

Computer Science 5204

Fall, 2005

Dr. Dennis Kafura

Operating Systems

Course Overview

Major Topics

- 1. Interaction**
- 2. Security**
- 3. Ordering**
- 4. Fault Tolerance**
- 5. Naming and Data**

1. Interaction

How do computational entities interact?

What are alternative syntaxes and semantics for interactions among computational entities?

On what basis is the communication structured?

What are example technologies?

How can processing be structured on a single processor?

Why is mobility of processing desirable and how can it be achieved?

How can a workload be distributed across multiple systems?

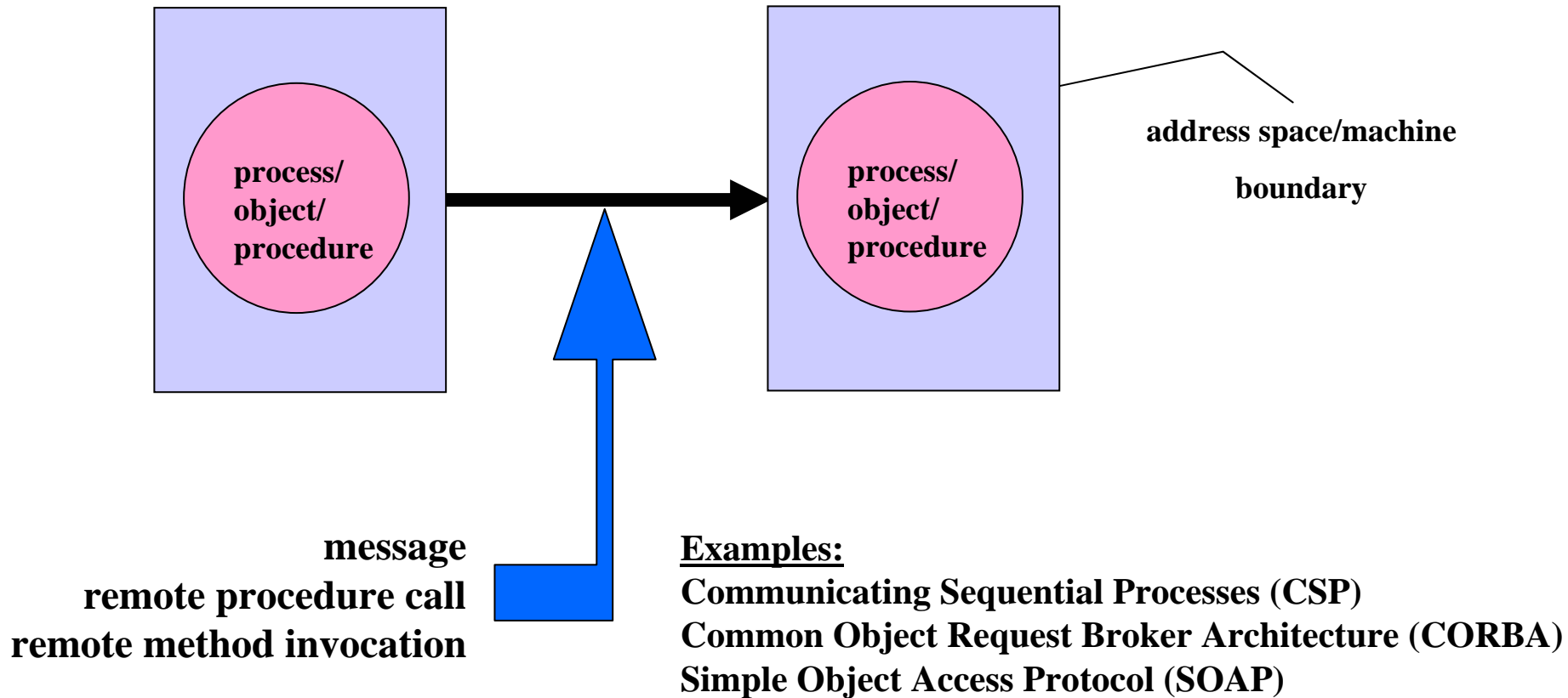
Interaction Forms

Temporal

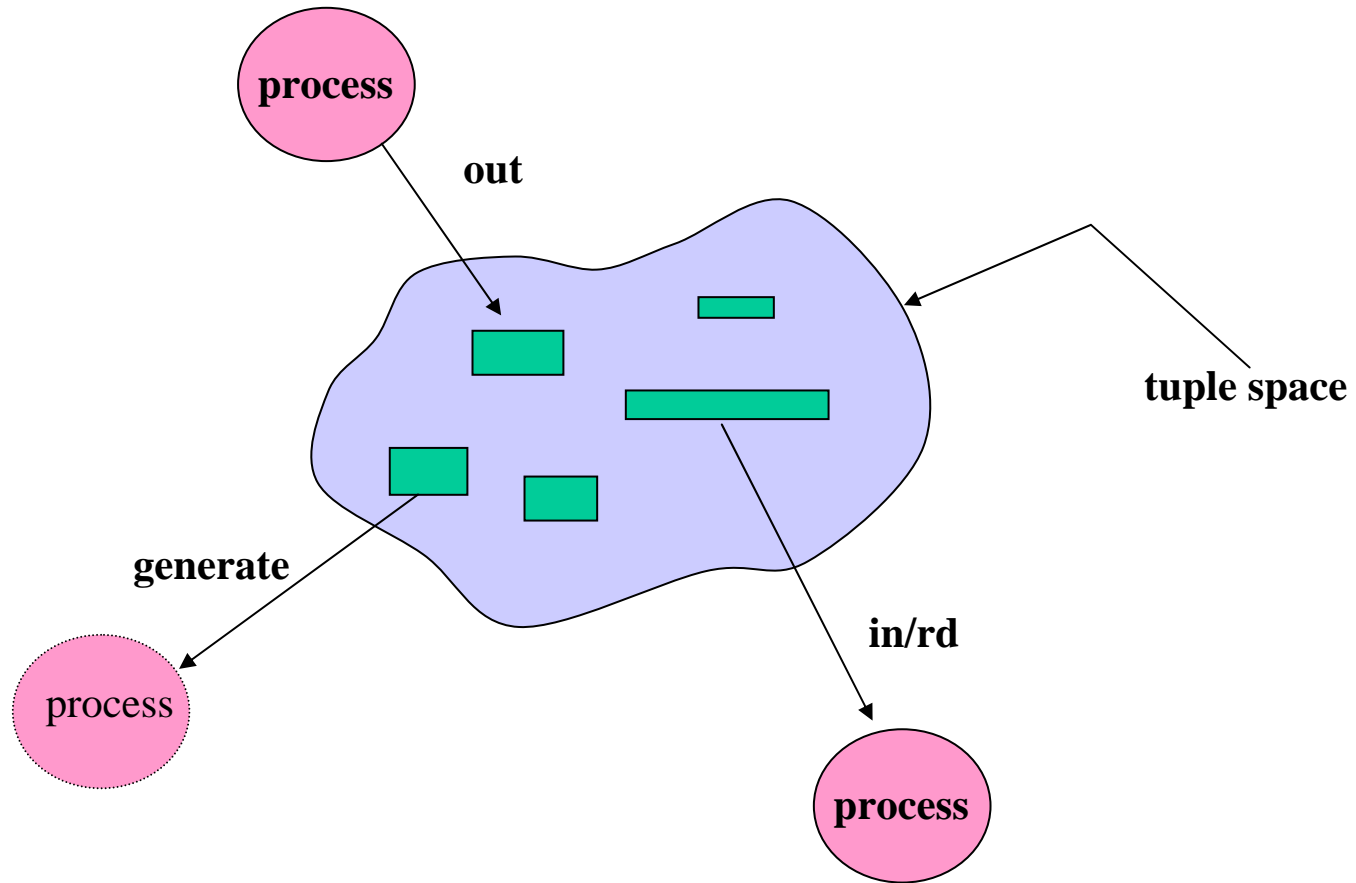
Referential

	Coupled/ Synchronous	Uncoupled/ Asynchronous
Coupled/ Identity-based	RPC RMI	messaging
Uncoupled/ Content-based	Events broadcasting	Tuple spaces

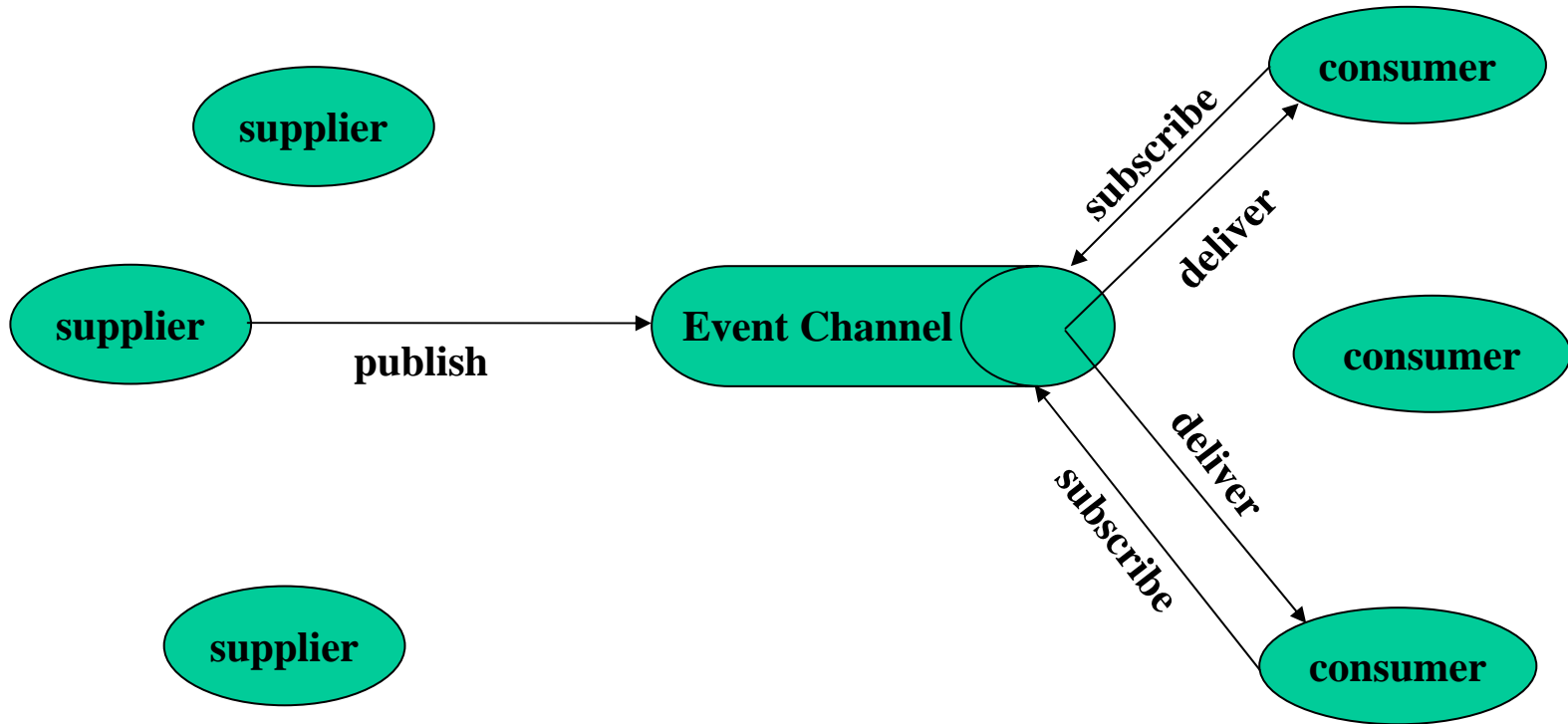
Identity-Based Communication



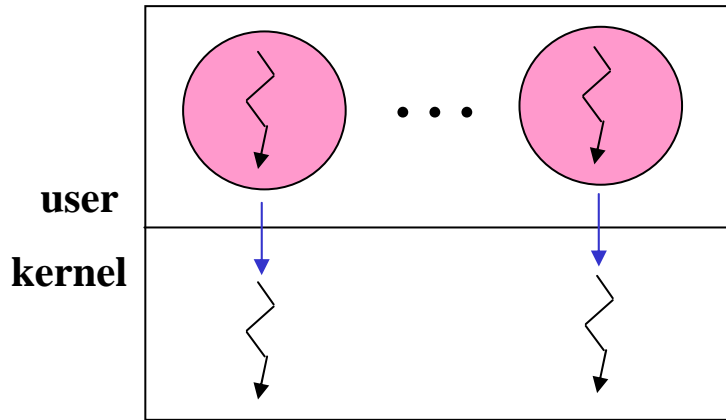
Content-Based Communication: Tuple Space Model



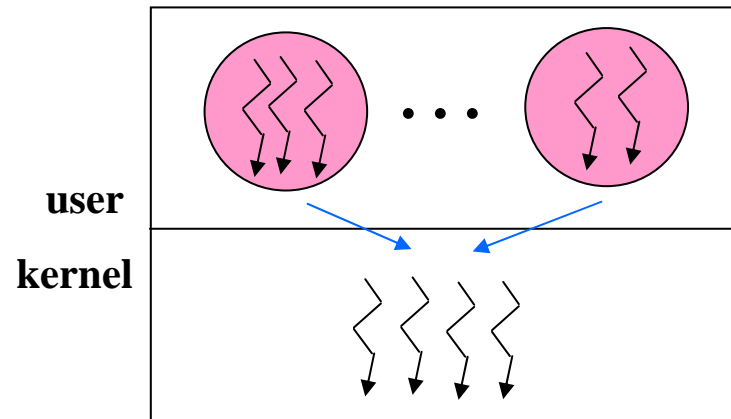
Content-Based Communication: Events



Structuring Activities

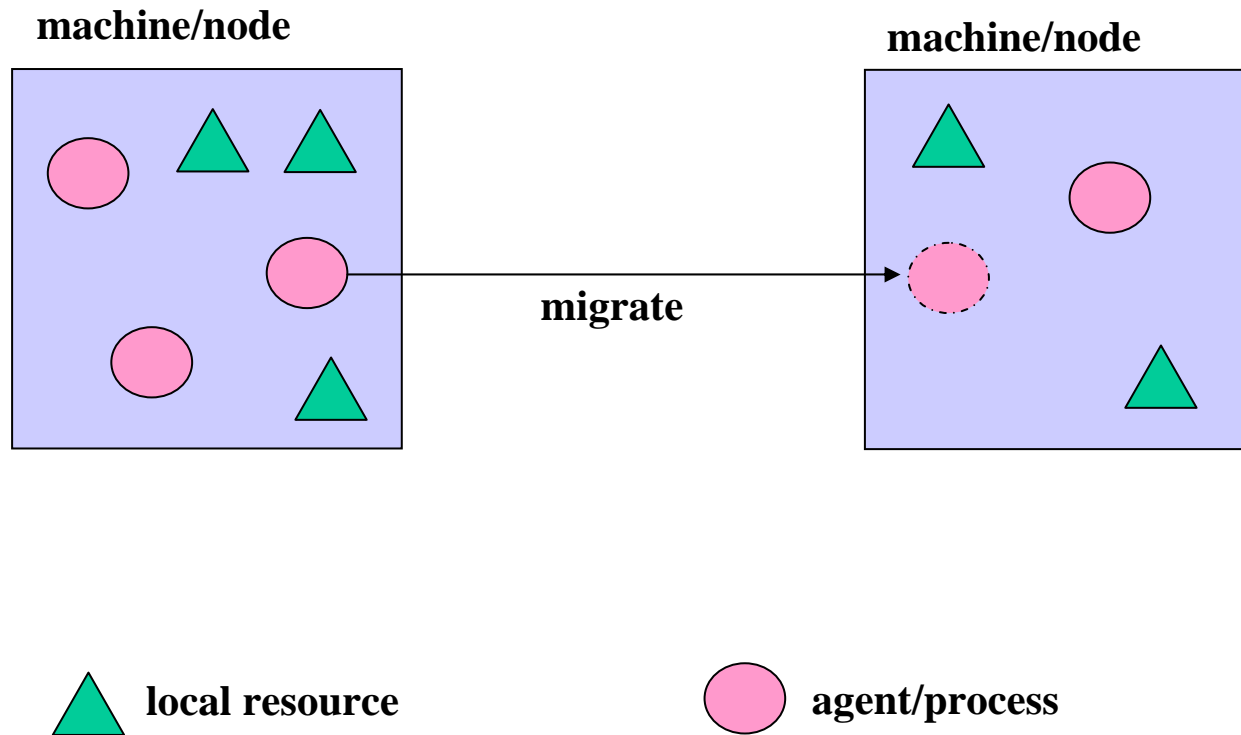


process-centered



thread-centered

Mobile Agents



2. Security

How can rights for access control be structured for effective use and management?

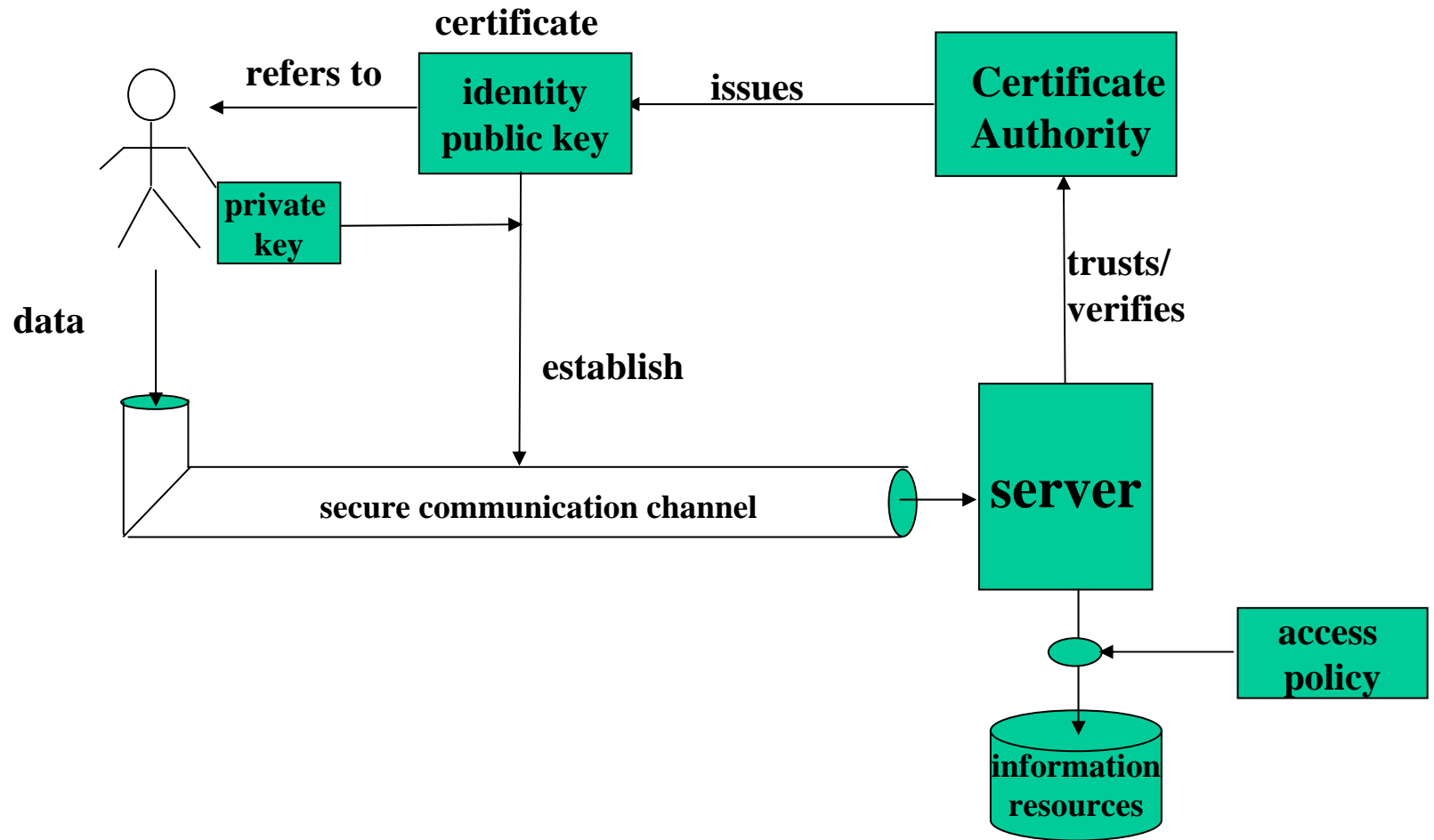
How can a digital document be “signed” so as to identify authorship?

How can communicating parties be confident of each other’s identities?

How can distributed systems authenticate clients and servers to each other?

How can access policies be expressed and enforced?

Security Overview



3. Ordering

How can events be ordered in a distributed system lacking a shared clock?

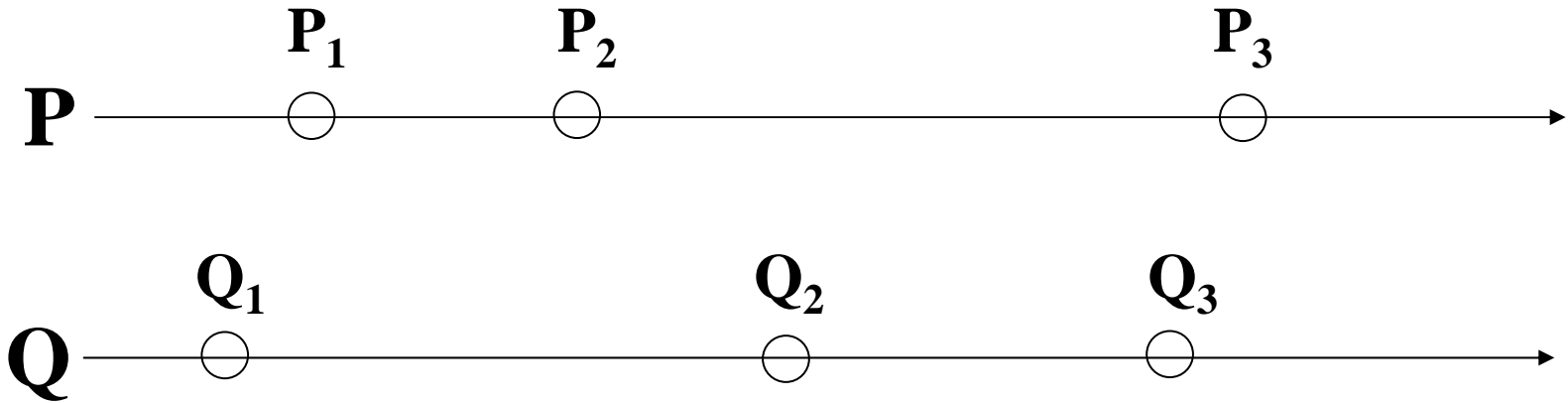
Can this ordering give rise to a form of virtual time?

How can the state of a distributed system be recorded?

What order must be imposed on the accesses to distributed data to preserve semantic integrity?

How can this order be enforced?

Event Ordering

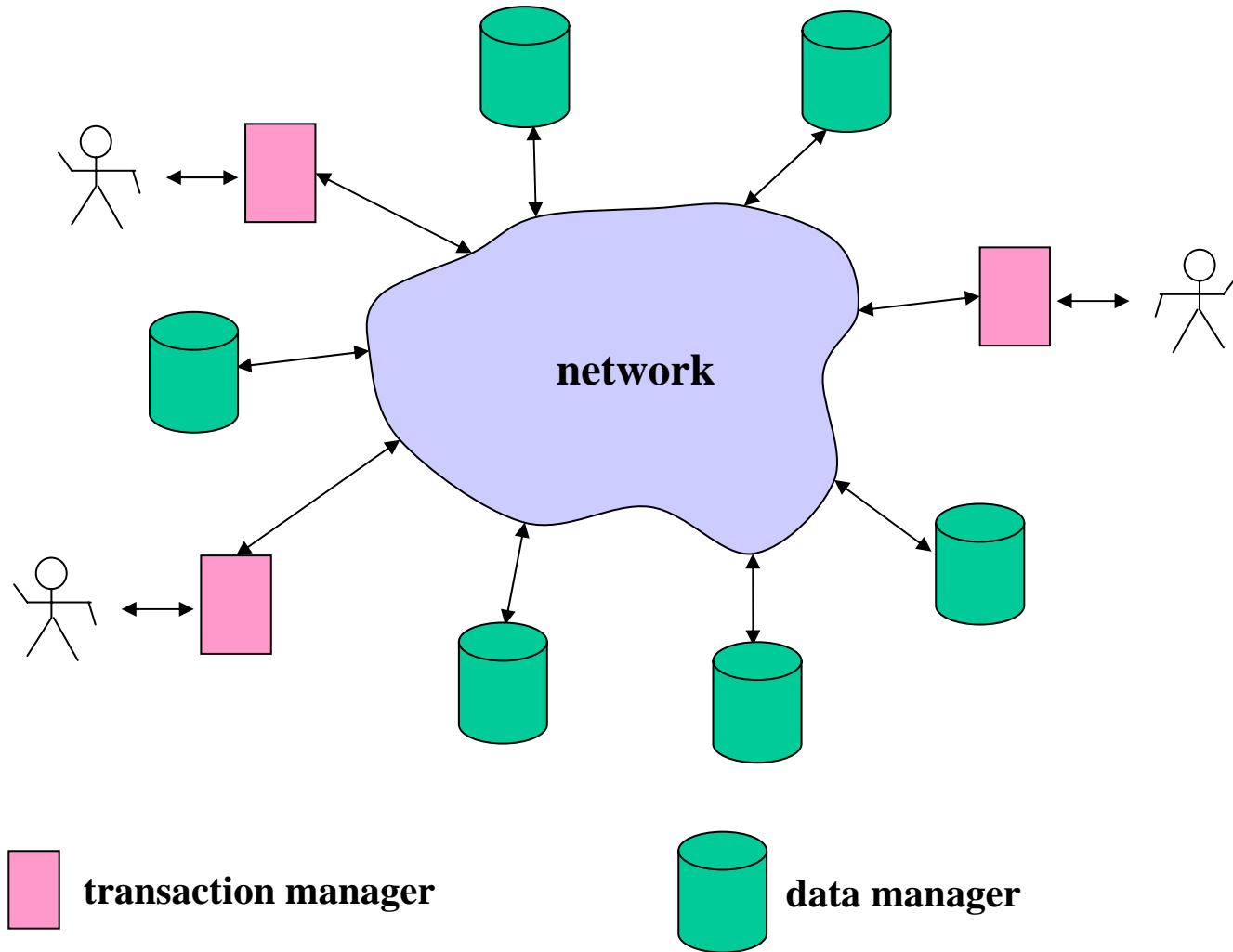


How can the events on P be related to the events on Q?

Which events of P “happened before” which events of Q?

When does it matter how we answer these questions?

Transaction Model



4. Fault Tolerance

What system objectives can be achieved by replicating data?

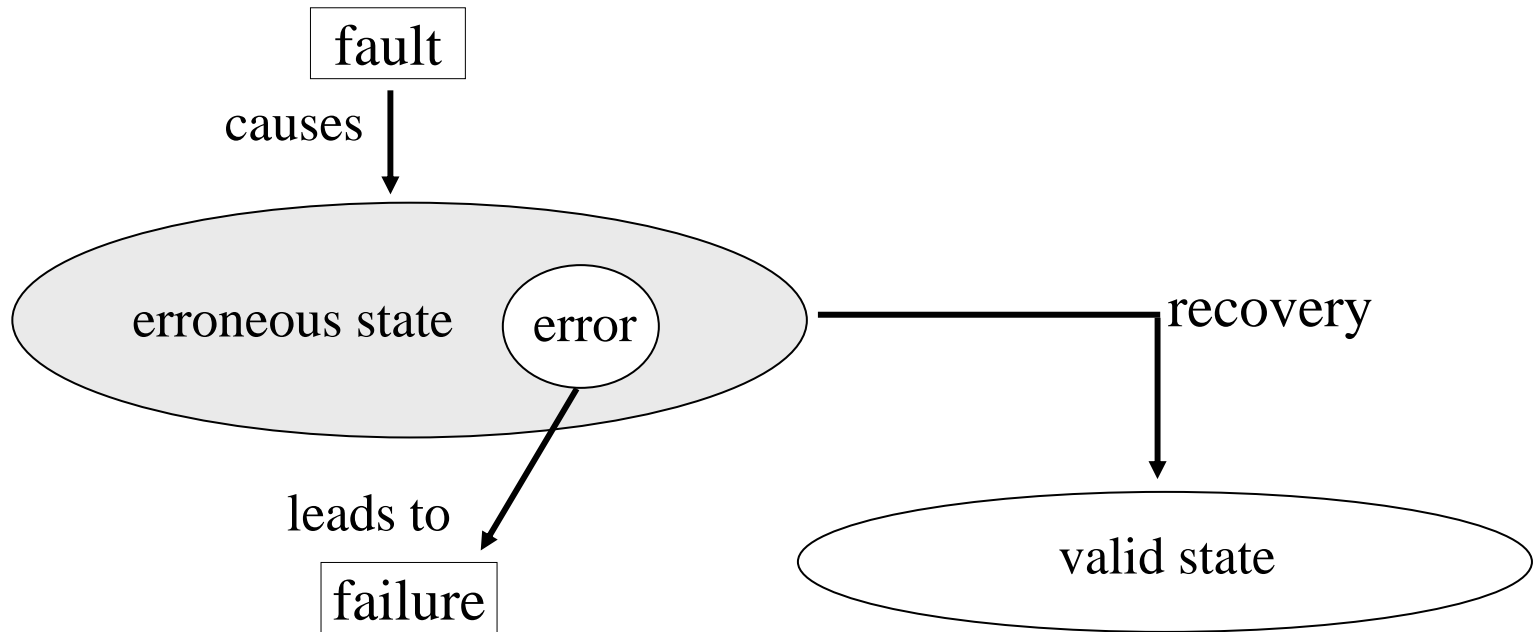
What are different ways in which replicated data can be accessed in a meaningful way?

What protocols enforce a meaningful sequence of accesses to replicated data?

How can concurrent activities reach agreement in the presence of (limited) failures?

How can a system recover from failure?

Recovery



An error is a manifestation of a fault that can lead to a failure.

Failure Recovery:

- backward recovery
 - operation-based (do-undo-redo logs)
 - state-based (checkpoints)
- forward recovery

5. Naming and Data

How are name spaces organized and managed?

How are distributed file systems organized and managed?

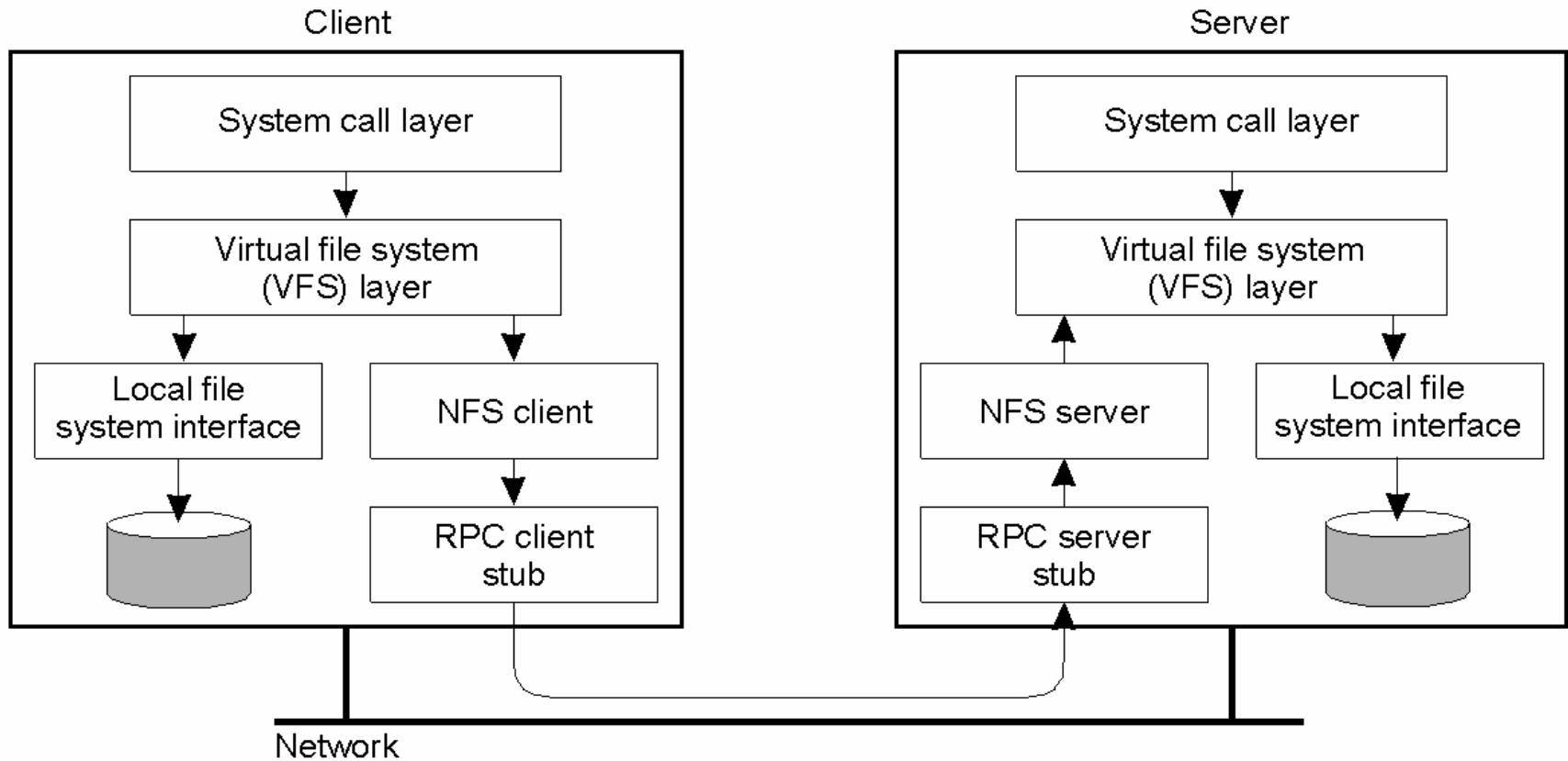
Where are the files stored while they are being operated on?

How can files/file trees be shared?

How are peer-to-peer (P2P) file systems organized?

How can anonymity be preserved in a distributed file system?

NFS Architecture



Overview of Coda

