# From Grid to Pervasive Computing

***Where is the breakthrough of next IT boom?***

**Xian-He Sun**

Scalable Computing Software Laboratory

Department of Computer Science

Illinois Institute of Technology

Sun@iit.edu

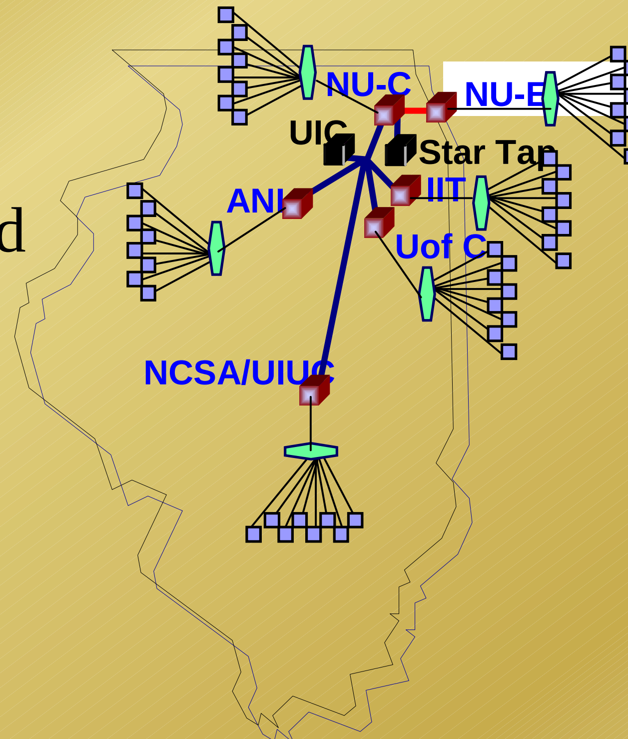
April 7, 2004

**IEEE Lecture**

# Scalable Computing Software (SCS) Lab.

Parallel Computers at SCS

Distributed
Optical Testbed
(Grid)





Pervasive Computing
Environments at SCS

# Outline

- Introduction
- Pervasive Computing
  - Context-awareness and mobility
  - Case study: Context-aware infrastructure
- Grid Computing
  - Service and open service architecture
  - Case study: Resource management
- Conclusion

# Introduction

## The Three Waves of Computing Revolutions

### First Wave

- Hardware, silicon chip
- Lasted 10-15 years into the 90s
- Silicon Valley
- Open system versus closed system
- "Coopetition" versus competition

### Second Wave

•Software, video game

•Begin mid-80s into the 90s

•Everywhere, Microsoft

•Distribution versus technology

# The Third Wave of Computing Revolutions

- Network, communication, and interconnectivity
- Begin in the late 90s until now
- Machine/machine, software/software, people/people
- Anytime, anywhere, WWW
- The communications landscape is shifting

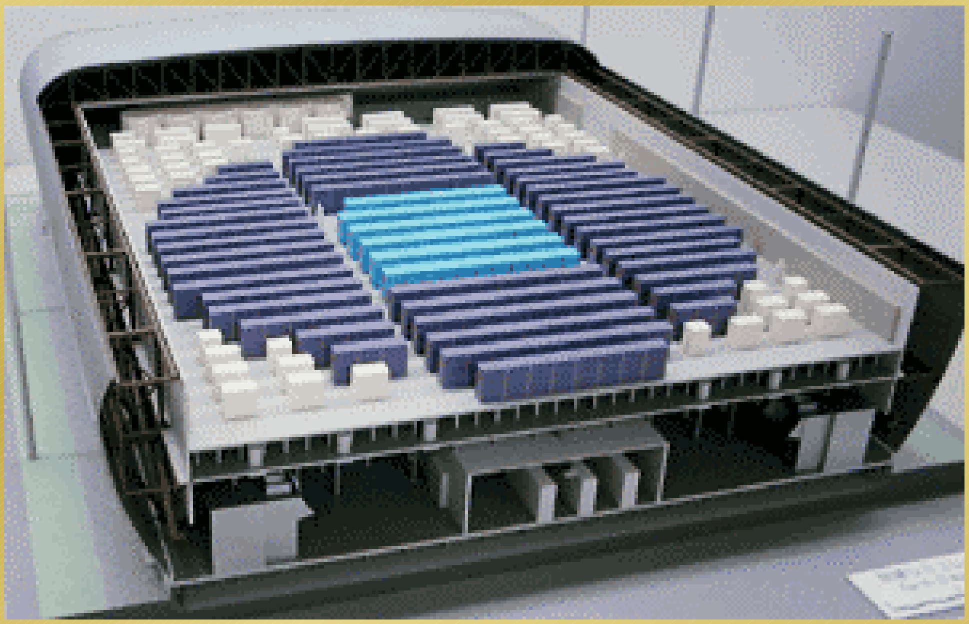
What is the next?

- – Grid computing? Pervasive computing? Ubiquitous computing?

# Evolution of Computing

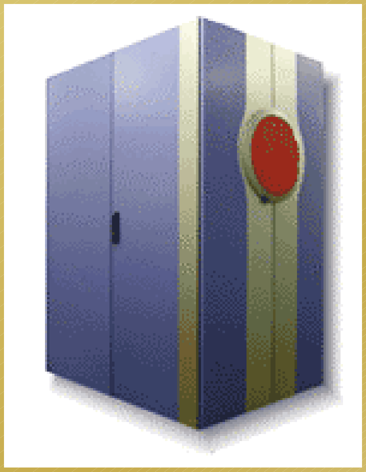
Bigger becomes even bigger

Smaller becomes ever smaller, & connected



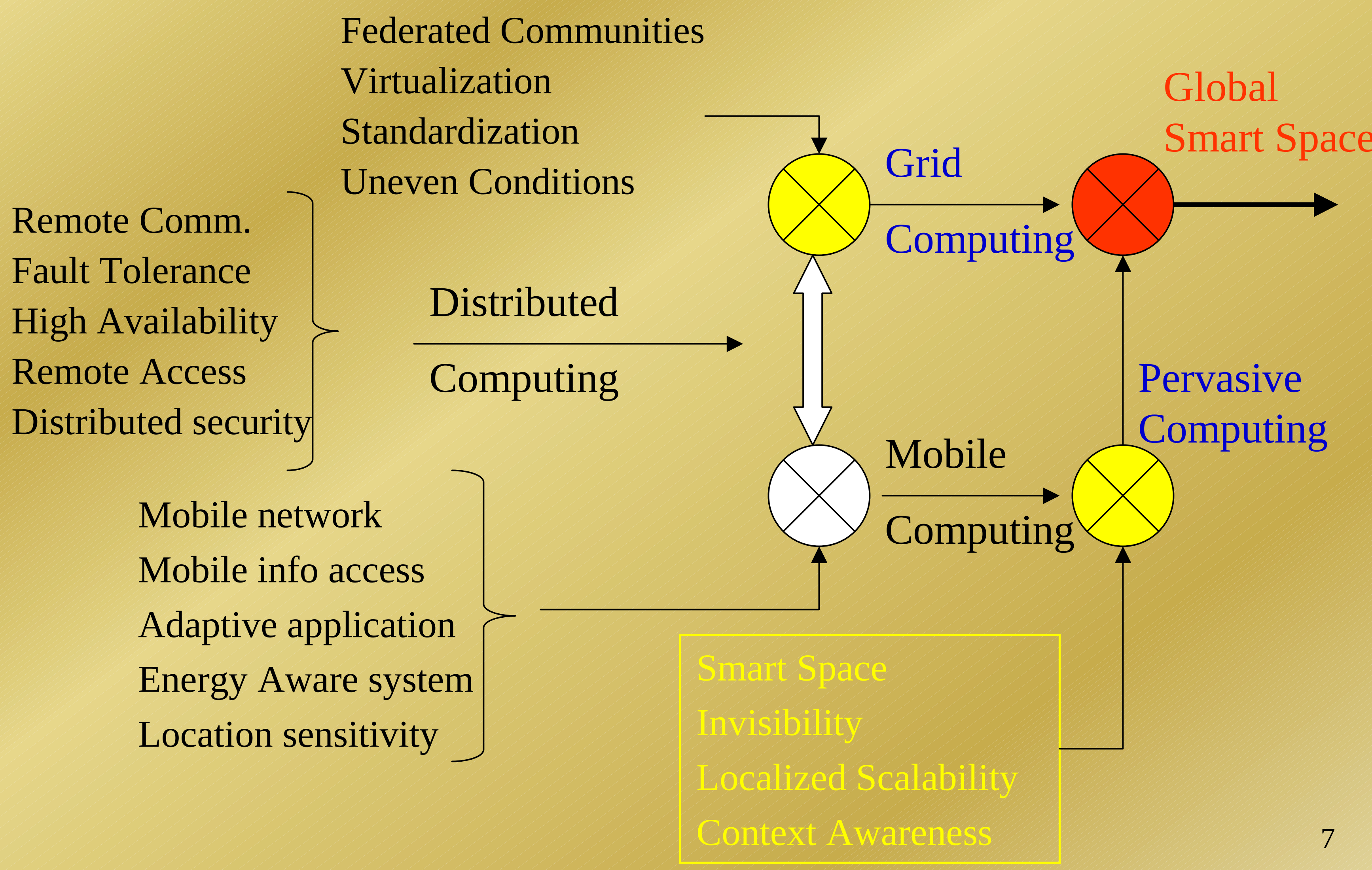
Japan’s Earth Simulator
•640 processor nodes (PNs)
•Each PN is a system with 8 vector-type arithmetic processors (APs)
•Peak performance 40Tflops

approx; 50m x 65m x 17m



1.4m x1m x 2m

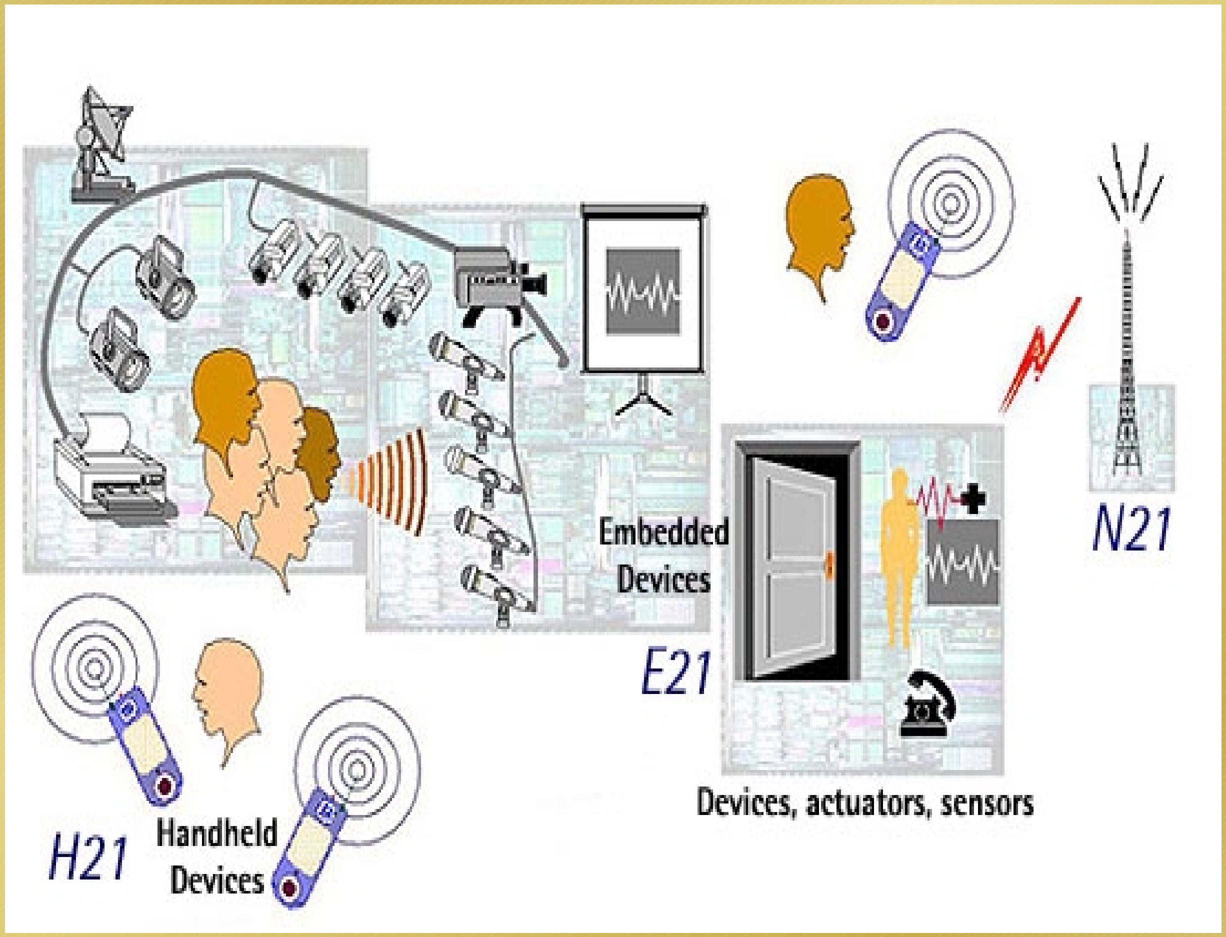
# Evolution of Computing



# Embedded Systems: What is the new

- Devices become smaller and more powerful
- Devices are coordinated via network
- From “autonomous computing” to coordinated “human-center computing”

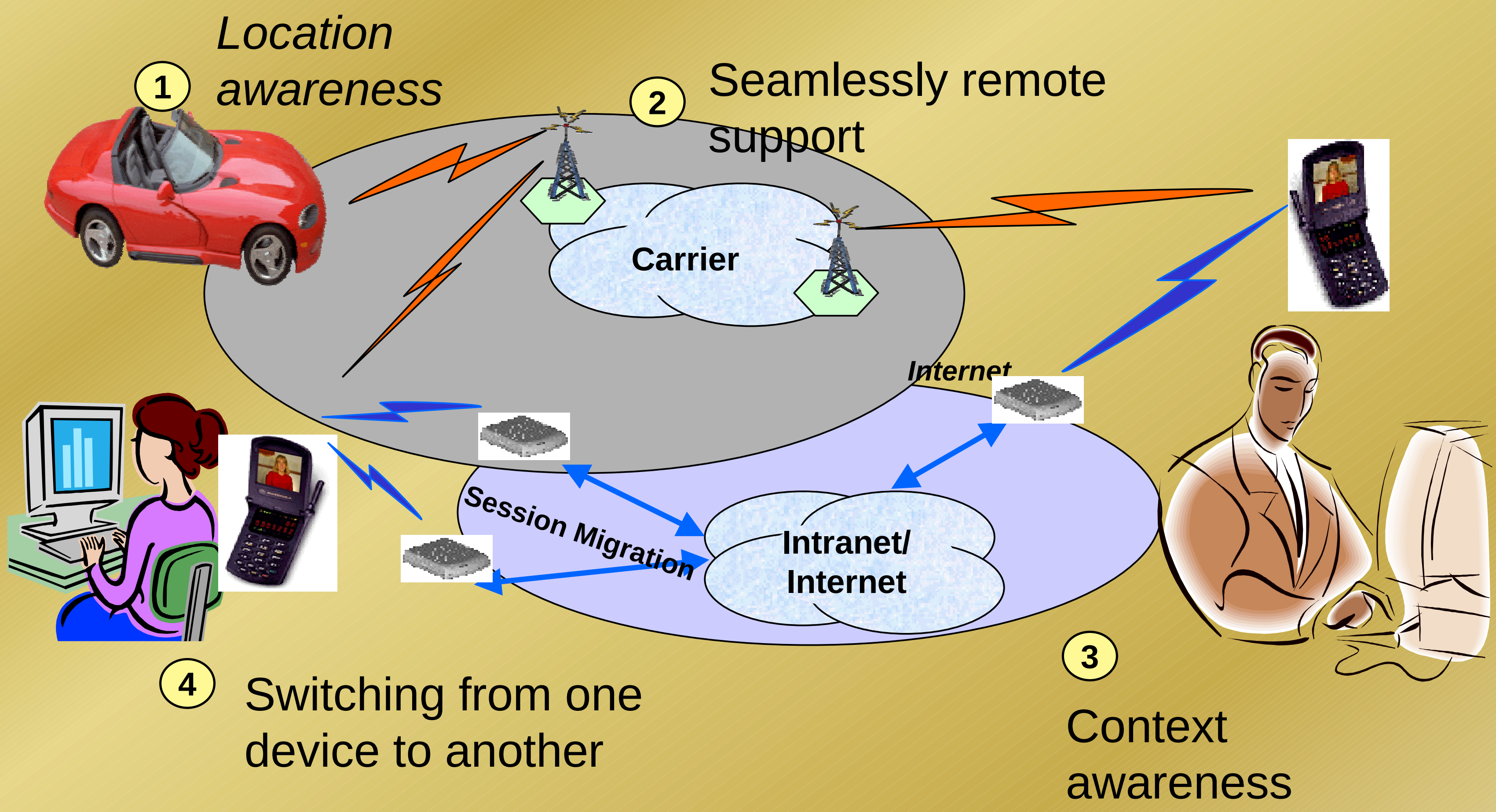
# Coordinated Embedded System – Smart Space



# Pervasive Computing

- Computers have become an embed intrinsic part of a sophisticated, networked, pervasive and ubiquitous computing environments around humans.
- Pervasive Computing: create a ubiquitous environment that combines processors and sensors with network technologies (wireless and otherwise) and intelligent software to create an immerse environment to improve life.

# Pervasive Computing Applications



# Some Current Projects


AHRI：www.cc.gatech.edu/fce/ahri
Aura: www-2.cs.cmu.edu/~aura/
Endeavour: endeavour.cs.berkeley.edu/
HawkTour：http://www.cs.iit.edu/~scs/
Portolano: portolano.cs.washington.edu/
“Oxyen”：http://oxygen.lcs.mit.edu
“Smart Space”: http://www.nist.gov/smartspace/

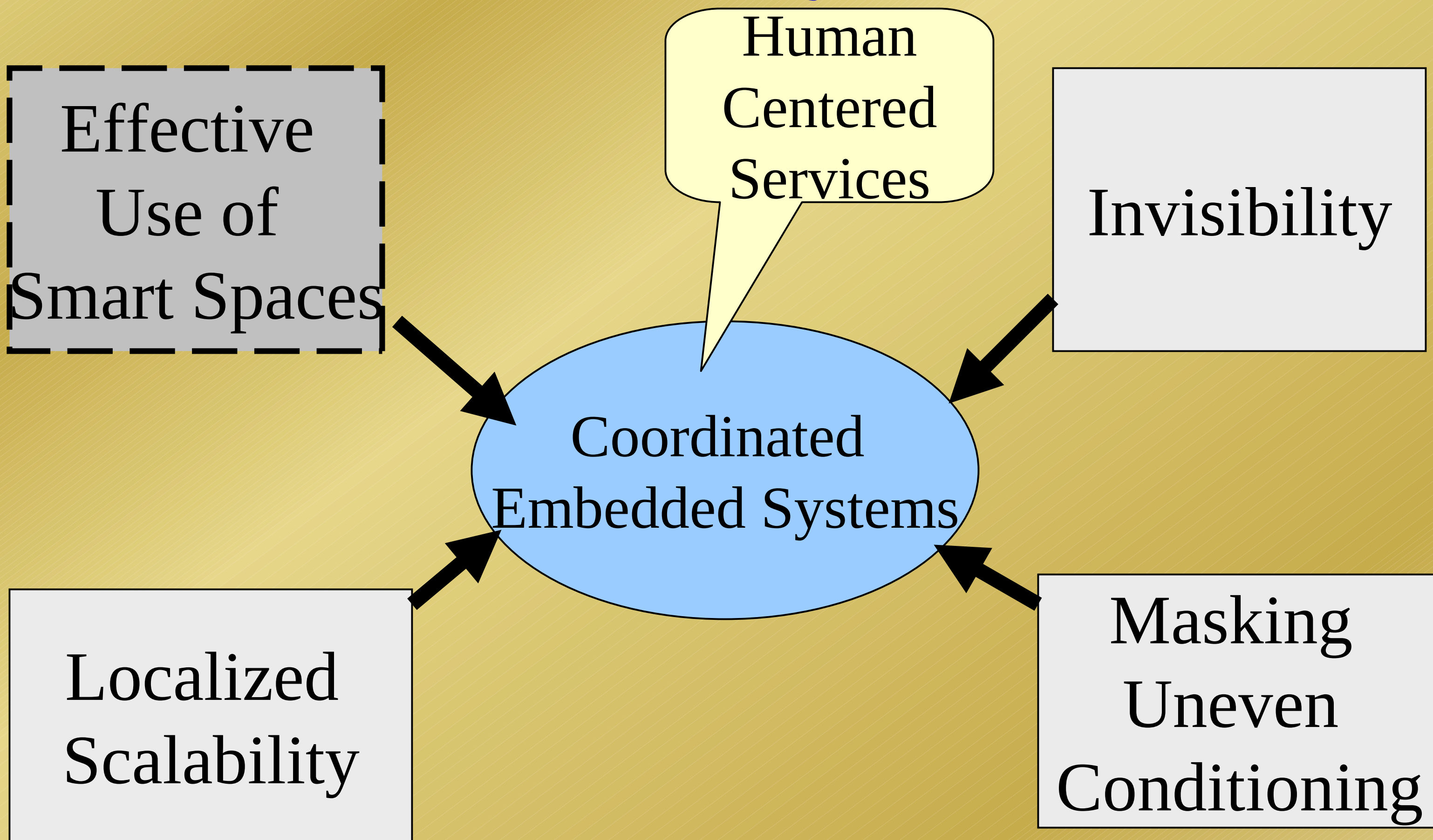
Sentient Computing: www.uk.research.att.com/spirit/
Cooltown: www.cooltown.com
EasyLiving: research.microsoft.com/easyliving
WebSphere Everyplace: www-3.ibm.com/software/pervasive

# Technical Challenges, & current achievements

| | |
|---|---|
| Context Aware Interfaces<br>Context aware | Dialog processing, Pen input, Geometry sensitive devices, Gaze tracking, Gesture recognition, Speaker identification, Acoustic imaging, Camera array processing, Speech recognition, Sensor fusion, etc |
| Mobility and Networking<br>Continuously | Directory services, Security management, Wireless protocols, Service discovery, Mobile session management, Remote sensors, Cross-network service |
| Pervasive Devices<br>Coordination | Smart notebooks, Portable sensors, Electronic books, Palm top computers, Smart badges/tags |
| Information Access<br>Locate | Visual document indexing, Spoken document indexing, Distributed multimedia data bases, Spoken document retrieval, Text retrieval |

# Software Challenges

## Context awareness and Mobility



# Context Aware Embedded System

- Context
  - Useful information other than user input
- Context Awareness
  - Ability to *capture*, *understand* and *adap*t to surrounding context information
- Context aware embedded system
  - Capture context information via 'embedded' devices Takes action without explicit user input
  - Improves user experience by achieving collaboration and integration of embedded systems

# Role of Context

### Traditional Class Environment

- Professor T informs students about the updated course website for lecture slides
- They need to bring the slides in the class for better understanding
- Some of the students either did not read the notification
- some of them forgot about it before the class

### Smart Class Environment

- If
  - Professor T is moving towards the projector and
  - lights in the room are off
- Then the environment pervasively transfers the presentation slides from the professor's handheld device to students' handheld device
- The projector starts the presentation

# Challenges of Context Awareness

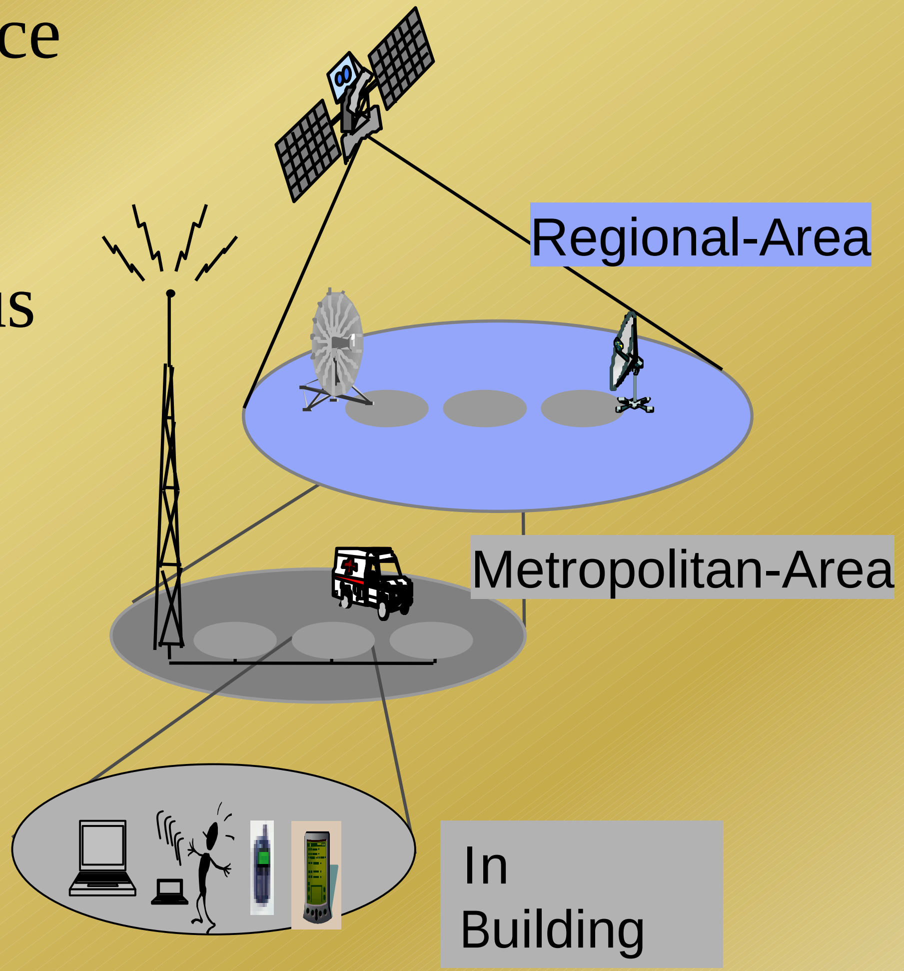
- Context awareness: aware of the user's state and surroundings and help the system to adapt its behavior accordingly
  - How does the system represent context internally?
  - How frequently does the system need to consult contextual information
  - What are the minimal services that an environment must provide?
  - What are the relative merits of different location-sensing technologies
  - Trade off between awareness and transparency

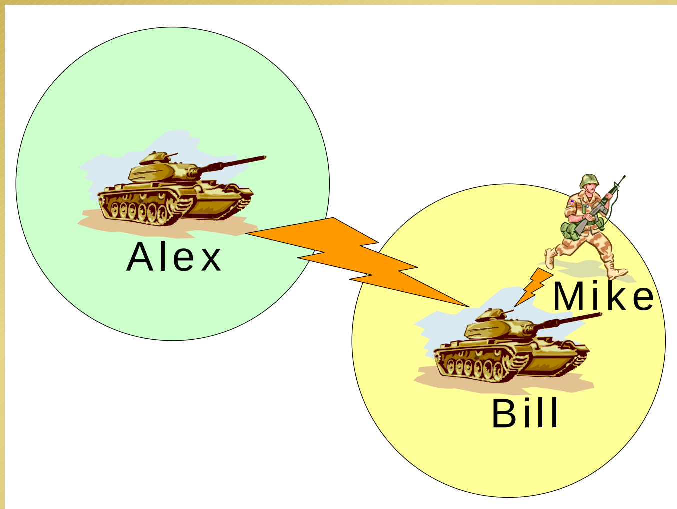
# Mobility in Pervasive Computing

- User Mobility: Providing users with a uniform view of their preferred working environments
- Terminal Mobility: Allowing devices to transparently move and connect to different points of attachment
  – Network mobility
  – Code mobility, computing mobility
- Mobile Access: Dynamic adaptation of mobile-aware resource and service
  – Service mobility

# Role of Mobility

- Any time, any where service
- Device, network mobility
- Adaptation, context aware
- Application software versus infrastructure system



# Case Study: Context Aware System

Scarlet: the IIT context aware infrastructure

- Separate infrastructure functionality with application-specific functionality
- Functional Requirements
  - Context collection, Context Storage/Management, Context Subscription/Delivery, Context Analysis/Composition Ability
- Non Functional Requirements
  - Scalability, Modularity, Cross platform, Security, Extensibility, Ability to Evolve, Quality of Service, Fault Tolerance, Mobility, User Friendly Interface

# Separation of Application and Infrastructure

## Infrastructure

(2) Decide and find the devices, which can provide above requested information

(3) Collect and store context information on timely basis

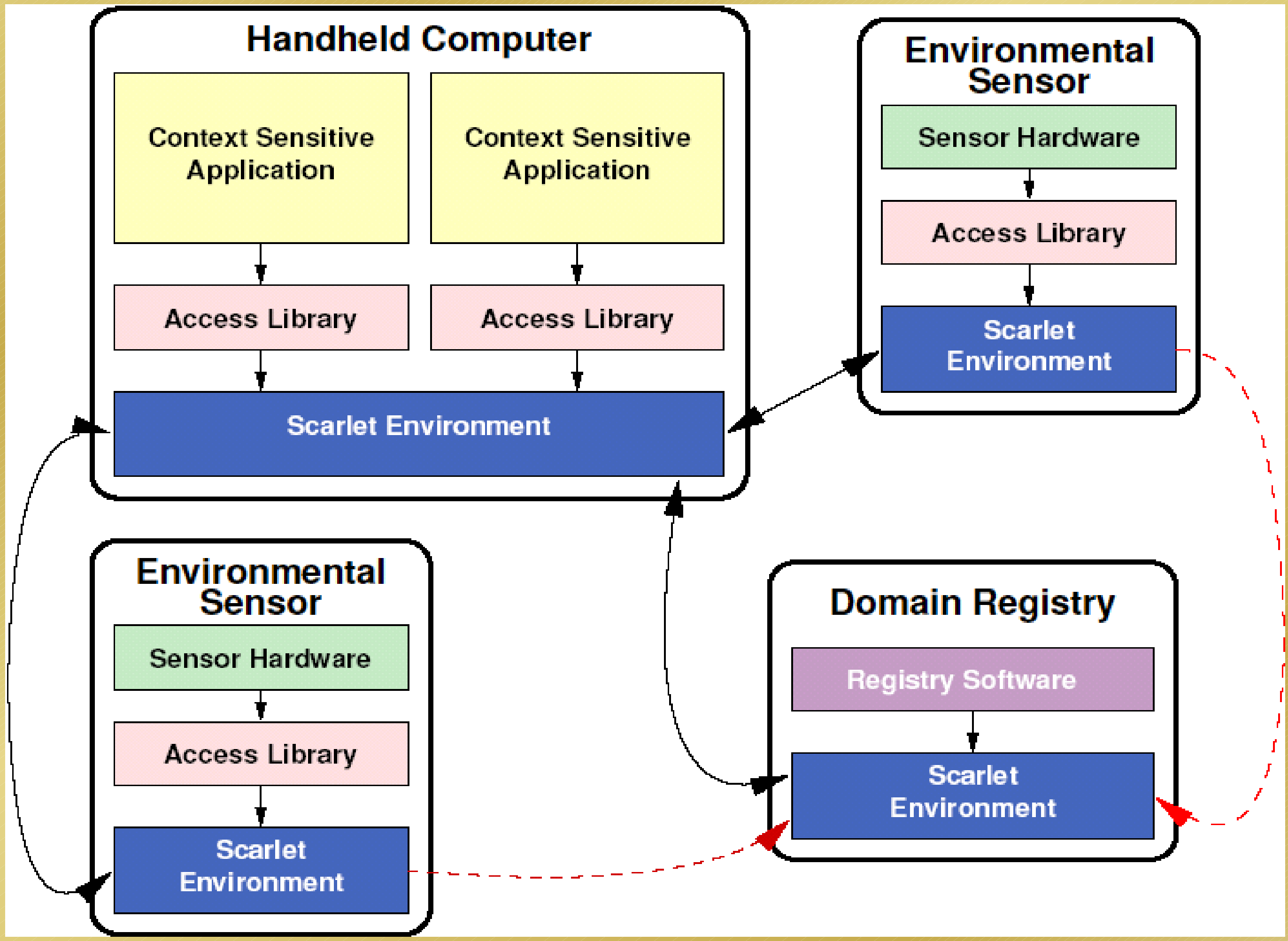
(4) Compose raw context information to meaningful situation

## Application

(1) Decide the desired context information

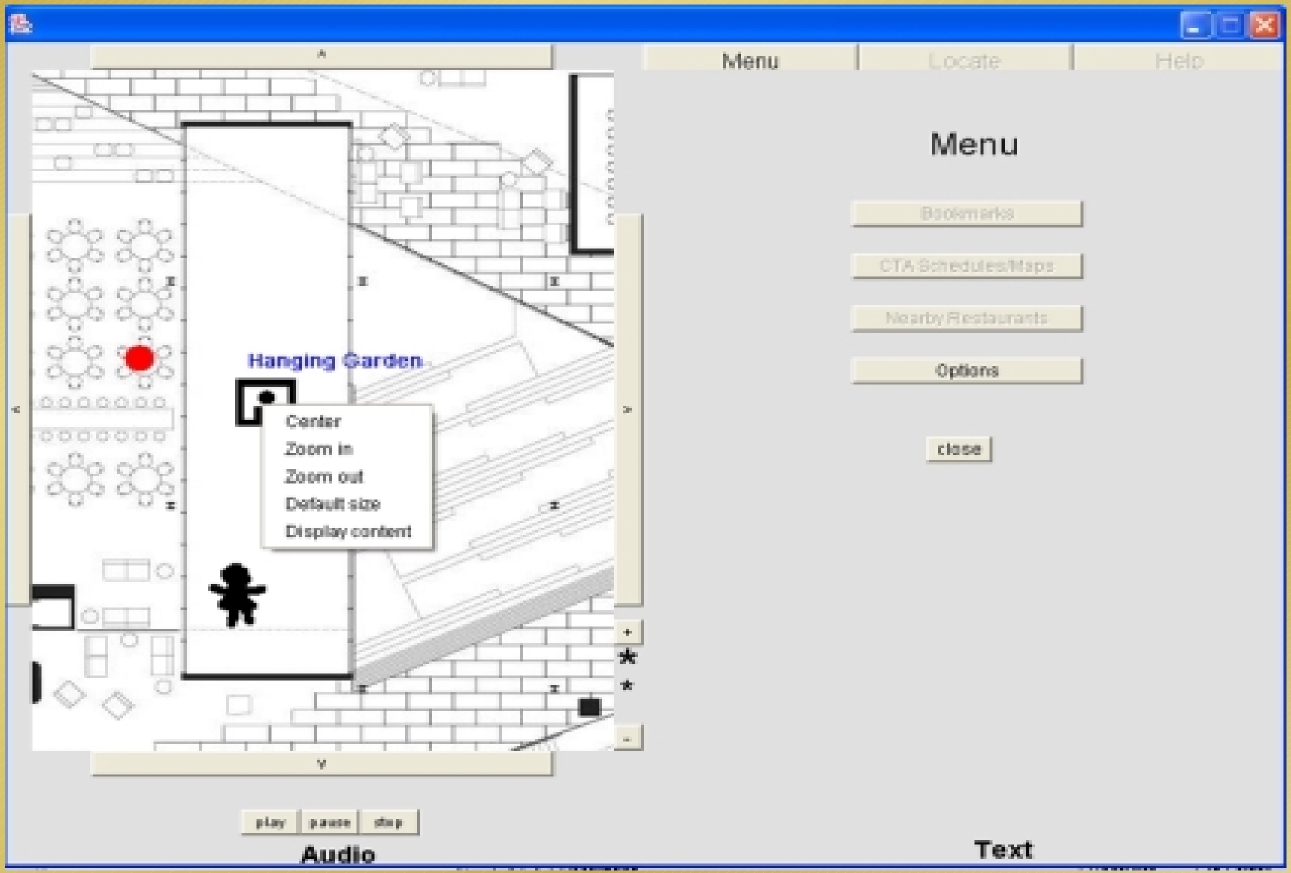
(5) If a meaningful situation occurs then take some action.

# The Scarlet System

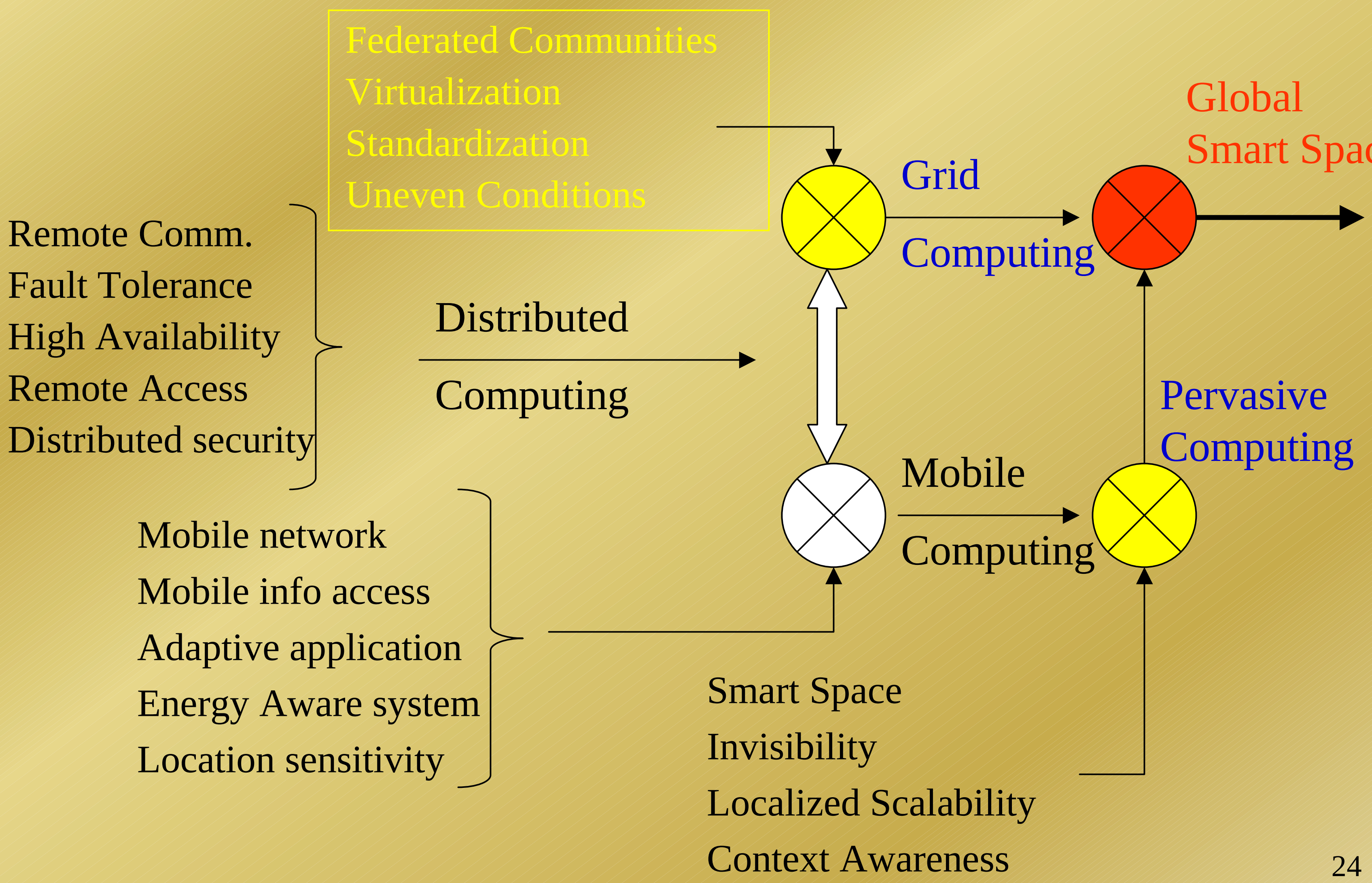


# Sample Applications

- Service Browser
- Wireless Strength Monitor
- Television Assistant
- Tour Guide

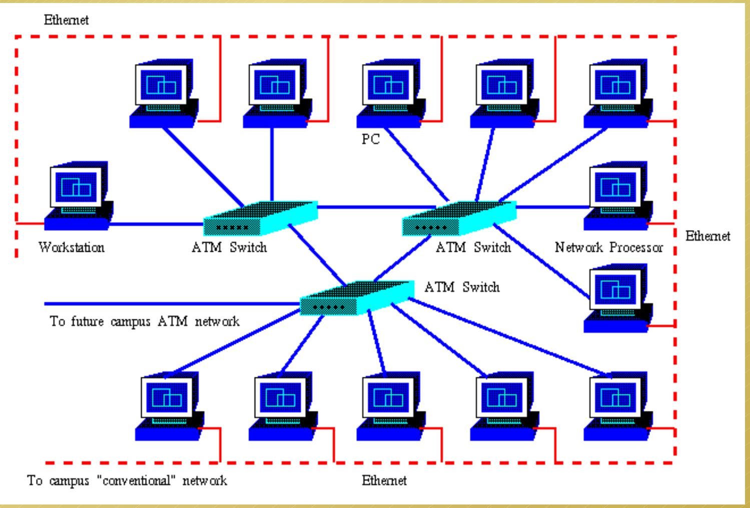


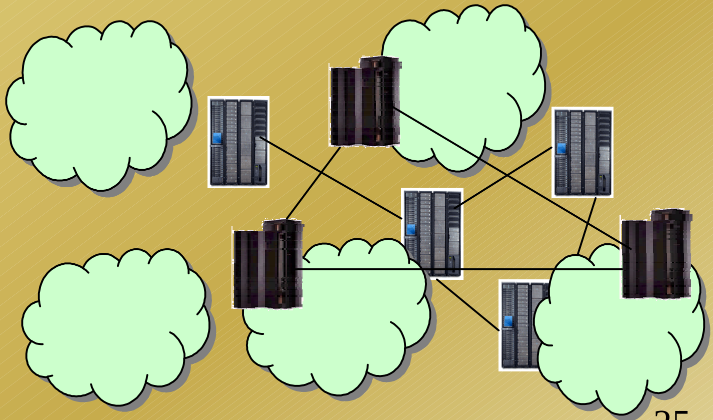
# Evolution of Computing



# Distributed Computing: What is the new

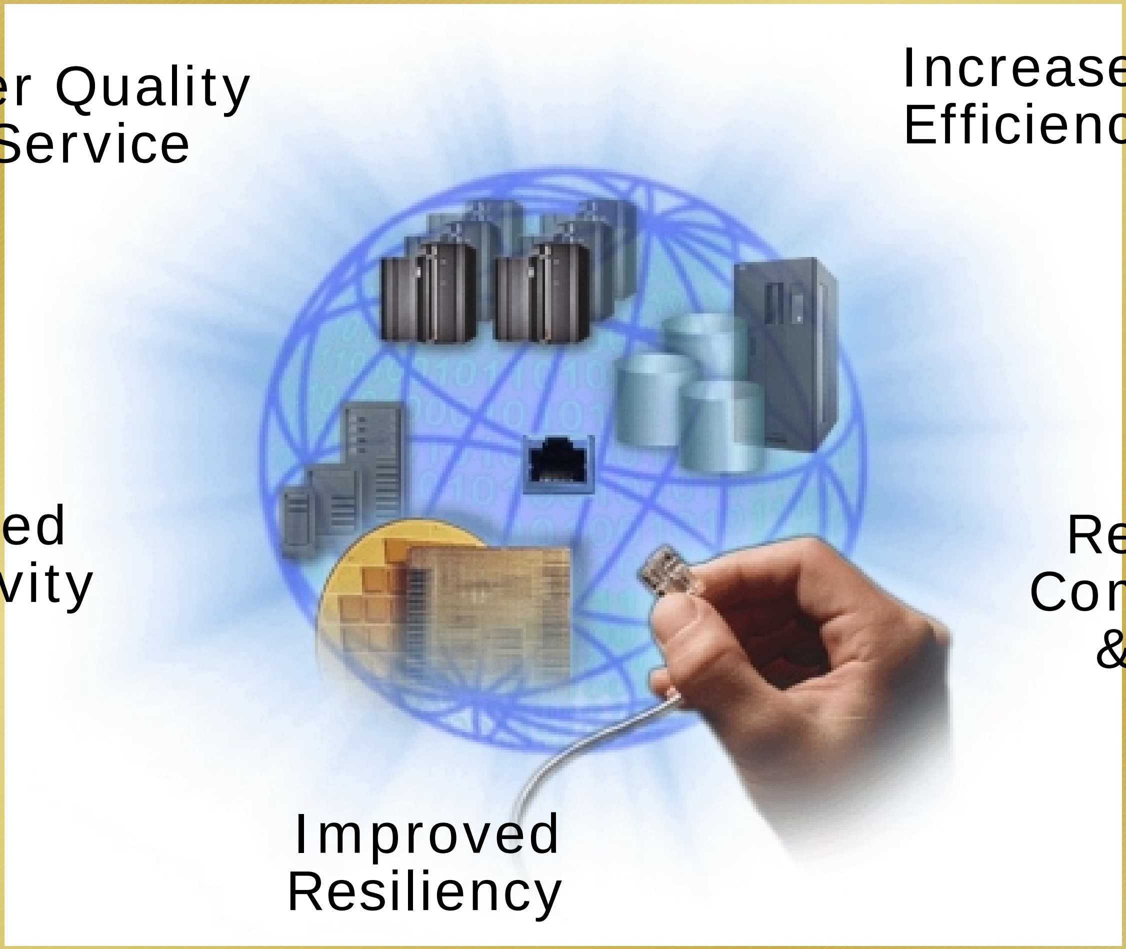
- Supercomputers become ever powerful
- Communities of "Virtual organizations" are formed
- No VO possesses all required skills and resources
- From "community sharing" to "information grid"



# Integrated VOs: the Grid

Mimic the electrical power grid

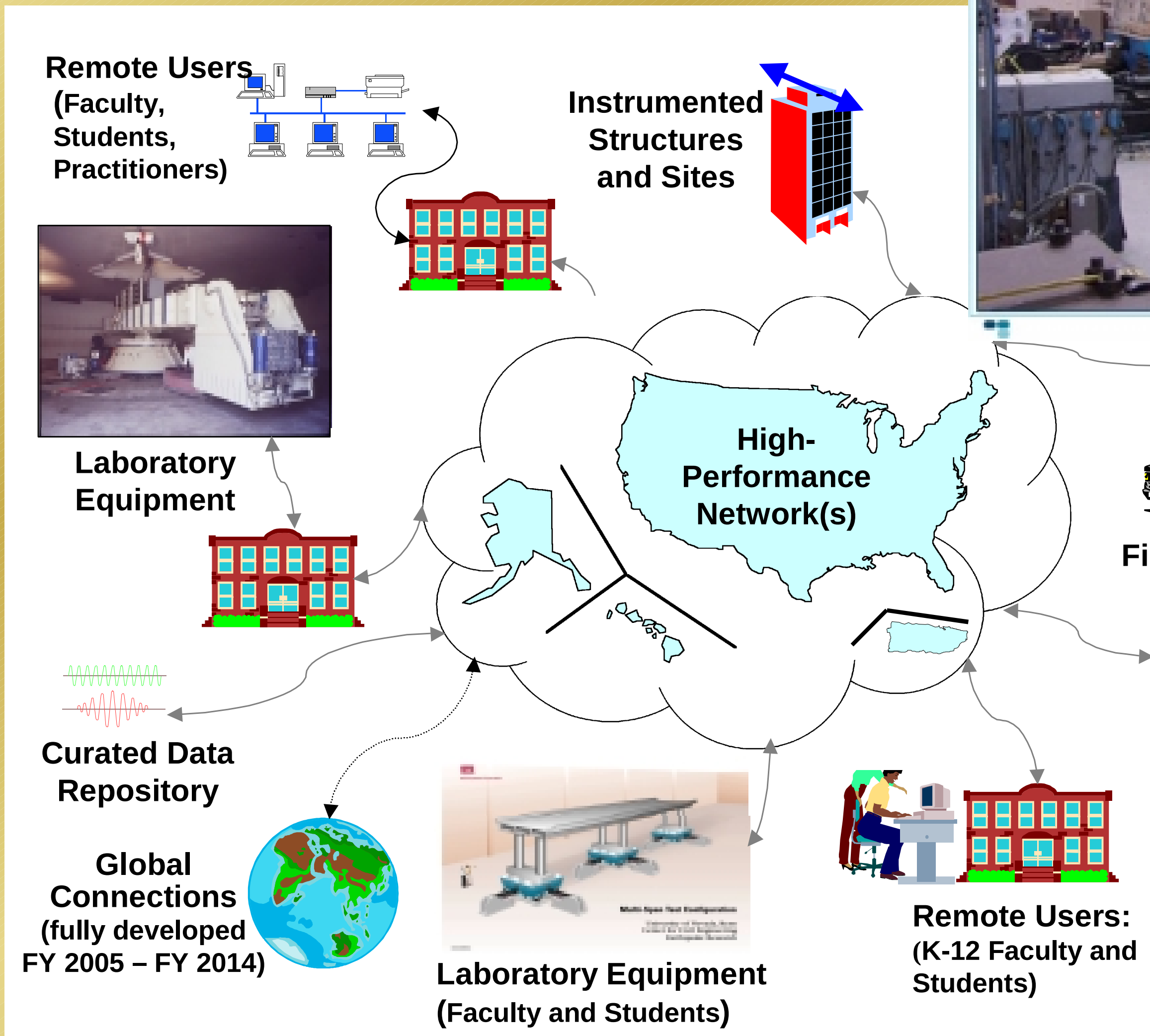


# The Grid Computing

- Infrastructure (“middleware” & “services”) for establishing, managing, and evolving multi-organizational federations
- Mechanisms for creating and managing workflow within such federations
- Three key criteria
  - Coordinates distributed resources …
  - using standard, open, general-purpose protocols and interfaces …
  - to deliver non-trivial qualities of service.

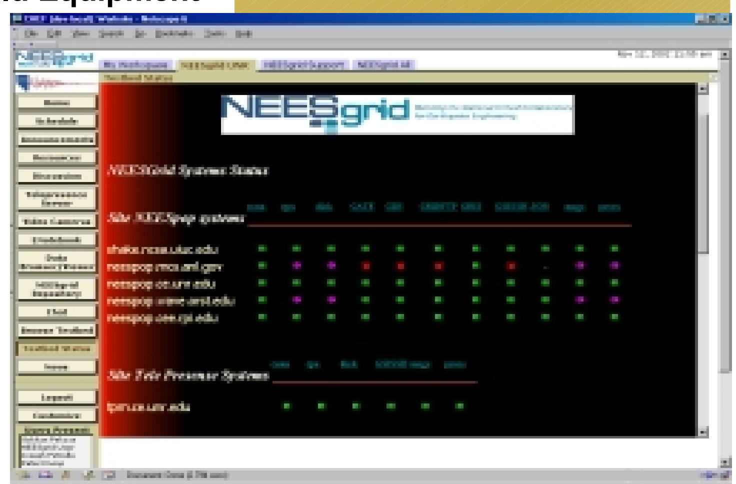
# Grid Computing Application NEESgrid

Earthquake Engineering Collaboratory




U.Nevada Reno

www.neesgrid.org



# More Information of Grid

- The Globus Alliance®: www.globus.org
- Global Grid Forum: www.ggf.org
- GlobusWORLD: www.globusworld.org
- GCC 2003
  - http://www.cs.sjtu.edu.cn/gcc2003/index.htm
- Projects:
  - The EcoGRID (Economy Grid) project at Monash University
  - The Legion project at the University of Virginia
  - The Polder project at the University of Amsterdam
  - The MOL project at the University of Paderborn

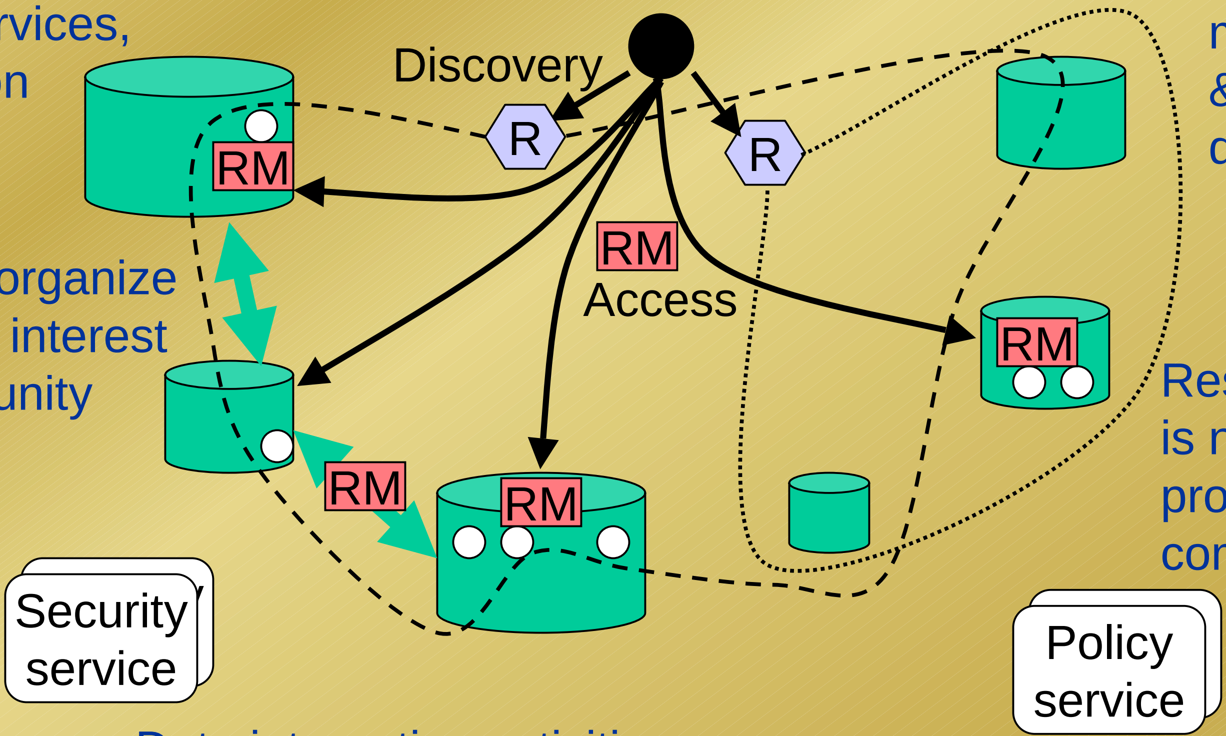
# The Challenge of Resource Integration

## Standardization and Service

Many sources of data, services, computation

Security & policy must underlie access & management decisions

Registries organize services of interest to a community




Resource management is needed to ensure progress & arbitrate competing demands

Data integration activities may require access to, & exploration/analysis of, data at many locations

Exploration & analysis may involve complex, multi-step workflows

# Why Open Standards Matter

- Ubiquitous adoption demands open, standard protocols
  - Standard protocols enable *interoperability*
  - Avoid product/vendor lock-in
  - Enables innovation/competition on end points
- Further aided by open, standard APIs
  - Standard APIs enable *portability*
  - Allow implementations to port to different vendor platforms
- Open architecture and specification (infrastructure)
  - Internet and Web as exemplars
  - Web Service as a basis

# Web Services

- XML-based distributed computing technology
- Web service = a server process that exposes typed ports to the network
- Described by the Web Services Description Language, an XML document that contains
  - Type of message(s) the service understands & types of responses & exceptions it returns
  - "Methods" bound together as "port types"
  - Port types bound to protocols as "ports"
- A WSDL document completely defines a service and how to access it

# Open Grid Services Architecture

- Everything is a service
- A standard substrate: the Grid service
  - A Grid service is a Web service
  - Standard interfaces and behaviors that address key distributed system issues: naming, service state, lifetime, notification
- Supports standard service specifications
  - Agreement, data access & integration, workflow, security, policy, diagnostics, etc.
  - Target of current & planned GGF efforts
- Supports arbitrary application-specific services based on these & other definitions

# Challenges of OGSA

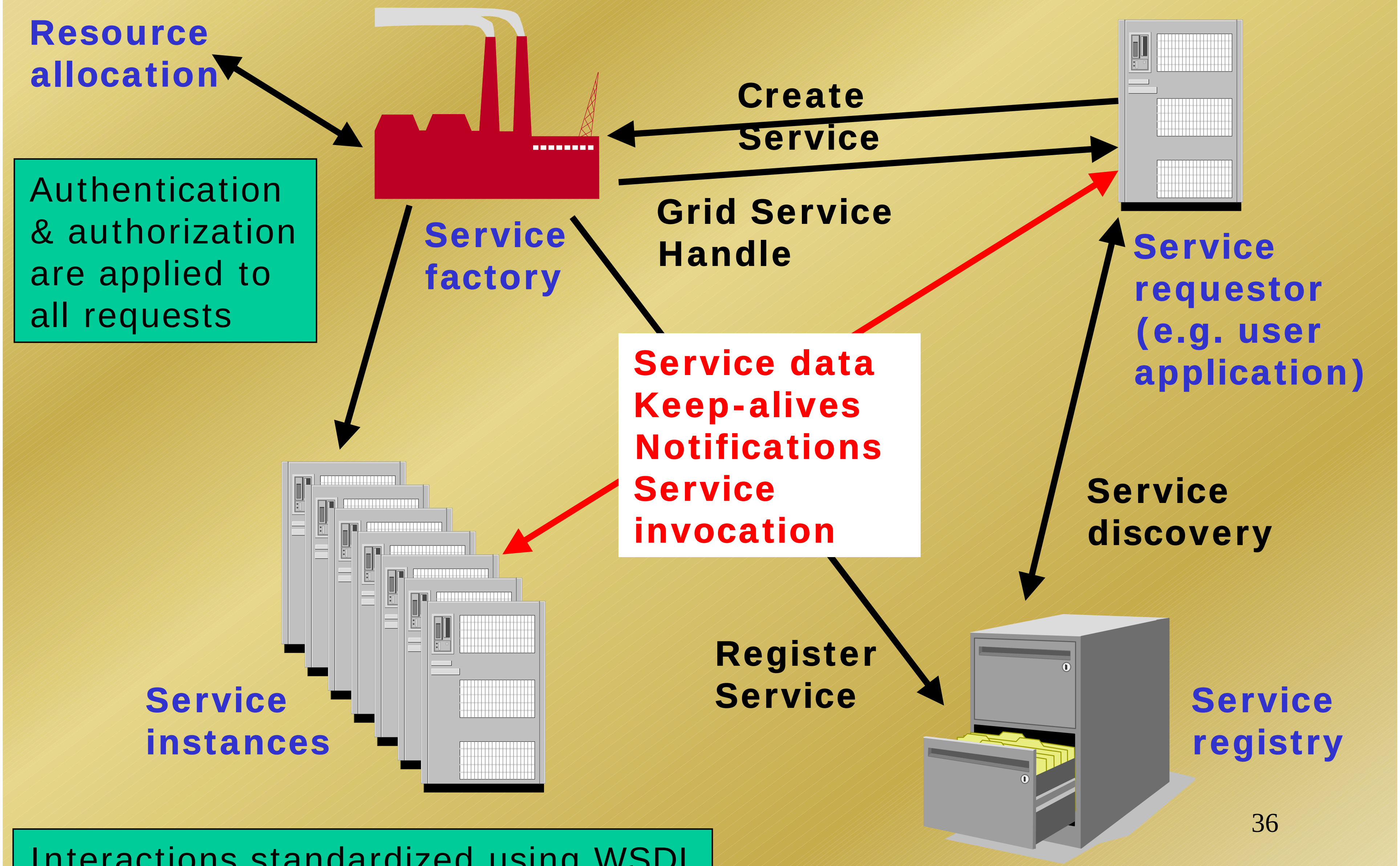
- From Web services
  - Standard interface definition mechanisms
  - Address discovery & invocation of persistent services
  - Evolving set of other standards: security, etc.
- From Grids (Globus Toolkit)
  - Service semantics, reliability & security models
  - Support transient services, created/destroyed dynamically
  - Lifecycle management, discovery, other services
  - Interfaces to the states of distributed activities

    E.g. workflow, video conf., dist. data analysis
- A framework for the definition & management of composable, interoperable services

# OGSI Specification

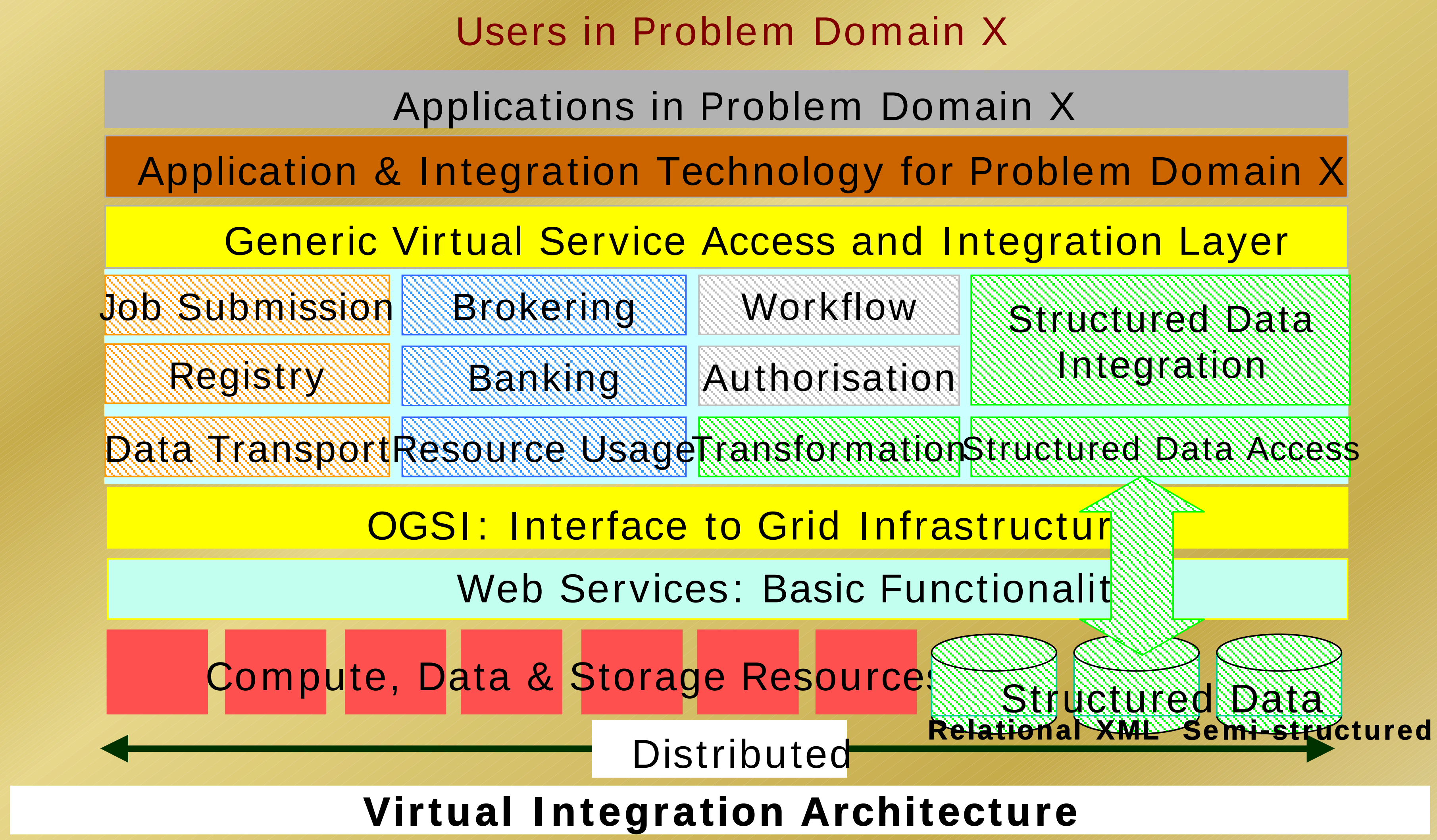
- Defines fundamental interfaces (using extended WSDL) and behaviors that define a Grid Service
- Defines basic patterns of interaction
- Specification focuses on:
  - Naming and bindings (basis for virtualization)
  - Lifecycle (basis for fault resilient state management)
  - Information model (basis for monitoring & discovery)
  - Service Groups (basis for registries & collective svcs)
  - Base Fault type
- Foundation to develop application specific services

# Open Grid Services Infrastructure (OGSI)




Interactions standardized using WSDL

# Open Grid Services Architecture

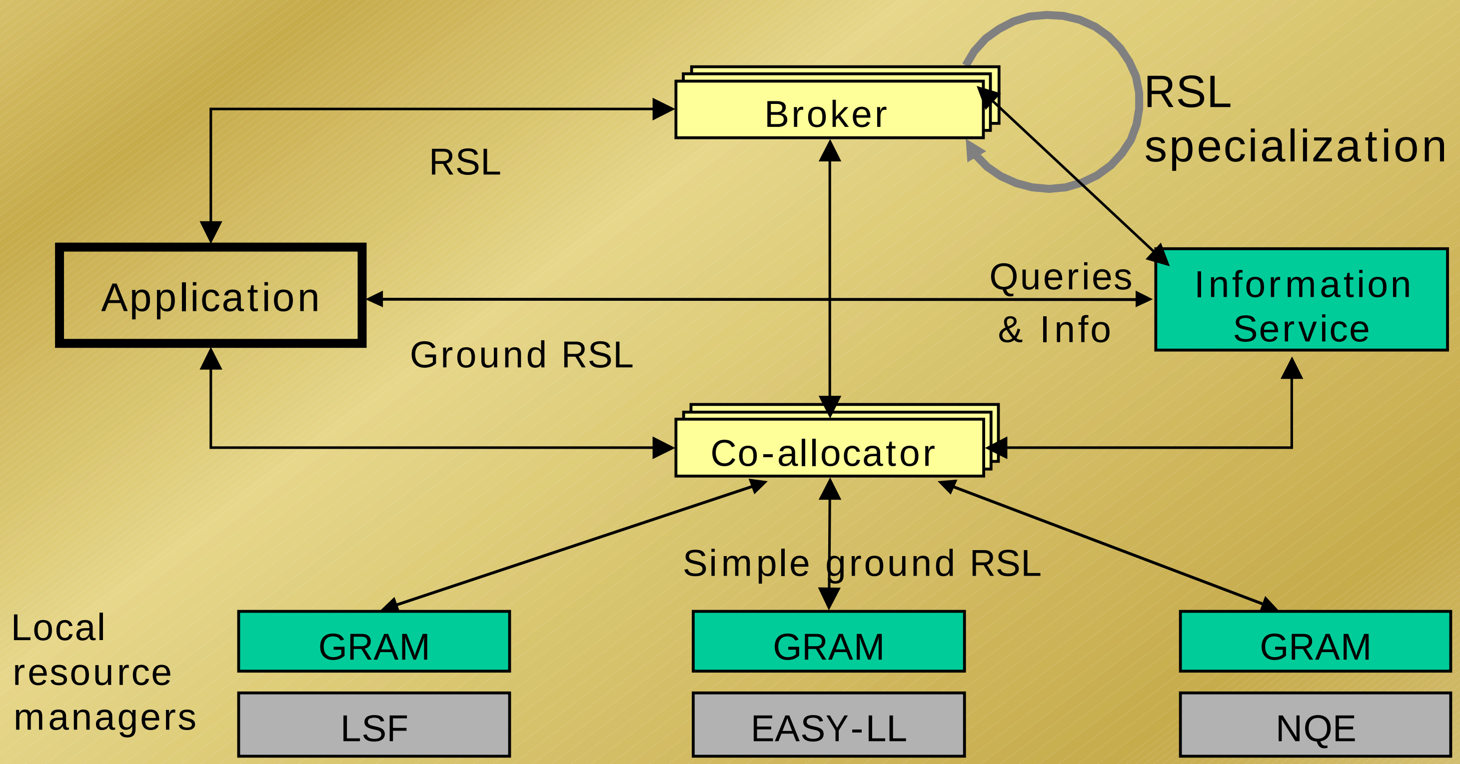


# Grid Service: Resource Management

- Discovery/characterization
  - Find relevant service providers
- Diversity in resource type, local managers, policy
  - Requires normalized remote interface
- Limited knowledge, abstraction
  - Impact of local users, other VOs
  - Hidden policy, operational issues
- Virtualization
  - 3rd party providers, brokers, mediators
- Coordination

# Example: Resources Management

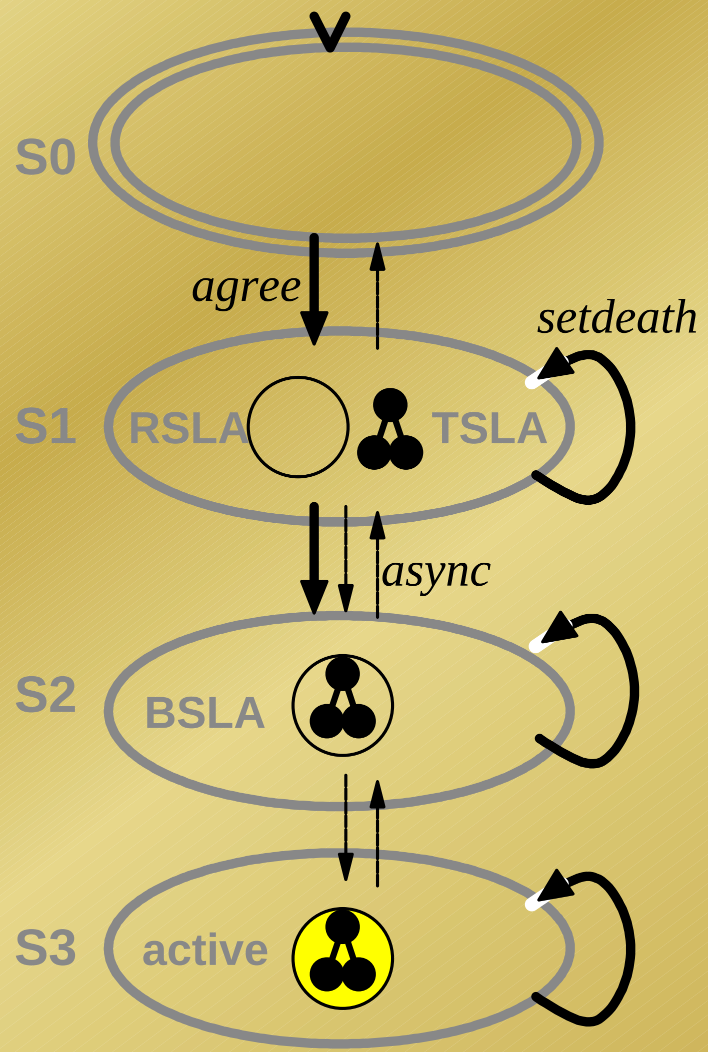
The Globus approach (Argonne, UC & USC)



# An Agreement Based Approach

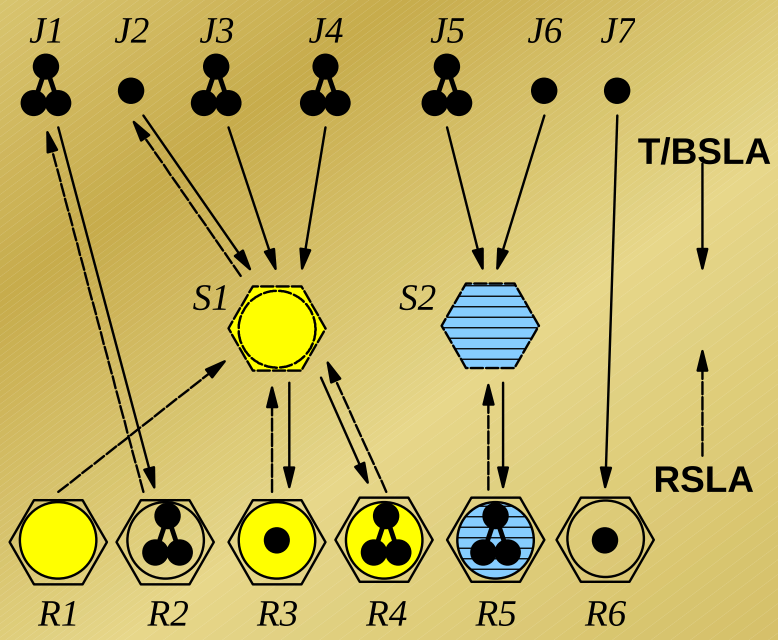
- Service level agreements (SLA) to abstract resource
  - You are what you negotiate
- Resource SLA (RSLA), i.e. reservation
  - A promise of resource availability
    - Client must utilize promise in subsequent SLAs
- Task SLA (TSLA), i.e. execution
  - A promise to perform a task
    - Complex task requirements
    - May reference an RSLA (implicit binding)
- Binding SLA (BSLA), i.e. claim
  - Binds a resource capability to a TSLA
    - May reference an RSLA (otherwise obtain implicitly)
    - May be created lazily to provision the task

# Resource Lifecycle




- S0: Start with no SLAs
- S1: Create SLAs
  - TSLA or RSLA
- S2: Bind task/resource
  - Explicit BSLA
  - Implicit provider schedule
- S3: Active task
  - Resource consumption
- Backtrack to S0
  - On task completion
  - On expiration
  - On failure

# Community Scheduling Example




- Individual users
  - Require service
  - Have application goals
- Community schedulers
  - Broker service
  - Aggregate scheduling
- Individual resources
  - Provide service
  - Have policy autonomy
  - Serve above clients

# Summary

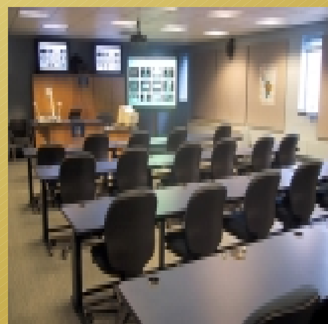
- IT extreme success is the disappear of computer!
- Bigger machines become even bigger forming the computing Grid, computing in remote
- Small computing devices become even smaller forming the smart space,  devices invisible
- Smart space is the interface, Grid is the backbone
- Both Grid and Pervasive Computing are in their infancy stage but real

# Questions?

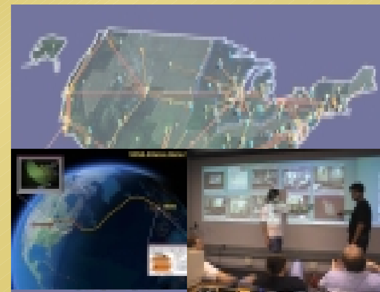
- Web triggered the last IT boom. Would Web Service (Grid) leads to another boom?
- How Pervasive technologies would improve the quality of life?
- What is the killer technology?
- What are the killer applications?
- Dedicated, special purpose Grid and pervasive service
  - High physics application, utility computing, the eco-system, etc.
  - Where is the triggering special service?

# The View of Future Computing

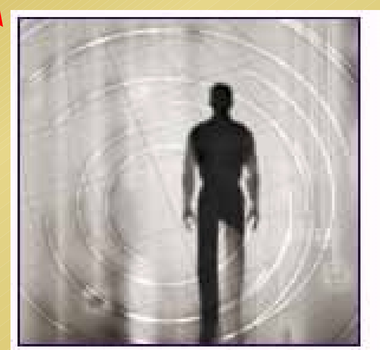
## Human-centered



They are connected to form `smart space’

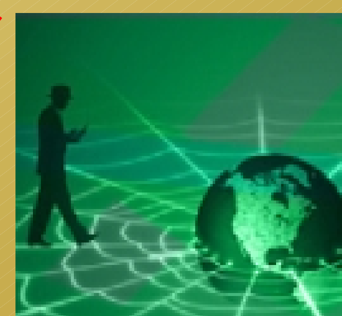


Devices become smaller and powerful



Grids link `smart spaces’ to support `global smartness’

A device is an entry of the cyber world



# Conclusion

- The technical foundation of next IT boom is emerging
- It may need the change of computing infrastructure and concept, and may start slow
- Waiting for the killer applications
- www.cs.iit.edu/~scs/

BUSINESS

'Plain old telephone service' could get a lot more exciting

PHONE