T2TRG: Thing-to-Thing Research Group

IETF 103, November 6, 2018, Bangkok, TH

Chairs: Carsten Bormann & Ari Keränen

Note Well

- You may be recorded
- The IPR guidelines of the IETF apply:
  see **http://irtf.org/ipr** for details.

Administrivia (I)

- Pink Sheet
- Note-Takers
- Off-site (Jabber, Hangout?)
  - **xmpp:t2trg@jabber.ietf.org?join**
- Mailing List: **t2trg@irtf.org** — subscribe at: **https://www.ietf.org/mailman/listinfo/t2trg**
- Repo: **https://github.com/t2trg/2018-ietf103**

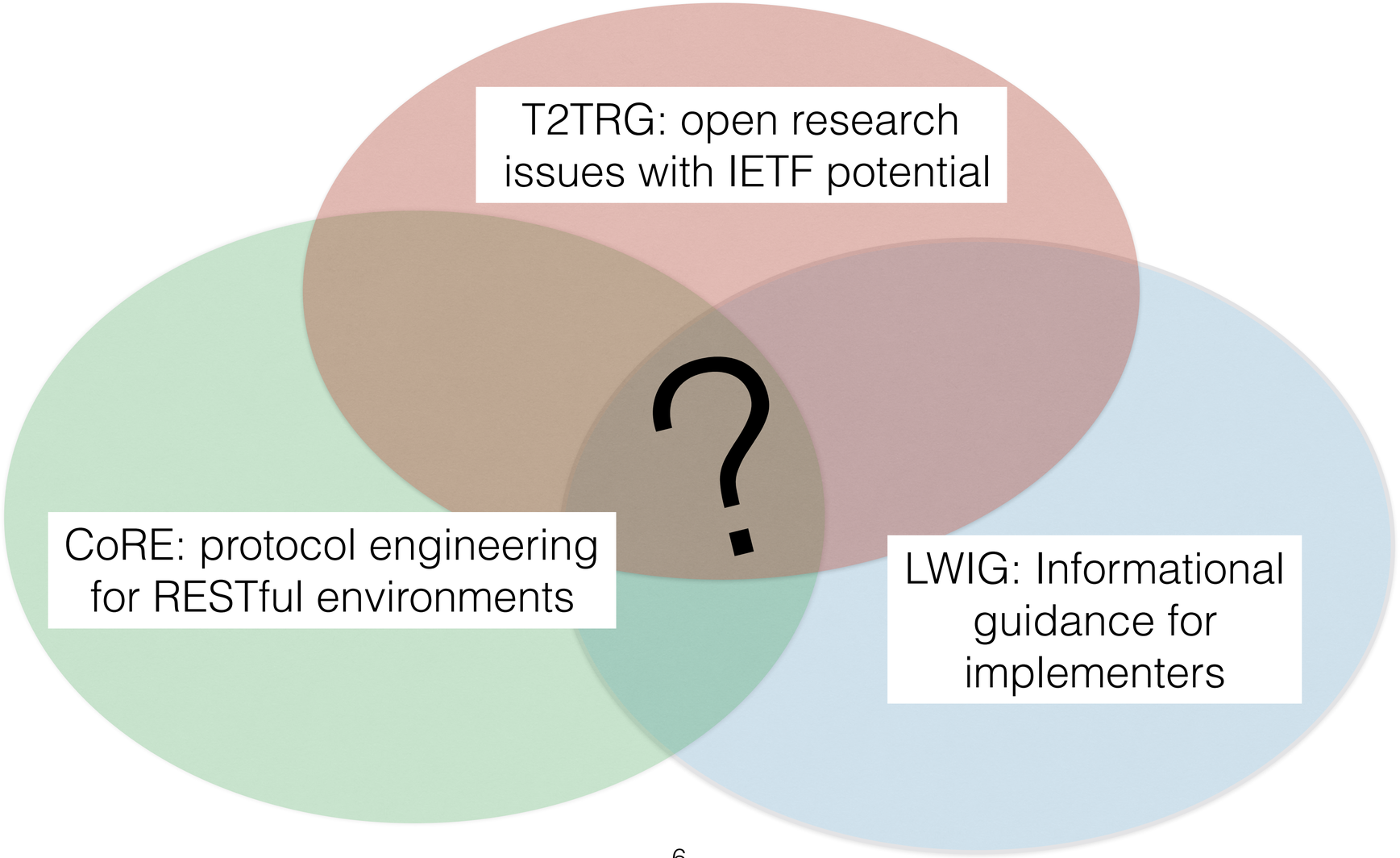
Agenda

Time	Who	Subject	Docs
16:10	Chairs	Intro, RG Status	draft-irtf-t2trg-iot-seccons draft-irtf-t2trg-rest-iot
16:20	Chairs, various	Report from WISHI and Hackathon	
16:45	Michael Koster	brief iot.schema.org update	
17:00	Matthias Kovatsch	W3C WoT update	
17:15	Chairs, various	core-apps, CoRAL — division of work	
17:45	Chairs, various	Intro to Friday's work meeting	
18:00	Chairs	Meeting Planning, Wrapup	
18:10		end of meeting	

T2TRG scope & goals

- Open research issues in turning a true "Internet of Things" into reality
  - Internet where low-resource nodes ("things", "constrained nodes") can communicate among themselves and with the wider Internet
- Focus on issues with opportunities for IETF standardization
  - Start at the IP adaptation layer
  - End at the application layer with architectures and APIs for communicating and making data and management functions, including security

IRTF and IETF?



Recent activities

- Work on IoT/Semantic Hypermedia Interoperability (WISHI): bi/tri-weekly calls and hackathons

- Semiphysical/WebEx sessions with OCF on CoRE technologies

Next meetings

- Work meeting this Friday (with Breakouts)
- Regular WISHI calls (~ monthly)
- Virtual meetings with OCF
- Virtual meetings with OMA SpecWorks (LwM2M & IPSO)
- Prague IETF 104
  - WISHI hackathon?
- Co-locating with academic conferences 2019?

RG Doc Status

- “State-of-the-Art and Challenges for the IoT Security” ready
- “RESTful Design for IoT” (next slides)
- Upcoming:
  - Document(s) to be shaped from CoRAL and CoRE Apps?
  - Inter-network Coexistence in IoT?

RESTful Design for IoT

New in -02:

- FETCH/(i)PATCH method considerations
- Caching considerations
- CoRE Apps draft for more details on how to define IoT hypermedia apps in a structured way
- And a bunch of IoT details discussed in Montreal

RESTful Design for IoT next steps

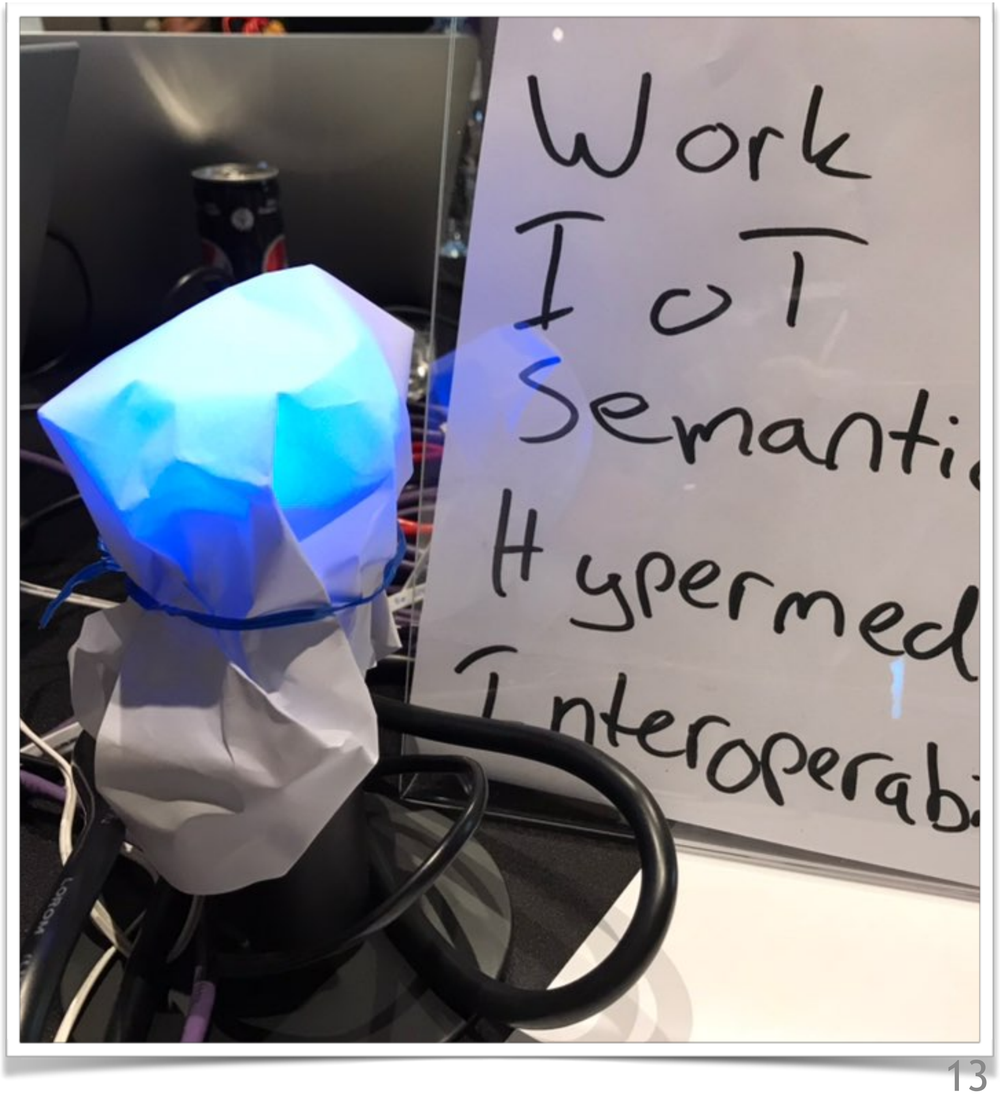
- Experiences from building IoT systems with (constrained) RESTful+ methods
  - W3C Web of Things?
  - OMA SpecWorks LwM2M?
  - OCF?
  - IoT platforms?
- More outside (of IRTF/IETF) reviews
- Ready for publication by IETF 105?

WISHI

- Four Web meetings since IETF102, discussing e.g.:
  - iot.schema.org definitions for semantic annotation
  - Semantic Style Sheets:
    adding semantics to existing instances of data
  - Declarative Data Conversion for JSON
  - LwM2M-WoT integration with iot.schema.org semantics
  - iot.schema.org with IPSO/LwM2M and OCF models
  - Notes on Semantics and Engineering principles

WISHI hackathon results

- 4th WISHI IETF Hackathon
- 8 participants
  (2 remotely)
- Connecting things from different ecosystems using shared semantics and hypermedia



What got done

- Key achievements
  - Turned a lamp on (and off) – hands off
  - Semantic interop for data and actions between LwM2M clients, Philips Hue lights, CoMI Toaster (kind of)
  - New Tiny Thing Directory implementation
  - Improved RD implementation
- Good discussions
  - Adding semantics to binary data
  - Hypermedia safety for IoT
  - Semantics and engineering principles; semantic uncertainty and usable semantics

What we learned

- Semantics is hard(er than you think)
- Setting up and testing stuff even more beforehand helps a lot – but we're getting better
- Bunch of new potential research topics for T2TRG

iot.schema.org

T2TRG Review

November 6, 2018

Overview and status

- SSN Workshop
- Charter
- Explainer and introductory slides
- Integration with schema.org
- Developer tools
- Work on modeling target ecosystems
- Work on automating consumed and exposed APIs
- Developer-user tools
- Going forward

SSN Workshop at ICSW2018

- Presented iot.schema.org at the SSN Workshop last week
- Presentation is in the teleconferences folder
- Discussion:
  - Action, Event, Property terms are badly overloaded
  - When will the definitions be available on schema.org?
  - How do we create and use definitions?
  - What tools are available for definitions and annotation
  - How do we use definitions with existing device ecosystems?

SSN Workshop (contd)

- Presentations on Automotive, Building Management, Home Care use cases
- Clear focus on Feature of Interest concepts
- Gap analysis for Semantic IoT
  - Taxonomy of Observable Properties
  - FoI Vocabularies
  - Sensor/Actuator Vocabulary
  - Vocabulary for processes and procedures

Organization

- W3C CG Charter
- Introductory materials
  - Explainer
  - Slide set for introduction
  - SSN workshop slides
- Integration with schema.org
  - May not be a sub-domain, e.g. become part of schema.org
  - We need to enable the schema browser for iotschema definitions

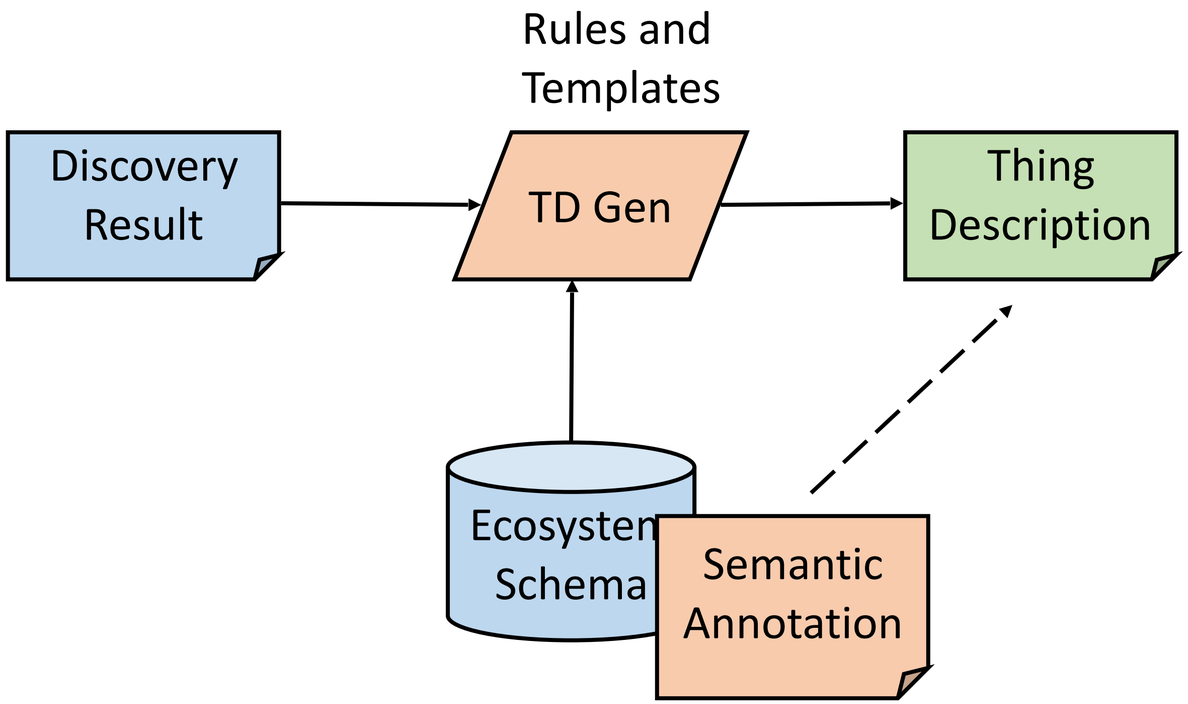
Developer tools

- How to create and maintain definitions
- How to use definitions in deployed systems
- **How to apply definitions to existing device ecosystems and FoI definitions**
  - OMA LWM2M
  - OCF
  - W3C WOT Thing Description
  - Genivi VSS
  - Haystack/Brick
  - What about Amazon Alexa, SmartThings, etc.
  - Other APIs using OAS/Swagger, HAL, JSON Hyperschema

Applying iot.schema.org definitions to existing ecosystems

- Existing definitions in some machine-readable format
  - XML, JSON-Schema, JSON, others e.g. YAML
- Annotate the definitions with Semantic terms to describe affordances
  - JSON-LD schema can be annotated as in WOT TD
  - Other annotation techniques (WISHI Research)
  - Use existing definition or create new definitions
- Generate hypermedia controls from the annotated definitions
  - TD Generator
  - Other annotations of instances

Process


Rules and
Templates
Discovery
Result
TD Gen
Thing
Description
Ecosystem
Schema
Semantic
Annotation

Annotation of a JSON Schema fragment using JSON-LD

```
{
  "type": "object",
  "properties": {
    "name": "bri",
    "@type": ["iot:LevelData" ],
    "type": "integer",
    "min": 0,
    "max": 254
}
```

- Annotated schema is used to generate hypermedia controls for instances
- E.g. a link with a target attribute containing the annotation

Work on API automation

- Abstraction to semantic annotation
- Consumed and exposed APIs
- Abstract interactions
  - Property – read, write
  - Action – invoke
  - Event – subscribe, unsubscribe
- Programmatic abstract API
- Node-RED examples

Semantic API Examples

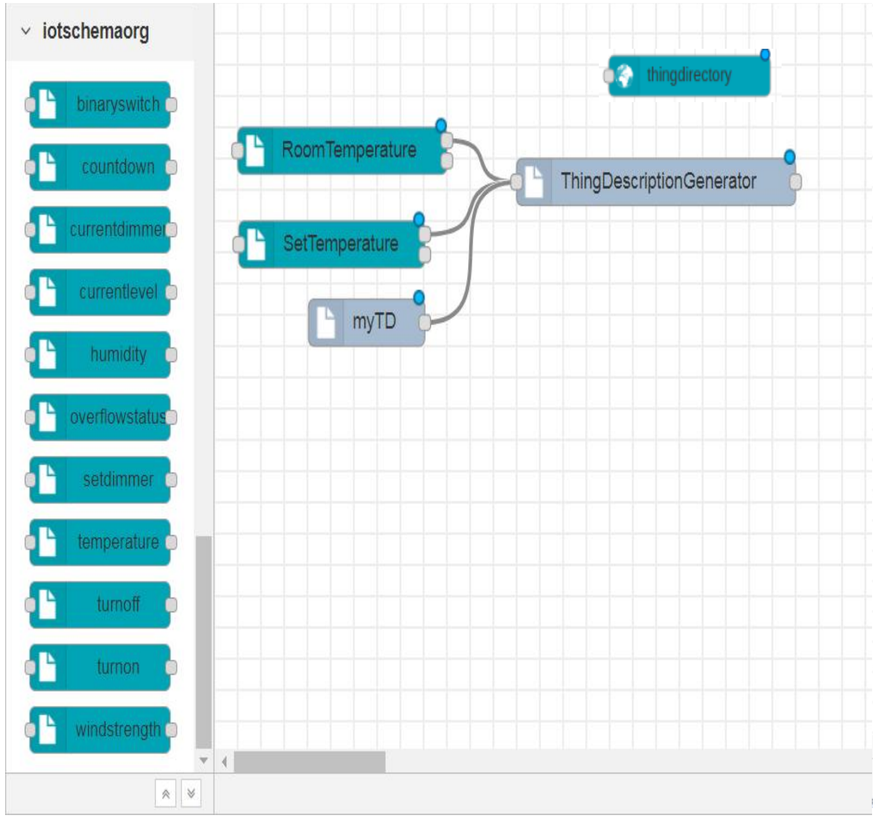
```
// Semantic Lookup returns instances capable of semantic lookup
thing = local-directory.lookup-by-simple-template;
light = thing( {"@type": ["iot:Light", "BinarySwitchCapability"] } )
switch = light.property( {"@type": "iot:BinarySwitch"} )
rgbcolor = light.property( {"@type": "iot:RGBColor"} )
turnon = light.action( {"@type": "iot:TurnOnAction"} )
setlevel = light.action( {"@type": "iot:SetLevelAction"} )

// read() function with and without DataItem filter
>>> console.log( switch.read( {"@type": "iot:BinarySwitchData"} ))
true

>>> console.log( switch.read() )
[{ "@type": "iot:BinarySwitchData", "value": true },
  { "@type": "iot:ApplicationTypeData", "value": "tester" }]

// write() function
switch.write( {"@type": "iot:ApplicationTypeData", "value": "Light"} )
```

Semantic API Examples (2)

```
// Write of multiple DataItems in a structured DataInstance
rgbcolor.write( [
  {"@type": "iot:RedColorData", "value": 255},
  {"@type": "iot:GreenColorData", "value": 255},
  {"@type": "iot:BlueColorData", "value": 255} ] )

// invoke() function
turnon.invoke()

setlevel.invoke( [{"@type": "iot:LevelData", "value": 170},
{"@type": "iot:TransitionTimeData", "value": 100}] )

// chained semantic references
>>> console.log( thing({"@type": ["iot:Light","BinarySwitchCapability"]})
.property({"@type": "iot:BinarySwitch"})
.read({"@type": "iot:BinarySwitchData"}) )
true
```

Enriching the device models with iot.schema.org Semantics

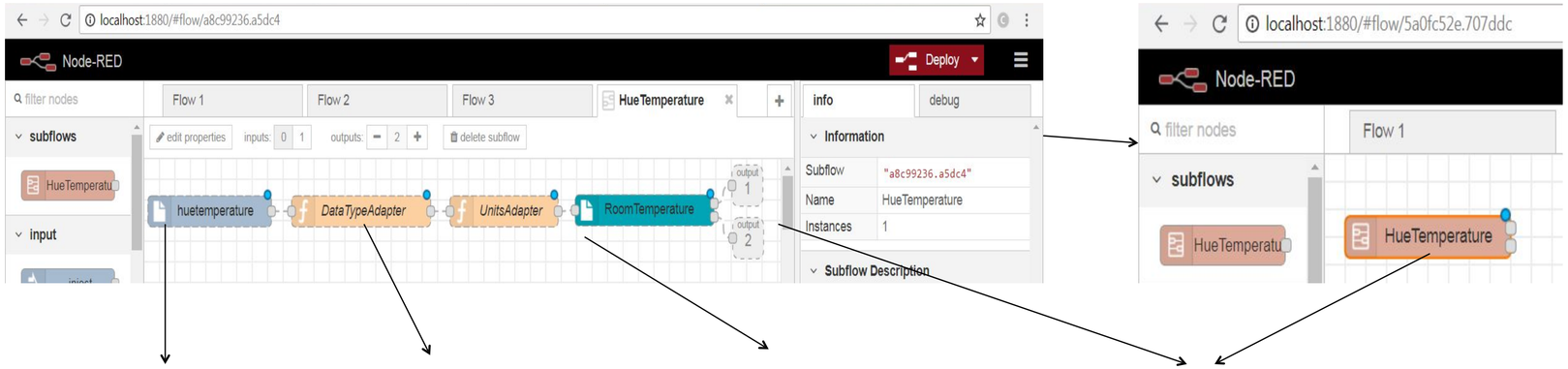
Type 2 Nodes




```
"@type": [ "Thing", “iot:Thermostat" ],
  "id":
  "urn:dev:wot:panasonic:airconditioner",
  "security": [{"scheme": "basic"}],
  "iot:isAssociatedWith" : { "@id": "Room1",
  "@type": "iot:Room" },
  "properties": {
     "temperature": {
        "@type": "iot:Temperature",
        "iot:capability":
  {"@id":"iot:Thermostat"},
        "io:isPropertyOf": { "@id": "Room1",
  "@type": "iot:Room" },
        "type": "object",
        "properties": {
          "temperatureValue": { "type":
                "number",          "minimum":
                10.0,  "maximum": 40.0 ,
                “iot:unitCode”: “iot:Celcius” } },
        "writable": false,    "observable": true..
```

Semantic Integration of Existing Things with iot.schema.org

Type 2 Nodes



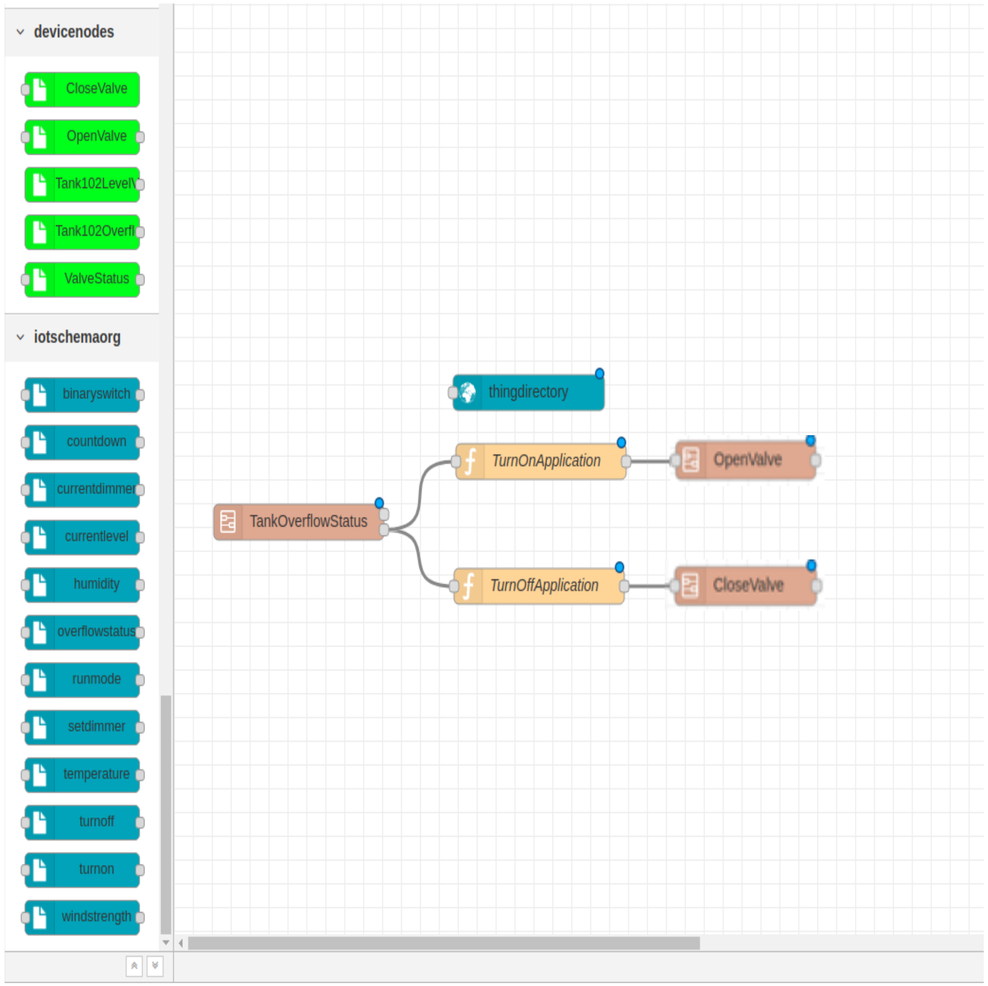
{
celsius: 25,
timestamp:13:00
}

Datatype adaptor
(Int to float)
{
input: {
celcius: **25**,
timestamp:13:00
},
output: {
celcius: **25.0**,
timestamp:13:00
}
}

UnitCode adaptor
{
value: **77.0**,
unit: **fahrenheit**,
timestamp: 13:00
}

Output1: {
TemperatureShape,
}
Output2: {
value: 77.0,
iot:unitCode: fahrenheit,
timestamp: 13:00
}

Recipe Flow Creation
Application Creation



Recipe: A template that defines orchestration of Things.

- Models Things required for orchestration
- Describes how Things should interact

Node-RED Node: Recipe ingredient

Node-RED Wire: Recipe interaction

Use Cases:

- Create a Recipe as Node-RED flow.
- Add context to flow JSON description
- Store Recipe to Thing Directory

Going Forward

- Set up the CG
- schema.org integration
- Accept definitions for target ecosystems
  - LWM2M/IPSO (Ericsson), OCF, SmartThings
- Work with IIC to create testbeds for semantic interoperability

Upcoming Teleconferences

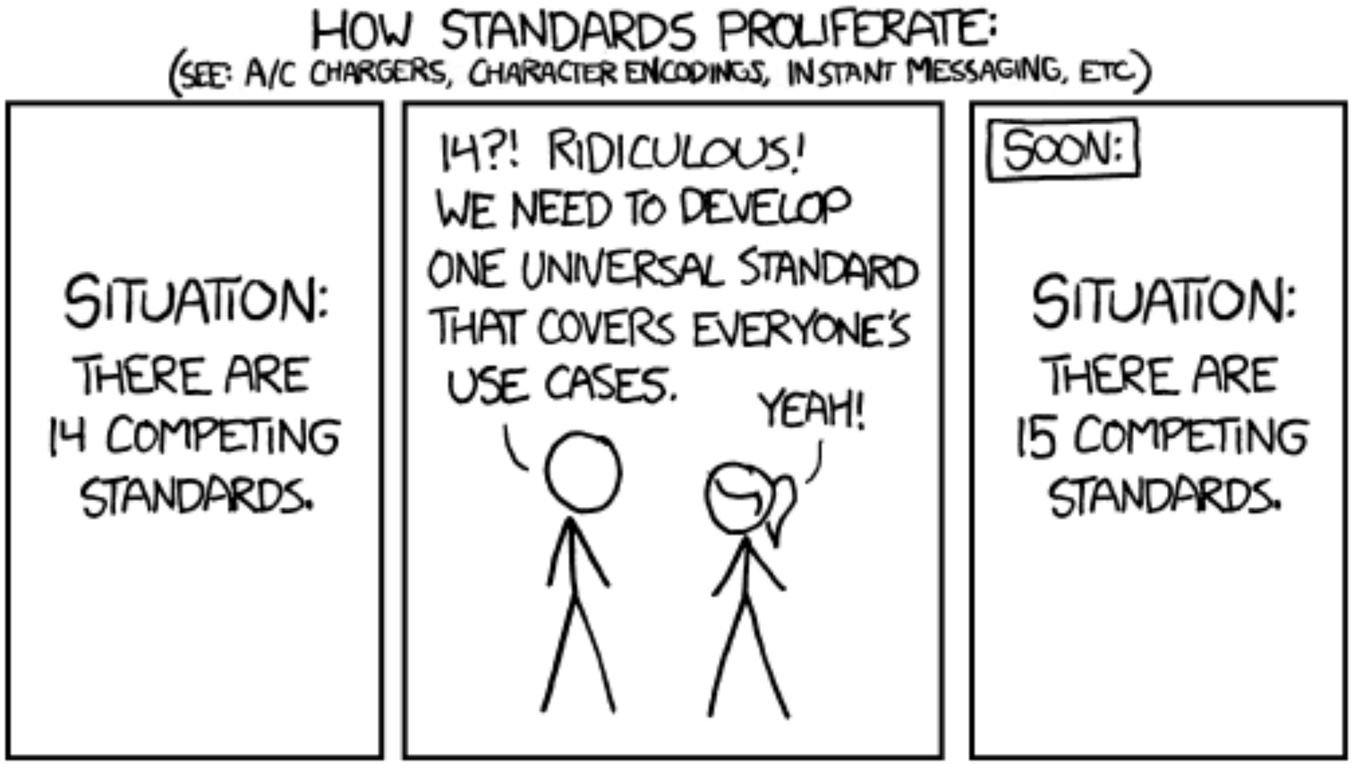
- Dr. Amelie Gyrard – Semantic Web of Things
  - Industry-wide survey of existing definitions
- Bruce Nordman – Lawrence Berkeley Laboratory
  - Device descriptions for energy monitoring

SSN Workshop
Exit Keynote
(condensed)

ISCW 2018

October 9, 2018

This is the Problem being solved:

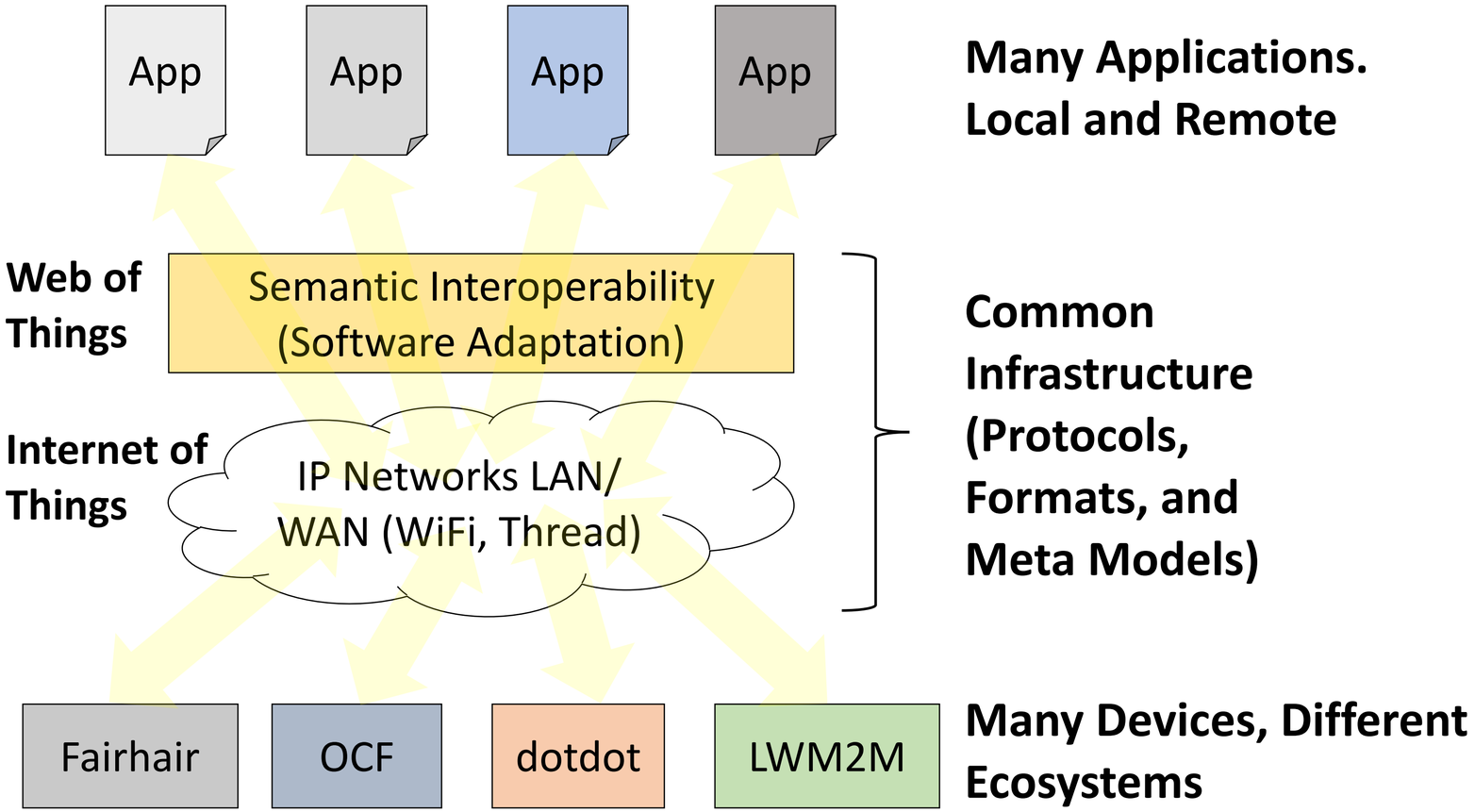


Source: https://xkcd.com/927/

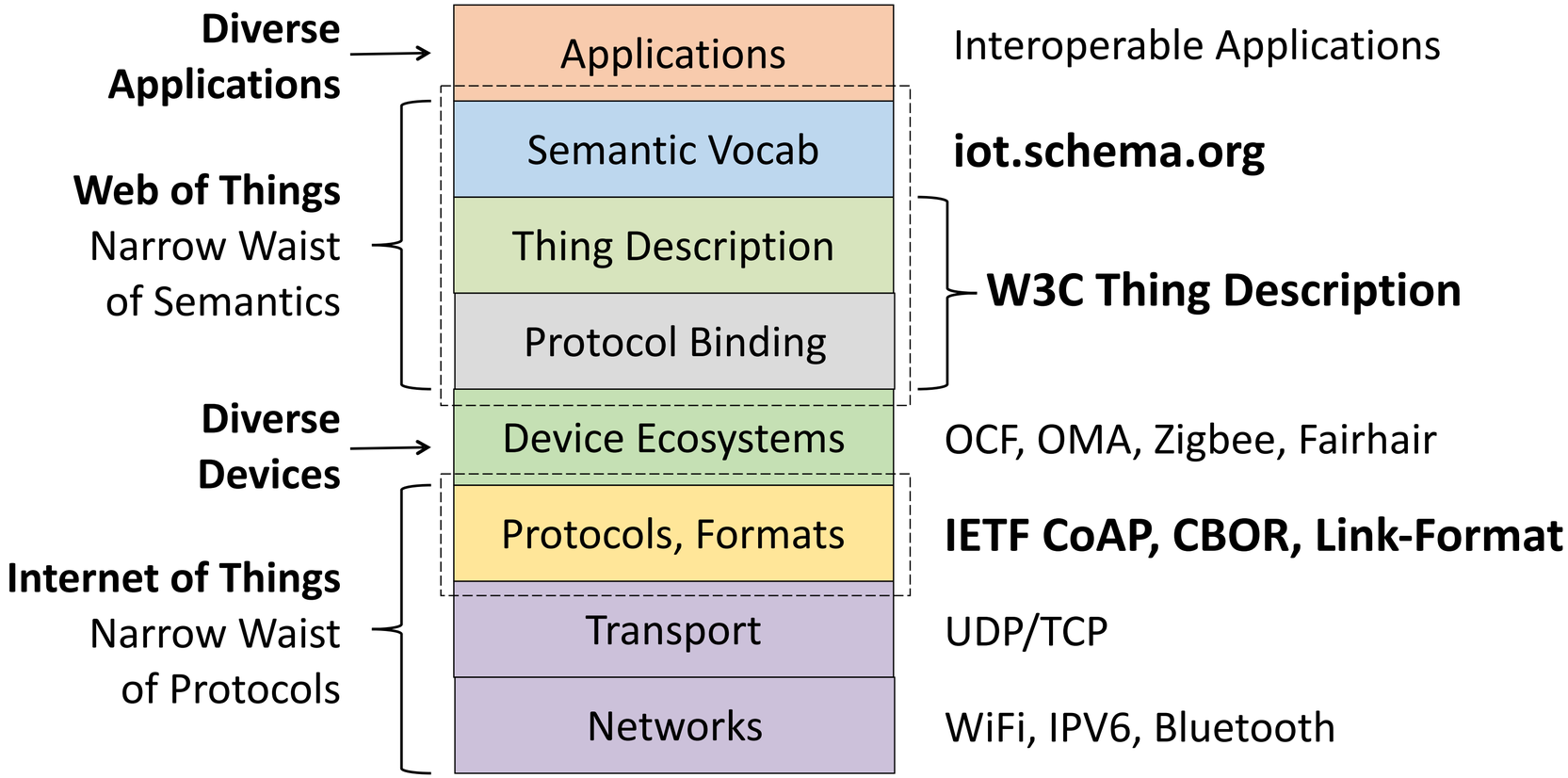
Problem being solved – Semantic Interoperability for IoT

- Acknowledge the diversity of IoT device ecosystems
  - Not another device standard
  - Adaptive to diverse protocol, language, and data models
  - Distill the common and stable operational features
  - Second "narrow waist" for systems above IP networks
- Address the ease of use of Semantic Web for IoT and use of IoT for Semantic Web
  - Not another IoT ontology
  - A conceptual layer that models connected things in relation to existing ontologies

Narrow Waist in System Design

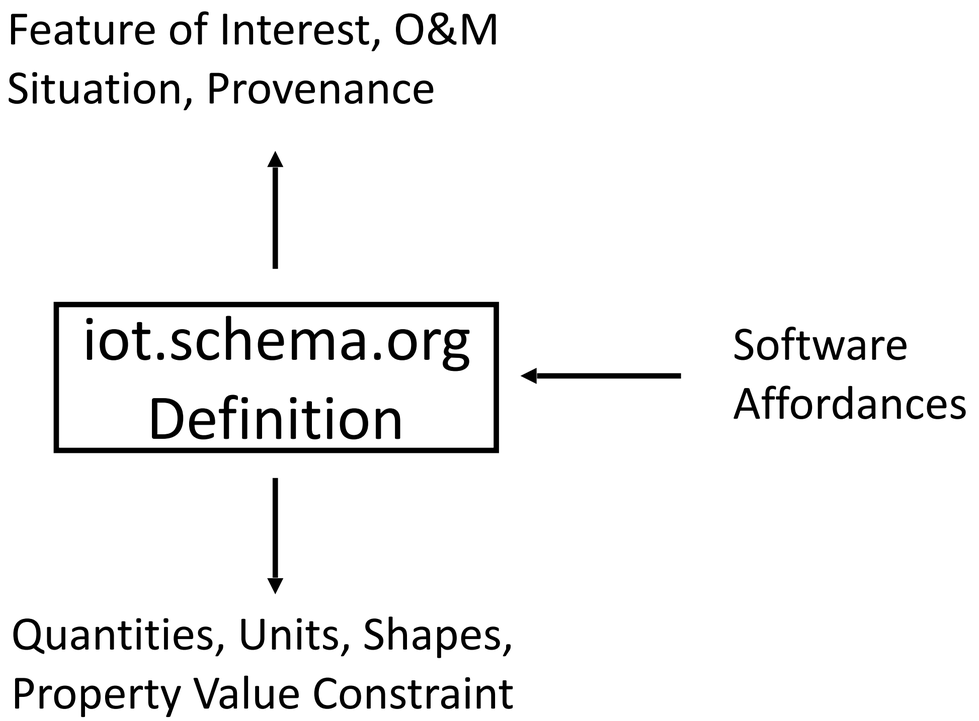


Diverse Devices and Applications, Common Protocols and Semantics

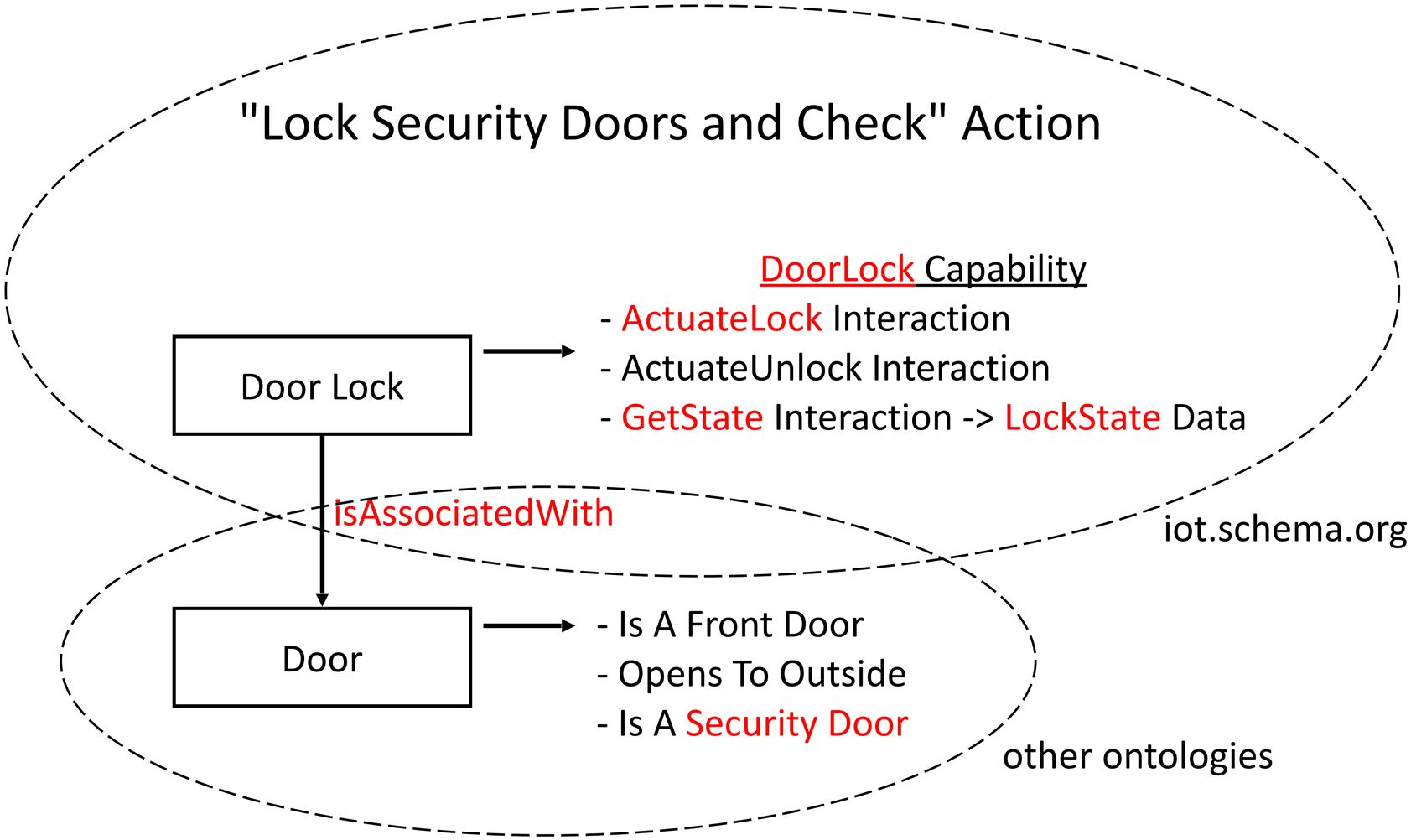


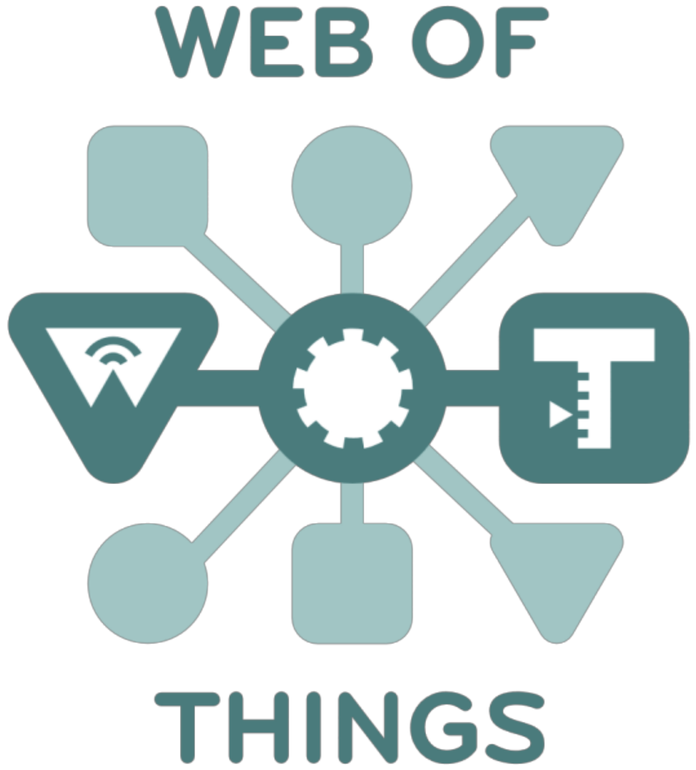
Integration with other Ontologies

Enables Well-Characterized interactions with Physical Entities



Connect things to the real world



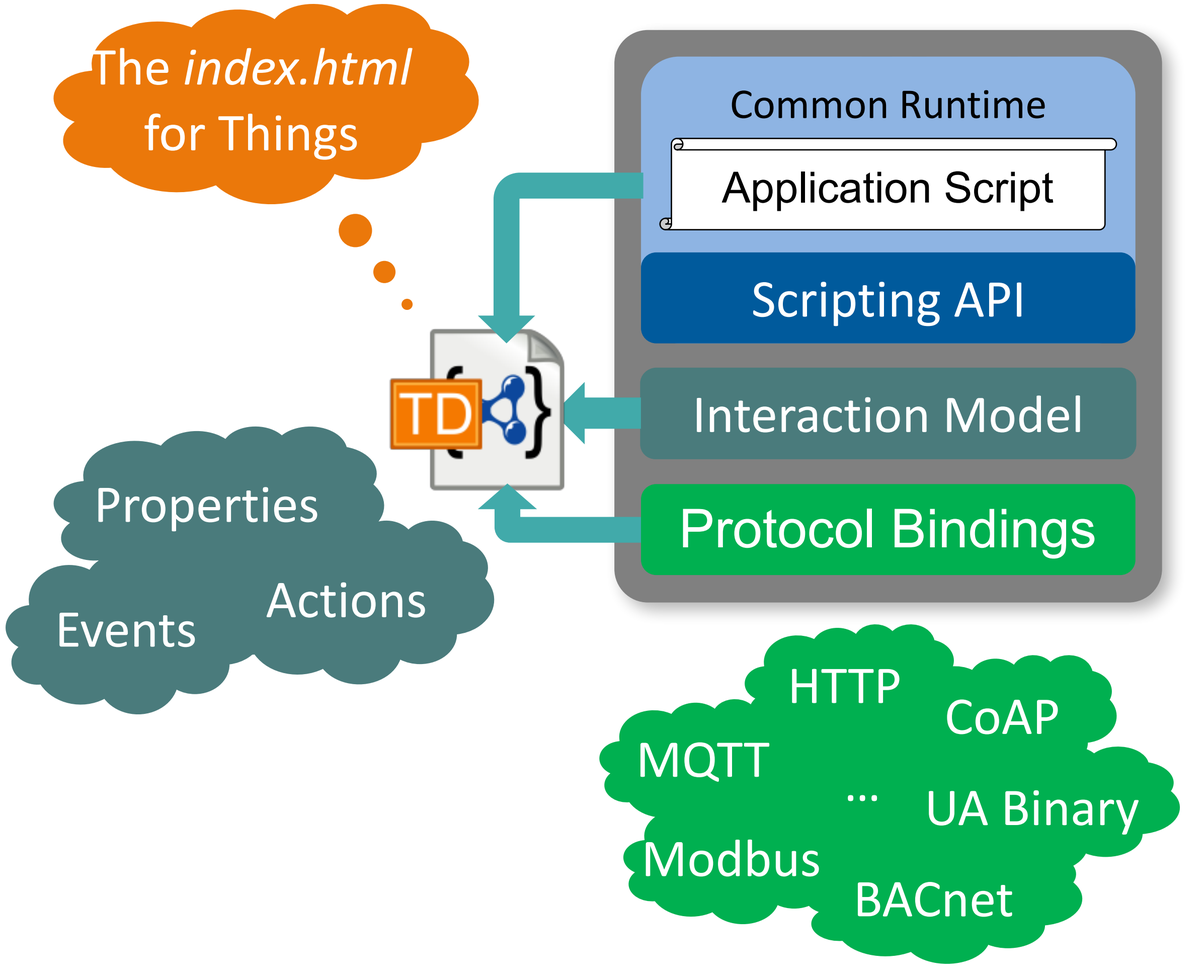


W3C Web of Things Update

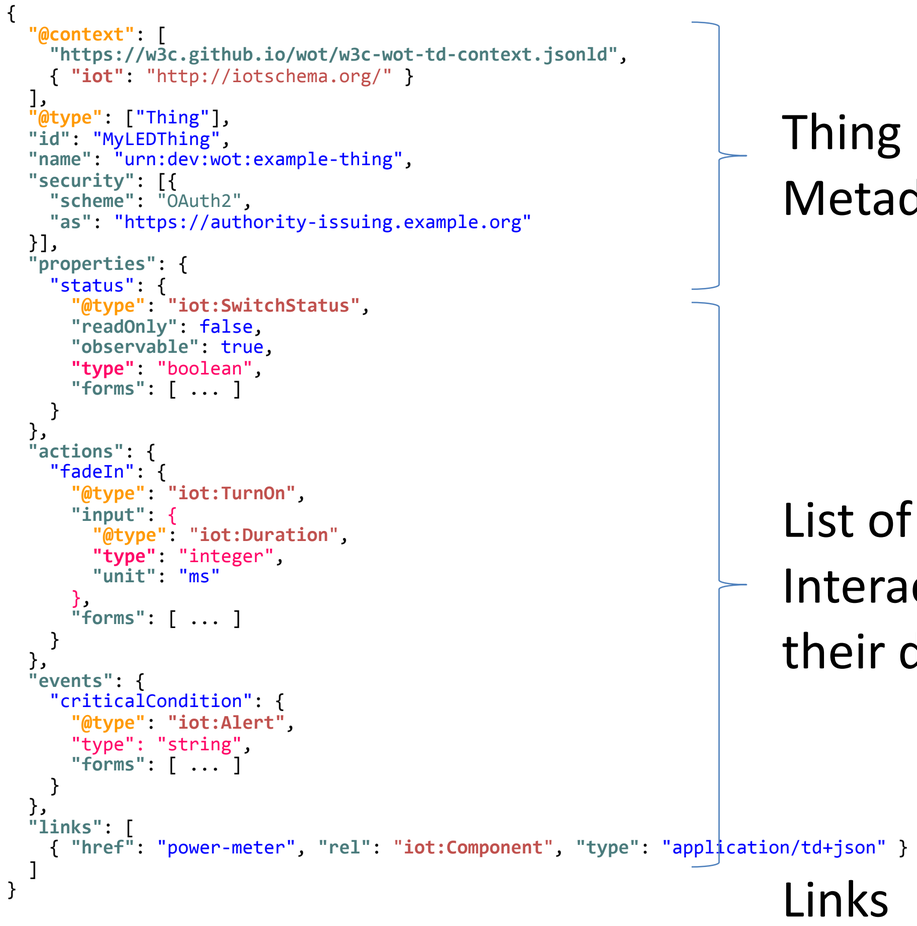
IETF 103, T2TRG, Bangkok, Thailand, Nov 2018

W3C WoT Working Group

- Mission
  - Counter the fragmentation in the IoT by adopting Web technologies to **describe** and **complement** existing ecosystems
- Deliverables
  - WoT Architecture
  - WoT Thing Description ("TD")
  - WoT Binding Templates (informative)
  - WoT Scripting API
  - Security & Privacy Guidelines (informative)



WoT Thing Description – a JSON-based Format



- Semantic Annotations
  - Linked Data for machine-understandable metadata
  - **JSON-LD** processing for Semantic Web tooling, e.g., reasoning, semantic queries (SPARQL)
  - Raw JSON processing for programmatic handling, e.g., embedded devices, user interfaces, scripts
- Data Schema
  - **JSON Schema** vocabulary in Linked Data
  - Compatible with existing validator implementations
- Hypermedia Controls
  - Links to express relations to additional metadata and related Things (e.g., to model complex system)
  - Forms to express how to construct requests that are understood by existing IoT devices

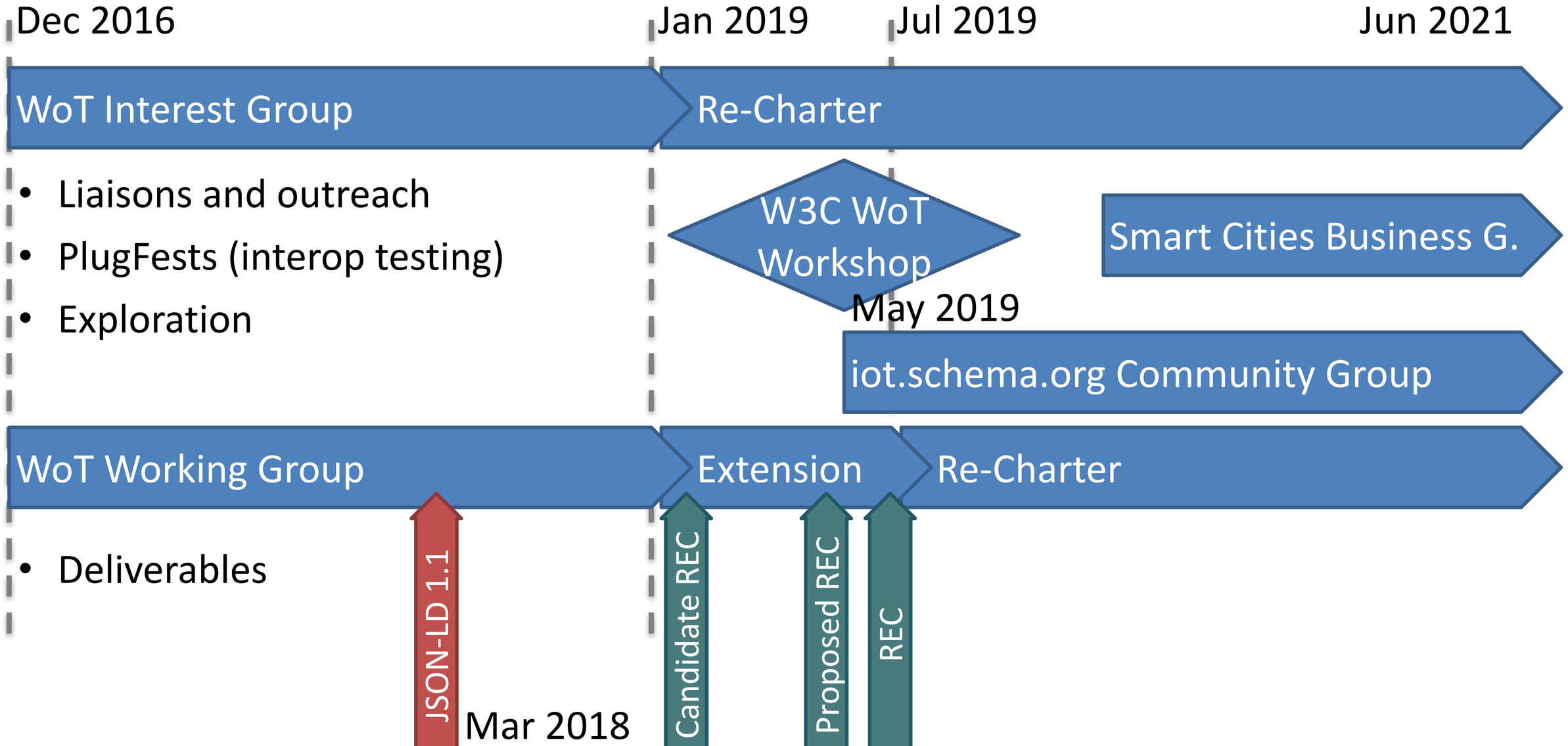
WoT Thing Description – a JSON-based Format

- TD extension points
  - Pluggable domain vocabularies (cf. Linked Data)
    - Refine TD to have meaning within application context
    - Existing, e.g., SSN, SAREF
    - Collaborative, e.g., schema.org / iot.schema.org
    - Converted, e.g., OPC UA Companion Standards
  - WoT Binding Templates
    - Describe concrete operations of Interactions using Web forms with information how to construct messages (e.g., method, headers)
    - IoT available, e.g., HTTP, CoAP, MQTT, OCF, LWM2M
    - Industrial to do, e.g., Modbus, BACnet, OPC UA

Recent Changes

- Features
  - Event parameterization (subscription, data, and cancellation subnodes in Event)
  - URI Templates
- Term alignment
  - `writable` → `readOnly` (JSON Schema compatibility, typo avoidance)
  - `label` → `title`
  - form `mediaType` → `contentType` (to define required media type parameters)
  - from `rel` → `op` (to much pushback on "form *rel*ation types", now *op*erations)
- New terms
  - `version`, `created`, `lastModified`, `safe`, `idempotent`, `unit` (UCUM)

W3C WoT Roadmap



Next Steps and At-Risk Features

- Todos
  - Algorithm to transform JSON-LD 1.1 syntax to JSON-LD 1.0 REC
    (JSON-LD 1.1 is still in draft phase, *"@container": "@path"* feature missing)
  - WoT Arch and TD Candidate Recommendations (CR)
  - Definition of test cases
  - Implementation of Test Suite (good existing basis due to PlugFests)
  - WoT IG Proposed Charter

- Features still under discussion
  - Meta-Interactions
    (read all Properties, write multiple, list active Events, …)
  - URI Template abstraction (integration into Action input)

Contact

Matthias Kovatsch

matthias.kovatsch@siemens.com

CoRE Applications

- Convention and template for application designers building hypermedia-driven application interfaces in a structure way
  - draft-hartke-core-apps-08
- Goal: implementors can easily build interoperable clients and servers; others can re-use components more easily

CoRE app API Components

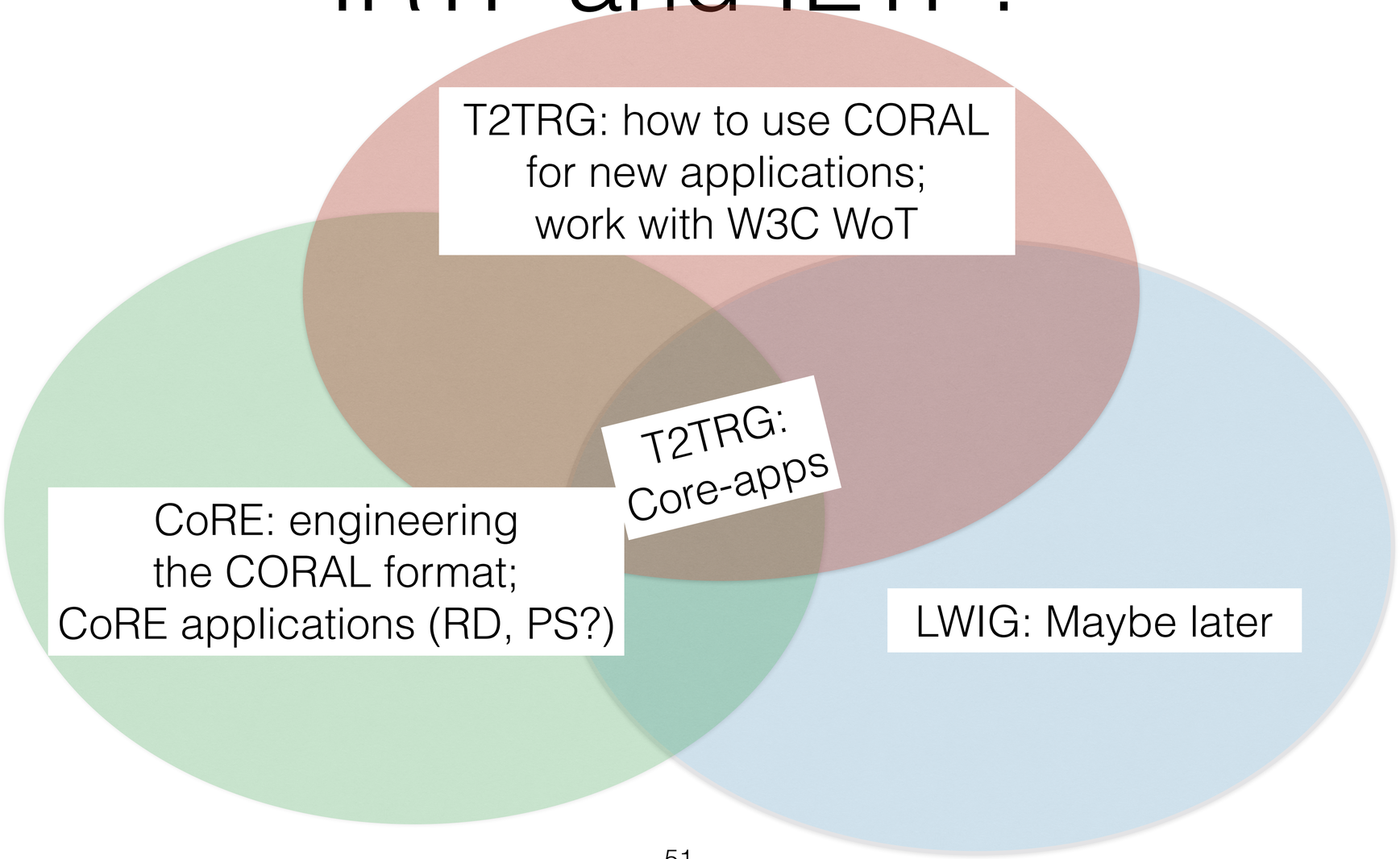
- Communication protocols, identified by URI schemes
- Representation formats, identified by Internet media types
- Link relation types
- Form relation types
- Template variables in templated links
- Form field names in forms
- Well-known locations

in-band instructions
to a client for interfacing
with a given application

Template for CoRE Apps

- Human-readable information about API components (see previous slide) and other useful information:
  - Application name
  - Interoperability considerations
  - Security considerations
  - Contact person
  - Change controller / author

Working on CORAL in IRTF and IETF?



CoRAL

The Constrained RESTful Application Language

Klaus Hartke

CoRAL is a *hypermedia representation format* for the hypermedia model described in draft-hartke-core-apps:

- Links

  change *application state*.

  "{context} has a {link relation type} resource at {target URI}, which has {target attributes}"

- Forms

  change *resource state*.

  "To {form relation type} the {context}, make a {method} request to {target URI}"

53

CoRAL aims to reduce the cost of hypermedia:

- **Reduce size of representations**
  - Encode links and forms in a compact, binary format
  - Use numbers instead of strings
  - Use sensible default values

  Most links and forms can be expressed in a few bytes

- **Reduce number of roundtrips**
  - Embed a representation of the link target and forms manipulating the link target at the link source

54

- **Simplify implementations**
  - Same option concept as CoAP

  This simplifies URI parsing and reference resolution a lot

CoRAL

A language for the description of typed connections between resources on the Web ("links") and possible operations on such resources ("forms") as well as simple resource metadata for automated software agents.

* Data and interaction model

* Compact, binary format
  -- suitable for constrained environments

* Lightweight, textual format
  -- easy to read and write by humans

CoRAL Examples: Textual format

- Interchange format is binary (CBOR)
  - Could use CBOR diagnostic notation to discuss
  - “Ready to munch” format (including CIRIs) gets tedious quick
- Instead: Use separate textual format
  - **Danger**: textual format can shape thinking away from actual data
  - **Danger**: textual format can acquire “syntactic processing” that is not actually part of the binary format
  - **Danger**: hand-made examples [https://github.com/t2trg/wishi/blob/master/slides/hand-made-examples.pdf]
  - Keep these dangers in mind ➔ textual format best way to discuss

```
<!-- HTML5 -->
<link rel="stylesheet" href="/style.css">
<link rel="icon"       href="/favicon.png">
<link rel="license"    href="/license">

// CoRAL
stylesheet </style.css>
icon       </favicon.png>
license    </license>
```

link relation type

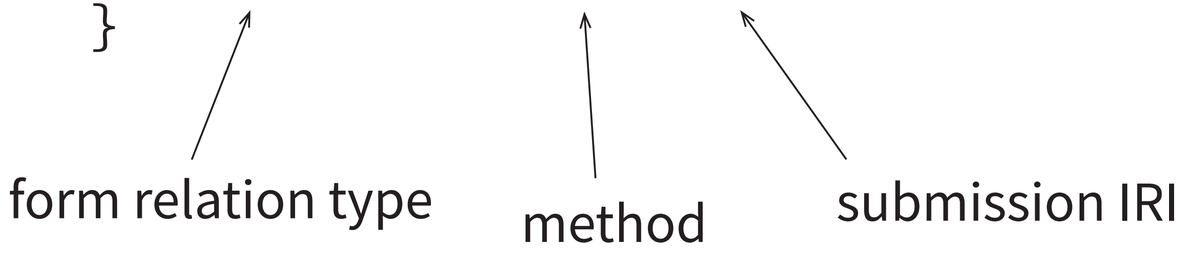
link target (IRI)

```
// representation of <coap://robbie.robot/>

id       354675
name     "Robbie the robot"
likes    <coap://susie.robot/>
likes    <coap://nikki.robot/> {
   likes <coap://chris.robot/>
}
```

link target (literal)

link from nikki to chris

```
// representation of <coap://susie.robot/>

id              827446
name            "Susie"

power-led       </leds/power1>
power-led       </leds/power2>
status-led      </leds/status>
headlight       </leds/head> {
    update -> PUT <> [accept "example/boolean"]
}
```

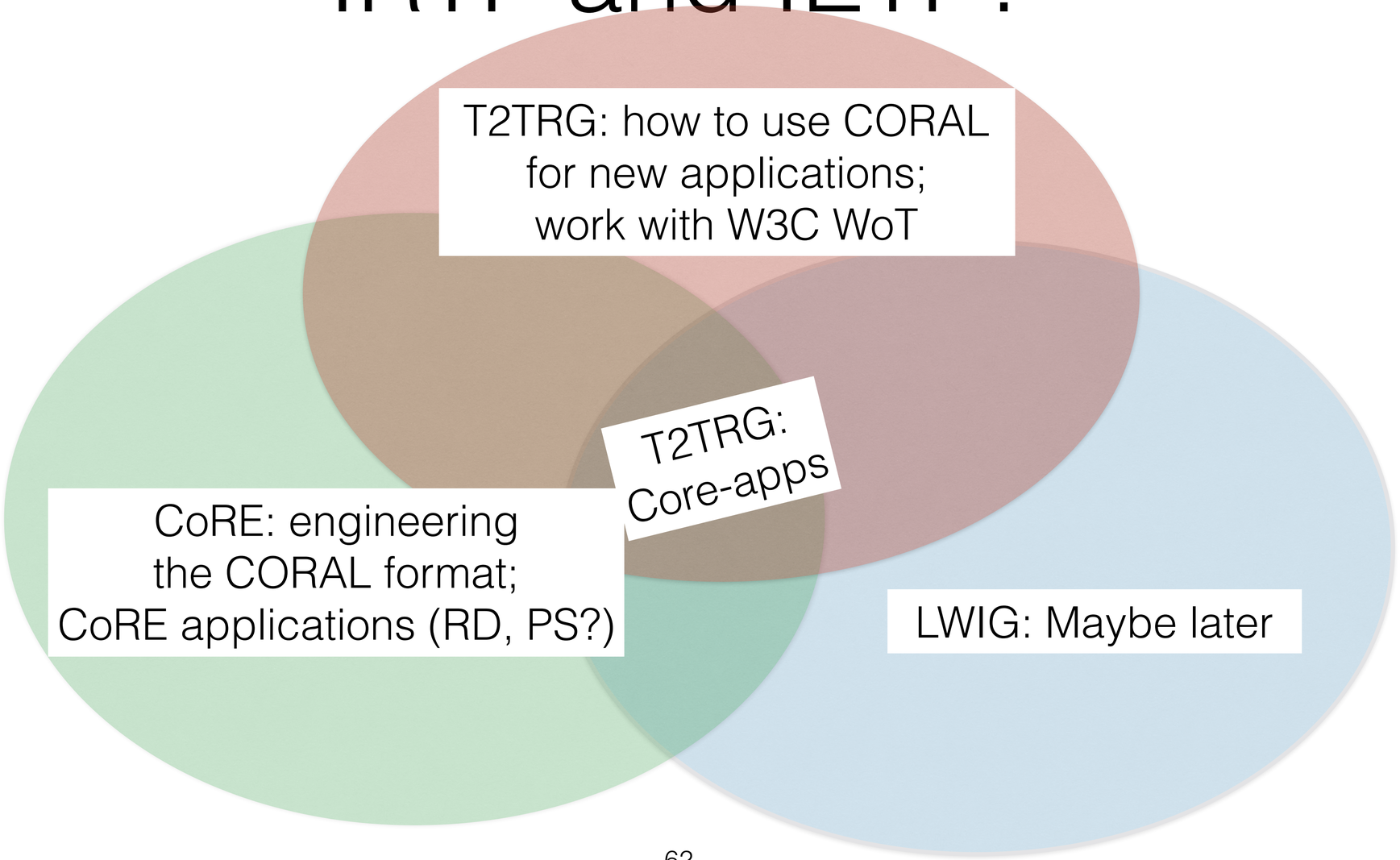


```
// representation of <coap://susie.robot/tasks>

item </tasks/1> {
   description "Pick up the kids"
}
item </tasks/2> {
   description "Return books to the library"
}
create 60-> POST <> [accept "example/task+coral"]
```

```
// representation of <coap://susie.robot/tasks/3>

description "Take out the trash"
collection </tasks>
update -> PUT </tasks/3>
delete -> DELETE </tasks/3>
```

61

Working on CORAL in IRTF and IETF?



Friday Work Meeting

- 8:30 to 13:20, room Boromphimarn 4
- Breakouts from 10:00 to 12:00
  - E.g., Edge computing, Security, Hypermedia
  - Also: COIN (Computing in the Network, room Boromphimarn 3) side meeting, relevant to IoT

Friday Work Meeting

Time	Presenter(s)	Topic
8:30	Chairs	Welcome & Short Introduction. T2TRG/IETF work.
8:40	Various	Plenary
	Jungha Hong	Problem Statement of IoT integrated with Edge Computing
	Erik Nordmark	Computing at the Edge
	Thorsten Dahm	Automated IoT Security
	Mohit Sethi	Enabling Network Access for IoT devices from the Cloud
9:40		Breakout planning
9:50		Break for breakouts
10:00	Various	Breakouts (see below)
12:00		Plenary (discussion, next steps)
		Consolidating results from the breakouts
		Consolidating results from the hypermedia discussions
13:20		meeting ends

Problem Statement of IoT integrated with Edge Computing

- New challenges for IoT services originated from the changes in the IoT environment
- Edge computing as an emerging technology in IoT
- Use cases of Edge computing in IoT (two demo videos)
  - Smart constructions utilizing EdgeX
  - Real-time control system by Rotary Inverted Pendulum system

Friday: Computing at the edge

draft-nordmark-t2trg-computing-edge-00

Look at edge computing from a compute perspective (cpu, memory, storage, connectivity) to determine network needs

Consider e.g., applications deployed in cloud (as containers or VMs) and what it would mean to deploy them at the edge

Automated IoT Security

- Automating Risk Analysis, Vulnerability Assessment ➜ Secure Configuration
- Automating continuous monitoring and audit

Solving the mismatch between

- The security capabilities and settings with which IoT devices are designed / manufactured / deployed
- The actual security requirements of the IoT devices in different environments over time