# Location-Based IoT Messaging for Indoor Smart Space

("Create Natural Intelligence in IT World")

**Prof. Soon Ju Kang, Ph.D.**

**College of IT**

**Center of Self-Organizing Software Platform (CSOS)**
http://www.csosp.org/
**Kyungpook National University**

Center of Self-Organizing Software-Platform

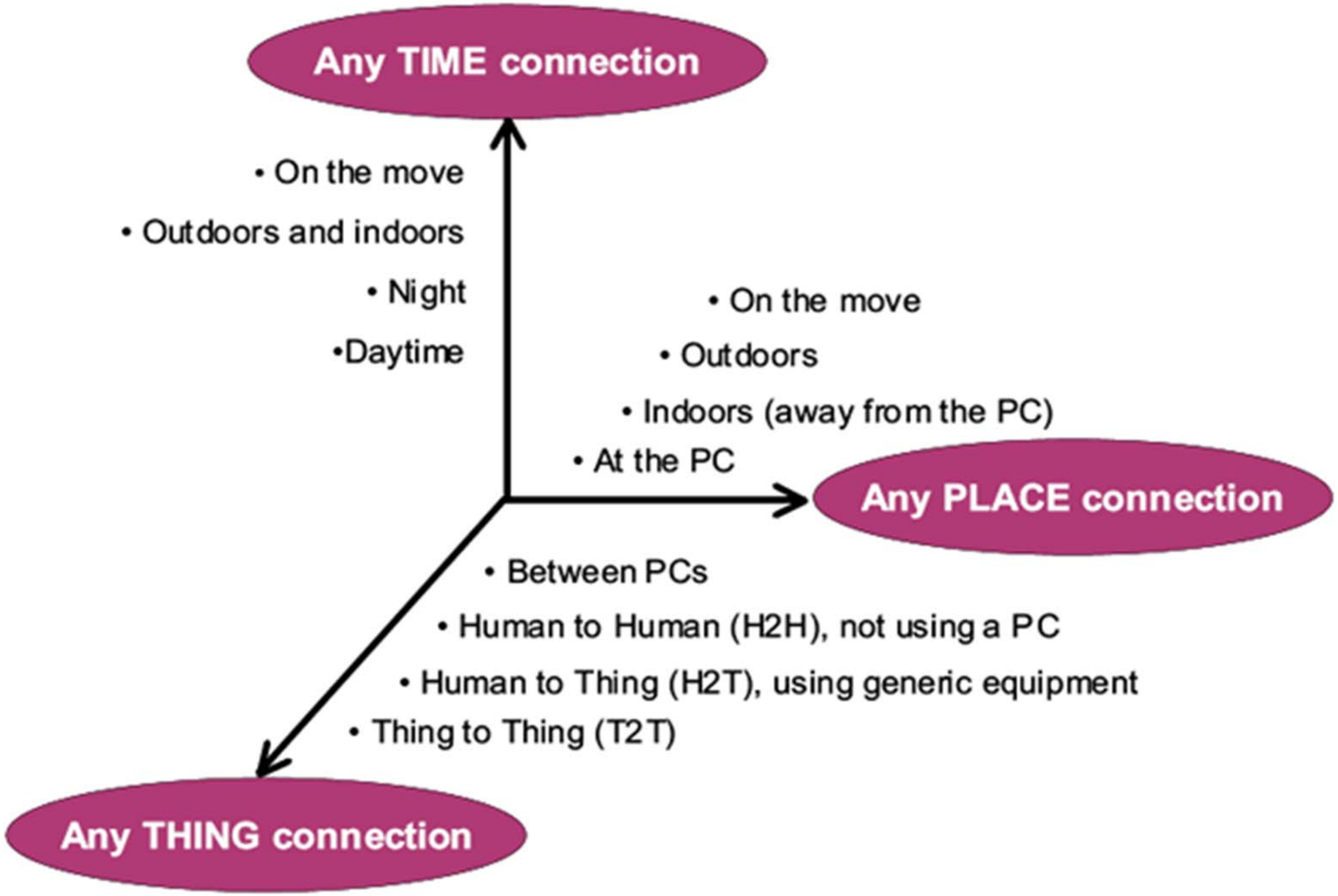
# Contents.

# Part Ⅰ:
# IoT overview and Problem definition

# Future IoT World

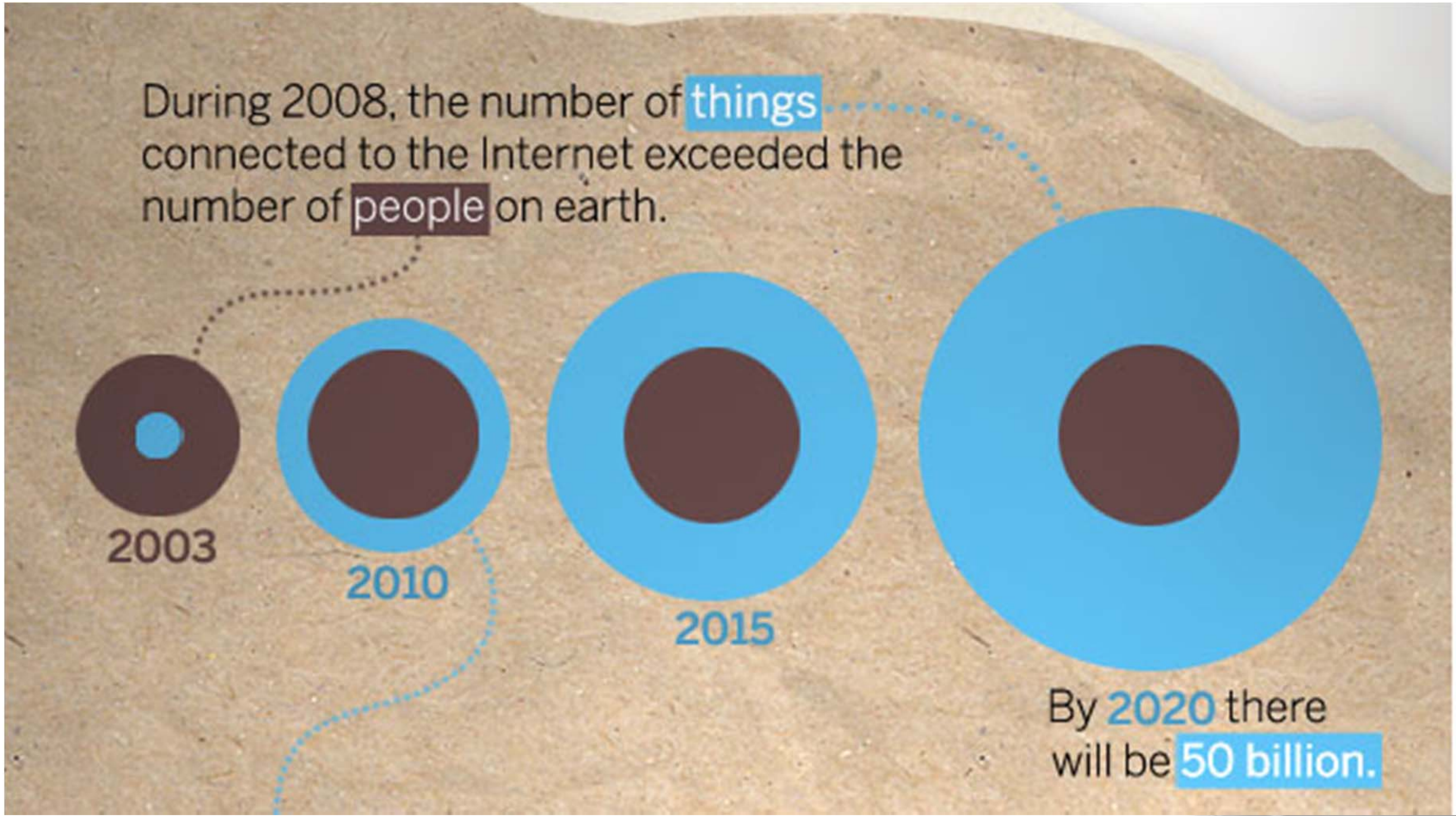
(source: Helen Gill, Ph.D. CISE/CNS NSF presented in the Symposium of Safe & Secure Software and Systems, June 15,2010)

# Future Internet




*Source*: ITU adapted from Nomura Research Institute

# "Thing" connected to the internet



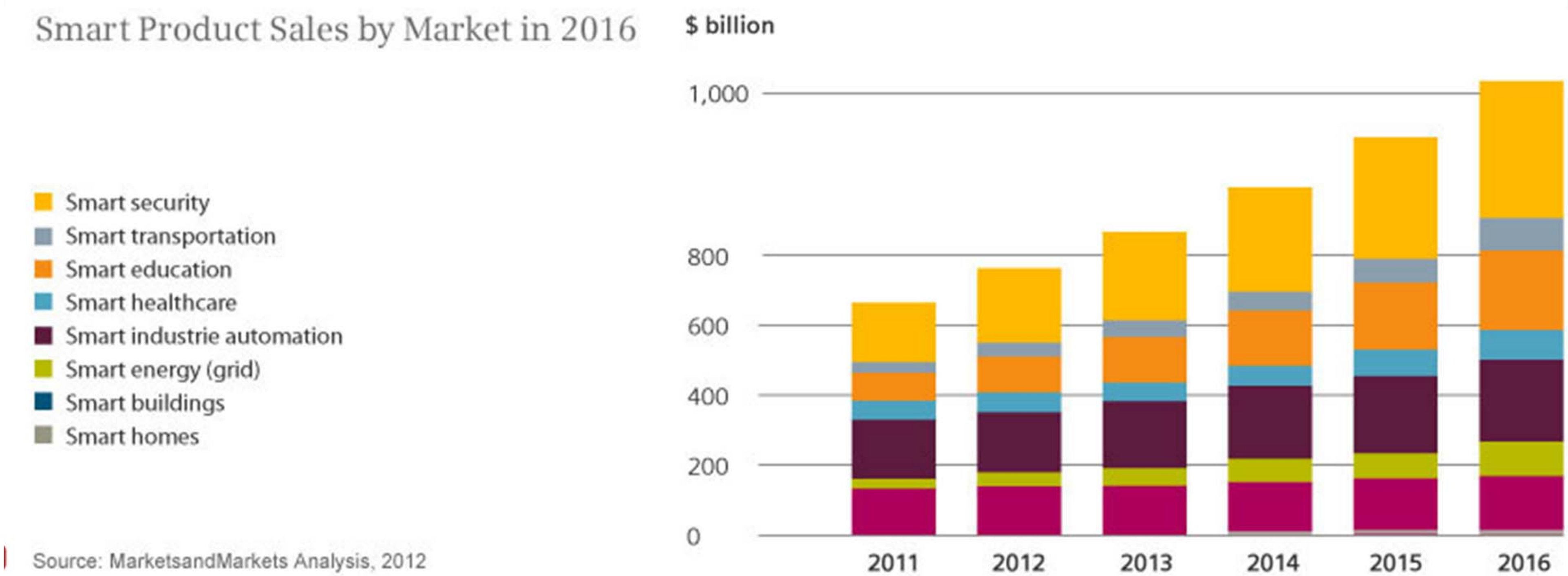

Sources: Cisco IBSG, Jim Cicconi, AT&T , Steve Leibson, Computer History Museum, CNN, University of Michigan, Fraunhofer

Image Courtesy: : CISCO

Center of Self-Organizing Software-Platform


6

# IoT product sales

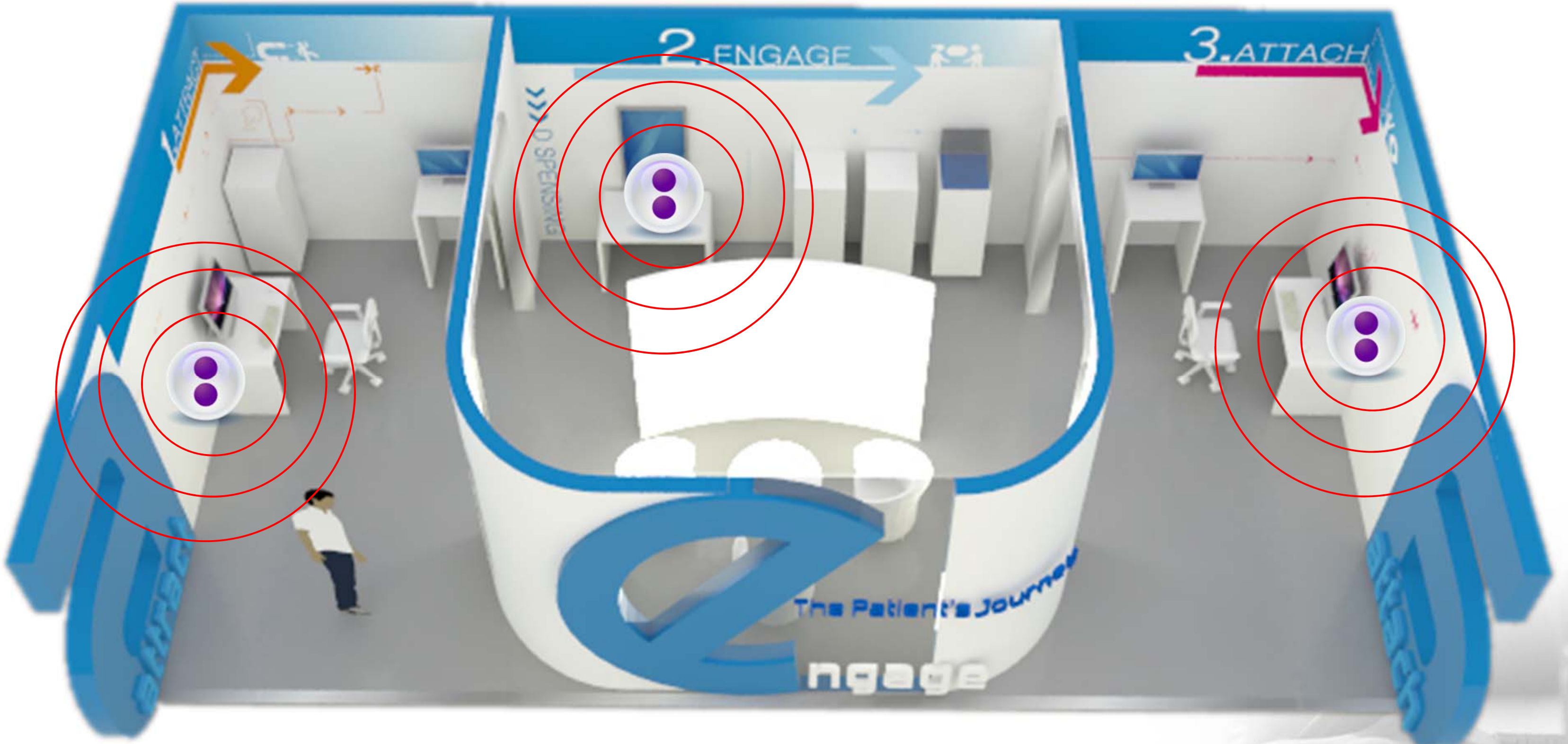



Source: Siemens, http://www.siemens.com/innovation/apps/pof_microsite/_pof-fall-2012/_html_en/facts-and-forecasts-growth-market-of-the-future.html

Center of Self-Organizing Software-Platform


7

# Indoor Positioning Apps (iBeacon)




- Promotion notification
- Survey/ Business Lead

- Bluetooth Low Energy, on the other hand, works in a similar way to NFC, but enables long-distance connections between devices up to 100 metres apart, so you no longer need a fixed payment terminal or checkout to complete a transaction.

CSOS Center of Self-Organizing Software-Platform

# Augmented Reality (AR) for outdoor



- Directional guide for attractions, retails, establishments etc

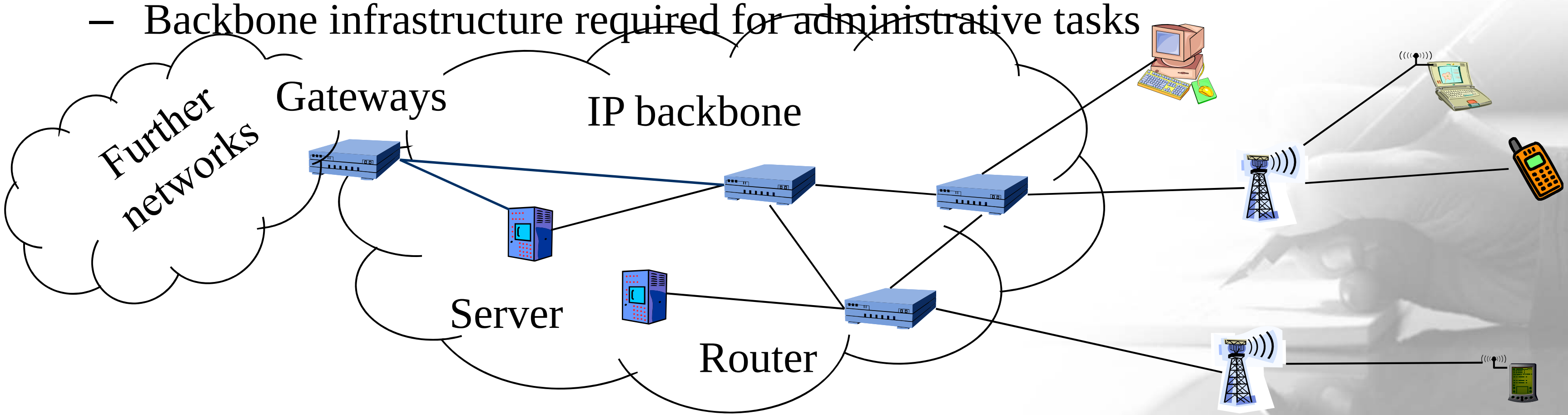
Center of Self-Organizing Software-Platform

# Problems of current IoT

- ❖ How to build the service infra-structure ?
- ❖ Why and how to messaging among mobile and stationary IoT ?
- ❖ The Big Data technology increase the quality of human life ?

# Centralized Infrastructure-based IoT Service

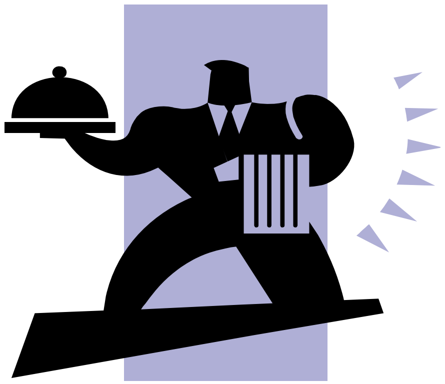
- Typical wireless network are based on infrastructure
  - E.g., GSM, UMTS, WLAN, …
  - Base stations connected to a wired backbone network
  - Mobile entities communicate wirelessly to these base stations
  - Traffic between different mobile entities is relayed by base stations and wired backbone
  - Mobility is supported by switching from one base station to another
  - Backbone infrastructure required for administrative tasks

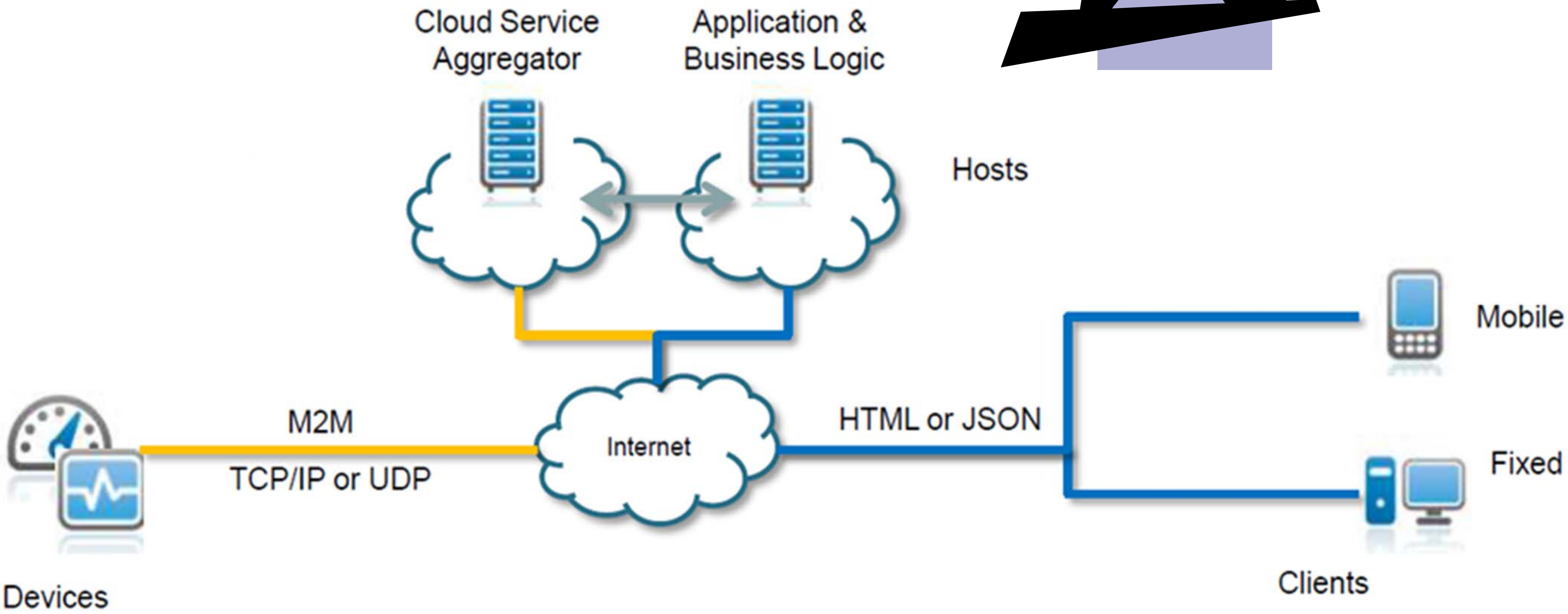



Center of Self-Organizing Software-Platform

# Centralized or Public-Infrastructure

**Big Data**
≈ Big Brothers





# Big Data Generation

- Everyday around 20 quintillion (10^18) bytes of data are produced (Source: [http://www-01.ibm.com/software/data/bigdata/](http://www-01.ibm.com/software/data/bigdata/)).

- This data includes textual content (unstructured, semi-structured, structured) to multimedia content (images, video and audio), on a variety of platforms (enterprise, social media, and sensors).

CSOS Center of Self-Organizing Software-Platform


13

# Problems in Centralized IoT Service

- Though the services operate on a local level, all data are stored and managed from a centralized server,
  → causing a tremendous waste of communication cost and energy.

- Even the highly personalized data such as the biometric signals, personal fitness information, and house utility usage are monopolized by the global service providers
  → 2nd Naver in Korea

- Even though(Everyday around 20 quintillion (10^18) bytes of data are produced) the massive quantity of the raw data, is it really valuable?

- Finally, there is the problem of 'Big Brother' corporations holding monopoly over the entrepreneurial rights, excluding potential newcomers within the industry

THE WALL STREET JOURNAL.

# Forget 'the Cloud'; 'the Fog' Is Tech's Future

**Write to** Christopher Mims at
christopher.mims@wsj.com

I'm as big a believer in the transformational power of cloud computing as anyone you'll meet. Smartphones, which are constantly seeking and retrieving data, don't make sense without the cloud, and any business that isn't racing to push its data and software into someone else's data center is, in my view, setting itself up for disruption by a competitor who is.

Brian Ajhar

But cloud advocates are fond of declaring that 100% of computing will someday reside in the cloud. And many companies are in business to sell you on that notion.

Here's the reality: Getting data into and out of the cloud is harder than most engineers, or at least their managers, often are willing to admit.

The problem is bandwidth. If you're a company simply seeking to save the cost and headache of storing data yourself, the cloud is great as long as all you need to do is transfer data back and forth via high-speed wiring.

But in the world of mass connectivity—in which people need to get information on an array of mobile devices—bandwidth is pretty slow. Any business that sends data to mobile devices, be it airline reservation systems for consumers or business data for a mobile sales force, grapples with the limitations of wireless networks. Overall, according to the World Economic Forum, the U.S. ranks 35th in the world in terms of bandwidth per user.

# Fog Computing – middle ground between devices and the cloud

## Internet of Things Shifts Computing From the Cloud to the Fog

JUL 10, 2014 BY ZEUS KERRAVALA

Mark Twain is rumored to have once said, "The coldest winter I ever spent was a summer in San Francisco". And if you've ever been to San Francisco any time of the year, you'll know that one of the distinguishing characteristics of the city is how cool, damp, and foggy it can be. I've been up to the Marin Headlands a few times to snap a picture of the city behind the Golden Gate Bridge and rarely is it not foggy. In fact, the fog can be so thick sometimes that it's hard to distinguish where the fog ends and the clouds start.

I bring this up because this year's Cisco Live user conference was held in the typically foggy San Francisco, where the company was pushing the concept of 'fog computing' as a way of scaling the Internet of Things (IoT) — fog, San Francisco, Cisco… it all fits neatly.

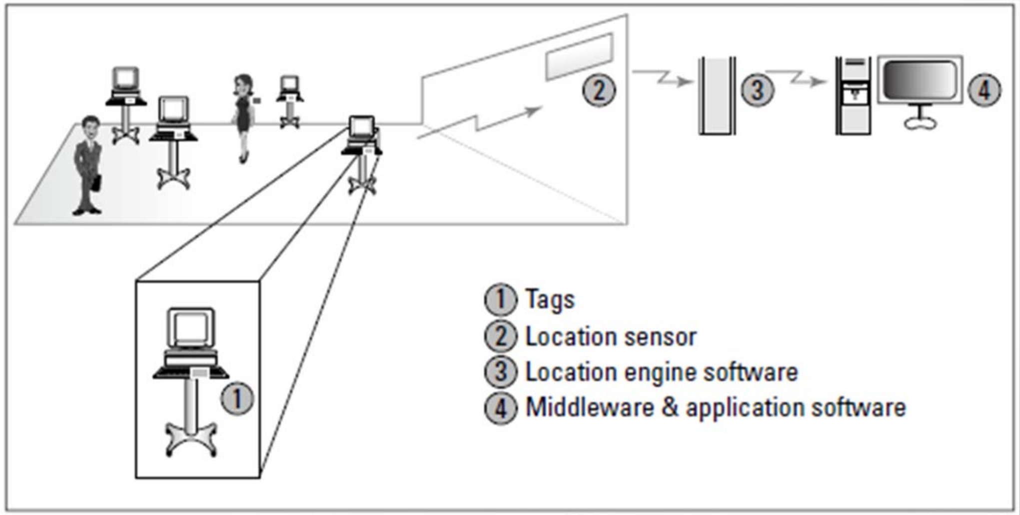
One of the highlights of any Cisco event is the demonstrations, and during this year's demos this concept of fog computing was introduced. While Cisco has been pushing this term for some time, I imagine it was the first time the majority of the over 25,000 in attendance will have been introduced to the concept. In reality, Cisco isn't the only company pushing the concept of fog computing, they're just the only mainstream IT vendor doing so today, which makes sense, given the company is looking at it as an IoT enabler.

For those not familiar with the concept of fog computing, it's a concept where IT resources (computing, storage, application and network) get distributed to get closer to the data. While cloud computing is all the rage today, it does have some scale issues with IoT. Data has to be collected locally and then transported back to the cloud, processed, analyzed, and then action can be taken. If the data set is large and/or the geographic distance to the cloud is too far, this latency can impair the performance of the IoT-based application.

# Part Ⅱ:
# Core technology of Localized IoT(LIoT)

# Understanding How an LIoT Works (1/3)

- **With LIoT, you locate and track people and assets by associating a tag (a small wireless device) with each person or asset.**
- **The parts of an LIoT**
  - **Tags**: A mobile device that's enabled with location technology.
  - **Location sensors**: Devices that usually have a known position.
  - **Location engine**: The software that communicates with tags and location sensors to determine the location of the tags.
  - **Middleware**: The software that resides among the pure LIoT technology components and the business applications.
  - **Application**: The software that interacts with the LIoT middleware and does work users are directly interested in.



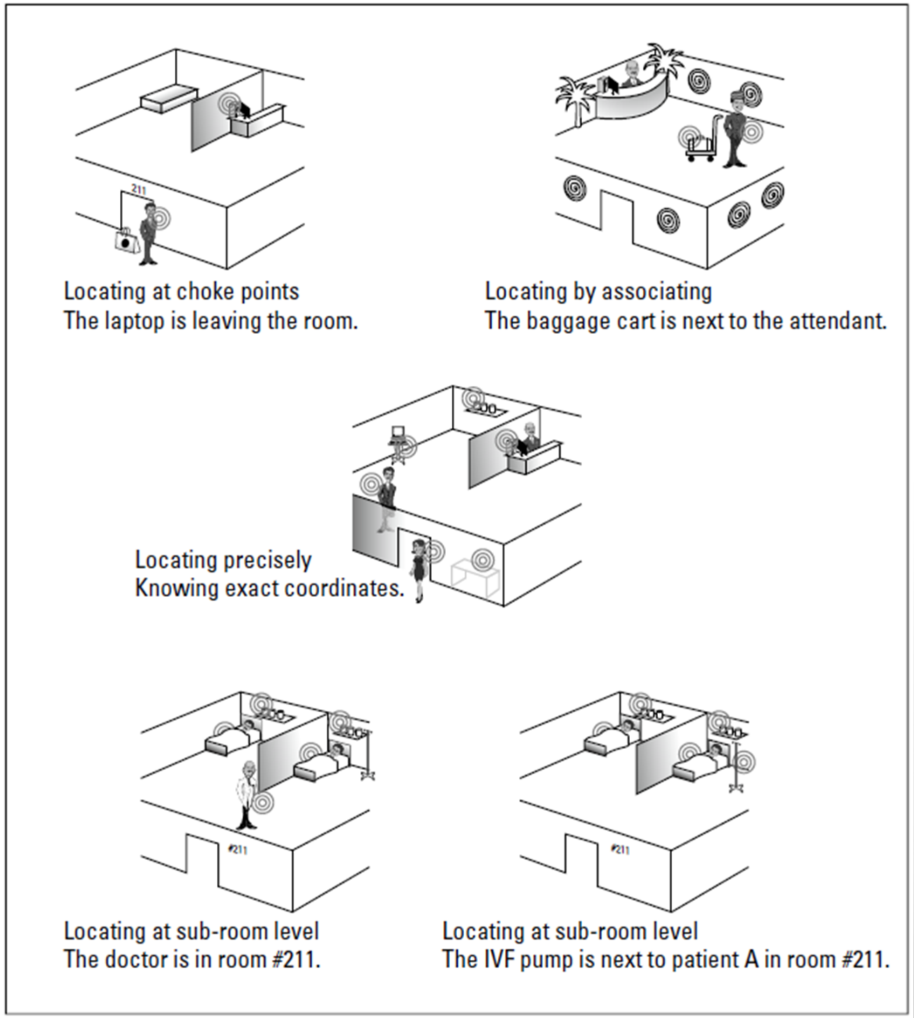

[source: Ajay Malik, "RTLS for dummies" (2009)]

# Understanding How an LIoT Works (2/3)

- **Adding value with bells and whistles**
  - Push buttons: panic button or status indication button
  - Voice to voice: location-based voice messages
  - Buzzers, LEDs, LCD screens, and vibrators
  - Sensors: motion sensors, a temperature sensor or a carbon monoxide sensor
  - Connectors: for getting specific details
  - Writeable memory: for storing some user data for that asset
- **Knowing the location models**
  - Type of the physical location
    - Absolute position: absolute coordinates (latitude, longitude, and altitude)
    - Relative position: distance in three dimensions with reference to a fixed point
    - Symbolic position: presence in a specific area

# Understanding How an LIoT Works (3/3)

- **LIoT location models (cont.)**
  - **Presence-based locating**: the tag location is returned as to whether it's present in a given area.
  - **Locating at room level**: the tag location is returned as presence in a specific room.
  - **Locating at sub-room level**: the tag location is returned as presence in a specific part of the room.
  - **Locating at choke points**: the tag location is returned by a specific choke point(an entry or exit point).
  - **Locating by associating**: the tag location is returned as proximity with respect to another tag.
  - **Locating precisely**: the tag location is pinpointed precisely.




[source: Ajay Malik, "RTLS for dummies" (2009)]

# Locating at Room Level

- For many applications, you need to know the exact location of a person or asset; for others, your needs can be satisfied by just knowing what room a person or asset is in.
- This kind of location reporting is called *symbolic locating*.
- For many places of businesses — such as hospitals, hotels, and schools — a room identifier plays a significant role as a location indicator.
- Because of this, room-level (or subroom-level) locating has always received special attention from researchers and vendors.

# Recognizing the Need for an LIoT

❖ **Tagging assets**
- Locate assets on demand
- Track assets
- Protect assets
- Monitor usage of assets
- Trace assets
- Improve asset utilization
- Manage and plan inventory

❖ **Tagging people**
- Locate people on demand
- Protect and track people
- Monitor people's movements
- Provide emergency response
- Manage evacuations
- Police restricted areas
- Improve workflow
- Improve customer service and response times
- Improve structure of facilities

# Characterizing tags

- **Passive tags: typically passive radio frequency identification(RFID)**
  - Passive RFID tags have no battery and obtain the operational power to transmit data from the RF filed emitted by a corresponding interrogator.
  - The tag doesn't send a traditional RF transmission, but instead communicates back to the interrogator with a technique known as backscatter.
- **Semi-passive tags (=Battery assisted passive tags)**
  - To send data to the interrogator, they use the same backscatter technique as passive tags.
  - The battery is used to either monitor environmental conditions or offer greater range and reliability than passive tags. (Not used to generate RF energy.)
- **Active tags**
  - They contain an onboard radio(transmitter or transceiver of RF) and are typically powered by an internal battery.
  - They usually have a long range and can two-way communicate without being prompted. So these tags can be located in realtime.
  - Because these tags have batteries, battery life is an important concern.

# Making Sense of Location Sensors

❖ **Location sensors (**≈ Location Anchor Node)

– devices within an LIoT that typically have a known position and detect the location of tags.

– Location sensors locate tags by using a **physical parameter**, or a measurement, that exists between the sensors and the tags.

– Depending on the technology, these sensors can be some devices that leverage the **infrastructure already present** in the facility, or they can be **a brand new infrastructure** added to the facility.

# Inspecting the Location Engine

## ❖ The location engine

– the software that communicates with tags and/or location sensors to **determine the location of tags** and report it to middleware and/or applications.

– Location determination basically consists of:

  - **Ranging techniques** to estimate distance between the tag and a set of location sensors
  - **Position estimation** techniques that derive the position of the tag

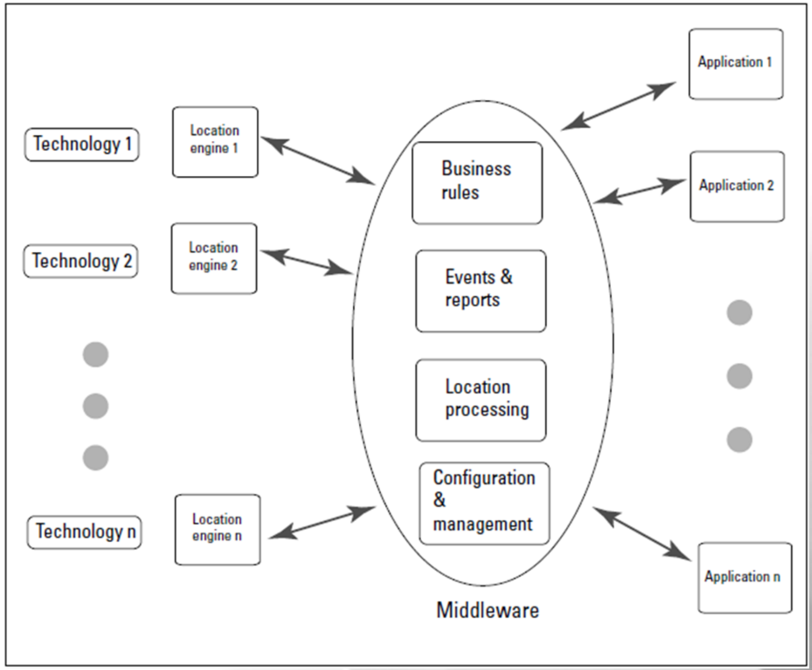
# Ranging techniques

- **Proximity**: Proximity measures the nearness to a known set of points.
- **Time of Arrival(TOA)**: TOA basically makes use of the time it takes for a signal to travel from the location sensors to the tag or vice versa.
- **Angle of Arrival(AOA)**: AOA uses direction-sensitive antennas by the receivers to determine the direction of a signal from the tag to the location sensor.
- **Time Difference of Arrival(TDOA)**: TDOA measures the difference in transmission times between signals received from each of the transmitters to a tag or vice versa.
- **Time of Flight(TOF)**: The TOF method uses measured elapsed time for a transmission between a tag and a location sensor based on the estimated propagation speed of a typical signal.
- **Round Trip Time(RTT)**: RTT uses the total time fro a signal to start from the location sensor and the acknowledgement to be received back.
- **Received Signal Strength Indicator(RSSI)**: RSSI is a measurement of the power present in a received radio signal.

CSOS Center of Self-Organizing Software-Platform

# Understanding Middleware

- **In the context of an LIoT, middleware is the software that resides among the pure LIoT technology components(tags, sensors, and the location engine) and the business applications.**
- **Middleware functionality**
  - Tag, location sensor, and location engine management
  - Location data management:
    - Filtering
    - Consolidation
    - Routing
    - Storing
  - Application integration
  - Business rules and process management
  - Architecture scalability and administration




[source: Ajay Malik, "RTLS for dummies" (2009)]

Center of Self-Organizing Software-Platform

# Related Research Work(1/2)

❖ **Active Badge System**
- Developed at the Olivetti Research Laboratory (now AT&T Cambridge) is an indoor badge-sensing system.
- The location of badge is based on which infrared receiver(s) captured the identifier from the tag.

❖ **Active BAT Location System**
- Developed by AT&T researchers after Active Badge to provide more accurate locating.
- The locating infrastructure consists of a grid of ultrasonic receivers (location sensors) in ceiling tiles connected by a wired serial network and a radio transmitter (controller).
- The location is computed using multilateration in which distance between the receiver to the bat is estimated by using the ToF of the ultrasonic signal.

❖ **Cricket**
- Makes use of RF and ultrasound, much like the Active Bat system.
- The tags use the time difference between the receipt of the first bit of RF information and the ultrasonic signal to determine the distance to the beacon.

# Related Research Work(2/2)

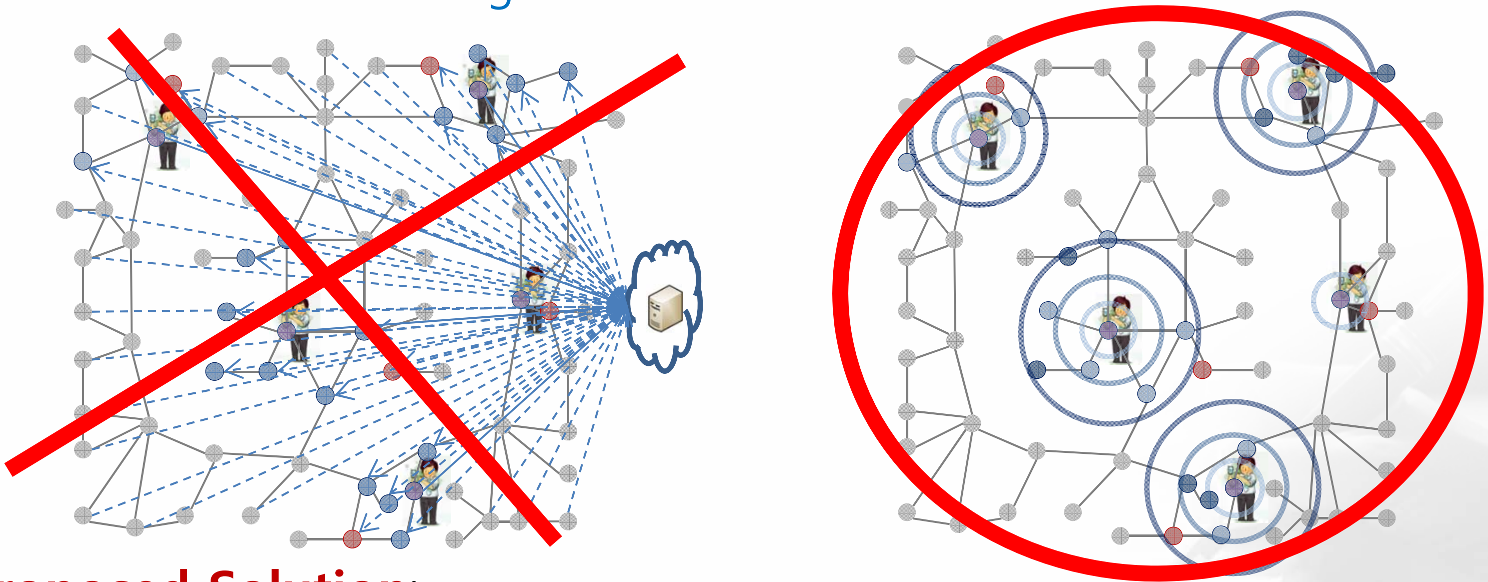
- **RADAR**
  - Building-wide tracking system based on the IEEE 802.11 WaveLAN wireless networking technology.
  - The locating infrastructure consists of RF transceivers called the base station (location sensors).
  - The location engine uses the signal strength and signal-to-noise ratio of the signals to compute the 2D position within the building by combining empirical measurements (also referred to as scene analysis or fingerprinting).
- **LANDMARC**
  - The locating infrastructure consists of RFID readers at fixed locations and reference tags.
  - By placing readers in fixed locations, the whole region can be divided into a number of sub-regions.
  - The readers continuously scan (read) for the tags. In this process, it is constantly reading the reference tags.
- **NearMe**
  - Instead of locating tags using readers, readers are located using tags.
  - A grid of inexpensive passive RFID tags (location sensors) is spread all over the building, and room identification is stored in each tag.

# Part III :
# Self-Organized Localizing Iot Messaging(SLIM) hub

# →KNU CSOS Solution

# Traditional Centralized LIoT System

- **Disadvantages**
  - It requires global knowledge to complete geographical indoor map.
  - It has the problem with a single point of failure.
  - When numerous lookups are concurrently requested,
    It causes the traffic congestion.

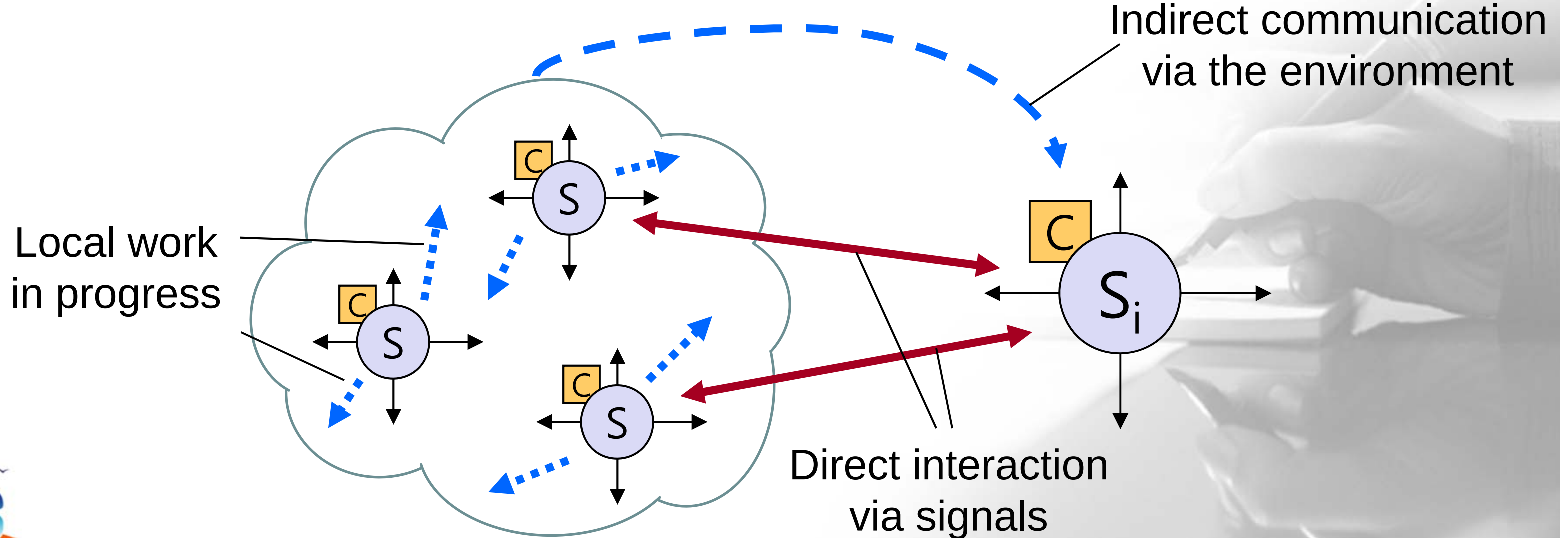


➔ **Proposed Solution**:
Fully distributed **S**elf-**o**rganizing **S**ervice **p**latform **(SoSp)**

# Self-Organization in Natural Systems

# Characteristics of Self-Organizing Systems

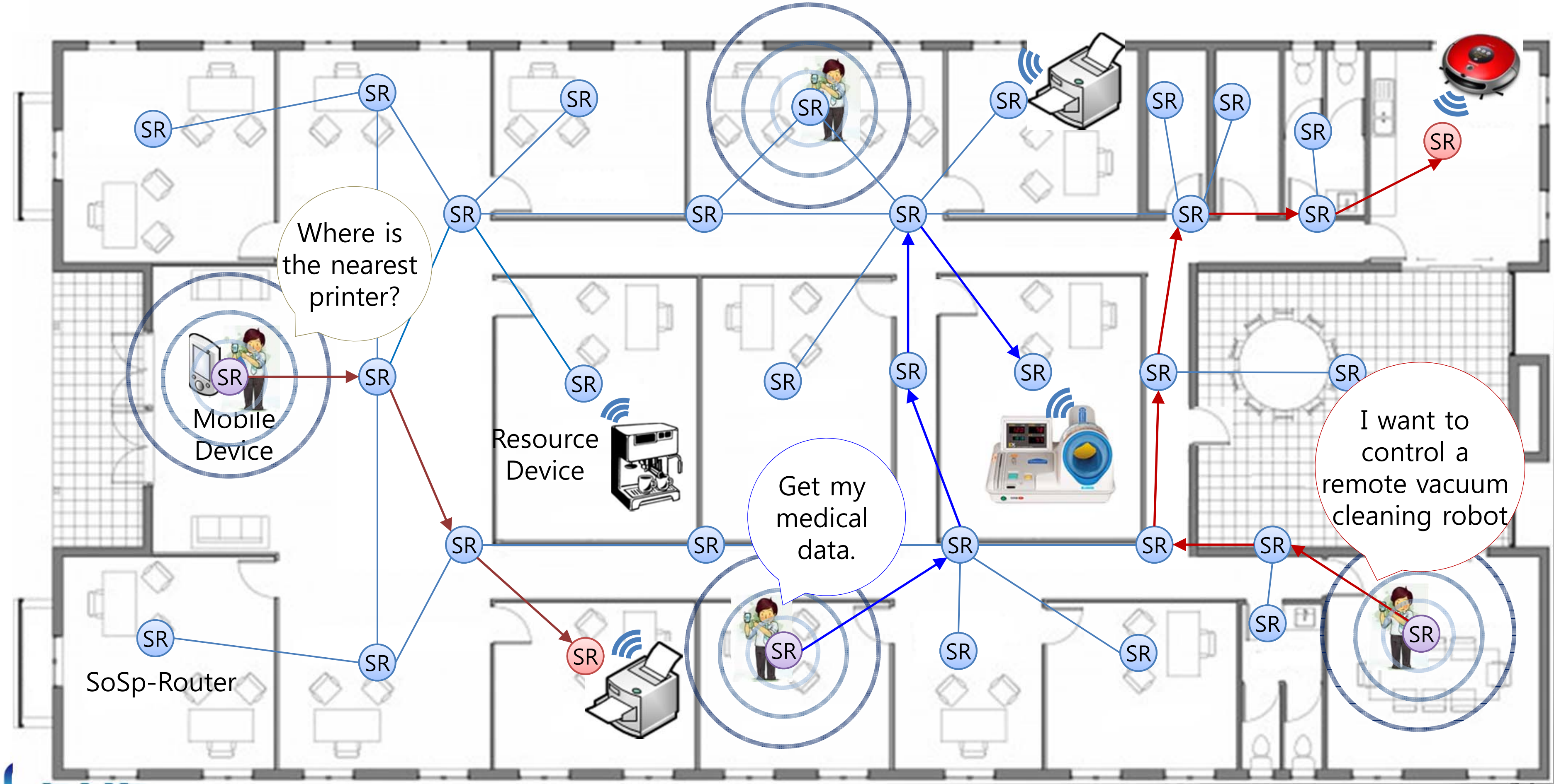
- **Introducing Among Individuals and with the Environment**
  - Direct communication among neighboring systems (gossiping)
  - Indirect communication via the environment (stigmergy)
  - Interaction with (stimulation by) the neighbors and environment
    - Identifying location of itself and neighbors
    - Clustering and collaboration
    - Delay Tolerant Communication
  - Opportunistic service
    (activated when the time, the location, the users just matched)



# Proposed LIoT Architecture → SoSp Service Infra

❖ **The roles of the SoSp-Router (SR1) :** **SoSp-Router = SLIM hub**

1) Anchor (location sensor) node for location positioning.
2) Access point for various wireless communications (such as WiFi, Bluetooth, ZigBee, and so on).
3) Node in a distance graph to configure an overlay network.



# Assumption

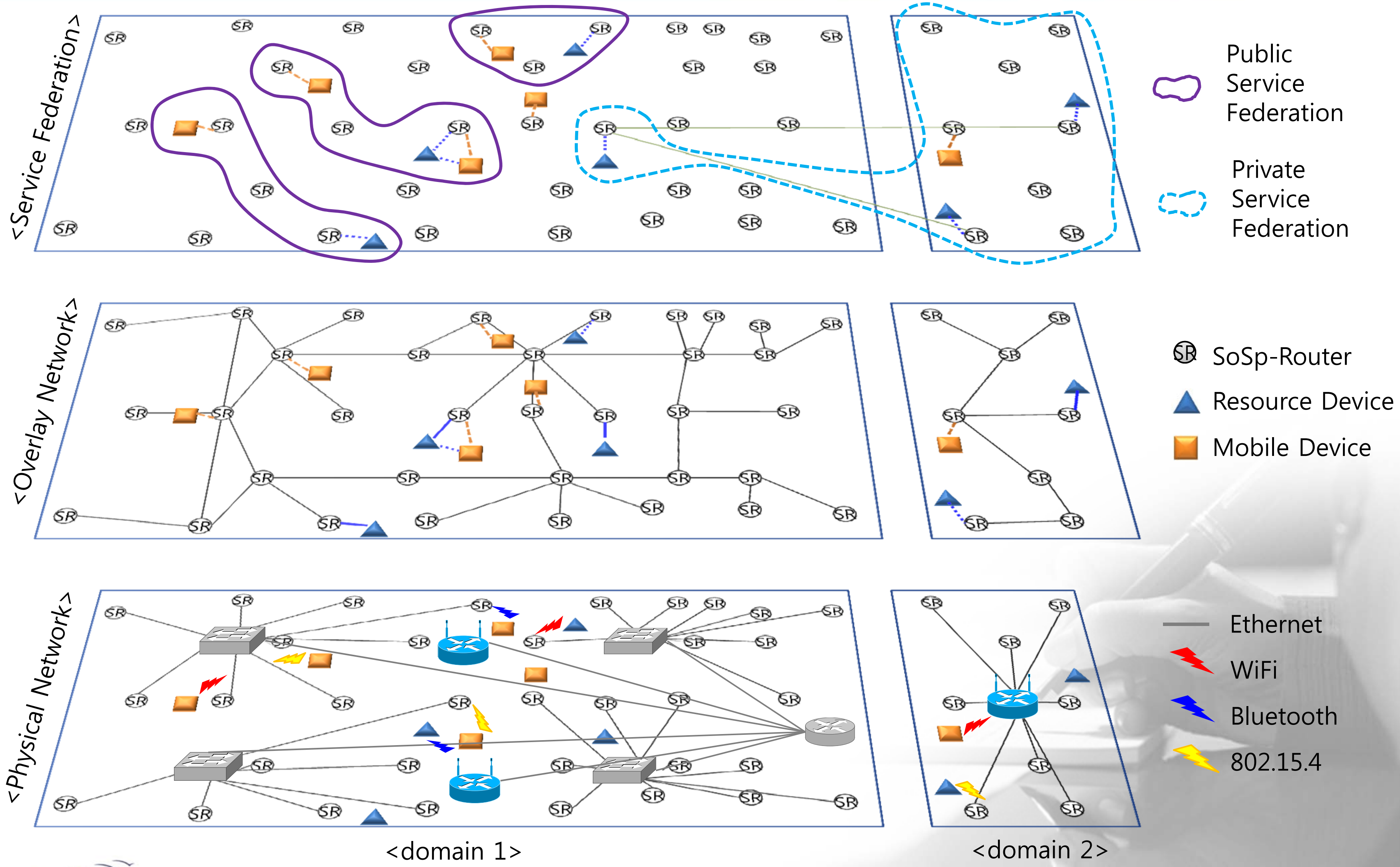
❖ **Stationary Nodes(R) (SoSp-Router)**

- Know current location → anchor node
- Know neighbor stationary nodes → neighbor SR
- Can notify current location to the mobile nodes which are located in its area
- Can be added or deleted dynamically
- No. centralized management server(self-organizing overlay network)
- Several wireless and wired protocol support →WiFi,Zigbee,BT4.0,ANT,etc

❖ **Mobile Nodes (SoSp-Client) → mobile tags**

- ID-client : act as a Personal ID device
- Resource-client: device supply a certain service
- Can move freely
- Low power consumption is essential(should be activated under the opportunistic computing manner)
- Communication protocols:
  - Smart phone(pad) → WiFi
  - Watch, tags, etc → Zigbee,BT.4.0, ANT+,etc

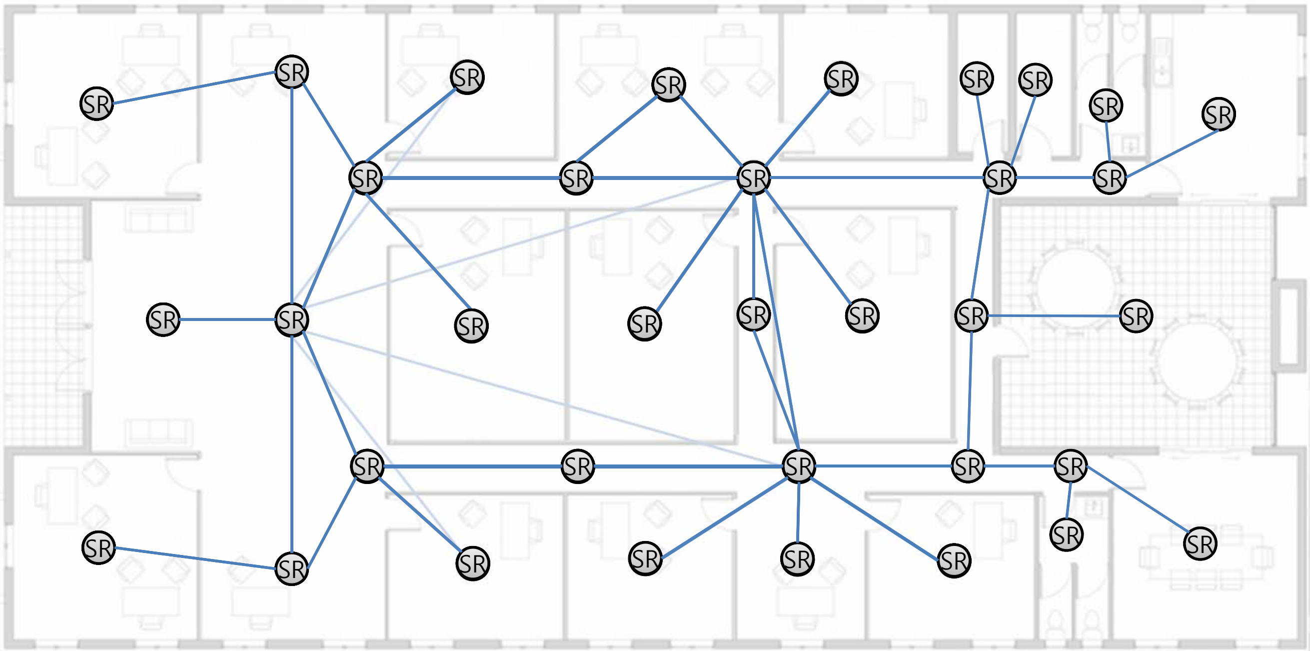
# Configuring SoSp Overlay Network




Center of Self-Organizing Software-Platform

# Dynamic Space Expansion & Reduction

- A method of expanding and reducing for supporting scalability in self-organization property
- Insert / Delete / Re-activate



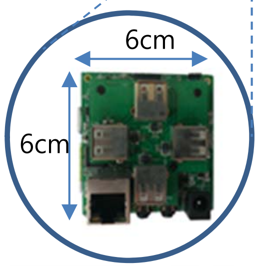
# Distance Graph based Overlay Network

- **Node**: SoSp-Router
- **Edge**: Physical distance between two neighbor SoSp-Routers path

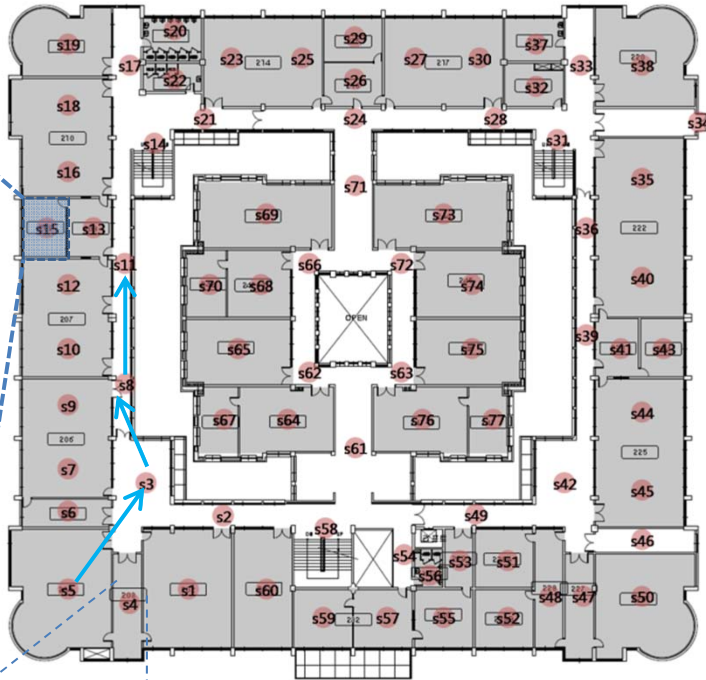
→ Each SoSp-Router has only the information of neighboring nodes connected through physical path.

**SoSp-Clients**
(Resources, ID)

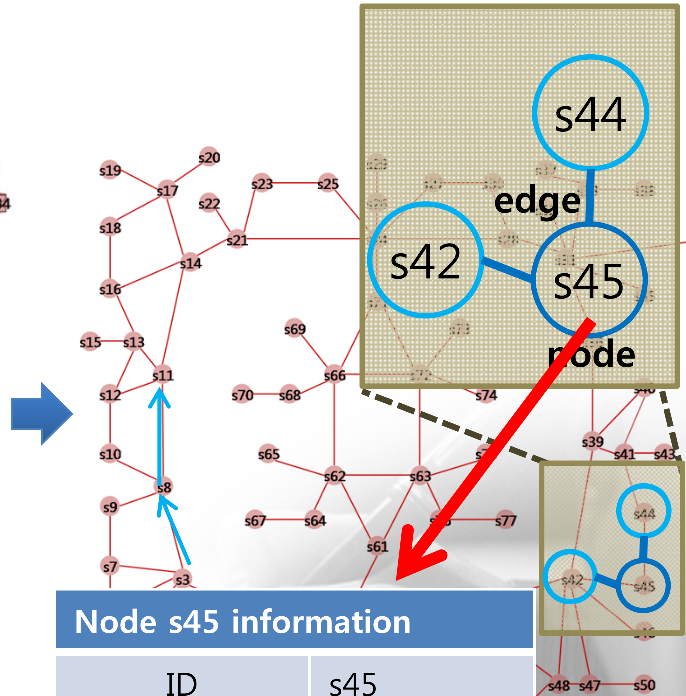
<Room>




**SoSp-Router**



<Floor plan>

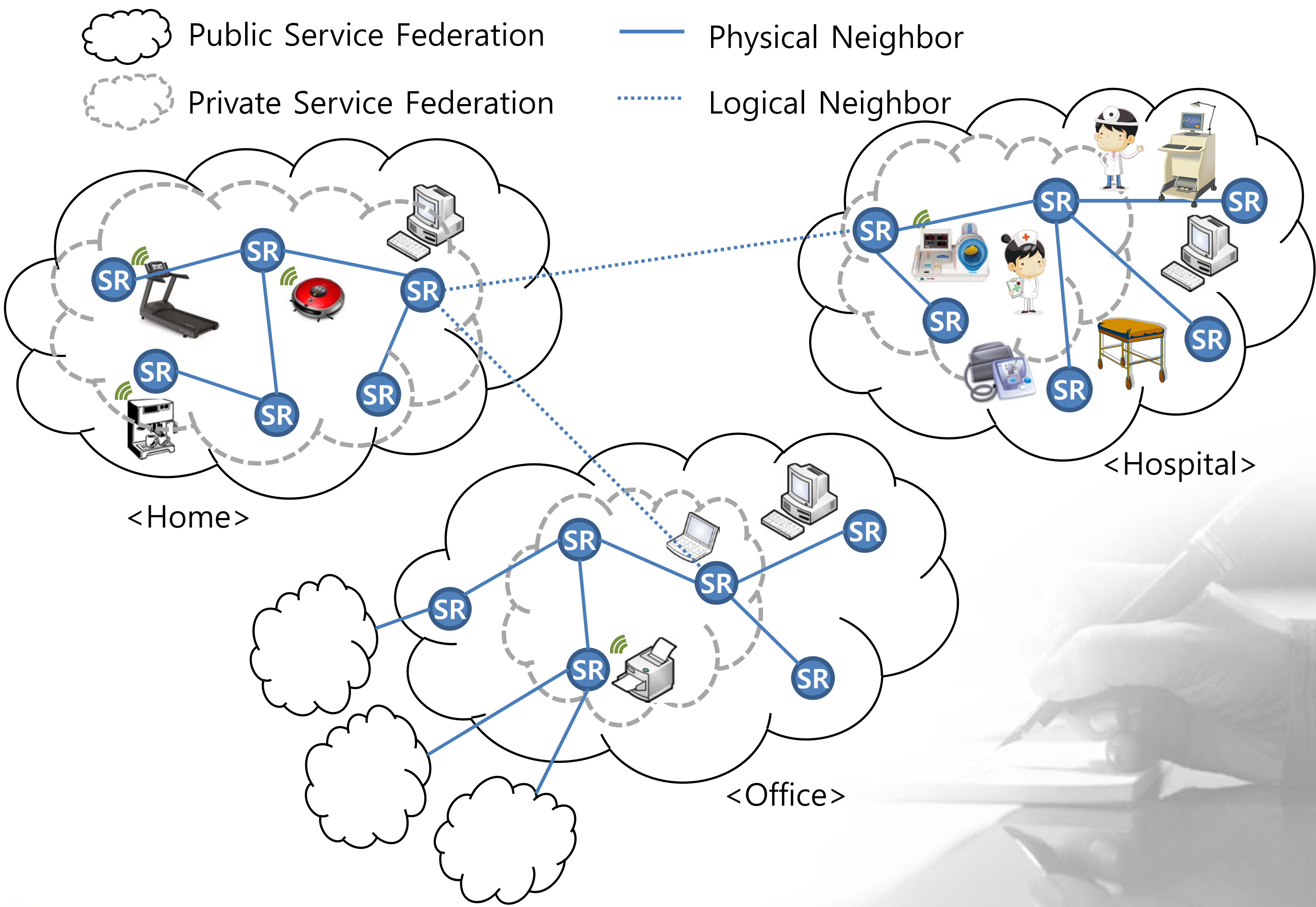



aph>

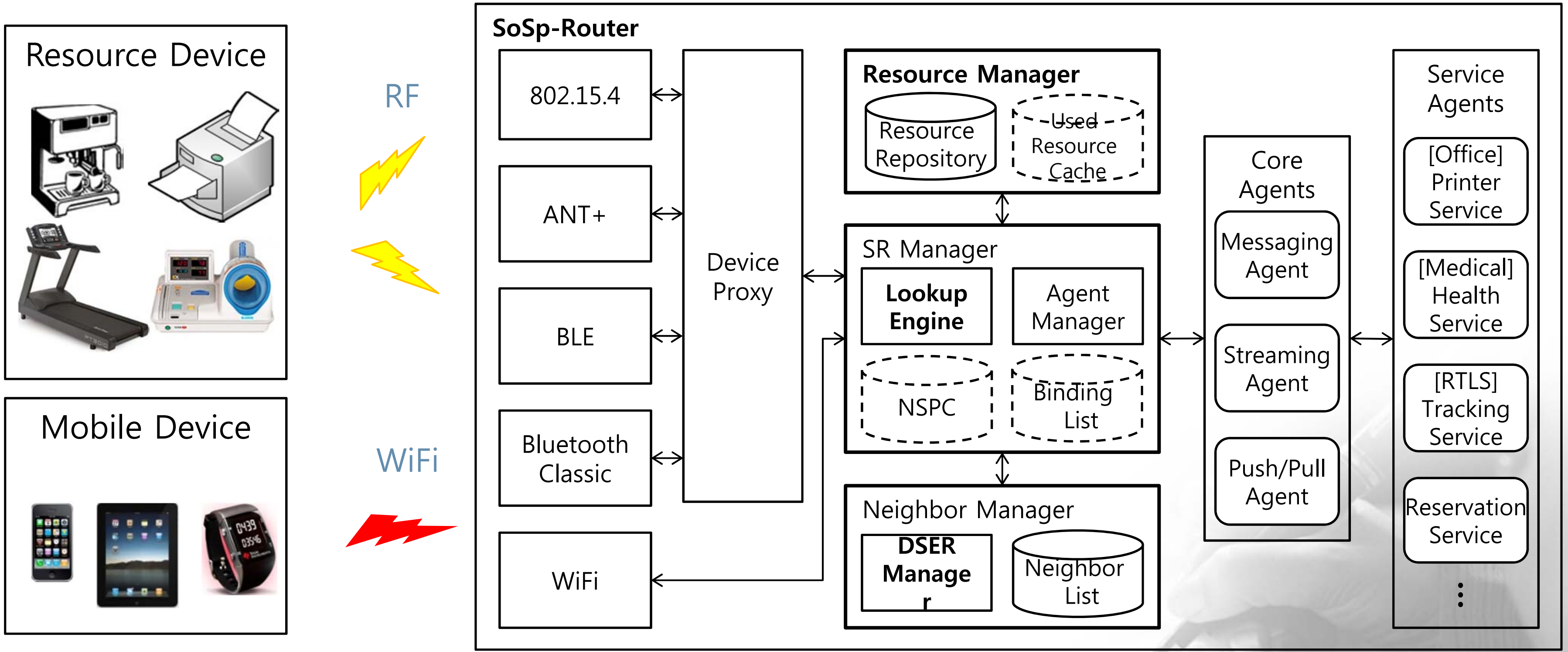
| Node s45 information | |
| --- | --- |
| ID | s45 |
| IP | 192.168.0.5 |
| Group | RTLAB |
| Location | E10-501 |
| Neighbor List (dist, node_id) | { (2, s42), (3, s44) } |

Center of Self-Organizing Software-Platform

# Long-Distance Service Space Extension



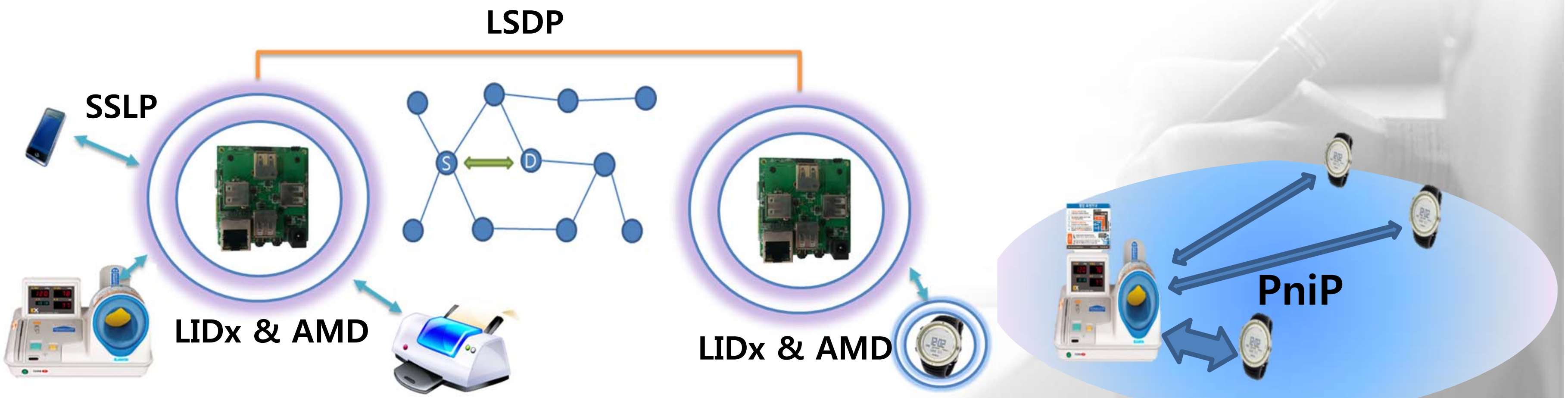
# Middleware in SoSp-Router




Center of Self-Organizing Software-Platform

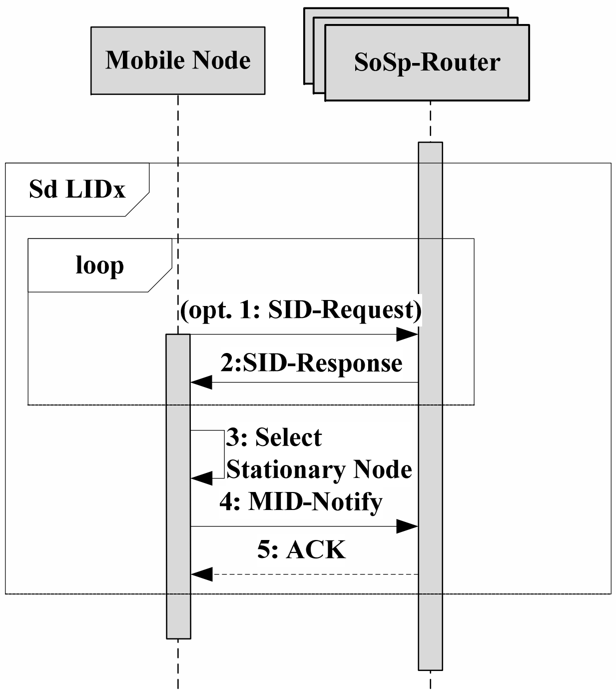
# Suggested Protocols in SoSp Infra

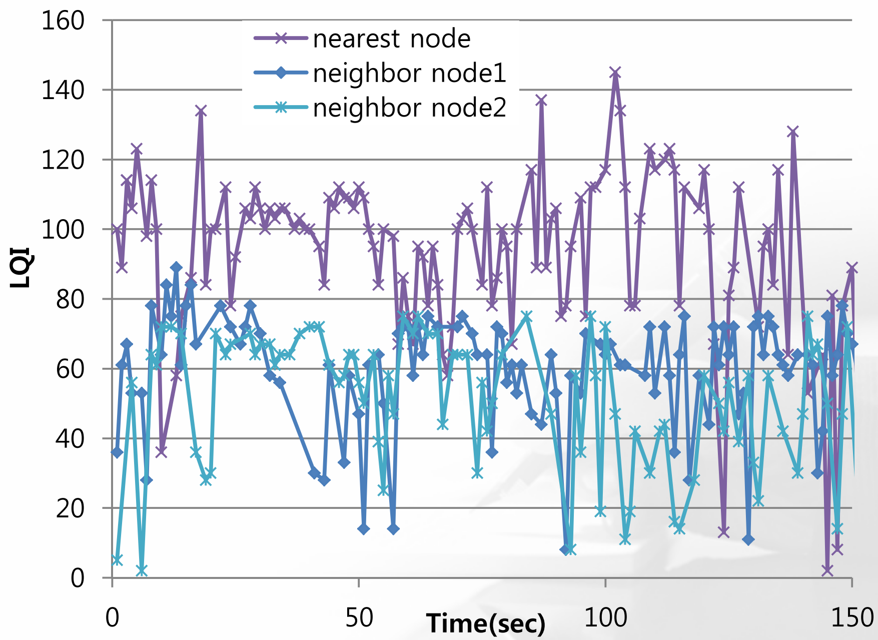
- SoSp-Router ↔ Smart Mobile Device(Smart phone or pad)
  - **SSLP** (**S**martDevice to **S**oSp-Router Service **L**ookup **P**rotocol)
- SoSp-Router ↔ SoSp-Router
  - **LSDP** (**L**ocation-based **S**ervice **D**iscovery **P**rotocol)
- SoSp-Router ↔ SoSp-Client
  - **LIDx&AMD** (**L**ocation **ID** e**x**change and **A**synchronous **M**essage **D**elivery)
- SoSp-Client(Resource) ↔ SoSp-Client(Mobile ID device)
  - **PniP** (**P**roximity based **N**eighbor **I**dentification **P**rotocol)



# Localization Protocol

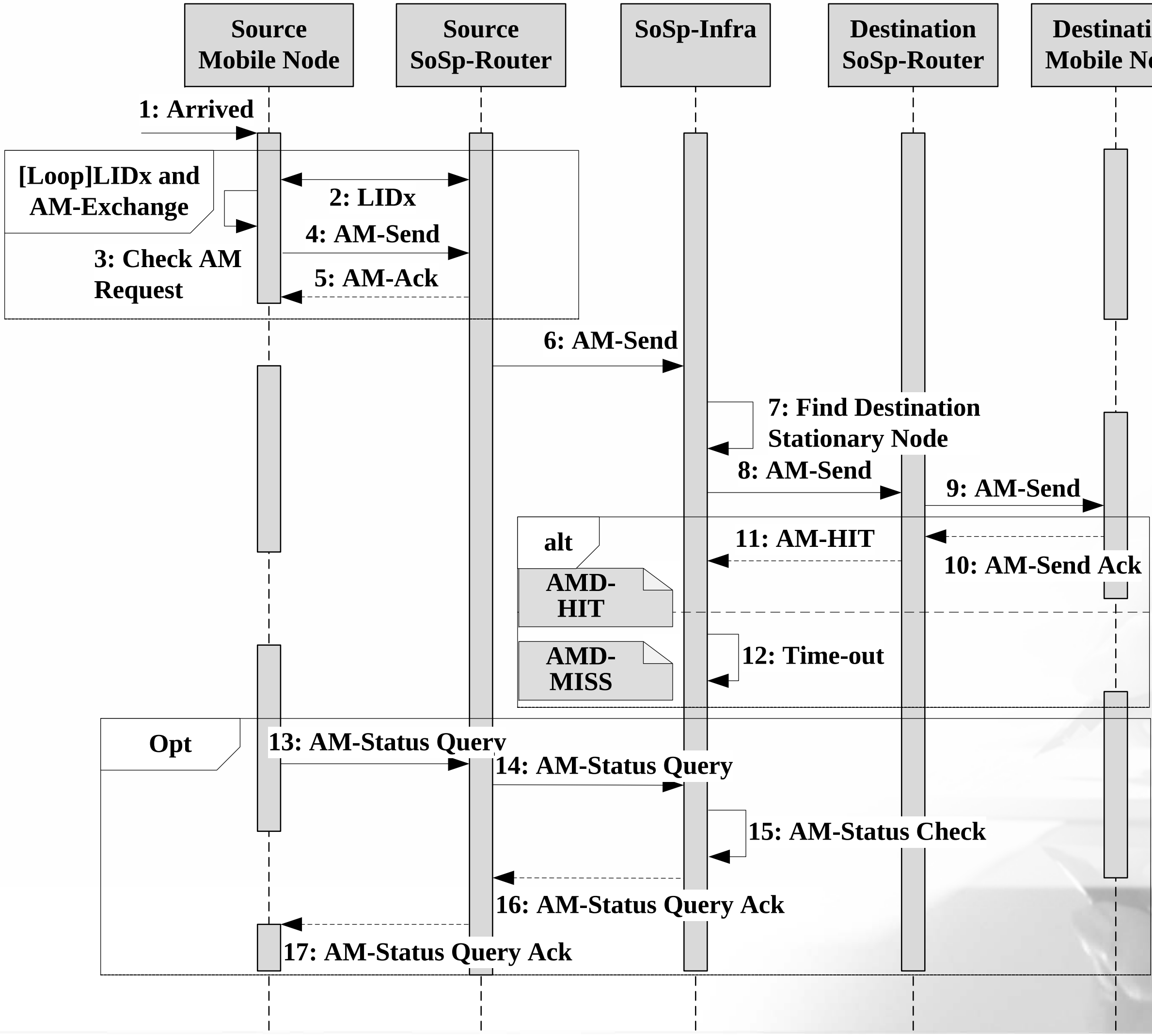
❖ **LIDx (Location ID exchange) protocol**
- ID exchange between a mobile node and SoSp-Routers
- Cell based localization
- Support numerous mobile nodes



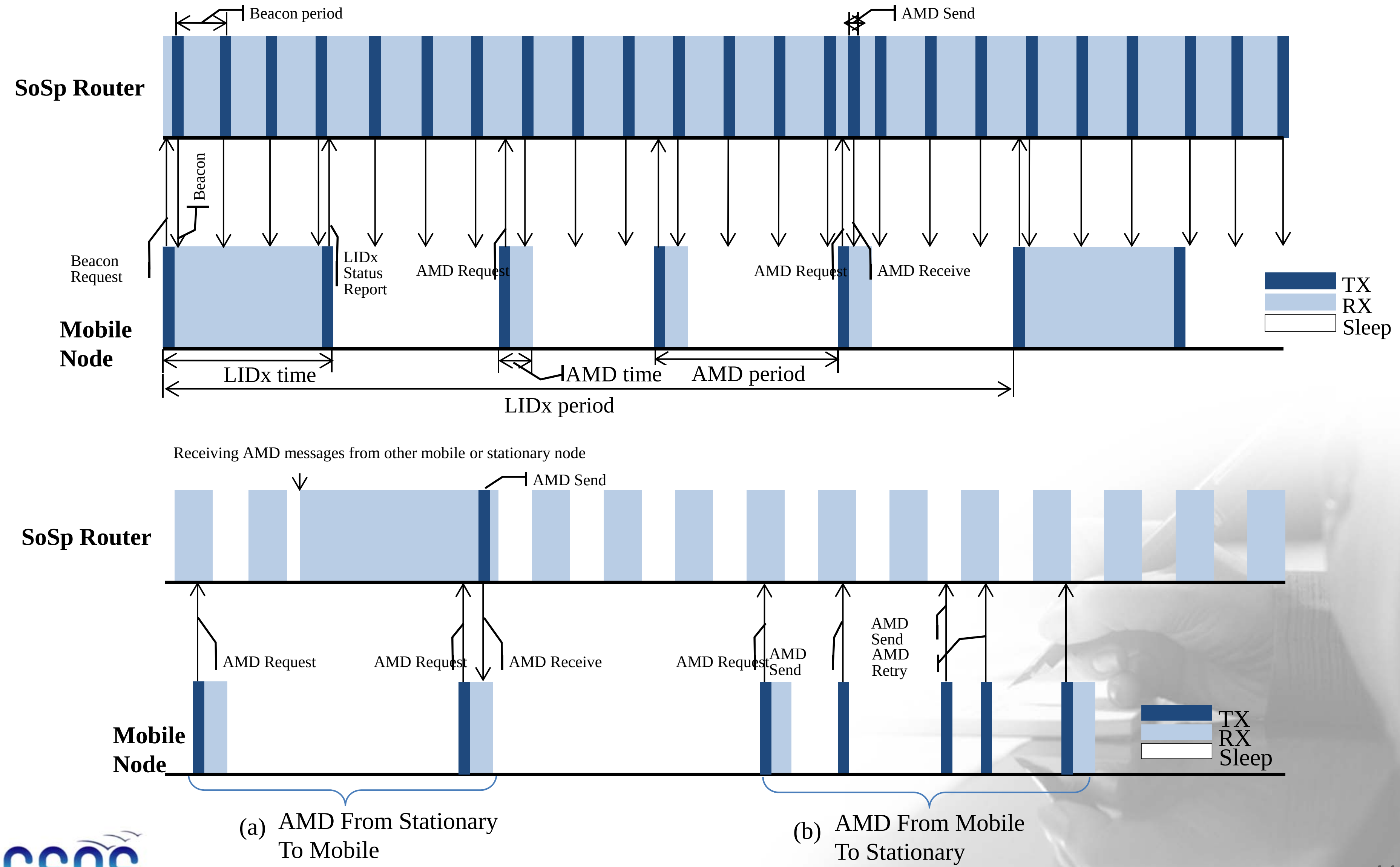
# Message Delivery Protocol

❖ AMD (Asynchronous Message Delivery) Protocol




CSOS
Center of Self-Organizing Software-Platform

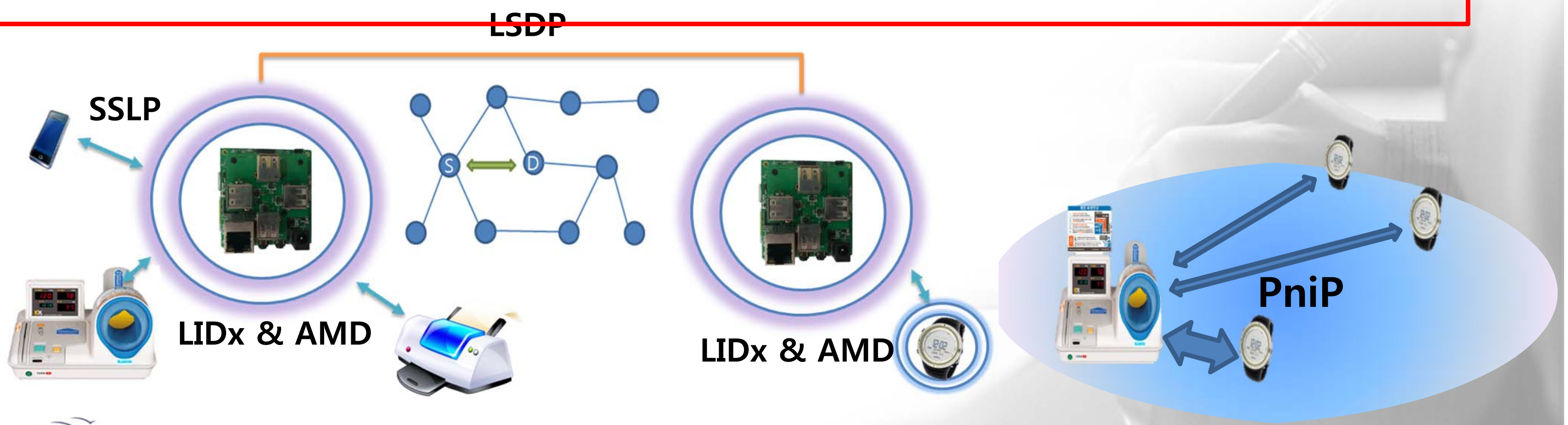
# Timeline of LIDx & AMD




CSOS Center of Self-Organizing Software-Platform

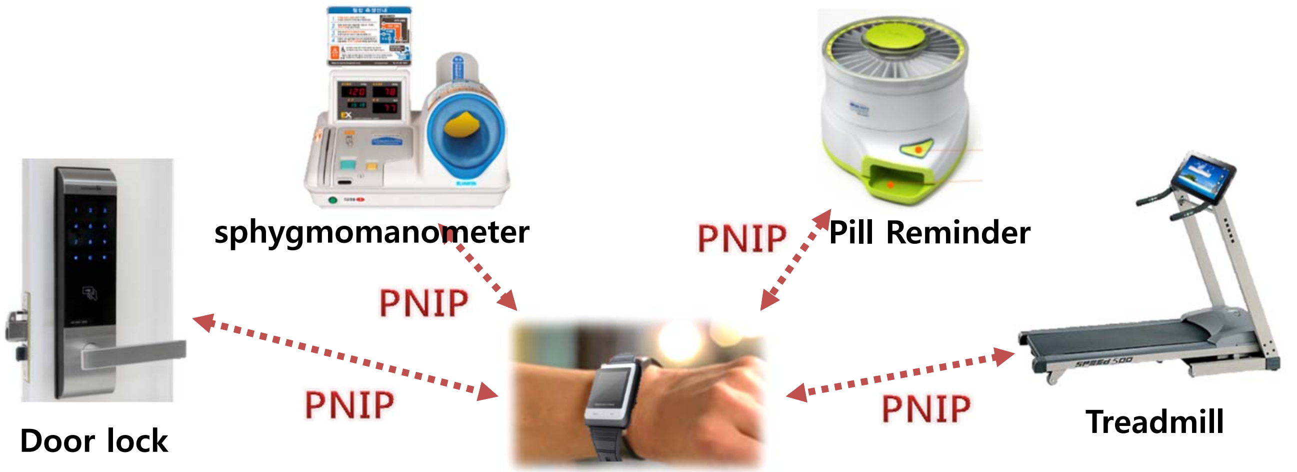
# Suggested Protocols in SoSp Infra

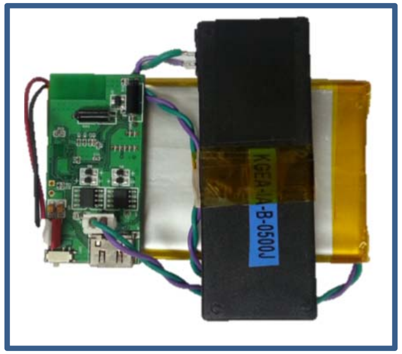
- SoSp-Router ↔ Smart Mobile Device(Smart phone or pad)
  - **SSLP** (**S**martDevice to **S**oSp-Router Service **L**ookup **P**rotocol)
- SoSp-Router ↔ SoSp-Router
  - **LSDP** (**L**ocation-based **S**ervice **D**iscovery **P**rotocol)
- SoSp-Router ↔ SoSp-Client
  - **LIDx&AMD** (**L**ocation **ID** e**x**change and **A**synchronous **M**essage **D**elivery)
- SoSp-Client(Resource) ↔ SoSp-Client(Mobile ID device)
  - **PniP** (**P**roximity based **N**eighbor **I**dentification **P**rotocol)




CSOS Center of Self-Organizing Software-Platform

# Concept of PNIP



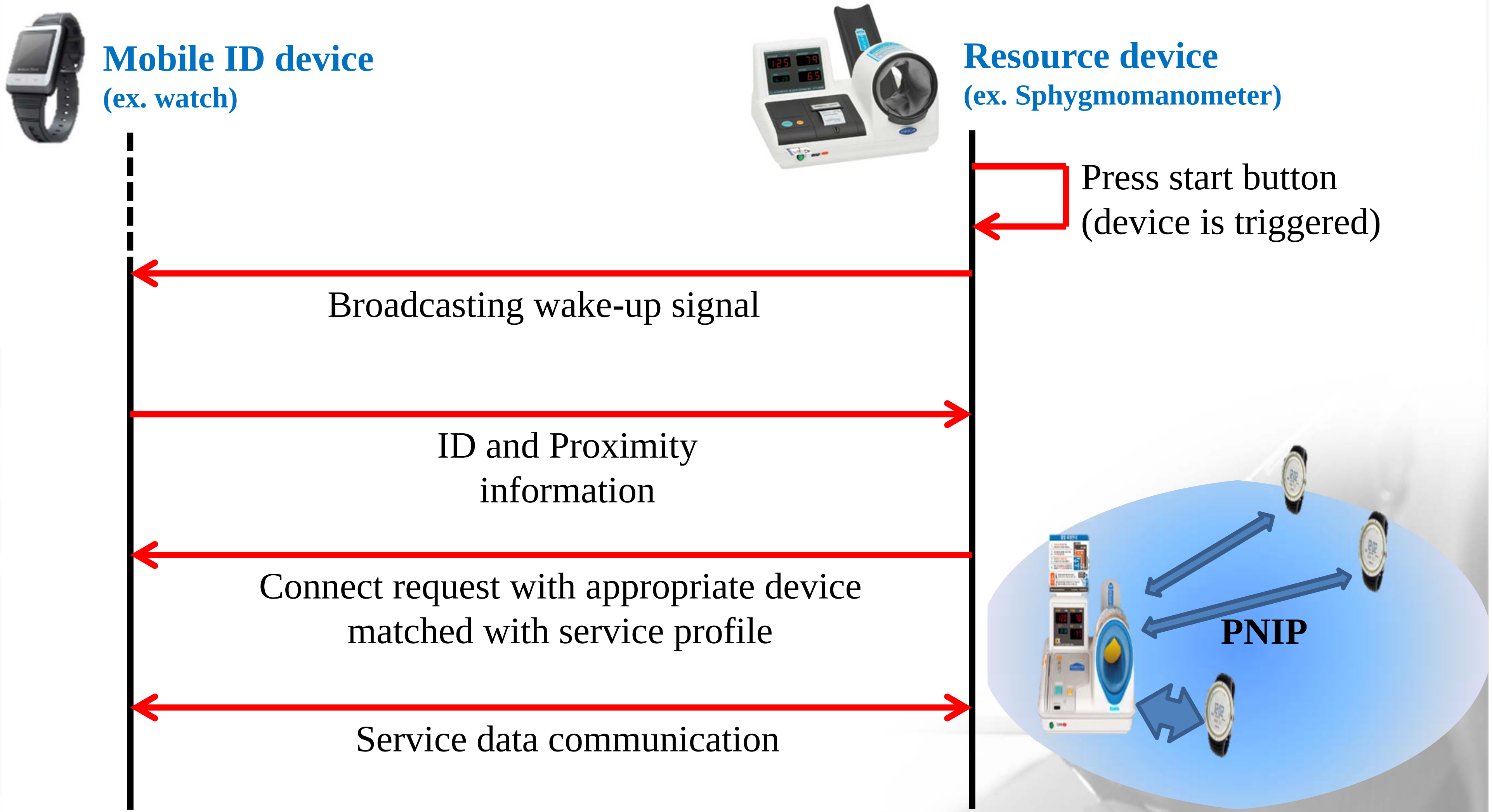
PNIP Host Module

**PNIP** (Proximity-based Neighbor Identification Protocol)

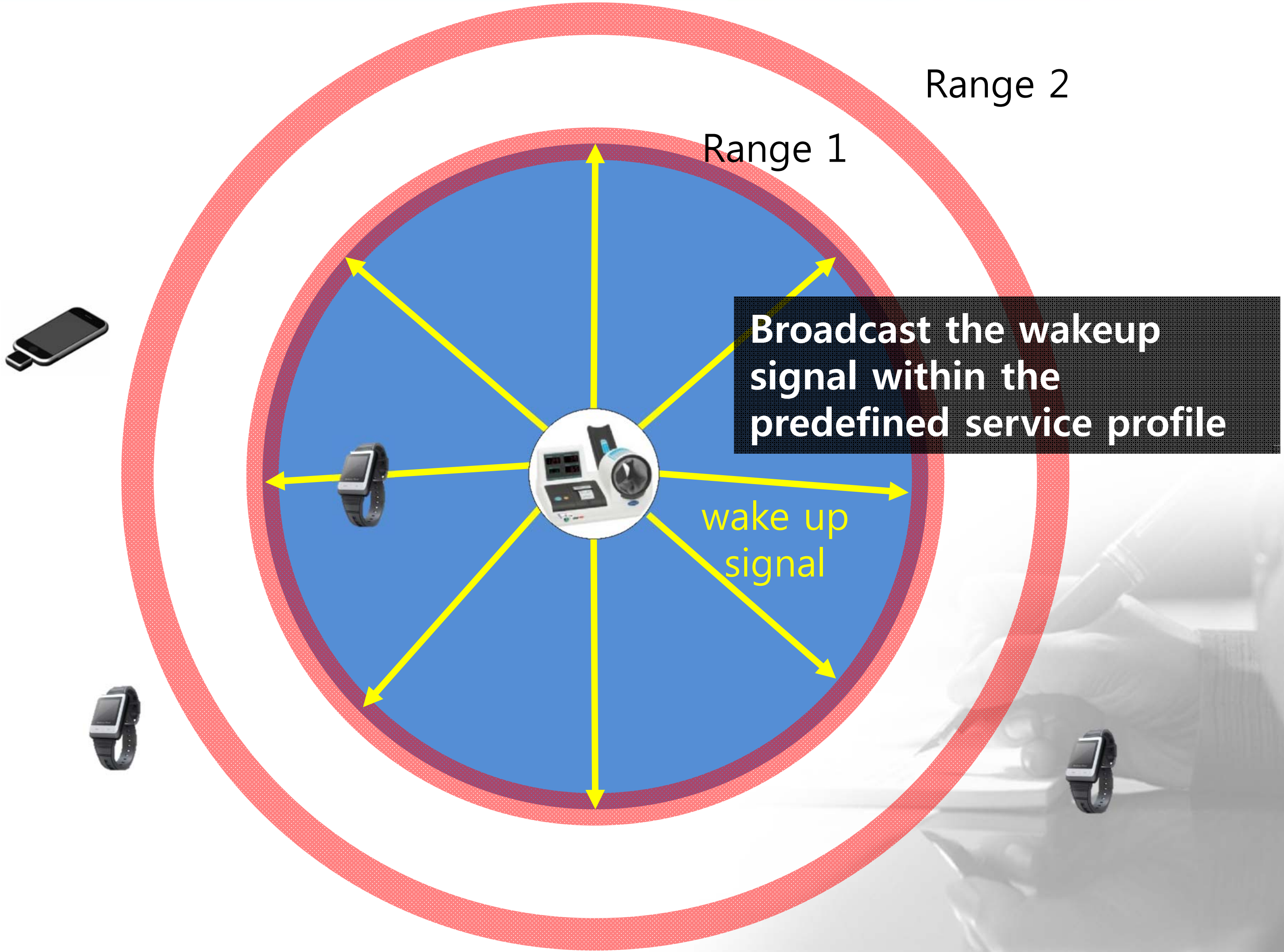
**Opportunistic Communication** | **User Identification** | **Low power Consumption**

- **PNIP Client (ex: watch) has unique ID of user.→ ex) Who was use the burner?**
- **When a resource device is used (ex: burner, health machine), PNIP host and client automatically exchange the ID and data.**
- **Requirement to achieve low power consumption: Opportunistic Direct M2M**

# Basic Sequence diagram of PNIP

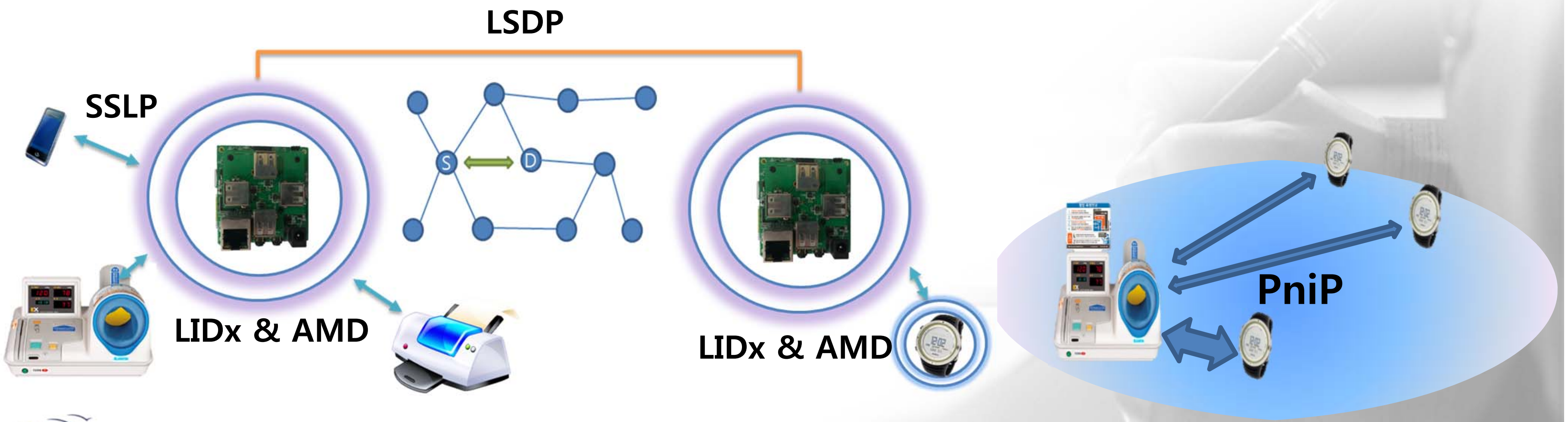


# Service profile based on the range



# Suggested Protocols in SoSp Infra

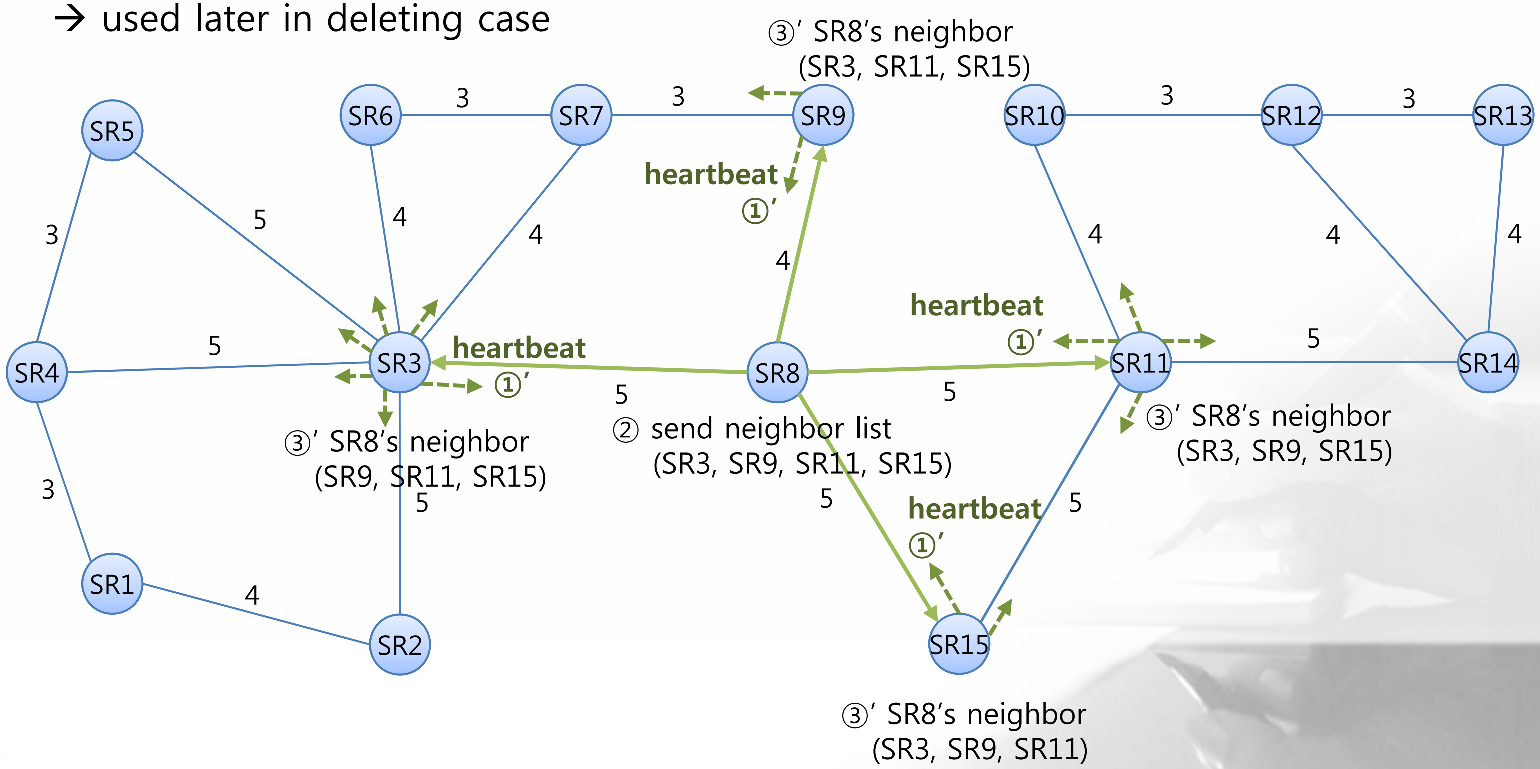
- SoSp-Router ↔ Smart Mobile Device(Smart phone or pad)
  - **SSLP** (**S**martDevice to **S**oSp-Router Service **L**ookup **P**rotocol)
- SoSp-Router ↔ SoSp-Router
  - **LSDP** (**L**ocation-based **S**ervice **D**iscovery **P**rotocol)
- SoSp-Router ↔ SoSp-Client
  - **LIDx&AMD** (**L**ocation **ID** **ex**change and **A**synchronous **M**essage **D**elivery)
- SoSp-Client(Resource) ↔ SoSp-Client(Mobile ID device)
  - **PniP** (**P**roximity based **N**eighbor **I**dentification **P**rotocol)




Center of Self-Organizing Software-Platform

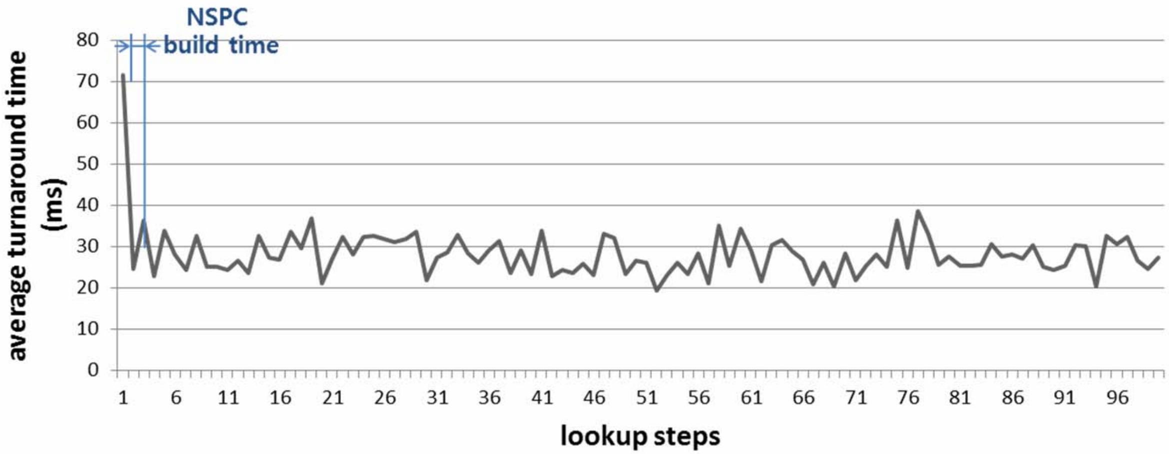
# Heartbeat of SoSp-Router for Monitoring Neighbors

- **Heartbeat for detecting the removed neighbors:**
  - A periodic signal for detecting the changed neighbors
- **Heartbeat request & reply**
  - ① send periodically msg of 'request_heartbeat' to all neighbors
  - ② response heartbeat msg including each current neighbor_list
  - ③ have provisionally neighbors' neighbor_list
    - → used later in deleting case

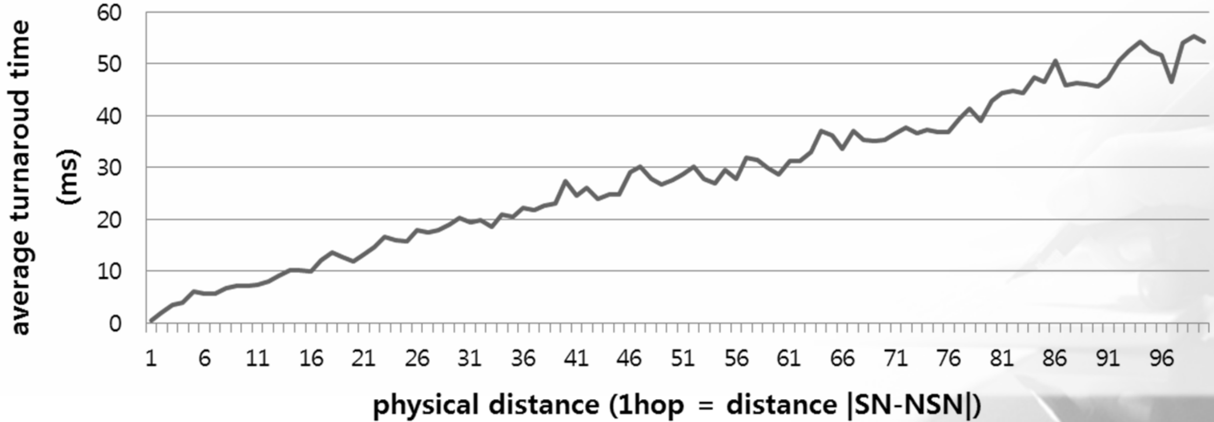


# Scalability Evaluation : Simulation (1/4)

- Response Time 1) average turnaround time by lookup steps in one SoSp-Router



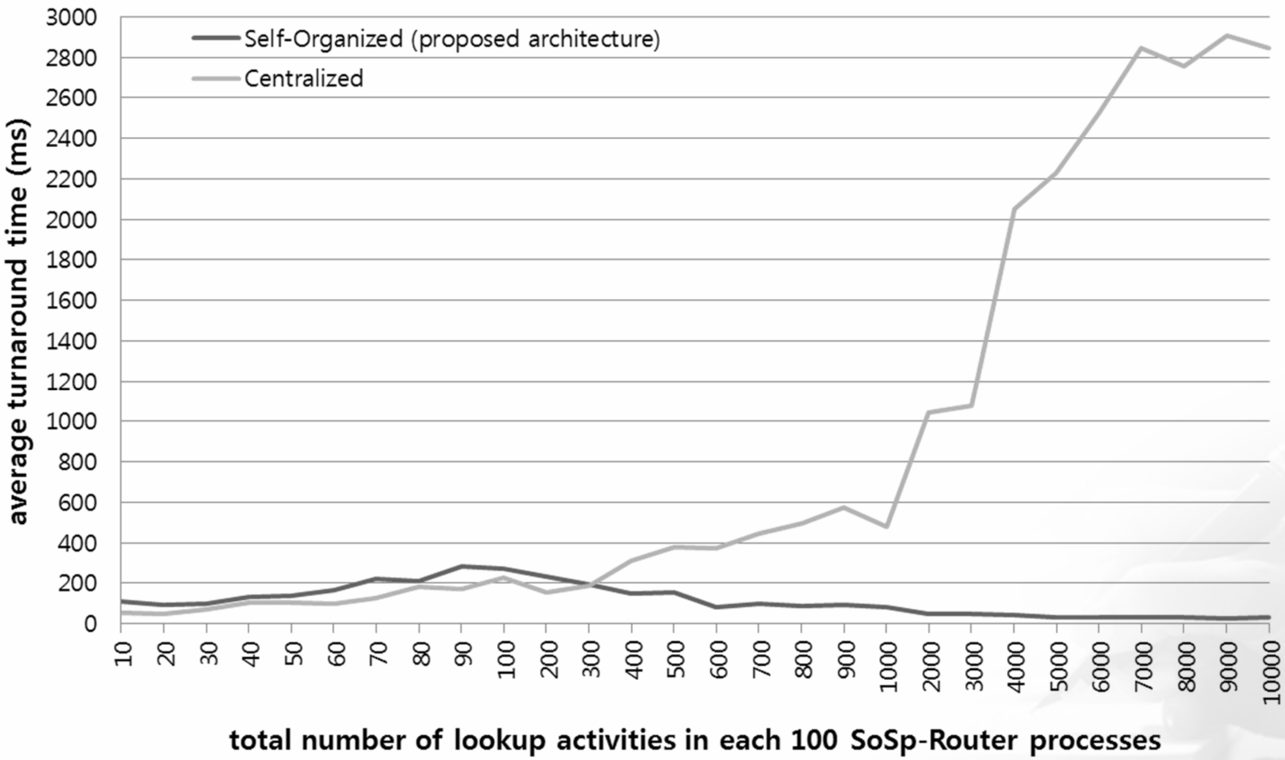

- Response Time 2) average turnaround time by distance in one SoSp-Router




Center of Self-Organizing Software-Platform

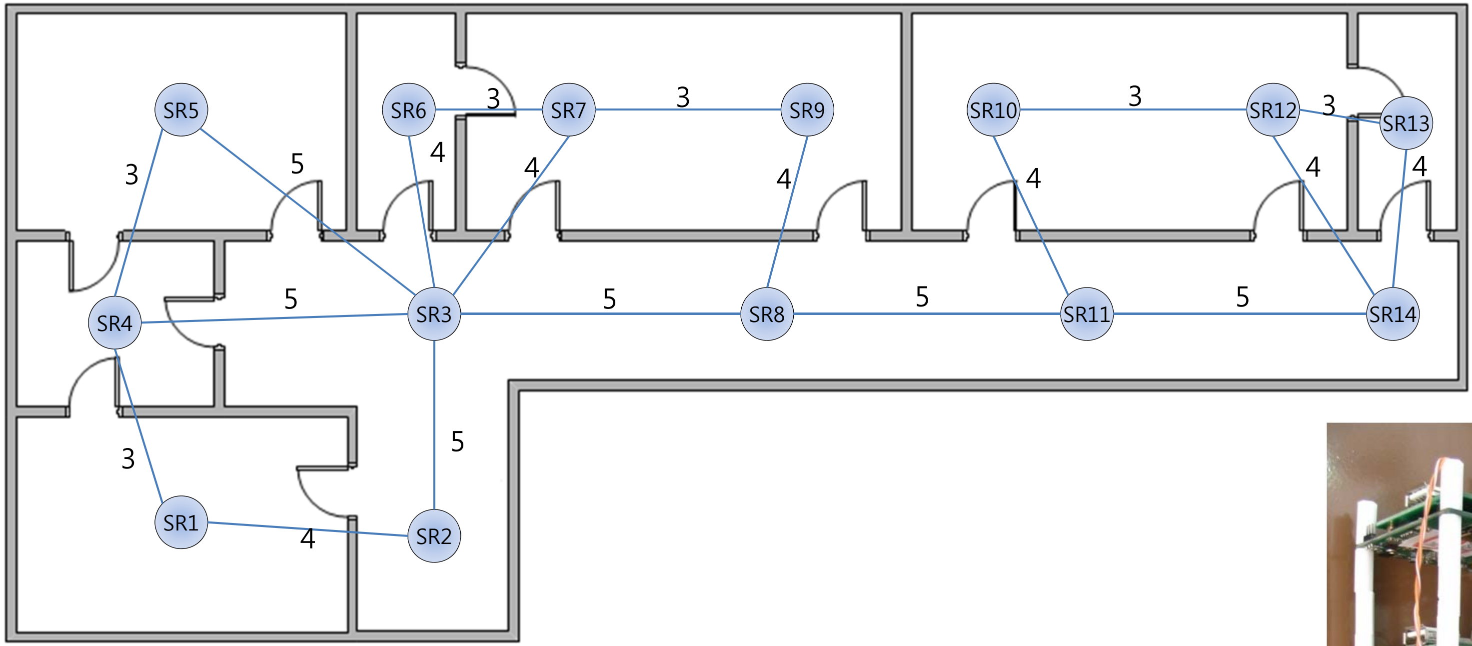
# Scalability Evaluation : Simulation (2/4)

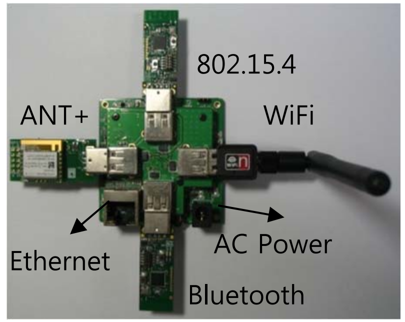
❖ **Comparison between proposed self-organizing and centralized architecture**




Center of Self-Organizing Software-Platform

# Scalability Evaluation : Real Test-bed (1/2)



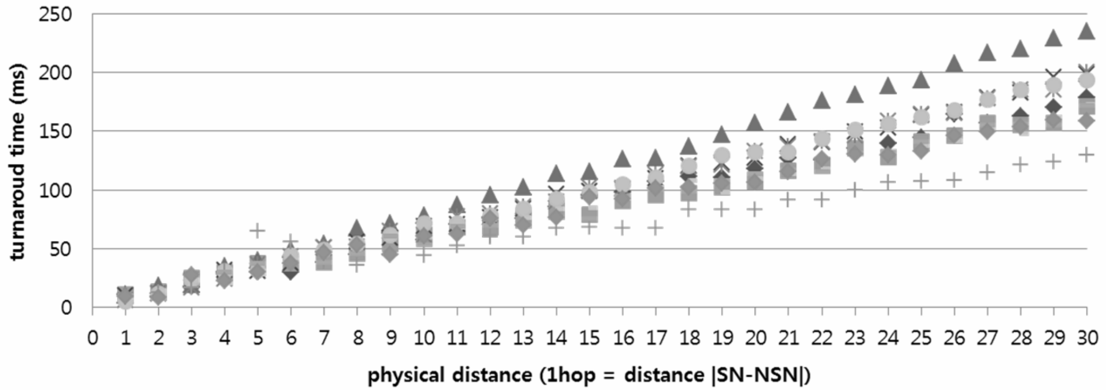




<SoSp-Router Board>

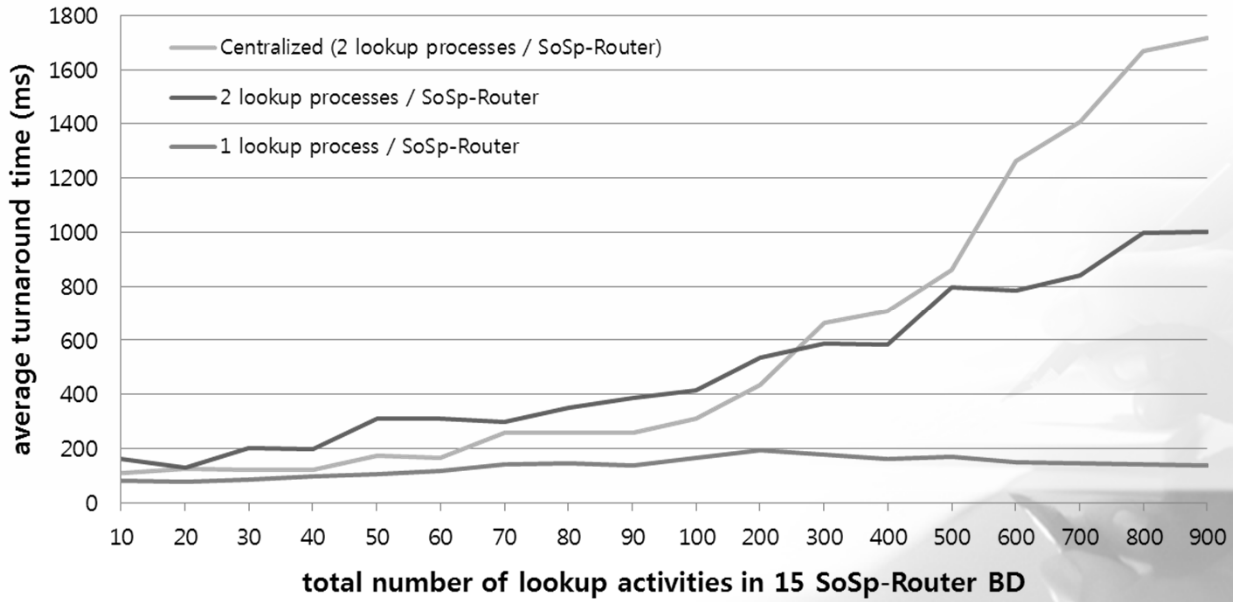
- ❖ **Test-bed Environment**
  - – 14 SoSp-Router Boards comprised of above floor plan
  - – One Ethernet Hub
  - – PC with intel T8300 2.4GHz dual core
    - 1 server for comparison with centralized architecture
    - Mobile and resource threads emulate the behaviors of real mobile node (such as LIDx&AMD, location movement, service processing, and so on)

# Scalability Evaluation : Real Test-bed (2/2)

❖ **Response Time: turnaround time by distance in one SoSp-Router**




❖ **Comparison between proposed self-organizing and centralized architecture**




Center of Self-Organizing Software-Platform

# Performance of Mobile Service Rebinding (1/2)

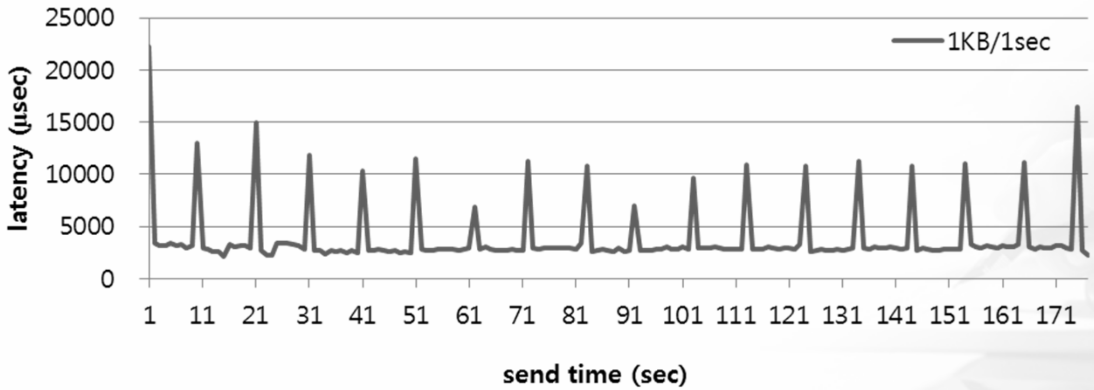
❖ **Environment for Rebinding Evaluation**

– Mobile and resource threads in simulation computer emulate the behaviors of real mobile node (such as LIDx&AMD, location movement, service processing, and so on)

– The emulated threads of mobile node continuously sends the control command of specific size, and the resource threads also send the status data for responding.

– The mobile threads periodically move per 10 msec.

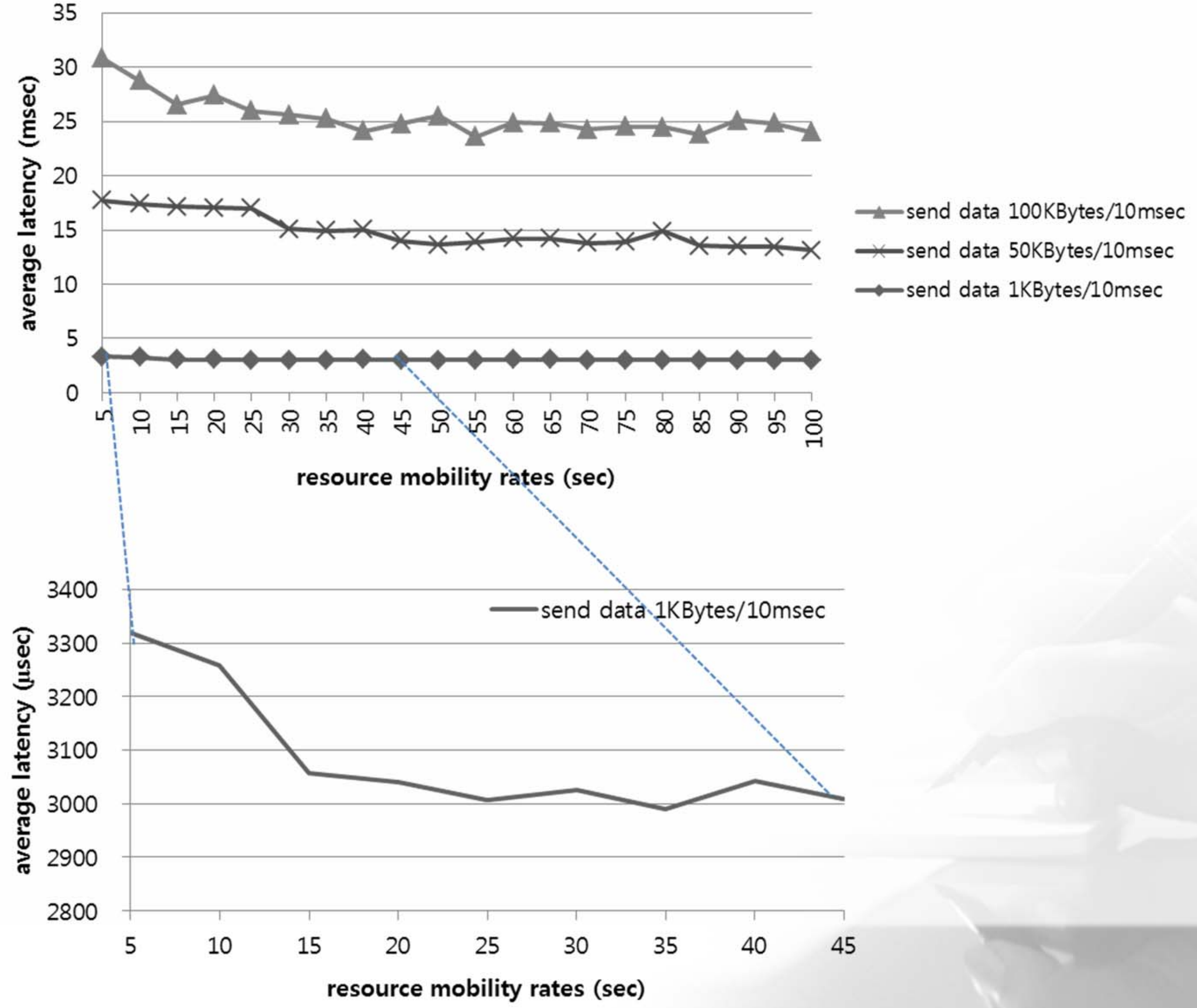
❖ **Total latency by mobility 10 sec rates for 180 sec.**




Center of Self-Organizing Software-Platform

# Performance of Mobile Service Rebinding (2/2)

- Average latency by periodic mobility rates from 5 to 100 sec for a total of 180 sec.



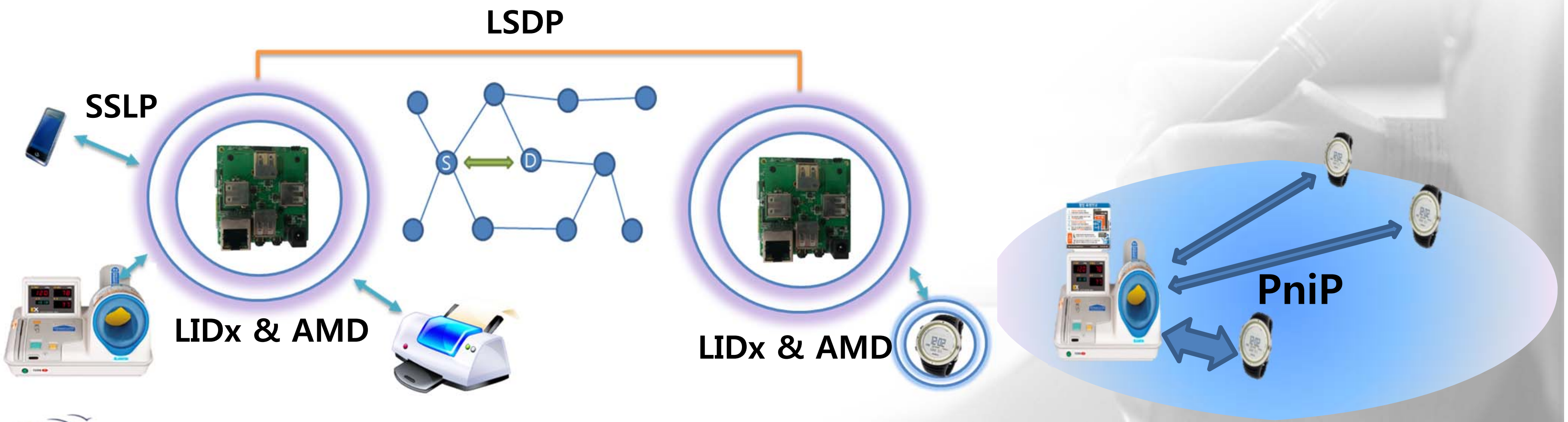
# Part IV:
# ePost-It : Locaized IoT Messaging Middleware

# Suggested Protocols in LIoT Infra

- SoSp-Router ↔ Smart Mobile Device(Smart phone or pad)
  - **SSLP** (**S**martDevice to **S**oSp-Router Service **L**ookup **P**rotocol)
- SoSp-Router ↔ SoSp-Router
  - **LSDP** (**L**ocation-based **S**ervice **D**iscovery **P**rotocol)
- SoSp-Router ↔ SoSp-Client
  - **LIDx&AMD** (**L**ocation **ID** e**x**change and **A**synchronous **M**essage **D**elivery)

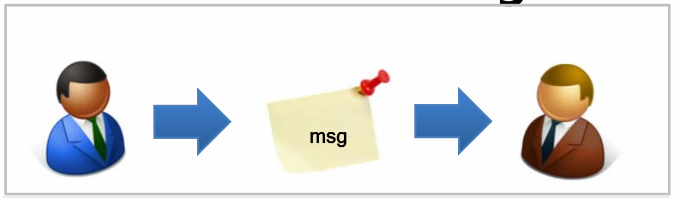
ePost-it (middleware in SLiM)

- SoSp-Client(Resource) ↔ SoSp-Client(Mobile ID device)
  - **PniP** (**P**roximity based **N**eighbor **I**dentification **P**rotocol)

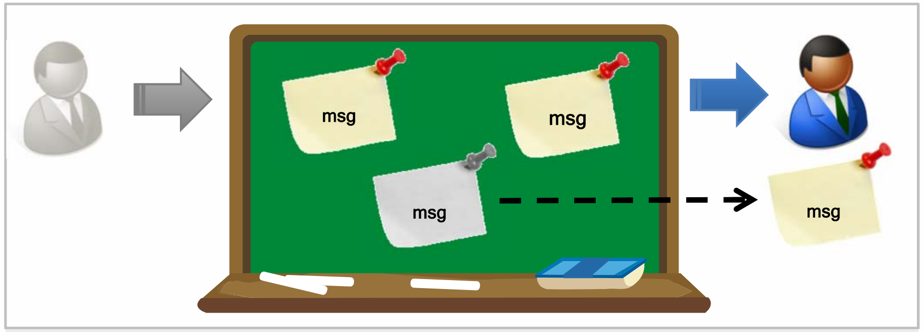


# Concept of ePost-it

- Post-it concept of real-world is applied to implement delay tolerance and location based message delivery
- Message Delivery Middleware running in SLiM hub
- Three feature of Message Delivery :




① Deliver a message directly




② A message is asynchronously delivered through common space like board.


③ If a message is posted in board, it is regionally broadcasted.

# Requirements of ePost-it

❖ **Common message format**
- To communicate between heterogeneous devices
- Including text, audio, streaming…

❖ **Push/Pull service**
- To achieve low power consumption

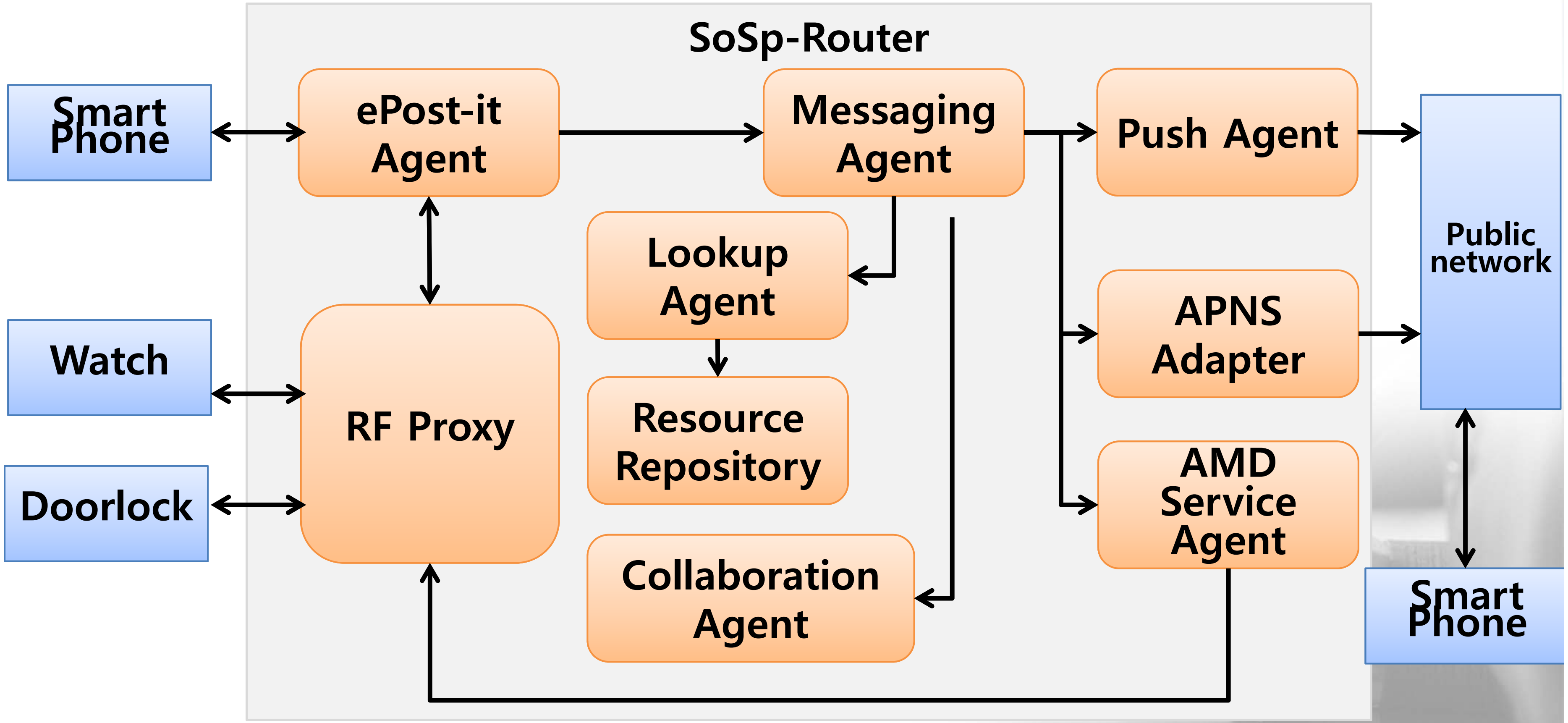
❖ **Public message board**
- Asynchronous delivery and Broadcasting

❖ **Directory service**
- To manage a group of users and routers(locations)

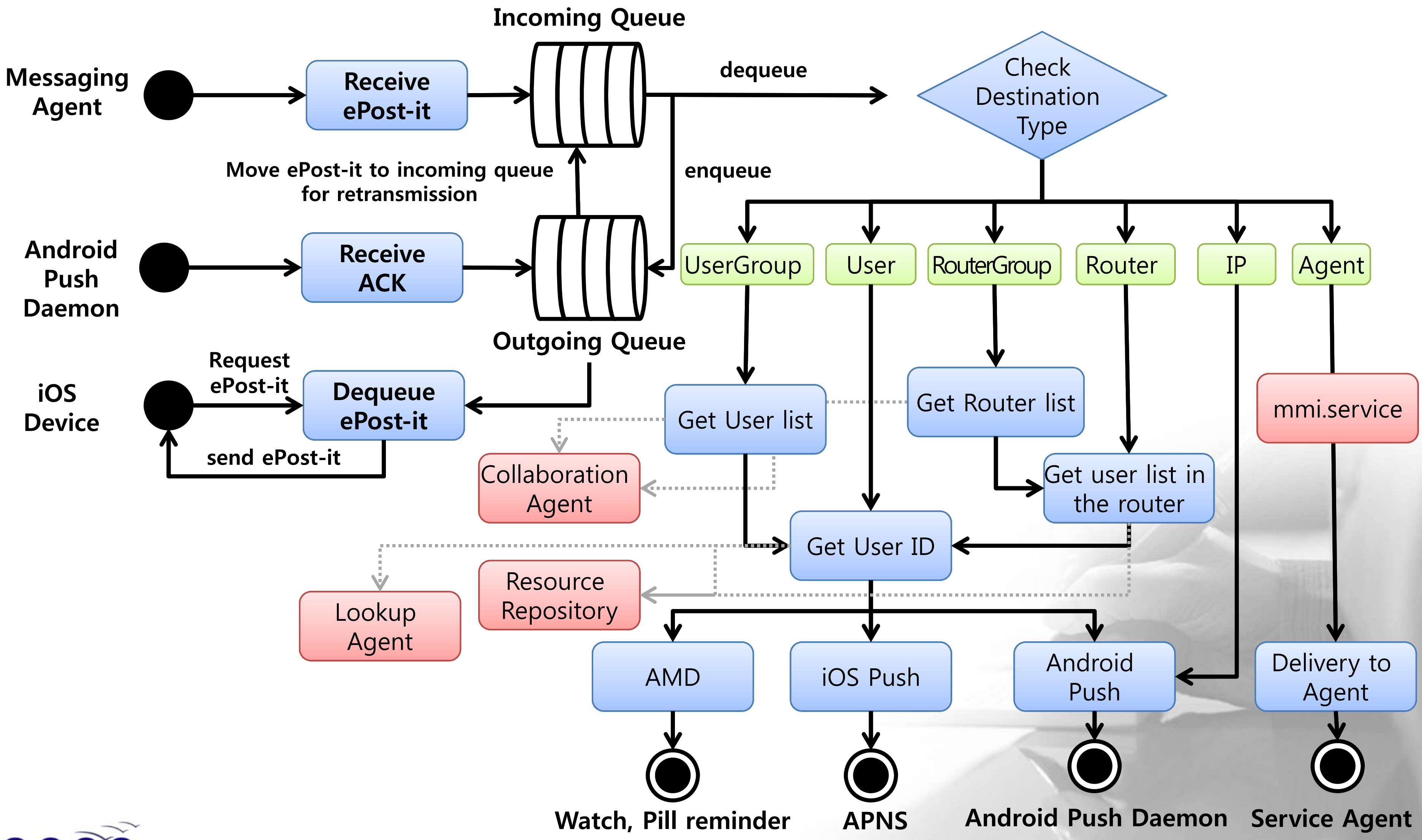
# ePost-it Message Format

| Elements | Attributes | Comments |
|---|---|---|
| Header | VERSION | Format version |
| | ID | Unique ID of ePost-it |
| | NAME | Receiver App/Agent Name |
| | SEND_TIME | Transmitted time |
| | PRIORITY | Priority of Message |
| | CALLBACK | Callback request flag |
| | STREAM | Stream open request flag |
| | SEQUENCE | Sequence # |
| | TTL | Time To Live |
| | SUBJECT | Subject of Message to describe briefly |
| Source | ADDRESS_TYPE | Address type (ex, user, router, usergroup, router group) |
| | DEVICE_TYPE | Device type(ex, phone, watch, resource device ...) |
| | ADDRESS | Network address (ex, IP, BT address, ZigBee address ...) |
| Destination | ADDRESS_TYPE | Address type (ex, user, router, usergroup, router group) |
| | DEVICE_TYPE | Device type(ex, phone, watch, resource device ...) |
| | ADDRESS | Network address (ex, IP, BT address, ZigBee address ...) |
| Contents | TYPE | Contents type(ex, text, audio,..) |
| | BODY | Contents body |

# Agent Structure in SoSp-Router



# Data Flow to Push ePost-it




CSOS Center of Self-Organizing Software-Platform

# PAAR Watch Platform

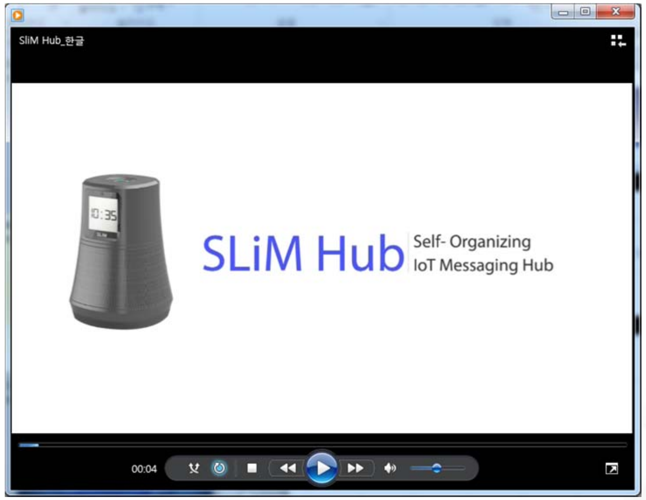
- ❖ **Indoor Localization**
- ❖ **Direct Communication among external devices**
- ❖ **Exercise Activity and gesture recognition**
- ❖ **Hardware plug-in**
  - – Pulse Oximeter, Glucose meter
- ❖ **Smart Phone tethering**
  - – iPhone with own messaging
- ❖ **Personal activity reminding and assistance**


Center of Self-Organizing Software-Platform

# A Demonstrative Video of our platform

❖ PAAR Watch


❖ SLiM Hub




CSOS Center of Self-Organizing Software-Platform

# Conclusion

- ❖ **Propose a new Indoor IoT Software Architecture : SoSp**
  - Scalability, decentralization, fairness, robustness
  - No need to maintain a global knowledge such as building map.
  - No need to set any configurations for using services.
- ❖ **Self-Organizing Distributed Middleware in SoSp Architecture**
  - Each SoSp-Router only knows the information of 1-hop neighboring SoSp-Routers.
  - Support the method of Dynamic Space Expansion & Reduction (DSER) with self-organization property.


CSOS
Center of Self-Organizing Software-Platform


Thank you.