How to improve the gLite security features

Eng. Francesco Tusa
ftusa@unime.it
http://mdslab.unime.it

Università degli Studi di Messina, Facoltà di Ingegneria
Contrada di Dio, S. Agata, 98166 Messina, Italy

LAGRID 2008 – 29 October 2008, Campo Grande (Brazil)

Outline

- The scope of the work is to integrate new **security solutions** in the gLite grid middleware
- To store users' credentials on a smart card in order to perform:
  - Safe **resources access**
  - Secure **data storing**
  - Secure **data transmission**
- RSA algorithm and XML wrapper employment

gLite: Grid Security Infrastructure

GSI (Grid Security Infrastructure) of gLite is based on:

- **Public key** encryption
- **X.509** certificates
- **SSL** (Secure Sockets Layer) communication protocol

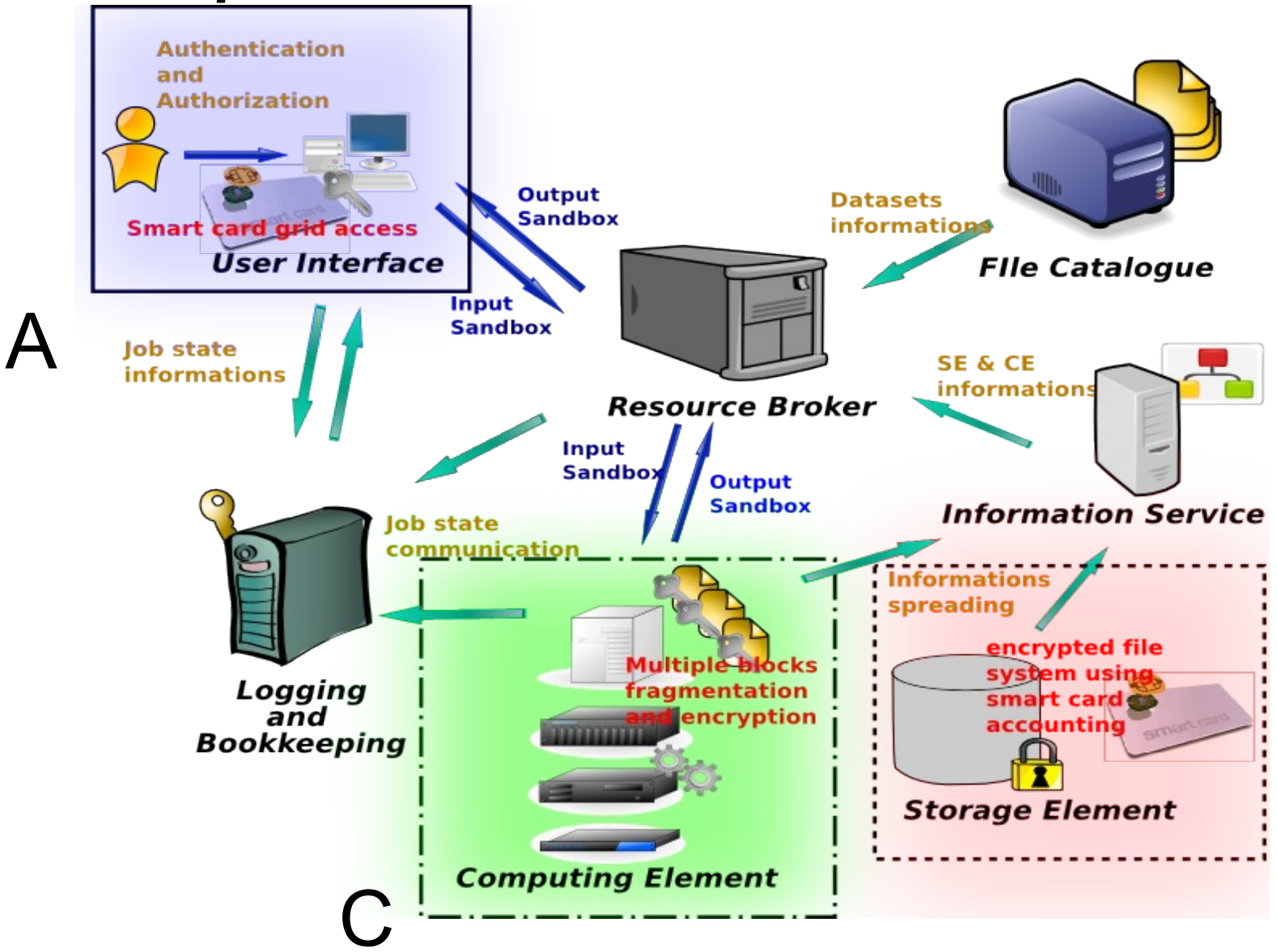
To authenticate himself to the Grid, a user needs to have a trusted X.509 certificate, issued by a Certification Authority (CA)

gLite: Security deficiencies

Pointing out the current GSI infrastructure, two different **threat types** have been identified:

- **Internal Weakness** such as malicious Grid users or malicious Grid system administrators.
- **External Weakness** refers to the different types of security attacks represented by external malicious threats (such as communication sniffing, exploit, denial of service and man in the middle attacks).

Proposed Solution: Overview



Proposed Solution: Overview

Our proposed solution **increases the complexity** of the system but aims to **improve Grid security**, a mandatory requirement in a **business context**. In brief, our proposed solution is able to solve the issues related to:

- **Access to Grid** for users and servers with smart card or crypto-token
- **Storage of user files** on Storage Elements with encryption and a XML wrapper
- **Distribution of user data** toward Computing Elements (CE), along with N different chunks, encrypted with AES/RSA algorithms

Proposed solution: User Interface (A)

- In the traditional grid infrastructure, a user has to employ his own **RSA private key**, stored **inside his UI home directory**, in order to execute the cryptographic operation needed to access the Grid file system
- Our architectural improvement exploits **smart cards** for executing the asymmetric cryptographic operations related to the **management of the credentials**

Proposed solution: Storage Element (B)

- For the time being, in the default gLite middleware implementation, data is stored in the Grid in an **insecure way** (exeception of Hydra module)
- The solution allows data storing on the Grid file system by means of a **security module** that encrypts files, combining together **asymmetric RSA** and **symmetric AES** cryptographic algorithms and **XML data encapsulation**
- An **XML wrapper** guarantees a strong level of **independence** of the system, since it shows a high degree of **flexibility** and a **low** degree of **invasivity**.

Proposed Solution: CE/WN Computing (C)

- New mechanism to **transfer data** among various Grid middleware components
- **Partitioning input data** of the job that has to be executed into **disjointed subsets**.
- Each subset will be transmitted over a **separated secure channel** minimizing the previously discussed external weakness of the current security infrastructure.
- Each **chunk** rappresents a **sub-part** of the **whole file**. While the **user** is **aware** of the **subdivision schema**, the **attacker is not**
- The **smart card** has to be employed in order to **decrypt the AES** key releted to each chunk

Security solution implementation

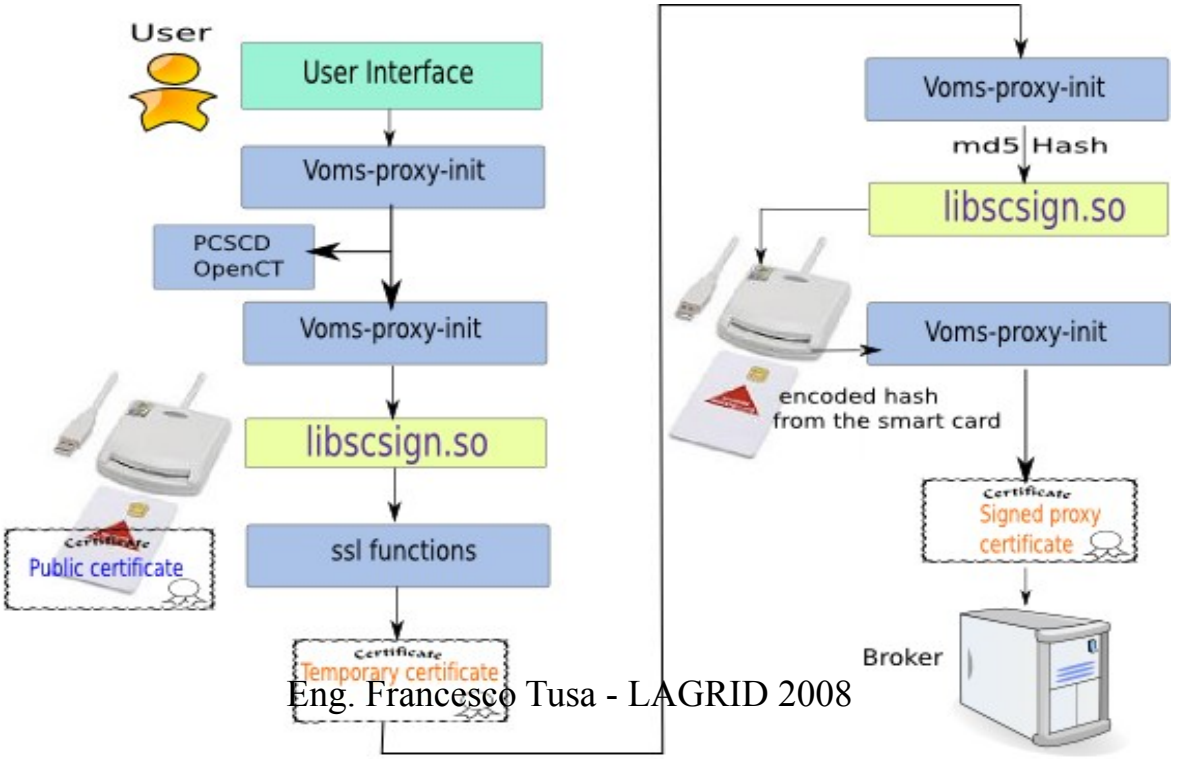
- **Accessing** the grid through smart card in order to grant safe access to resources
- **SE file system encryption** in order to grant data confidentiality
- **Secure data transfer for computation** among the Computing Elements/Worker Nodes

Accessing the grid through Smart Cards

- To grant access to the Grid a **proxy certificate** is generated using the certificate of the user who wants to **access** the Grid infrastructure
- This **certificate** is **signed** using the **RSA user's private key**.
- User's certificate and the associated private key are **stored** in the **User Interface**: it is relatively **easy to stole** the **private key** of a user, and to **spoof** his **identity**
- The **private key,** used for granting the identity of the user, is **stored on a secure device** such as a **smart card**, where it is also **generated**

Accessing the grid through Smart Cards

- A **temporary certificate** is created and a **digest** is performed.
- The **digest is encoded** with the RSA algorithm, using the **user's private key**.
- The **private key** is **stored on a smart card**, since it is not possible to extract it from there.



File system encryption

- Idea based on the employment of **asymmetric RSA** and **symmetric AES** cryptographic algorithms, in order to increase **data integrity** and **confidentiality** on the file storage system
- Symmetric **AES key** (plain format 256 bits) is stored on the **SE together with** the **encrypted data**
- Secret symmetric AES key is **encoded** using the **owner's RSA public key**
- A user who can access the AES key, can also access the corresponding encrypted file. In order **to retrieve** the **AES key**, a file owner has to **employ the own private key** which is stored on the UI (on a crypto token device such as **smart card**)

XML Wrapper

A further idea consists on the **XML wrapper** for storing **data file** and the **related metadata**.

- Data **chunks** of a file are stored inside an **XML node** through a **BASE-64** encapsulation mechanism. In the current implementation, information sets of the XML wrapper are (in addition to data chunk itself (D)):
  - Set of **file owners** (X.509 public certificate) (E)
  - Set of **encrypted AES keys** (encoded using different user RSA public keys) related to the binary file (F)
  - **Hidden XML node** for data recovery (useful for smart card credential loss)
  - Other **various file information** (Data creation, File size, File protection level, etc)

XML Wrapper

```
<keys>
<EncryptedKey Id='01' xmlns='http://www.w3.org/2001/04/xmlenc#'>
          <EncryptionMethod
           Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
          <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>
            <ds:KeyName>Francesco Tusa</ds:KeyName>
          </ds:KeyInfo>
          <CipherData><CipherValue>3gLnAMiDDHZmm</CipherValue></CipherData>
          <ReferenceList>
            <DataReference fid='01'/>
          </ReferenceList>
        </EncryptedKey>
<EncryptedKey Id='02' xmlns='http://www.w3.org/2001/04/xmlenc#'>
          <EncryptionMethod
           Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
          <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>
            <ds:KeyName>Gianluca Triolo</ds:KeyName>
          </ds:KeyInfo>
          <CipherData><CipherValue>DAQABo4IBNTgta</CipherValue></CipherData>
          <ReferenceList>
            <DataReference fid='01'/>
          </ReferenceList>
        </EncryptedKey>
</keys>
```

F

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<file>
    <owners>
        <user id='01'>
            <name>Francesco Tusa</name>
                <public_certificate>
                   MIIESjCCAzKgAwIBAgICH1Mw [...]
                </public_certificate>
        </user>
        <user id='02'>
            <name>Gianluca Triolo</name>
                <public_certificate>
                  a8mEjMiZE/JigHN5RtaKtjt6 [...]
                </public_certificate>
        </user>
    </owners>
```

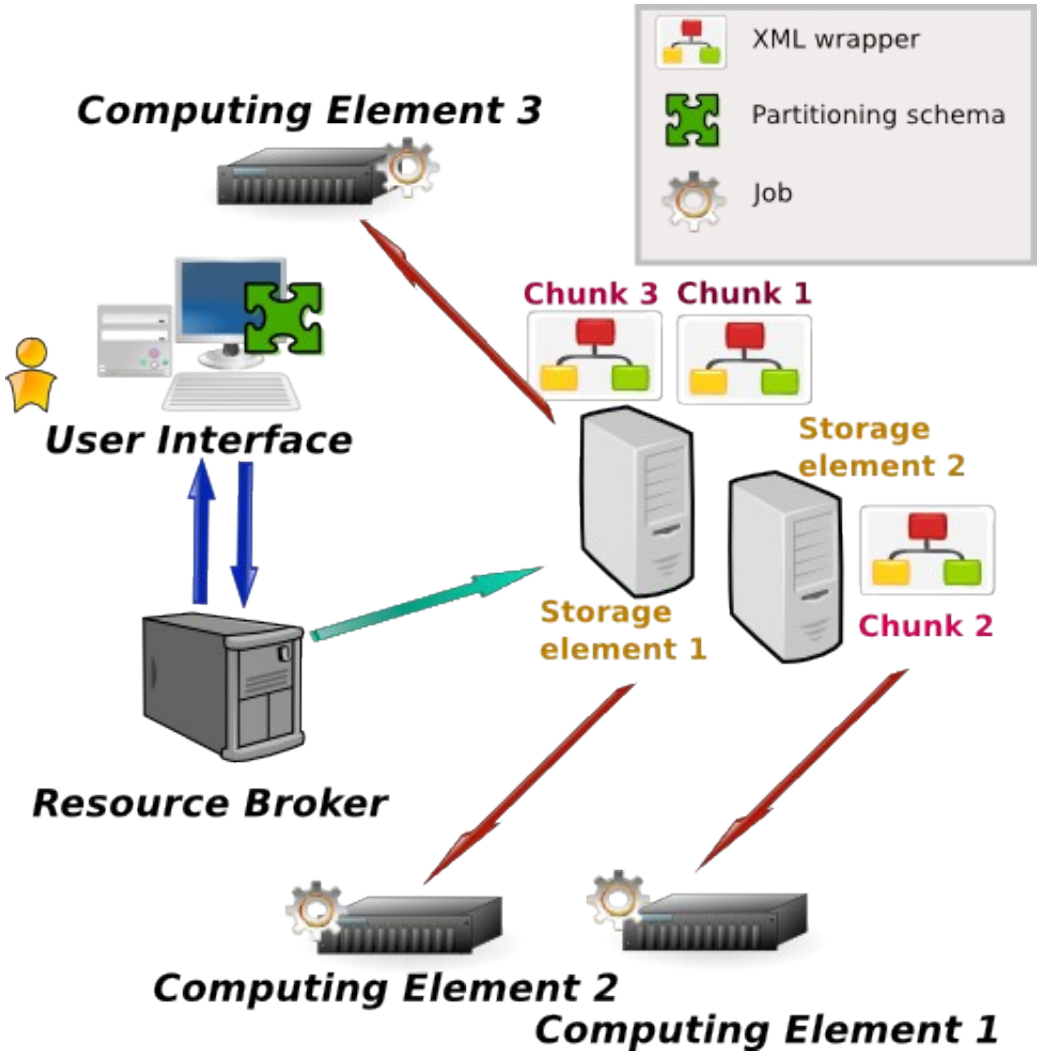
E

```
<content>
   <name>sim_result.dat</name>
   <date>01/01/1970</date>
   <size_b>2132</size_b>

  <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
                 MimeType='binary/base64' fid='01'>
    <CipherData>
      <CipherValue>
        a8mEjMiZE/JigHN5RI8K6taKtjL/jE
        ZOGbvlKsROxzJPM6bO4GJdYO+qhK9E
        5HsbV699DYyukBfUB6ChtD6G [...]
      </CipherValue>
    </CipherData>
  </EncryptedData>
</content>
```

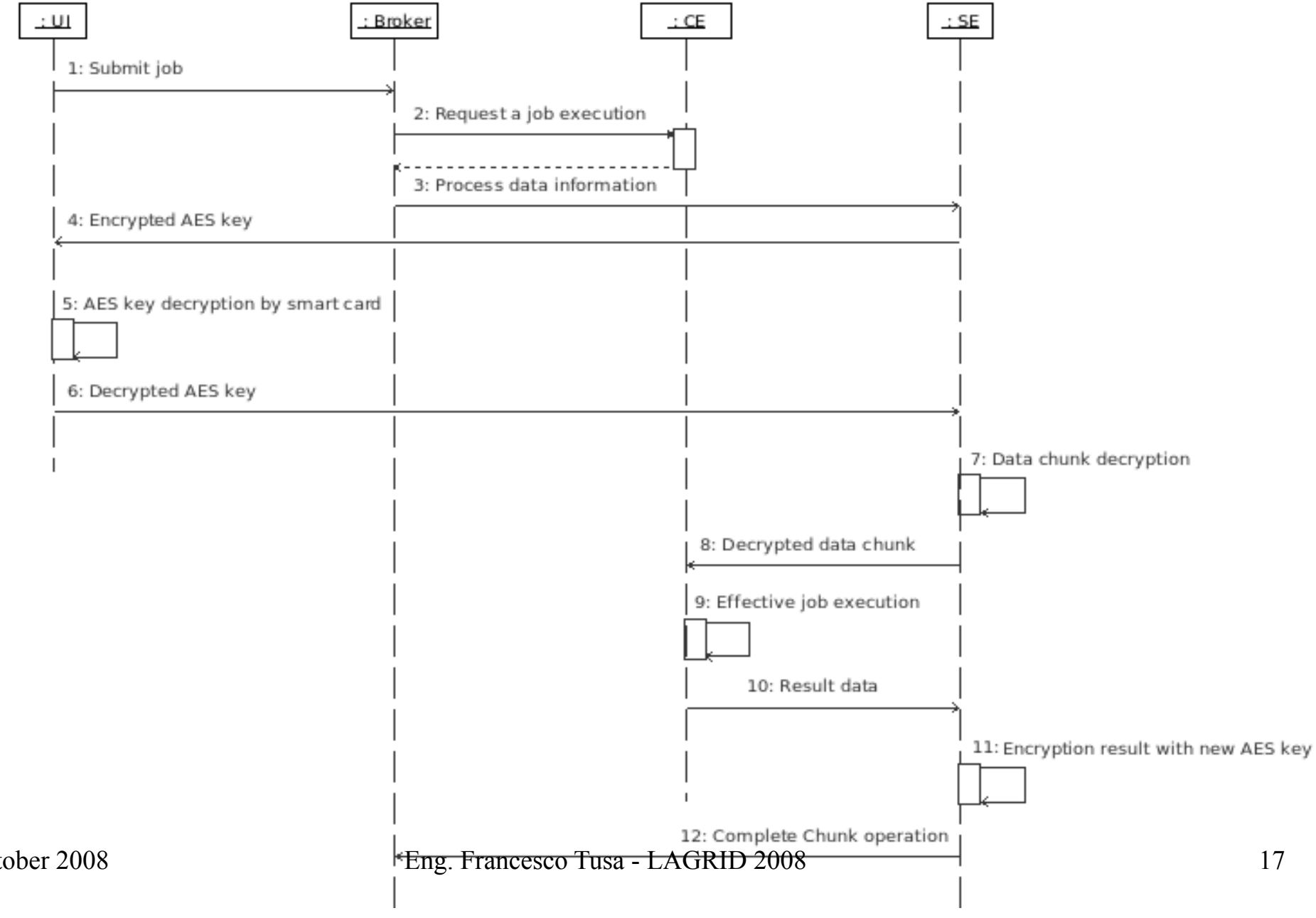
D

Secure data transfer for computation



- A **preliminary phase** (environment configuration) **creates data chunks** on the different available SEs.
- The **effective job execution** includes the **traditional** job **execution** mechanism, **encryption/decryption** of data and **secure data transmission**

Sequence of operations



Conclusions and future works

- A gLite **security architectural improvement** has been proposed
- The security system introduced involves **many parts** of the **infrastructure**, as well as **resources access**, secure **data storing** and **transmission**
- The target is to give an answer to the **requests** pointed out by **business companies** that wish to use grid for **commercial applications**

THANK YOU

Eng. Francesco Tusa

ftusa@unime.it