

System sterowania oświetleniem - PoC dla standardu ETSI SmartM2M



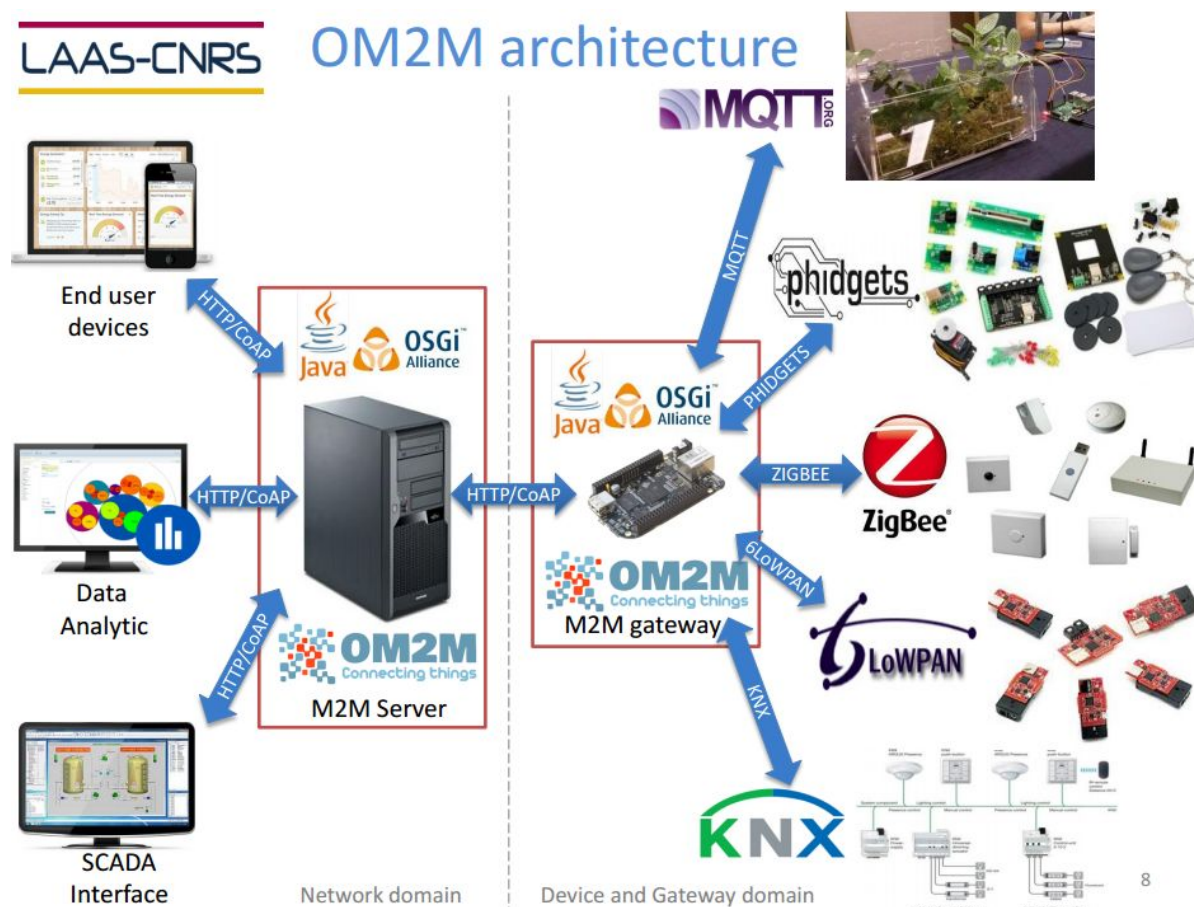
**Technologie Internetu Rzeczy
WIEiT Informatyka 2016**

Autorzy:
Marcin Górka
Piotr Kruczek
Łukasz Wawrzyk

1. Cel i opis projektu

Projekt ma na celu uruchomienie i przetestowanie przykładowego PoC dla standardu SmartM2M na podstawie jego implementacji Eclipse OM2M. W projekcie posłużono się przykładową aplikacją znajdującą się w pakiecie Eclipse OM2M, którą zmodyfikowano, w celu lepszego poznania środowiska programistycznego i architektury SmartM2M. System ten służy do zarządzania oświetleniem. Pozwala on na czasowe włączanie światła (np. za pomocą czujnika ruchu) oraz gromadzenie statystyk na temat ilości włączeń lampy w danym pokoju (co pozwala np. stwierdzić, które pokoje są najczęściej odwiedzane). Dostęp do tych danych jest możliwy ze zdalnego serwera, dzięki czemu można przeprowadzić ogólną analizę. Dokładny opis architektury zostanie przedstawiony w kolejnym punkcie.

2. Architektura systemu



Architektura OM2M składa się z dwóch podstawowych warstw:

- Server
- Gateway

2.1 Server

Server jest pośrednikiem między aplikacjami klienckimi, a gateway'em. Jeden serwer może komunikować się z wieloma gateway'ami. Komunikacja może odbywać się za pomocą protokołu HTTP lub CoAP, zarówno z gateway'ami jak i aplikacjami klienckimi. Serwer w OM2M zawiera specjalny interfejs webowy, który pozwala na podgląd połączonych z nim gateway'ów, a także na dostęp do danych, które dostępne są na danym gateway'u.

2.2 Gateway

Gateway jest szkieletem aplikacji OM2M. W jego skład wchodzi usługi serwowane za pomocą REST API. Usługi te można podzielić na dwa interfejsy: *north* i *south*. Interfejs *north* wystawia API do zarządzania urządzeniami, pozwala wysyłanie notyfikacji zawierających pomiary sensorów, pomaga w wyszukiwaniu urządzeń. Interfejs *south* zawiera w sobie dwie drogi komunikacji: *proxy-in* (pobieranie danych z czujników), *proxy-out* (wysyłanie danych do urządzeń).

Gateway może komunikować się z urządzeniami za pomocą różnych protokołów przystosowanych do wymagań IoT, np. MQTT, 6LoWPAN, ZIGBEE

3. Funkcjonalność systemu

System ma za zadanie pomóc w sterowaniu automatycznymi światłami. Światła uruchamiane są za pomocą odpowiedniego aktywatora (np. czujnika ruchu) na pewien określony czas. Dodatkowo możliwe jest pobranie ilości uruchomień danej lampy - może to pozwolić na stworzenie statystyk, np. określenie jaka ulica jest najbardziej uczęszczana w porze nocnej. Wszystkie te dane można pobrać z poziomu serwera.

Serwer może być uruchomiony na maszynie umieszczonej w budynku administracyjnym miasta, natomiast gateway'e mogą znajdować się w poszczególnych dzielnicach. Urządzeniem obsługującym aplikację gateway'a może być na przykład Raspberry PI.

4. Uruchomienie systemu

W celu uruchomienia projektu należy zbudować go za pomocą narzędzia Maven, wydając polecenie w katalogu *org.eclipse.om2m*:

```
$ mvn clean install
```

Po zbudowaniu należy najpierw uruchomić serwer, aby to zrobić należy przejść do katalogu z modułem serwera, nadać prawa uruchamiania dla skryptu *start.sh* a następnie go uruchomić.

```
$ cd org.eclipse.om2m/org.eclipse.om2m.site.in-cse/target/
products/in-cse/linux/gtk/x86_64/

$ chmod u+x start.sh

$ ./start.sh
```

Podobnie sekwencję poleceń należy wykonać, aby uruchomić gateway (różni się jedynie katalog modułu).

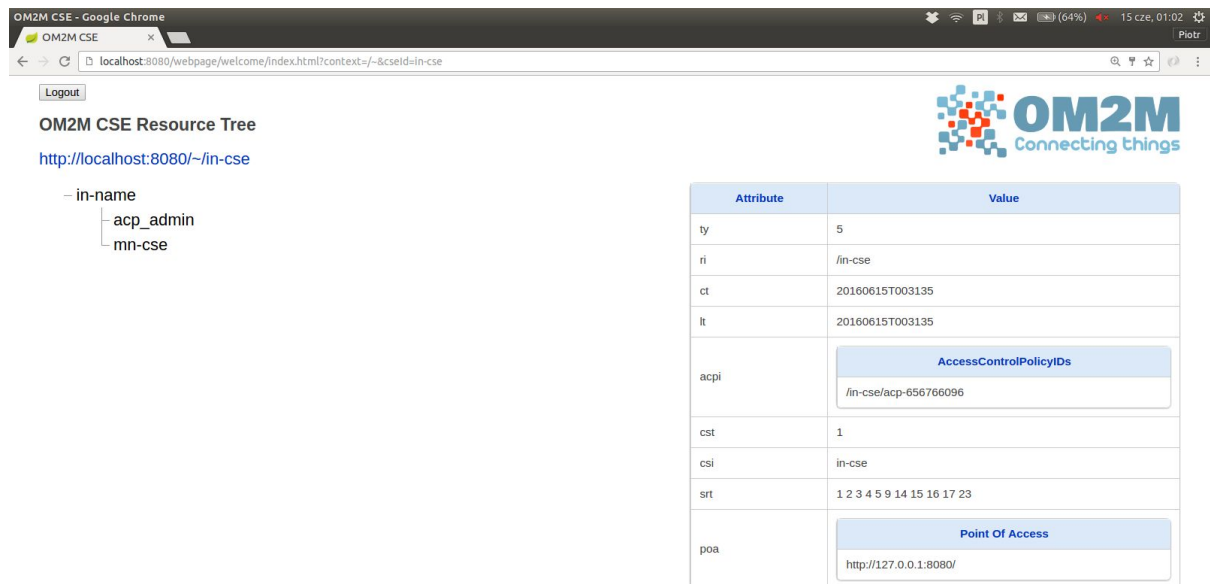
```
$ cd org.eclipse.om2m/org.eclipse.om2m.site.mn-cse/target/
products/mn-cse/linux/gtk/x86_64/

$ chmod u+x start.sh

$ ./start.sh
```

Dodatkowo należy uruchomić moduł sterowania oświetleniem w gateway'u. Listę dostępnych modułów można wylistować poleceniem `ss`, natomiast moduł zostanie uruchomiony po wykonaniu komendy `start ID`, gdzie `ID` to numer modułu.

Serwer udostępnia webowy interfejs, w którym możemy zobaczyć połączone gateway'e. Interfejs ten dostępny jest pod domyślnym adresem <http://localhost:8080/webpage>



Attribute	Value
ty	5
ri	/in-cse
ct	20160615T003135
lt	20160615T003135
acpi	<div>AccessControlPolicyIDs</div> <div>/in-cse/acp-656766096</div>
cst	1
csi	in-cse
srt	1 2 3 4 5 9 14 15 16 17 23
poa	<div>Point Of Access</div> <div>http://127.0.0.1:8080/</div>

4. Obsługa systemu

Po uruchomieniu modułu MN-CSE (Gateway) zostanie wyświetlone proste GUI symulujące obsługę świateł, a także przedstawiające aktualny ich stan. Dodatkowo pod każdym włącznikiem możemy odczytać ilość uruchomień danej lampy.



Aplikacja kliencka może odpytać serwer o aktualny stan danej lampy wywołując zapytanie na serwerze:

POST

```
http://localhost:8080/~mn-cse/mn-name/LAMP_0_2?op=getStateDirect  
&lampid=LAMP_0_2
```

Headers:

```
X-M2M-Origin:user:pass
```

W odpowiedzi otrzyma aktualny stan lampy:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<obj>
  <str val="LAMP" name="type"/>
  <str val="Home" name="location"/>
  <str val="LAMP_0_2" name="lampId"/>
  <bool val="false" name="state"/>
  <int val="13" name="usage"/>
</obj>
```

Pole *state* wskazuje na aktualny stan lampy (zaświecona - *true*, wyłączona - *false*).

Pole *usage* zawiera ilość uruchomień danej lampy.`

5. Bibliografia

Użyte materiały:

- <http://www.eclipse.org/om2m/>
- <http://www.eurecom.fr/fr/publication/4318/download/cm-publi-4318.pdf>
- https://www.iot-lab.info/wp-content/uploads/2014/09/LAAS_OM2M-IOT-LAB.pdf