



Enabling Grids for E-science

File Transfer Software and Service SC3

Gavin McCance – JRA1 Data Management Cluster

*Service Challenge Meeting
April 26 2005, Taipei*

www.eu-egee.org



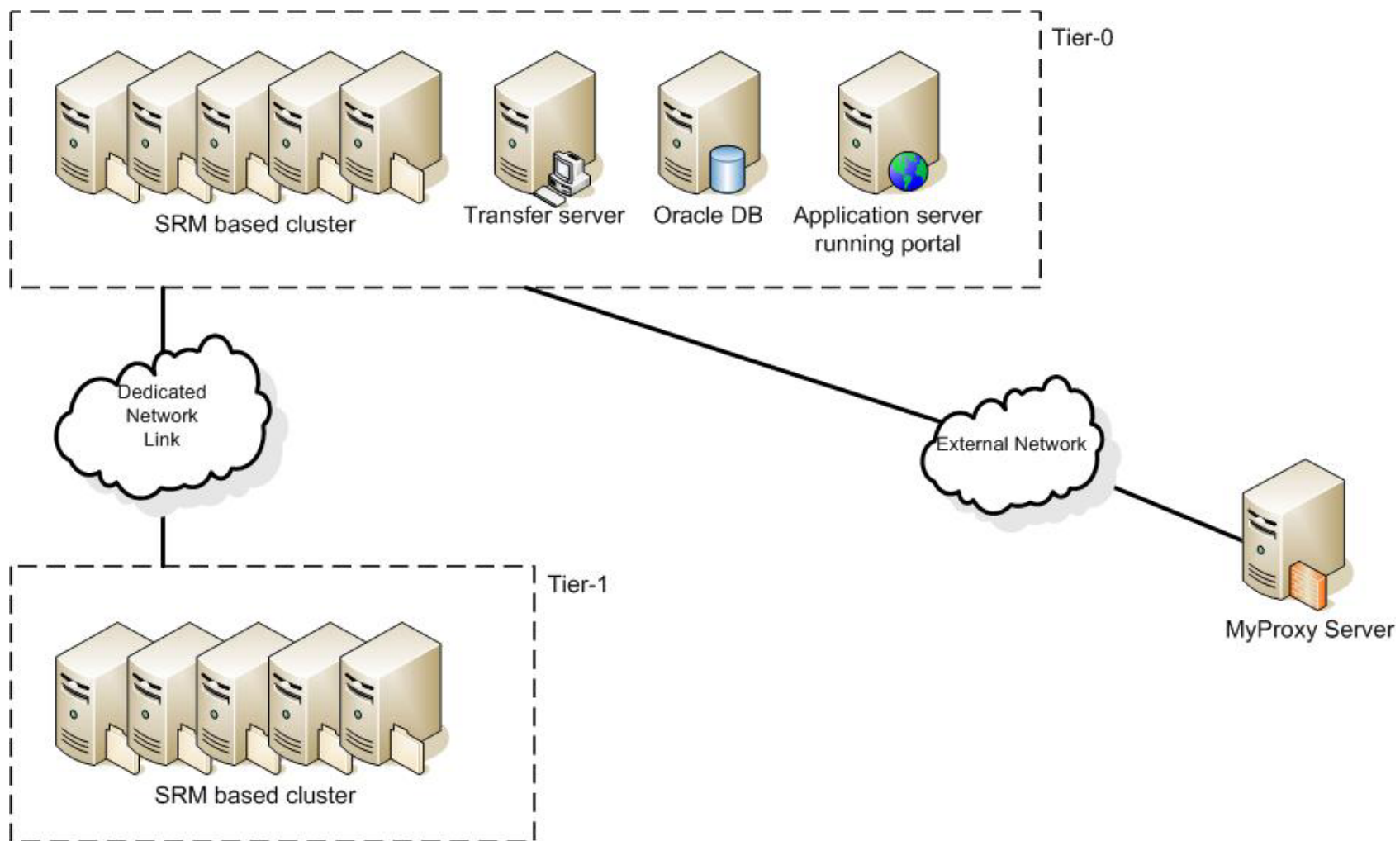
- **Overview of Components**
- **Tier-0 / Tier-1 / Tier-2 deployment proposals**
- **Initial test / early-access setup**
- **Experiment integration**

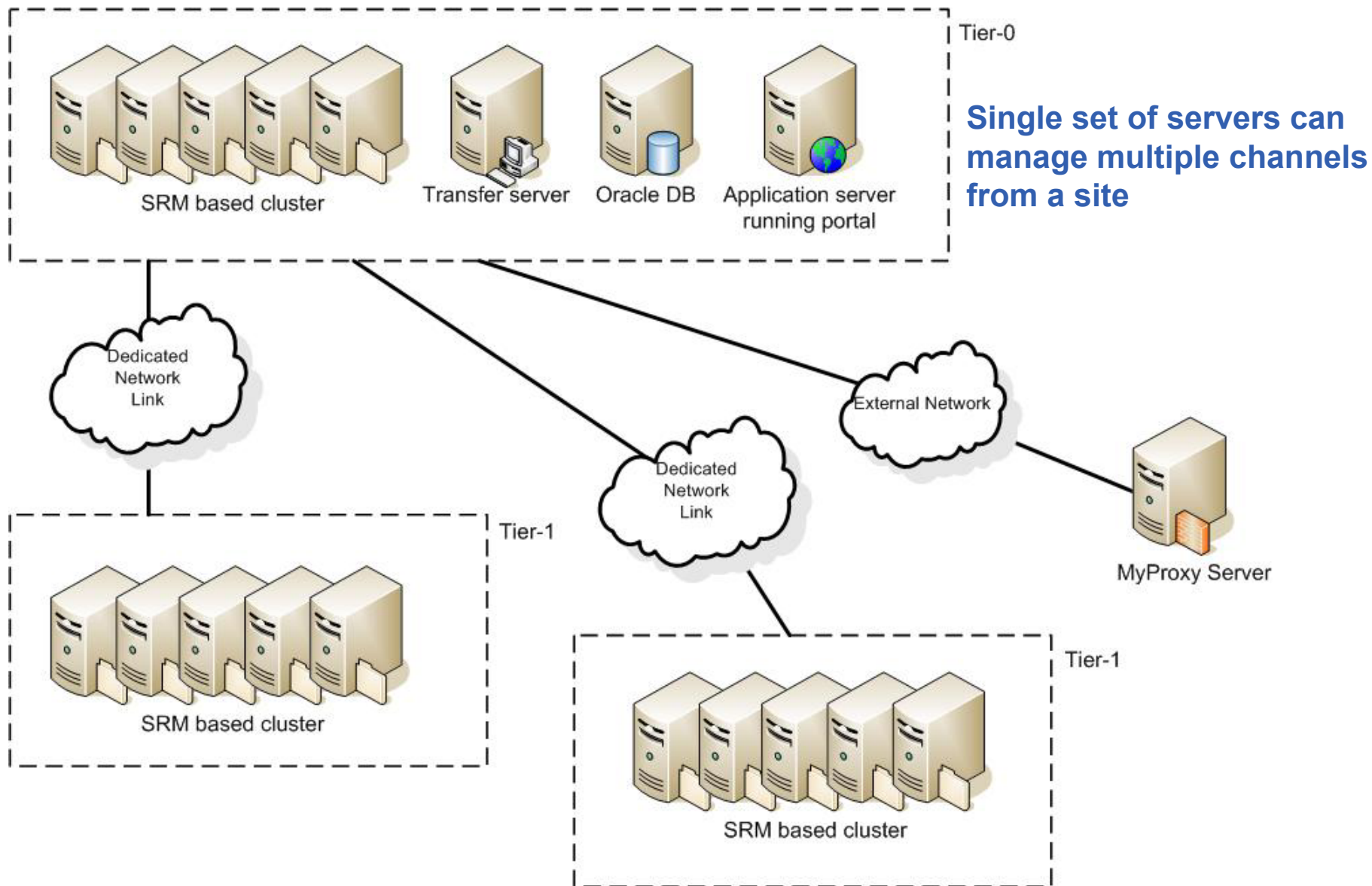
- **LCG created a set of requirements based on the Robust Data Transfer Service Challenge**
- **LCG and gLite teams translated this into a detailed architecture and design document for the software and the service**
 - A prototype (radiant) was created to test out the architecture and was used in SC1 and SC2
 - Architecture and design have worked well for SC2
 - gLite FTS (“File Transfer Service”) is an instantiation of the same architecture and design, and is the candidate for use in SC3
 - Current version of FTS and SC2 radiant software are interoperable

- **File Transfer Service is a fabric service**
- **It provides point to point movement of SURLs**
 - Aims to provide reliable file transfer between sites, and that's it!
 - Allows sites to control their resource usage
 - Does **not** do 'routing' (e.g like Phedex)
 - Does **not** deal with GUID, LFN, Dataset, Collections
- **It's a fairly simple service that provides sites with a reliable and manageable way of serving file movement requests from their VOs**
- **We are understanding together with the experiments the places in the software where extra functionality can be plugged in**
 - How the VO software frameworks can load the system with work
 - Places where VO specific operations (such as cataloguing), can be plugged-in, if required

- **Channel is a point to point network connection**
 - Dedicated pipe: CERN to T1 distribution
 - Not dedicated pipe: T2's uploading to T1
- **Focus of the presentation is upon deployment of the gLite FTS software**
 - Distinguish server software and client software
 - Assume suitable SRM clusters deployed at source and destination of the pipe
 - Assume MyProxy server deployed somewhere

- **Server software lives at *one end* of the pipe**
 - It's doing a 3rd party copy...
 - **Propose** deployment models take highest tier approach
- **Client software can live at both ends**
 - (...or indeed anywhere)
 - **Propose** to put it at both ends of the pipe
 - For administrative channel management
 - For basic submission and monitoring of jobs





- **Tier-0 and Tier-1 in the proposal:**
- **An Oracle database to hold the state**
 - MySQL is on-the-list but low-priority unless someone screams
- **A transfer server to run the transfer agents**
 - Agents responsible for assigning jobs to channels managed by that site
 - Agents responsible for actually running the transfer (or for delegating the transfer to srm-cp).
- **An application server (tested with Tomcat5)**
 - To run the submission and monitoring portal – i.e. the thing you use to talk to the system

- **Tier-0, Tier-1 and Tier-2 in the proposal:**
- **Client command-lines installed**
 - Some way to configure them (where's my FTS service portal?)
 - Currently static file or 'gLite configuration service' (R-GMA)
 - BDII? (not integrated just now)
- **Who will use the client software?**
 - Site administrators: status and control of the channels they participate in
 - Production jobs: to move locally created files
 - Or.. The overall experiment software frameworks will submit directly (via API) to relevant channel portal, or even into relevant channel DB (?)

- **Tier-0 to Tier-1 distribution**
 - **Proposal:** put server at Tier-0
 - This was the model used in SC2
- **Tier-1 to Tier-2 distribution**
 - **Proposal:** put server at Tier-1 – push
 - This is analogous to the SC2 model
- **Tier-2 to Tier-1 upload**
 - **Proposal:** put server at Tier-1 – pull
- **Other models?**
 - Probably...
 - For SC3 or for service phase beyond?

- **Initial small-scale test setups have been running at CERN during and since SC2 to determine reliability as new versions come out**
 - This small test setup will continue to smoke-test new versions
- **Expanding test setup as we head to SC3**
 - Allows greater stress testing of software
 - Allows us to gain further operational experience and develop operating procedures
 - *Critical*: allows experiments to get early access to the service to understand how their frameworks can make use of it

- **Move new server software onto CERN T0 radiant cluster**
 - Provisioning of necessary resources underway
 - Internal tests in early May
 - Staged opening of evaluation setup to willing experiments mid May
- **Start testing with agreed T1 sites**
 - As and where resources permit
 - **Same topology as SC2:** transfer software only at CERN T0
 - Pushing data to T1s mid / late May
 - Which T1s? What schedule?
- **Work with agreed T1 sites to deploy server software (which T1s?)**
 - Identify one or two T2 sites to test transfers with (which?)
 - Early June
 - Tutorials to arrange for May

- **Schedule experiments onto the evaluation setup**
- **Some consulting on how to integrate frameworks**
 - Discuss with service challenge / development team
 - Already presented ideas at LCG storage management workshop
 - Comments:
 - seems fairly easy, in principle
 - different timescales / priorities for this
- **Doing to actual work**
 - Should be staged
 - people are busy
 - easier to debug one at a time
 - Work out schedule ASAP

- **Technical discussions to happen...**
- **...this will be easier once you have an evaluation setup you can see**

- **Outlined server and client installs**
- **Propose server at Tier-0 and Tier-1**
 - Oracle DB, Tomcat application server, transfer node
- **Propose client tools at T0, T1 and T2**
 - This is a UI / WN type install
- **Evaluation setup**
 - Initially at CERN T0, interacting with T1 a la SC2
 - Expand to few agreed T1s interacting with agreed T2s
- **Experiment interaction**
 - Schedule technical discussions and work