openMOS

open dynamic manufacturing system for smart plug-and-produce

Dr. Chih-Hong Cheng
fortiss - An-Institut Technische Universität München

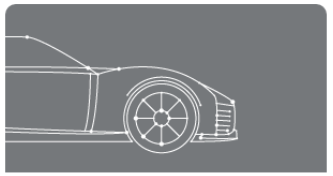
This work is supported by the H2020 project openMOS, GA no. 680735.

fortiss

Landesinstitut des Freistaats Bayern

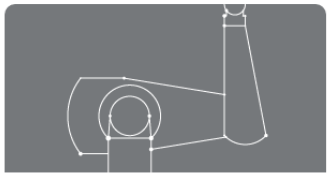
fortiss GmbH is an independent non-profit research and transfer institute for software-intensive systems and services

AUTOMOTIVE INDUSTRY



fortiss entwickelt integriert-modulare IKT Architekturen und neuartige Funktionalitäten für das software-basierte Automobil von morgen.

INDUSTRIAL AUTOMATION



fortiss entwickelt Software-Steuerungen und kombinierte Mensch-Roboter-Arbeitsplätzen zur effizienten Realisierung wandelbarer Produktionsprozesse.

AEROSPACE

fortiss develops integrated-modular methods, processes, and tools for a seamless and compositional development, maintenance, and (re-)certification process.

PUBLIC ADMINISTRATION

fortiss entwickelt Methoden und Werkzeuge zur Gestaltung und zum Management IKT-gestützter integrierter Bürgerservices.

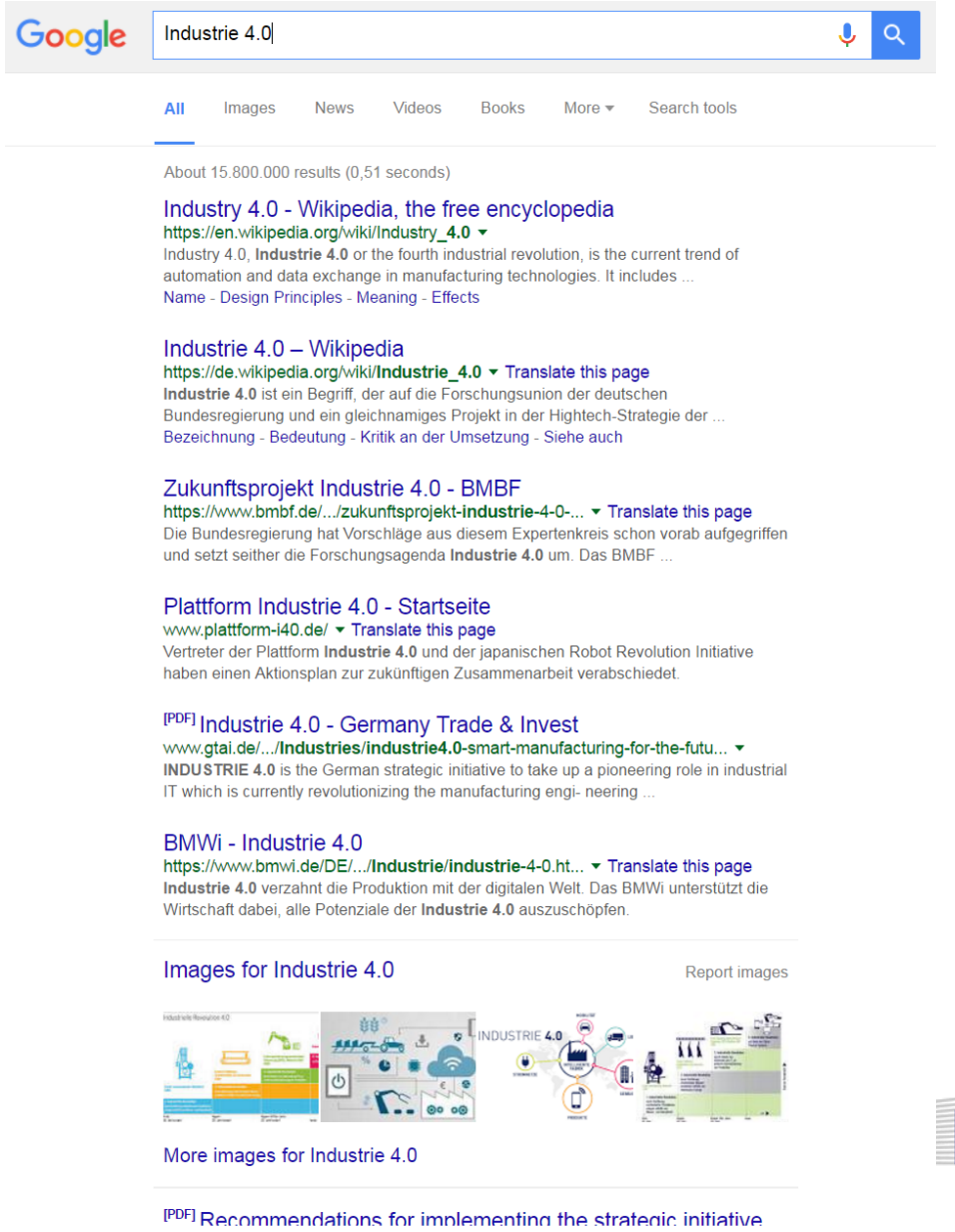
Agenda

- fortiss
- Industrie 4.0 and the H2020 openMOS project
  - Concept
  - Architecture
  - From system & product specification to machine configuration
  - Security and safety
- Conclusion & next steps

Industrie 4.0 (DE)

OR

Industry 4.0 (EN)



European Commission

Industrie 4.0 – Modular Smart Factories

- Orchestra
  - Pre-configured slot
  - Perfect engineering / tuning
  - For larger audience (Large-volume production)

- Jazz band
  - Freedom to join / leave at any time
  - No excessive pre-engineering
  - For smaller audience (E.g., production of size 1)

H2020 openMOS Project

1) Embed ***plug-and-produce*** (PnP) capabilities into automation devices, robots and machines
2) Enable ***vertical and horizontal connectivity*** between PnP automation components and higher-level control and business functions
3) Create a easily extendable and adaptable manufacturing operating system (MOS) that permits the ***easy introduction of new products, work orders and changes in the equipment*** and allows easy deployment of optimization and changeover management strategies.

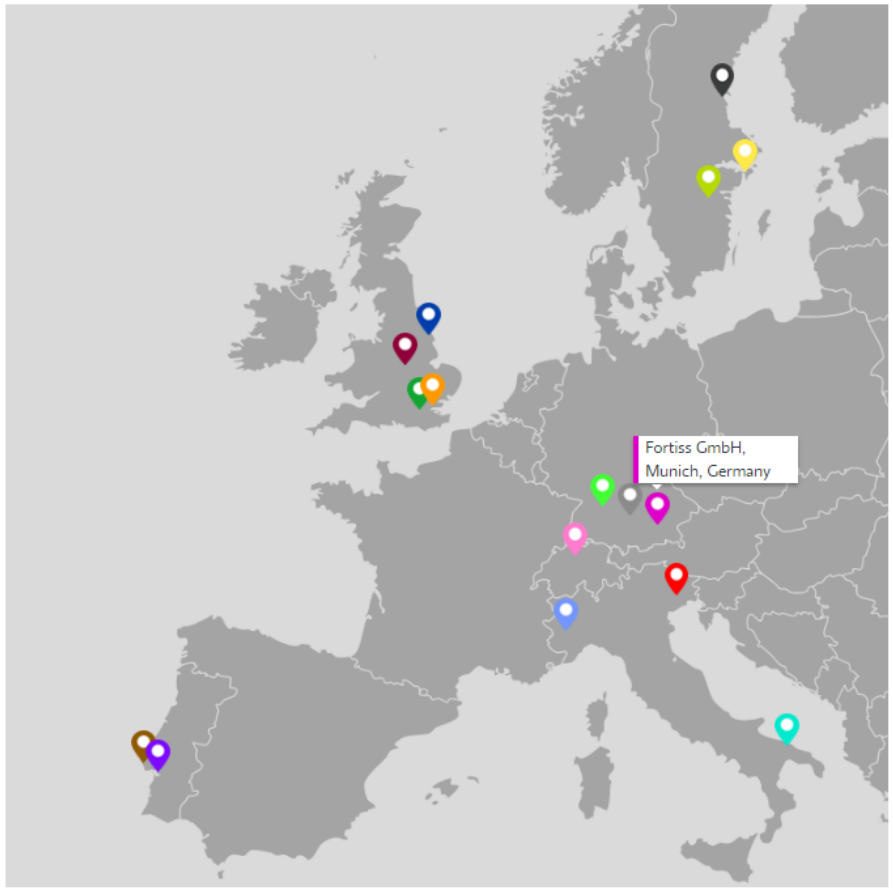
openMOS

Home | About | Work Packages | Partners | News & Events | Downloads | Contacts

Home >

Partners

- Afag Automation AG
- Asys Automatisierungssysteme GmbH
- Centre for Engineering and Manufacturing Excellence Ltd Lbg
- Electrolux Italia S.P.A.
- Elrest Automationssysteme GmbH
- Ford Motor Company Ltd
- Fortiss GmbH
- Inotec Ltd
- Introsys - Integration for Robotic Systems, Integração de Sistemas Robóticos, SA
- Kunliga Tekniska Hoegskolan
- Linkopings Universitet
- Loughborough University
- Masmec SPA
- SenseAir AB
- Uninova - Instituto de Desenvolvimento de Novas Tecnologias
- We Plus S.r.l.



This project is supported by the European Commission under the Horizon 2020 programme.

openMOS

Member countries

- UK
- Sweden
- Portugal
- Italy
- Germany
- Switzerland

fortiss

European Commission

Technology exists, but imperfect

Goal	State-of-the-art
Embed ***plug-and-produce*** (PnP) capabilities into automation devices, robots and machines	• Plug-and-Play for printer, mouse, keyboard • Robots with program-by-teaching capability • Machine-level: some common look-and feel (e.g., PackML), still far from arguing PnP
Enable ***vertical and horizontal connectivity*** between PnP automation components and higher-level control and business functions	• Connectivity exists, but engineering time can be long • Cloud-based solution possible, but IoT can be scary for security and privacy
Create a easily extendable and adaptable manufacturing operating system (MOS) that permits the ***easy introduction of new products, work orders and changes in the equipment*** and allows easy deployment of optimization and changeover management strategies.	• Available in the automation pyramid (from ERP to MES), but too expensive for SME • Still hard to argue the process is „easy“

Solution Concepts

- Use skill to abstract details and focus on capabilities
- Production = executing skill-executor(skill plan, concrete parameters)

Skills

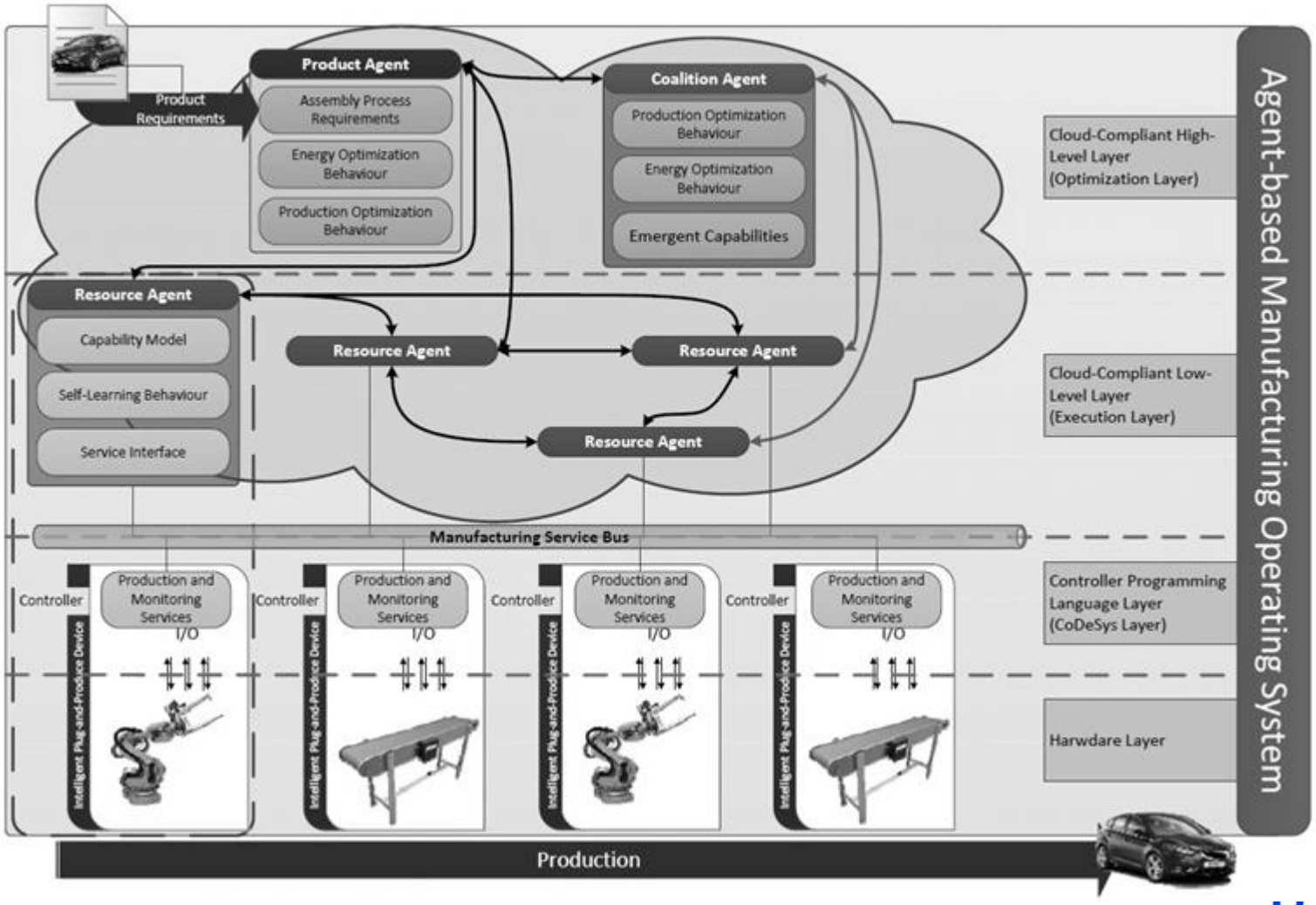
- Represent the physical entity in the cloud, as an virtual entity
- Allow connectivity between physical entity and virtual entity

Agents

- Act as the centralized communication mean
- Move from automation pyramid to automation platform

Manufacturing Service Bus (MSB)

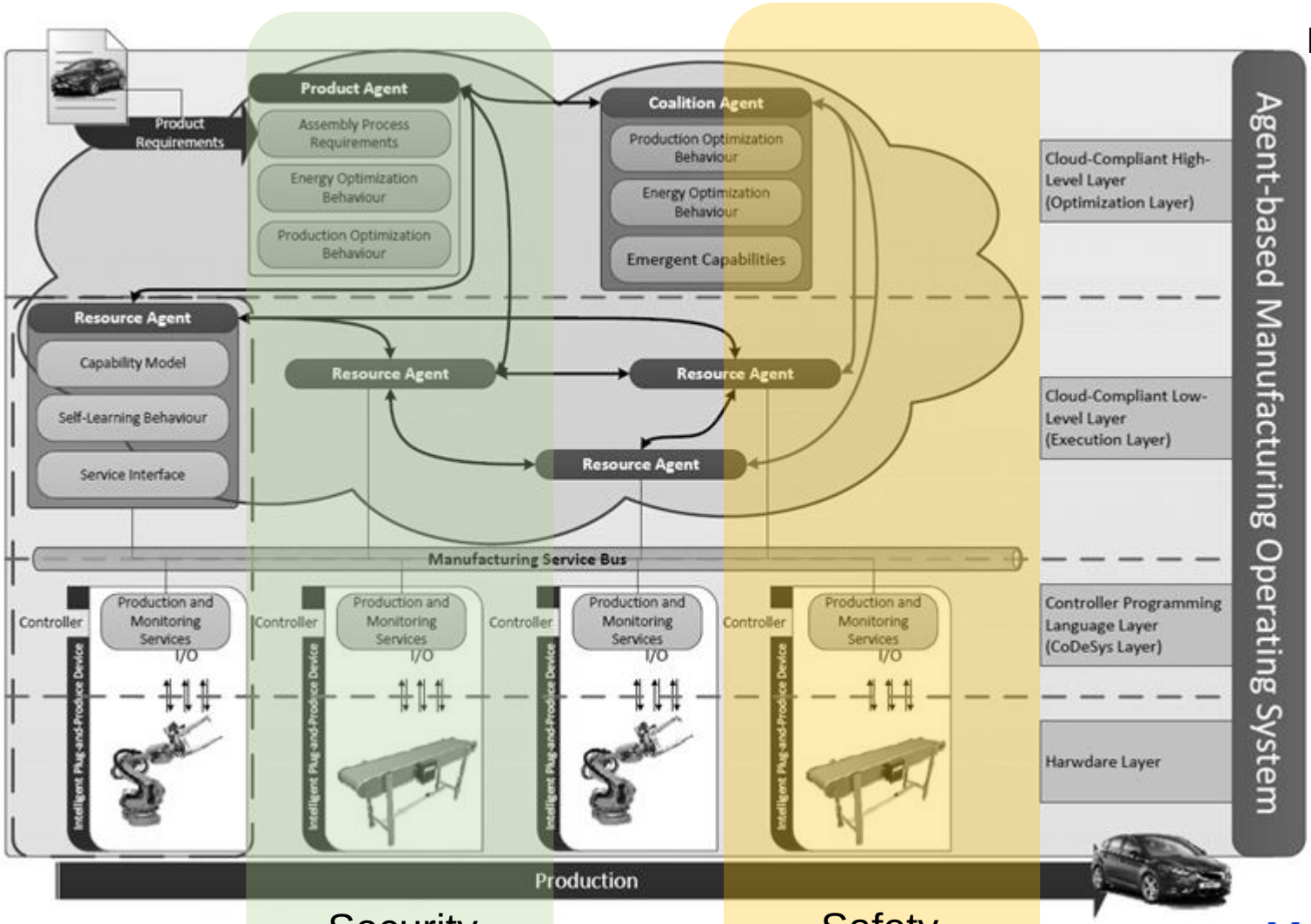
Conceptual diagram



Consortium-based metastandards

Foundation for Intelligent Physical Agents

IEEE P2660.1 WG
oneM2M
Platform 4.0 (RAMI 4.0)
IIC (IIRA)
AIotI
IEC Working Group
China Manufacturing 2025



ISO/IEC/OMG/IEEE Standards

ecl@ss Collada CAEX

RDF / RDFS CORA

AutomationML

OPC UA MQTT DDS

PackML (TR88)

IEC 61499

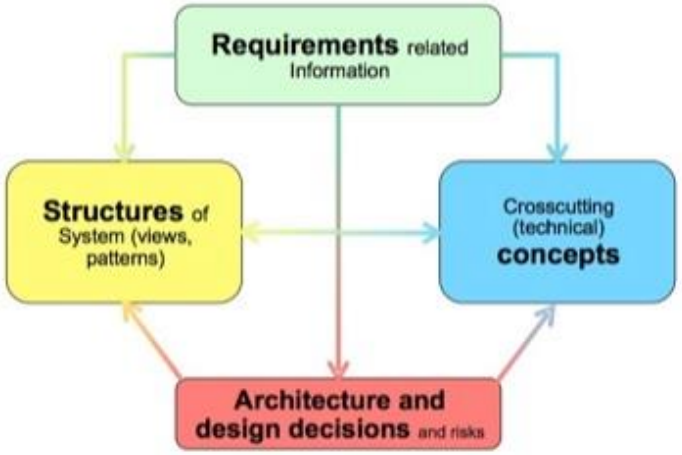
FDI

Agenda

- fortiss
- Industrie 4.0 and the H2020 openMOS project
  - Concept
  - Architecture
  - From system & product specification to machine configuration
  - Security and safety
- Conclusion & next steps

Architecture design

- Adapt (standardized) software engineering process
  - System overview
  - Views and mapping between views
    - Static view (components and logical dependencies)
    - **Run-time view** (how components interact, using UML sequence diagrams)
    - Deployment view
  - Architecture decisions
  - Concepts (cross-cutting concerns)
  - …



Source: Arc42.org

What actions are needed?

(from the view of designing intelligent components)

			Phase 1: Discovery	Phase 2: From product requirement to machine configuration	Phase 3: Production and change
Scope			**Reach the stage of "administration shell" by connecting to its virtual partner, i.e., a Industry 4.0-ready machine**	Receive line-level specific instructions and product-specific parameters	The process of triggering production and performing job switch, run-time reconfiguration, error handling
Features	Smart decisions			x	
	Plug		**x**		
	Produce				x
	Unique device addressing		**x**		
	System requirement			x	
	Communication	Requirement		x	x
		Skills		x	x

fortiss

European Commission

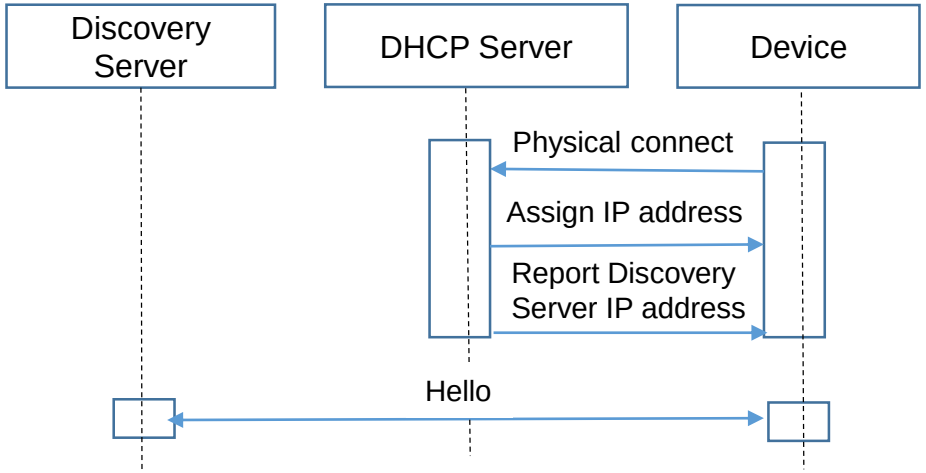
Phase 1: easy and problematic steps

- The process of
  - Physical insertion (easy)
  - IP assignment (easy)
  - Being informed over IP of UA discovery server (doable via zeroconf)

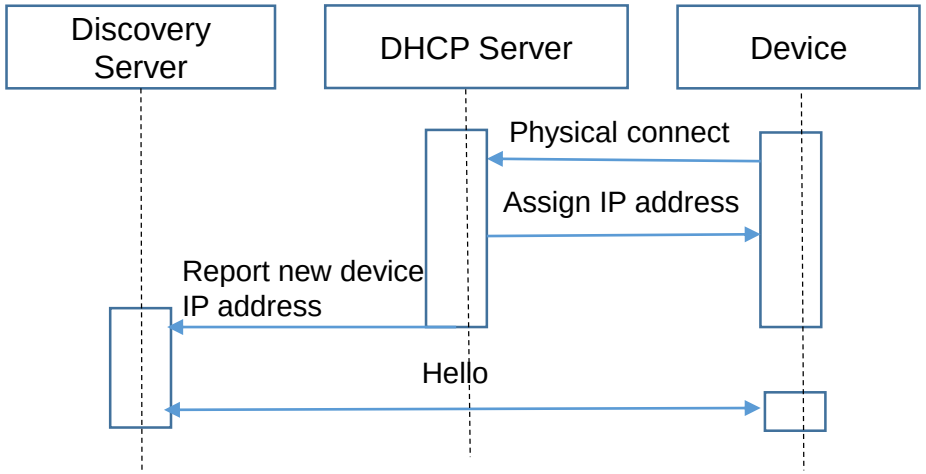
  - Understanding topological information
  - Reaching the state where resource agent and skill executor are connected to co-work as a smart component (administration-shell in I4.0)

Phase 1: Discovery (Physical insertion, IP assignment, discovery server)

Option 1:



Option 2:

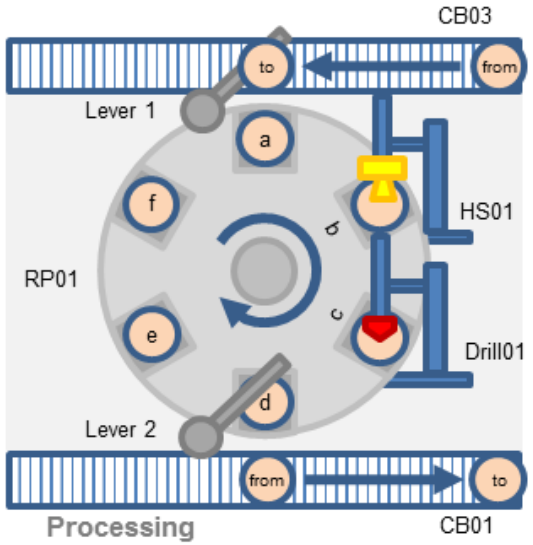


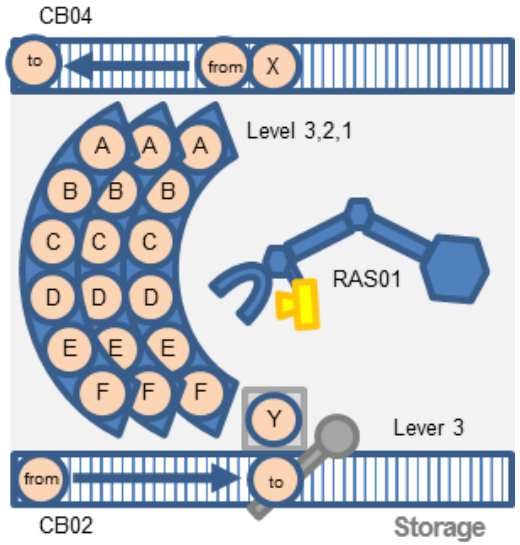
Q: Option 3, 4, 5…?

Unique device addressing via IPv6 or other means?

Phase 1: Discovery (Topological Information)

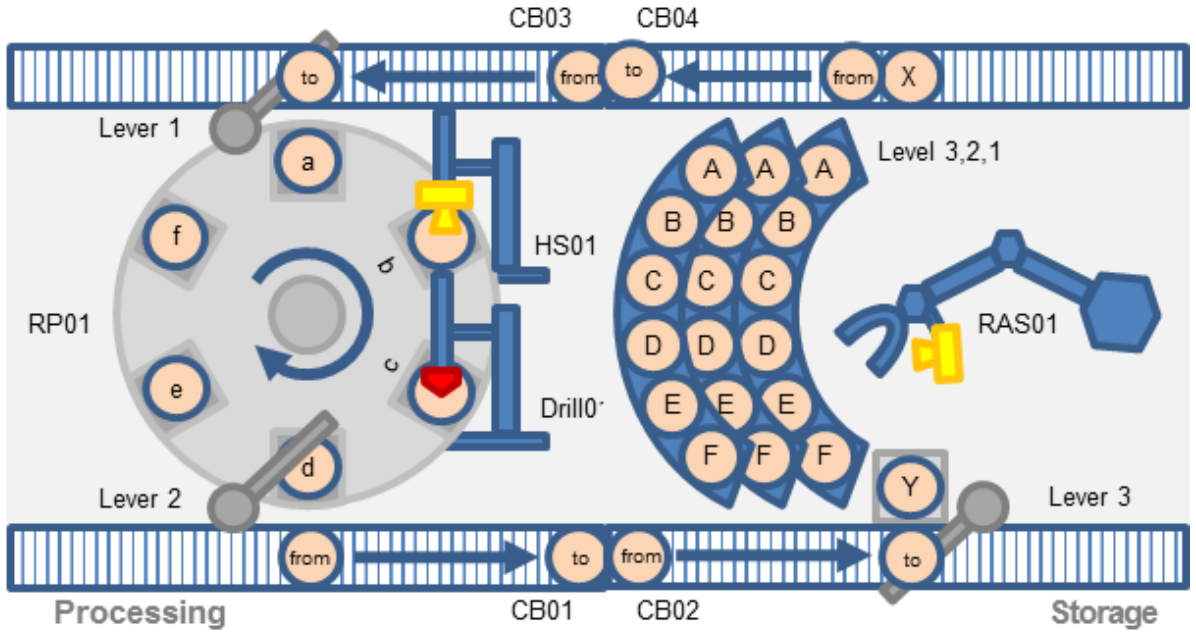
- Need to know (infer) the topological information (e.g., spacial overlap, neighboring information), when it is being plugged.





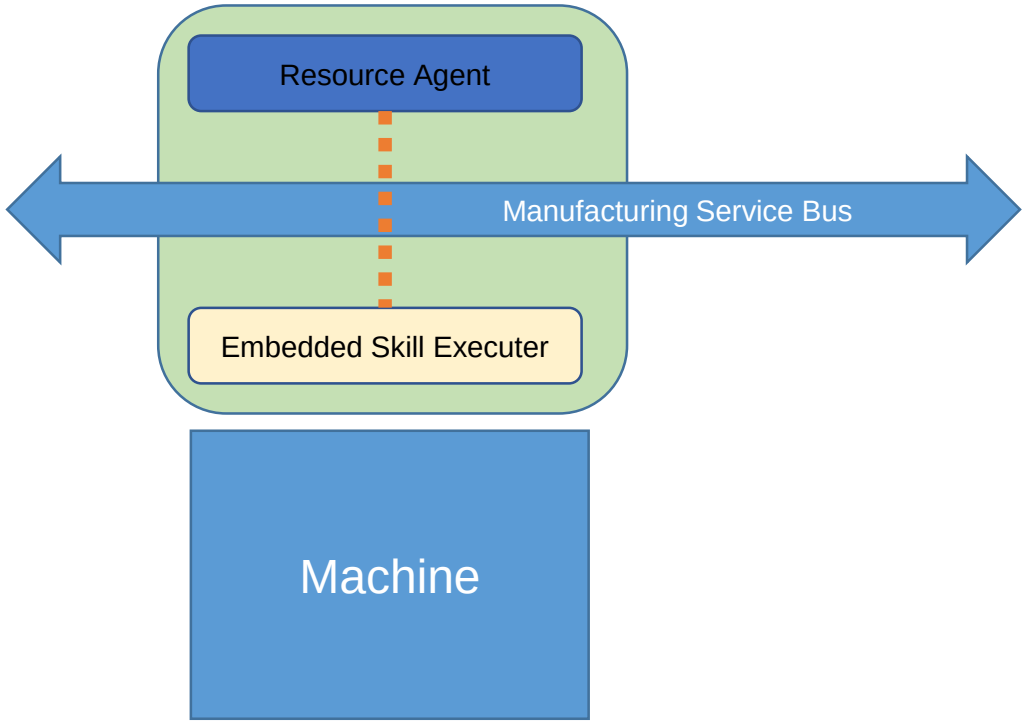
Phase 1: Discovery (Topological Information)

- Need to know (infer) the topological information (e.g., spacial overlap, neighboring information), when it is being plugged.



Phase 1: Discovery (Reaching ``smart component´´)

- There is a need to connect the machine with its “virtual representation“, i.e., to connect resource agent and embedded skill executor



Q: How & when is a resource agent created?

- Option 1 – created automatically when the machine is connected
- Option 2 – created before and need a configuration step

Scope of smart component / adminsitration shell

Agenda

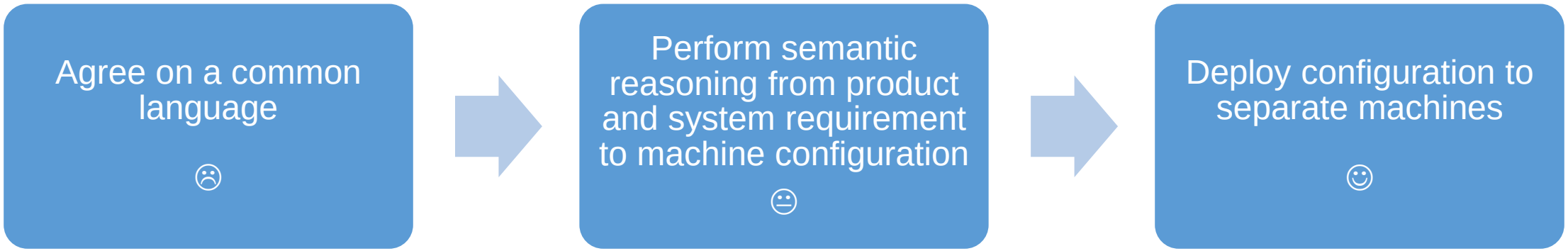
- fortiss
- Industrie 4.0 and the H2020 openMOS project
  - Concept
  - Architecture
  - From system & product specification to machine configuration
  - Security and safety
- Conclusion & next steps

What actions are needed?

(from the view of designing intelligent components)

			Phase 1: Discovery	**Phase 2: From product requirement to machine configuration**	Phase 3: Production and change
Scope			Reach the stage of “administration shell” by connecting to its virtual partner, i.e., a Industry 4.0-ready machine	**Receive line-level specific instructions and product-specific parameters**	The process of triggering production and performing job switch, run-time reconfiguration, error handling
Features	Smart decisions			**x**	
	Plug		x		
	Produce				x
	Unique device addressing		x		
	System requirement			**x**	
	Communication	Requirement		**x**	x
		Skills		**x**	x

Challenge starts with mutual understanding



Agree on a common language

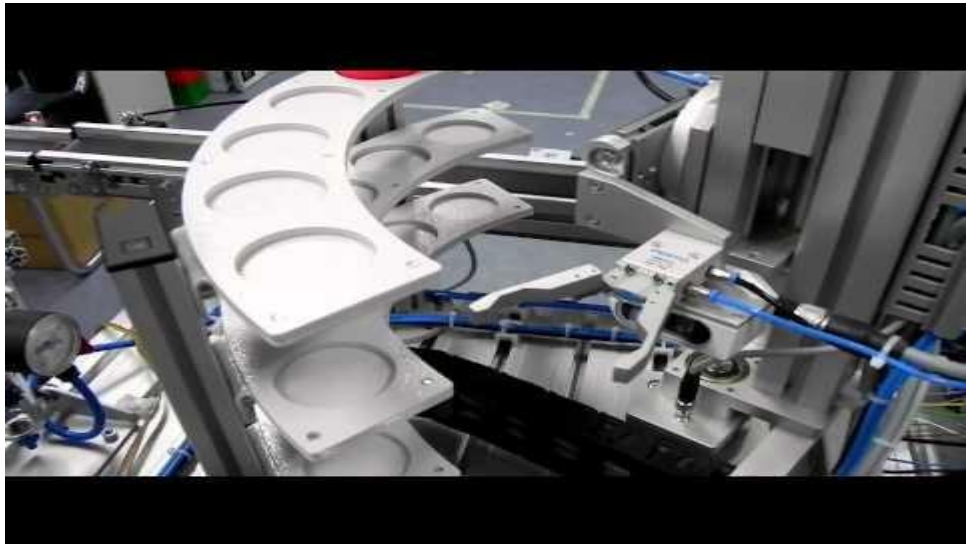
- Easy to achieve on the project scope, difficult to achieve on a business scope
  - One goal of Platform Industrie 4.0
  - Agreement can be on various levels
  - Different industry standards provide agreement over different levels

TABLE I. FEATURES PROVIDED BY STRUCTURAL DEGREE

Structural Degree	*Exposed Information*	*Industrial Context*
S0: Repository	Unstructured/structured data	Devices catalogues, file structure,
S1: Terminology	Controlled vocabulary	Tags, annotations
S2: Glossary	Description over vocabularies	Human readable documentation, HMI
S3: Thesaurus	Basic relationships (association), similarities	Profile mapping, technology Integration
S4: Taxonomy	Tree Structure, parent-child relations, classifications	Abstraction (typing), device classification
S5: Ontology	Typed elements and relations	Interfaces, inheritance, topologies

Then, semantic reasoning

- Software synthesis / automated deduction for industrial automation [CAV´12]: generating distributed recipes + orchestration plans
  - Skill library of every machine
  - High-level specification (line-level error handling, product specification, etc)
  - Model uncertainty & error as environment moves → game-theoretic setup



Process and sort based on color

Error handling

What can we do now (in the project, currently)?

- Stay with assumptions
  - Assume that all components speak the same language (semantic harmonization)
  - Assume that the distributed recipe exists (i.e., there exists an oracle for synthesizing such a recipe)
    - Synthesis of distributed controller is a research topic of its own
      - Distributed controller synthesis for LTL is undecidable in general (POPL´89)
      - Deciding chreography is distributable if behavior is the same on 1-bounded asynchronous model (POPL´12)
      - Bounded synthesis (ATVA´07), …
- Focus on deployment and execution of recipe
  - It requires a scripting engine executed during run-time (e.g., Lua)

Agenda

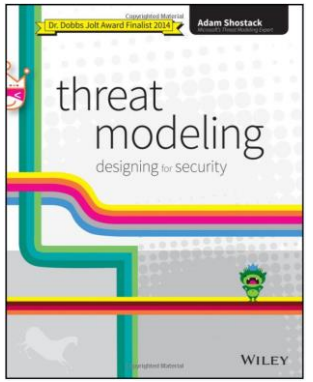
- fortiss
- Industrie 4.0 and the H2020 openMOS project
  - Concept
  - Architecture
  - From system & product specification to machine configuration
  - Security and safety
- Conclusion & next steps

Security and safety

- [Strategy] examine existing approaches (process-based) and try to harmonize with them

Security

- Apply threat modelling as specified in the Microsoft Security Development Lifecycle (SDL)



Safety

- No advanced methodology apart from simple component replacement, or zone-based control

Some threats being modelled for device adaptor (incomplete list; preliminary analysis following SDL)

Threat description	Strategies which address each threat
Spoofing of MSB or agent	Mitigation: Device adaptor should be equipped with mechanisms to understand digital signature Transfer: The MSB should also support digital signature (this also requires that MSB needs to, prior to deployment, know the digital signature of the device adaptor)
File spoofing or tempering – when the system is plugged-out from system, malicious user can download or change the recipe or parameter	Mitigation: When network is disabled, disable untrusted local modification of the recipe or parameter, preferably via access control list (ACL).
An attacker can try one credential after another.	Mitigation: Enable maximum trial of password
Attacker can reuse password	Mitigation: Password management (e.g., periodic change of password to reduce the risk)
Device adaptor is shipped with a default admin password	Mitigation: Enforce that by shipping, the password is always generated randomly.
Repudiation: dispute due to change of logs	Transfer: The record of logging stored in the resource agent should only be modifiable by 3rd party, i.e., independent to machine builder (who creates a machine) and product builder (who designs a recipe).

MSB

- The original responsibility of automation pyramid is now to a single (logical) bus.
- Not OK to say *it´s enough, as we use OPC UA / DDS / MQTT*
- Challenge – from semi-formal analysis (expert review-based) to formal analysis (theorem proving)
  - E.g., do security analysis using applied Pi-Calculus

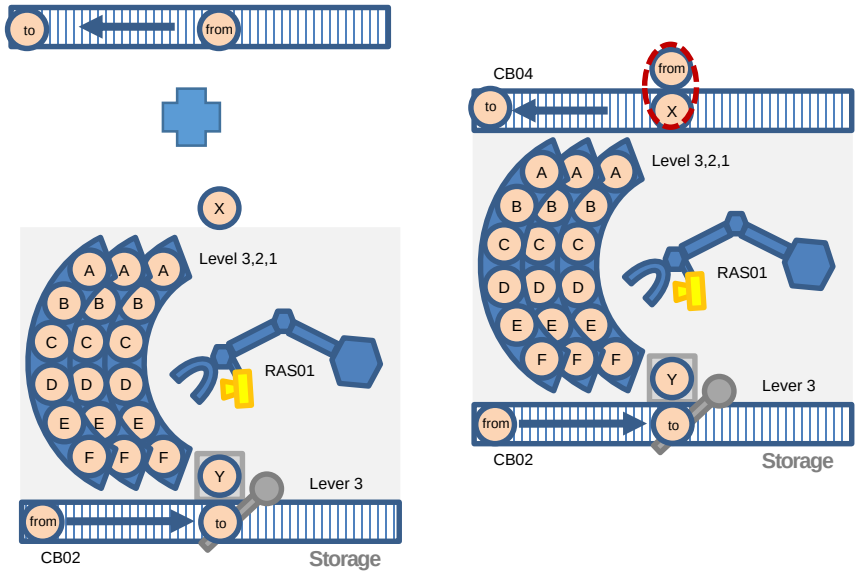
Safety (Process and Function-wise)

- From Plug-and-Produce (PnP) to Plug-Verify-Produce (PvP)?
  - Can be a way to go, via formal methods or virtual co-simulation
  - Still encountering problems like model fidelity and process issues

- Our architecture description is unfortunately semi-formal (UML diagrams)
  - Still draw meaningful analysis out of it?

Safety (Process and Function-wise)

- Some issues
  - Change of topological (physical) structure might create safety problems
  - Replacing to a faster component might impose safety problem
  - Environment conditions of a skill to operate correctly can be incomplete (model fodelity issue)
- Some mediation to comply with safety standards:
  - Assume zone-based control
    - Plug-and-produce only on pre-defined zones with safe-guarding mechanisms predefined, pre-analyzed, and pre-enforced
      - over-engineering
    - Work on simple component replacement scheme



Moving from standard compliance to function correctness

…

These self-integrating systems promise significant benefit, but also have the potential for harm, so as they integrate they should adapt and configure themselves appropriately and should construct an “assurance case” for the utility and safety of the resulting system.

Thus, trustworthy self-integration requires autonomous adaptation, synthesis, and verification at integration time, and this means that **embedded automated deduction** (i.e., theorem provers) will be the engine of integration in the Internet of Things.

…

(abstract of John Rushby‘s talk for the ARISE seminar)

Agenda

- fortiss
- Industrie 4.0 and the H2020 openMOS project
  - Concept
  - Architecture
  - From system & product specification to machine configuration
  - Security and safety
- Conclusion & next steps

openMOS

- Make intelligent plug-and-produce possible

- Due to project constraints, need to establish work based on assumptions
  - Commonly agreed language
  - Oracle for synthesizing distributed production plan

- Security and safety – story told (analysis done), but not yet reaching a stage of „formally verified“
  - Will reach some formality in the project to perform analysis

fortiss

European Commission

Kontakt // Dr. Chih-Hong Cheng

fortiss GmbH
An-Institut Technische Universität München
Guerickestraße 25 · 80805 München · Germany

tel +49 89 3603522 513 fax +49 89 3603522 50

cheng@fortiss.org
www.fortiss.org

fortiss

European Commission