Grid and The Enterprise: Service Oriented Architecture

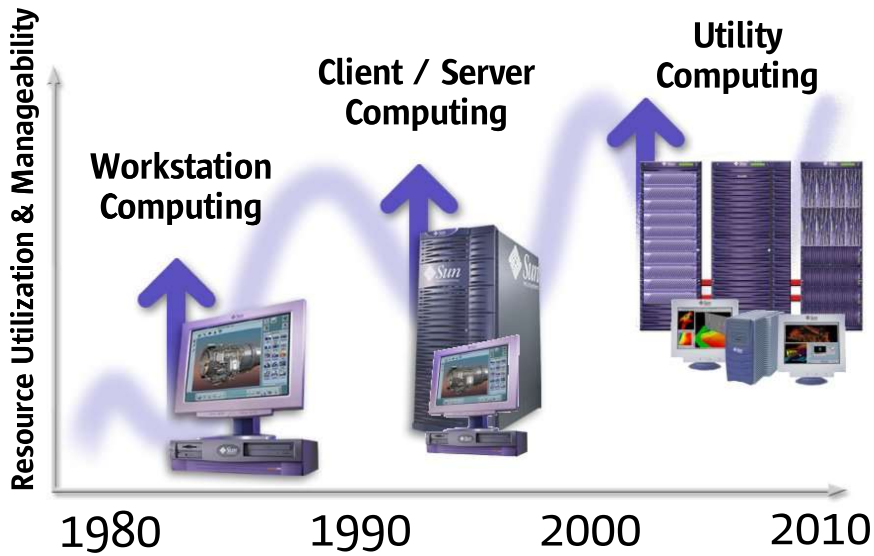
Peter ffoulkes
Director of Marketing
High Performance & Technical Computing

What is Grid Computing?

"A SERVICE FOR SHARING COMPUTER POWER AND DATA STORAGE CAPACITY OVER THE INTERNET."

CERN, GRID CAFÉ, "WHAT IS A GRID,"
HTTP://GRIDCAFE.WEB.CERN.CH/GRIDCAFE/WHATISGRID/WHATIS.HTML.

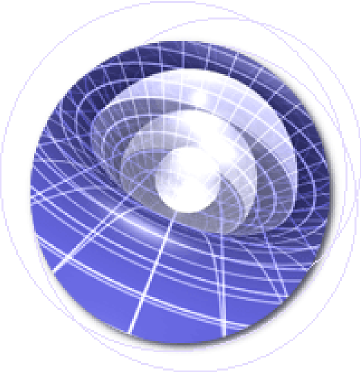
Grid is a milestone on the journey



“Grid computing makes Sun's famous pronouncement, The Network is the Computer, an even more workable proposition”

Clay Shirky, 2003

What is Grid Computing?

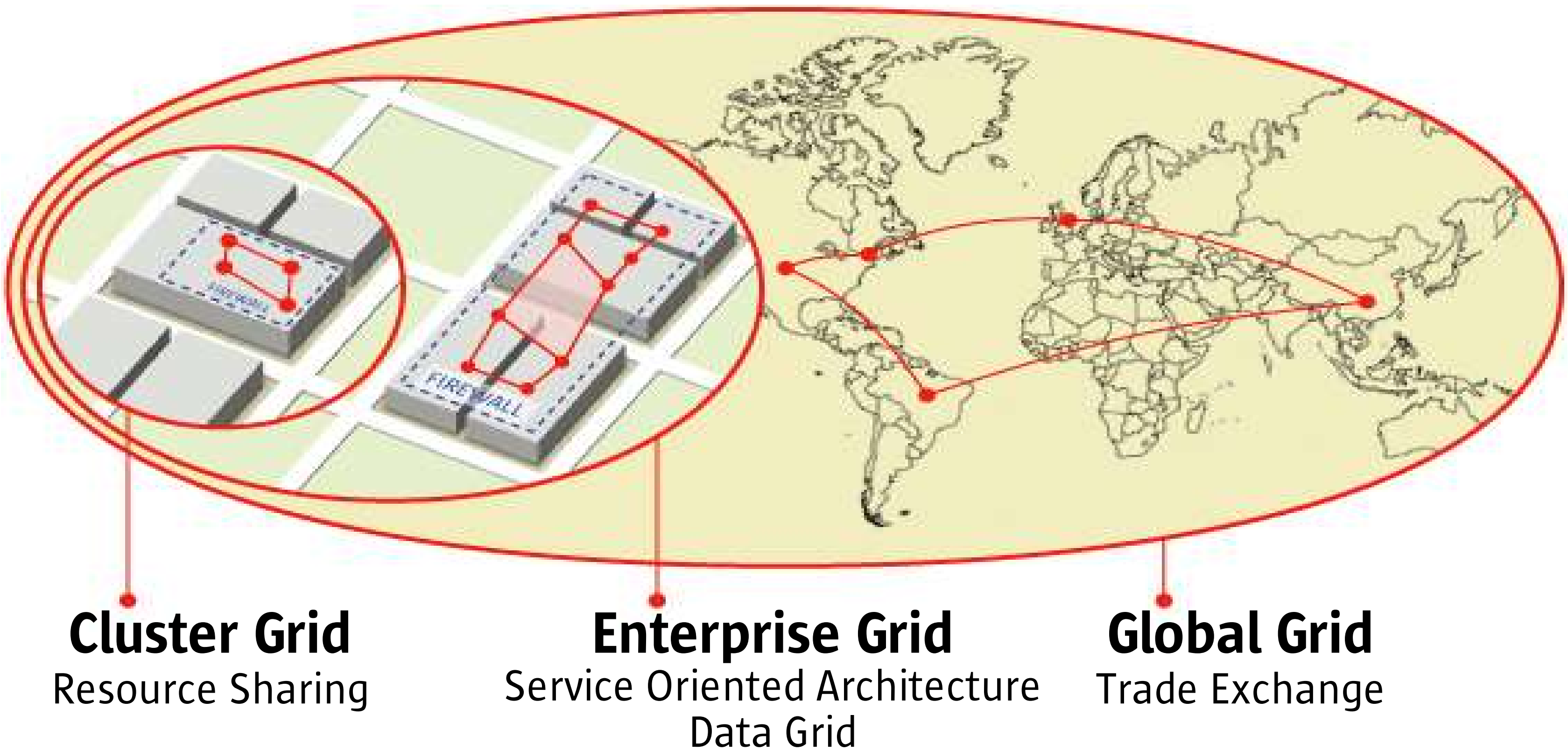


Grid computing is a coordinated way of dynamically managing and sharing disparate sets of compute resources

Grid computing provides:

- A natural evolution of distributed computing
- Horizontal scaling par excellence
- A "Return on Asset" based cost model

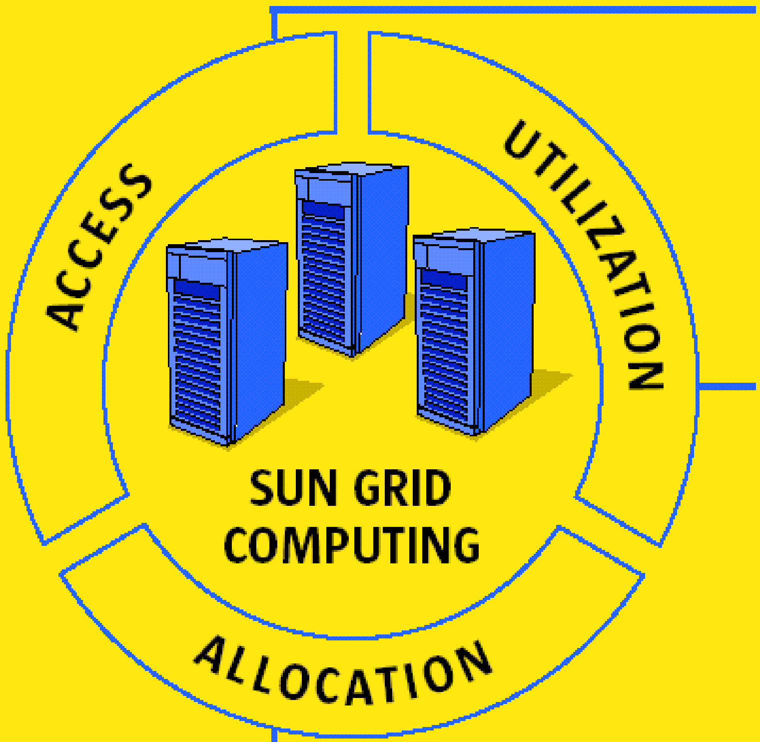
Virtual Consolidated Data Center



Virtualization of Resources

Single point of access to...	massive computing power as a network service – dynamically available where and when needed most.
Transparent scalability of...	CPU cycles, storage, databases and devices across all IT resources.
Access that is...	Service oriented, easy, dependable, consistent, pervasive, inexpensive and secure.

Grid – Rethinking Computing



ACCESS: Makes it easy to access shared resources, so users have the compute power they need to get the job done.

UTILIZATION: All tasks are ideally matched to resources for up to 100% utilization, 24 x 7.

ALLOCATION: Sophisticated controls allocate resources dynamically to meet changing business priorities and deliver high quality of service to user groups.

ACCOUNTING:
Service Level Agreements and Charge Back mechanisms for contractual computing

GRID is an Operational Concept

An information service workflow mapping and virtualization

Data | Transformation | Representation | Interpretation

Capability & Capacity Computing Services

Data Storage, Management and Access Services

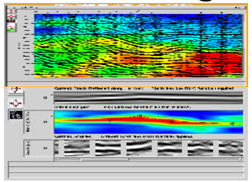
Collaboration, Graphics, and Visualization Services

Upstream Oil and Gas Workflow

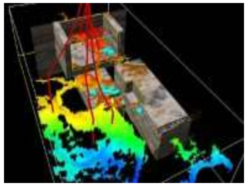
Data Acquisition

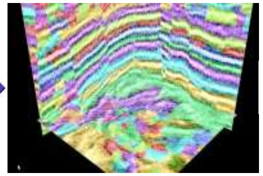
Data Management

Seismic Processing

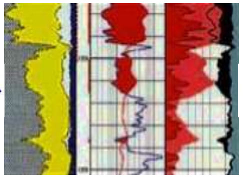


Visual Interpretation

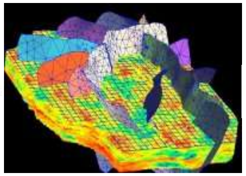




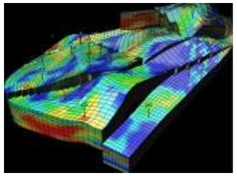
Modeling Automation



Petrophysical Analysis



Property Modeling



Simulation

Grid Computing

- Mapping the information supply chain defines the solution requirements
- Each stage of the information supply chain has unique requirements
- Each information supply service must interact seamlessly with the others
- Information supply service solutions must offer best of breed components
- The system must deliver a contracted level of business service and cost

Workflow Courtesy of Landmark – A Halliburton Company

Oregon State University:
Managing the End-to-End Information Flow

Petabytes

Multi-platform, multi-parameter, high spatial and temporal resolution, remote & in-situ sensing

Autonomous, In-space Calibration and Data Reduction

Terabytes

Interaction Between Modeling/Forecasting and Observation Systems

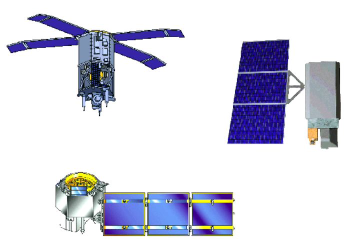
Gigabytes

Interactive Dissemination

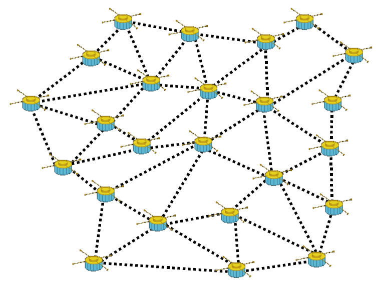
Predictions

Megabytes

Advanced Sensors

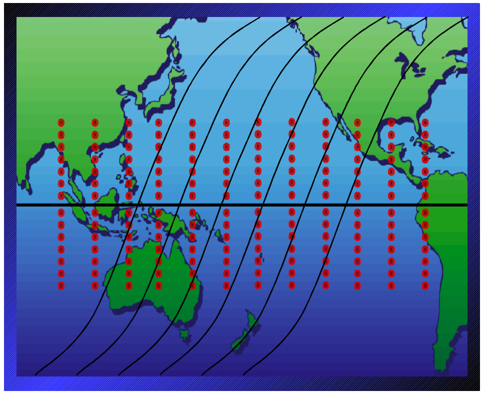


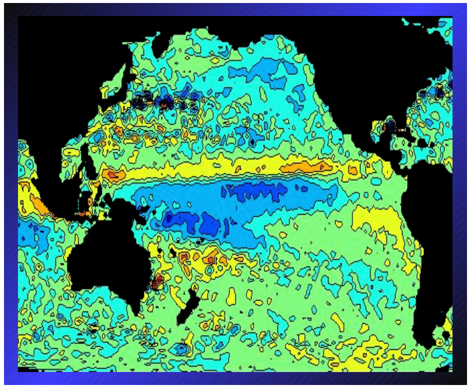
Sensor Webs

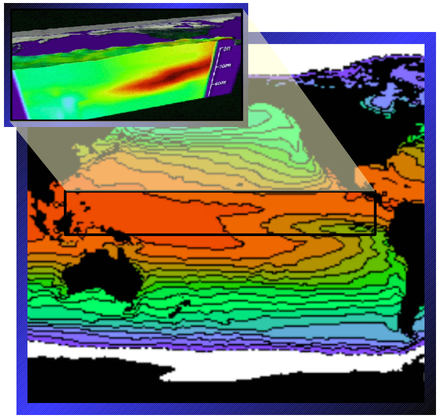


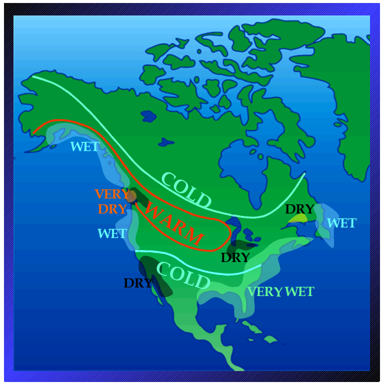
Information Synthesis

Access to Knowledge

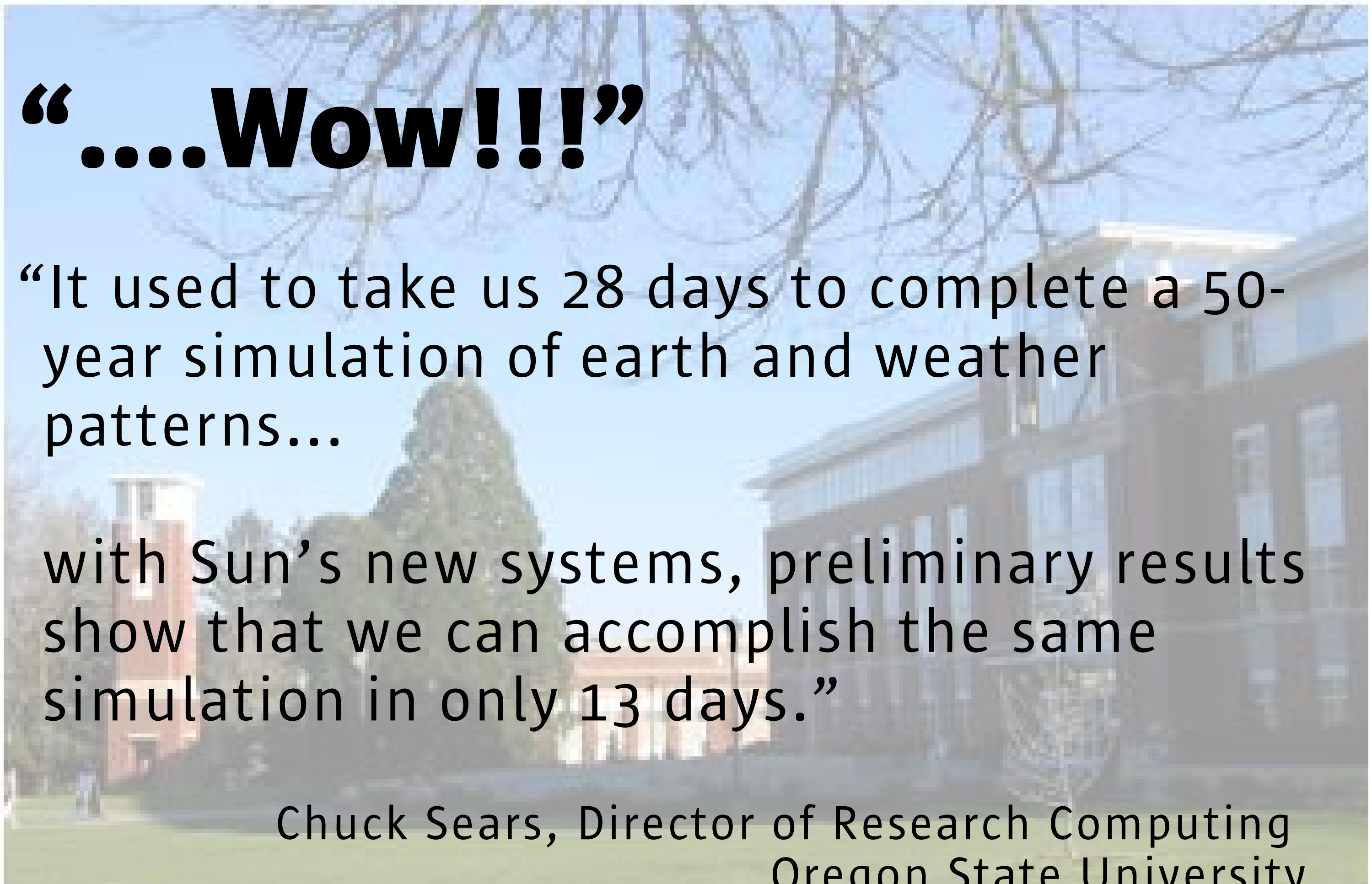








Grid is about service optimization



"....Wow!!!"

"It used to take us 28 days to complete a 50-year simulation of earth and weather patterns...

with Sun's new systems, preliminary results show that we can accomplish the same simulation in only 13 days."

Chuck Sears, Director of Research Computing
Oregon State University

Oregon State University:
“Grid is about service optimization”

- Thousands of sensors and platforms
- Continuous data streams
- Live data feeds
- Autonomous sensors
- Multi-disciplinary studies
- Integration with models

Oregon State University: "Grid brings new approaches to IT"

- Think about problems in new ways
- No more data silos, processing pipelines
- New ways of thinking about workflows
- No more point solutions
- Information models about data, tools, and the relationships between data objects
- Next steps are determining how to classify, seek new relationships, and reconstitute processes into distributed services
- Problems cannot be broken easily into hardware/software components

Requirements

- Virtualization
  - Treat distributed resources as a pool
- Automation
  - Reduce and simplify manual tasks

Dynamically match data flow and service workloads to resources in accordance with policy

Enterprise Computing:
Evolution of a Dynamic Network Architecture

Business Model

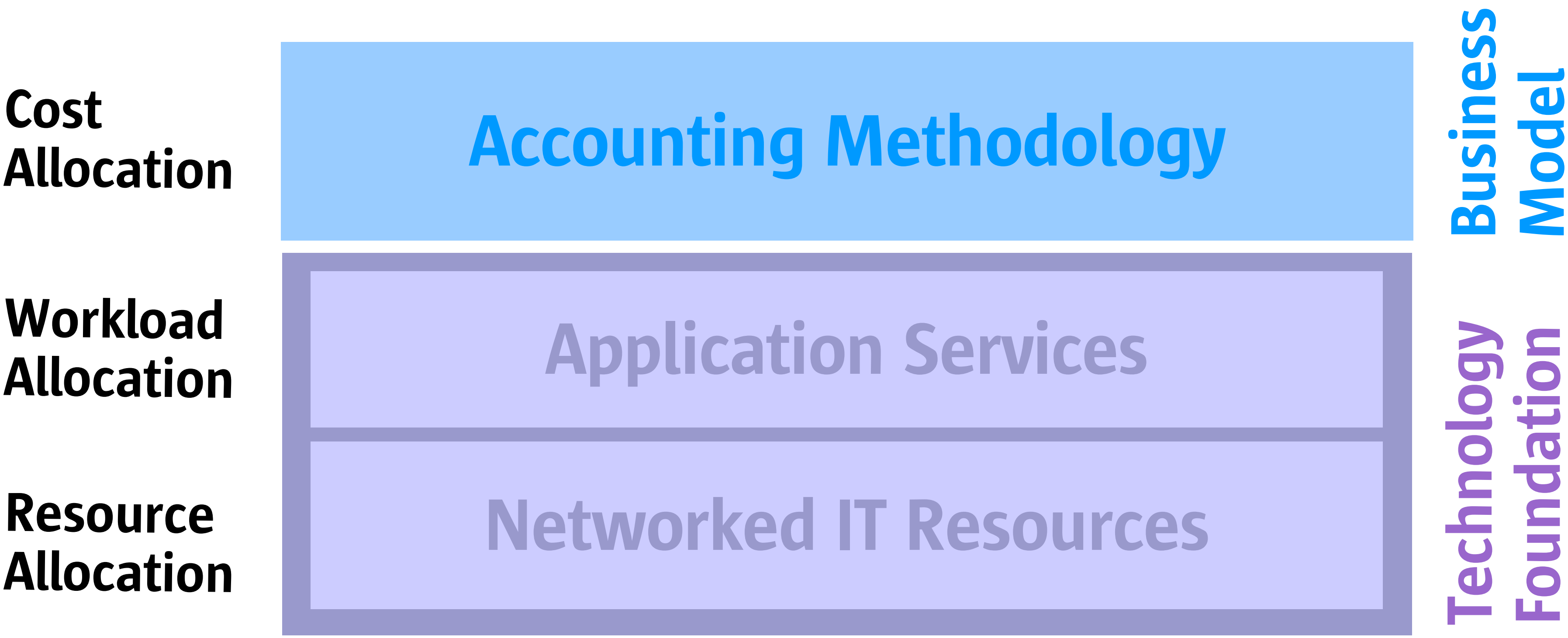
Accounting Methodology

Technology Foundation

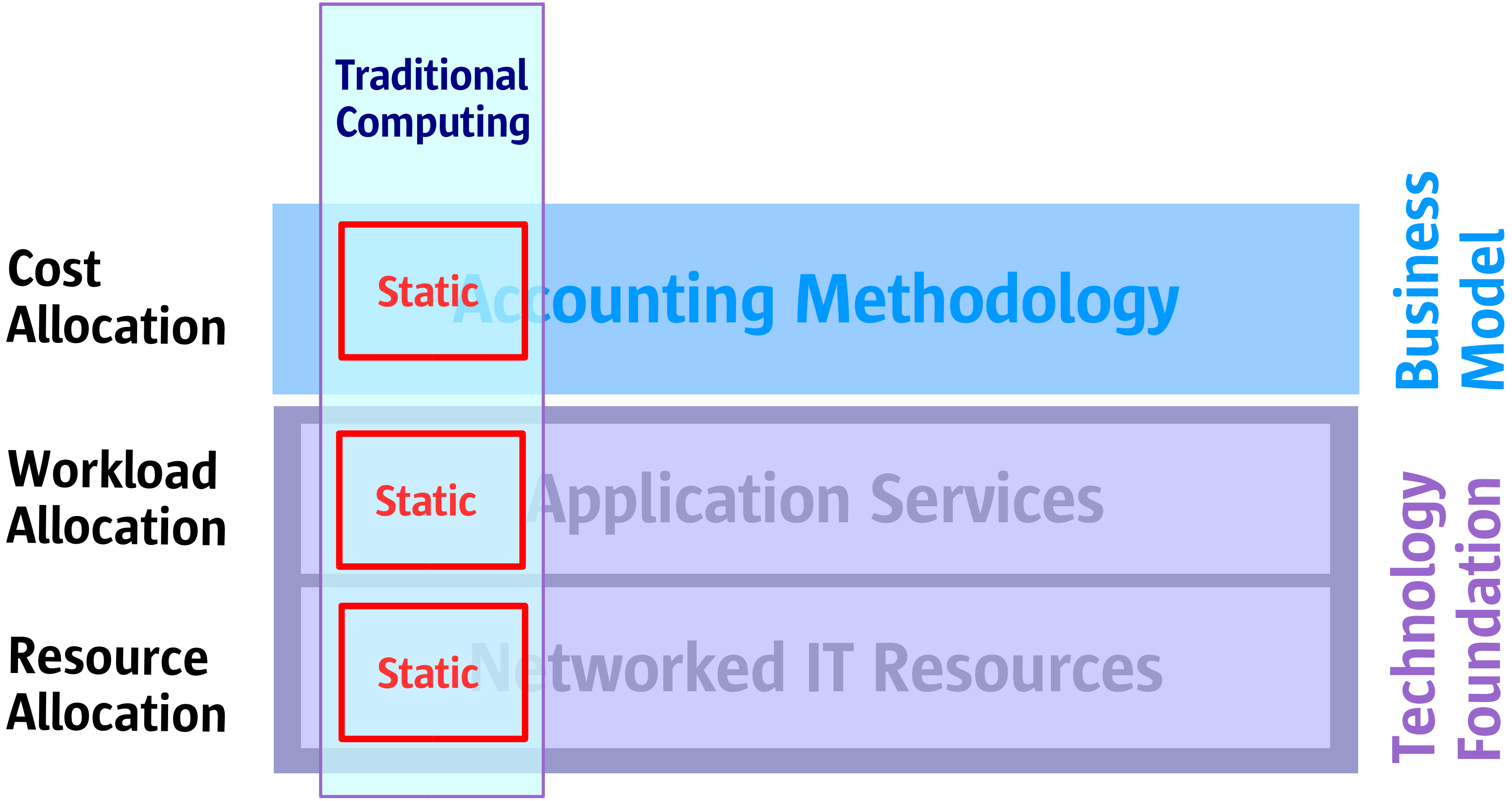
Application Services

Networked IT Resources

Enterprise Computing:
Evolution of a Dynamic Network Architecture



Enterprise Computing:
Evolution of a Dynamic Network Architecture

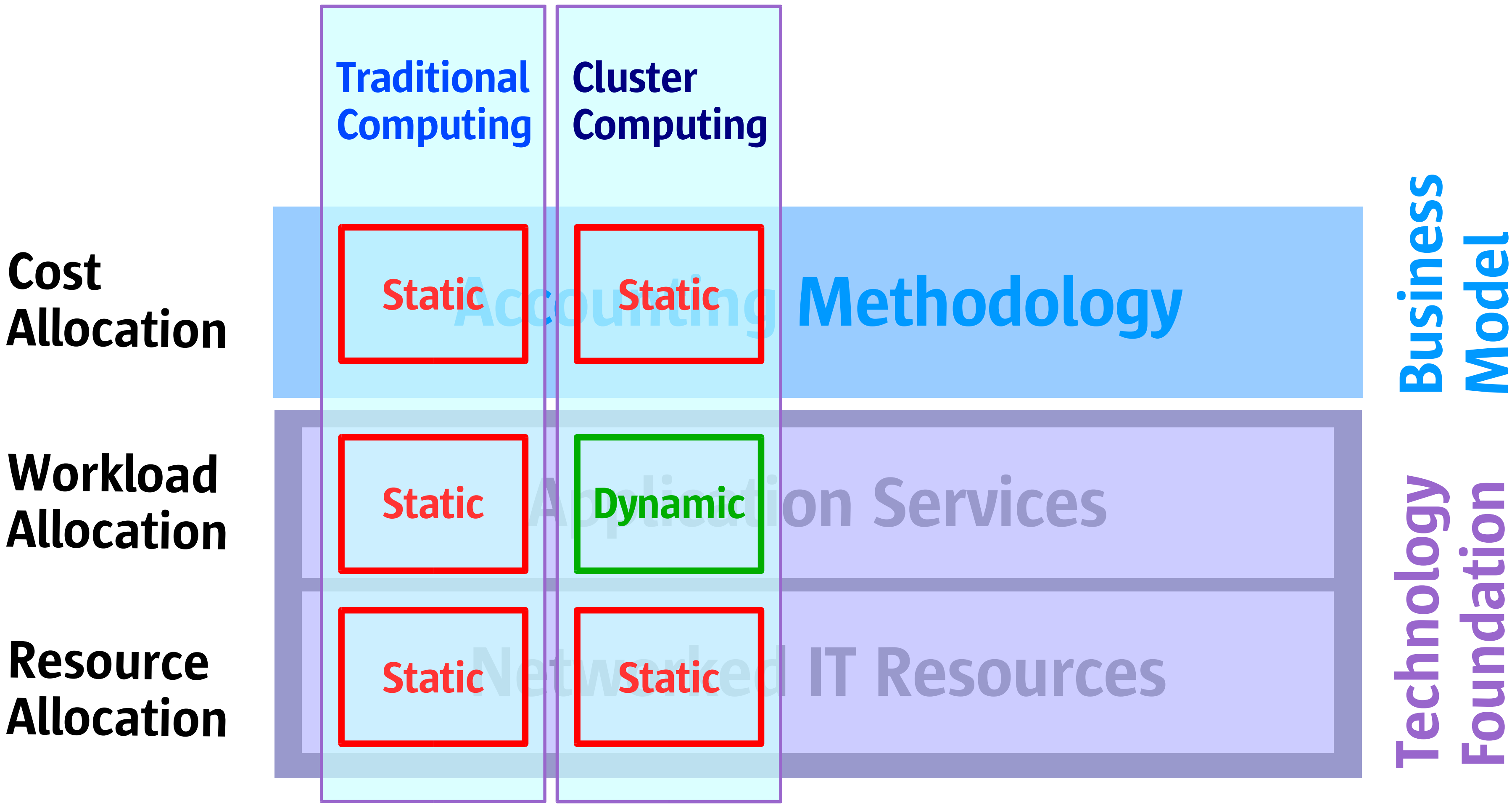


Sun microsystems

Enterprise Computing:
Evolution of a Dynamic Network Architecture

	Traditional Computing	Implementation	
Cost Allocation	Static	Fixed cost structure: Bought or leased resources In-house IT staff	Business Model
Workload Allocation	Static	Application services manually deployed on specific IT assets	Technology Foundation
Resource Allocation	Static	Fixed pool of IT assets managed and provisioned individually	

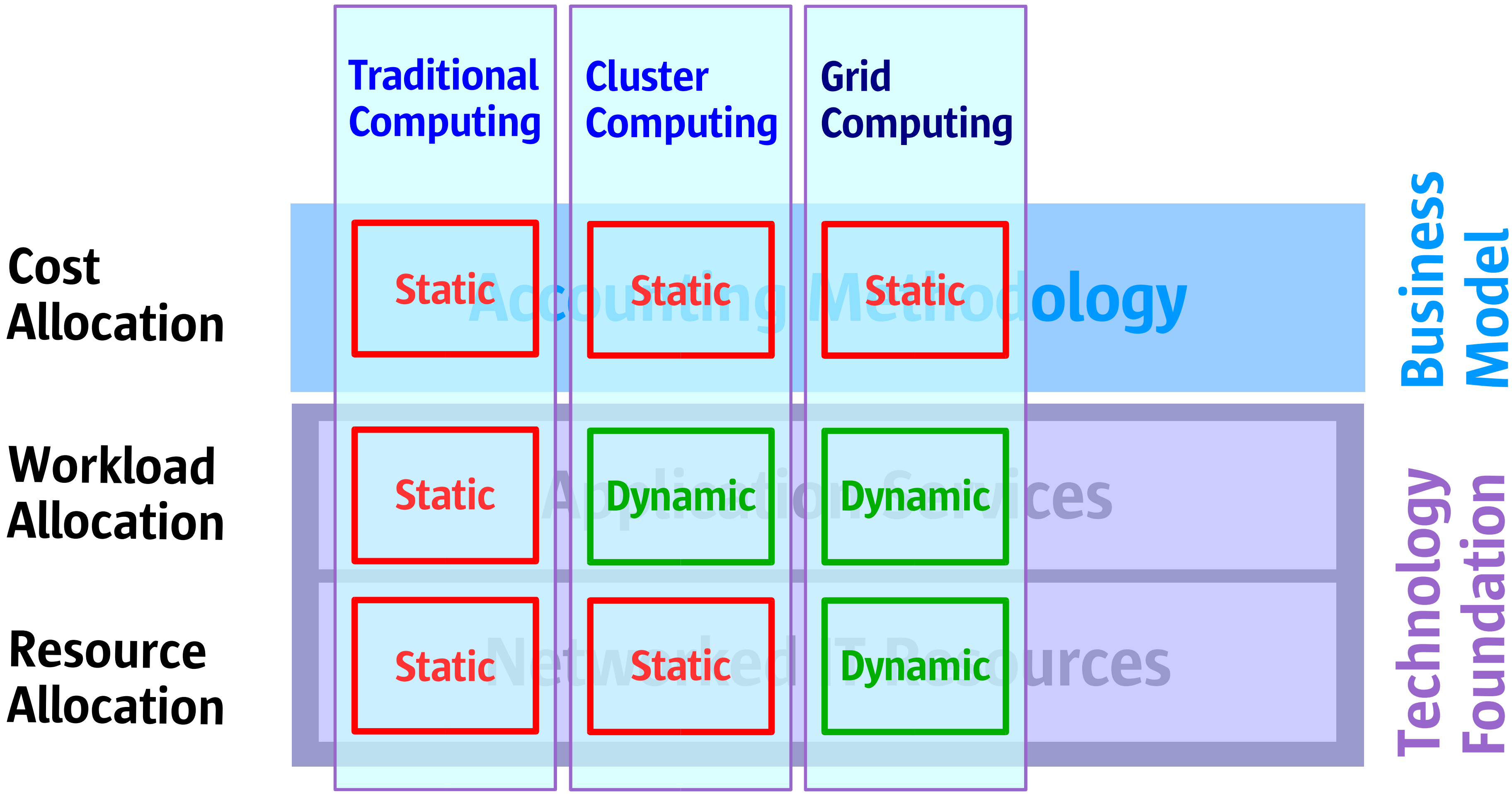
Enterprise Computing:
Evolution of a Dynamic Network Architecture



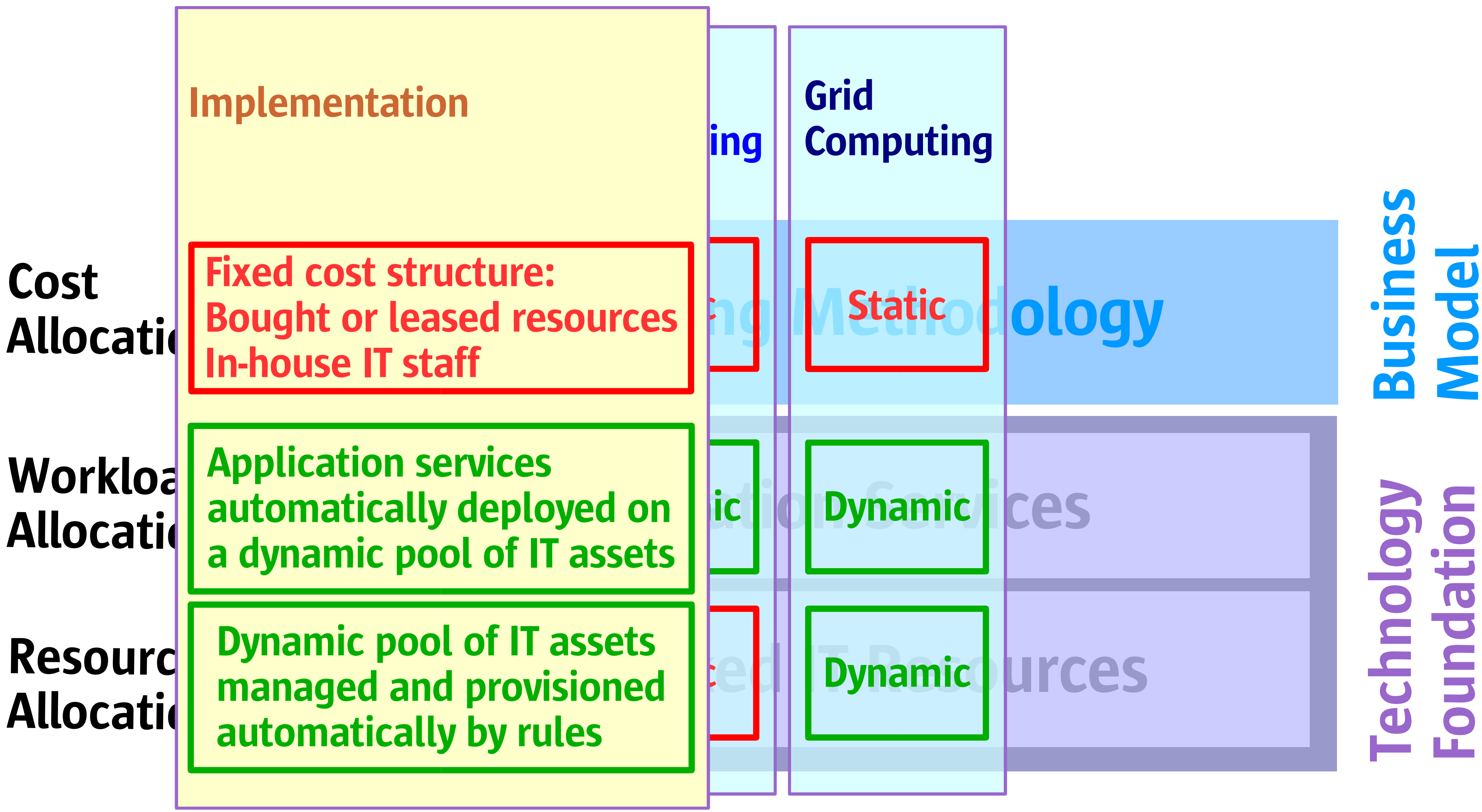
Enterprise Computing:
Evolution of a Dynamic Network Architecture

	Traditional Computing	Cluster Computing	Implementation
Cost Allocation	Static	Static	Fixed cost structure: Bought or leased resources In-house IT staff
Workload Allocation	Static	Dynamic	Application services automatically deployed on a fixed pool of IT assets
Resource Allocation	Static	Static	Fixed pool of IT assets managed and provisioned individually

Enterprise Computing:
Evolution of a Dynamic Network Architecture



Enterprise Computing:
Evolution of a Dynamic Network Architecture



Enterprise Computing:
Evolution of a Dynamic Network Architecture

	Traditional Computing	Cluster Computing	Grid Computing	Utility Computing
Cost Allocation	Static	Static	Static	Dynamic
Workload Allocation	Static	Dynamic	Dynamic	Dynamic
Resource Allocation	Static	Static	Dynamic	Dynamic

Accounting Methodology

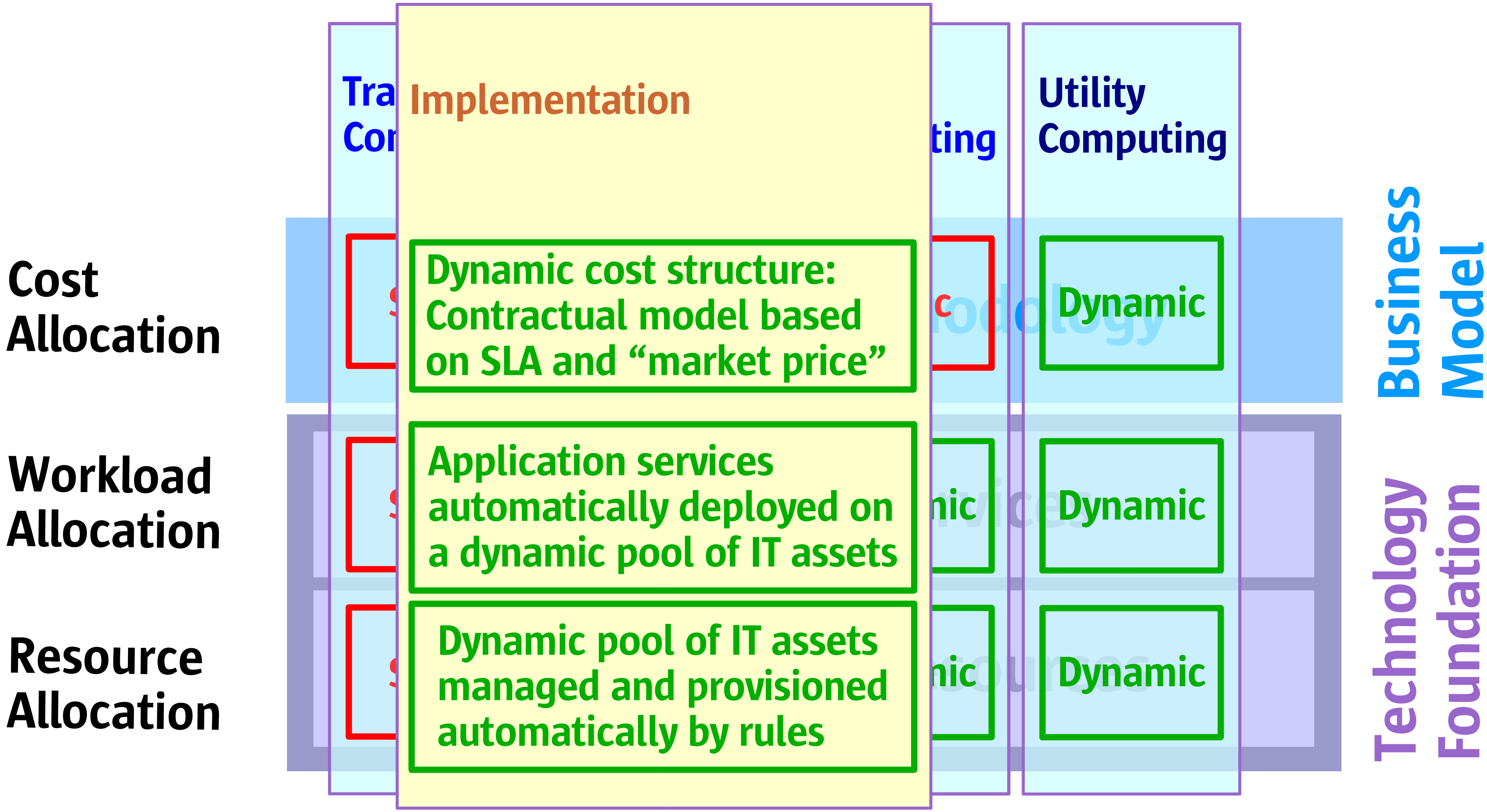
Application Services

Networked IT Resources

Business Model

Technology Foundation

Enterprise Computing:
Evolution of a Dynamic Network Architecture



Enterprise Computing:
Dynamic Network Architecture

Traditional Computing | Cluster Computing | Grid Computing | Utility Computing

Cost Allocation — Dynamic Accounting Model — Business Model

Workload Allocation — Dynamic Application Services — Technology Foundation

Resource Allocation — Dynamic Network Resources

What Is Being Said About Grid:

"You will be able to rent your compute needs across the Internet..."

"It's utility computing..."

"All you need is enough small boxes..."

"EGA, Global Grid Forum, OGSA, and Globus are the only way to go..."

"Grid computing is ready for mainstream in your datacenter..."

How Real Is This Stuff?

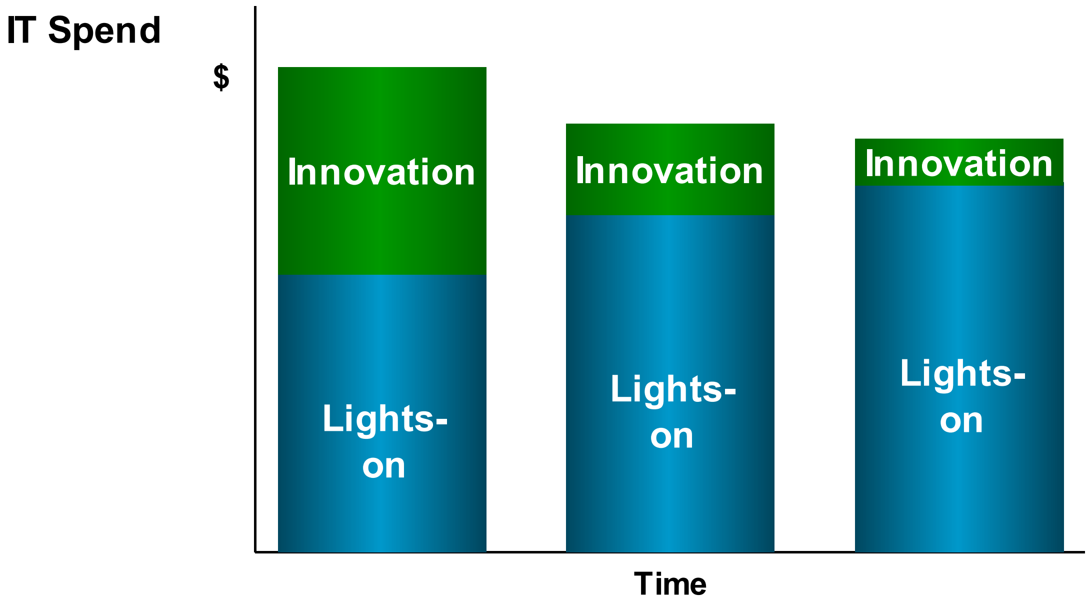
What can you take to the bank?

WHERE SOA MEETS GRID

A Perspective from a VP of IT Architecture
at a well known global bank

Problem - The Innovation Squeeze

The combination of operational complexity and lowered budgets has resulted in the reduced budget for innovation



Innovation is being squeezed out due to decreased budget and increased operational costs

A lack of automation has caused an increased the focus on lights-on cost

- This has crippled IT's ability to innovate for the business
- IT must re-invent itself with a new computing model

Perspective from a Global Banking Institution

Solution – Service oriented computing

Starts with business requirements – these are expressed as SLAs

- IT becomes accountable for what it delivers
- Our customers understand what they are paying for, and can choose a level of service that suits their needs (and wallets)

Business service delivery (realising the SLA) is accelerated by composition of underlying IT services within an SOA

- Reuse (and where necessary modify) what has been built before
- Buy in commodity services (and sell services to those lacking the resources to build their own)

SOA is facilitated by service orientation in the infrastructure

- Hardware provisioned independently of the applications that it services
- Improves deployment efficiency, easier to provide scalable and resilient deployments

Increased Agility – Coarse grained services

SOA is a means not an end. The promise is to deliver competitive advantage through increased agility.

- An SOA built on Web Services delivers agility by:
  - Simple integration and aggregation for client apps (e.g. Portals)
  - Easier update/maintenance/change pathways
  - A Shorter path between modelling business processes and implementation

- Agility is facilitation by composition
  - The composition lifecycle is evolving:
    1. Manual composition (glue code)
    2. Orchestration (BPEL is the score for engines to act as conductors)
    3. Choreography (follow the steps laid down in WS-CDL)

- Politics have disrupted integration in the past, and hindered agility
  - Web Services dodges the platform / language wars
    - Beware! This isn't always a good thing.

To be a real service

Services must have Service Level Agreements (SLAs)

- A web service can be created by running Tomcat on a workstation, and the public UDDIs are full of localhost:8080 entries that nobody can use.
  - This isn't the sort of service that we want in an SOA
- The alternative is a best effort oriented architecture – do we want this?

All of our businesses have boilerplate for our contracts. We need the same for our SLAs

- This starts with the artefacts of the environment – e.g. XSDs, WSDLs
  - Static Governance policies – what happens in the tools
- WS-SLA - the missing standard?
  - We need something that can be read and understood by business decision makers, lawyers and systems.

SLAs need to be monitored and managed

- Look at what is passing through the SOA, measure it, take corrective action when limits are exceeded
  - Dynamic Governance policies – what happens on the wire

Terminology - services

Coarse grained services – are about messages

- A service exposing a public interface that responds to XML messages
- Typically SOAP messages, but could also be REST
- Can be described using WSDL
- Can be found in an SOA registry (the thing that fits the hole that UDDI doesn’t)
- Transport will be HTTP or message oriented middleware (MOM)

Fine grained services – are about behaviour

- The building blocks of coarse grained services
- Have language/platform dependent interfaces and transports
- Are usually described by a public interface to an object
- Can be found in an implementation specific registry (e.g. JNDI, JINI registry, Windows registry, Active Directory or via an LDAP lookup).

Terminology - grid

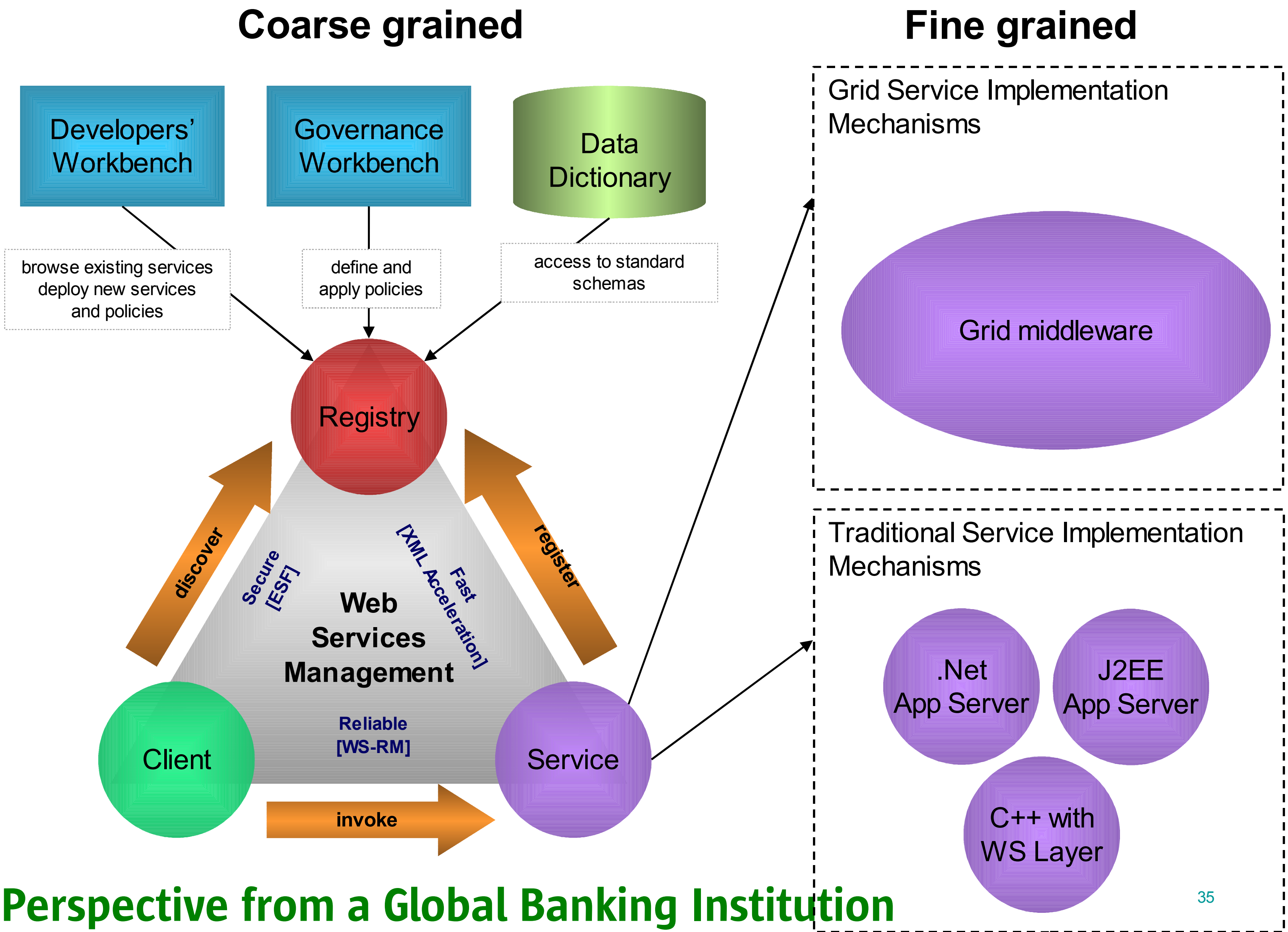
A grid is a system composed of an arbitrary number of machines (nodes), running software that virtualises access to the underlying resources.

- Grids require minimal effort to add new resources and show minimal impact when an individual unit of resource is removed (contrast with clusters where adding a new node can be labour intensive, and removing a node may take away up to 50% of capacity or resiliency)

There are various categories of software associated with grid

- **Compute** – used for processing of CPU or volume intensive jobs
- **Data** – used to distribute data so that it is in the right place at the right time
- **Utility** – used for management of the data centre (e.g. provisioning)

Architecture for SOA With Governance (Q1-05)



Perspective from a Global Banking Institution

Hosting web services on a grid - benefits (and drawbacks)

Better

- ✓ Cross platform / language (in some cases)
- ? Simple APIs - reduced developer expertise required
- ✘ Data distribution can quickly become a real headache

Faster

- ✓ Scales quickly
- ✓ Large numbers of machines can be brought to bear (provided that a problem can be broken up)
- ✘ May be some overhead introduced by additional management layers and network hops

Cheaper

- ✓ Better hardware utilisation – this isn't just about the CAPEX for kit
- ✘ Licensing for grid software
- ✘ Possible vendor lock in (until standards shake out)

Challenges ahead

XML Performance

- ✘ XML parsing introduces processor and memory overheads
- ✘ XML is verbose, leading to greater network utilisation and latency

Description and discovery

- ? Where to find services

The S word

- All secure roads lead back to X509
  - Auth - SAML hides this rather than fixing it
  - SOA artifacts might need to be signed so that they can be trusted
- ✘ Working with certificates is hard

Starting in the right place

- ? Which comes first – SLA, WSDL or implementation
- Need to have round tripping between these things

XML Performance

Avoid chained parsing operations (e.g. security policy, service policy, application code).

- Parse once (to a common binary token format) and then pass the parsed object through the pipeline
- Needs more work in the standards sphere

Use co-processors for CPU/memory intensive operations

- Operations on large documents create huge amounts of temporary objects, and these in turn lead to issues with garbage collection / memory management.
- This avoids sizing an application for the largest document that passes through it (e.g. quarterly or end of year reports may be much larger than daily/weekly)
- This functionality can also be expected to become part of the network / fabric

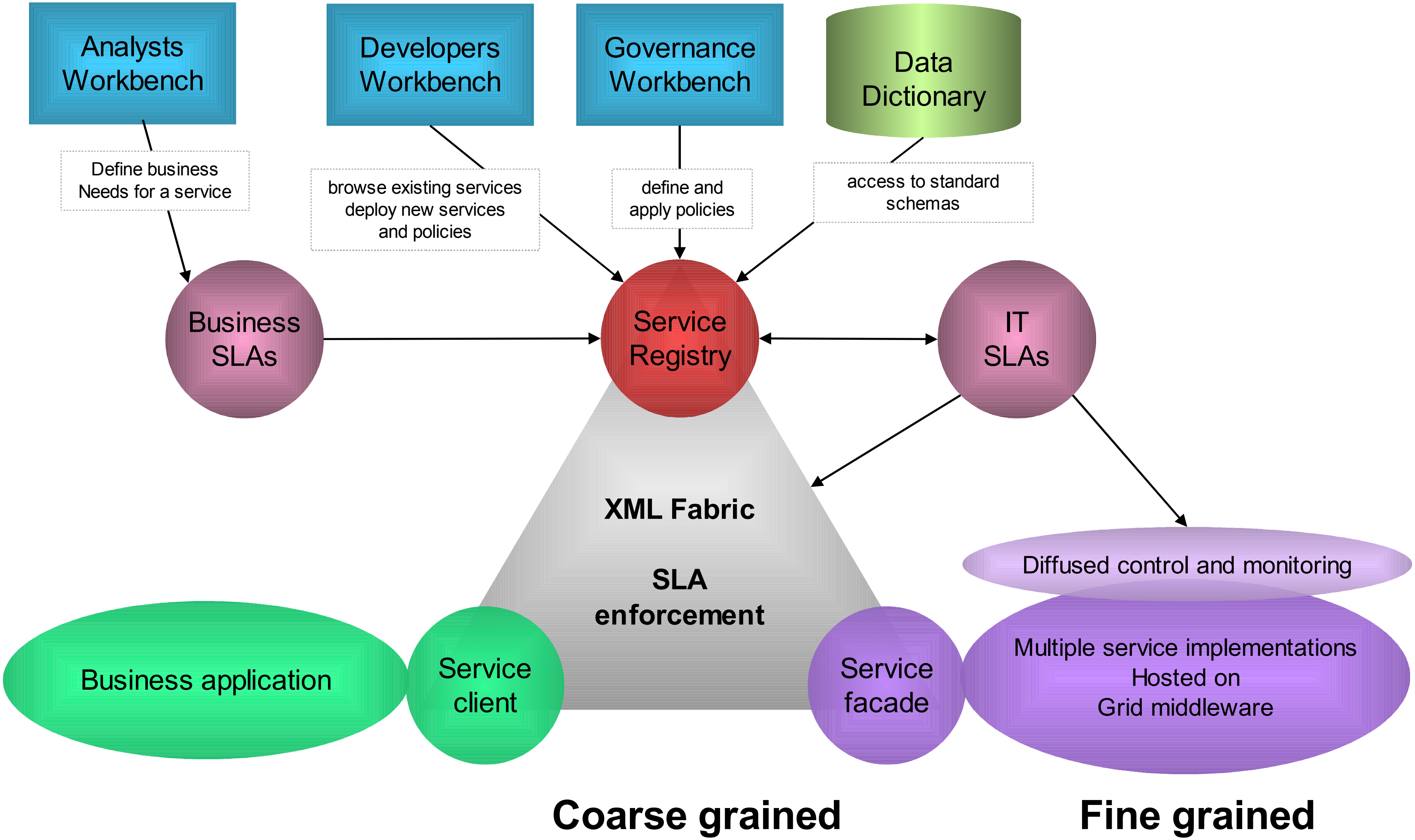
Description and discovery

Learn the lessons from the past. Why did previous SOA attempts (using DCE, CORBA, DCOM, RMI and J2EE) not give us everything we wanted?

- Need to avoid point to point linkages, as these grow unmanageable
  - This can be fixed using a proxy (e.g. within a Web Services Management package)
  - Such functionality can drop down into the network / fabric

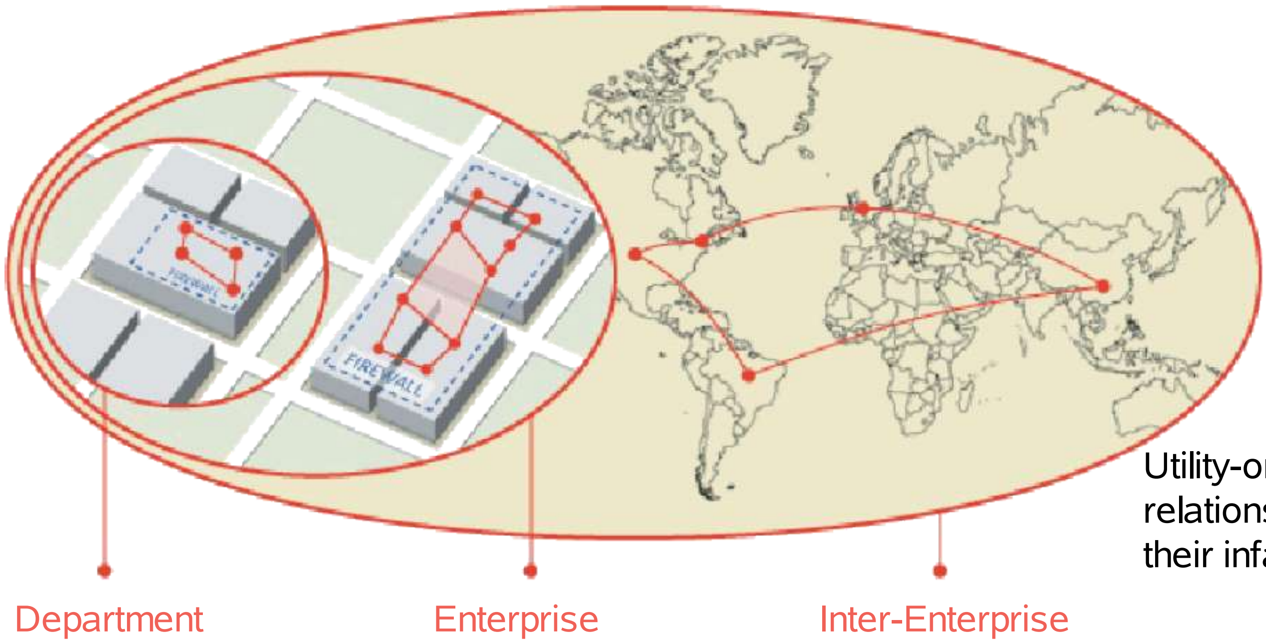
- Need to be able to find services
  - Registries are important
    - UDDI doesn’t really cut it for all of the registry use cases, so something better is required
  - Must have a meaningful language for describing services
    - Semantic integration (using language constructed from what you have) is probably a better place to start than creating vocabularies from scratch (as happens with tModels)

The end state?



Perspective from a Global Banking Institution

Enterprise Grid Reality

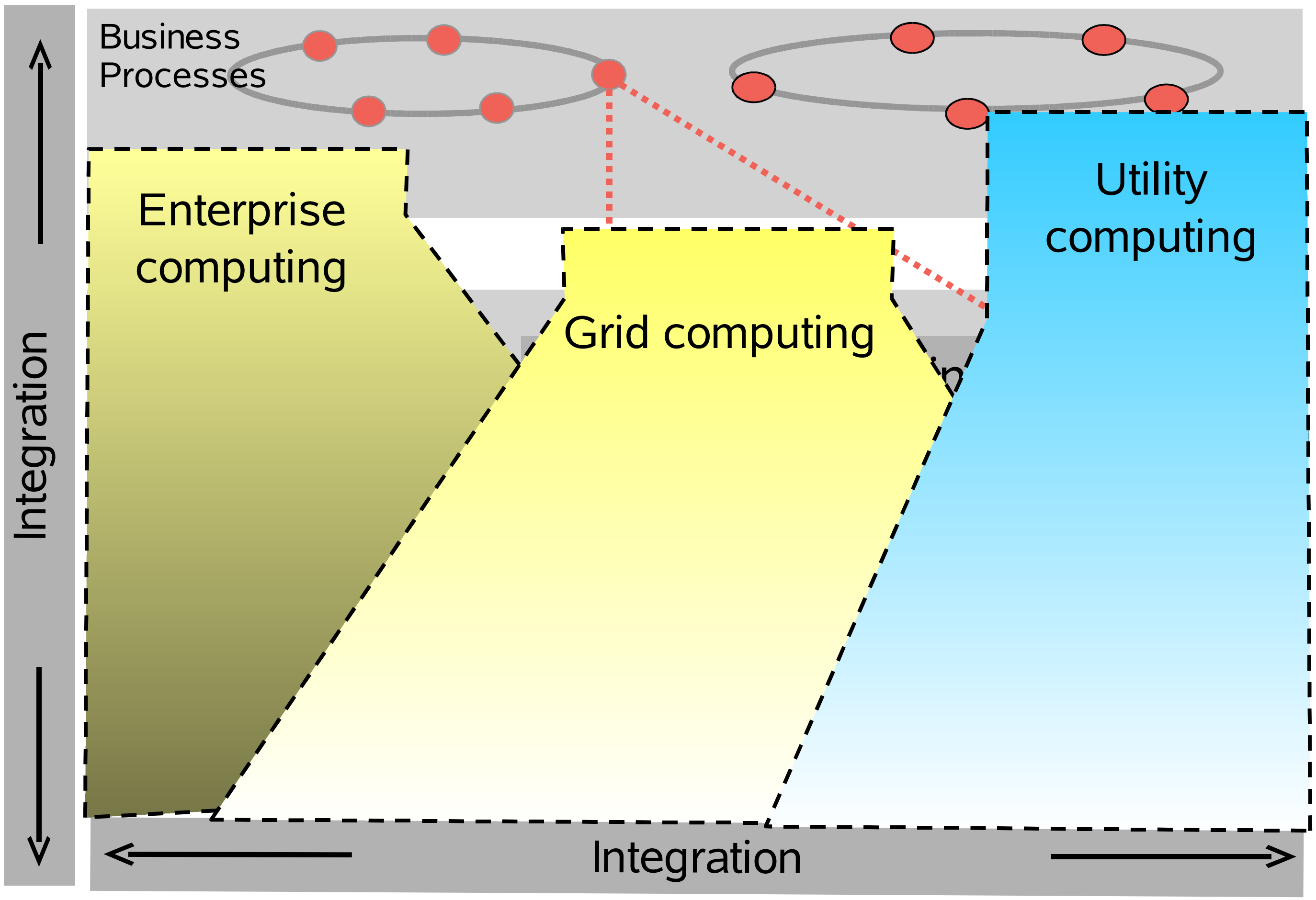


Departments are effectively utilizing compute clusters today

Utility-oriented IT relationships are still in their infancy

Leading CIO's are moving to a "shared infrastructure, shared services model" based on SOA concepts and virtualization technologies

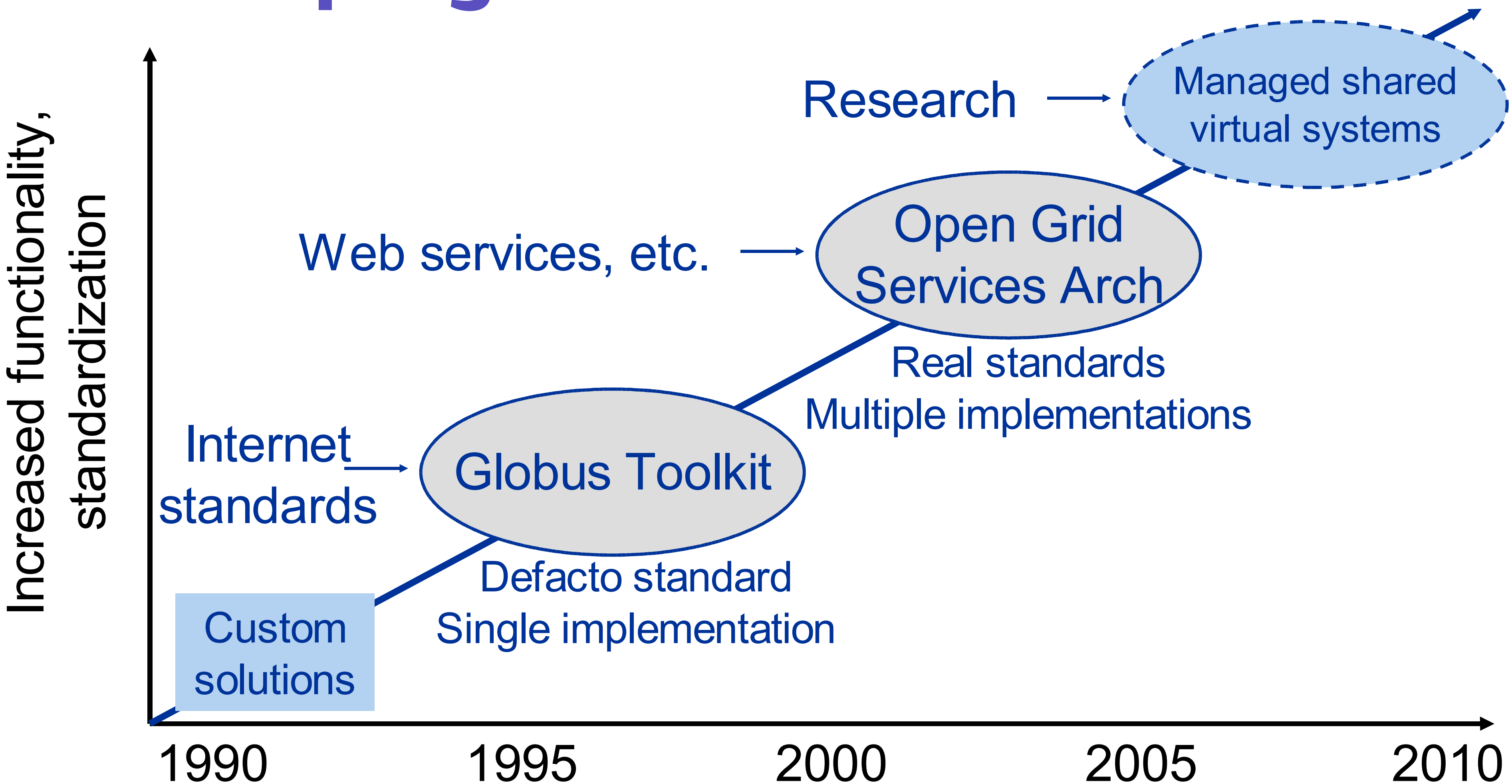
Enterprise Grid Dilemma



Common and compelling themes

- Underlying IT architecture instrumented to business processes
- Applications expressed as services in support of business processes
- SOA for virtualization, integration and management automation
- Shared resources for efficient and dynamic utilization

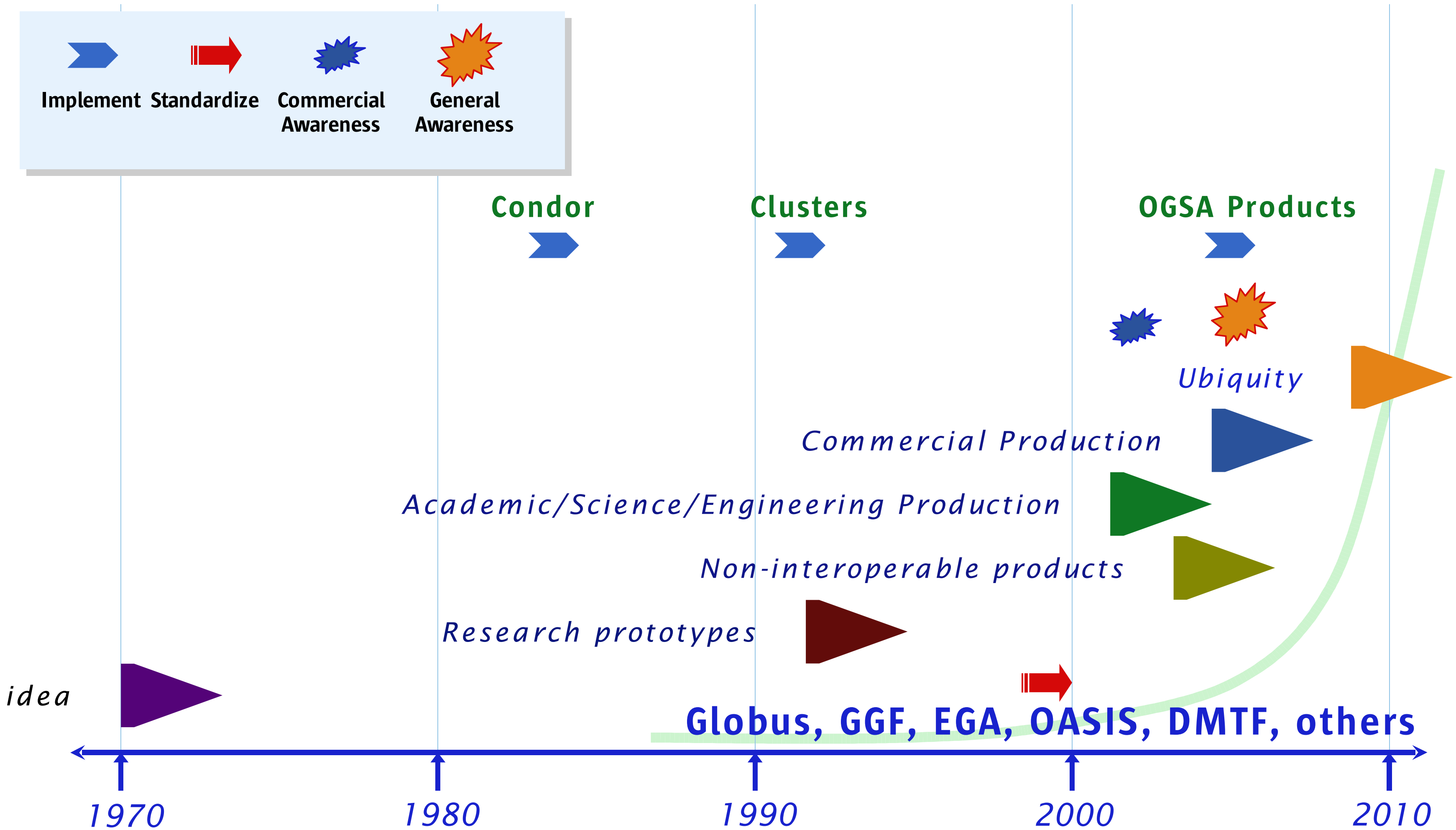
Developing Grid Standards



Source: Ian Foster - foster@mcs.anl.gov

Distributed* ⇒ Utility Computing

*(Distributed systems ⇒ Metacomputing ⇒ Grid Computing / Web Services)



Enterprise Grid Alliance

Accelerating the Adoption of Grid Solutions in the Enterprise

www.gridalliance.org

EGA Participants

Board Members	EMC² where information lives; Fujitsu Computers Siemens; hp invent; intel; NEC; NetApp The evolution of storage.; ORACLE; Sun microsystems
Sponsor Members	Ascential; Paremus
Contributor Members	AMD; Enigmatec Corporation; e-Science; Cassatt; optena; Novell.
Associate Members	the 451 group; AvarSys Incorporated; Cisco Systems; Citrix; Data Synapse; Dell; Force10; MCNC Grid Computing & Networking Services; Qlusters; Topspin; UBS; Univa; Unisys; Voltaire

EGA Technical Working Groups

Five EGA technical working groups are focused on addressing the obstacles to adoption of grid in the enterprise

- Reference model
- Component provisioning
- Data provisioning
- Utility accounting
- Grid security

EGA and other standards bodies

- **The EGA is specifically addressing inhibitors to the adoption of grid computing in the enterprise**
- **The EGA is encouraging liaison with other consortia and standards organizations**
  - Most of the EGA members actively participate in other standards bodies
  - Currently developing reciprocity agreements
  - Evaluating other organization's efforts and developing channels to communicate requirements
  - EGA is working with: DMTF, GGF, Globus, SDL, SNIA and others

GGF and EGA working together

GGF – use cases, architectures, specifications and best practices

EGA – Enterprise requirements, solutions and adoptions

OGSA

Web Services Standards & Distributed Management Standards

Reference Implementations & Interoperability testing

Pervasive products and services

EGA will deliver detailed Enterprise use cases in 3-4 months

EGA will then evaluate OGSA as a solution to these use cases

EGA will contribute use cases to GGF and provide feedback on OGSA

Globus and EGA working together?

Globus – ????

EGA – Enterprise requirements, solutions and adoptions

?

?

Do EGA and Globus work together?

Will EGA then evaluate Globus as a solution in these use cases?

EGA will deliver detailed Enterprise use cases in 3-4 months

The Case for Collaboration

- Opportunity to make significant progress on industry standard distributed computing (a.k.a. grid, etc.)
- In Grid space, must address how we describe, discover, access, monitor, manage, account and charge for resources
  - Magnitude and scope of the work is greater than any one standards organization – requires collaboration
  - A lot of related work is already carried out in numerous standards bodies
- Collaboration is everyone's best interests
  - Avoid overlapping efforts and competing specifications
  - Share common frameworks, taxonomy, roadmaps, etc.
  - Increase efficiency and deliver more effective results faster

“It takes a community to raise a (useful) enterprise grid”

Sun, Globus, and the Enterprise Grid: Collaborating on the Future

Peter ffoulkes
peter.ffoulkes@sun.com