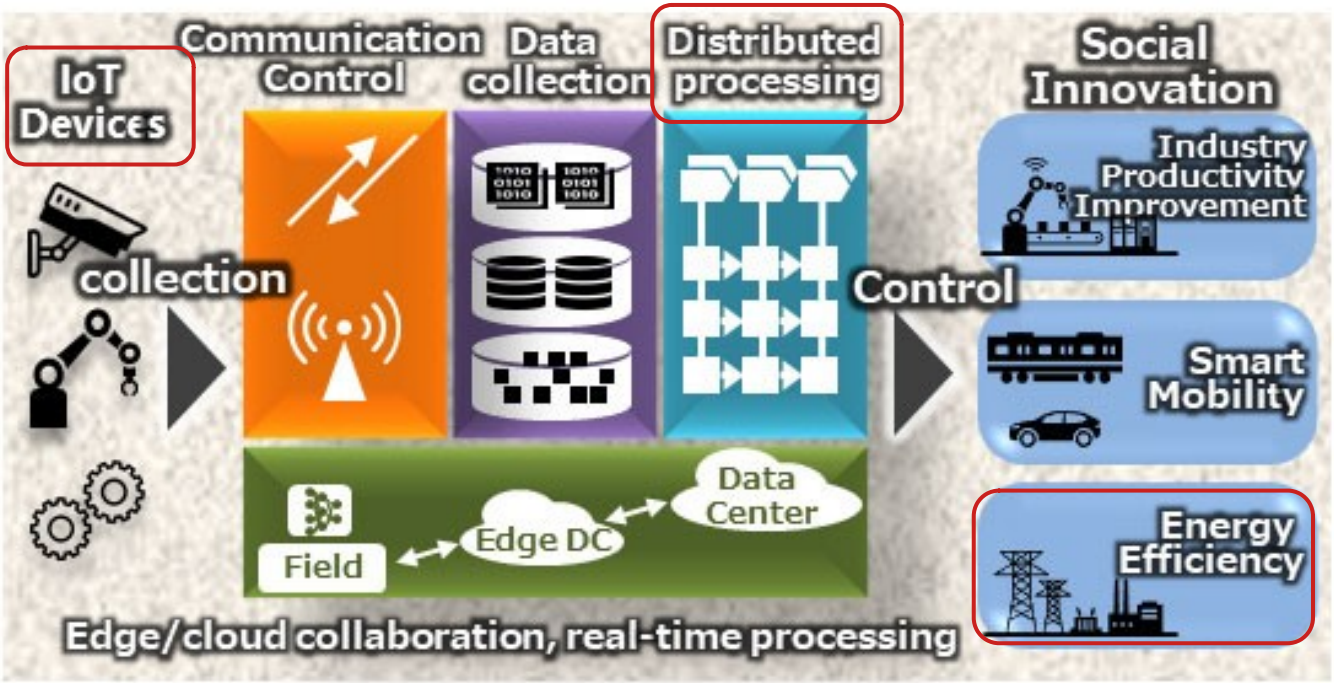
Modeling of Cyber-Physical Systems and Digital Twin Based on Edge Computing, Fo g Computing and Cloud Computing Towards Smart Manufacturing

IoT and CPS convergence, CPS view (2/3)



https://www.hitachi.co.jp/recruit/en/newgraduate/jm-navi/ai_digital/

IoT and CPS convergence, CPS view (2/3)



https://www.hitachi.co.jp/recruit/en/newgraduate/jm-navi/ai_digital/

IoT and CPS convergence, CPS view (3/3)

https://www.hitachi.co.jp/recruit/en/newgraduate/jm-navi/ai_digital/

IoT and CPS convergence, CPS view (3/3)

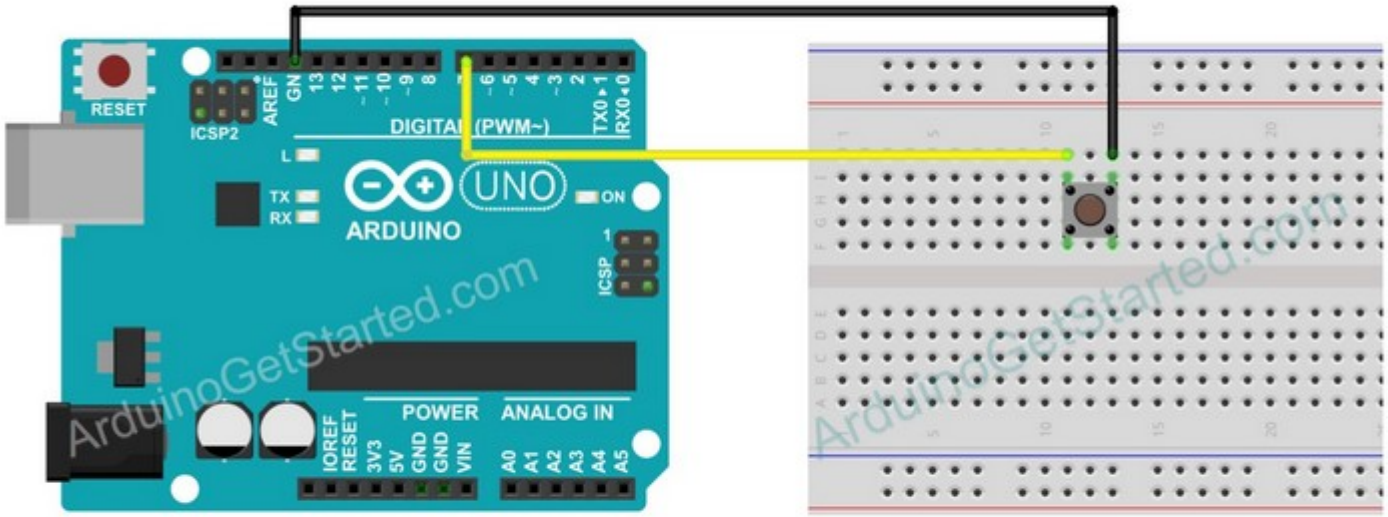
https://www.hitachi.co.jp/recruit/en/newgraduate/jm-navi/ai_digital/

Why this course ?

- Importance of the « hardware » architecture :
  - Physical world ↔ Thing/device ↔ edge/fog ↔ cloud.
  - heterogeneous hardware, exhibiting heterogeneous resources and capabilities
- Emphasis on different concerns :
  - Energy efficiency (how long my battery will last ? How long the energy infrastructure will last ?)
  - Resilience (can the system adapt to a degraded mode ? Can it restart after a crash?)
  - Safety and security (what if someone controls your vehicle ? What if the network is down for some minutes?)
  - Compliance with regulations and laws (can I spread any data in the cloud?)
  - Time constraints, since physical environment is ruthless ; no « pause » of physical laws.
  - *And more !*

Why this course ?

Wiring Diagram

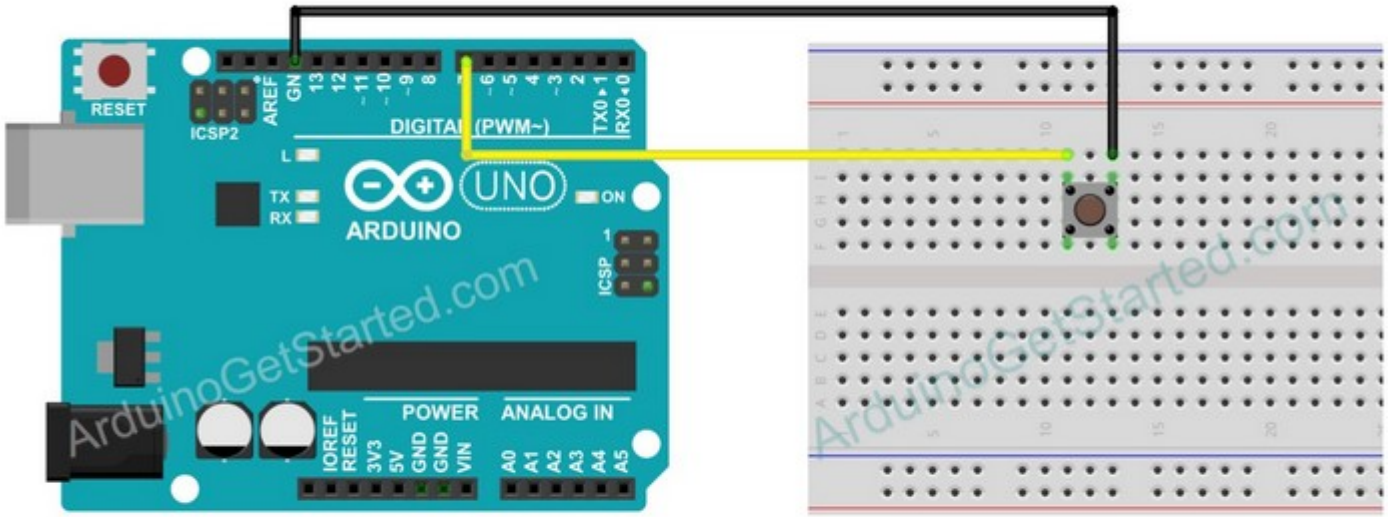


ArduinoGetStarted.com

```
16 void loop() {
17   // read the state of the switch/button:
18   currentState = digitalRead(BUTTON_PIN);
19
20   if(lastState == LOW && currentState == HIGH)
21     Serial.println("The state changed from LOW to HIGH");
22
23   // save the last state
24   lastState = currentState;
25 }
```

Why this course ?

Wiring Diagram



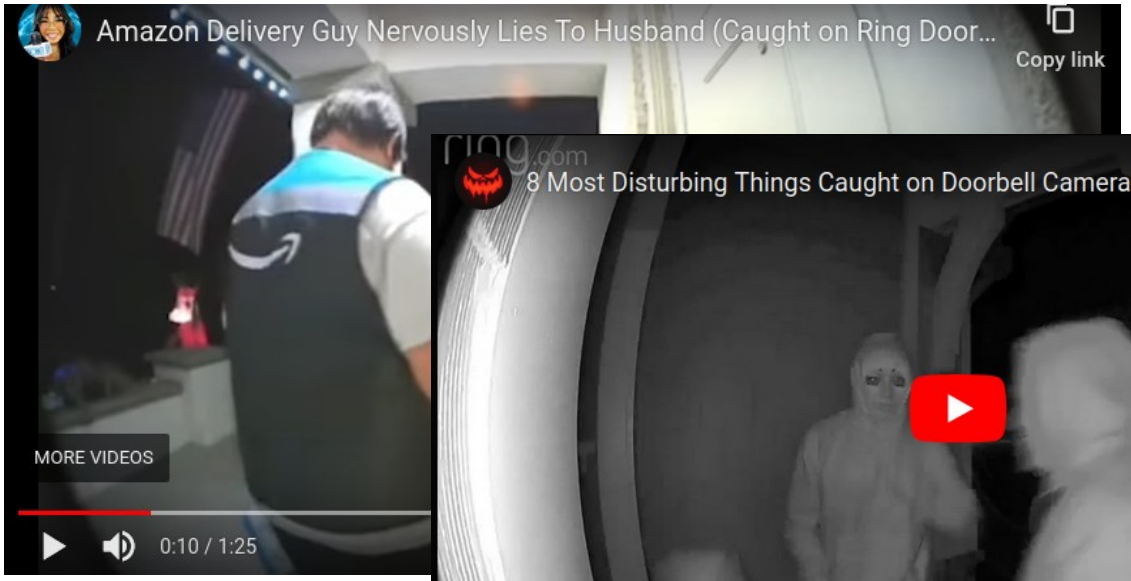
ArduinoGetStarted.com

```
16 void loop() {
17   // read the state of the switch/button:
18   currentState = digitalRead(BUTTON_PIN);
19
20   if(lastState == LOW && currentState == HIGH)
21     Serial.println("The state changed from LOW to HIGH");
22
23   // save the last state
24   lastState = currentState;
25 }
```

Energy efficiency ?

Why this course ?

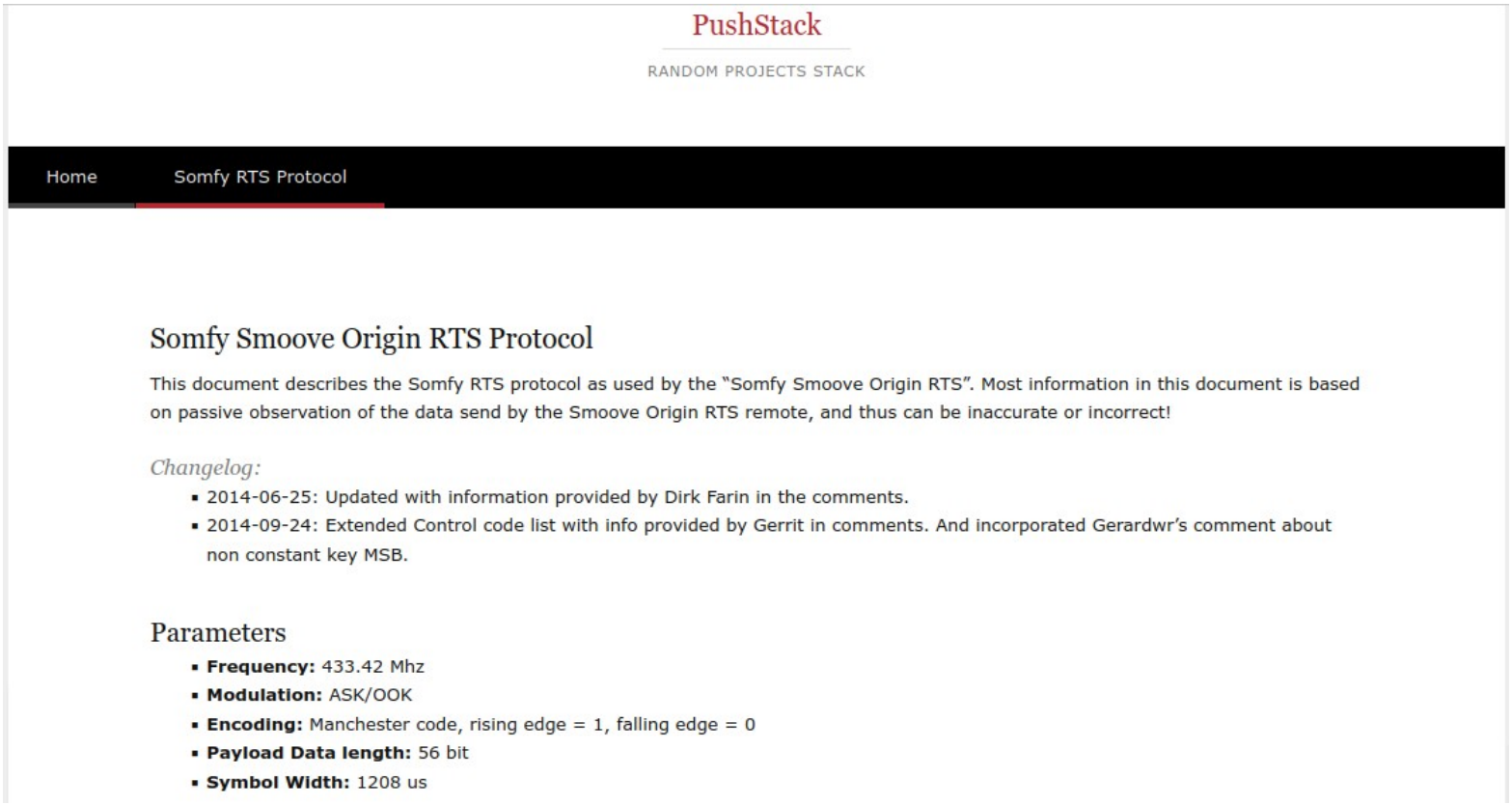
https://support.ring.com/hc/fr-fr/articles/360008085463-Obligations-de-Ring-en-vertu-du-RGPD



Privacy law ?

https://support.ring.com/hc/en-us/articles/360000885463-Obligations-de-Ring-en-vertu-du-RGPD

Why this course ?



PushStack

RANDOM PROJECTS STACK

Home | Somfy RTS Protocol

Somfy Smoove Origin RTS Protocol

This document describes the Somfy RTS protocol as used by the "Somfy Smoove Origin RTS". Most information in this document is based on passive observation of the data send by the Smoove Origin RTS remote, and thus can be inaccurate or incorrect!

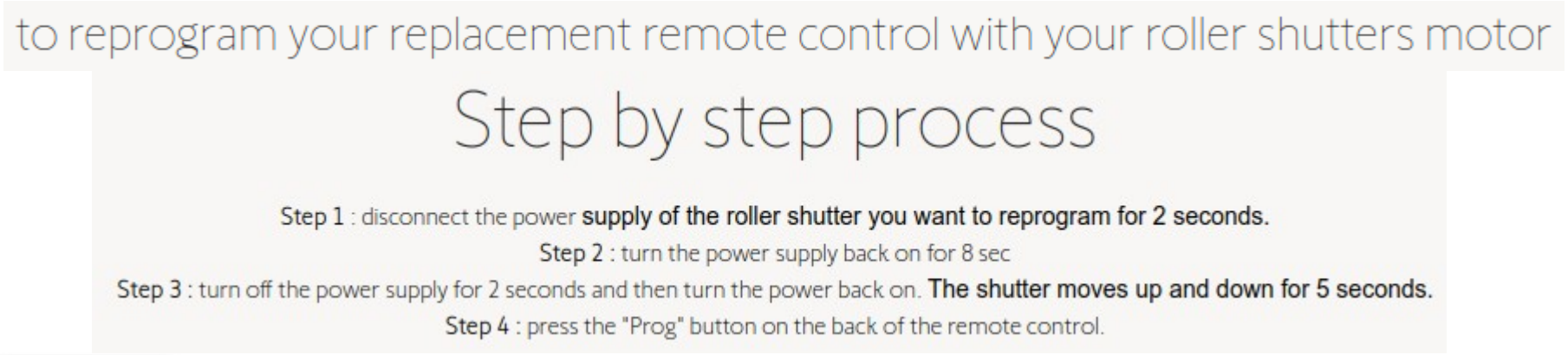
Changelog:

- 2014-06-25: Updated with information provided by Dirk Farin in the comments.
- 2014-09-24: Extended Control code list with info provided by Gerrit in comments. And incorporated Gerardwr's comment about non constant key MSB.

Parameters

- **Frequency:** 433.42 Mhz
- **Modulation:** ASK/OOK
- **Encoding:** Manchester code, rising edge = 1, falling edge = 0
- **Payload Data length:** 56 bit
- **Symbol Width:** 1208 us

https://pushstack.wordpress.com/somfy-rts-protocol/



to reprogram your replacement remote control with your roller shutters motor

Step by step process

Step 1 : disconnect the power **supply of the roller shutter you want to reprogram for 2 seconds.**

Step 2 : turn the power supply back on for 8 sec

Step 3 : turn off the power supply for 2 seconds and then turn the power back on. **The shutter moves up and down for 5 seconds.**

Step 4 : press the "Prog" button on the back of the remote control.

Why this course ?

PushStack

RANDOM PROJECTS STACK

Home | Somfy RTS Protocol

Somfy Smoove Origin RTS Protocol

This document describes the Somfy RTS protocol as used by the "Somfy Smoove Origin RTS". Most information in this document is based on passive observation of the data send by the Smoove Origin RTS remote, and thus can be inaccurate or incorrect!

Changelog:

- 2014-06-25: Updated with information provided by Dirk Farin in the comments.
- 2014-09-24: Extended Control code list with info provided by Gerrit in comments. And incorporated Gerardwr's comment about non constant key MSB.

Parameters

- **Frequency:** 433.42 Mhz
- **Modulation:** ASK/OOK
- **Encoding:** Manchester code, rising edge = 1, falling edge = 0
- **Payload Data length:** 56 bit
- **Symbol Width:** 1208 us

https://pushstack.wordpress.com/somfy-rts-protocol/

Security ?

to reprogram your replacement remote control with your roller shutters motor

Step by step process

Step 1 : disconnect the power **supply of the roller shutter you want to reprogram for 2 seconds.**

Step 2 : turn the power supply back on for 8 sec

Step 3 : turn off the power supply for 2 seconds and then turn the power back on. **The shutter moves up and down for 5 seconds.**

Step 4 : press the "Prog" button on the back of the remote control.

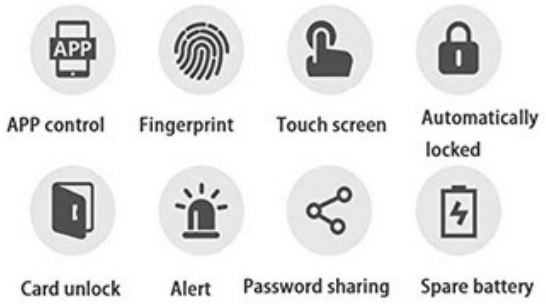
Why this course ?

Black Smart Fingerprint Deadbolt Lock WiFi Appliquer Remote Control Auto-Lock IC Card Door Lock Smart Home

Marque : RLGS

230^{01} €

Intelligent digital lock, compact size and favorable price,this lock brings you security and convenience



Get the unlock record when someone open the lock;

Why this course ?

Black Smart Fingerprint Deadbolt Lock WiFi Appliquer Remote Control Auto-Lock IC Card Door Lock Smart Home

Marque : RLGS

230[01] €

reliability ?

privacy ?

Intelligent digital lock, compact size and favorable price,this lock brings you security and convenience

Get the unlock record when someone open the lock;

Why this course ?

Black Smart Fingerprint Deadbolt Lock WiFi Appliquer Remote Control Auto-Lock IC Card Door Lock Smart Home

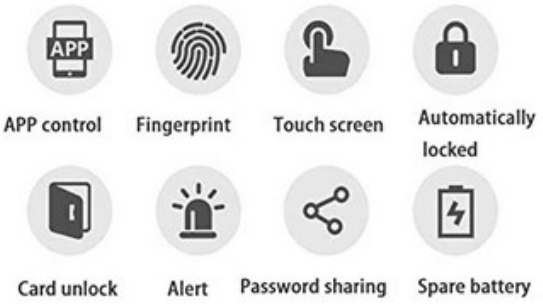
Marque : RLGS

230^{01} €

reliability ?

privacy ?

Intelligent digital lock, compact size and favorable price,this lock brings you security and convenience



Get the unlock record when someone open the lock;

Question : bjr ! cet serrure est elle étanche ? ces pour un portail extérieur

Réponse : Non, cette serrure à télécommande n'est pas prévue pour l'extérieur.
Par Timide MEURTEAU le 1 juin 2022

https://www.amazon.fr/Verrouillage-t%C3%A9l%C3%A9commande-Intelligent-contr%C3%B4le-Deadbolt/dp/B07QHT9ZCD/ref=sr_1_5?__mk_fr_FR=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=1HLQ3X3V1RPF8&keywords=remote+door+lock&qid=1668945602&sprefix=remote+doorlock%2Caps%2C75&sr=8-5

Why this course ?

- Do you think these products have all been badly developed ?
  - NO ! Often there is no ultimate solution. This is a matter of trade-offs between different concerns (costs, time to market, society evolution embracing, targeted environment, etc)

- Trade-offs can be done only when you have done a global risk analysis and correctly defined (some) extra functional requirements.
  - This cannot be done by thinking at the code level : need for architecture reasoning

→ This is what will be investigated with an IoT project development in this course.

Tentative Agenda

1) Introduction (me)
    1) **IoT architecture description**
    2) project discovery
2) Cloud in the IoT, pros and cons (Nicolas Ferry).
3) Review of Functional Correctness (me)
4) Extra functional concerns #1 (me)
    - Timing aspects, Power concern
5) Extra functional concerns #2 (Nicolas Ferry and me ?)
    - Security, Reliability, Resilience, Safety
6) Review of the proposed architecture (me)
7) Project Result Demo (Nicolas Ferry and me ?)
8) Project presentations (Nicolas Ferry and me)

IoT architecture description

- Software Architecture
  - Software architecture refers to the **fundamental structures** of a software system and the discipline of creating such structures and systems. Each structure comprises **software elements, relations among them, and properties of both elements and relations.** [Wikipedia]
  - The software architecture of a system represents the **design decisions related to overall system structure and behavior**. Architecture **helps stakeholders understand and analyze** how the system will achieve **essential qualities** such as modifiability, availability, and security [SEI].
  - Software architecture supports analysis of system qualities when teams are making **decisions about the system rather than after implementation, integration, or deployment** [SEI].

⇒ Language and tools for software architecture can be different depending on what is fundamental for you, what are the key properties for you, what are the essential qualities for you, and more globally what the decisions are based upon

IoT architecture description

- Software Architecture for IoT application
  - Not supported by a dedicated language yet; mainly because it depends on the properties and essential qualities you want to focus on.
  - Should embrace the main characteristics of IoT systems
    - Many IoT systems are thought in terms of time series (aka data flow or data stream)

      ⇒ notion of flot port more than interface. Also more based on Components than Object
    - The retention policy between parts of the system and more generally explaining how and when data are moved is usually interesting
      - Not specially the protocole but for example the data is transmitted only on specific condition, or each time a new data is available or only the average on a specific sliding window, etc.

        ⇒ high abstraction level specification of data retention policy
    - When is a functionnality executed, is usually important
      - Periodically every XXXms, when a new data arrive on a specific port, every 4 data, when at least one data arrived on each port, etc

        ⇒ component triggering may be explicitely described
    - Computation time ? Functional Chain ?

IoT architecture description

- Software Architecture for IoT application
  - Not supported by a dedicated language yet; mainly because it depends on the properties and essential qualities you want to focus on.
  - Should embrace the main characteristics of IoT systems
    - Many IoT systems are thought in terms of time series (aka data flow or data stream)

      ⇒ notion of **flow port** more than OO interface. Also more based on **Components** than Object
    - The transfer/communication/rentention policy between parts of the system. More generally explaining how and when data are moved is usually interesting
      - Not specially the protocol but for example specifying if the data is transmitted only on specific condition, or each time a new data is available or only the average on a specific sliding window, etc.

        ⇒ high abstraction level specification of data transfer policy
    - When is a functionnality executed is usually important
      - Periodically every XXXms, when a new data arrive on a specific port, every 4 data, when at least one data arrived on each port, etc

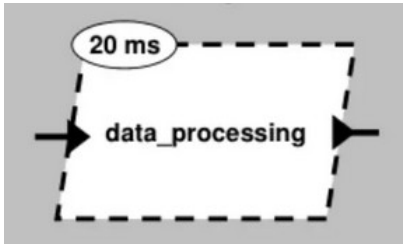
        ⇒ component triggering may be explicitely described

⇒ All this will be kept informal during this course since I do not want any technical debt in the use of a new language/tool. However, you should be consistent and provide legends on how to read your architecture

IoT architecture description

- Software Architecture examples

```
-- model a schedulable flow of control
thread bar_thread                              --  bar_thread is a sporadic thread :
features                                       --  dispatched whenever it
  in_data : in event data port foo_data;       --  receives an event on its "in_data"
properties                                     --  port
  Dispatch_Protocol => Sporadic;
end bar_thread;
```



Thread Dispatch Protocols

Periodic thread (50ms)
- Periodic dispatch of threads, typically with hard deadlines

Aperiodic thread (A)
- Dispatch based on events with arbitrary arrival patterns

Sporadic thread (S:10ms)
- Dispatch based on events with a minimal time between dispatches

Background thread (B)
- Dispatch once and execute until completion

Timed thread (T:10ms)
- Dispatch based on events, or based on timeout if no events occur

Hybrid (H:10ms)
- Dispatch based on events and periodically (combines periodic and aperiodic dispatch protocols)

https://www.slideshare.net/iivanoo/aadl-42305750

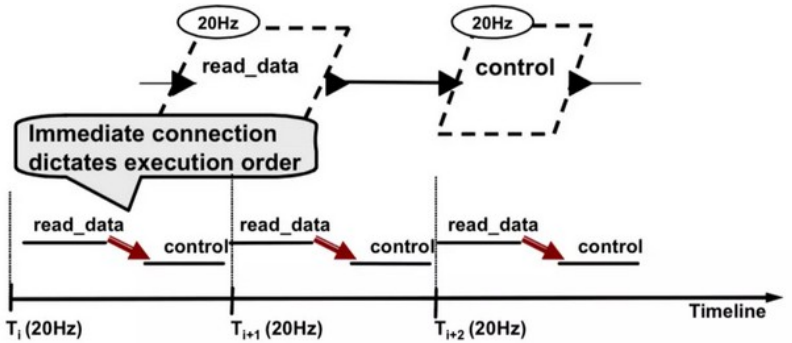
IoT architecture description

- Software Architecture examples

Data connection policies

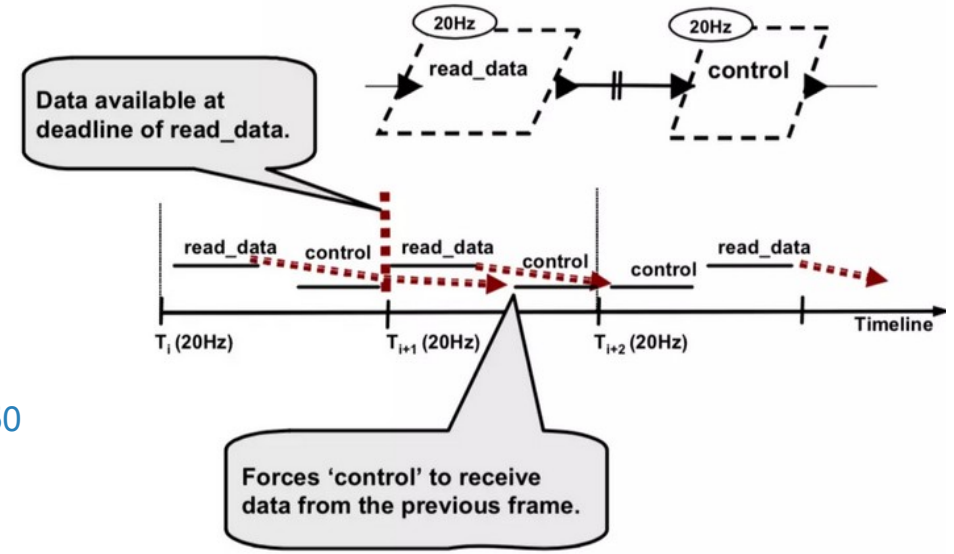
Immediate

receiver thread is immediately awaken, and will read data when emitter finished



Delayed

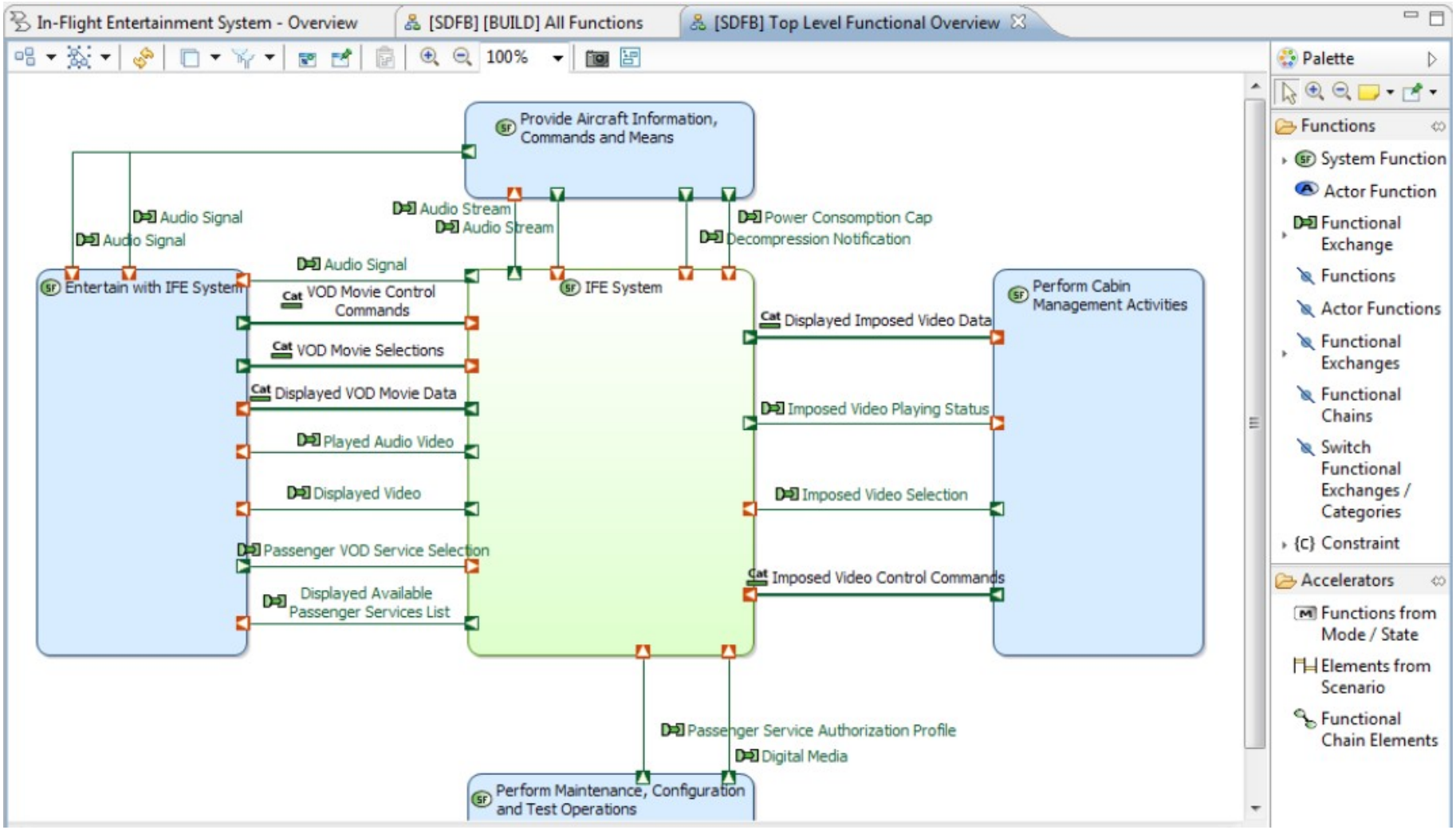
actual transmission is delayed to the next time frame



https://www.slideshare.net/iivanoo/aadl-42305750

IoT architecture description

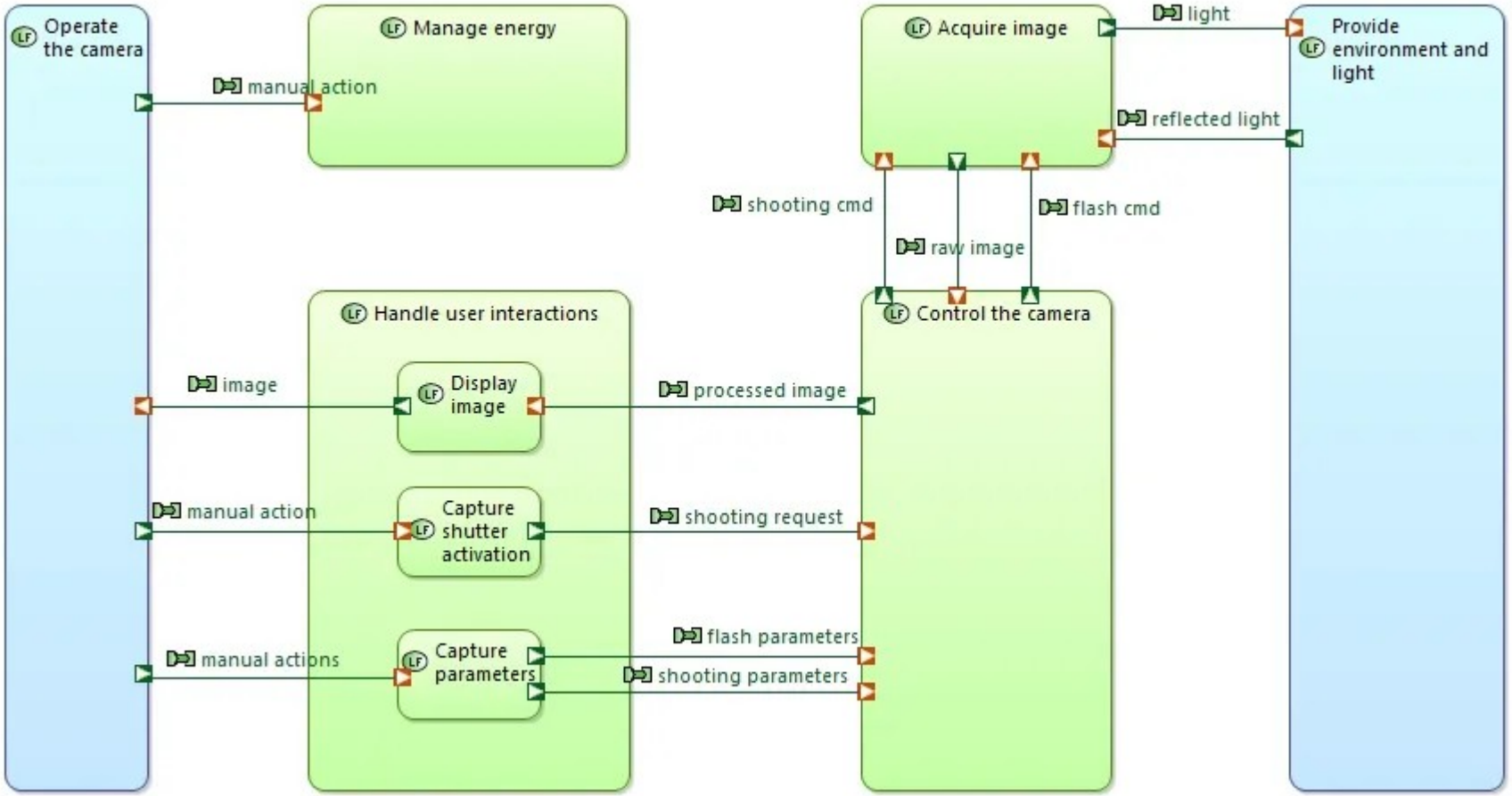
- Software Architecture examples



Par Christophe.gatti — Travail personnel, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=44334222

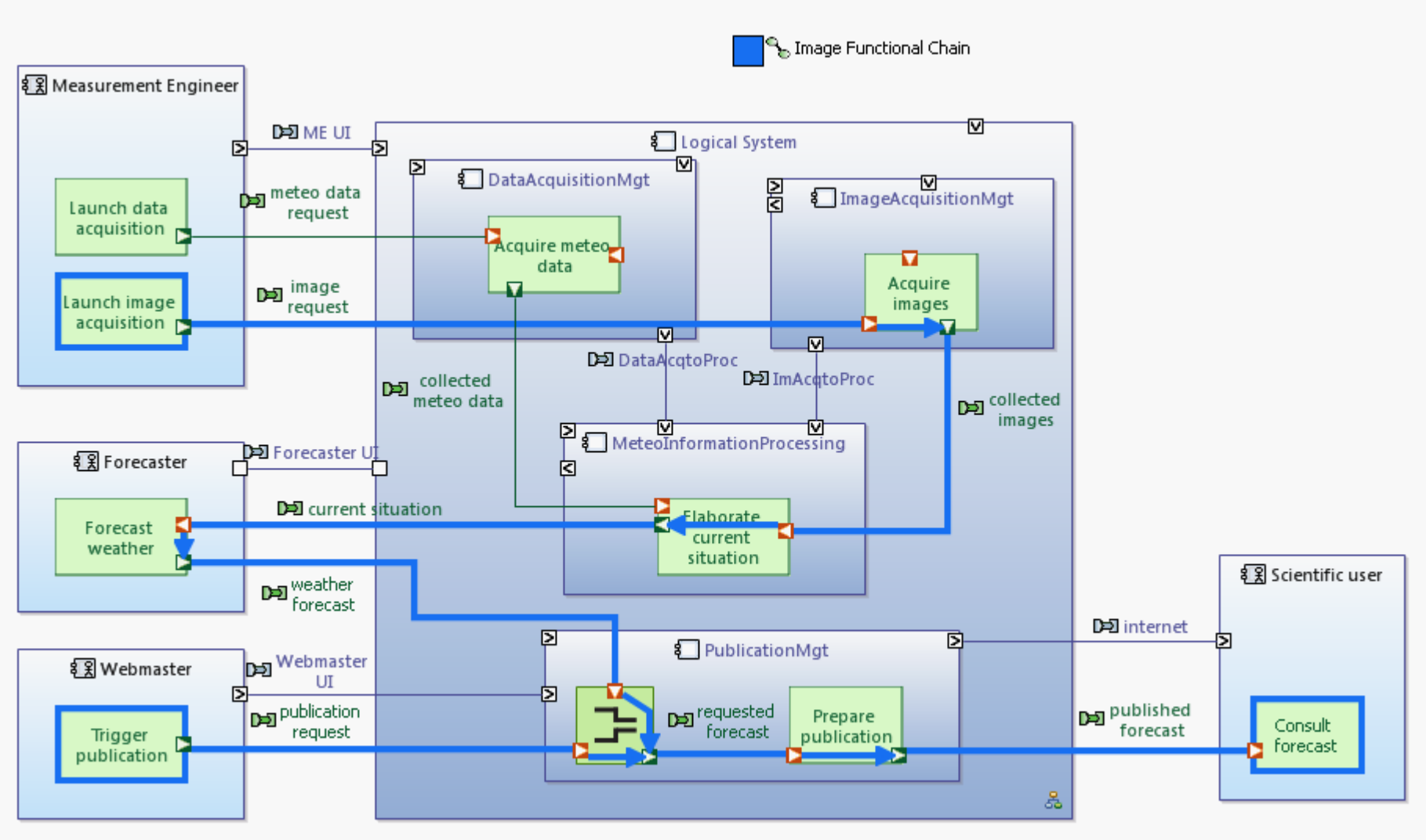
IoT architecture description

- Software Architecture examples



IoT architecture description

- Software Architecture examples

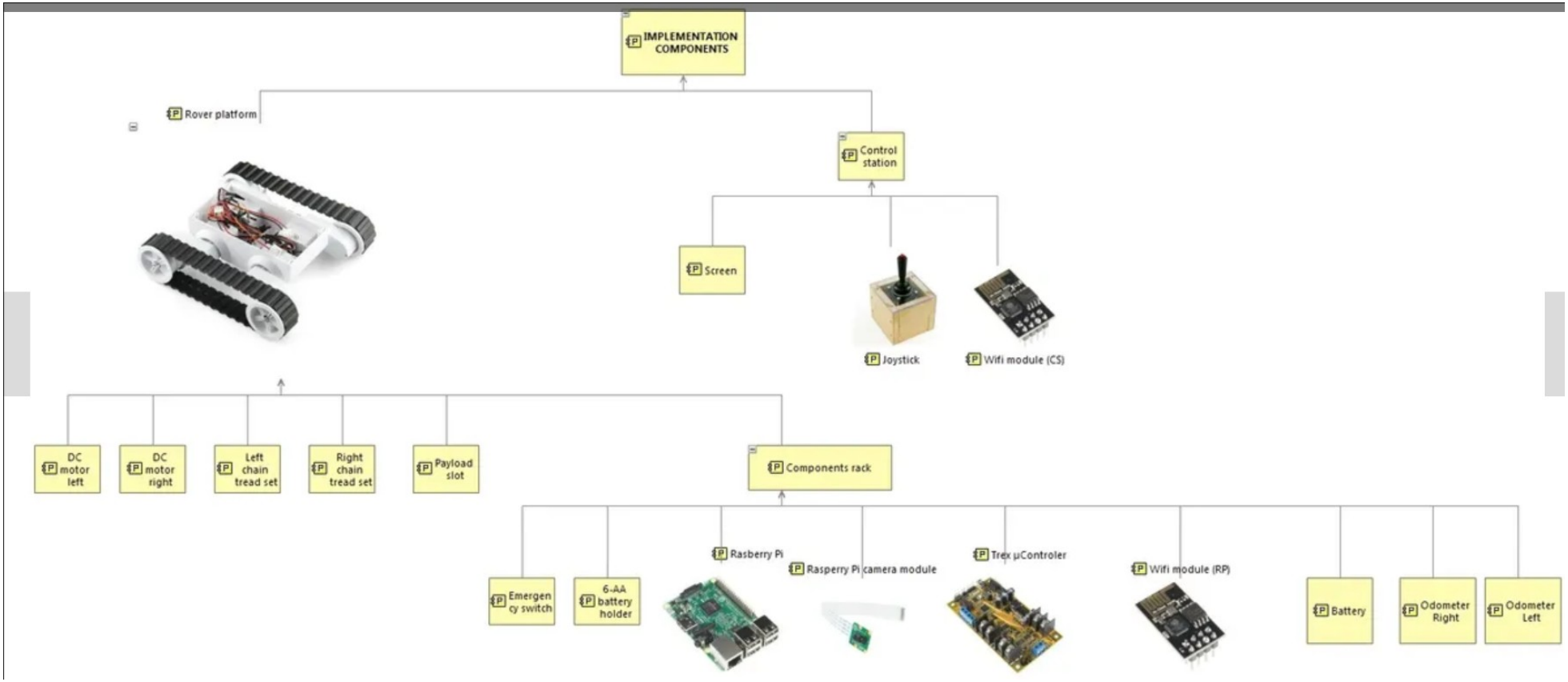


IoT architecture description

- Hardware/platform Architecture for IoT infrastructure
- The goal here is not at all to define what is part of the Field, the edge, the fog or the cloud as classically seen in the literature.
  - The goal is to specify the resources available and their inter connections ; so that you will be able to reason about a placement strategy (see next slides)
- The detail level depends on what you want to focus on.
  - It may be interesting to differenciate sensors and actuators devices (with none or few computational resource) from computation nodes ; since the later is able to execute your software while the former not necessarly.
  - It may be interesting to have an abstract view of the main properties of each node (on sector or battery ; water protected or not ; communication capabilities, etc)
- Simple is beautiful… But you should be able to take some decision based on the description of the hardware architecture.

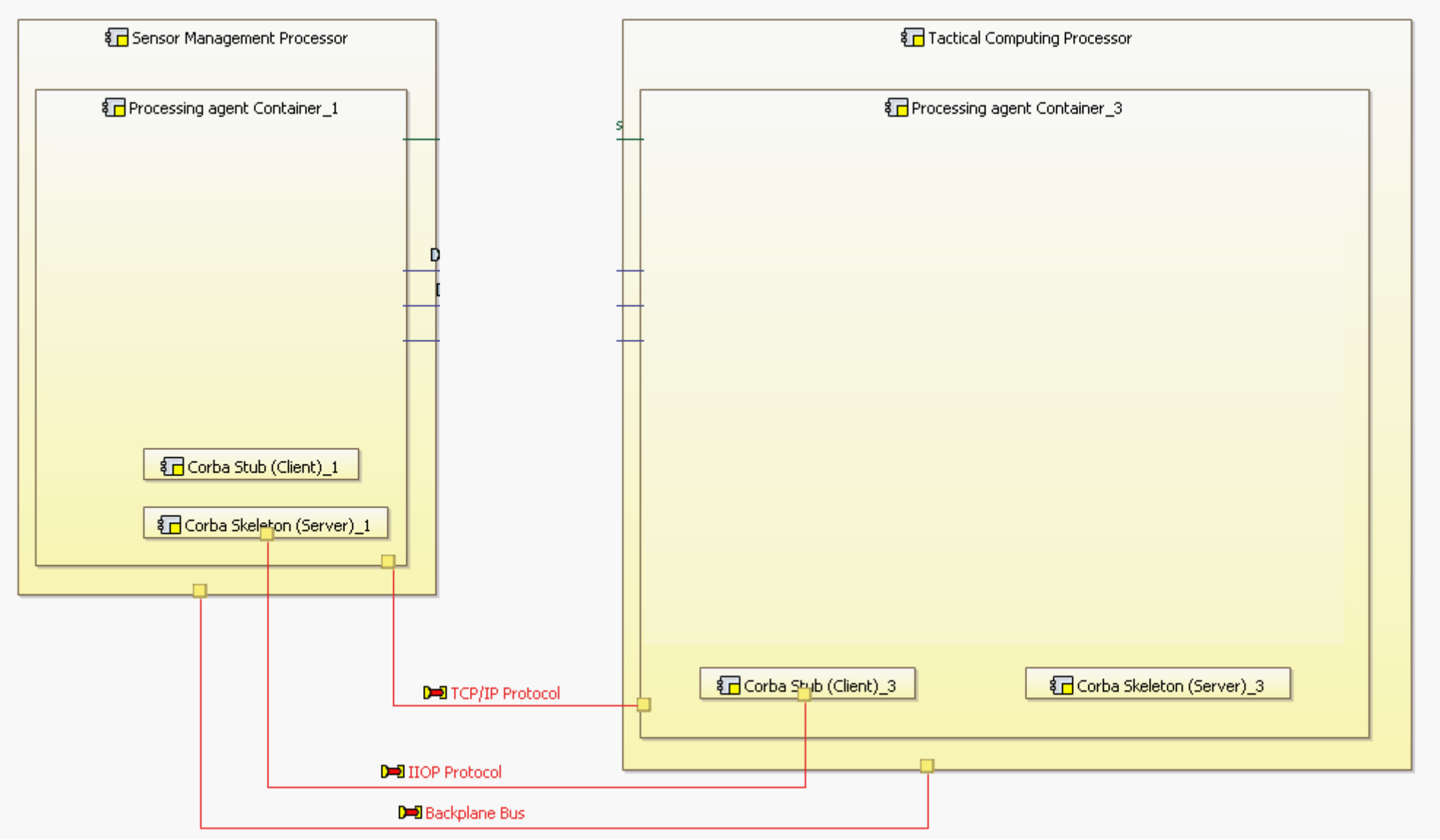
IoT architecture description

- Hardware/platform Architecture examples



IoT architecture description

- Hardware/platform Architecture examples

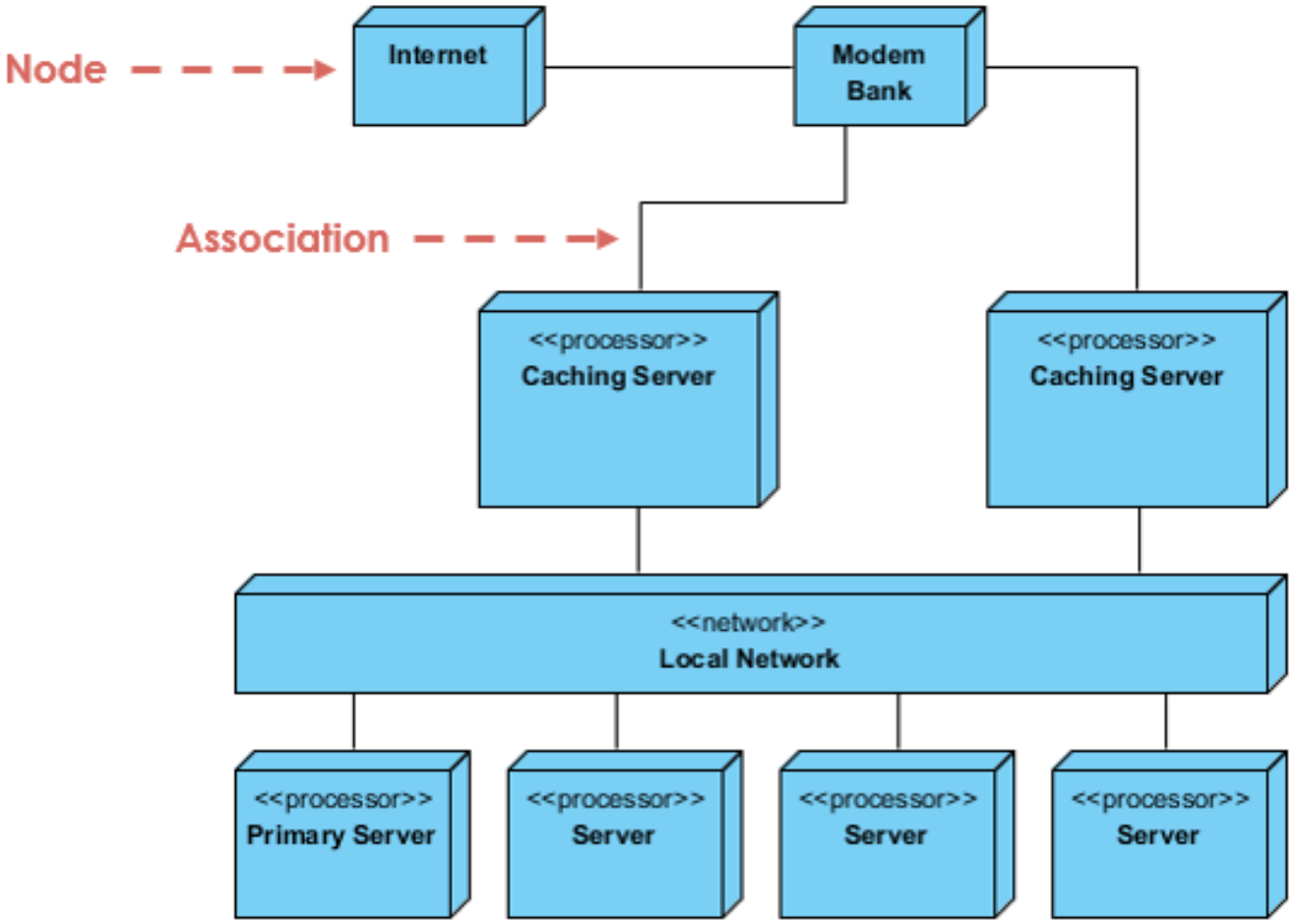


Par Christophe.gatti — Travail personnel, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=44334222

modified

IoT architecture description

- Hardware/platform Architecture examples



https://www.archimetric.com/what-is-deployment-diagram/

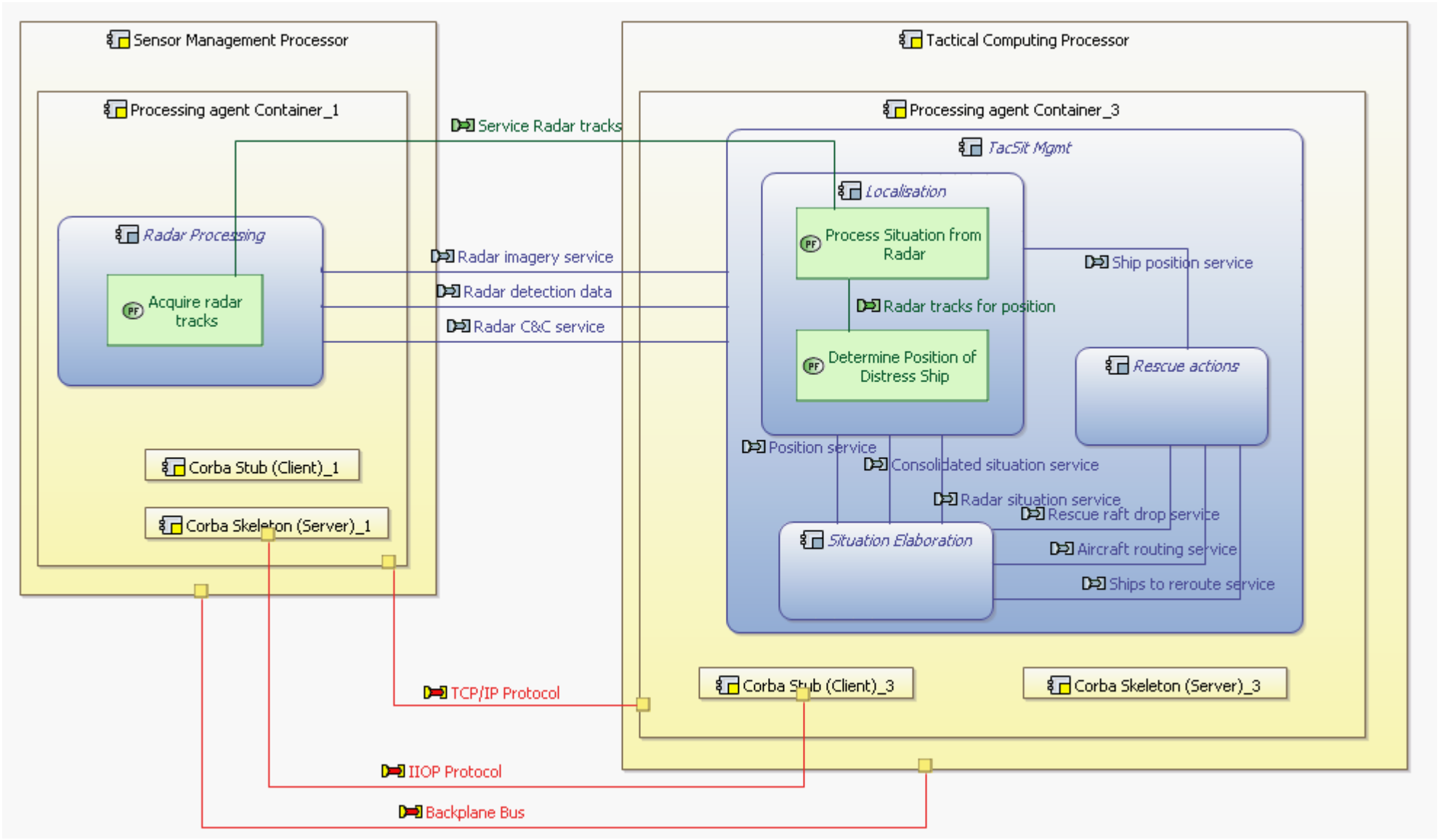
IoT architecture description

- ***Placement*** for IoT system
- Placement defines how the elements of the software architecture are actually mapped/hosted on the elements of the hardware architecture
  - This step is very important and can lead to very different characteristics of the system
  - There should be an adequation between the software elements and the hardware elements. It may be the case that one need to be *shaped* to fit the other…
  - Placement may go beyond stating who is executing on who. It may specify the software support used/required to execute the software (e.g., the required OS, the use of containers, the use of an hypervisor, the presence of a middleware, the requirement of a library, a software dependency, etc)

⇒ Architecture representations, and more importantly placement, does not require to be graphical. It could be a textual description, a tabular representation, etc. The goal is to understand as quick as possible potential problems in the IoT system

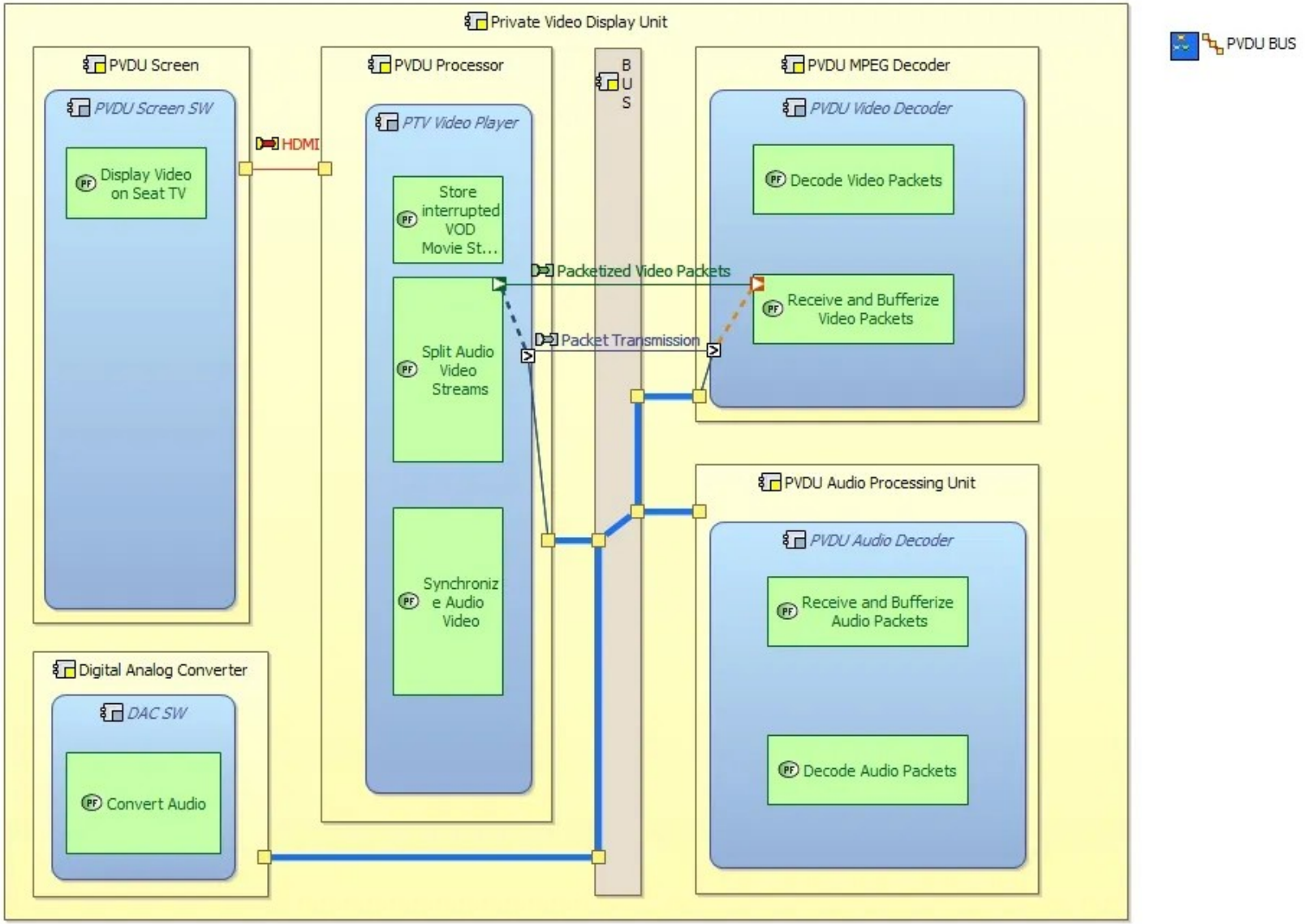
IoT architecture description

- ***Placement*** for IoT system, examples



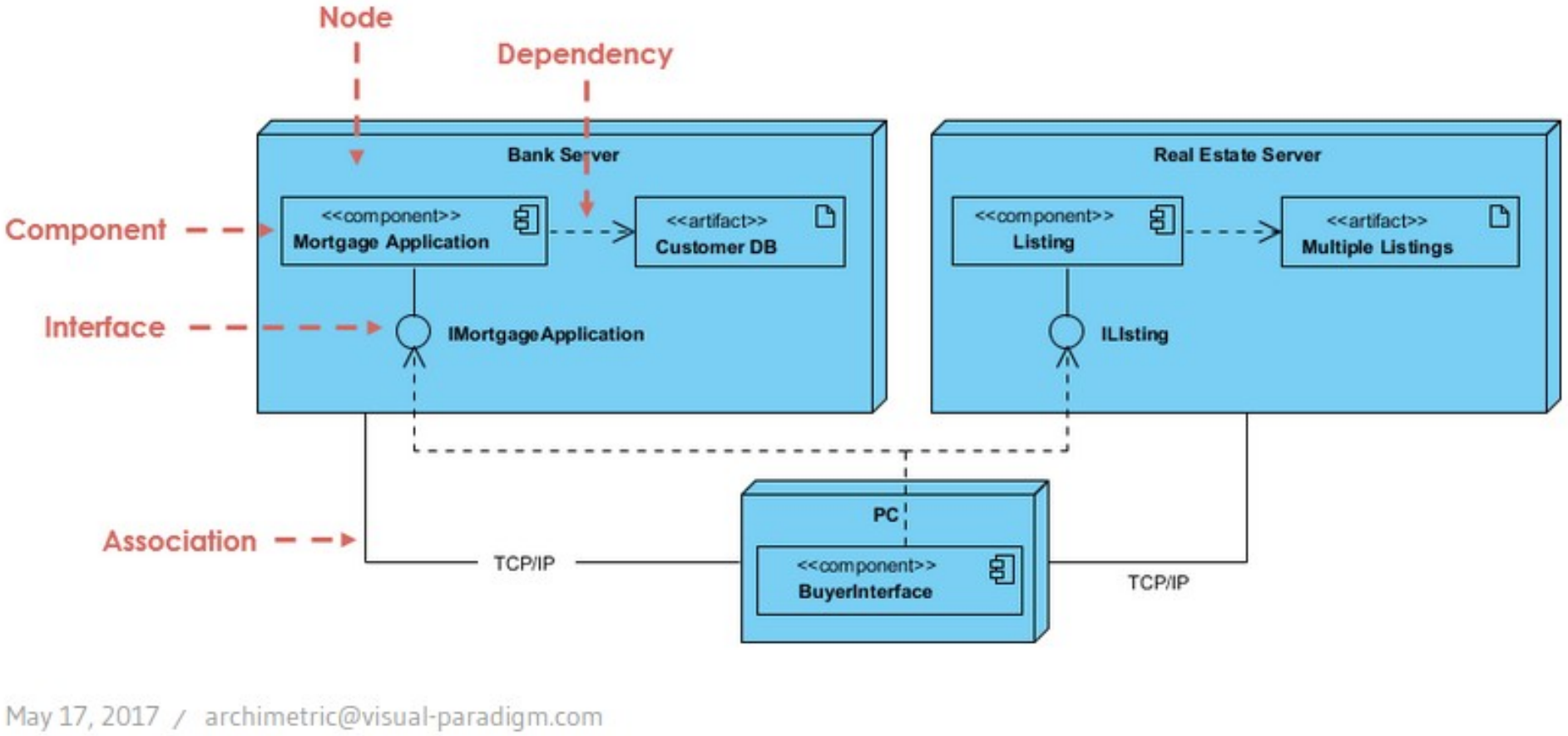
IoT architecture description

- ***Placement*** for IoT system, examples



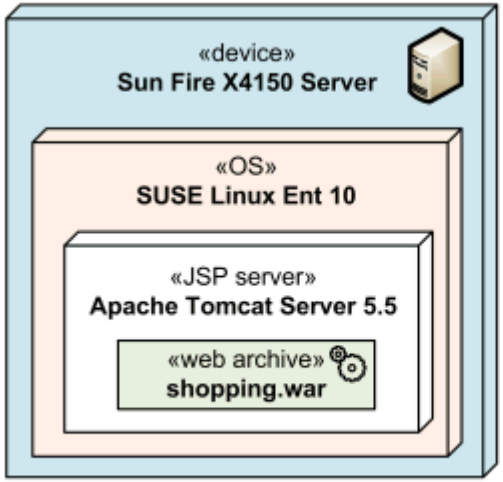
IoT architecture description

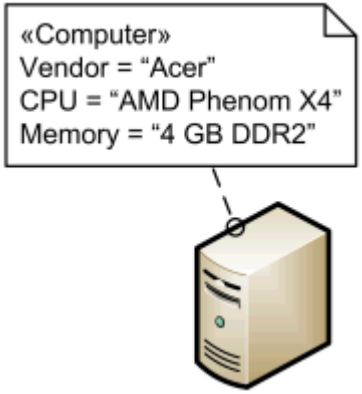
- ***Placement*** for IoT system, examples

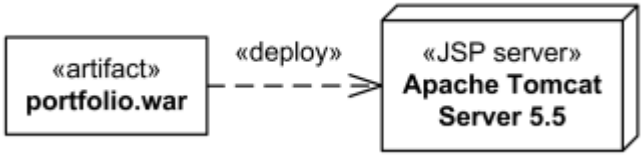


IoT architecture description

- ***Placement*** for IoT system, examples







Several execution environments nested into server device

https://www.uml-diagrams.org/deployment-diagrams.html

IoT architecture description

- ***Placement*** for IoT system, examples
  - placement tables

IoT architecture description

- ***Placement*** for IoT system, examples

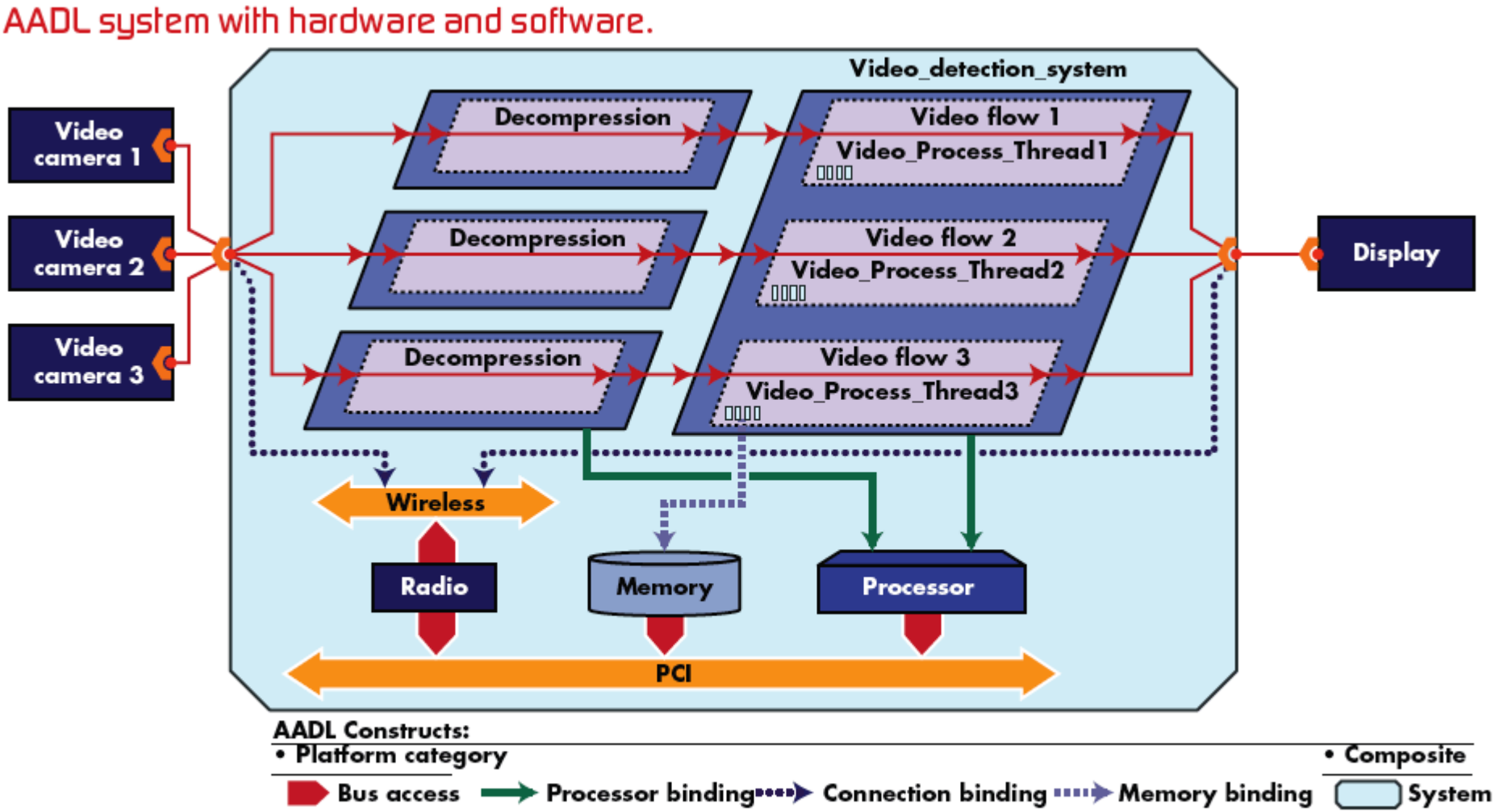


Figure 3

https://www.embedded.com/use-aadl-to-analyze-and-design-embedded-systems/

IoT architecture description

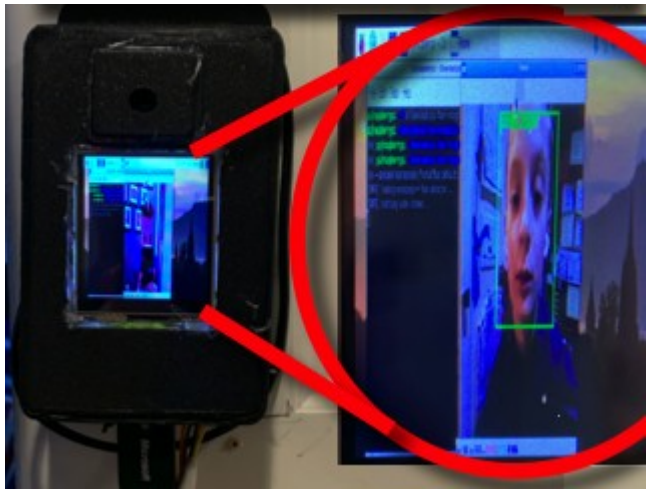
- Any architecture should be more than boxes and arrow for the expert of the domain (i.e., you). This is usually different from the one you put in commercial slides...
- Since you are free to choose any technology, library, framework for the project, we will not do any technical assistance !
- Instead, we will evaluate your architecture and how it evolves through time and understanding of extra functional considerations.
- There are plethora of solutions for the project but one that fits is one for which you are aware of its own threats and mitigation; and for which you can rationalize the choices.

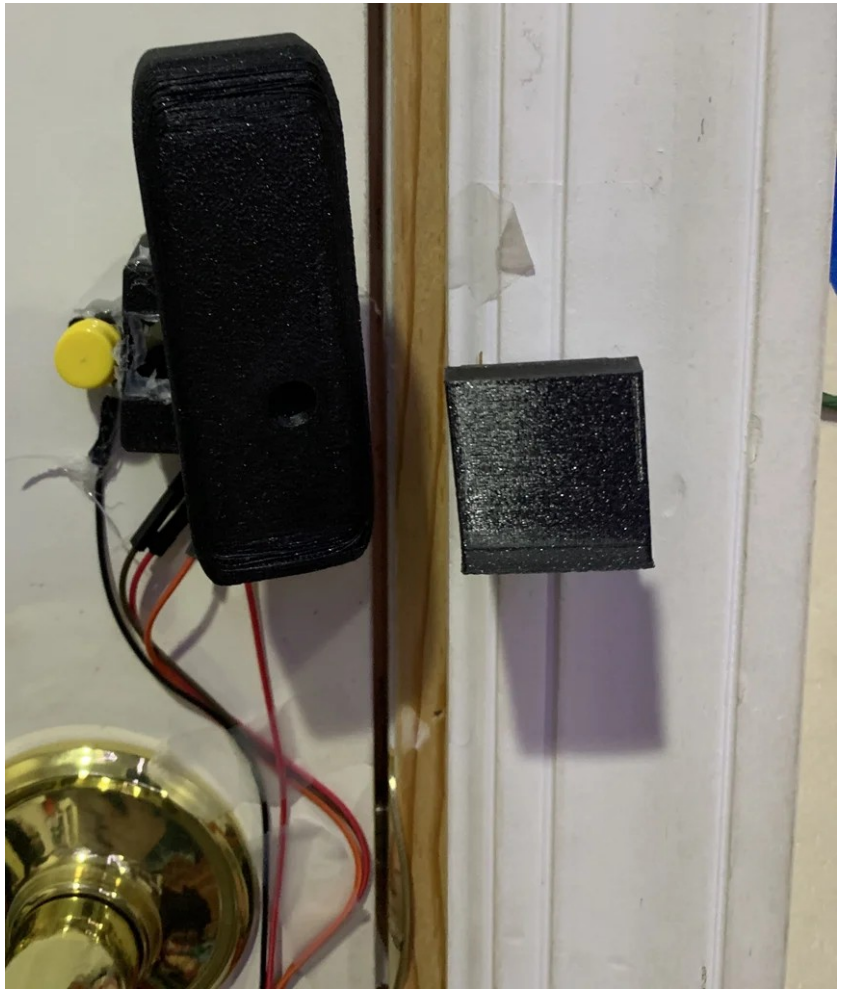
Project description

- We want to propose an electronic lock system where the nominal behavior is that the users are identified by both a facial recognition and by a RFID badge. When both identification methods are successful, the entrance is unlock. It should be possible to register faces and badges
- You can add features.
- You will add requirements.
- You have to respect the required time to market (23rd of January)

Project description

- We want to propose an electronic lock system where the nominal behavior is that the users are identified by both a facial recognition and by a RFID badge. When both identification methods are successful, the entrance is unlock. It should be possible to register faces and badges





UI design and ergonomics are not taken into account

Project description

- Available hardware
  - Raspberry PI 3
  - Arduino boards with a shield with classical sensors/actuators (leds, buttons, temperature sensors, ...)
  - raspberry hat for arduino sensor usage
  - a pi camera module.
  - a SLC3711 USB RFID reader
  - a USB ampere-meter (to be shared among teams)
  - some radio frequency emitter/receivers
  - a maximum of two laptops in your teams to be used as you feel is appropriate
- “Extra" hardware:
  - You can mock extra hardware that you main need by using simple Sensors/Actuators from Arduino.

Project description

- You will work in group of 4 (one of 5)
- Choose your team yourself
- You can share knowledge among teams

  ⇒ the goal is to obtain the best architecture**S**
- Do not redo everything from scratch :
  - https://www.tomshardware.com/how-to/raspberry-pi-facial-recognition
  - Nfc python module ?
  - …