Dependability Issues in the Emerging Web Services-Based Grid Computing Standards

Matti Hiltunen
AT&T Labs - Research
Florham Park, NJ 07928, USA
hiltunen@research.att.com

Acknowlegements:

Part of material based on Xianan Zhang, Matti Hiltunen, Keith Marzullo, Rick Schlichting, “Managing Service States According to Durability”, Draft.

Other grid-collaborators: Dr. Francois Taiani (Lancaster U), Ryoichi Ueda, Toshiyuki Moritsu (Hitachi).

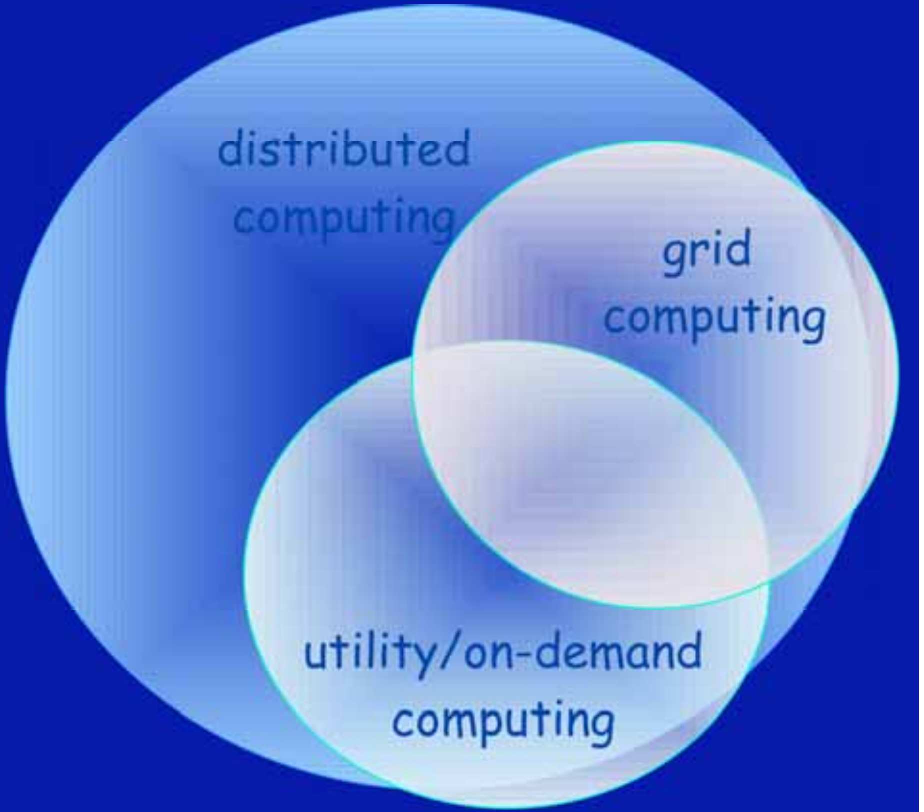
Opinions expressed in this talk do not reflect those of AT&T.

AT&T

Concepts

Grid computing: collaborative use of computers, networks, databases, scientific instruments, and data; potentially owned and managed by multiple organizations.

Utility/on-demand computing: computing resources are made available to the user as needed. The resources may be maintained within the user's enterprise, or made available by a service provider.



AT&T

Why should we study dependability in grid computing?

Because it is there.

- Grid computing seems to be catching on both in academia and industry (Intel, Cadence, Wachovia, Hartford, Bank of America, Johnson & Johnson, ...)
- Dependability becoming more important due to the size of grid platforms and new grid application domains.
- Opportunity to apply our techniques.

There might be some interesting (new) problems and possibilities in grid computing.

What is different in grid computing?

Scale: grids of thousands of machines common.

- Failures will occur frequently.
- Automatic recovery (management) very useful.

Geographical distribution: world-wide grids common.

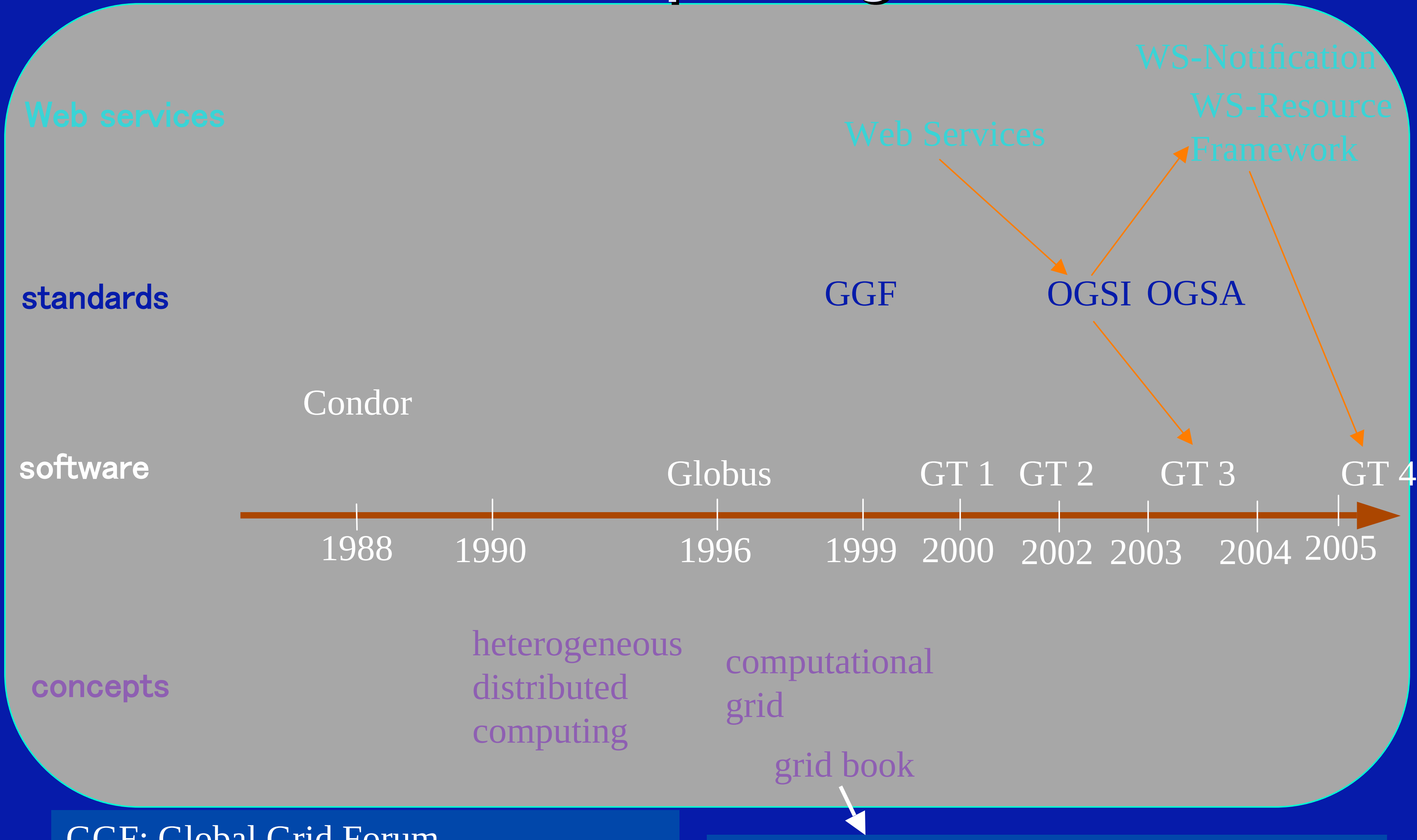
- Transfer of large volumes of data across the world.

Potentially span multiple administrative domains.

- Trust issues: executing tasks on potentially untrusted computers (secret data, secret code, secret results).
- Accounting/billing issues: various types of fraud possible.

Grids (clusters) popular targets for attackers (a high-performance grid makes a powerful botnet).

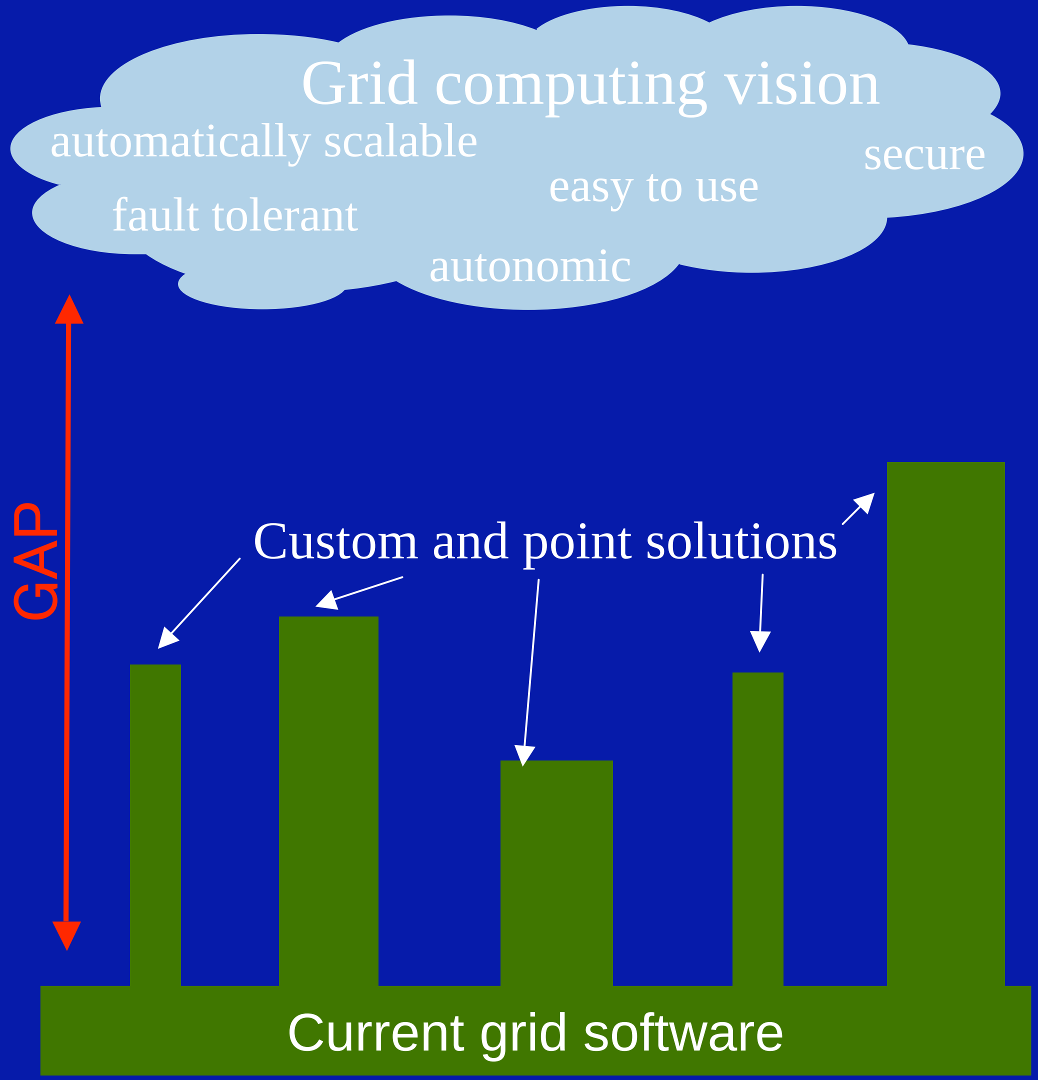
Grid computing timeline



GGF: Global Grid Forum
OGSI: Open Grid Services Infrastructure
OGSA: Open Grid Services Architecture

The Grid: Blueprint for a New Computing Infrastructure, Foster and Kesselman

Vision vs. current status



Why should we care about standards?

The concept of grid computing is not based on, or require, any standards.

However ..

- Interoperability requires standards (can your grid platform talk to mine).
- Commercial users of grid computing demand standard compliance to avoid locking in with one vendor.
- Basing your work on existing standards and existing implementations can speed up your work (do not need to implement everything from scratch – just the parts that you are interested in).
- Publishability (think transport protocols vs. TCP).

Opportunity for impact:

- The specifications at GGF are still in early stages – it is still possible (easy) to define or refine these specifications.
- It is possible to add your pieces into open source grid platforms such as Globus.

AT&T

Current direction: Grid Services

Grid computing is defined as an extension to web services.

Grid service = “web service that is designed to operate in a Grid environment, and meets the requirements of the Grid(s) in which it participates.”

Grid Computing Platform = a collection of grid services (infrastructure services).

WSRF (Web Services Resource Framework): extension that allows the implementation of stateful grid services.

Stateful grid service = web service + WS-Resourses

AT&T

Is this a good idea?

Positives:

- Can leverage existing web service platforms and web service standards.
- Ride on the popularity of web services – easier acceptance.

Negatives:

- Large performance impact (response time from 100+ms to 10s of seconds for trivial grid services in Globus 3.9.4).
    - Note that web service protocols are only needed for interaction between different grid services (not between nodes in a grid application).
- Complexity of the resulting grid middleware (number of layers).
- WS specifications are still evolving and competing.

AT&T

Too many standards

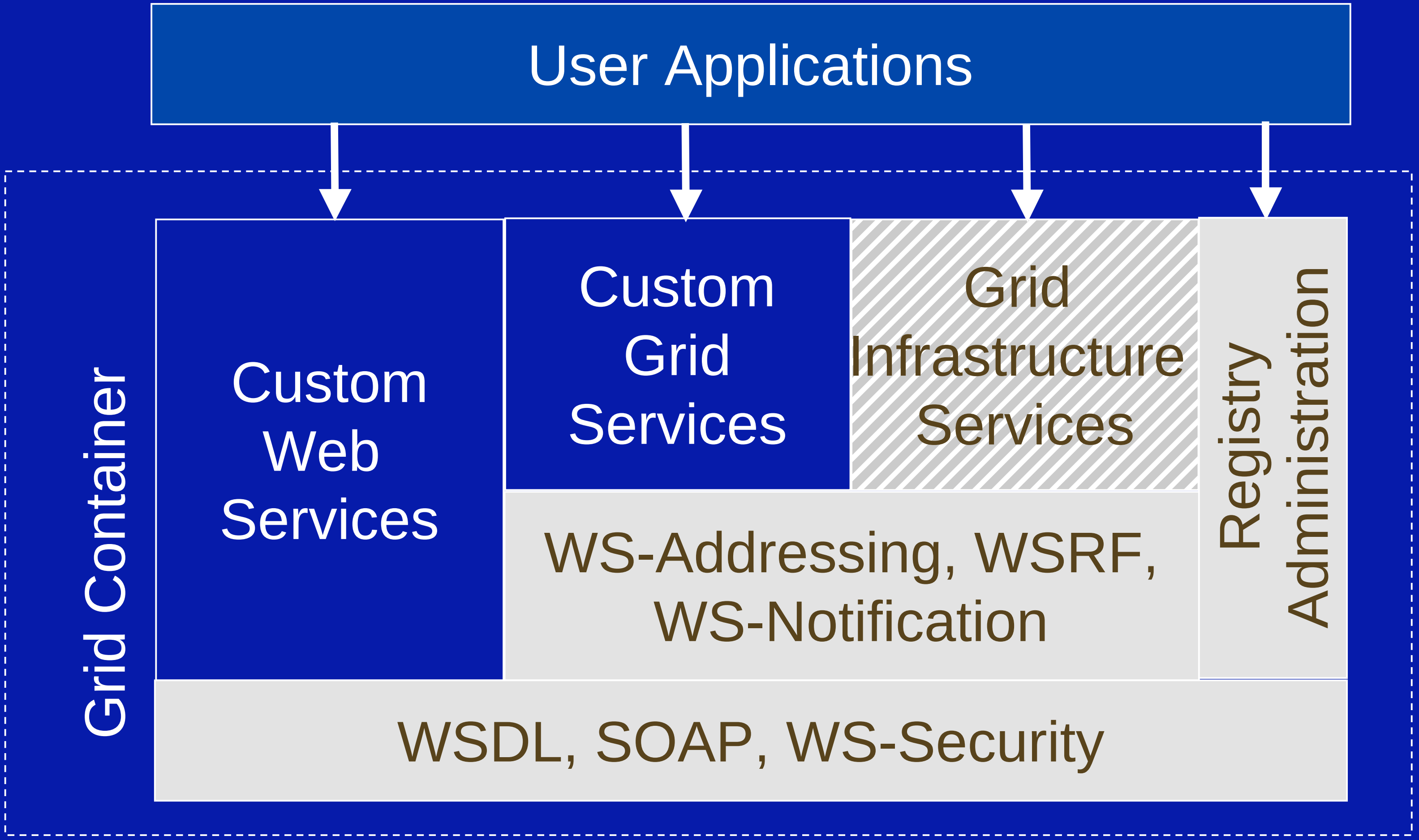
Grid computing is now being defined by standards, specifications, and recommendations from multiple organizations:

- GGF (Global Grid Forum): OGSA, OGSA-DAI, DRMAA, GridFTP, GridRPC, …
- OASIS (Organization for the Advancement of Structured Information Standards): WS-Resource Framework, WS-Reliability, WS-Security, WS-Transactions, …
- W3C (World Wide Web Consortium): WSDL, SOAP, …
- EGA (Enterprise Grid Alliance): Reference Architecture.

Existing grid computing solutions do not fully match, or implement only a part of, these recommendations:

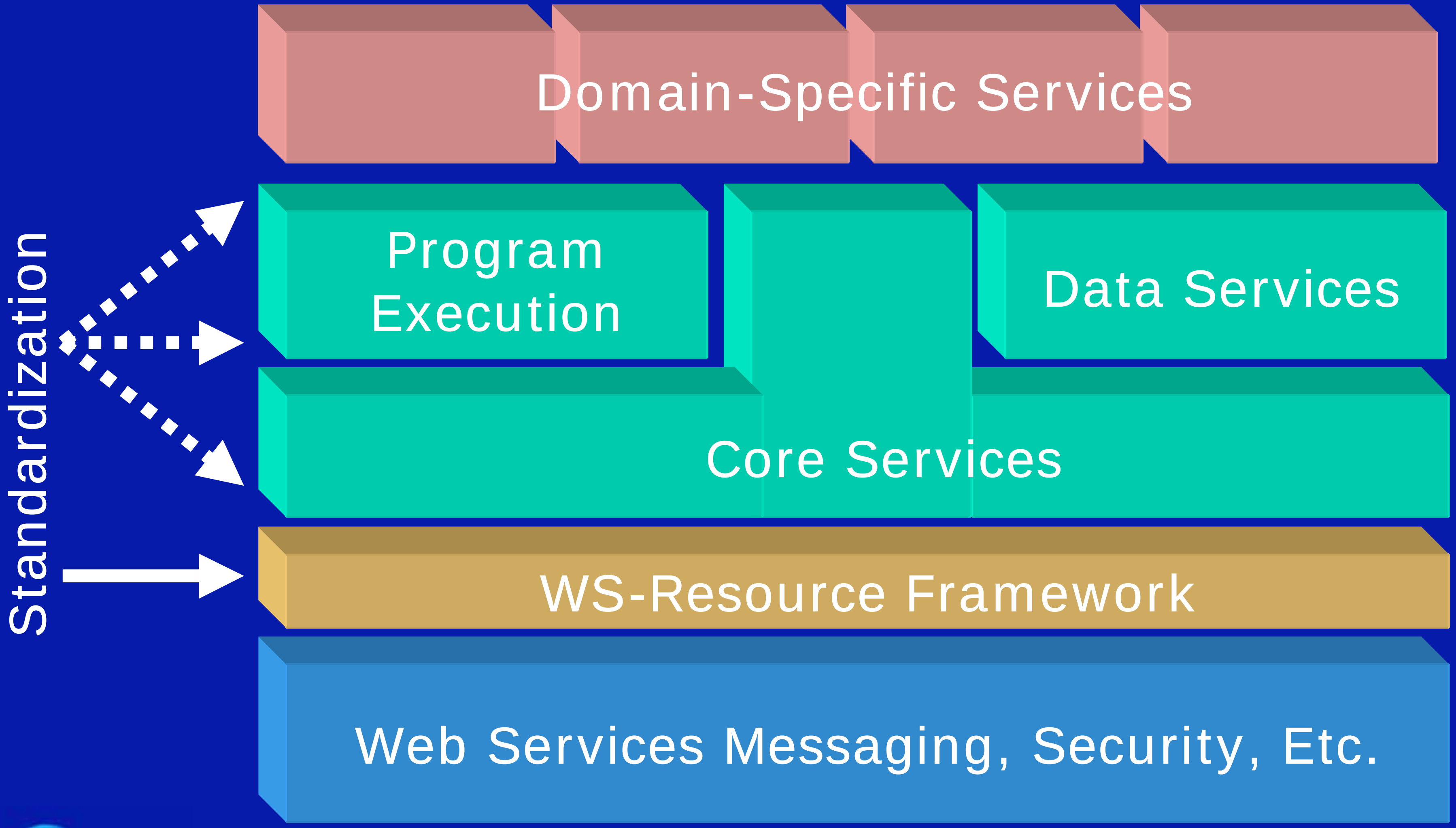
- Globus, Condor, Sun GridEngine, DataSynapse, Grid MP Enterprise (United Devices), …

Grid Services



AT&T

Open Grid Services Architecture



OGSA: Lots of services!!

Execution Management Services:

- Job Manager, Execution Planning Service, Candidate Set Generator, Reservation services, Deployment and Configuration Service, Naming, Information Service, Monitoring, Fault-Detection and Recovery Services, Auditing, Billing, and Logging Services.
- To start the execution of a job, half a dozen service interactions may be required!

Data Services

Resource Management Services

Security Services

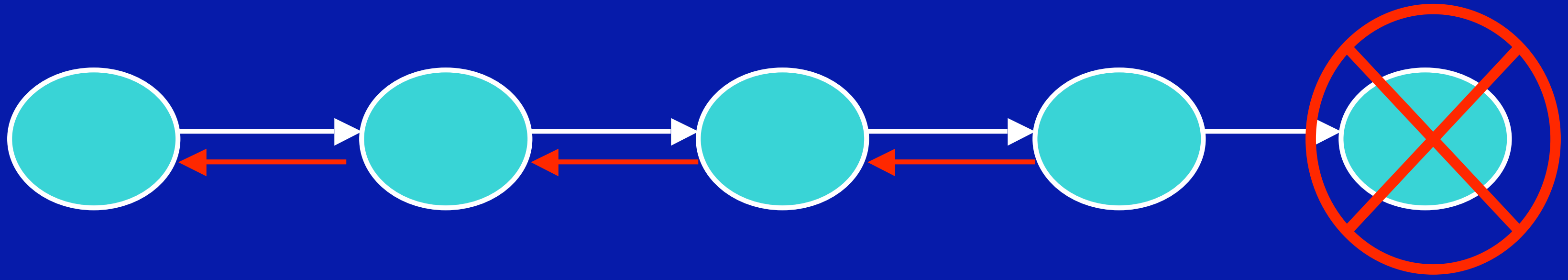
Self-Management Services

Information Services

AT&T

Importance of high availability

Grid Service Architecture = “System where the failure of a service you have never heard of prevents you from running your grid application”?



It is important for the grid infrastructure services to be highly available since each service may affect most/all of other grid services and grid applications.

Dependability in Grid Services

Different grid applications have different requirements.

Traditional scientific grid applications did not have many dependability requirements:

- no security, real-time
- domain specific fault-tolerance techniques:
  - parallel computation: checkpointing.
  - master-worker: easy to deal with the failure of worker
  - fault tolerance used to reduce average latency of task execution.

AT&T

Relevant specifications

Reliability:

- WS-Reliability: Reliability guarantees for asynchronous message delivery including Guaranteed delivery, Duplicate Elimination, and Message Ordering. The receiver of a Reliable Message must store the message in persistent storage and mask any recovery actions.
- WS-Transactions: two flavors of transactions – 2 phase commit, business transaction.
- Nothing to ensure high availability of grid services.

Security:

- WS-Security: message integrity, confidentiality, and single message authentication; support for security tokens (e.g., certificates).
- GGF: focus on authorization: who is allowed to use what resources/services.

Real-time:

- Nothing to my knowledge

Highly Available Grid Services

Availability can be provided on

- Hardware level.
- (WS-)Resource level.
- (Grid) Service level.
- On composite service-level: Independent services provided by different providers collaborate to provide highly available service.

Availability can be provided by the services themselves and/or external services (Monitor/Controller Service).

May be completely transparent to the client or require some client interaction (rebinding to the service).

AT&T

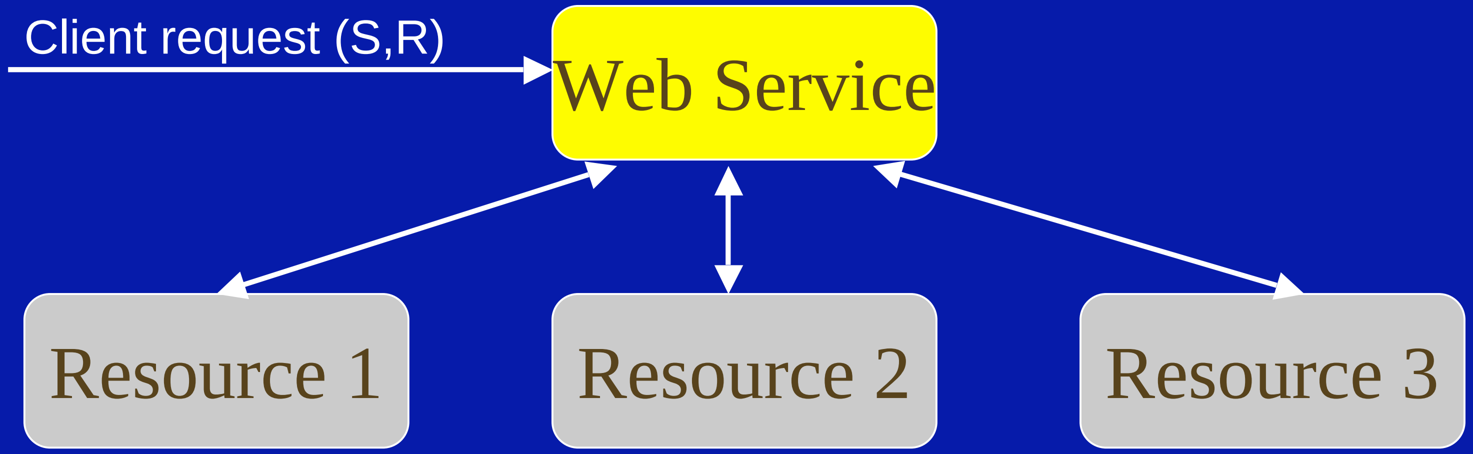
State in distributed services

- Distributed Object Model (CORBA/Java RMI):
  - State part of the object.
- Open Grid Services Infrastructure (OGSI):
  - Grid Service is a stateful “object”.
- Web Services:
  - Officially stateless, service state is implicitly maintained in a database (typically).
- WS-Resource Framework (WSRF):
  - A refactoring and evolution of OGSI.
  - Stateless (Web) Service + stateful resources
  - A web service reference contains both the service and the resource the service is to operate on.

Stateful grid service

Based on WS-Resource Framework (WSRF)

- Separate the state of the service from the function of the service.



Service State Characteristics

Service state (WS-resources) can be characterized by attributes:

- Durability: what kinds of failures, and how many, should the state survive.
- Consistency: read-only, time-bounded staleness allowed, commutative updates, …
- Latency: response time for read/write.

Different mechanisms for providing durability with different characteristics:

- Database: normal, in-memory, replicated
- Disk: local disk, RAID disk
- Replicating across a set of servers

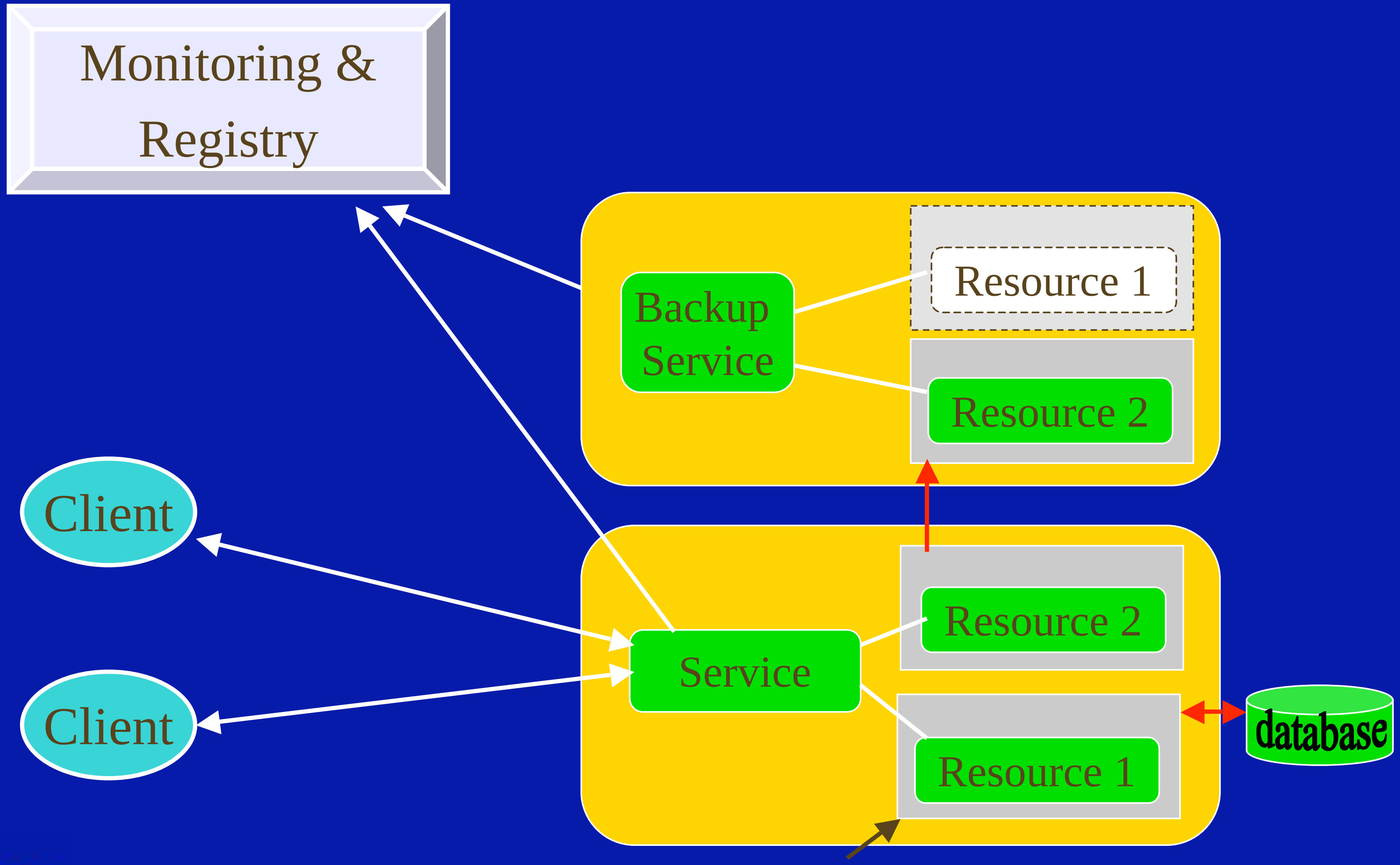
……

Research idea

1. By making resources (= state) durable, it is easy to construct highly available grid services.
2. Durability level and mechanism should be easily customizable for each resource.
3. Mechanisms should be reusable.
   - Durability wrappers: database wrapper, primary-backup wrapper.
4. Goal of automatic service + resource transformation.

AT&T

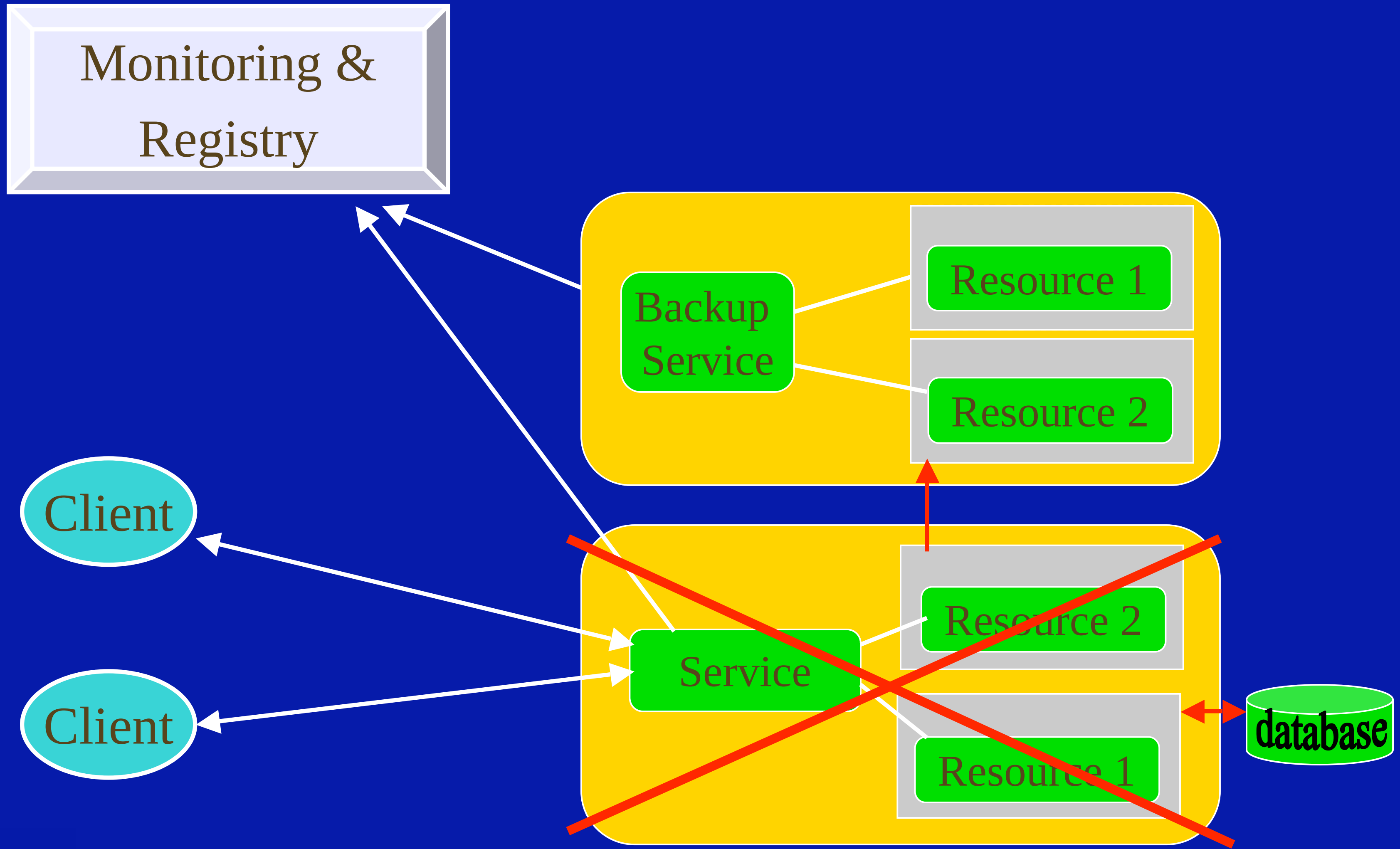
Proposed Architecture



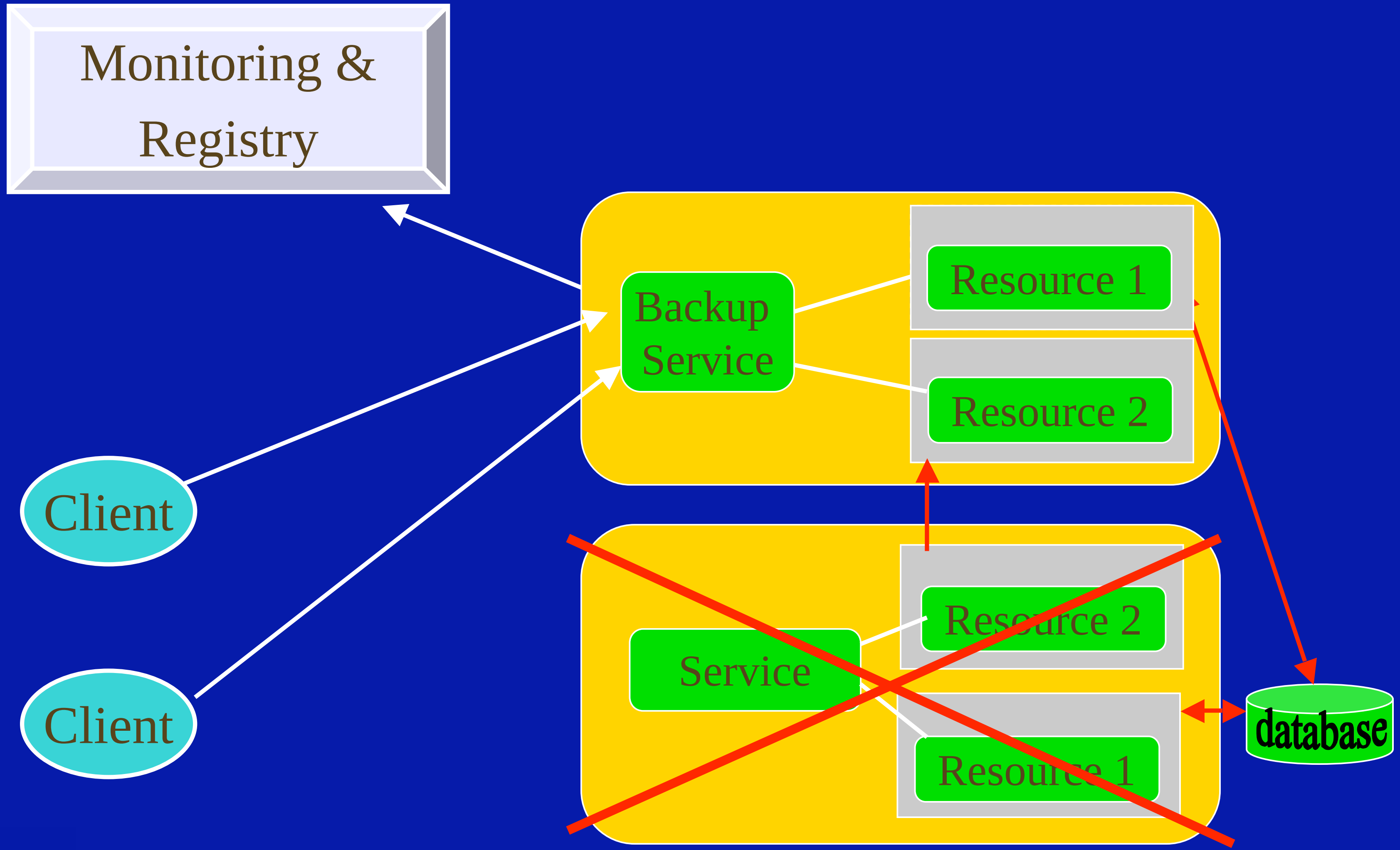
Durability wrapper

AT&T

Failure


Monitoring & Registry
Backup Service
Resource 1
Resource 2
Client
Client
Service
Resource 2
Resource 1
database

Recovery



AT&T

Goals

Transparency of durability:

- Web service and resources are written without considering durability.

Challenges:

- Different state representation.
- Atomic action boundaries (maintaining state consistency between resource and its backup).
- Different recovery operations.

Solutions:

- Java dynamic proxies used to wrap resources.
- Configuration files to provide information to “durability compiler”

AT&T

“Durability compiler”

Generates code to make the web service highly-available:

- Uses configuration file + web service and resource Java code.
- Generates a durability proxy for each resource.
- Extends web service code:
  - ``I’m alive’’ message sending to Monitoring Service
  - Invocations to resources to indicate action boundaries (“begin action”, “end action”)
  - Code for “Backup Service”
  - Might be possible to implement using dynamic proxies as well.

AT&T

Configuration File

General information about the web service

– Such as the service URL, the resources the service uses…

- The information on the state update for each resource class.
- Information about transaction.

Example: Info for database proxy

Proxy Type	Database proxy	
Initialization	CREATE TABLE bills (clientID INT, balance INT) ENGINE=INNODB;	
Failover	SELECT * FROM bills; For (each line) insertBill(clientID, balance)	
Update methods	insertBill	INSERT INTO bills VALUES (arg[0], arg[1]);
	setBill	UPDATE bills SET balance=arg[0] WHERE clientID=arg[1];

AT&T

Example 1: Counter Service

The Counter Service uses WSRF to maintain state: the value of the counter.

- Service RTT:
  - The original counter service – 139 ms.
  - Using primary-backup proxies – 139 ms.
  - Using a database proxy – 170 ms.

Example 2: Matchmaker Service

Service that maps available computing requests to client requests (and accounts for usage).

State:

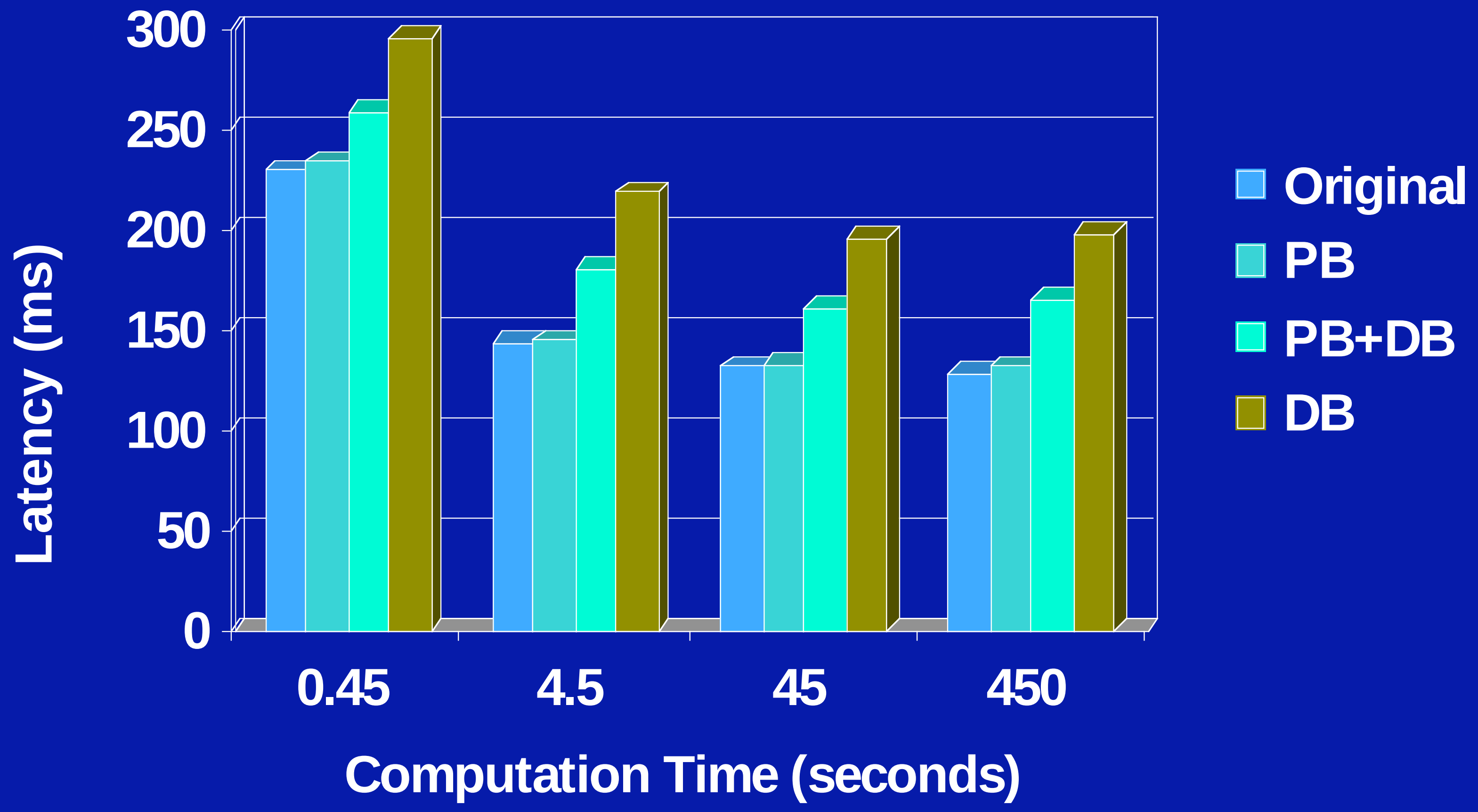
- a *machine queue* – a queue of available machines.
- an *account set* – billing records for all the clients.

Characteristics:

- machine queue can be reconstructed with time,
- accounting info impossible to reconstruct.

AT&T

Matchmaker Performance



AT&T

Future directions

- “Fundamental” fault-tolerance issues (Paxos).
- Grid specific security issues:
  - How to run secret algorithms or algorithms that use secret data in a shared grid environment
  - How to protect the grid environment from rogue grid applications (DoS, spying, etc)
- Performance improvement.
- Personal goal: write some “real” grid applications.

Conclusions

- Grid computing is here to stay.
- Dependability is becoming more important.
- There are some novel research challenges.
- Do we want to wait for somebody else to make grid computing dependable?

Publications

- X. Zhang, D. Zagorodnov, M. Hiltunen, K. Marzullo and R. Schlichting, “Fault-tolerant Grid Services Using Primary-Backup: Feasibility and Performance”, Cluster 2004.
- R. Wu, A. Chien, M. Hiltunen, R. Schlichting, S. Sen, “A High Performance Configurable Transport Protocol for Grid Computing”, CCGrid 2005.
- R. Ueda, M. Hiltunen, R. Schlichting, “Applying Grid Technology to Web Application Systems”, CCGrid 2005.
- F. Taiani, M. Hiltunen, R. Schlichting, “The Impact of Web Services Integration on Grid Performance”, HPDC 2005.

AT&T