Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2012 Proceedings | Proceedings

A Cloud-based Messaging Service for Cross-Enterprise Data Exchange with Smart Objects

Jens Strueker
Department of Telefmatics, Institute of Computer Science and Social Studies, Freiburg, Germany., jens.strueker@gmail.com

Harald Weppner
SAP USA, Palo Alto, CA, United States., harald.weppner@sap.com

A Cloud-based Messaging Service for Cross-Enterprise Data Exchange with Smart Objects

Jens Strüker
Albert-Ludwigs Universität Freiburg
strueker@iig.uni-freiburg.de

Harald Weppner
SAP Labs Palo Alto
harald.weppner@sap.com

ABSTRACT

In this paper, we explore common communication needs for the rapidly increasing number of Internet-connected devices, which are appearing in a growing number of domains. We argue that with the rise of these smart objects business cooperation will increase. So-called smart meters then serve as example that a direct and flexible information exchange across enterprise boundaries, markets and even industries is needed. Based on experiences from integration projects and qualitative interviews with experts we deduce key requirements for an abstract communication system. We then map established communication paradigms to these requirements and finally introduce a cloud-based communication system for smart objects we call Virtual Object Warehousing Service. We explore its key characteristics and conclude by providing an outlook how such a general-purpose cloud-based messaging service could satisfy the communication needs of smart objects.

Keywords

Internet of Things, Ubiquitous Computing, Cloud Computing, Smart Metering.

INTRODUCTION

The so-called Internet of Things takes concrete shape (e.g. Mattern and Flörkemeier 2010). More and more companies are using devices connected to the Internet via IP and, thus, embedding an IP communication stack is becoming recognized as a pragmatic and long-term solution in favor of proprietary closed approaches (Dunkels and Vasseur 2008) (Hui et al 2009) (Zigbee Alliance 2009). If the way the Internet has revolutionized the cooperation of enterprises is any indication, we can only begin to fathom how linking the physical with the virtual world, i.e. sharing identification and sensor information between market actors and interacting with these smart objects, will further intensify cooperation.

However, even with IP as a least common denominator for communication, exchanging data between smart objects and enterprises is challenging because it will only become valuable information once it is in fact accessible in a timely manner by the systems that can put the data into context with other data sources. By taking smart metering as an example, we show that changing out the meter-reader and connecting them to the Internet is not just a simple technology shift. This is a transformation of the way utilities conduct business in many ways. One of the resulting challenges is to deploy an appropriate communication infrastructure, i.e. integrate the smart objects into existing business processes and IT infrastructures under the condition of enabling an interaction between intelligent objects and business applications in a generic, flexible and scalable manner. This contribution addresses the communication needs of Internet-connected smart objects by proposing a *cloud-based messaging framework for smart objects*. We argue that Internet-scale, general-purpose data exchange services can facilitate, control and monitor the *borderless* delivery of messages among Internet-connected devices and enterprise applications.

Our paper is structured as follows. In the first part, we describe in what way the cross-enterprise data exchange between smart objects and applications is a decisive hurdle on the way to intensified business cooperation. We therefore first describe how smart objects exchange data today. Subsequently, we deduce from our analysis of the smart meter scenario four common principles for the data exchange with smart objects. As data exchange necessitates the integration of smart objects with enterprise information systems, we check the suitability of today's integration approaches for smart objects. We can show that the most common integration models are not appropriate to match crucial communication needs of smart objects. In part two, we propose a *cloud-based messaging framework for smart objects* to address the data exchange challenge. Based on experiences from integration projects and qualitative interviews with experts, we set up key requirements for an abstract communication system, map established communication paradigms to these requirements and finally introduce a *cloud-based messaging*

model for smart objects we call *Virtual Object Warehousing*. We explore its key characteristics and then illustrate, based on the smart metering scenario, some major advantages of a *Virtual Object Warehousing Service* in part three. We conclude by proposing an outlook how a general purpose Internet service could satisfy the communication needs of smart objects.

PART I: CROSS-ENTERPRISE DATA EXCHANGE AS A HURDLE

Status Quo of the Integration of Smart Objects with Enterprise Information Systems

We define a smart object as a physical object equipped with a special-purpose computer system. This smart device is able to sense information from or perform actions affecting its environment (the material/ physical world) and which is able to digitally communicate with other networked computer systems. We impose no requirements on the system architecture of the device nor do we assume anything about a device's physical dimensions or its possibly severe constraints on energy or communication bandwidth. Given this definition, a smart device can be very small or very large, geographically fixed or mobile, possess one or more communication channels, be always on or only occasionally connected and might perform either extremely simple or very complex computations. This definition of a smart device consequently includes embedded systems, sensor networks as well as RFID-systems.

Smart devices today play a key role in vehicles, elevators, medical equipment, building automation and many more application domains (e.g. Vasseur and Dunkels 2010). The integration of smart devices with enterprise information systems can be – as a matter of principle – divided into the three categories shown in Figure 1. Note that the depicted boundaries are to be viewed as logical separations of administrative domains and not to imply any particular networking topology.

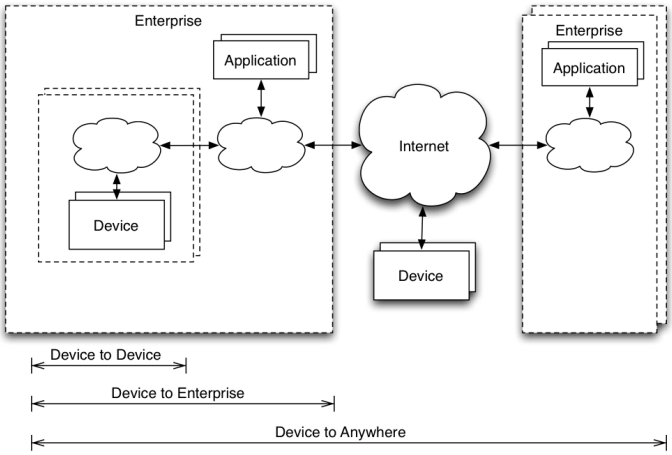

Figure 1: Device Integration Categories

The device-to-device category represents the family of scenarios where devices communicate among each other – usually when they are within close physical proximity. This form of integration is characterized by synchronous, very low-latency communication networks and the sender-receiver relationship is usually static with some select cases of one-to-many communication using broadcasting or multicasting. One interesting observation is that a system composed of connected devices can itself be viewed as a smart object when viewed at a higher level of abstraction. In the industrial automation domain, device-to-device communication has played a crucial role for decades: sensors and actuators interact with control units and thereby enable the coordination of production and manufacturing processes (Chen et al. 2010). These devices have a huge impact on the efficiency of businesses (Shanbagaraman 2010): reading and processing status information such as liquid level, pressure, temperature, flow rate, end positions, rotational speeds and control status at the field level enable plant operation round-the-clock in industries such as automotive, plastics, fabrication metal or food and beverages.

The device-to-enterprise category represents scenarios in which devices are integrated with applications at different levels of an enterprise such as shop floor control or Enterprise Resource Planning (ERP) systems. Integration tends to be predominantly asynchronous with some cases of synchronous communication, the latency is still relatively low and the sender-receiver relationships are either *one-to-one* or *one-to-many*. The relationships are relatively stable but often managed independently in middleware systems to allow for a reasonable degree of decoupling and flexibility.

The device-to-anywhere category represents integration scenarios where devices are either freely roaming (mobile) or are located permanently within the domain of another enterprise. In both cases, the device connects back to an enterprise using the Internet either via a public access point or via virtual private networks. For instance, enterprises connect costly and critical assets to information systems in order to perform remote monitoring and remote diagnostics. Allmendinger and Lombreglia (2005) have demonstrated the broad business impact of such devices and identified nine different industries

(venues) when investigating one single application of remote device management. Wireless systems communicate sensor readings to a storage unit within a container or a central aggregation point on the ship (Wright 2009). Similar systems have been installed in the air conditioning system of hotels (Gislason 2008). The energy industry can serve as a further example for illustrating the impact of extending the Internet to smart devices (Vasseur and Dunkels 2010).

Four Principles of Information Exchange with Smart Objects

The Internet has dramatically expanded the enterprises' capabilities to exchange data and the emerging Internet of Things promises a new level of cooperation between enterprises: Since the Internet is logically linked together by a globally unique address space based on the Internet Protocol (IP), IP-capable objects can communicate directly with each other. Accordingly, connecting smart objects to the Internet potentially means to share identification and sensor information and interacting with these smart objects without limits. We will now illustrate by the means of the smart metering phenomenon that this promise is tied to requirements concerning the information exchange between smart objects and business applications. We will generalize these requirements and define *four principles of the information exchange with smart objects*.

Smart meters represent one class of smart objects, which are replacing millions of mechanical meters worldwide. This is primarily driven by the attempt to lower operational costs and the need for more detailed electricity consumption information (e.g. EU Commission 2006). European Union's directives have mandate accurate metering, which affects over 20 million enterprises and over 210 million households. The deployment of millions of smart meters is catapulting the electricity supply system into the Information Age and promises a plethora of new business opportunities (Vasseur and Dunkels 2010). In conjunction with other smart objects such as communication-enabled domestic appliances, light switches and electric vehicles, the need for a smarter electricity grid is created, which will in turn enable new forms of energy management and support new services. Markets such as home and building automation, electricity generation and distribution, automotive as well as telecommunication and information technology will steadily converge at a mega-market that is said to be one of the biggest infrastructure projects to come. Regardless of whether this estimation is justifiable, the world's largest energy, hardware, software and telecommunication companies and countless startups set their hopes high on this market and are investing billions of dollars (GTM Research 2010).

If one replaces a mechanical meter with a smart meter, processes such as reading and transmitting consumption data and remotely turning on and off the supply of energy and customer switching are now digitized, i.e. two-way communication from and to a smart object must be possible. In deregulated energy markets such as in the European Union, there is still one electricity grid operator responsible for any given geographical area but customers can choose from several meter operators, a couple of electricity generators and many retailers. In Europe's biggest market Germany, for instance, consumers can choose among hundreds of energy retailers. As customers are permitted to switch energy retailers in deregulated markets they are expected to do so at about the same frequency with which they change cell phone providers. That leads to dynamic and volatile business relationships among the market actors and will require a high degree of **flexibility** how information is exchanged.

In a deregulated energy market, the electricity generator, the grid operator and the retailer all require energy usage information captured by a smart meter for billing, load balancing etc. Note that some of these market actors each serve unique geographies and endpoints creating a substantial combinatorial problem of who should access what information. Due to the sheer magnitude of the numbers, this information exchange task is challenging (Bieser 2009). Additionally, the information has to be exchanged **across enterprise boundaries** and – due to new market actors such as telecommunication or IT companies – markets and industries.

Besides the fact that there are many addressees and market actors, the demand on automated meter readings differs significantly among the entitled parties with regard to content and time (Strüker et al 2011). While an energy retailer in general is satisfied with retrieving usage information on a daily basis, a grid operator will require a more detailed view of the energy grid and seek 15 min or lower interval data. If one of the parties is then forced to go through the other only to retrieve information from a smart meter, this unequal and **mediated access** is likely to generate enormous frictions between the parties. Likewise, a smart meter requiring to know which retailer it has to send messages to considerably limits spontaneous connections and the setup of the communication in general. Flexibility of communication will also decrease, if application systems and smart meters require to both be active at the same time to communicate and if a smart meter does not continue to meter the energy consumption, irrespective of whether it has to send or receive any information.

Things get even more complicated when demand-side participation comes into play. Retailers, for instance, can offer the customer an attractive pricing tariff if the customer allows them to *actively* manage some of his/her home appliances: e.g. cooling down the refrigerator in anticipation of peak times during which it can then be switched off periodically or managing a thermostat through micro-adjustments (e.g. ecofactor.com). The necessary condition is **two-way communication**, i.e. interaction with smart meters and smart devices. Appliances such as refrigerators can be used as a buffer to stabilize the network to compensate for the increasing number of renewable and often fluctuating energy sources. In order to be effective, grid operators have to leave a considerable amount of refrigerators off. The drawback becomes apparent when giving them

permission to return by means of a multicast message: grid operators then immediately have to cope with this block of additional demands.

Based on the described smart meter scenario, we can now state our key principles for the information exchange with smart objects:

1. **Flexible:** communication shall not require coupling in space, time or synchronization.
2. **Borderless:** information shall be seamlessly exchanged across enterprise boundaries, markets and industries.
3. **Non-mediated:** systems shall be able to communicate with a smart object without requiring an application-level system as the mediator.
4. **Two-way:** systems shall be able to receive from and/or send information to smart objects.

Inappropriate Integration Models for Cross-Enterprise Data Exchange

The smart metering case exemplifies how the spread of millions of smart objects is going to alter the coordination challenge of exchanging data among enterprises (blinded). The question is how a *flexible*, *borderless*, *non-mediated* and *two-way* data exchange can be realized. Unfortunately, current available inter-organizational data-sharing infrastructures like EDI (Electronic Data Interchange) and the EPC Network for RFID are not designed for the *interaction* with smart objects: Exchanging business documentation in a machine-processable format (EDI) and object identification data (EPC network) fundamentally differ from gathering and distributing sensor and environmental data. Both data exchange infrastructures are incapable of enabling *active and direct remote management*: It is impossible to upload commands and messages directly to the smart objects as well as to download data. However, this kind of interaction between smart objects and various market actors is a key element in order to further drive specialization and outsourcing of business functions (Rode et al. 2011).

Since approved data exchange infrastructures are not appropriate we turn to the *integration* of smart objects with enterprise information systems. We therefore drill down into the most common integration models within the *device-to-anywhere* integration category illustrated in Figure 1. One standard model we have seen in use is depicted in **Figure 2**. It assumes that devices and enterprise systems are all located within a *single* administrative network domain and they are connected directly with each other or via one or more middleware solutions. Often, device vendors bundle entire solutions that include the device, the middleware and even the enterprise application.

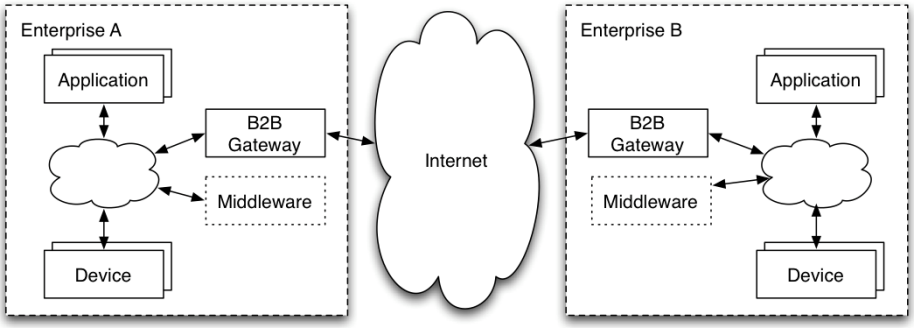

Figure 2: Classical integration model

The key point is that *other enterprises cannot access devices directly* and have to rely on other B2B infrastructures to indirectly receive from or send information to a device. This integration method consequently works reasonably well in all scenarios where devices are associated with exactly one enterprise and only affect functions that are *internal* to the enterprise. While many devices can be managed at the same time, installation, operation and maintenance are made difficult and obtaining interoperability between applications across enterprises needs time-consuming agreements on the business process level. Integrating via applications therefore decreases flexibility with regard to potential cooperation partners: integration costs cause switching costs, i.e. enterprises face a lock-in effect. A good example is sending electricity consumption information collected by a smart meter via a meter operator's business application (e.g. ERP) to different addressees: it is unrealistic that the business applications of the grid operator, other electricity retailers and generators and home automation providers will use the same applications with the same semantics – regardless of future agreements on industry-wide standards concerning data formats. Moreover, the different addressees are likely to need the consumption information/ readings differently with regard to frequency and accuracy.

With the Internet, a *point-to-point integration model* evolved, which is shown in Figure 3. It relaxes the constraint requiring devices to be located within a single administrative network domain. In this point-to-point approach, devices located outside of an enterprise domain (either freely roaming like vehicles or stationary like smart meters) or within another enterprise

domain (like printing machines at a customer site) send information via the Internet to a specified server located in another enterprise domain. The server handling the information flow from/to the device in turn acts as an application-level gateway to enable the integration with other internal enterprise applications.

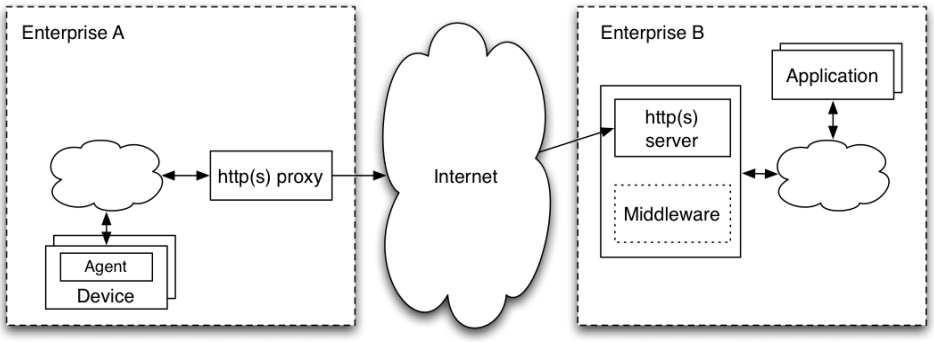

Figure 3: Point-to-point integration model

Figure 3 shows the exemplary use of http(s) as it is a common protocol choice because of its capability to be proxied and thus bypass firewall restrictions. Allemdinger and Lombreglia (2005) describe how Heidelberger Druckmaschinen, a leading German manufacturer of high-end printing machines, uses this model to provide various customer services by monitoring and accessing devices remotely. The machines communicate over the Internet and relay information about their status to Heidelberger's regional and global technical support specialists who are proactively notified in case of problems and are enabled to track down root causes of a problem to minimize any disruption. Utilities deploying smart meters first utilized point-to-point connections layered across several hierarchies. In Italy, the largest deployment yet, meters send to/receive from a localized system which then exchanges information via a centralized system (three tier hierarchy). The centralized management system uses GSM as a circuit-switch technology to manage the batch transfer of information from these localized systems, e.g. overnight because the communication costs are lower at that time (since people are asleep). Meanwhile, utilities prefer point-to-point communications with meters over public wireless networks, i.e. a direct point-to-point integration between smart meters and an enterprise system.

Because of its simplicity, the point-to-point integration model has been utilized in a number of established commercial deployments on a small and medium-size scale in thousands of devices, but it has the severe drawback that any given device can practically only communicate with a single designated enterprise. This is in direct conflict with the trend towards specialization and outsourcing more and more business functions, which all potentially require access to and control of a device. If one simulates a one-to-many integration in a point-to-point integration architecture, a device must make several connections and thus repeatedly tap into its potentially constrained resources while sending redundant information. Moreover, the device must now manage its connection endpoints, which leads to administration problems and a lack of flexibility as a result. Finally, it is impossible for a device to authenticate and authorize a potentially limitless number of endpoints, which – due to security concerns – often leads to the situation that a device not even reachable from the open Internet.

Both described integration models in the device-to-anywhere integration category most often utilize an Internet Protocol based network for the cross-enterprise communication (blinded). Faced with the choice between enterprises being unable to access devices directly (cp. classical integration model depicted in **Figure 2**) or a device that can only communicate with a single enterprise (cp. point-to-point integration model depicted in Figure 3), we state that neither of them are appropriate to match the afore-deduced communication principles for smart objects.

PART II: A CLOUD-BASED MESSAGING FRAMEWORK

We started this contribution by illustrating how smart objects are about to considerably intensify business cooperations. The foundation for this to become reality is the ability to organize the information exchange between smart objects and business applications in an un-constrained yet controlled manner. We identified four information exchange principles: borderless, non-mediated, two-way and flexible. As we showed, neither available data exchange infrastructures nor today's integration models match these principles. In the following, we propose a communication model that comply with our four principles for cross-enterprise information exchange and then discuss its key characteristics in the scope of a first realization/concrete system. Against this background, we first identify core requirements for such a communication model, i.e. an abstract communication system.

Insights from Integration Projects

We start by documenting several insights collected from real world integration projects and then, based on these insights, continue by deducing communication system requirements. The research design implemented for achieving this task is an

interpretive, case study approach (Walsham, 1995). Based on one of the authors' longtime experience as a solution architect for machine-to-machine integrations for a global business software company and on both of the authors' experiences as research project leaders in the field of smart objects in industry and academia, reports and personal notes of the project leaders and documents (process and technical descriptions) of smart objects integration projects and research projects were used to analyze and illustrate the particularities and commonalities of integrating smart objects. The studies followed an explorative, multi-method case study research design (Eisenhardt 1989; Lee 1989), whereas the methods applied in the scope of the research projects included interviews with semi- structured guidelines and on-site observations. The studies addressed the overarching research question 'What do the implemented integration solutions mean for cross-enterprise data exchange with smart objects?'. The smart object projects have their origin in the utility, engineering & construction, automotive, retail and high tech industry. While the integration projects were conducted on behalf of industry clients and, accordingly, also paid by them, the research projects were non-profit and, in two cases, supported by the National German Research Foundation. The participants requested anonymity. We draw up seven key insights and interviewed experts in industrial automation in order to test and adjust our insights. While we intend to establish the parameters which can be applied to all research by means of our case studies, we point out that our list makes no claim of completeness.

Message orientation In a majority of cases, the application-level integration requirements can be satisfied with exchanging discrete units of data, i.e. messages. Notable deviations from this requirement are applications that require streaming of data, such as multi-media content.

Limited connectivity Even though devices in our definition can vastly differ in form, function and underlying constraints, security remained a key concern in all cases. To complement traditional access control mechanisms, most often network-level security systems such as firewalls, have been deployed to reduce risk by minimizing the total number of communication end points a device can be accessed through. This, in turn, often precluded systems located outside the network from establishing connectivity to a particular device because it was located behind a firewall.

Publisher confidentiality A client requesting a message to be exchanged with others regularly did not know the exact identities of the intended recipients yet it still had a vested interest that only authorized parties are given access to it. Particularly a device was often not in possession of all context information to accurately identify eligible recipients (end-points).

Multiple communication channels Mobile devices featured multiple communication channels with varying capabilities in respect to bandwidth and latency and equally different implications on cost and energy consumption. The ability to utilize a particular communication channel was heavily dependent on the state of the device, such as its power reserve or location in relation to the nearest network access point.

No real-time expectations Devices that are not themselves embedded into a higher-level device often imposed no particular hard or even soft real-time requirements for the communication with external systems. Very often, such devices were satisfied with the assumption of asynchronous communication.

Application level end-to-end There was an implicit notion of criticality for any given message exchange, which was explicit only by the specific choice of protocols. At the low end of the spectrum, connection-less protocols such as UDP (User Datagram Protocol) were favored, messages of medium importance were exchanged using TCP to provide at least some guarantee that the information arrived on the other side but for extremely critical messages applications relied on their own application-level mechanisms.

Memory loss Devices reset and lost their state such that in order to properly resume operation, they required access to previous messages they had exchanged.

Requirements for an Abstract Messaging System

Based on these insights and following the four proposed information exchange principles for smart object, we have developed a list of key requirements for a generic, i.e. non application-specific, communication system. This communication system provides for a borderless, non-mediated, two-way and flexible information exchange.

- **Message opaqueness:** the system shall be future-proof and thus not make any assumptions on the particular type of payload/message being exchanged.
- **Message confidentiality:** the sender shall be able to specify the conditions a client must meet to entitle it to receive a message. The system shall enforce these conditions and prohibit a client from accessing a message unless it meets the criteria.
- **Message spam control:** the receiver should be able to specify the conditions a sender must meet that entitles it to send a message.
- **Receiver notification control:** the receiving client shall be in control of when what messages are received and through which communication channel.

- **Continuous access control:** the system shall adhere to the criteria governing access to a message throughout the lifetime of a message - not only at the time it is sent.
- **Client state transparency:** clients shall communicate information about their identity and state to the system so it can evaluate the set of messages it has access to at any point in time.
- **Sender revocation right:** a sender shall be permitted to revoke a message when it deems necessary. The sending client should also be permitted to specify a duration after which the message is automatically revoked.

Evaluation of Established Communication Paradigms

We now evaluate existing communication approaches against requirements. We have chosen these approaches because they are representative of the many different systems that are available today. Eugster et al (2003) surveyed the publish-subscribe communication approach and compared it to other established approaches including RPC (Remote Procedure Calls) (Spector 1982; Tay and Ananda 2008), notifications and shared spaces. RPC and notifications assume a relatively strong coupling in time and space and are thus ignored here. We focus our evaluation to the publish-subscribe, the shared spaces and the message queue paradigms.

The key tenets of publish-subscribe can informally – for a formal definition see (Mühl 2002) – be described as i) a receiver must have subscribed *before* it is notified about events matching the subscription, ii) receiving clients are determined at the time a message is published, iii) the system implementing publish-subscribe actively notifies the client and iv) guarantees to do so exactly once. The direct consequences of this definition and fundamental to the publish-subscribe paradigm are:

- A client cannot access messages that were published prior to issuing a subscription.
- There is no pre-defined security concept that governs which clients can subscribe to what messages.
- A client has no control over which messages it is notified of when.

Early work on publish-subscribe systems by Segall et al (2000) already conceded that some form of persistence was a common request and that the addition of this feature created considerable complexity. Most other systems based on the publish—subscribe paradigm were also forced to extend it to allow some form of disconnectedness but this did not change the temporal ordering requirement that a client must have previously issued a subscription before it receives matching notifications.

Segall and Arnold (1997) were also one of the first to recognize the need that in a commercial context it may be desirable to ensure that only *authorized* subscribers can receive notifications. Later, Wang et al (2002) explored security issues and requirements for publish-subscribe systems in general. They coined the concept of *publication confidentiality* for the same idea and outlined two possible alternatives i) a mechanism of sharing keys independent of the publish-subscribe system (out-of-band) and ii) trusting the publish-subscribe system and extending its interface to allow the definition of an access control policy while designing such a security framework enforcing these policies. They warned that it was no trivial undertaking. Segall et al (2000) took the out-of-band approach while Belokosztolszki et al (2003) showed how role-based access control concepts could possibly be incorporated in a publish-subscribe system. Gianpaolo and Migliavacca (2008) proposed a context-aware extension to the publish-subscribe communication paradigm, which similar to us recognized that the matching semantics of traditional content-based publish-subscribe systems are not always suitable. While there have been proposals for many extensions it remains a fundamental assumption that a client must subscribe *before* it can receive any message (notification), which violates our requirement for continuous access control. Receiving clients generally have no control over when they receive what message, which is in direct contradiction to our requirement for receiver notification control.

Shared spaces, as introduced in (Eugster et al 2003), provide three basic operators to export and import tuples from an address space that is addressable by a number of distributed systems. *out* exports a tuple while *in* and *read* import a tuple with *one-of-n* and *one-to-n* semantics respectively. Fundamental to this paradigm and as described in the original work by Gelernter (1985) is, that in principle, a tuple may be received by *any* in or read statement, i.e. there is again no pre-defined mechanism to enforce which client can access what messages. We've extended our evaluation to the familiar concept of message queuing, which is popular for the device-to-device and device-to-enterprise integration scenarios. It involves an out-of-band agreement on which queues a particular client uses and the associated access control is entirely under administrative control, which limits its application to scenarios where the client population is smaller and well known.

In another area of research Fall (2003) investigated a specific class of Internets characterized by a very significant link delay, a general lack of an end-to-end routing path or end nodes without continuous power or otherwise constrained. He showed that either one of the properties fundamentally breaks assumptions the design of the IP suite relied upon. The proposed solution is a delay tolerant network architecture forming an overlay network that is capable of tying together a wide range of diverse networks via gateways that provide store and forward, routing and protocol translation capabilities. It embodies a notion of late-binding of intended recipient names that may represent an individual or a group of endpoints and therefore follows a one-to-many asynchronous notification communication style that is de-coupled in synchronization and time but only partially de-coupled in space, i.e. end-points sending a message must provide the logical address of the receiver but it cannot know whether that represents an individual or a group of receiving end-points. Table 1 summarizes our findings and indicates that

particularly the requirements message spam control and continuous access control are a poor fit for existing communication approaches. We believe that two of the root causes are that in their original designs it was not considered that a device may have multiple communication channels and that a device's context is a major factor for matching messages and not just an extension.

Requirement / Communication Paradigm	Message opaqueness	Message confidentiality	Message spam control	Continuous access control	Client state transparency	Sender revocation right	Receiver notification control
Publish/subscribe	✓ (Topic-based), ✕ (Content-based)	(✕)[a]	✕	✕	(✕)[b]	✕	(✕)[c]
Shared space	(✓)[d]	✕	✕	✕	✕	(✕)[e]	(✓)[f]
Message queue	✓	(✕)[g]	✕	✕	✕	✓	(✕)[h]
Delay-tolerant networking	✓	(✕)[i]	✕	✕	✕	✕	✕

[a] Unless extended with a notion of inversion of matching as described by [4]
[b] Ionescu and Marsic [11] extended the subscription concept to filter messages based on the client's state held on server
[c] Clients can change subscriptions but if their frequency is not considerably lower than the frequency of messages the system is in an unstable state and messages may be lost
[d] If we consider a tuple element to be opaque
[e] Conceptually, a client could remove its own message (tuple) with an `in` command but there is no notion of time-based expiration
[f] Generally lacks ability to introspect message prior to reading
[g] Access control is entirely under administrative control
[h] A client usually has a provision to browse individual messages in a queue but they're usually read in their entirety
[i] Constrained to a logical address presenting an individual or a group of clients

Table 1: Evaluation of communication approaches against key requirements

PART III: VIRTUAL OBJECT WAREHOUSING

We now outline the key concepts for a cloud-based communication model that was conceived based on the requirements described in the previous sections. Inspired by the concepts of a physical warehouse, we reuse the notion of a warehouse that keeps track of *virtual objects*, i.e. discrete units of digital information, and refer to it as the *virtual object warehouse*. Its core responsibility is to receive virtual objects from clients and make them available to eligible clients for withdrawal. All other concepts exist to support these core functions.

The key challenge is to efficiently govern what virtual objects a client may withdraw. If a virtual object represents a message this is equivalent to ensuring publication confidentiality. Moreover, a client (directly or via a delegate) informs the warehouse about the withdrawal policies it wishes to be adhered to. These may refer to information about the virtual object and/or the specific state a sending or receiving client must be in. In contrast to a physical warehouse, virtual objects are of digital form and can thus be copied and retrieved any number of times. In effect, withdrawal of virtual object from the warehouse merely provides a copy to the requesting client. This raises the question of the lifetime of a virtual object in the warehouse. As it is an application level decision and we cannot assume an infinite storage capacity of the warehouse, we anticipate the sending client to decide the duration of time a message is to reside in the virtual object warehouse before it is purged. Once a object is purged it is no longer made available for withdrawal to any client. The warehouse provides the following set of services:

Account management allows administrative entities to be created, updated and deleted. These provide a grouping function for clients and are primarily defined to associate all relevant charges incurred by utilizing any of the other services.

Client management establishes and uniquely identifies clients that are associated to exactly one account in order to securely exchange messages with the virtual object warehouse.

Matching rule management allows the definition, update and deletion of withdrawal policies that instruct the virtual object warehouse which clients it may grant to withdraw which virtual objects under what circumstances.

Message management enables a client to send and receive virtual objects to/from the virtual object warehouse and is explained in further detail below.

> **Message push** allows a client to send a message to the virtual object warehouse consisting of three main parts: a message payload, a message header and a message expiration time.
>
> **Message list** retrieves information about all messages the virtual object warehouse has made accessible to a client at that point in time.
>
> **Message pull** enables a client to retrieve a complete copy of a particular message.

We subsequently refer to the notion of a virtual object warehouse with the set of services above as the new communication paradigm of *virtual object warehousing*.

Warehouse Operation

An account groups a number of registered clients and defines the level at which withdrawal rules are defined. If a client meets any of the criteria specified in one of the rules that govern a particular combination of a sender, message and receiver and if the message has not been purged, the message appears in the client's list of accessible messages. Such a client may pull a specific message as long as it is included in the message list at that point in time.

One of the key differentiators to existing communication paradigms is that virtual object warehousing at its core has a mechanism for managing publication confidentiality. It thereby strikes a balance between decoupling clients, i.e. not necessarily having knowledge of each other a priori, the intention to restrict who may receive a message (publication confidentiality) and the ability to discover who has received what message. Fundamental to the concept is that it does not impose temporal dependencies between pushing a message and defining matching rules. Because these activities can occur in any order, clients may be entitled to receive a message although they may have not even existed at the time a message was pushed. Similarly, messages may appear or disappear from the list of available messages for a given client as there are relevant changes to a client's state over time. This provides for extremely versatile communication models.

Also of importance is the notion that a client is in control over what messages it is to pull when. That allows it to tailor a client's receipt of messages to the specific state it is in as well as the set of available communication channels (in respect to bandwidth and cost) at a specific point in time and space. This approach also ensures that a client can more easily traverse network firewalls by being the initiator of a connection – and using a protocol that can be proxied such as http – instead of listening to connections, which is often blocked due to network security constraints. Since the receiving client initiates a message pull the virtual object warehouse cannot guarantee message delivery since a receiving client may never request it. Instead, the warehouse can keep a detailed log of withdrawal events, which can be made available to the sending client to take appropriate action if such a quality of message delivery is required. We felt this to be a desirable compromise, as it does not unnecessarily impact the scalability and performance of a system implementing the described concepts.

A Cloud-Based Virtual Object Warehousing Service

Figure 4 illustrates a globally accessible, logically centralized Internet software service that implements the virtual object warehousing communication model. We call it the *Virtual Object Warehousing Service (VOWS) (Karabulut et al. 2010)*.

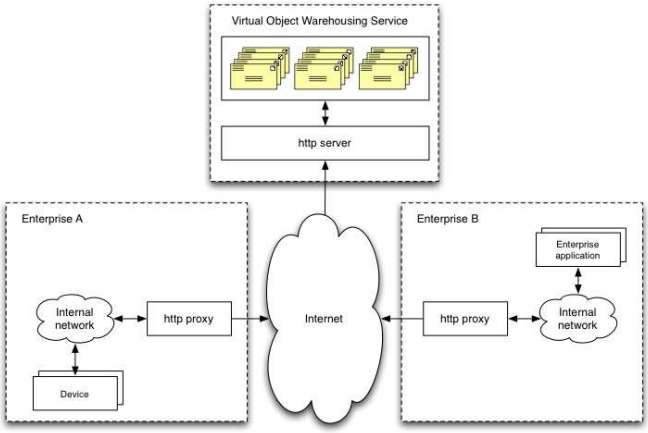

Figure 4: The virtual object warehousing service

This service provides an Internet-based integration between smart devices and business applications in a generic, flexible and scalable manner. We have already developed several iterations of such a service (blinded). The first application scenario has been in continuous operation for over 12 months and tracks the details in vehicle status of a local fleet of electric vehicle shuttles that serves the campus of a global business software company. These messages are withdrawn to visualize their location and provide insight into the battery charge status of each vehicle. In addition, the same messages have been consumed and loaded into a data warehouse where reports have been created that tracked the carbon footprint savings based on very detailed route information provided by each vehicle.

Smart Metering as a Promising Application Field

A *Virtual Object Warehousing Service* would enable the communication of millions of IP-based smart meters and many enterprise systems. Withdrawal policies would ensure that automated meter readings are flexibly allocated and only made

available to eligible market participants. Even market communication between retailers, meter operators, grid operators and generators would easily be set up to coordinate the orderly switching of a customer from one energy retailer to another – without any changes to the technical infrastructure. Both the energy retailer and the grid operator could simultaneously affect smart meters and home area networks to actively manage energy demand, e. g as part of a particular tariff or to thwart blackouts in peak situations. Smart meter operators in deregulated energy markets will face this challenge of distributing a large amount of smart meter data to ever-changing recipients. This and the magnitude of data volume require another type of operation for data collection, data communication and processes.

The UK's smart meter implementation schedule already foresees that communications with smart meters will be coordinated centrally, on a national basis (DECC 2009). The British government announced a call for tender in 2009 and is now looking for a suitable communication provider. After intensive market research and a couple of interviews with experts, we came to the conclusion that there is currently no communication solution for the deregulated market available that can handle, among others, the exchange of millions of consumption messages and price updates a day between changing market actors (e.g. 15,000 client changes per day in UK today) and across enterprise boundaries. In the face of the European Energy market, software services such as the *VOWS* therefore seem to be a promising approach.

In a scenario that was demonstrated at CeBIT 2010, the VOWS was used to integrate millions of e-vehicles, thousands of charging stations and hundreds of utility companies (all simulated) that were responsible for charging the customer and providing energy to the charging station, which need not be the same. This highlighted several strenghts of this approach such as the discovery of charging stations and the seamless exchange between the e-vehicle (a mobile smart meter which identified the account), the charging station, which recorded the exact amount of energy delivered and the two utility companies who could coordinate the roaming charges and present the transaction details including the final price directly back to the vehicle.

DISCUSSION AND OUTLOOK

In future, there will be a significantly larger number of enterprises, which will rely on an information exchange with a device and they will not necessarily be the owner of the device. Accordingly, we argue that there is a need in the emerging Internet of Things for generic, cross-enterprise information exchange services. As existing communication approaches such as publish-subscribe, message queuing and filtering architectures in systems are not appropriate, we propose the messaging system *Virtual Object Warehouse Service (VOWS)*. Its key objective is to receive units of digital information (virtual objects) from network members and make them available for withdrawal by only eligible members. While we do not maintain that communication approach is qualitatively better or worse than existing ones, we believe it applies very well in situations that transcend across enterprise boundaries and share a great degree of commonality with the requirements we documented in this contribution. We also recognize that security and, in particular, message confidentiality is a pre-requisite to the acceptance of such a service. We therefore explored in (Karabulut et al. 2010) how to use an identity-based encryption approach to ensure end-to-end message confidentiality.

Unlike the integration solutions for smart objects in use today, our proposed cloud-based messaging service tries to break the strong tie that is usually made between the communication and the application. This is like seeing the system, the VOWS, like a switch. At the same time the VOWS is able to control the information flow based on other behavioral observations. This is like seeing the VOWS like a programmable switch with QOS attributes and a provider can provide additional value-added services by, for instance, combining this service with added value services such as analytics: withdraw data from the *VOWS* into a data warehouse and then run Business Intelligence (BI) analytics. The aforementioned carbon footprint reports of the electric vehicle shuttles have been created this way.

With regard to scalability, one of the key aspects is that services such as the *VOWS* do not require reading the entire message in order to distribute it. In this manner, it is future-proof and not limited to specific applications or industries. All kinds of data from/to smart objects can be exchanged: intelligent cargo containers, e-vehicles, machines etc. Consequently, different smart objects and industries mean an improved capacity utilization and lower unit costs for a provider of such services. Today's large distributed storage systems show how the necessary scalability is both feasible and cost-effectively manageable. We therefore believe that facilitating, controlling and monitoring the secure borderless delivery of messages among internet-connected devices and enterprise applications is a promising business model.

REFERENCES

1. Allmendinger, A. and Lombreglia, R. "Four strategies for the age of smart services", in *Harvard Business Review*, (83:10), 2005, pp. 131 – 145.
2. Belokosztolszki, A., Eyers, D. M., Pietzuch, P. R., Bacon, J. and Moody, K. "Role-based access control for publish/subscribe middleware architectures" in *International* Workshop *on Distributed Event-Based Systems (DEBS03)*, ACM SIGMOD. ACM, 2003.
3. Bieser, G. "IT-Plattformen für die Geschäftsprozesse der Energiewirtschaft", in *E-Energy – Wandel und Chance durch das Internet der Energie,* Picot, A. and Neumann, K.-H. (Ed.), 2009. Springer Verlag Berlin-Heidelberg, 2009, pp. 127 – 138 (in German).

4. California Public Utilities Commission 2009, "Order Instituting Rulemaking to Consider Smart Grid Technologies Pursuant to Federal Legislation", December 29, 2009 http://goo.gl/r6Fhe
5. Chen, J., Cao, X., Cheng, P., Xiao, Y., & Sun, Y. (2010). Distributed Collaborative Control for Industrial Automation With Wireless Sensor and Actuator Networks. IEEE Transactions on Industrial Electronics, 57(12), 4219-4230.
6. Commission of the European Communities "A portrait of e-business in 10 sectors of the eu economy", *5th synthesis report of the e-business w@tch,* Technical Report, 2007.
7. Cugola, G. and Migliavacca, M. "On context-aware publish-subscribe", Fast abstract on *DEBS'08*, 2008.
8. DECC, British Department of Energy and Climate Change, Press release 02 December 2009, http://goo.gl/9cTE9
9. Dunkels, A. and Vasseur J. P. "IP for Smart Objects", *Internet Protocol for Smart Objects (IPSO) Alliance*, 2008, www.ipso-alliance.org/Documents/IPSO-WP-1.pdf
10. Eisenhardt, K. "Building theories from case studies", Academy of Management Review 14(4), 1989, 532–550.
11. EU Commission "Directive 2006/32/EC on energy-use efficiency and energy services", http://goo.gl/ljJF3
12. Eugster, P. T., Felber, P. A., Guerraoui, R. and. Kermarrec, A.-M. "The many faces of publish/subscribe", in *ACM Comput. Surv.,* (35:2), 2003, pp. 114 – 131.
13. Fall, K. "A delay-tolerant network architecture for challenged internets" in *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 27 – 34, New York, NY, USA, ACM.
14. Gelernter, D. "Generative communication in Linda", in *ACM Transactions on Programming Languages*, 7, 1985, pp. 80 – 112.
15. Gislason, D. "Zigbee wireless networking". Oxford, Newnes, 2008. http://goo.gl/3bn4M.
16. GTM Research "U.S. Smart Grid Market Forecast 2010-2015", Executive Summary, David J. Leeds, September 2010. http://goo.gl/FeimT
17. Hunt, S. and Morgan, R. "Relationship marketing in the era of network competition", in *Marketing Management*, (3:1), 1995, p. 19–28.
18. Hui, J. and Culler, D. and Chakrabarti, "6LoWPAN – Incorporating IEEE 802.15.4 into the IP architecture", *white paper No 3*, 2009. http://www.sics.se/~adam/dunkels08ipso.pdf
19. Karabulut, Y., Weppner, H., Nassi, I., Nagarajan, A., Shroff, Y., Dubey, N., Shields, T. (2010) End-to-end confidentiality for a message warehousing service using Identity-Based Encryption. ICDE Workshops 2010: 33-40.
20. Lee, A. "A scientific methodology for MIS case studies", MIS Quarterly 13(1), 1989, 33–50.
21. Mattern, F. and Flörkemeier, C.: From the Internet of Computers to the Internet of Things. Informatik-Spektrum, 2010, 33(2).
22. Mühl, G. "Large-scale content-based publish/subscribe systems", thesis, TU Darmstadt, Germany, 2002.
23. Nayyar, D. "Globalisation: The past in our present," *Presidential Address to the Seventy-Eighth Annual Conference of the Indian Economic Association*, December 28 - 30, Chandigarh, 1995.
24. Rode, J.; Küstner, M.; Majumdar, A.: Connecting Business Systems to the Real World: A Lean and Scalable Middleware for Device Integration, Tagungsband: RFID SysTech 2011, 7th European Workshop on Smart Objects: Systems, Technologies and Applications (ITG-FB 229).
25. Segall, B. and Arnold, D. "Elvin has left the building: A publish/subscribe notification service with quenching", in *Proceedings of AUUG97, 1997*.
26. Segall, B. and Arnold, D, Boot, J., Henderson, M. and Phelps, T. "Content based routing with elvin4", in *Proc. of AUUG2K*, 2000.
27. Shanbagaraman, K. "Wireless devices in the factory automation – an overview of adoption trends", Sensor Review, Volume 29, Issue 3, 2010.
28. Spector, A. Z "Performing remote operations efficiently on a local computer network", in *Communications of the ACM*, (25:4), 1982, pp. 246 – 260.
29. Strüker, Jens; Weppner, Harald; and Bieser, Gero, "INTERMEDIARIES FOR THE INTERNET OF ENERGY – Exchanging Smart Meter Data as a Business Model" (2011). ECIS 2011 Proceedings. Paper 103. http://aisel.aisnet.org/ecis2011/103
30. Tay, B. H. and Ananda, A. L "A survey of remote procedure calls", in *SIGOPS Oper. Syst. Rev.*, (24:3), 1990, pp. 68 – 79.
31. Vasseur, J. P. and Dunkels, A. "Interconnecting Smart Objects with IP: The Next Internet", Burlington, USA: Morgan Kaufman, 2010.
32. Walsham, G. "Interpretive case studies in IS research: nature and method", EJIS 4(1), 1995, 74–81.
33. Wang, C., Carzaniga, A., Evans, A., Wolf, A. L. and Wolf, E. L. "Security issues and requirements for internet-scale publish-subscribe systems", in *Proceedings of the Thirtyfifth Hawaii International Conference on System Sciences (HICSS-35)*, 2002, Big Island, USA.
34. Word, J. (Ed.) "Business Network Transformation: Strategies to Reconfigure Your Business Relationships for Competitive Advantage", Jossey-Bass, San Francisco, USA, 2009, pp. 1 – 304.
35. Wright A. "Making Sense of Sensors", Communications of the ACM, 52, 2, pp. 14-15, 2009.

ZigBee Alliance, "ZigBee Alliance plan further Integration of Internet Protocol Standards", press release, 2009. http://goo.gl/r8YlH