

# WoT Profiles

Michael.Lagally@oracle.com

24.6.2020

# Motivation

- The [W3C Web of Things Architecture](#) and [Web of Things Thing Description](#) define a powerful mechanism and a format to describe myriads of very different devices, which may be connected over various protocols.
- The format is very flexible and open and puts very few normative requirements on devices that implement it.
- Use Cases require „out of the box interoperability“
- A generic client is impossible to implement.

# Is there a problem?

Current WoT spec defines a generic description language

Implementers are free to pick what they like, only very few constraints

No out of-the-box device interaction across different stakeholders

Several stakeholders raised interoperability concerns (e.g. TAG)

# Is there a problem?

## **A generic TD consumer is unimplementable!**

Because:

- Device implementers are free to pick TD features as they like, only very few constraints in the TD spec.
- Some things are not described in the TD, e.g. error behavior
- Significant implementation differences are permitted (e.g. PUT vs. POST to set a property).

Implications:

- No out of-the-box device interaction across different stakeholders.
- Each new device class (potentially) requires implementation additional features at the consumer.

# How to ensure Interoperability?

## **Specification**

- Unambiguous specification language, i.e. normative guarantees (RFC2119)
- Complete specification that covers all corner cases
- Domain / Target Specific Profiles

## **Reference Implementation**

## **Compliance Framework**

- Test Suites with guaranteed specification coverage
- Certification and validation program

# How to ensure Interoperability?

## **Specification**

- Unambiguous specification language, i.e. normative guarantees (RFC2119)
- Complete specification that covers all corner cases
- Domain / Target Specific Profiles

## Reference Implementation

## Compliance Framework

- Test Suites with guaranteed specification coverage
- Certification and validation program

# How do other specs solve this?

Example: DVB-GEM 1.3.1 / ETSI TS 102 728 V1.2.1

[http://www.etsi.org/deliver/etsi\\_ts/102700\\_102799/102728/01.02.01\\_60/ts\\_102728v010201p.pdf](http://www.etsi.org/deliver/etsi_ts/102700_102799/102728/01.02.01_60/ts_102728v010201p.pdf)

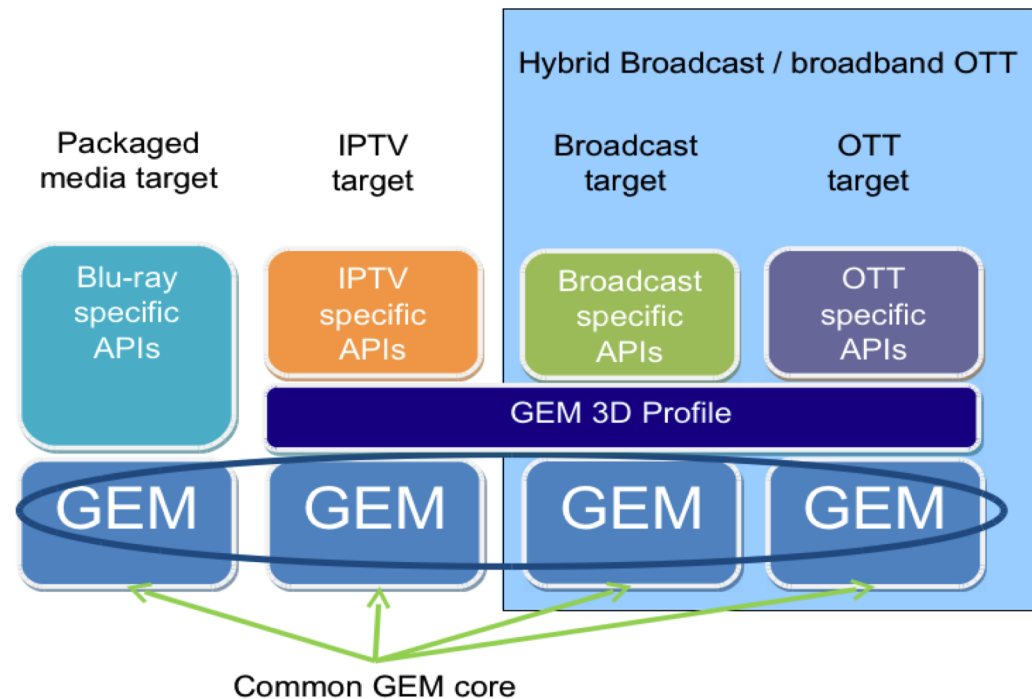
- Selecting a target specific subset of specification features
- Additional normative constraints
- Additional clarifications of ambiguities

# How do other specs solve this?

## Example: DVB-GEM

[http://www.etsi.org/deliver/etsi\\_ts/102700\\_102799/102728/01.02.01\\_60/ts\\_102728v010201p.pdf](http://www.etsi.org/deliver/etsi_ts/102700_102799/102728/01.02.01_60/ts_102728v010201p.pdf)

- Common core profile ensures cross target interoperability
- Combined profiles are possible
- Conformance testing regime enables interoperability guarantees





# How do other specs solve this?

## Example: DVB-GEM

- Selecting a target specific subset of specification features
- Additional normative requirements / constraints
- Additional clarifications of ambiguities

261 ETSI TS 102 728 V1.1.1 (2010-01)

Table 61: Platform profile definitions

Area	Specification	IPTV Target	Broadcast Targets		Packaged Media Targets	
			E.P.	I.P.	E.P.	I.P.
Static Formats						
Bitmap pictures	7.1.1.3, "PNG" + 15.1, "PNG - restrictions"	M	M	M	M	M
	7.1.1.3, "PNG" without restrictions	-	-	-	-	-
	7.1.1.4, "GIF"	-	-	-	-	-
	7.1.2, "MPEG-2 I-Frames"	O	M	M	M	M
	7.1.1.2, "JPEG" + 15.3, "JPEG - restrictions"	-	M	-	M	-
Audio clips	7.1.1.2, "JPEG" without restrictions	M	-	M	-	M
Video drips	7.1.4, "Monomedia format for audio clips"	M	M	M	M	M
Text encoding	7.1.3, "MPEG-2 Video "drips""	O	M	M	M	M
	7.1.5, "Monomedia format for text"	M	M	M	M	M
Media Streaming formats						
Video	7.2.2, "Video"	M	M	M	M	M
Audio	7.2.1, "Audio"	M	M	M	M	M
Subtitles	7.2.3, "Subtitles"	-	-	-	-	-
Fonts						
Built in	Character set see annex E, "Character set"	O	M	M	O	O
	Metrics see annex D, "Text presentation"					
Downloadable	Face: UK RNIB "Tiresias"					
	7.4.1, "PFIF"	O	M	M	O	O
	7.4.2, "OpenType"	O	O	O	O	O
Broadcast channel protocols						
MPEG-2	6.2.2, "MPEG-2 sections"	O	M	M	M	M
Object Carousel	6.2.5, "Object carousel"	O	M	M	-	-
IP Multicast	IP Multicast stack based on:					
	6.2.6, "Protocol for delivery of IP multicast over the broadcast channel"					
	6.2.7, "Internet Protocol (IP)"	-	O	Ro	-	-
	6.2.8, "User Datagram Protocol (UDP)"					
	6.2.10, "IP signalling"					
Interaction channel protocols						
TCP/IP	6.3.3, "Transmission Control Protocol"	M	-	M	-	M
UDP/IP	6.3.2, "Internet Protocol"	M	-	M	-	M
	6.3.9, "User Datagram Protocol"					
HTTP	6.3.7.1, "HTTP 1.1"	O	-	O	-	O
	6.3.7.2, "GEM profile of HTTP 1.0"	M	-	M	-	M
DSMCC-UU	6.3.4, "UNO-RPC"	-	-	O	-	-
	6.3.5, "UNO-CDR"					
RPC	6.3.6, "DSM-CC User to User"					
DNS	6.3.9, "DNS"	M	-	M	-	M
HTTPS	6.3.7.3, "HTTPS"	M	-	M	-	M
Interaction Channel File System	6.4.1, "File system implemented only by the interaction channel"	O	-	M	-	M
DSMCC / HTTP hybrid	6.4.2, "Hybrid between broadcast stream and interaction channel"	O	-	M	-	M
IPTV	5, "Basic architecture"	O	O	O	O	O
Application Model						
Application Model	All parts of clause 9, "Application model" except those clauses (and their subclauses) identified below	M	M	M	M	M
	9.6.1, "Applications loaded from the interaction channel"	O	O	O	O	-
	9.6.2, "Stored services"	O	O	O	O	-
	9.7, "Lifecycle of internet access applications"	O	-	M	-	O
	9.9, "Stored and cached applications"	O	O	O	-	-
	9.13, "Unbound Applications"	-	O	O	O	O
Application Signalling						
Application Signalling	10, "Application signalling"	M	M	M	M	M

# How do other specs solve this?

## Example: DVB-GEM

- Selecting a target specific subset of specification features
- Additional normative requirements / constraints
- Additional clarifications of ambiguities

256

ETSI TS 102 728 V1.1.1 (2010-01)

Where both an `organisation_id` and `application_id` are combined into an application identifier, they will be represented as a single hexadecimal number using the previously described encoding with the `organisation_id` as the most significant bits and the `application_id` as the least significant bits.

### 14.6 Filename requirements

#### 14.6.1 Persistent storage

Receivers shall support path and file names as specified by `persistentsubstring` and `persistentsubstring` in the following BNF:

```

lowalpha = "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" |
           "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" |
           "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"

upalpha = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" |
          "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" |
          "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"

alpha = lowalpha | upalpha

digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

punct = "_"

persistentsubstring := alpha | digit | punct

persistentsubstring := persistentsubstring | persistentsubstring persistentsubstring

persistentsubstring = persistentsubstring
                    | persistentsubstring "."
                    | persistentsubstring "." persistentsubstring
persistentsubstring = persistentsubstring | persistentsubstring "/" persistentsubstring

persistentsubstring = persistentsubstring | persistentsubstring persistentsubstring

```

Receivers are required to support:

- `persistentsubstrings` of length less than or equal to 8 characters.
- `persistentsubstringsuffixes` of length less than or equal to 3 characters.

Receivers are not required to reject filenames which exceed these requirements but applications using such filenames are not compliant. These restrictions do not apply to stored applications.

Receivers shall have filesystems for persistent storage which are either case sensitive or "case preserving". Applications shall be written to work on both of these. "case preserving" filesystems are case insensitive when opening an existing file but preserve the case which was used when the file was initially created.

# How do other specs solve this?

## Example: DVB-GEM

- Selecting a target specific subset of specification features
- Additional normative requirements / constraints
- Additional clarifications of corner cases / ambiguities

### A.2.6 Simple Section Filter

The description of the `getSection` method shall be considered to have the following text added:

- This method shall return null if no section has been successfully filtered for this instance and the owning `SectionFilterGroup` is not attached.

### A.2.7 `org.davic.media`

#### A.2.7.1 `FreezeControl.resume()`

Add the following to the description of the semantics for this method:

- If the player is started and if decoding of the media stream is not frozen then calls to this method shall have no effect.
- If the player is not started then the exception shall be thrown.

#### A.2.7.2 `MediaTimePositionChangedEvent`

Add the following constructor:

```
MediaTimePositionChangedEvent (
    Controller from,
    int previous,
    int current,
    int target,
    Time mediaTime)
```

With the following definition of parameters:

Parameters:

- `from` - the controller whose media position was changed;
- `previous` - the state the controller was in before this event;
- `current` - the state the controller was in at the time the event was generated;
- `target` - the state that the controller is heading to;
- `mediaTime` - the media time after the change.

# Scenarios, Use Cases and Requirements

# Key Scenarios

- As an end user, I want to know whether a device will work with my system before I purchase it to avoid wasting money.
  - Installers of IoT devices want to be able to determine if a given device will be compatible with the rest of their installed systems and whether they will have access to its data and affordances.
- As a developer, I want TDs to be as simple as possible so that I can efficiently develop them.
  - Here "simple" should relate to the end goal, "efficiently develop"; that is, TDs should be straightforward for the average developer to complete and validate.
- As a developer, I want to be able to validate that a Thing will be compatible with a Consumer without having to test against every possible consumer.

# Use Cases

## Multi-Vendor System Integration

- Out of the box interoperability of devices. [wot-profile.md](#)
- Digital twin to analyze and troubleshoot physical assets in real time, predict future problems, minimize downtime, and perform simulations. [digital-twin.md](#)
- Multi vendor and protocol interoperability by communicating across different protocols. [X-Protocol-Interworking.md](#)

Other use cases benefit from having profiles too.

# Proposed Requirements (under discussion)

Interoperability

Limit and reduce complexity

Ambiguities

Human readability

Developer guidance

Multiple profiles

Composable profiles

Validatable TDs

Identification of profiles

Profile should define a finite set of features and capabilities to implement by the consumer.

Limit resource consumption

Follow Security and Privacy Best Practices

Developer Mode

See: <https://github.com/w3c/wot-profile/blob/master/REQUIREMENTS/requirements.md>

# Strawman Proposal



# Objective

- Define a profile on basis of the published TD and architecture specifications.
- Drive WoT adoption by publishing the profile as soon as possible.
- Don't invent new features, rather constrain and clarify existing specifications.

# WoT Profile

The **WoT Profile** specification serves two purposes:

## Generic Profiling Mechanism

- to describe a profile in an unambiguous way. This mechanism can be used to define additional profiles.

## Core Profile

- Define a subset the Thing Description for use with selected protocols.
- Formalize the results of several plug-fests that were conducted by the WoT Interest Group and of tests that were conducted as part of the development.
- It is expected that additional profiles for thing templates and other protocols will be defined in the near future.

# Profiling mechanism

- Constraints on the vocabulary of TD classes
  - Make specific vocabulary terms mandatory
- Constraints on class relationships
  - limited cardinality, e.g. only one form per interaction affordance
- Constraints on values on vocabulary terms
  - e.g. only a single string, where string or array of strings is permitted
- Constraints on data schemas
  - e.g. no arbitrary nested objects, arrays of arrays
- Constraints on security
  - Security mechanisms are selected only at top level

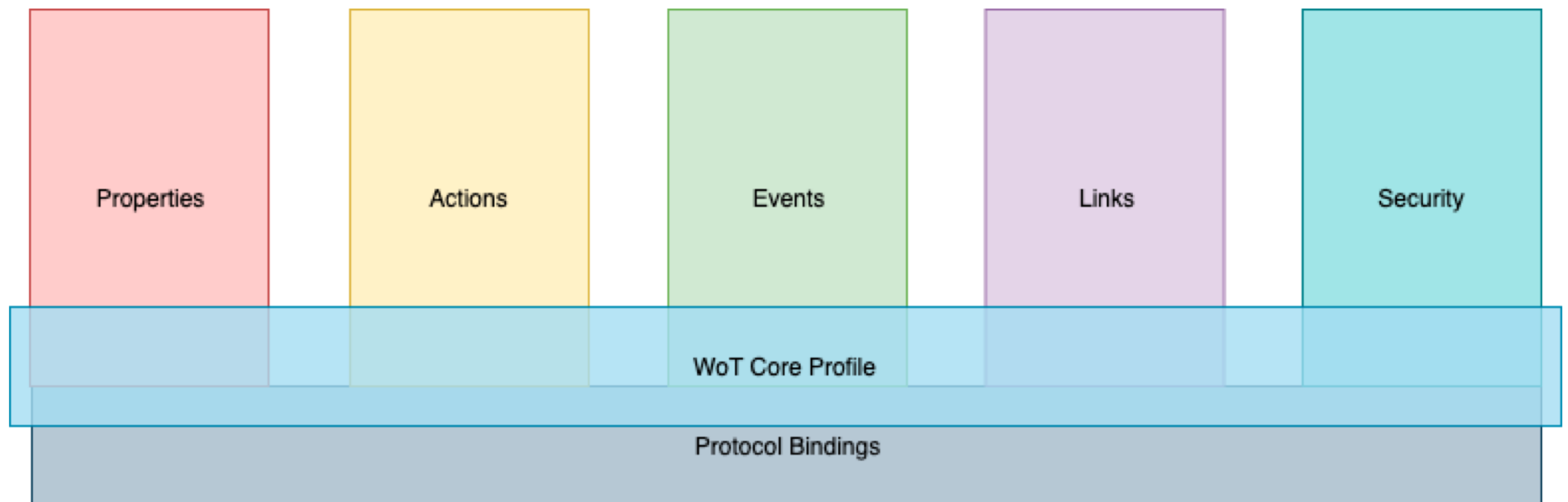
# Core device profile

- Guarantees a minimum level of interoperability
  - Reading and writing of properties
  - Invoking actions
- How?
  - Formalize proven interoperability results of the plug-fests
  - Define additional constraints, e.g. no „at risk“ features
  - Focus on „easy to implement“

# Core Profile Proposal

- Constraints on the protocol binding - Default Binding Profile for HTTP(S).
- Pre-defined mapping of http verbs to operations, e.g. PUT to writeproperty, POST with return payload for invokeaction, etc.
- Only a single “Forms” endpoint per interaction affordance.
- Constrained set of data types (e.g. no arrays of arrays) in addition to constrained payload structure.
- Limited subprotocol(s) to handle observe and async events.
- Only a single security profile at top level for a TD.
- Additional profiles may be developed that allow (require) protocol, payload, and data type adaptation.

# WoT Core Profile



# Profile Status

- Last year a strawman proposal was submitted
  - Includes a generic profiling mechanism
- Architecture TF recently focused on Use Cases / Lifecycle
  - -> work stalled for several months
- Profile work will be resumed at upcoming (virtual) WoT F2F

# Strawman Walkthrough

Strawman proposal: <http://w3c.github.io/wot-profile/>



# References

- WoT Architecture
  - <https://github.com/w3c/wot-architecture>
- Profile Repo @ Github
  - <https://github.com/w3c/wot-profile>
- Strawman proposal:
  - <http://w3c.github.io/wot-profile/>