IoT Applications

Niels Olof Bouvin

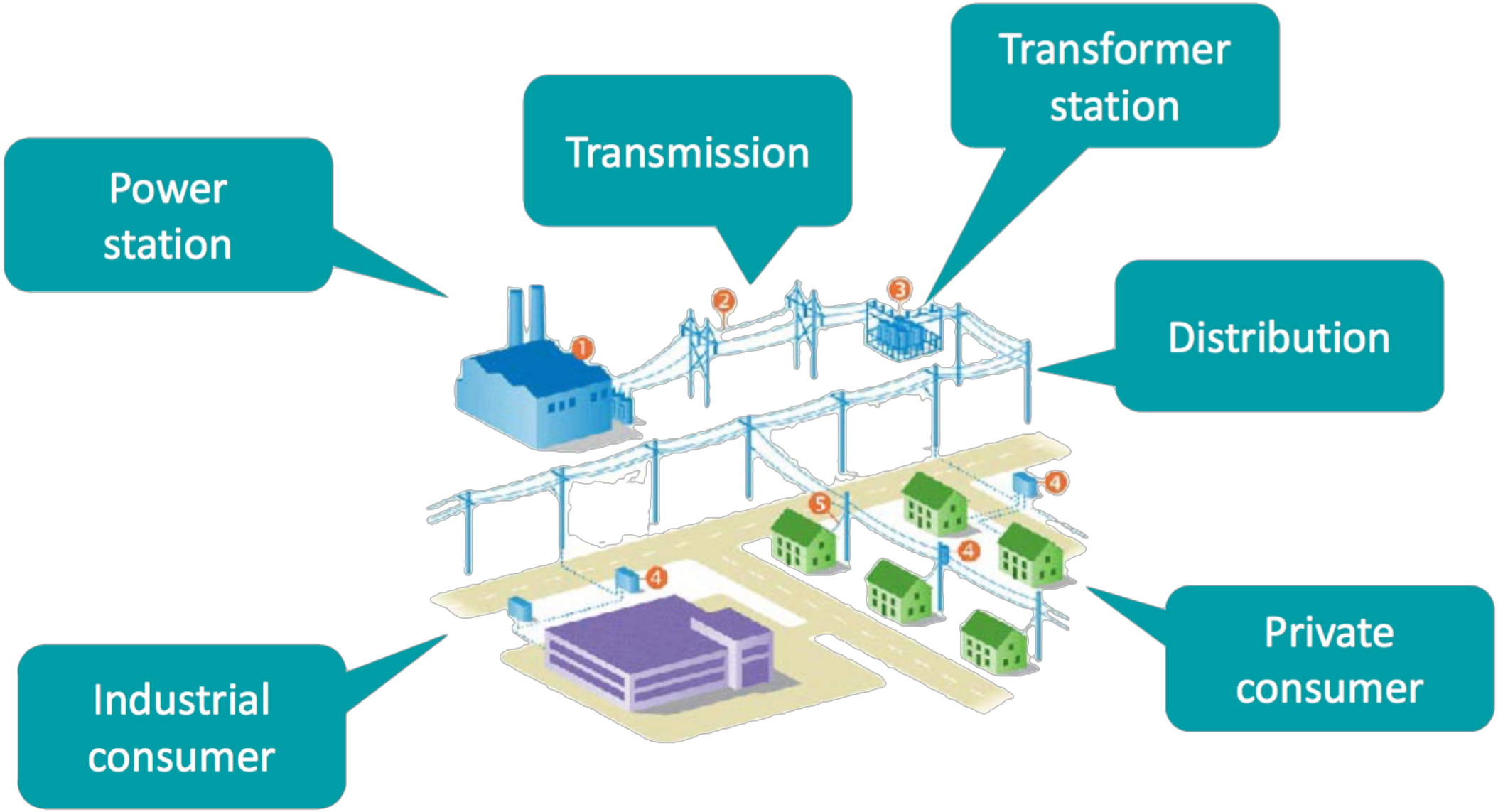
Overview

- **The Smart Grid**
- **Unifying the Internet of Things**
- **Trigger-action IoT Programming**
- **The Report**

Internet of *what* Things?

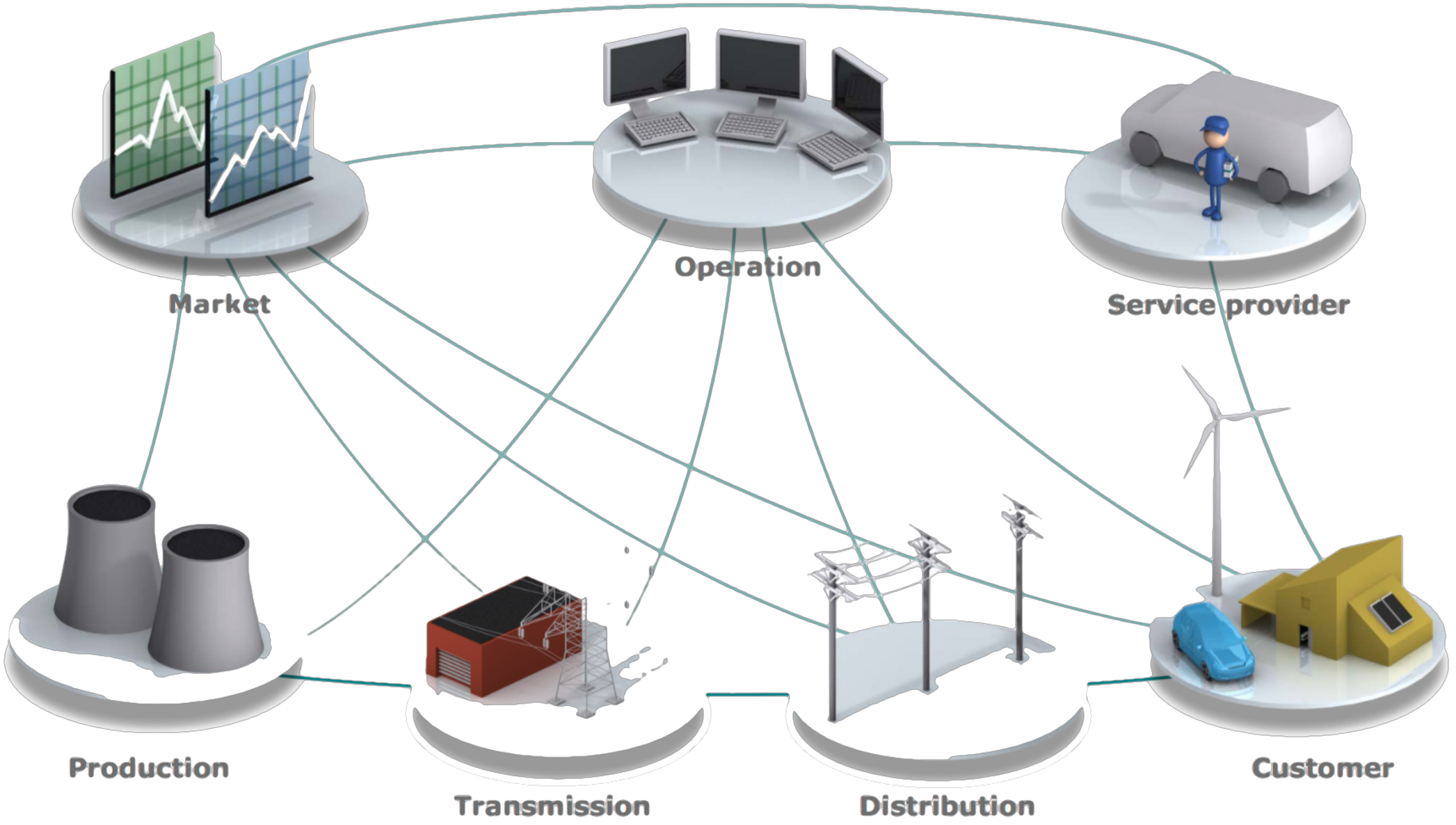
- The Internet of Things have come to cover many different areas
- Many things have been extended to become IoT or "smart" — not always well advised
- Where can the Internet of Things make a difference?

The Power Grid



- Many components; different actors and consumers

The Smart Grid



- Enabling overview and control of the entire grid

Benefits of Smart Metering

- **For the consumer**
  - better energy efficiency
  - smart appliances using power at price or demand optimal schedules
- **For the distributor**
  - better resource management through better understanding of the demand
  - better ability to cope with failures (the US have seen some cascading failures recently)
- **For the power producers**
  - better planning
  - better understanding of peak and sustained use

Existing and Intelligent Grid

Existing Grid	Intelligent Grid
Electromechanical	Digital
One-Way Communication	Two-Way Communication
Centralized Generation	Distributed Generation
Hierarchical	Network
Few Sensors	Sensors Throughout
Blind	Self-Monitoring
Manual Restoration	Self-Healing
Failures and Blackouts	Adaptive and Islanding
Manual Check/Test	Remote Check/Test
Limited Control	Pervasive Control
Few Customer Choices	Many Customer Choices

Elements of the Smart Grid

- **Power distribution becomes bidirectional**
  - e.g., using the batteries in electric vehicles as offsite storage
- **The power grid becomes interconnected across (more) national borders**
  - better use of renewable energy
- **Energy use can be directed/nudged depending on circumstances**
  - smoothening peak energy use is better use of existing infrastructure
  - e.g., staggered charging of electric vehicles, or ditto of other power hungry use cases

Challenges for the Smart Grid

- **Very large existing infrastructure ⇒ impossible to upgrade swiftly**
- **The introduction of the Smart Grid must happen gradually over the course of many years**
  - the consequences of getting it wrong would be dire
- **A good starting point could be *smart metering*, i.e., collecting information about use throughout the grid**

Connecting, collecting, and controlling

Connecting?

Wi-Next: low-power WiFi

Veniam: vehicular WiFi hotspots

Helium: metropolitan 15.4

Weighless SIG: whitespace WAN

Ell-i: intelligent PoE

Thread Group: residential 15.4

Spire: nanosatellites

Zuli: residential BLE mesh

SigFox: unlicensed cellular WAN

Connectivity for the future

- **IPv6 is a good candidate for a future proof communication between IoT devices**
  - based on open industry standards
  - many standard protocols and services
  - long history of adapting and incorporating different technologies
  - huge address space (128 bit)
  - can interoperate with IPv4

From IPv6 to smart meter

- **IPv6**
  - addressing
  - address auto-configuration
  - RPL routing, Multicast, QoS
- **6LowPan**
  - compression and fragmentation
- **IEEE 802.15.4**
  - MAC
- **RF channel**
  - radio communication to the meters

RPL routing between meters and concentrator

- **Standardised by IETF (RFC 6550)**
  - for Routing Over Low power and Lossy network (ROLL)
- **RPL "routes-over" IPv6**
  - routing metrics include link qualities, latency, energy, and node state
- **Various traffic flows**
  - multi-point to point (upwards routing), point to multi-point (downwards routing), point to point
- **Upwards routing**
  - elect best parent based on objective function
- **Downwards routing**
  - source routed from root in non-storing mode

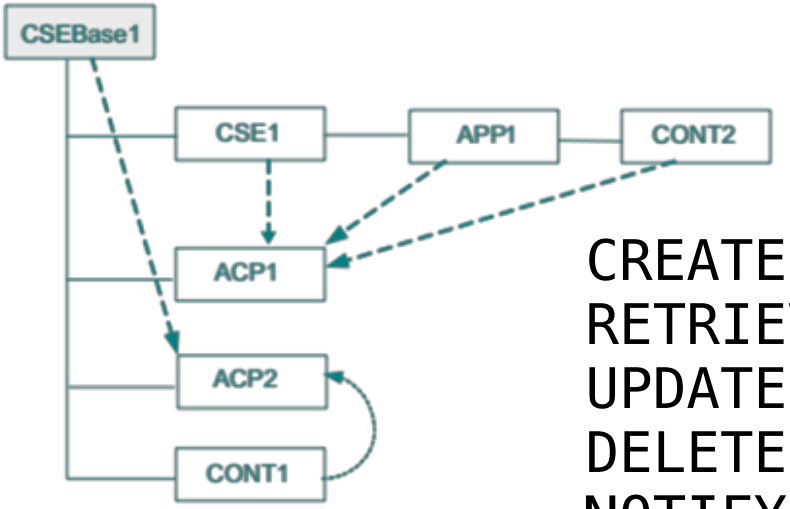
Relevant services (a selection)

- **Constrained Application Protocol (CoAP) (RFC 7252)**
  - HTTP for embedded devices
  - RESTful protocol design
  - Low overhead and parsing complexity URI and content-type support
- **Network Time Protocol (NTP)**
  - for clock synchronisation between nodes
- **Simple Network Management Protocol (SNMP)**
  - for managing devices in a network
- **DLMS/COSEM**
  - electricity meter data exchange and modeling

OneM2M

- **The vast majority of IoT communication is expected to be machine-to-machine (M2M)**
  - there is a **ridiculous** number of different ways to do this depending on sector
  - . . . and that is ok, because different fields have different needs
- **OneM2M aims to create standards for *interfacing* between heterogeneous systems**
  - i.e., not creating a whole new standard top to bottom to replace everything
- **If no such standard is established, IoT is going to be a up-hill struggle**
- **Membership includes a slew of international standards organisations, and hundreds of companies**

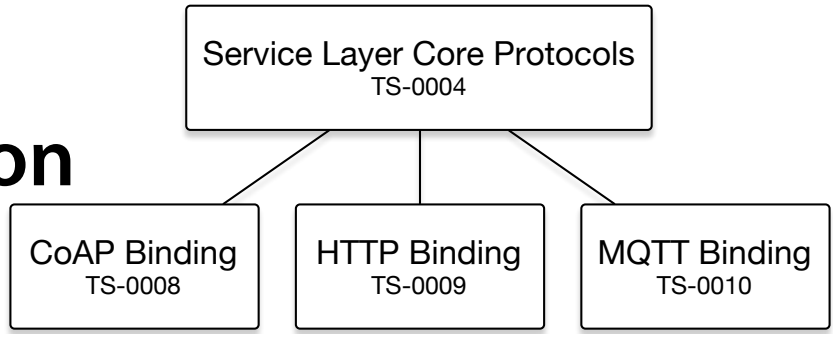
OneM2M basics



- **URI based naming**
  - works with IPv4 and IPv6
- **RESTful approach**
  - A given Resource can be identified with a Uniform Resource Identifier
  - A given Resource is of one of the defined Resource Types
  - The Resource Type determines the semantics of the information in the Resource
  - Resources can be Created, Read, Updated or Deleted to manipulate the information
  - Resources are organised in a tree-like structure and connected by links
  - Links either as the tree hierarchy or to another part or the tree

OneM2M communication

- Uses existing protocols:
- XML or JSON content serialisation
- Uses existing security protocols
  - TLS/DTLS for communication, PSK/PKI/MAF for credentials and authentication
- HTTP example:



REQUEST

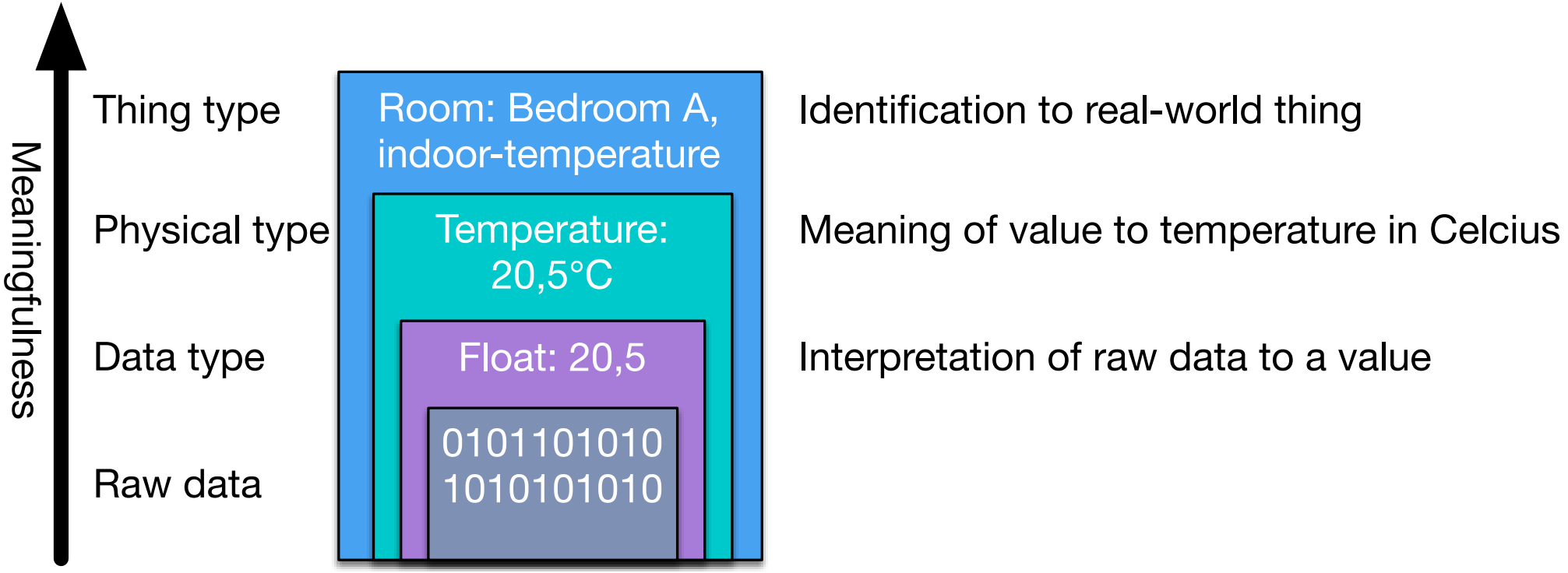
```
GET http://provider.net/home/temperature HTTP/
1.1
Host: provider.net
From: //provider.net/CSE-1234/WeatherApp42
X-M2M-RI: 56398096
Accept: application/onem2m-resource+json
```

RESPONSE

```
HTTP/1.1 200 OK
X-M2M-RI: 56398096
Content-Type: application/onem2m-resource+json
Content-Length: 107
{"typeOfContent":"application/json",
"encoding":1,
"content": “{'timestamp':1413405177000,'value':25.32}"}
```

Semantics and interoperability

- **Communication and data exchange is the basis**
  - interoperability requires parsing (syntax) and understanding (semantics)



- **oneM2M currently uses semantic annotations through ontology references**

Standardisation is hard work

- But what are the alternatives?
- If general standards (and methods of standardisation) are not established, sectors and vendors will make their own
- Vendor lock-in is dangerous/costly for any industry
- By focusing on making established systems interoperate, OneM2M would seem to be on the right course

Summary

- **The Smart Grid is the one of the Big Things of IoT**
- **As a field, it requires stability and security over decades**
  - long term planning and investments
  - incremental/evolutionary change rather than revolutionary change
  - “The S in IoT is short for Security”
- **Solid industry standards are required (hopefully!)**
  - interoperability a must
  - security essential—should be using established best practices

Overview

- The Smart Grid
- **Unifying the Internet of Things**
- **Trigger-action IoT Programming**
- **The Report**

Some absurd & some hopeful directions

- **Industry IoT is one thing**
  - established (often domain-specific) standards
  - large pre-existing investments in equipment
  - many aspects highly regulated
- **IoT for the home something else**
  - equipment turnover much faster
  - novelty an attraction in itself
  - investments much smaller
  - this has led to a number of "smart things" characterised mainly by having an associated app on a phone

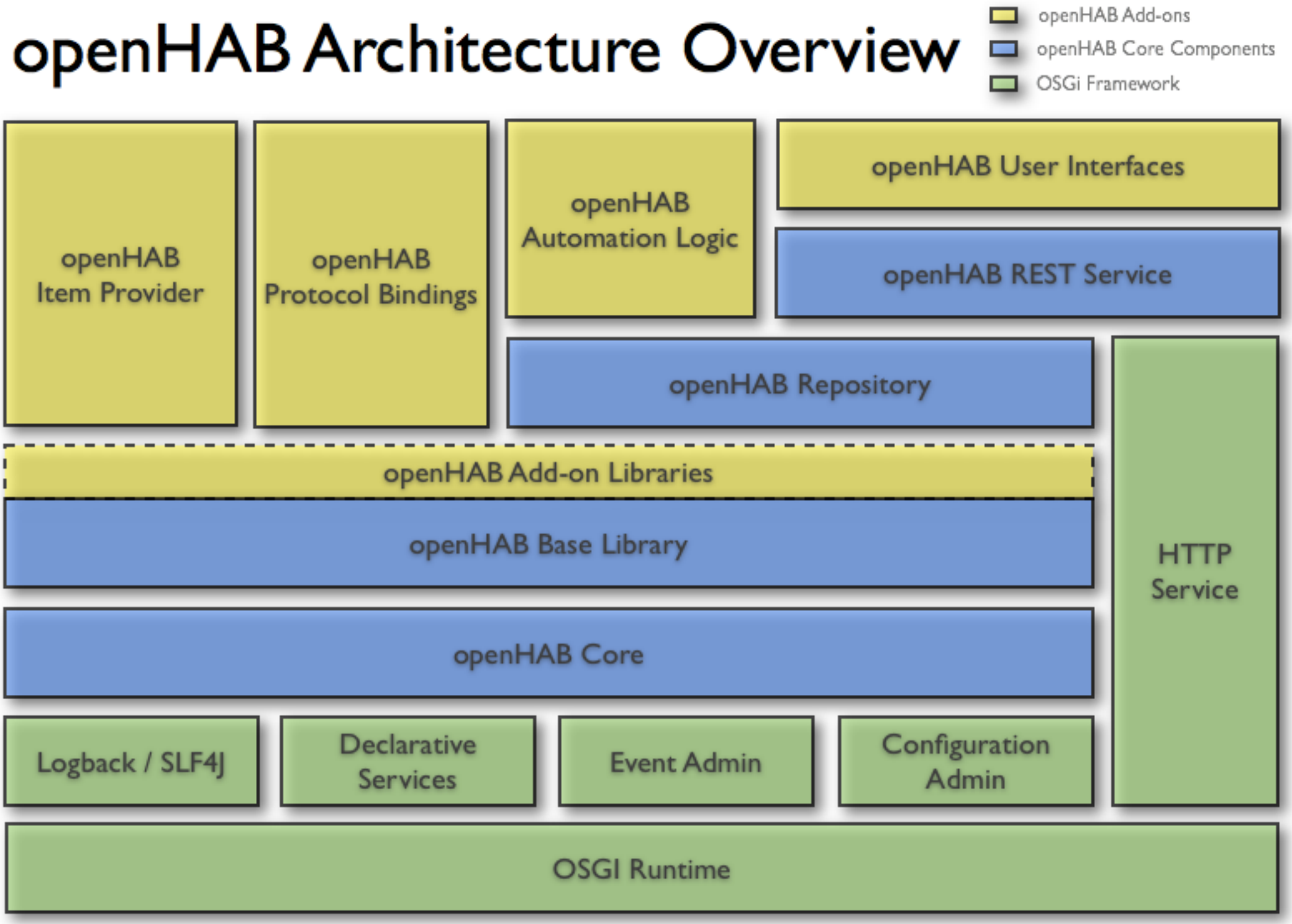
IoT open (source) directions

- **The industry giants are battling it out for control of your home and living room**
- **As no single victor seems likely at this point, the end result is either**
  - homes that are vendor-specific ("*this* is a *Google*-home!")
  - homes that are balkanised into islands of technology, each with their app and infrastructure
  - what happens if your choice of vendor goes out of business?
  - how can data security and privacy be ensured across many different vendors?
- **Surely, we can do better?**
- **Are there any alternatives to the Web of Things?**

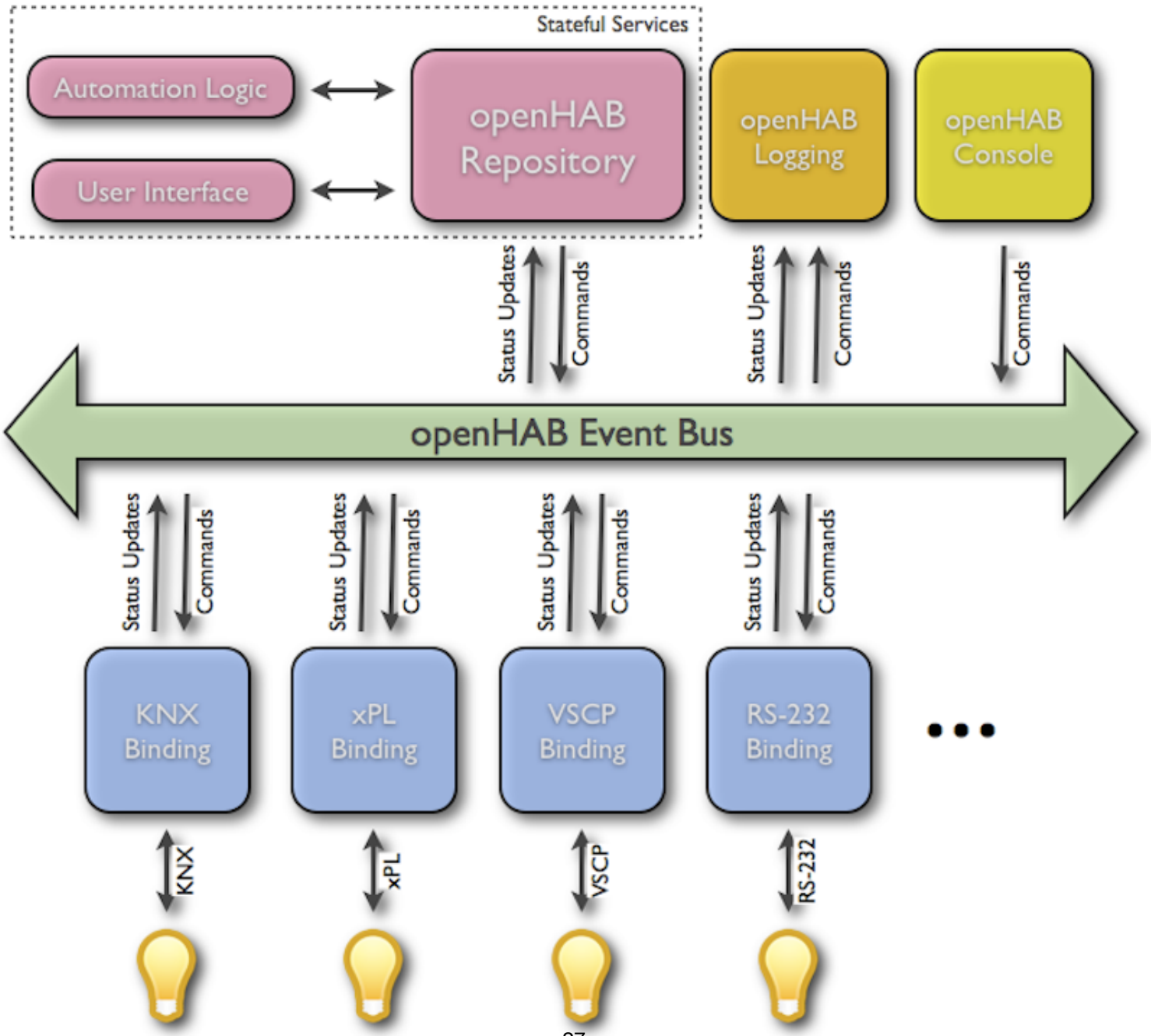
openHAB

- **http://www.openhab.org/**
  - Integrates (a lot of) existing smart home technologies
  - Vendor, network, and platform agnostic
- **Java-based, open source**
- **Used to create rules and scripts that enable seamless integration between different systems**

openHAB Architecture Overview



Events in openHAB



Hardware and software platforms

- Particle.io Photon
- Raspberry Pi 0-3
- ESP8266
- Arduino (many variants)
- …

- Eclipse IoT initiative
  - OS MQTT, CoAP, LWM2M, OneM2m
- Node-RED
- The Thing System
- Souliss
- …

Convergence — one way or another

- The benefits of IoT are found with devices and humans working in concert
- Balkanisation works directly counter to this
- At least, grass roots and (smaller) businesses can exist to address this
  - though this is sadly potentially fragile

Overview

- The Smart Grid
- Unifying the Internet of Things
- **Trigger-action IoT Programming**
- **The Report**

End-user programming for the home

- **If the Internet of Thing is to be realised (beyond what is already the case), users will need a way to control the devices in their homes, on their persons, in their cars, as well as their internet services**
  - simple control (directly or through a UI) is relatively straightforward, especially if a unified approach (such as WoT) can be realised
  - but what about more complex interactions?
- **Home owners are not programmers, nor should they have to be in order to be successful operators of their new devices or services**

Trigger-action programming

- **Satisfaction of a condition (the "trigger") results in the immediate execution of an action**
  - If *this* then *that*
- **In its simplest form, one condition and one action per statement**
  - no boolean logic, no compound statements, no delayed actions
- **Conditions based on the state of supported entities**
- **Actions limited to manipulating these entities**
  - so if something is not supported, it is not going to happen

If This Then That

https://ifttt.com

- **Commercial web site/app that binds many internet services together**
  - "send me an email, when <this string> appears in RSS/Twitter/etc"
  - "automatically backup my forum posts to my Evernote account"
  - "automatically save mail attachment to Dropbox"
  - …
- **Now also integrates many IoT devices**
  - B&O, BMW, D-Link, Honeywell, LaMetric, LG, Nest, Philips Hue, Samsung, WeMo, …

Can users *do* this?

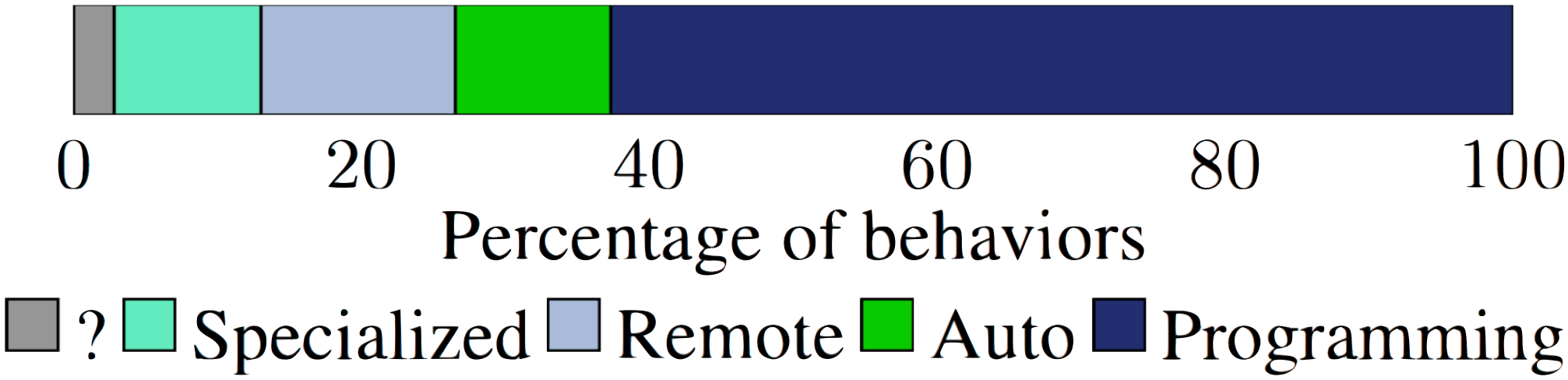
- **Two part study, published two years apart**
  - are trigger-action programming a good match for IoT in the home?
  - is the simple one condition, one action sufficient?
  - how are users doing this in the wild?

Specifying behaviour of smart devices

- **What would end-users have smart homes do?**
  - Mechanical Turk workers were asked open-ended questions on five things they would like a smart home to do
  - Half were given examples of trigger-action scenarios; the rest were not instructed

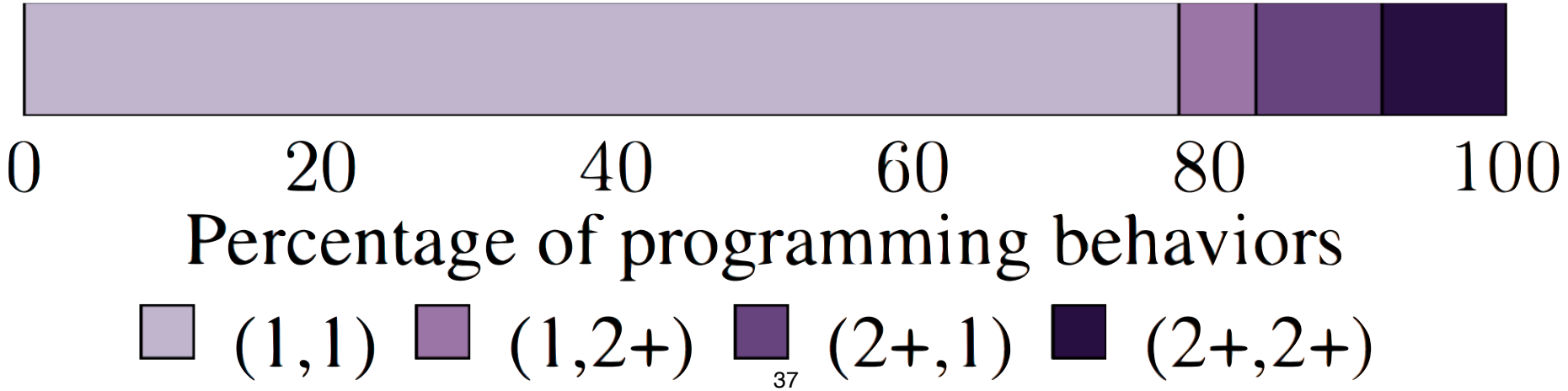
Results

- **The responses were coded into four categories**
  - programming (68,9% from those provided with examples and 51% from those without)
  - automatic self-regulation
  - remote control
  - specialised functionality



Programming?

- **Examples:**
  - *"I want the fan in my room to turn on when it is hot."*
  - *"Notify me if my pet gets out of the backyard."*
  - *"Start brewing coffee 15 minutes before my alarm."*
  - *"Lights...dim according to the level of outside light."*
  - *"I would like my home to automatically clean the floors on a daily basis while no one is in the room."*
- **All could be formulated as trigger-action statements**



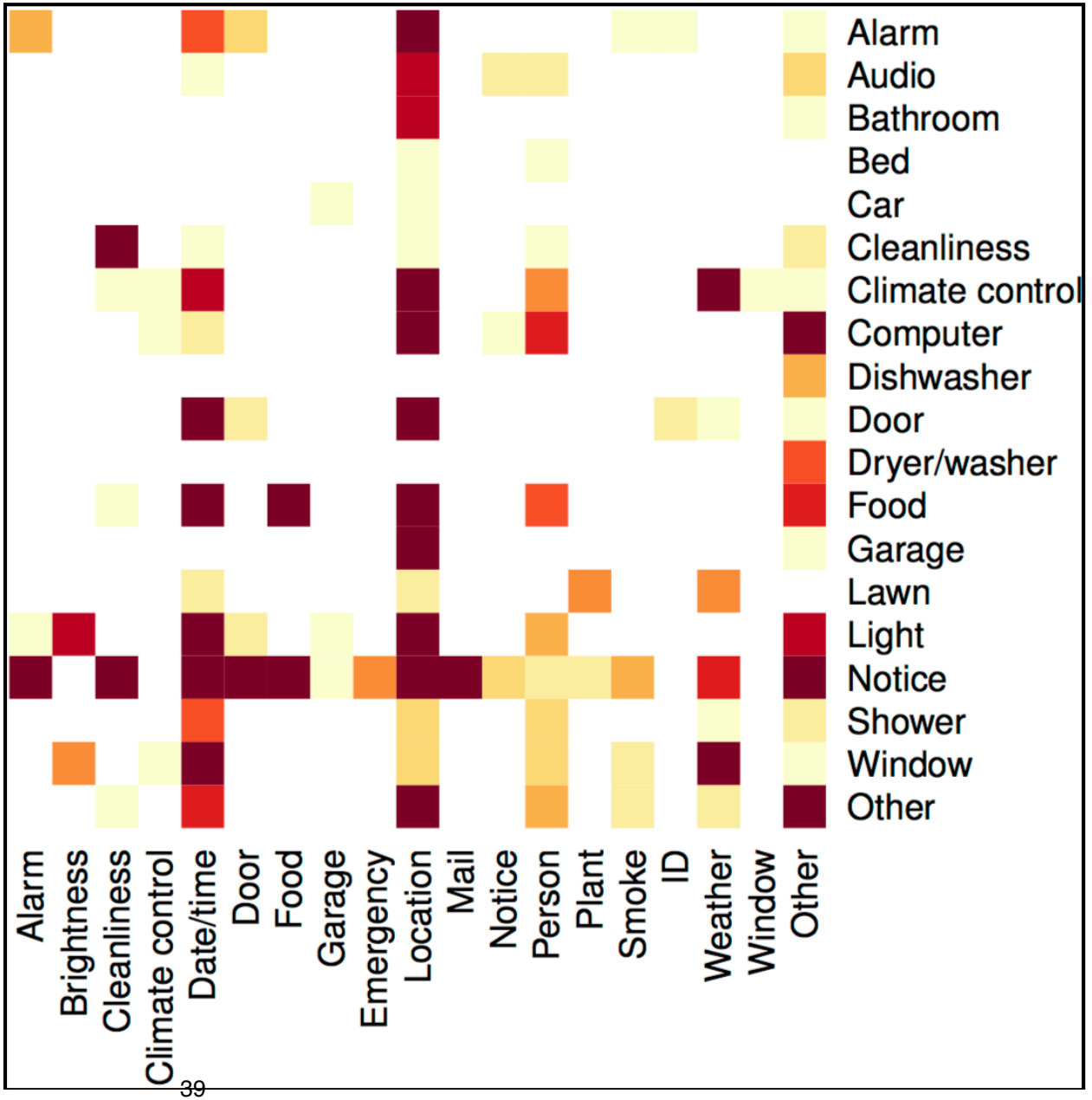
Levels of abstractions

- **The respondents' concepts of triggers varied, though none mentioned sensors per se**
  - direct sensing: "when the door bell rings"
  - more abstract: "when no one is in the room"
  - fuzzy: "when my cat meows"

- **Some of these are straightforward, others a little more involved, and some may only be possible with new sensors, rich data sets, and/or machine learning**

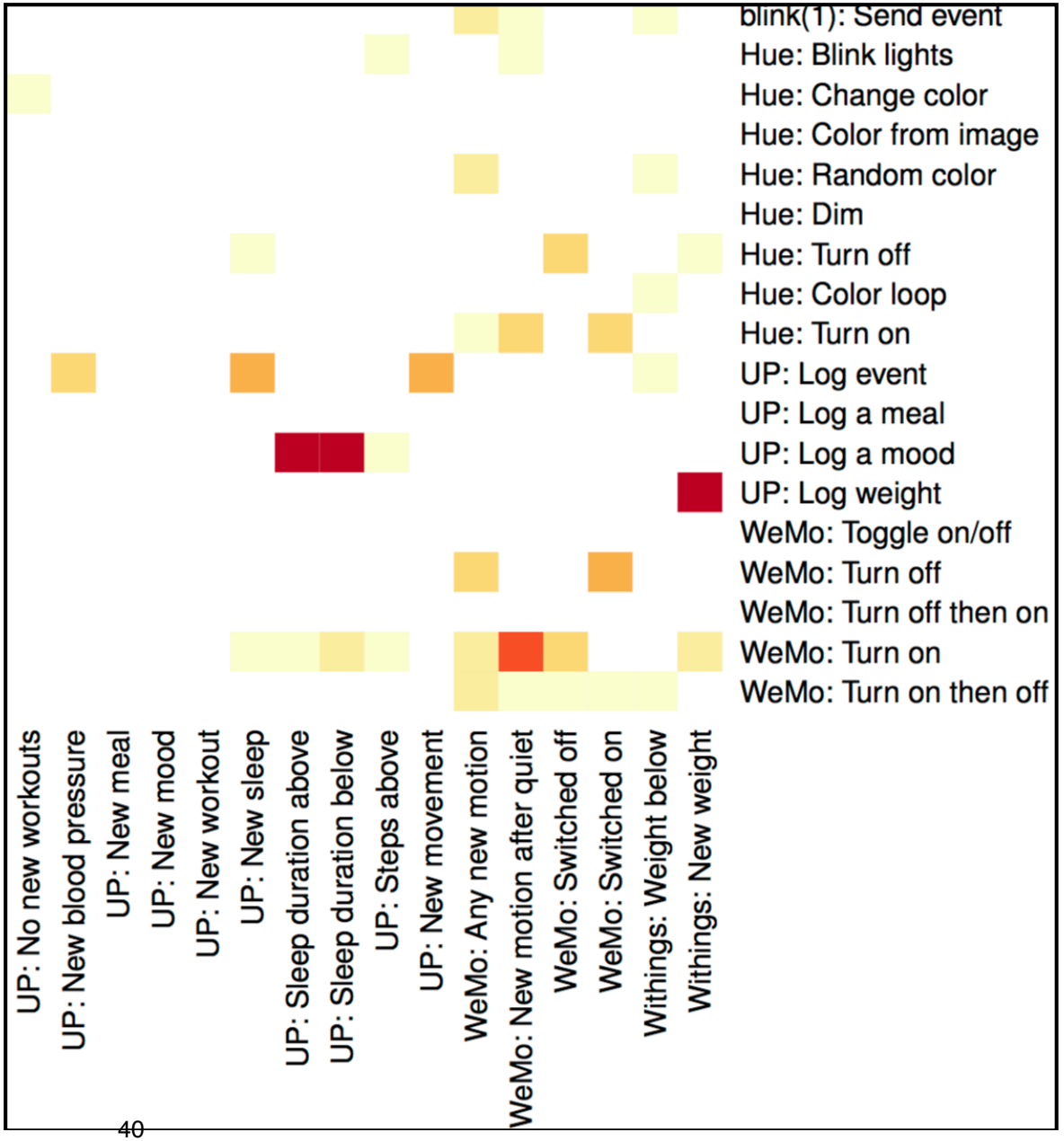
Grouping the desired trigger-actions

- The researchers coded the rules into trigger-actions and grouped the triggers and the actions
- Triggers: x-axis
- Actions: y-axis



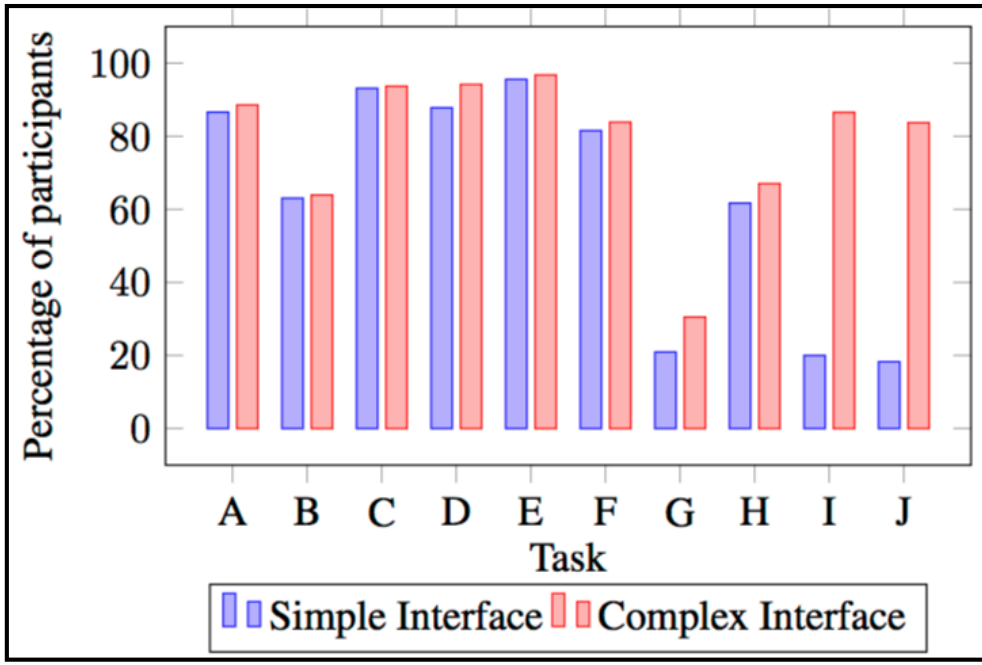
What do people do in the Wild?

- 67169 public trigger-actions ("recipes") scraped from IFTTT's website
- They limited their study to six physical devices (in 2013)
- 92 recipes with physical trigger and actions depicted here



Can end-users program?

- Respondents were given 10 tasks that they should express in either a system (1 trigger, 1 action), or a *complex* system (multiple triggers and actions)
  - they were able to complete most of the tasks
  - they became better using the tools with time



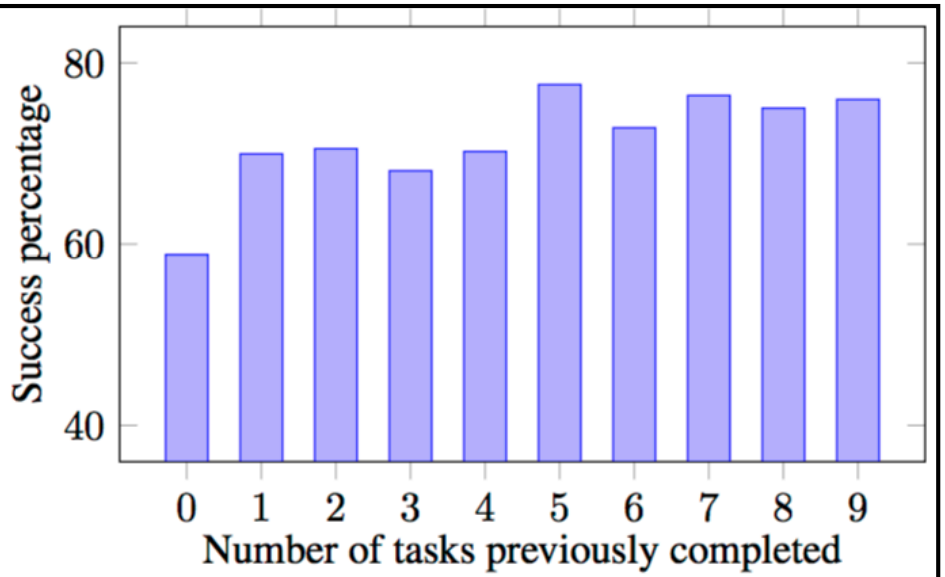


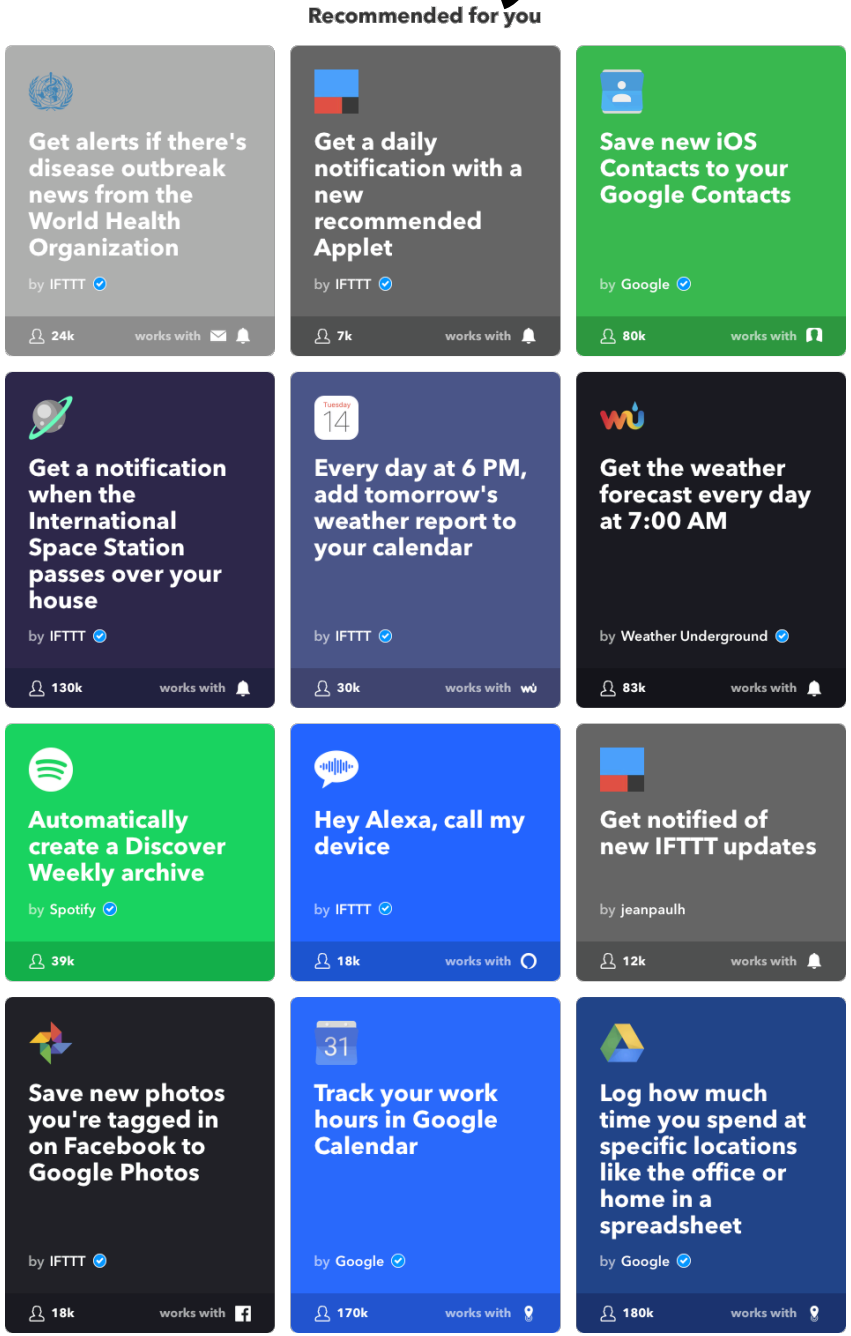
Figure 9. The percentage of participants who successfully complet a task versus the number of tasks they had previously completed. W

Results

- Trigger-action programming seems to be a good fit for many smart home oriented activities
- End-users can express tasks in such systems
- One trigger/one action may be too limited
- Some triggers will be challenging to capture

IFTTT: If This Then That (today)

- **Have moved beyond 1:1 rules**
  - 'applets' with conditions and multiple actions
  - now integrated with iOS/Android app
- **Integrates 100s of services**
  - https://ifttt.com/search/services
  - all sorts of Internet services and smart devices
- **$199-499+ to be a partner**
  - free to be a *Maker*, creator of 'applets' (JS API)
  - complex rules possible



What characterises the recipe makers?

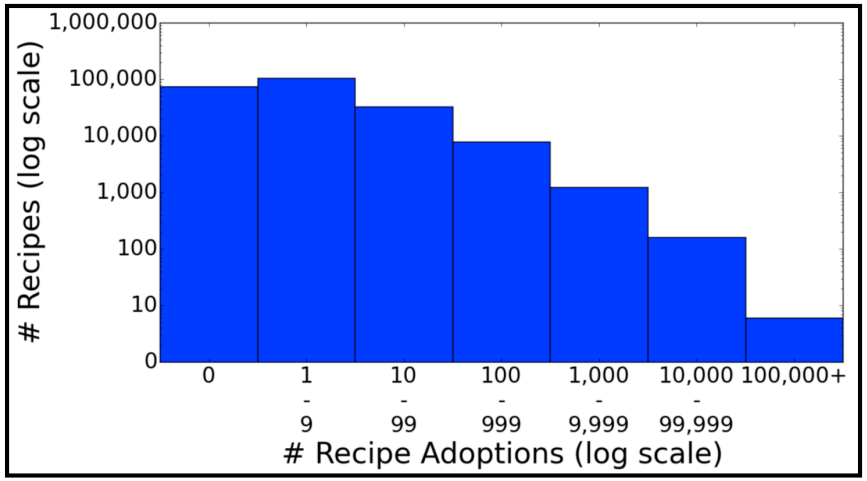
- Second paper: two years later, a scrape and analysis of 224590 recipes (as they were still known then)

Table 1. Key characteristics of the 2015 and 2013 IFTTT datasets.

Characteristic	2015	2013
# of trigger channels	177	52
# of triggers	768	180
# of action channels	143	45
# of actions	368	106
# of recipes	224,590	67,820
# of authors	106,452	35,495
# of adoptions (across recipes)	11,718,336	1,293,639
Mean # of adoptions per recipe	52.2	19.1
Median # of adoptions per recipe	1	1

Adoption/popularity of recipes

- The vast majority of recipes are used by a very few
- A few recipes are used by a lot of users
- A few authors are very prolific and popular



- an author with **N** h-index has shared **N** recipes, each of which has been adopted by at least **N** users

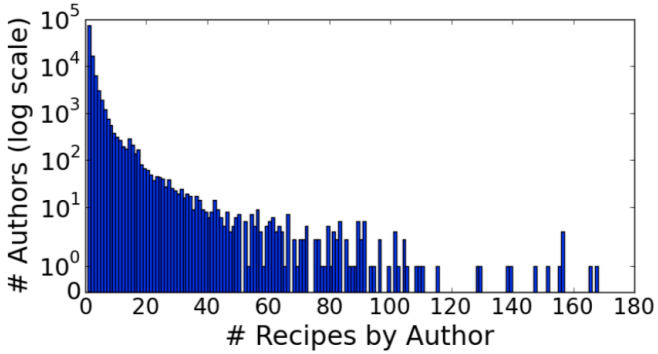


Figure 2. The number of authors who created a given number of recipes, with outliers (the handful of extremely prolific authors) removed.

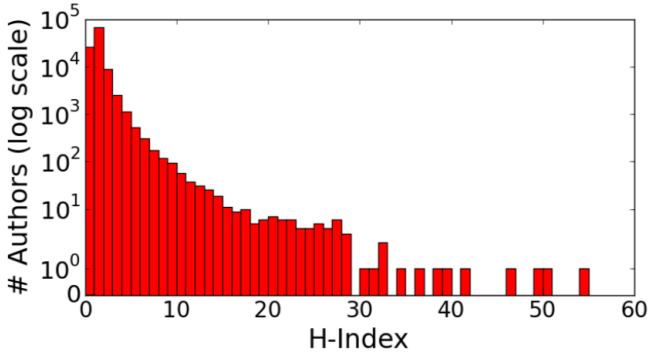
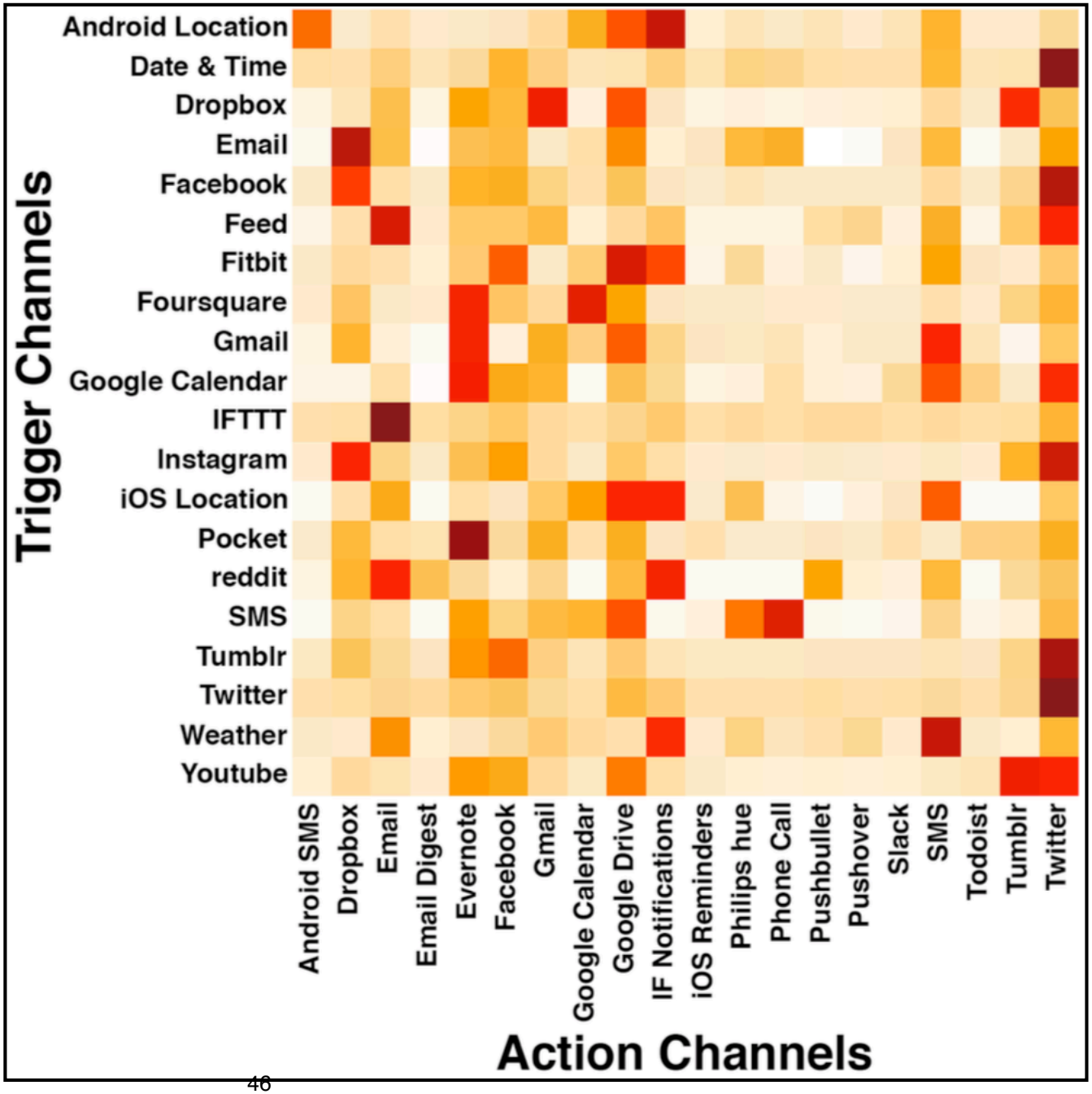


Figure 3. Author h-indices. We exclude an outlier with h-index 117.

Trigger-action channel connections

- The most connected channels
- Though some are more popular than other, the spread is wide



Results

- Trigger-action programming *really* seems to be a viable (and growing) approach to end-user programming of IoT devices
- Balancing the simplicity of trigger-action with more advanced demands (such as a device's history, or triggers that adapt) are unsolved problems
- Getting the end-user engaged is crucial for success

Summary

- **The Internet of Thing is already happening**
  - in the very large and in the small
- **Planning and control on the large scale**
- **Control and convenience on the small scale**

Overview

- The Smart Grid
- Unifying the Internet of Things
- Trigger-action IoT Programming
- **The Report**

Report writing

- A crucial part of your project is writing the report about what you have done
- It is *really* important for the exam, so you should work on it in parallel with your other project work

The report template

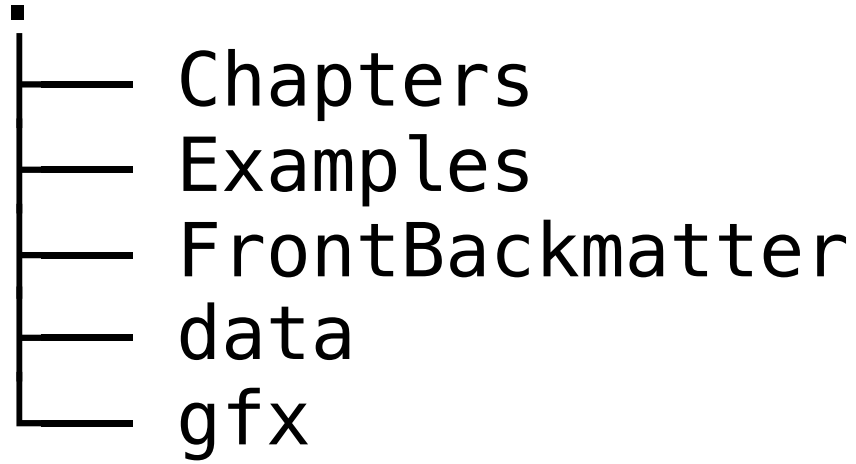
- It's *my* job to read your reports, so *I* get to decide, how they are structured and what they look like!
- A LaTeX template based on the work by André Miede, who in turn was inspired by the excellent *"The Elements of Typographic Style"* by Robert Bringhurst
- I have used it for a couple of years with my master's thesis students, but it works just as well for reports
  - the structure is exactly the same, even if the scope is smaller

Getting the template

```
git clone https://users-cs.au.dk/bouvin/git/thesis.git
```

- As well as a link to a PDF and zip-file on the frontpage
- The zip-file can be uploaded to ShareLaTeX as a project (and it probably works with Overleaf, too)

Overall structure of files and folders



- **`./ClassicThesis.tex`**
  - ties everything together, inputs all chapters and sections
- **`./classicthesis-config.tex`**
  - used for configuration, such as title and authors' names
- **`Chapters/, data/, gfx/`**
  - used to contain chapters, data collected, and figures for the report

The overall structure of the document

- **Part I**
  - the proper structure of a thesis (written by yours truly)
- **Part II & Part III**
  - formatting instructions and suggestions (by André Miede)

The proper structure of a report

- Introduction
- Related work
- Analysis
- Design
- Implementation
- Evaluation
- Conclusion

Introduction

- What is it? Why is it interesting?
- What is the use case for it?
- What are the central questions/hypotheses about it?
- How can those questions be tested?

Related work

- Scientific and technical works that are *relevant* and *contribute* to the analysis of the problem domain
- Identify central aspects in the literature
- Use those aspects to systematise your write-up of each article or system
- Discuss *only* the work of others—your own work comes later!