

Capturing Sensor Data From Mobile Phones Using GSN Middleware

Charith Perera (ANU-CSIRO), Arkady Zaslavsky (CSIRO), Peter Christen (ANU),
Ali Salehi (CSIRO), Dimitrios Georgakopoulos (CSIRO)

09 September 2012

SEMANTIC DATA MANAGEMENT / INFORMATION ENGINEERING LAB

www.csiro.au



Agenda

Background

The Problem

The Proposed Solution

Performance Advantage

Evaluation

Future Work

Background



Background

- **Mobile Phones getting more powerful**
 - Processing Power (Ex: 1.4Ghz dual core processors)
 - Memory (more than 1GB RAM)
 - Storage (around 64 GB)
 - Number of mobile (5.6 billion mobile phones)
 - Built-in sensors (more than 12 in Android + camera + microphone)
 - Becomes cheaper and smaller
- **What does it mean... ? Already deployed, mobile (moving), sensors and sinks with decent amount of processing capability that are regularly charged by humans...**

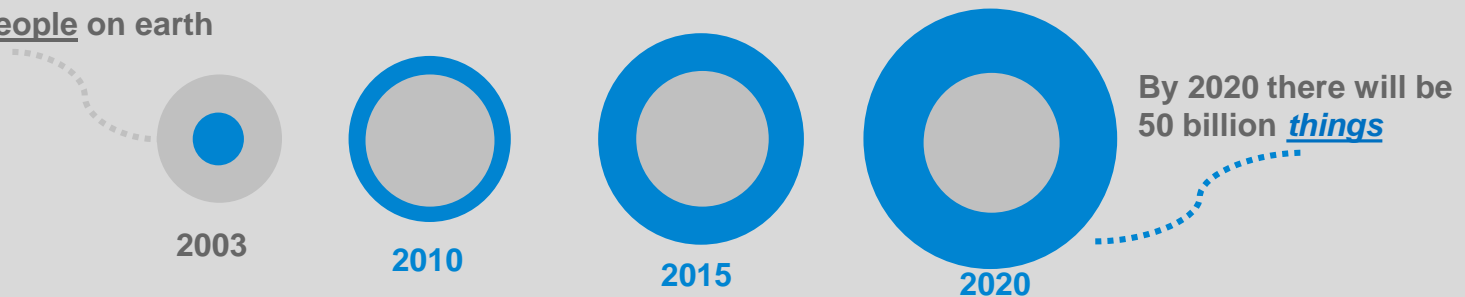
Background

- **Internet of Things**

- 20 billion things to be connected to internet by 2020
- Things = Sensors + actuators + processing/storage/communication
- More data to be collected and processed

“...The Internet of Things allows people and things to be connected Anytime, Anyplace, with Anything and Anyone, ideally using Any path/network and Any Service¹...”

During 2008, the number of things connected to the internet exceed the number of people on earth



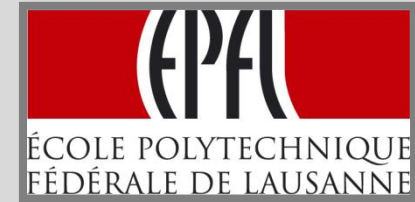
¹ P. Guillemin and P. Friess. Internet of things strategic research roadmap, Technical report, The Cluster of European Research Projects, 2009.

Background

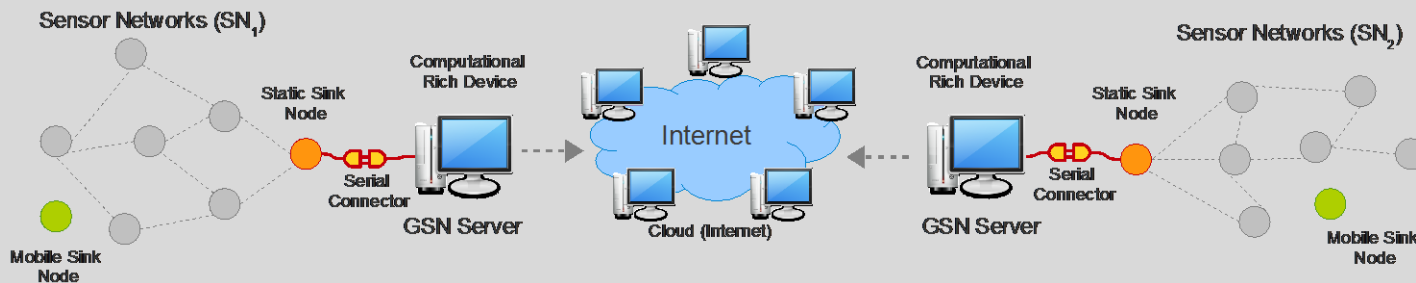
- **The Role of Mobile Phones in the IoT Paradigm**
 - Collect sensor data (from other sensors via Bluetooth)
 - Annotate sensor data (context annotation)
 - Generate sensor data (using built-in sensors)
 - Already deployed → less deployment, maintenance cost
- **How to process collected data... ? Data processing engines/middleware solutions are required to fuse sensor data from multiple sensors or multiple devices that collect sensor data...**

Data Stream Processing Engine

Global Sensor Network (GSN)



- GSN¹ project started in 2005 at EPFL in the LSIR lab by Ali Salehi (now @ CSIRO IEL) and Prof. Karl Aberer.
- A platform aimed at providing flexible middleware to address the challenges of sensor data acquisition, integration and distributed query processing
- It is used widely in over ten EU/Swiss funded research projects
- Foundation middleware for OpenIoT²/ SenseMA / Phenonet³ projects



¹ sourceforge.net/apps/trac/gsn

² openiot.eu: Open Source blueprint for large scale self-organizing cloud environments for IoT applications FP7-ICT-2011-7

³ phenonet.com : wireless sensors in agriculture

The Problem



The Problem

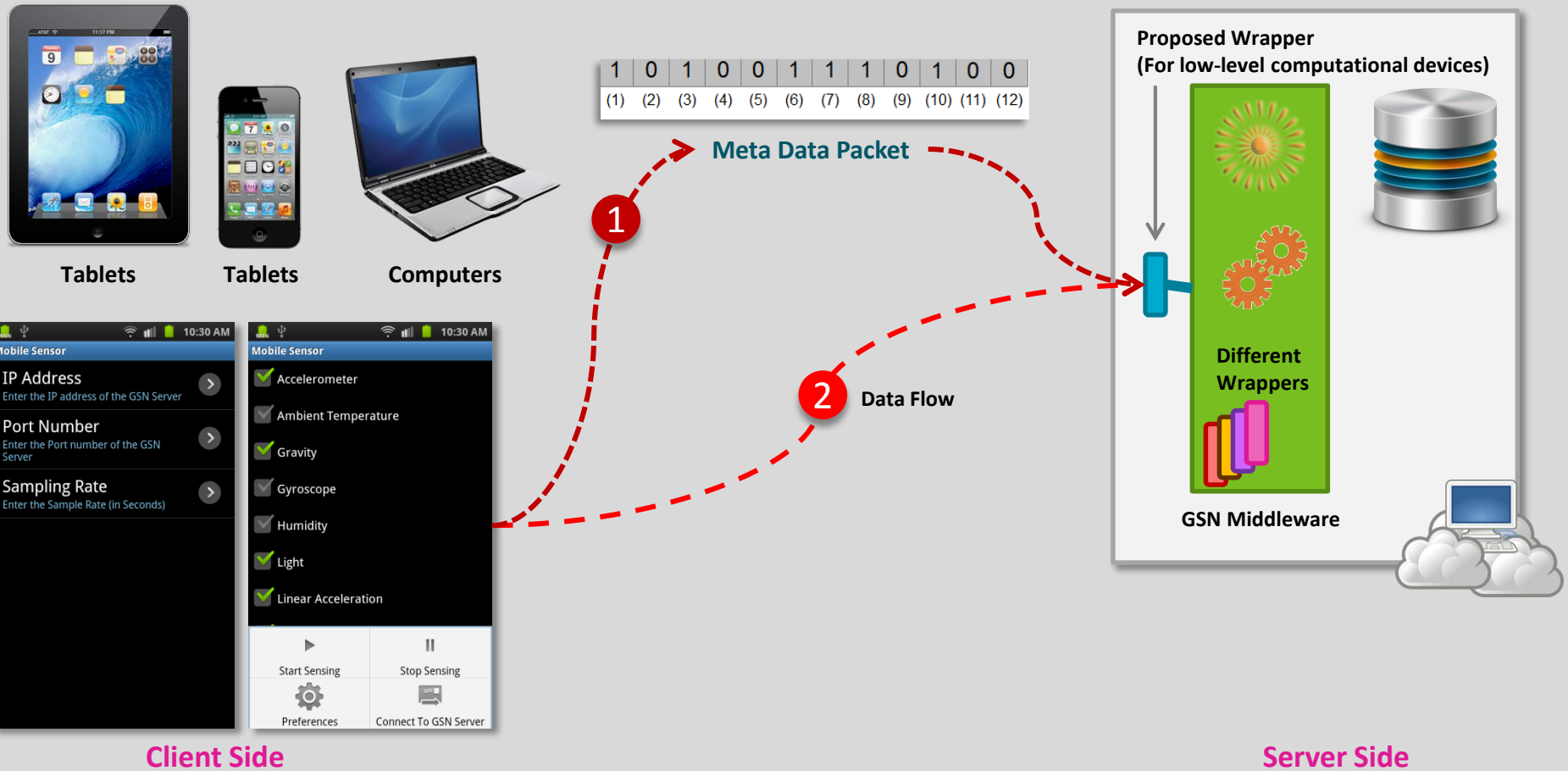
- **Is it feasible to process sensor data in the mobile... ?**
Processing and storage is still limited in mobile phones and significant amount of data processing will consume lot of energy that will discharge the battery very quickly
- Data processing engines such as GSN can be ported in to the mobile it self → Do the processing in the mobile
- Simplified version of GSN will be required.
- Is it energy efficient...?
- **Why not uploading data into a GSN instance in the cloud → Do the processing in Cloud → HOW ?**

The Proposed Solution



The Solution Proposed

Data Acquisition Model For GSN (DAM4GSN) Architecture



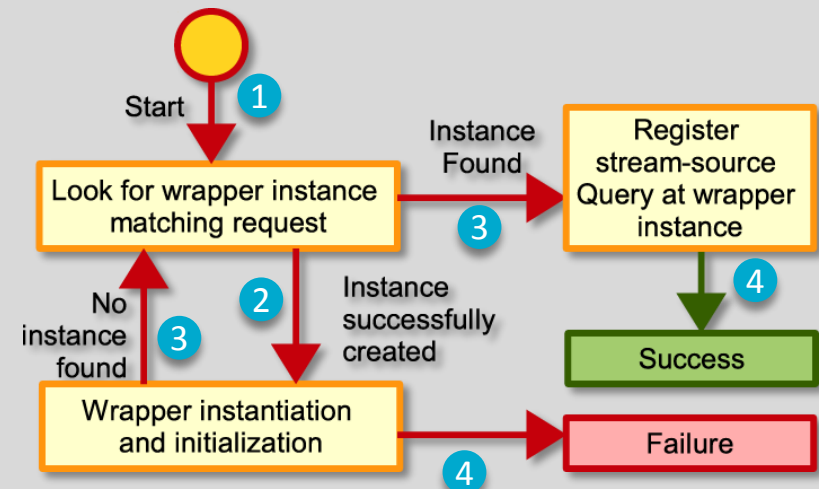
Data Stream Processing Engine

Global Sensor Network (GSN)

Wrappers == gateways, handlers, proxies, mediators...

```
public class EmptyWrapper extends AbstractWrapper {
    public boolean initialize ( ) { 1
        1. Wait for the Client to send meta data packet
        2. Analyse the Meta data packet and identify the enabled
           sensors in the client side
        3. createDataFieldCollection (enabledSensors)
        return true;
    }
    public void run ( ) { 5
        while (isActive()) {
            1. Wait for the Client to send Sensor data
            2. mapSensorData(sensorData, enabledSensors)
            .....
            StreamElement streamElement = new StreamElement (...);
            postStreamElement( streamElement )
        }
    }
    public DataField[] getOutputFormat ( ) { .... } 2
    public String getWrapperName ( ) { .... } 3
    public void finalize ( ) { .... } 4
}
private DataField[]
createDataFieldCollection(boolean[] enabledSensors) {...} 6
private void
mapSensorData(String[] sensorData, boolean[] enabledSensors) {...} 7
}
```

Android Wrapper



GSN Wrapper Life Cycle

- All the wrappers need to extend the Java class [*gsn.wrapper.AbstractWrapper*](#)
- Every wrapper should implement four methods (numbered in 1-4):
1. initialise(), 2. finalise(), 3. getWrapperName(), 4. getOutputFormat()

Performance Advantage



Performance Advantage

- **Less installation or configuration of GSN:**

GSN assumes that sensors are connected to a server that is running GSN middleware through a sink. However, installing and configuring GSN in low-level computational devices such as mobile phones and tablets would be a overwhelming task and may not be feasible due to lack of resources.

- **Scalability:**

As we do not port (install) GSN into mobile devices, scalability is preserved at the server level, probably in the cloud. Therefore, Scalability do not depend on the resource availability on the device (i.e. mobile phone).

- **No continuous update for GSN middleware:**

Any form of update may only be required to be done in the client side (i.e. in mobile phones, tablets). No update is required in GSN server.

- **Easy to extend:**

Sensing capability of the mobile phones can be extended by attaching additional hardware components. It is not required to do any changes in wrappers in GSN server.

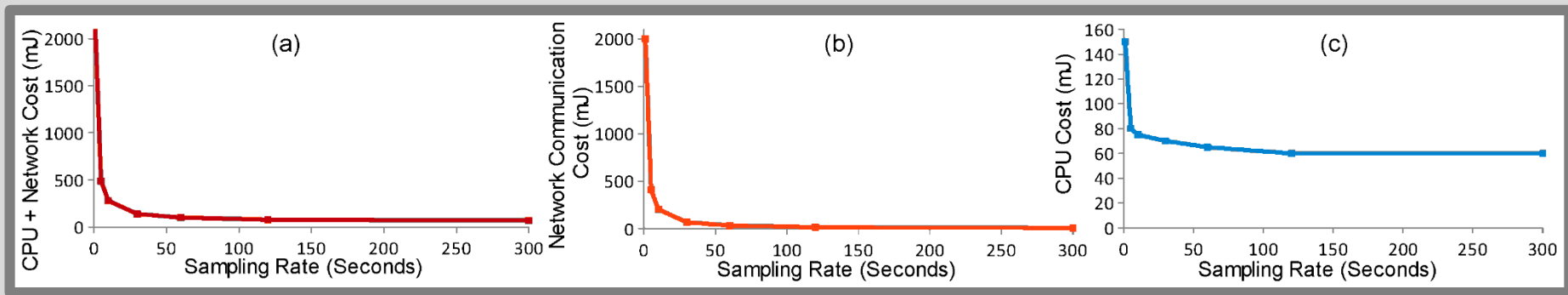
- **Support variety of low-level computational devices:**

Can be used by any mobile device or low end computing devices (e.g. mobile phones, tablets, laptops, etc.). The only capability that a mobile device need to have is sensor data collection, packet structure generation and network communication (i.e. Wi-Fi, 3G).

Evaluation



Evaluation



Energy consumption in mJ per minute

- **Experiments Setup: Samsung Galaxy S, Android platform 2.3 and PowerTutor¹ app, Intel Core i7 CPU, 6GB ram, CSIRO IE Wi-Fi network**
- **Network communication > CPU energy cost**
- **Network communication parameters such as sampling rate² should be carefully selected**

Sensor	Power (mA)
Accelerometer	0.20
Gravity	0.20
Linear Acceleration	0.20
Proximity	0.75
Light	0.75
Magnetic Field	4.00
Rotation Vector	4.20
Orientation	4.20

¹ ziyang.eecs.umich.edu/projects/powertutor

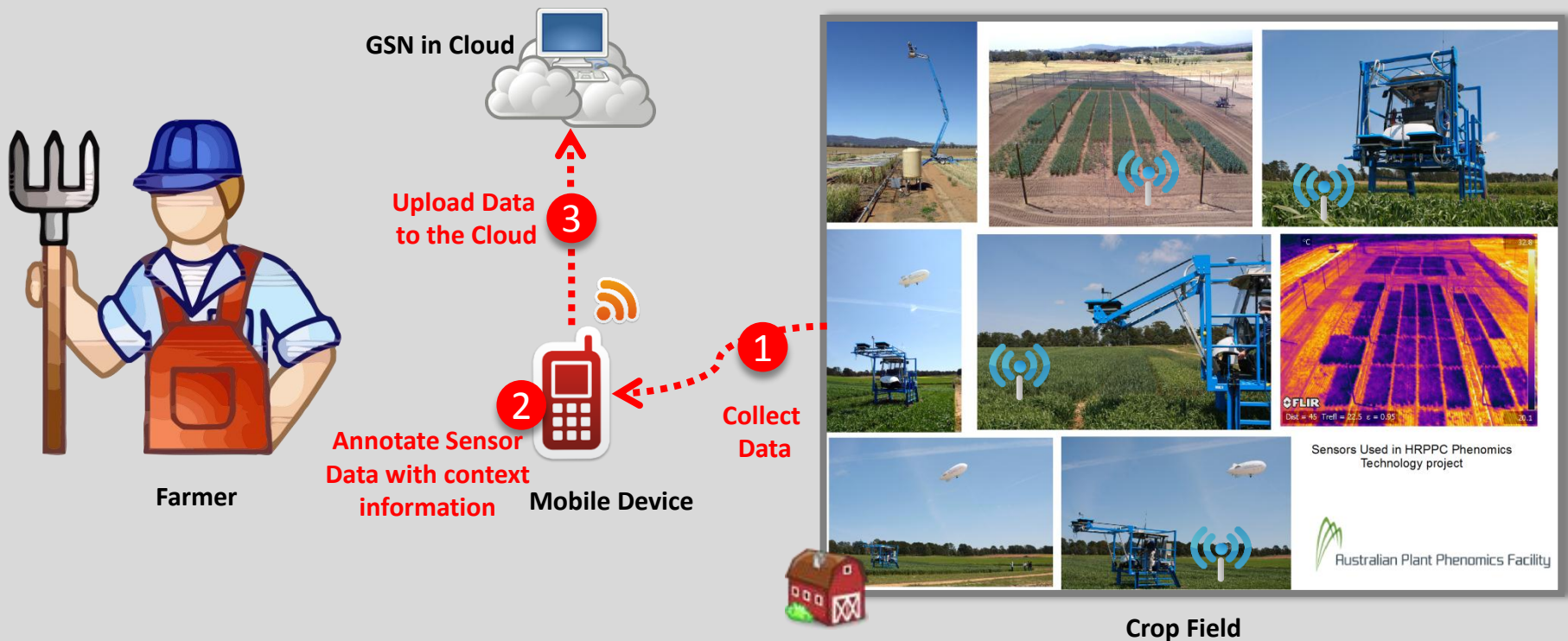
² Google I/O 2012 <http://www.youtube.com/watch?v=PwC1OIJ05VM> (Total 58 mins. Efficiency: 16:43)

Future Work



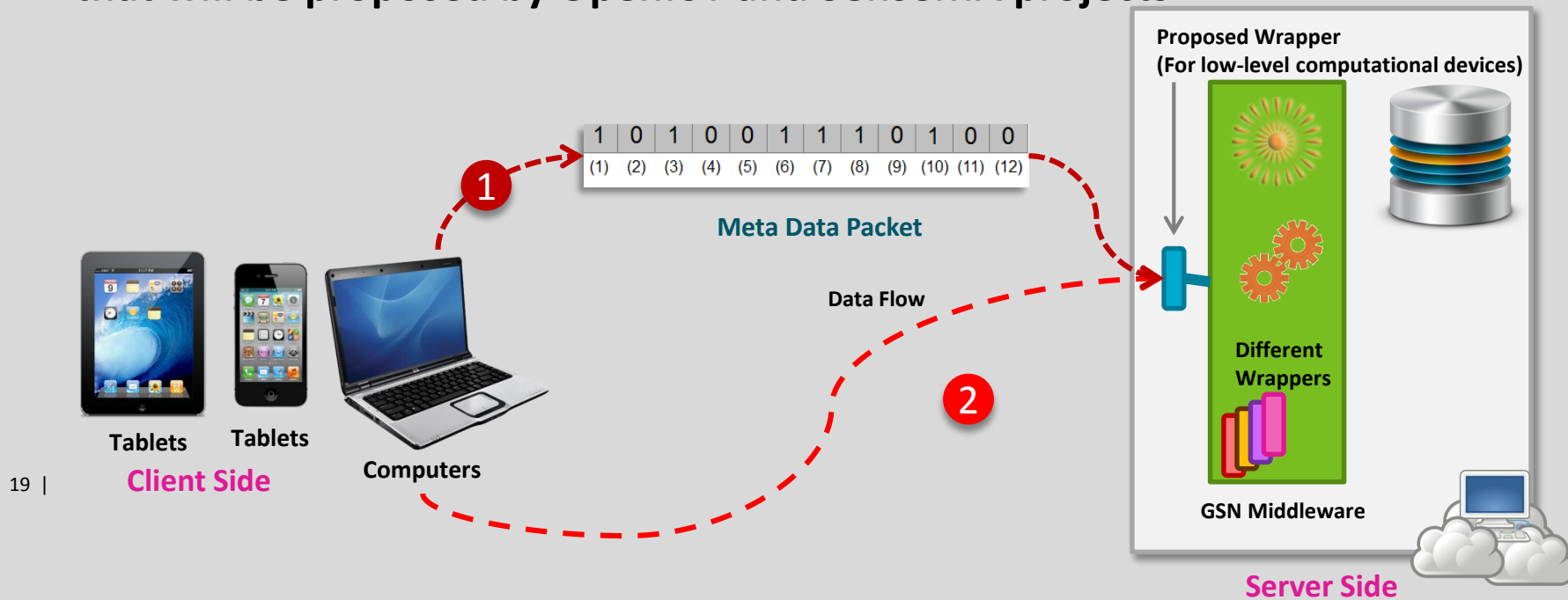
Application Scenario

A **farmer** visits his field of crops and **collects sensor data** from variety of different sensors deployed. The mobile phone **annotates** collected raw sensor data with various **context information** such as **location, time**, etc. and sends them to GSN for storage, analysis, and interpretation.



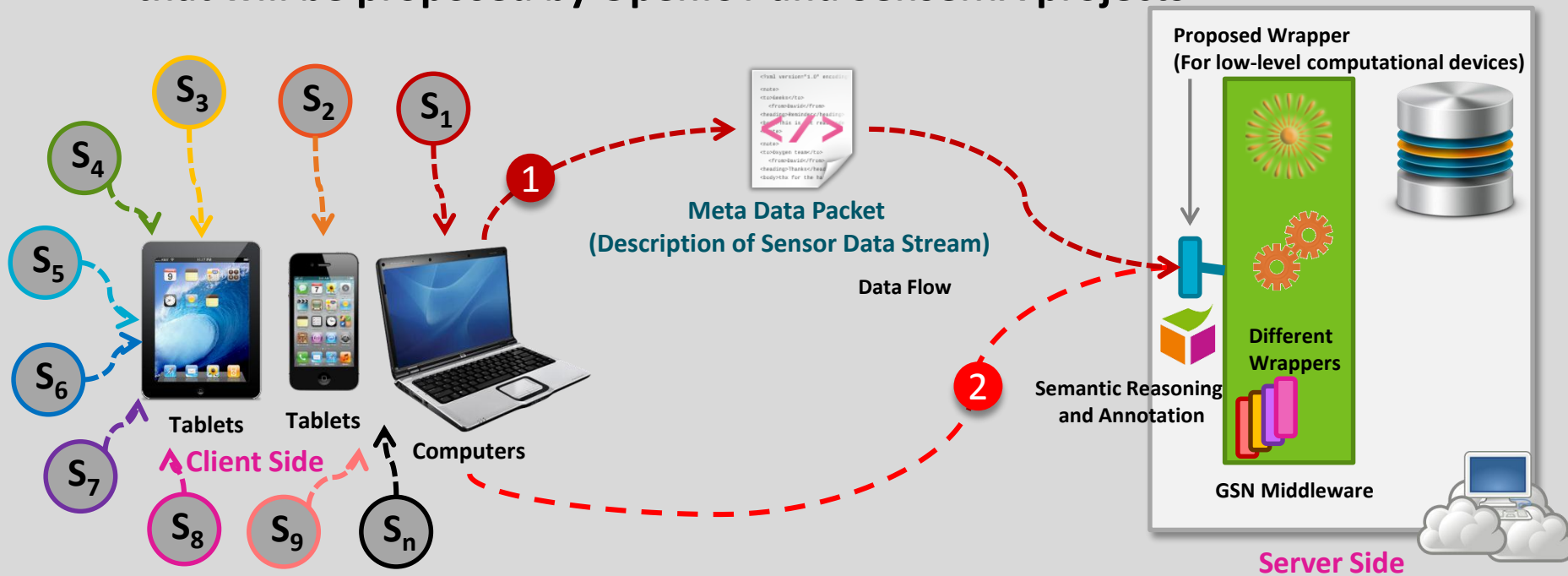
Future Work

- Auto-generation and configuration of wrappers. Generating / Configure program code based on XML descriptions.
- Combine context capturing, discovering and semantic technologies with processing of sensor data inside the wrapper itself.
- Build the DAM4GSN architecture into GSN with the other improvements that will be proposed by OpenIoT and SenseMA projects



Future Work

- Auto-generation and configuration of wrappers. Generating / Configure program code based on XML descriptions.
- Combine context capturing, discovering and semantic technologies with processing of sensor data inside the wrapper itself.
- Build the DAM4GSN architecture into GSN with the other improvements that will be proposed by OpenIoT and SenseMA projects



Thank You!

CSIRO ICT Center
Information Engineering Laboratory
Charith Perera
PhD Student

t +61 2 6216 7135
e Charith.Perera@csiro.au
w www.csiro.au/charith.perera

SEMANTIC DATA MANAGEMENT / INFORMATION ENGINEERING LAB
www.csiro.au



Appendix I:

Possible usage of sensors built-in to the mobile phones

- **Motion Sensors:**

- ✓ Accelerometer (HS) (activities, moving speed, location)
- ✓ Gravity (SS) OR (HS)
- ✓ Gyroscope (HS) (activities, moving speed, location)
- ✓ linear accelerometer (SS) OR (HS)
- ✓ rotation vector (SS) OR (HS)

- **Position Sensors:**

- ✓ Orientation (SS)
- ✓ geomagnetic field (HS)
- ✓ proximity (HS) (determine how close the face of a device is to an object)
- ✓ GPS (HS) (determine location, movements)

- **Environment Sensors:**

- ✓ Light (HS) (climate to complement weather information),
- ✓ Pressure (HS) (ambient air pressure)
- ✓ Humidity (HS) (ambient relative humidity)
- ✓ Temperature (HS) (ambient air temperature)

