Bedeutung von Interoperabilität und Standards in Grid Infrastrukturen

LMU SS 2012
Grid Computing

Morris Riedel
Federated Systems and Data
Jülich Supercomputing Centre
Forschungszentrum Jülich
m.riedel@fz-juelich.de

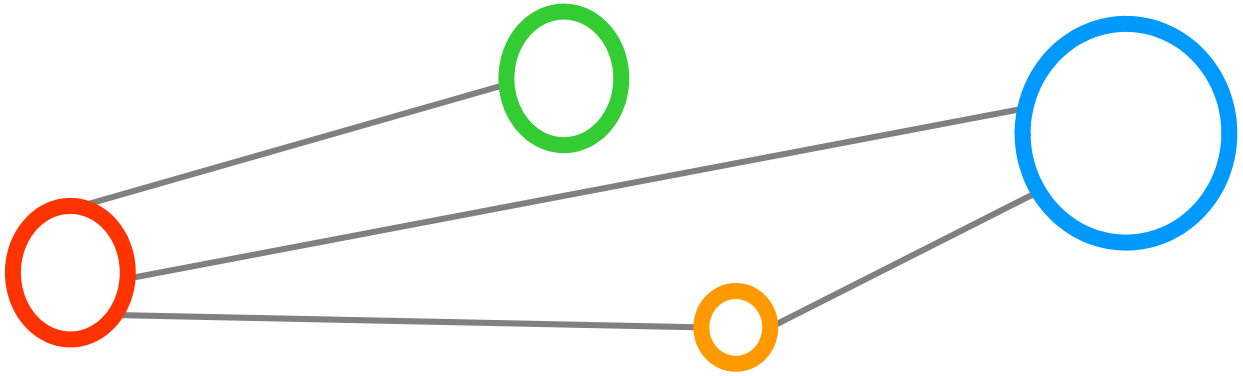
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

JÜLICH FORSCHUNGSZENTRUM

Inhalt mit Lernzielen

- Interoperabilität als Problem
  - Was bedeutet Interoperabilität
  - Interoperabilität von Infrastrukturen als Problem
  - *Beispiel: Inkompatible EU und US Grid Infrastrukturen*
- Verschiedene Lösungsansätze
  - Kurzfristige Lösungsansätze durch Transformationslogik
  - Langfristige Lösungsansätze durch Standardisierung
  - *Beispiel: Middleware von Grid Infrastrukturen*
- Vorteile von Interoperabilität
  - Grid Infrastruktur Betreiber und Benutzer Sicht
  - Software Engineering Prinzipien im Kontext
  - *Beispiel: Benutzung mehrerer Infrastrukturen*
- Zusammenfassung

Interoperabilität als Problem



Was bedeutet Interoperabilität?

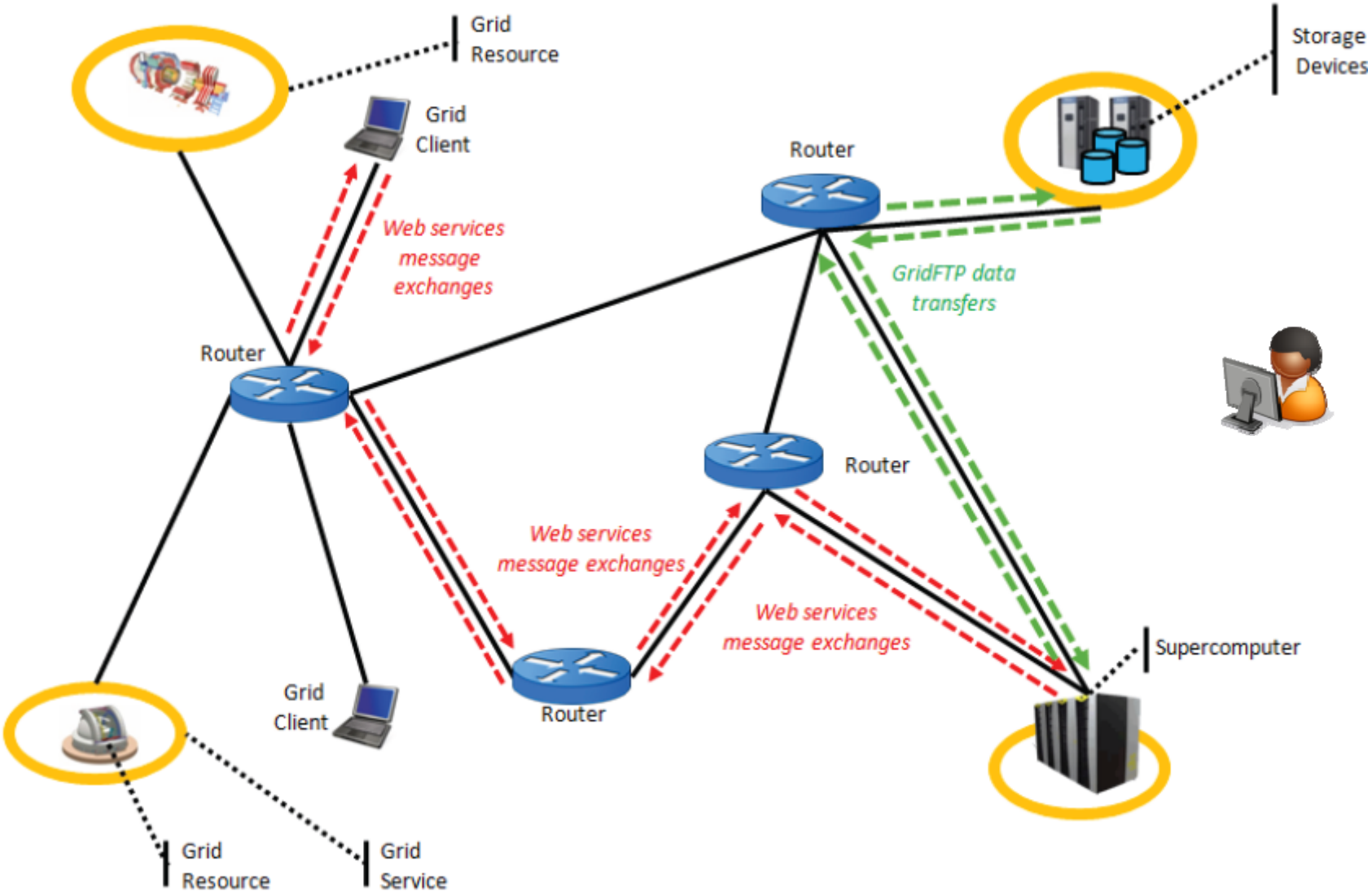
Als Interoperabilität bezeichnet man die Fähigkeit zur Zusammenarbeit von verschiedenen Systemen, Techniken oder Organisationen. Dazu ist in der Regel die Einhaltung gemeinsamer Standards notwendig. Wenn zwei Systeme miteinander vereinbar sind, nennt man sie auch kompatibel.

Interoperabilität ist die Fähigkeit unabhängiger heterogener Systeme, möglichst nahtlos zusammenzuarbeiten, um Informationen auf effiziente und verwertbare Art und Weise auszutauschen bzw. dem Benutzer zur Verfügung zu stellen, ohne das dazu gesonderte Absprachen zwischen den Systemen notwendig sind.

Wikipedia, 07/2012

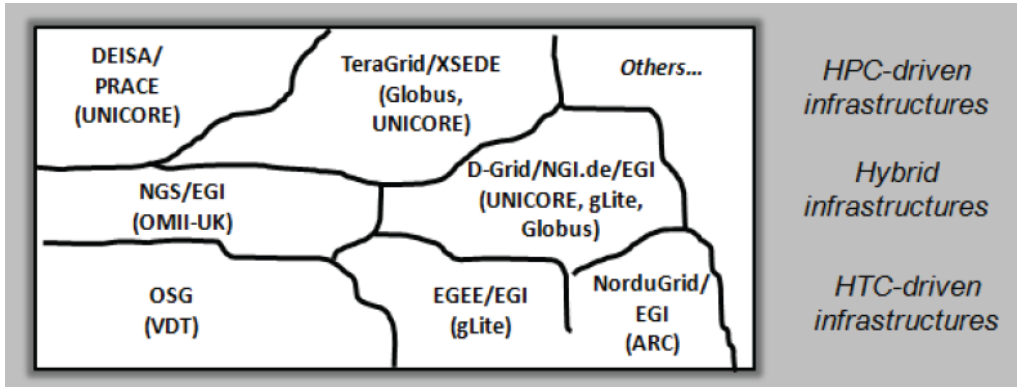
- Begriffsbildung
  - Abgeleitet von lateinisch opera=Arbeit und inter=zwischen
  - Im Jahr 2000 hatte IEEE jedoch vier unterschiedliche Definitionen
- Kontext Grid
  - Grid Infrastrukturen sollen zusammen vom Benutzer nutzbar sein
  - Grid Middleware Systeme sollen (alle) zusammen funktionieren

Beispiel: Nachrichtenaustausch



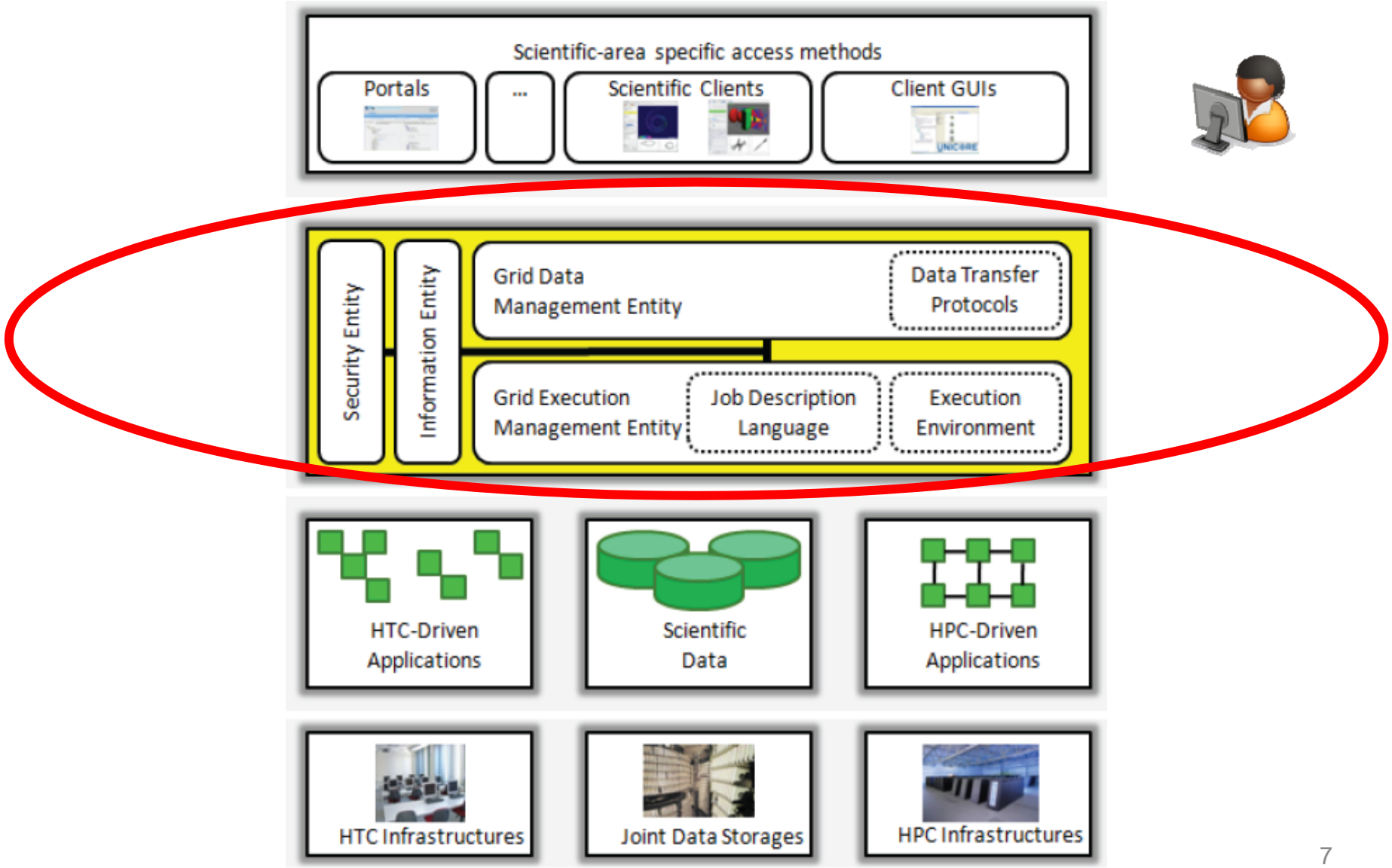
Benutzung mehrerer Grid Infrastrukturen

- Häufiger Anwendungsfall
  - Benutzer wollen mehr als eine Grid Infrastruktur nutzen
  - Grid Infrastrukturen bieten unterschiedliche attraktive Funktionalität



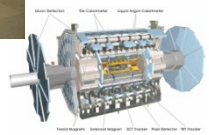
- Auftretende Probleme → Barrieren zwischen Infrastrukturen
  - Unterschiedliche Sicherheitsmechanismen: Öfters Passwort Eingabe
  - Inkompatible Protokolle/Middleware: Kein Nachrichtenaustausch
  - Andere Datenformate: Semantik und Informationsgehalt anders
  - Limitierte Benutzung: Middleware mit “anderen/weniger” Funktionen

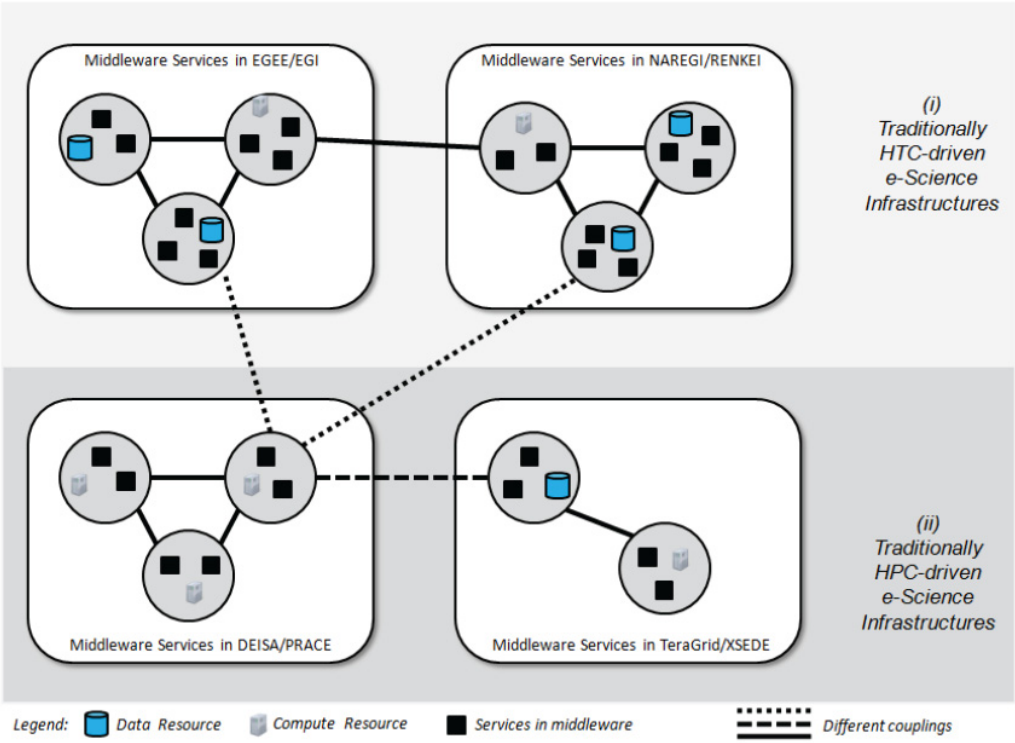
Interoperabilität in Technischen Bereichen



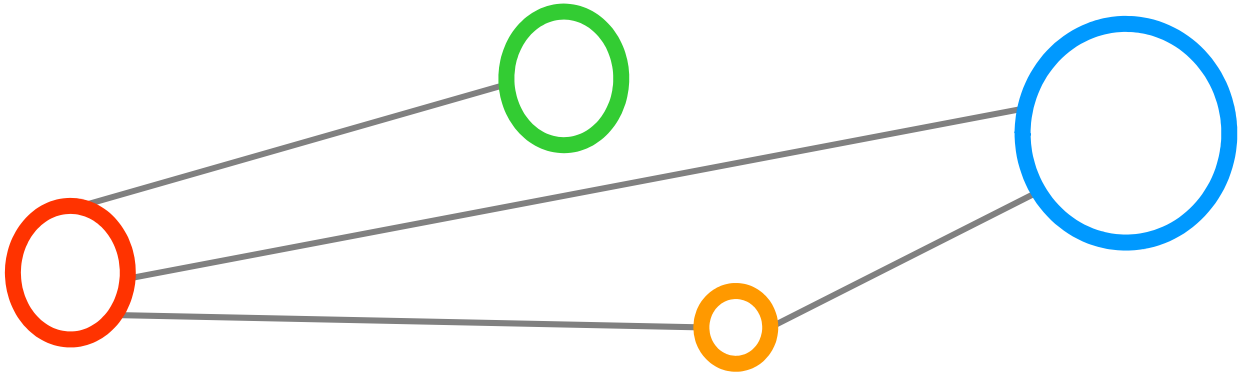
Beispiel: EU und US Grid Infrastrukturen

- Unterschiedliche Funktionalität
  - US XSEDE: Supercomputer (und teilweise Campus-Computer)
  - EU PRACE: Supercomputer und gutes Netzwerk zwischen Zentren
  - EU EGI: Campus-Computer und Daten aus LHC



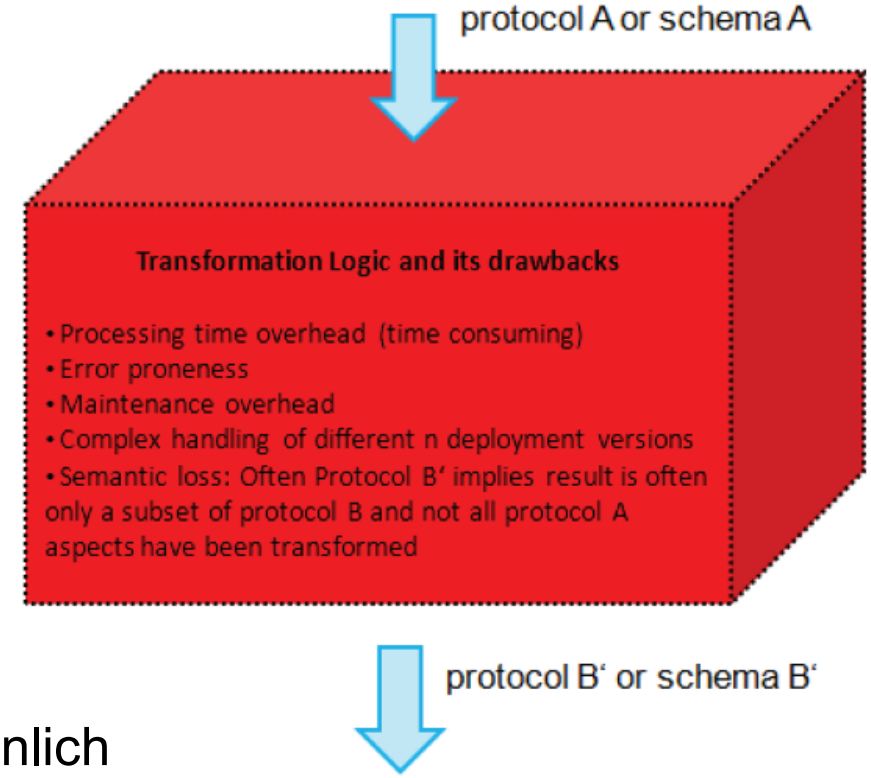


Verschiedene Lösungsansätze



Kurzfristige Lösungsansätze

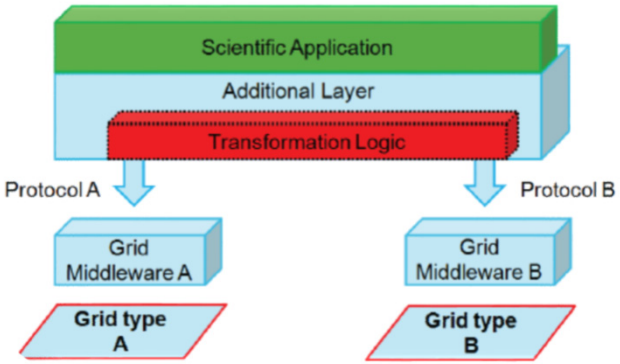
- Benutzung einer “Transformationslogik”
  - Prinzip: Protokoll A nach Protokoll B “übersetzen”



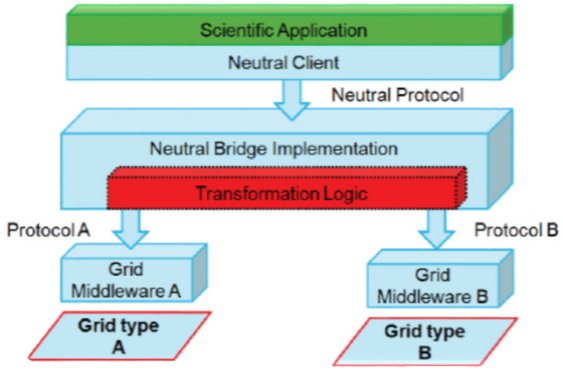
- Nachteile
  - Kurzfristige Lösung da Protokolländerung wahrscheinlich
  - Schlechte Wartbarkeit (engl. “maintenance”) und oft fehlerhaft
  - Deployment auf Infrastrukturen in unterschiedliche Versionen
  - Oft Informationsverlust (bspw. unterschiedliche Datenmodelle)

Kurzfristige Lösungsansätze Überblick

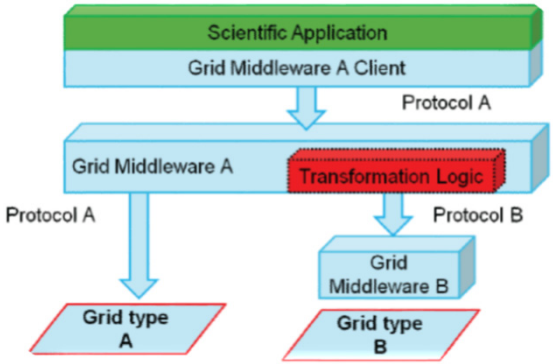
- “Transformationslogik” wird unterschiedlich implementiert
  - Aber Prinzip bleibt: Protokoll A in Protokoll B transformieren



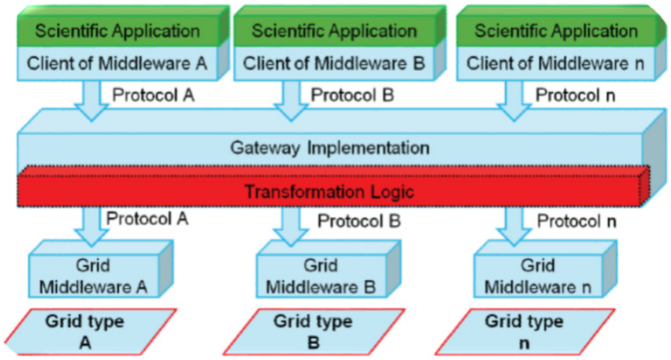
Additional Layer



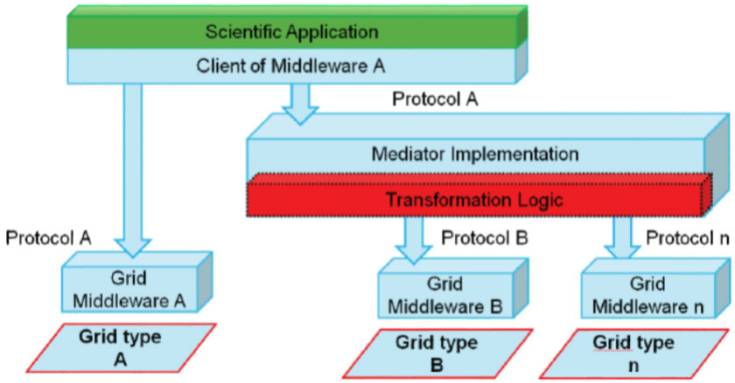
Neutral Bridge



Adapter



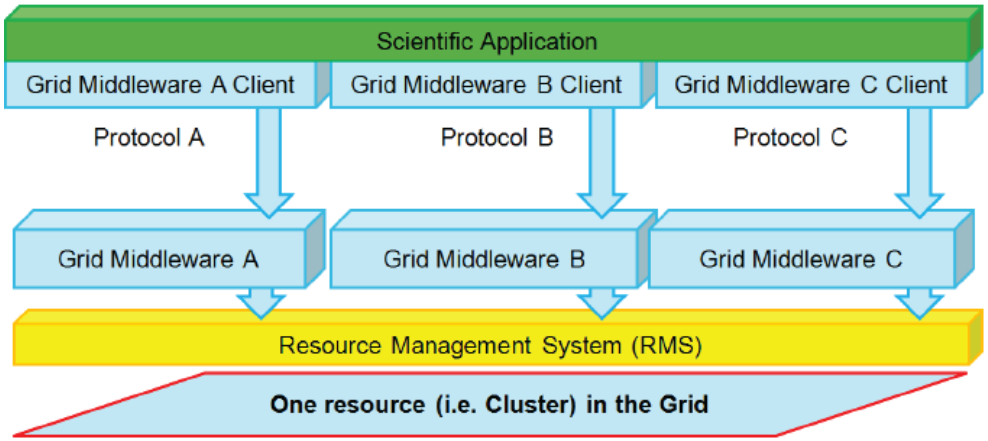
Gateway



Mediator

Exkurs: n Middleware als Deployment

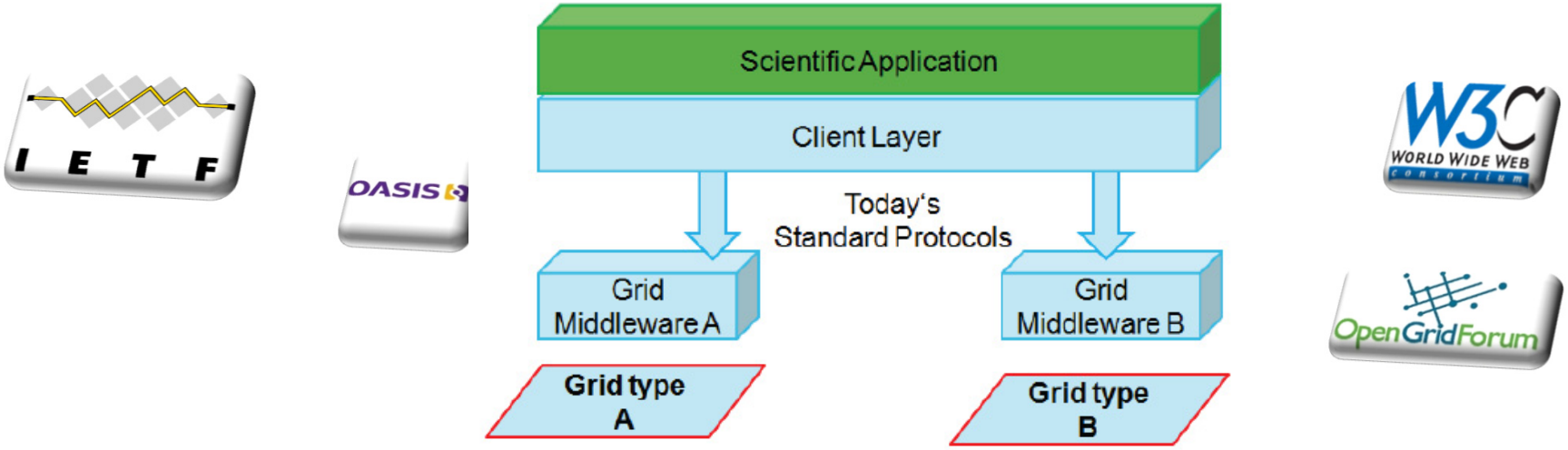
- Weitere Methode: Alle Middleware Produkte installieren
  - Idee: Transformationslogik nicht mehr notwendig



- "Eher gutes Beispiel wie man es nicht machen sollte"
  - Aber: Unter Umständen schnellste Lösung (keine Entwicklung)
- Nachteile
  - Hoher Aufwand durch Installation/Kenntnisse von jeder Middleware
  - Sehr schlechte Wartbarkeit (engl. "maintenance")

Langfristige Lösungsansätze: Standards

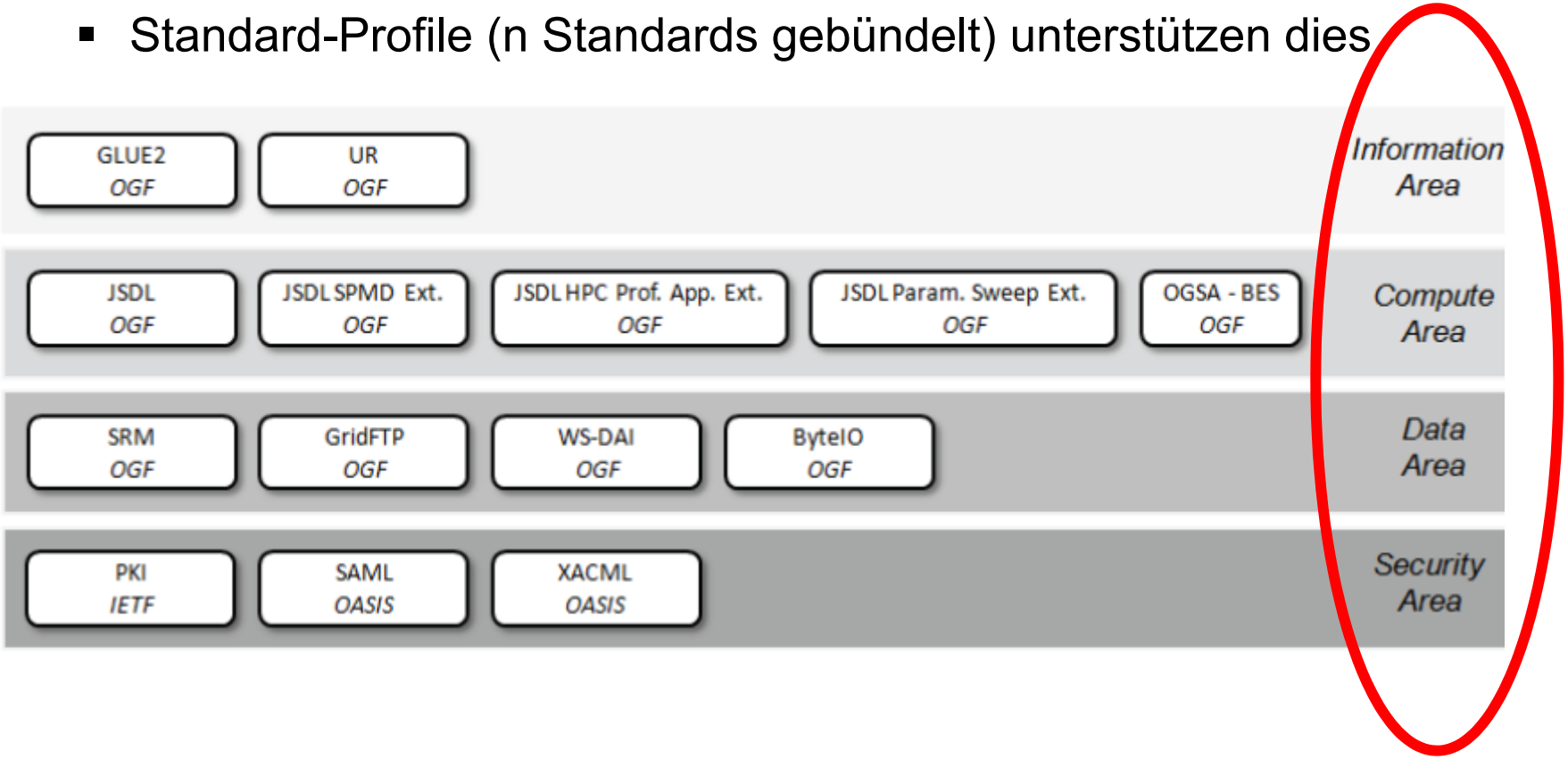
- Benutzung von offenen Standards
  - Prinzip: Einigung auf ein Basisprotokoll mit Kernfunktionalitäten
  - Idee: Transformationslogik nicht mehr notwendig



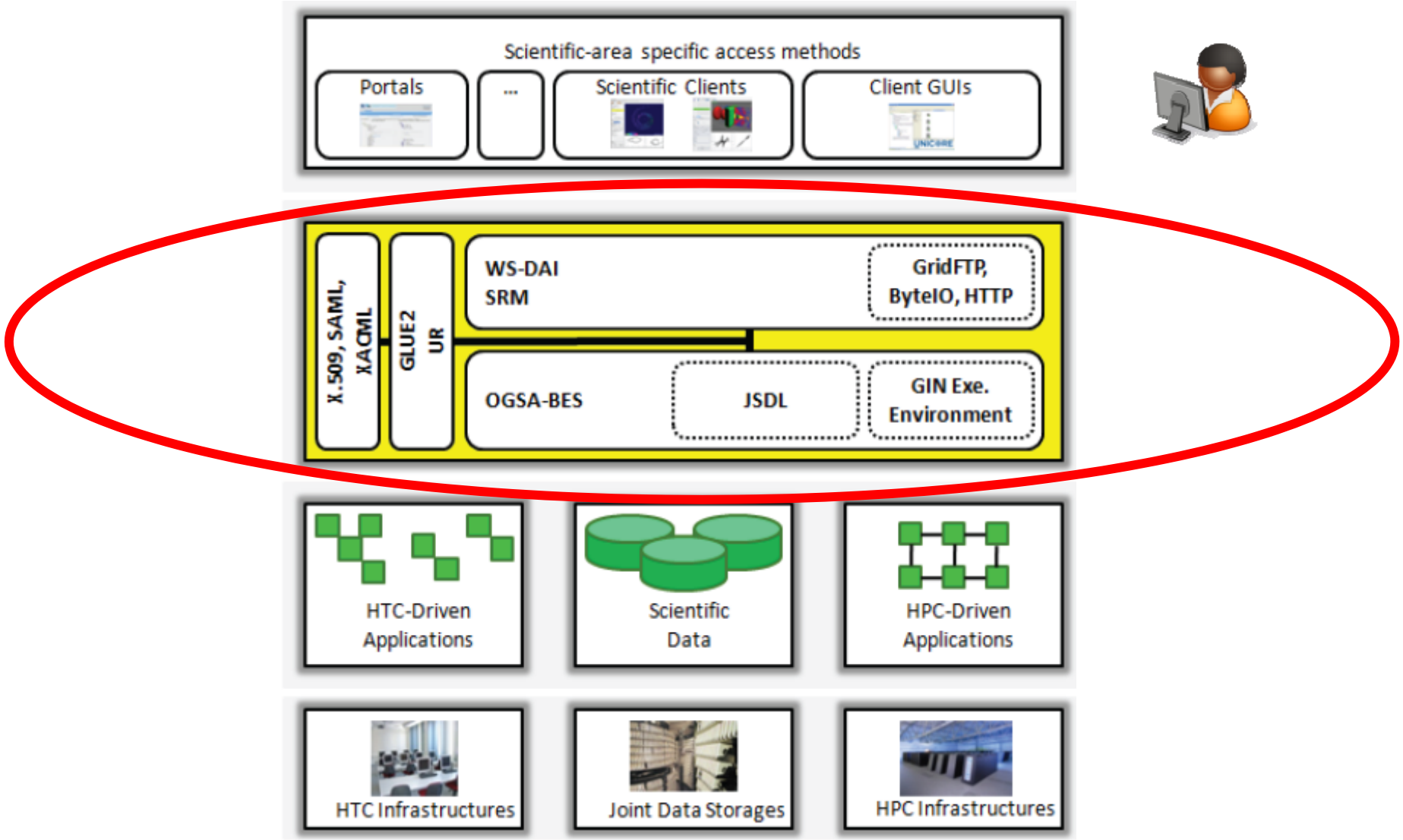
- Nachteile
  - Standardisierungsorganisationen bieten überlappende Standards
  - Einigung auf gewissen Standard zwischen Herstellern nicht leicht
  - Gemeinsame Standardisierung von den Herstellern dauert

Achtung: Unterschiedliche Bereiche

- Ein Standard allein führt nicht unbedingt zur Interoperabilität
  - Richtige Kombination unterschiedlicher Standards ist wichtig
  - Standard-Profile (n Standards gebündelt) unterstützen dies

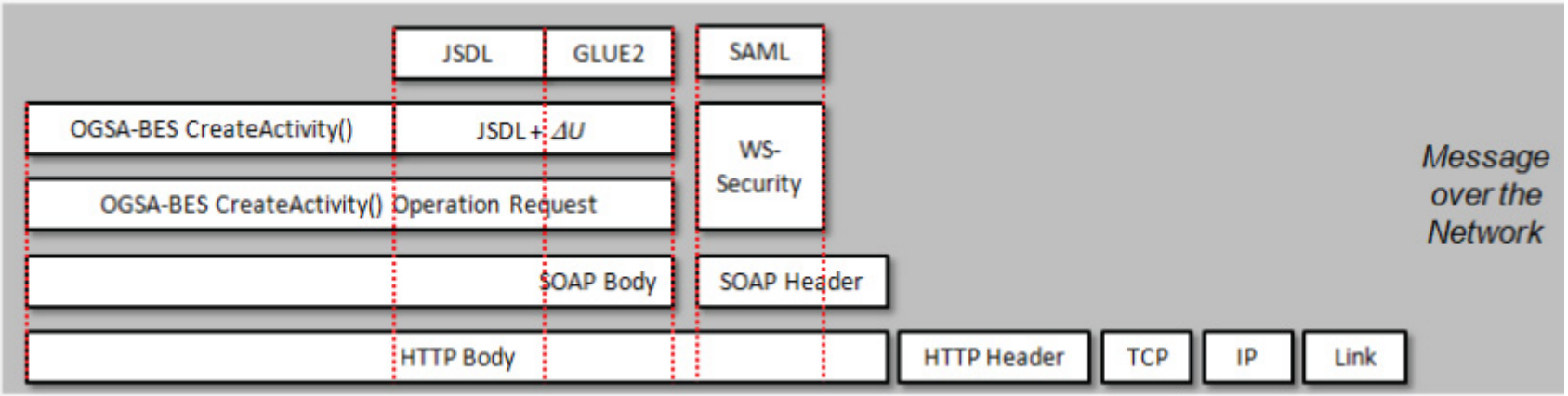


Standards in Technischen Bereichen



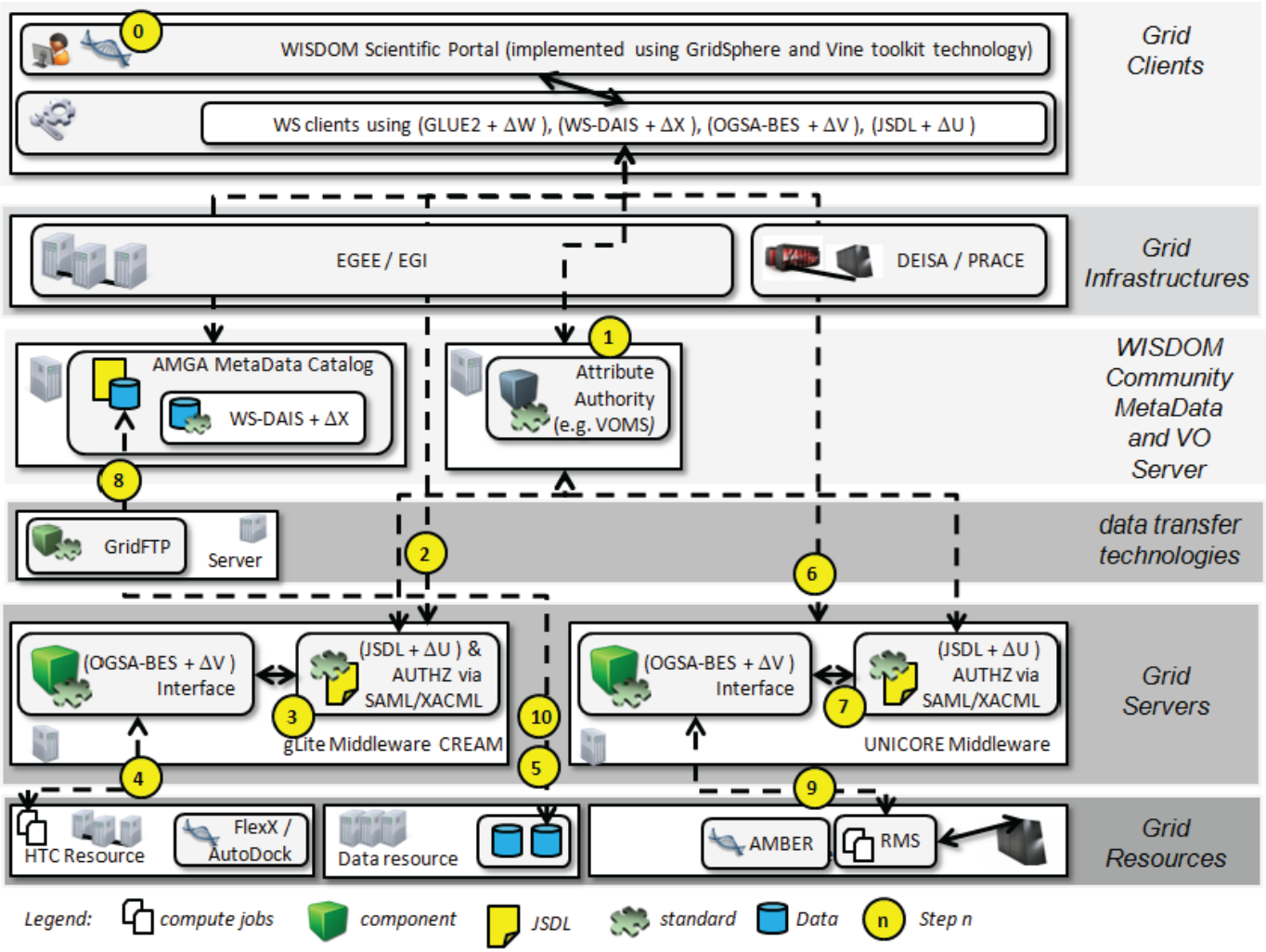
Beispiel: Kombination von vier Bereichen

- Nachrichtenaustausch beinhaltet vier technische Bereiche
  - JSDL & OGSA-BES: Instruktionen für die Benutzung der Ressource
  - GLUE2: Informationen über die jeweilige ausgewählte Ressource
  - SAML & WS-Security: Sicherheitsangaben über den Benutzer
  - (SOAP/HTTP, etc. als Nachrichtenprotokollstack für Web services)



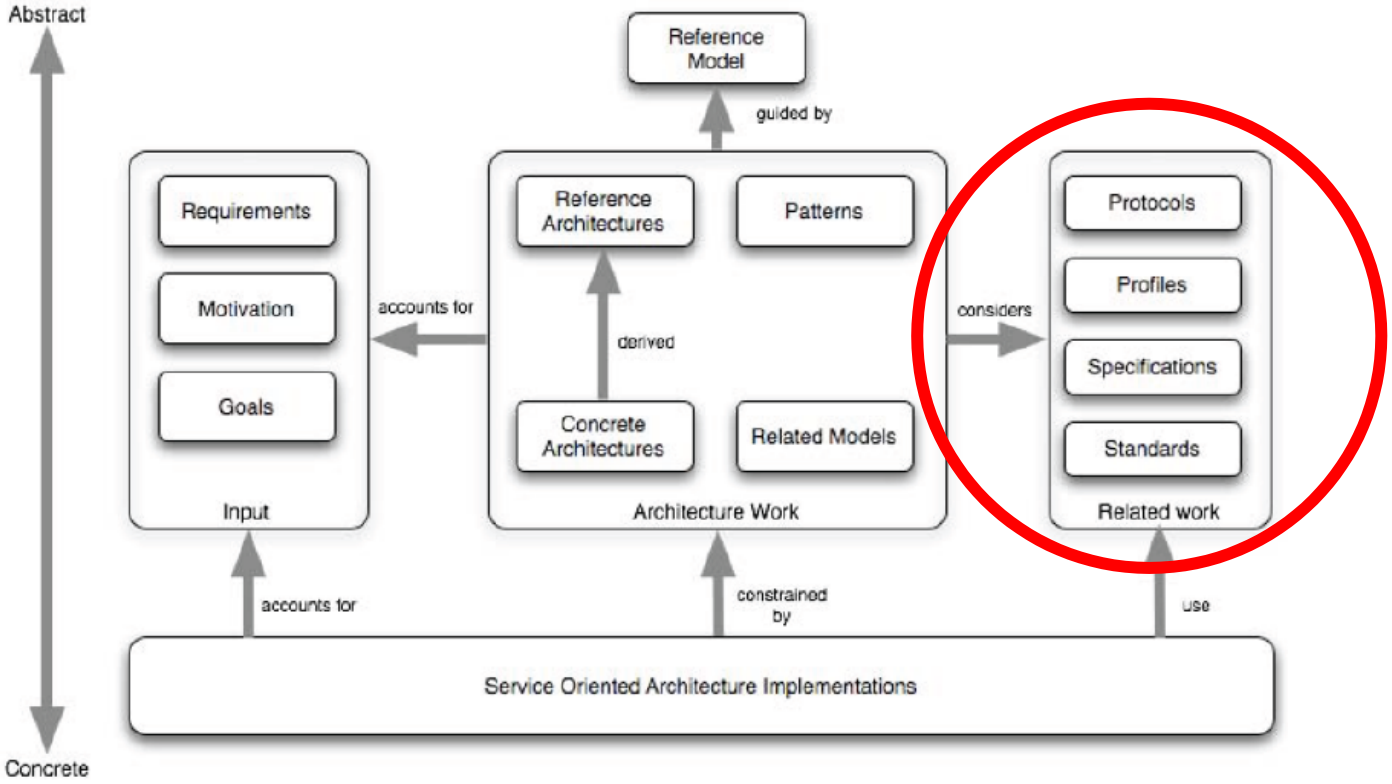
Beispiel: Middleware Interoperabilität

- Nutzung von Standards bei wissenschaftlicher Anwendung

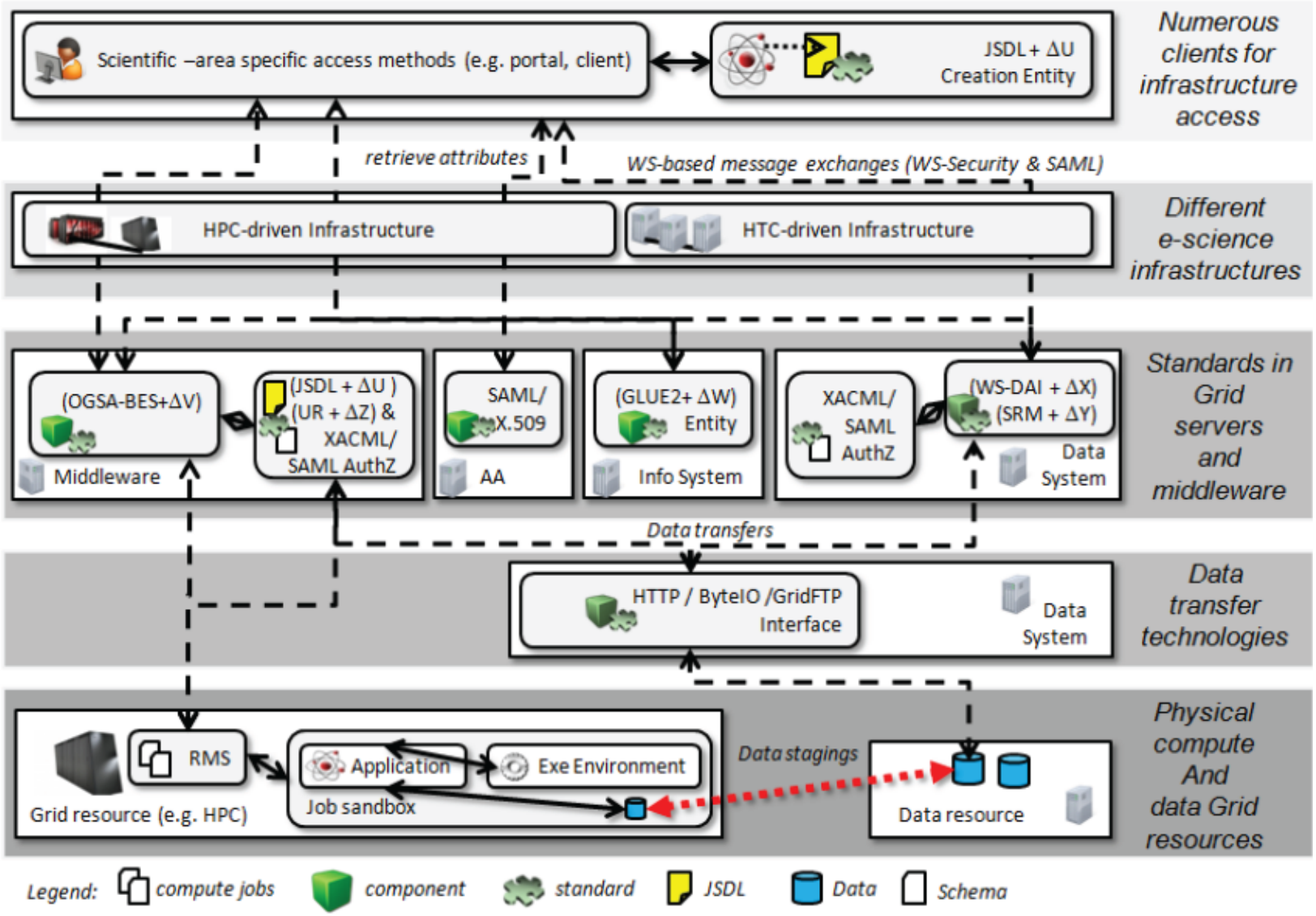


Langfristige Lösungsansätze: Planung

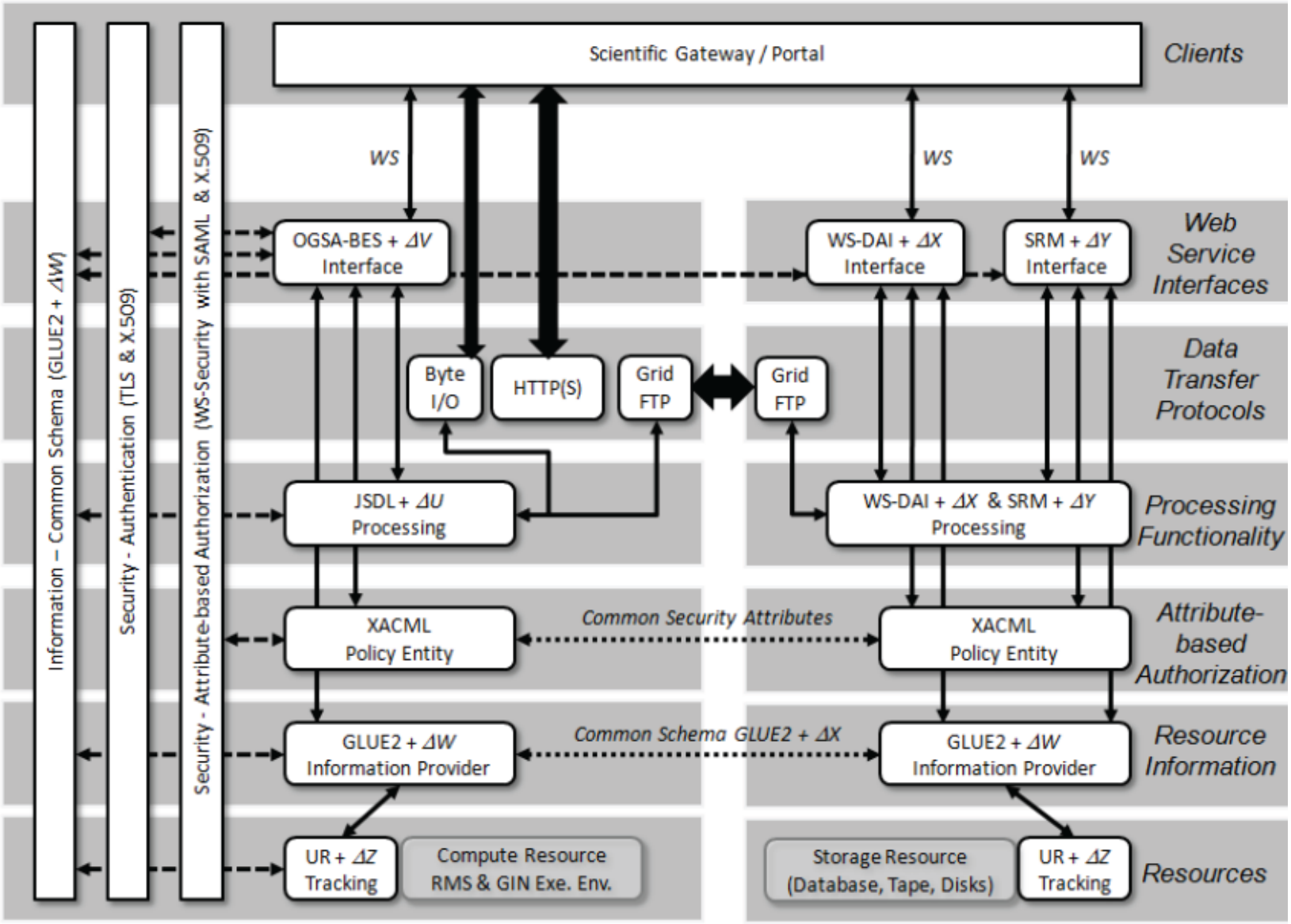
- Langfristige Planungen als Ergänzung zu Standards
  - Vorhaben von Herstellern dokumentieren (engl. “roadmaps”)
  - Idee: Benutzung mehrerer Standards zusammen (mit Zeitangabe)
  - Prinzip: Referenz- und Referenz-architekturen wegen Komplexität



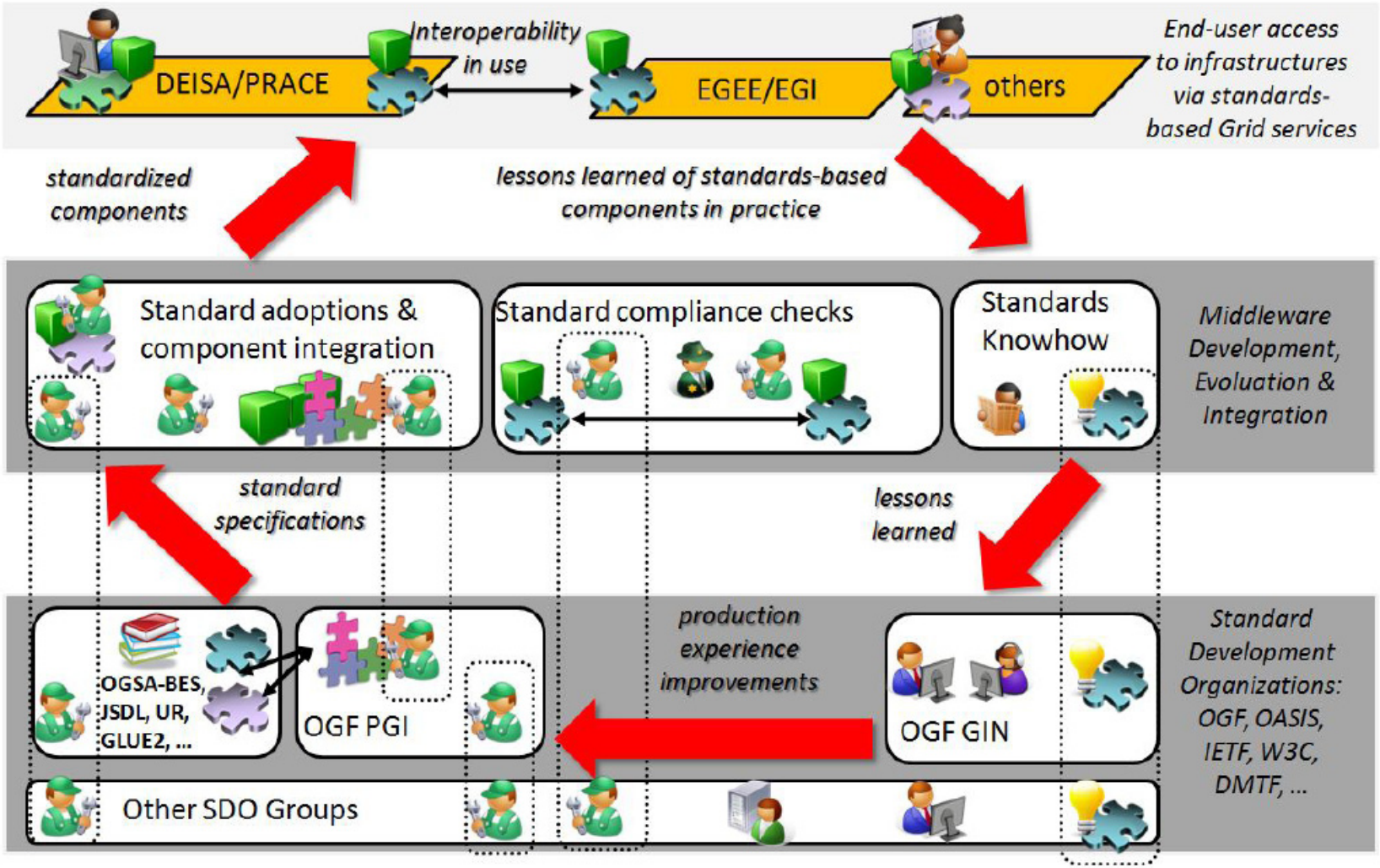
Beispiel: Komplexität mit n Standards



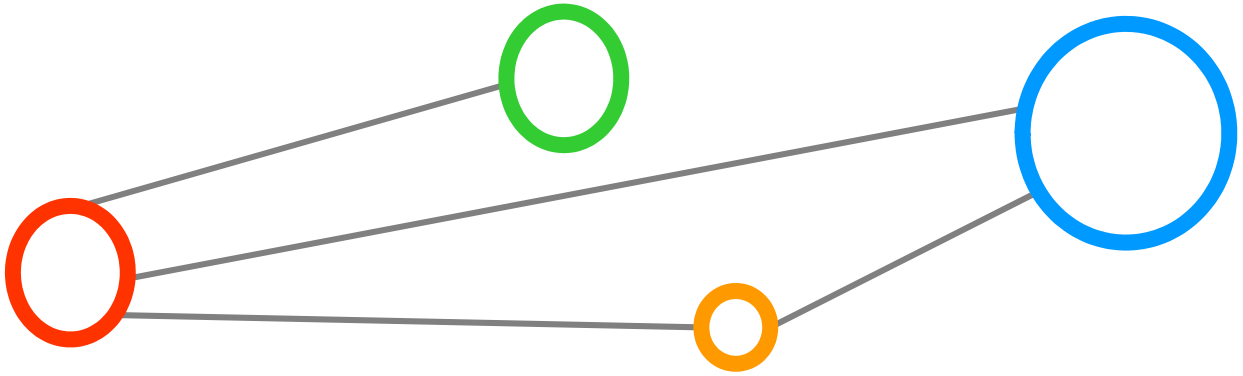
Beispiel: Mögliche Referenzarchitektur



Verbesserung der Standards (+ ΔX)



Vorteile von Interoperabilität

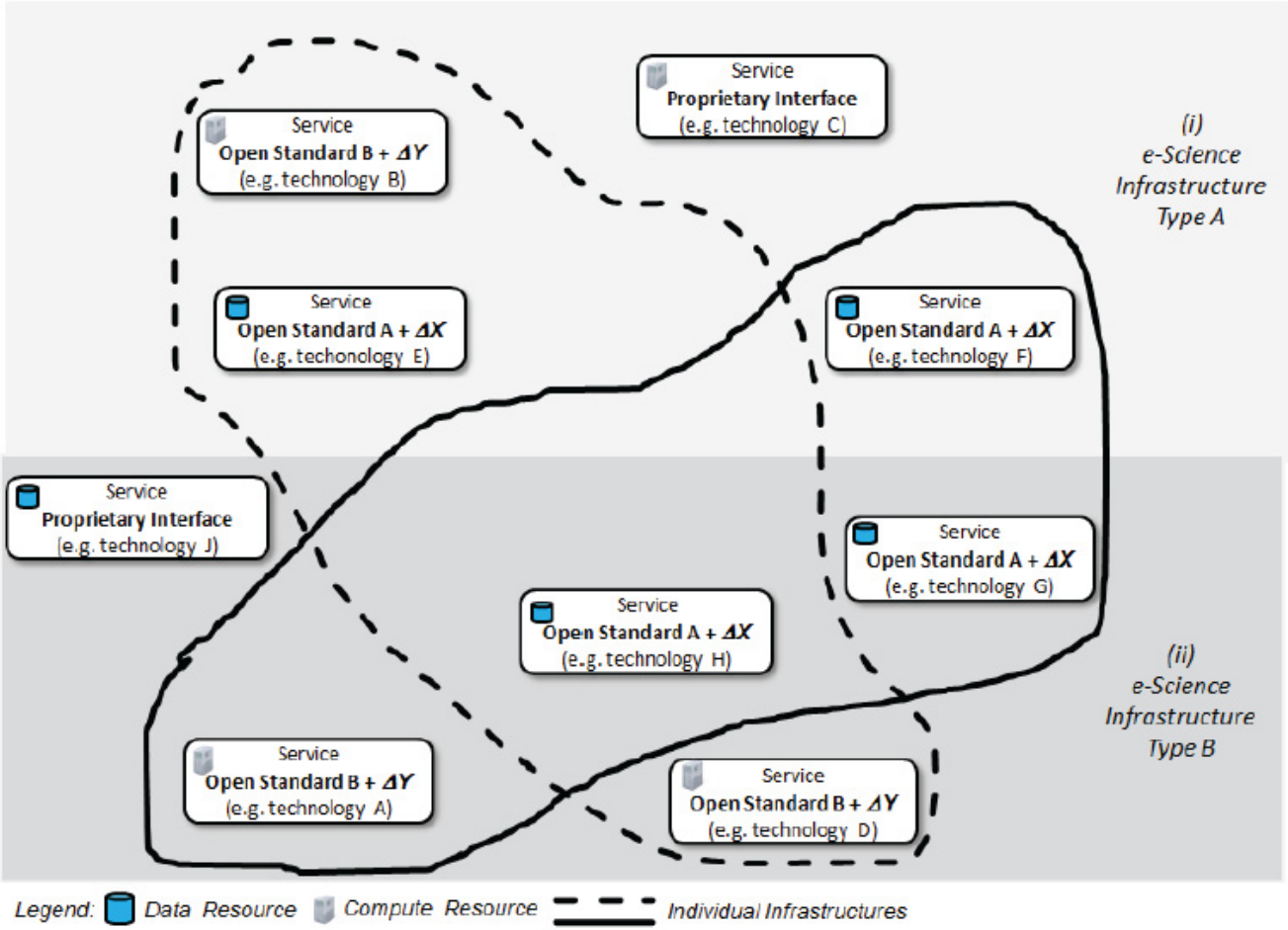


Vorteile: Grid Infrastruktur Betreiber (1)

- Keine Herstellerabhängigkeit (engl. “vendor-locks”)
  - Basisfunktionalität ist interoperabel mit anderer Software
  - Zusätzliche Software Funktionen machen Hersteller attraktiv
  - Weniger Risiko: Hersteller bestimmt nicht so leicht den “Preis”
- Auswechselbarkeit der Software
  - Zugangsmechanismus bleibt gleich wenn Middleware sich ändert
  - Software kann man (prinzipiell) ersetzen wenn interoperabel
  - Gleicher Client greift auf unterschiedliche Middleware zu
- Wiederverwendung
  - Wichtig: Standards sollten ‘atomar’ sein um das zu fördern
  - “Standard Profile” können mehrere atomare Standards vereinen
  - Kombinationen von Standards aus unterschiedlichen Bereichen
  - Gleichen Standard für verschiedene Probleme/Ressourcen nutzen

Beispiel: Keine Vendor-locks

- Unterschiedliche Technologien unterstützen Standards
  - Freie Wahl für Benutzer wird möglich (z.B. keine proprietäre nutzen)

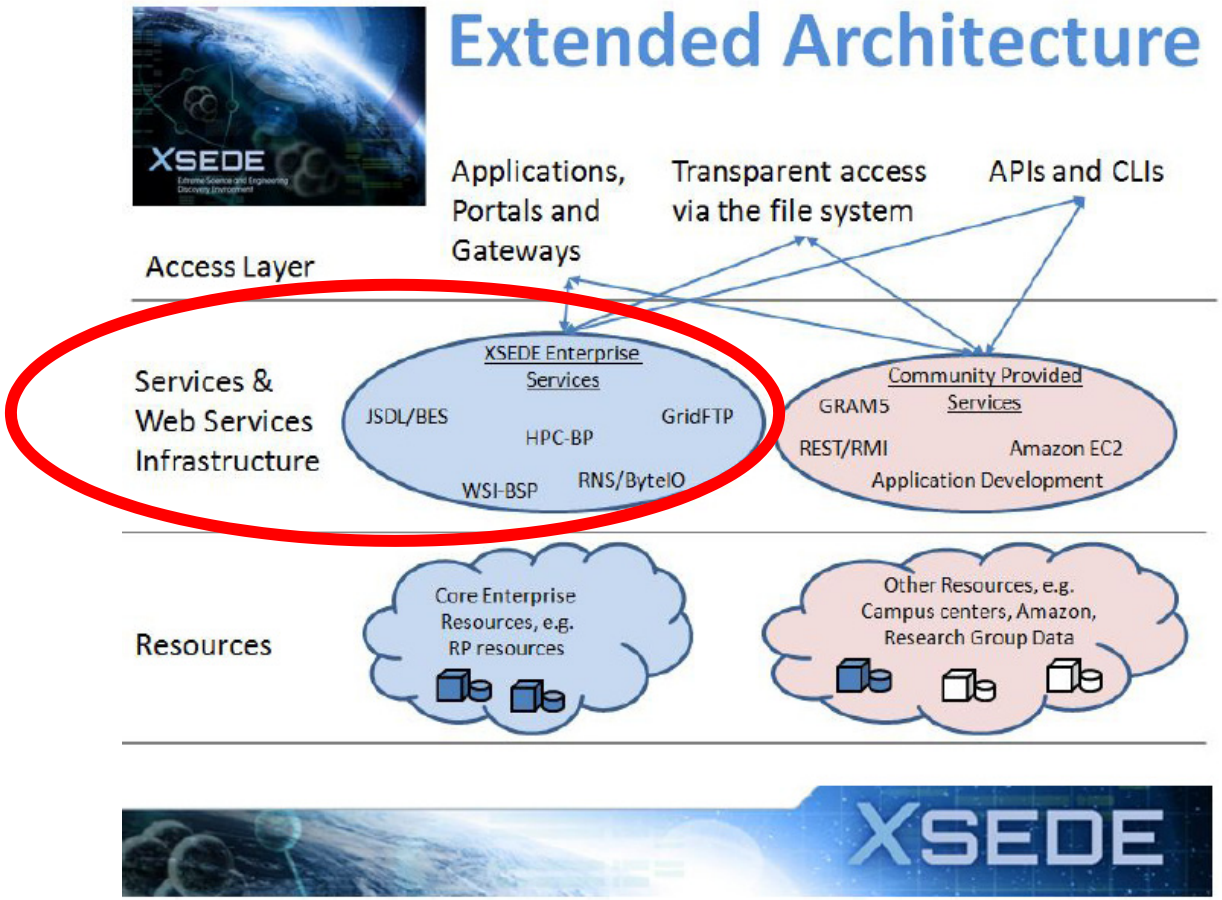


Vorteile: Grid Infrastruktur Betreiber (2)

- Hängen oft zusammen mit Software Engineering Prinzipien
- Clevere Erweiterbarkeit um neue Funktionen
  - Nutzung interoperabler Dienste anderer Infrastrukturen möglich
  - Infrastruktur muss nicht alle Dienste selbst entwickeln/anbieten
- Trennung Architektur von ihrer Implementierung
  - Offene Schnittstellen und Protokolle definieren Architektur
  - “Wandel über Zeit” durch neue Middleware beeinflusst nur begrenzt
- Lösung “nicht funktionaler” Anforderungen (engl. -ilities)
  - Bspw. Nur ein Client statt mehrere: “Usability” für Benutzer
- Weniger Entwicklungs- und Wartungs-aufwand
  - Keine komplexe Installationen von Transformationslogik (Adapter,…)
- Geringere Lernkurve für Administratoren
  - Standard-Funktionalitäten bleiben gleich bei Middleware Wechsel

Beispiel: XSEDE Architektur

- Standard-basierte Architektur
  - Implementierung: Generell Standard-basierte Middleware
  - Geplante Middleware: UNICORE, GENESIS, (Globus), andere…



Vorteile: Grid Infrastruktur Benutzer (1)

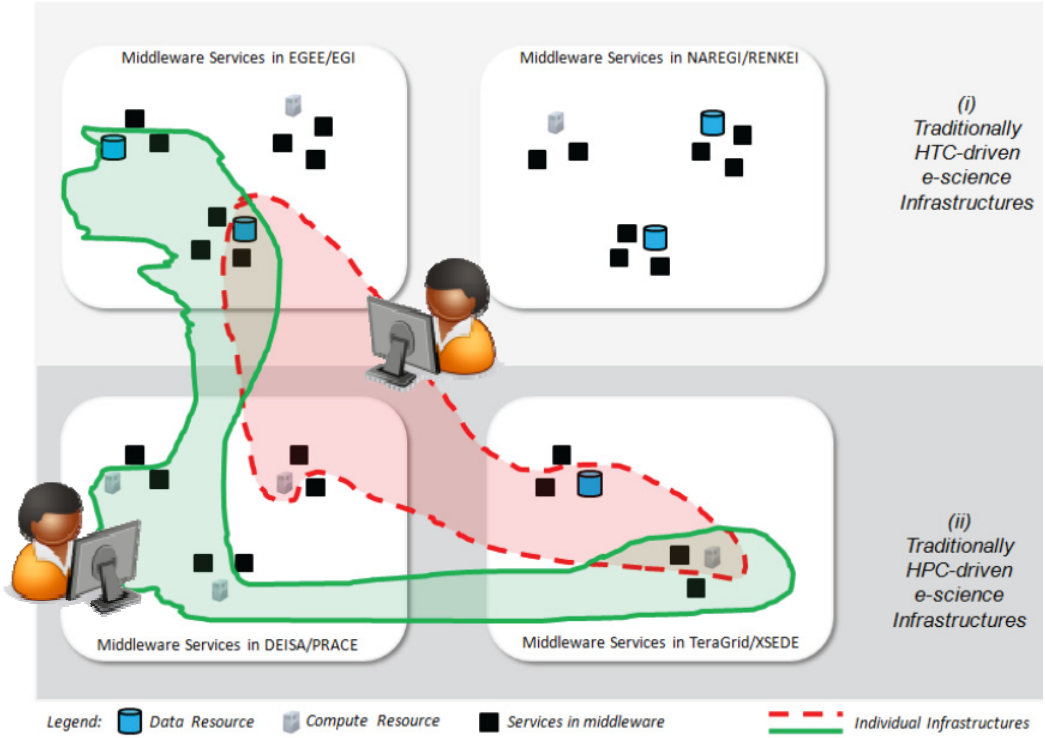
- Sichtweisen der Vorteile vielfaltig aber zusammenhängend
  - Teilweise nur andere Sicht auf die Betreiber Vorteile
  - Unterschiedliche Vorteile der Benutzer hängen auch zusammen
  - Nutzungsbedingungen der Infrastrukturen können hier beeinflussen
- “Ansatz der Benutzung” bleibt gleich
  - Standard beeinflußt oft auch indirekt auch ein Benutzungsprozess
  - Die Art wie ein Standard benutzt wird wirkt sich auf Benutzer aus
  - Die Art selbst bleibt bei anderer Middleware jedoch gleich
  - Beispiel: Job definieren und losschicken vs. abholen lassen (BOINC)
- Freiheit der Wahl
  - Benutzer können Ihre Middleware Tools wählen die sie brauchen
  - Kombination der Ressourcen nicht limitiert auf eine Grid Infrastruktur
  - Öffnung von Daten-Silos wo vorher nur proprietäre Protokolle waren

Vorteile: Grid Infrastruktur Benutzer (1)

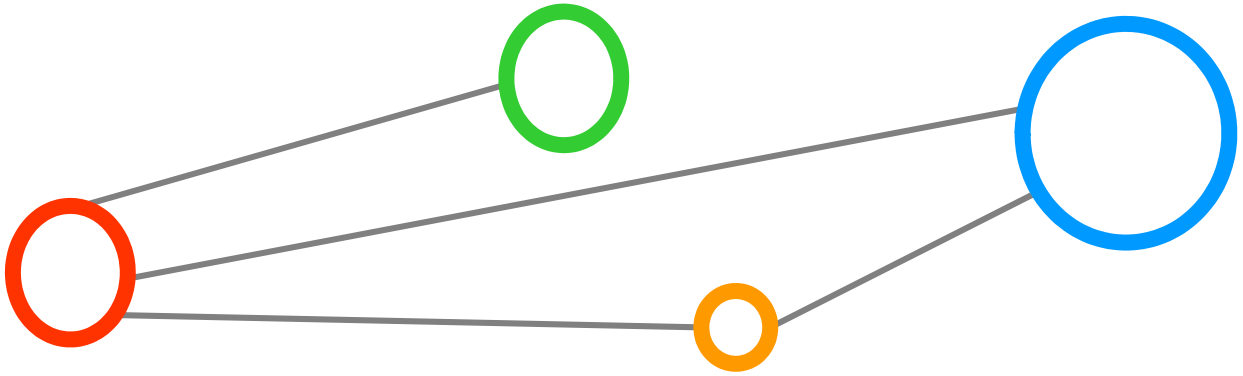
- Höhere Anzahl aggregierter Ressourcen
  - Zugriff auf mehr Ressourcn durch Interoperable Grid Infrastrukturen
- Weniger Zeit zur Lösung
  - Breiteres Spektrum von Ressourcen verfügbar
  - Schnellere/bessere Ressourcen können Problemlösung erleichtern
- Zugang zu einzigartigen Funktionalitäten
  - Einige Ressourcen sind einzigartig im Grid: Teleskope, LHC, etc.
  - Datensammlungen in bestimmten Infrastrukturen (e-Humanities)
- Sparen kostbarer Ressourcen
  - Testläufe auf kleineren Ressourcen, Produktionsläufe auf großen
- Vermeidung von Ressourcen- und Datenduplikation
  - Benutzer können Ressourcen aus anderen Grids wiederverwenden
  - Beispiel: Gewissen Daten nicht spiegeln (bspw. NASA Daten)

Beispiel: EU und US Grid Infrastrukturen

- Erstellung einer ‘individuellen Infrastruktur’ durch Benutzer
  - Benutzung der relevanten Funktionalitäten aus den Infrastrukturen
  - Grenzen der Infrastrukturen werden ‘technisch irrelevant’
  - [Aber Nutzungsbedingungen möglicherweise nicht interoperabel]



Zusammenfassung



Zusammenfassung

- Interoperabilität klingt logisch, ist aber sehr komplex
  - Vergleich: Weltweites Stromnetz ist nur teilweise interoperabel
  - Bedeutung: Middleware/Infrastrukturen funktionieren zusammen
  - Probleme: Inkompatibilität von Schnittstellen, Protokolle, Sicherheit,..
  - Kurzfristige Lösungen helfen bringen aber mehr und mehr Aufwand
- Standards unterstützen Interoperability
  - Verschieden technische Bereiche muss man zusammen betrachten
  - Standards allein machen nicht Interoperability von Middleware aus
  - Gleiche Kombination mehrerer Standards führt zu Interoperability
  - Bei hoher Komplexität helfen Referenz-modelle und -architekturen
- Interoperability bietet Vorteile
  - Unterschiedliche Sichtweisen und Nutzungsbedingungen existieren
  - Betreiber: Einsparung Kosten, Risikominimierung Middleware, etc.
  - Benutzer: Erstellung “individueller Infrastrukturen” je nach Bedarf