



Enabling Grids for E-science

File Transfer Service

*Presenter : Markus Schulz, CERN, IT-GD-ITR
On behalf of the CERN FTS team
CHEP07, 2-7 Sept 2007, Victoria BC Canada*

www.eu-egee.org

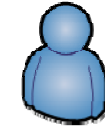


- **FTS Overview**
- **APIs and Security Model**
- **Transfer channels**
- **Server architecture**
- **Deployment on WLCG**
- **Future Plans**

- **The File Transfer Service (FTS) is a data movement service**
 - It is a multi-VO service, used to balance usage of site resources according to VO and site policies
 - The starting assumption is that all these data streams are such high volume that they need to be managed
 - Application level prioritisation (between VOs, within a VO)
 - Prevent network overload
 - Prevent storage overload
 - Monitor and understand problems on the service
 - Even a 0.1% failure is a large amount of data to manually 'rescue' or follow-up when you're transferring petabytes!

- **Why is it needed ?**

- For the user, the service it provides is the reliable point to point movement of files



Experiment user

- For the site / service manager, it provides a reliable and manageable way of serving file movement requests from their experiments



Site Manager

- Control and monitoring to meet the SLA targets

- For the VO production manager, it provides ability to control requests coming from his users



VO Production Manager

- Re-ordering, prioritization,...

- **The focus is on the “service”**

- **Its purpose is to make it easy to do these things well (with limited manpower)**

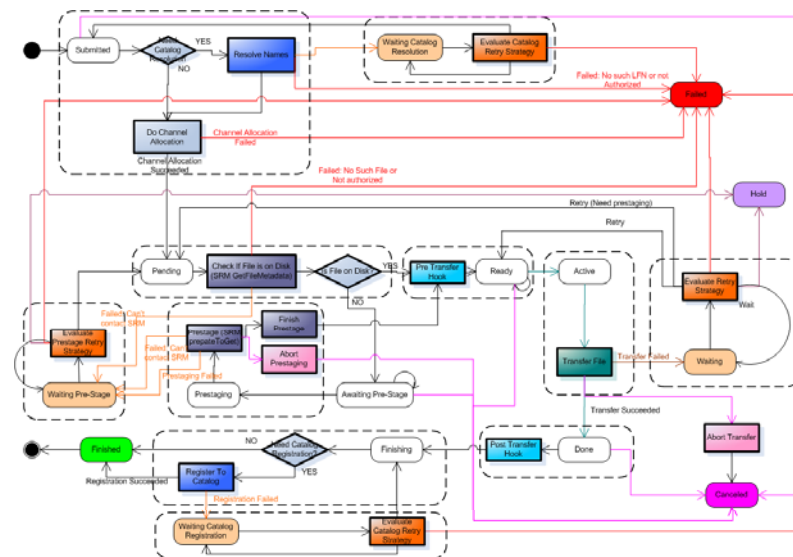
- **FTS is middleware: it's used by experiment software frameworks**
 - Typically end-users do not interact directly with it – they interact with their experiment framework that submits into the FTS
 - Production managers sometimes query it directly to debug / chase problems
- **Experiment framework 'decides' it wants to move a set of files**
 - The expt. framework is responsible for staging-in from tape
 - It packages up a set of source/destination file pairs and **submits** transfer jobs to FTS
 - The state of each job is tracked as it progresses through the various transfer stages
 - The experiment framework can **poll** the status at any time

- **Job submission / tracking API**

- Used by the applications to submit transfer job requests and poll for the status
- The files progress through a well-defined state machine

- **Service / channel management API**

- Used by service administrators and VO production managers to control the service

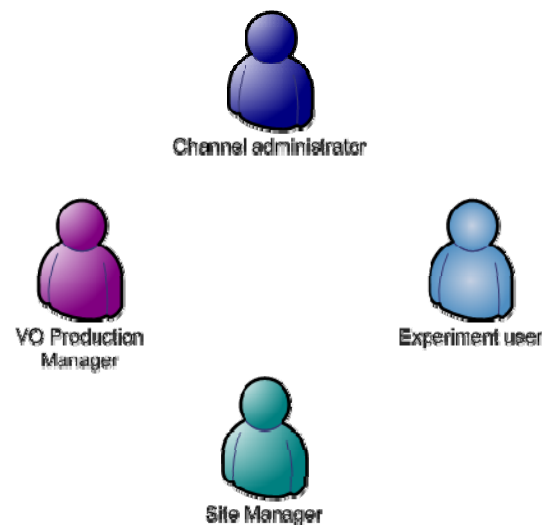


- **Monitoring tools**

- Providing aggregate statistics on what the service has been doing, current failure rates, classes, etc
- Make sure the information is available to all interested stakeholders - everyone wants their view...
 - WLCG, Site and VO management to verify MoU compliance
 - Site admins to understand the current service problems

- **Peer services (FTS / SRM) do not trust each other**
 - Different administrative / legal domains
- **Transfers are always run using the originating clients' X509 credentials**
 - These are delegated by the client to the service (impersonation)
 - Full audit on all operations
 - VOMS credentials (attribute certificates) are now used (and renewed as necessary) in FTS 2.0

- **VO production manager roles**
- **Channel administrator roles**
- **Service manager role**



- For management ease, the service supports splitting jobs onto multiple “channels”

- Once a job is submitted to the FTS it is assigned to a suitable channel for serving

- A channel may be:

- A point to point network link (e.g. we manage all the CERN data-export links on distinct channels)

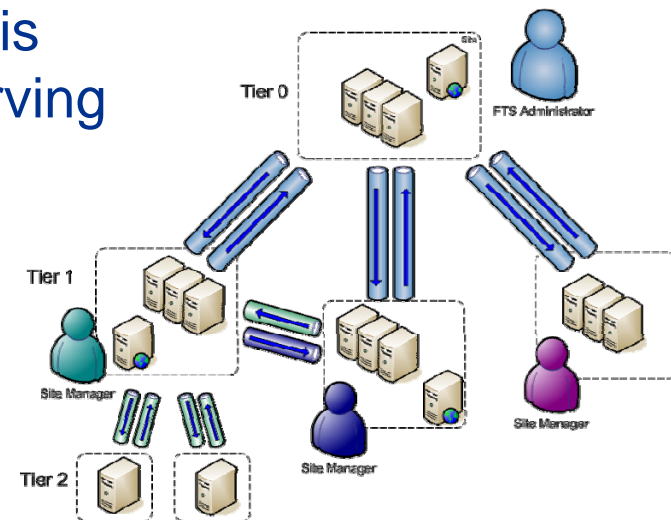
- Various “catch-all” channels

- (e.g. everything else coming to me, or everything to one of my tier-2 sites)

- More flexible “grouping of sites” channel definitions coming...

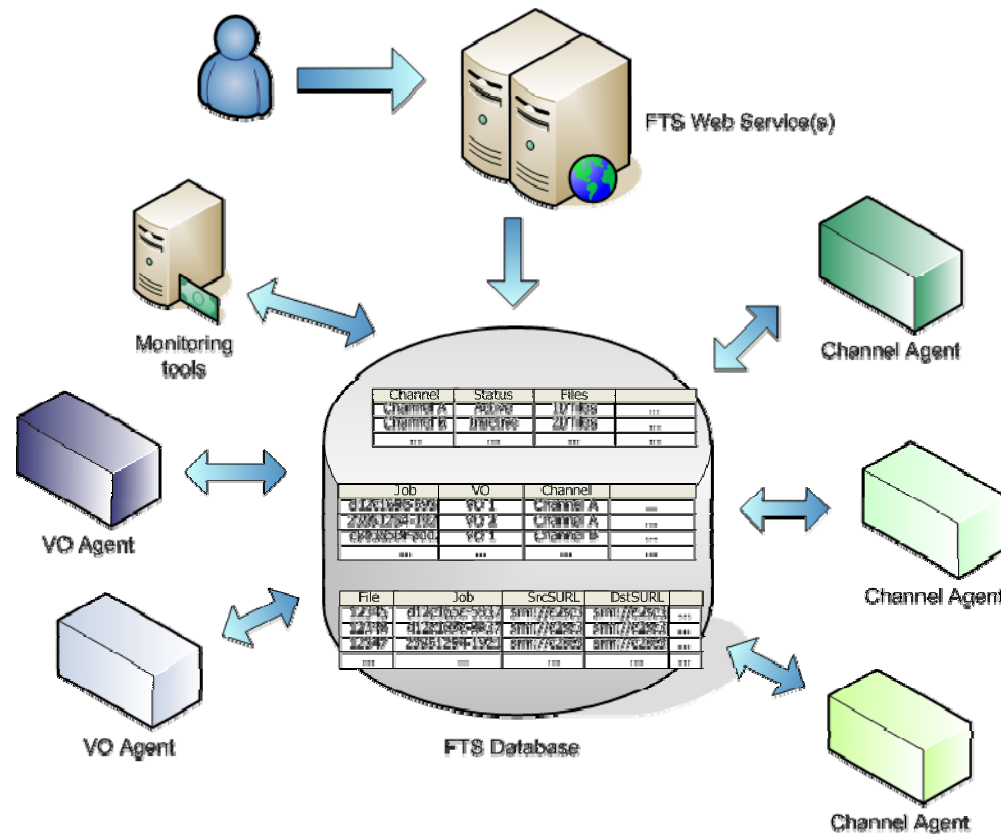
- Channels are uni-directional

- e.g. at CERN we have one set for the data-export and one set for the data-import



- **“Channel”**: it’s not a great name
 - It always causes confusion... (but we’re ~stuck with the name now)
 - It isn’t (necessarily) tied to a physical network path
 - It’s just a management concept
 - “Queue” might be a better name ☺
- **All file transfer jobs on the same channel are served as part of the same queue**
 - Inter-VO priorities for the queue (Atlas gets 75%, CMS gets the rest)
 - Internal-VO priorities within a VO
 - Each channel has its own set of transfer parameters
 - Number of concurrent files running, number streams, TCP buffer, etc
- **Given the transfers (i.e. from where to where) your FTS server is required to support (as defined by your SLA) channels allow you to split up the management of these as you see fit**

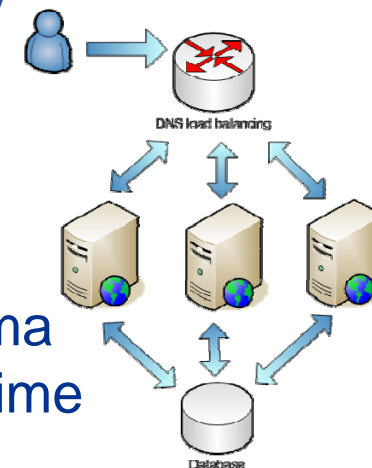
- **FTS provides the reliable and manageable transport layer**
- **It does not (and will not) provide more complex data scheduling!**
 - Multi-hop transfers
 - Broadcast transfers
 - Dataset collation
- **But it may be used as the underlying management layer for services providing this**
- **Much of this extra functionality is currently provided in the experiment layer**
 - It's quite application computing-model dependent



- Experiments interact via web-service
- VO agents do VO-specific operations (1 per VO)
- Channel agents do channel specific operation (e.g. the transfers)
- Monitoring and statistics can be collected via the DB

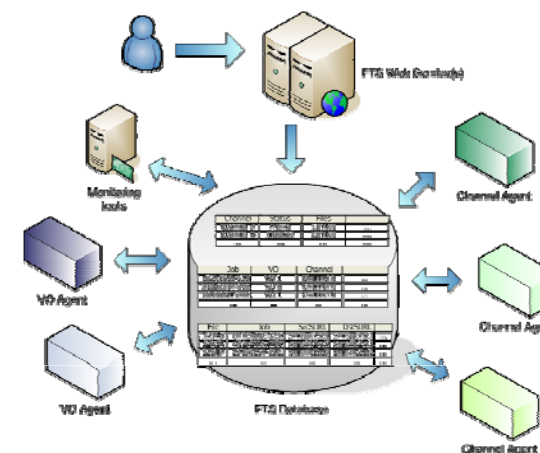
- All components are decoupled from each other
 - Each interacts only with the (Oracle) database

- Designed for high availability and scalability
- User front-end web-service is stateless and (should be) load balanced to provide availability and scalability
 - Service interventions that don't require a DB schema upgrade can be made with zero user-visible downtime

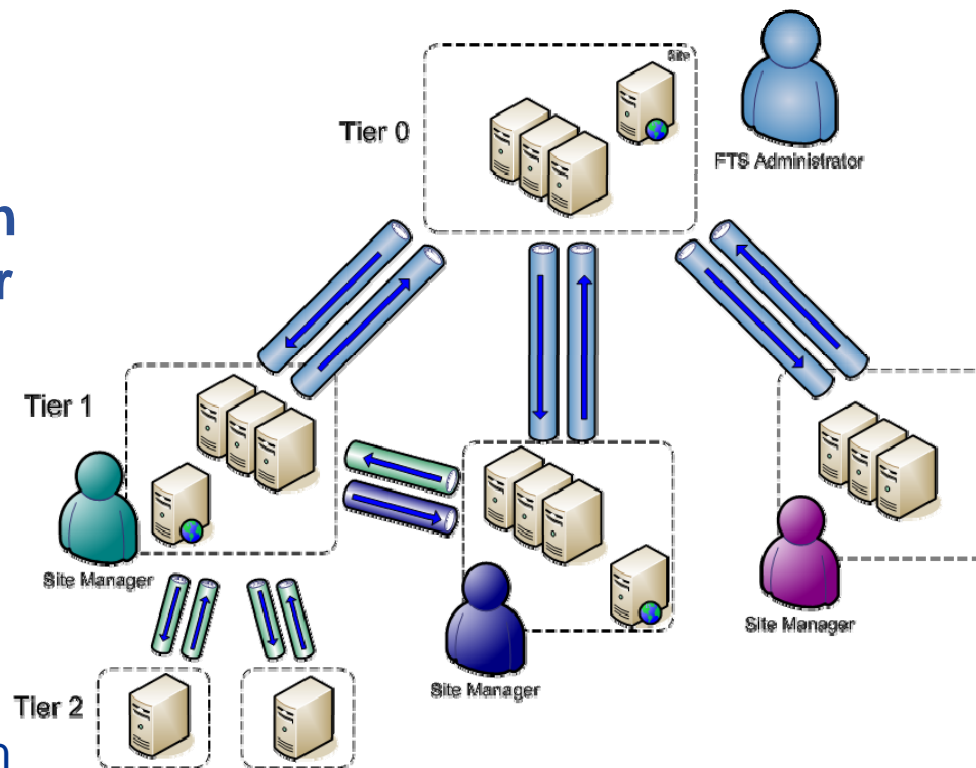


- Agent daemons are designed to scale over multiple nodes as necessary with load

- Critical component is central DB
 - WLCG production services on Oracle RAC to provide high-availability and scalability



- Based on WLCG tiered model
- Full WLCG service has been in production for well over a year
- FTS servers are located at CERN and Tier-1 sites
 - Put the servers where the support is
 - CERN provides tier-0 export and “CAF” service
 - Tier-1 servers move data between themselves and to and from tier-2 sites



- **Over 10 petabytes of data exported from CERN in last year!**

- **FTS infrastructure runs well**
 - CERN and T1 sites ~understand the software
 - Most problems ironed out in the last 2 years' service challenges and experiment data challenges
 - Remainder of the problems are understood with experiments and sites and we have a plan to address them
- **Still problems with 'overall transfer service'**
 - SRMs / gridFTP / FTS / experiment frameworks: many layers and many places where things break
 - We use FTS as our tool to debug and understand the distributed 'overall transfer service'

- **Based on feedback from Amsterdam FTS workshop**
- **Better security model**
 - We now use proper certificate delegation rather than retrieval from MyProxy
- **Improved monitoring capabilities**
 - This is critical for the reliability of the ‘overall transfer service’
- **Beta SRM 2.2 support**
- **Better database model**
 - Improved performance and scalability
- **Better administration tools**
 - Make it easier to run the service
- **Placeholders for the future functionality**
 - To minimise the service impact of future upgrades

- **Focus continues upon service monitoring and making it easier to run the transfer service**
- **Continue testing SRM 2.2 support**
- **SL4 and VDT 1.6 support**
- **A variety of operational improvements**
 - We're collecting the various annoyances from service admins...
- **Improve the monitoring for service operations**
 - **FTS 2.0 now collects a lot of data about transfer service and SRM reliability that should be summarised and made available**
- **Closer integration with experiment software frameworks**