# Distributed Systems

Security

Chapter 9

# Course/Slides Credits

Note: all course presentations are based on those developed by Andrew S. Tanenbaum and Maarten van Steen. They accompany their "Distributed Systems: Principles and Paradigms" textbook (1st & 2nd editions).
http://www.prenhall.com/divisions/esm/app/author_tanenbaum/custom/dist_sys_1e/index.html

And additions made by Paul Barry in course CW046-4: Distributed Systems
http://glasnost.itcarlow.ie/~barryp/net4.html

# Outline

- Security Threats
- Security Policy
- Security Mechanisms
- Design Issues

# Security Threats

- Types of security threats to consider:
  - Interception
  - Interruption
  - Modification
  - Fabrication

# Security Threats : Interception

- The concept of interception refers to the situation that an unauthorized party has gained access to a service or data.
  - Example: where communication between two parties has been overheard by someone else.
- Interception also happens when data are illegally copied.
  - Example: after breaking into a person's private directory in a file system.

# Security Threats : Interruption

- An example of interruption is when a file is corrupted or lost.
- More generally interruption refers to the situation in which services or data become unavailable, unusable, destroyed, and so on.
  - Example: denial of service attacks by which someone maliciously attempts to make a service inaccessible to other parties is a security threat that classifies as interruption.

# Security Threats : Modification

- Modifications involve unauthorized changing of data or tampering with a service so that it no longer adheres to its original specifications.
  - Examples: intercepting and subsequently changing transmitted data, tampering with database entries, and changing a program so that it secretly logs the activities of its user.

# Security Threats : Fabrication

- Fabrication refers to the situation in which additional data or activity are generated that would normally not exist.
  - Examples
    - an intruder may attempt to add an entry into a password file or database.
    - break into a system by replaying previously sent messages.
- Note that interruption, modification, and fabrication can each be seen as a form of data falsification.

# Security Policy

- Simply stating that a system should be able to protect itself against all possible security threats is not the way to actually build a secure system.
- What is first needed is a description of security requirements, that is, a *security policy.*
- A security policy describes precisely which actions the entities in a system are allowed to take and which ones are prohibited.
- Once a security policy has been laid down, it becomes possible to concentrate on *the security mechanisms* by which a policy can be enforced.

# Security Mechanisms

- Important security mechanisms are:
  1. Encryption
  2. Authentication
  3. Authorization
  4. Auditing

# Security Mechanisms: Encryption

- Encryption is fundamental to computer security.
- Encryption transforms data into something an attacker cannot understand.
- In other words
  - Encryption provides a means to implement data confidentiality.
- In addition, encryption allows us to check whether data have been modified.
- It thus also provides support for integrity checks.

# Security Mechanisms: Authentication

- Authentication is used to verify the claimed identity of a user, client, server, host, or other entity.
- In the case of clients, the basic premise is that before a service starts to perform any work on behalf of a client, the service must learn the client's identity (unless the service is available to all).
- Typically, users are authenticated by means of passwords, but there are many other ways to authenticate clients.

# Security Mechanisms: Authorization

- After a client has been authenticated, it is necessary to check whether that client is authorized to perform the action requested.
- Example: Access to records in a medical database is a typical.Depending on who accesses the database, permission may be granted to read records, to modify certain fields in a record, or to add or remove a record.

# Security Mechanisms: Auditing

- Auditing tools are used to trace which clients accessed what, and in which way.
- Although auditing does not really provide any protection against security threats.
- Audit logs can be extremely useful for the analysis of a security breach, and subsequently taking measures against intruders.
- For this reason, attackers are generally keen not to leave any traces that could eventually lead to exposing their identity.
- In this sense, logging accesses makes attacking sometimes a riskier business.

# The Globus Security Architecture

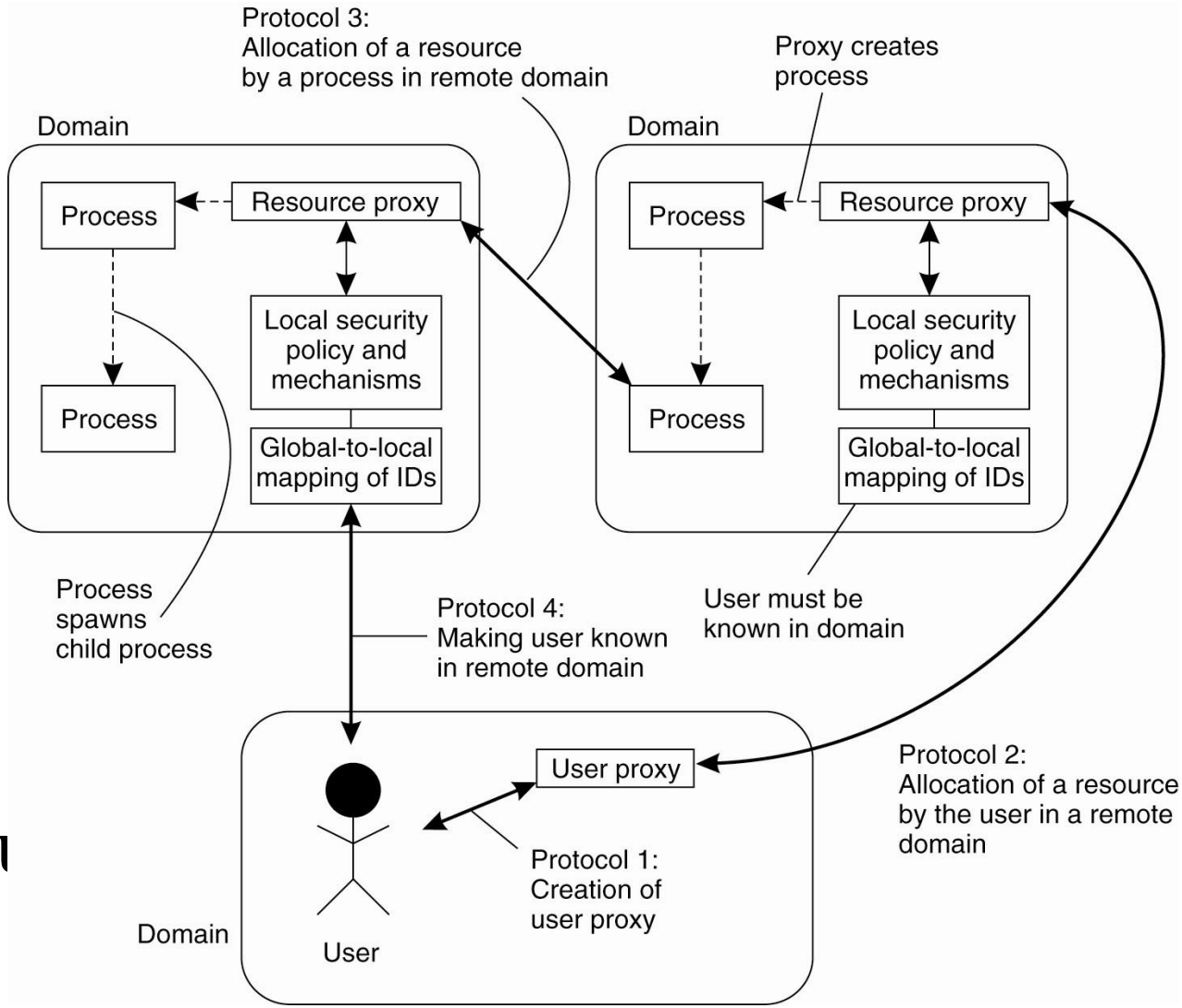
1. The environment consists of multiple administrative domains.
2. Local operations are subject to a local domain security policy only.
3. Global operations require the initiator to be known in each domain where the operation is carried out.

# Example: The Globus Security Architecture (2)

4. Operations between entities in different domains require mutual authentication.
5. Global authentication replaces local authentication.
6. Controlling access to resources is subject to local security only.
7. Users can delegate rights to processes.
8. A group of processes in the same domain can share credentials.

# Example: The Globus Security Architecture (2)

The Globus security architecture



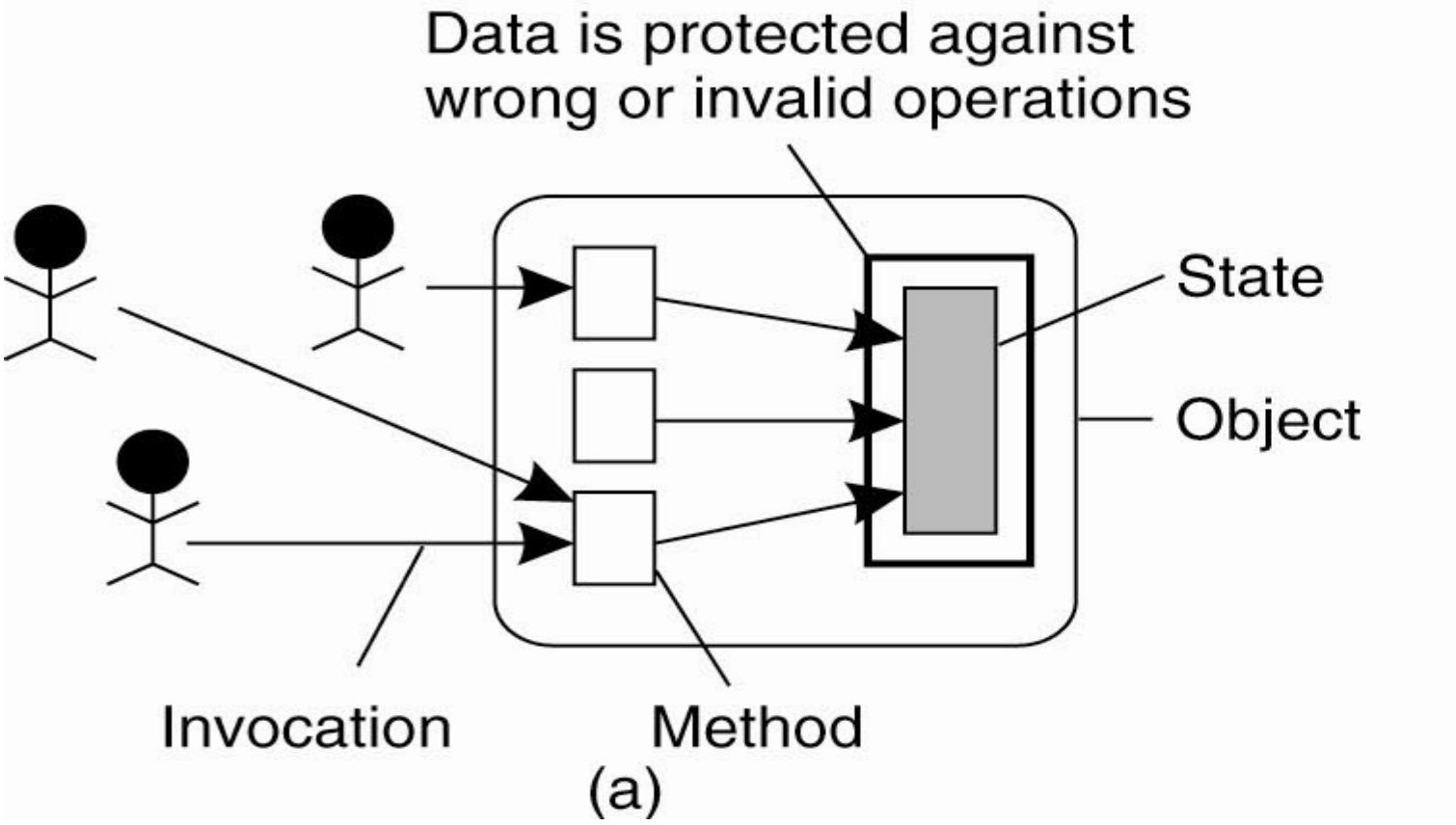
# Design Issues

- A distributed system, or any computer system for that matter, must provide security services by which a wide range of security policies can be implemented.
- There are a number of important design issues that need to be taken into account when implementing general-purpose security services.
- We discuss three of these issues: focus of control, layering of security mechanisms, and simplicity

# Design Issues: Focus of Control

- When considering the protection of a (possibly distributed) application, thereare essentially three different approaches that can be followed:
- The first approach: concentrate directly on the protection of the data that is associated with the application.
  - By direct, we mean that irrespective of the various operations that can possibly be performed on a data item, the primary concern is to ensure data integrity.
  - Typically, this type of protection occurs in database systems in which various integrity constraints can be formulated that are automatically checked each time a data item is modified.

# Design Issues: Focus of Control



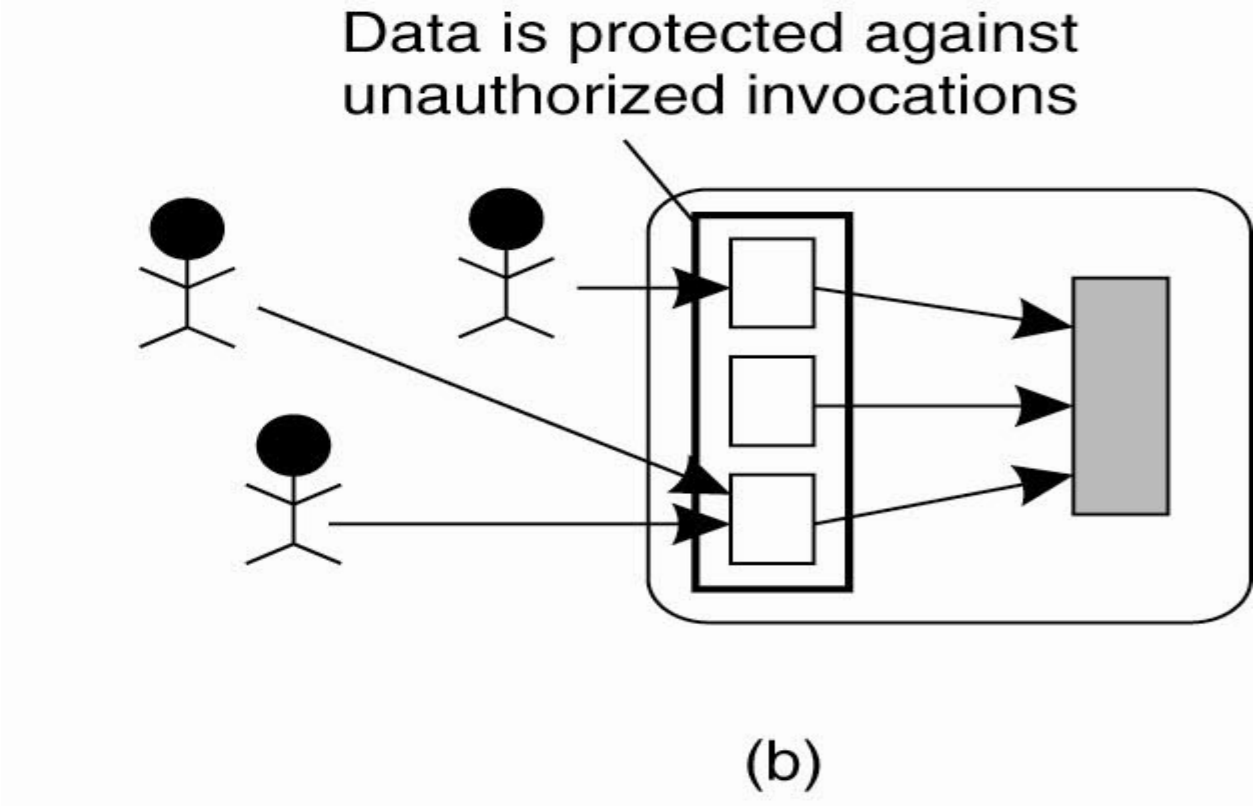

Three approaches for protection against security threats.
(a) Protection against invalid operations.

# Design Issues: Focus of Control

The second approach is to concentrate on protection by specifying exactly which operations may be invoked, and by whom, when certain data or resources are to be accessed.

- In this case, the focus of control is strongly related to access control mechanisms.
- For example, in an object-based system, it may be decided to specify for each method that is made available to clients which clients are permitted to invoke that method.
- Alternatively, access control methods can be applied to an entire interface offered by an object, or to the entire object itself.
- This approach thus allows for various granularities of access control.

# Design Issues: Focus of Control

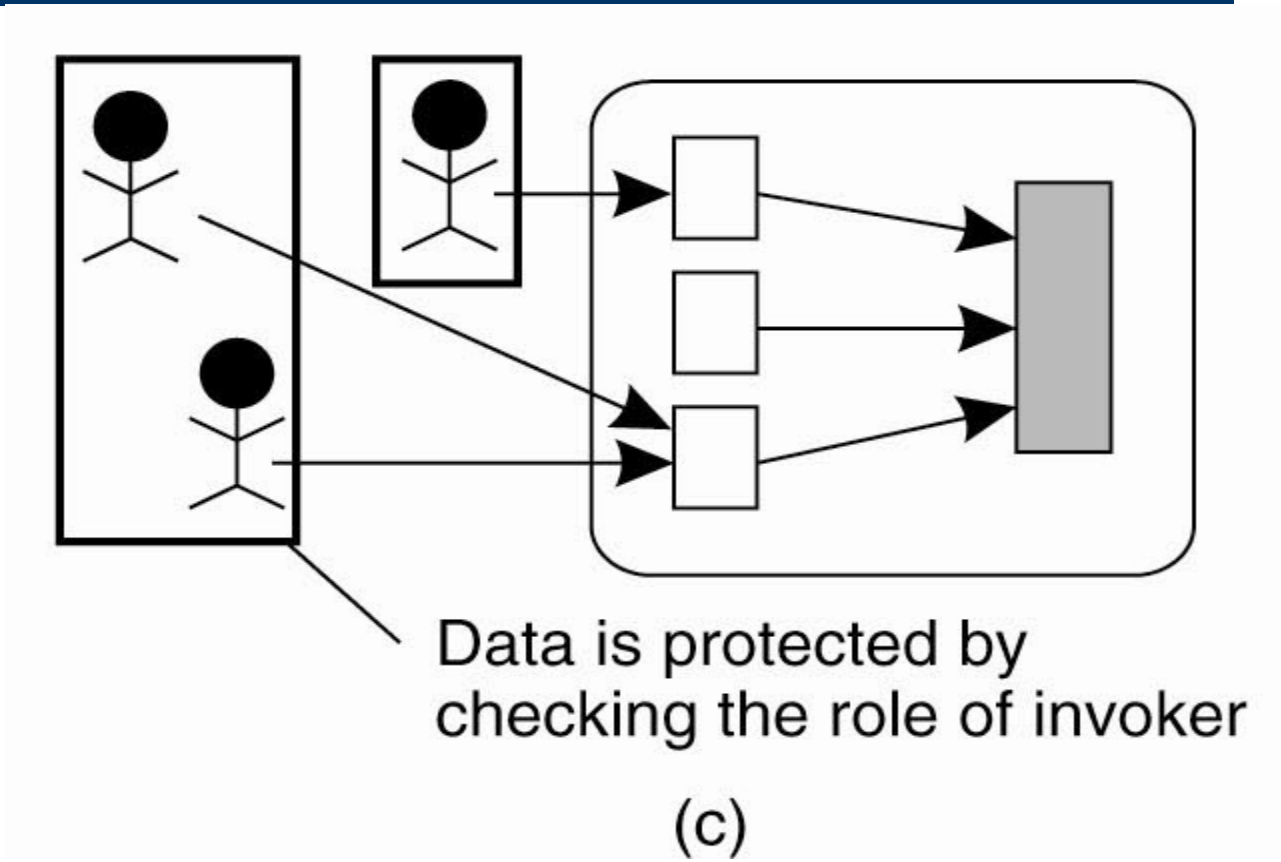



Three approaches for protection against security threats.
(b) Protection against unauthorized invocations.

# Design Issues: Focus of Control

- A third approach is to focus directly on users by taking measures by which only specific people have access to the application, irrespective of the operations they want to carry out.
- Example:
  - a database in a bank may be protected by denying access to anyone except the bank's upper management and people specifically authorized to access it.

# Design Issues: Focus of Control

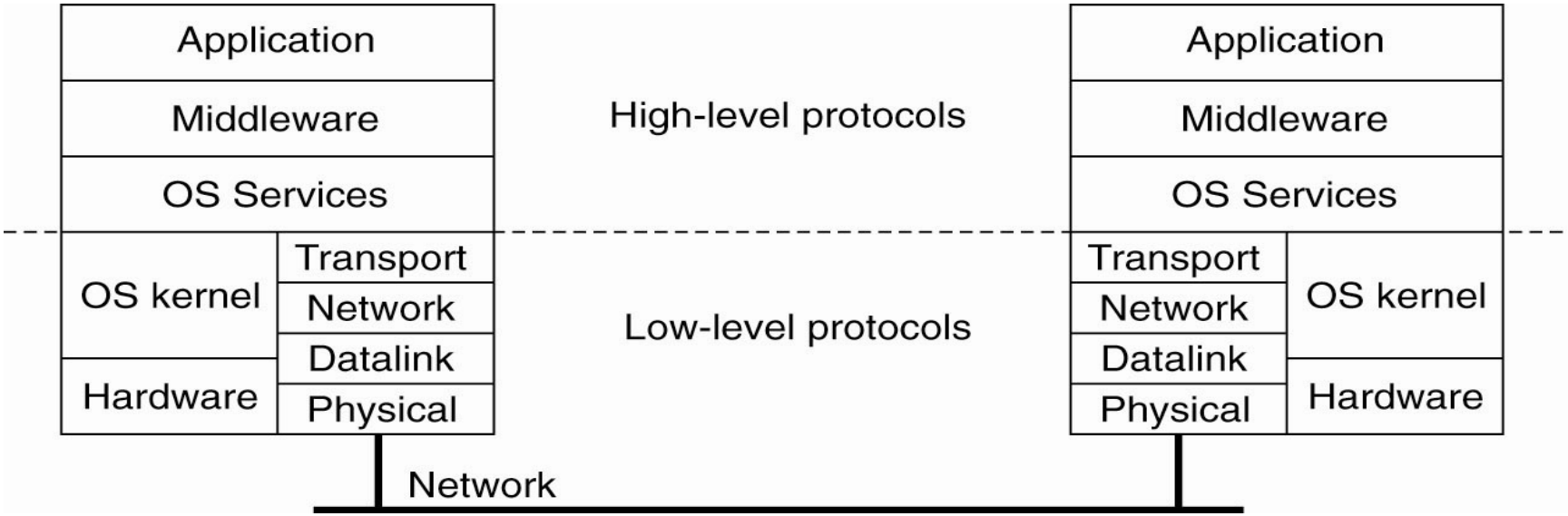



Three approaches for protection against security threats.
(c) Protection against unauthorized users.

# Design Issues: Layering of Security Mechanisms

- An important issue in designing secure systems is to decide at which level security mechanisms should be placed.
- A level in this context is related to the logical organization of a system into a number of layers.
- For example, computer networks are often organized into layers following some reference model.
- In Chapter 1, we introduced the organization of distributed systems consisting of separate layers for applications, middleware, operating system services, and the operating system kernel.

# Design Issues: Layering of Security Mechanisms




- The logical organization of a distributed system into several layers
- In distributed systems, security mechanisms are often placed in the middleware layer.
- Security services that are placed in the middleware layer of a distributed system can be trusted only if the services they rely on to be secure are indeed secure.

# Design Issues: Simplicity

- Another important design issue related to deciding in which layer to place security mechanisms is that of simplicity.
- Designing a secure computer system is generally considered a difficult task.
- Consequently, if a system designer can use a few, simple mechanisms that are easily understood and trusted to work, the better it is.