T2TRG Meeting at Lisbon 24-25th September

Chairs: Carsten Bormann & Ari Keränen
Note takers: chairs, Maxime Lefrançois, Matthias Kovatsch

Chairs: Welcome, Meeting overview, T2TRG Status

Slides:
https://www.ietf.org/proceedings/interim-2016-t2trg-04/slides/slides-interim-2016-t2trg-04-sessa-chair-slides-01.pdf

For joining discussions, register to the mailing list at https://www.irtf.org/mailman/listinfo/t2trg

We'll look at CoRAL, (https://tools.ietf.org/html/draft-hartke-core-apps) and HSML, (https://tools.ietf.org/html/draft-koster-t2trg-hsml), followed by events, time series, transcriptions, streams and should we develop a taxonomy or something for these. Afternoon we discuss LWM2M model translation to CoMI and YANG, Seluxit REST-ful API for Lemonbeat devices. The Sunday (some won't be here anymore) topics are mainly about Security.

Meeting potentially at IETF 97 Sunday Nov 13 co-located with ICNRG.
Following one maybe in February @EWSN

News and Surprises from W3C WoT, Agenda Bashing

Discoveries from WoT discussions: Actions and properties are bound to physical properties. If actions cascaded, results in problems. Also who did what in which order. How to enable origin server; thing to decide what is the value of property.

Mirror in the cloud: it's challenging how to do synchronization between the two.

Are events needed? Is there more to events than just subscribing to them? For example, confirm alarm. Event exists after fact; can operate on events. What kind of actions do we want to distinguish? There is a point about events, are they needed or not? Actions are prototypically involving real processes. Properties can be updated instantaneously.

Matthias: it was a question by Dom; is it more than subscribing to a value? Then its response was "yes" and he used the alarm example. An alarm is not a value, it has actions associated, it acts like an event but after that it exists on its own. You are notified about but then it lives on its own.
When do we need events; might not need when it's just a value change?

Daniel Lux: What about longer running actions? Many actions put into a unique action such as a transaction?

This discussion is something that could lead to an ontology. How we can agree what is an action etc. Important to have taxonomy. For example "event" means many things. Longer running events and many actions that result in a process.

Well known ontology called DUL (SSN was based on and trying to align with). Now work on separating SSN and DUL but have mapping between them. DUL originally from philosophy domain. Focusing on conceptual model on basic things. One of the most commonly accepted ontology for defining the basic concept of the world. Both syntax and conceptual issues. See http://www.ontologydesignpatterns.org/ont/dul/DUL.owl# (ex, use tool Protégé http://protege.stanford.edu/ and load this ontology from its URL).

How to secure end-to-end. Take resourceful model as the base; resources and handful of verbs. Each request with OAuth token. Some things missing; rate limiting and DDoS. In security we need to go cross layer and deep down. WoT work around IAM topics. Humans vs. Things in IAM.

How do we verify/validate and benchmark? Need to capture scenarios and come with solutions to them. WoT architecture document capturing some but more work needed.

Can we use resourceful infrastructure to orchestrate data flows. Have that on different level than control. For example, use property to initiate video streaming but not accessing the resource from that property. Data and control have often different requirements for reliability and latency.

Johannes: can we use the REST as the infrastructure for streaming data? Shouldn't we separate content and interactions? What about the subscriptions?
Carsten: we want the control to be reliable, but for the data it sometimes makes more sense to have low latency than reliability. This is part of a later session, will add "data flows" to that session

Klaus Hartke: CORAL vs. HSML -- way forward?

Slides:
https://www.ietf.org/proceedings/interim-2016-t2trg-04/slides/slides-interim-2016-t2trg-04-sessa-coral-00.pdf

CoRAL is hypermedia model to describe APIs. Links and Forms in "human web". Can we use same for things? Optimizing size of the data and number of round trips for hypermedia controls.

Links are like properties, they change application state: {context} has a {link relation type} resource at {target URI}, which has {target attributes}.

Forms are as actions, they change resource state: to {form relation type} the {context}, make a {method} request to {target URI}.
Johannes: similar to HAL. But also forms. How is payload embedded?
Klaus: data would embed CoRAL. Links, but also literals (e.g. "nr of wheels in car" not needing to be new resource). HAL, extended with forms, in compact CBOR. Hydra similar but different terminology.

Johannes: do we need to differentiate forms and links?
Klaus: links don't really tell how to interact with the resources. Two concepts seems useful.
Matthias: URI templates have similarities. Cascading intermediaries in between bring issues with forms. Link is simplified form. But makes sense to keep separate.
Carsten: HTTP only has parameterless safe methods. With FETCH coming, we want to separate parameterless from the parametrized.
Matthias: with link it's fire and forget; with form you want to often monitor what happens. There is difference between the two. If you have a link you should know if it is safe and operations on it will not change the resource state. If you have forms you have to care about the impacts, potentially monitor.
Klaus: links are nouns "car is a wheel". Actions are verbs "withdraw money".
Johannes/Klaus: On wire level both end up the same. Difference in semantics.
Johannes/Carsten: traditionally only forms had payload; now changing with FETCH. The way we were using query parameters was already changing that you can do query parameters, or posting a search.
Klaus: CoRAL has URI templates, you can construct a URI with its parameters

Michael Koster: HSML vs. CORAL -- way forward?

Slides:
https://www.ietf.org/proceedings/interim-2016-t2trg-04/slides/slides-interim-2016-t2trg-04-sessa-hsml-00.pdf

HSML built on top of CoRE link format, and SenML. Standardized data model and interaction model. New design patterns that extend REST (actions, properties, asynchronous things happening, …).

HSML Collections are extending CoRE link format and SenML. often the data and controls don't need to be all referenced at the same time. In HTML you have metadata in RDFa or microdata, forms in the same page etc., that's not something we can afford in the IoT?
HTML forms are strongly bound to HTTP verbs. We need an abstraction for other protocols. Transfer layer abstraction to unify REST and pub/sub.
How to have hypermedia interactions but stateless client. How can stateless clients do discovery. A state change, even if not immediate, is still a state change. It should be covered by REST. A message may be a request to change a state of the system?

Machine proxy, "device shadow' interaction: we don't really know the machine we are interacting with, we are just interacting with a proxy, you can do things with it as if you were interacting with the device.

Using links to describe links: have links that point to forms. Forms are metadata that can still be annotated with hyperlinks.

Matthias: precise "stateless client", it makes global confusion.
Michael: the client doesn't have to remember the state of the server or the last thing he did on the server.
Matthias: shouldn't it be the server that be stateless? But the client know what he wants to do and the sequence of actions that are required?
Michael: a session is not required
Johannes: we agree that all the information about the session is held in the request and response messages, no need to store it between requests
Michael: a stateless interaction yet with a state but that is included in the messages themselves
Carsten: client usually is not stateless because it knows what he wants to do. Web browsers may be stateless because the plan is in the head of the user.
Michael: clients may want to cache the resource state?
Matthias: Good to update the slides to avoid confusion about the state

Johannes: link annotation?
Michael: put terminology in the links that contain descriptors about application semantics. For example, temperature sensor: 'temperature' would be a term in the link. Using CoRE 'rt', but rt can have values that describe application semantics.
The terminology needs to be annotated, but the full description of the semantics of the controls does not need to be contained in the link themselves, it can be described elsewhere. This is some kind of a discovery workflow where the client discovers on the fly the semantics of the actions he may do on the resources.

Carsten: what is also interesting is not only the semantics of the controls and the actions, but also the relations between the resources. For instance, two temperature sensors may be located in different places.
Michael: true, it's like annotation, but might not be relevant to describe this in 'rt' attribute? Do we need another hypermedia control or another term for that?

Matthias: although CoRE link format is on top of web linking, one interesting aspect of web linking was the richness of the "rel" relation types. Whereas rt is the absolute type, rel can be used for discovery following relations.
Michael: rel is needed to describe the context, We should be able to reuse any of the linking vocabulary that has been defined.
Matthias: good to be explicit that you're doing a common horizontal vocabulary in this work
Carsten: also need taxonomy of the vocabulary; distinguish categories

Matthias: have master's student looking into case of electric vehicle (EV) charging to exemplify this. Found small set of terms that seem to be re-usable.
Daniel L: different levels here; a thing needs only low-level values. Someone configuring would need more semantics. System configuring whole system would need another class. Would be good to categorize the levels. Device, GW, System; would have different levels of semantics.
Matthias: even at lowest level need common semantics.
Daniel: a relay is more generic than a light switch. It can also turn on and off, but you don't know in the end the semantics of this on/off It may be light, valve pump, or anything else.
Michael: collections in HSML address this. Links annotated with descriptions.
Carsten: another thing: composition. Switch + PID controller + heater = room temperature control. Want to use that as whole unit.
Daniel L: what is also difficult is temporary manual properties changes, that should be reset automatically later.
Matthias: BACNET has safety over ride with priorities etc. Priority with access levels of the controller/person. Question is do we need this to be modeled?
Johannes: Need vocabulary of semantics for links
Matthias: re-usable vocabulary for starting, monitoring, exiting process via forms. Context in hypermedia. Inside context can use re-usable terms to go through process.
Michael: matches work-flow concept
Carsten: which parts are acted by system and which by application programmer? Application programmer does not want to focus on "following links" but to the application requirements.

Johannes: how does client now the context it is in?
Matthias: that's the "application state" in the client. Are you on Wikipedia or Banking site.
Johannes: does the URL give the state/context always?
Matthias: the context includes the location, process, and goal. "You are in charging process" (EV example). A context may be the list of links that you have followed, a bit like the forward back buttons in the browser. Noticed in current work that you need to mark when you leave a (sub)context.
Michael: clients need to remember where they come from. Relation types you've discovered on the discovery path. Maybe pre-aggregate context at the server? Make discovery state persistent.
Matthias: works only for entry points; can't jump into middle of flight booking. Current state is made of all the relation types you've followed.
Maxime: state does not necessarily tell about the intentions
Johannes: my way from the root to current state is the context; not all de-tours
Matthias: could be intention to do some part twice
Michael: state it's the whole graph. Maybe applies only to discovery though.
Matthias: we don't want sessions; important to only keep such state at the client
Michael: correct term now should be: evolving workflows
Matthias: session term need to be defined at the RESTful draft: shared knowledge between the client and the server.

Michael presented slides to compare CoRAL and HSML.

Differences in CoRAL and HSML include data models (which could be translatable), and where CoRAL uses media types for application semantics, HSML uses link annotation.

Carsten: is the media types vs. link annotation choice or do you want really both?
Michael: HSML could cover also other media types and point to them. Could also use types like hsml+lightning. May not be just a binary choice.
Carsten: in the end we need media types for more complicated data structures. For something like value "three" you don't need much.
Michael: That's why we use SenML; can define many different values.
Carsten: maybe re-visit data modeling and media types in a session later. Media types is flat namespace. Combinatorial explosion if you want to combine characteristics to new media type. Simple things are efficient but complex things not. Link annotations could be one way for solving the problem.
Michael: seeing that in composite media types at CoRE

Next steps: Looking into prototype involving both approaches. Trying to make tradeoffs and converging to single way.

Taking lightning example, Bulletin Board (BB) of CoRAL and implementing in HSML. Resource Directory (RD) as alternative discovery to BB. Comparing RD and BB.

Carsten: one key difference: CoRAL has optimized way to take data a part and put together.
Michael: haven't yet looked at any optimizations or mappings yet. CoRAL can be used to do binary mapping of HSML.
Matthias: thinking of proposing split: concept of control in CoRAL and encoding mechanism. Control part could be unified and have one serialization in SenML and other using CoRAL mechanisms with binary representation.
Carsten: SenML has binary representation too
Johannes: what CoRAL uses to make it efficient?
Michael: In the end it's just single set of keywords between CoRAL and HSML. SenML keywords are just hypermedia terms (even if they are only few characters).

Carsten: This is at the lower tier of data model. We should try to separate from semantic categories. Can you do HSML in CoRAL?
Klaus: yes
Carsten: could do HAL and HSML version of CoRAL and see what changes at semantic level. How about BB and RD?
Matthias: freedom on what to register. OCF wants to do some. BB just a media format to present things. Could have CoRE link format and BB lookups. In WoT same problems that are

solved by RD. At register you need end point identifier etc. BB does not have that functionality. Using the format and RD functionality could get something that unifies both.

Matthias: write down how things work together.

Klaus: should be possible to convert RDs to BBs. Just take RD items to BB.
Matthias: converting RD to BB lose all nice functionality of RD. Other way around: implement BB at RD. Registration with BB format; RD fills internal database with the info. RD could also support BB lookup.
Michael: status of RD draft: Peter implemented some of the stuff, but not sure of exact status.
Matthias: will combine this action item to see that RD draft now supports arbitrary link list formats. Useful for WoT and OCF
Michael: Accessing also data via BB? Goes towards Mirror Server.
Matthias: More lookups possible/needed for OCF (e.g., resolving UUIDS to endpoint addresses)

Matthias: experimental BB could be confusing as RFC. How far at OCF?
Michael: example of using OCF in RD would be fine.

Carsten: seems that we have a plan. IETF internet draft cut off at 31st of October.

Carsten Bormann: Events and time series

Slides (slide 21...):
https://www.ietf.org/proceedings/interim-2016-t2trg-04/slides/slides-interim-2016-t2trg-04-sessa-chair-slides-01.pdf

Many names for many things; different understanding for events, time series, streams, etc. Can we get taxonomy? One way to split: what level are we? Transports, transfer, serialization, data modeling.

Michael: one example of streaming at transfer layer: event source API. Keepalive and chunk encoding. HTML5 feature.
Carsten: rather weird piece; not using any of REST architecture. If you want to integrate to your app, need to go through lots of hoops.

Aspects of stream; "streamy aspects": possibly more than one packet per RTT. Periodic data. May be high volume. Separation of data and setup.
Michael: also applies to pub/sub

Conversational latency. Streaming: latency important but tolerable. Reliability often more important than latency.

Matthias: jitter could also be relevant

Carsten: this really should be about worst case delay; jitternot important in packet based systems
Ari: jitter makes knowing worst case delay hard
Matthias: one term for two different aspects now

Kaz: conversational interaction may include RPC like aspects
Carsten: use "conversational" here because that's familiar. For example robotics has very similar requirements.

Time series aspects. May be need for time synchronize over system boundaries.

Example: web streaming. Control file m3u8 containing links to snippets of data. Shows power of linking and REST for this kind of things. Client may negotiate the quality of each snippet with the server at any time.
Daniel P: assumes you can perfectly stream control file.
Carsten: not something you would necessarily use for phone calls
Johannes: other metadata in control file?
Carsten: sequence of links. When towards end of previous sequence, start retrieving next one. Weird things like wanting to open new TCP stream for data which goes through slow start. Not simple, but it works.
Kaz: want to include WebRTC here?
Carsten: not streaming mechanism as such, but works in different way; maybe good to add example

Example: enterprise service bus. Here events MUST NOT be lost. Bus here operates on events in many ways. Very different to how "bus" understood in many other fora.

Ari: seems we need a terminology/taxonomy document on this
Johannes: at WoT have been also looking into how people understand these concepts. Got caught up in what is a stream etc. Took step back to look into use cases and have criteria what the use cases give.
Carsten: microphone good example; big difference in microphone in telephone vs. in concert.
Johannes: didn't do too much of what is the criteria. E.g. "is it OK to lose one for lower latency".
Carsten: subscriptions.md at WoT github.
Johannes: Working on this. Options on what you would like to subscribe, etc. Very much cross-layer thing. Good for W3C and IRTF to work together on this.

Johannes posted link:
https://github.com/w3c/wot/blob/master/proposals/subscriptions/subscriptions.md
Ideas for the management on REST-layer, based on REST Hooks
https://github.com/w3c/wot/blob/master/proposals/subscriptions/README.md

Jaime Jiménez: Mapping from LWM2M model to CoMI YANG model

Slides:
https://www.ietf.org/proceedings/interim-2016-t2trg-04/slides/slides-interim-2016-t2trg-04-sessa-mapping-lwm2m-model-to-comi-yang-00.pdf

Preliminary work. Translating LwM2M model to CoMI and YANG. Many SDOs use hierarchical models specified with XML. Common patterns that can be expressed with YANG.

Humidity object as example. Rows are "resources" columns "attributes". Read/write/execute operations. Some mandatory resources for interoperability.
Hopefully in future can have script for automatic translation from LwM2M to YANG. YANG has hundreds of statements and many ways to express many things. "Exec" attribute of LwM2M mapped to RPC of YANG. Object instance now using "instance" key attribute; could use some other way eventually.

Carsten: "instance number" and "instance" the same thing?
Jaime: 3301 object ID; typo in the slides
Michael: objectID/instance number in URIs
Jaime: instance number the key in YANG. If you have query; you have to define the instance as query parameter.
Michael: instance of LwM2M object is instance of YANG module

URI conversion. RESTCONF URI the same as with LwM2M. With CoMI key as query parameter.

YANG doesn't allow identifiers as integers; need to be strings. For hierarchy you use container in YANG. Felt it was not necessary. For composite versions may need it but seemed overkill.

Takeaways: example 2 seems best fit. YANG way more expressive than LwM2M. For simple IoT cases may be overkill. Much of noise, but would be good to have compressed versions. Would be good to have similar thing we have for documents with markdown and XML also for documents. We don't have "write only" and need to work on that. Would like to have automatic script ready in couple of months. At OMA meeting where some people doing this; maybe also at W3C?

Carsten: underlying question; why do we do modeling first place. Then to discuss where to do translation. In the example have range such as "10 .. 66.6", probably don't want it here. "Fraction-digits 2" also depends a lot on sensor.

Johannes: is in the typing system possible some of the objects to inherit things (like IPSO). E.g., turn on/off. Expect now that we'd have generic sensor object and inherit from there humidity. Is there way to inherit?
Jaime: In case of IPSO not sure we have that. You can take well known resources and plug them into device. Can take humidity sensor and then add new resources that fulfill your requirement. If use same resource IDs, device would know that you are sensor. No concept of inheritance. That's why using leafs/nodes in YANG.

Carsten: seems we have two systems developed to solve different problems. In IPSO resources composed to objects. Picked some object to have value and another to take min/max part. In YANG no type system but you describe data store. The data store description done to tree level -- some ways to define types -- but limited function in YANG and normally not used. Specify everything, so very verbose. But you see what you have.
Jaime: want to have discussion if such complexity is needed for IoT cases. Or to support legacy? Network oriented companies use YANG. Can use same system to manage these devices.
Matthias: understood that lots of people push this to reuse existing code. For LPWAN need to manage network. Traditionally YANG used. Re-use the code for application part. That's why map application concepts to YANG.
Jaime: looking inside feedback from YANG experts

Matthias: W3C WoT perspective; very repulsive for web developers. Want to enable web developers to build nice apps and killer apps. Being creative with this kind of things might be hard. Network management folks probably have tools, but it's different people writing apps.
Carsten: network management works with data structures that are highly standardized. App developer will not specify new network management concepts. Amount of pain for network management to tolerate is higher. OTOH, it's very explicit; you know what you have there and are about to standardize. Reasonable way to communicate during standards development process.

Matthias: do I have to use 64 bits for decimal?
Carsten: based on XML. No way for binary floating points. YANG has type for decimal64 that takes 64 bit integer and adds concept of fraction digits. Good example; in management you don't need floating point so it's not there.
Matthias: in embedded also often prefer fixed point
Jaime: you do have binary in YANG
Carsten: yes, for base64 encoded stuff

Carsten: should continue exploring LwM2M way of models -- lots of models that seem to work. Should also look at YANG and see how useful it is as model. Translation between the two sufficiently hard. Thanks for doing that.

Ari Keränen: Bluetooth URIs

Draft: https://tools.ietf.org/html/draft-bormann-t2trg-ble-uri-00
Slides:
https://www.ietf.org/proceedings/interim-2016-t2trg-04/slides/slides-interim-2016-t2trg-04-sessa-ble-uri-scheme-and-media-types-00.pdf

Does it makes sense to define URI schemes to identify local bluetooth resources?
Maxime: would that mean the same URI could identify two different resources on two different devices?
Carsten: sure, similar to the "file" URI scheme
Maxime: ok so not conforming to the Web architecture principles there
Matthias: WoT may want to identify specific resource on a device with URI. If same characteristics on two devices, might be at different places. Need discovery mechanism.

Matthias: right elements; resource would be specific attribute to specific device.
Johannes: without prior knowledge be able to access resource on a device
Carsten: not sure what BT SIG is doing on relying long-term stable identifiers. Need to do discovery process but get predictable URIs back that are not dependent on specific server, but scheme would define how server uses the URI.
Ari: so instead of BT master, the BT slave should be deciding the URI elements

Kaz: possibly need other scheme for secured version?
Ari: always using Bluetooth security
Matthias/Daniel L: also application security possible

Concluded that should look into using device specific identifiers in authority part of URIs.

Daniel Lux: Seluxit REST-ful open API for Lemonbeat devices

Slides:
https://www.ietf.org/proceedings/interim-2016-t2trg-04/slides/slides-interim-2016-t2trg-04-sessa-seluxit-rest-ful-open-api-for-lemonbeat-devices-00.pdf

Using heavily UUIDs and microservices. GW gets UUID during manufacturing, GW generates UUIDS for devices, and data items. Access control lists iptables style. JSON based APIs and XSD used to describe.

Johannes: devices appear under devices URI?
Daniel L: different GWs give different UUIDs to same device; "included" parameter to show if device is included.

Services have values. Control and reported values separated; can make one device change and other reading it. Know based on service grouping that they are together. Read-only service would have only "report" value. Data types strings and numbers. Boolean as number. Type system designed in 2006 but has held well since. Want to have "computer science types"; which the devices can easily interact with. Makes device-to-device interaction easier. Don't want single point of failure.

Matthias: were there something that needed workarounds?
Daniel L: maybe booleans have too much info. Integers are sent as floats. Due to implementation in device. Timestamp is 64 bit.

Daniel P: device needs to know how to handle e.g. number as boolean
Daniel L: only the configurator needs to understand that

Johannes: Similar type system in AR with numbers 0 to 1. Surprised that it hold for many use cases.Devices keep UUIDs after firmware update but change it if excluded from a network to keep data separate if device given to different user.

Carsten: where get the time?
Daniel L: each devices runs NTP. GW retrieves time from server. RF module from the GW. In tests between two devices, biggest difference was 50 usec over 24 hours.

Support for calculations like bigger than, smaller than, equal etc. State machine with actions and next state; actions can also be grouped.

Daniel P: if one action in group fails; what happens?
Daniel L: will get status report that an action in group failed, but after time next action happens.

Daniel P: is it possible to prioritize actions?
Daniel L: first come first served

Using collection of configuration data before pushing to server in order to save battery. Configuration change allows changing functionality by changing the state machine and rules.

Matthias: how generic is the creation of rules and state machines?
Daniel L: rules are done by programmer. Draw state machine and specify with Lemonbeat DSL.
Matthias: how would this look with our scripts? Still needs someone who thinks about this
Johannes: instead of actions you could do scripts
Daniel L: in cloud platform using javascript instead of the state machines

Matthias: in WoT could run something like this in Lua script.
Daniel L: can run the logic on server, GW or device. Always same logic and view.

Matthias: how to resolve UUIDs?
Daniel L: GW does translation. Lemonbeat device can send to IPv6 address. Routable back-bone. Configurations shown by GW are abstracted from the device config.

Carsten: useful to have some of these examples in the repository
Daniel L: can share these
(now available at https://github.com/t2trg/2016-09-w3c-wot/blob/master/slides/T2TRG_Lisbon_Seluxit_Open_API_sample.json)

Carsten Bormann: "type systems" impulse talk & discussion

Slides (slide 31…): https://www.ietf.org/proceedings/interim-2016-t2trg-04/slides/slides-interim-2016-t2trg-04-sessa-chair-slides-01.pdf

Data can be modeled at specification time or to control runtime behavior. We have choice of using self-describing data or have metadata separate. Data modeling enables code generation, conformance checking, semantics for run-time, specification time for humans.
Can be modeling data being interchanged (XML, JSON; syntax and semantics) or at rest (YANG). Hard to convert between the two kind models. Many "REST interfaces" work like using data being interchanged but really use data at rest. Languages optimized for humans, interchange formats for machines.
Modeling languages with different goals and theory behind; hard to impossible to translate between.
CDDL example for data interchange. At relatively high data model level. Abstraction based on JSON/CBOR. Production system based on tree grammars.
Model translation; what can be translated? At-rest != in-motion data; tree vs graph. Different expressive power.

Matthias: interested in nailing down what we want. Type system that connects different systems with different data models? Come up with common understanding what we want to have? Semantic interop at WoT with TD and express across different platforms what they provide. Structure of data in motion.

Johannes: pays out to be pragmatic. Most of the time simple structures. Most RESTful APIs have either very complex structures or very simple. Important how can we integrate in tools we have. Avoid need of developers to learn new tools etc.

Carsten: web developer needs to learn something. But maybe web devs not ones generating new models. For example ZigBee CL currently in PDFs. Would rather have in translatable format. IPSO going this direction. Something like Rosetta stone needed.

Kaz: data model at W3C; EMMA (data exchange format: https://www.w3.org/TR/2015/WD-emma20-20150908/) and SCXML (definition of state charts: https://www.w3.org/TR/2015/REC-scxml-20150901/).

Johannes: starting point could be landscape of available formats. Now been using XML schema, JSON schema, and CDDL.

Matthias: need to decide what we want to achieve. Have vague theory: with loosely coupled systems, don't care how it's implemented in other system. Need to map semantically individual elements to interchange format and map that to internal data store model. Must have notion of temperature if use that. Other system may have other data at rest structure, but don't want to have to learn about it. Semantic annotation for elements to copy to my model. Want to know atomically what is what. Be able to rearrange atoms in a model between models.

Johannes: evaluating type systems so far. Need to be able to generate URI. Need simple way to generate UI elements and validate data.

Matthias: don't want to care how many bits are used at the sensor in other side.
Carsten: problem is that need to make identifier decisions. Hard to ignore at data model. How are things named. Min/max humidity; which one is which. Need to manage the identifiers. Hard to abstract at serialization layer.
Matthias: have semantic annotation on each elements to achieve that.

Carsten: for example, with CBOR, you don't care value size at DM level.
Daniel P: XMLschema need use xsd-byte one byte. Handier to have simple levels or range.
Carsten: with XML can't even say something is number; xsd for that. xsd2 has weird stuff.
Matthias: xsd-byte etc. is baggage
Carsten: One reason for JSON vs. XML is that it makes you do less choices in data modeling.
Johannes: some cases where you want to control range (e.g. 0 .. 100%).
Daniel L: set-point for temperature control range could be 18-30 degrees celsius.
Johannes: UI widgets are very dependent on knowing values that make sense
Carsten: JSON has given set that work fairly well; missing binary blob.
Daniel P: still want something that can be used with XML; it has not disappeared yet
Carsten: it has some years ago in practice. JSON taken lead. Makes sense to have more than one serialization
Matthias: looking just JSON we are overlooking something

Johannes: with RAML tool used to have JSON and XML. Now simple schema language that generates JSON/XML. For data model using JSON. If can make that simple, it's good. Tend to build too sophisticated systems. Need to find narrow waste.
Matthias: want to know what are primitive types. Semantic annotations.
Daniel P: could try RAML in one of plugfests.
Matthias: don't have semantics. Idea of WoT to complement what is there with what is needed.

Daniel P: dangerous to create your own if you don't find one that fits your need perfectly
Carsten: agree with Daniel P's sentiment but Matthias may still be right (maybe we need +1 standard)
Matthias: good work done for WoT DM survey table but further work needed.
Daniel P: Need good list of requirements

Carsten: work tomorrow on requirements lists on a breakout? People could sketch something tonight.

Will have tomorrow morning security and some data modeling before lunch.

Michel: good to stay in type system discussion since that's where people stumble into.

Day ended (18:20)

Day 2

Daniel Lux: IoT Proxy scheme for secure constrained devices

Slides:
https://www.ietf.org/proceedings/interim-2016-t2trg-04/slides/slides-interim-2016-t2trg-04-sessa-iot-proxy-scheme-for-secure-constrained-devices-00.pdf

If you don't trust the cloud or GW, you don't have end-to-end security without object layer security. How to compact messages.
Homomorphic encryption allows basic operations on encrypted data. University project where looking into this.
Typically messages in a GW's queue. In end-to-end traditional security you can only pass all messages one by one. Communicating over GW could just get burst of on/off messages.
Looking into JSON Web Signatures (JWS) so GW can do smart things. Can e.g., take only the last value of a set of values to change resource or drop further attempts if first is rejected (DoS protection). Idea to have zero-config proxy which would not need pre-configuration but each packet would indicate if they can be compacted. Horizon 2020 proposal.

Matthias: Endpoints don't want to reveal too much to GW.
Daniel L: Content will be encrypted, but signature verifiable
Matthias: attacker can correlate messages etc.
Daniel L: Need some level of trust to infra. Can still give better e2e security. Today GW can normally read everything.
Matthias: PhD student working at ETH; integrated in LoRa. Would be interest to add object security in that use case. Today LPWAN approach HTTP-CoAP proxy seems nice. Others may

go to YANG models later. In WoT we have different transports already. Don't want to need model everything in YANG and a single protocol.

Carsten: have been talking of caching strategies for object security. Interesting how security associations are established. How proxy gets knowledge to do validation? How to maintain privacy? Very interested in caching strategies and interested to join.

Kaz: messages between proxy and end device; is possible to add priority etc data?
Daniel L: We'll end up with caching mechanism at object layer but security poses extra needs

Carsten: should be visiting this discussion in the future. Can't ignore the path to protect network (especially important for millibit networks). Want to support multiparty security associations? How to support with low overhead?

Aaron Yi Ding: Securebox and IoT research at TUM Connected Mobility

Slides:
https://www.ietf.org/proceedings/interim-2016-t2trg-04/slides/slides-interim-2016-t2trg-04-sessa-securebox-and-iot-research-at-tum-connected-mobility-00.pdf

Covering IoT research at TUM and securebox project. IoT testbed awarded for Google IoT research pilot award. Combining Google stuff with external devices like RPi. Trying to preserve privacy in IoT environment decentralized proximity detection.

Securebox addressing IoT security problems. Many IoT devices today not having security as high priority. Bringing SDN ideas to IoT. Instead of buying device and getting security from that; getting security as service from the securebox. Often security ends up not used/implemented due to resource (time, money, expertise) constraints. Tons of devices in different classes deployed to Internet with severe vulnerabilities. Sometimes hard coded credentials and no way to change them. IoT devices can today impact physical world (turn heat, open doors, etc.).

Securebox: cloud-assisted security service. "Charge for network service" model. IoT devices connected to RPi set top box and connected to SDN controller. Frontend metadata analyzed in backend (docker-based). Accepting / blocking based on analysis. Performance evaluation shows only small latency impact with securebox. First traffic flow has more impact but further flows can use local cached version.

Ongoing work to integrate e.g. with F-secure sense product.

Carsten: we have boxes that are run by external that secure the internals. Somehow need to be part of the whole security model. Can't rely on DPI because of encryption but don't want service providers have all data either.

Ari: mostly firewall kind of functionality in secure box?
Aaron: opening and blocking depends on analysing traffic patterns, where data uploaded to. Actions at frontend at the moment use firewall like functionality. Open to suggestions on what kind of things could be done beyond just blocking.

Matthias: how secure box differs from set of HW that looks into each step of my setup and data?
Aaron: want to have secure services easily deployed at backend. Users having interface to service and with reasonable price subscribe to service.
Matthias: you pointed out in analysis challenges, including that users don't want to send all data and behavior to the cloud. Securebox sends everything to cloud?
Aaron: depends how implemented. Now only sending first pieces of metadata. Don't do constant tunneling like the traditional services.
Matthias: metadata can be even more sensitive since you get automatically information who is talking to who etc.
Aaron: keeping the analysis and intrusion to minimal level, but open to suggestions how can be done better

Carsten: When designing protocols, we should be keeping in mind the kind of information we'd be willing to give path elements (firewalls etc.) and distinguish that from more privacy relevant information where we want proper e2e security.

Johannes: at Germany debate ongoing on about if you should be allowed to bring your own router to network.
Aaron: approaching ISPs about the business model. Want to do in RPi kind of device so that you can bring small device to network to bring security

Kaz: any idea between securebox and existing/expected automotive cloud. Almost all car vendors making their own secure cloud.
Aaron: securebox can be alternative besides car manufacturer secure services. Vendors often want to put everything in their ecosystem.

Carsten Bormann: "security models" impulse talk and discussion

Slides (slide 41...):
https://www.ietf.org/proceedings/interim-2016-t2trg-04/slides/slides-interim-2016-t2trg-04-sessa-chair-slides-01.pdf

Taking the data model and security discussions together. Presenting coffee mug and coffee machine scenario. Can we use IoT technology to make this more usable environment. Coffee mug and machine with NFC. Communication and negotiation of the two to fulfill each other's preferences and objectives. Mug can communicate to Internet via machine and NFC. Mug has been accredited.

Basic idea: CoAP POST with form and "make coffee" relationship. Security needs additional functionality: tokens to proof that payment has been done, mug is part of a group, perhaps that user is over 18 to add some rum on the coffee. Coffee machine does not need to know how (where) mug gets tokens, just to validate them. Authorization Managers (AM) for mug and machine: services behind them. Mug could use machine's Internet connection to talk to its AM.

Would like to ensure data modeling, metadata functions, and hypermedia controls can carry the security related information etc.

Johannes: in the scenario mug itself is quite active component. If the coffee machine is bigger node, would make sense that to access and handle the information.
Carsten: 15 years ago that would have been the answer; where coffee machine has access to all the information about the mug and then make coffee. Today we don't necessarily want to reveal all the information (e.g., age if not needed).
Johannes: There was Bosch platform presentation earlier at WoT where they care about user preferences, and things around me can ask for my preference. Service they are building and open sourcing. Could be related to what discussed here and worth comparing.
Carsten: In this example more than just preference; mug initiates a process. Key difference to classic car case where seat adjusts to preference is that in that case both endpoints belong to me. Here they don't.
Johannes: Also discussed at WoT encrypted tokens that intermediaries can't read

Kaz: NFC services popular in Japan. Two different interfaces; passive and active. Makes sense to think about these differences here.
Carsten: Here using NFC just as communication mechanism; not highly integrated as in credit card or mobile. Power can come e.g from wireless power that is getting today very convenient in this kind of cases.

Michael Koster: Data typing

Slides:
https://www.ietf.org/proceedings/interim-2016-t2trg-04/slides/slides-interim-2016-t2trg-04-sessa-data-type-system-for-iot-00.pdf

Should rather describe than tag data ("range of 0-255 with step of 1" instead of "uint8"). Overlap on access control and application semantics on can one read/write something. Using schemas to describe JSON structures may not scale well. Hypermedia controls and collections may work better. JSON schema define structures. RDF and schema.org define relationships between atoms; better for change. WoT framework MVP; REST and hypermedia make key building blocks, especially in the intersection of T2TRG and WoT. Prefer something what we have and simple; hypermedia controls etc, instead of heavy typing and schemas.

Carsten: for the coffee mug case; how to convey that information in the model?

Michael: agent can use the identity to arrange proper transactions; if one is paying coffee at machine, it does not even need to know identity. Machine just proxy using the knowledge of the preferences to make coffee.
Carsten: how do I ship all the data needed by coffee machine. How to convey the data? Tons of hypermedia controls needed to actuate?
Michael: presence and ID of the mug trigger the machine to do something. Machine becomes the proxy for you.

Johannes: coffee machine would not only need to know the data but also the structure
Michael: not sure if it matters where the data is stored
Matthias: how we model this is that we don't have fixed structures that have fixed elements. Would not scale with change. Would assume that coffee mug would have notion of "milk", "rum", "strength", but my preference understands "whisky". Still could use "milk" and "strength". Need flexible description where the fields are stored.

Michael: in one case coffee makers have to decide on schema that everyone is using. In the other case, we make granular hypermedia control that can describe coffee the features. Already know how to describe "strength" from many other domains.

Michael: server would not have to return everything that is available but only based on queries.

Daniel P: In the end need something that specifies how the data looks like
Matthias: application and server can share application level semantics.
Johannes: form would give you the structure
Matthias: description needs to tell if it's hex, float, something else
Daniel P: need both semantic level and the structure
Matthias: yes, we need nail down the blob structure
Michael: two ways to use form to brew coffee: set values of the attributes at the brewing machine. Attributes or parameters of the machine. Can do that with single payload. The packet triggers the brewing to start. Hypermedia controls can offer both types of interaction.
Matthias: with HTML forms the way is specified (URL encoding), but we need to define what is CBOR or BLE payload.
Michael: that's where schema.org schemas come to work. Coffee makers don't have to specify schema but just the vocabulary. Anyone involved in coffee can use the same vocabulary.

Matthias: need description how to pack this in representation format. Machine readable specification format for representations. In WoT server tells what it expects and client adapts.
Michael: just need a driver (ZigBee, Bluetooth..) to pack these structures.
Matthias: trying to achieve really loose coupling
Johannes: Always brownfield; need to be able to accommodate for that

Breakout: data models

Daniel P: can we use existing models or need to define new ones?
Matthias: looking at existing models we find useful features. Survey of what's out there and fitting to taxonomy important and useful. So far superficial. See what goals they are fulfilling and if they are important goals for us. Something we have overlooked?
Uday: need to look into requirements from existing things
Matthias: Need to have clear arguments why (if) we want something new. Also branding is important (c.f., "JSON schema" and "BSON").

Carsten: is it about data store, thing description, or interaction?
Matthias: bridging between data in rest and motion. Linked data for example. How do we get from concept to payload?
Daniel L: At system level you have different requirements for data ("device does not need to know if it's in/out-door"). Divide and conquer. Start with device and make it as simple as possible. Then go to system of devices, etc.
Matthias: With linked data we have powerful mechanism. With specific product you define the vocabulary.
Daniel L: hopefully don't have to define much/any but can use existing semantics
Matthias: metamodeling and creating instances
Daniel L: Very hard to start from the high level modeling. We have useful tools already; linked data, (many) serialization format.
Matthias: good check to do if one can generate user interface from it; is it intuitive?

Matthias: Alternatives to linked data; hay stack, OPC-UA. Can't know if two concepts are related. Need the linked data step.
Daniel L: need also typing; celsius/fahrenheit. Well-known tags giving meaning. How is that conveyed; numbers/strings?
Matthias: Can we write SenML spec in machine readable form? Convert simplest SenML another format?
Ari: should be possible by linking the vocabulary of SenML to existing ontology

Carsten: mapping units is relatively easy; context is harder. Outside, balcony, terrace, etc.
Michael: working that for vehicles at W3C Genivi. Overall sounds like schema-org type activity
Carsten: names in SenML are situated names. Unless no specific knowledge of context, hard to use that.
Kaz: automotive decided to refer to Genivi definitions

Automotive WG wiki on GENIVI's VSS (Vehicle Signal Specification):
https://www.w3.org/auto/wg/wiki/Vehicle_Information_Service_Specification#VSS_Tree

Michael: link format used to give context for SenML in HSML. Each SenML item is resource. SenML way to transfer collections.

Carsten: modeling languages we have define structures. But need mechanical way of building structures from data. Things that are rooted in semantics.
Michael: if you understand semantics well enough, structure is not important.

Matthias: want to easily install en/decoders to systems.
Michael: no easy way to go from OCF model to SenML; goes to collection. Need normalization across models.
Matthias: useful to have way to describe normalization of specific format; represent in normal way. Converting from one format to another only common elements survive and you lose information. Need to identify concepts on both sides and map to interchange format.
Michael: identifying concepts probably as much work as doing the mapping

Carsten: For example, if Bluetooth body composition service can be mapped to another format; we are probably close to there (see: https://www.bluetooth.com/specifications/gatt/viewer?attributeXmlFile=org.bluetooth.service.body_composition.xml). Lots of domain knowledge that would need to survive normalization.

Matthias: someone who created this would need to explain what is what
Michael: at OCF dealing with this. Health stuff defined IEEE 11073. "We just need to match them up"; but hard to do in practice.
Matthias: want to create mashups; and support more systems than the original ones. Does not scale in the end. Use body composition measurement to control AC? Want to pick and use useful parts and use in another domain. Where in the blob of data is the useful info.
Michael: At 11073 they don't have body temperature, but armpit, under tongue, etc. Can't map to ontology that understands "temperature" alone.
Matthias: that's where bridging ontology comes along.
Michael: no way machine can do this. Hard to identify what things match.
Matthias: but only have to do it once. Classic example of going from tags to linked data structures.
Michael: go from tags to RDF system and ontology.
Matthias: missing piece: one side can create BLE message that matches the concept and can deliver to me. I can unpack it and use it. Say, linked data vocabulary for BLE. Need to know how to describe the messages.
Johannes: JSON-LD vocabulary easy part. How do we write something that works for JSON/XML/binary. Have plug-in replace with something that works with JSON/binary.
Matthias: Could have TD with BLE URI and property for mass of some sort. In the "data type" need to have description of how the data looks like.
Johannes: current TD says it's XML/JSON and then have JSON schema.
Matthias: that's why JSON schema doesn't work and we need something else
Johannes: how do we use the data may not be in the same description file

Matthias: when getting a blob, need to know how to normalize the value ("multiply by 200" like in BT case). That description needs to be there.
Johannes: one piece is translating the data and another what does the data mean.
Matthias: both context/application semantic information and then units/value etc. Two different things. Normalization rules for data needs to be in semantic description.

Michael: what is the assumption on what software will receive? Always binary packet? Bridges do transforming in ZigBee for example.
Johannes: from normal format to have adaptors to specific formats. How to define the adaptors?
Carsten: don't want to define language for conversions; would need Turing-complete language.
Matthias: parts could be done with TDs
Carsten: would write DSL and code that interprets DSL. Do we want to do that?
Michael: needs to be done by someone
Ari: maybe subset doable? For example bit fields and simple arithmetic?
Carsten: for example user ID in BLE. How to map to context information?
Matthias: that part would be in application logic / script. When writing mashup, should not need to read specification to know how to use data. Normalized form of units etc. BLE good way to check if we have powerful-enough ways to describe

Templates and decoders.

Carsten: thermometer does not need to know if it's inside or outside; that is context.

Three aspects for the modeling mechanism:

- Application semantics: atomic concepts that exist on each device/service
- Interchange format structure
- encoding / extract & transform (ETL): convert it to your local representation

Helps with automatic normalization

Organization/scheduling of semantic reasoning etc. to not drown in sea of facts:

- Similar to splitting into Abox and Tbox
- Work on Tbox at configuration time (bridge vocabularies, offline/big machine reasoning)
- Work on Abox just in time (filter during discovery, limit to facts known by application logic, ignore rest that is floating in the sea)
- Offload reasoning from constrained devices to a reasoning service (e.g., convert exotic units to something implemented on the device -> include service in mashup or even download script for WoT runtime to instantiate local microservice)

Carsten: Calculation language. A minimal form of mobile code that could be run on very constrained devices. Cf. OCF derived models -- manual process, look at the comments and go implementing (from OIC, to OIC)

Examples for something more complex than conversion: resolve a user ID to a person.

Concluding

- Looked at number of models. Normalization needs more work.
- Discussed translating data; translating between models still not tackled (in detail), but maybe that's not something we actually want to do -- instead we'll try to extract schema.org-style properties from all of them.
- Might make sense to make normalized data model abstract, that is, it is never instantiated, because it would require a too large number of features (superset of all existing data models); just enable 1:1 conversions through a normalized description mechanism
- Use bottom-up semantics; start with schema.org style, devise how to bag concepts later

For Bluetooth URI could be useful to have way to address specific bits (inside attributes, handles?) with URIs.

Instead of model translation, we might devise methods to upload information from specific models to schema.org.

Michael has mapped smartthings models to schema.org-style properties.

Action plan:

- Trying to model some of the things in BLE and Smartthings as RDF
- Model and enable conversion of the domain specific units (e.g., BLE kilogram)
- Idea for instructions for message parsers (how to find the right values in the blobs). **Selectors**, e.g. from JSON structure (e.g., via RFC 6901 JSON pointer), BLE structure, etc.

Michael will work on RDF for Smartthings and ZCL stuff. Carsten will work on conversion of units (ZCL has funny stuff, too).

Seoul topics:

- Checkpoint on the above actions
- WoT vs. T2TRG -- what gets done where?
  - Modeling is more of a W3C topic (semantic Web etc.)
  - Protocol work is IETF, in particular Security
  - For example, PoP structures (ACE) for OT (e.g., Profinet)
  - TD elements (AoT) + problem responses and other steps (JIT) for setting up security
  - Intermediaries as legitimate components of a security architecture