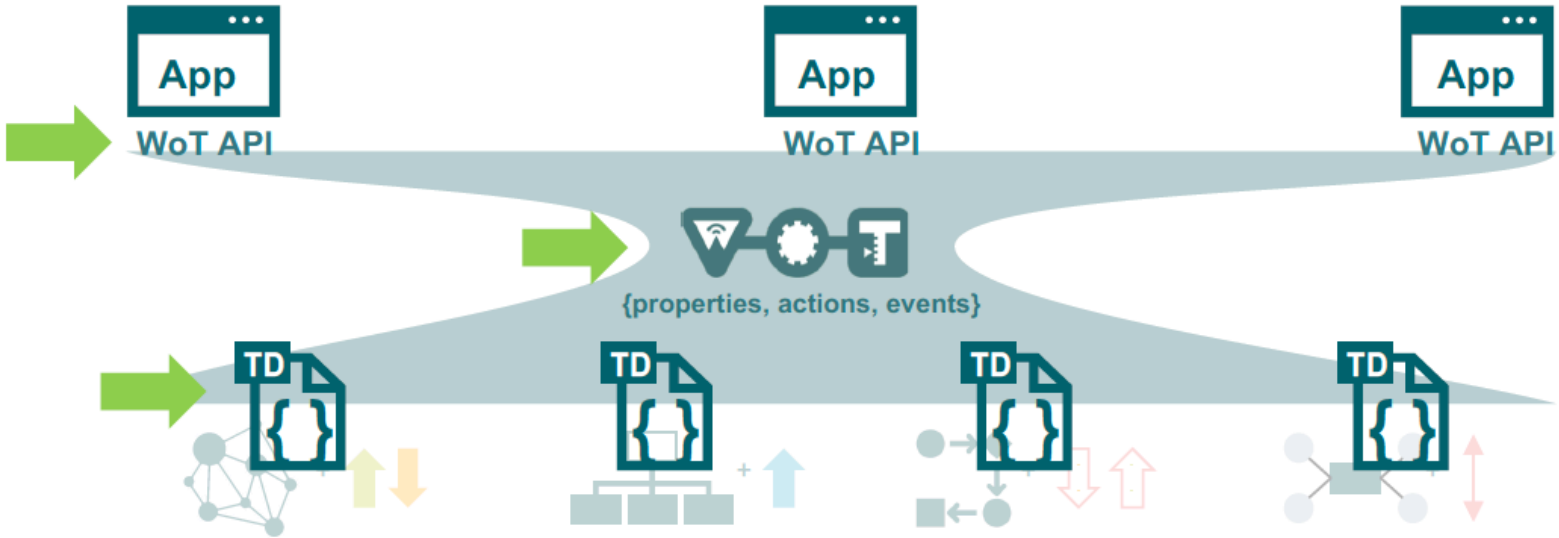
WoT Intro and Update

Michael McCool, Intel

W3C Web of Things (WoT)

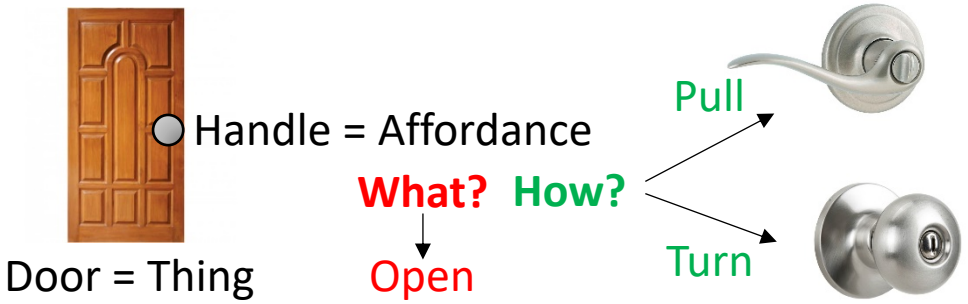
- W3C Working Group: goal is adapting web technologies to IoT
- Recently published: standard Thing Description (TD) metadata format
  - TD describes the available interactions (network API) of a Thing
- New standards work includes discovery
  - How does a potential client obtain the TDs for a Thing?



WoT Descriptive Interoperability

WoT Architecture

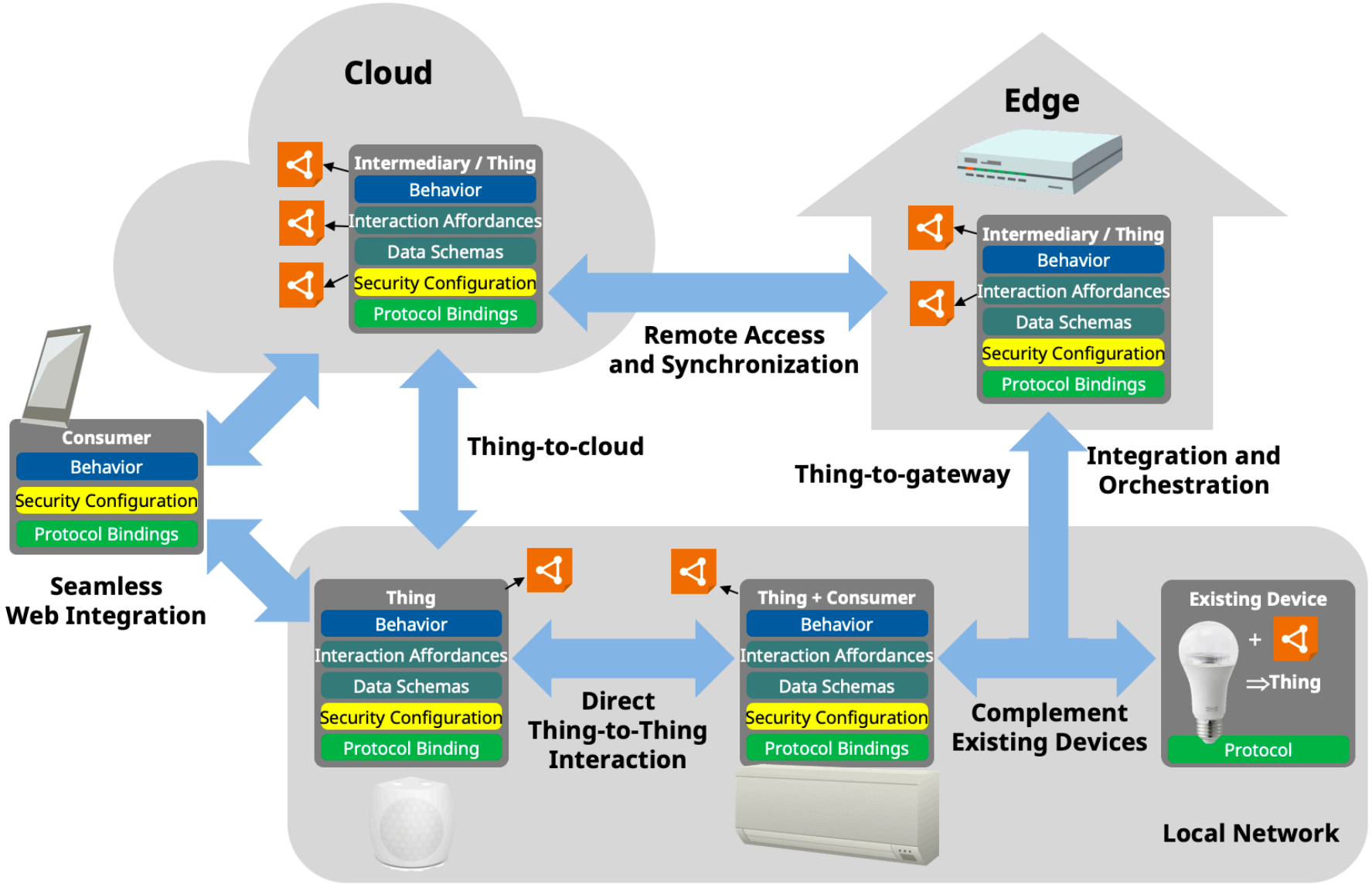
- Constraints
  - Things must have TD (W3C WoT)
  - Must use hypermedia controls (general WoT)
  - URIs, standard set of methods, media types
- Interaction Affordances
  - TD describes the possible choices (what) to Consumers, suggests how Consumers may interact with the Thing



WoT Thing Description (TD)

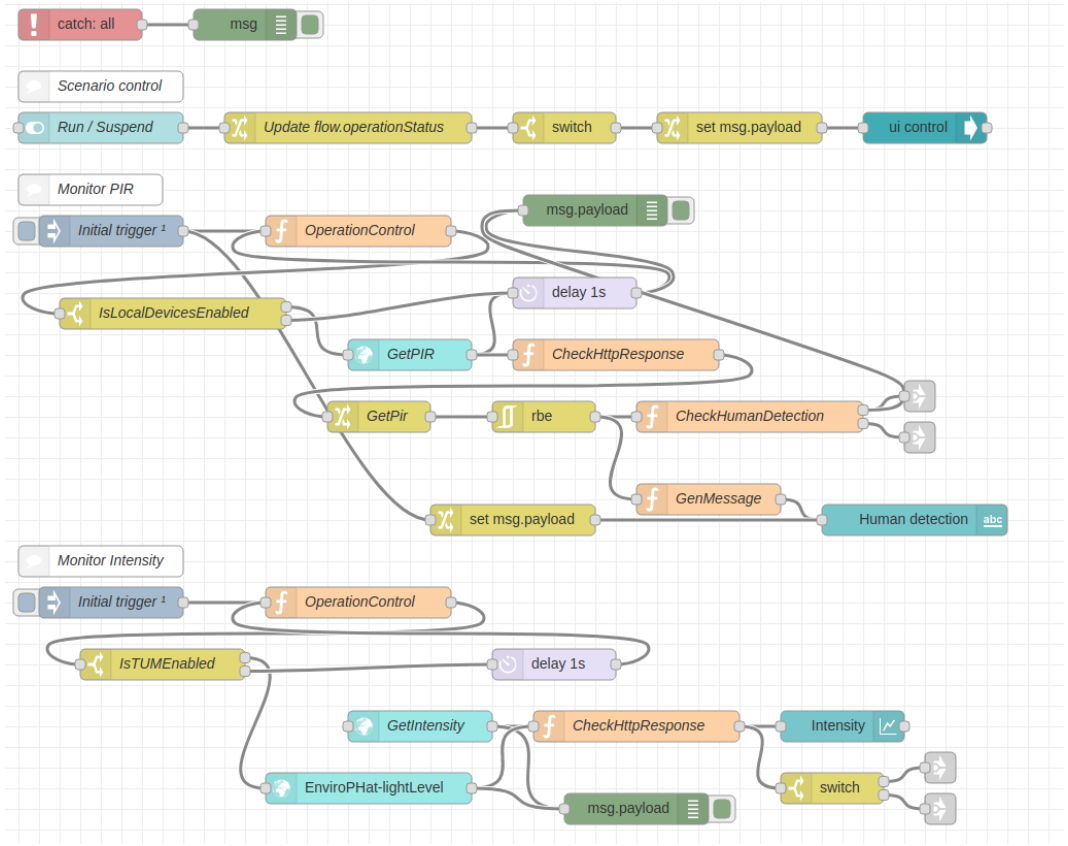
```
{
  "@context": [
    "https://www.w3.org/2019/wot/td/v1",
    { "iot": "http://iotschema.org/" }
  ],
  "id": "urn:dev:org:32473:1234567890",
  "title": "MyLEDThing",
  "description": "RGB LED torchiere",
  "@type": ["Thing", "iot:Light"],
  "securityDefinitions": ["default": {
    "scheme": "bearer"
  }],
  "security": ["default"],
  "properties": {
    "brightness": {
      "@type": ["iot:Brightness"],
      "type": "integer",
      "minimum": 0,
      "maximum": 100,
      "forms": [ ... ]
    }
  },
  "actions": {
    "fadeIn": {
      ...
```

Use Case Overview



WoT Orchestration

Node-RED/node-gen



WoT TD benefit: Module autopopulation

node-wot/Scripting API

```
WoTHelpers.fetch( "coap://localhost:5683/counter" ).then( async (td) => {
 // using await for serial execution (note 'async' in then() of fetch())
 try {
  let thing = await WoT.consume(td);
  console.info( "=== TD ===" );
  console.info(td);
  console.info( "==========" );

  // read property #1
  let read1 = await thing.readProperty( "count" );
  console.info( "count value is" , read1);

  // increment property #1 (without step)
  await thing.invokeAction( "increment" );
  let inc1 = await thing.readProperty( "count" );
  console.info( "count value after increment #1 is" , inc1);

  // increment property #2 (with step)
  await thing.invokeAction( "increment" , {'step' : 3});
  let inc2 = await thing.readProperty( "count" );
  console.info( "count value after increment #2 (with step 3) is" , inc2);

  // decrement property
  await thing.invokeAction( "decrement" );
  let dec1 = await thing.readProperty( "count" );
  console.info( "count value after decrement is" , dec1);

 } catch(err) {
  console.error( "Script error:" , err);
 }

}).catch( (err) => { console.error( "Fetch error:" , err); });
```

New WoT WG Charter Work Items

Architectural Requirements, Use Cases, and Vocabulary

- Understand and state requirements for new use cases, architectural patterns, and concepts.

Link Relation Types:

- Definition of specific link relation types for specific relationships.

Observe Defaults:

- For protocols such as HTTP where multiple ways to implement "observe" is possible, define a default.

Implementation View Spec:

- More fully define details of implementations.

Interoperability Profiles:

- Support plug-and-play interoperabilty via a profile mechanism
- Define profiles that allow for finite implementability

Thing Description Templates:

- Define how Thing Descriptions can defined in a modular way.

Complex Interactions:

- Document how complex interactions can be supported via hypermedia controls.

Discovery:

- Define how Things are discovered in both local and global contexts and Thing Descriptions are distributed.

Identifier Management:

- Mitigate privacy risks by defining how identifiers are managed and updated.

Security Schemes:

- Vocabulary for new security schemes supporting targeted protocols and use cases.

Thing Description Vocabulary:

- Extensions to Thing Description vocabulary definitions.

Protocol Vocabulary and Bindings:

- Extensions to protocol vocabulary definitions and protocol bindings.

W3C WoT Resources

- W3C WoT Wiki
  - https://www.w3.org/WoT/IG/wiki
    (IG/WG organizational information)
- W3C WoT Interest Group
  - https://www.w3.org/2016/07/wot-ig-charter.html
    (old charter)
  - https://www.w3.org/2019/10/wot-ig-2019.html
    (new charter)
  - https://lists.w3.org/Archives/Public/public-wot-ig/
    (mailing list)
  - https://github.com/w3c/wot
    (technical proposals)
- W3C WoT Working Group
  - https://www.w3.org/2016/12/wot-wg-2016.html
    (old charter)
  - https://www.w3.org/2020/01/wot-wg-charter.html
    (new charter)
  - https://www.w3.org/WoT/WG/
    (dashboard)
- W3C WoT Candidate Recommendations
  - https://www.w3.org/TR/wot-architecture/
  - https://www.w3.org/TR/wot-thing-description/
- W3C WoT Working Drafts / Group Notes
  - https://www.w3.org/TR/wot-binding-templates/
  - https://www.w3.org/TR/wot-scripting-api/
  - https://www.w3.org/TR/wot-security/
- W3C WoT Editors' Drafts and Issue Tracker
  - https://github.com/w3c/wot-architecture/
  - https://github.com/w3c/wot-thing-description/
  - https://github.com/w3c/wot-binding-templates/
  - https://github.com/w3c/wot-scripting-api/
  - https://github.com/w3c/wot-security/
  - https://github.com/w3c/wot-security-best-practices/
  - https://github.com/w3c/wot-profile/
- Reference Implementations and Tools: node-wot
  - node-wot: https://github.com/eclipse/thingweb.node-wot
  - TD playground: https://github.com/thingweb/thingweb-playground

Contacts

https://www.w3.org/WoT/WG/

Dr. Michael McCool
Principal Engineer

Intel
Technology Pathfinding

michael.mccool@intel.com

Dr. Sebastian Kaebisch
Research Scientist

Siemens
Corporate Technology

sebastian.kaebisch@siemens.com