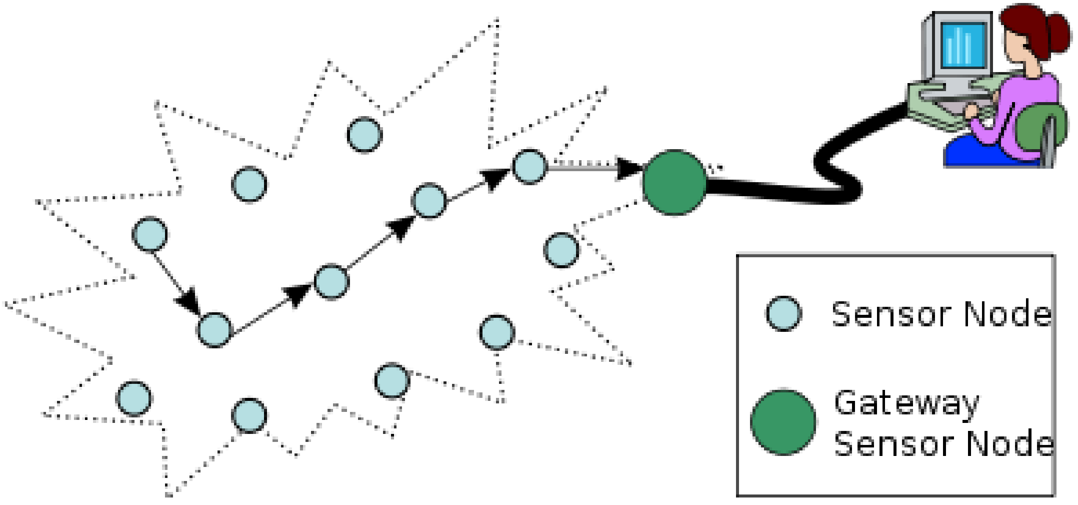
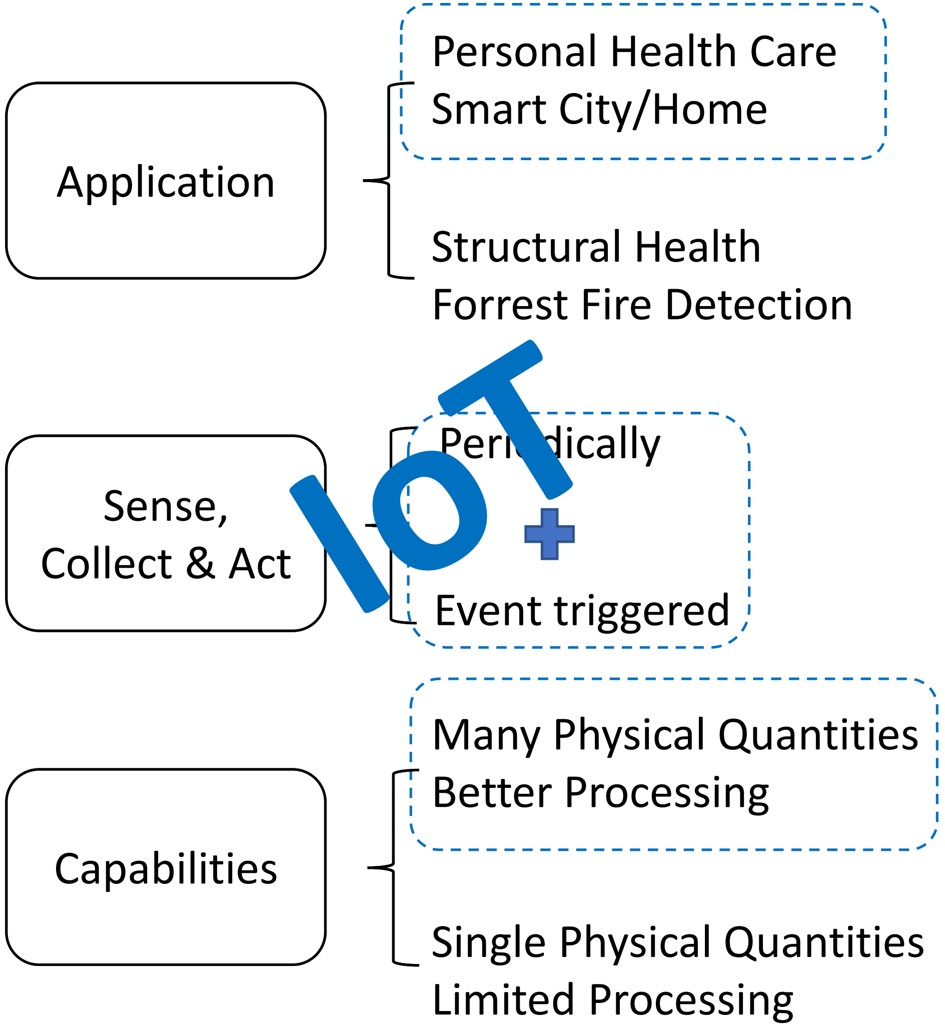
Context-Aware Sensor Networks

Shashank Gaur

Wireless Sensor Networks





Application
- Personal Health Care
- Smart City/Home
- Structural Health
- Forrest Fire Detection

Sense, Collect & Act
- Periodically
- +
- Event triggered

IoT

Capabilities
- Many Physical Quantities
- Better Processing
- Single Physical Quantities
- Limited Processing

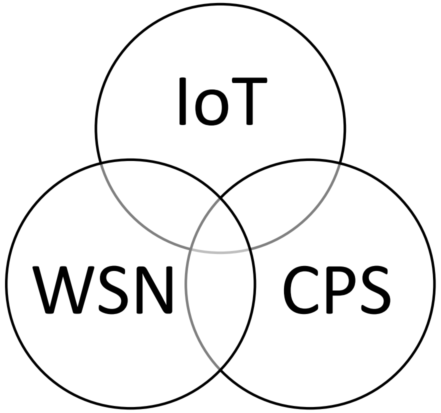
Internet of Things

- Define IoT?
    - Buzz word!?
- CPS
    - Another Buzz word!?
    - US characterization, started with
- Examples
    - Smart Grid
- Difference with IoT
    - “Internet”
    - “Things”

“CPS is a system of collaborating computational elements controlling physical entities”

WSN & IoT

- WSNs are NOT subset of IoT
- Famous phrases
  - WSN based IoT, WSN enabled IoT
- Actuation?
  - WSAN
- No need of “Internet” and “Things”



Internet of Things

- Features
  - Internet, Things, Connected, Sensing/Actuation, Intelligence, etc.

“An IoT is a network that connects uniquely identifiable ‘Things’ to the Internet. The “Things” have sensing/actuation and programmability capabilities.”

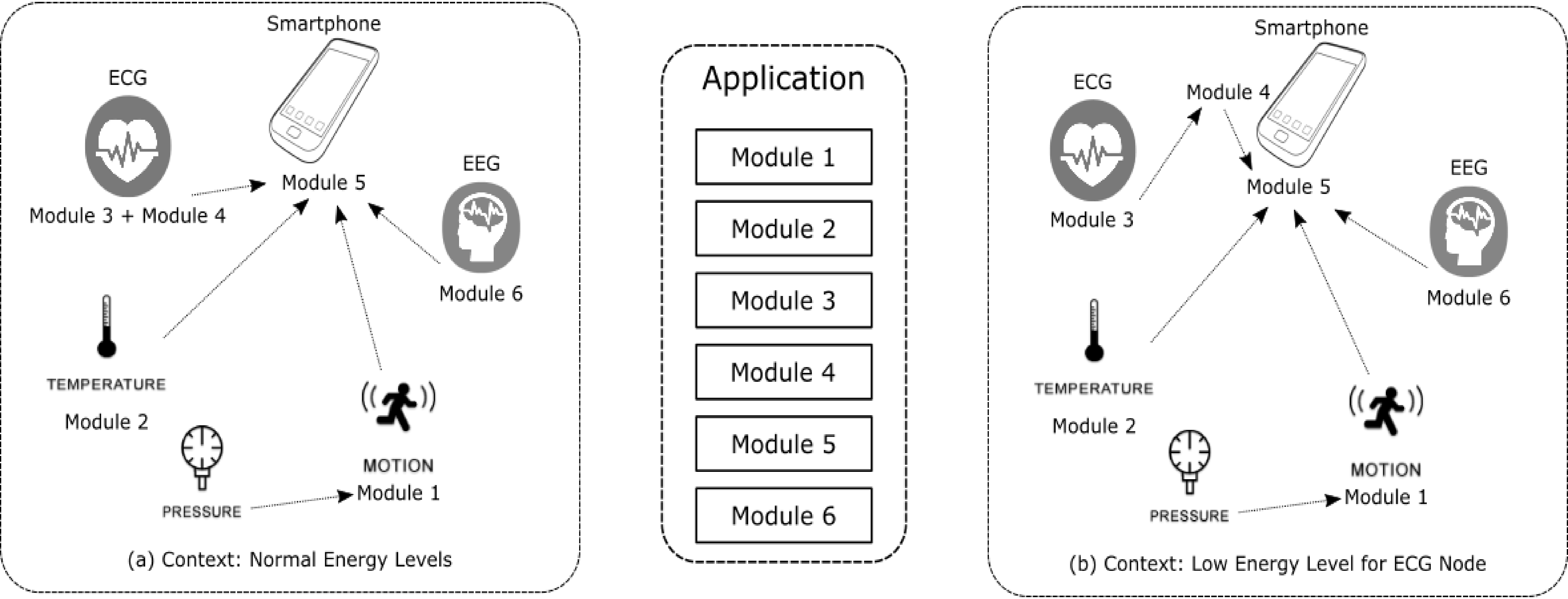
Small environment scenarios

“IoT is a complex network that interconnects ‘Things’ to Internet and create a physical or virtual representation in a global system which provides services of ‘Things’ anywhere, anytime and for anyone”

Large environment scenarios

Context-Awareness

- IoT paradigm captures different devices with different capabilities
- With such a wide spectrum, there are changes
  - Changes in Execution Environment
  - Changes in Network Capacity, Connectivity and Cost of Communication
  - Changes in Accessibility of Devices and/or User
- To provide efficient and reliable services through IoT, such changes much anticipated
  - Context-Awareness in IoT
- Context-Awareness involves
  - Being aware of the "*context*" in which applications are executed
  - Detect changes in the *"context"* and deploy new set of applications accordingly
- Context-Awareness is not New
  - But IoT brings in new issues such as macroprogramming, interoperability, etc.



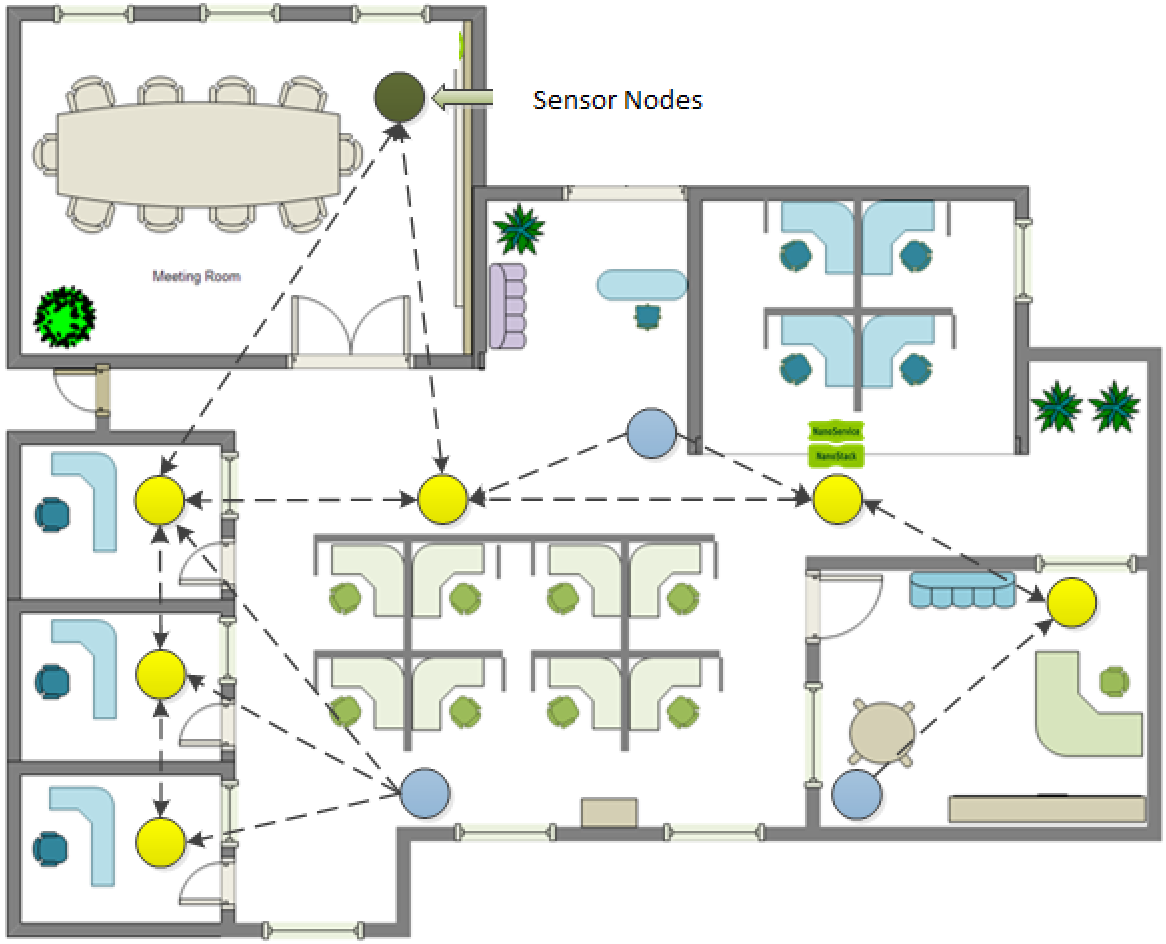
(a) Context: Normal Energy Levels

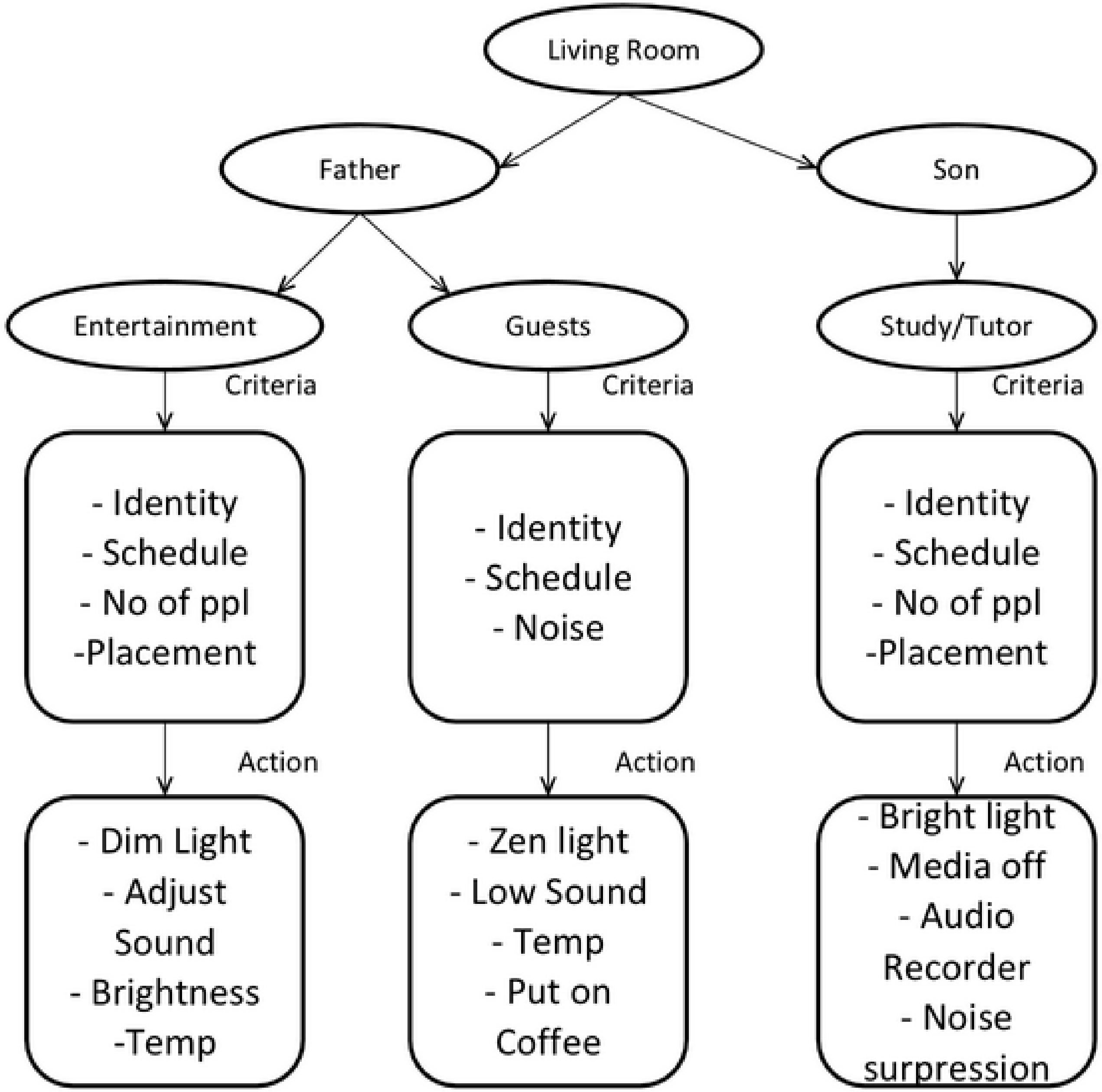
(b) Context: Low Energy Level for ECG Node

Normal Habitat
Check Presence
And
Maintain Heating

Fire Alarm
Detect Presence
And
Open Doors

Fire Alarm
Monitor high
temp area &
Notify Firemen





Living Room

Father

Son

Entertainment

Criteria

- Identity
- Schedule
- No of ppl
-Placement

Action

- Dim Light
- Adjust Sound
- Brightness
-Temp

Guests

Criteria

- Identity
- Schedule
- Noise

Action

- Zen light
- Low Sound
- Temp
- Put on Coffee

Study/Tutor

Criteria

- Identity
- Schedule
- No of ppl
-Placement

Action

- Bright light
- Media off
- Audio Recorder
- Noise surpression

Context

- What is Context?
  - Simple information retrieved by set of simple rules
  - Complex information retrieved via machine learning
- Set of simple rules apply for WSNs
  - Precise data available from sensors and actuators
- Context is information about the whole ecosystem
  - User, System and Physical Environment
- Each Context has 3 important aspects:
  - Where you are?
  - When you are?
  - How you are?
- Additional aspects:
  - Who you are with?

Why?

- Where is the need?
- Healthcare
  - Identify local environment upon arrival
  - Support for medical staff with rapidly changing patients, schedules, etc.
- Production Industry
  - Detecting and configuring mounted machinery
  - Free manual interaction with information handling
  - Amazon Echo

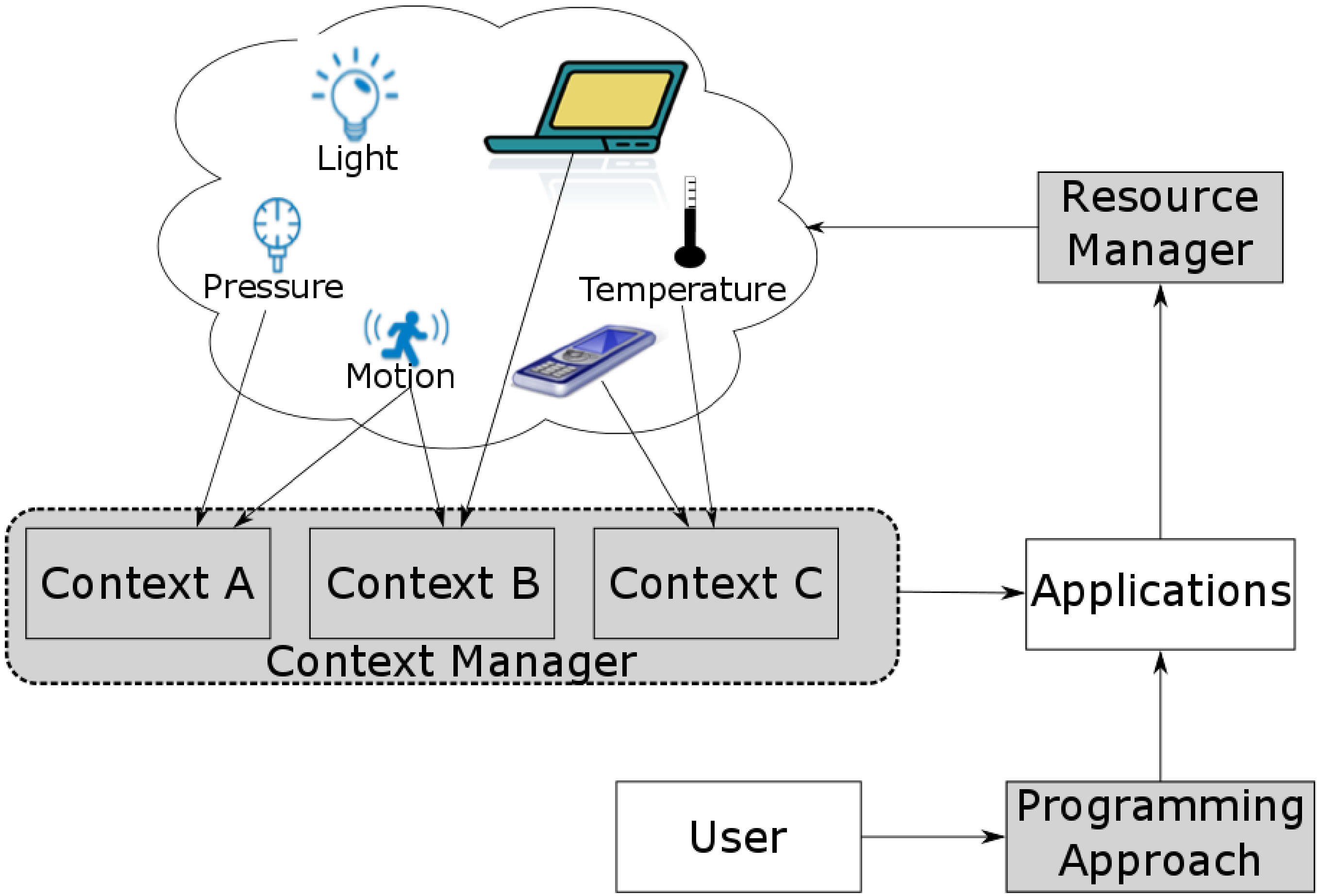
Problem Statement

- With each evolution towards IoT, the middleware solutions changes.
  - Users are expected to adapt to those changes
  - System performance are affected.

“There is a need of middleware solution which can adapt to new changes while providing almost-to-zero effect on ***programmability*** *for users. In addition, the same solution must able to support* ***interoperability*** *for effective system performance“*

Context Aware Framework

Intelligence based on collected data



Change in Resources

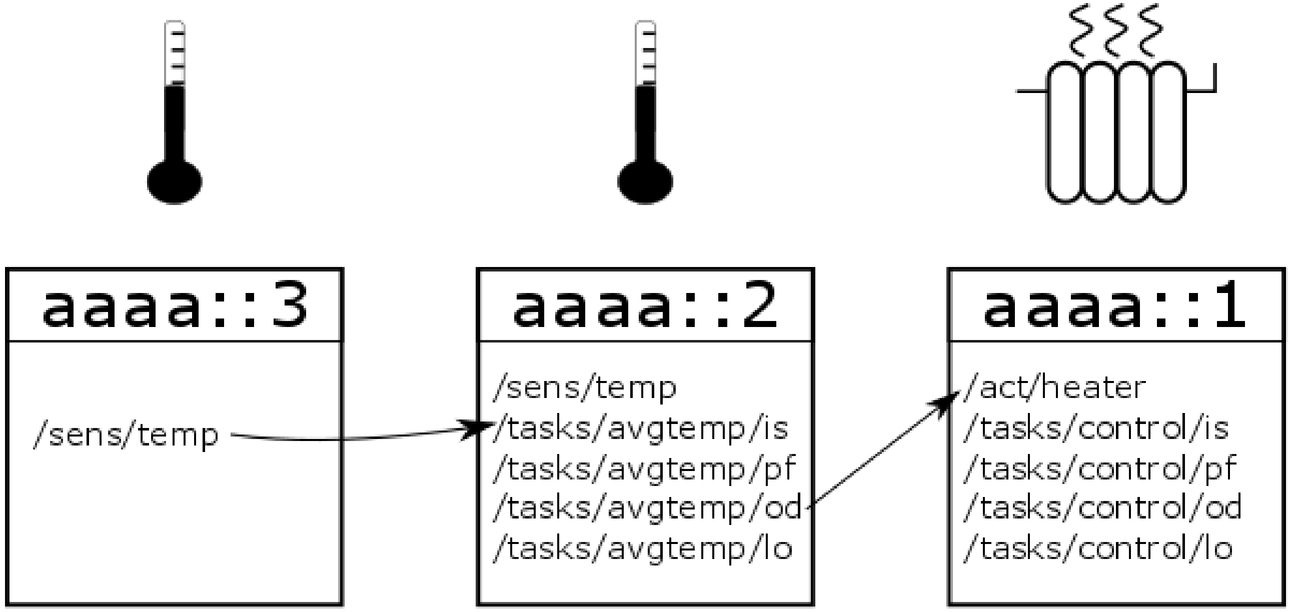
Addition of new tasks

Programming Approach

- Required Features
    - Easy to build - Abstraction
    - Not at all Resource centric - Modularity
    - Both Applications and Resources may change - Mobility
- Building upon existing state of the art
    - T-Res provides programming abstraction for IoT
        - Still lacks other features

T-Res

- Tasks in T-Res
  - Input Source
  - Processing Function
  - Output Source
  - Last Output
- Uses CoAP operations
  - Push, Post, Get, Observe
- IPv6 addresses for resources
- T-Res uses CoAP agent Copper for Firefox
  - Requires user instructions for each CoAP operation
  - Can be automated easily



aaaa::3

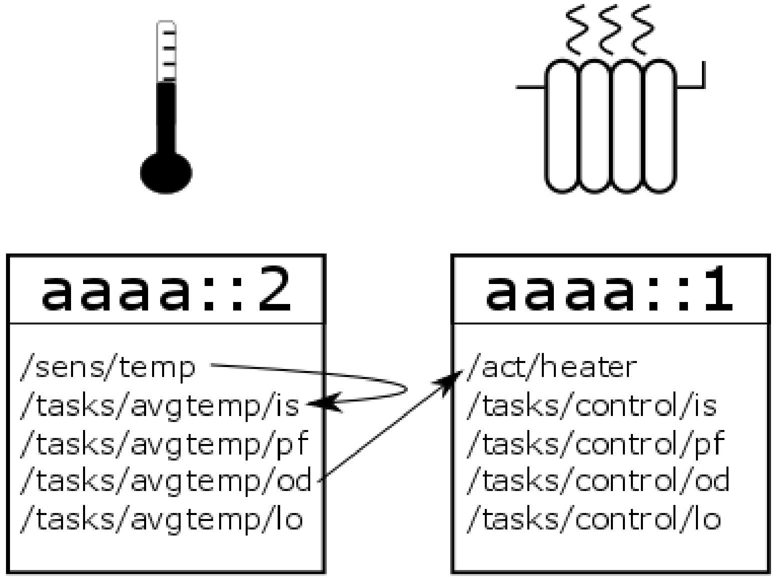
/sens/temp

aaaa::2

/sens/temp
/tasks/avgtemp/is
/tasks/avgtemp/pf
/tasks/avgtemp/od
/tasks/avgtemp/lo

aaaa::1

/act/heater
/tasks/control/is
/tasks/control/pf
/tasks/control/od
/tasks/control/lo



aaaa::2

/sens/temp
/tasks/avgtemp/is
/tasks/avgtemp/pf
/tasks/avgtemp/od
/tasks/avgtemp/lo

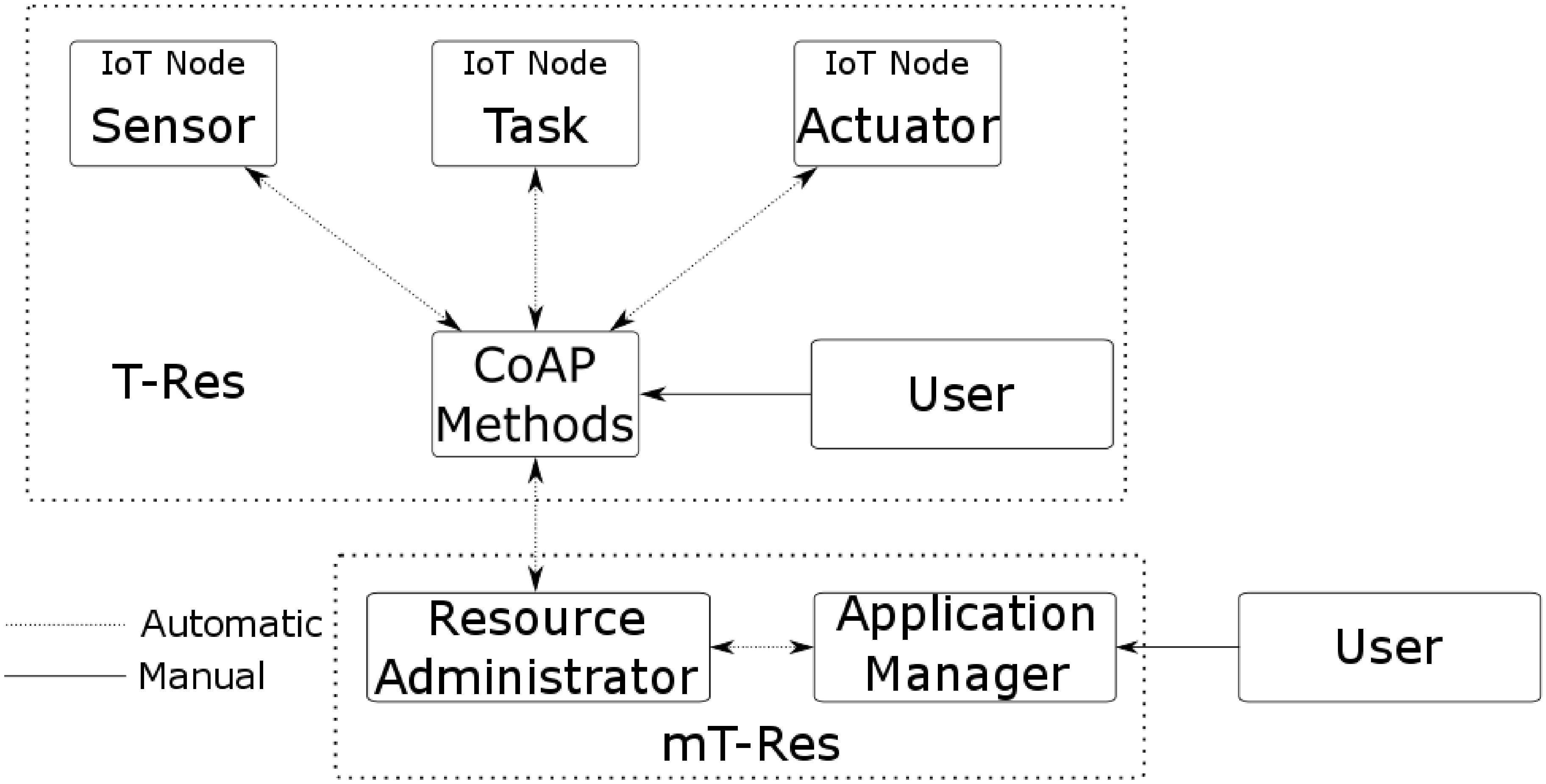
aaaa::1

/act/heater
/tasks/control/is
/tasks/control/pf
/tasks/control/od
/tasks/control/lo

mT-Res

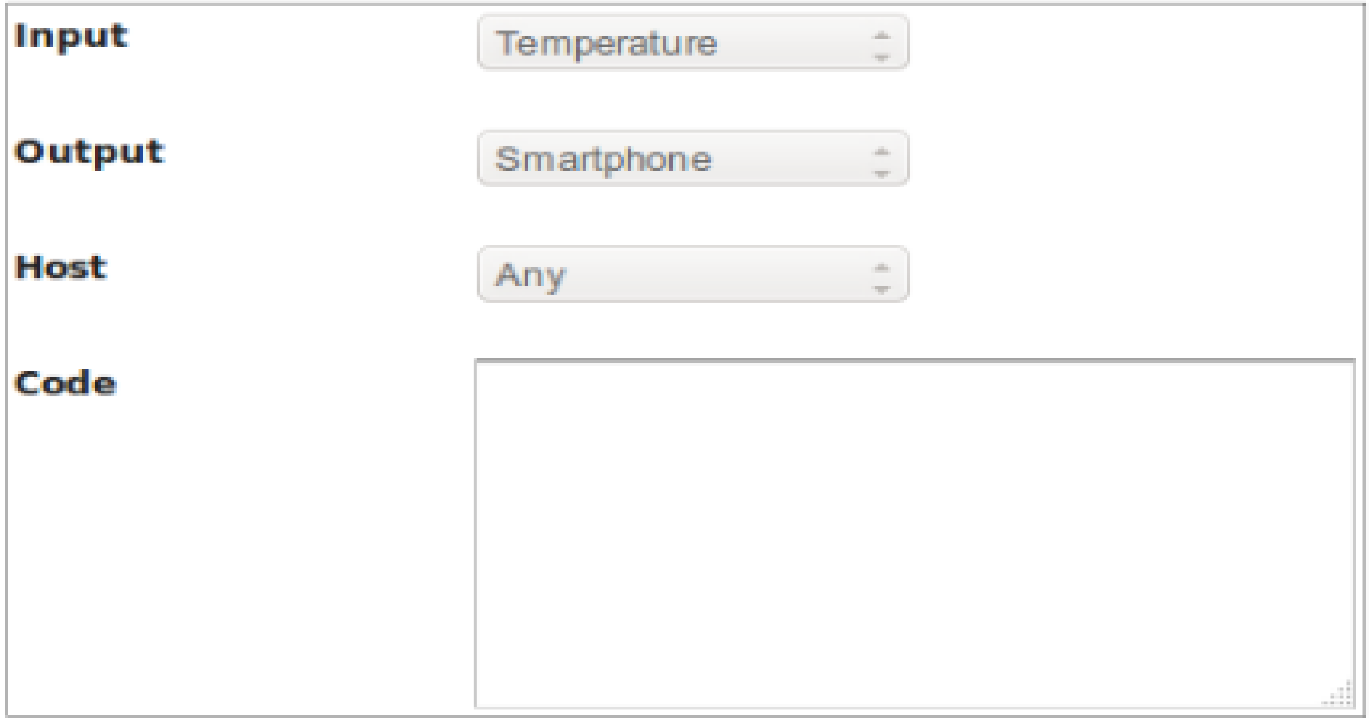
- Mobility T-Res
  - Ability to move Code and Resources around each other
- mT-Res extends T-Res with
  - Automated CoAP operations
  - Limited sense of Context-Awareness
- Simple applications such as
  - node failure
  - Change of host node
  - New application for each node

mT-Res


IoT Node
Sensor
IoT Node
Task
IoT Node
Actuator
T-Res
CoAP Methods
User
Automatic
Manual
Resource Administrator
Application Manager
User
mT-Res

mT-Res: Application Manager

- Web framework in Django
- For user to submit tasks
  - Input Type
  - Output Type
  - Host (Fixed or Any)
  - Code



Input: Temperature

Output: Smartphone

Host: Any

Code

- Wraps T-Res code with small functions
  - Conditional flags

mT-Res: Resource Administrator

- Python Scripts enabling
  - CoAP functions such as Push, Pull, Get, Observe
- Always active and updating resources
  - Provides a table to Application Manager
- Works along the Application manager
  - After tasks are submitted, allocates resources
- For any change detected in Resources
  - Restarts the resource allocation

mT-Res

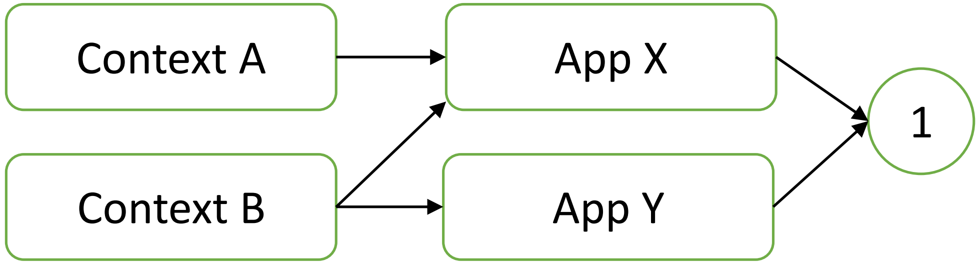
- Implemented in Python and Django
- Available for development and use
  - https://bitbucket.org/shashankgaur_/tres_extension
- v2.0 and further
  - TelosB motes
  - IoT devices: Smartphone, projector
  - Application scenarios

What is Next?

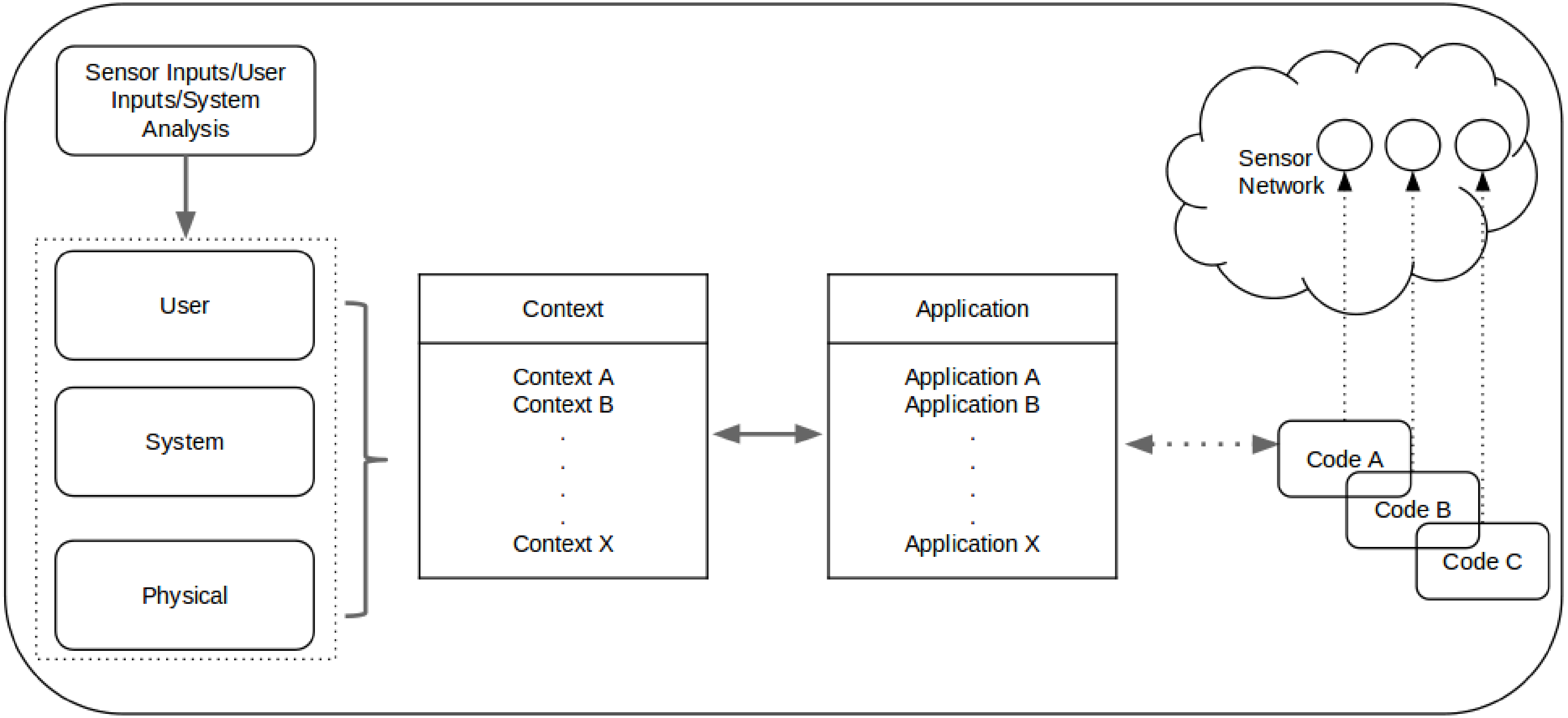
- Remaining features of Programming Approach
    - Abstraction
    - Mobility
- Resource Manager
- Context Manager

Resource Manager

- Problem: For multiple contexts, (re)assign the resources
- Conflicting contexts at same time
  - Energy-saver and Security
  - Entertainment and Security
  - Energy-saver and Entertainment



Context Manager



Sensor Inputs/User Inputs/System Analysis

User

System

Physical

Context
Context A Context B . . . . Context X

Application
Application A Application B . . . . Application X

Context

- Defining Context
    - Who
        - Temperature sensor
    - What
        - Provides Temperature
    - Where
        - Geographical location or network location
    - When
        - Time or status according to applications deployed
    - How
        - Status of the Entity, energy levels, communication cost etc.

Internet of Things

Context-Aware ~~Sensor Networks~~

Thank you