GRID

ossia

Il calcolo distribuito mediante griglie computazionali

Massimo Masera

Dipartimento di Fisica Sperimentale dell'Universita` e I.N.F.N.

Torino

(ringrazio Federico Carminati, Fabrizio Gagliardi e Claudio Grandi a cui ho "*rubato*" molte delle trasparenze che seguono)

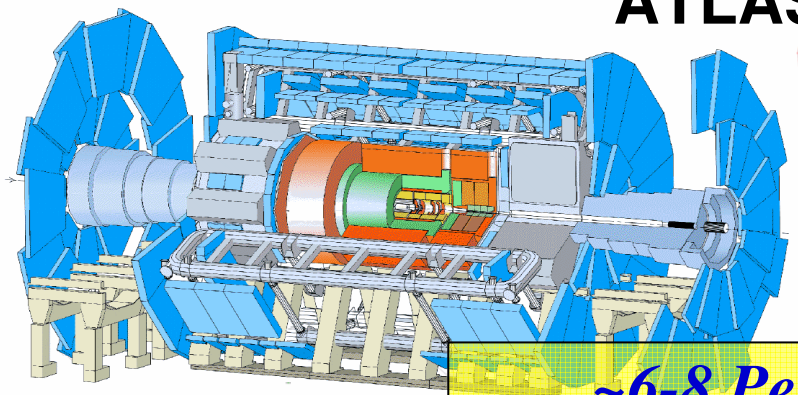
Sommario

- ✓ Il problema del calcolo al Large Hadron Collider (LHC)
  - ➢ Necessita` di usare risorse distribuite su scala mondiale
- ✓ Gestione di risorse distribuite: il paradigma della GRID.
  - ➢ Un esempio: GLOBUS
- ✓ Le sperimentazioni in corso nell'INFN
- ✓ Possibile interesse per applicazioni diverse dalla Fisica delle Alte Energie
- ✓ LCG (LHC Computing Grid)

Gli esperimenti al LHC

ATLAS

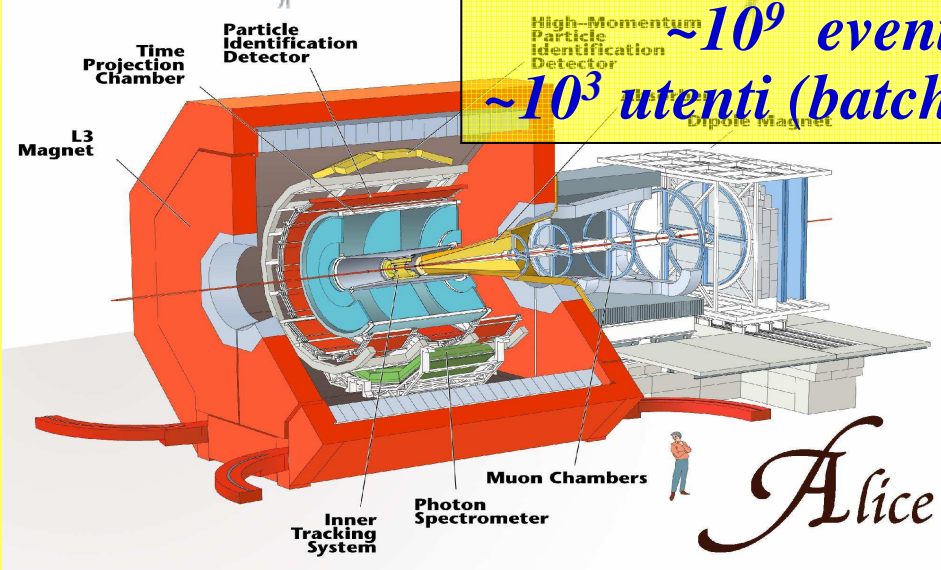
CMS

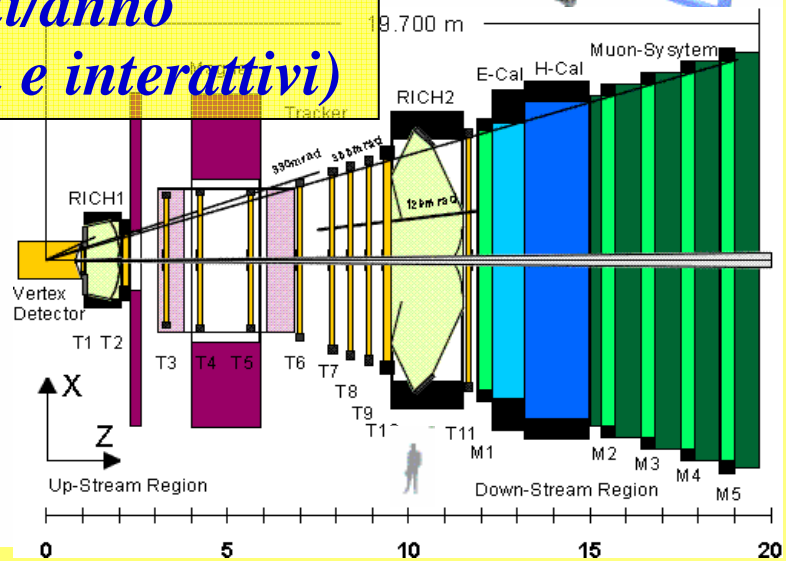


~6-8 PetaBytes / anno

~10^9 eventi/anno

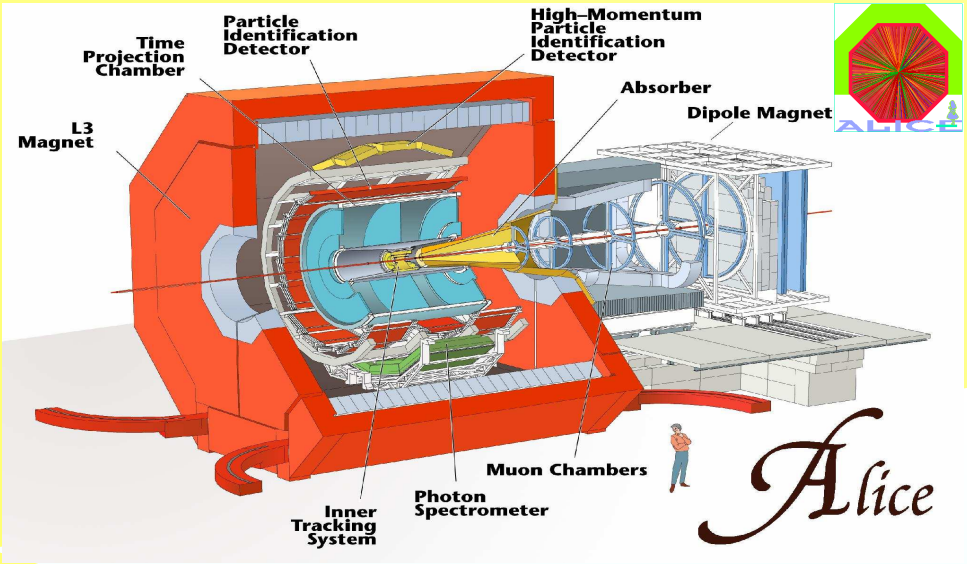
~10^3 utenti (batch e interattivi)





LHC*b*

Alice collaboration



online system

multi-level trigger
filter out background
reduce data volume

Total weight	**10,000t**
Overall diameter	**16.00m**
Overall length	**25m**
Magnetic Field	**0.5 Tesla**

8 kHz (160 GB/sec)

level 0 - *special hardware*

200 Hz (4 GB/sec)

level 1 - *embedded processors*

30 Hz (2.5 GB/sec)

level 2 - *PCs*

30 Hz (1.25 GB/sec)

data recording & offline analysis

Total weight	**2,000t**
Overall length	**17.3m**

Total weight	**53,000t**
Overall length	**270.4m**

E il calcolo? Come? Dove?

- ✓ I bisogni del calcolo a LHC sono enormi, per esempio per ALICE
  - 1.25 GB/s durante i run con ioni pesanti
  - ~4 PB/anno di nastri, ~2 PB di dischi
  - ~22000 kSI2000 (~30,000 CPU 2003)
  - ~ 8MEuro di solo hardware
  - Milioni di linee di codice da sviluppare e mantenere per >20 anni
- ✓ Ci sono ragioni pratiche che impongono nuove soluzioni rispetto al passato
  - La complessita` dei programmi di simulazione ed analisi e l'elevato numero di utenti (qualche centinaio per esperimento) ha imposto l'adozione di nuovi linguaggi di programmazione, quali il C++, che dovrebbero facilitare la gestione e il mantenimento di programmi grandi
  - La concentrazione di grandi risorse in un unico centro (il CERN) pone evidenti problemi ...

Centri legati al CERN nel mondo

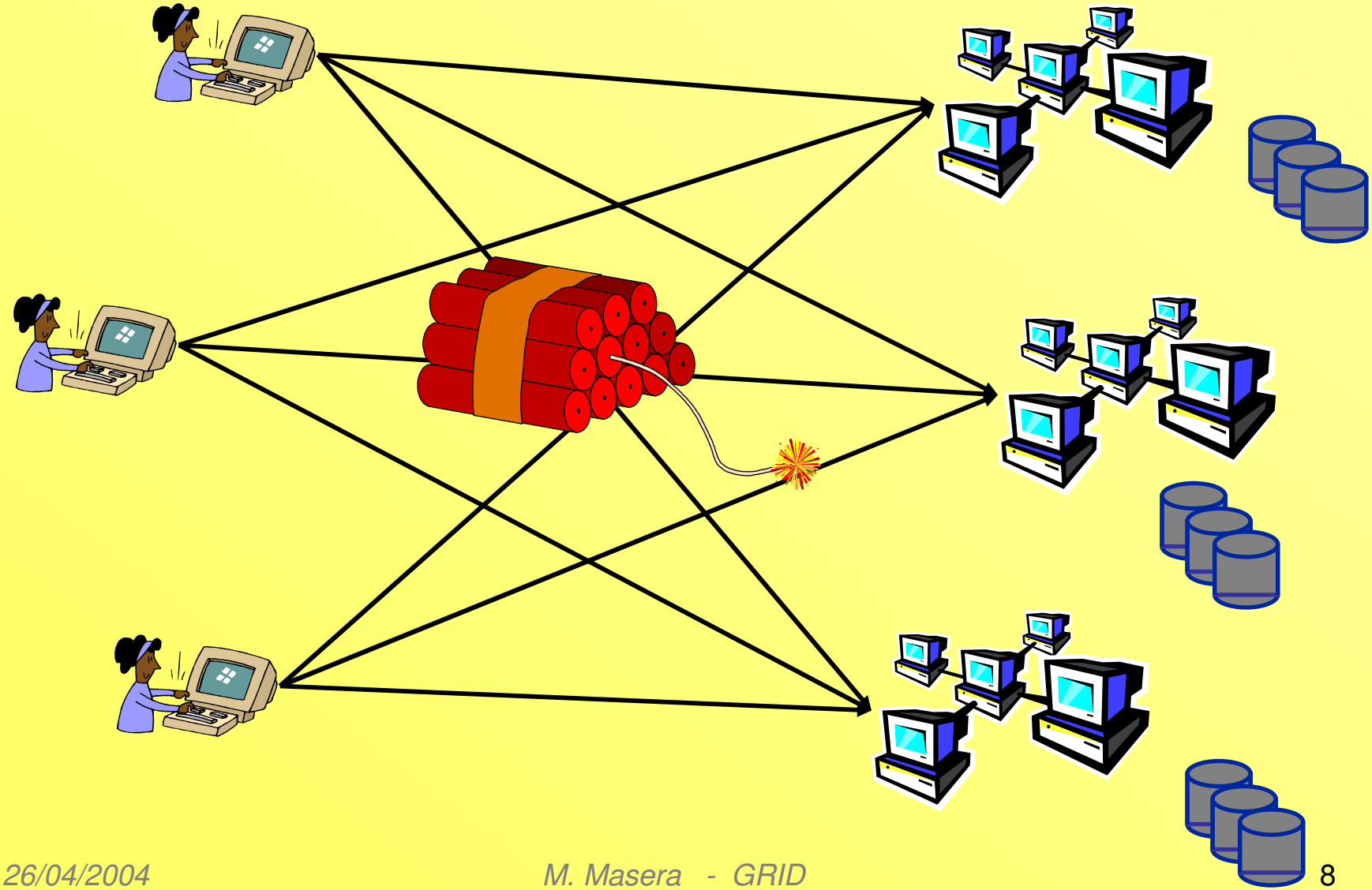
Europa:	*267 Istituti, 4942 utenti*
Altrove:	*208 Istituti, 1752 utenti*

Perché il calcolo distribuito?

- ✓ Tecnicamente, politicamente e sociologicamente non si può concentrare tutto al CERN
  - ➢ Le “funding agencies” (per noi l’INFN) preferiscono investire il proprio denaro all’interno piuttosto che all’estero
  - ➢ Computing scientist inviati all’estero con scarsa ricaduta per il Paese di provenienza
  - ➢ Le competenze sono naturalmente distribuite
  - ➢ Il modello che prevede un unico grande centro e` intrinsecamente poco robusto; non ci sono possibilita` di backup

- ✓ La disponibilita` di reti, relativamente economiche, ad alta velocita` consente di creare una **rete di centri** regionali **distribuiti** nei vari Paesi
- ✓ Ma il modello distribuito è tecnicamente più difficile da realizzare e in principio globalmente più costoso
- ✓ La sfida è quindi massimizzare l’efficienza minimizzando i costi

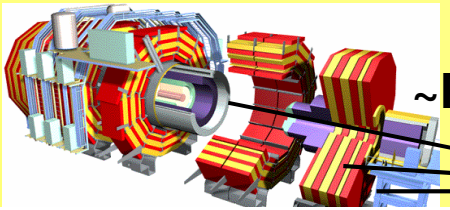
Accesso a risorse distribuite



MONARC

- ✓ Il problema legato alle risorse distribuite per il calcolo e` gia` nato nell'era LEP (si pensi all'uso di ***CONDOR*** per la condivione di cicli di CPU)
- ✓ Un primo modello riguardante l'architettura di un sistema distribuito e` stato proposto alla fine degli anni 90 dal progetto **MONARC** *(Models of Networked Analysis at Regional Centres for LHC experiments)*
- ✓ Le risorse di calcolo e storage sono suddivise secondo uno schema gerarchico su piu` livelli → *"TIERS"*

Modello di calcolo *a-la* Monarc



Esperimento (CMS)

~PByte/sec

Online System

~100-1500 MBytes/sec

Tier 0 +1

CERN Center
PBs of Disk;
Tape Robot

Tier 1

~2.5-10 Gbps

IN2P3 Center

RAL Center

FNAL Center

CNAF Center

2.5-10 Gbps

Tier 2

Tier2 Center TORINO

~2.5-10 Gbps

Tier 3

Institute

Physics data cache

0.1 to 10 Gbps

Workstations

Tier 4

Dal punto di vista dell'uso degli strumenti di calcolo, e` necessario "nascondere" all'utente la complessità del sistema sottostante

Il modello di calcolo distribuito

- ✓ Idea di base
  - ➢ Ogni fisico deve avere in principio uguale accesso ai dati e alle risorse di calcolo
- ✓ Il sistema finale sarà molto complesso a causa del numero
  - ➢ dei siti e dei componenti in ogni sito
  - ➢ delle diverse operazioni da effettuare in parallelo: simulazione, ricostruzione, analisi coordinata e caotica
- ✓ La cattiva notizia è che gli strumenti di base mancano (con *production quality*)
  - ➢ Gestione di risorse distribuite
  - ➢ Spazio virtuale distribuito dei nomi dei files e degli oggetti
  - ➢ Autenticazione e autorizzazione distribuita
  - ➢ Gestione di grandi cluster locali
  - ➢ Replicazione e caching dei dati
  - ➢ Monitoraggio e diagnostica delle reti WAN/LAN
- ✓ La buona notizia è che non siamo soli
  - ➢ Tutti i problemi elencati sopra sono al centro degli sviluppi che in Europa e USA vanno sotto il nome collettivo di **GRID**

Il problema da risolvere

- ✓ Offrire un uso *naturale* di risorse distribuite …
  - ➢ Una persona (production manager, utente) sottomette i job (programmi da esguire) a una *coda*
  - ➢ I files (output e dati) finiscono su un *file system* dove possono essere manipolati
  - ➢ Lo stato delle code e delle risorse è monitorato
- ✓ … nascondendo (per quanto possibile) il fatto che siano distribuite …
  - ➢ Ovviamente all'utente, il sistema risultante sarà più complesso
- ✓ … e ottimizzando il loro uso …
  - ➢ Rispetto al sistema equivalente non distribuito
- ✓ … con un software a <u>livello di produzione</u> …
  - ➢ Cioè basato il più possibile su standard e software esistente e solido

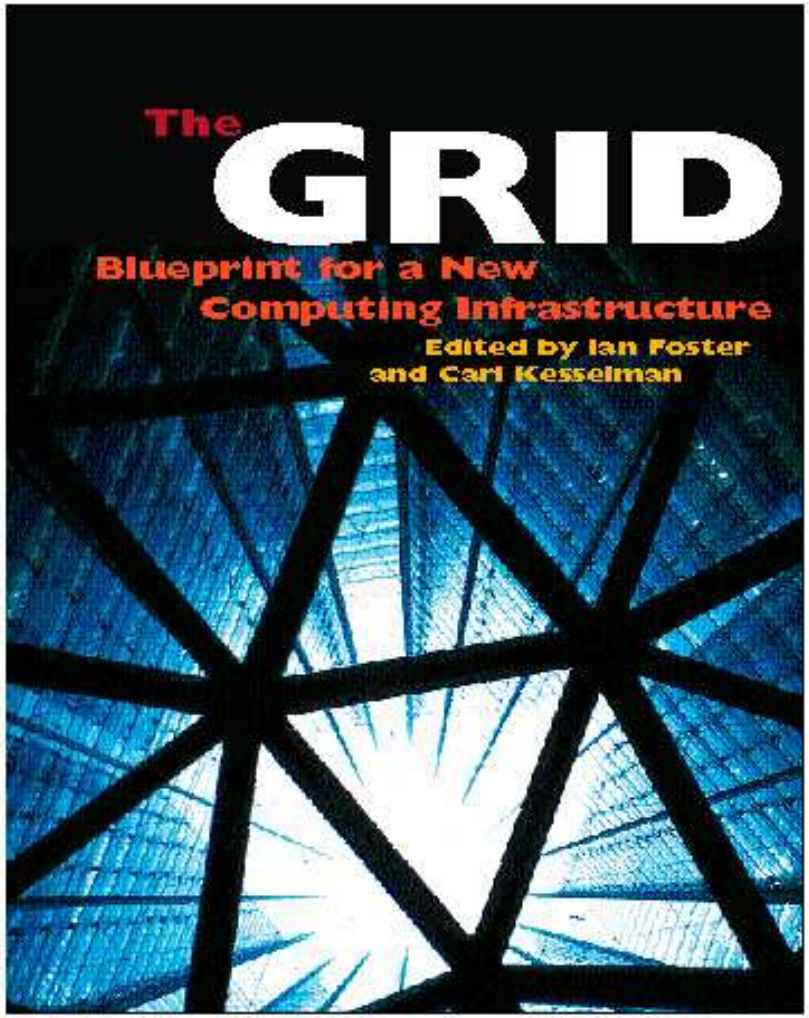
E` impossibile?

Senza griglie computazionali, si`

The Grid: Blueprint for a New Computing Infrastructure

I. Foster, C. Kesselman (Eds), Morgan Kaufmann, 1999

- ISBN 1-55860-475-8
- 22 capitoli scritti da diversi autori
- Il concetto della GRID e' introdotto da Kesselman e Foster come ***paradigma*** per l'utilizzazione di risorse distribuite
- E' stato introdotto con ***GLOBUS***, un sistema di strumenti software nato per interconnettere alcuni centri di calcolo in USA



"A source book for the history of the future" -- Vint Cerf

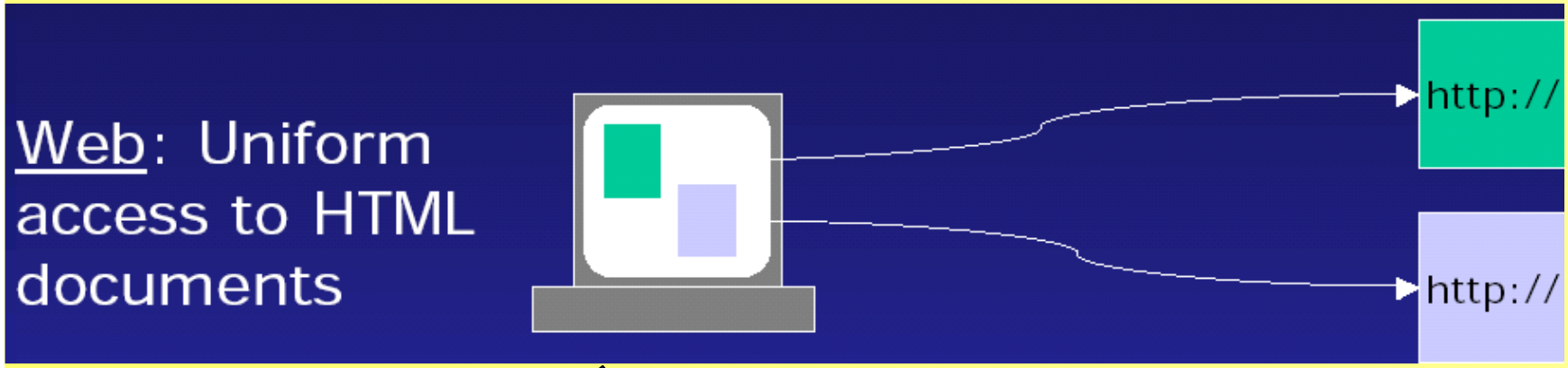
http://www.mkp.com/grids

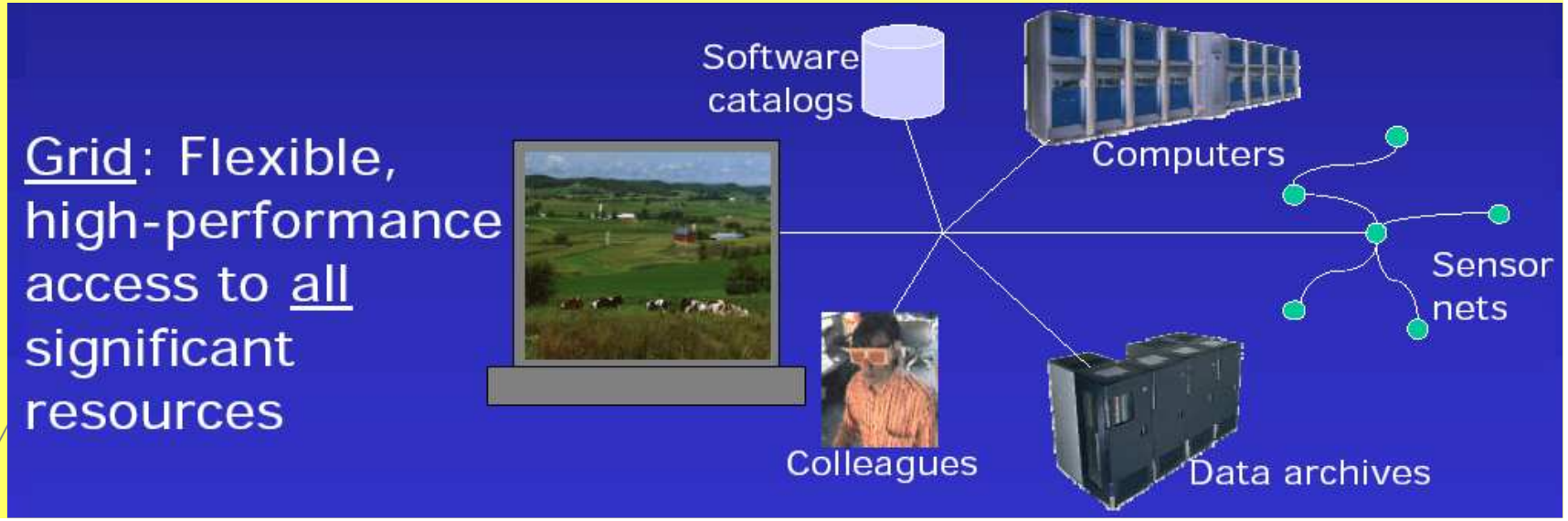
Una breve storia

- ✓ Inizio anni 90
  - Primi test con reti al Gigabit, metacomputing
- ✓ Seconda meta` degli anni 90
  - Primi esperimenti e progetti di software quali GLOBUS
  - Struttura a Tiers: MONARC
- ✓ 2001 – Oggi
  - Si e` formata una comunita` per grandi applicazioni
  - Grandi infrastrutture di calcolo sono gia` operative
  - La base tecnologica e' in crescita
  - Global Grid Forum: ~500 persone, >90 organizzazioni, 20 Paesi
- ✓ *Grid è l'insieme delle tecnologie software che consentono la condivisione di risorse e la risoluzione coordinata di problemi nell'ambito di organizzazioni virtuali multi istituzionali, dinamiche e scalabili (I. Foster)*
  - L'attenzione e' rivolta ai dati piuttosto che al calcolo numerico intensivo, come nel passato
  - C'e` un'analogia con la "griglia" della fornitura elettrica

Accesso alle risorse con Gr

Dal WEB alla GRID





La metafora della GRID

La grid puo` essere vista in analogia con la fornitura di energia elettrica: l'utente non sa i dettagli di come e dove l'energia elettrica sia prodotta e distribuita

Il middleware della grid provvede a: autenticazione, trovare disponibilita` di CPU, reperire i dati di input, archiviare i dati in uscita, gestire e sincronizzare le repliche di file.

Middleware = software che gestisce le risorse di calcolo distribuite

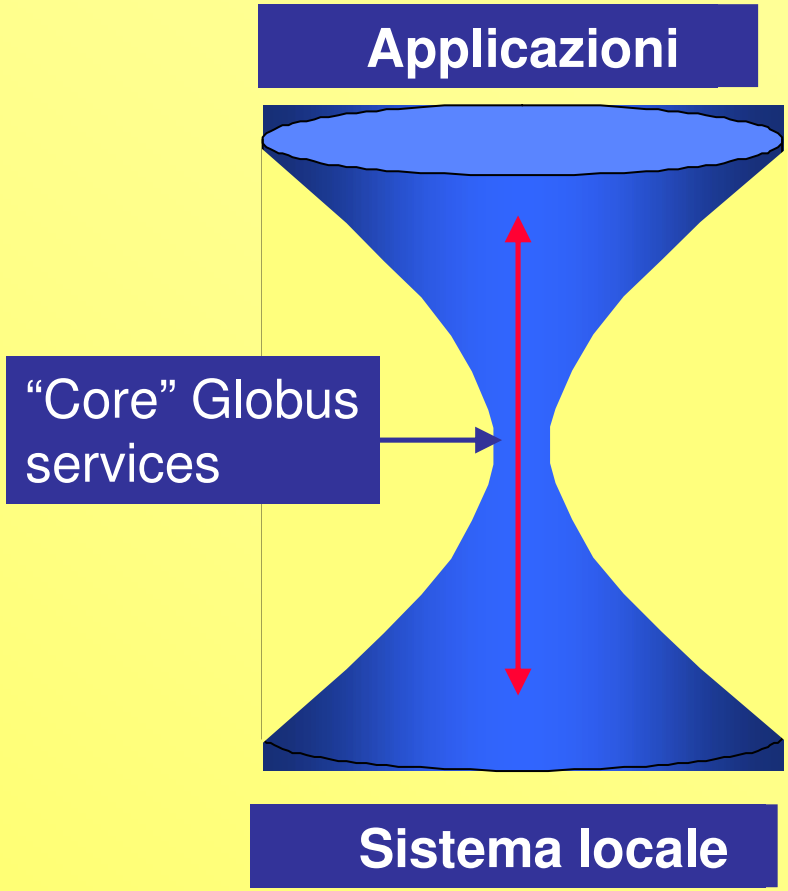
Globus: un modello per il middleware

Il modello e' focalizzato su aspetti architetturali

Alle due estremita` della "clessidra" si trovano da un lato le applicazioni degli utenti e dall'altro farm di computer.

Nel mezzo si possono individuare diversi livelli di servizi che possono essere implementati in modo generale:

- Controllo locale (a livello di "fabric")
- Connettivita` tra risorse e gestione della sicurezza
- Controllo dell'uso condiviso delle risorse
- Coordinamento di gruppi di risorse



Globus: componenti base

Fabric Layer:

- meccanismi di base:
  - start di programmi
  - accesso ai files
  - ...

Connectivity Layer:

- comunicazione via IP
- sicurezza tramite *Grid Security Infrastructure* (*GSI*)
  - basata su meccanismi a chiave pubblica (TSL/SSL, certificati X.509 & *Certification Authorities*, ecc…).
  - meccanismo uniforme per la autenticazione e autorizzazione degli utenti

Autenticazione: verifica dell'identità di un utente (o una risorsa)
Autorizzazione: accesso alle risorse ad uno specifico utente

Globus: componenti base

Resource Layer:

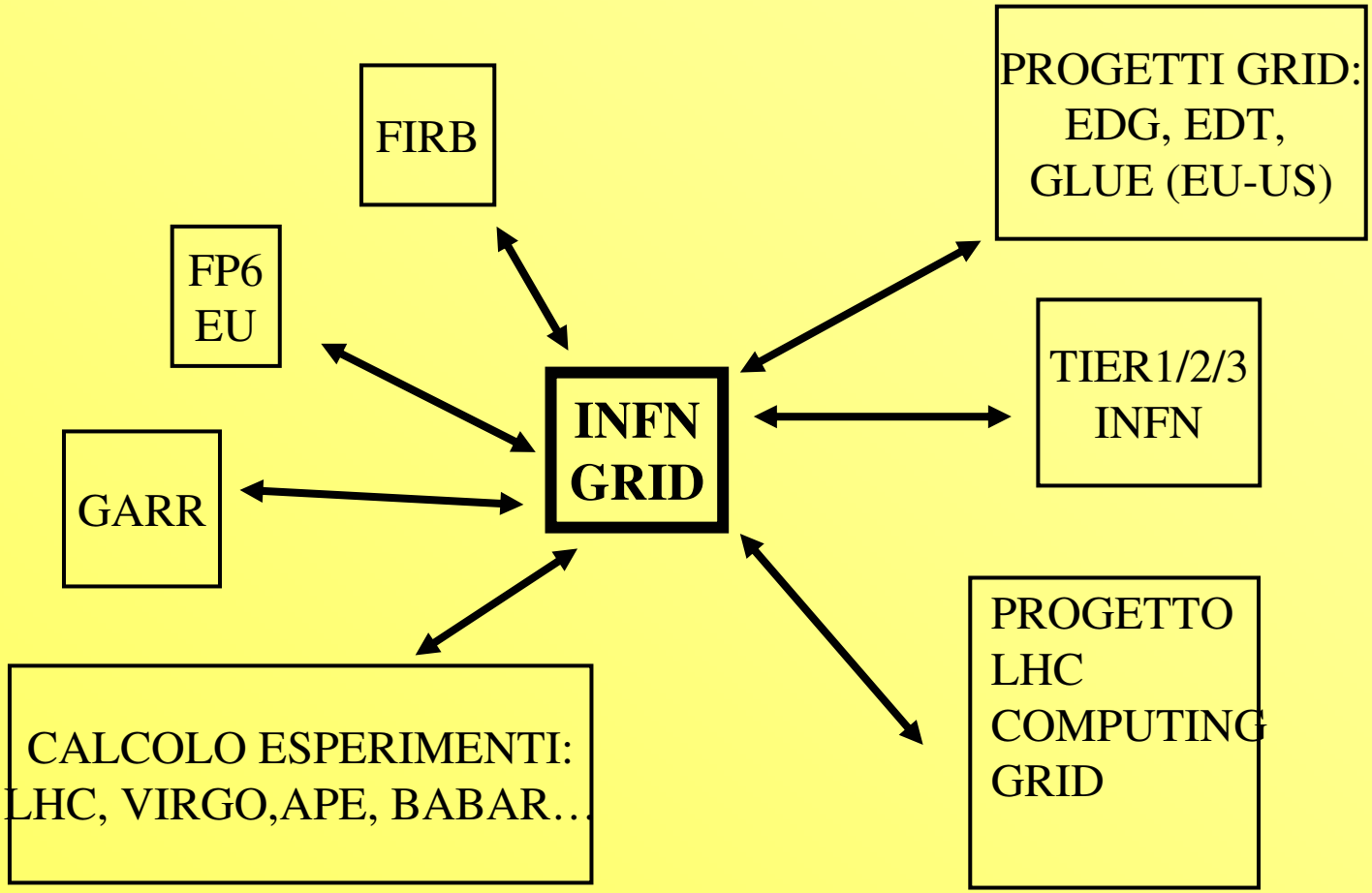
- Usa *GSI* per la sicurezza
- *Grid Resource Allocation Management* (*GRAM*) gira su ogni risorsa; garantisce il controllo remoto delle risorse
- *GridFTP* (FTP modificato) per il trasferimento files
- *Grid Resource Information Server* (*GRIS*) raccoglie e pubblica dinamicamente le informazioni sulla risorsa (ad es. memoria installata, code dponibili, ecc…)

Collective Layer: esempi:

- *Metacomputing Directory Service* (*MDS*): *GRIS* e *GIIS*
- *Grid Index Info Server* (*GIIS*) raccoglie, e pubblica informazioni selezionate da diversi *GRIS*
- Resource Broker seleziona risorse adatte per un job
- Replica Catalog e Replica Services gestiscono l'accesso e la copia dei dati distribuiti

INFN GRID & i progetti di GRID

- ✓ Nel 2000 e' nato il Progetto Speciale INFN Grid che "consiste nello sviluppo e attuazione comune di un prototipo di sistema di calcolo, basato sul middleware di GRID e finalizzato a rispondere alle esigenze degli esperimenti futuri con particolare riguardo ad LHC"
- ✓ Attraverso INFN-GRID, l'Ente partecipa a diversi progetti relativi al calcolo distribuito (EDG – European Data Grid – terminato 4 mesi fa, DataTag, EGEE - nuovo)
- ✓ Attraverso INFN-Grid si trasferiscono le tecnologie di Grid ad altre discipline: progetto FIRB (Fondo per gli Investimenti della Ricerca di Base), IG-BIGEST (Italian Grid- for Business Industry Government EScience & Technology)
- ✓ Fornisce il contributo comune INFN al progetto LCG (LHC Computing Grid)



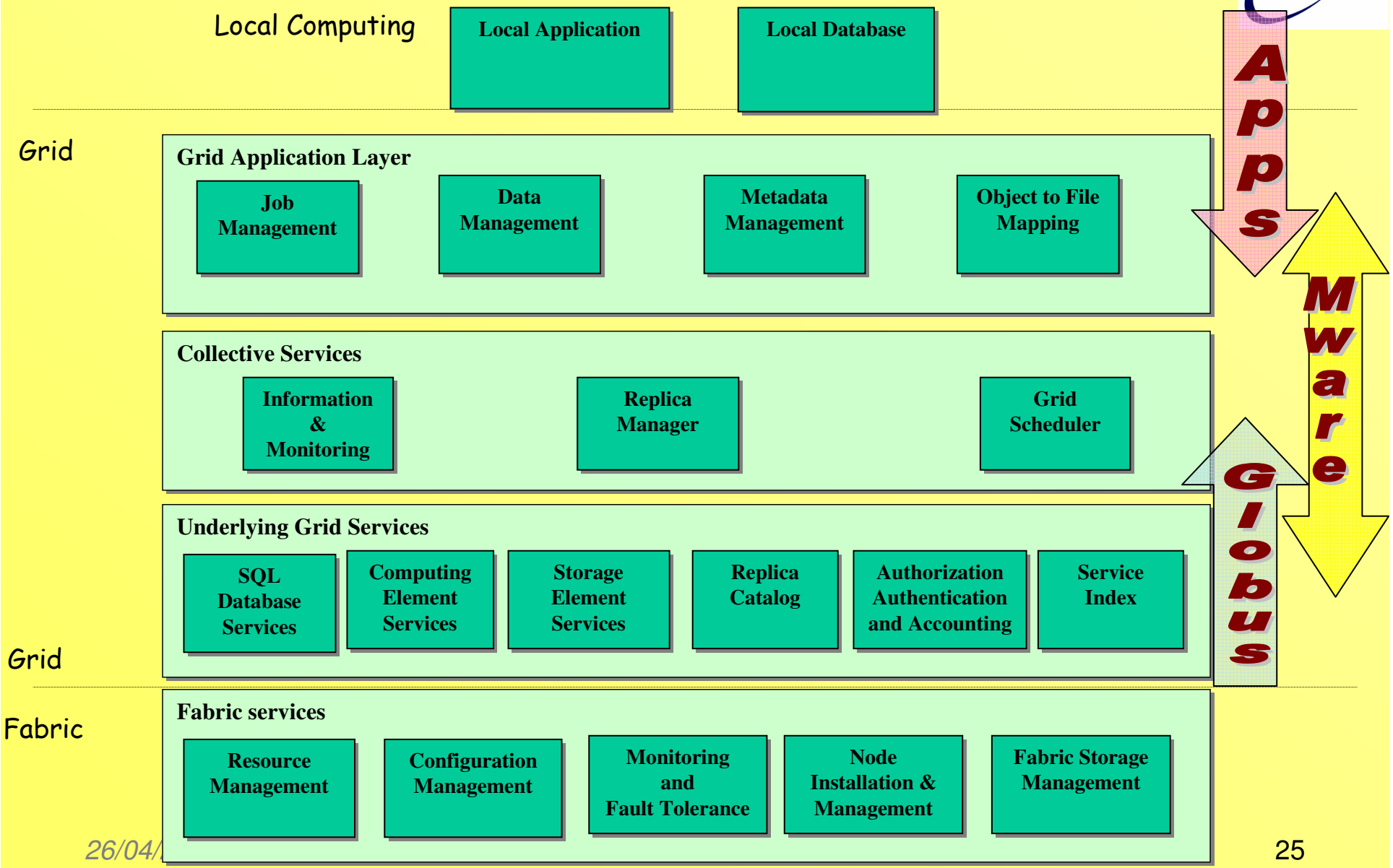
Schema dei progetti e attivita' a cui il progetto INFN Grid deve far fronte nella fase attuale.

La GRID per LHC

European Data Grid

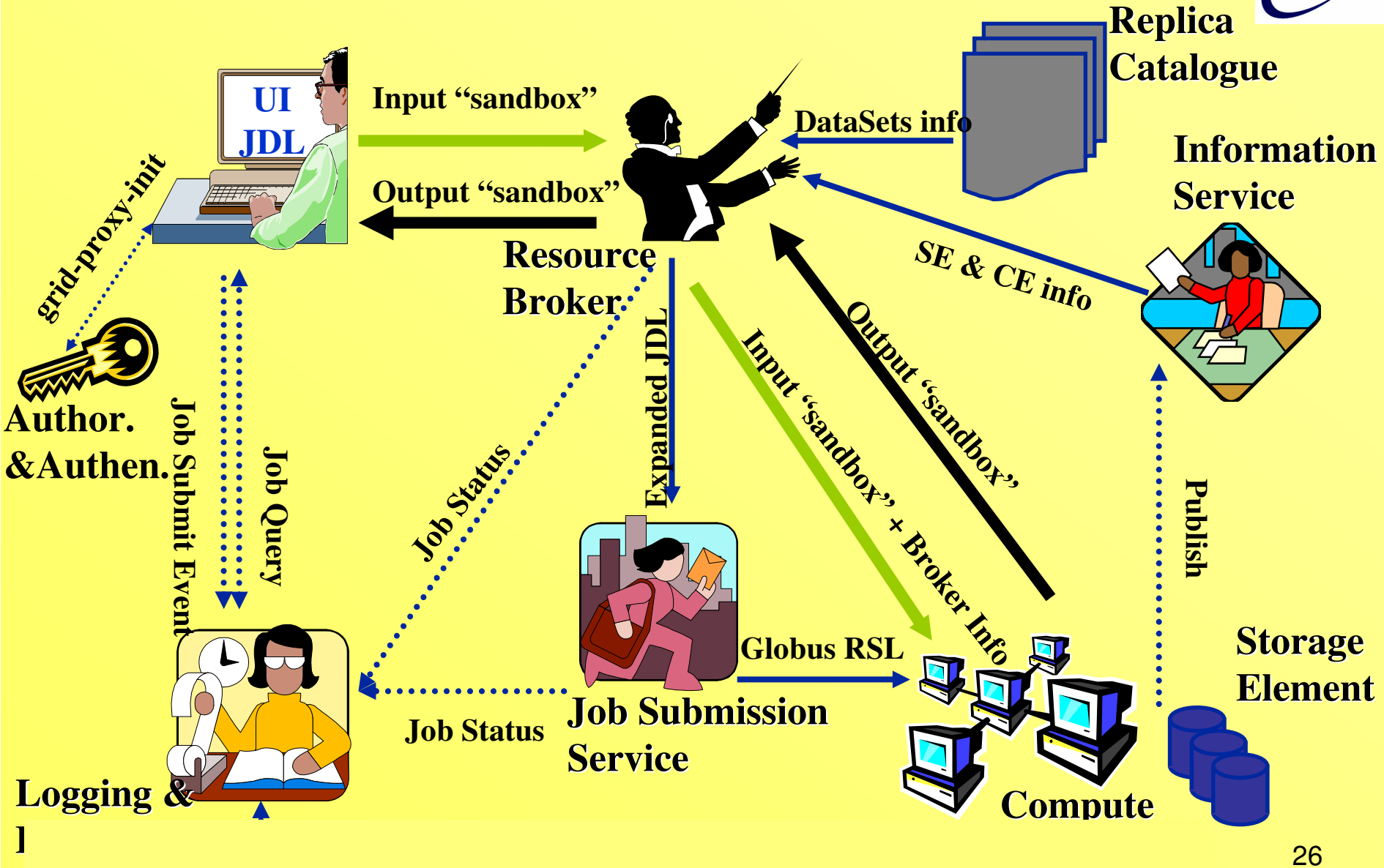
LCG

Architettura di DataGrid



Da: B.Jones

Esempio di *Job Submission*

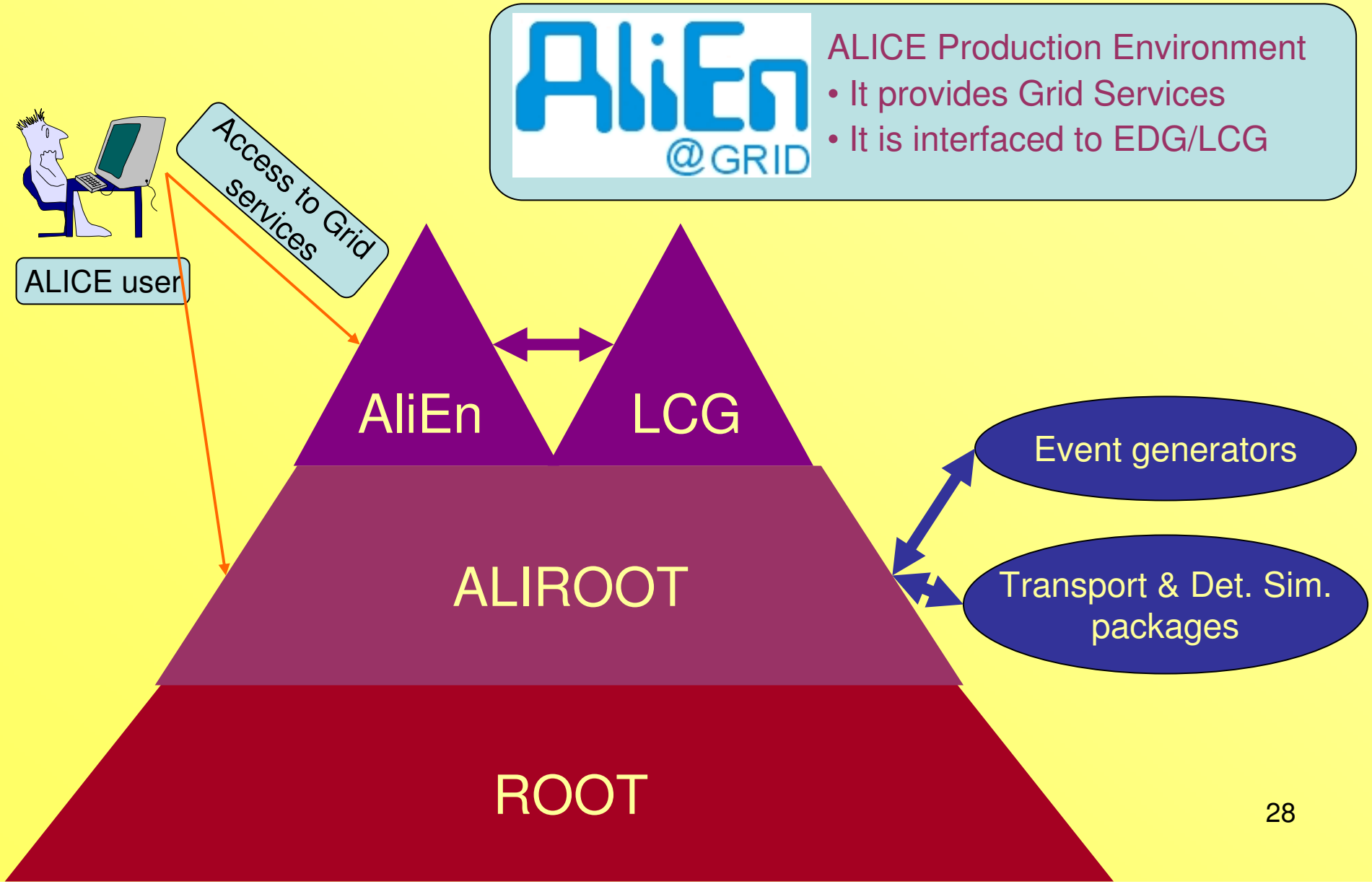


Da: B.Jones

da EDG a LCG

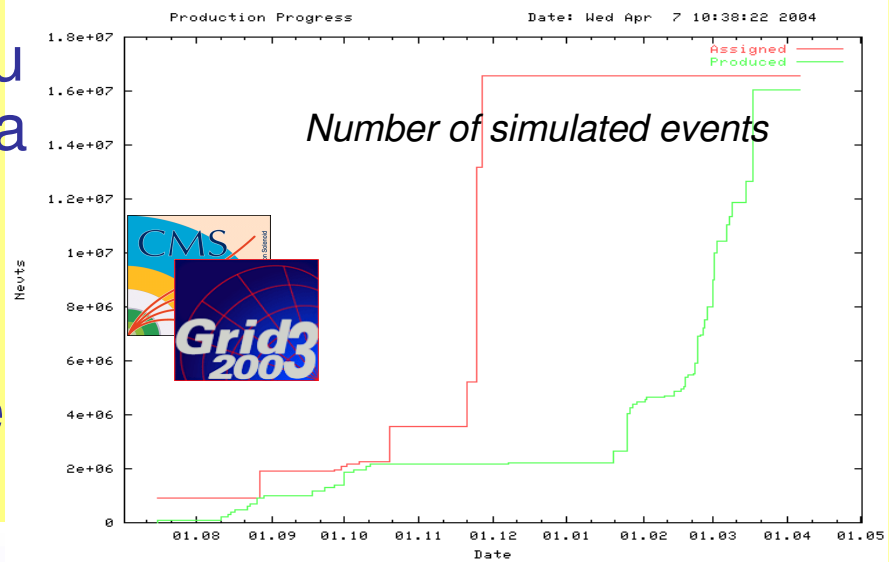
- ✓ EDG ha prodotto un sistema di middleware basato su Globus e Condor
- ✓ Ha fornito servizi per discipline diverse dalla fisica delle alte energie
- ✓ Nel 2002 e' partito il progetto LHC Computing Grid con lo scopo di supportare tutte le esigenze di calcolo di LHC.
- ✓ Per la Grid, l'ultima versione di EDG e' confluita in LCG.
- ✓ E' in fase iniziale il progetto ARDA
  - ➢ Middleware e strumenti software comuni agli esperimenti per il supporto all'analisi dei dati
  - ➢ Userà anche software dagli esperimenti, in particolare AliEn: sistema grid per il supporto all'analisi dati distribuita di ALICE, che usa direttamente middleware di basso livello (non dai "grid projects")

ALICE COMPUTING FRAMEWORK

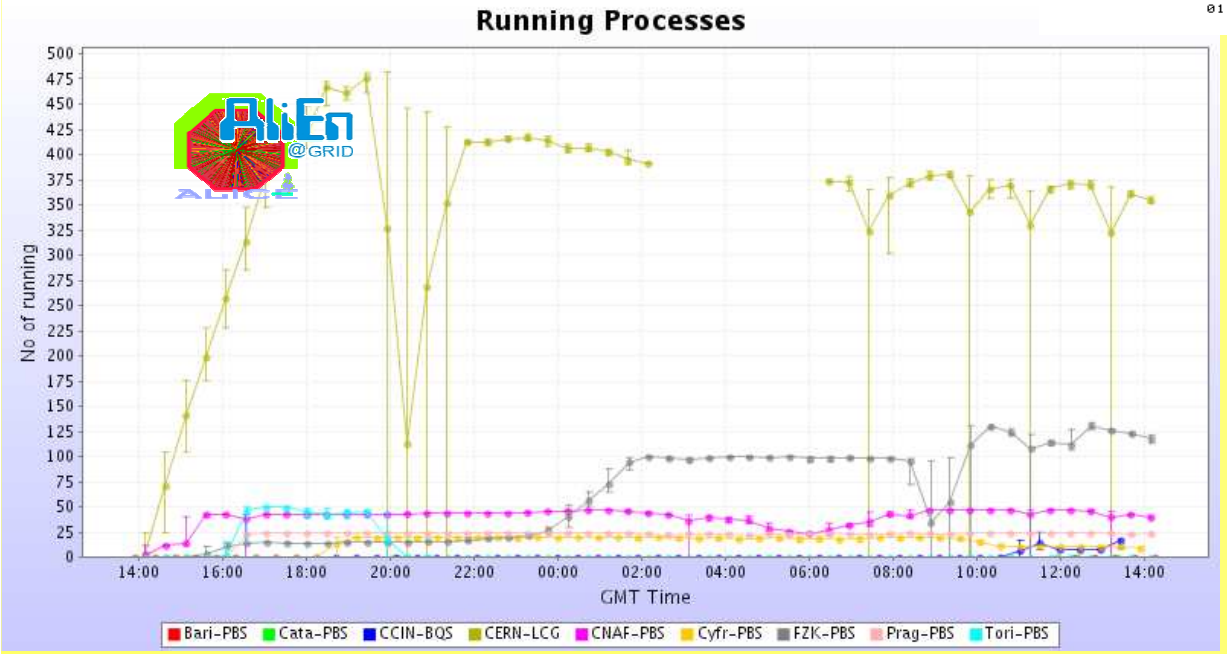


Data Challenges su Grid del 2004

Produzione dati simulati su Grid3 negli USA per il Data Challenge 2004 di CMS

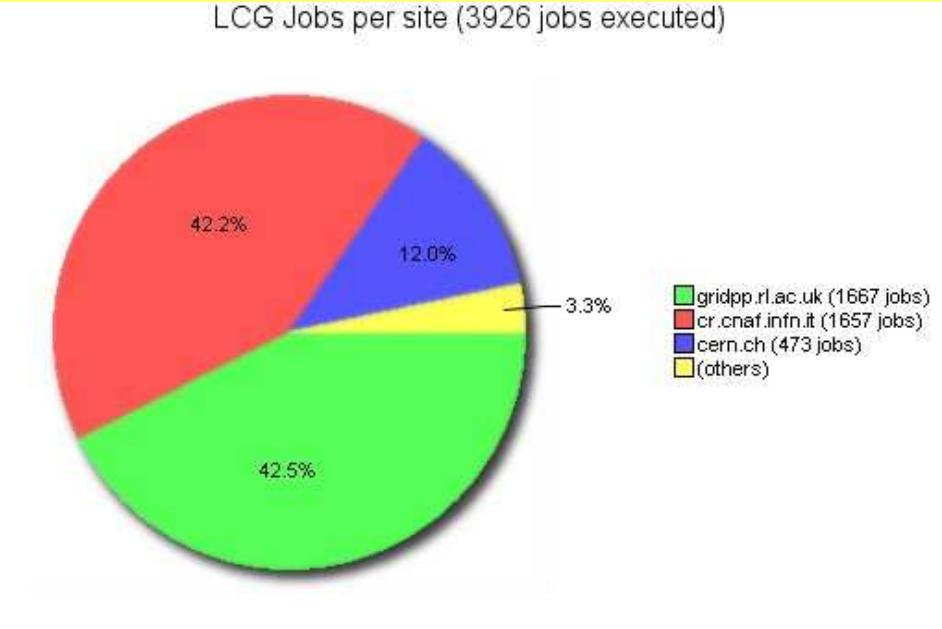


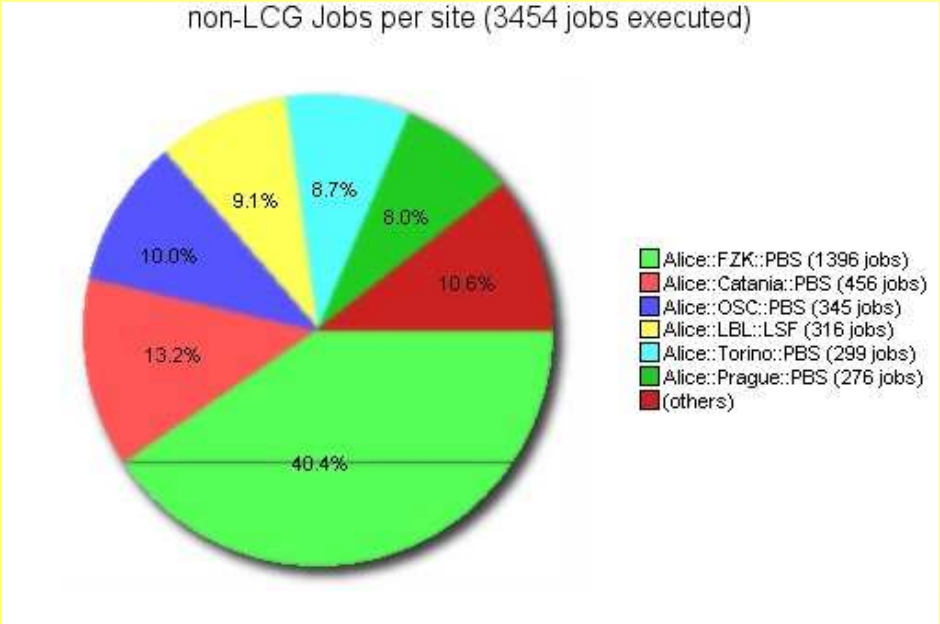
Jobs nel sistema AliEn/LCG di Alice durante il Data Challenge 2004



Data Challenge Statistics

✓ First+Second round, started on Mar 18th





Conclusioni

- ✓ Gli strumenti *grid* forniscono una grande opportunità per gestire l'accesso a dati e risorse distribuiti
- ✓ Tre anni di faticosa esperienza con i progetti *grid* hanno permesso di identificare le componenti base
- ✓ Gli esperimenti hanno costruito prototipi per i loro sistemi di calcolo anche se a volte privilegiando aspetti diversi
- ✓ Alcuni strumenti sono già "in produzione"
- ✓ La prossima sfida e` rappresentata dall'analisi di dati distribuiti