



Enabling Grids for E-science

FTS Server Installation and Configuration

Gianni M. Ricciardi
INFN – Catania

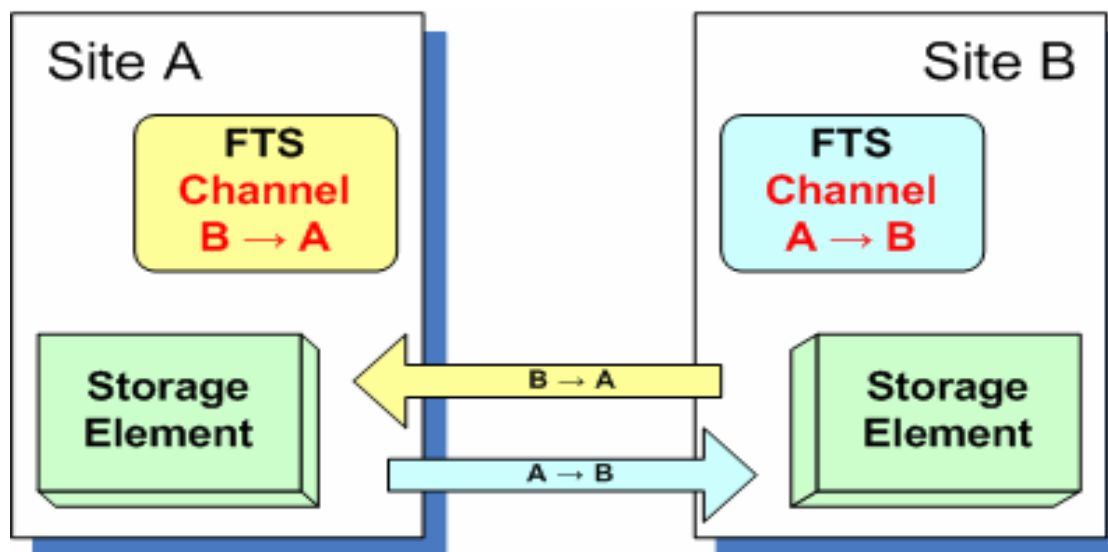
EMBRACE-EGEE Tutorial
Clermont-Ferrand, 9 October – 13 October 2006

www.eu-egee.org

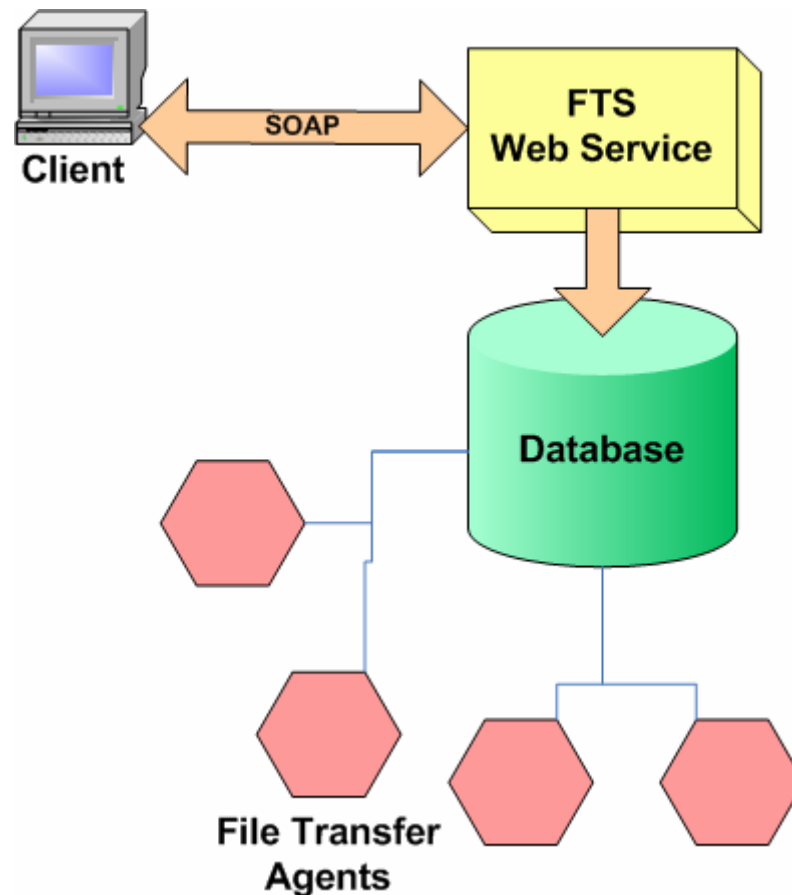


- **Introduction to File Transfer Service (FTS)**
- **How to install and configure a FTS Server**
- **Testing and troubleshooting**

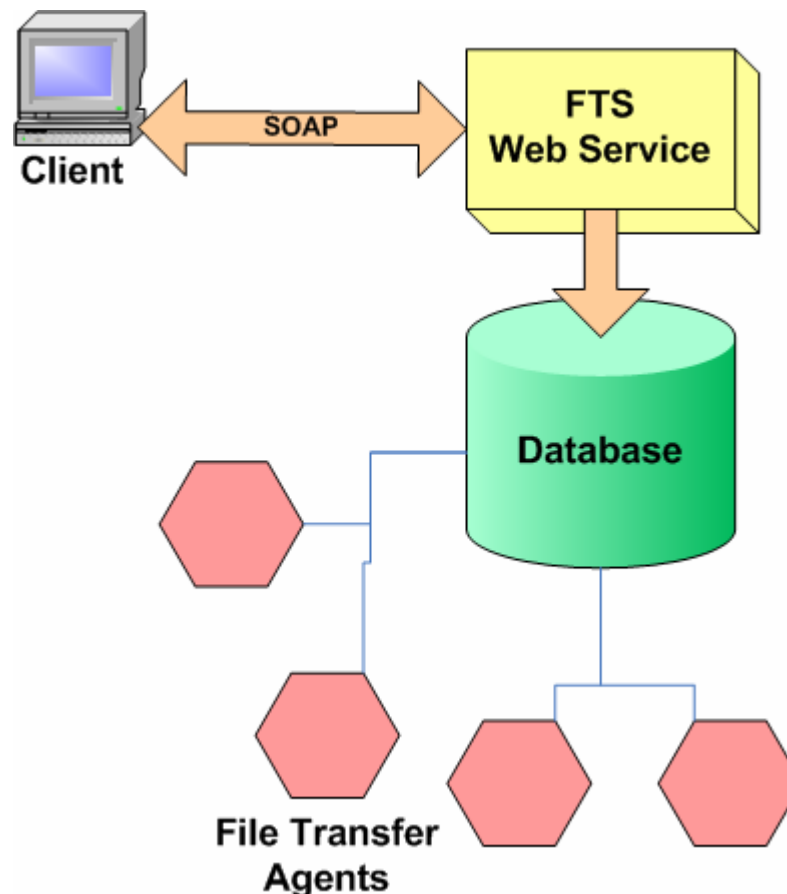
- Moves data from SE (SRM) to SE (SRM).
- The FTS performs file transfers on *channels*.
- An FTS *instance* serves a configurable set of channels.
- Every channel is unidirectional, i.e. it is intentional that different FTS instances may serve the two directions of a network link between two sites.



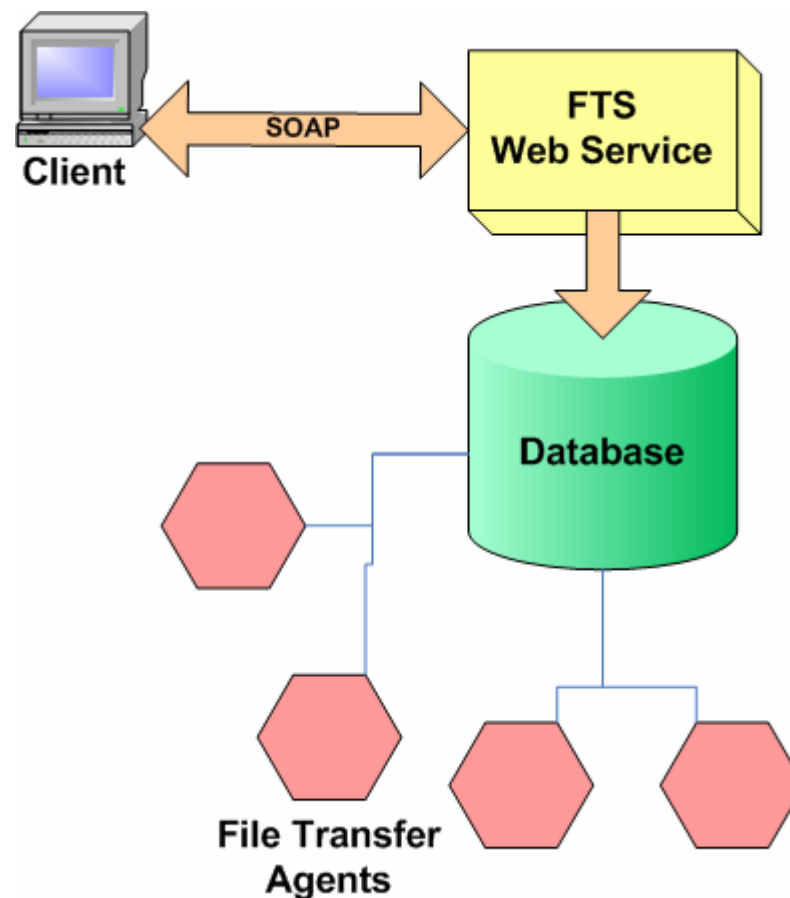
- The FTS client libraries or command line tools are used by the applications to communicate with the FTS.



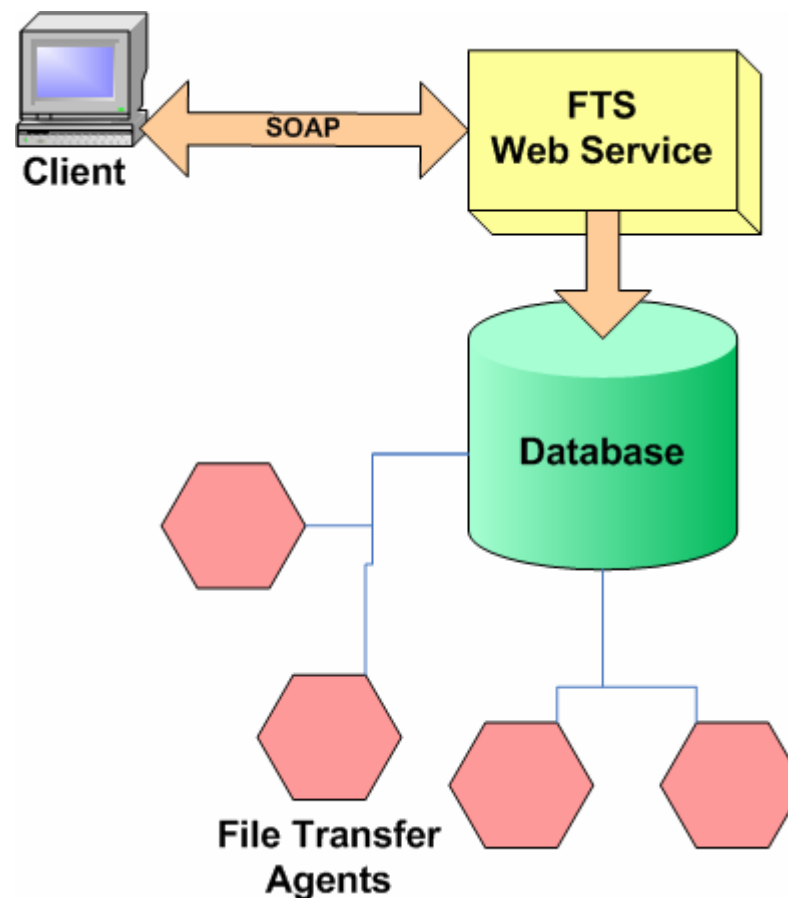
- The web service component is implemented as a Tomcat web application.
- There can be many Tomcat containers deployed that access the same underlying database, which is one of the possibilities to scale the system.
- The web service connects to the database through JDBC using a Tomcat database connection pool.



- There is a MySQL and an Oracle implementation of the FTS schema. This is the only persistency point in the system and it scales only as well as the corresponding backend allows. There can be only a single instance of this database for a given FTS instance.



- The agents will act based on certain conditions in the database. They make sure that the finite state machine of the FTS is adhered to and perform all state transitions for the transfer jobs and the files being transferred. It is an agent that actually performs the file transfers on the channels that the FTS manages.



FTS Server Installation

- Start with a machine with Scientific Linux CERN 3.0.X installed.
- Verify that these packages are installed and properly configured:
 - Java SDK
 - NTP daemon
 - ca_GILDA (<https://gilda.ct.infn.it/RPMS/>)
 - edg-voms-vo-gilda (<https://gilda.ct.infn.it/RPMS/>)
 - glite-yaim-3.0.0-* (<http://www.cern.ch/grid-deployment/gis/yaim/>)
 - gilda_ig-yaim-3.0.0-* (https://grid018.ct.infn.it/apt/gilda_app-i386/utils)
- Also check that your host certificates are present in */etc/grid-security* and have proper permissions:

```

-rw-r--r--    1 root root  1127    Jun 14 12:27 hostcert.pem
-r-----    1 root root   887    Jun 14 12:28 hostkey.pem

```

- Control that this role is present in `/etc/sysconfig/iptables`:
- A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 8443 -j ACCEPT
- If it is no present, insert the above red row and then restart the service iptables with the commands:

`>/etc/init.d/iptables start`

and verify its status

`>/etc/init.d/iptables status`

- **Download the RPM of Oracle Express by the Oracle site:**
 - <http://www.oracle.com/technology/products/database/xe/index.html>
- **Download RPMs of Oracle Instant Client by:**
 - <http://www.oracle.com/technology/software/tech/oci/instantclient/index.html>
- **Install the package *libaio* and *libxslt***
 - *# apt-get install libaio*
 - *# apt-get install libxslt*
 - *# export*
LD_LIBRARY_PATH=\${LD_LIBRARY_PATH}:/usr/lib/oracle/10.1.0.4/client/lib

- **Install the four RPMs:**
 - # rpm -ivh oracle-instantclient-basic-10.1.0.4-1.i386.rpm
 - # rpm -ivh oracle-instantclient-jdbc-10.1.0.4-1.i386.rpm
 - # rpm -ivh oracle-instantclient-sqlplus-10.1.0.4-1.i386.rpm
 - # rpm -ivh oracle-xe-univ-10.2.0.1-1.0.i386.rpm
- **Source *oracle_env.sh* :**
 - # *./usr/lib/oracle/xe/app/oracle/product/10.2.0/server/bin/oracle_env.sh*
- **Configure Oracle Express:**
 - # */etc/init.d/oracle-xe configure (answer with default values and pay attention to SYSTEM ORACLE password)*

- **Create file *oracle.sql*:**

```
create user fts_user identified by <YOUR PASSWORD>;
```

```
grant resource to fts_user;
```

```
grant create session to fts_user;
```

```
grant create synonym to fts_user;
```

```
grant connect to fts_user;
```

```
grant create any procedure to fts_user;
```

```
grant create any sequence to fts_user;
```

```
grant create trigger to fts_user;
```

```
grant create type to fts_user;
```

```
grant debug any procedure to fts_user;
```

```
grant debug connect session to fts_user
```

- **Run:**

```
#sqlplus
```

```
system/YOUR_PASSWORD@"(DESCRIPTION=(LOAD_BALANCE=no)(ADDRESS=(PROTOCOL=TCP)(HOST=FTS_HOST)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=xe)))" @oracle.sql
```

- **Copy /opt/glite/yaim/examples/gilda_ig-site-info.def into /root/my-site-info.def and edit these fields:**
 - MY_DOMAIN=<your_domani>
 - FTS_HOST=<your_FTS_serve>.\$MY_DOMAIN
 - JAVA_LOCATION="/usr/java/j2sdk1.4.2_12"
 - ORACLE_LOCATION=/usr/lib/oracle/10.1.0.4 (create it if not already there)
- **Identify what** agents go on which machines
 - the hostname of the machines you want to be running the FTA agents;

- **Add the following variables:**
 - FTA_MACHINES="ONE"
 - FTA_AGENTS_ONE_HOSTNAME="FTA_agent_hostname"
 - FTA_AGENTS_ONE="GILDA CHANNELAB CHANNELBA "
 - FTA_GILDA="VOAGENT"
 - FTA_CHANNELAB="URLCOPY"
 - FTA_CHANNELBA="URLCOPY"
 - FTA_GLOBAL_DBTYPE=ORACLE
 - FTA_GLOBAL_DB_CONNECTSTRING="(DESCRIPTION=(LOAD_BALANCING=no)(ADDRESS=(PROTOCOL=TCP)(HOST=hostname)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=XE)))"
 - FTA_GLOBAL_DB_USER=fts_user
 - FTA_GLOBAL_DB_PASSWORD= YOUR_FTS_USER_PASSWORD
 - FTA_GLOBAL_LOG_PRIORITY=DEBUG

- FTA_TTYPEDEFAULT_URLCOPY_GUC_LOGLEVEL=DEBUG
- FTA_TTYPEDEFAULT_VOAGENT_ACTIONS_MAXFAILURES=4
- FTS_DBURL="jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=no)(ADDRESS=(PROTOCOL=TCP)(HOST=hostname)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=XE)))"
- FTS_DB_TYPE=ORACLE

- Run the following command to download and install all needed packages:
 - **# /opt/glite/yaim/scripts/install_node site-info.def GILDA_ig_FTS
GILDA_ig_FTA**
- Load the schema of Oracle DB:
 - **#sqlplus
fts_user/YOUR_PASSWORD" @(DESCRIPTION=(LOAD_BALANCE=no)(
ADDRESS=(PROTOCOL=TCP)(HOST=HOSTNAME)(PORT=1521))(CONN
ECT_DATA=(SERVICE_NAME=XE)))" @/opt/glite/etc/glite-data-transfer-
fts/schema/oracle/oracle-schema.sql**
- Configure the node:
 - **# /opt/glite/yaim/scripts/configure_node site-info.def GILDA_ig_FTS**
 - **# /opt/glite/yaim/scripts/configure_node site-info.def GILDA_ig_FTA**

- **Edit file `/opt/glite/etc/services.xml` :**

```
<services>
```

```
  <service name='EGEEfts'>
```

```
    <parameters>
```

```
      <endpoint>https://FTS_FQDN:8443/glite-data-transfer-fts/services/FileTransfer</endpoint>
```

```
      <type>org.glite.FileTransfer</type>
```

```
      <version>3.0.0</version>
```

```
    </parameters>
```

```
  </service>
```

```
  <service name='EGEEchannel'>
```

```
    <parameters>
```

```
      <endpoint>https://FTS_FQDN:8443/glite-data-transfer-fts/services/ChannelManagement</endpoint>
```

```
      <type>org.glite.ChannelManagement</type>
```

```
      <version>3.0.0</version>
```

```
    </parameters>
```

```
  </service>
```

- ***`/opt/glite/etc/services.xml`***

```

<service name='EGEEstats'>
  <parameters>
    <endpoint>https://FTS_FQDN:8443/glite-data-transfer-fts/services/FileTransferStats</endpoint>
    <type>org.glite.FileTransferStats</type>
    <version>3.0.0</version>
  </parameters>
</service>

```

```

<service name='MyProxyCt'>
  <parameters>
    <endpoint>myproxy://grid001.ct.infn.it</endpoint>
    <type>MyProxy</type>
    <version>1.14</version>
  </parameters>
</service>

```

- ***/opt/glite/etc/services.xml :***

```

<service name='NAME'> <!-- For each SE, insert an entry of this kind -->
  <parameters>
    <endpoint>httpg://SE_FQDN:8443/srm/managerv1</endpoint>
    <type>SRM</type>
    <version>1.1.0</version>
    <site>healthgrid.org</site>
    <param name='SEMountPoint'>/dpm/healthgrid/home/gilda</param>
  </parameters>
</service>
[...]
</services>

```

- **Some Oracle libraries may be not properly installed in */opt/glite/lib* , so run the following commands if they are not present:**
 - *#cd /opt/glite/lib*
 - *#ln -s /usr/lib/oracle/10.1.0.4/client/lib/libclntsh.so.10.1 libclntsh.so.10.1*
 - *#ln -s /usr/lib/oracle/10.1.0.4/client/lib/libnnz10.so libnnz10.so*
 - *#ln -s /usr/lib/oracle/10.1.0.4/client/lib/libociei.so libociei.so*
 - *#ln -s /usr/lib/oracle/10.1.0.4/client/lib/libocci.so.10.1 libocci.so.10.1*

- It is recommended put at least one subject on the admin mapfile:

- */opt/glite/etc/glite-data-transfer-manager-mapfile*

```
"/C=IT/O=GILDA/OU=Host/L=EMBRACE/CN=training244v.healthgrid.org/emailAddress=gilda-ca@ct.infn.it" null
```

```
"/C=IT/O=GILDA/L=CLERMONT/CN=CLERMONT15/Email=gilda-ca@ct.infn.it" root
```

```
"/C=IT/O=GILDA/L=CLERMONT/CN=CLERMONT15/Email=gilda-ca@ct.infn.it" null
```

- To get subject of your certificate:

```
#openssl x509 -in yourcert.pem -noout -text -subject
```

- *And in the submit mapfile:*

- */opt/glite/etc/glite-data-transfer-submit-mapfile*

- **FTS start comes together with tomcat start up:**
 - # service tomcat5 [re]start

- Before starting File Transfer Agents, it is needed to create the channel agent previously declared. The channel creation can be done both from the FTS server or any client, such as the User Interface. In the latter case, be sure that the fts client you are using is pointing correctly the service and that the subject of your certificate is within */opt/glite/etc/glite-data-transfer-manager-mapfile* in fts server; of course you must have a valid proxy.

- Use the *-s <end-point>* option in all the following *glite-transfer-** commands.

- *glite-transfer-channel-add CHANNELAB site1 site2*
- *glite-transfer-channel-add CHANNELBA site2 site1*
- List their properties
- *glite-transfer-channel-list CHANNELAB*

Channel: CHANNELAB

Between: site1 and site2

State: Active

Contact: (null)

Bandwidth: 0

Nominal throughput: 0

Number of files: 0, streams: 0

Number of VO shares: 0

- **To set needed parameters:**
 - `glite-transfer-channel-set -b 100 -f 5 -T 10 -t 100 CHANNELAB`
 - ***-b: available badwidth (in Mb/s)***
 - ***-t: throughput goal to achieve***
 - ***-f: number of concurrent transfers***
 - ***-T: Number of concurrent transfer streams***
 - ***glite-transfer-channel-setvoshare CHANNELAB gilda 100***

This will define a 100% share (de facto an unique VO) for the VO gilda on that channel.

- **If all of the declared channel have been created, each File Transfer Agent should start correctly:**
 - # service transfer-agents [re]start
- **To start a single agent daemon:**
 - # service transfer-agents [re]start --instance glite-transfer-channel-agent-urlcopy-CHANNELAB
- **Now it should be possible use the FTS for file transferring among the declared Storage Elements.**

- Create a MyProxy certificate with
 - **myproxy-init -d**
(-d because FTS need to know Real User)

- Start a transfer
 - **glite-transfer-submit <file1> <file2>**
 - Schedule a transfer from file1 to file2
 - where:
 - file1= srm://SE1_FQDN/srm/managerv1?SFN=/dpm/[...]/file
 - file2=srm://SE2_FQDN/srm/managerv1?SFN=/dpm/[...]/newfile
 - **glite-transfer-status <uuid>**
 - Check the status of a job

- **FTS Server Installation Guide**
 - <https://uimon.cern.ch/twiki/bin/view/LCG/FtsServerInstall15>
- **GILDA gLite3.0 Wiki**
 - <https://grid.ct.infn.it/twiki/bin/view/GILDA/GliteElementsInstallation>

