

# **Das Web auf dem Smart-Meter**

## **IPv6, REST und andere technologische Grundlagen**

**Christina Bricalli**

**Departement für Informatik, ETH Zürich**

**bricallc@ethz.ch**

### **Zusammenfassung**

Die Herausforderung, Smart-Meters ans Internet anzuschliessen, ist eines der Hauptthemen vieler aktuellen Studien. Als Teil des Webs ermöglichen sie die Entwicklung neuer Applikationen mit schon bestehenden Programmiersprachen wie HTML, JavaScript oder PHP. So können diese Geräte einfacher global miteinander kommunizieren. Ihr Energiekonsum wäre besser überschau- und steuerbar, und sie wären leichter in schon vorhandene Webapplikationen integrierbar.

Dieser Report zeigt verschiedene Technologien auf, wie Smart Meters mit dem Web verbunden werden können. Dabei werden die Standard-Webservices (WS-\*), eine *resource-oriented* Architektur (z.B. REST) und das IPv6-LoWPAN-Protokoll vorgestellt. Zum Schluss werden einige innovative Applikationen gezeigt, die durch diese neuen Technologien ermöglicht werden.



# 1 Einführung

Wer wünscht sich nicht manchmal ein programmierbares Haus? Mit den heutigen Technologien und weiterer Forschung sollte das bald möglich sein. Wie steht das mit den Themen Smart-Energy und Energieeffizienz in Beziehung? Im Bemühen, mit den immer knapperen Energieressourcen auf der Erde auszukommen, sind Sensornetze ein wertvolles Hilfsmittel, um die Energieeffizienz von Gebäuden zu erhöhen, ohne viel von unserer gewohnter Lebensqualität einzubüßen [13].

Wenn wir die Geräte in unserem Alltag mit dem Internet verbinden und diese dann mit einem Verbrauchsmessgerät ausstatten, hätten wir einen besseren Überblick und eine bessere Kontrolle über ihren Energieverbrauch. Auf diesem Aspekt liegt im folgenden der Fokus.

Ökologische Nachhaltigkeit und der bewusste Umgang mit Energie sind heute sehr wichtig geworden und spielen in vielen Bereichen, wie zum Beispiel der Wirtschaft, der Umwelt oder der Politik, eine Rolle [10]. Es wird geschätzt, dass Konsumenten zwischen 5 und 15% ihres Energieverbrauches sparen können, wenn sie in Echtzeit sehen, wie viel Strom sie momentan verbrauchen [4]. Bei zusätzlichen Investitionen in Geräte zur Steigerung der Energieeffizienz sind noch mehr Ersparnisse möglich.

Deswegen wollen wir versuchen, diesen Konsum möglichst genau und in verschiedener Weise darzustellen. Sobald die Informationen auf dem Internet verfügbar sind, können bekannte Technologien zur Entwicklung von Webapplikationen genutzt werden, um vielfältige und innovative Anwendungen zur Visualisierung und Kontrolle des Energiekonsums zu entwickeln. Deshalb ist es wichtig, Smart-Meters mit Web-Technologien zugänglich zu machen.

## 2 Hintergrund

Smart-Meters sind Geräte, die helfen den Energieverbrauch zu messen. Es gibt verschiedene Sorten davon. Einerseits diejenigen, welche bei einem Haus z.B. beim Stromkasten fest installiert sind und den ganzen Energiekonsum des Gebäudes messen. Andererseits kabellose Apparate, welche verteilt den Energieverbrauch messen. Auf letzteren liegt der Fokus in diesem Bericht. Diese Smart-Meters messen den Energieverbrauch von verschiedenen elektronischen Geräten in Echtzeit und können deren Betrieb kontrollieren [10]. Deren Integration mit dem Web ist an sich nicht effizient: Da diese Geräte selbst Strom verbrauchen, sparen wir dadurch auch keine Energie. Die Energieersparnisse können durch die Anwendungen realisiert werden, welche auf den Smart-Meters aufbauen. Durch diese können die Energiedaten visualisiert werden und so unnötiger Stromkonsum erspart werden.

Wir wollen diese Geräte ins Web einbinden, um die von ihnen zur Verfügung gestellten Informationen vielfältig nutzbar zu machen. Als mit dem Internet verbundene Objekte wären sie wie normale Webseiten von jedem Browser aus zugänglich und erlauben so eine Kontrolle des eigenen Energieverbrauchs auf eine einfache Art und Weise. Beispiele für solche Smart-Meters sind die Ploggs oder die Current-Cost-Geräte.

### 2.1 Plogg

Ein Plogg ist eine intelligente Steckdose, die den Energieverbrauch von angeschlossenen elektronischen Geräten messen kann (Abbildung 1). Jedes Plogg ist auch ein kabelloser Sensorknoten, der via Bluetooth

oder Zigbee Daten übertragen kann. Leider ist die Schnittstelle, die von den Ploggs zur Verfügung gestellt wird, nicht offen verfügbar, was die Entwicklung von Applikationen und die Webintegration erschwert [6].

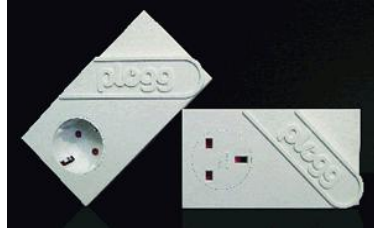


Abbildung 1: Plogg [6]

## 2.2 Current Cost

Current Cost (Abbildung 2) ist ein kleines handliches Gerät, dass überall im Haus Platz findet. Sein Aufbau ist recht einfach. Es besteht aus einem Sensormodul, die um die Elektrizitätskabel geklemmt wird, einem Transmitter und einem kleinen Display. Das neueste Modell des Displays, Envi, zeigt ausser der Temperatur und der Zeit den aktuelle Energieverbrauch des angeschlossenen Gerätes an. Der Preis des Stroms ist einstellbar, damit die Anzeige immer auf den neusten Stand ist. Ausserdem wird noch der Verbrauch der letzten Tage angezeigt. Die mit "Nacht", "Tag" und "Abend" beschriftete Säulen erlauben dem Benutzer zu sehen, zu welcher Zeit wieviel Energie verbraucht wurde. Auch ein Durchschnitt der letzten Woche und des letzten Monats ist anzeigbar. Somit ist das Display eine Uhr mit Energiedaten und lässt sich ideal in den Alltag integrieren [1].



Abbildung 2: Envi, die zweite Generationen von Displays von Current Cost. Es zeigt Zeit, Temperatur und auch noch den momentanen Energieverbrauch sowie den der letzten paar Tage an [1].

### 3 Smart-Meters als Serviceanbieter

Die Informationen von Smart-Meters sollen als Dienst im Web angeboten werden. Dafür werden Webservices eingesetzt, die ihre Daten mittels HTTP über das Netz anbieten. Diese Dienste unterstützen die Zusammenarbeit zwischen verschiedenen Anwendungen, die auf unterschiedlichen Plattformen basieren. Die zwei dafür am meisten verwendeten Technologien sind Webservices (WS-\*) und *Representational State Transfer* (REST).

#### 3.1 Webservices (WS-\*)

WS-\* verwenden ein Protokoll zum Austausch von Objekten (SOAP<sup>1</sup>) und die Webservice-Beschreibungssprache (WSDL<sup>2</sup>).

Ein anschaulicher Vergleich ist der, dass Webservices für Rechner das sind, was Webseiten für den Menschen sind. Webservices sind für Softwaresysteme gedacht, die automatisiert Daten austauschen oder Funktionen auf entfernten Rechnern aufrufen. Diese Dienste orientieren sich an der service-orientierten Architektur (SOA<sup>3</sup>) und vereinen so verteilte und objektorientierte Programmierstandards [16].

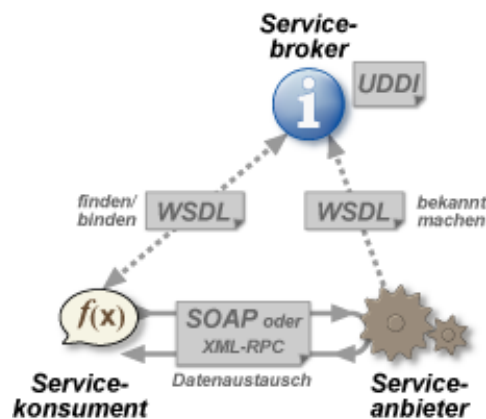


Abbildung 3: Funktionsweise von Webservices (WS-\*) [16]

Wie in Abbildung 3 ersichtlich ist, gibt es in der WS-\* -Architektur drei Rollen: ein Serviceanbieter, ein Servicekonsument und ein gemeinsames Verzeichnis (Servicebroker). Der Ablauf eines WS-\* -Aufrufs fängt damit an, dass der Serviceanbieter seine Dienste beim gemeinsamen Verzeichnis anmeldet. Dies geschieht mittels WSDL Sprache. Das Verzeichnis bietet auch einen Dienst an, und zwar UDDI<sup>4</sup>, der für die Registrierung und später das Finden der Dienste verantwortlich ist. Nachher sucht der Servicekonsument in diesem gemeinsamen Verzeichnis nach dem gewünschten Dienst und bekommt dann einen Verweis darauf. Erst jetzt entsteht die eigentliche Kommunikation zwischen Serviceanbieter und Servicekonsument, die dem Konsument erlaubt, auf Methoden des Anbieters zuzugreifen. Dieser Datenaustausch geschieht dann

<sup>1</sup>Simple Object Access Protocol

<sup>2</sup>Web Services Description Language

<sup>3</sup>Service Oriented Architecture

<sup>4</sup>Universal Description, Discovery and Integration

per SOAP oder XML-RPC<sup>5</sup>.

Webservices ist ein Standard, der verwendet wird, um von unterschiedlichen Applikationen aus einheitlich auf Geräte zuzugreifen. Obwohl dieser seinen Ursprung im Enterprise-Bereich hat, kam jemand auf die Idee, diese Dienste für eingebettete Systeme zu verwenden. Da viele verschiedene Applikationen darauf aufgebaut werden können, sind Webservices gut für Smart-Meters. Ein möglicher Anwendungsbereich für Webservices wäre in einem Sensornetz. Typischerweise werden Webservices von jederzeit laufenden Systemen angeboten. Da jedoch die Ressourcen auf Knoten in einem Sensornetz knapp sind, werden Optimierungen und Anpassungen nötig. In der Literatur wurde gezeigt, dass mit der richtigen Implementierung ein solches System in einem Sensornetz energieeffizient betrieben werden kann [12].

Der Einsatz von Webservices bietet folgende Vorteile:

- **Kompatibilität:** Sensoren können von verschiedenen Applikationen einheitlich angesprochen werden. Dies ermöglicht es, neue Anwendungen für bereits bestehende Sensornetze zu entwickeln, ohne dass an den Sensoren selbst etwas geändert werden muss.
- **Programmierbarkeit:** Die WSDL-Beschreibung eines Dienstes kann direkt von Entwicklungsumgebungen genutzt werden, um dem Programmierer den Zugriff auf einen Sensorknoten so einfach wie einen Methodenaufruf auf das richtige Objekt zu machen.
- **Integration:** Einfaches Einbinden mit bestehenden Systemen und dem Internet.

All diese Vorteile, die ein Webservice mit sich bringt, haben auch eine negative Seite, und zwar der hohe Verbrauch an Ressourcen. Dies ist in einem Sensornetz besonders problematisch, da die Knoten mit Batterie betrieben werden und deswegen nicht immer eingeschaltet sein können. Aus diesem Grund wurde das *Devices Profile for Web Services* (DPWS) entwickelt, um Energie zu sparen. Dieser Standard ermögliche den Einsatz von Webservices auf eingebetteten Systemen, also auf Hardware mit eingeschränkten Ressourcen. DPWS kombiniert hierfür eine Auswahl bestehender WS-\*-Spezifikationen zusammen mit einigen Erweiterungen [15]. Ein anderer Nachteil ist, dass diese WS-\* auf der Vermittlungsschicht des OSI-Schichtenmodells<sup>6</sup> implementiert. Sie verwenden die Anwendungsschicht nur für den Transport von Informationen. Das heißt, dass diese Webservices alle Vorteile vom Web nicht direkt nutzen können, da die dafür verantwortliche Schicht nur für den Transport verwendet wird. Deswegen müssen dann die geholten Informationen noch in einem Programm, z.B. geschrieben in C oder Java, eingebettet werden, damit sie auf dem Web verfügbar sind. Dieser Umweg der Daten muss nicht unbedingt sein, und REST zeigt eine Alternative dafür.

### 3.2 Representational State Transfer (REST)

REST ist eine Client-Server-Architektur, die bereits für das Web eingesetzt wird, z.B. Facebook verwendet eine REST-Schnittstelle. Diese Architektur hat als Ziel, jeden Gegenstand im Netz durch eine einheitliche Schnittstelle erreichbar zu machen. Clients schicken Anfragen an die Server. Diese werden dort bearbeitet, und eine Antwort wird zurückgeschickt. Anfragen und Antworten sind Dokumente, die den momentanen oder gewünschten Zustand einer Ressource beschreiben. Eine Ressource ist dabei irgendein durch eine URI eindeutig adressierbares Objekt.

---

<sup>5</sup>Remote Procedure Call

<sup>6</sup><http://de.wikipedia.org/wiki/OSI-Modell>

REST baut auf HTTP auf. Deswegen sind die Interaktionen zwischen Client und Server zustandlos. Nur vier der möglichen HTTP-Operationen sind erlaubt, was zu einer einheitlichen Schnittstelle führt [8]:

- GET: Der Client fragt nach der Darstellung einer Ressource, zum Beispiel dem aktuellen Energiekonsum eines Sensors.
- PUT: Wird verwendet, um den Status einer Ressource zu aktualisieren oder um eine neue Ressource anzulegen. Beispiel dafür ist das Ein- und Ausschalten einer Lampe.
- POST: Erstellt eine neue Ressource.
- DELETE: Löscht eine Ressource.

Dies ist für uns relevant, weil diese Idee es erlaubt, eine universelle Programmierschnittstelle (API<sup>7</sup>) für intelligente Geräte zu bauen [8]. Mittels REST kann direkt auf die zur Verfügung gestellten Daten zugegriffen werden als seien sie Webseiten. So kann man mittels einer erweiterten URI direkt auf gerätespezifische Eigenschaften zugreifen. In Abbildung 4 ist eine REST-Abfrage mit Antwort wiedergegeben. Es wird der Status einer Zimmerlampe abgefragt. Die Antwort illustriert ganz gut die Struktur des HTTP-Protokolls. Das JSON<sup>8</sup>-Format zeigt die Einfachheit, mit der die notwendigen Daten zurückgeliefert werden. Diese sind gut dargestellt und ermöglichen ein leichtes Verständnis und Integration in andere Anwendungen.

```
GET /EnergieVisible/SmartMeters/RoomLamp [...] HTTP/1.x
200 OK
Content-Type: application/json
{
  "deviceName": "RoomLamp",
  "currentWatts": 60.52,
  "KWh": 40.3,
  "maxWattage": 80.56,
  ...
}
```

Abbildung 4: Beispiel für eine REST-Abfrage mit Antwort [8]

Stärken von REST sind auch, dass es mehrere Formate der Daten unterstützt und gut skalierbar ist. Da im Laufe der Zeit immer mehr Geräte ans Internet angeschlossen werden, gibt es auch mehr Knoten. Diese sind dann einfach mit einer neuen URI ausstattbar und so schon mit dem Web integriert. Deswegen ist die Skalierbarkeit von zentraler Bedeutung. Ein weiterer Vorteil ist, dass REST die bereits bekannte Infrastruktur des Internets verwendet. Deshalb sind schon viele Optimierungen vorgenommen worden, und es gibt schon etablierte REST-Frameworks. Caching, Browsing und Bookmarking können gleich wie beim Web verwendet werden, da REST direkt mit der Anwendungsschicht integriert ist und diese nicht nur zum Datentransport (wie bei den WS-\*) verwendet.

REST hat aber einen grossen Nachteil im Vergleich zu den Webservices: die Sicherheit. Die Sicherheitsaspekte bei REST müssen z.B. in den Bereichen Transitionen und Verschlüsselung noch verbessert werden.

---

<sup>7</sup>Application Programming Interface

<sup>8</sup>JavaScript Object Notation

## 4 Die Verbindung zum Internet

Um all diese Vorteile auch nutzen zu können, müssen Smart-Meters mit dem Web verbunden werden. Eine Möglichkeit ist mittels IPv6 LoWPAN<sup>9</sup> (kurz 6LoWPAN) oder sonst durch Smart-Gateways. Smart-Meters können entweder über IPv6 direkt mit dem Internet verbunden werden oder, wie für nicht IPv6-fähige Geräte, wird die Verbindung durch Smart Gateways hergestellt.

### 4.1 IPv6 und 6LoWPAN

Der Internet-Protokoll Version 6 (IPv6) ist der Nachfolger des heute am meisten verbreiteten IPv4. IPv6 stellt mehr Adressen zur Verfügung, da im Vergleich zu IPv4 128 statt 32 Bits für Adressen verwendet werden. Es wird auch bevorzugt, da es eine effizientere Implementierung ermöglicht [9]. Dieses Protokoll spezifiziert die Vorgänge, die zur Vermittlung von Daten durch ein paketvermittelndes Datennetz notwendig sind, wie etwa die Adressierung der beteiligten Netzwerkknoten und den Vorgang der Paketweiterleitung zwischen Teilnetzen.

Die Zielgeräte für IP-Networking sind die, die eine kabellose Verbindung brauchen. Diese haben aber meistens nur knappe Ressourcen zur Verfügung und müssen so mit wenig Leistung und tiefen Datenraten umgehen können. Ein treffendes Beispiel sind genau unsere Smart-Meters. Deshalb wurde 6LoWPAN entwickelt, der speziell für eingebettete Systeme ist. Dieser Standard enthält Verpackungsmethoden und Kompressionsmechanismen, die es erlauben, Pakete in kabellosen Netzwerken effizienter zu übertragen. Zigbee beispielsweise baut auf 6LoWPAN auf [11]. Auf jeden Smart-Meter, der nicht selber IP-fähig ist, wird ein selbständiger Webserver eingebaut. So werden diese Messgeräte direkt mit dem Web verbunden (Abbildung 5 rechts). Dies bietet den Vorteil, dass das Hinzufügen von zusätzlichen Smart-Meters unabhängig voneinander erfolgen kann, da diese ihren eigenen Webserver haben und dieser so spezifisch für sie eingerichtet werden kann.

Heute gibt es aber immer noch Geräte, die nicht IP-fähig sind. Für diese werden dann Smart-Gateways eingesetzt, um die Überbrückung zum Web zu erlauben.

### 4.2 Smart-Gateway

Smart-Gateways sind Geräte, die die eigentliche Kommunikation zwischen Smart-Meters und dem Web hinter einer Webservice- oder REST-Schnittstelle verstecken. Sie verstehen die Low-level-Protokolle der Smart-Meters und stellen die Kommunikation nach aussen her. Sie erlauben den Zugang zum momentanen Status der einzelnen Smart-Meters (Abbildung 5 links). Daneben bieten sie auch zusätzlich aggregierte Informationen an, zum Beispiel die gesamte Menge an konsumierter Energie zur gegebenen Zeit [8].

Ein Gateway kann Informationen über Ressourcen in verschiedenen Formaten zurückgeben. So zum Beispiel als (X)HTML, um direkt von einem Browser aus darauf zuzugreifen, oder im JSON-Format zur Weiterverarbeitung durch Webapplikationen.

Smart-Gateways ermöglichen, komplette web-basierte Überwachungsapplikationen und auch einfache und nützliche Interaktionen wie Bookmarking und Steuerung dieser Geräte. Natürlich ist es ein Hauptvorteil, dass all diese Informationen von jedem Browser aus erreichbar sind.

---

<sup>9</sup>Internet Protocol Version 6 over LoW Power wireless Area Networks



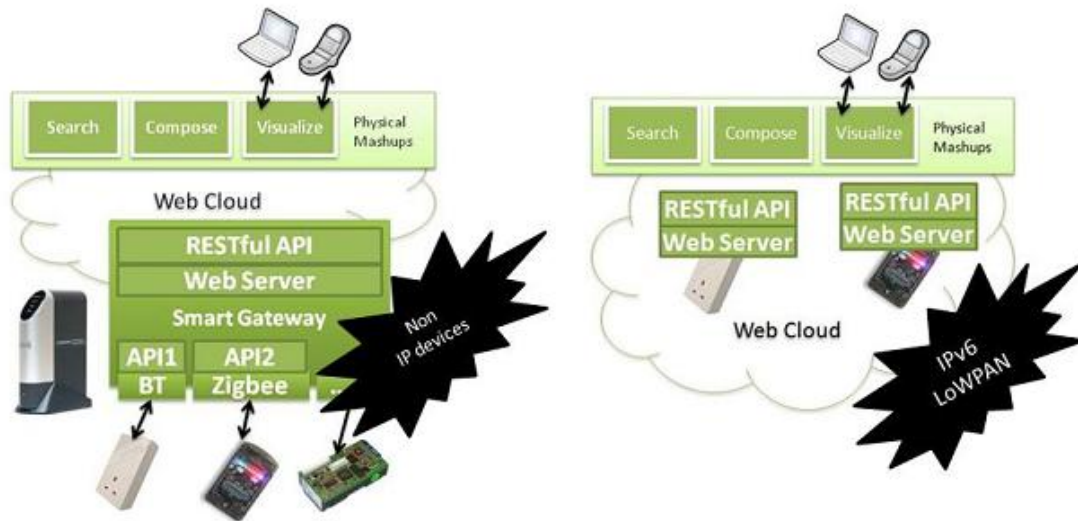


Abbildung 5: Web und Internet Integration von Smart Metern mit Smart Gateways (links) und direkte Integration mit IPv6 (rechts) [8].

### 4.3 digitalSTROM

Wieso drahtlose Datenübertragung verwenden und neue Smart Meters einbauen, um die Daten ins Netz einzuspeisen, wenn alle Geräte bereits mit einem Stromkabel ausgestattet sind? Mit dieser Frage hat sich digitalSTROM auseinandergesetzt und präsentiert eine Lösung, die an der ETH Zürich entwickelt wurde [2].

Bei dieser zukünftigen Technologie wird, im Gegensatz zu Smart Gateways und IPv6, nicht via Bluetooth oder Zigbee kommuniziert, sondern über das bestehende Stromnetz. Was es dazu braucht sind digitalSTROM-fähige Geräte. Dann müssen diese nur an der Steckdose angeschlossen werden. Da alle Geräte miteinander über das Stromnetz verbunden sind, können diese untereinander kommunizieren und sind von jedem Schalter aus steuerbar.

Diese Lösung ist billiger als die momentanen, da sie einfach ist und keine zusätzlichen Smart-Meters nötig sind. Sie bietet mehr Komfort, da beim Verlassen des Hauses einfach mit einem Knopfdruck alle Geräte am Stromnetz ausgeschaltet werden können, was wiederum auch mehr Sicherheit bietet. Energiegewinn gibt es auch, da ein Kühlschrank mittels digitalSTROM zum Beispiel lernen kann, mehr Strom zu verbrauchen, wenn die Stromtarife niedriger sind.

Das klingt wie die lang erwartete Lösung, jedoch ist diese Methode noch in der Entwicklungsphase. Ausserdem müssen die Hersteller zuerst standardmässig ihre Geräte mit den entsprechenden Chips ausstatten, bevor das System ohne grossen Aufwand installiert werden kann. Was noch hinzu kommt, ist, dass diese Daten noch nicht mit dem Web verbunden sind. Dafür müsste noch ein kleiner Webserver z.B. im Stromkasten installiert werden.

## 5 Applikationen

Im folgenden werden einige Applikationen vorgestellt, die die oben genannten Technologien verwenden. Die Veröffentlichung von Verbrauchsdaten kann aus einer Sicherheitsperspektive problematisch sein, wird aber in dieser Report nicht weiter angeschaut.

### 5.1 Energie Visible

*Energie Visible* ist ein Projekt der ETH Zürich, das auf dem Plug&Play-Prinzip beruht. Die frei verfügbare Software hilft jedem, seinen eigenen Energiekonsum in Echtzeit zu visualisieren und zu steuern. Um Energie Visible verwenden zu können braucht es zusätzlich zur Software auch noch mindestens einen Plogg.

Als erstes werden Ploggs an Steckdosen angeschlossen. Dann werden die gewünschten Haushaltsgeräte an diese angeschlossen und die Software auf dem Computer gestartet. Diese scannt die Umgebung auf Bluetooth-Geräte. Die Ploggs melden sich beim Software-Gateway an und schicken ihm ihre Energiekonsumsdaten via Bluetooth. Das Smart-Gateway erteilt jedem Plogg eine eindeutige URI. So stehen durch dieses Gateway die Funktionalitäten der Apparate mittels einer einfachen REST-Schnittstelle im Web zur Verfügung (Abbildung 6).

Die dazugehörige Anwendung zeigt dann Graphen der verschiedenen Geräte und ihren Energiekonsum im Verlaufe der Zeit. Die überwachten Geräte können direkt über diese Website ein- und ausgeschaltet werden. Ziel von Energie Visible ist es, den eigenen Energiekonsum visualisieren zu können und unnötigen Energieverbrauch, zum Beispiel durch Geräte im Standby-Modus, zu minimieren [3]. Der Nachteil dieser Lösung ist, dass pro Gerät ein Plogg gekauft werden muss. Das ist nicht so günstig.

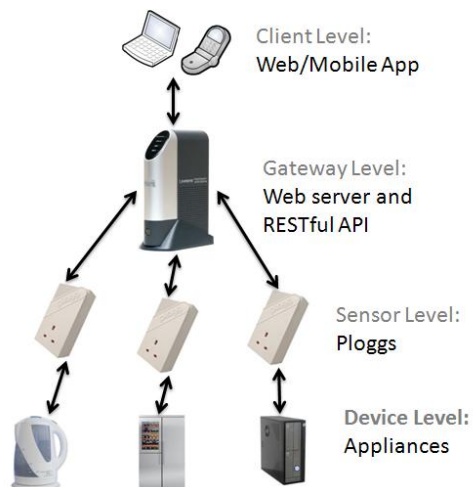


Abbildung 6: Aufbau von Energie Visible [8]

## 5.2 eMeter

Die neusten Mobiltelefone sind alle mit Webzugang ausgestattet. Das eMeter ist eine handliche, attraktive und installationsgünstige Lösung, um Energiekonsumdaten immer bei sich zu haben, und wurde an der ETH Zürich entwickelt. Diese Lösung basiert auf dem Ausstatten von Smart Metern mit REST-Schnittstellen.

Dieses Projekt bietet ein interaktives Lastbeobachtungssystem, welches augenblickliches Feedback zur Energieverwendung von Haushaltsapparaten und einzelnen Geräten liefert. Das System ermöglicht dem Benutzer, grosse Energieverbraucher zu identifizieren und hilft ihm, ihren Energiekonsum zu optimieren [14].

Diese Applikation bietet den Benutzern auch die Möglichkeit, interaktiv Messungen und Konsumdaten zu vergleichen sowie die Kosten und das Energieverbrauch der Geräte nachzuschauen.

## 5.3 Google PowerMeter

Google PowerMeter ist ein freies Energieüberwachungstool, das Konsumenten hilft, Energie und Geld zu sparen. Diese Software erlaubt den Konsumenten, ihren Energiverbrauch von überall her online zu sehen (Abbildung 7). Die dazu nötigen Energiedaten werden von den Smart-Metern bereitgestellt [4]. Diese werden dann von den Stromanbietern in den richtigen Formaten verpackt und an Google geschickt.

Momentan sind aber nur einige Stromkonzerne am Projekt beteiligt, das heisst, nur wenige Stromkonzerne erlauben Google den Zugriff auf ihre Datenbanken, um diesen Projekt am Laufen zu halten.



Abbildung 7: Darstellung des Energieverbrauchs mit Google PowerMeter [4]

## 5.4 Pachube und Sensor.Networks

Pachube ist ein Webservice. Er erlaubt das Speichern, den Zugriff und das Auffinden neuer Sensoren, Energie- und Umgebungsdaten von Apparaten und Gebäuden in der ganzen Welt. Es ist eine zentrale Datenbank, eine Sammlung von Sensordaten, wo alle Kunden ihre Energiedaten speichern und in Echtzeit veröffentlichen. Pachube funktioniert ähnlich wie YouTube, nur dass hier Leute ihren Energiekonsum beobachten können und ihre aktuelle Energiedaten mit anderen teilen können [5].

Pachube vereinfacht die Applikationsentwicklung dank einer umfassenden REST-Schnittstelle in verschiedenen Formaten. Die Informationen können von Geräten, welche REST-Schnittstellen ansprechen können, direkt geschrieben werden, oder Pachube holt sie sich von Geräten, die die benötigten Informationen zur Verfügung stellen.

Ein anderer Vorteil ist, dass es zum Beispiel mit einem Current-Cost-Gerät möglich wird, nicht nur Energieinformationen zu sehen, sondern auch andere nützliche Applikationen darauf aufzubauen. Dies erlaubt die Entwicklung von informativeren Anwendungen. Eine solche Anwendung ist zum Beispiel "Carbon Footprint", die es erlaubt, den verursachten CO<sub>2</sub>-Ausstoß der Geräte in Echtzeit zu berechnen [5].

Ein ähnliches Konzept wie Pachube bietet auch Sensor.Networks als globale zentrale Datenbank für den Austausch von Sensorinformationen. Es erlaubt Forschern und Wissenschaftlern, ihre Daten mit anderen vertrauten Partnern zu teilen. Dieser Service bietet auch Tools für die Analyse und Visualisierung der Daten. Sensor.Networks verwendet JavaScript, das von allen Hauptbrowsern wie Firefox, Internet Explorer und Safari unterstützt wird [7].

## 6 Schlussfolgerungen

Es gibt verschiedene Varianten zur Ausstattung eines Haushalts mit Smart-Meters. Alle haben ihre Stärken und Schwächen, aber im Moment deckt keine das allgemeine Bedürfnis ab, das heisst keine davon ist billig, attraktiv und frei von Installationsaufwand. Je nachdem was der Konsument sucht, muss er die für ihn passendste Lösung wählen.

Im Hinblick auf die Zukunft werden immer mehr Smart-Meters mit dem Netz verbunden. Das Internet ist nicht mehr nur ein Computernetzwerk, sondern hat sich zu einem Netz von Apparaten aller Arten und Grössen entwickelt. Diese teilen ihre Daten und kommunizieren ständig untereinander. Jeden Tag werden wir mehr und schneller neue Daten produzieren. Das wird zu einer sehr grossen Datenflut führen. Diese neuen Informationen, nicht nur über den Energiekonsum, können in verschiedenen Bereichen Vorteile bringen. Beispielsweise können durch geeignete Anwendungen das Umweltbewusstsein von Konsumenten gesteigert werden [7].

Wie bei digitalSTROM schon erwähnt, wird ein anderes Ziel in der Zukunft sein, Daten von Smart-Metern direkt über die elektrischen Kabel ins Web zu bringen anstatt mittels kabelloser Netzwerke. Diese Technologie erfordert aber noch Forschung, da Interferenzen mit dem normalen Stromnetz zu Problemen führen könnten.

Auf der Applikationsseite wird es so sein, dass immer mehr Leute Anwendungen für Smart-Meters schreiben werden, da dank REST-Schnittstellen nicht nur Programmierer einfach auf die Daten zugreifen können. Auf diese Weise wird es irgendwann auf sozialen Netzwerken wie Facebook möglich sein, die eigenen Energieverwendungsdaten mit anderen zu vergleichen. Das kann dann zu Gruppenzwängen oder Konkurrenzspielen führen, und somit zu noch mehr Energieersparnissen. Und das alles ist nur möglich, wenn Smart-Meters mit dem Internet verbunden sind.

## Literatur

- [1] Current Cost [Online]. <http://www.currentcost.com/index.html>, April 2010.
- [2] digitalSTROM [Online]. <http://www.digitalstrom.org/>, April 2010.
- [3] Energie Visible [Online]. <http://www.webofthings.com/energievisible/>, April 2010.
- [4] Google PowerMeter [Online]. <http://www.google.org/powermeter/>, April 2010.
- [5] Pachube [Online]. <http://www.pachube.com/>, April 2010.
- [6] Plogg [Online]. <http://www.plogg.co.uk/index.htm>, April 2010.
- [7] Sensor.Networks [Online]. <http://sensor.network.com/rest/login.jsp>, April 2010.
- [8] D. Guinard, V. Trifa, and E. Wilde. Architecting a Mashable Open World Wide Web of Things. *Technical Report 663, ETH Zurich, Institute for Pervasive Computing*, Februar 2010.
- [9] J. W. Hui and D. E. Culler. IP is dead, long live IP for wireless sensor networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, pages 15–28, Raleigh, NC, USA, 2008. ACM.
- [10] A. Kamlaris, V. Trifa, and D. Guinard. Building web-based infrastructure for smart meters. In *Proc. of the First Workshop on Energy Awareness and Conservation through Pervasive Applications at Pervasive 2010*, Helsinki, Finland, May 2010.
- [11] M. Kovatsch, M. Weiss, and D. Guinard. Embedding Internet Technologies for Home Automation. In *Proc. of the 15th IEEE Int. Conference on Emerging Technologies in Factory Automation*, Bilbao, Spain, sep 2010.
- [12] N. B. Priyantha, A. Kansal, M. Goraczko, and F. Zhao. Tiny Web Services: Design and Implementation of Interoperable and Evolvable Sensor Networks. *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys 2008)*, Raleigh, NC, USA, 5.-7. November 2008.
- [13] L. Schor, P. Sommer, and R. Wattenhofer. Towards a Zero-Configuration Wireless Sensor Network Architecture for Smart Buildings. *Proceedings of the 1st ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings (BuildSys 2009)*, Berkeley, California, USA, 3. November 2009.
- [14] M. Weiss, D. Guinard, T. Staake, and W. Roediger. eMeter: An interactive energy monitor. In *Adjunct Proceedings of Ubicomp 2009 (International Conference on Ubiquitous Computing)*, Orlando, Florida, USA, sep 2009.
- [15] Wikipedia. Devices Profile for Web Services [Online]. [http://de.wikipedia.org/wiki/Devices\\_Profile\\_for\\_Web\\_Services](http://de.wikipedia.org/wiki/Devices_Profile_for_Web_Services), April 2010.
- [16] Wikipedia. Webservices [Online]. <http://de.wikipedia.org/wiki/Webservice>, April 2010.