Layered Architecture for IoT Interoperability above Connectivity

Jianfei He, Bingyi Guo
Huawei Technologies Co. Ltd.
Email: {jeffrey.he, guobingyi}@huawei.com

I. Internet-wide Federation for IoT

Internet of Things (IoT) interoperability can be investigated from the protocol stack perspective [1]. The key open issue of interoperability is at the traditional application layer, which is above TCP/IP connectivity. A framework should be firstly established as a big picture as a base of detail discussion. The following aspects should be taken into account for such a framework.

- **Internet-wide federation:** The IoT networks are expected to be built and administrated by many organizations independently. Therefore, Internet of things is composed of these autonomous domains peering together. In this sense, the IoT world should be an "Internet-wide federation". Some design principles of current Internet are applicable to IoT.
- **Gateway with information repository:** "Things" could be as powerful as PCs or smart phones. But for a more general framework, it is better to assume that "things" are resource-constrained devices such as sensors. Usually, these devices connect to outside world through gateways, that is different from the current Internet based on E2E framework. Current Internet assumes that networks provide IP-based connectivity and terminals implement all application functions. In the IoT world, the gateways may not only play as a role of proxy to translate upper-layer protocols, but also be a repository of information generated by "things" to serve the requests even when these things are not online.
- **Data processing in Cloud:** Usually, issues about semantics only happen in machine-to-machine communication. For human-to-machine communication, existing mechanisms such as HTML/WWW are sufficient. For example, one IoT gateway can provide web-based services as a way for people to get information and even to operate things. The ambiguity of semantics can be resolved by human intelligence. As far as semantic interoperability is concerned, the assumption is that "machines" should understand semantics as the base of further automatic processing. In an era of cloud computing, without loss of generality, it could be supposed that the data processing functions are in the "Clouds".

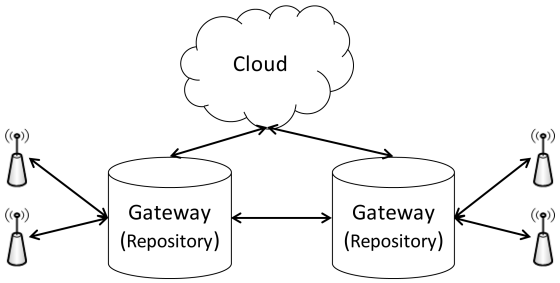

Fig. 1. The framework of IoT.

Therefore, a general framework of IoT is proposed as illustrated in Figure.1 where (1) information from things is shared via gateways, (2) gateways and the Clouds processing data are federated.

II. Layered Architecture for Internet-wide IoT Federation

With the aforementioned big picture, this position paper emphasizes that the peering interfaces between IoT gateways and Clouds should keep stable so that this federation could grow to be Internet-wide. If these interfaces have to change with the introduction of new services, this Internet-wide federation will break into fragments because

it is difficult for all the domains to update to a consensus version. The XMPP federation of instant messaging is an example, according to comments on Matrix: "*Baseline feature set is so minimal that fragmentation of features between clients and servers is common, especially as interoperability profiles for features have fallen behind (as of July 2015)*" [2].

At an architectural level, the design principles of current Internet may be helpful to solve this issue. Current Internet is based on layered architecture and "end-to-end argument" [3]. It is discussed in detail in the book "Internet Architecture and Innovation" [4]:" *a function or service should be carried out within a network layer only if it is needed by all clients of that layer, and it can be completely implemented in that layer*. This argument suggests that when we introduce new functions in an infrastructure like Internet, these functions should be fundamental enough that would be used by most of client use cases. In IoT scenario, this paper proposes that the interoperability (above the connectivity) involves two kinds of functions, which can be decoupled into two layers as show in Figure.2: (1) how to share and access information (Information Sharing Layer), (2) how the applications understand the specific semantics of the information. Correspondingly, information model for semantic interoperability should also reflect this two-layer architecture.

Information Sharing Layer can be the basis of a stable infrastructure, which are less semantics-sensitive, just like file systems and database systems. As a result, the functions and corresponding interfaces in this layer can keep stable for a long term, just like the current TCP/IP in Internet. However, these functions probably are more than the straightforward operations such as Create, Retrieve, Update and Delete (CRUD), because this layer aims to convey various and evolving applications above it. For example, this layer may include access control for security and privacy, message and file sharing, and so on. It shouldn't be underestimated the difficulty to define a function set above connectivity, which is generic but still useful in a real IoT world. But it is worth exploring this way, which also aligns with the design philosophy of current Internet, i.e., "*A key concept of the Internet is that it was not designed for just one application, but as a general infrastructure on which new applications could be conceived, as illustrated later by the emergence of the World Wide Web*" [5].

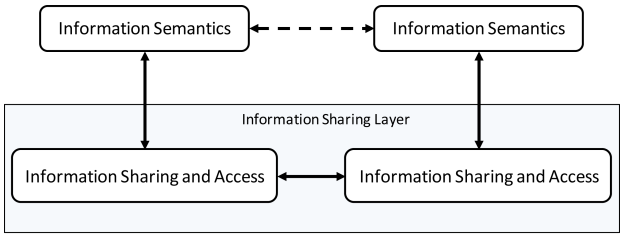

Fig. 2. Two-layer architecture for interoperability.

Information semantics above the infrastructure is dynamically evolving, by nature. New kinds of semantics come with new "things" such as sensors of new pollution metrics. This upper layer should embrace the evolution and innovation of different domain-specific protocols and languages in different use cases. In the long term, the evolution may have a conclusion that whether one best solution dominates, or several alternatives coexist for different scenarios.

By using two-layer architecture for interoperating of IoT rather than mixing into one layer, two goals can be realized at the same time: (1) information can be shared among IoT domains and clouds to avoid IoT silos, (2) keeping the infrastructure stable to realize an Internet-wide federation. Information semantics in IoT will naturally change with new things and new context emerging. This is inevitable for evolving IoT scenarios. The adaptation to new information semantics can be achieved by the entities at the "end" of the infrastructure such as new things, new software processing information in the cloud. The methodology here is very similar to current Internet based on "end-to-end argument".

III. Design of the Infrastructure

It is an open issue how to define the lower-layer functions in the two-layer architecture. Many practices in the distributed information acquisition and sharing can be used as references. For example,

- Federated user authentication, such as Eduroam [6].
- Federated instant message protocols, such as XMPP [7].
- Federated file systems [8].

However, these need to be re-investigated within a unified framework in a holistic approach, which can be described using an information model.

Just as an illustrative example, several components are considered:

- Information objects and generic attributes.
- Generic "user relationships" for information sharing like social networks.
- Relationships between information objects and users, which are essentially the access control.

Based on these components, two sets of functions are provided: 1) Operations on data, such as Create, Retrieve, Update and Delete (CRUD), at the object level and at a finer granularity like "records". 2) Data sharing services based on "user relationships" in an easy and secure way:

- Federated identity management.
- Federated authentication and authorization to fulfill the access control necessary for data sharing across domains.
- Several features to enable social-network applications:
  - A user could establish a group, then messages sent to the group will be automatically forwarded by the infrastructure to each member.
  - Users may "follow" other users. When new data are published, the infrastructure will proactively send notifications to followers.
  - Some operations may need approvals from other users, then the infrastructure will automatically send messages to request approval before the operations are fulfilled.

These functions look like a simple combination of several existing applications. But it is challenging to justify them with the "end-to-end argument": " *a function or service should be carried out within a network layer only if it is needed by all clients of that layer, and it can be completely implemented in that layer*" [4].

IV. Conclusion and Future Work

A common framework is needed for the discussion of IoT semantic interoperability. To achieve IoT as an Internet-wide federation, a stable infrastructure should be defined. Therefore, it is proposed that IoT interoperability above TCP/IP connectivity should be implemented with two layers: 1) Information Sharing Layer which is a less semantics-sensitive infrastructure, 2) the applications with dynamic semantics above the infrastructure, which may evolve with domain-specific protocols and languages. As for the infrastructure design, an illustrative example is briefly described. This is still an open issue, which is probably a suitable topic for research communities for example, IRTF, before going for a direct standardization effort.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] "http://matrix.org/blog/home/,"

[3] J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer Systems (TOCS)*, vol. 2, no. 4, pp. 277–288, 1984.

[4] B. Van Schewick, *Internet architecture and innovation*. MIT Press, 2012.

[5] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, "A brief history of the internet," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 5, pp. 22–31, 2009.

[6] T. Wolniewicz , *RFC 7593: The eduroam architecture for network roaming*, 2013.

[7] P. Saint-Andre , *RFC 6120: Extensible messaging and presence protocol (xmpp): Core (2011)*.

[8] J. Lentini, C. EVERHART, D. Ellard, R. Tewari, and M. Naik , *RFC 5716: Requirements for federated file systems*, 2010.