

# CAS Digital Industry



Dez. Computing:  
iot, mobile, platforms, communication / cloud, *security*

Jürg Luthiger / Martin Gwerder  
University of Applied Sciences Northwestern Switzerland  
Institute for Mobile and Distributed Systems

# Zu meiner Person

- Studium Maschineningenieur an der ETH (1987)
- Nachdiplomstudium Mechatronik an der ETH (1989)
- Dissertation in der Robotik (1996)
  
- Entwicklungsingenieur bei Stäubli AG, Robotik (1992)
- Wissenschaftlicher Mitarbeiter & Gruppenleiter bei den Informatikdiensten der ETH (1998)
- Informatikdozent & Gruppenleiter an der FH Solothurn (2006)
- Institutsleiter, Institut für Mobile und Verteilte Systeme (ab 2006)

# Inhalte

## ■ Die Themen

- Vormittag (Jürg Luthiger)
  - IoT, Plattformen ... **Kommunikation**
- Nachmittag (Martin Gwerder)
  - Cloud ... Security

## ■ Vormittag 3 Teile

1. Szenarien IoT: Vorstellung und Problemstellung
2. Theorie IoT
  - Einführung
    - Begriffe, Rolle von IoT
  - Datenakquise
  - Datentransport
  - Datenanalyse
3. Szenarien IoT: Lösungskonzepte

# Übersicht

## ■ Inhalte

- Einführung: Lernziele
- Szenarien IoT: Vorstellung und Problemstellung
- Theorie IoT
- Szenarien IoT: Diskussion / Lösungsvorschläge

## ■ Aufbau der Veranstaltung

- Wissensvermittlung mit Folien
- Vertiefung mit Diskussionen

# Lernziele

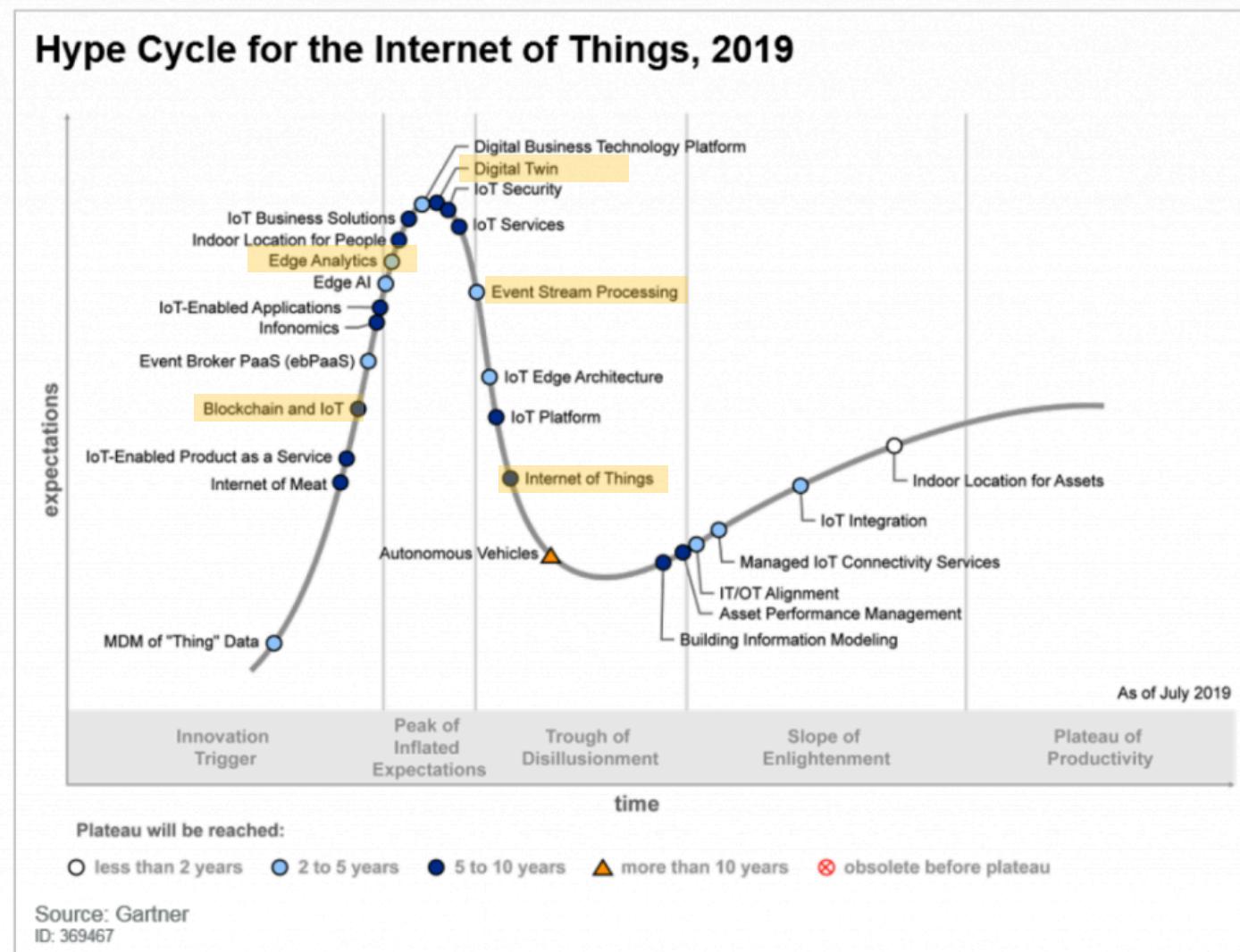
Die Studierenden

- kennen Mittel, um ein IoT Device über das **Internet** anzusprechen
- kennen verschiedene Varianten **IoT Systeme zu gestalten**, zu konzipieren und können Vor- und Nachteile nennen
- kennen die zentralen **Herausforderungen** einer IoT Anwendung

# Begriffe

- Internet of Things (IoT)
  - Computer, Sensoren und Aktoren, die über Internet-Protokolle miteinander kommunizieren können → Machine-to-Machine (M2M)-Kommunikation
- Cyber Physical Systems (CPS)
  - Systeme, welche die physikalische mit der virtuellen Welt verbinden z.B. mit Hilfe des Internets
  - Nutzt IoT
- Industrie 2025 (CH) / Industrie 4.0 (D) / Industrial Internet of Things IIoT (USA)
  - CPS in der industriellen Fertigung
  - Nutzt CPS (und IoT) als Enabler
  - Marketing

# Gartner Hype Cycle "Internet of Things" 2019



Source: Gartner, Hype Cycle for Internet of Things.

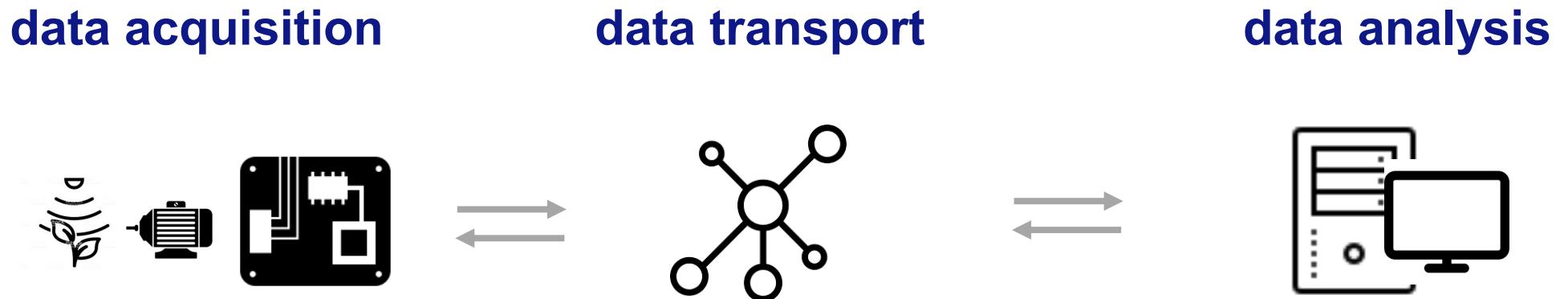
# Trends in IoT

## Top 20 Emerging IoT Trends That Will Shape Your Future Soon

see <https://www.ubuntupit.com/top-20-emerging-iot-trends-that-will-shape-your-future-soon/>

1. **Big Data Convergence**
2. **Data Processing with Edge Computing**
3. Greater Consumer Adoption
4. “Smart” Home Demand Will Rise
5. The Healthcare Industry Embraces IoT
6. **Auto-ML (Machine Learning) for Data Security**
7. IoT – Massive Growth Coming
8. **Blockchain for IoT Security**
9. **Better Data Analytics**
10. Smart Cities to Become Mainstream
11. Blockchain as Efficient Backend
12. Personalization of the Retail Experience
13. **Predictive Maintenance Boost up by IoT**
14. **Cloud Computing: The Future of IoT**
15. Software-as-a-Service Becomes the Norm
16. IoT Security Awareness and Training
17. Creation of Unified Framework for Integration
18. Energy and Resource Management
19. A Shift in Voice Control: From Mobile Platforms to Managing IoT Ecosystems
20. Stay Aware and Allied: Further Expansion of IoT

# IoT Szenarien: The Big Picture



# Szenarien IoT (Konkrete Beispiele aus dem Institut)

1. **LoRaWAN**: Monitoring eines Low Power Wireless Networks auf dem Campus Brugg-Windisch
2. **Fast Data**: Monitoring und Analyse (in "Echtzeit") von Indoor Sensordaten auf dem Campus Brugg-Windisch
3. **Production-as-a-Service**: IT Security in einem industriellen Umfeld
4. **Predictive Maintenance**: für die Instandhaltung von traditionellen Industrieanlagen
5. **Distributed Sensors**: Monitoring und Analyse von Sensoren im Feld

# Szenario: LoRaWAN

## ■ Ausgangslage

LoRaWAN steht für Long Range Wide Area Network. Es ist eine Kommunikationsinfrastruktur für kleine, drahtlose, batteriegetriebene Systeme und interessant, um für IoT Devices einen Kommunikationskanal über längere Distanzen bereitzustellen zu können. Die Hochschule für Technik (HT) betreibt auf dem Campus Brugg-Windisch ein solches LoRaWAN (u.a. auch Swisscom).

## ■ Ziel

Da das LoRaWAN vermehrt als Infrastruktur in der Lehre eingesetzt werden soll, braucht die HT ein Monitoring des Campus-LoRaWAN. Das Monitoring soll den Campus abdecken und über Kennwerte wie Signalstärke eines LoRa-Device die aktuelle Kommunikationsleistung des LoRaWAN an verschiedenen Standorten (Indoor vs. Outdoor) visualisieren zu können.

# Szenario: Fast Data

## ■ Ausgangslage

Mit Stream-Processing sollen Datenströme (Taktrate 1min) in Echtzeit analysiert werden, um auf entsprechende Ereignisse, wie bestimmter Threshold ist erreicht, sofort reagieren oder Daten aggregieren zu können, bevor diese an eine Datenbank für eine Big Data Analyse gespeichert werden.

## ■ Ziel

Verarbeitung und Monitoring von Sensordaten (Indoor, Feuchtigkeit, Temperatur), um Thresholds aus einem Datenstrom schnell erkennen, verarbeiten und anzeigen zu können.

# Szenario: Production as a Service on Site

## ■ Ausgangslage

Ein Schweizer Maschinenhersteller will seine Produktionsmaschinen temporär, jedoch kostenlos, bei seinen Kunden installieren und ausschliesslich über die konkret, produzierten Teile abrechnen. Dabei sammelt auf der Maschine eine Soft-SPS mit einem IPC alle relevanten Prozess- und Produktionsdaten. Da die Kunden über die ganze Welt verteilt sind, können sich der Schweizer Maschinenhersteller nicht auf eine funktionierende Kommunikationsinfrastruktur vor Ort verlassen.

## ■ Ziel

Dieses neue Geschäftsmodell verlangt eine Kommunikationsinfrastruktur zwischen den Kunden und Maschinenhersteller, so dass alle Prozess- und Produktionsdaten, da sie rechnungsrelevant sind, sicher und unverfälscht zum Maschinenhersteller übermittelt werden, so dass er die Daten über ein Monitoring visualisieren und auf Basis von validen Daten die Rechnungen für den Kunden auslösen kann.

# Szenario: Predictive Maintenance

## ■ Ausgangslage

Eine Schweizer Instandhaltungsfirma ist für den reibungslosen Betrieb von unterschiedlichen Maschinen verantwortlich, wie Seilbahnen (Antriebsmaschine) oder chemische Fabrikationslinien. Das Unternehmen will diese Maschinen und Anlagen smart machen, das heisst bestehende Anlagen mit Sensorik (wie Vibration, Temperatur, Dehnmessstreifen) ausrüsten, um aus den Daten Rückschlüsse auf den Zustand der jeweiligen Schlüsselkomponenten ziehen zu können. Die Anlagen sind mit einer Soft-SPS und IPC ausgerüstet. Betreute Anlagen können in eher abgelegenen Täler der Schweiz stehen.

## ■ Ziel

Dieses Geschäftsmodell verlangt eine automatisierte Analyse der Sensordaten und ein entsprechendes Monitoring, um daraus Rückschlüsse auf den Zustand der Anlage treffen zu können.

# Szenario: Distributed Sensors

## ■ Ausgangslage

Fehlerfrei funktionierende Infrastrukturkomponenten wie Hochspannungskabel sind für die moderne Gesellschaft eminent wichtig. Ausfälle können zu sehr hohen Kosten führen. Deshalb ist eine kontinuierliche Überwachung und eine regelmässige Wartung für die Betreiber solcher Infrastrukturanlagen zentral.

## ■ Ziel

Die Überwachung soll durch ein automatisiertes Monitoring ersetzt werden. Dazu werden im Feld verschiedene Sensoren (Temperatur, Feuchtigkeit, Vibration, ...) an Schlüsselpositionen verlegt, um aus den Sensordaten in einem entsprechenden Operation Center des Betreibers das Monitoring, inkl. Analyse und Visualisierung aufbauen zu können.

# IoT Theorie

## ■ Datenakquise

- Sensoren
- Microcontroller
- IPC (Industrial PC)

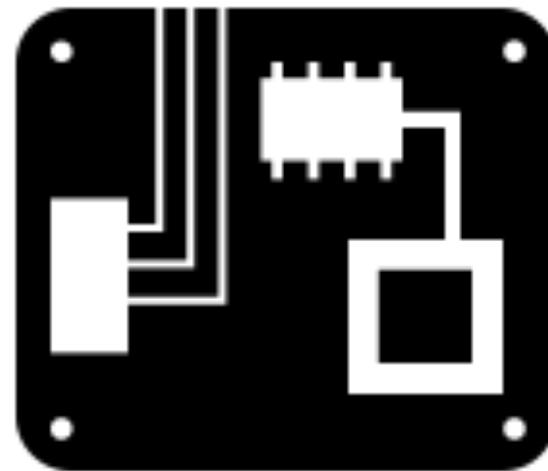
## ■ Datentransport

- Wireless Netzwerke
- Internet Protokolle

## ■ Datenanalyse

- Daten-Vermittlung
- Daten-Analyse
- Daten-Visualisierung

# Datenakquise: Hardware



# IoT Hardware

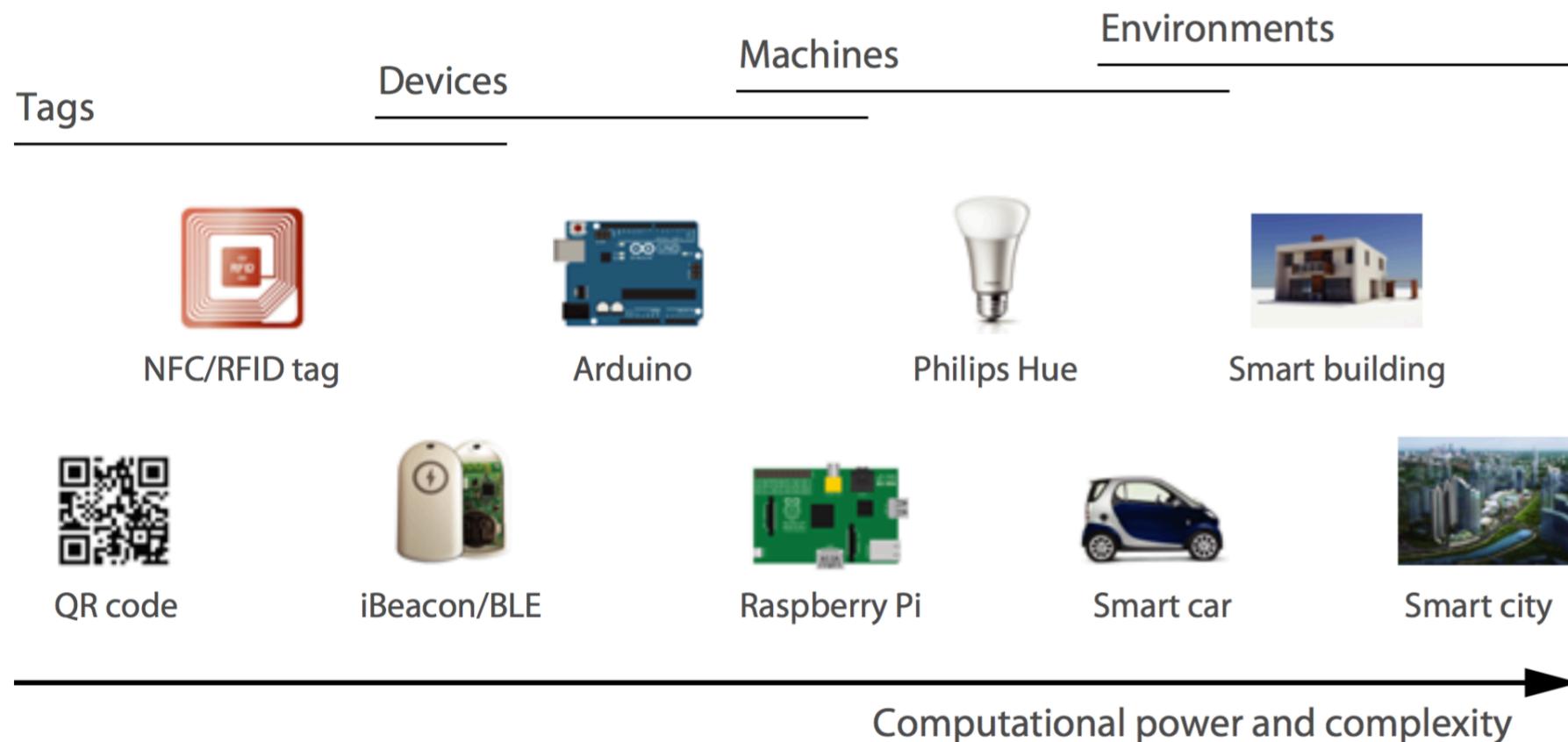
- Jedes Gerät
  - das auf die **physische Welt wirken kann**,
    - mit Sensoren misst,
    - mit Aktoren bewegt
  - und das über das **Internet kommunizieren** kann
  
- Beliebiger Formfaktor
  - mehrheitlich aber eingebettet,
  - deshalb als Embedded Computer mit Mikrocontroller
    - oft klein,
    - mit wenig Ressourcen (Speicher, Prozessor)
    - und ohne Bildschirm

# IoT und Digital Twins

- Digital Twins sind virtuelle Abbilder von physischen Objekten oder Systemen. Obligatorische Vorstufe des digitalen Zwillings ist das Internet der Dinge (IoT). Digital Twins weisen vier wesentliche Charakteristika auf:
  - Sensoren, die einen aktuellen Status ermitteln
  - Konnektivität, welche das Objekt vernetzt
  - Definierte Datenstrukturen, die geringstenfalls grundlegende Analytics-Funktionalitäten ermöglichen
  - Ein User Interface, das die relevanten Daten visualisiert

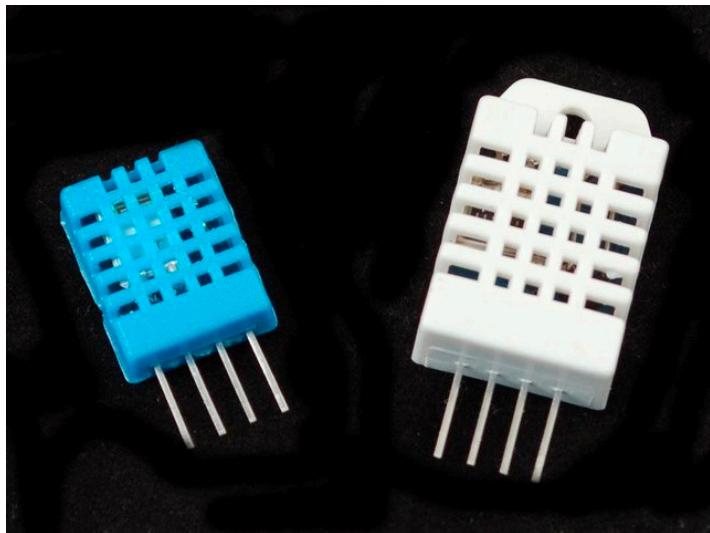
aus <https://www2.deloitte.com/de/de/pages/technology-media-and-telecommunications/articles/digital-twins.html>

# IoT Landkarte

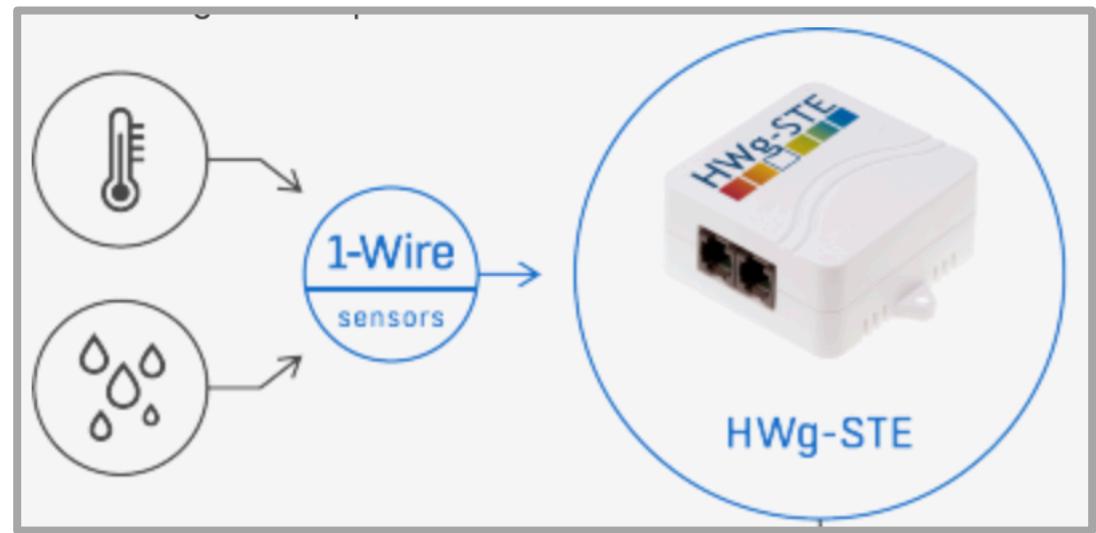


# IoT Hardware: Sensoren

- Sensoren bei Mouser
- Konkrete Beispiele:



DHT11 for temperature & humidity  
<https://learn.adafruit.com/dht/overview>



HWg-STE for temperature & humidity  
<https://www.hw-group.com/device/hwg-ste>

# IoT Hardware: Microcontroller

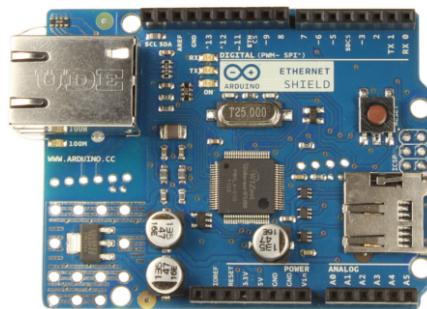


Raspberry Pi 3

CPU: 64bit ARM Cortex-A53 @ 1.2 GHz  
Flash: 1 GB  
Radio: WiFi + Bluetooth Low Energy

Arduino Uno

CPU: ATmega328 @ 16 MHz  
Flash: 32 KB  
Radio: ---



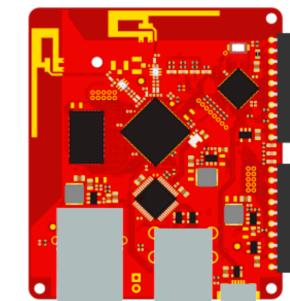
Intel Edison

CPU: Atom dual core @ 500 MHz  
Flash: 1 GB RAM + 4 GB Flash  
Radio: WiFi + Bluetooth



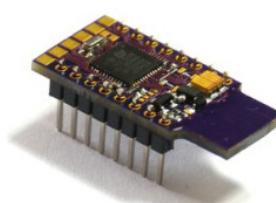
Tessel 2

CPU: Mediatek MT7620n @ 580 MHz  
Flash: 64 MB DDR2 RAM & 32 MB Flash  
Radio: WiFi



Espruino Pico

CPU: ARM Cortex M4 32-bit @ 84 MHz  
Flash: 384kb Flash, 96kb RAM  
Radio: ---



Particle Photon

CPU: ARM Cortex M3 @ 120 MHz  
Flash: 128KB RAM + 1MB Flash  
Radio: WiFi



# IoT Hardware: Evaluation Boards für ESP32

## ESP32

WIFI 2.4 GHz  
support 802.11 b/g/n  
support WPA/WPA2/...  
integrated Security  
Bluetooth  
...



# IoT Hardware: Industrie PC

## Beispiel Beckhoff

- Windows Betriebssysteme
  - Windows CE
  - Windows Embedded
  - Windows 10 IoT Enterprise
- TwinCAT als Soft-SPS
- unzähligen Funktionsblöcken für
  - Cloud Anbindung (MQTT, AMQP, OPC-UA, ...)
  - I/O Module
  - Machine Learning
  - ...



# Zusammenfassung IoT Hardware

## ■ Entwicklung

- Grosse Auswahl an Entwicklungsboards, kostengünstig
- Gute Unterstützung für Programmerstellung
- Unterstützt verschiedene Programmiersprachen
  - C, C++, JavaScript, Python, ...
- Hervorragend geeignet für das Prototyping (=> JavaScript)

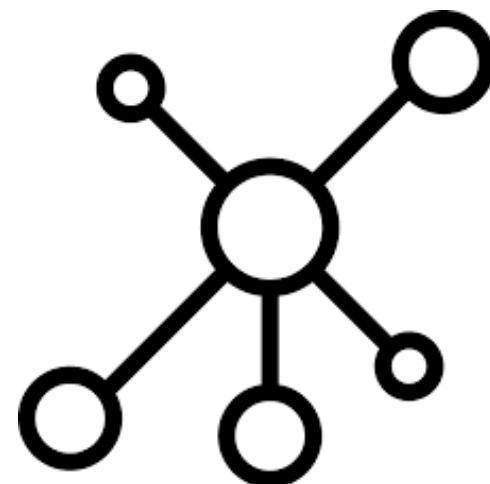
## ■ Produktion

- Grosse Auswahl
- Billig
- Kleine Formfaktoren, gut verbaubar
- Wifi-fähig mit integriertem TCP/IP Stack
- Low-level Programmierung möglich (C)

**=> Für Entscheidung ist eine gute Evaluation notwendig**

**=> Prototyp erstellen**

# Datentransport: Netzwerke



# "Internet" vs. "Internet of Things"

## ■ Internet

- Computer, die über Internet Protokolle miteinander kommunizieren
- Dokumente anzeigen und manipulieren

## ■ Internet of Things (IoT)

- Computer, Sensoren und Aktoren, die über Internet Protokolle miteinander kommunizieren
- Physikalische Größen messen, analysieren und manipulieren

# Übertragungsarten

## ■ LAN/WAN Verbindungsarten

- Drahtgebundene Verbindungen - Ethernet
- Drahtlose Verbindungen - Wi-Fi
- Mobilfunkverbindungen – LTE/3G/4G/...

## ■ Lokale Verbindungsarten (Gateway zum Internet notwendig)

- USB (drahtgebunden)
  - über Desktop Computer
- Near Field Communication (NFC) (drahtlos, bis 10 cm)
  - über 3G/Wi-Fi eines Smartphones
- Bluetooth/BLE (drahtlos, bis 10 m)
  - über 3G/Wi-Fi eines Smartphones
- ZigBee (drahtlos, 10 bis 100 m)
  - über ZigBee Gateway

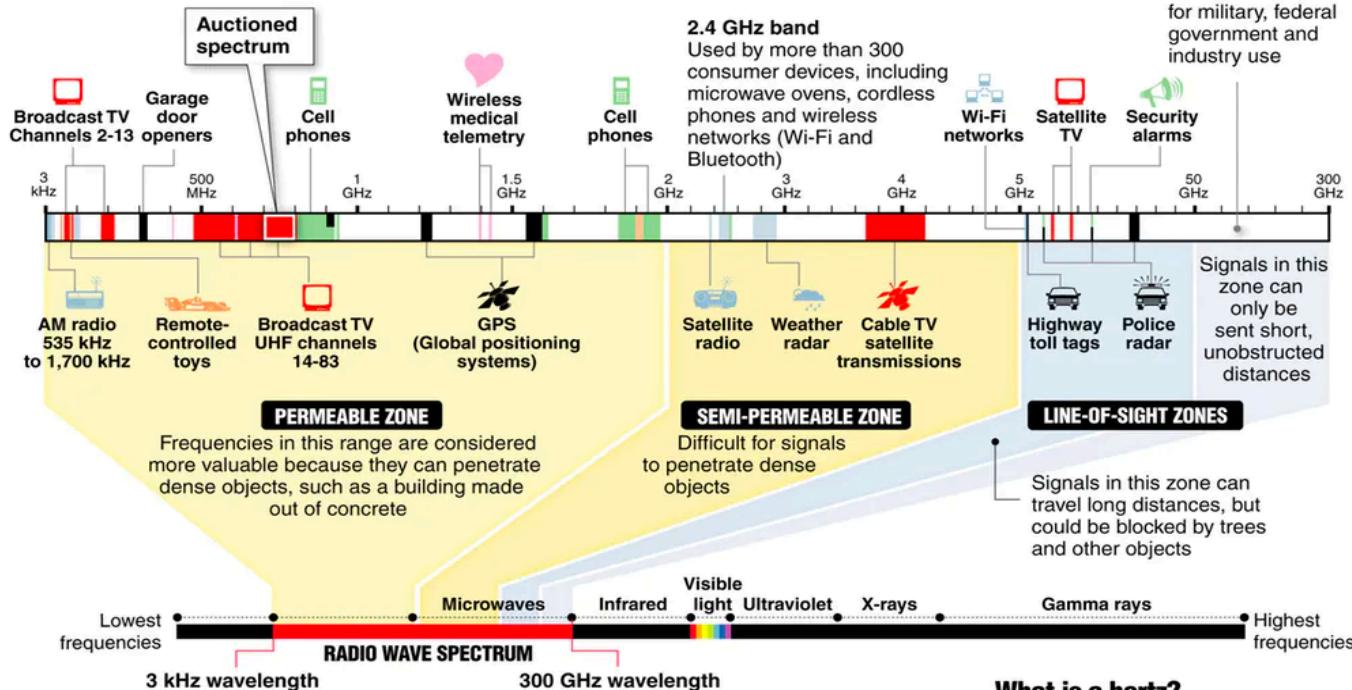
# Wireless Technology

- Bluetooth
- Wi-Fi
- Next Generation Mobile Networks: 3G - UMTS - LTS - 4G - 5G
- LoRaWAN

# Wireless Technologien: Frequenzen

## Inside the radio wave spectrum

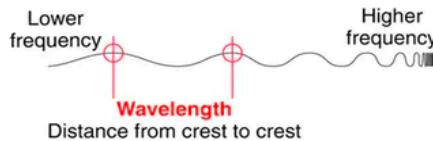
Almost every wireless technology – from cell phones to garage door openers – uses radio waves to communicate. Some services, such as TV and radio broadcasts, have exclusive use of their frequency within a geographic area. But many devices share frequencies, which can cause interference. Examples of radio waves used by everyday devices:



### The electromagnetic spectrum

Radio waves occupy part of the electromagnetic spectrum, a range of electric and magnetic waves of different lengths that travel at the speed of light; other parts of the spectrum include visible light and x-rays; the shortest wavelengths have the highest frequency, measured in hertz

Source: New America Foundation, MCT, Howstuffworks.com  
Graphic: Nathaniel Levine, Sacramento Bee



### What is a hertz?

One hertz is one cycle per second. For radio waves, a cycle is the distance from wave crest to crest

- 1 kilohertz (kHz) = 1,000 hertz
- 1 megahertz (MHz) = 1 million hertz
- 1 gigahertz (GHz) = 1 billion hertz

© 2008 MCT

# Bluetooth Low Energy (BLE)

- BLE ist eine Funktechnik, mit der sich Geräte in einer Umgebung von etwa **10 Metern** vernetzen lassen.
- BLE sendet im **2.4 GHz** Bereich.
- Im Vergleich zum „klassischen“ Bluetooth soll BLE deutlich **geringeren Stromverbrauch und geringere Kosten** mit einem ähnlichen Kommunikationsbereich haben.
  - Kleine BLE-Geräte können fast zwei Jahre lang mit einer einzelnen Knopfzelle URLs aussenden, falls nur im Sendemodus betrieben.

# Wi-Fi (WLAN)

- Wi-Fi ist eine Funktechnik, mit der sich Geräte in einer Umgebung von etwa **50 bis 100 Meter** vernetzen lassen.
- Netzwerk-Geschwindigkeiten

Netzwerktyp	≤ Brutto (maximal)	≤ Netto ein Stream <sup>1</sup>	≤ Netto mehrere Streams <sup>1</sup>
Fast Ethernet	100	94	94
WLAN 11n (Wifi 4)	600	100	200
WLAN 11ac (Wifi 5)	6900	400	800
WLAN 11ad (Wigig 1)	6700	k. A.	1400
WLAN 11ax (Wifi 6)	9600	500	1900
Gigabit-Ethernet	1000	940	940
NBase-T 2,5	2500	2400	2400
NBase-T 5	5000	4000	4700
10-Gigabit-Ethernet	10000	6000	9500

alle Werte in Megabit/s (MBit/s), WLAN-Datenrate grundsätzlich variabel, abhängig u. a. von Signalqualität, Kanalbreite, Anzahl Antennen, Nachbar-WLANS<sup>1</sup> typischer Durchsatz bei einem einzelnen / mehreren parallelen Downloads (und guter Verbindung im WLAN mit gutem Client)

aus <https://www.heise.de/select/ct/2019/2/1546773697424925>

# Mobilfunk

- 3G (UMTS, HSPA, HSPA+, LTE)
  - Beginn 2000er Jahre
  - Übertragungsrate bis 42 Mbit/s
  - Schwachpunkte vor allem in der Sicherheit
- 4G (LTE+)
  - Beginn 2010er Jahre
  - Übertragungsrate bis 1000 Mbit/s  
(Video-Streaming wird möglich)
  - Schwachpunkte in Sicherheit über Altlasten aus 3G
- 5G
  - Beginn 2020er Jahre
  - Übertragungsrate bis zu mehreren Gbit/s
  - Vor allem für die Industrie; Vernetzung von Maschinen (IoT)

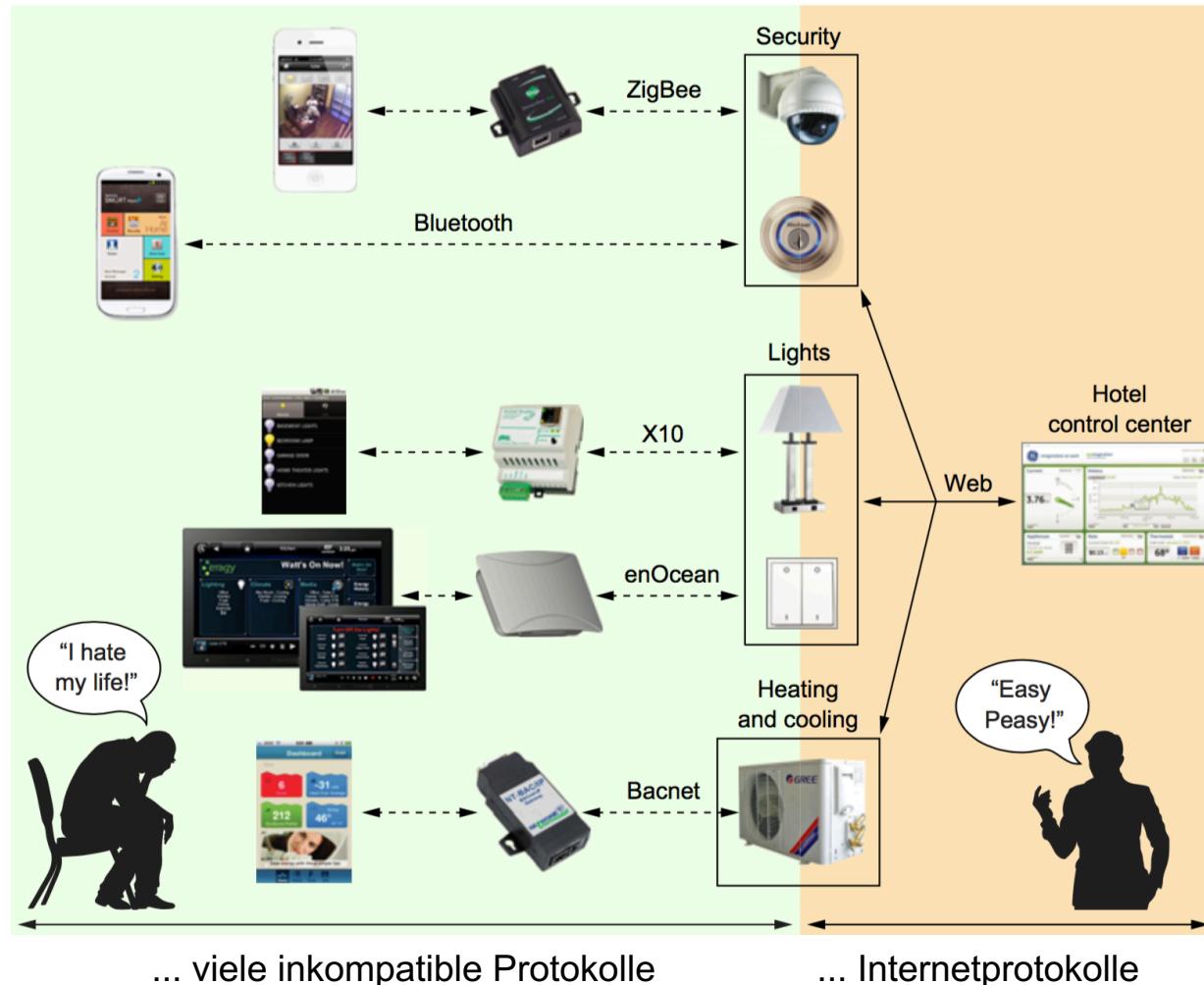
# LoRaWAN

- LoRa (Long Range)
    - Wireless Technologie, Übertragungsverfahren
    - proprietär; kontrolliert durch LoRa Alliance
    - für Kommunikation mit langer Reichweite und niedrigem Datendurchsatz
    - für M2M- und IoT-Anwendungen
  - LoRaWAN (Long Range Wide Area Network)
    - Low Power Wireless Network Protokoll basierend auf LoRa Technologie
    - Betreiber:
      - lokal: FHNW Brugg-Windisch (=> TTN)
      - national: Swisscom
      - international: The Things Network (TTN)
- LoRa-Device => LoRa Gateway => TTN Server => Applikation**

# Internet Protokolle: Auswahl

- HTTP
- WebSocket
- MQTT

# Protokoll Wirrwarr!



# Die Internetprotokollfamilie

OSI-Schicht	TCP/IP-Schicht	Beispiel
Anwendungen (7)	Anwendungen	HTTP, UDS, FTP, SMTP, POP, Telnet, OPC UA
Darstellung (6)		SOCKS
Sitzung (5)		
Transport (4)	Transport	TCP, UDP, SCTP
Vermittlung (3)	Internet	IP (IPv4, IPv6), ICMP (über IP)
Sicherung (2)	Netzzugang	Ethernet, Token Bus, Token Ring, FDDI, IPoAC
Bitübertragung (1)		

## ■ Anwendungsschicht

- Die Anwendungsschicht umfasst alle Protokolle, die mit Anwendungsprogrammen zusammenarbeiten.

## ■ Transportschicht

- Die Transportschicht ermöglicht eine Ende-zu-Ende-Kommunikation.

## ■ Internetschicht

- Die Internetschicht ist für die Weitervermittlung von Paketen und die Wegewahl (Routing) zuständig. Kern dieser Schicht ist das Internet Protocol (IP) in der Version 4 oder 6.

## ■ Netzzugangsschicht

- Die Netzzugangsschicht ist als Platzhalter für verschiedene Techniken zur Datenübertragung von Punkt zu Punkt zu verstehen z.B. durch Protokolle wie Ethernet, FDDI, PPP(Punkt-zu-Punkt-Verbindung) oder 802.11 (WLAN).

# HTTP Geschichte (1/2)

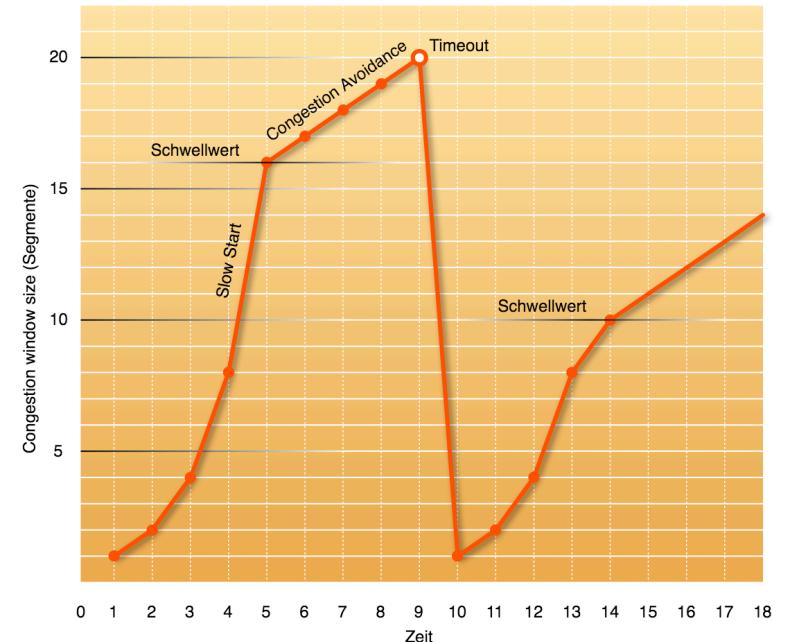
## ■ HTTP/0.9 (1989-1991)

- Roy Fielding, Tim Berners-Lee und andere entwickeln am CERN, das Hypertext Transfer Protocol, zusammen mit den Konzepten URL und HTML.
- Grundlagen des World Wide Web

## ■ HTTP/1.0 (1996)

- RFC 1945
- Jede Anfrage baut neue TCP-Verbindung auf, die nach der Antwort wieder geschlossen wird

→ **Effizienzprobleme** wegen  
TCP Slow Start Algorithmus

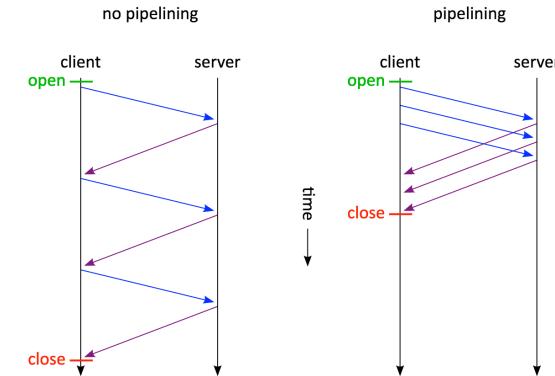


# HTTP Geschichte (2/2)

## ■ HTTP/1.1 (2014)

- RFC 2616
- Persistent Connection & HTTP Pipelining:  
Erlaubt mehrere Request/Response  
über die gleiche Verbindung, d.h.  
keine Verbindungsabbau.
- Caching:  
Zusätzliche Header Felder, um das Caching besser kontrollieren zu  
können.
- Method OPTIONS:  
Ein Client kann mit der HTTP Methode die Fähigkeiten des HTTP  
Servers abfragen
- ...

## ■ HTTP/2 (seit 2015) → RFC 7540



# HTTP Basics

- Stateless vs. Stateful
  - **HTTP ist „stateless“**
  - „stateful“ muss vom Server mit Session-Management unterstützt werden.  
Die entsprechende SessionId wird an der Client übertragen, dort z.B. in einem Cookie gespeichert und bei jedem neuen Request an der Server übertragen, so dass dieser die korrekte Session wieder laden kann.
- **URL** (Uniform Resource Locator)
  - Default Protokoll: HTTP (oder HTTPS)
  - Default Port: 80 (oder 8443)

http://www.domain.com:1234/path/to/resources?a=b&x=y



# HTTP Methods

## ■ GET

- Lesen (Read) einer Ressource
- Muss **idempotent** sein
- URL sollte auf 255 Bytes begrenzt werden

## ■ POST

- Erzeugen (Create) einer neuen Ressource
- Daten befinden sich nicht in der URL, sondern werden als Payload dem Request angehängt

## ■ PUT

- Ändern (Update) einer vorhanden Ressource
- Daten befinden sich nicht in der URL, sondern werden als Payload dem Request angehängt

## ■ DELETE

- Löschen (Remove) einer existierenden Ressource

## ■ ...

# HTTP Status Codes

- Der Server antwortet mit einer Response **und mit einem Status Code.**
- Status Codes sind wichtig und zeigen dem Client, wie er die Server-Response interpretieren soll.

## **1xx: Informational Messages**

### **2xx: Successful**

200 OK

...

### **3xx: Redirection**

303 See Other (Post/Redirect/Get Design Pattern)

304 Not Modified

...

### **4xx: Client Error**

404 Not Found

...

### **5xx: Server Error**

500 Internal Server Error

...

# URL, Methods, Status Codes

- Die fundamentalen Elemente einer HTTP Kommunikation



see: <http://code.tutsplus.com/tutorials/http-the-protocol-every-web-developer-must-know-part-1--net-31177>

# "Beyond HTTP": Motivation

## ■ Firewall

- Anforderung: Eine verteilte Anwendung muss im Internet über eine Standard-ICT-Infrastruktur kommunizieren können, inkl. den Firewalls.
- Restriktion Firewall: Der Aufbau einer Kommunikation ist nur von einem Client aus möglich.

## ■ "Echtzeit"

- Anforderung: Neue Daten auf dem Server in "Echtzeit" zum Client kommunizieren können.
- Restriktion HTTP: HTTP beschränkt die Kommunikation auf ein Request/Response Modell mit dem Client als Initiator des Kommunikationsaufbaus (siehe Firewall).

## ■ Bidirektionale Kommunikation

- Anforderung: Client und Server als gleichwertige Kommunikationspartner.
- Restriktion HTTP: Request/Response Modell von HTTP ist Half-Duplex.

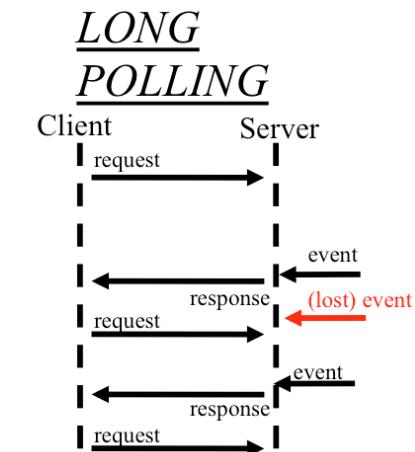
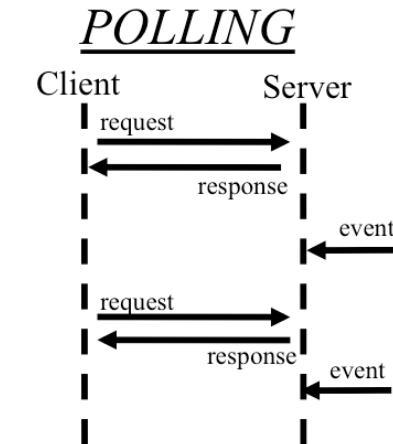
# Lösungsansätze (1/2)

## ■ Polling

- Beim Polling wird der Server in einem festgelegten Intervall gefragt, ob er neue Informationen hat.
- Nachteile
  - Grosser Overhead des Netzwerkes
  - Zusätzliche, unnötige Last auf dem Server

## ■ Long Polling

- Beim Long Polling wird eine separate HTTP-Verbindung zum Server erst geschlossen, wenn neue Daten verfügbar sind. Nachdem der Browser sie verarbeitet hat, sendet er einen neuen Request zum Server, um auf weitere Updates zu warten.
- Nachteile
  - Möglichkeit Events zu verpassen
  - Zusätzliche Last auf dem Server um die offenen HTTP Verbindungen zu verwalten



<http://www.heise.de/developer/artikel/WebSocket-Annaeherung-an-Echtzeit-im-Web-1260189.html>

# Lösungsansätze (2/2)

## ■ WebSocket

- erlaubt Full-Duplex Kommunikation über TCP-Verbindungen,
- wurde im 2011 als RFC 6455 durch IETF standardisiert.
- startet beim Client, der wie bei HTTP einen Request initiiert, mit dem Unterschied, dass nach der Übertragung der Daten zum Verbindungsaufbau die zugrundeliegende TCP-Verbindung bestehen bleibt.

WebSocket-Protokoll	
<b>Familie:</b>	Internetprotokollfamilie
<b>Einsatzgebiet:</b>	bidirektionale Verbindung zwischen einer Webanwendung und einem WebSocket-Server
<b>Port:</b>	80/TCP
WebSocket im TCP/IP-Protokollstapel:	
<b>Anwendung</b>	<b>WebSocket</b>
<i>Transport</i>	TCP
<i>Internet</i>	IP (IPv4, IPv6)
<i>Netzzugang</i>	Ethernet Token Bus Token Ring FDDI ...
<b>Standards:</b>	RFC 6455 (The WebSocket Protocol Version 13, 2011)

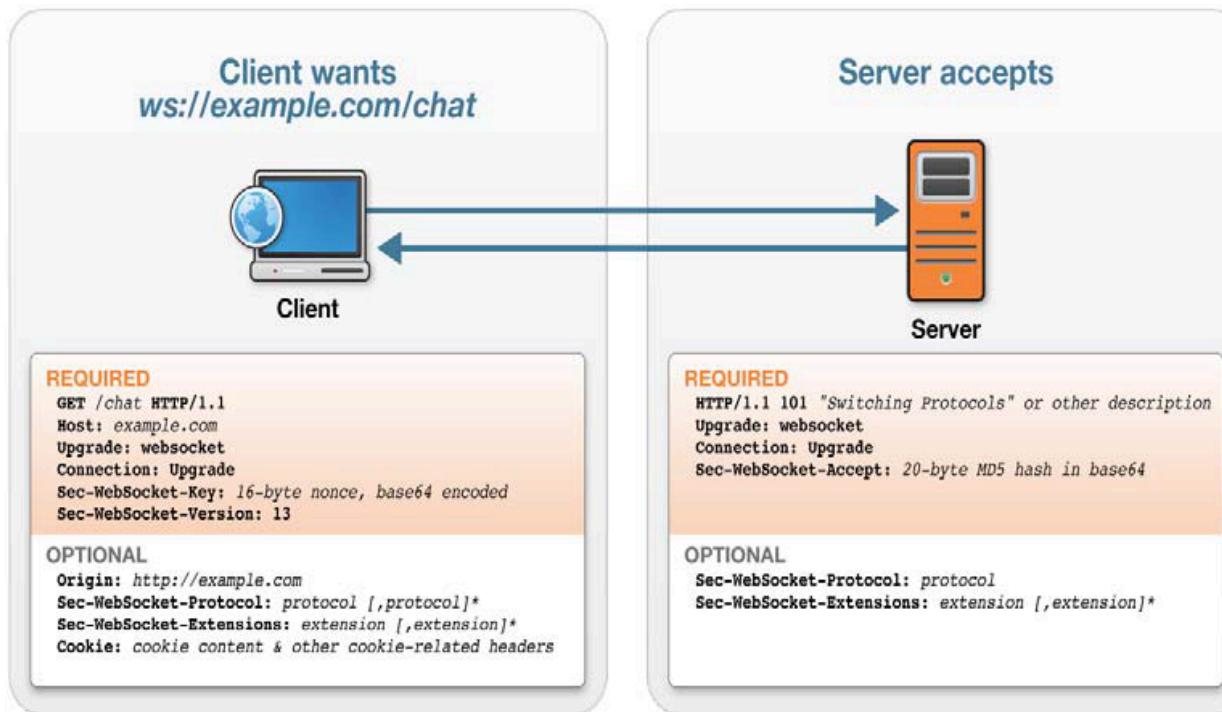
<https://de.wikipedia.org/wiki/WebSocket>

# WebSocket: Verbindungsauflaufbau (1/2)

- Einfache Integrierbarkeit in bestehende ICT-Infrastrukturen des Internets
  - WebSocket ist als **Upgrade des HTTP-Protokolls** konzipiert:  
=> Das Opening Handshake ist ein valider HTTP-Request
  - WebSocket baut nach dem Upgrade eine stehende TCP-Verbindung zwischen Client und Server auf.
- Die WebSocket-Spezifikation definiert zwei neue URI-Schemas
  - **ws**: für unverschlüsselte Verbindungen
  - **wss**: für verschlüsselte Verbindungen

# WebSocket: Verbindungsauflaufbau (2/2)

- Der WebSocket-Upgrade als Handshake zwischen Client und Server.
- Der Client startet den Upgrade Request.



<https://dzone.com/refcardz/html5-websocket>

# WebSocket: Datenübertragung

- WebSocket ist ein **Low-Level Protokoll** ("socket on the web")  
=> Ein **zusätzliches Applikationsprotokoll ist notwendig**, das sowohl vom Client wie auch vom Server interpretiert werden kann, um
  - z.B. CRUD-Operation auf Ressourcen auszuführen oder
  - Caching, Routing, Gzipping etc. zu implementieren

(Alles Funktionen, die von HTTP zur Verfügung gestellt werden.)

# WebSocket Eigenschaften

- bidirektional (full-duplex)
- internet-tauglich
- sicher; mittels TLS
- plattformunabhängig; Java, JavaScript, C++, ...

## ABER NICHT ...

- **dauerhaft**
  - WebSocket-Verbindungen werden in einem Internet-Kontext betrieben, d.h. Webserver, Firewall, Routers, ... Deshalb können WebSocket-Verbindungen durch solche Komponenten abgebaut werden, falls Inaktivität über eine gewisse Periode festgestellt wird.

# Keep Alive mit "Ping/Pong" Messages

## 5.5.2. Ping

The Ping frame contains an opcode of 0x9.

A Ping frame MAY include "Application data".

Upon receipt of a Ping frame, an endpoint MUST send a Pong frame in response, unless it already received a Close frame. It SHOULD respond with Pong frame as soon as is practical. Pong frames are discussed in [Section 5.5.3](#).

An endpoint MAY send a Ping frame any time after the connection is established and before the connection is closed.

NOTE: A Ping frame may serve either as a keepalive or as a means to verify that the remote endpoint is still responsive.

## 5.5.3. Pong

The Pong frame contains an opcode of 0xA.

[Section 5.5.2](#) details requirements that apply to both Ping and Pong frames.

A Pong frame sent in response to a Ping frame must have identical "Application data" as found in the message body of the Ping frame being replied to.

If an endpoint receives a Ping frame and has not yet sent Pong frame(s) in response to previous Ping frame(s), the endpoint MAY elect to send a Pong frame for only the most recently processed Ping frame.

aus <https://tools.ietf.org/html/rfc6455>

# "Beyond HTTP & Websocket": Motivation

## ■ **HTTP**

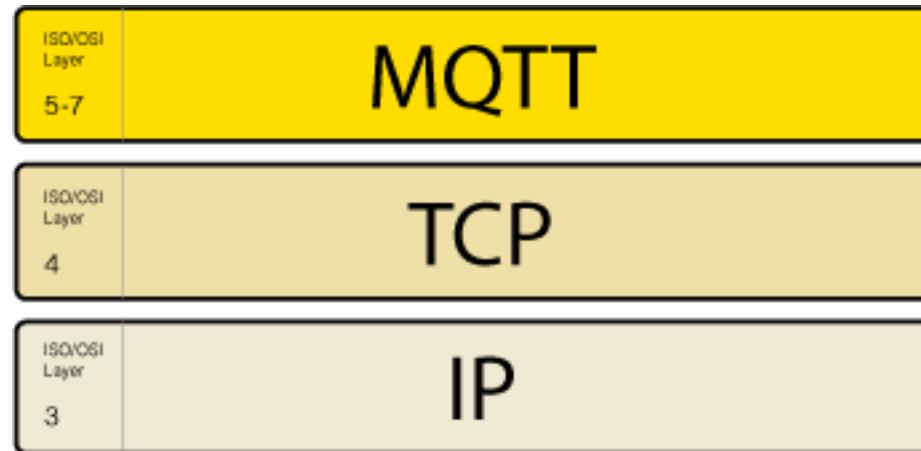
- Punkt-zu-Punkt Verbindung (one-to-one)
- Request-Response Modell => Unidirektional => Client/Server
- Schwergewichtig

## ■ **WebSocket**

- Punkt-zu-Punkt Verbindung (one-to-one)
- Bidirektional => gleichwertige Kommunikationspartner
- Kein Applikationsprotokoll spezifiziert

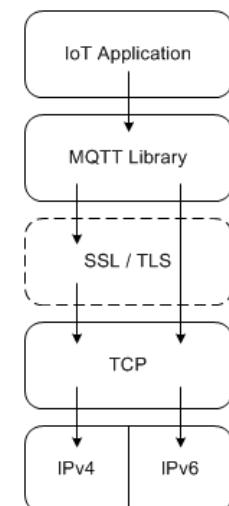
- ⇒ **soll leichtgewichtiges Protokoll sein**
- ⇒ **soll One-to-Many Verbindungen unterstützen**
- ⇒ **soll geringen Kommunikations-Overhead aufweisen**

# Alternative zu HTTP/Websocket: MQTT



# MQTT (Message Queue Telemetry Transport )

- MQTT ist als **Protokoll des Internet der Dinge** standardisiert.
- MQTT ist ein **asynchrones Protokoll**.
- MQTT implementiert das **Publish/Subscribe-Pattern**.
  - Der Paradigmenwechsel von einer Request/Response- zu einer ereignisgesteuerten Publish/Subscribe-Architektur ist hierbei der zentrale Aspekt. Sie ersetzt die Punkt-zu-Punkt-Verbindungen durch einen zentralen Server (Broker), zu dem sich Datenproduzenten und -nutzer gleichermaßen verbinden können.
  - Das Senden (publish) und Empfangen (subscribe) von Nachrichten funktioniert über sogenannte **Topics**.
- MQTT's Eigenschaften sind:
  - ist für unzuverlässige Netze mit geringer Bandbreite und hoher Latenzzeit entwickelt
  - ist robust bei Datenverlusten und schlechten Verbindungen
  - hat geringen Kommunikationsoverhead
  - kann mit TLS verschlüsselt werden



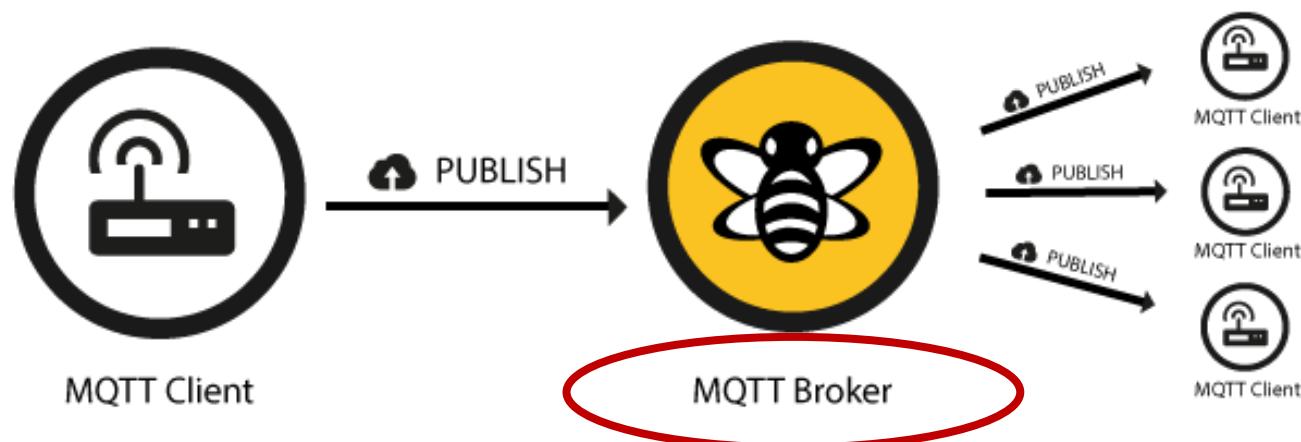
# MQTT Komponenten

## ■ MQTT Broker

- Zentrale Komponente => Kommunikationsserver
- Unterschiedliche Implementationen verfügbar
  - Referenzimplementation Eclipse Mosquitto => <https://mosquitto.org/>

## ■ MQTT Client

- Einfach
- Unterschiedliche Implementationen als Bibliotheken verfügbar



# MQTT QoS zur Qualitätskontrolle

- Ein weiteres wichtiges Konzept sind die **drei Servicequalitäten bei der Datenübertragung 0, 1 und 2.**
  - MQTT basiert auf TCP/IP, deshalb ist eine grosse Zuverlässigkeit bei der Übertragung bereits gegeben.
  - Bei mobilen Netzen mit vielen Übertragungsfehlern durch Verbindungsprobleme ist dies aber nicht ausreichend.
  - Daher hat das MQTT Protokoll Mechanismen eingebaut, die das erfolgreiche Übertragen von Nachrichten garantieren: Die Zusicherung variiert von **keiner Garantie (Level 0)** über die, dass die Nachricht **mindestens einmal ankommt (Level 1)**, bis hin zur Garantie, dass die **Nachricht genau einmal ankommt (Level 2)**. (Der Unterschied zwischen Level 1 und 2 liegt darin, dass es bei Level 1 passieren kann, dass eine Nachricht öfter einen Client erreicht.)
  - Je nach Anwendungsfall sollte der passende Level gewählt werden, denn je höher der Level, desto höher ist die benötigte Bandbreite.

# Die 5 wichtigsten Features von MQTT

- MQTT "publish/subscribe" Architektur
- Ideal für Constrained Netzwerke
  - "low bandwidth, high latency, data limits, and fragile connections"
- Quality of Service (QoS)
  - "The higher the QoS, the lower the performance"
- MQTT Clients können bei einem Verbindungsunterbruch benachrichtigt werden
  - "Last will and testament"
- MQTT Clients sind einfach zu implementieren

# MQTT und LoRa (TTN)

- MQTT und LoRa sind Protokolle für IoT
- LoRa ist proprietär bis zum LoRa Gateway
- Zwischen LoRa Gateway und LoRa Server (TTN) können folgende Protokolle zum Einsatz kommen:
  - UDP
  - MQTT (wird beim LoRaWAN der FHNW genutzt)
  - gRPC (binäres Protokoll von Google)
- Zwischen LoRa Server und Applikation wird MQTT genutzt

# Plattformen für die Daten-Vermittlung

- Verteilung LoRa-Messages über TTN.
  - The LoRaWAN Network Server <https://www.thethingsnetwork.org/>
- Verteilung MQTT über **MQTT-Broker:**
  - **VerneMQ** <https://vernemq.com/>  
VerneMQ is open source software, extendable, and enterprise support is available.
  - **Flespi** <https://flespi.com/>  
Cloud Platform with integrated MQTT Broker
  - **Eclipse Mosquitto** <https://mosquitto.org/>  
Eclipse Mosquitto is an open source (EPL/EDL licensed) message broker that implements the MQTT protocol versions 5.0, 3.1.1 and 3.1. Mosquitto is lightweight and is suitable for use on all devices from low power single board computers to full servers.

# Datenanalyse: Plattformen



# IoT Platforms: Daten-Analyse

- Verarbeitung von Daten durch Streaming Plattformen, wie:
  - **Apache Kafka** <https://kafka.apache.org/>  
Open-Source-Software-Projekt der Apache Software Foundation, das insbesondere der Verarbeitung von Datenströmen dient.
  - **Apache Flink** <https://flink.apache.org/>  
Open Source Stream Processing Framework entwickelt von der Apache Software Foundation.
  
- Speicherung von Time Series, wie
  - **InfluxDB** <https://www.influxdata.com/products/influxdb-overview/>  
InfluxDB ist ein Open Source Datenbankmanagementsystem (DBMS), speziell für Zeitreihen (TimeSeries).
  - **Prometheus** <https://prometheus.io/>  
Kostenlose Softwareanwendung zur Ereignisüberwachung und -warnung

# IoT Platforms: Daten-Visualisierung

## ■ Verschiedene Anbieter, wie:

- Grafana** <https://grafana.com/>

"Grafana is the open source analytics & monitoring solution for every database"

- Kibana** <https://www.elastic.co/de/products/kibana>

Browserbasierte Open-Source-Analyseplattform, die auf der Suchmaschine Elasticsearch aufbaut.

## ■ Google Trend

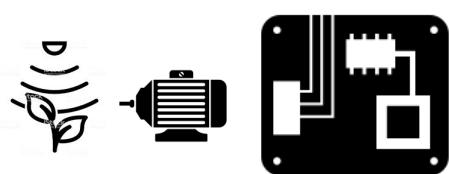
- <https://trends.google.de/trends/explore?q=grafana,kibana>

# IoT Platform: Komplettsysteme

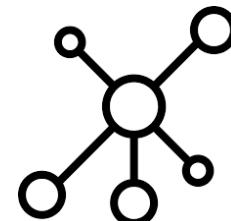
- Verschiedene Anbieter, wie:
  - **Thingsboard** <https://thingsboard.io>  
Open-Source IoT Platform
  - **Adafruit** <https://io.adafruit.com>  
"The internet of things for everyone"
  - **Azure IoT** <https://azure.microsoft.com/de-de/overview/iot/>  
Services für IoT: IoT Hub, Digital Twins, IoT Central Application, ...
  - **AWS IoT** [https://aws.amazon.com/de/iot/?nc2=h\\_m2](https://aws.amazon.com/de/iot/?nc2=h_m2)  
IoT-Services für industrielle, private und kommerzielle Lösungen
  - **Google Cloud IoT** <https://cloud.google.com/solutions/iot/?hl=de>  
Plattform für intelligente IoT-Dienste
- Liste von Plattformen mit einem Funktionsvergleich:  
<https://www.postscapes.com/internet-of-things-platforms/>

# IoT Szenarien: Diskussion

Datenakquise



Datentransport



Datenanalyse



# Szenario: LoRaWAN

## ■ Ausgangslage

LoRaWAN steht für Long Range Wide Area Network. Es ist eine Kommunikationsinfrastruktur für **kleine, drahtlose, batteriegetriebene Systeme** und interessant, um für IoT Devices einen Kommunikationskanal über längere Distanzen bereitstellen zu können. Die Hochschule für Technik (HT) betreibt auf dem Campus Brugg-Windisch ein solches LoRaWAN (u.a. auch Swisscom).

## ■ Ziel

Da das **LoRaWAN** vermehrt als Infrastruktur in der Lehre eingesetzt werden soll, braucht die HT ein Monitoring des Campus-LoRaWAN. Das Monitoring soll den Campus abdecken und über Kennwerte wie **Signalstärke** eines LoRa-Device die aktuelle Kommunikationsleistung des LoRaWAN an verschiedenen Standorten (Indoor vs. Outdoor) **visualisieren** zu können.

# Szenario: Fast Data

## ■ Ausgangslage

Mit **Stream-Processing** sollen Datenströme (Taktrate 1min) in Echtzeit analysiert werden, um auf entsprechende Ereignisse, wie bestimmter Threshold ist erreicht, **sofort reagieren oder Daten aggregieren** zu können, bevor diese an eine **Datenbank** für eine Big Data Analyse gespeichert werden.

## ■ Ziel

**Verarbeitung** und **Monitoring** von Sensordaten (Indoor, Feuchtigkeit, Temperatur), um Thresholds aus einem Datenstrom **schnell erkennen, verarbeiten und anzeigen** zu können.

# Szenario: Production as a Service on Site

## ■ Ausgangslage

Ein Schweizer Maschinenhersteller will seine Produktionsmaschinen temporär, jedoch kostenlos, bei seinen Kunden installieren und ausschliesslich über die konkret, produzierten Teile abrechnen. Dabei sammelt auf der Maschine eine **Soft-SPS mit einem IPC alle relevanten Prozess- und Produktionsdaten**. Da die Kunden über die ganze Welt verteilt sind, können sich der Schweizer Maschinenhersteller nicht auf eine funktionierende Kommunikationsinfrastruktur vor Ort verlassen.

## ■ Ziel

Dieses neue Geschäftsmodell verlangt eine **Kommunikationsinfrastruktur** zwischen den Kunden und Maschinenhersteller, so dass alle **Prozess- und Produktionsdaten**, da sie rechnungsrelevant sind, **sicher und unverfälscht** zum Maschinenhersteller übermittelt werden, so dass er die Daten über ein **Monitoring** visualisieren und auf Basis von validen Daten die Rechnungen für den Kunden auslösen kann.

# Szenario: Predictive Maintenance

## ■ Ausgangslage

Eine Schweizer Instandhaltungsfirma ist für den reibungslosen Betrieb von unterschiedlichen Maschinen verantwortlich, wie Seilbahnen (Antriebsmaschine) oder chemische Fabrikationslinien. Das Unternehmen will diese Maschinen und Anlagen smart machen, das heisst bestehende Anlagen mit Sensorik (wie **Vibration, Temperatur, Dehnmessstreifen**) ausrüsten, um aus den Daten Rückschlüsse auf den Zustand der jeweiligen Schlüsselkomponenten ziehen zu können. Die Anlagen sind mit einer **Soft-SPS und IPC** ausgerüstet. Betreute Anlagen können in eher abgelegenen Täler der Schweiz stehen.

## ■ Ziel

Dieses Geschäftsmodell verlangt eine **automatisierte Analyse** der Sensordaten und ein entsprechendes **Monitoring**, um daraus Rückschlüsse auf den Zustand der Anlage treffen zu können.

# Szenario: Distributed Sensors

## ■ Ausgangslage

Fehlerfrei funktionierende Infrastrukturkomponenten wie Hochspannungskabel sind für die moderne Gesellschaft eminent wichtig. Ausfälle können zu sehr hohen Kosten führen. Deshalb ist eine **kontinuierliche Überwachung** und eine regelmässige Wartung für die Betreiber solcher Infrastrukturanlagen zentral.

## ■ Ziel

Die Überwachung soll durch ein **automatisiertes Monitoring** ersetzt werden. Dazu werden im Feld **verschiedene Sensoren** (Temperatur, Feuchtigkeit, Vibration, ...) an Schlüsselpositionen verlegt, um aus den Sensordaten in einem entsprechenden Operation Center des Betreibers das **Monitoring, inkl. Analyse und Visualisierung** aufbauen zu können.

# Kooperationsmöglichkeiten mit der FHNW

## ■ Projektarbeiten

- aF&E Projekte**  
z.B. Innosuisse Finanzierung, Hasler Stiftung,  
Aargauer Forschungsfond (nur Firma aus dem Kanton Aargau!)
- Beratung und Dienstleistung
- Studierendenprojekte**
  - P1-4: ca. 2000 Personenstunden (5-8 Studierende)
  - P5: 180 – 360 Personenstunden (1-2 Studierende)
  - P6: 360 – 720 Personenstunden (1-2 Studierende)

## ■ Gast-Vorträge, Kurse

## ■ Kontaktmöglichkeiten mit Studierende

- Career-Day Technik, jeweils im Mai

# Vielen Dank



<http://www.fhnw.ch/technik/imvs>  
[juerg.luthiger@fhnw.ch](mailto:juerg.luthiger@fhnw.ch)