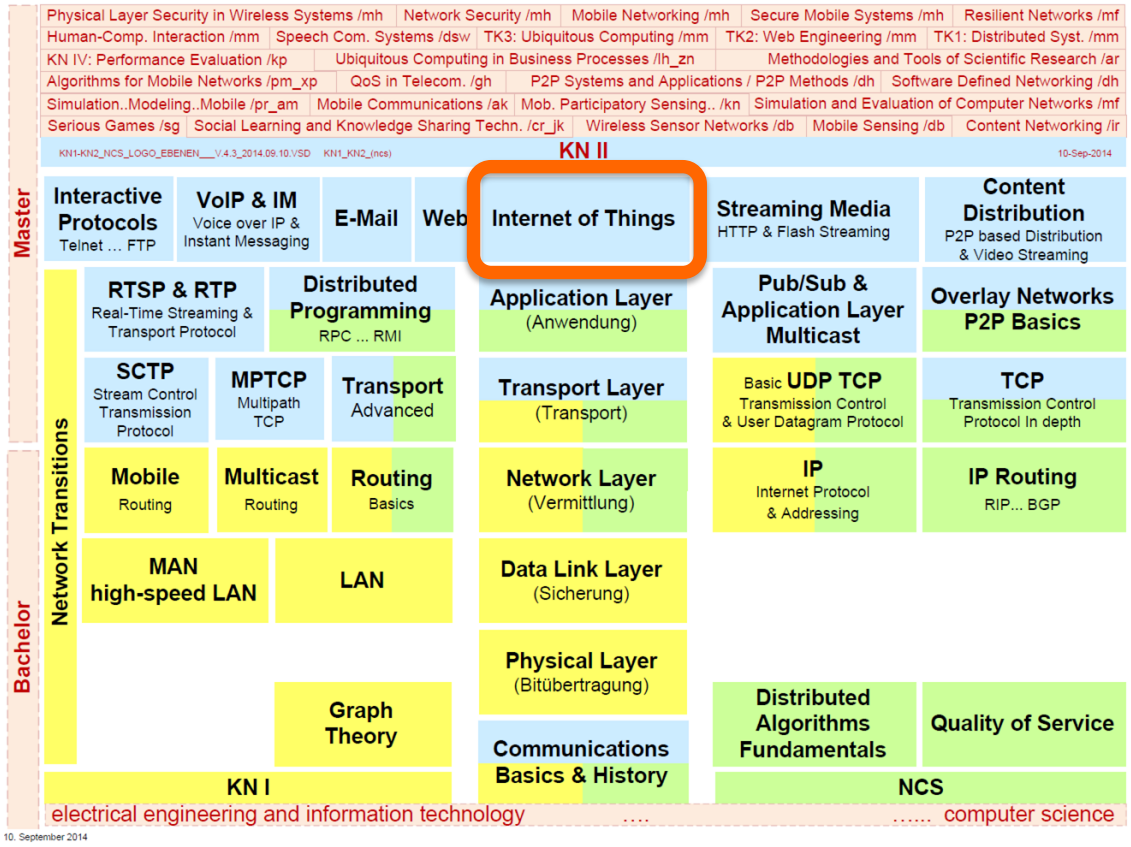
Communication Networks II

Internet of Things and Machine to Machine Communication



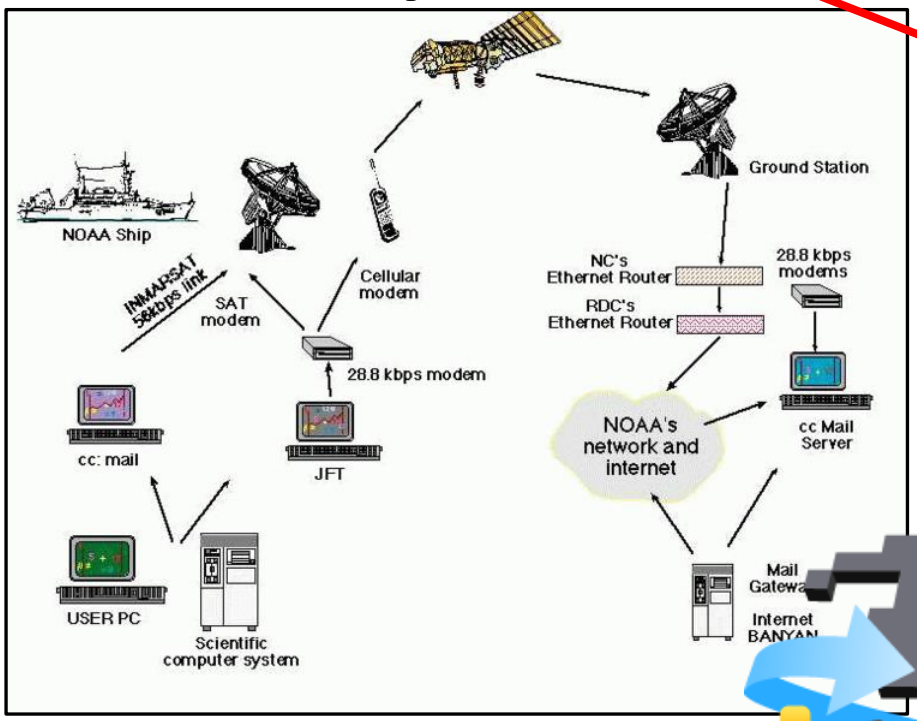
Slides partially adopted from Rune Hylsberg Jacobsen

Prof. Dr.-Ing. **Ralf Steinmetz**
KOM - Multimedia Communications Lab

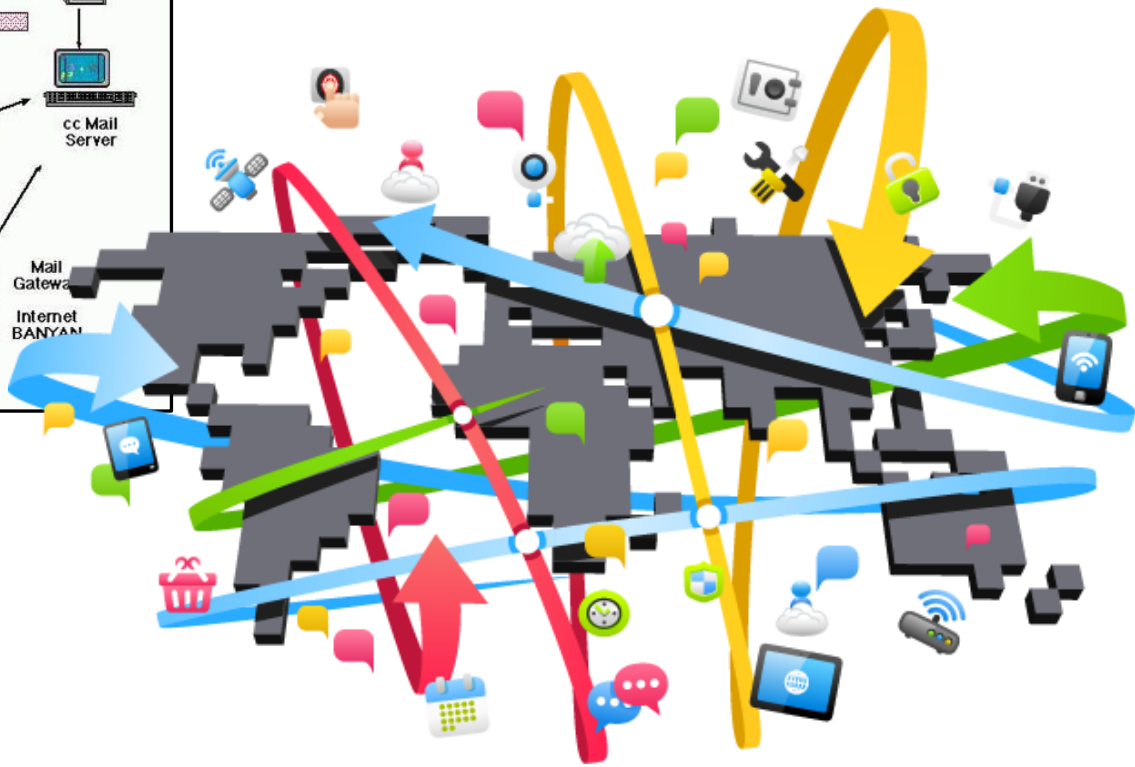
Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Internet in year 2000

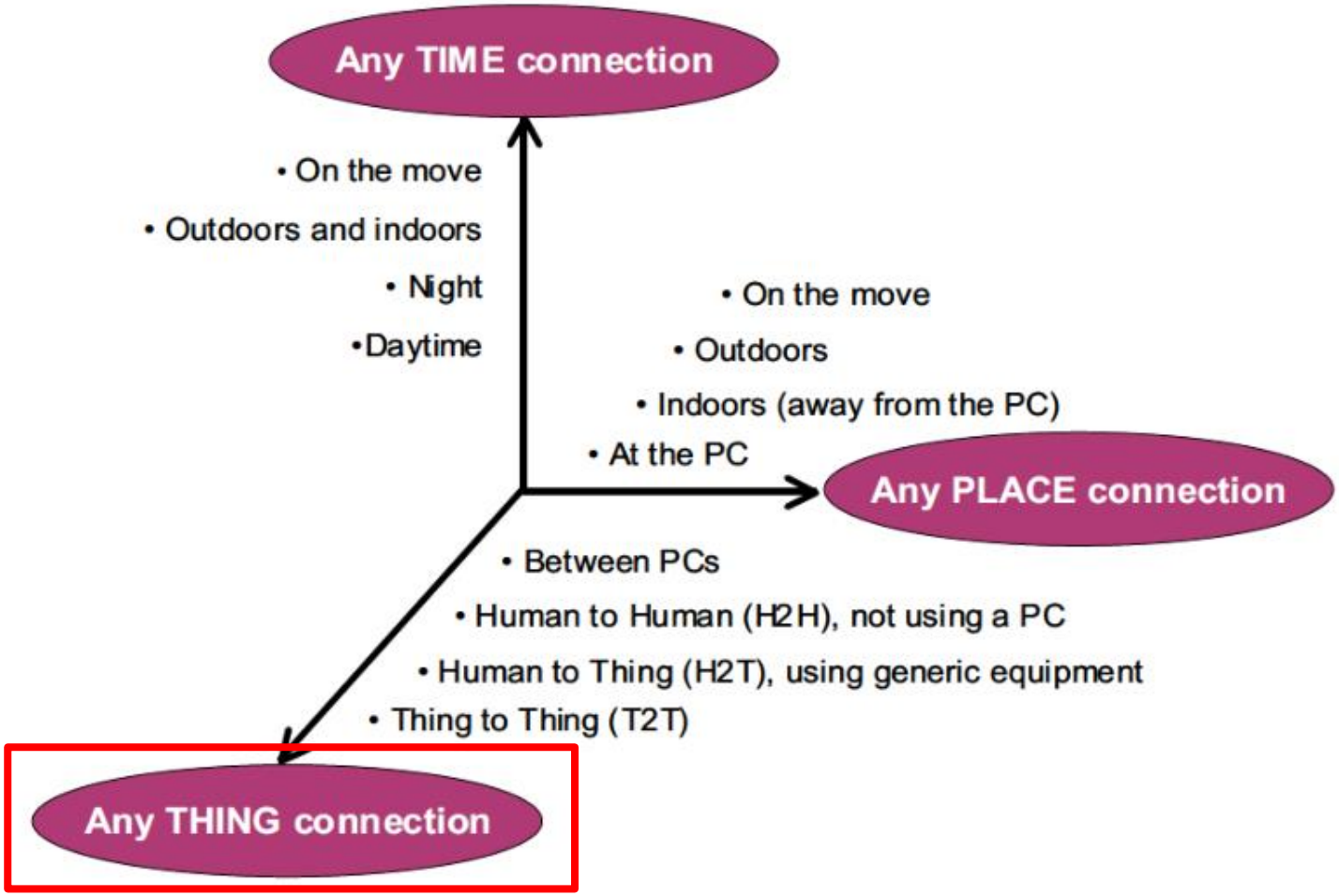


Internet today



Source: Wikipedia / Simon Staffans

Anything will be connected



Source: ITU adapted from Nomura Research Institute

Definitions - IoT

TECHNISCHE UNIVERSITÄT DARMSTADT

„A global infrastructure for the Information Society, enabling advanced services by **interconnecting** (physical and virtual) **things** based on, existing and evolving, interoperable information and **communication** technologies.”

ITU-T Y.2060, 2012

International Telecommunication Union

Standard Blueprint

Definitions (2)

TECHNISCHE UNIVERSITÄT DARMSTADT

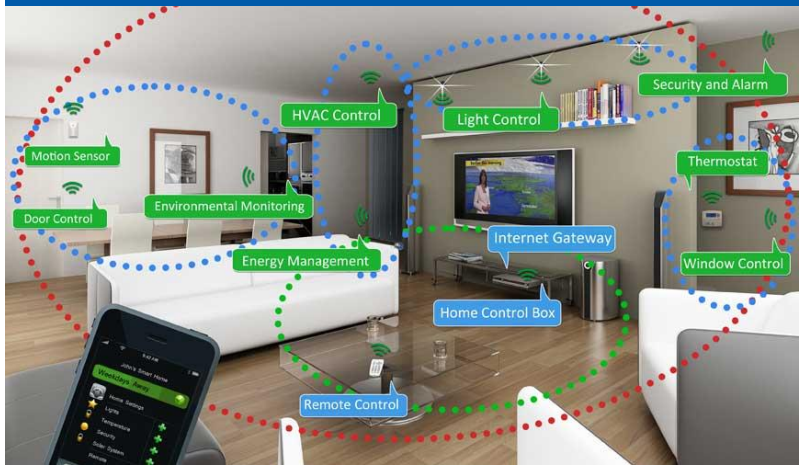
„A world where physical objects are **seamlessly integrated** into **the information network**, and where the physical objects can become active participants in business processes. Services are available to **interact** with these 'smart objects' **over** the **Internet**, query and change their state and any information associated with them…"

SAP Research, 2011

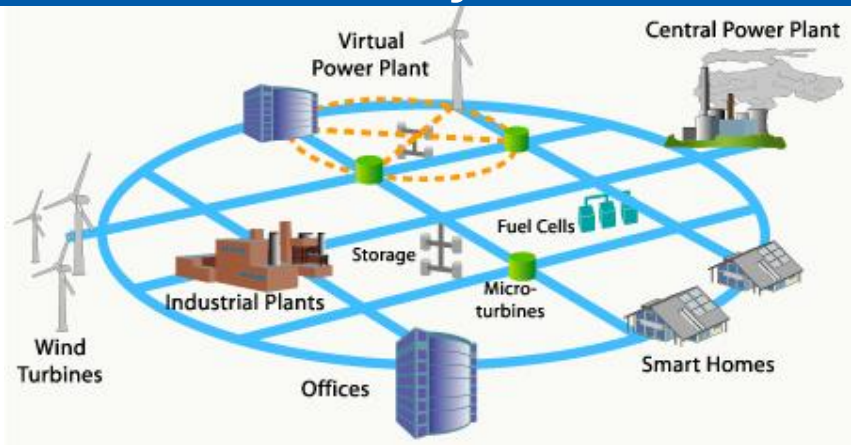
Fields of Application

TECHNISCHE UNIVERSITÄT DARMSTADT

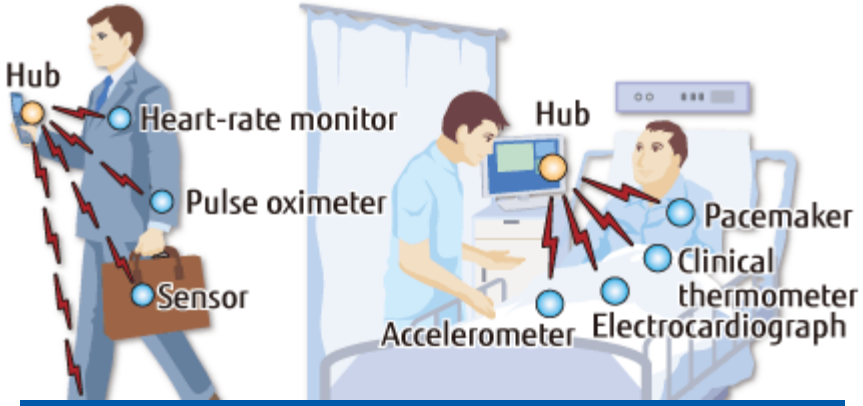
Home Automation



Electricity Grid



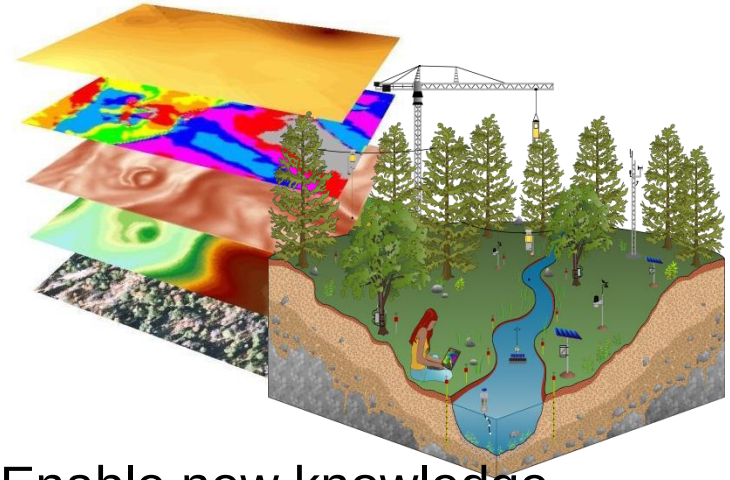
Transportation & Logistics



Health Care

… And many more

Benefits



Enable new knowledge

Save resources

Improve productivity

Increase comfort

Enhance safety & security

Prevent failures

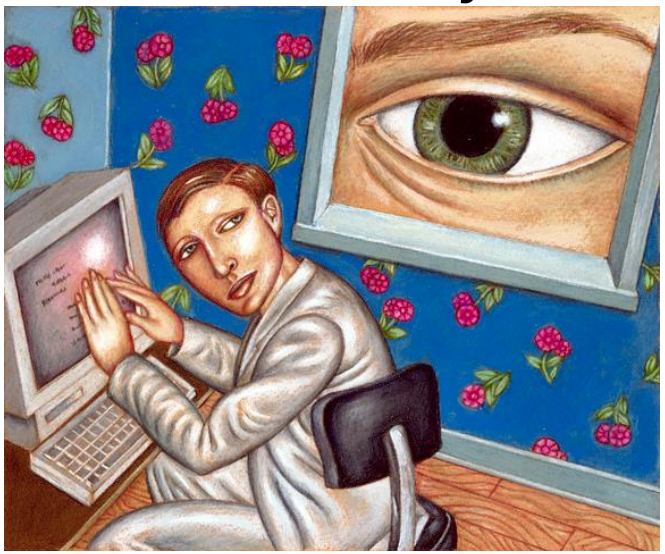
Save transport costs

Improve food & H20

Protect health

Downsides

Privacy?

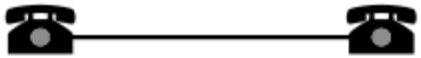


Security?

Interoperability?

Robustness?

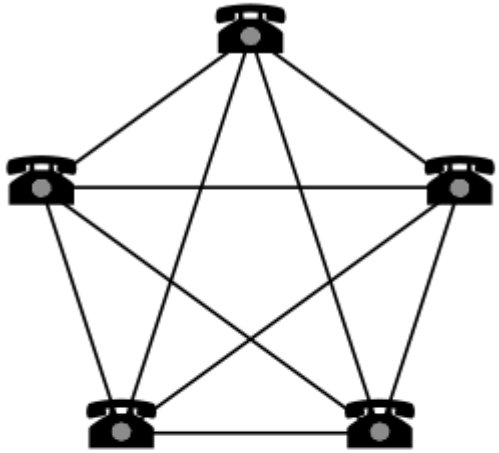
Metcalfe's Law



„Value (or power) of a network grows exponentially as a function of the number of network members.

As network members increase, more people want to use it (demand for network access increases).“

Metcalfe, ~1980



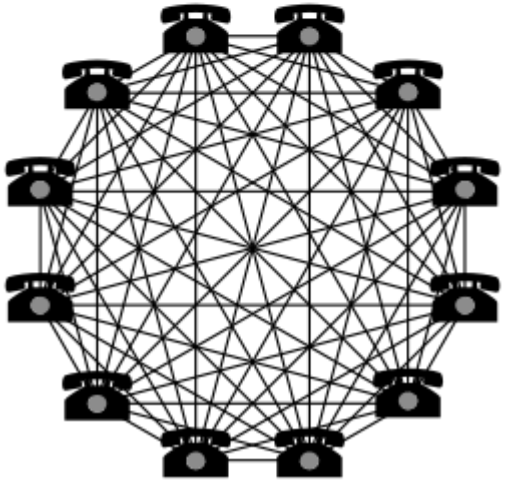


Image Source: Wikipedia: http://en.wikipedia.org/wiki/Metcalfe%27s_law

Objectives of this Lecture

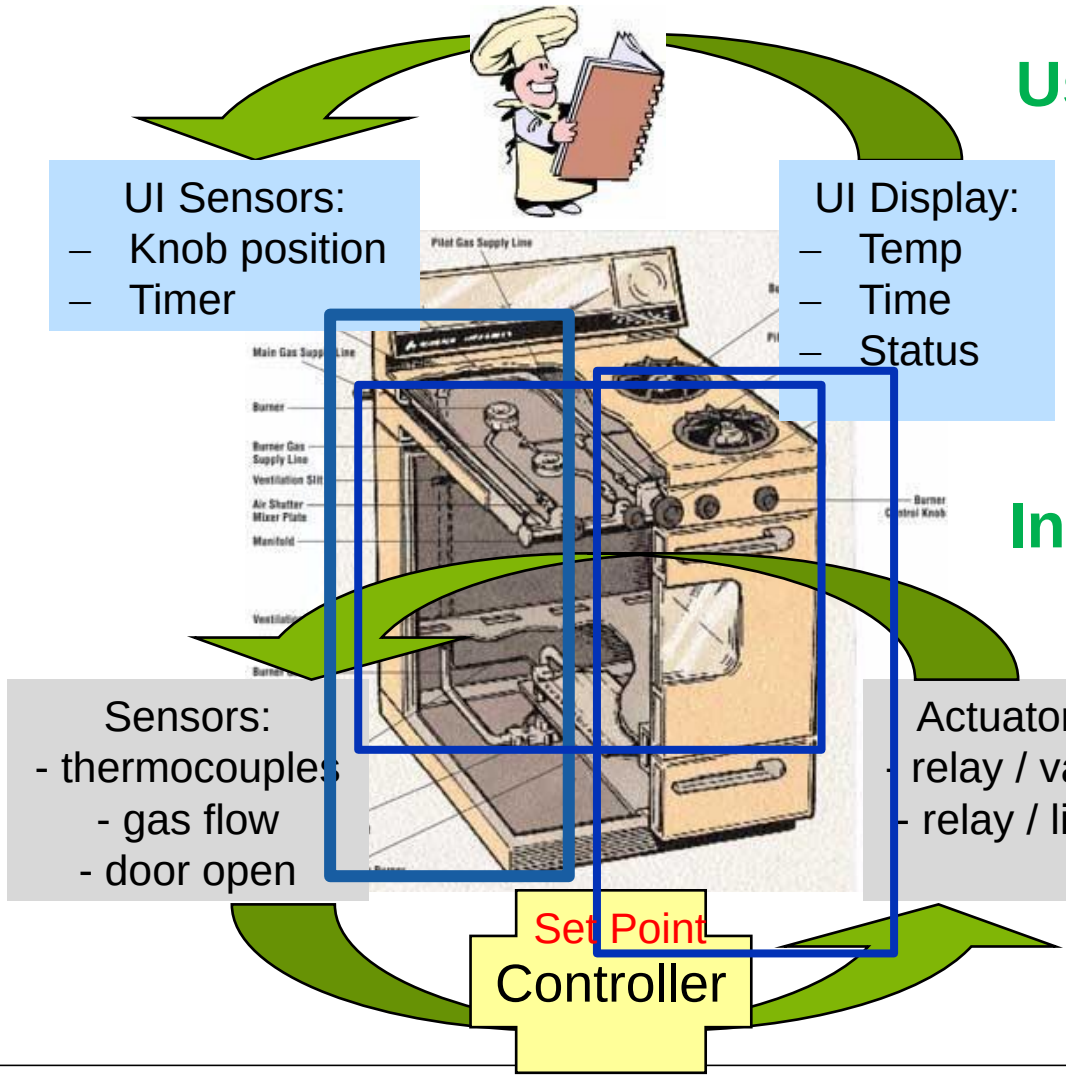
- Introduction to **smart objects**. What and where are they? What do they bring?
- To get an overview of the **technologies** that enables smart objects?
- Get a first understanding of Internet of **everyday things**.
- Re-examine the **IP architectural principles** and see why it is suitable for networking of smart objects.
- Get a high level understanding of how smart objects are constructed in terms of **hardware and software**.

- Where are they?
- What services do the bring?
- What technologies come into play?

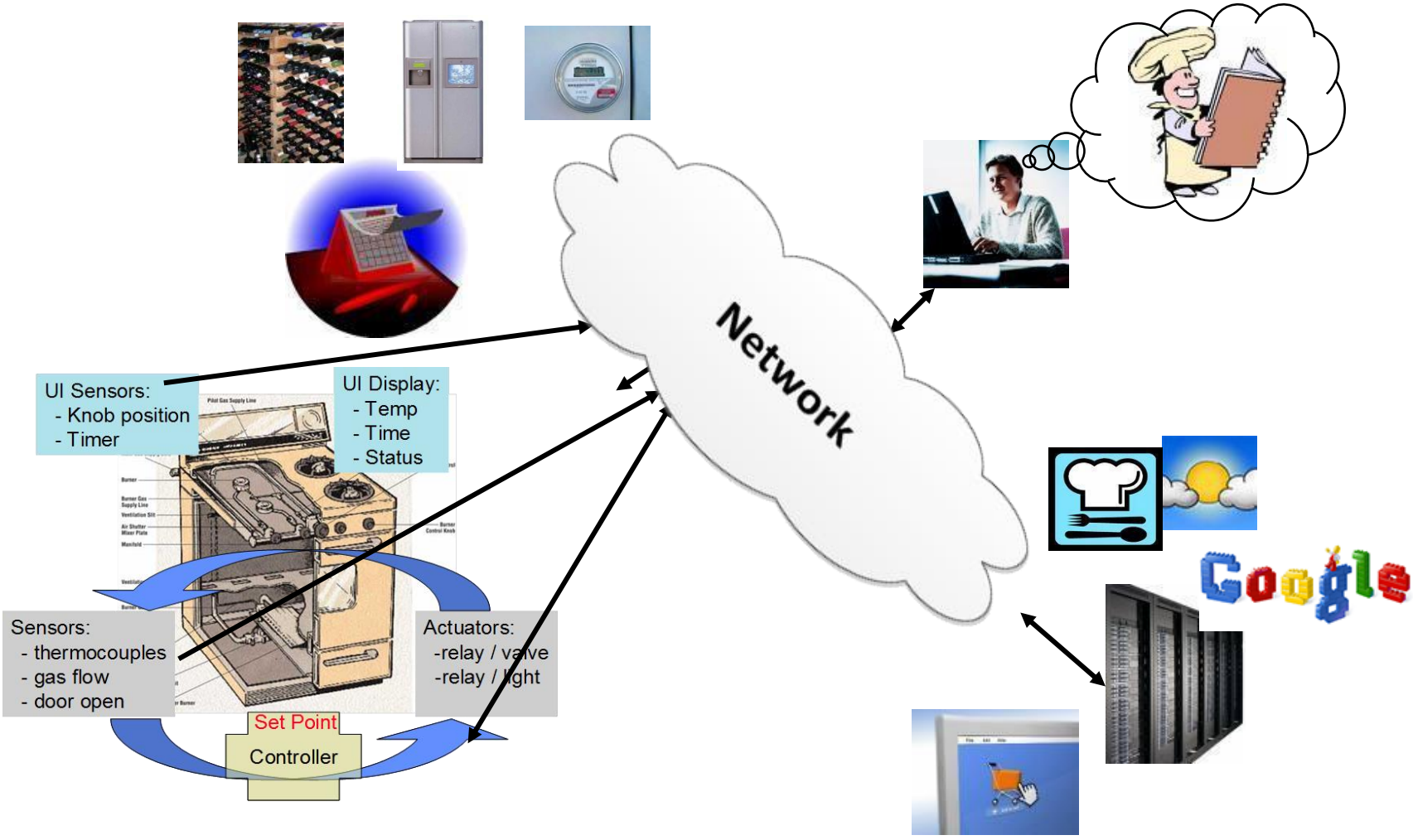
"Smart objects are everyday physical objects enhanced with sensing, processing and **communication** abilities."

Achilles Kameas, 2003

Example of an “Everyday Thing”



Networked Smart Objects



Example - The Modern Car

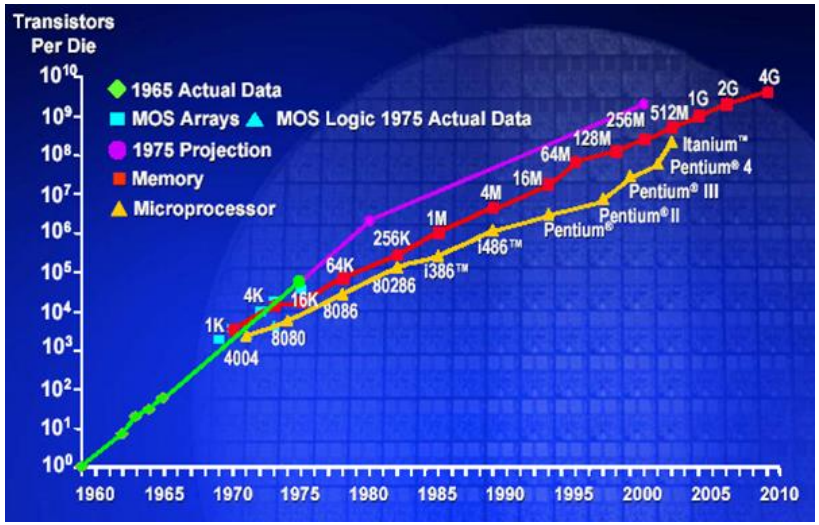
Embedded systems within a car as example

Source: Daimler

- **What are the sensors? How to interact?**
- **What control loops are there?**
- **Which sensors form networks?**
- **What could you imagine would come if these sensors were accessible via a global network?**

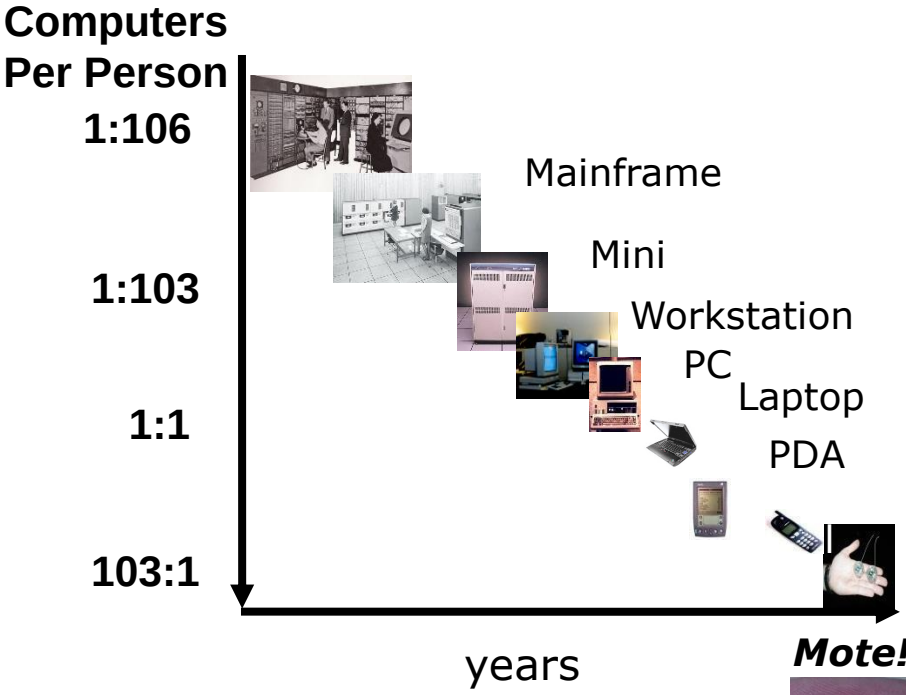
Technology Trends

Moore's Law: # transistors on cost-effective chip doubles every 18 months



Today: 1 million transistors per $

Bell's Law: a new computer class emerges every 10 years



Same fabrication technology provides CMOS radios for communication and micro-sensors

Enabling Technologies

Microcontroller

Flash Storage

Radio Communication

IEEE 802.15.4

Sensors

Challenges

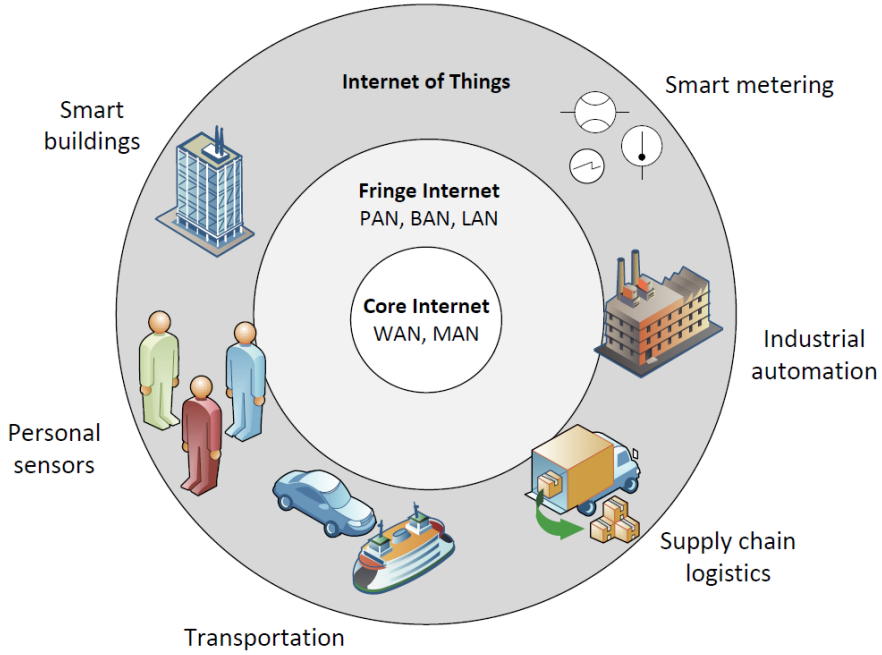
Node/device level
- Harsh environment
- Constrained sensor devices (power, cost, size)

Network level
- Lossy environments
- Large scale deployments

Standardization
- Enable reuse
- Integration in existing systems

Interoperability
- Interworking of components from different vendors

Privacy & Security
- of critical systems



Architecture – IP Based

Overview

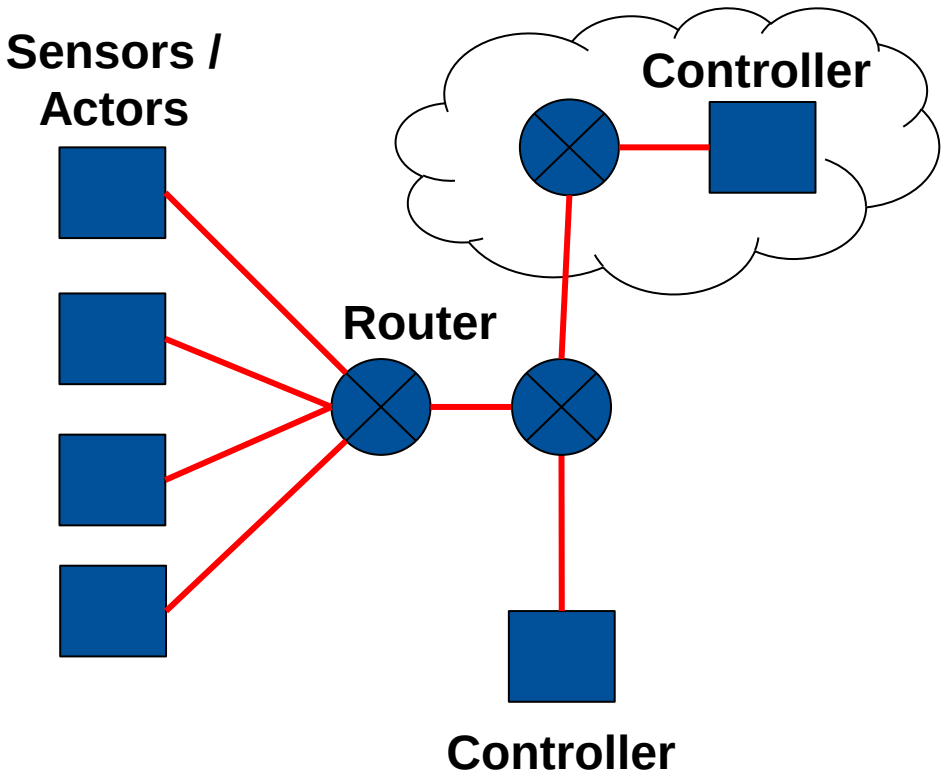
- IP Based communication
- In the whole network
- Full End to End connectivity

Advantages

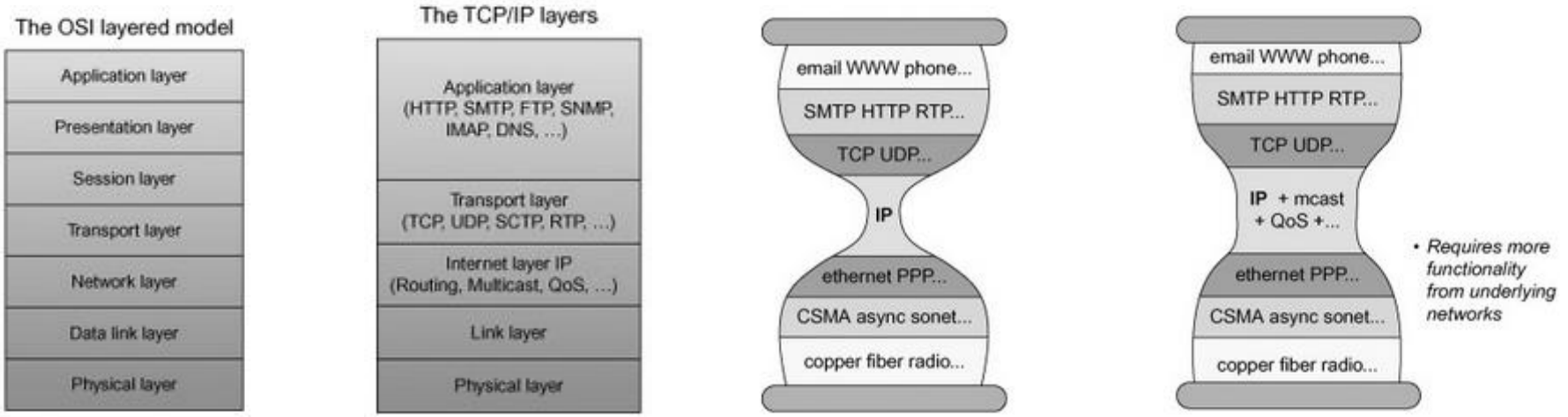
- Open standard based (IETF)
- Open for extensions
- Easy integration
- Evolvability
- Scalability
- Interoperability at the network layer

Disadvantages

- Usage of IP causes protocol overhead
- Hardware (might be) more expensive



TCP/IP Protocol Layered Models



"IP over everything and everything over IP"

Source: Vasseur and Dunkels (2010)

Architecture - Non-IP based End Systems

Overview

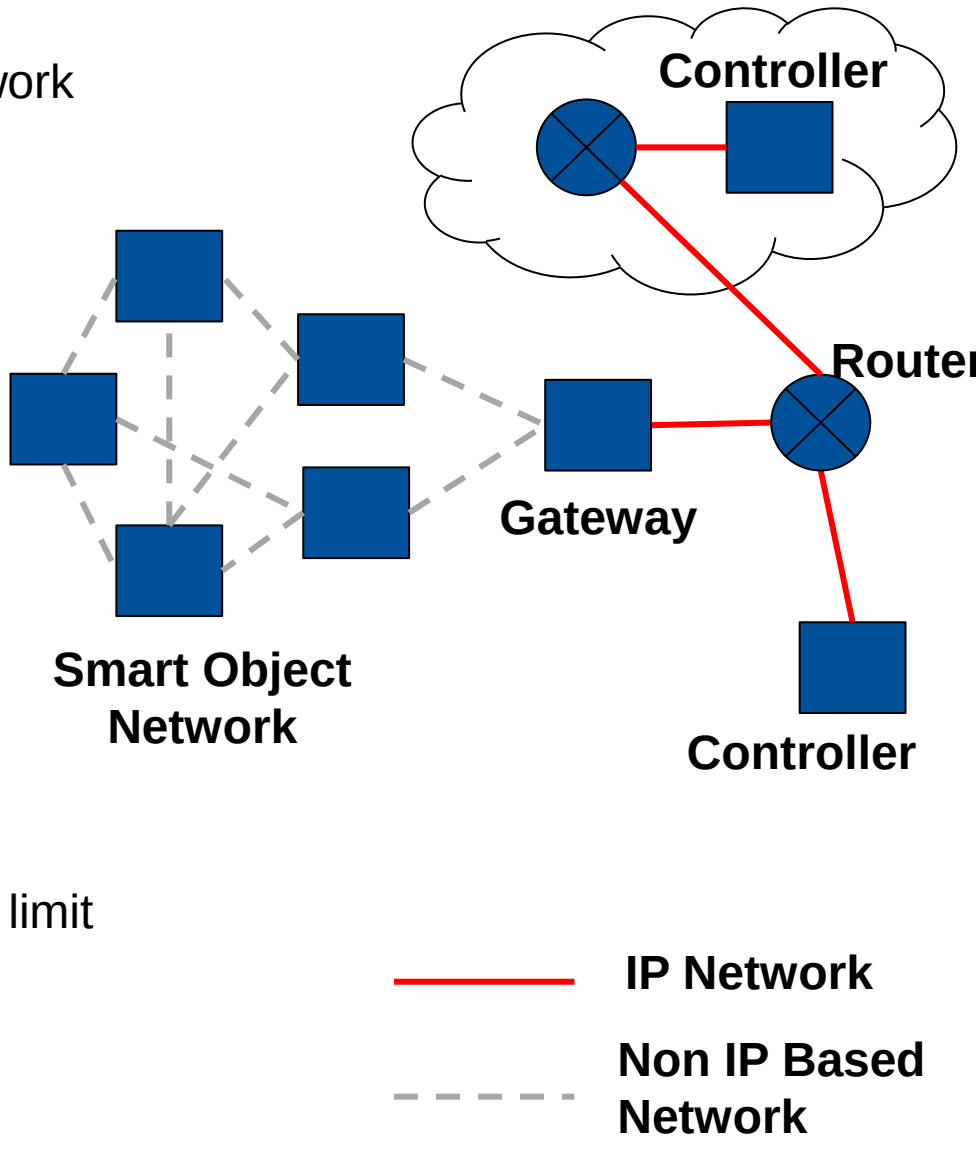
- Non-IP Protocol in Smart Object Network
- Gateway interconnects Smart Object Network with IP based Networks

Advantages

- Potentially lower hardware demand

Disadvantages

- Inherent complexity
- E.g. complex language and protocol translation mechanisms
- Lack of flexibility and scalability
- E.g. potential stateful gateways could limit the scalability



Protocols - Overview

Requirements

- Usable in constraint environments
  - Potentially low bandwith available
  - Networking might be expensive
  - Networking might be unreliable
- Scalability
- Security
- Maybe: Reliability Constraints

Available Protocols

- CoAP
- MQTT
- 6LoWPAN
- (…)

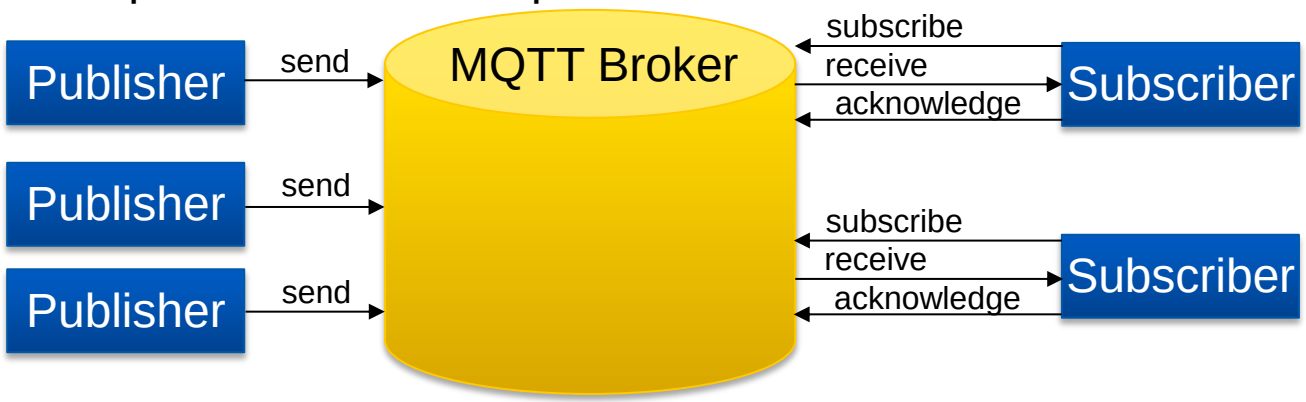
TECHNISCHE UNIVERSITÄT DARMSTADT

MQTT

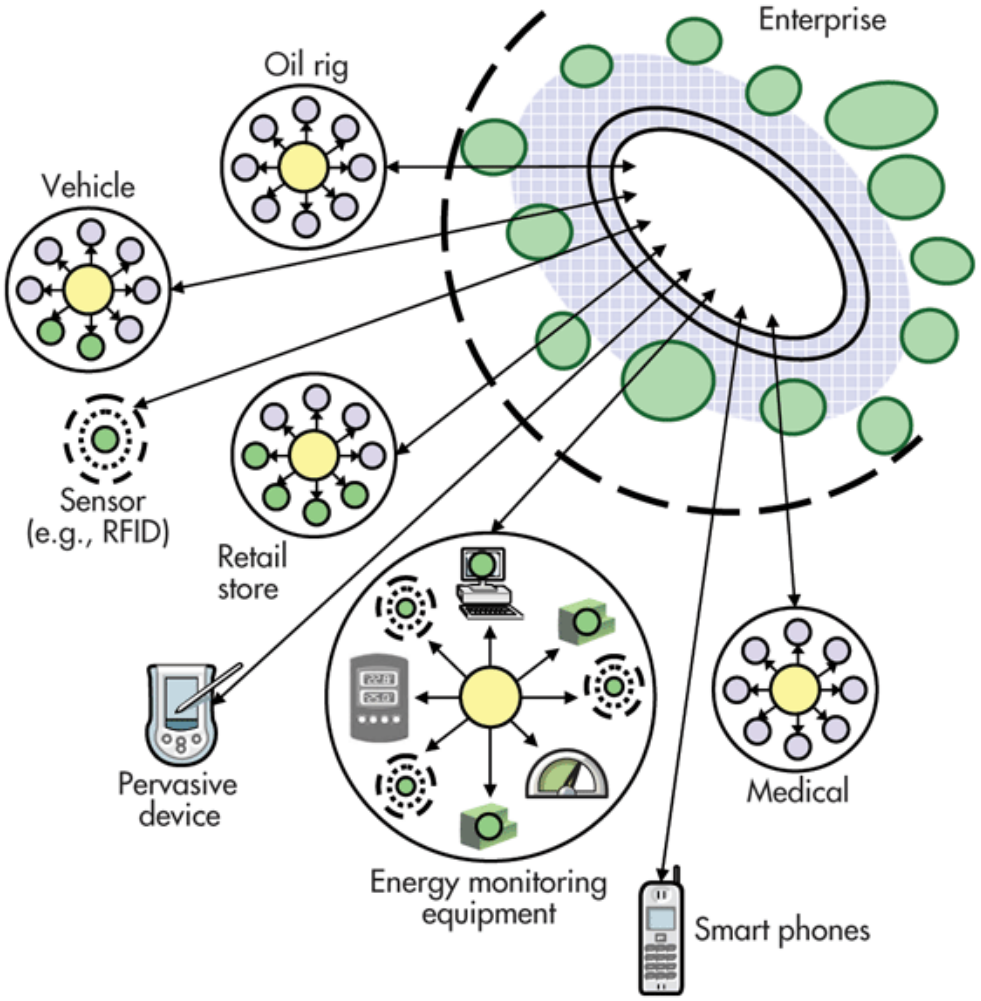
- Proposed as OASIS Standard
- **MQ T**elemetry **T**ransport (MQ stands for **M**essage **Q**ueue)
- lightweight broker-based publish/subscribe messaging protocol
- Based on TCP
- Allows delay tolerant, reliable messaging
- Interconnect “things” with the cloud

Features

- Reliable Messaging
- Minimal the on-the-wire footprint
- Expect and cater for frequent network disruption → Last Will and Testament



Protocols - MQTT



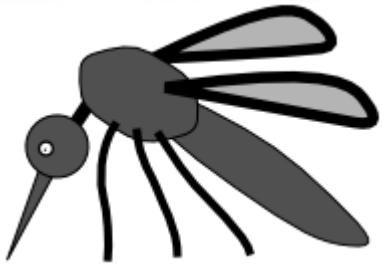
Protocols - MQTT

Implementations

- Mosquitto
- StormMQ
- RabbitMQ
- Eclipse Paho
- IBM WebSphere

Biggest (known) Deployment

- Facebook Messenger (>200M users)



Message Format

<Fixed Header>[<Variable Header>]<PayLoad>

Fixed Header

- Length: 2 Bytes
- Required for all messages
- Fields:
  1. Message Type
  2. DUP flag
  3. QoS level
  4. RETAIN
  5. Message Length

Variable Header

- Protocol Name + Version
- Connect Flags
- Last Will

Messages

- **Connect** - Client request to connect to Server
- **Publish** - Publish message
- **Subscribe** - Client Subscribe request for a certain topic
- **Unsubscribe** - Remove Subscription
- **PingReq** - Keep Alive Messages
- **Ack** - Acknowledge received message
- **Disconnect** - Client is disconnecting

Subscription

- SUBSCRIBE message allows a client to register an interest in one or more topics with the server
- Messages published to these topics are delivered from the server to the client as PUBLISH messages
- Persistent subscriptions survive disconnects. Messages published after a disconnect are stored until the client connects again

TECHNISCHE UNIVERSITÄT DARMSTADT

Topic

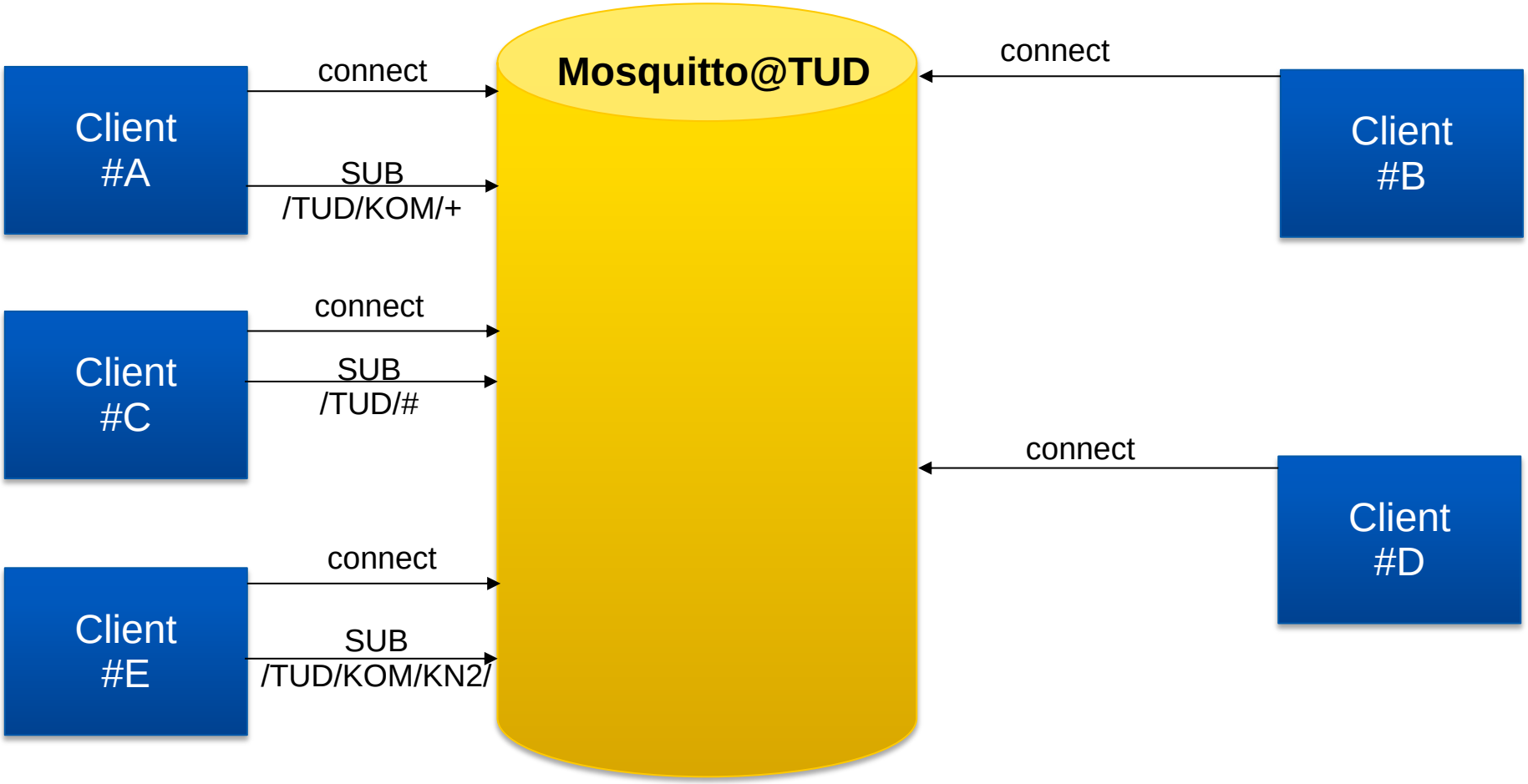
- Named channels.
- Client could PUBLISH messages to such a channel
- The published message is deliverd to all clients who registered an interest with this channel
- Naming is hierarchical – Delimiter is /
- Wild cards for subscription: (# for multi level, + for single level)
- **Examples**
  - TUD/KOM/FRANK
  - TUD/KOM/IRINA
  - TUD/MOODLE/KN2/NEWS
  - TUD/#
  - TUD/KOM/+

} Subscription only

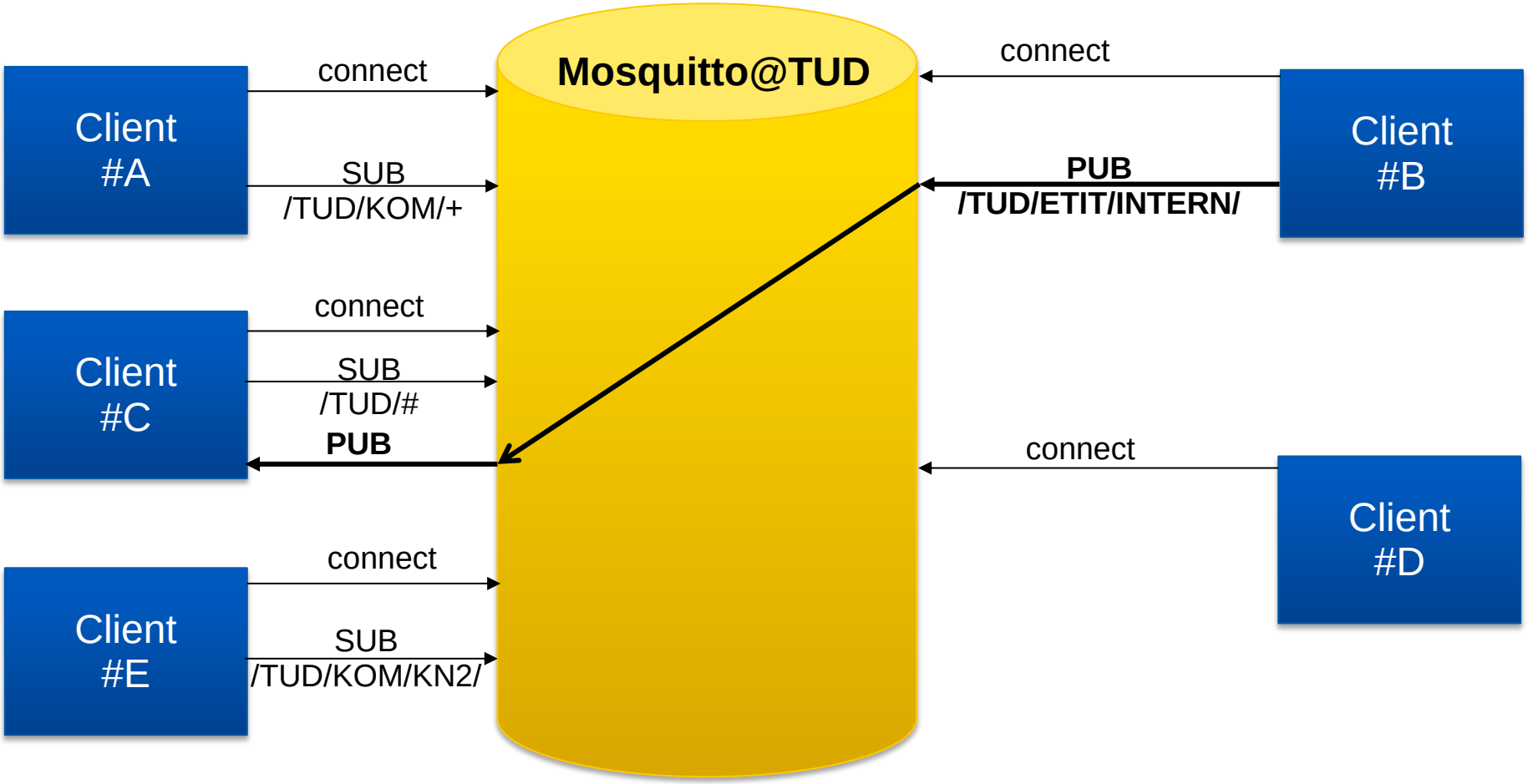
Last Will

- Client leaves an arbitrary message at the server
- Server sents this message only if client disconnects unexpectedly
- Useful for actions required on disconnect

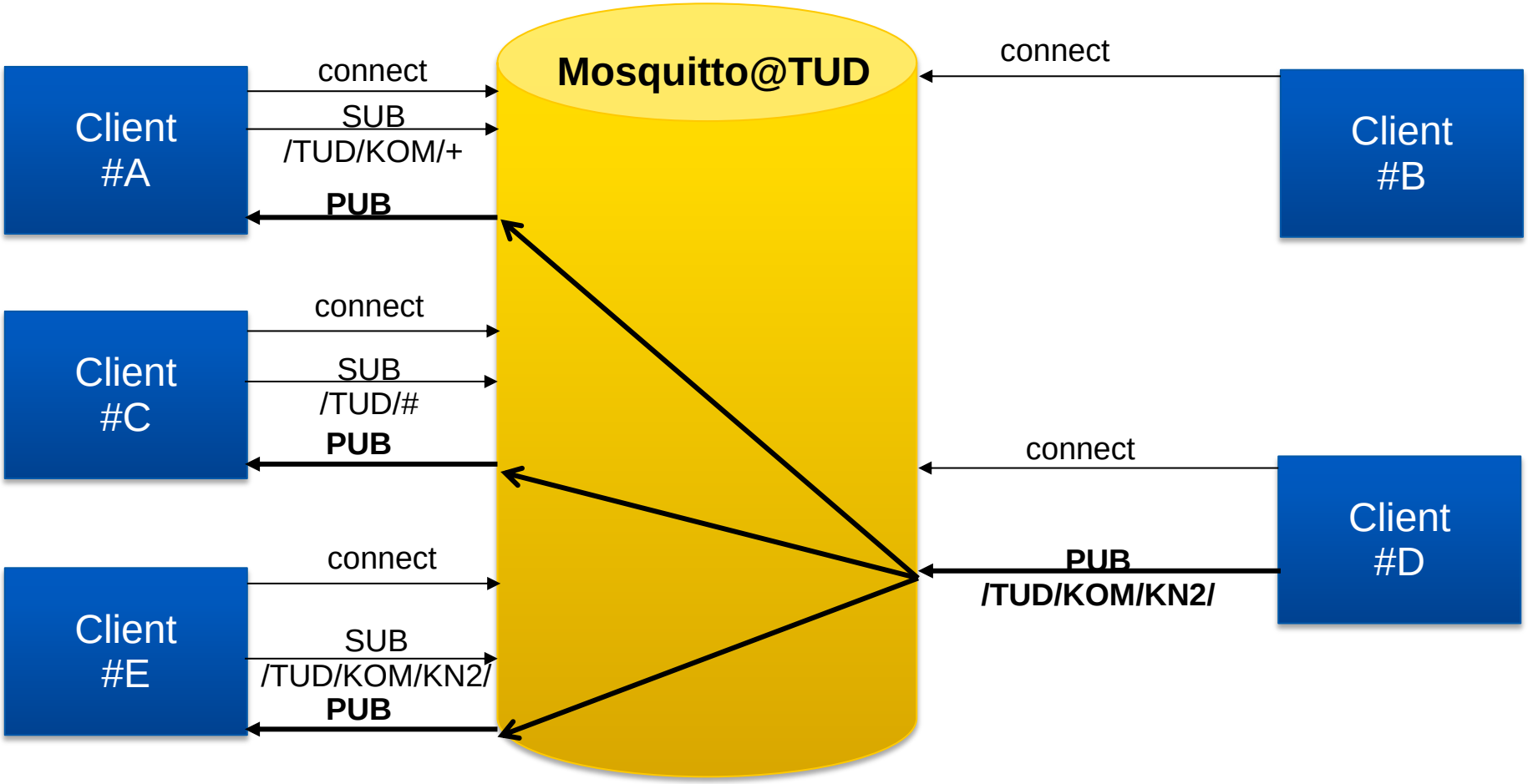
MQTT Example



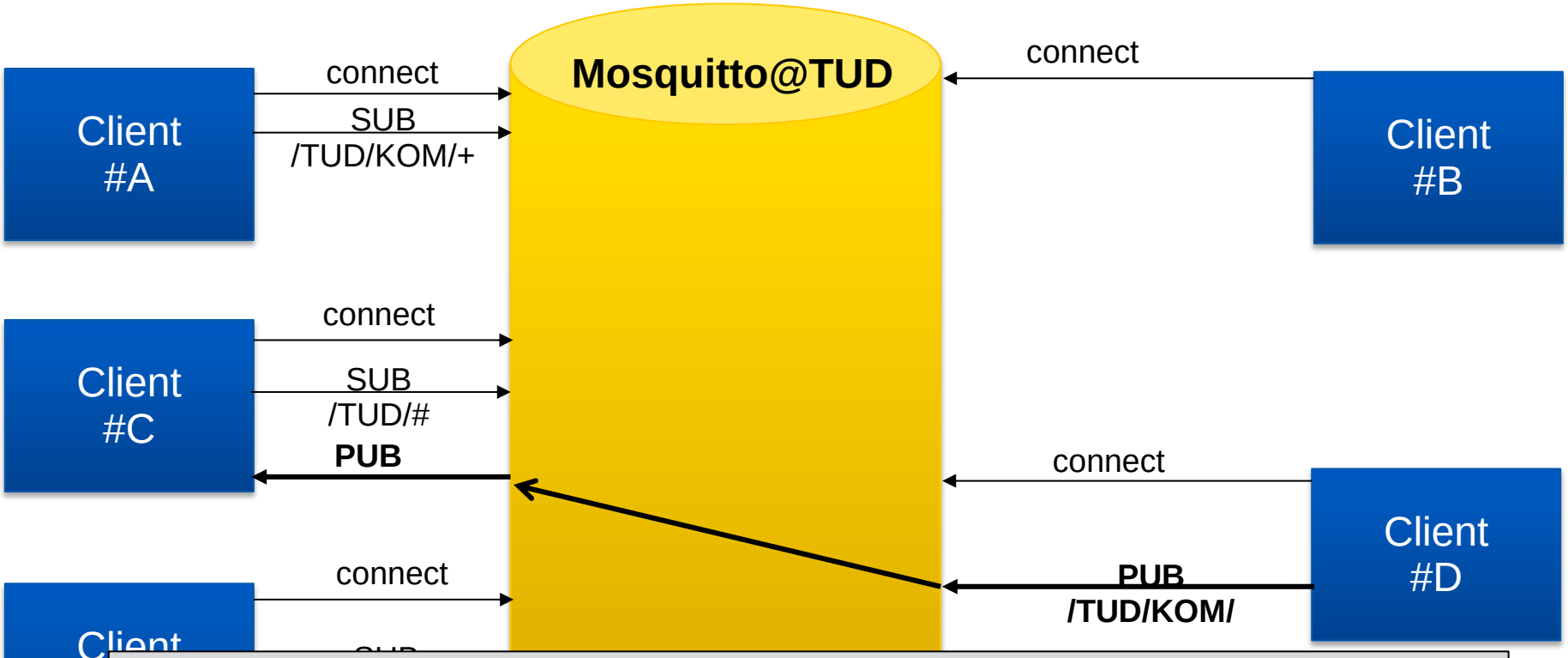
MQTT Example



MQTT Example



MQTT Example

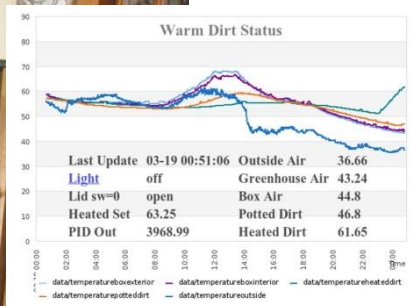


Explanation:

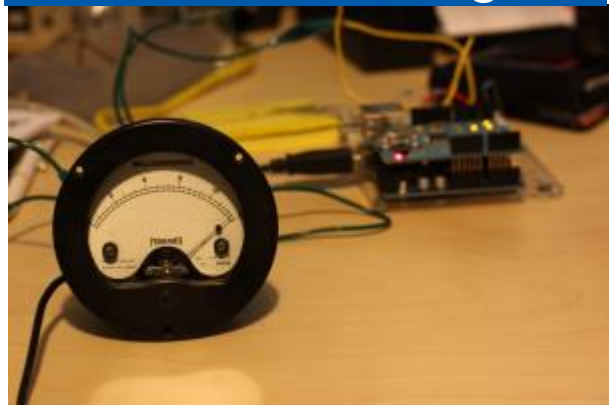
\+ does not match empty topic levels

(Fun) Use Cases

Smart Garden



Smart Metering



Smart Taxi

Real Use Case

TECHNISCHE UNIVERSITÄT DARMSTADT

facebook Registrieren

Building Facebook Messenger

von Lucy Zhang, Freitag, 12. August 2011 um 09:00

On Tuesday we introduced Facebook Messenger, a new stand-alone messaging app that enables people to send messages 1-on-1 or to groups of friends. I joined Facebook five months ago with my two other co-founders, Ben Davenport and Jon Perlow, who worked with me to build a group messaging application called Beluga. Our new app, Facebook Messenger, represents the best of both worlds — it combines the ease and simplicity of Beluga with the scale and integration of Facebook Messages.

I had a couple of life experiences that led me to create Beluga. For instance, a group of us were meeting up for a movie at the Tribeca Film Festival in New York City but the hour leading up the movie was a total communication failure. I had a friend texting to tell me another friend was running late. We were saving a ticket for another friend, but he decided not to come and no one knew. It wasn't until hours later when I returned home that I saw an IM from the friend who was late and an email from the friend who had bailed. We think Facebook Messenger's ability to integrate chat, text messages and email helps solve this exact problem.

We started building Beluga as a tool for group coordination but we discovered that enabling lightweight, private, instant communication can change the way a group of people connect with each other much more broadly. Instead of emailing out the vacation photos weeks after a trip, people start to share more in the moment. The instant nature of the messages enables group conversations to start up spontaneously and bridges the gap between people connecting from their computers and those on mobile devices. Messenger's integration with Facebook Chat now makes that scenario a reality.

Facebook Messenger

Home Automation

Open Source Home Automation

openHAB is a software solution developed in Java and is designed to be absolutely vendor-neutral as well as hardware/protocol-agnostic. It can run on devices like the Raspberry Pi and lets you integrate an abundance of different home automation technologies into one. As it is fully open source and is maintained by a passionate and growing community, it is the best way to make sure that your Smart Home stays flexible in the future and that new upcoming technologies can be easily integrated with it and that you are not trapped in any vendor lock-in.

Overview