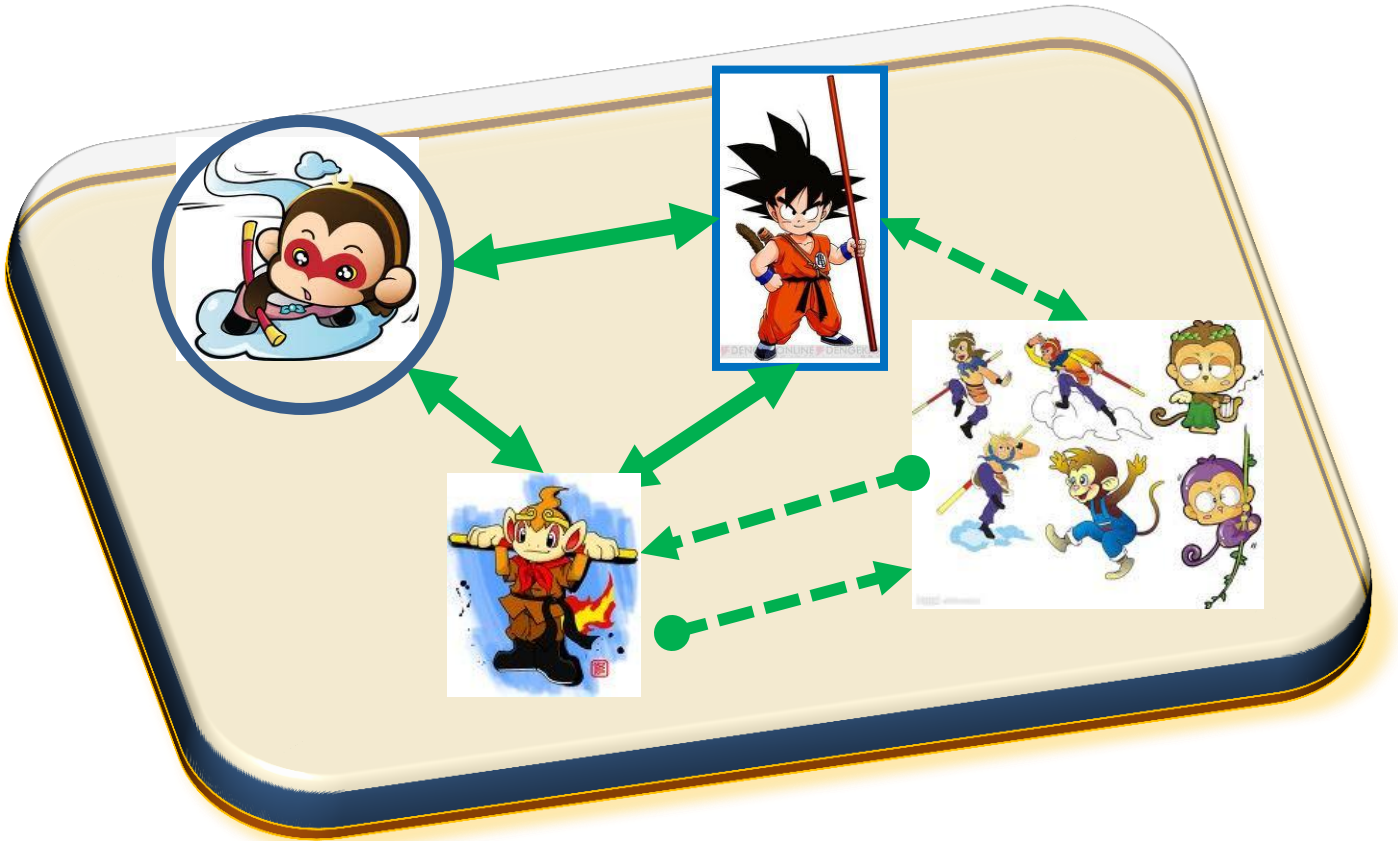
Building Intelligent Middleware for Large Scale M2M Systems

Kwei-Jay Lin*, Chi-Sheng Shih, Yu-Chung Wang, Jane Y. Hsu

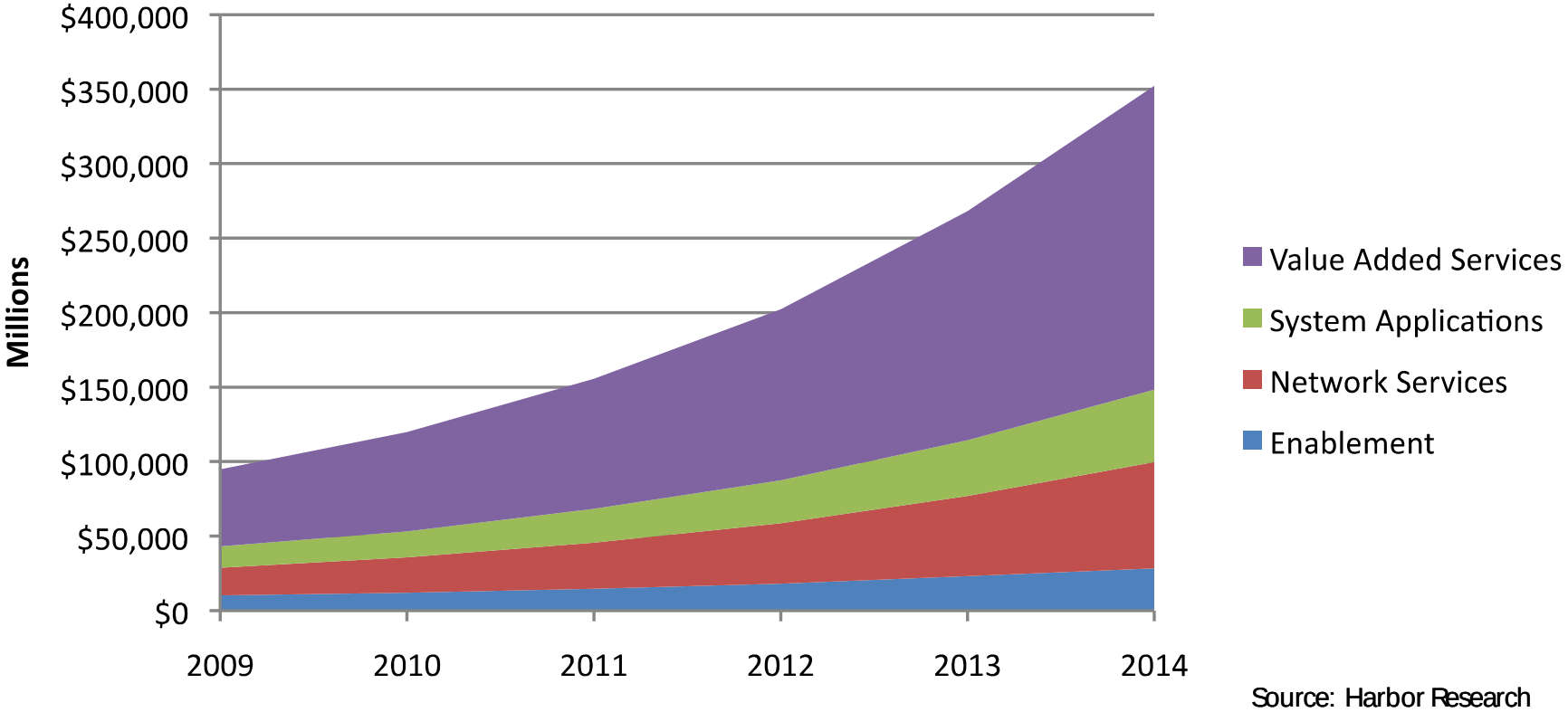
*National Taiwan University, * University of California, Irvine*



M2M Future (*by Harbor Research, Inc.*)

Intel-NTU Center

- M2M future is on many new services to be deployed
- These services must be easy to develop, to deliver and to defend
- We need a powerful and intelligent platform to support future M2M



Source: Harbor Research

Embedding Intelligence in Future M2M

Intel-NTU Center

- "The next chapter in the M2M arena will be driven by cloud platforms, intelligent devices and back office systems that seamlessly integrate with one another and thereby unlock the full potential of smart connected devices.
- Uninformed human activity in customer support will be replaced by embedded intelligence in devices and smart systems.
- The complexities of disparate networks, devices and hardware will be removed thus reducing time to market and overall costs and risks for custom application development. All of this will be done with enterprise solutions engineered for deployment on the world's leading wireless networks."

"Next Generation Platform Innovation In M2M", whitepaper, Harbor Research, Inc. March 2011 [http://www.harborresearch.com/HarborContent/whitepapers.html]

INC NSC

Application Development

- “Having nearly reached the saturation point with traditional enterprise application development and deployment, professional IT services firms are now turning their attention to *non-IT devices capable of being connected to a network and integrated into cloud services*.
- While the IoT (Internet of Things) is a huge opportunity with many new market entrants who are predicting that enormous volumes of connected devices are “just around the corner”, … the biggest challenge will be finding enough new technology and industry fluent players to develop all the applications required to inform this expanding opportunity. “

“Who's Developing All The Applications For 50 Billion Devices?”, *Harbor Research, Inc.* [http://harborresearch.com/_blog/Smart_Business_Blog/post/Who's_Developing_All_The_Applications_For_50_Billion_Devices/]

Technology Challenge (I): Delivery

Intel-NTU Center

Enhanced Service Delivery Platforms (SDP)

- Connecting and managing networkable devices, has traditionally been a problematic area for customers.
  - In the past, it took several months to get a device network certified. Once the device was connected, there was often little visibility into how it was performing on the network, as well as, limited back end control.
- SDPs provide configuration services, provisioning, SIM management and reporting, billing, upgrades, and basic asset-related application services.

"Next Generation Platform Innovation In M2M", whitepaper, Harbor Research, Inc. [http://www.harborresearch.com/HarborContent/whitepapers.html]

INC NSC

Technology Challenge (II): Development

Intel-NTU Center

Application Development & Delivery

- Today, IoT applications are cumbersome and complex to develop.
  - Whether the application is developed by the company deploying it or a third party, they are very custom in nature and often configured for the environment in which they operate.
  - Application development for connected devices today entails a very high level of engineering complexity
- Smart system application development has focused primarily on … technology for provisioning, management and billing for connected devices – … [but needs] application development and application services delivery.

"Next Generation Platform Innovation In M2M", whitepaper, Harbor Research, Inc. [http://www.harborresearch.com/HarborContent/whitepapers.html]

INC NSC

Introducing *Wu-Kong*

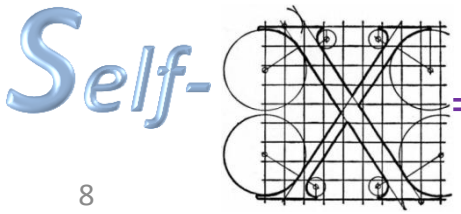
Intel-NTU Center

- The Wu-Kong project is to build an *Intelligent Virtual Middleware* for M2M, that can
  1. recognize heterogeneous devices in connection, user context and needs;
  2. configure and transform devices into service components;
  3. adapt and distribute app codes to achieve the best result;
  4. do all of the above via remote access to sensors.
- Another goal is to use the Wu-Kong middleware as the foundation of next generation M2M application development for constructing
  1. high level code that is sensor- and context-independent
  2. dynamic code that evolves with sensor capabilities and missions

Goal for Current Stage: Self-X M2M

Intel-NTU Center

- The objective is "zero-cost deployment" (i.e., cost refers to human effort)
  - Problem: manual effort is often the bottleneck in large-scale deployment and long-term sustainability of M2M systems in the field.
- The project is to provide an intelligent M2M framework
  - Self-X stands for self-configuration, -protection, -healing, and -optimization for sensor nodes and gateways.
- Based on user-defined policy and context, the proposed platform services automatically perform
  - sensor node and network configuration,
  - application deployment,
  - fault handling, and
  - system reconfiguration.



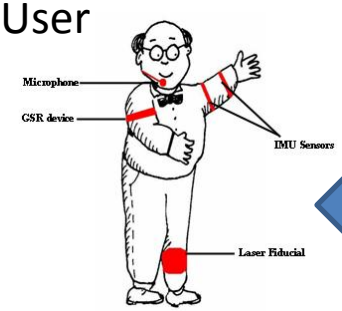
= configuration, protection, healing, optimization

INC NSC

Wu-Kong: Who's Who

Intel-NTU Center

User



App (on Cloud)

Master

Infrastructure Provider

Gateway

Sensor Nodes

Communication Media

- Human
- Broadband
- Wireless

User (naïve and demanding)
- Sends the request (via some user interface) to apps
- Defines context and high-level policy

Apps (running on user device and/or cloud)
- Interact with user
- Have access to unlimited computing power and intelligence

Nodes (sensor devices)
- Physical world sensing and actuating
- Need only limited computing power to sense/send data

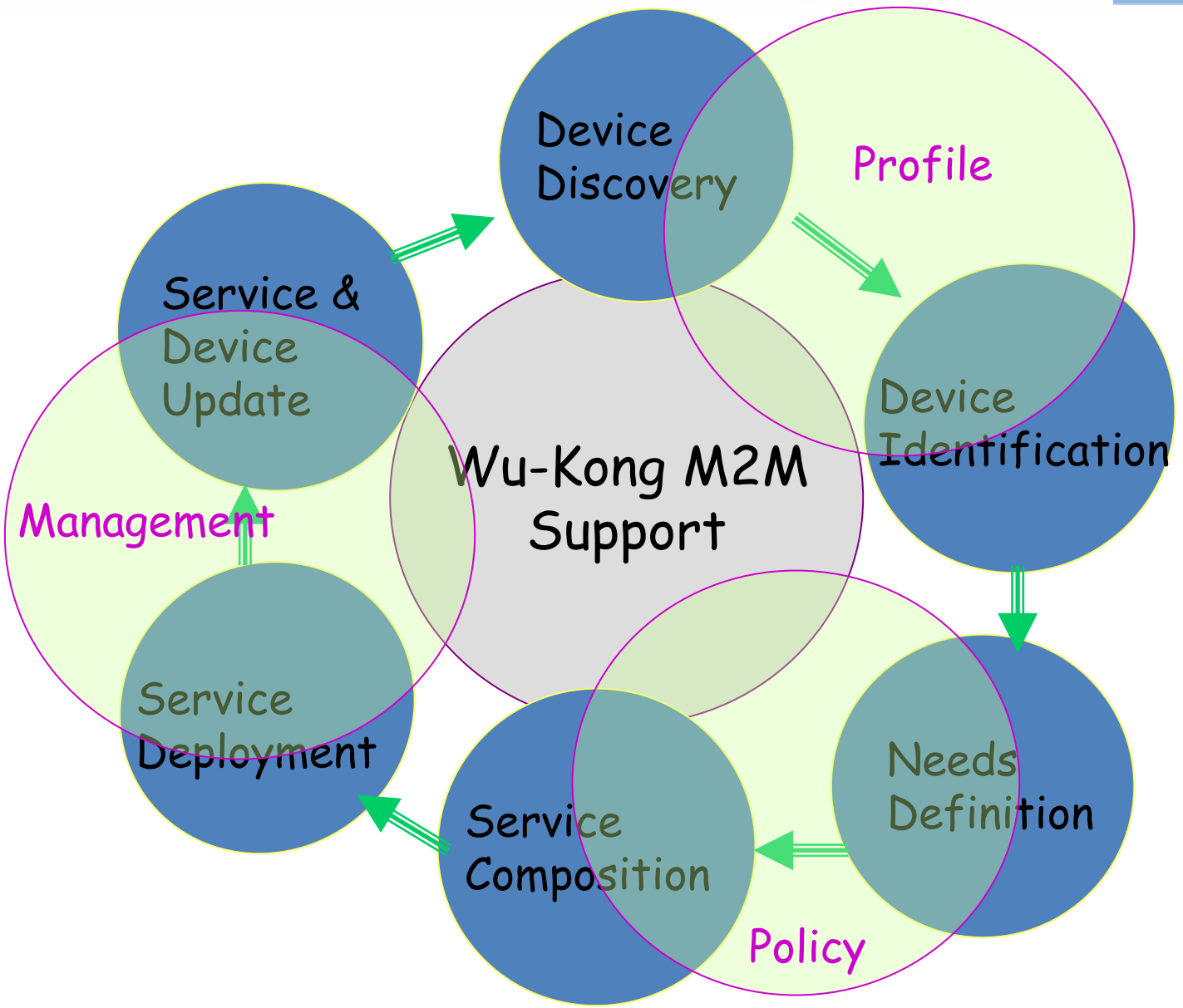
Master (coordinator for WSN)
- Has computing power to make coordination decisions
- Connects WSN to outside (as one of the *Gateways*)

Gateway (Cohort for Master)
- provides extra computing/connection
- provides backup coordination

Infrastructure Provider
- provides management support
- needs provision and self-X control

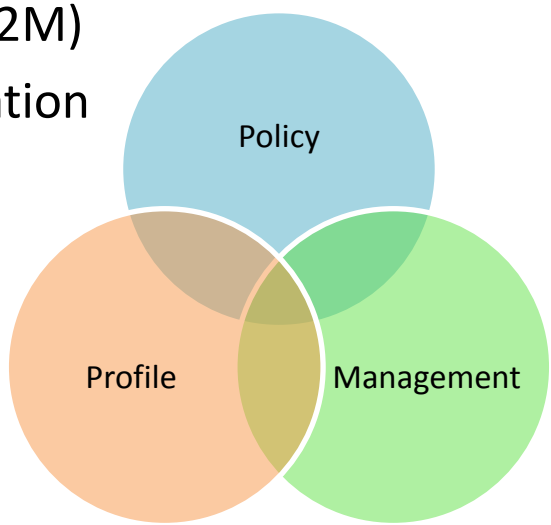
INC NSC

Wu-Kong M2M Deployment Lifecycle



Wu-Kong Technology: Tri-Framework

- Profile Framework (abstraction for heterogeneous nodes)
  - Device classes (capabilities): discover, share and control
  - Heterogeneous and virtual sensor sharing
- Policy Framework (adaption for M2M context)
  - Configuration decision and constraint optimization
  - Configuration <-> fault tolerance, security, trust
- Management Framework (embedding intelligence in M2M)
  - Sensor to Master to cloud distribution and coordination
  - Tools for application access and control anywhere, anytime



Profile Framework

- Abstraction for heterogeneous sensor nodes so that Master can discover, access and control them
- Related work: uPnP (2006), DPWS (2006), WSDD (2008), TinyDB (2005), sMAP (SenSys 2010)
  - uPnP provides a service-oriented architectural framework for self-configuring, self-describing devices on top of TCP/IP, HTTP and SSDP
  - WSDD defines a lightweight version of DPWS using Web service framework
  - sMAP defines a RESTful interface for universal device access
- Complexity and machine self-automated are critical for Wu-Kong
- We want to implement a general and extensible profile framework
  - **"BIOS" for sensors**: (*from wiki*) "When the PC starts up, the first job for the BIOS is to initialize and identify system devices such as the video display card, keyboard and mouse, hard disk drive, optical disc drive and other hardware. The BIOS then locates boot loader software held on a peripheral device (designated as a 'boot device'), such as a hard disk or a CD/DVD, and loads and executes that software, giving it control of the PC."

Wu-Kong Profile Framework

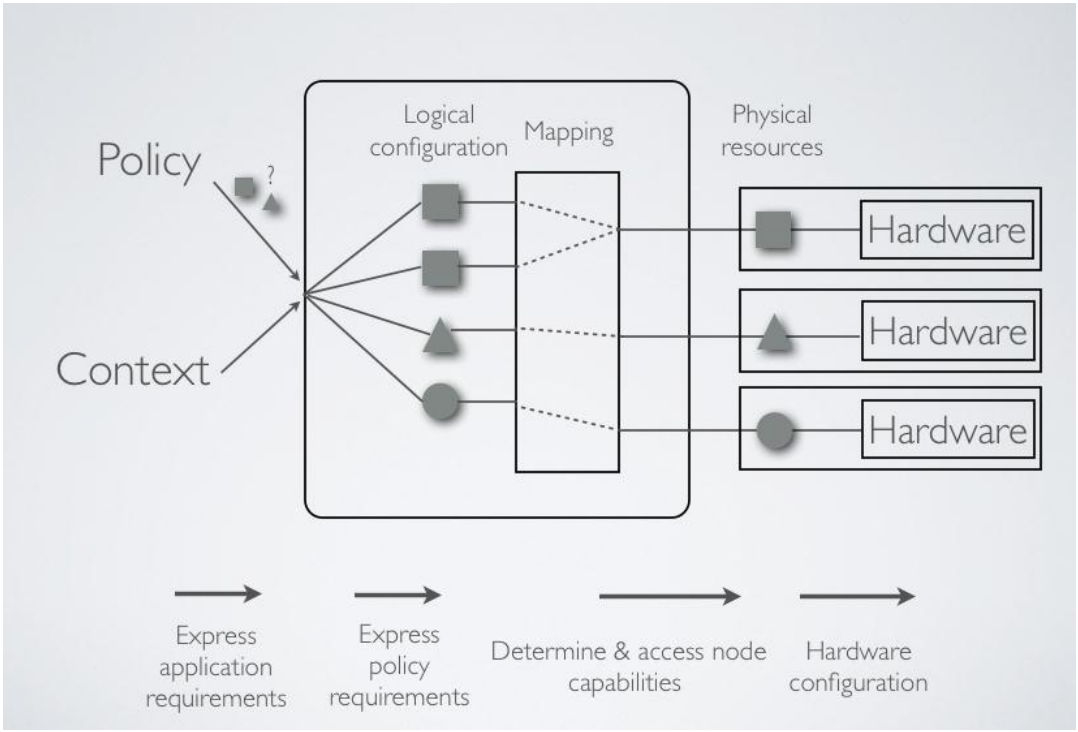
- To configure the M2M network, Master needs to be able to determine what sensor resources are available.
- Three phases:
  1. Determine what devices are on the network (*discovery*)
  2. Determine what those devices can do (*profile*)
  3. Determine what those devices should do (*policy*)
- We separate the discovery of nodes from the profile because
  - discovery may depend on the scenario
  - discovery may depend on the specific network technology used
  - handful of nodes at home vs thousands for environmental monitoring
- After Master has "discovered" devices and queried them, each device profile provides:
  - A way to determine what resources are available on a device
  - A way to access to those resources

Wu-Kong Profile Framework

- In Wu-Kong, each profile represents a single capability on a device.
  - Multiple profiles, possibly at the same time, can be present on a single device.
  - *E.g.* two temperature sensors with different characteristics (accuracy, power, etc), results in two "Temperature Sensor" profiles with different content
- Logical vs. Physical sensor devices
  - Each physical sensor device can be used to serve one or more profiles
  - Each application is designed to run with some logical devices, each defined by a profile
  - *E.g.* An application is designed to read data from a logical “temperature sensor” and a logical “light sensor”. In the target system, however, there may be a physical device with both “temperature sensor” and “light sensor” profiles.
- A profile model is defined by a data structure
  - Similar to Zigbee and S-Map (*in contrast to Z-Wave and enOcean, which define the profile as a list of service operations*)
  - Each physical device stores its profile data in a protected area in its device memory
  - Each application is compiled to produce a list of required logical devices.
  - There must be a binding between logical and physical devices (both are expressed in the well-defined profile data structure) before application execution.

Binding in the Profile Framework

- The application and context are used to produce a logical configuration of profiles
- Logical devices must be mapped to physical sensor nodes available on network with compatible profiles
- Mapping can be 1-1, n-1, 1-m, or n-m



Wu-Kong Profile Definition

- In Wu-Kong, each profile may have three different types of data:

1. A *common* profile is present on all devices, describing basic properties
   - CPU speed, Memory size, Power rating, Connectivity media, etc.
2. Some *specific* profiles are present to expose a device's specific features
   - Temperature Sensor, Humidity Sensor, AC controller, etc.
3. Master can add *extended* profile to expand a device's capability
   - e.g. integrate Motion Sensor and Light Sensor into a Sleep Sensor
   - Functionality beyond the device's original hardware design can be added by uploading function code to a device.

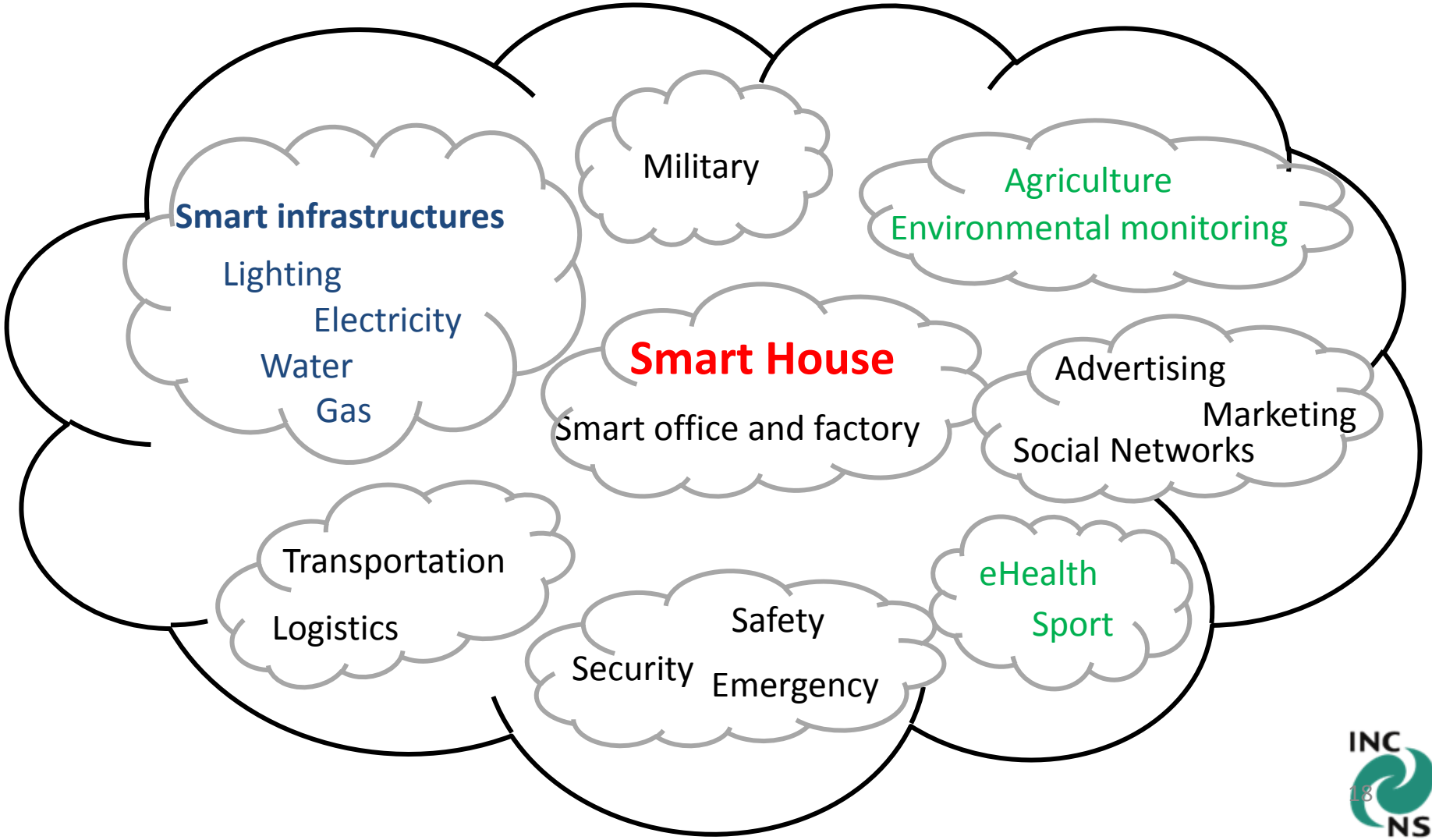
- Such a functionality is referred as an "extended" profile
  - This profile isn't (directly) tied to the node's hardware
  - It is implemented in software (dynamically loaded code)
- In this way virtual sensors can be implemented, combining data from different sensor sources into a new 'sensor'.

Policy Framework

- High level specification for M2M management
  - User or app defines an intuitive objective statement
  - Policy interpreter and configuration engine produce the detailed setup for target systems, enabling higher-level thinking/coding
  - By specifying policies declaratively and independent of actual devices, it is possible to change the behavior on-the-fly for better flexibility.
- Related work: Security (SPF), OS Policy, Ponder/Ponder2 (2001/2006), DSN (SenSys 2007), ADAE (SenSys 2010)
- Configuration and constraint satisfaction engine can take many attributes (e.g. context, fault tolerance, security, trust) into consideration for a better, more optimal performance

Wu-Kong Demo Application Areas

Intel-NTU Center



INC NSC

Demo Application - *comfort@home*

Intel-NTU Center

A smart home utility: User Comfort at Home

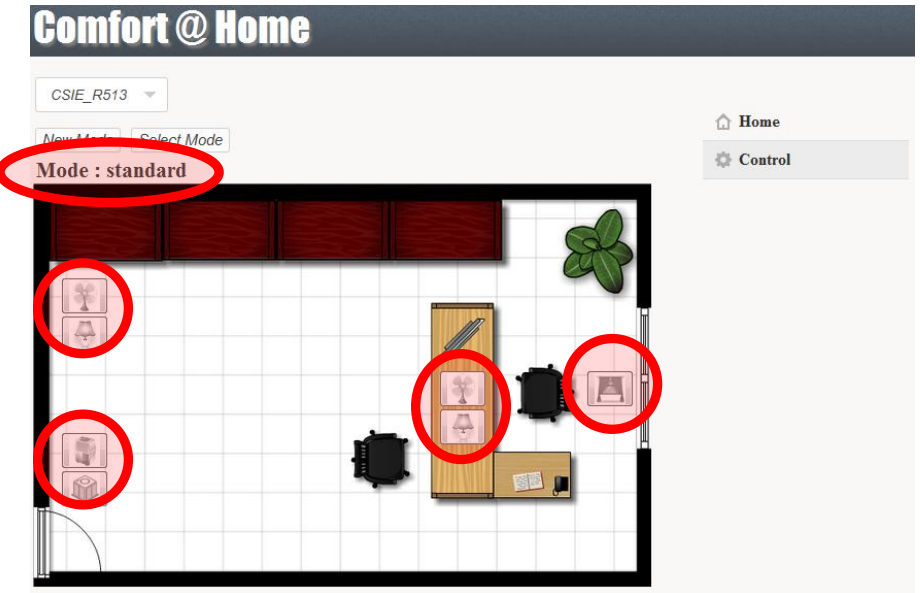
- **What to monitor**
  - human presence
  - motion
  - light
  - temperature
  - humidity, CO_2
- **What to change**
  - if someone is there, set the appropriate comfort level
  - turn lamps on and off as needed
  - turn fans on and off as needed
  - …

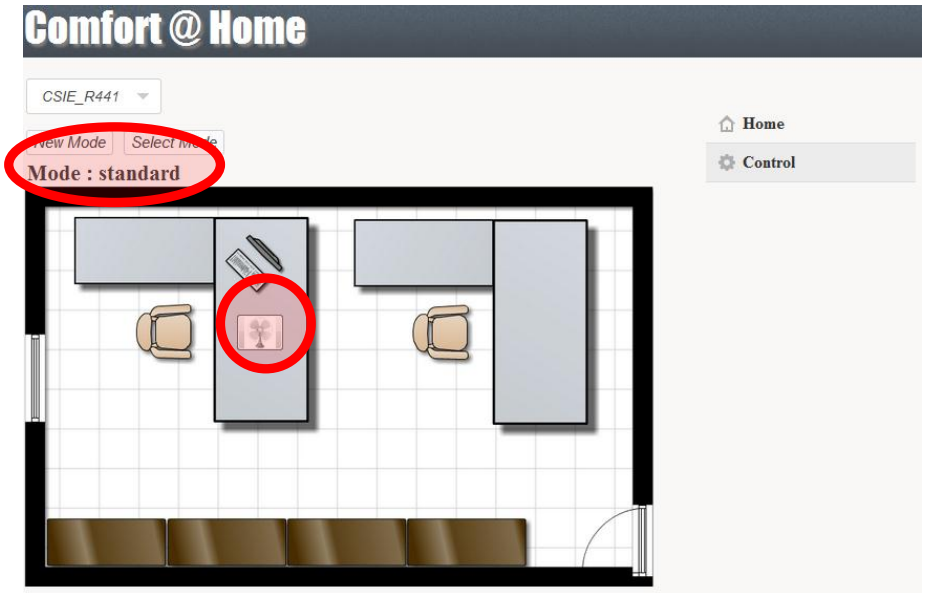
INC NSC

comfort@home – Control Page

Intel-NTU Center

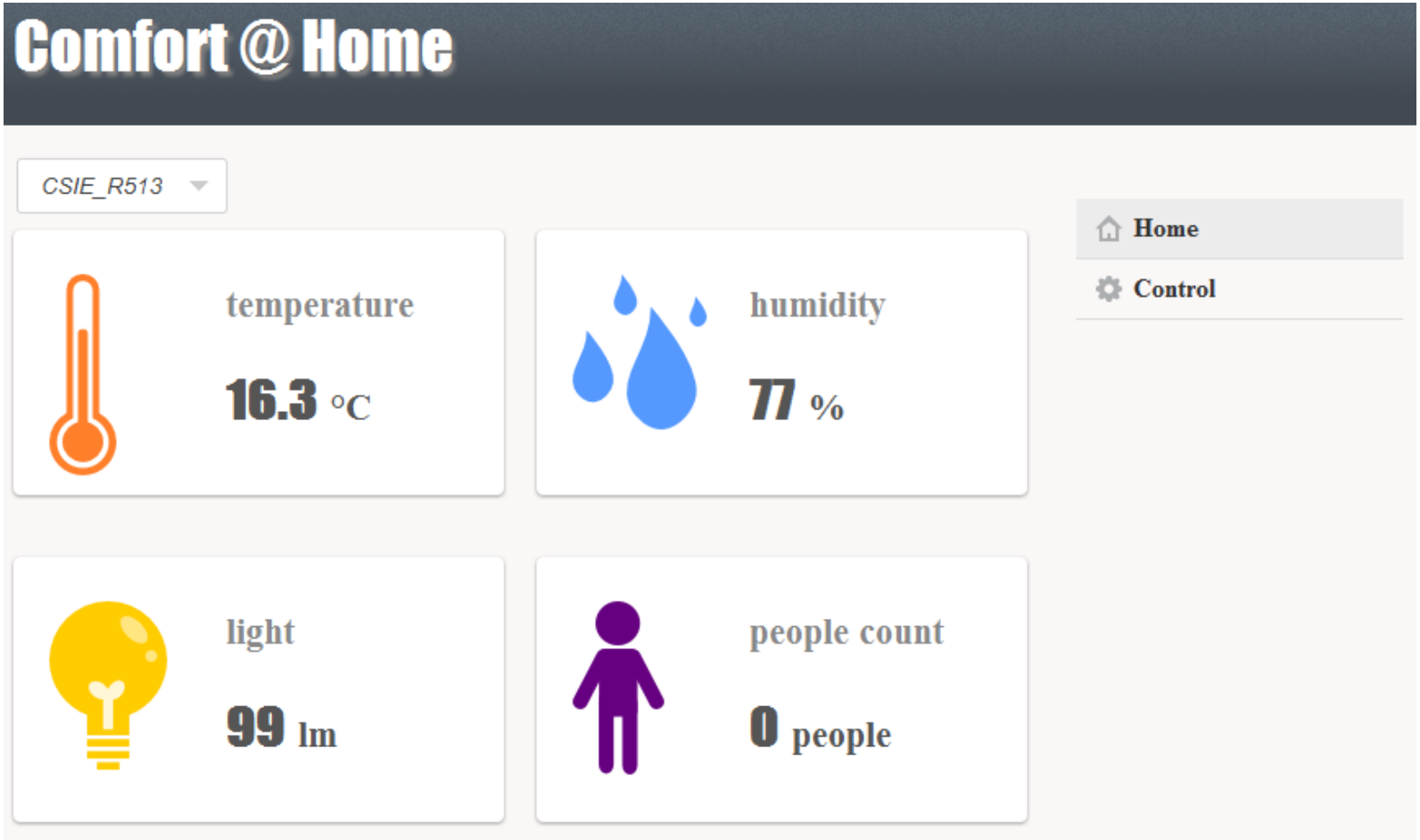
- User go to the "control" page to see the deployment of sensors in each room, as well as actuators and policy.





comfort@home – Status Page

- The dashboard page shows the reading of each sensor in the room.



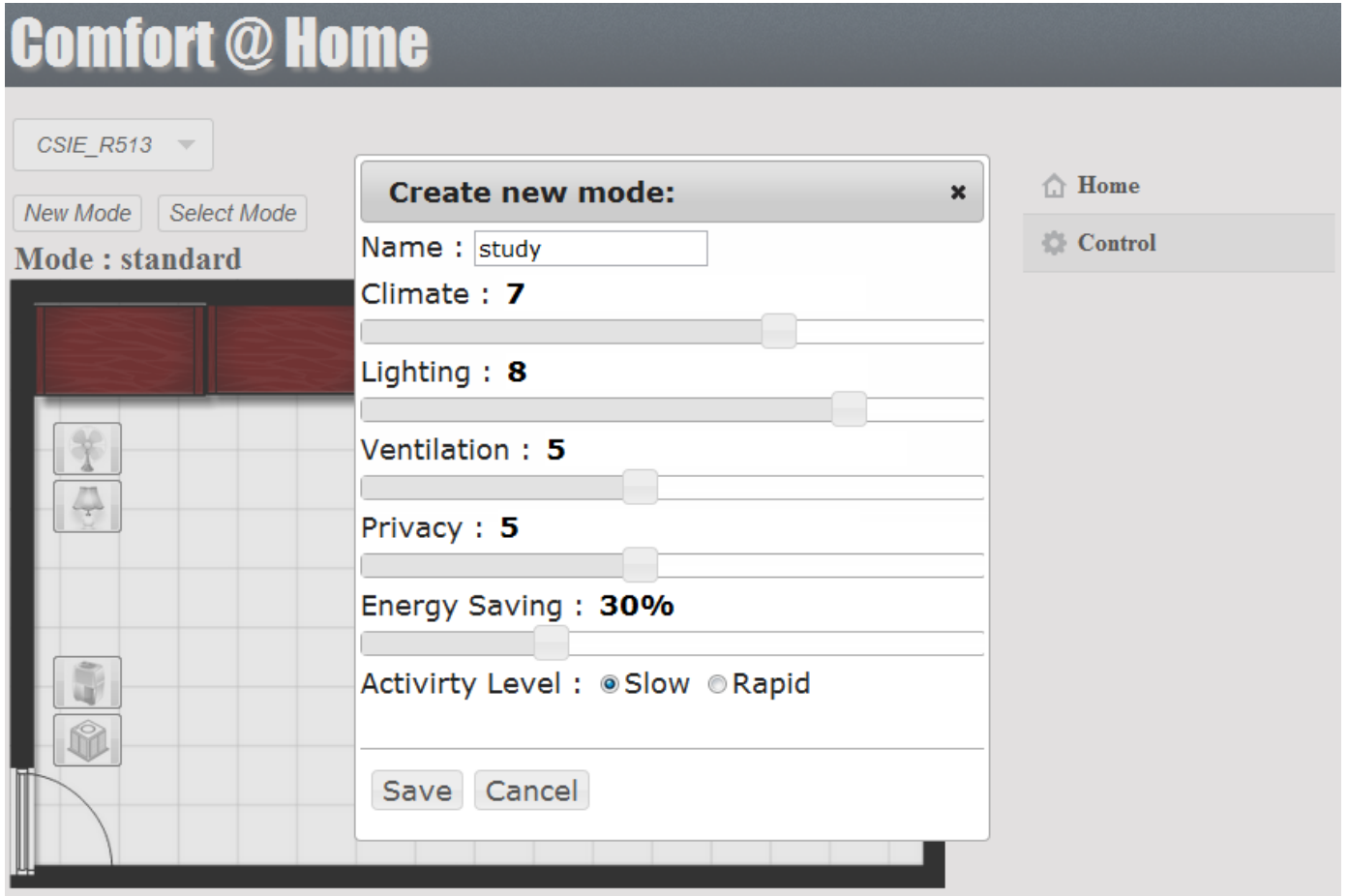
comfort@home – Historical Data

Intel-NTU Center

- User and Master can see/use historical data to help them make more intelligent decisions.

comfort@home – Mode Definition Page

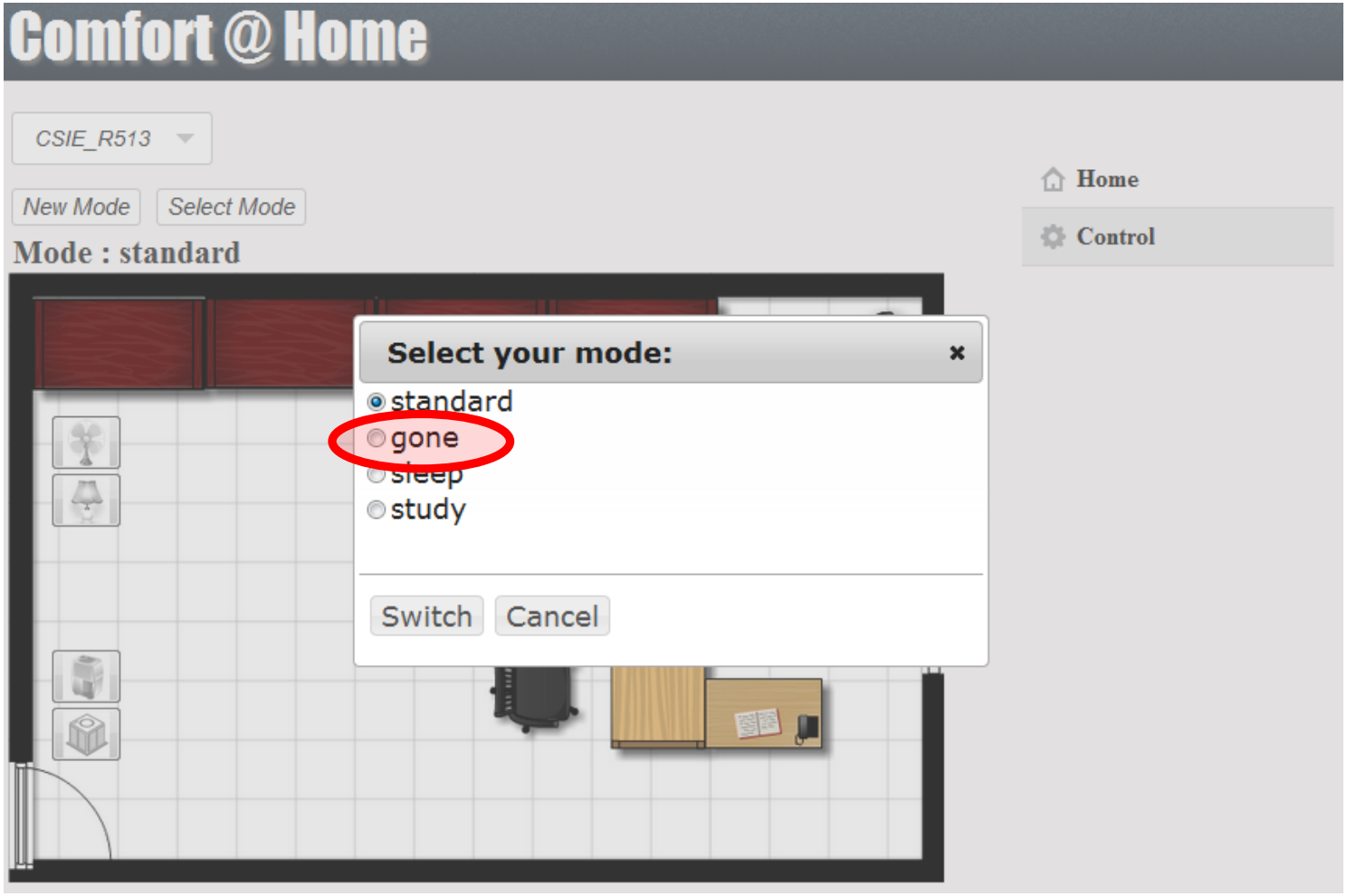
- User creates a mode definition with desirable attribute values, then clicks the “Save” button.



comfort@home – Mode Selection Page

Intel-NTU Center

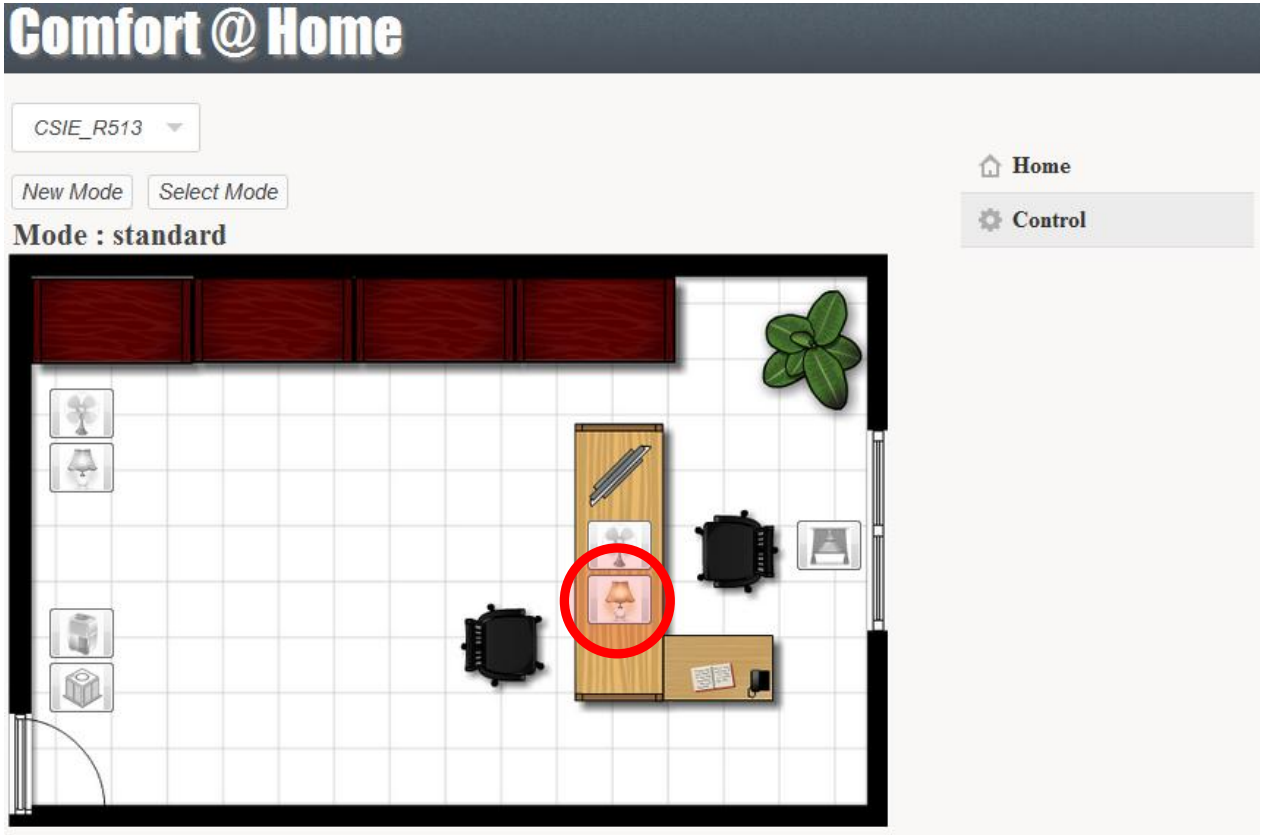
- The user or application can change mode dynamically: switching to "away" mode will preserve more energy and increase security level.



comfort@home – Manual Overwrite

Intel-NTU Center

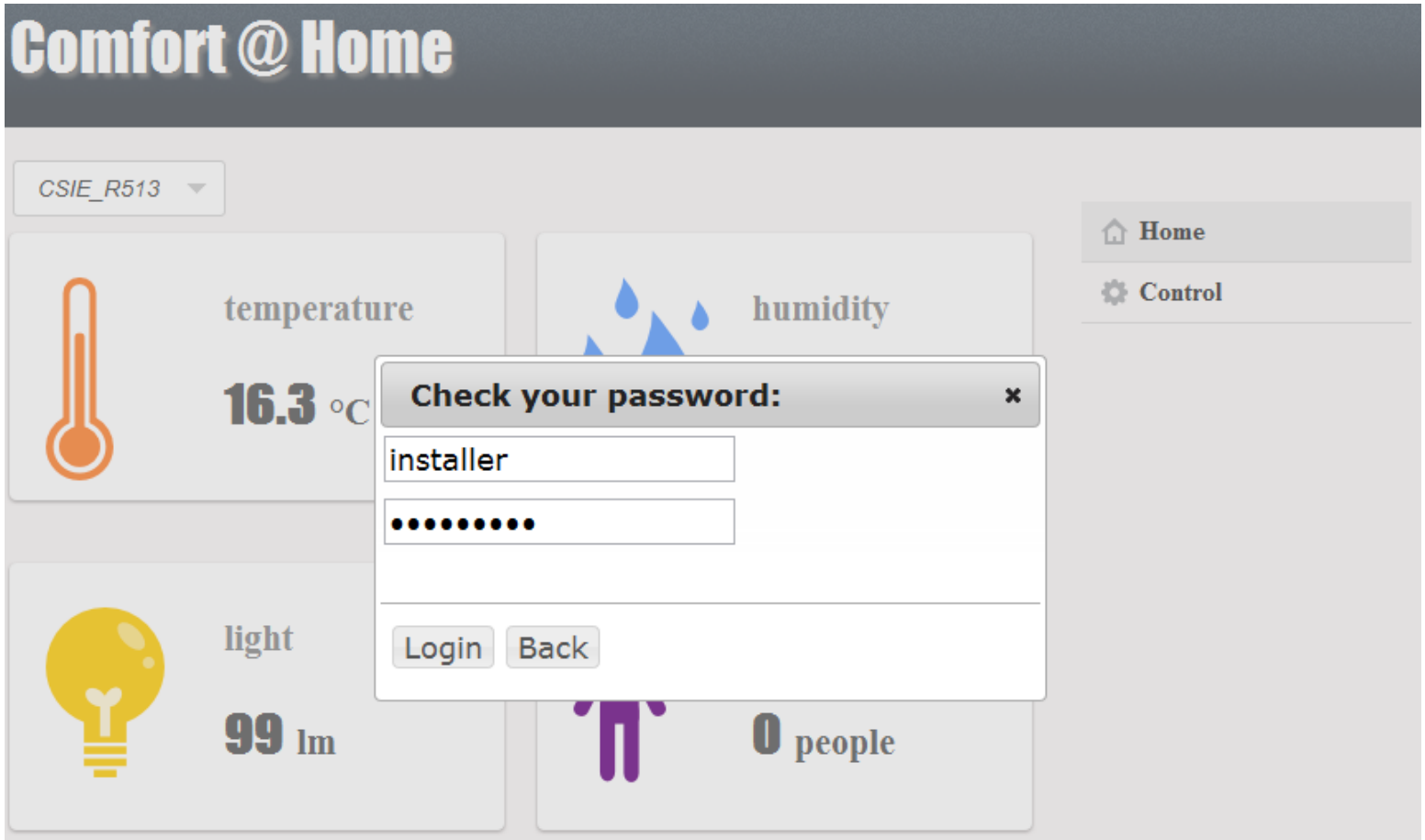
- A lamp can be manually controlled. If the user thinks it's too dark, he can turn on the lamp by clicking on the icon.
- The application can learn user's needs and adjusts its policy accordingly.



comfort@home – Installer Page

Intel-NTU Center

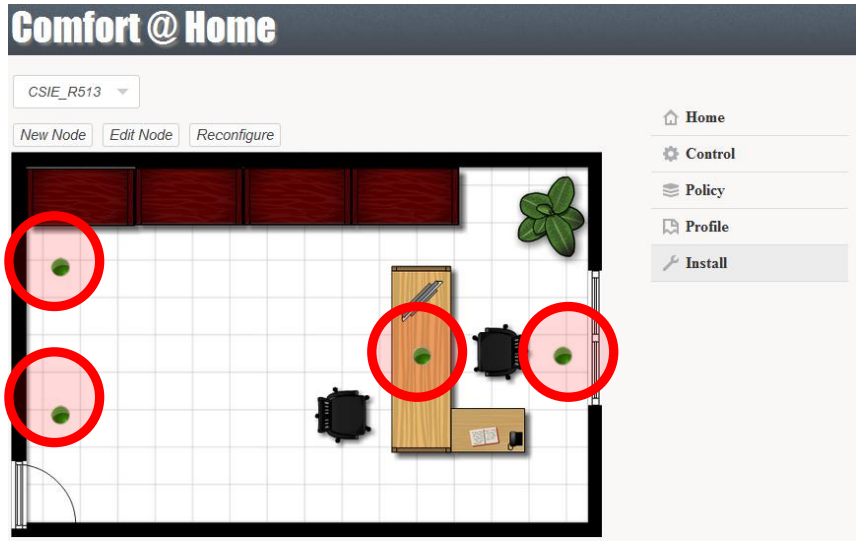
- Installer can change the policy setting, profile specification and deployment.

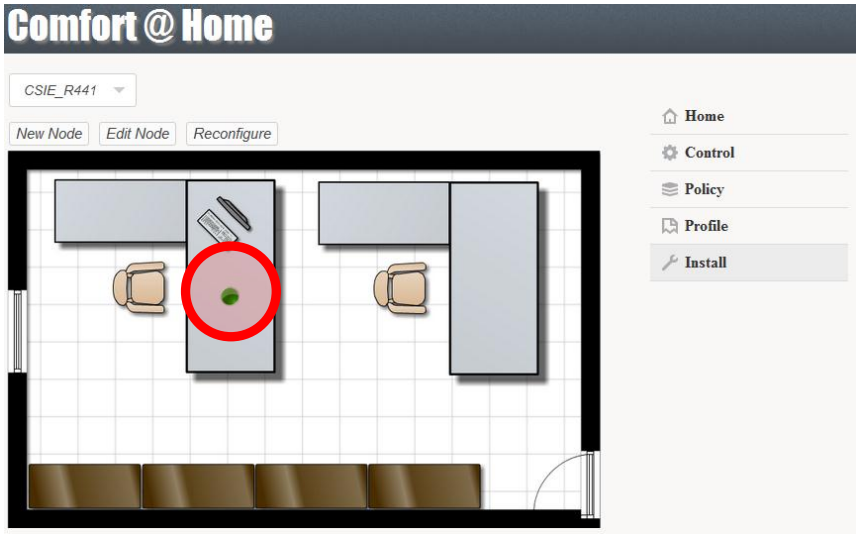


comfort@home – Installer Detail Page

Intel-NTU Center

- Installer can see the whole deployment with node-view. They are not allowed to see the mode status unless the user allows it.

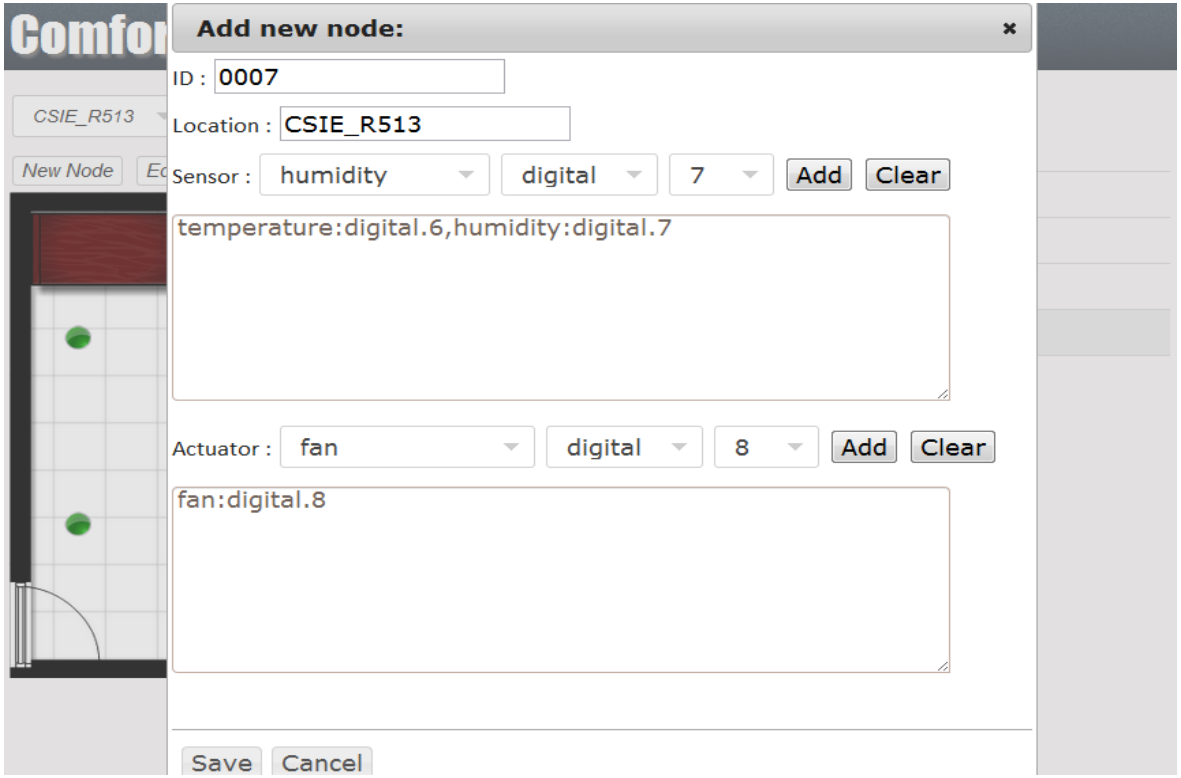




comfort@home – Installer Update Page

Intel-NTU Center

- Installer can add new nodes if he has the right authority. He can enter the information in a node profile.
- The binding between logical and physical sensors are now manually configured and should be automatically configured after WuKong's profile framework is deployed.



What's the profile specification?

- In our demo application, we have…
  - Node
    - A detailed information with node ID and location.
    - What sensors and actuators on this node can be controlled.

node_ID	location	sensor	actuator
0001	CSIE_R513	[temperature: D.6, humidity: D.7, light: A.2, sound: A.4, motion: D.4]	[fan: D.10, lamp: D.12, monitor: D.8]
0002	CSIE_R513	[temperature: D.6, humidity: D.7, light: A.2, sound: A.4, motion: D.4]	[fan: D.10]
0003	CSIE_R441	[temperature: D.6, humidity: D.7, motion: D.4]	[fan: D.10]

What's the profile specification?

- In our demo application, we have…
  - Sensor
    - A detailed information with sensor ID, location and pin number.
    - Sampling period, type, valid range, sensitivity, voltage, power resource, attribute, etc… so sensors can be designated for different functionality/applications.

sensor_ID	sensor_name	sample period	type	valid range	attribute
0001	temperature	100 ms	digital	[-40,125] ˚C	climate
0002	humidity	100 ms	digital	[0,100] %	climate
0003	CO2	100 ms	analog	[0,2500] ppm	climate
0004	light	1 ms	analog	[0,1023] lumen	lighting
0006	motion	1 ms	digital	0,1	activity_level

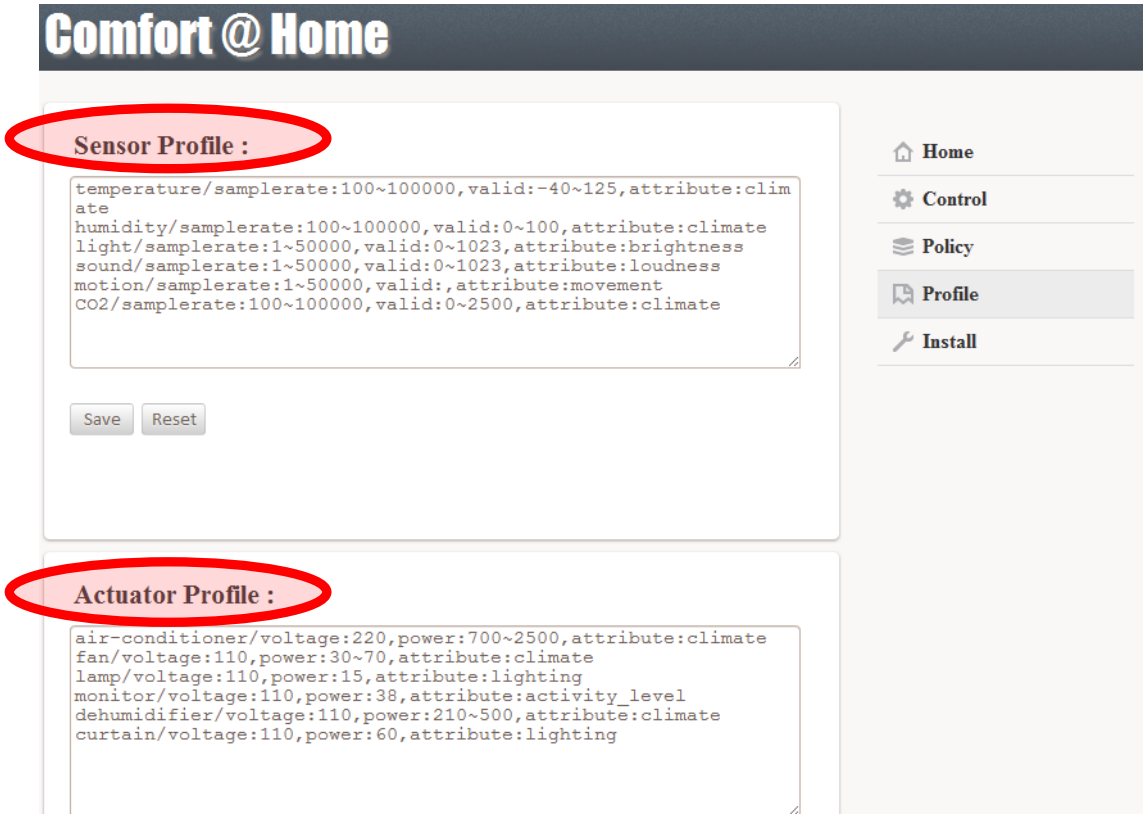
What's the profile specification?

- In our demo application, we have…
  - Actuator
    - A detailed information with actuator ID, location and pin.
    - Voltage, power, power resource, attribute, etc… so actuators can be designated for different functionality/applications.

actuator_ID	actuator_name	voltage	power	power resource	attribute
0001	air-conditioner	220V	[700,2500] W	wall socket	climate
0002	fan	110V	[30,70] W	wall socket	climate
0003	lamp	110V	15 W	wall socket	lighting
0004	dehumidifer	110V	[210,500] W	wall socket	climate
0005	curtain	110V	60 W	wall socket	lighting

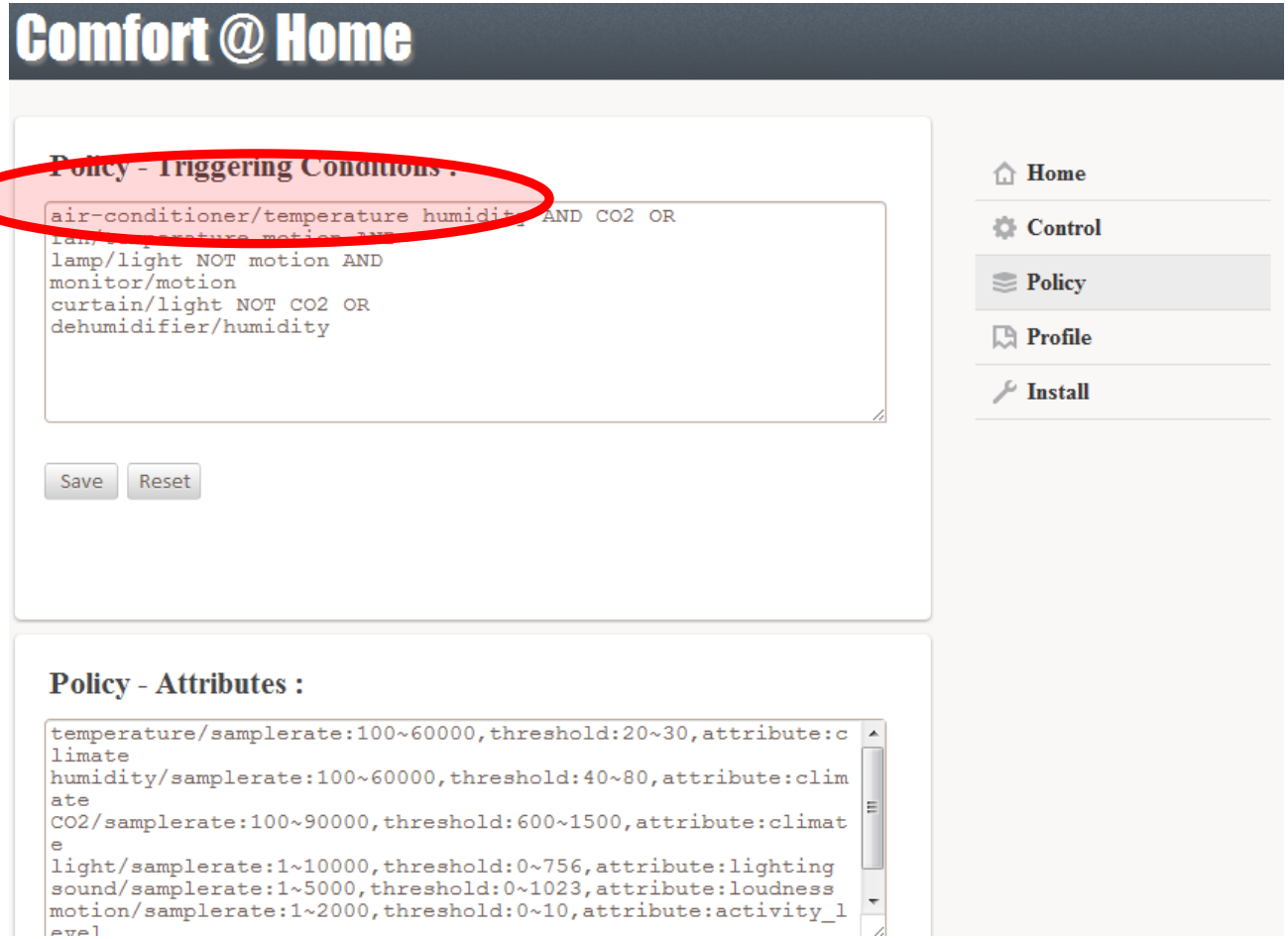
Sensor Profile

- Installer can add new sensor in the profile page, and save its hardware specification.



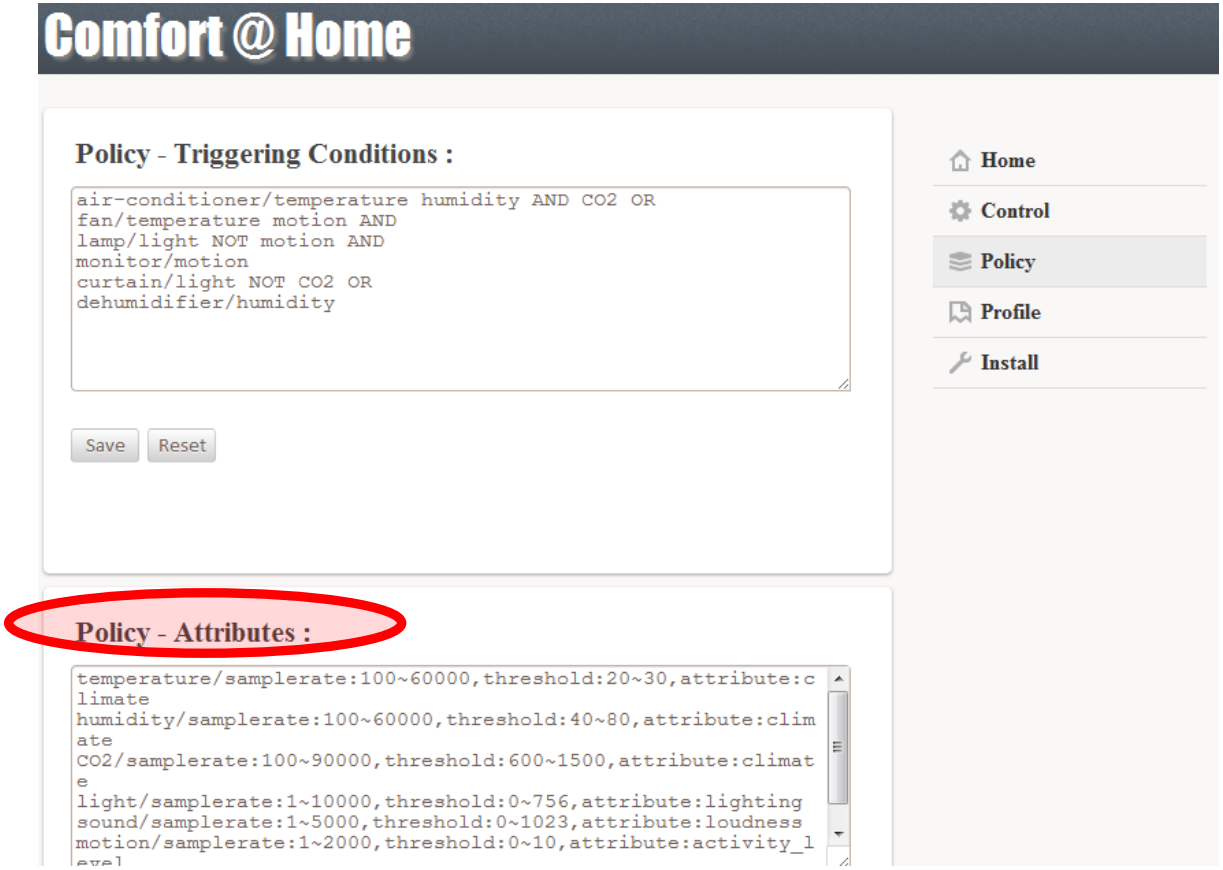
Policy – Triggering Conditions

- Installer can manually edit the triggering conditions with each actuators.



Policy – Attributes

- Installer can manually edit the attribute values if end-user have a special request, but it is still followed by profile constraints.



Wu-Kong : Progress Report

- Framework design
  - *Profile framework*
  - *Policy definition*
- Software development
  - *comfort@home apps*
  - *NanoKong VM* (discussed next)
- Security assessment

NanoVM: A Very Simple VM

- Built for simplicity
  - NanoVM is an open-source implementation of the JVM. The NanoVM was initially developed to run on the Atmel AVR ATmega8 used in the Asuro Robot. It was ported to run on the C't-Bot and the Nibo-robot
  - NanoVM 1.6 released on July 2007 with Asuro ATmega168 support
  - Very small (8-16K code, 768 bytes RAM)
    - Fit on almost any sensor platform
  - Very simple architecture
    - Easy learning curve
    - Easy to port, adapt and extend
  - *Rather slow*
- Simplicity and flexibility make it ideal for initial designs.
- Several options exist to improve performance.

NanoKong – VM for *Wu-Kong* Sensors

Intel-NTU Center

- Porting NanoVM to Arduino
  - Completed
- Arduino IO
  - Completed: Added support for most IO functions present on Arduino but originally not supported by NanoVM
  - Working on: low-power mode
- Radio
  - Completed: Generic interface to support multiple radio technologies
  - Completed: Z-Wave support
  - Working on: Zigbee support
- Code update
  - Completed: Uploading new code, replacing the complete Java program
  - Working on: blocks of code to implement extended profiles

INC NSC

Research Team

- Dr. Kwei-Jay Lin (Professor, UC Irvine & NTU; Ph.D. Maryland)
- Dr. Daniel Shih (Associate Professor, NTU; Ph.D. Illinois)
- Dr. Jane Hsu (Professor, NTU; Ph.D. Stanford)
- Dr. Yu-Chung Wang (Senior Researcher, Ph.D. UC Irvine)

- Niels Reijers (PhD student)
- Tiffany Yi-Ting Tsao (PhD student)
- Penn Su (MS student)
- Yi-Long Tsai (MS student)
- Bo-Lun Tsai (MS student)
- Guan-Fan Wu (MS student)

Conclusion: M2M Future

Intel-NTU Center

- M2M future is on many new services to be deployed
- Wu-Kong is our answer to this grand challenge of building an intelligent M2M platform
- Users can benefit by receiving optimal services
- Operators can benefit by providing more efficient and effective services
- New programming paradigm is needed to unleash the full power of future M2M systems