UNICORE SUMMIT 25th August 2009, Delft, Netherlands

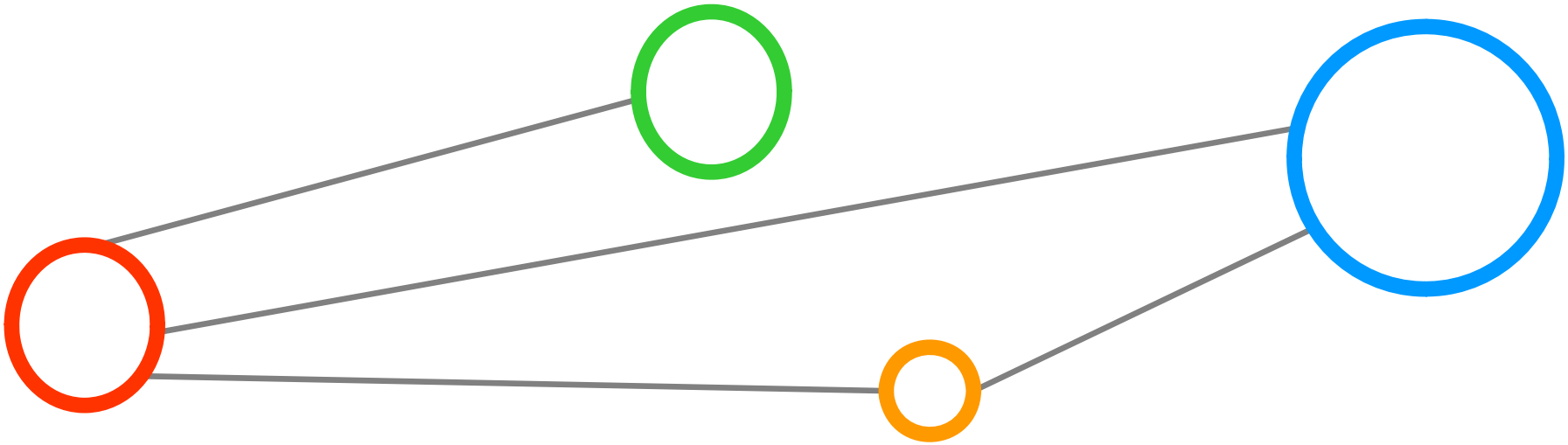
OpenGridForum
OPEN FORUM | OPEN STANDARDS

Interoperability of Production e-Science Infrastructures

Taking Lessons Learned into Standardization

Morris Riedel
Group Co-Chair Grid Interoperation Now & Production Grid Infrastructure

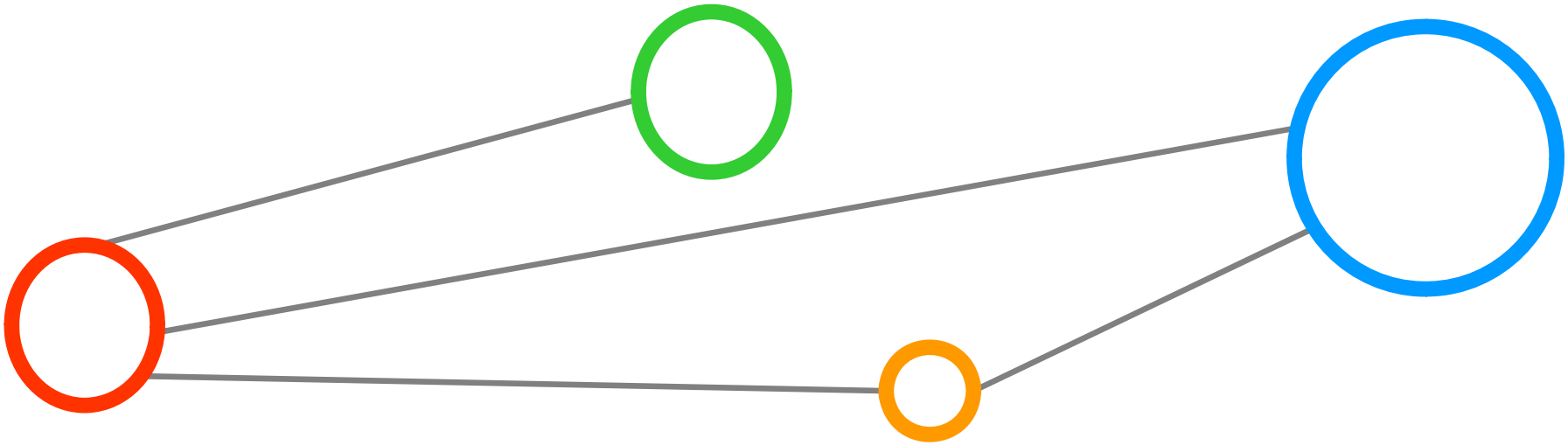
Outline



Outline

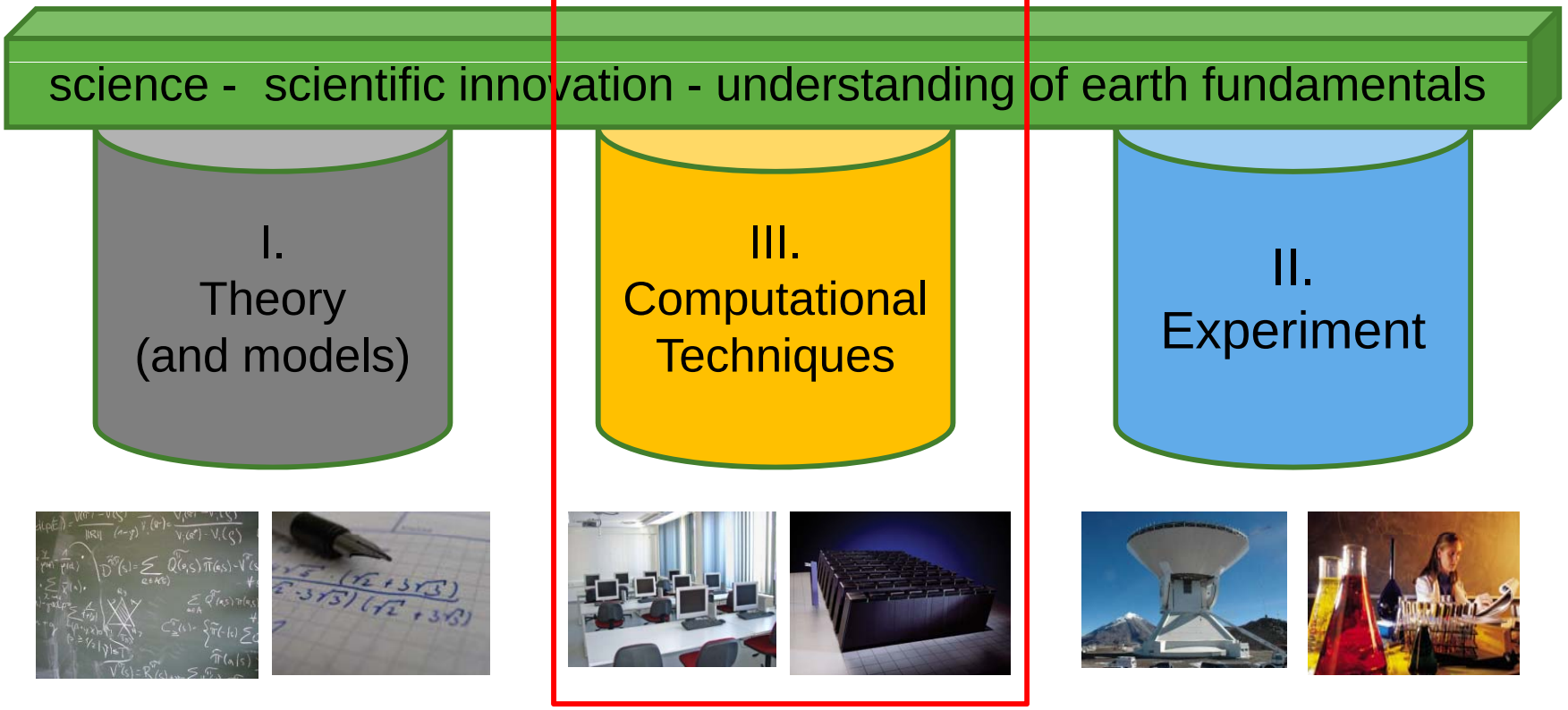
- e-Science
- Motivation for Interoperability
- Emerging Open Standards
- Interoperability Reference Model
- Computing Refinement Concepts
- Other Refinement Concepts
- Conclusion

Motivation for Interoperability



Traditional Scientific Computing

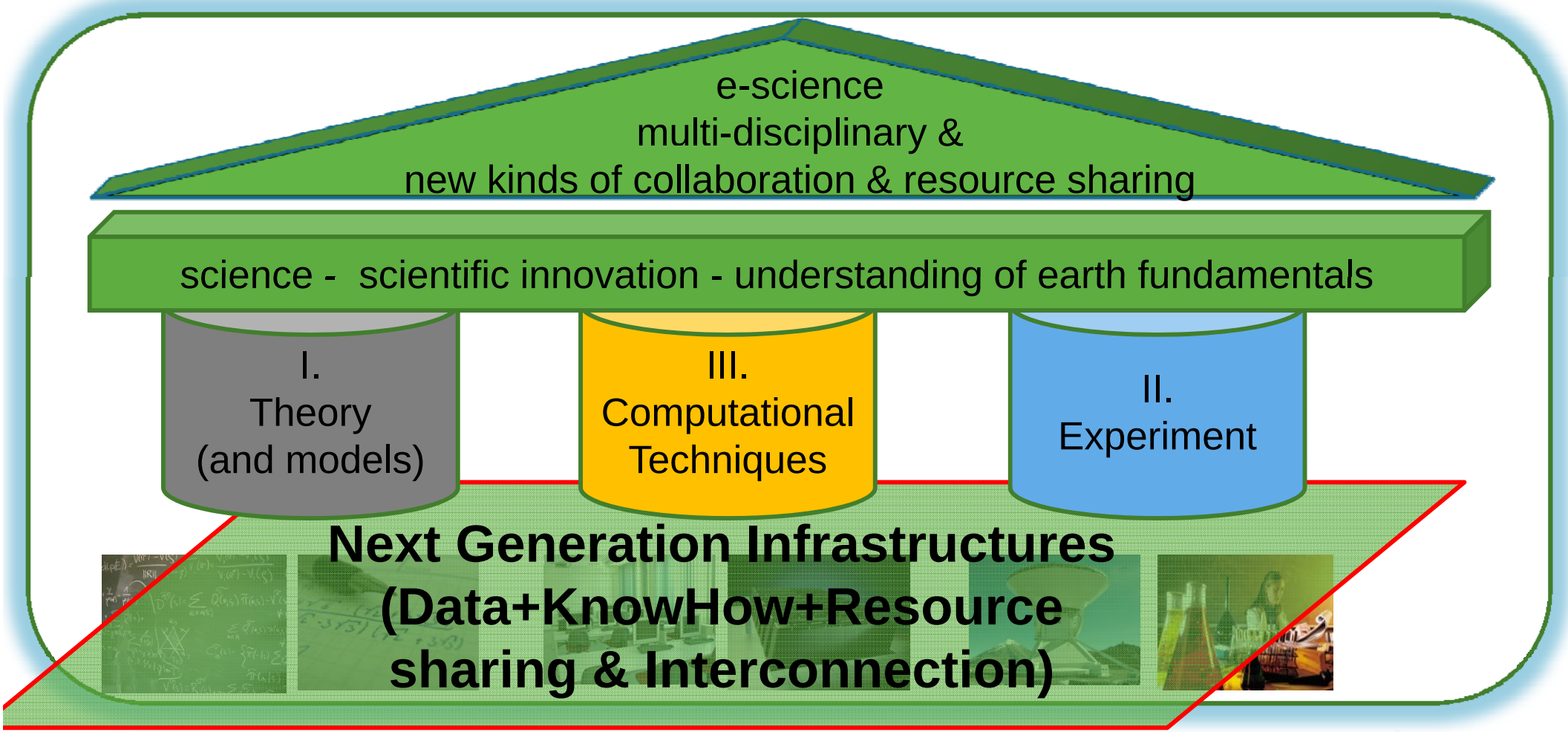
*'Today, the natural sciences regard **computational techniques** as a third pillar alongside experiment and theory'*



[1] Lippert et al., Interview *'Europe's biggest supercomputer'*

Enhanced Science (e-Science)

*'e-Science is about global collaboration in key areas of science and the **next generation infrastructure** that will enable it'*



[2] John Taylor, *'The definition of e-science'*

Production Grid Infrastructures

High Throughput Computing (HTC) Infrastructures

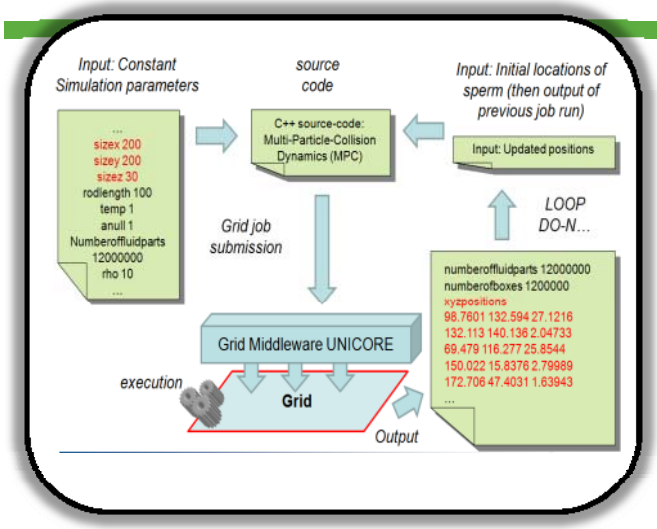
High Performance Computing (HPC) Infrastructures

Different Technologies

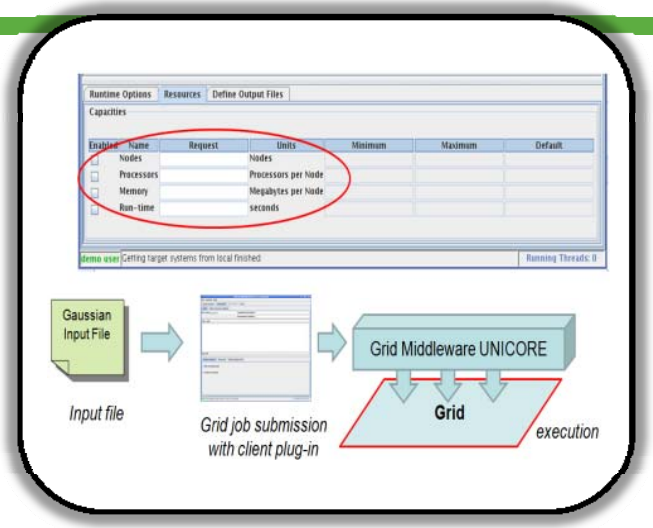
High Throughput Computing (HTC) Infrastructures

High Performance Computing (HPC) Infrastructures

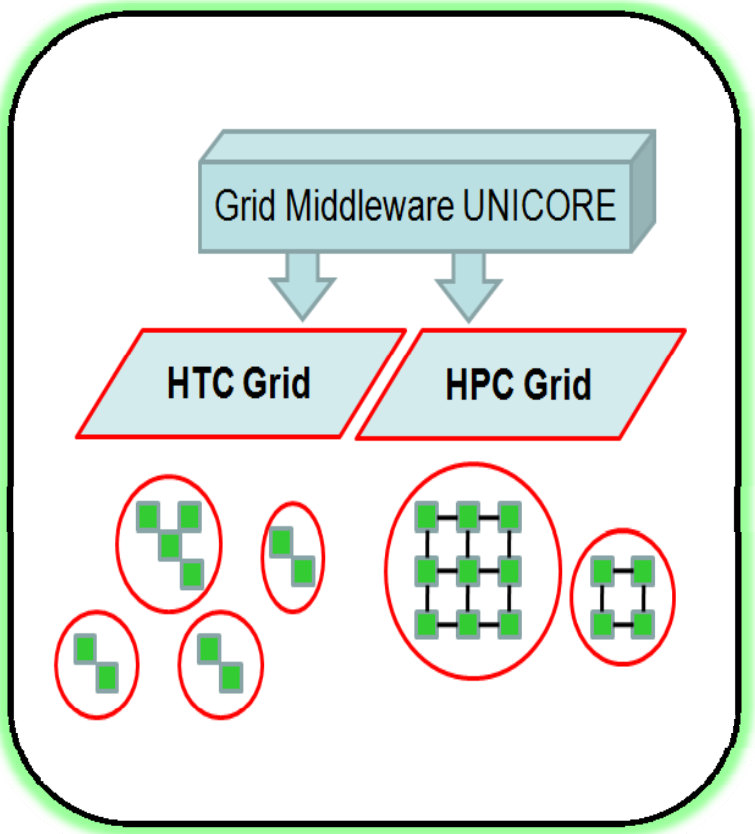
Different Approaches for e-Science



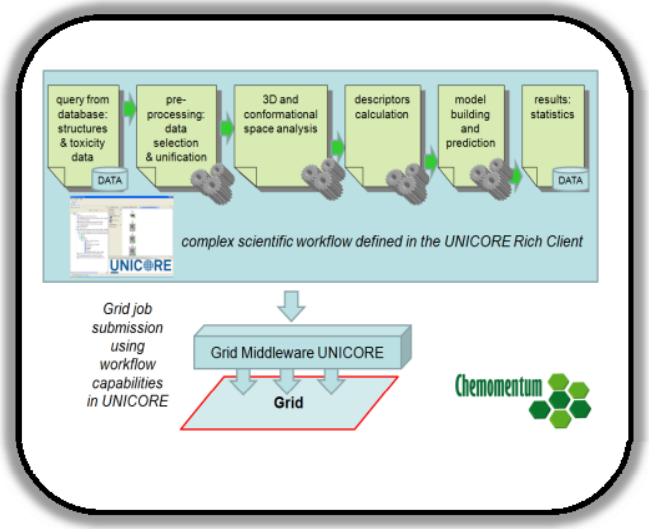
Simple Scripts & Control



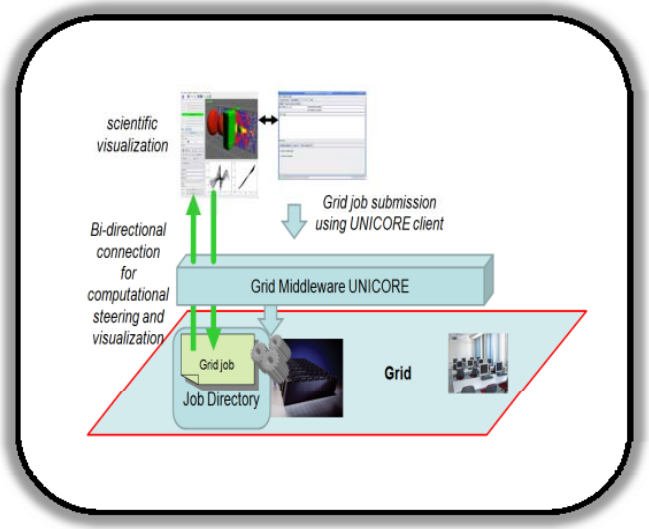
Application Plug-ins



Grid Interoperability



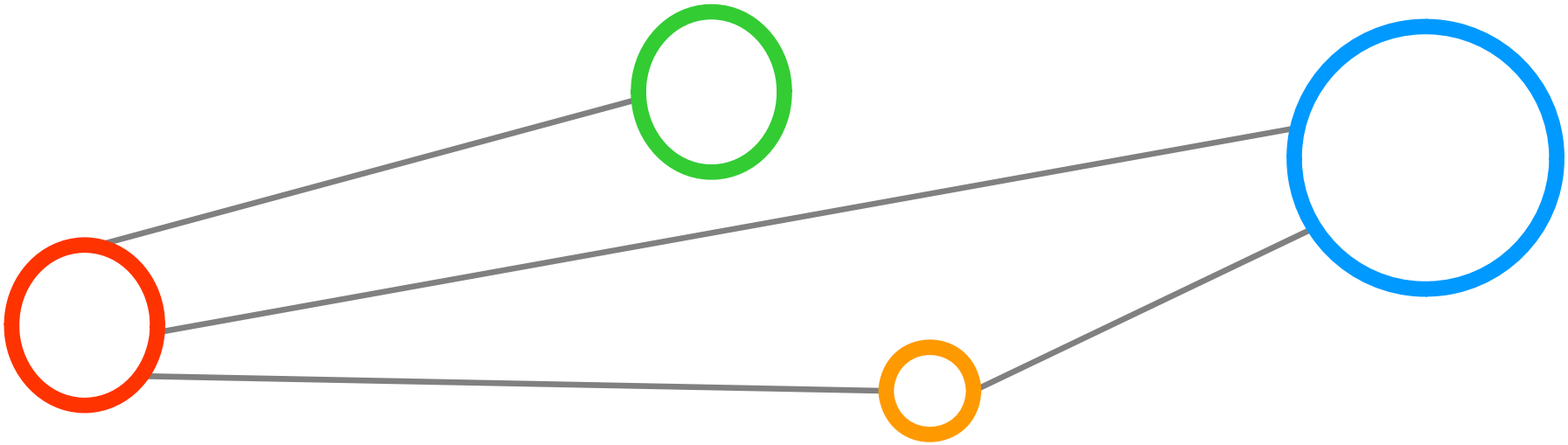
Complex Workflows



Interactive Access

[3] Riedel and Kranzlmueller et al.,
'Classification of Different Approaches for e-Science Applications in Next Generation Computing Infrastructures '

Motivation for Interoperability



Motivation

Use different types of resources

Better load-balancing

Combine resources for more realistic simulations

Unified access & single sign-on

Save computational time on rare & costly HPC resources

Synergy in technology development

‚Embarassingly Parallel' Farming Jobs

Massively Parallel Jobs

eGee Enabling Grids for E-sciencE

gLite

HTC Jobs

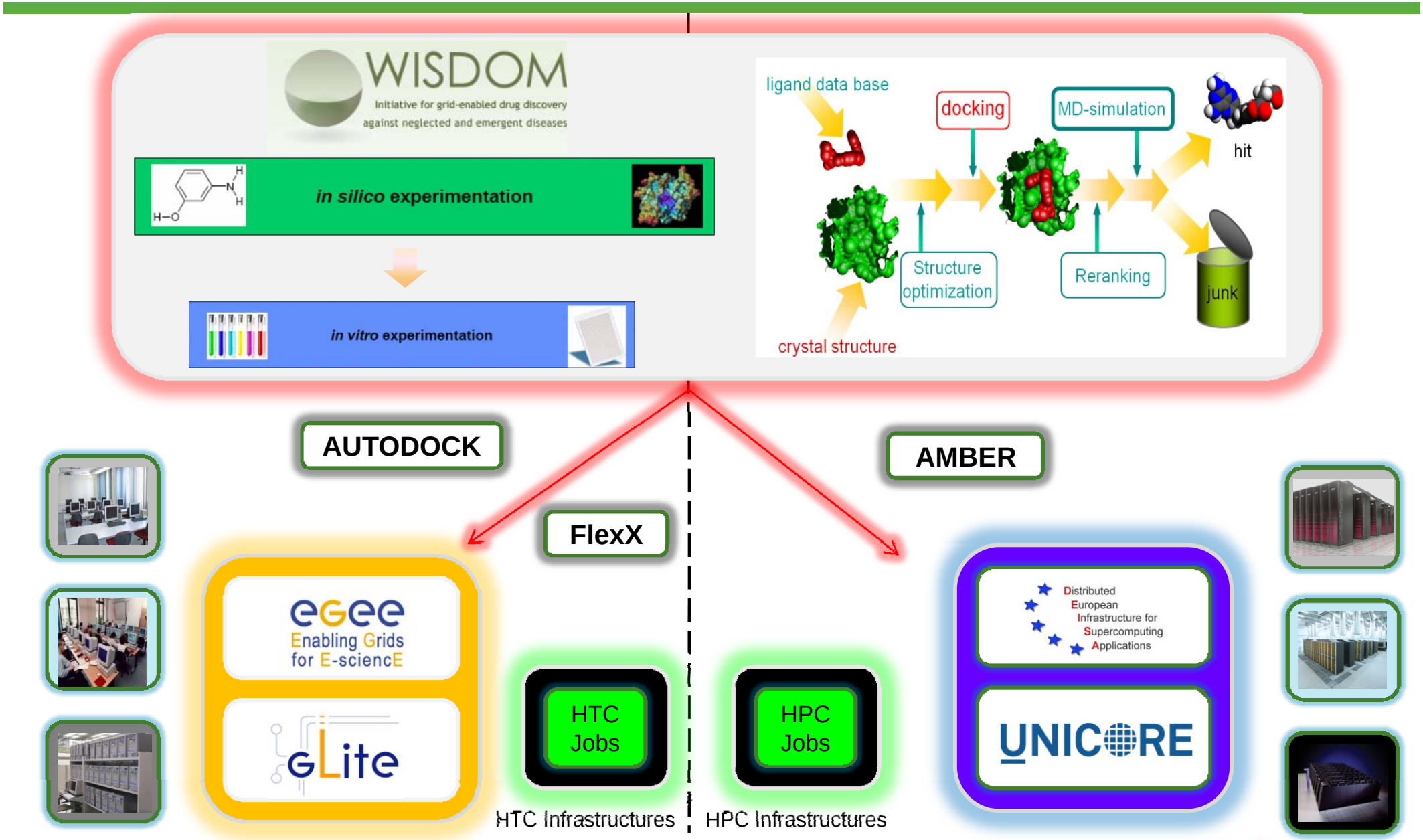
HPC Jobs

Distributed European Infrastructure for Supercomputing Applications

UNICORE

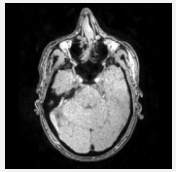
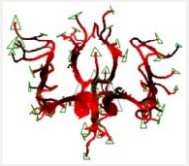
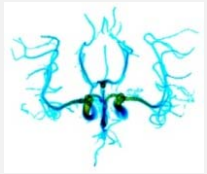
HTC Infrastructures | HPC Infrastructures

e-Health Use Case HTC/HPC



[4] Riedel et al., *'Improving e-Science with Interoperability of the e-Infrastructures EGEE and DEISA'*

e-Health Use Case HPC/HPC


Quantify uncertainties, reduce time-to-solution, different job runs with same code ‚same time'

HEMELB

HEMELB



NGS National Grid Service

omii-uk Software Solutions for e-Research


JÜLICH FORSCHUNGSZENTRUM

rzg IPP RECHEN-ZENTRUM GARCHING

Distributed European Infrastructure for Supercomputing Applications

UNICORE



HPC Infrastructure

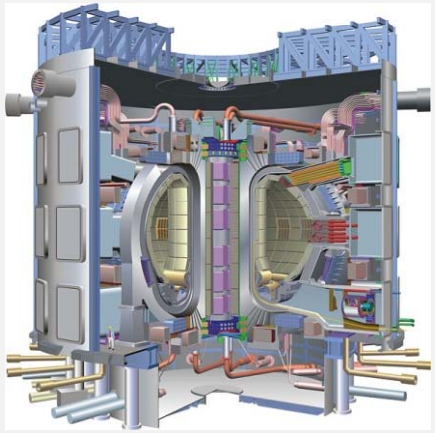
HPC Infrastructure

[10] Manos & Riedel et al., *DEISA Newsletter December 2008*

Fusion Use Case Example

EUFORIA

Advanced cross-computational paradigm simulation of future power generating power plants



Fusion HTC code suite

eGee Enabling Grids for E-sciencE

gLite

HTC Jobs

HPC Jobs

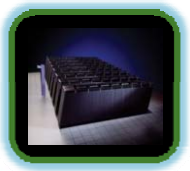
HTC Infrastructures

HPC Infrastructures

Fusion HPC code suite

Distributed European Infrastructure for Supercomputing Applications

UNICORE



www.ogf.org

[4] Riedel et al., *'Improving e-Science with Interoperability of the e-Infrastructures EGEE and DEISA'*

Challenges

One Client (command-line, portal, application with integrated API)

Embarassingly Parallel Jobs

Massively Parallel Jobs

Different job description languages	Different Data Transfer Techniques	Different security setups
Different job submission interfaces & protocols	Different Storage Access Techniques	Different information semantics

eGee Enabling Grids for E-sciencE

gLite

HTC Jobs

HPC Jobs

HTC Infrastructures

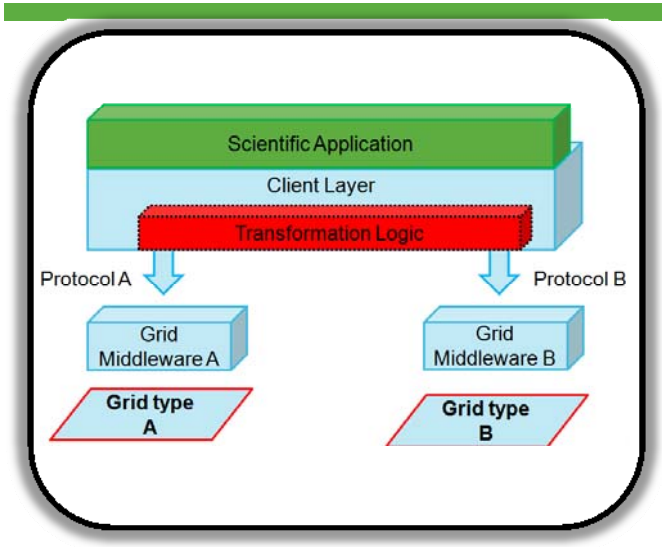
HPC Infrastructures

Distributed European Infrastructure for Supercomputing Applications

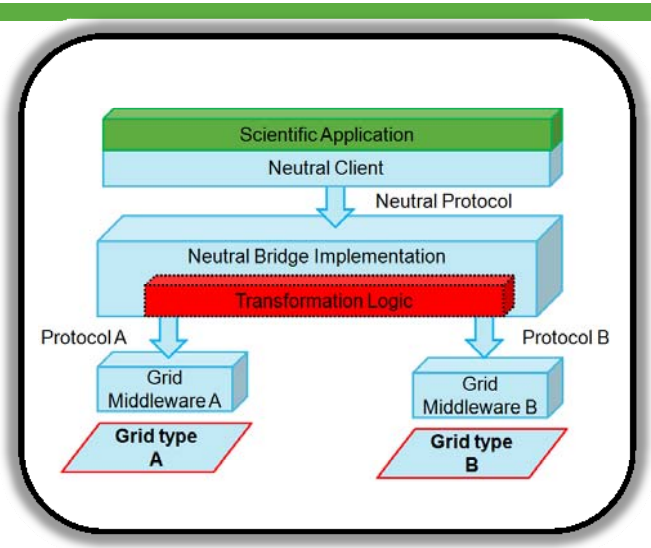
UNICORE

[5] Riedel et al., *'Experiences and Requirements for Interoperability between HTC- and HPC-driven e-Science Infrastructures'*

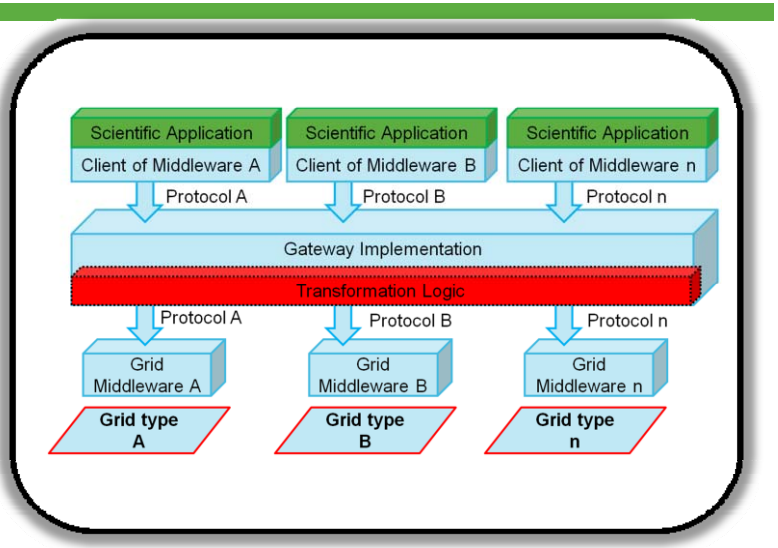
Different Approaches for Interoperability



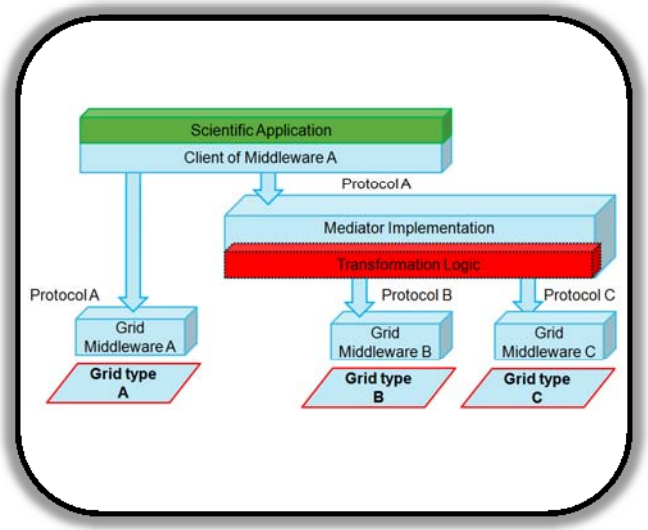
Client Layer Approach



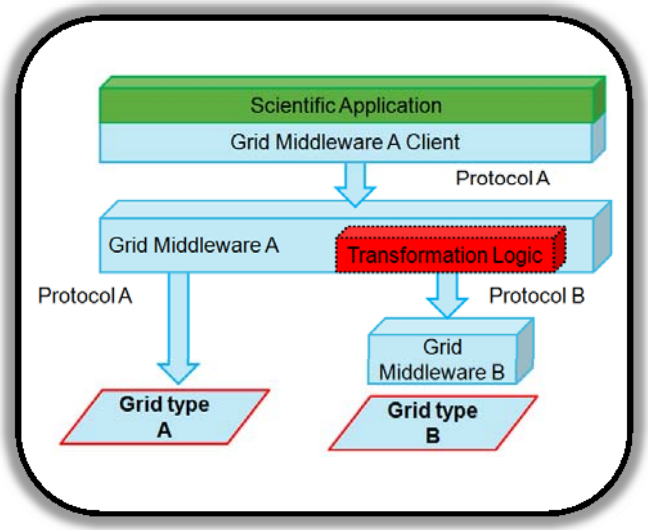
Neutral Bridge Approach



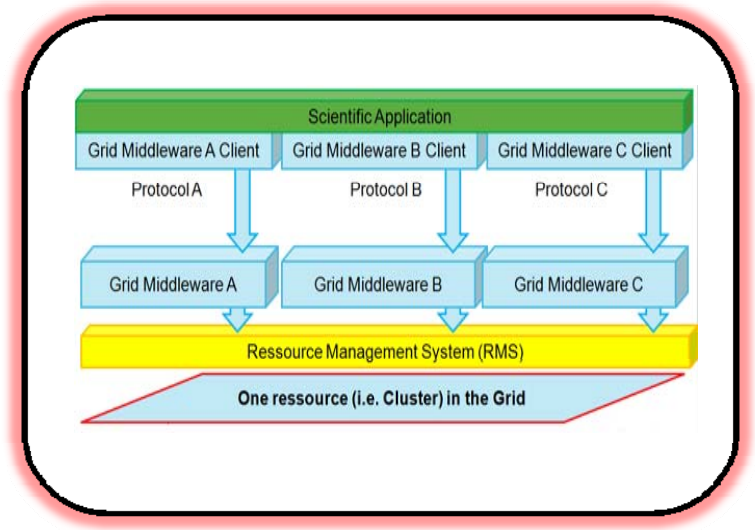
Gateway Approach



Mediator Approach



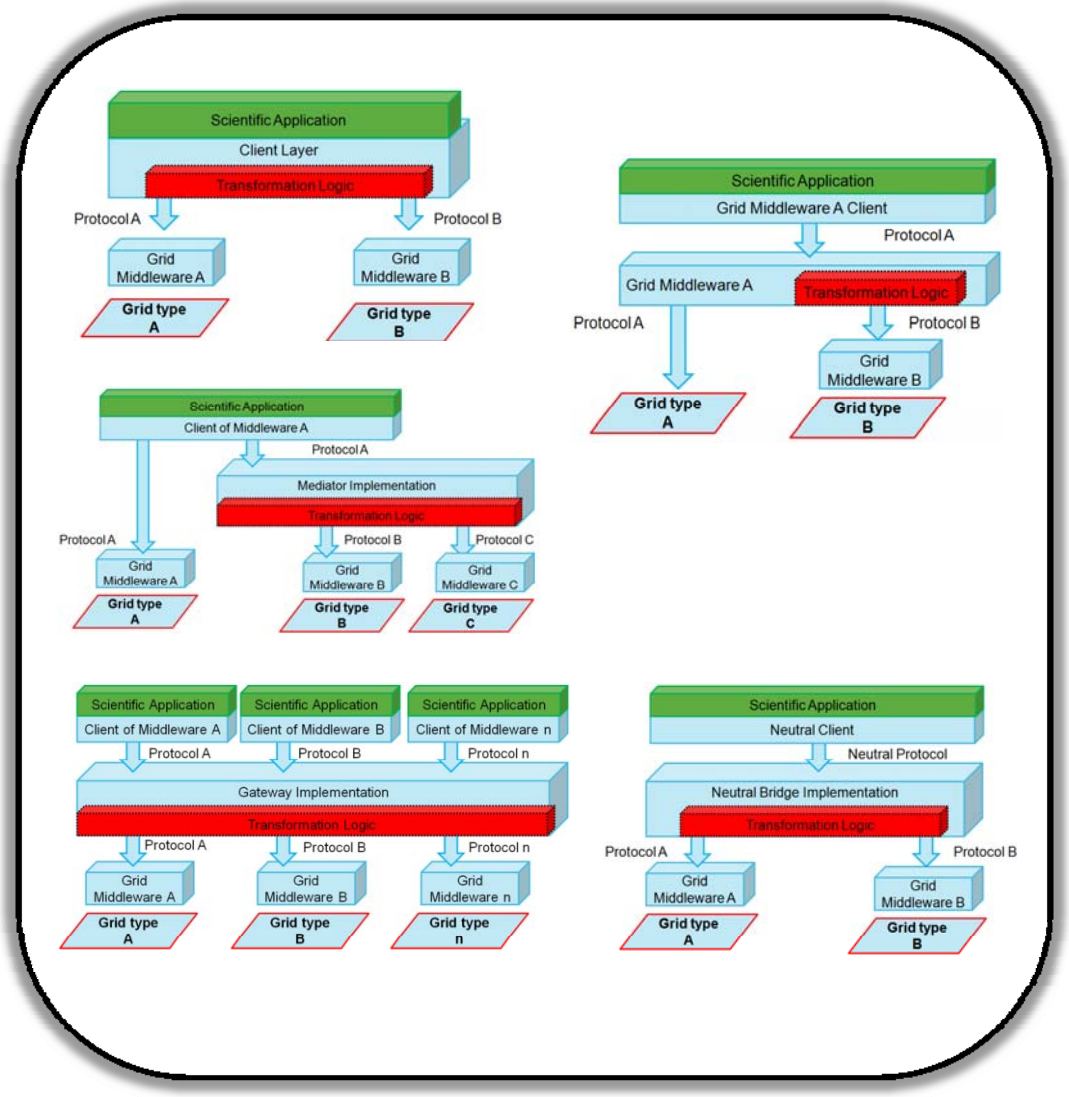
Adapter Approach



Middleware Co-Existence

[X] Riedel et al., '*Research Advances by using Interoperable e-Science Infrastructures – The Infrastructure Interoperability Reference Model applied in e-Science* '

Transformation Logic



Approaches that require transformation logic

protocol A or schema A

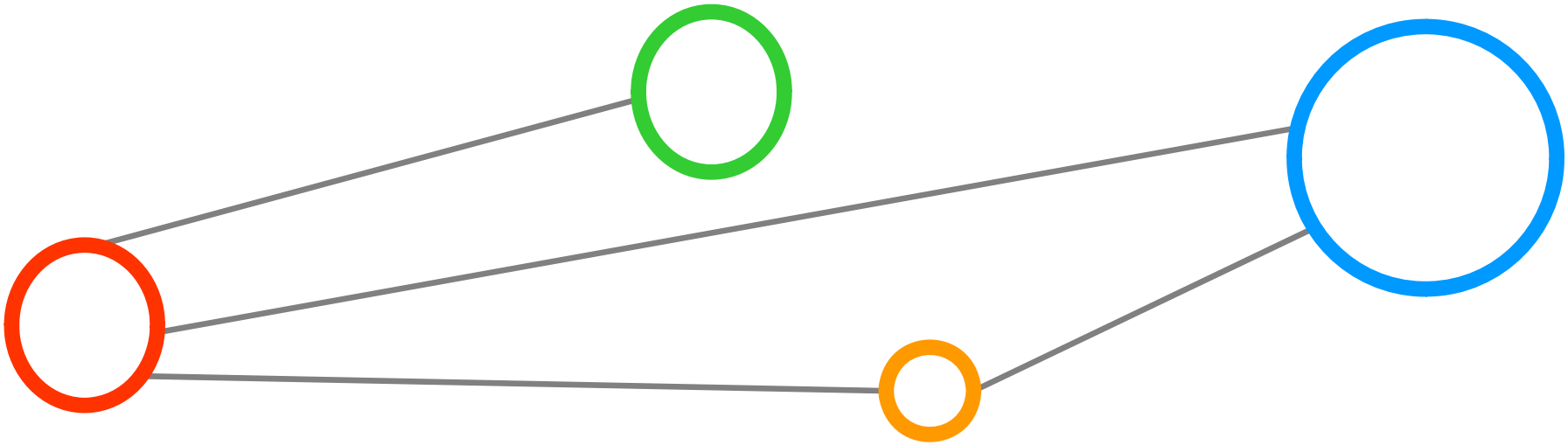
Transformation Logic

- Time consuming & error-prone
- Difficult to maintain (n versions)
- Protocol B‘ implies result is often only a subset of protocol B

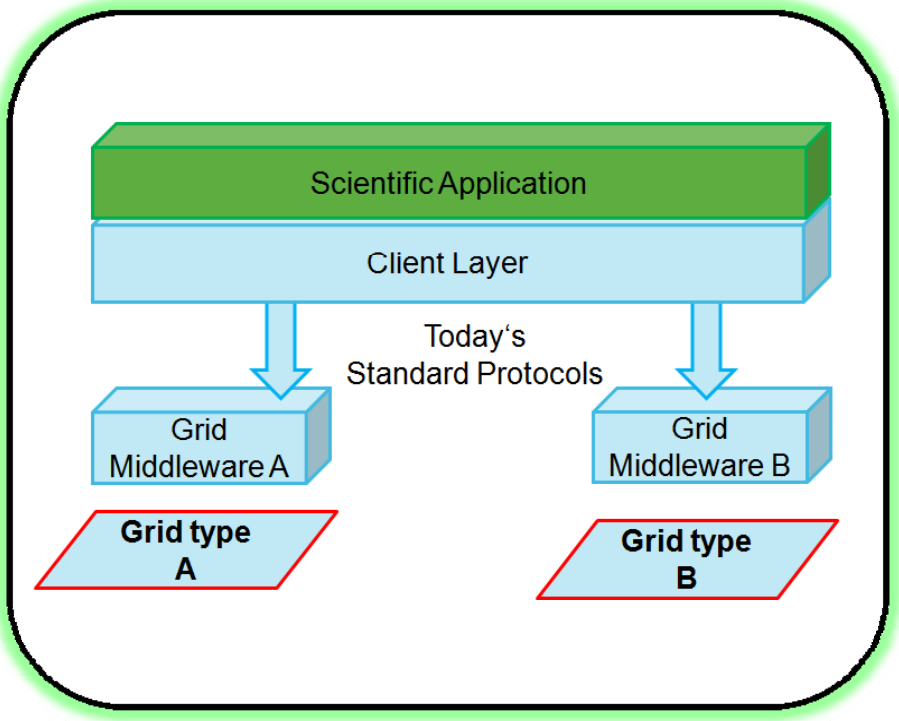
protocol B‘ or schema B‘

[X] Riedel et al., ‘*Research Advances by using Interoperable e-Science Infrastructures – The Infrastructure Interoperability Reference Model applied in e-Science ‘*

Emerging Open Standards



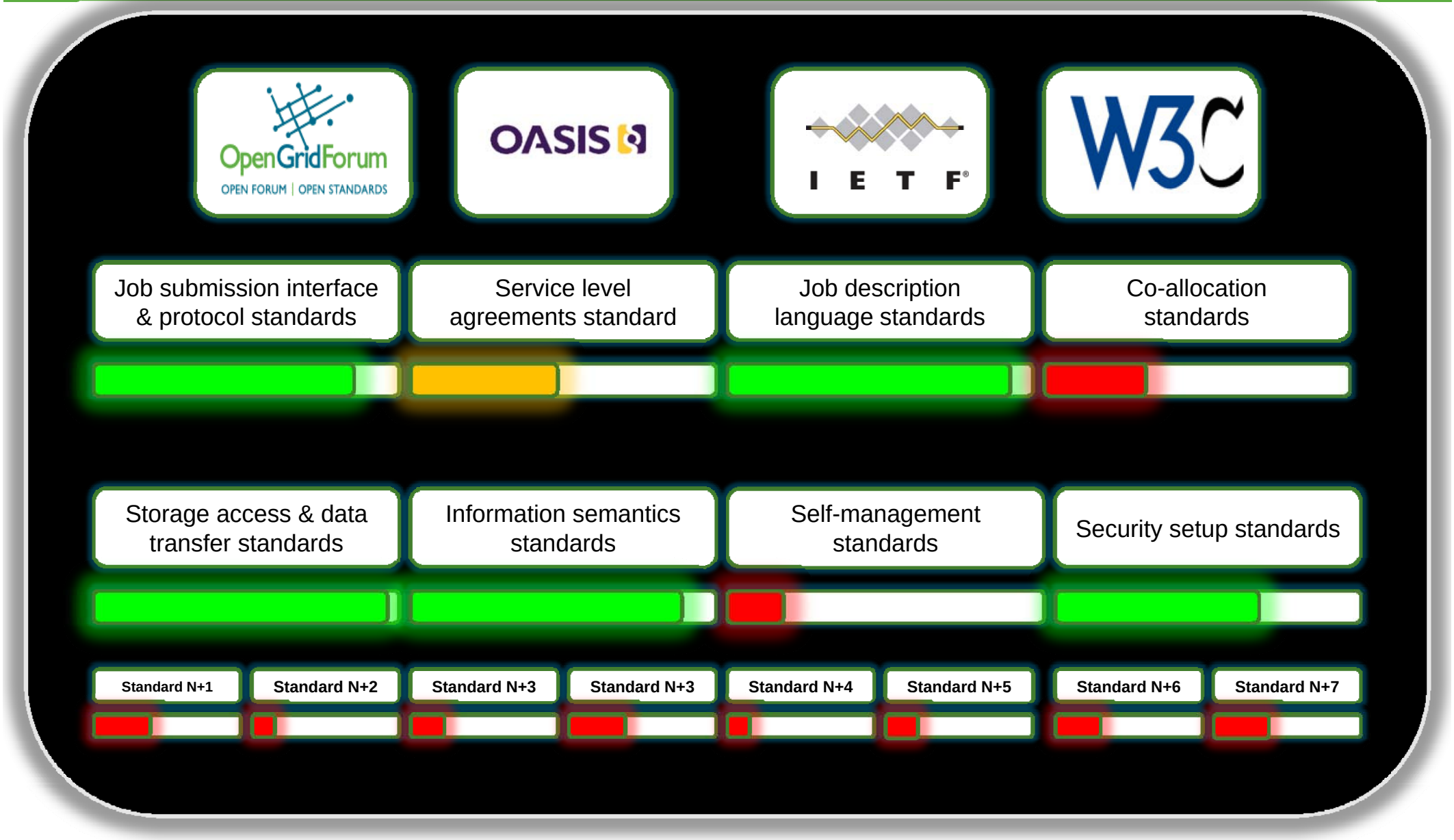
Open Standards Approach



Open Standards Approach

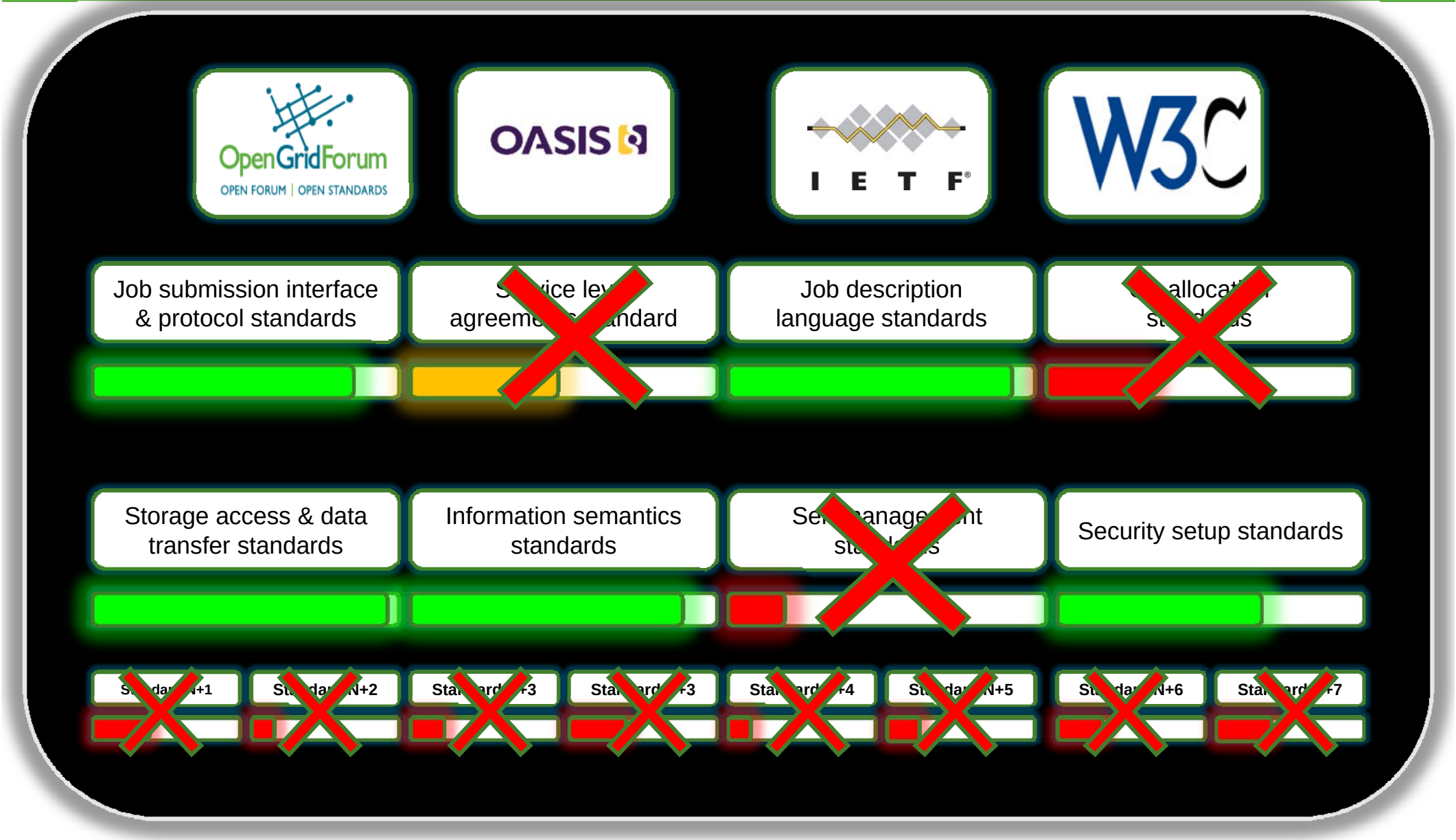
- No transformation logic required
- Requires substantial effort to reach an agreement between middlewares that adopt them
- Should not only be based on (rather theoretical) use cases
- Instead they should also take lessons learned from real production usage into account

OGSA Standards & Adoption



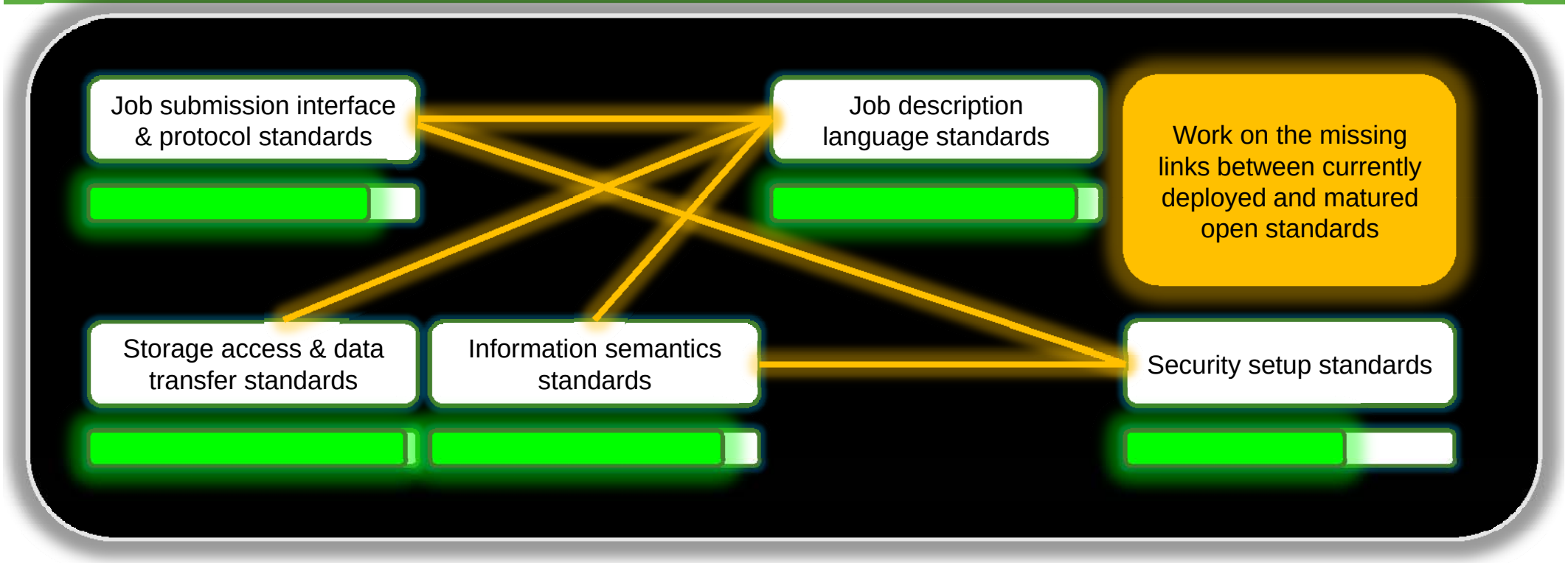
[7] Foster et al., *'The Open Grid Services Architecture'*

GIN Production Experience

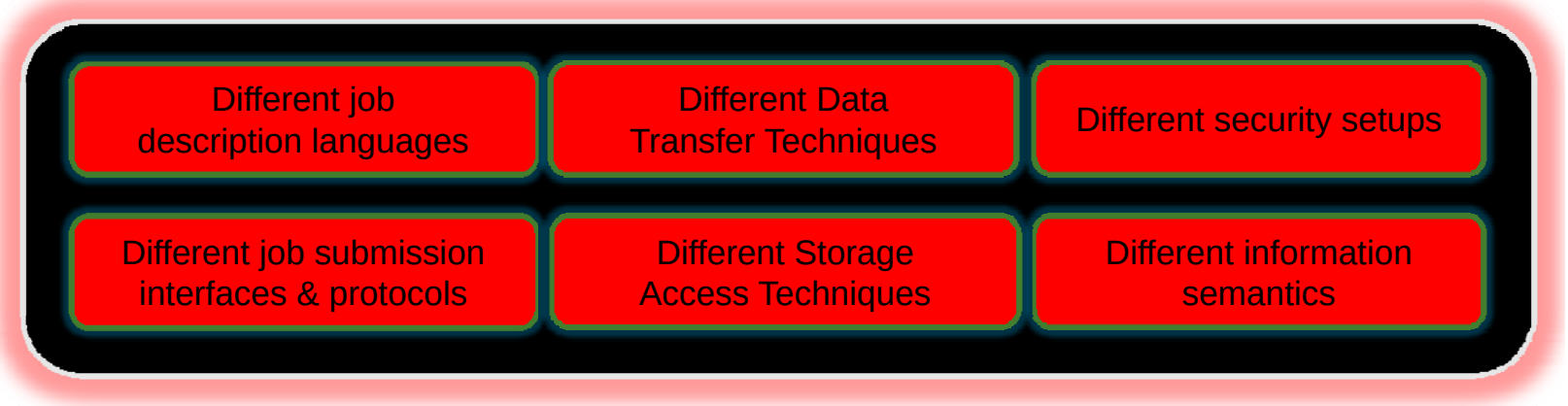


[8] Riedel et al., *'Interoperation of World-Wide Production e-Science Infrastructures '*

PGI Approach (1)

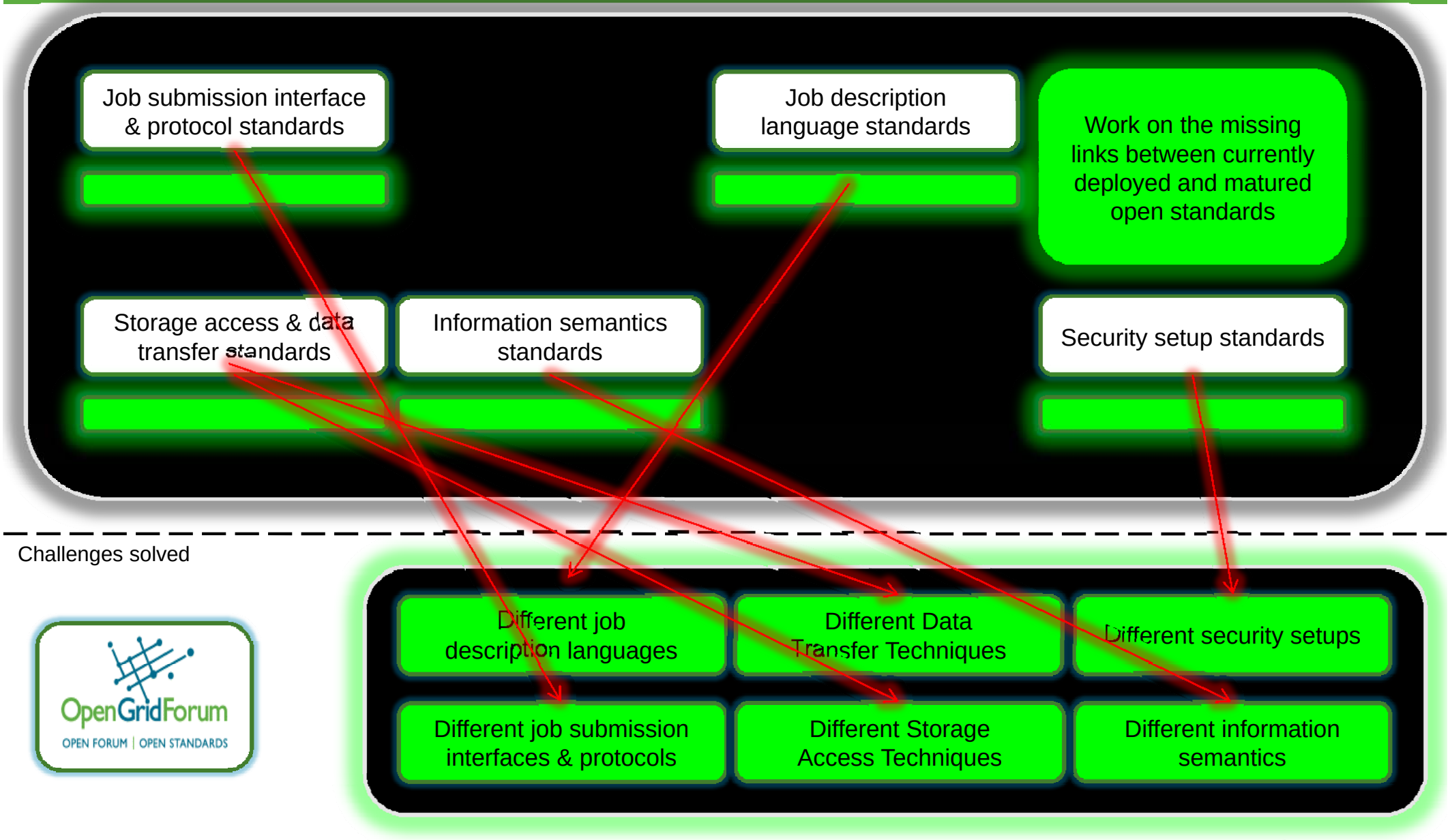


Challenges



[5] Riedel et al., *'Experiences and Requirements for Interoperability between HTC- and HPC-driven e-Science Infrastructures'*

PGI Approach (2)



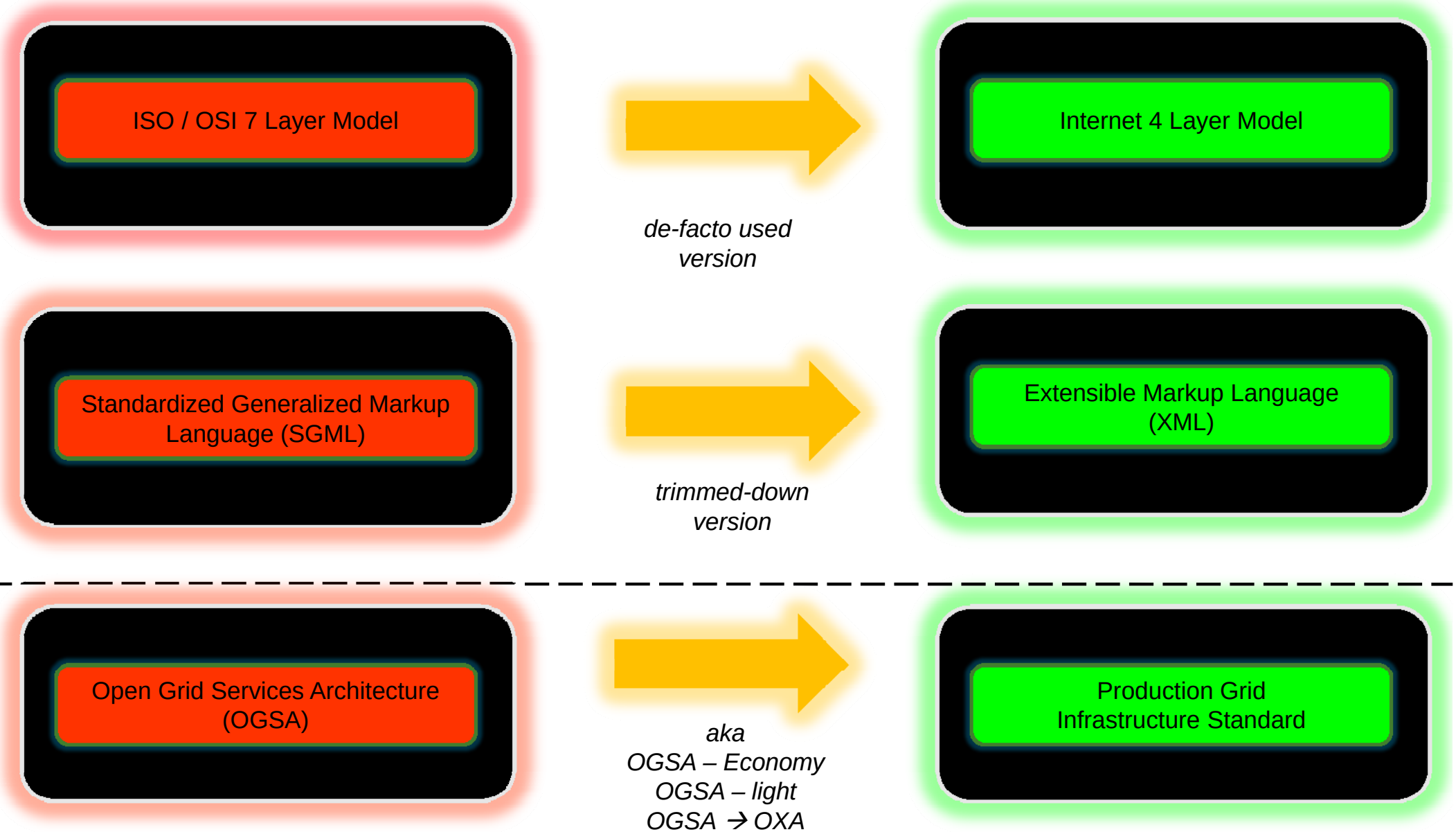
[5] Riedel et al., *'Experiences and Requirements for Interoperability between HTC- and HPC-driven e-Science Infrastructures'*

PGI Scope

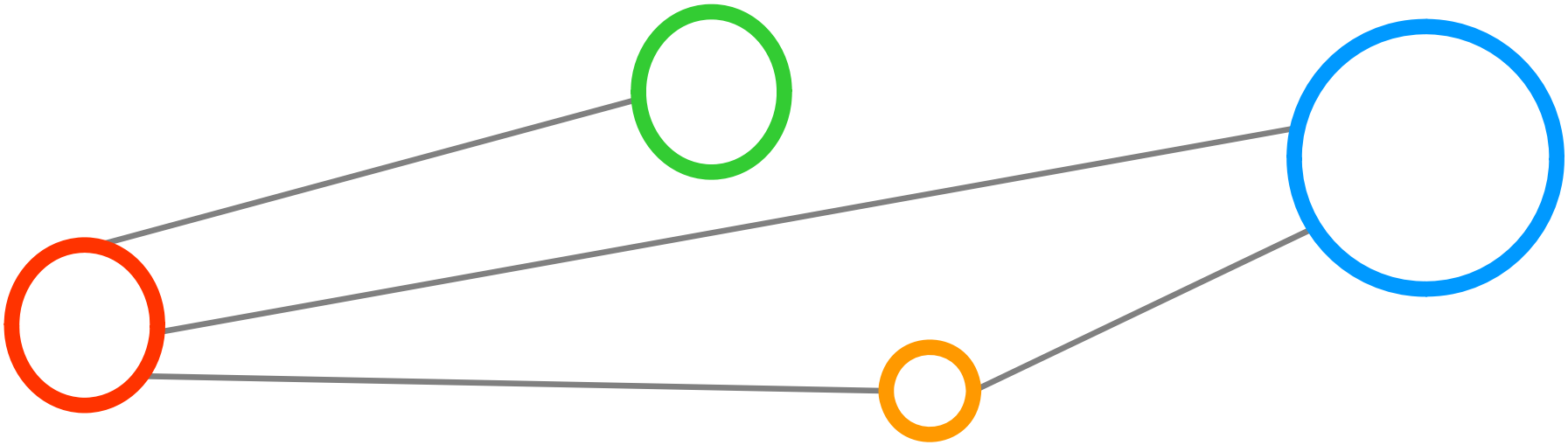
- Only matured specifications
- Specification adoption exist in production middleware systems
- Experience exists in production infrastructures
- Interoperability tests have been regularly performed
- Real scientific use cases require these standards
- Sometimes only refinements necessary and not complete specification re-definitions

→ ‘Low hanging fruits’

Compare History of Computer Science



Interoperability Reference Model



Often Used Functional Interfaces

GIN Interoperation demonstrations from numerous world-wide projects

SC07 is the International Conference for High Performance Computing, Networking, Storage and Analysis

OpenGridForum
OPEN FORUM | OPEN STANDARDS

SC08 is the International Conference for High Performance Computing, Networking, Storage and Analysis

Work with emerging open standards on real production Grid applications

WISDOM

EUFORIA

Virtual Physiological Human network of excellence

International Grid Interoperability & Interoperation Workshops 2007, 2008 & Grid Computing Journal Special Issue Interoperability 2009

GridFTP
OGF Specification GFD

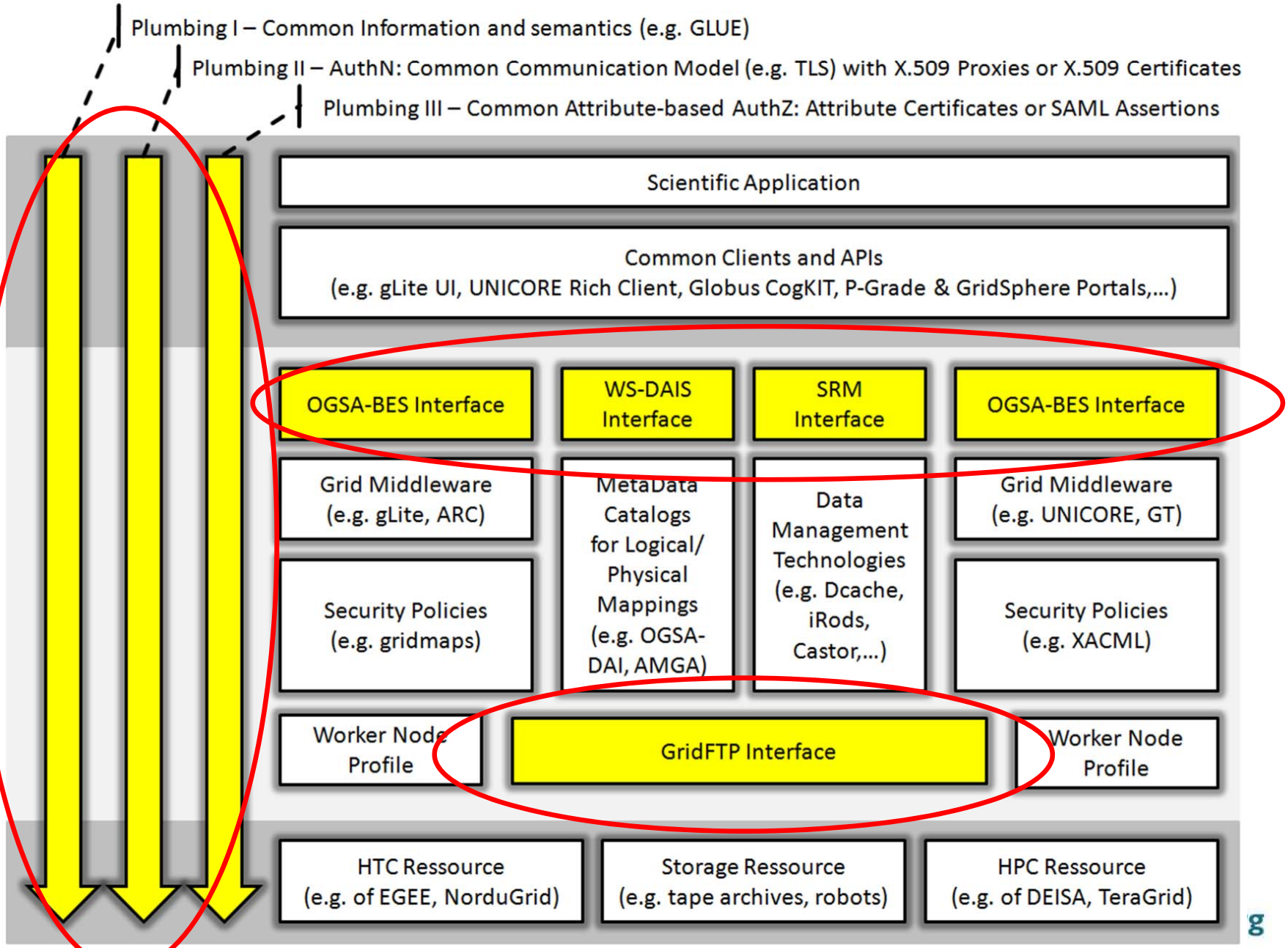
Storage Ressource Manager (SRM)
OGF Specification GFD

OGSA – Basic Execution Service (BES)
OGF Specification GFD

Job Submission & Description Language (JSDL)
OGF Specification GFD

WS-Data Access&Integration Service (DAIS)
OGF Specification GFD

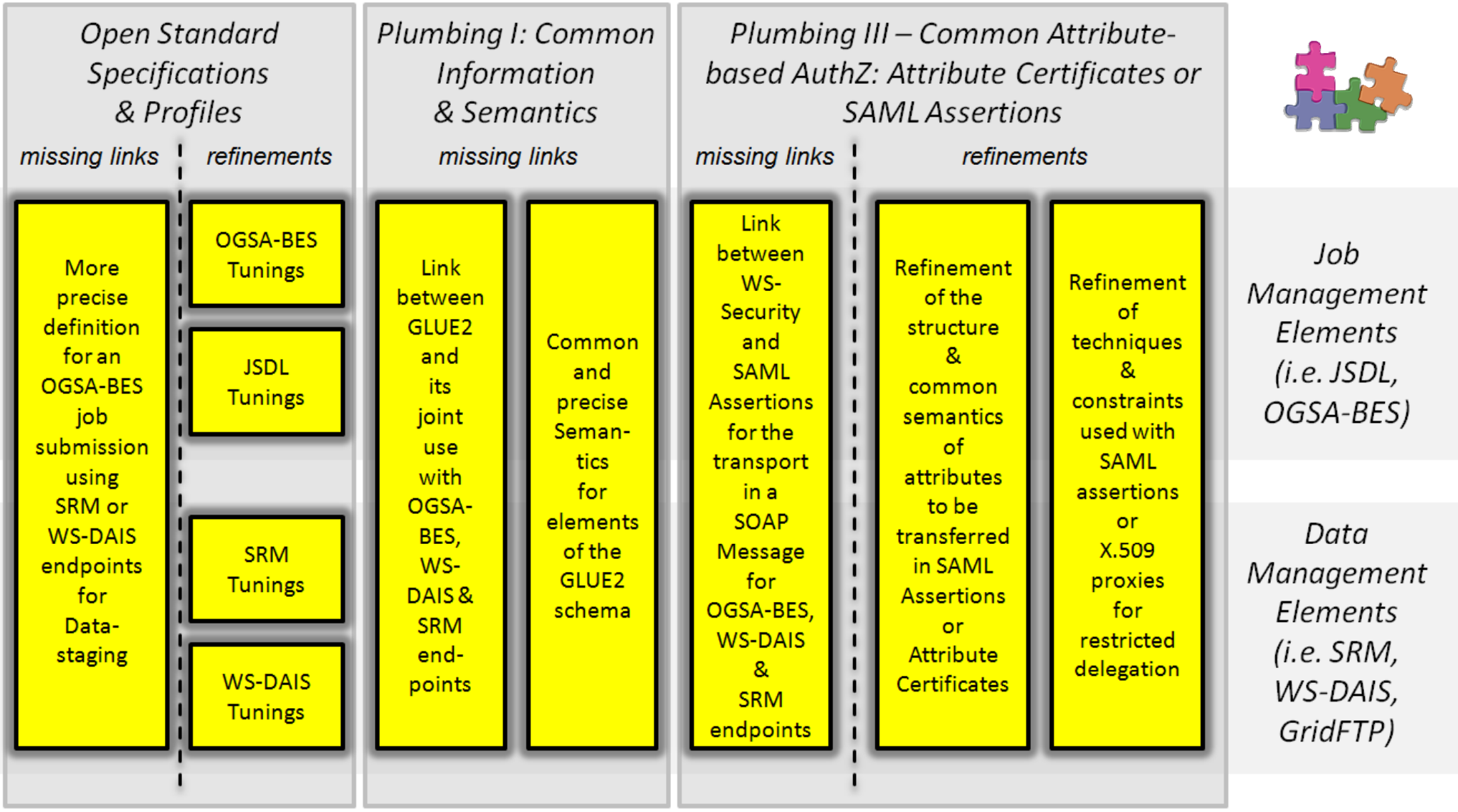
Reference Model Overview



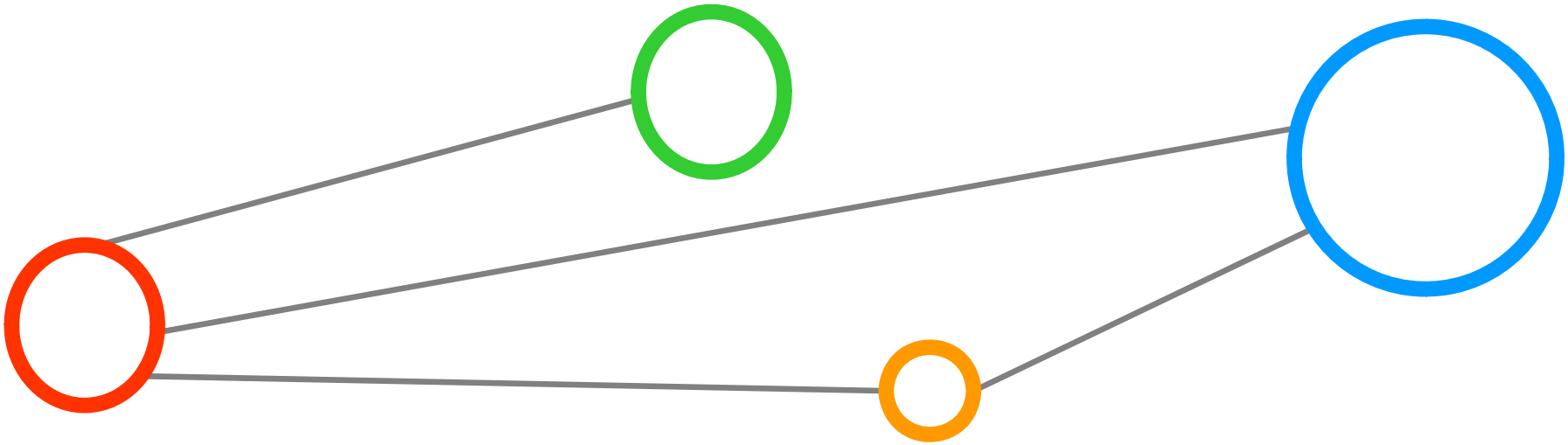
Plumbings Concept

- Plumbings can be used to put different ‚elements‘ through
  - E.g. warm water (full X.509 certificates) vs. Cold water (X.509 proxies)
- Many plumbings can be installed in parallel – while not crossing the other plumbings
  - E.g. modern container concepts allow easily addition of n handler that can take care of the elements by n plumbings
- Different plumbings can use the same source and can be sink into the same achievement/functionality
  - E.g. Attribute-based VOMS system vs. SAML-based VOMS system
  - Both based on same VO DBs but convey attributes differently
  - However, authZ decision based on these attributes can be again usable for both approaches (e.g. one XACML policy file)
- Plumbings may be removed over time while new plumbings are already deployed in infrastructures
  - E.g. support for ‚old deprecated production standards‘

Missing Links & Refinements



Computing Refinement Concepts



Emerging Standards in Context

- OGSA – Basic Execution Service (BES)
  - OGF Specification GFD108, out since 2007-08-07
  - Provides a functional interface to manage computational jobs
  - Implies the use of JSDL as jobs description language
  - Defines a job state model that is simple – but extensible
  - Since 2007 in use in many different use cases and some middleware
- Job Submission and Description Language (JSDL)
  - OGF Specification GFD56, out since 2005 / 2006
  - Some standardized extensions since then: Single Process Multiple Data (SPMD) – 2007, HPC-Profile – 2007, Parameter Sweep – 2009
  - Since 2005 in use in many different use cases and many middleware
- OGSA-BES and JSDL already a good starting point
  - No need to start from scratch and a good base for refinements
  - Lessons learned: Over the years many additional required concepts have been identified mostly driven by the needs of e-scientists

Refinement Concepts Overview

Concepts	OGSA-BES / JSDL	Improvements
Simple job submission	Yes	Yes
Cancellation of submitted jobs	Yes	Yes
Getting submitted job states	Yes	Yes
Remote management operations	Yes	No
Client initiated data-staging	No	Yes
Immediate job working directory access	No	Yes
Predefined hold points	No	Yes
Manual manipulation of job states	No	Yes
Data-staging in state model	No	Yes
Wipe-out of submitted jobs	No	Yes
Standardized information model	No	Yes
Recent HPC resource support	No	Yes
Pre-/post processing	No	Yes
Data-transfer delegation	No	Yes
Multiple computing share support	No	Yes

Fundamental Concepts Ok

Concepts	OGSA-BES / JSDL	Improvements
Simple job submission	Yes	Yes
Cancellation of submitted jobs	Yes	Yes
Getting submitted job states	Yes	Yes

- Simple job submission
  - Refers to run one executable on a remote machine with limited resource requirements (CPUs) and automatic data-staging
  - OGSA-BES & JSDL (with extensions) supports this already via the ‚application' elements in JSDL
- Cancellation of submitted jobs
  - Refers to once submitted jobs can be cancelled
  - OGSA-BES / JSDL supports this already via *TerminateActivities()* operation and the ‚cancelled' job state
- Getting submitted job states
  - Refers to the ability to request the up-to-date state of the job
  - OGSA-BES / JSDL supports this via *GetActivityStatuses()* operation

Remote Management

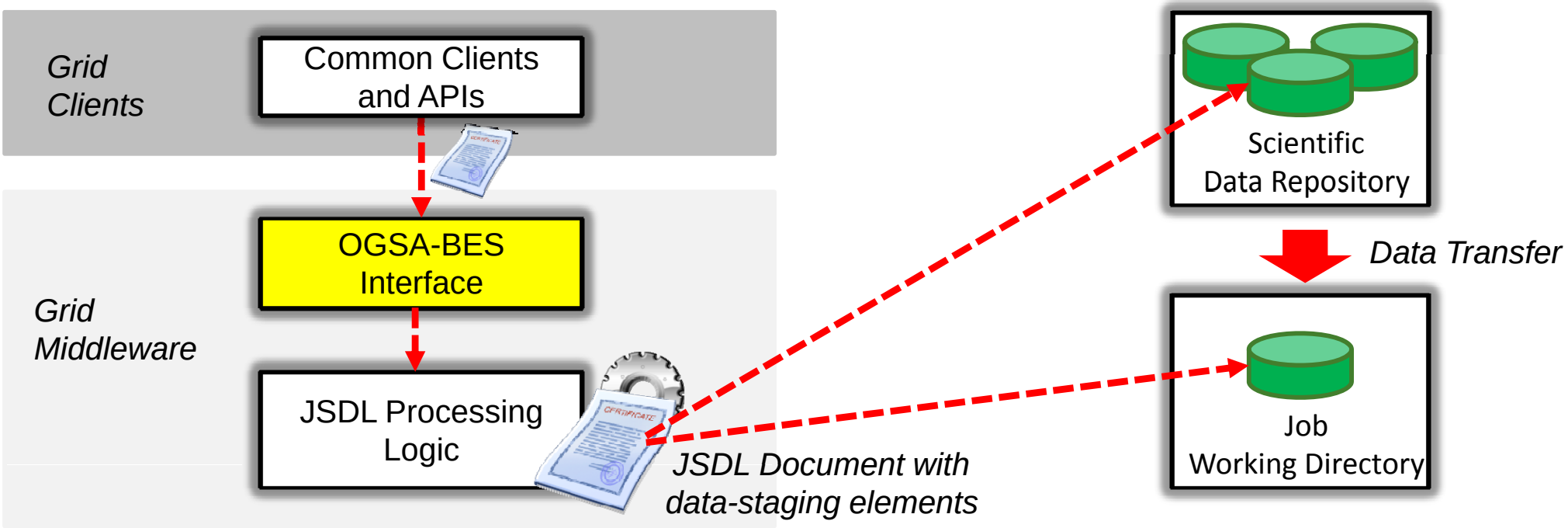
Concepts	OGSA-BES / JSDL	Improvements
Remote management operations	Yes	No

- OGSA-BES / JSDL define funtionality for remote management in terms of ‚accepting new activities‘
  - OGSA-BES provides a BES-Management portType with two operations
  - StartAcceptingNewActivities() / StopAcceptingNewActivities()
  - IsAcceptingNewActivities as boolean for BES Factory attributes that describe the fundamental properties of one computing site
- Improvements (here reduction)
  - The BES-Management concept is marked as ‚deprecated‘
  - Major reason is that production use reveals that this concept is rather rarely remotely used in production Grids
  - Site property is preferred configured locally by site administrators

Client initiated data-staging (1)

Concepts	OGSA-BES / JSDL	Improvements
Client initiated data-staging	No	Yes

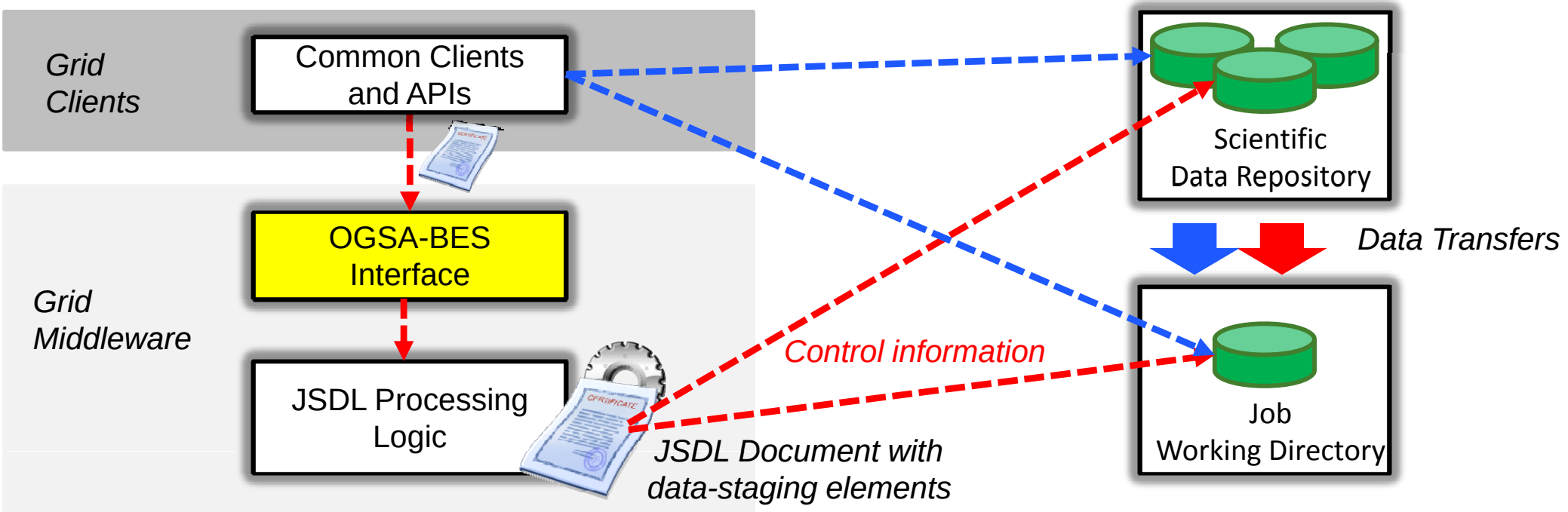
- OGSA-BES / JSDL define funtionality for staging data automatically performed via the middleware
  - Works via data-staging-in and data-staging-out JSDL elements
  - Can be considered as a kind of ‚data-pull‘ concept



Client initiated data-staging (2)

Concepts	OGSA-BES / JSDL	Improvements
Client initiated data-staging	No	Yes

- Improved OGSA-BES / JSDL defines funtionality for staging data manually performed via the client
  - Identified via data-staging-in and data-staging-out JSDL elements
  - Can be considered as a kind of ‚data-push‘ concept
  - Requires other concepts ‚holdpoints‘ & ‚Working Directory Access‘

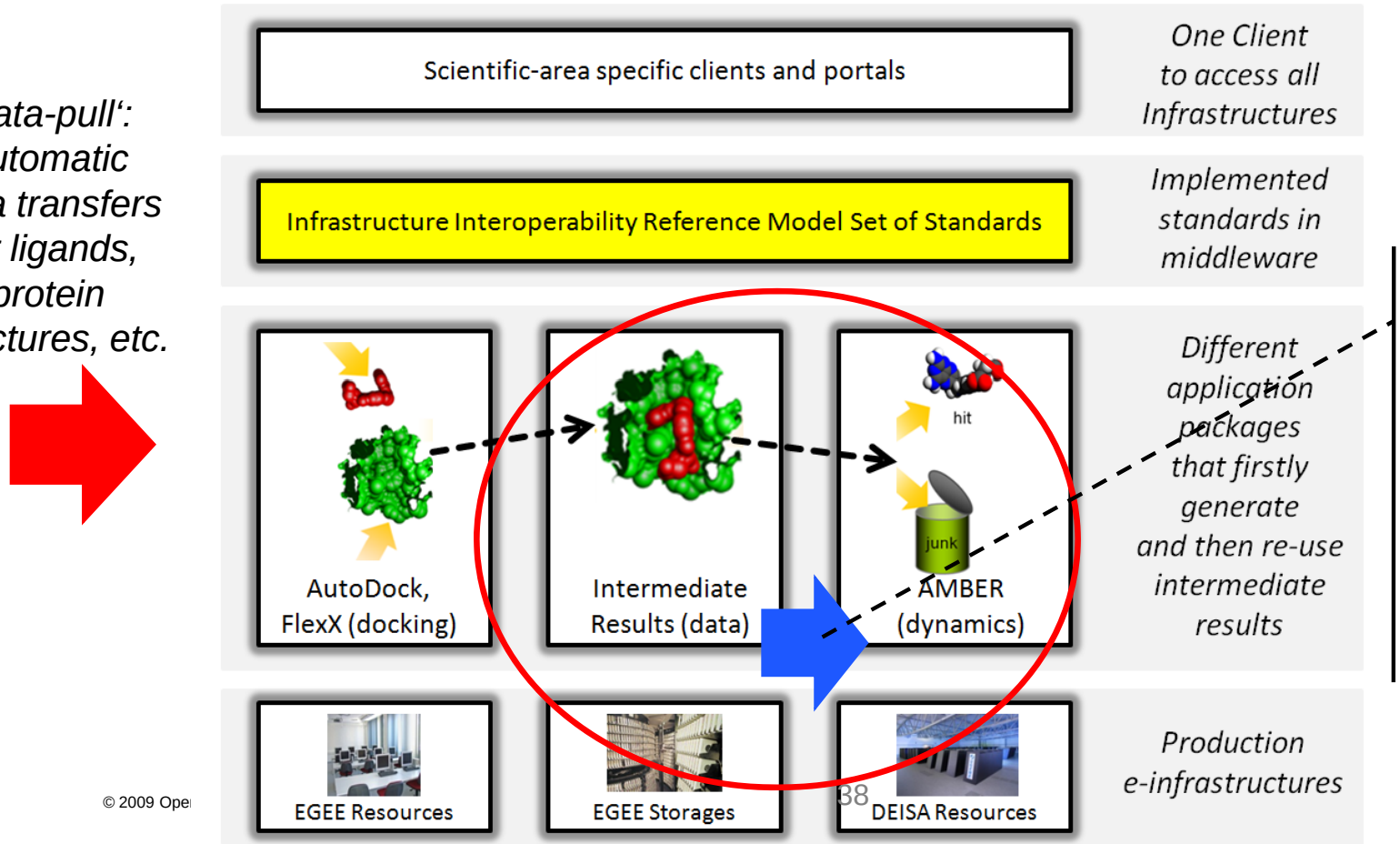


Client initiated data-staging (3)

Concepts	OGSA-BES / JSDL	Improvements
Client initiated data-staging	No	Yes

- Example of this requirements from an e-science perspective
  - Manual: Only a subset of ‚valuable' intermediate data is used in costly HPC computing

‚data-pull': automatic data transfers for ligands, protein structures, etc.



‚data-push': manual data transfers initiated on the client level with strong participation of the e-scientists during/after his/her analysis and evaluation

Client initiated data-staging (4)

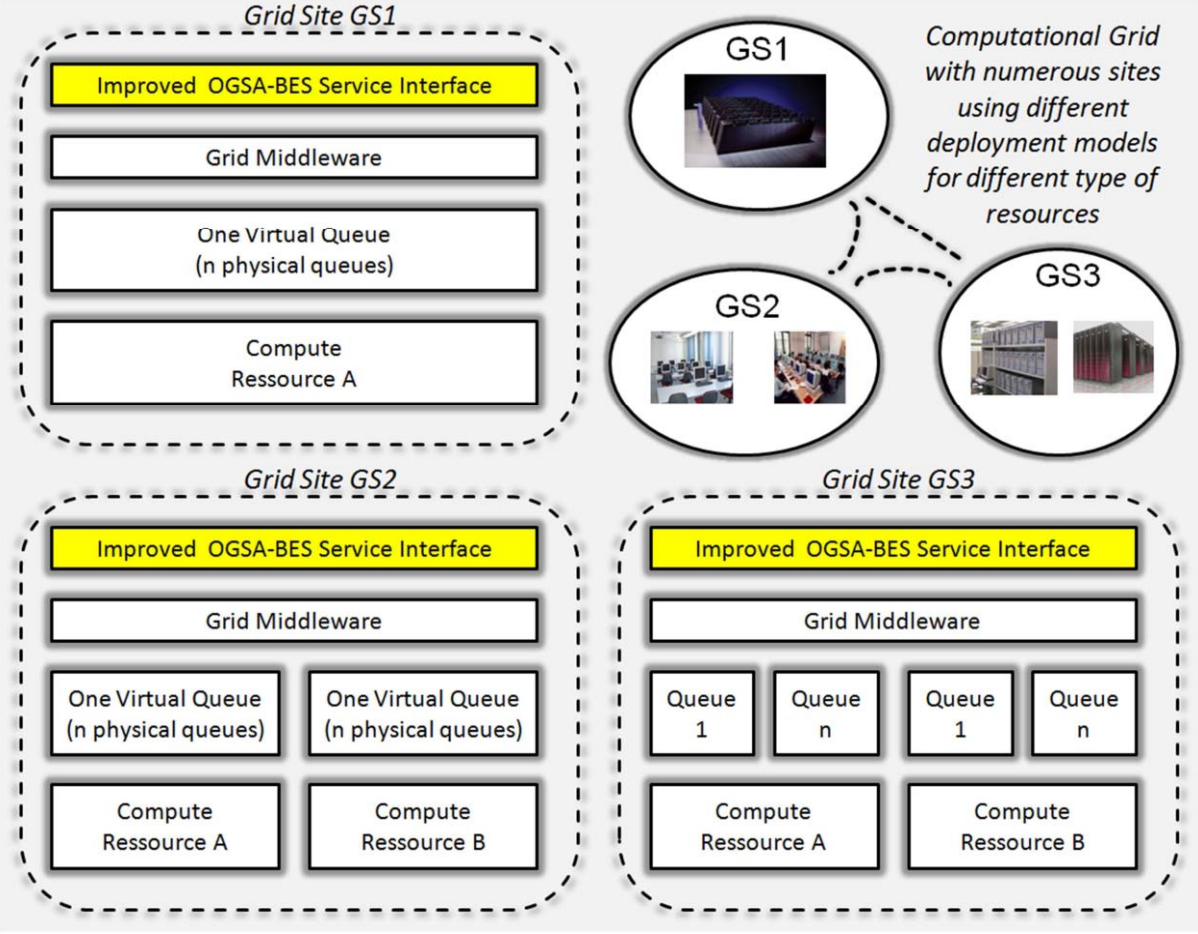
Concepts	OGSA-BES / JSDL	Improvements
Client initiated data-staging	No	Yes
Immediate job working directory access	No	Yes
Predefined hold points	No	Yes
Manual manipulation of job states	No	Yes

- ‚Client initiated data-staging‘ concept requires other concepts
- ‚Immediate job working directory access‘ concept
  - Once job is created the improved OGSA-BES returns the job working directory in order to know where to manually ‚stage-data in&out‘
- ‚Predefined hold points‘ concept
  - Hold points in improved JSDL enables stop of job processing
  - Provides e-scientists with all the time they need to stage-in manually
  - Cp. ‚breakpoints‘, but ‚holdpoints‘ have no direct executable impact
- ‚Manual manipulation of job states‘ concept
  - In order to resume the ‚holded processing‘ a manualy manipulation of states (i.e. continute in hold) is provided via the improved OGSA-BES

Multiple Share Concept

Concepts	OGSA-BES / JSDL	Improvements
Multiple computing share support	No	Yes

More fine-granular URIs are required to specify exactly which ‚computational share' / site:
https://jump.fz-juelich.de:8080/besservice/FZJ/JUMP/c bench
https://jugene.fz-juelich.de:8080/besservice/FZJ/JUGENE/res vph



Other concepts (1)

Concepts	OGSA-BES / JSDL	Improvements
Data-staging in state model	No	Yes
Wipe-out of submitted jobs	No	Yes
Standardized information model	No	Yes

- ‚Data-staging‘ in state model concept
  - Users have to know all the time what the system does
- ‚Wipe-out of submitted jobs‘ concept
  - Instead of ‚only cancelled‘ some jobs should be not tracked by the system anymore
- Standardized information model concept
  - Use of GLUE2 for resource requests in improved JSDL

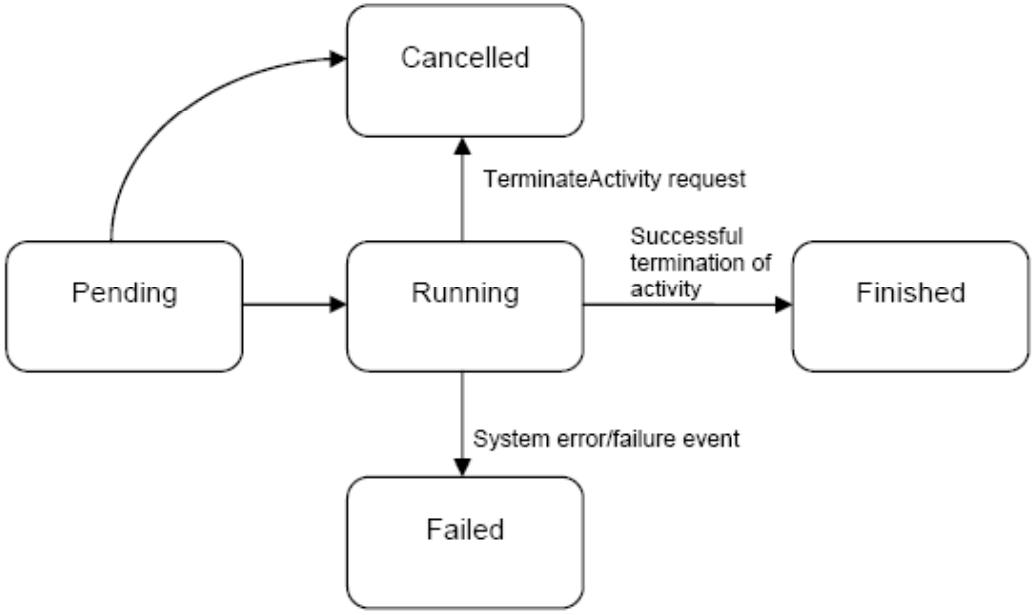
Other concepts (2)

Concepts	OGSA-BES / JSDL	Improvements
Recent HPC resource support	No	Yes
Pre-/post processing	No	Yes
Data-transfer delegation	No	Yes

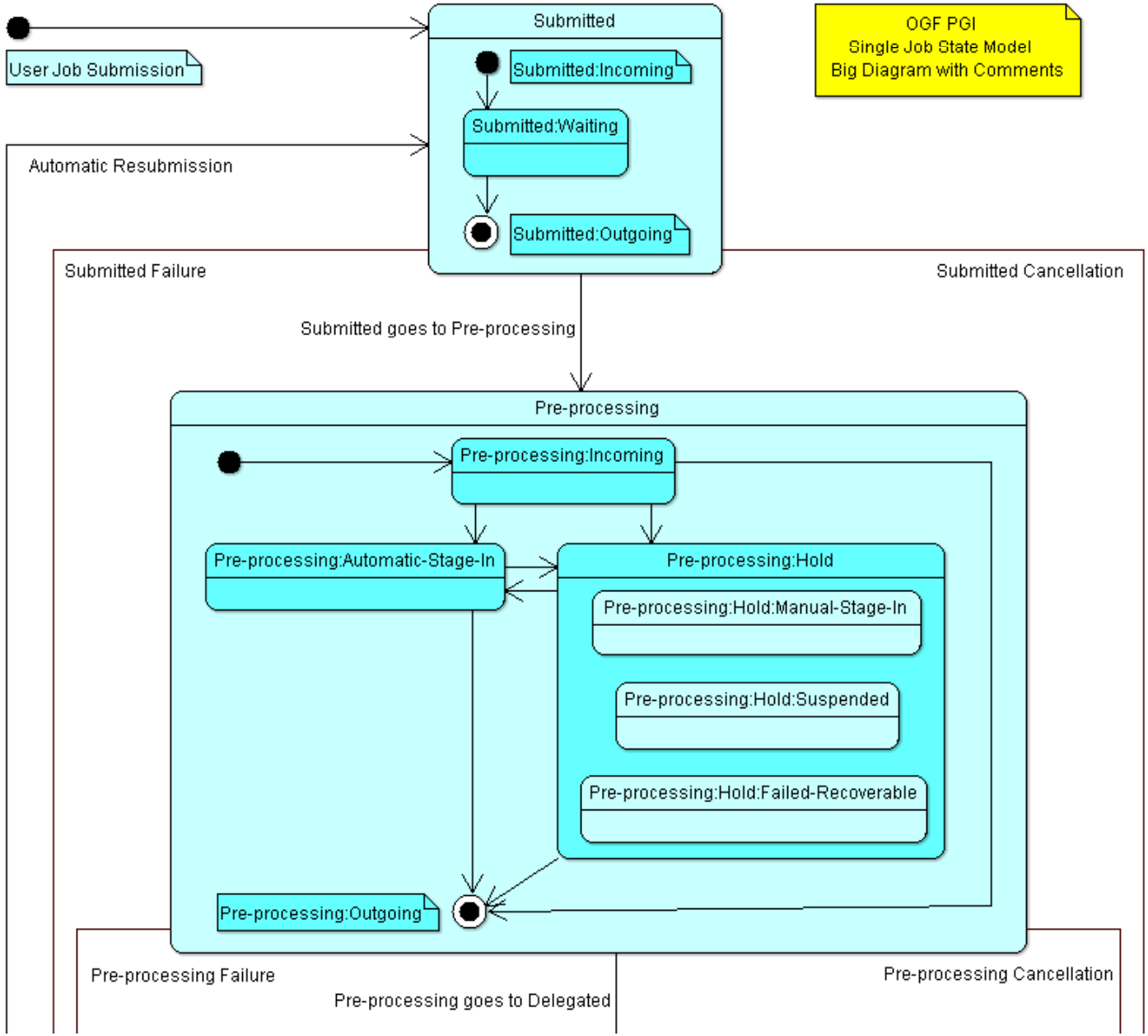
- ‚Recent HPC resource support‘ concept
  - Describe state-of-the art HPC resources with Improved JSDL
  - Covers multi-threading, network connectivity (e.g. torus), libraries,…
- ‚Pre-/post processing‘ concept
  - e-Scientists often require small program (executed non-parallel) before the (parallel) executable starts to run (or after)
- Data-transfer delegation
  - Third-party credentials – how to transfer n different credentials (with different attributes) to a service that should perform a data-staging on behalf of myself later in data-stagings
  - Improved OGSA-BES provides a portType to create a delegated credential in a two phase operation protocol – enables use of different credentials in data-stagings

OGSA-BES Basic State Model

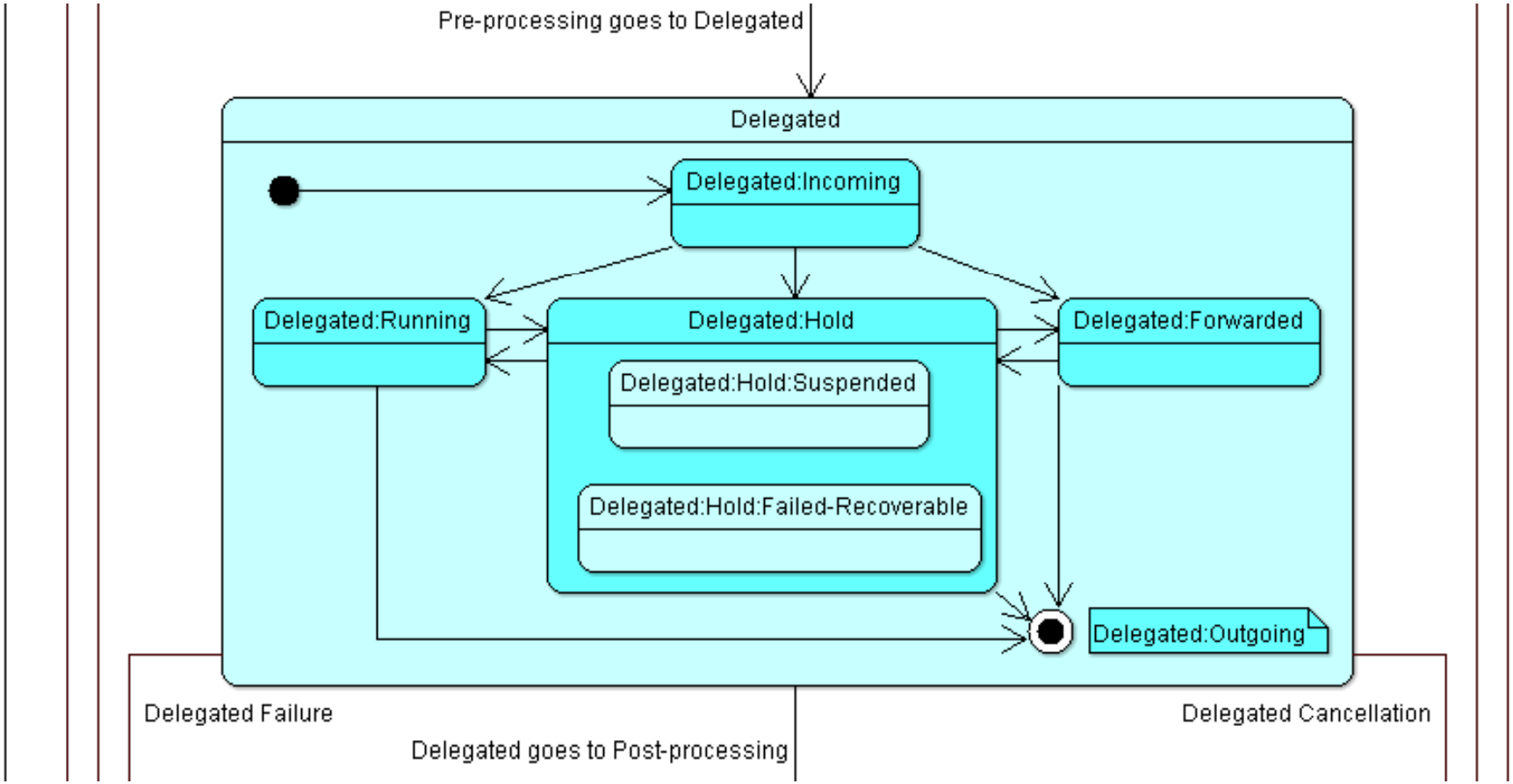
- Simple and plain according to OGSA-BES specification
  - But the means of possible extensions are provided, i.e. ‚state specialization' by putting in sub-states (not mandatory)
  - Production use reveals ‚feedback' to users is important in terms of what the system does (e.g. data-staging for 2 hours not shown)



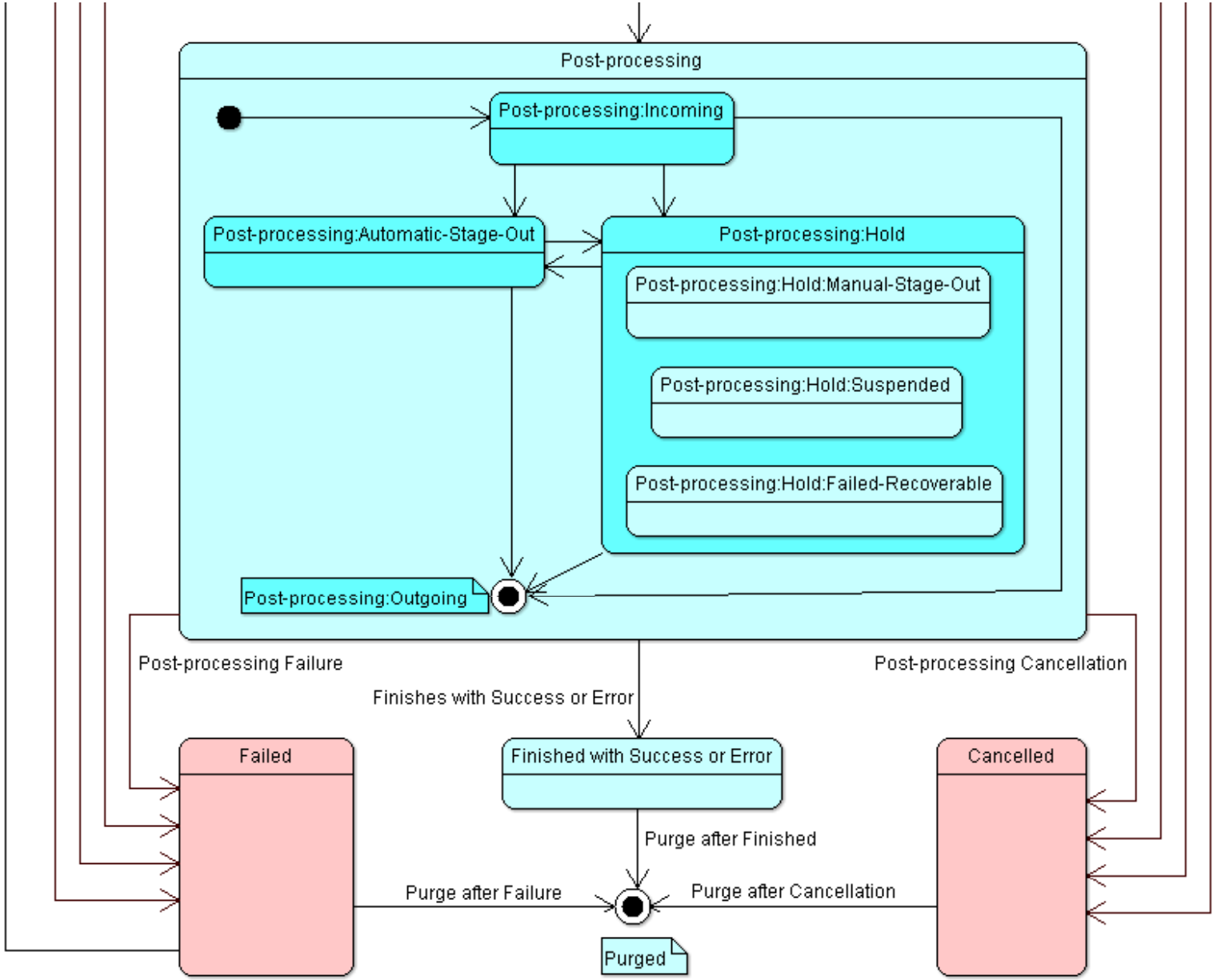
Production State Model (1)



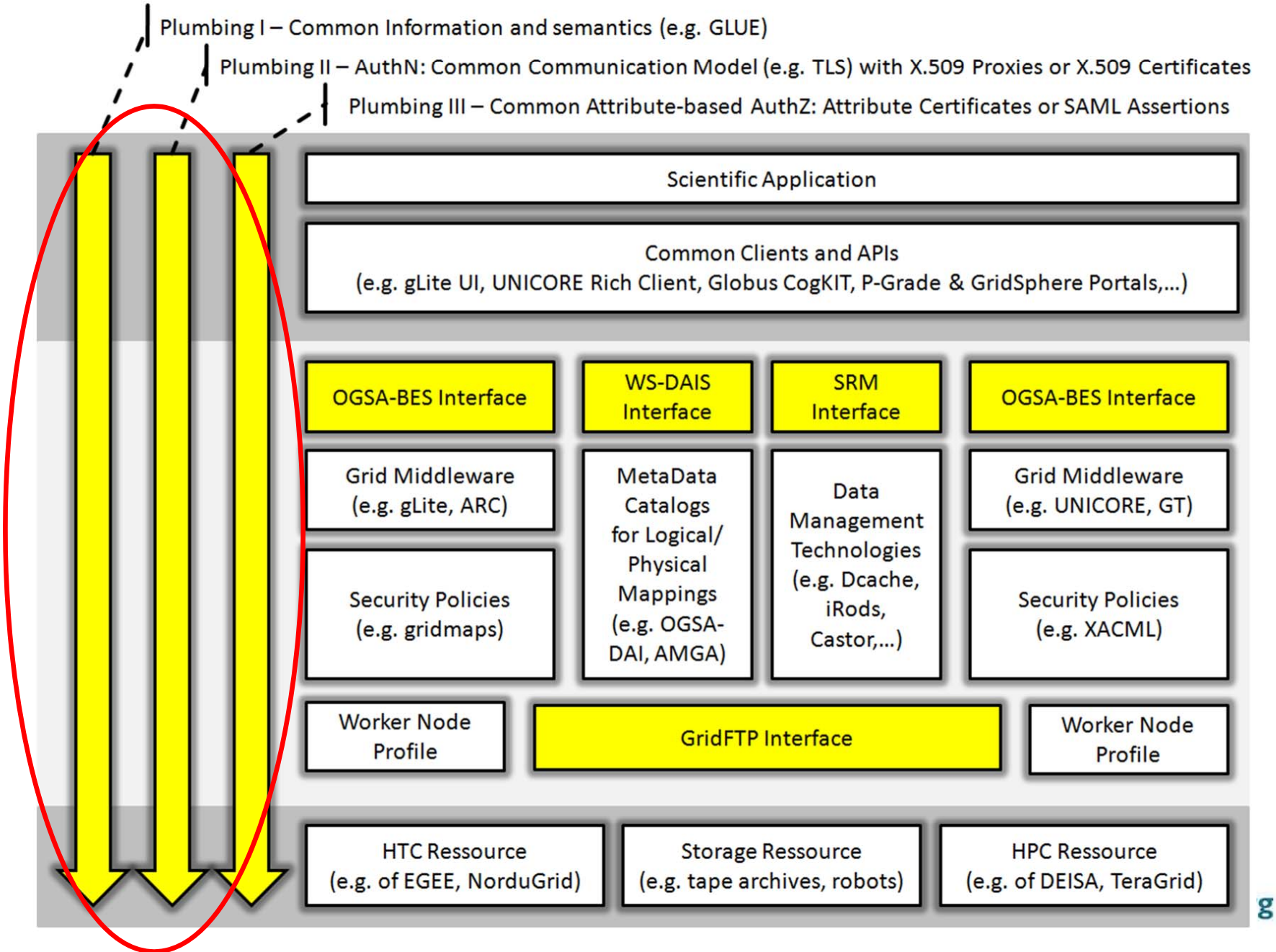
Production State Model (2)



Production State Model (3)



Reference Model Impact

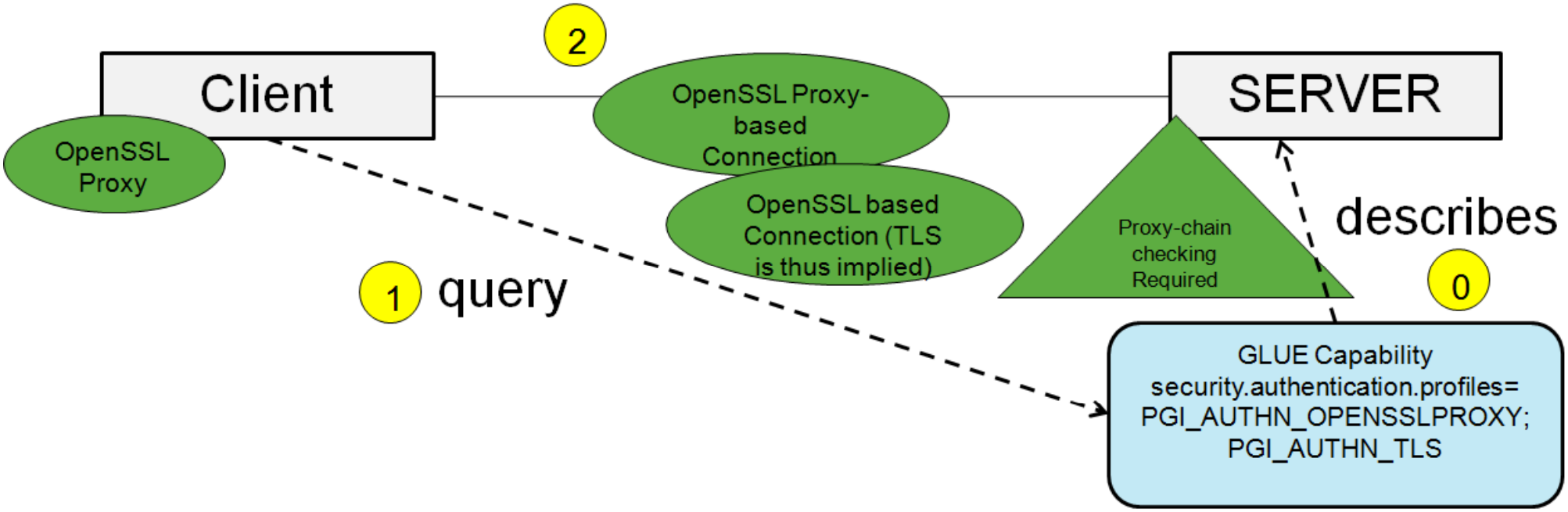


Broader Reference Model Impact

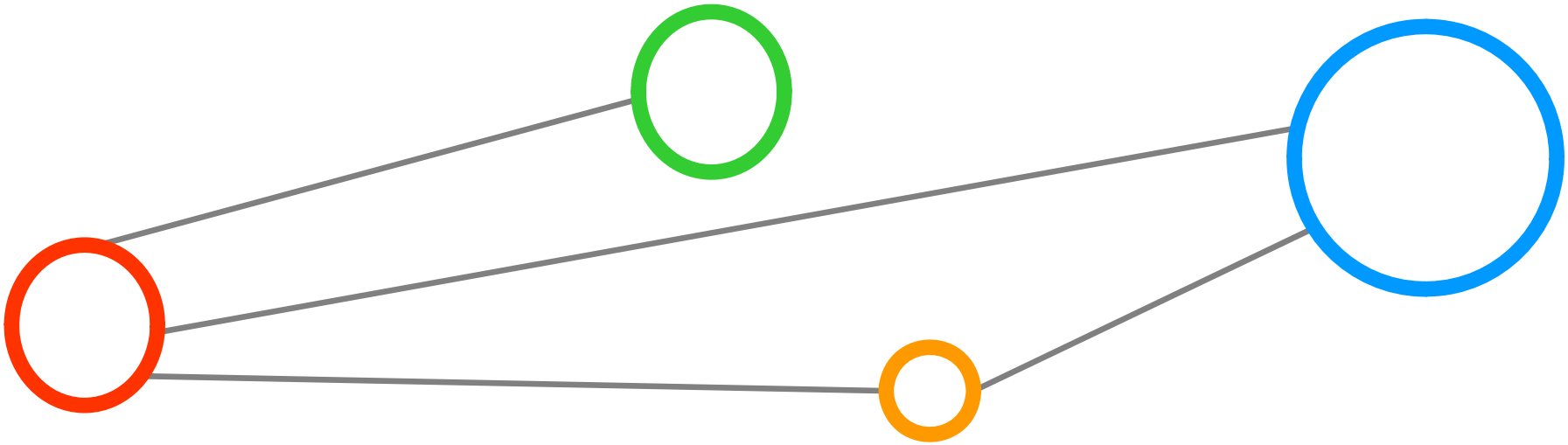
- Vertical Impacts on horizontal standards
  - Some use of ‚vertical standard areas' are orthogonal to the rather horizontal functional interfaces, compare with plumbings in model
  - Also important is that the emerging standards and the additional concepts are well embedded in the broader model ecosystem
  - Sounds trivial, but often major show-stopper since profiling approach is (unfortunately) too flexible that lead to non-interoperable setups
- Security
  - A well-defined security setup that fits production needs
  - E.g. attribute-based authorization is required and used in production
  - Cp. HPC-Profile username/password very rarely used in production
- Information
  - A standardized information model driven by production needs
  - E.g. GLUE2 is all about lessons learned from GLUE1.x (out of OGF) that is deployed since years in the EGEE Grid
  - Already part of functional interface (job context), but also broader → ‚meta-level' describing the properties of a whole site

Broader Reference Model Example

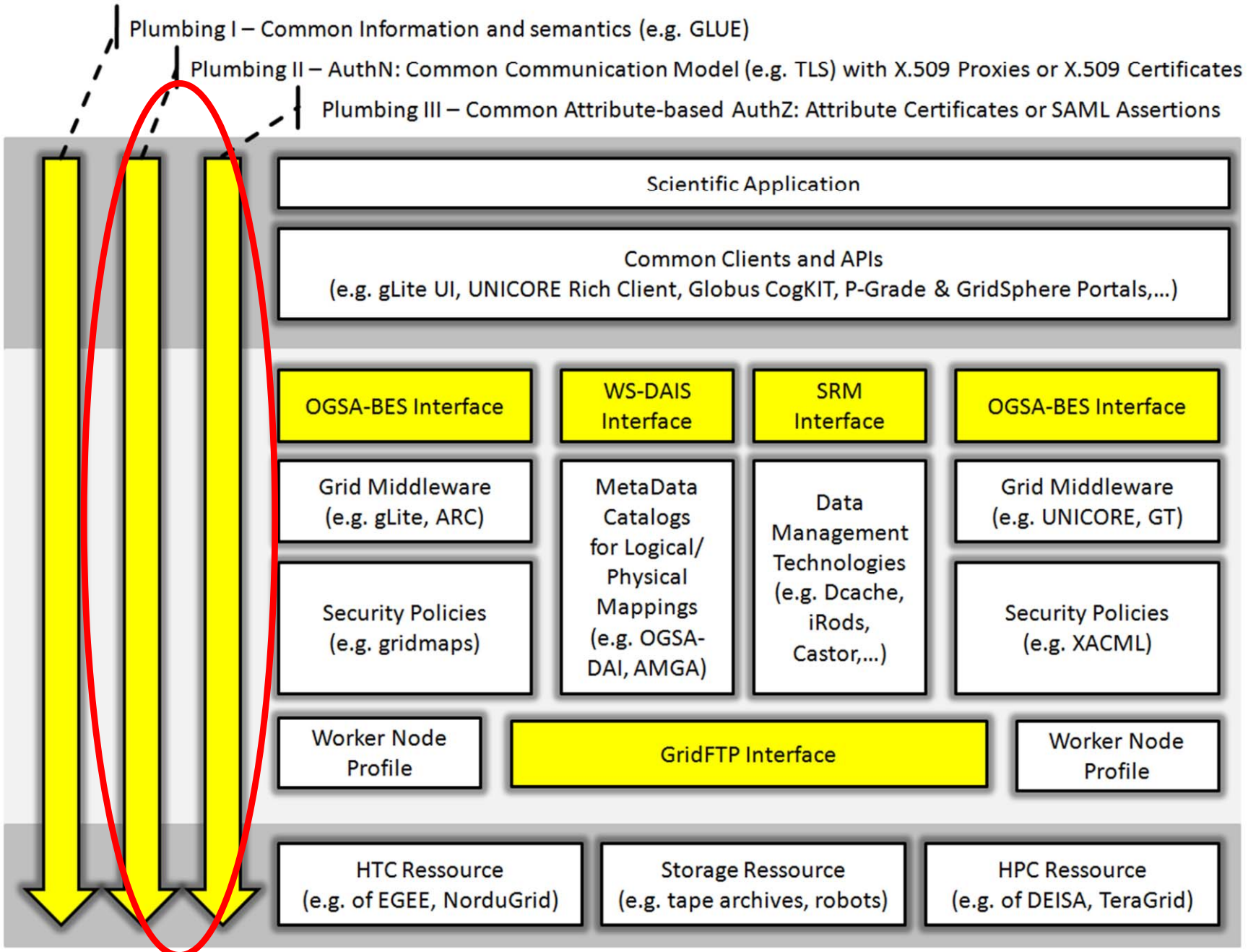
- Security and Information are special…



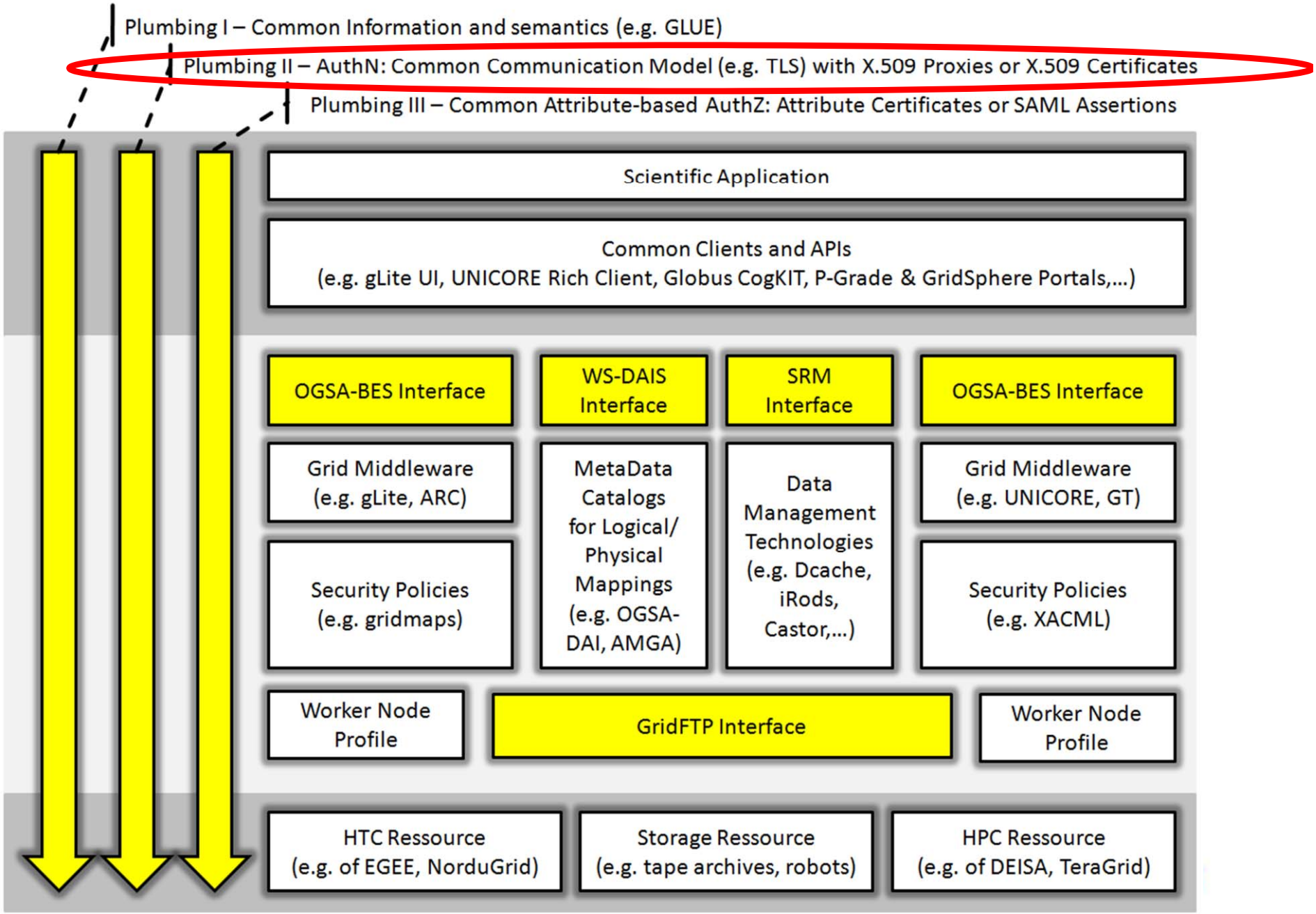
Other Refinement Concepts



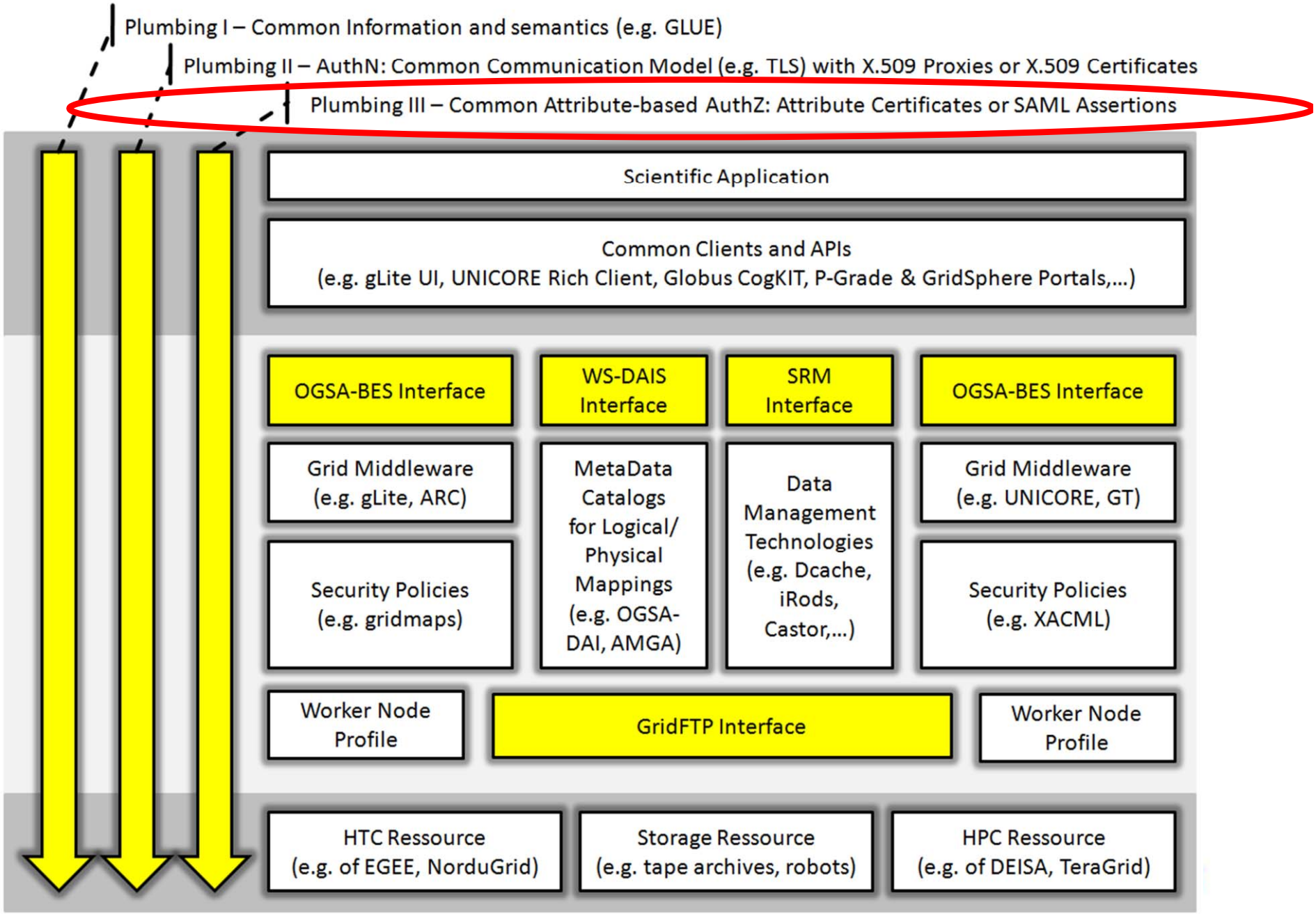
Orthogonal Security: Plumbings



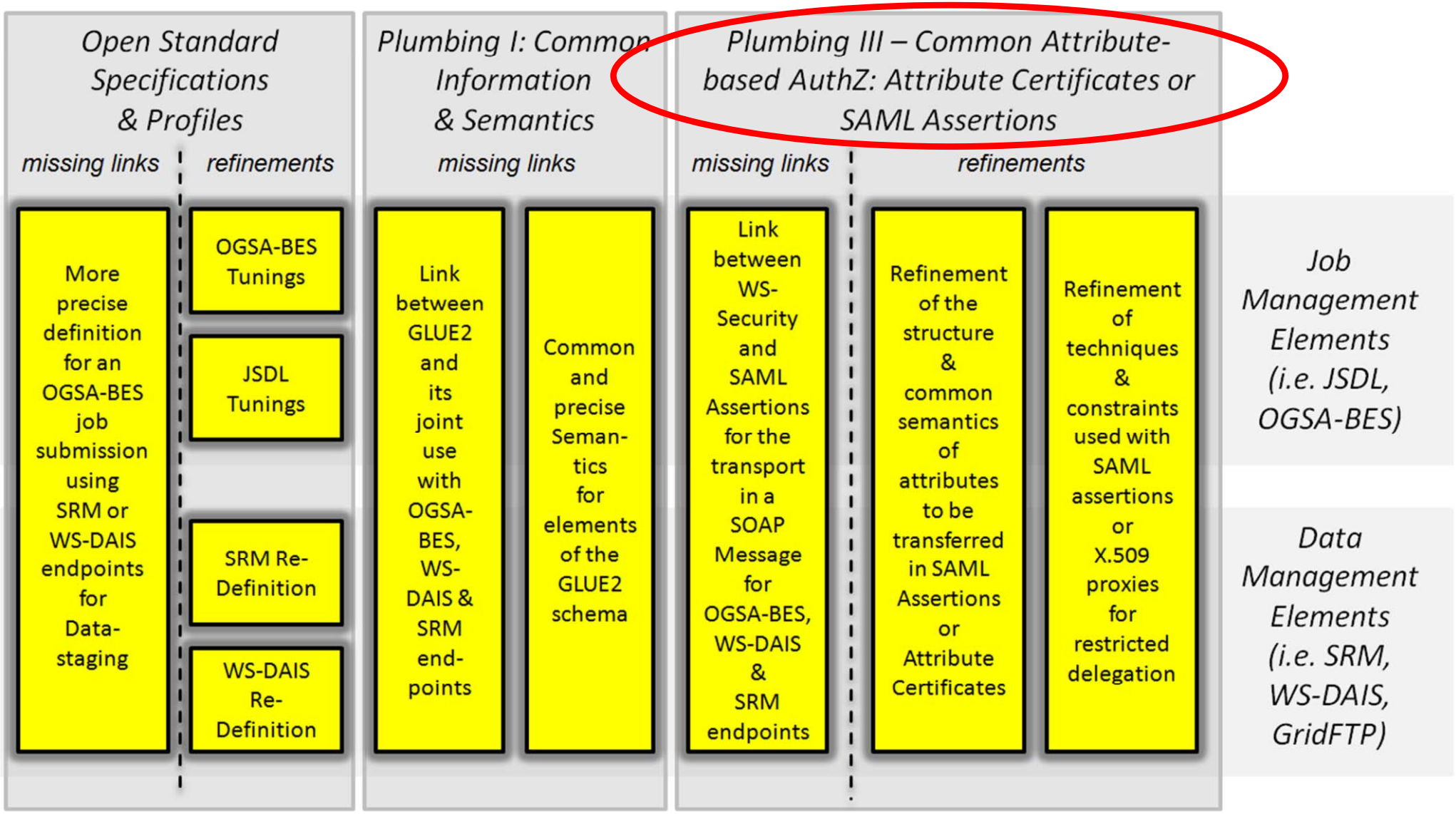
Plumbing II – AuthN w/o GSI



Plumbing III – Attribute AuthZ

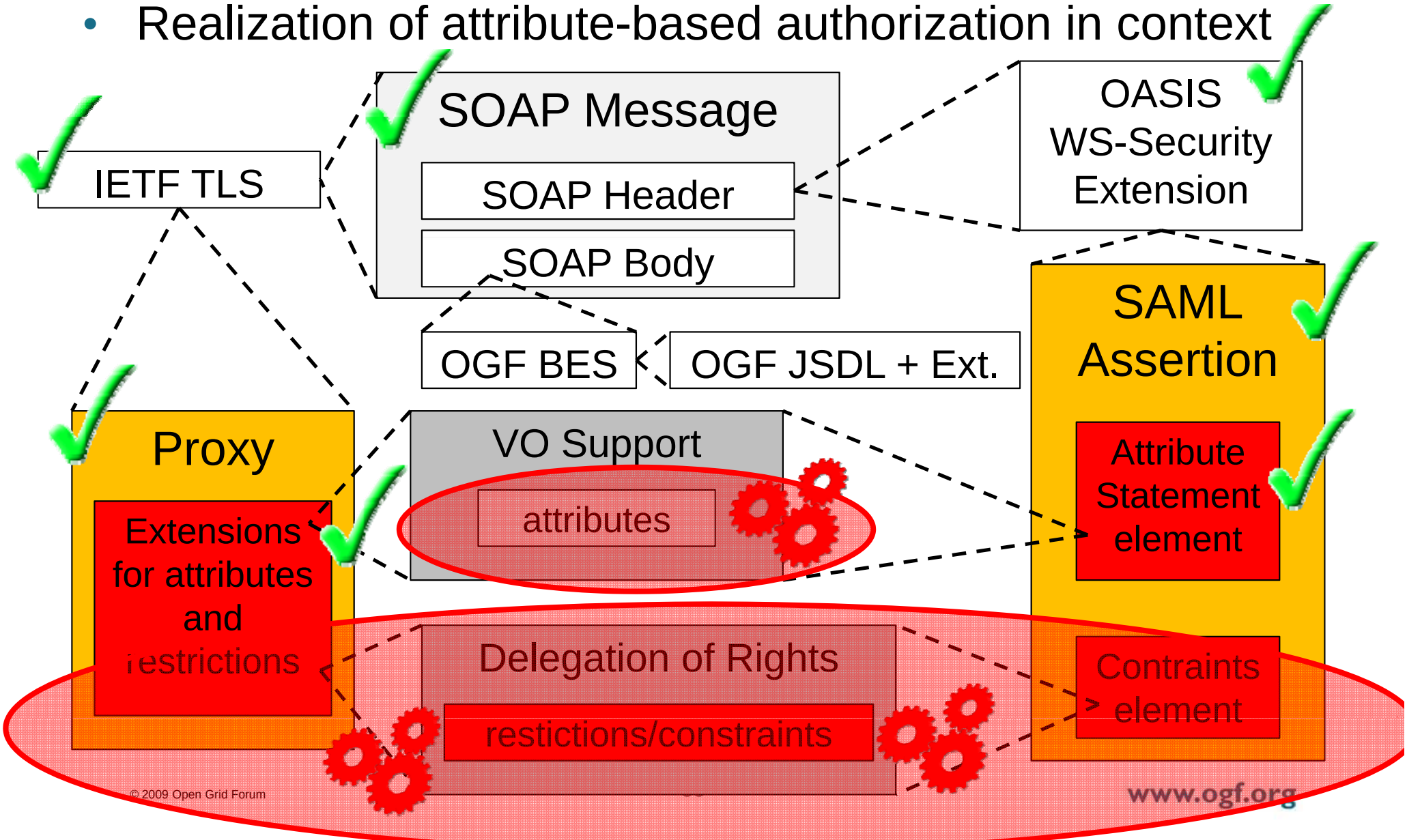


Missing Links & Tunings



Security Refinements in Context

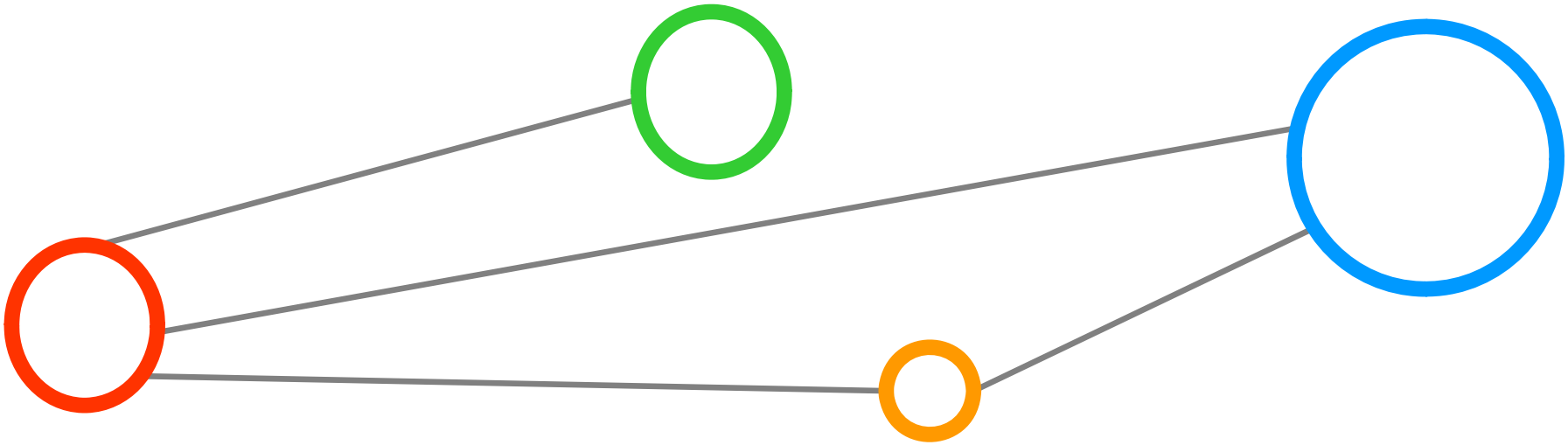
- Realization of attribute-based authorization in context



Data Refinements in Context

- WS-DAIS Refinements
  - We learned a lot of OGSA-DAI that was once a reference implementation of WS-DAI
  - Refinements necessary that are scalable for production use
  - How can be WS-DAI requests used in data staging via OGSA-BES
- Storage Resource Manager (SRM)
  - OGF Specification GFD129, while being defined much earlier
  - Many SRM implementations already exist and are used in production (dCache, Castor, Storm, DPM, …)
  - All implementations tend to be basically interoperable
  - But a significant fraction of the SRM functionality is not interoperable that is often a major showstopper in interoperability use cases
  - Profile which operations work and which operations can be omitted (easier said than done – since storage is complex as computing)
  - Use of two-phase SRM requests (or movements like copyto) during improved OGSA-BES-described data-stagings

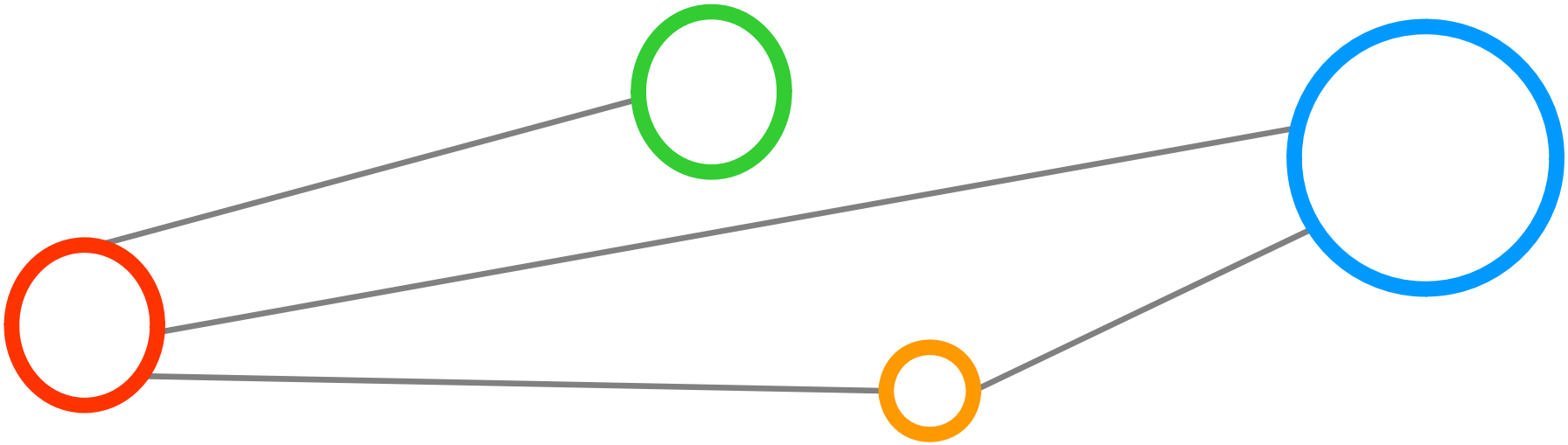
Conclusions



Conclusions

- More and more e-science projects require resources in more than one Grid → Grid interoperability problem
  - Many approaches exist – only production-aware standards help
  - Production Grid Infrastructure (PGI) standardization process
- OGSA exists, but…
  - Hard to maintain, nearly half of all specs defined, missing links,…
- Comparison with history of computer science
  - Cp. XML & SGML, Internet model vs. ISO / OSI model
  - Bottom-up (from production) instead of top-down architecture
- Reference model obtained from real scientific use cases
- Interoperability reference model (or aka profiles) make sense
  - Scientific use cases proof feasibility of initial reference model
  - Might be a milestone towards full OGSA-conformance roadmaps

References



References

To be provided later…

Full Copyright Notice