

# **X-Sensor ver. 2: a mobile-agent supported sensor network testbed**

---

Tomoki Yoshihisa, Osaka University

---

# Contents

---

1. Introduction
2. System Design
3. Implementation
4. Conclusion

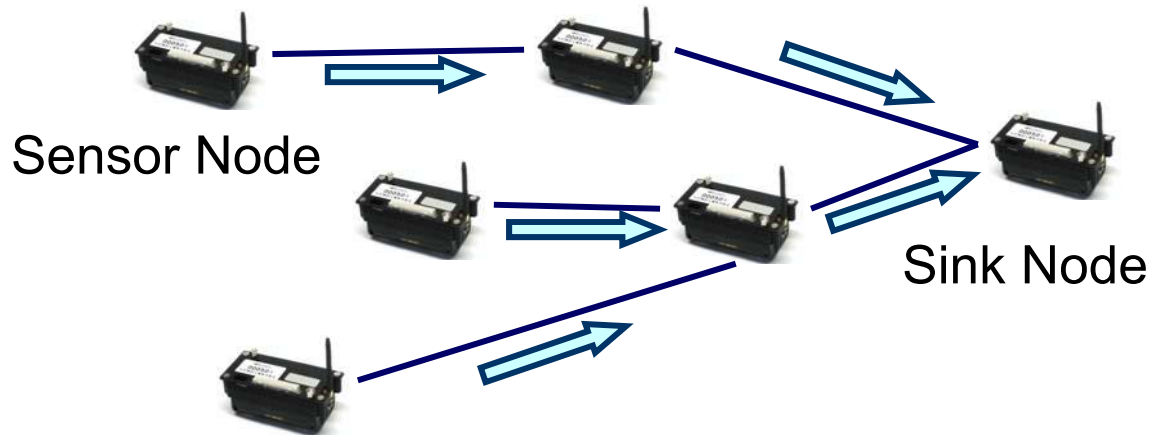
# Contents

---

1. Introduction
2. System Design
3. Implementation
4. Conclusion

# Sensor Network

Sink nodes collect sensor data (temperature, humidity).



- All sensor nodes transmit their data to the sink node.
- ⊕ The number of sensors has an upper limit.
  - The bandwidth and the computational power of sink nodes are restricted.

It is necessary to integrate multiple sensor networks to construct a huge sensor network.

# Federated Sensor Network

Federated Sensor Network

Site

We can realize various applications by collecting all sensor data from all sites.

But it takes long time to collect all sensor data since the data amount is large.

Application example:



User

- Each laboratories in a university has a site.
- To check there are someone in the university or not, the user finds the sites that illumination sensor data exceeds the threshold.

# Related Work

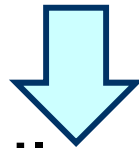
Several researches reduce the time to collect data.

- ⊕ Model-based sensor network [Wang2008]
  - The system does not collect some actual sensor data by conjecturing sensor data using a model for the sensor network.
- ⊕ B&C method [Yoshihisa2007]
  - The system broadcasts conjectured sensor data to all sensors.
  - If the error of the conjectured data is large, the sensor transmits the actual sensor data.

Systems cannot collect actual data for all sensors.

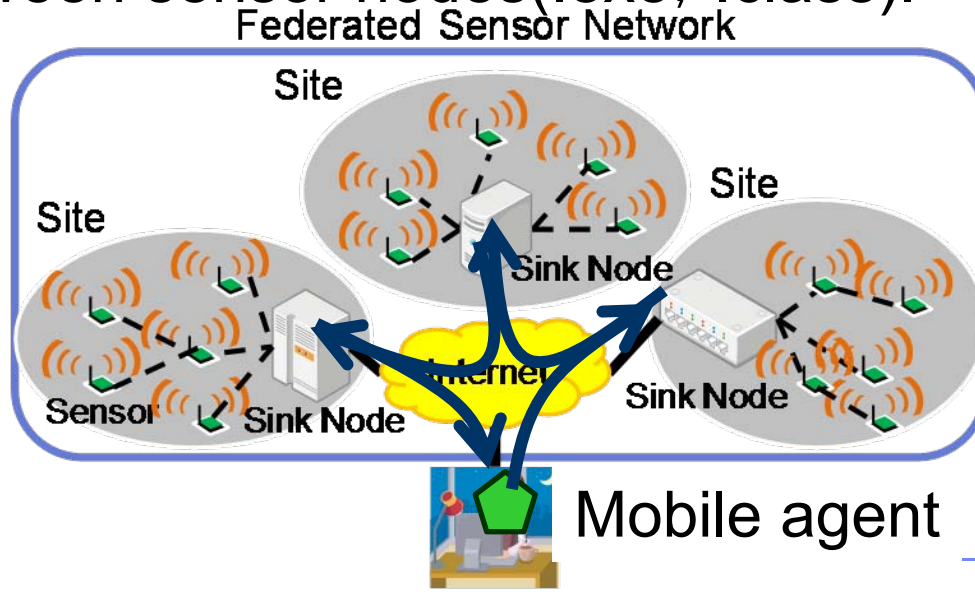
# Objective

Collecting actual sensor data rapidly from a federated sensor network



Our approach: Mobile agent

Mobile agents are executable programs that process data while migrating between sensor nodes (.exe, .class).



# Contents

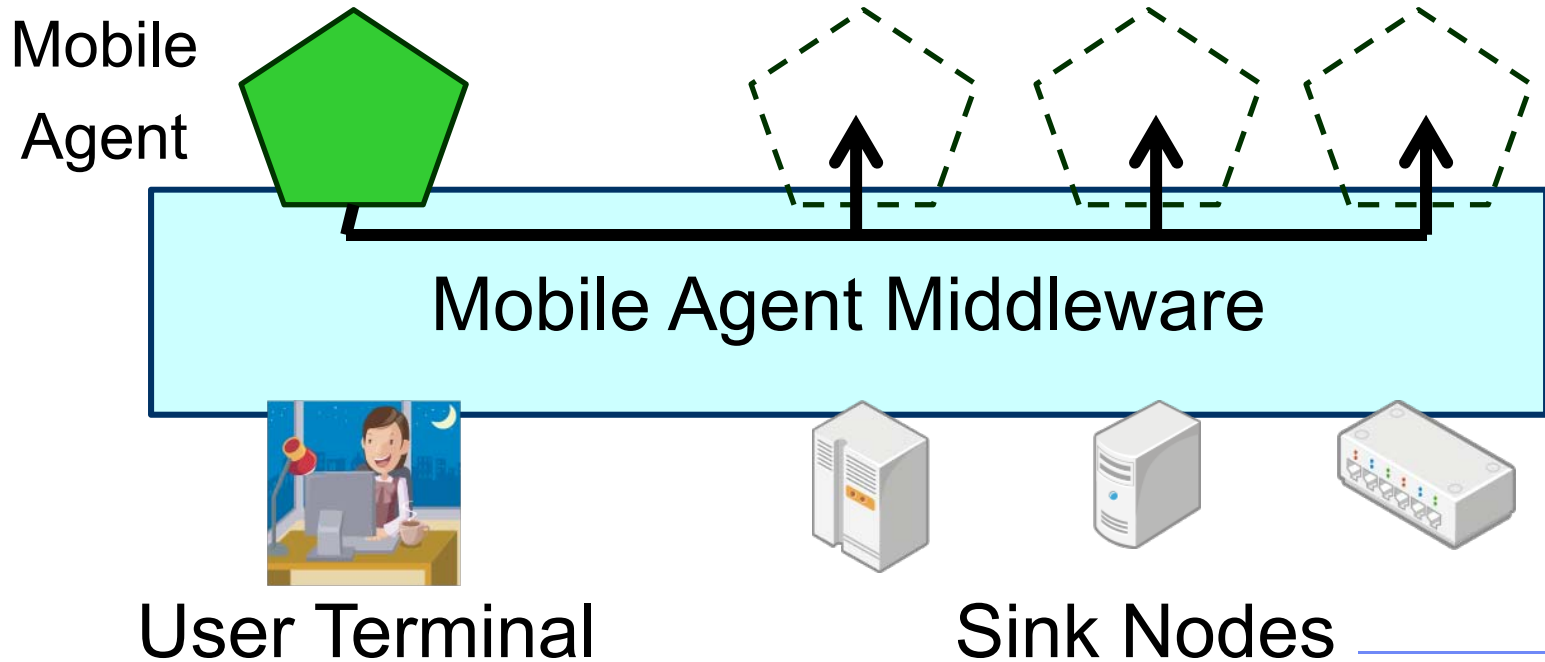
---

1. Introduction
- 2. System Design**
3. Implementation
4. Conclusion



# Outline for Data Collection System using Mobile Agents

- ⊕ Mobile Agent Middleware
- ⊕ User Terminal
- ⊕ Sink Nodes
- ⊕ Mobile Agent



# Main Necessary Functions (1/2)

## ⊕ For mobile agent middleware

### ■ Mobile agent control

- That is, The middleware generates, migrates, and controls mobile agents.

### ■ Construction of the federated sensor network

- Since the middleware has to find sink nodes in the federated sensor network.

## ⊕ For user terminals

### ■ Mobile agent generation

- User terminals generate mobile agents using the mobile agent middleware.
- Users describe processes for mobile agents using user s' terminals.

# Main Necessary Functions (2/2)

## ⊕ For sink nodes

- Data collection: Sink nodes collect sensor data.
- Sink nodes have to be able to use mobile agent middleware.

## ⊕ For mobile agents

- Mobile agents have to be able to execute the user described processes on sink nodes.
- Mobile agents have to be able to be controlled by the mobile agent middleware.

# Contents

---

1. Introduction
2. System Design
- 3. Implementation**
4. Conclusion

# Implementation

---

We implemented X-Sensor2

⊕ X-Sensor2 extends X-Sensor so that it can use mobile agents.

⊕ Mobile agent middleware: PIAX

⊕ Federated sensor network: X-Sensor

# PIAX

A P2P platform for mobile agents

⊕ The platform satisfies the previous necessities.

- Mobile agent control
- Construction of the federated sensor network

⊕ PIAX peers (Peers in PIAX)

- Run on each sink node.
- Mobile agents migrate between PIAX peers.

⊕ Mobile agent

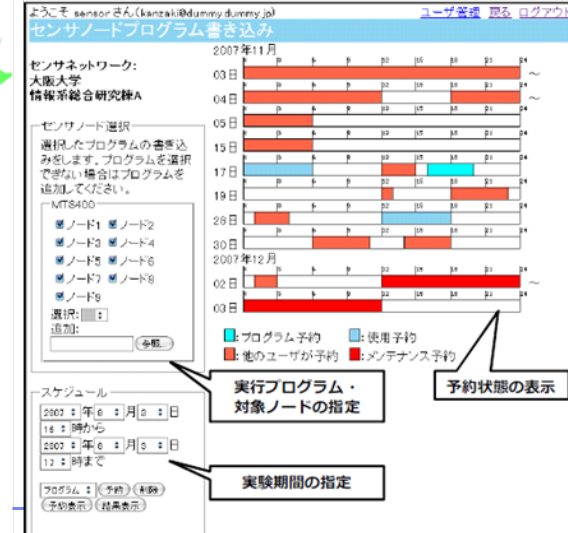
- Java implementation → independent from OS

# X-Sensor

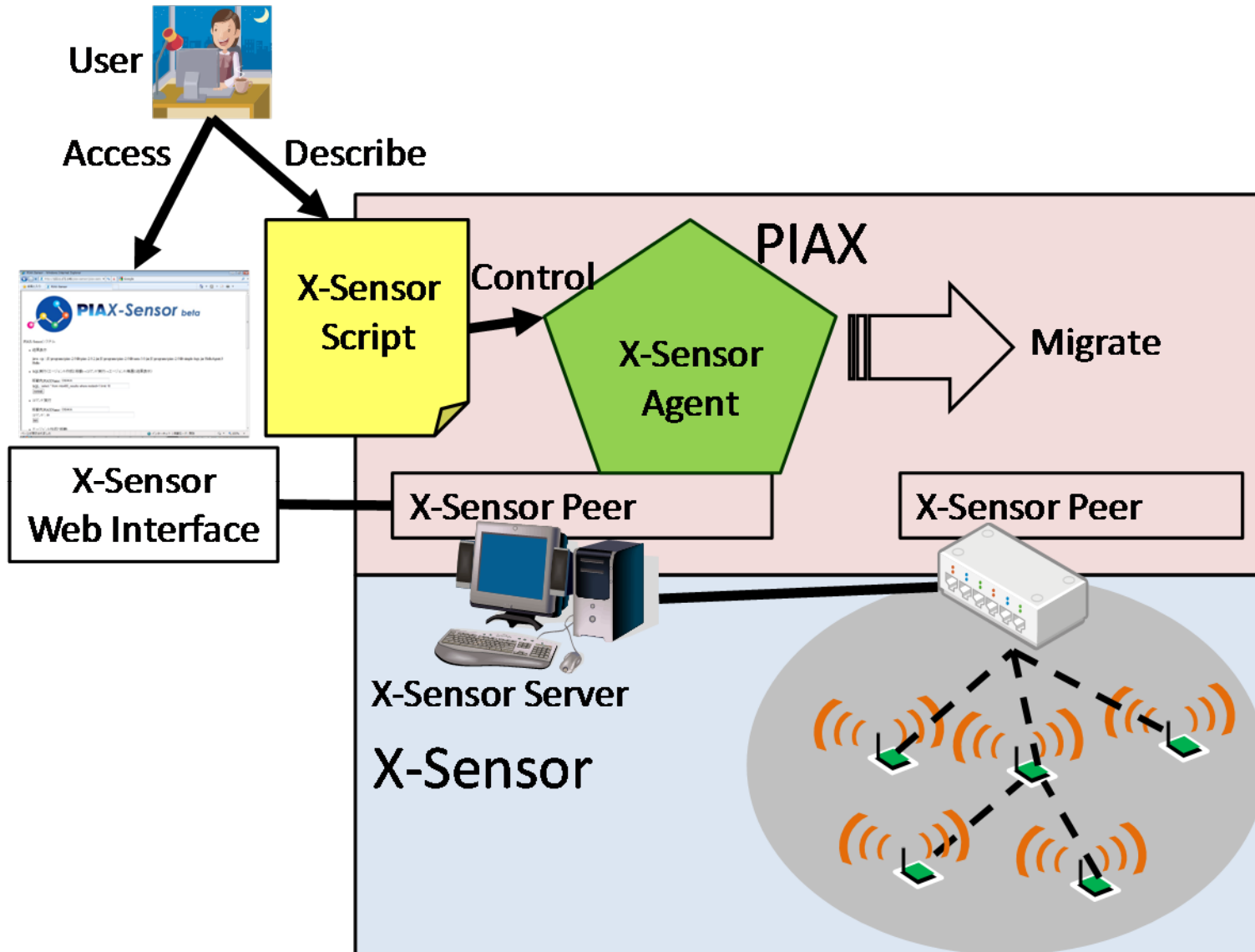
A federated sensor network built as a sensor network testbed.

- Several Japanese universities join to the testbed as sites.
- There are approximately 100 sensors in X-Sensor.
- X-Sensor is a relatively large sensor network and suitable for this research.

- Crossbow MICAz
- Temperature, humidity  
Illumination, acceleration



# System Architecture for X-Sensor2





# Components for X-Sensor2

---

- ⊕ X-Sensor peer
- ⊕ X-Sensor agent
- ⊕ X-Sensor web interface
- ⊕ X-Sensor script

# X-Sensor Script

- ⊕ Describes processes executed by mobile agents
- ⊕ Global Part: global variable definitions
- ⊕ Local Part: Processes executed every mobile agents migrate

Method	Argument	Content
SetCommand	<i>Command</i>	Stores executable command <i>Command</i>
Execute	<i>None</i>	Execute command
SQL	<i>Query</i>	Execute SQL sentence <i>Query</i>
Return	None	Returns the mobile agent to the X-Sensor server
Travel	<i>PeerID</i>	Migrates the mobile agent to X-Sensor that PeerID is <i>PeerID</i>

変数名	内容
Result[]	Array stores results
ResultCount	# of lines in Result[]
ResultCode	Result code for the process
PeerIds[]	The PeerIds that the mobile agent migrate to
PeerIdsPosition	The current position in PeerIds[]

### 3: Implementation

The mobile agent migrates two sites, OSAKA and KYOTO.  
Then, it calculates the maximum value.

## Global Part

```
public String OSAKA;  
public String KYOTO;
```

## Local Part

```
if(PeerIdsPosition==0){  
    SQL("select max(humtemp) from mts400_results_OSAKA");  
    OSAKA=Result[2];  
}else if(PeerIdsPosition==1){  
    SQL("select max(humtemp) from mts400_results_KYOTO");  
    KYOTO=Result[2];  
    ResultCount=0;  
    Result[ResultCount++]="OSAKA:"+OSAKA;  
    Result[ResultCount++]="KYOTO:"+KYOTO;  
    if(OSAKA.compareTo(KYOTO)>0){  
        Result[ResultCount++]="MAX:"+OSAKA;  
    }else{  
        Result[ResultCount++]="MAX:"+KYOTO;  
    }  
}  
}
```

Executed  
in the first  
PIAX  
peer

Executed  
in the  
second  
PIAX  
peer

# Demonstration Movie

---

- ⊕ We show 9 sites located in Japan
- ⊕ The mobile agent moves to Akashi, Osaka, Wakayama, and Kobe to make temperature graph for these areas.
- ⊕ Finally, the mobile agent depicts the result graph.
- ⊕ To show the movement of the mobile agent, we made the speed of the mobile agent very slow.

# 内容

---

1. Introduction
2. Design
3. Implementation
- 4. Conclusion**

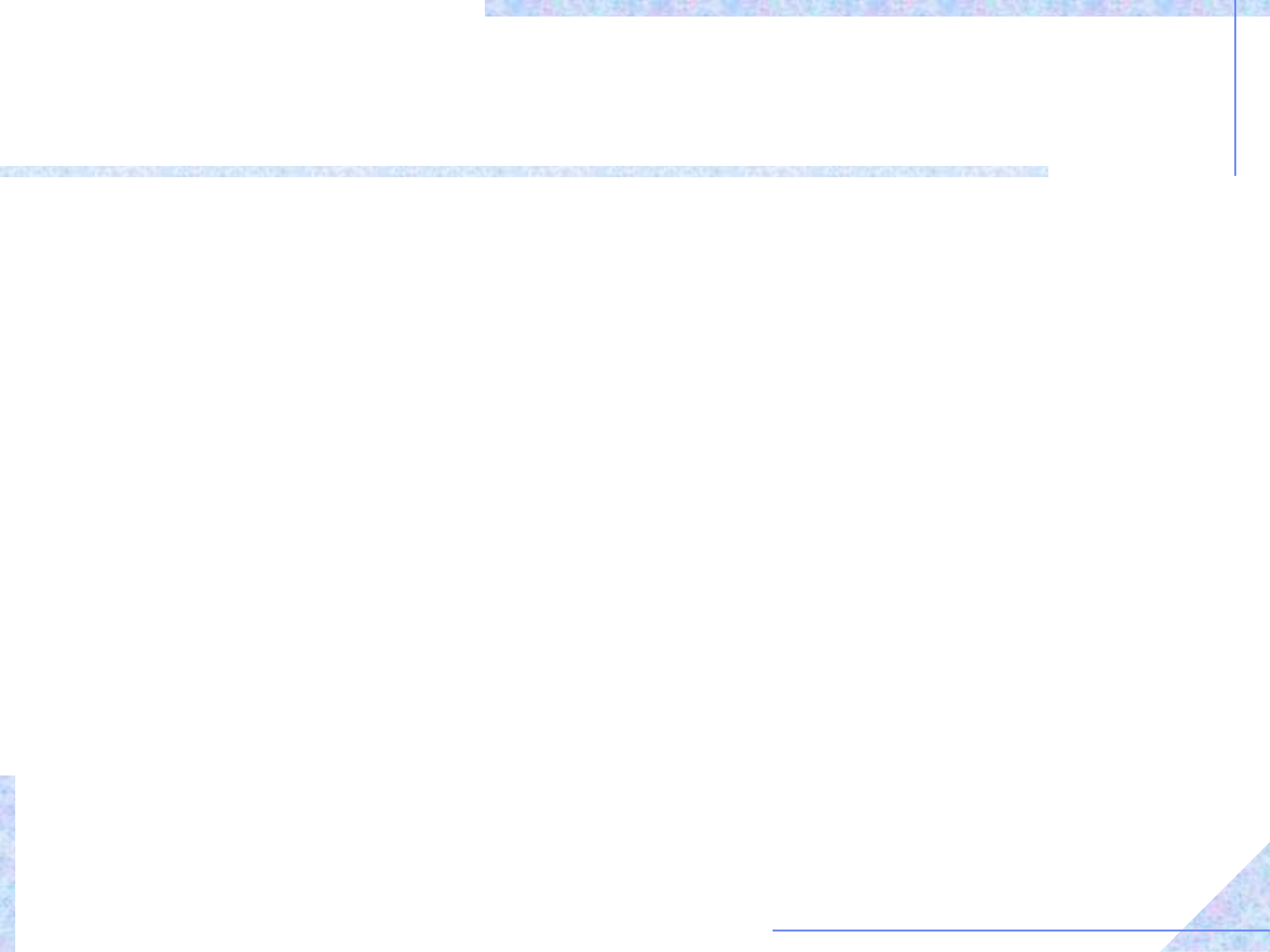
# Conclusion

---

- ⊕ We designed and implemented a data collection system for federated sensor networks using mobile agents.
- ⊕ The system does not need to collect all sensor data from all sensors since mobile agents migrates between sink nodes to collect sensor data.

## [Future Work]

- ⊕ Evaluation
  - ⊕ Cooperation of mobile agents
-



# Federated Sensor Network

## Applications for federated sensor networks

### 1. The query needs to compare multiple sites

The mobile agent calculates the maximum temperature to find the hottest laboratory.

### 2. The query does not know where to ask.

The mobile agent finds the site of that illumination data exceeds the threshold to find whether someone in the room.

### 3. The query needs data for multiple sites.

The mobile agent calculates the average for all sites.

We can realize these applications by collecting all data from all sites. But, this causes much traffics and takes too long time.



# How to get metadata

Each site is organized by different organizations and the sensor data format differs.

- ⊕ The system sets a metadata server and mobile agents get the metadata from it.
- ⊕ By providing changing formula from the user described format to the actual sensor data format, mobile agents can process the data even when the format differs.

# Merit and demerit of mobile agents

## ⊕ Merit

- We can calculate certain values from sensor data while the mobile agents are migrating.
  - We can calculate the maximum value of temperature while collecting data.

## ⊕ Demerit

- Users have to describe the movement of mobile agents in detail for an effective data collection.
  - Where to migrate agents? When migrate?
- But, by making mobile agents move autonomously, this demerit can be relieved.

# Demonstration

## ⊕ Cites

- OSAKA Cite→A lab. In Osaka university
- KYOTO Cite→A lab. In Kyoto University

## ⊕ X-Sensor Script

- Obtains the maximum temperature from OSAKA
- Obtains the maximum temperature from KYOYO  
Compare the value with that of OSAKA

# Screen shot

The screenshot shows a web browser window titled "PIAX-Sensor - Windows Internet Explorer". The address bar shows the URL "C:\Users\yoshihisa\Documents\2009\論文" and the search engine is set to "Google". The page content includes the "PIAX-Sensor" logo and the text "X-Sensor2".

Under the heading "PIAX-Sensorシステム", there is a list of items:

- PIAX-Sensorスクリプト

The "移動先(スペース区切り):" field contains the text "OSAKA KYOTO".

The "スクリプト(グローバル):" field contains the following code:

```
public String OSAKA;  
public String KYOTO;  
  
if(PeerIdsPosition==0){  
    SQL("select max(humtemp) from mts400_results_20090606");  
    OSAKA=Result[2];  
}else if(PeerIdsPosition==1){  
    SQL("select max(humtemp) from mts400_results_20090606");  
    KYOTO=Result[2];  
    ResultCount=0;  
    Result[ResultCount++]="OSAKA:"+OSAKA;
```

The "スクリプト(ローカル):" field is empty.

At the bottom of the page, under "結果表示", the results are shown:

```
OSAKA: 7231  
KYOTO: 6356  
MAX: 7231
```

Annotations on the image include:

- "Migrate to" with a downward arrow pointing to the "移動先" field.
- "X-Sensor Script" with an arrow pointing to the code editor.
- "Results" with an arrow pointing to the output box.

The Windows taskbar at the bottom shows the system tray with the text "ページが表示されました" and "コンピューター | 保護モード: 無効".