FIWC 2009, SNU
Feb. 24, 2009

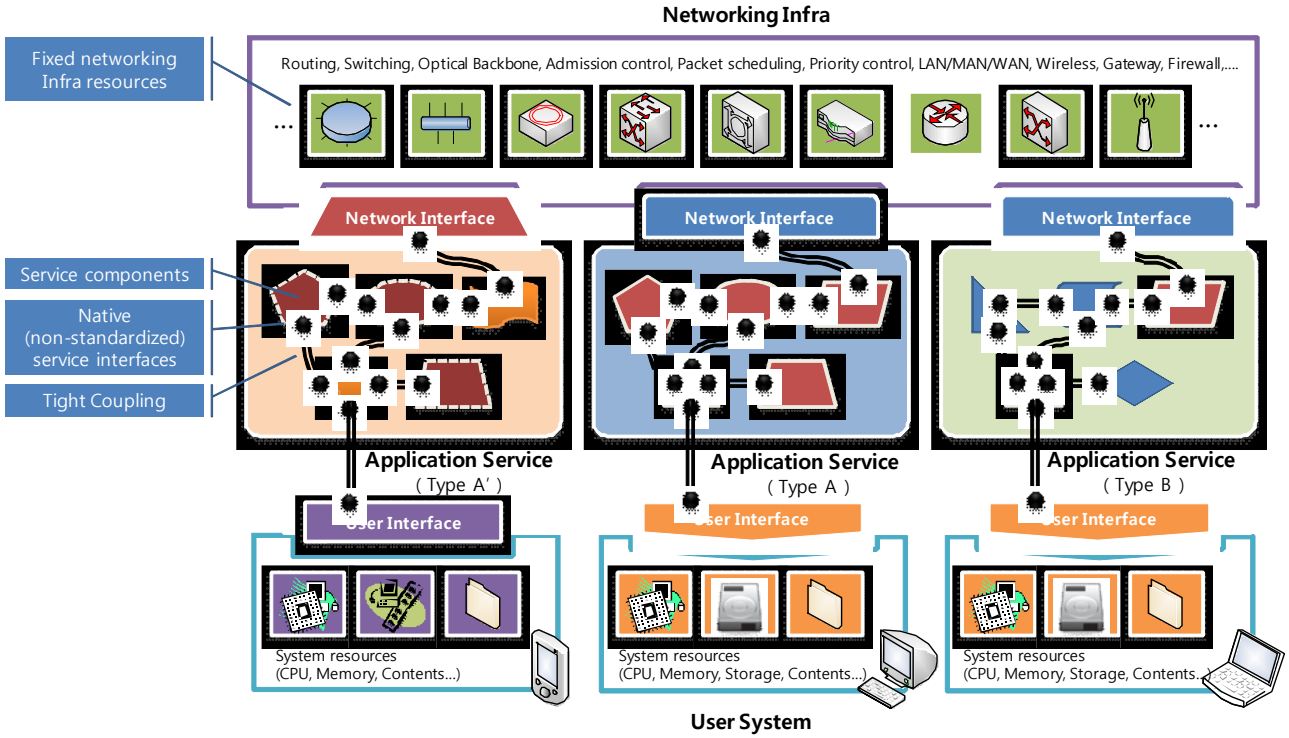
Experience with MultiAgent-Based Distributed Service Composition

Sang Woo Han and JongWon Kim
{swhan, jongwon}@nm.gist.ac.kr

Networked Media Lab.,
Dept. of Information and Communications
Gwangju Institute of Science and Technology (GIST)

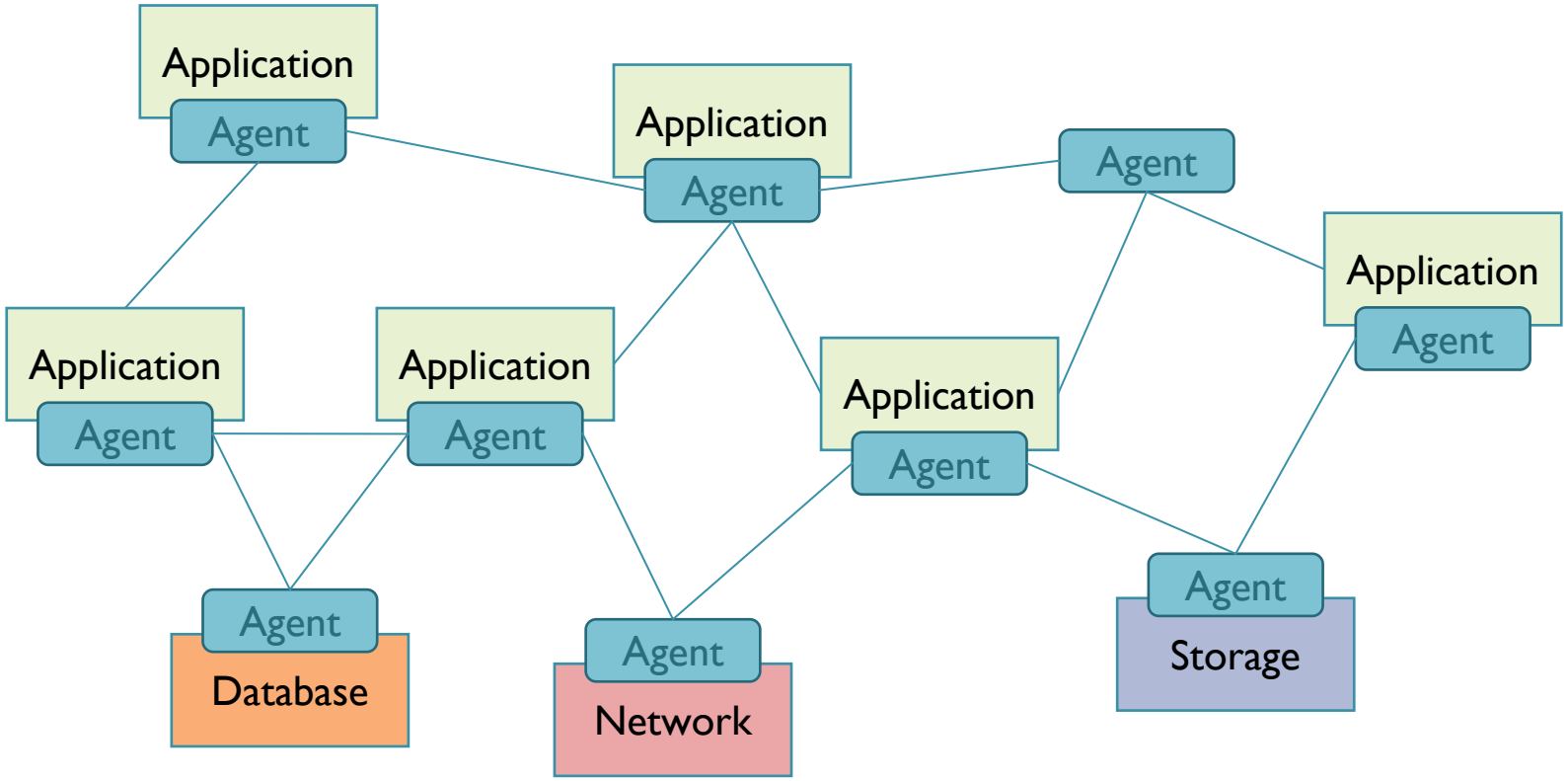
Motivation

- From the viewpoint of applications
  - Non-awareness of user context, system resources and infra status
  - Low scalability, availability, adaptability caused by tightly coupled service components via the native (non-standardized) interfaces



Objectives

- Easy experiment control and management of distributed services using MultiAgent Systems



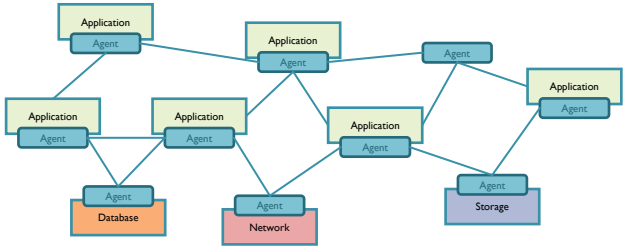
** The above illustration is referred to M. P. Singh and M. N. Huhns, "Service-Oriented Computing," WILEY, 2005.*

Application-Specific Requirements

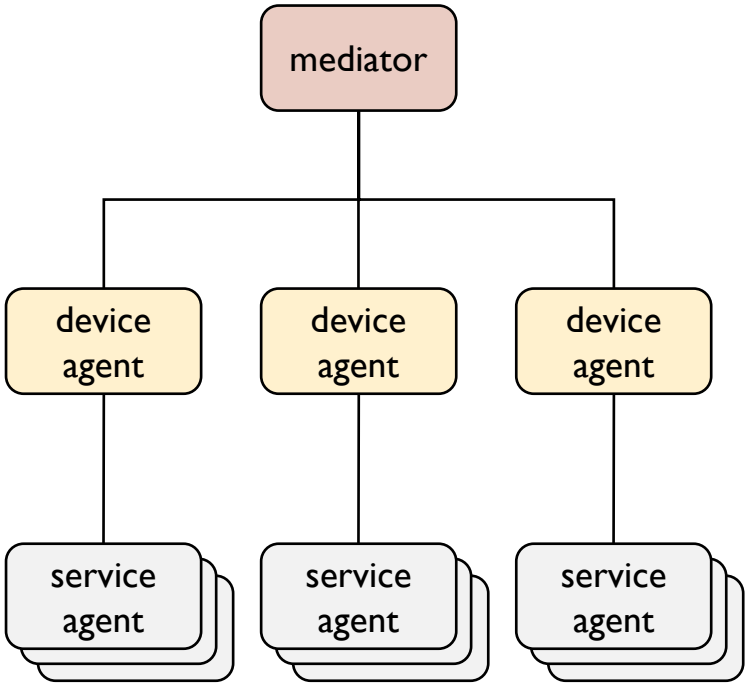
- Description
  - Context
  - Open interfaces
- Formal Experiment Specification
  - Graphical user interfaces
  - Command-line interfaces
  - Programmactic methods
- Operation and Control
  - Discovery
  - User operation
- Leasing Resources
  - SLA contract and negotation
- Management
  - Service/SLA monitoring
  - Distributed service transaction
  - Experiment schedulling and service allocation

** The black-colored parts are partially covered by this work.*

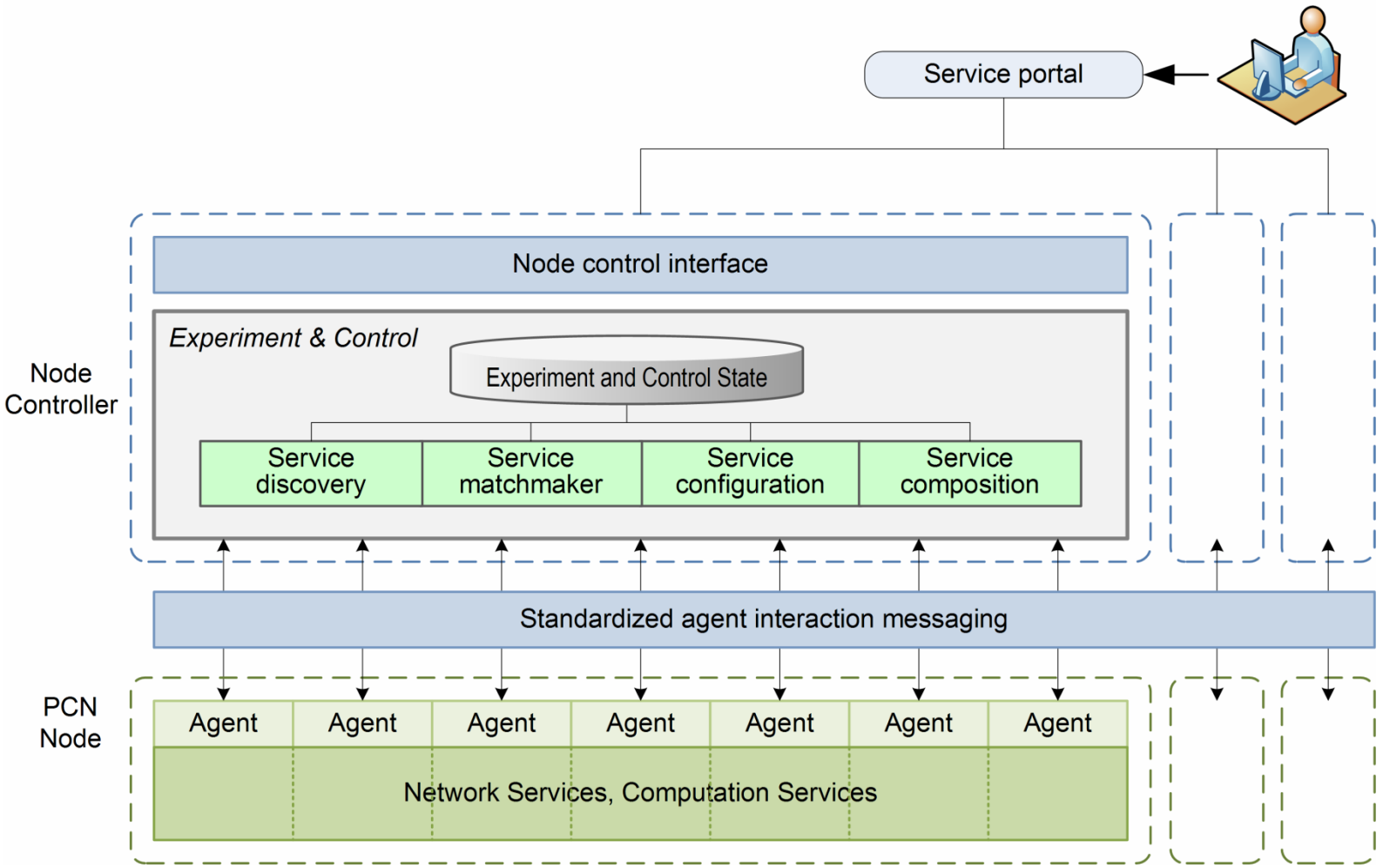
Testbed Structure



- Mediator
  - Service and device discovery, configuration, and composition
- Device
  - Service holder placed on a device
- Service
  - Software to offer an access to networking/computing resources



Agent-based Service Operation, Composition and Control

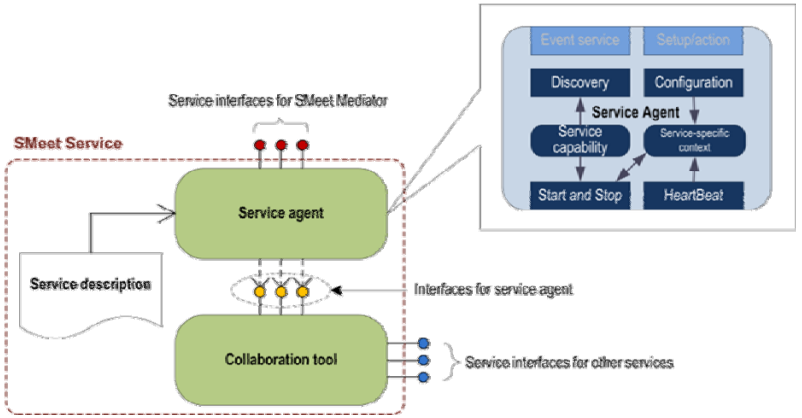


Description

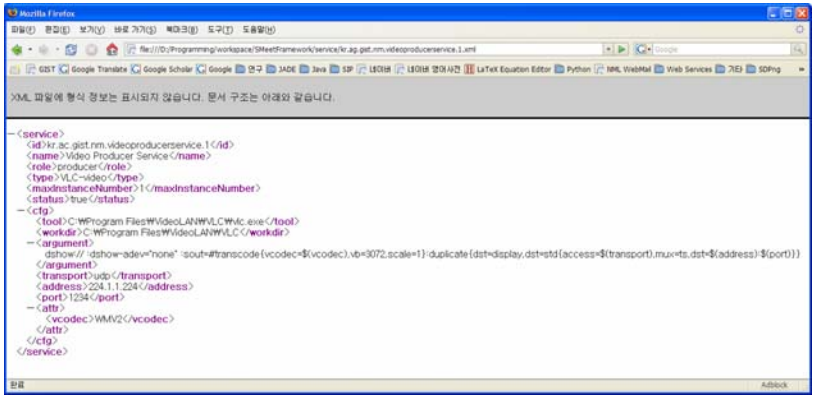
- Open interfaces
  - Described as agent actions by FIPA Interaction Protocols (e.g., request, query, contract net, and auction)
  - These open interfances can be exposed as WebServices
- Context decription
  - Possible to describe informative elements (e.g., capability, state) in terms of services and devices, using ontology
  - For example, ontologies of video producer service and network monitoring service

Description (cont')

- Service to accomodate diverse applications and tools

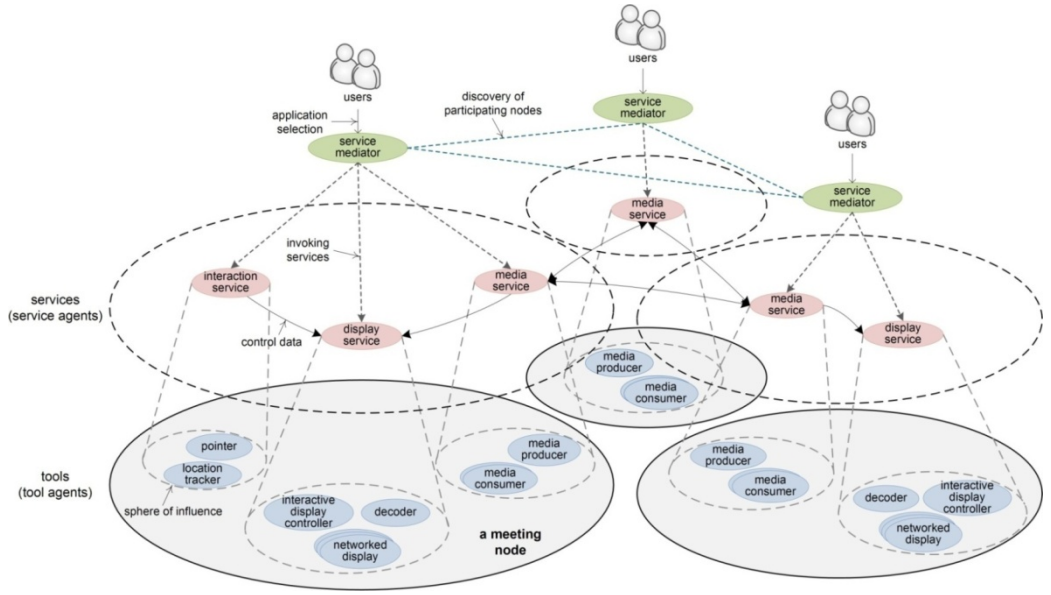


- Service to accomodate diverse applications and tools



Describing Service Composition

- Formal experiment specification (in the context of Future Internet) to combine a set of selected component services connected by:
  - Programmatic methods
  - Functional dependency graphs



Describing Service Composition (cont')

- Programmatic Methods
  - Predefining experiment code with node control API

Name	Parameters	Description
nodes		Get a list of participating nodes Return value: a set of node ID(s)
startup	properties	Launch an agent matching with the given properties Return value: agent ID
shutdown	agent ID	Shutdown an agent having the given agent ID
search	node ID, service ID, properties	Search agent(s) having given properties Return value: a set of agent ID(s)
request	agent ID, a request message	Send an agent request message to call a interface

  - Mediator compose involved service agents that begin actual interface binding and inter-working procedures.

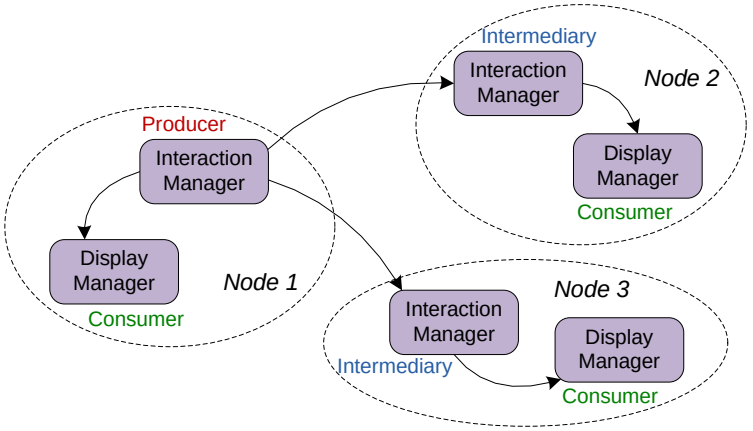
Experiment code

```
service code      →   public class MediaService extends ServiceAgent {
                          …
(a) initialization  →     void init() {
                            recv = startup("receive", "video");
                            disp = startup("display", "video");
                            // event registration
                             register(RECEIVE, receive)
                          }

(b) main            →     void receive() {
    procedure               // request the receiving tool to decode video
                             request(recv, …);
                            // request the display tool to show video
                            request(disp, …);
                          }

(c) shutdown        →     void shutdown() {
                            shutdown(recv);
                            shutdown(disp);
                          }
                      }
```

Describing Service Composition (cont')

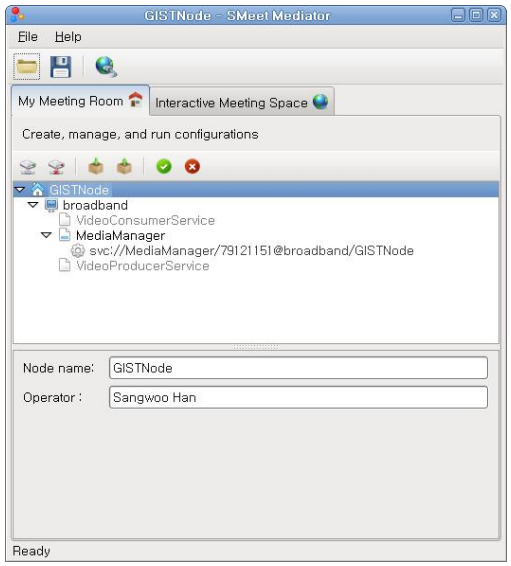
- Functional dependency relation description (using directed acyclic graph)

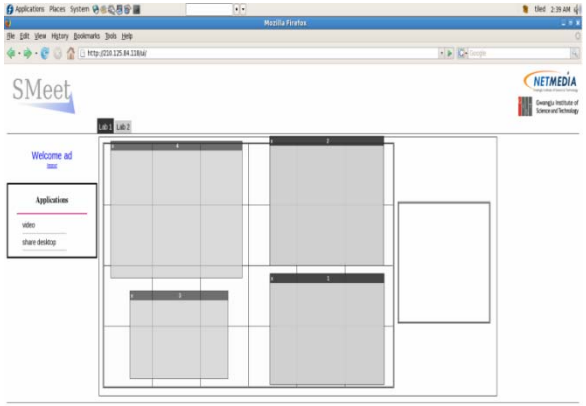
```
<task>
<service managers>
<manager name="mgr1" type="interaction manager"
role="producer" mode="pointer-location-mode">
<usercontext> to be determined
    </usercontext></manager>
<manager name="mgr2" type="interaction manager"
role="intermediary" mode="pointer-location-mode">
<usercontext> to be determined
    </usercontext></manager>
<manager name="mgr3" type="display manager"
role="consumer" mode="tiled-display-mode">
<usercontext> to be determined
    </usercontext></manager>
</service managers>
<composition pattern >
<adjacency src="mgr1"> <dest name="mgr2"/>
    </adjacency>
<adjacency src="mgr2"> <dest name="mgr1"/>
    </adjacency>
<adjacency src="mgr2"> <dest name="mgr3">
    </adjacency>
</composition pattern>
</task>
```



Operation and Control

- Operator Interface
  - Service discovery
  - Dtatus monitoring,
  - Composite service selection
- User Interface
  - Composite service operation





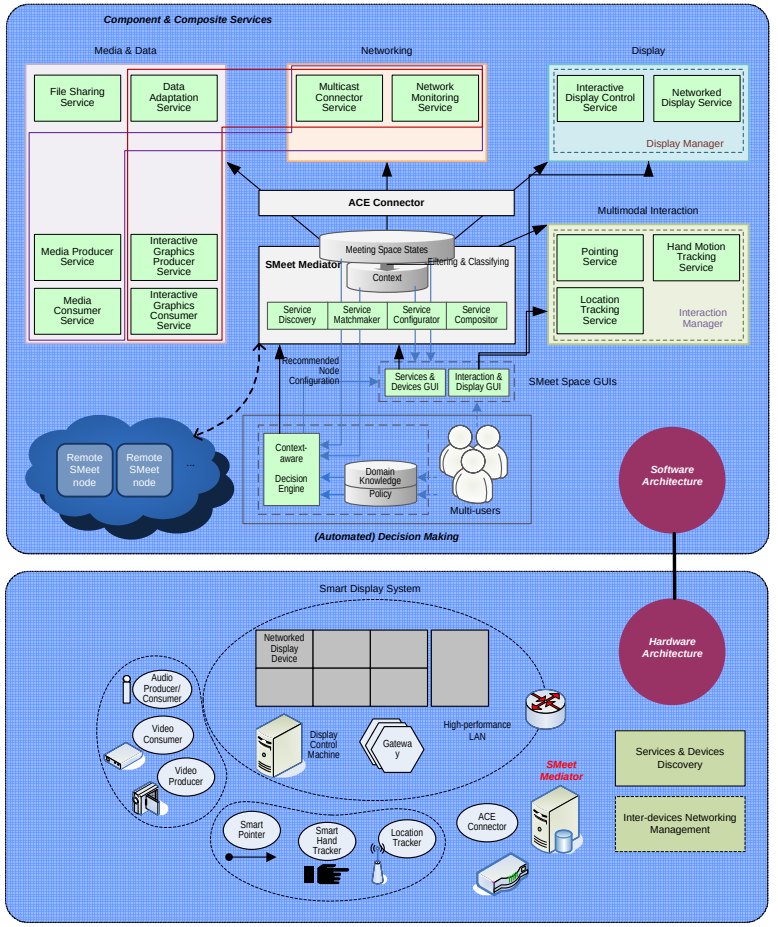
Implementation Status Review

- Version Information
  - MultiAgent-based Management Toolkit version 0.6
  - Source line of code: 13,000 lines

- Development Environment
  - OS: Windows XP/VISTA, Linux
  - Language and IDE: Java (J2SE 5.0)
  - Selected S/W Libraries:
    - Multi-Agent System: JADE 3.6 (Java Agent DEvelopment Framework)
    - GUI: SwingLabs 9.3
    - Data Structure: JSDL 2.1.1
  - External Tools:
    - Ontology Development: Protégé 3.3.1
    - Media Tools: RAT, VIC, VLC, SMOD, and Skype

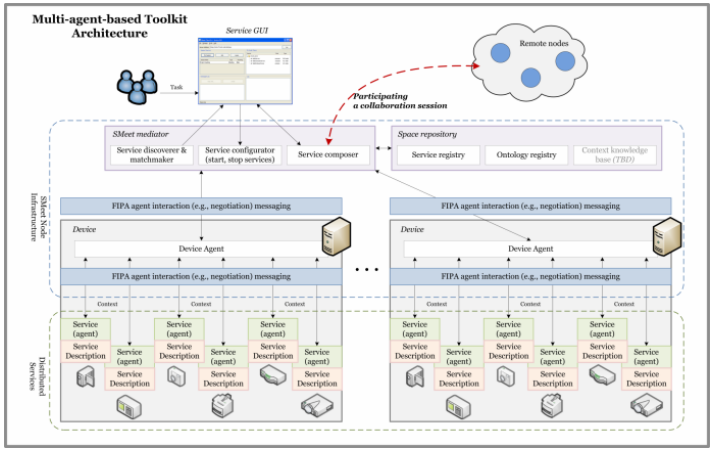
Prior Work (2006-2008): Smart Meeting Space (SMeet)

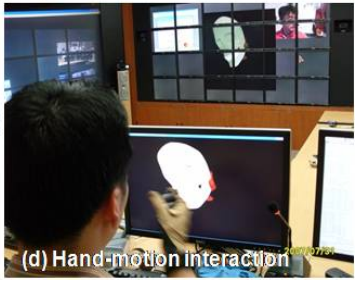
- Goal
  - Easy node operation in order to enable the customization of distributed services and devices
- Approaches
  - Component services for flexibility
  - Context-based configuration of meeting rooms



Smart Meeting Space (cont')

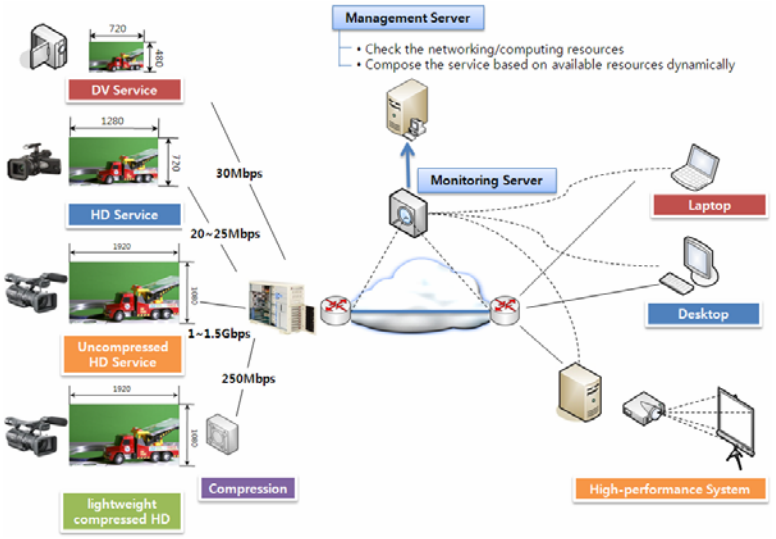
Service	Description
Media devices	• Two HD and four PTZ cameras • Microphones, speakers, and an echo canceller
Tiled display	• LCD (22 inch) 6 by 4 arrays supporting 9600×6400 resolution • A workstation for display control and 12 workstations for display service • Two workstations for display interaction and GUI operation
Display pointer	• Pointing errors under 40 pixels • IR laser pointer and camera (VGA, 29 fps) with IR optical-filter
Hand-motion tracker	• Response time under 0.5 sec • Two virtual gloves having multiple acceleration and gyro sensors
Localization sensors	• Accuracy: less than 0.8 m • A localization server and six ultra-sound and RF signal based localization sensor nodes
Networking	• 10 Gbps multicast-enabled network





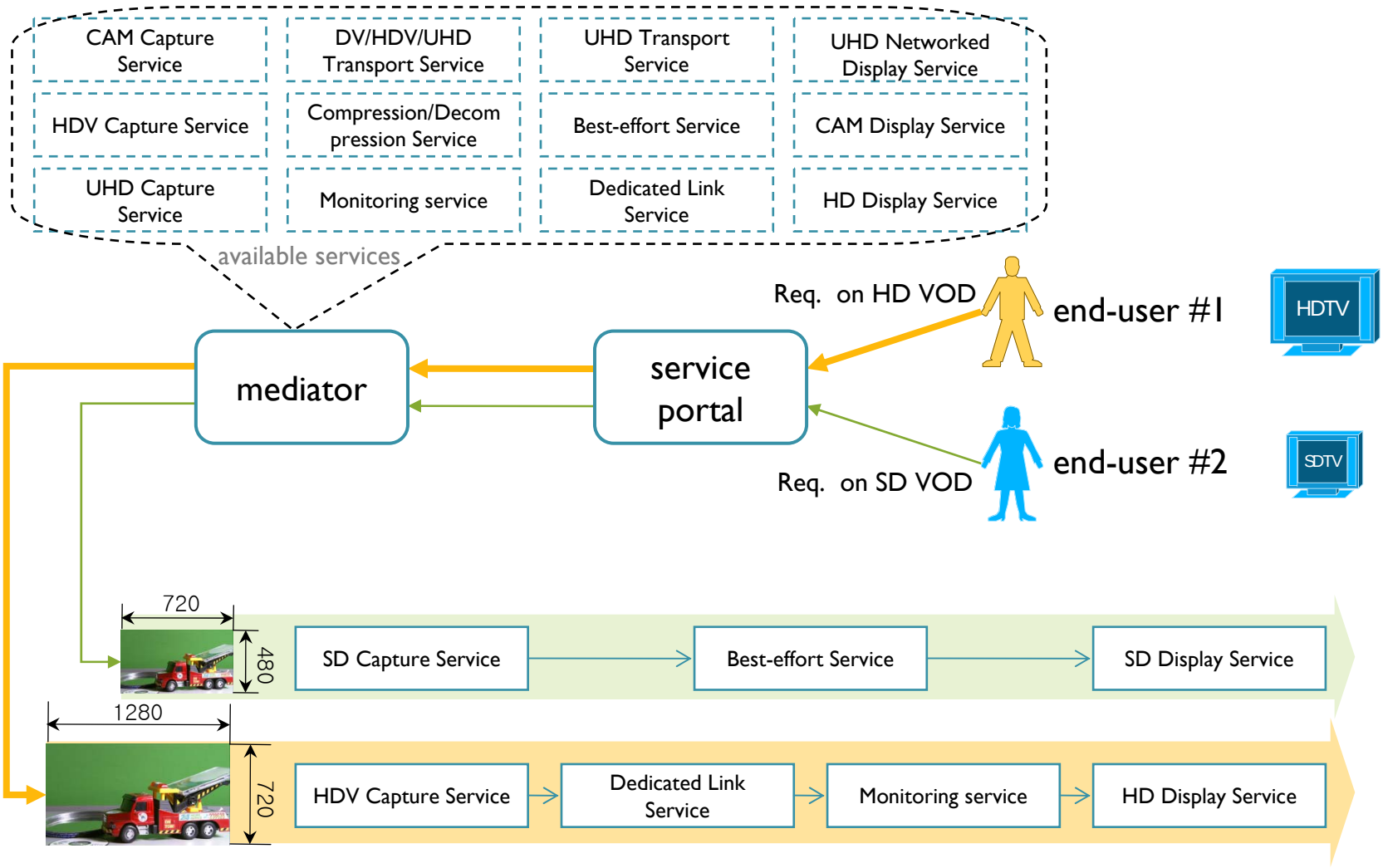
Ongoing Work (2008~Current) :
Preliminary State of Service Composition

- Goal
  - Pre-defined multimedia service composition of diverse computing/networking resources
- Approaches
  - programmatic method based service composition
  - VOD streaming combining video producer, RTP monitoring, and video conmer services

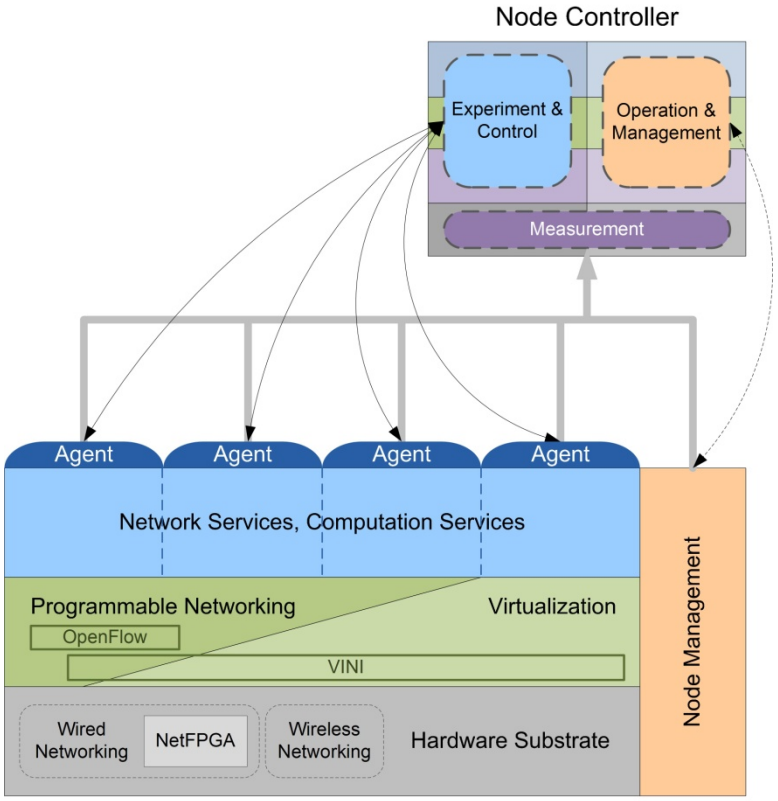


Future Internet Exhibition@SNU

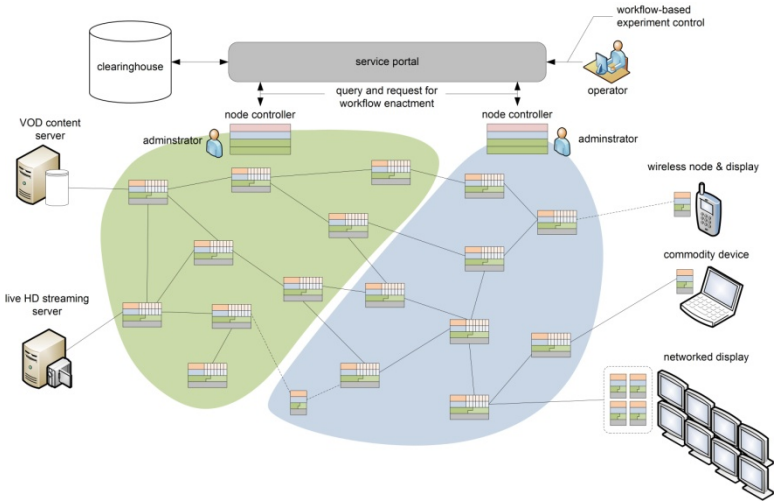
Preliminary State of Service Composition (cont')



Future Work (2009~): Agent-based Service Operation, Composition and Control over Virtualized Infrastructure



Programmable Computing/Networking **Node**



Summary and Future Work

- Summery
  - Abstractions of nodes, devices, services
  - Progammatic methods for service composition
  - Implementation of preliminary service composition using JADE MAS middleware
- Toward Service Composition over Dynamic Virtualized Slices
  - Integrating MAS-based software with virtualized networking/computing testbed
  - Interworking of two main behaviors that includes virtualization control and service composition
  - Service control following the workflows of user experiments