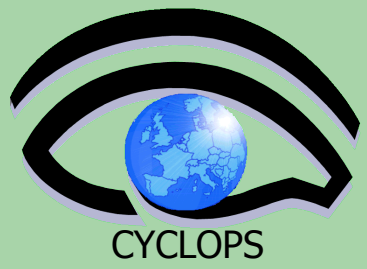
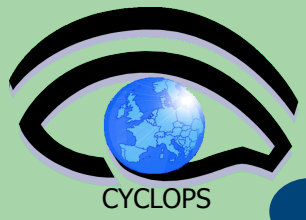
A Secure Storage Service for the gLite Middleware

Antonio Calanducci
on behalf of
Diego Scardaci – INFN Catania

CYCLOPS Second Training Workshop
Chania (Crete), 05th-07th May 2008



Thanks to Giordano Scuderi
UNICO S.R.L.



Outline

- Data Encryption and Secure Storage
  - Insider Abuse: Problem and Solution
- The Secure Storage Service for the gLite Middleware:
  - Command Line Applications
  - Application Programming Interface (API)
  - The Keystore

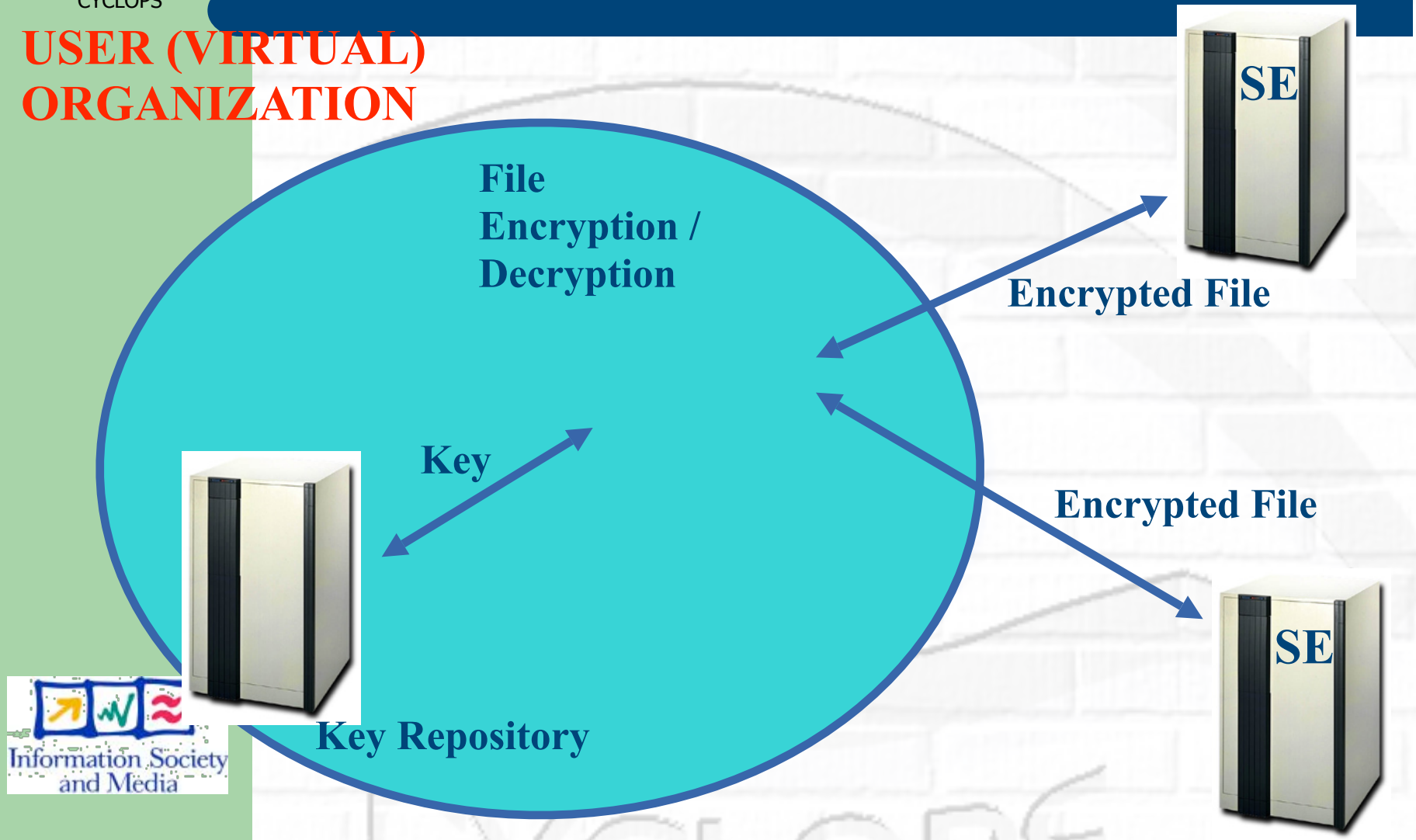
Data Encryption and Secure Storage

- The Secure Storage project is carried out by **UNICO S.R.L.** (http://www.unicosrl.it/) in collaboration with **INFN Catania** in the context of the TriGrid VL Project (http://www.trigrid.it).
- The objective of the project is to create a mechanism to **store in a secure way and in an encrypted format** data on the grid storage elements.
- Thanks to this solution we want to solve the **insider abuse** problem.

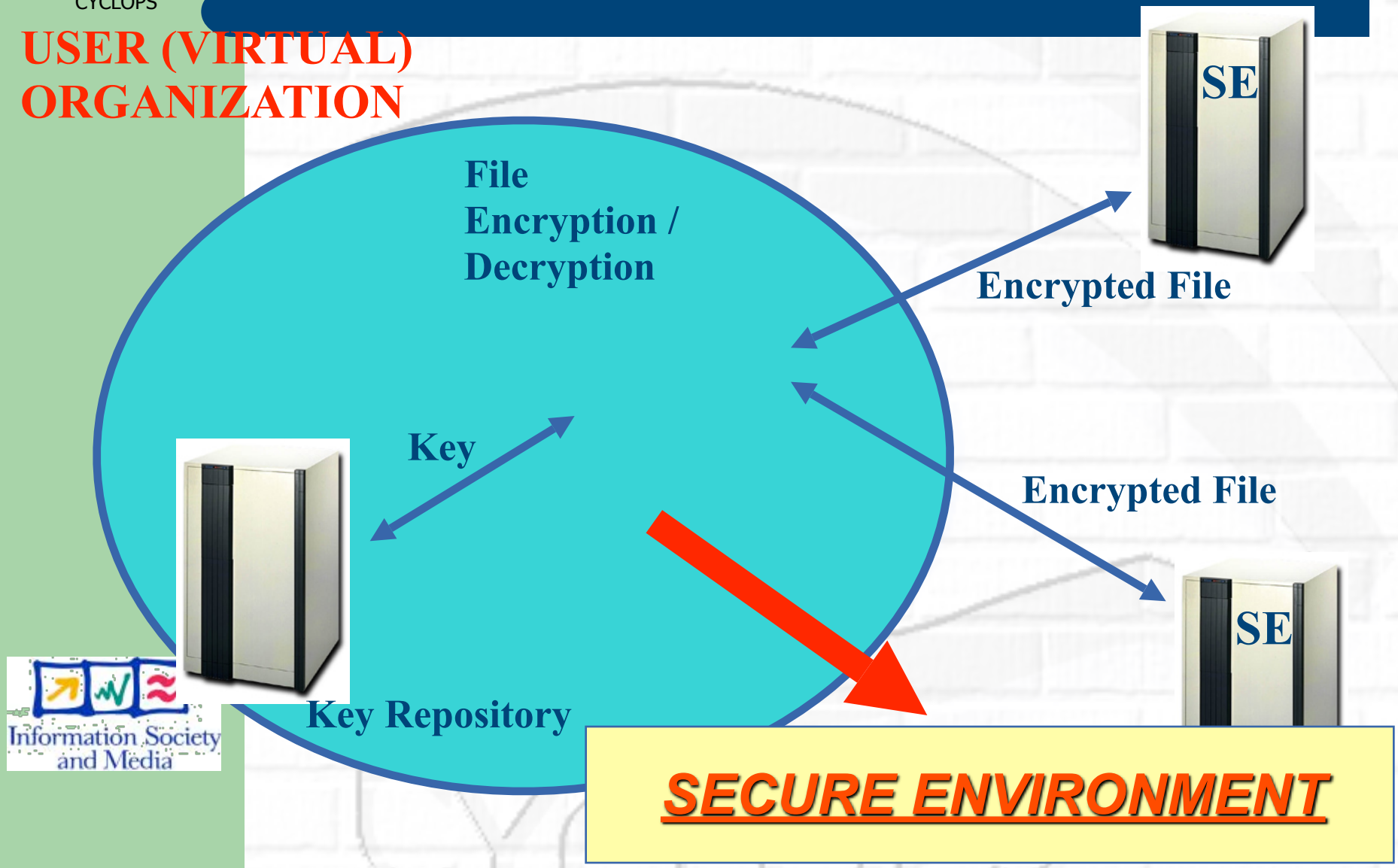
Insider Abuse: Problem

- A grid user could store **sensitive data** in a Storage Elements managed by external organizations.
- Storage Elements Administrators could access data (but the **data are sensitive!**). For this reason data MUST be stored in an encrypted format.
- Data Encryption/Decryption MUST be performed **inside user secure environment** (for example inside the user's organization).

Insider Abuse: A Solution



Insider Abuse: A Solution



A Secure Storage service for the gLite Middleware

- Provides **gLite users** with suitable and simple tools to **store confidential data** in storage elements in a **transparent** and **secure** way.

The service is composed by the following components:

- **Command Line Applications**: commands integrated in the gLite User Interface to encrypt/upload and decrypt/ download files.
- **Application Program Interface**: allows the developer to write programs able to manage confidential data .
- **Keystore**: a new grid element used to store and retrieve the users' keys.

Command Line Applications

- We provide a new set of commands on the gLite User Interface:
  - Like lcg-utils commands, but they work on encrypted data.
  - Encryption and decryption process are **transparent** to the user.
  - Example (copy a local file on a GRID Storage Element):

```
lcg-cr -d <destination SE> -l <lfn>  <source>
lcg-scr -d <destination SE> -l <lfn>
<source>
```

- These commands allow to make the main Data Management operations:
  - Copy data/file on Storage Elements
  - Read data/file from Storage Elements
  - Delete data/file on Storage Elements
  - ….

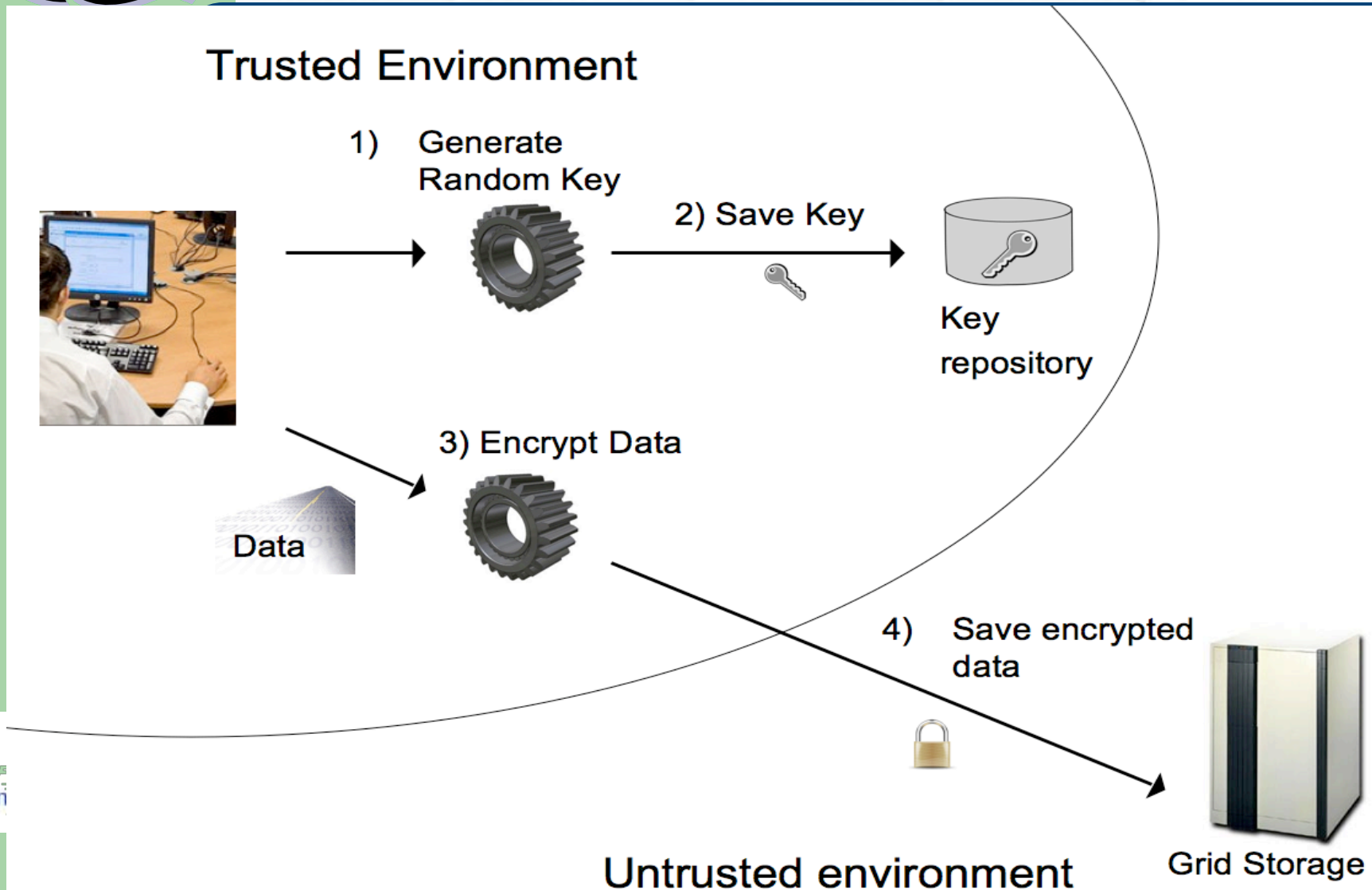
Command Line Applications

- Main Commands:
  - ***lcg-scr***
    - encrypts a file and uploads it on a storage element, registering its Logical File Name in a LFC catalog. Moreover, it stores the key used to encrypt the file in a key repository. An ACL will be associated to each key on the repository. This ACL will contain all users authorized to access the file.
  - ***lcg-scp***
    - downloads an encrypted file, gets the key to decrypt the file from the repository, decrypts the file and store it on a local file-system. Only authorized users (inserted into an ACL) can access the key necessary to decrypt the file.
  - ***lcg-sdel***
    - deletes one or all the replicas of a file. It also deletes the key associated to this file (only if you delete the last replica!)

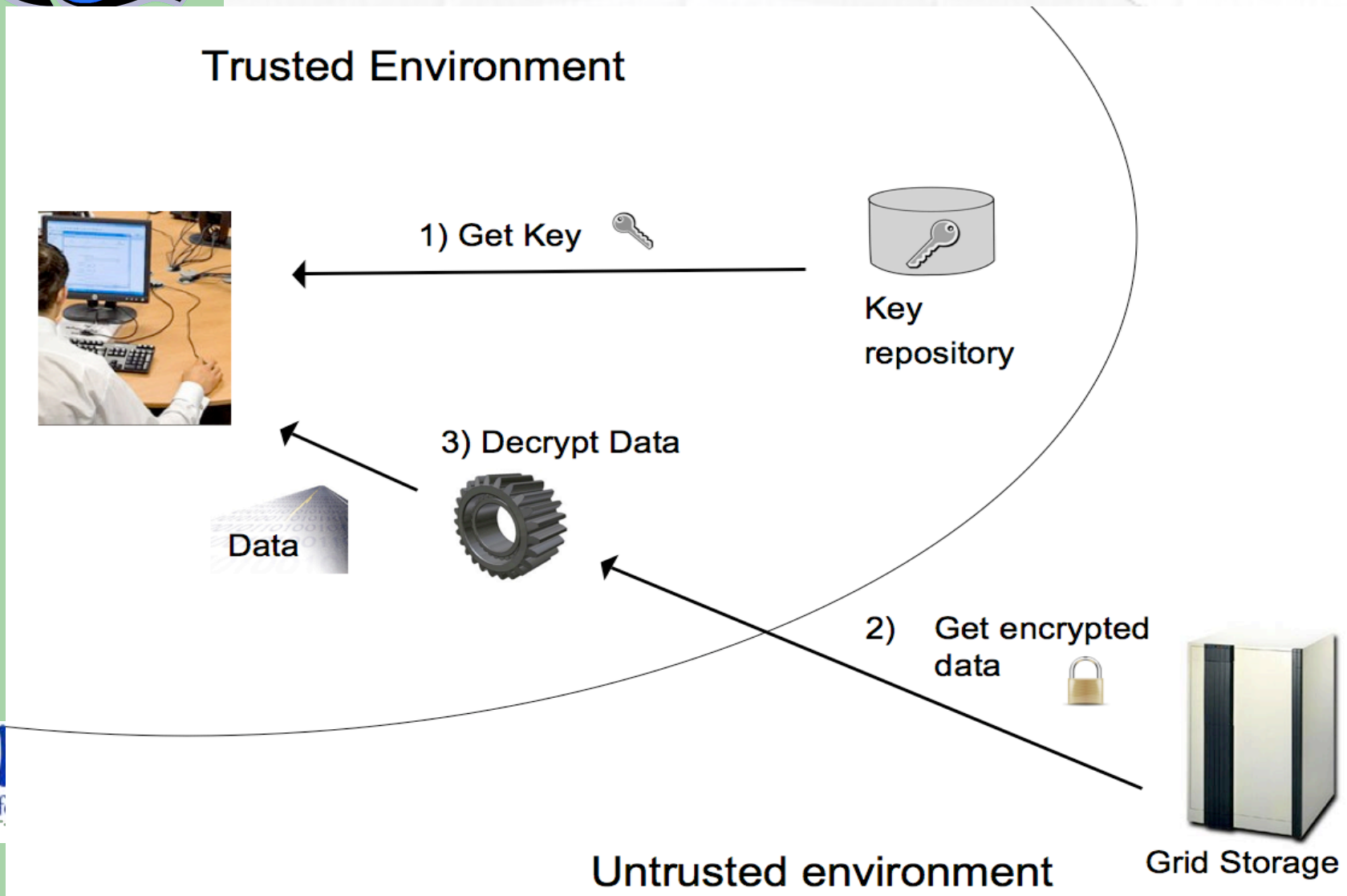
Command Line Applications details

- Details about the main commands:
  - lcg-scr
  - lcg-scp

lcg-scr: Encryption and Storage



lcg-scp: Retrieval and Decryption



Secure Storage API

- Secure Storage C API (like lcg API encrypt and decrypt entire file):
  - int lcg_scr (char *src_file, char *dest_file, char *guid, char* lfn, char *vo, char *relative_path, char *conf_file, int insecure, int verbose, char *actual_gid);
  - int lcg_scp (char *src_file, char *dest_file, char *vo, char *conf_file, int insecure, int verbose);
  - int lcg_sdel (char *src_file, int aflag, char *se, char *vo, char *conf_file, int insecure, int verbose, int

Secure Storage API 2

- Development of API like GFAL (encrypt and decrypt block of data):
  - **int securestorage_open(char *lfn, int flags, mode_t mode);**
  - **int securestorage_write (int fd, void *buffer, size_t size);**
  - **int securestorage_read (int fd, void *buffer, size_t size);**
  - **int securestorage_close (int fd);**
- Read and Write encrypted data like plain data!
  - open
  - read/write
  - close

The Keystore (1)

- The Keystore is a new grid element used to store and retrieve the users' key in a secure way.

The Keystore:

- is identified by an host X.509 digital certificate;
- all its Grid transactions are mutually authenticated and encrypted as required by the GSI model;
- should be placed in a trusted domain and should be appropriately protected by undesired connections;
- is a black box with a single interface towards the external world. This interface accepts only GSI authenticated connections;

The Keystore (2)

The Keystore:

- the client request is processed only if the client is a member of a enabled users list and/or it belongs to an enabled Virtual Organization;

- if the client want to retrieve a key, the keystore checks if the request is coming from an authorized user inserted on the ACL associated to the request key.

CYCLOPS

Any questions ?

Information Society and Media