

Datagrid WP4: Fabric Management

Enrico Ferro - LNL

- ❖ **Datagrid WP4: overview**
- ❖ **WP4, Installation & Configuration subtask: overview**
- ❖ **LCFG**
- ❖ **Sviluppi futuri**

- ❖ **Il progetto è diviso in “Work Packages”:**
 - **WP1: Workload Management**
 - **WP2: Data Management**
 - **WP3: Monitoring Services**
 - **WP4: Fabric Management**
 - **WP5: Mass Storage Management**
 - **WP6: Testbed**
 - **WP7: Network**
 - **WP8-10: Applications**
- ❖ **La durata complessiva è di 3 anni (2001-2003).**
- ❖ **Milestones: month 9 (Sept 2001), month 21 (Sept 2002), month 33 (Sept 2003)**

- ❖ “Aims to deliver a computing fabric comprised of all the necessary tools to manage a centre providing Grid services on clusters of thousands of nodes”
- ❖ Si tratta di strumenti orientati ad una gestione della farm
 - totale
 - centralizzata
- ❖ WP4 è diviso in 6 tasks/sottosistemi
 - Configuration management (CERN + PPARC)
 - Resource management (ZIB)
 - Installation & node management (CERN + INFN + PPARC)
 - Monitoring (CERN + INFN)
 - Fault tolerance (KIP)
 - Gridification (NIKHEF)

Architecture overview

- Interface between Grid-wide services and local fabric;

- Provides local authentication, authorization and mapping of grid credentials.

- provides transparent access to different cluster batch systems;

- enhanced capabilities (extended scheduling policies, advanced reservation, local accounting).

- provides the tools to install and manage all software running on the fabric nodes;

- software repositories;

- Node management to install, upgrade, remove and configure software packages on the nodes.

Resource Broker (WP1)

Grid Info Services (WP3)

Fabric Gridification

Resource Management

Farm A (LSF)

Farm B (PBS)



Installation & Node Mgmt

Monitoring & Fault Tolerance

Configuration Management

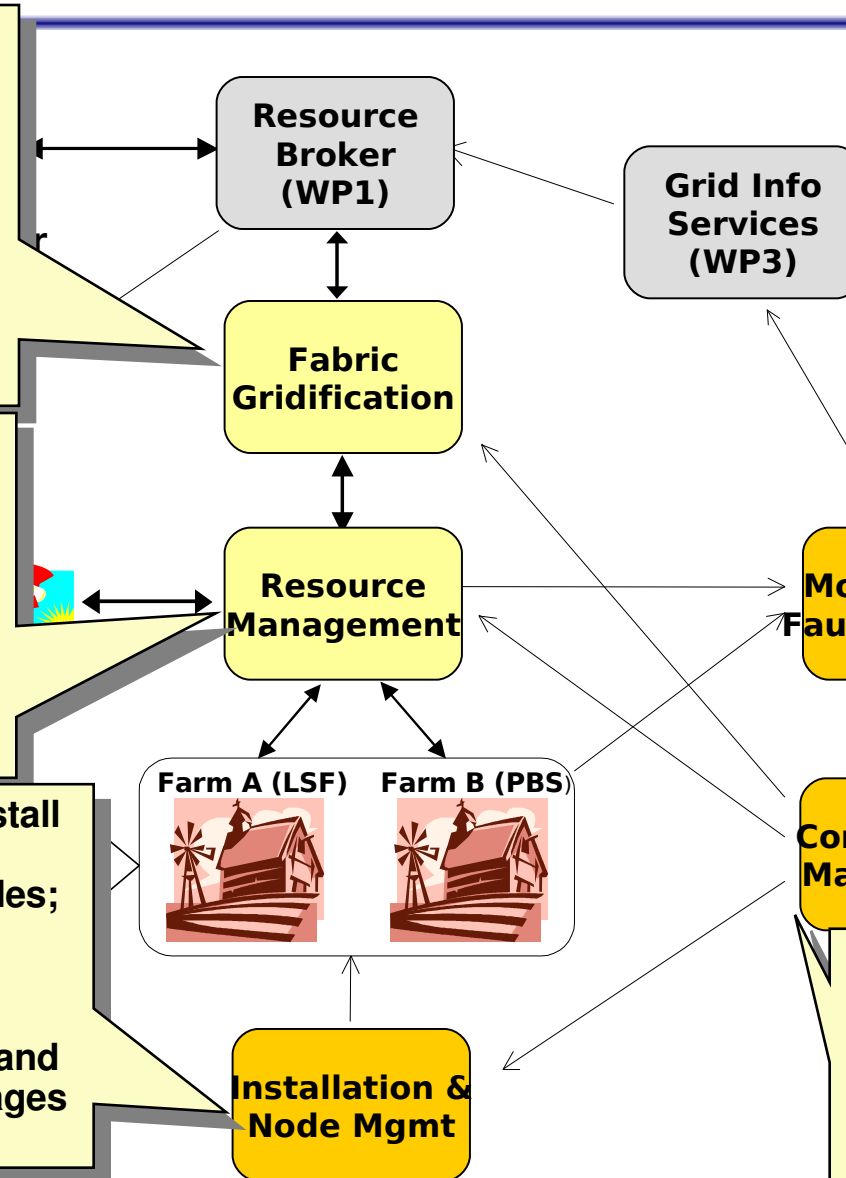
- provides the tools for gathering and storing performance, functional and environmental changes for all fabric elements;

- central measurement repository provides health and status view of services and resources;

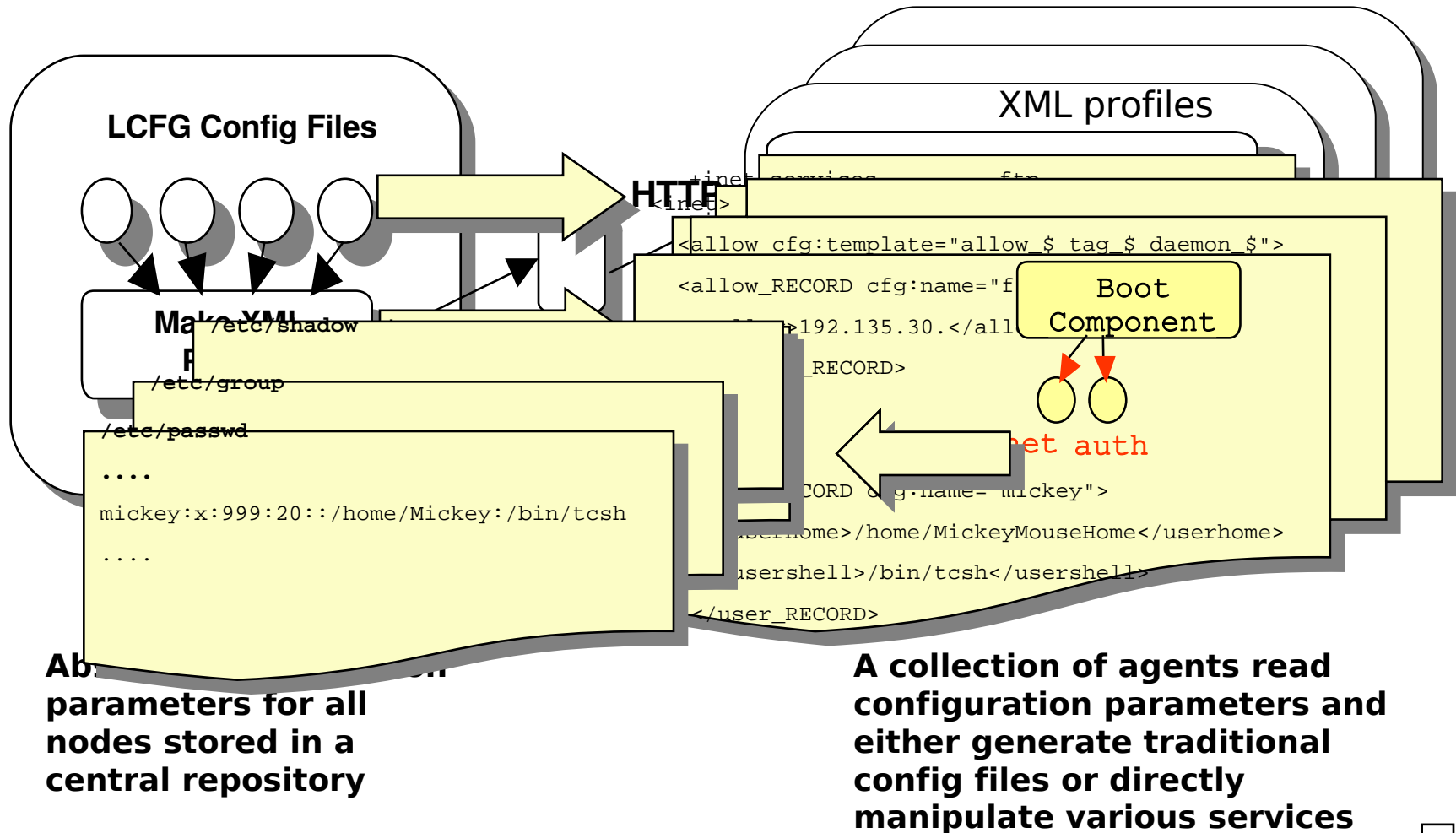
- fault tolerance correlation engines detect failures and trigger recovery actions.

- provides a central storage and management of all fabric configuration information;

- central DB and set of protocols and APIs to store and retrieve information.



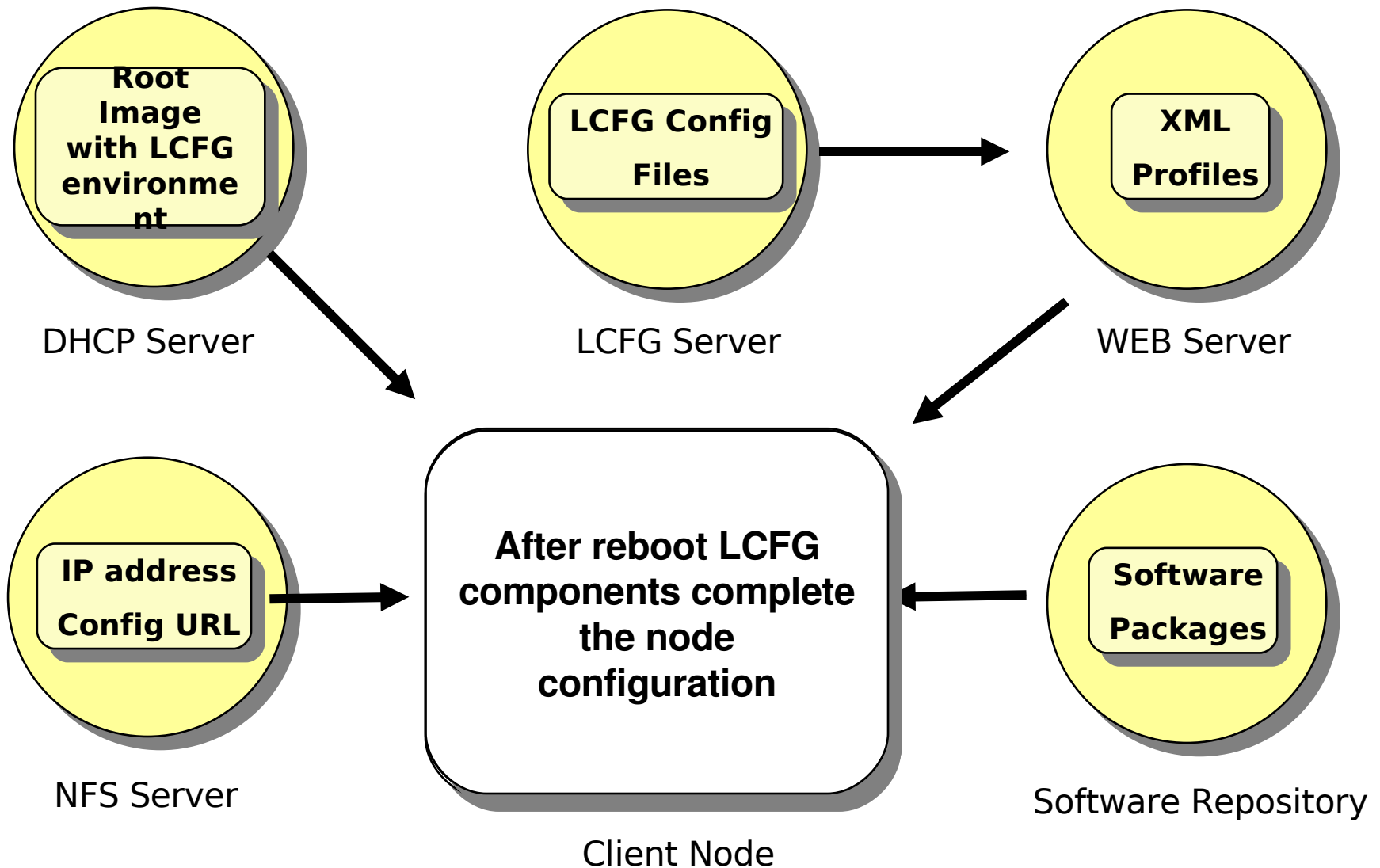
- ❖ E' basato su un tool sviluppato all'Università di Edinburgo: LCFG (Large Scale Linux Configuration)
- ❖ Caratteristiche principali:
 - installazione automatica del S.O.
 - installazione/upgrade/remove di tutti i pacchetti software (basato su RPM)
 - gestione centralizzata della configurazione
 - modulare
 - estendibile
- ❖ <http://www.lcfg.org>



- ❖ All'avvio, viene caricato un demone (*rdxprof*) che resta in ascolto su una porta UDP. Inoltre un componente "master" (*boot*) esegue tutti i componenti di LCFG.
- ❖ Quando e' notificato un cambiamento della configurazione dal server, se necessario viene riscaricato il profilo (=documento XML con la configurazione completa del nodo) via HTTP
- ❖ Una volta ricevuto il profilo XML, viene salvato su una cache locale e sono riattivati i componenti la cui configurazione è mutata
- ❖ E' anche possibile che i clients interroghino periodicamente il server (*polling*)
- ❖ Ci sono componenti non supportano la possibilità di riconfigurarsi "a caldo" (es. modifica partizioni)

- ❖ È uno script shell/PERL
- ❖ Ciascun componente fornisce un certo numero di “metodi” (start, stop, config,...) che sono invocati al momento opportuno (all'avvio del sistema, in fase di shutdown, in caso di cambiamento della configurazione, etc...)
- ❖ Accede alla cache locale contenente la configurazione del nodo
- ❖ Configura gli opportuni servizi, traducendo i parametri di configurazione in normali files di configurazione (es. inetd.conf) oppure gestendo direttamente i demoni

- ❖ **LCFG mette a disposizione i componenti per gestire tutti i servizi standard di una macchina: inet, syslog, nfs, cron, ...**
- ❖ **Un amministratore può creare nuovi componenti custom per configurare e gestire le applicazioni nei propri sistemi:**
 - **definisce i nomi e la struttura gerarchica dei propri parametri di configurazione in un file .def (le informazioni da aggiungere al profilo XML)**
 - **crea il componente (è uno script); deve includere il componente "generic", nel quale sono definite delle funzioni comuni (accesso ai dati, log, messaggi standardizzati, ...)**
 - **ridefinisce i "metodi" vuoti (start, stop, config, ...) con il proprio codice e utilizzando i servizi offerti da "generic"**



❖ Pro:

- In uso da anni in un ambiente complesso ed eterogeneo, con centinaia di nodi da gestire, dal cluster ai notebook
- Supporta la completa installazione e gestione di tutto il software (sia O.S. che applicazioni)
- Molto flessibile, estendibile e customizzabile

❖ Contro:

- Complesso: curva di apprendimento iniziale ripida
- Manca monitoraggio centralizzato in caso di problemi sui client (es. un componente non riesce a killare un demone)
- Mancanza di tools user-friendly per la creazione e gestione dei files di configurazione (ed eventuali errori possono essere molto pericolosi!).
- Il software da installare deve essere in formato RPM

- ❖ Nuova versione di LCFG (fine maggio)
- ❖ Linguaggio di alto livello per la configurazione (attualmente è descritta con semplici coppie nome+valore), check più rigorosi sulla consistenza della configurazione
- ❖ Miglioramento del supporto a RPM
- ❖ Integrazione con software del Monitoring Subtask di WP4
- ❖ Scalabilità
- ❖ Passaggio ad HTTP per la distribuzione dei package da installare (attualmente NFS)
- ❖ Installazione con PXE
- ❖ Installazione con tools RedHat (*kickstart files*)