Table of Contents

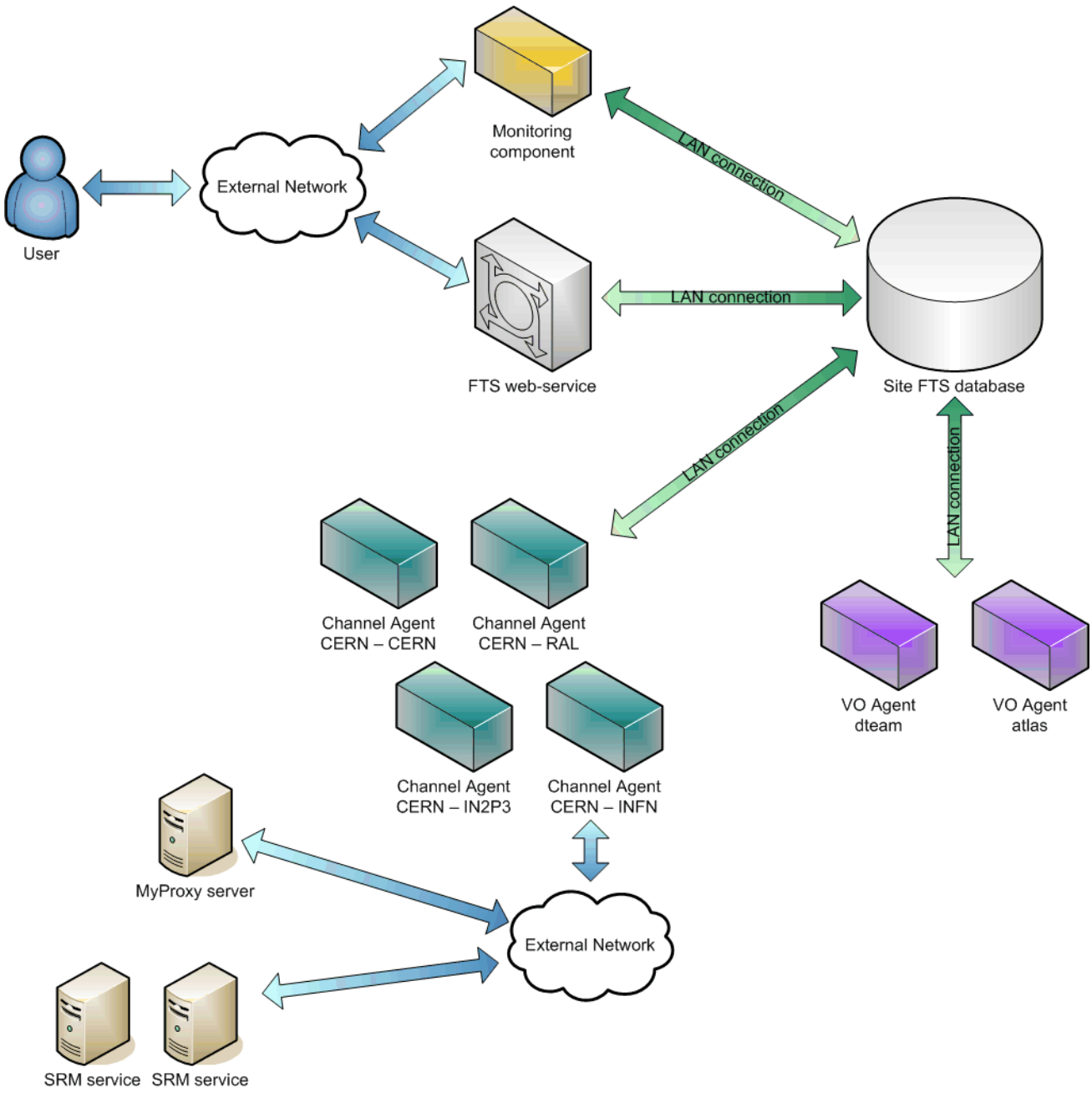
FTS service deployment

This document describes the impact of outages of service components on the overall service quality, and suggests possible deployment scenarios, and is organized as follows:

- identify the logical elements of an FTS installation
- analyze the impact of interventions and problems on different components
- suggest possible deployment scenarios for different needs, based on the previous analysis

Logical deployment

The following diagram shows the logical elements of an FTS installation.



FTS is constituted by:

- one or more web-service (tomcat) servers. The web-service is stateless, so it's possible to install several servers on different nodes for higher availability.
- one channel agent daemon for each served channel (4 in the example).
- one VO agent daemon for each served Virtual Organization.
- one (optional) monitoring component.
- the central database

All components communicate with the central database through the internal LAN connection.

External connectivity is needed for

- the user to interact with the web service, for jobs submissions and service administration
- the service (agent daemons) to interact with other grid services (MyProxy, SRM interfaces over storage elements)

Service review

A description of the impact on the overall service of interventions and problems on different components.

Service components and service impact of outages

FTS is split into three distinct components, so there are three main classes of availability to consider:

1. **web-service**: ability to submit new jobs, query their status and administer the channels. This is determined by the availability of the FTS web-service.
2. **data transfer**: the file transfer jobs currently in the system are being processed correctly. This is determined by the availability of the FTS agents and the other external services upon which the FTS depends.
3. **monitoring**: the monitoring system. This is determined by the availability of the node (and apache server) that publishes the monitoring information.

All three parts of the service are well factorised from each other, so one part of the service can be down while the others stay up.

All three components synchronise on the backend database - if this database in unavailable, all parts of the service are down.

FTS web-service

The FTS web-service is used:

- by clients to submit new jobs and query the status of existing ones
- by VO administrators to cancel jobs and change the priorities of existing ones
- by site administrators to change the properties of the transfer channels affecting their site

It is a stateless SOAP-based RPC web-service running inside the Tomcat 5.5 J2EE container.

Because it is stateless, it can be trivially load-balanced and this procedure also increases the availability. The currently recommended deployment scheme is to use DNS load-balancing. Upon loss of one node, the DNS will be made to point at remaining nodes only; this is either automatic or an operator procedure depending on the type of failure.

Impact of downtime of one of the nodes:

- Class 1 service (web-service): will run in degraded mode (potentially exhibiting overload on the remaining nodes in the cluster); excess requests should be cleanly refused.
- Class 2 service (data transfer): no impact
- Class 3 service (monitoring): no impact.

Intervention type:

- Automatic - the DNS load-balancing dropout should be set up to happen automatically if a node fails or the web-service on it become unresponsive. Subsequent recovery of the failed node is manual.

Resilience to glitches:

- Poor. Short glitches will be noticed by clients since the DNS propagation is not fast enough to hide it.

Recovery:

- Upon restart of the problematic node (after DNS propagation) the service is fully back up. No state will be lost (job submission requests are atomic).

FTS agents

The FTS agent daemons are responsible for processing the job (i.e.for doing the file copies). There are three different types of agent:

- Channel agent. You run one of these daemons for every channel defined in the system. Each is responsible for running the transfers on its associated channel. Typically several 10's.
- VO agent. You run one of these daemons for every VO defined in the system. Each is responsible for handling the requests belonging to its VO. Typically under 10.
- Proxyrenewal agent. This renews expired proxies from MyProxy. You run one of these.

For load-balancing the configuration allows the agent daemons to be spread arbitrarily over multiple nodes.

The agents are completely factorised from each other - loss of one of them will not affect the correct operation of the others.

Channel agents

Downtime of a channel agent will result in all transfers on that channel being suspended. The currently **running** transfers will finish, but no new transfers will be put on the network. Work queued on the channel will stay queued until the channel agent is back up to serve it again. Symptom: unstarted jobs will stay in `Ready` state, partially served jobs will stay in `Active` state.

Impact of downtime of a channel agent:

- Class 1 service (web-service): no impact.
- Class 2 service (data transfer): 100% stoppage for the given channel only. There is no impact on the service provided by the other channel agents.
- Class 3 service (monitoring): no impact.

Intervention type:

- Manual. There is no automatic fail-over supported by the software.

Resilience to glitches:

- a few minutes - currently running (i.e. on the network) transfers will continue to run while the agent is down, so a short stoppage of the agent will result in no noticeable impact on the hourly throughput.

Recovery:

- once the agent restarts, the transfers will start again; no jobs or job-state will be lost.

VO agents

Downtime of a VO agent will result in no new jobs being assigned for that VO to any channel, and jobs which have finished not being moved to their `Finished` state. Jobs currently assigned to a channel will continue running (so the data export will continue) but eventually the channel queues will become exhausted (for the given VO) since no new jobs from that VO are being assigned to them. The symptom is that all new jobs will

be stuck in `Submitted` state and all running jobs will be stuck in `Done` or `Failed` state.

Impact of downtime of a VO agent:

- Class 1 service (web-service): no impact.
- Class 2 service (data transfer): gradual degradation for the given VO only. The other VOs are not affected. State machine for transfers which have finished will not be updated while the agent will down, so clients will not 'see' finished jobs as finished. Job cancellation will not function reliably.
- Class 3 service (monitoring): no impact.

Intervention type:

- Manual. There is no automatic fail-over supported by the software.

Resilience to glitches:

- several 10s of minutes to several hours, depending on the depth of the VOs queue in the FTS: provided there are jobs assigned to channels, they will process at the normal export rate.

Recovery:

- once the VO agent restarts, the agent will assign all new jobs to the correct channel and update the state of 'finished' jobs. No jobs or job state will be lost.

FTS monitoring component

The (to be released) FTS monitoring component will run on a standard apache2 (httpd) server. It is currently not expected to run more than one for either load-balancing or availability.

Impact of downtime of a single apache server:

- Class 1 service (web-service): no impact.
- Class 2 service (data transfer): no impact.
- Class 3 service (monitoring): 100% downtime; potential loss of summarisation data for items that are generated by scripts running on the node (if the node's filesystem is lost). The raw data (from which the summaries are calculated) should remain in the DB.

Intervention type:

- Manual - the apache server or node must be recovered.

Resilience to glitches:

- Poor - any glitches on the server will not be hidden from clients of the monitoring data.

Recovery:

- once the monitoring server is back up, the dynamic data generation will restart.

Oracle database component

All parts of the system sychronise their state and obtain information from the backend Oracle database cluster.

Impact of downtime of the Oracle RAC:

- Class 1 service (web-service): 100% unavailability on all nodes: users will receive a "Can't connect to DB" message.
- Class 2 service (data transfer): 100% stoppage on all channels.
- Class 3 service (monitoring): degradation - new reports will not be generated and dynamic data will not be available. Statically generated reports (e.g. daily summaries) should remain available.

Intervention type:

- Automatic - all servers will keep retrying to to connect to the DB

Resilience to glitches:

- Poor - glitches on the DB are exposed as service problems to the clients and stoppages in the transfers.

Recovery:

- assuming full database recovery, no state will be lost and the FTS will resume when the DB is back.

External network connectivity

The service requires access to the external network to transfer data.

Impact of downtime of the external network:

- Class 1 service (web-service): unavailability of all nodes to external clients.
- Class 2 service (data transfer): The software remains operational and correctly records the network errors, but the overall service will register 100% failures on all channels ("can't connect to remote SRM").
- Class 3 service (monitoring): unavailability to external clients.

Intervention type:

- Automatic - the service recovers OK once the network is back. A manual procedure to pause failing channels can help control the impact of this.

Resilience to glitches:

- Poor - clients cannot connect to service and file transfers fail.

Recovery:

- Automatic - the service will recover when the network returns. Failed transfers are retried by default.

Internal network connectivity

Regarding the individual switches that the service nodes are connected to. The services should be balanced over as many switches as practicable to ensure that the outage of any one switch doesn't affect too much of the service.

Impact of downtime of one internal switch:

- Class 1 service (web-service): 100% unavailability of any web-service nodes on that switch. The DNS load-balancing should be configured to detect this and remove these nodes from the alias. The other

nodes on good switched are unaffected.
- Class 2 service (data transfer): 100% unavailability of any agents running on nodes connected to the switch. The behaviour is the same as if the individual agents had gone down and is described earlier. Provided the DB RAC is not routable from these nodes behind the same failing switch, only the currently active transfers on the network will fail - since the agent on the failing nodes will not be able to access to DB for new jobs to attempt. Other agents on different switches will not be affected.
- Class 3 service (monitoring): unavailability to any clients.

Intervention type:

- Automatic - the service recovers OK once the network is back.

Resilience to glitches:

- Poor - clients cannot connect to service and file transfers fail.

Recovery:

- Automatic - the service will recover when the network returns. Very few file transfers should fail (assuming the node behind the failing switch cannot see the database). No state will be lost.

Summary

Faulty component	Effects of downtime on:			Intervention type	Resilience to glitches	Recovery
	web-service	data transfer	monitoring			
Web service	Run in degraded mode. Should refuse excess requests	none	none	Automatic DNS load-balancing dropout. Manual recovery of failed node.	Poor (DNS propagation).	Complete recovery upon service restart. No jobs or jobs' state loss.
Channel agents	none	Complete stop for managed channel. No impact on the service provided by other agents.	none	Manual	Short (few minutes) stop of an agent should not result in noticeable impact	Complete recovery upon agent restart. No jobs or jobs' state loss.
VO agents	none	Gradual service degradation for managed VO. Job cancellation unreliable. No impact on the service provided by other agents.	none	Manual	From 10s of minutes to hours	Complete recovery upon agent restart. No jobs or jobs' state loss.
Monitoring component	none	none	100% downtime. Potential loss of summarisation data.	Manual	Poor	Dynamic data generation will restart with the service.

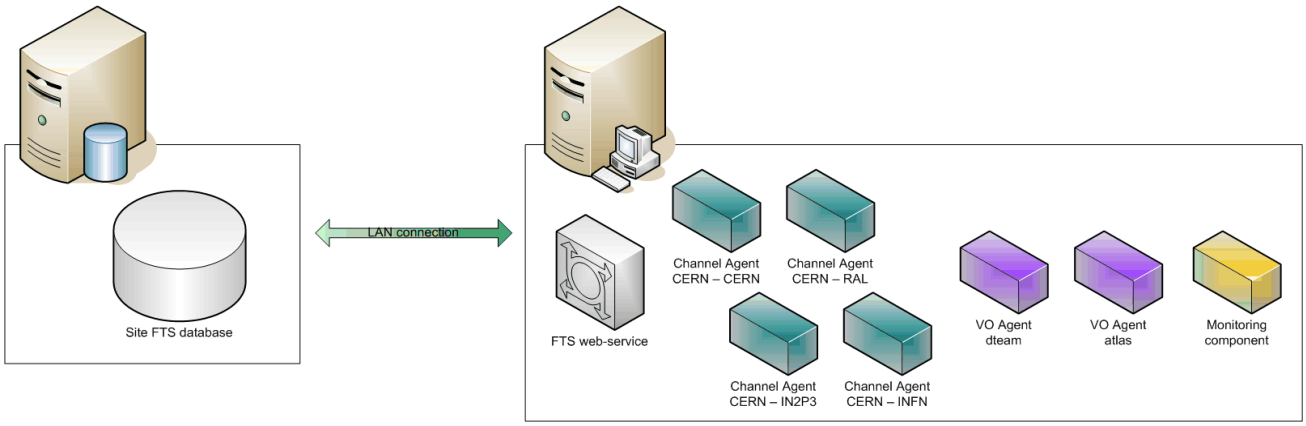
Oracle database	100% unavailabilty.	100% stoppage.	Degradation. New reports not generated Dynamic data not available. Static data still available.	Automatic (all servers will keep retrying to connect).	Poor.	No loss assuming full DB recovery.
External network connectivity	Unavailability to external clients.	All transfers will fail (cannot connect to SRMs).	Unavailability to external clients.	Automatic.	Poor.	Automatic.
Internal network connectivity	100% unavailability of web-service nodes on the faulty switch.	100% unavailabilty of agents running on nodes connected to the faulty switch.	Unavailability to all clients.	Automatic.	Poor.	Automatic.

Deployment scenarios

Simple deployment

The most basic deployment consists in running all the daemons (web service, agents and optionally monitoring component) on a single machine, connected over the LAN to the DB server.

Suitable only for very low traffic or test installations.

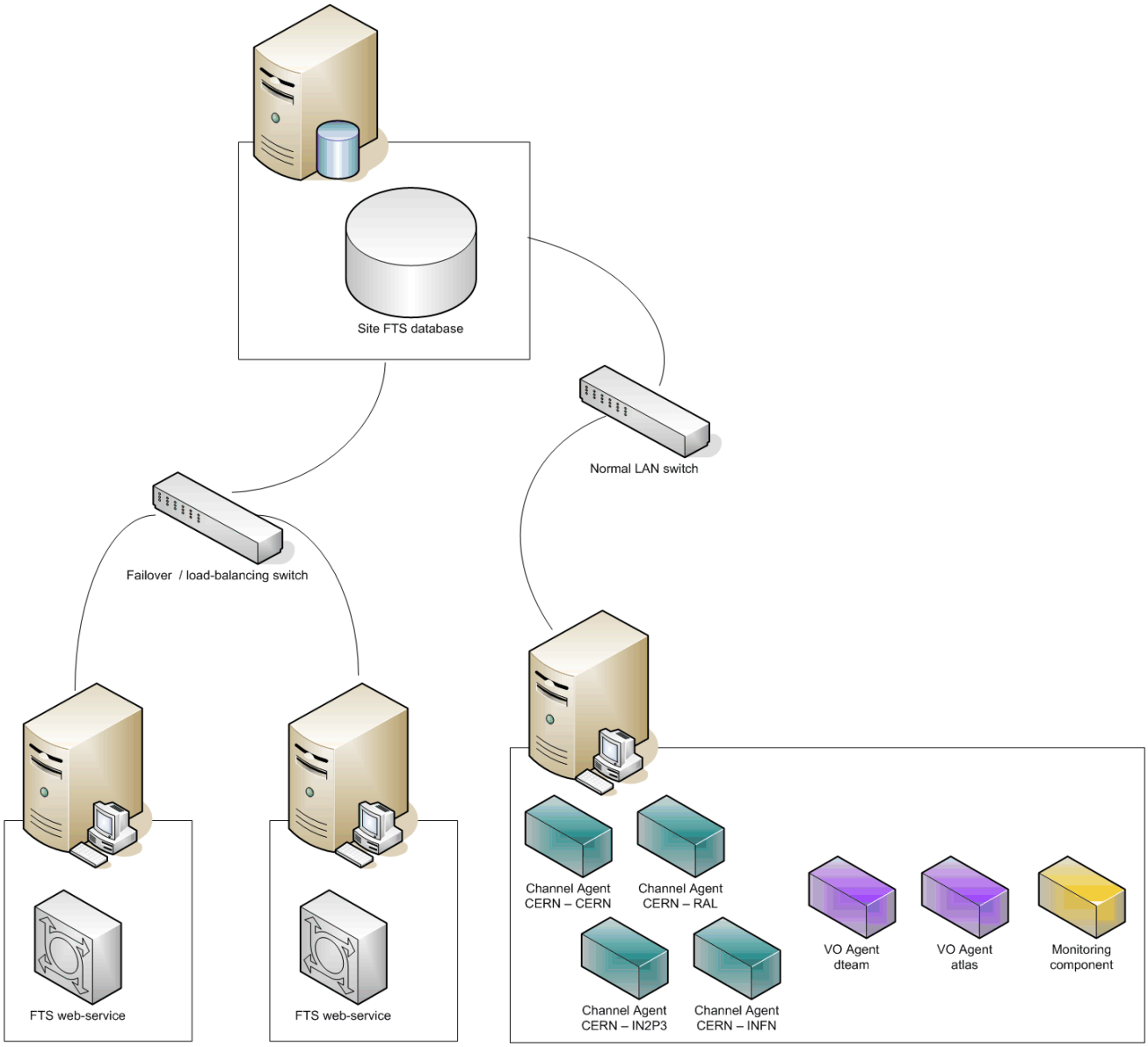


Split deployment

The model used for FTS's pilot service.

The web service is load-balanced over more than one machine.

Agents' daemons run on a separate box.



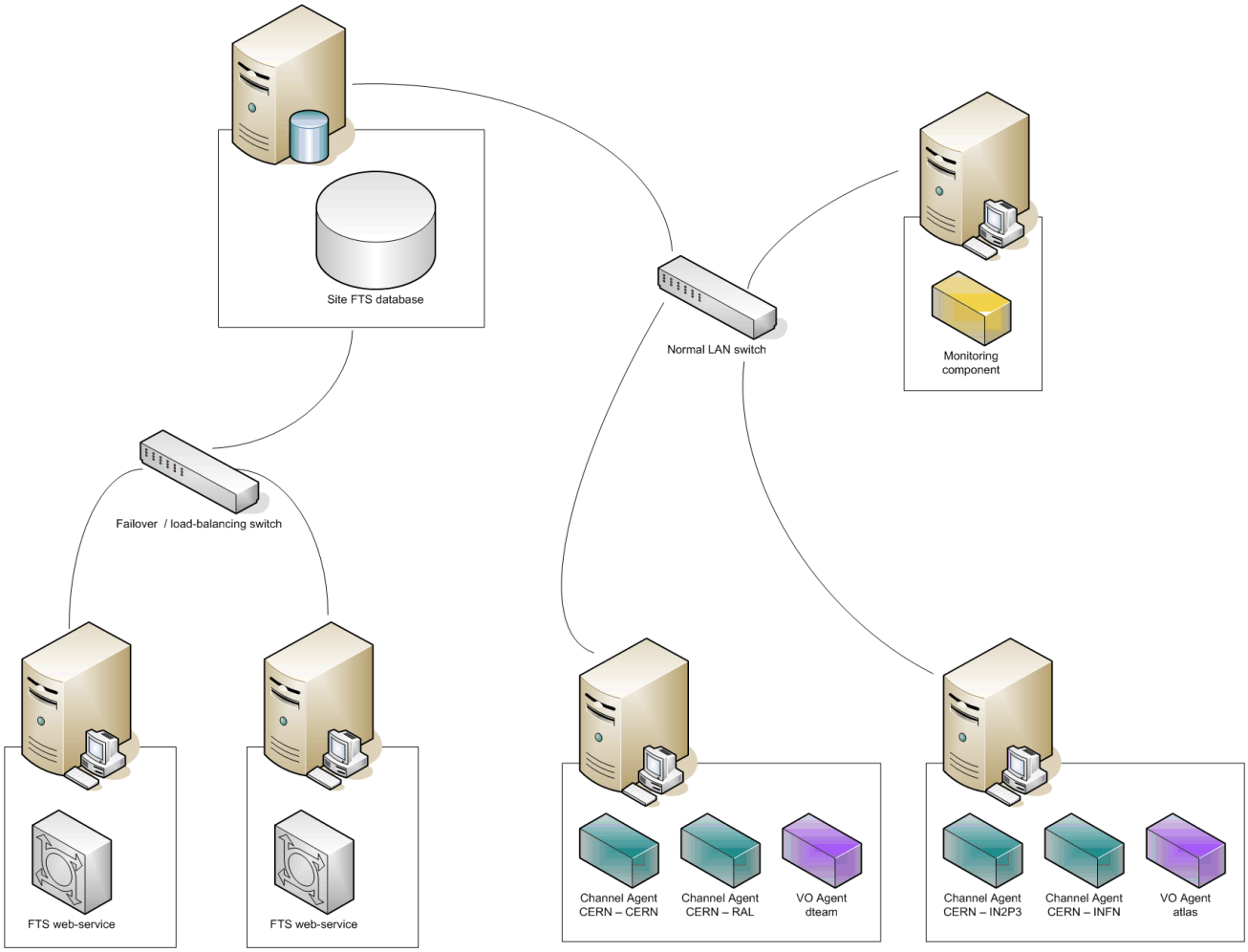
High availability deployment

This is the recommended configuration for WLCG tier-1 sites.

A deployment model for very high loads, granting high availability and reducing impact of nodes maintenance or fault.

The web service is load-balanced over more than one machine.

Agents' daemons are split across several machines.



Last edit: PaoloTedesco on 2009-01-14 - 10:19
Number of topics: 1
Maintainer: PaoloTedesco

This topic: LCG > FtsServiceDeploymentModel
Topic revision: r6 - 2009-01-14 - PaoloTedesco