Seminar Review

Topics

- Grid Computing Middleware
- Grid Resource Management
- Grid Computing Security
- Applications of SOA and Web Services
- Semantic Grid
- Grid & E-Science
- Grid Economics
- Cloud Computing

Grid Computing Middleware

- Definitions & functions
- Middleware components
- Globus
- gLite

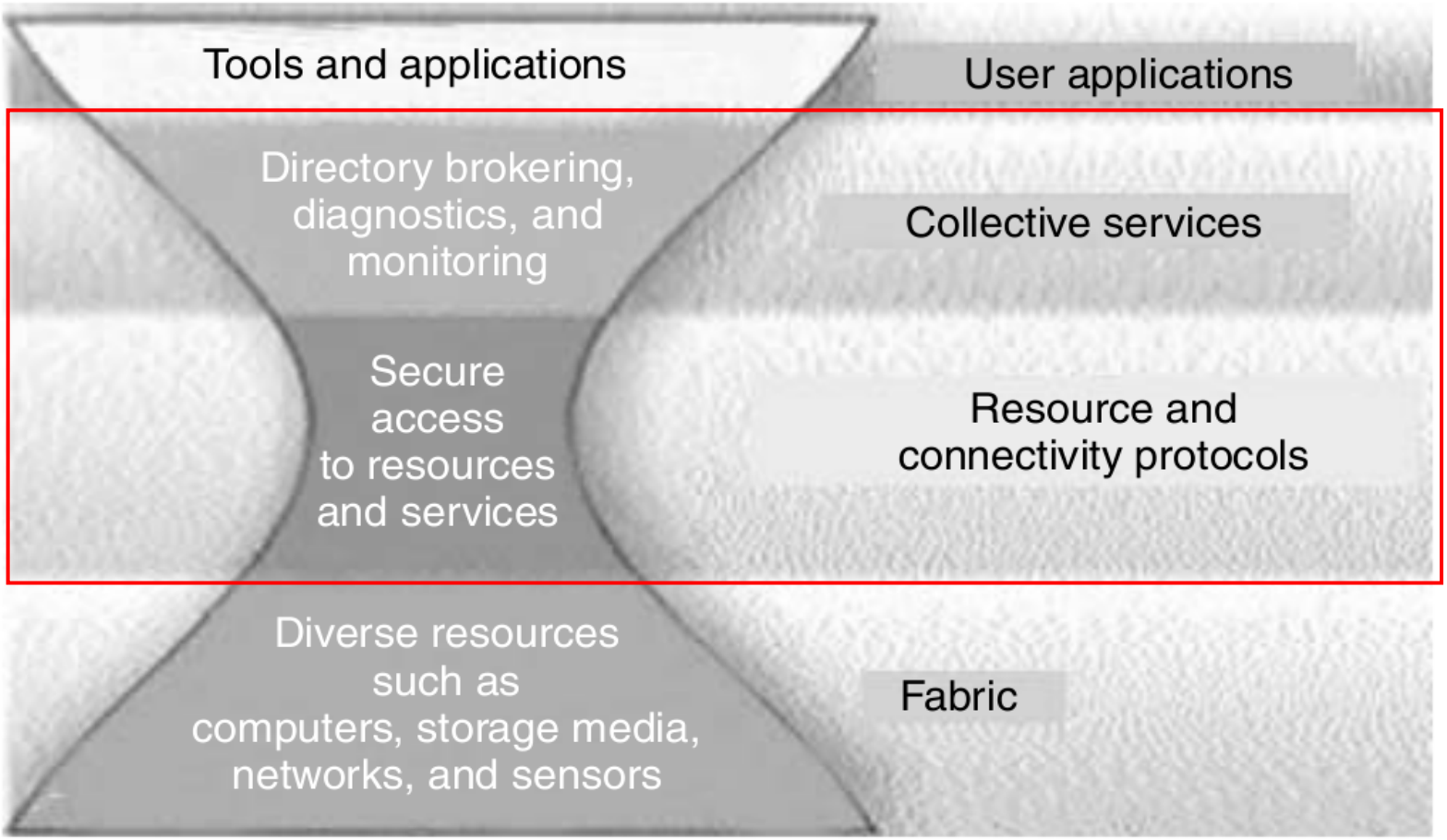
What is Grid middleware?

- System software between applications and OS
  - Provide services to applications
    - Discovery
    - Execution
    - Storage
    - Data movements
    - Information
    - Service integration
    - Failure detection and recovery
    - Resource monitoring
  - Hide all complexities of the Grid environment

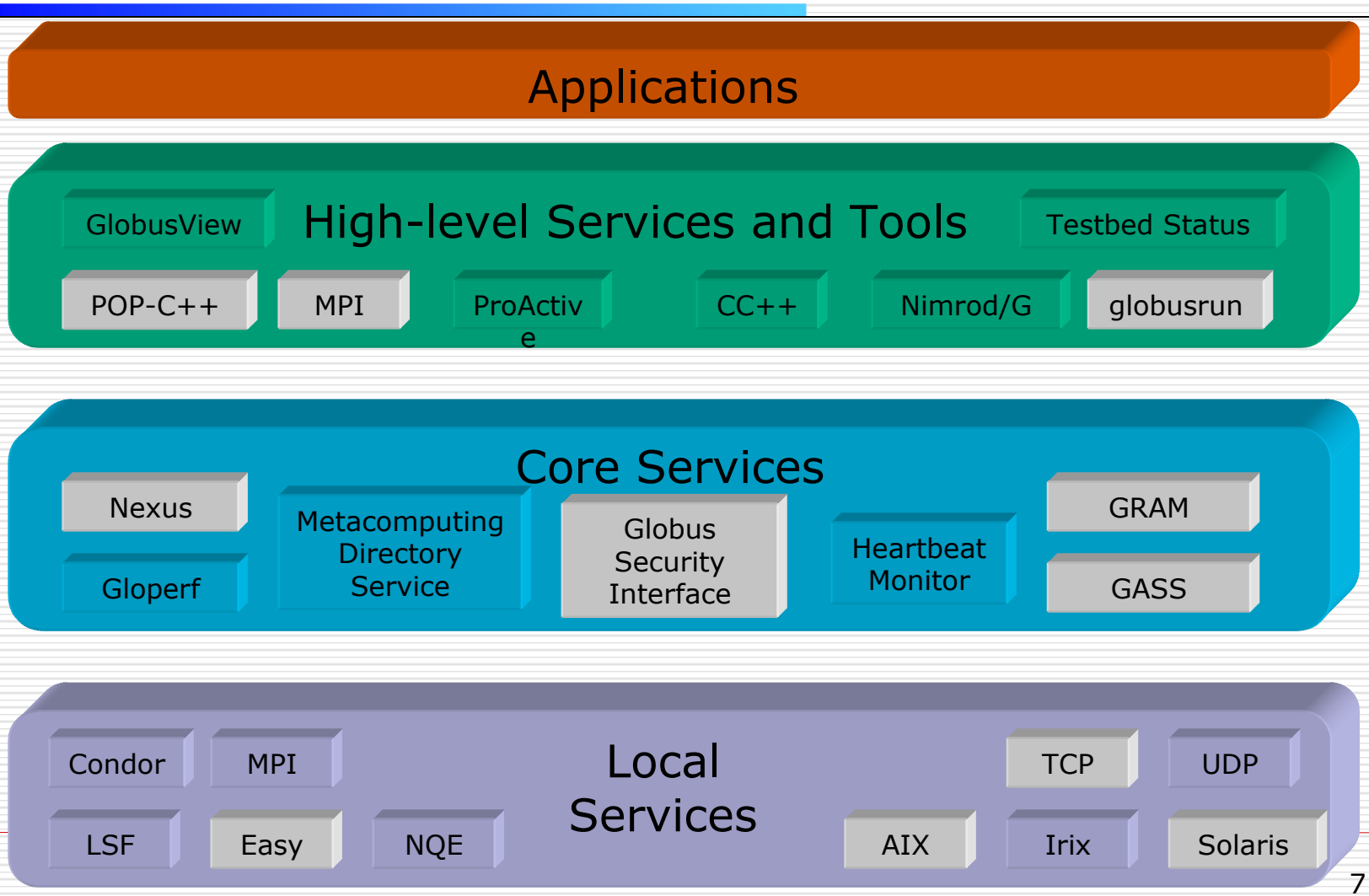
Purposes of Middleware

- Open, general-purpose and standard
- Standard protocols
  - Defines the contents and sequence of message exchanges used to request remote operation
  - Important and essential to achieve the interoperability that Grid depends on
- Standard APIs
  - Interfaces to code libraries
  - Facilitate construction of Grid components by allowing code components to be reused

Middleware Components



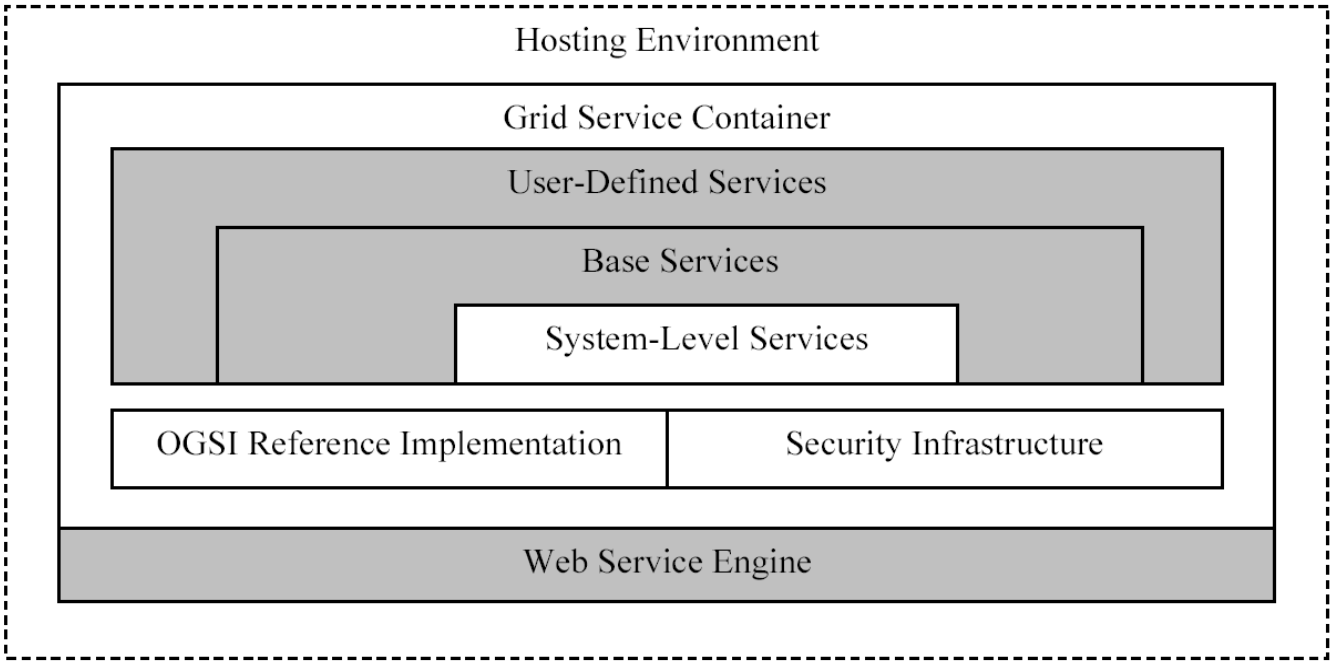
Globus layered architecture



Globus Toolkit 4x

- Sustainable changes on the services interoperability and infrastructure
  - Open Grid Services Architecture (OGSA)
    - Stateful Web Services
    - Enable the integration of user specific Grid services
  - Define standard interfaces
    - How to access Grid services
- Disadvantage
  - Slow

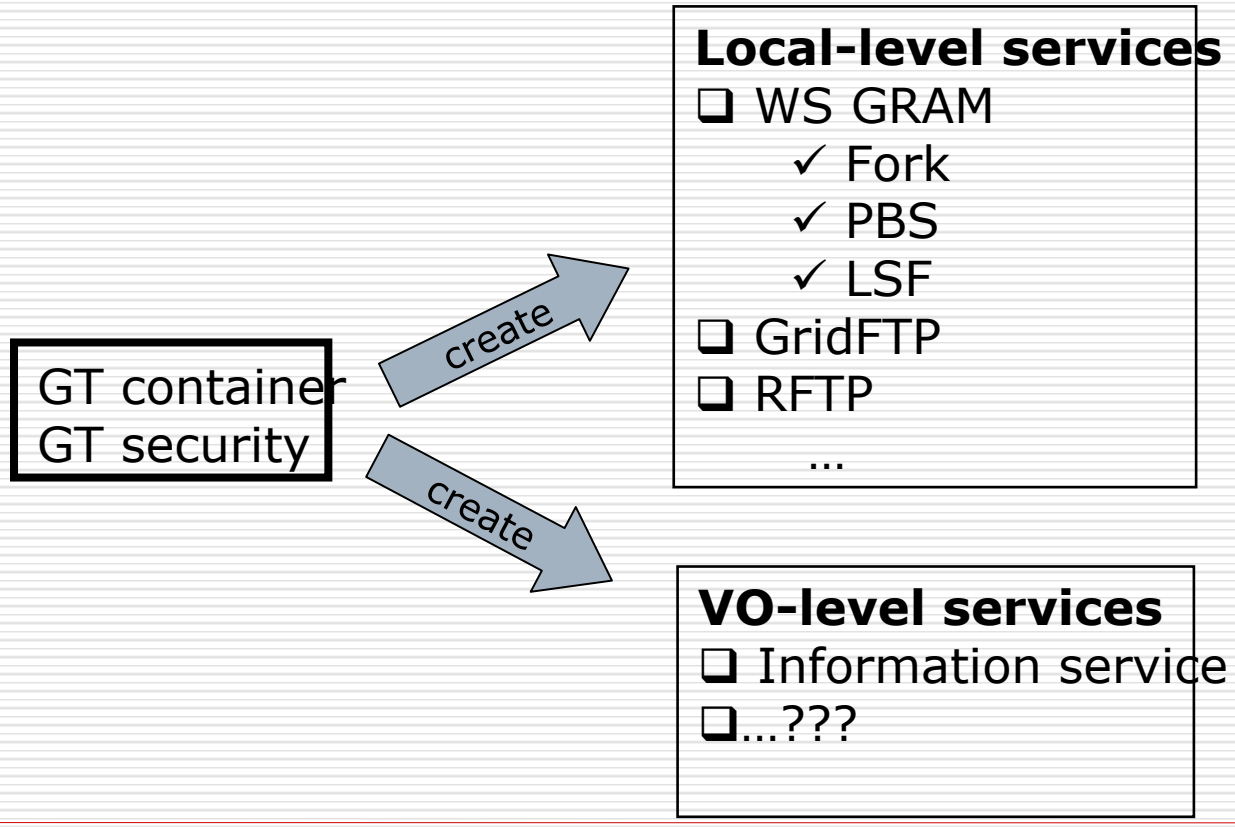
GT: Core service architecture



Globus Toolkit

- Grid Service Specification
  - How to write, publish and use a Grid service
- GT components:
  - GT core
    - Meta-services used to implement other services and service behaviors (e.g. service creation, destruction)
  - GT base services
    - Use the GT core to implement Grid capacities: resource management, information services, data transfer, etc.
  - Other Grid services
    - Implemented by user to enable some enhancement capacities

GT: from other perspectives



What GT DOES NOT address

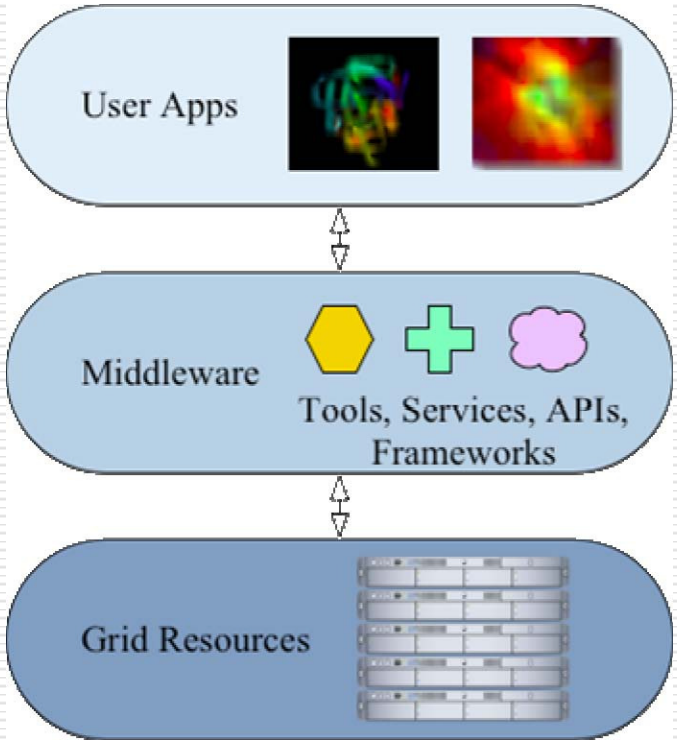
- GT focus on accessing local resources
- Things still missing
  - Coordination services
    - Resource/service discovery
  - Information collection
  - Resource connectivity
  - Programming models/tools
- Things to be improved
  - Performance!

gLite Toolkit

- The Enabling Grids for E-sciencE project (EGEE)
- Middleware stack that combines components developed in various related projects.

gLite middleware

- A layer between services and resources
- gLite follows a Service Oriented Architecture



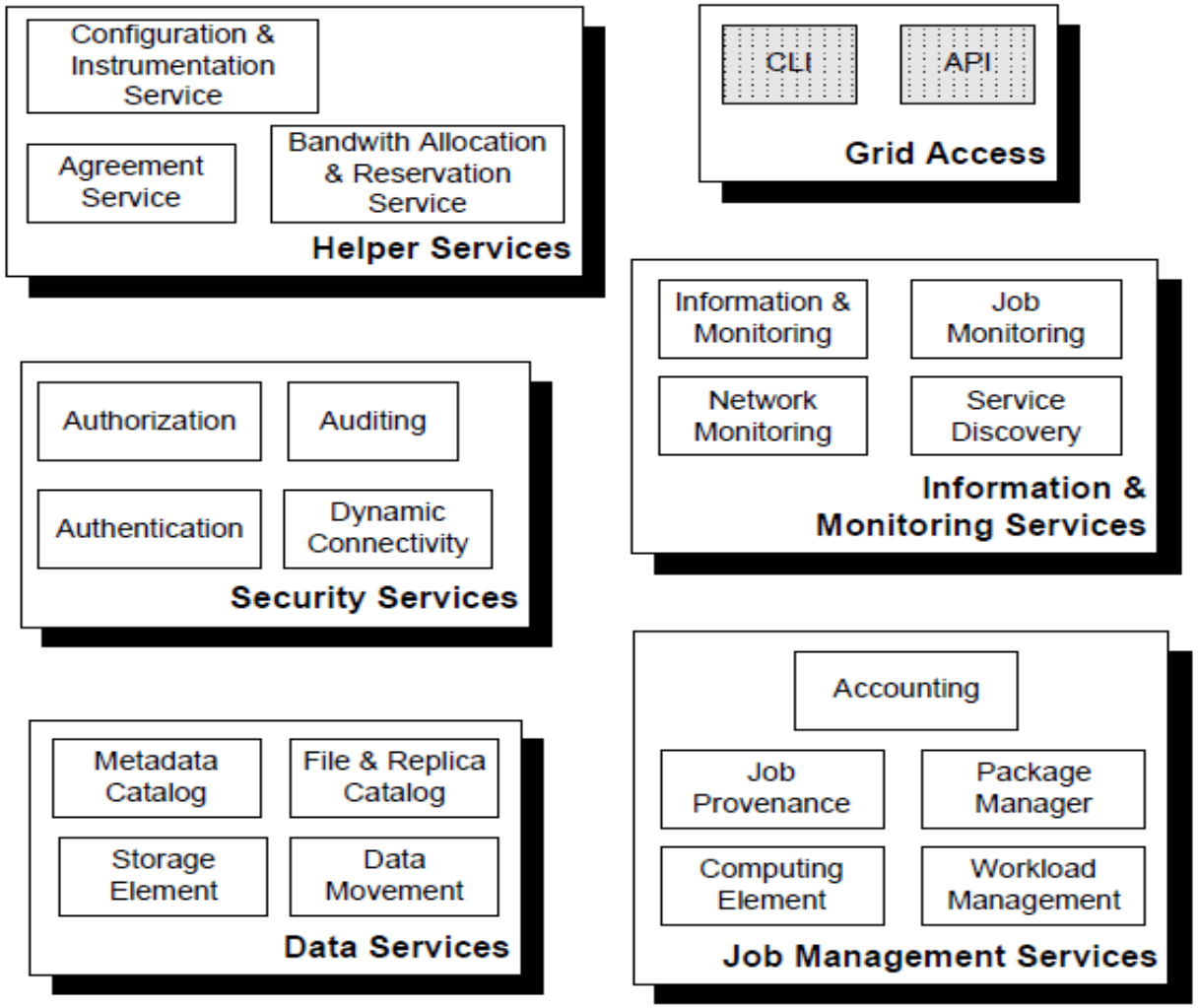
gLite - guiding principles

- Service oriented approach
  - Allow for multiple interoperable implementations
- Lightweight (existing) services
  - Easily and quickly deployable
  - Use existing services where possible
    - Condor, EDG, Globus, LCG, …
- Portable
  - Being built on Scientific Linux and Windows
- Security
  - Sites and Applications

gLite - guiding principles

- Performance/Scalability & Resilience/Fault Tolerance
  - Comparable to deployed infrastructure
- Co-existence with deployed infrastructure
  - Co-existence with LCG-2 and OSG (US) are essential for the EGEE Grid services
- Site autonomy
  - Reduce dependence on ‘global, central’ services
- Open source license

gLite Architecture



Middleware Comparison

	GT4	gLite	UNICORE
Authentication	X.509 Certificates		
Web Services	OGSA afford	no, due to 3.1 Manual	WSRF 1.2
I/O	GridFTP		?
API	C, Java, Python	C	Java
GUI	web	Scientific Linux 3/4 IA 32/64	java swing
Servers	unix-ish		unix-ish

Grid Resource Management

- Definitions
- Functions
- Resource Management Models

Resources

- Physical resource: computer, disk, database, networks, scientific instruments.
- Logical resource: Executing applications, Complex workflows…

Definition

- The term resource management refers to the operations used to control how capabilities provided by Grid resources and services are made available to other entities, whether users, applications or services.

Functions

- Discovery
- Allocations
- Negotiation
- Access

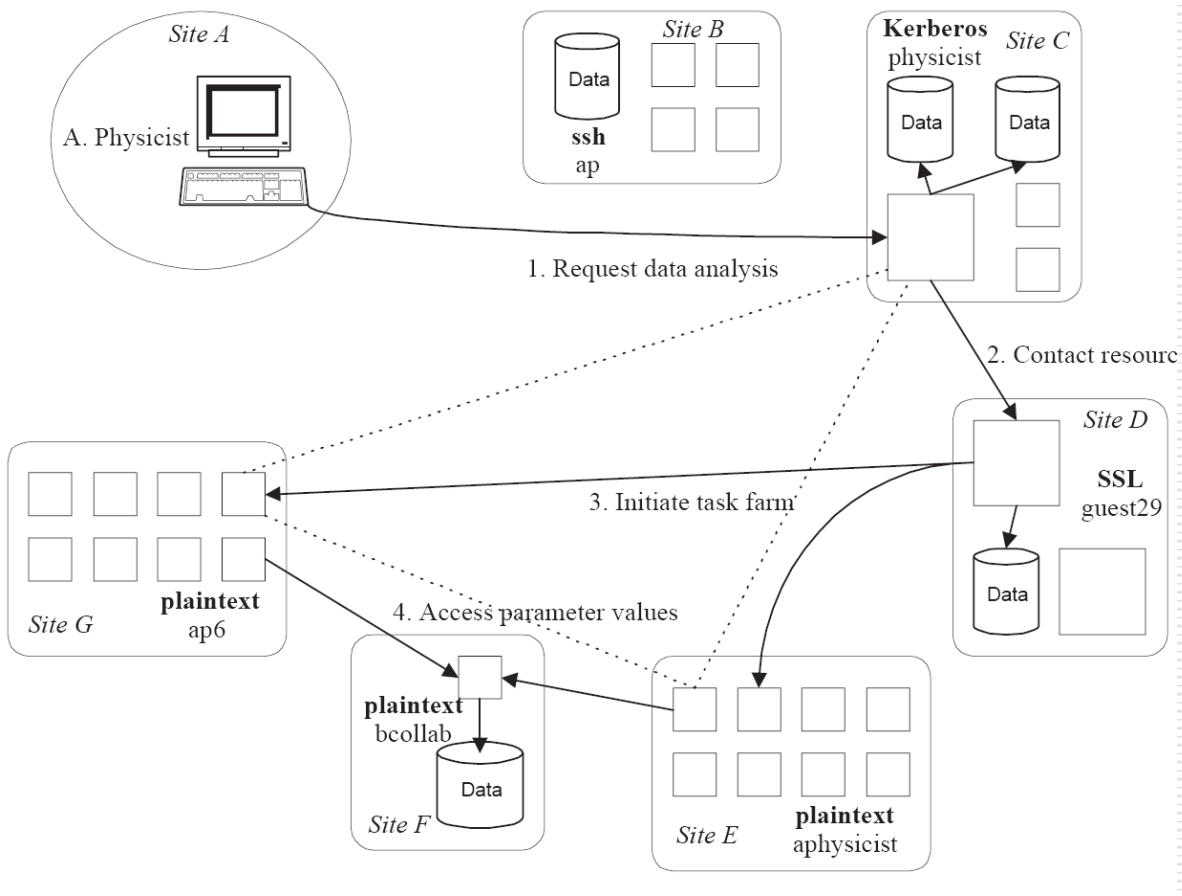
Models

- Hierarchical
- Abstract Owner
- Economics

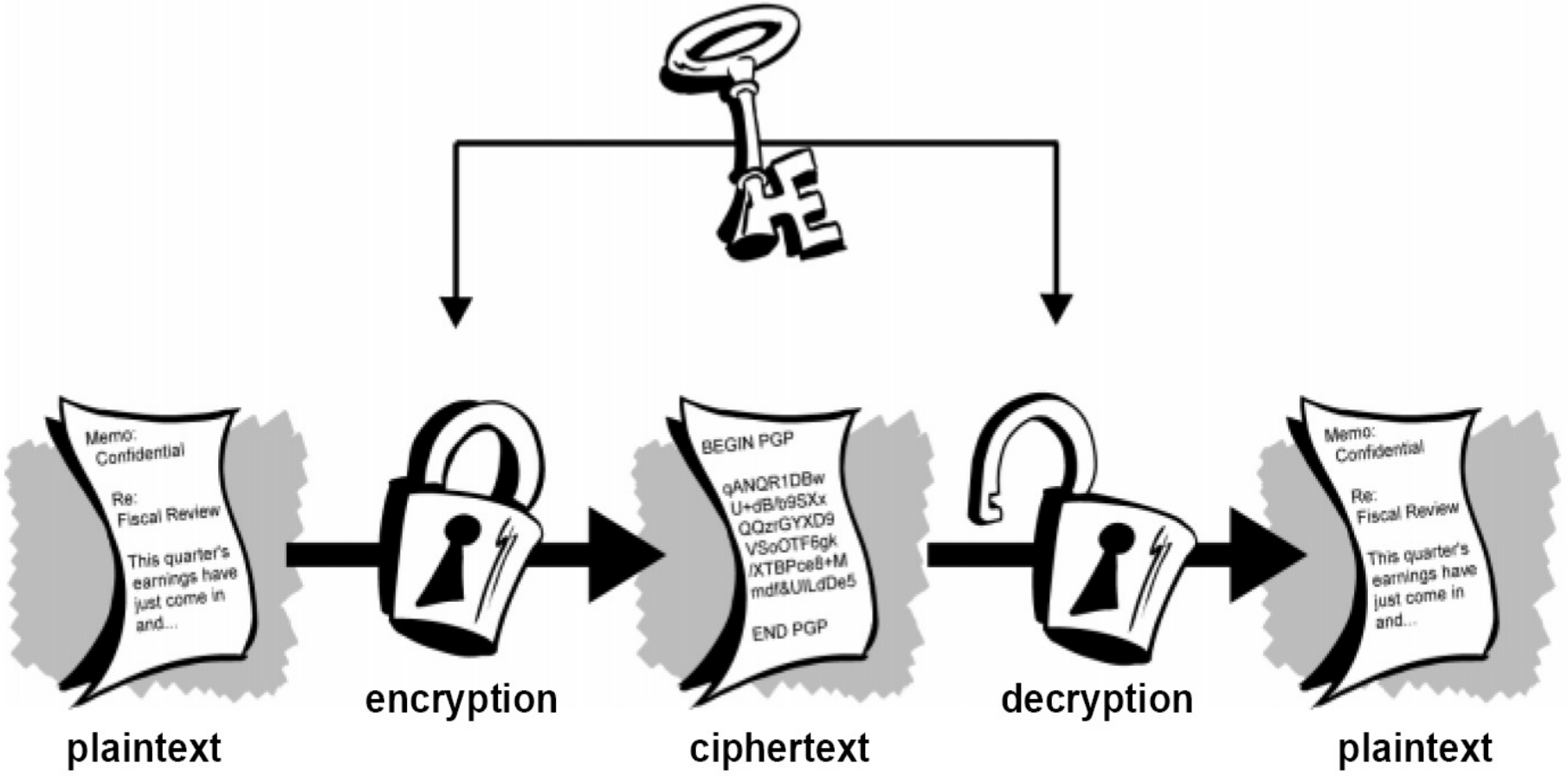
Grid Security

- Scenarios
- Symmetric keys & Asymmetric keys
- Digital signature
- Grid authentication and deligation

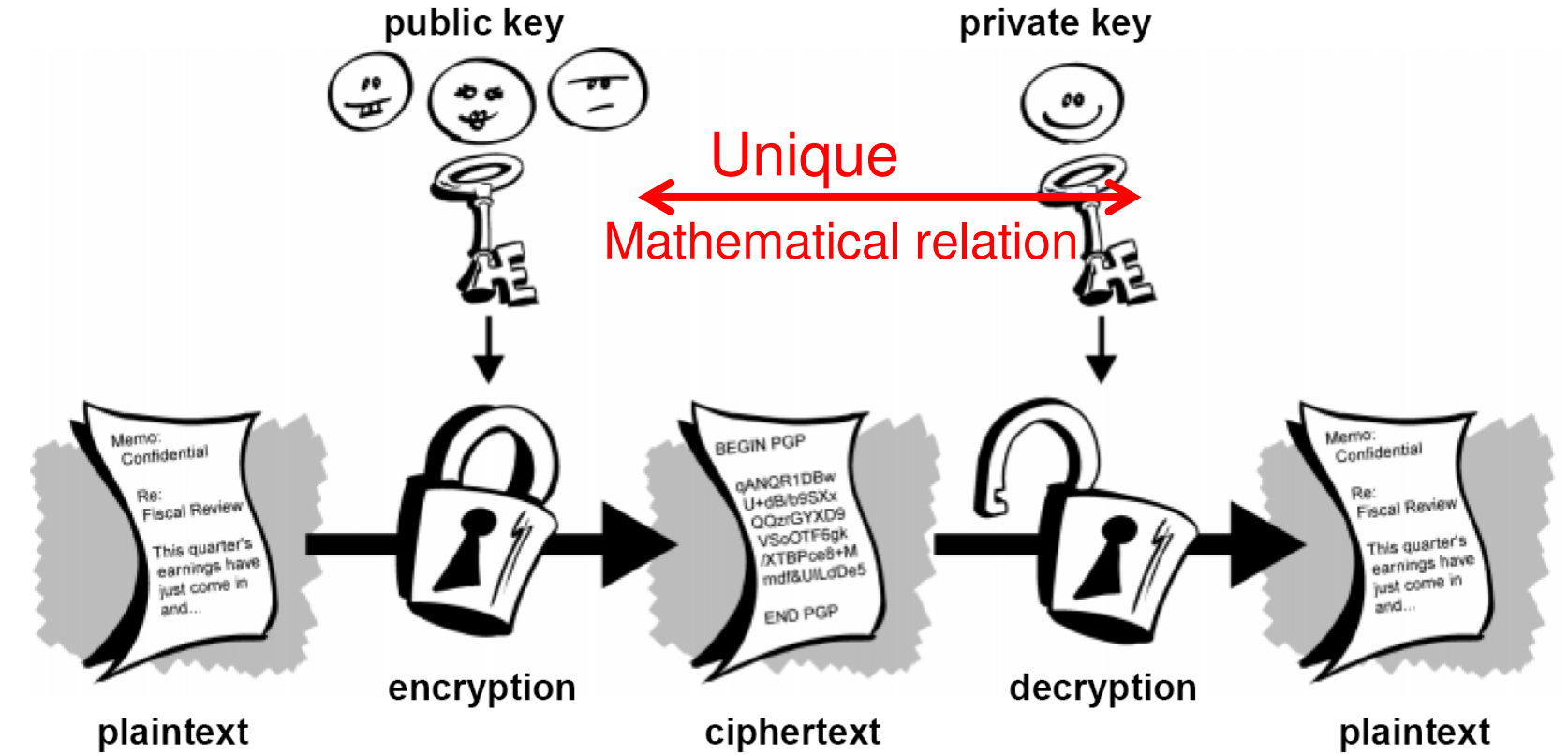
Grid security: a scenario



Symmetric key

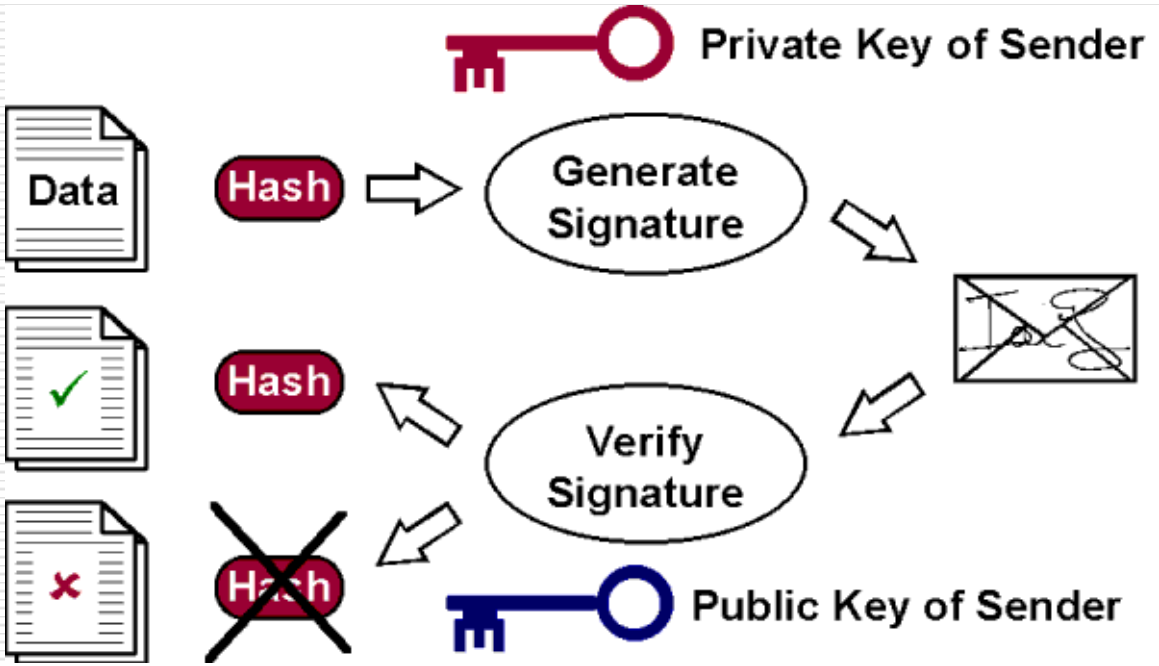


Asymmetric cryptography



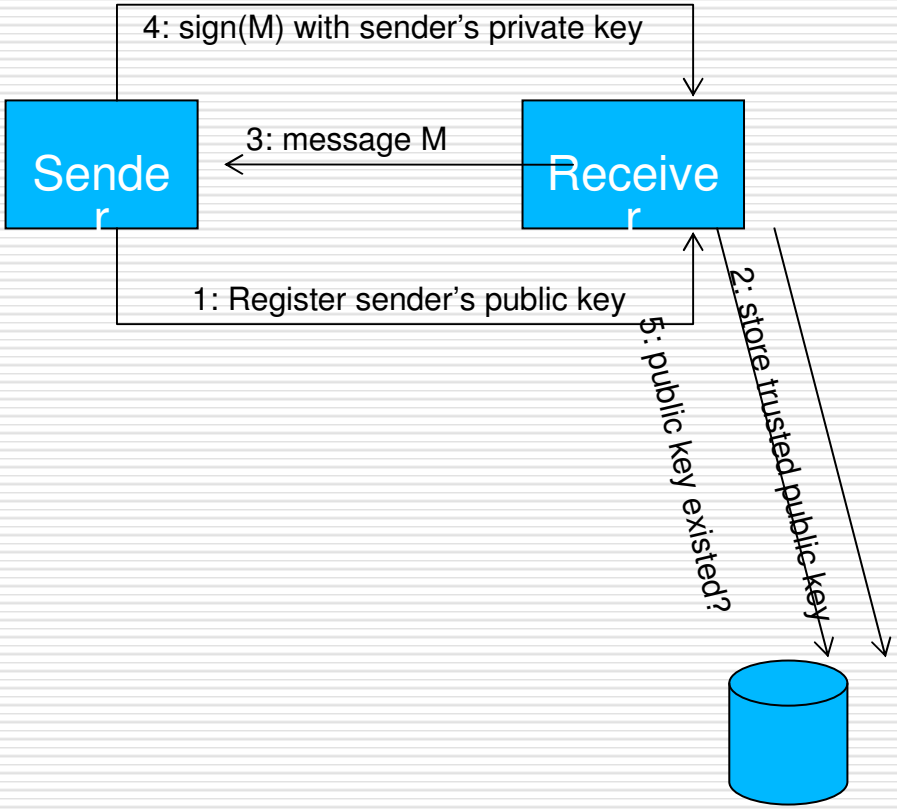
Digital Signatures

- A message signed by the private-key

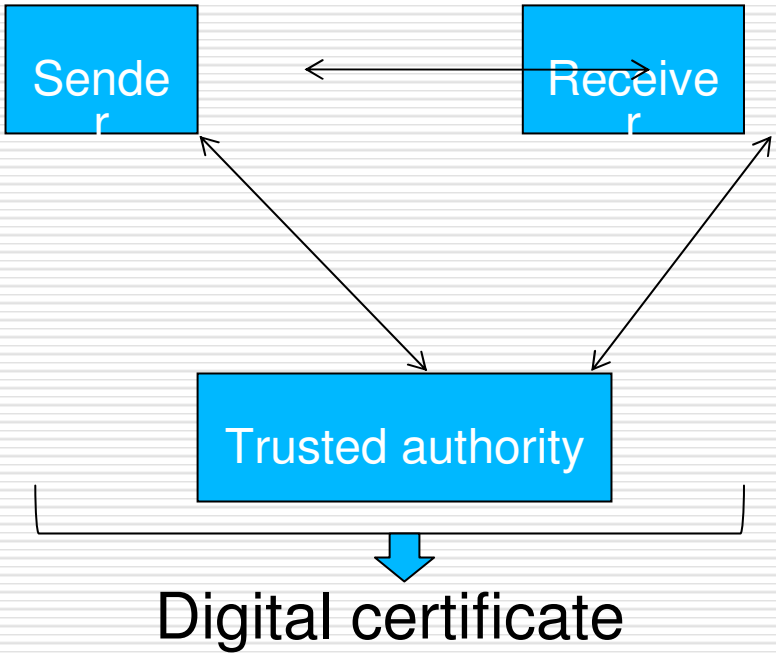


Authentication models

Direct-trusted model

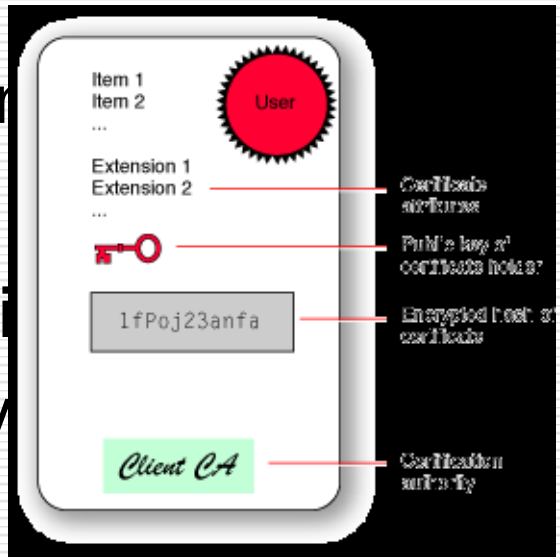


Third-party trusted model



Digital certificate

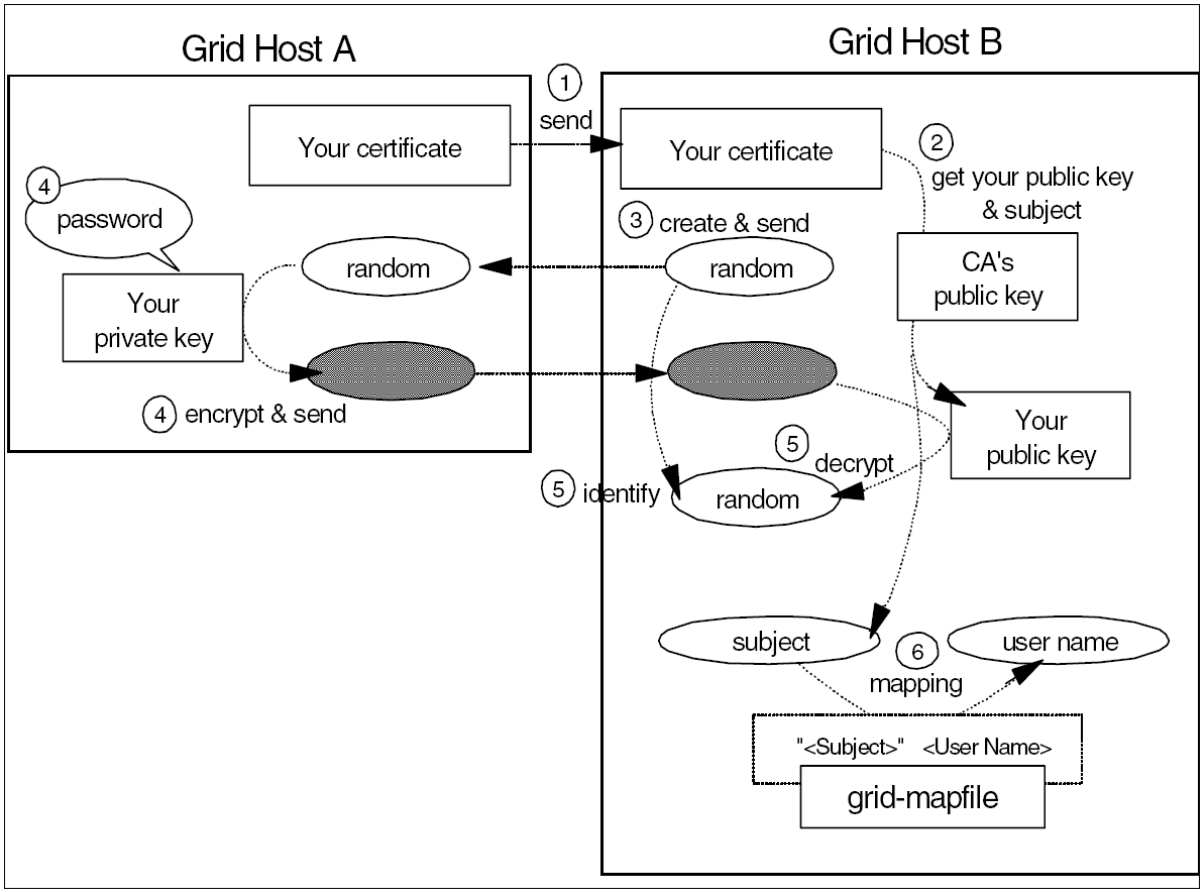
- Based on asymmetric cryptography
- Consist of
  - Public key
  - User identity information (user r
    organization, address, …)
  - One or more digital signature si
    well known Certificate Authority



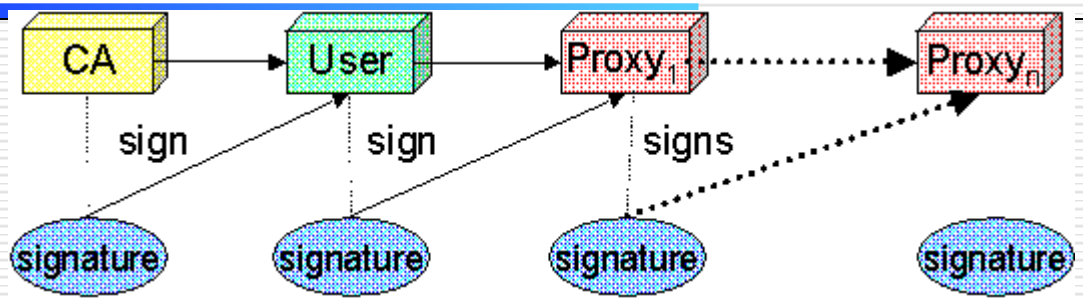
Certificate authority

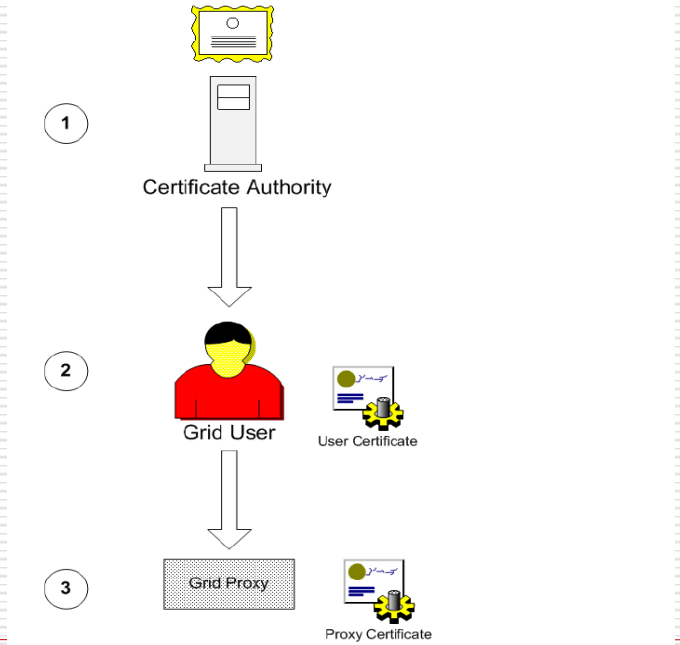
- Responsible for
  - Positively identify entities requesting certificates
  - Issuing, removing, and archiving certificates
  - Protecting the Certificate Authority server
  - Maintaining a namespace of unique names for certificate owners
  - Serve signed certificates to those needing to authenticate entities
  - Logging activity

Grid authentication

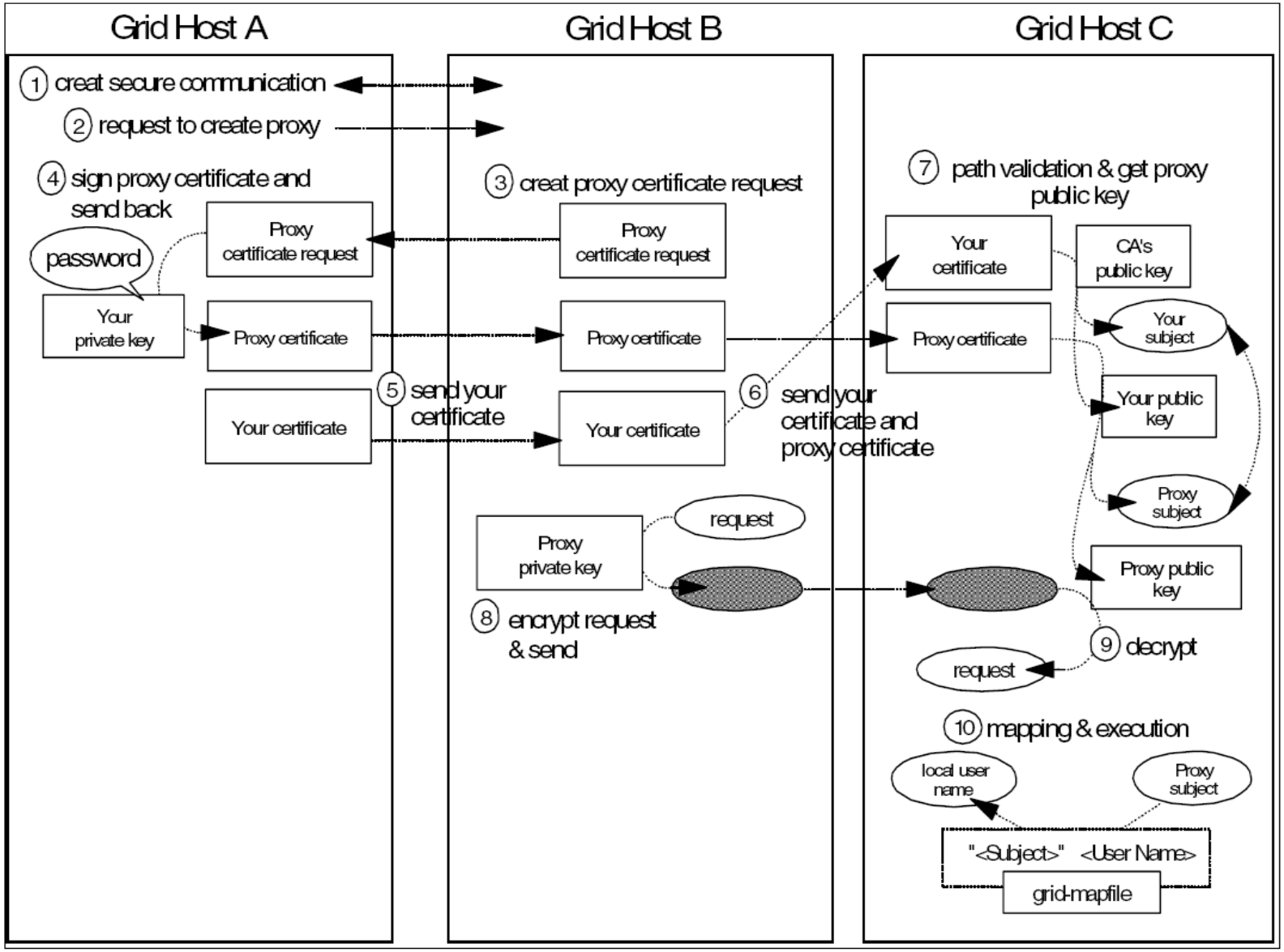


Grid Delegation





Delegation


Grid Host A
Grid Host B
Grid Host C
1 creat secure communication
2 request to create proxy
3 creat proxy certificate request
4 sign proxy certificate and send back
5 send your certificate
6 send your certificate and proxy certificate
7 path validation & get proxy public key
8 encrypt request & send
9 decrypt
10 mapping & execution
password
Your private key
Proxy certificate request
Proxy certificate
Your certificate
Proxy certificate request
Proxy certificate
Your certificate
Proxy private key
request
Your certificate
CA's public key
Proxy certificate
Your subject
Your public key
Proxy subject
Proxy public key
request
local user name
Proxy subject
"<Subject>" <User Name>
grid-mapfile

SOA & Grid

- Service Oriented Architecture
- Web Service
- Grid Service
- Open Grid Service Architecture (OGSA)
- Web Service Resource Framework