SensorAct: A Privacy and Security Aware Federated Middleware for Building Management

Pandarasamy Arjunan[1], Nipun Batra[1], **Haksoo Choi**[2], Amarjeet Singh[1], Pushpendra Singh[1], Mani B. Srivastava[2]

IIIT-Delhi[1], UCLA[2]

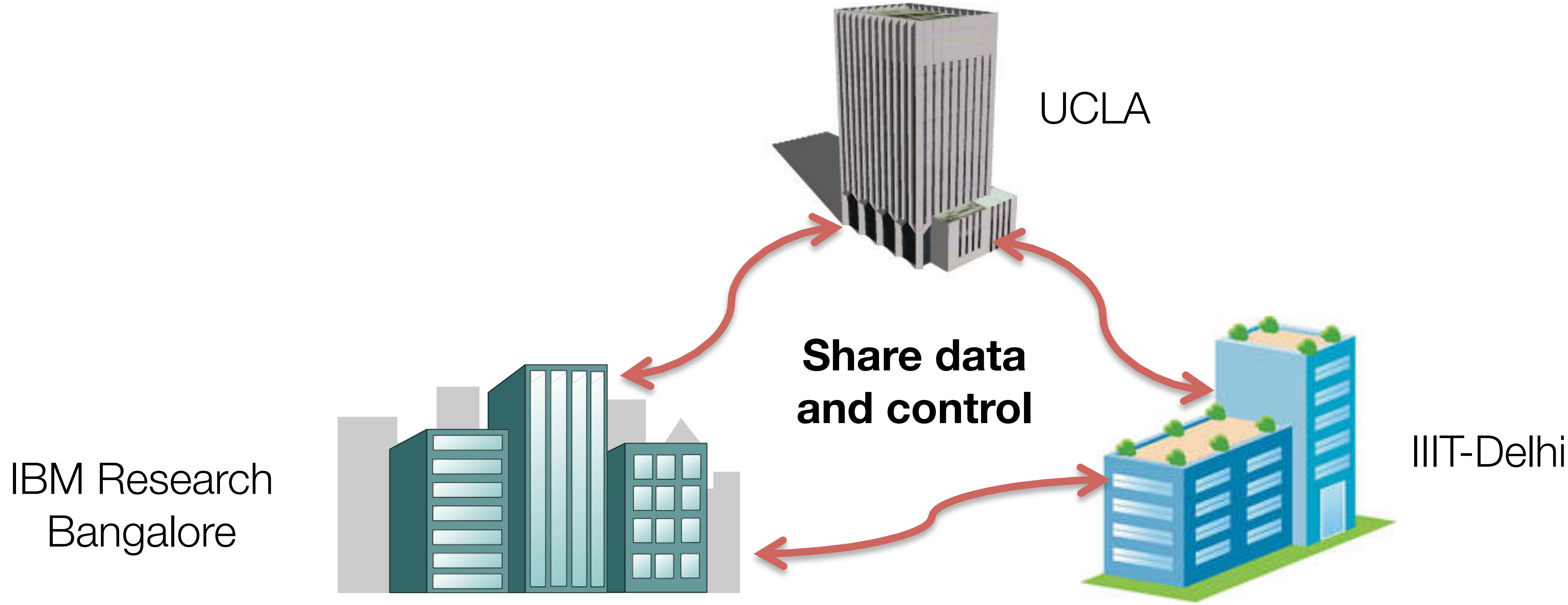
ACM BuildSys 2012

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

UCLA

Motivation

- Richer ecosystem for sharing building resources (both sensing and actuation) across several collaborators
- Globally distributed Indo-US testbed raises challenges not encountered in a building and campus scale deployments.



Motivation

Privacy & Security

Selective Data Sharing
Protected Actuation
Behavioral Privacy

Sense-Decide-Actuate

Closing Control Loop within Middleware to Address Latency

Diverse Sensors and Actuators

Electricity, Water, Gas, Occupancy, and More

- Existing systems
  - Building Scale: Trane, Johnsoncontrols, etc.
  - Home Scale: Micasaverde, etc.
  - Cloud-based: Cosm, Thingspeak, etc.

  - Provide some support in each one of these requirements.
  - Not scale to globally distributed testbed across organizations.

SensorAct in a Nutshell

- **Our Goals**
  - **Scalable**
    - Deployable from homes to across organizations
  - **User-Centric**
    - Participatory engagement of occupants
  - **Versatile**
    - Diverse sensing, actuation, and applications

A Tiered and Distributed Architecture

Powerful Sensor and Actuator Guard Rules

Lightweight Tasking Framework

Support Diverse Applications and Researches

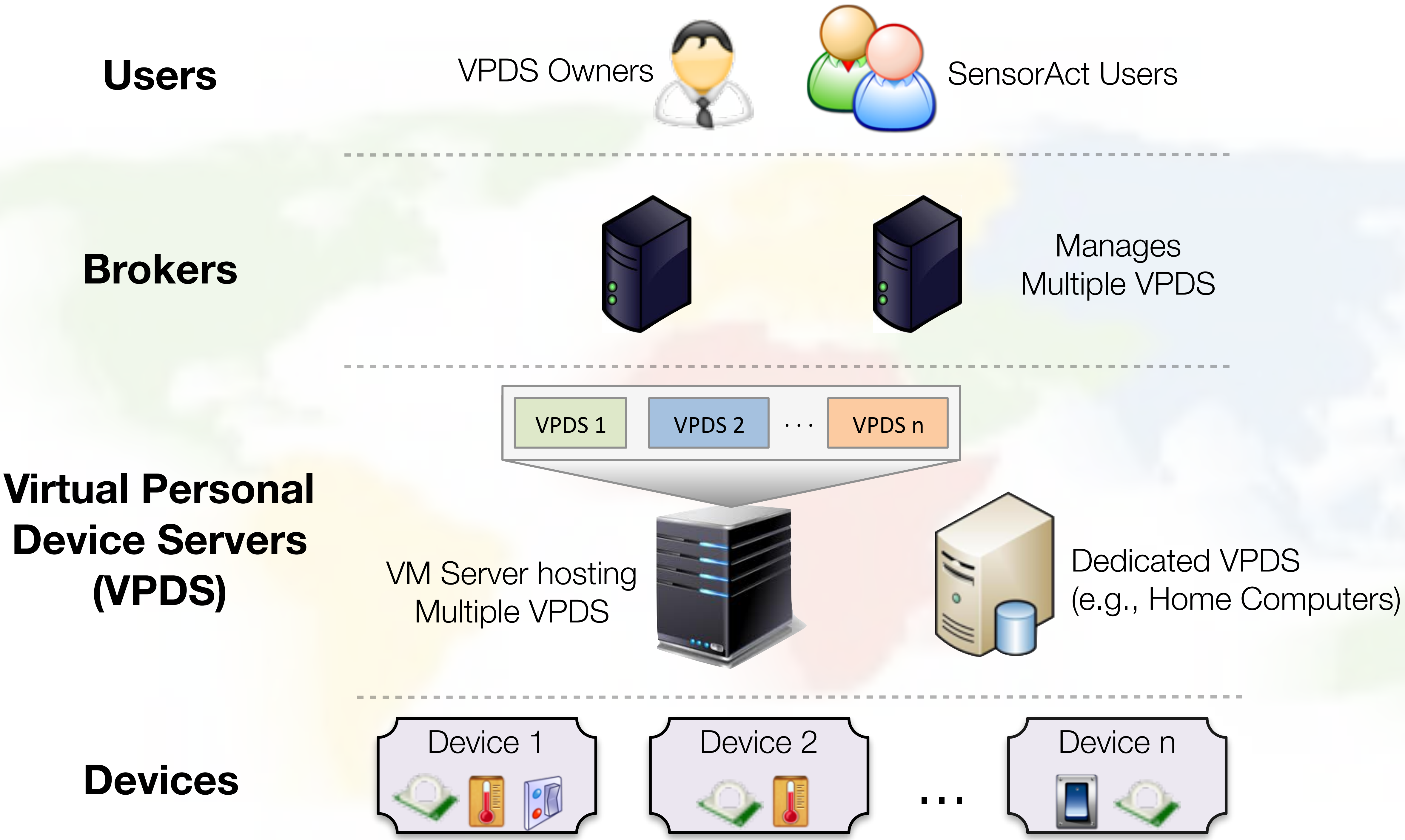
Energy (electricity, water, and gas) management

Sustainable buildings

Resource management and utilization analysis

...

Introducing SensorAct Architecture



Users

VPDS Owners

SensorAct Users

Brokers

Manages Multiple VPDS

Virtual Personal Device Servers (VPDS)

VPDS 1

VPDS 2

· · ·

VPDS n

VM Server hosting Multiple VPDS

Dedicated VPDS (e.g., Home Computers)

Devices

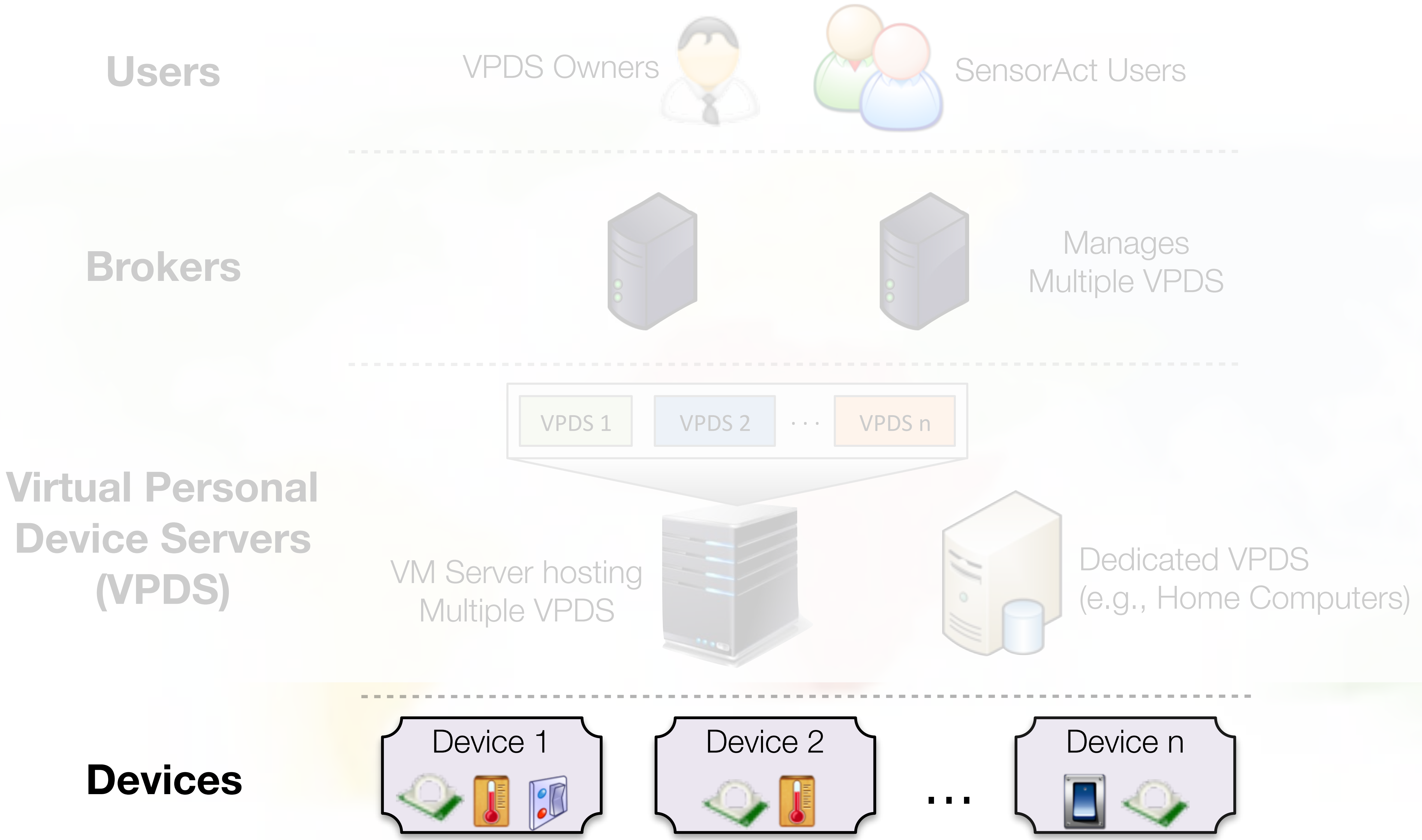
Device 1

Device 2

. . .

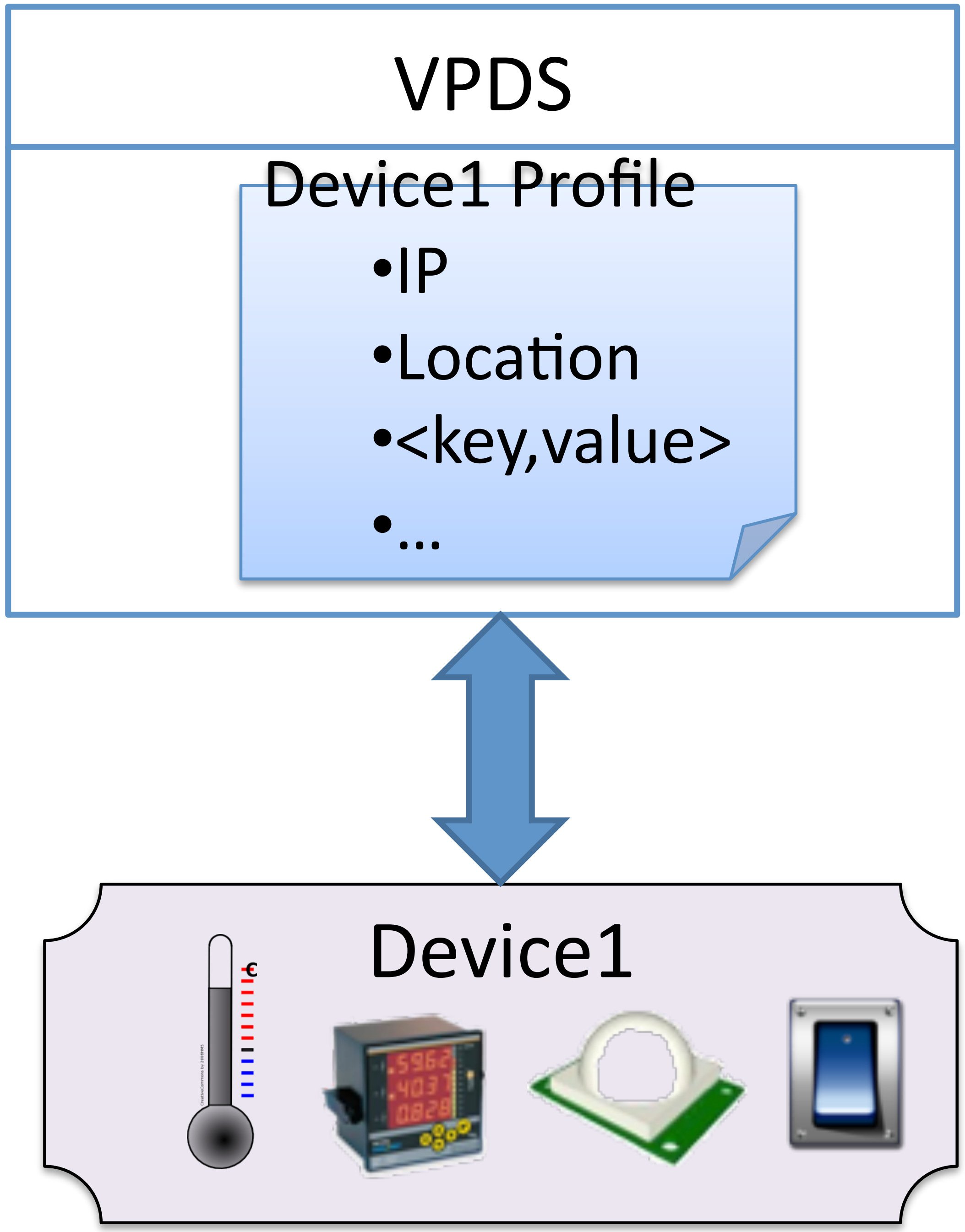
Device n

SensorAct Devices



SensorAct Devices

- Supporting diverse COTS or custom sensors and actuators with diverse interface
- Device
  - Collection of Sensors and Actuators
  - Multi-channel
- Device Profile
  - Various attributes
  - IP, Location, etc.
- Communication with VPDS
  - Device initiated or VPDS initiated
  - A device can communicate with multiple VPDS



Computed Sensor, Grouped Actuator

- For convenience and simplicity in accessing sensors and actuators
- Expose computed higher level information from a single or multiple sensors
- Assign single semantic meaning to multiple actuators
  - Similar to the concept of "Scenes" in commercial home automation systems
- e.g., In Meeting, Power off the room

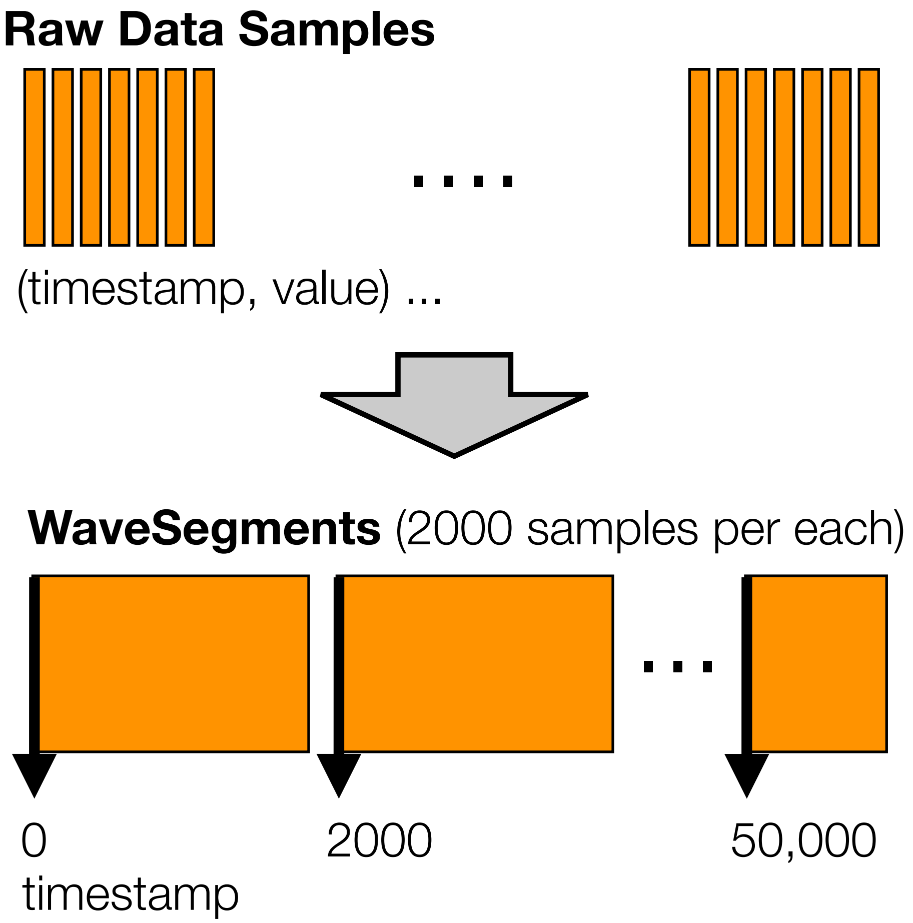
InMeeting Sensor

Computed Sensor

Grouped Actuator

Efficient Representation of Sensor Data

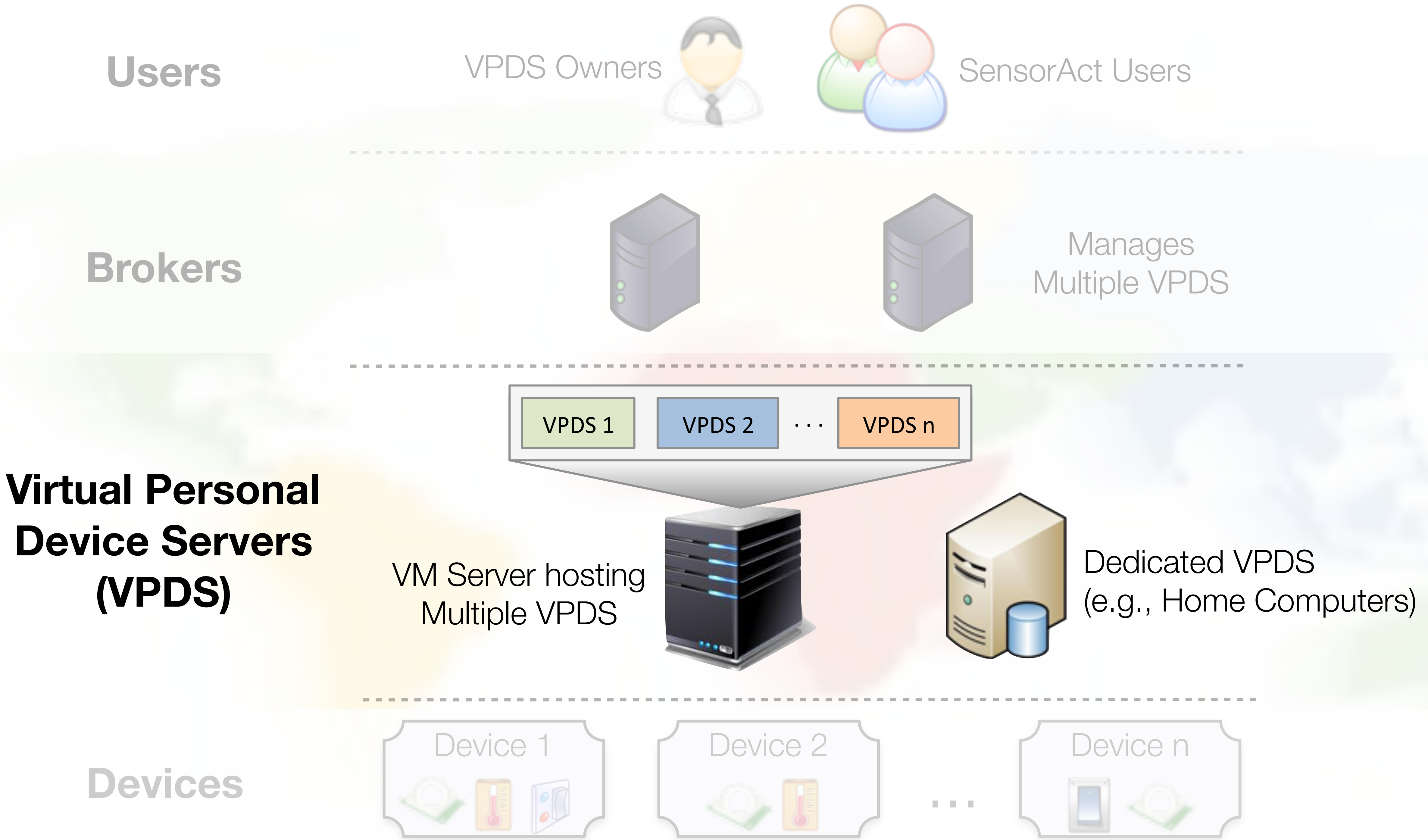
- Storing individual samples
  - Inefficient disk space and query processing time
- Inspired by SigSeg in MIT's XStream
- WaveSegs: non-overlapping windows of sensor stream with meta data
- Support diverse sampling schemes and sensor attributes



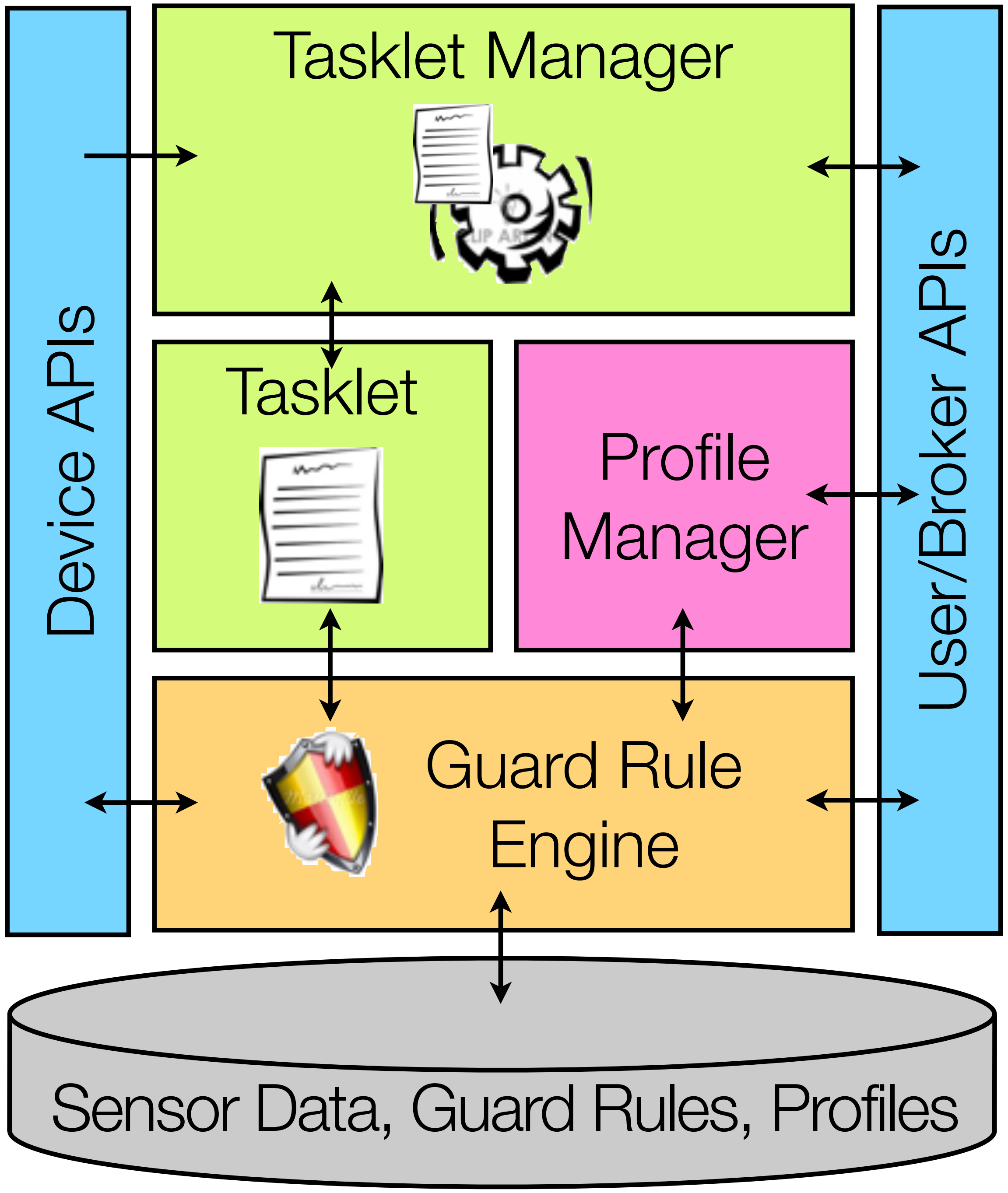
```
{
  "DEVICE_NAME": "Office_Flyport",
  "SENSOR_NAME": "MultiSensor",
  "SENSOR_ID": 1,
  "SAMPLING_INTERVAL" : 1,
  "EPOC_TIME": 1344147449,
  "CHANNELS": [
    {
      "NAME": "Temperature",
      "UNIT": "Celsius",
      "READINGS": [28.1,28.2,28.6,28.5,28.2...]
    },
    {...}
  ]
}
```

JSON Representation of WaveSeg

SensorAct VPDS

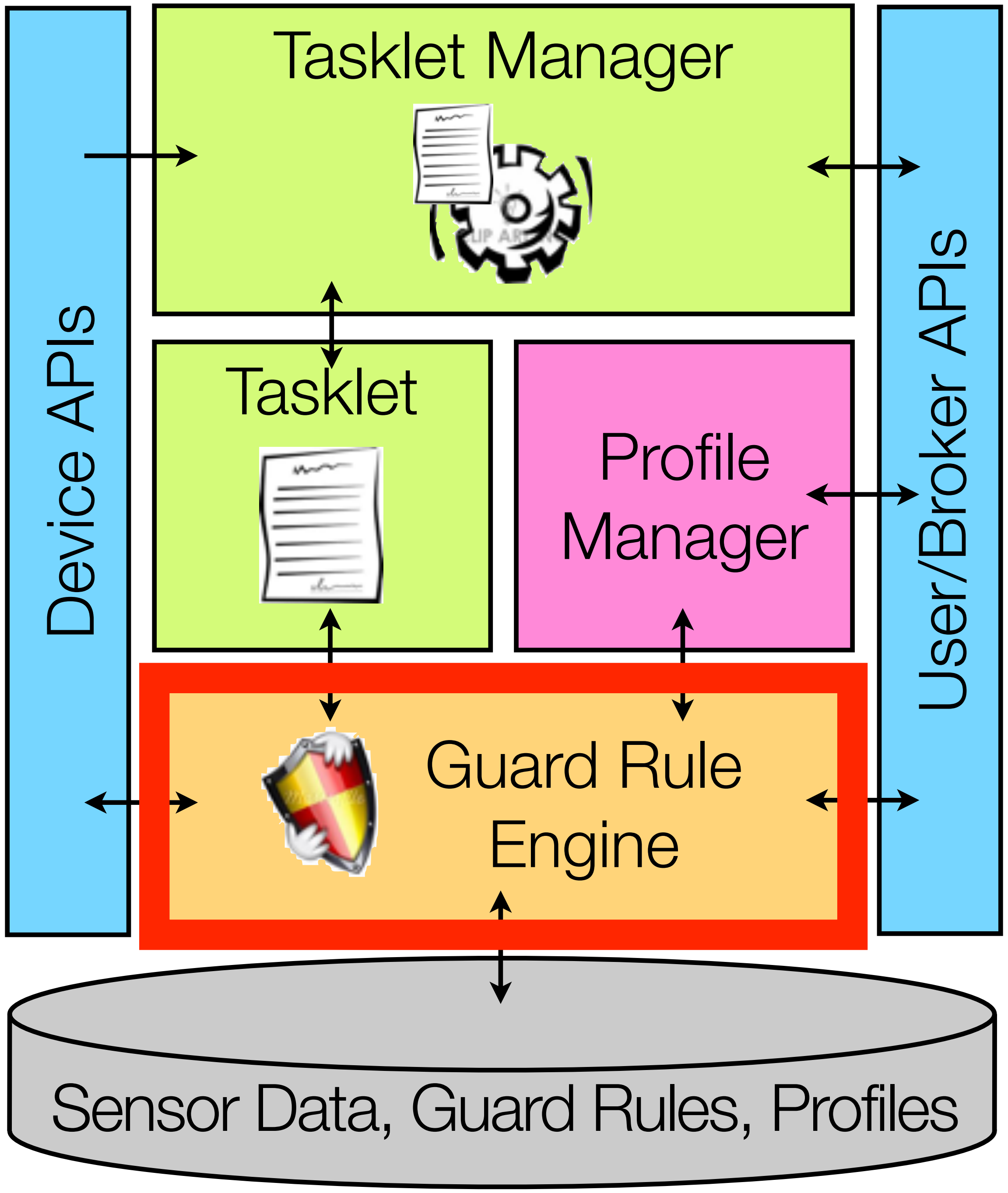


SensorAct VPDS



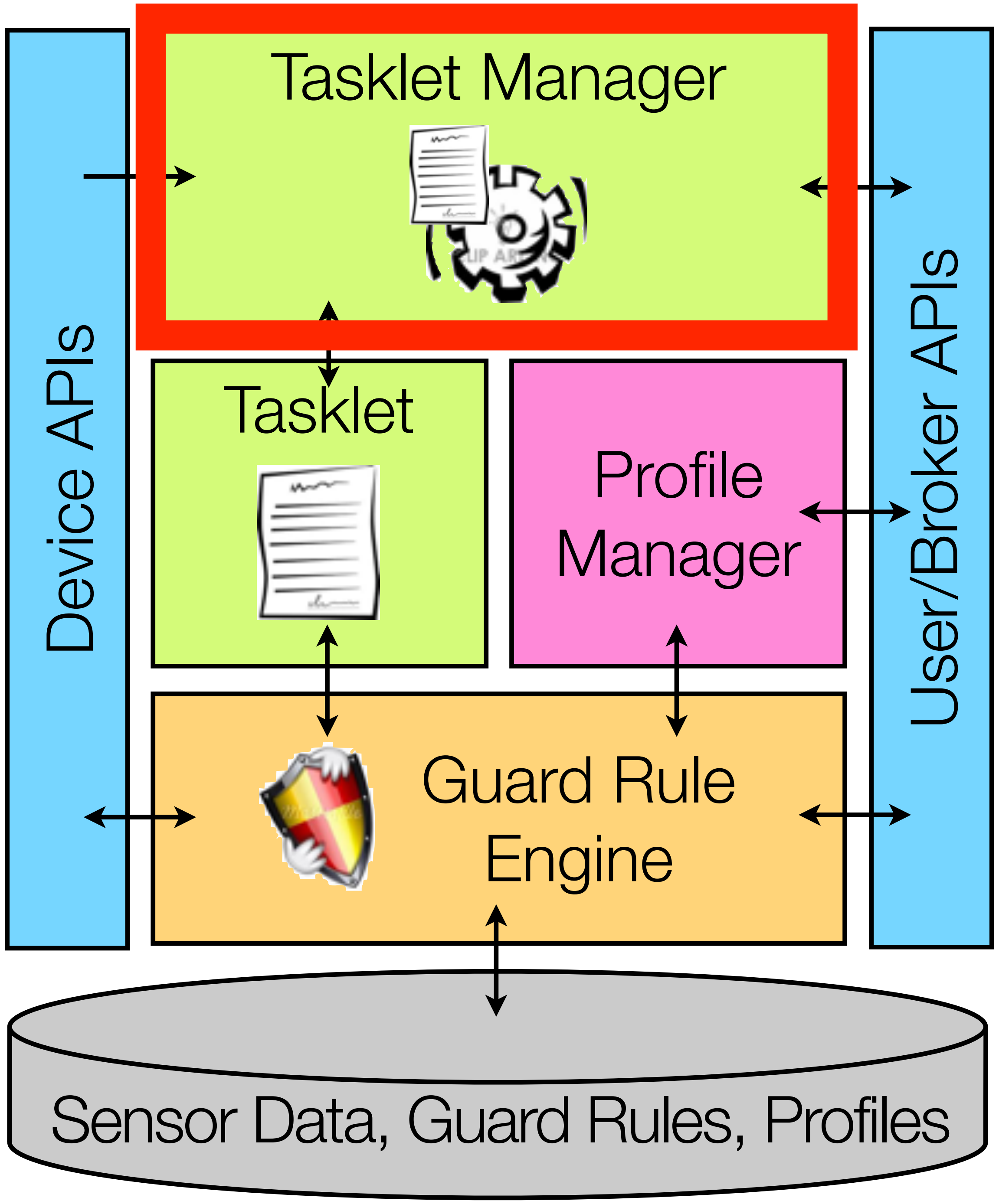
- Per user basis ensuring data ownership
- Guard Rule Engine
  - Protect privacy and security in accessing sensors and actuators
- Tasklet Manager
  - Manage and executes user written application logic
- Profile Manager
  - Manages user and device information
- Database
  - Stores sensor data, guard rules, device profiles.
- APIs
  - Device APIs, User/Broker APIs

SensorAct VPDS



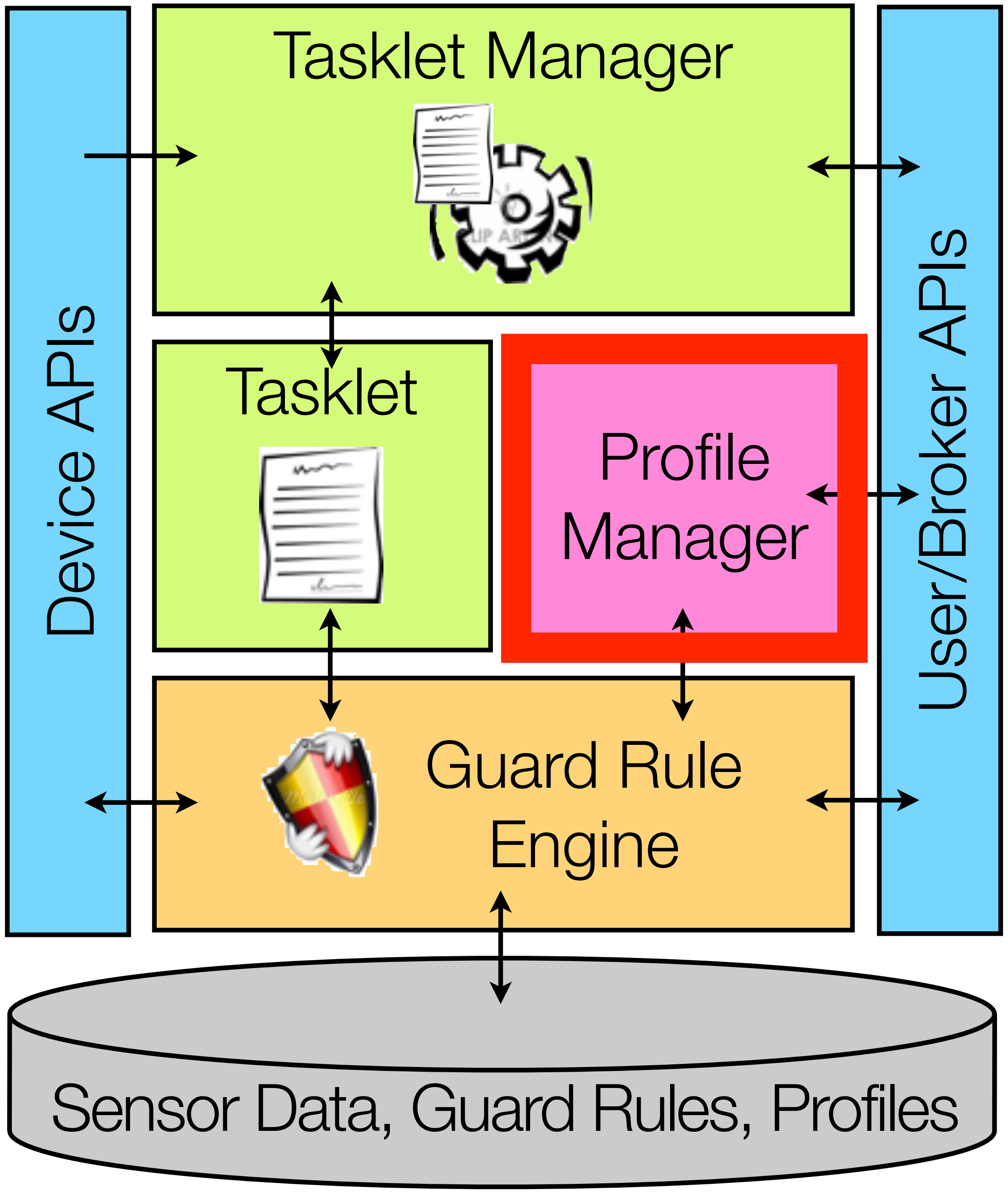
- Per user basis ensuring data ownership
- Guard Rule Engine
  - Protect privacy and security in accessing sensors and actuators
- Tasklet Manager
  - Manage and executes user written application logic
- Profile Manager
  - Manages user and device information
- Database
  - Stores sensor data, guard rules, device profiles.
- APIs
  - Device APIs, User/Broker APIs

SensorAct VPDS



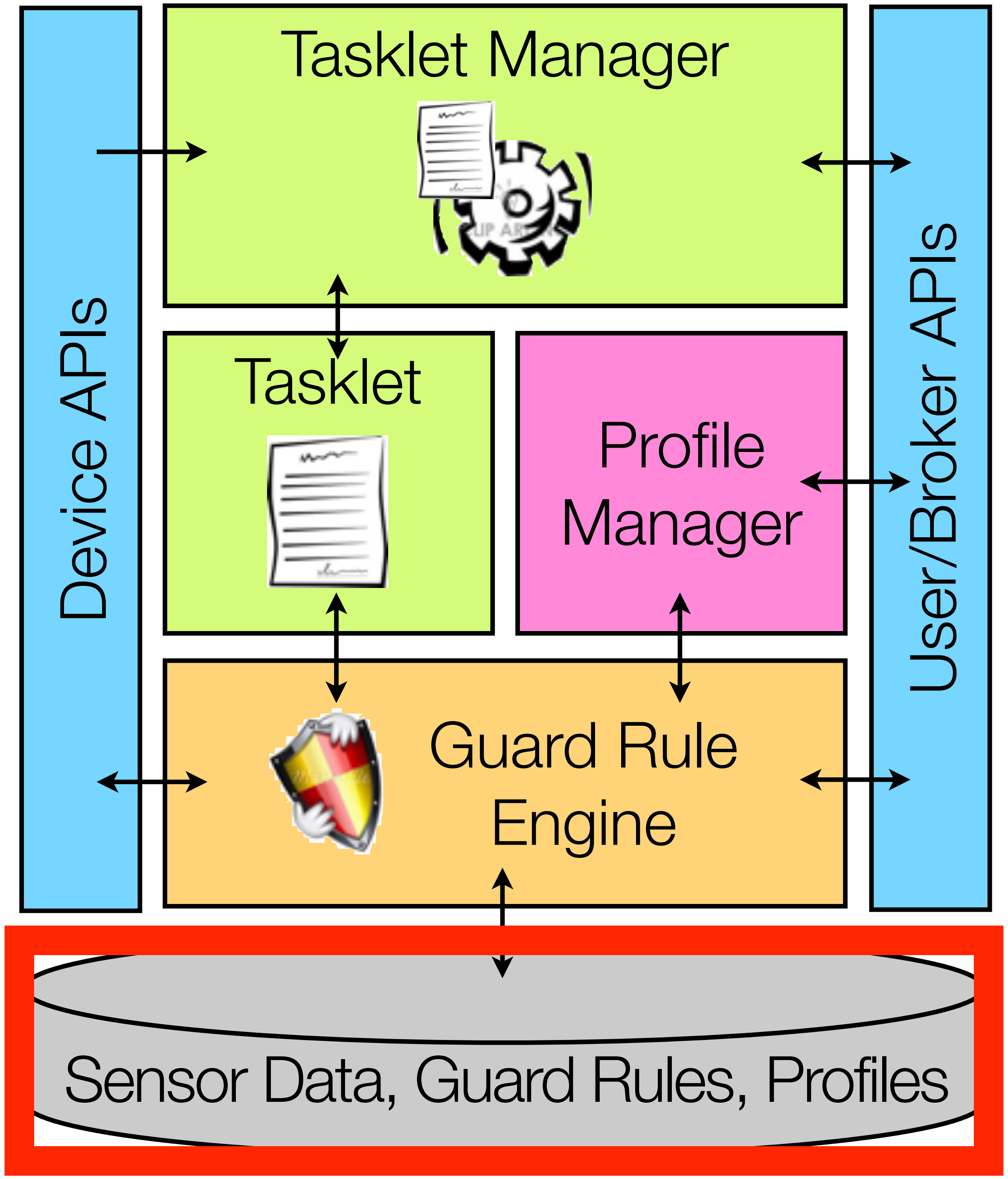
- Per user basis ensuring data ownership
- Guard Rule Engine
  - Protect privacy and security in accessing sensors and actuators
- Tasklet Manager
  - Manage and executes user written application logic
- Profile Manager
  - Manages user and device information
- Database
  - Stores sensor data, guard rules, device profiles.
- APIs
  - Device APIs, User/Broker APIs

SensorAct VPDS



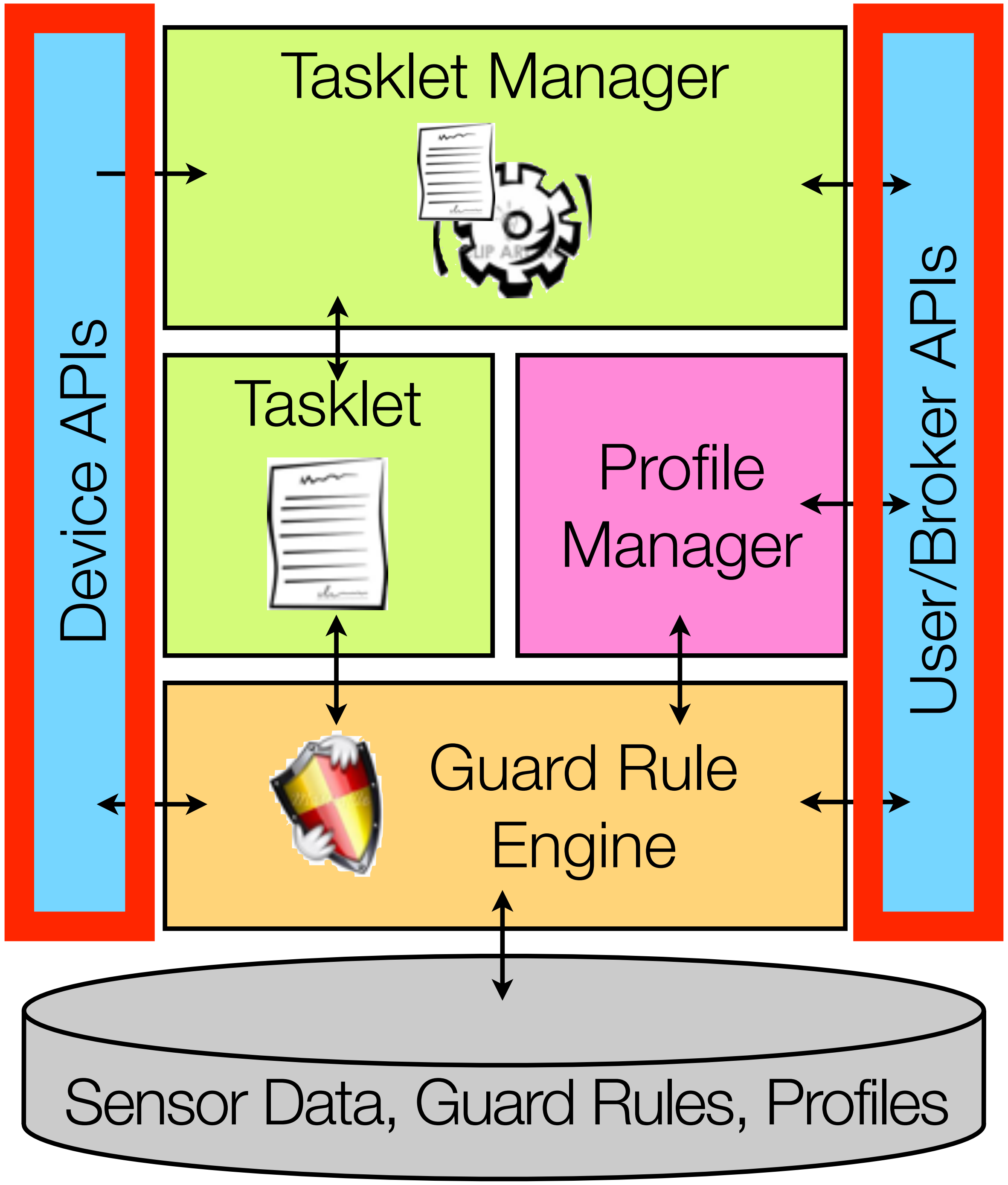
- Per user basis ensuring data ownership
- Guard Rule Engine
  - Protect privacy and security in accessing sensors and actuators
- Tasklet Manager
  - Manage and executes user written application logic
- Profile Manager
  - Manages user and device information
- Database
  - Stores sensor data, guard rules, device profiles.
- APIs
  - Device APIs, User/Broker APIs

SensorAct VPDS



- Per user basis ensuring data ownership
- Guard Rule Engine
  - Protect privacy and security in accessing sensors and actuators
- Tasklet Manager
  - Manage and executes user written application logic
- Profile Manager
  - Manages user and device information
- Database
  - Stores sensor data, guard rules, device profiles.
- APIs
  - Device APIs, User/Broker APIs

SensorAct VPDS

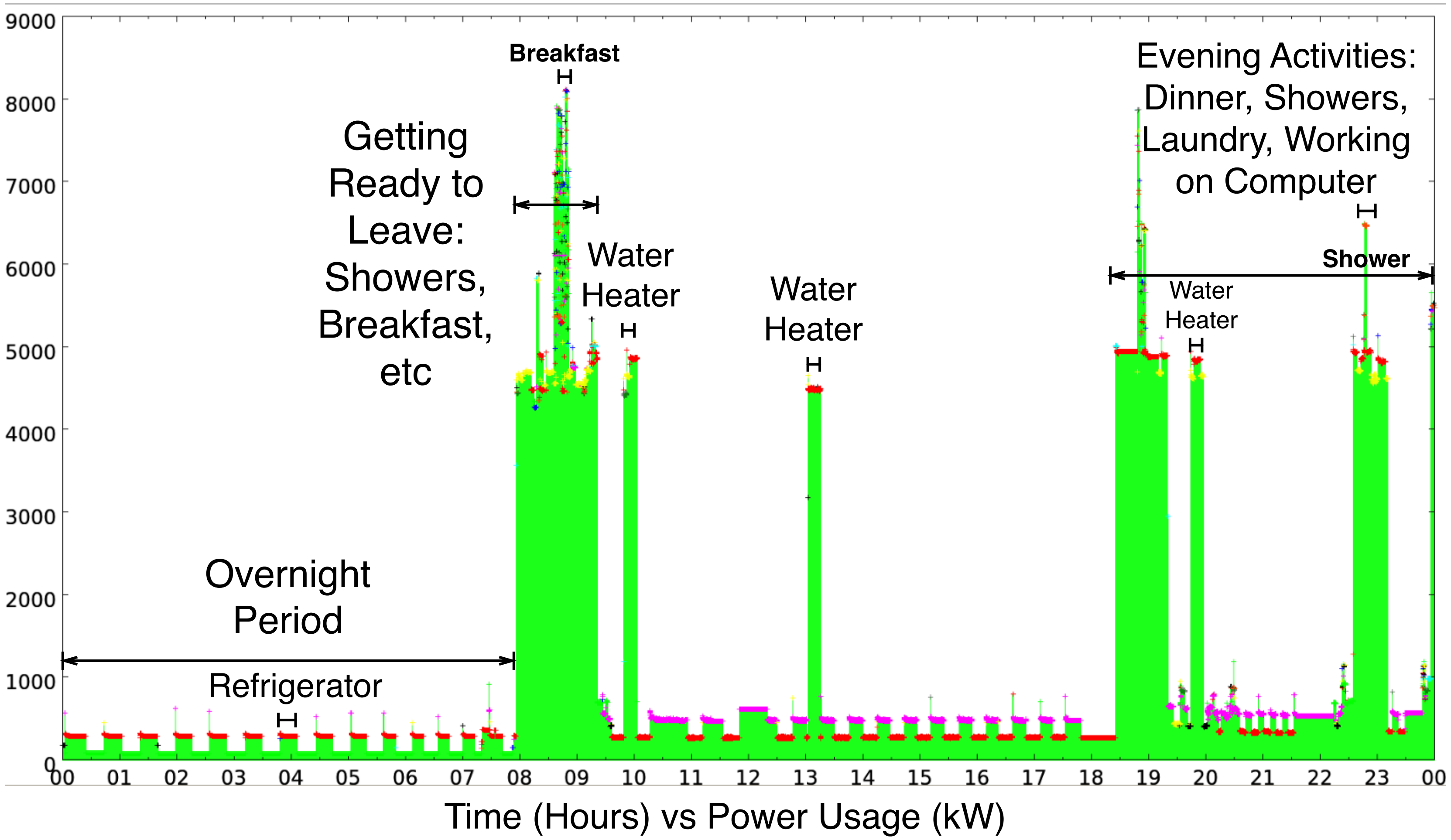


- Per user basis ensuring data ownership
- Guard Rule Engine
  - Protect privacy and security in accessing sensors and actuators
- Tasklet Manager
  - Manage and executes user written application logic
- Profile Manager
  - Manages user and device information
- Database
  - Stores sensor data, guard rules, device profiles.
- APIs
  - Device APIs, User/Broker APIs

Privacy & Security: Problem

- Traditional Solutions to Privacy
  - All or nothing mechanisms based on user authentication level

- Selective Sharing & Behavioral Privacy
  - Share only what users want with fine-grained control over what, how much, when, where, with whom
  - Not just anonymity: User identity often required by applications
  - Protecting information that can be inferred from raw sensor data



Privacy & Security: Problem

- Protected Actuation
  - Protect against unsafe, inadequate control by users
  - Enforcing policies on actuation
  - Examples
    - Too frequent control of A/C or heaters
    - Temperature control outside of permitted range

Unsafe, Inadequate Control

Not Conforming to Policies

Privacy & Security: SensorAct Approach

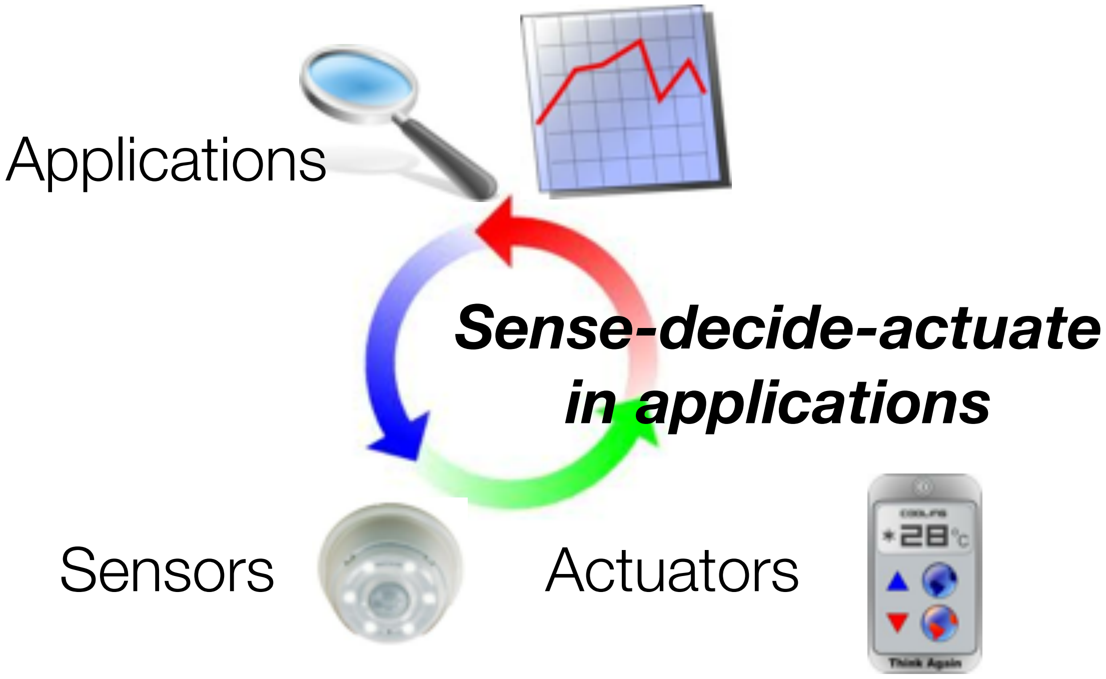
- Guard Rules for Sensors and Actuators
  - Users define necessary rules for sensor data or actuators
  - Fine-grained access control to sensor data based on user identity, location, time, and sensor values.
  - Context-based suppression and data obfuscation
  - Rich conditions to express various context

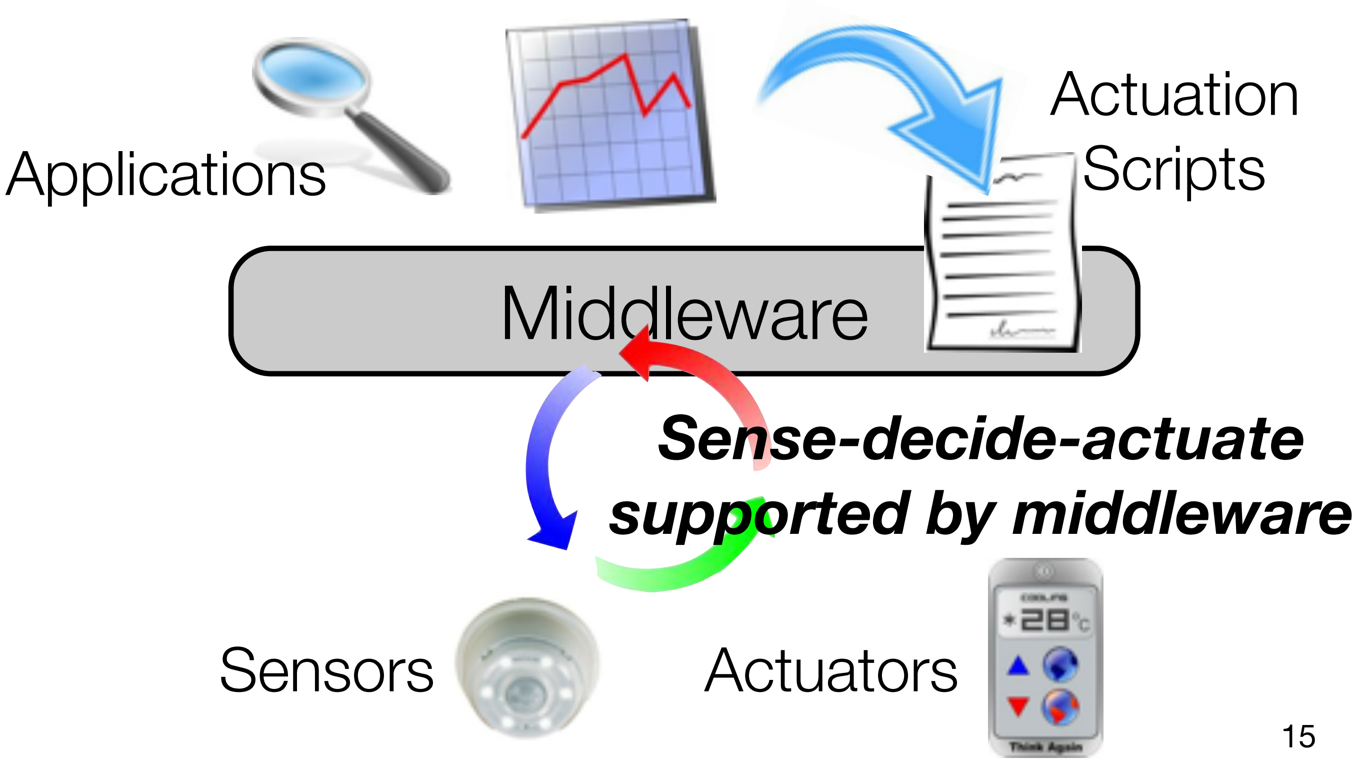
```
{
 “NAME”: “RuleForBob”,
 “DESCRIPTION”: “Allow bob@SensorAct.edu to access BuildingA
                 data with values ranging from 60 to 100
                 and collected during July 2012(PDT)”,
 “TARGET_OPERATION”: “READ”,
 “PRIORITY”: 1,
 “CONDITION”: “USER.email == bob@SensorAct.edu
               && LOCATION_TAG == BuildingA
               && TIME >= 1341126000 && TIME < 1343804400
               && VALUE >= 60 && VALUE <= 100”,
 “ACTION”: “ALLOW”
}
```

Guard Rule Example

Lightweight Tasking Framework

- Traditional Query Interface
  - Application control logic resides outside framework
  - Latency in actuation: heavy data traffic
- Lightweight Tasking Framework
  - Closes the control loop inside the framework
  - Enables audit of what information leaves the framework





Lightweight Tasking Framework

- Various Task Scheduling
  - One-shot
  - Periodic
  - Event driven
  - Periodic and event driven

Tasklet Request

```
{
  "NAME": "Monitor_AC",
  "PARAMS": {
    "T1": "Mickey:Room1:Tempr:1",
    "A1": "Mickey:Room1:AC:1",
    "MINS": 5,
    "LIMIT": 30
  }
  "INPUT": {
    "TIMER1": "[0 0/2 10-18 * *]"
  }
  "WHEN": "TIMER1",
  "EXECUTE": "[monitor_ac.lua]"
}
```

Tasklet Example

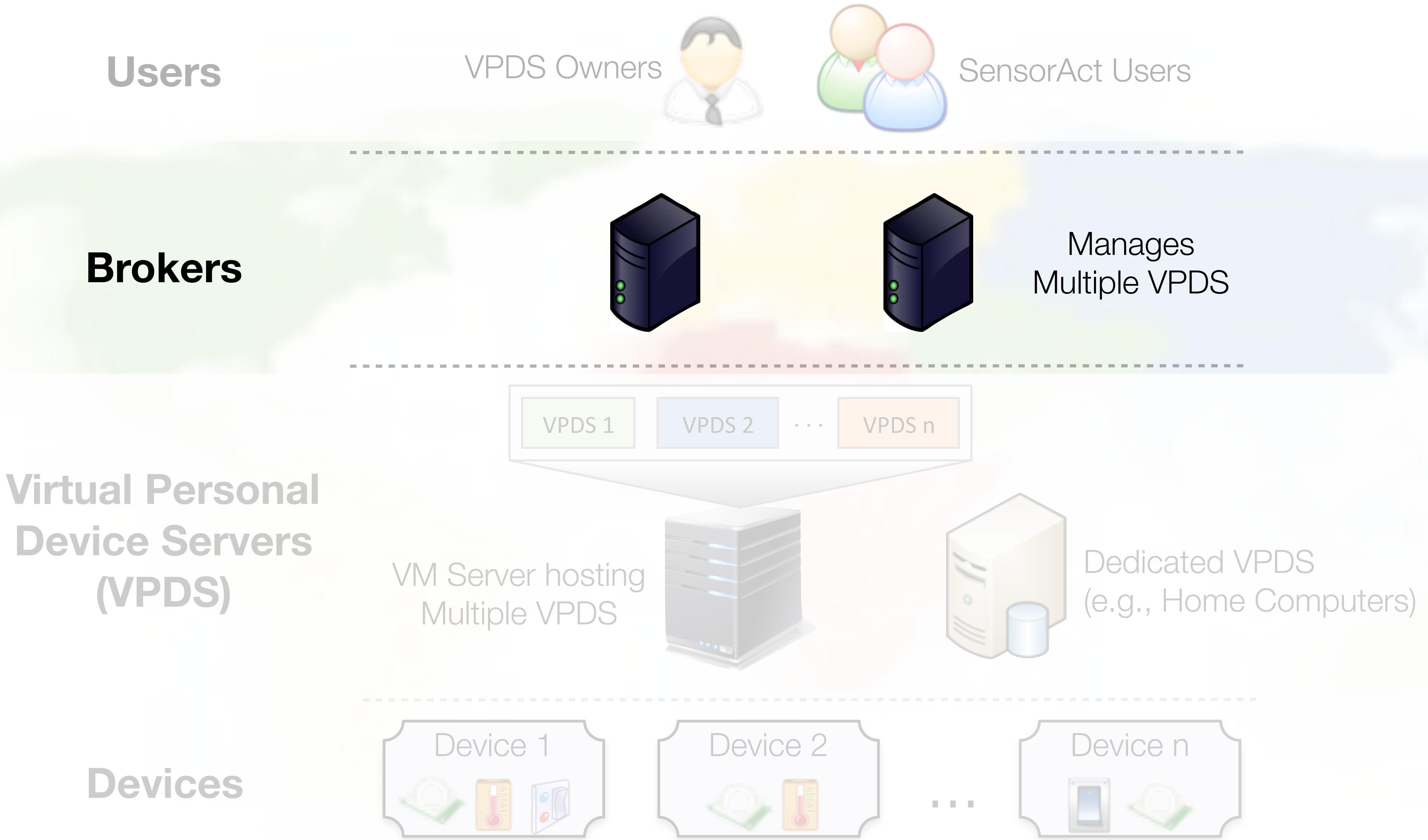
```
-- Reads sensor readings and
-- actuate appliance accordingly

-- epoc MINS minutes before
epocNmin = os.time() - (60*MINS)

-- read MINS minutes avg value
avgTr = VPDS:readAvg(T1,epocNmin)

-- Check and turn ON
if avgTr>LIMIT then
VPDS:write(A1,VPDS:TURNON)
end
```

SensorAct Brokers



Users

VPDS Owners

SensorAct Users

Brokers

Manages
Multiple VPDS

VPDS 1

VPDS 2

· · ·

VPDS n

Virtual Personal
Device Servers
(VPDS)

VM Server hosting
Multiple VPDS

Dedicated VPDS
(e.g., Home Computers)

Devices

Device 1

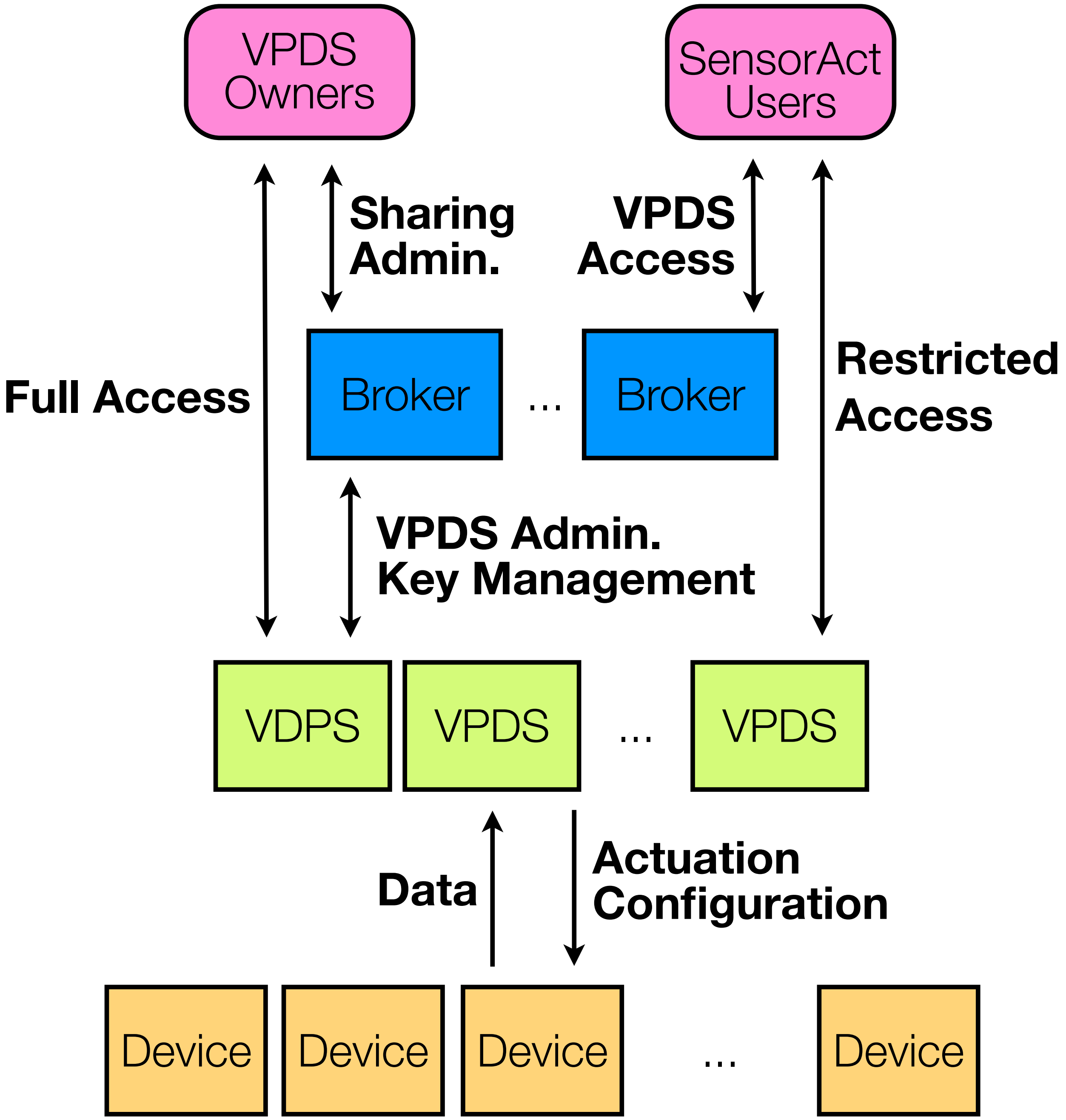
Device 2

. . .

Device n

SensorAct Brokers

- Brokers
  - VPDS is “personal” data store. Need coordination when sharing.
  - Mediators b/w users/applications and VPDS.
  - User registration/authentication, discovering VPDS
  - VPDS takes responsibility of enforcing privacy and security of data

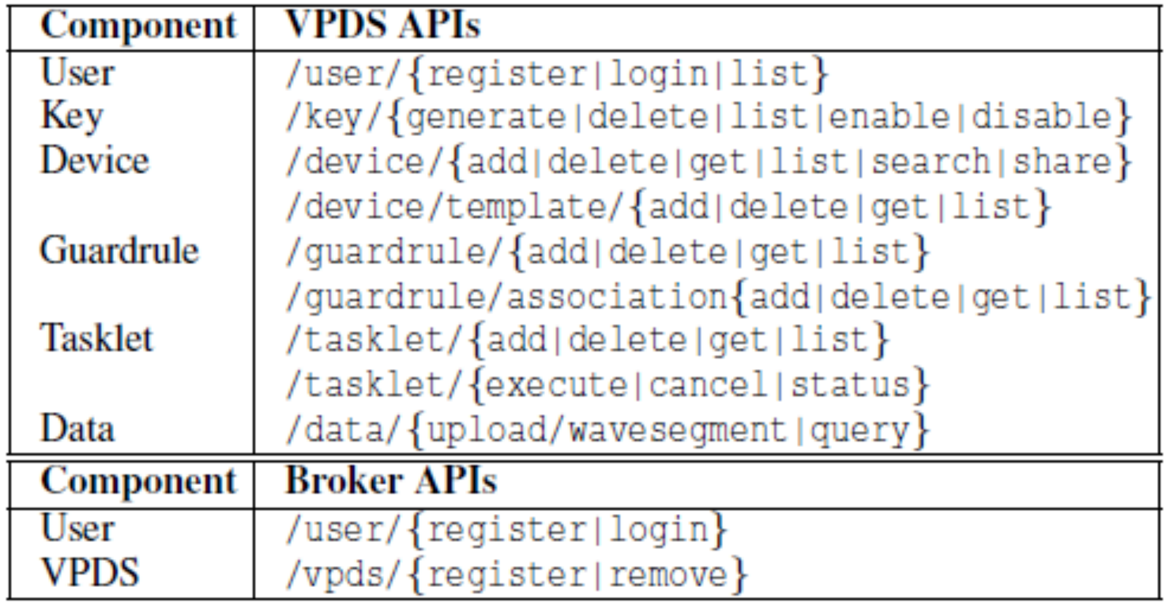


Implementation

- RESTful APIs
- Java - Play Framework
- MongoDB
- Quartz
- Lua
- Web 2.0
- Open source

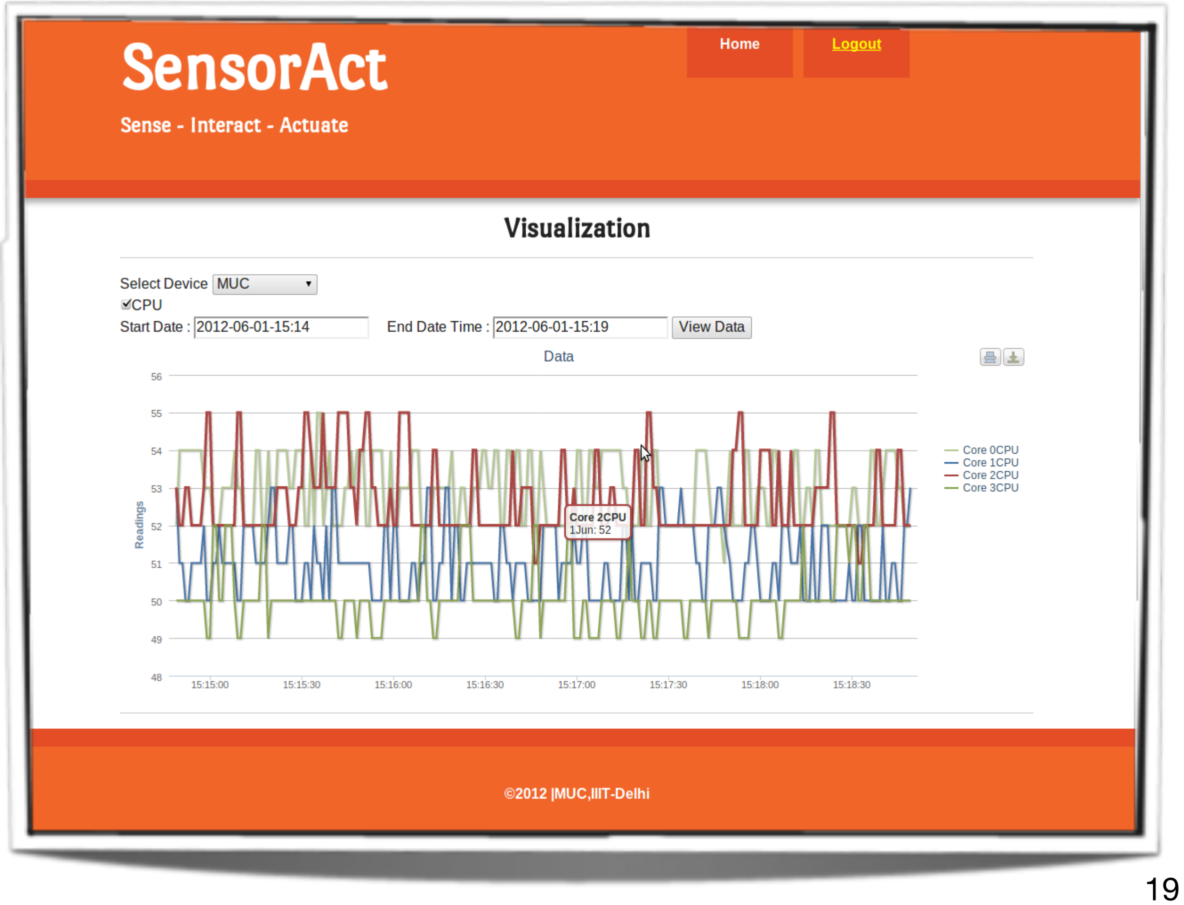
github
SOCIAL CODING

- https://github.com/iiitd-ucla-pc3



Component	VPDS APIs
User	/user/{register|login|list}
Key	/key/{generate|delete|list|enable|disable}
Device	/device/{add|delete|get|list|search|share}
	/device/template/{add|delete|get|list}
Guardrule	/guardrule/{add|delete|get|list}
	/guardrule/association{add|delete|get|list}
Tasklet	/tasklet/{add|delete|get|list}
	/tasklet/{execute|cancel|status}
Data	/data/{upload/wavesegment|query}

Component	Broker APIs
User	/user/{register|login}
VPDS	/vpds/{register|remove}



SensorAct

Sense - Interact - Actuate

Visualization

Select Device MUC

☑CPU

Start Date : 2012-06-01-15:14 End Date Time : 2012-06-01-15:19 View Data

©2012 |MUC,IIIT-Delhi

Preliminary Microbenchmark

- Simulated our largest deployment (400 dorm rooms)
- On a Laptop (2.3 GHz Intel i7, 8GB RAM)

Data Collection

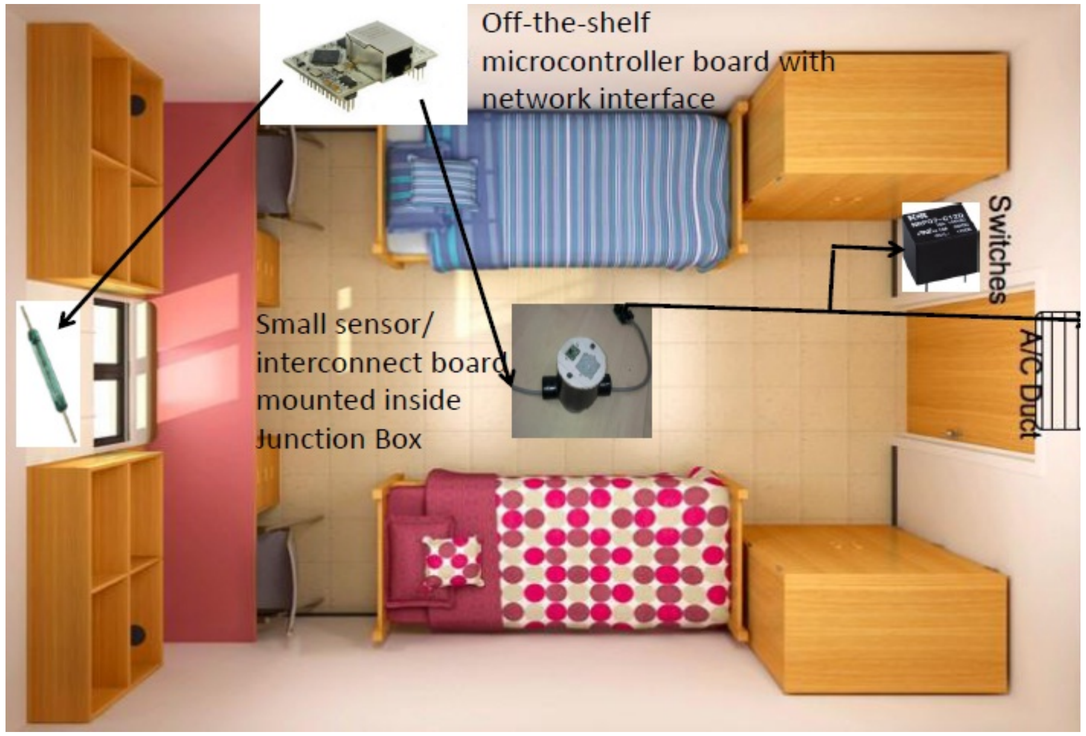
# Devices	# Channels	Sampling Rate	Publish Rate	CPU	RAM
400	10	1 Hz	Every 10 secs	14.6%	3.7%

Query Processing

Querier	# WaveSegs	# Guard Rules	Processing Time
User	90 (10 readings each)	3	3.3 secs
Owner		0	11 ms

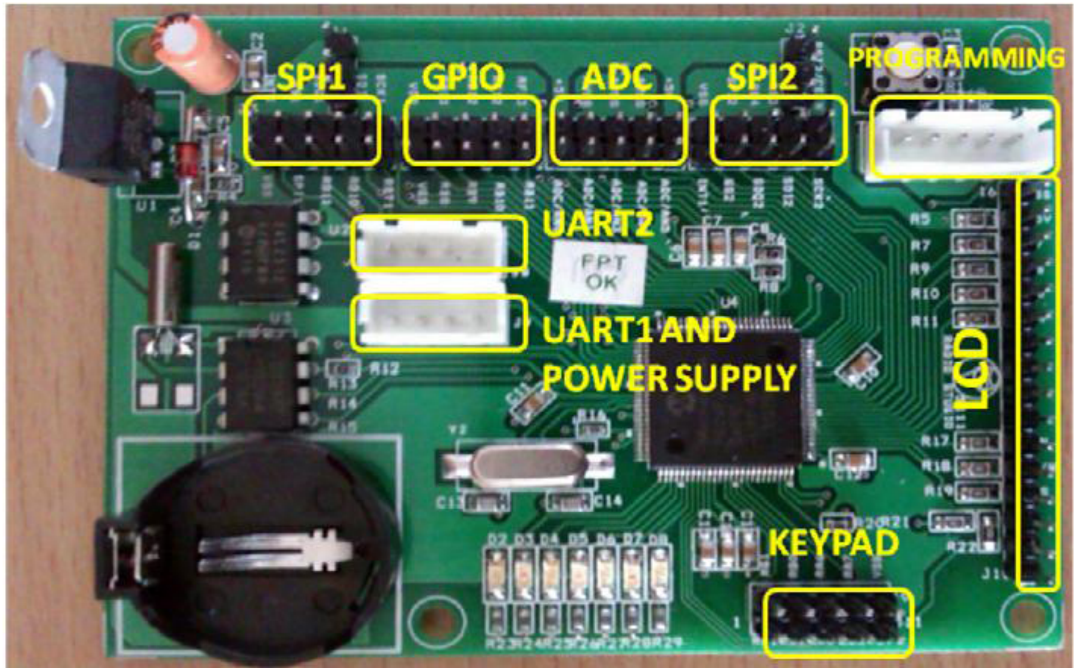
Ongoing Deployment @ IIIT-Delhi

- 400+ student dorm rooms
- Temperature, motion, light and door status
- Smart electricity/water meter across the campus
- Integrates with commercial HVAC systems


Off-the-shelf microcontroller board with network interface
Switches
A/C Duct
Small sensor/ interconnect board mounted inside Junction Box

Wi-Fi

Ethernet



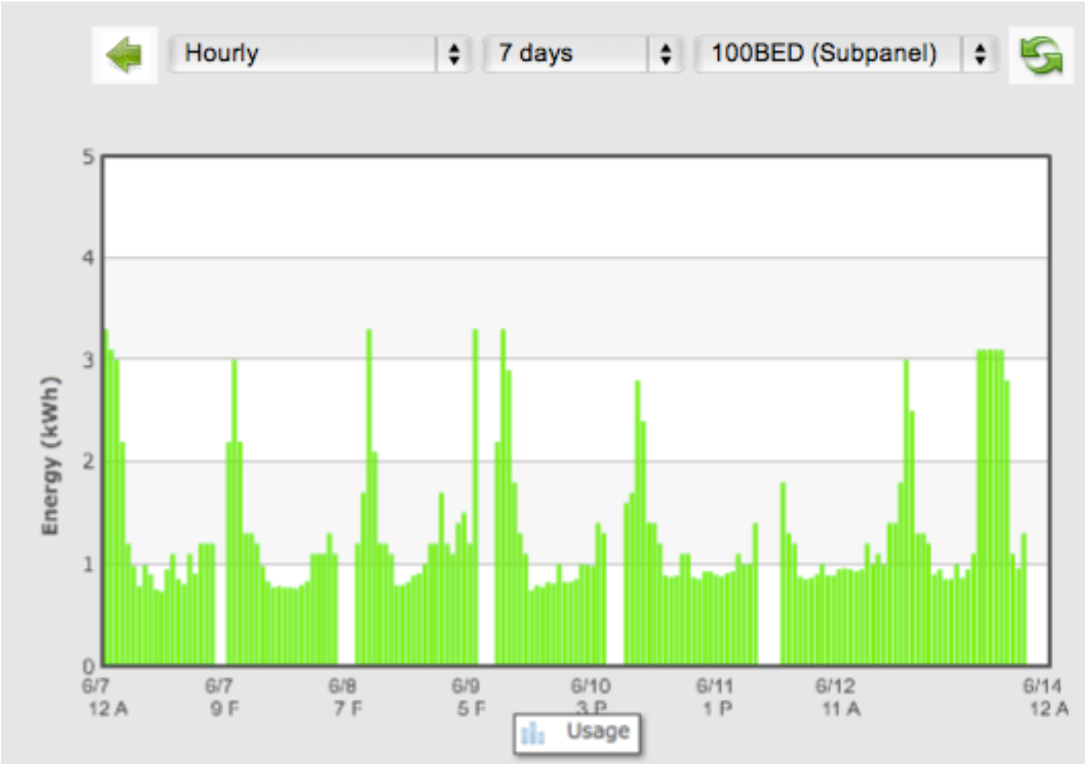
Ongoing Deployment @ UCLA

Interior Sensor Kit

Circuit Panel Kit

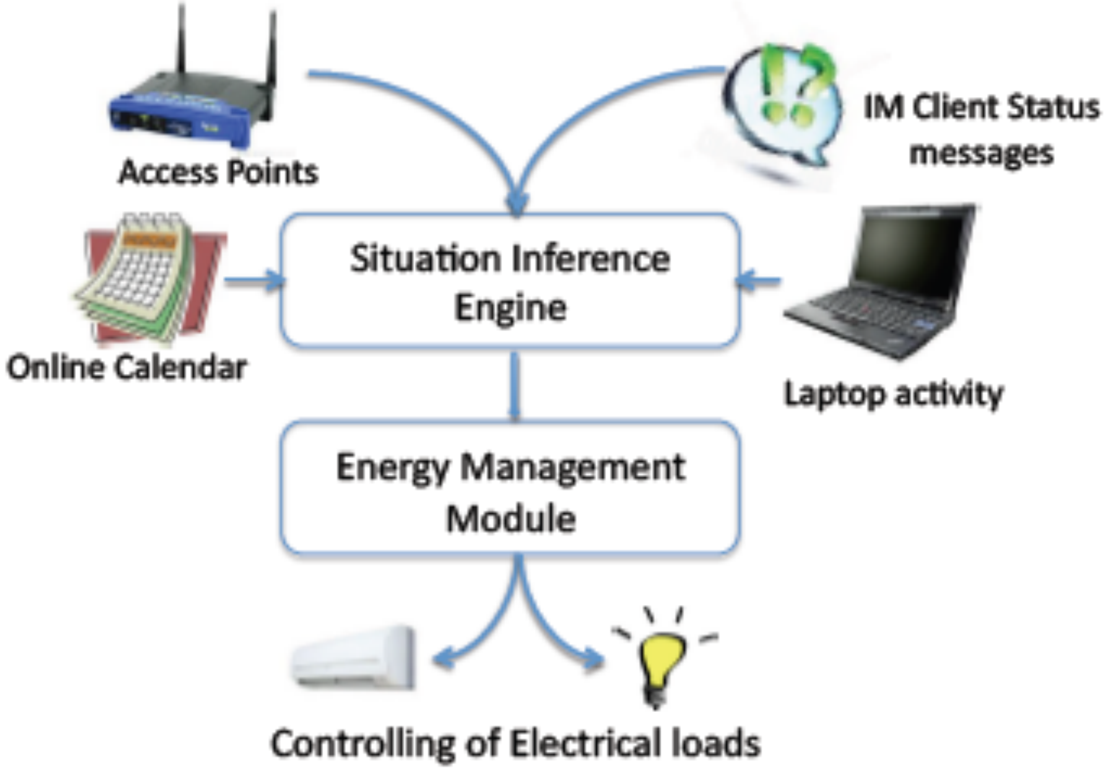
Main Water Flow

- Wet & Dry Lab deployment
  - Main panel, circuit, outlet level electricity
  - Meter & outlet level water
  - Occupancy
  - Ambient (temperature, light, and pressure)
- Residential deployment
  - Main panel, circuits, outlets
  - Main water



Ongoing Deployment @ IBM

- SoftGreen Testbed
  - Occupancy detection using existing soft sensors
- Software service collects context source status
  - Wi-Fi, Ethernet, CPU
  - IM status, online calendar
- 100+ volunteers
- For ground truth
  - 70+ readers
  - Temperature, motion and light sensors
  - 802.15.4 tags for identity



Access Points
IM Client Status messages
Online Calendar
Situation Inference Engine
Laptop activity
Energy Management Module
Controlling of Electrical loads

Image Courtesy: IBM Research, India

Conclusion & Future Work

- Experiences from isolated building and campus to international testbed across organizations
- System need to be more responsible
  - A Tiered Distributed Architecture, Sensor and Actuator Guard Rules, Lightweight Tasking framework
- Open to public, work in progress
  - On-going deployment
  - Naming and resource discovery
  - Guard Rules with behavioral privacy
  - Seamless integration of devices
  - Federated Brokers
  - Usability Study

Thank You