Embedding Internet Technology for Home Automation

Matthias Kovatsch, Markus Weiss, Dominique Guinard
Institute for Pervasive Computing
ETH Zurich
8092 Zurich, Switzerland
{kovatsch,mweiss,dguinard}@inf.ethz.ch

Abstract

As more and more digital appliances populate our homes, networking them to form a home automation (HA) system not only becomes an option, but almost a necessity. While comfort, security, and energy efficiency can be provided by many existing systems, they all remain complex islands that are difficult to expand and customize. We propose Internet technology to remedy the situation and to become the future solution for HA. For this, we analyze the feasibility of today's Internet technology with regard to traditional HA solutions. Furthermore, we present two case studies that substantiate the benefits of our proposal. As we will show, with IPv6 and 6LoWPAN, a single network serves the classical as well as emerging aspects of HA while concepts from the Web provide benefits for both, developers and users.

1 Introduction

Today's homes are populated by a plethora of appliances to support and entertain ourselves. Home automation (HA) aims for the orchestration of all these devices to provide users with real comfort but also security and the ability to monitor multiple dwellings. For this, HA systems traditionally cover heating, lighting, shading, and door/window control. However, computers and IT networks entered the home entailing modern entertainment systems that for example consist of network-attached storage (NAS) and Ethernet-enabled TVs. With information and communication technology (ICT) having incessantly become cheaper, this trend is expected to continue leading to smart appliances that enrich their primary functionality with additional services as well as context-sensitive behavior [10], e.g. the stereo reducing volume when the telephone rings or the freezer cooling down at a time when electricity is cheap. The latter also presents an example for a new objective and challenge of HA, namely taking care of energy efficiency. This is currently not only a hot topic in industry, transportation, and commercial services, but also has become central in the residential area [15].

In the past, specialized solutions such as EIB/KNX or Xcomfort[1] were designed to handle the tasks of HA. However, almost all existing solutions were developed for particular use cases and optimized only for them. Consequently, multiple additional standards are required to cover all emerging aspects of HA, e.g. DLNA[2] for multimedia equipment. HA is now facing the challenge to network and coordinate these areas which will often show a much more dynamic behavior than classical controllers. This raises the question whether the classical architectures and networking standards of the home and building automation domain are still adequate.

Already in 2001 the question appeared whether IP is "the way ahead for building automation" [5]. Back then, though, Internet technology was considered too heavy-weight for low-cost controllers, e.g. simple light switches, and thus remained an expensive idea with too much overhead. However, recent advances in the fields of wireless sensor networks and embedded Internet technology drastically change the situation. It was shown that RFC-compliant IP stacks are feasible for 8-bit microcontrollers [1] and in 2008 IPv6 solutions for small embedded devices have been developed [11, 3]. Likewise, Web protocols and servers have been successfully adapted to resource-constrained systems up to a point where light-weight HTTP servers can be deployed on inexpensive embedded devices [2]. Consequently, IPv6 now allows for a single network layer that covers all aspects relevant for HA. It is the basis for a well-known infrastructure that allows for easy application development and familiar interaction concepts from which vendors and users respectively benefit. We thus believe that Internet technology is the right choice as a future solution for HA systems.

In this paper, we first present the requirements of HA systems before we show how IPv6 meets them (sec. 2). In sections 3 we describe the most important traditional standards and compare their capabilities to IPv6. The feasibility of our vision is further substantiated with experiences gathered from two case studies (sec. 4 and 5) that demonstrate the ease of development and the ease of use gained from using Internet technology. We conclude this paper in section 6 with a discussion and an outlook on future work.

[1] www.knx.org, www.xcomfort.com
[2] www.dlna.org

978-1-4244-6850-8/10/$26.00 ©2010 IEEE

2 Internet technology for the home

Today's Internet technology, in particular IPv6 and the Web application layer, is to our belief well-suited for home automation. In this section, we first evaluate the particular features and requirements of HA systems. We then justify that our belief is well-founded by briefly explaining the relevant technological issues and by arguing that the mentioned requirements can essentially be met.

2.1 Home automation requirements

HA systems consist of networked components that cooperate and that need to be coordinated somehow. Hence, they basically form a distributed system that, compared to typical distributed computing systems, has a number of particular features and requirements.

Future-proof. Compared to the short innovation cycles of ICT, buildings are extremely long-lived as they exist for several decades. Therefore, the technology for HA has to be both well-proven and future-prove. The first refers to the reliability that has to be achieved by such systems, the latter targets the fact that once installed, the systems usually cannot easily be exchanged or upgraded.

Moderate cost. Cheap entities, like light switches or thermostat controls, have to be connected to a HA system. However, due to their numerous installation throughout the user's home, communication must be realized with inexpensive technology. Otherwise, the benefits cannot outweigh the costs for HA.

Installation overhead. HA solutions often require complex installation. This not only increases the cost but also often discourages users from its deployment. To be applied, HA systems should come at low installation overhead, in particular modifications to the building.

Configuration effort. Configuring the system should be possible without the need for qualified personnel and time-consuming instructions. In fact, in analogy to the plug and play paradigm of computers, zero configuration concepts have to be established that allow the integration and operation of new components on their arrival.

Connectivity. The systems have to provide different means of connectivity to enable remote access, interoperability among different appliances, and convergence of the diversified aspects of HA. Although gateways can bridge between different systems, the required translation effort must remain appropriate.

User interaction. Interfaces have to be easily accessible as well as understandable to meet user needs. Preferably, they are already integrated into the users' daily life. This helps to lower the usage barrier since users know how to handle such components. An appealing design of interaction and representation of information attracts the users' attention and encourages to engage with the system.

Security. Involving actuators, building security, and real costs, e.g. for energy, the systems must provide security concepts for possible threats, e.g. invasion of privacy, unauthorized access, or simply vandalism.

2.2 IPv6 for home automation

To our belief, IPv6 is a well-suited solution for future home automation systems. Thus, after having elaborated on the requirements of HA systems, we next evaluate how Internet technology performs regarding each criterion. We argue that it is ready for HA considering the ongoing trend of ever decreasing cost and increasing level of ICT in home environments, as well as the features of IPv6 and today's Web.

Matured and future-proof. A fundamental driver for the success of the Internet is the Internet Protocol Suite which has proven to be extremely future-proof. Its development going back to the '70s and a general overhaul in 1998 (IPv6) show that the involved protocols have to be acknowledged as matured networking concepts. Due to their openness and flexibility, they were able to cope with the progress in ICT and the rapid development of the Internet itself. On the one hand, the narrow waist at the network layer (IP) allows for many different physical and link layers without significantly affecting the higher layers, particularly the application. On the other hand, higher protocols, e.g. TCP and HTTP, scaled with the increasing number of participants and diversified applications. To be future-proof also requires not to be dependent on a single vendor. This is most likely to be achieved by open standards such as the widely available specifications and RFCs of the Internet Protocol Suite.

Low-cost wireless with 6LoWPAN. In terms of cost, it is critical that the hardware for communication and control remains cheap. Here, HA can benefit from the research in wireless sensor networks (WSNs). Central in this field are inexpensive embedded devices that wirelessly communicate to act as distributed sensor nodes but also as actuators, just as required for automation tasks. Recently it was shown that these resource-constraint devices are capable to run an IP network stack [1] and that IPv6 is ideal to directly include sensor networks into the existing network infrastructure [11]. In addition, the applicability of IPv6 in terms of reliability, throughput, and timeliness was proven to be feasible for WSN applications [12]. The key is an abstraction layer that provides compression and fragmentation to suit small frame sizes and low data rates. For this, a specifically designed standard called 6LoWPAN[3] was defined.

Easy installation. Using wireless embedded devices and establishing IPv6 as common network layer drastically lowers the installation overhead. Nowadays, LANs with Ethernet and Wi-Fi are typically available as backbone network in many homes. Border routers, integrated into traditional wireless routers, then extend the network via 6LoWPAN without the necessity of laying cables. Consequently, all devices participate in the same open and flexible network (see fig. 1). The low-cost, low-power 6LoWPAN modules can easily be embedded into inexpensive small appliances as well as battery-powered controllers like sensor nodes or mobile light switches. Major

[3] RFC4944: www.ietf.org/rfc/rfc4944

Figure 1. The architecture of IPv6-based HA

appliances on the other side, such as refrigerators, stoves, and expensive multimedia equipment, could access the backbone network directly. Not only does the higher price level of these appliances justify the cost of an Ethernet or Wi-Fi interface for control and energy monitoring, but also qualifies these products for additional services that require higher data rates, e.g. media streaming or interactive manuals and troubleshooting guides.

Autoconfiguration. Additionally, the elaborated networking concepts of IPv6 are available for device configuration. Classical HA systems often require a trained technician to integrate new devices and manually assign the corresponding addresses required for communication. IPv6 provides the *autoconfiguration* mechanism[4] that generates interface addresses from link-layer addresses and router advertisements. No human intervention is required for this so-called stateless approach. If tighter control over address assignment is required, a DHCPv6 server can be added. The autoconfiguration of devices can even be pushed further with service discovery, e.g. by Multicast DNS[5] (mDSN) which has already been implemented for embedded devices [18].

Wide-scale connectivity. Having embedded devices natively providing IPv6 connectivity facilitates their seamless integration to the Internet with a globally routable address thanks to the large address space. As a consequence, devices become easily addressable and accessible on a wide scale without the need for relatively complex gateway systems performing protocol translations [19]. The problem with such application level gateways is that they have to be implemented specifically for each application and also require more processing power for the translation. Global connectivity also is the first step that enables appliances to become smart [17]. The networked embedded devices are able to access information from local sensors as well as global services from the Internet. By *smart appliances* we refer to all devices participating in a HA system, from controllable lights, shades, and thermostats to a Hi-Fi system or hair dryer that reacts to a ringing phone so the user can hear it. Moreover, thanks to open standards, communication between smart appliances does not suffer from vendor lock-in anymore, opening the path to truly interoperable home appliances.

Natural user interaction. IPv6 connectivity also allows to integrate smart appliances to the de facto application layer of the Internet, namely the Web. Recent advances in the field of embedded software have led to Web servers with impressively low footprints up to a point where they can be directly deployed on inexpensive embedded devices [2]. This enables users to interact with these devices via a Web browser that is ubiquitously available and portable, i.e. on laptops, mobile phones, etc., and that offers natural interaction for users since Web browsers are well integrated into the daily life of many people [14]. For this integration, several projects unified under the umbrella of *Web of Things projects* [9] propose to reuse and adapt patterns commonly used for the Web. Smart appliances and their functionalities get transportable URIs that one can exchange, reference on Web pages, and bookmark. Appliances are then discovered simply by browsing and their loosely coupled services can be combined in the style of Web 2.0 mashups. On top of that, open source toolkits, e.g. the Google Web Toolkit, offer content-rich representations that make browser-based user interfaces suitable for controlling and monitoring home appliances.

Security. The Internet is the largest distributed system and already provides well-established security schemes like firewalls, VPN, IPSec, and SSL/TLS. These can be used to secure the backbone network of the homes and manage remote access. Despite the AES encryption/authentication mechanism of IEEE 802.15.4, 6LoWPAN still requires a better integration into the schemes mentioned before, however. Solid mechanisms for secure bootstrapping, key management, and end-to-end security have to be researched. However, this is a system independent problem for all components that cannot afford the established cryptographic protocols.

3 Home automation standards and IPv6

Among the existing automation solutions, BACnet and LonWorks[6] are standards primarily used in building automation which is a closely related field to home automation with larger scale installations and slightly different objectives. While further details on building automation can be found in [13], an important trend to notice is the shift towards IP and Web services. However, only tunneling and application level gateways are used to utilize existing IT networks and allow remote access. In this section, we first present the most important systems for HA that offer comparable features to our proposal, before we provide an in-depth comparison of the capabilities.

[4] RFC4862: www.ietf.org/rfc/rfc4862.txt

[5] Draft: tools.ietf.org/html/draft-cheshire-dnsext-multicastdns

[6] www.bacnet.org, www.lonmark.org

	X10	KNX	ZigBee	dS	IPv6
Medium	PLC, RF 310MHz (US), 433MHz (EU)	TP, RF 868MHz	RF 2.4GHz, 868MHz (EU), 915MHz (US)	PLC	Ethernet, Wi-Fi, RF 2.4GHz
Network size	2^8	2^{16}	2^{16}	2^{16}	2^{64} per subnet
Data rate	$20b/s$	$9.6kb/s$	$20..250kb/s$	$200b/s$	$250kb/s..1Gb/s$
Interface	custom solutions	application level gateway	application level gateway	Web services	UDP, TCP, RESTful Web
Maturity	1975	2002 (1990)	2004	2010	1998 (1969)
Costs	low	high	medium	medium	low
Installation overhead	low	high	low	medium	low
Connectivity	low	medium	medium	medium	high
Security	none	high (EIBsec)	medium (AES)	low (private circuits)	medium (6LoWPAN AES only)

Table 1. Home automation solutions in comparison

3.1 Traditional home automation standards

There exist several proprietary as well as open HA solutions that are used in commercially available products. We describe the most important classical HA standards that we later compare to our proposed IPv6 approach.

X10. In the residential domain, X10[7] is widespread. Being around since the '70s, this illustrates the preference for conservative technology in HA. X10's power line communication (PLC) features a low installation overhead as the existing electrical wiring is used to connect the devices. Controllers and controlled modules are plugged into the power outlets without additional modifications. In addition, bridges are available that translate radio frequency (RF) packets from wireless controllers to ordinary X10 power line control packets.

KNX is the dominating standard in Europe. It is an open EN and ISO standard, and the convergence of three previous standards, the European Installation Bus (EIB) to which KNX is downwards compatible, the BatiBUS, and the European Home Systems Protocol (EHS). On the one hand, the installation is relatively expensive and dedicated twisted pair (TP) cables have to be laid, which is difficult, especially in environments where no ductwork exists. On the other, the system is open and supported by many vendors. In addition, with EIBsec [6], KNX is the only solution providing a persistent security concept reaching from the backbone down to the controller level.

ZigBee. A relatively new standard is ZigBee which provides inexpensive, low-power wireless communication between devices based on IEEE 802.15.4. Just like other home and building automation solutions, it provides application and device profiles such as *Home Automation* or *Smart Energy* and *Dimmable Light* or *Thermostat*, respectively. These profiles simplify the configuration effort of an installation and specify the interface and capabilities of a device which allows for easy commissioning.

digitalSTROM. Aiming to become a world-wide standard for device control and energy management, digitalSTROM[8] (dS) builds on simplicity and ease-of-use. The technology is based on an IC chip that works over the existing $230V$ mains. Due to its size, it can be integrated into any device or a screw terminal placed inline with simple consumer loads like bulbs. The existing electrical wiring is used to communicate with a bus master, the *dS-Meter*. Nonetheless, one master per circuit and a *dS-Server* (dSS) to connect them have to be installed, usually in the fuse box. The dSS also requires Ethernet access as it acts as a gateway to other systems, including the Internet. For security, dS relies on the integrity of the building's circuits.

3.2 Comparison of IPv6 with traditional standards

For a general overview, table 1 provides reference values and empirical metrics corresponding to our requirements (see sec. 2.1). Configuration and user interaction are omitted here because they strongly depend on the realized functionality and thus cannot be compared well by coarse grain classifications, in our case *none*, *low*, *medium*, and *high*.

Costs and installation overhead mostly go hand in hand as both depend on the medium. Wireless solutions like ZigBee and 6LoWPAN (IPv6) are inexpensive to deploy as no cables have to be laid. This also results in a marginal installation overhead. It occurs that the two metrics only diverge for ZigBee. Its wireless installation is simple, however, this proprietary solution requests its own infrastructure that cannot profit from available facilities like LAN for 6LoWPAN/IPv6. PLC (X10, dS) is a good trade-off as it is cheap and as suitable as RF technology for rental apartments or other dwellings that do not allow modifications of the building. With less features, X10 is consequently cheaper than dS and requires no central units such as the dSS. In contrast, KNX remains a complex solution with high entry costs that offers rich functionality, though.

The overall connectivity is defined by addressability, data rate, and ubiquity of the used protocols. X10 offers poor capabilities due to a tiny address space which only provides 16 subnets with 16 devices each. Furthermore, the data rate of $20b/s$ is extremely low and only sufficient for on/off commands sent by simple push button interfaces. The limited bandwidth also applies to digitalSTROM. However, the central dSS provides classic Web services as well as an interface for browser scripting to support own control applications and visualizations which increases the connectivity. While KNX and Zig-

[7] www.x10.com

[8] www.digitalstrom.org

Bee provide average network sizes and data rates, they require application level gateways and the functionality is forced into their profile models. Because of the vast address space and seamless integration to the Internet, IPv6 provides the best connectivity. However, this openness comes along with the open question for a common interface. The next section will present a possible approach, the REST paradigm.

As mentioned in subsection 2.2, the integration of 6LoWPAN into the Internet's security schemes has not yet been solved. This is still a drawback of IPv6 and has to be addressed in future work. Apart from KNX though, no solution provides a comprehensive security concept for both, authentication and channel security.

4 Home mashups with the Web of Things

As mentioned before, a seamless integration of smart devices to the Internet brings them one step closer to the Web. Much is to gain from Web integration as it drastically eases the usually rather tedious development of applications on top of embedded computers. In this Section, we briefly introduce the basic steps towards the Web integration of smart devices further described in [9]. Next, we describe how these principles ease the development and customization of applications. We then introduce the prototype of a mashup framework. For this, we adapted an existing mashup editor to a number of Web-enabled smart appliances and thereupon developed a mobile editor.

4.1 Towards Web-enabled smart appliances

IPv6 connectivity of smart appliances integrates them to the Internet (network layer). On top, Web servers are needed on devices to truly achieve Web integration (application layer). Yet, having a Web server running on an appliances is the first step. Indeed, there is still a need for modeling access to the appliance resources in a Web-oriented manner. Several projects advocate using the representational state transfer (REST) architectural style [4] for interfacing with smart appliances. The essence of REST is to focus on creating loosely coupled services so that they can be easily reused [16]. REST is actually core to the current Web and uses URIs for encapsulating and identifying services. In its Web implementation, it also uses HTTP (1.1) as a true application protocol and its verbs, e.g. *GET*, *POST*, *PUT*, etc., to describe the basic semantic of services in a uniform manner. It finally decouples services from their presentation and provides mechanisms for clients to select the best possible formats, for instance JavaScript Object Notation (JSON). This, as well as the pervasive availability of HTTP libraries, makes REST (and HTTP) an ideal and natural candidate to build a universal application programming interface (API) for smart appliances.

To illustrate this, we Web-enabled several devices. We first developed a Web API for Sun SPOT sensor nodes [8] which offer a broad range of sensors (temperature, acceleration, light) and actuators (LEDs, equipped I/O ports). Requests for these sensors and actuators are formulated using standard URLs. For instance, typing a URL such as `http://.../sensors/temperature` in a browser, requests the resource *temperature* of the resource *sensors* of a Sun SPOT with the HTTP method *GET*. In order to avoid constant polling of the sensors, a client can also 'subscribe' to a resource by passing a callback URL to be notified whenever a condition is fulfilled. This pattern is sometimes referred to as *WebHook*[9]. As an example, *POST*ing a callback URL to `http://.../sensors/temperature/rules` alongside with a threshold will notify the callback every time the threshold is met. Similarly, we Web-enabled the Plogg energy meters[10] which allows us to control (turn on/off) and monitor the energy consumption of plugged appliances, e.g. a lamp, in a RESTful manner.

4.2 Eased job for developers, customization for users

As appliances get on the Web, applications using them can be developed with popular Web languages, e.g. HTML, JavaScript, PHP, JSON, and toolkits, e.g. DOJO, jQuery. This significantly eases the development on the vendor's side since applications can be built on languages for which a plethora of libraries and frameworks are available. Furthermore, the use of popular languages makes it easier to find adequate developers. This also unveils the possibility for external developers to create small Web applications and plug-ins on top of smart appliances. Open APIs and communities of developers have long become vital for service companies on the Web such as Facebook, Twitter, or Google. This is now also closely contributing to the commercial success of the devices. Hardware on the market such as the Chumby alarm clock[11] already have significant communities of voluntary Web developers creating dozens of small applications for each platform. As an example, the success of home entertainment devices such as the Squeezbox HiFi system[12] significantly

[9] www.webhooks.org
[10] www.webothings.com/energievisible
[11] www.chumby.com
[12] www.logitechsqueezebox.com

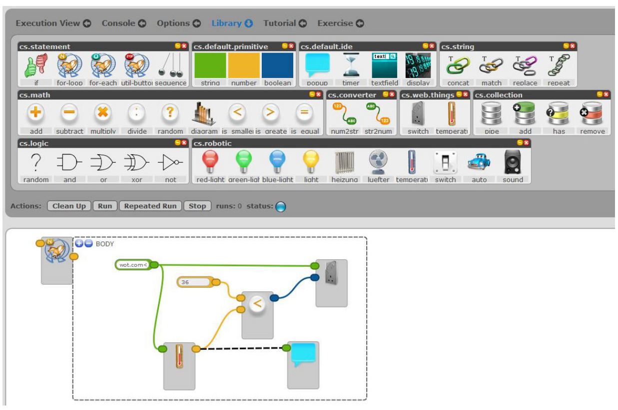

Figure 2. Clickscript Web mashup editor

depends on its ability to communicate with online music services such as Last.fm and Deezer for which plug-ins were first developed by externals.

The Web enablement of smart appliances also delivers more flexibility and customization possibilities for end-users. As an example, tech-savvys, i.e. end-users at ease with new technologies, can also build small applications on top of their appliances. Following the trend of Web 2.0 participatory services and in particular Web mashups [22], tech-savvys can create applications mixing real-world devices such as home appliances with virtual services on the Web. This type of applications is often referred to as *physical mashups* [21, 8]. As an example, a HiFi system could be connected to Facebook or Twitter in order to post the songs one mostly listens to. On the Web this type of small, ad-hoc application is usually created through a mashup editor [22], e.g. Yahoo Pipes, which is a Web platform that enables people to visually create simple rules to compose Web sites and data sources. These concepts and tools can now also be applied to empower the user to create small applications on top of their things.

4.3 Home mashup editors

To better understand the concepts and requirements of a physical mashup editor, we began by testing and extending an existing Web mashup editor, Clickscript[13]. It is a Firefox plug-in written on top of an AJAX library that allows people to visually create Web mashups by connecting building blocks of resources (Web sites) and operations (greater than, if/then, loops, etc.). Since it is written in JavaScript, it cannot use resources based on low-level proprietary service protocols. However, it can easily access RESTful services such as those provided by Web-enabled smart appliances. Thus, it is very straightforward to create Clickscript building blocks representing smart things. We used this approach to create building blocks for all the sensors and actuators of the Ploggs' and Sun SPOTs' RESTful API. As an example, we created a mashup shown in figure 2 that gets the room temperature by *GET*ting the temperature resource of a RESTful Sun SPOT. If this is smaller than 36 degrees Celsius, it turns off the air-conditioning system plugged to a RESTful Plogg.

From this early experience we extracted the requirements for building a mashup framework suitable for physical mashups and discussed them in [7]. As shown in figure 3, the system is composed of four main parts. We first have Web-enabled devices and appliances. In our prototype, we tag them with small 2-D barcodes in order to ease their identification with mobile phones. We then have 'virtual' services on the Web such as Twitter, Google Visualization API, Google Talk, etc. In the middle, the mashup server framework allows to compose services of different smart appliances as well as virtual services on the Web. It is in charge executing the work flows created by end-users in their mashup applications. It discovers, listens, and interacts with the devices over their RESTful API. The last component are the mashup editors. These applications allow users to create their mashup in a very easy manner.

[13] www.clickscript.ch

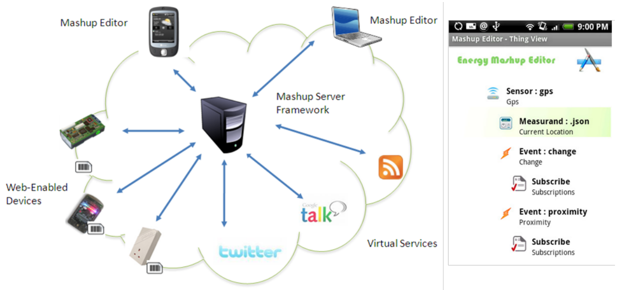

Figure 3. Physical mashup framework (left) and a mobile mashup editor using it (right)

To test the framework, we developed the *Energy Aware* mobile mashup editor. It is created for Android and makes use of the framework's RESTful API and execution engine. Thanks to the out-of-the-box availability of HTTP libraries for Android, making it communicate with the smart appliances was straightforward. Similar to Clickscript, the mobile editor allows for creating simple mashups. However, due to the screen constraints of the mobile phone, a mashup is created by going through a wizard. Users first have to select the appliances they want to include in their mashup. They do this simply by scanning the appliances' barcodes using the phone's camera. These codes are basically pointing back to the root URLs of the appliances' RESTful APIs. They then setup the rules they want to enforce and the virtual services they want to interact with. As an example, users can create a mashup that switches their appliances using the RESTful Ploggs, e.g. turning the heating on, whenever their phone detects that they are coming home (based on their GPS traces). The right part of figure 3 shows one of the wizard screens used to create this mashup.

5 The eMeter for direct feedback

In the following, we provide an overview of another contribution related to the area of home automation. The developed eMeter system fulfills typical HA tasks, namely energy monitoring and energy conservation by direct feedback. It allows users to draw conclusions on the relation between the electricity consumption and the operation of appliances [20]. We first present the architecture of the system before explaining the benefits gained from developing such a system with Internet and Web technology. The section closes with a discussion on the prototype and its integration in the users' daily life.

5.1 Smart metering architecture

The architecture of the eMeter system is based on three independent components (see fig. 4): a smart electricity

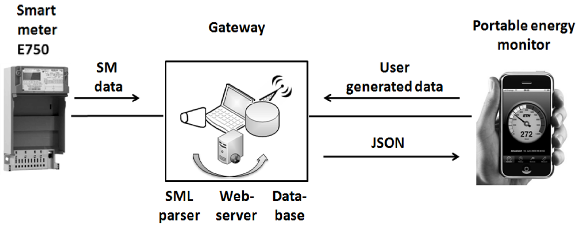

Figure 4. Smart meter communicating with the UI utilizing standard Web components

meter supplying and monitoring appliances in the household, a gateway that manages and provides access to the data acquired by the smart meter, and a portable UI on a mobile device that allows for real-time feedback on the energy consumption. For the whole system, we rely on standard components and protocols that are well-known and thus ease the development process of applications that require support for different platforms.

The first component, the smart electricity meter, measures and logs the energy consumption of devices that are attached to the electric circuit of the household. Such digital meters are installed in large numbers (as required by law in many countries) in the US and Europe over the next years anyhow. In contrast to the classical electricity meters used today, they feature a communication interface that can be polled to acquire the measurement data.

The second component, the light-weight gateway, is a software module running on an embedded computer that connects to the smart electricity meter using the Smart Message Language[14] (SML). Ideally, it will later be integrated in the smart meter and thus avoid additional installation overhead. Its main purpose is to provide the smart meter's sensor values and enhanced functionalities such as per appliance consumption in a RESTful manner on the Web. For this, it runs a MySQL database and a tiny Web server.

The third component of our prototype is the portable UI. In order to access the database and to dynamically present real-time information on the energy consumption, the UI calls the gateway URIs together with the corresponding HTTP verb and converts the resulting JSON message for presentation. It demonstrates the flexibility of the IP/REST architecture as a variety of different hardware platforms can easily be supported.

5.2 Flexible application development

The eMeter UI shows an example for flexible development on top of an Internet-based infrastructure. At first, we only provided a common Web interface to access the smart meter, i.e. a Web page that graphically presents the data gathered from the gateway's RESTful API. This UI could be used with any device running a Web browser. However, it turned out that this presentation can exceed

[14]www.sym2.org

Figure 5. Current consumption view of the mobile UI (left) and user determining the consumption of a light bulb (right)

the perception of users on mobile devices. Thus, a specialized UI was developed for the iPhone that overcomes the problem by using the familiar patterns and features of the platform.

Different views allow for intuitive illustration as well as interaction like the current overall consumption (see figure 5 left) and the detection of the electricity consumption of a single appliance (see figure 5 right), respectively. Further views allow to show historical consumption data in form of a load curve or the accumulated consumption for the previous days, weeks, or months. An inventory gives the overview of all the previously measured and saved appliances with the possibility to sort them by assigned location or the used power, so that the biggest energy guzzler appears at the top.

The new UI is realized as additional presentation layer without changing the gateway. Because of the RESTful API, information access and parsing on the iPhone are straightforward using IP connectivity via WiFi or GPRS as well as HTTP and JSON libraries. These basic features and libraries are available for every relevant platform which generally accelerates the development process.

5.3 Open API for better integration

The developed architecture shows a possibility how future electricity monitoring systems can be designed to provide real-time and fine granular feedback. Many commercially available energy monitoring systems require complex installation and provide feedback on separate displays utilizing proprietary protocols. However, since energy monitoring is a low involvement topic for many people, systems should be designed to allow for easy, encouraging interaction, preferably on devices already integrated in users' daily life. Since multi-touch screens are still too expensive for single-purpose controllers, the smart phone appears ideally capable to integrate energy feedback in daily routines of inhabitants; the required application can be installed within short time. Instead of developing yet another energy visualization from scratch, our system shows how providing a RESTful API on top of HTTP enables easy reuse of components and interoperability with other applications. In addition, it allows users to continue using familiar devices, instead of introducing ever more proprietary solutions.

6 Conclusion and future work

In this paper, we presented the requirements of home automation systems and evaluated how current Internet technology compares to the capabilities of traditional standards. Where Internet technology was once seen too heavy-weight and expensive for low-cost hardware, recent advances, such as 6LoWPAN and embedded Web servers, foster our belief in Internet technology to become the future standard of home automation utilizing IPv6 as a 'virtual installation bus' that is able to cover all emerging aspects. To summarize, IPv6 forms an established and proven system whose features are well-suited for HA. Utilizing existing LAN infrastructures lowers the installation overhead while 6LoWPAN extends the network to inexpensive wireless embedded devices. Web technology on top eases development and ensures usability as it is already part of the users' daily life. Furthermore, the Internet's ubiquity and wide range of application possibly turns 6LoWPAN devices into mass market products which then will outrank specialized systems in price and reliability.

We also presented two case studies that demonstrate further benefits of Internet technology in the home. The Web application layer allows for loosely coupled services following the REST paradigm. This way, the development process of applications is eased as standard components, libraries, and toolkits are available. For users, well-known concepts like scripting and mashups enable unprecedented customization options. For mobile access, in our case smart meter feedback on the iPhone, a standalone 'app' that fully exploits the devices' features can be better suited than a Web page. Thanks to a RESTful API and widely available support for Web programming, vendors can easily port the presentation layer to different platforms.

The convergence of networked embedded devices and the Web is still ongoing. Recent RFCs standardized most of the communication. However, future work needs to address feasible mechanisms for convenient bootstrapping and end-to-end security. In the core, the question about the optimal application layer for embedded devices is still open. While HTTP is ubiquitous, multicast and server push have to be emulated. The Constrained Application Protocol[15] (CoAP) remedies both, however, still has to be evaluated. Finally, Web standards need to be adapted to the special requirements of these new applications, in particular discovery and description, and real-time Web protocols. Here, the soon-coming HTML 5.0 specification gives good hope for a scalable and widely interoperable Web of smart appliances.

[15] Draft: tools.ietf.org/html/draft-ietf-core-coap

References

[1] A. Dunkels. Full TCP/IP for 8-bit architectures. In *Proc. MobiSys*, New York, NY, USA, 2003.

[2] S. Duquennoy, G. Grimaud, and J.-J. Vandewalle. Smews: Smart and Mobile Embedded Web Server. In *Proc. CISIS*, Fukuoka, Japan, 2009.

[3] M. Durvy, P. Wetterwald, B. Leverett, E. Gnoske, M. Vidales, G. Mulligan, N. Tsiftes, N. Finne, and A. Dunkels. Poster Abstract: Making Sensor Networks IPv6 Ready. In *Proc. SenSys*, Raleigh, NC, USA, 2008.

[4] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[5] E. Finch. Is IP everywhere the way ahead for building automation? *Facilities*, 19(11):396–403, 2001.

[6] W. Granzer, W. Kastner, G. Neugschwandtner, and F. Praus. Security in Networked Building Automation Systems. In *Proc. WFCS*, Torino, Italy, 2006.

[7] D. Guinard. Mashing Up Your Web-enabled Home. In *Adj. Proc. the International Conference on Web Engineering (ICWE 2010)*, Vienna, Austria, 2010.

[8] D. Guinard, V. Trifa, T. Pham, and O. Liechti. Towards Physical Mashups in the Web of Things. In *Proc. Networked Sensing Systems*, Pittsburgh, PA, USA, 2009.

[9] D. Guinard, V. Trifa, and E. Wilde. Architecting a mashable open world wide web of things. Technical Report 663, ETH Zurich, 2010.

[10] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. The Gator Tech Smart House: A Programmable Pervasive Space. *Computer*, 38:50–60, 2005.

[11] J. Hui and D. Culler. IP is dead, long live IP for wireless sensor networks. In *Proc. Embedded network sensor systems*, Raleigh, NC, USA, 2008.

[12] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler. Design and Implementation of a High-Fidelity AC Metering Network. In *Proc. Information Processing in Sensor Networks*, Washington, DC, USA, 2009.

[13] W. Kastner, G. Neugschwandtner, S. Soucek, and H. M. Newman. Communication systems for building automation and control. *Proc. IEEE*, 93(6):1178–1203, 2005.

[14] T. Kindberg et al. People, places, things: web presence for the real world. *Mobile Networks and Applications*, 7(5):365–376, 2002.

[15] F. Mattern, T. Staake, and M. Weiss. ICT for Green - How Computers Can Help Us to Conserve Energy. In *Proc. e-Energy*, Passau, Germany, 2010.

[16] L. Richardson and S. Ruby. *RESTful Web Services*. O'Reilly, 2007.

[17] A. Schmidt and K. Van Laerhoven. How to build smart appliances? *IEEE PerComm*, 8(4):66–71, 2001.

[18] L. Schor, P. Sommer, and R. Wattenhofer. Towards a Zero-Configuration Wireless Sensor Network Architecture for Smart Buildings. In *BuildSys*, Berkeley, CA, USA, 2009.

[19] V. Trifa, S. Wieland, D. Guinard, and T. M. Bohnert. Design and implementation of a gateway for web-based interaction and management of embedded devices. In *Proc. IWSNE*, Marina del Rey, CA, USA, 2009.

[20] M. Weiss, F. Mattern, T. Graml, T. Staake, and E. Fleisch. Handy feedback: Connecting smart meters with mobile phones. In *Proc. MUM*, Cambridge, UK, 2009.

[21] E. Wilde. Putting things to REST. Technical Report 2007-015, School of Information, UC Berkeley, 2007.

[22] N. Zang, M. B. Rosson, and V. Nasser. Mashups: who? what? why? In *Proc. CHI*, Florence, Italy, 2008.