T2TRG: Thing-to-Thing Research Group

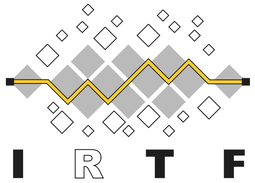
pre-IETF 111 “Summer” Summary Meeting, June 21, 2021

Chairs: Carsten Bormann & Ari Keränen

Note Well

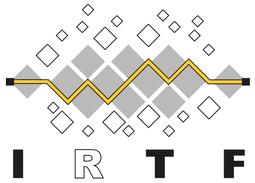
- You may be recorded
- Be nice
- The IPR guidelines of the IETF apply:
  see **http://irtf.org/ipr** for details.

Note Well – Intellectual Property



- **The IRTF follows the IETF Intellectual Property Rights (IPR) disclosure rules**
- By participating in the IRTF, you agree to follow IRTF processes and policies:
  - If you are aware that any IRTF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion
  - The IRTF expects that you file such IPR disclosures in a timely manner – in a period measured in days or weeks, not months
  - The IRTF prefers that the most liberal licensing terms possible are made available for IRTF Stream documents – see RFC 5743
  - Definitive information is in RFC 5378 (Copyright) and RFC 8179 (Patents, Participation), substituting IRTF for IETF, and at https://irtf.org/policies/ipr

Note Well – Privacy & Code of Conduct



- As a participant in, or attendee to, any IRTF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public
- Personal information that you provide to IRTF will be handled in accordance with the Privacy Policy at https://www.ietf.org/privacy-policy/
- As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (https://www.ietf.org/contact/ombudsteam/) if you have questions or concerns about this
- See RFC 7154 (Code of Conduct) and RFC 7776 (Anti-Harassment Procedures), which also apply to IRTF

Goals of the IRTF

- The Internet Research Task Force (IRTF) focuses on longer term research issues related to the Internet while the parallel organisation, the IETF, focuses on shorter term issues of engineering and standards making

- **The IRTF conducts research; it is not a standards development organisation**

- While the IRTF can publish informational or experimental documents in the RFC series, its primary goal is to promote development of research collaboration and teamwork in exploring research issues related to Internet protocols, applications, architecture, and technology
- See “An IRTF Primer for IETF Participants” – RFC 7418

Administrivia (I)

- (Blue sheets maintained by meetecho)
- Note-takers: **https://codimd.ietf.org/notes-ietf-interim-2021-t2trg-01-t2trg**
- Jabber (= Meetecho chat)
  - **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/2021-06-summary**

Agenda

Time (UTC)	Who	Subject	Docs
15:00	Chairs	Intro, RG status, upcoming meetings and activities	draft-irtf-t2trg-rest-iot
15:10	Chairs	Reports from WISHI and other activities	
15:30	Michael McCool	W3C WoT update	
15:50	Michael Koster	OneDM update	
16:10	Xavier de Foy	IoT Edge Challenges and Functions	draft-irtf-t2trg-iot-edge-02
16:25	Mohit Sethi, Michael Richardson	Initial Security Setup	draft-irtf-t2trg-secure-bootstrapping draft-richardson-t2trg-idevid-considerations-03
16:55	Chairs	Wrap-up	
17:00	Chairs	end of meeting	

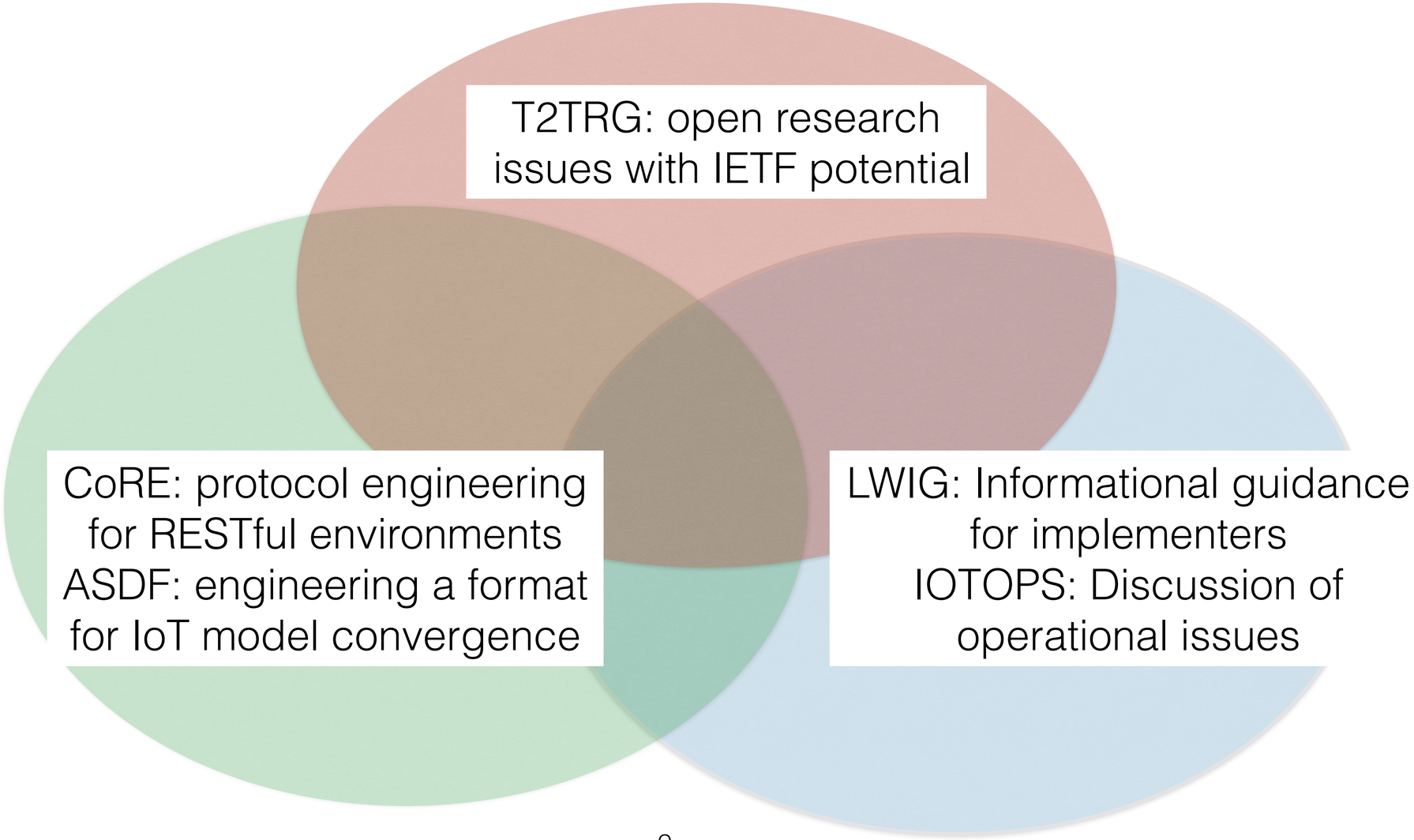
T2TRG scope & goals

- 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

IRTF
(Research)

IETF
(Engineering)



Next meetings

- Regular WISHI calls (~ monthly, resuming early September)
- (No T2TRG meeting at IETF 111, but:) WISHI Hackathon week July 19–23
- Online meetings with OCF / OMA SpecWorks (LwM2M&IPSO)/W3C WoT?
- “Online co-locating” with academic conferences? (Stay tuned!)
- Physical meeting before IETF 112 (Madrid)?

RG Doc Status

- “RESTful Design for IoT” (PR in progress, author/interest team meeting RSN)
- Edge & IoT (discuss today, getting ready for RG last-call) 2021ish
- Secure Bootstrapping for IoT (discuss today, potential rename) 2021ish
- (Related: discuss draft-richardson-t2trg-idevid-considerations today) 2022ish
- Ramping up: IoT Information-Model Standards Description and related work on Semantic Landscape/Nutrition Labels in WISHI
- Also: WISHI notes (see WISHI wiki, e.g. terminology rosetta stone)

Recent Work on IoT Semantic/Hypermedia Interoperability (WISHI)

- Follow-up on Azure Digital Twins Definition Language (DTDL) and IETF SDF
  - SDF & DTDL enhancements for improved expressiveness and interwork
  - Using external ontologies with SDF & DTDL
  - Further alignment on use of units
  - SDF - DTDL conversion implementations
  - Sharing and converting models
- SDF – YANG conversions sdf-yang-converter.org
- "IoT Information-Model Standards Description"
  - T2TRG draft forthcoming

Work on IoT Semantic/Hypermedia Interoperability (WISHI)

- Next WISHI: around September
- Continue work on SDF↔DTDL, SDF↔YANG conversion
- Discussion about W3C WoT Thing Models
- Attaching additional information to SDF models:
  “Mapping files for SDF”, instance vs. class, composition

SDF - YANG

Jana Kiesewalter

SDF-YANG-Converter

2021-06-21

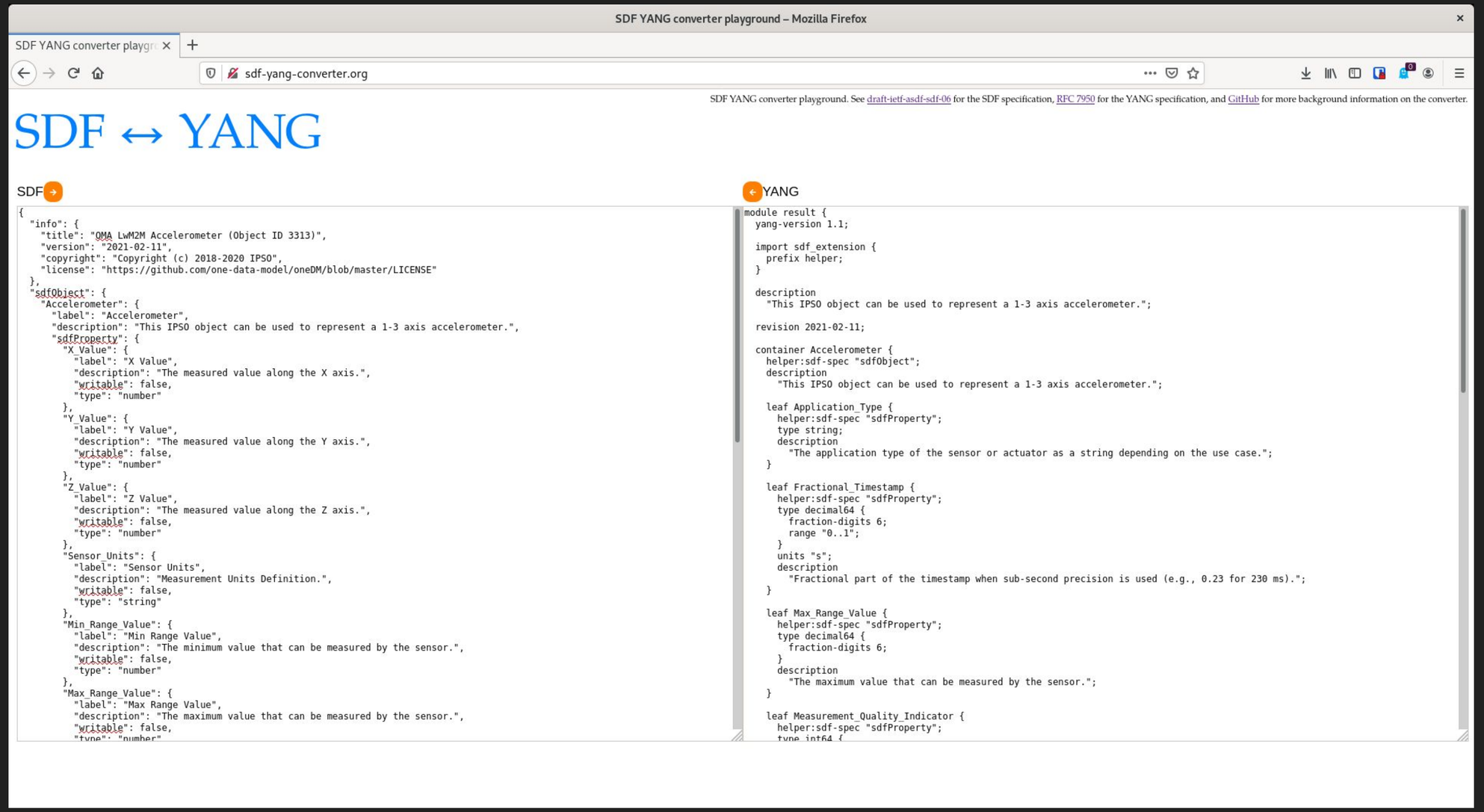
Mapping YANG ➡ SDF (selection)

YANG	➡ SDF
module	SDF model (i.e. info block, namespace section & definitions)
container	sdfObject (top-level container) | sdfProperty of compound-type (container that is a child node of top-level container) | property of a compound-type element (container on any other level)
list	sdfProperty of type *array*, items of type *object* (top-level list or one level below) | property of a compound-type element of type *array*, items of type *object* (any other level)
leaf-list	sdfProperty of type *array*, items of simple types (top-level leaf-list or one level below) | property of a compound-type element of type *array*, items of simple types (any other level)
leaf	sdfProperty of a simple type (top-level leaf or one level below) | property of a compound-type element of a simple type (any other level)
typedef	sdfData of a simple type
grouping	sdfData of compound-type

Mapping SDF ➡ YANG (selection)

SDF	**➡ YANG**
SDF model (i.e. info block, namespace section & definitions)	module
sdfThing | sdfObject	container
sdfProperty	Container (sdfProperty of compound-type) | Leaf (sdfProperty of simple types) | Leaflist (sdfProperty of type *array* with items of simple types) | List (sdfProperty of type *array* with items of compound-type)
sdfAction	RPC with input/output nodes for sdfInputData/sdfOutputData
sdfEvent	notification with nodes for sdfOutputData
sdfData	Grouping (sdfData of compound-type) | Typedef (sdfData of simple types) | Grouping mit Leaflist (sdfData of type *array* with items of simple types) | Grouping mit List (sdfData of type *array* with items of compound-type)

Converter Demo at sdf-yang-converter.org



ASDF/WISHI Hackathon Week

- 2021-07-19..-23, starting with WISHI call on 2021-07-19 (1400Z?)
  (Week before IETF111 ➔ register for hackathon ($0 and get a T-Shirt :-)
- Continue work on SDF ↔ other converters, such as:
  - DTDL converter (http://wishi.nomadiclab.com:8083/odm2dtdl)
  - sdf-yang-converter.org
  - WoT TD
- continue development of the "mapping file" concept

W3C Update

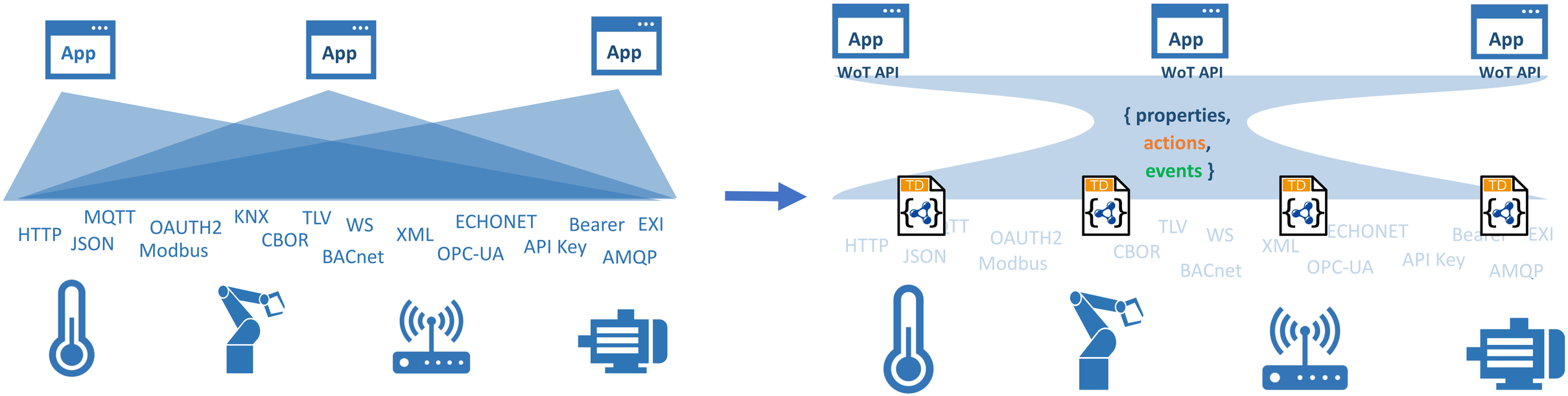
Web of Things Update

Michael McCool

June 2021

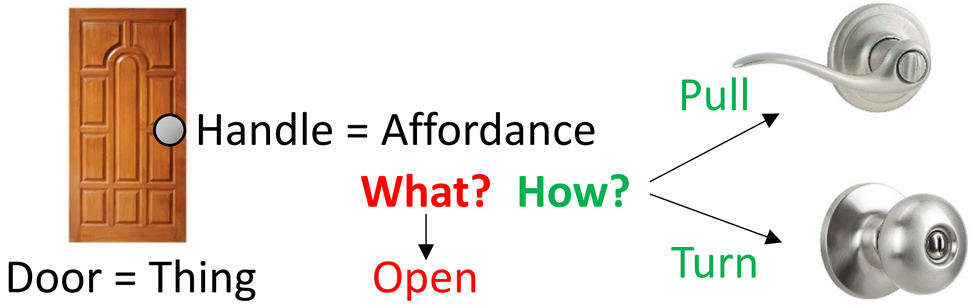
W3C Web of Things (WoT)

- **W3C WoT Working Group goal:** Adapting web technologies to IoT
- **Published:** Thing Description (TD) metadata format
  - TD describes the available interactions (network API) of a Thing
- **In Progress:** TD 1.1 Update, Thing Models, Discovery, Profiles
  - How to obtain TDs? How to ensure interoperability?



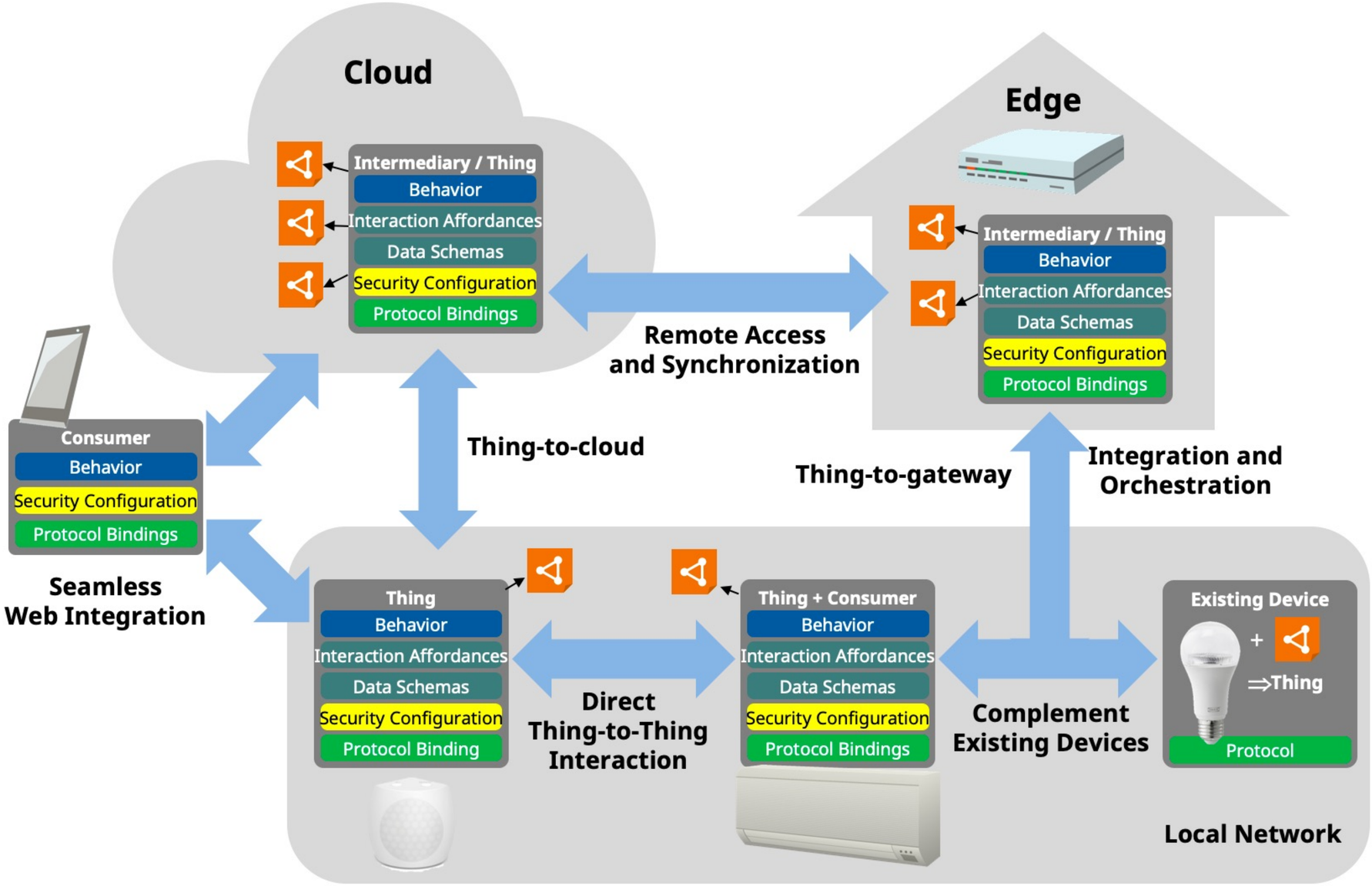
WoT Thing Descriptions

- WHAT the possible choices are
  - Properties
  - Events
  - Actions
- HOW to interact with the Thing
  - Protocol operations and options
  - Data schemas and content types
  - Security requirements




```
{
  "@context": [
    "https://www.w3.org/ns/td",
    { "iot": "http://iotschema.org/" }
  ],
  "id": "urn:dev:ops:32473-WoTLamp-1234",
  "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": {
      ...
```

Usage Patterns



Current 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 Models:

- 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.

New Deliverables

- Thing Description 1.1
  - Canonicalization (and WIP, Signing)
  - Validation levels
  - Thing Model
  - various other extensions, e.g. to security, data schemas, etc.
- Discovery
  - Introductions: DNS-SD, DID, CoRE RD
  - Directory Service: HTTP API for searchable database of TDs
  - Self-Description: .well-known, fetching of TD directly from Things
- Profiles
  - Emphasis on "hub" use-case, http/json
- Use Cases and Requirements (informative document)

Thing Description 1.1: Updates

- Canonicalization and Signing
  - WIP, but proposal is based on JOSE/JWS/JWA (incl. RFC 8037)
  - Can extract parts of a TD to sign using JSONPointer/JSONPath/XPath queries
- Security Scheme Improvements
  - URI Templates
  - Security information in body
  - OAuth "device" flow
- Thing Model
  - TD describes instance, TM describes class
  - Provides templating/parameterization mechanism
  - TD can reference *one* TM using a link
  - TMs can reference or extend other TMs (and *parts* of other TMs)

Discovery: Goals

Capabilities

- Support both local and global/remote discovery (unconstrained by network domain)
- Support "localizable" discovery (constrainable by location)
- Support both "syntactic query" (keywords) and "semantic query" (linked data)
- Support a directory service for searching large repositories of Things
- Support peer-to-peer (self-identifying "smart object") discovery

Privacy-Preserving Architecture

- Respect device and information Lifecycle
- Distribute TDs only to authenticated and authorized users
- Don’t leak private data to unauthorized users
- Don’t leak data that can be used to INFER private information to unauthorized users

Alignment with Existing and Evolving Standards

- IETF CoRE Resource Directories, CoRE Link Format, DID, OGC, WGS84, XPath, …
- Compatible with WoT Scripting API

Discovery: Two-Phase Architecture

Phase 1: Introduction

- “First Contact” Protocol
  - Answers the question: How to initiate discovery from zero knowledge?
- Open
  - Can be accessed with no or limited access controls
  - Based on existing standards, and can be extended to new standards
- Lightweight
  - Does not use significant resources on responder
  - Resistant to Denial of Service attacks
- Provides intentionally limited information
  - Avoid leaking any metadata that can be used to infer private data
  - This includes types of devices, device ids, owners, timestamps, etc.

Phase 2: Exploration

- Authentication and authorization required
- Supports more complex query and filtering capabilities (JSON Path, XPath, SPARQL)
- Provides access to rich metadata (TDs)
- Access controls can limit data returned

Discovery: Status

Introductions

- DNS-SD (including mDNS) – new service names
- CoRE RD – resource types
- DID – endpoint types
- Well-known URLs: to "guess" URL from an IP
- Direct: anything else that returns a URL
- **Note:** link types distinguishing a Directory and a Thing are useful but not mandatory

Exploration

- "Smart Objects": Retrieve TD directly from Thing
- Directory service API: described using a TD
- Provides multiple query types:
  - JSONPath – mandatory
  - XPath – optional
  - SPARQL – optional
- Pagination, etc.

What is a WoT Profile?

- A **WoT Profile** is a normative subset of a *WoT Thing Description* with a normative binding to a selected protocol.

- Profiles guarantee **interoperability** between compliant implementations, multiple profiles are possible.

- The **WoT Profile Specification** defines a **normative** set of *constraints and rules* on the **data model**, **representation format** and **protocol binding**.

- These constraints and rules provide clarifications and make decisions that reduce the complexity for implementers of the WoT standard.

- The rules are prescriptive, to ensure that compliant implementations satisfy the semantic guarantees implied by them.

Profiles: Constraints

Constraints on	*Rationale*	*Example*
vocabulary of Thing Description classes	guaranteed set of metadata fields	Make specific vocabulary terms mandatory, remove others
class relationships	unambiguous structure	limited cardinality, e.g. only one form per operation per interaction affordance.
values of vocabulary terms	simplified processing	Limit the length of characters per string. Always use arrays, where the spec permits a string or an array of strings.
data schemas	simplified processing	Limits on nesting
security	reduced implementation effort	Only a restricted set of security mechanisms
protocol binding	guaranteed protocol semantics	limited protocol(s) and protocol features, Example: predefined mapping of http verbs (GET/PUT) to operation verbs, similar constraints for other protocols.

Profiles: Current Work

- Defining a core/baseline profile with a HTTP binding.
- Identifying constraints and rules on the data model.
- Unambiguous interaction semantics for properties, actions and events.
- Constraints on payload formats.
- Protocol binding semantics, e.g. headers, response codes.
- Security constraints.
- Representation format constraints.

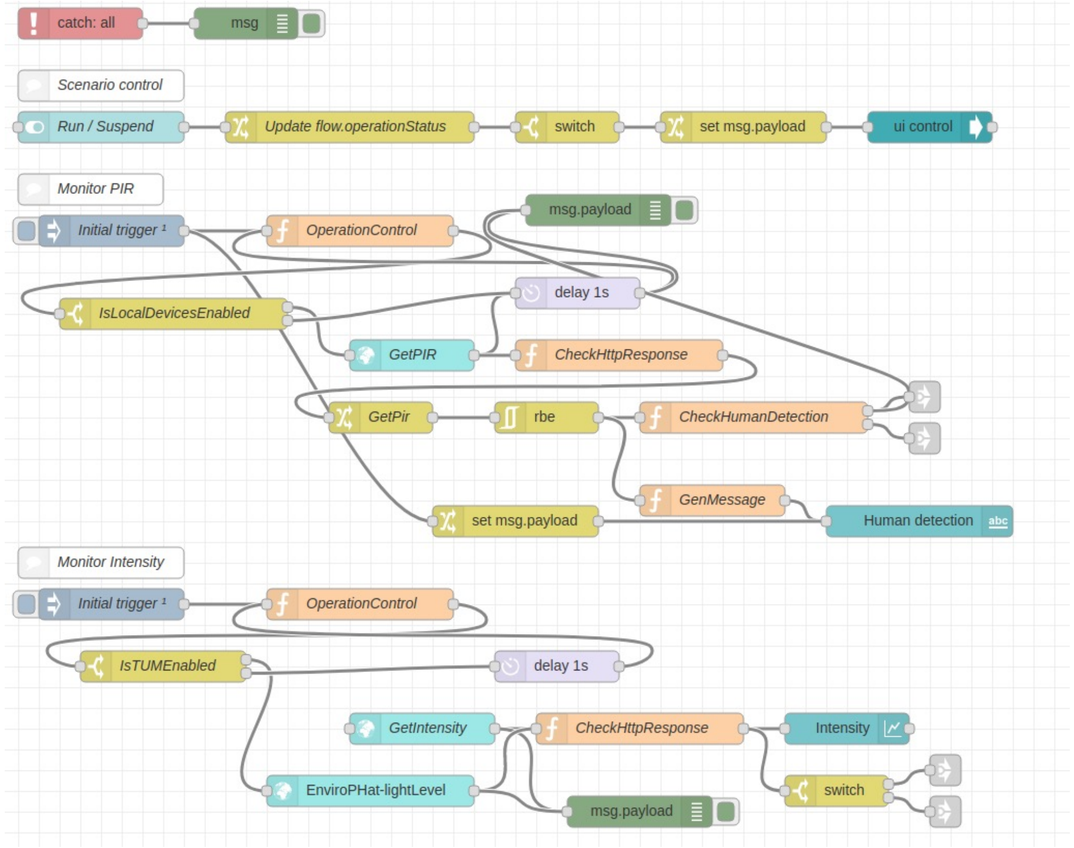
Use Cases – W3C Smart City Workshop

Others

- Law Enforcement
- Parking
- Accessibility
- Traffic and Logistics
- Public Transportation
- Air Quality and Weather
- Cultural Space Mgmt
- Construction Services
- Land Management
- Emergency Services
- Water Management
- Hybrid Ruralization

WoT Orchestration

Node-RED/node-gen



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); });
```

Documents and Resources

New/Updated Normative Documents in Draft Status:

- Architecture 1.1: https://github.com/w3c/wot-architecture
- Thing Description 1.1: https://github.com/w3c/wot-thing-description
- Discovery: https://github.com/w3c/wot-discovery
- Profiles: https://github.com/w3c/wot-profile

New/Updated Informative Documents in Draft Status:

- Binding Templates: https://github.com/w3c/wot-binding-templates
- Scripting API: https://github.com/w3c/wot-scripting-api
- Use Cases and Requirements: https://github.com/w3c/wot-usecases

Other Resources:

- Web Site: https://www.w3.org/WoT/

Contacts

https://www.w3.org/WoT

Dr. Michael McCool

Principal Engineer

Intel

Technology Pathfinding

michael.mccool@intel.com

Dr. Sebastian Kaebisch

Senior Key Expert

Siemens

Technology

sebastian.kaebisch@siemens.com

Backup

Image Credits

- Solar Installation Vietnam: By Intel Free Press - https://www.flickr.com/photos/intelfreepress/7169063498/sizes/o/in/photostream/, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=28011974
- Telemedicine Consult: By Intel Free Press - https://www.flickr.com/photos/intelfreepress/6948764580/sizes/o/in/photostream/, CC BY 2.0, https://commons.wikimedia.org/wiki/File:Telemedicine_Consult.jpg

OneDM Update

One Data Model
T2TRG Update

June 21, 2021

Contents

- Status
- Roadmap
- Provisional models
- Technical
  - Sensor modeling
  - Semantic Proxy
- Backup information

Status

- SDF Standardization - in process
  - What is our view of the the feature-schedule tradeoff?
  - Ongoing pressure test with new features
- ~200 models in the playground representing OMA LWM2M, OCF, Bluetooth Mesh, and ZCL
- Model adoption process for convergence
  - Life cycle, tracking, versioning, documentation
  - Initial models selected to test the process

Roadmap

- 1Q2021 – model adoption process agreed
- 2Q2021 - start provisional models through process
- Re-engage with contributors
- 3Q2021 publish first provisional models

Provisional Models

- OCF and OMA LWM2M contributions queued up
- OCF-style process based on constructive feedback
  - Make concrete proposals to make it better
  - IETF Chairs drive consensus and help break deadlocks
  - RFC 2418 WG chair role defined
- OMA model contribution strawman voltage and current sensor – discussion around how to handle quantities and units being measured

OCF dishwasher and dryer

https://openconnectivityfoundation.github.io/devicemodels/docs/index.html

oic.d.dishwasher contains the mandatory resources:
oic.r.switch.binary, oic.r.mode

oic.d.dryer contains the mandatory resources
oic.r.switch.binary, oic.r.operational.state

These 3 resources are already as SDF in the playground:
https://github.com/one-data-model/playground/blob/master/sdfObject/sdfobject-switch_binary.sdf.json

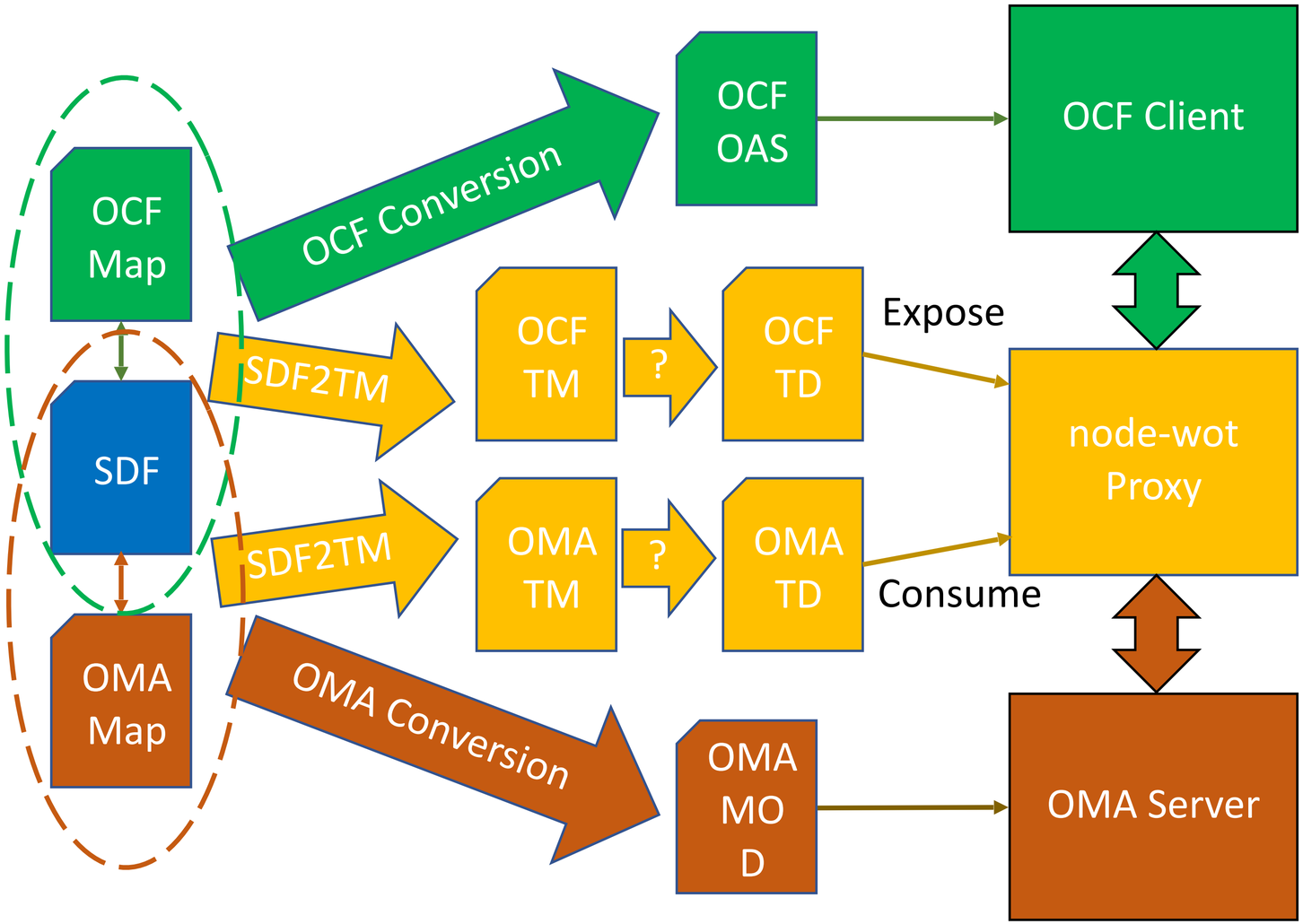
https://github.com/one-data-model/playground/blob/master/sdfObject/sdfobject-mode.sdf.json

https://github.com/one-data-model/playground/blob/master/sdfObject/sdfobject-operational_state.sdf.json

Sensor Modeling Issues

- Quantities and Units
  - Binding of units to sensor type
- Common pattern, specialized by what is sensed
  - Temperature, Voltage, Current, Flowrate
  - sdfData can be specialized for Quantity and Unit
  - E.g. TemperatureData in Degrees K
  - TemperatureSensor object, CurrentValue property, TemperatureData data type
- Multiple sensed quantities == multiple sensors
- What about combining data e.g. in columns?
- Bluetooth Mesh and BACnet – industry alignment

Semantic Proxy



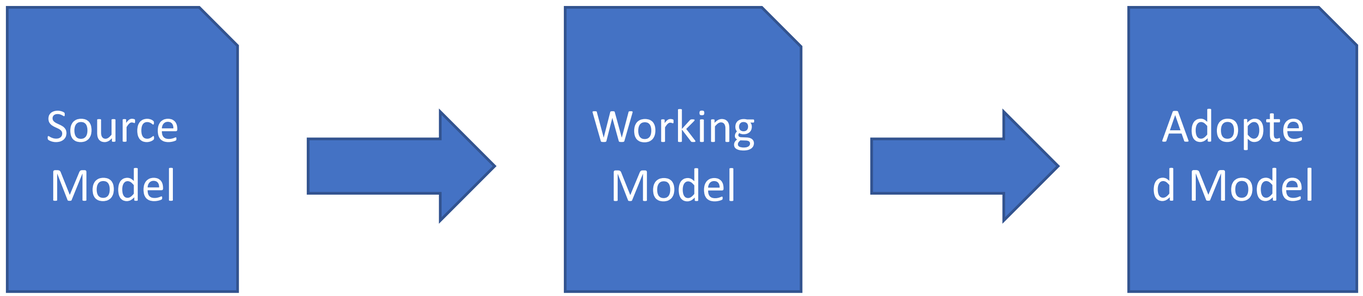
Backup

W3C WoT and SDF/OneDM

- Conversion workflow: SDF => Thing Model => Thing Description
  - Mapping files, protocol bindings, TD Forms
- SDF Processing for external references
  - TD Validation of annotations
- Semantic Proxy using node-wot
  - Multiple proof points

Model Lifecycle Tracking

- Namespace + version

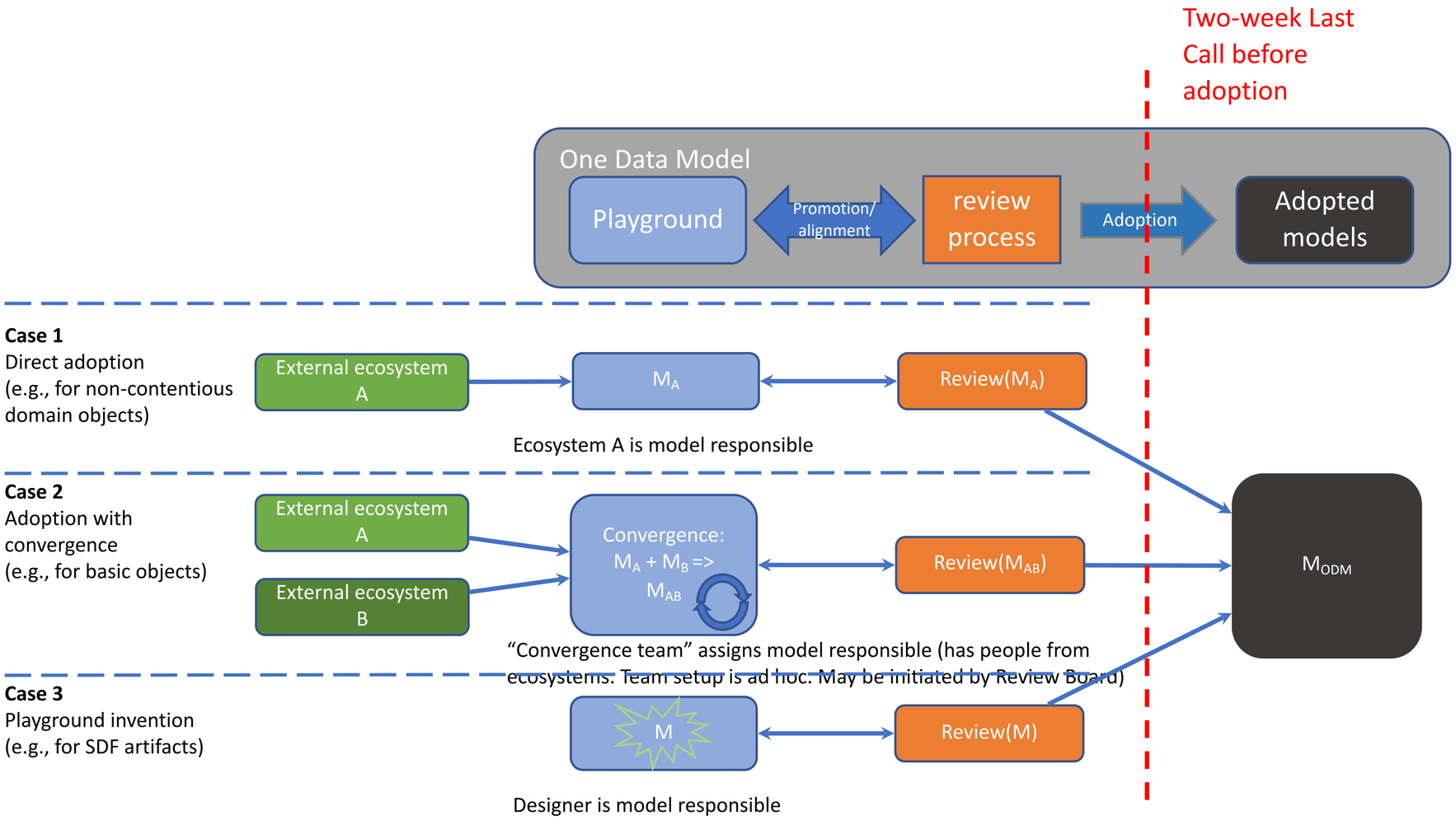


Private namespace
https://onedm.org/ipso/...
Private repository
No onedm version tracking

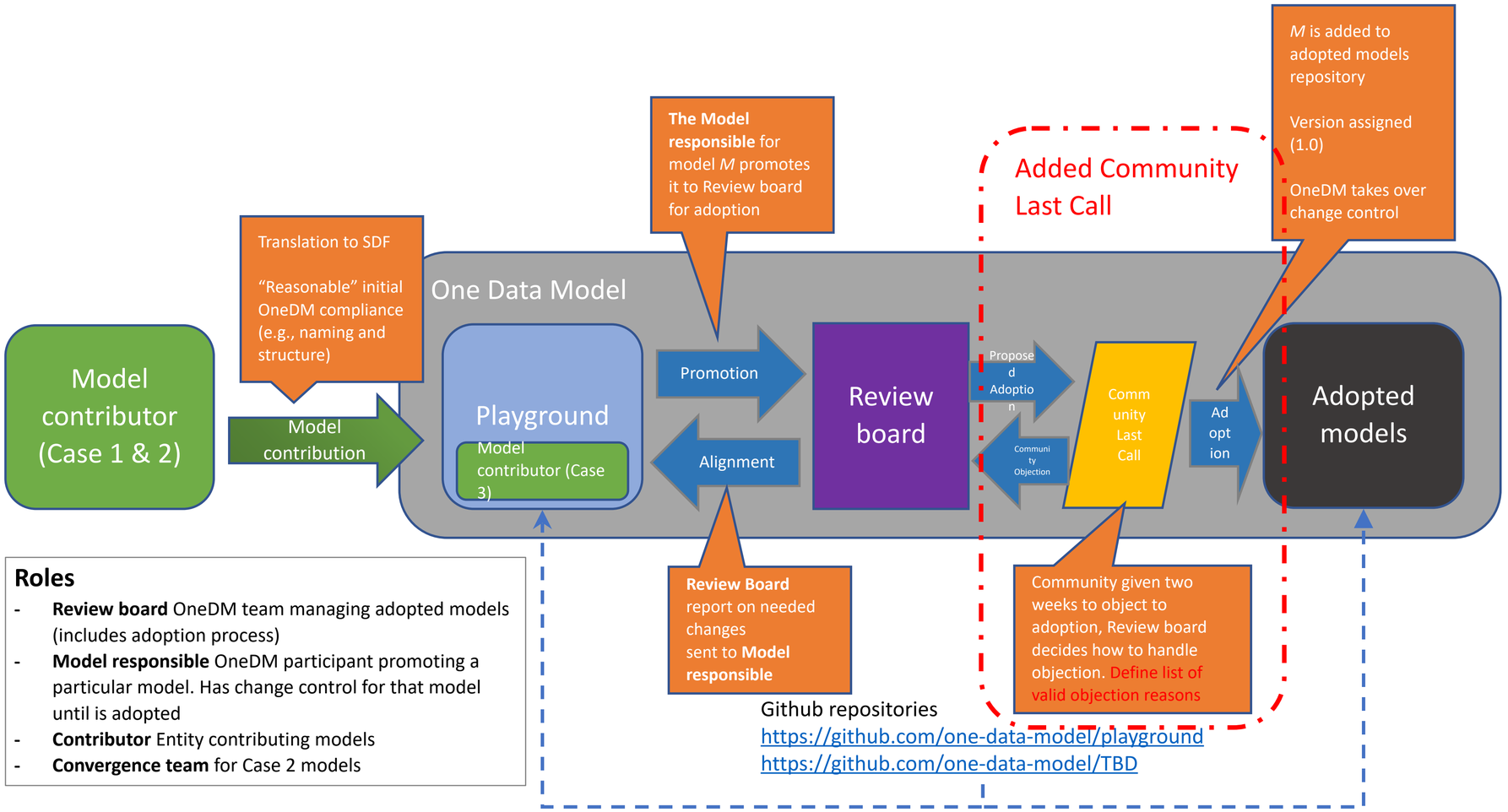
Working namespace
Working repository
(Playground default)
Version + identifier
e.g. 0.1.0-ipso.org

Adopted namespace
https:onedm/?
Stable repository
Version + no identifier
e.g. 1.0.0

Model origins



Model adoption process



Roles

- **Review board** OneDM team managing adopted models (includes adoption process)
- **Model responsible** OneDM participant promoting a particular model. Has change control for that model until is adopted
- **Contributor** Entity contributing models
- **Convergence team** for Case 2 models

Contributor Repositories

- Multiple states of vetting
- 1st level is validated and has only contributor guarantees
- 2nd level is guaranteed stable from OneDM
- Agreement for the originator to adopt the model back – different levels of involvement
- Exploratory for SDF features, what about new models
- Provide a way for contributors to use OneDM CI in their own repositories

Affiliation and Participation

- OCF – Cisco, Resideo, Shaw
- OMA LWM2M – Ericsson, ARM, Qualcomm
- Zigbee Alliance – SmartThings, Sensus, Comcast, Schneider Electric
- Z-Wave – Silicon Labs
- Bluetooth SIG - Silvair
- SunSpec Alliance - DER
- Invited experts – Bruce Nordman
- IETF - Carsten Bormann
- W3C WoT – Sebastian Kaebisch

IoT Edge Computing Challenges and Functions

https://tools.ietf.org/html/draft-irtf-t2trg-iot-edge-02

J. Hong, Y-G. Hong, X. de Foy, M. Kovatsch, E. Schooler and D. Kutscher

T2TRG Interim Meeting, June 2021

History of the Draft

- draft-hong-iot-edge-computing-01 (IETF 103)
  - Draft was presented along with two demo videos of use cases for IoT Edge computing (smart construction and real-time control system)
- draft-hong-iot-edge-computing-02 (IETF 104)
  - In a discussion on Edge and IoT in the T2TRG meeting, this draft was considered a possible starting point for a group document. New co-authors joined.
- draft-hong-t2trg-iot-edge-computing-00 (IETF 105)
  - Draft was integrated with *Survey and gap analysis, a* presentation made in T2TRG at IETF 100
- draft-hong-t2trg-iot-edge-computing-01 (IETF 106)
  - Focus changed from use case examples to Edge function analysis.
  - Draft changed from showing one Edge architecture to a range of models. Did not promote/preclude a particular model.
- draft-hong-t2trg-iot-edge-computing-02/3 (IETF 107)
  - Reorganized the draft, extended the background section and the list of functions
- draft-hong-t2trg-iot-edge-computing-04/05 (IETF 108)
  - Addressed comments from Thomas, including improvements to IoT challenges and to the draft structure; completed section 4 with additional text on distributed model, and developing research challenges associated with functions; started the RG adoption process
- draft-irtf-t2trg-iot-edge-computing-00/01 (IETF 110)
  - Addressed comments from Marie-Jose and Carlos, including new use cases
- draft-irtf-t2trg-iot-edge-computing-02 (T2TRG interim meeting, June 2021)
  - Addressed comments from Milan

Quick Overview

1. Introduction

2. Background

- IoT, cloud computing, edge computing, use cases

3. IoT Challenges Leading Towards Edge Computing

- Time sensitivity, connectivity cost, resilience to intermittent connectivity, privacy and security
  - (Reasons that motivate the use of edge computing for IoT)

4. IoT Edge Computing Functions

- Overview of IoT edge computing today, general model, distributed model
- Functions/components, listing research challenges
  - OAM components: resource discovery and authentication, edge organization and federation, multi-tenancy and isolation
  - Functional components: in-network computation, edge caching and caching, northbound/southbound communication, communication brokering, other services
  - Application components: IoT end devices management, data management and analytics
- Simulation and emulation environments

5. Security Considerations

Updates 1/2

1. Comment: “Some [use cases], like smart construction and smart water system, feel a bit generic and as described do not seem to exemplify the need for edge processing.”
   - Re-wrote the smart construction use case and deleted the smart water system one. Reordered some use cases.
2. Comment: “While most of the IoT traffic flow tends to be “upstream”, I think that the availability and cost of connectivity can be challenging in various IoT settings and suggest retitling and recasting this section as Connectivity Cost. It also states that many IoT deployments are not challenged by constrained network bandwidth, citing Wi-Fi 6 and 5G links. Since those are not yet widely deployed or suitable for a variety of IoT installations, I suggest changing “many” to “some”.”
   - Changed the title to Connectivity Cost and made other updates to take into account the constrained network bandwidth case.
3. Comment: “Resilience in IoT often entails the ability to operate autonomously in periods of disconnectedness in order to preserve the integrity and safety of the controlled system, possibly in a degraded mode. It might be useful to add that IoT devices and gateways are often expected to operate in the always-on and unattended mode, thus adding design challenges of fault detection and unassisted recovery to operational states.”
   - Added text based on this comment in section 4.1 (which describes IoT EC today)

Comment thread: https://mailarchive.ietf.org/arch/msg/t2trg/UKTksEMeqT9yAXg8PCwxUeiYhWY/

Updates 2/2

4. Comment: “Section 4 - Should be revised to separate description of edge functions from the implementation mechanisms […] I suggest dividing the description into key IoT functions of the edge - […]”

We have updated and re-organized sections 4, especially:

- Virtualization Management is now Multi-Tenancy and Isolation
- External APIs is now Northbound/Southbound Communication
- Data Management section was expanded to include Analytics

5. Comment: “[The section on ‘Edge Caching’] should clarify that the edge node may offer local data storage (persistence subject to retention policies), caching (anticipatory best effort), or both.”
   - We updated and retitled this section (now ‘Edge Storage and Caching’) to include both local storage and caching
6. Also added new research challenges, corresponding to new papers added in reference.
7. Editorial comments were addressed

Comment thread: https://mailarchive.ietf.org/arch/msg/t2trg/UKTksEMeqT9yAXg8PCwxUeiYhWY/

Plans for the Draft

- To our knowledge, all outstanding comments are addressed, the draft is in a stable state.
  - Additional comments are welcome.

- Last call?

Initial Security Setup

Secure IoT Bootstrapping: A Survey

draft-irtf-t2trg-secure-bootstrapping-00

63

Mohit Sethi, Behcet Sarikaya, and Dan Garcia-Carillo

Secure Bootstrapping

- Goals of this document:
  - Overview of bootstrapping related terminology.
  - Identify common patterns and provide recommendations on the applicability of terms.
  - Illustrative examples of bootstrapping techniques (cover many IETF and non-IETF protocols).
  - ~~Classify techniques based on requirements and assumptions.~~

64

Terminology

- Current list:
  - Bootstrapping - Registration
  - Provisioning - Commissioning
  - Onboarding - Configuration
  - Initialization - Discovery

65

- Bootstrapping one example among many

Terminology

- New title: Terminology and processes for initial security setup of IoT devices
- Break down protocols into:
  - Players: What are the parties. E.g.: manufacturer, user, network administrator.
  - Beliefs:
    - Pre-setup: What knowledge is available before setup. E.g.: manufacturer issued certificates containing IDevID
    - Post-setup: What knowledge is instilled during setup. E.g.: SSID, network key, etc.
  - Processes: Sequence of events and interactions required setup? E.g.: power up device and scan a QR code.

66

Device Provisioning Protocol (DPP)

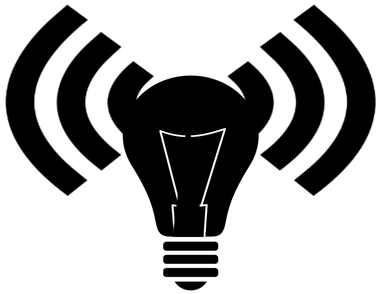
- Wi-Fi alliance protocol for user friendly Wi-Fi setup
- Relies on a configurator, e.g. a smartphone application, for setting up all all other devices, called enrollees, in the network.
- Following three phases/sub-protocols:
  - Bootstrapping: configurator obtains bootstrapping information from the enrollee using an out-of-band channel such as scanning a QR code or tapping NFC
  - Authentication: provides authentication of the responder to an initiator. Can optionally authenticate the initiator to the responder
  - Configuration: Using keys established from the authentication protocol, the enrollee asks the configurator for information such as the SSID and passphrase

Device Provisioning Protocol (DPP)

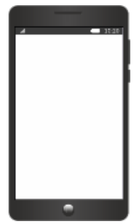
- Players:
  - Manufacturer installs a key pair and prints the public-key and other metadata on device/packaging
  - User also the device owner
  - Companion device aka smartphone
- Beliefs:
  - Pre-setup: Manufacturer installed asymmetric key pair
  - Post-setup: Device is instilled with knowledge such as target network, SSID, passphrase, etc.
- Processes:
  - User scans QR code[68] or taps NFC for authentication
  - Twice if mutual authentication is desired
  - Send information such as SSID, passphrase of home AP

Bluetooth Mesh - Provisioning

- Provisioning: adding a new device to the mesh network
- Provisioner: smartphone for provisioning new devices



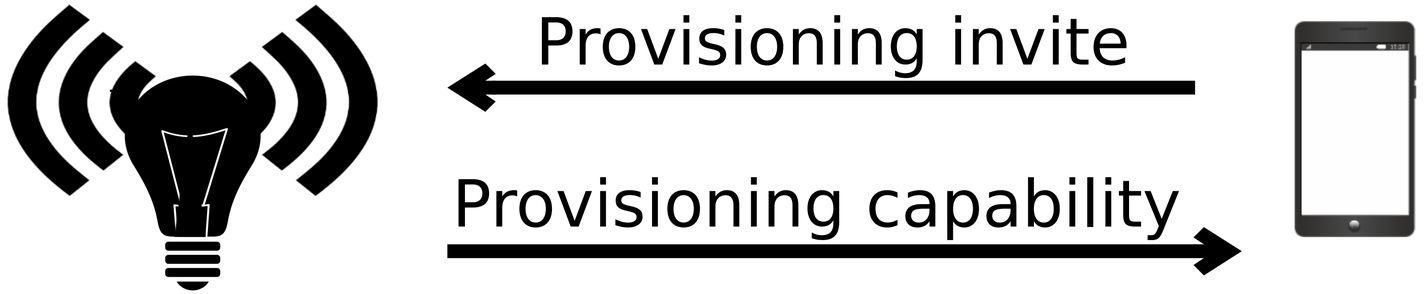
Unprovisioned device



Provisioner

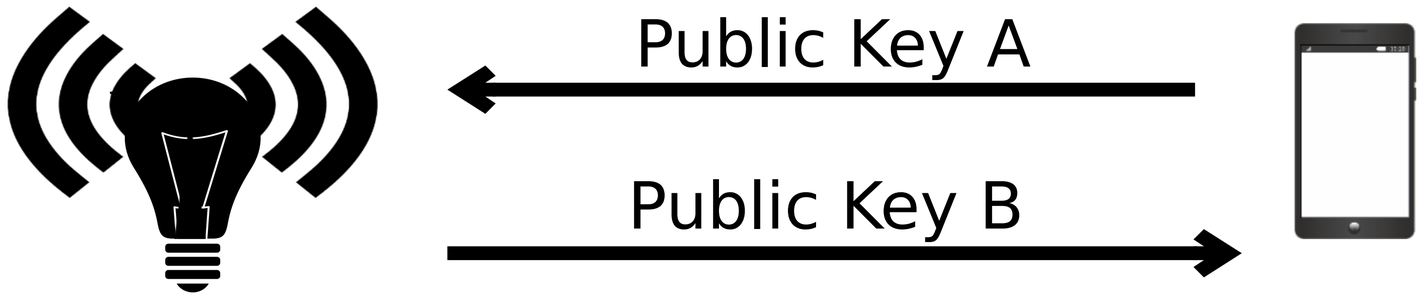
Bluetooth Mesh - Provisioning

- Invitation: provisioner discovers new device via beacon and sends an invitation.
- New device responds with provisioning capabilities (including elements, security algorithms, I/O capability etc.)



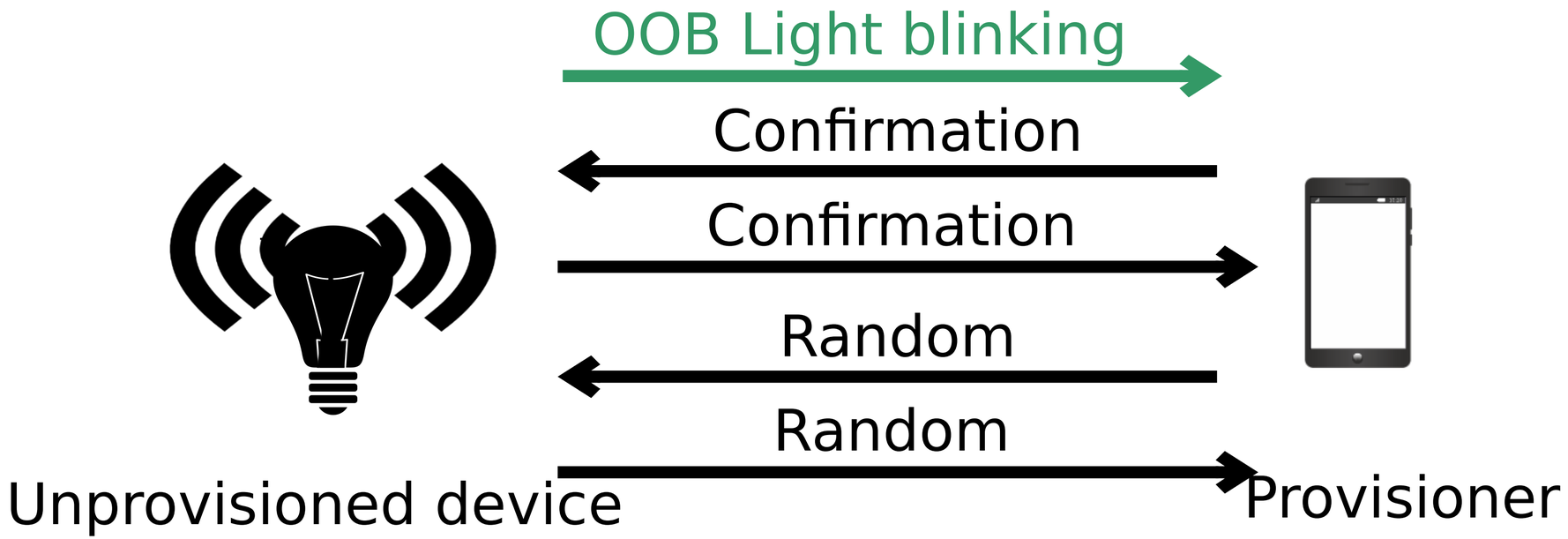
Bluetooth Mesh - Provisioning

- Public key exchange: ECDH key exchange with fresh keys (if OOB input or OOB output authentication used)



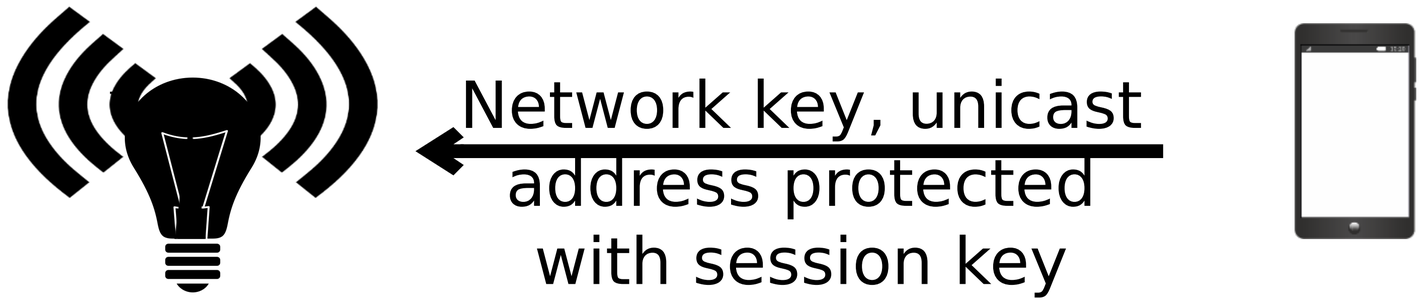
Bluetooth Mesh - Provisioning

- Authentication: Device or Provisioner generate and show a random number (as blinking LED, audio etc.) that is input on the other side. Both send commitments with random number and reveal random numbers after. Generate session key



Bluetooth Mesh - Provisioning

- Distribution of provisioning data: Provisioner sends data: network key, IV index, unicast address assigned etc



Bluetooth Mesh - Provisioning

- Players:
  - User also the device owner
  - Provisioner aka smartphone
- Beliefs:
  - Pre-setup: None no installed/hard-coded credentials
  - Post-setup: Device learns about the target network, credentials, application (lighting etc.)
- Processes:[74]
  - User scans a blinking light
  - Sends information such as application/group etc.

Status

- Draft on github: https://github.com/t2trg/sbootstrapping
- Pull Requests and issues on github and mailing list are welcome.

75

IDevID Considerations

https://datatracker.ietf.org/doc/draft-richardson-t2trg-idevid-considerations/

This document grew out of MASA-considerations

And ANIMA Registrar-considerations

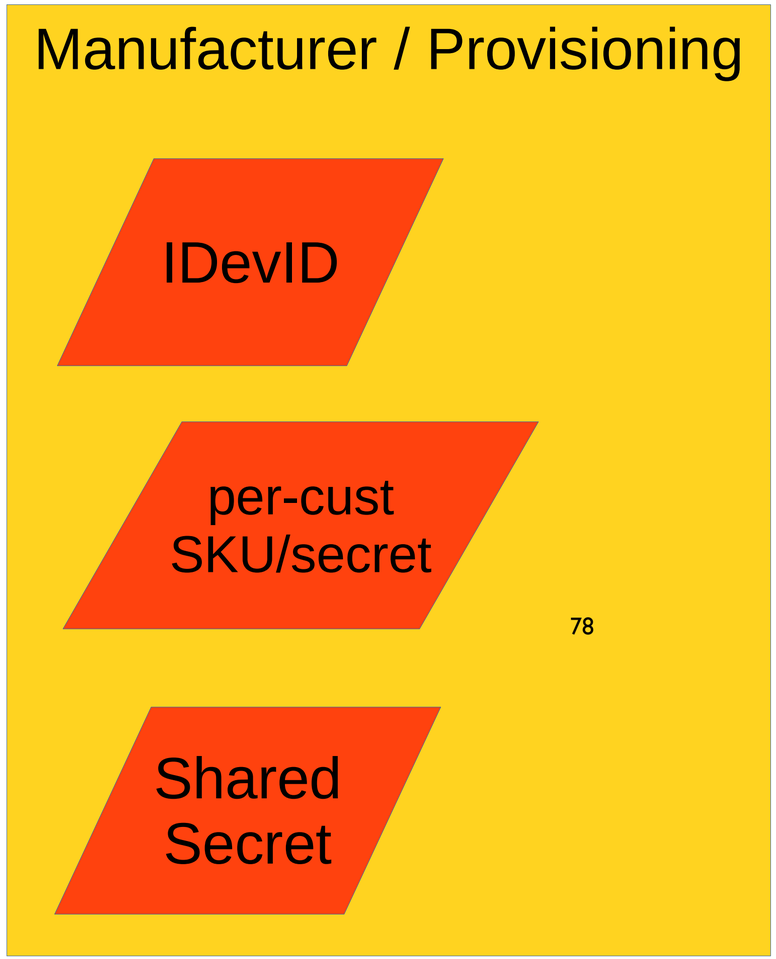
Both involve building and operating a PKI

The security of network depends upon security of PKI

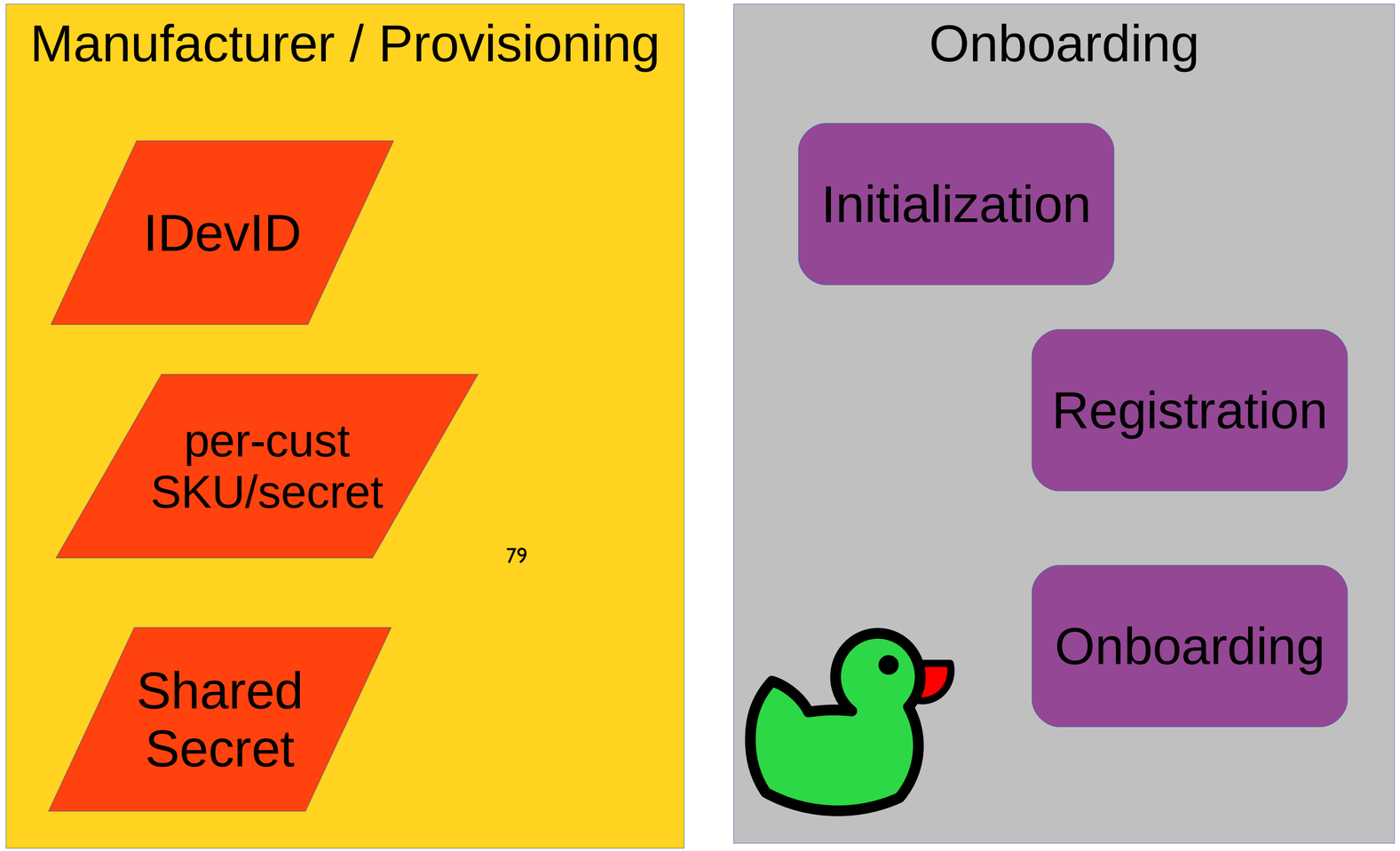
Some reviewers assumed Enterprise operated PKI would be totally insecure, or so security aware as to be unuseable.

Taxonomy of IoT lifecycle

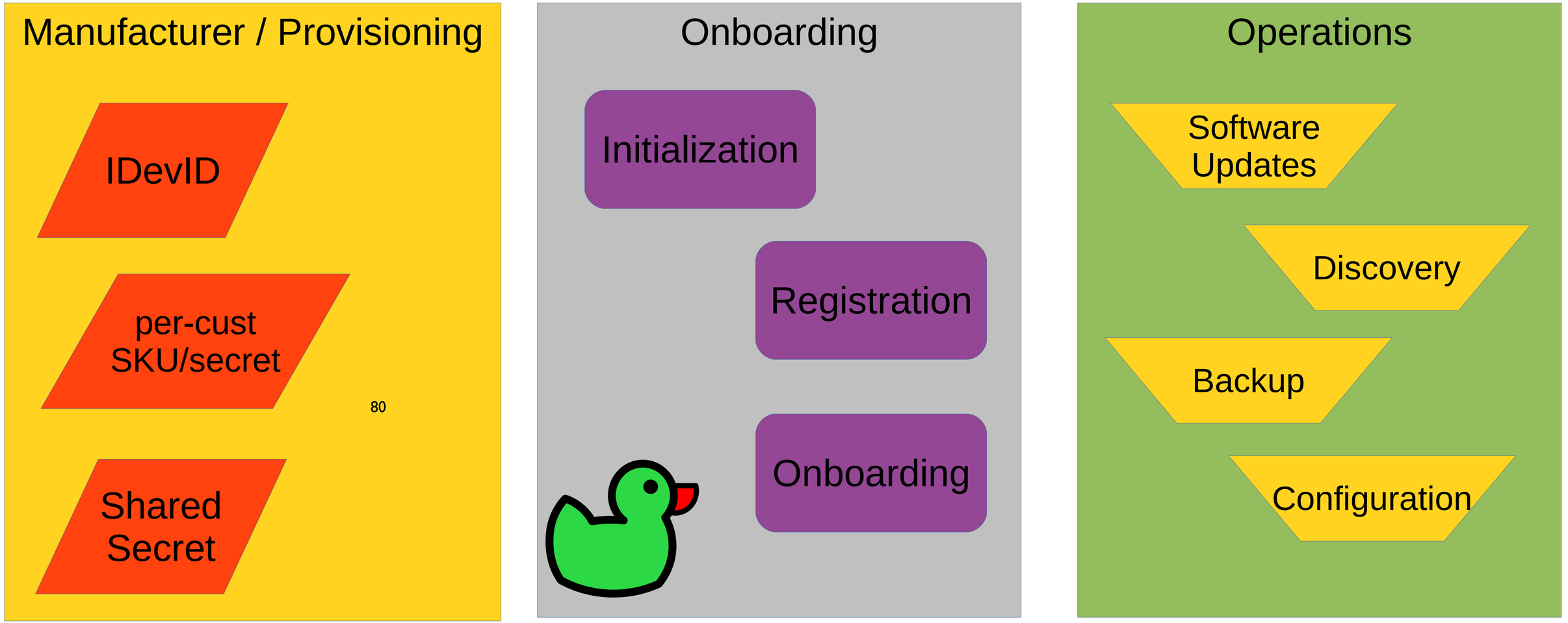
Taxonomy of IoT lifecycle



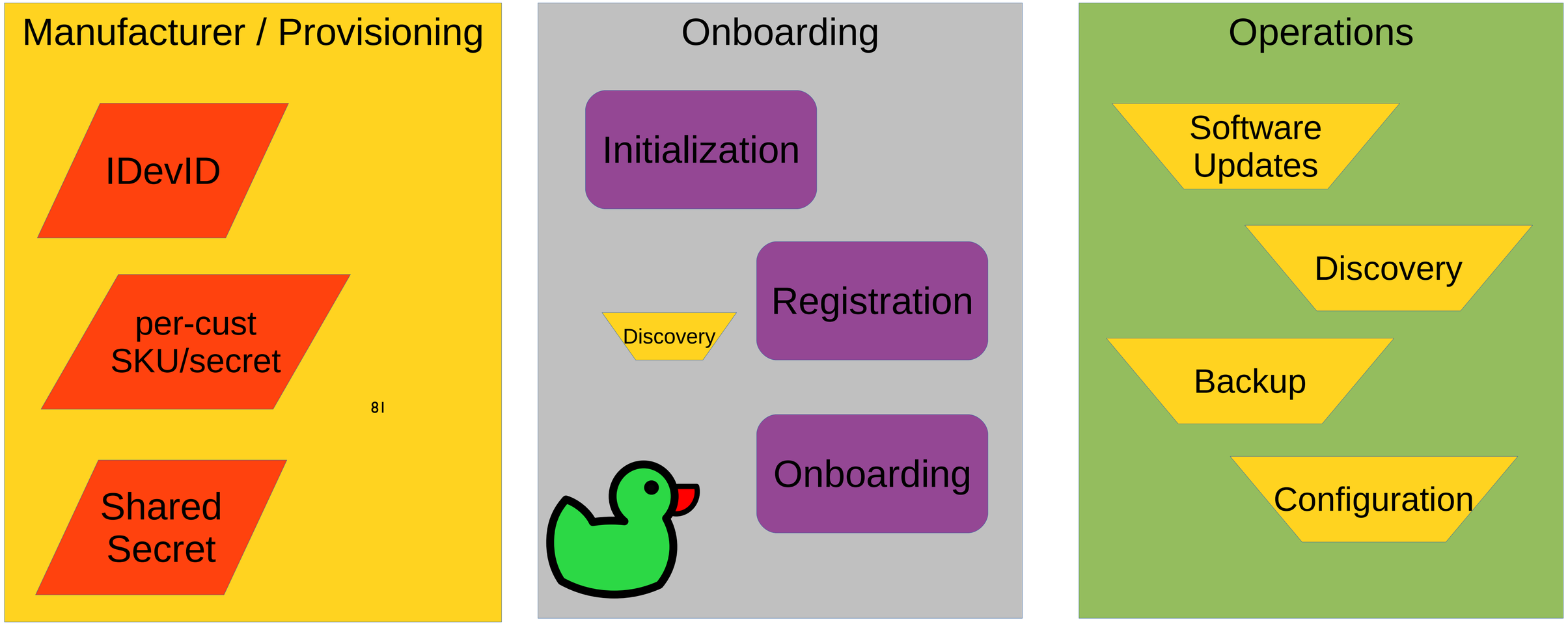
Taxonomy of IoT lifecycle



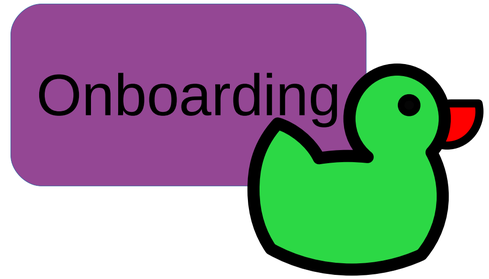
Taxonomy of IoT lifecycle



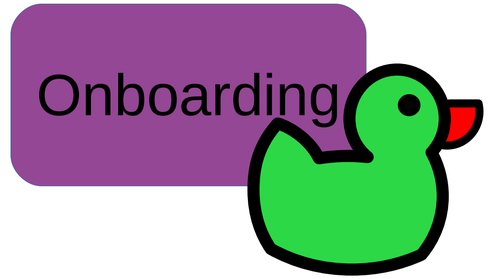
Taxonomy of IoT lifecycle



IDevID taxonomy

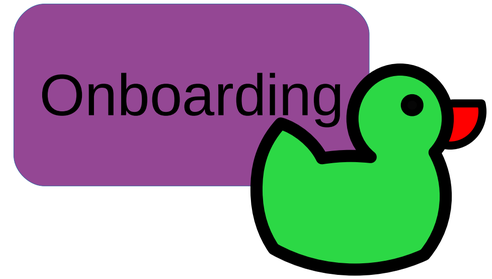


IDevID taxonomy



Manufacturer / Provisioning

IDevID taxonomy



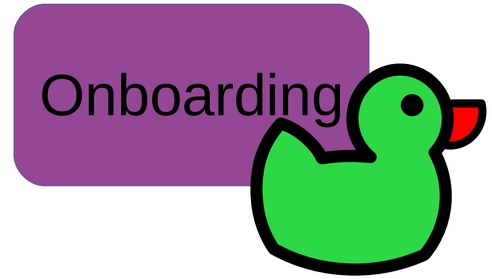
Manufacturer / Provisioning

Trust anchors

Shared Secret (seed)

IDevID

IDevID taxonomy



Manufacturer / Provisioning

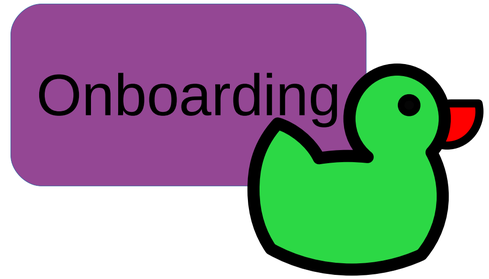
Excel File (?)

Trust anchors

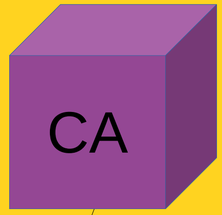
Shared Secret (seed)

IDevID

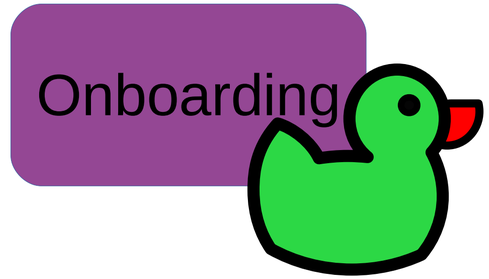
IDevID taxonomy



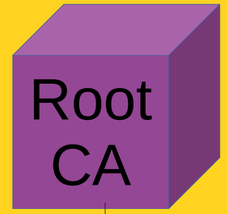
Manufacturer / Provisioning

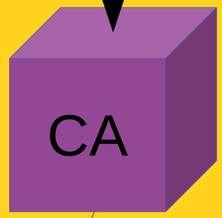


IDevID taxonomy

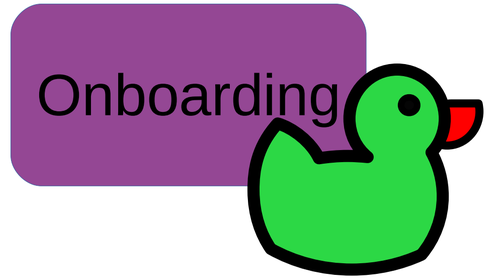


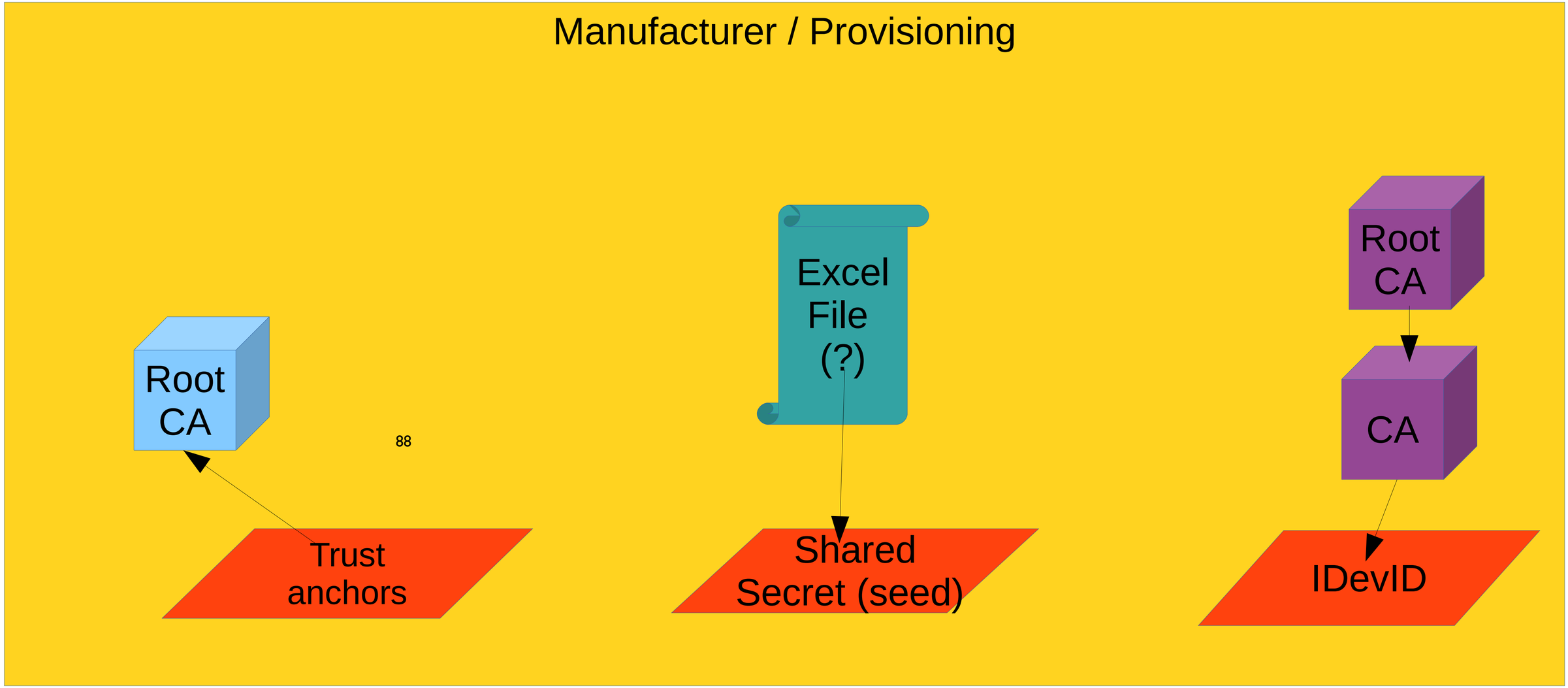
Manufacturer / Provisioning



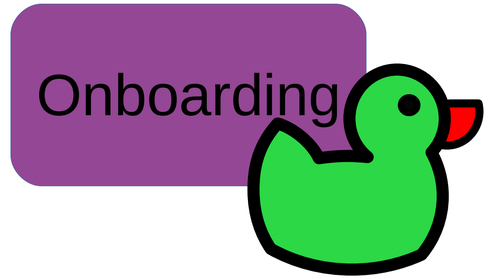


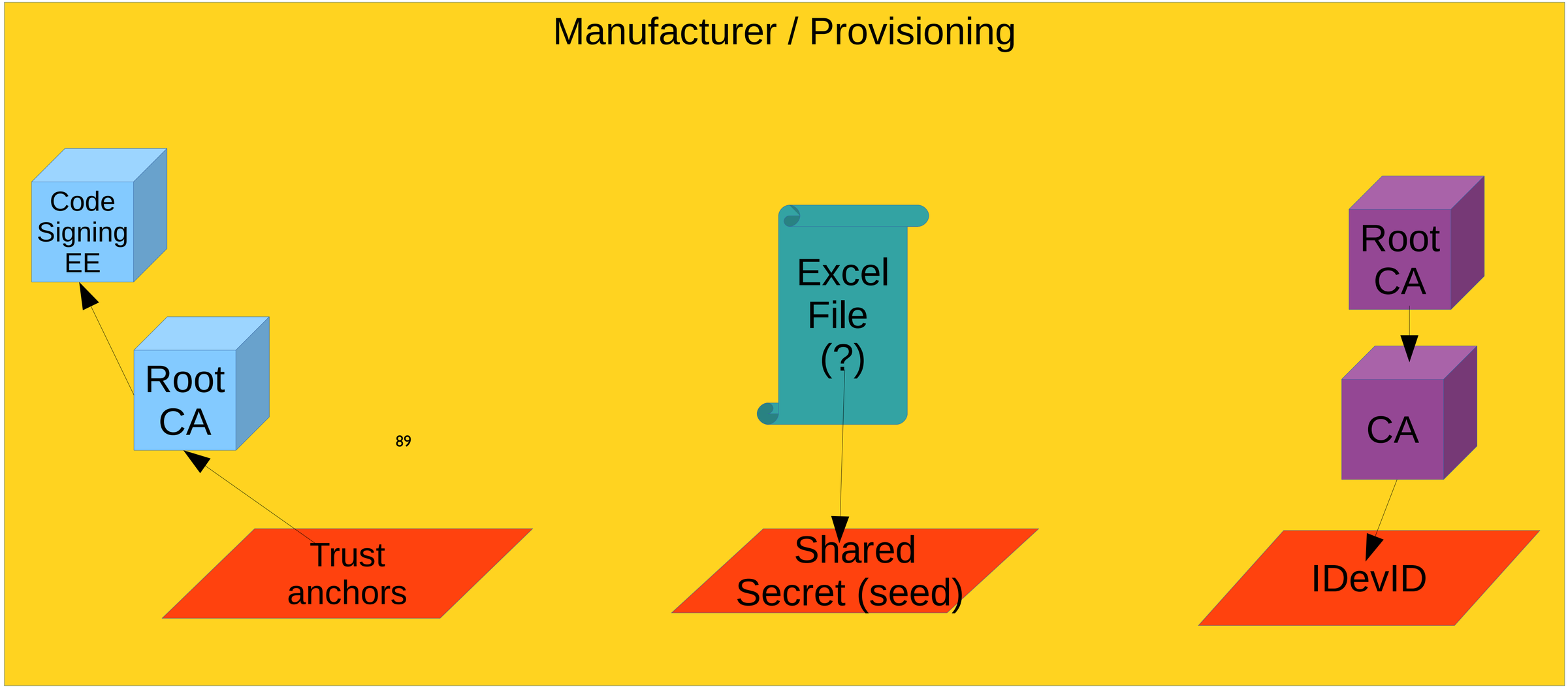
IDevID taxonomy



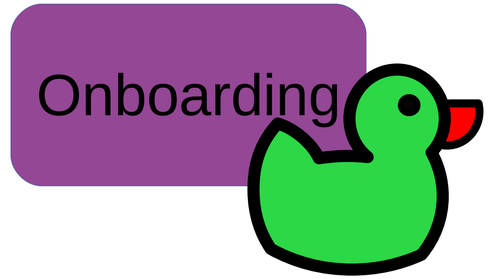


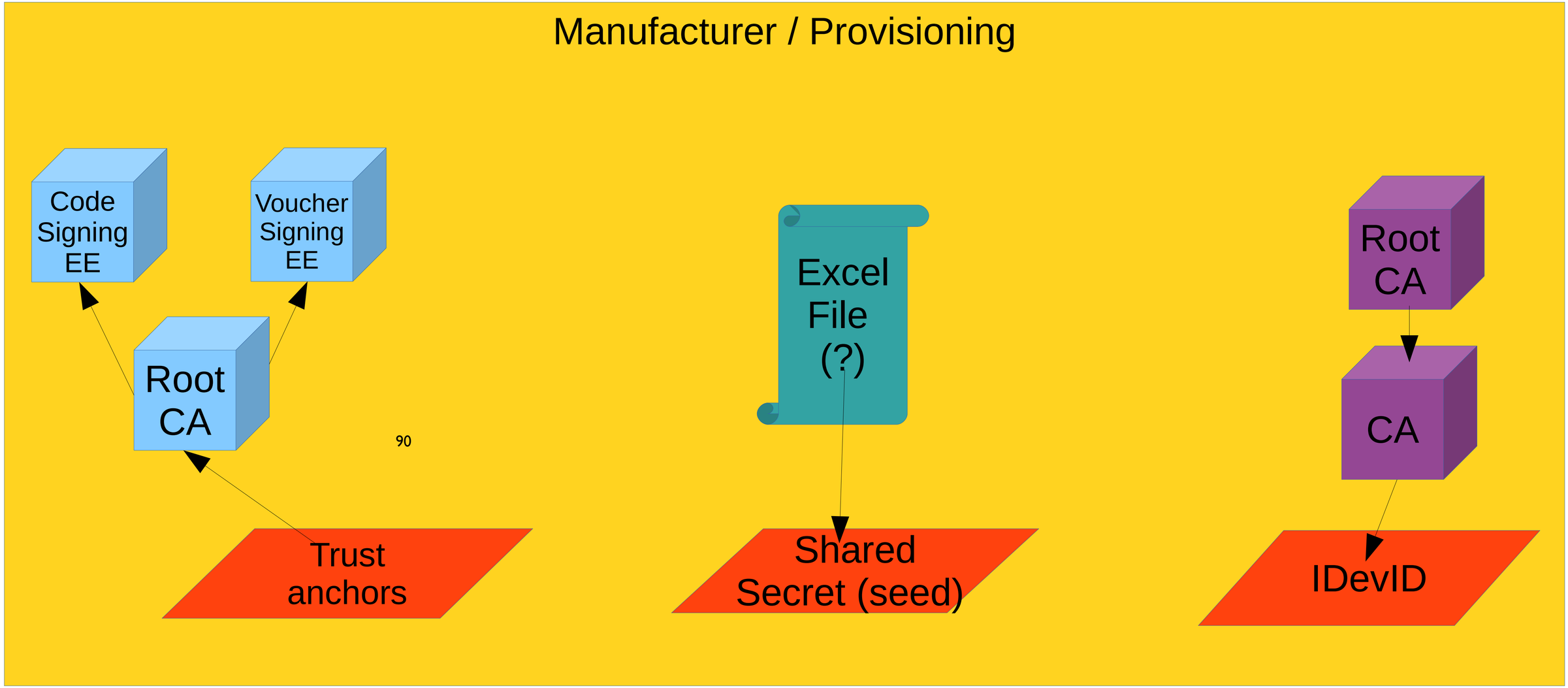
IDevID taxonomy



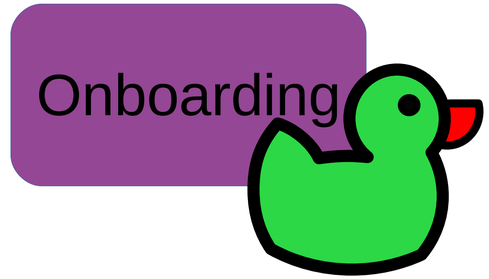


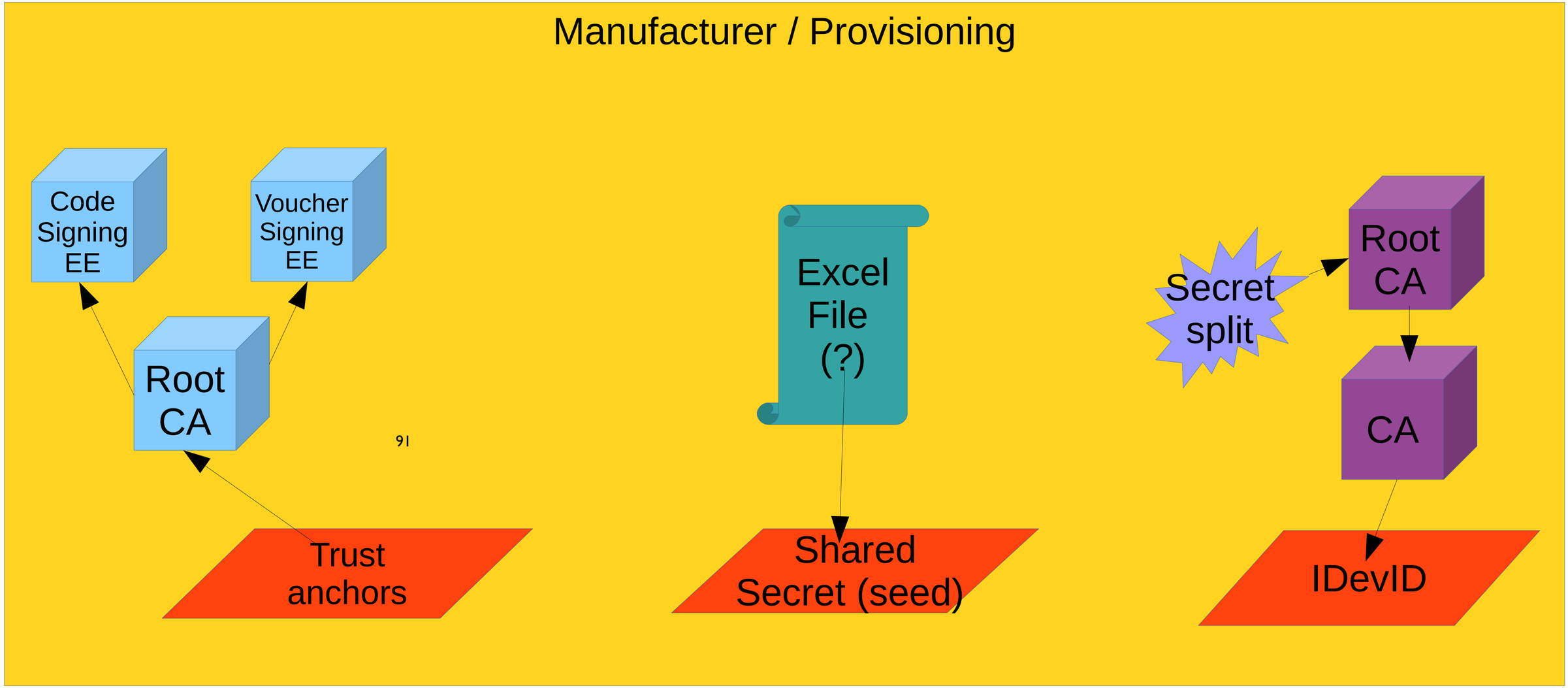
IDevID taxonomy



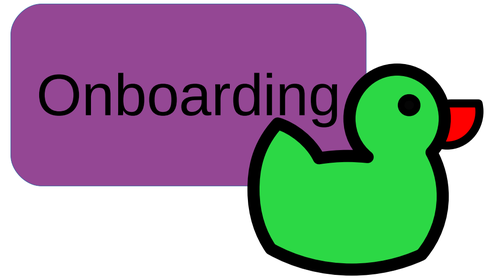


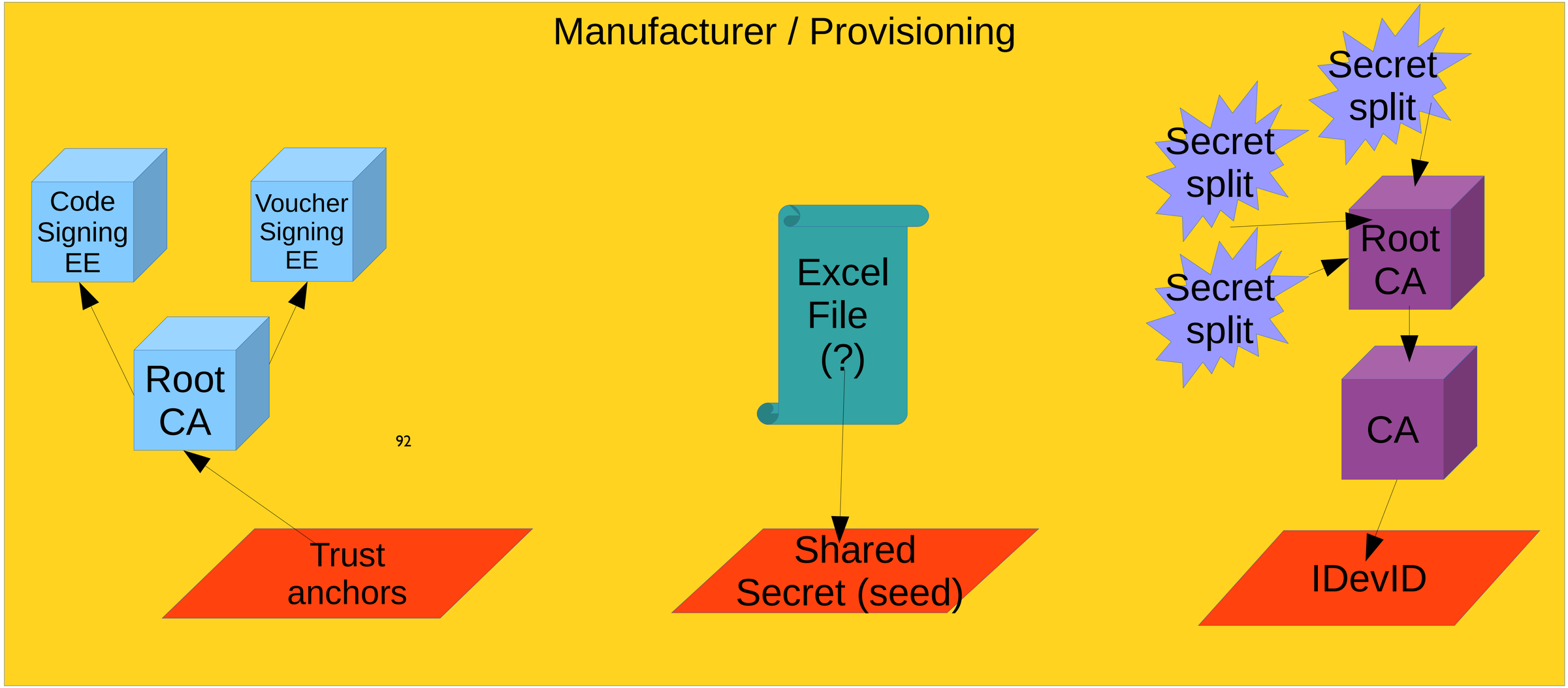
IDevID taxonomy



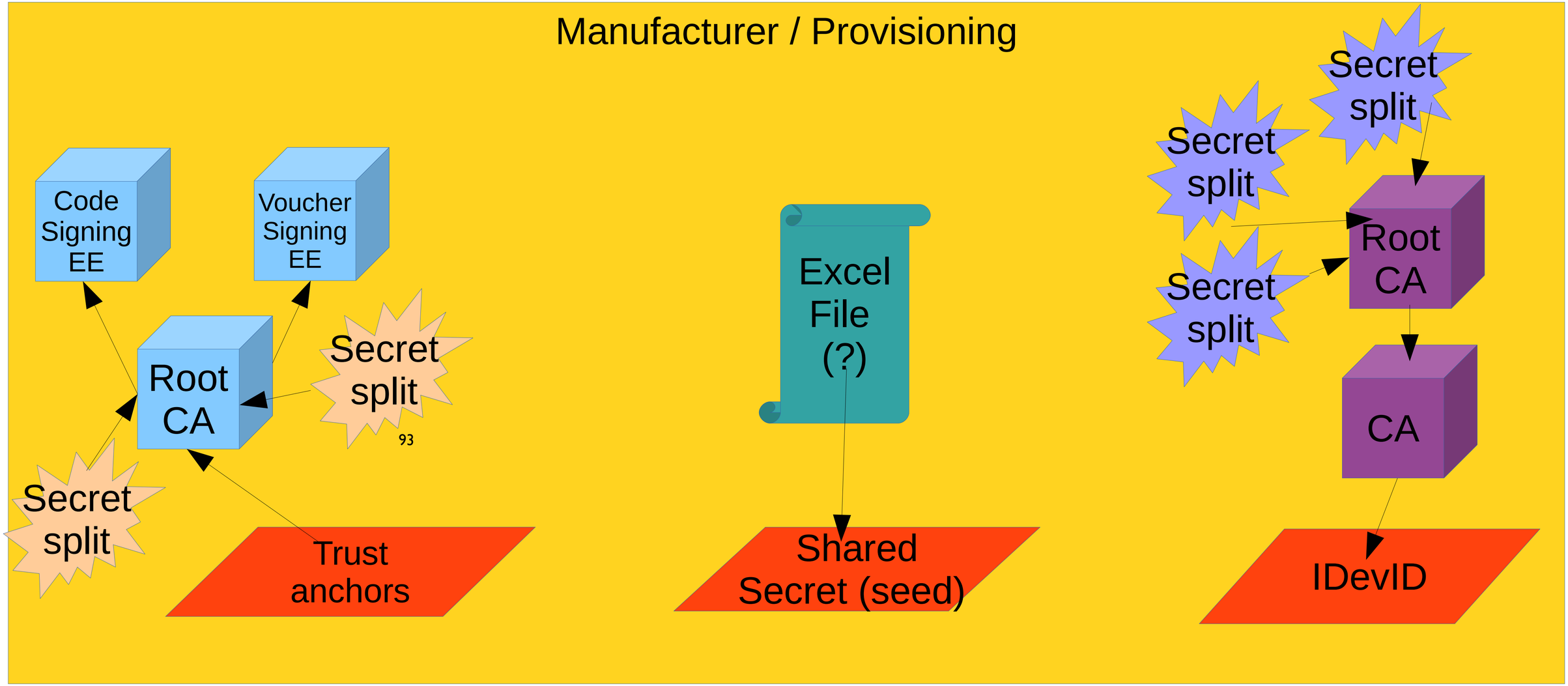


IDevID taxonomy





IDevID taxonomy



IDevID taxonomy