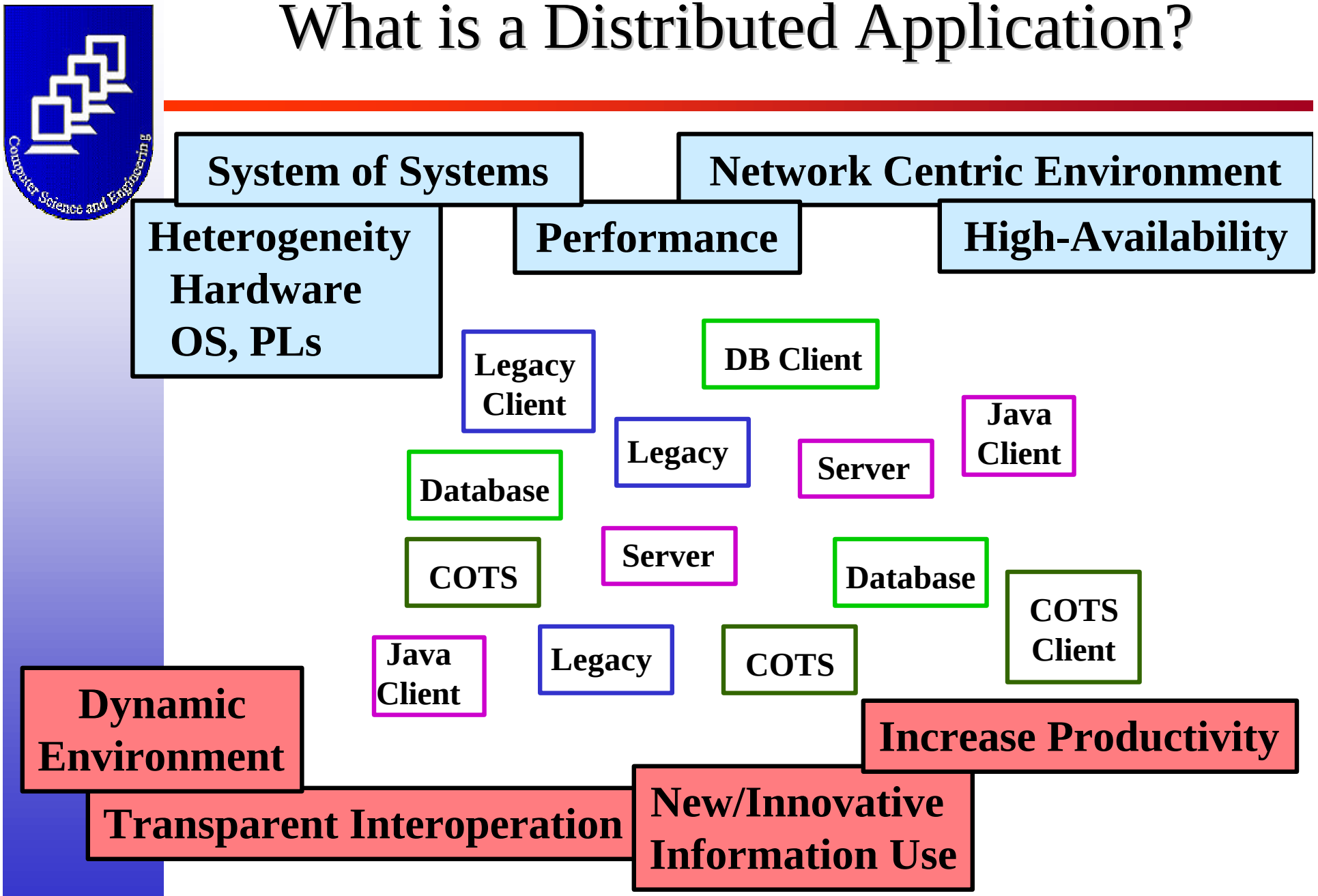
JINI: A Technology for 21st Century
Is it Ready for Prime Time?

Prof. Steven A. Demurjian, Sr.
Computer Science & Engineering Dept.
191 Auditorium Road, Box U-155
The University of Connecticut
Storrs, Connecticut 06269-3155
steve@engr.uconn.edu
www.engr.uconn.edu/~steve
Tel: 860-486-4818
Fax: 860-486-4817

Dr. Paul Barr
The MITRE Corp
145 Wyckoff Road
Eatontown, New Jersey 07724
poobarr@mitre.org
Tel: 732-935-5584
Fax: 732-544-8317

UCONN

What is a Distributed Application?



UCONN

Goals of Research Effort

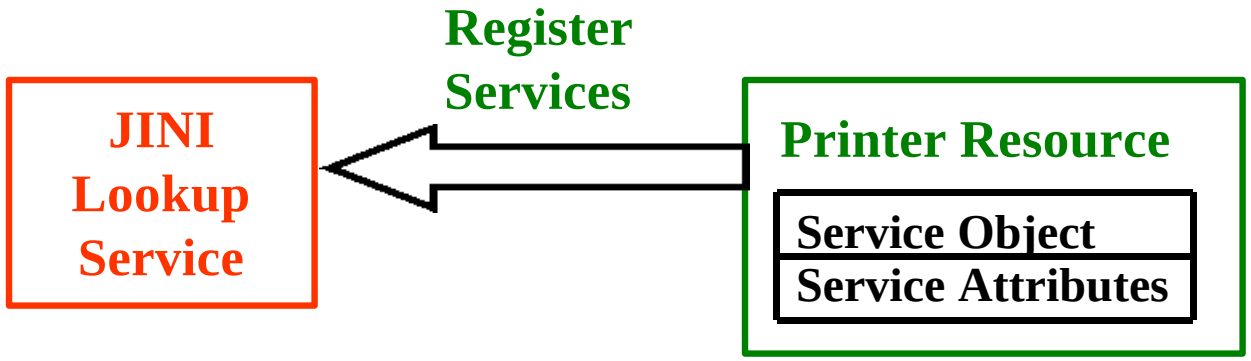
- Can JINI Support **Highly-Available Distributed Applications**?
- Can JINI Support a Network-Centric Environment with **Dynamic Clients and Services**?
- Will Clients Continue to Operate Effectively if **Replicated Databases Services Fail**?
- Can JINI be Utilized to **Maintain "minutes-off" Data Consistency** of Replicas?
- Is JINI Easy to Learn and Use?
- What is Maturity Level of JINI Technology?
- **Is JINI Ready for Prime Time???**

UCONN

Sun's JINI Technology
Key JINI Concepts and Terms

- A ***Resource*** Provides a Set of Services for Use by Clients (Users) and Other Resources (Services)
- A ***Service*** is Similar to a Public Method
  - Exportable - Analogous to API
  - Any Entity Utilized by Person or Program
  - Samples Include:
    - Computation, Persistent Store, Comm. Channel
    - Software Filter, Real-Time Data Source
    - Sensor or Probe, Hardware (Printer, Display, etc.)
    - **Anything that is Relevant for Your Domain!**
- Services Register with ***Lookup Service***
  - **Clearinghouse** for Resources to Register Services and Clients to Locate Services

Sun's JINI Technology Resources & Services



- Sun's Initial Perspective
  - JINI for Hardware
  - Printers, Digital Cameras, etc.
  - Plug-and-Play on Network
- Services Registered
  - Class by Class Basis
  - Granularity up to SWE

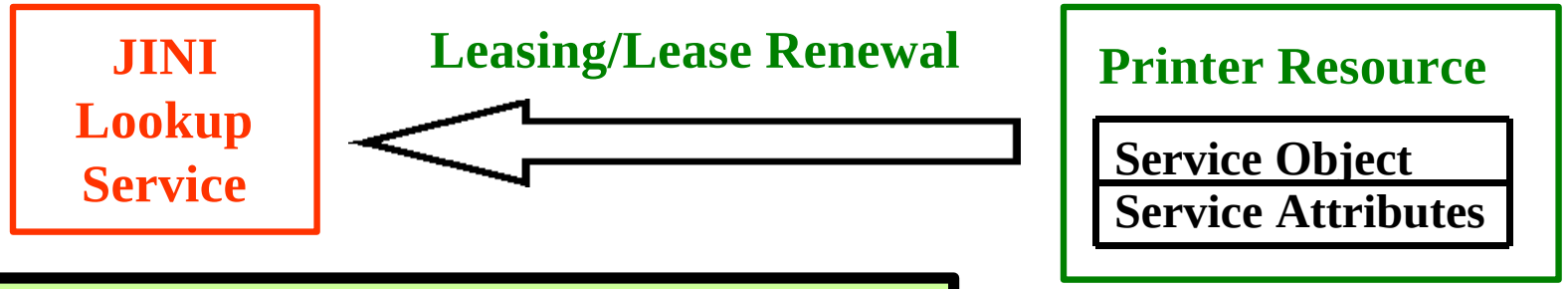
Class and Methods Define Services to be Registered

PrinterActions Class
- **enqueuePrintJob**
- **dequeuePrintJob**
- **getPrinterStatus**
- **getPrinterType**
- **installPrinter**
- **removePrinter**
- **startJob**
- **cancelJob**

UCONN

Sun's JINI Technology Registration & Leasing

- FOREVER or EXPIRATION DATE (millisecs)
- Renewal Must Occur Prior to Expiration
- JINI Provides Lease Renewal Manager to Allow Resource to Delegate Renewal Responsibility

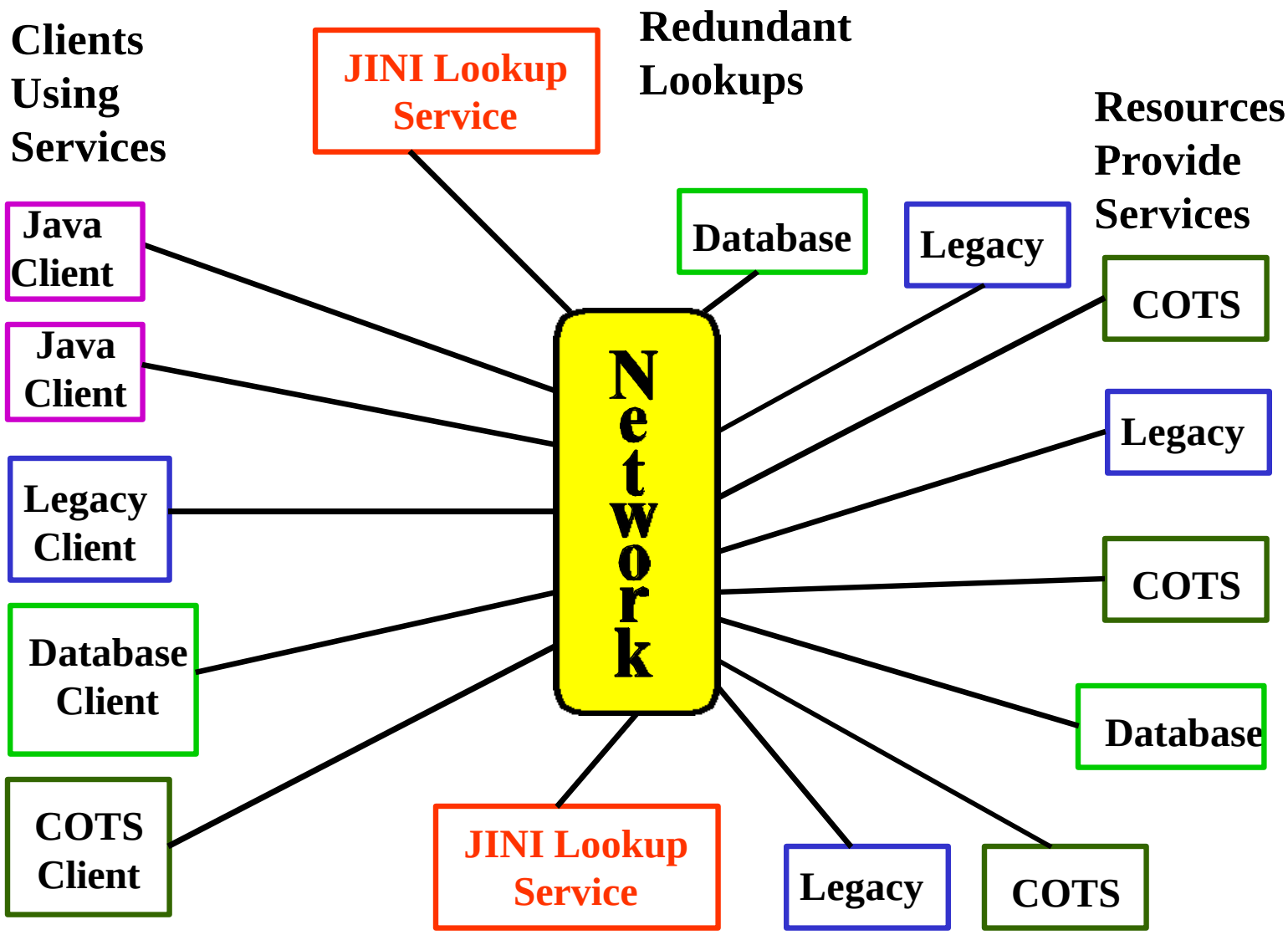


Lease for 5 minutes (3000000 msec)
Must Renew Before 5 Minutes Expire
If Not Renewed, Lookup Removes
If Failure, Lookup May Still Supply Service Until Expiration (5 mins)
Client MUST be SMART!

Class and Methods Define Services to be Registered

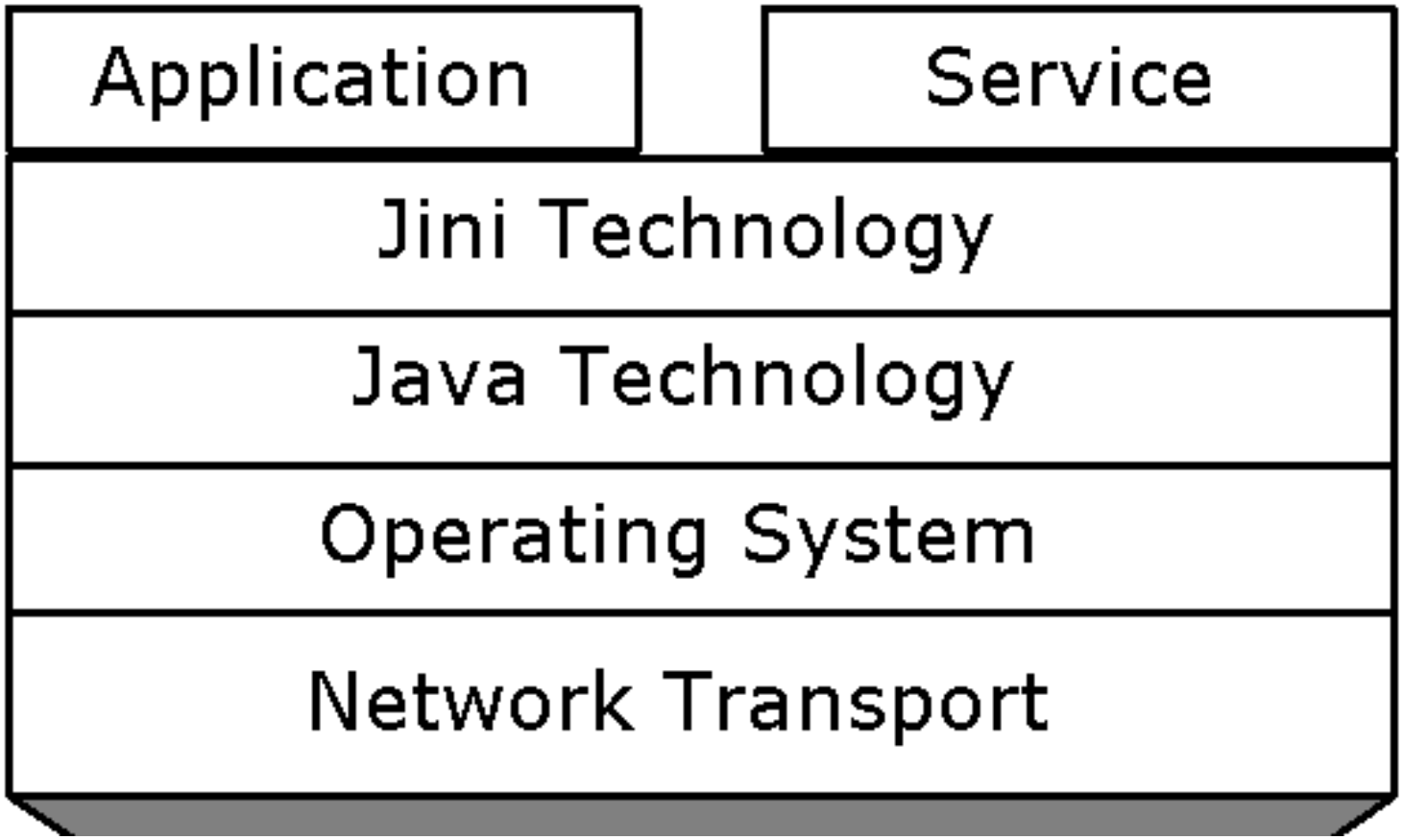
- PrinterActions Class
  - enqueuePrintJob
  - dequeuePrintJob
  - getPrinterStatus
  - getPrinterType
  - installPrinter
  - removePrinter
  - startJob
  - cancelJob

Sun's JINI Technology Support for Distributed Computing

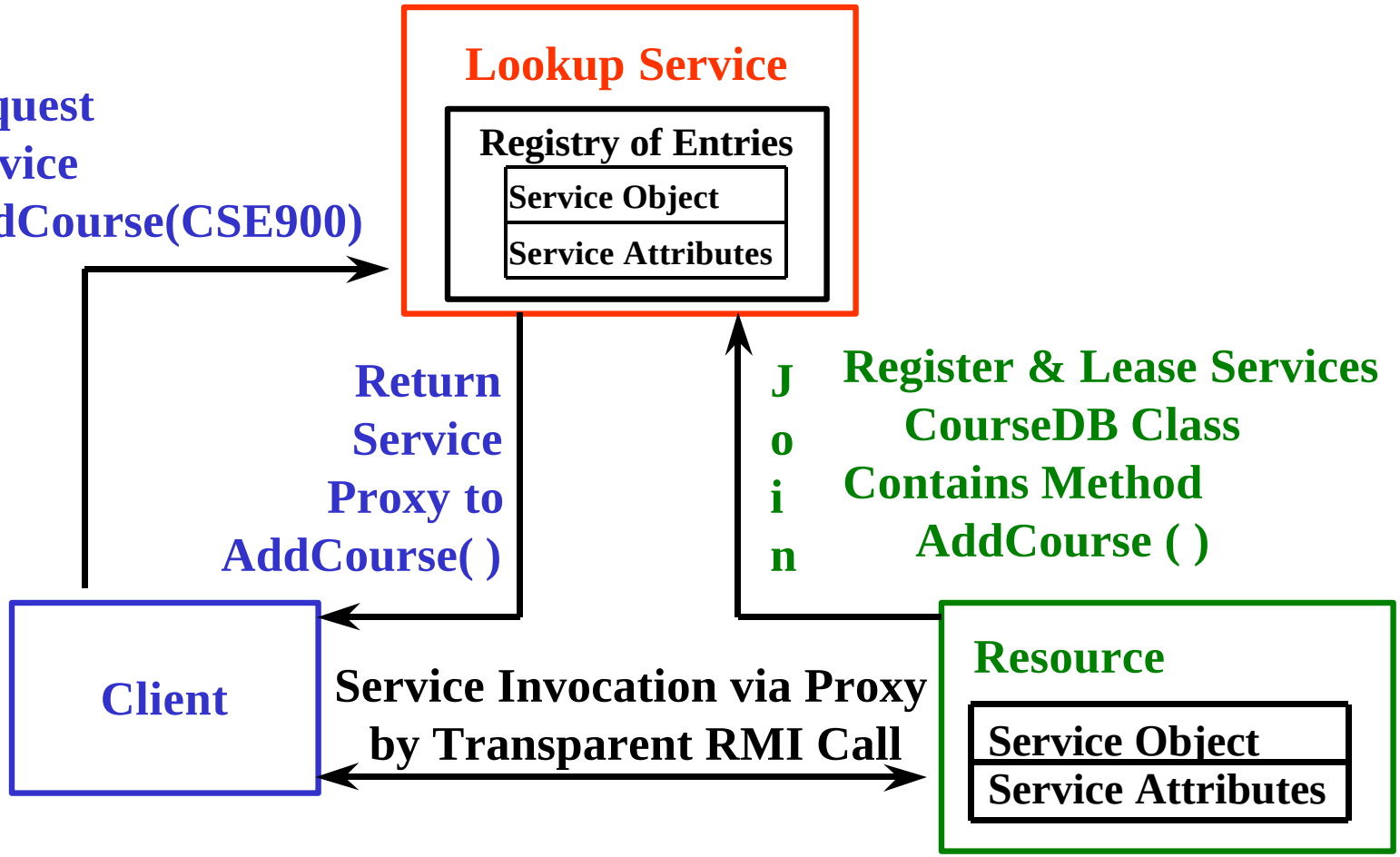


UCONN

Sun's JINI Technology Overall Computing Architecture and JINI.



Sun's JINI Technology
Join, Lookup, and Service Invocation



1. Client Invokes AddCourse(CSE900) on Resource
2. Resource Returns Status of Invocation

UCONN

Experimental Prototyping Effort Goals and Objectives

- High Availability of Services and Data
- Volatility of Resources, Clients, and Network
- Clients Rely on Lookup Service to Locate and Execute Services Against Replicated Resources
  - Databases Replicated in Multiple Workstations
  - Redundant Services Available if Failure
  - “Minutes-Off” Allowed - Sync Over Time
  - No Lost Updates During Modification Process
- **Characteristics of Enterprise Applications**
  - **Movement of Clients/Reconfigure Networks**
  - **Need for Data Availability on Demand**
  - **Receive “Current” Data if Client/Resource Plugs Back In**

UCONN

Experimental Prototypes Rapid, Incremental Design/Development

- First Prototype: Explore JINI and Develop Baseline University Application: Single Computer
- Second Prototype:Client and Services Spread to Two Computers: Lookup with Database Services
- Third Prototype: Extend Second to Multiple Clients and Three Computers
- Fourth and Fifth Prototypes
  - Single Client, Three Replicated Databases
  - **Testing of Replica Failures on Application**
  - Fifth: Multiple Clients/**Simultaneous Updates**
- **Sixth Prototype: Extends Fifth PT with Pre-Lookup Services for Locking During Updates**
- Six Prototypes in Six Weeks by 2 Grad Students

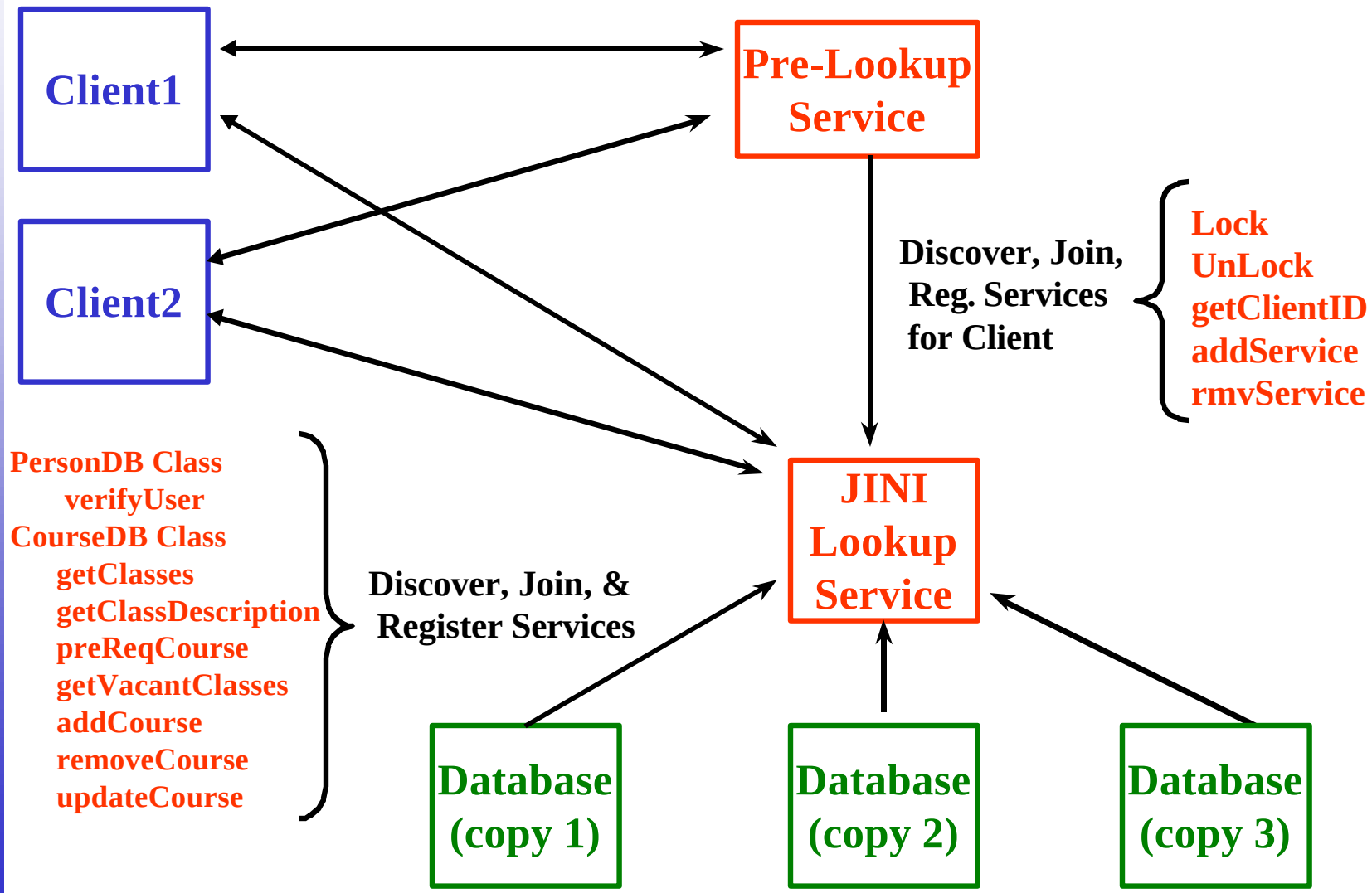
UCONN

Prototype Six

- Functionality:
  - Incorporate Pre-Lookup Service that Insures Only One Client Updates Replicas
  - Use a Exclusive Write/Simultaneous Reads Protocol
  - Client Interacts with Pre-Lookup Service to “Request” Locks
  - Client Then Asks Lookup Service for Services
  - Client Receives and Updates All Replicas
- Purpose:
  - Bootstrap: Use JINI to Solve Update Problem
  - Eliminate Consistency Problems Across Replicas by Locking Databases During Update

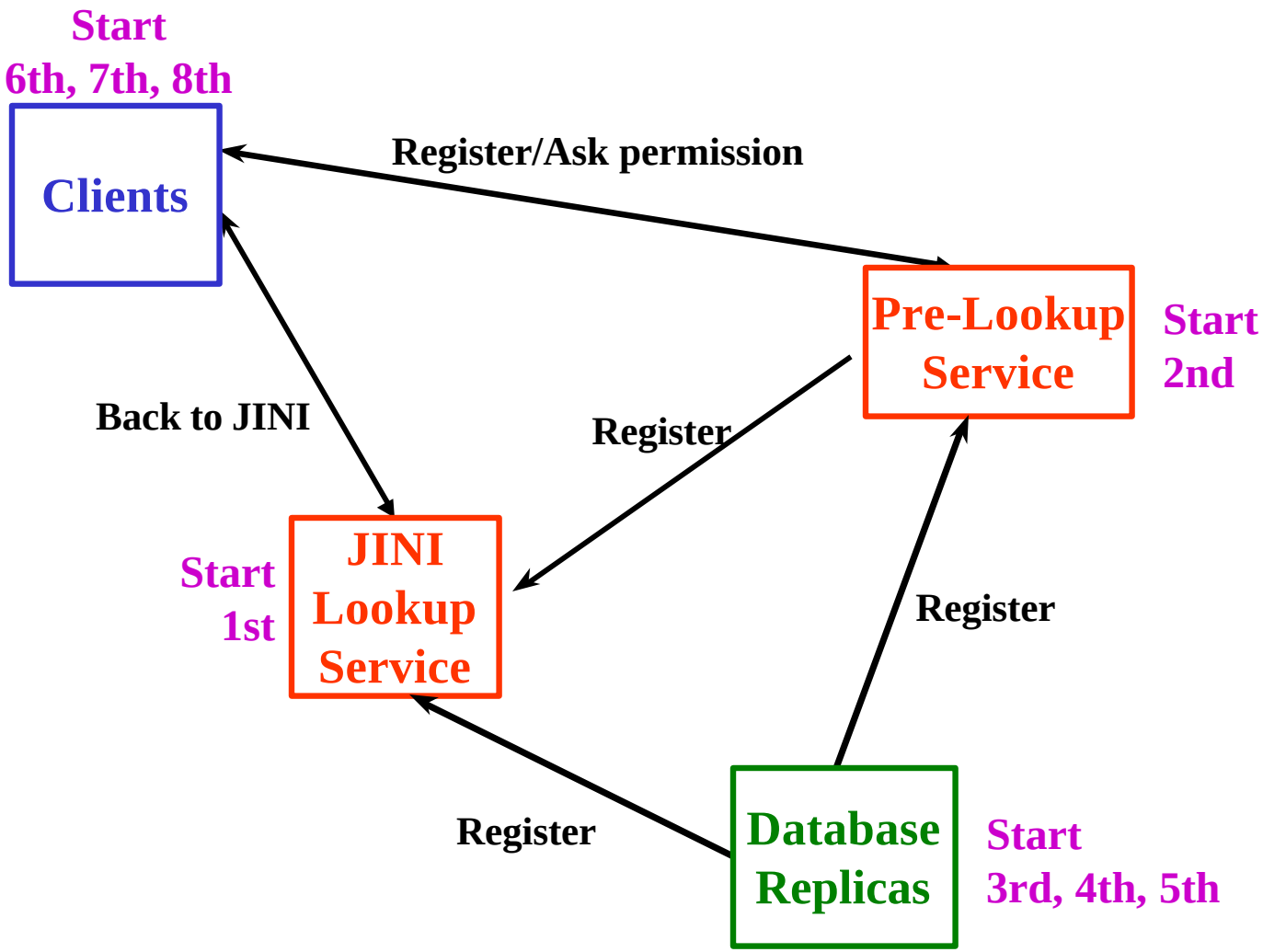
UCONN

Services in Prototype Six

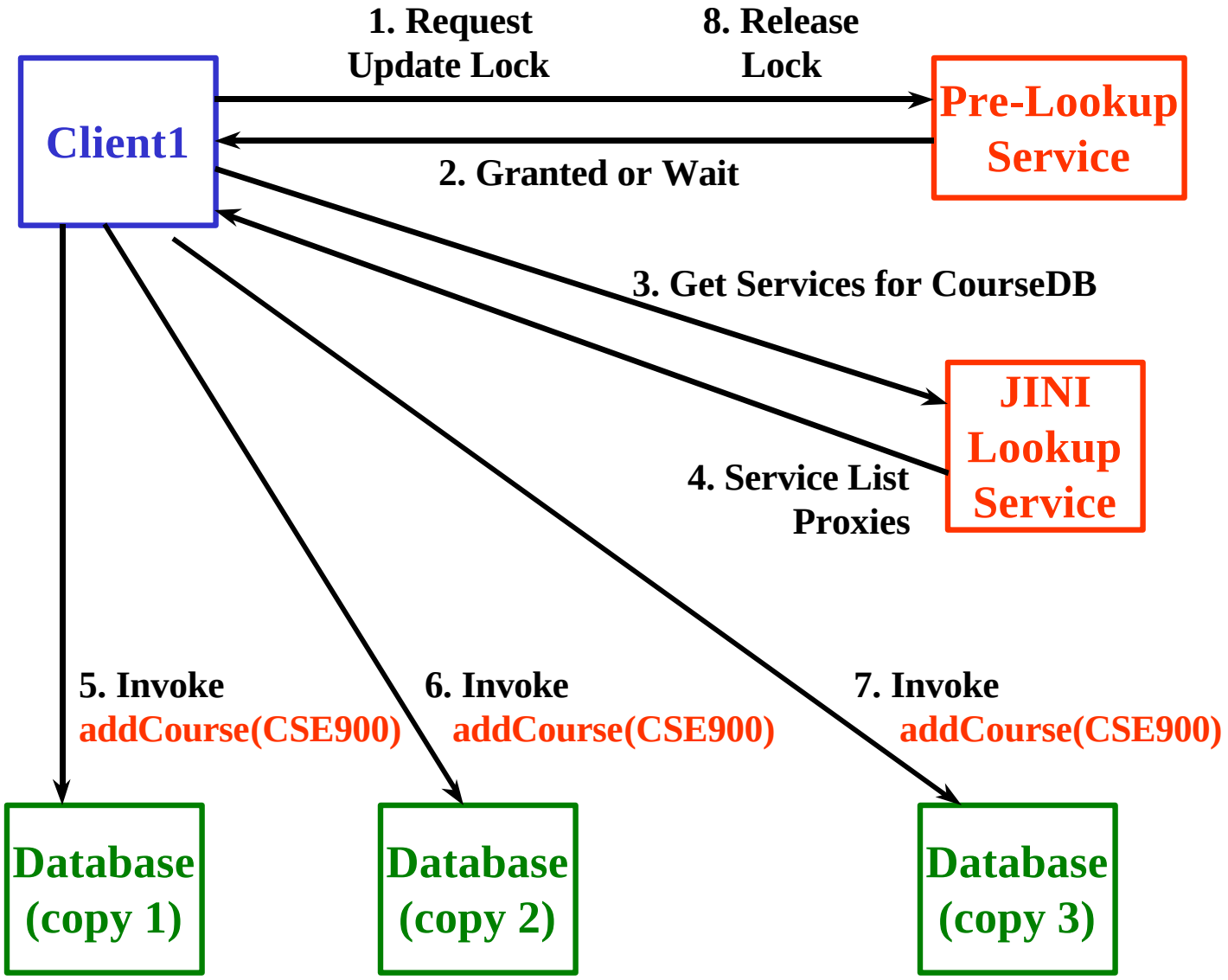


UCONN

Order to Start Application

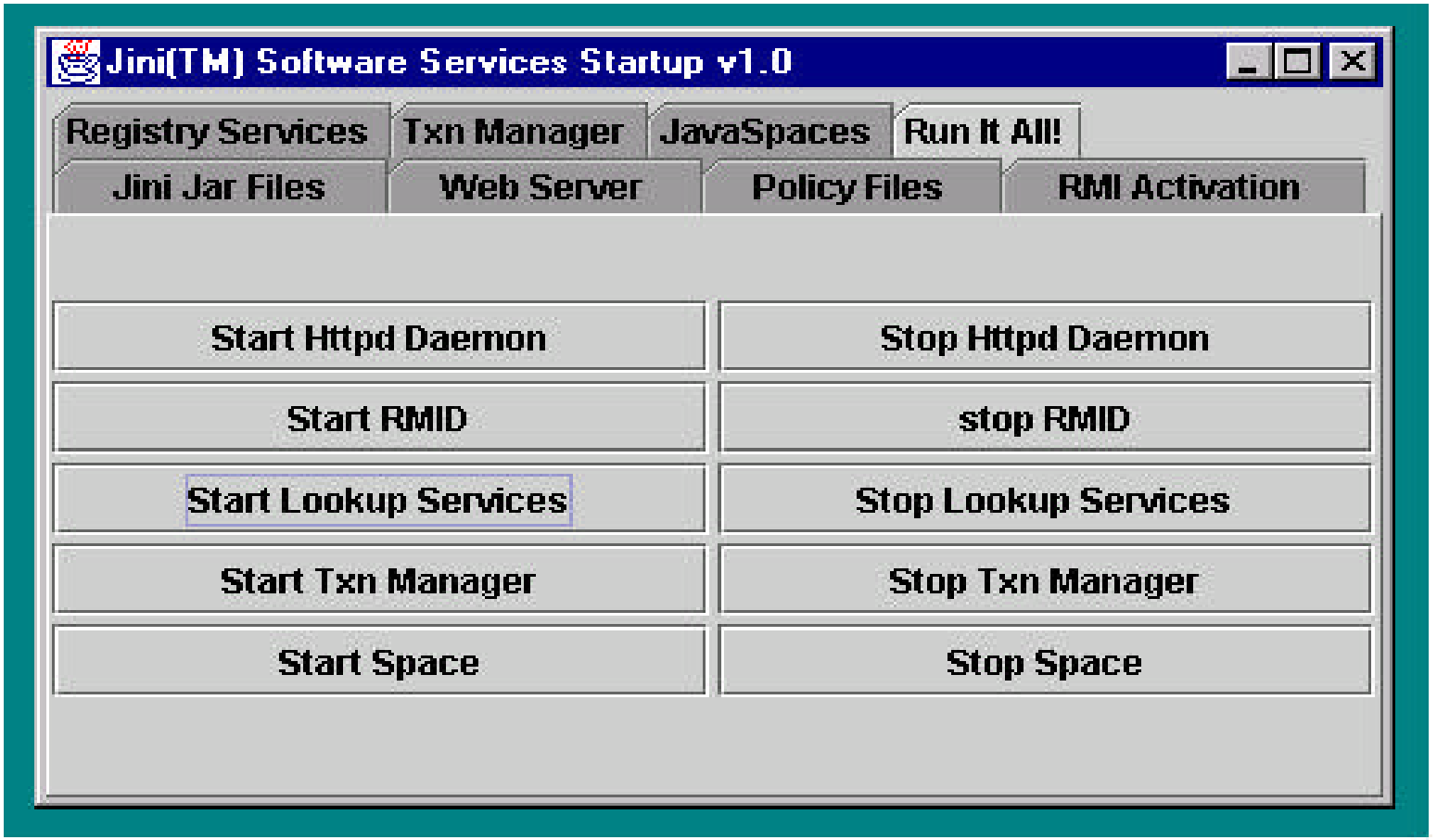


Execution Process in Prototype Six

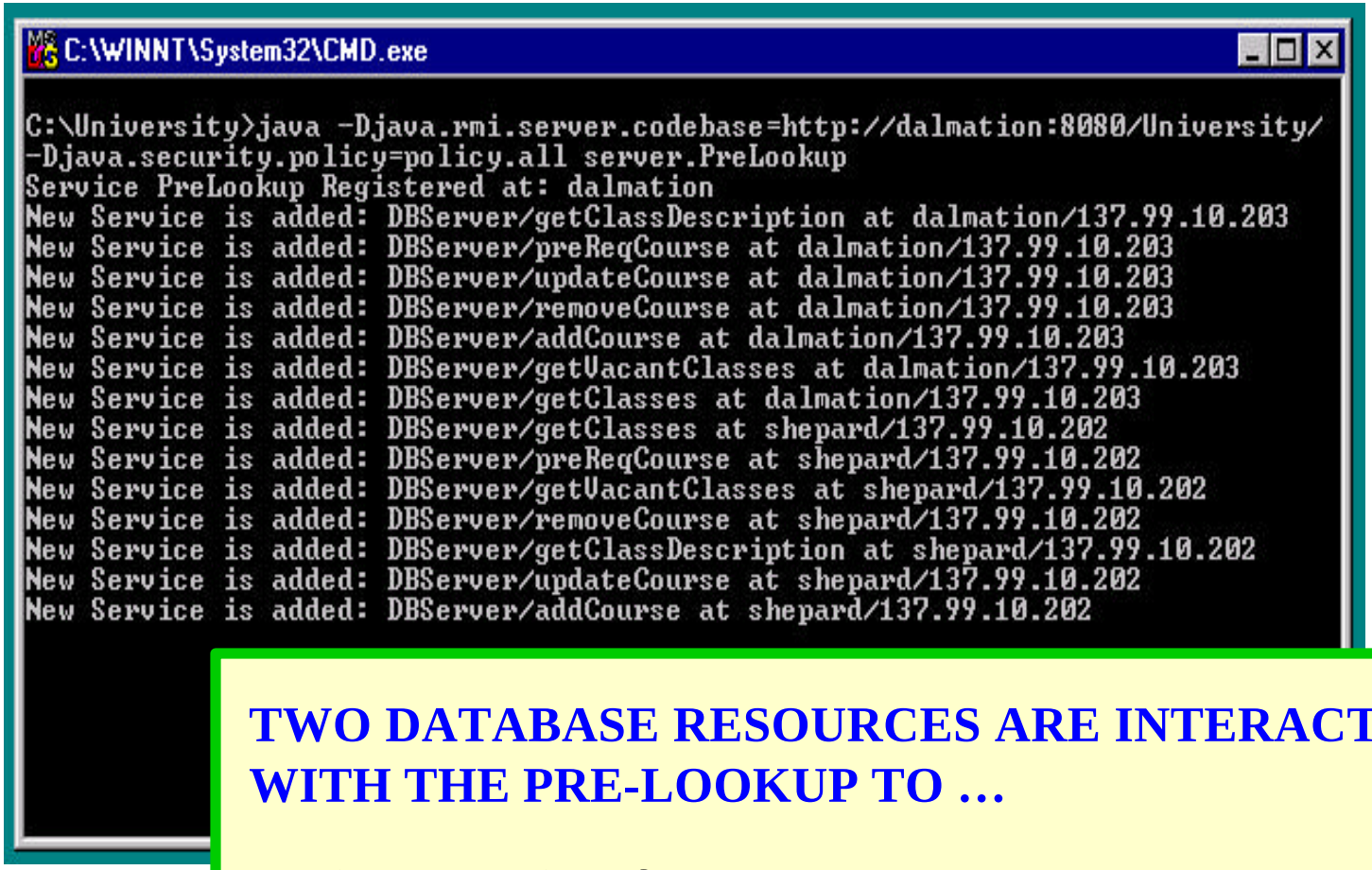


UCONN

Services GUI



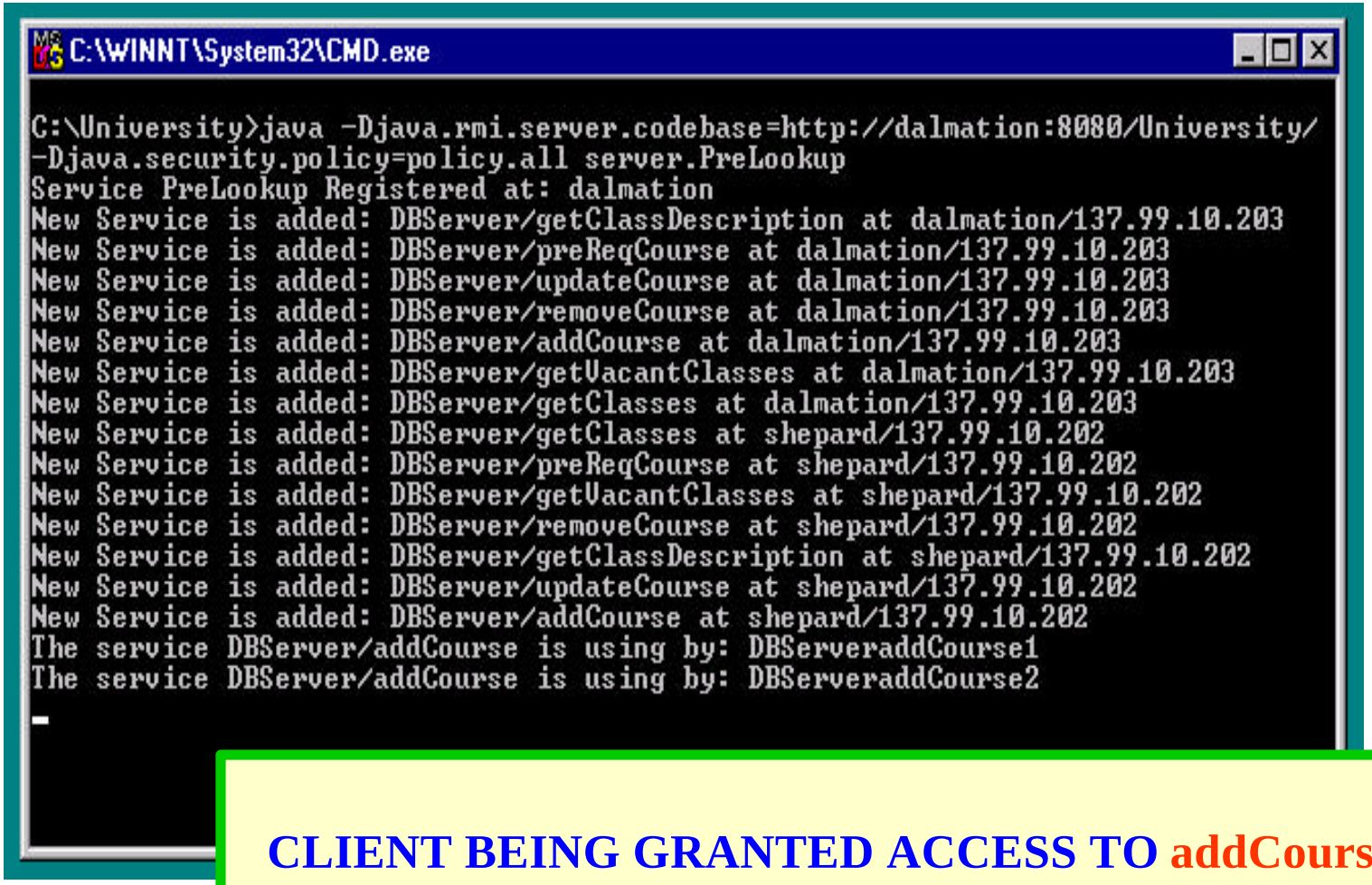
Pre-Lookup when Two Replicated Database Resources Register



TWO DATABASE RESOURCES ARE INTERACTING WITH THE PRE-LOOKUP TO …

Register Services from DALMATION & SHEPARD:
getClassDescription, preReqCourse, updateCourse, removeCourse, addCourse, getVacantClasses, getClasses

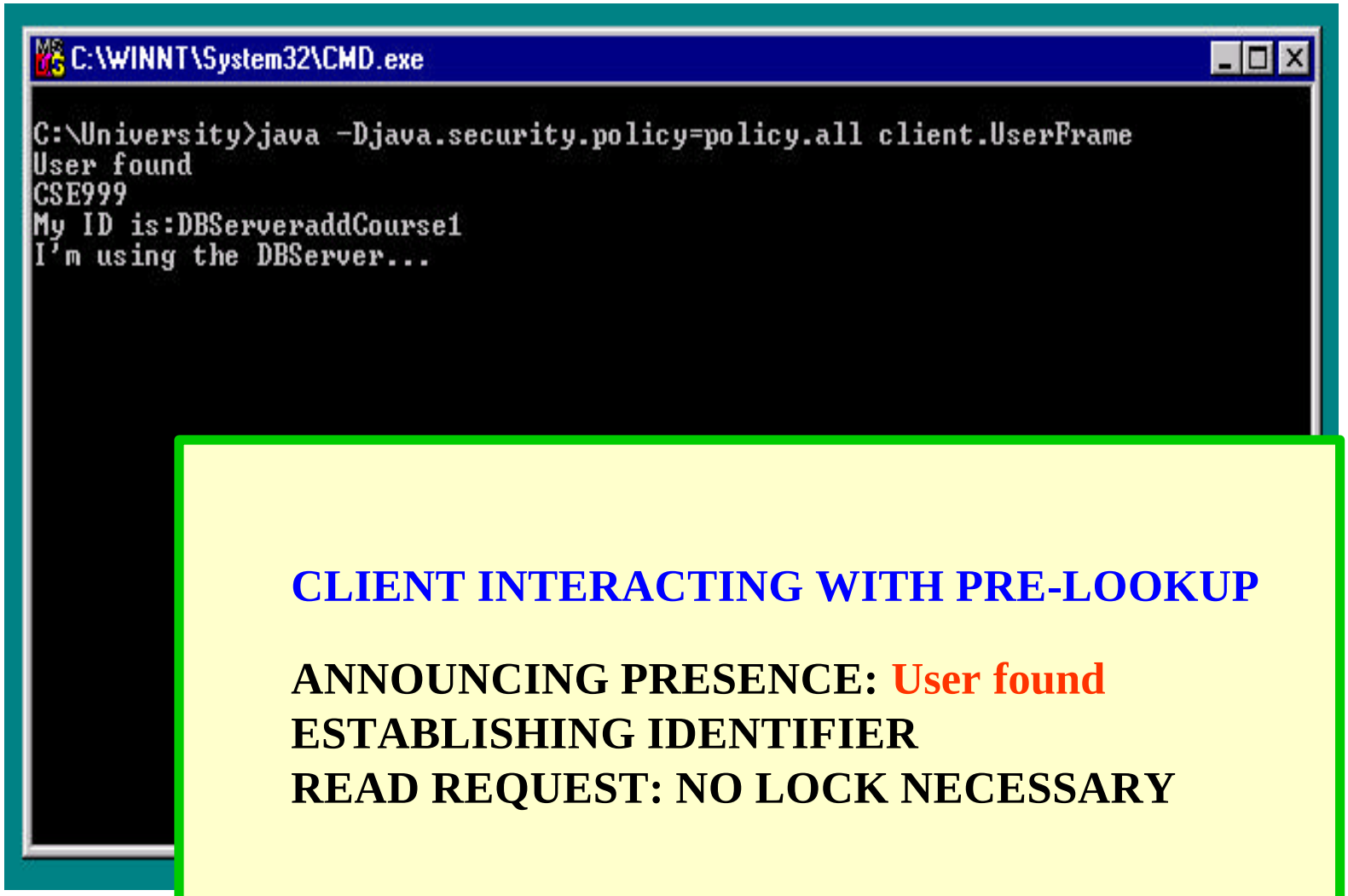
Pre-Lookup when Client Invokes "addCourse" Service



CLIENT BEING GRANTED ACCESS TO addCourse SERVICE ON DALMATION & SHEPARD

UCONN

Client Passed Through Pre-Lookup

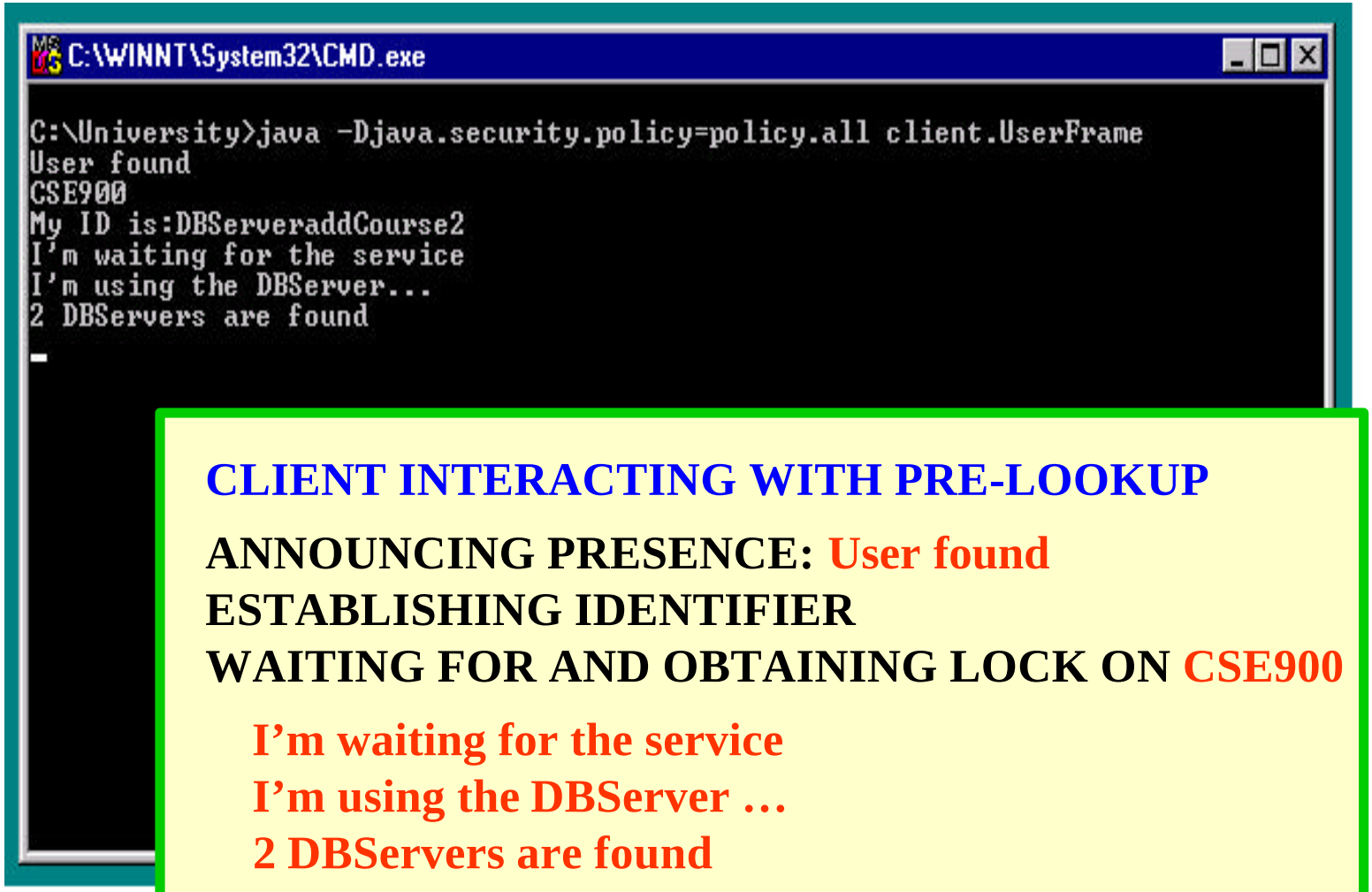


CLIENT INTERACTING WITH PRE-LOOKUP

ANNOUNCING PRESENCE: User found
ESTABLISHING IDENTIFIER
READ REQUEST: NO LOCK NECESSARY

UCONN

Client Locked by Pre-Lookup



CLIENT INTERACTING WITH PRE-LOOKUP

ANNOUNCING PRESENCE: User found
ESTABLISHING IDENTIFIER
WAITING FOR AND OBTAINING LOCK ON CSE900

I'm waiting for the service
I'm using the DBServer …
2 DBServers are found

UCONN

Results from Prototype Six

- Achievements:
  - **Any** Number of Clients Can Do the **Read** Operation on the Database
  - Only **One** Client Can **Modify** a Replicated Database Resource at a Given Point of Time
  - Pre-Lookup Service Can be Modified to Support other Locking Protocols
- Drawbacks:
  - **Can't Force Stakeholders to Consult Pre-Lookup Service - Rely on SW Practice**
  - No Redundant Lookup/Pre-Lookup Services
  - Failure of Lookup/Pre-Lookup - Total Restart
  - **No Logging and Recovery When Replicated Server Application Fails and is Restarted**

UCONN

Experimental Prototyping Effort Future Work Possibilities

- **Verification of Write-Once-Run Anywhere**
  - Extend to Win95, 98, NT, Solaris, and Other Databases (Sybase, Oracle, …)
  - **CRITICAL for Comprehensive Evaluation of JINI's Readiness for 21st Century!**
- **Explore Other JINI Capabilities**
  - Transactions/Two-Phase Commit and JINI Security Model for Granting/Denial of Services
  - **Response, Consistency, Authorization**
- **Software Agents and JINI**
  - Ongoing Project this Semester
  - **Can Technologies Work Together?**
- **Others Include (JavaSpaces, Performance, …)**

UCONN

Conclusions and Recommendations
Revisiting Goals of Research Effort

- Can JINI Support **Highly-Available Distributed Applications**?
  - PTs 4, 5 and 6 Demonstrate Ability to Architect Highly Available Solutions via JINI
- Can JINI Support an Environment with **Dynamic Clients and Replicated Services**?
  Will Clients Continue to Operate Effectively if **Replicated Databases Services Fail**?
  - Yes, PTs 4, 5, 6 All Support Starting and Stopping of Clients and Database Resources
- Can JINI be Utilized to **Maintain "minutes-off" Data Consistency** of Replicas?
  - PT 6 Superior Due to Pre-Lookup Service Guarantee of No Lost Updates

UCONN

Conclusions and Recommendations
Is JINI Ready for Prime Time???

- **Compatibility of JINI with Java**
  - Homogeneity of JINI and Java Unlike ORBs and IDL which are Heterogeneous
  - **But - Verify Across Heterogeneous HW/SW**
- **Track Record of Java and Sun**
  - Java's Increasing Dominance in Agents, E-Commerce, EJB, Data Mining, etc.
  - **But - Incompatibility of Java Releases and Danger of Evolving Technologies**
- **Understandability and Ease of Use of JINI**
  - 400 Hours Total for Obtaining Familiarity with JINI, Visual Café, JDBC/ODBC, 6 PTs, etc.
  - Faster Speed Up w.r.t. CORBA/ORB

UCONN

Conclusions and Recommendations
Important Caveats

- **Future Work Clearly Indicates that More Effort is Required to Tilt Scales Toward JINI**
  - Work Extensive w.r.t. JINI Technologies
  - Potential Role of JavaSpaces
- **Experiences on Whole Positive**
  - Leasing Issues Need to be Explored
  - Impact of Registry Currency on Clients
- **Continued Evolution of JINI Specification**
  - Will JINI 1.1 be Compatible with JINI 1.0?
  - Java has Had Compatibility Problems in Past (Deprecated APIs)
- **Conclusion: JINI Great Promise as a Successful Technology in 21st Century!**