Cloud & Cyber-Physical Applications (Machines, IoT, Robots)

Josef Spillner <josef.spillner@zhaw.ch>
Service Prototyping Lab (blog.zhaw.ch/icclab)

August 10, 2017 | Parque Technológico Itaipu

PTI
Parque Tecnológico Itaipu

Background

Compound lecture based on previous work:

- workshop paper „Autonomous Participation in Cloud Services“ (2012)
- demo paper „Secure Distributed Data Stream Analytics in Stealth Applications“ (2015)
- lecture „Current and Future Platforms for Robotics“ (2015)
- research proposals „RoboPaaS“ and „Synergetic Emergent Behaviour in Cyber-Physical Systems of Systems“ (2016/17)
- lecture „ROS and Cloud Robotics“ (2017)
- new workshop paper submission „Distributed SLAM and Autonomous Exploration on PaaS“ (2017)

But also new material (unreviewed!)

When Cyber (Digital) is not enough

[icc.mtu.edu]

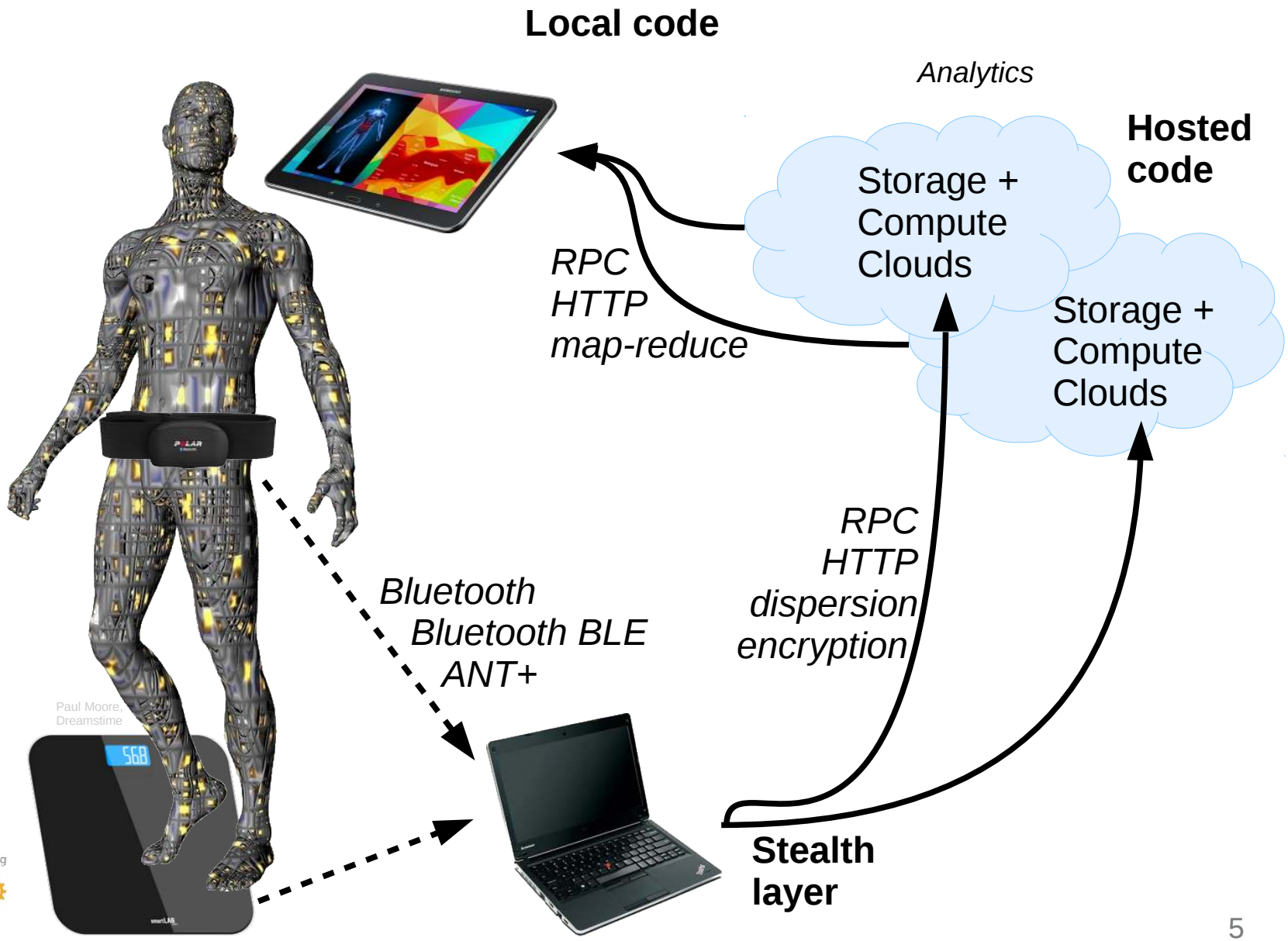
Cyber-Physical Application (CPA): A composite application executed in parallel across physical and virtual spaces.

CPA-Related Terms and Trends

	*			Simple interfaces (“services“, “applications“)
Hardware	Software	Hybrid (CPS)	Hierarchical (CPSoS)	Complex systems (“compositions“)
Robots	Devices	Containers	Functions	Physical entities (“components“)
Sensors	Actors	Logic	Control Loop	Building blocks (“objects“)
Adaptronic Materials	Reconfigur. Hardware	Virtualisation	Programmable Platforms	Physical foundation (“code“)

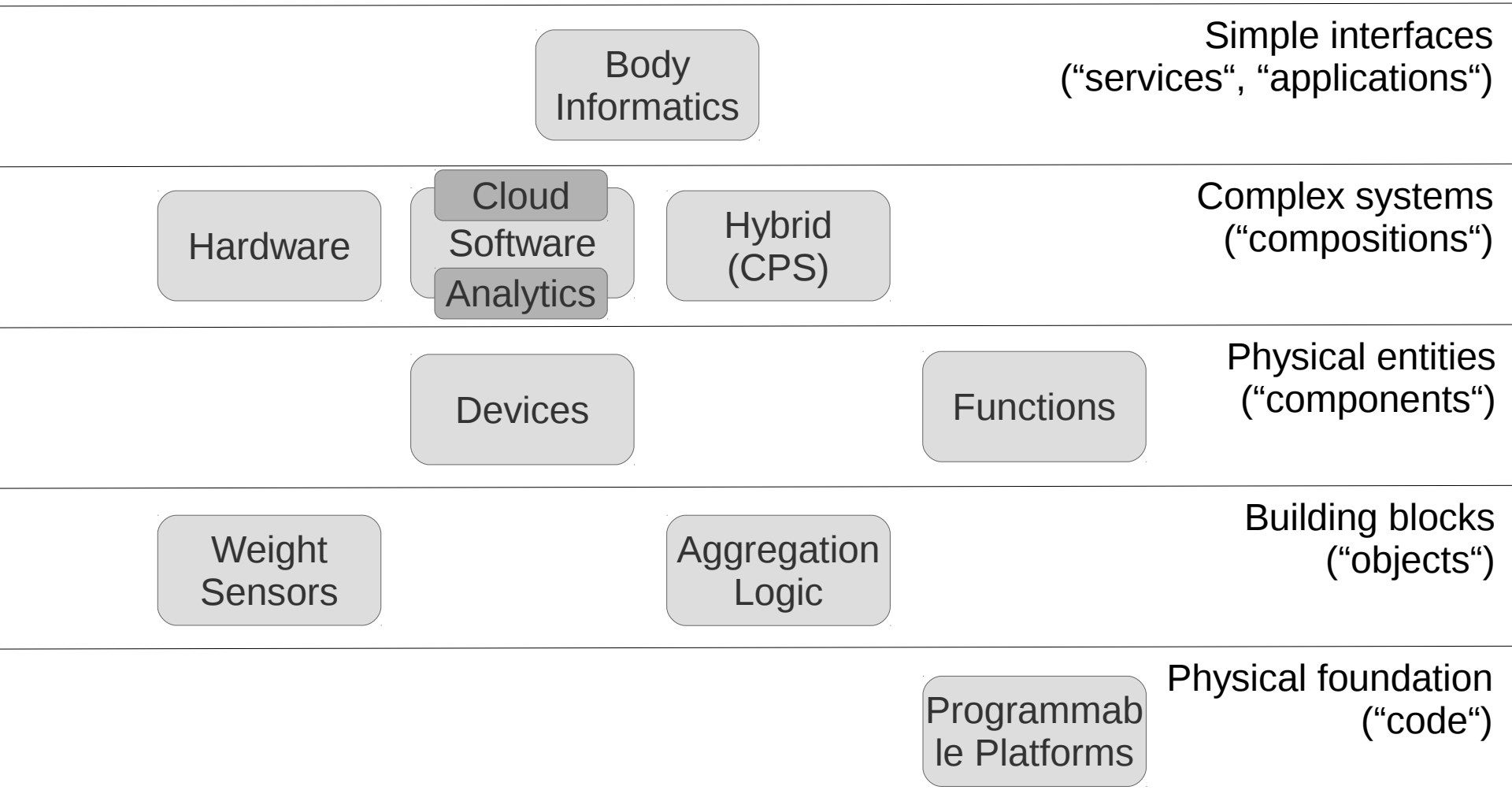
Service Engineering Research Area
ICCLAB SPLAB

Example: Body Informatics



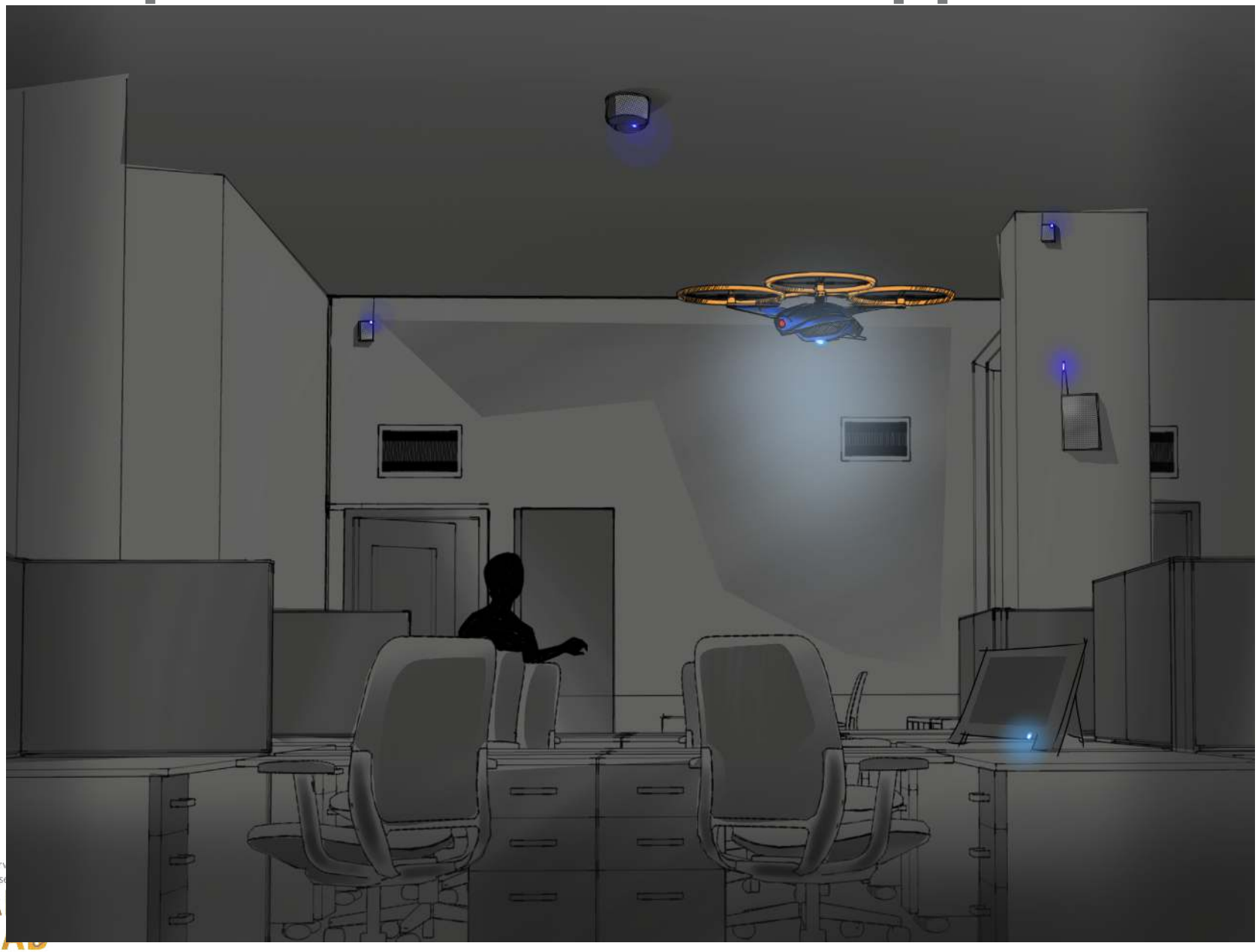
Service Engineering Research Area
ICCLAB SPLAB

Example: Body Informatics



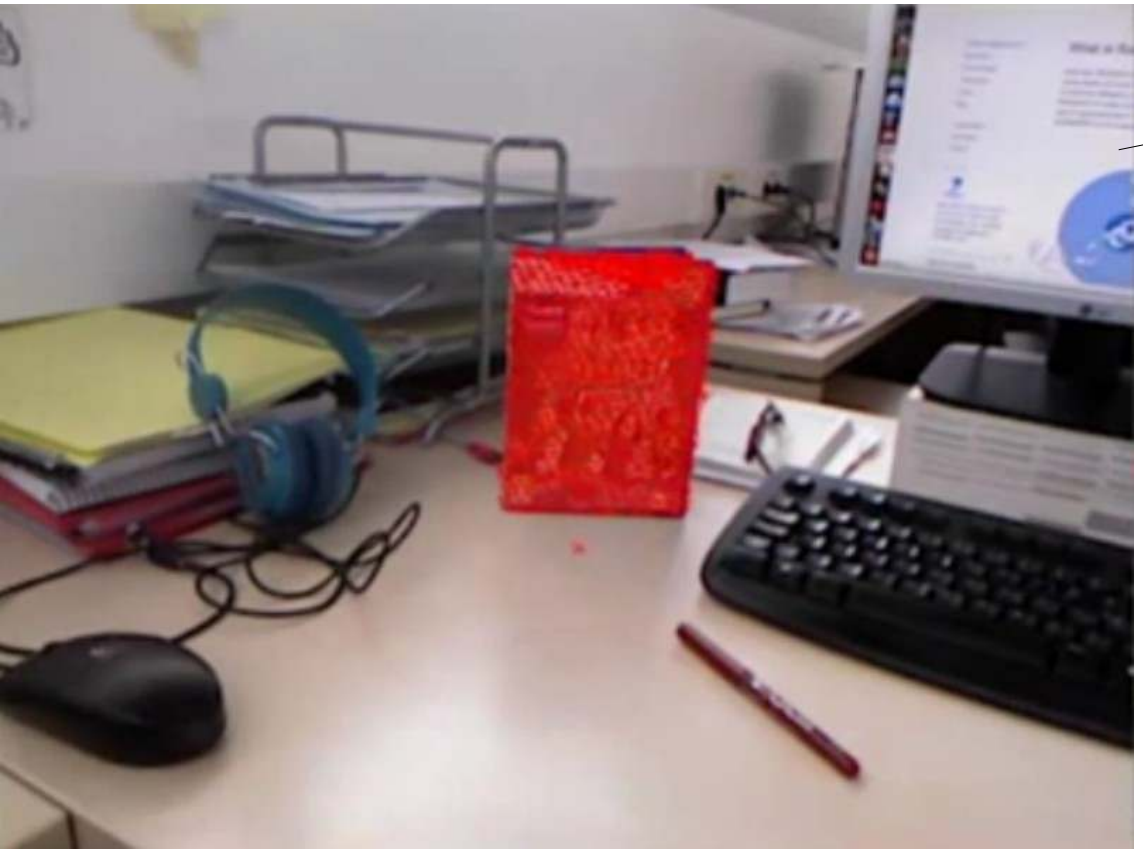
Service Engineering Research Area
ICCLAB SPLAB

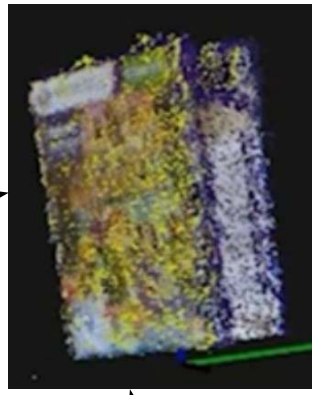
Example: Cloud Robotics Application



Example: Cloud Robotics Application

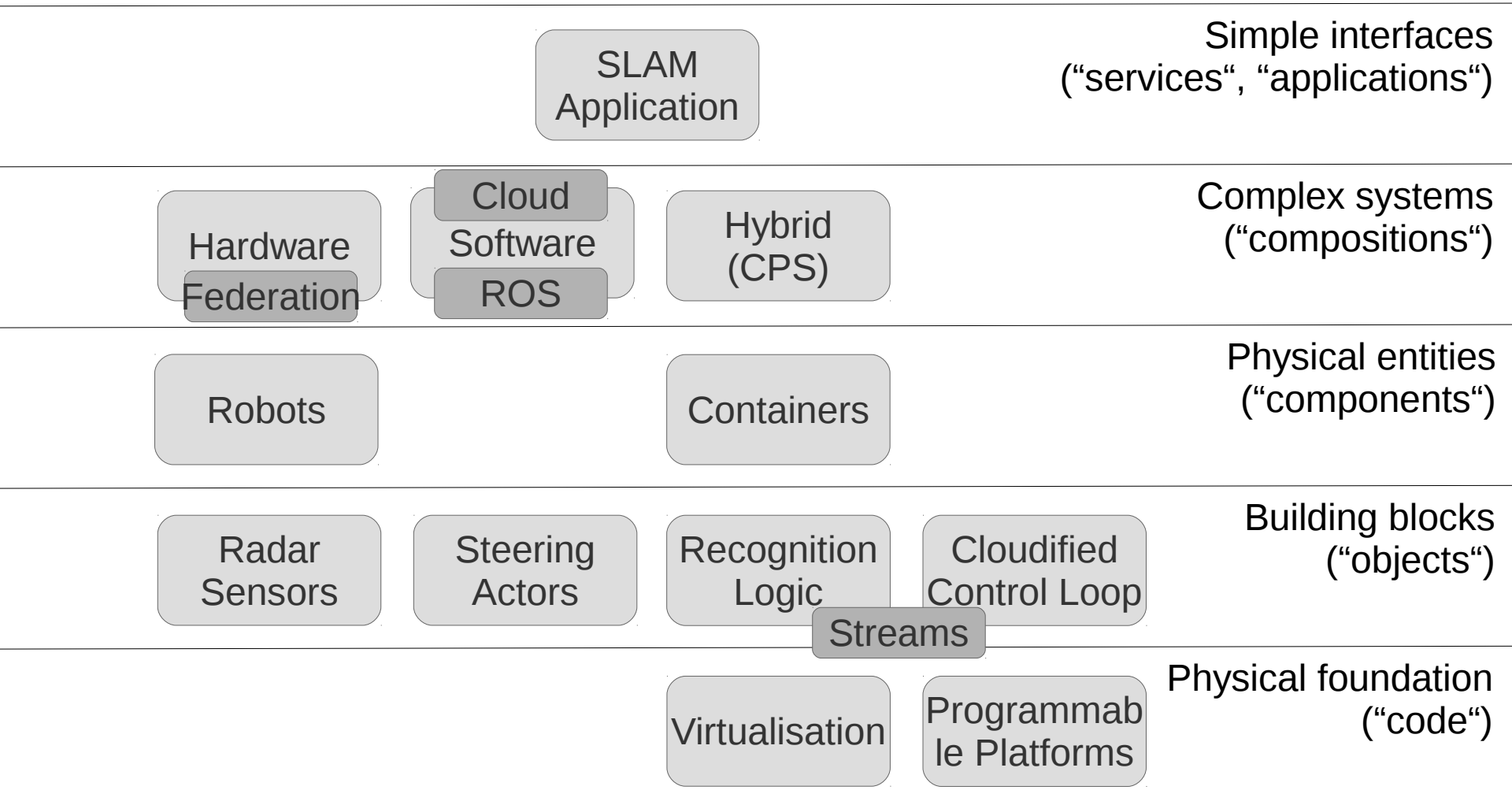
SLAM: Simultaneous Localization and Mapping (SLAM)
Sensoric capability: Mounted camera
Logic capability: Photo (feature) catalogue linked to object model





"video cartridge"

Example: Cloud Robotics Application



Service Engineering Research Area
ICCLAB SPLAB

Lecture Outline

1. Resource abstraction
   Practice: Using platforms and registries

2. Software encapsulation
   Practice: Using containers

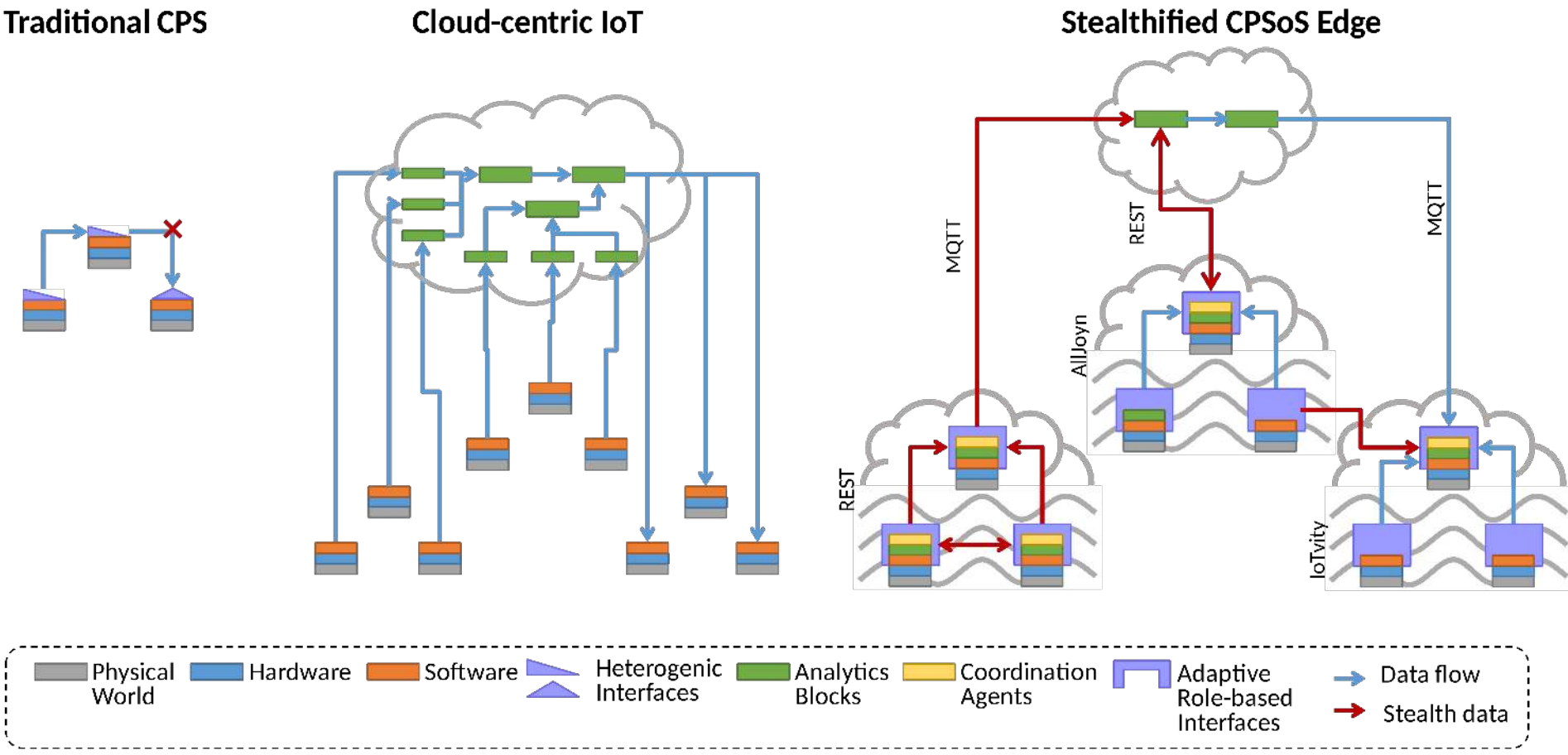
3. Autonomous service interaction
   Practice: Using automated signup

4. Controlled resource use
   Practice: Using stealth databases

5. Towards Platforms

Resource Abstraction / 1

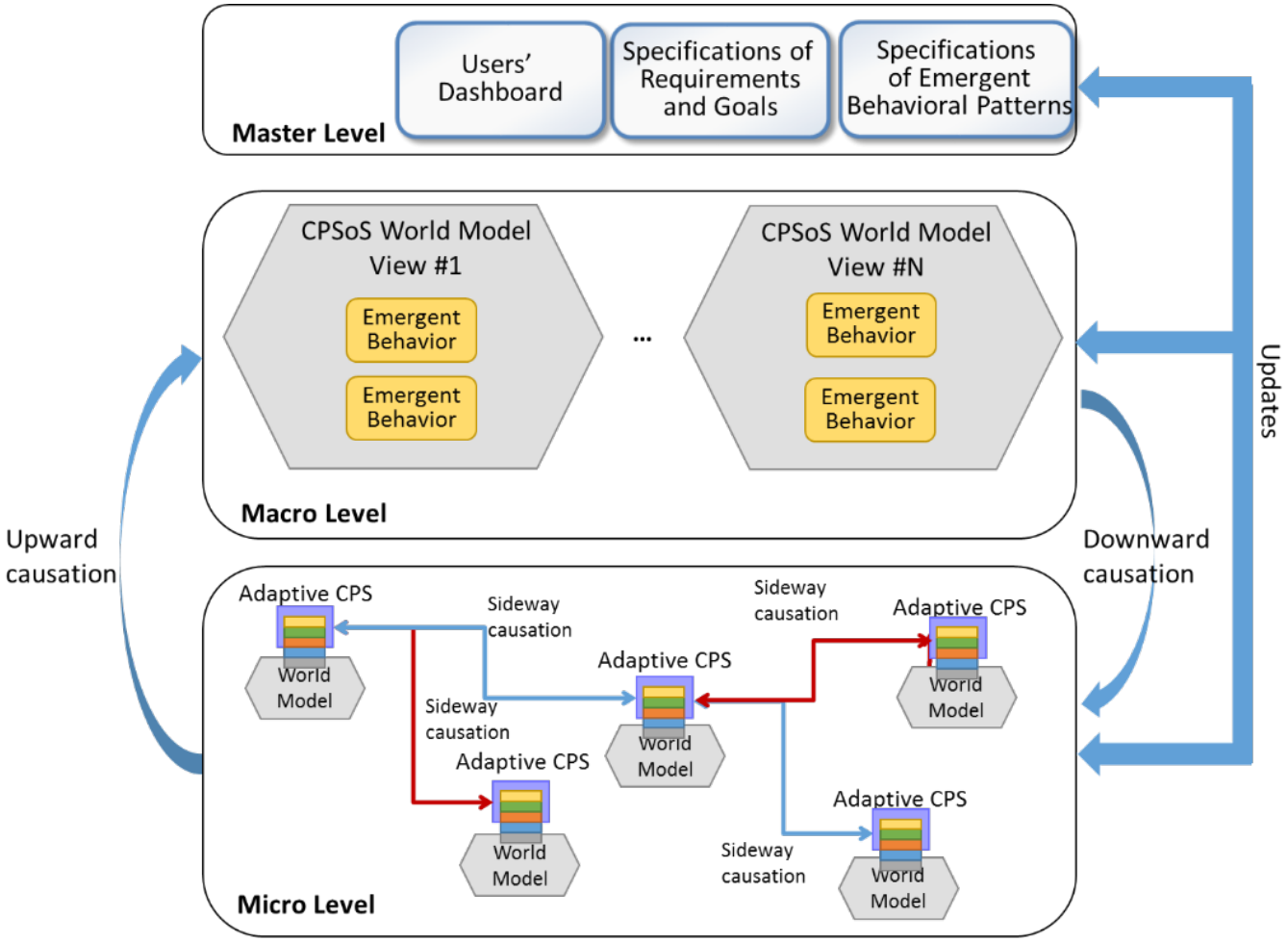
Architecture approach



Service Engineering Research Area
ICCLAB SPLAB

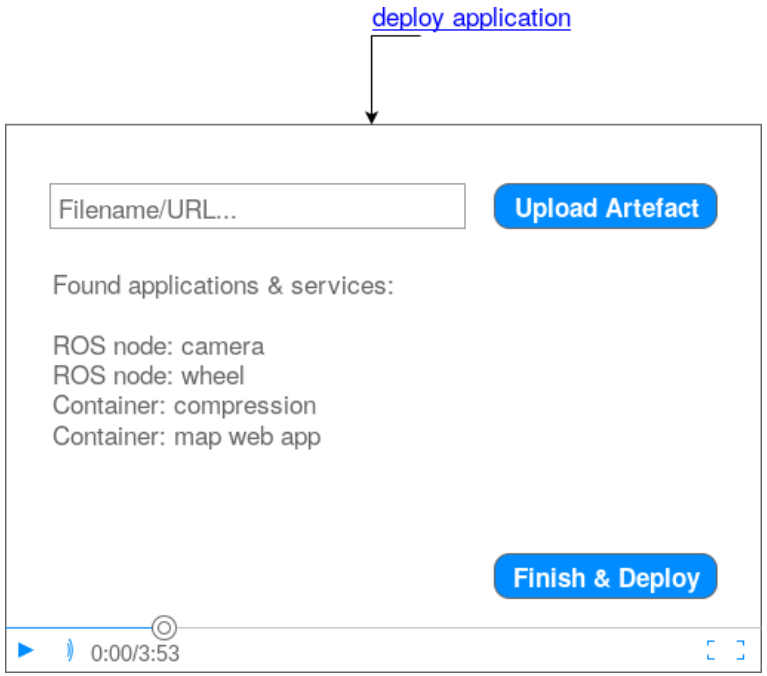
Resource Abstraction / 2

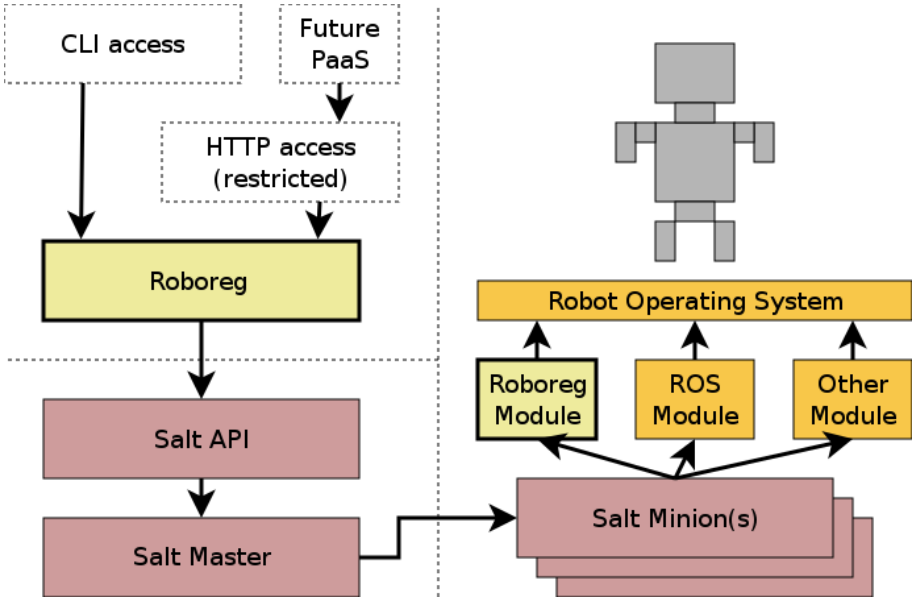
Support for emergent behaviour



Resource Abstraction / 3

Concrete architecture for federated robots





Compared to general CPS architecture

- analytics blocks implemented by cloud-hosted containers
- data flow implemented by web and messaging services
- selective messaging through system properties (multicast)

Service Engineering Research Area
ICCLAB SPLAB

Resource Abstraction / Practice

Using platforms and registries

Prerequisites:

- Salt Stack (infrastructure management)
  - `https://github.com/saltstack`
- Roboreg (registry for robots with discoverable capabilities)
  - `https://github.com/serviceprototypinglab/roboreg.git`

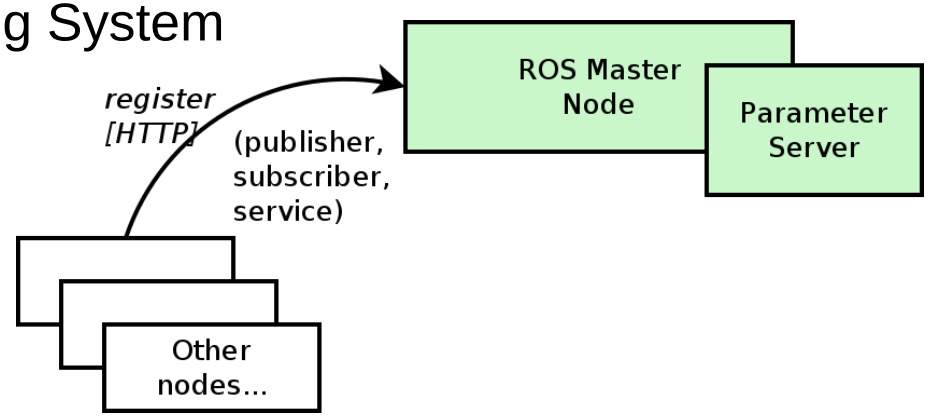
Instructions:

- follow Roboreg README (step-by-step guide)
- understand mix of HTTP calls, MQ operations and runners
- think of executing ROS commands on actual robots
- think of PaaS access to this functionality

Service Engineering Research Area
ICCLAB SPLAB

Software Encapsulation / 1

ROS: Robot Operating System



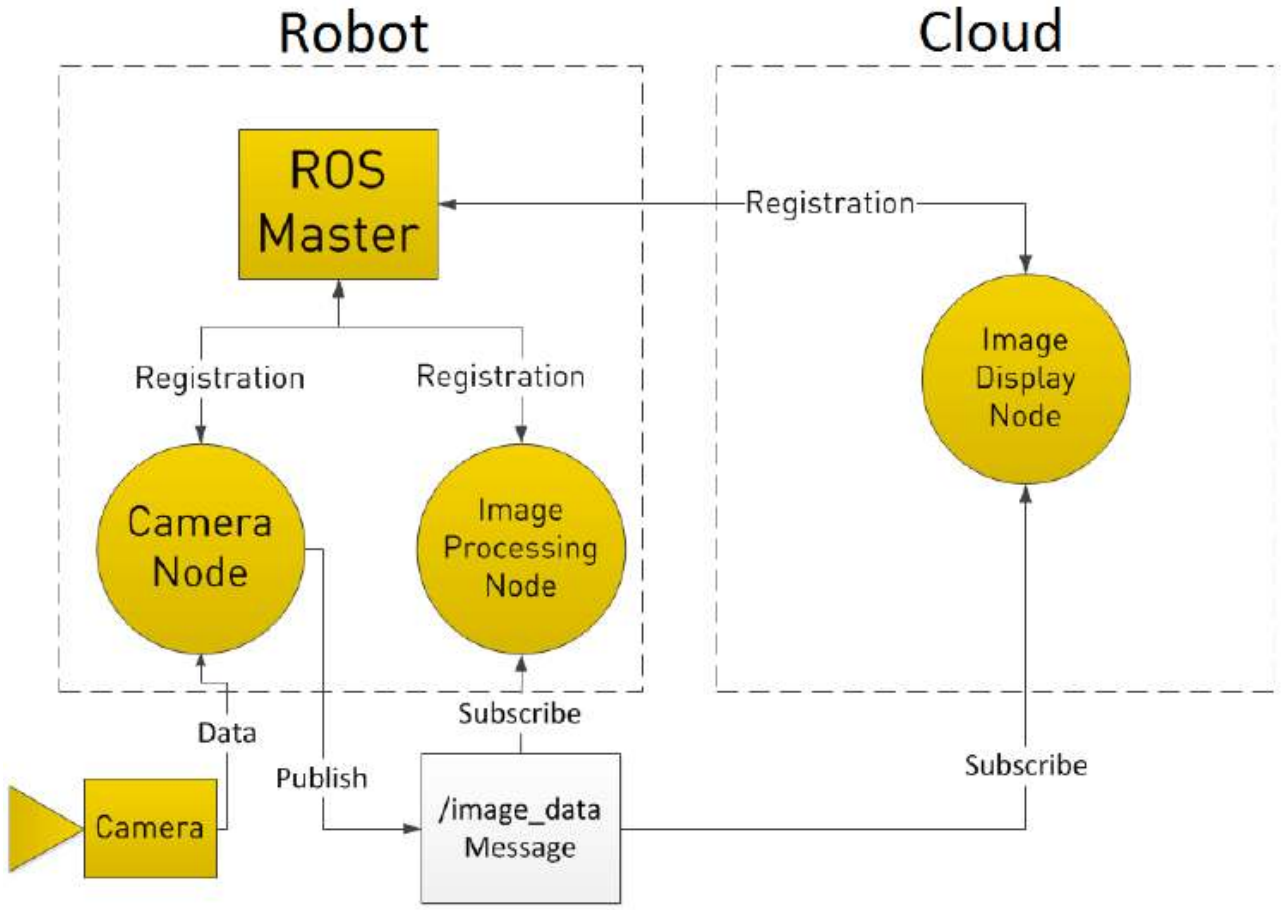
Nodes
- communicating executables
- connected by graphs
- advantages: fault tolerance, complexity reduction

Messaging
- simple types and arrays/structs
- transport via UDP and TCP, negotiated at connection time
- relies on 'Master' registry

Service Engineering Research Area
ICCLAB SPLAB

Software Encapsulation / 2

ROS nodes + cloud



Service Engineering Research Area
ICCLAB SPLAB

Software Encapsulation / 3

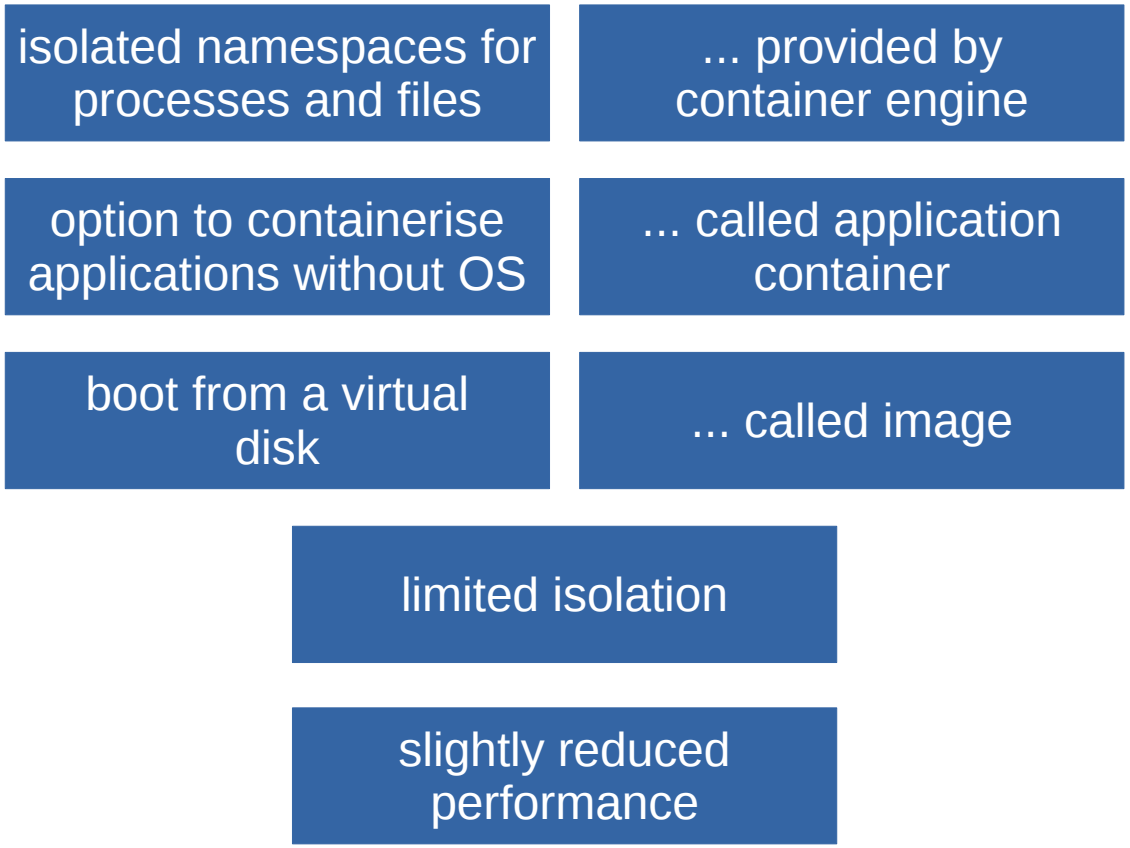
Portable software deployment
- across systems, architectures and platforms
- most suitable for CPS
- underpinning of service offerings for SaaS

Formats
- language-specific packages (e.g. JARs, eggs, gems, modules)
- system-level packages (e.g. DEB, RPM)
- executable container images (e.g. Docker)

Distribution and discovery
- repositories: versioning, search, download
- e.g. Docker Hub for Docker images

Software Encapsulation / 4

Container characteristics



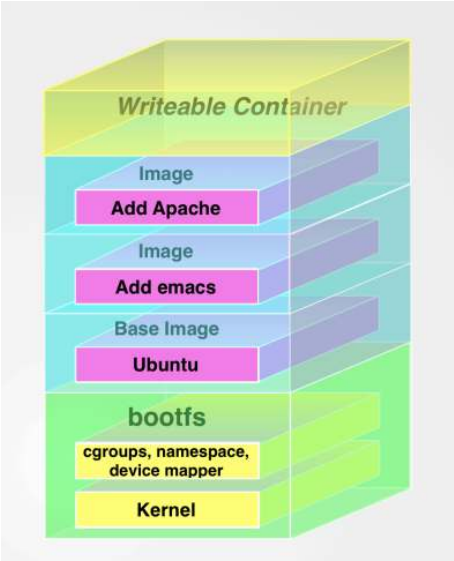
Limitation: OS requirements often not fulfilled on older hardware

Service Engineering Research Area
ICCLAB SPLAB

Software Encapsulation / 5

Container image creation with Dockerfile (specific to Docker)
Layered modifications

- e.g. base image with cloud/hardware-specific additions



[collabnix.com]

```
FROM ubuntu
MAINTAINER xxxx@zhaw.ch

RUN apt-get install -y emacs
RUN apt-get install -y apache

EXPOSE 80 10001

ADD files/micro-inetd /opt/

CMD [“/opt/micro-inetd“, “10001“, “/usr/games/cowsay“, “hello“]
```

Software Encapsulation / Practice

Using containers

Prerequisites:

- Docker engine (container runtime)
  - `https://docker.com`

Instructions:

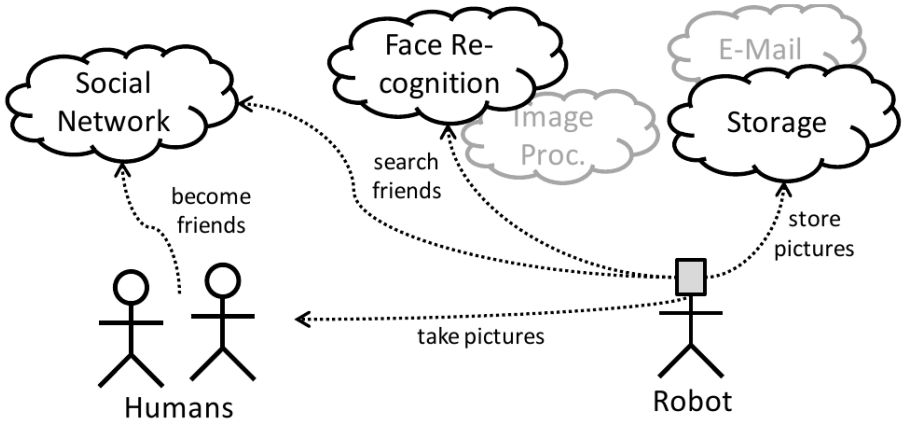
- Pull image with ROS components (may take a long time...)
  - on Docker Hub: `gtoff/rosmaster`
- Run image in interactive mode
- Launch ROS master node

Service Engineering Research Area
ICCLAB SPLAB

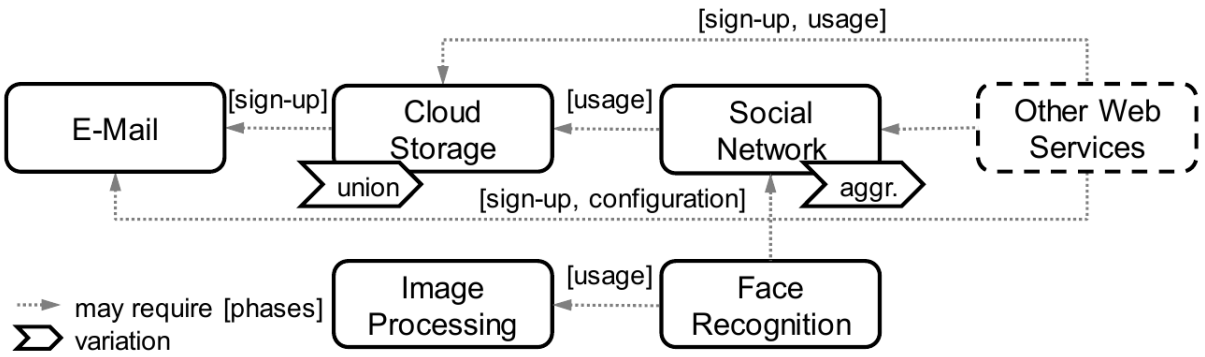
Autonomous Service Interaction / 1

How to connect autonomous systems with service-oriented environments?

Example: robots



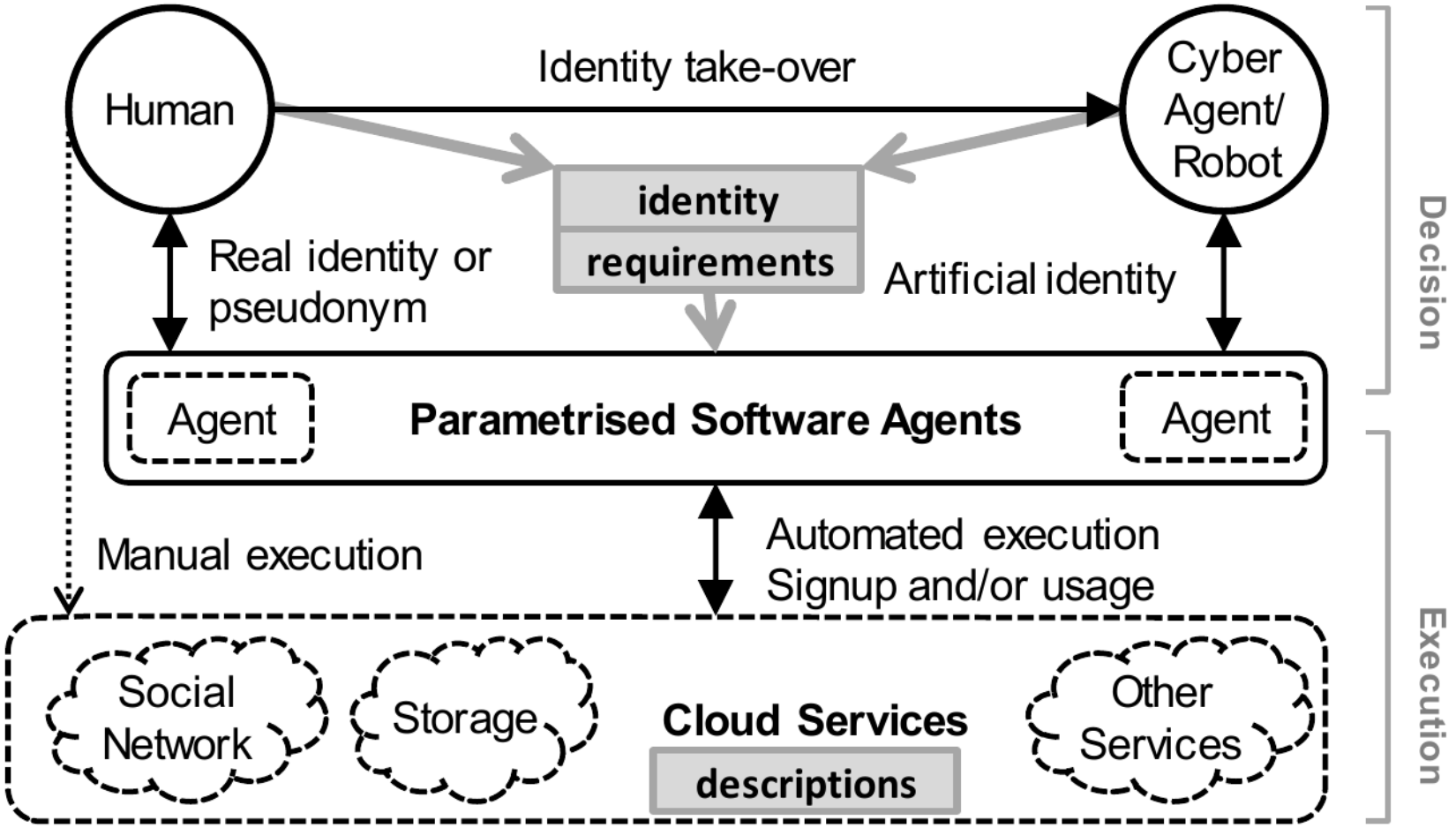
Requirement: identities & dependencies



Service Engineering Research Area
ICCLAB SPLAB

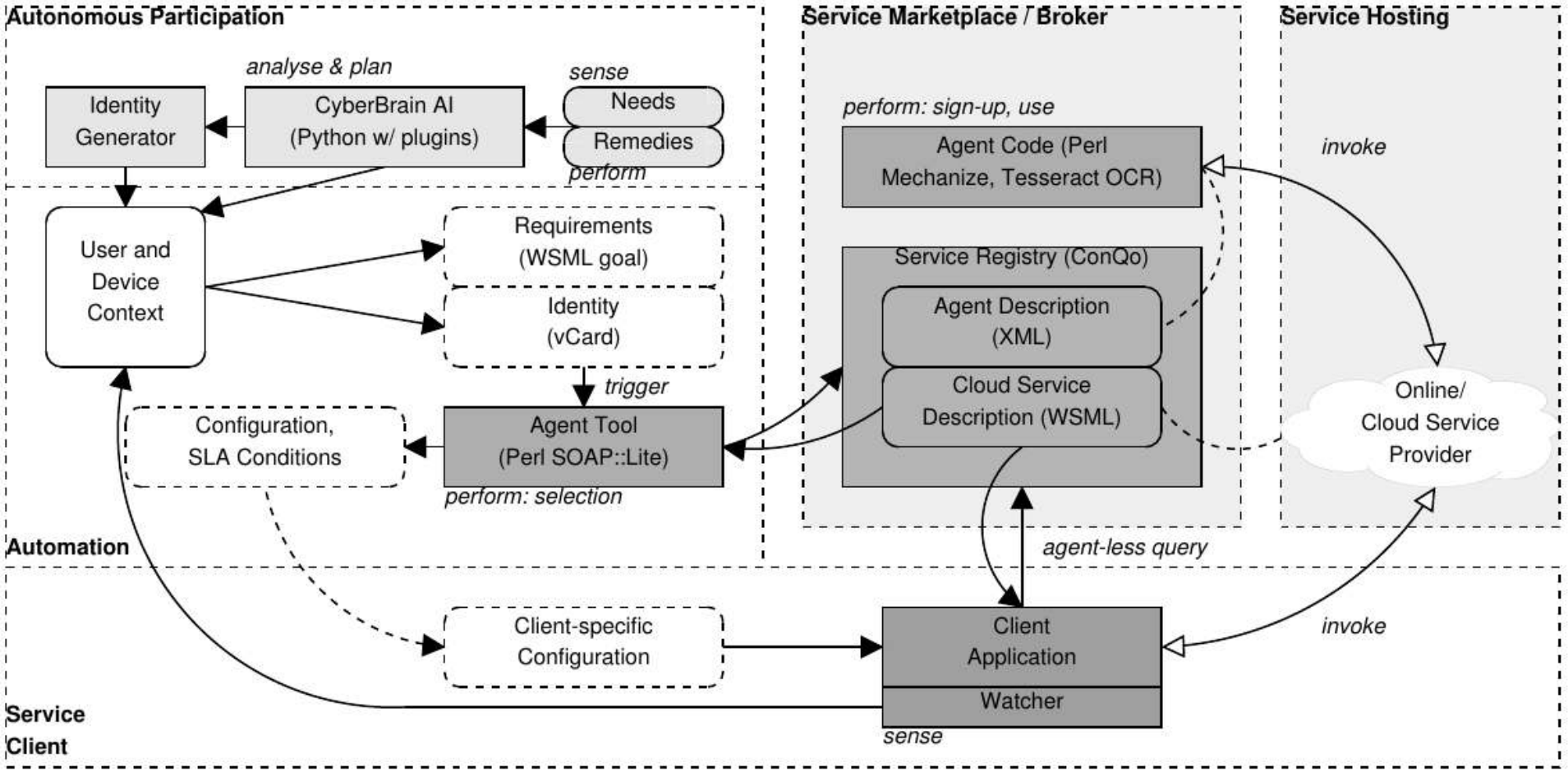
Autonomous Service Interaction / 2

AdAPtS Architecture: Advanced Autonomous Participation Scheme



Autonomous Service Interaction / 3

AdAPtS Implementation



Autonomous Service Interaction / Practice

Using automated signup

Prerequisites:

- OSST (Online Service Signup Tool)
  - `https://github.com/cloudmaster/osst`
- A bunch of Perl modules

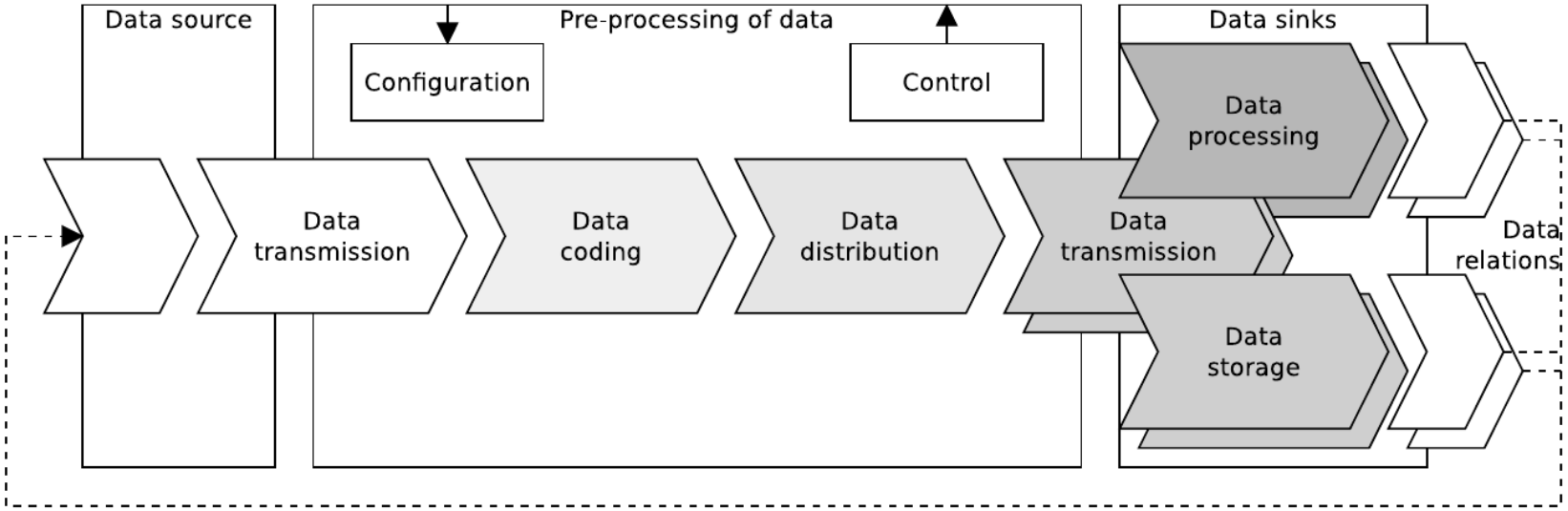
Instructions:

- study OSST README file for background information
- generate a vCard with a custom identity
- understand concepts of accounts, identities, service descriptions, agents and web automation

Controlled Resource Use / 1

In rugged environments, data should be processed with care.

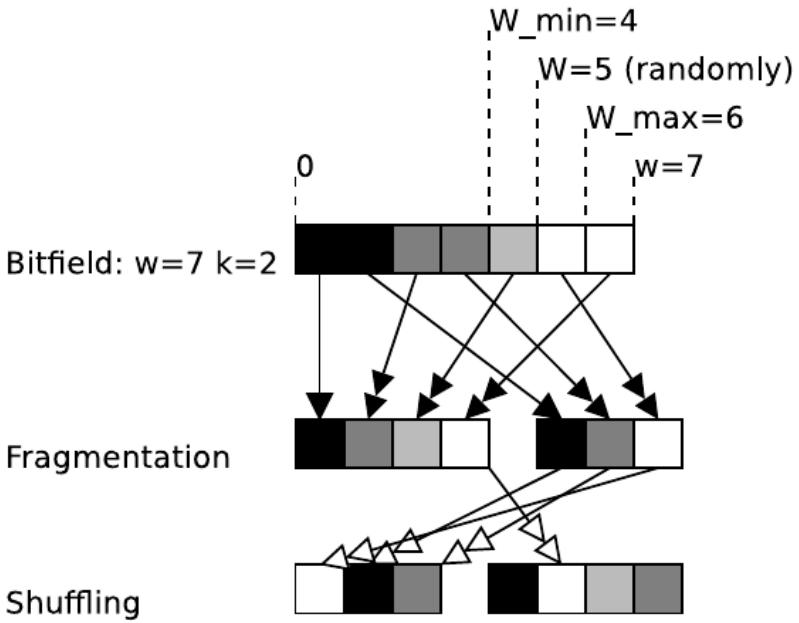
Key concerns: resource capacity, price, confidentiality, performance

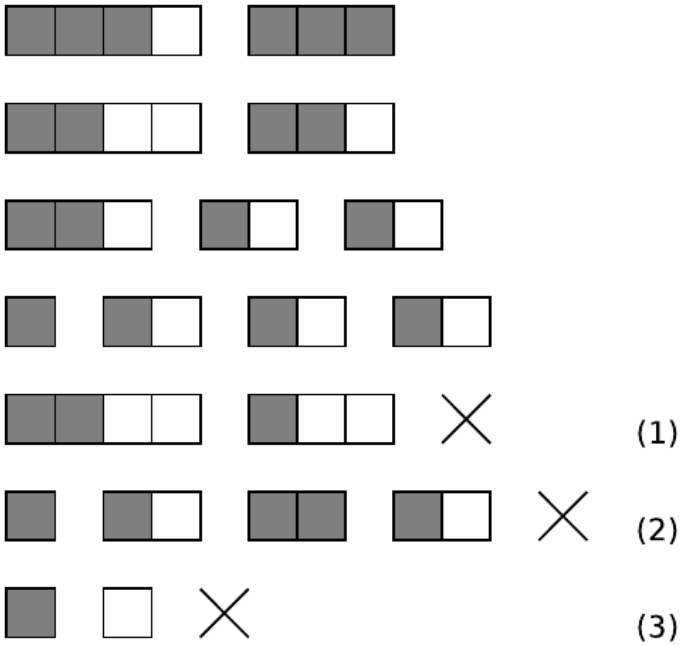


Service Engineering Research Area
ICCLAB SPLAB

Controlled Resource Use / 2

Example: Secret-sharing-aware bit expansion algorithm





Fixed assignment
Rotated assignment
Random assignment

Error cases:
(1) one bit too few set, Hamming weight wrongly determined
(2) one bit too much set, secret-sharing rule not honoured
(3) non-decidable variant and fragment without information

Service Engineering Research Area
ICCLAB SPLAB

Controlled Resource Use / 3

Data coding overview

Coding technique	Minimal fragment number $\tilde{k}$	Tolerated failures
Block creation/Chunking	$= k$	0
Erasure Coding	$= k$	m
Replication	= 1	m
Secret Sharing	$\geq k$	0 / m
Interpolation	$\leq k$	$k-1$
Bitsplitting	$= k$	1

Coding technique	Structure preservation	Recursion	Processing	Redundancy
Block creation/Chunking	non-deterministic	no	limited	0%
Erasure Coding	non-deterministic	no	limited	0-100%,...
Replication	yes	yes	yes	100%,200%...
Secret Sharing	no	no	no	0% / >0%
Interpolation	yes	no	yes	0%
Bitsplitting	partial	yes	partial	1x (50%,...)

Service Engineering Research Area
ICCLAB SPLAB

Controlled Resource Use / Practice

Using stealth databases

Prerequisites:

- StealthDB
  - `http://lab.nubisave.org/stealthdb/`

Instructions:

- Clone “dispersedalgorithms“ repository
- Run `db/stealthdb`
- Explore the unique commands:
  - `USE CLOUDS` and `ALTER TABLE ... USE CLOUDS`
  - `SELECT ... OPTIMIZE FOR`
  - `SELECT ... FOREVER`
- Create a dispersed in-memory table and perform some CRUD on it

Service Engineering Research Area
ICCLAB SPLAB

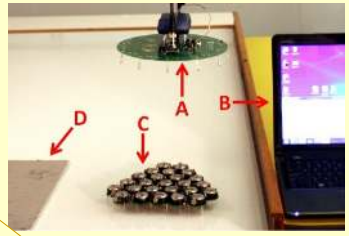
Summary

Cyber-Physical Applications based on diverse Cyber-Phyiscal Systems (cloud-connected sensors, robots, machines) remain a challenging research topic.

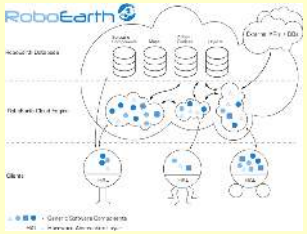
This talk has emphasised on:
- hardware abstraction / fleet management
- software encapsulation and portability
- redundancy exploitation and controlled traded-offs
- autonomous and autonomic participation management
- possible future platforms

Towards Platforms

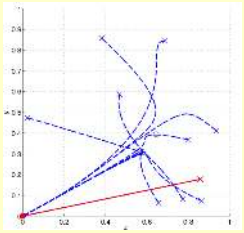
Coordination Platform



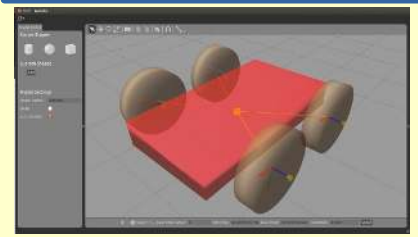
Collective

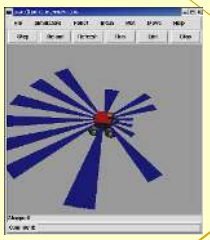


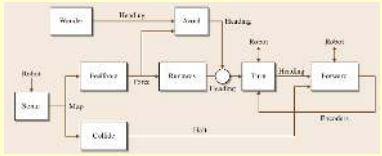
RoboEarth


Human-Swarm

Development Platform


Gazebo Model Editor


Pyro Trace

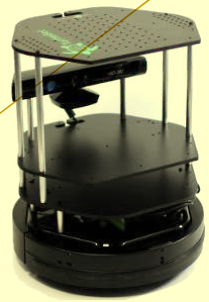

Subsumption

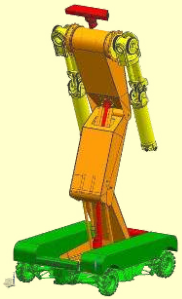
Runtime Platform


Lego Mindstorms

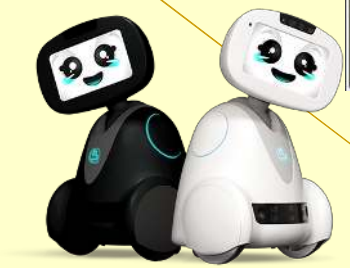

ROS

Hardware Platform


ROP Turtle

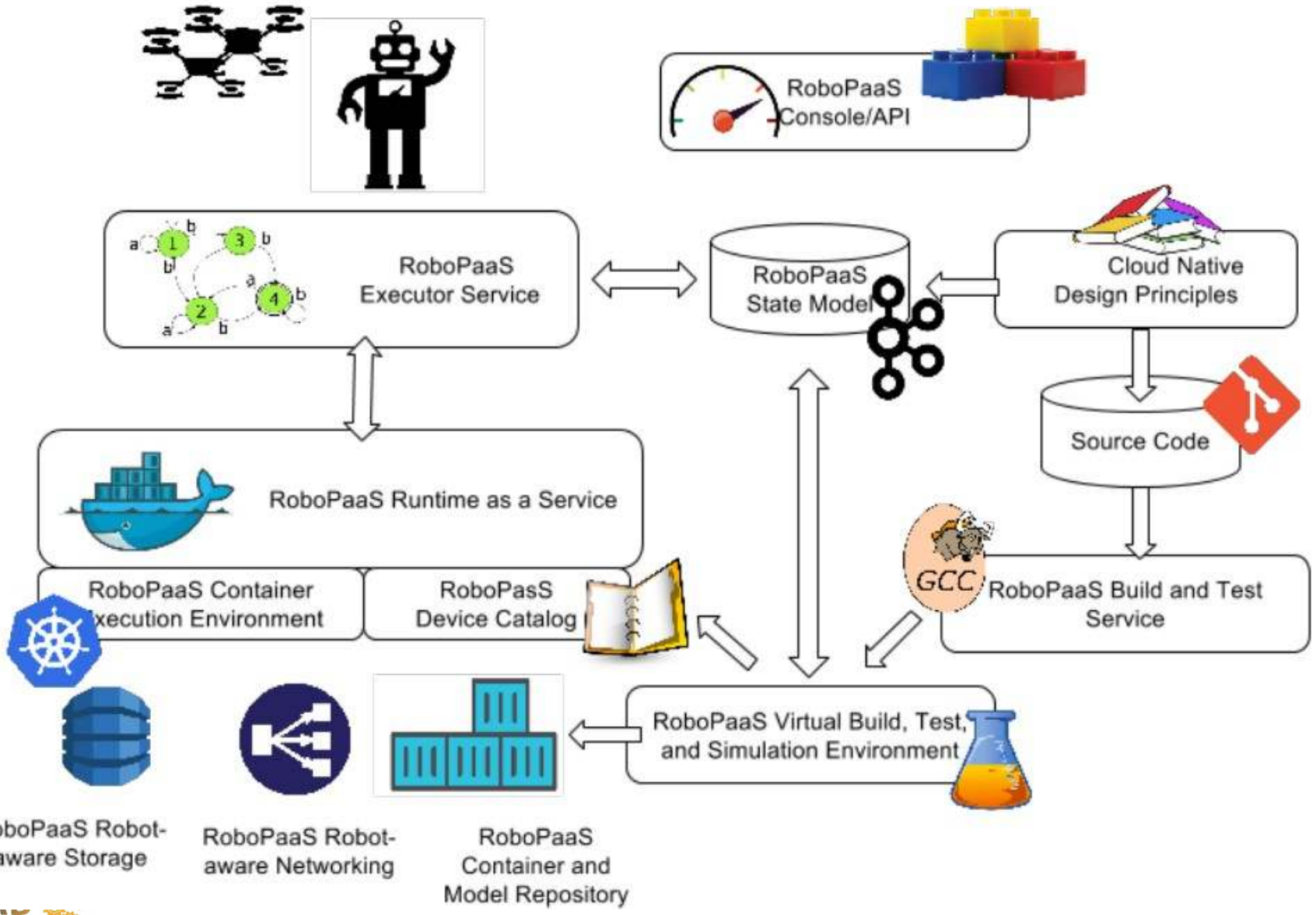

ROP Sergio


Harvard Kilobot


Bluefrog Buddy


NAOqi

Towards Platforms



Publications

J. Spillner and C. Piechnick and C. Wilke and U. Aßmann and A. Schill: *"Autonomous Participation in Cloud Services"*, 2nd International Workshop on Intelligent Techniques and Architectures for Autonomic Clouds (ITAAC), pp. 289-294, November 2012, Chicago, Illinois, USA.

J. Spillner: *"Secure Distributed Data Stream Analytics in Stealth Applications"*, 3rd IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), May 2015, Constanța, Romania.