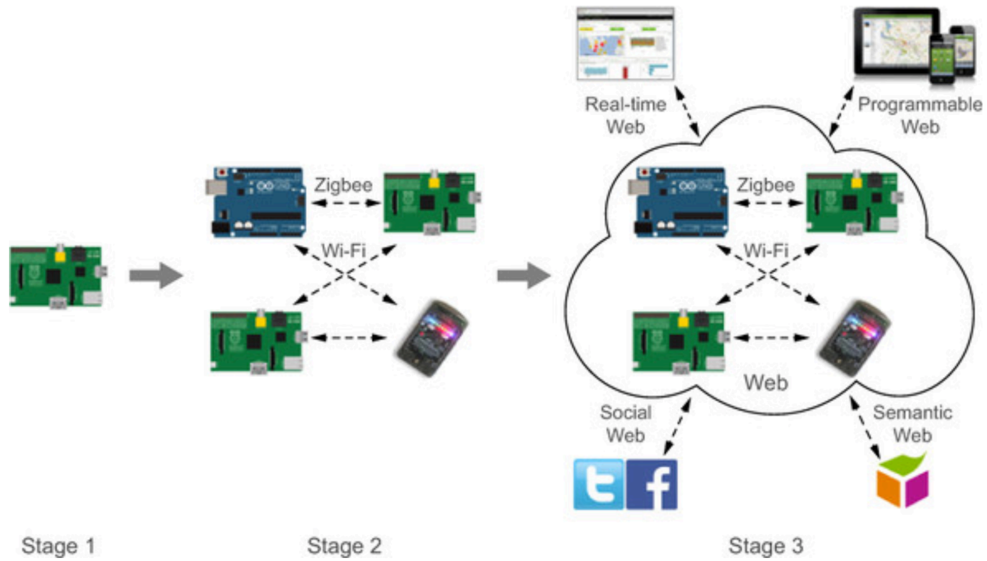
Networking for IoT

Niels Olof Bouvin

Overview

- Network communication for IoT
- IoT application layer

The more, the merrier

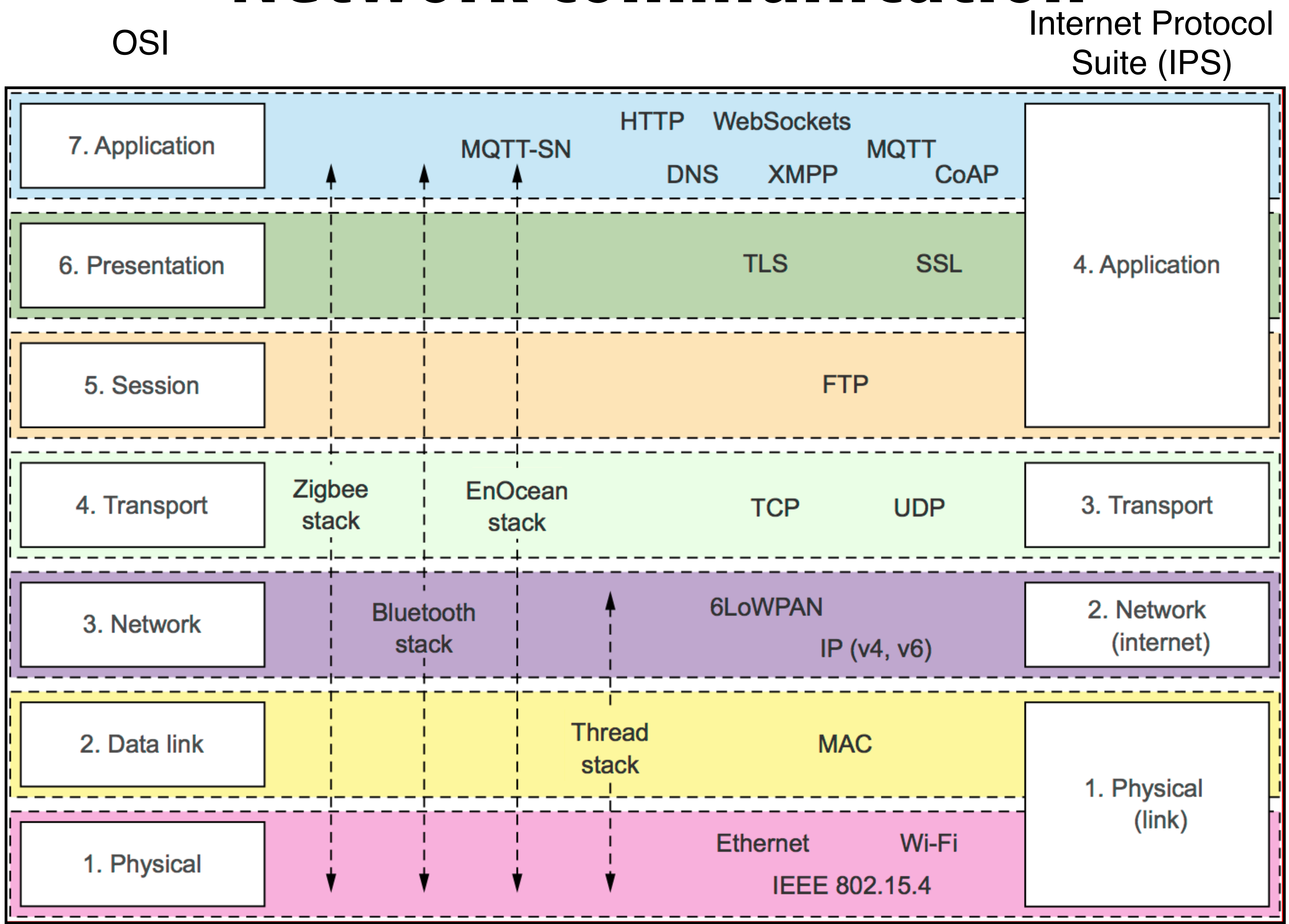


- A device may be useful on its own, but the possibilities for usefulness increase, if we add more interconnected devices or sensors, possibly even connecting across the Internet, integrating other services
- This of course requires networking

Basic networking topologies

- Point to point: often from device to app
  - fitness armband reporting to app on phone
  - app controlling device
- Star topology: a central hub
  - mobile phones communicating with cell antenna
- Star of stars topology: many hubs, one central node
  - devices talking to their respective gateways, which in turn talks to the home router
- Mesh networking: no central component
  - devices talk to each other, and handle routing themselves
  - (more about this in Mobile Ad-hoc Networks next week)

Network communication



A layered approach

- **There are multitudinous network protocols**
  - from very low level physical to high level application protocols
- **There are many different networking stacks**
  - the Internet Protocol Stack, while the most widespread, is one among many
- **Within a stack, the protocols are layered**
  - depending on the layers below
  - providing services for the layer above
- **In a well designed stack, lower layers can be replaced without affecting higher layers (if protocols are kept)**
  - your web browser does not care if you're on 4G, WiFi, or Ethernet, even though these are quite different, both physically and functionally

Wireless communication for IoT

- **Many choices and one size does not fit all**
- **Different scales:**
  - Personal
  - Domestic or building
  - City or larger
- **Different traffic scenarios**
  - Periodic sensor data
  - High, constant data rate
  - Low latency actuator activation
- **Throughput, Range, Energy-efficiency**
  - Pick two

Bluetooth®

- **Started out as a PAN (Personal Area Network)**
  - very widespread—few phones, tablets, or computers without it
  - quite a few different subprotocols—RFCOMM universally supported
  - pairing usually necessary to connect devices—more secure, but also a hassle
  - supports in principle general networking, but most cases are 1-to-1 connections
  - range up to 100 m, though typically much less
- **Recent revisions have expanded IoT aspects**
  - BT 4.0 Low Energy: sufficiently energy efficient to work in cell battery driven beacons
  - BT 5: enables trading range for speed in low energy communication
- **Not a part of the Internet Protocol Stack — must be bridged using, e.g., 6LowPAN (RFC 7668)**

ZigBee

- **IEEE 802.15.4 specified protocol**
- **Low range WPAN, simpler than Bluetooth**
- **Datarate: <250 kb/s; highly energy efficient**
- **Supports star, tree, and mesh networking**
  - usually controlled through a hub, though, e.g., a switch and light may connect directly
- **Notably used by Philips Hue and IKEA Trådfri**
  - ZigBee Light Link
  - thus, IKEA Trådfri can be paired and used with the Philip Hue hub
- **ZigBee IP links to the IP stack**

- **IEEE 802.11a-ac**
- **Ubiquitous in domestic or commercial settings**
  - speeds up to 1 Gbps
  - range typically well below 100 m with omni-directional antennas
- **Completely integrated with the IP stack**
- **Not especially energy efficient**
  - unsuited for battery powered sensors
  - 801.11ah designed to address this, but not yet here

GPRS, 3G, 4G, …

- **Mobile phone data network**
- **Works well with the IP stack**
- **Good, if not complete, coverage, but**
  - not energy efficient
  - expensive
- **Not really intended for a lot of devices**
- **5G, which is not here yet, is expected/supposed to be able to handle a massive number of devices effectively**

LPWAN

- **Low Power Wide Area Networks**
- **Typically a star topology**
  - infrastructure may be provided by the operator
- **Great range, low bandwidth, great energy efficiency**
- **Excellent for collecting data from sensors**
  - but if we need to connect to the device, we either have high latency or low energy efficiency, as the device either sleeps most of time conserving energy, or is kept awake to receive messages

- **The existing standard for SigFox communications supports up to 140 uplink messages a day, each of which can carry a payload of 12 Bytes (Excluding message header and transmission information) and up to 4 downlink messages per day, each of which can carry a payload of 8 Bytes. [Source: Wikipedia]**
- **Hardware is cheap; SigFox (and partners) operate the infrastructure financed through a subscription model**
  - Only one operator in an area; if your area is not covered: tough
  - SigFox provides various services, including geolocation and IP connectivity

LoRa

- **Competing standard to SigFox**
  - LoRa Alliance
- **Companies may run their own infrastructure**
  - Aarhus municipality uses LoRa to collect sensor data from, e.g., garbage containers
- **While most use cases are data collection, messages can be sent back, even to battery constrained units**
  - when a device uploads data, it can listen for a little while, and the central station can use that window to send a message back
  - if low latency is required, the device must necessarily be listening

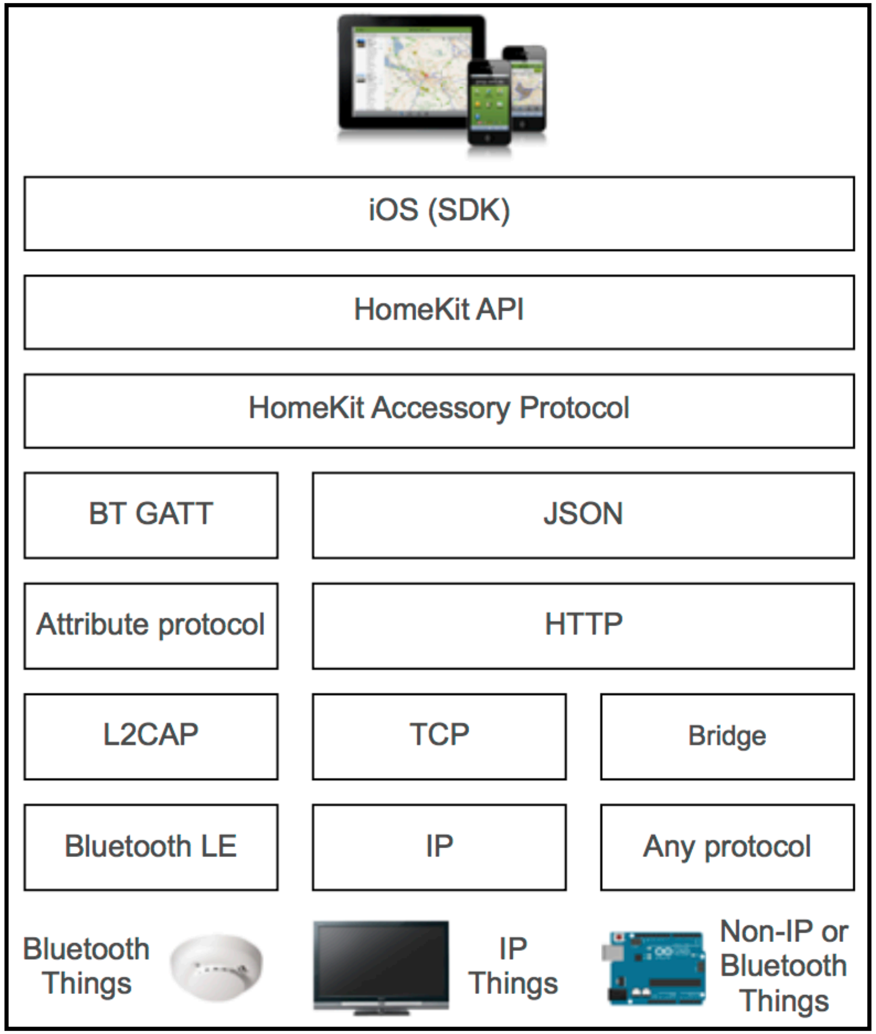
Overview

- Network communication for IoT
- **IoT application layer**

So, you want to build a system…

- **Most standards, such as Bluetooth or ZigBee have predefined profiles**
  - controlling thermostats, curtains, lights, air-conditioning, etc.
  - discovery of services
- **But, these often require specific SDKs, not necessarily widely supported across languages and platforms**

Apple HomeKit & Google Weave



Works with Apple HomeKit

- Runs on Thread
  - IEEE 802.15.4 standard like ZigBee
- Creates a mesh network
- Integrates with IP stack
- The basis for Nest
- News flash: Apple has joined the Thread Group (!)

THREAD CERTIFIED COMPONENT

MQTT & CoAP

- **MQ Telemetry Transport**
- **A lightweight publish/ subscribe protocol**
  - messages are pushed to a *broker*, who then publishes to subscribers
  - if the broker is strong, this can scale very well
- **Persistent connections**
- **QoS levels**
- **More about this later!**

- **Constrained Application Protocol**
- **UDP based and works well with 6LoWPAN**
- **REST like protocol**
- **IKEA Trådfri uses CoAP**

The WoT perspective

- **The main advantage of a Web based architecture is the wide support from frameworks to web browsers**
- **But, the Web was not designed for embedded systems**
  - basic services such as discovery and service description must be added

- **Access**
  - using RESTful API to access devices
- **Find**
  - defining semantics of devices
  - supporting indexing
- **Share**
  - secure access to devices
- **Compose**
  - combine services

Layer 1: Access

- The basic level for all other functionality
- Things are accessed through a RESTful API over HTTP, exchanging data encoded as JSON

- This is the way most services work on the Web today already

Layer 2: Find

- How can we programmatically find Things?
- How can we programmatically identify Things' capabilities?

- This requires semantic description of services that can be indexed

Layer 3: Share

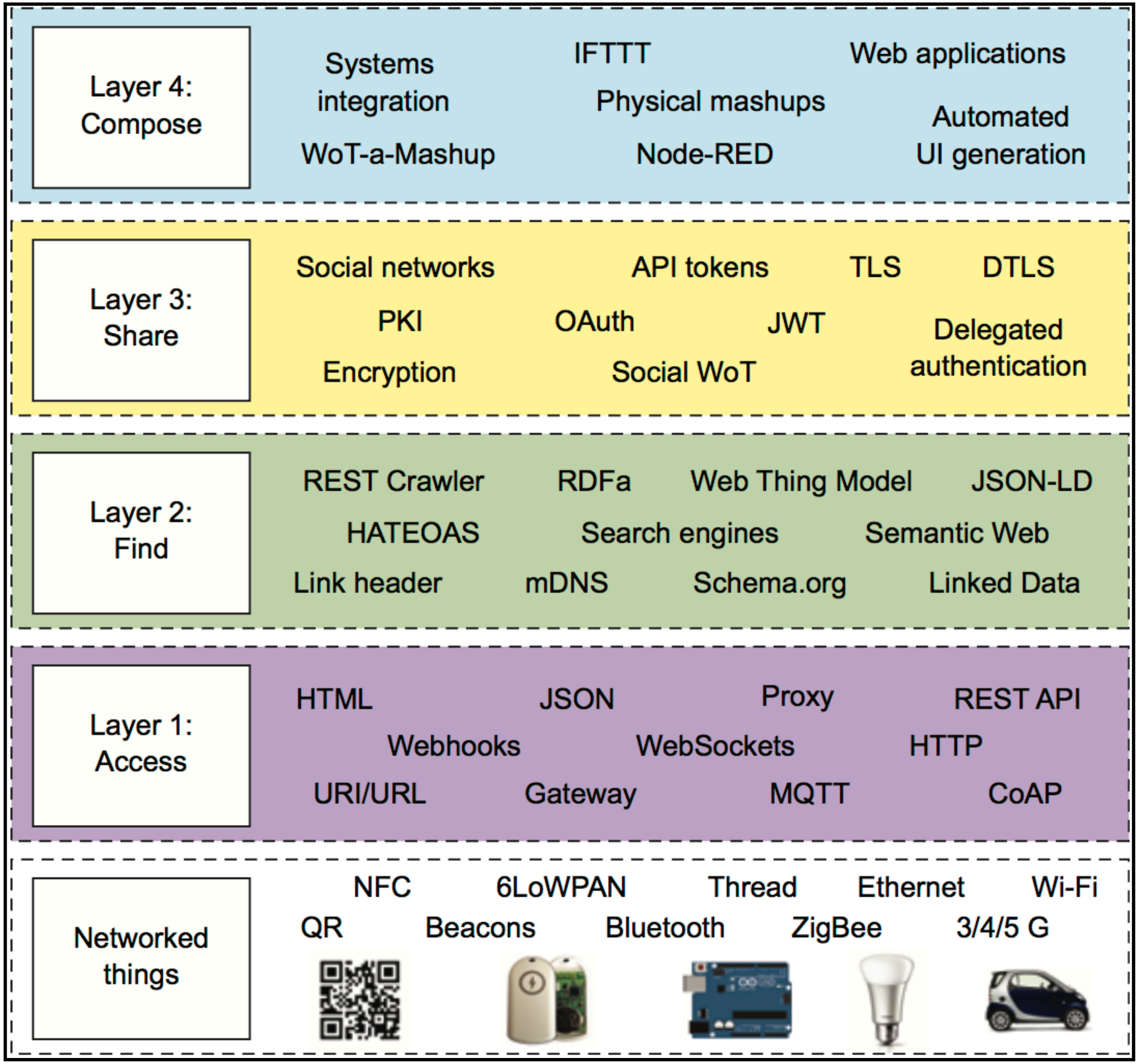
- How can we securely share and connect Things?
- Secure authentication and transmission protocols are well established on the Web, and should be used for the WoT as well

Layer 4: Compose

- Given Things with findable, well defined interfaces that can communicate with each other securely, it becomes possible to combine their services in new ways

- Combining services to offer new functionality

The WoT perspective



Summary

- **There are many protocols and technologies available for IoT devices**
  - some are well established, others are relatively new
  - some are proprietary, others are open
- **No solution fits all use cases, but openness to the IP stack enables higher interoperability**
  - and less vendor lock-in