# A Survey of Commercial Frameworks for the Internet of Things

Hasan Derhamy, Jens Eliasson and Jerker Delsing
Dept. of Computer Science, Space and Electrical Engineering
Luleå University of Technology
Luleå, Sweden
Email: hasan.derhamy@ltu.se

Peter Priller
AVL List GmbH
H. List Platz 1
8020 Graz, Austria

***Abstract*—In 2011 Ericsson and Cisco estimated 50 billion Internet connected devices by 2020, encouraged by this industry is developing application frameworks to scale the Internet of Things. This paper presents a survey of commercial frameworks and platforms designed for developing and running Internet of Things applications. The survey covers frameworks supported by big players in the software and electronics industries. The frameworks are evaluated against criteria such as architectural approach, industry support, standards based protocols and interoperability, security, hardware requirements, governance and support for rapid application development. There is a multitude of frameworks available and here a total 17 frameworks and platforms are considered. The intention of this paper is to present recent developments in commercial IoT frameworks and furthermore, identify trends in the current design of frameworks for the Internet of Things; enabling massively connected cyber physical systems.**

## I. Introduction

For more than a decade the Internet of Things (IoT) has boosted the development of standards based messaging protocols. Recently, encouraged by the likes of Ericsson and Cisco with estimates of 50 billion Internet connected devices by 2020 [1], attention has shifted from interoperability and message layer protocols towards application frameworks supporting interoperability amongst IoT product suppliers.

The IoT is the interconnection of ubiquitous computing devices for the realization of value to end users [2]. This definition encompasses "data collection" for the betterment of understanding and "automation" of tasks for optimization of time. The IoT field has evolved within application silos with domain specific technologies, such as health care, social networks, manufacturing and home automation. To achieve a truly "interconnected network of things" the challenge is enabling the combination of heterogeneous technologies, protocols and application requirements to produce an automated and knowledge based environment for the end user.

In [3], Singh et al. elaborate on three main visions for the IoT: Internet Vision, Things Vision and Semantic Vision. Depending on which vision is chosen the approach taken by a framework will differ and provide a better result for those applications. As surveyed by Perera et al. in [4], there are many existing IoT products and applications available. These however are based on proprietary frameworks which are not available for development of customized applications. The frameworks presented in this survey are all targeted as a basis for development of IoT applications.

978-1-4673-7929-8/15/$31.00 

This paper presents a survey of highly regarded commercial frameworks and platforms which are being used for Internet of Things applications. Many of the frameworks rely on high level software layers to assist in abstracting between protocols. The high level software layer provides flexibility when interconnecting between different technologies and is well suited for working in cloud environments. In some cases the frameworks look into standardizing interfaces, defining a software service bus or simply opting to choose a single network protocol and set of application protocols. This is further discussed as follows; in Section II introduces the concept of frameworks and defines three categories of frameworks used in this survey. Sections III and IV then introduces the frameworks and platforms studied, grouped by application area. In Section V a discussion of a comparative analysis of the frameworks and platforms is presented. The survey finishes with a few concluding remarks in Section VI.

## II. What is an Internet of Things Framework

Framework in the context of this report is a set of guiding principles, protocols and standards which enable the implementation of Internet of Things applications. It can but does not need to be an active participant of the overall IoT system. Frameworks can enhance IoT application development by; rapid implementation, interoperability, maintainability, security and technology flexibility.

To achieve rapid implementation many of the "boiler plate" tasks can be computer aided or removed completely. For example, a web service will not be concerned with handling the routing of messages, this is taken care of by underlying protocol or framework, such as IP the stack. The Internet protocols are based on layered architecture. Interoperability must extend across each layer. For example data packets can be forwarded between nodes on the same network so long as they all support the same network protocol, such as IP. In the context of service oriented architecture, interoperability requires that applications are able to discover service providers and consume the services in order to perform the system

functions. This requires shared understanding of interface and shared understanding of data semantics.

A key criteria for opening closed systems to the Internet is security. Authentication, authorization, confidentiality, integrity and privacy are of most interest to IoT applications. To be able to upload data to an untrusted cloud service provider and then share the data with different service consumers requires complex security functions.

### A. Internet of Things Approaches

The approach used by a framework will determine its suitability for different application spaces and impact information latency, data collation, feature interdependency, module design and network topology. This section will provide an overview of the high level approaches employed in current frameworks.

Many frameworks take a data centric or data driven approach. Utilizing a global cloud, they focus on enabling collation, visualization and analytics on data. This architecture is well suited for applications such as asset tracking, logistics and predictive maintenance [5]. In some cases the framework will allow creation of local hosted instances [6] but do not detail a method of interconnecting multiple cloud instances within the framework. This approach is suitable for providing data as a service but will generally leave the implementation specifics of the end-points to application developers. The framework simplifies the operation of end-points to only feeding data back to a central repository which will then implement complex security authorizations and usage tracking. The increasing computing power at the edge of networks is not leveraged in such frameworks and can introduce inefficiencies in bandwidth and latency. Figure 1 illustrates the concept of a global cloud through which IoT applications connect and communicate.

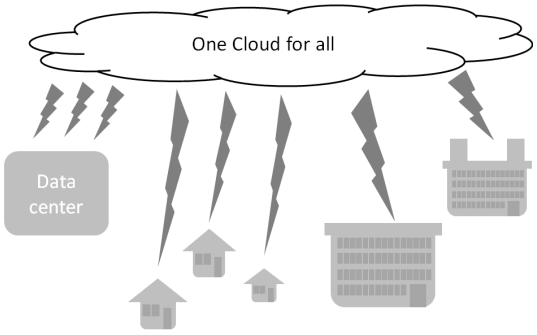


Fig. 1. Global cloud concept

A smart objects approach makes the endpoints active participants within the framework. The end points are included as key aspects of the framework which means the focus of the framework is on interconnecting the end-points. This approach is well suited for distributed automation tasks which require a high level of device independence, such as home and building automation and manufacturing.

While the devices can be very small they are often directly addressable from the Internet, see [7]. Because of this focus on automated end points and functional behavior, many of these frameworks do not go into the specifics of cloud integration and so do not provide good support for data collection. The data is produced by end points and consumed by end-points, which within this context will usually have some predefined understanding of the end-point pairing. The implementation of the data in the cloud is left to each IoT designer with minimal support from the framework.

In both of the approaches mentioned, many features such as end-to-end security and layered interoperability suffer due to ad-hoc development either at the end-points or within the cloud. A third approach becoming more prevalent, is taking into account the need to satisfy real-time automation requirements while not hindering the value of semantic big data and data analytics. Figure 2 illustrates the local cloud concept with independent local operation and shared global functionality.

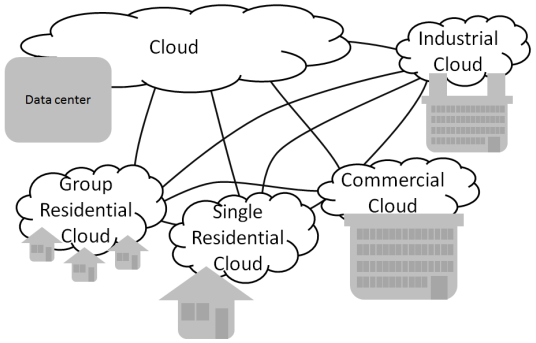


Fig. 2. Local cloud concept

Frameworks based on an approach which enables standards based development of end points and also of data warehousing, will enable secure interoperation of modularized and distributed applications. Key to the framework is leveraging intelligence at the end points and utilizing global cloud and local cloud concepts [8]. Enabling specialization of clouds for the requirements of the user or end points, whilst enabling data and event movement between clouds. This approach scales peer-to-peer networks into integrated cloud solutions.

## III. Frameworks

### A. IoT Frameworks for home automation

Home automation has been a key area for development of IoT-based applications. With the reduction in costs of manufacturing of IoT enabled devices, there are three major frameworks trying to gain support from device manufacturers and application developers. IoTivity - backed by Intel and Samsung, AllJoyn - backed by Qualcomm, LG and Sony, and Thread - back by ARM and Google.

*1) IPSO Alliance:* The IPSO Alliance framework is interested in standardizing semantic description in the IoT and supports a resource based object model [9] for other frameworks to build on. Using SenML and either XML or JSON encoding the IPSO alliance fills only part of the framework stack. But other frameworks have chosen to work with the

IPSO framework and indeed IoTivity and OMA-LWM2M do build on IPSO Alliance work. The IPSO specification builds on top of IETF CoRE standards such as 6LoWPAN, CoAP and SenML [9]. .

*2) IoTivity:* The IoTivity framework is developed by the Open Interconnect Consortium initially targeting IoT in smart homes and looking to further expand to other IoT silos. It is based on CoAP and its key building blocks are the Connectivity Abstraction (CA) layer and a Resource Introspection (RI) layer. The framework is being extended to HTTP and other communication protocols are supported through protocol plug-ins [10].

The IoTivity stack is shown in Figure 3. The thin block stack supports resource constrained devices. IoTivity makes use of D/TLS for security. Another interesting feature of IoTivity is Soft Sensor concept [10], which supports processing raw sensor data at intermediate or edge nodes.

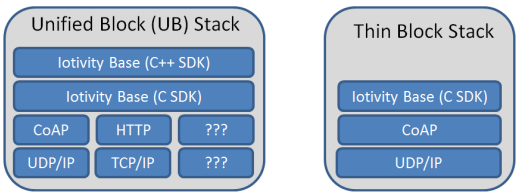


Fig. 3. IoTivity framework stack

As described in the Things Manager [10], IoTivity adopts a similar approach to the IPSO Alliance by using a object and resource based model. LWM2M and IoTivity find convergence here with the IPSO Alliance model and possible opportunity for interoperability.

*3) AllJoyn:* The AllJoyn framework, developed by the AllSeen Alliance, is designed for enabling interoperability for home automation [11] and industrial lighting [11] applications. The AllJoyn core operates as a software bus [12] between devices. Devices must implement a bus attachment responsible for message marshaling and serialization. Constrained devices use a thin library [12], and do not have a bus attachment, so must connect to an AllJoyn router. Thin devices work at the messaging level, while full devices communicate by remote procedure calls. Running such frameworks introduces overheads which limit real-time performance and participation of resource constrained and low power devices.

AllJoyn core library provides authentication and encryption for end-to-end security [12]. Authentication is provided by means of Simple Authentication and Security Layer (SASL) security framework defined by the D-Bus specification [13]. It supports both point-to-point ”session” keys and point-to-multipoint ”group” keys. Thin Client devices [12] don’t support this security. It is transport agnostic [12] and is currently running on WiFi, Ethernet, serial, and Power Line.

*4) Thread:* The Thread groups framework defines a protocol stack based on Nest’s early implementation of the smart thermostat; it uses the IETF IP stack, UDP and builds up additional security and commissioning functionality [14]. The Thread protocol can address devices directly and is able to perform peer-to-peer communications in a mesh network. It is seen as an evolution from the traditional ZigBee stack to an IP based stack [14].

Being built atop the standard IEEE 802.15.4 radio allows them to make use of already mass produced ZigBee chips [14]. There is limited information available as Thread is only available to member companies and the Thread group will be providing a certification process.

### *B. The Arrowhead Framework*

A light framework aimed at enabling the Industrial Internet of Things and to improve interoperability between applications. It is based on Service Oriented Architecture (SOA). It is made up of three core systems; Service Register, Orchestration, and Authorization. Supporting SOA-based design principles including; standardized service contracts, service loose coupling, service abstraction and service autonomy. In order to be Arrowhead compliant, applications must register services they produce with an Arrowhead service registry. Then use the Arrowhead authorization service to accept or reject consuming applications. The Orchestration system allows dynamic reconfiguration of the service consumer and provider end-points.

The framework can operate with a hierarchical set of core systems allowing a single machine to operate its own Arrowhead cloud, allowing local authorization and orchestration rules. Inter-cloud service discovery is supported meaning that local clouds can consumer outside services or provide data as a service to outside consumers.

A sample Arrowhead network is shown in Figure 4. In this network two embedded devices provide one service each to a third device which then provides a single service to an operator’s computer. All security, discovery and connection rules are supported by the Arrowhead framework.

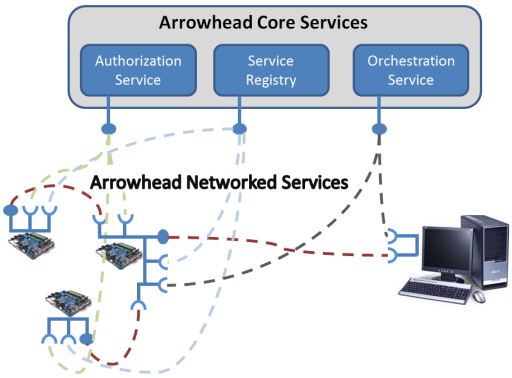


Fig. 4. Arrowhead Core Services and network example

The Arrowhead framework prescribes a method of documentation in order to enable native interoperability between services, and uses service transparency to enable interoperability between services which have used different semantics and communications protocols. Applications are responsible for

implementation details such as messaging and business logic. Once the base services are developed or 3rd party services are chosen, Arrowhead provides a framework for governing and interconnecting the services.

### C. OMA - LWM2M

The Open Mobile Alliance - Light Weight Machine to Machine (LWM2M) framework has been developed for the purpose of monitoring, provisioning and managing connections of networked devices [15]. LWM2M is based on CoAP and defines a layer above CoAP. The initial release addressed the interfaces between two devices in the areas; Bootstrap, client registration, device management and service enablement, and information reporting [15]. The framework uses the standard CoAP to UDP binding and also defines a new binding for CoAP to SMS. The LWM2M stack is shown in Figure 5, other than the addition of SMS, LWM2M utilizes IETF standards.

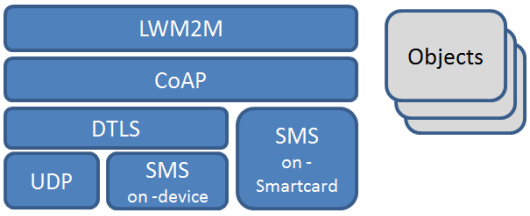


Fig. 5. OMA - LWM2M framework stack

The framework defines a resource and object model whereby the LWM2M client holds objects which each contain some resources. These objects can be instantiated by either a remote LWM2M server or by the client itself. Once instantiated the objects can be operated upon via the interfaces mentioned above.

As shown in the diagram of the LWM2M stack the security layer is handled with DTLS [15]. It is , but not recommended, possible to send the data without security. Access control is specified at the object layer with the use of an Access Control Object Instance [15]. They define what operations are allowed on a per Object Instance case. So within a given LWM2M client a single object instance will hold resources with different access control rights depending on the LWM2M server performing the access.

The framework unifies the CoAP server and client model into LWM2M server and LWM2M client which allows bi-directionally RESTful communications [15]. For example, a LWM2M server is able to POST to a LWM2M client and vice versa. This means the application development architecture has a more flexible concept of client/server.

### D. Other frameworks

Other frameworks include many groups either targeting niche or specific aspects of the IoT, or are looking to provide oversight and guidance to the development of the IoT technologies. The IPSO Alliance targets semantics and standardizing smart object interaction by defining the information model and RESTfull message exchange. The Industrial Internet Consortium (IIC) is looking to provide leadership and guidance to IoT framework organizations to bring direction towards convergence in IoT requirements, best practices and overall architecture. The Smart Energy Profile 2.0 (SEP2.0), originally from the ZigBee Alliance, is a well-defined specification for smart meters and energy management; it is now standard of the IEEE-SA. SEP2.0 is based on IP and uses a RESTfull interface. It fully defines governance aspects such as, encryption, authentication and key exchange [16]. The AXCIOMA [17] framework developed by Remedy IT uses Object Management Group (OMG) technologies and standards such as CORBA and DDS. AXCIOMA has defined the need for supporting Request-Response and Publish-Subscribe patterns and supports both these interfaces. The framework is targeting real-time and embedded networks.

## IV. Platforms

### A. Cloud-based IoT Platforms

Centralized platforms offer a simple method of integrating sensors into IoT applications. By employing a global cloud approach, Cumulocity, ThingWorx and Xively provide an integration platform for organizations to build IoT applications on. They recognize that many commercial organizations will be interested in gaining value from the data provided by embedded sensors.

*1) Cumulocity:* In the Cumulocity platform, sensor nodes are clients which connect to the cloud through a RESfull HTTPS API. Sensor nodes are modeled as objects with properties and methods for access and manipulation. Commands are pulled by devices from the Cumulocity server [18]. Depending on the pull frequency there will be a delay from the issued command until the device receives it. Constrained networks not operating on HTTPS use an ”agent” [18] to connect to the Cumulocity server. A server side ”event” language syntactically resembling SQL scripting [18] loads triggers to be performed in reaction to events. Cumulocity only supports RESTful HTTP/S.

*2) ThingWorx:* The ThingWorx platform targets application integration through model driven development. It composes services, applications and sensors as data sources and interconnects these through a virtual bus. The framework is transport agnostic and has been ported to run with CoAP, MQTT, REST/HTTP and Web Sockets [6]. Treating other clouds as data sources ThingWorx integrates with other cloud providers such as Xively and web services such as Twitter and weather services. Communication between devices, services and applications must be routed through the ThingWorx bus, thereby not enabling peer to peer communication. Using a Mashup builder, organizations are able to quickly connect data sources to dashboards, for tracking and monitoring assets and gathering data from many data sources to perform data analytics in real-time [19].

*3) Xively:* Third in this group is the Xively platform, formerly known as Pachube. This platform similar to the previous two provides a central message bus which routes messages

between devices of different protocols. The components which make up the Xively architecture can be seen in Figure 6.

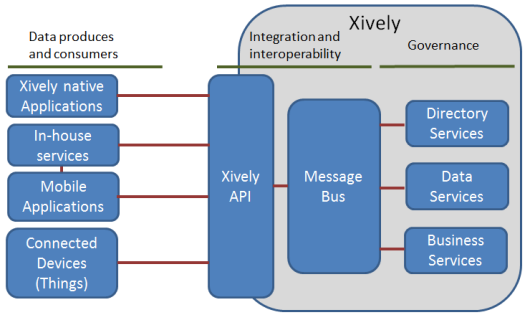


Fig. 6. Xively platform architecture

The message bus combined with the Xively API for MQTT, HTTP, and Web Sockets to provide an interoperability layer. It is a data driven platform with ability to give fine grain access to data streams and data feeds. Based on the client server model, they have a centralized method of device configuration where each device has a virtual presence and when a device comes online it uses its serial number and some form of mutual authentication to receive its configuration parameters setup on the Xively server. The framework has additional services which allow for Business Services, Systems Integration and Business Opportunities for companies and assist with governance of the network.

### B. Device to device platforms

In this section two platforms are described: Echelon's IzoT platform and the ThingSquare platform.

*1) IzoT:* The IzoT platform is made up of a communication stack intended for peer-to-peer communications [20] consisting of several proprietary high level protocol services which run on top of UDP [20]. Supporting priority messaging and end-to-end acknowledgements on unicast and multicast messages, the communication stack can support multiple simultaneous messaging on unconstrained devices. It has built in discovery and interface publishing, and can run on many networks including 6LoWPAN, free topology twisted pair, WiFi and potentially any medium which can support UDP sockets [20].

IzoT supports symmetric and asymmetric key encryption and authentication. Using a proprietary communication stack limits the ability for IzoT to be adopted widely for general IoT applications.

*2) ThingSquare:* The ThingSquare platform is founded from the development of the Contiki OS and is strictly based on IETF communications stacks. Their offering includes cloud based device governance and boot-strapping, but is limited in terms of cloud based application integration and data analytics. The focus of the framework is enabling automation, control and monitoring of smart objects through the Internet. Contiki OS boasts the smallest IP stack according to ThingSquare [21]. There operating system has been ported to run on many of the current IoT micro-controllers [21].

The ThingSquare framework uses cloud based services for device management, authentication and authorization [21] of new devices to the network. The ThingSquare framework will only allow authorized users and devices to register and control other ThingSquare devices. A simplified network is shown in Figure 7, along with the network stack supported.

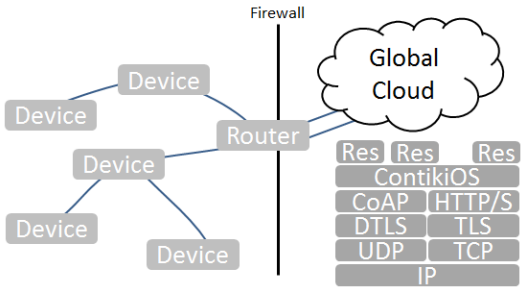


Fig. 7. ThingSquare network and stack

### C. Platforms for Cloud to Gateway Integration

Intel, Microsoft and IBM have all formed strong offerings in the cloud to device IoT market. This group of platforms offer platform support in the cloud and also offer solutions in intelligent gateways, however do not provide much application development support for end-points.

*1) Intel:* Intel have partnered with Wind River and McAfee to produce an IoT framework which includes hardware for things, intelligent gateways, cloud and Platform as a Service [22] [23]. Intel's hardware technologies, Wind River's Operating Systems (OS) and McAfee's security products can be utilized in different layers of the IoT from embedded to the cloud.

VxWorks [24] can scale down to a 20kb foot print for use in constrained embedded systems, while Wind River's gateway OS, based on Linux, can support many application environments including Lua, Java, and OSGi [23]. Whitelisting binaries means that binaries without the correct signature cannot be executed on a device [23]. Role-based access control is used to provide a learning mode to generate security policy rules [25].

*2) Microsoft:* Microsoft supports the IoT at three layers. The Microsoft Azure Cloud provides an excellent platform for developing and integrating distributed applications using its proprietary Enterprise Service Bus. Device connectivity and governance is supported by Microsoft Azure Intelligent Systems Service (ISS) [26]. Microsoft StreamInsight is a platform for in-memory data analytics and processing [27]. It allows IoT applications to process data without the latency involved with traditional databases. It can be run as a local Web service or in the cloud as a hosted service in Azure. The device layer is supported by Microsoft Windows embedded [28] and the .net microframework (.netmf) [29].

Microsoft Research have developed the HomeOS platform [30] as a multi-protocol home automation server. It is a virtual OS running on a COTS computer and providing

inter-connection between multi-vendor COTS devices [31]. Although not yet, commercially available, it occupies the same space as AllJoyn, IoTivity and Thread [31].

*3) IBM:* There are many offerings from IBM and by combining them it is possible to run an end-to-end industrial or consumer IoT systems with MQTT-based communications and enterprise middleware. The key offerings are BlueMix application server, WebSphere enterprise integration middleware, MobileFirst application development platform, Informix database, and the MessageSight MQTT broker. These different IBM products are shown in Figure 8.

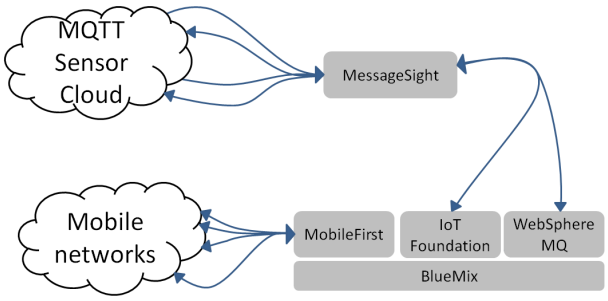


Fig. 8. IBM product framework

IBM BlueMix is based on Cloud Foundry [32] application server, offering additional management capabilities. The IBM IoT Foundation is hosted on Bluemix and provides governance of IoT devices. IBM WebSphere MQ offers proven and robust enterprise application integration and supports MQTT networks by integrating WebSphere with the IBM MessageSight Appliance [33]. Previously known as IBM WorkLight Foundation [34] MobileFirst is a platform enabling machine to machine connections through mobile devices. It is a platform for web application and native mobile application development and provides hosting for those applications.

IBM MessageSight is a communications appliance which can handle high volumes of MQTT communications. Built with performance in mind, the IBM MessageSight appliance can process over 350K MQTT v3.1 messages per second and can publish over 15M messages per second to consumer applications [35]. The IBM MessageSight appliance can be run in a Virtual environment [36].

## V. Discussion

The frameworks and platforms have been introduced and a high level over view of their key features has been described. Now we will discuss the frameworks within the context of the evaluation criteria. The evaluation criteria are; the framework approach, industry support, underlying protocols, security, applicability to constrained devices and support for rapid application development.

Architectural approaches were introduced in section II-A and the frameworks can be categorized as in Table I. This categorization is indicative of the target application space for the frameworks.

TABLE I
THE THREE APPROACHES DISCUSSED AGAINST THE FRAMEWORKS AND PLATFORMS STUDIED

| Approach | Frameworks, Platforms and Protocols |
|---|---|
| Global Cloud | Cumulocity, Xively, ThingWorx, IBM, Microsoft, Intel, LWM2M |
| Peer to Peer | IPSO, Thread, Thingsquare, IzoT, SEP 2.0, AllJoyn, IoTivity |
| Local Cloud | Arrowhead |

Cumulocity, Xively, ThingWorx, IBM, Microsoft and Intel are serving the global cloud approach to running IoT Applications. They provide the hosting platforms and application API for interacting between devices from applications running in the cloud. This is a well-known model used for business systems which prefer a centralized application. LWM2M joins this group as a device governance framework with a centralized approach. IPSO, Thread, ThingSquare, IzoT, SEP 2.0, AllJoyn and IoTivity have approached IoT application development from a device level and support a high level of peer-to-peer operation. This approached serves their customers well in home automation and device management. Within the local cloud category, Arrowhead sits alone. The unique approach of this framework is the support for integration of applications between secure localized clouds. The Quality of Service, Security and scalability of industrial automation has necessitated this approach.

The qualities of peer-to-peer communication, mesh network support, 6LoWPAN and low power make the ThingSquare attractive for running edge of network in conjunction with cloud platforms such as Xively, Cumulocity or ThingWorx.

Table II shows a few of the larger IoT framework backers against the frameworks in this paper. Platforms are not shown here as they are usually supported by the platform provider only.

TABLE II
FRAMEWORKS ORGANIZATIONS MEMBERS AND SUPPORTERS.

| Framework | Arrowhead | AllJoyn | Thread | IoTivity | LWM2M |
|---|---|---|---|---|---|
| Google | | | ✓ | | |
| Microsoft | | ✓ | | | ✓ |
| IBM | | | | | |
| Intel | | | | ✓ | ✓ |
| Cisco | | ✓ | | ✓ | |
| GE | | | | ✓ | |
| AT&T | | ✓ | | | ✓ |
| Samsung | | | ✓ | ✓ | ✓ |
| ARM | | | ✓ | | ✓ |
| Motorola | | | | | ✓ |
| Qualcomm | | ✓ | | | ✓ |
| LG | | ✓ | | | |
| Schneider Elec. | ✓ | | | | ✓ |
| AVL | ✓ | | | | |
| STM | ✓ | | | | |
| Members | 81 | 101 | 81 | 55 | 96 |

LWM2M has support from many larger organizations as it falls under the OMA umbrella of specifications. Frameworks with support from large organizations mitigates sudden end of support for a chosen framework. It can be seen that

Microsoft and Intel, while providing an IoT platform, are also members of the IoT framework development. The number of member organizations within each framework also indicates the confidence industry has in each framework.

The centralized frameworks mentioned earlier offer message protocol flexibility and will usually support MQTT, REST and sometimes CoAP and XMPP. Table III shows the frameworks against common protocols. Under the 'other column' are either proprietary protocols or less common protocols such as DDS.

TABLE III
COMMUNICATIONS PROTOCOLS SUPPORTED BY THE STUDIED FRAMEWORKS AND PLATFORMS

| | MQTT | XMPP | CoAP | REST | Other |
|---|---|---|---|---|---|
| IPSO Alliance | | | ✓ | ✓ | |
| LWM2M | | | ✓ | | |
| Arrowhead | ✓ | ✓ | ✓ | ✓ | ✓ |
| SEP 2.0 | | | | | ✓ |
| AXCIOMA | ✓ | ✓ | ✓ | ✓ | ✓ |
| Thread Group | | | | | ✓ |
| AllJoyn | | | | | ✓ |
| ThingSquare | | | ✓ | ✓ | |
| IzoT | | | | ✓ | ✓ |
| ThingWorx | ✓ | ✓ | ✓ | ✓ | ✓ |
| Xively | ✓ | ✓ | ✓ | ✓ | ✓ |
| Cumulocity | | | | ✓ | |
| IBM | ✓ | | | | ✓ |
| Microsoft | | | | | ✓ |

To mention open source frameworks and the protocols they rely on is important to developers looking for flexibility in choosing libraries, vendor platforms and interoperability. Many of the frameworks are proprietary, but most support at least one open source messaging protocol. IPSO Alliance, OMA-LWM2M, AllJoyn, IoTivity and Arrowhead are open-source and based on open-source technologies. IBM and Microsoft are proprietary and the Thread specification is only available for member companies. The platforms sch as ThingWorx, ThingSquare, Xively and Cumulocity are proprietary and so moving an application from one to another would be a costly exercise.

In table IV the frameworks and protocols in this survey are categorized by layer. Most of the data centric frameworks sit in the application layer, while many of the control centric frameworks are in the messaging layer. In this case messaging layer is referring to a layer above transport while still not offering a rich application layer support.

TABLE IV
FRAMEWORKS AND PLATFORMS CATEGORIZED BY MOST REPRESENTATIVE LAYER

| Layer | Frameworks or Protocols |
|---|---|
| Application | Arrowhead, IPSO Alliance, Xively, Cumulocity, ThingWorx, Smart Energy Profile 2.0, Microsoft, IBM, ThingSquare, Industrial Internet Consortium, AllJoyn, IoTivity, IzoT |
| Messaging | Web-Sockets, XMPP, MQTT, CoAP, HART, Thread, AllJoyn |
| Transport | TCP, UDP, WirelessHART, SMS |
| Network | IP, ZigBee |

Some of the frameworks such as, IoTivity and AllJoyn support a dual stack implementation, supporting a reduced functionality stack for constrained devices. However this is not always the case, such as Xively, Cumulocity and ThingWorx who do not support constrained devices and rely on intermediary agents or gateways to integrate resource constrained devices. Security aspects are important for hardware requirements. Crypto hardware support is required by IzoT, AllJoyn and IoTivity. Echelon's IzoT platform offers hardware components as part of their framework and also provides adaption layers for non-IzoT devices. IBM supports its MQTT based framework with a dedicated server appliance which runs the MQTT broker.

Rapid application development, (re-)configurability, scalability and deployment considerations are important characteristics. It is difficult to make evaluation on such aspects, but it is worth mentioning frameworks with comparative strengths. IBM and Microsoft's strong background in enterprise service bus means they have a good advantage for scaling up as business needs grow. ThingWorx, Cumulocity and Xively demonstrate strength in rapid application development and focus on value added work. Thread, Iotivity and AllJoyn tend focus on customers using commercial off the shelf devices and therefore simplify the deployment. Arrowhead's strength is in its re-configurability, through the use of dynamic orchestration of services and systems.

Further on governance and management of the devices, services and interfaces will assist with rapid application development and maintenance. The cloud platforms such as Cumulocity, ThingWorx and Xively offer great application governance and some device management of an active device. However AllJoyn, IzoT platform and ThingSquare offer good device management and lesser support for application governance. IBM and Microsoft both have mature cloud application governance and management. Microsoft has good device management through its embedded OS family and also it's embedded .net run time. Arrowhead provides application configuration and authorization governance through its core services. Applications can discover services, download configurations and authorize access. The primary purpose of LWM2M is the governance and management of devices, at a scale for cellular operators. This tends to suggest it will have good performance for large scale device networks.

## VI. CONCLUSION

As the market for IoT applications grows it industry has worked with academia to create a standardized set of communications protocols. Next frameworks and platforms for the IoT are being developed by industrial consortia. This is in order to lay down a foundation at the application layer which will enable deployments of large scale, either in instance size or in instance number, IoT applications.

This survey has presented a number of commercially available frameworks and platforms for developing industrial and consumer based IoT applications. The studied frameworks have each approached IoT from the perspectives and priorities

of their customer needs. The priority was either on; centralizing distributed data sources for cloud-based applications, referred to as a global cloud approach; or supporting integration of devices for home(building)-automation, referred to as the peer-to-peer approach; or integrating devices and clouds together for factory and industrial automation systems, referred to as the local cloud approach.

A comparative analysis of the frameworks was conducted based on industry support, use of standards based protocols, interoperability, security, hardware requirements, governance and support for rapid application development. Based on this analysis academia and industry can identify frameworks most suitable for their future projects and identify gaps in the current frameworks.

Finally, for platforms and frameworks to succeed they must recognize and facilitate:

1) Enable devices, applications and systems to securely expose API's for 3rd party systems and to facilitate API management.
2) Enable systems to have protocol interoperability with other 3rd party API's and ensure they are extendable for new protocols.
3) Enabling constrained devices to participate into application networks. That is size, bandwidth, power supply(battery) and processing power constraints.
4) Governance - Enabling management and governance of heterogeneous networks of devices and applications.

The value of the whole is greater than the sum of its parts.

## ACKNOWLEDGEMENTS

The Authors would like to thank AVL List GMBH for supporting this survey of commercial frameworks for the Internet of Things.

## REFERENCES

[1] D. Evans, "The internet of things how the next evolution of the internet is changing everything," White Paper, Cisco, April 2011.

[2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *COMPUTER NETWORKS*, vol. 54, no. 15, pp. 2787–2805, OCT 28 2010.

[3] D. Singh, G. Tripathi, and A. Jara, "A survey of internet-of-things: Future vision, architecture, challenges and services," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, March 2014, pp. 287–292.

[4] C. Perera, C. Liu, S. Jayawardena, and M. Chen, "A survey on internet of things from industrial market perspective," *Access, IEEE*, vol. 2, pp. 1660–1679, 2014.

[5] Xively solutions by application. Xively. [Online]. Available: https://xively.com/solution/by_opportunity/

[6] Thingworx platform - dzone. DZone. [Online]. Available: http://dzone.com/zb/products/thingworx-platform

[7] Technical information. IPSO Alliance. [Online]. Available: http://www.ipso-alliance.org/technical-information

[8] P. Varga, F. Blomstedt, L. L. Ferreira, J. Eliasson, M. Johansson, and J. Delsing, "Making system of systems interoperable - the core components of the arrowhead technology framework (accepted for publication)," *IEEE Internet of Things Journal*, August 2015.

[9] The ipso application framework draft-ipso-app-framework-04. IPSO Alliance. [Online]. Available: http://www.ipso-alliance.org/wp-content/media/draft-ipso-app-framework-04.pdf

[10] Home iotivity. IoTivity. [Online]. Available: https://www.iotivity.org/

[11] Allseen alliance wiki. AllSeen Alliance. [Online]. Available: https://wiki.allseenalliance.org/

[12] Learn - allseen alliance. AllSeen Alliance. [Online]. Available: https://allseenalliance.org/developers/learn

[13] H. Pennington, A. Carlsson, A. Larsson, S. Herzberg, S. McVittie, and D. Zeuthen. D-bus specification. [Online]. Available: http://dbus.freedesktop.org/doc/dbus-specification.html#auth-protocol

[14] Thread group - home. Thread Group. [Online]. Available: http://www.threadgroup.org

[15] Machine to machine (m2m) solution. Open Mobile Alliance. [Online]. Available: http://openmobilealliance.org/about-oma/work-program/m2m-enablers/

[16] Smart energy profile 2 application protocol standard. ZigBee Alliance. [Online]. Available: http://splintered.net/z/Zigbee-smart-energy-profile-2.pdf

[17] An extendable component-based interoperable open model-driven architecture. Remedy IT. [Online]. Available: http://www.remedy.nl/en/axcioma

[18] Interfacing devices. Cumulocity. [Online]. Available: http://www.cumulocity.com

[19] Joy mining connected products success story. ThingWorx. [Online]. Available: http://www.thingworx.com/learning_content/connected-products-success-story-joy-mining-2/

[20] B. Dolin. Requirements for the industrial internet of things. Echelon. [Online]. Available: http://info.echelon.com/IIoT-Requirements-Whitepaper.html

[21] Thingsquare - connecting the internet of things. Thingsquare. [Online]. Available: http://www.thingsquare.com/

[22] Transform business with intelligent gateway solutions for iot. Intel. [Online]. Available: http://www.intel.com/content/www/us/en/internet-of-things/gateway-solutions.html

[23] Intel gateway solutions for the internet of things. Intel. [Online]. Available: http://www.mcafee.com/us/resources/solution-briefs/sb-intel-gateway-iot.pdf

[24] Vxworks. Wind River. [Online]. Available: http://windriver.com/products/platforms/network_equipment/#content_3

[25] Intel iot gateway - security profiles. Intel. [Online]. Available: http://www.intel.com/content/www/us/en/embedded/design-tools/evaluation-platforms/gateway-solutions/iot-security-profiles-white-paper.html

[26] Microsoft azure intelligent systems service. Microsoft. [Online]. Available: https://connect.microsoft.com/ISS

[27] C. M. Torsten Grabs. (2012, March) Microsoft streaminsight - building the internet of things. [Online]. Available: https://msdn.microsoft.com/en-us/magazine/hh852591.aspx

[28] Intelligent systems: A new level of business intelligence. Microsoft. [Online]. Available: http://www.microsoft.com/windowsembedded/en-us/intelligent-systems.aspx

[29] .net micro framework. Microsoft. [Online]. Available: http://www.netmf.com/

[30] C. Dixon, R. Mahajan, S. Agarwal, A. J. Brush, B. Lee, S. Saroiu, and P. Bahl, "An operating system for the home," in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 25–25. [Online]. Available: http://dl.acm.org/citation.cfm?id=2228298.2228332

[31] P. Oliphant. (2014, May) Alljoyn vs homeos for the connected home. [Online]. Available: http://www.connectedhomeworld.com/content/alljoyn-vs-homeos-connected-home

[32] What is ibm bluemix. IBM Corp. [Online]. Available: http://www.ibm.com/developerworks/cloud/library/cl-bluemixfoundry

[33] V. Lampkin. (2012, March) What is mqtt and how does it work with websphere mq? [Online]. Available: https://www.ibm.com/developerworks/community/blogs/aimsupport/entry/what_is_mqtt_and_how_does_it_work_with_websphere_mq?lang=en

[34] A. Trice. (2014, November) So, what is ibm mobilefirst? [Online]. Available: http://www.tricedesigns.com/2014/11/19/so-what-is-ibm-mobilefirst/

[35] A. S. Arnaud Mehieu. (2014, November) Ibm messagesight. [Online]. Available: https://www-950.ibm.com/events/wwe/grp/grp004.nsf/vLookupPDFs/IBM%20MessageSight/$file/IBM%20MessageSight.pdf

[36] Ibm messagesight - ibm messaging. IBM Corp. [Online]. Available: https://developer.ibm.com/messaging/messagesight/