Internet Architecture Board (IAB)
Request for Comments: 8477
Category: Informational
ISSN: 2070-1721

J. Jimenez
H. Tschofenig
D. Thaler

October 2018

Report from the Internet of Things (IoT) Semantic Interoperability (IOTSI) Workshop 2016

Abstract

This document provides a summary of the "Workshop on Internet of Things (IoT) Semantic Interoperability (IOTSI)", which took place in Santa Clara, California March 17-18, 2016. The main goal of the workshop was to foster a discussion on the different approaches used by companies and Standards Developing Organizations (SDOs) to accomplish interoperability at the application layer. This report summarizes the discussions and lists recommendations to the standards community. The views and positions in this report are those of the workshop participants and do not necessarily reflect those of the authors or the Internet Architecture Board (IAB), which organized the workshop. Note that this document is a report on the proceedings of the workshop. The views and positions documented in this report are those of the workshop participants and do not necessarily reflect IAB views and positions.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Architecture Board (IAB) and represents information that the IAB has deemed valuable to provide for permanent record. It represents the consensus of the Internet Architecture Board (IAB). Documents approved for publication by the IAB are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8477.

Table of Contents

1. Introduction

The Internet Architecture Board (IAB) holds occasional workshops designed to consider long-term issues and strategies for the Internet, and to suggest future directions for the Internet architecture. The investigated topics often require coordinated efforts from many organizations and industry bodies to improve an identified problem. One of the targets of the workshops is to establish communication between relevant organizations, especially when the topics are out of the scope of the Internet Engineering Task Force (IETF). This long-term planning function of the IAB is complementary to the ongoing engineering efforts performed by working groups of the IETF.

With the expansion of the Internet of Things (IoT), interoperability becomes more and more important. Standards Developing Organizations (SDOs) have done a tremendous amount of work to standardize new protocols and profile existing protocols.

At the application layer and at the level of solution frameworks, interoperability is not yet mature. Particularly, the work on data formats (in the form of data models and information models) has not seen the same level of consistency throughout SDOs.

One common problem is the lack of an encoding-independent standardization of the information, the so-called information model. Another problem is the strong relationship between data formats and the underlying communication architecture, such as a design in Remote Procedure Call (RPC) style or a RESTful design (where REST refers to Representational State Transfer). Furthermore, groups develop solutions that are very similar on the surface but differ slightly in their standardized outcome, leading to interoperability problems. Finally, some groups favor different encodings for use with various application-layer protocols.

Thus, the IAB decided to organize a workshop to reach out to relevant stakeholders to explore the state of the art and identify commonality and gaps [IOTSIAG] [IOTSIWS]. In particular, the IAB was interested to learn about the following aspects:

- What is the state of the art in data and information models? What should an information model look like?
- What is the role of formal languages, such as schema languages, in describing information and data models?
- What is the role of metadata, which is attached to data to make it self-describing?

- How can we achieve interoperability when different organizations, companies, and individuals develop extensions?
- What is the experience with interworking various data models developed from different groups, or with data models that evolved over time?
- What functionality should online repositories for sharing schemas have?
- How can existing data models be mapped against each other to offer interworking?
- Is there room for harmonization, or are the use cases of different groups and organizations so unique that there is no possibility for cooperation?
- How can organizations better work together to increase awareness and information sharing?

2. Terminology

The first roadblock to interoperability at the level of data models is the lack of a common vocabulary to start the discussion. [RFC3444] provides a starting point by separating conceptual models for designers, or "information models", from concrete detailed definitions for implementers, or "data models". There are concepts that are undefined in that RFC and elsewhere, such as the interaction with the resources of an endpoint, or "interaction model". Therefore, the three "main" common models that were identified were:

Information Model
: An information model defines an environment at the highest level of abstraction and expresses the desired functionality. Information models can be defined informally (e.g., in prose) or more formally (e.g., Unified Modeling Language (UML), Entity-Relationship Diagrams, etc.). Implementation details are hidden.

Data Model
: A data model defines concrete data representations at a lower level of abstraction, including implementation- and protocol-specific details. Some examples are SNMP Management Information Base (MIB) modules, World Wide Web Consortium (W3C) Thing Description (TD) Things, YANG modules, Lightweight Machine-to-Machine (LwM2M) Schemas, Open Connectivity Foundation (OCF) Schemas, and so on.

Interaction Model
: An interaction model defines how data is accessed and retrieved from the endpoints, being, therefore, tied to the specific communication pattern that the system has (e.g., REST methods, Publish/Subscribe operations, or RPC calls).

Another identified terminology issue is the semantic meaning overload that some terms have. The meaning can vary depending on the context in which the term is used. Some examples of such terms are as follows: semantics, models, encoding, serialization format, media types, and encoding types. Due to time constraints, no concrete terminology was agreed upon, but work will continue within each organization to create various terminology documents. The participants agreed to set up a GitHub repository [IOTSIGIT] for sharing information.

3. What Problems to Solve

The participants agreed that there is not simply a single problem to be solved but rather a range of problems. During the workshop, the following problems were discussed:

- Formal Languages for Documentation Purposes

To simplify review and publication, SDOs need formal descriptions of their data and interaction models. Several of them use a tabular representation found in the specification itself but use a formal language as an alternative way of describing objects and resources for formal purposes. Some examples of formal language use are as follows.

The Open Mobile Alliance (OMA), now OMA SpecWorks, used an XML Schema [LWM2M-Schema] to describe their object and resource definitions. The XML files of standardized objects are available for download at [OMNA].

The Bluetooth Special Interest Group (SIG) defined Generic Attribute Profile (GATT) services and characteristics for use with Bluetooth Smart/Low Energy. The services and characteristics are shown in a tabular form on the Bluetooth SIG website [SIG] and are defined as XML instance documents.

The Open Connectivity Foundation (OCF) uses JSON Schemas to formally define data models and RESTful API Modeling Language (RAML) to define interaction models. The standard files are available online at <oneIoTa.org>.

The AllSeen Alliance uses AllJoyn Introspection XML to define data and interaction models in the same formal language, tailored for RPC-style interaction. The standard files are available online on the AllSeen Alliance website, but both standard and vendor-defined model files can be obtained by directly querying a device for them at runtime.

The World Wide Web Consortium (W3C) uses the Resource Description Framework (RDF) to define data and interaction models using a format tailored for the web.

The Internet Engineering Task Force (IETF) uses YANG to define data and interaction models. Other SDOs may use various other formats.

o Formal Languages for Code Generation

Code-generation tools that use formal data and information modeling languages are needed by developers. For example, the AllSeen Visual Studio Plugin [AllSeen-Plugin] offers a wizard to generate code based on the formal description of the data model. Another example of a data modeling language that can be used for code generation is YANG. A popular tool to help with code generation of YANG modules is pyang [PYANG]. An example of a tool that can generate code for multiple ecosystems is OpenDOF [OpenDOF]. Use cases discussed for code generation included easing development of server-side device functionality, clients, and compliance tests.

o Debugging Support

Debugging tools are needed that implement generic object browsers, which use standard data models and/or retrieve formal language descriptions from the devices themselves. As one example, the nRF Bluetooth Smart sniffer from Nordic Semiconductor [nRF-Sniffer] can be used to display services and characteristics defined by the Bluetooth SIG. As another example, AllJoyn Explorer [AllJoynExplorer] can be used to browse and interact with any resource exposed by an AllJoyn device, including both standard and vendor-defined data models, by retrieving the formal descriptions from the device at runtime.

o Translation

The working assumption is that devices need to have a common data model with a priori knowledge of data types and actions. However, that would imply that each consortium/organization will try to define their own data model. That would cause a major interoperability

problem, possibly a completely intractable one given the number of variations, extensions, compositions, or versioning changes that will happen for each data model.

Another potential approach is to have a minimal amount of information on the device to allow for a runtime binding to a specific model, the objective being to require as little prior knowledge as possible.

Moreover, gateways, bridges and other similar devices need to dynamically translate (or map) one data model to another one. Complexity will increase as there are also multiple protocols and schemas that make interoperability harder to achieve.

o Runtime Discovery

Runtime discovery allows IoT devices to exchange metadata about the data, potentially along with the data exchanged itself. In some cases, the metadata not only describes data but also the interaction model as well. An example of such an approach has been shown with Hypermedia as the Engine of Application State (HATEOAS) [HATEOAS]. Another example is that all AllJoyn devices support such runtime discovery using a protocol mechanism called "introspection", where the metadata is queried from the device itself [AllSeen].

There are various models, whether deployed or possible, for such discovery. The metadata might be extracted from a specification, looked up on a cloud repository (e.g., oneIoTa for OCF models), looked up via a vendor's site, or obtained from the device itself (such as in the AllJoyn case). The relevant metadata might be obtained from the same place or different pieces might be obtained from different places, such as separately obtaining (a) syntax information, (b) end-user descriptions in a desired language, and (c) developer-specific comments for implementers.

4. Translation

In an ideal world where organizations and companies cooperate and agree on a single data model standard, there is no need for gateways that translate from one data model to another one. However, this is far from reality today, and there are many proprietary data models in addition to the already standardized ones. As a consequence, gateways are needed to translate between data models. This leads to (n^2)-n combinations, in the worst case.

There are analogies with gateways back in the 1980s that were used to translate between network layer protocols. Eventually, IP took over, providing the necessary end-to-end interoperability at the network layer. Unfortunately, the introduction of gateways leads to the loss

of expressiveness due to the translation between data models. The functionality of IP was so valuable in the market that advanced features of other networking protocols became less attractive and were not used anymore.

Participants discussed an alternative that they called a "red star", shown in Figure 1, where data models are translated to a common data model shown in the middle. This reduces the number of translations that are needed down to 2n (in the best case). The problem, of course, is that everyone wants their own data model to be the red star in the center.

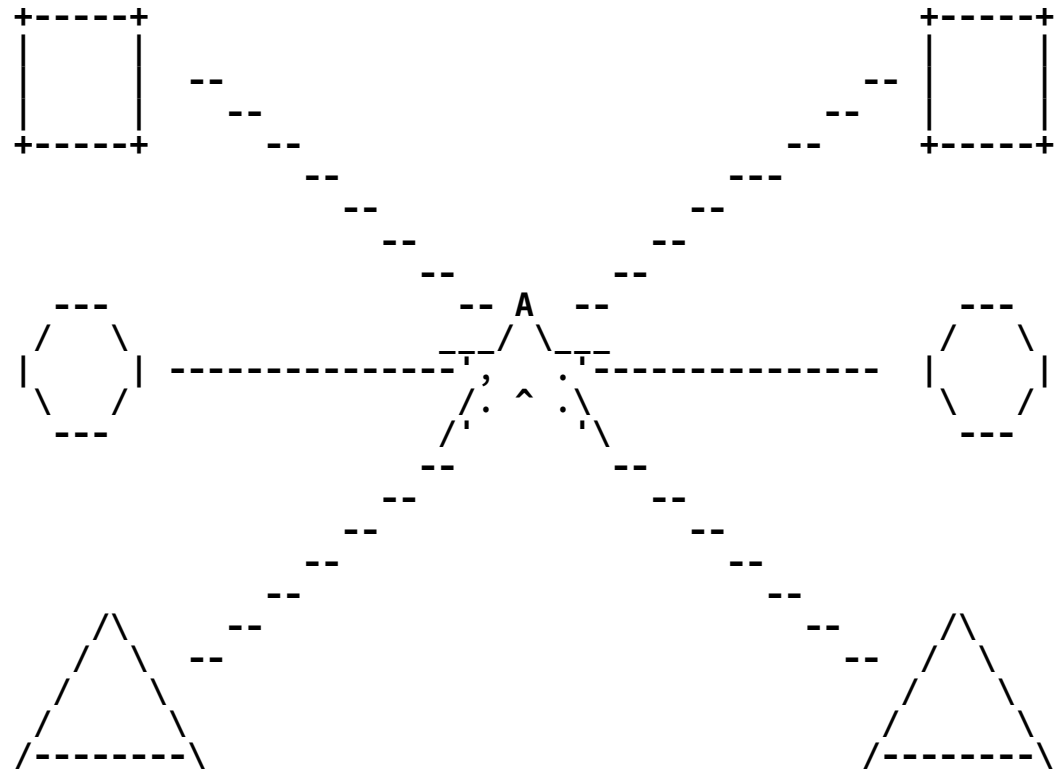

Figure 1: The "Red Star" in Data/Information Models

While the workshop itself was not a suitable forum to discuss the design of such translation in detail, several questions were raised:

- Do we need a "red star" that does everything, or could we design something that offers a more restricted functionality?
- How do we handle loss of data and functionality?

- Should data be translated between data models, or should data models themselves be translated?

- How can interaction models be translated? They need to be dealt with in addition to the data models.

- Many (if not all) data and interaction models have some bizarre functionality that cannot be translated easily. How can those be handled?

- What limitations are we going to accept in these translations?

The participants also addressed the question of when translation should be done. Two use cases were discussed:

(a) Design time: A translation between data model descriptions, such as translating a YANG module to a RAML/JSON model, can be performed once, during design time. A single information model might be mapped to a number of different data models.

(b) Run time: Runtime translation of values in two standard data models can only be algorithmically done when the data model on one side is algorithmically derived from the data model on the other side. This was called a "derived model". It was discussed that the availability of runtime discovery can aid in semantic translation, such as when a vendor-specific data model on one side of a protocol bridge is resolved and the translator can algorithmically derive the semantically equivalent vendor-specific data model on the other side. This situation is discussed in [BridgeTaxonomy].

The participants agreed that algorithm translation will generally require custom code whenever one is translating to anything other than a derived model.

Participants concluded that it is typically easier to translate data between systems that follow the same communication architecture.

5. Dealing with Change

A large part of the workshop was dedicated to the evolution of devices and server-side applications. Interactions between devices and services and how their relationship evolves over time is complicated by their respective interaction models.

The workshop participants discussed various approaches to deal with change. In the most basic case, a developer might use a description of an API and implement the protocol steps. Sometimes, the data or

information model can be used to generate code stubs. Subsequent changes to an API require changes on the clients to upgrade to the new version, which requires some development of new code to satisfy the needs of the new API.

These interactions could be made machine understandable in the first place, enabling for changes to happen at runtime. In that scenario, a machine client could discover the possible interactions with a service, adapting to changes as they occur without specific code being developed to adapt to them.

The challenge seems to be to code the human-readable specification into a machine-readable format. Machine-readable languages require a shared vocabulary to give meaning to the tags.

These types of interactions are often based on the REST architectural style. Its principle is that a device or endpoint only needs a single entry point, with a host providing descriptions of the API in-band by means of web links and forms.

By defining IoT-specific relation types, it is possible to drive interactions through links instead of hard-coding URIs into a RESTful client, thus making the system flexible enough for later changes. The definition of the basic hypermedia formats for IoT is still a work in progress. However, some of the existing mechanisms can be reused, such as resource discovery, forms, or links.

6. IANA Considerations

This document has no IANA actions.

7. Security Considerations

There were two types of security considerations discussed: use of formal data models for security configuration and security of data and data models in general.

It was observed that the security assumptions and configuration, or "security model", varies by ecosystem today, making the job of a translator difficult. For example, there are different types of security principals (e.g., user vs. device vs. application), the use of Access Control Lists (ACLs) versus capabilities, and what types of policies can be expressed, all vary by ecosystem. As a result, the security model architecture generally dictates where translation can be done.

One approach discussed was whether two endpoints might be able to use some overlay security model across a translator between two ecosystems, which only works if the two endpoints agree on a common data model for their communication. Another approach discussed was simply having a translator act as a trusted intermediary, which enables the translator to translate between different data models.

One suggestion discussed was either adding metadata into the formal data model language or having it accompany the data values over the wire, tagging the data with privacy levels. However, sometimes even the privacy level of information might itself be sensitive. Still, it was observed that being able to dynamically learn security requirements might help provide better UIs and translators.

8. Collaboration

The participants discussed how best to share information among their various organizations. One discussion was around having joint meetings. One current challenge reported was that organizations were not aware of when and where each other's meetings were scheduled, and sharing such information could help organizations better collocate meetings. To facilitate this exchange, the participants agreed to add links to their respective meeting schedules from a common page in the IOTSI repository [IOTSIGIT].

Another challenge reported was that organizations did not know how to find each other's published data models, and sharing such information could better facilitate reuse of the same information model. To facilitate this exchange, the participants discussed whether a common repository might be used by multiple organizations. The OCF's oneIoTa repository was discussed as one possibility, but it was reported that its terms of use at the time of the workshop prevented this. The OCF agreed to take this back and look at updating the terms of use to allow other organizations to use it, as the restriction was not the intent. <schema.org> was discussed as another possibility. In the meantime, the participants agreed to add links to their respective repositories from a common page in the IOTSI repository [IOTSIGIT].

It was also agreed that the iotsi@iab.org mailing list would remain open and available for sharing information between all relevant organizations.

9. Informative References

[AllJoynExplorer]
Microsoft, "AllJoyn".

[AllSeen] Thaler, D., "Summary of AllSeen Alliance Work Relevant to Semantic Interoperability", 2016, <https://www.iab.org/wp-content/IAB-uploads/2016/03/AllSeen-summary-IOTSI.pdf>.

[AllSeen-Plugin]
Rockwell, B., "Using the AllJoyn Studio Extension", August 2015.

[BridgeTaxonomy]
Thaler, D., "IoT Bridge Taxonomy", IAB IOTSI Workshop 2016, <https://www.iab.org/wp-content/IAB-uploads/2016/03/DThaler-IOTSI.pdf>.

[HATEOAS] Kovatsch, M., Hassan, Y., and K. Hartke, "Semantic Interoperability Requires Self-describing Interaction Models: HATEOAS for the Internet of Things", Proceedings of the IAB IoT Semantic Interoperability Workshop 2016, <https://www.iab.org/wp-content/IAB-uploads/2016/03/2016-IAB-HATEOAS.pdf>.

[IOTSIAG] IAB, "IoT Semantic Interoperability Workshop Agenda", 2016, <https://www.iab.org/activities/workshops/iotsi/agenda/>.

[IOTSIGIT]
"Starting place for the IoT Semantic Interoperability Workshop (IOTSI) Information Resource", commit ff21f74, July 2018, <https://github.com/iotsi/iotsi>.

[IOTSIWS] IAB, "IoT Semantic Interoperability Workshop 2016", 2016, <https://www.iab.org/activities/workshops/iotsi/>.

[LWM2M-Schema]
OMA, "LWM2M XML Schema - LWM2M Editor Schema", July 2018.

[nRF-Sniffer]
Nordic Semiconductor, "nRF Sniffer: Smart/Bluetooth low energy packet sniffer".

[OMNA] OMA, "OMA LightweightM2M (LwM2M) Object and Resource Registry".

[OpenDOF] OpenDOF, "The OpenDOF Project", <https://opendof.org>.

[PYANG] "An extensible YANG validator and converter in python", commit 15c807f, September 2018, <https://github.com/mbj4668/pyang>.

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003, <https://www.rfc-editor.org/info/rfc3444>.

[SIG] Bluetooth SIG, "GATT Specifications", <https://www.bluetooth.com/specifications/gatt>.

Appendix A. Program Committee

This workshop was organized by the following individuals: Jari Arkko, Ralph Droms, Jaime Jimenez, Michael Koster, Dave Thaler, and Hannes Tschofenig.

Appendix B. Accepted Position Papers

- Jari Arkko, "Gadgets and Protocols Come and Go, Data Is Forever"
- Carsten Bormann, "Noise in Specifications hurts"
- Benoit Claise, "YANG as the Data Modelling Language in the IoT space"
- Robert Cragie, "The ZigBee Cluster Library over IP"
- Dee Denteneer, Michael Verschoor, and Teresa Zotti, "Fairhair: interoperable IoT services for major Building Automation and Lighting Control ecosystems"
- Universal Devices, "Object Oriented Approach to IoT Interoperability"
- Bryant Eastham, "Interoperability and the OpenDOF Project"
- Stephen Farrell and Alissa Cooper, "It's Often True: Security's Ignored (IOTSI) - and Privacy too"
- Christian Groves, Lui Yan, and Yang Weiwei, "Overview of IoT semantics landscape"
- Ted Hardie, "Loci of Interoperability for the Internet of Things"
- Russ Housley, "Vehicle-to-Vehicle and Vehicle-to-Infrastructure Communications"
- Jaime Jimenez, Michael Koster, and Hannes Tschofenig, "IPSO Smart Objects"
- David Jones, "IOTDB - interoperability Through Semantic Metastandards"
- Sebastian Kaebisch and Darko Anicic, "Thing Description as Enabler of Semantic Interoperability on the Web of Things"

- Achilleas Kemos, "Alliance for Internet of Things Innovation Semantic Interoperability Release 2.0, AIOTI WG03 - IoT Standardisation"

- Ari Keraenen and Cullen Jennings, "SenML: simple building block for IoT semantic interoperability"

- Dongmyoung Kim, Yunchul Choi, and Yonggeun Hong, "Research on Unified Data Model and Framework to Support Interoperability between IoT Applications"

- Michael Koster, "Model-Based Hypertext Language"

- Matthias Kovatsch, Yassin N. Hassan, and Klaus Hartke, "Semantic Interoperability Requires Self-describing Interaction Models"

- Kai Kreuzer, "A Pragmatic Approach to Interoperability in the Internet of Things"

- Barry Leiba, "Position Paper"

- Marcello Lioy, "AllJoyn"

- Kerry Lynn and Laird Dornin, "Modeling RESTful APIs with JSON Hyper-Schema"

- Erik Nordmark, "Thoughts on IoT Semantic Interoperability: Scope of security issues"

- Open Geospatial Consortium, "OGC SensorThings API: Communicating "Where" in the Web of Things"

- Jean Paoli and Taqi Jaffri, "IoT Information Model Interoperability: An Open, Crowd-Sourced Approach in Three Parallel Parti"

- Joaquin Prado, "OMA Lightweight M2M Resource Model"

- Dave Raggett and Soumya Kanti Datta, "Input paper for IAB Semantic Interoperability Workshop"

- Pete Rai and Stephen Tallamy, "Semantic Overlays Over Immutable Data to Facilitate Time and Context Specific Interoperability"

- Jasper Roes and Laura Daniele, "Towards semantic interoperability in the IoT using the Smart Appliances REFerence ontology (SAREF) and its extensions"

- Max Senges, "Submission for IAB IoT Sematic Interoperability workshop"
- Bill Silverajan, Mert Ocak and Jaime Jimenez, "Implementation Experiences of Semantic Interoperability for RESTful Gateway Management"
- Ned Smith, Jeff Sedayao, and Claire Vishik, "Key Semantic Interoperability Gaps in the Internet-of-Things Meta-Models"
- Robert Sparks and Ben Campbell, "Considerations for certain IoT-based services"
- J. Clarke Stevens, "Open Connectivity Foundation oneIoTa Tool"
- J. Clarke Stevens and Piper Merriam, "Derived Models for Interoperability Between IoT Ecosystems"
- Ravi Subramaniam, "Semantic Interoperability in Open Connectivity Foundation (OCF) - formerly Open Interconnect Consortium (OIC)"
- Andrew Sullivan, "Position paper for IOTSI workshop"
- Darshak Thakore, "IoT Security in the context of Semantic Interoperability"
- Dave Thaler, "IoT Bridge Taxonomy"
- Dave Thaler, "Summary of AllSeen Alliance Work Relevant to Semantic Interoperability"
- Mark Underwood, Michael Gruninger, Leo Obrst, Ken Baclawski, Mike Bennett, Gary Berg-Cross, Torsten Hahmann, and Ram Sriram, "Internet of Things: Toward Smart Networked Systems and Societies"
- Peter van der Stok and Andy Bierman, "YANG-Based Constrained Management Interface (CoMI)"

Appendix C. List of Participants

Andy Bierman, YumaWorks
Carsten Bormann, Uni Bremen/TZI
Ben Campbell, Oracle
Benoit Claise, Cisco
Alissa Cooper, Cisco
Robert Cragie, ARM Limited
Laura Daniele, TNO
Bryant Eastham, OpenDOF
Christian Groves, Huawei
Ted Hardie, Google
Yonggeun Hong, ETRI
Russ Housley, Vigil Security
David Janes, IOTDB
Jaime Jimenez, Ericsson
Shailendra Karody, Catalina Labs
Ari Keraenen, Ericsson
Michael Koster, SmartThings
Matthias Kovatsch, Siemens
Kai Kreuzer, Deutsche Telekom
Barry Leiba, Huawei
Steve Liang, Uni Calgary
Marcello Lioy, Qualcomm
Kerry Lynn, Verizon
Mayan Mathen, Catalina Labs
Erik Nordmark, Arista
Jean Paoli, Microsoft
Joaquin Prado, OMA
Dave Raggett, W3C
Max Senges, Google
Ned Smith, Intel
Robert Sparks, Oracle
Ram Sriram, NIST
Clarke Stevens
Ram Subramanian, Intel
Andrew Sullivan, DIN
Darshak Thakore, CableLabs
Dave Thaler, Microsoft
Hannes Tschofenig, ARM Limited
Michael Verschoor, Philips Lighting

IAB Members at the Time of Approval

Jari Arkko
Alissa Cooper
Ted Hardie
Christian Huitema
Gabriel Montenegro
Erik Nordmark
Mark Nottingham
Melinda Shore
Robert Sparks
Jeff Tantsura
Martin Thomson
Brian Trammell
Suzanne Woolf

Acknowledgements

We would like to thank all paper authors and participants for their contributions and Ericsson for hosting the workshop.

Authors' Addresses

Jaime Jimenez

Email: jaime.jimenez@ericsson.com

Hannes Tschofenig

Email: hannes.tschofenig@arm.com

Dave Thaler

Email: dthaler@microsoft.com