

Mobile Agents

Diwakar Yagyasen
Asst. Prof. Deptt of
CSE, BBDNITM,
Lucknow, UP, India
dylucknow@gmail.com

Outline

- **Overview of Agents and Mobile Agents**
- Characterization of Mobility
- Advantages of Mobile Agents
- System Components
- Language / Design Issues
- Applications
- Challenges
- Survey of Mobile Agent Systems
- Summary

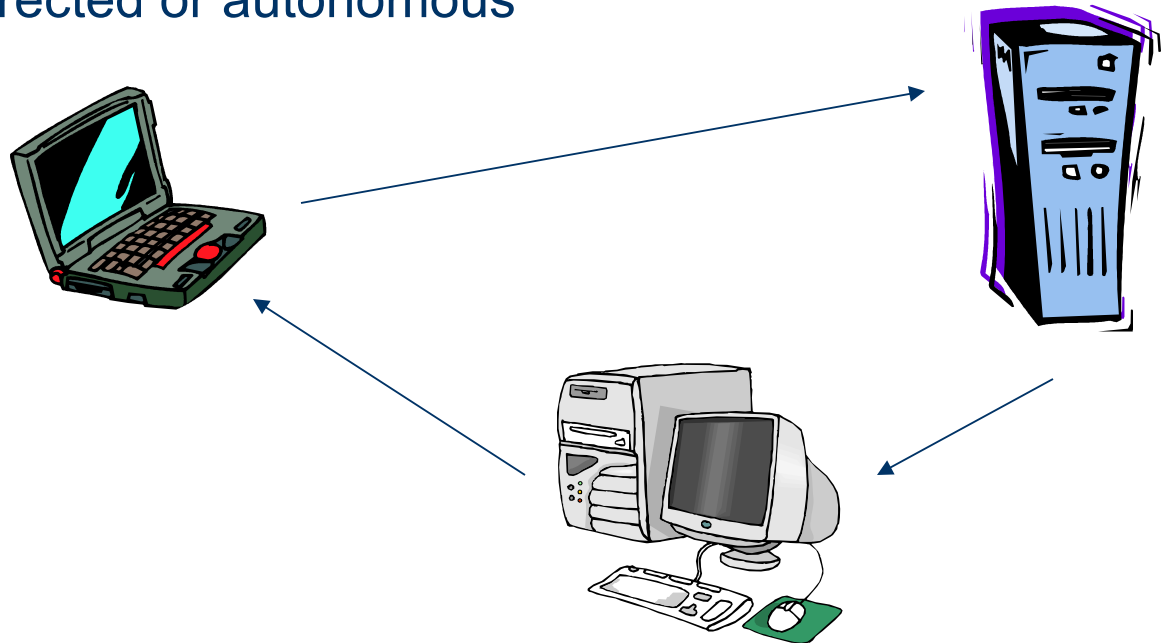
Definition of Software Agents

- Computer program
- Autonomous behavior
- Represents some entity
- Has authority (delegation)
- Reacts and learns about environment
- Communicates using high-level Agent Communication Languages (ACLs)

Mobile Agents

Definition

- Software agents
- Move from one computer to another
 - User-directed or autonomous

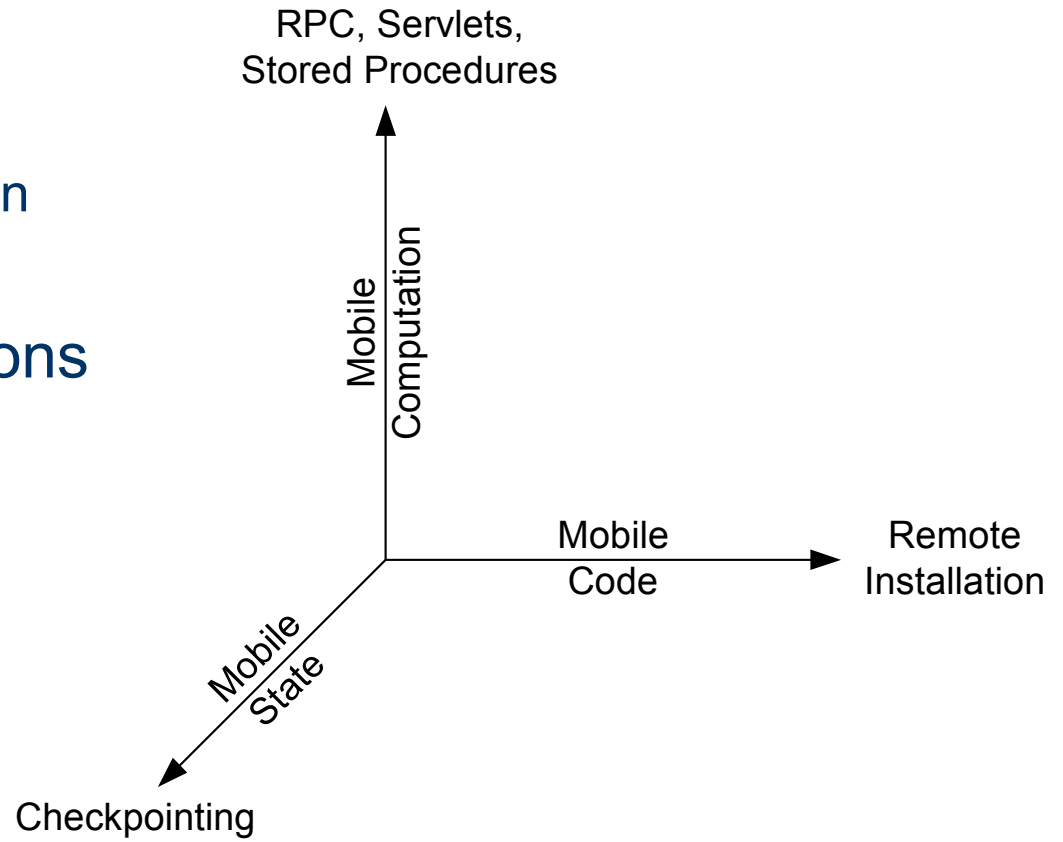


Outline

- Overview of Agents and Mobile Agents
- **Characterization of Mobility**
- Advantages of Mobile Agents
- System Components
- Language / Design Issues
- Applications
- Challenges
- Survey of Mobile Agent Systems
- Summary

Characterizing Mobility

- Three capabilities
 - Mobile code
 - Mobile computation
 - Mobile state
- Various combinations are possible



Mobile Code

- Allows executable code to be moved to a new host
- May use the push or pull model
 - Pull: Applets
 - Push: Remote Installation
- Mobile agents use push
 - Sometimes, an agent push may result in a code pull
- Code may be binary (intermediate or native) or source

Mobile Code

- Advantages:
 - Dynamically change capabilities
 - Download new code to add / change / update capabilities of platform
 - Remove code when no longer needed
- Problems:
 - Security concerns due to untrusted / unchecked code
 - Code could be malicious, buggy, and/or tampered

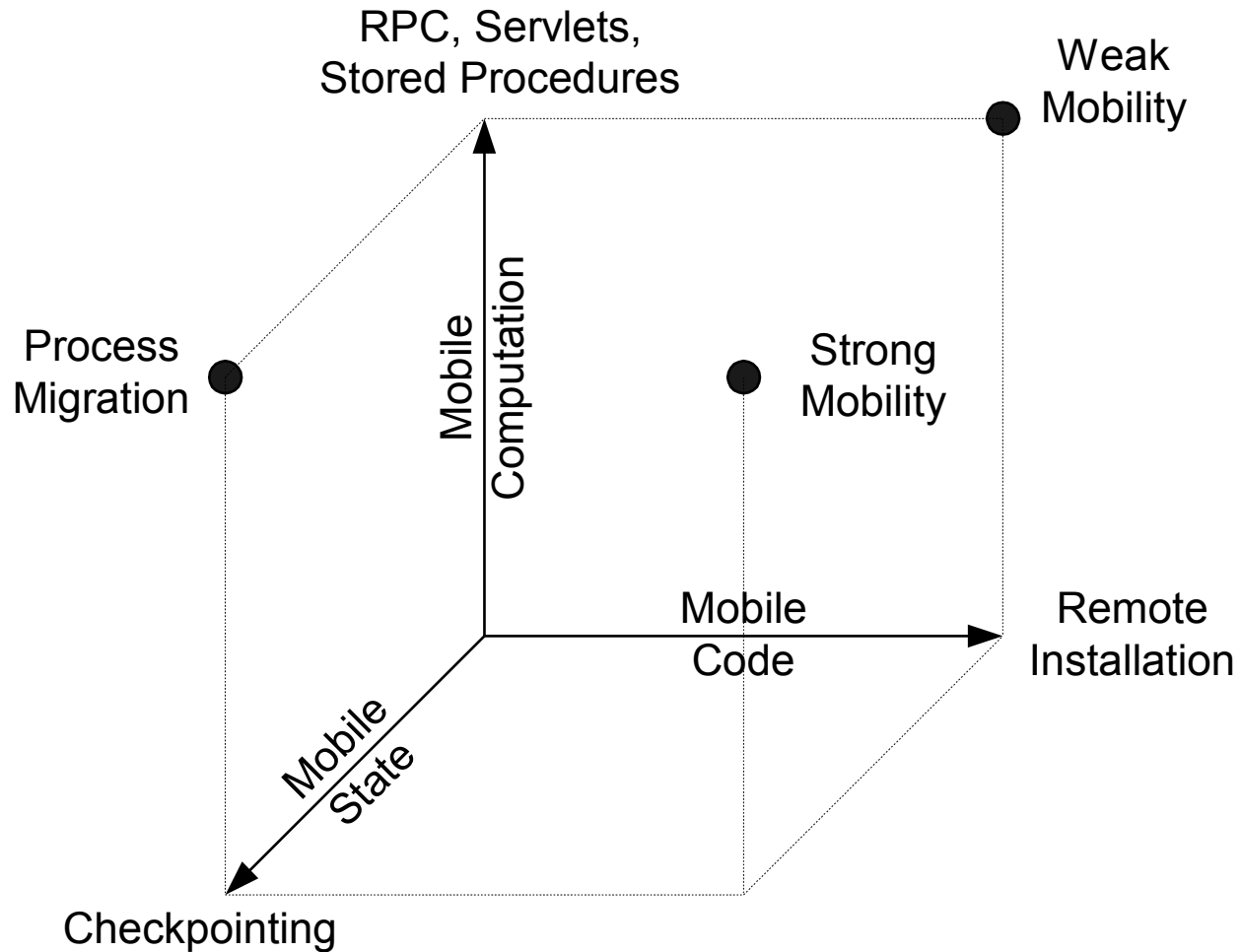
Mobile Computation

- Evolution of Remote Computation
 - RPC, RSH, RMI, Servlets, Stored Procedures, CORBA
- Allows one system to run a computation on another system
- Utilize resources on remote system
 - CPU, memory
- Access resources on remote system
 - Files, databases, etc.

Mobile State

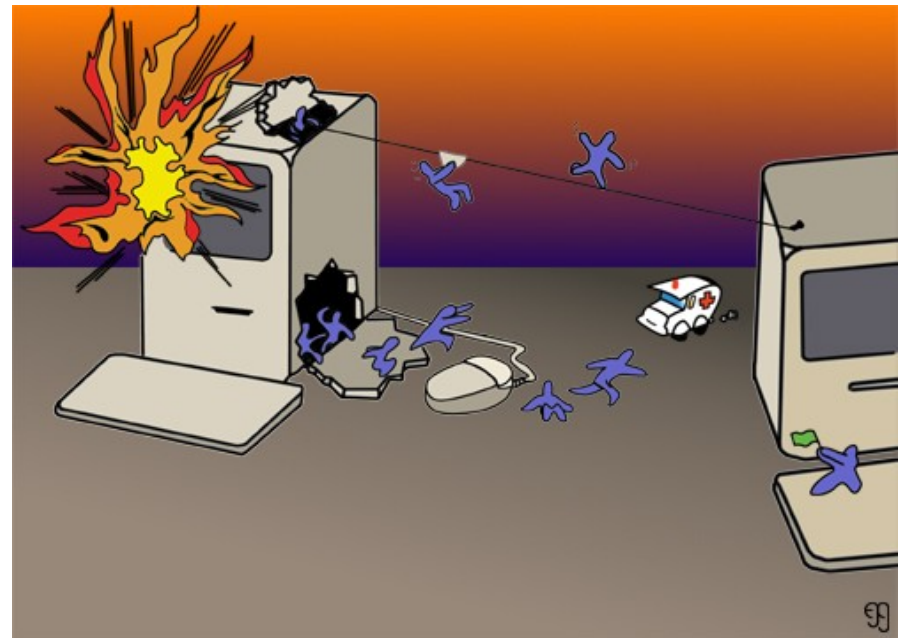
- Evolution of State Capture
 - Checkpointing
- Allows execution state of a process to be captured and moved
- State may be machine specific or machine independent
- May contain
 - State of single or multiple threads
 - Code

Combinations of Capabilities



Strong Mobility

- Move execution state with agent
- Why is it important?
 - Computationally equivalent to weak mobility
 - However, simpler, more natural abstraction
 - Therefore, easier to write mobile agents
- More importantly –
mobile state allows
forced mobility



Weak Mobility Example

```
public class Example extends Aglet {
    boolean _theRemote = false;
    public void onCreate (Object init) {
        addMobilityListener(
            new MobilityAdapter() {
                public void onArrival (MobilityEvent e) {
                    _theRemote = true;
                }
            }
        );
    }
    public void run() {
        if (!_theRemote) {
            System.out.println ("On Source");
            dispatch(destination);
        }
        else {
            System.out.println ("On Destination");
        }
    }
}
```

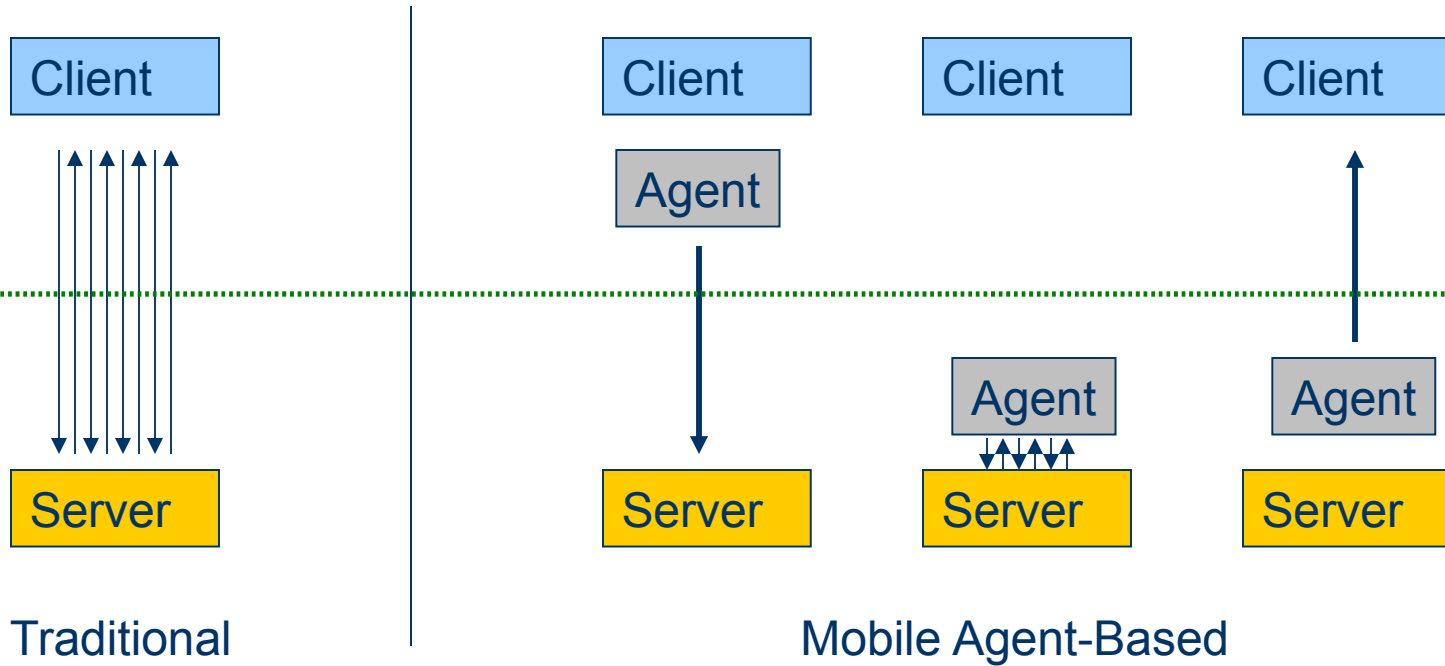
Strong Mobility Example

```
public class Example extends Agent
{
    public static void main (String[] args)
    {
        System.out.println ("On source");
        go (destination);
        System.out.println ("On destination");
    }
}
```

Outline

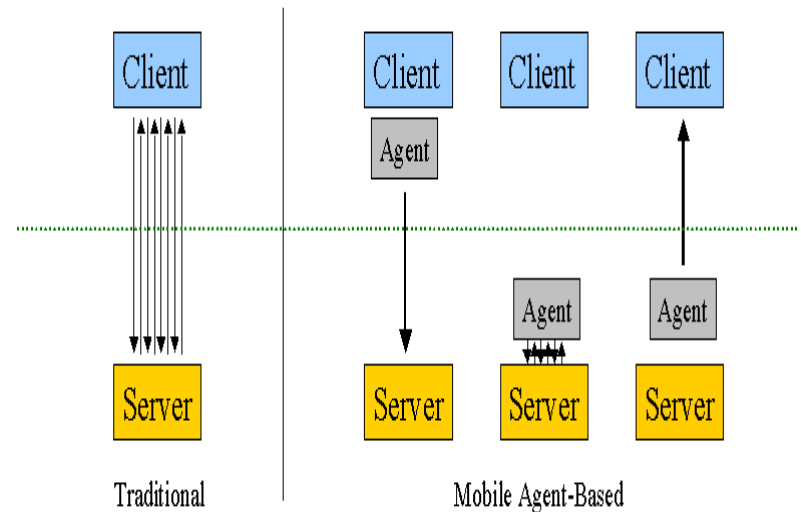
- Overview of Agents and Mobile Agents
- Characterization of Mobility
- **Advantages of Mobile Agents**
- System Components
- Language / Design Issues
- Applications
- Challenges
- Survey of Mobile Agent Systems
- Summary

Client-Server Versus Mobile Agents

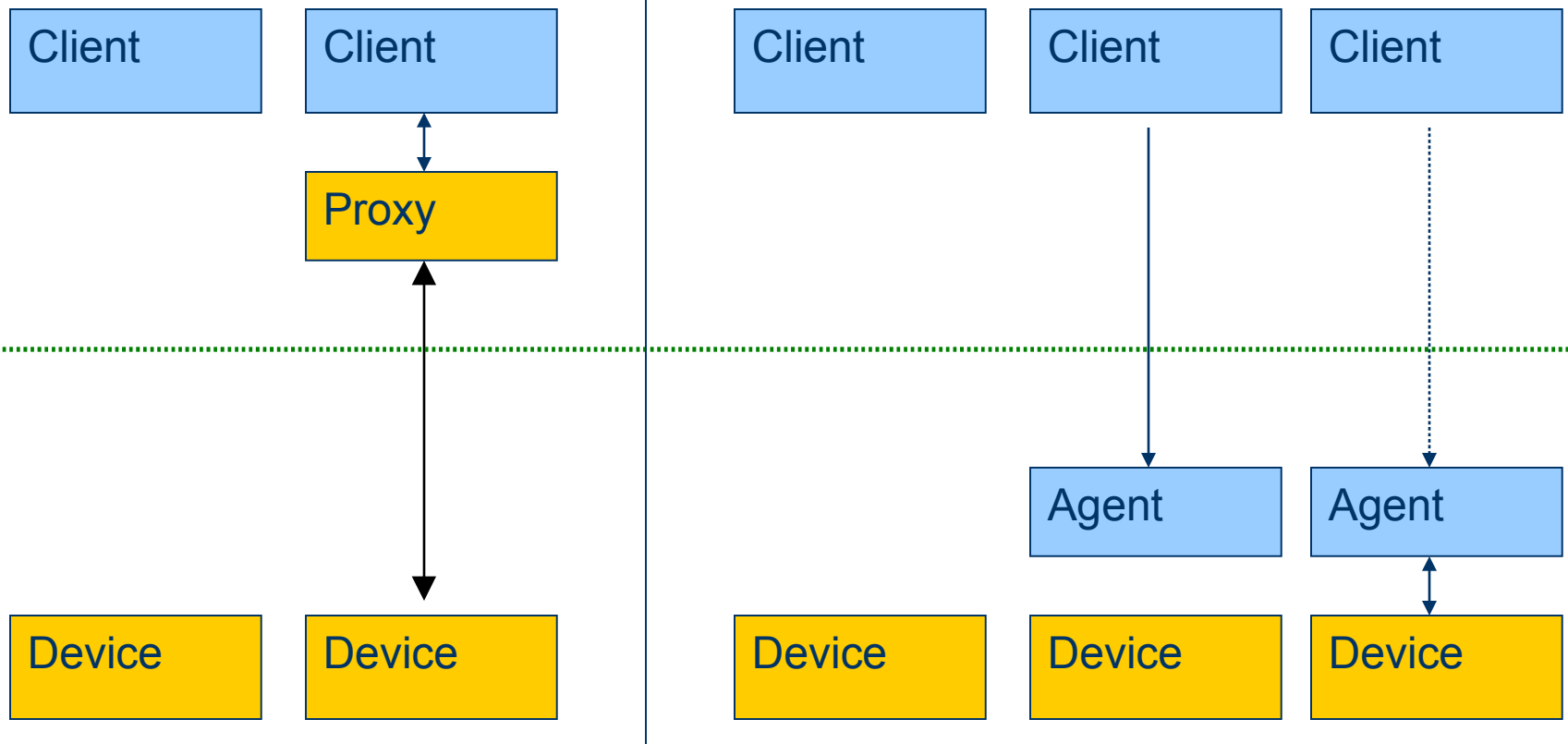


Advantages of Mobile Agents

- Reduced network bandwidth
- Disconnected operation
 - Short “On-Line” times
 - Low-power requirements
 - Support for mobile units
- Low-latency interaction



Jini Versus Mobile Agents



Advantages of Mobile Agents

- Highly Dynamic and Flexible Systems
 - Enabled my mobile code
 - Download new capabilities / services
 - Remove old / unused capabilities
 - “Swapping” for memory constrained devices
 - Structure systems around mobile code
 - “Universal” server
 - Open Services Gateway Initiative (OSGI)

Advantages of Mobile Agents

- Unique capability:
 - Send an executable program that does your bidding on someone else's computer
- Very powerful but... can be dangerous!
 - Only one step removed from a Virus

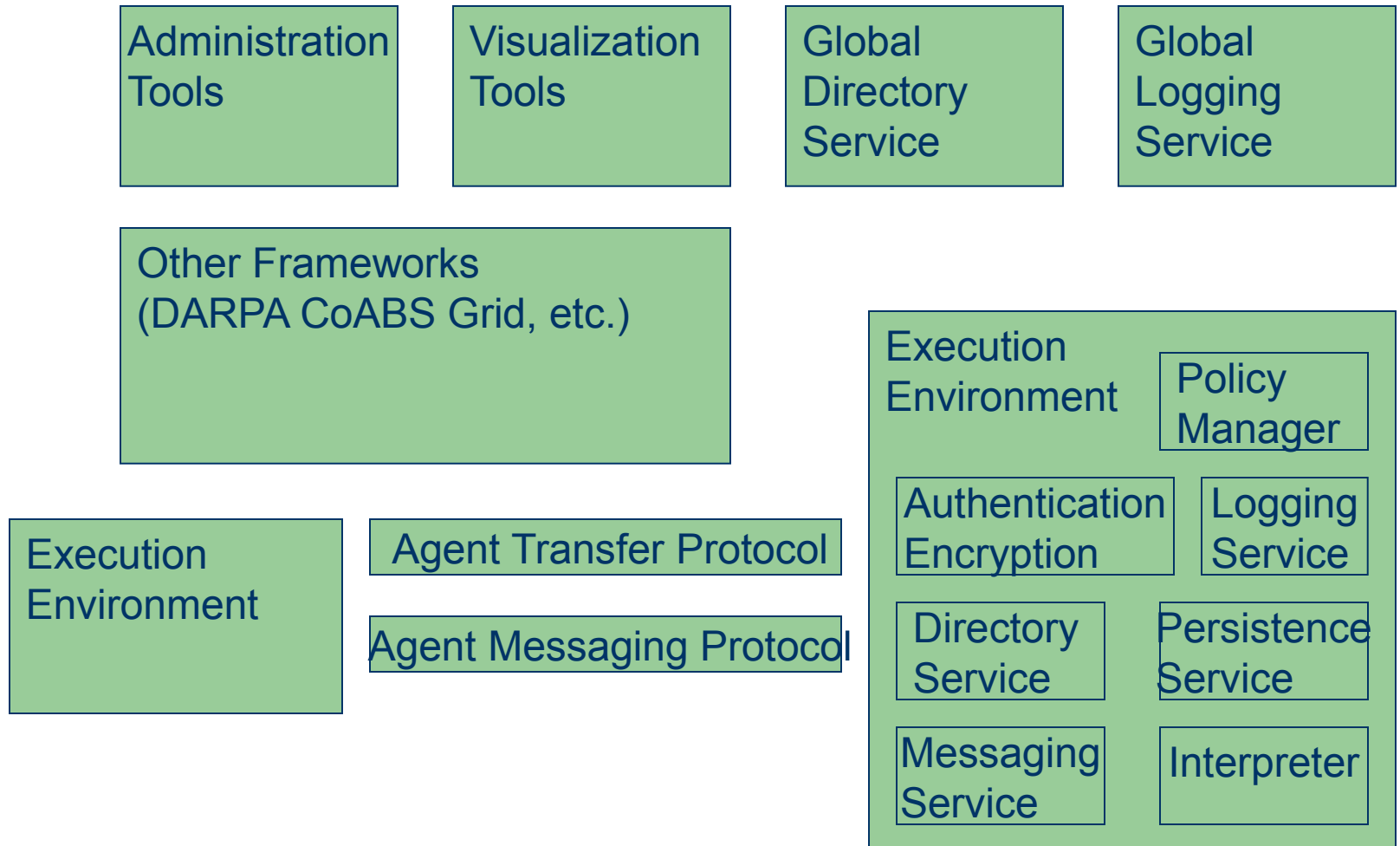
Agent Programming Paradigms

- Itinerant Agents
 - Use an itinerary that dictates mobility of agents
- Reactive Agents
 - Event-based approach
 - Events trigger mobility of agents
- Agent-Minion
 - Minions: Small mobile agents spawned by larger agents

Outline

- Overview of Agents and Mobile Agents
- Characterization of Mobility
- Advantages of Mobile Agents
- **System Components**
- Language / Design Issues
- Challenges
- Survey of Mobile Agent Systems
- Summary

System Components



Outline

- Overview of Agents and Mobile Agents
- Characterization of Mobility
- Advantages of Mobile Agents
- System Components
- **Language / Design Issues**
- Applications
- Challenges
- Survey of Mobile Agent Systems
- Summary

Languages for Mobile Agents

- Java
 - Numerous Systems
- TCL
 - D'Agents, SMIA
- C/C++
 - Omniware
- Miscellaneous
 - Telescript
 - Lisp, Scheme, custom, etc.

Advantages of Java

- Platform independent
- Virtual Machine execution environment
 - Important for isolation
- Small footprint
- “Real” programming language
 - Compared to TCL
- Most security conscious
 - Though still not adequate!

Limitations of Java

- No state capture
- Importance:
 - Mobile state
 - Mobile agent systems
 - Load balancing (distributed systems)
 - Forced migration
 - Cloning (fault tolerance)
 - Checkpointing
 - Faster VM startup
 - Restarting crashed applications
 - Persisting processes for later resumption

Limitations of Java

- No resource control
- Importance:
 - Protect against denial-of-service attacks
 - Malicious code
 - Buggy code
 - Prioritize tasks
 - Foundation for providing QoS guarantees

Limitations of Java

- No resource accounting
- Importance:
 - Measuring resource consumption
 - Charging / billing resource usage
 - Observing behavior of code

Directory Service Issues

- Difficult to find an agent
 - Chase agent around
- Fast Moving Agents
 - May be difficult/inefficient to update directory service

Authentication Issues

- Agent Anonymity
 - Agent may want to be anonymous on a host
 - Analogy: Window shopping
- Multiple Hop
 - Agent may not want to carry credentials
 - Credentials could be stolen by malicious hosts

Outline

- Overview of Agents and Mobile Agents
- Characterization of Mobility
- Advantages of Mobile Agents
- System Components
- Language / Design Issues
- **Applications**
- Challenges
- Survey of Mobile Agent Systems
- Summary

Applications of Mobile Agents

Information Retrieval

- Mobile agents reduce network bandwidth
- Depends on:
 - Quantity of information searched
 - Quantity of information retrieved
 - Size of mobile agent
- Comprehensive study for DARPA
 - Dartmouth College, Lockheed Martin ATL, University of West Florida

Applications of Mobile Agents

- Monitoring
 - Computer programs can be very patient ù
- Remote Control
- Dynamic Systems
 - Universal servers
- Active Mail
 - Send executable content as email

Outline

- Overview of Agents and Mobile Agents
- Characterization of Mobility
- Advantages of Mobile Agents
- System Components
- Language / Design Issues
- Applications
- **Challenges**
- Survey of Mobile Agent Systems
- Summary

Challenges for Mobile Agents

- Security Issues
 - Protecting network communication
 - Protecting hosts from agents
 - Illegal access
 - Denial of service
 - Protecting agents from hosts
 - Tampering
 - Extracting information
 - Capture / Replay
- System-wide Administration / Management
 - Policies
 - Tracking / Visualization

Challenges for Mobile Agents

- Access to non-mobile resources
 - Network endpoints
 - Files
- Deployment (of environments)
- Interoperability
 - OMG MASIF – Not successful
 - DARPA CoABS – We shall see...
- Debugging
 - Highly Asynchronous

Outline

- Overview of Agents and Mobile Agents
- Characterization of Mobility
- Advantages of Mobile Agents
- System Components
- Language / Design Issues
- Applications
- Challenges
- **Survey of Mobile Agent Systems**
- Summary

Survey of Mobile Agent Systems

Commercial Systems

- Telescript/Odyssey - General Magic
- Voyager - ObjectSpace
- Aglets - IBM
- Concordia - Mitsubishi Electric ITA
- Jumping Beans - AdAstra

Survey of Mobile Agent Systems

Research Systems

- NOMADS
- KAoS
- D'Agents
- Agents for Remote Action (ARA)
- Mole
- Sumatra
- Many others...

Mobile Agent List:

<http://www.informatik.uni-stuttgart.de/ipvr/vs/projekte/mole/mal/mal.html>

NOMADS

Java-based mobile agent system

- Strong mobility
 - Capture full execution state of running agents
 - Provides **anytime** mobility – simplifies writing mobile agents
 - Provides **forced** mobility – arbitrary Java code can be moved
- Strong security
 - Dynamically control resource usage (rates and quantities)
 - Allows platform owner full control over agent execution
 - Protect against denial of service attacks

Strong Security

- Limitations of current Java systems
 - Rely on JDK security
 - Does not provide resource control
 - Either assume agents are safe
 - Does not scale
 - Or rely on code signing
 - Not a preventive measure
- Goal: Build secure execution environments
 - Run **untrusted** or **partially-trusted** code

Aroma Virtual Machine

- Clean-room implementation
- State capture mechanism
- Dynamic, fine-grained resource control
 - Disk, Network, CPU
- JDK 1.2.2 compatible
 - Uses Java Platform API from JRE 1.2.2
 - No AWT / Swing
- Ported to Win32 (x86), Linux (x86), Solaris (SPARC)
- No Just-In-Time compilation
- (Almost) No optimization

State Capture

- Aroma supports two modes
 - Full VM state
 - All threads, loaded classes, objects
 - State may be restored into a “blank” VM
 - Size: Approx 1.5 MB
 - Individual thread state
 - Method stack and all reachable objects
 - Thread may be restored into running VM
 - Size: Approx 4 KB

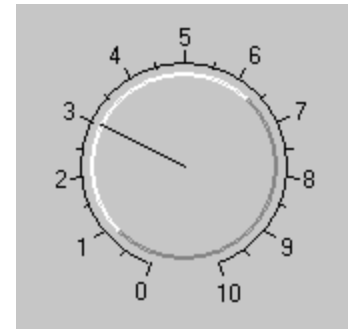
State Capture

- State capture is fine-grained
 - Between any two Java bytecode instructions
 - Supports blocked, waiting, sleeping, and suspended threads
- State is platform independent
- State may be stored in memory, saved to disk, or streamed over the network

Resource Control

Rate control

- Control rate at which resources are used
- Dynamically adjustable
- Fine grained
- Examples:
 - CPU limited to 10%
 - Disk write rate limited to 30 KB/sec
 - Network read rate limited to 10 KB/sec

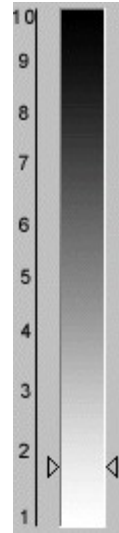


Disk Usage Rate

Resource Control

Quantity control

- Control quantity of resources used
- Dynamically adjustable
- Fine grained
- Examples:
 - Disk space limited to 1.8 MB
 - Total network writes limited to 1024 KB
- Still needed: memory



Disk Usage Quantity

Benefits of Resource Control

- Protect host from malicious agents
 - Prevent denial of service attacks
- Simplify agent writer's task
 - Agents do not need to worry about resource control / limits
- Means of prioritization
 - Raising limits increases priority
- Basis for Quality of Service
- Means of accounting

Outline

- Overview of Agents and Mobile Agents
- Characterization of Mobility
- Advantages of Mobile Agents
- System Components
- Language / Design Issues
- Applications
- Challenges
- Survey of Mobile Agent Systems
- **Summary**

Summary

- Mobile Agents are Good
 - Code mobility adds significant flexibility
- Security is Critical
- Deployment is Difficult
- No Interoperability

MH

MP

MPD