W3C WoT IG
Sunnyvale face to face meeting

TF-AP discussion digest

Breakout introduction of
Task force on API and protocol mapping
by Johannes Hund
(johannes.hund@siemens.com)

Agenda points for the break-out

- Recap of discussion / sync
- Report from joint F2F with IRTF T2T pRG
- Joint discussion of security&privacy topics
- AP Discussion topics:
  - Resource model (Review, prepare consensus)
  - Architecture (Merging and prepare consensus)
  - Tech landscape (Structure, Contributions, Caretakers)
  - Example protocol mappings
  - White gaps (Brainstorming, Filtering, Detailing)
  - Next Steps

Short Recap / Sync

Discussion topics of TF-AP

- Abstract Resource Model
  - Protocol-agnostic typed resources for web things: Properties, Actions and Events/Subscriptions
- Architecture Model
  - Reach consensus for an architecture model
- Technology Landscape
  - Relevant IoT resp. Web protocols and technologies
- Use cases & Requirements
  - Collection and dissemination of IoT scenarios

Abstract Resource Model

Consensus proposal

Protocol-agnostic resource model for web things

- Properties
  - Dynamic properties of the Thing
  - Static properties: metadata should be in Thing Description with no own runtime rep
- Actions
  - Invocable actions on a thing
  - May or may not result in state change
- Subscriptions/Event Sources
  - Intention to be notified on a certain condition
  - Including a method to avoid subscriptions on static properties

Things can be exposed through several endpoints
Chosen resp. „negotiated“.

Properties

- Read-only/read-write Properties
  - Read-only data
  - scalar or lists/ structured types
- Configuration
  - Read-write / possibly CRUD
- Dynamic
  - continuous timeline of value changes (Streams)
  - Spontaneous events

Streams: filtered read, timeline
Subscribing to a value could be seen as a special case of an event (scripting, value constrains)

Actions

- Invocable action on the physical thing.
- Retrieve a description, invoke execution
- Possibly manage running execution
- Can or cannot issue a state change
- Enables:
  - atomic change of multiple resources
  - Long-running executions
  - Semaphores
- Consider: scalability, security, robustness

Events

- Spontaneously emitted event
- Can be subscribed to resp. observed
- Subscription handling on own resource
- Enables:
  - Subscription of several resources (aggregation)
  - Events not related to state change

Discussion

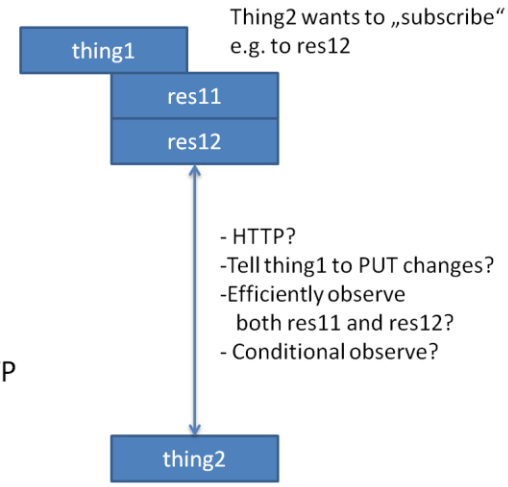
- Do we have the same understanding?
- Differentiation between property change and event
  - History / timeline vs. spontaneous nature?
  - Resources should always be pollable
- (Relation to e.g. CoRE interfaces)
- Media types / Datastructures (link to TF-TD)

Subscription resource

A Proposal to fill the white gap
protocol-independent subscriptions

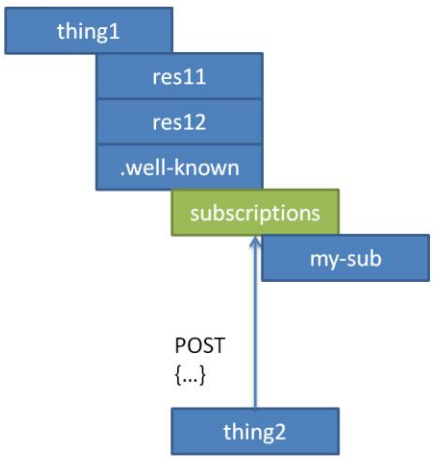
Problem: RESTful but protocol-agnostic subscription

- Observe works for single-resource in coap
- How to:
  - Observe if threshold is exceeded
  - Observe multiple resources
  - Set max frequency
  - Be compatible with HTTP
  - Use reversed approach (á la PubSubHubbub)



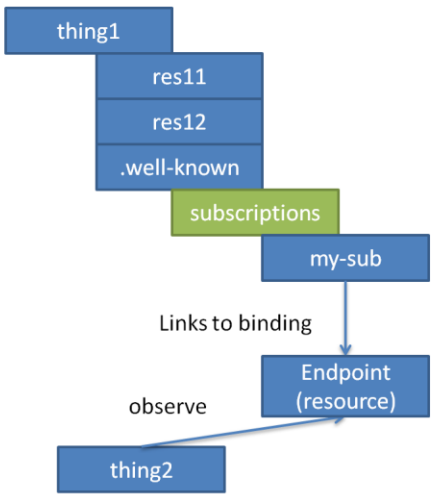
Solution sketch

- Define a media type and/or resource interface for managing subscriptions
- Comprises:
  - Links to observed resource(s)
  - Subscription constraints (threshold, frequency, sync of events, higher semantics)
  - Endpoints for subscription



Dataflow

- Subscription resource links to endpoint
- Subscription resource is protocol-agnostic
- Endpoint is protocol-specific



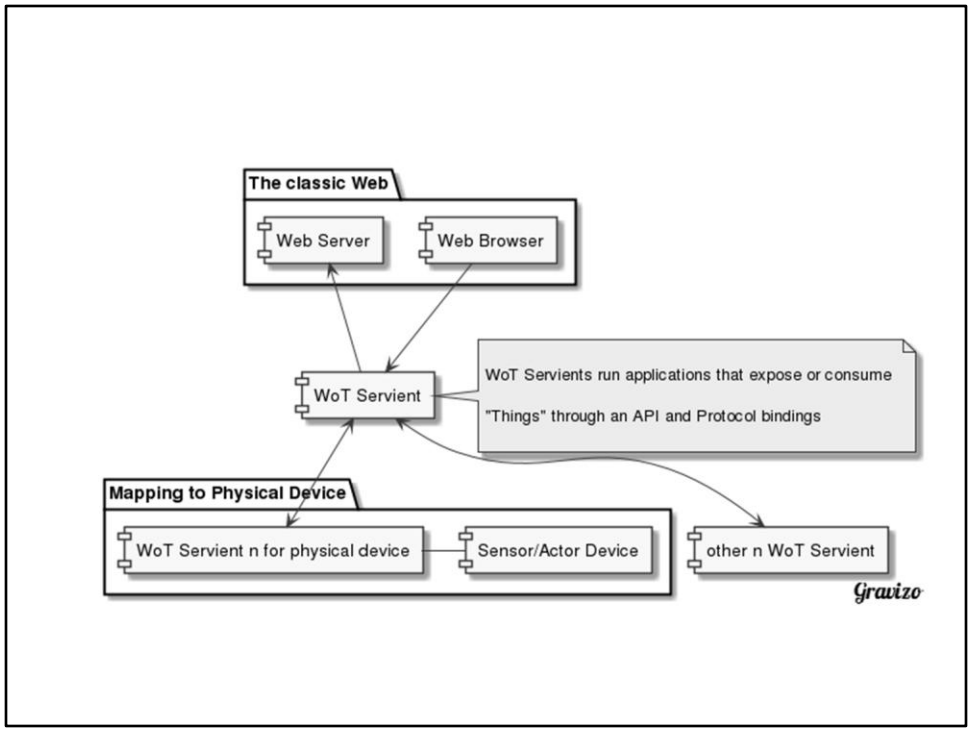
-Notes:

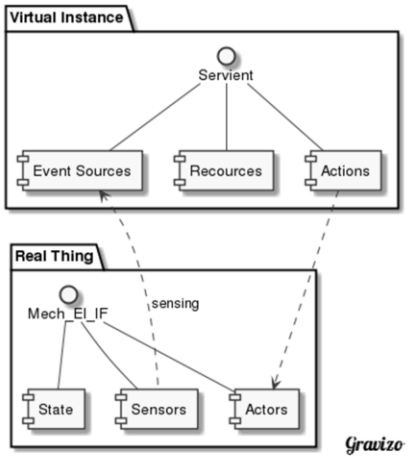
-Who is subscribing? (DOS prevention, physical resource management)

-Possibly LWM2M, Oauth token?

Architecture Model

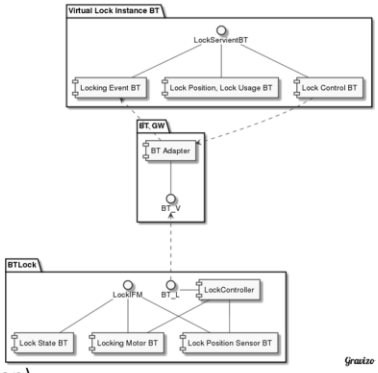
Recap and consensus proposal


The classic Web
Web Server
Web Browser
WoT Servient
WoT Servients run applications that expose or consume
"Things" through an API and Protocol bindings
Mapping to Physical Device
WoT Servient n for physical device
Sensor/Actor Device
other n WoT Servient
Gravizo



Assumptions on Virtual Instance:

- Virtual Instance has actions, resources, and event sources.
- Actions, resources, and event sources contain APIs of a thing and code that how to actuate or sense a thing including protocol binding.
- A Virtual Instance ties to a real thing.
- More than one Virtual Instances can be allocated to a WoT Servient.
- Virtual Instance is instantiated by WoT Servient from script or code (i.e. Device Driver).
- A real thing can have more than one virtual instance representing it

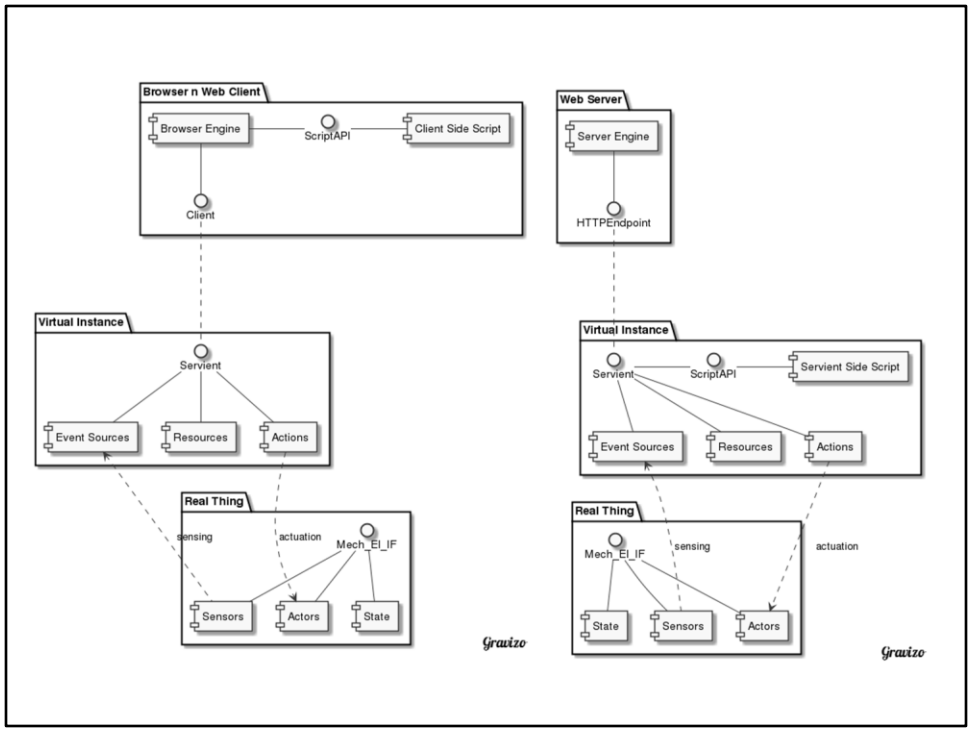


Behavior of WoT Servient

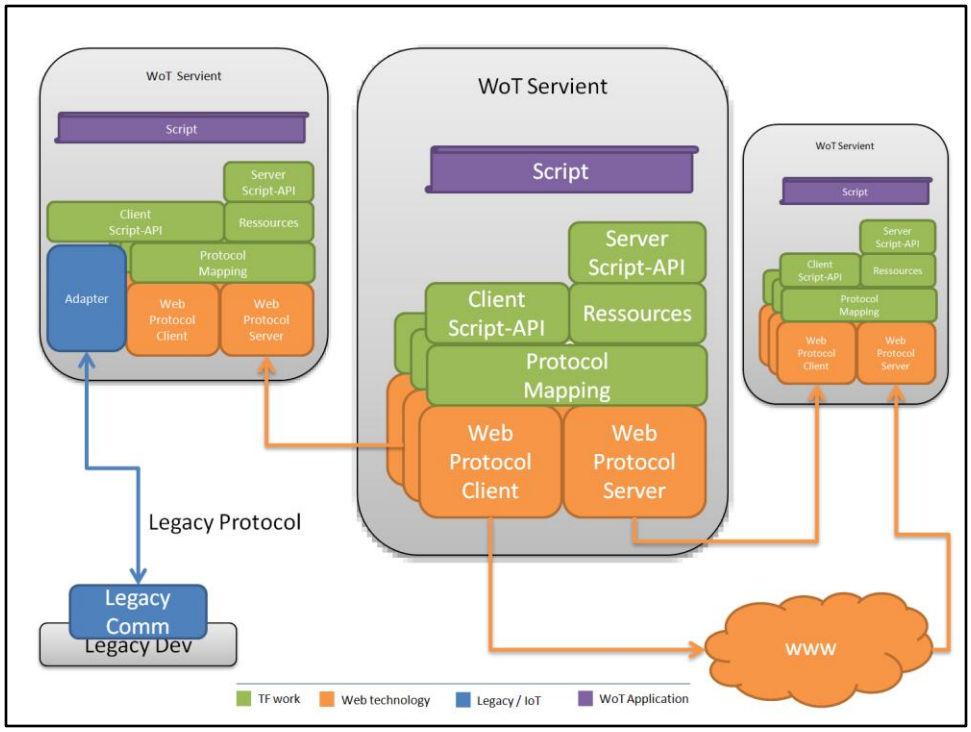
- WoT Servient gets a device profile on discovery.
- WoT Servient invokes a virtual instance
  and installs a Device Driver (Thing Description)
  that corresponds to the device profile.
- WoT Servient exposes APIs of the virtual instance.

Assumptions of BT GW

- BT GW has a media conversion capability such as BLE or Bluetooth 2.1.
- BT GW accommodates an API
  such as GATT API, WebBluetooth, or HDP(Health Device Profile)
  in order to expose functions of a real thing.
- WoT Servient binds the real thing and the Virtual Instance through the BT GW.



Server-side: expose inputs and outputs as resouces
Needs databinding of resources



Orange vs. Blue – when is a protocol considered a web protocol
Add constrained device: One protocol, fixed ressources, no APIs
What is the minimal servient?
Add one servient hosting several virtual instances
Add services in the cloud

Tech landscape

Status and next steps

Technology landscape of TF-AP

- Protocols
  - Protocols that allow to map the abstract model
  - Protocols that can be generically adapted
  - Legacy protocols
- Resource Models
  - Common resource models
  - Models from consortia or domains
- API patterns
  - Patterns used in scripting APIs that interface the web

Security & Privacy topics for joint meeting

What are the common points to discuss between S&P and TF-AP?

Lifecycle states of a WOT Servient

What are the states and transitions we face?

States

- Offline
  - Not in a network
- Online
  - In a network, but not in the WoT
- Registered / paired
  - Paired with a backend / a WoT device
- Activated
  - Active connection and control