IoT Harmonization using XMPP

Providing trust in cross-domain smart city networks

Peter Waher
Editor IEEE P1451.99
peter.waher@ieee.org

***Abstract*—The world of Internet of Things (IoT) is comprised of a multitude of incompatible islands, separated by different protocols and communication patterns. Making security decisions in heterogenous networks is complex. This position paper describes how IEEE P1451.99 IoT Harmonization and the eXtensible Messaging and Presence Protocol (XMPP) can be used to bridge these seemingly incompatible islands in real-time and to harmonize the Internet of Things, regardless of underlying technology. XMPP is standardized by the Internet Engineering Task Force (IETF) and provides an open and free alternative to commercial or bespoke middleware platforms. It also provides a method to inject trust on which security decisions can be made, in globally scalable decentralized networks. Using this trust, resources can be monetized, incentivizing interoperation.**

***Index Terms* — IoT, Harmonization, Interoperation, Trust, Monetization, Protocols, XMPP, IEEE 1451**

I. Introduction

The Internet of Things has been dominated by commercial and bespoke enterprises, creating an avalanche of new technologies, communication protocols and standardization efforts. Most of these efforts have also been dominated by large companies willing to invest in new technology. The backside of this *ad hoc* development is that different choices result in incompatible devices. Furthermore, companies want to protect their interests and control the market, resulting in walled gardens, where efforts to promote their own technologies overshadow concerns for interoperability between existing technologies. Such concerns are mainly raised from the public investing in the technology. This approach further blocks the potential of *Smart Cities*, were the interoperability between services and things is a key enabler.

This paper presents an approach to bridge existing technologies, creating a global interoperable network of things, apt for *Smart Cities*. It does so without limiting the use of existing or future communication technologies or requiring proprietary middleware. Instead of trying to devise a new standard for communication between things, a new standard for interoperability between a plurality of communication protocols is proposed. This standard can then be implemented in any middleware or gateway, to achieve the capability to interoperate across technology or domain boundaries on the Internet. It is also shown how trust can be injected into such a heterogenous network, allowing devices, services and users to make better security decisions. Furthermore, any smart city standard must be based on open, free, scalable and proven technologies, suitable for the purpose, and without limiting the possibilities of the underlying technologies used.

II. Layers

Bridging different IoT solutions must be done on multiple levels. It is not sufficient to simply map semantics from one system to another, or from one protocol to another. Bridging must be done first on a transport level, then on a semantical level (or application layer).

The transport level includes network topology concerns: Who can connect to whom? How are messages distributed? We will denote these topics *Communication Patterns*. Different protocols support different patterns for communication.

Semantics start where communication patterns end. It relates to what operations can be performed, and how *interoperability* between entities in the network is achieved.

Arching over the Internet of Things also hangs the veritable Sword of Damocles: *Security* and *Privacy*. Without taking these into account, an infrastructure is not complete.

III. Network Topology

Different protocols impose different restrictions on network topology. Clients that connect to servers require servers to be accessible from the clients. Servers are in the general case not able to connect to its clients, for instance, due to firewall constraints. XMPP allows any client in the global federated network to reach any other client, regardless of firewalls, as long as there is consent between them. From a topology point of view, XMPP therefore allows consent-based bridging between entities in different networks that both require connectivity to the Internet, i.e. the Internet of Things, even if they reside behind different firewalls.

IV. Communication Patterns

To be able to bridge across the plurality of protocols that exist today, you need to build on a technology that supports the varying communication patterns that are in use. There are mainly four patterns in common use:

a) *Asynchronous messages* are sent from one entity to another.

b) *Request/Response*, where one entity requests information from another, which responds once for each received request.

c) *Publish/Subscribe*, where publishers publish information to a broker, optionally on a *topic*. The broker might persist the messages. It then distributes them to negotiated sets of subscribers.

d) *Multicasting* works much like *Publish/Subscribe*, but without data persistence and roles. Participants both send and receive messages in *groups*.

e) *Queues* allow data *providers* to push data to them, when available. C*onsumers* pull data from queues, when they are ready to process it.

XMPP has native support for (a)-(c), through the `message`, `iq` and `presence` *stanzas*, defined by the IETF [1]. The name for packets being transmitted in the XMPP network, is *stanza*. The presence stanza supports persistence of the last content published by a sender and does not use topics. A second Publish/Subscribe method with multiple options is available in XEP-0060 [2]. The Personal Eventing Protocol (PEP) in XEP-0163 [3] provides a simplified Publish/Subscribe pattern, suitable for devices. Multicasting is available through XEP-0045 [4]. Queues have no standardized extension yet but is straight-forward to implement. For these reasons, XMPP allows simple bridging of information using any of the well-known and well-used communication patterns available.

V. Interoperability

For the transfer of IoT-related content, interoperability interfaces exist, such as for sensor data [5], control operations [6] and concentrators ("thing of things"), integration of subsystems and bridging between protocols [7]. These were originally published by the XMPP Standards Foundation (XSF), but has been moved by their author for the purpose of managing them and their features within the IEEE IoT Harmonization working group [8] instead. Mapping of HTTP, including semantic web technologies, over XMPP, is also possible [9].

The interoperability interfaces defined for IoT in XMPP are all *loosely coupled*. They also contain sufficient meta-data to be able to encapsulate all information required, including localized information, by both devices and humans. As such, it is very easy to map existing data models from other protocols to the XMPP interfaces. It is easier to map messages in a *tightly coupled* architecture to messages in a *loosely coupled*, than vice versa.

VI. Security

XMPP has very competent support for security. Apart from performing *authentication* using SASL [10], it has a built-in consent-based *authorization* mechanism built into it, in the form of presence subscription negotiation [11], blocking of users [12] and spam reporting [13]. The authenticated identities of senders are always forwarded in stanzas, which makes spoofing very difficult. IEEE P1451.99 furthermore has support for fine-grained *provisioning* within the realm of Internet of Things [14]. This includes decision support for things in real-time, allowing owners to control who can access their devices, and do what with them, including partial permissions. It also allows for management of *ownerships* and transfers of *ownerships*, to make sure things know who their owners are [15]. Apart from transport encryption being built into XMPP, XMPP also support interoperable interfaces for *end-to-end encryption* using OTR [16], OpenPGP [17] and OLM [18]. IEEE P1451.99 also includes End-to-End encryption and Peer-to-Peer communication capabilities.

VII. Privacy

The interoperability interfaces do not require central storage of data related to things. Since such data might be related to physical individuals, it should be considered personal information, and must be treated as such. XMPP provides a means to exchange such data, without central storage, protecting the privacy of any data subjects concerned by design and by default.

VIII. Scalability

XMPP is by its design *federated*. Brokers authenticate users on their domain, and then cooperate to exchange stanzas between domains. XMPP always forwards the identities of senders of stanzas, facilitating authorization in distributed environments. Brokers validate the identities of each other to avoid the insertion of malicious brokers [19].

The federated nature of XMPP, and its extensive usage today within mobile, chat and social interaction with billions of users, makes it a good candidate as an IoT architecture and infrastructure.

IX. TRUST

Federation not only provides a globally scalable decentralized infrastructure, extensible by anyone. It also provides a means for XMPP broker operators to inject trust into the network. Entities connected to a domain are represented by an address which includes the name of the domain, much like an email address: `account@domain`. Since the authenticated address of each sender is always forwarded in all stanzas, it can be used by devices, services and users to base security decisions on. As mentioned earlier, XMPP also includes a consent-based authorization scheme called presence subscription. This mechanism is protected by the broker. Without an approved presence subscription, communication is effectively restricted. The broker can also issue tokens for distributed transactions to devices, services and users. All these can also be used as a basis of authorization. Lastly, as an electronic notary, the broker attests to the integrity of smart contracts hosted by the broker, helping digital parties to validate the consistency and integrity of legal agreements, important in autonomous cross-domain systems. Such contracts can be used, among other things, to automate decision-making and monetize resources. A Trust Provider, hosting a broker, is therefore an integral part of any cross-domain heterogenous network supporting autonomous systems (i.e. smart city). The Trust Provider injects trust into the network equal to the amount of security and integrity it manages to provide for its broker. Trust in a Trust Provider allows devices to use its decision support capabilities to make good security decisions. IEEE P1451.99 assures decisions are deterministic and made in accordance with the desires of the corresponding owners of the devices making decisions.

X. OPENNESS

XMPP is standardized by the *Internet Engineering Task Force*, and XMPP extensions by the *XMPP Standards Foundation*, a non-profit organization with a free membership. The technology is free to use and driven by a large community in individuals. IEEE P1451.99 IoT Harmonization uses XMPP to provide an infrastructure for open cross-domain interoperable networks and publishes its interfaces openly [20]. Test brokers are available [21].

XI. CONCLUSION

IEEE P1451.99 IoT Harmonization provides a mechanism to build infrastructures for autonomous cross-domain systems in open heterogenous networks. It can be used to harmonize and bridge IoT-technologies based on a variety of technologies, such as Web of Things (HTTP, CoAP), LWM2M, OneM2M, UPnP, etc., and traditional or proprietary M2M technologies such as those based on MQTT, AMQP or a myriad of other protocols, without imposing limitations on the underlying technologies, and without requiring changes be made to the underlying devices. As long as mapping is possible, reverse bridging is also possible, so that devices talking one protocol can communicate with devices somewhere else using another protocol, bridged seamlessly by XMPP in between. IEEE P1451.99 also provides data protection mechanisms to protect sensitive information and the privacy of any users involved. It provides a means for Trust Providers to inject trust into the network, and act as arbiters, providing decision support to devices that need to make security decisions in real-time, based on the desires made by slow humans. Another consequence is the possibility to monetize resources in the network, incentivizing interoperation.

REFERENCES

[1] P. Saint-Andre, “Extensible Messaging and Presence Protocol (XMPP): Core”, RFC 6120, “Instant Messaging and Presence”, RFC 6121, “Address Format”, RFC 6122, Internet Engineering Task Force (IETF).

[2] P. Millard, P. Saint-Andre, R. Meijer, “XEP-0060: Publish-Subscribe”, XMPP Standards Foundation (XSF).

[3] P. Saint-Andre, K. Smith, “XEP-0163: Personal Eventing Protocol” , XMPP Standards Foundation (XSF).

[4] P. Saint-Andre, “XEP-0045: Multi-User Chat”, XMPP Standards Foundation (XSF).

[5] P. Waher, “XEP-0323: Internet of Things – Sensor Data”, XMPP Standards Foundation (XSF).

[6] P. Waher, “XEP-0325: Internet of Things – Control”, XMPP Standards Foundation (XSF).

[7] P. Waher, “XEP-0326: Internet of Things – Concentrators”, XMPP Standards Foundation (XSF).

[8] IEEE Project “P1451.99 - Standard for Harmonization of Internet of Things (IoT) Devices and Systems”, hosted by the DASH - Devices and Systems Harmonization Working Group.

[9] P. Waher, “XEP-0332: HTTP over XMPP transport”, XMPP Standards Foundation (XSF).

[10] P. Saint-Andre, “Extensible Messaging and Presence Protocol (XMPP): Core”, RFC 6120, section 6, “SASL Negotiation”, p 77 ff.

[11] P. Saint-Andre, “Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence”, RFC 6121.

[12] P. Saint-Andre, “XEP-0191: Blocking Command”, XMPP Standards Foundation (XSF).

[13] S. Whited, “XEP-0377: Spam Reporting”, XMPP Standards Foundation (XSF).

[14] P. Waher, “XEP-0324: Internet of Things – Provisioning”, XMPP Standards Foundation (XSF).

[15] P. Waher, R. Klauck, “XEP-0347: Internet of Things – Discovery”, XMPP Standards Foundation (XSF).

[16] S. Whited, “XEP-0364: Current Off-the-Record Messaging Usage”, XMPP Standards Foundation (XSF).

[17] T. Muldowney, “XEP-0027: Current Jabber OpenPGP Usage”, XMPP Standards Foundation (XSF).

[18] A. Straub, “XEP-0384: OMEMO Encryption”, XMPP Standards Foundation (XSF).

[19] J. Miller, P. Saint-Andre, P. Hancke, “XEP-0220: Server Dialback”, XMPP Standards Foundation (XSF).

[20] “IEEE XMPP IoT Interfaces Working Group”, GitLab, https://gitlab.com/IEEE-SA/XMPPI/IoT

[21] Test integration server (broker), at https://cybercity.online/