Systèmes Distribués

Master MIAGE 1

Andrea G. B. Tettamanzi
Université de Nice Sophia Antipolis
Département Informatique
andrea.tettamanzi@unice.fr

CM - Séance 6 – Partie II

Tolérance aux pannes et sécurité

(chapitres 8 et 9)

Basic Concepts

Dependability (= *sûreté de fonctionnement*) Includes

- Availability (= disponibilité)
- Reliability (= fiabilité)
- Safety (= sécurité)
- Maintainability (= maintenabilité)

Types of Failures

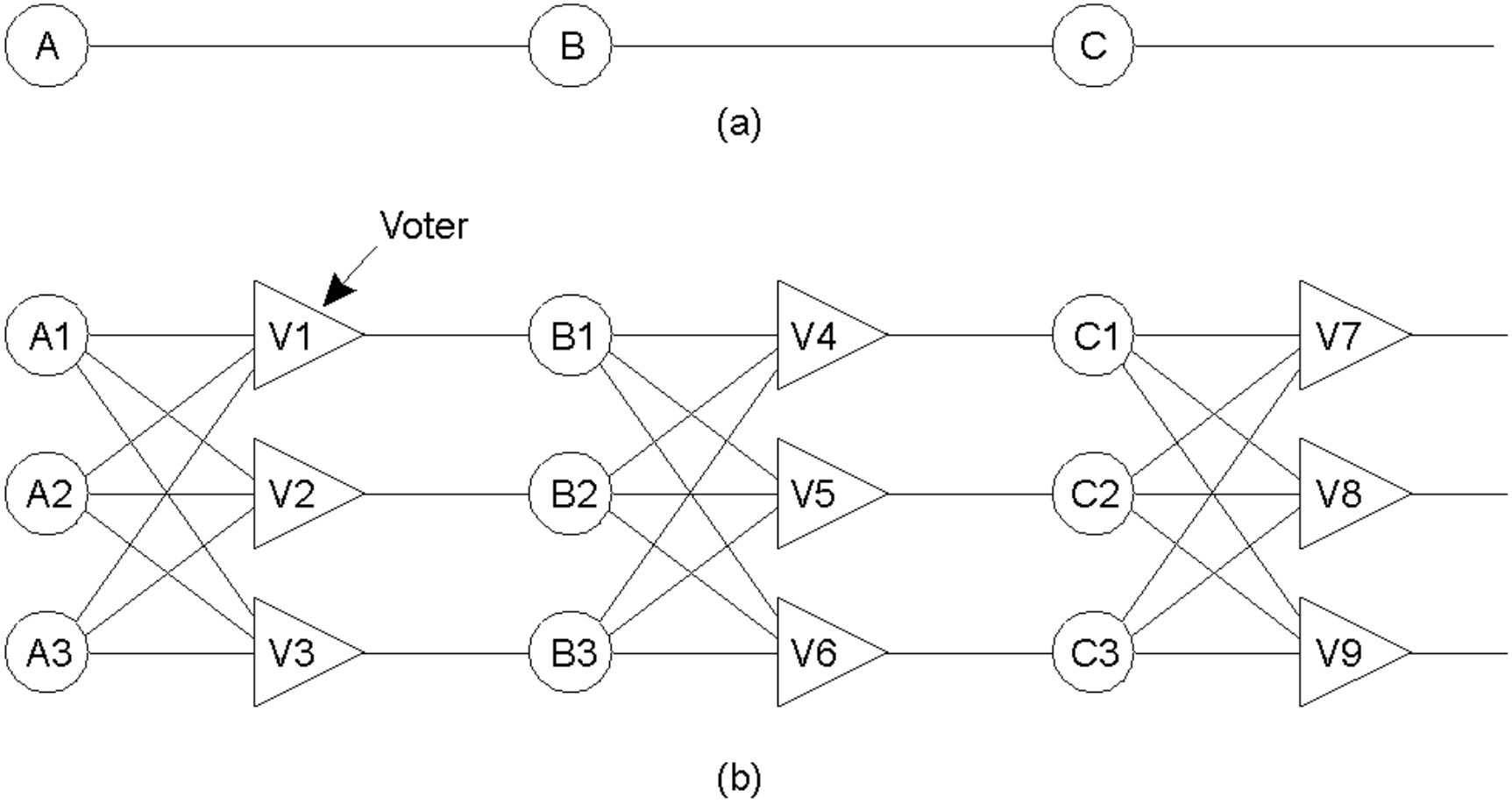
- Transient Failures
- Intermittent Failures
- Permanent Failures

Failure Models

Type of failure	Description
Crash failure	A server halts, but is working correctly until it halts
Omission failure *Receive omission* *Send omission*	A server fails to respond to incoming requests A server fails to receive incoming messages A server fails to send messages
Timing failure	A server's response lies outside the specified time interval
Response failure *Value failure* *State transition failure*	The server's response is incorrect The value of the response is wrong The server deviates from the correct flow of control
Arbitrary failure	A server may produce arbitrary responses at arbitrary times

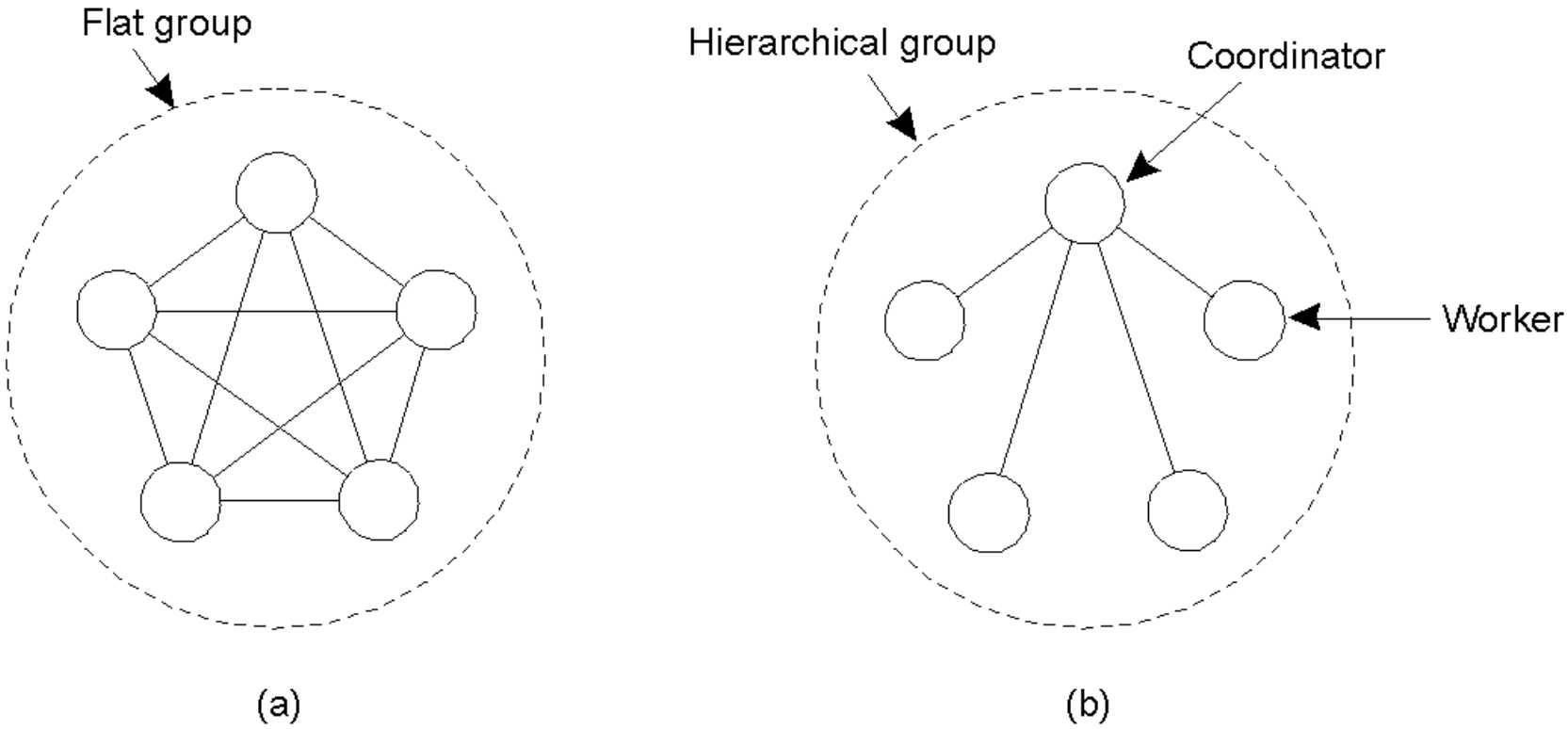
Different types of failures.

Failure Masking by Redundancy



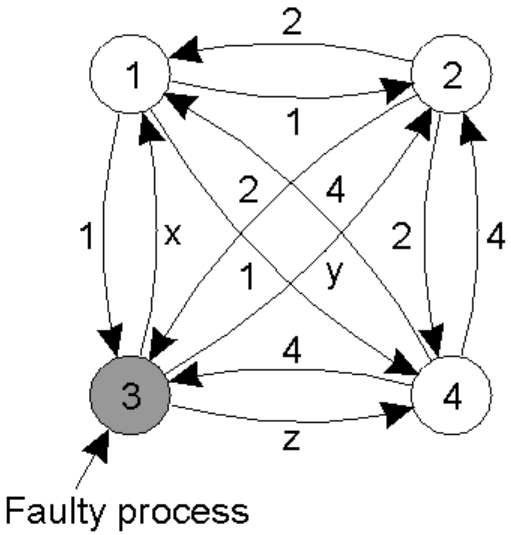
Triple modular redundancy.

Flat Groups versus Hierarchical Groups



a) Communication in a flat group.
b) Communication in a simple hierarchical group

Agreement in Faulty Systems (1)



(a)

1 Got(1, 2, x, 4)
2 Got(1, 2, y, 4)
3 Got(1, 2, 3, 4)
4 Got(1, 2, z, 4)

(b)

1 Got	2 Got	4 Got
(1, 2, y, 4)	(1, 2, x, 4)	(1, 2, x, 4)
(a, b, c, d)	(e, f, g, h)	(1, 2, y, 4)
(1, 2, z, 4)	(1, 2, z, 4)	(i, j, k, l)

(c)

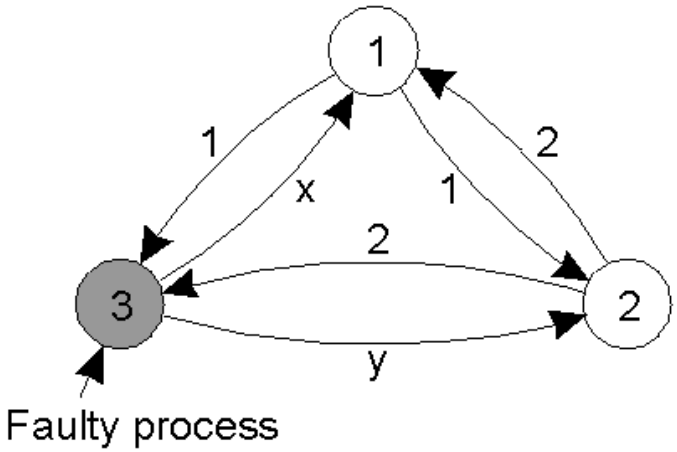
The Byzantine generals problem for 3 loyal generals and1 traitor.

a) The generals announce their troop strengths (in units of 1000 soldiers).

b) The vectors that each general assembles based on (a)

c) The vectors that each general receives in step 3.

Agreement in Faulty Systems (2)



(a)

1 Got(1, 2, x)
2 Got(1, 2, y)
3 Got(1, 2, 3)

(b)

1 Got	2 Got
(1, 2, y)	(1, 2, x)
(a, b, c)	(d, e, f)

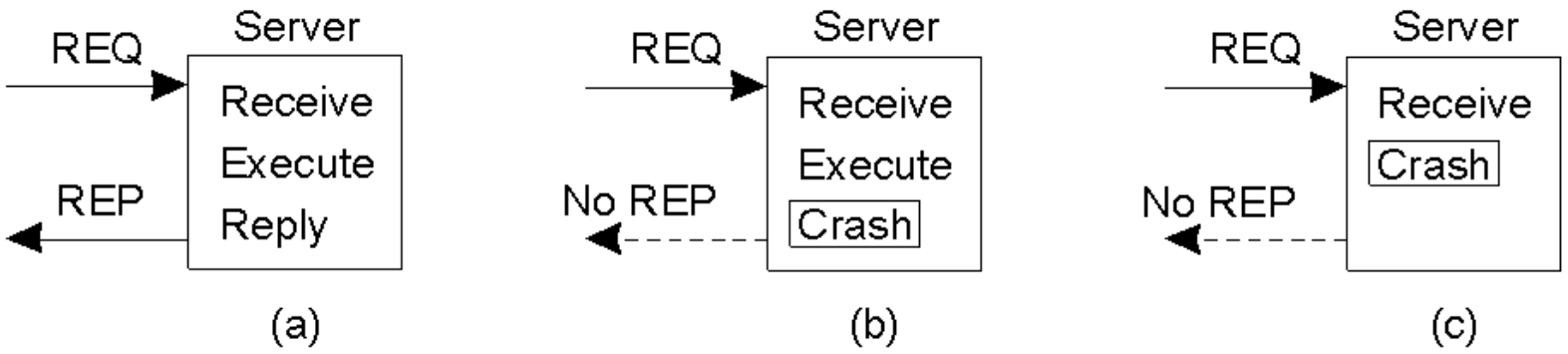
(c)

The same as in previous slide, except now with 2 loyal generals and one traitor.

Byzantine Fault Tolerance

- Feldman and Micali (1997) have shown that there are solutions to the problem only when #generals ≥ 3 · #traitors + 1.
- An alternative solution requires public-key cryptography to ensure autenticity of messages, and maintains Byzantine fault tolerance in presence of an arbitrary number of traitors.
- Castro and Liskov introduced the Practical Byzantine Fault Tolerance (PBFT) algorithm in 1999.
- Many BFT protocols have been implemented since.
- One example of BFT in use is **Bitcoin**, a peer-to-peer digital currency system, where a proof-of-work chain is the solution to the problem.

Lost Request Messages
Server Crashes (1)



A server in client-server communication

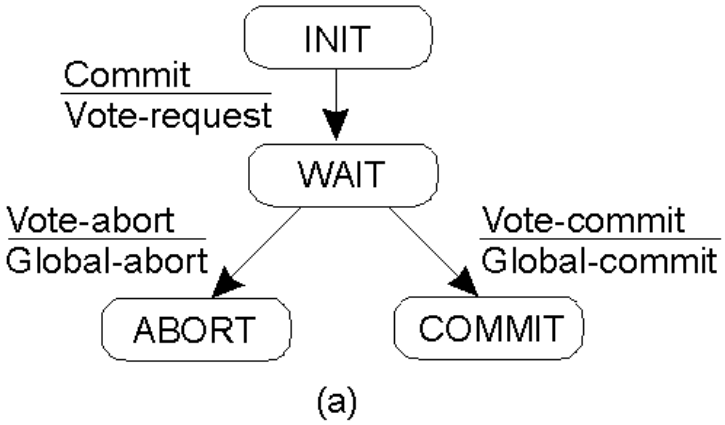
a) Normal case
b) Crash after execution
c) Crash before execution

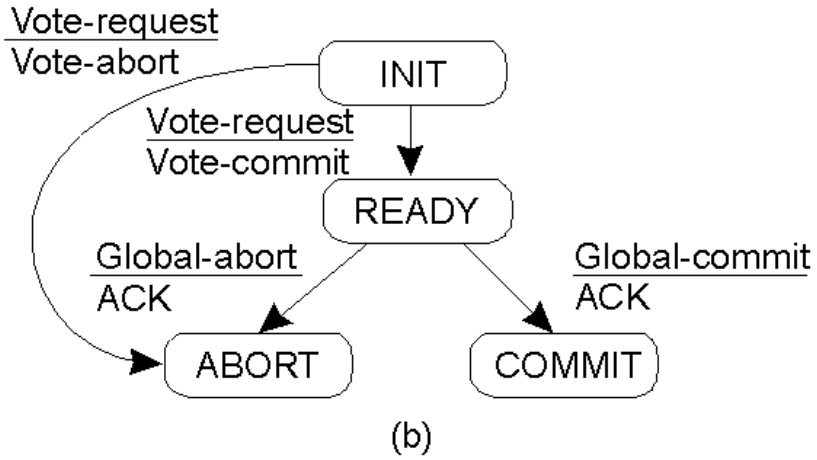
Server Crashes (2)

Client	Server					
	Strategy M -> P			Strategy P -> M		
Reissue strategy	**MPC**	**MC(P)**	**C(MP)**	**PMC**	**PC(M)**	**C(PM)**
Always	DUP	OK	OK	DUP	DUP	OK
Never	OK	ZERO	ZERO	OK	OK	ZERO
Only when ACKed	DUP	OK	ZERO	DUP	OK	ZERO
Only when not ACKed	OK	ZERO	OK	OK	DUP	OK

Different combinations of client and server strategies in the presence of server crashes.

Two-Phase Commit (1)





a) The finite state machine for the coordinator in 2PC.

b) The finite state machine for a participant.

Two-Phase Commit: Assumptions

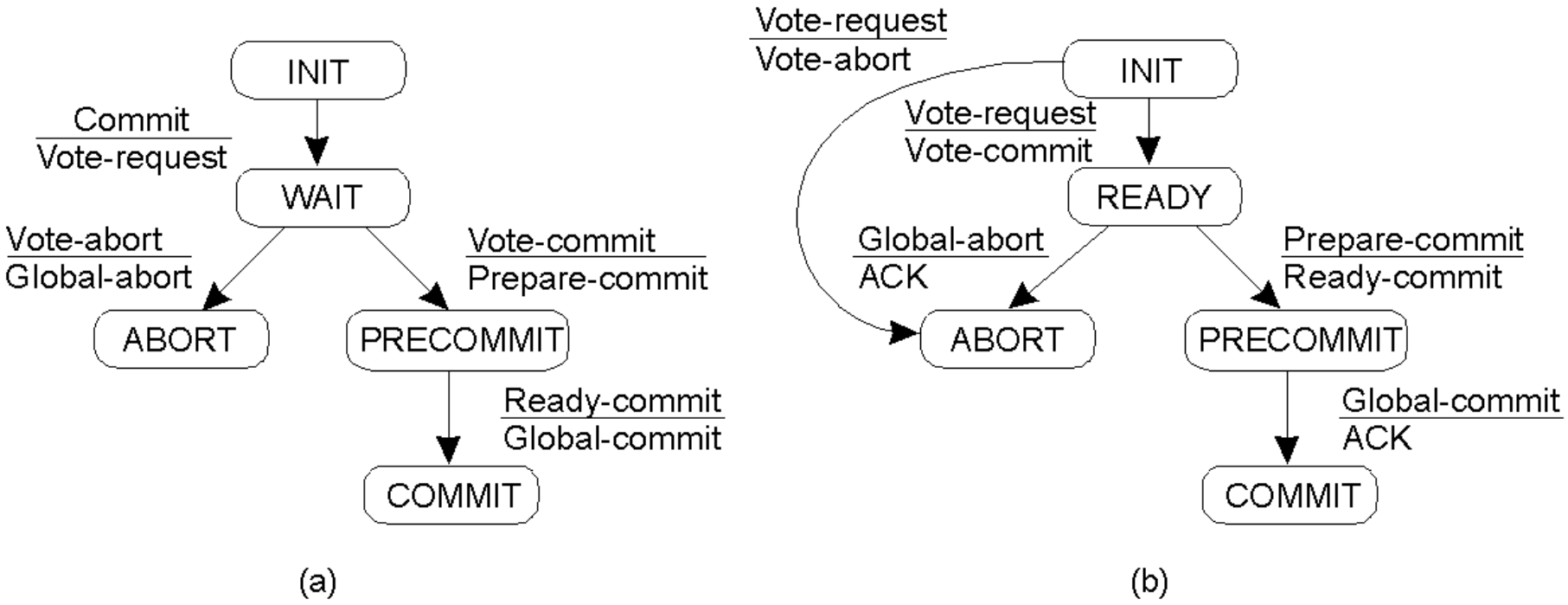
- Assumptions made by the protocol :
  - There is stable storage at each node with a write-ahead log
  - No node crashes forever
  - The data in the write-ahead log is never lost or corrupted in a crash
  - Any two nodes can communicate with each other.
- Comments :
  - The last assumption is not too restrictive, as network communication can typically be rerouted.
  - The first three assumptions are much stronger: if a node is totally destroyed, then data *may* be lost.

Two-Phase Commit (2)

State of Q	Action by P
COMMIT	Make transition to COMMIT
ABORT	Make transition to ABORT
INIT	Make transition to ABORT
READY	Contact another participant

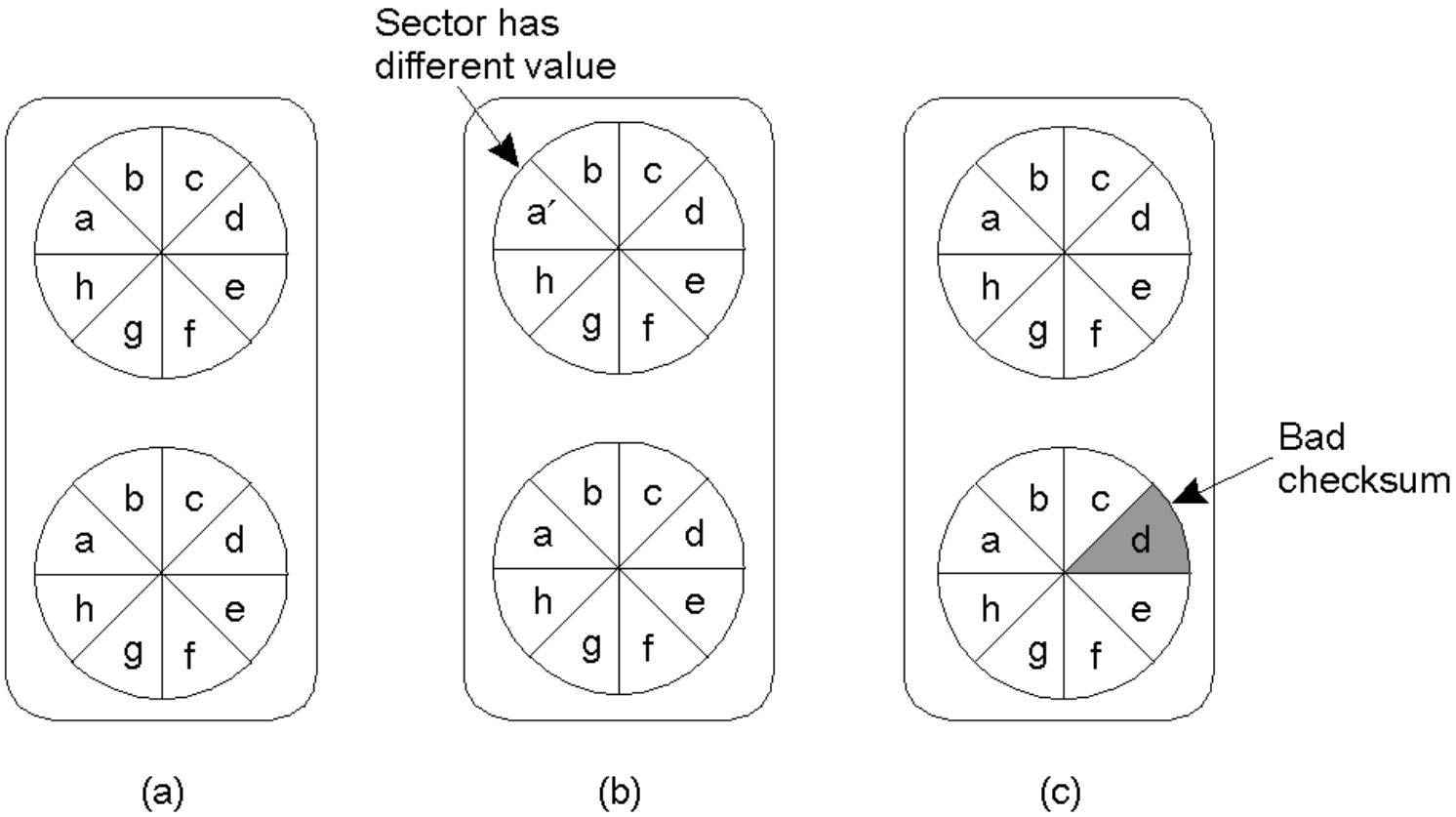
Actions taken by a participant *P* when residing in state *READY* and having contacted another participant *Q*.

Three-Phase Commit



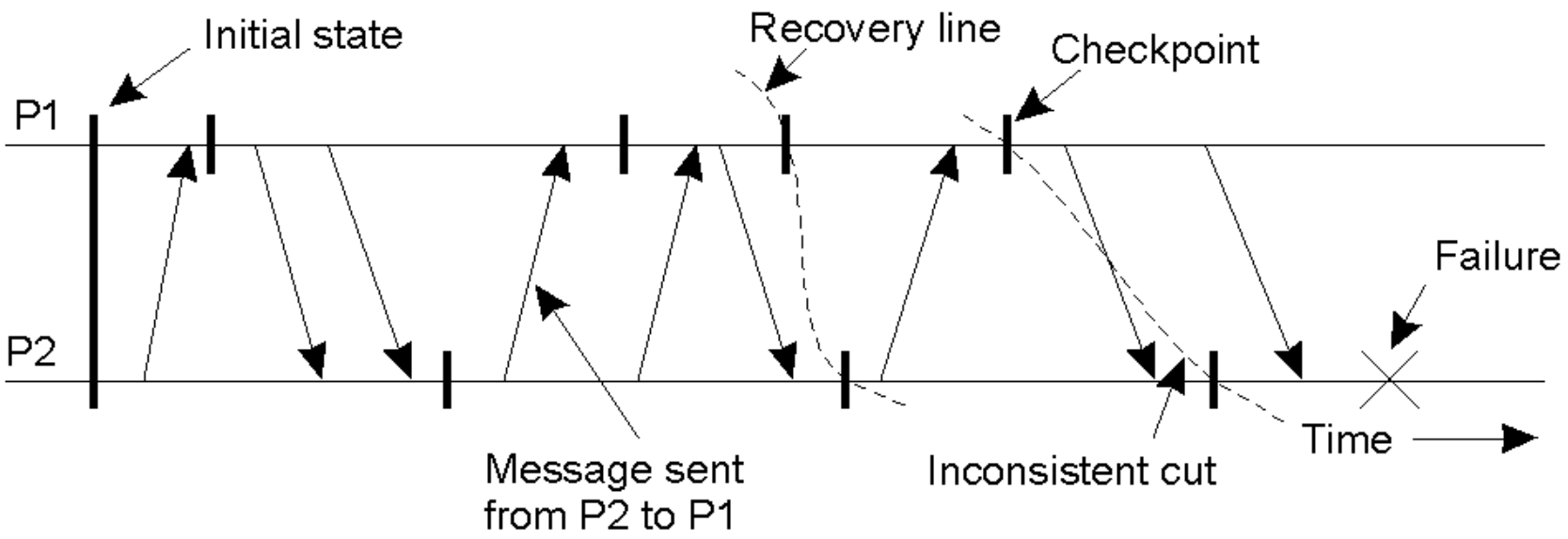
a) Finite state machine for the coordinator in 3PC
b) Finite state machine for a participant

Recovery Stable Storage



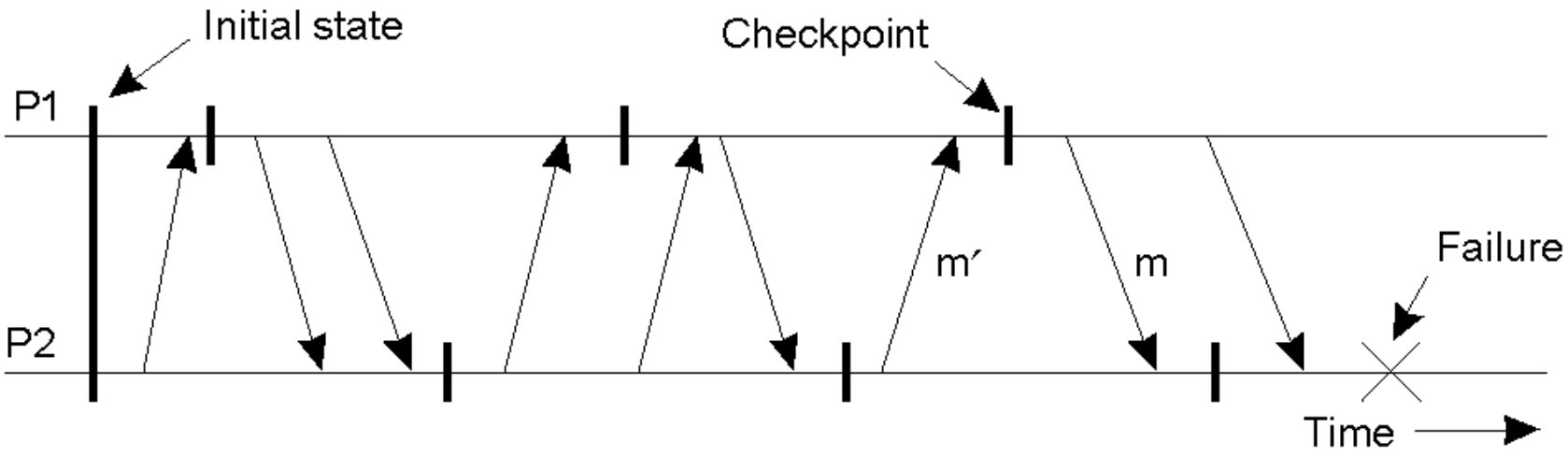
a) Stable Storage
b) Crash after drive 1 is updated
c) Bad spot

Checkpointing



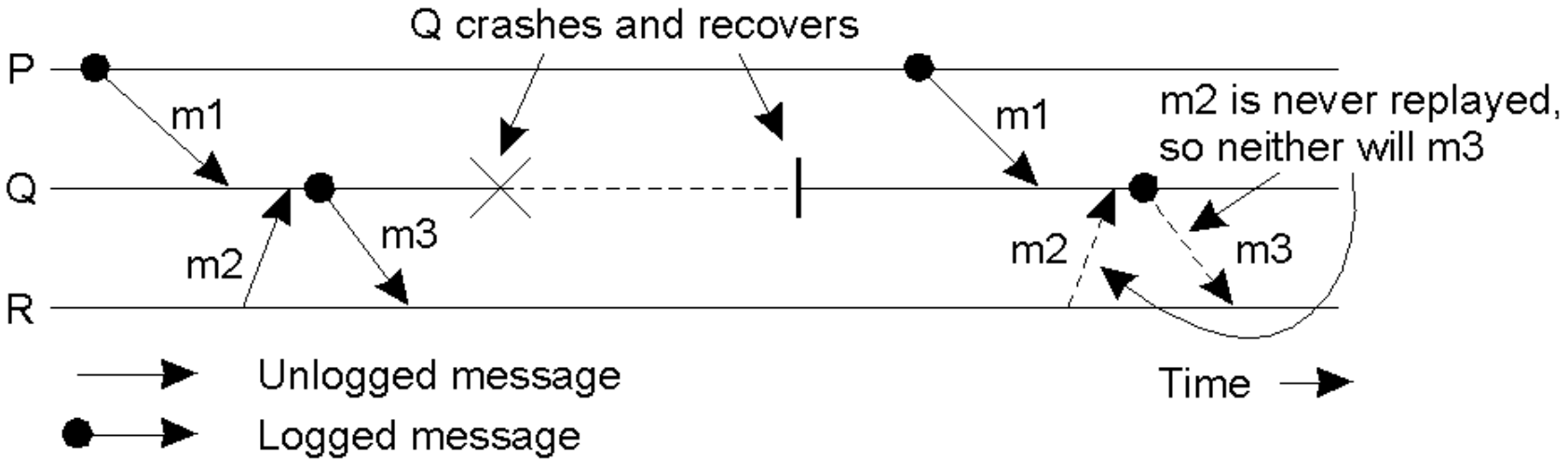
A recovery line.

Independent Checkpointing



The domino effect.

Message Logging



Incorrect replay of messages after recovery, leading to an orphan process.

Security

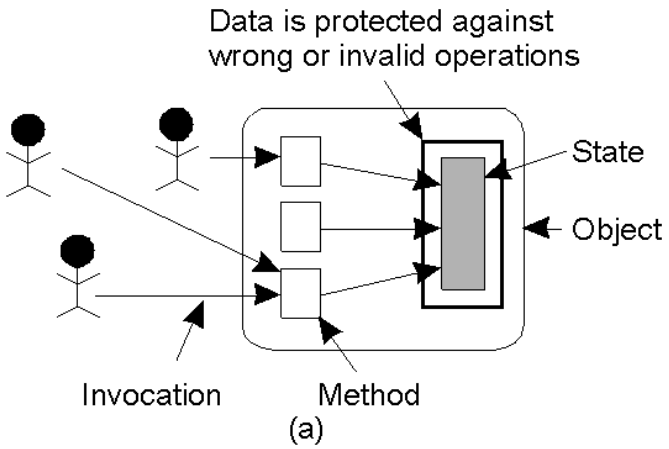
Security : Types of Threats

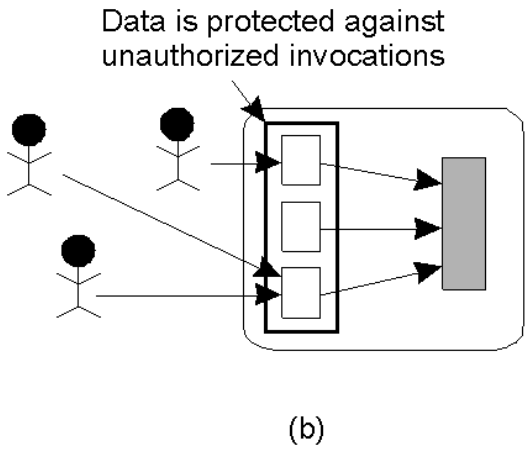
- Interception
- Interruption
- Modification
- Fabrication

Security Mechanisms

- Encryption
- Authentication
- Authorization
- Auditing

Focus of Control



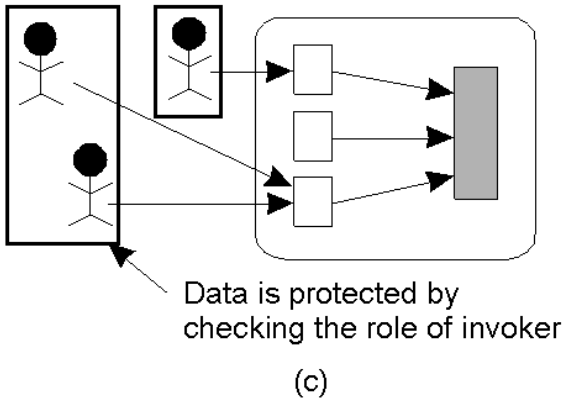


Three approaches for protection against security threats

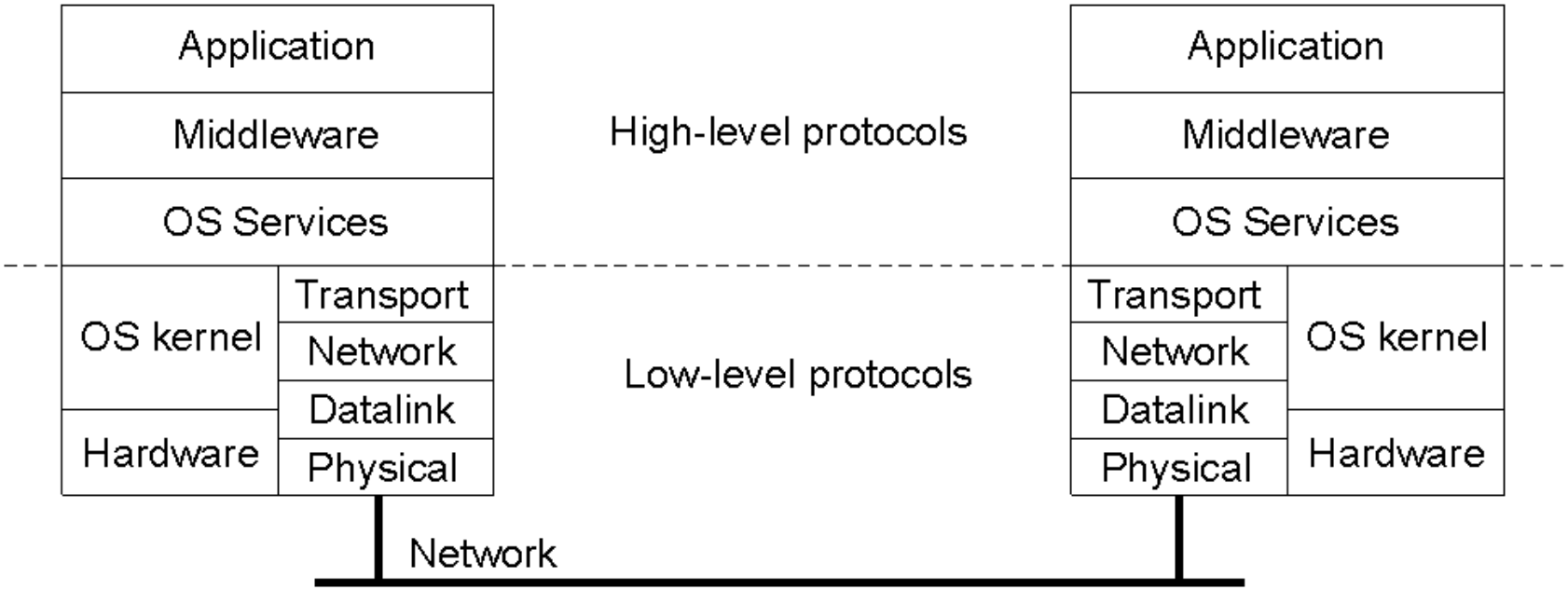
a) Protection against invalid operations

b) Protection against unauthorized invocations

c) Protection against unauthorized users

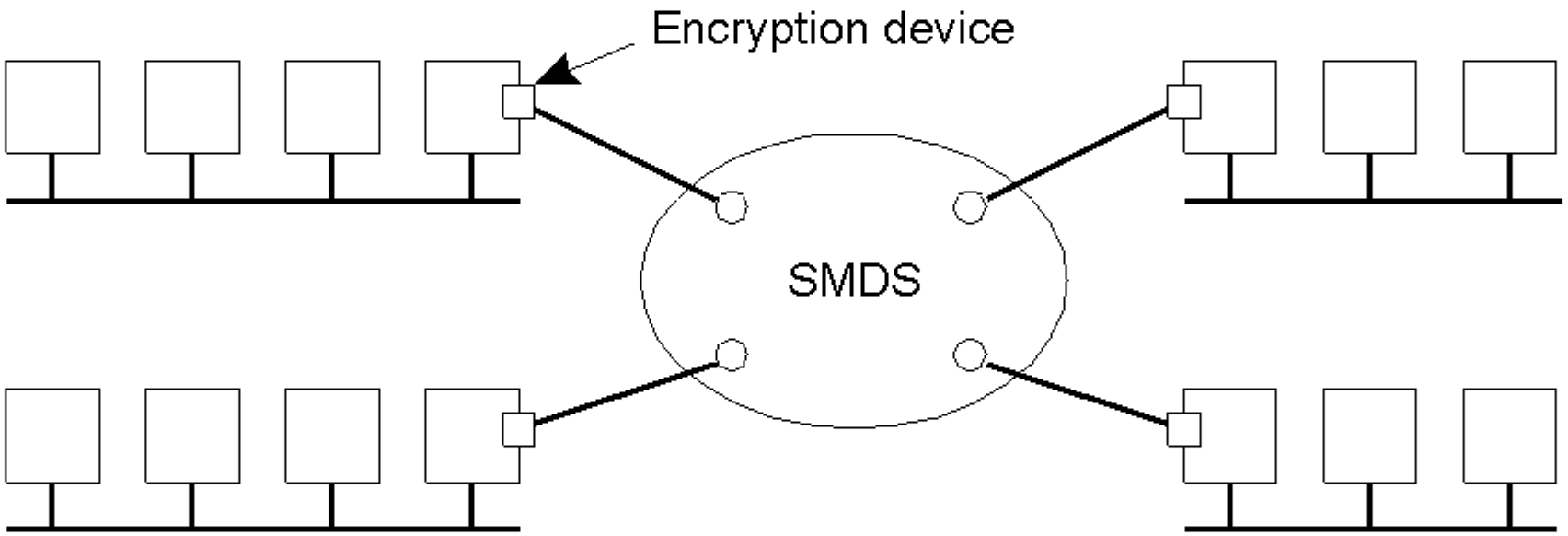


Layering of Security Mechanisms (1)



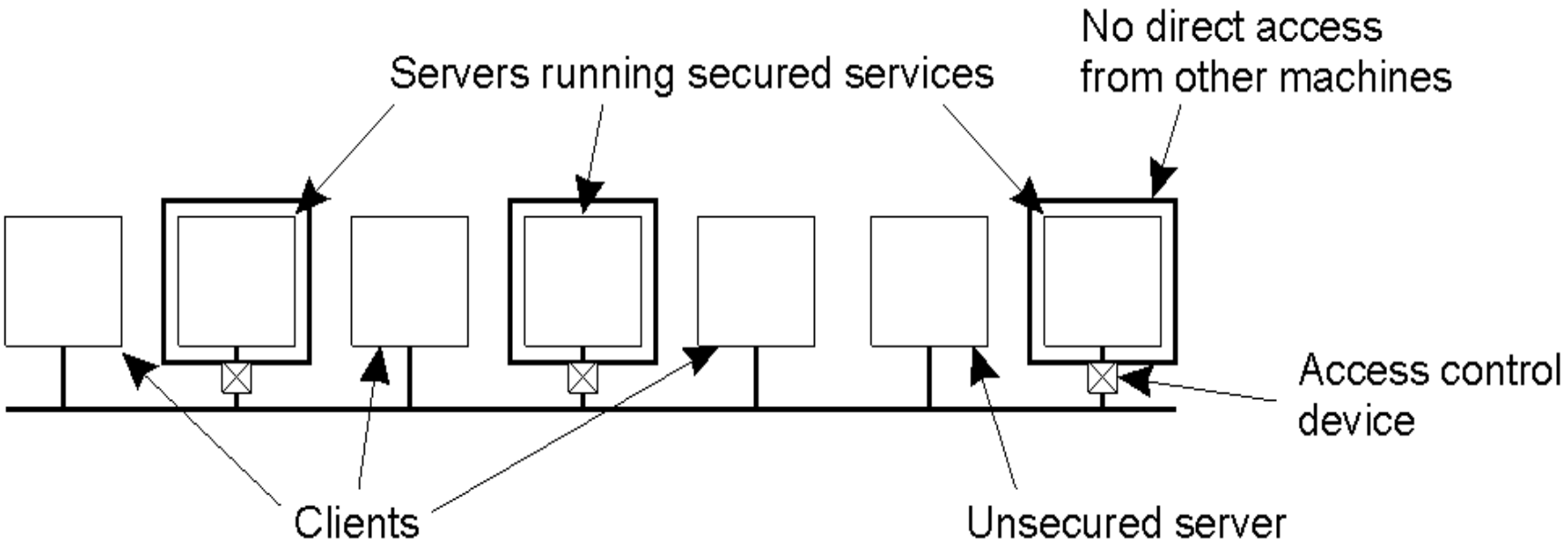
The logical organization of a distributed system into several layers.

Layering of Security Mechanisms (2)



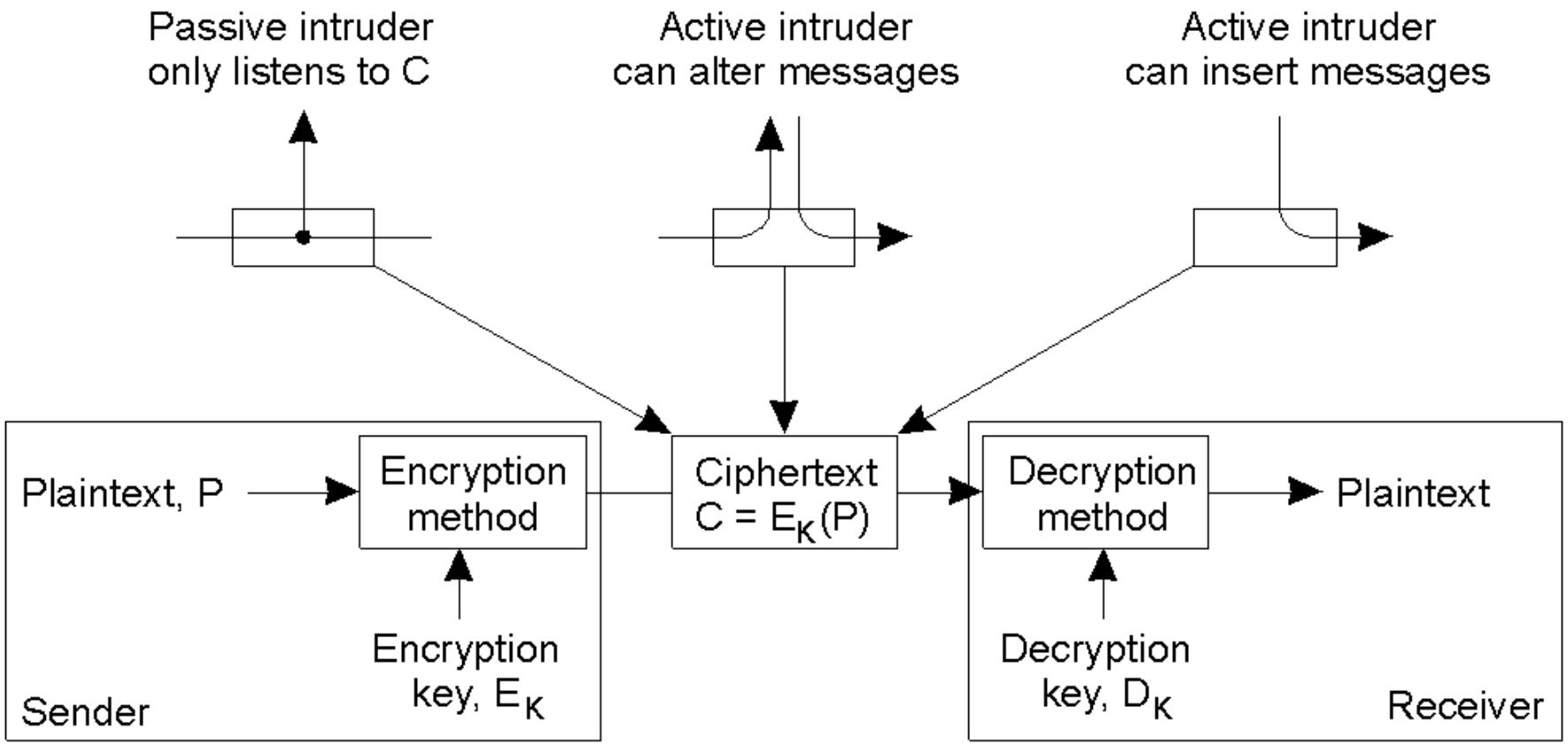
Several sites connected through a wide-area backbone service.

Distribution of Security Mechanisms



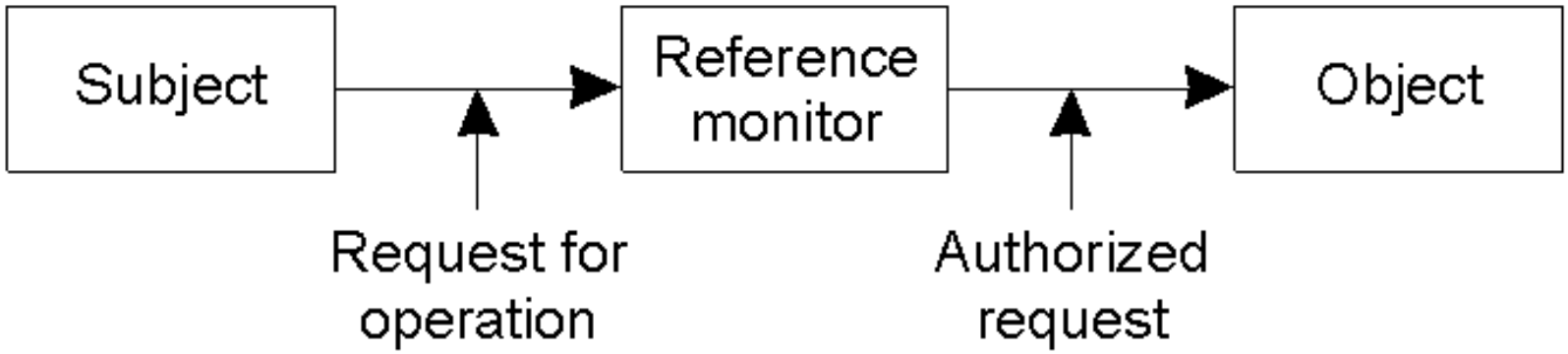
The principle of RISSC as applied to secure distributed systems.

Cryptography



Intruders and eavesdroppers in communication.

General Issues in Access Control



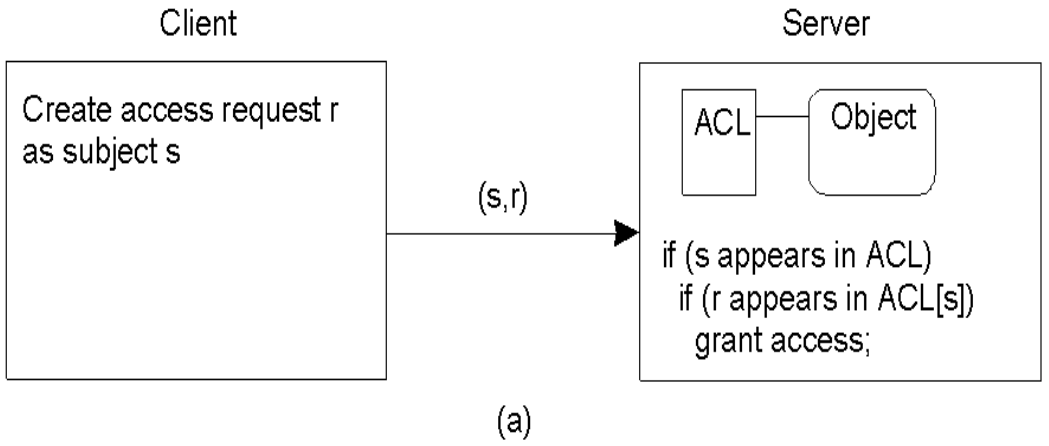
General model of controlling access to objects.

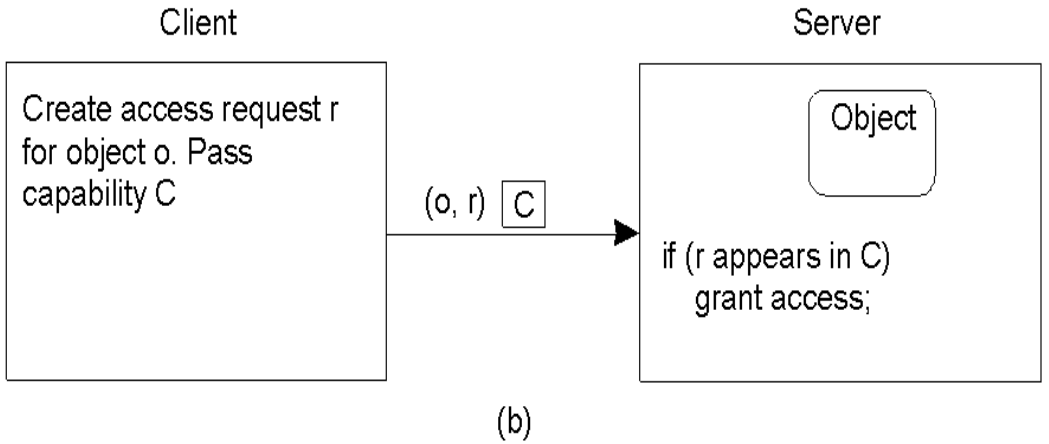
Access Control Matrix

Comparison between ACLs and capabilities for protecting objects.

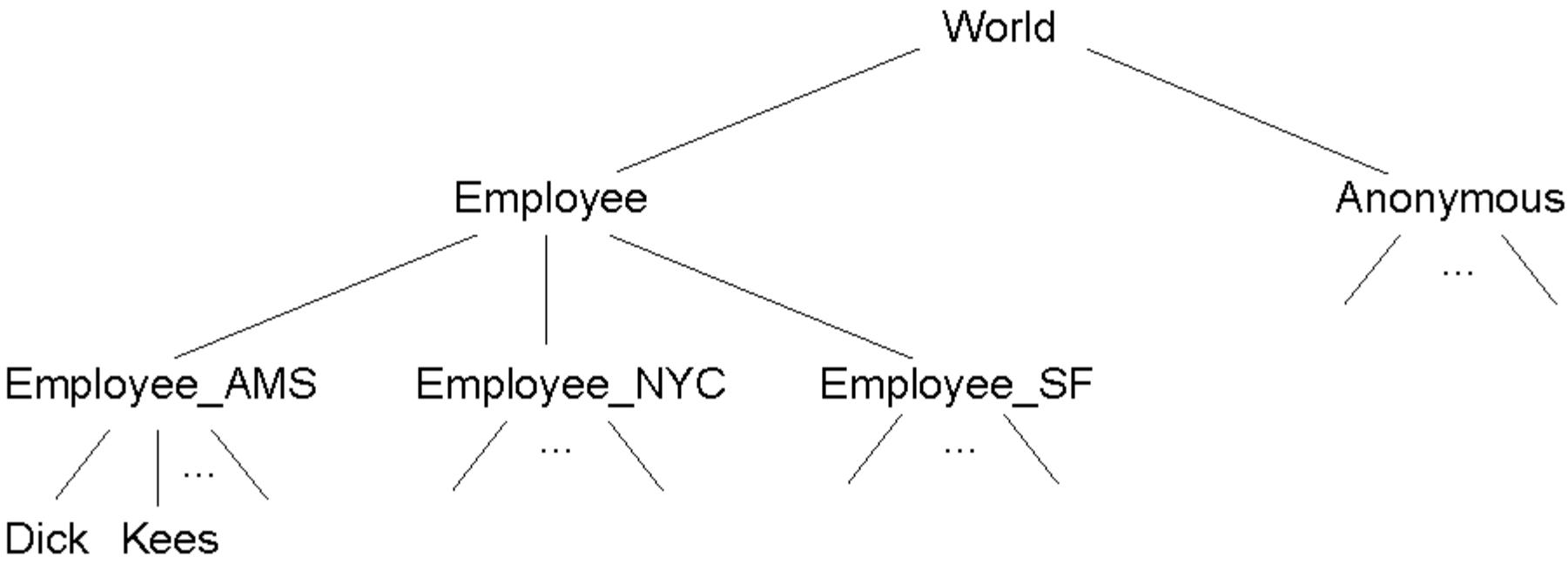
a) Using an ACL

b) Using capabilities.



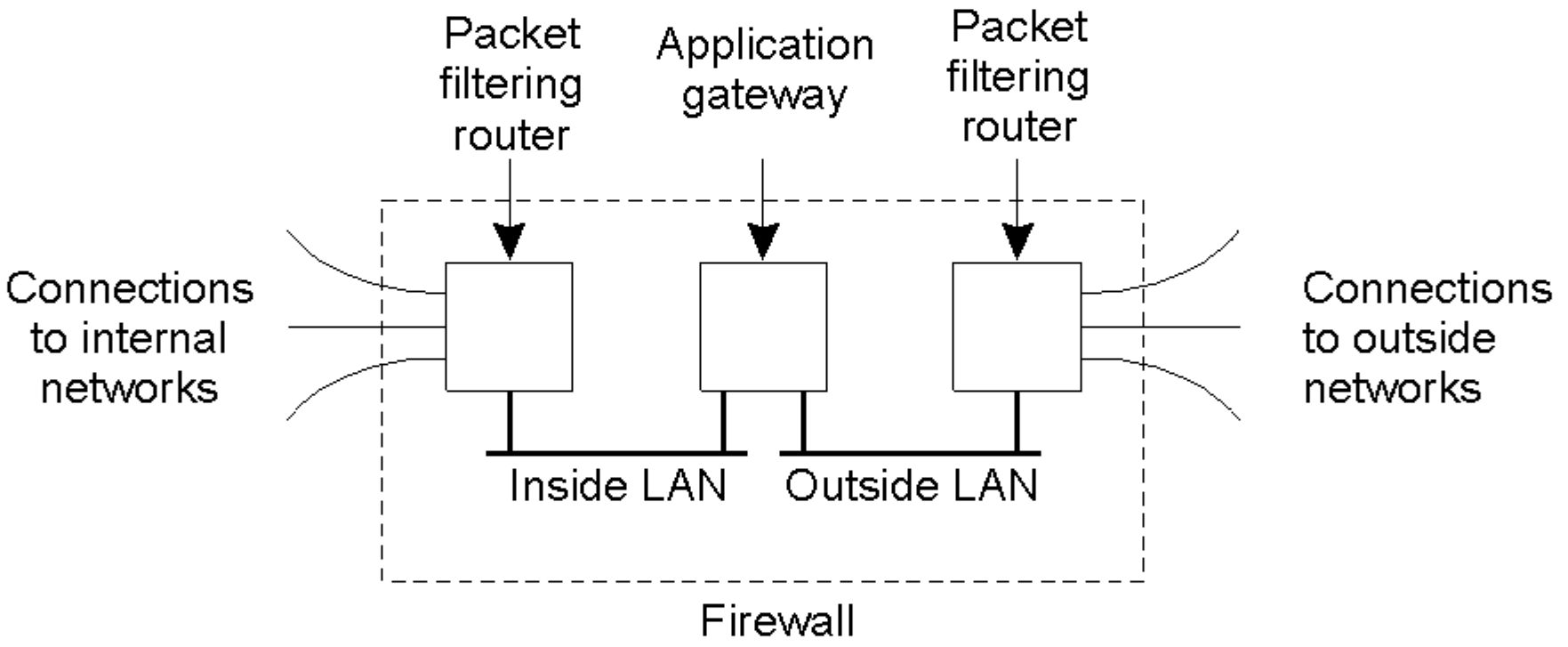


Protection Domains



The hierarchical organization of protection domains as groups of users.

Firewalls



A common implementation of a firewall.

Merci de votre attention