GT4: An introductory overview

Institut für Wissenschaftliches Rechnen

Olaf Schneider

using material of the Globus Alliance and
the National e-Science Centre, Edinburgh

Outline

- introduction and basic concepts
  - What is GT4?
  - History
  - SOA, Web Services and WSRF
- GT4 components and architecture overview
  - main functions of a grid system
  - WS components versus Pre-WS components
  - some details on selected components
  - a typical use case
  - interoperation architecture

What is GT4?

- GT4 is Globus Toolkit Version 4
- "The Globus® Toolkit is an open source software toolkit used for building grids." [www.globus.org/toolkit]
- GT4 is a WSRF compliant application development framework.
- GT4 is collection of ready-to-use of high-level Grid services.

History

- first ideas in the ealier 90s (metacomputing, I-WAY project)
- 1997 "Globus: a Metacomputing Infrastructure Toolkit" (Forster, Kesselman)
- 1998 GT 1.0
- 2002 GT 2.0
- 2003 GT 2.4 (used in gLite)
- 2003 GT 3.0 (OGSI)
- 2005 GT 4.0.0 (WSRF)
- 2006 GT 4.0.3 (current stable release)

What GT4 provides?

- **collection of solutions**, which frequently comes up when trying to build collaborative distributed applications
- impementation of **standards**:
  - use of existing standards (IETF, W3C, OASIS, GGF)
  - provide reference implementations of new/proposed standards
- a set of **services** focused on infrastructure management
  - common interfaces to heterogeneous resources
- standard-based security infrastructure
- client APIs and command line programs for accessing services
- a **toolkit** for building
  - new Web Services
  - service-oriented architectures (SOA)

Service Oriented Architecture (SOA)

- "Service Oriented Architecture is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations." (OASIS)

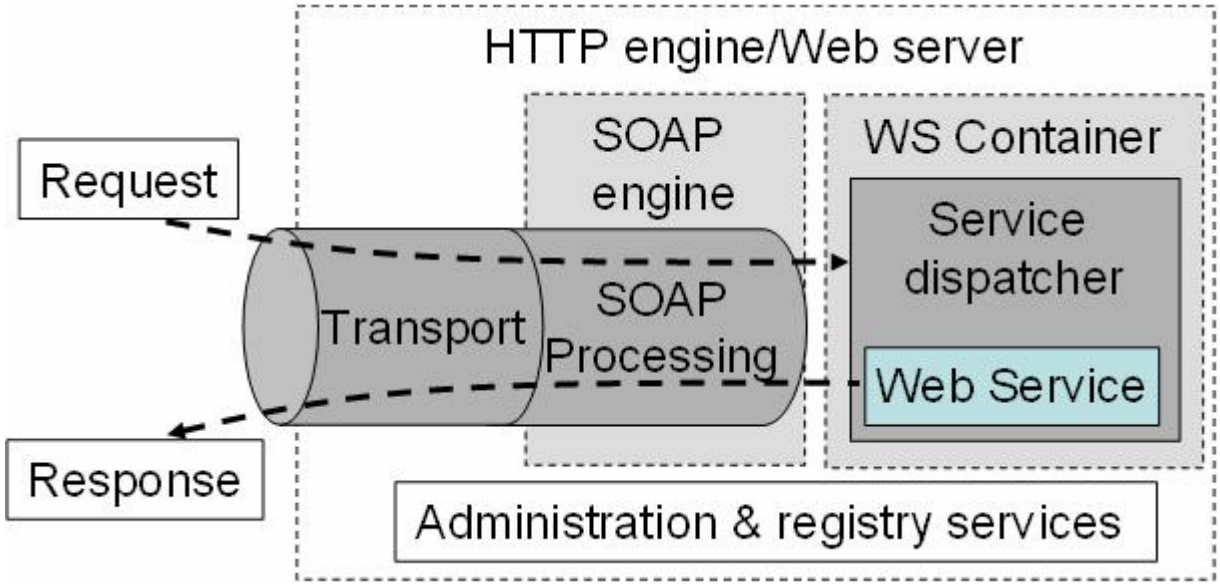
- SOAs are systems and applications structured as communicating services with uniform ways for
  - service describtion
  - invocation
  - secure access
  - …

Web Services

What are Web Services?

- A Web Service is a software system (programm, agent):
  - interoperable over a network (WAN), typical via HTTP
  - interface discribed in a machineprocessable format (WSDL)
  - interaction with the WS using SOAP messages (request/response scheme)
- characteristics of the WS architectur:
  - family of protocols (which heavily use XML)
  - machine-to-machine interaction
  - mechanisms for service registry and discovery (UDDI)
  - classical WS are stateless, provide only functionality
- A Web Service is *not*:
  - a Web site or portal

Web Service Architecture



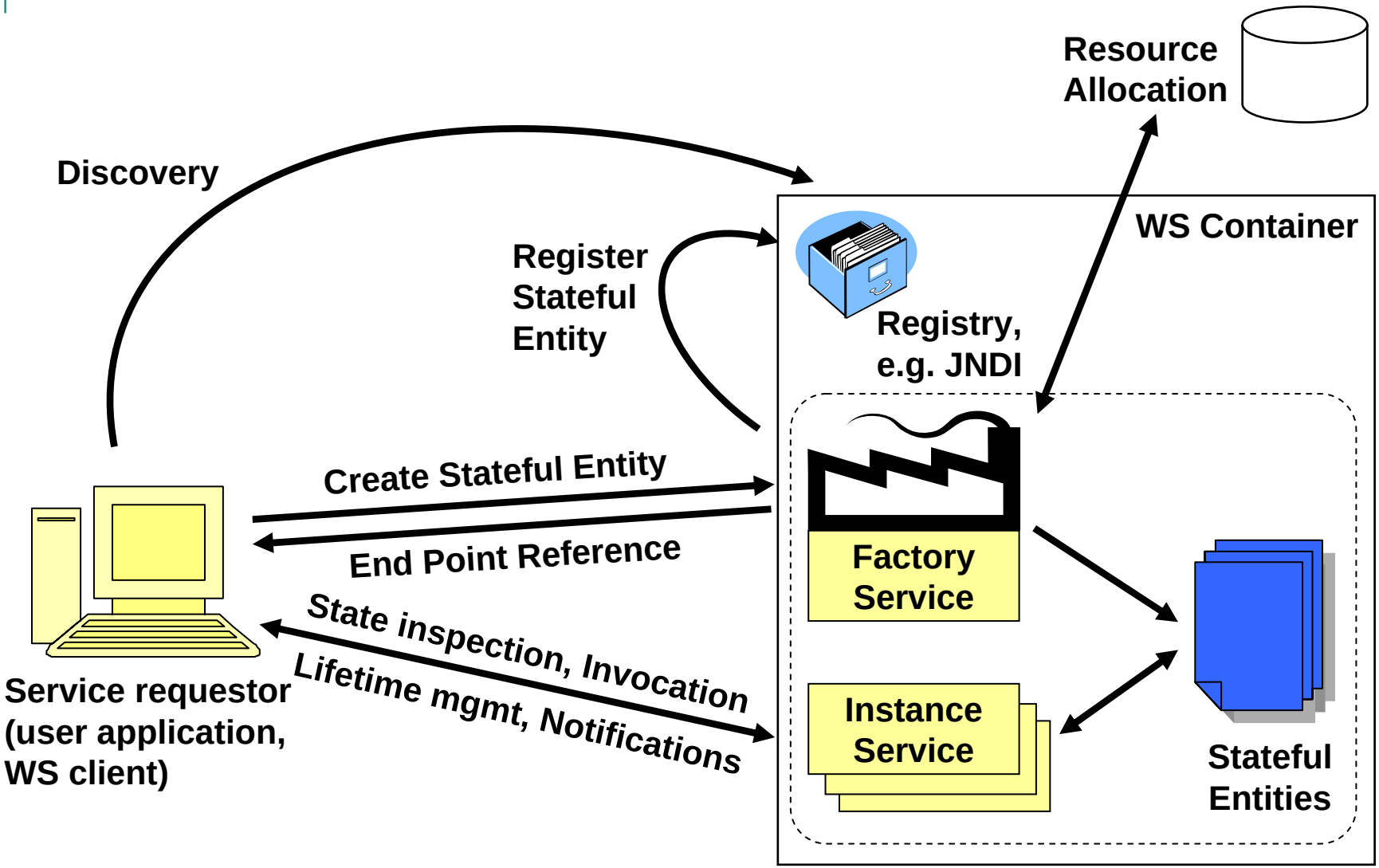
examples of existing implementations:

WS Container	Jakarta Tomcat, GT4 container
SOAP engine	Apache AXIS
Web server	Apache HTTP Server

Grid Services as Web Services

- stateless services, but stateful *resources*
  - factory and instance service
  - persistent state
  - life cycle (create, destroy)
  - properties (accessed via a stateless WS)
  - notification (on state changes)
- standardised in the Web Service Resource Framework
  - WS-Resource
  - WS-ResourceProperties
  - WS-ResourceGroup
  - WS-ResourceLifetime
  - WS-BaseFaults
- convergence into common WS specifications

Modeling State: Factory and Instance Service



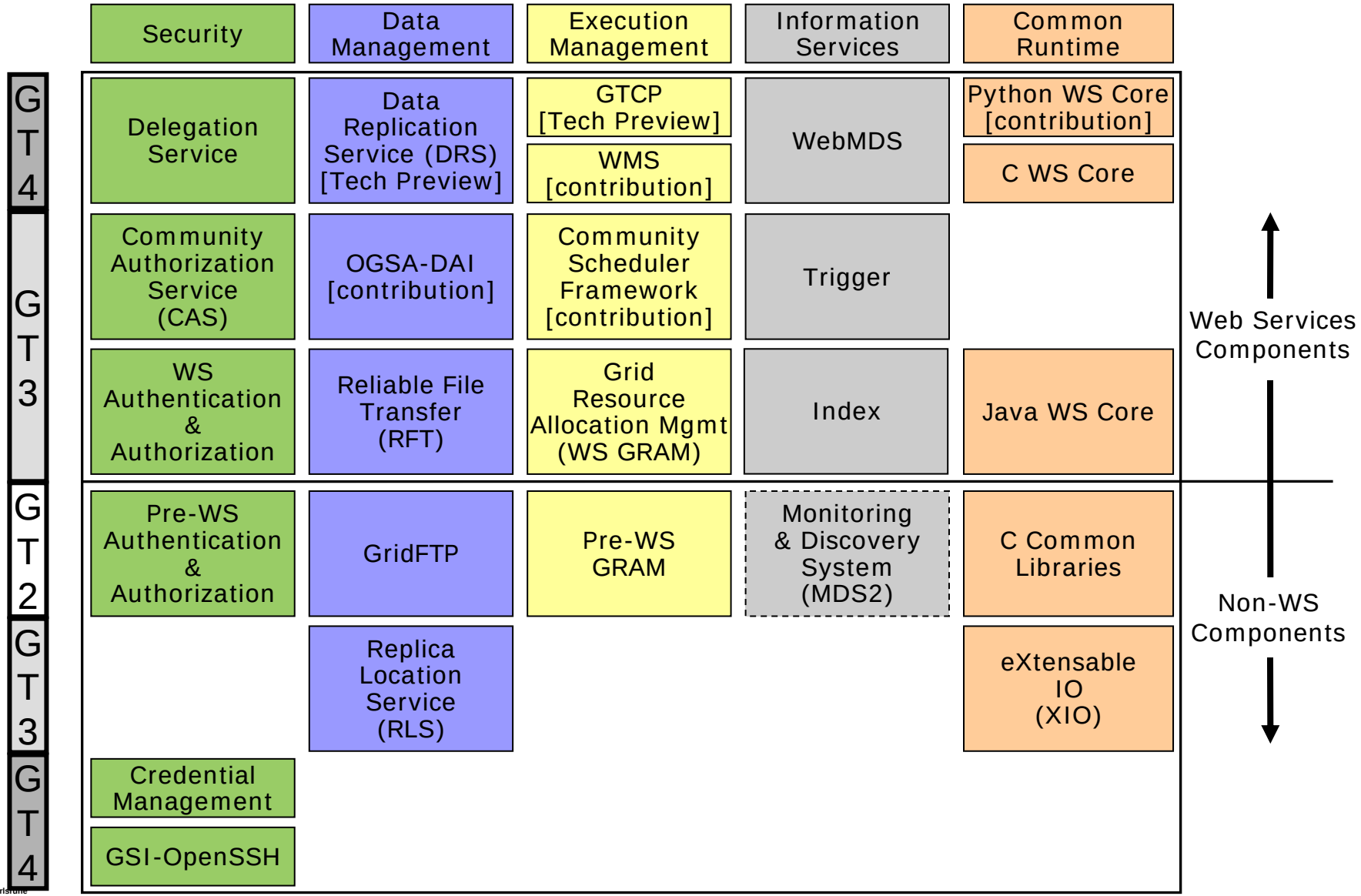
Outline

- introduction and basic concepts
  - What is GT4?
  - History
  - SOA, Web Services and WSRF
- GT4 components and architecture overview
  - main functions of a grid system
  - WS components versus Pre-WS components
  - some details on selected components
  - a typical use case
  - interoperation architecture

Basic Functions of a Grid System

- Security
  - Authentication and Authorization
  - Certificate Management
- Data Management
  - access and move data
  - replica management
  - storage brokering
- Execution Management
  - submitting and monitoring jobs
  - manage execution environments (sandboxes)
  - job brokering
- Information Services
  - resource and service discovery
  - monitoring the grid

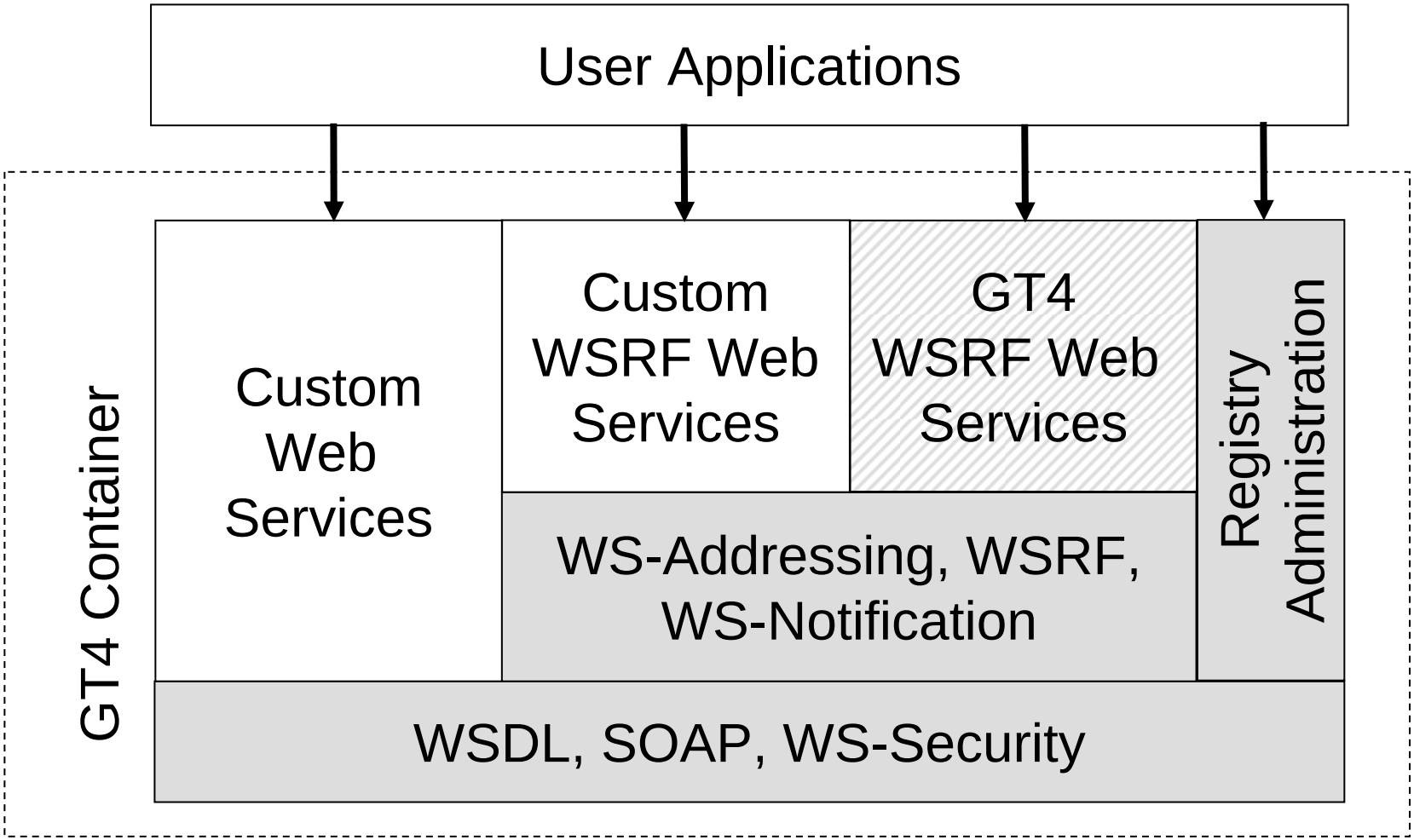
GT4 components



Common Runtime

- XIO: extensible input/output library, provides a single API for multiple protocols (TCP, UDP, HTTP, GSI_FTP, GSSAPI_FTP, TELNET, ...)
- WS core libraries
  - support both Globus services (GRAM, RFT, Delegation, etc.) & user-developed services
  - Leverages existing WS standards
    - WS-I Basic Profile: WSDL, SOAP, etc.
    - WS-Security, WS-Addressing
  - Adds support for emerging WS standards
    - WS-Resource Framework, WS-Notification
- Java, Python, & C hosting environments

GT4 Web Service Core



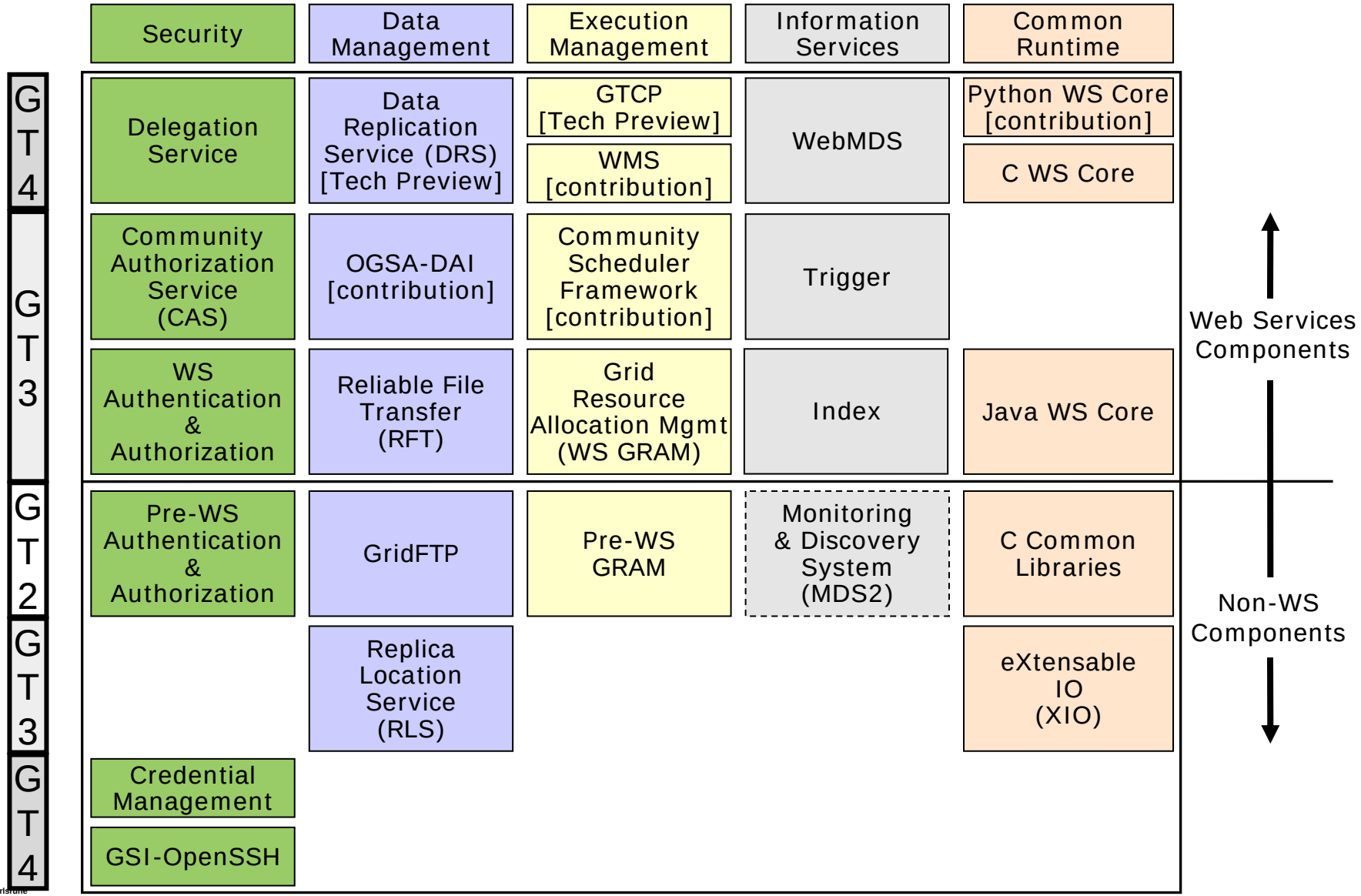
GT4 Web Service Container

- to start all Web Services (standard and user deployed), just start the container

- the WS container can run in two security modes
  - non-secured (http), standard port 8080
  - SSL secured (https), standard port 8443

- substitution by Tomcat possible (but has some drawbacks)

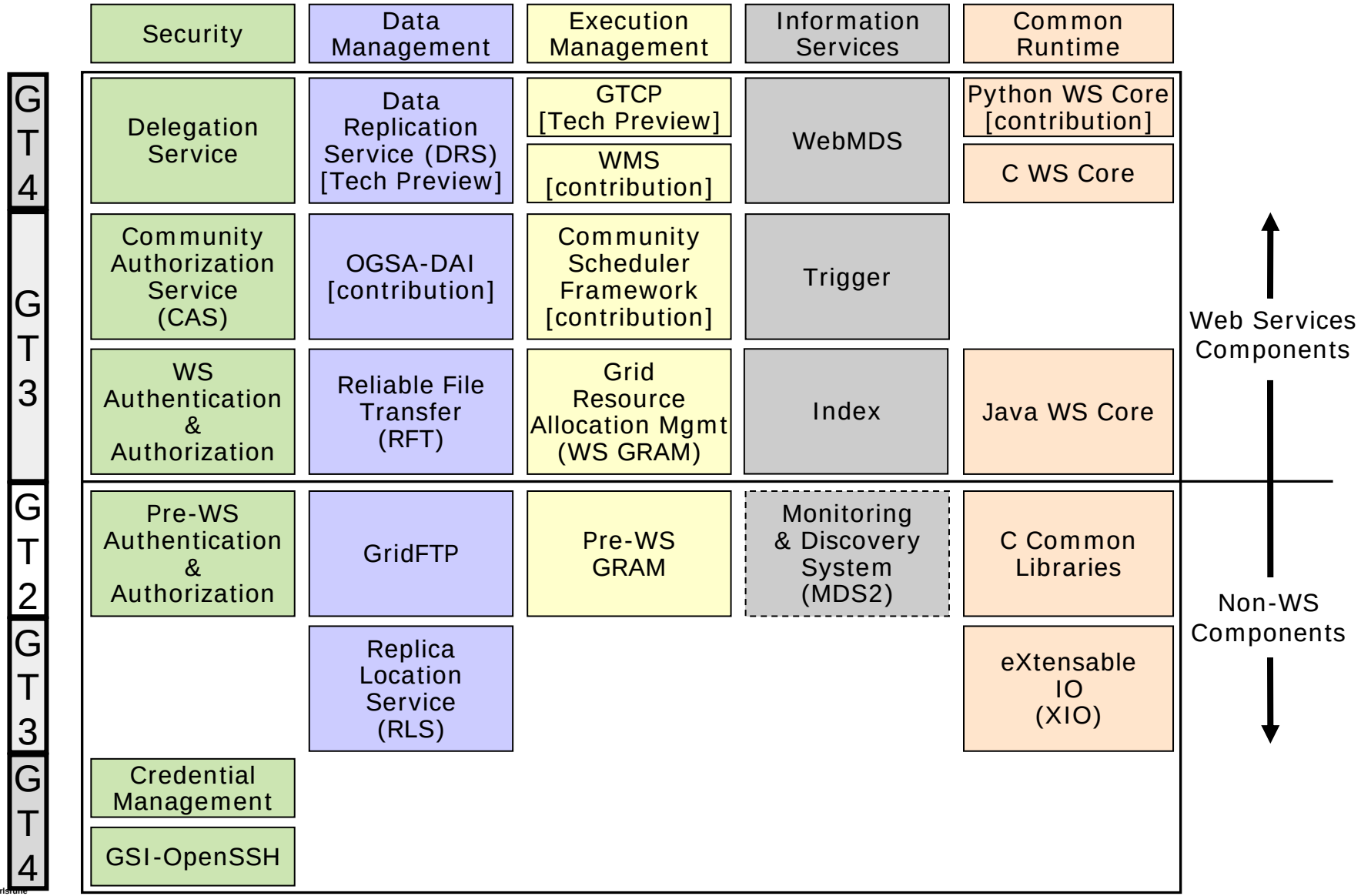
GT4 security



GT4 security

- Secured WS container
  - delegation service needed for service orchastration
- classical command line tools
  - grid-cert-request, grid-proxy-init, …
- Authorization uses Grid-Mapfile
  - statical mapping of cert. subjects onto UNIX accounts
  - VOMS integration only as Tech. preview
- Credential management services
  - MyProxy (Web portal integration!)
  - SimpleCA (poor man's credential generation)
  - Community Authorization Service
- GSI-OpenSSH
  - interactive shell access via Grid credentials

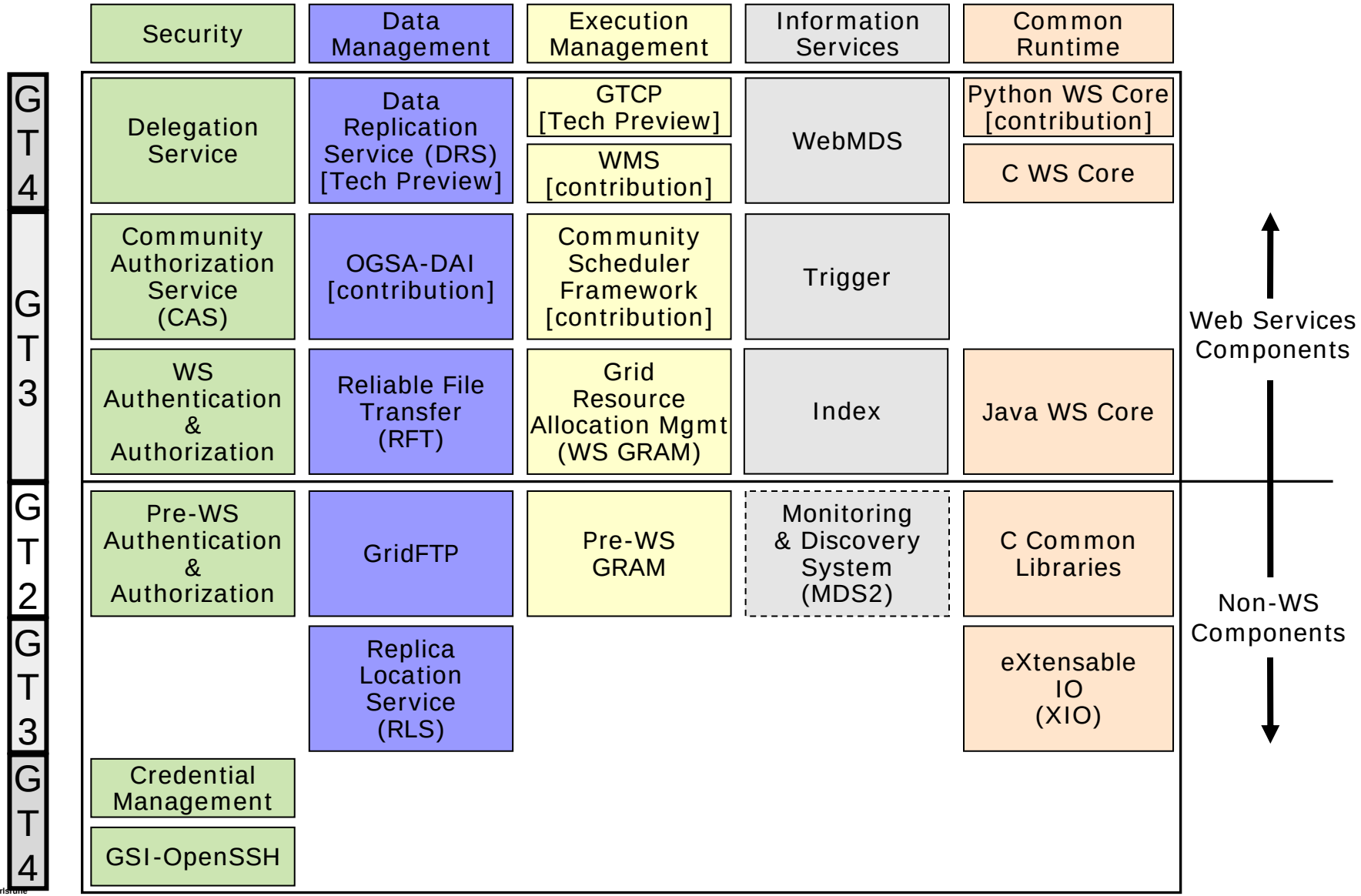
Information Services



Information Services

- GT2 Monitoring and Discovery Service is depricated.
- new WS-based implementation:
  - Index Service
    - aggregator service collects information from registered sources
    - to build an hierarchie of indexes (upstream or downstream)
    - automated registration of all services in the GT4 container
    - custom Ressource Property provider for non-WS resorces like batch systems (PBS, LSF, SGE) or monitoring tools (Ganglia)
    - access via Xpath queries (XML mapped GLUE schema)
  - Trigger Service
    - perform action on condition (event driven)
  - WebMDS
    - browser-based client for the index
    - adjustable via XSLT

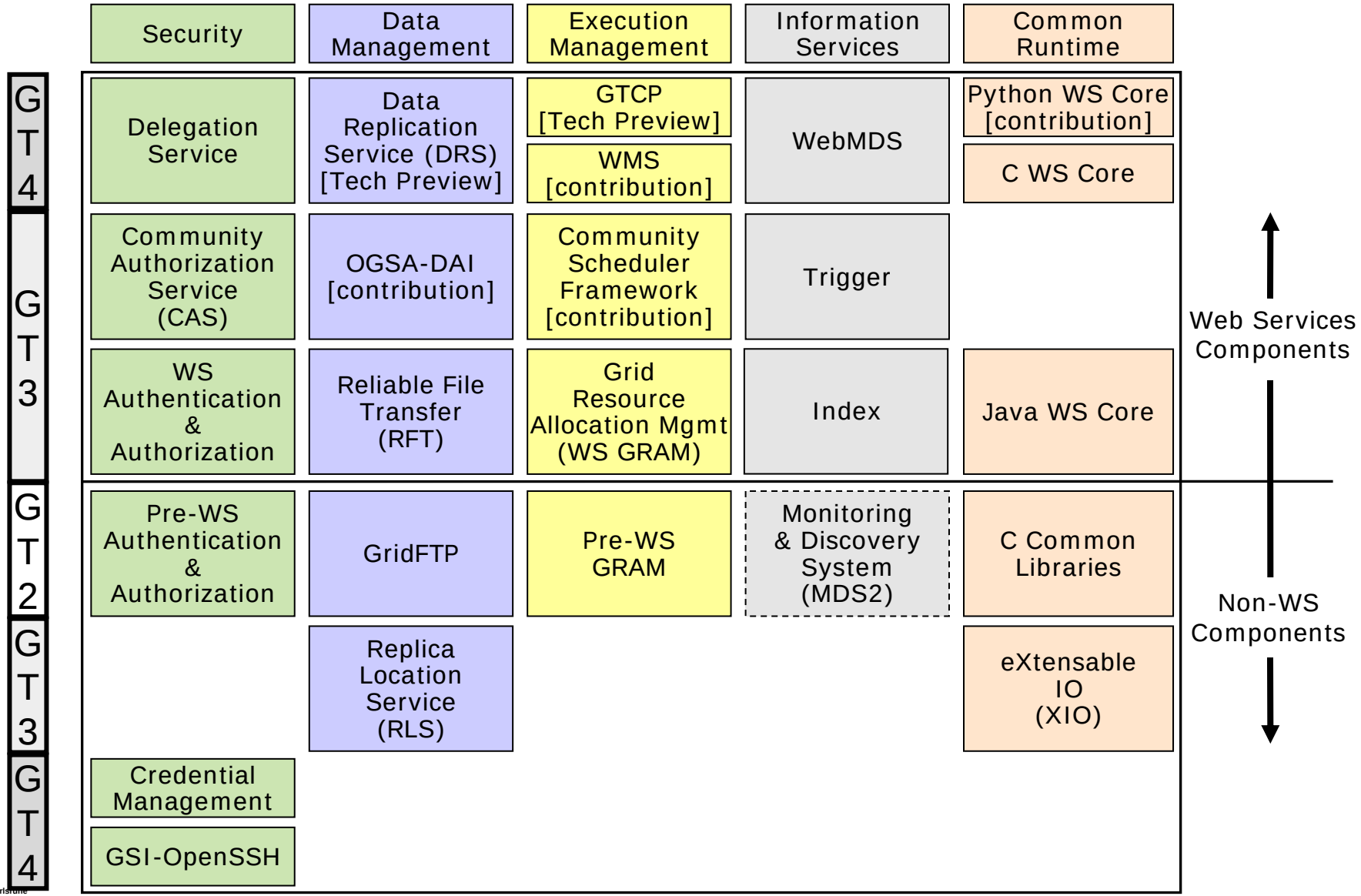
Data Management



Data Management

- GT4 GridFTP is a reimplementation of the older versions
  - 100% backward compatible
  - no licensing issues
  - access to data sources via XIO
- Reliable File Transfer (RFT)
  - WS interface for GridFTP
    - only GridFTP, no data transfer via SOAP messages
    - this requires an running GridFTP server with open ports on every host in the grid
  - recovery after broken transfer
- OGSA-DAI (data access interface)
  - provide service-based access to structured data resources
    - Relational: MySQL, Oracle, DB2, SQL Server, Postgres
    - XML: Xindice, eXist

Execution Management



Execution Management

- WS-GRAM
  - details on next slide

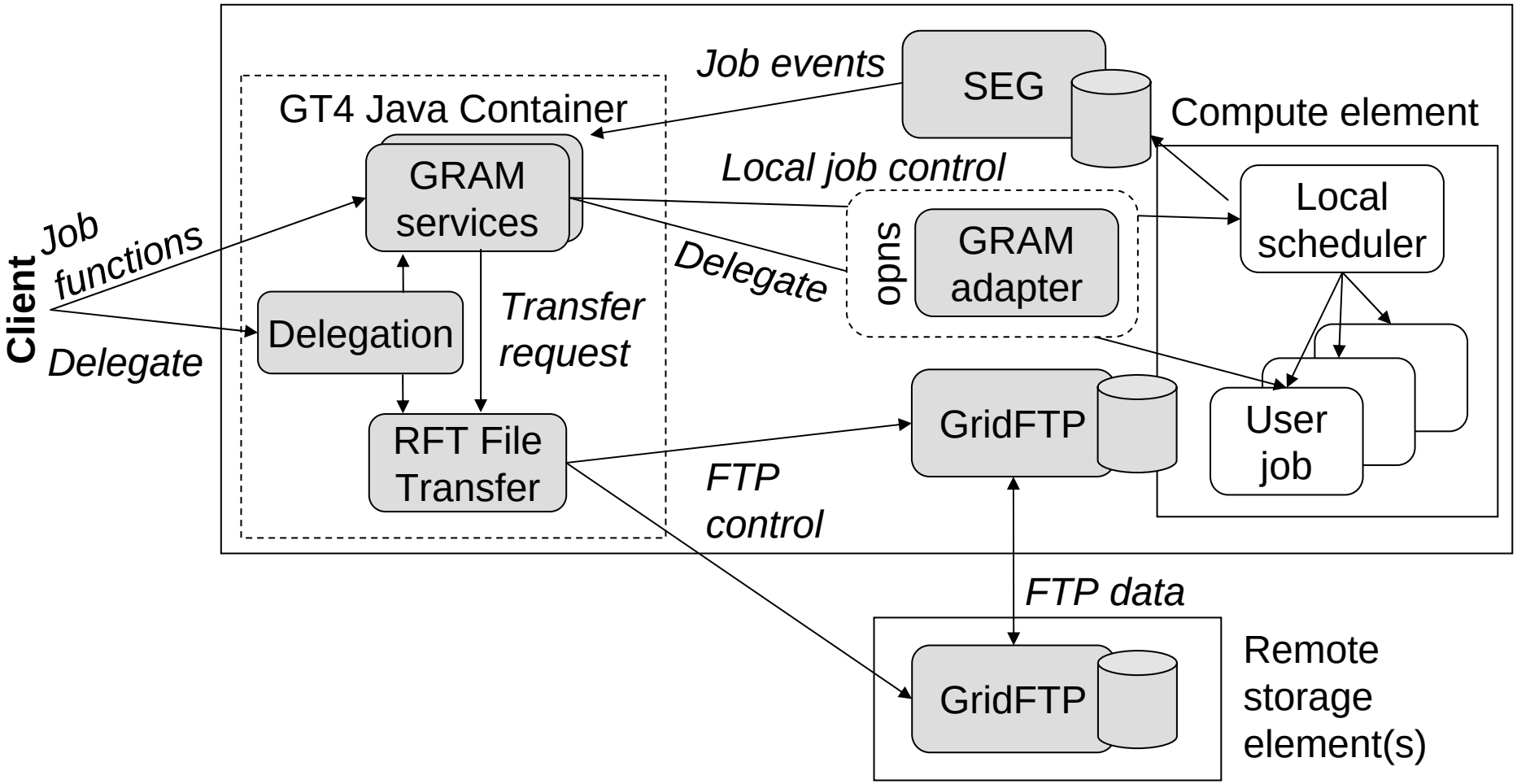
- Job Broker is not part of core GT4
  - several projects in the community with potential to fill this gap
  - the Community Scheduler Framework is one of them, but heavily biased to LSF

- other Tech. previews
  - Grid TeleControl Protocol (GTCP) service
    - for manageing instruments like microscopes
  - Workspace Management Service (WMS)
    - dynamic allocation of UNIX accounts (sandbox)
    - vitual machine support is prototyped

WS-GRAM

- independend of older GT2 GRAM
- provides a common WS interface to schedulers
  - UNIX-Fork, PBS, Condor, SGE, LSF, LoadLeveler
  - the scheduler actually wanted is transmitted to the WS-GRAM Factory Service as a parameter called FactoryType
- Job describtion in XML
  - Ressource Specification Language (RSL)
- uses Delegation Service for credential management
- uses RFT (and GridFTP) for data staging and streaming output
- command line client available but no GUI
  - this lack will probably filled by Web Portal Fremworks like GridShere

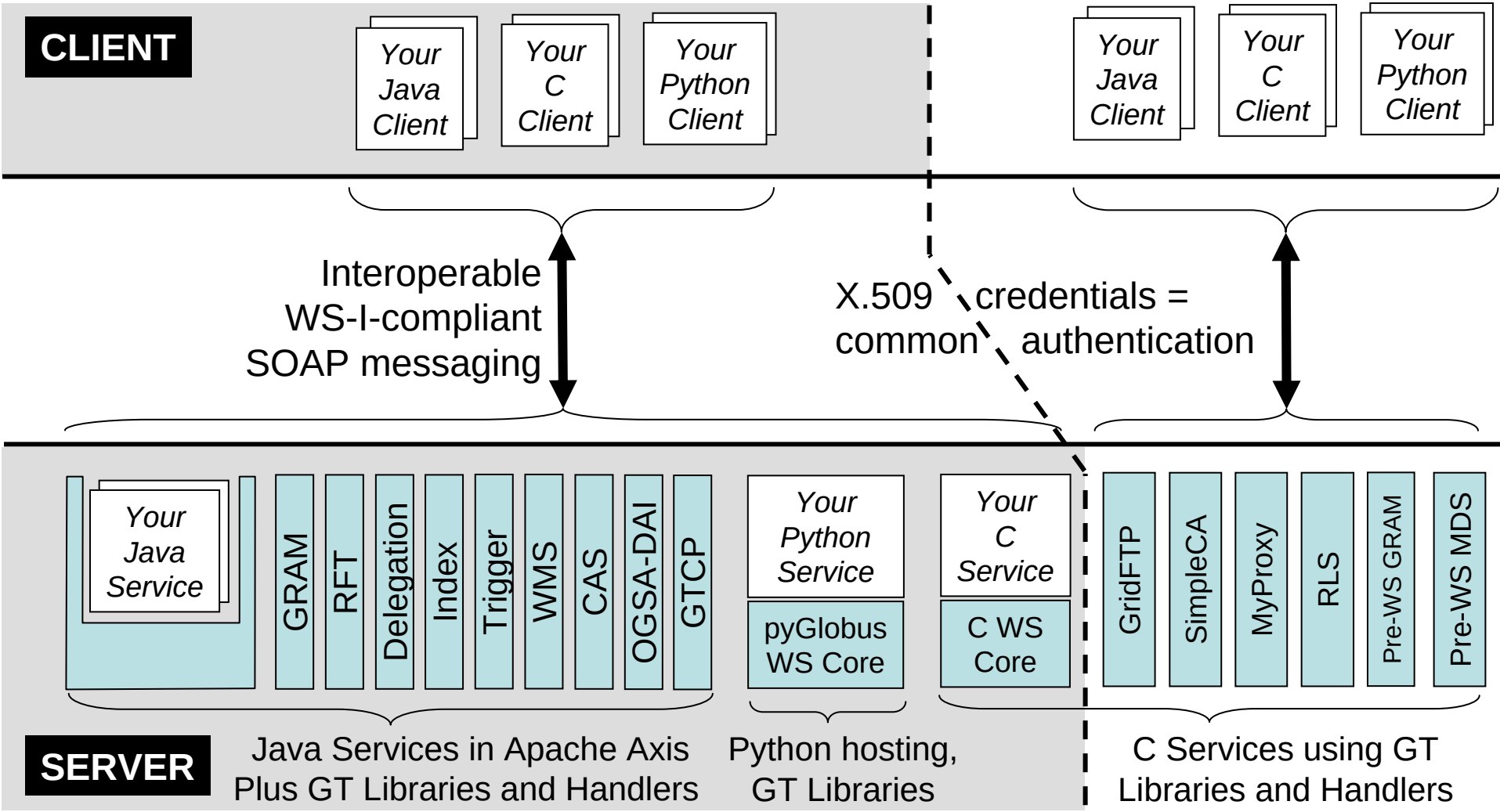
WS-GRAM


Job events
SEG
Compute element
GT4 Java Container
GRAM services
Local job control
Local scheduler
sudo
GRAM adapter
Client
Job functions
Delegate
Delegation
Transfer request
Delegate
GridFTP
User job
RFT File Transfer
FTP control
FTP data
GridFTP
Remote storage element(s)

A typical usecase for GT4

- situation:
  - you are running a compute cluster with several nodes
  - jobs are managed by a LRMS like PBS, Condor, SGE, …
  - shared file system over all nodes for user home directories
  - cluster management and user admin is done on a head node
- aim
  - integrate your cluster as computing resource in a Grid
- steps
  - install and configure GT4 on cluster head node
  - configure a WS-GRAM FactoryType for your batch system
  - start GT4 container and GridFTP-Server on the head node
  - get the Cert Subjects of all Grid users and add them to the Grid-Mapfile
  - optionally
    - allow access via GSI-OpenSSH to the head node
    - register your MDS-Index-Service into the global Index of your Grid

GT4 Components Architecture



[I. Forster: A Globus Primer]

Further Readings

- http://www.globus.org/toolkit/docs/4.0/
  - see also http://www.globus.org/toolkit/docs/development/
- I. Forster
  - A Globus Primer
    http://www.globus.org/toolkit/docs/4.0/key/GT4_Primer_0.6.pdf
  - Globus Toolkit 4: Software for Service-Oriented Systems
    NPC2005, LNCS 3779, pp. 2-13, 2005
  - http://www-fp.mcs.anl.gov/~foster/talks.htm
- B. Sotomayor
  The Globus Toolkit 4 Programmer's Tutorial
  http://gdp.globus.org/gt4-tutorial/

Thank you! Questions?