UbiComp Summary

Frédéric Vogel `vogelfr@ethz.ch` , ETH Zürich, FS18

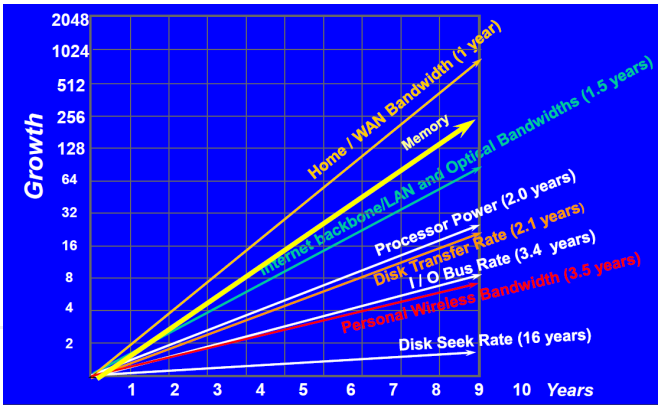
- UbiComp Summary
- Selected UbiComp drivers
  - Moore's Law
  - New Materials
  - Progress in Communication Technologies
    - Communication by Touching (NFC)
    - Intrabody Communication
  - Better Sensors
    - Surface Acoustic Wave Sensors
- Wireless Communication
  - Basics
    - Radio duty cycling
    - Energy harvesting
    - Passive WiFi
    - Multiplexing
  - Long Distance
    - 5G
    - LPWAN
    - SigFox
    - LoRa
  - Short Distance
    - Bluetooth
    - IEEE 802.15.4

- RFID
  - Power supply
  - Operation frequency
  - Communication, coding, modulation
    - Reader -> Tag
    - Tag -> Reader
  - Anti-collision protocols
    - Deterministic protocols
    - Stochastic protocols
    - Adaptive round algorithms
  - RFID vs NFC
- Internet of Things, Web of Things
  - Issues
  - Representational State Transfer (REST)
    - Shared identification model
    - Uniform interfaces
    - Self-describing representations
    - Stateless interactions
    - Hypermedia As The Engine Of Application State (HATEOAS)
    - REST Architectural Style
  - HTTP
    - HTTP Requests
    - HTTP Header and Response
    - Cross-origin Resource Sharing (CORS)
  - Constrained Application Protocol (CoAP)
    - Observing resources
    - Group communication
    - CoAPS
    - Reliability
  - Semantic Technologies
- Smart cards
  - SIM
  - Wireless smartcards
    - Authentication
- Location

Selected UbiComp drivers

Moore’s Law

- Most important technology parameters double every 1-3 years
- But: doesn’t apply to battery capacity, and some others



New Materials

Materials determine technology and shape society
Examples:

- flexible displays
- electronic ink
- organic semiconductors
- graphene

Progress in Communication Technologies

- Fiber optics: from Gbps to Tbps
- Wireless
  - GSM, UTMS, 4G, 5G
  - LPWAN, LoRa
  - Wireless LAN, WiFi (>> 10Mbps)
  - Bluetooth, Zigbee
  - NFC

Communication by Touching (NFC)

- Almost no energy
- Very small transmitters
- Cheap
- No addressing
- No routing

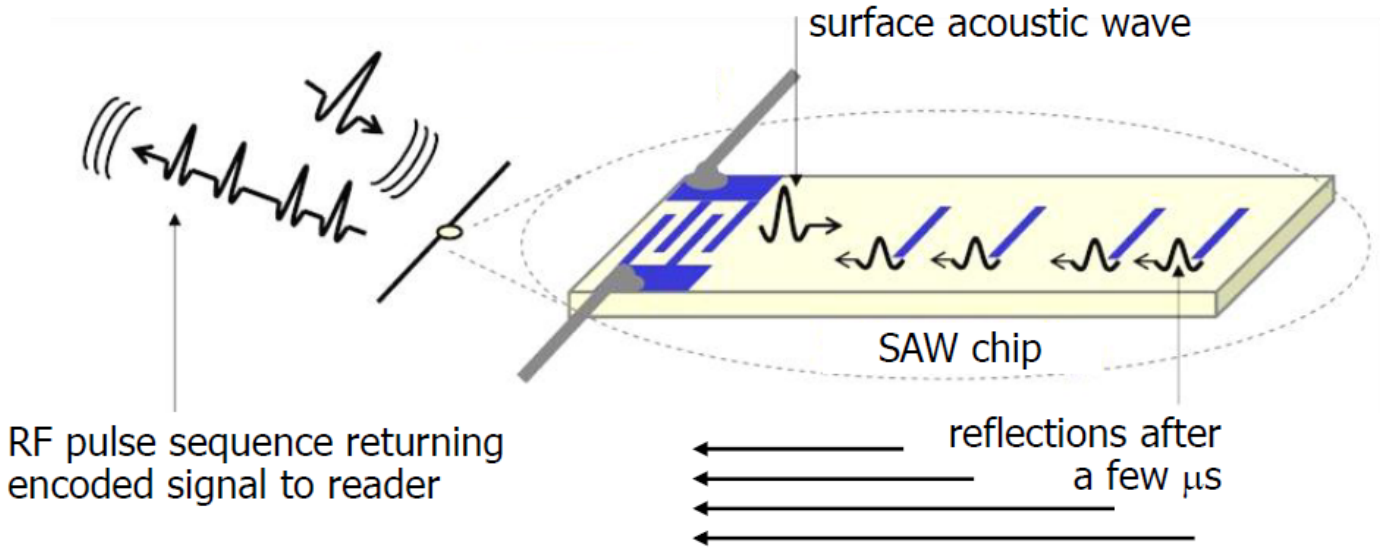
Intrabody Communication

- Low power electrical signals sent through human body
- Allows wearable devices to communicate
- Enables touch-selective communication

Better Sensors

- Miniaturized cameras, microphones, ...
- Biometric sensors
- Sensors for temperature, humidity, acceleration, ...
- Location sensors
- Autonomous perception of the user's environment
    - Context awareness
    - Situation awareness

Surface Acoustic Wave Sensors



- RF pulse $\rightarrow$ SAW
- SAW reflected by Piezzo reflector
- Reflected SAW to RF pulse sequence as answer

Leads to Cyberphysical Systems (CPS)

Wireless Communication

UbiComp promises lots of sensing in the field and lots of processing power in the cloud. How to connect UbiComp functionality?

Basics

Radio duty cycling

Core Idea
Exploit energy assymmetry in sensor networks

Duty Cycle
Fraction of a period in which a given system is active

Low-power Listening
Receiver periodically turns on radio and polls medium

Low-power Probing
Receiver periodically send probes, sender listens for probes and sends payload immediately after receiving probe

Energy harvesting

- Photovoltaic
- Piezoelectricity
- Pyroelectricity
- Atmospheric pressure changes
- Ambient radiation

Passive WiFi

- RF components offloaded to single plugged-in device in network
- Device creates single-frequency tone
- Passive Wi-Fi devices communicate by reflecting this tone via digital switch $\rightarrow$ “real” 802.11b transmissions

Multiplexing

Multiplexing
Multiple use of shared medium

Multiple options:

- Space (build three motorways)
- Frequency (build multi-lane motorways)
- Time (Have cars use lanes one after the other)
- Code (Put people into a bus)

DS-CDMA

Senders use unique pseudorandom binary chipping sequences, XOR their signal with sequence
Example: 802.11b WiFi

FH-CDMA
Discrete changes of carrier frequency determined via pseudorandom number sequence
Example: Bluetooth

Long Distance

5G

Goal: improve 4G services by combination of upgrades, new technologies

- Wider bands
- More antennas
- Softwareization and adaptivity (e.g. SDN / logically isolated network partitions)

Milimeter wave bands
26 GHz band opened on EU level

- Pseudo-optical propagation characteristics
  - Blocked by walls, significant attenuation by plants
  - High atmospheric attenuation and rain fade
  - Doppler shift problematic even at low speeds

Massive MIMO

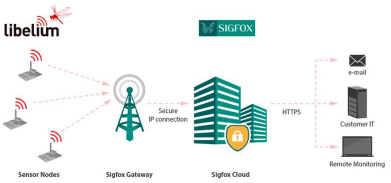
- Spacial multiplexing
  - Transmit separately encoded data signals from each of the antennas
  - Increases channel capacity + possiblity to transmit to multiple users (SDMA)
- Precoding / Multi-stream beamforming
  - Emit same signal from all antennas
  - Set phase and gain to maximize singla power at receiver
- Diversity coding
  - If transmitter has no challenge knowledge
  - Send identical copies of the data stream from different antennas with different encoding

LPWAN

- Long range, low number of base stations, low base price
- Low bandwidth (< 10kbps)
- Ideally low latency

- Low power (batter lifetime > 10 years on 5Wh battery)

SigFox



- Public ISM band (868MHz in Europe, 902MHz in US/FCC)
- Range of several km, up to 1000km
- Device transmission strictly regulated
  - max 30s per hour
  - max data range 100bps, max message size 96bits
- Unique Sigfox ID per device
- Fire-and-forget transmission mode (no ACKs)
- Bidirectional communication
  - device is informed of "demand for control" at next uplink → switches to receive mode

→ better for unidirectional extreme long range links
if subscription model is acceptable

LoRa

- Public ISM band
- Range of several km, up to 10s of km
- Chirp Spread Spectrum
- 3 Device classes: A (basic), B (beacon with scheduled receive slots), C (continuous)
- Data rate of 0.3 - 50kbps, larger messages than SigFox
  - allowed duty cycle for LoRa is 1% (in EU)
  - The Things Network Fair Access Policy: 30s airtime per device per day
- Device names based on IEEE EUI64 (MAC++)

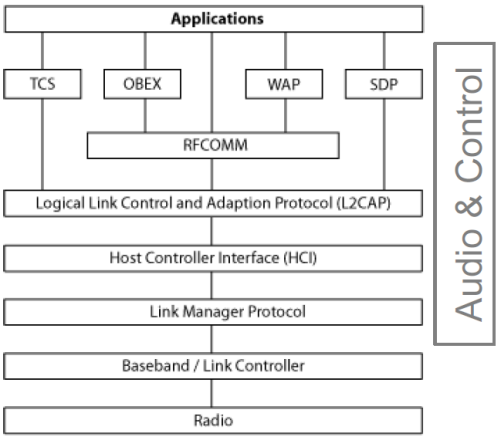
→ better for true bidirectionality, requires denser infrastructure
more open, more expensive hardware

Short Distance

Bluetooth

- Original goal: much smaller, cheaper, less power than WLAN
- spontaneous, infrastructure-less networks
- Now competes in many respects with 802.11
- BLE marketed as *the* short range IoT-technology
- complex API
  - complete communication stack (802.11 defines only 2 layers)

Ad-hoc networks with no a-priori central coordinator



- ISM band (2.4GHz)
- 79 channels à 1MHz width ($2402\text{MHz} + k\text{MHz}, k = 0, \ldots, 78$)
- Frequency hopping: 1600 hops/s (625µs)
- 1mW transmission power

Frequency hopping

- Interference protection, enhanced robustness
- Enhances security (hopping pattern not known to outsiders)
- Collisions with adjacent Bluetooth links possible
- Many other devices use same frequency band
- Collision? Retransmit packet on different channel

Baseband layer

Specifies:

- Hopping sequence
- Searching and connecting to other devices
- Packet formats
- Error handling, retransmissions
- Master/slave roles

Addressing

- each device has 48-bit device ID (compatible with IEEE 802 MAC)
- 3 bit active member address (AMA) when active in piconet $\rightarrow$ max 7 slaves, address 0 for broadcast to all slaves
- 8 bit parked member address (PMA) $\rightarrow$ max 255 parked slaves in piconet

Piconet

- Bluetooth units sharing single (frequency-hopping) channel
- Up to 2.1Mbps
- Time multiplex and duplex within piconet
- Master sets clock for all slaves
- All devices hop together (determined by master device ID)
- Point-to-point master-slave or multicast from master to all slaves

Inquiry

Source (“Inquiry”):

- Inquiry packet transmitted on new frequency every 312.5µs (3200 hops/s, listen after each packet)

Destination (“Inquiry Scan”):

- Changes listening frequency all 1.28s (“Set device to discoverable”)

To find each other devices must be in complimentary states (inquiry / inquiry scans) on same channel

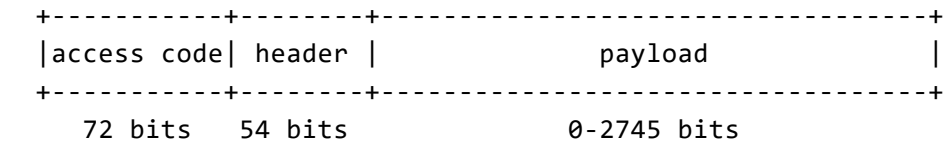
Paging
Invitation to known device to join, paging device becomes master

Communication

Data packets
Filled in dedicated slots of 0.625ms, stricly assigned to master or specific slave

Control packets
Usually on next slot, ACKs + Flow control
`NULL` when slave has nothing to send, `POLL` when master has nothing to send

Packet format



Access code identifies all packets of piconet

Header (repeated 3 times), contains:

- 3 bits AMA
- 4 bits payload type indication
- 1 bit stop/go flow control switch set by receiver
- 1 bit ACK/NAK flag
- 1 bit odd/even seq. number
- 8 bit header checksum

Error handling

Automatic Repeat Request (ARQ)
Determined from 1 bit ACK/NAK, 1 bit seq. number

Forward Error Correction (FEC)

Reduces number of repeat requests, but induces general overhead

Low-power modes

Saving power by

- Adapting tx power
- Adapting hopping sequence
- Compressing data
- Low power modes

Sniff state (keep AMA)
Slave wakes up every x ms to check if packet is available, master knows when slaves listens

Hold state (keep AMA)
Master puts device on hold temporarily, slave returns automatically

Park state (release AMA)
Stays virtually connected to piconet, but no participation anymore. Listens only to synchronisation beacons.

BLE: 15mA peak, 1µA average

Security

- Fast frequency hopping
- Low tx power
- Link layer encryption (128bit AES)
- Pairing
  - PIN
  - Number comparison
  - NFC

L2CAP

- Adapts upper layer protocols to baseband
- Segmentation and reassembly of upper layer protocol packets
- Protocol multiplexing over single air interface

RFCOMM

Emulates serial port, similar to TCP

Bluetooth Profiles

Vertical slices through protocol stack, specifications for interoperable applications

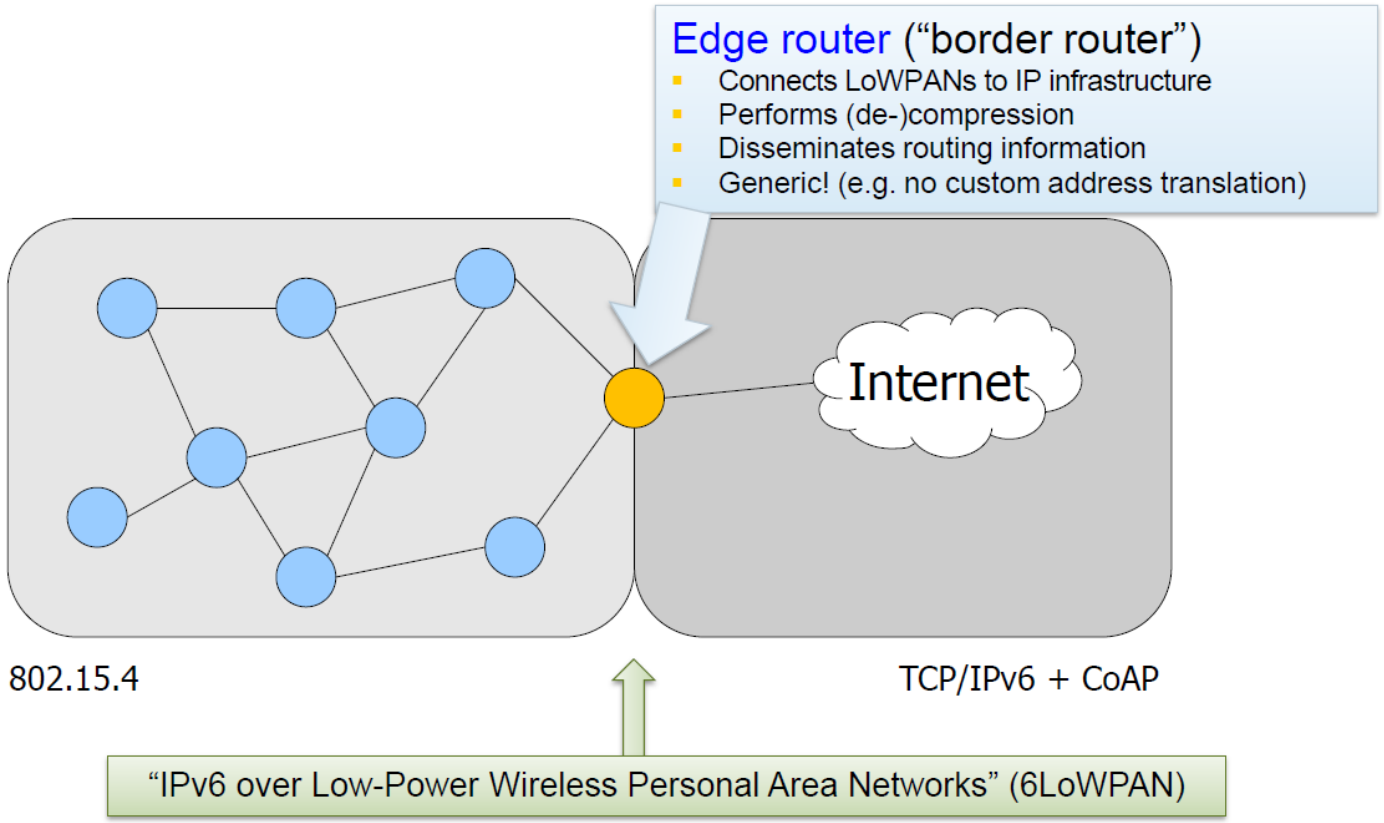
BLE

- Diminshed power, up to 260kbps
- Fewer but broader channels
- Modulation scheme optimised for low power usage
- Faster connection setup

Generic Attribute Profile (GATT)

BLE application layer protocol, enables reading and writing of remote variables ("characteristics")
Related characteristics grouped to *services*

IEEE 802.15.4



IEEE 802.15.4

- Defines PHY and MAC layer
- uses CSMA and TDMA
- provides frame security with AES

Low-power wireless personal area network (LoPWAN)

- 20kbps - 250kbps, 127Bytes per frame
- 16 channels in 2.4GHz band, 10 channels in 915MHz band, 1 channel in 868MHz band
- Master-slave or P2P operation

6LoWPAN

- Adaptation of IEEE 802.15.4 lower layers to IP
- Goal: direct end-to-end internet integration of resource-constrained embedded devices
  - Use IP as narrow waist
  - Simple transformation between compressed IPv6 and uncompressed format
- IPv6 packets over IEEE 802.15.4 networks
  - 6LoPWAN provides standard socket API (RFC 4944)
  - Header compression given dedicated 802.15.4 context
  - Packet fragmentation
- Two routing strategies
  - Mesh-Under: emulate local link over multiple hopsin link layer
  - Route-Over: perform routing at IP level
- Pure adaptation protocol

ZigBee

Based on 802.15.4 lower layers

- Defines network and application layers
- Implements routing and multi-hop topology
- Provides useful general services (discovery, security, etc.)
- Used in home application, smart energy, hospital care, toys, ...

RFID

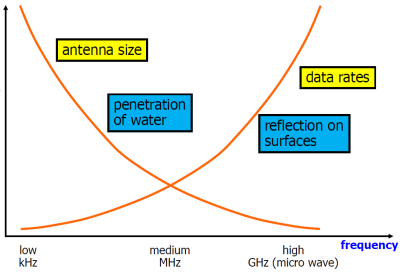
3 Categories

1. Low-end features
   - read only
   - tag repeatedly sends serial number (while powered)
   - no collision detection
2. Medium-range features
   - collision detection (30items/s)
   - read-write memory (EEPROM/SRAM)
3. High-end features
   - complex functions such as crypto $\rightarrow$ contactless smartcards

Power supply

Inductive coupling (magnetic field)
EEPROM need significantly more energy than ROM
Need multiplexed antennas with different orientations (magnetic induction $u = u_0 \cos \delta$)

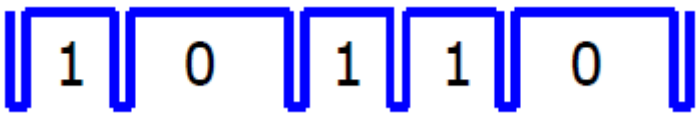
Operation frequency



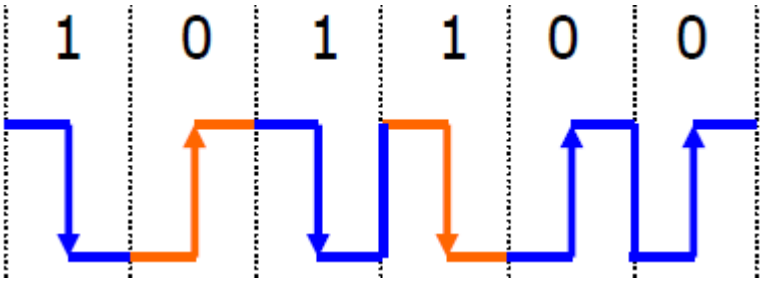
Communication, coding, modulation

Reader -> Tag

Tag -> Reader



Anti-collision protocols



- Transponders don't hear signals from other transponders, only from reader
- FDMA?
  - Limited number of frequencies / different channels
  - More expensive readers
  - Only suitable for particular applications

Deterministic protocols

- All present tags detected in read cycle
- Induces high overhead in reader -> tag communication

Tree-walking anti-collision algorithm

- With Manchester encoding it is possible to locate bits where two signals differ

1. Reader broadcasts `sync` command to all transponders
2. Reader requests ID number of all transponders
3. Reader determines leftmost bit b that yields collision. Collision?
   5. Reader broadcasts `mute for value 0 at position b`
   6. Only transponders with value 1 at b move to next round (repeat from 2) all others remain mute
4. Reader requests data from unique transponder x
5. Reader sends `halt` command to x
6. Repeat from 2

Stochastic protocols

ALOHA

Stochastic TDMA, transponders repeatedly send out data with random long quiet periods in between

But: higher load? more collisions

Slotted ALOHA

- Reader periodically sends `sync` command

- All transponders randomly select one of following n time slots to send serial number
- If reader gets unique serial number s in a time slot (no collision) it sends `select(s)` command
- Selected transponder reacts by sending pay-load data (exclusive access), then remains quiet

Adaptive round algorithms

Transponder replies are in slots, number of slots is dynamically altered

RFID vs NFC

	RFID	**NFC**
Communication	One-way	Two-way
Scan distance	Up to 1m	Up to 10cm
Scan tags simultaneously	Yes	No

Internet of Things, Web of Things

Issues

- Connecting things
  - Edge device can look very different from Internet of Computers
  - Low bandwidth, more “bursty”, wish to talk to many sensors at once
  - IoT apps should not require knowledge of these details
- Interconnecting things across IoT silos
  - Broad: resources come from variety of source
  - Deep: involve taking multiple steps / using multiple APIs
- Enable machines to use IoT functionality
  - APIs designed for people difficult to use by machines (buy book -> add to cart -> go to checkout -> etc.)

Web architecture can help:

- Ubiquity

- Openness
- Clearly defined protocol semantics

Representational State Transfer (REST)

1. **Shared identification model**
   allows everybody to refer to available resources (URI)
2. **Uniform interfaces**
   can be used to access those resources
3. **Self-describing representations**
   allow clients to understand those resources

Shared identification model

Name everything you want to talk about

“Things” can refer to *anything*

- Products in online shop
- Categories used for grouping products
- Customers
- Shopping carts

Client state transitions are also represented as resources

- `Next` links on multi-page submission process
- Paged results with URIs that identify following page

Uniform interfaces

The same small set of operations applies to everything (i.e. every resource)

Example: HTTP

- Interacting with resources is possible *without* knowing it beforehand
- Interaction language *only* about making interaction possible and scalable
- Exchanged representation based on *Shared Representation Model*

Self-describing representations

Resources are *abstract* entities (can not be used per se)

- *Shared identification* guarantees clear identification
- Accessed through *uniform interface*

Interaction happens via *resource representation*

- It is communicated which kind of representation is used
- Representation formats can be negotiated between peers
- Whatever the representation, it *must* support links
- Depending on client or machine access representation will (should) be different

Need also *shared representation model* for *shared understandability*

Stateless interactions

- State is kept in resources and clients, *not* in transaction
- *Resource state* managed on server
- *Client state* managed by client

Hypermedia As The Engine Of Application State (HATEOAS)

- Resource representations contain links to identified resources
- Servers guide interactions by providing links
- Links are *possible state transitions* of the client/server application

REST Architectural Style

REST enables scalability, mashup-ability, usability, accessibility

HTTP

- Text-based protocol
- HTTP methods indicates action to be performed on resource
  - same semantics with every resource (uniform interface)

HTTP Requests

HTTP GET
Retrieve representation of addressed resource
Safe: doesn't modify resource representation -> can be cached, can be pre-fetched

HTTP PUT
Modify addressed resource's representation
Idempotent: applying it n times has same effect as applying once -> fault-tolerant APIs (clients can safely retry if time-out from server)

HTTP Header and Response

Request header

Accept, user-agent, connection properties

Response header
Content, metadata, lifetime, etc.

Servers/clients *must* ignore unknown fields (additions can enter protocol over time)

Response codes
Numeric status code + text
2xx for ok
3xx for redirections
4xx for client side problems
5xx for server side problems

Cross-origin Resource Sharing (CORS)

Core security concept: Same-origin policy (script in one web page may access data in second page iff both pages have same origin (URI scheme + host name + port number))

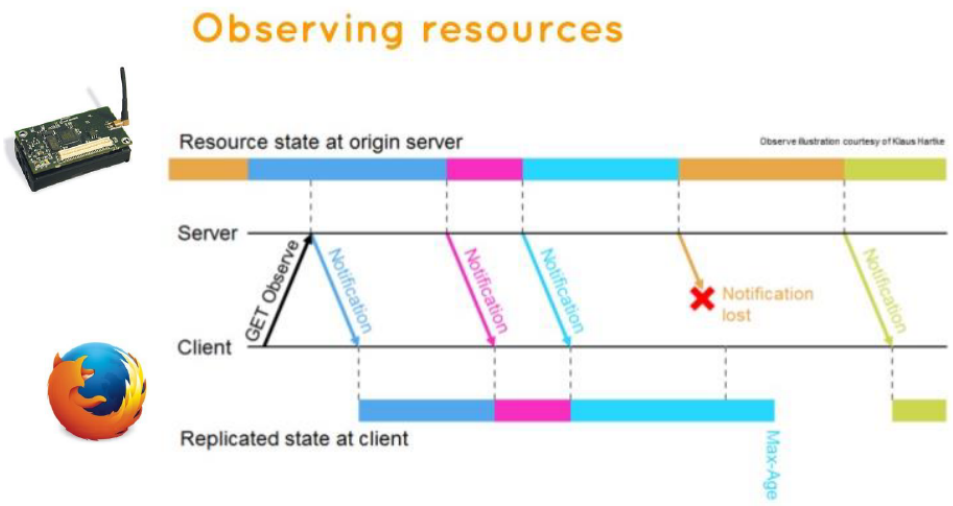
CORS idea: allow target resource to control who may interact with it

Constrained Application Protocol (CoAP)

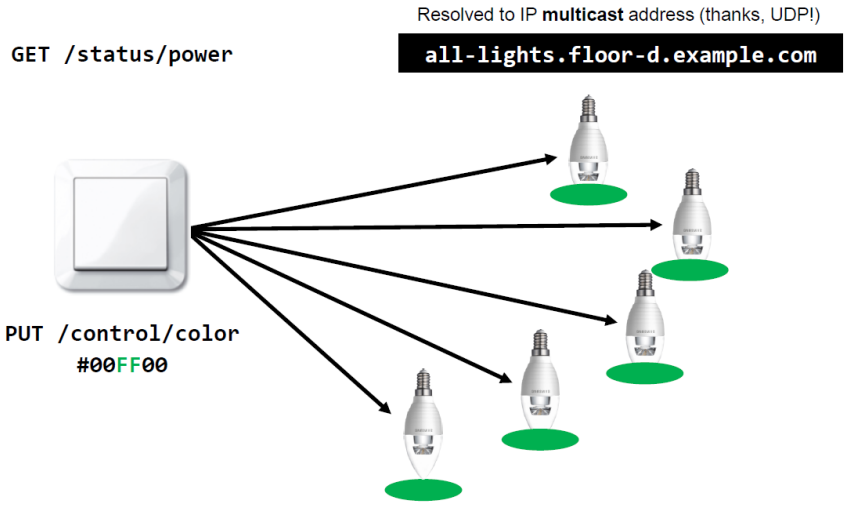
REST protocol optimised for bandwidth and processing efficiency

- Extended subset of HTTP with transparent mapping to HTTP
- Better for near-real time
- No connection managment overhead
- Smaller headers
- Lightweight security
- Specialised for M2M applications: Multicast & Observe
- Easy to proxy to/from HTTP
- Designed for datagram transport protocols

Observing resources



Group communication



CoAPS

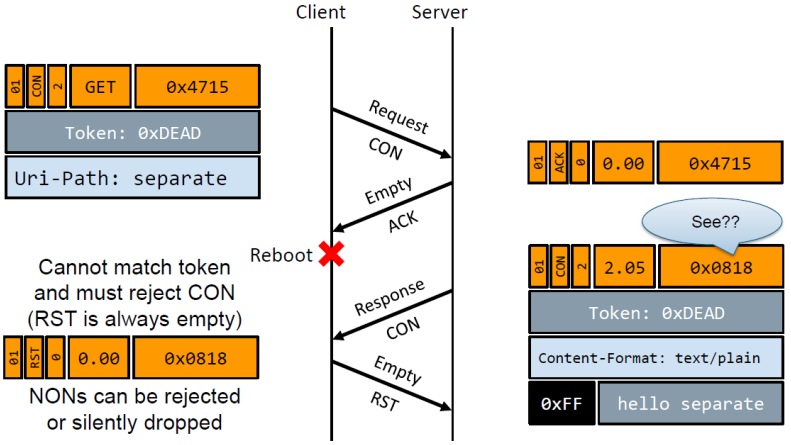
- End-to-end security
- Based on DTLS (TLS for datagrams)
- Focus on ECC
- Pre-shared secrets, certs, or raw public keys
- Hardware acceleration in System-on-a-chip

Reliability

Confirmable (CON)
reliable message with retransmission (randomised timeout of 2-3s, binary exponential back-off, stops after ACK or 4 **re**transmissions)

Non-confirmable (NON)
best-effort transmission

Acknowledgment (ACK)



confirms CON message

Reset (RST)
used when ON (or NON) can not be processed

ACK is matched to CON through MID
Response is matched to request through token

Semantic Technologies

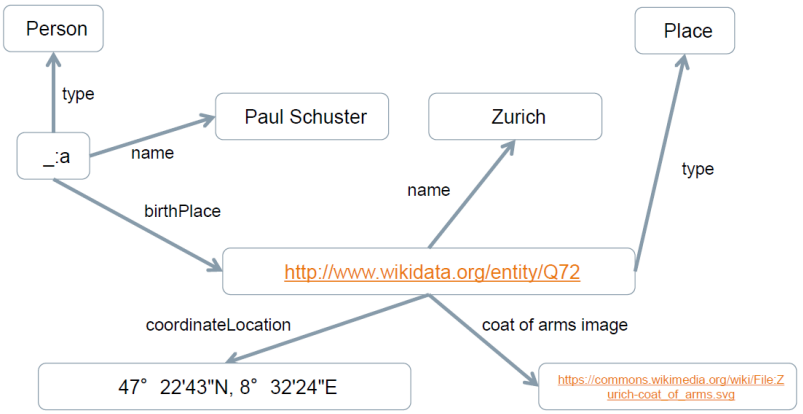
Semantic technology
A software technology that allows the meaning of, and association between, information to be known and processed at execution time

Semantic information model
Formal representations of concepts and their associations to convey conceptual meaning

About giving machines *shared understanding* of the world

- **Classes** represent hierarchical structure and content
- **Properties** represent relations
- **Instances** are non-decompositional

Semantic Web
provides common framework that allows data to be shared and reused across application, enterprise and community boundaries
→Linked data



Ideally: always have a single shared information/data model
Costs: high up-front modelling costs

Smart cards

Main use: security

- portable secure container for secret data
- secure execution environment for crypto algorithms
- authentication and authorisation

Communication

1. Cards gets Vcc and CLK, does `power-on reset`
2. Card sends `answer to reset` (ATR) (basic information about card)

3. Terminal sends first command `application protocol data unit` APDU (max 254 bytes) to card
4. Cards answers with response APDU

SIM

- Identified by unique IMSI (sent to backend system (encrypted) -> one-sided authentication)
- 22 commands
- File system (access PIN-restricted)

Wireless smartcards

Authentication

Terminal verifies cards (internetl auth)

- terminal sends random number to card to be hashed or encrypted using key
- card provides hash or cyphertext
- prevents card clones

Card verifies terminal (external auth)

- terminal asks card for challenge and sends response back to card to verify
- cryptographic challenge-response (based on shared secret)

Card holder's authentication

- terminal asks user to provide password (PIN)
- password is sent to card for verification (in encrypted form)

Location