



Digitalization
Industrie 4.0

Smart Production
E-Mobility
Smart Energy

Energy Efficiency
Smart Infrastructure

Smart Buildings
Renewables

Bert van der Linden | 2020

Open OT architectures have a great future!



Open OT architectures have a great future!

Goals and Target group

- The participant understands the current trends – like component-based development, ecosystems and virtualisation – in OT systems.
- The participant understands the relation between open OT architectures and frameworks.

- Chief Digital officers (CDO), Chief Technical officers (CTO), IoT Engineers, Technical consultants, PLC programmers

Approach

- Several domains – discrete production automation and process automation – develops new architectures – like Open Process Automation (OPA), Namur Open Architecture (NOA), (Industrial) IoT, Industry 4.0 - and what can we learn of these new architectures.
- A more conceptual approach.

- Two views of openness:
 - Architectural
 - Component

Open OT architectures have a great future!

Agenda

- Current OT architectures
- Open software architectures
- Virtualisation



Digitalization
Industrie 4.0

Smart Production
E-Mobility
Smart Energy

Energy Efficiency
Smart Infrastructure

Smart Buildings
Renewables

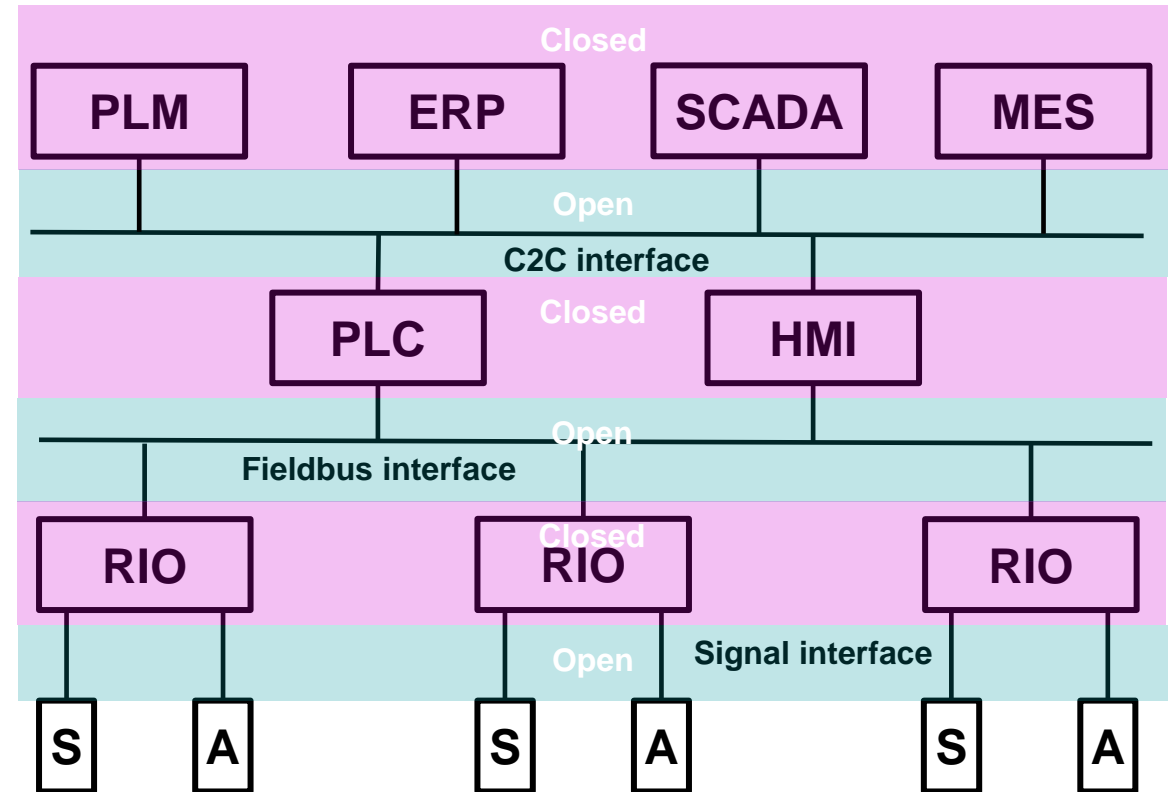
Bert van der Linden | 2020

Current OT architectures

Open OT architectures have a great future!

Programmable Logic Controller: Hardware architecture

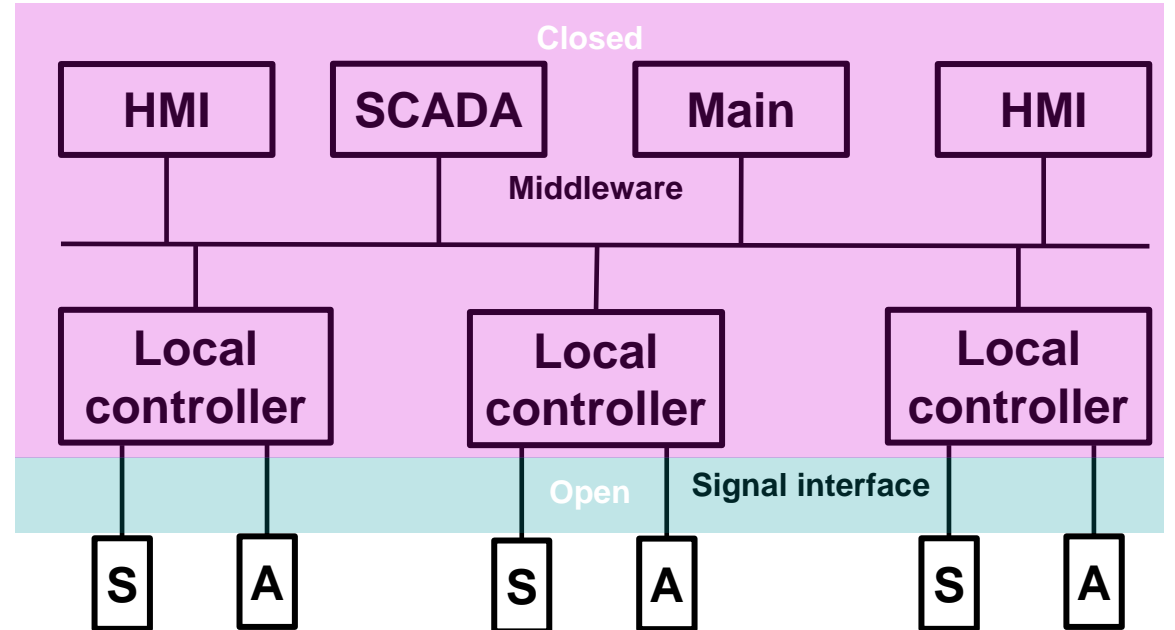
- A PLC is a **special purpose** computer and therefore usually a **closed** system.
- A PLC features **special IO** relative to a general purpose computer (PC).
- The local system buses are **closed** systems.
- The **open** standard signal interfaces (4-20mA, 0-10V, IO-link, HART) and **open** standard fieldbuses (PROFINET, Ethernet/IP) makes a PLC hardware architecture **open**.
- OPC UA makes the C2C interface **open**.
- SCADA is based on **COTS** framework /applications.



Open OT architectures have a great future!

Distributed Control Systems (DCS)

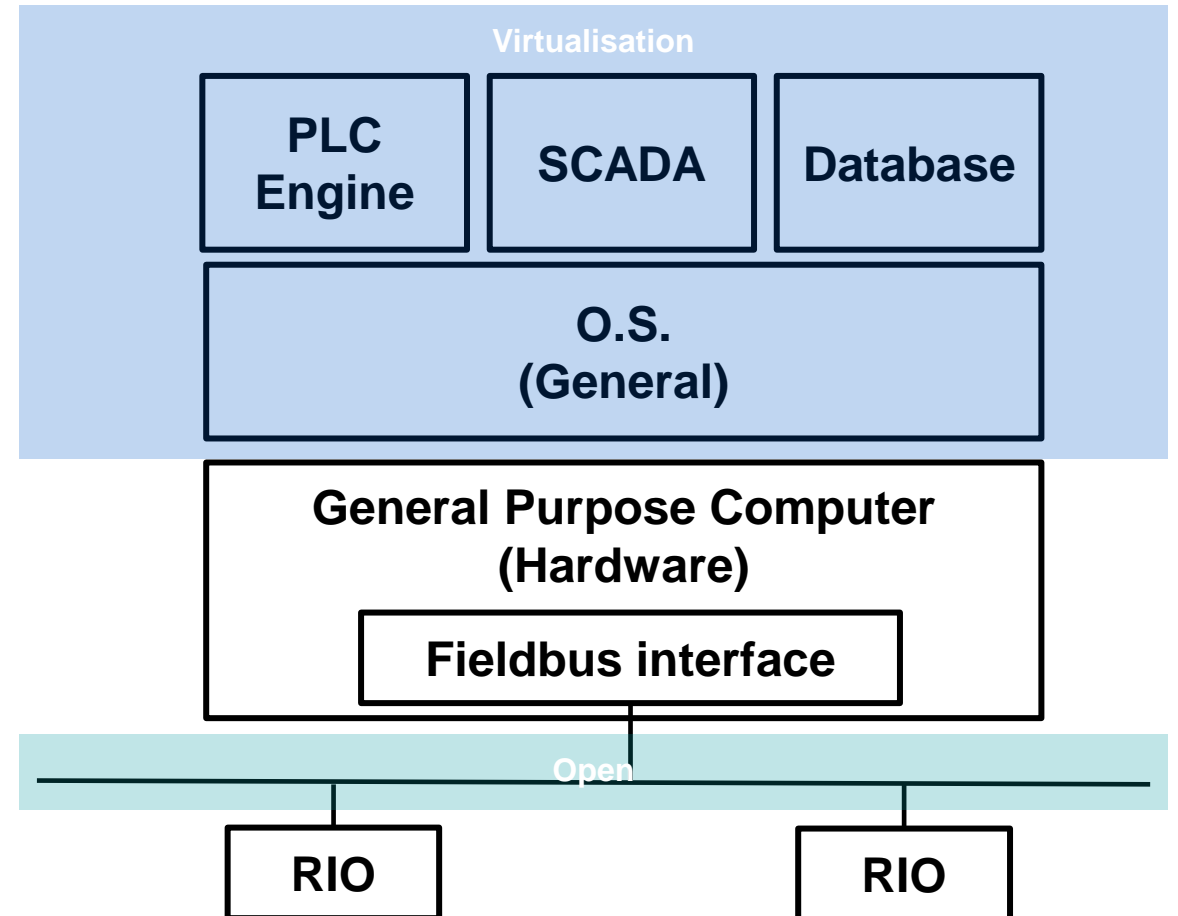
- DCS uses a ***distributed*** architecture.
- The computing power is distributed over several ***smaller*** local controllers.
- A DCS has the characteristics:
 - Higher scalability,
 - Better availability,
 - ..
- DCS uses ***fewer open*** standards for the interface between the local controllers (middleware).



Open OT architectures have a great future!

PC-based control

- PC-based control uses a **general-purpose** computer.
 - SoftPLC;
 - Real-time Kernel-based PLC.
- PC-based control can also **use** the IEC61131-3 standard.
- PC-based control uses the **GUI power** of a general purpose os, like Windows or Linux.
- PC-based control take advantage of the developments in the computer industry, for example virtualisation.
- PC-based control improves the **consolidation**.



Open OT architectures have a great future!

Open architecture : Criteria

- Components:
 - Controllers
 - Remote IO stations
 - Frequency drives
- Key interfaces:
 - Signal interfaces
 - Open fieldbus systems
 - C2C interfaces

What are open architectures?

Architecture : Criteria	Status
Modular design based on modules (components).	✓
Key interfaces that are based on wide accepted standards (high installed base).	✓
Key interfaces that are small and explicit .	✓

https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/66098/sosa_open_sys_strategy_nov12.pdf

Open OT architectures have a great future!

Components : Criteria

- An engineer can compose applications of third-party components because the engineering-software accepts configuration files of other parties.
- The engineer does not need to modify the components, but only needs to configure the components.



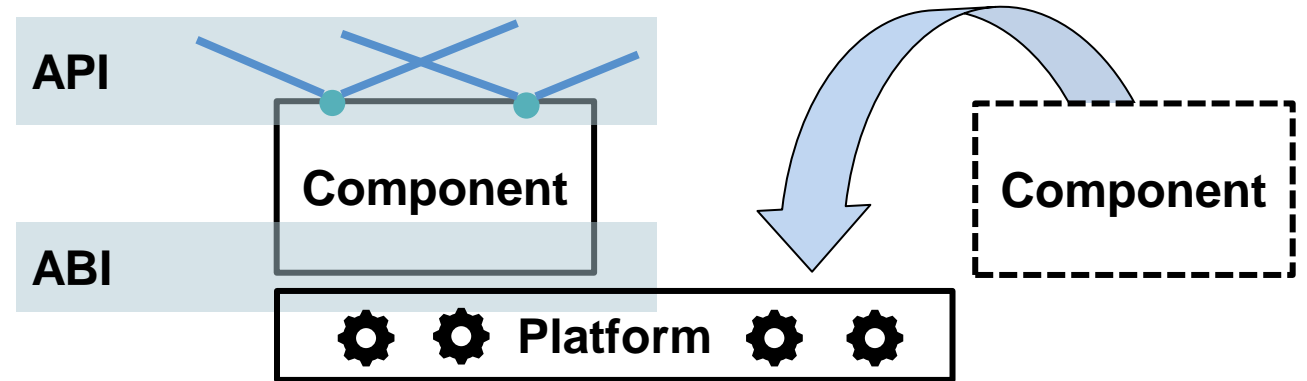
What about components?

Components : Criteria	Status
A unit of <i>independent deployment</i> and with out modification (additional work).	✓
A unit of <i>third-party</i> composition.	✓
A unit with a general context.	✓

Open OT architectures have a great future!

Current challenges : Software components

- Technology refreshments/upgrades.
 - New protocol,
 - New service (for example FTP).
- Vendor lock-ins.



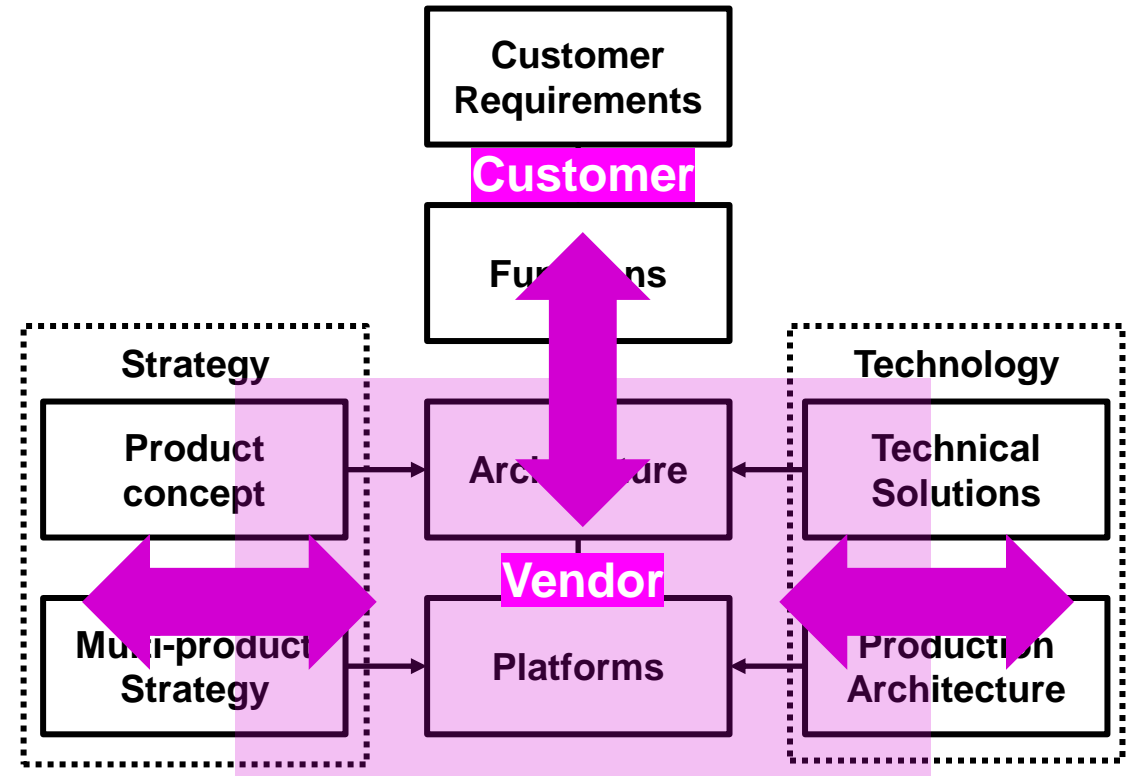
Components : Criteria	Status
A unit of independent deployment and with out modification (additional work).	✗
A unit of third-party composition.	✗
A unit with a general context.	✓

OT software architectures are not open!

Open OT architectures have a great future!

What can openness mean for you?

- Improve interoperability.
- Improve integration.
- Use of general (commercial) Off-The-Shelf (COTS) products and preferably open source.
- Openness increases *innovation*, competition stimulates *closed* systems.
- Technology *refreshment* vs product refreshment (vendor lock-In, maintainability).



Muffatto, Moreno & Roveda, Marco. (2002). Product architecture and Platforms.



Digitalization
Industrie 4.0

Smart Production
E-Mobility
Smart Energy

Energy Efficiency
Smart Infrastructure

Smart Buildings
Renewables

Bert van der Linden | 2020

Open software architectures

Open OT architectures have a great future!

The challenge of component-based development



- Variability
- Extensibility
- Adaptation

<https://cdn.instructables.com/FR8/2549/GE7LBTJX/FR82549GE7LBTJX.LARGE.jpg>

Solutions : **frameworks** and **patterns**

but

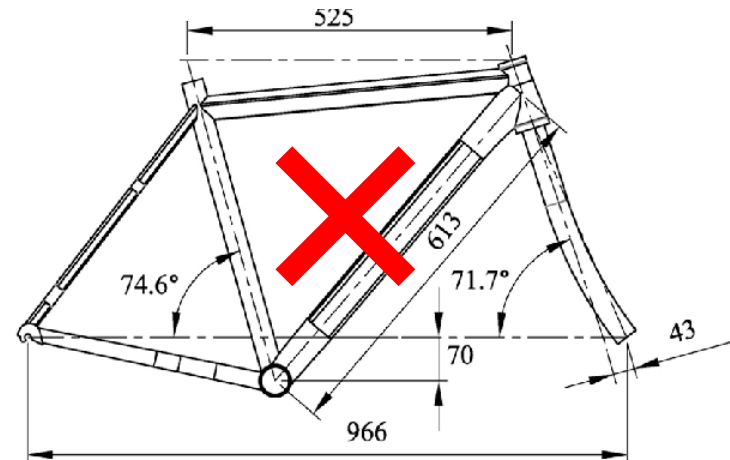
How **soft** is software?

<https://hollandbikeshop.com/img/prod/burley-kazoo-aanhangfiets-blauw-134597-0-1.jpg>

Open OT architectures have a great future!

General introduction extensibility : Components and Frameworks

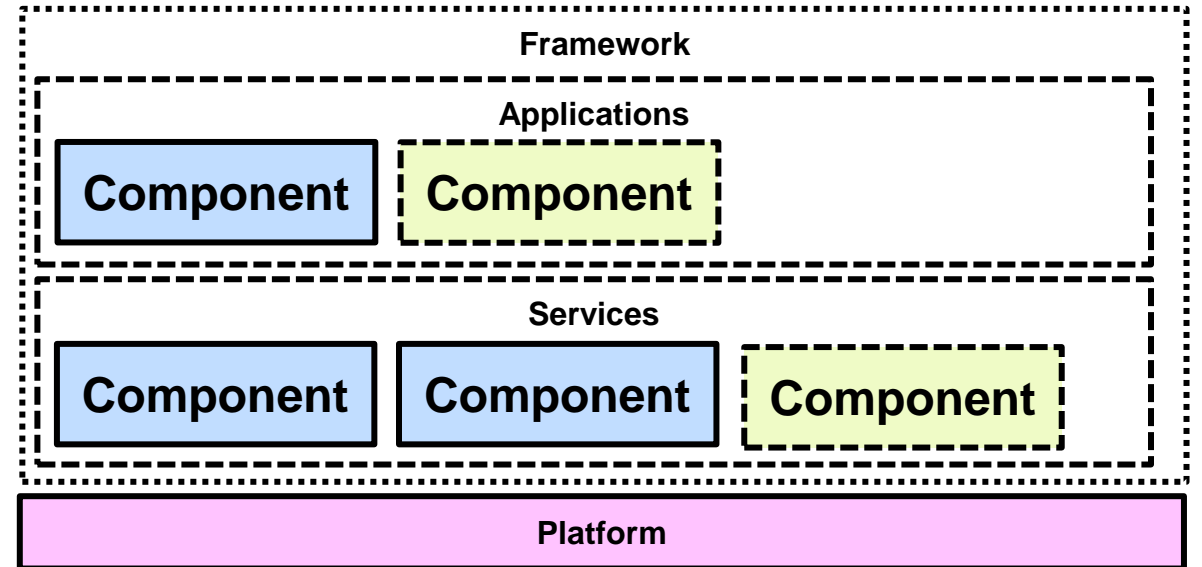
- An **incomplete** application for a family of components.
 - A set of **mounting points** for a compatible subcomponents.
 - A general decomposition of a **problem**.
- A frame(work) extension is any component added after frame(work) **deployment**. (added to a concrete frame).
 - Extensibility is about the addition of **new/modify functionality** to the frame(work).



Open OT architectures have a great future!

Frameworks

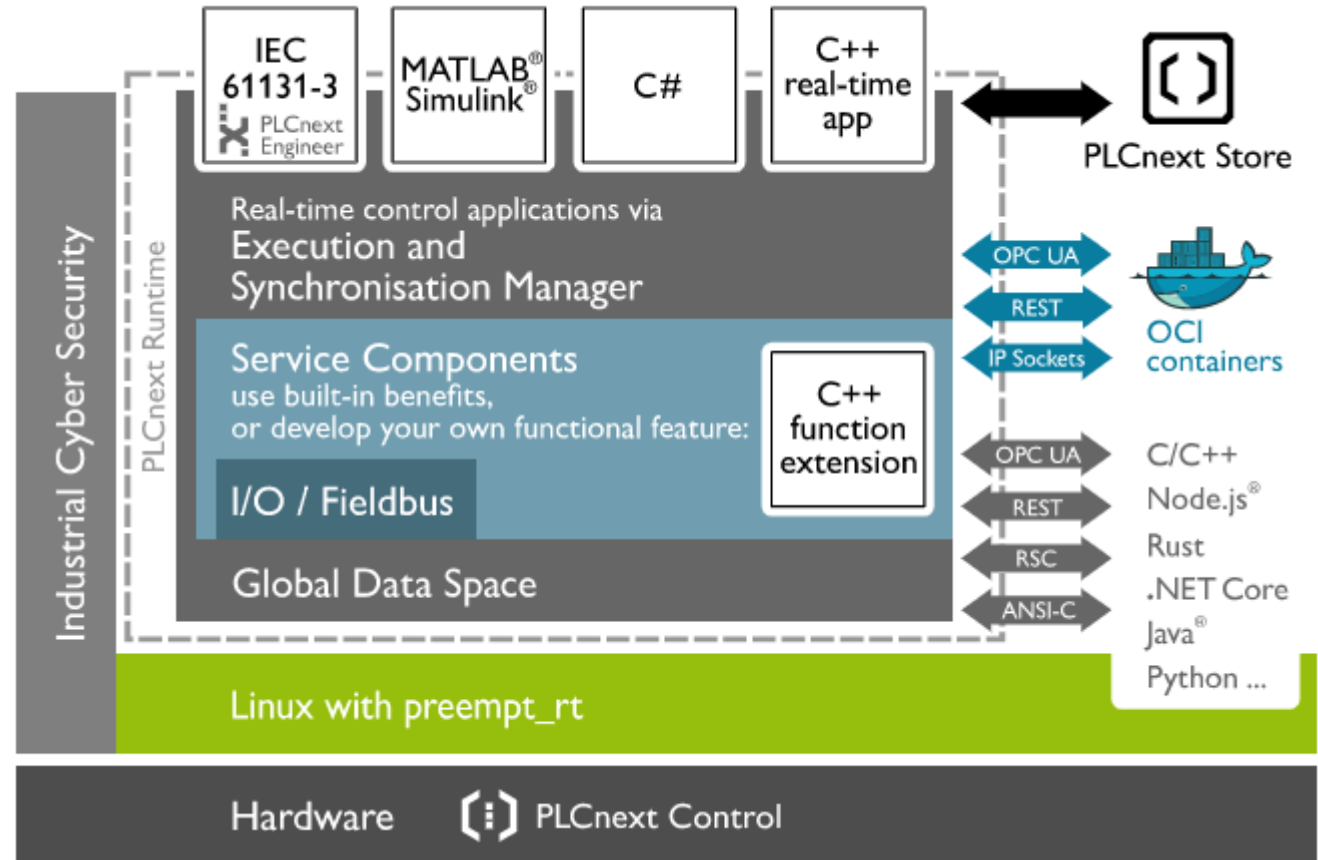
- A framework is a **configurable** and **extensible architecture** designed for multiple uses within an category of applications (Messerschmitt, 2003).
- Custom development or Commercial Off-The-Self (**COTS**)?
- Examples IT frameworks:
 - Dot.Net
 - Cocoa



Component : criteria	Status
A unit of independent deployment and with out modification.	✓
A unit of third-party composition.	✓
A unit with a general context.	✓

Open OT architectures have a great future!

The open OT architecture : PLCnext Technology

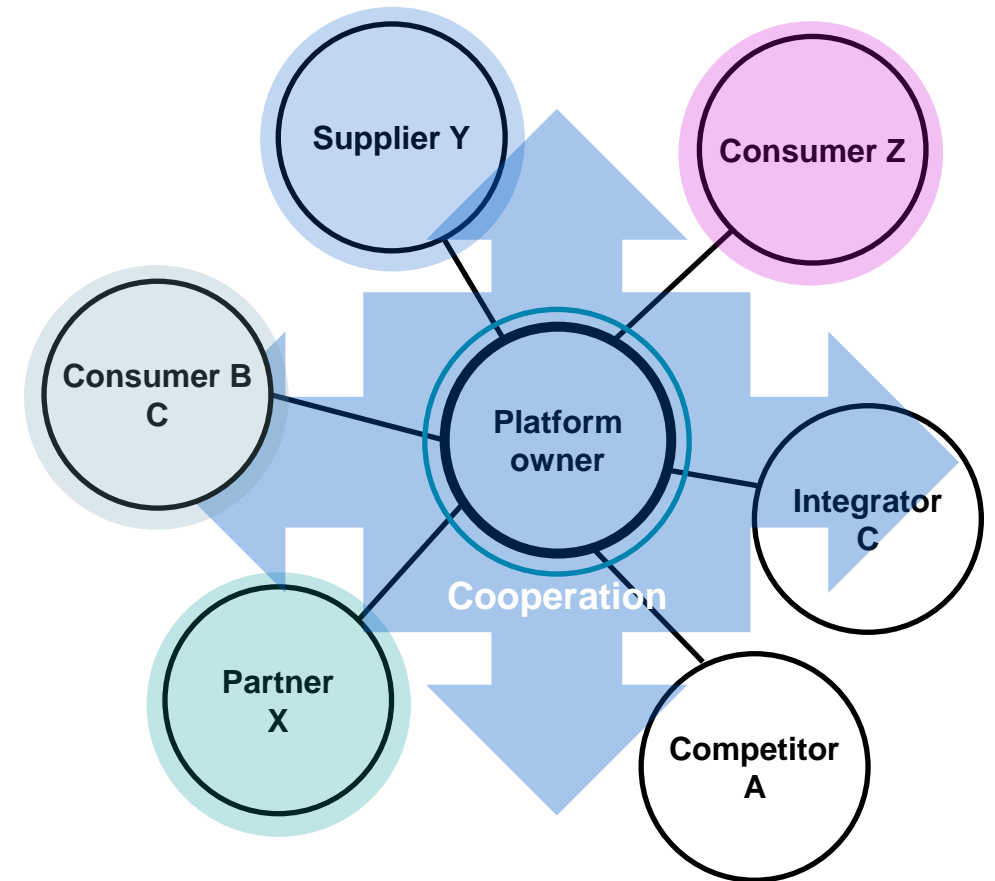


http://plcnext-infocenter.s3-website.eu-central-1.amazonaws.com/PLCnext_Technology_InfoCenter/PLCnext_Technology_InfoCenter/Home.htm?d_f=false

Open OT architectures have a great future!

Platforms and frameworks enables ecosystems!

- Two different software development paradigms:
 - **Isolated/independent** development in a company.
 - **Cooperation** in an ecosystem via platforms.
- Benefits ecosystems:
 - Decreases **costs** involved in software development and distribution.
 - Supports **cooperation** and knowledge sharing among multiple and independent software vendors
 - Enables better analysis of requirements and communication among stakeholders
 -



Jansen, Slinger & Cusumano, Michael. (2013). Defining Software Ecosystems

Open OT architectures have a great future!

Ecosystem : Characteristics & Products

Characteristics	Options
Base technology	<ul style="list-style-type: none">• Application platform• Service platform
Coordinators	<ul style="list-style-type: none">• privately owned• consortium
Extension market	<ul style="list-style-type: none">• No extension market• A list of extensions• an extension market• a commercial extension market• multiple extension markets
Accessibility	<ul style="list-style-type: none">• for free• after a screening• after making a payment

Base technology



Extension market



Dilemma : platform (hardware/software) dependent?

Jansen, Slinger & Cusumano, Michael. (2013). Defining Software Ecosystems



Digitalization
Industrie 4.0

Smart Production
E-Mobility
Smart Energy

Energy Efficiency
Smart Infrastructure

Smart Buildings
Renewables

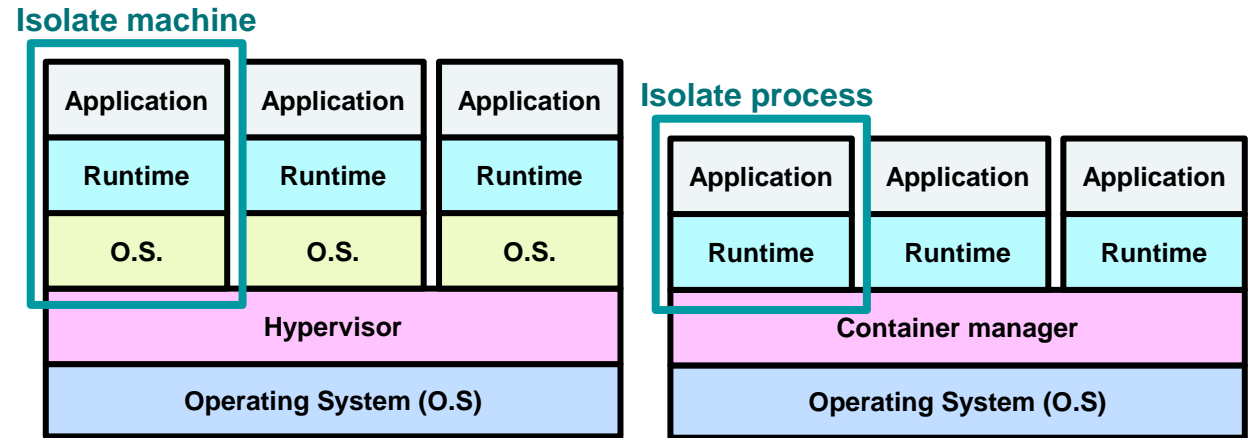
Bert van der Linden | 2020

Virtualisation of OT architectures

Open OT architectures have a great future!

Virtual machines versus containers

- Two paradigms of virtualisation:
 - Machine level (Virtual machines);
 - Process level (Containers).
- The Open container initiative opens the container technology.
- Virtualisation moves dependencies to the **software** layers.
- Still dependent of software platform.
- Virtualisation makes **consolidation** possible.

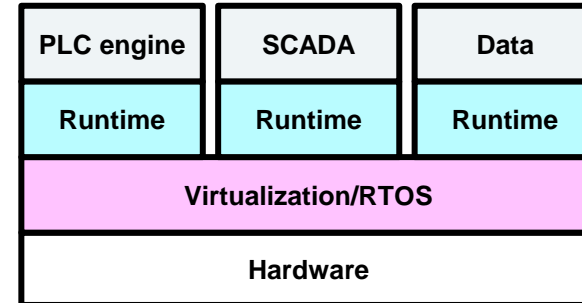


Where is the platform?

Open OT architectures have a great future!

Virtual control systems (DCS, PLC)

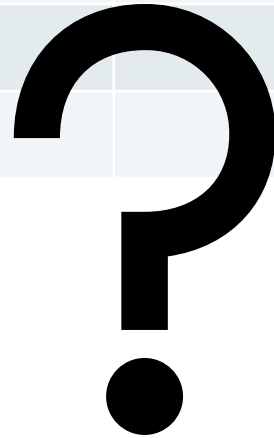
- Benefits:
 - Workload consolidation.
 - A great benefit with edge computing.
 - ..
- Challenges:
 - Needs **real-time** hypervisor/solution.
 - Worst case execution time (Control system in the cloud?)
 - Bandwidth capability
 - Logical **isolation** (Fault, **spatial**, **temporal**)
 - Determinism



Open OT architectures have a great future!

Effect trends on the non functional requirements?

Trends	Maintainability	Scalability	Compatibility	Simplicity	Reliability
Open OT architecture	+	+++				
Component-based development						
Containerisation						
Virtualisation						



Thank you

Bert van der Linden

Program Manager Technical Education

PHOENIX CONTACT B.V.

Postbus 246

6900 AE ZEVENAAR

Tel. +31 (0)316 59 17 10

Mobile +31 (0)6 82 15 52 72

mailto:blinden@phoenixcontact.nl

www.phoenixcontact.nl



Bert van der Linden | 2020

Webinar Open OT-architecturen (NIEUW)



Classificatie



Investering

Kosteloos

Data:

5 oktober 2020	16:00 - 16:45	Eigen device
18 november 2020	09:00 - 09:45	Eigen device
1 december 2020	16:00 - 16:45	Eigen device

Tip!

Direct inschrijven (mijzelf en/of anderen)