MiMurcia
Murcia Smart City Project
Antonio Skarmeta
Univ. Murcia/OdinS
Jose Guillen and Jose Marquez
Ayto. Murcia
Smart Murcia: MiMurcia

7th city of Spain

Previous experiences on smart initiatives

Energy efficiency and sustainable mobility

Citizen participation

Murcia's size, geographic dispersion and previous experiences makes it the perfect "incubator" for pilot project deployments in the area of innovation and ICTs

Murcia project funded by Red.ES in the II Call Smart Cities Project. 8 Meuros for a 30 month project

MURCIA, suitable scenario for Smart Cities
ONE PERSON
ONE CITY COUNCIL

- The city council in search for the citizen
- Smart city council looks for the citizen
Vision: MiMurcia

One person: one townhall

A town hall that has a personal differentiated and unique relation with each citizen
Proposal objectives

- Use of the most appropriate channel
- Information:
  - Cultural, Feasts, environmental information.
  - Customized, geo-localized, useful, required and contextualized
- Transparency and clarity
- Participatory democracy mechanism
- Integration into Open Data initiative
- Unified SDI-GIS
- Business attraction
- Data Marketplace and innovation support
- Innovative paperless administration.
- Close to the citizen
- Administration modernization
- Reduce the documentation Administration 3.0
- Smart urban mobility: State of the city, public transport, waste collection
- Urban quality: Energy efficiency, reduction of the use of own vehicle, p&g management
Architecture

MiMurcia

Communicate
Solve
Open
Sustainable

SMART	CITY	PLATFORM

Living
Enjoy
Municipal

Murcia
Murcia
Services

Sensorization

CRM (PROACTIVE)
Bussines
Call
Web
Social
Apps
Extranet

Offices
Center
Portals
networks

Urban
Urban
Parks
Lighting

mobility
quality
systems

...}

CEUS
(System Intelligence)

Business Intelligence (BI)

Data sources

MiMurcia
Smart City Platform

- Integration and interoperability layer
- Analysis and storage layer
- Advanced services layer
- Balanced scorecard
- Access identification and authorization layer
- Configuration, management and monitoring layer
- Data publishing layer (OpenData)
Data sources

- Regulated Parking Service and private car parks
- Mix-modal public transport and the use of bicycle
  - Citizen Card
- Traffic management
- Lighting system
- Watering systems for parks and gardens
- Noisy zones detection
- Waste collection
- Incidences of citizens
- Commerce promoting in the centre of the city (iBeacons)
Main Objective

- Integrate data from sensors, open data sources and internal database by means of common data model (NSGI)
- Provide facilities for orchestrating new services based on connecting different municipality areas of information
- Create new channels of communication with citizens based on social networks contextualized information
- Increase the efficiency of services and reaction time based on the real time information of the city
Interoperability

- Create end-point for integration existing vertical
- Identify communication options to give better coverage
- Define mechanism for supporting data exchange
- Municipality agreement for requesting any further tender involving ICT components to be compatible with the Smart City platform and provide NGSI interfaces for interoperability:
  - New tenders on traffic management, public parking, garden and parks maintenance
Integration of existing vertical

OPEN DATA PLATFORM

Information Broker

SMART CITY PLATFORM

DB

Sensorized values from park sensors

Actuation on watering sensors
Smart Irrigation of Garden and Parks
SCADA Integration
Street Lighting
WIFI coverage

MiMurcia
LoRA Connectivity

MiMurcia
Sonometers for Noisy Area

MiMurcia

WEB SERVICES

SONÓMETROS

SISTEMA DE COMUNICACIONES

SISTEMA DE MONITORIZACIÓN DE RUIDO

PLATAFORMA SMART CITY DE MURCIA

EQUIPAMIENTO DE MONITORIZACIÓN

INTERNET

Red móvil

m2m

Repositorio de Datos

Gestor de comunicaciones

Plataforma web de Monitorización de Ruido

CABON ITM DE MURCIA

MiMurcia
Informative Panels
Visiting places and activities promoting in the city centre

- iBeacons
- Interact with your mobile phone!
- Send push notifications!
- Advertisements!
- Promotions!
- Offers!
- Discounts!
- News!
- Deployment!

MiMurcia
Citizen profiling

- Using location based information (cellular or beacons, etc).
  - Schedule and orchestrate a strategy for:
    - Derivate traffic improving quality of living
    - Incentivate and promote public transport
    - Balance the city council resources to assure security, confortability, and a great variety of services to tourist
    - Avoid overcrowding and provide mechanisms to handle it.
  - Identify tourist flows
  - Special dates movements and how affect city
    - Christmas
    - Summer holidays
    - Easter, ...
FIWARE platform deployment for Smart City

- Heterogeneous information
- Different nature of sources of information
- Set up and develop different ways for integration
  - Using enablers such as COMET and CYGNUS
  - Developing new connectors to integrate the information
Integrated services

- Incidences
- Temperature of town hall buildings
- Energy consumption of buildings
- Traffic measurements
- Parking slots of parking sites
- Free parking slots of public rental bike service
- Tramp
- Bus stops and vehicle locations
- Rainfall
- Solar panels
Service Map
Quater View
Platform deployment

Enablers:

- ORION Context Broker
- COMET Short Therm Historic
- CKAN Open Data
- CYGNUS
Integration examples – Urban bus

- They provide an API using SOAP
  - We have to develop a Python-based connector to extract the information and integrate it into our platform
  - Using suds – Lightweight SOAP client
- Organization of the information
  - fiware-service: autobuses
  - fiware-servicepath: /murcia
Bus-stops are represented as points. They contain:
- Location
- Bus lines in each stops
  - Id
  - direction
Integration examples – Urban bus

class API:
    def __init__(self):
        self.client = Client(config['wsdl'])

    def LinesDiscovery(self):
        request = self.client.factory.create('ns2:WsLinesDiscoveryStructure')
        request.Request.AccountId = config['username']
        request.Request.AccountKey = config['password']
        request.Request.RequestTimestamp = now()
        return self.client.service.LinesDiscovery(request)

    def GetStopMonitoring(self, monitoringRef):
        request = self.client.factory.create('ns2:StopMonitoringRequestStructure')
        request.ServiceRequestInfo.AccountId = config['username']
        request.ServiceRequestInfo.AccountKey = config['password']
        request.Request.RequestTimestamp = request.ServiceRequestInfo.RequestTimestamp = now()
        request.Request.MonitoringRef = str(monitoringRef)
        return self.client.service.GetStopMonitoring(request)

Our Python module connects to the remote resource *ns2:WsLinesDiscoveryStructure* to get the information of bus lines.

*ns2:StopMonitoringRequestStructure* for bus stops.
Integration examples – Urban bus

```python
import httpplib
import json

from HTMLParser import HTMLParser
h = HTMLParser()

class NGSIContextBroker:
    def __init__(self, url, fiwareService, fiwarePath):
        self.url = url
        self.headers = { 'Content-Type': 'application/json; charset=utf-8',
                         'Accept': 'application/json',
                         'fiware-service': fiwareService,
                         'fiware-servicepath': fiwarePath}

    def updateContextByContextElements(self, contextElements):
        conn = httpplib.HTTPConnection(self.url)
        params = json.dumps({'contextElements': contextElements, 
                              "updateAction": "UPDATE" }, ensure_ascii=True)
        conn.request("POST", "/v1/updateContext", params, self.headers)
        response = conn.getresponse()
        print response.status, response.reason
        data = response.read()
        print data
        conn.close()
```
Integration examples – Bike rental service

- They provide a REST API
  - We developed nodejs connector to extract the information and integrate it into our platform

- Organization of the information
  - fiware-service: bicis
  - fiware-servicepath: /murcia
Integration examples – Bike rental

Representation of bike parking slots:
- Id: BikeParkingSite:*  
- Free slots  
- Occupied slots  
- Enabled  
- Description  
- Location

```javascript
{
  type: "Sensor",
  isPattern: "false",
  id: "AparcamientoBicis:" + obj[i].id_aparcamiento,
  attributes: [{
    name: "libres",
    isPattern: "false",
    id: "AparcamientoBicis:" + obj[i].id_aparcamiento,
    attributes: [{
      name: "libres",
      type: "number",
      value: obj[i].libres+""
    },
    { 
      name: "ocupados",
      type: "number",
      value: obj[i].ocupados
    },
    { 
      name: "habilitado",
      type: "number",
      value: obj[i].habilitado
    },
    { 
      name: "descripcion",
      type: "string",
      value: "encodeURIComponent(obj[i].descripcion.trim())" 
    },
    { 
      name: "geoposicion",
      type: "coords",
      value: obj[i].latitude+"",+obj[i].longitude,
      metadatas: [{
        name: "location",
        type: "string",
        value: "WGS84"
      }] 
    }
  }
}
```
Integration examples – Bike rental service

```javascript
var req = http.request({
  method: "post",
  path: "/v1/updateContext",
  host: hostAddr,
  port: 1026,
  headers: {
    "Content-Type": "application/json",
    "Content-Length": strjson.length,
    "Accept": "application/json",
    "fiware-service": service,
    "fiware-servicepath": servicePath
  }
}, function(response) {
  // TODO: Comprobar que es OK
  response.on("data", function(dat){
    // console.log(dat.toString());
  });

  console.log("Petición OK");
});
```
Integration examples – Tramp service

- Two different services:
  - Information in tramp stops
  - Information and location of tramp vehicles
  - We developed a nodejs conector

- Organization of the information
  - fiware-service: tranvia
  - fiware-servicepath: /murcia
Integration examples – Tramp service

Tramp stops:
- Id TrampStop.
- Location
  - State: info of both directions

Tramp vehicle:
- Id Tramp.
- Location
Integration examples – Tramp service

```javascript
var strjson = JSON.stringify({
    contextElements: context,
    updateAction: "UPDATE"
});

var req = http.request({
    method: "post",
    path: "/v1/updateContext",
    host: hostAddr,
    port: 1026,
    headers: {
        "Content-Type": "application/json",
        "Content-Length": strjson.length,
        "Accept": "application/json",
        "fiware-service": service,
        "fiware-servicepath": servicePath
    }
}, function(res) {
});
```

Updating context to our FIWARE platform
Comet Integration

We need to configure it:

```javascript
config.server = {
    // The host where the STH server will be started.
    // Default value: "localhost".
    host: 'fiware-dev.inf.um.es',
    // The port where the STH server will be listening.
    // Default value: "8666".
    port: '8666',
    // The service to be used if not sent by the Orion Context Broker in the notifications.
    // Default value: "testservice".
    defaultService: '',
    // The service path to be used if not sent by the Orion Context Broker in the notifications.
    // Default value: "/testservicepath".
    defaultServicePath: '/','
    // A flag indicating if the empty results should be removed from the response.
    // Default value: "true".
    filterOutEmpty: 'true',
    // An array of resolutions the STH component should aggregate values for.
    // Valid resolution values are: 'month', 'day', 'hour', 'minute' and 'second'
    aggregationBy: ['day', 'hour', 'minute'],
    // Directory where temporary files will be stored, such as the ones generated when CSV files are requested.
    // Default value: "temp"
    temporalDir: 'temp'
};
```

A manual subscription was also used to start the service.
Comet/Cygnus Integration - Subscription

curl localhost:1026/v1/subscribeContext -s -S --header 'Content-Type: application/json' \ --header 'Fiware-service: tranvia' --header 'Fiware-servicepath: /murcia' --header 'Accept: application/json' -d | python -m json.tool) <<EOF

"entities": [
  {
    "type": "Vehiculo",
    "isPattern": "true",
    "id": "Tranvia:*"
  }
],
"attributes": [
  "geoposicion"
],
"reference": "http://sth-host:port/notify",
"duration": "P1M",
"notifyConditions": [
  {
    "type": "ONCHANGE",
    "condValues": [
      "geoposicion"
    ]
  }
],
"throttling": "PT5S"

EOF

Details about subscription

End point of subscriber
Security components

- Enablers
  - KeyRock: Id Management

- New components
  - Capability Manager: Authorization
  - PEP_PROXY: Authorization enforcement and data encryption using CP-ABE
KeyRock is the component responsible for verifying user credentials providing authentication verdict.
Authorization

Capability Manager is accessed via POST API REST
Authorization – Capability Token

```
{
  "id": "eg3fq:fb5r23tra3",
  "ii": 1485172121,
  "is": "issuer@odins.es",
  "su": "zNwS5FetB4rwzSKsWwSBAxm5wDa=JgLjHU8zSnmeSFQgSG9HhdsJrE8=",
  "de": "coap://sensortemp.floor1.computersciencefaculty.um.es",
  "si": "SbUudG4zuXswFBxDeHB87N6t9hR=PBQqCN3gpu7nSkuPzDk7kaR3dq1=",&
  "ar": [
    {
      "ac": "queryContext",
      "re": "temperature"
    }
  ],
  "nb": 1485172121,
  "na": 1485174121
}
```
Authentication and authorization

Interactions with FIWARE enriched with security functionality
CP-ABE Encryption integration

"contextElements": [
    {
        "type": "Test",
        "isPattern": "false",
        "id": "Test:1",
        "attributes": [
            {
                "name": "cipheredAttribute",
                "type": "cyphertext",
                "value": "hello",
                "metadatas": [
                    {
                        "name": "cpabe-policy",
                        "type": "string",
                        "value": "floor1 and admin"
                    }
                ]
            }
        ]
    }
]

PEP_Proxy will use the highlighted information and encrypt the cipheredAttribute with the corresponding CP-ABE policy floor1 and admin
CP-ABE Encryption performance
MiMurcia Open Innovation Smart City Lab (MiOS):

- IoT-based living lab provided by MiOS with several sensors deployed over the city
- Promote and improve the business innovation using data provided by the smart city platform
- Offer possibility to define new services/apps based on the data available of the city behaviour
- Create open APIs and foster meetup and co-creation workshops
How we achieve it

Governance

- Smart Murcia Office, Centro Único de seguimiento (CEUS)
- Management
- Executive Committee
- Smart Murcia Work Group
- Citizen’s participation roundtables
- Red.es
- Smart Murcia Advisory Committee
CEUS: The intelligence of the project

- There is a huge amount of information provided by different devices and sensors along the city
- An smart brain is needed to process this information
  - Analysis of the whole information
  - Decision making
  - Action plans elaboration

- CEUS
  - The intelligence of the city, coordinating actuations and areas of the city council
  - Training, information and interaction point with the citizen
  - A demonstration and support place for the citizen
Conclusions

- Great complexity in City with new and legacy solutions
- We have integrated heterogeneous information into our FIWARE PoC platform.
- Important to provided security and privacy
- Most important → create a team