

# SWE 1, 3 Semester

---

## Übungsbeispiel – *embedded sensor cloud*

### Aufgabenstellung

---

Auf einem <<tragen Sie bitte hier Ihr Embedded System Ihrer Wahl ein>> soll ein Messdaten Erfassungssystem implementiert werden. Seine Aufgabe ist es, mittels Plugins Messdaten zu erfassen und zu speichern. Die Messdaten können dann mittels einer REST Schnittstelle von der Sensor Cloud ausgelesen werden. Die Konfiguration des Systems erfolgt über ein Web interface.

Aus dieser kurzen Angabe ergeben sich folgende konkrete Aufgabenstellungen

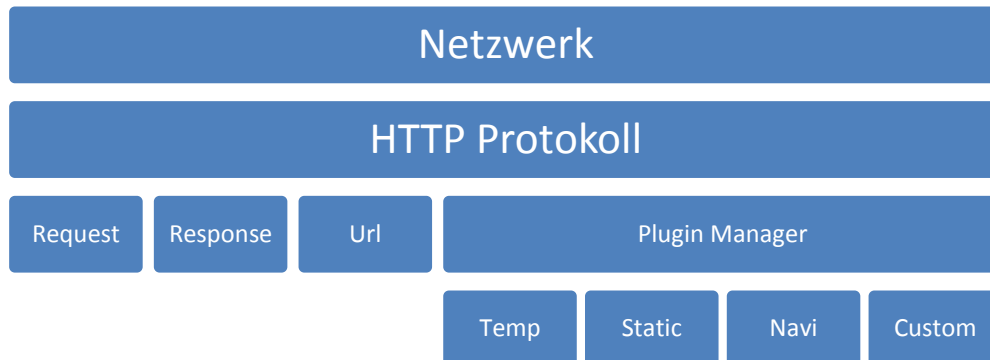
- Implementierung eines WebServer
  - Annahme: Auf einem Embedded System sind die Ressourcen begrenzt, daher kann kein Windows bzw. LAMP Server ausgeführt werden
  - Annahme: Jedoch ist das System in der Lage, ein Linuxsystem mit Java bzw. dem Mono-Framework auszuführen
  - Für die Übung: Der WebServer ist selbst zu implementieren, einzig die Methoden URLEncode/URLDecode dürfen aus der BCL verwendet werden.
- Der WebServer kann über den Port 8080 erreicht werden
- Der WebServer muss multiuserfähig sein
- Der WebServer muss über ein Plugin System verfügen. Diese Plugins führen den konkreten Request aus.
  - Plugins müssen dynamisch nachgeladen werden können (über eine Konfiguration)
- Die Messdaten müssen in einer Datenbank Ihrer Wahl gespeichert werden.
  - Um die Übung zu vereinfachen müssen die Plugins NICHT die Datenbank verwalten können. Sie dürfen sich auf ein korrektes Schema verlassen.

- Temperatur Messung
  - Ein eigener Thread liest laufend einen Sensor aus
  - Für die Übung: Es werden Zufallszahlen oder ein Sinus, etc. berechnet
  - Diese Messwerte werden in der Datenbank ihrer Wahl gespeichert
  - Auf der WebOberfläche können die Messdaten dargestellt werden
  - Für die Übung: Erzeugen Sie mindestens 10.000 Messwerte verteilt über die letzten 10 Jahre.
  - Bei 10.000 Messwerten muss die Oberfläche zwangsläufig eine Suche bzw. Blättern unterstützen
  - Eine REST Abfrage <http://<server>:8080/GetTemperature/2012/09/20> soll alle Temperaturdaten des angegebenen Tages als XML zurück geben. Das XML Format ist frei wählbar.
- Statische Dateien
  - Dieses Plugin soll einfach nur Dateien aus dem Filesystem an den Browser zurück senden.
- Navi
  - Als extra Feature, weil das Embedded System Speicherkarten aufnehmen kann
  - In einem Textfeld wird ein Straßenname eingegeben
  - Das Plugin gibt anschließend eine Liste aller Orte aus, in denen die Straße existiert.
    - Arbeiter Strandbad Straße
    - -> In Wien.
  - Eine Karte mit allen Straßen sowie Orten kann von OpenStreetmap bezogen werden.
    - Die Datei müsste ein ca. 4 GB großes XML Dokument sein
    - <http://download.geofabrik.de/osm/europe/>
    - Es genügt, wenn einer/eine die Datei organisiert und allen anderen weitergibt.
  - Eine eigene Seite im Navi Plugin kann den Befehl: „Straßenkarte neu aufbereiten“ auslösen
    - Dieser Befehl liest die OpenStreetmap Karte (neu) ein und erstellt eine interne Straßen <-> Ort Zuordnung im Hauptspeicher (z.B. in einem Dictionary oder HashTable). Wer möchte, kann das auch in der Datenbank speichern, ist aber nicht Pflicht.
    - Dies ist mittels SAX Parsers zu realisieren
    - Es genügen die POI Tags! Keine geografischen Zuordnungen durchführen!
    - `//node|way/tag[@k,@v]`
    - Während dieser Aufbereitung darf das Plugin keine anderen Abfragen annehmen dürfen. Es muss stattdessen eine Warnmeldung ausgeben (Multiuserfähigkeit)
- Ein weiteres Plugin, welches Sie selbst bestimmen und entwerfen können.

## Nichtfunktionale Anforderungen

- Implementieren Sie mind. 20 Unittests
- Entwerfen Sie die Architektur und dokumentieren Sie diese
  - Listen Sie dabei jede Klasse auf und beschreiben Sie deren Aufgabe
  - Ca. 1 A4 Seite Text

## Architekturübersicht



Bewertung	Punkte	Max. Punkte
<b>Sockets</b>		
Verbindung ist möglich		1
Multiuser fähig		1
<b>http</b>		
Request & URL wird Aufbereitet		1
Response ist gekapselt		1
<b>PluginManager</b>		
Plugins können dynamisch geladen werden		2
Requests werden an die Plugins weitergereicht		2
<b>Plugins</b>		
Temperaturmessung		2
Statische Dateien		2
Navi		2
Karte neu laden Befehl funktioniert korrekt		1
Eigenes Plugin		1
<b>Nichtfunktionale Anforderungen</b>		
20 Unittests		3
Dokumentation der Architektur		1
<b>Summen</b>	<b>0</b>	<b>20</b>
<b>Gruppennote</b>	<b>?</b>	