Introduction To JINI

"..Any device, any time, any where..."

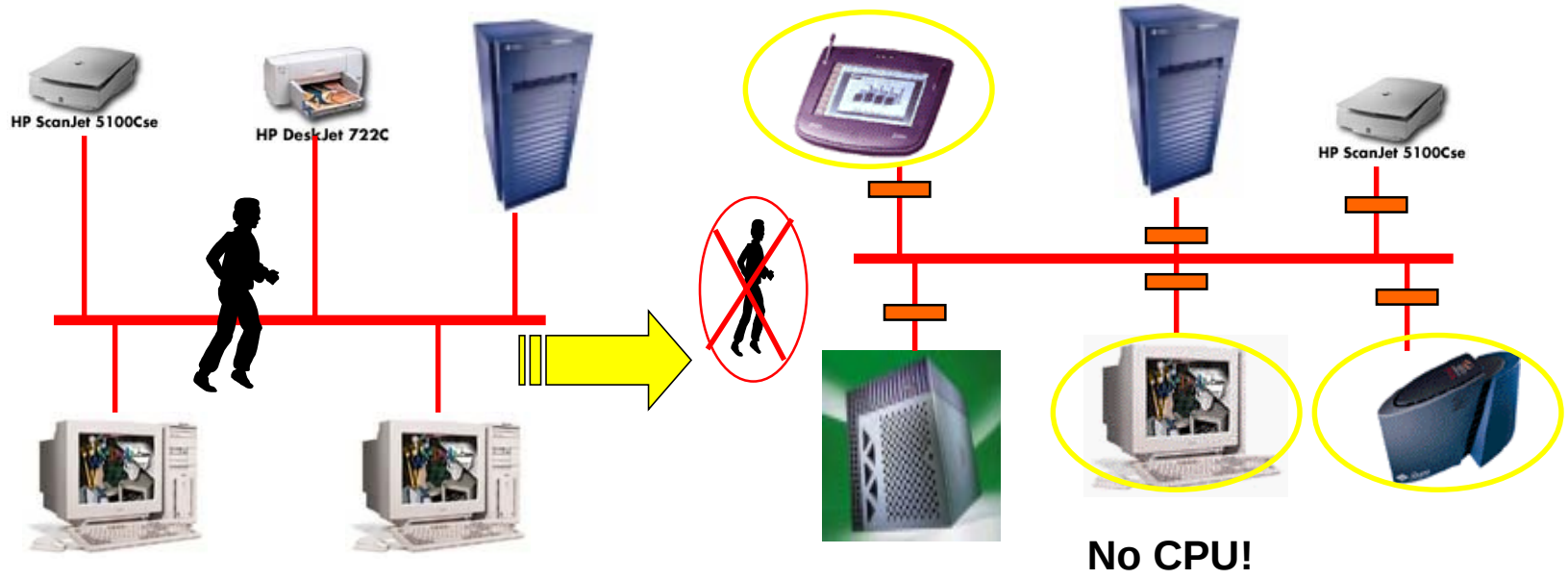
Jinn (jinni, genie) 😐

One of a class of spirits lower than the angels, capable of appearing in human and animal forms, and exercising influence over mankind for good and evil

New Webster's Dictionary, 1981

By **BIREN DESAI**
GMU INFT 803

JINI: The Network Really Is The Computer



Today **Future**

"JINI promises to make setting up a computer/device as simple as hooking up a telephone. "

JINI Vision

- Digital Camera, Printer, Cell phone etc.. Plug them into TCP/IP network and use the variety of other devices/service
- Any resource available on the network is available to your JINI-enabled devices
- Also includes Desktop and enterprise software

WHO

- JINI comes from some of the best minds in distributed computing today.
- Dream Team
  Bill Joy, Jim Waldo, Ann Wollrath, Ken Arnold, Bob Scheifler

Classical Network System

- Move data to the code - not code to data
- Using technology like XDR (external data representation).
  - Take care of diff. Notion of byte orders, floating point..
- Via RPC or ROS (CORBA, DCOM), make the call of function on remote machine look (to the developer) like the call of a local function.
- Their goal is to make network transparent and to make network simply “go away”

The Network is not Transparent

- It is not hard to pack data into portable forms, causing the invocation of a remote procedure somewhere on the net and so forth
- Hard part is to take care of the things about network that can't be ignored
  - latency, performance, network failure, concurrency and consistency….

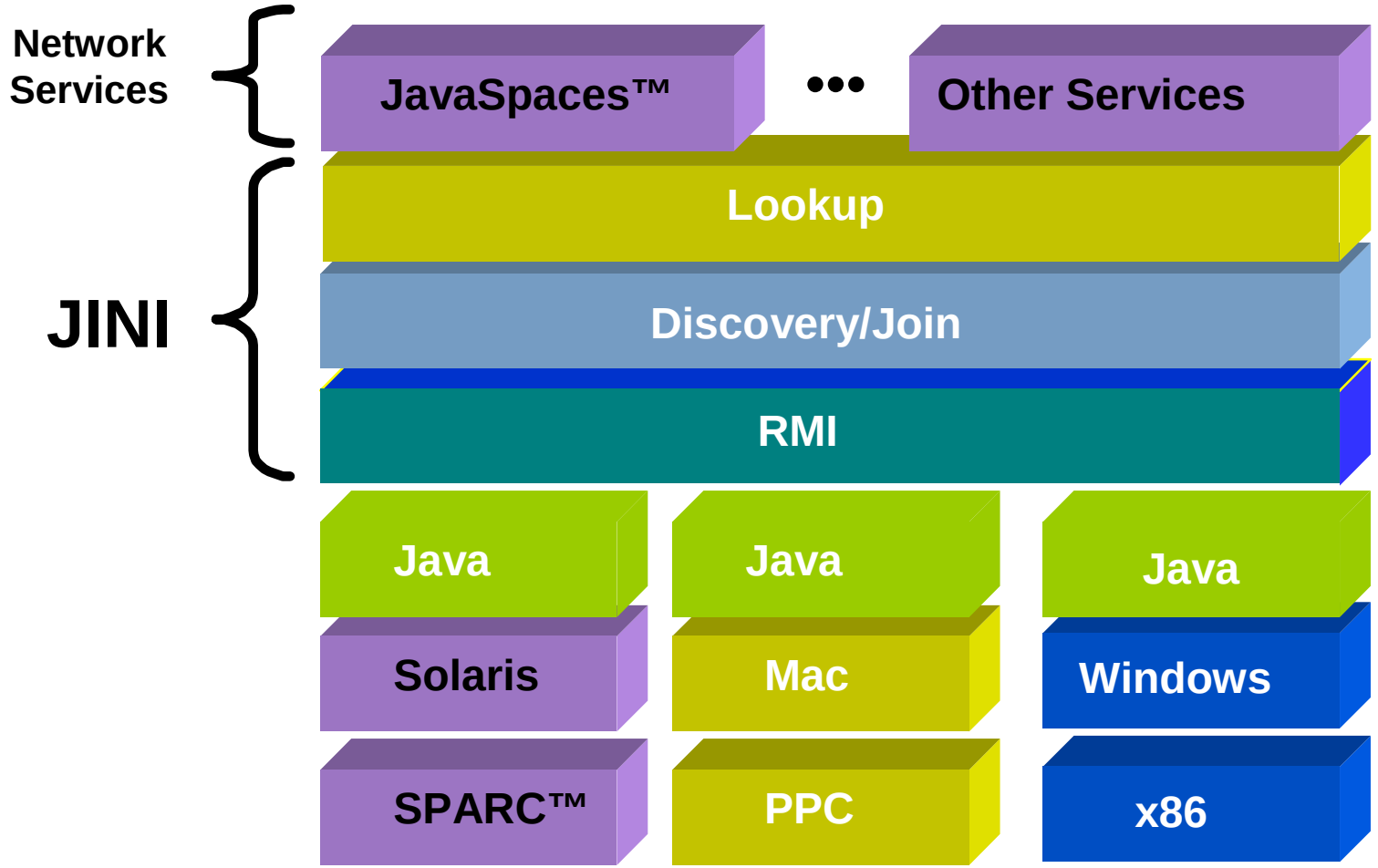
Seven fallacies of distributed computing
by Peter Deutsch

"..Seven assumption, all prove to be wrong in a long run.."

- The network is reliable
- Latency is zero
- Bandwidth is infinite
- The network is secure
- Topology doesn't change
- There is one administrator
- Transport cost is zero

" ...*Much of the work in JINI is designed to explicitly acknowledge these fallacies, rather than pretend they don't exist.."*

JINI Overview



Challenges

- Has to be robust
  - Toaster, TV can not fail with message asking
    - "Abort, Retry, Ignore?
- Has to support true,effortless "Plug and Play"
  - Just like Telephone
- Must be evolvable
  - Software services and devices to be able to use each other without massive reconfiguration.

JINI Architecture

	Infrastructure	Programming Model	Services
JINI	Discovery Lookup Extended Security	Lease Event Transaction	JavaSpaces TX Manager
Java Language	Java VM RMI Security	Java APIs Beans ...	Enterprise JavaBeans™ JNDI JTS ...

"JINI can be seen as an extension of the infrastructure, programming model, and services of Java"

JINI Infrastructure

	Infrastructure	Programming Model	Services
JINI	*Discovery* *Lookup* *Extended Security*	Lease Event Transaction	JavaSpaces TX Manager
Java Language	Java VM RMI Security	Java APIs Beans ...	Enterprise JavaBeans JNDI JTS ...

"The set of components that allow the building of a JINI system"

Discovery

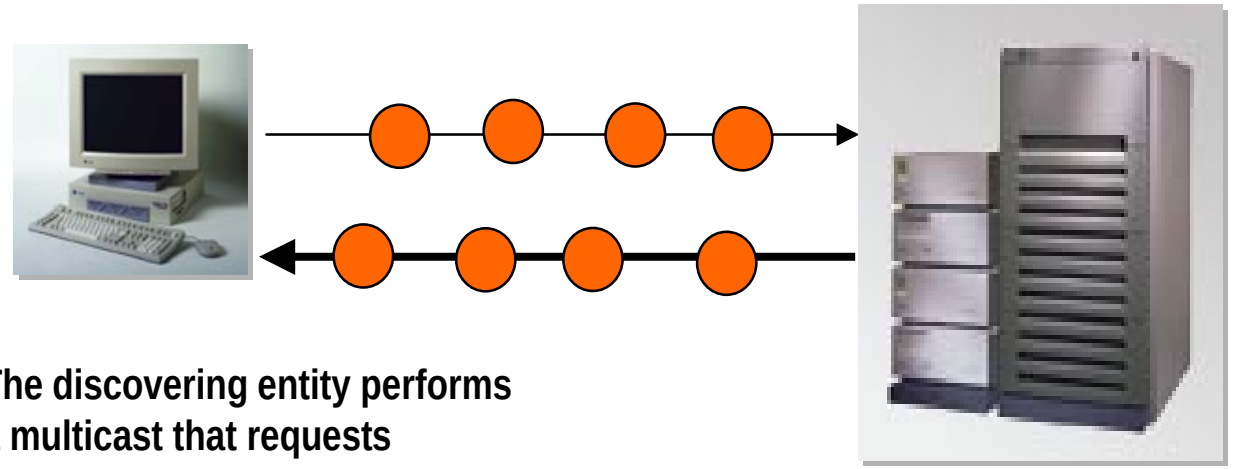
- Allows JINI services (both hardware and software) to:
  - **Find and join a group of JINI devices**
  - **Advertise its capabilities**
  - **Provide any required software and attributes**
  - **Works with JVM-enabled or non-JVM devices**
    - **Send out a multicast packet with reference to yourself**
    - **Receive a RMI reference to the Lookup service**

"Discovery solves the problem of finding the place to start in an unknown network"

Discovery in Brief

2. Lookup servers run instances of the discovery service which listen for multicast requests from discovering entities.

1. The device advertises and looks for a service.



3. The discovering entity performs a multicast that requests references to the lookup service.

4. The lookup server calls a remote method on the discovering entity's exported object instance passing a remote reference to its lookup service as the parameter.

Lookup

- Repository of available services
- Stores service as an extensible set of Java objects
  - ID, interface, GUIs, attributes, drivers ...
- Service objects downloaded to user as required
- May be federated with other lookup services
- Lookup service interface:
  - Registration, Access, Search, Removal

"The lookup service binds the federation together"

Distributed Security

- Extends the Java technology security model
- Identity carried in remote invocation
- Capabilities
  - Authentication
  - Integrity
  - Confidentiality
  - Delegation

"A simple model that minimizes the impact of security on the developer"

JINI Programming Model

	Infrastructure	*Programming Model*	Services
JINI	Discovery Lookup Extended Security	*Lease* *Event* *Transaction*	JavaSpaces TX Manager
Java Language	Java VM RMI Security	Java APIs Beans ...	Enterprise JavaBeans JNDI JTS ...

"The set of interfaces that allow the construction of reliable services"

Lease

- Protocol for managing resources using a renewable, duration-based model
- Contract between objects
- Resources can be shared or non-shared

"Leasing provides a method of managing resources in an environment where network failures can occur"

Leasing

- Jini leases expire after a predetermined time unless renewed by the holder
  - Can be cancelled early (without penalty!)
- Third-party leasing
  - A service can delegate the process of lease-renewal to a third-party
- Leasing vs Garbage Collection
  - Key difference: garbage collection used to free up resources when no active references to the resource exist. In leasing, there are typically never any references to begin with, e.g. a service that is registered but not being used

Event

- Extends Java event model to allow it to work in a distributed network
- Register interest, receive notification
- Allows for easy use of “event managers”
- Can use numerous distributed delivery models
  - Push, pull, filter ...
- Uses leasing protocol

“JINI events deal with peculiarities of messages in the networked environment, such as latency and network failure”

Remote Events

- Generic Delegates
  - Jini has the ability to create third party event listeners that can respond to any event type
  - Multiple event delegates can be composed or stacked
- Jini does not provide any guarantees about event delivery
  - Leaves it to applications to decide what policies they need
  - Example: create a logging event delegate that logs all events a service is interested in. If the service is disconnected, when it regains connectivity it can explicitly ask the logging delegate for its events

Transaction

- Designed for distributed object coordination
- Light weight, object-oriented
- Supports various levels of ACID properties:
  Atomicity:Consistency:Isolation: Durability
- Supports nested transactions

"Allows distributed entities to cooperate in such a way that the changes to the group occur automatically or not at all"

Transactions

- Provides two phase commit protocol
  - However, Jini does not specify what is done in each step of the commit protocol, I.e. prepare(), commit(), abort() are just interfaces. Jini runs through the 2 phase commit protocol but what participants do when these methods are called is up to them
  - Less restrictive but less safe notion of a transaction

JINI Services

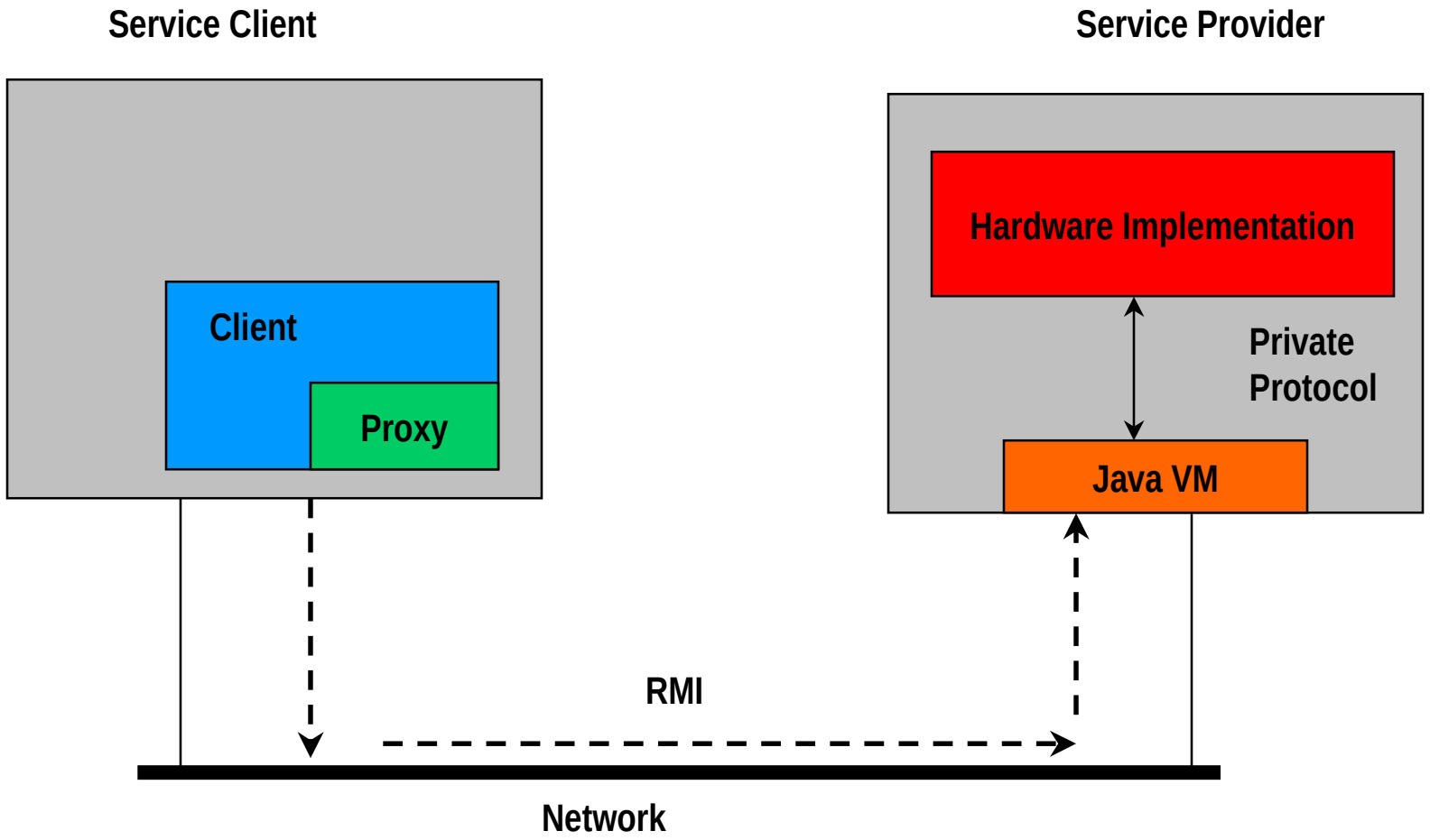
	Infrastructure	Programming Model	Services
JINI	Discovery Lookup Extended Security	Lease Event Transaction	*JavaSpaces* *TX Manager*
Java Language	Java VM RMI Security	Java APIs Beans ...	Enterprise JavaBeans JNDI JTS ...

"JINI services provide simple solutions to complex problems faced by developers in the distributed environment"

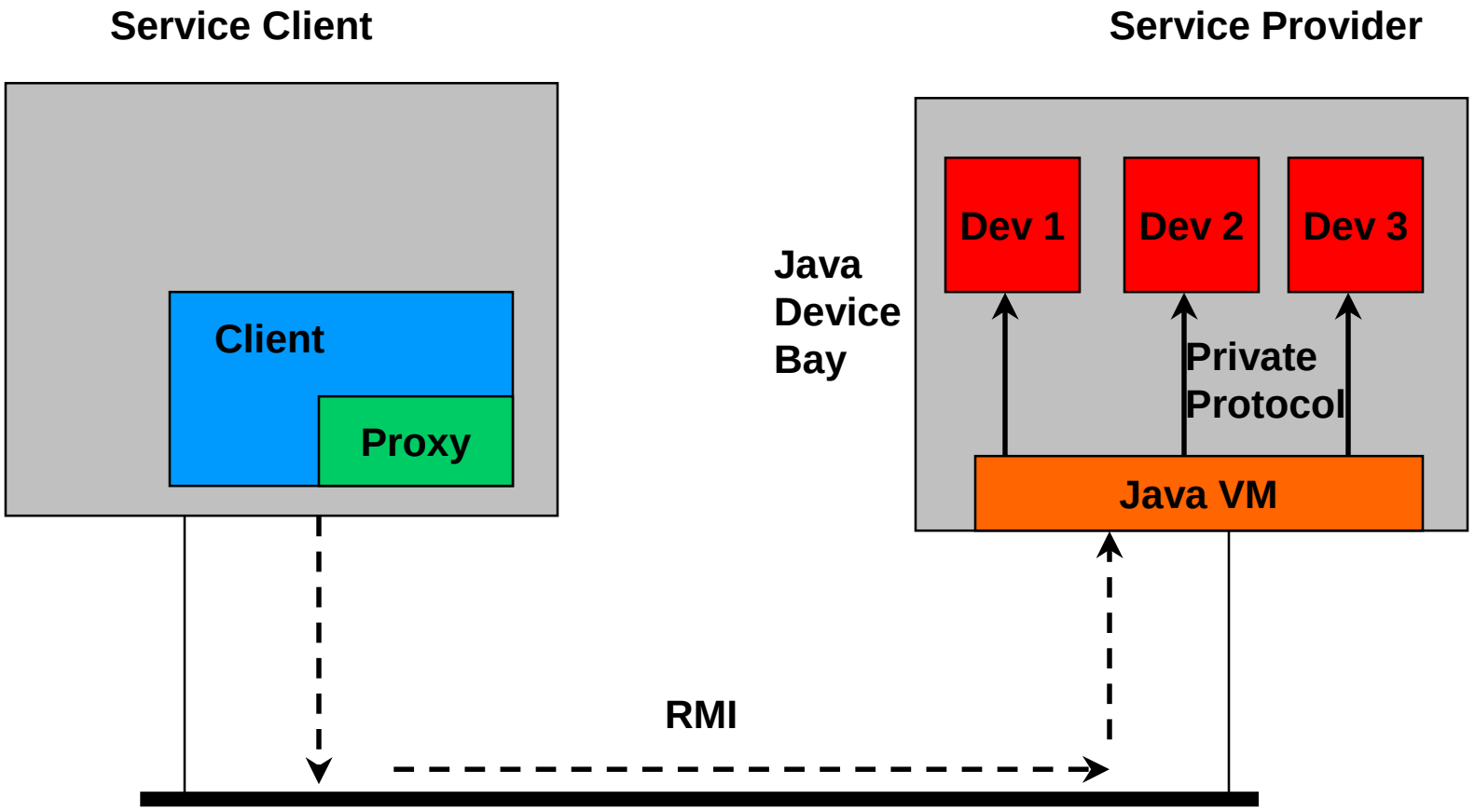
JavaSpaces

- JINI provides the distributed system services for look-up, registration, and leasing
- JavaSpaces manages features such as object processing, sharing, and migration
- JINI acts as a quartermaster allocating equipment and armaments to the JavaSpaces troops. The troops use the allocated equipment to carry out their orders in battle.

Service Model 1



Service Model 2



Scenario

"Let's take a look at what exactly happens when we plug in my brand new "JINI-enabled" printer? "

1. The printer broadcasts/multicasts a discovery packet (512 bytes) on a well-known port. The packet contains just enough information to contact the printer.

2. The lookup service running somewhere on the network listens on the well-known port/multicast group and receives the packet. It registers the information about the printer.

3. The lookup service replies to the printer with its location. With this, discovery is complete.

Scenario cont..

4. The printer is now ready to "join" the network. It adds each of its services to the lookup service.

5. All resources are now ready to use the printer, and the printer too is ready to use other resources on the network.

"what happens when a resource in the federation needs to use the printer service? "

1. The resource first queries the lookup service for a particular interface for accessing the printer's services.

2. If the interface is present in the lookup service, the client proxy for the print service is returned to the resource.

3. The resource invokes methods of the client proxy interface. These methods in turn activate the printer according to which methods were called.

The Value of JINI

- **Consumer**
  - Plugs network attachable devices and software as simple as plugging in a phone today
- **Service providers**
  - Simplifies management of services delivery
- **Product manufacturer**
  - Opens whole new markets
- **Java programmers**
  - Simplifies the task of writing distributed applications

"JINI technology creates a network dialtone for networks"

Conclusions

- JINI provides an elegant distributed programming model
- JINI provides the plumbing for a new generation of network-aware devices
- Check out our web sites...

http://www.sun.com/jini

http://java.sun.com/products/jini

http://www.jini.org/

http://www.w3com.com/paulcho/javalinux/jini.shtml