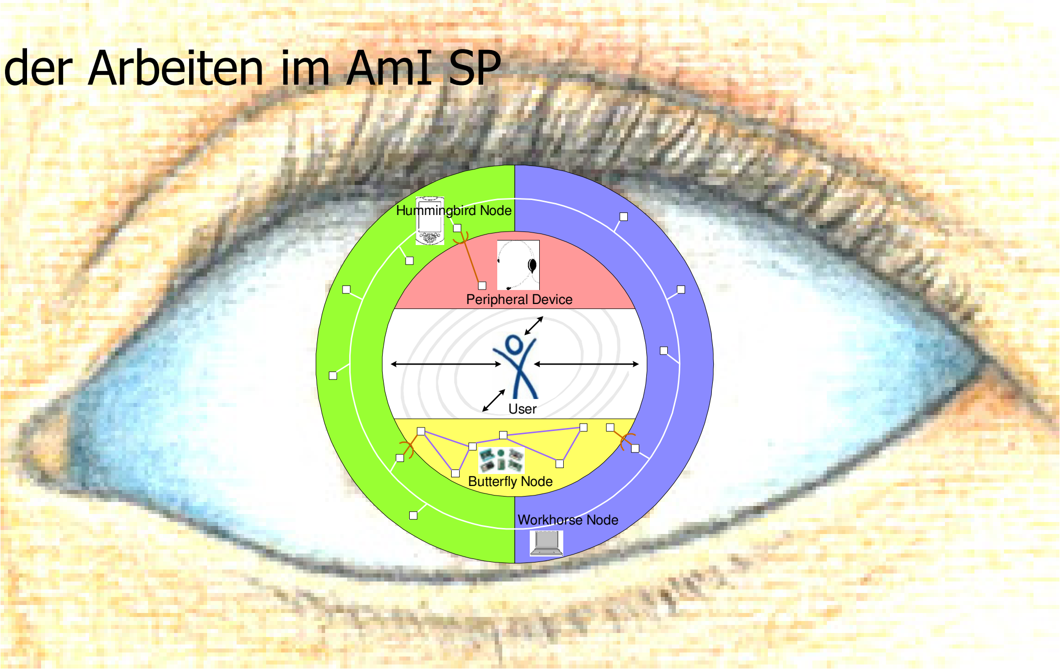
DAiSI –
Dynamic **A**dap**ti**ve **S**ystem **I**nfrastructure

Stand und Abschluss der Arbeiten im AmI SP



Andreas Rausch

Technische Universität Kaiserslautern
Fachbereich Informatik
AG Softwarearchitektur

Agenda

- Motivation
- Application Example
- Overview of DAiSI Architecture
- Implementing DAiSI components
- Status and Further Issues

Motivation – New Kinds of Systems like Ambient Intelligence, Persuasive Systems, Ubi…

- Those Systems
  - support us **unintrusively** but **ubiquitous**
    - different distributed devices
      (sensors, actors, interaction devices)
    - different networks
    - Integration of (unknown) devices to an AmI-system at runtime is hardly possible in our days
    - automatic adaption towards environment and user is important
      => **dynamic adpatation**
- E.g. „Ambient Intelligence refers to electronic environments that are sensitive and responsive to the presence of people.“
  - Scenarios:
    - Assisted Living, Working, Training…

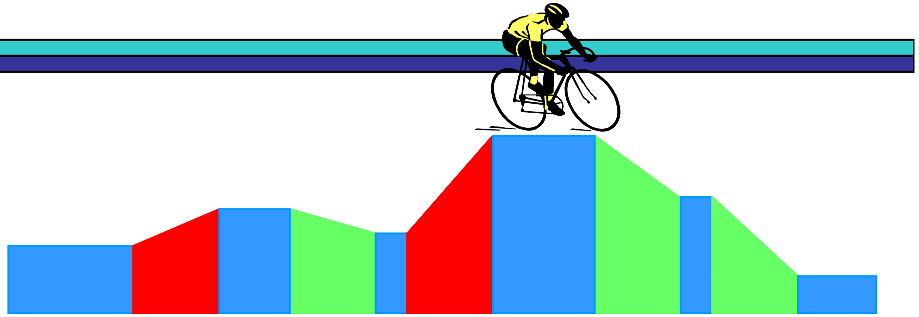
Outline

- Motivation
- Application Example
  - Assisted Bicycle Training System
  - Food Control System
- Overview of DAiSI Architecture
- Implementing DAiSI components
- Status and Further Issues

Application Example – Assisted Training (1)

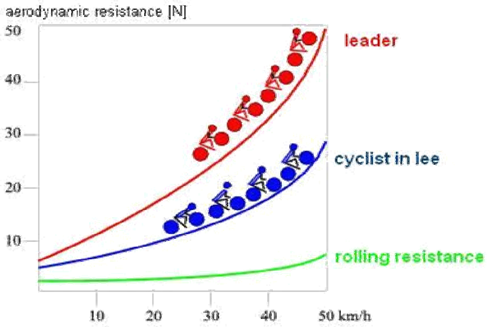
- Training group of (racing) cyclists
- Given track profile

Track Section
🟥 Uphill
🟦 Even
🟩 Downhill

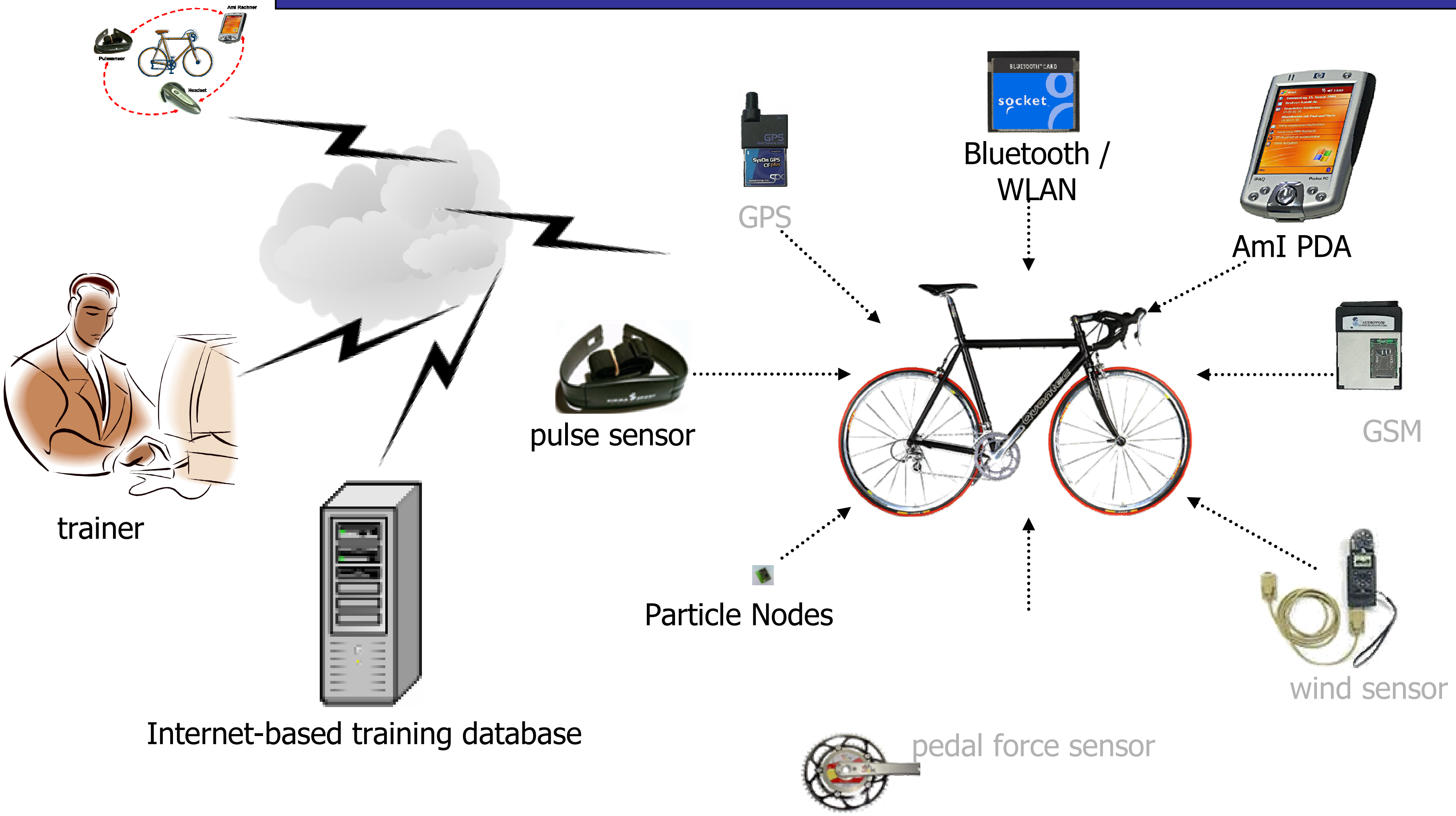


- Cyclists have different qualities regarding the track sections
- Each cyclist has a given overall physical condition and an individual training plan
- Training effect depends on the speed of the group and the position of the cyclist within the group formation

Main goal of the system
•Optimization of the single training effect
•Optimization of the group training effect



Application Example – Assisted Training (2)



AmI Rechner

Pulssensor

Headset

GPS

Bluetooth / WLAN

AmI PDA

pulse sensor

GSM

trainer

Particle Nodes

wind sensor

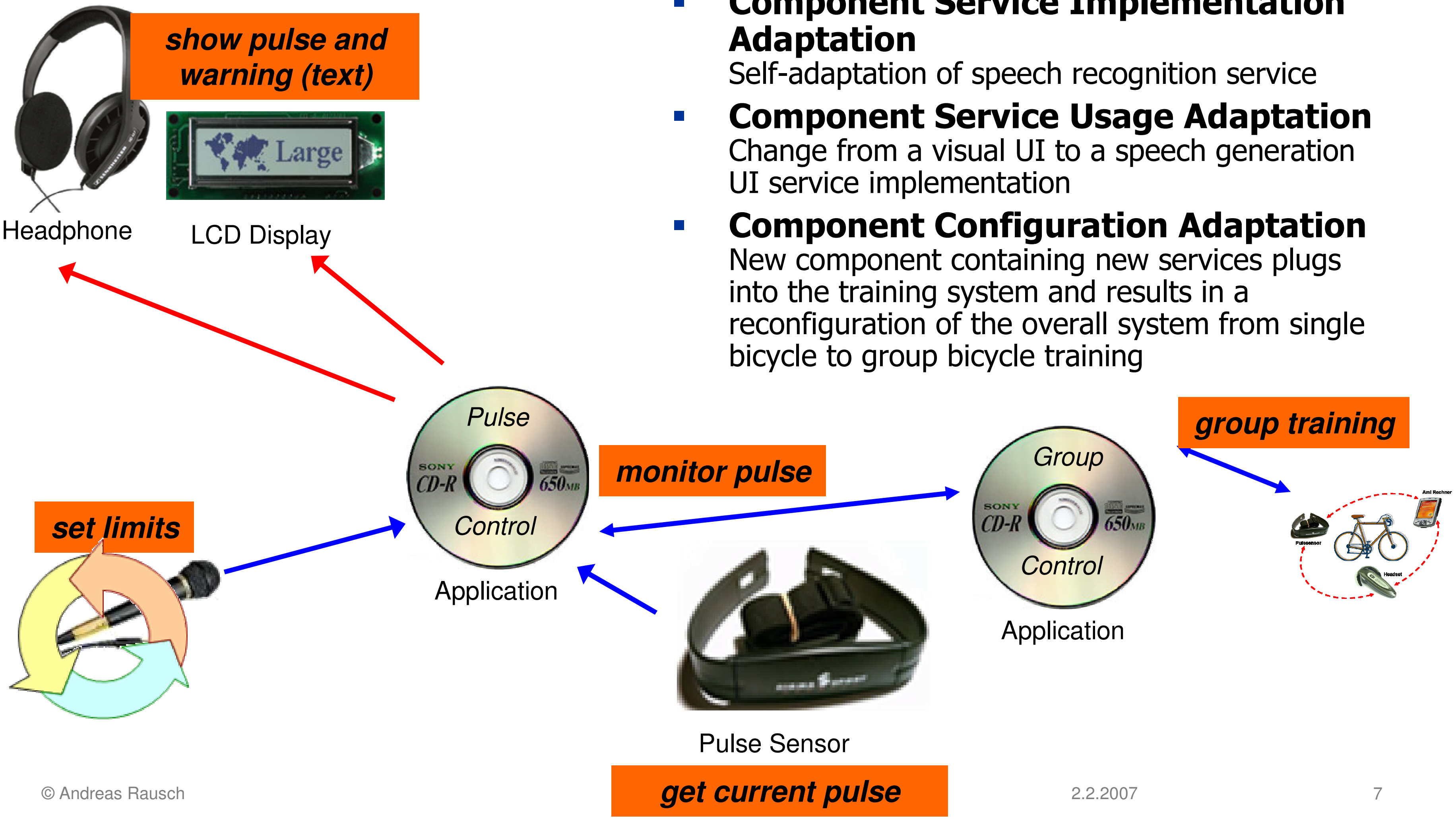
Internet-based training database

pedal force sensor

Application Example – Assisted Training (3)

(Currently) explored adaptation variants

- **Component Service Implementation Adaptation**
  Self-adaptation of speech recognition service
- **Component Service Usage Adaptation**
  Change from a visual UI to a speech generation UI service implementation
- **Component Configuration Adaptation**
  New component containing new services plugs into the training system and results in a reconfiguration of the overall system from single bicycle to group bicycle training



show pulse and warning (text)

Headphone

LCD Display

set limits

Pulse Control Application

monitor pulse

Group Control Application

group training

Pulse Sensor

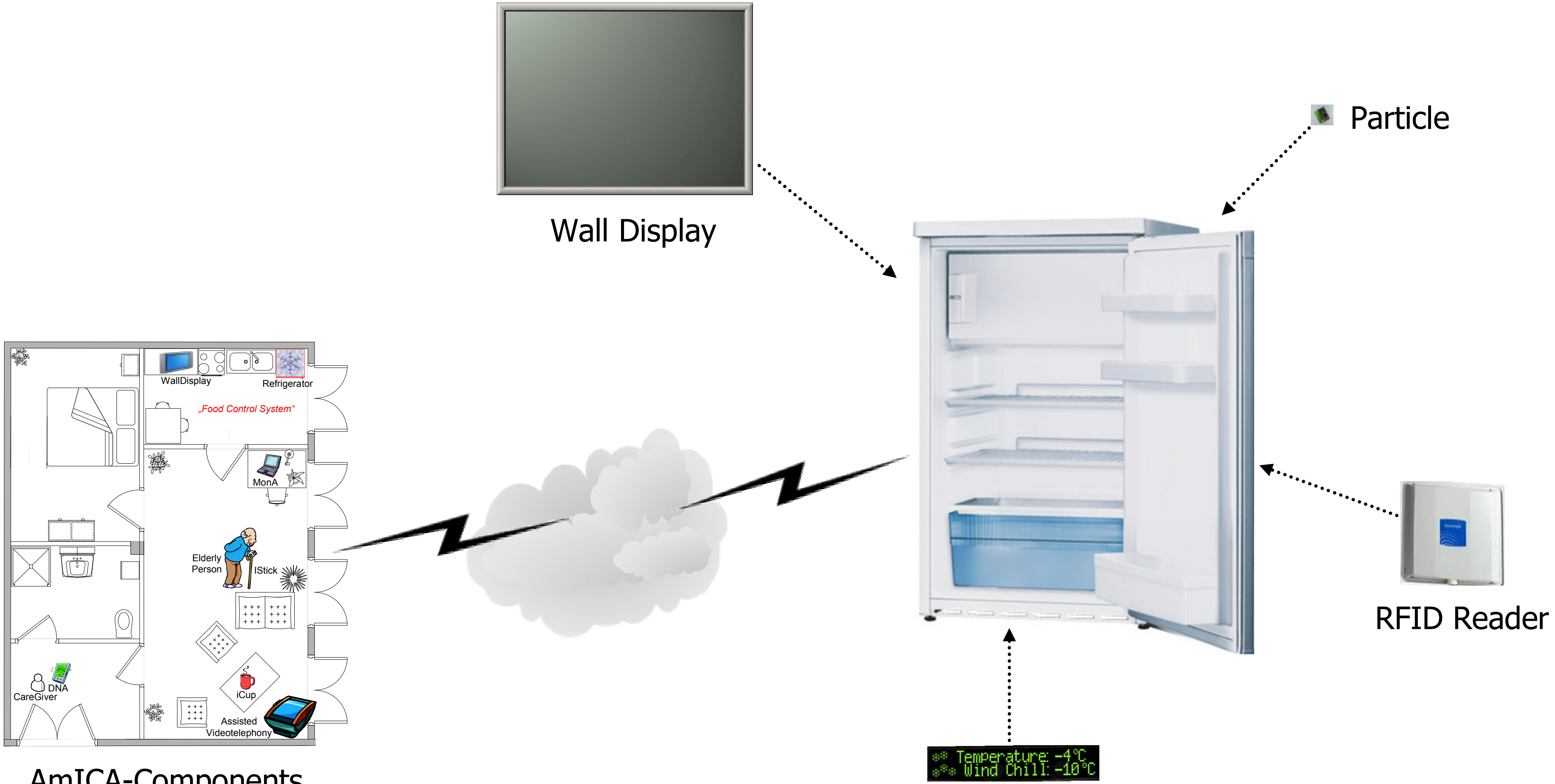
get current pulse

Application Example – Dynaptive Food Control System (1)

- Prevent that Elderly people eat spoilt food
  - Warn in case of refrigerator malfunction
  - Warn if expired food is inside the refrigerator

- Provide comfort features
  - Offer recipes (context-sensitive)

Application Example – Dynaptive Food Control System (2)



Wall Display

Particle

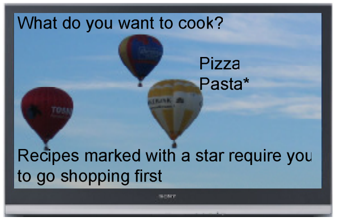
RFID Reader

AmICA-Components

LCD Display

Application Example – Dynaptive Food Control System (3)

- **Component Configuration Adaptation**
  Integration of an RFID Reader
- **Component Service Usage Adaptation**
  Integration of a Wall Display
- **Component Service Implementation Adaptation**
  Self-adaptation of Recipe Service



What do you want to cook?

Pizza
Pasta*

Recipes marked with a star require you to go shopping first

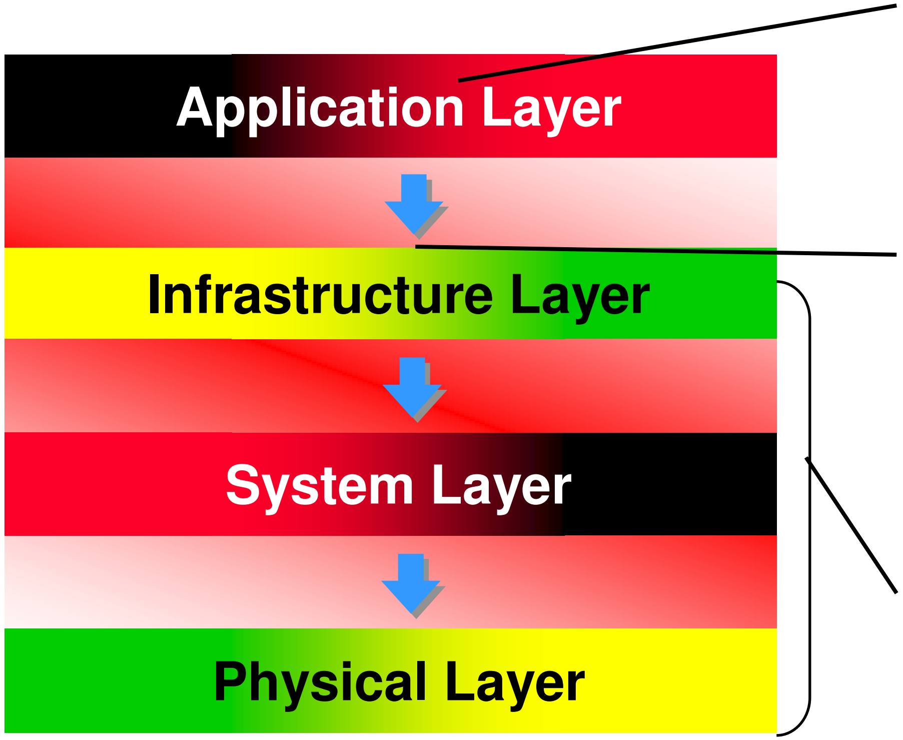
LCD-Display
RFID-
Reader

Outline

- Motivation
- Application Example
- Overview of DAiSI Architecture
  - Physical Layer
  - System Layer
  - Infrastructure Layer
  - Application Layer
- Status and Further Issues

Overview of DAiSI Architecture

DAiSI supports engineers to develop (resp. design) **adaptive and distributed** AmI systems by **encapsulating** the **heterogeneous** (mobile, and embedded) environment as well as the **underlying physical and system layer**.
-> focus on application not on technology

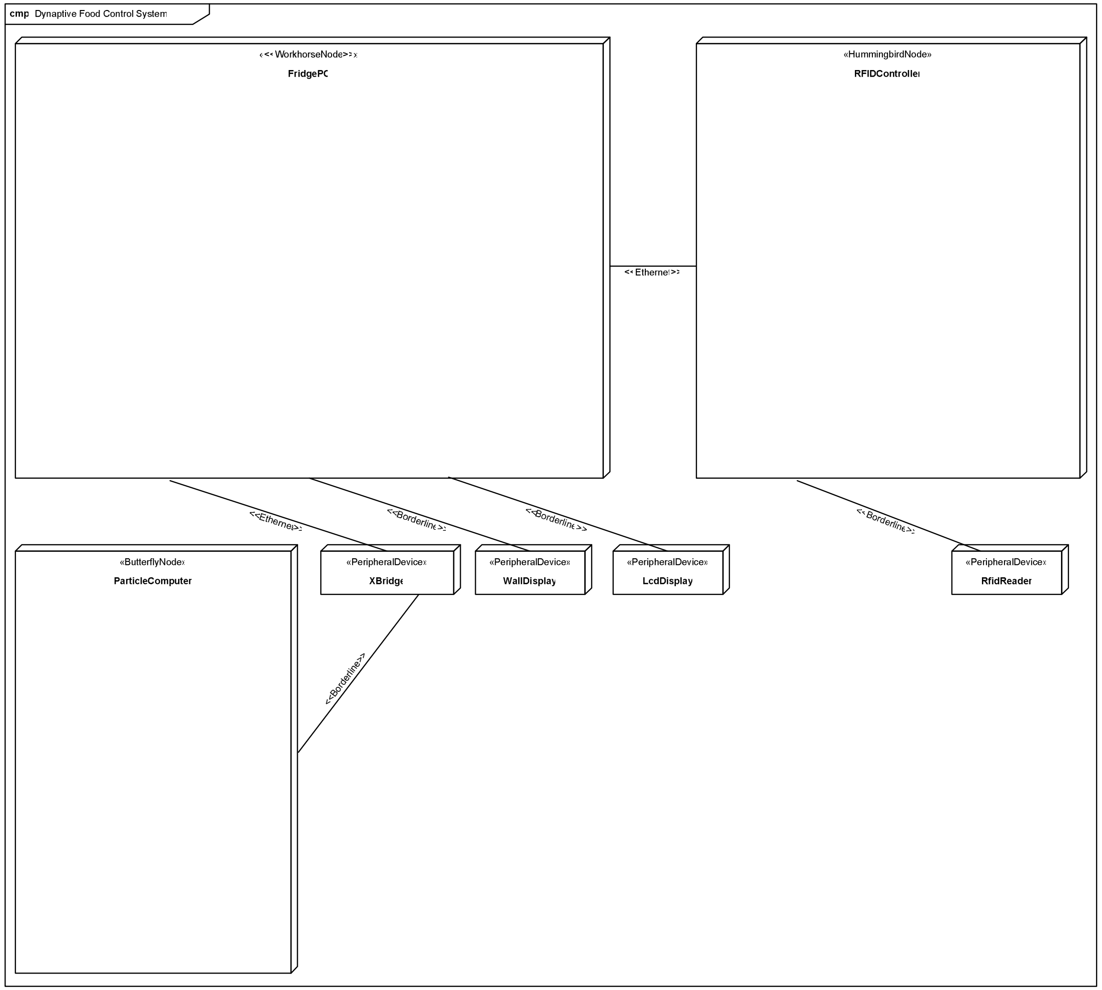


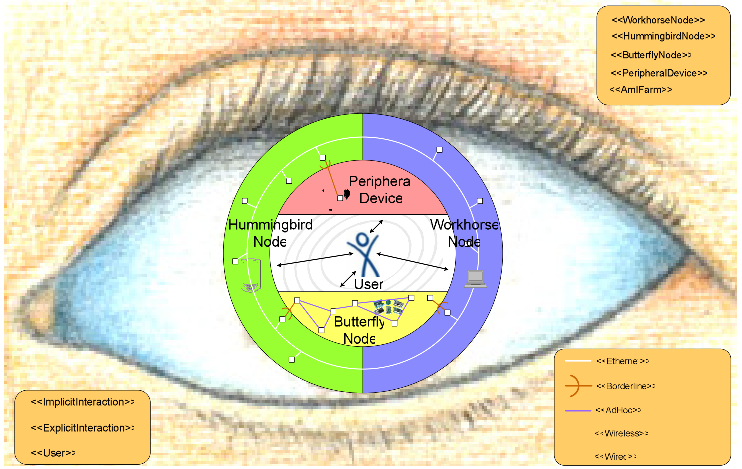
Architecture Once, Develop Any-AmI
DAiSI provides a set of architectural guidelines to develop AmI components and systems on top of DAiSI

Write Once, Run Anywhere
DAiSI defines a set of interfaces (models) that can be used by programmers (resp. designers) to develop (design) AmI components and systems

Reference Once, Allow Any-Solution
DAiSI offers a single reference implementation of the defined interfaces (models) with respect to a restricted system and physical layer. Other (better) solutions may be provided by others

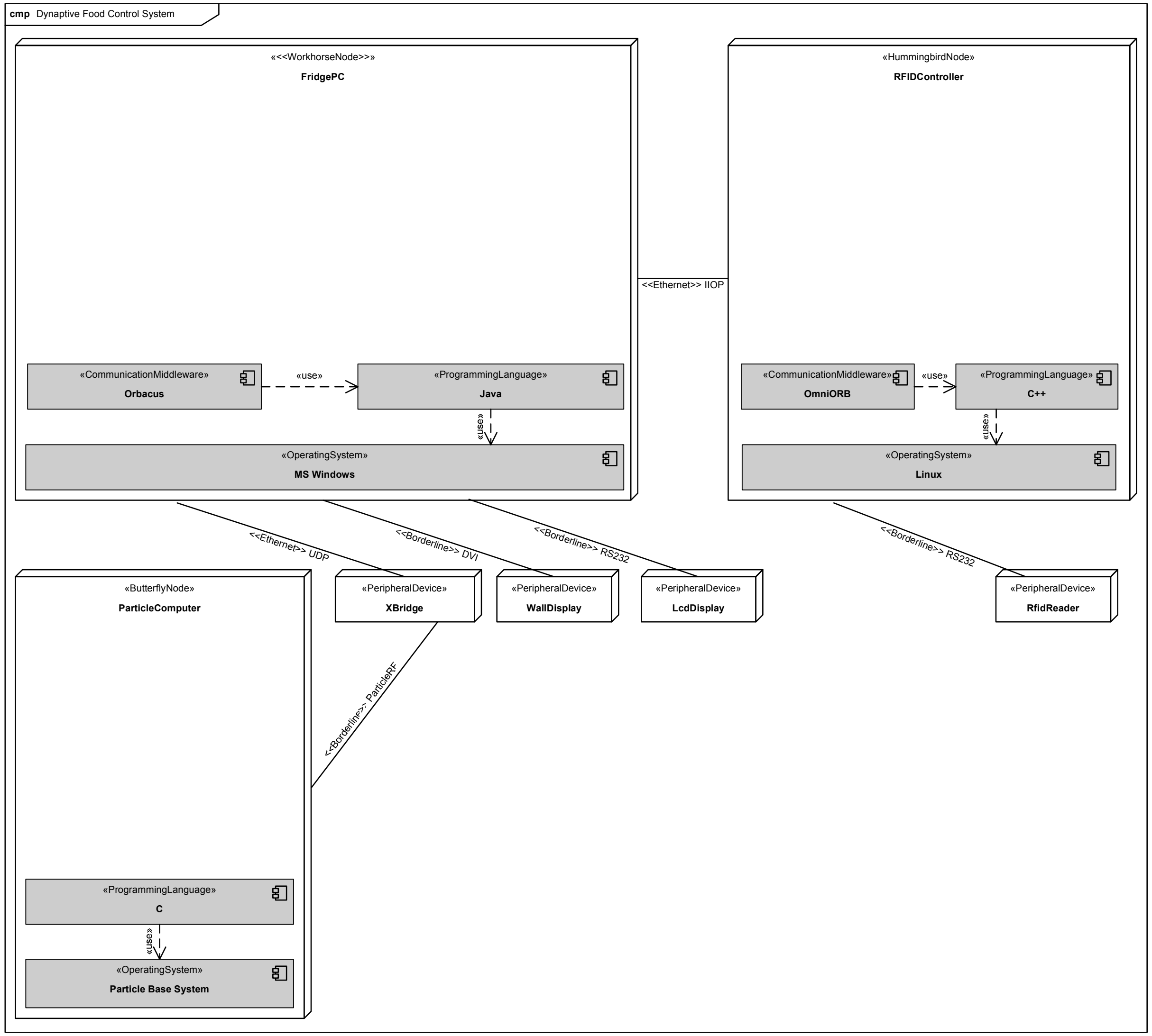
Physical Layer



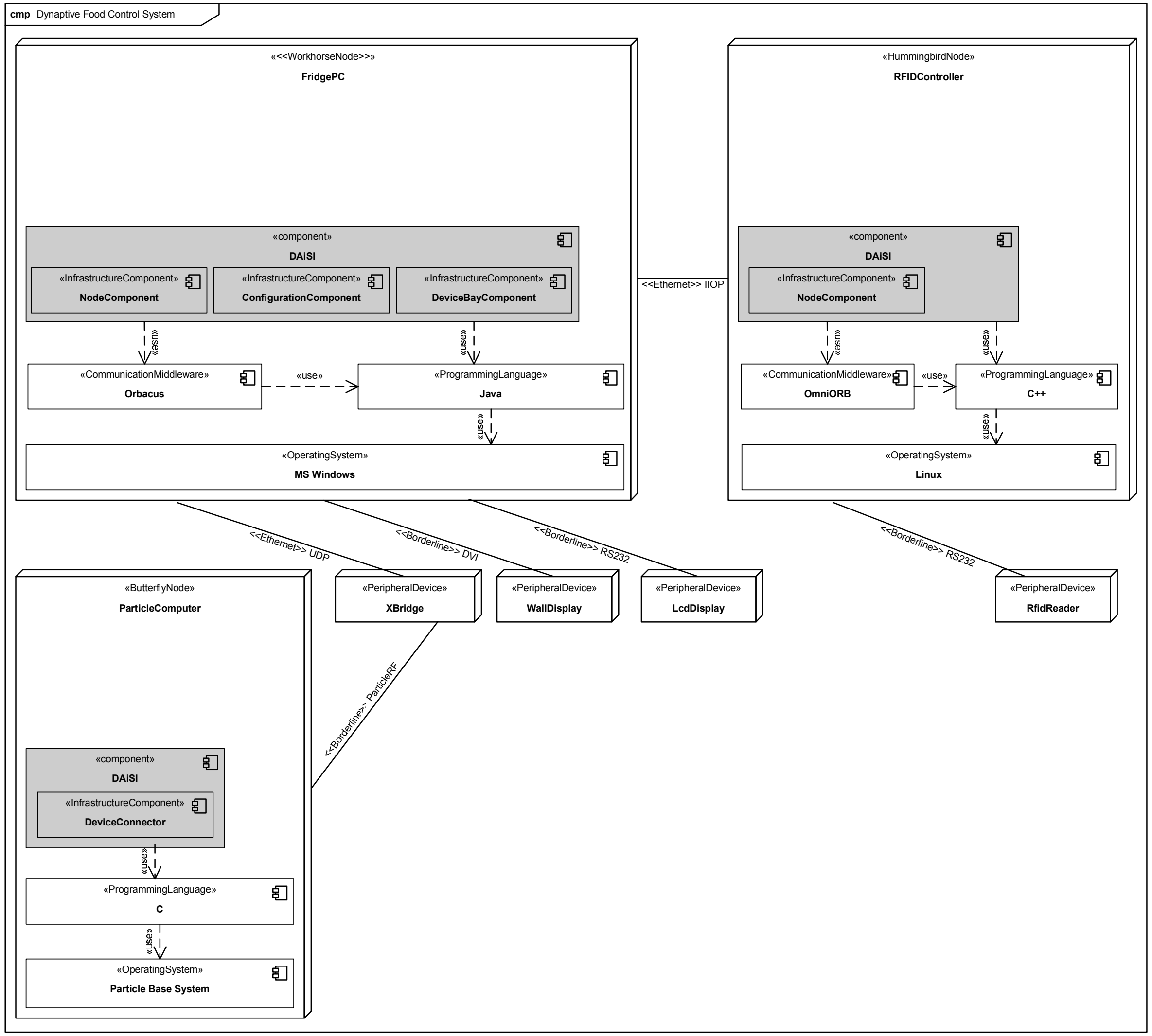


- Communication Types:
  - Ethernet
  - Point to Point
  - Ad hoc

System Layer



Infrastructure Layer (1)



cmp Dynaptive Food Control System

«<<WorkhorseNode>>»
FridgePC

«component»
DAiSI

«InfrastructureComponent»
NodeComponent

«InfrastructureComponent»
ConfigurationComponent

«InfrastructureComponent»
DeviceBayComponent

«use»

«use»

«CommunicationMiddleware»
Orbacus

«use»

«ProgrammingLanguage»
Java

«use»

«OperatingSystem»
MS Windows

<<Ethernet>> IIOP

«HummingbirdNode»
RFIDController

«component»
DAiSI

«InfrastructureComponent»
NodeComponent

«use»

«use»

«CommunicationMiddleware»
OmniORB

«use»

«ProgrammingLanguage»
C++

«use»

«OperatingSystem»
Linux

<<Ethernet>> UDP

<<Borderline>> DVI

<<Borderline>> RS232

<<Borderline>> RS232

«ButterflyNode»
ParticleComputer

«PeripheralDevice»
XBridge

«PeripheralDevice»
WallDisplay

«PeripheralDevice»
LcdDisplay

«PeripheralDevice»
RfidReader

<<Borderline>>: ParticleRF

«component»
DAiSI

«InfrastructureComponent»
DeviceConnector

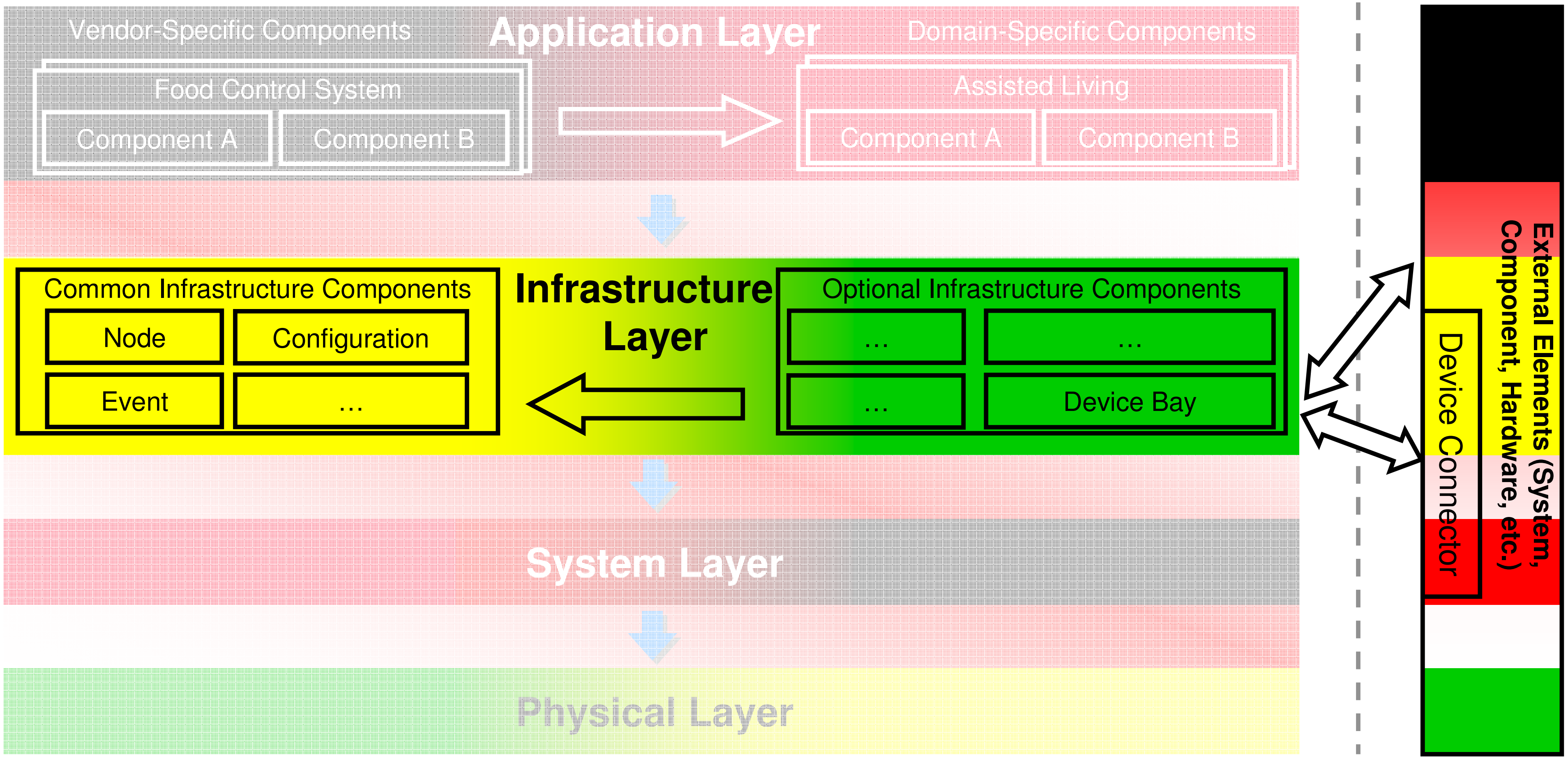
«use»

«ProgrammingLanguage»
C

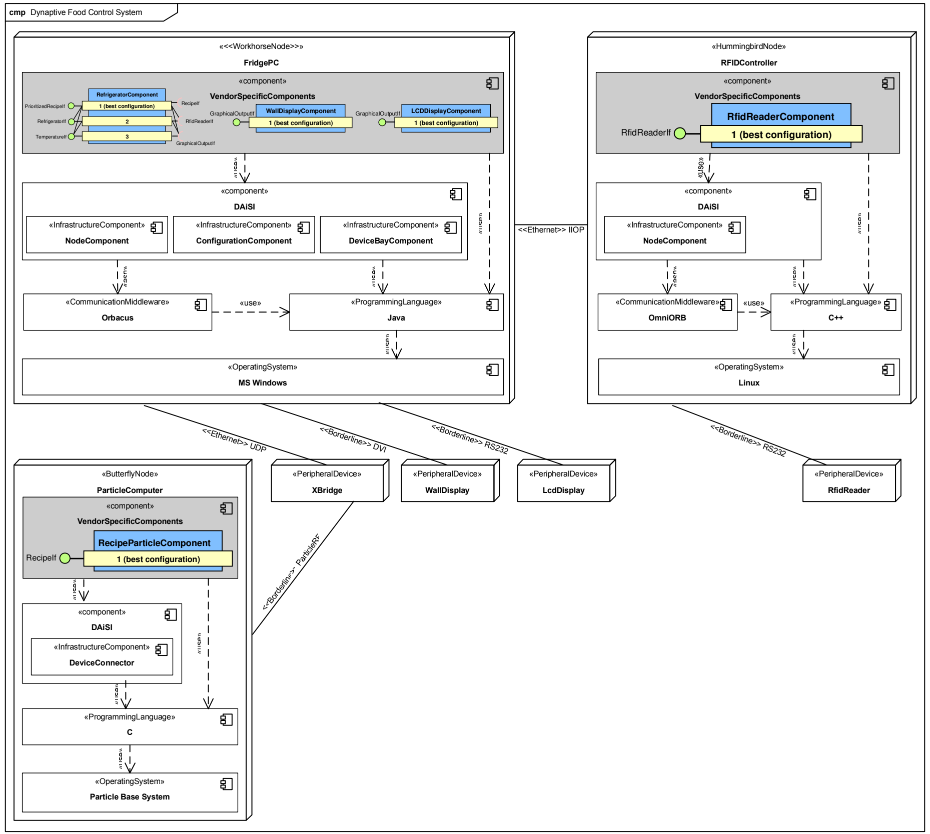
«use»

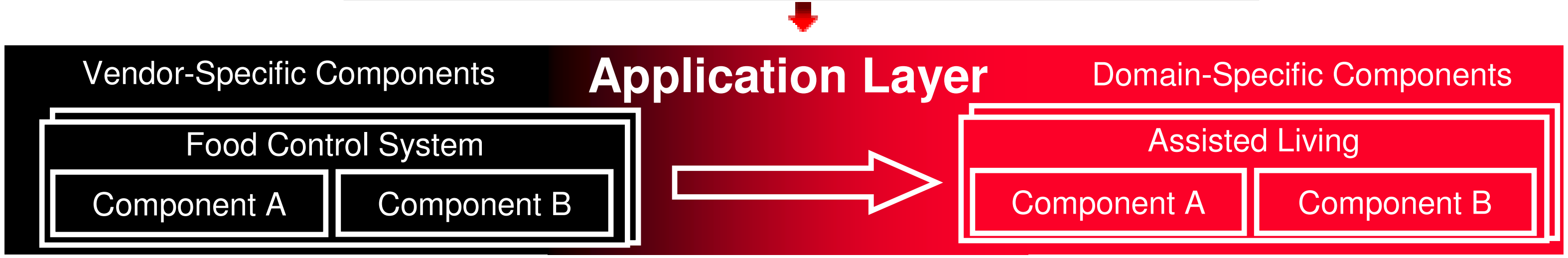
«OperatingSystem»
Particle Base System

Infrastructure Layer (2)



Application Layer




Vendor-Specific Components

Application Layer

Domain-Specific Components

Food Control System

Component A

Component B

Assisted Living

Component A

Component B

Outline

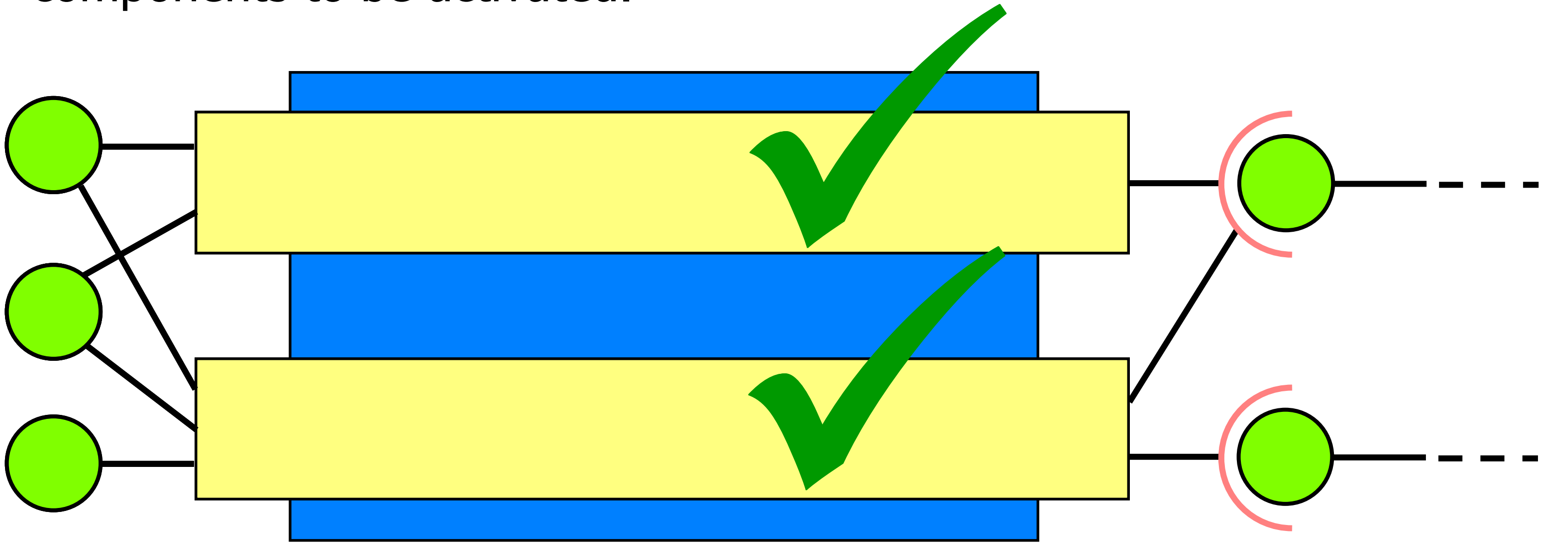
- Introduction
- Application Example
- Overview of DAiSI Architecture

➡ Implementing DAiSI components

- Status and Further Issues

DoAmI‘s Service-oriented Architecture Approach

- Each AmI system is composed out of **service components**
- Each service components supports an **ordered set of configurations**
- Each configuration
  - **provides** a set of **service interfaces** and therefore
  - **requires** a set of **service interfaces** from other service components to be activated.

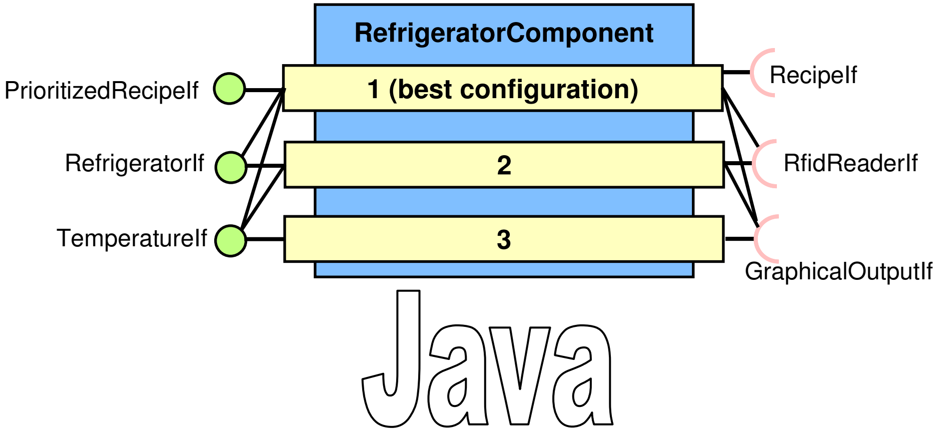


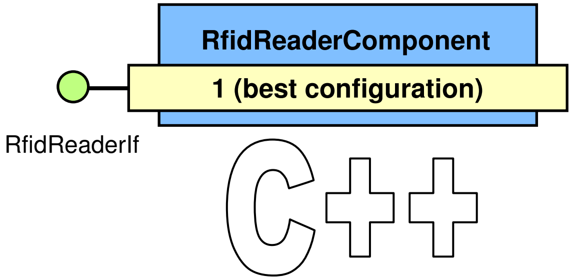
Source: DoAmI Architect Workshop April 2006

Implementing DAiSI Components

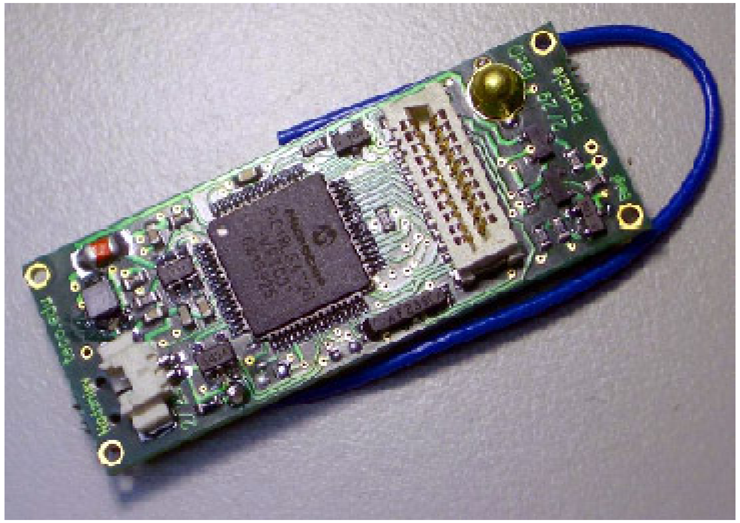
- DAiSI provides a Framework for Component Developers
- Two versions exist
  - Java
  - C++ (Limited)

- Two Examples





The Particle Computer System

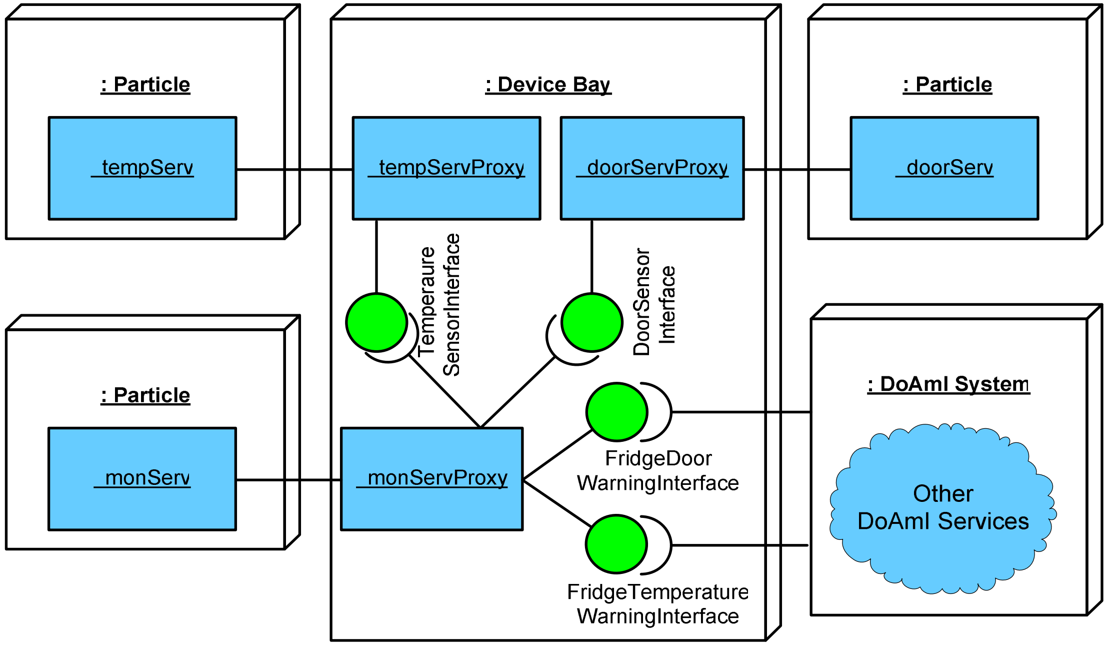


http://particle.teco.edu

- Processor:
  - PIC at 20 MHz, 128kb Flash, 4kb RAM, 1kb EEPROM
- RF Communication:
  - Through RFM TR1001, 125kbit bandwidth, 868.35 MHz
- Interface:
  - 21 pin multi purpose connector with I2C, SPI, serial + parallel Bus, Interrupt input lines, analog + digital I/O lines
- Sensors:
  - Acceleration Sensor, Force and Pressure Sensor, Light Sensor, Temperature Sensor, Microphone, Real Time Clock
- Networking:
  - Distributed peer to peer, ad-hoc, no master-node needed
- Size:
  - 45x18 mm (45x27 mm with AAA battery)
- Price: 120,- Eur

→ Very limited resources!

Example – Fridge Integration into DoAmI

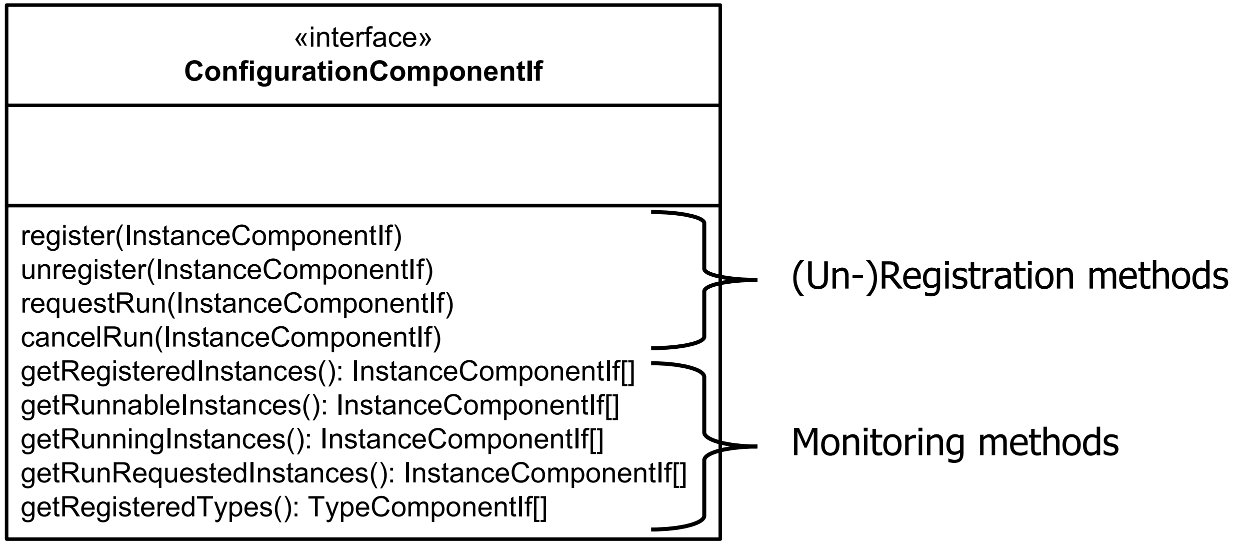


- Fridge Monitoring Service on one Particle using:
  - DoorService (provides *DoorSensorInterface*) on second Particle
  - TempService (provides *TemperatureSensorInterface*) on third Particle

→ Provides *FridgeDoor-* and *FridgeTemperatureWarningInterface* for other DoAmI-Services

Configuration Component

- Heart-Beat of DAiSI



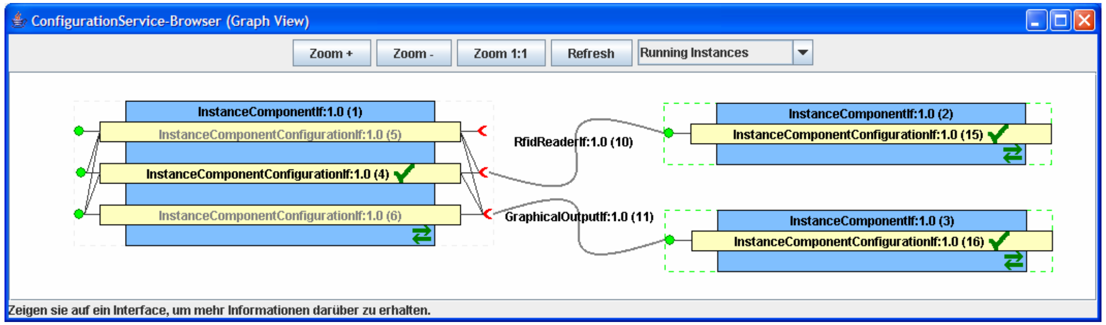
«interface»
ConfigurationComponentIf

register(InstanceComponentIf)
unregister(InstanceComponentIf)
requestRun(InstanceComponentIf)
cancelRun(InstanceComponentIf)
getRegisteredInstances(): InstanceComponentIf[]
getRunnableInstances(): InstanceComponentIf[]
getRunningInstances(): InstanceComponentIf[]
getRunRequestedInstances(): InstanceComponentIf[]
getRegisteredTypes(): TypeComponentIf[]

(Un-)Registration methods

Monitoring methods

- Basic Monitoring implemented in Configuration Browser:

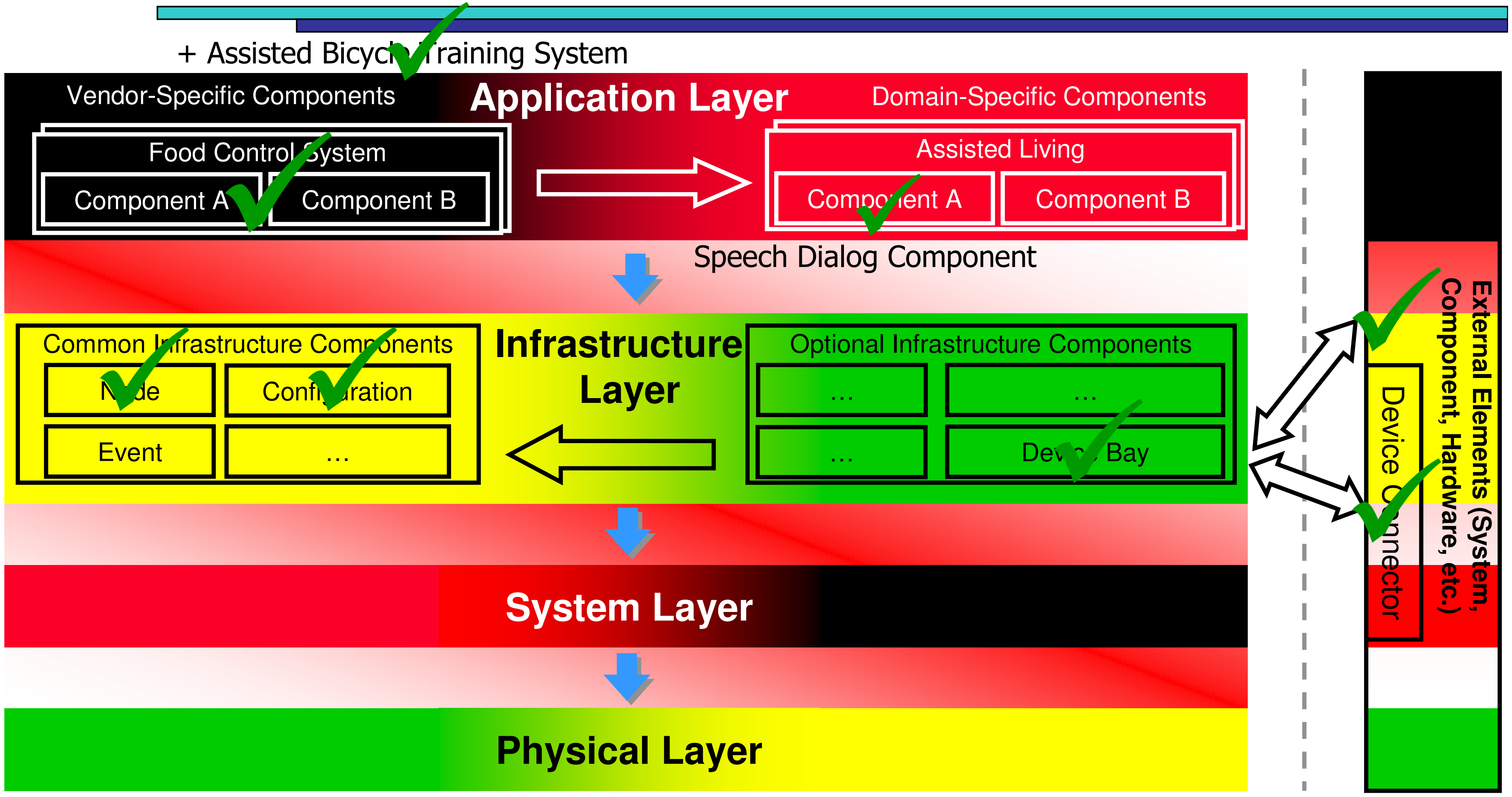

ConfigurationService-Browser (Graph View)
Zoom +
Zoom -
Zoom 1:1
Refresh
Running Instances
InstanceComponentIf:1.0 (1)
InstanceComponentConfigurationIf:1.0 (5)
InstanceComponentConfigurationIf:1.0 (4)
InstanceComponentConfigurationIf:1.0 (6)
RfidReaderIf:1.0 (10)
GraphicalOutputIf:1.0 (11)
InstanceComponentIf:1.0 (2)
InstanceComponentConfigurationIf:1.0 (15)
InstanceComponentIf:1.0 (3)
InstanceComponentConfigurationIf:1.0 (16)

Zeigen sie auf ein Interface, um mehr Informationen darüber zu erhalten.

Outline

- Motivation
- Application Example
- Overview of DAiSI Architecture
- Implementing DAiSI components
- Status and Further Issues

Conclusion: Current state



+ Assisted Bicycle Training System

Vendor-Specific Components

Application Layer

Domain-Specific Components

Food Control System

Component A

Component B

Assisted Living

Component A

Component B

Speech Dialog Component

Common Infrastructure Components

Node

Configuration

Event

…

Infrastructure Layer

Optional Infrastructure Components

…

…

…

Device Bay

System Layer

Physical Layer

External Elements (System, Component, Hardware, etc.)

Device Connector

Further Issues (1)

- C++ Implementation bidirectional
  - Planned for a future DAiSI Release
- Event Component supporting Inheritance for Events
  - First Prototype, no DAiSI Integration yet
  - Planned for a future DAiSI Release
- Test of the Correctness of the dynamic Integration
  - Planned for a future DAiSI Release

Further Issues (2)

- A Component requires (or offers) multiple Services of the same type
  - Each requires / provides may be a set of services respectively service references?
- A component crashs
  - First concepts for failure detection sketched and sent around within G6
- Quality of Service
  - Task force exists together with G2

Thank you for your attention

Any Questions