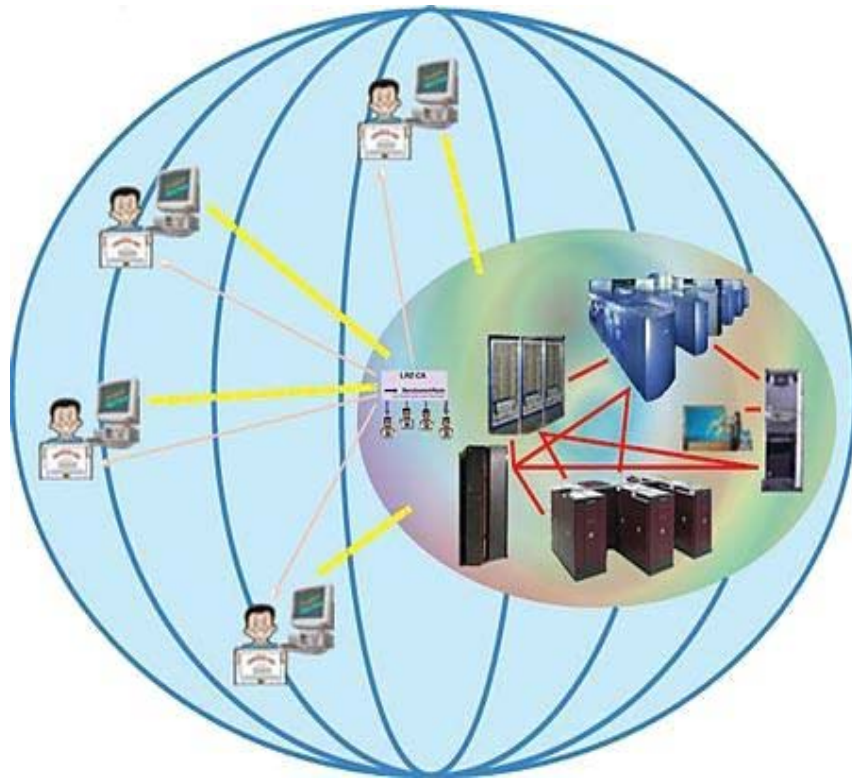


Grid Technology

kit minimo di sopravvivenza



Sommario

- introduzione
- il paesaggio internazionale
- Open Grid Services Architecture
- OGSA specification
- OGSA reference implementation: GT4
- sample GT4 service

Introduzione

Analogia con la Rete Elettrica

Rete elettrica (Power Grid)

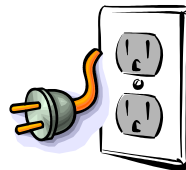
Un utente ha bisogno di energia elettrica per i propri consumi



Firma un contratto con la società che gestisce la rete



Adesso l'utente può usare l'energia elettrica per i propri consumi



La società elettrica fornisce un punto per accedere alla rete elettrica

Centrali elettriche disperse sul territorio producono energia elettrica



L'energia elettrica viene distribuita tramite opportune reti elettriche

Analogia con la Rete Elettrica

Griglia Computazionale



Definizioni

- *flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources*

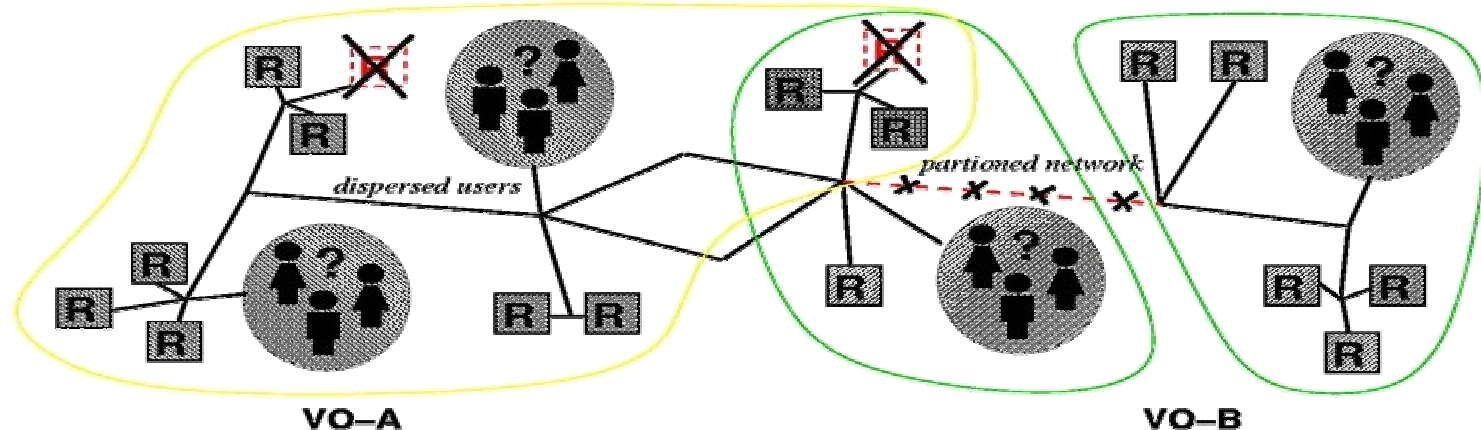
I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int'l J. Supercomputer Applications*, vol. 15, no. 3, 2001, pp. 200-222.

- *When the network is as fast as the computer's internal links, the machine disintegrates across the net into a set of special purpose appliances"*

G. Gilder, "Tele-Cosm: how infinite bandwidth will revolutionize our world" The Free Press, 2000

Organizzazioni Virtuali

- confederazioni transitorie organizzate attorno ad obiettivi comuni
- con membri, regole di appartenenza e capacita' diverse (in termini di persone, risorse computazionali, risorse dati, ecc.)
- diversa distribuzione geografica
- forte dinamicita' per adattarsi alle situazioni (carico, errori...)
- condivisione ben controllata ma richiedente una conoscenza minima delle caratteristiche fisiche delle risorse
- costruzione di capacita' di livello piu' elevato componendo capacita' esistenti (analogamente a SOA)



Risorse

- Ogni entita' che una VO mette a disposizione su una Grid e' una risorsa:
 - macchina (CPU)
 - storage
 - file
 - database
 - strumento
 - sensore
- Le risorse vengono condivise realizzando un virtual pool.
- Gli utenti non hanno conoscenza a priori dello stato, tipo, caratteristiche delle risorse del pool.
- Gli utenti possono avere accesso temporaneo a risorse che non potrebbero accedere direttamente (ad ex. no login) altrimenti
- E' necessario astrarre a livello di virtual pool Utenti (Global Identity) e Risorse (Servizi), provvedendo un mapping opportuno al momento dell'utilizzo.

To be or not to be (a Grid)

La checklist di Ian Foster

Una Grid...

- coordina risorse che non sono soggette a controllo centralizzato
- usa protocolli ed interfacce standard, aperti, generici
- realizza elevata qualita' di servizio

assumendo quindi che:

- non esista un "sito centrale"
- non si abbia la completa conoscenza
- non esistano relazioni affidabili

Per cui:

Jxta non e' una Grid, Condor non e' una Grid.....

Grid: la storia in breve

- **1st Generation Grids**

- "Metacomputers" locali con servizi di base quali file systems distribuiti e single sign-on su tutto il sito
- "Metacenters" esploranti le problematiche di integrazione fra organizzazioni diverse
- Integralmente custom, top to bottom, in pratica delle proofs of concept.

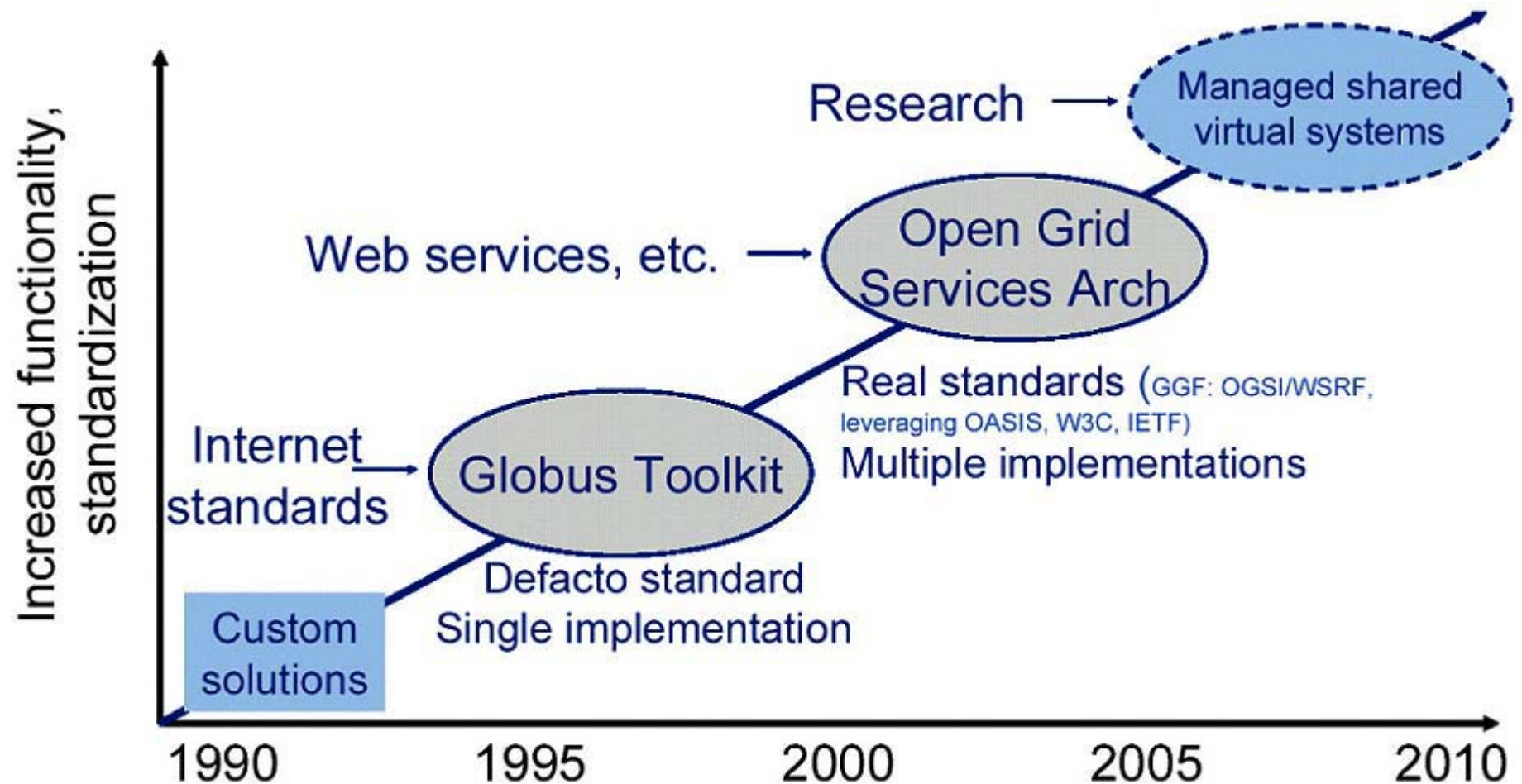
- **2nd Generation Grids**

- utilizzano servizi software e protocolli di comunicazione sviluppati da progetti grid: *Condor*, *Globus*, *Unicore*, etc.
- richiedono pesante customizzazione per il raggiungimento di una soluzione completa
- l'interoperabilita' e' molto difficile

- **3rd Generation Grids**

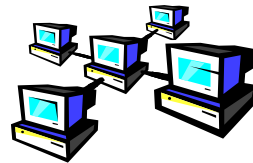
- un insieme di *common interface specifications* supporta l'interoperabilita' di servizi discreti, sviluppati indipendentemente
- coesistono competizione ed interoperabilita' non solo fra applicazioni e toolkits, ma anche fra implementazioni di servizi chiave.

L'evoluzione vista da lontano

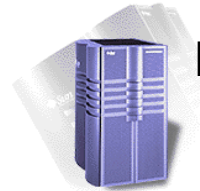


Perche' Grids

- Perché la quantità di risorse di calcolo necessarie alle applicazioni cresce a dismisura
- Perché l'ammontare dei dati richiede sistemi di immagazzinamento sempre più complessi
- Per rendere più semplice la collaborazione tra persone e risorse appartenenti a diverse organizzazioni
- Per accedere a strumentazioni che non sono facilmente accessibili in altro modo
- Perché è la naturale evoluzione della computazione distribuita



Migliaia di CPU che lavorano per ottenere un risultato comune



Da centinaia di Gigabytes a Petabytes (milioni di gigabytes) di dati da immagazzinare



Persone di organizzazioni diverse che lavorano per raggiungere i medesimi scopi



Perché non possono essere spostate, replicate o i costi per far ciò sono troppo alti



Per creare un nuovo mercato di servizi di calcolo su Internet

I requisiti che soddisfa una Grid (I)

1. interoperabilita' e supporto per ambienti dinamici ed eterogenei

- virtualizzazione delle risorse
- modalita' di gestione comuni
- discovery e query di risorse
- protocolli e schemi standard

2. condivisione di risorse fra organizzazioni distinte

- name space globale
- servizi per metadati
- autonomia dei siti
- dati di utilizzo delle risorse

3. ottimizzazione

- ottimizzazione "demand-side"

4. qualita' del servizio

- service level agreement
- service level attainment
- migrazione

I requisiti che soddisfa una Grid (II)

5. esecuzione di JOB

- supporto per diversi tipi di job
- job management
- scheduling
- provisioning di risorse

6. servizi dati

- specifica e gestione di policy
- data storage
- data access
- data transfer
- data location management
- data update
- data persistency
- data federation

I requisiti che soddisfa una Grid (III)

7. sicurezza

- autenticazione ed autorizzazione
- infrastrutture di sicurezza multiple
- soluzioni di sicurezza perimetrale
- isolamento
- delega
- scambio di policy
- intrusion detection, protezione, logging sicuro

8. riduzione dei costi amministrativi

- gestione basata su policy
- gestione delle informazioni sulle applicazioni
- determinazione dei problemi

9. scalabilita'

10. affidabilita'

11. facilita' d'uso ed estensibilita'

Il paesaggio internazionale

Grid Actors: enti di standardizzazione



- Grid technologies



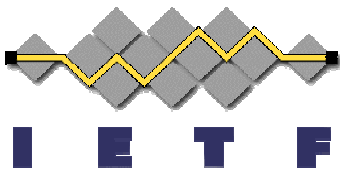
- Web Service technologies



- Internet base technologies (SOAP, WSDL)



- Common Resource Models



- Internet software and hardware technologies (TCP/IP, SNMP)

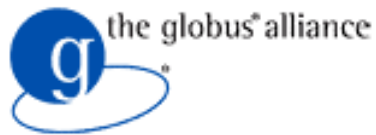


- Storage device Technologies (NAS & SAN)

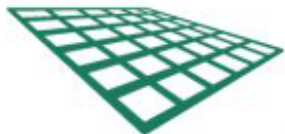


- Digital Identity

Grid Actors: organizzazioni



Enterprise
Grid Alliance



■

■

■

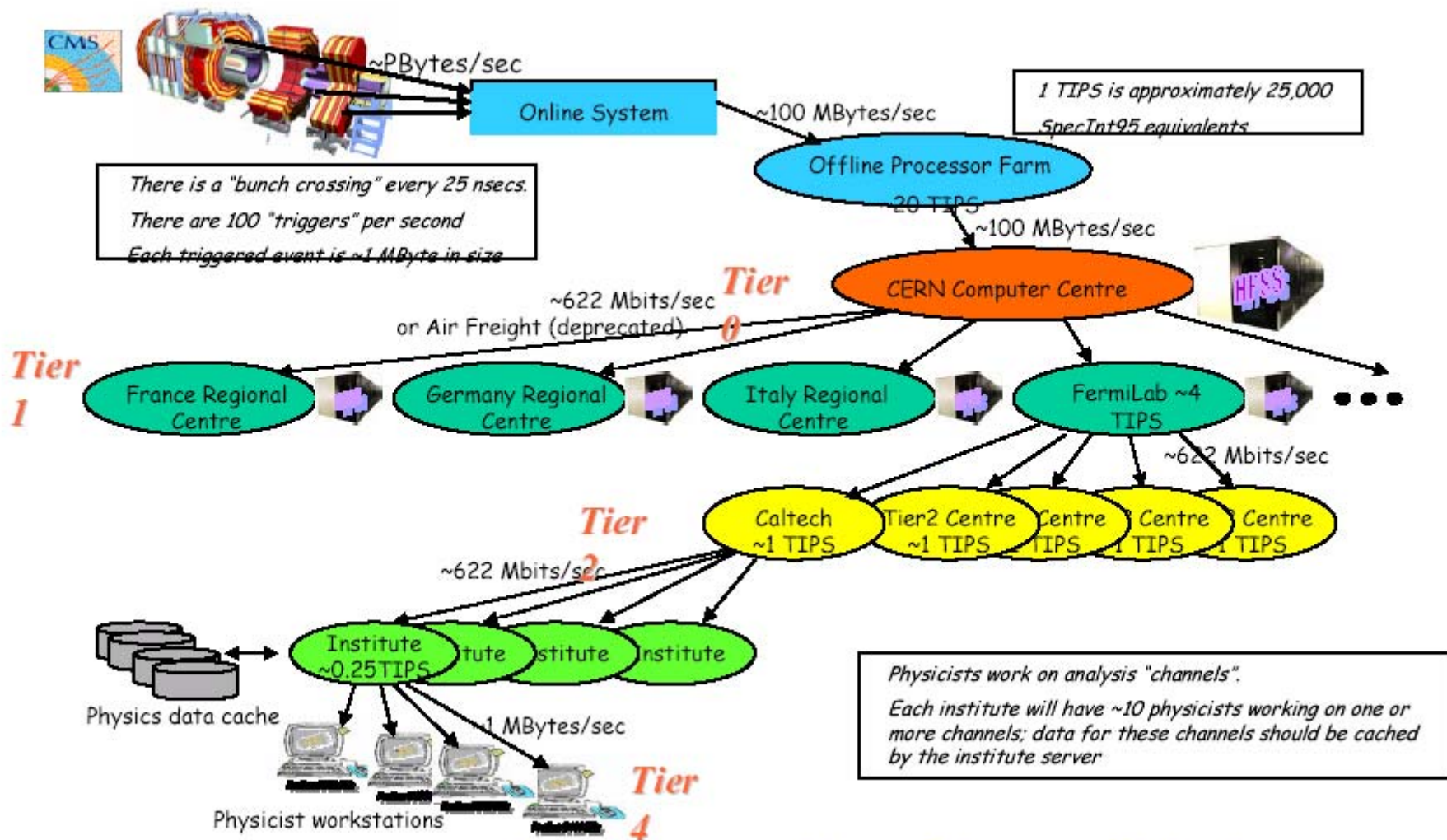
■

Grid Actors: industria

- Hewlett-Packard
- IBM
- Intel
- Microsoft
- Silicon Graphics, Inc.
- AMD
- Computer Associates
- EMC
- Fujitsu
- Hitachi Data Systems
- KISTI
- Oracle
- Platform Computing
- Shell Exploration
- Sun Microsystems
- Altair Engineering
- AT&T
- Cassatt Corporation
- CIENA Corporation
- Cisco Systems* CoreGRID
- Dell
- Force10 Networks
- Foundry Networks
- Hitachi, Ltd.
- Informatica
- NEC
- Novartis

Non Grid ma Grids...

Data Grid per la fisica delle alte energie (courtesy Harvey Newman, Caltech)



Non Grid ma Grids...

Home Computers per l'analisi di farmaci contro l'AIDS

- migliaia di utenti
- Philanthropic computing vendor (Entropia)
- gruppo di ricerca (Scripps)
- obiettivo comune: avanzamento nella ricerca e studio dell'AIDS



fightAIDS@home the Olson laboratory at The Scripps Research Institute
computing toward a cure

powered by entropia

Free Software for Your PC - By [downloading Entropia](#) onto your PC, **FightAIDS@Home** uses your computer's idle resources to accelerate powerful new anti-HIV drug design research!

FightAIDS@Home is a computational research project conducted by the [Olson laboratory](#) at [The Scripps Research Institute](#) in La Jolla, California. The project uses Entropia's global Internet computing grid, which runs both commercial and research applications on PCs.

How Your PC Helps - FightAIDS@Home uses your computer to generate and test millions of candidate drug compounds against detailed models of evolving HIV viruses, a feat previously impossible without dozens of multi-million dollar supercomputers. Every PC matters!

September 22, 2000

Download

Getting started is easy - [download and install](#) Entropia's free software now!

Get Project News via E-mail

Enter your email address below to receive **FightAIDS@Home** news and announcements!

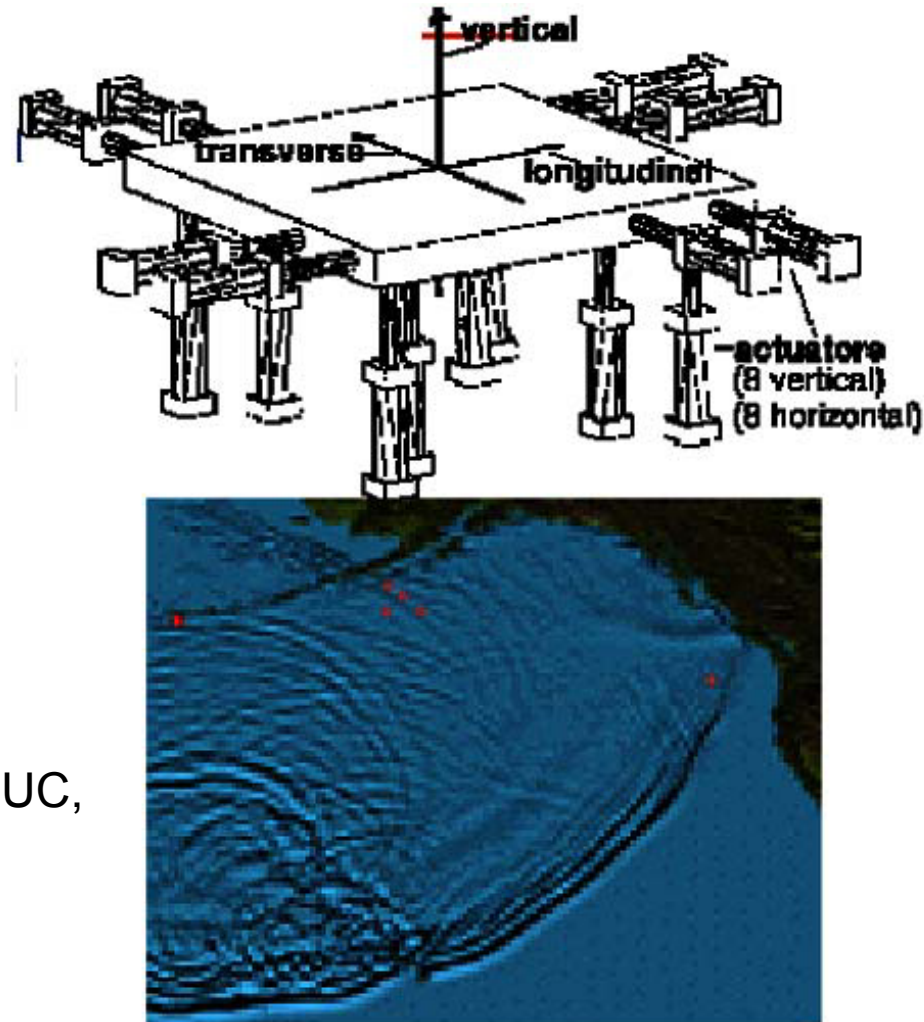
submit

Non Grid ma Grids...

Rete per la simulazione di terremoti

- NEESgrid: infrastruttura nazionale per la collaborazione tra ingegneri civili e l'uso di sensori, databases, computers e altro
- accesso on-demand ad esperimenti, dati, calcolo, archivi, collaborazioni

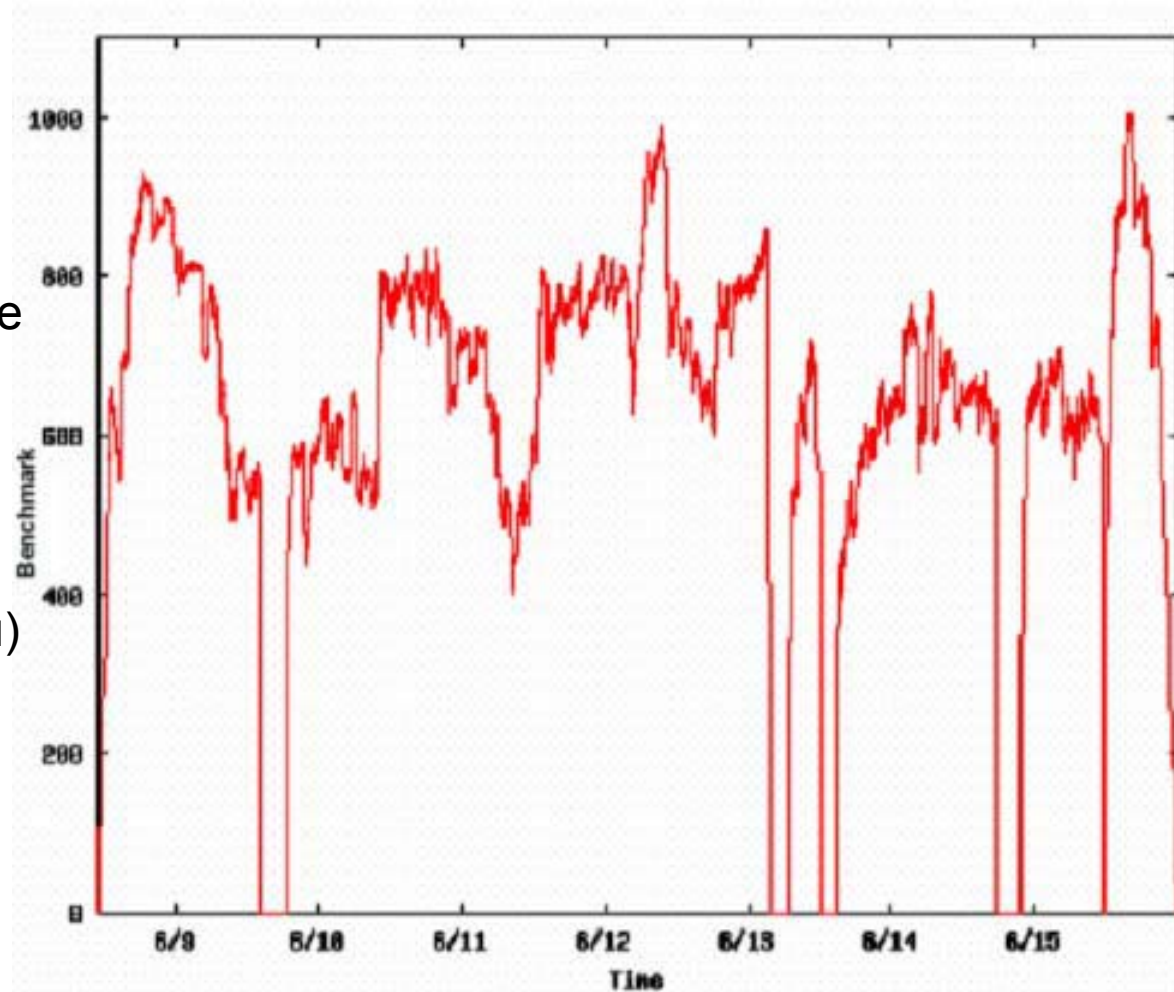
NEESgrid: Argonne, Michigan, NCSA, UIUC, USC



Non Grid ma Grids...

Risoluzione del NUG30

- ricerca della soluzione del problema dell'assegnamento quadratico NUG30
- collaborazione informale di matematici e informatici
- Condor-G ha gestito $3.46E8$ secondi di CPU in 7 gg. (peak 1009 cpu) in USA e Italia

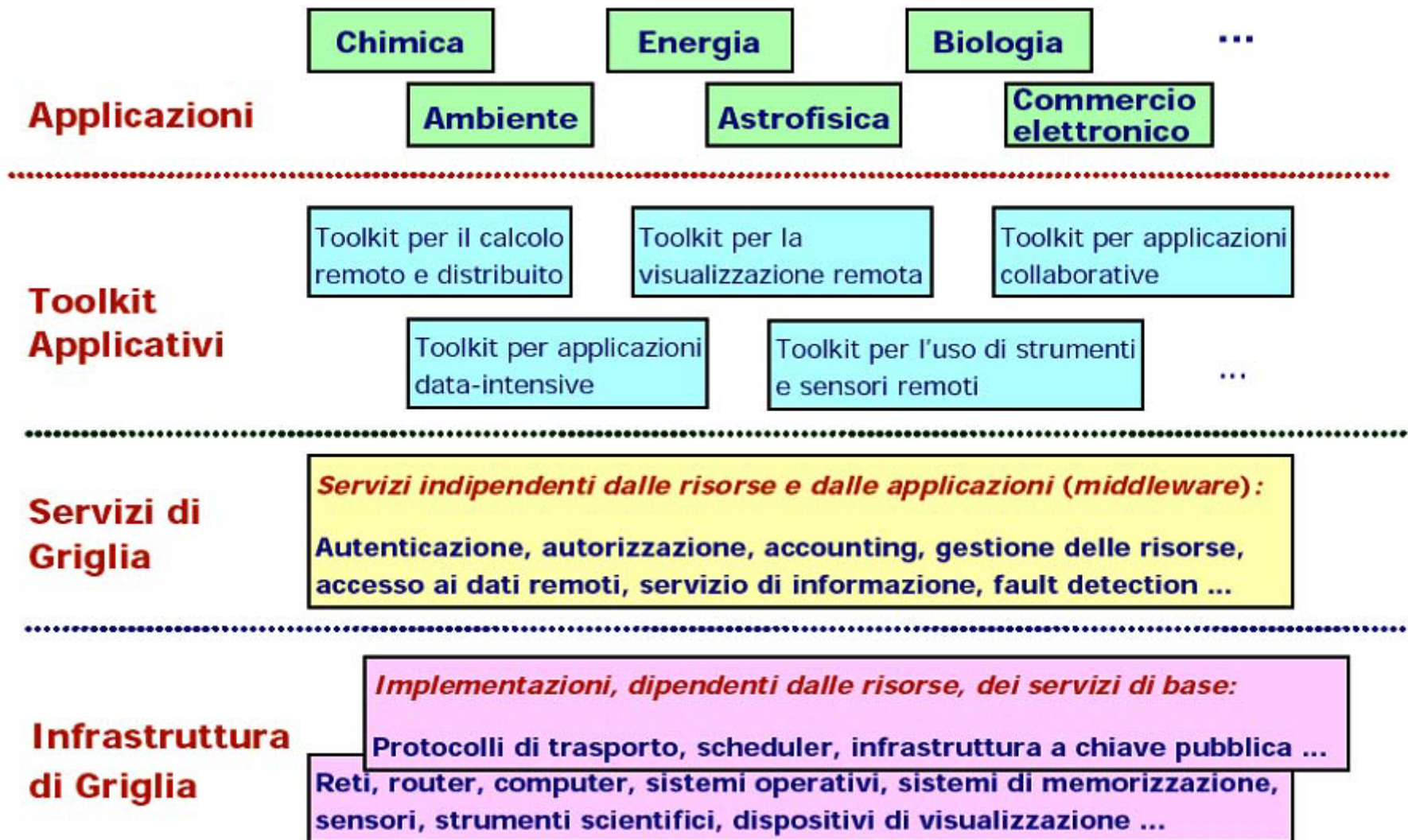


Grids & piattaforme

- **distributed supercomputing:** risorse di calcolo aggregate per applicazioni ad alta complessita' computazionale
- **high throughput computing:** uso dei cicli inutilizzati di processori per ottenere lavoro utile
- **on-demand computing:** soddisfare le necessita' a breve termine di risorse di cui non e' conveniente disporre localmente
- **data-intensive computing:** sintetizzare nuova informazione da dati mantenuti in database geograficamente distribuiti
- **collaborative computing:** abilitazione ed intensificazione delle interazioni umane

- **TeraGrid**
- **EGEE/GLite**
- **Unicore**
- **Globus (GT4)**
-

Architettura Generale



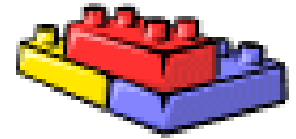
Open Grid Services Architecture

- OGSA specification

OGSA: perche' e cosa vuol essere

- per comprendere il panorama di piattaforme e modelli di Grid
- per sviluppare il framework architetturale in cui inserire gli standards per Service Oriented Grids
- per fornire indicazioni e motivazioni per l'avvio di nuove attivita' di standardizzazione per realizzare Grid veramente interoperabili
 - valutando l'applicabilita' di standards esistenti in ambito Grid
- e' una Open Service-Oriented Architecture:
 - le risorse sono le entita' fondamentali
 - creazione e distruzione dinamica di servizi/risorse
- prevede un incapsulamento *coarse-grained*
 - gli elementi dell'architettura sono *pluggabili*
- e' customizzabile
 - supporto per contenuto dinamico, specifico del dominio applicativo,....
 - entro il medesimo framework standardizzato
- costruita su un'infrastruttura basata su Web Services
- e' la **flagship architecture** del GGF e il **blueprint** per la realizzazione di piattaforme di grid computing basate su standards

Perche' usare SOA?



1. vista logica

- fornisce una visione logica e astratta delle funzionalita' richieste

2. funzionalita' riutilizzabili *coarse grained*

- incapsula comportamenti complessi all'interno di servizi (FT, parallelismo)
- composizione di servizi per costruire nuovi comportamenti

3. estensibilita'

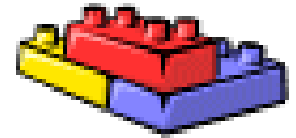
- fornisce il framework ed i costrutti naturali per l'estensibilita'

4. neutralita' rispetto alle piattaforme

- interazioni e funzionalita' costruite in nodo standardizzato e *platform agnostic*

Perche' usare i Web Services?

- forte supporto da parte dell'industria
- massimizzazione del valore del tempo
 - possibilita' di sfruttare robusti tools di sviluppo presenti per i Web Services
 - riduzione della curva di apprendimento e della frammentazione dell'implementazione
- abbracciare ed estendere
 - approfittare dello sforzo nello sviluppo e guida del consenso sugli standards
 - focalizzare risorse limitate in estendere/aggiungere standards in aree ove e' necessario



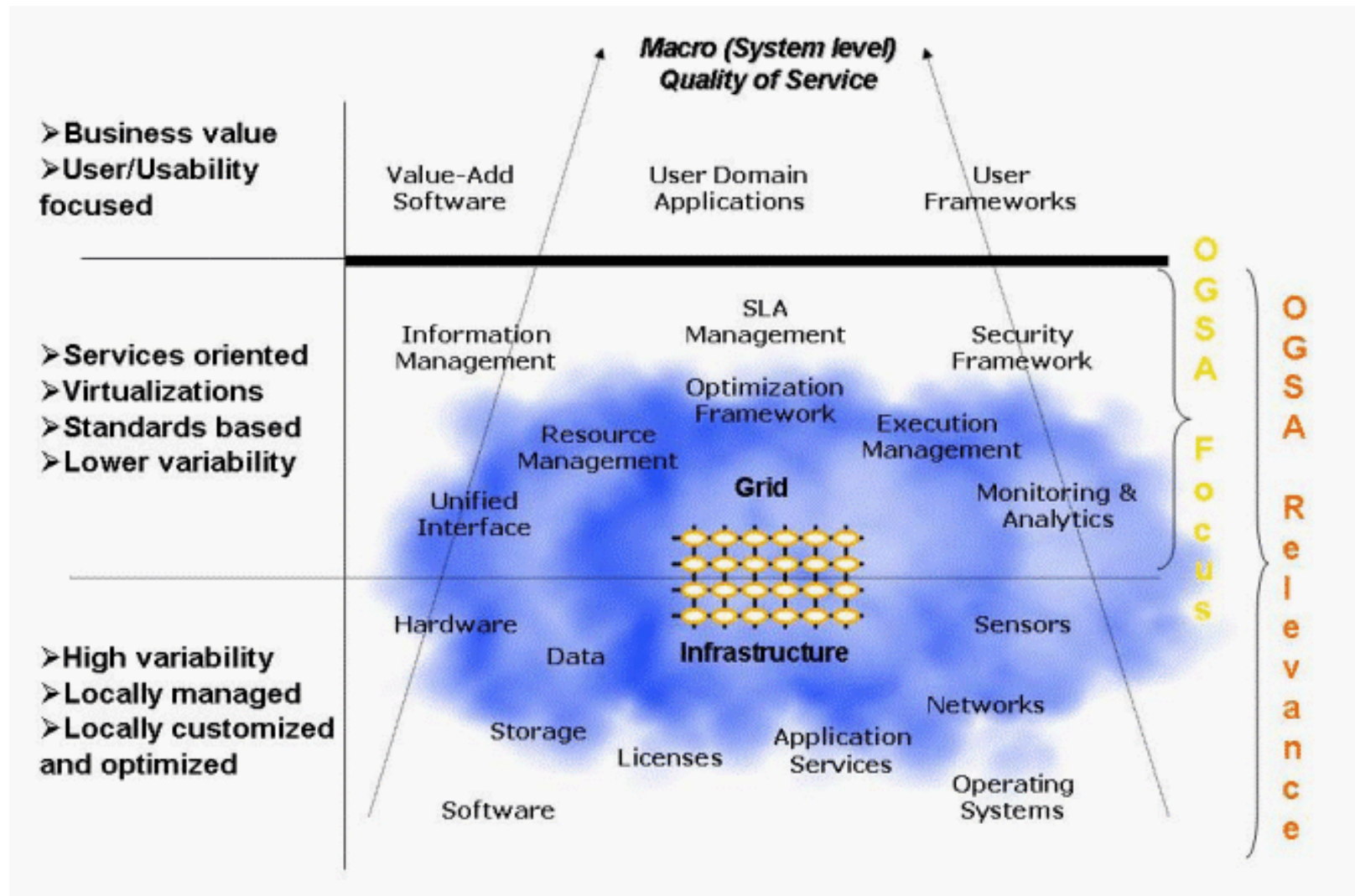
OGSA: motivazioni in dettaglio

- Obiettivi
 - soluzioni interoperabili per applicazioni Grid based
 - utilizzando e sviluppando tecnologie per loosely coupled distributed computing
- Approccio
 - Standardizzazione a livello architetturale
 - Simile al profiling.
 - sviluppata prima e/o durante lo sviluppo degli standards
 - Uso di standards e tecnologie esistenti ove possibile
 - Usa gap analysis guidata da casi d'uso
 - Gapssono riempiti proattivamente
 - Non escusivamente in ambito GGF
- Filosofia
 - Non si puo' prevedere tutto – estensibilita'
 - Separazione di policies e comportamenti
- Valorizzare il contributo dell'industria ove possibile
- Progettare per ottenere “on the wire interoperability”
- Incoraggiare la convergenza facendo interagire le comunita'
- Promuovere la discussione aperta e la sperimentazione

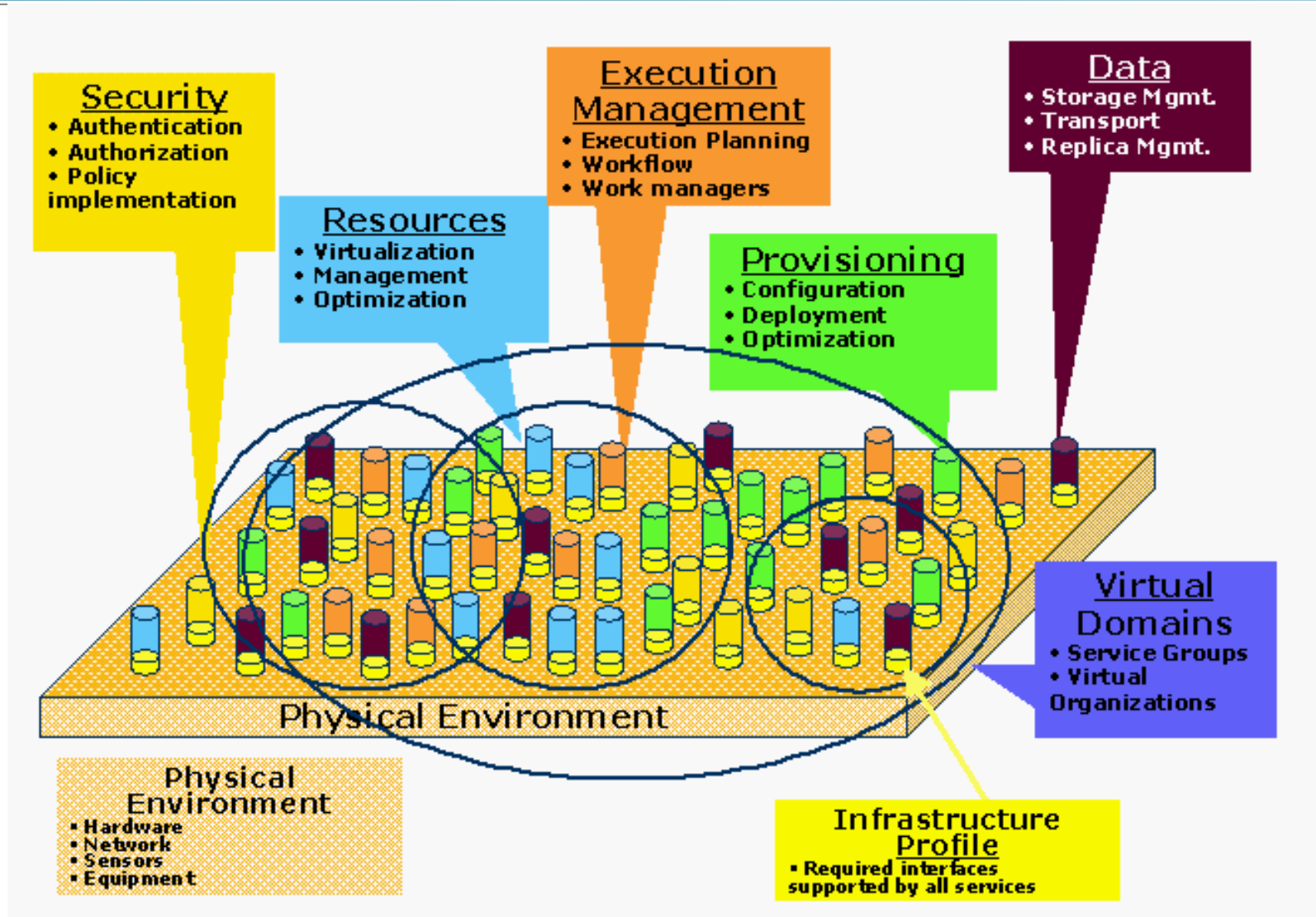
OGSA specification v. 1.5.008 (10/3/2006)

- definisce un insieme di *core capabilities* e comportamenti
- analisi basata su requisiti, problematiche tecniche, casi d'uso, esperienze precedenti, stato dell'arte in settori collegati
- e' basata su, e contribuisce a, le specifiche tecniche definenti l'architettura dei Web Services
- non e' stratificata e non e' strettamente object oriented
- i servizi sono dei *pari* lascamente accoppiati che, sia singolarmente o come parte di un gruppo di servizi, realizzano le capabilities di OGSA attraverso implementazione, composizione, interazione con altri servizi
- presuppone l'esistenza di un core-set di interfacce, standard, e comportamenti che i servizi devono implementare per partecipare ad una Grid OGSA. Questo insieme di implementazioni e manifestazioni comuni in supporto di OGSA e' definito *infrastructure services* o *Grid fabric*

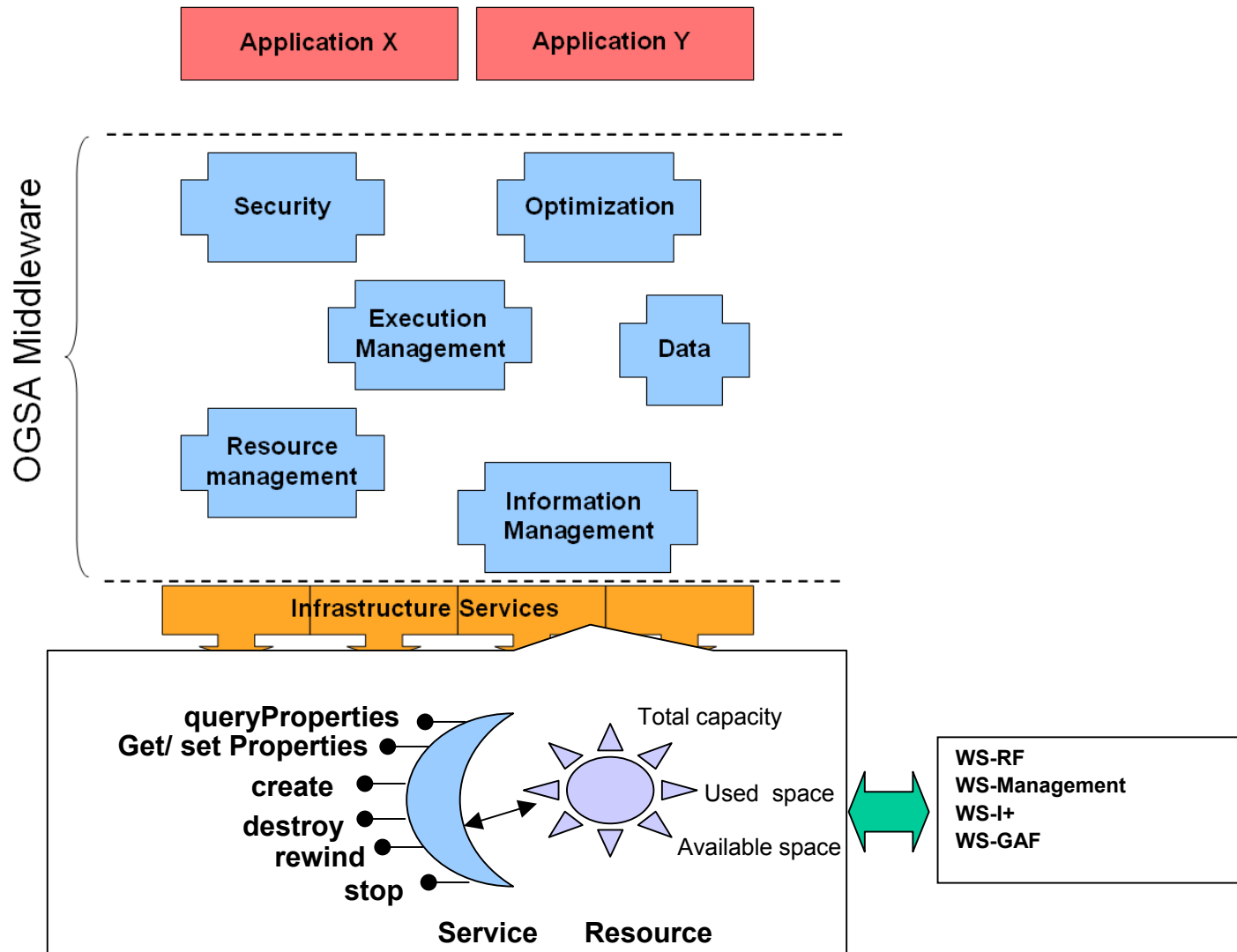
OGSA: non solo strati (I)



OGSA: non solo strati (II)



OGSA e WS-RF



Infrastructure Services

- **WS foundation**
 - i sistemi e le applicazioni in OGSA sono strutturati secondo i principi di SOA, e le interfacce dei servizi sono descritte in WSDL
 - XML e' la lingua franca per descrizioni e rappresentazioni
 - SOAP e' il formato di scambio messaggi primario
 - le definizioni sono consistenti con WS-I
- **Naming:**
 - human-oriented
 - abstract (ad ex. WS-Naming/Legion)
 - address (EPR, memory, IP/Port)
- **Naming Policy:** WS-Addressing
- **Security Services:** WS-Security, WS-Agreement, SAML, XACML
- **End to end security:** XML encryption,, digital signatures, TLS, IPsec
- **State:** WS-RF e WS-Management
- **Notification:** WS-Notification, WS-Eventing
- **Transaction:** WS-TX, WS Composite Application Framework
- **Orchestration:** per choreografy, orchestration,workflow, vedere WS
- **Interoperability:** OGSA Profiles

Execution Management Services

Obiettivo: istanziare e gestire unita' di lavoro (OGSA o legacy) dette JOB

Problemi (alcuni)

- trovare le locazioni candidate per l'esecuzione
- scegliere la locazione per l'esecuzione effettiva
- preparare per l'esecuzione
- iniziare l'esecuzione
- gestire l'esecuzione

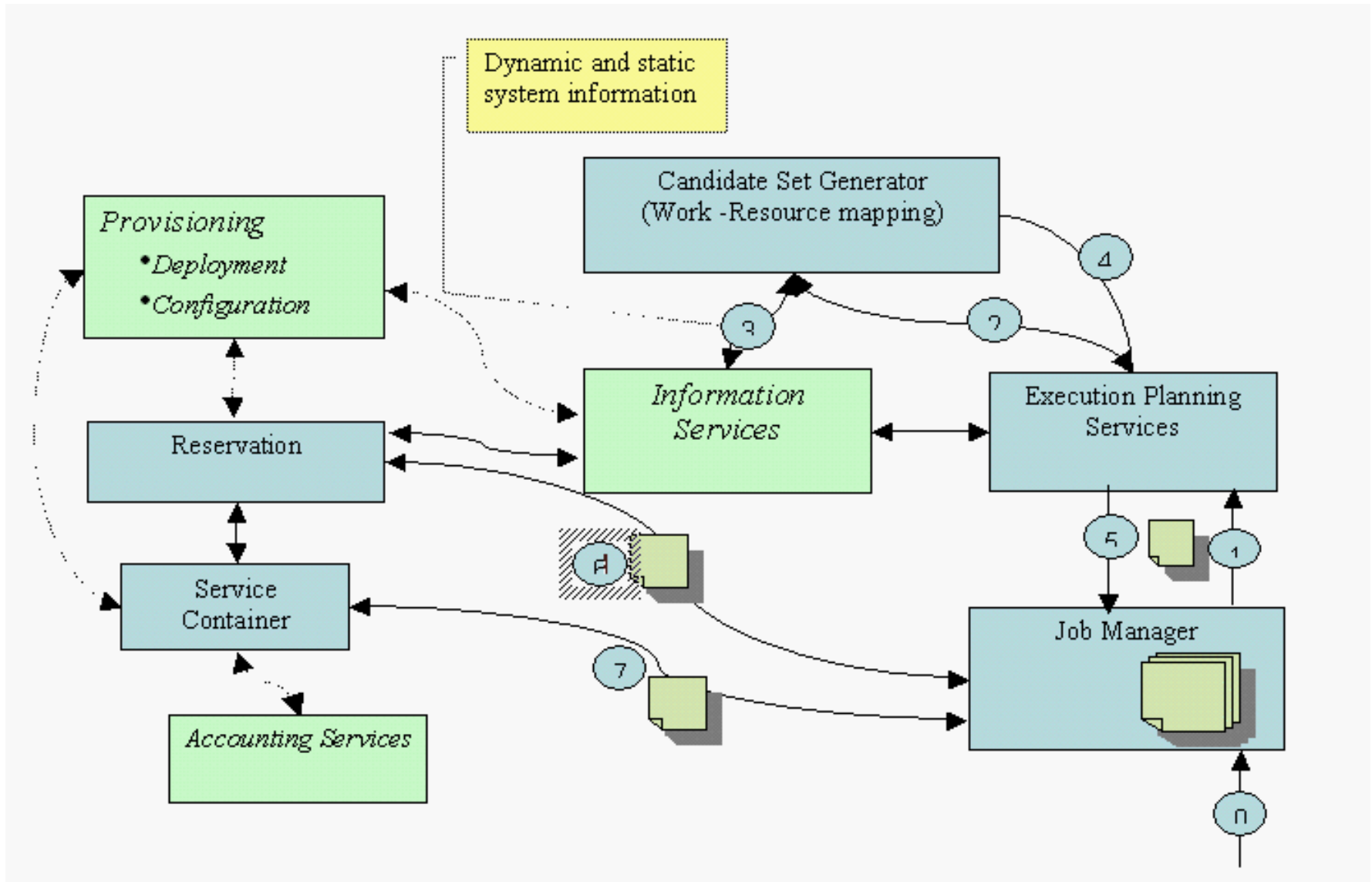
EMS Services

- Resources (service container, generico)
- Job management and monitoring services
- Resource selection services
 - Execution Planning
 - Candidate Set Generator
 - Reservation Services

Relazioni con altri servizi (I)

- Deployment & Configuration Service
- Naming
- Information Service
- Monitoring
- Fault Detection & Recovery Services
- Logging
 - Metering
 - Auditing
 - Billing
 - Provenance

Relazioni con altri servizi (II)



Data Services (I)

Obiettivo: spostare dati, gestire copie, eseguire query ed aggiornamenti, federare risorse dati, gestire metadati (in particolare, provenance).

I Data Services NON si basano e non specificano la semantica dei dati trattati.

Tipi di risorse dati:

- flat files: accesso posix-like, query in formato db-like, XML Query ove appropriato
- streams: query e trasformazioni
- DBMS: relazionali, XML, OO
- catalog: ad ex. directory
- derivazioni: risultati di query asincrone o trasformazioni
- data services: per esempio, sensori o programmi che generano dati

Data Services (II)

Scenari di utilizzo:

- accesso remoto
- staging
- replica
- federazione
- derivazione
- metadati

Problemi (alcuni)

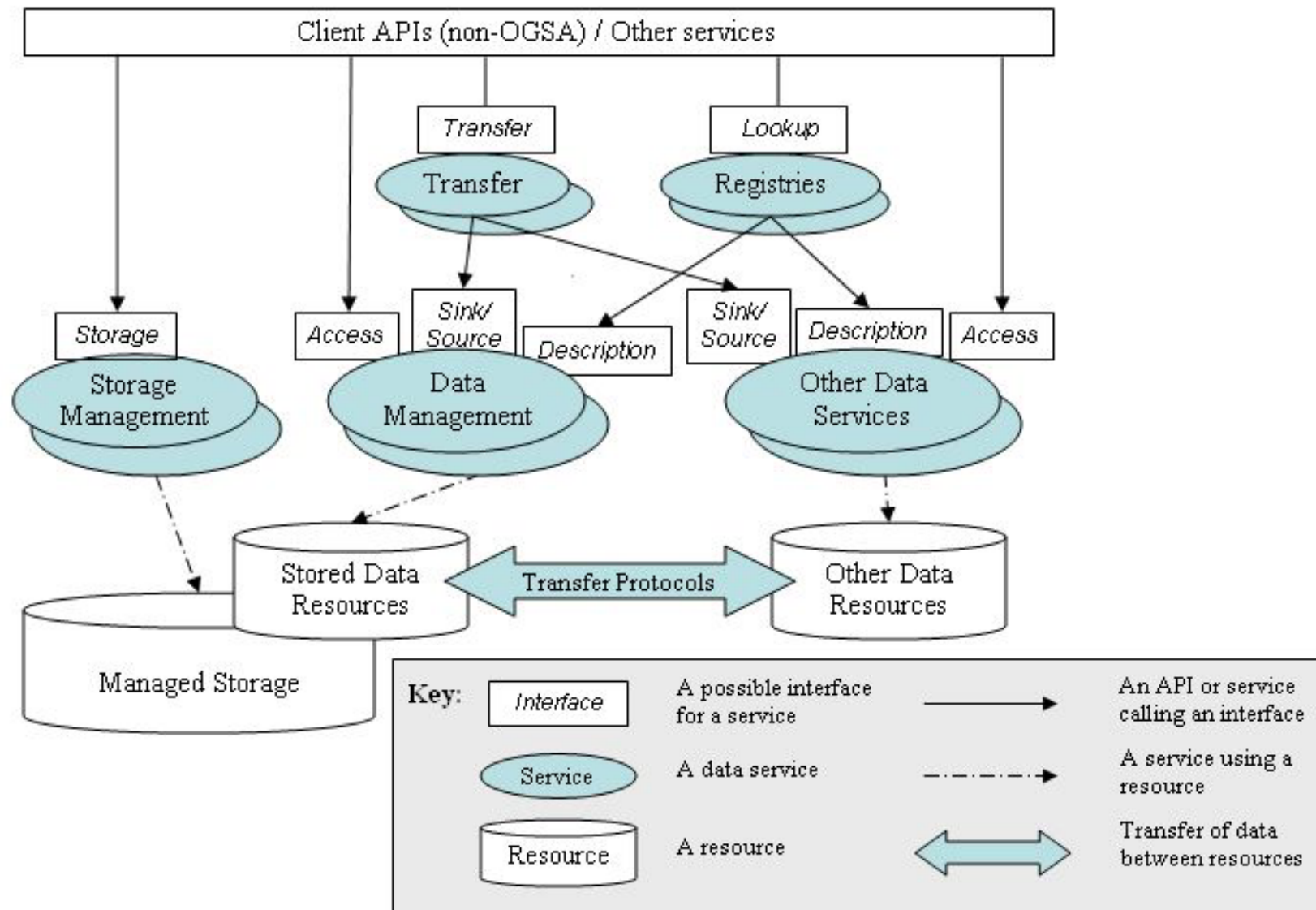
- trasparenza e virtualizzazione
- possibilita' di utilizzare client API (NSF, JDBC, ODBC,....)
- tipi di dati e operazioni estensibili

Data Services (III)

Capabilities funzionali:

- data transfer
- storage management
- simple access (read/write ByteIO)
- query: sincrone, asincrone
- federation
- location management
- update
- transformation
- security extensions
- resource & service configuration
- metadata catalogs
- data discovery (infrastructure)
- provenance

Data Services (IV)



Relazioni con altri servizi

- transazioni
- logging
- deployment & configuration services
- workflow
- execution planning & job management
- resource reservation
- discovery
- security
- network management
- naming
- notification

Resource Management Services (I)

Obiettivi:

- gestione delle risorse fisiche e logiche (ad ex. reboot di un host)
- gestione delle risorse di una Grid OGSA esposte attraverso interfacce di servizi (resource reservation, job submission, monitoring)
- gestione dell'infrastruttura di una Grid OGSA, esposta attraverso le proprie interfacce di gestione (ad ex. monitoraggio di un registry service)

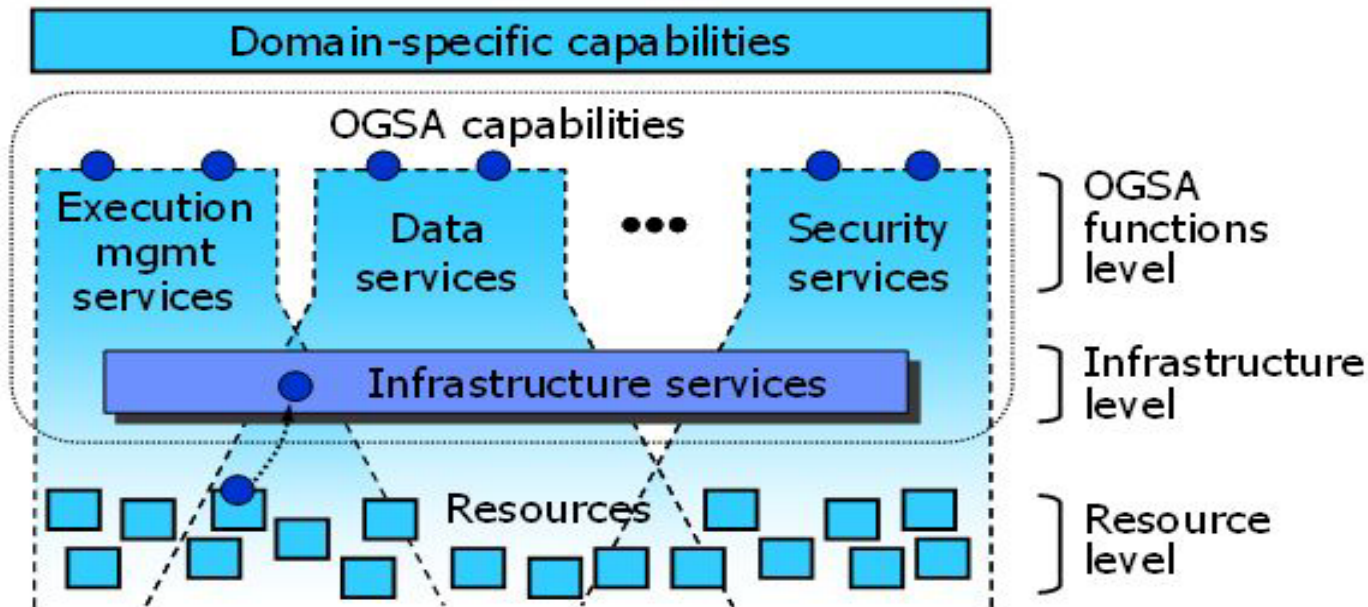
Modello: a tre livelli

Type of management	Level of interface	Interface
Management of the physical and logical resources	Resource level	CIM/WBEM, SNMP, etc.
	Infrastructure level	WSRF, WSDM, etc.
Management of the OGSA Grid Resources	OGSA functions level	Functional interfaces
Management of the OGSA Grid Infrastructure		Specific manageability interfaces

- basato sulle interfacce:
 - un servizio puo' implementarne piu' di una
 - un servizio puo' essere separato dalla risorsa che rappresenta

Resource Management Services (II)

1. livello resource: monitoring, setup & control, basato su un *information model*
2. livello infrastructure: standardizzazione
 - manageability model di base: WS-RF
 - generic manageability interface
3. livello functions:
 - functional interface
 - manageability interface: eventualmente, estensione della generic interface



Resource Management Services (III)

- capabilities a livello 2: basate su WSDM, in particolare su MOWS e MUWS
- information model: CIM ed estensioni
- capabilities a livello 3:
 - policy & policy management
 - reservation
 - monitoring & control
 - VO management
 - security management
 - problem determination & fault management
 - metering
 - discovery
 - deployment
- interazione con gli altri servizi: TUTTI

Security Services (I)

Obiettivi:

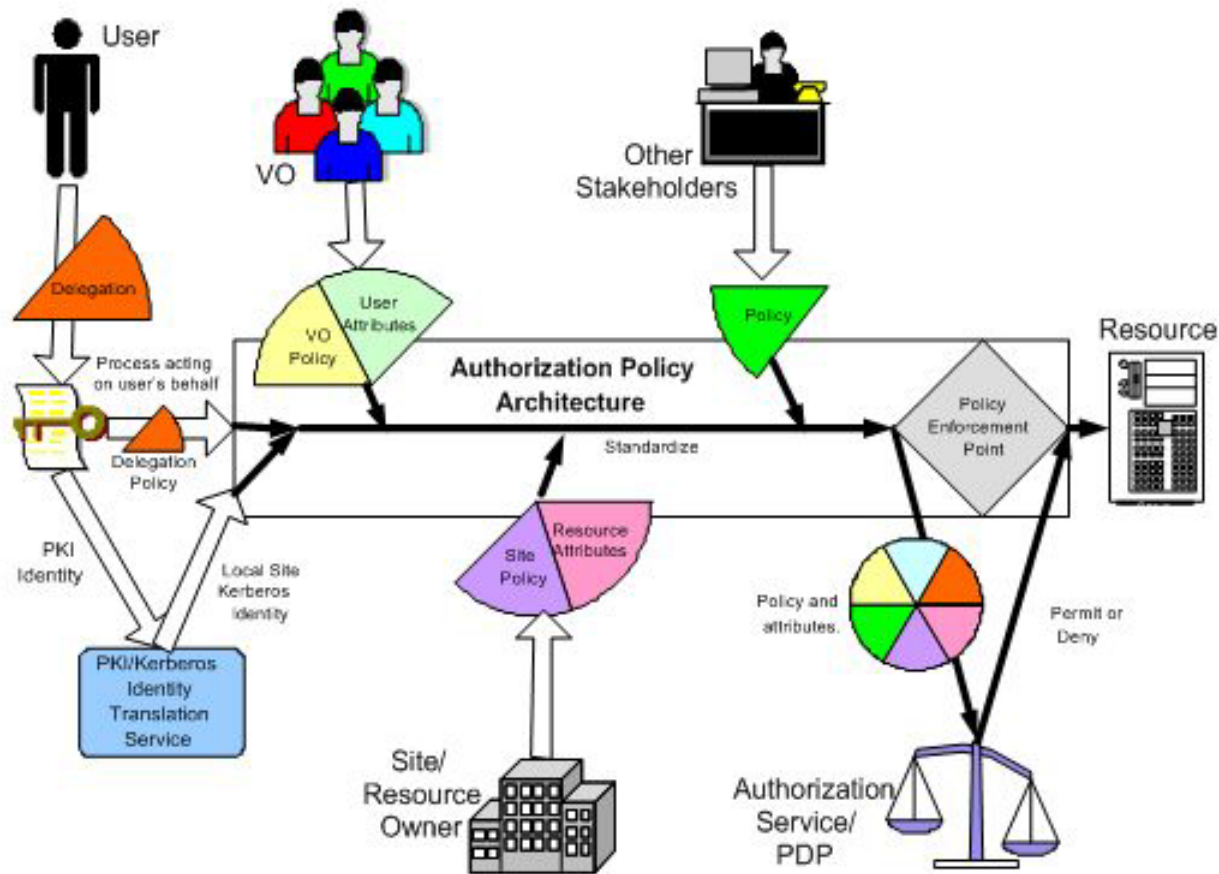
- permettere il soddisfacimento delle policy legate alla sicurezza all'interno di una VO
- tutte le interazioni associate ad un'attività in Grid devono sottostare alle policy locali E CONTEMPORANEAMENTE a quelle della VO
- l'architettura deve essere *implementation agnostic*
- estensibile
- integrabile con servizi di sicurezza esistenti
- deve garantire che i servizi agenti attraverso più domini siano in grado di interagire: interoperabilità a più livelli
- federazione di meccanismi di sicurezza

Security Services (II)

Il modello:

- entita': utenti, soggetti, servizi
- interazioni: mail, HTTP, SOAP, SSL,....
- contesto: localizzazione di un'interazione
- attributi: per descrivere (identificare, raggruppare)
- attributi *inescapable*: auto-identificativi (private/public key, impronte digitali)
- tutti gli altri attributi sono legati a questi attraverso un *issuer* (authority)
- policy: affermazioni (regole) restrittive sui valori degli attributi e sulle loro relazioni
- policy: espresse in termini di entita', risorse, contesto
- security services: entita' con modelli di interazione che facilitano l'amministrazione, espressione, pubblicazione, scoperta, comunicazione, verifica, rinforzo e riconciliazione delle policy di sicurezza
- il modello concettualmente coincide con WS-Sec, pero' centrato sul soddisfacimento concorrente di molteplici policies, ciascuna delle quali deve essere valutata nel proprio contesto

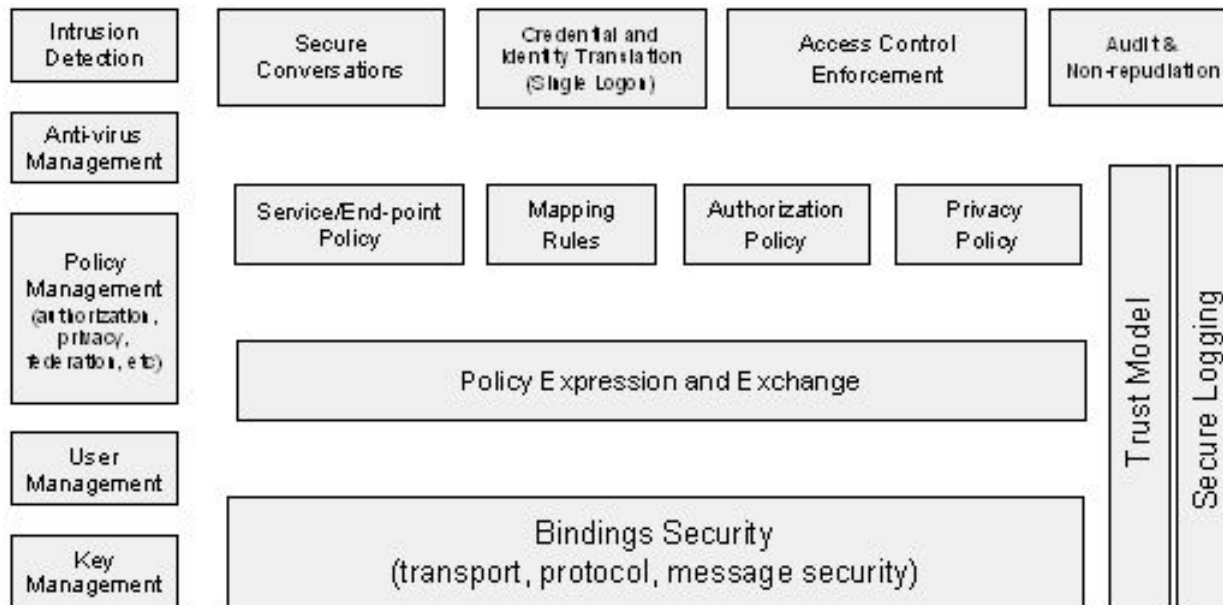
Security Services (III)



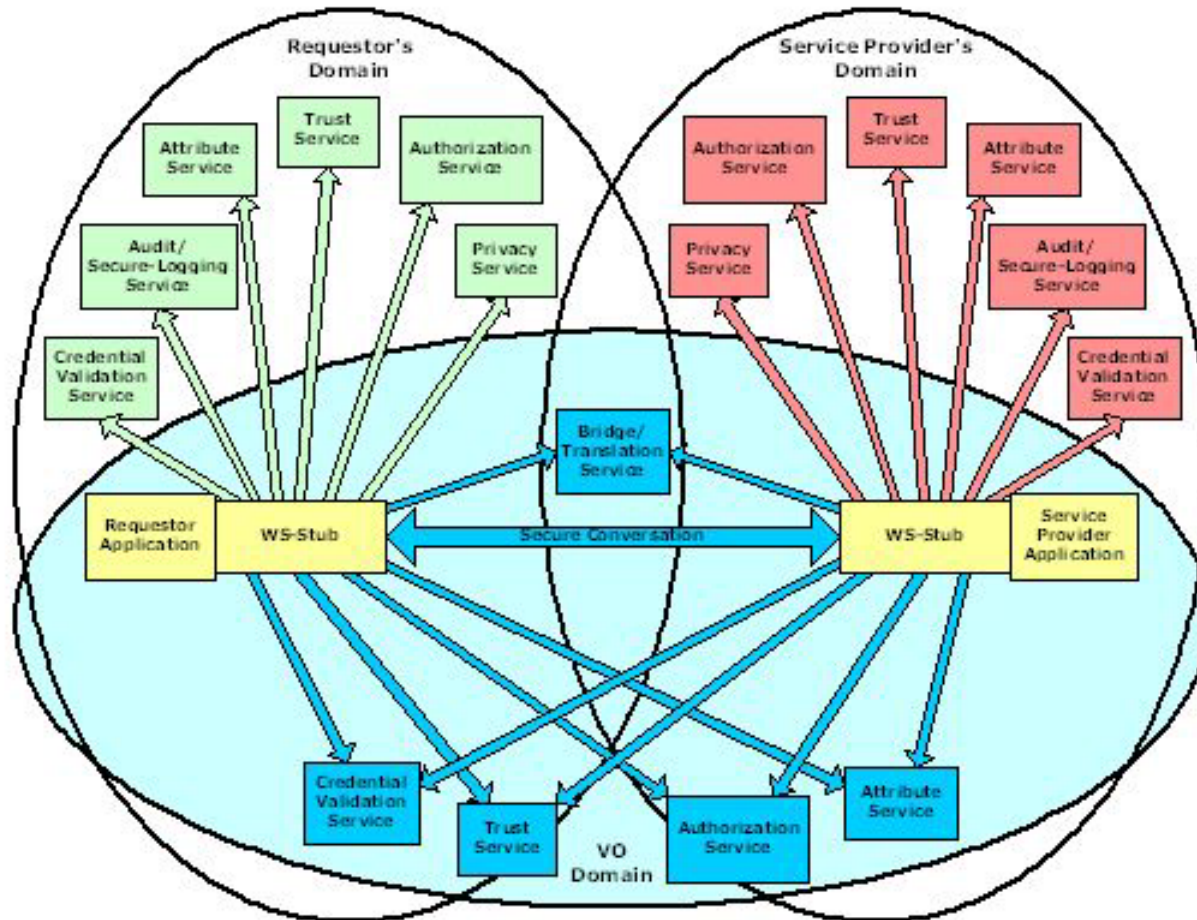
Security Services (IV)

Capabilities funzionali e servizi:

- autenticazione: Credential Validation Service, Trust Service
- identity mapping: Trust Service, Attribute Service, Bridge/Translation Service
- autorizzazione: Authorization Service
- conversione delle credenziali: Trust Service, Attribute Service, Bridge/Translation Service
- Audit & Secure Logging: Audit Service
- Privacy: Privacy Service



Security Services (V)



Interazioni con gli altri servizi: TUTTI

Self-Management Services

Obiettivi:

- ridurre il costo e la complessita' di gestione
- raggiungere i livelli di servizio attesi
- utilizzando Service Level Managers
- consentendo composizioni e gerarchie di SLM

E' un ambito
ancora
strettamente di
ricerca, collegato al
campo degli
autonomic systems

Capabilities funzionali: sono strettamente intercorrelate

- self configuring
- self-healing
- self-optimizing

Relazioni con altri servizi:

- discovery
- logging e monitoring
- reservation
- workflow
- composition
- security
- resource management

Information Services (I)

Obiettivi:

- accedere e manipolare efficientemente informazioni su applicazioni, risorse, servizi
- bilanciare la necessita' di servizi generici con requisiti su semantiche e QoS specifici
- Per facilitare riuso ed interoperabilita', sfruttare le capabilities dell'infrastruttura OGSA, in particolare *notification*

Modello:

- produttori, consumatori, intermediari
- produttori e consumatori sono disaccoppiati
- il modello non e' prescrittivo: correntemente, XML-based e SQL-based
- alle informazioni (eventi, messaggi) sono associati metadati
- e' auspicabile uno schema di eventi OGSA standard

Information Services (II)

Capabilities funzionali

- discovery
- message delivery
- logging
- monitoring
- General Information and Monitoring Service (GMA)

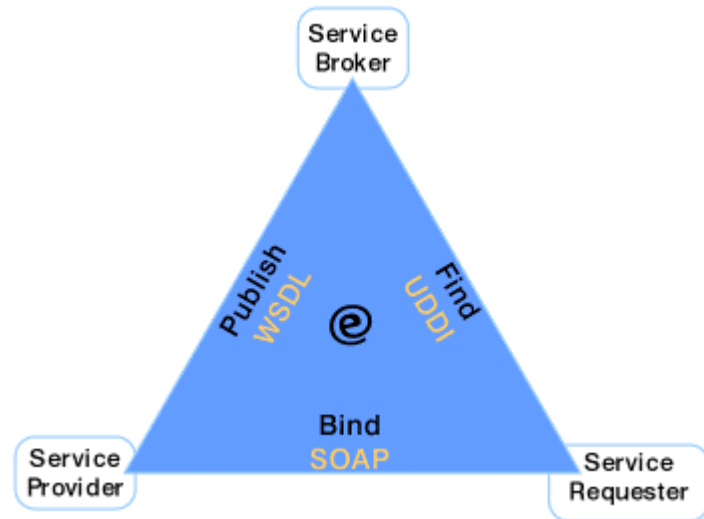
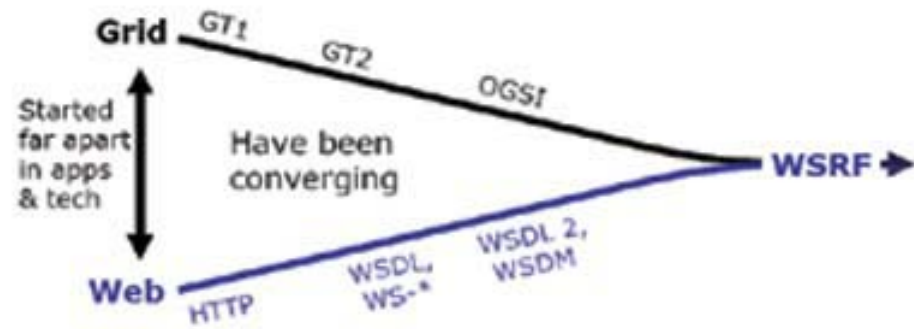
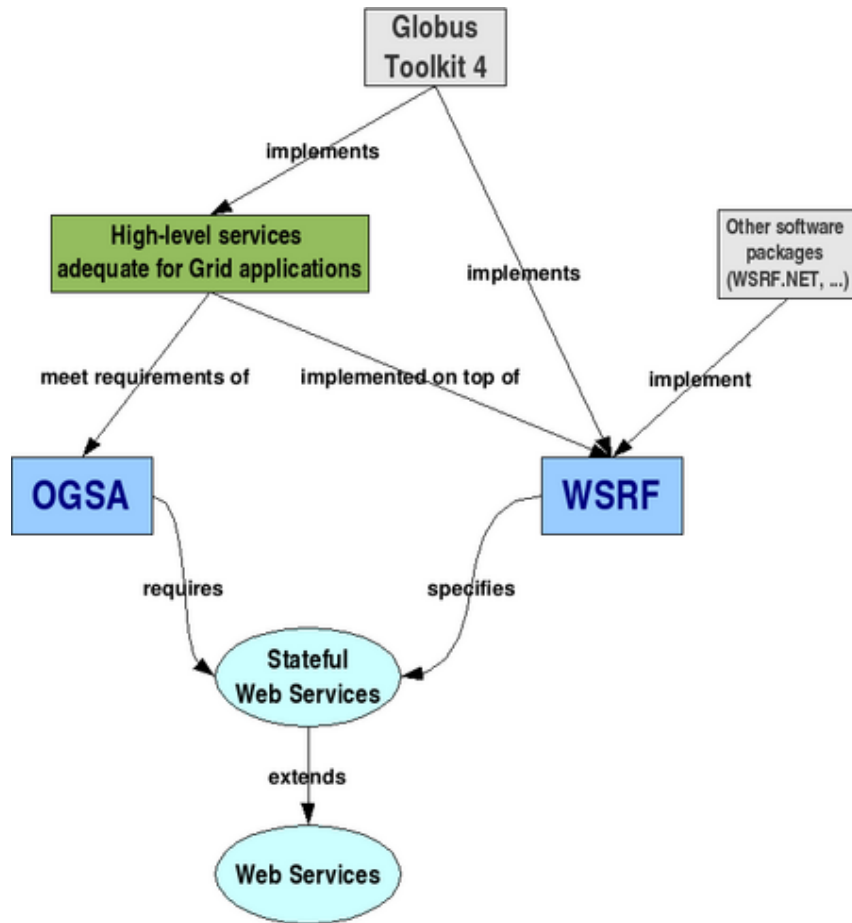
Interazioni con altri servizi

- allineamento con i servizi che gestiscono le risorse
- security services
- data federation
- replication

Open Grid Services Architecture

- reference implementation: Globus Toolkit 4

GT4 & Web Services



GT4 & Web Services Specs

- SOAP1.2 WSDL1.1

Security

- TLS (transport-level), WS-Security, WS-SecureConversation.
- X.509 End Entity Certificates, Username & Password
- X.509 Proxy Certificates, WS-Trust
- SAML assertions

Addressing

- WS-Addressing

Resource

- WS-RF
- WS-ResourceProperties
- WS-ResourceLifetime
- WS-ServiceGroups
- WS-BaseFaults

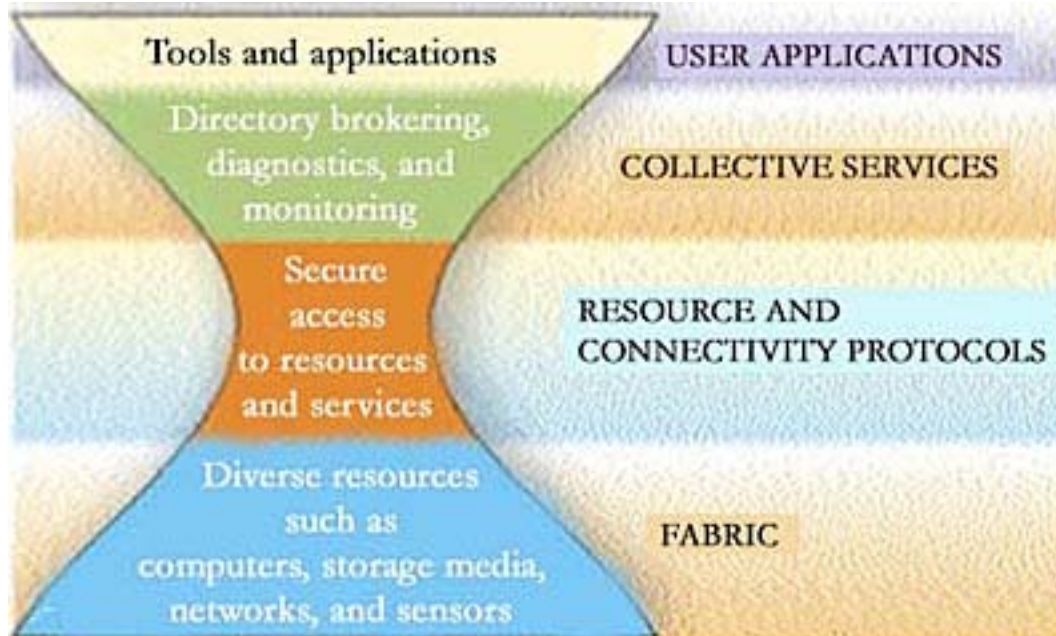
In via di adozione:

- WSDM
- WS-CIM

Notification

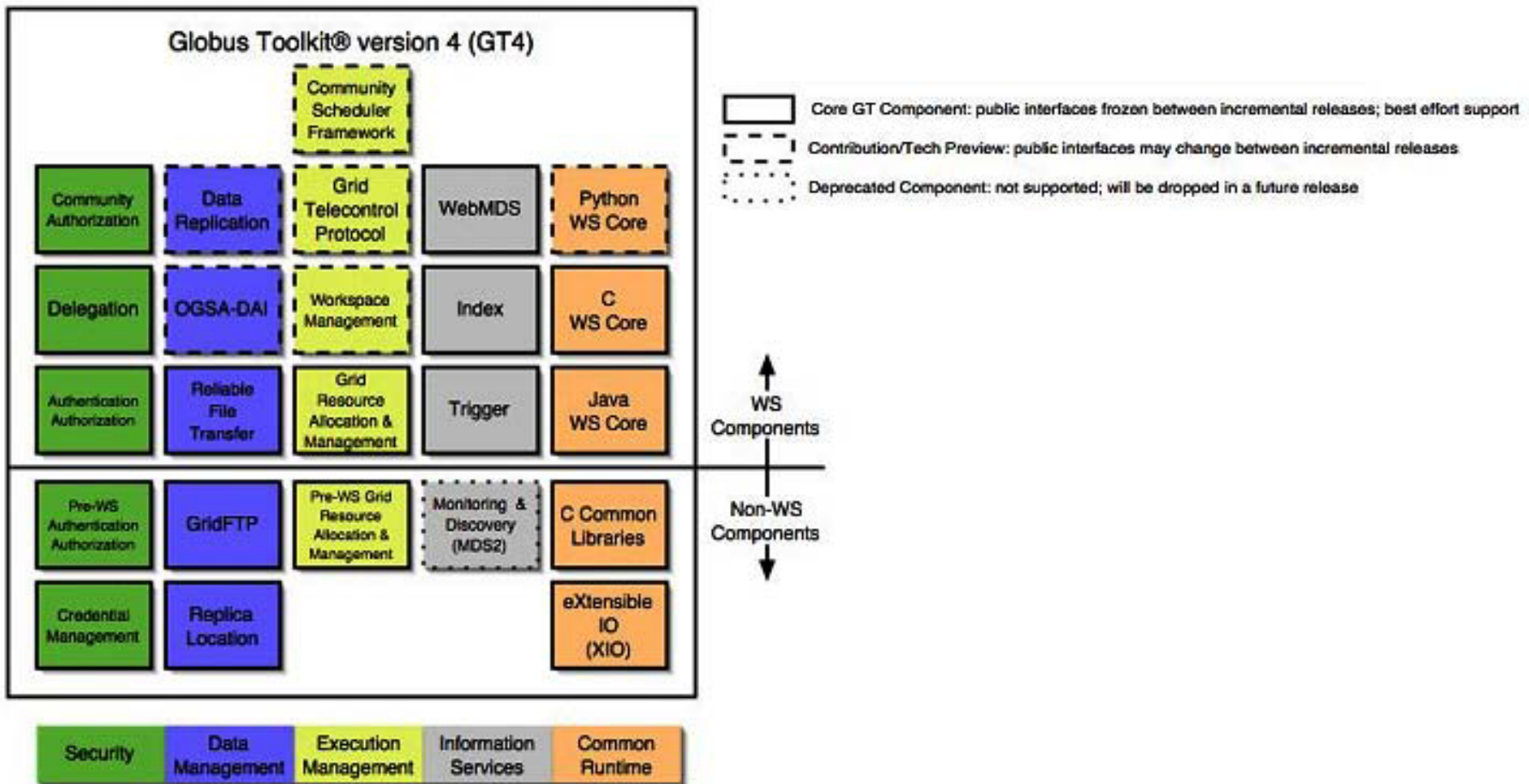
- WS-Notification
- WS-Topics
- WS-BrokeredNotification

GT4 Hourglass Model

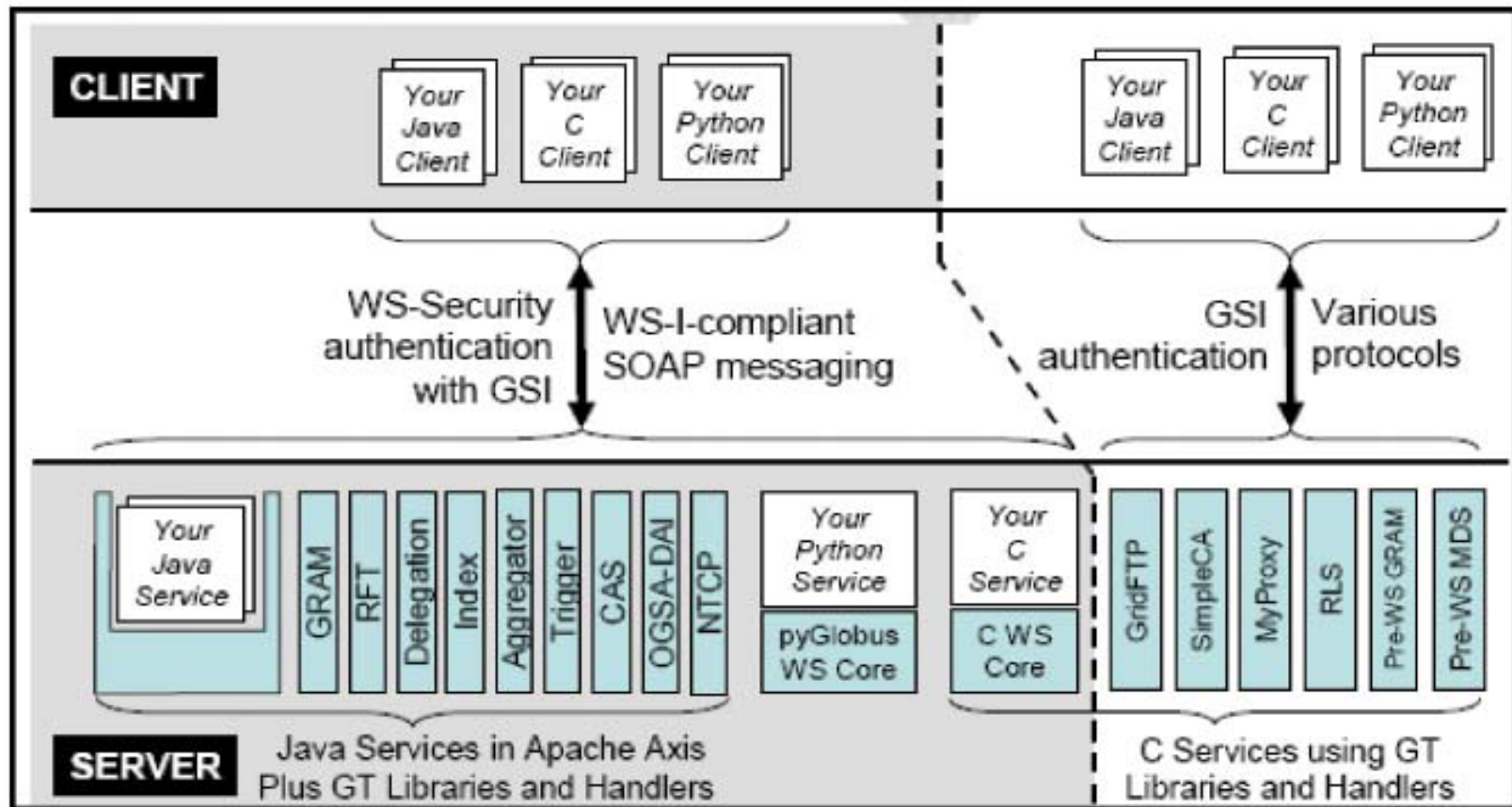


- coordinare risorse multiple: servizi di infrastruttura ubiqui, servizi distribuiti application specific
- condividere risorse singole, negoziare l'accesso, controllare l'uso, comunicazione e sicurezza
- controllare le cose localmente: accesso e controllo, interfaccia con sistemi locali

GT4 Layers



GT4: Services & pre-Services



GT4 Security: Grid Security Infrastructure

	Message-level Security w/X.509 Credentials	Message-level Security w/Username and Passwords	Transport-level Security w/X.509 Credentials
Authorization	SAML and grid-mapfile	grid-mapfile	SAML and grid-mapfile
Delegation	X.509 Proxy Certificates/ WS- Trust		X.509 Proxy Certificates/ WS- Trust
Authentication	X.509 End Entity Certificates	Username/ Password	X.509 End Entity Certificates
Message Protection	WS-Security WS-SecureConversation	WS-Security	TLS
Message format	SOAP	SOAP	SOAP

Fornisce

- Autenticazione (one-way o mutua)
- Integrità dei messaggi e riservatezza
- Delegation

Estende lo standard di certificazione X.509 per includere i *proxy certificates* per delegation e singolo accesso (sign-on)

Due modi operativi: **Transport-level** e **Message-level** security.

Transport-Level: implementazione originale di GSI

- Usa SSL/TLS, esteso per single-sign-on e delegation
- Assume un protocollo di trasporto connection-based (es., TCP).
- Usa certificati X.509 per autenticazione e per stabilire session keys.

Message-level: nuova implementazione di GSI

- Usa WS-Security, XML-Signature e protocolli associati
- Session-based security (che assume un protocollo di trasporto connection-based e usa session keys)
- Per-message security (che non richiede protocollo di trasporto connection-based).

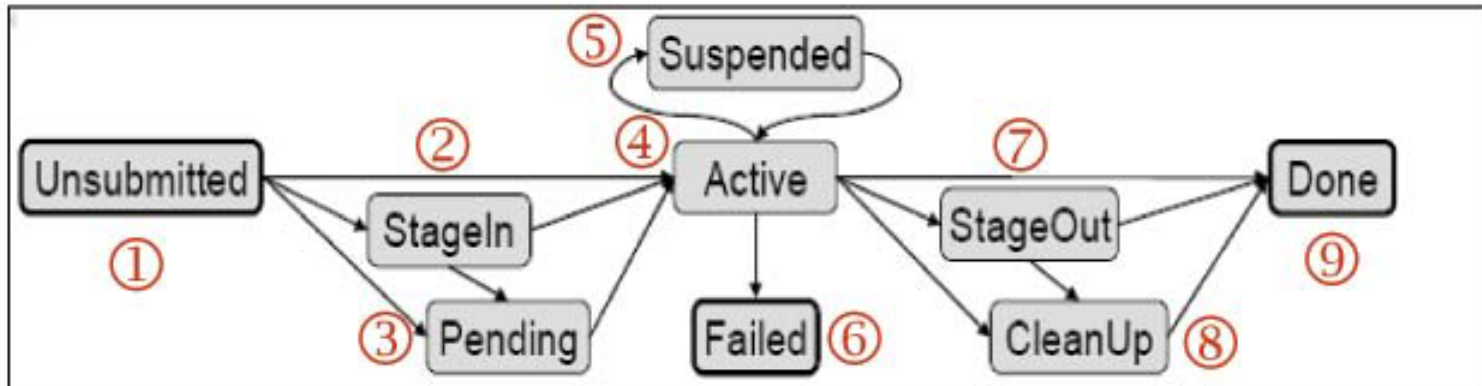
GT4 Execution Management

- GT4 fornisce una serie di web service per eseguire, monitorare e terminare l'esecuzione di job su una Griglia.
- Essi compongono il WS_GRAM che estende le funzionalità del componente GRAM (Grid Resource Allocation and Management) esistente prima dell'uso dei web services.
- Interfacce a linee di comandi e API in Java, C e Python.
- Il WS_GRAM permette la gestione dell'esecuzione remota di programmi in maniera affidabile, con meccanismi di monitoring, gestione delle credenziali, staging dei file da usare, e interazione con gli scheduler locali.
- Gestisce:
 - Gli eseguibili
 - Lo stato del processo
 - L'I/O
 - Il controllo remoto
 - Gli scheduler locali
 - Il monitoraggio dei processi

Job Management Tools

<i>Grid Resource Allocation & Management service</i>	<i>GRAM service supports submission, monitoring, and control of jobs on computers. Interfaces to Unix shell ("fork"), Platform LSF, PBS, and Condor schedulers; others may be developed. Includes support for MPICH-G2 jobs: multi-job submission, process coordination in a job, sub-job coordination in a multi-job.</i>
<i>Java CoG Kit Workflow</i>	<i>Uses the Karajan workflow engine that supports DAGs, conditions, & loops; directs tasks to GRAM servers for execution.</i>
<i>Community Scheduler Framework</i>	<i>CSF is an open source meta-scheduler based on the WS-Agreement specification.</i>

Job Status



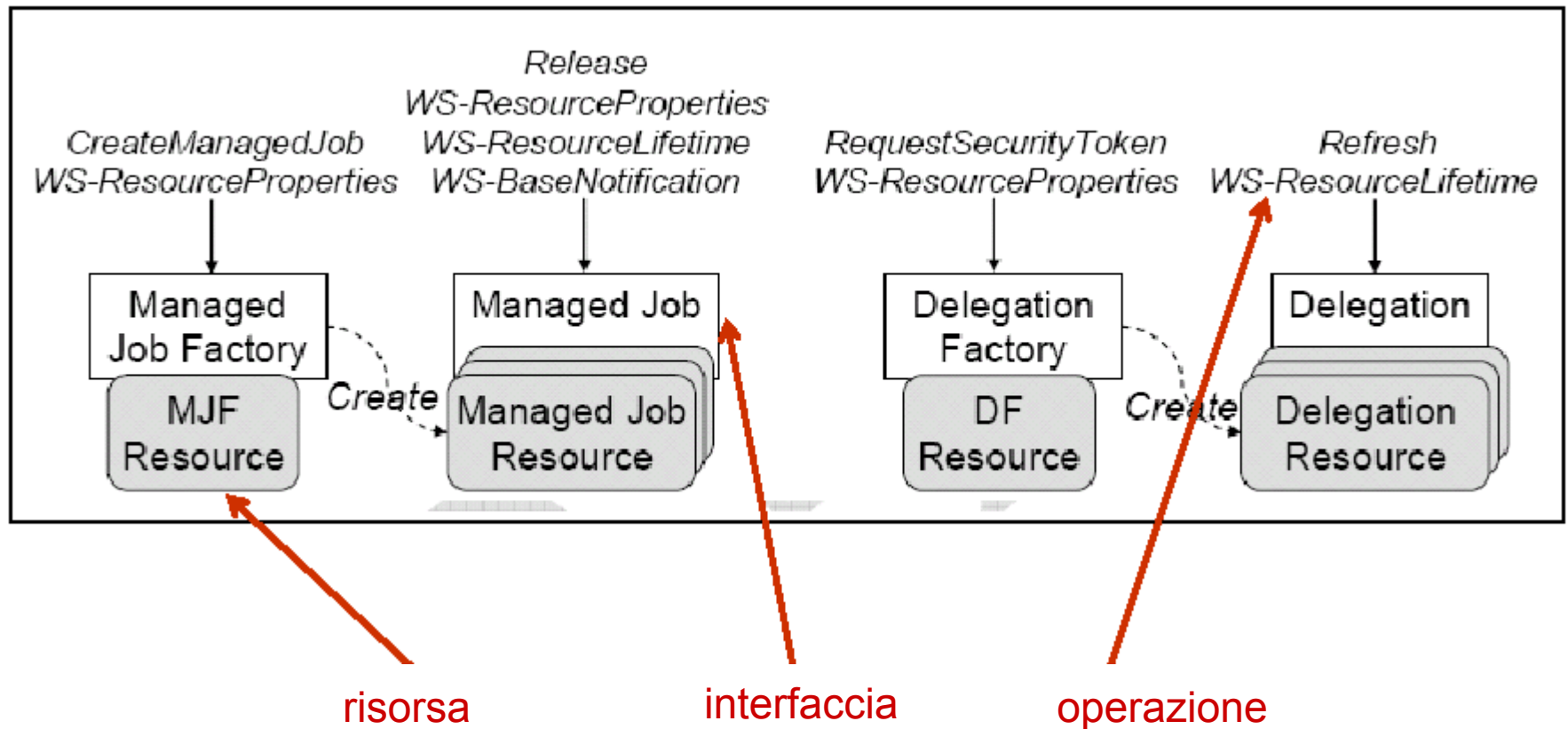
Stato	Descrizione
Unsubmitted	Di prossima esecuzione
StageIn	In attesa che i file vengano resi disponibili per l'esecuzione
Pending	In attesa che lo scheduler locale ne decida l'esecuzione
Active	In esecuzione
Suspended	Job sospeso
StageOut	Esecuzione completata e file di output resi disponibili
CleanUp	Rimozione delle risorse usate
Done	Job completato con successo
Failed	Job fallito

GRAM Client API

L'interfaccia client del GRAM è basata su 4 portTypes che definiscono operazioni su 4 WSResources che rappresentano:

1. Lo stato del GRAM server,
 2. Lo stato di un singolo job,
 3. La catena dei certificati di una delegation factory,
 4. Una credenziale.
-
1. La Managed Job Factory (MJF) portType gestisce lo stato del GRAM server (WS-ResourceProperties) e gestisce la creazione di job.
 2. La Managed Job (MJ) portType definisce operazioni sulla ManagedJob WS-Resource.
 3. La Delegation factory portType definisce operazioni sulla catena dei certificati associati alla delegation factory.
 4. La Delegation portType definisce operazioni sul lifetime di una WS-resource e l'acquisizione di nuove credenziali.

GRAM Client API

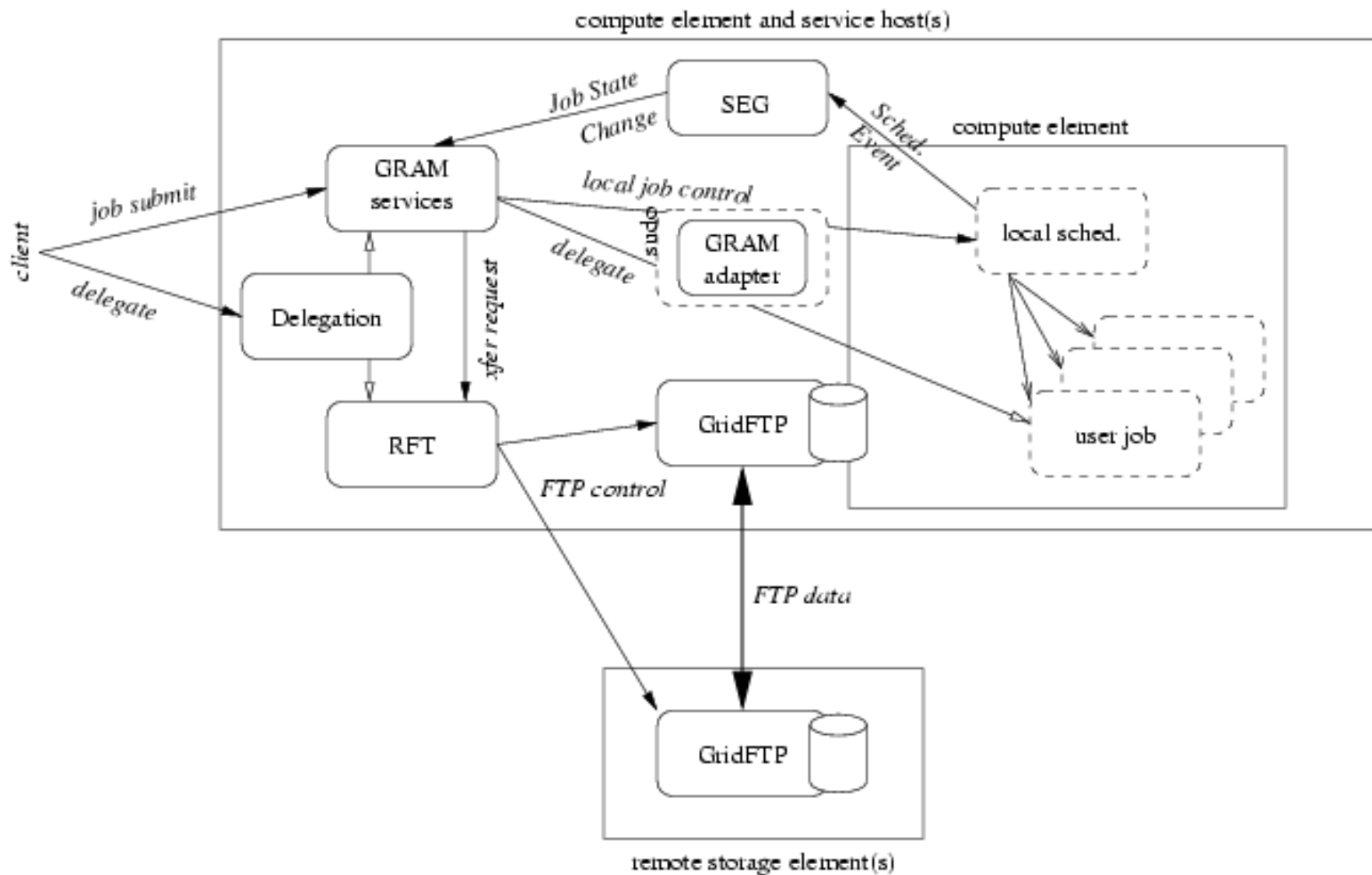


GRAM Client API

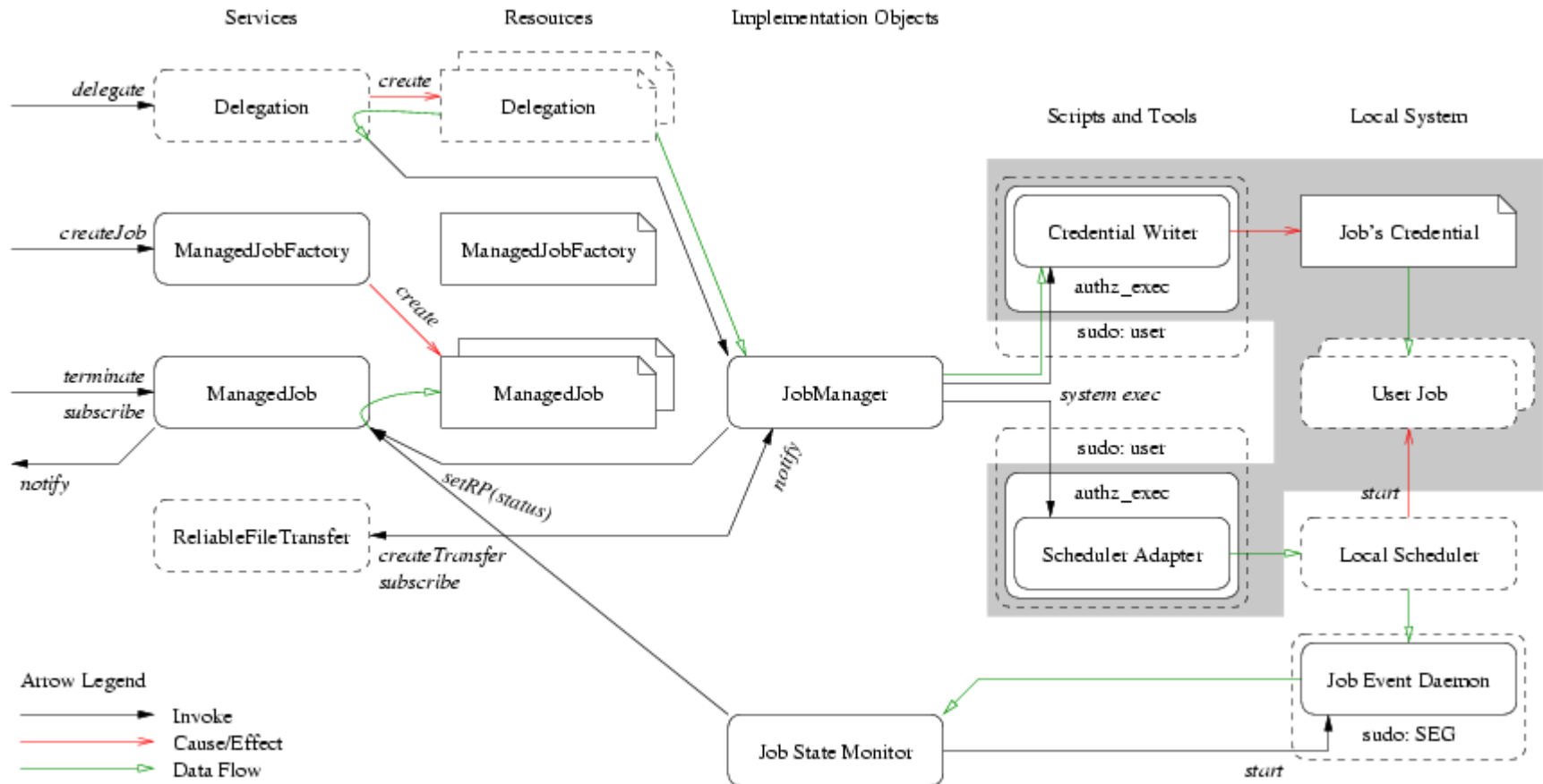
Proprieta' delle risorse:

Resource Property	Description
localResourceManager	Local resource manager type: e.g., Condor, Fork, LSF, Multi, PBS.
globusLocation	The location of the Globus Toolkit installation that these services are running under.
hostCPUType	The job host CPU architecture (i686, x86_64, etc...)
hostManufacturer	The host manufacturer name. May be "unknown."
hostOSName	The host OS name (Linux, Solaris, etc...)
hostOSVersion	The host OS version.
scratchBaseDirectory	The directory recommended by the system administrator to be used for temporary job data.
delegationFactoryEndpoint	The endpoint reference to the delegation factory used to delegate credentials to the job.
stagingDelegationFactoryEndpoint	The endpoint reference to the delegation factory used to delegate credentials to the staging service (RFT).
condorArchitecture	Condor architecture label (for Condor schedulers).
condorOS	Condor OS label (for Condor schedulers).
GLUECE	GLUE data (data in GLUE schema format [18]).
GLUECESummary	GLUE data summary.

Funzionamento del GRAM



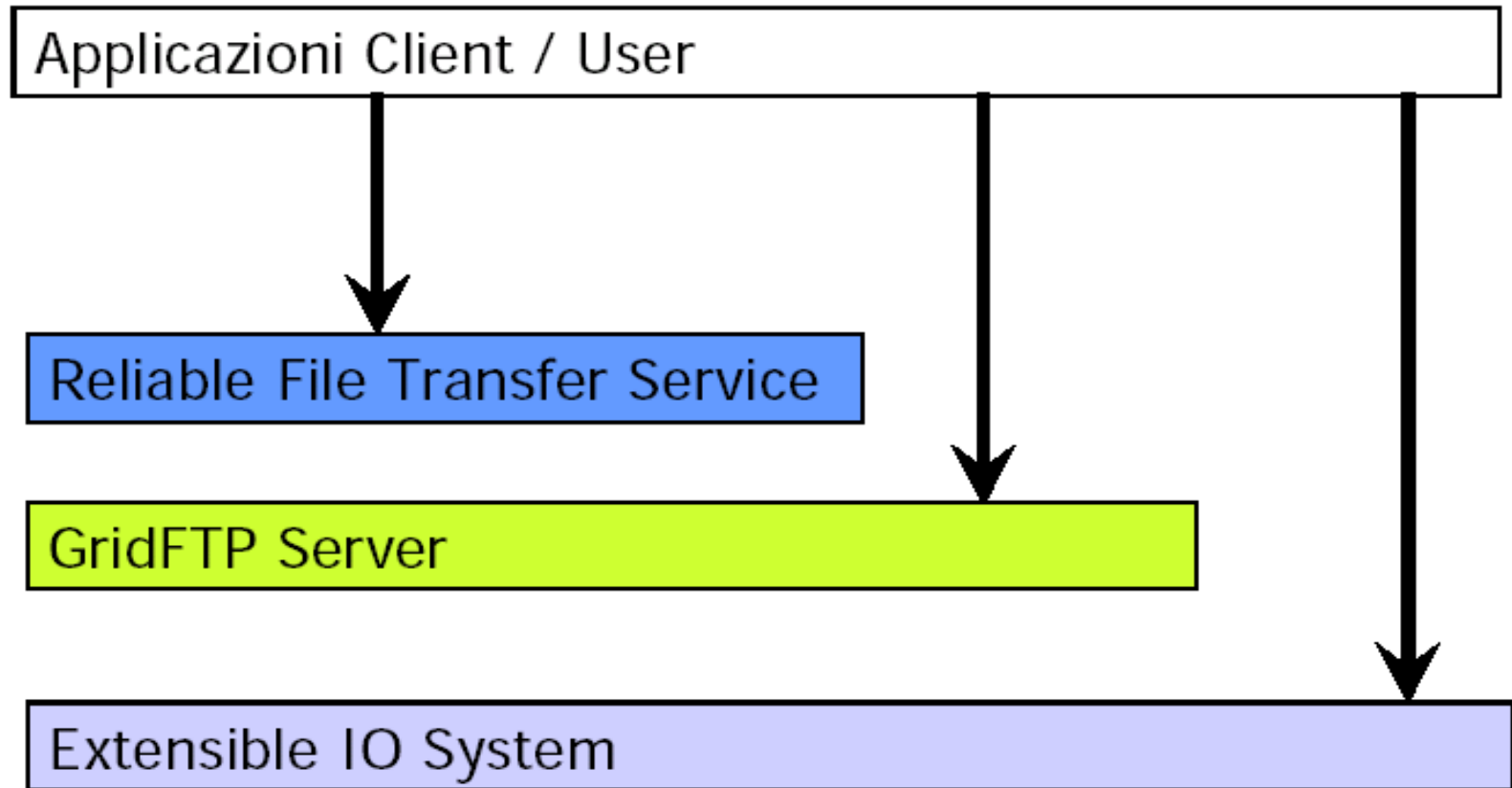
Architettura Software del GRAM



GT4 Data Management

- **GridFTP:** High-performance Data transfer protocol
- **The Reliable File Transfer Service (RFT):** Servizi di trasferimento di dati in GT4
- **The Replica Location Service (RLS) :** Registry Distribuito per gestire copie di dati
- **Data Access and Integration Service (OGSA-DAI):** Servizi di accesso a databases relazionali e XML

Data Transport



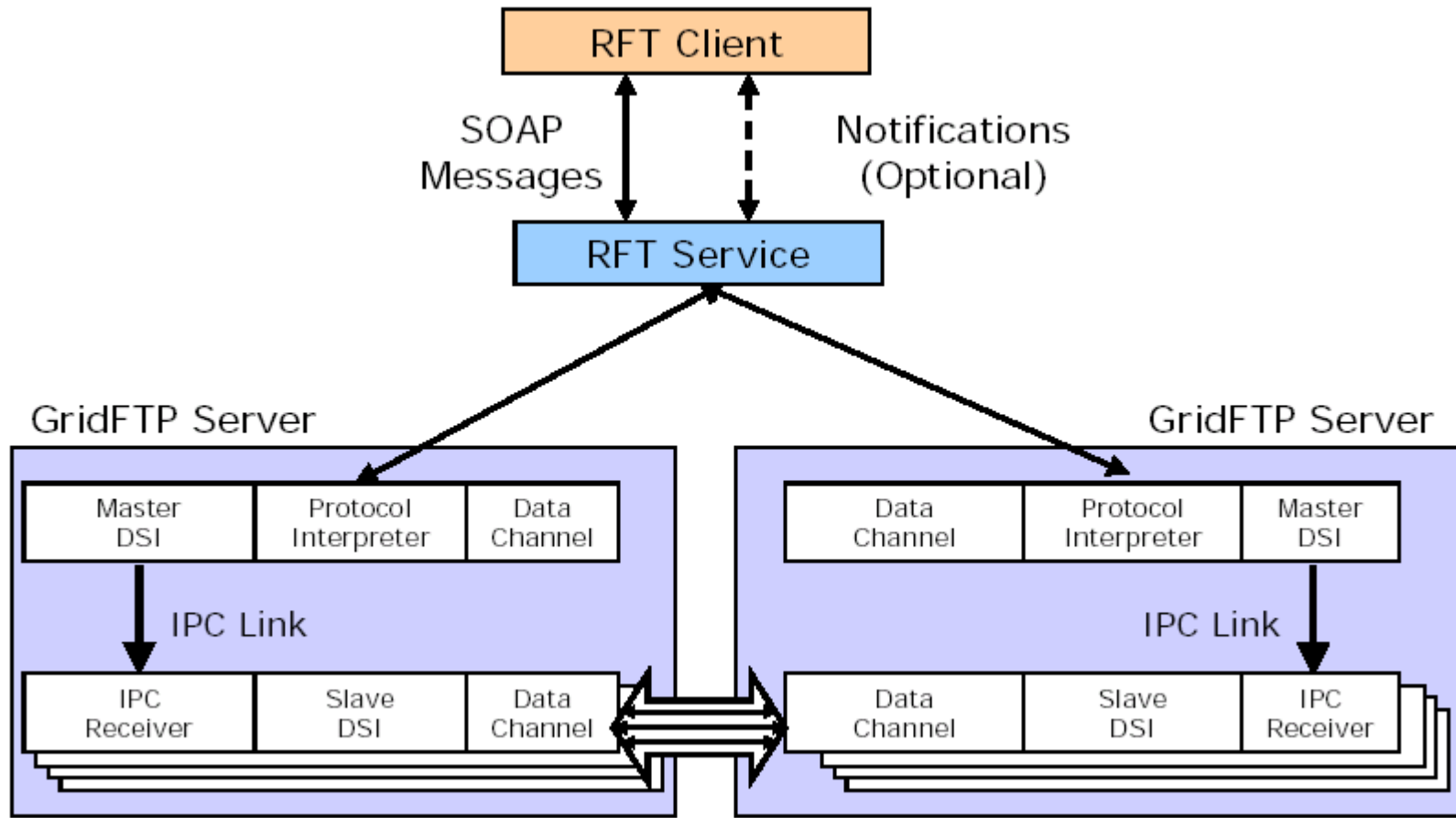
Grid FTP

- Protocollo di data transfer sicuro, robusto, efficiente, standard.
- Più implementazioni indipendenti possono interoperare
- Globus Toolkit fornisce una implementazione di riferimento:
 - Server
 - Strumenti Client (globus-url-copy)
 - Librerie
- Basato sulla libreria XIO (eXtensible Input/Output)
- Definisce funzioni standard: Read/Write/Open/Close
- Molto modulare per permettere l'integrazione con una varietà di sorgenti data sources (files, mass stores, etc.)
- Implementa lo striping
- Include supporto per IPV6

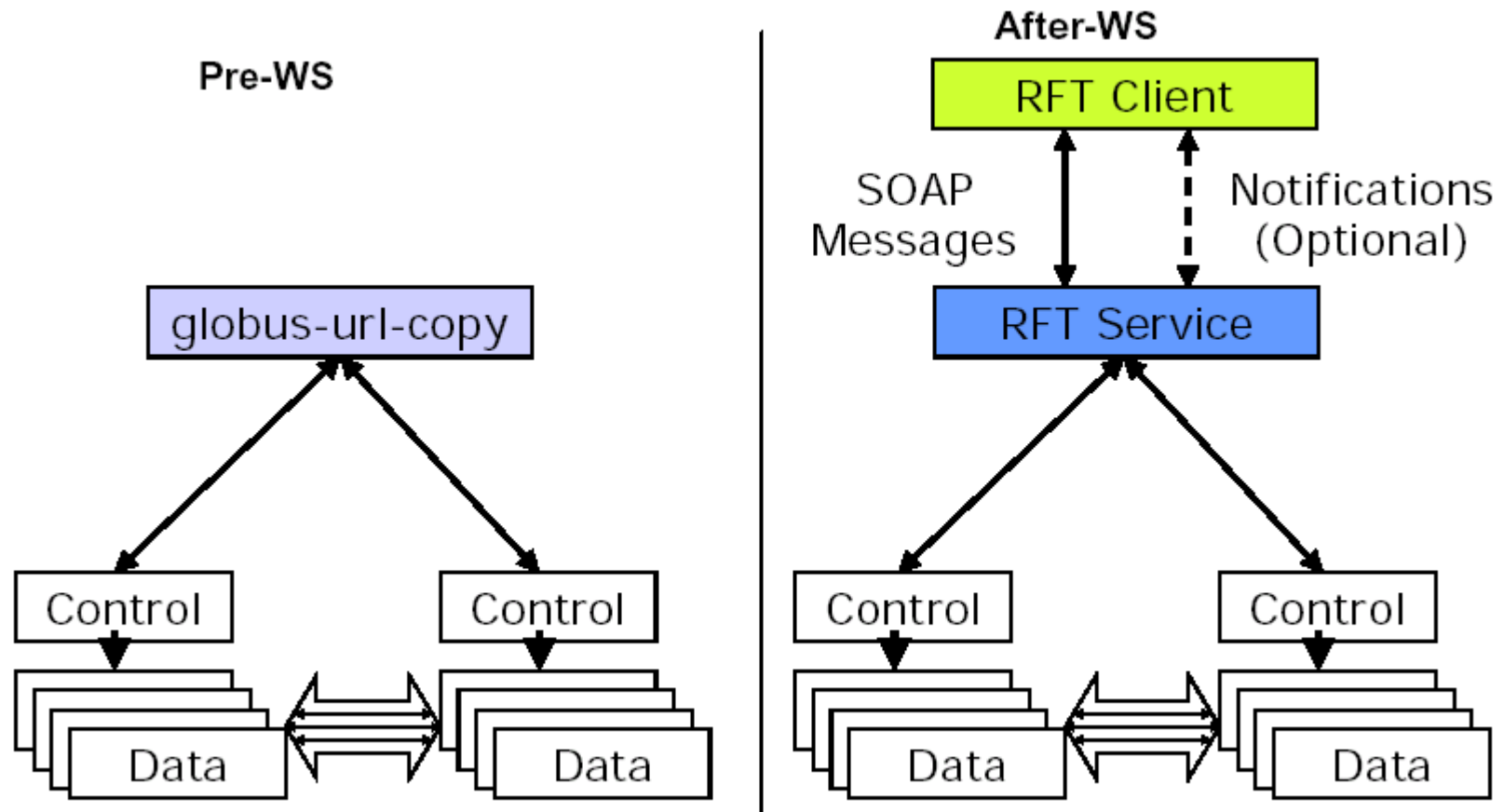
Reliable FTP

- Servizio di trasferimento dati ad alte prestazioni WS-RF compliant e Fault-tolerant.
 - Notifications/Query
- Affidabilità sulle alte prestazioni fornite da GridFTP
 - Integrated Automatic Failure Recovery
 - Network level failures.
 - System level failures.
- Scheduler per trasferimento di dati con una politica FIFO di gestione della coda.

Reliable FTP

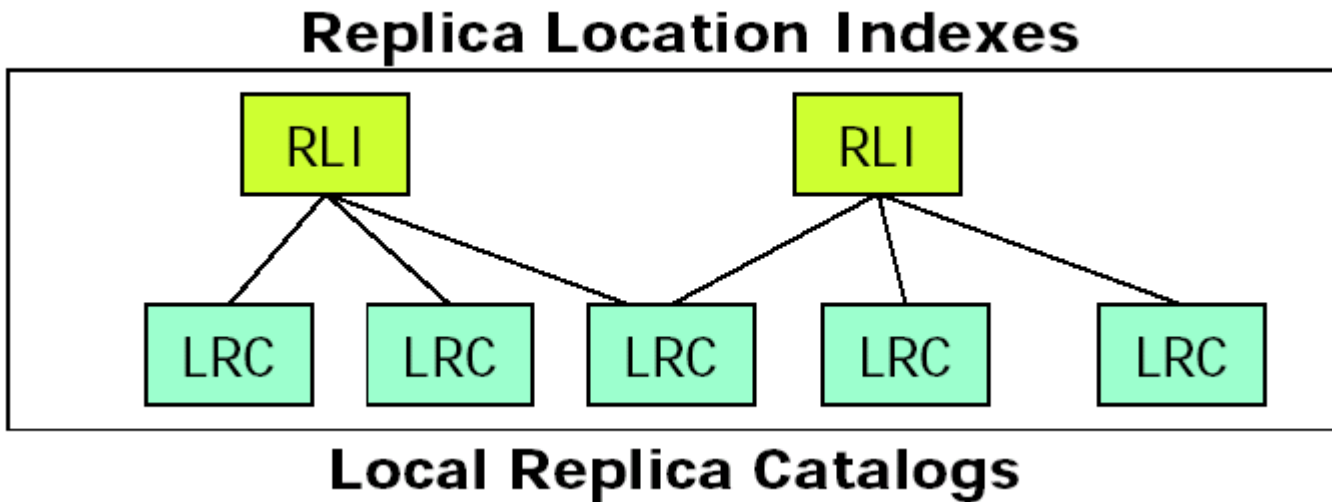


GridFTP e RFT



Replica Location Service

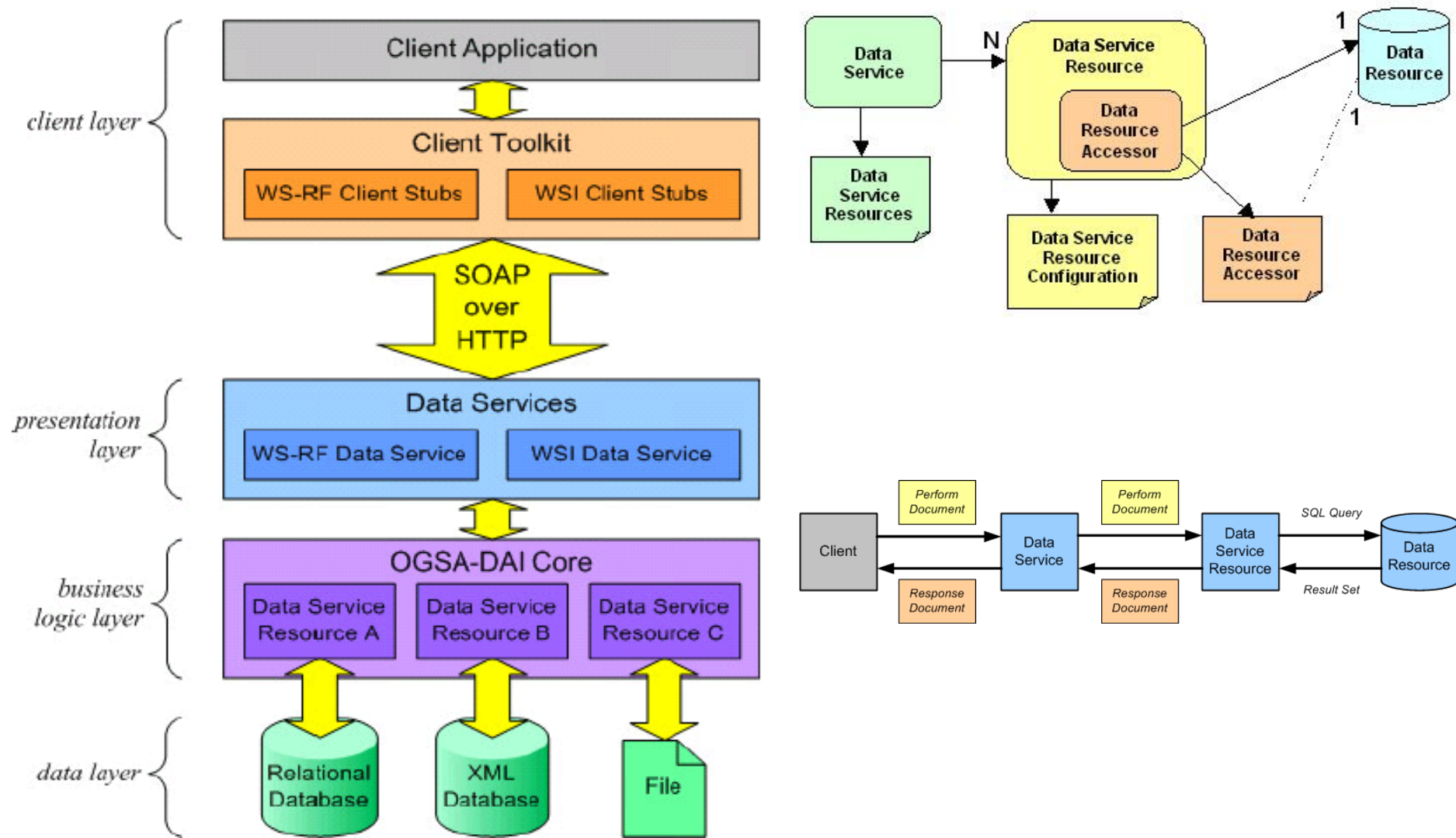
- E' un registry distribuito che memorizza copie di dati e gestisce la localizzazione di repliche
- RLS mantiene i mapping tra gli identificatori *logici* e i *target names* (localizzazioni dei dati replicati)
- RLS è un componente del Replica Management system di Globus
- unitamente ai consistency services, replica selection services, reliable file transfer, ecc.



OGSA-DAI

- Un modello per l'accesso integrato a database XML e relazionali:
- Interfaccia consistente per database su Grid
- Integrazione con diversi DBMS (Oracle, DB2, MySQL, PostgreSQL, SQL Server ...)
- Una base per sviluppare servizi di più alto livello
 - Data federation
 - Distributed query processing
 - Data mining
 - Data visualisation

OGSA-DAI



GT4 Information Services

- Il Monitoring and Discovery System e' un insieme di web services.
- Consente agli utenti di scoprire quali risorse fanno parte di una VO e monitorarle
- Le interfacce implementate consentono di:
 - interrogare i servizi WSRF per informazioni sulle proprieta' delle risorse
 - eseguire un programma per acquisire dati
 - interfacciarsi con sistemi di monitoraggio esterni

Index Service: registry analogo ad UDDI, piu' flessibile.

- gli Indexes raccolgono e pubblicano *resource properties*
- i client usano le interfacce standard WSRF per interrogare/ricevere notifiche
- gli indexes si possono registrare reciprocamente per aggregare informazioni
- ogni entry ha un lifetime

Trigger Service: raccoglie informazioni e le confronta con un set di condizioni

- definite in un file di configurazione
- quando se ne verifica una, intraprende un'azione

GT4 Information Services

Aggregator Framework: usato per costruire servizi che aggregano informazioni

- raccolte via Aggregator Sources (java classes), XML data
- usano un meccanismo di configurazione comune: relazione fra servizio e Aggregator Source
- si registrano usando WSRF Service Group all'Aggregator Service
- ogni registrazione ha un lifetime

Aggregator Sources

- Query aggregator Source: interroga un servizio WSRF, raccoglie resource property info.
- Subscription Aggregator Source: raccoglie dati da un servizio WSRF via subscription/notification
- Execution Aggregator Source: esegue un programma custom

GT4 Information Services

- Information Providers: il software usato da un'Aggregation Source per creare ed aggiornare le proprie properties
- Hawkeye: raccoglie dati su risorse appartenenti ad un pool Condor usando il mapping XML del GLUE schema, e le riporta ad un WS GRAM service che le pubblica come le proprie resource properties
- Ganglia information provider: analogo per cluster basati su Ganglia
- WS GRAM: il job submission service. Pubblica informazioni sullo scheduler locale
- RFT: stato del server, transfer status per files, transfer attivi...
- Community Authorization Service: info per identificare la VO che serve
- tutti i servizi che pubblicano resource properties: ogni servizio GT4

GT4 Tools

- installazione e configurazione:

<http://www.globus.org/toolkit>

<http://www.globus.org/toolkit/docs/4.0/admin/docbook/quickstart.html>

- serve: un account globus, un account globus_user, due certificati (host, utente), un grid-mapfile per il container
- ant, jdk (NON GJC)
- GLOBUS_LOCATION
- source \$GLOBUS_LOCATION/etc/globus_devel_env.sh
- source \$GLOBUS_LOCATION/etc/globus_user_env.sh
- per impacchettare il file .gar: globus-build-service.sh
- per fare il deployment: globus-deploy-gar.sh

Sample GT4 Service

- MathService: implementa le 4 operazioni (port types)
- ogni versione con caratteristiche di sicurezza diverse
- per dimostrare che la sicurezza NON influenza l'interfaccia di un servizio
- service: stateless
- resource: stateful class (interfacce Resource, ResourceProperties)
- resource home: gestisce la risorsa (estende SingletonResourceHome)

Sample GT4 Service v. 1

GSI Secure Conversation

- service: security descriptor
- <authz value="none">
- collegamento fatto nel file WSDD
- client: `_setProperty()`, nelle stubs
- tre call per creare un *security context*, solo il corpo del messaggio e' criptato.
- codice del servizio:
/org/globus/example/services/security/first/impl/MathService.java
/org/globus/example/services/security/first/impl/MathResource.java
/org/globus/example/services/security/first/impl/MathResourceHome.java
- file wsdl:
/schema/examples/MathService_instance_4op/Math.wsdl
- security descriptor:
org/globus/examples/services/security/first/etc/security-config-first.xml
- deployment descriptor:
org/globus/examples/services/security/first/deploy-server.wsdd
- codice del client:
/org/globus/example/clients/MathService_instance_4op/Client_GSISecConv_Encrypt.java

Sample GT4 Service v. 1

- build & deploy service
`./globus-build-service.sh sec_first globus-deploy-gar`
`$EXAMPLES_DIR/org_globus_examples_services_security_first.gar`
- compile client
`javac -classpath ./build/stubs/classes/:$CLASSPATH \`
`org/globus/examples/clients/MathService_instance_4op/Client_GSISecConv_Encrypt.java`
- start container:
`globus-start-container -nosec`
- create proxy
`grid-proxy-init`
- run client
`java -classpath ./build/stubs/classes/:$CLASSPATH \`
`org.globus.examples.clients.MathService_instance_4op.ClientGSISecConv_Encrypt \`
`http://127.0.0.1:8080/wsrf/services/examples/security/first/MathService`
- java e javac: dell'sdk, NON GJC
- il container non supporta l'hot deployment: va fermato (ctrl+c) e riavviato

Sample GT4 Service v. 2

Client con security descriptor

- <GSISecureConversation>
- client: path al security descriptor, _setProperty(), nelle stubs
- il container ha il proprio security descriptor:
 - path alle credenziali
 - context-lifetime
 - grid-mapfile (se usato)
 - e' in \$GLOBUS_LOCATION/etc/globus_wsrf_core/global_security_descriptor.xml
- security descriptor del client:
\$EXAMPLES_DIR/org/globus/examples/clients/MathService_instance_4op/GSISecConv_Encrypt.xml
- codice del client:
org/globus/examples/clients/MathService_instance_4op/Client_SecDesc.java
- per provarlo, analogo a prima (il servizio e' lo stesso).

Sample GT4 Service v. 3

Service authentication

- e' possibile specificare un valore di default : <auth-method>
 - <GSISecureConversation/>
 - <GSISecureMessage/>
 - <GSITransport/>
- usando il tag <method>, si puo' specificare un valore diverso
- usando il tag <protection-level>, si puo' specificare il grado di protezione
 - <integrity/> (digital signatures)
 - <privacy/> (encryption)
- ipotesi: il servizio usa le credenziali del container in /etc/grid-security/containercert.pem
- se non e' cosi', va modificato il path (il servizio puo' avere le sue credenziali)
- il security descriptor:
\$EXAMPLES_DIR/org/globus/examples/services/security/first/etc/security-config-auth.xml
- il nome del servizio: MathAuthService

Client authentication

- Analogo a prima.
- Se si vuole GSI Secure Message (no Security Context), si deve specificare come recuperare la public key del server
- <peer-credentials> nel Security Descriptor
- oppure un po' di righe di codice nelle stubs

Sample GT4 Service v. 3.1

- MathAuthService:
 - add: GSI Secure Conversation
 - subtract: GSI Secure Message
 - multiply: GSI Secure Conversation oppure GSI Secure Message
 - divide: GSI Transport
 - altri metodi: qualsiasi (GetResourceProperty)
- client: uno dei due precedenti, con GSI Secure Conversation
 - da programma
 - con security descriptor

OKKIO ai risultati...

Sample GT4 Service v. 3.2

- service: lo stesso di prima
- container: TLS
- client: TLS
- il codice del client:

`$EXAMPLES_DIR/org/globus/examples/clients/MathService_instance_4op/Client_GSITransport.java`

- faccio partire il container con `globus-start-container -nosec`

OKKIO ai risultati....

- faccio partire il container con `globus-start-container` , in modo da fargli usare TLS: cambiano protocollo e porta

```
java -classpath ./build/stubs/classes/:$CLASSPATH \
org.globus.examples.clients.MathService_instance_4op.ClientGSITransport \
https://127.0.0.1:8443/wsrf/services/examples/security/first/MathAuthService
```

OKKIO ai risultati....

Sample GT4 Service v. 4

- service: autorizzazione con gridmapfile specifico per il servizio
 - <authz>: valore "gridmap", path al file \$GLOBUS_LOCATION/etc/gridmapfile
 - \$EXAMPLES_DIR/org/globus/examples/services/security/gridmap/security-config.xml
 - codice del servizio:
\$EXAMPLES_DIR/org/globus/examples/services/security/gridmap/impl/MathService.java
 - build & deploy
./globus-build-service.sh sec-gridmap
globus-deploy-gar \$EXAMPLES_DIR/org_globus_examples_services_security_gridmap.gar
 - client: stesso di prima (Client_GSISecConv_Encrypt.java)
- OKKIO ai logs server side (mapping dell'identita' del client)
- modificare il gridmapfile cambiando identita' e riprovare

Sample GT4 Service v. 5

Subjects e Run-as Modes

1. System subject
2. Service subject
3. Resource subject

Se Resource non e' specificato, si assume Service.

Se Service non e' specificato, si assume System

Invocation subject: importante per delegation.

- Run as caller
- Run as system
- Run as service
- Run as resource

Per ogni metodo si puo' specificare un parametro <run-as> (<caller-identity/>,...)

Sample GT4 Service v.5

- MathService
 - protetto da gridmapfile (lo stesso di prima)
- PhysicsService: invoca MathService per stampare la risposta (getAnswer), non richiede autorizzazione
 - senza delega
 - con delega
- nel WSDD sono definiti due servizi, ciascuno con un Security Descriptor diverso
 - senza delega: nessuna autorizzazione, GSI Secure Conversation
 - con delega: nessuna autorizzazione, GSI Secure Conversation, run-as caller identity
- client:
 - senza delega (fallisce)
 - con delega: `_setProperty()` nella stub

Sample GT4 Service v.5

- l' esempio completo:
\$EXAMPLES_DIR/org/globus/examples/services/security/delegation
 - ./impl/PhysicsService.java
 - ./etc/security-config-nodeleg.xml
 - ./etc/security-config.xml
- il file wsdl: \$EXAMPLES_DIR/schema/examples/PhysicsService/Physics.wsdl
- il file wsdd: \$EXAMPLES_DIR/org/globus/examples/services/security/delegation/deploy-server.wsdd
- build & deploy
./globus-build-service.sh sec_delegation
globus-deploy-gar \$EXAMPLES_DIR/org_globus_examples_services_security_delegation.gar
- non-delegating client: MathService si rifiuta di collaborare perche' PhysicsService si presenta con le sue credenziali di default
\$EXAMPLES_DIR/org/globus/examples/clients/PhysicsService/ClientNoDelegation.java
- delegating client: MathService riconosce PhysicsService perche' si presenta con le credenziali dell'utente
\$EXAMPLES_DIR/org/globus/examples/clients/PhysicsService/Client.java

GGF Full Copyright Notice

Copyright (C) Global Grid Forum (2006). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the GGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the GGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the GGF or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE GLOBAL GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."