Authorisation in MAFTIA

Yves Deswarte

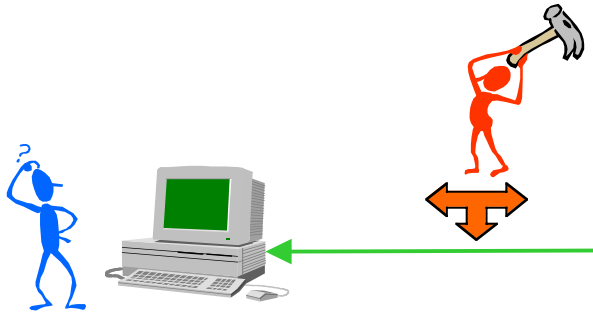
deswarte@laas.fr
LAAS-CNRS
Toulouse, France

MAFTIA Workplan

- WP1: Conceptual model and architecture
- WP2: Dependable middleware
- WP3: Intrusion detection
- WP4: Dependable trusted third parties
- WP5: Distributed authorisation
- WP6: Assessment

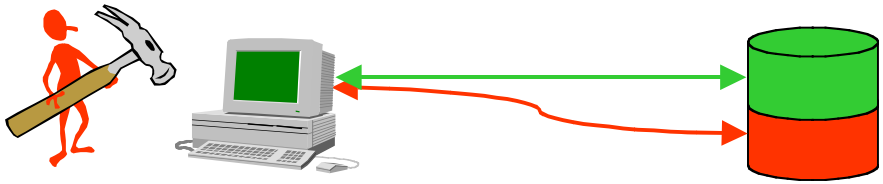
Who are the intruders?

1: "Outsider"



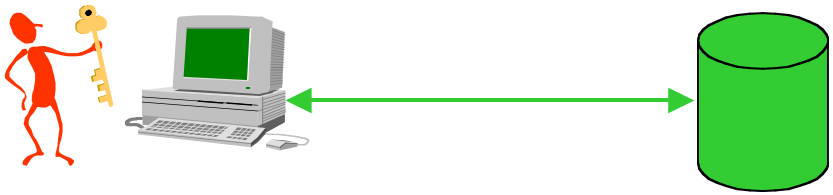
☺ Authentication
☺ Authorisation

2: User



☹ Authentication
☺ Authorisation

3: Privileged User



☹ Authentication
☹ Authorisation

because the "least privilege principle" is not implemented

Intrusion Tolerance

Intrusion into a part of the system should give access only to non-significant, non-sensitive information

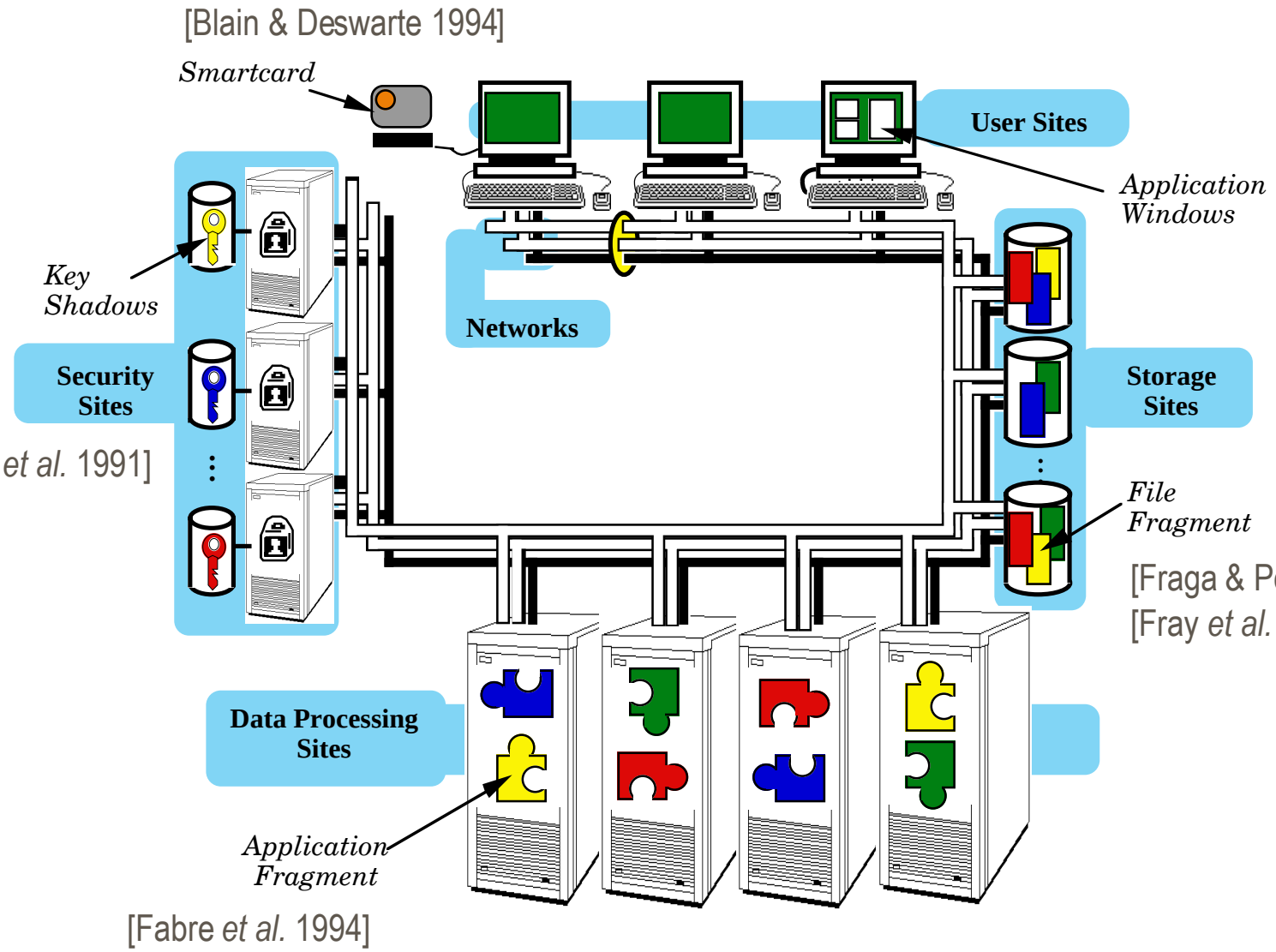
FRS: Fragmentation-Redundancy-Scattering

- Fragmentation: split the data into fragments so that isolated fragments contain no significant information: *confidentiality*
- Redundancy: add redundancy so that fragment modification or destruction would not impede legitimate access: *integrity + availability*
- Scattering: isolate individual fragments

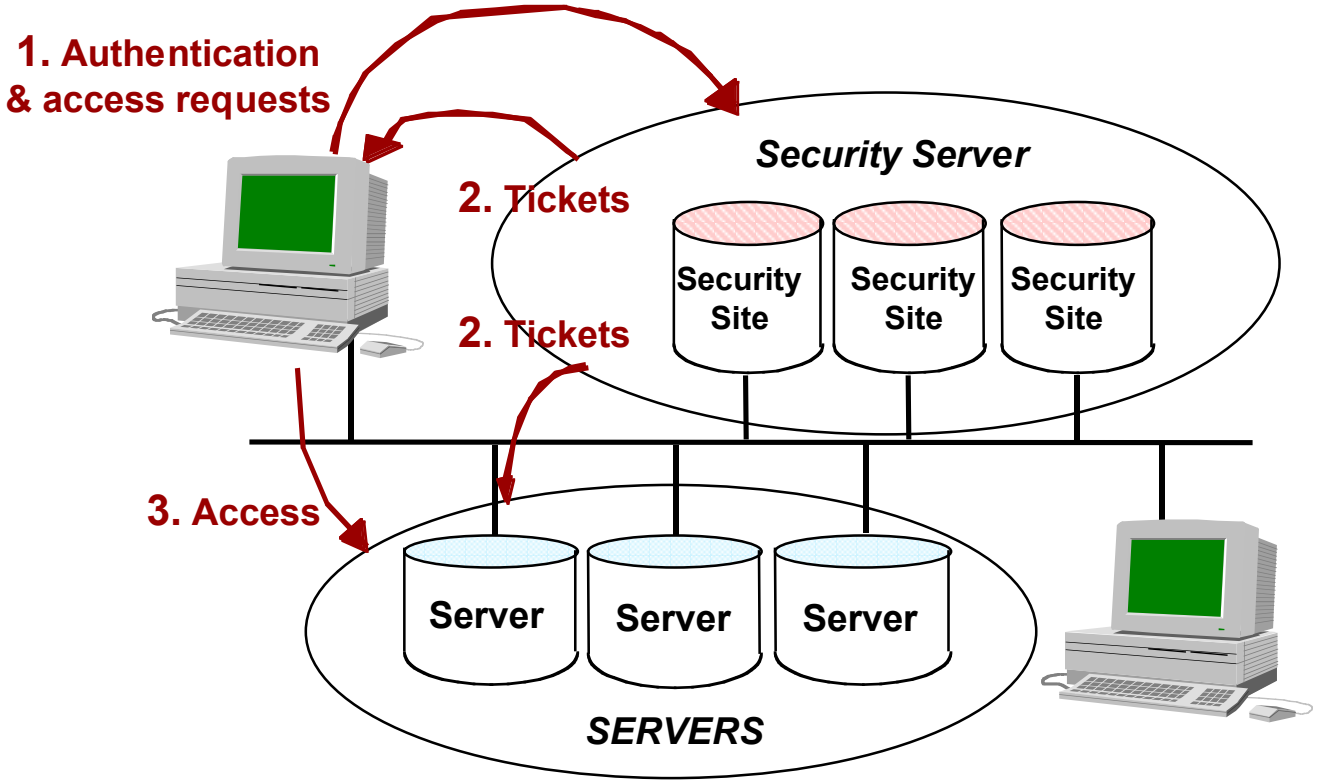
Different kinds of scattering

- Space: use different transmission links and different storage sites
- Time: mix fragments (from the same source, from different sources, with jamming)
- Frequency: use different carrier frequencies (spread-spectrum)
- Privilege: require the co-operation of differently privileged entities to realise an operation (separation of duty, secret sharing)

Prototype (1986-1996)

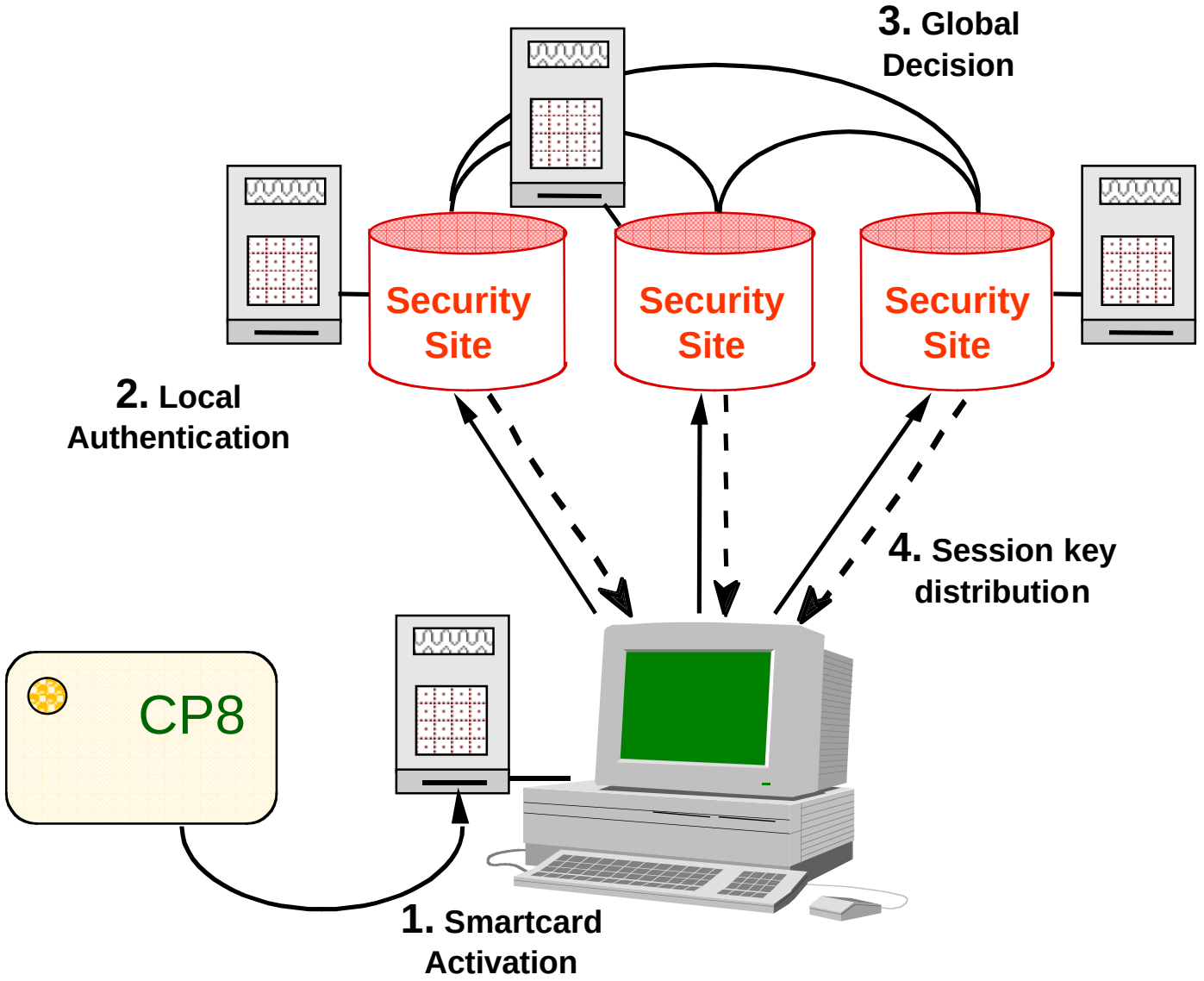


FRSed Security Management

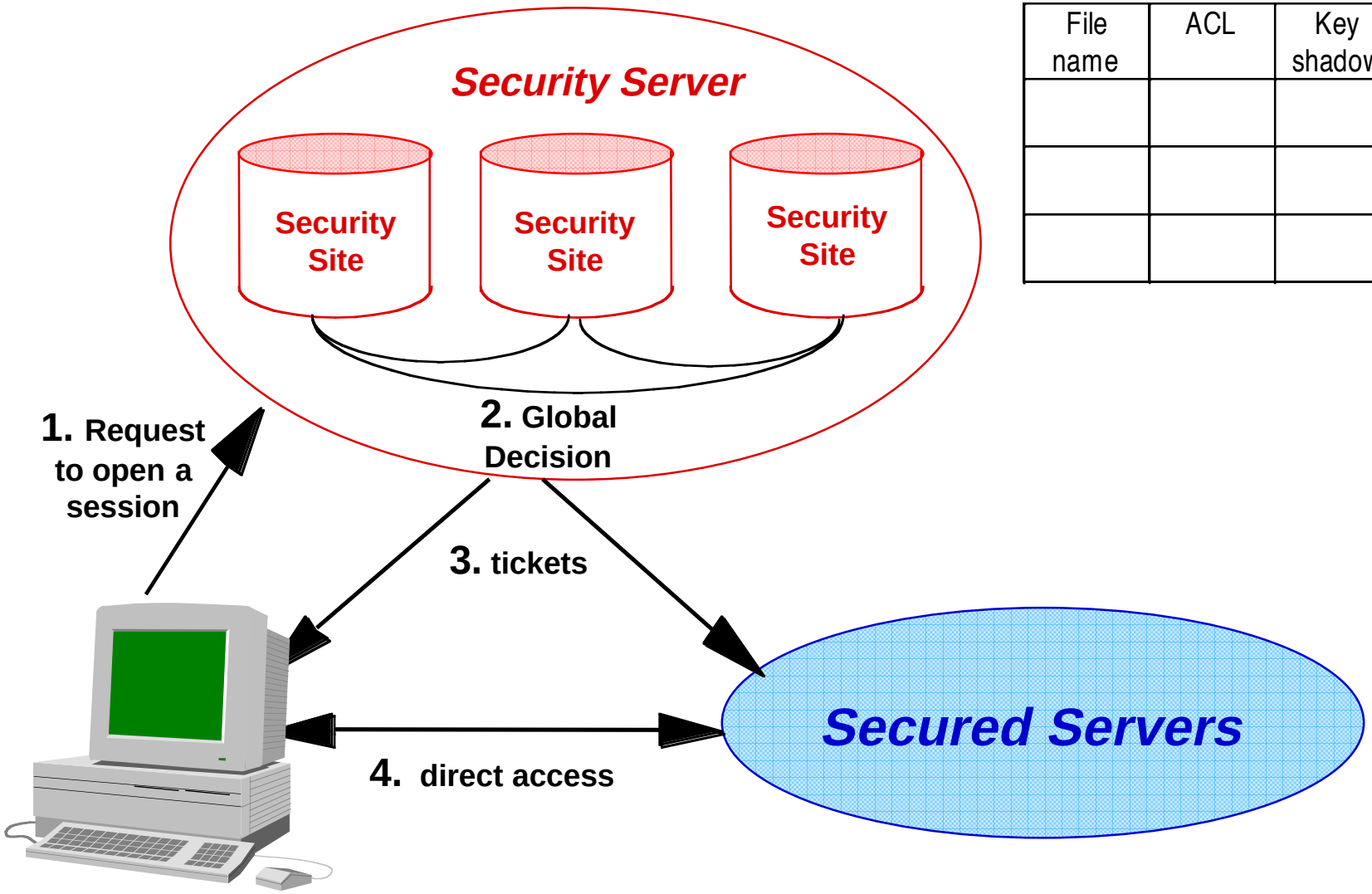


- No single trusted site or administrator
- Global trust in a majority of security sites (and administrators)

Authentication

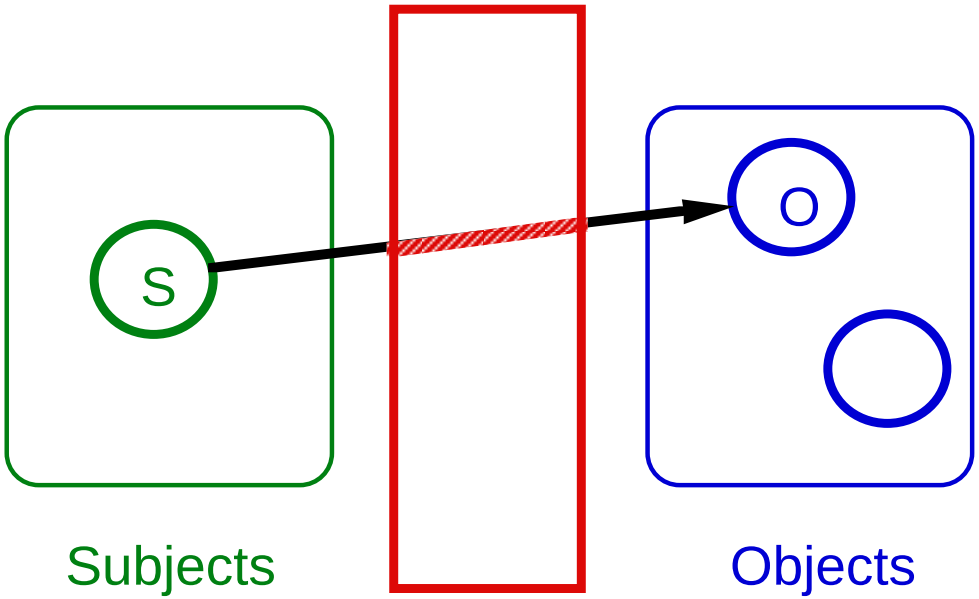


Authorisation



File name	ACL	Key shadow

Authorisation: reference monitor

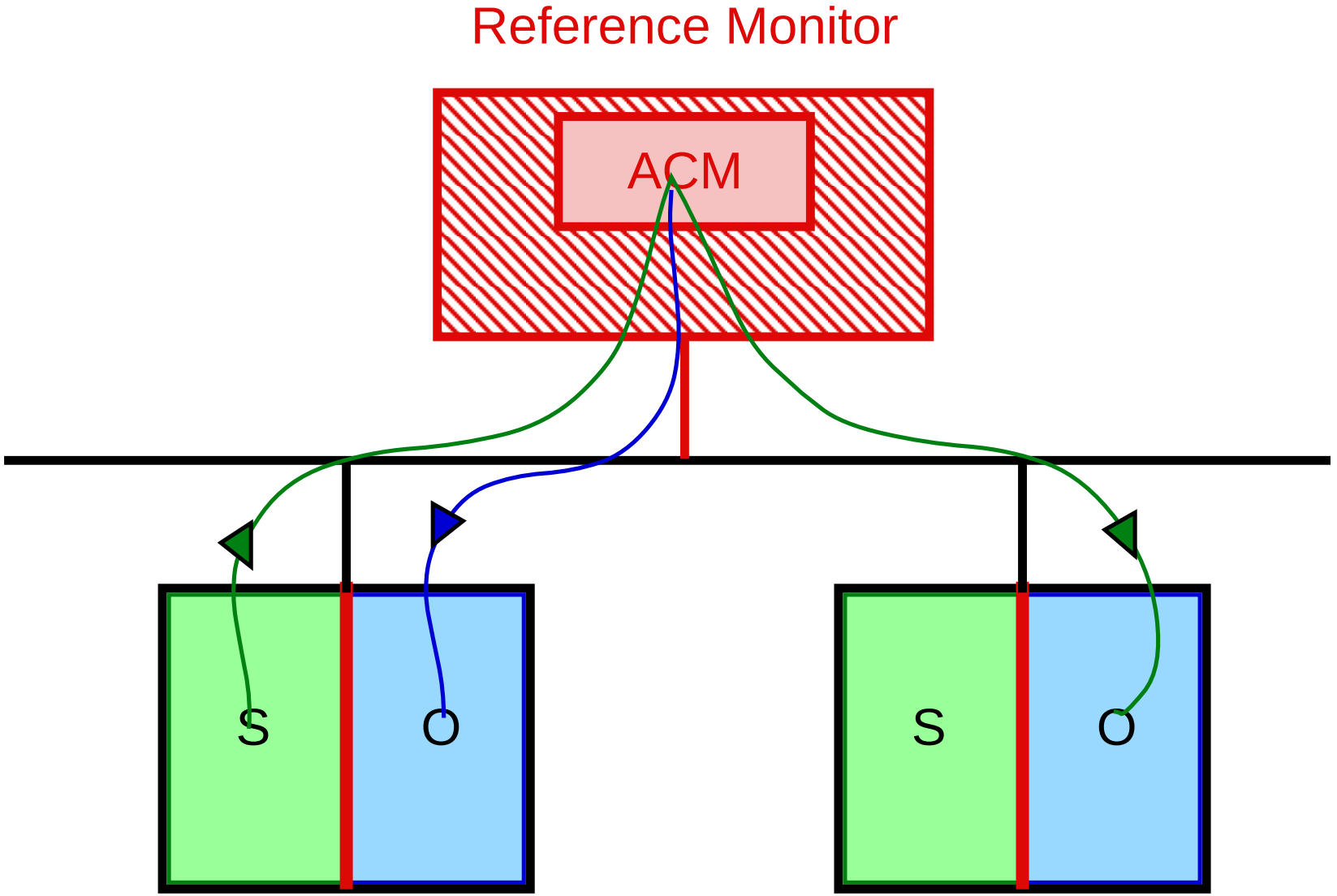


Subject (active)
≈ process

Object (passive)
≈ information containers

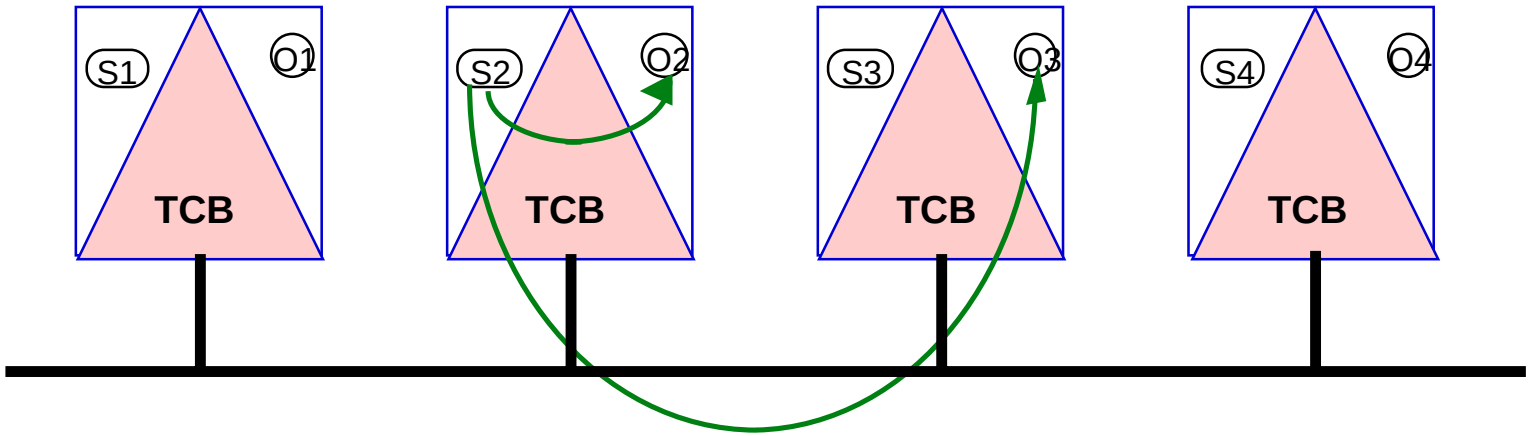
Reference Monitor:
- tamperproof
- always invoked
- verifiable as correct

Distributed Authorisation ? (1)



Distributed Authorisation ? (2)

Red Book (TNI)

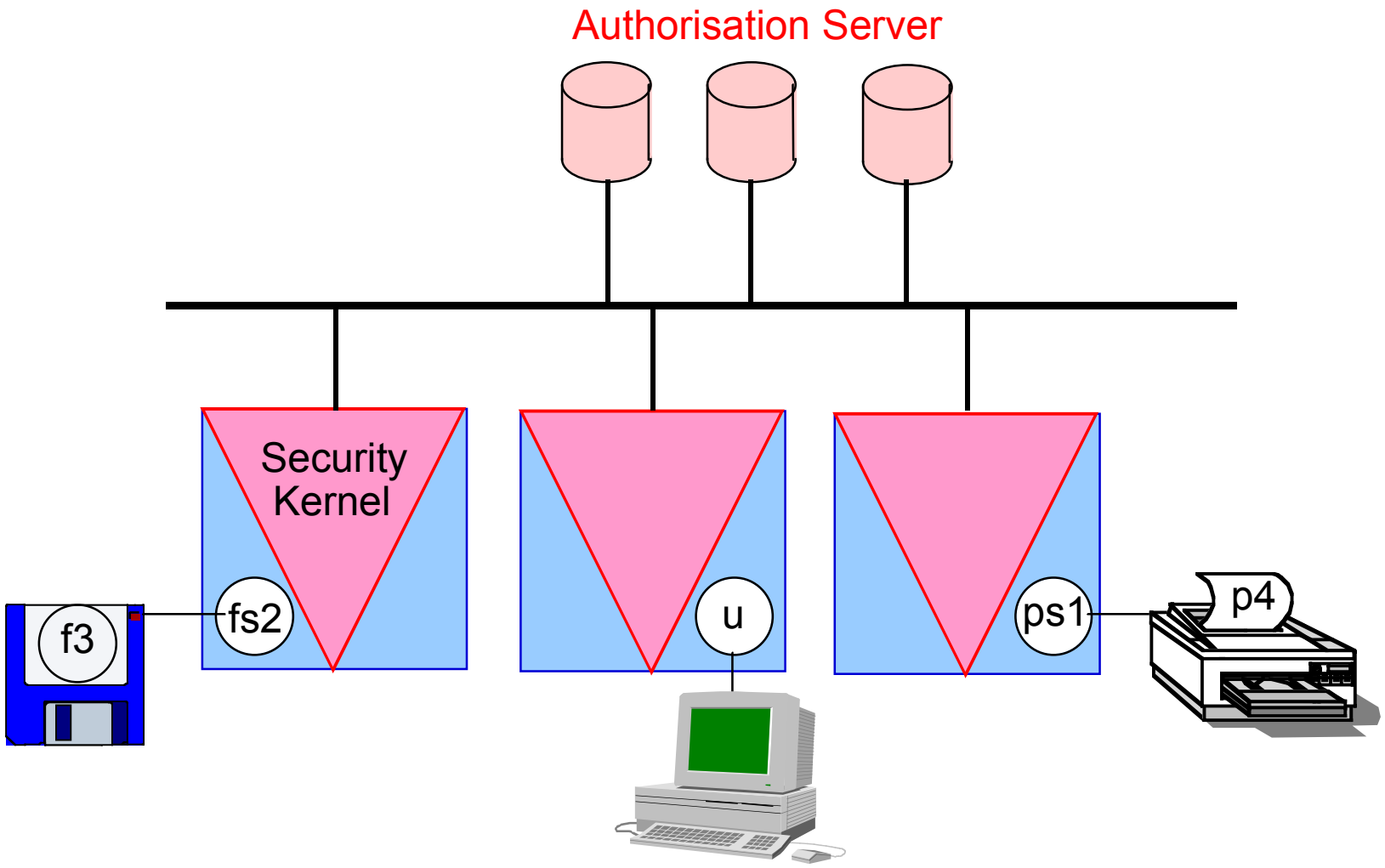


☺ No bottleneck, no single-point-of-failure

☹ Mutual trust between TCBs, consistency?

Authorisation Scheme for DOOS

[Nicomette & Deswarte 1997]



Authorisation Scheme

Access Matrix :

Method rights: corresponding to the authorisation for an object to call another object methods

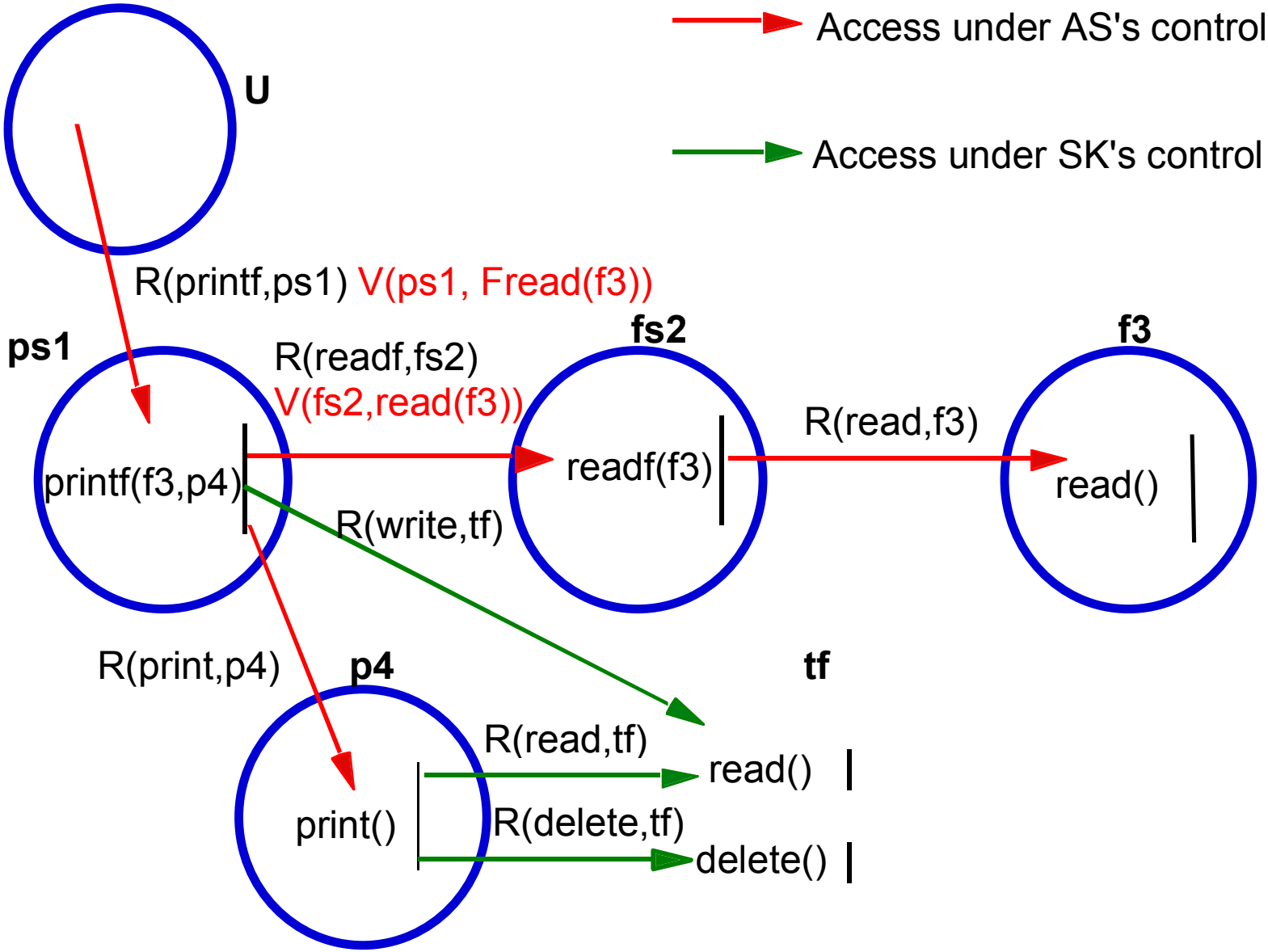
Symbolic rights: corresponding to the authorisation for an object to execute high level operations

	ps1	fs2	f 3	p 4
u			**PF(this,PRINTER)**	**PF(FILE,this)**
ps1				print
fs2				

Symbolic right rules: to check authorisation for high level operations

Capability creation rules: to grant capabilities and *vouchers* to enable high level operations

Example: u:: PF(f3,P4)



MAFTIA Authorisation

- Intrusion tolerant authorisation servers
- Multi-party transactions
(not simple client-server relations)
- Local protection

IT Authorisation Servers

Same technique as for FRS Security servers:

- Non-confidential information is replicated
- Confidential information is fragmented (threshold crypto)
- Global consensus (majority voting or Byzantine agreement)
- Distribution of capabilities/vouchers (threshold crypto)

Multi-party transactions

Based on DOOS Authorisation Scheme, but…

- Implementing "separation of duty" (transactions initiated jointly by several users)

Local protection

- Internet applications => no modification of user workstations
- No security kernel, but JVM (?)
- Capabilities/vouchers -> applets
- Possibly enforced by smartcards (e.g., JavaCards) … which are already necessary for authentication

Use cases / Scenarios

- Medical:

  French Healthcare Professional Network

  - Patient smartcard Vitale
  - Professional smartcard CPS
  - Electronic prescriptions
  - Network, PKI, anonymisation, delegation, …

- Electronic Commerce:

  Auction services, privacy enforced payment,…

References

- Blain, L. and Deswarte, Y. (1994). A Smartcard Fault-Tolerant Authentication Server, in *1st Smart Card Research and Advanced Application Conference (CARDIS'94),* Lille, France, pp.149-165.
- Deswarte, Y., Blain, L. and Fabre, J.-C. (1991). Intrusion Tolerance in Distributed Systems, in *Symp. on Research in Security and Privacy,* Oakland, CA, USA, pp.110-121.
- Deswarte, Y., Fabre, J.-C., Laprie, J.-C. and Powell, D. (1986). A Saturation Network to Tolerate Faults and Intrusions, in *5th Symp. on Reliability of Distributed Software and Database Systems,* Los Angeles, CA, USA, pp.74-81, IEEE Computer Society Press.
- Fabre, J.-C., Deswarte, Y. and Randell, B. (1994). Designing Secure and Reliable Applications using FRS: an Object-Oriented Approach, in *1st European Dependable Computing Conference (EDCC-1),* Berlin, Germany LNCS 852, pp.21-38.
- Fraga, J. and Powell, D. (1985). A Fault and Intrusion-Tolerant File System, in *IFIP 3rd Int. Conf. on Computer Security,* (J. B. Grimson and H.-J. Kugler, Eds.), Dublin, Ireland, Computer Security, pp.203-218.
- Fray, J.-M., Deswarte, Y. and Powell, D. (1986). Intrusion-Tolerance using Fine-Grain Fragmentation-Scattering, in *Symp. on Security and Privacy,* Oakland, CA, USA, pp.194-201.
- Nicomette, V. and Deswarte, Y. , An Authorization Scheme for Distributed Object Systems, in *Symp. on Security and Privacy,* Oakland, CA, USA, pp. 21-30.

Commercials

- MAFTIA: http://www.research.ec.org/maftia/
- SQUALE: http://www.research.ec.org/squale/
- Int. Conf. Dependable Systems and Networks (DSN 2000): 26-28 June 2000, New York
  http://www.dependability.org
  - Workshop on Dependability despite malicious faults
- Recent Advances in Intrusion Detection (RAID 2000): 2-4 October 2000, Toulouse
  http://www.raid-symposium.org
- European Symp. on Research in Computer Security (ESORICS 2000): 4-6 October 2000, Toulouse
  http://www.esorics.org