



Enabling Grids for
E-science in Europe

www.eu-egee.org

Induction Course, Edinburgh, 27 April 2004

NA3 Strategy for Future Grids

Malcolm Atkinson
Director of the National e-Science Centre



Contents

- What is the Web Service Resource Framework
 - **Why has it emerged?**
 - What has it to do with Grids?
 - **What are the parts of WSRF?**
 - What is the status of WSRF?
 - Standards process
 - Implementations
- Globus Alliance Plans
- WSRF in Perspective
 - What is important?



Reminder: what are *our* goals?

- Address *the* challenge
 - build inter-enterprise systems
- **NOT *just*** connecting systems
- **WORK** together to build infrastructure
 - That persistently and adaptively supports multiple Virtual Organisations
 - VOs that span organisational structures
 - Distributed implementation and operation
- Pioneering new ways of working
 - John Taylor's vision

We pioneer & transfer results to industry

**Easy tasks
can use any
technology**

**Only *some*
of our goals
align with
industry**

Why OGSI adopted web services

- Expectation: WS would meet several Grid needs
 - E.g. Standard interface definition language
 - Foundation for better engineering
 - E.g. Standard invocation mechanism
 - Foundation for interoperability
 - But other channels used for performance
 - Good commercial tooling (eventually)
 - Reliability and performance
- Service-Oriented Architecture
 - Has valuable scalability and durability properties
 - E.g. ICENI using Jini

Web Services components and framework

- Not a silver bullet or a complete solution!
 - Most of the engineering effort
 - What you do when you get a message
 - Not how you address, package and deliver it
 - Most of the standardisation effort
 - Agreeing how to factor large systems and the s
 - Agreeing conventions for information in message
 - Confusing & Rival standards proposals
 - Limited quality *public* implementations
- Don't give up – engage and help fix it?
 - Is this the role of EGEE?
 - Is there just one answer?
- Incremental adoption of WS-I, WS-Security, W
- Incremental adoption of WSRF as it emerges

We all agree that it is a good strategy to use web services. The issue are: Which ones to adopt when? and What conventions organise our systems?

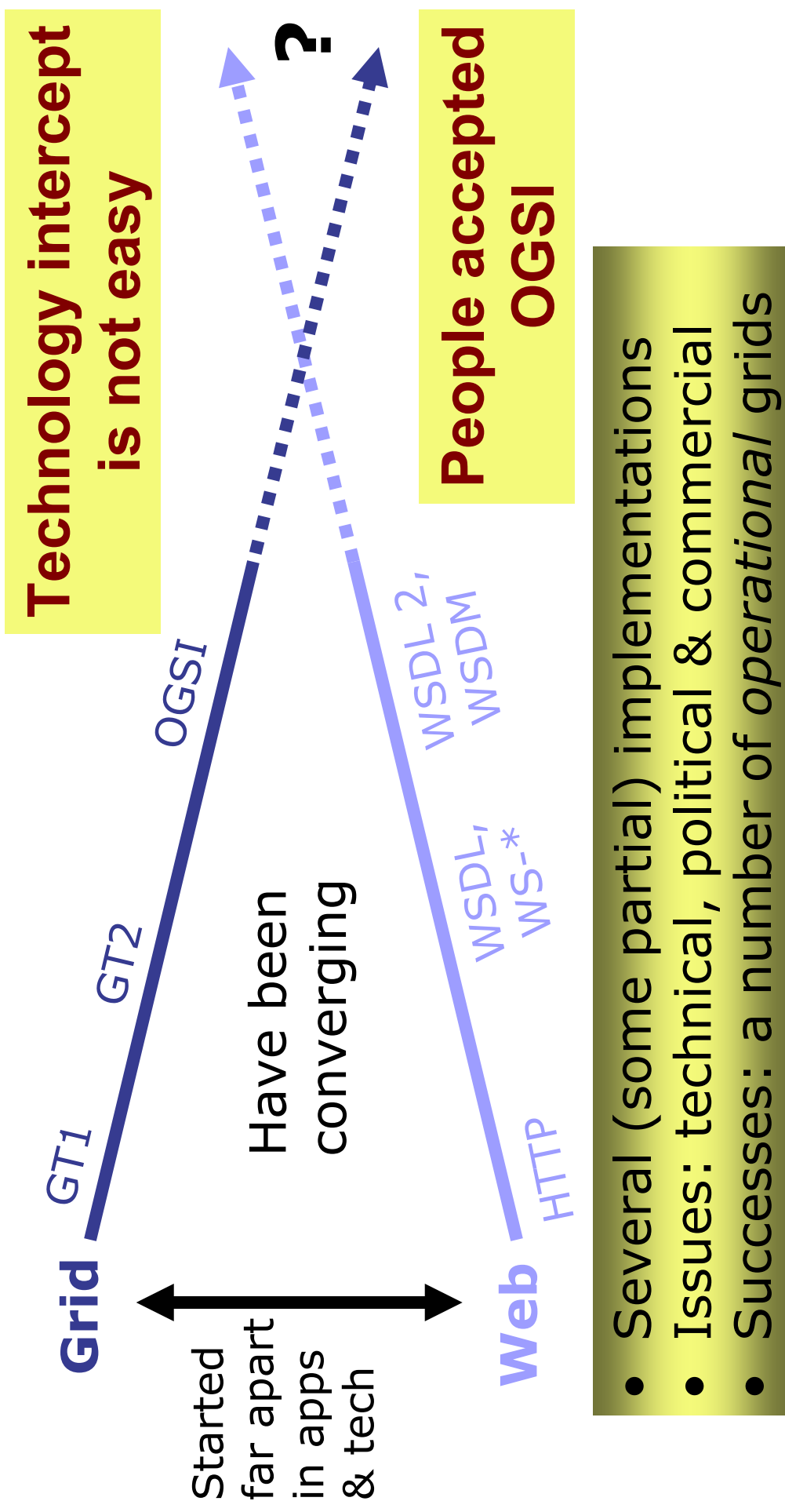
OGSI & GT3 investment

Three forms of investment

- Architectural effort
 - OGSA: Use cases, Design Patterns, ...
 - Factoring & describing a complex engineering
- Standardisation effort
 - OGSI, DAIS, WS-Agreement, etc.
 - WSDL 2.0, WSDM, WS-Security, etc.
- Implementation effort
 - Combined OGSI & Grid component work

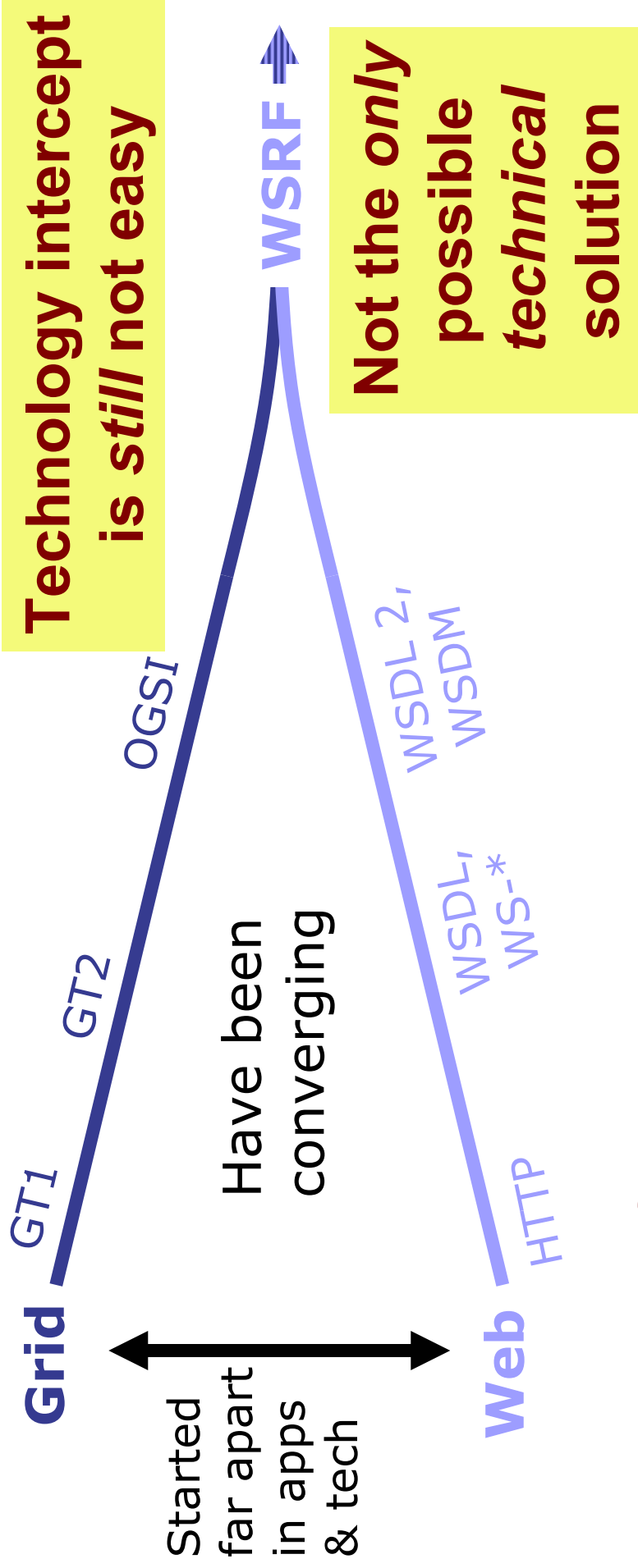
This investment carries forward into WSRF

Combining Grid and Web Services – First try



- Several (some partial) implementations
- Issues: technical, political & commercial
- Successes: a number of *operational* grids

Combining Grid and Web Services: Second try



**Support from major WS vendors
especially service management suppliers
e.g., CA, HP, IBM, Fujitsu, BEA, SAP, ...**

Core Ideas in WSRF

- Preserves OGSI functionality
 - Lifetime, properties, notification, error types, ...
- Separates service from resource
 - Service is static and stateless
 - Resource is dynamic and stateful
- Builds on WS-Addressing
- Is WS-I compliant
 - But note that WS-I alone doesn't make the problems go away, still need to worry about how to manage lifetime, naming, state, ...

“Components” of WSRF

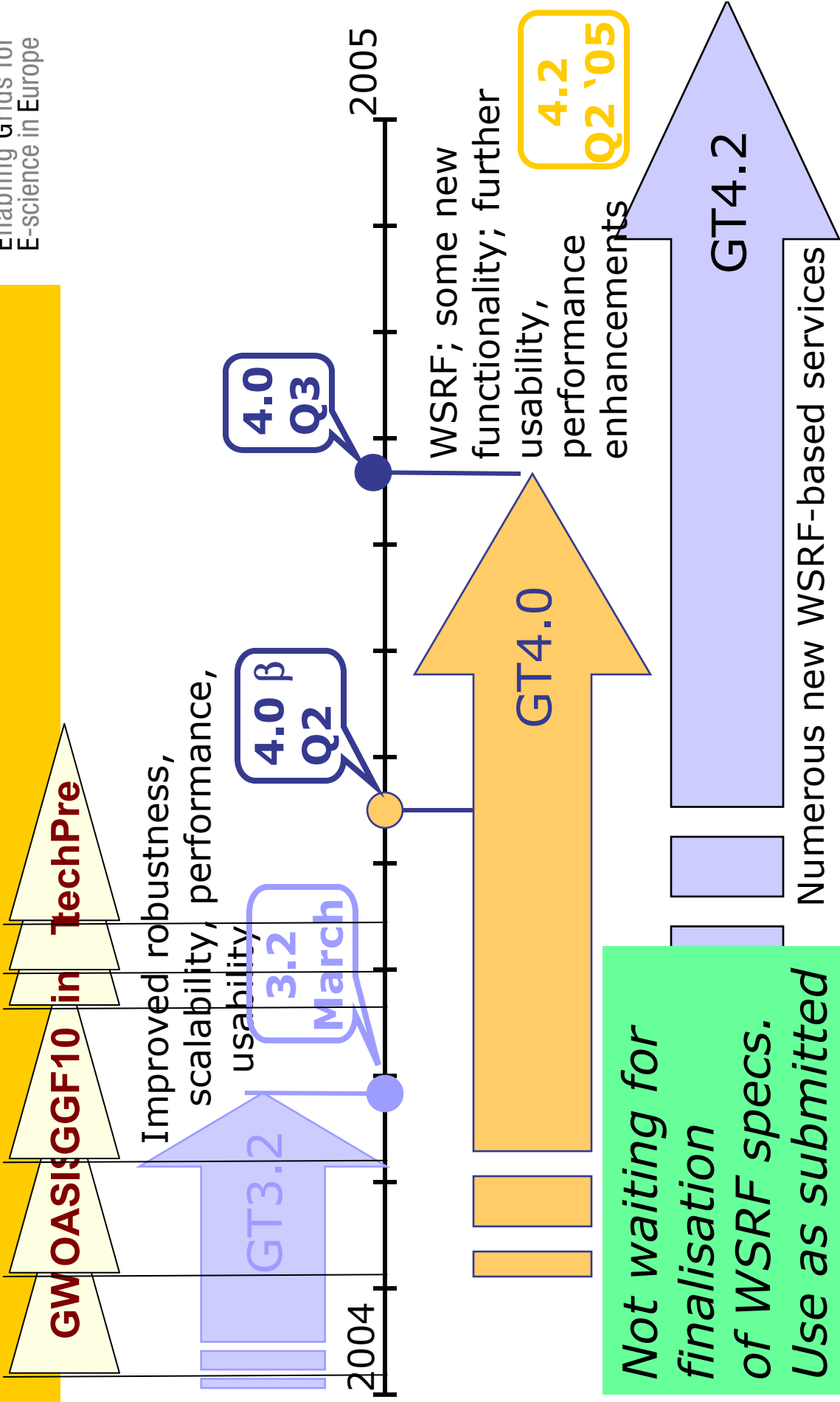
WS-Addressing	March 04	
		www-106.ibm.com/developerworks/library/specification/ws-add/
WSRF		
White paper on modelling stateful resources		www.globus.org/wsrf/
WS-ResourceLifetime	March 04	www.globus.org/wsrf/
WS-ResourceProperties	March 04	www.globus.org/wsrf/
WS-BaseFaults	March 04	www.globus.org/wsrf/
WS-RenewableReferences	March 04	www.globus.org/wsrf/
WS-ServiceGroup	March 04	www.globus.org/wsrf/
WS-Notification		
WS-BaseNotification	March 04	.../specification/ws-notification/
WS-Topics	March 04	.../specification/ws-topics/
WS-BrokeredNotification	March 04	.../specification/ws-pubsub/

From OGSI to WSRF: Refactoring and Evolution

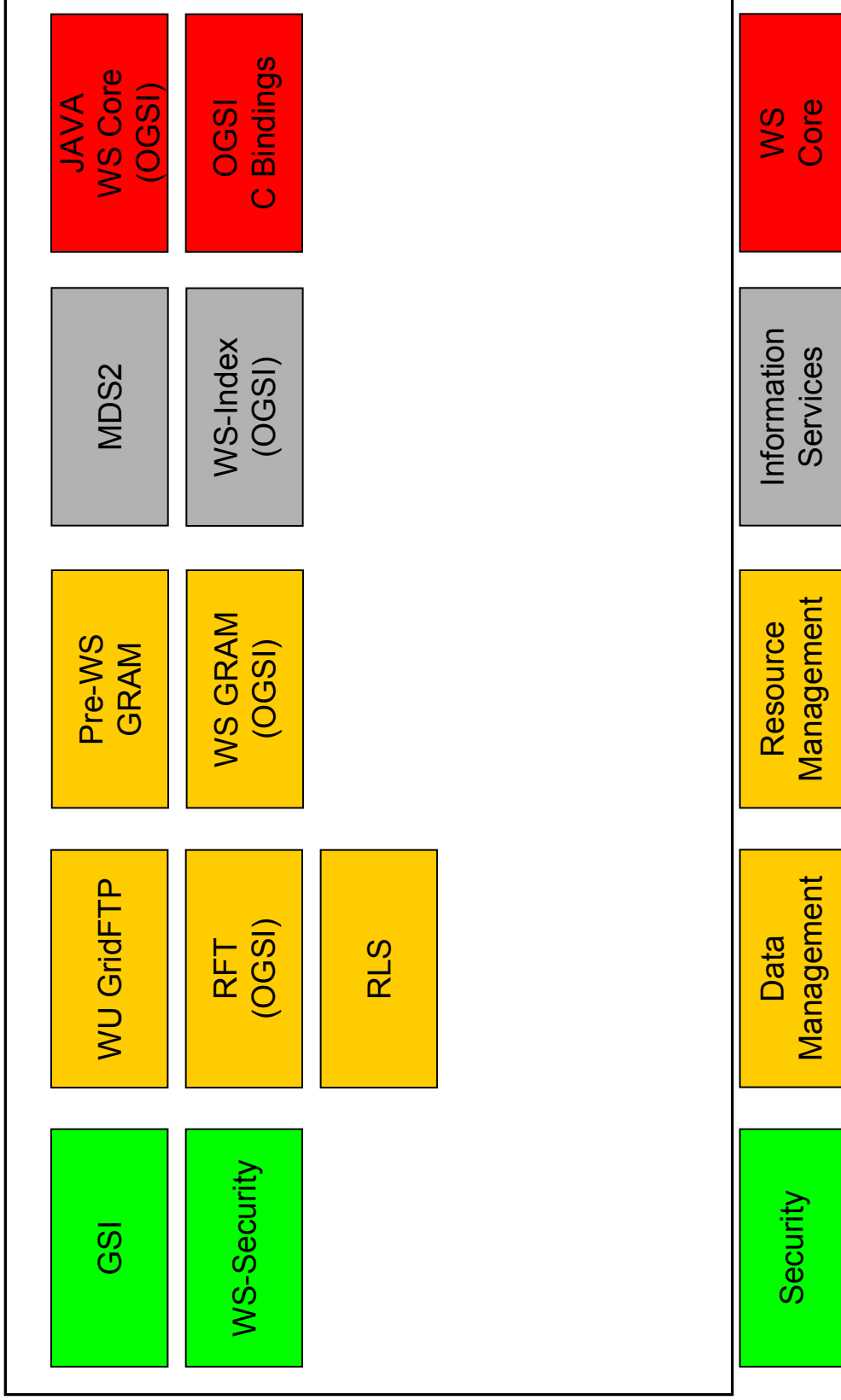
Identity & naming is being done by OGSA

OGSI	
Grid Service Reference	WS-Addressing
Grid Service Handle	WS-Addressing Extension
HandleResolver portType	WS-RenewableReferences
Service data defn & access	WS-ResourceProperties
GridService lifetime mgmt	WS-ResourceLifetime
Notification portTypes	WS-Notification
Factory portType	Treated as a pattern
ServiceGroup portTypes	WS-ServiceGroup
Base fault type	WS-BaseFaults

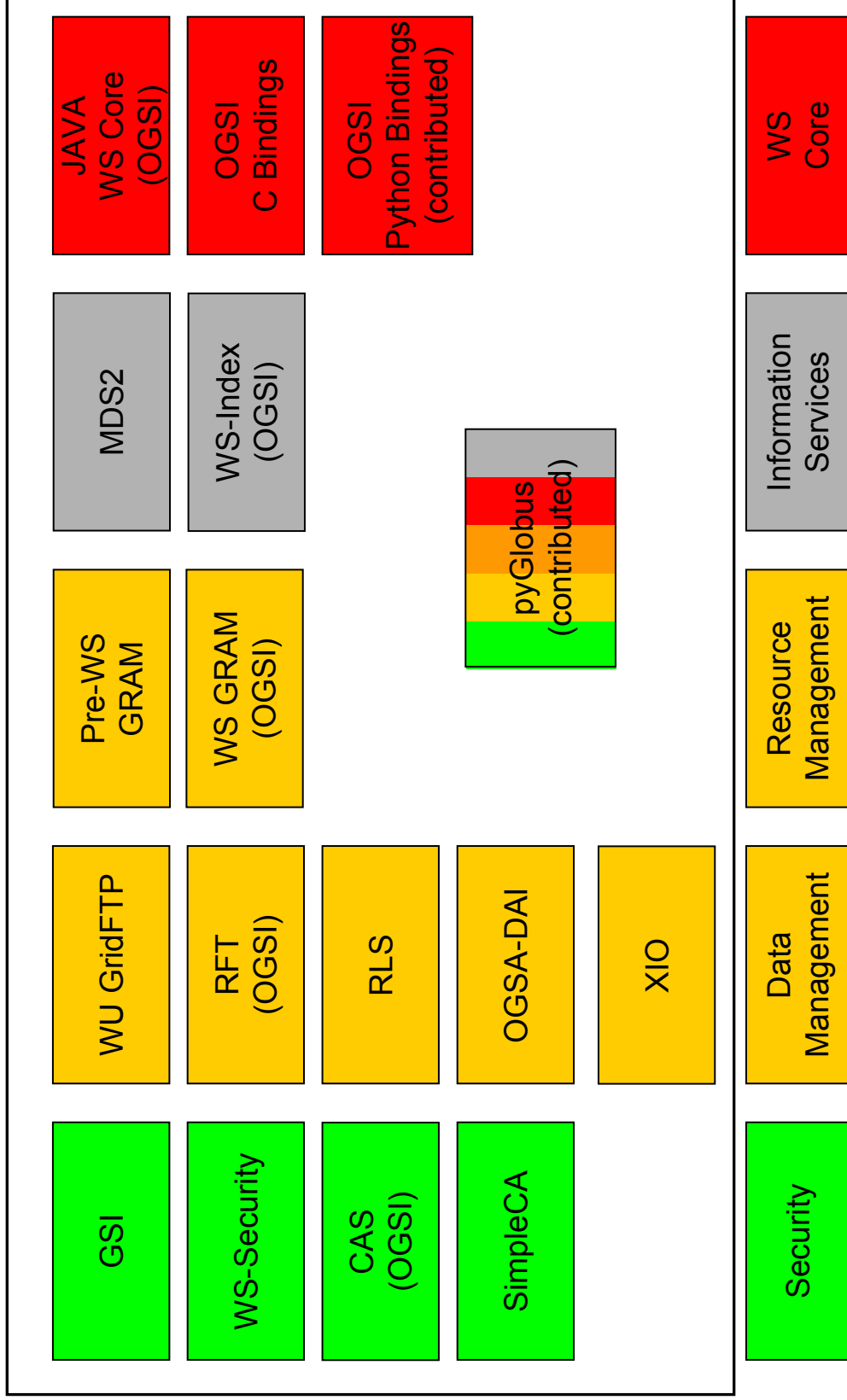
GT & WSRF Timeline



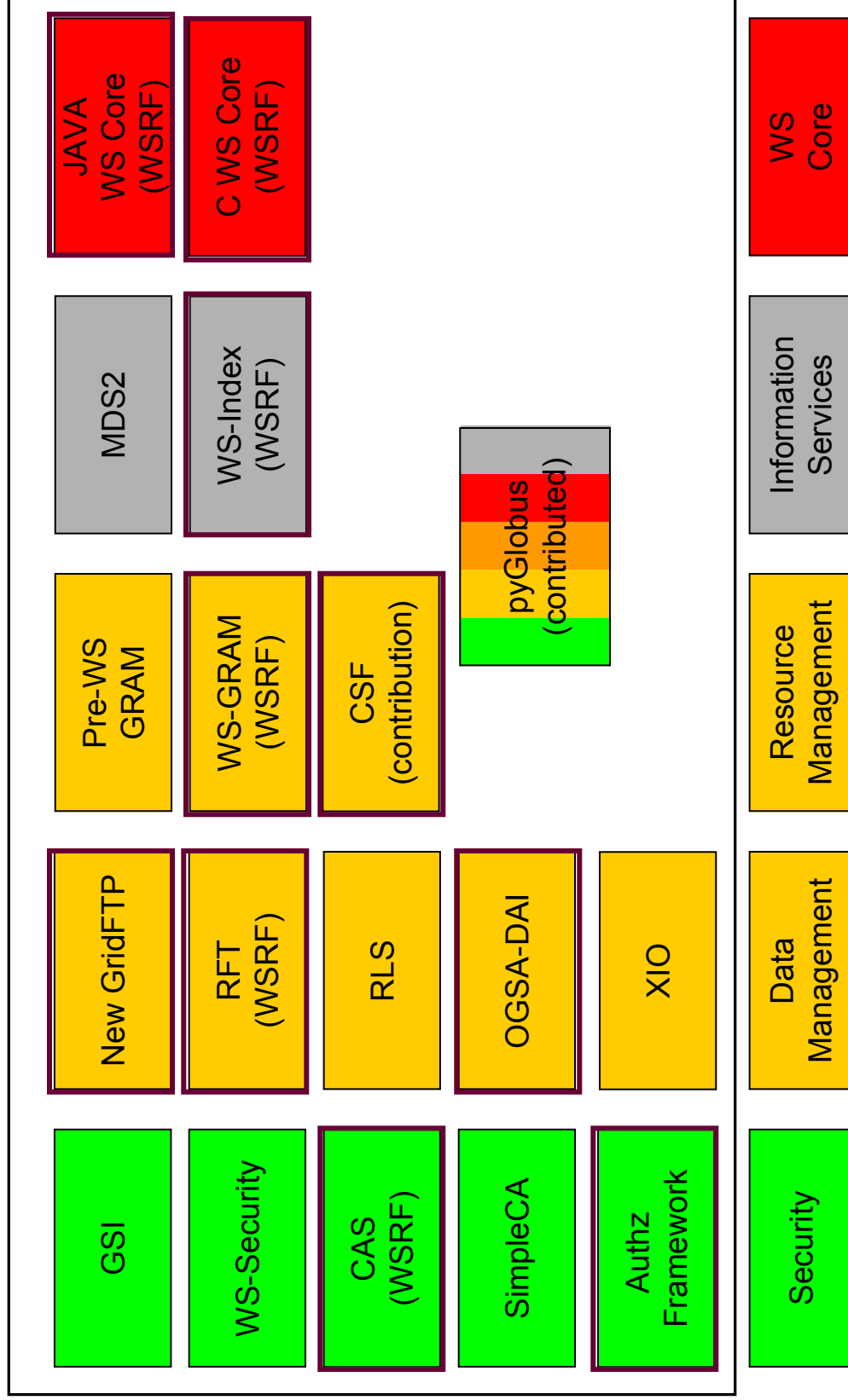
Components in GT 3.0



Components in GT 3.2



Planned Components in GT 4.0



Importance of collaboration: VDT

- A highly successful collaborative effort

- VDT Working Group
- VDS (Chimera/Pegasus) team
 - Provides the “V” in VDT
- Condor Team
- Globus Alliance
- NMI Build and Test team
- EDG/LCG/EGEE
 - Middleware, testing, patches, fe
- PPDG
 - Hardening and testing
- Pacman
 - Provides easy installation capab
 - Currently Pacman 2, moving to Pacman 3 soon

Used by many projects
Systematic testing
Rich integration of
components
EGEE is part of this –
exploit test bed
contribute components

Thanks to Miron Livny

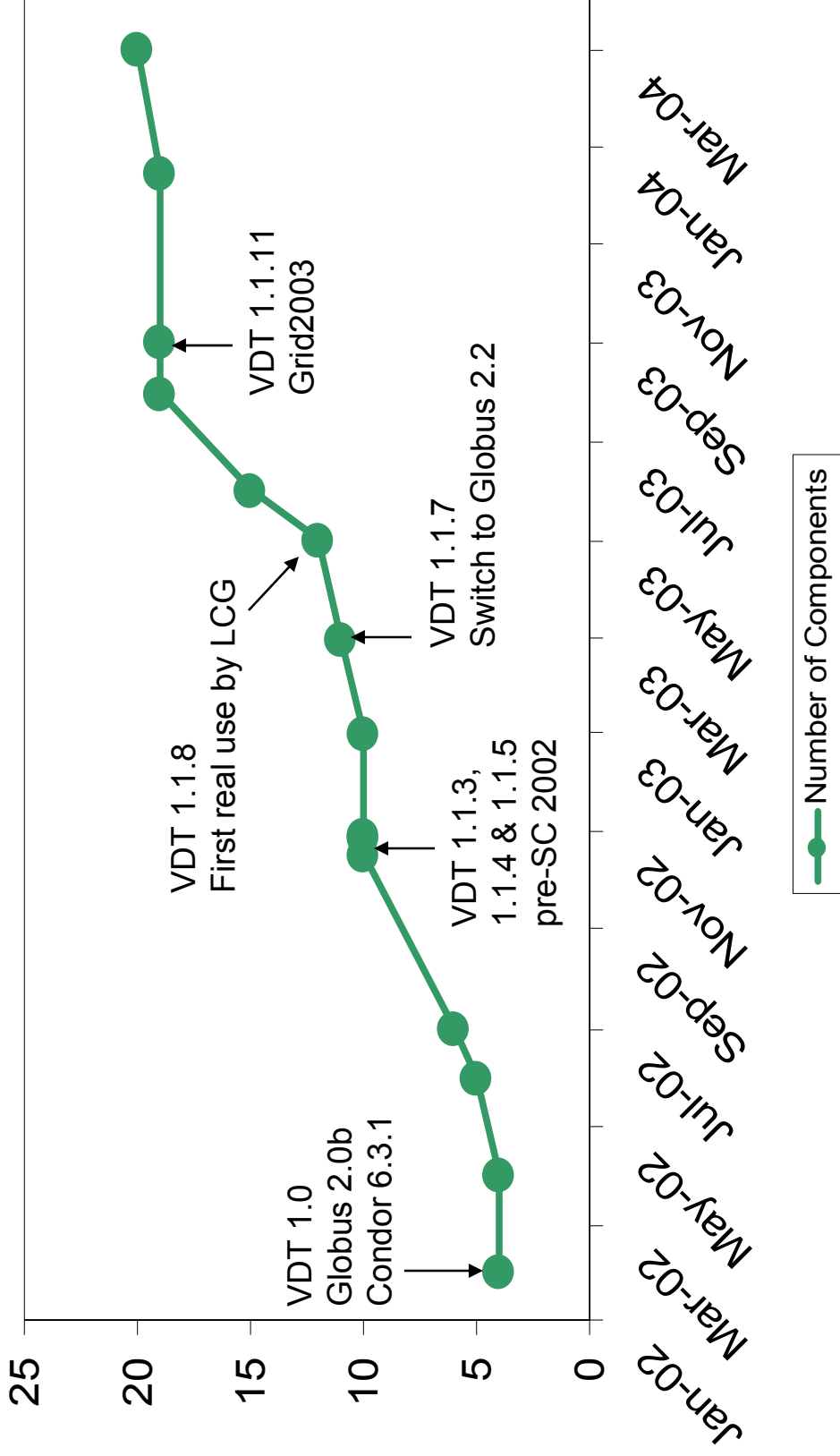
Collaboration is a two way street – or should be!

VDT Timeline Highlights

- **Fall 2001:** VDT started by GriPhyN and iVDGL
 - Supported US-CMS testbed in 2002
- **March 2002:** VDT support system inaugurated
- **Early 2003:** Adopted by **European Data Grid & LHC Computing Grid**
- **April 2003:** VDT Testers group started
- **Fall 2003:** Supporting Grid3
- **Fall 2003:** Adopted by **Particle Physics Data Grid**
- **Nov 2003:** Nightly test infrastructure deployed

Thanks to Miron Livny

VDT Growth



Thanks to Miron Livny

Tools in the VDT 1.1.13

- Condor Group
 - Condor/Condor-G
 - DAGMan
 - Fault Tolerant Shell
 - ClassAds
- Globus Alliance
 - Job submission (GRAM)
 - Information service (MDS)
 - Data transfer (GridFTP)
 - Replica Location (RLS)
- EDG & LCG
 - Make Gridmap
 - Certificate Revocation List Updater
 - Glue Schema/Info prov.
- ISI & UC
 - Chimera & Pegasus
- NCSA
 - MyProxy
 - GSI OpenSSH
 - UberFTP
- LBL
 - PyGlobus
 - Netlogger
- Caltech
 - MonaLisa
- VDT
 - VDT System Profiler
 - Configuration software
- Others
 - KX509 (U. Mich.)

Thanks to Miron Livny

Relative Importance

- What envelopes you put your messages in
 - How they are delivered
 - Infrastructure to organise a common technical platform – the foundations of communication

Relative Importance

- What envelopes you put your messages in
 - How they are delivered
 - Infrastructure to organise a common technical platform – the foundations of communication
- What information you send in your messages
 - Their patterns of Use - sequences that mean something
 - Their Contents
 - The Grammar and Vocabulary of Communication
 - Agreed Interpretations

Relative Importance

- What envelopes you put your messages in
 - How they are delivered
 - Infrastructure to organise a common technical platform – the foundations of communication
- What information you send in your messages
 - Their patterns of Use - sequences that mean something
 - Their Contents
 - The Grammar and Vocabulary of Communication
 - Agreed Interpretations
- What you do when you get a message
 - The Application Code you Execute
 - The Middleware Services
 - Security, Privacy, Authorisation, Accounting, Registries, Brokers, ...
 - Integration Services
 - Multi-site Hierarchical Scheduling, Data Access & Integration, ...
 - Portals, Workflow Systems, Virtual Data, Semantic Grids
 - Tools to support Application Developers, Users & Operations
 - Incremental deployment tools, diagnostic aids, performance monitoring, ...

Technical Experts

Relative Importance

- What envelopes you put your messages in
 - Instructions to organise a communication platform - the foundations of communication
 - How they are delivered
- What information you send in your messages
 - Their patterns of Use - sequences that mean something
 - Their Contents
 - The Grammar and Vocabulary of Communication
 - Agreed Interpretations
- What you do when you get a message
 - The Application Code you Execute
 - The Middleware Services
 - Security, Privacy, Authorisation, Accounting, Registries, Brokers, ...
 - Integration Services
 - Multi-site Hierarchical Scheduling, Data Access & Integration, ...
 - Portals, Workflow Systems, Virtual Data, Semantic Grids
 - Tools to support Application Developers, Users & Operations

• Creative Actions and Judgements of Researchers, Designers & Clinicians

- Data, Models & Analyses
- In Silico Experiments, Design, Diagnosis & Planning
- Creating the Scientific Record

Domain Experts

Conclusions - Strategy

- If you are making long-term plans
 - Plan to use WSRF
- If you develop or research middleware
 - Engage with groups developing WSRF
 - If you need those or related functions
 - E.g. to notify or handle state without incremental resource loss
- If you run distributed Grid operations
 - Plan to use WSRF
 - But only when components using it are robust
 - Incremental transition is possible – even necessary

Conclusions - Tactics

- Value your team's skills & momentum
- Change only if necessary
- If you're an applications researcher
 - Stay with what you have working
 - WS-I +GSI; GT2, VDT, LCG2, GT3, ...
- If you're a computing researcher
 - If your platform serves your investigation stay on it
 - WS-I +GSI; GT2, VDT, LCG2, GT3, ...
- If you're doing middleware R&D
 - Hard choices & frustrating times – keep going
 - Those who understand the new order will reap advantage
 - Therefore engage with WSRF
- WSRF principles and design patterns
 - Are a useful guide to building distributed infrastructure
 - Adopt them, even while WSRF details are being agreed
 - To ease transition

Conclusions - Final

- WSRF
 - Good enough for recurrent platform requirements
 - Has significant commercial and technical momentum
 - Improves engagement with industry
 - Only sensible flag to rally behind
- Must collaborate internationally
 - Scale of challenge & international virtual organisations
- Discourage localised alternatives
 - Avoid effort fragmentation and unnecessary arguments
- Coping well with transitions ...

Is a primary Darwinian selector!