



Web of Things

Krzysztof Lis







MY HOUSE



MY DISHWASHER

22

Minutes



MY REFRIGERATOR

-1° 38°

Freezer

Refrigerator



DUET DRYER

1 25

Hour

Minutes



DUET WASHER



Appliances



Energy



Settings

Komunikacja

- Wymagania:
 - Niewielkie zużycie zasobów
 - Uniwersalny standard
- „Internet Rzeczy”
 - Warstwa internetu
 - IP
- Co w warstwie aplikacji?
 - Też potrzebny standard...

Web Service

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 299
SOAPAction: "http://www.w3.org/2003/05/soap-envelope"
```

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <m:GetStockPrice xmlns:m="http://www.example.org/stock">
      <m:StockName>IBM</m:StockName>
    </m:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

Web of Things

- Sama komunikacja to nie wszystko
- Potrzebna architektura
- Idea WWW
- Integracja z resztą sieci

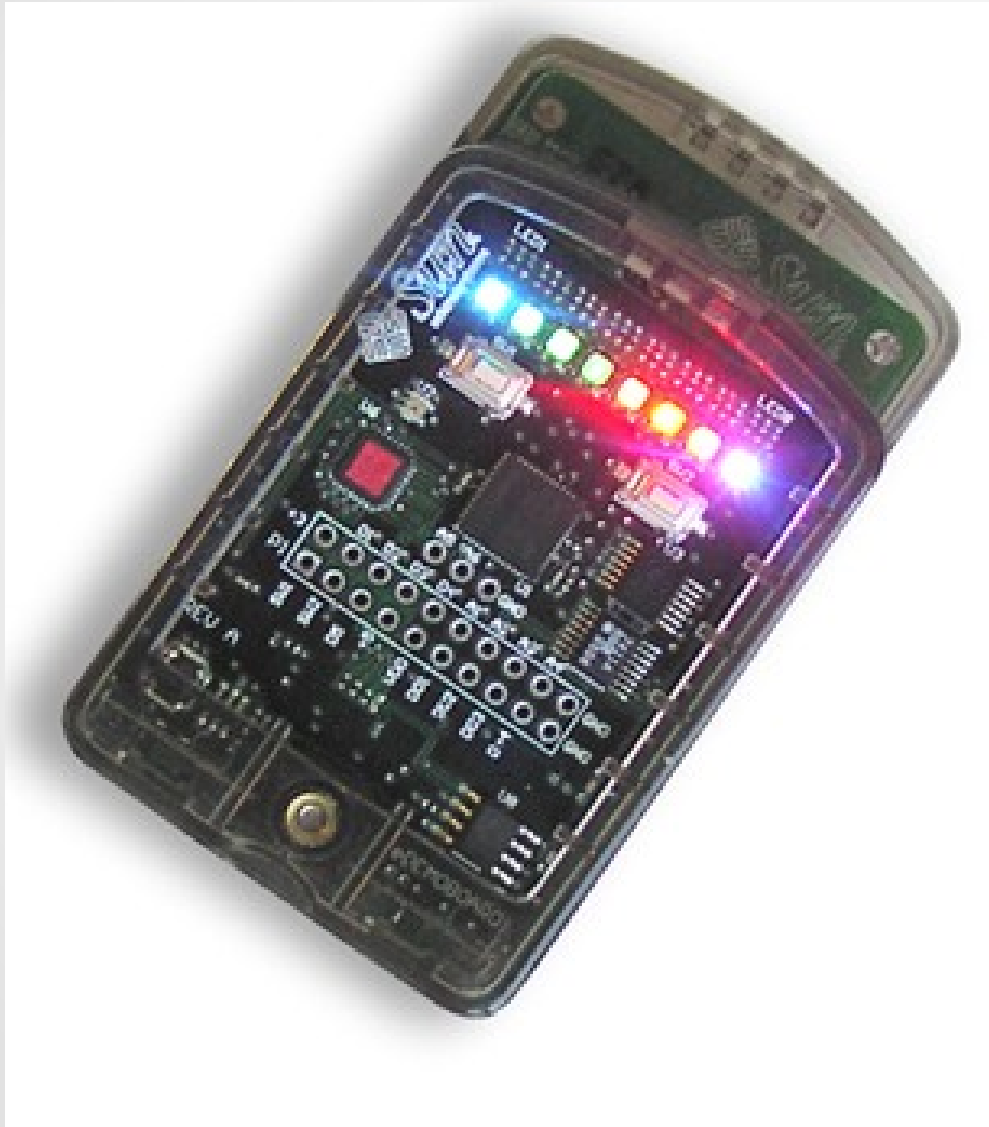
Technologie WWW

- Sprawdzone i rozwijane
- Dla maszyn: JSON/XML
- Dla ludzi
- Przetwarzanie po stronie klienta

REST

- Klient-serwer
- Bezstanowość
- Cache
- Warstwowość
- Kod dla klienta (niekoniecznie)
- Jednorodny interface

Sun SPOT



Architektura zorientowana na zasoby

- REST: zasób identyfikowany przez URI
- Podróżuj po hierarchii zasobów
 - <http://.../sunspots>
 - <http://.../sunspots/spot1>
 - <http://.../sunspots/spot1/sensors>
 - <http://.../sunspots/spot1/sensors/light>
 - <http://.../sunspots/spot1/sensors/temperature>
- Odnośniki i manipulacja URI

Reprezentacja zasobów

```
{
  "resource": {
    "methods": ["GET"],
    "name": "Temperature",
    "children": [],
    "content": [
      {
        "description": "Current Temperature",
        "name": "Current Ambient Temperature",
        "value": "27.75"
      }
    ]
  }
}
```

Web of Things - Resource Temperature



Home



Parent



Refresh



Atom Feed



Get current
temperature: 24.0 C

Interface

- **GET** .../spot1/sensors/temperature
- **PUT** .../sunspots/spot1/actuators/leds/1
{“status”: “on”}
- **POST** .../spot1/sensors/temperature/rules
{“threshold”: 42}
- **DELETE** .../spot1/sensors/temperature/rules/1
- **DELETE** .../sunspots/spot1
- **OPTIONS** .../sunspots/spot1/sensors/light

HTTP: Negocjacja formatu

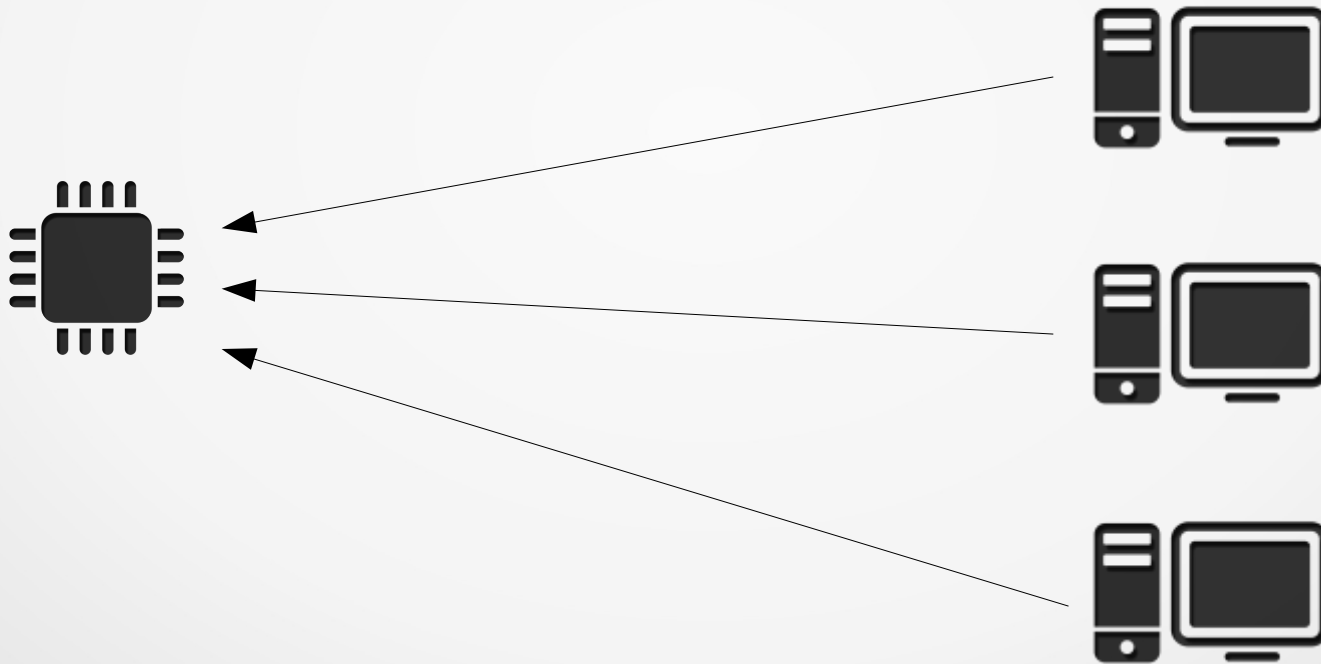
- HTTP pozwala negocjować format zwracanych informacji poprzez ustawienie odpowiednich pól w nagłówku zapytania
- **GET** .../spot1/sensors/temperature HTTP/1.1
Accept: application/json;q=1, application/xml;q=0.5

HTTP: kod odpowiedzi

- Standardowe odpowiedzi, łatwe do wygenerowania na prostym urządzeniu
- POST .../sunspots/spot1/sensors/light
...
→ 405 Method not allowed

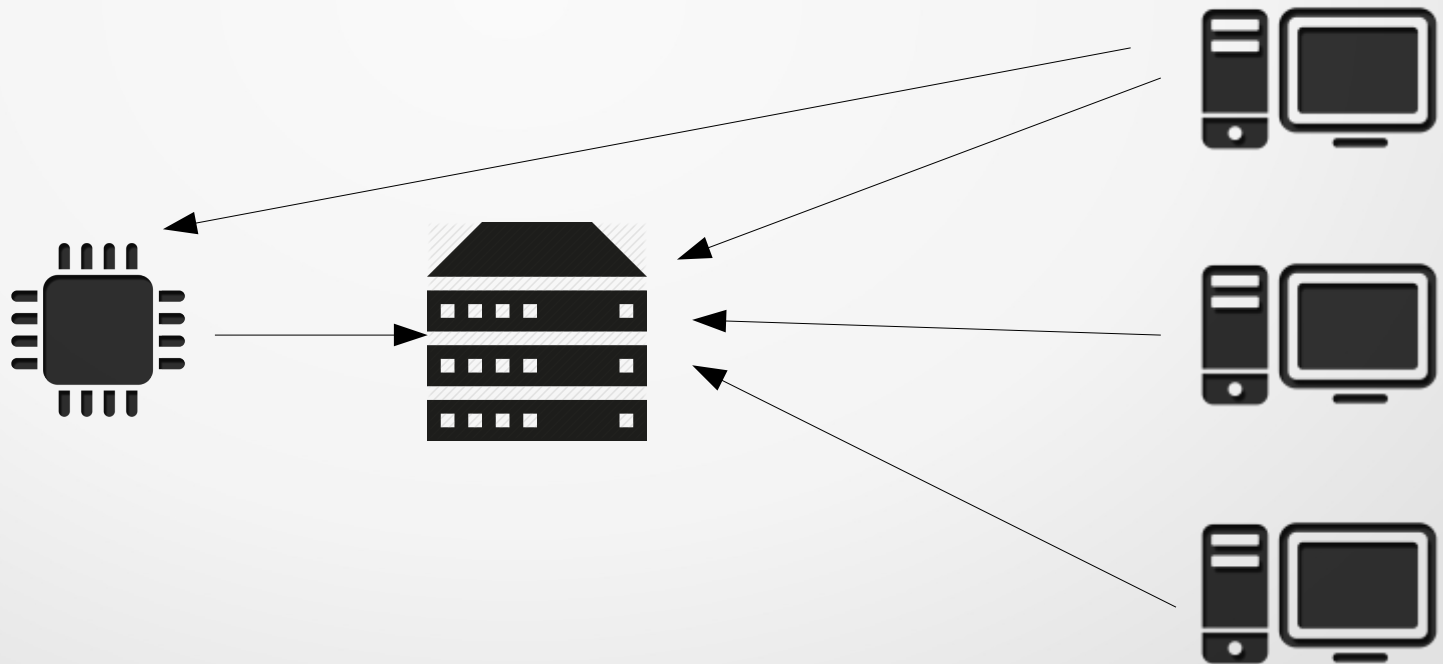
Agregacja

Gdy wielu klientów chce korzystać z czujnika:



Agregacja

- Atom
- Tworzenie kanału:
POST .../spot1/sensors/temperature/rules
{“threshold”: 42}

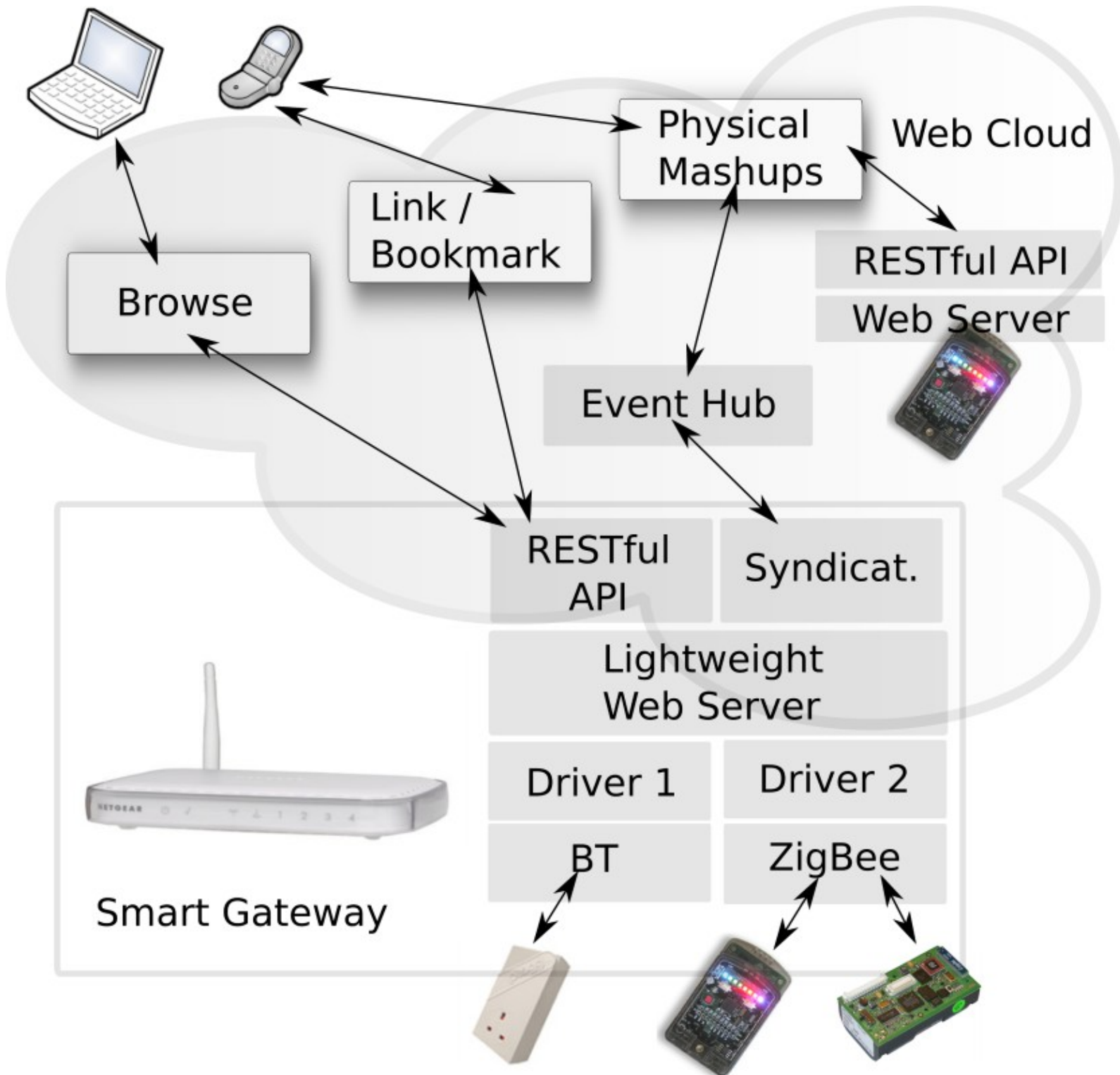


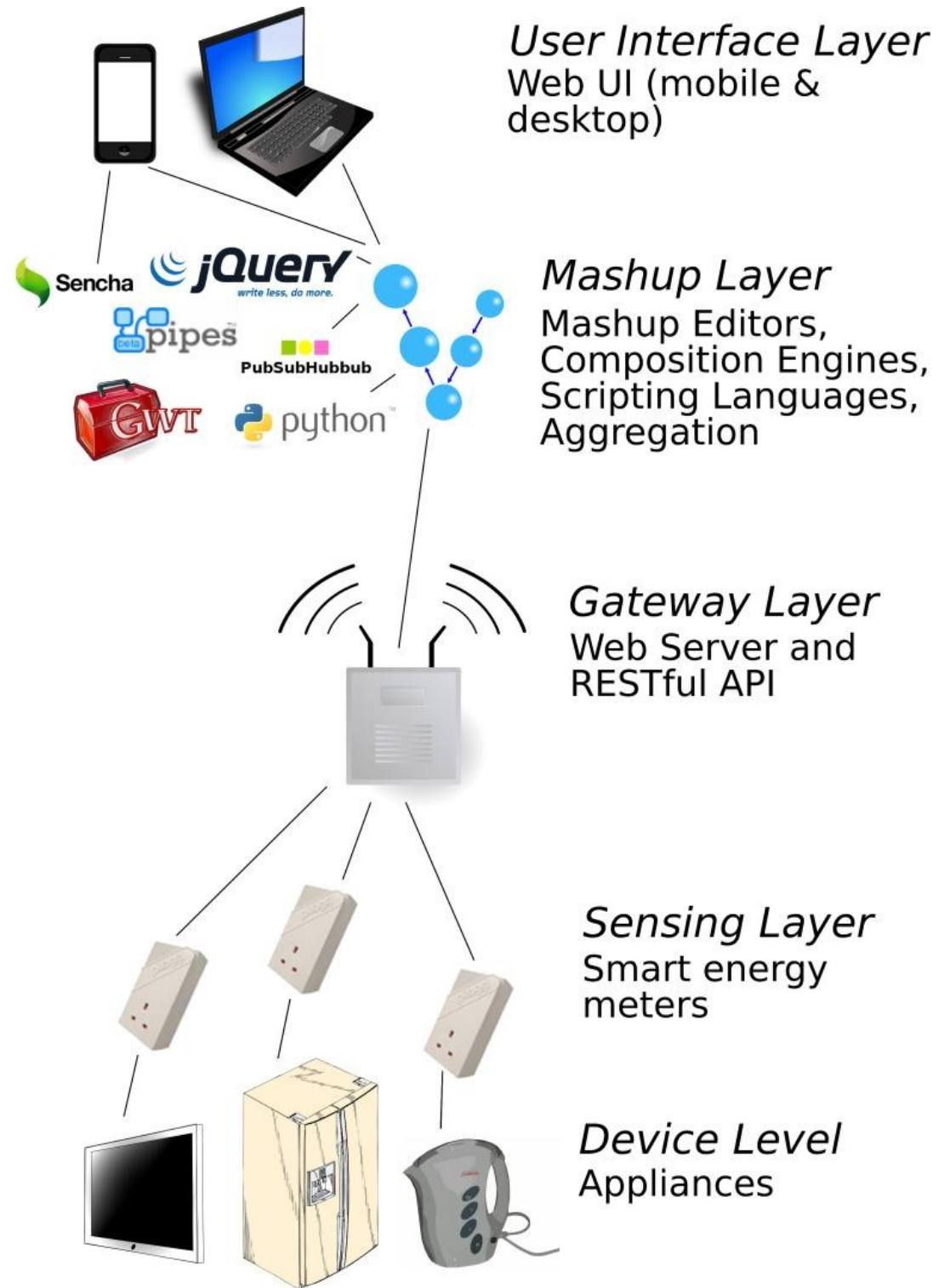
Agregacja

- Sensor odciążony, ale klienci nadal muszą odpytywać regularnie
 - Problematiczne dla urządzeń, które powinny czekać na zdarzenie
- Webhook
 - <http://.../sunspots/spot1/sensors/light/rules>
POST z cechami zdarzenia i adresem zainteresowanego
 - Urządzenie wyśle do niego zapytanie...
jak nie będzie za zaporą

Pośrednictwo

- Nie każde urządzenie może być serwerem HTTP
- REST: warstwowość
- Potrzebny pośrednik
 - Nie widać różnicy
 - Może świadczyć dodatkowe usługi





GET → <http://.../EnergieVisible/SmartMeters/RoomLamp>

GET /EnergieVisible/SmartMeters/RoomLamp

[...] HTTP/1.x 200 OK

Content-Type: application/json

{

“deviceName”: “RoomLamp”,

“currentWatts”: 60.52,

“KWh”: 40.3,

“maxWattage”: 80.56

“links”:

[{“aggregate”: “../all”},

{“load”: “../load”},

{“status”: “/status”}]

}, {...}]

Sieć w czasie rzeczywistym

- HTTP nie jest optymalny do komunikacji w czasie rzeczywistym
 - Bezstanowy
 - Architektura klient-serwer
 - Wymaga odpytywania
- RTP / RTSP – protokoły do przesyłania (zazwyczaj dźwięku i filmów) w czasie rzeczywistym
- XMPP
- HTTP Comet
- Współczesne przeglądarki implemetują WebSocket

Prezentacja danych

- Dla ludzkiego użytkownika sensor udostępnia stronę HTML

Web of Things - Resource Temperature



Home



Parent



Refresh



Atom Feed



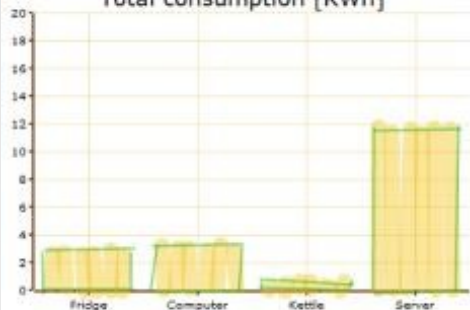
Get current
temperature: 24.0 C

Prezentacja: Mashup

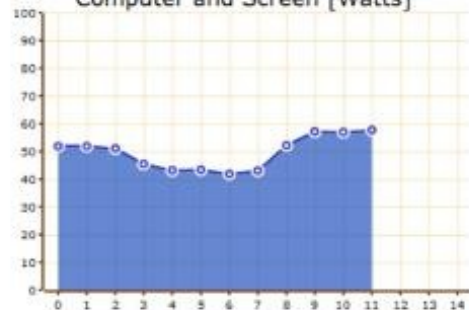
- „web mashup” - strona/aplikacja, która powstaje przez połączenie danych z wielu źródeł
- Wymaga odpowiedniego API od łączonych zasobów
- Łączenie danych z wielu źródeł po stronie klienta to ciekawy pomysł w kontekście systemów rozproszonych

Mashup „fizyczny”

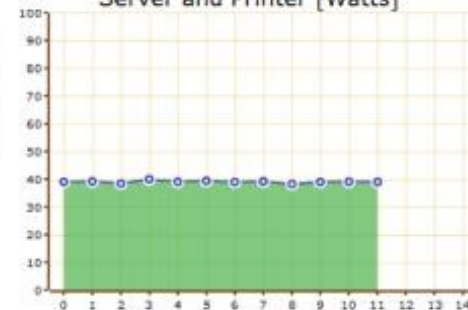
Total consumption [KWh]



Computer and Screen [Watts]



Server and Printer [Watts]

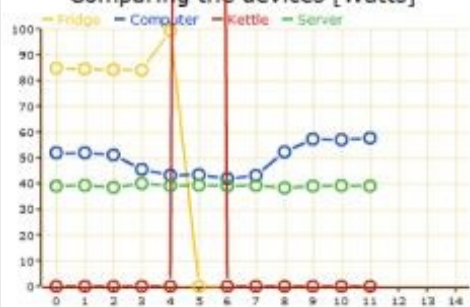


Last:
57.66 W
Max:
62.13 W

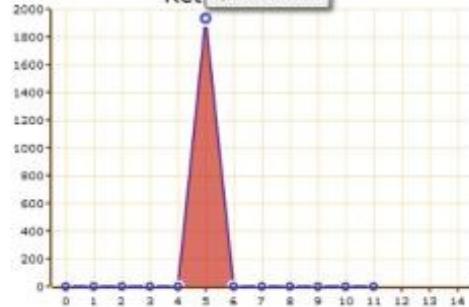


Last:
39.02 W
Max:
865.21 W

Comparing the devices [Watts]



Kettle [Watts]

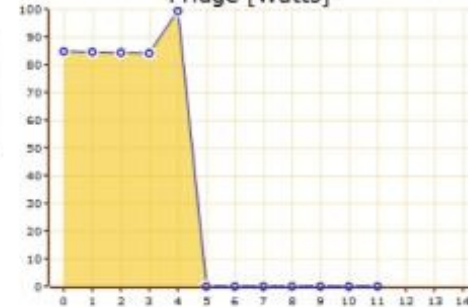


Last:
0 W
Max:
1990.94 W



Last:
0 W
Max:
288.92 W

Fridge [Watts]



Time of last observation
Tue Nov 25 11:50:05 GMT+100 2008



Did you know ?
A small air-conditioning system uses about 1900 watts, a good ventilator about 100 watts!



What's that thing ??
This shows the real time consumption in watts of the main devices running in the Cudrefin02.ch office.



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Yahoo Pipes

The screenshot displays the Yahoo Pipes interface for a pipe titled "Pizza Restaurants Near Earthquakes". The interface includes a sidebar with categories like Sources, User inputs, Operators, and My pipes. The main workspace shows a flow of operations: a Fetch step with a URL, a Location Extractor, a For Each: Replace step using Yahoo! Local, a Filter step with a rule, a Sort step, and a Pipe Output step. The bottom of the screen shows a debugger window with a list of 22 items.

Sources

- Yahoo! Search
- Yahoo! Local
- Fetch
- Google Base
- Flickr

User inputs

Operators

- Url
- String
- Date
- My pipes

Fetch

- URL
- <http://earthquake.usgs.gov/>

Location Extractor

For Each: Replace

Replace each item in input feed with output from

Yahoo! Local

Find within of

Filter

items that match of the following

Rules

Sort

Sort by

in order

Pipe Output

Debugger: Sort (22 items)

Time taken: 4.595497s [Refresh](#)

- ▶ **Brick Oven Pizza**
- ▶ **Brick Oven Pizza**
- ▶ **Round Table Pizza**
- ▶ **Godfather's Pizza**
- ▶ **Pizza Peddler**
- ▶ **Round Table Pizza**
- ▶ **Domino's Pizza**
- ▶ **Mario Pizza & Italian Rstrnt**

Użytkownik może sam oprogramować swój dom

The image displays the ClickScript IDE interface, titled "ClickScript - IDE, Web of Things Version." The interface includes a menu bar with "Execution View", "Console", "Options", "Library", "Tutorial", and "Exercise". Below the menu bar is a library of modules organized into several categories:

- cs.statement**: if, for-loop, for-each, url-button, sequence
- cs.default.primitive**: string, number, boolean
- cs.default.ide**: popup, timer, textfield, display
- cs.string**: concat, match, replace, repeat
- cs.math**: add, subtract, multiply, divide, random, diagram, is smaller, is greater, is equal
- cs.converter**: num2str, str2num
- cs.web.browse**: open, source-cod, fill-form, click, print
- cs.web.misc**: download
- cs.web.things**: switch, temperatur, light, mfdReader, acceleratio
- cs.SAP.bbody**: notification
- cs.collection**: pipe, add, has, remove
- cs.logic**: random, and, or, xor, not
- cs.robotic**: red-light, green-light, blue-light, light, heizung, lueber, temperatur, switch, auto, sound

At the bottom of the interface, there is an "Actions" bar with buttons for "Clean Up", "Run", "Repeated Run", and "Stop", along with a status indicator "runs: 0, status: [stop icon]".

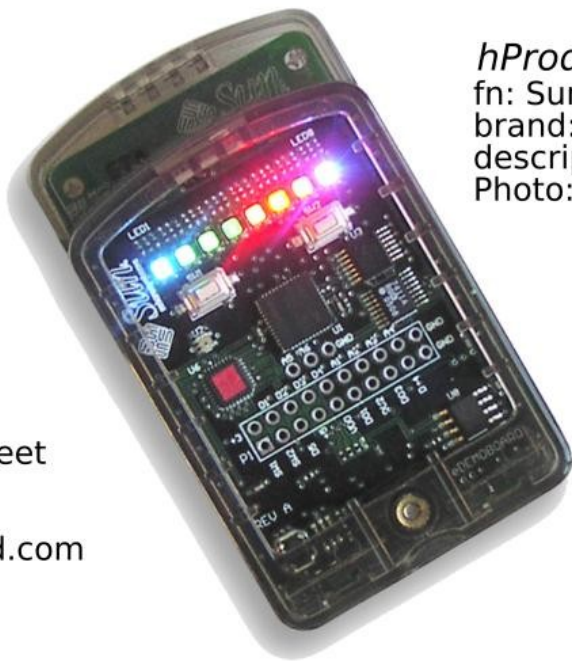
Below the library, a visual programming flowchart is shown. It consists of several interconnected blocks: a "wol.com" block, a "sensor1" block, a "36" block, a comparison block with a "<" symbol, and a "light" block. The flowchart illustrates a logic sequence where data from "wol.com" and "sensor1" is processed through a comparison block to control a "light" block.

Prezentacja: Wyszukiwanie urządzeń

- Wyszukiwanie fundamentem WWW
- Opis urządzeń - „semantyczny internet”
- Microformat (ale teraz w standardzie HTML5 jest Microdata)

hReview
summary: reliable,
long battery-life
rating: 5

hCard
address: ETH Street
region: Zurich
country: CH
url: sunspotworld.com



hProduct
fn: Sun SPOT
brand: Sun Labs
description: [...]
Photo: [...]

Geo
lat: 47.37821
long: 8.54953

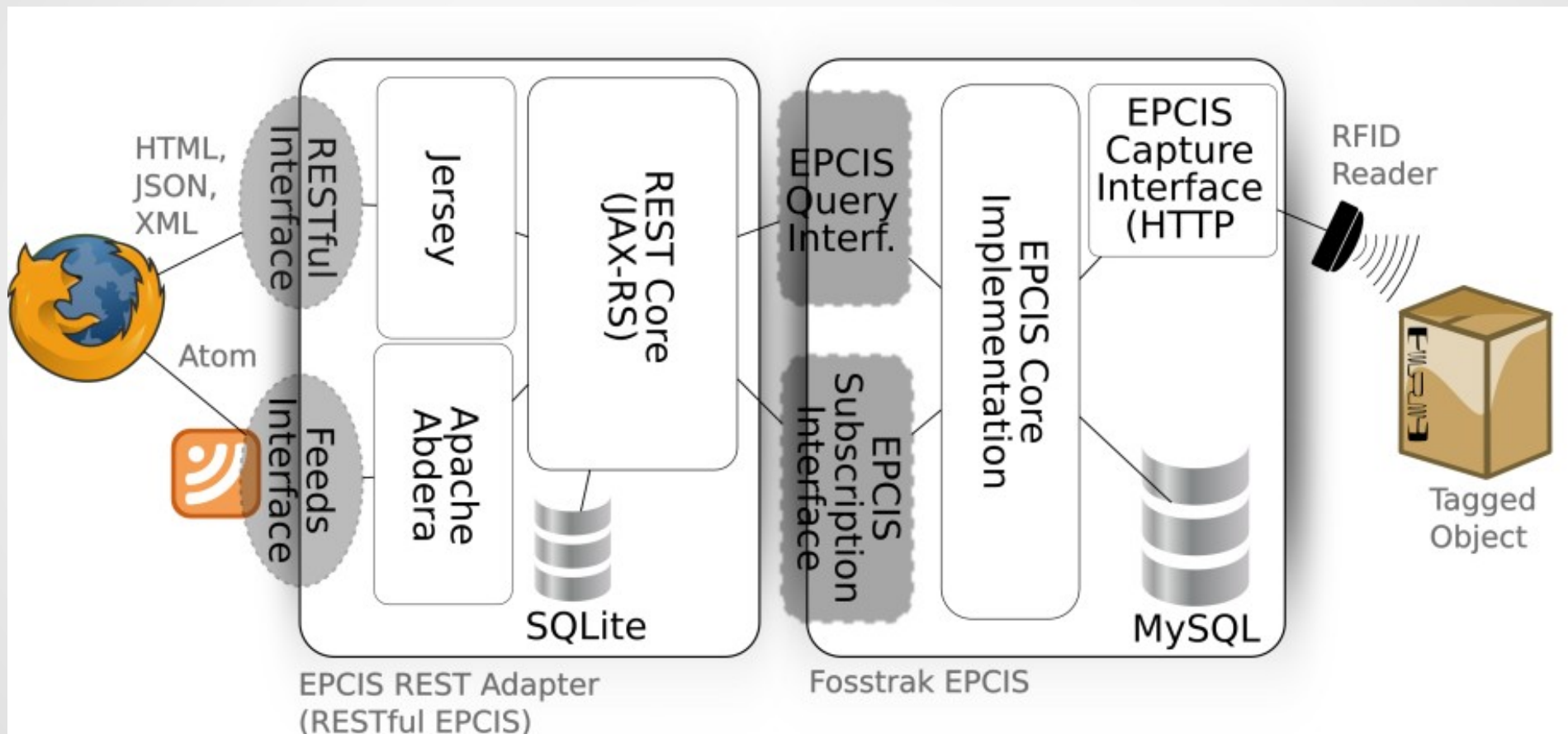
...

```
<span class="fn">Sun SPOT</span>  
<span class="URL">  
<a href="http://sunspotworld.com">  
</span>
```

Identyfikacja produktów

- W sieci rzeczy nie tylko urządzenia elektroniczne
- EPC – Electronic Product Code
 - Standard identyfikacji produktów oparty o RFID
- EPCIS – EPC Information Service
 - Udostępnia informacje o produktach przez web service
- Chcemy np. wyświetlić informacje o produkcie po zeskanowaniu RFID
- Tworzymy warstwę pośrednią udostępniającą produkty (i związane z nimi zdarzenia) jako zasoby sieci rzeczy

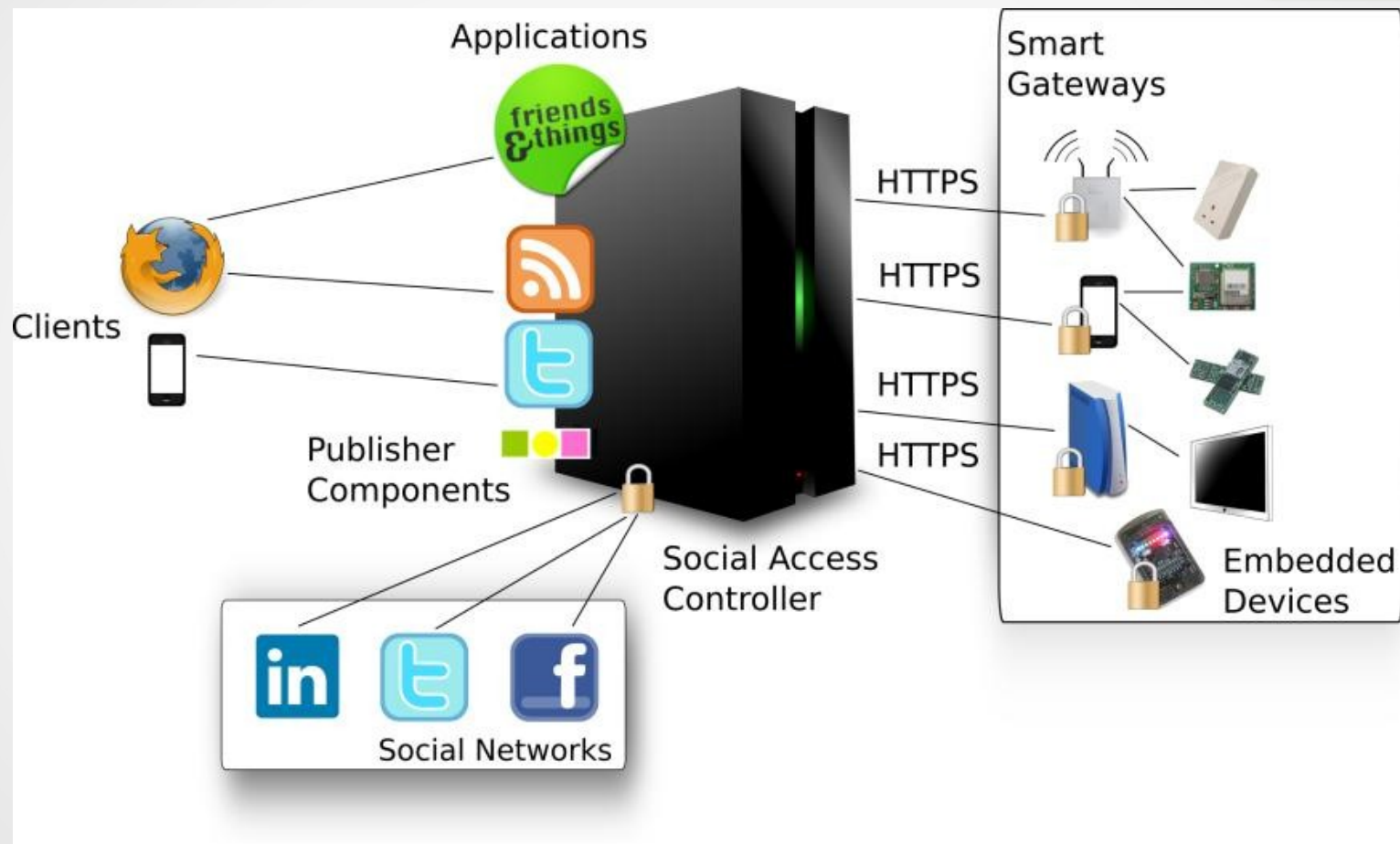
EPCIS zgodny z REST



Integracja z innymi usługami

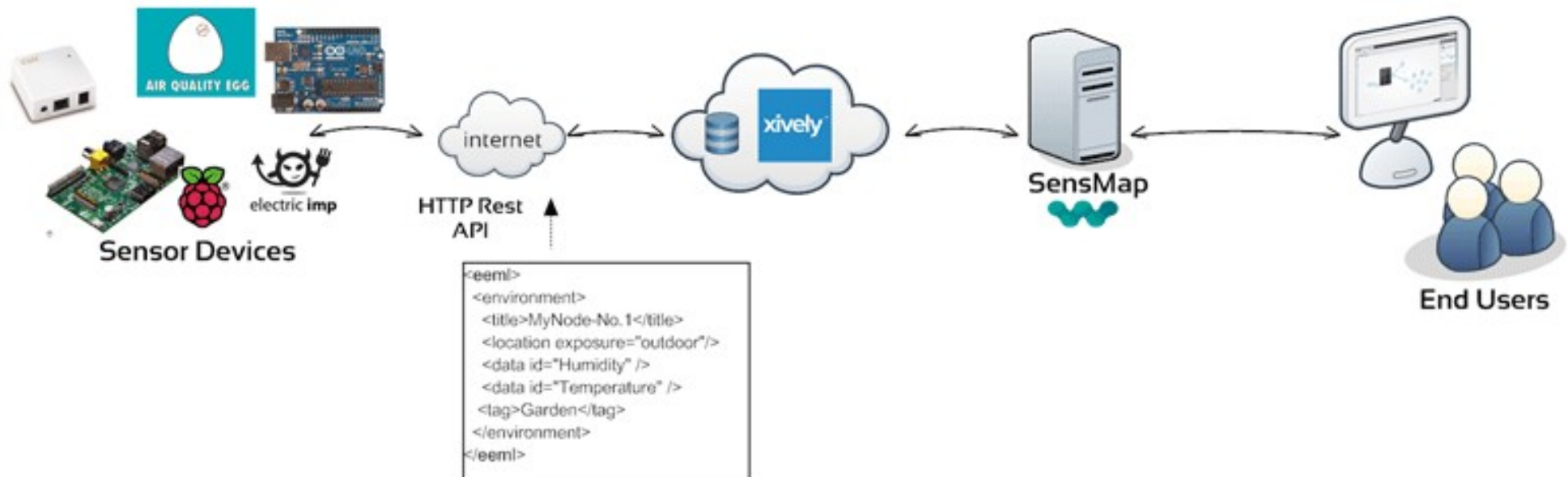
- Użytkownicy chcą udostępnić innym dane ze swoich sensorów
- Potrzebny pośrednik do kontroli dostępu
- Korzystamy z istniejącej infrastruktury:
 - Udostępnianie innym
 - Kontrola dostępu

Social Access Controller



Xively (Pachube)

- Zcentralizowany model sieci rzeczy



Bezpieczeństwo urządzeń

- Tajność, integralność, dostępność
- Przykre skutki włamania
- Radio i dostęp fizyczny
- Niewielkie zasoby
 - Zazwyczaj urządzenia zbyt słabe by szyfrować asymetrycznie
 - Atak przez brak snu
- Autentykacja
 - Może nie być centralnego serwera
 - Brak UI i wpisywania hasła
- Wskrzeszanie kaczki
- Przesyłanie klucza podczas fizycznego kontaktu
- Komunikacja: IPsec, TLS

Podsumowanie

- Wykorzystanie istniejących technologii
- Łatwość tworzenia aplikacji
- Odporność na awarie maszyn (niektórych)
- Hierarchiczna architektura
- Problemy z bezpieczeństwem

Bibliografia

- **From the Internet of Things to the Web of Things: Resource Oriented Architecture and Best Practices**
Dominique Guinard, Vlad Trifa, Friedemann Mattern, Erik Wilde
- **Interconnecting Smart Objects with IP**
Jean-Philippe Vasseur, Adam Dunkels



Dziękuję za uwagę