Location-based Services in Ubiquitous Computing Environments

National Institute of Informatics

Ichiro Satoh

Email: ichiro@nii.ac.jp

Outline

1. Motivation
2. Approach
3. Design and Implementation
4. Applications
5. Conclusion

Location-based Services

Context-aware services:

- location-based information services
- personalized information services

For example

- Location-dependent advertising
- Tourist navigation information
- Regional/local personalized portals

Existing Approaches

Context-aware services on portable computing devices:

- **User moves with computing devices equipped with poitioning systems.**
  e.g. Cooltown (HP) and NEXUS (Stuttgart University)

Context-aware services in smart spaces:

- **A space is equipped with tracking systems and services can be provided at computers near the locations of users.**
  e.g. Sentient Computing (Cambridge University) and EasyLiving (Microsoft Research)

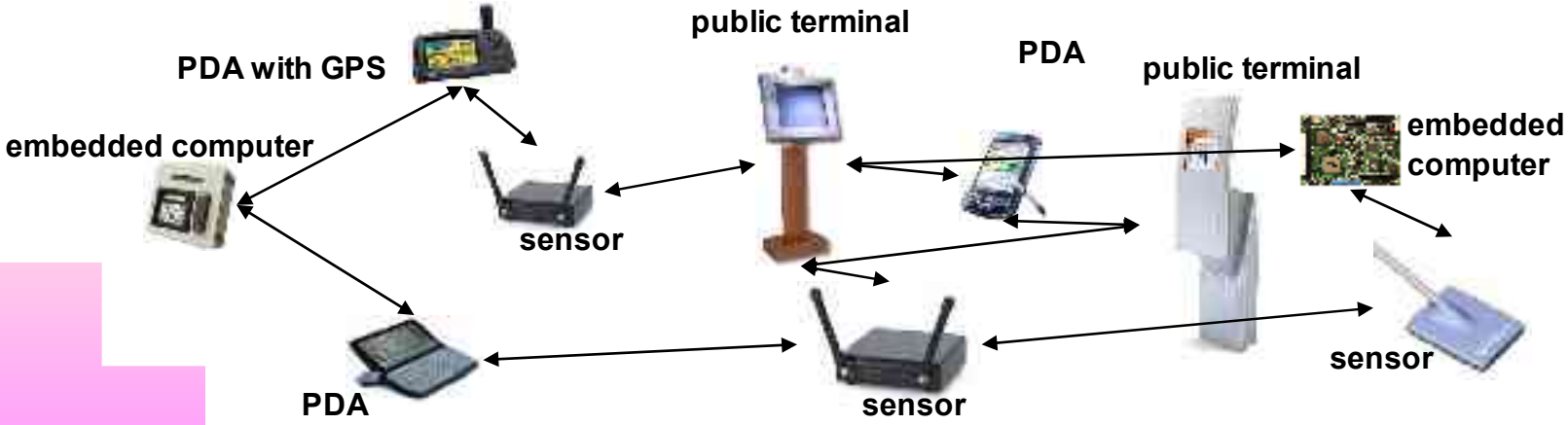
The two approaches aim at the same goals but are posed as polar opposites.

A unified framework for integrating the two approaches is needed.

Problems

Several problems in existing location-based services:

- A ubiquitous computing environment consists of various computing devices, e.g. mobile and embedded computers.
- Ubiquitous computing devices have limited resources so that they can support only their initial services.
- Existing services are designed dependently on particular vendors' sensors.

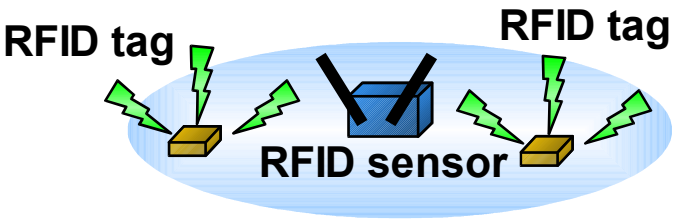


Positioning Sensors and Tracking Sensors

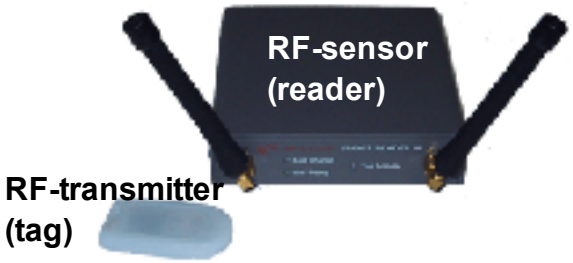
Two types of location-sensing systems:

- Tracking sensors

  e.g., RFID tag sensors and computer vision systems



RF-based Location Sensor



- Positioning sensors

  e.g., GPS and wireless LAN

The framework supports various sensors, including RFID tags, and hides differences between sensing systems.

Requirements

- **Adaptability**

  Services can adapt not only their contents but also their locations according to changes in the positions of people and physical objects.

- **Modularity**

  Services must be modular and reusable independently of the underlying systems as much as possible.

- **Application-neutral**

  Various and multiple services can be supported.

- **Generality**

  A unified approach for integrating the following settings:

  - stationary computers and mobile computers
  - stationary sensors and mobile sensors
  - tracking sensors and positioning sensors

Approach

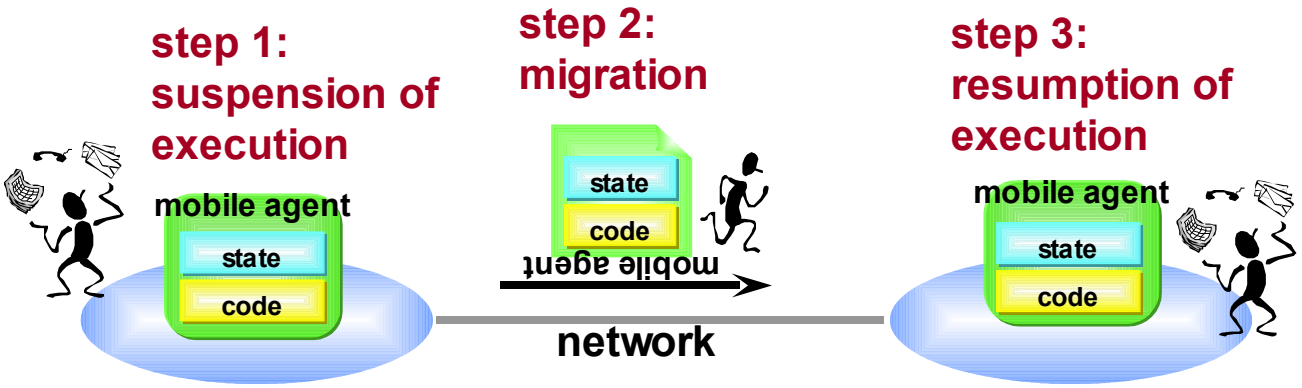
A framework for building and managing location-based and personalized services in ubiquitous computing environments.

- It supports services for annotating and supporting people, things, and places.
- Each service is implemented as a mobile agent.
- Each service can be spatially bound to people, objects, and places and dynamically deployed at proper computers.
- The framework seamlessly supports stationary and mobile computing devices, and tracking and positioning sensors,

The framework enables most existing mobile agent platforms, including non Java-based platforms, to migrate their mobile agents according to the locations of physical entities or places.

Dynamically Deployable Services

Services are implemented within mobile agents and can migrate to computing devices according to the locations of the physical entities that they are bound to.



- Mobile agent-based service providers are self-contained and mobile software units.
- They can be deployed at and continue its execution at the destination only while they are needed.
- They can be locally executed without network latency and conserve the limited resources of computing devices.

Spatial Links between Physical Entities or Places and Services

- All application-specific services cannot monitor the underlying systems, including location-sensors.
- It is complex and difficult for services to adapt their contents and locations according to results measured by the sensors.

The framework introduce a simple policy for the deployment of service provider mobile agents.

It migrates services bound to physical entities or places to computers near the current location of the physical entities or places.

To interact with users locally, services do not have to handle their own positions, because they can be automatically deployed at devices near the users by the framework.

Architecture

The framework can monitor the location of each tag and instruct the mobile agents that is bound to the tag to migrate to proper computers in the region that contains the tag.

Location Information Server:

- A location-aware directory service for mobile agents according to the locations of entities in physical world

Agent Host:

- A computer for executing and migrating mobile agents.

Service Provider Agent (Mobile Agent):

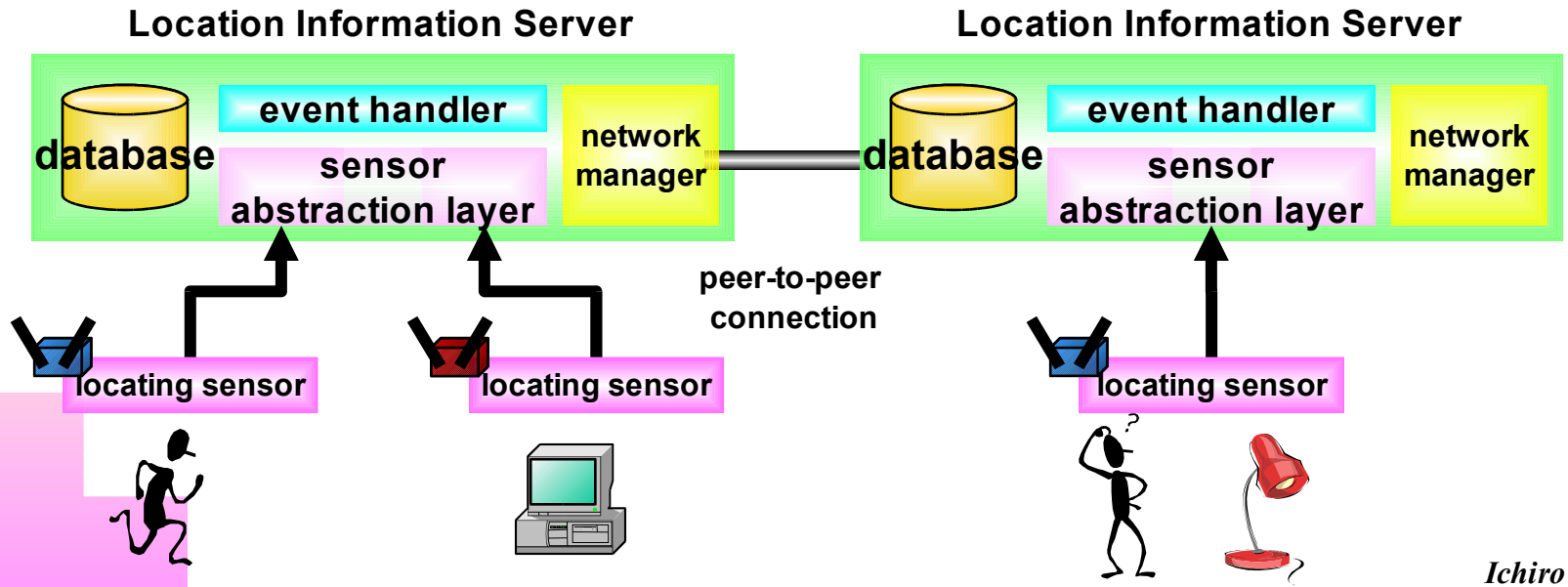
- A service-provider and virtual counterpart of physical entity or place.

Location Information Server

Each LIS can run on a stationary or mobile computer and be coordinated with other LISs in a peer-to-peer manner.

- Management of more than one locating sensing system

  e.g. tracking sensors and positioning sensors
- Discovery of agent hosts and service provider agents
- Context-aware deployment of mobile agents.

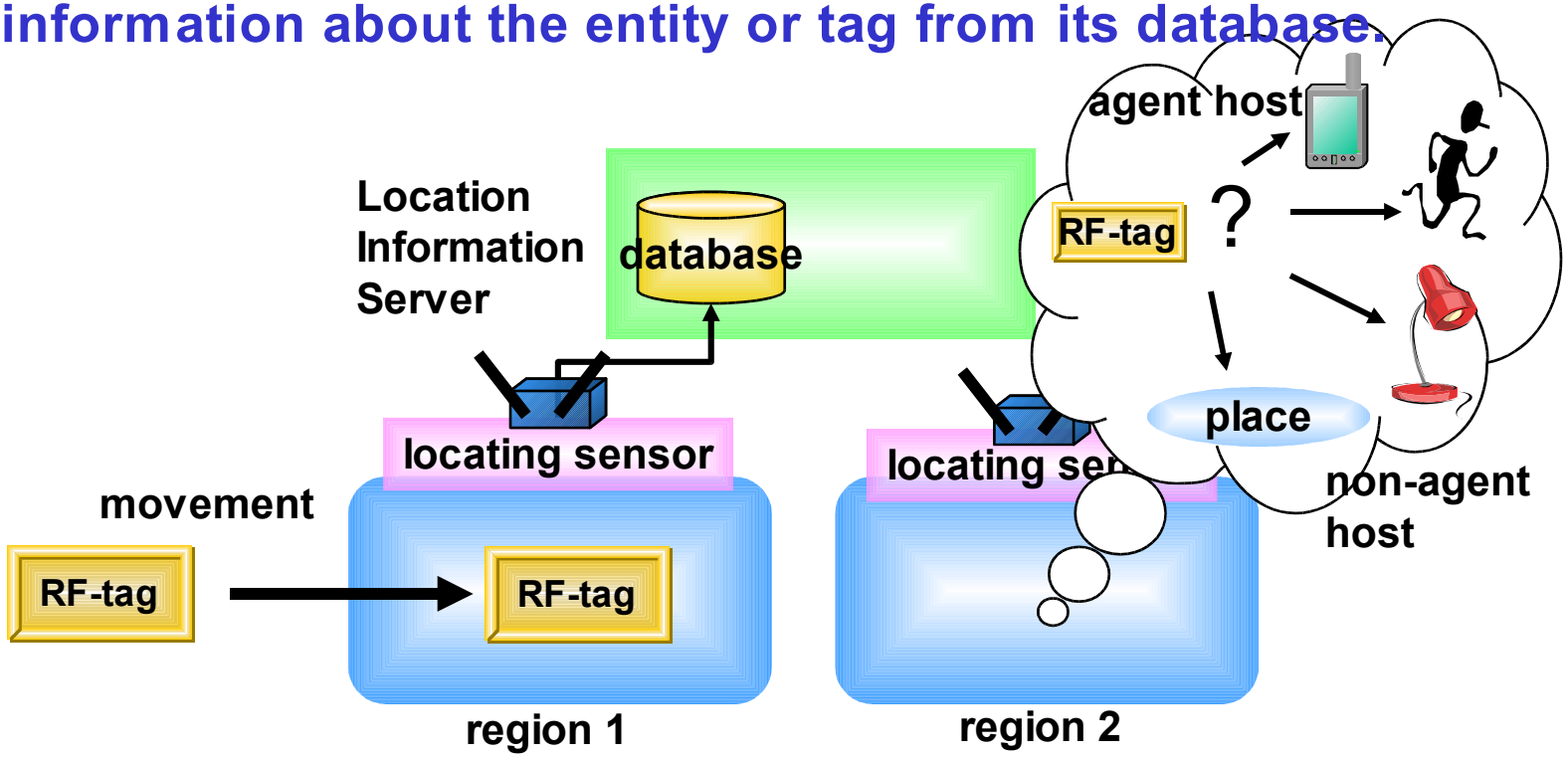


Ichiro Satoh

Tag Resolution

Each LIS has a database for information about agent hosts and service provider agents.

When an LIS detects a new entity or tag, the LIS searches information about the entity or tag from its database.



Each tag may be bound to an agent host or non-agent host.

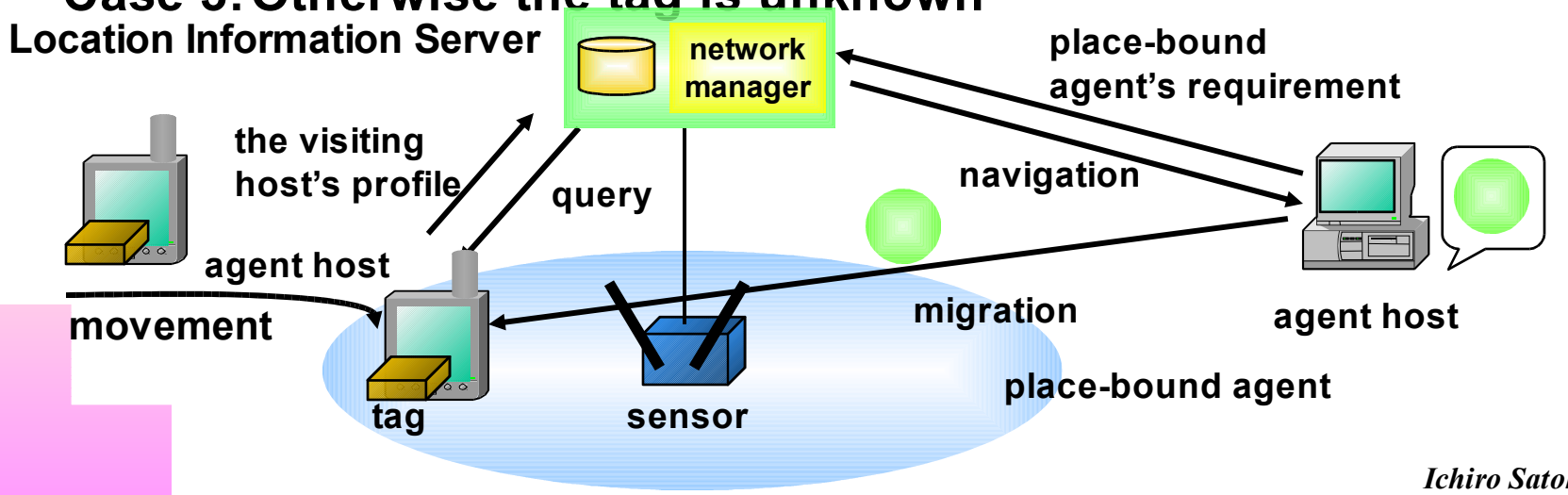
Service Deployment

When an LIS can multicast a query about a newly visiting RFID tag (i.e. physical entity or place), other LISs process

Case 1: If the tag is attached with an entity or place, other LISs search agents bound to the tag and then inform the agents about candidate agent hosts.

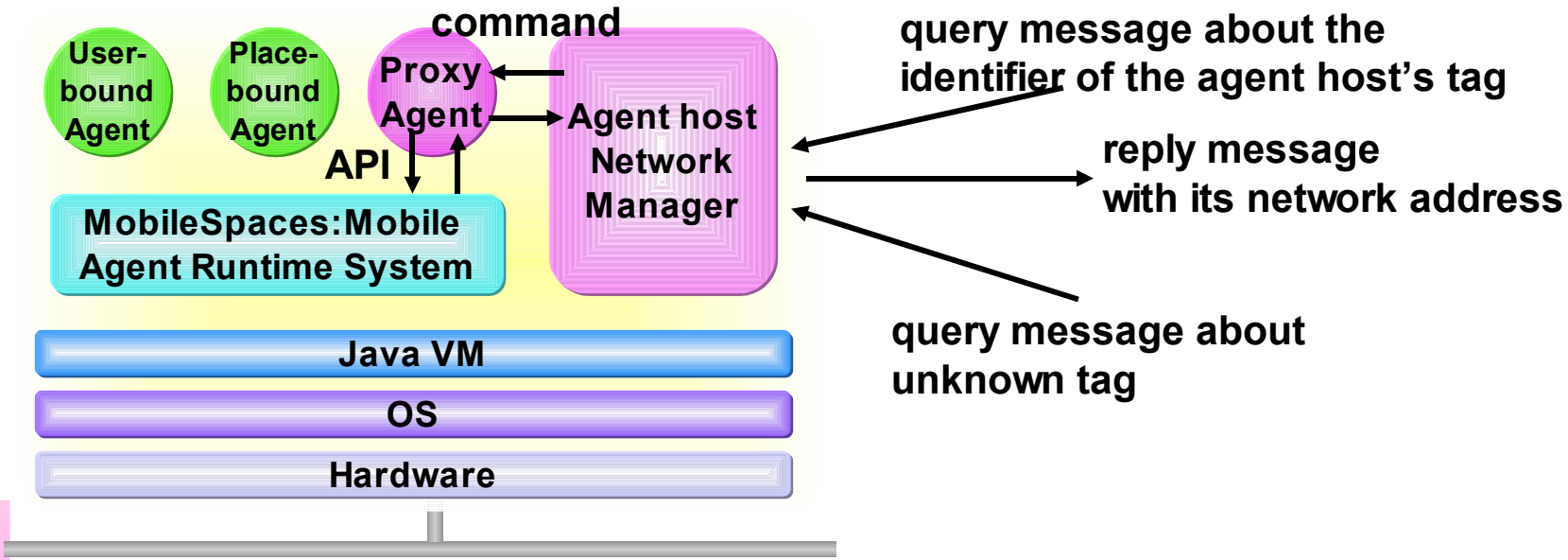
Case 2: If the tag is attached with an agent host, other LISs or the agent host returns its profile to the LIS.

Case 3: Otherwise the tag is unknown



Agent Host

- Each agent host is a stationary or mobile computing device attached with an RFID tag.
- It provides a mobile agent platform for executing and migrating agents to other agent hosts through a TCP connection.



Service Provider Agent

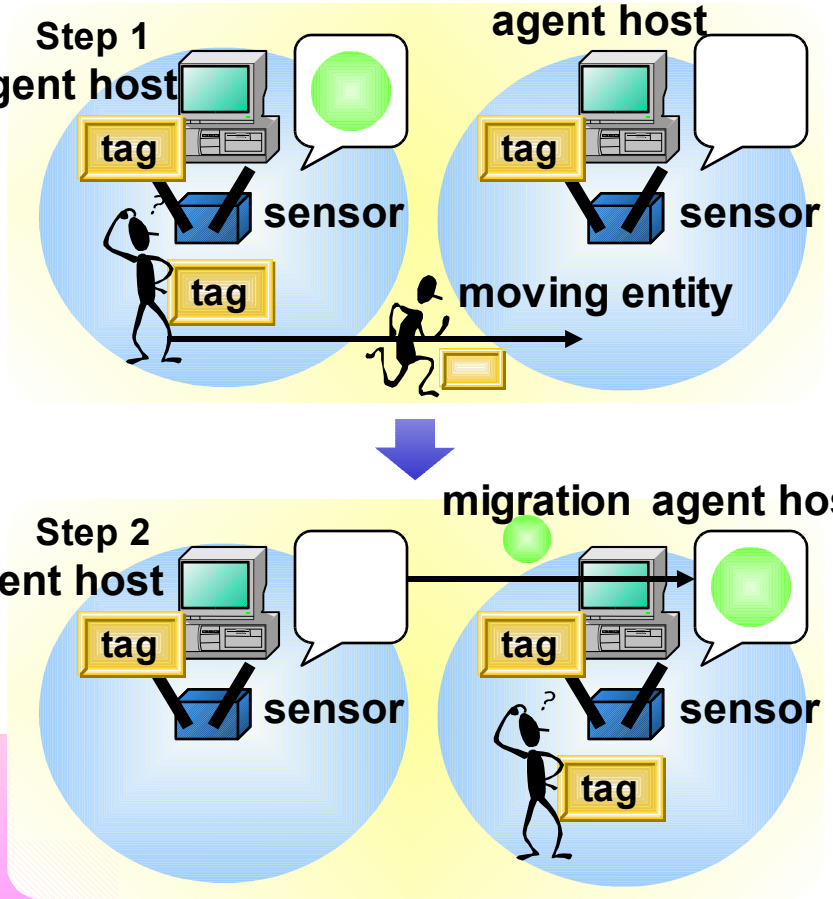
- Since application-specific services are encapsulated within agents, the framework is independent of any applications.
- Each mobile agent can registry the identifier of its tag and the host requirements written in CC/PP.

```
class TaggedAgent extends Agent implements Serializable {
  void go(URL url) throws NoSuchHostException { ... }
  void duplicate() throws IllegalAccessException { ... }
  void destroy() { ... }
  void setTagIdentifier(TagIdentifier tid) { ... }
  void setAgentProfile(AgentProfile apf) { ... }
  URL getCurrentHost() { ... }
  boolean isConformableHost(HostProfile hfs) { ... }
  ....
}
```

Please refer to the Appendix of the paper for detail information on programming interfaces for service provider agents.

Location-Aware Services (the first linkage)

A moving entity is attached with an RFID tag and agent hosts and RFID sensors are stationary in places.



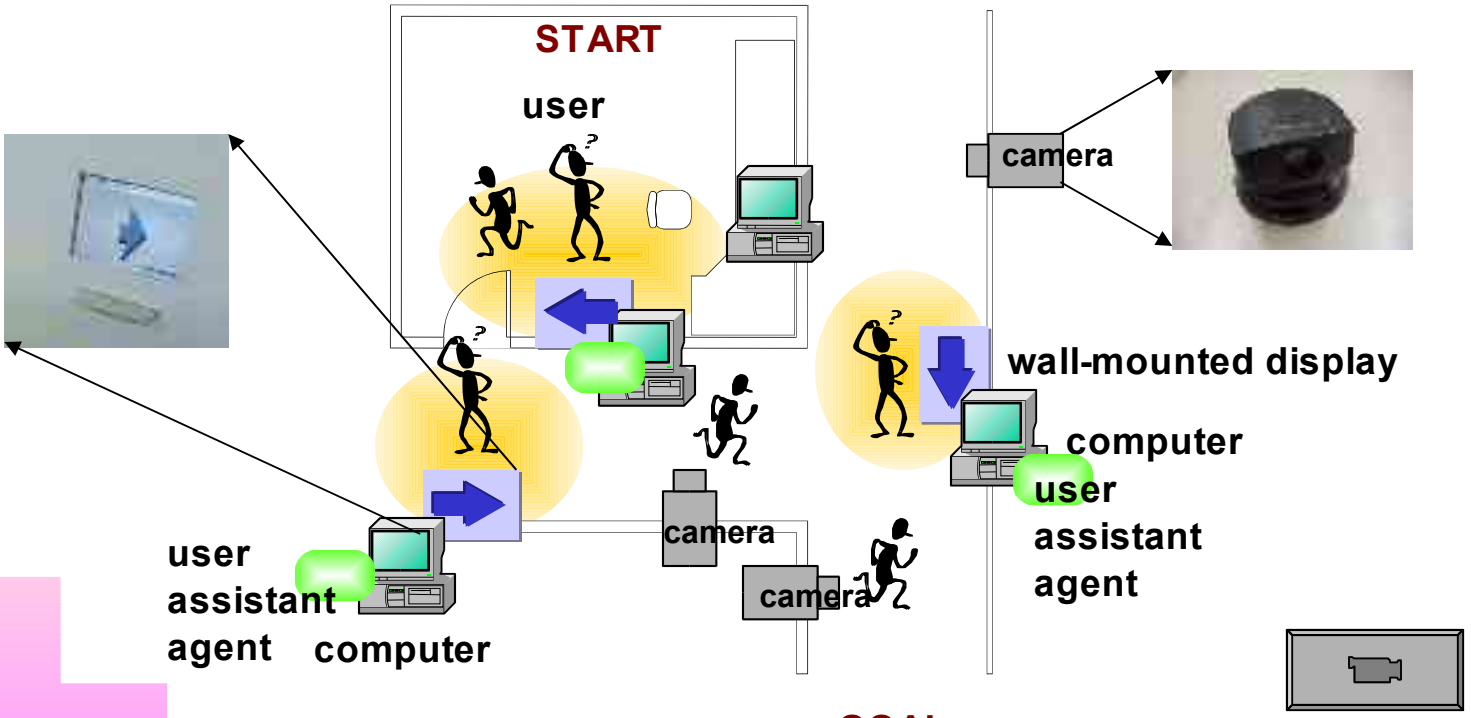
Step 1:
An RFID sensor in a place detects the tag of a moving entity.

Step 2:
An agent bound to the entity migrates to an agent host in the same place.

Follow-Me Services

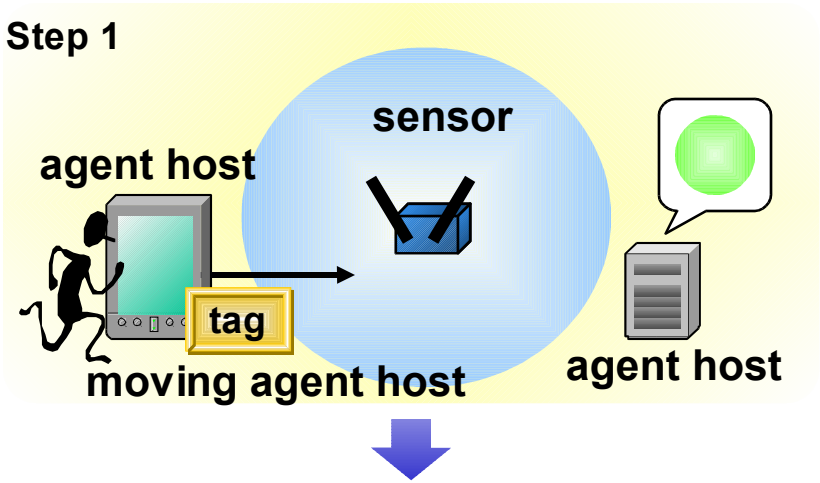
Navigation system for patient in a hospital:

- User navigating agents are bound to particular partients.
- Agents are automatically migrated to the nearest computer from the current positions of users.



Location-Aware Services (the second linkage)

A moving entity carries an agent host and a place is equipped with a stationary RFID sensor.



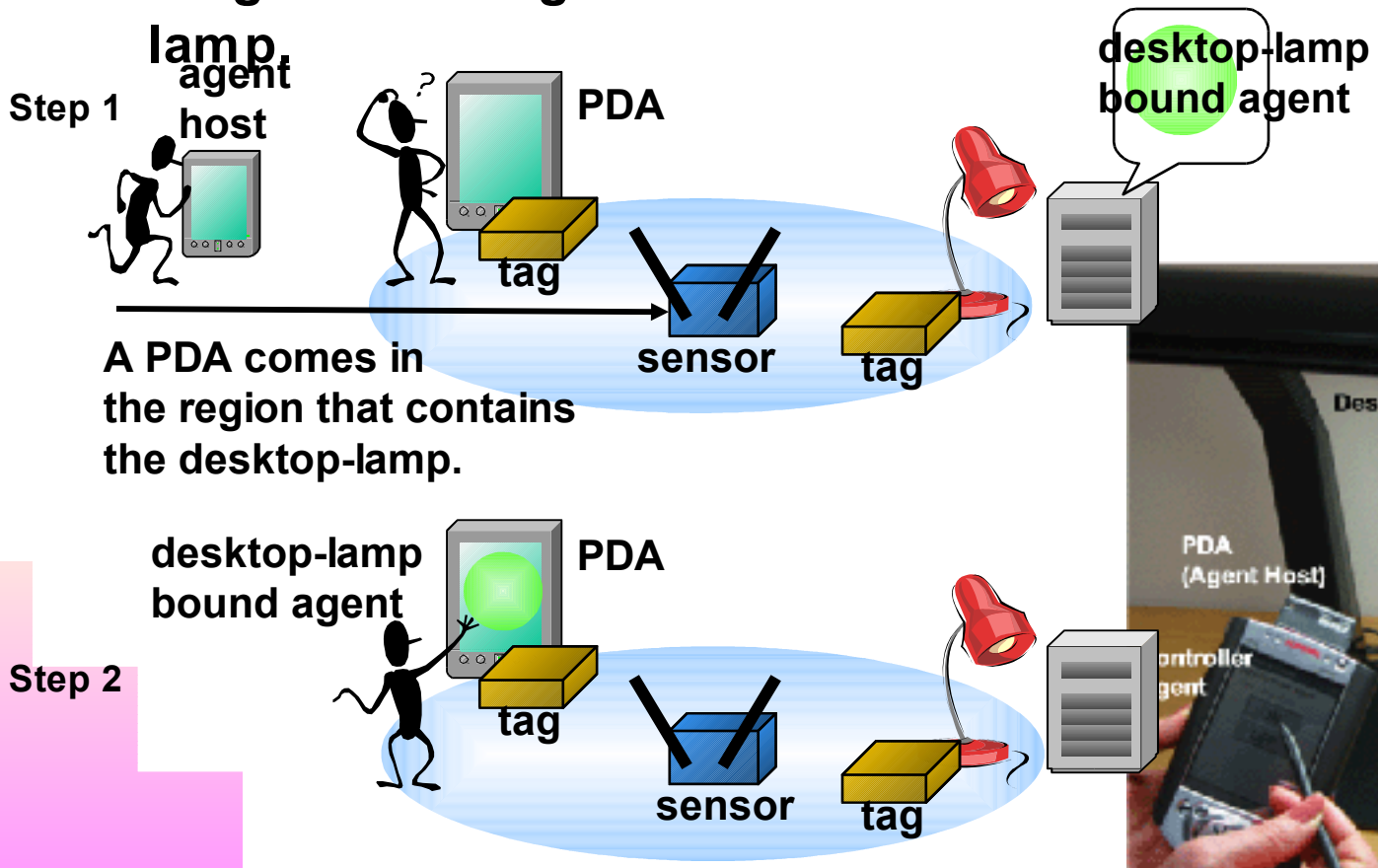
Step 1:
An RFID sensor in a place detects the tag of an agent host.

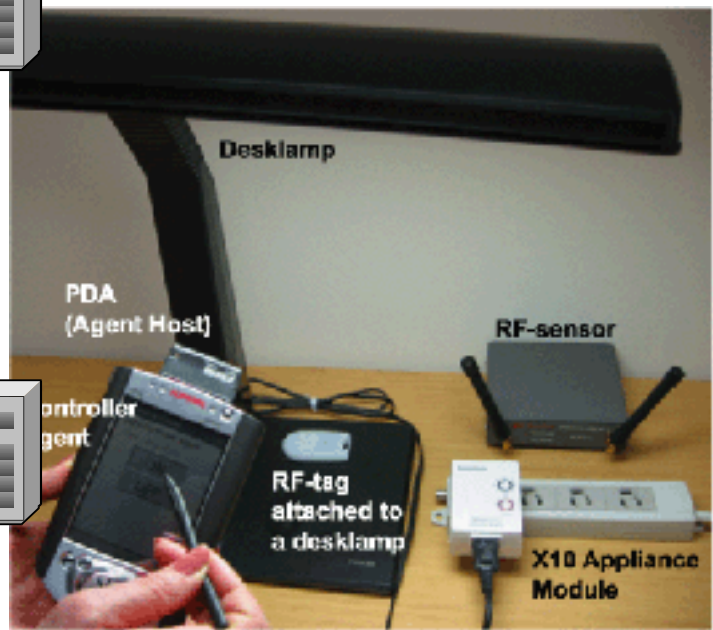
Step 2:
An agent bound to the place migrate to the visiting agent host.

Location-aware Universal Controller for Appliances

A desktop-lamp bound agent communicates with X10-based servers to switch the desktop-lamp on or off.

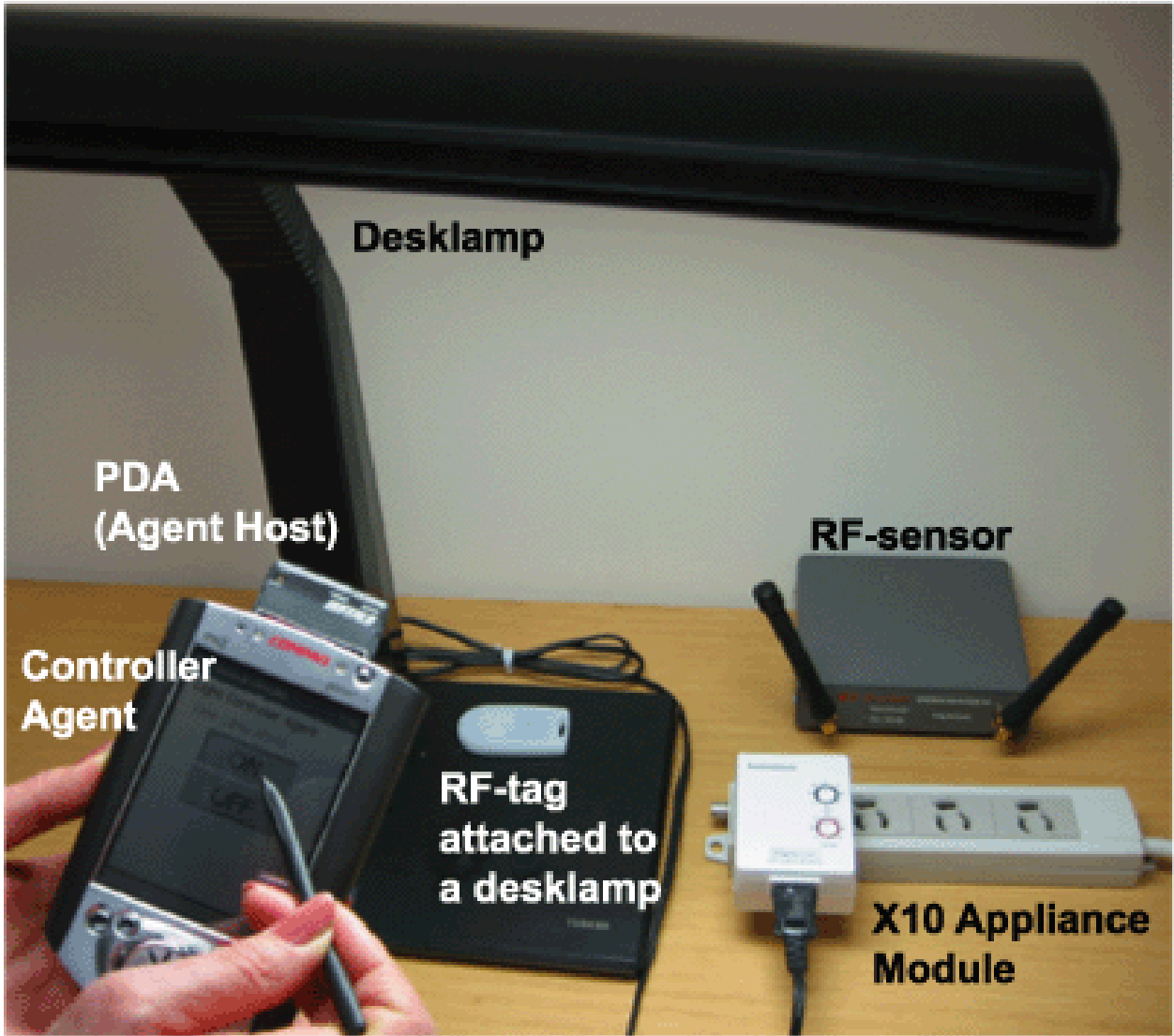
The agent can migrate to a PDA near from the desktop-lamp.





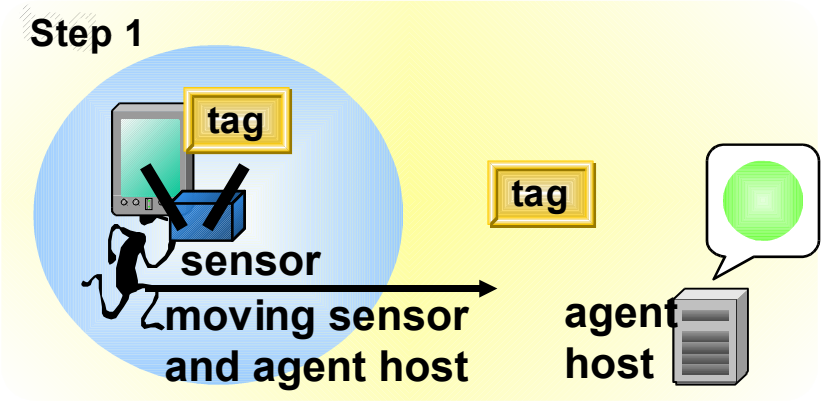
Location-aware Universal Controller for Appliances

When a PDA enters in the area that contains a desktop-lamp, the agent bound to the lamp migrates to the PDA.

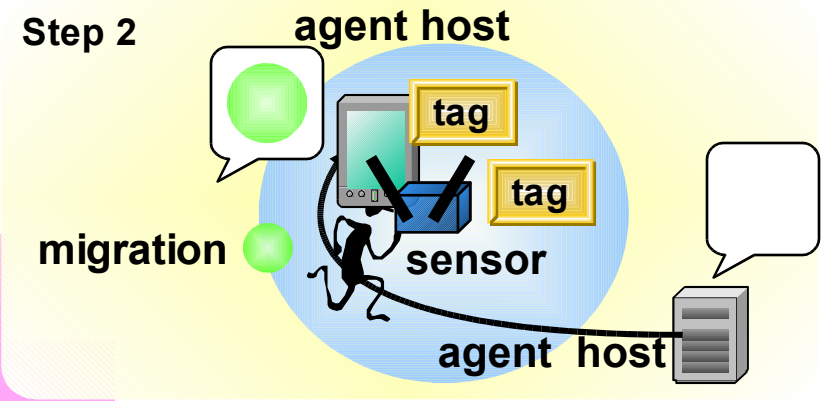


Location-Aware Services (the third linkage)

A moving entity carries an agent host and an RFID-based positioning sensor and a place is attached with an RFID



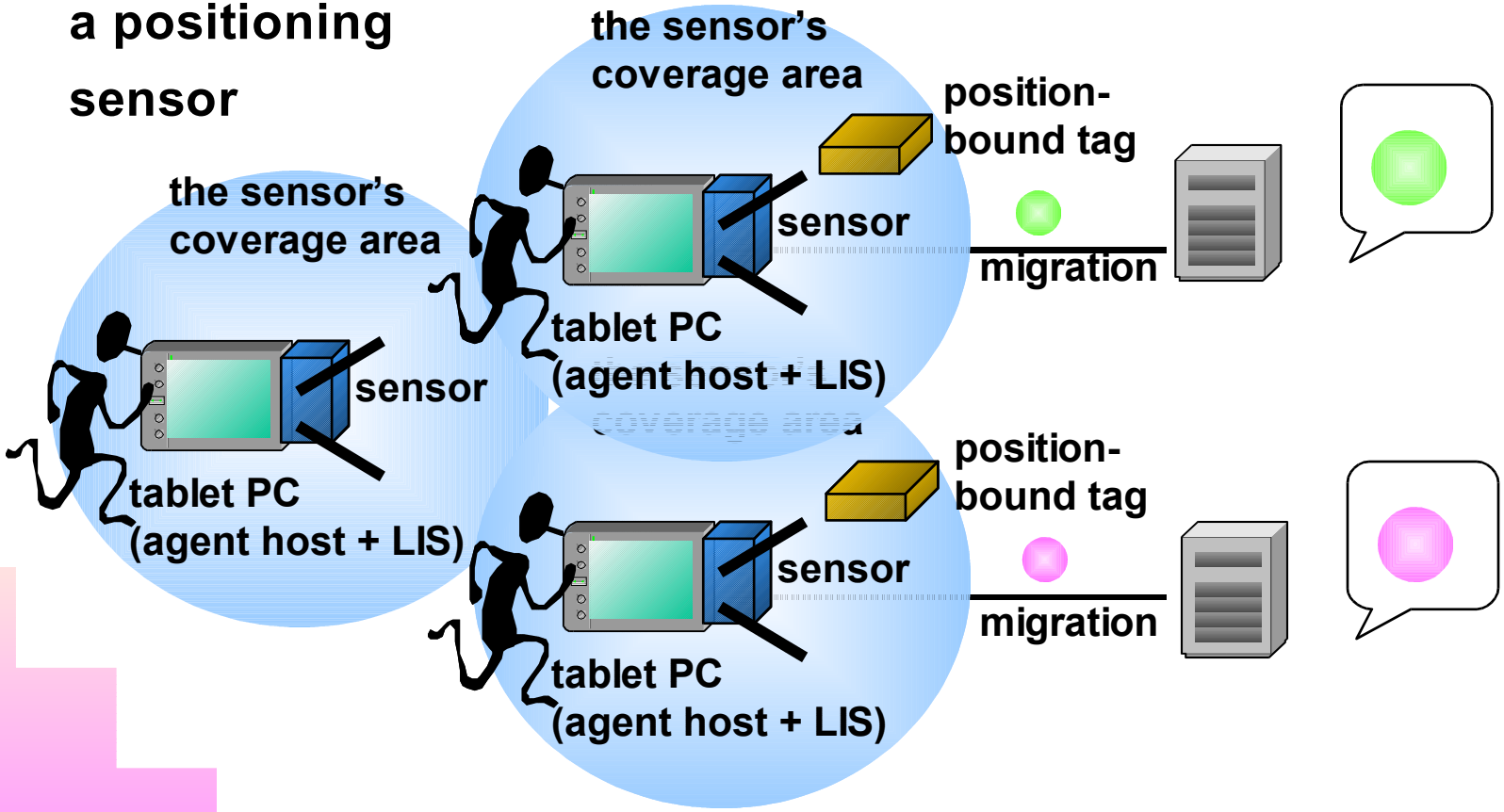
Step 1:
A moving sensor detects the tag of a place.



Step 2:
An agent bound to the place migrate to the visiting agent host.

User Navigation System

A location-based navigation on a Tablet-PC equipped with a positioning sensor



Why the framework is not based on web-service

Mobile agent-based services have several advantages in comparison with web-services.

Ubiquitous computing devices are often connected through wireless networks.

- Mobile agents can be executed even when their source hosts are disconnected from a network.
- Web-services assume that networks are connected and reliable anytime.

Ubiquitous computing services are often managed without any servers.

- Mobile agent runtime systems can exchange agents with other systems in a peer-to-peer manner.
- Web-services are essentially managed and operated by servers.

Ubiquitous computing environments must support proactive services rather than request-and-response-based services.

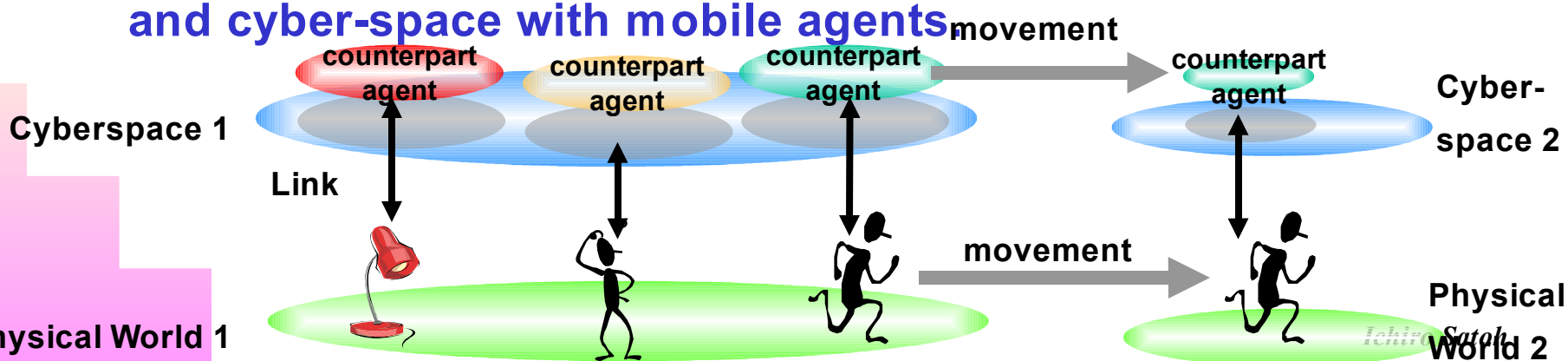
- Mobile agents are autonomous and programmable entities.
- Web-services are based on request-and-response paradigm

Conclusion

A framework for building and managing location-aware services.

- It binds service provider agents with physical entities and places.
- It dynamically deploys agents to computers whose capabilities can satisfy the requirements of the agents and which are near the location of the entities and places.

The framework can spatially bridge between physical world and cyber-space with mobile agents.



Future Work

Several new results will appear in upcoming papers:

- Other linkages between physical entity and agents[IEEE MDM'04]
- Dynamic aggregation of computing devices [IEEE PerCom'04]
- Specification for migration policies of services [IEEE ICDCS'04]
- Rapid development and testing for network/location-aware software [IEEE Trans. Software Engineering vol. 29, no. 12, 2003]

Future issues:

- Enhancement of privacy and security mechanism
- Inter-service provider agents communication
- Adaptive mechanisms for resource-dependent services
- More applications