INTERNET OF THINGS CONFERENCE

Christian Götz & Dominik Obermaier | dc-square

# Bau deine eigene IoT Cloud

# Wer sind wir?

**CHRISTIAN GÖTZ**
@goetzchr

**DOMINIK OBERMAIER**
@dobermai

INTERNET OF THINGS CONFERENCE

# dc-square

HiveMQ

HiveMQ

HiveMQ Plugin Entwicklung

IoT Beratung

Workshops & Schulung

# Ziel des Workshops

## Architektur einer IoT Plattform

## Implementieren eines Prototypen

# Endergebnis

# Entwicklungumgebung

## Was jeder installiert haben sollte

- Java JDK 7 ✔
- Java IDE (IntelliJ, Eclipse, Netbeans, vi, ...) ✔
- GIT ✔
- Maven ✔
- HiveMQ 2.0.2 ✔
- MySQL Datenbank ✔
- MySQL Datenbankviewer ✔
- Aktueller Webbrowser ✔

# GIT Repository

clone

https://github.com/dc-square/build-your-own-iot-cloud-workshop

oder

**USB-Stick**

INTERNET OF THINGS CONFERENCE

# IoT Cloud Platform

INTERNET OF THINGS CONFERENCE

# Internet der Dinge

INTERNET OF THINGS CONFERENCE

# Technologie die Geräte über Kabel oder Drahtlos vernetzt

INTERNET OF THINGS CONFERENCE

# IoT Cloud Platform

Komponenten

**Live Daten von Sensoren**

HTTP, MQTT, CoAP, ...

**Data Storage**

Datenbank mit historische Daten

**Identity Management**

User, Devices

**IoT Cloud**

**Device Management**

Configure, Update Firmware, Monitor, Statistics

**Administration & Betrieb**

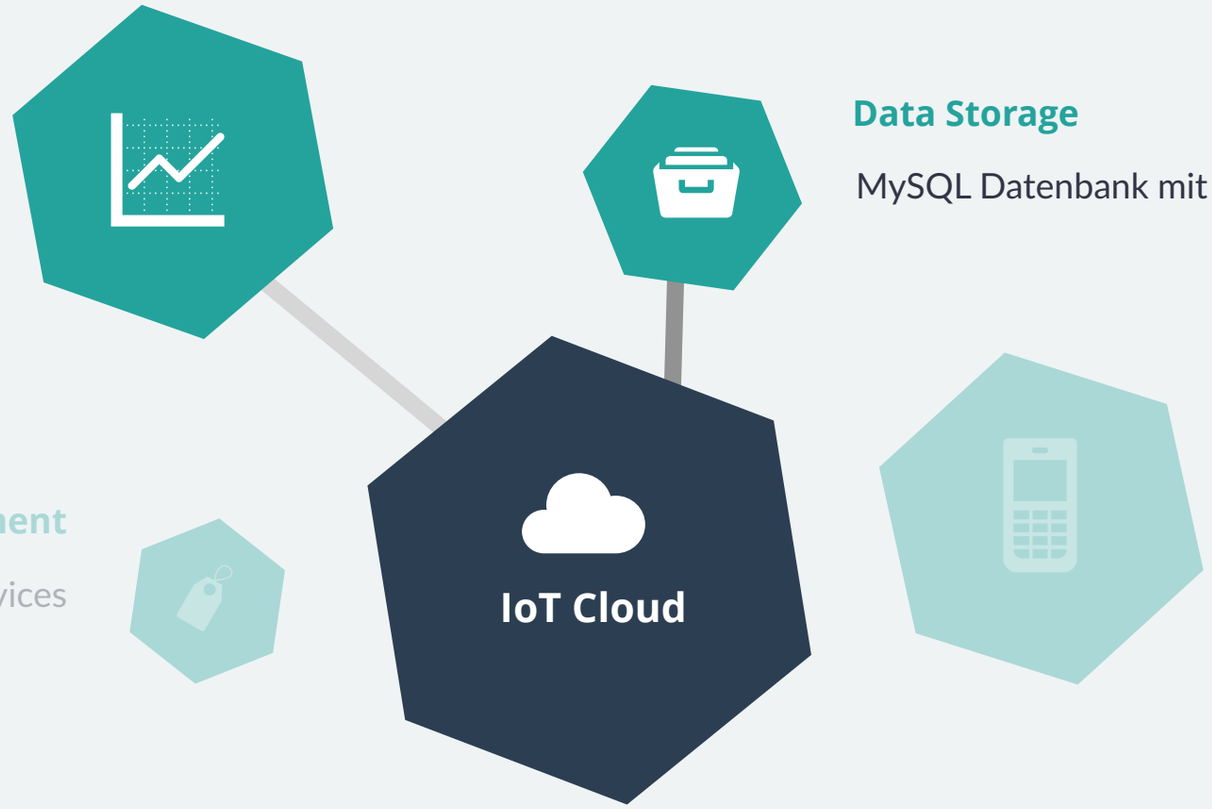
Web UI, Einfache Bedienung

**Security**

Transportsicherheit, Authentifizierung, Autorisierung

# IoT Cloud Platform

Komponenten, die wir heute benutzen




**Live Daten von Sensoren**

HTTP und MQTT

**Data Storage**

MySQL Datenbank mit REST API

**Device Management**

Configure, Update Firmware, Monitor, Statistics

**Identity Management**

User, Devices

**Administration & Betrieb**

Web UI, Einfache Bedienung

**Security**

Transportsicherheit, Authentifizierung, Autorisierung

# Technologien

Prototyp




Eclipse Paho, HTTP **Geräte**

HiveMQ **MQTT Broker**

Dropwizard **REST API**

HTML/JS **Web App**

GOAL

**IoT Platform Prototyp**

# Technologien

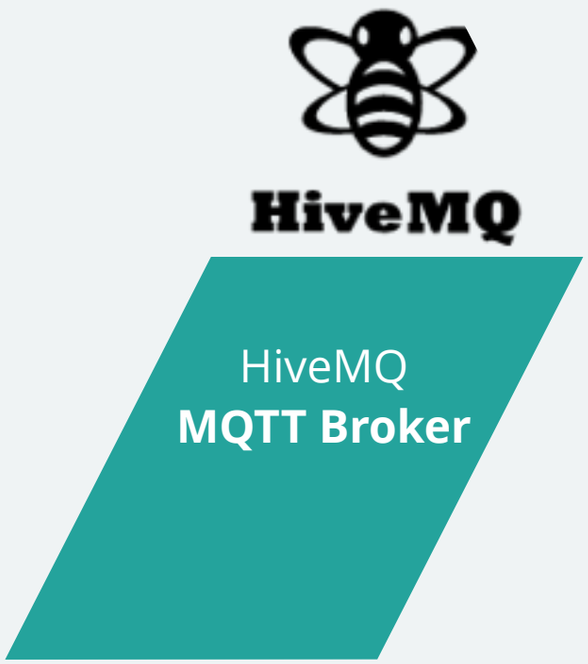
Geräte

Eclipse Paho,
HTTP
**Geräte**

- **Wetterstationen liefern Live Daten per Push (MQTT)**
  - ➡ Simulation der Wetterstationen mit *Eclipse Paho*
- **manche Geräte liefern Daten per HTTP**

# Technologien

HiveMQ




- **Austausch von Daten zwischen Geräten**
- **Persistieren der Daten in die Datenbank**

# Technologien

Dropwizard

Dropwizard
**REST API**

- **Historische Daten zur Verfügung stellen**
- **eingehende POST Requests an HiveMQ weiterleiten**

# Technologien

Webapp


HTML/JS
**Web App**

- **Anzeige von Live Daten über MQTT**
- **Anzeige von historischen Daten über REST API**

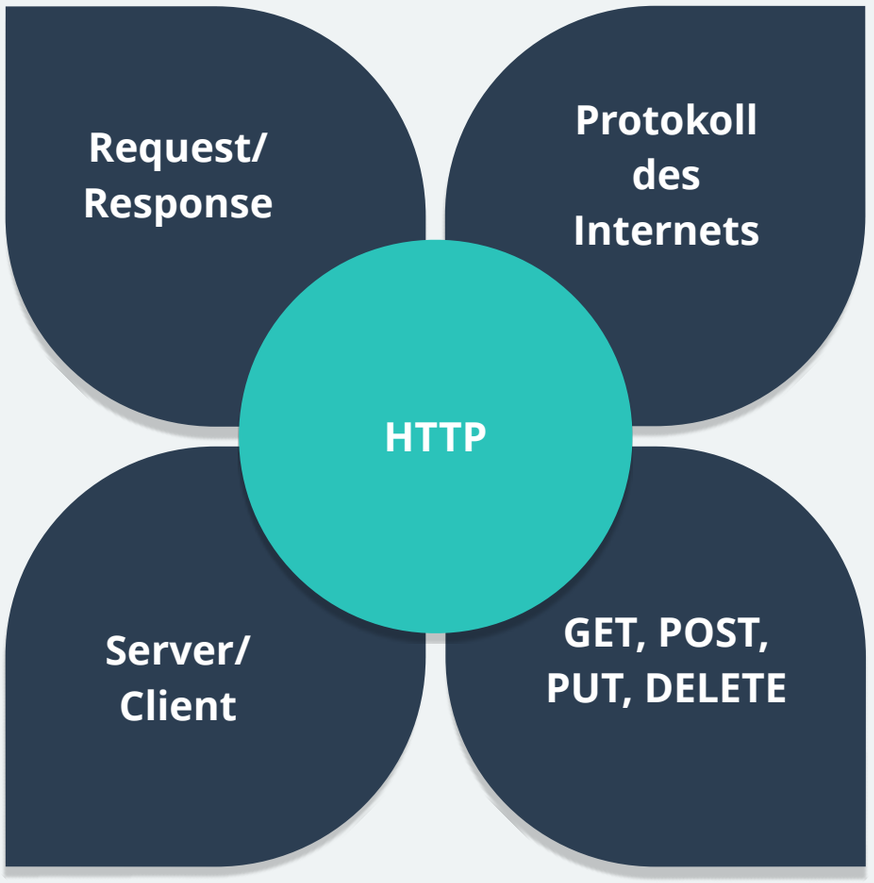
INTERNET OF THINGS CONFERENCE

# Teil 1: Geräte

# MQTT, HTTP

# HTTP

Fakten


Request/
Response
Protokoll
des
Internets
HTTP
Server/
Client
GET, POST,
PUT, DELETE

# HTTP

Vorteile/Nachteile

- ✓ Bekannt und Bewährt
- ✗ Polling
- ✗ nicht bidirektional
- ✗ Header ist verbose
- ✗ Keine Garantien
- ✗ Stateless

# MQTT

Kurzübersicht

**Einfach**

Connect, Publish, Subscribe, Unsubscribe, Disconnect

**Messaging Protokoll**

Austausch von Nachrichten, teilweise Queuing

**basiert auf TCP**

Port 1883

**Publish/Subscribe**

Clients abonnieren auf Topics auf denen andere Clients Nachrichten veröffentlichen

**Minimaler Overhead**

teilweise nur 2 Byte

**Ausgelegt für unstabile Netze**

individuelle Garantien möglich für jede Nachricht

# MQTT History

Timeline

1999

**Entwickelt**
IBM/Arccom

2011

**MQTT released**
royalty free

2013

**Eclipse IoT (M2M)**

2014

**OASIS Standard**

2015

# MQTT - Topics

Feature

1

**iotcloud/device1**

> einfacher Topic

2

**iotcloud/device1/temp**

> einfacher Topic

3

**iotcloud/+/temp**

> Single Wildcard

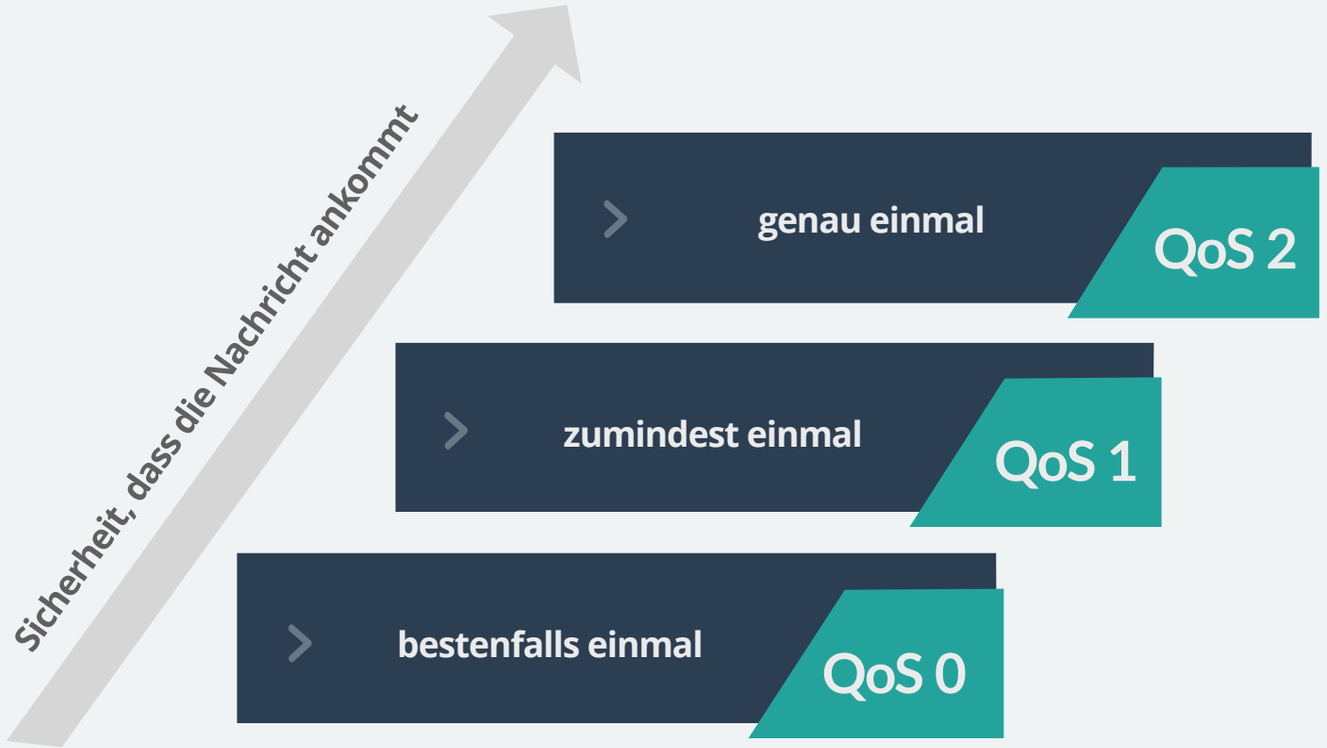
4

**iotcloud/#**

> Multi-level Wildcard

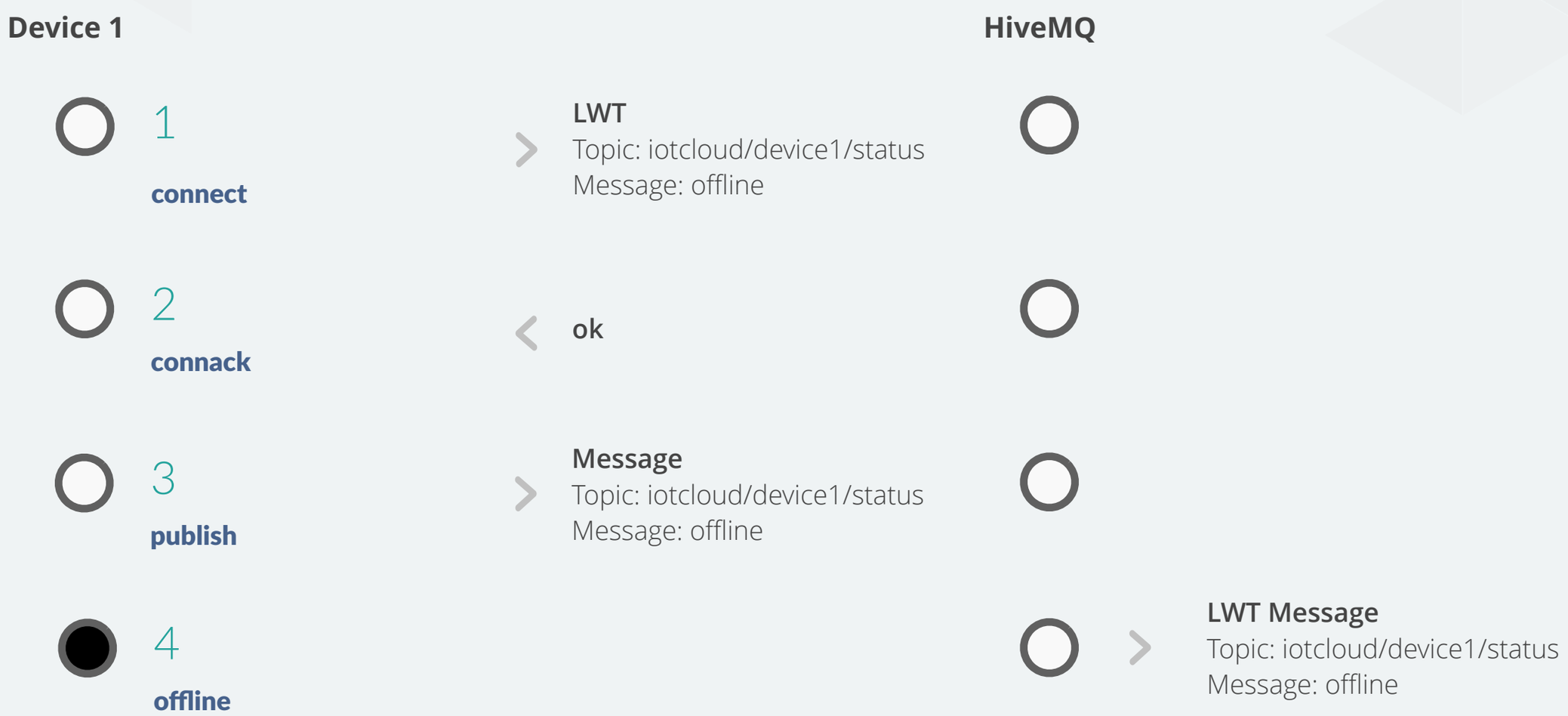
# MQTT - Quality of Service

Feature



# MQTT - LWT

Feature

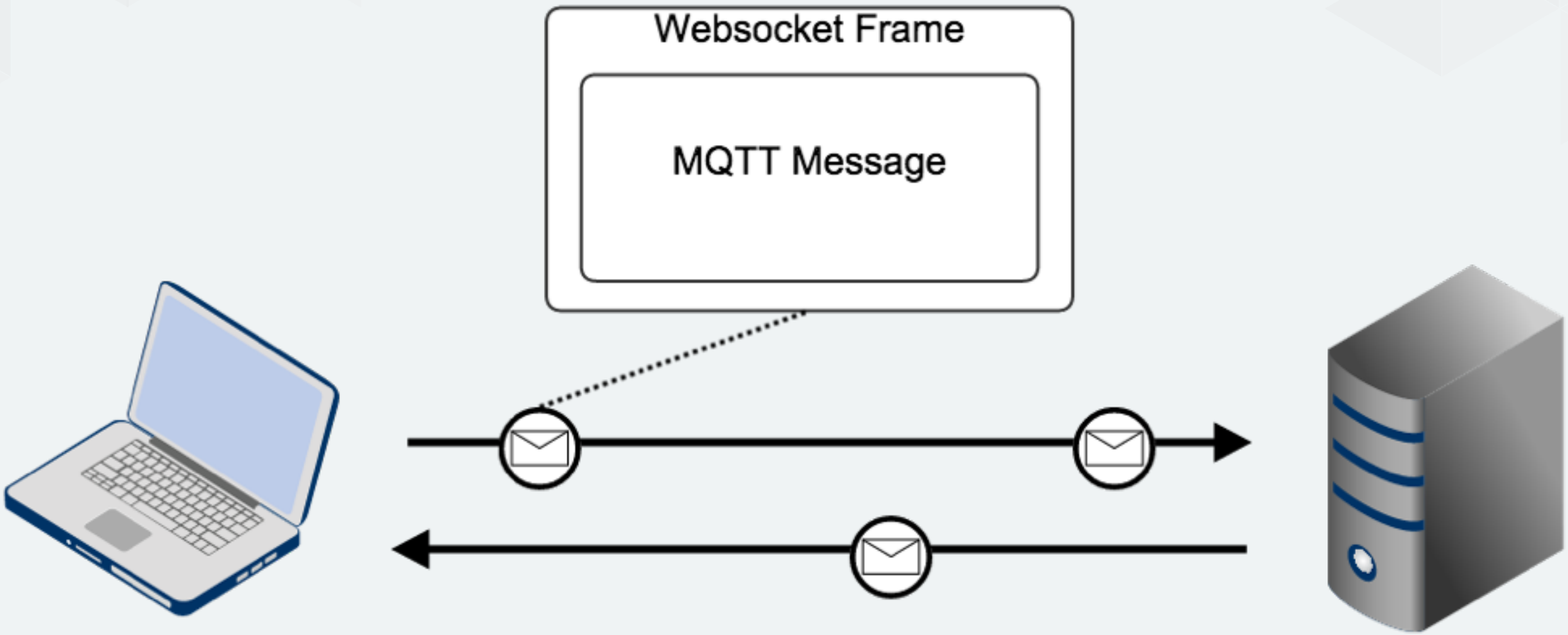


# MQTT - Retained Msg

Feature

# MQTT over Websockets

Feature



# MQTT Broker

Kurzübersicht

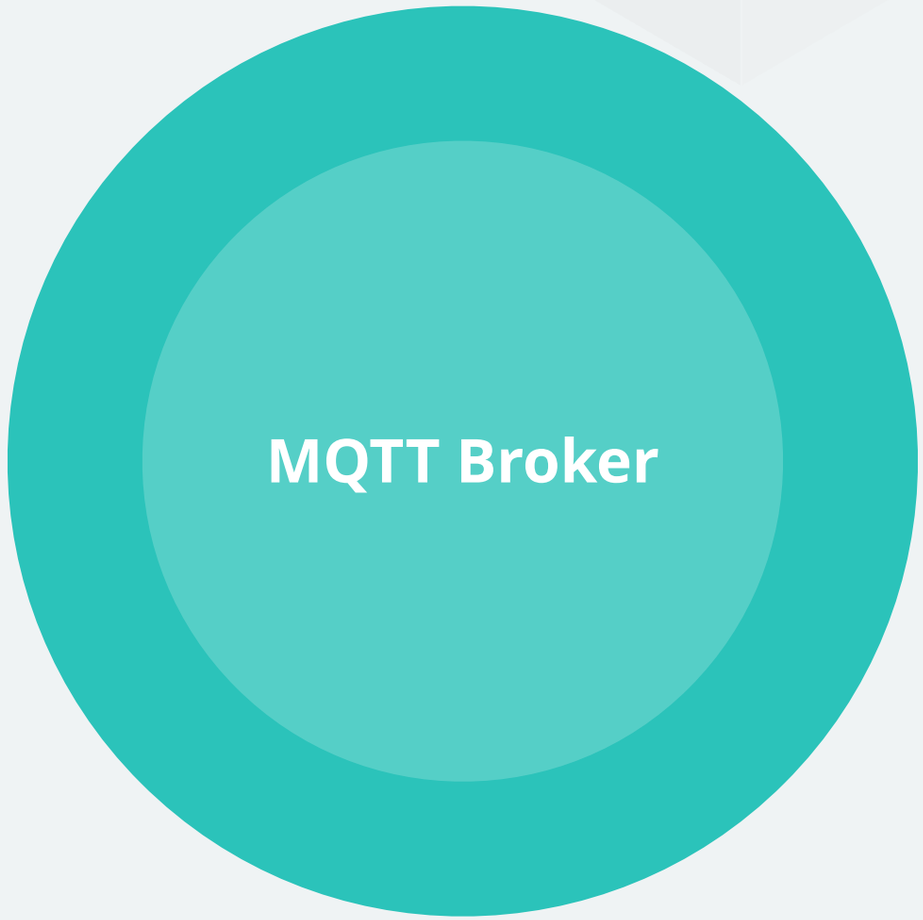
*Herzstück von MQTT*

*Versenden der Nachrichten an Clients*

*Benutzt nur Topics*

*Topics sind dynamisch*

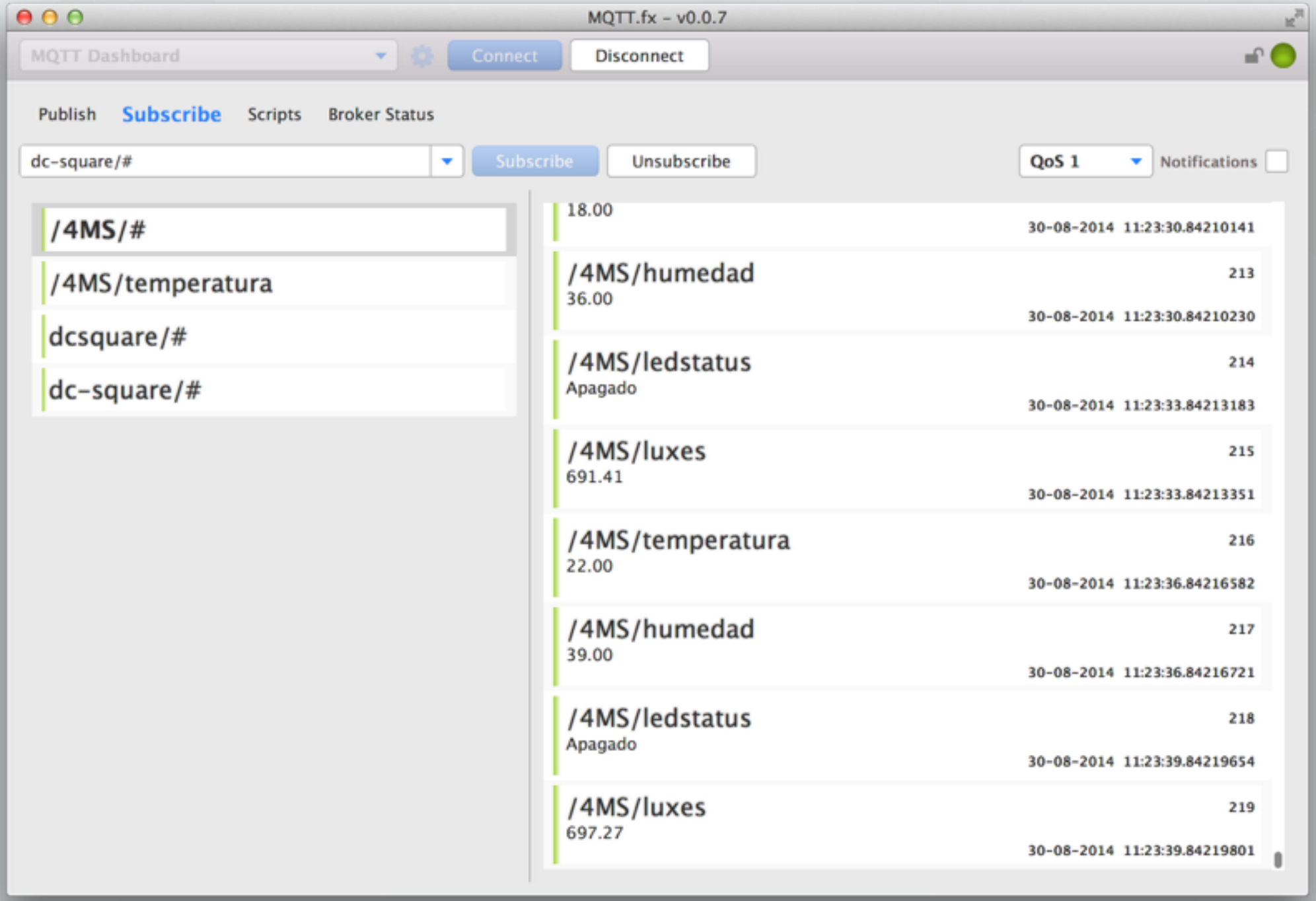
*Benutzerdefinierte Funktionen*



INTERNET OF THINGS CONFERENCE

# MQTT GUI Tools

# MQTT.fx 0.0.7

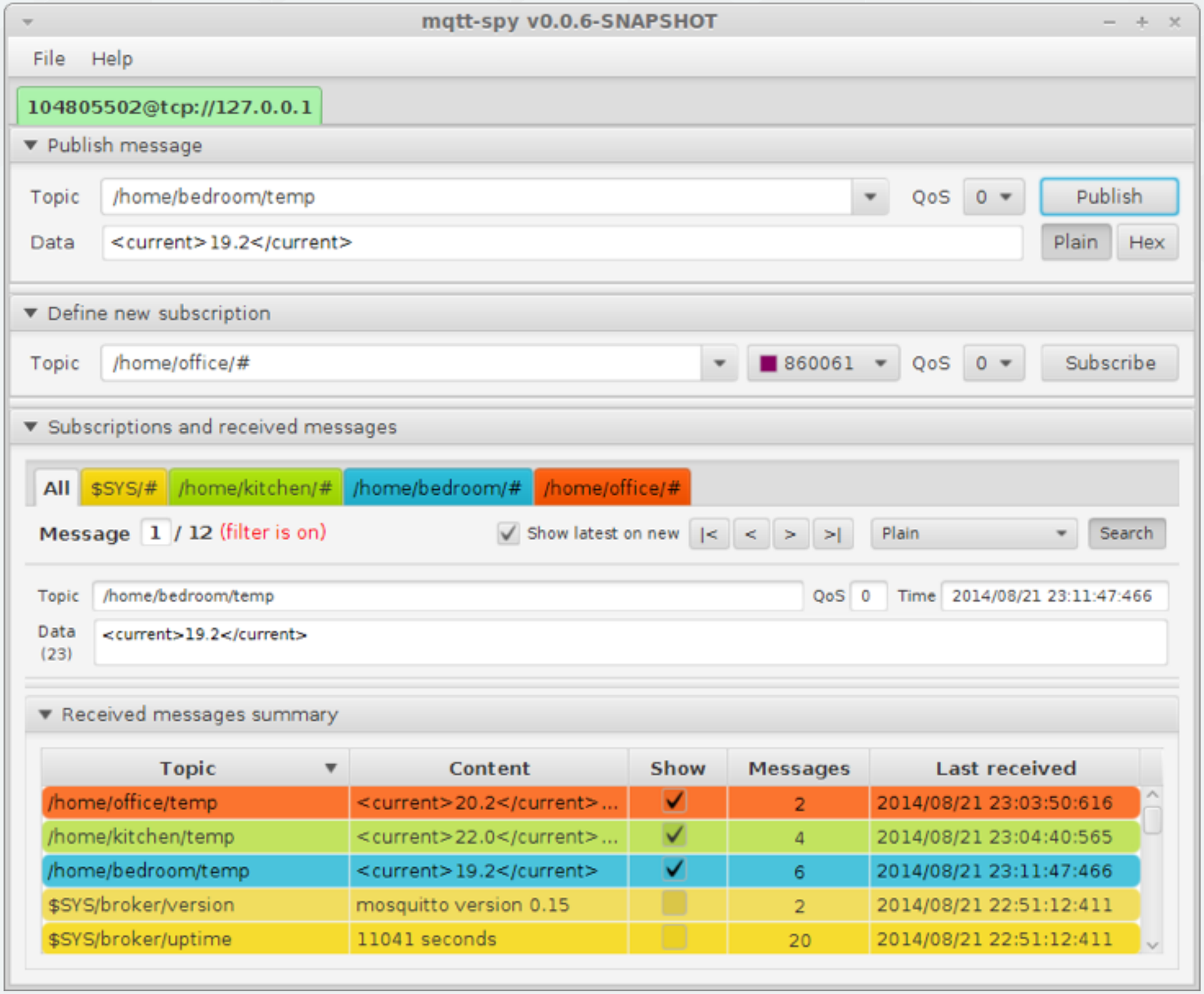



http://www.jensd.de/wordpress/?p=1423

INTERNET OF THINGS CONFERENCE

# MQTT GUI Tools

# MQTT spy 0.0.6




https://code.google.com/p/mqtt-spy/

INTERNET OF THINGS CONFERENCE

# MQTT CMD Tools

# mosquitto_pub/_sub

# Mosquitto Clients

Publish/Subscribe

```
$ mosquitto_pub -h broker.mqttdashboard.com
-t iotcloud/test -m “test” -q 1
```

http://mosquitto.org/man/mosquitto_pub-1.html

—

```
$ mosquitto_sub -h broker.mqttdashboard.com
-t iotcloud/test
```

http://mosquitto.org/man/mosquitto_sub-1.html

INTERNET OF THINGS CONFERENCE

# MQTT Libraries

# Eclipse Paho


Open Source

“Reference Implementation”

Many languages: Java, Javascript, Lua, C, C++, Go, Python

Active Community

JS Library uses MQTT over Websockets

INTERNET OF THINGS CONFERENCE

# MQTT Libraries

# FuseSource MQTT Client

# MQTT-Client

Open Source

3 API Styles

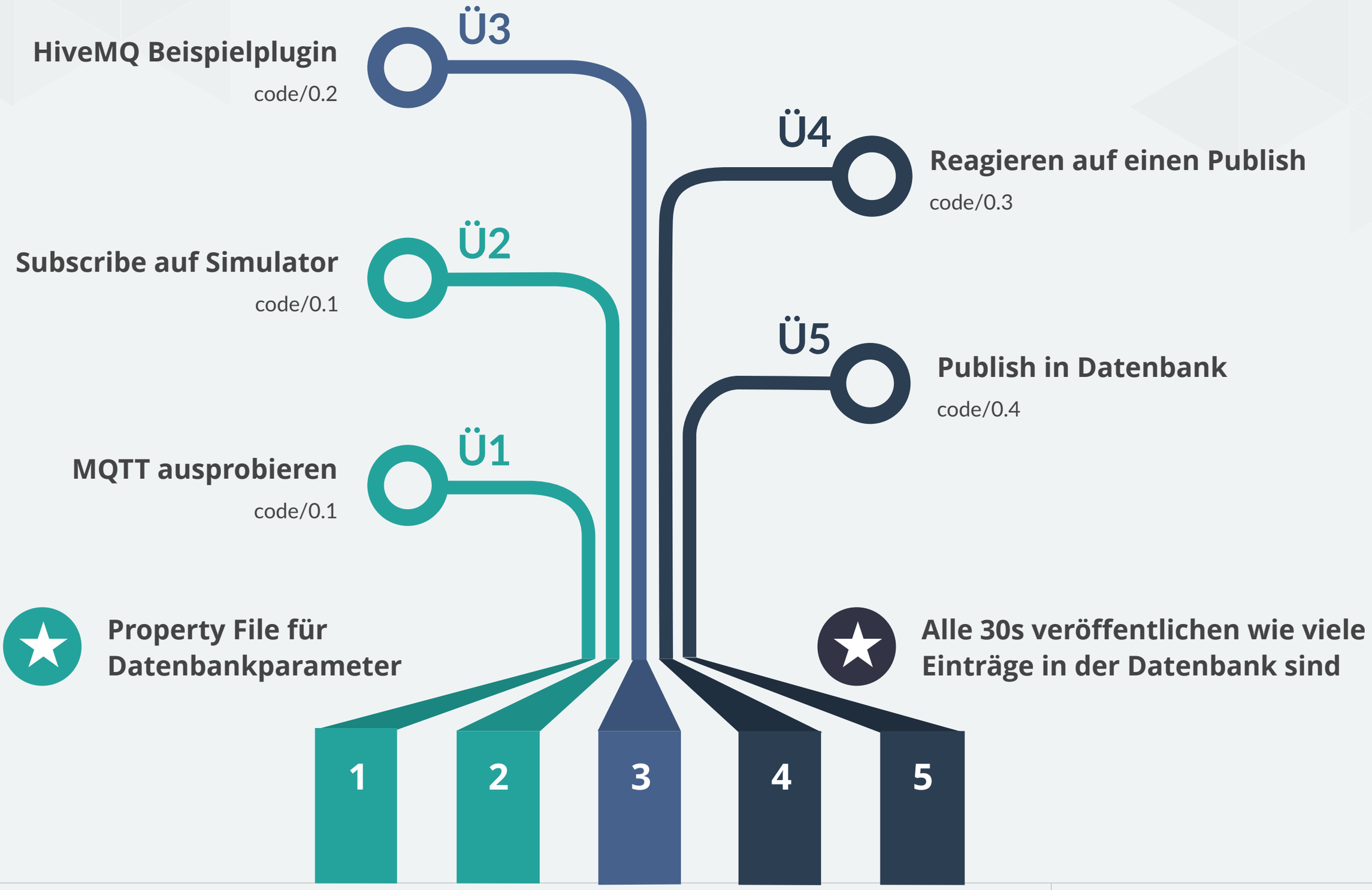
Automatic Reconnect

Maven Central

Less active Community

# MQTT + HiveMQ

Hands-on



INTERNET OF THINGS CONFERENCE

# Übung 1

MQTT ausprobieren

Version

**0.1**

**STEPS**

## HiveMQ starten

**Unix:** /bin/run.sh
**Win:** run.bat *starten*

## Websocket Client

**mqtt-websocket-client/index.html** öffnen

## Verbinden

**Verbinden**
Erfolgreich wenn HiveMQ ClientId ausgibt

## Subscribe

**Subscribe zu einfachem Topic**
**Subscribe zu Wildcard Topic**

## Publish

**Publish von Nachrichten auf verschiedenen Topics**

# Übung 2

Simulator benutzen und Daten von Gerät 2 empfangen

Version

**0.1**

**STEPS**

## HiveMQ starten

**Unix:** /bin/run.sh
**Win:** run.bat *starten*

## Simulator starten

***simulator* in Java IDE** öffnen
Ausführen von **DeviceSimulator**

## Websocket Client

**mqtt-websocket-client/index.html** öffnen

## Verbinden

**Verbinden**
Erfolgreich wenn HiveMQ ClientId ausgibt

## Subscribe

**Herausfinden auf welchem Topic Gerät 2 schickt**

INTERNET OF THINGS CONFERENCE

# Teil 2: MQTT Broker

# HiveMQ

INTERNET OF THINGS CONFERENCE

# HiveMQ

MQTT Broker

**Hochskalierbarer MQTT Broker**

> 100.000 Verbindungen

**Offenes Plugin System**

Einhängen von benutzerdefinierter Logik, Integration in bestehende Systeme

**Cloud ready**

Amazon Web Service, Microsoft Azure, Own Datacenter

**Implementiert den MQTT Standard**

Gleichzeitiges Verbinden von MQTT Clients der Version 3.1.1 und 3.1 möglich

**Sicherheit**

TLS Verschlüsselung, X.509 Zertifikate, Benutzerdefinierte und feingranulare Authentifizierung und Autorisierung

**Einfache Benutzung und Betrieb**

$SYS Topics, Ausführliche Dokumentation, Monitoring über JMX und Graphite

# Plugin System

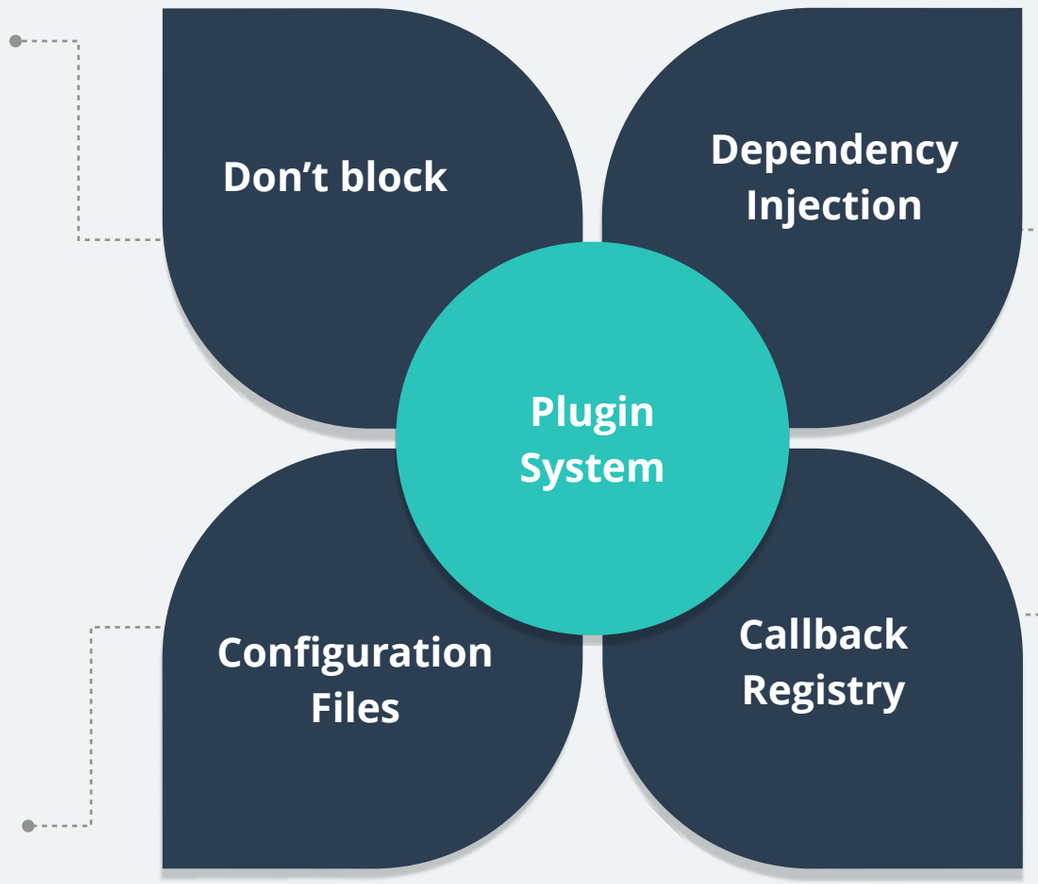
Hands-on

**Don’t block in a Plugin. Never.**

Andere Threads benutzen

**Inject everything!**

Steigert die Testbarkeit!




**Apache Configuration Support**

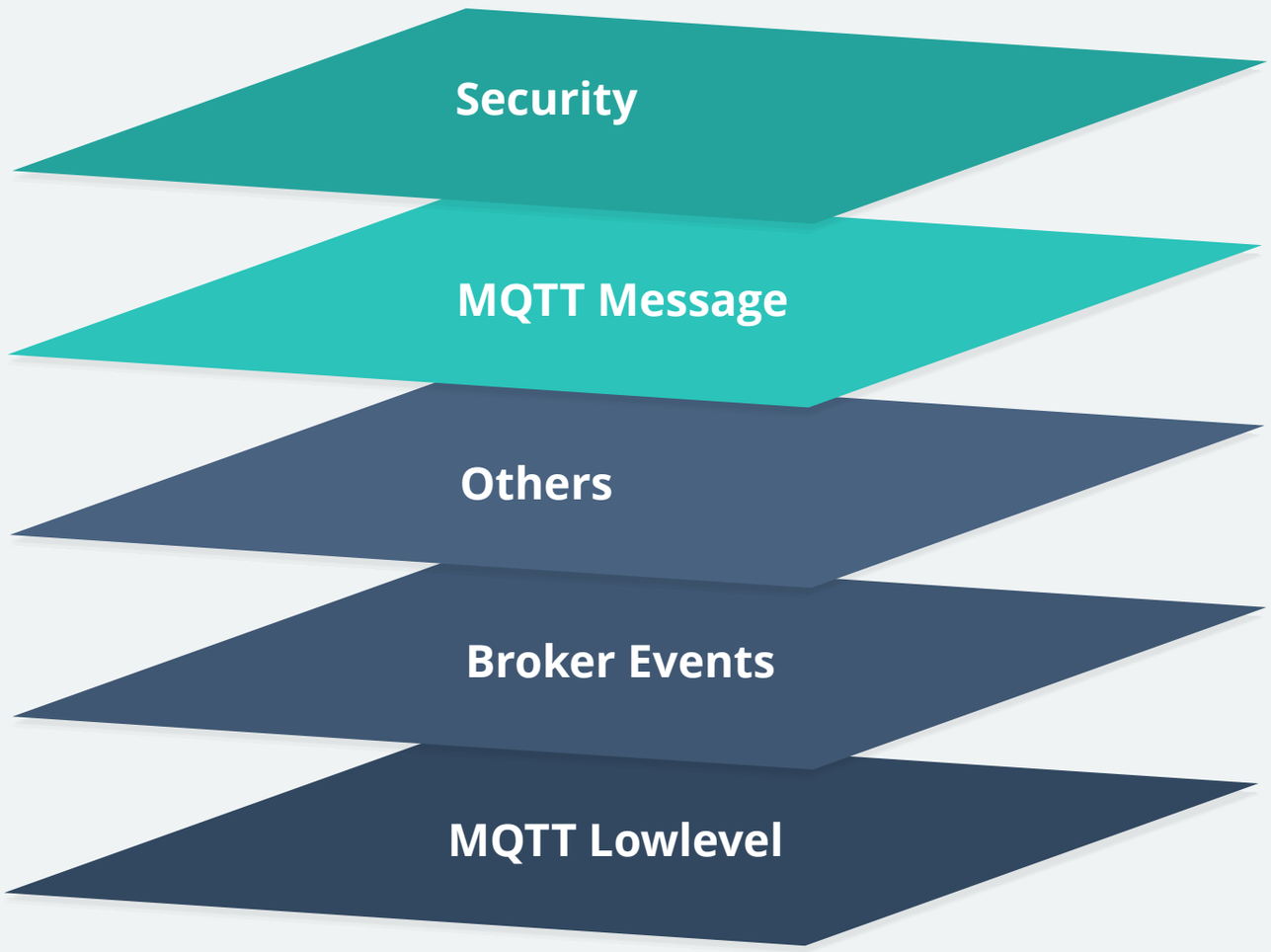
Einfach Config Files erstellen und automatisch nachladen

**Callback bekannt machen**

Callback muss hier registriert werden, ansonsten wird er nicht ausgeführt

# Plugin Callbacks

Einhängen von benutzerdefinierter Logik




1. Client Connect, Client Publish/Subscribe, After Login,
2. Connect, Publish, Subscribe, Unsubscribe, Disconnect
3. Scheduled
4. Start, Stop, Statistiken aktualisiert
5. PubAck, PubRel, PubComb, Ping, ...

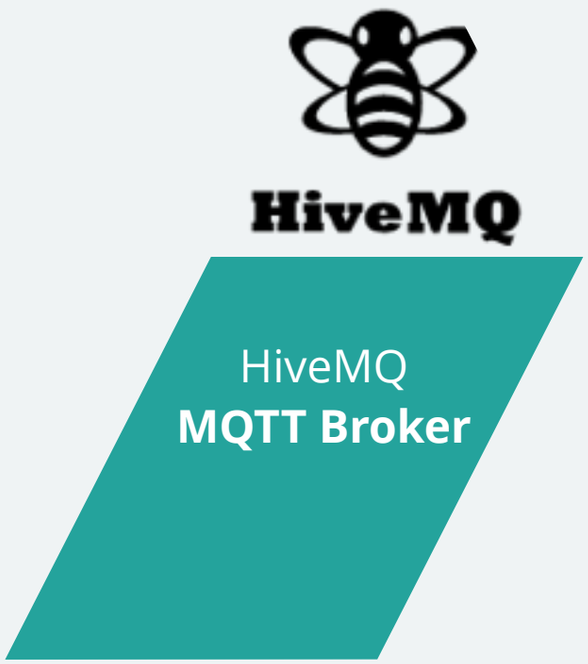
# Plugin Entwicklung

Tools für den Entwickler

- HiveMQ SPI
  - Vorraussetzung um entwickeln zu können
- Maven Plugin
  - Plugin mit HiveMQ starten
  - Debuggen im Server/Client Modus
- Assembly Plugin
  - Erstellung eines auslieferbaren Archiv
- JavaDoc

# Technologien

HiveMQ




- **Austausch von Daten zwischen Geräten**
- **Persistieren der Daten in die Datenbank**

# Übung 3

Plugin mit HiveMQ starten

Version

0.2

STEPS

- hivemq-plugin
  - ***hivemq-plugin* in Java IDE** öffnen
- HiveMQ Pfad
  - Ordner von HiveMQ in ***pom.xml*** setzen
- HiveMQ starten
  - HiveMQ mit Plugin über Maven ***package*** starten
    Profil ***RunWithHiveMQ*** aktivieren

# Übung 4

Benutzerdefinierte Logik für Publish

Version

0.3

STEPS

- **hivemq-plugin** > ***hivemq-plugin*** **in Java IDE** öffnen
- **HiveMQ Pfad** > Ordner von HiveMQ in ***pom.xml*** setzen
- **OnPublish** > Ausgabe auf der Console von ClientId, Topic und Nachricht
- **HiveMQ starten** > HiveMQ mit Plugin über Maven ***package*** starten
  Profil ***RunWithHiveMQ*** aktivieren

# Übung 5

Schreiben in die Datenbank

Version

0.4

STEPS

- **hivemq-plugin**
  - ***hivemq-plugin*** **in Java IDE** öffnen
- **HiveMQ Pfad**
  - Ordner von HiveMQ in ***pom.xml*** setzen
- **OnStart**
  - Datenbankverbindung aufbauen
- **OnPublish**
  - ClientId und Payload in Datenbank schreiben
- **HiveMQ starten**
  - HiveMQ mit Plugin über Maven ***package*** starten
    Profil ***RunWithHiveMQ*** aktivieren

# Verbesserungen

HiveMQ Plugin

- Connection Pool für Datenbankverbindung
- JDBC in extra Thread
- ORM-Framework einsetzen
- Authentifizierung und Autorisierung der Clients
- Transportverschlüsselung über TLS

INTERNET OF THINGS CONFERENCE

# Teil 3: REST API

# Dropwizard

# Technologien

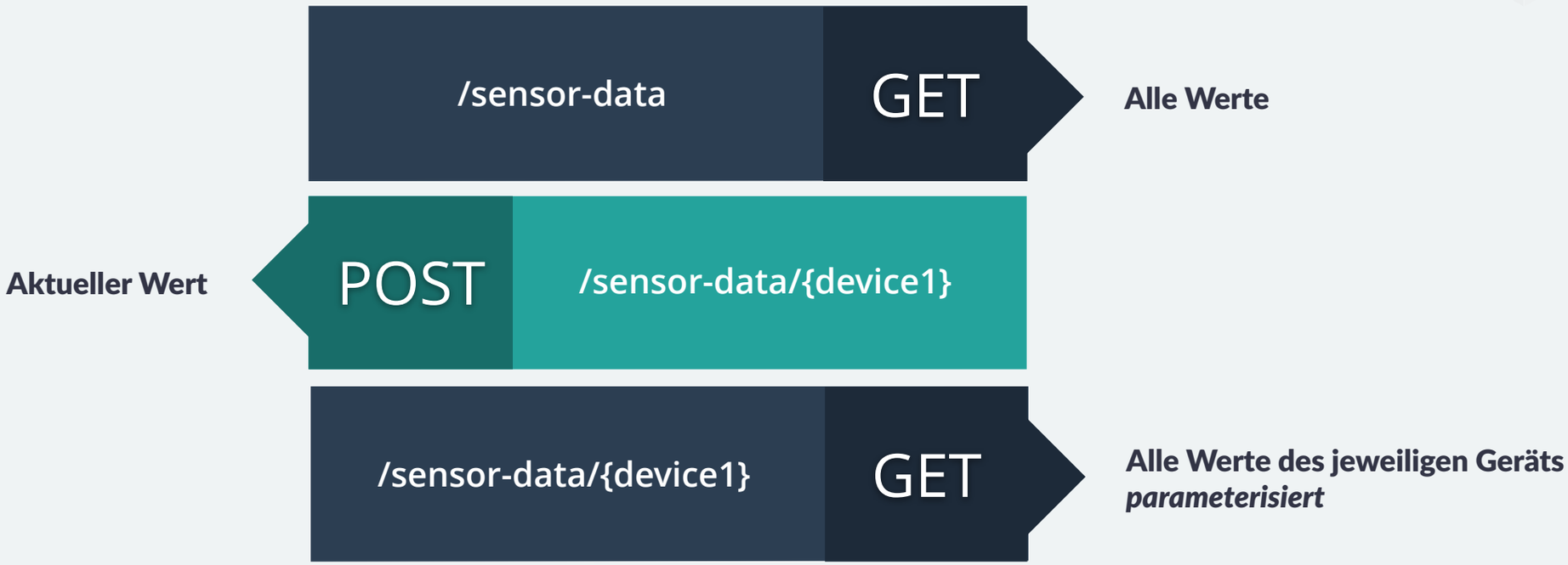
Dropwizard

Dropwizard
**REST API**

- **Historische Daten zur Verfügung stellen**
- eingehende POST Requests an HiveMQ weiterleiten

# REST Ressourcen

IoTCloud




**Format: JSON**

# Dropwizard

Kurzübersicht

"Produktiv einsatzbereite, hochskalierbare **REST API**"

Ausgerichtet auf den einfachen Betrieb

Modulbasiert mit vielen mitgelieferten Modulen

# Technologien

Dropwizard

jetty://

HTTP

Jersey

REST

Jackson

JSON

metrics

*powered by*

Apache maven

# Getting Started

Dropwizard

# Übung 6

Erstes Dropwizard Projekt

Version

0.5

STEPS

## Projekt öffnen

> ***rest-api* in Java IDE** öffnen

## Projekt ausführen

> Run Configuration
> **IoTCloudApplication server iotcloud.yml**

## HelloWorld

> HelloWorld Ressource im Webbrowser aufrufen

INTERNET OF THINGS CONFERENCE

# Übung 7

GET /sensor-data

Version

0.6

STEPS

## Projekt öffnen

> ***rest-api* in Java IDE** öffnen

## Datenbank

> Datenbank anlegen mit ***create.sql***
> Beispieldaten importieren ***exmple-data.sql***

## Vorbereiten

> SensorDataItem POJO anlegen
> Datenbankverbindung herstellen (JDBC)

## Ressource

> Daten aus Datenbank holen
> Zurückgeben einer Liste von SensorDataItems

# Übung 8

JDBI statt JDBC

Version

0.6

STEPS

## Projekt öffnen

- ***rest-api* in Java IDE** öffnen

## Konfiguration

- Datenbankparameter in iotcloud.yml

## Vorbereiten

- JDBI initialisieren
  Mapper

## Ressource

- Interface DAO mit SQL Query schreiben
  Zurückgeben einer Liste von SensorDataItems

# Verbesserungen

HiveMQ Plugin

- Caching einbauen
- Datenbankverbindung mit ConnectionPool
- ORM Framework einsetzen (JDBI, Hibernate)
- Authentifizierung, Autorisierung
- Mehrere Ressourcen
- Transportverschlüsselung mit TLS

INTERNET OF THINGS CONFERENCE

Teil4: WebUI

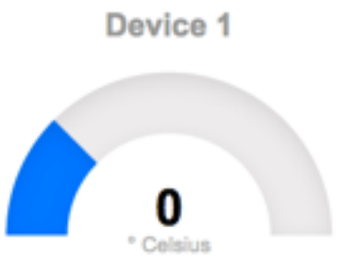
# Erster Test für die IoT Cloud

INTERNET OF THINGS CONFERENCE

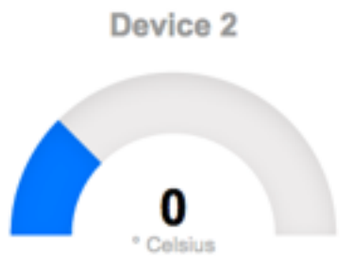
# WebUI

HiveMQ Plugin

## Temperature 1




## Temperature 2




**Live Daten über MQTT**

Show 10 entries

Search:

| Id | Timestamp | Device ID | Value |
|---|---|---|---|
| 1 | 31.8.2014 04:22 | device1 | 0 °C |
| 2 | 31.8.2014 04:22 | device2 | 0 °C |
| 3 | 31.8.2014 04:22 | device1 | 0 °C |
| 4 | 31.8.2014 04:22 | | 3 °C |
| 5 | 31.8.2014 04:22 | device1 | 3 °C |
| 6 | 31.8.2014 04:22 | device2 | 6 °C |
| 7 | 31.8.2014 04:22 | device1 | 4 °C |
| 8 | 31.8.2014 04:22 | device2 | 9 °C |
| 9 | 31.8.2014 04:22 | device1 | 4 °C |

**Historische Daten über REST**

INTERNET OF THINGS CONFERENCE

# Übung 9

WebUI Ausliefern

Version

0.7

STEPS

## Projekt öffnen

> ***rest-api* in Java IDE** öffnen

## Assetbundle

> ***Assetbundle*** konfigurieren

## webapp

> ***webapp*** in ***asset*** Ordner kopieren

## Dropwizard

> starten und ***localhost:8080/asset*** öffnen

INTERNET OF THINGS CONFERENCE

# Zusammenfassung

INTERNET OF THINGS CONFERENCE

# IoT Cloud Platform

Komponenten

**Live Daten von Sensoren**

HTTP, MQTT, CoAP, ...

**Data Storage**

Datenbank mit historische Daten

**Identity Management**

User, Devices

**IoT Cloud**

**Device Management**

Configure, Update Firmware, Monitor, Statistics

**Administration & Betrieb**

Web UI, Einfache Bedienung

**Security**

Transportsicherheit, Authentifizierung, Autorisierung

# Ausblick

**Skalierbarkeit, Verfügbarkeit**
HiveMQ und Dropwizard sind skalierbar

**Mehr Protokolle**
CoAP, XMPP, ...

**Device Management**
Überwachen der Geräte

**Security**
Authentifizierung, Autorisierung

# Welche Herausforderungen sind noch zu lösen?

INTERNET OF THINGS CONFERENCE

# Danke!

# IoT Con HiveMQ Special

http://www.hivemq.com/iot-con-special-2014/