Smart Monitoring System

Zielsetzung des Teilprojektes 2

- Fokussierung auf
  - Effiziente Kommunikation und Speicherung aller Datentypen
  - Universell einsatzfähige Datenbank
  - Entwicklung von Algorithmen für große Datenmengen zur Datenanalyse
  - Persuasive Programmierung einer benutzerfreundlichen Applikation
  - Visualisierung der Energieeffizienz von Wohnungen

Smart Monitoring System

Zielsetzung des Teilprojektes 2

- Smart Monitoring Frontend – Webapplikation
  - Webanwendung zur persuasiven Visualisierung von Messdaten
  - Basiert Clientseitig auf HTML5, CSS und JavaScript
  - Clientseitige Kommunikation mittels JSON zu Server-Schnittstellen
  - Forschungsobjekt zur Datenerfassung und Auswertung von Betriebsdaten bei z.B. Solaranlagen, intelligenten Gebäuden und Messeinrichtungen

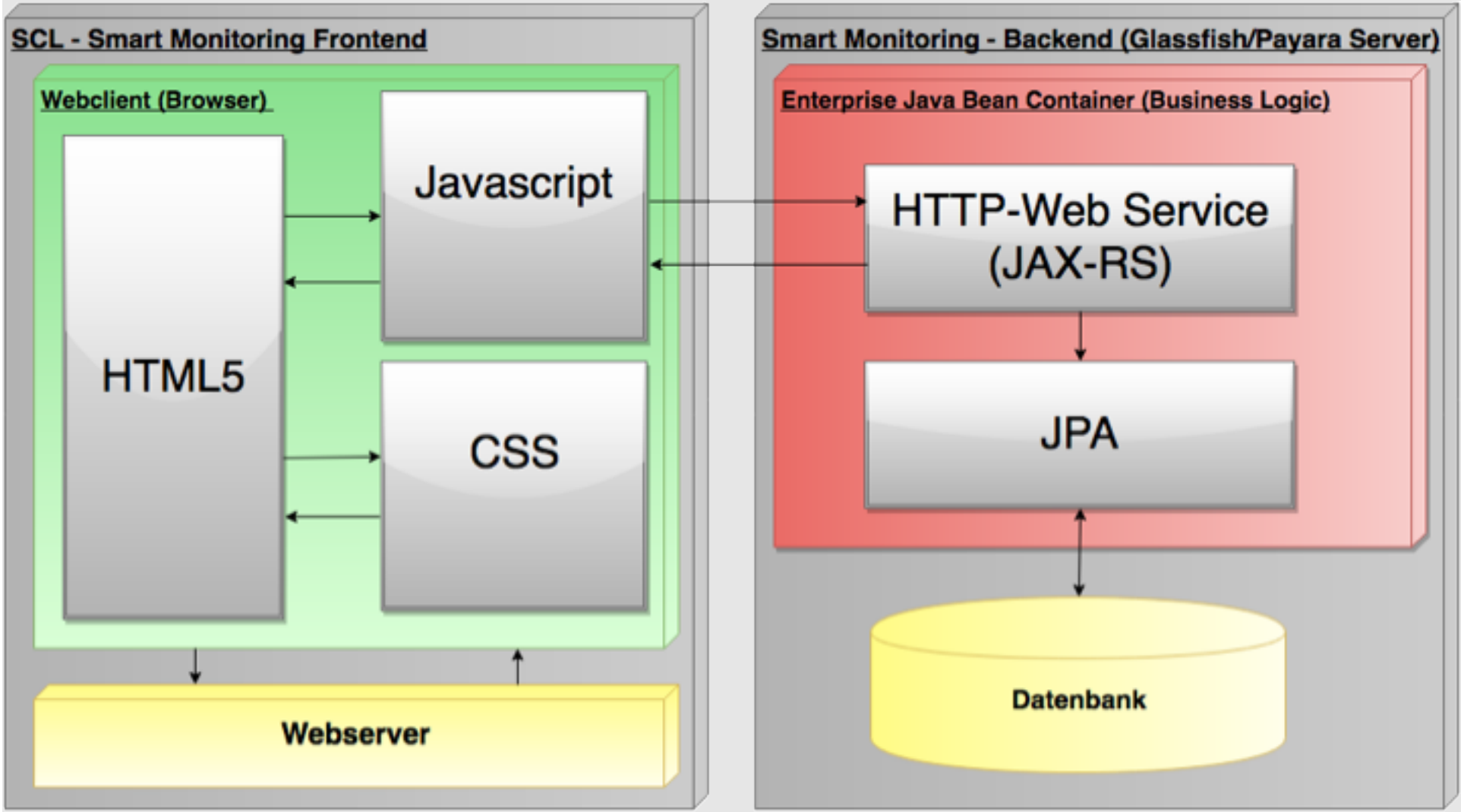
Smart Monitoring System

Zielsetzung des Teilprojektes 2

- Smart Monitoring Backend – Serveranwendung
  - Umsetzung von Open Source Technologien von Java EE

    z.B. EJB, JNDI, JPA, JAXB, JAX-RS
  - Bereitstellung von RESTFul Webservices Schnittstellen
  - JSON basierte Client-Server-Kommunikation
- Frontend und Backend – Client- und Serveranwendungen
  - Finden Verwendung zur Lehre für den praktischen Umgang mit modernen verteilten Softwaresystemen mit Schwerpunkt Webentwicklung

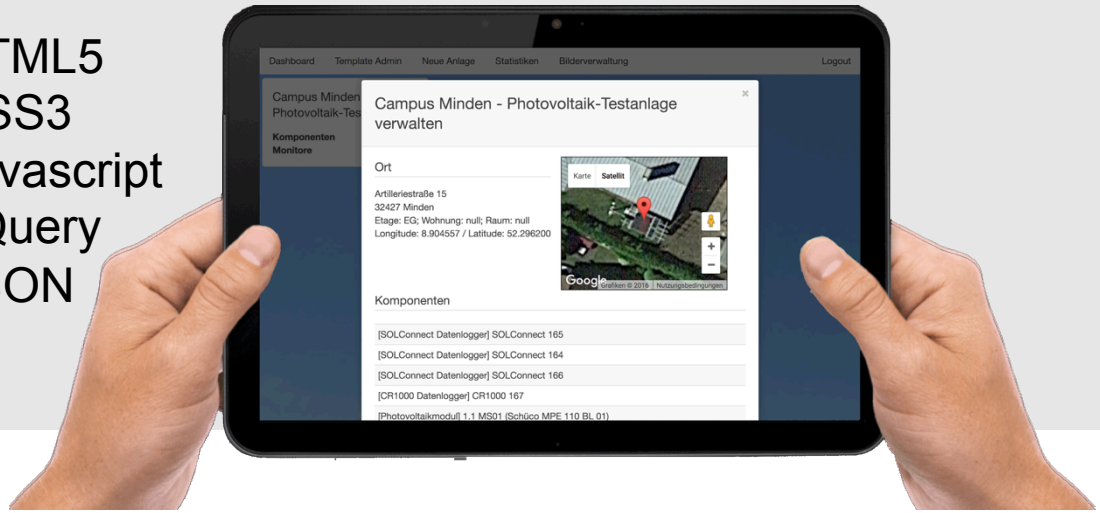
Smart Monitoring System



Frontend

Modernes responsive Design

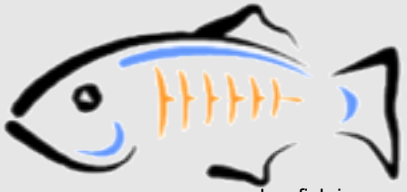
- Dynamische Webanwendung für modernen Einsatz in Browsern
- Benutzerunterstützende Webanwendung
- Skalierbarkeit bei verschiedenen Endgeräten und Browsern
- Automatisierte Anordnung von Webcontent zur besseren Visualisierung auf verschiedenen Endgeräten
- Persuasive Programming mit Fokus auf Energieeffizienz
- Verwendung moderner clientseitiger Webtechnologien:
  - HTML5
  - CSS3
  - Javascript
  - JQuery
  - JSON



Backend - Technologien

Open Source Webstandards

GlassFish



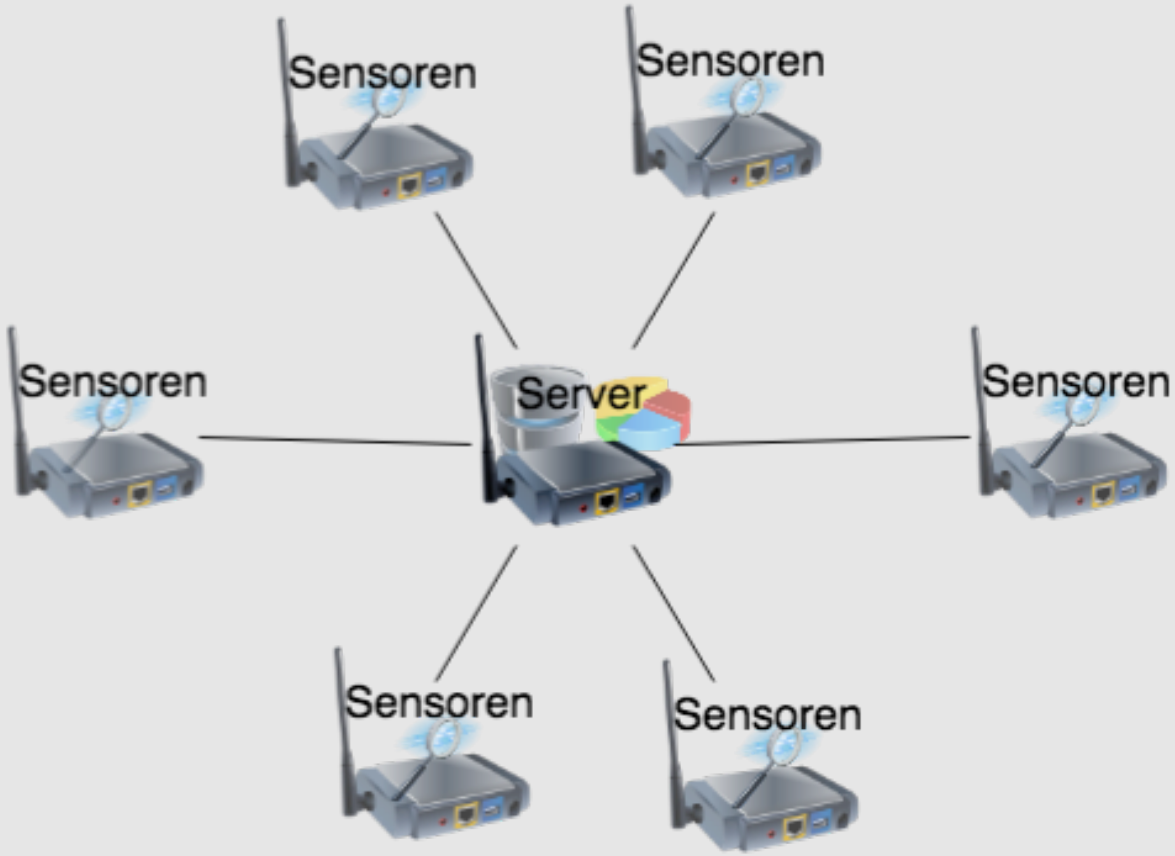
glassfish.java.net

- Verwendung von Java Platform Enterprise Edition (Java EE)
  - Enterprise JavaBeans (EJB)
  - Java Persistence API (JPA)
  - Java API for RESTful Web Services (JAX-RS)
  - Java API for JSON Processing (JSON-P)
  - Java Architecture for XML Binding (JAX-B)
- Payara bzw. Glassfish 4 Application Server
- Client-seitige Kommunikation mittels JSON (JavaScript Object Notation)

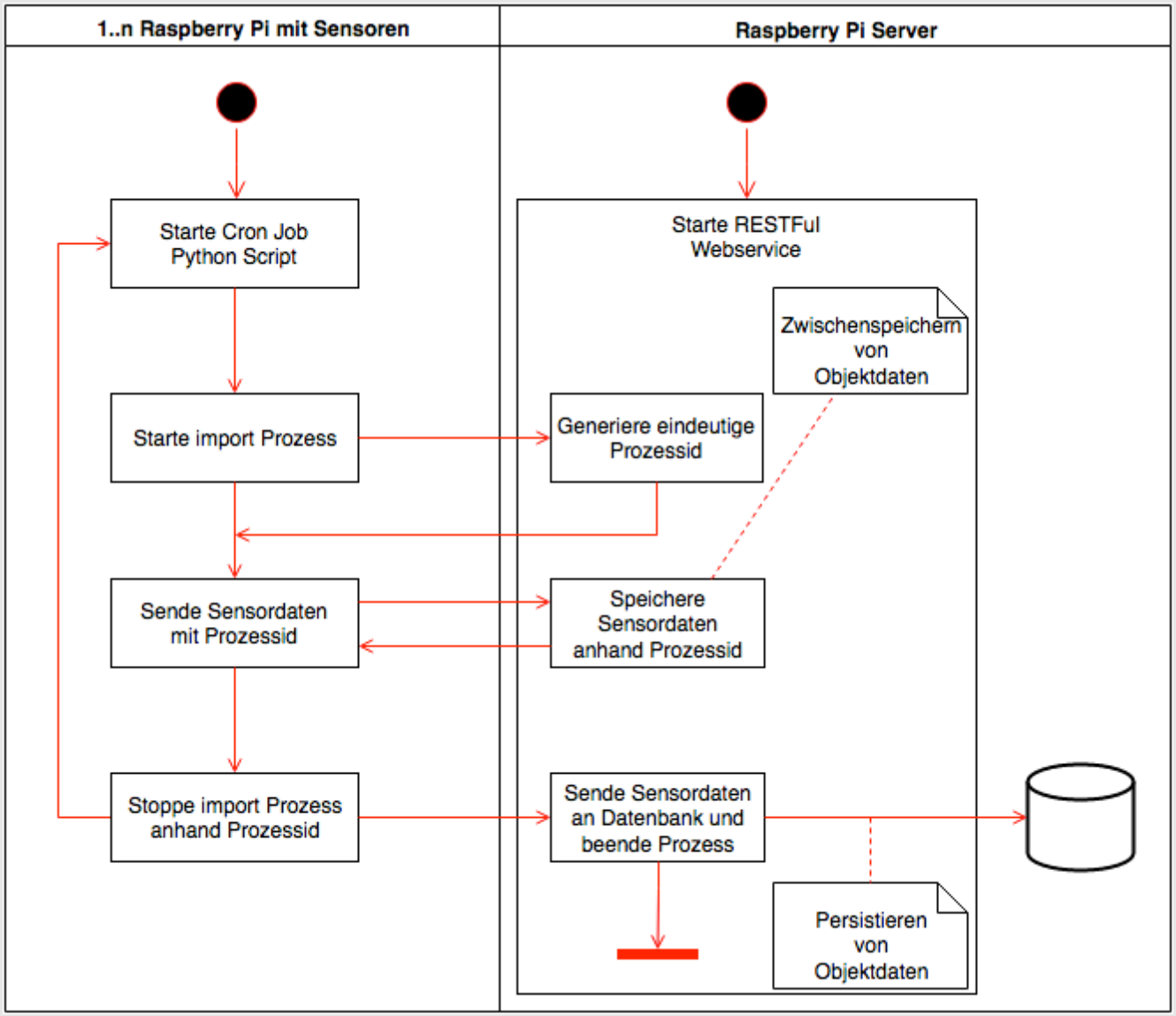
www.oracle.com

Raspberry Pi - Topologie

1..n Sensor-Raspberry Topologie



Sensor-Messdaten - Erfassung

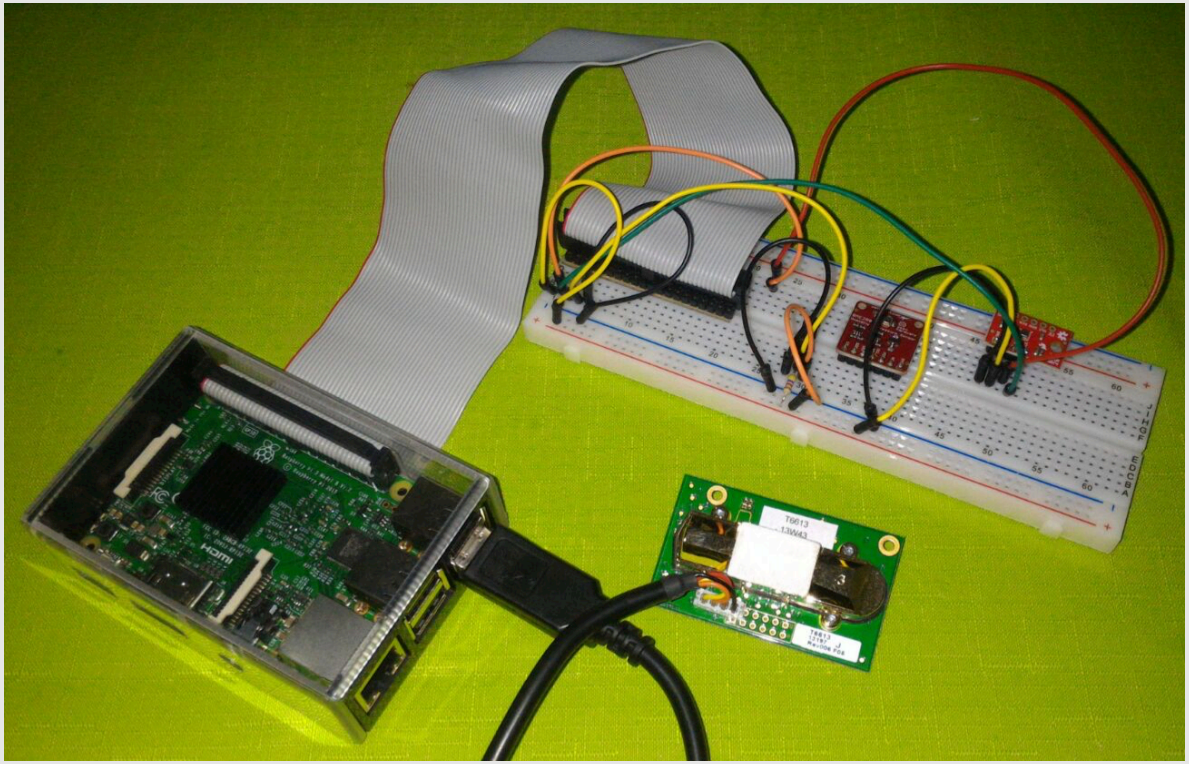


Das Messsystem in Entwicklung– IFE

Raspberry Pi Server (links)

Entwicklungsumgebung

- Glassfish Applikationsserver
- Java Platform Enterprise Edition (Java EE)
- PostgreSQL Datenbank

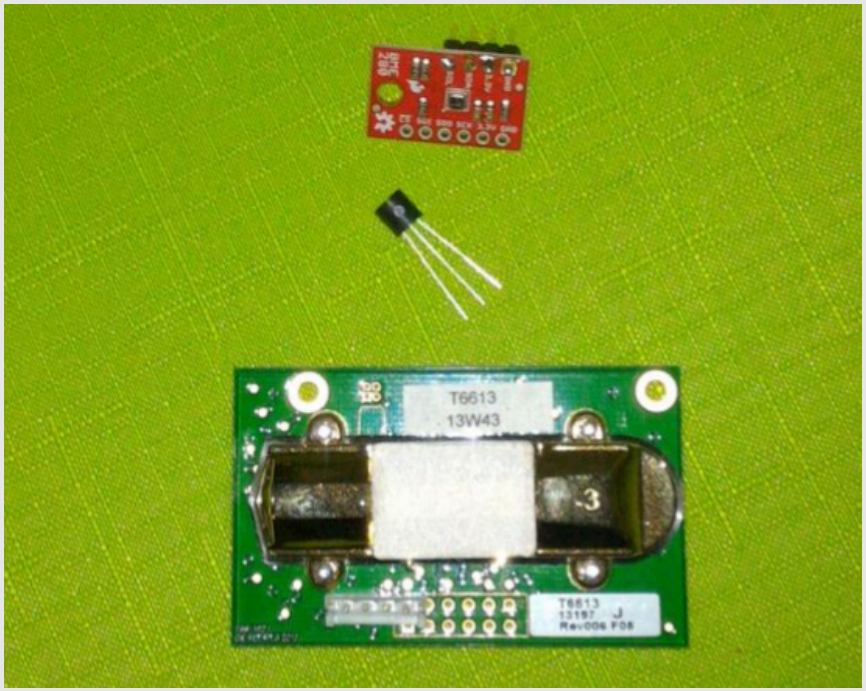


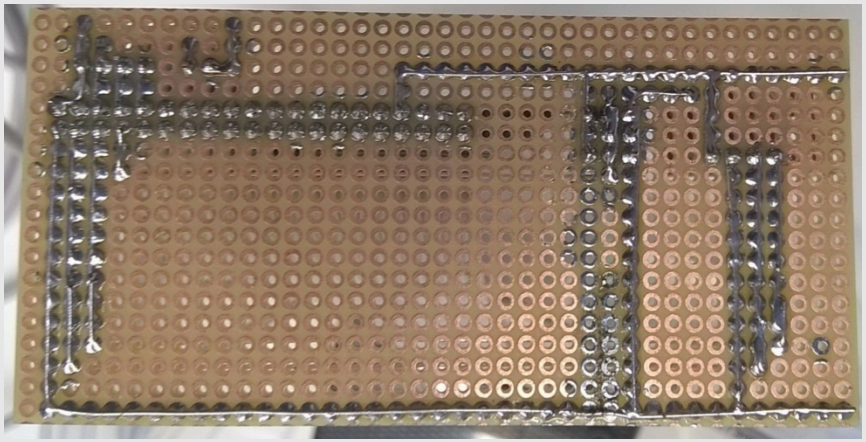
Raspberry Pi mit Sensorsteckbrett (rechts)

Entwicklungsumgebung

- Cron-Job
- Python Script
- Sensorplatine
- Sensormodule

Das Messsystem im Einsatz – IFE

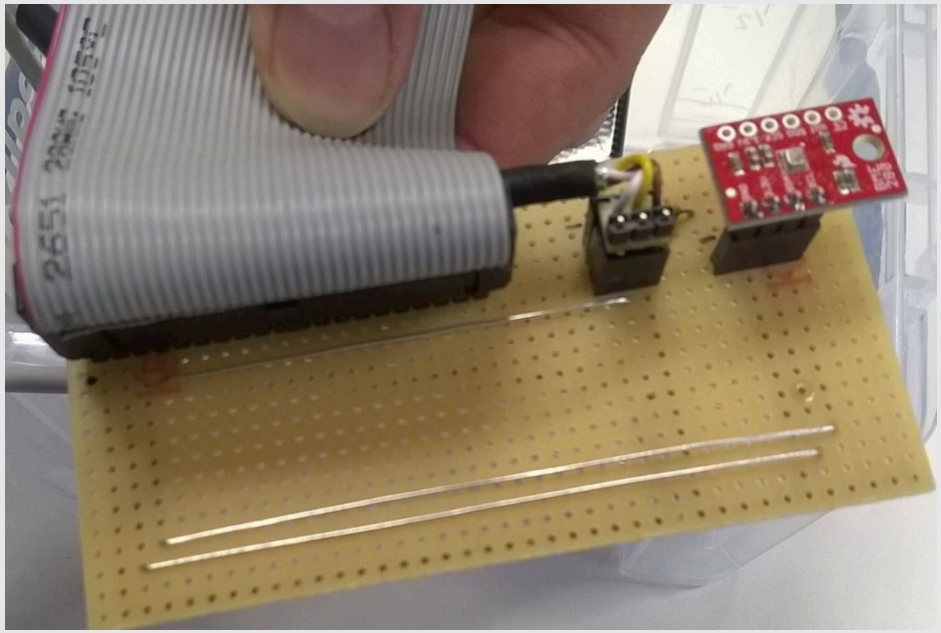




Raspberry Pi - Sensorplatine

Produktivumgebung im Einsatz

- BME 280
  (Luftfeuchtigkeit, -druck und -temperatur)
- DS18B20
  (Raumtemperatur)
- T6613
  (Kohlenstoffdioxid)



Solar Computing Lab

FH Bielefeld
University of
Applied Sciences

Das Messsystem im Einsatz – IFE

Das Messsystem im Einsatz – IFE

Raspberry Pi - Messsystem

Produktivumgebung im Einsatz

- Raspberry Pi mit Sensoren (links)
- Raspberry Pi Server (rechts)
- Gesichertes Netzwerk
- Sensoren sind in der Wohnung verteilt
- Zentraler Server als Datensammelpunkt
- Webanwendung zur Visualisierung

Frontend

Ausblick

- Einsatz von ausgewählten Technologien
  - Erweiterbar-, Skalierbar-, und Austauschbarkeit einzelner Komponenten
- Messsystem beliebig erweiterbar und anpassungsfähig
  - Messsystem-Konstellationen
- 3D-Thermokopter
  - Generierung und Visualisierung von z.B. Gebäuden oder Solarmodulen innerhalb 3D-Landschaften mithilfe von Thermo-, Normal- und Infrarotbildmaterial
  - Analyse und Wärmezuordnung zu Bildpunkten
  - GPS-Allokation und Standort-Mapping