Distributed Coordination-Based Systems

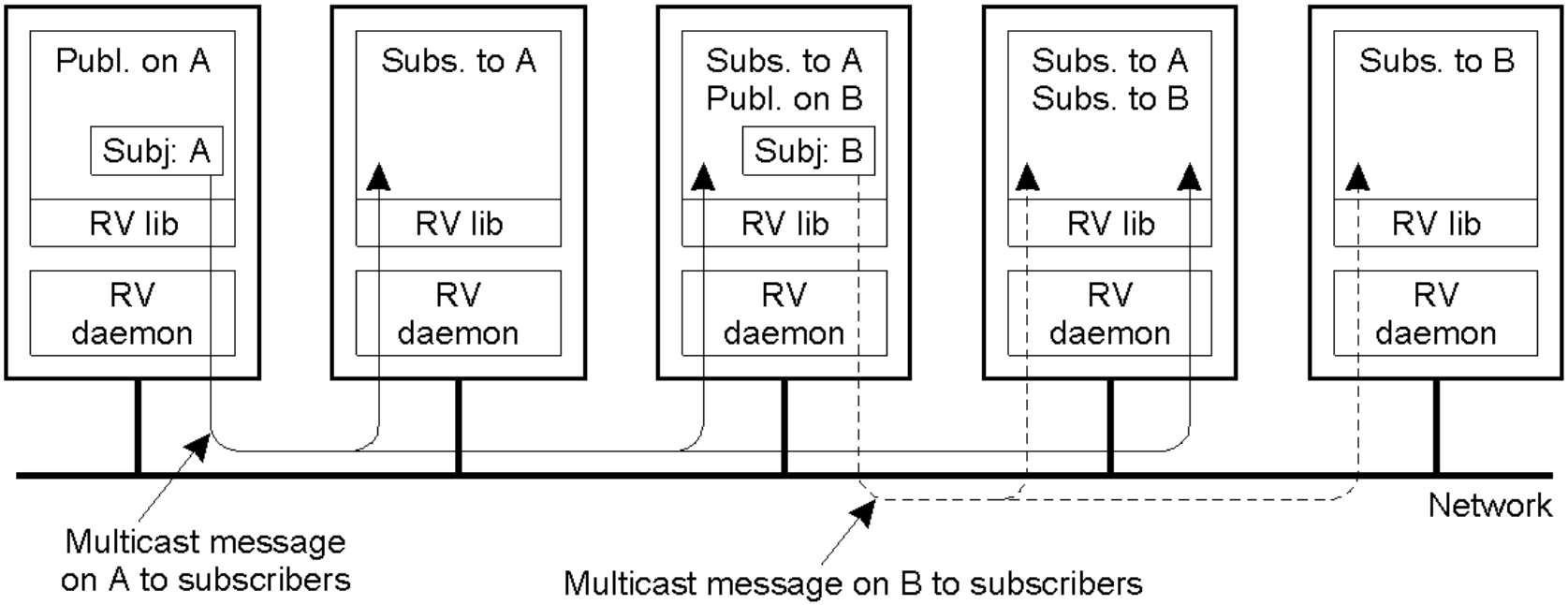
Chapter 12

Introduction to Coordination Models

Referential \ Temporal	Coupled	Uncoupled
Coupled	Direct	Mailbox
Uncoupled	Meeting oriented	Generative communication

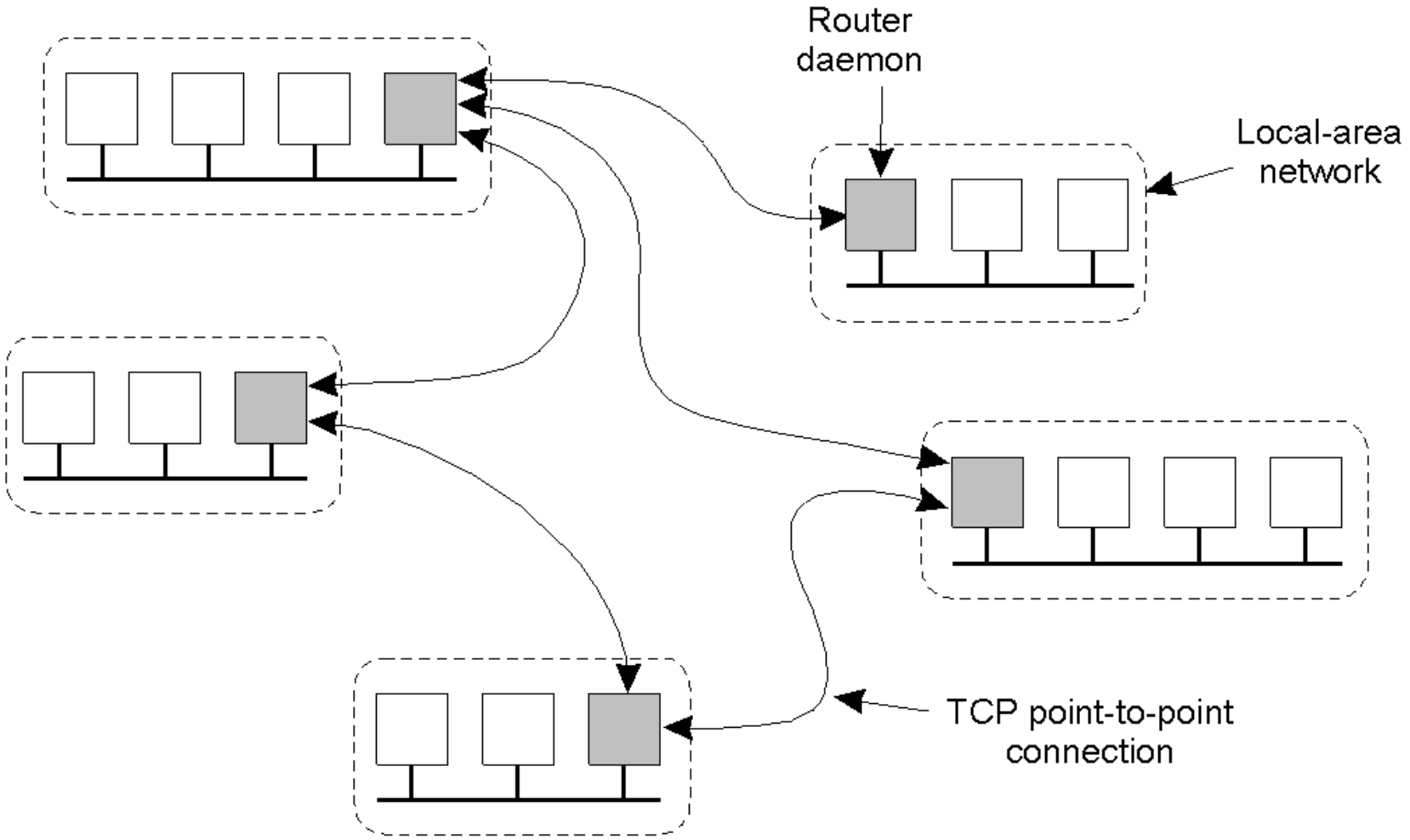
A taxonomy of coordination models (adapted from [cabri.g2000])

Coordination Model (1)



The principle of a publish/subscribe system as implemented in TIB/Rendezvous.

Coordination Model (2)



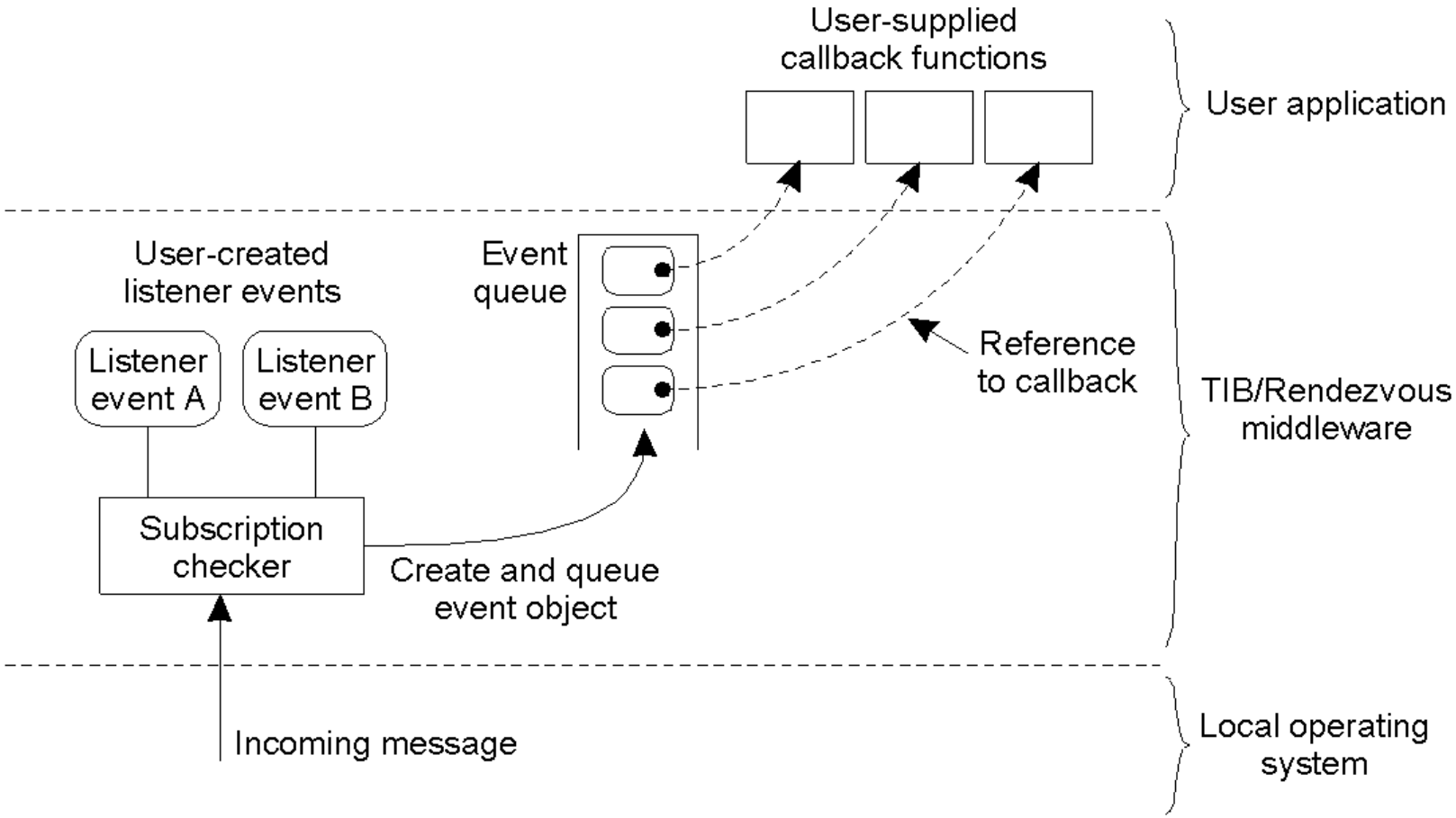
The overall architecture of a wide-area TIB/Rendezvous system.

Basic Messaging

Attribute	Type	Description
Name	String	The name of the field, possibly NULL
ID	Integer	A message-unique field identifier
Size	Integer	The total size of the field (in bytes)
Count	Integer	The number of elements in the case of an array
Type	Constant	A constant indicating the type of data
Data	Any type	The actual data stored in a field

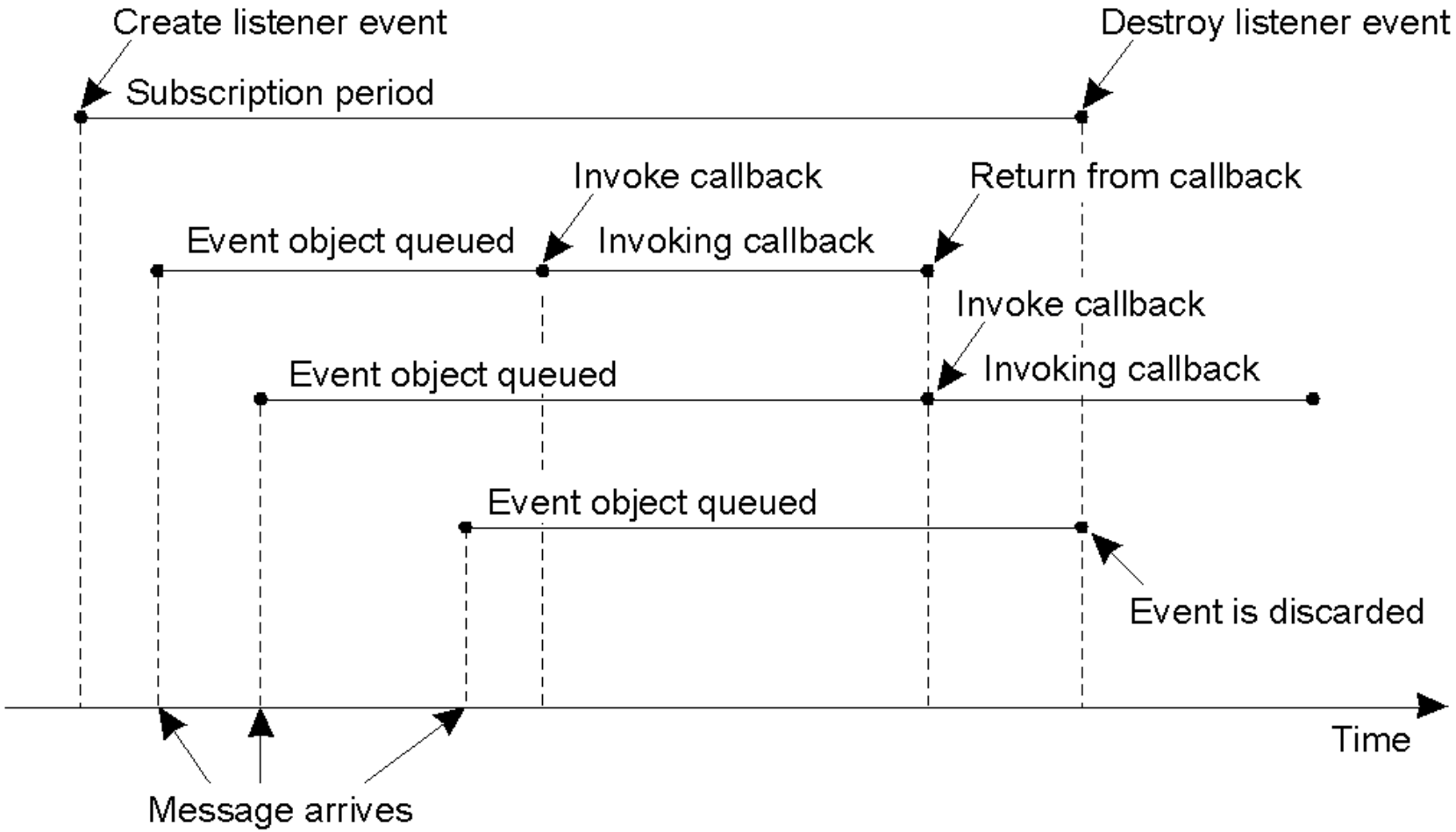
Attributes of a TIB/Rendezvous message field.

Events (1)



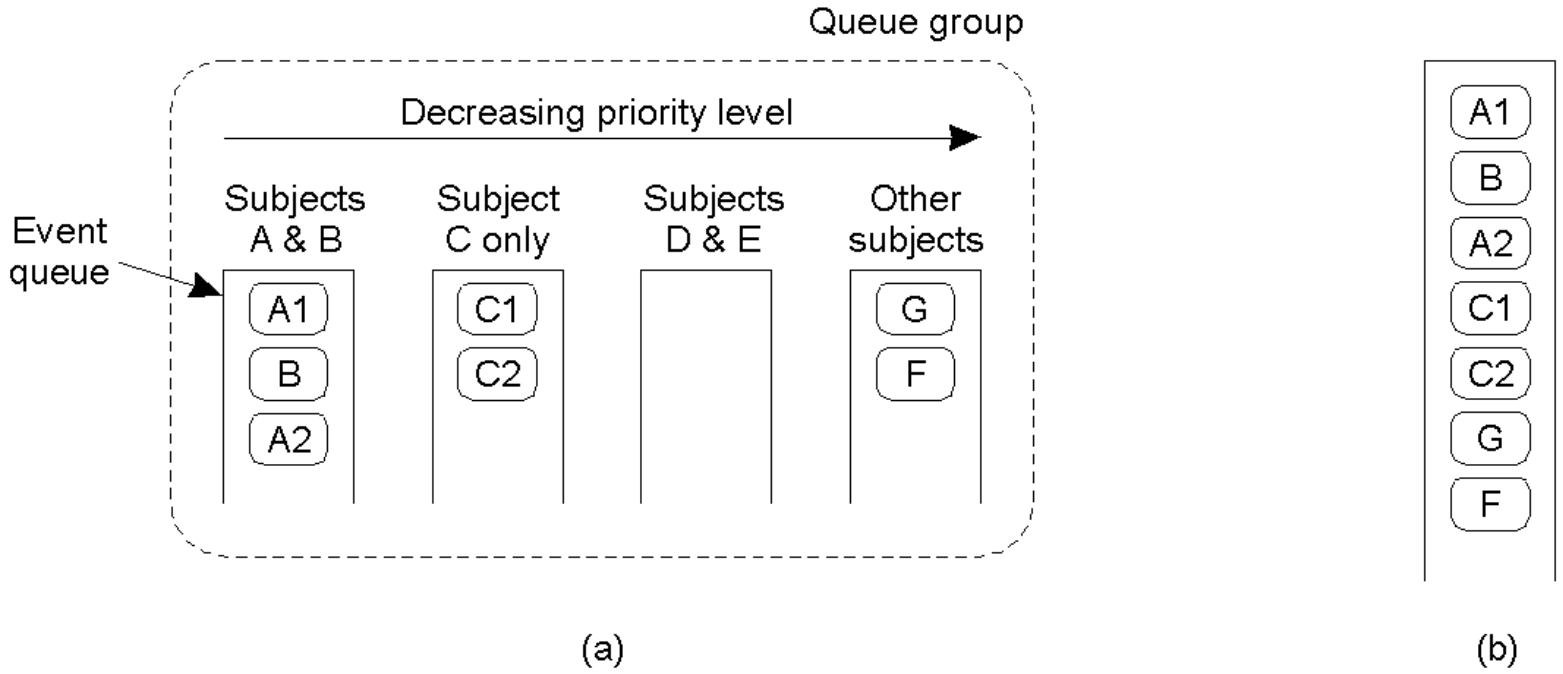
Processing listener events for subscriptions in TIB/Rendezvous.

Events (2)



Processing incoming messages in TIB/Rendezvous.

Processes



a) Priority scheduling of events through a queue group.

b) A semantically equivalent queue for the queue group with the specific event objects from (a).

Naming (1)

Example	Valid?
Books.Computer_systems.Distributed_Systems	Yes
.ftp.cuss.vu.nil	No (starts with a '.')
ftp.cuss.vu.nil	Yes
NEWS.res.com.so	Yes
Marten..van_Steen	No (empty label)
Marten.R.van_Steen	Yes

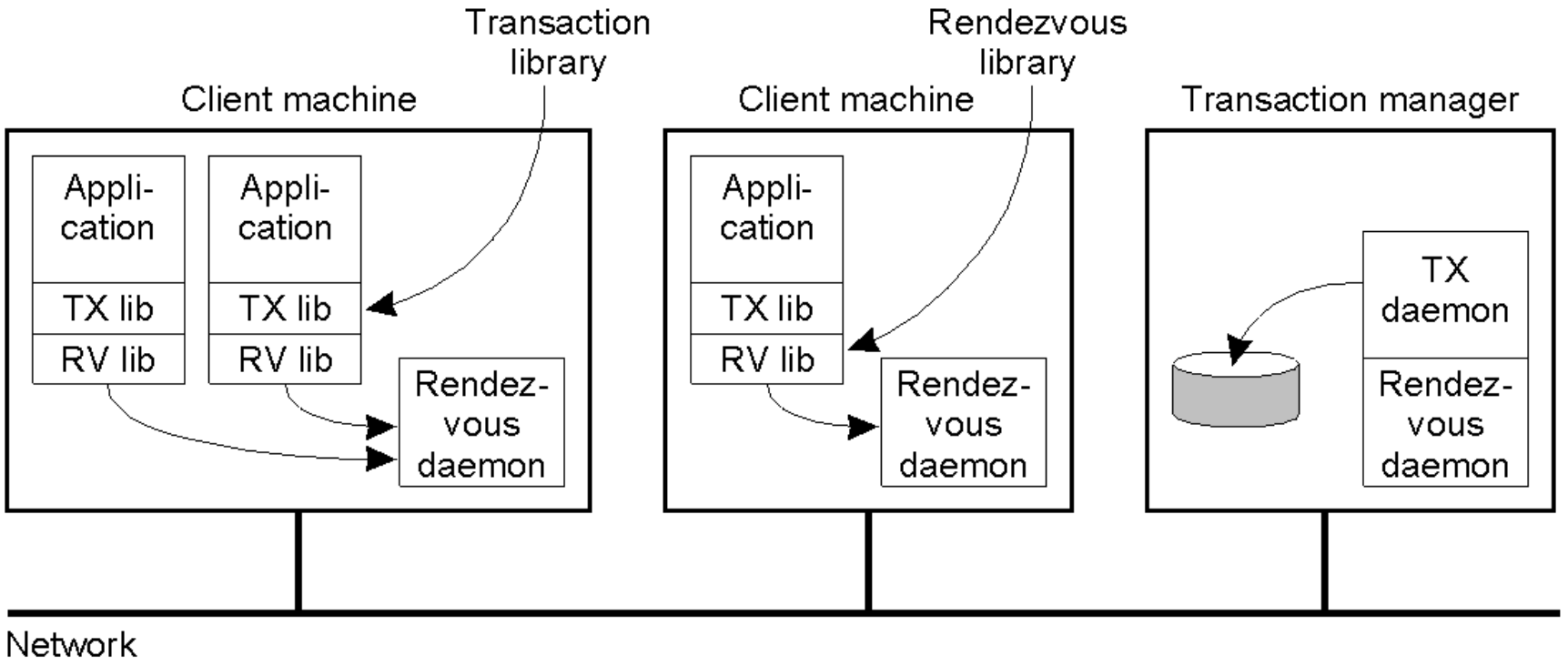
Examples of valid and invalid subject names.

Naming (2)

Subject Name	Matches
*.cuss.vu.nil	ftp.cuss.vu.nil www.cuss.vu.nil
ni.vu.>	nil.vu.cuss.ftp nil.vu.cuss.zephyr nil.vu.few.www
NEWS.comp.*.books	NEWS.comp.so.books NEWS.comp.ai.books NEWS.comp.se.books NEWS.comp.theory.books

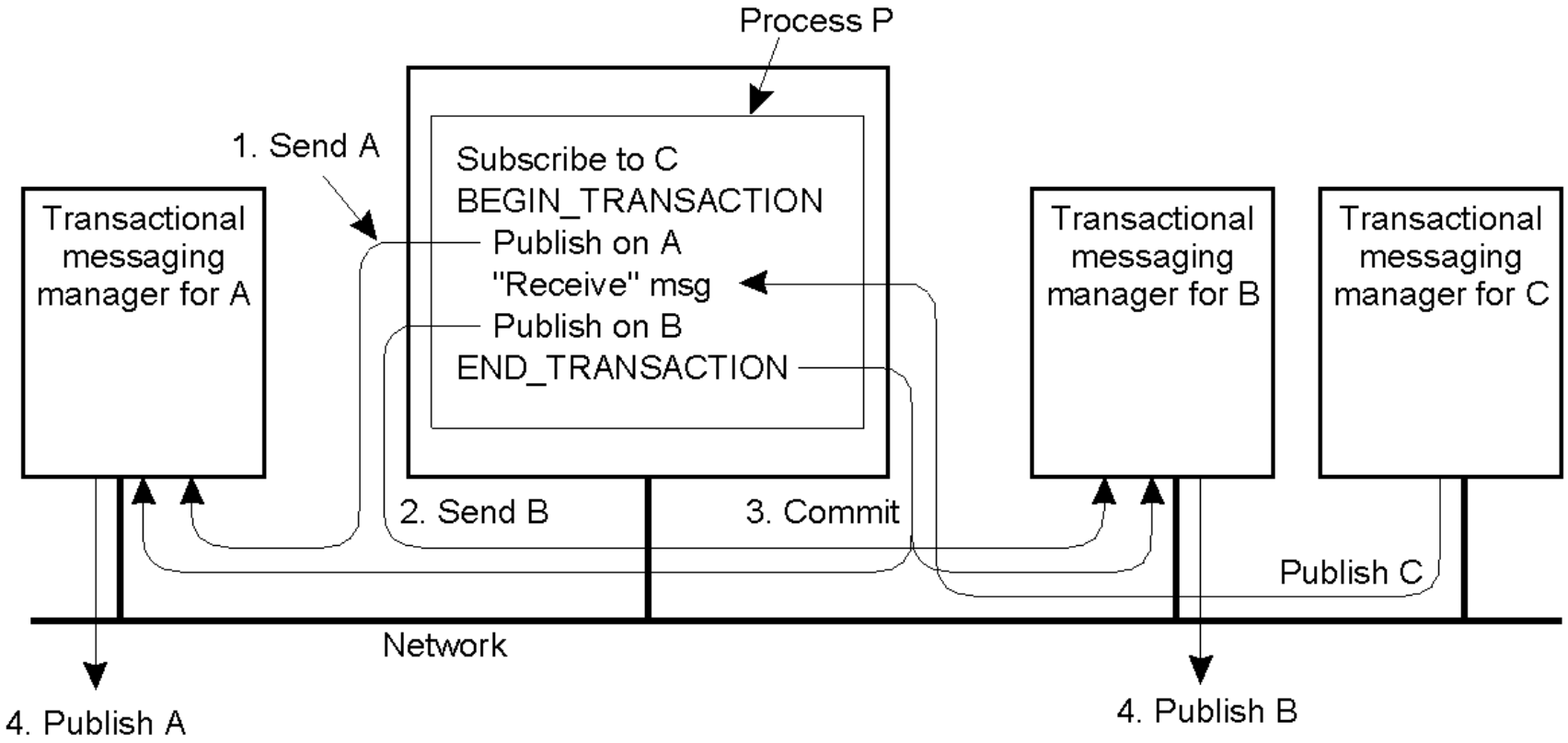
Examples of using wildcards in subject names.

Synchronization (1)



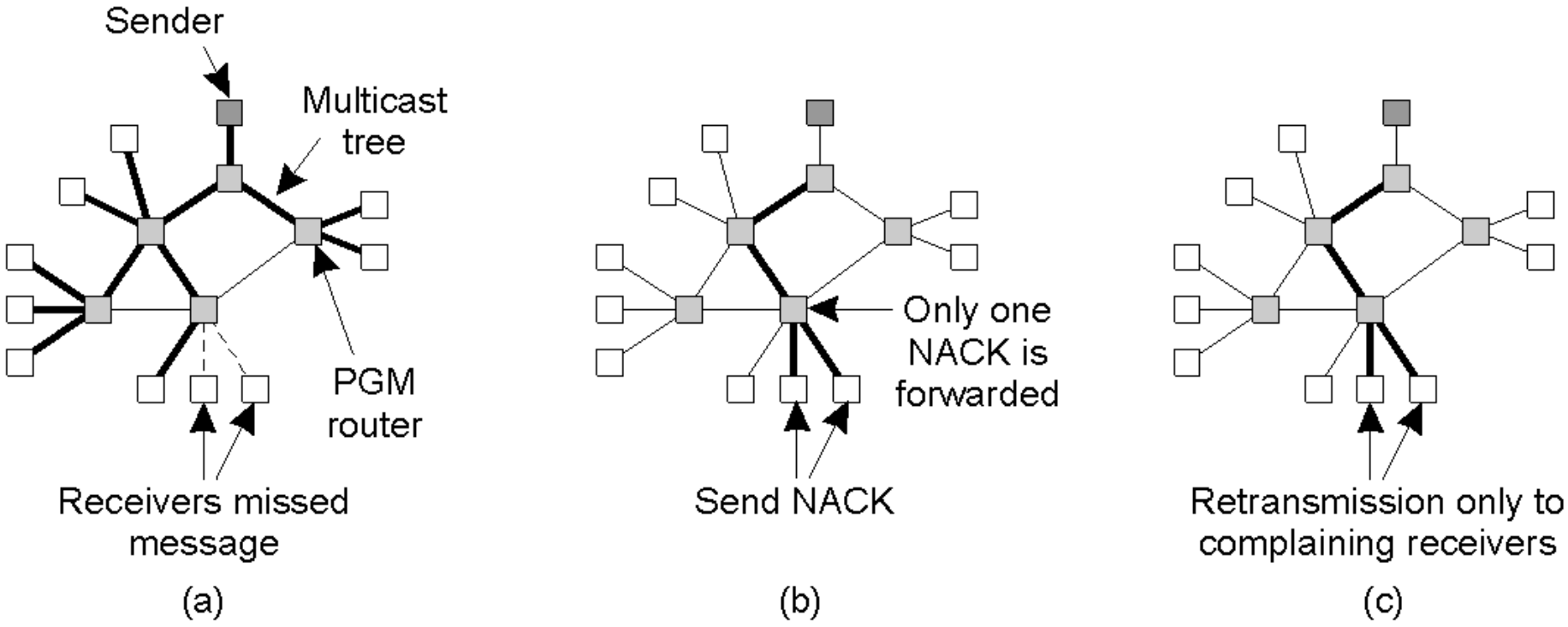
The organization of transactional messaging as a separate layer in TIB/Rendezvous.

Synchronization (2)



The organization of a transaction in TIB/Rendezvous.

Reliable Communication



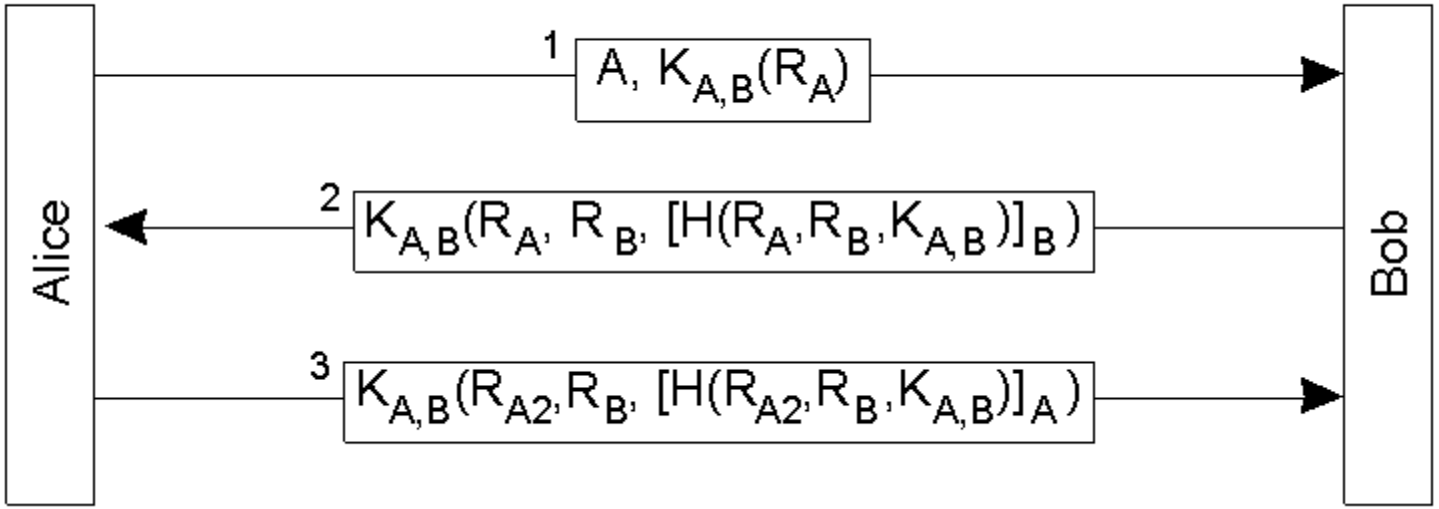
The principle of PGM.

a) A message is sent along a multicast tree

b) A router will pass only a single NACK for each message

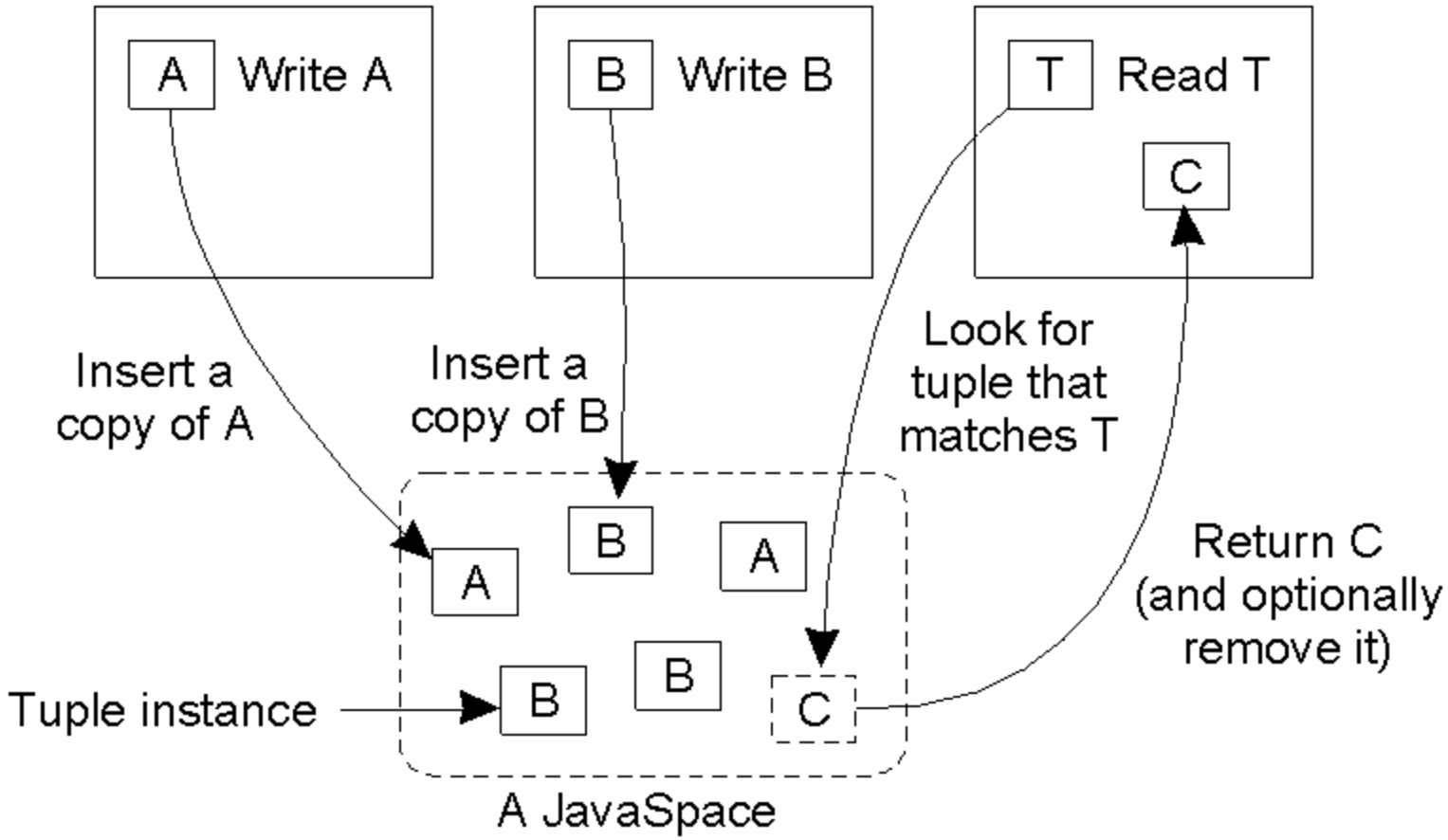
c) A message is retransmitted only to receivers that have asked for it.

Security



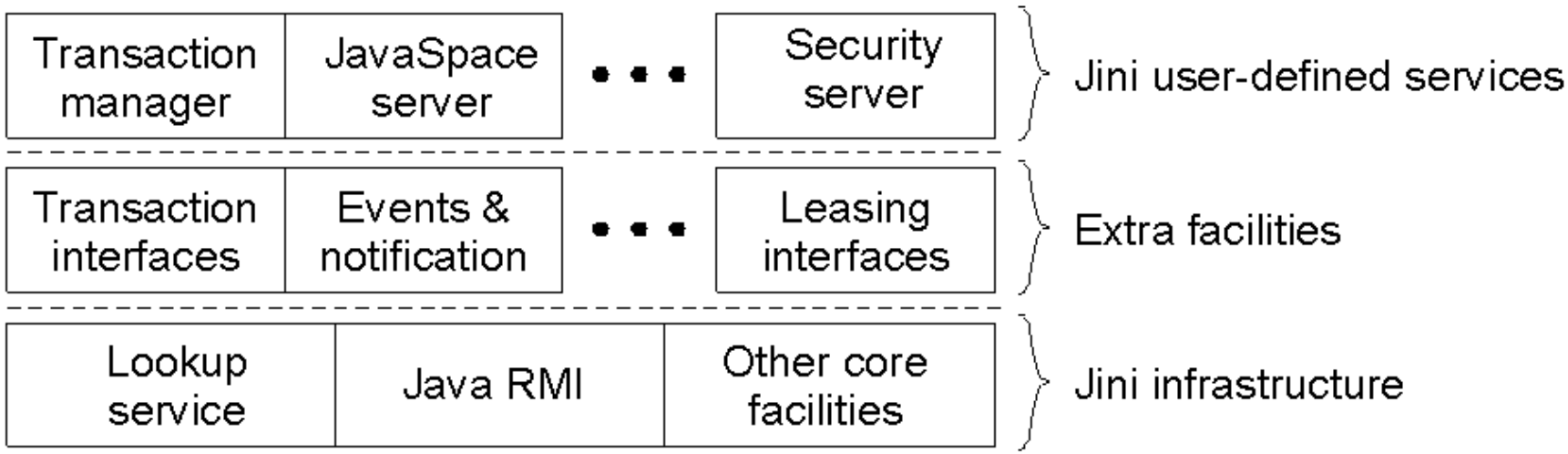
Establishing a secure channel
in TIB/Rendezvous.

Overview of Jini



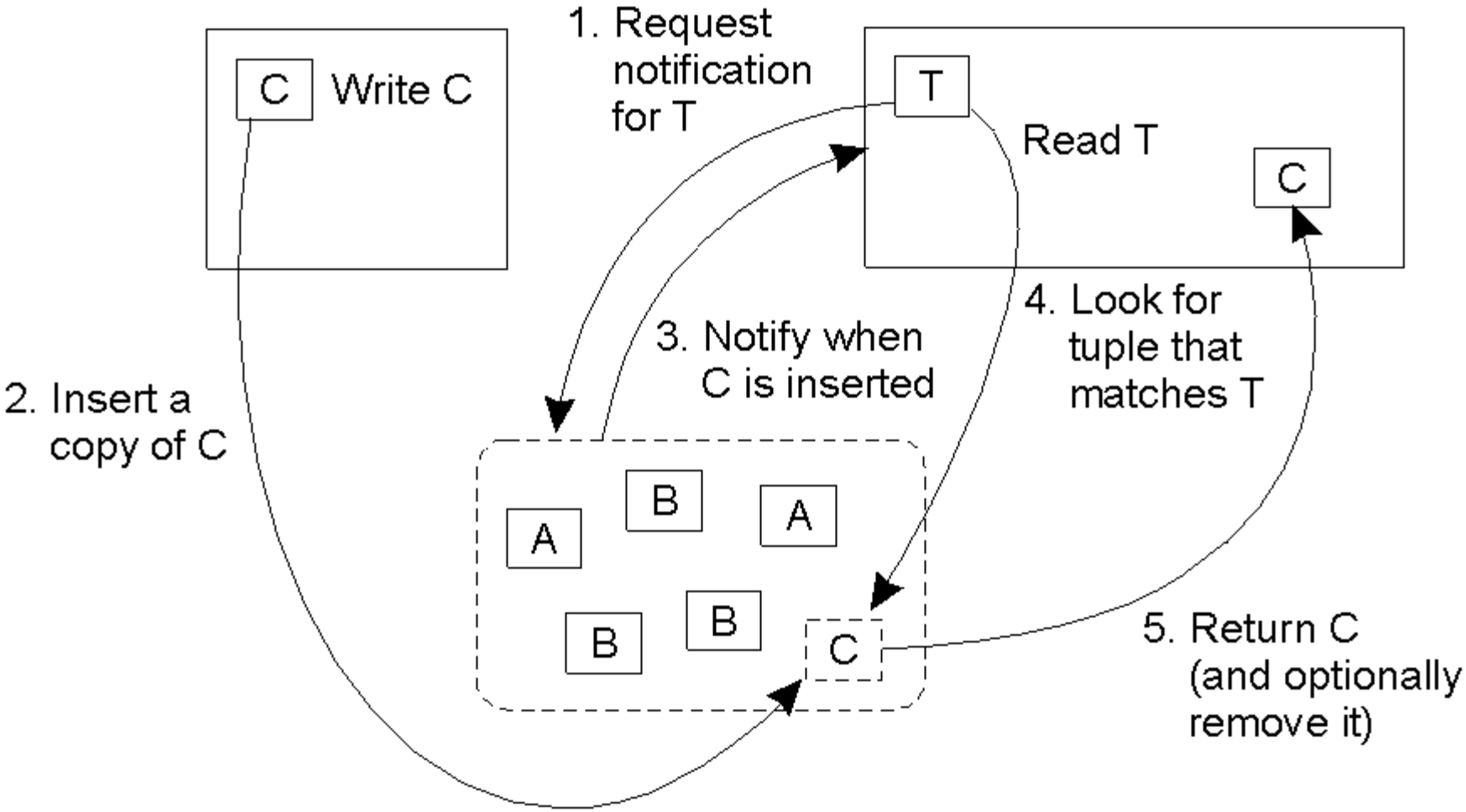
The general organization of a JavaSpace in Jini.

Architecture



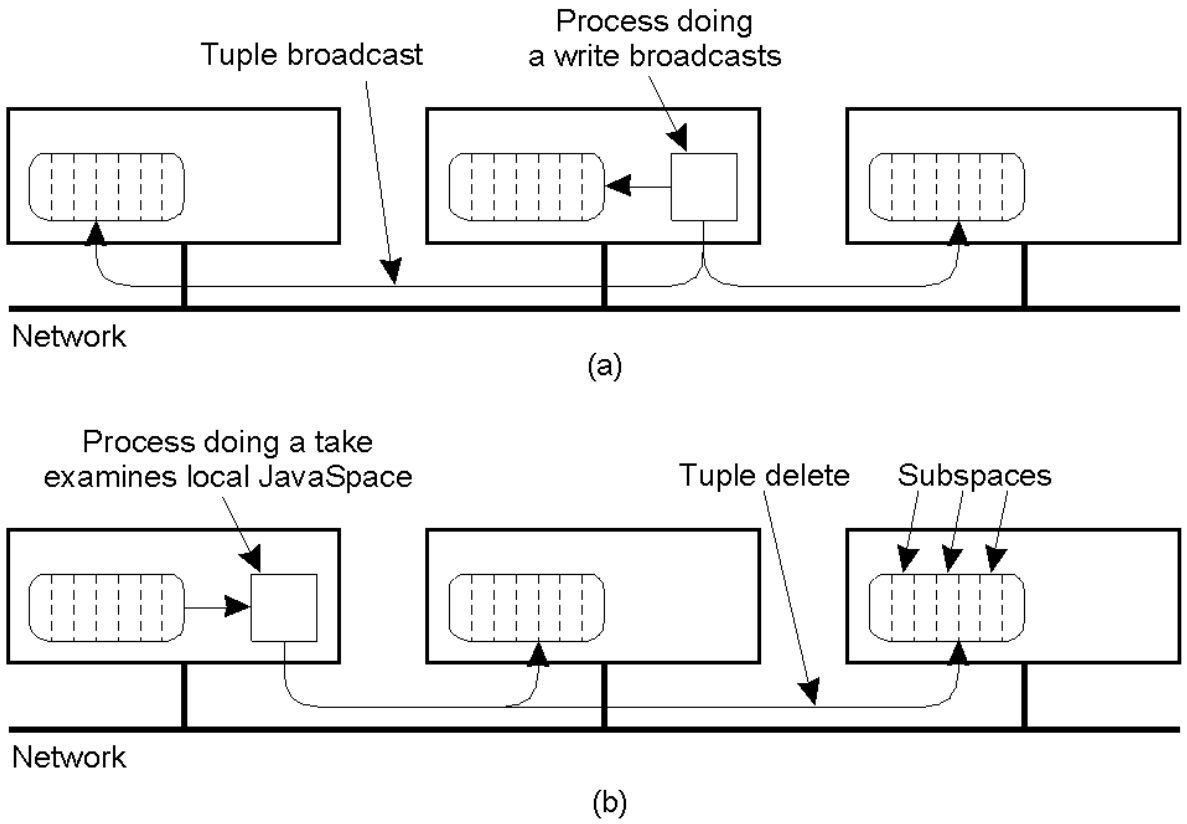
The layered architecture of a Jini System.

Communication Events



Using events in combination with a JavaSpace

Processes (1)

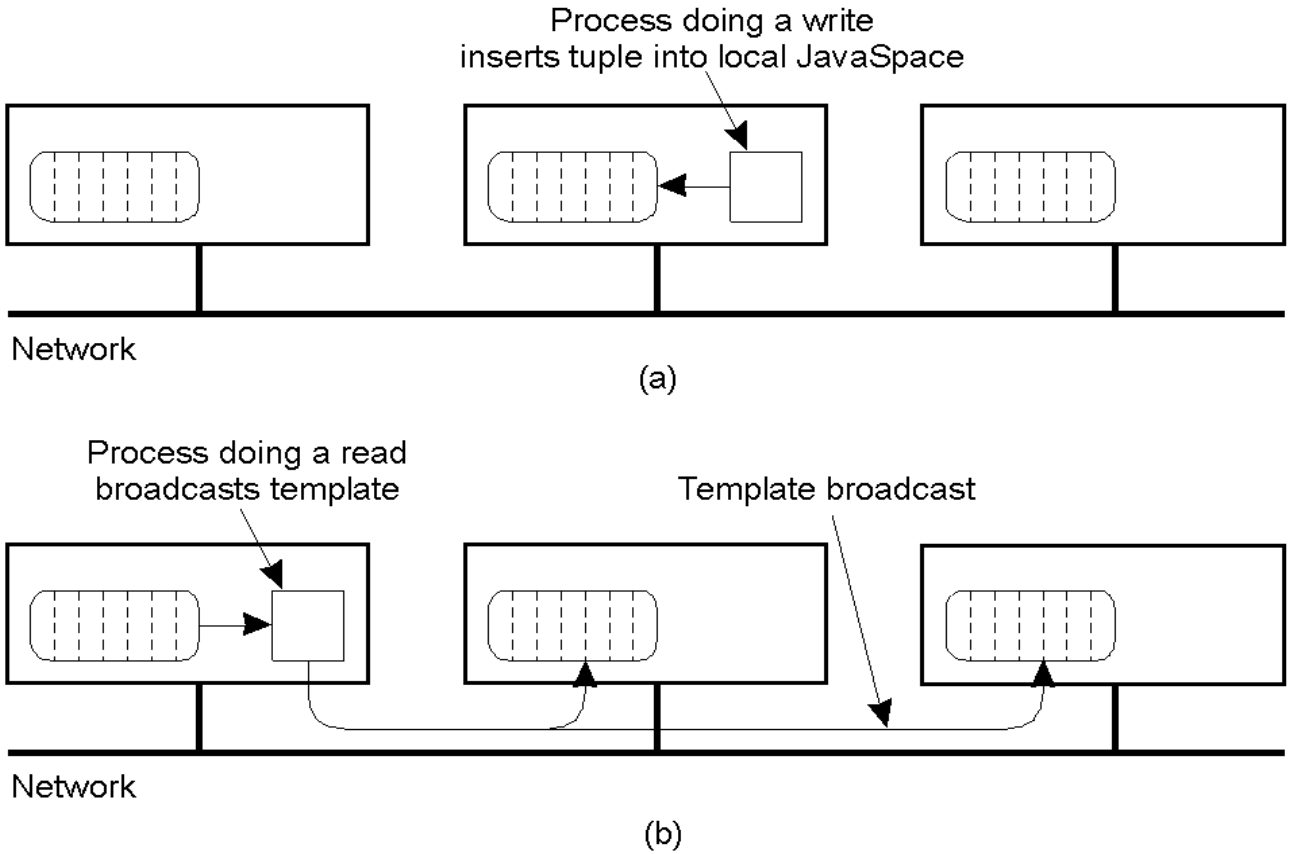


A JavaSpace can be replicated on all machines. The dotted lines show the partitioning of the JavaSpace into subspaces.

a) Tuples are broadcast on WRITE

b) READs are local, but the removing of an instance when calling TAKE must be broadcast

Processes (2)

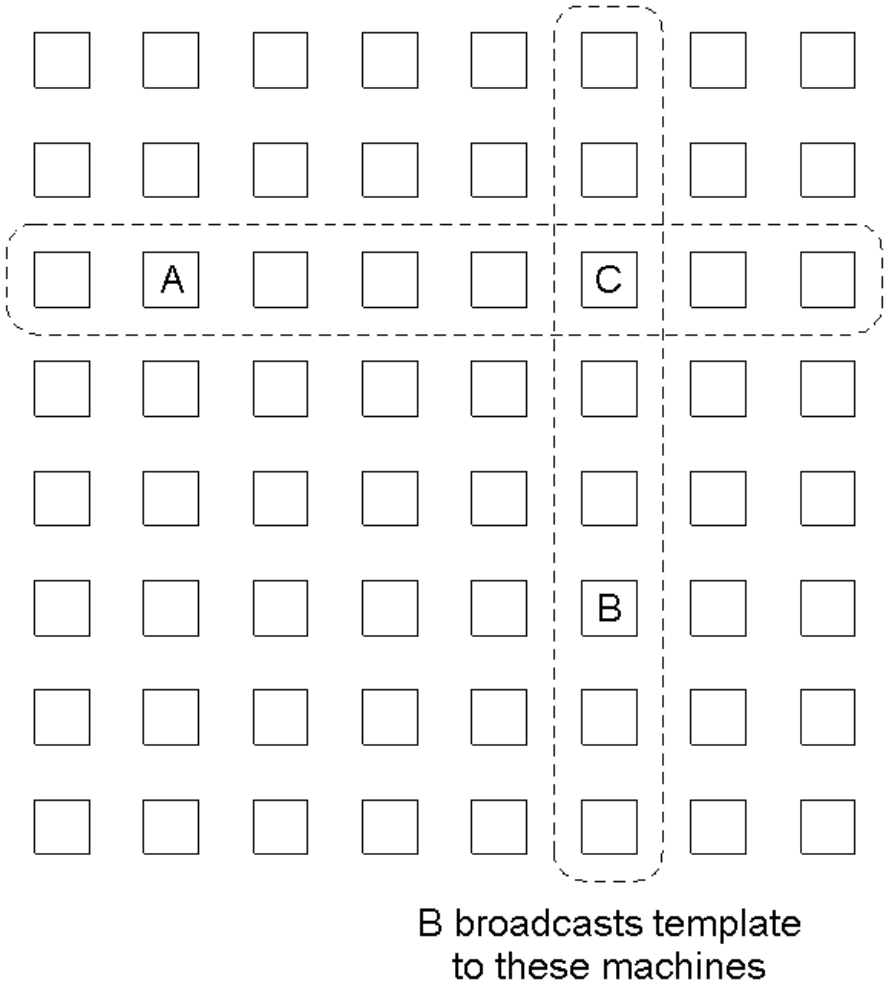


Unreplicated JavaSpace.

a) A WRITE is done locally.

b) A READ or TAKE requires the template tuple to be broadcast in order to find a tuple instance

Processes (3)



Partial broadcasting of tuples and template tuples.

The Jini Lookup Service (1)

Field	Description
ServiceID	The identifier of the service associated with this item.
Service	A (possibly remote) reference to the object implementing the service.
AttributeSets	A set of tuples describing the service.

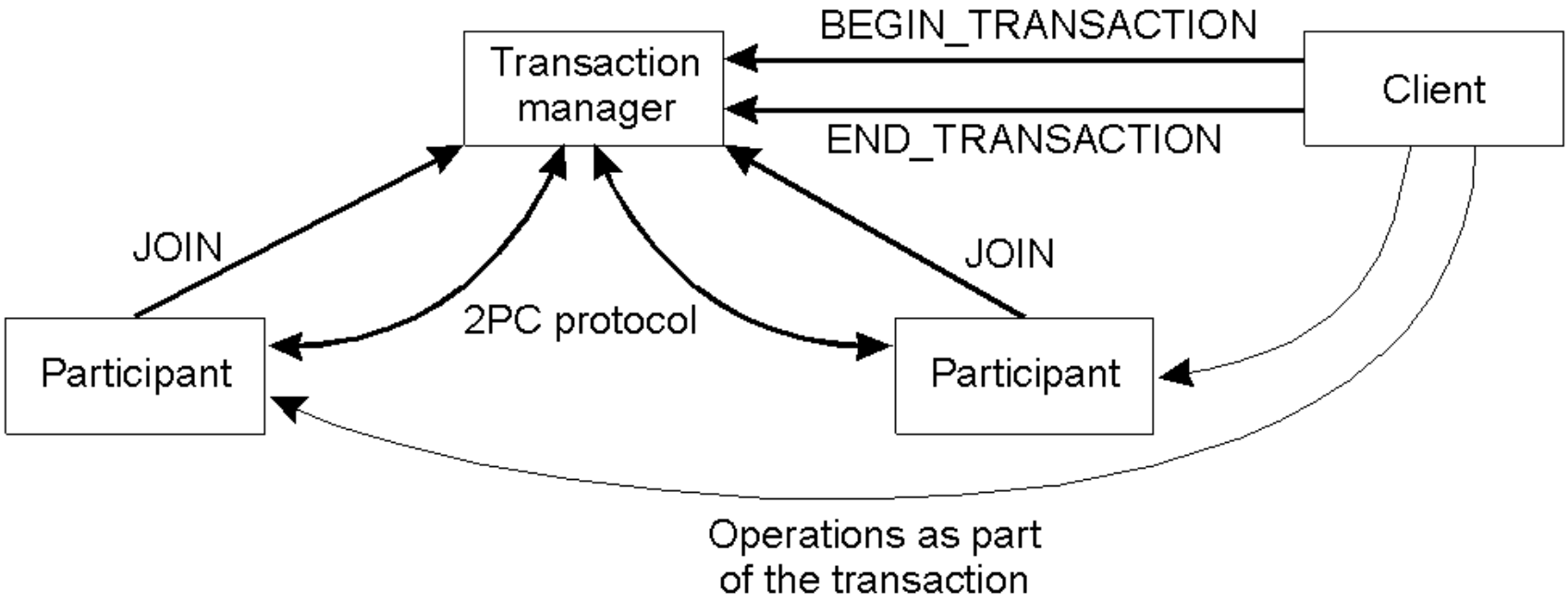
The organization of a service item.

The Jini Lookup Service (2)

Tuple Type	Attributes
ServiceInfo	Name, manufacturer, vendor, version, model, serial number
Location	Floor, room, building
Address	Street, organization, organizational unit, locality, state or province, postal code, country

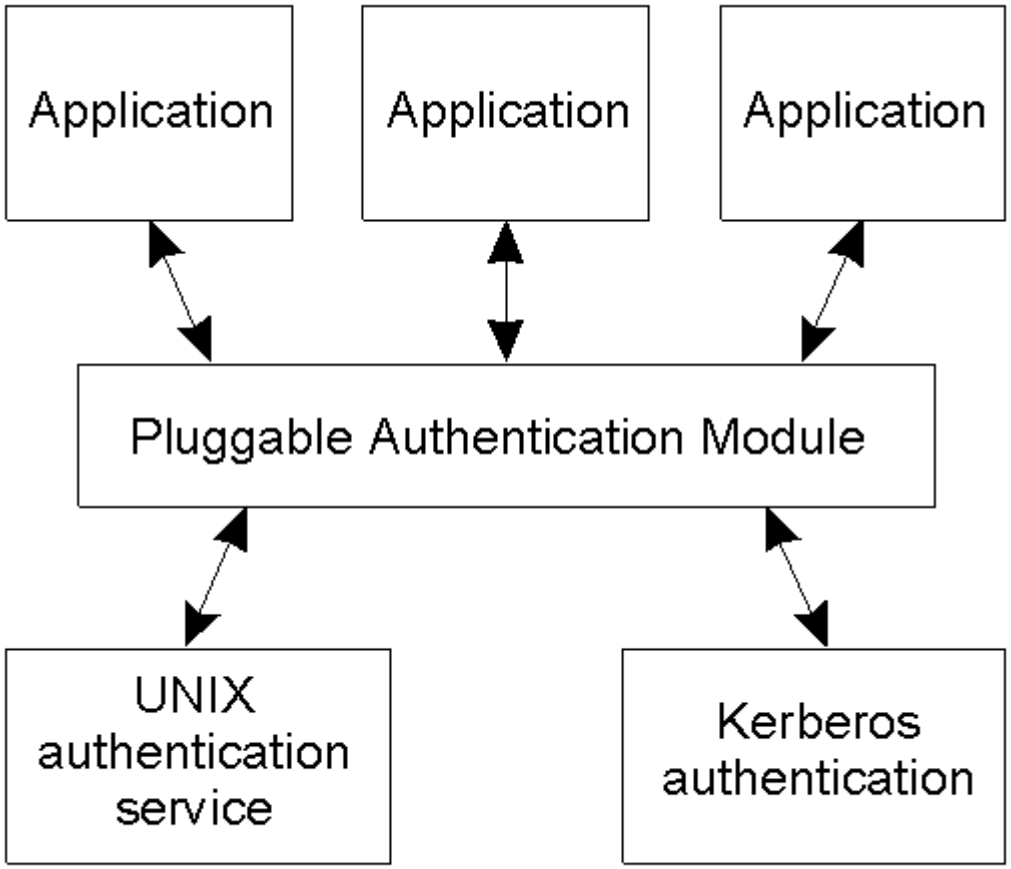
Examples of predefined tuples for service items.

Synchronization of Transactions



The general organization of a transaction in Jini. Thick lines show communication as required by Jini's transaction protocol

Caching and Replication



The position of PAM with respect to security services.

Comparison of TIB/Rendezvous and Jini

Issue	TIB/Rendezvous	Jini
Major design goal	Uncoupling of processes	Flexible integration
Coordination model	Publish/subscribe	Generative communication
Network communication	Multicasting	Java RMI
Messages	Self-describing	Process specific
Event mechanism	For incoming messages	As a callback service
Processes	General purpose	General purpose
Names	Character strings	Byte strings
Naming services	None	Lookup service
Transactions (operations)	Messages	Method invocations
Transactions (scope)	Single process (see text)	Multiple processes
Locking	No	As JavaSpace operations
Caching and replication	No	No
Reliable communication	Yes	Yes
Process groups	Yes	No
Recovery mechanisms	No explicit support	No explicit support
Security	Secure channels	Based entirely on Java