



POLITECNICO
DI TORINO



e-Lite

Designing the System Architecture

Aml Design Process

Fulvio Corno

Politecnico di Torino, 2017/2018



System architecture

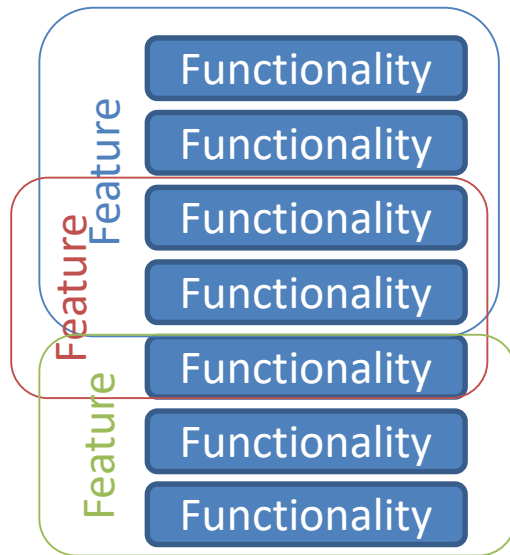
- An architecture description is a formal **description** and representation of a **system**, organized in a way that supports reasoning about the **structures** and **behaviors** of the system.
- The fundamental organization of a system, embodied in its **components**, their **relationships** to each other and to the environment, and the principles governing its design and evolution.

https://en.wikipedia.org/wiki/Systems_architecture and linked sources

System architecture

- A representation of a system, including a mapping of **functionality** onto **hardware** and **software components**, a mapping of the **software architecture** onto the **hardware architecture**, and **human interaction** with these components. [CMU SEI]

A mapping...



Functional design

Feature: a user-visible, user-callable, well-defined, self-contained system behavior, that can be understood and brings value to the user.

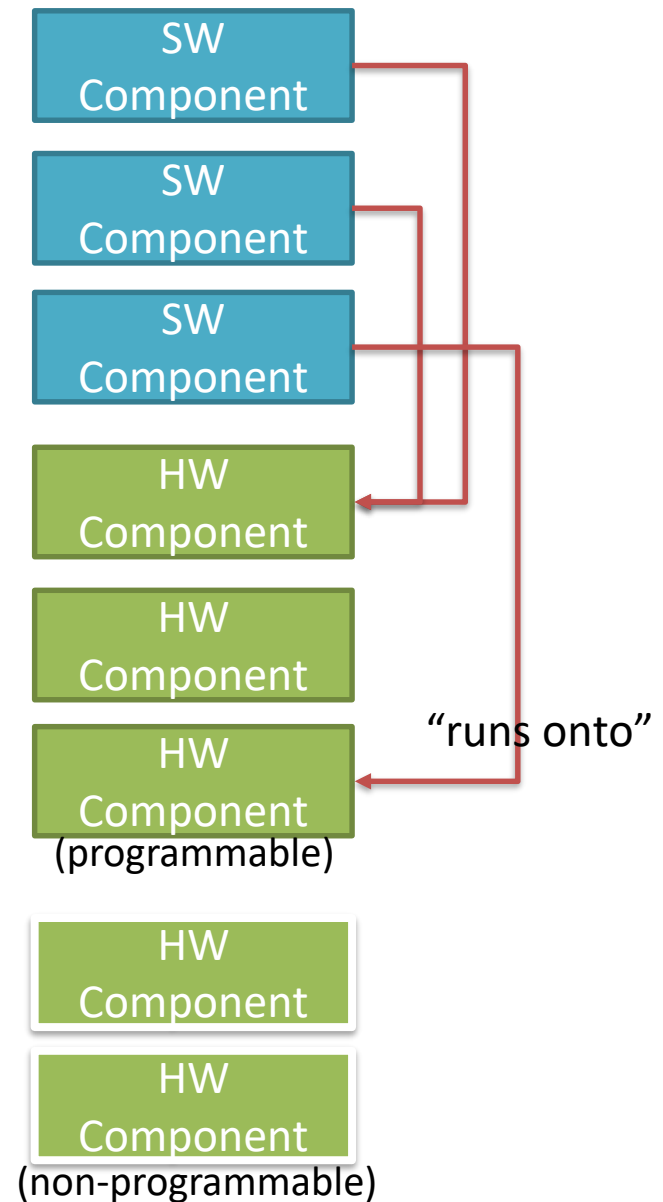
Functionality: functions, actions, reactions, computations, ... implemented by the system.

A mapping...

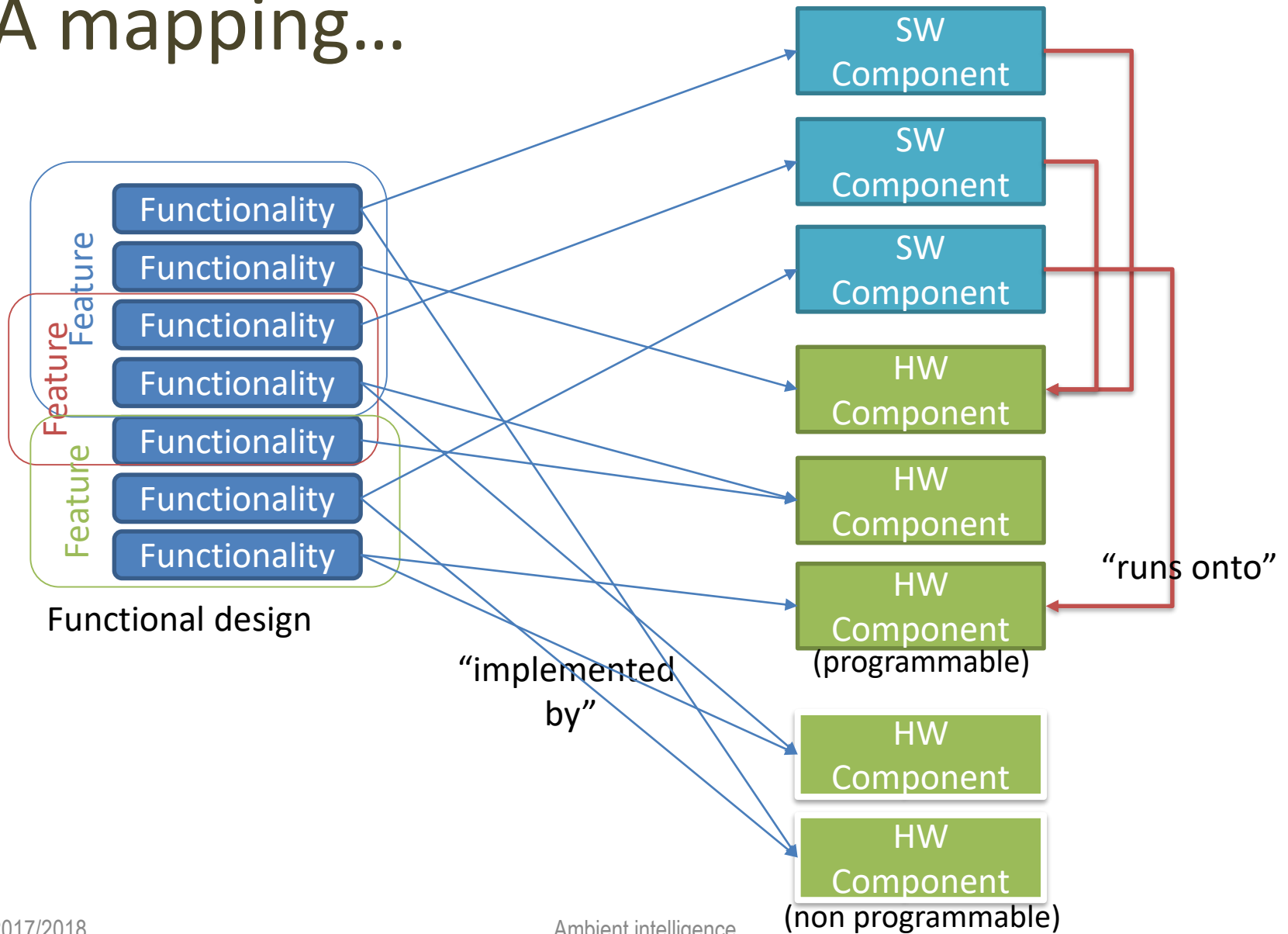
SW component: a software-implemented library, function, module, API, computation, ...

Programmable HW component: computer, server, raspberry, microcontroller, smartphone, tablet, ...

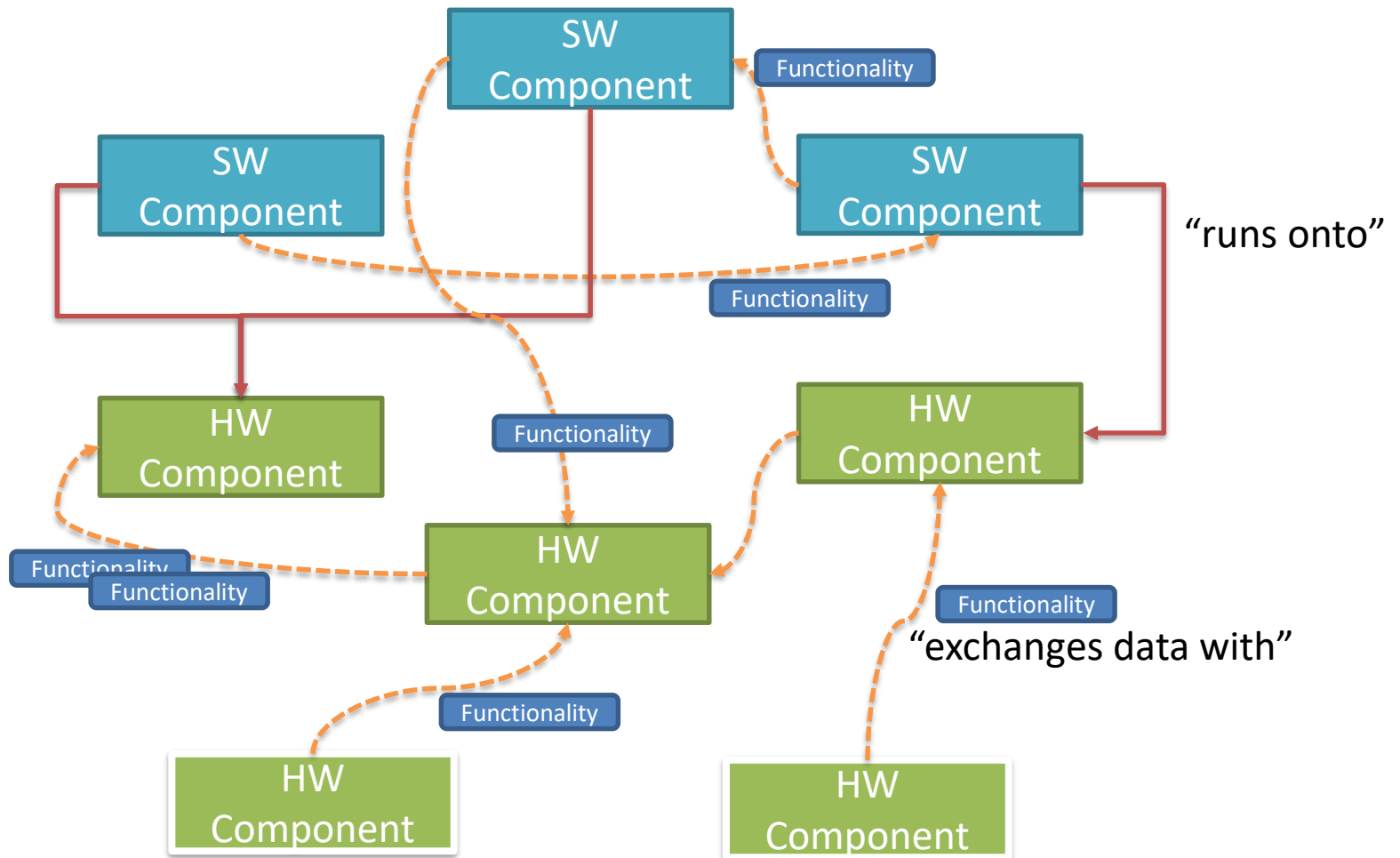
Non-programmable HW component: sensor, actuator, embedded device, smart home device, display, ...



A mapping...



A mapping...



Preliminary analysis

- What are my system features?
- What are the functionalities needed to implement such features?
- What are the **alternatives** for implementing a given functionality?
 - Software (running on some programmable hardware)
 - Hardware (non-programmable)
 - A combination of the two?
 - **Always explore several alternatives!**

Exploring alternatives

- Always start from the user-visible features
- E.g., How to recognize and object?
 - Leaves space for different way of implementing it
- Not: How to sense a tag placed on the object
 - Already chose a technical solution
 - Cannot explore alternative ways of reaching the same goal
- Always seek for the **simplest** and **safest** way

Programmable Hardware (computing nodes)

- How many?
 - The less, the better
 - Except if needed...
- How powerful?
 - Is a Raspberry PI sufficient? Need a PC? Better an Arduino?
 - Consider computing power but also I/O capabilities
- Located where?
 - Needs to communicate with non-programmable hardware? → Local
 - Needs to be accessible from outside the house? → Server
- How connected?
 - Ethernet? Wi-Fi? Bluetooth? 3G/4G? Serial? ...
- Scalability
 - If I have more users or more locations, which components need to be replicated and which may be shared?

Non-programmable hardware (devices)

- Which ones?
- Where/how are they connected?
 - A computing node?
 - A microcontroller?
 - A central hub/controller?
 - Wired/wireless?
- Who reads/controls them?
 - What software module?
- How many protocols do we need?

Examples

- Distance sensor –(mounted on)->Arduino –(serial)-> RaspberryPI -> program for polling data
- Z-Wave multisensor –(z-wave)-> RaZberry –(mounted on)-> Raspberry PI -> Z-Way APIs –(Wi-Fi)-> Smartphone -> read and display data
- Oven –(plugged)-> Smart Plug –(z-wave)-> ...

Software

- Functionality?
- Where does it run? (among the computing nodes)
- Where does it store its data?
 - Is it local? Does it need to go through a remote API?
 - Consider scalability in defining data location storage
- Which other software modules does it need to exchange data with?
 - Are they on the same computing node? (use local APIs, shared database, sockets, ...)
 - Are they on different computing nodes?

Defining the Architecture

- System Architecture
 - Overall view
 - HW+SW
 - Major interconnections
 - The user(s)
- Hardware Architecture
 - Computing nodes
 - Devices
- Software Architecture
 - Applications (clients and servers)
- Network Architecture
 - Who connects to whom?
 - What protocol(s)?

License



- These slides are distributed under a Creative Commons license “**Attribution – NonCommercial – ShareAlike (CC BY-NC-SA) 3.0**”
- **You are free to:**
 - **Share** — copy and redistribute the material in any medium or format
 - **Adapt** — remix, transform, and build upon the material
 - The licensor cannot revoke these freedoms as long as you follow the license terms.
- **Under the following terms:**
 - **Attribution** — You must give [appropriate credit](#), provide a link to the license, and [indicate if changes were made](#). You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
 - **NonCommercial** — You may not use the material for [commercial purposes](#).
 - **ShareAlike** — If you remix, transform, or build upon the material, you must distribute your contributions under the [same license](#) as the original.
 - **No additional restrictions** — You may not apply legal terms or [technological measures](#) that legally restrict others from doing anything the license permits.
- <http://creativecommons.org/licenses/by-nc-sa/3.0/>

