RIPSAC - Objectives

- Make it easy to develop applications that use
  - sensors that can be reached using web protocols
  - Sensor available on smart phones
- Unlock sensor data from application silos
- Provide a scalable and secure cloud based platform for these apps
- Provide a scalable platform for analytics can be performed
RIPSAC - A PaaS platform for IoT Applications

**Overview**
- Java PaaS for Sensor Web & Mobile Crowd Sensing Apps
- Provides OGC SWE standards

**RIPSAC PaaS Cloud Use Case**

**SDK and APIs provided in RIPSAC**

**For Platform Provider**
- Core IoT Services
- Identity, Security, Privacy
- End User License Mgmt
- Ad Delivery
- Multi-tenancy
- Sandboxes
- Operation Support Systems

**For Sensor Provider**
- Feature, Phenomena & Sensor description
- Define feeds & sensor streams
- Publish & share sensor streams
- Define access control and privacy preferences

**For Application Developer**
- Dev & Test Sandboxes
- SDK & APIs
- Test Data
- Publish Apps
- Define EULAs
- Manage App Life Cycle

**For End User**
- Download Apps
- Subscribe / Unsubscribe Services
- Control Privacy Settings
- View usage history, billing etc.
How Sensors connect with RIPSAC

Internet

RIPSAC platform services

Web Services interfaces exposed by RIPSAC Platform services

Web Services calls over Wireless / cellular networks / m2m networks

Sensors connected to Internet via gateways or handheld devices

Sensors directly connected to internet

Gateway Devices

handhelds

Sensor Networks

Sensors directly connected to handheld device

TATA CONSULTANCY SERVICES
Experience certainty.
RIPSAC provides the following services and APIs
- Multi-tenant platform with secure virtual environment for each tenant
- Java and JavaScript APIs
- Sensor Services
  - Sensor & sensor observation description, metadata & discovery
  - Sensor observation recording and query
  - Support for geo-spatial and spatio-temporal queries
  - Can support any type of sensor & sensor observations
- Database & Storage
  - Relational and Document database services
  - Load balanced and scale-out services
- Analytics & Visualization Services
  - Auto-scaling and load balanced ‘R’ server farms
  - Message driven batch analytics framework
  - JavaScript APIs for visualizing and exploring Sensor Data
RIPSAC Applications

Sensors, Gateway Devices, Handhelds (Source of sensor observations)

End User Application outputs
Integration with Infrastructure Cloud Services

- RIPSAC Applications
- RIPSAC Platform Services
- Auto Scaling & Load Balancing Services
- Infrastructure Cloud Services

Virtual Machines: VM, VM, VM

API calls: Infrastructure Cloud Services API calls
Components in RIPSAC

Sensor Application @ Edge Gateway

Application @ Backend Gateway

Consumer Application
- Visualization
- Control Logic
- Analytics

- RDBMS for App
- MongoDB
- ObjectStore

- Portal
- App Life Cycle Mgt
- DB Mgt
- Analytics Dashboard
- Service Dashboard

- Sensor Observation Service
- Maven Archetype
- SOS RDBMS
- Sensor discovery

- SOS Authentication
- Java SOS Library
- JavaScript SOS Library

- Multi Tenancy
- Load Balancer
- Task Queue

- R
- R

- Visualization Library

On Openstack based TCS Cloud
On Amazon ASW Cloud
## RIPSAC Features

### Feature List

- OGC Sensor Observation Service (SOS)
- Sensor catalog service / registry service
- Time Series Database
- Sensor Simulator
- Template Project/Code Generator
- Authentication & Logging for SOS
- APIs
- Java Client libraries
- JavaScript client libraries
- Relational Database Service
- Document Database (NoSQL) service
- Play Framework support
- Task Queues

### Feature List

- R based Scalable Analytics Service
- Batch jobs framework
- Tenant Management
- Multi-tenant app deployment support
- Application life cycle management
- Service Dashboard
- OpenStack infrastructure cloud integration
- JavaSacript Sensor Data Visualization libraries
- Horizontal scalability of sensor and analytics services
- Runs on OpenStack based TCS Cloud
- Runs on Amazon AWS cloud
Example RIPSAC Applications
Smart Transportation Solution

Bus Tracking System
- Location (GPS), Speed, Accelerometer, Passenger IDs
- Valid passenger lists, Route Info

Parking Management
- Images / video feeds of entry & exit ramps of underground parking lots

RIPSAC + Apps
- Sensor Interface Services
- Storage & Database Services
- Analytics Services

Reports & Alerts

Cab & Passenger Tracking
- Images of cab registration number, Passenger IDs
Adaptive Wind Forecasting
- Boost forecast accuracy of a given predictor
- Adaptively combine multiple predictors for better accuracy

Benefits
- Better asset utilization
- Real-time competitive advantage in energy trading markets
- Minimize imbalance charges

Features
- Adaptive forecast
- Program maintenance
- Data security
- 24x7 service
- SLA compliance
- Reporting
Intelligent Healthcare – Remote Medical Consultation

Health Center / Home

- Body Fat Analyzer
- Blood Pressure Monitor
- Pulse OxyMeter

Mobile gateway

433MHz Wireless gateway

Web Request

RIPSAC

Patient Records

Healthcare Portal

Expert Doctor
RIPSAC and Big Data
Big Data Approaches in RIPSAC

- Scale out of SOS and Analytics Server
- Map Reduce on Mongo DB
- Export of SOS Data to HDFS
- Meta Data in SOS, Big Data in Hadoop HBase
Scale out of SOS and Analytics Server

Sensor Observation Service (SOS) Front-end

SOS Server  SOS Server  SOS Server

Load Balancer

Application

R  R  R

Analytics Server Farm
Map Reduce on Mongo DB

Sensor Observation Service

Map Reduce on Mongo

Mongo DB
Export of SOS Data to HDFS

Sensor Observation Service → SOS to HDFS Adapter → Map Reduce → HDFS
Sensor Services in RIPSAC
OGC Sensor Web Enablement

- Sensor Web Vision
  - Sensor Data Discovery
  - Sensor Data Access
    - Synchronous
    - Asynchronous
  - Control Sensor parameters

<table>
<thead>
<tr>
<th>Information Model</th>
<th>Interface Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>• SensorML</td>
<td>• SOS</td>
</tr>
<tr>
<td>• Observation &amp; Measurement (O &amp; M)</td>
<td>• SES</td>
</tr>
<tr>
<td></td>
<td>• SPS</td>
</tr>
</tbody>
</table>
• Describe Sensor or Sensor System
• Used to model Meta Data about sensors
• Can describe physical sensor, sensor system, soft-sensors – method, algorithm

<table>
<thead>
<tr>
<th>Sensor ID</th>
<th>URN preferred</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Location</td>
<td>Last known location - latitude, longitude, (optionally) altitude</td>
</tr>
<tr>
<td>Sensor Attributes</td>
<td>Is it Mobile?  Is it Active?</td>
</tr>
<tr>
<td>Input Phenomena to Sensor</td>
<td>Optional</td>
</tr>
<tr>
<td>Sensor Outputs</td>
<td>One or more fields along with their type</td>
</tr>
<tr>
<td>Offering</td>
<td>Sensor outputs are clustered into offering</td>
</tr>
<tr>
<td>Sensor Components</td>
<td>Optional</td>
</tr>
</tbody>
</table>
### Observation & Measurement

- **Generic schema for observations & measurements from heterogeneous sensors**

<table>
<thead>
<tr>
<th>Sampling Time</th>
<th>Observation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor ID</td>
<td>Sensor that produced this observation</td>
</tr>
<tr>
<td>Feature of Interest – ID, Location</td>
<td>Real-world object which is being observed</td>
</tr>
<tr>
<td>Observed Property</td>
<td>Sensor Outputs</td>
</tr>
</tbody>
</table>
| Result              | • Numeric, Text, Category, Time, Location, etc  
|                     | • Scalar, Record, Array of Record  
|                     | • Inline, Out-of-Band     |
Sensor services

**Sensor Observation Service (SOS)**
- Provides access to observations for heterogeneous sensor systems.
- Horizontal model since it applies to all domains that use sensors to collect data.
- Spatio-temporal query support

**Sensor Event Service (SES)**
- Provides publish subscribe based service on different events exchange between producer and consumer.
- Provides registration and publishing of event service.
- Provides subscribe service and subsequently notification of service.
- Provides subscribe/unsubscribe of choice able events through different filtration mechanism.
Sensor services

Sensor Planning Service
- Control sensors and provides operations for task management in Sensor.
- Checks feasibility, define, submit a task in sensor.
- Provides getStatus, update or modify, cancellation of an running task.
- Uses WNS to communicate Client in asynchronous manner.
- New task created via SPS would subsequently is a service of SOS.

Web Notification Service (WNS)
- Provides standard web service interface for asynchronous delivery of messages or alerts.
- The SES either sends the alert directly to the client, or makes use of the WNS in order to deliver the alert message.
<?xml version="1.0" encoding="UTF-8"?>
<GetObservation xmlns="http://www.opengis.net/sos/1.0"
    xmlns:ows="http://www.opengis.net/ows/1.1"
    xmlns:gml="http://www.opengis.net/gml"
    xmlns:ogc="http://www.opengis.net/ogc"
    xmlns:om="http://www.opengis.net/om/1.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/sos/1.0
    http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd"
    service="SOS" version="1.0.0" srsName="urn:ogc:def:crs:EPSG::4326">

<offering>GAUGE_HEIGHT</offering>
<observedProperty>urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel</observedProperty>
$responseFormat$text/xml;subtype="om/1.0.0"$
</GetObservation>
SOS : Spatio-Temporal Query

<offering>GAUGE_HEIGHT</offering>

<eventTime>
  <ogc:TM_During>
    <ogc:PropertyName>om:samplingTime</ogc:PropertyName>
  </ogc:TM_During>
</eventTime>


<observedProperty>urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel</observedProperty>

<featureOfInterest>
  <ogc:BBOX>
    <ogc:PropertyName>urn:ogc:data:location</ogc:PropertyName>
    <gml:Envelope srsName="urn:ogc:def:crs:EPSG::4326">
      <gml:lowerCorner>50.0 7.0</gml:lowerCorner>
      <gml:upperCorner>53.0 10.0</gml:upperCorner>
    </gml:Envelope>
  </ogc:BBOX>
</featureOfInterest>

[result]
  <ogc:PropertyIsGreaterThan>
    <ogc:PropertyName>urn:ogc:def:phenomenon:OGC:1.0.30:waterlevel</ogc:PropertyName>
    <ogc:Literal>5</ogc:Literal>
  </ogc:PropertyIsGreaterThan>
</result>
RIPSAC Demonstrator - Siruseri Smart Campus Solution

**Bus Tracking System**
- Location (GPS), Speed, Accelerometer, Passenger Ids
- Valid passenger lists, Route Info

**Parking Management**
- Images / video feeds of entry & exit ramps of underground parking lots

**RIPSAC + Apps**
- Sensor Interface Services
- Storage & Database Services
- Analytics Services

**iDigi M2m Cloud**

**Reports & Alerts**
- Images of cab registration number, Passenger Ids

**Cab & Passenger Tracking**
- Valid passenger lists, Route Info

**Valid passenger lists, Route Info**

**Images of cab registration number, Passenger Ids**
Bus Fleet – Sensor Observation & Measurement Schema

- **Bus ID**
  - featureOfInterest

- **Gateway Device ID**
  - on the Bus

- **BusMovement**
  - Observation
  - Sampling Time Instant

- **Bus Movement on Route**

- **Bus Movement Result**
  - Bus Location (Lat/Long)
  - Route ID
  - Journey Direction (InBound, OutBound)
  - Speed
  - Acceleration Record
## Related Work

<table>
<thead>
<tr>
<th>Project</th>
<th>Who &amp; When</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooltown</td>
<td>HP Labs, 2001</td>
<td>Web based ubiquitous computing</td>
</tr>
<tr>
<td>IrisNet</td>
<td>Intel Research &amp; CMU, 2003</td>
<td>World wide sensor web design</td>
</tr>
<tr>
<td>CarTel</td>
<td>MIT CSAIL, 2007</td>
<td>Vehicular cyber-physical system</td>
</tr>
<tr>
<td>CitySense</td>
<td>Harvard, 2007</td>
<td>Urban scale wireless sensor network testbed</td>
</tr>
<tr>
<td>WikiCity</td>
<td>MIT Senseable City Lab, 2007</td>
<td>Real-time urban dynamics (”Real Time Rome”)</td>
</tr>
<tr>
<td>Nericell</td>
<td>Microsoft Research, 2008</td>
<td>Road and Traffic Condition monitoring using Mobile Crowdsensing</td>
</tr>
<tr>
<td>UBI</td>
<td>Univ of Oulu, Finland, 2009</td>
<td>Open Urban Computing testbed</td>
</tr>
<tr>
<td>Sensor Web</td>
<td>52 North, Germany</td>
<td>OGC SWE reference implementation</td>
</tr>
<tr>
<td>Spitfire</td>
<td>EU Project</td>
<td>Architecture for semantic applications involving Internet connected sensors</td>
</tr>
</tbody>
</table>
Further Reading

4. **Participatory Sensing: Applications and Architecture**, Deborah Estrin, January/February 2010, *IEEE Internet Computing*
Thank You
Contact: prateep.misra@tcs.com