Mobile Agents

Course: Theory and Practice of Software Engineering

Lecture 11

Lecturer: Alexander Vazhenin

E-mail: vazhenin@u-aizu.ac.jp

Topics

- Basic Architecture
- Attributes
- Benefits
- Mobile-agent applications

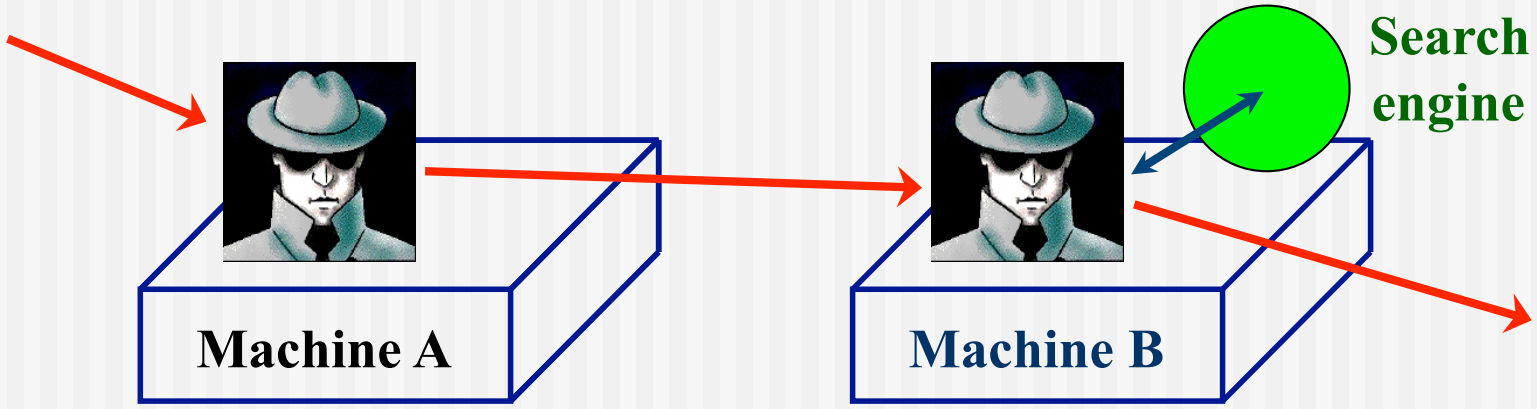
Introduction

- ***Mobile agent*** is a *distributed computing* paradigm.
- It has become viable, with recent technologies such as those provided by Java.
- It has great potential for network applications.
- It has not been widely deployed.

Is everything an "agent"?

- Not all programs are agents
- Agents are
  - customized
  - persistent
  - autonomous
  - adaptive

What is a mobile agent?

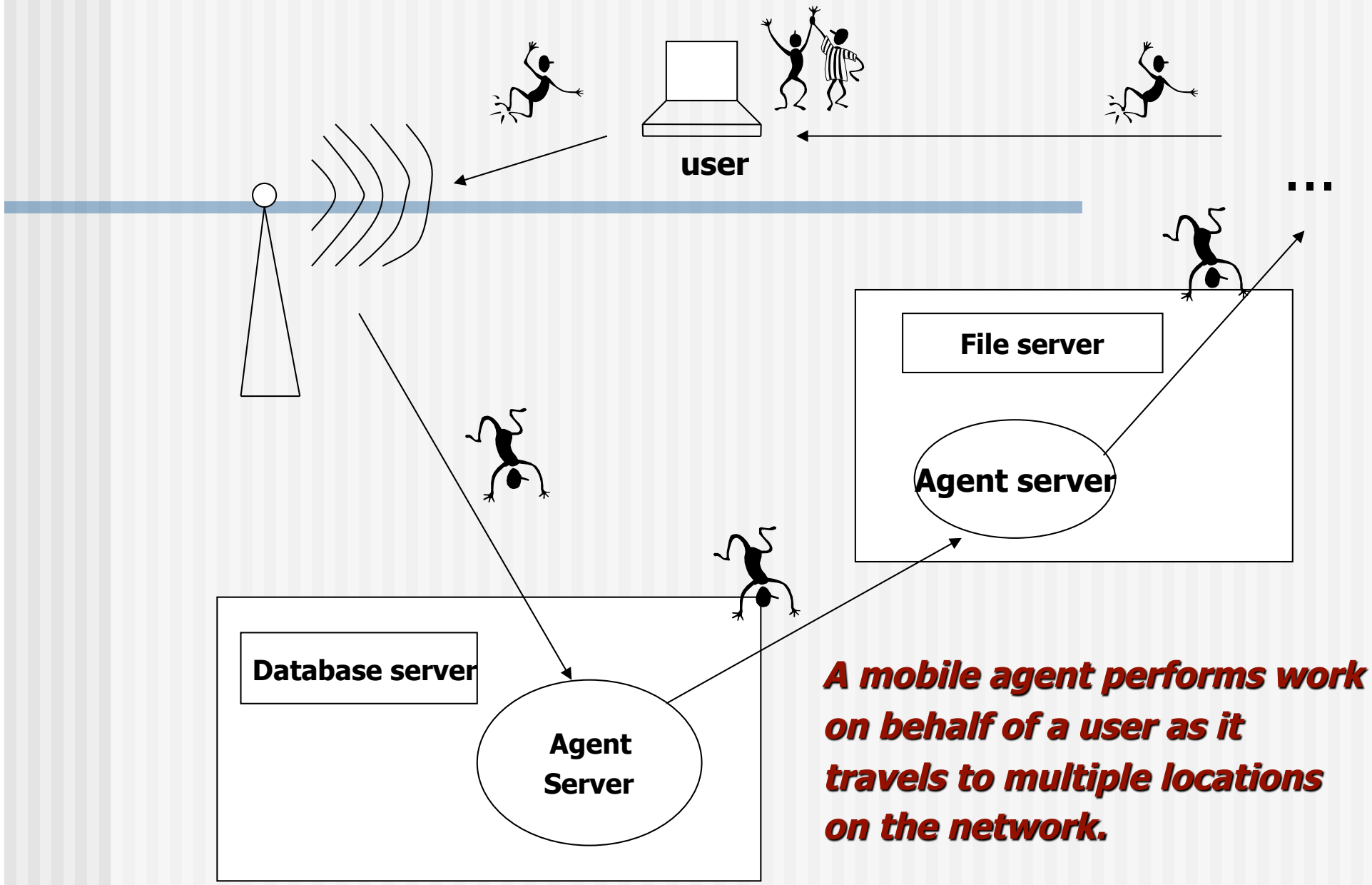


Mobile agent: Agent that

- migrates from machine to machine
- in a heterogeneous network
- at times of its own choosing

Mobile (transportable) agents

- An ***agent*** is "*an independent software program which runs on behalf of a network user*".
- A ***mobile agent*** is a program which, once it is launched by a user, can travel from node to node *autonomously*, and *can continue to function even if the user is disconnected from the network*.

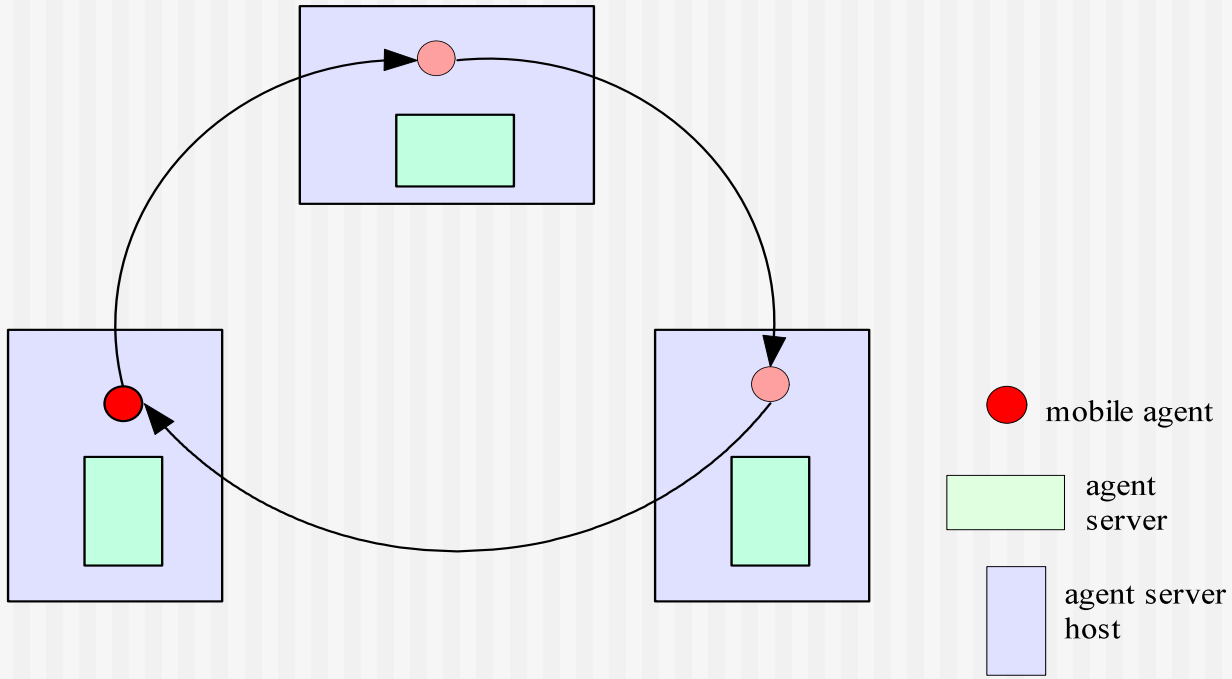


A mobile agent performs work on behalf of a user as it travels to multiple locations on the network.

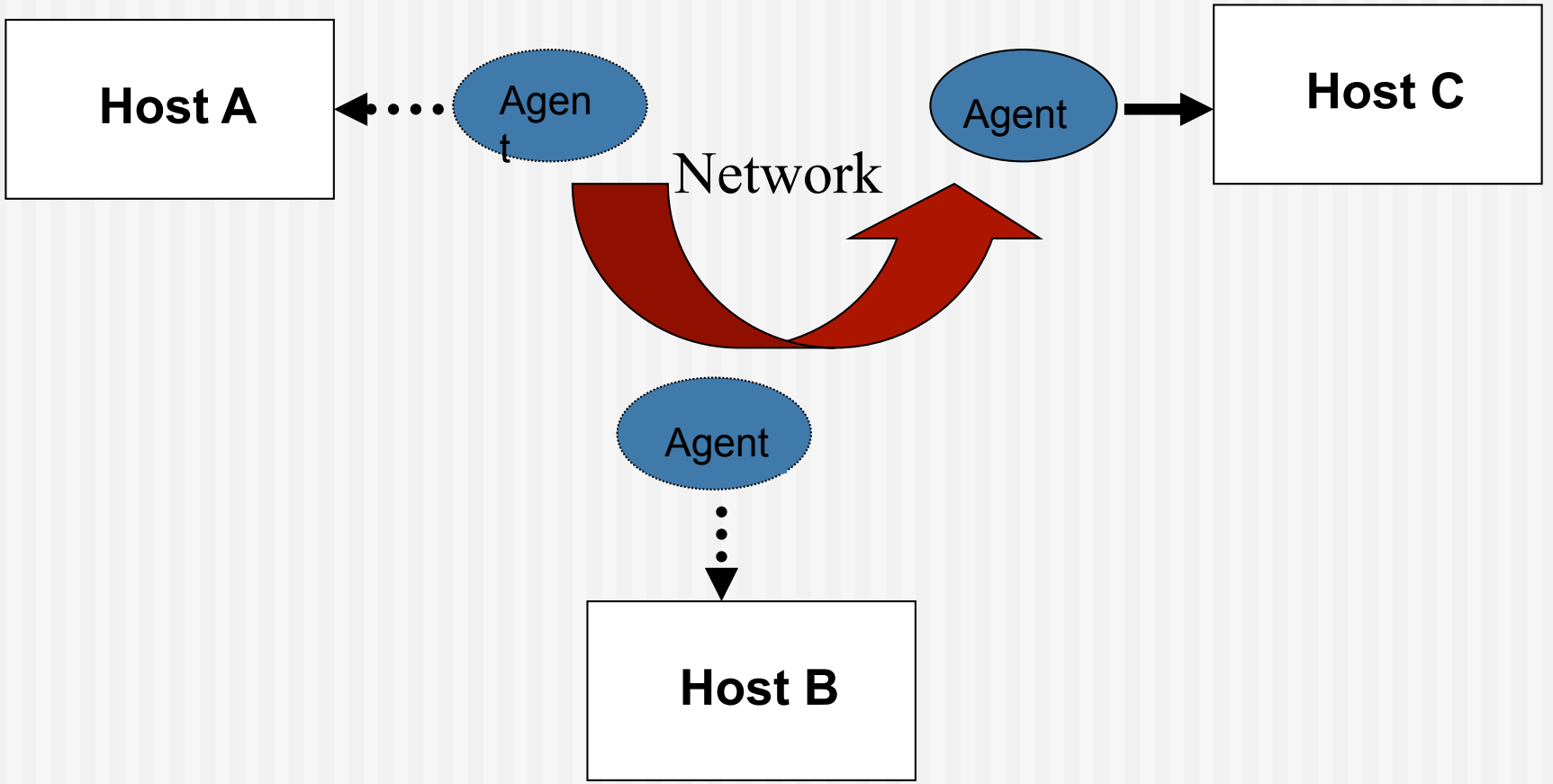
Basic Architecture

- An agent server process runs on each participating host.
- Participating hosts are networked through links that can be low-bandwidth and unreliable.
- An agent is a serializable object whose execution state can be frozen for transportation and reconstituted upon arrival at a remote site.

Basic Architecture



How it works?



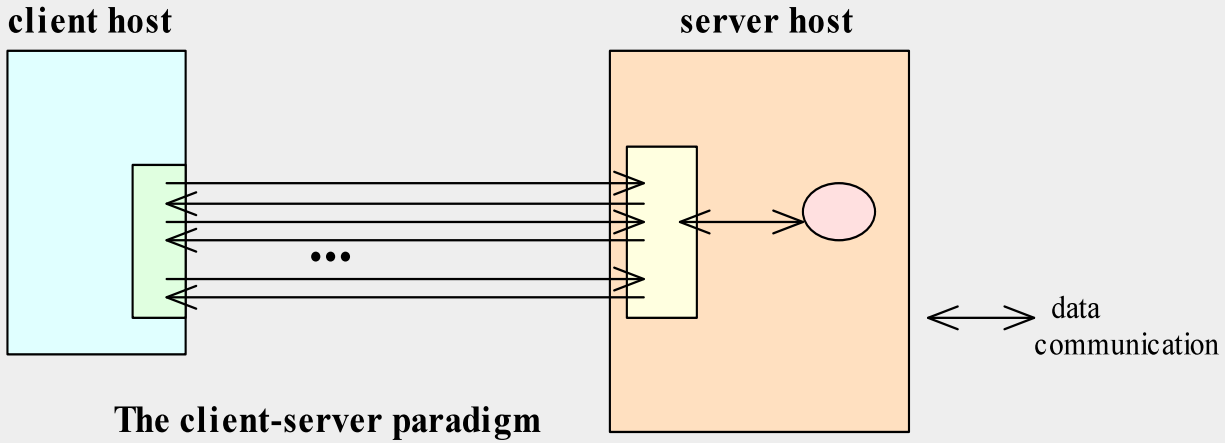
What's in the Agent?

- An agent is an ***object***, hence it contains *state data* and *methods*.
- Among the instance data is an *itinerary* of the sites to be visited, which may be *dynamically constructed* or *adjusted*.
- Other data may include an agent ID or other authentication data.
- The agent's behavior at each stop can be pre-programmed and dynamically adjusted.

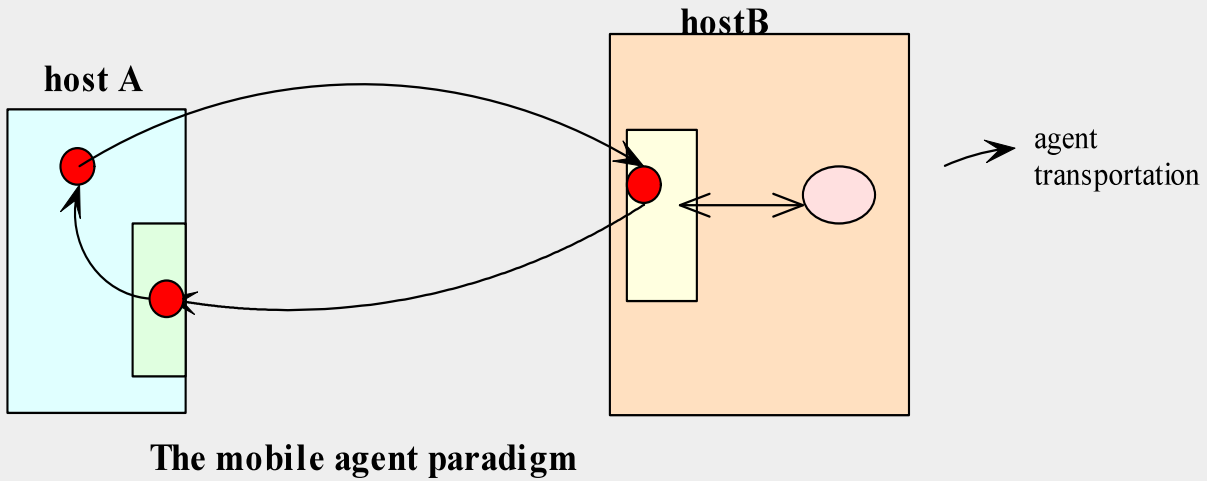
Mobile Agent Attributes

- Code
- State
  - Execution state
  - Object state
- Name
  - Identifier
  - Authority
  - Agent system type
- Location

The mobile agent paradigm vs. the client-server paradigm

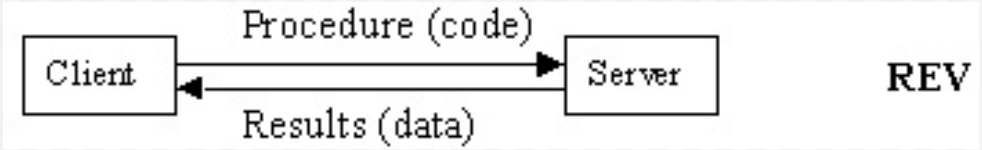


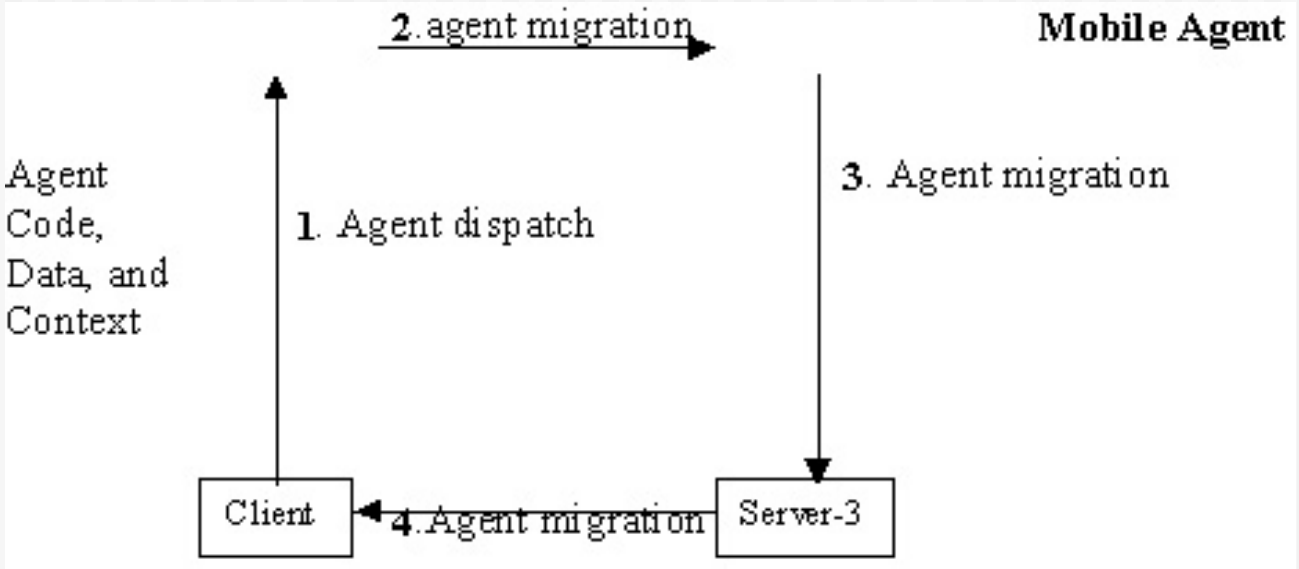
The client-server paradigm



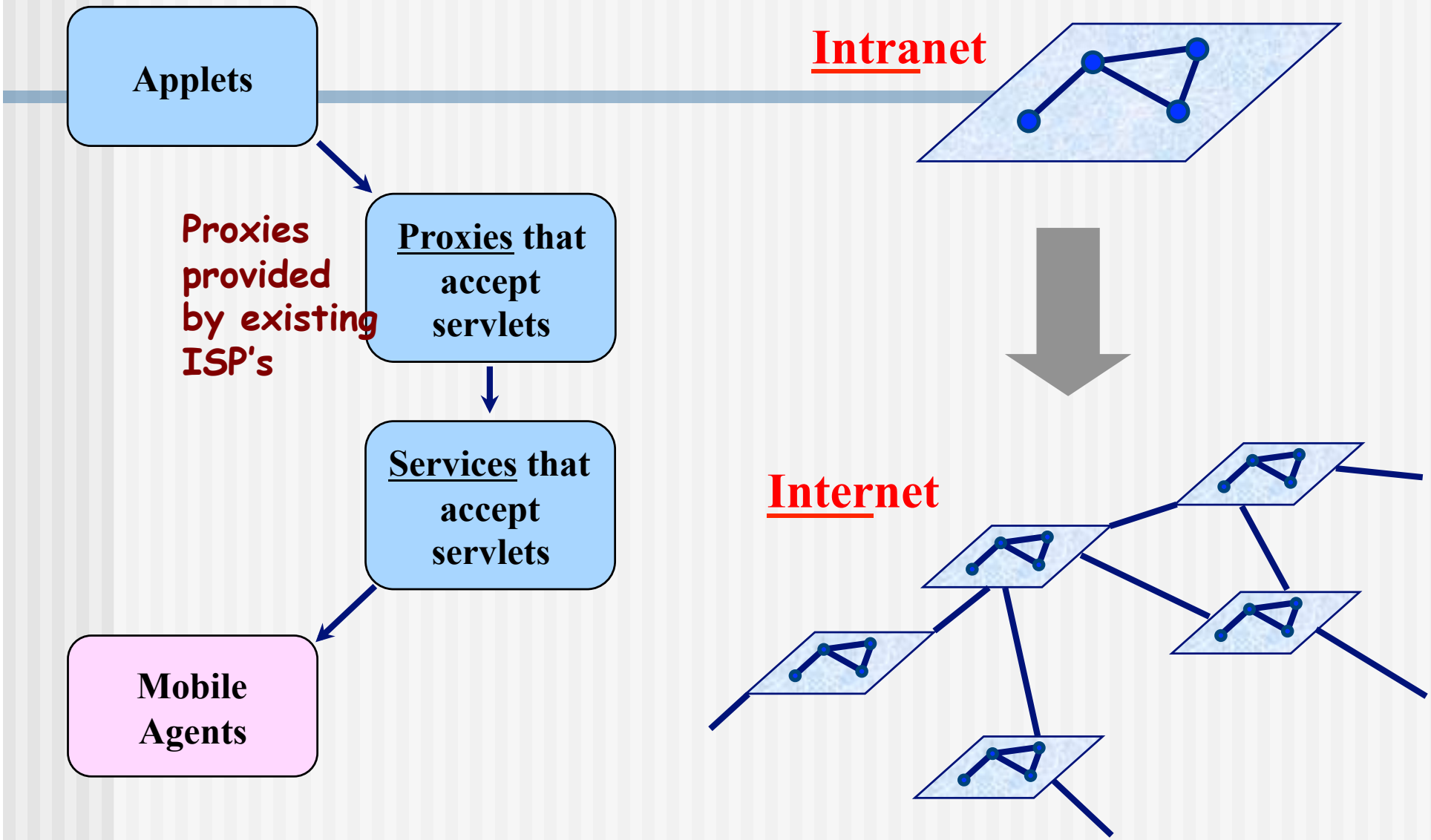
The mobile agent paradigm

Evolution of the "mobile agent" paradigm





Migrating to migrating code



Assumptions about computer systems violated by mobile agents

- Whenever a program attempts some action, we can easily identify a person to whom that action can be attributed, and it is safe to assume that that person intends the action to be taken.
- Only persons that are know to the system can execute programs on the system.
- There is one security domain corresponding to each user; all actions within that domain can be treated the same way.
- Single-user systems require no security.
- Essentially all programs are obtained from easily identifiable and generally trusted sources
- The users of a given piece of software are restrained by law and custom from various actions against the manufacturer's interests

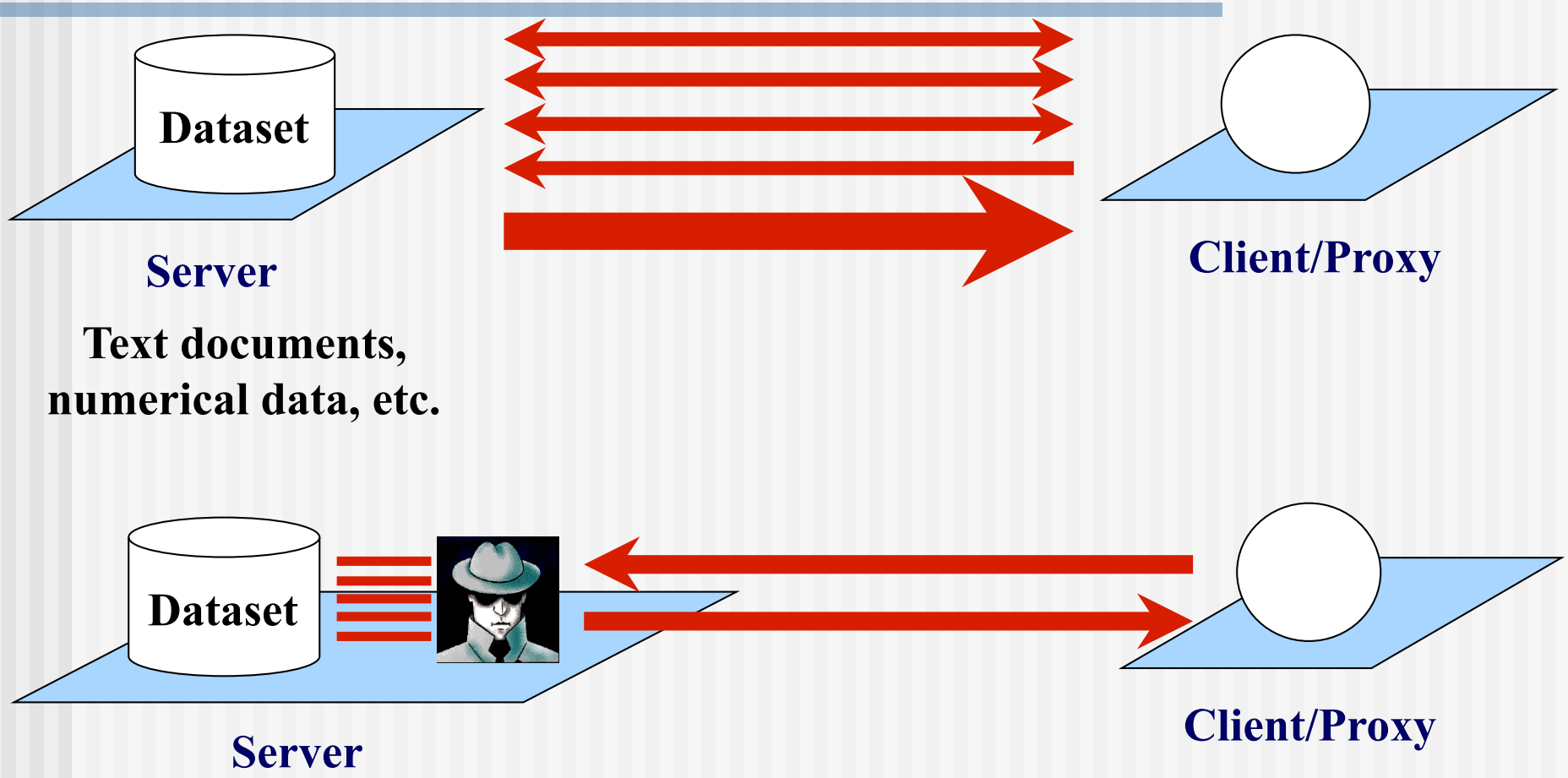
Assumptions about computer systems violated by mobile agents

- Significant security threats come from attackers running programs with the intent of accomplishing unauthorized results.
- Programs cross administrative boundaries only rarely, and only when people intentionally transmit them.
- A given instance of a program runs entirely on one machine; processes do not cross administrative boundaries at all.
- A given program runs on only one particular operating system.
- Computer security is provided by the operating system

Benefits of mobile agents

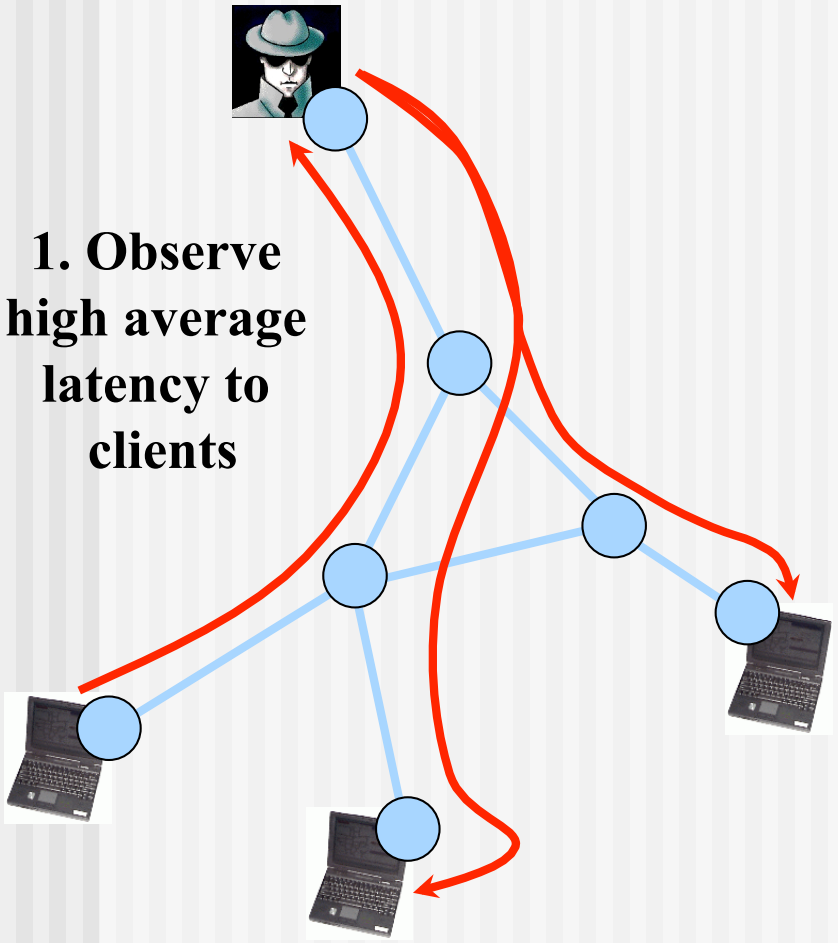
- Bandwidth conservation
- Reduction of latency
- Reduction of completion time
- Asynchronous (disconnected) communications
- Load balancing
- Dynamic deployment

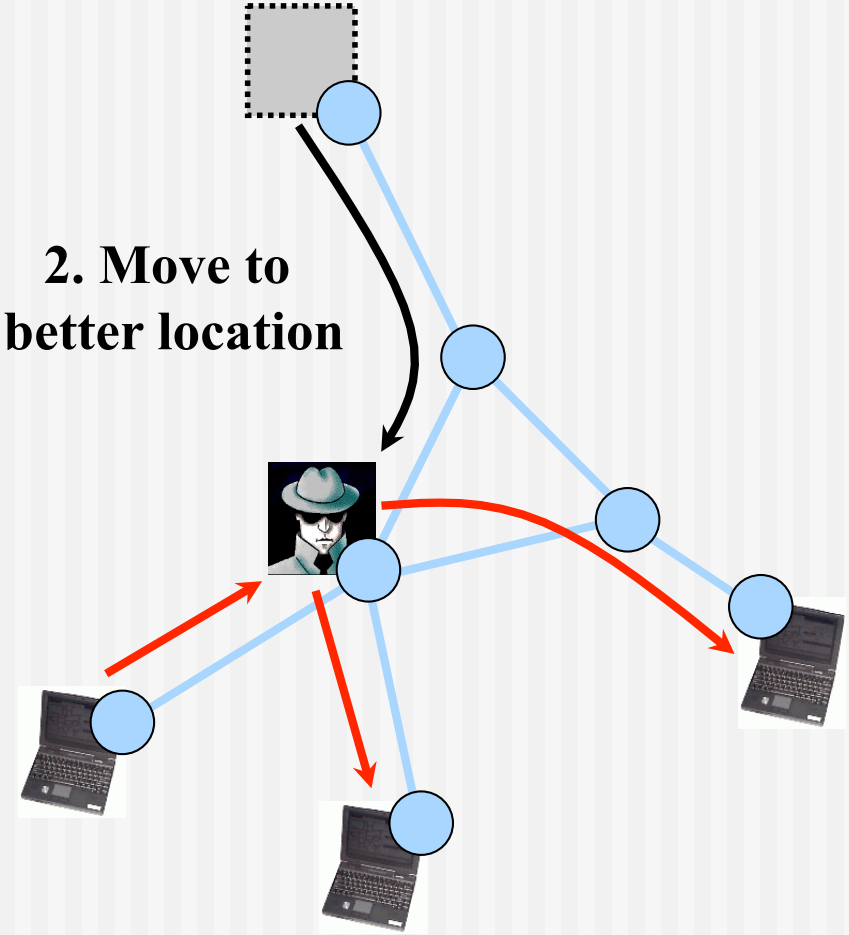
Reason 1: Bandwidth conservation



Reason 2: Reduce latency

Sumatra chat server
(a “reflector”)





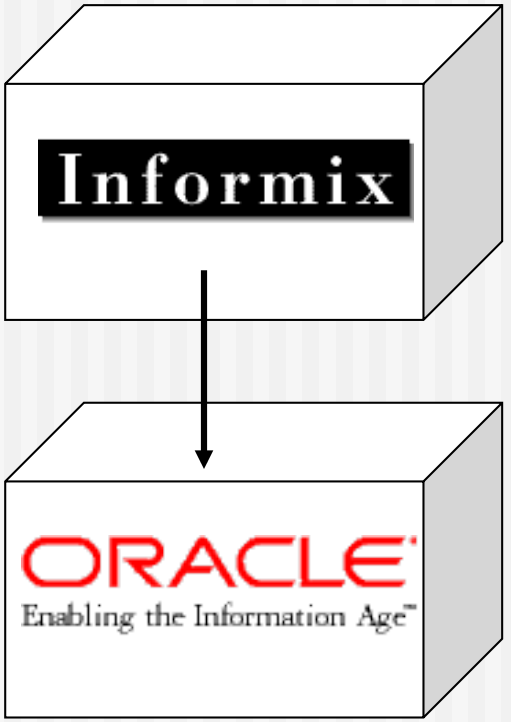
Reason 3: Reduce Completion Time

Efficiency

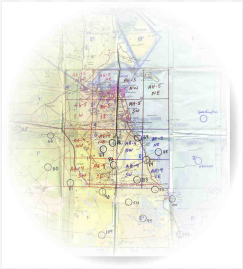
1. Send code with unique query

Mobile users

Low bandwidth channel

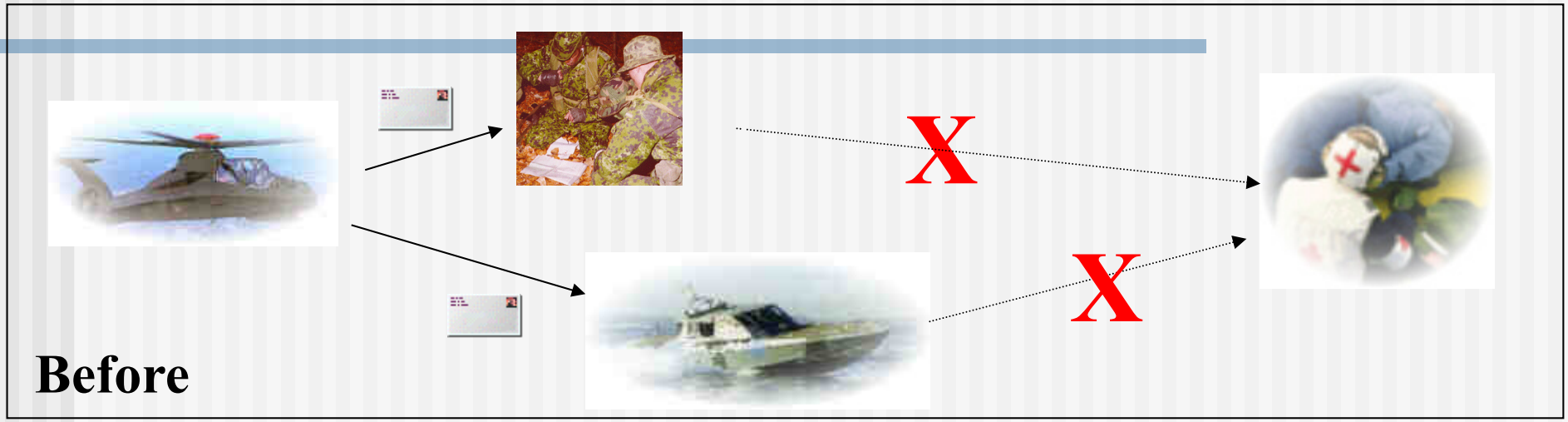


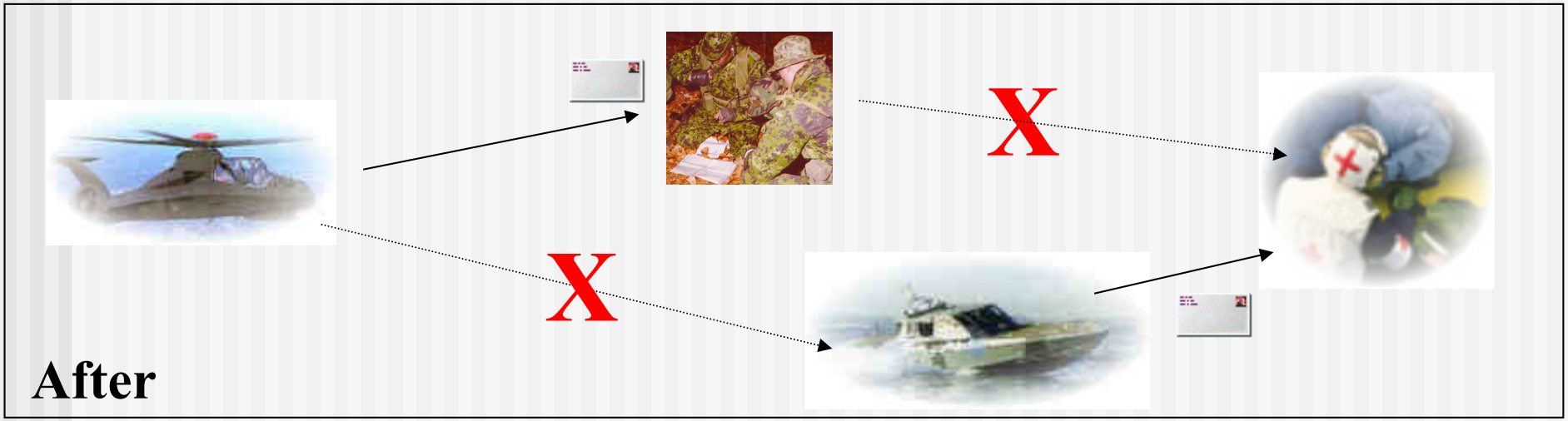
3. Return requested data



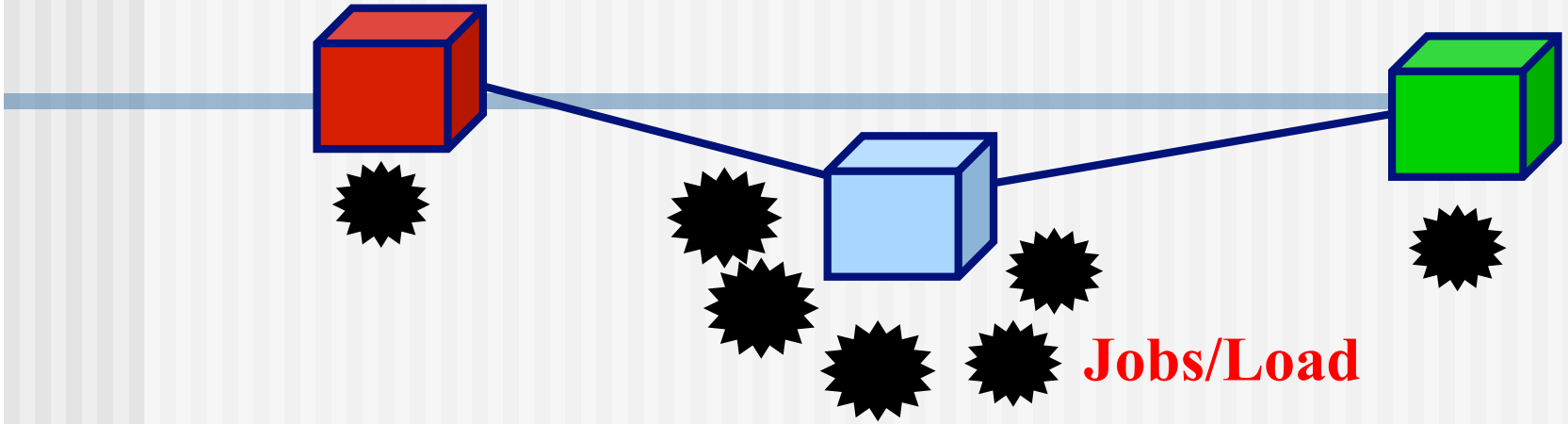
2. Perform multi-step queries on large, remote, heterogeneous databases

Reason 4: Disconnected communication and operation

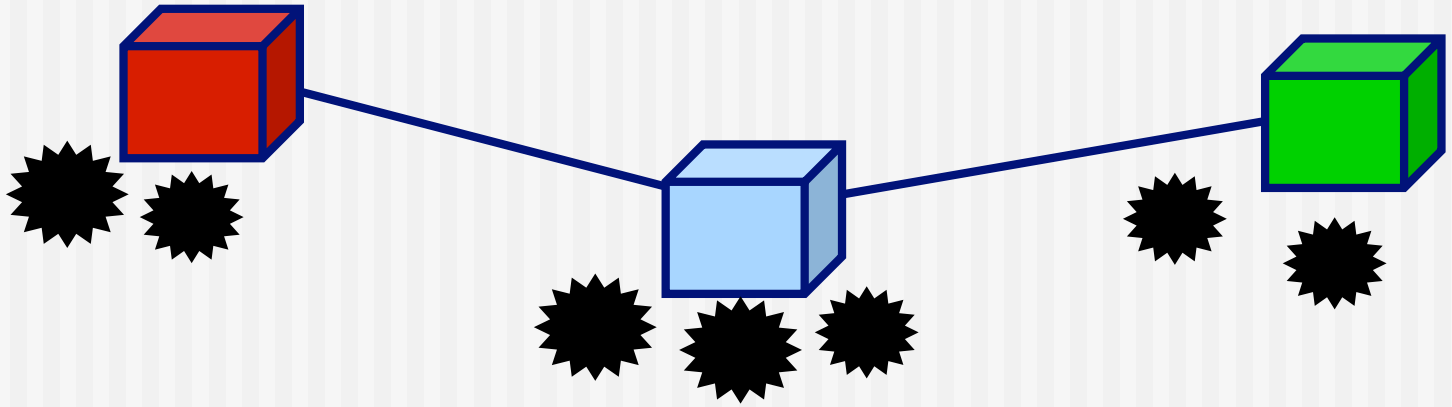




Reason 5: Load balancing

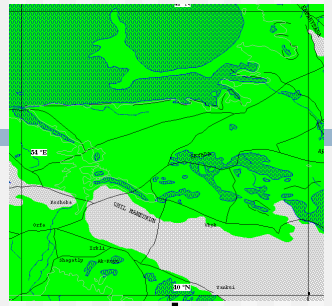


Jobs/Load



Jobs/Load migrate in a heterogeneous network of machines

Reason 6: Dynamic Deployment

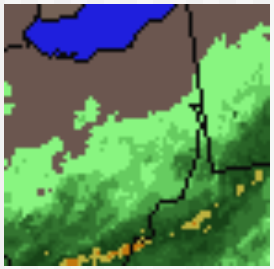


Map, terrain databases

Command post

Tactical updates

D'Agents



Weather

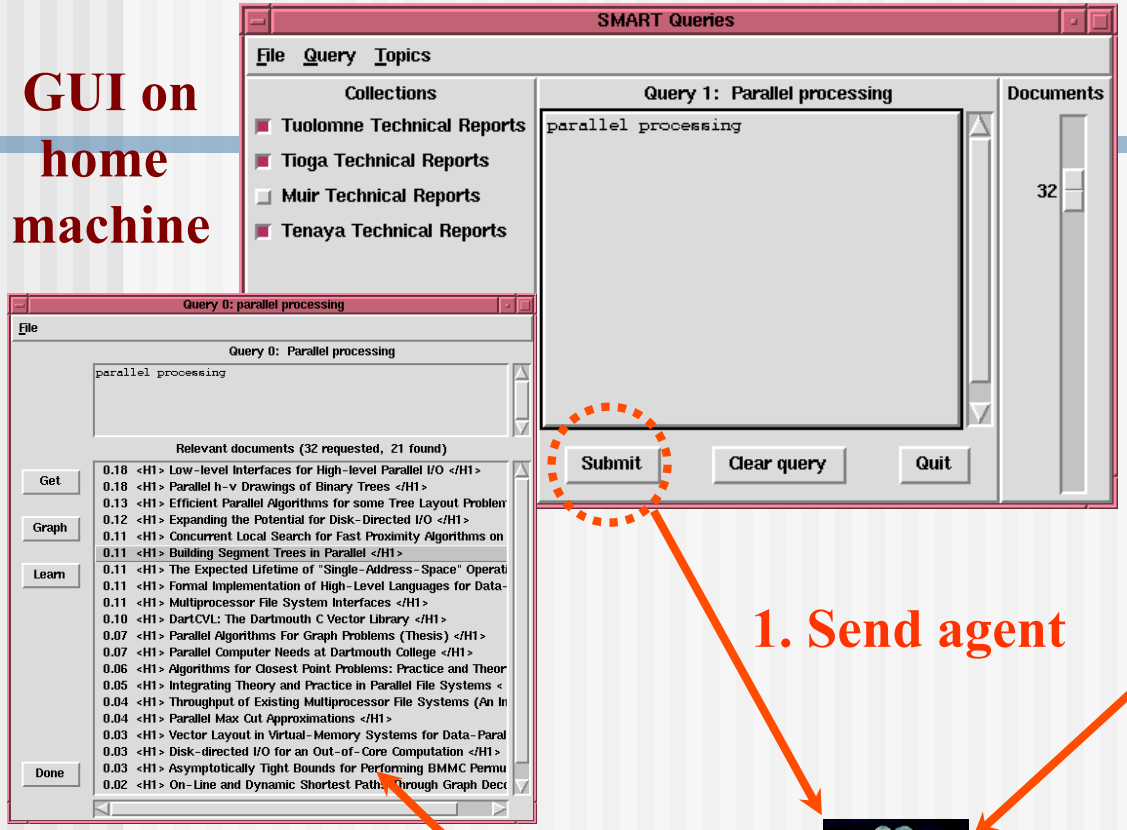
Unique needs:
maps,
weather,
tactical updates....

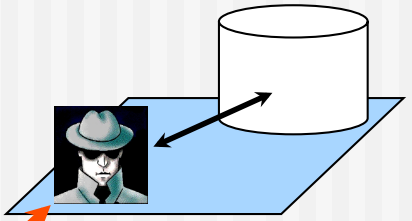
Mobile-agent applications

- Information retrieval
- Monitoring
- Virtual market-place/ meeting room
- Shareware
- Personal Mobile Agent white paper, http://www.x-fetch.com/common/X-Fetch_Personal_Mobile_Agent_White_Paper.pdf
- **IEEE Network Magazine** special issue on Applicability of Mobile Agents to Telecommunications, May-June 2002

Application: Technical reports

GUI on home machine





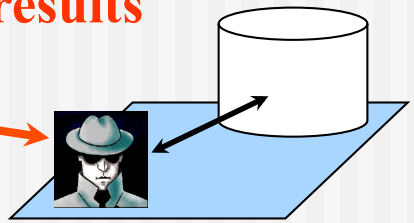
Machine 1

1. Send agent

2. Send child agents / collect partial results

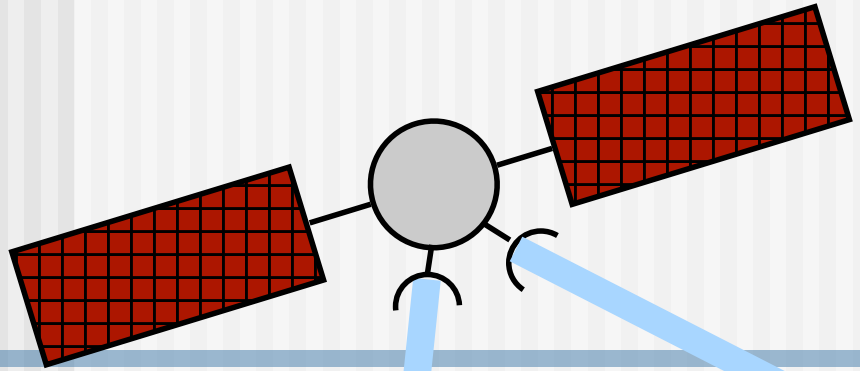
3. Return merged and filtered results

Dynamically selected proxy site



Machine n

Application: Military



Wired network

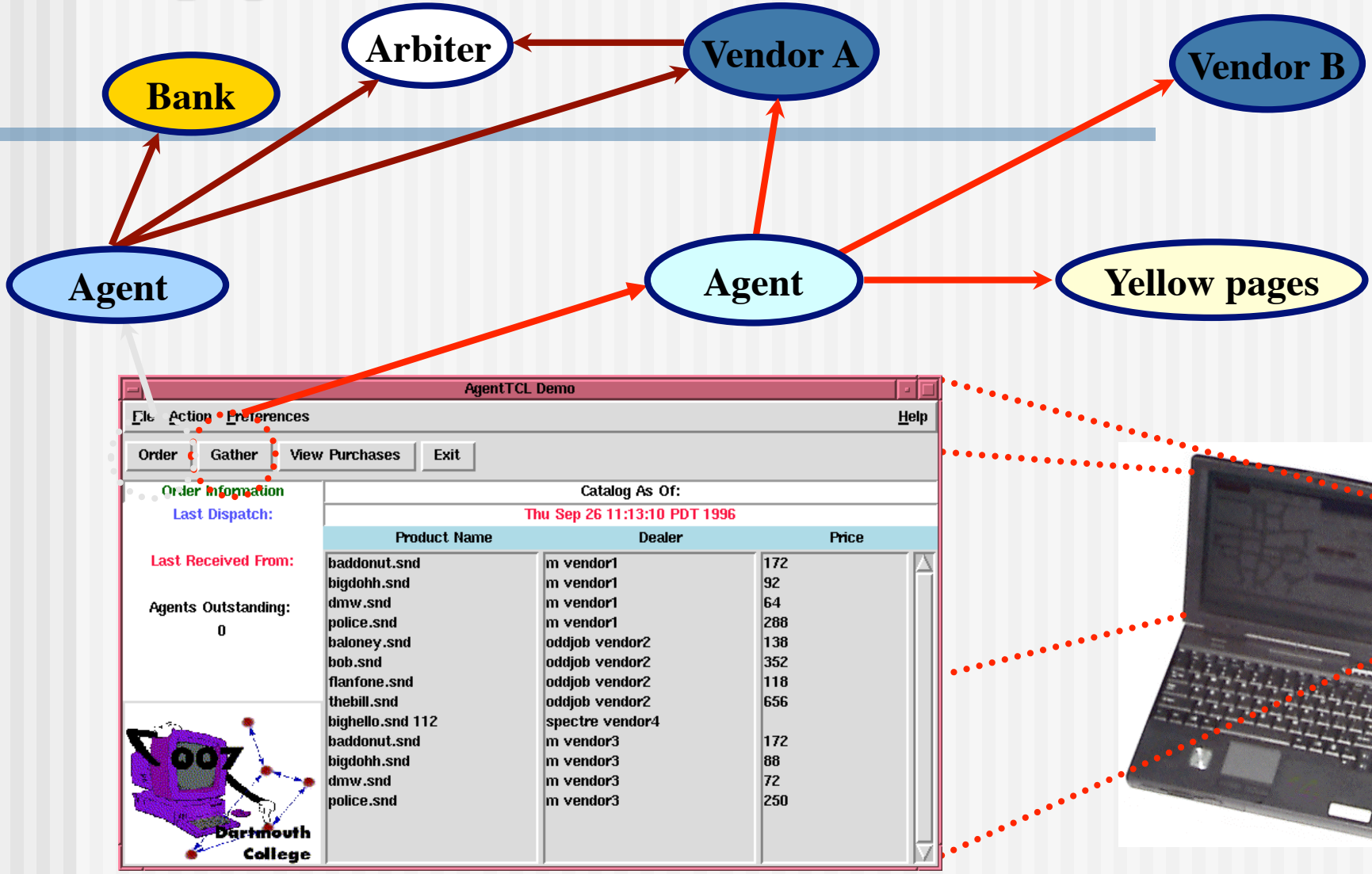
Wireless Network

Technical specs

Troop positions

Orders and memos

Application: e-commerce



More examples and “bots”

- Tryllian mobile agent system
- Bots
  - mysimon.com
  - amazon.com - customer preferences

Security in Mobile Agent Systems

http://mole.informatik.uni-stuttgart.de/security.html

- Security concern is the primary deterrent of deploying the mobile-agent technology.
- There are concerns for both the agent hosts and the mobile agents.
- Agent host concerns:
  - Malicious/unauthorized agents can misuse/ destroy system resources.
- Agent concerns:
  - Malicious hosts can destroy or alter an agent's logic, e.g.,Mobile agent's route can be altered.

Security in Mobile Agent Systems

http://mole.informatik.uni-stuttgart.de/security.html

Measures:

- *Authentication* – an agent must authenticate itself to the host, and an agent server must authenticate itself to the agent.
- *Encryption* – an agent encrypts its sensitive data.
- *Resource access* – a host enforces strict access control to its resources.

Threats posed by mobile agents

- Destruction of
  - data, hardware, current environment
- Denial of service
  - block execution
  - take up memory
  - prevention of access to resources/network

Threats posed by mobile agents

- Breach of privacy / theft of resources
  - obtain/transmit privileged information
  - use of covert channels
- Harassment
  - Display of annoying/offensive information
  - screen flicker
- Repudiation
  - ability to deny an event / action ever happened

Protection methods against malicious mobile agents

- Authenticating credentials
  - certificates and digital signatures
- Access Control and Authorization
  - Reference monitor
  - security domains
  - policies
- Software-based Fault Isolation
  - Java's "sandbox"

Protection methods against malicious mobile agents

- Monitoring
  - auditing of agent's activities
  - setting limits
- Proxy-based approach to host protection
- Code Verification - proof-carrying code

Threats to mobile agents

- Denial of service
- Unauthorized use or access of code/ data
- Unauthorized modification or corruption code/data
- Unauthorized access, modification, corruption, or repeat of agent external communication

Possible attacks on mobile agents

- Denial of service
- Impersonation
  - Host
  - Agent
- Replay
- Eavesdropping
  - Communication
  - Code & data
- Tamper attack
  - Communication
  - Code & data

Protection of mobile agents

- Encryption
  - code
  - payload
- Code obfuscation
- Time-limited black-box security

Mobile-agent framework systems

Using RMI to implement a mobile agent application is not generally recommended

(**http://developer.java.sun.com/developer/onlineTraining/rmi/RMI.html - MobileAgentArchitectures**)

"The solution to the mobile computing agent using RMI is, at best, a work-around. Other distributed Java architectures have been designed to address (security concerns and other issues.) These are collectively called *mobile agent architectures*. Some examples are IBM's Aglets Architecture and ObjectSpace's Voyager System. These systems are specifically designed to allow and support the movement of Java objects between JVMs, carrying their data along with their execution instructions."

Mobile agent systems

Mobile Agent System	Author	Language	Secure Communication	Server Resource	Agent Protection
Telescript	General Magic	Created their own	Agent transfer is authenticated	Capability-based	Not supported
		OO, type-safe	using RSA and encrypted	resource access. Quotas	
		language	using RC4	can be imposed.	
				Authorization based on	
				agent's identity	
Tacoma	Cornell University	Tcl, but is created	Not supported	Not supported	Not supported
	University of Tromso,	to be written in other			
	Norway	scripting languages			
D'Agents	Dartmouth College	Tcl interpreter, mo-	Uses PGP for authentication	Uses safe-Tcl as its	Not supported
		dified to execute	and encryption	secure execution envireon	
		scripts and capture		ment. No support for	
		state of execution at		owner-based authorization	
		thread level			
Aglets	IBM	Java. IBM developed	Not supported	Statically specified access	Not supported
		a separate class		rights, based on only two	
		library to create		security categories:	
		mobile agents		trusted and untrusted	
Voyager	ObjectSpace	Java. Unique feature	Not supported	Programmer must extend	Not supported
		is a utility which		Security Manager. Only	
		takes any Java class		two security categories:	
		and creates a remo-		native and foreign.	
		tely-accessible ver-			
		sion of it.			
Concordia	Mitsubishi Electric	Java. Has Itinerary	Agent transfer is encrypted and	SecurityManager screen	Agents protected from
		object, which keeps	authenticated using SSL	acceses using a statically	other agents via the
		track of an agent's		configured ACL based on	resource access
		migration path		agent owner identity	mechanism
Ajanta	University of	Java	Transfer is encrypted using	Capability-based resource	Mechanisms to detect
	Minnesota		DES and authenticated using	access. Authorization	tampering of agent's
			ElGamal protocol	based on agent's owner	state and code

The Mobile Agent System Interoperability Facility (MASIF)

- From the OMG (The Object Management Group) site:

  "Mobile agent platforms have been developed, built on top of different operating systems, based on different programming languages and technologies. Even new languages have been realized, exclusively designed for the support of mobile agents. However, within the last few years, common trends can be noticed: Interpreter-based programming languages like Java build the basis for most of today's agent platforms, and several approaches are associated with the integration of mobile agents and RPC-based middleware like CORBA."

MASIF

- "In course of time, several fundamental requirements have been identified due to experiences that have been made during research and development activities. These requirements which should be fulfilled by any state of the art mobile agent platform cover the following topics:
  - Management Support
  - Security Support
  - Mobility Support
  - Support for Unique Identification
  - Transaction Support
  - Communication Support "

MASIF

- Due to the considerations mentioned above, the OMG issued a Request for Proposal (Common Facilities RFP3) for a mobile agent standard in November 1995.
- The corresponding Mobile Agent System Interoperability Facility (MASIF) submission, developed by Crystaliz, General Magic, GMD FOKUS, IBM, and The Open Group, has been adopted by the OMG in February 1998.

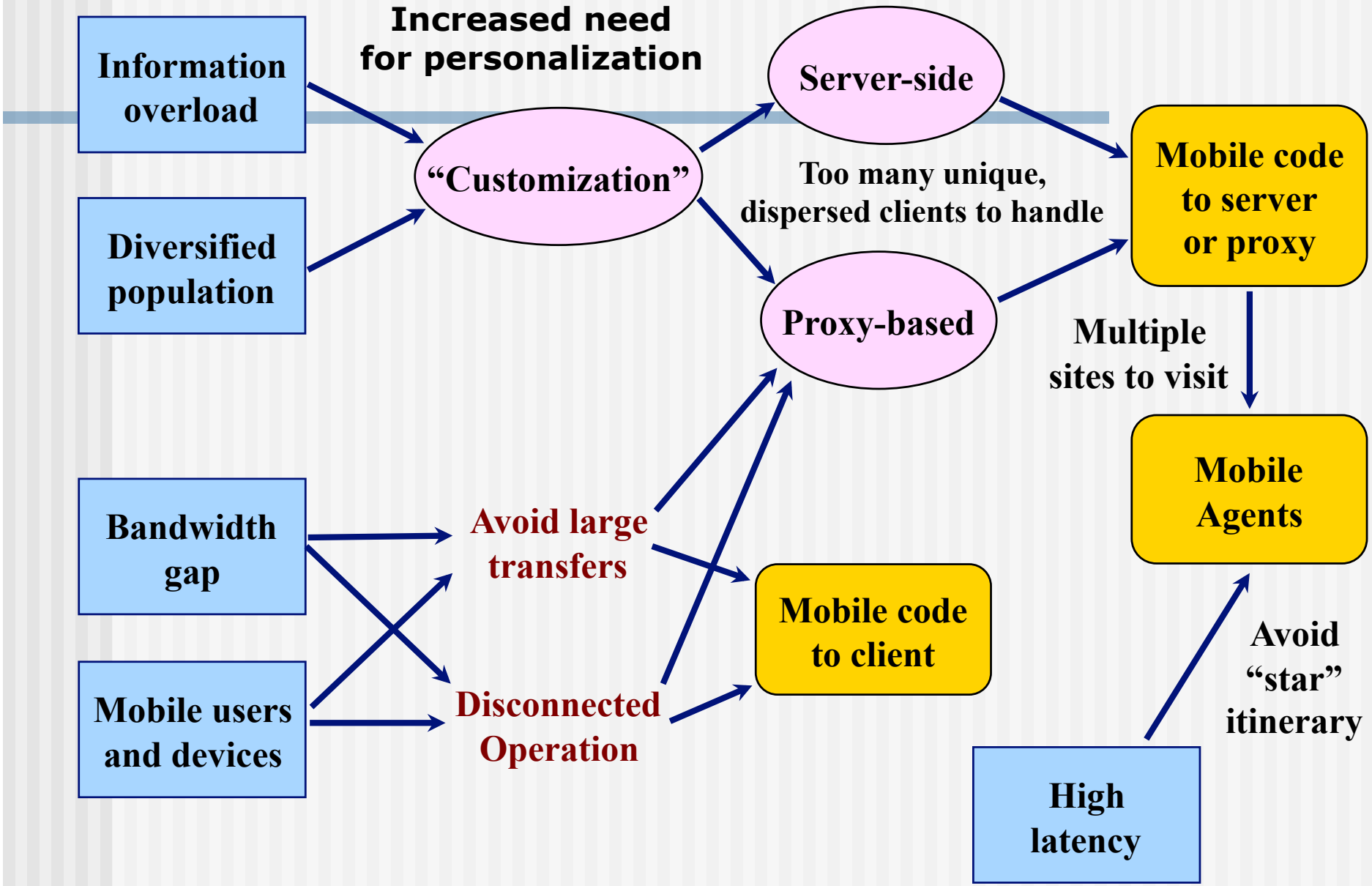
MASIF

- The idea behind the MASIF standard is to achieve a certain degree of interoperability between mobile agent platforms of different manufacturers without enforcing radical platform modifications.
- MASIF is not intended to build the basis for any new mobile agent platform. Instead, the provided specifications shall be used as an "add-on" to already existing systems.

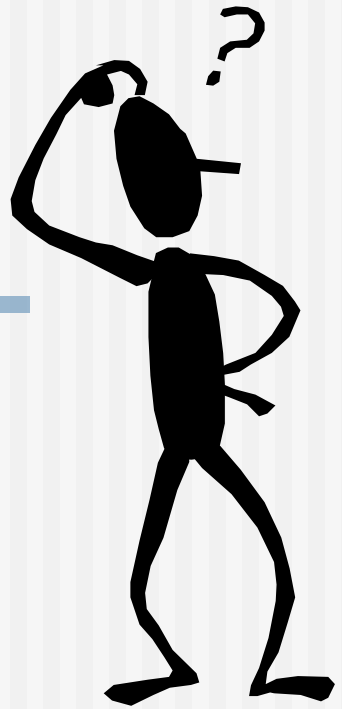
MASIF

- The following list comprises the mandatory requirements that were identified within the MASIF RFP:
  - Marshalling and un-marshalling of agent programs
  - Encoding of agent containers for transport
  - Transport of agents from one agent facility (i.e. execution engine) to another
  - Runtime registration and invocation of agent facilities
  - Runtime query of a named agent facility by agents
  - Runtime security of agents

Current trends lead to mobile agents



Conclusion



- Security is too big a concern
- Overhead for moving code is too high
- Not backward compatible with Fortran, C ….
- Networks will be so fast, performance not an issue

Conclusion

- A unifying framework for making many application more efficient
- Treats data and code symmetrically
- Multiple-language support possible
- Supports disconnected networks in a way that other technologies cannot
- Cleaner programming model

Sources of Information

- Mobile Agents Introductory http://www.infosys.tuwien.ac.at/Research/Agents/intro.html
- The Mobile Agent Listhttp://mole.informatik.uni-stuttgart.de/mal/mal.html
- Mobile Agent Applicationshttp://www.computer.org/concurrency/pd1999/pdf/p3080.pdf
- Software Engineering Concerns for Mobile Agent Systems, http://www.elet.polimi.it/Users/DEI/Sections/Compeng/GianPietro.Picco/ICSE01mobility/papers/cook.pdf