Webapps

Make use of web applications at DESY cryogenic controls dept.

Joerg Penning
EPICS-Conference at ITER June 2019

HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

DESY.

Overview

1 Motivation

2 Functional requirements

- Asset management
- Displaying archived data
- Displaying live data

3 Realization

- Implementation
- Tools
- Deployment

Motivation

Exchange of ideas

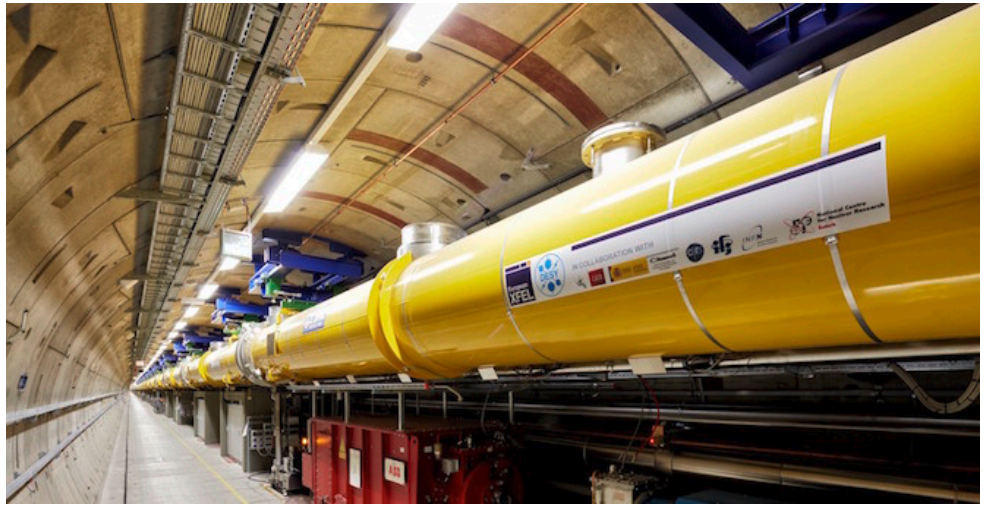
- We will be happy to learn what others do and tell what we do …
- Are we able to participate in an open source community?

Background

- Cryogenic controls
  i.e. IT infrastructure for all cryogenics
- Support for facilities group

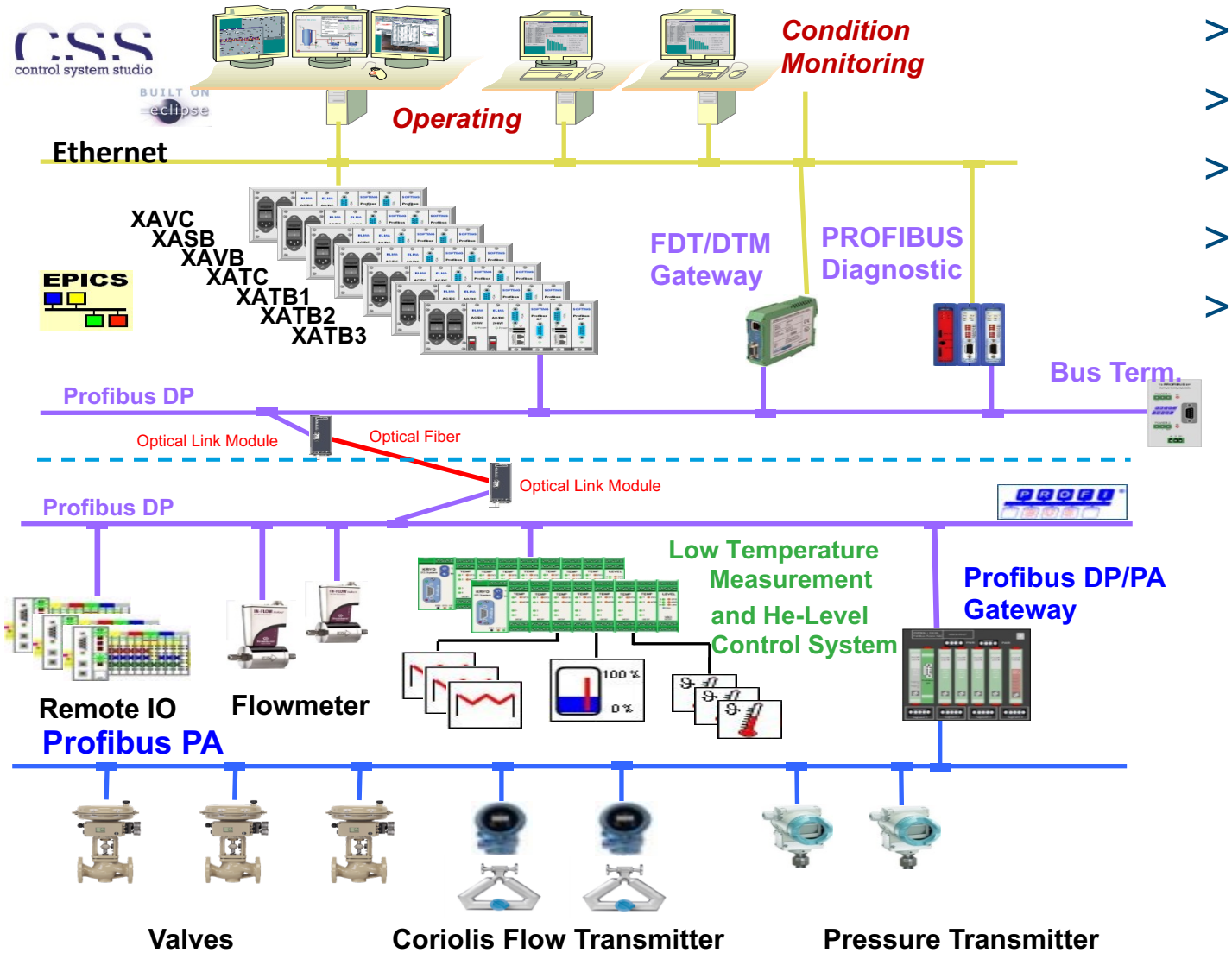
Open Source

- Of course



Motivation

A lot of different assets



- > 650 Valves
- > 2500 Temp. Sensors
- > 800 Press. Sensors
- > 100 Level Sensors
- > 200 Flow Sensors

Functional requirements

Should be accessible without further preparation, i.e. on the smart phone

Asset management

- Repair
- Stock
- Cost units

Archive data retrieval

- Watch trends when you are out-of-office on-site or on-call

Probe-style live data

- Find out about some record when you are out-of-office

OPI-style data display

- Watching the energy consumption
- Gained attention quickly at DESY

Asset management

A defective part must be exchanged

- Exchange part
  - Where are the spare parts?
  - Which procedure must be followed?
  - Which control system processes will be affected?
- Handle the defective part
  - Do we still have warranty?
  - Where to send to for repair?
  - Do we have to fill up stock?
- Take precautionary measure
  - Do other components of this type have problems too?
  - Where are they built in?

Asset management

Make use of heterogeneous databases

Asset-Database

- Manages orders and stock
- Enhanced to trace status (location)
- Enhanced to contain documentation
- IDs: Asset-ID, Inventory-ID

DCT-Database (Database Creation Tool)

- ID:s IOName, Channelname
- Side effect of creating EPICS-Databases

IO-Configuration-Database (‘Device Database’)

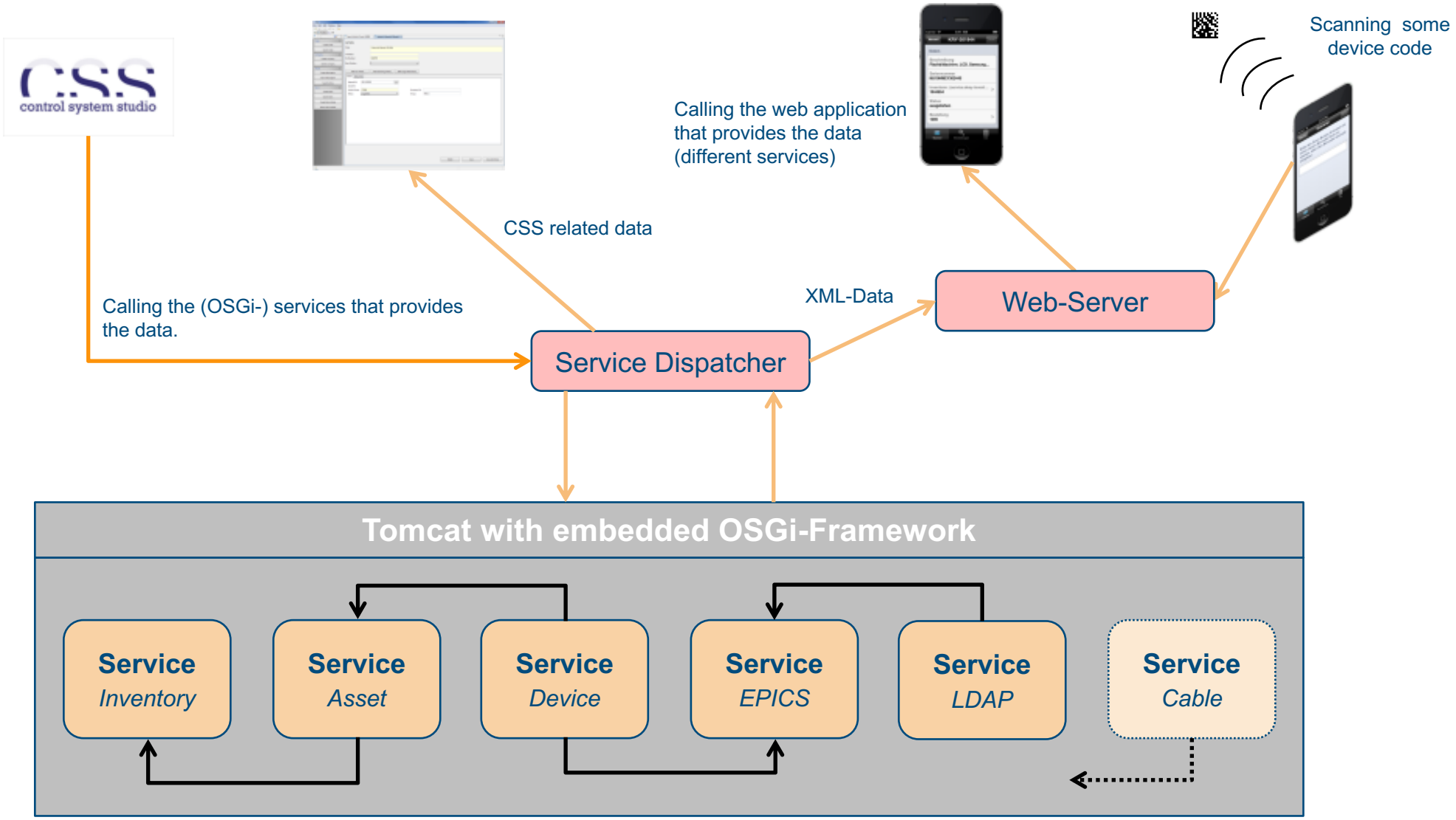
- IDs: IOName, Asset-ID
- Contains Profibus-Addresses, Documentation

DESY Inventory

- ID: Inventory-ID

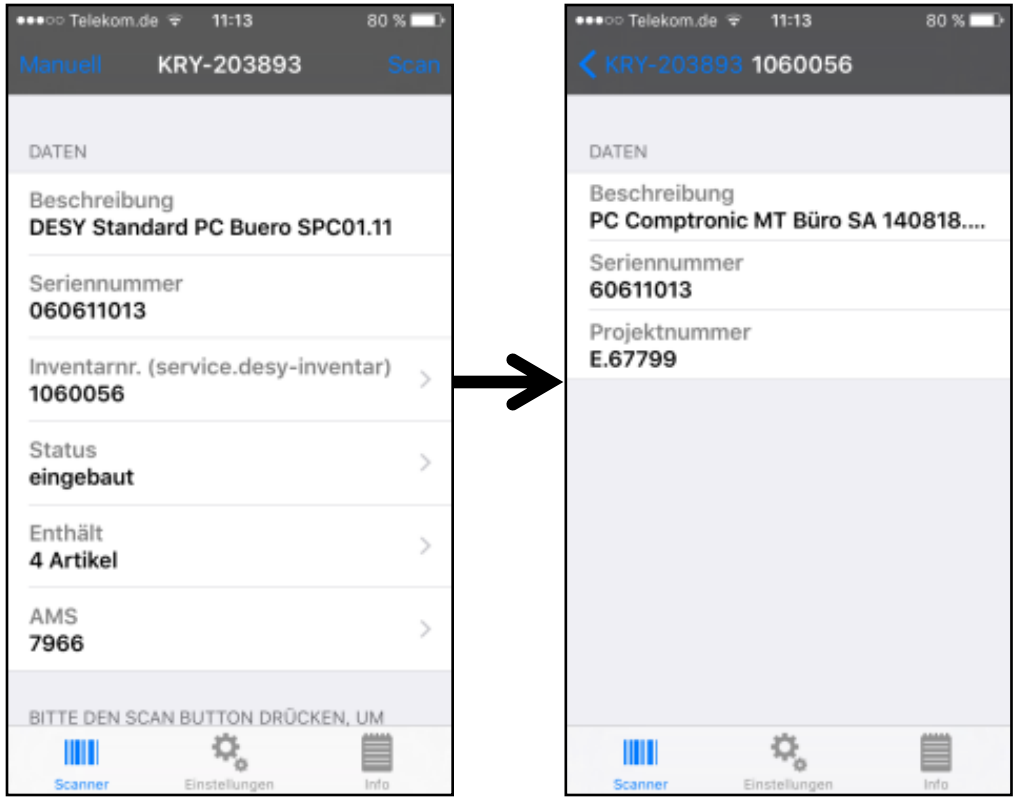
Software-Structure

Generic handling of xml data in the browser



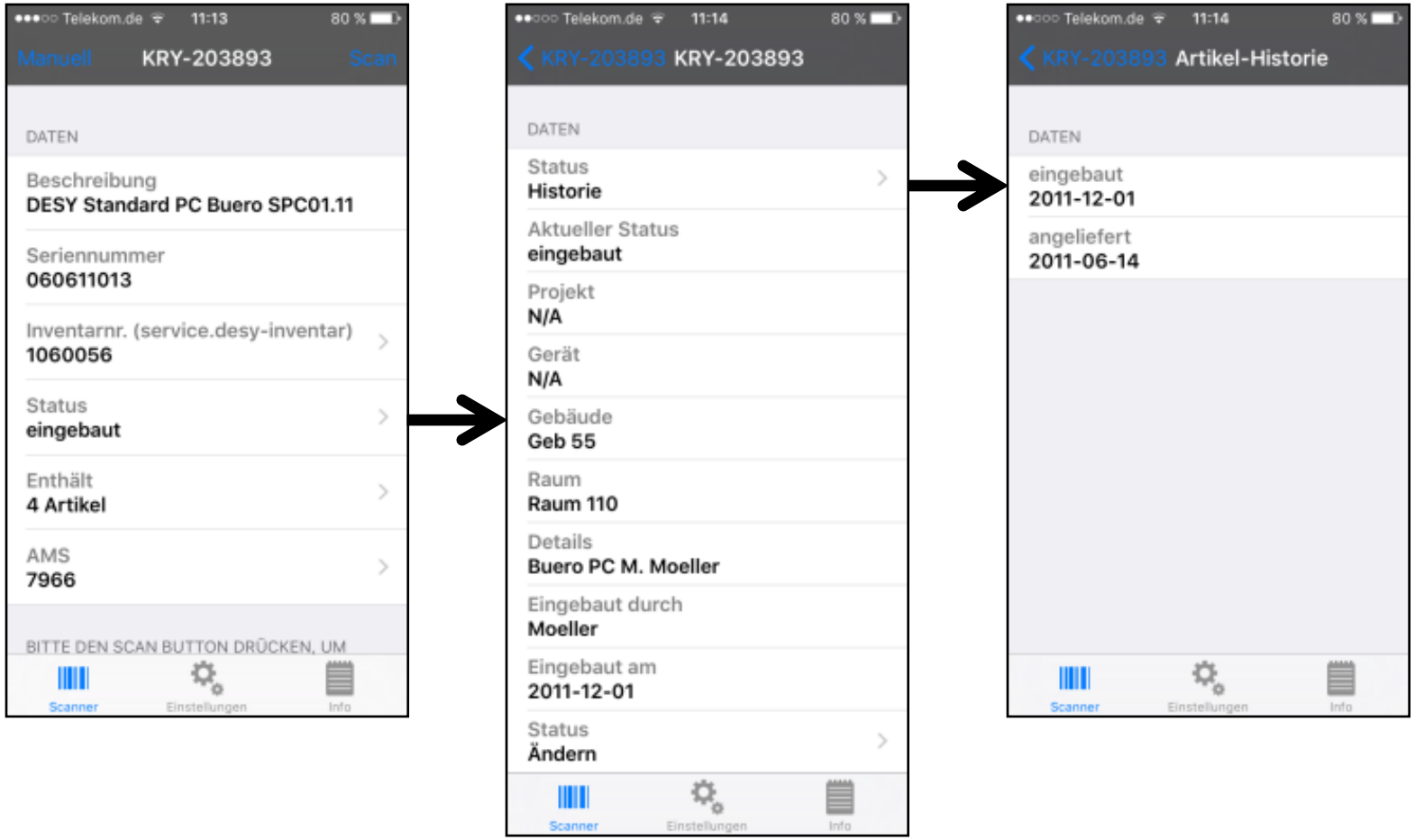
Asset management

From scanned label to inventory



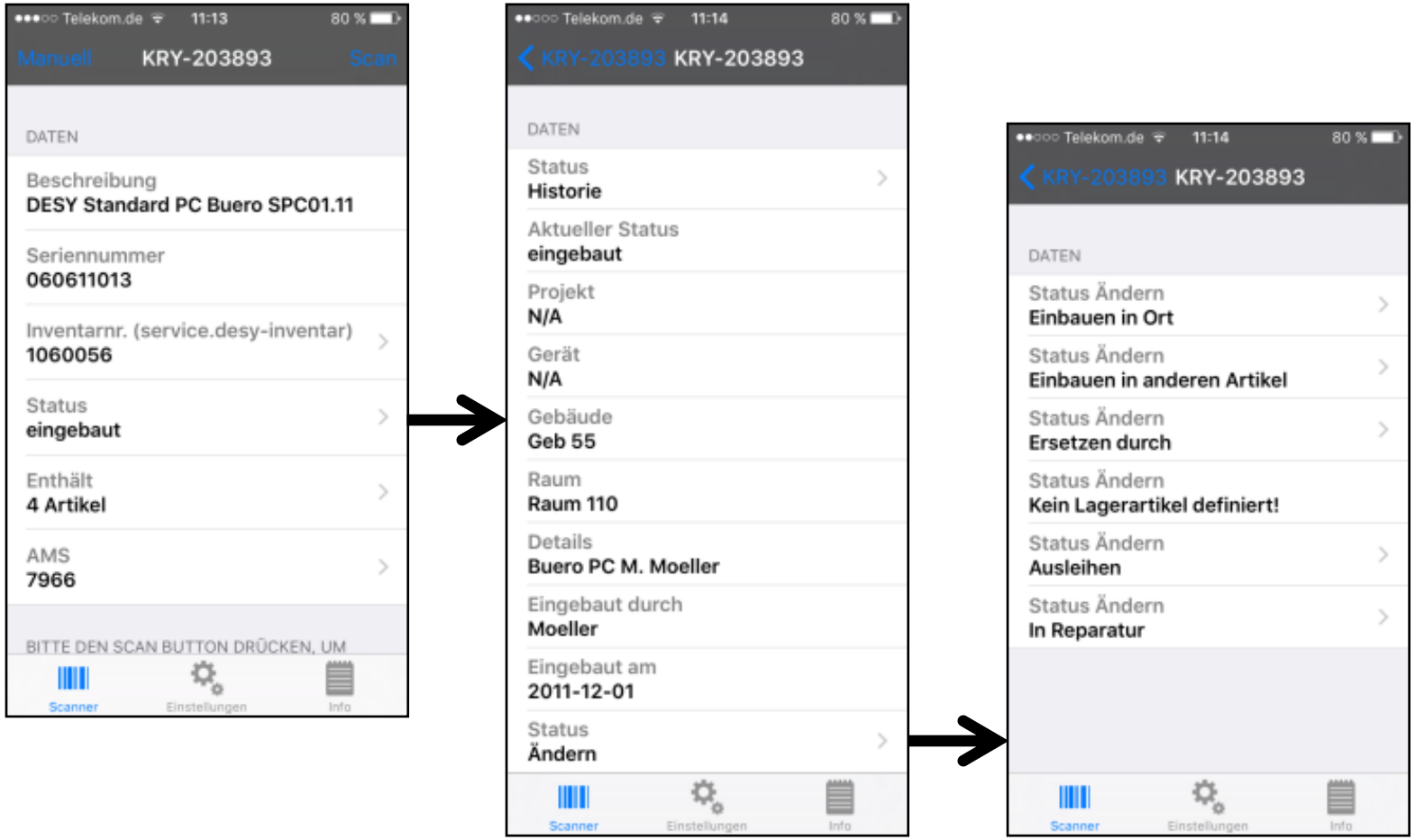
Asset management

Showing details and history



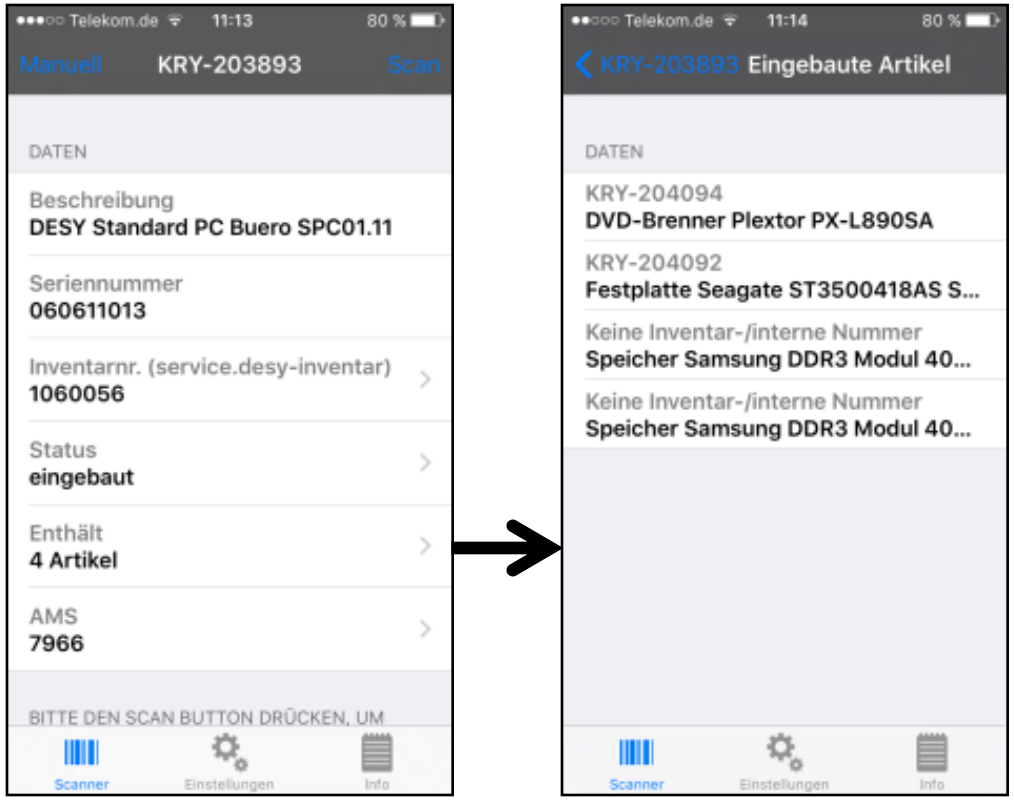
Asset management

Perform status updates



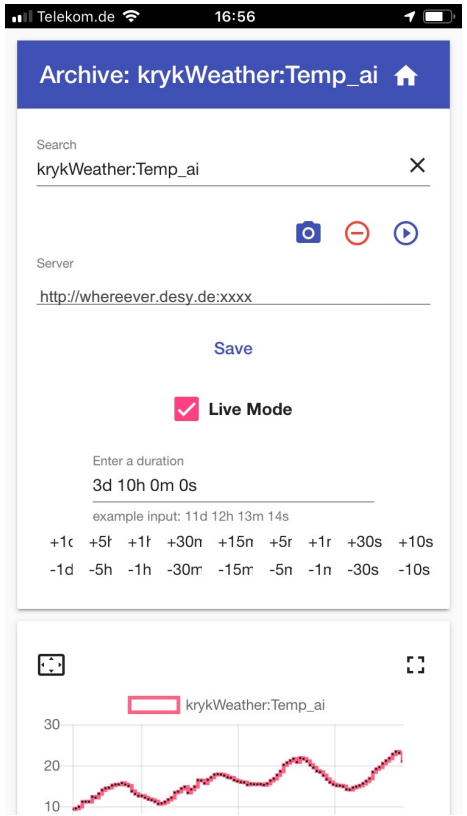
Asset management

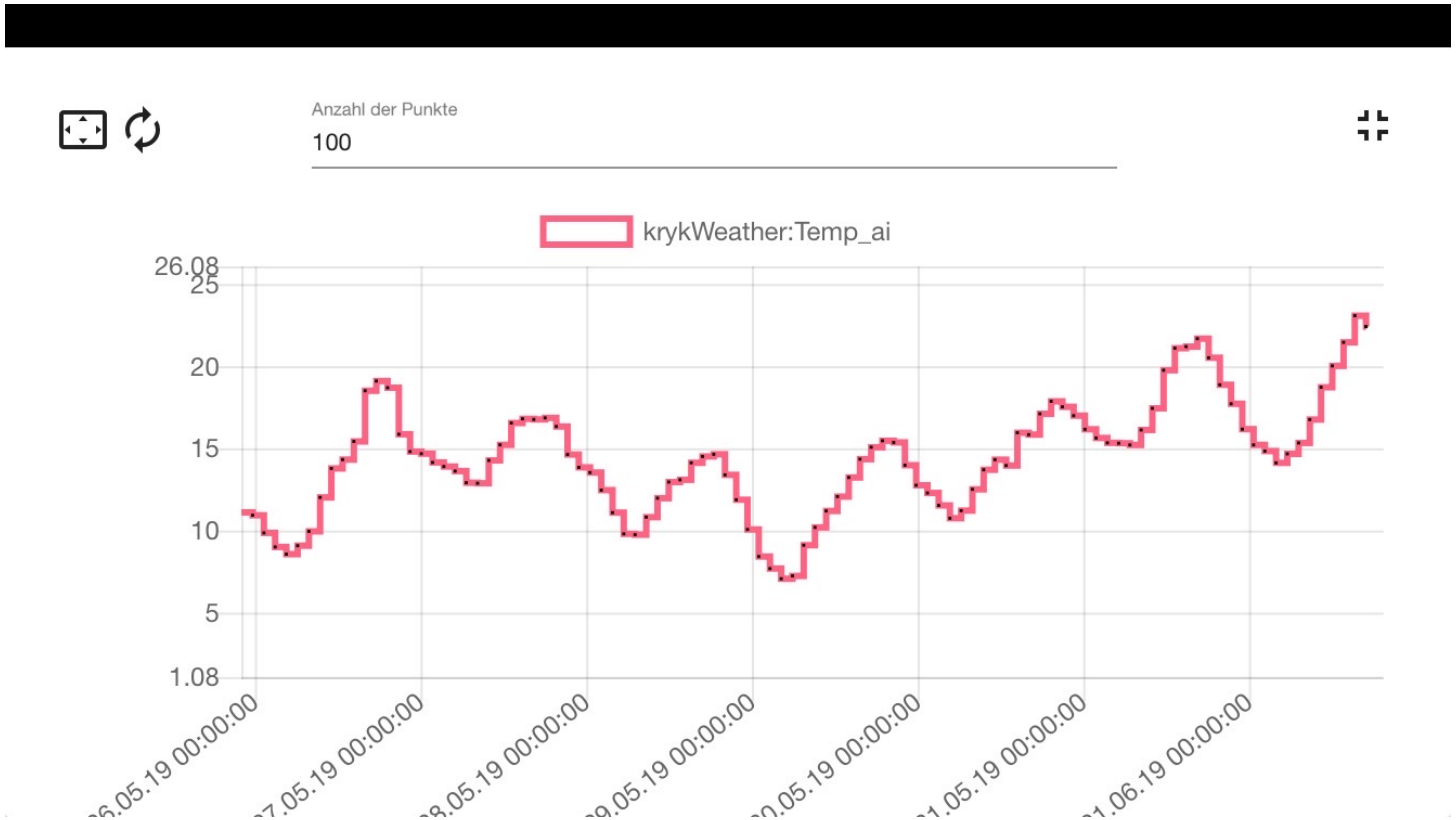
Make use of the ‘is-contained-in’ relationship



Data display

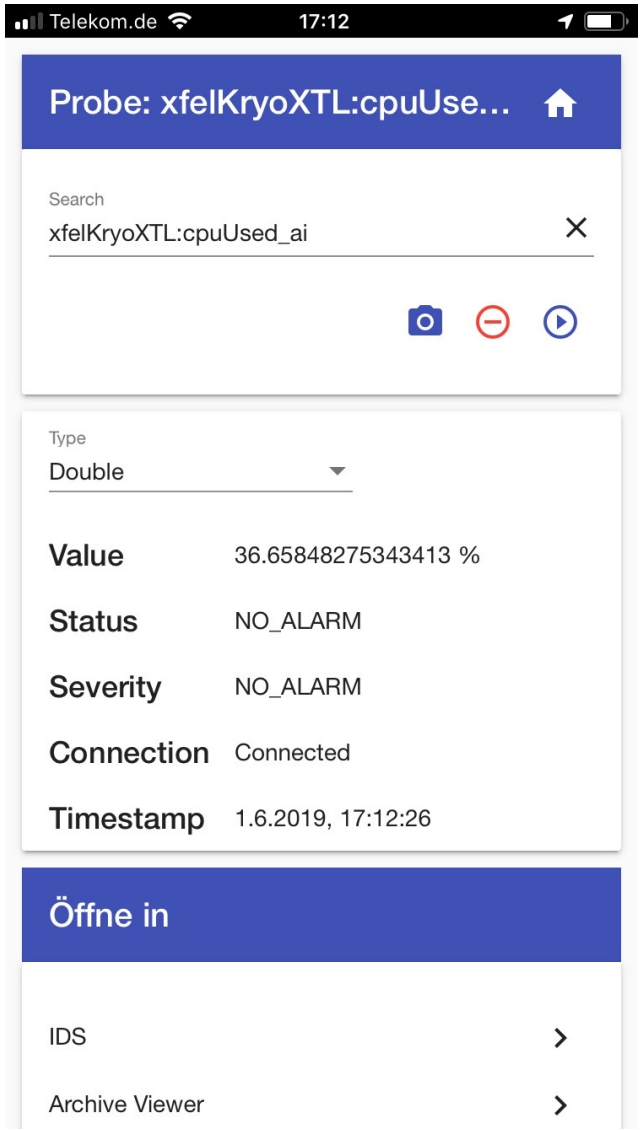
Archived data via Archive Request Server





Data display

Live data in Probe-style



Data display

Live data in OPI-style

Overview

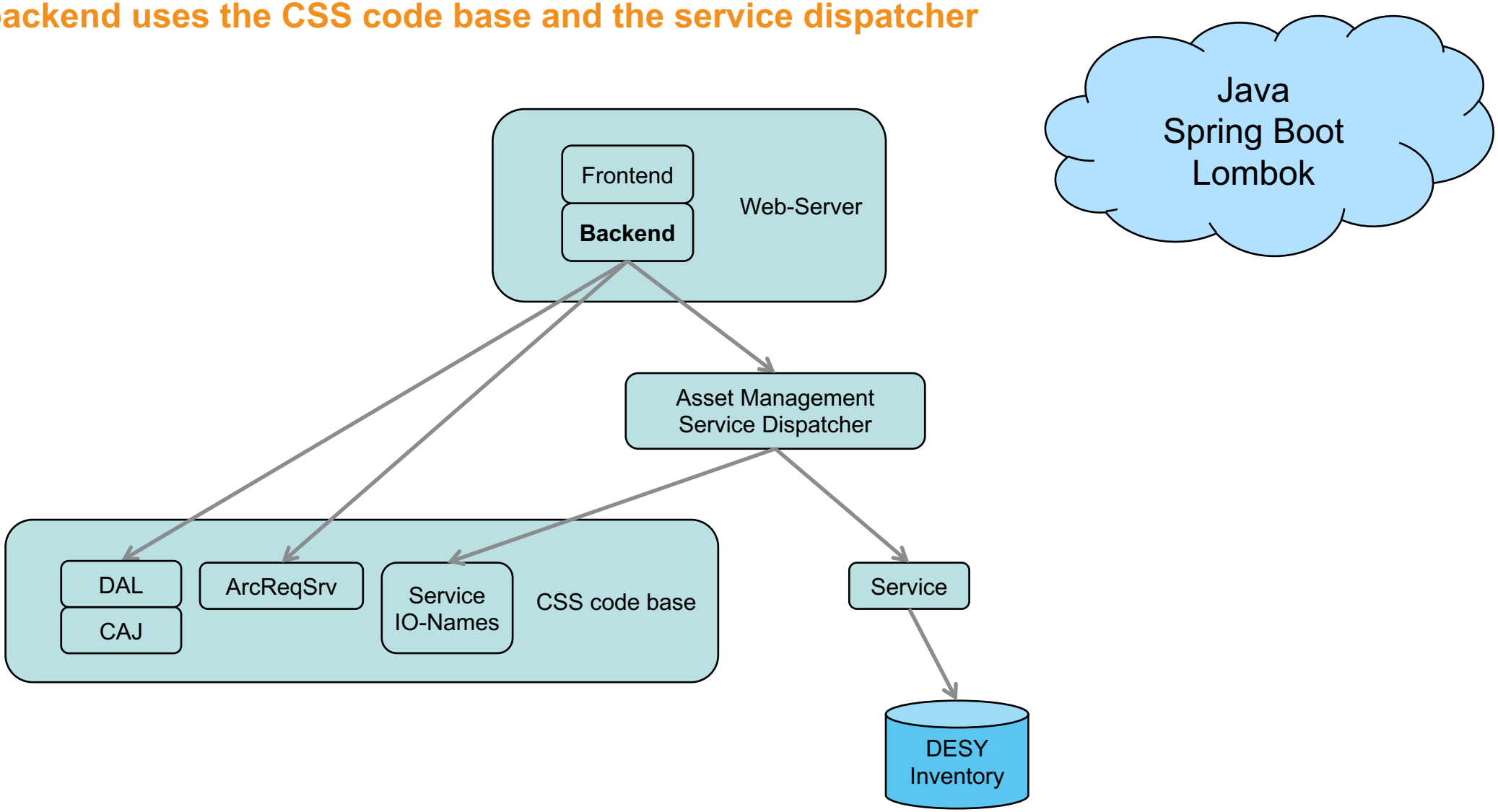
1 Motivation

2 Functional requirements
- Asset management
- Displaying archived data
- Displaying live data

3 Realization
- Implementation
- Tools
- Deployment

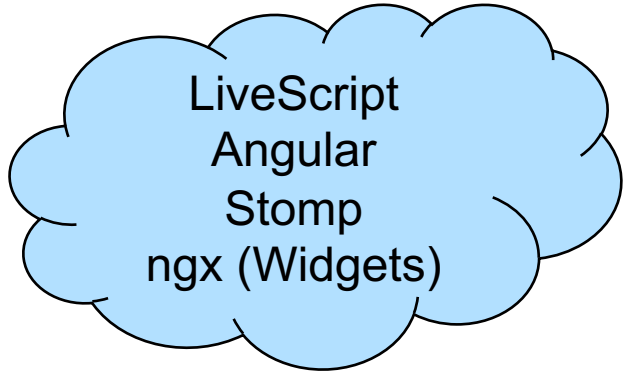
Implementation

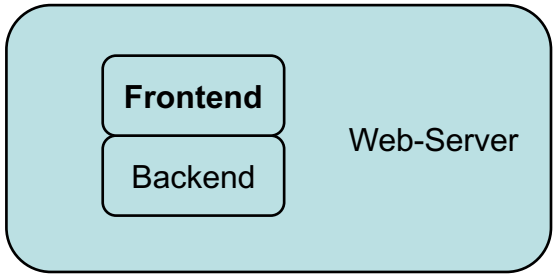
Web-Server backend uses the CSS code base and the service dispatcher

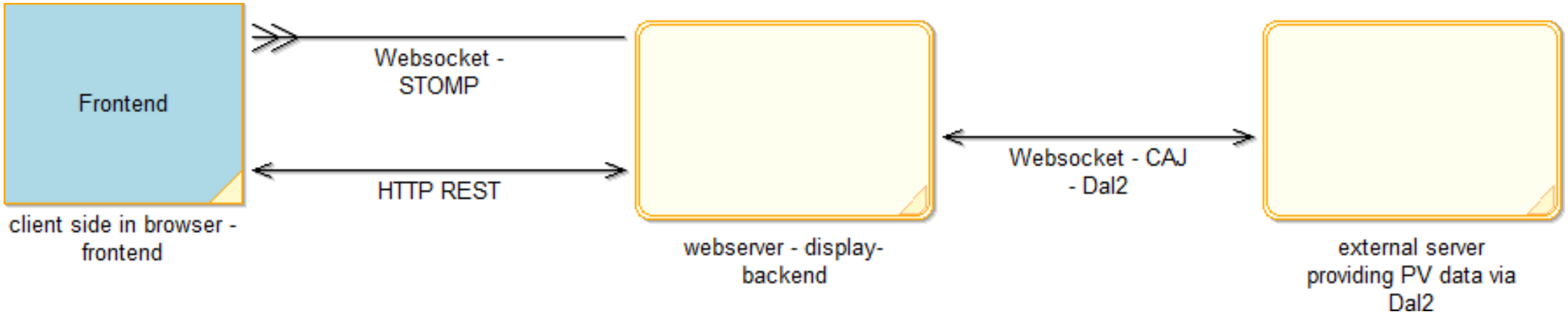


Implementation

Web-Server frontend

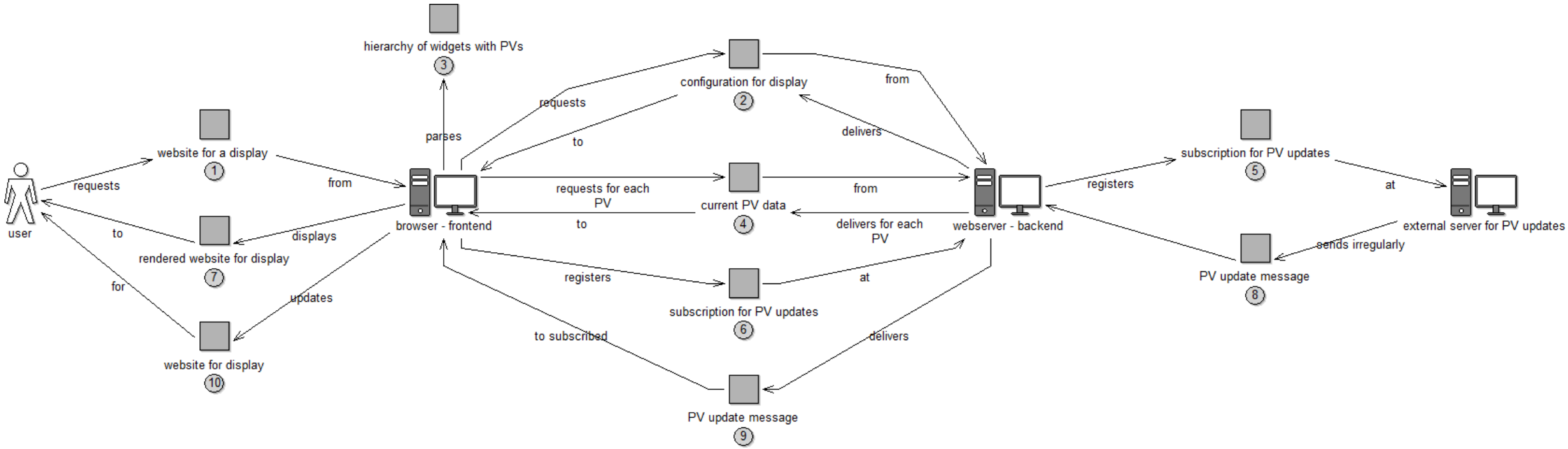






Implementation

Web-Server frontend communication



Tools

Eclipse

- Web-Server-Backend
- CSS code base

Visual Studio Code

- Web-Server-Frontend
- Live Script

Git-Repository

- stash.desy.de
- (not yet) ☹

Deployment

What do we need to make it run?

Web-Server

- Spring Boot application is deployed as a .war file to some DESY-Server
- IT opens the gate for us

Scanner-App

- Is required for scanning the data matrix labels
- Available for iOS and Android

Thank you

Contact

DESY. Deutsches
Elektronen-Synchrotron

www.desy.de

Jörg Penning
MKS-2 Cryogenic Controls
joerg.penning@desy.de