

COMP 212

Distributed Systems

Lecture 27

Boris Konev

Dept Computer Science

konev@liverpool.ac.uk

Distributed Coordination-Based Systems

New Look at Distributed Systems

Various components of a system are inherently distributed and independent; the real problem is

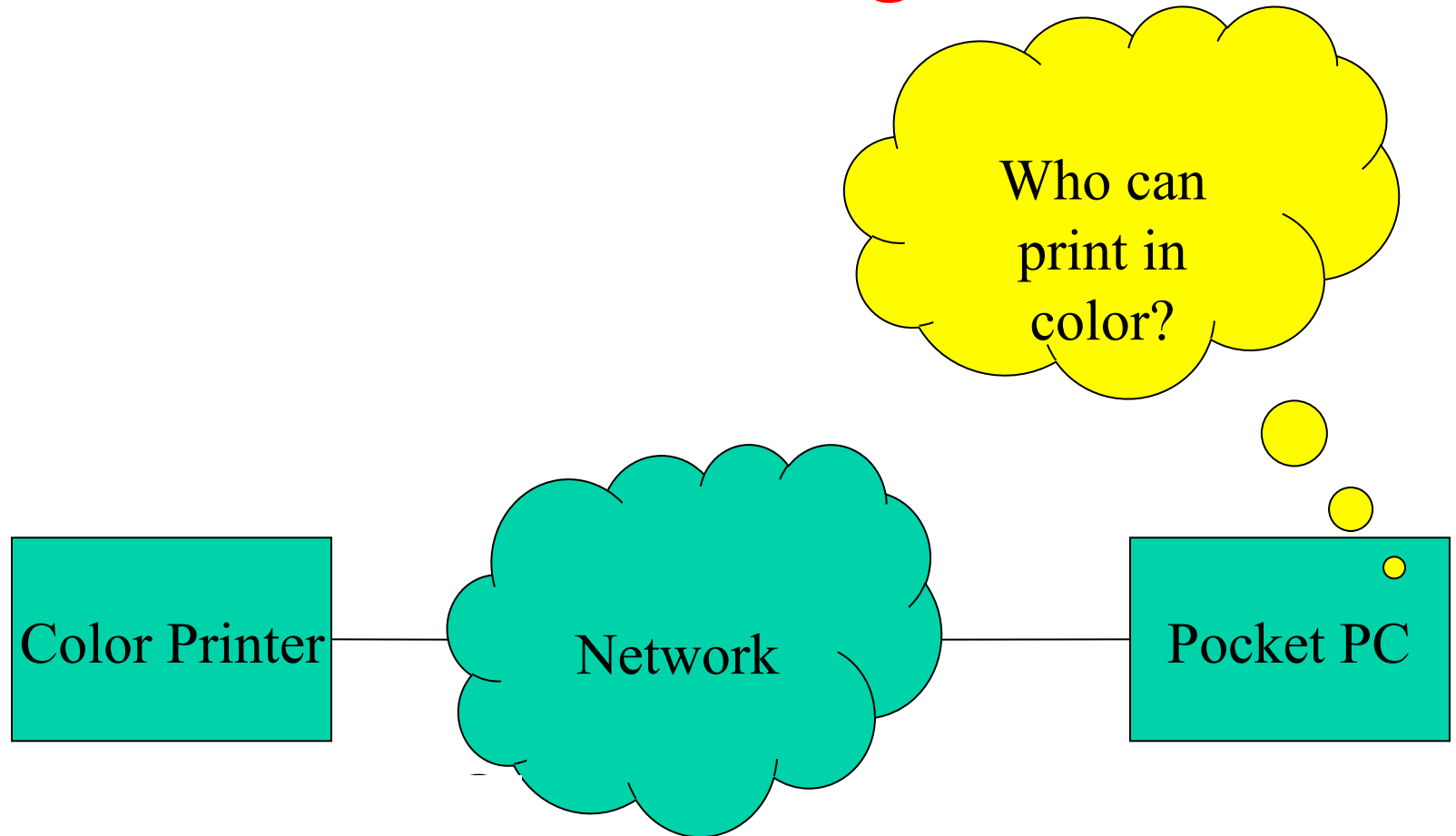
coordination of the activities of different components

- Separate between *computation* and *coordination*
 - Coordination is the part of a distributed system which handles *all the communication and cooperation between processes*

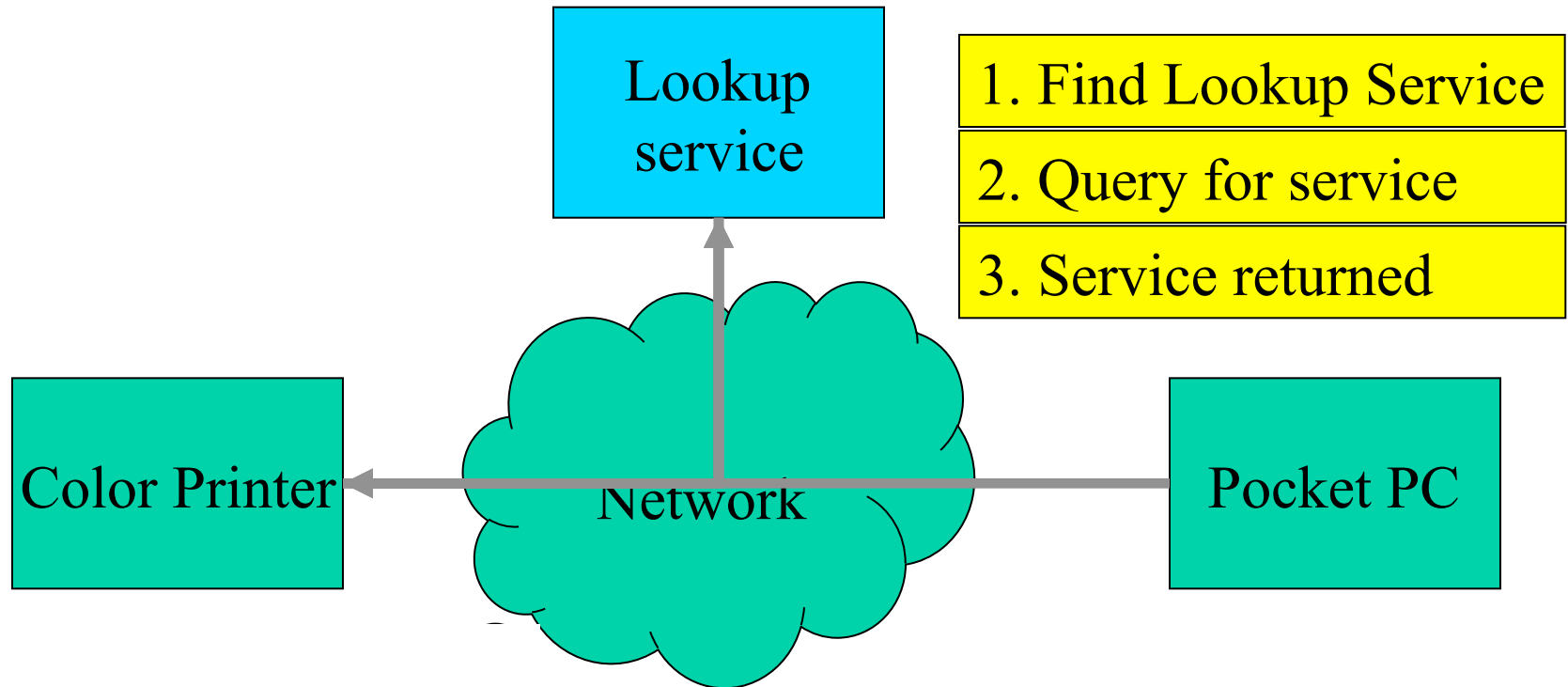
Jini

- Coordination-based (not only) system from Sun Microsystems.
- Strongly related to Java
 - One language everywhere
- Motivation:
 - Why can't services and devices enter and leave the network without reconfiguration?
 - Network plug-and-play
 - Example scenario: “Find all nearby color printers”

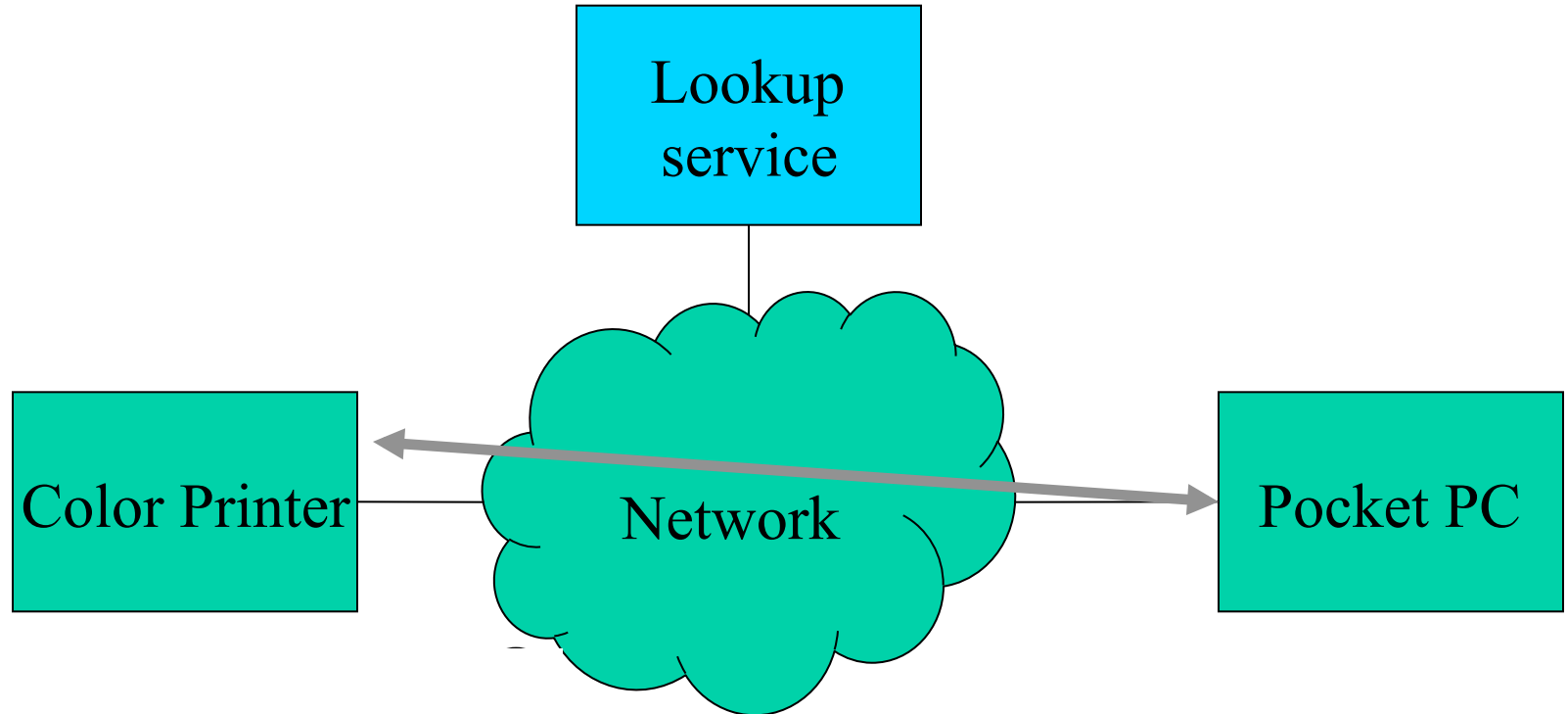
Problem of Finding Services



Lookup Service



Using Services

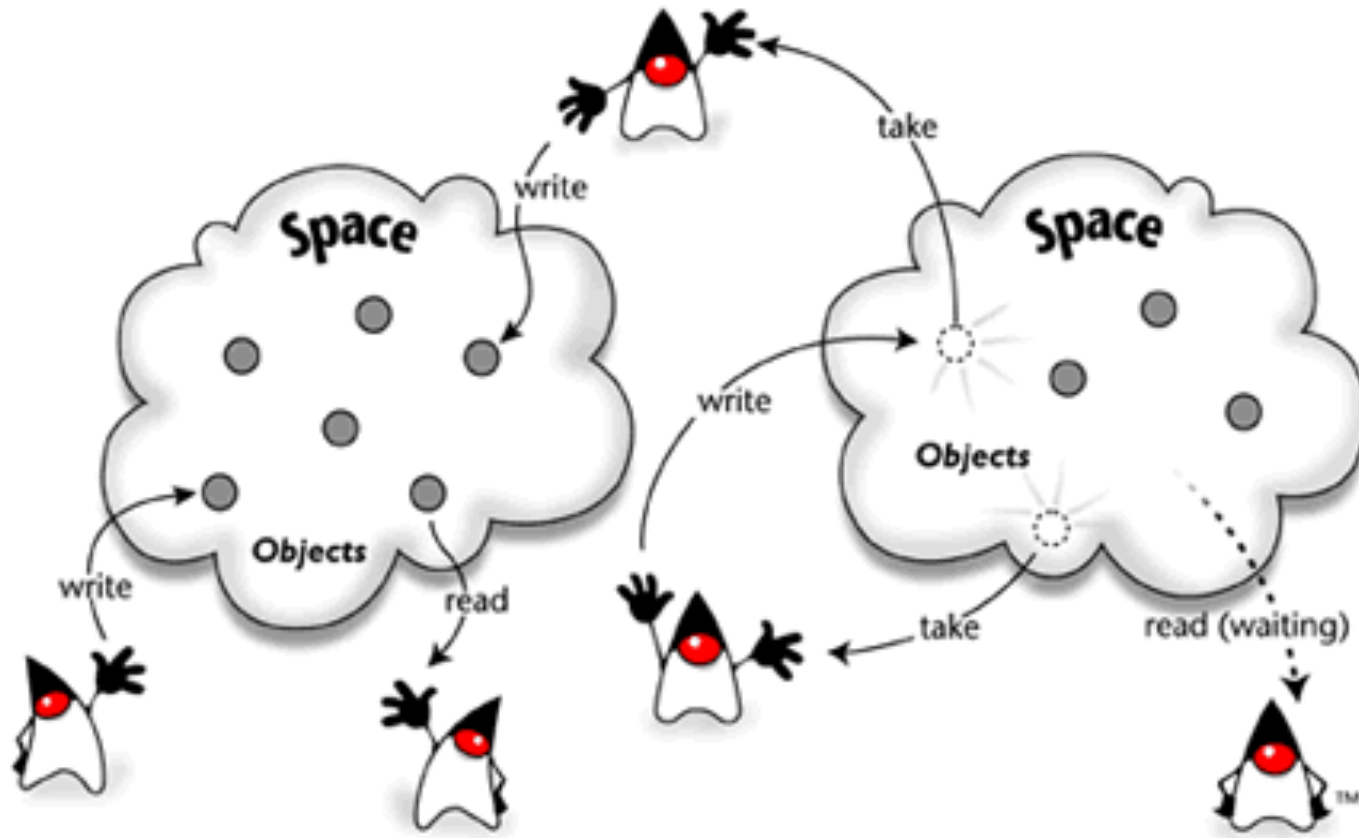


Can use any protocol to communicate to service

Basic Components Jini

- Lookup
 - Providing a central registry of ...
- ...Services
 - Entity that can be used, a device or another service
 - Discover the lookup service
 - Join the environment
 - Search for needed components
 - Flexible search on properties

Distributed Data Space

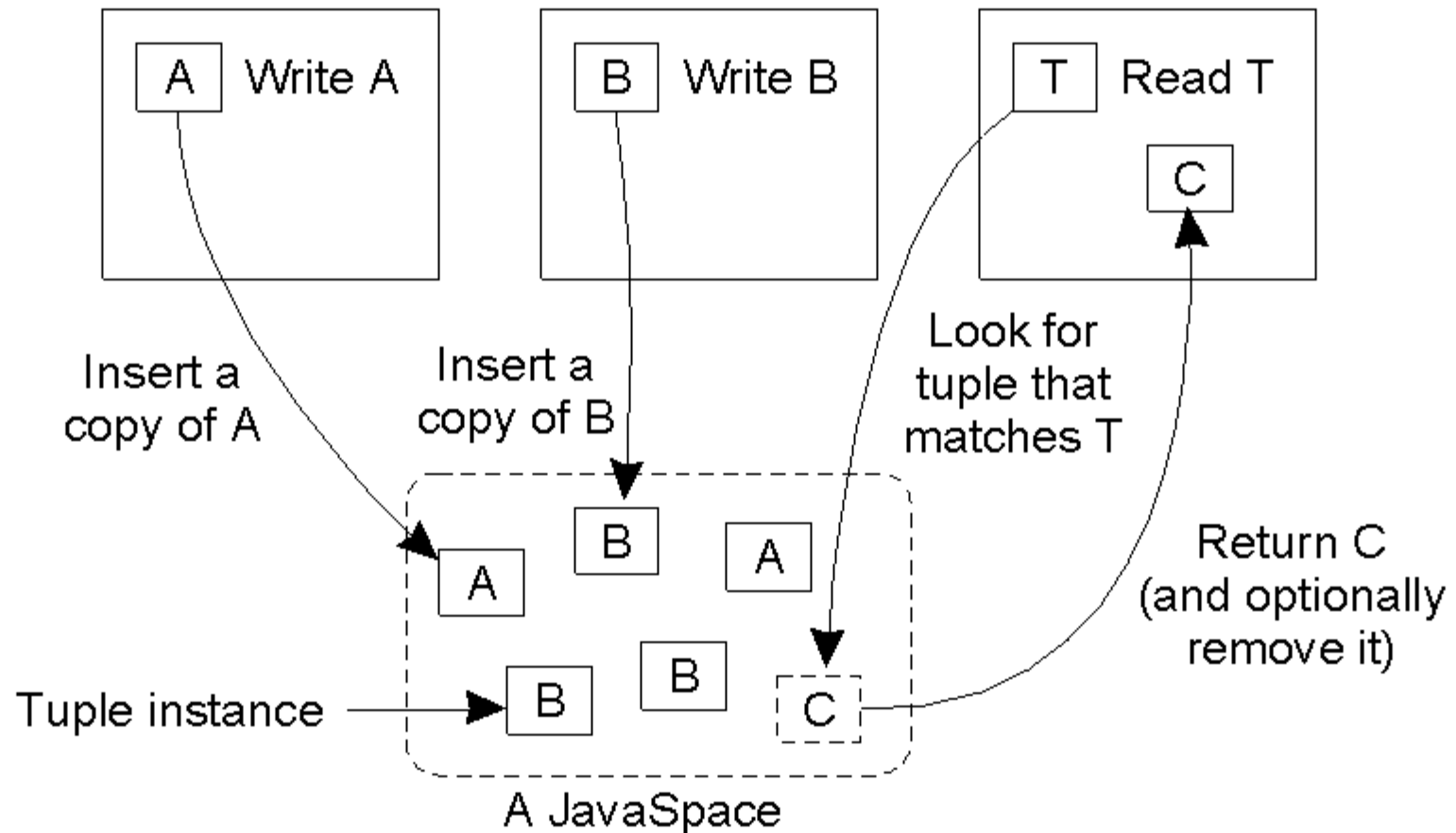


Clients and services have to exchange information and *coordinate* their activity

JavaSpaces

- Supports communication as a service called *JavaSpaces*
 - Shared dataspace that stores *tuples* representing a typed set of references to Java objects
 - No agreement on the structure of tuples
 - To *read* a tuple, a process provides a *template* for matching tuples as stored in a JavaSpace. A template is just another tuple with “wild characters”
 - *Processes that use Jini need not coexist at the same time*

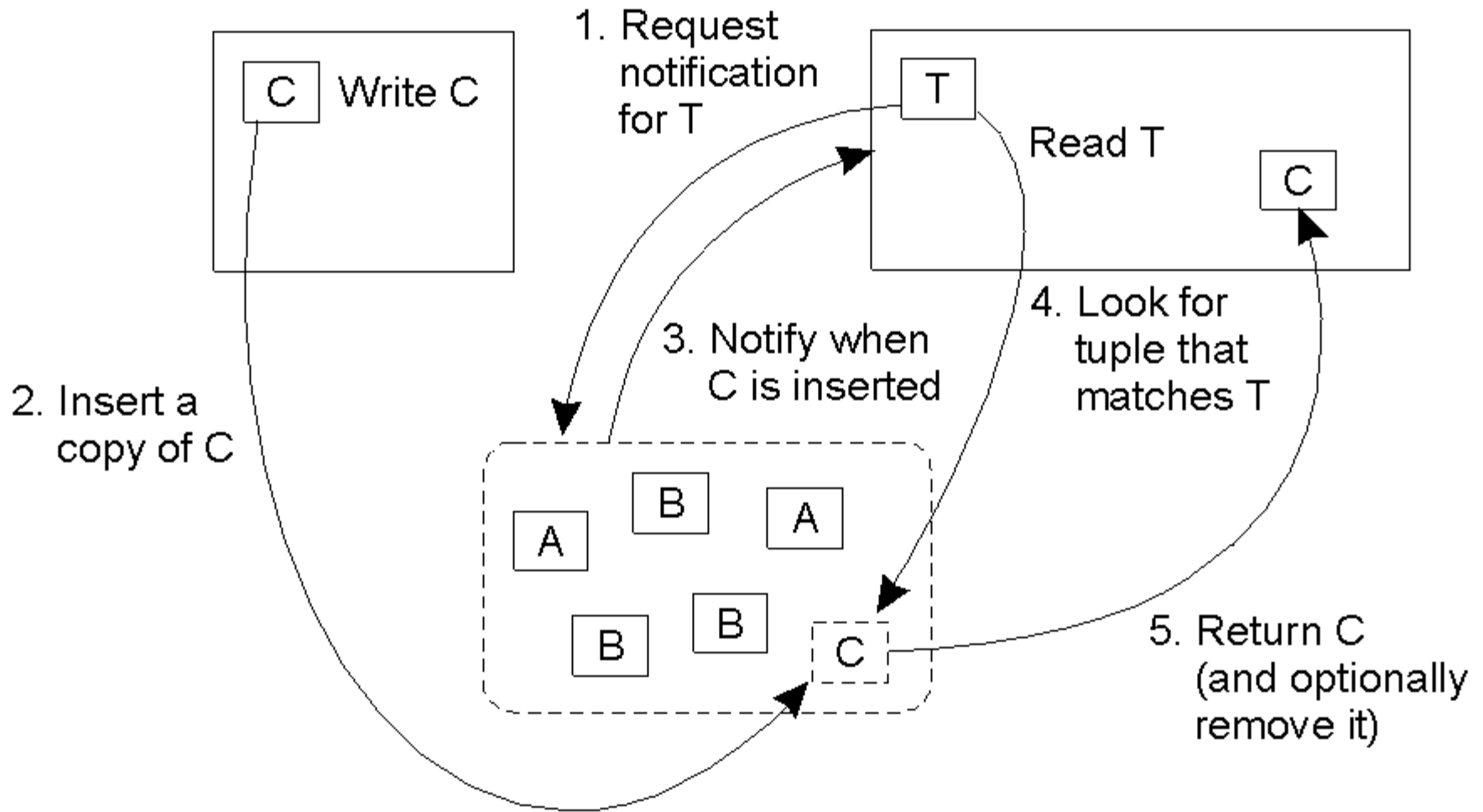
General Organization of a JavaSpace



Communication in Jini

- Based on Java RMI
- Provides simple event and notification subsystem
 - A client can register itself with an object
 - The object will notify the client if an event occurs
 - Based on RMI

Events and JavaSpaces



Using events in combination with a JavaSpace

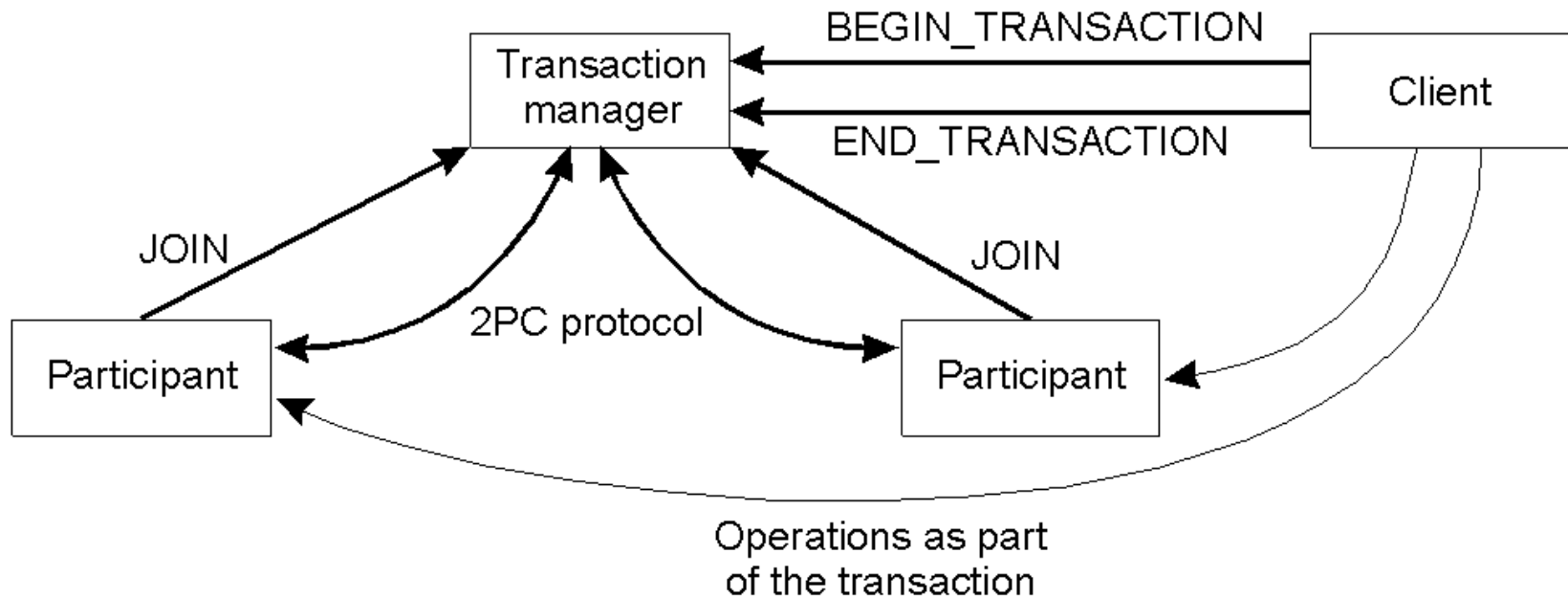
JavaSpaces Issues

- Scalability problems
 - Multicasting is needed to add/delete tuples
 - No efficient wide-area implementation of tuple spaces
- Software optimizations
 - Hashes, distributing the dataspace over several computers
- Hardware optimizations
 - Multiprocessor/multicomputer techniques

Naming: Lookup Service

- Lookup service
 - Allows clients to look up new services as they come available
 - Part of the lowest-level infrastructure level
 - Services are described in terms of attributes grouped in tuples
 - The lookup service selects those tuples that match a template
 - Multicasting is used to locate the lookup server (causing the scalability problems again)

Synchronization of Transactions



The support is given in the form of a set of interfaces

- ACID properties are *not* provided by Jini itself
 - Jointly implemented by various processes that take part in a transaction
- Default transaction manager is provided

Caching and Replication

No special means to provide caching or replication

Left to the application

Fault Tolerance

No additional support for fault tolerance except for a transaction manager

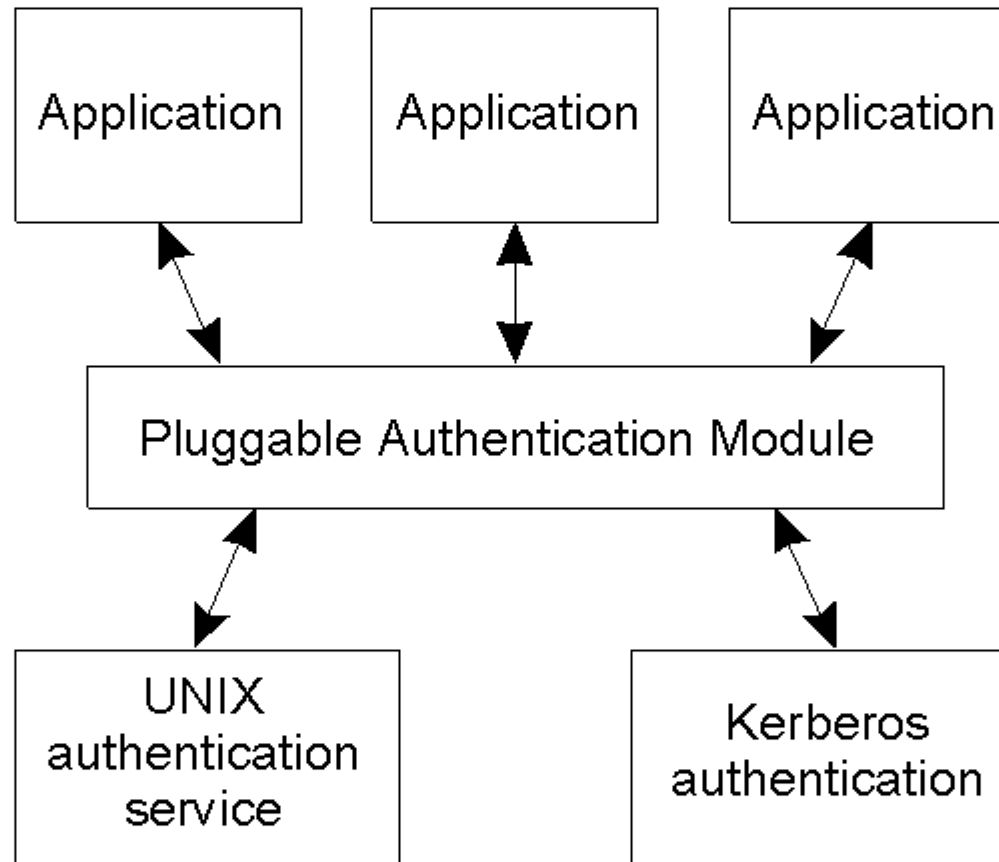
Applications should implement their own fault-tolerance measures as needed

All communication is done by means of Java RMI which itself is implemented using a **reliable** lower-level communication protocol (HTTP or TCP)

Security

- Security provided by Java RMI
 - Class-based access control
- Java Authentication and Authorization Service (JAAS)
 - Handles *user* authentication and authorization
 - Separates the interface and implementation by means of the *Pluggable Authentication Module (PAM)*

PAM and Security Services



The position of PAM with respect to security services.

Further Reading

- Jini Specification

<http://www.sun.com/jini/specs>

- Jini Developers

<http://www.jini.org>

- Apache River

<http://incubator.apache.org/river/>

Jini can be downloaded for free