

The logo for ZiM (Zentrum für Informationsmanagement) features the letters 'ZiM' in a stylized, blue, sans-serif font. The 'i' is lowercase and has a dot. The background is a dark blue gradient with several light blue speech bubbles of various sizes and orientations.The logo for 'Talk' features the word 'Talk' in a bold, red, italicized sans-serif font. Below it, the tagline 'WISSEN SCHAFFT IT' is written in a smaller, red, sans-serif font. The logo is contained within a white speech bubble with a drop shadow, set against a dark blue background with other speech bubbles.

# *Home Automation und Datenvisualisierung*



Daniel Biella ■ 21.02.2014

UNIVERSITÄT  
DUISBURG  
ESSEN

*Offen im Denken*

- **Home Automation**
  - ...and the internet of things
  - Grundbegriffe
- **openHAB (Open Home Automation Bus)**
  - Kurzer Überblick und „etwas Hardware“
  - „Fünf Kostbarkeiten“ - 5 „hands-on“ Beispiel-Szenarien für den Umgang mit *Items*, *Sitemaps* und *Rules*
- **Exkurs: Datenvisualisierung**
  - Dynamische SVG-Grafiken und exempl. Integration ins UI
- **Ausblick**

# Home Automation

**„[...] today's information technology is so dependent on data originated by people that our computers know more about ideas than things. If we had computers that knew everything there was to know about things – using data they gathered without any help from us – we would be able to track and count everything, and greatly reduce waste, loss and cost.“**

*Kevin Ashton, 2009*



ELEKTRONIKMESSE CES

## Wenn die Zahnbürste mit dem Smartphone spricht

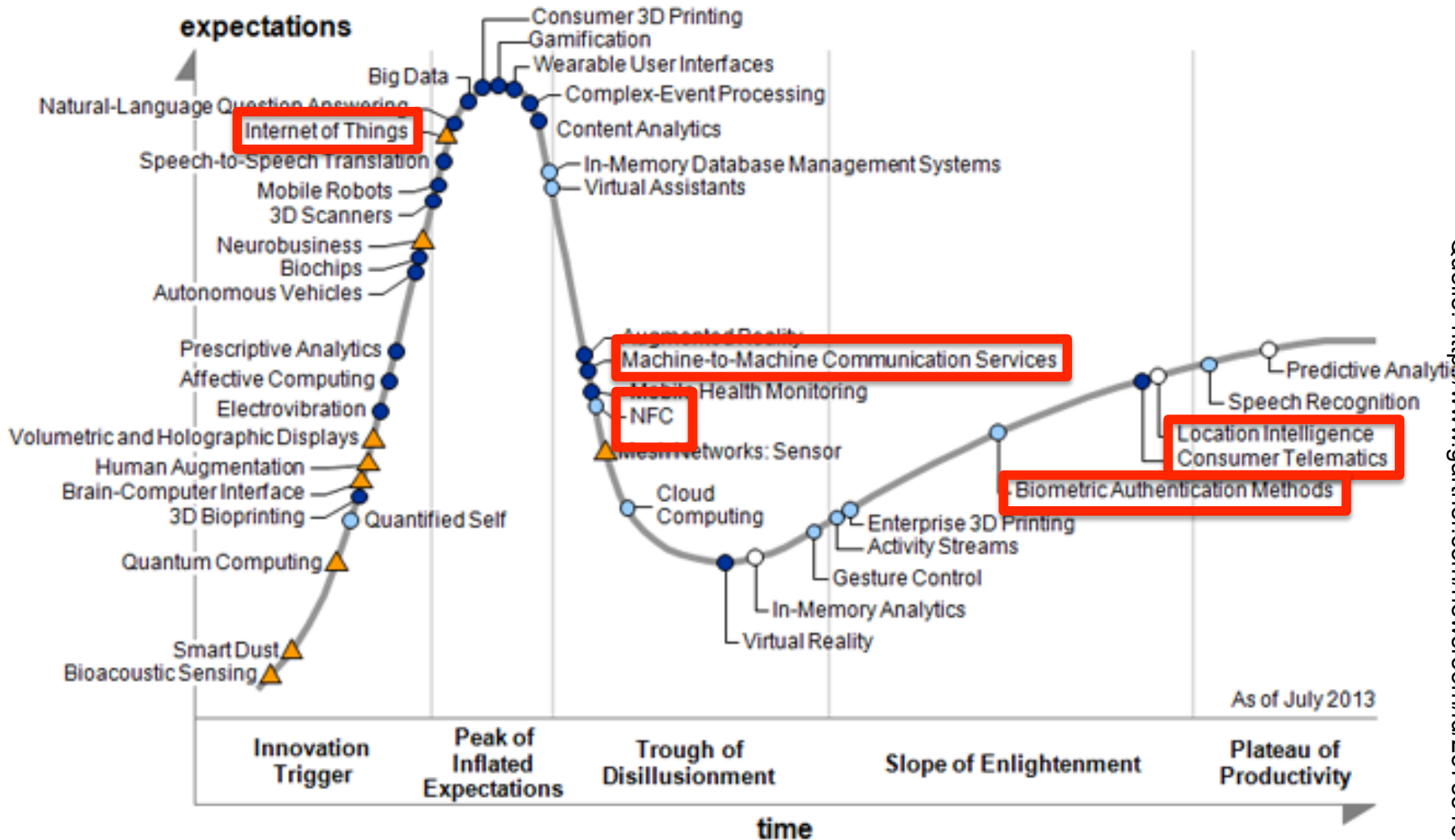
Heizungen, Türschlösser und Wärmepumpen  
In Las Vegas ist zu sehen  
das Leben erleichtern

VERNETZTER HAUSHALT

## Google kauft Hersteller interaktiver Thermostate

Die Firma Nest entwickelt intelligente Haushaltsgeräte.  
Google will die Firma jetzt für 3,2 Milliarden Dollar kaufen  
– und könnte dadurch an die Daten der Nutzer kommen.

# Gardner hype cycle und Themenbezug



Plateau will be reached in:

○ less than 2 years

● 2 to 5 years

● 5 to 10 years

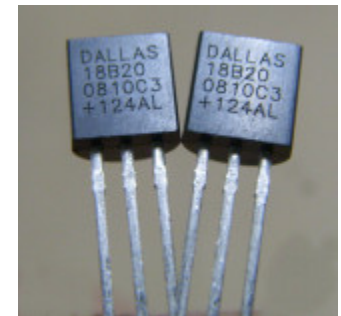
▲ more than 10 years

○ obsolete

⊗ before plateau

- **Sensorik**
- **Aktoren**
- **Digitale Datenübermittlung**
- **Transport-Protokolle**
- **Daten-Standards / Maßeinheiten**
- **Anzeige (Visualisierung) von Messwerten**
- **Automatisierte Regelung (Verknüpfung von Sensorik und Aktoren)**

- **Sensoren**
  - Drahtgebunden
  - Im Haus oder in unmittelbarer Umgebung
- **Bsp.: Temperatur-Sensor DS18B20+**
  - Digitaler 1-Wire Temperatursensor mit 9..12 Bit Auflösung
  - Betriebsspannung: 3...5.5 V
  - Messbereich: -55...+125 °C
  - Genauigkeit:  $\pm 0,5$  °C bei -10...+85°C





- **Entfernte Sensoren**

- Drahtlos abrufbar (IEEE802.11, 802.15, NFC, u.a.)
- Datenformate nicht zwingend genormt

- **Beispiele**

- Öffentliche Wetterdaten
  - Generator: Wetterstation
  - Datenformat: XML, Identifikation über ID
  - Echtheit und Integrität der Daten nicht zwingend gegeben
  - Anbieter: u.a. Yahoo, WeatherUnderground.com
- Koubachi

- **Umsetzung von Signalen in mechanische Bewegung oder andere physikalische Größen**
- **Aktoren & Sensoren: Teile des Regelkreises i.d. Steuer- und Regelungstechnik**
- **Allgemeine Beispiele:**
  - 433MHz-Empfänger mit Schalt-Relais (Schaltaktor)
  - Schrittmotor des S/L-Kopfes einer Festplatte (IT)
- **Auch: Effektoren (Robotik)**



- **1-wire**
- **CAN-Bus**
- **LON**
- **KNX**
- **Bluetooth**
- **NearField**
- **u.v.m.....**



- **Zumeist herstellergebunden (dadurch viele einzelne Apps)**
- **Wenige herstellerübergreifende Ansätze**
- **Erweiterbarkeit oft durch Sortimenten bzw. Geschäftsfelder eines Herstellers vorgegeben**



- **Uneinheitliche Visualisierungen**
- **Oft geringer Interoperabilitätsgrad zwischen verschiedenen Herstellern bzw. Protokollen**

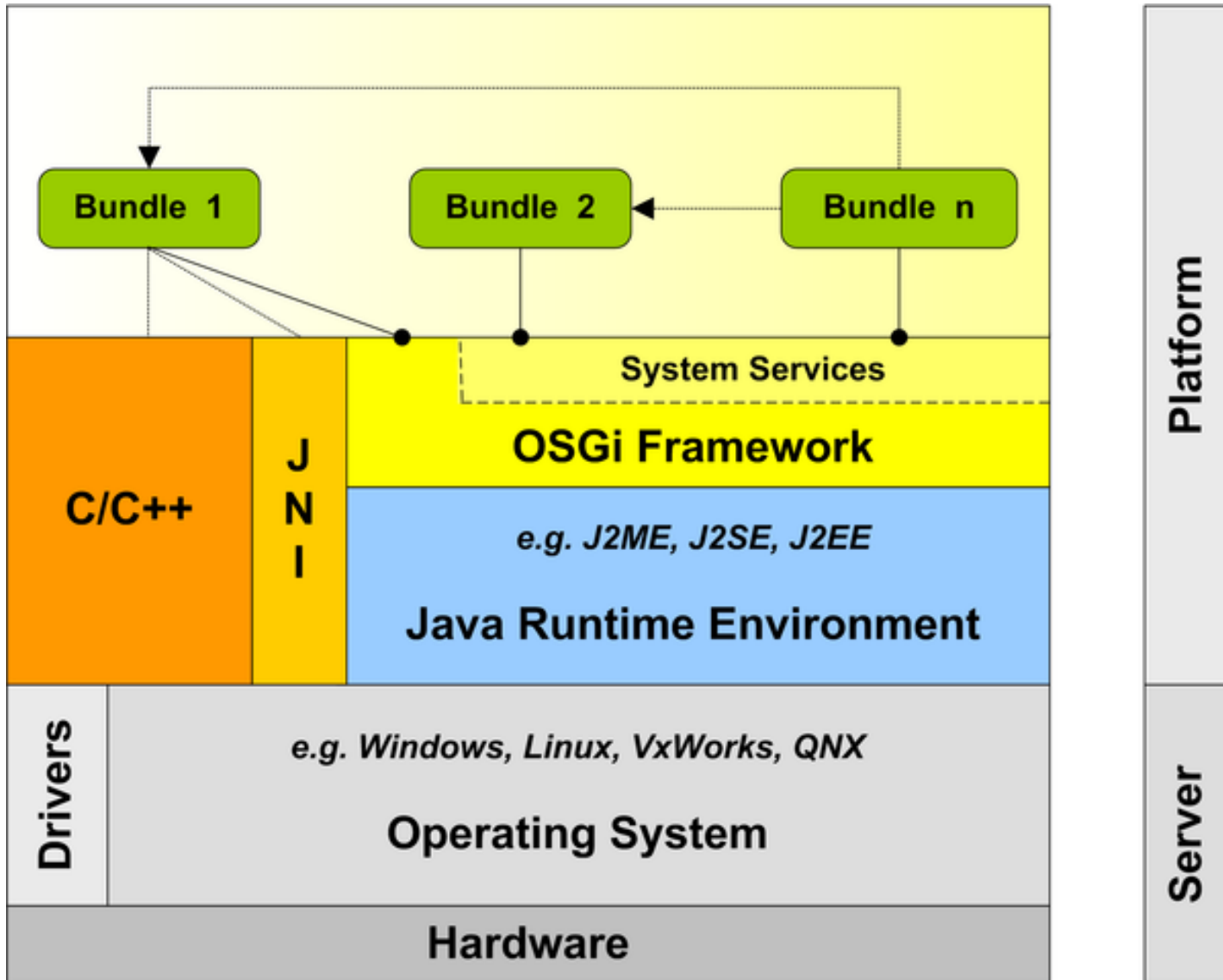


# Ein Lösungsansatz: Zentrale Anwendungslogik (z.B. OpenHAB)



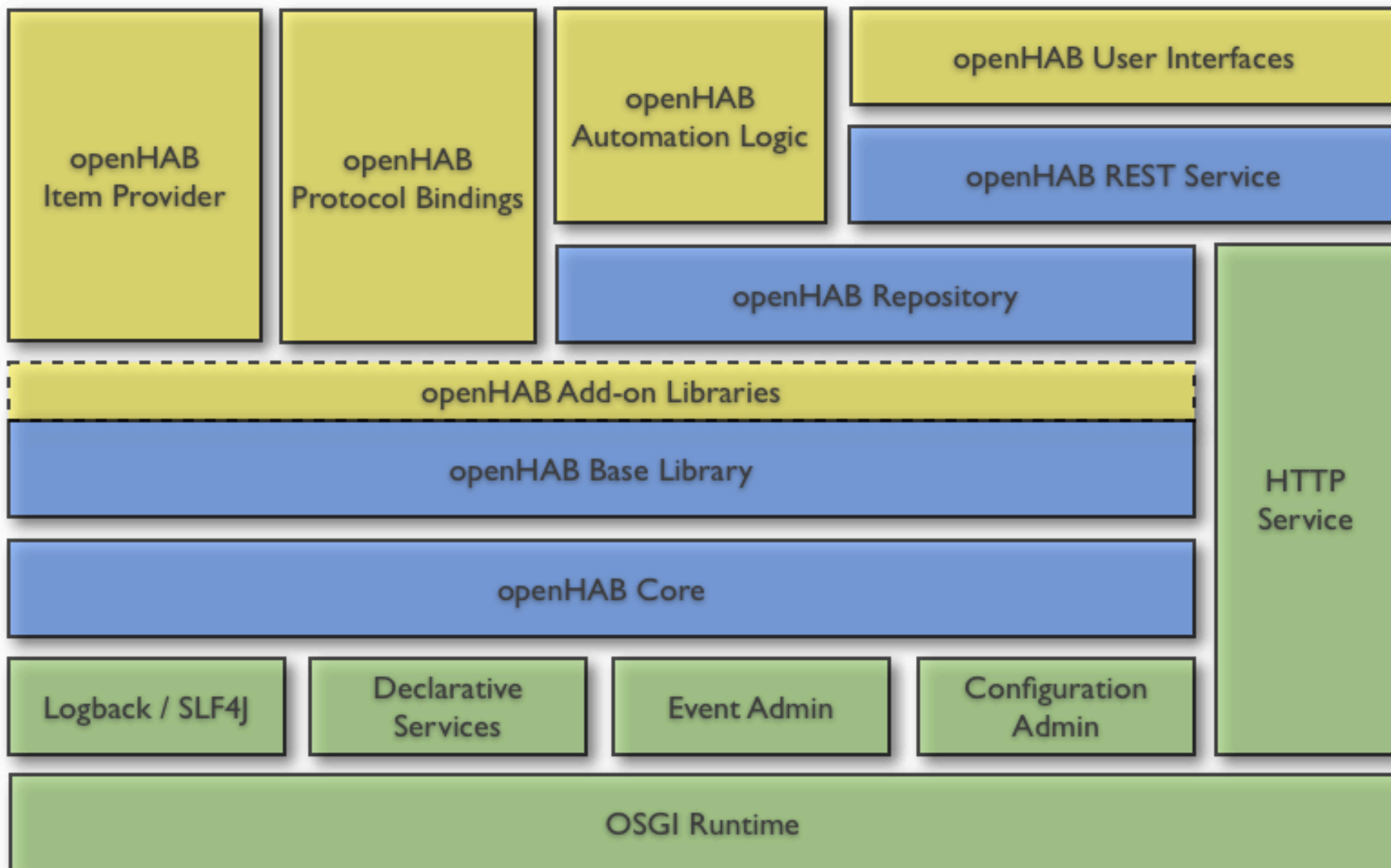
# OpenHAB - Überblick

- **Software OpenHAB besteht aus:**
  - RunTime (Server)
  - Designer (Konfigurationstool)
- **Entwickler: Kai Kreuzer, Thomas Eichstädt-Engelen u.v.a.**
- **Auf Basis von OSGi Komponenten**
  - **Open Services Gateway initiative** (<http://www.osgi.org>)
  - Equinox OSGi runtime
  - Jetty Web-Server
  - Impliziert: JVM (aka Java)
- **Herstellerübergreifendes Konzept**



# openHAB Architecture Overview

- openHAB Add-ons
- openHAB Core Components
- OSGi Framework





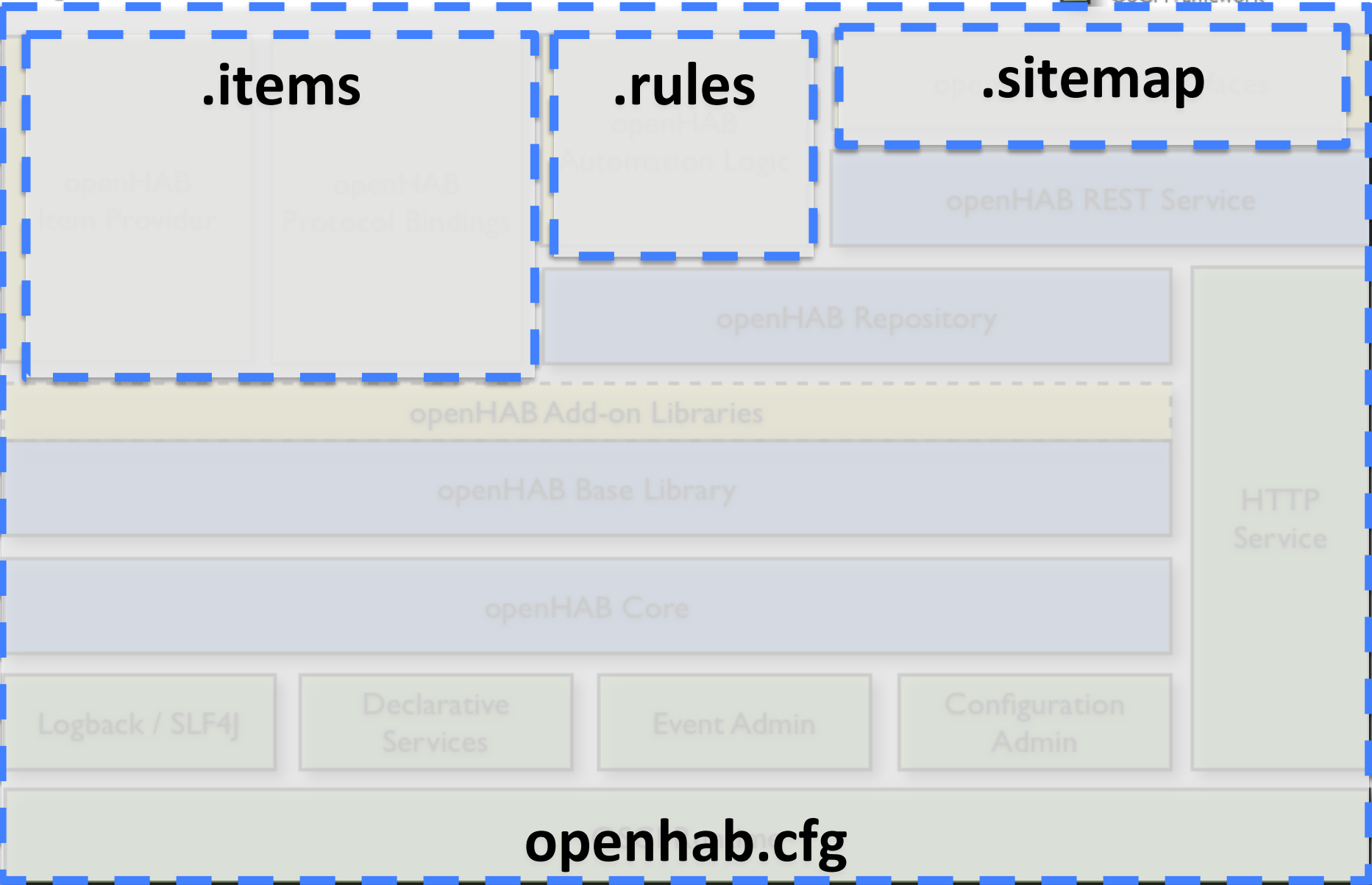
# openHAB Architecture Overview

- openHAB Add-ons
- openHAB Core Components
- OSGi Framework

**.items**

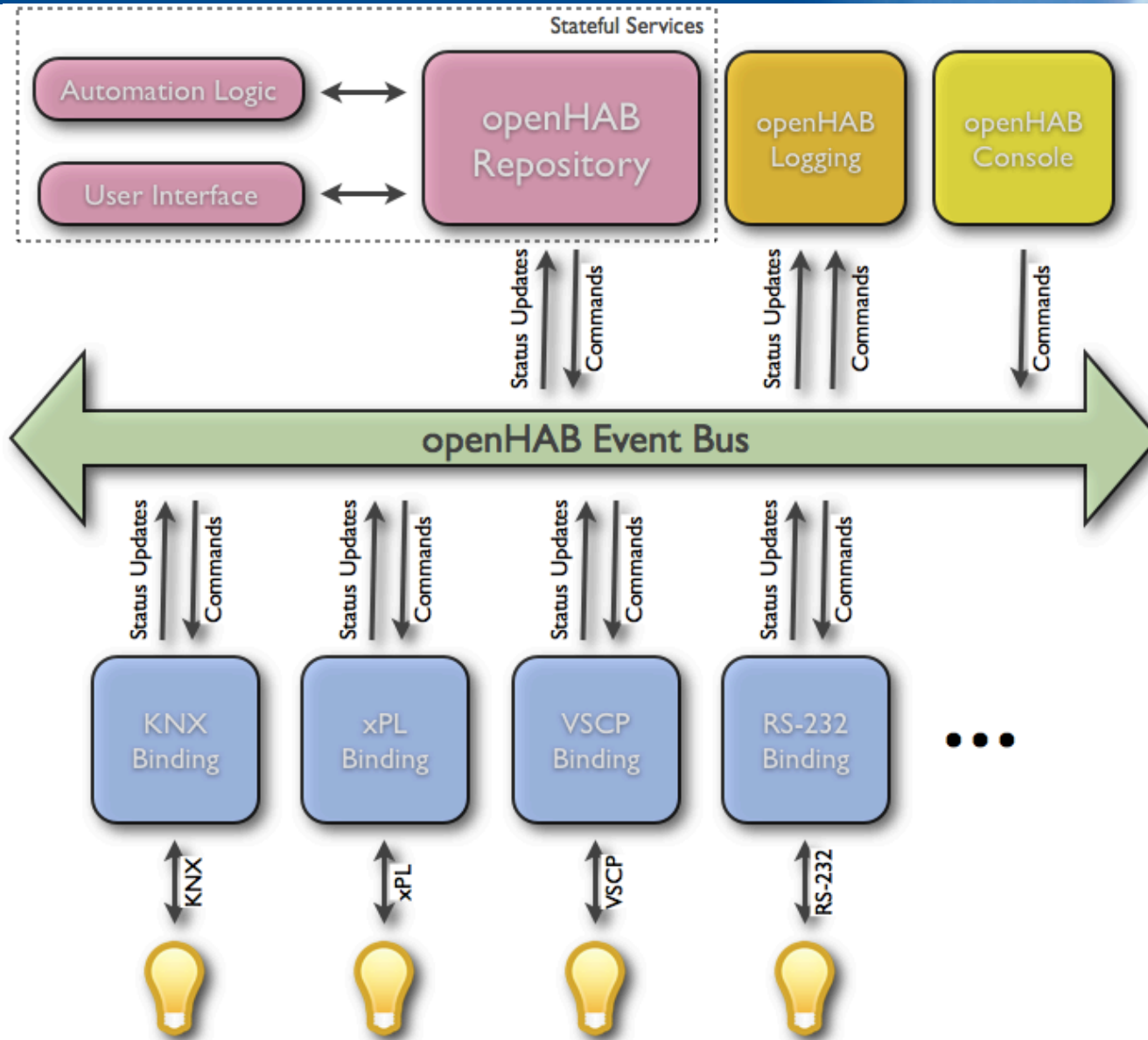
**.rules**

**.sitemap**



**openhab.cfg**

# OpenHAB - Kommunikationsmodell



- **Zentraler Zustandsspeicher**
  - Element (item)
  - Gerät (device)
- **Ereignis-Leitung (event bus)**
  - Asynchrone Schnittstelle
- **Ereignisarten:**
  - Befehl (Trigger für Ausführung od. Zustandswechsel)
  - Zustandswechsel (oft als Antwort eines Befehls)

***„Hands-on“***

# Download der Software – <http://openhab.org/>



Home

Features

Downloads

Community

Github

Wiki

my.openHAB

## Downloads

HOME / DOWNLOADS

Getting Started

Downloads

### openHAB Runtime



Runtime core

This package includes core functionality of openHAB installation

Download



Addons

This package includes all openHAB bindings and bundles

Download



Demo setup

This package includes everything needed to setup openHAB demo

Download

### openHAB Designer



Windows

Download



OS X

Download



Linux

Download

### UI applications



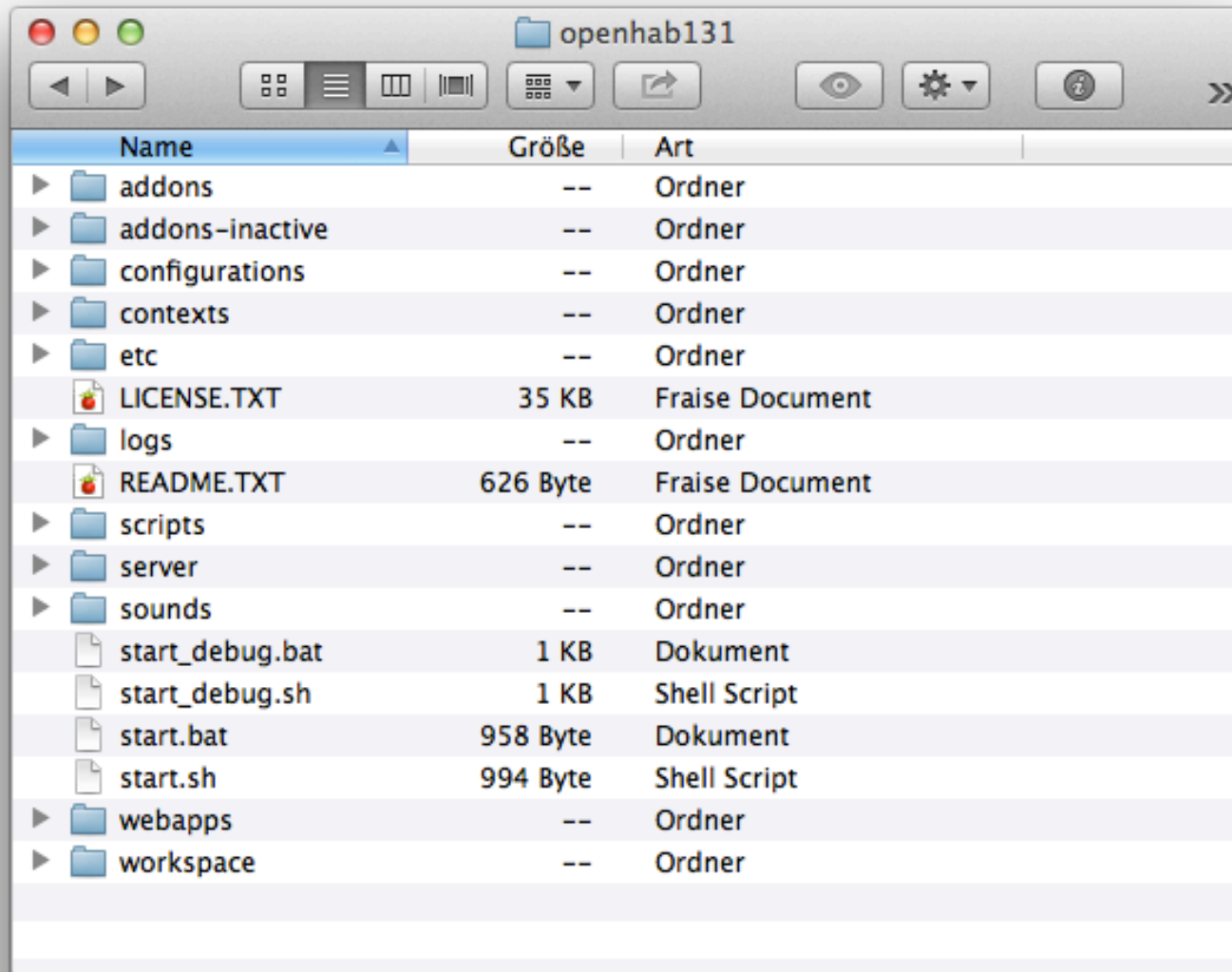
Android



iOS







# Vorstellungsrunde „*Hardware*“ für heutige Demo

# Beispiel: Schaltbare Steckdosen





# Beispiel: Schaltbare Steckdosen



# Beispiel: Funk-Schalter





# Beispiel: Funk-Schalter

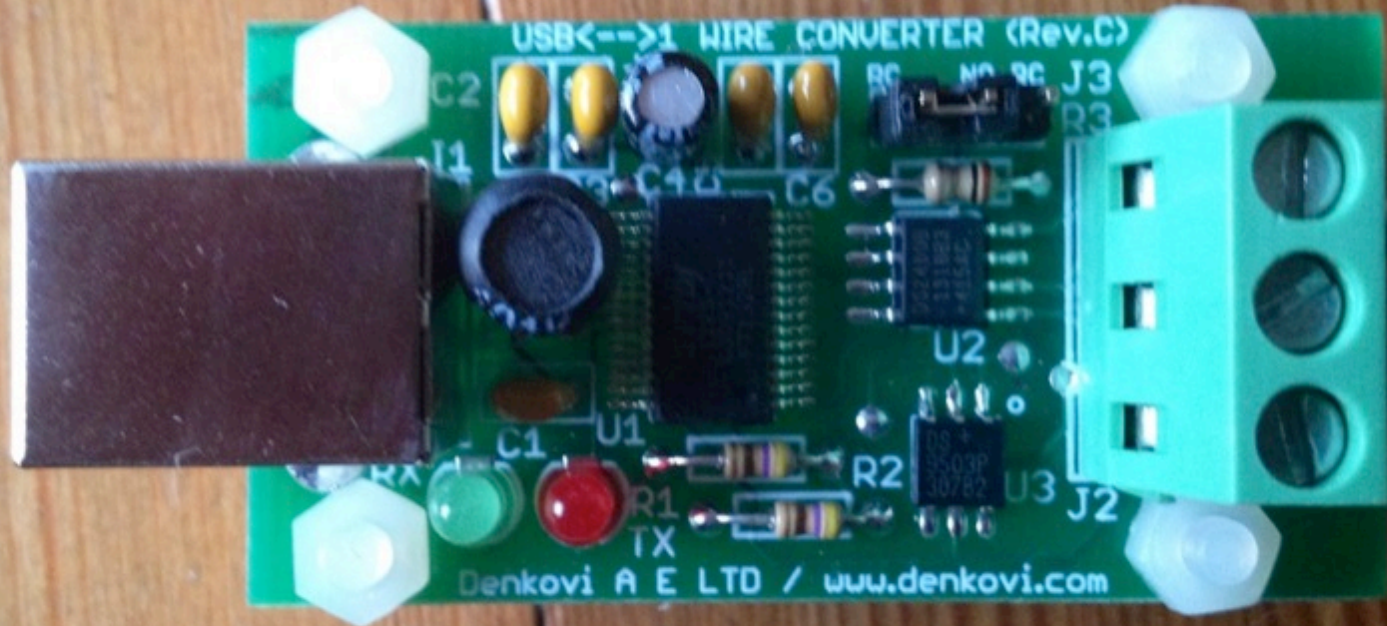


# Beispiel: Funk-Fernbedienung





# Beispiel: USB / 1-wire Converter





- IP-fähig
- Nutzung zur IP-basierten Sendung codierter Schaltsignale per *netcat* an 433MHz Empfänger
- Nur Funk-Transmitter, d.h. keinen Rückkanal

Stand Feb. 2014: Gerät ist nicht offenbar mehr neu erhältlich!

Alternativ evtl. Raspberry Pi mit GPIO nehmen (s. Folie „Weiterführende Literatur“)



# Beispiel: IP/433GHz-Gateway



- **Integration über *Exec binding***
- **PHP Skript**
  - Konfiguration der DIPs
  - Konfiguration IP/433GHz-Gateway (IP-Adresse)
  - Hersteller- und DIP-spezifische Generierung von Steuercodes
  - Ausführung des *netcat* Befehls
- ***PHP Skripte***
  - *Aus connair Software (<http://www.l3x.de/connair/>)*
  - *<http://simple-solutions.de/forum/viewtopic.php?t=260>*

# Aufbau, Konfiguration und Test-Szenarien

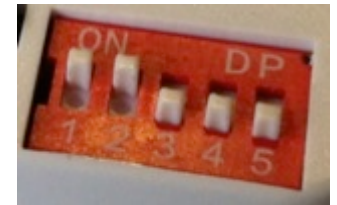
- **Codierung der Empfänger: DIP-Schalter**

Bsp.: 11110 | 01000

12345 | ABCDE



- **Optional: Codierung der Handsender (Fernbedienung, Funk-Schalter); DIPs beachten!**



- **Anschluss der Lampe**
- **Test mit Fernbedienung und Funkschalter**
- **Skript konfigurieren**

*„Houston, we're ready to go“*

- **Konfiguration der Items**
  - *Typ* *Bezeichner*
  - *< Icon >*
  - *( Gruppe(n) )*
  - *{ Binding-spezifische Konfiguration }*
- **Konfiguration der Sitemap**
  - Frames
  - Items
  - Gruppen

**Zeit für einen Test unter iOS...**



# „Und warum wurde jetzt ein Skript ausgeführt?“ – Übersicht der openhab Bindings

- Asterisk Binding
- Bluetooth Binding
- ComfoAirBinding
- CUPS Binding
- digitalSTROM Binding
- DMX512 Binding
- EnOcean Binding
- Epson Projector Binding
- Exec Binding
- Fritz!Box Binding
- Fritz AHA Binding
- Homematic Binding
- HTTP Binding
- IHC / ELKO Binding
- KNX Binding
- Koubachi Binding
- MAX!Cube Binding
- MiLight Binding
- Modbus TCP Binding
- MPD Binding
- MQTT Binding
- Network Health Binding
- Nibe Heatpump Binding
- Nikobus Binding
- Novelan/Luxtronic Heatpump Binding
- NTP Binding
- One-Wire Binding
- Onkyo AV Receiver Binding
- OpenSprinkler Binding
- OSGi Configuration Admin Binding
- Philips Hue Binding
- Piface Binding
- Plugwise Binding
- PLCBus Binding
- Pulseaudio Binding
- RFXCOM Binding
- Samsung TV Binding
- Serial Binding
- Snmp Binding
- Squeezebox Binding
- System Info Binding
- Somfy URTSI II Binding
- Sonos Binding
- TCP/UDP Binding
- TinkerForge Binding
- VDR Binding
- Wake-on-LAN Binding
- Z-Wave Binding

## • Inbetriebnahme

- Benötigt virtuellen COM Port Treiber, z.B. <http://www.ftdichip.com/Drivers/VCP.htm>
- Anbindung via OWFS (<http://owfs.org/>)
  - Installation über Paketmanager (z.B. Linux) oder Ports (Mac OS)
  - Getestet mit OWFS aus Homebrew (<http://brew.sh>)

## • Beispiel unter Mac OS X nach Installation und Anschluss des 1-wire Controllers

```
$ ll /dev/tty.*  
$ crw-rw-rw-  1 root  wheel  [...] /dev/tty.usbserial-DAE0001y  
$ owhttpd --debug -d /dev/tty.usbserial-DAE0001y -p 8001  
$ owserver -d /dev/tty.usbserial-DAE0001y
```

# Szenario 2: USB / 1-wire Converter

## Auswertung owhttpd

1-Wire Web: 28.5C09B6040000

localhost:8001/28.5C09B6040000

OWFS [Bus listing](#) [OWFS homepage](#) [Dallas/Maxim](#) by [Paul H Alfile](#)

**28.5C09B6040000**

[uncached version](#)

<a href="#">up</a>	directory
address	285C09B6040000B8
alias	<input type="text"/> <a href="#">CHANGE</a>
crc8	B8
errata	<a href="#">errata</a>
family	28
fasttemp	23
id	5C09B6040000
locator	FFFFFFFFFFFFFFFF
power	YES (1)
r_address	B8000004B6095C28
r_id	000004B6095C
r_locator	FFFFFFFFFFFFFFFF
scratchpad	70014B461FFF101071
<b>temperature</b>	<b>22.9375</b>
temperature10	23
temperature11	23
temperature12	22.9375
temperature9	23
temphigh	<input type="text" value="75"/> <a href="#">CHANGE</a>

- **Konfiguration der Items**
  - *Typ* *Bezeichner*
  - *< Icon >*
  - *( Gruppe(n) )*
  - *{ Binding-spezifische Konfiguration }*
- **Konfiguration der Sitemap**
  - Frames
  - Items
  - Gruppen

**Hands-on: Designer vs. iOS...**

# Szenario 3: XML-Daten einer Weather API

The screenshot shows the WunderMap website interface. The browser address bar displays [www.wunderground.com/wundermap/](http://www.wunderground.com/wundermap/). The page features a search bar with the text "Search & Favorites" and a magnifying glass icon. Below the search bar is a map of the Ruhr region in Germany, centered on Essen. The map shows several weather stations marked with green circles and numbered 7, 8, and 9. A sun icon is visible near station 7. A weather information panel is overlaid on the bottom right of the map, titled "Essen".

**Essen**

Aktuell    **Wettervorhersage**    Graph

Station ID: INORDRHE143    Lat: 51.44    Lon: 7.

Conditions	Temperatur
Sky conditions not available for this station	8 °C
	gefühlte 4 °C

# Szenario 3: Wetterdaten Beispiel (Auszug XML)

```
<?xml version="1.0" encoding="utf-8" ?>
  <current_observation>
    <credit>Weather Underground Personal Weather Station</credit>
    <credit_URL>http://wunderground.com/weatherstation/</credit_URL>
  [...]
    <observation_time_rfc822>Sat, 17 Aug 2013 16:22:03 GMT</observation_time_rfc822>
  [...]
    <temperature_string>85.5 F (29.7 C)</temperature_string>
    <temp_f>85.5</temp_f>
    <temp_c>29.7</temp_c>
    <relative_humidity>16</relative_humidity>
  [...]
    <wind_dir>North</wind_dir>
    <wind_degrees>350</wind_degrees>
    <wind_mph>4.0</wind_mph>
    <wind_gust_mph>6.0</wind_gust_mph>
  [...]
    <pressure_mb>1014.4</pressure_mb>
  [...]
  </current_observation>
```

Weitere Infos zur API:

<http://deutsch.wunderground.com/weather/api/d/docs>



- **XSL-Transformationen werden im Unterordner *transform* abgelegt**
- **Hier: Extraktion des numerischen(!) Wertes aus `/current_observation/temp_c`**
- **Die Maßeinheit (hier: „°C“) ist dann über die Visualisierung (*sitemap*) hinzuzufügen!**

```
<?xml version="1.0"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:yweather="http://xml.weather.yahoo.com/ns/rss/1.0" version="1.0">
  <xsl:output indent="yes" method="xml" encoding="UTF-8" omit-xml-declaration="yes" />
  <xsl:template match="/">
    <xsl:value-of select="/current_observation/temp_c" />
  </xsl:template>
</xsl:stylesheet>
```



# Visualisierung in openHAB

- **Eindimensionale Daten**

- Beispiele

- „Außen-Temperatur: 5°C“
- „Außen-Temperatur (mit Chillfaktor): 5°C (2°C)“ (Faltung)

- **Mehrdimensionale Daten**

- Beispiele

- Zeitreihen ( $f_1(t)$ )
- Multi-Sensoren-Messungen ( $f_1(t)$ ,  $f_2(t)$ )

- **Grafische Darstellung**

- Icons, Widgets
- Eigene Grafiken

## • Ausgabeschnittstellen

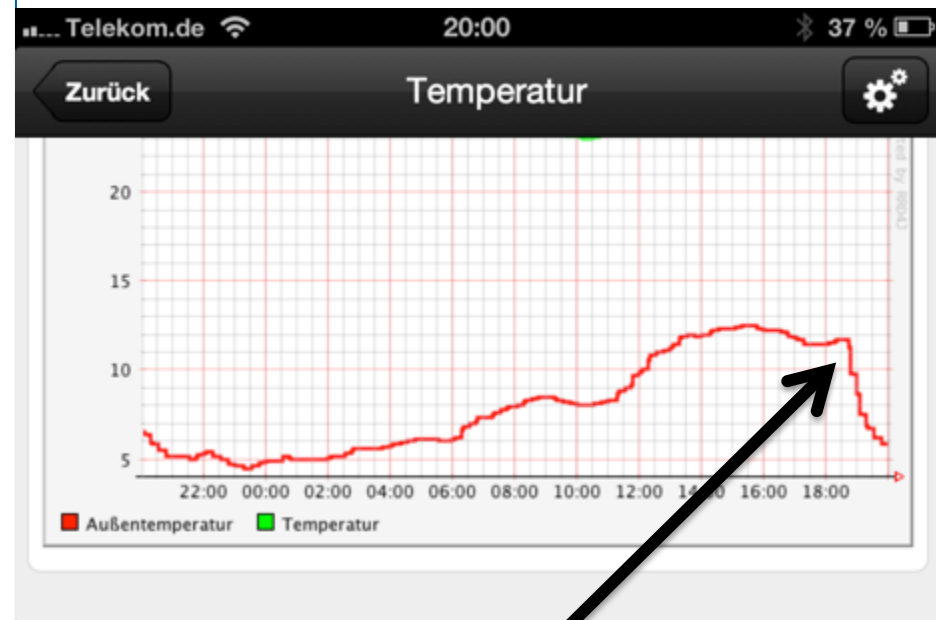
- „Classic UI“
- „GreenT“
- Native APP (iOS, Android)
- Comet Visu (ab 1.4)

## • Widgets

- <http://demo.openhab.org:8080/openhab.app?sitemap=demo>

## • Zeitreihen

- Über Persistenzlayer (z.B. Temperaturhistorie)
- Derzeit: db4o, **rrd4j**, sql, Sen.se, Cosm,...



Orkantief „Anna“ am 03.01.14

- **Ziel**

- Visualisierung der Windrichtung aus Szenario 3 mittels einer SVG-Windrose

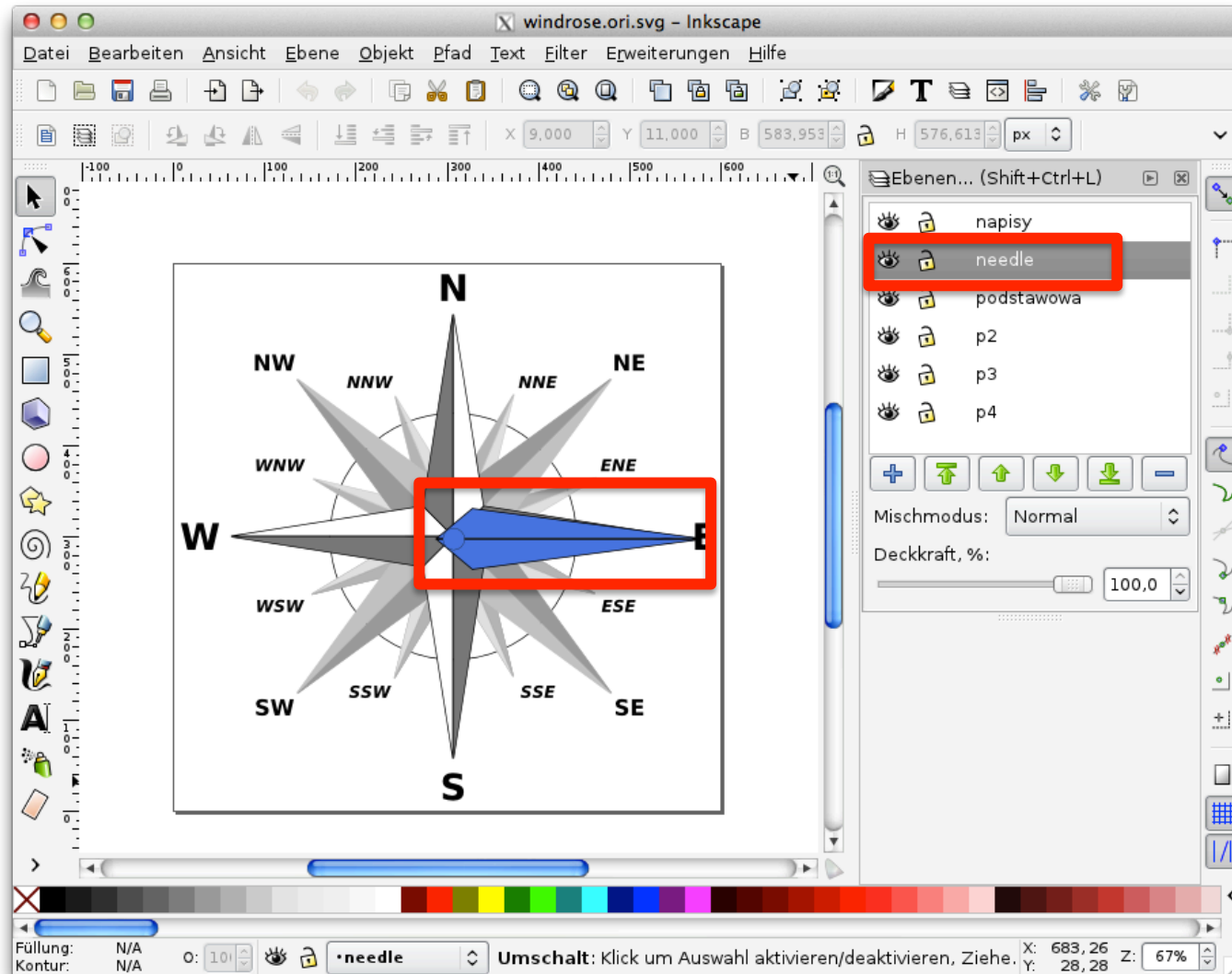
- **Warum Vektorgrafik (SVG)?**

- Grafik skaliert besser (z.B. für Tablet-User)
- Leichtes Einbringen dynamischer Daten in den Code
- Optional: Export in pixelbasierte Bilddatei möglich

- **Tools**

- Inkscape
- SVG-Datei einer Windrose
- XSLT/PHP/Apache (alternativ andere web-basierte XSLT Umgebung)
- openHAB (Integration als Web-Grafik / *image* )

# Zu Szenario 4: Modellierung der zu rotierenden Ebene



# Zu Szenario 4: Analyse des Inputs und Umsetzung

```
. dasharray:none;display:inline"
310     d="m 273.8573,266.28357 32.8331,32.8331 0,-0.016 0,-242.85201
    -32.8331,210.03488" />
311 </g>
312 <g
313   style="display:inline"
314   inkscape:label="needle"
315   id="g3058"
316   transform="rotate(90,306,302)"
317   inkscape:groupmode="layer">
318 <g
319   id="g3942">
320 <g
321   transform="matrix(1,
322   id="g3918">
323 <path
324   d="m 340.5235,266.
```

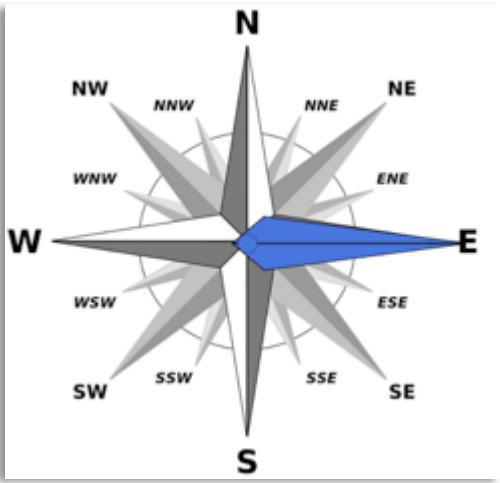
```
<wind_dir>NORTH</wind_dir>
<wind_degrees>350</wind_degrees>
<wind_mph>4.0</wind_mph>
<wind_gust_mph>6.0</wind_gust_mph>
```

XML aus Szenario3

SVG-Datei

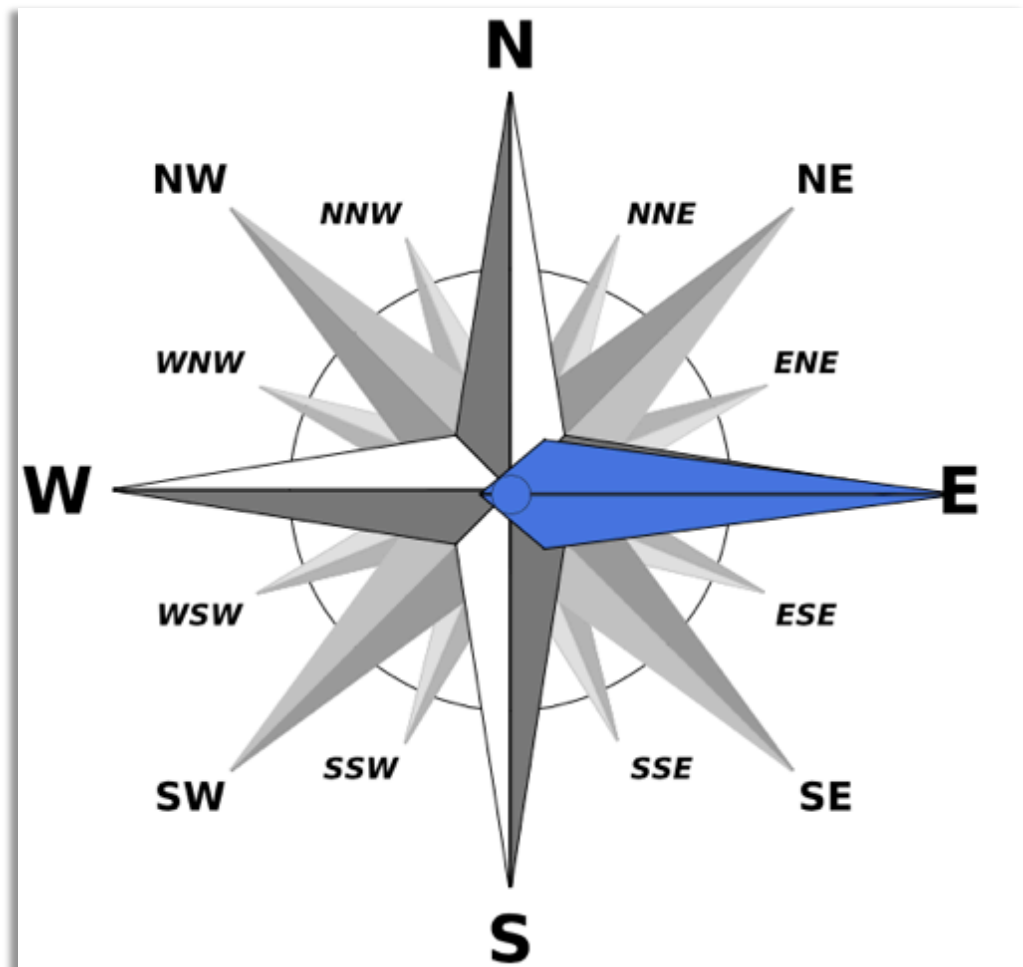
XSLT

```
<?php
$dom = new DOMDocument('1.0', 'utf-8');
// Lade SVG-Template
$dom->load('windrose.svg');
// Lade Sensorik-XML
$dom2 = new DOMDocument();
$dom2->load('u.xml');
$xml = new DomXPath($dom);
// Finde Gruppen-ID der SVG-Ebene (individuell!)
$res = $xml->query("//*[@id = 'g3058']");
// Extrahiere Parameter aus Sensorik-XML
$angles = $dom2->getElementsByTagName('wind_degrees');
foreach ($angles as $angle) {
    $rot = $angle->nodeValue;
}
// Ersetze Rotationswinkel von g3058 (hier: Ebene 'needle')
$value='rotate('.$rot.',306,302)';
$res->item(0)->setAttribute('transform',$value);
// SVG Ausgabe
header('Content-type: image/svg+xml; charset=utf-8');
echo $dom->saveXML();
?>
```



- **Test...**

- Öffentliche Wetterdaten
- Lokaler Webserver





# Regelwerke

- Bsp.-Regel: „Wind aus Norden => Lampe an!“
- Lampe (Szenario 1), Windwinkel (Szenario 3)
- Regeldatei default.rules

```
rule "Wind from the North"
when
    Item Weather_Wind_Degrees changed or
    Time cron "0 0 0 * * ?" or
    System started
then
    if (Weather_Wind_Degrees.state>315 || Weather_Wind_Degrees.state<45 ) {
        E02.sendCommand(ON)
        pushNotification("Breaking News","Wind aus nördlicher Richtung!")
    } else {
        E02.sendCommand(OFF)
    }
end
```

# Ausblick



- **Idee: Wenn Smartphone im Hausnetz/WiFi „auffindbar“, dann: setze Haus in Zustand X**
- **Umsetzung abhängig von**
  - Netzstruktur
  - OS (Phone)
  - Access Point
  - Bauliche Gegebenheiten
  - ...
- **Mögliche Ansätze der „Erkennung“**
  - Bluetooth, Wifi, NFC, ...

- **Raspberry Pi + Funkmodul am GPIO**
- **Push notifications via Prowl API**
- **VPN mit fixen IP-Adressen über Network Health monitoren**
- **IP Kameras über API in Alarm-Modus setzen**
- **IP Kameras über Funk-Stromstecker „still legen“**
- **Geräte mit hohem Energieverbrauch identifizieren und *smart* schalten**
- **u.v.m.**



- **Thema: Sicherheit (Weites Feld!)**
  - Nicht nach außen freigeben
  - Zugang von extern nur via VPN
  - Gästernetzwerk bzw. dezidiertes HA-Netz
- **Ausfallsicherheit: Redundanzen bzw. Workarounds berücksichtigen**
- **Verhalten bei Ausfall des OpenHAB-Servers testen 😊**
- **Weitere Infos / Anregungen**
  - Foren checken (openHAB, KNX-User, smart-solutions.de,...)
  - [www.steckerchecker.de](http://www.steckerchecker.de)

**Vielen Dank!**

- **Openhab**

- <http://www.openhab.org/>
- <https://github.com/openhab/openhab>
- <https://github.com/openhab/openhab/wiki>

- **Weiteres**

- <http://www.l3x.de/connair/>
- <http://simple-solutions.de/forum/viewtopic.php?t=260>
- D.Bachfeld: Ein Pi, sie zu knechten – Raspberry pi steuert Funksteckdosen, c't Heft Nr.3, 2014, S.101
- <http://www.steckerchecker.de>

-  25.10.2013 - Andreas Bischoff  
**ARM für Raspberry Pi, Phone Tablet und Server**
-  22.11.2013 – Burkhard Wald:  
**Die Idee des Jahres 2013: Kommunikation verschlüsseln**
-  20.12.2013 – Andreas Michels  
**Menschenbilder im Informationszeitalter**
-  24.01.2014 – Stefan Helker  
**Entwicklung einer mobilen Webanwendung**
-  21.02.2014 – Daniel Biella  
**Home automation und Datenvisualisierung**
-  21.03.2014 – Sandrina Heinrich & Steffi Engert  
**iPad in der Lehre**

14:00 Uhr  
Duisburg LE 105