

ASDF
Interim 05
2022-01-17

Chairs: Michael Richardson, Niklas Widell

Note Well

- You will be recorded
- Be nice, and be professional - notification of IETF Code of Conduct
- The IPR guidelines of the IETF apply: see <http://ietf.org/ipr> for details.

Repo: <https://github.com/ietf-wg-asdf/asdf-working-group-notes>
Notes: <https://notes.ietf.org/notes-ietf-interim-2022-asdf-01-asdf>

Note Well

This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.

As a reminder:

- By participating in the IETF, you agree to follow IETF processes and policies.
- If you are aware that any IETF 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.
- As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
- Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
- 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.

Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:

- [BCP 9](#) (Internet Standards Process)
- [BCP 25](#) (Working Group processes)
- [BCP 25](#) (Anti-Harassment Procedures)
- [BCP 54](#) (Code of Conduct)
- [BCP 78](#) (Copyright)
- [BCP 79](#) (Patents, Participation)
- <https://www.ietf.org/privacy-policy/> (Privacy Policy)



IETF Code Of Conduct Guidelines

RFC7154

1. Treat colleagues with respect
 2. Speak slowly and limit the use of slang
 3. Dispute ideas by using reasoned argument
 4. Use best engineering judgment
 5. Find the best solution for the whole Internet
 6. Contribute to the ongoing work of the group and the IETF
- Please keep these in mind both at the mic and on Jabber/Meetecho IM

Agenda

1. Note Well. <https://www.ietf.org/about/note-well/>
2. Logistics for Meeting
 1. CodiMD for notes <https://notes.ietf.org/notes-ietf-interim-2022-asdf-01-asdf>
 2. Webex
 3. Roll call/Blue sheet
3. WG status update (Chairs - 5 min)
 1. Next steps
 2. IETF 113 meeting plans?
4. Getting SDF ready for WGLC <https://datatracker.ietf.org/doc/draft-ietf-asdf-sdf/>
 1. Issue list processing
5. AoB

Administrative

Resources

- CodiMD for notes <https://notes.ietf.org/notes-ietf-interim-2022-asdf-01-asdf>
- Webex: <https://ietf.webex.com/ietf/j.php?MTID=m296df2f64b965a80b3531fdbd0d053b1>
- All ASDF notes at:
<https://github.com/ietf-wg-asdf/asdf-working-group-notes.git>

WG procedures

- Decisions on mailing list: <https://www.ietf.org/mailman/listinfo/asdf>
- Work on Github: <https://github.com/ietf-wg-asdf>
 - Use Issue tracker for issues (new features and fixes)
- Schedule (doodle) regular virtual interims between virtual physical meetings

Status update

- **Status:** ASDF WG was chartered in October 2020
- **Chairs:** Michael Richardson and Niklas Widell
- **Progress so far:**
 - Three IETFs with Hackathons, continuing work on converters (I-D for YANG, I-D mention of WoT, ...)
 - Four virtual interims
 - SDF 1.1 with Implementation draft status
 - is a second implementation draft needed?
- **Meeting plans:**
 - ASDF at 113
 - Additional interims to be **planned:** February 2022?

ASDF WG next steps

1.



SDF RFC

- Covers core SDF functionality
- Feature complete for basic OneDM models

1. Finish draft
2. Submit WGLC
3. Identify shepherd

2.



SDF Protocol
mapping



SDF Instances



SDF Relationships

1. Group appetite for more
2. Write drafts
3. Recharter?

Update from OneDM

OneDM (One Data Model Liaison Group) was created in 2018 to harmonise IoT data models across ecosystems (OCF, IPSO, Bluetooth, CSA/Zigbee etc.)

Objectives:

A common format for describing data and interaction models

Created initial version of SDF to support that interoperability, which was then sent to IETF for full specification in ASDF WG

Adopting “industry standard best practice” models in a OneDM repository

Process framework has been created, an initial review board has been set up and the first models are going through the adoption process right now

Getting SDF ready for WGLC

- Yesterday's draft -10 is a roll-up of minor changes
- Finish remaining issues/PRs
- Do a readiness check
- Run WGLC
- Identify Document Shepherd

PR #45

- Idea:
partially revert SDF 1.1 move to single-in/single-out for Actions/Events
- Disadvantages:
 - clouds up equivalence of Property and Event
 - Requires new mechanism where we already have “type”: “object”
- Recommendation: WONTFIX
 - But document how the pointers look like

PR #36

- 2 components:
 - sdfRef clarifications (including “resolved form”)
 - Addition of sdfRefFrom
- sdfRef clarifications are desirable, recommendation: cherry-pick
- sdfRefFrom is maybe not yet ready; a more thoroughly designed version could be added as an extension

Issue #49, PR #50

- Readiness: sdfProduct
- Recommendation: Remove and put back in a later extension

Issue #52

- Currently dataqualities: readable, writable, observable (r/w/o)
- These really describe interactions, not data
but we *interact* on Property/Action/Event, not on individual data
 - dataqualities might pertain to deeply nested data,
these can't be r/w/o on their own
- Move up to affordance level? Which affordances?

Readiness check

- sdfProduct and r/w/o: examples for pieces of SDF that (1) have not been checked out or (2) only have been checked out only in specific contexts
 - Also: vestiges of data modeling (e.g., pattern) vs. information modeling (e.g., the odd collection of format, sdfType values)
- Need to do a CDR (critical design review)
 - Before or as part of WGLC

Strategy for SDF 1 RFC

- Minimize further additions
 - Focus on finishing what we have
- Do check need and readiness
 - If in doubt, leave it for further extension
- Make sure extension points for this are available

Strategy for SDF 2 RFC

- Should be documented in SDF 1
- Preferably exercise extension points (e.g., adding new qualities)
- Might need to have some deeper extensions

Backup from 112

Getting SDF ready for WGLC

- Draft is in good shape
 - Some issues to be resolved (next slides):
Actual change requests,
FDT limitations we may have to live with,
Various tasks, tools work that needs to be completed,
Potential future work
- Identify Document Shepherd

Change requests: P/A/E (1)

- PR #45: Structured sdfInputData & sdfOutputData
 - Recently renovated in SDF 1.1
 - Pendulum swings back a bit
- Limited experience with Action/Event
 - Hard to understand why Event should have different structure at all from Property

IETF 111 slide

(1) Properties, Actions, Events

- Property: $[T \rightarrow \blacksquare \rightarrow T]$
(if writable) (if readable, observable)
- Action: $T_i \rightarrow \blacksquare \rightarrow T_o$
- Event: $\blacksquare \rightarrow T$
- Property and Action are distinct, but what about Property and Event?

Change requests: P/A/E (2)

- PR #45: Structured sdfInputData & sdfOutputData
 - Common for input to have multiple parameters
 - Useful to have SDF names for the parameters
 - can refer in application using SDF definitions using same techniques (e.g., JSON pointers) as with other SDF definitions
 - Simplifies definitions with multiple parameters (JSO Object definition vs. SDF definition); slightly more boilerplate for single parameter

Change requests: namespaces

- #28: Contribute to more than the default namespace?
 - We can indicate namespaces for imports (sdfRef), but not exports
 - Use cases somewhat weak
 - E.g., Thing with Objects from multiple namespaces
- Unclear: interaction with our use of JSON pointers (sdfPointer)

```
"sdfThing": {  
  "foo:ThingInFoo": {  
  }  
}
```

Limitations in FDTs

- #27: Factoring dataqualities in an sdfChoice
- Underlying problem: SDF grammar describes input to processing model
 - Sometimes needs to describe output!
 - Already seen in sdfRef
 - Can somewhat mitigate this: require that output also fits grammar

Tasks

- Clean up CDDL style (e.g., #44 CDDL comma use alignment)
- Clean up JSON examples (#40 Make sure that all JSON texts really are)
- #33 Add automated checks
- Collect and implement editorial reviews

Tool development needed

- #34 Split up jsonschema CDDL definition (co-occurrence)
- Expect this to be covered by significant CDDL tool advances in December

Future (1): info block

- Define more conventions for info block:
 - #29 Define "version" field in info block
 - #26 Rules for combining information from info blocks
- We are not alone...
([draft-ietf-opsawg-ol-00](#): Ownership and licensing statements in YANG)

Future (2): breadcrumbs

- PR #36: Improved sdfRef semantic breadcrumb with sdfRefFrom
- Similar considerations for processing of mapping files

Future (3): evolution guidelines

- #11: When do we use "sdfXxx", when do we use "xxx"?
- More generally, are there any **invariants** we want to record?

Timelines?

- Implementation draft (“SDF 1.2”) this year, if we make substantive changes
- Should be able to WGLC in during mid-January
- Processing of WGLC comments up to early February
- If shepherd and AD are fast, could have IETF LC during February
- Telechat date in March?

SDF.next: Mapping files

- [draft-bormann-asdf-sdf-mapping-00](#)
- Processing model: **Augmentation**
 - SDF spec + SDF mapping = SDF spec (with new qualities)
- Mapping file is:
 - Info block (optional content)
 - Namespace management, “default Namespace”
 - *(JSON pointer into SDF spec → new qualities to be added there)
- Problem: We don't have namespaces for **qualities** yet

Mapping file: example 1 (IPSO)

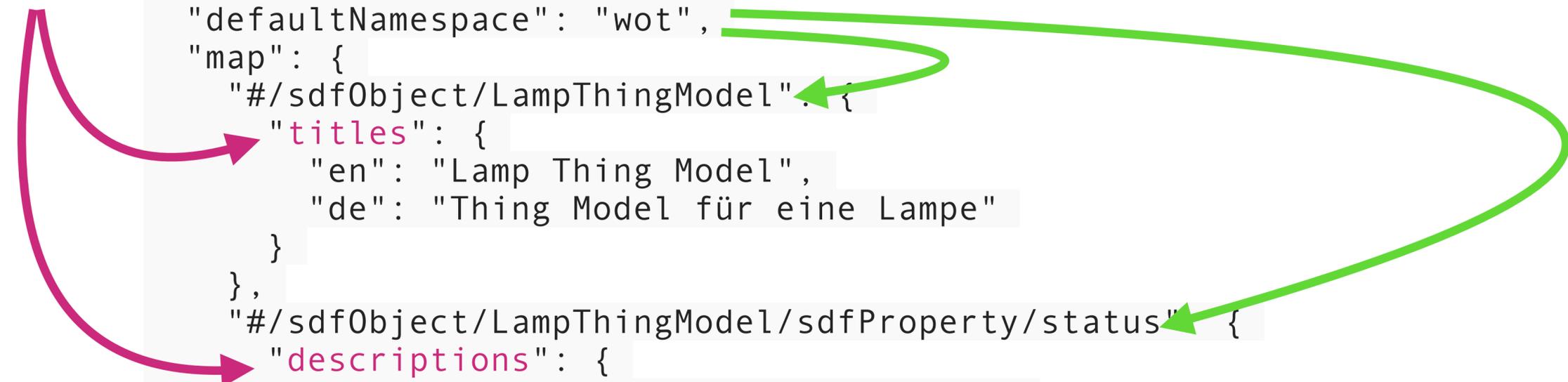
```
{
  "info": {
    "title": "IPSO ID mapping"
  },
  "namespace": {
    "onedm": "https://onedm.org/models"
  },
  "defaultNamespace": "onedm",
  "map": {
    "#/sdfObject/Digital_Input": {
      "id": 3200
    },
    "#/sdfObject/Digital_Input/sdfProperty/Digital_Input_State": {
      "id": 5500
    },
    "#/sdfObject/Digital_Input/sdfProperty/Digital_Input_Counter": {
      "id": 5501
    },
    .....
  }
}
```

Mapping file: example 2 (WoT)

```
{
  "info": {
    "title": "Lamp Thing Model: WoT TM mapping"
  },
  "namespace": {
    "wot": "http://www.w3.org/ns/td"
  },
  "defaultNamespace": "wot",
  "map": {
    "#/sdfObject/LampThingModel": {
      "titles": {
        "en": "Lamp Thing Model",
        "de": "Thing Model für eine Lampe"
      }
    },
    "#/sdfObject/LampThingModel/sdfProperty/status": {
      "descriptions": {
        "en": "Current status of the lamp",
        "de": "Aktueller Status der Lampe"
      }
    }
  }
}
```

Well, maybe not...

???



SDF Relation(ship)s

“How stuff relates to each other”

Stuff={things, objects, affordances, instances, ...}

Current SDF capabilities

- Parent-child / contained-in relations with SDF Thing/Object
- Thing (-> Thing) -> Object -> Affordance

“Definition relations”

- Beyond parent-child relations; at definition time (not run time)
 - “The Heater is heating a Boiler” or “The Switch controls a Light”
 - “From the 3 Robot Arms (A, B, and C) in a Thing, Arm A is next to B that is next to C”
 - Similar to DTDL Relationship and WoT Links?
- Parameters for relations
 - Semantics of the relations (referring to ontologies), cardinality, etc.
- Related: how a definition relates to other ontologies (@type)

Example

```
1  {
2    "sdfThing": {
3      "RobotArms": {
4        "sdfObject": {
5          "Arm-A": {
6            "sdfRelation": {
7              "next-to": {
8                "type": "terms:next-to",
9                "target": "#/sdfThing/RobotArms/sdfObject/Arm-B"
10             }
11           }
12         },
13         "Arm-B" : {
14           //...
15         }
16       }
17     }
18   }
```

“Instance relations”

- “This particular Light Switch controls that particular Light”
- New Affordance type or new Property type?
- Could be run time or definition time
- For example, LwM2M Object Link (links between objects in the same device; at run time)

Examples

Using new link property type

```
1 {  
2   "sdfObject": {  
3     "LightSwitch": {  
4       "sdfProperty": {  
5         "controlled-light" : {  
6           "type" : "link"  
7         },  
8         //...
```

Using new affordance type (or sdfRelation becomes one?)

```
1 {  
2   "sdfObject": {  
3     "LightSwitch": {  
4       "sdfRelationAffordance" : {  
5         "controlled-light" : {  
6           //...
```

SDF.next: Instance

- Two different approaches
 - Ari Keränen
 - Michael Koster

“Instance” definitions

- Value in being able to model a specific connected system of entities specified with SDF
- “Instance” as in “This particular light bulb based on SDF definition of Light thing/object” (in this building, and that room ...)
 - Need better term here? “Instance” already used in SDF spec for different use.
- “Class” -> “kind” -> “type” -> “instance”

“Instance” definitions with SDF?

- To give identifiers (/names) to instances
 - When SDF definitions used for provisioning things (devices)
 - Not describing state of the instance
- To describe relations between instances with SDF?
- (not about IP addresses, security credentials, etc.; can use e.g. WoT TD)

Instance SDF example

```
1 {
2   "namespace" : {"foo" : "https://example.org/foo"},
3   "defaultNamespace" : "foo",
4   "sdfInstance": {
5     "robot-arm-1": {
6       "sdfInstanceOf": "#/sdfObject/RobotArm",
```

Would result in RobotArm instance with name/id
<https://example.org/foo#sdfInstance/robot-arm-1>

SDF Use Case Features

Michael Koster

November 12, 2021

What's an Instance?

- Layered constraints on definitions
 - Start with base definitions; refine and compose
 - Reusability at all levels
- SDF Application Graph
 - Compose an application template for instances
 - Objects are connected and related
 - Portable and re-targetable, e.g. OMA LWM2M
- Protocol and Instance binding
 - Assign namespace IDs
 - On-the-air schemas and protocol settings

Language Features – Semantic Reference

- We need a way to refer to an SDF definition without implicit expansion – sdfRef with just the semantic endpoint – to use in relationships and composition
- sdfRef is a good name for this – too bad it's already in use
- Maybe we could still re-define sdfRef and add a new term for the expansion

Semantic Links

- S-P-O triples
- Subject can be present definition node
- Need a way to express relation types – predicates
- Object can be SDF pointer (SDF Semantic Reference) or an external reference as per RDF

Binding and Mapping

- WoT TD style protocol binding, with data schema and protocol vocabulary
- Protocol-specific qualities and constraints, e.g. ID numbers
- Protocol-specific refinements, value mapping for sdfChoice
- Mapping annotations could be external or in-line (good experience)
- May need SDF internal extension vocabularies

ASDF milestone review

- According to charter
 - ASDF is to develop SDF (Sept 2021) - **on track with slight delay**
- During specification process additional relevant work has been identified
 - Protocol mapping file definitions, Relations and Instances
 - Is this core to the document, or can it be an extension?
- Will we be required to extend the scope, to continue the work of ASDF:
 - charter revision will need to be developed and discussed on mailing lists and at upcoming interims
- (It would also be great to have at least one f2f meeting during the lifetime of the WG...)

AoB