Service-ORiented Computing EnviRonment (SORCER)

Michael Sobolewski
sobol@cs.ttu.com

Distributed Megaprogramming

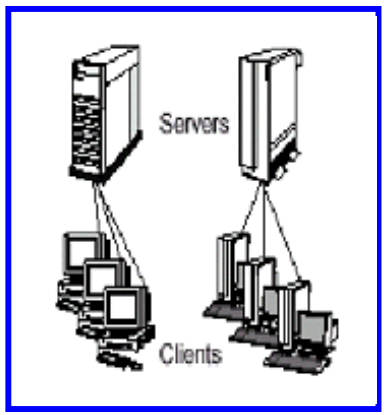
I do not believe traditional tools, technologies, and methodologies support

Distributed Megaprogramming, Service-Oriented Programming, or Self-Aware Computing

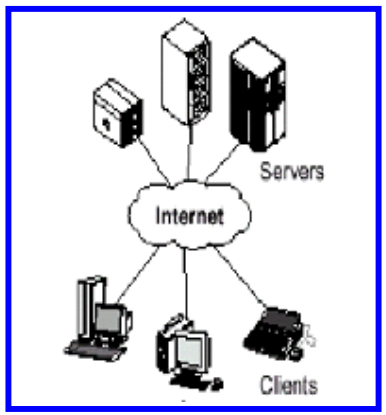
Presentation Agenda

- Distributed Self-Aware Service-Oriented Computing
- Federated S2S Environment
- Job/Task/Context/Method Paradigm
- SORCER Functional Architecture
- Design Issues (UML diagrams)
- Summary

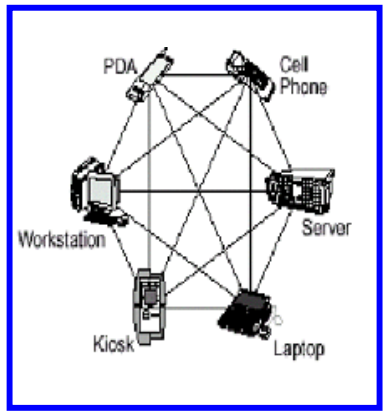
Evolution of Computing



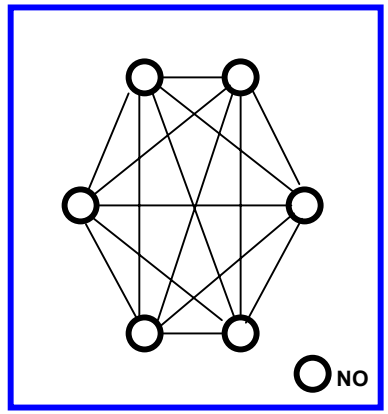
Client-server silos



Web-based computing



Peer-to-Peer

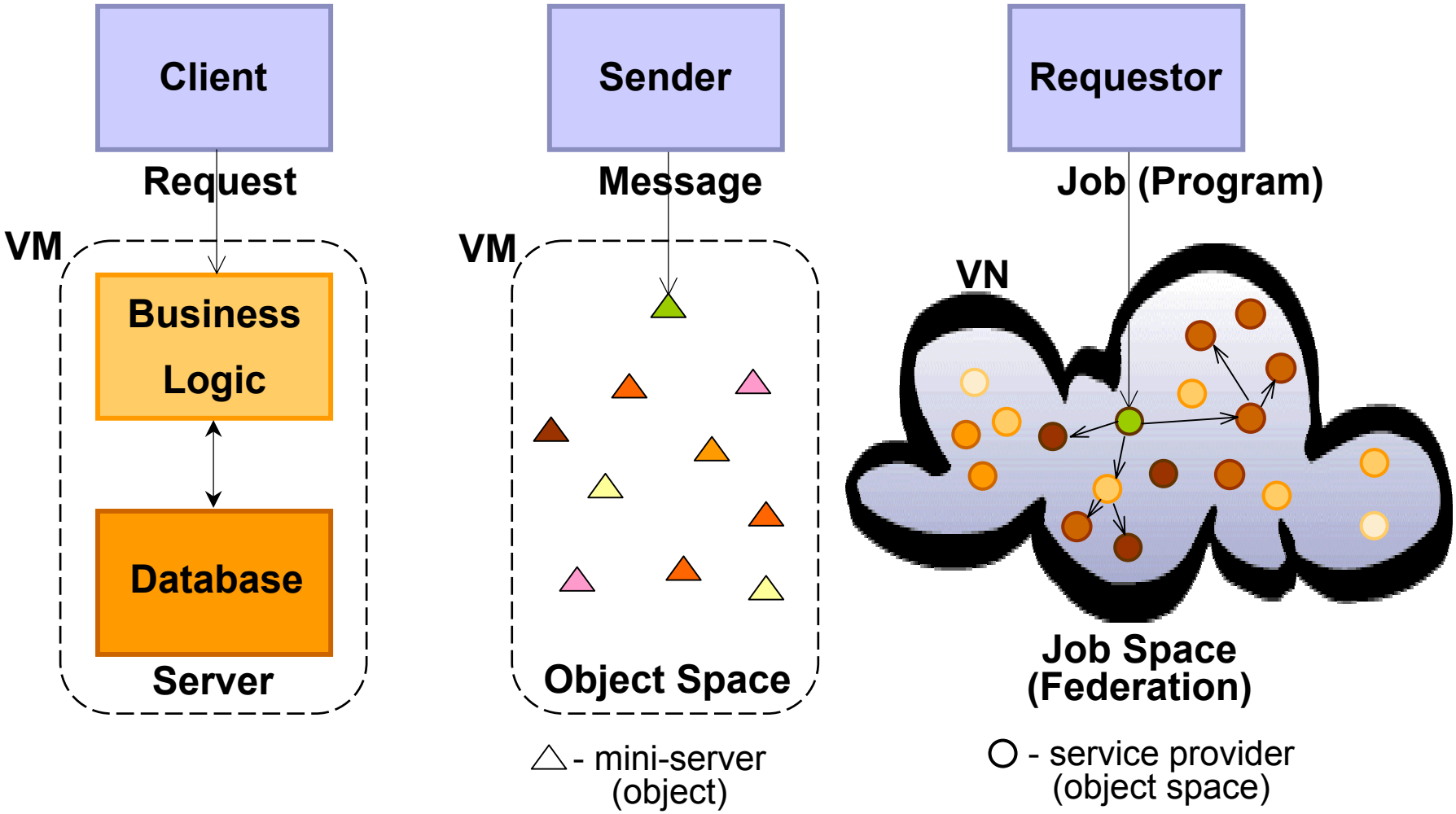


Service-to-Service

Object Space vs. Job Space

TEXAS TECH UNIVERSITY Computer Science

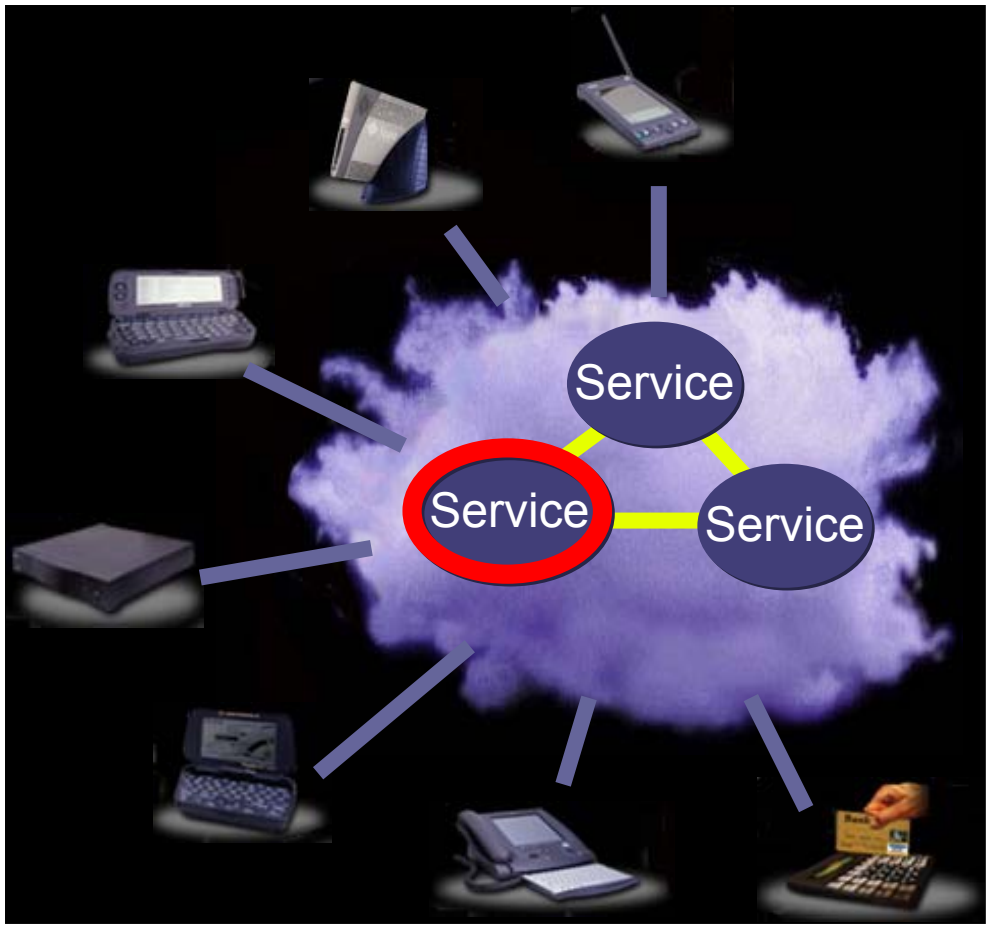
Data, Operations, and Control Strategy



Self-Aware Service-Oriented Computing

- A federation knows what is doing
- Learn from experience and adapt to surprises
- Is aware of its behavior and explain itself
- Is able to anticipate different scenarios and predict and plan for novel futures
- It would learn, not crash, when faced with a new situation
- Self-testing, self-debugging, and self-explaining within a federation

SORCER Vision

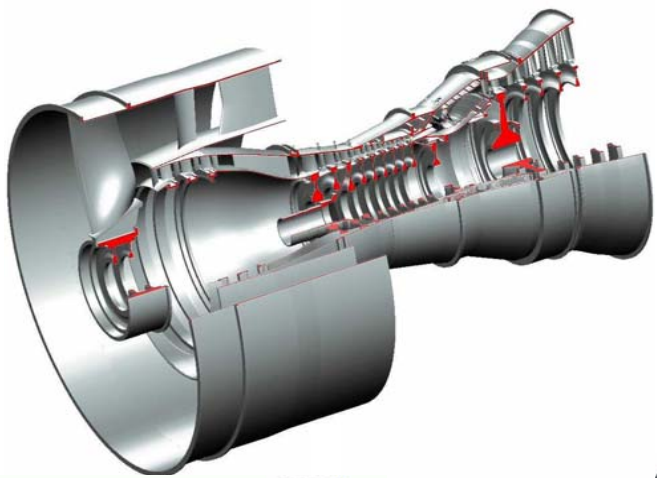


Federated S2S environment to ...

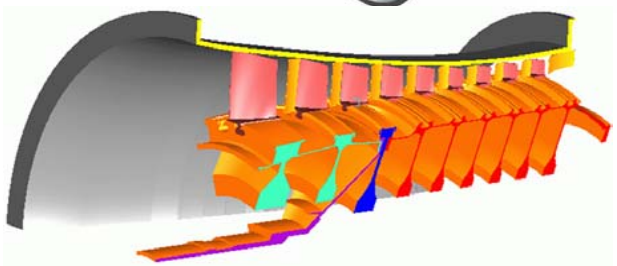
- Service — Build new services
- Service — Convert legacy apps to dynamic SORCER services (J2EE™ technology)
- Assemble SORCER services together (RMI, Jini, Rio technologies)
- Create modern clients accessing services

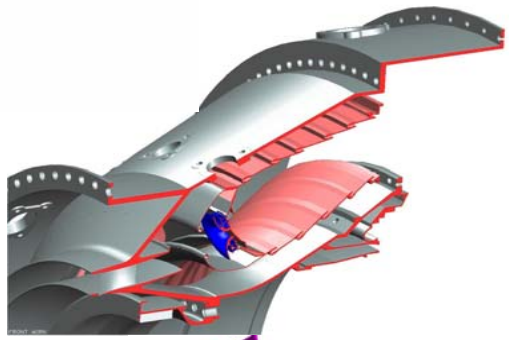
Megaprogramming Domain

System Design

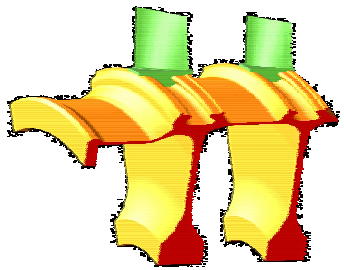


Subsystem Design



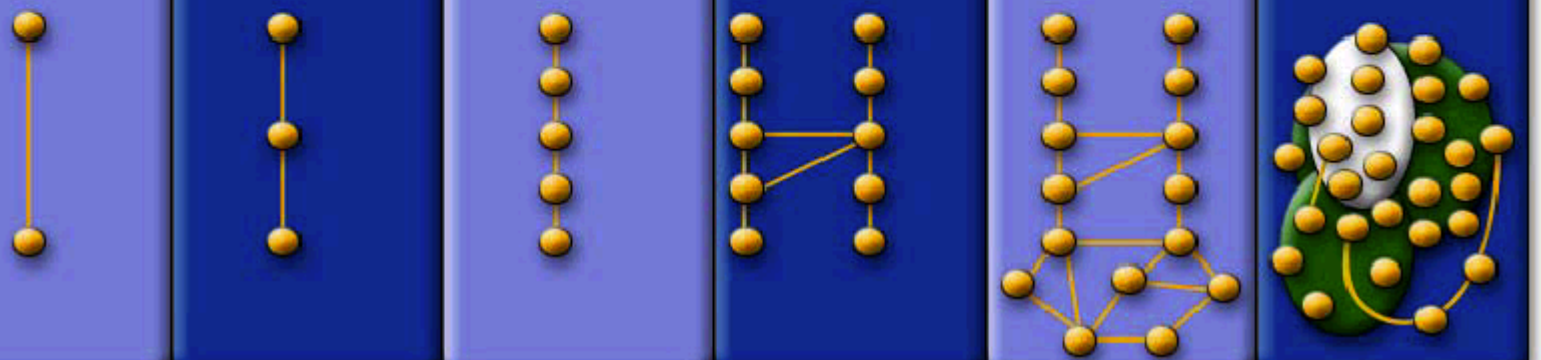


Component Design



Many Kinds of Networks

Client-Server	3/N-Tier	Net Apps	Net Services	Next	After that
The Network Is the Computer	Objects	Legacy to the Web	The Computer is the Network	Network of Embedded Things	Network of Things
100s	1000s	1000000s	10000000s	100000000s	1000000000s
1984/1987	1990/1993	1996/1999	2001/2003	1998/2004	2004/2007
X	X	+HTTP (+JVM)	+XML, Portal	+RMI	Unknown
NIS, NIS+	+ CDS	+ LDAP (*)	+UDDI	+ Jini	+ ?
RPC, XDR	+CORBA	+CORBA, RMI	+ SOAP, XML	+ RMI/Jini	+ ?

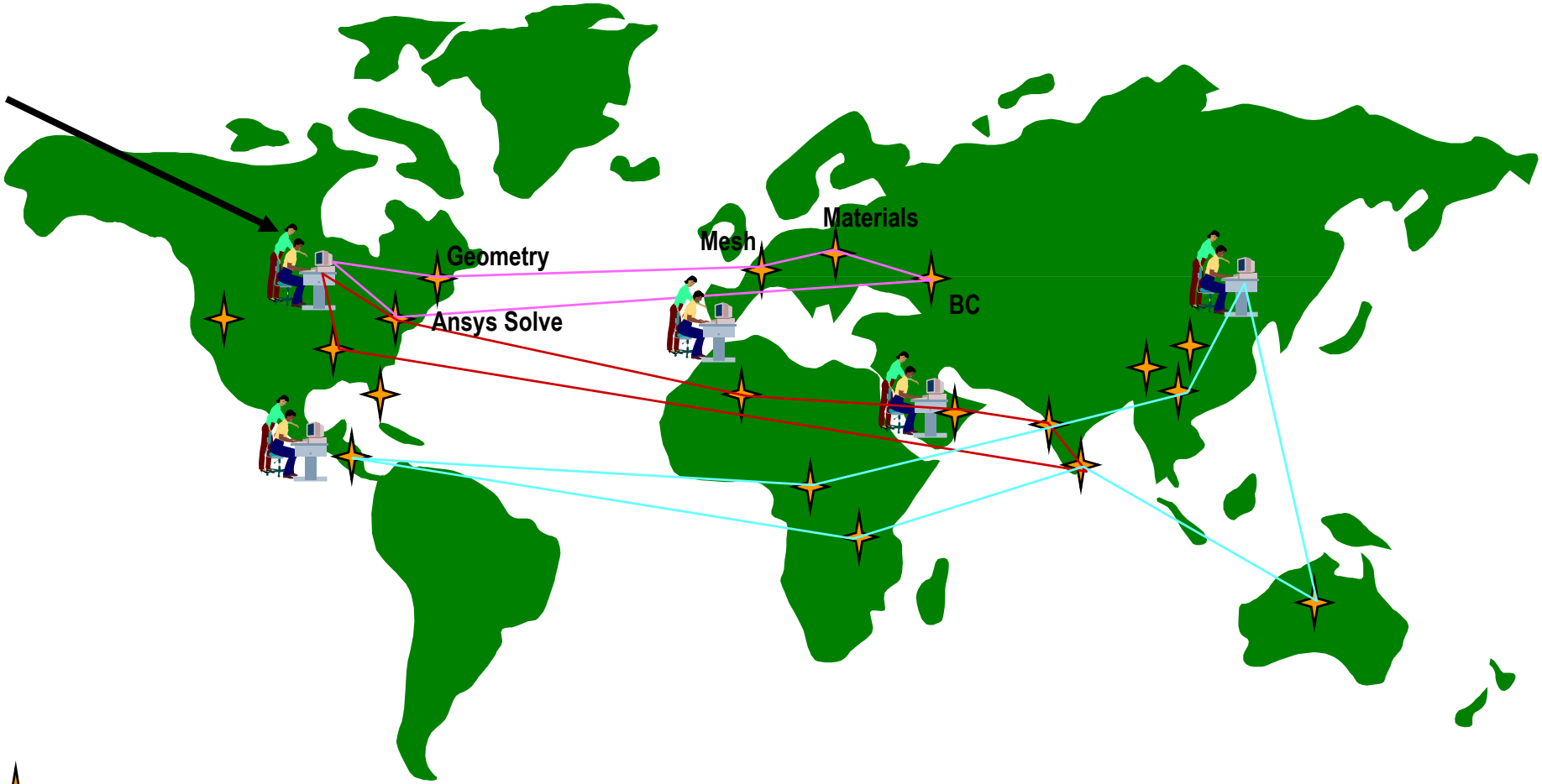


Many Types of Services

SORCER Paradigm

TEXAS TECH UNIVERSITY
Computer Science

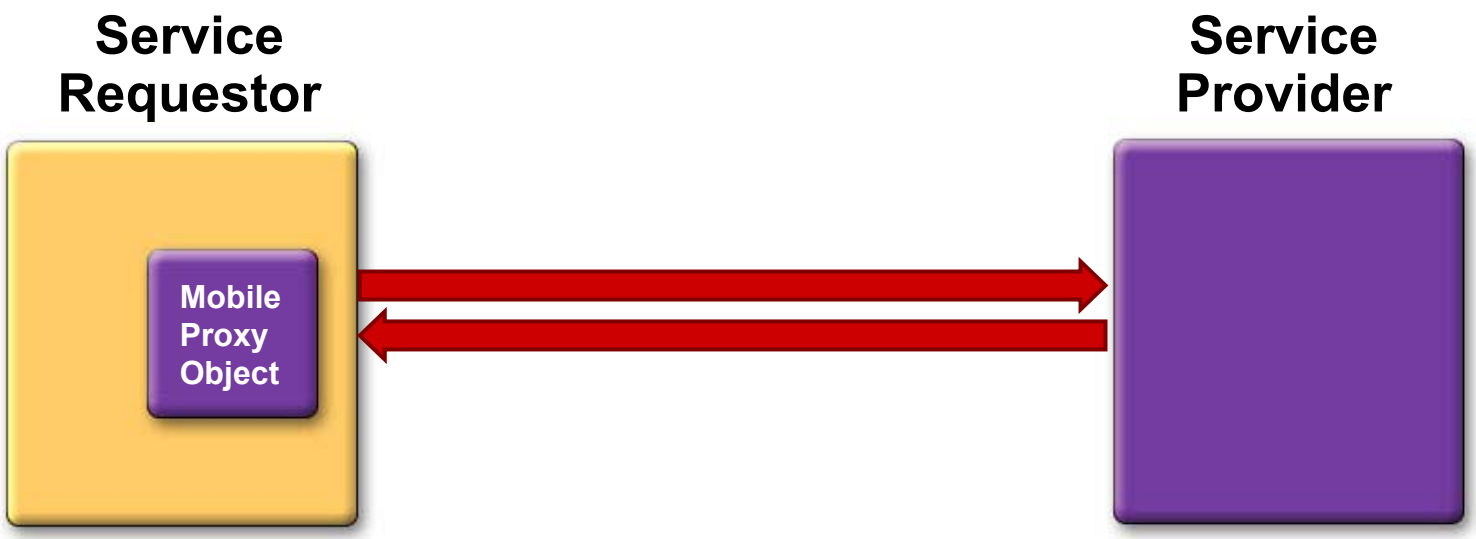
- **Clients Request Services from the Network**
  - DOE Services
  - Analysis Services
  - Optimization Services
- **Clients may not care where or who supplies the services**



SORCER Service: An entity that publishes (by attributes) functional capabilities on the network. (Mesh, Thermal Analysis, Print, etc..)

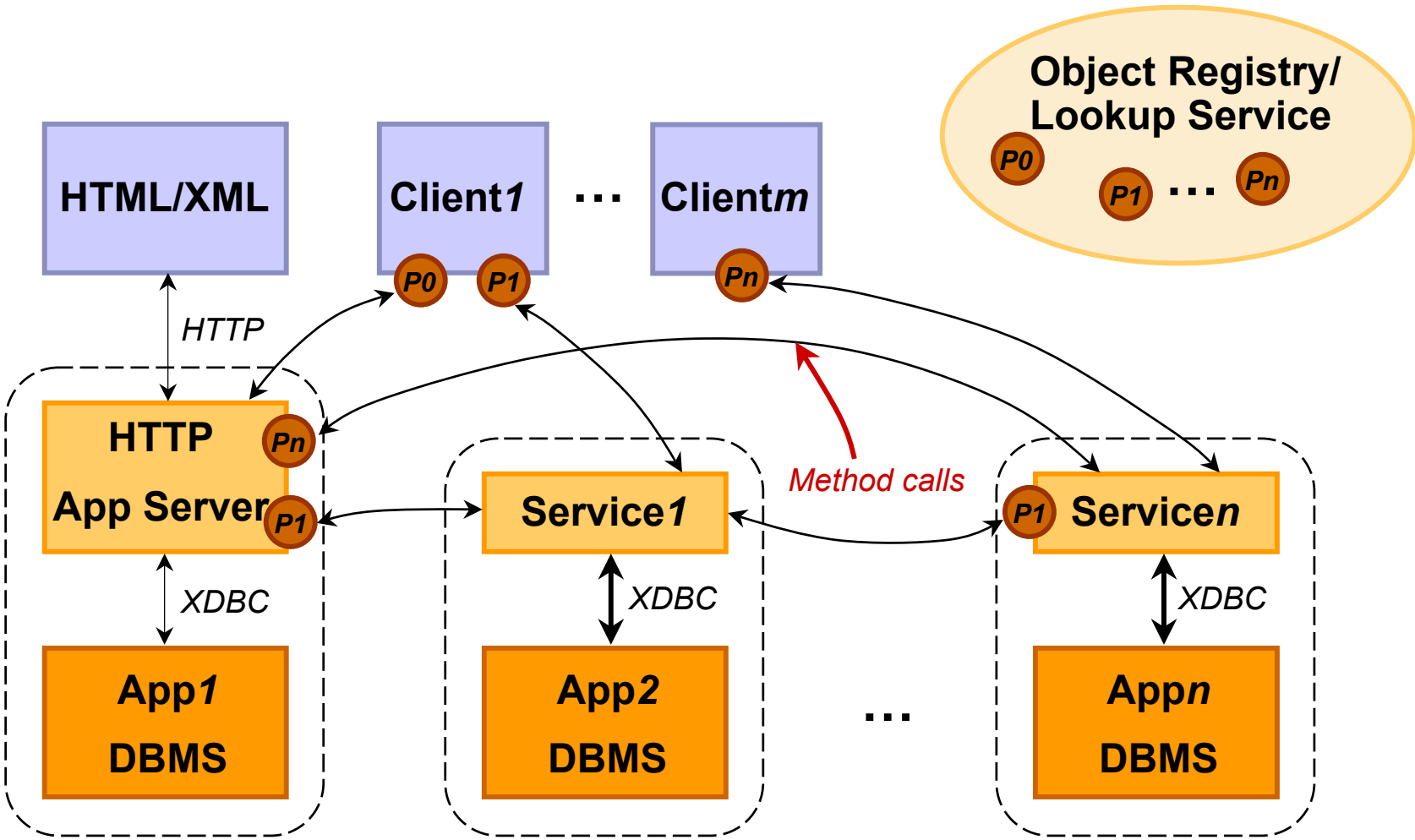
The computer is the network that exposes services to clients AWAT

Provider As a Network Object



Service-Oriented Computing

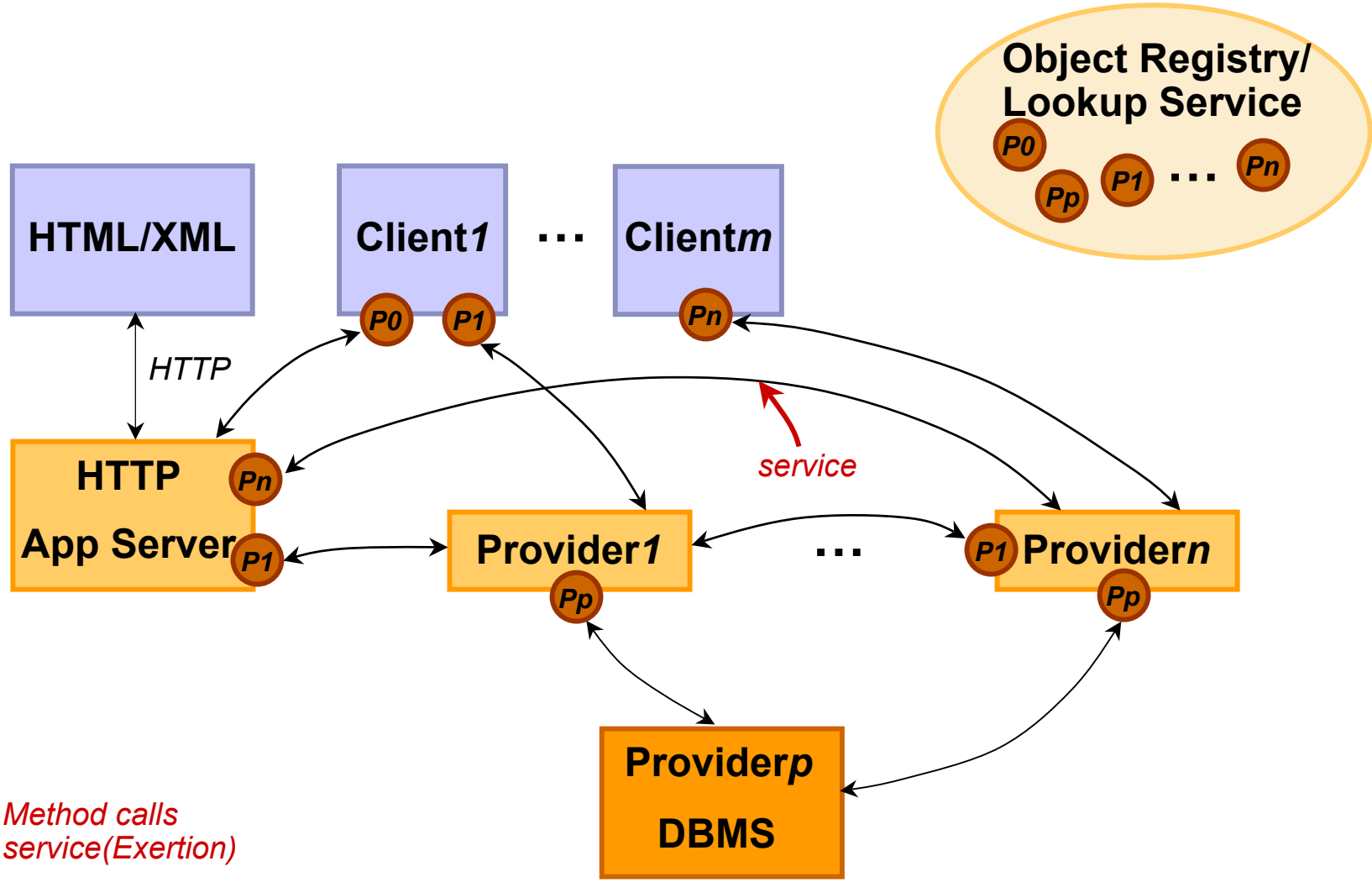
Network Objects



Service-to-Service (S2S)

TEXAS TECH UNIVERSITY Computer Science

Service Providers



Method calls
service(Exertion)

Applying NOO Techniques

- Service activity is a special object of type: Exertion
- Exertions are executed by network objects/service providers of type: Servicer
- Service providers form P2P environment
- Service is requested by calling the method: `service(Exertion)`
- Service providers are identified by a type with methods:

```
public ServiceContext
selector(ServiceContext)
```

Exertion Interface

- All service activities implement this interface:

```
public interface Exertion {
    // Apply this exertion method to the specified context
    public Exertion exert()
    throws RemoteException, ExertionException;

    …
}
```

Service Peer Interface: Servicer

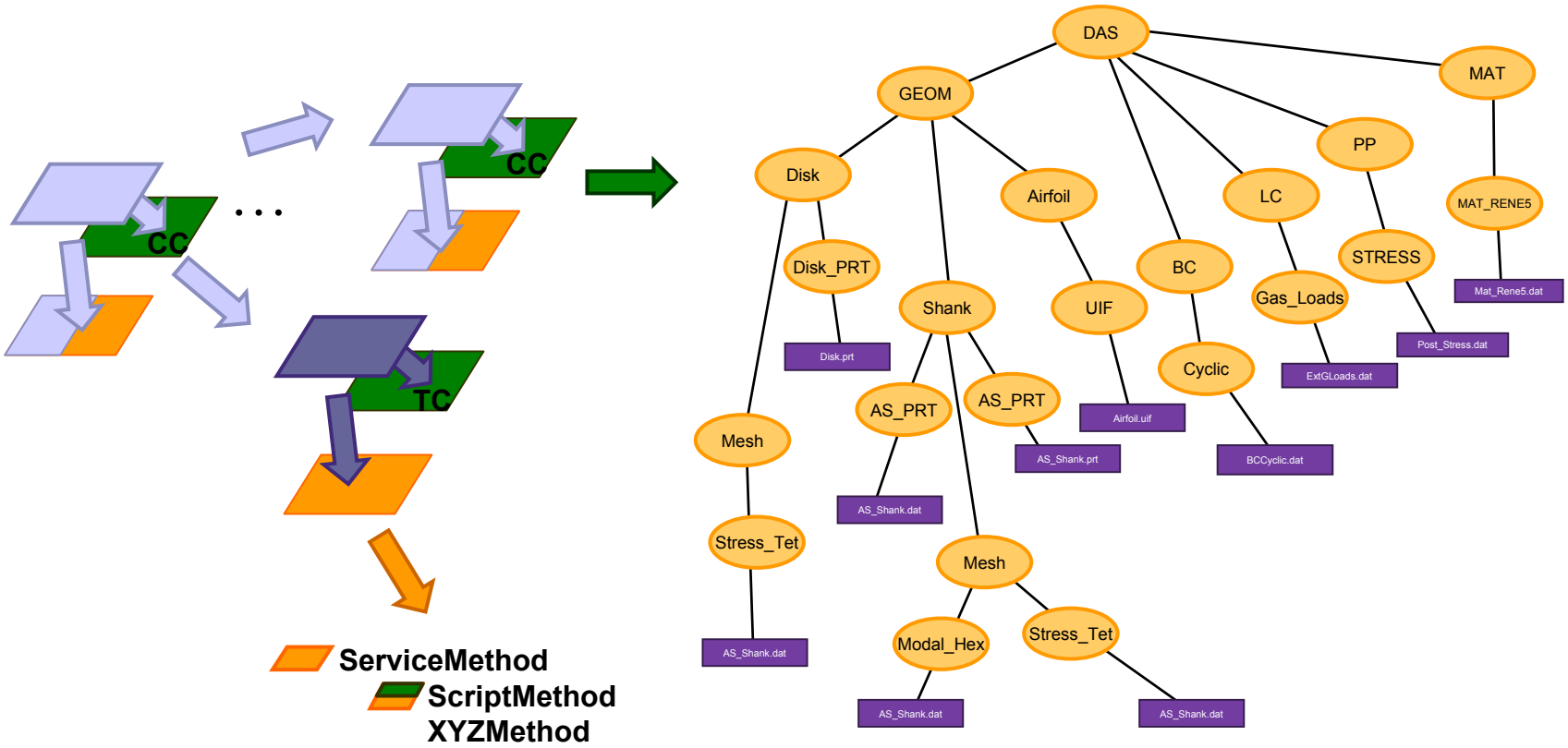
- All services implement this interface:

```
public interface Servicer {
    // Put into action the specified exertion
    public Exertion service(Exertion exertion)
        throws RemoteException, ExertionException;

    // Monitoring methods
    ...

}
```

Contexts and Context Methods



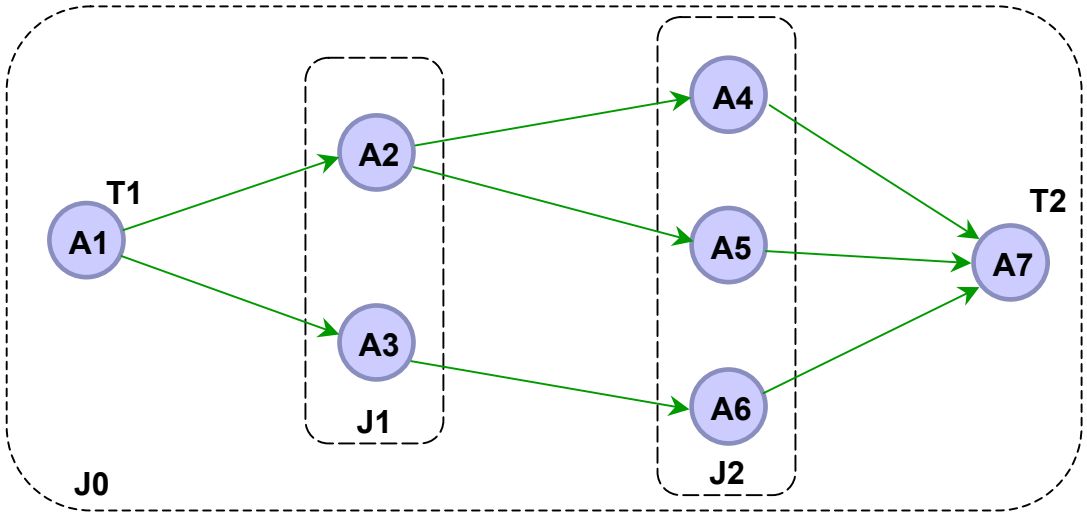
ContextMethod attributes: service type, selector, group, provider name, method type

Method type: preprocess, process, postprocess, append

TC – Task Context, CC – Control Context

Job Task Service Context Service Method

Workflow vs. Job



Workflow

W0 = { (A1, A2), (A1,A3),
(A2, A4), (A2, A5),
(A3, A6), (A4, A7),
(A5, A7), (A6, A7) }

Sequential relationship

Unidirectional aggregation

Inherent control strategy

Explicit all connections

Task/Job

J0 = (T1, J1, J2, T2)

J1 = (A2, A3)

J2 = (A4, A5, A6)

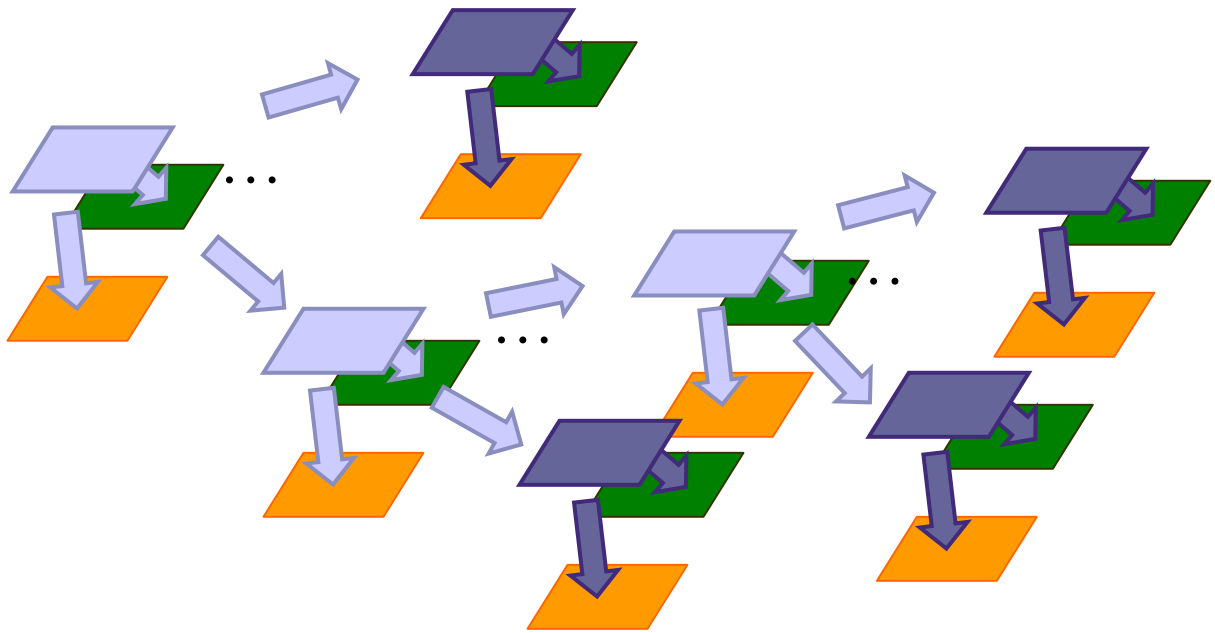
Is-part-of relationship

Bidirectional aggregation

Control strategy separated

Workflow defined implicitly

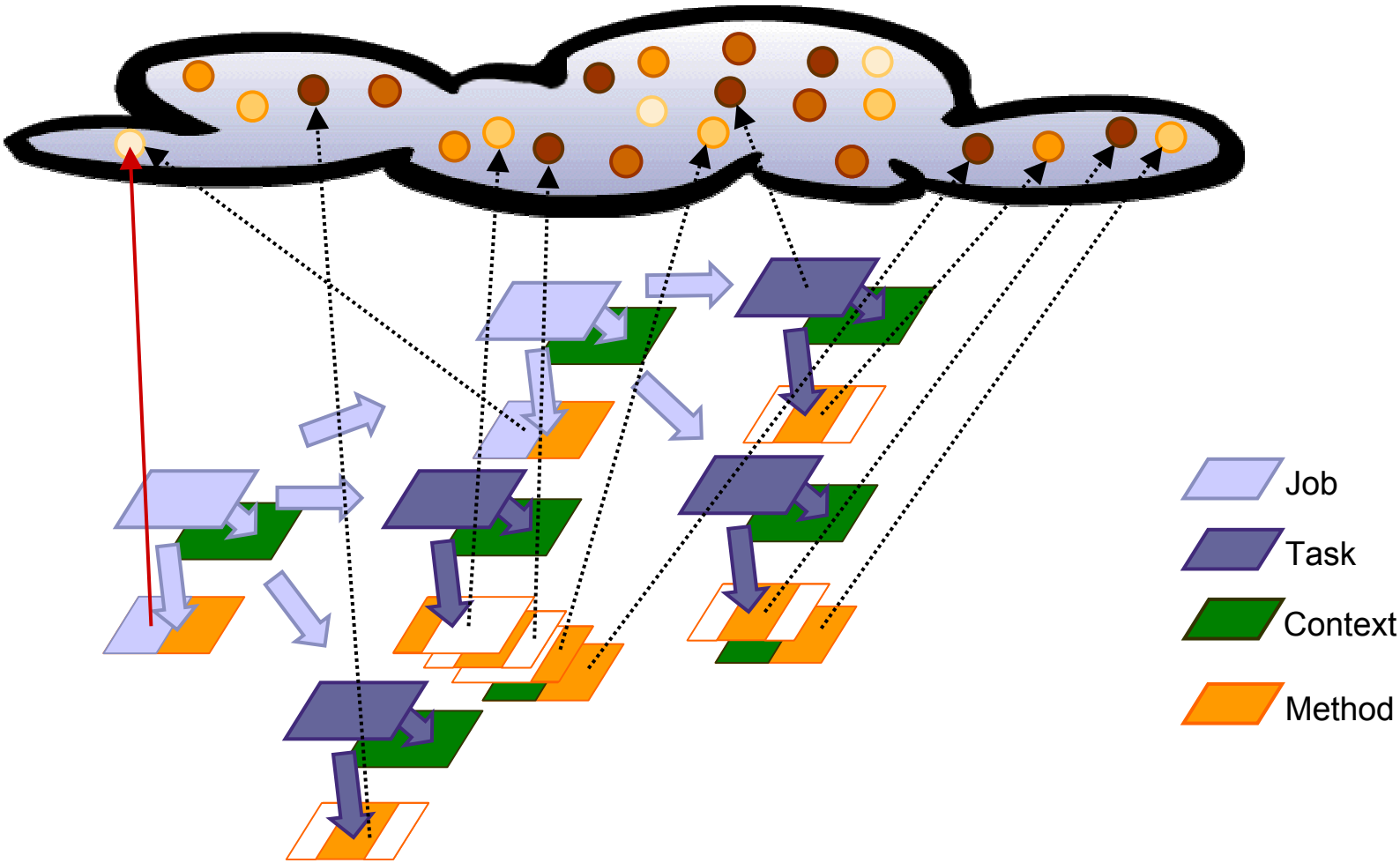
JTCM Paradigm



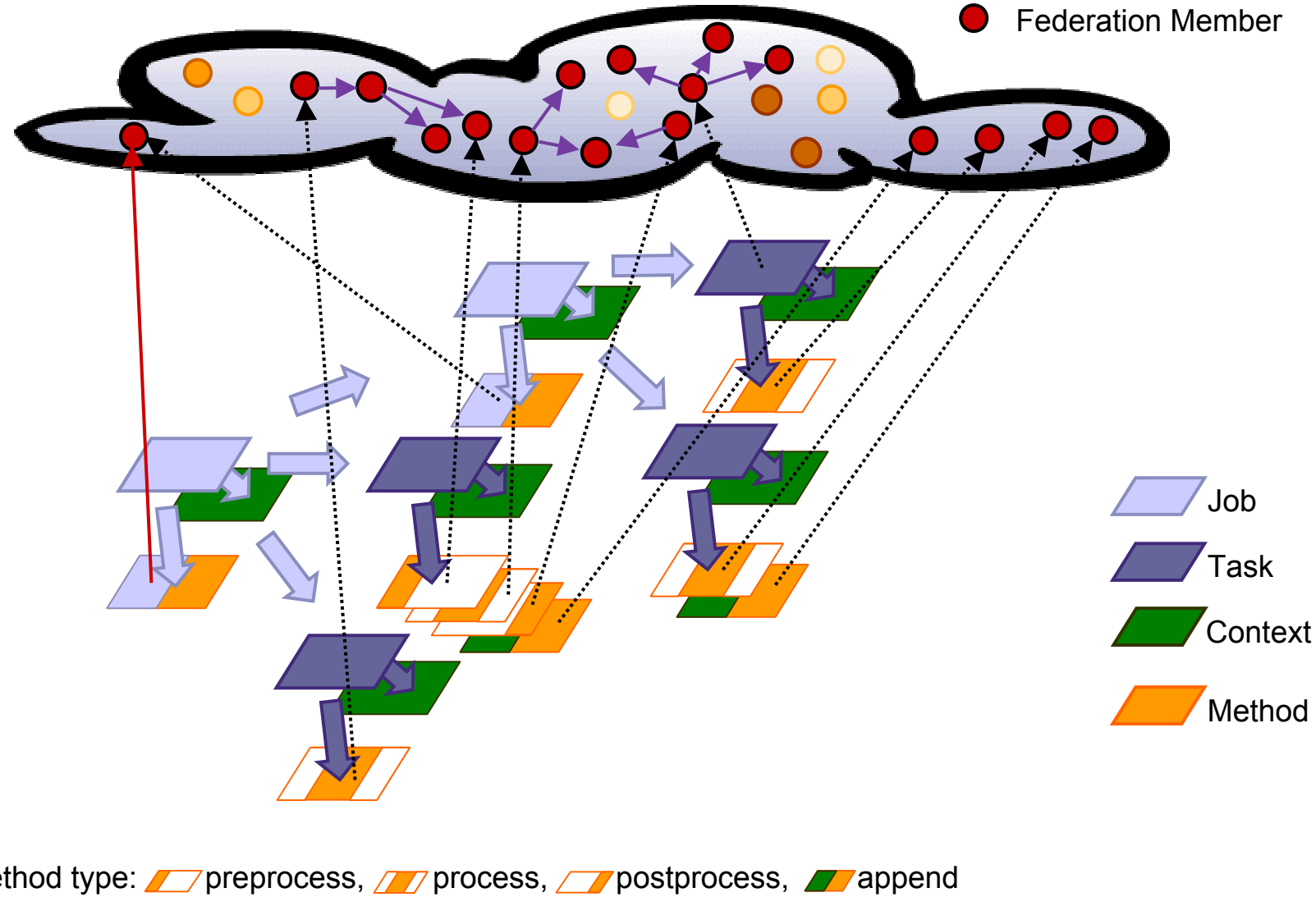
Job Task Service Context Service Method

Service Binding

Job as a Megaapplication



Federation of Services as a Job Runtime Environment



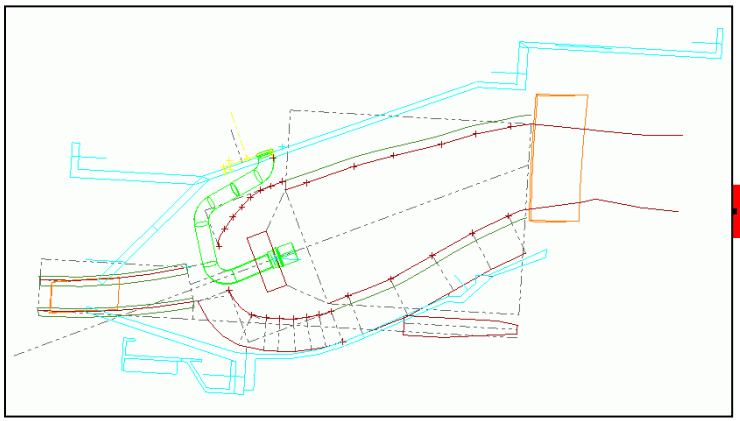
Nozzle Combustor CAD/IO B2B

TEXAS TECH UNIVERSITY Computer Science

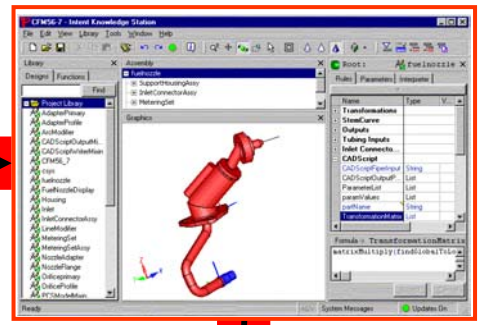
(UG)

(ProE) Parker Aerospace

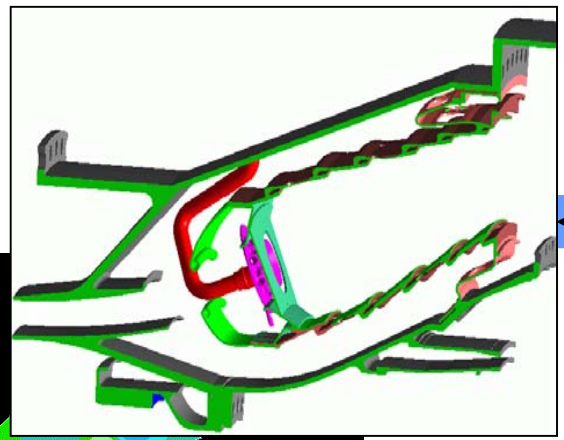
1. Update combustor PCS

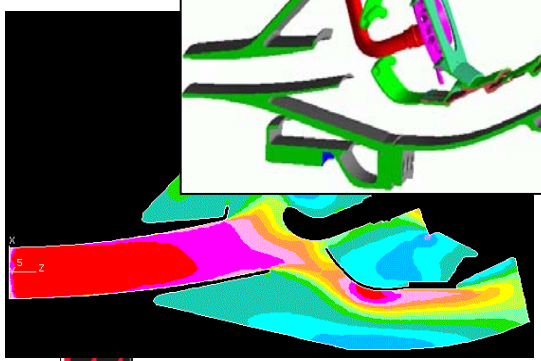


2. Request for nozzle validation



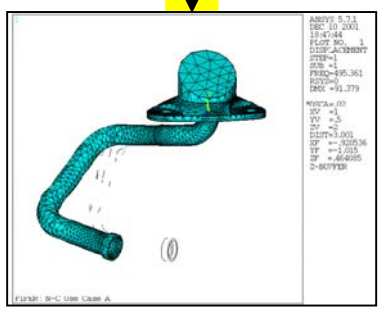
5. Perform CFD blow analysis



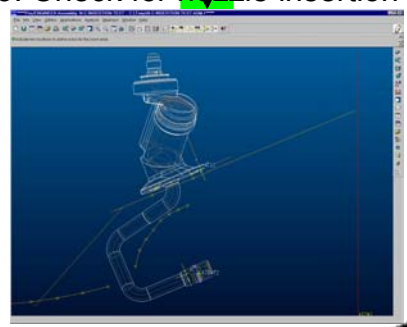


(Blow Analysis)

4. Perform modal analysis



3. Check for nozzle insertion



Question

What does it mean to be a service?

Answer

A service is an act of requesting a **`service(Exertion)`** operation from a service provider.

Service Requestor

`service(exertion)`

Service Provider

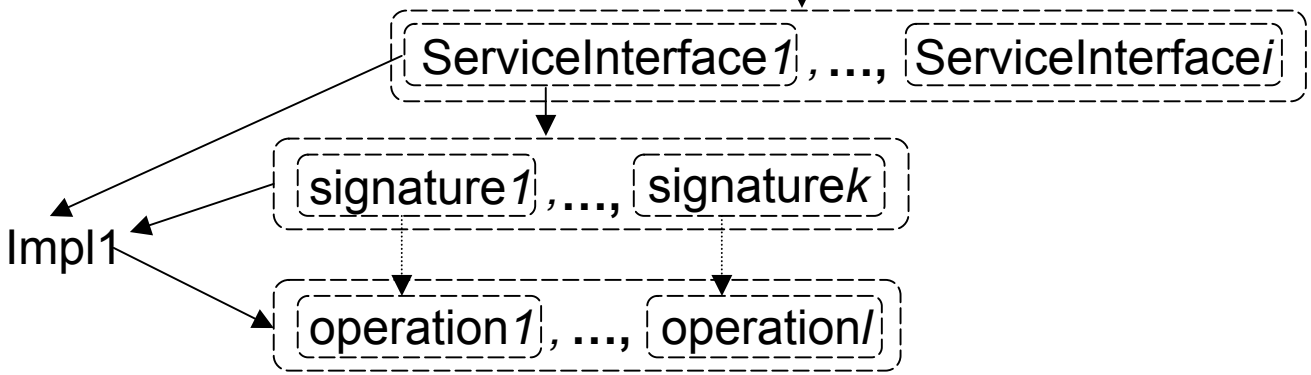
If accepted

then

`exertion.exert()`

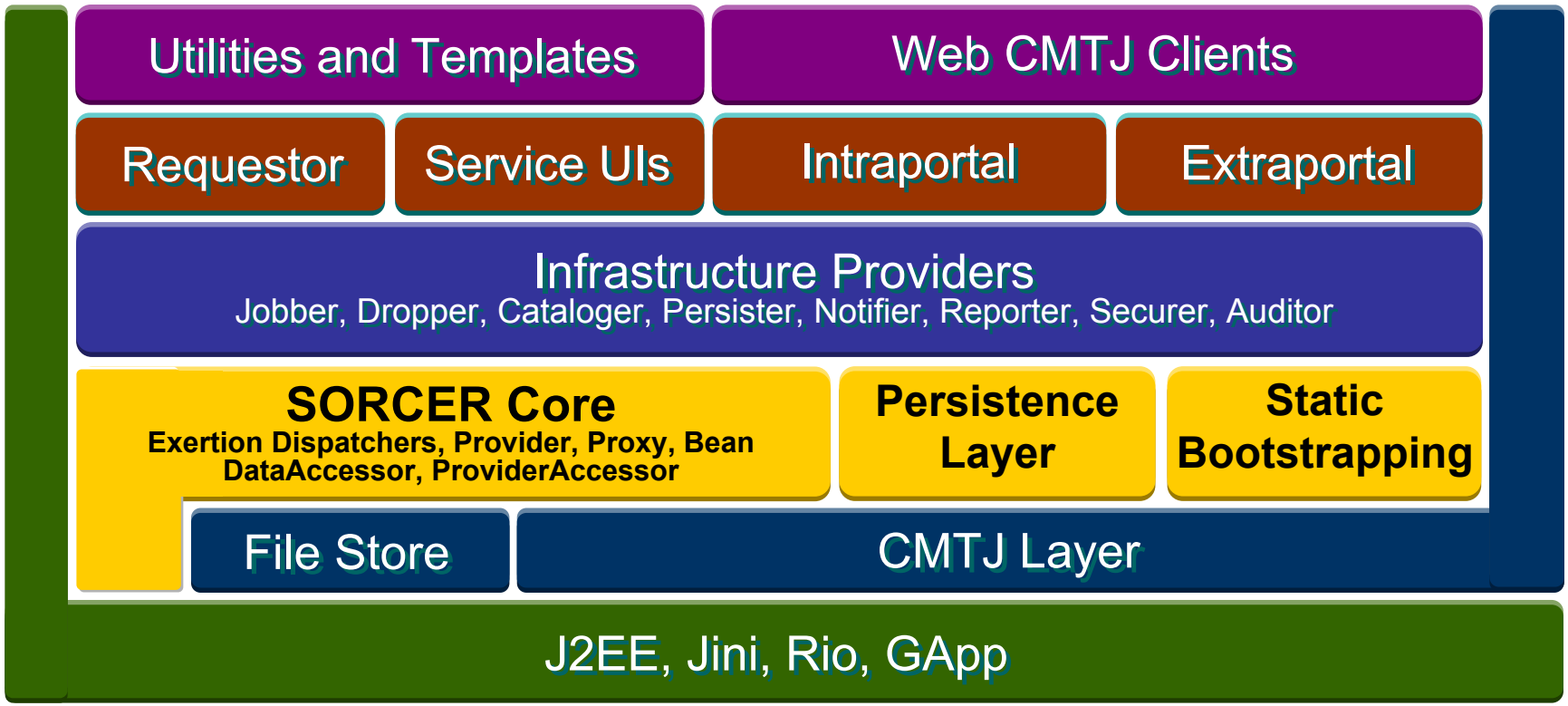
else

forward to a relevant service provider



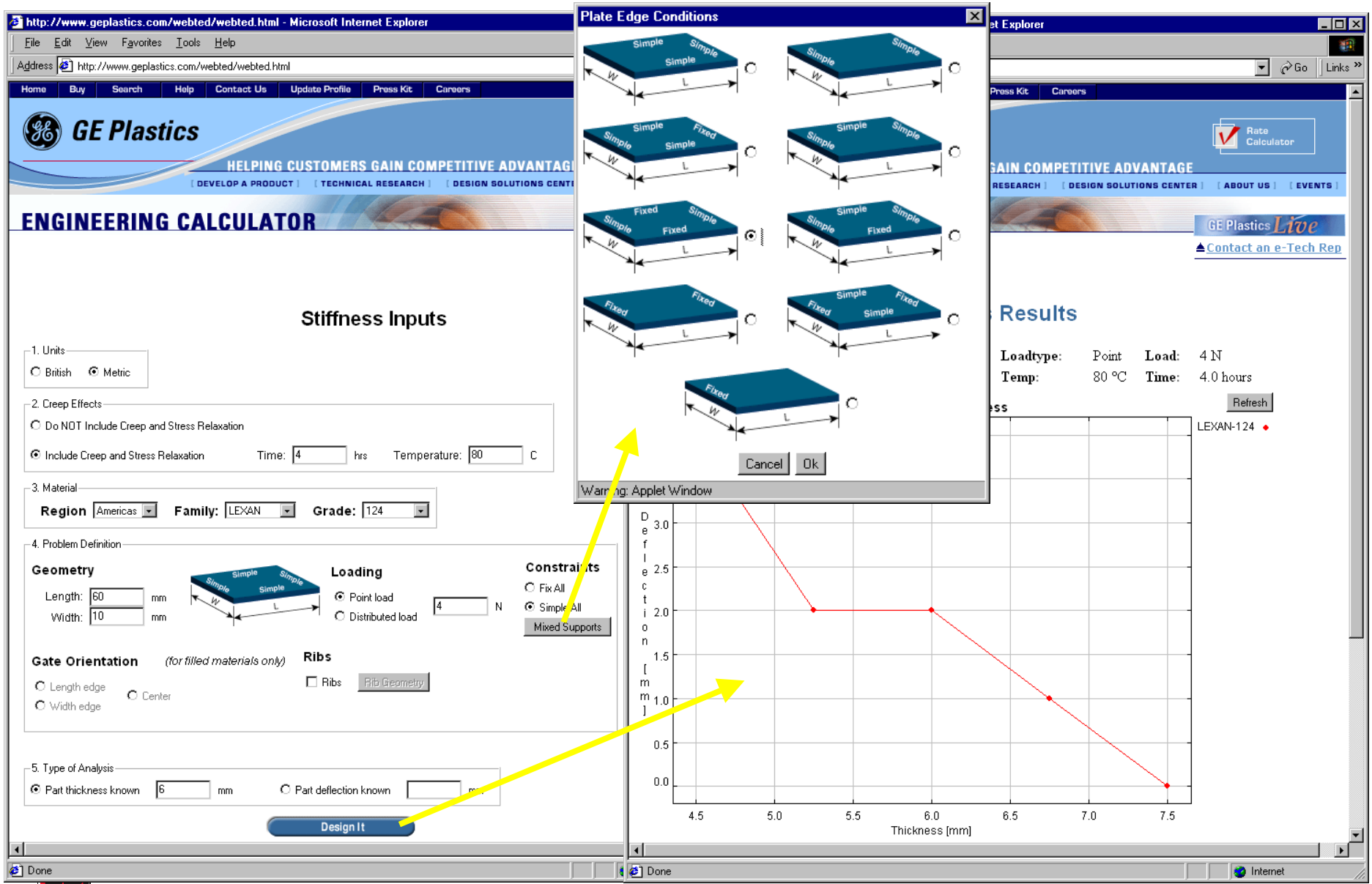
SORCER Functional Architecture

TEXAS TECH UNIVERSITY Computer Science



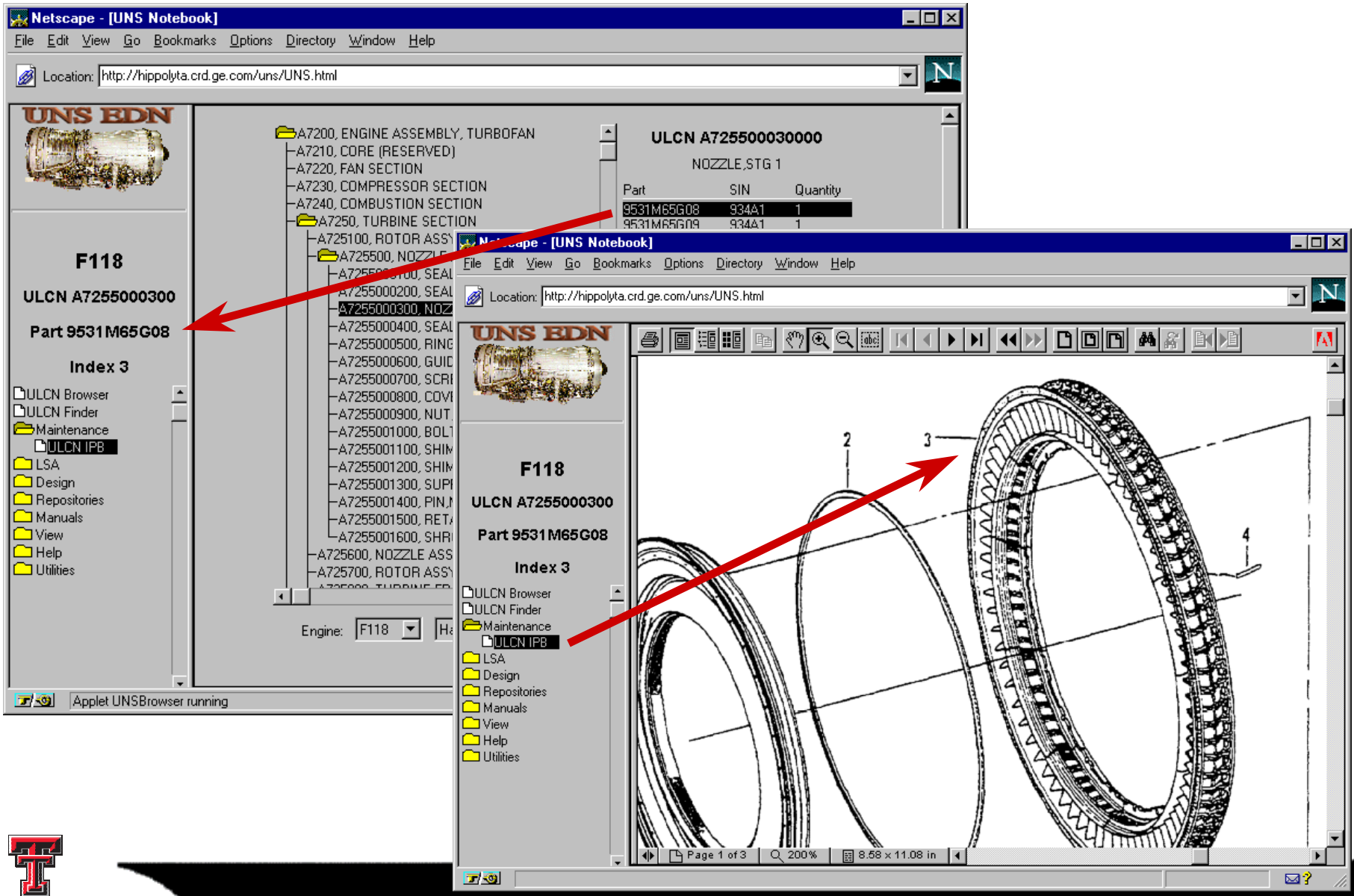
GEP Engineering Calculator

TEXAS TECH UNIVERSITY
Computer Science

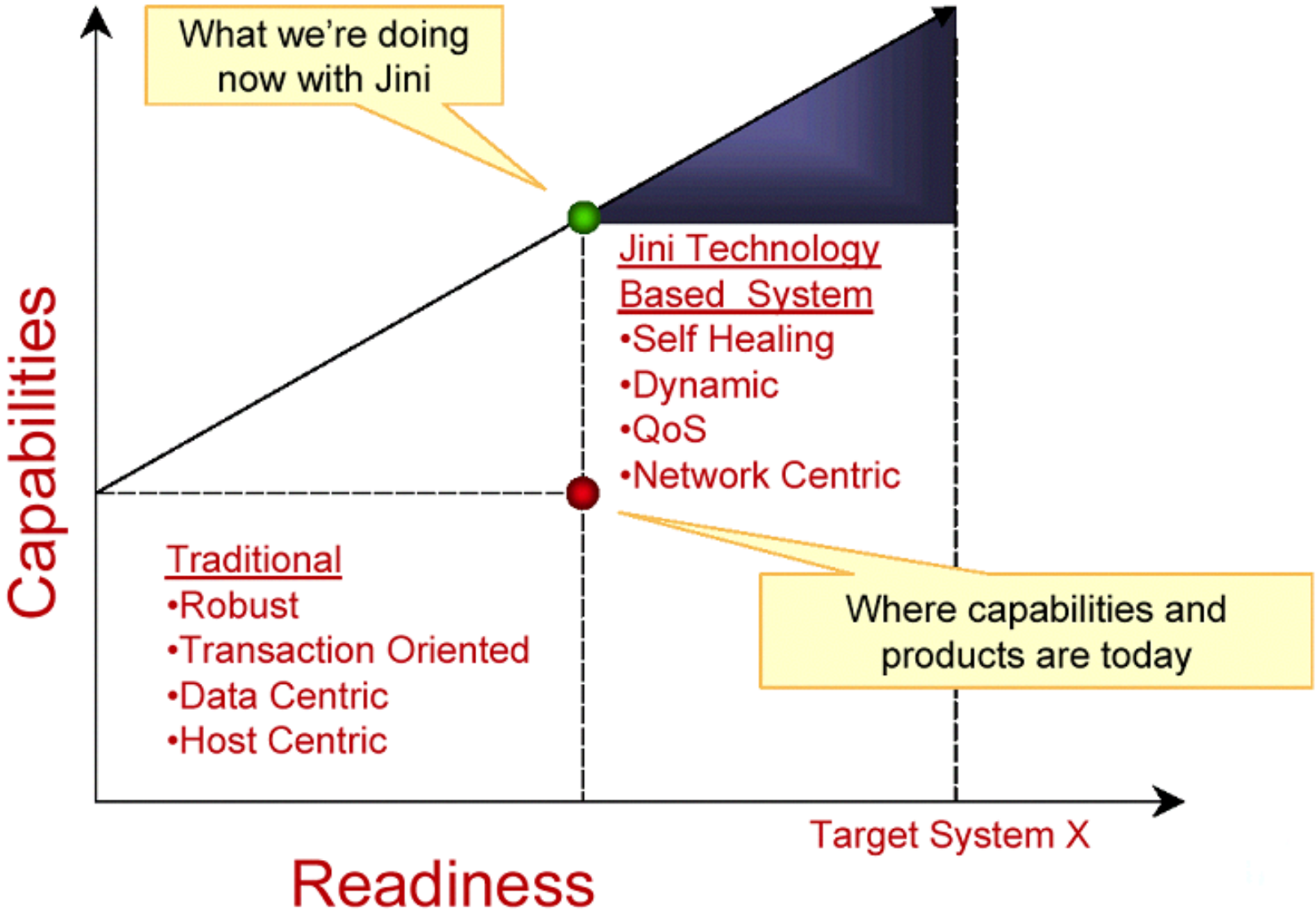


F118 Logistics
Electronic Design Notebook

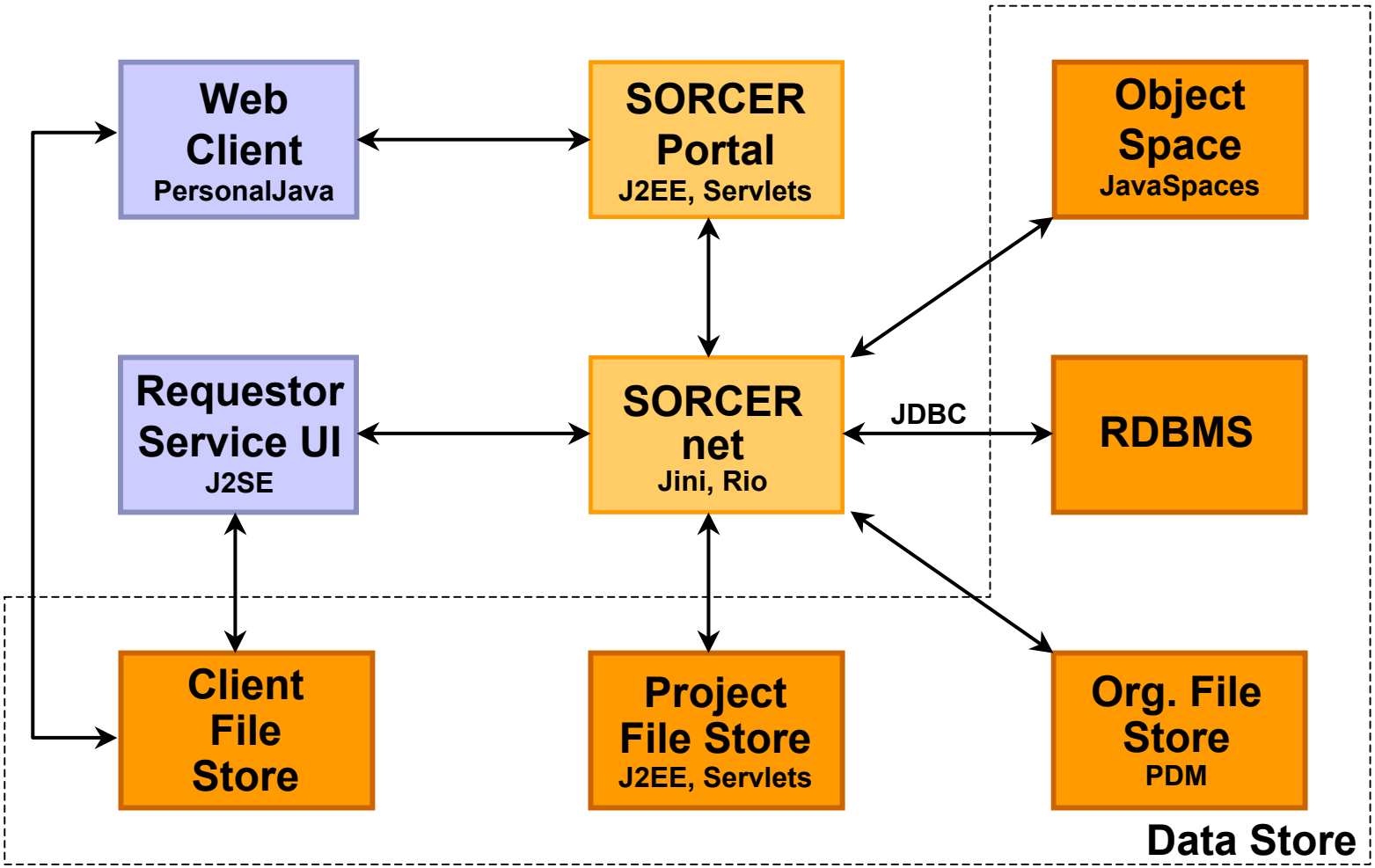
TEXAS TECH UNIVERSITY
Computer Science



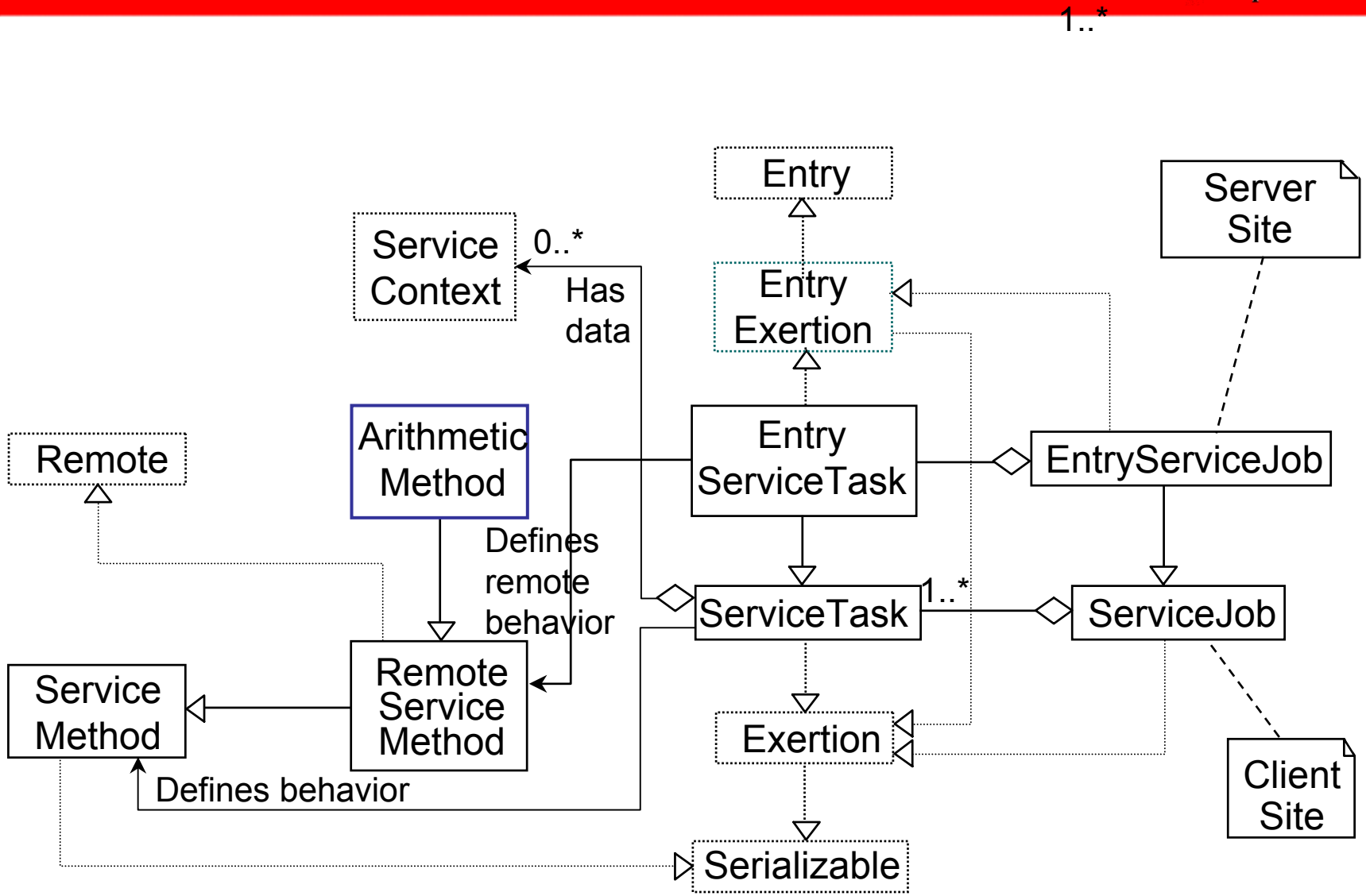
Dynamic Capability Trend



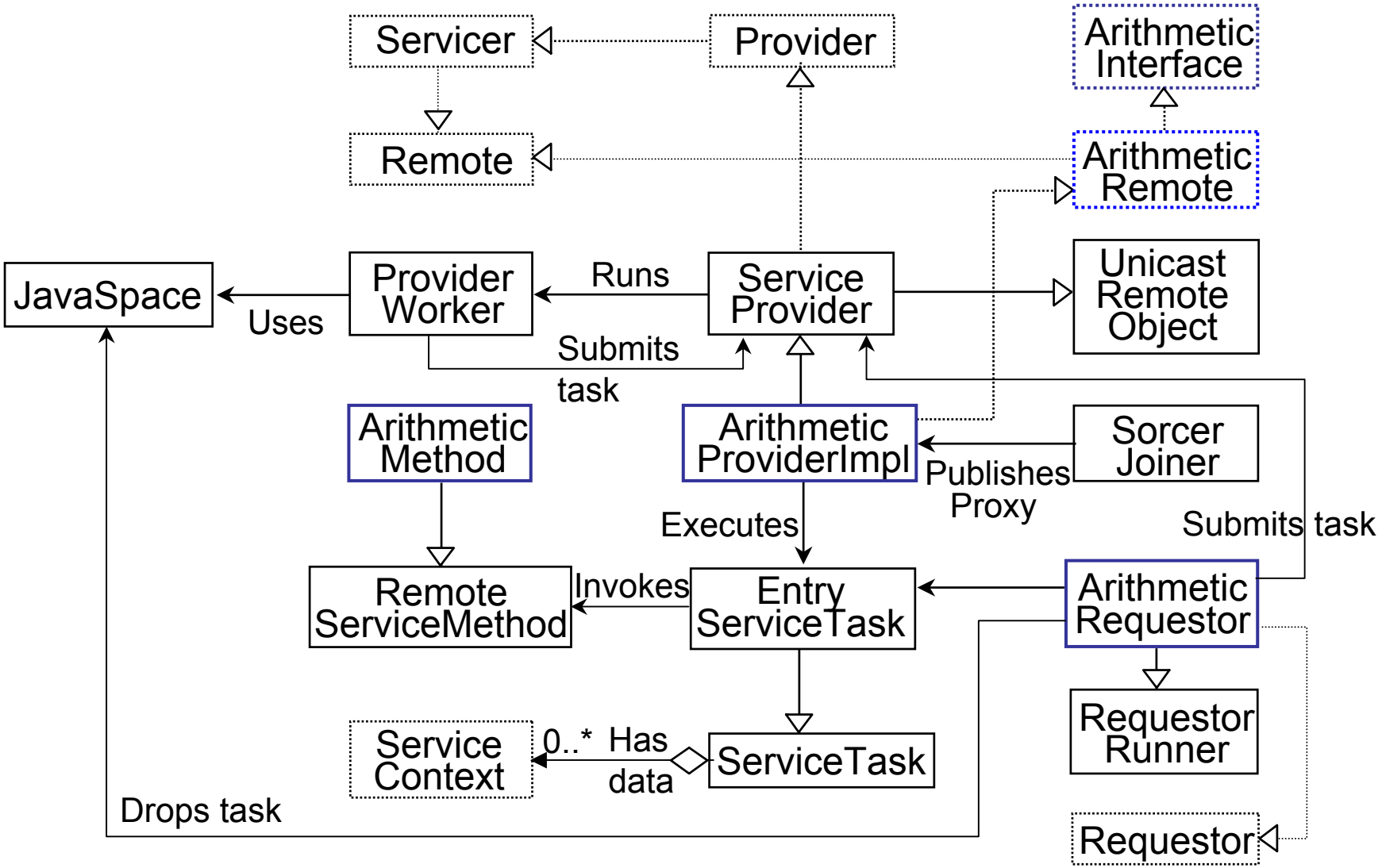
SORCER Organizational Architecture



Context/Method/Task/Job



Task Execution

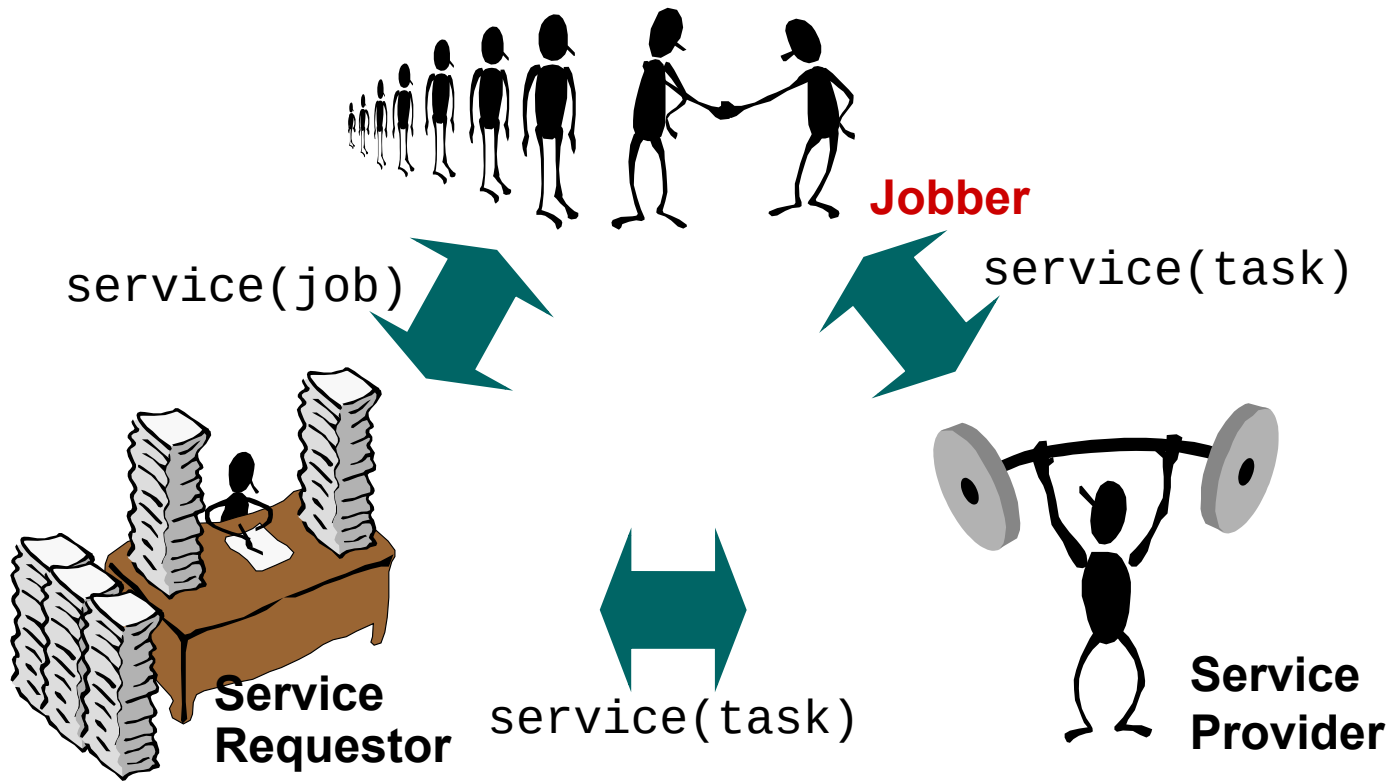


Question

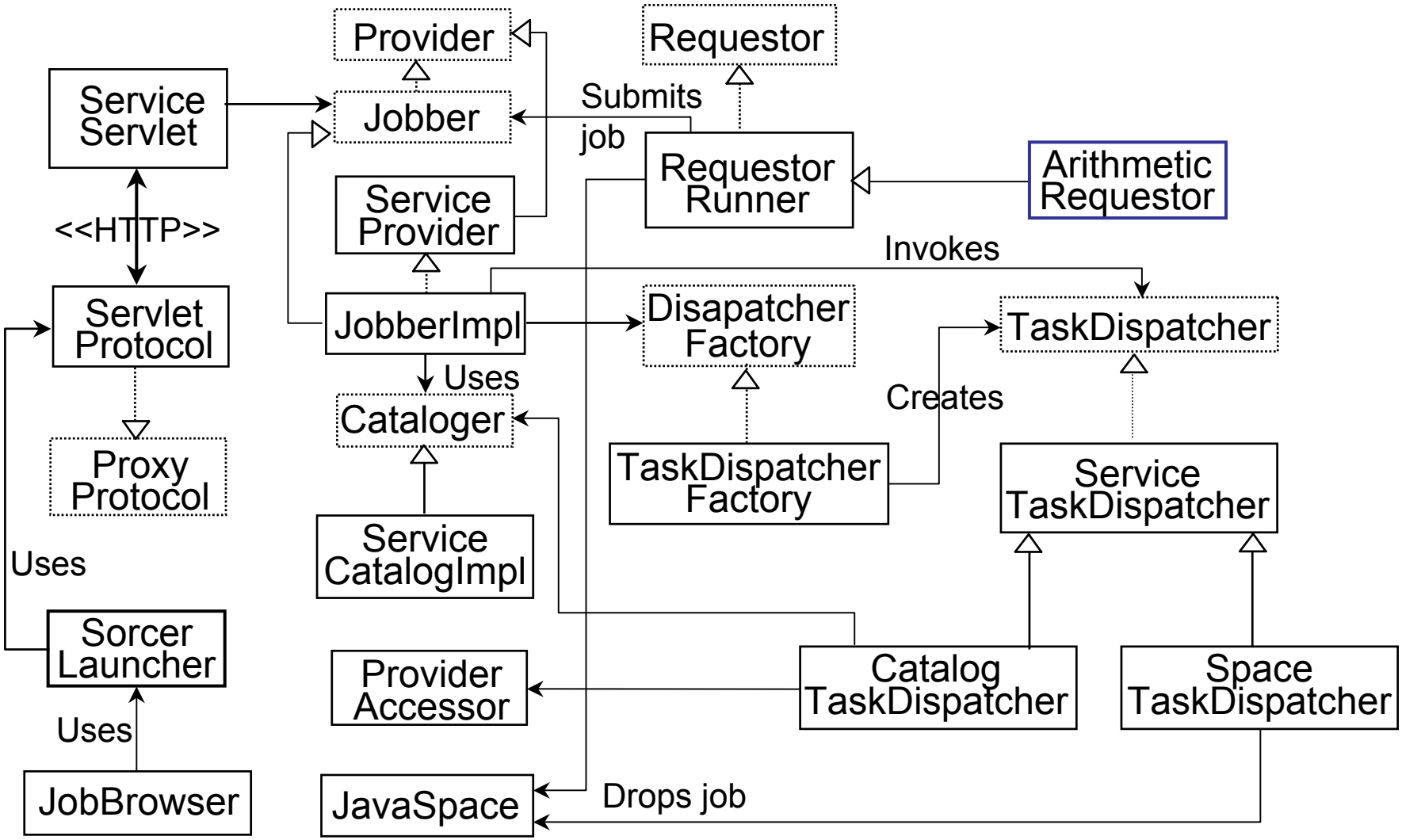
Does SORCER use service brokers?

Answer

A SORCER service broker is called a jobber.



Job Execution



Service access: Direct, Catalog, Space

Jini™ Network Technology

Service Beans

Rio

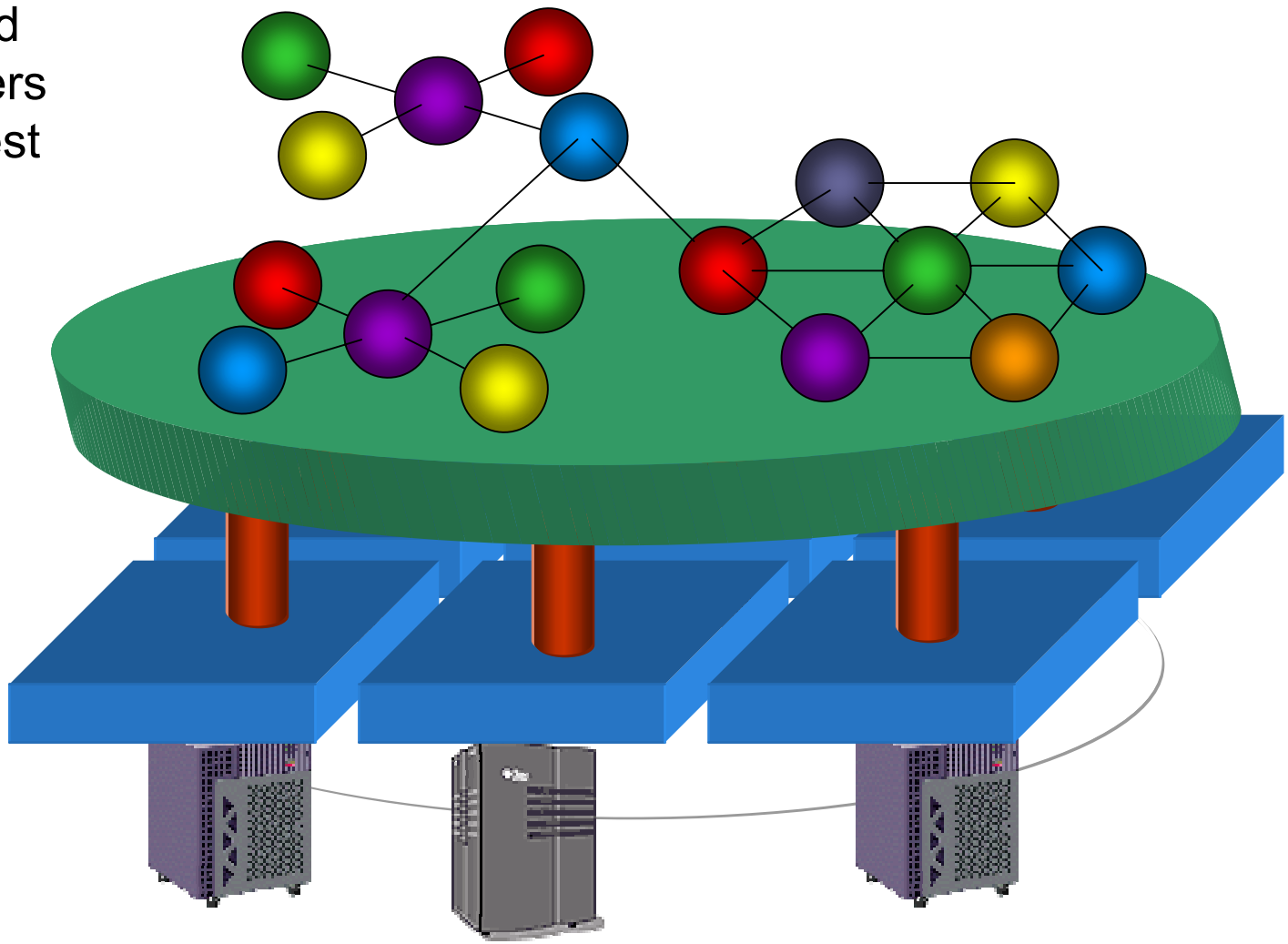
- Jini™ technology-based Service Beans ("JSBs") are the fundamental domain specific computational entities on the network
- Are provisionable based on their QoS attribute
- Jini technology-based Service Beans are instantiated by Cybernodes
  - Cybernodes run on computational resources
  - Cybernodes can contain multiple service beans

Service Delivery Network Grid

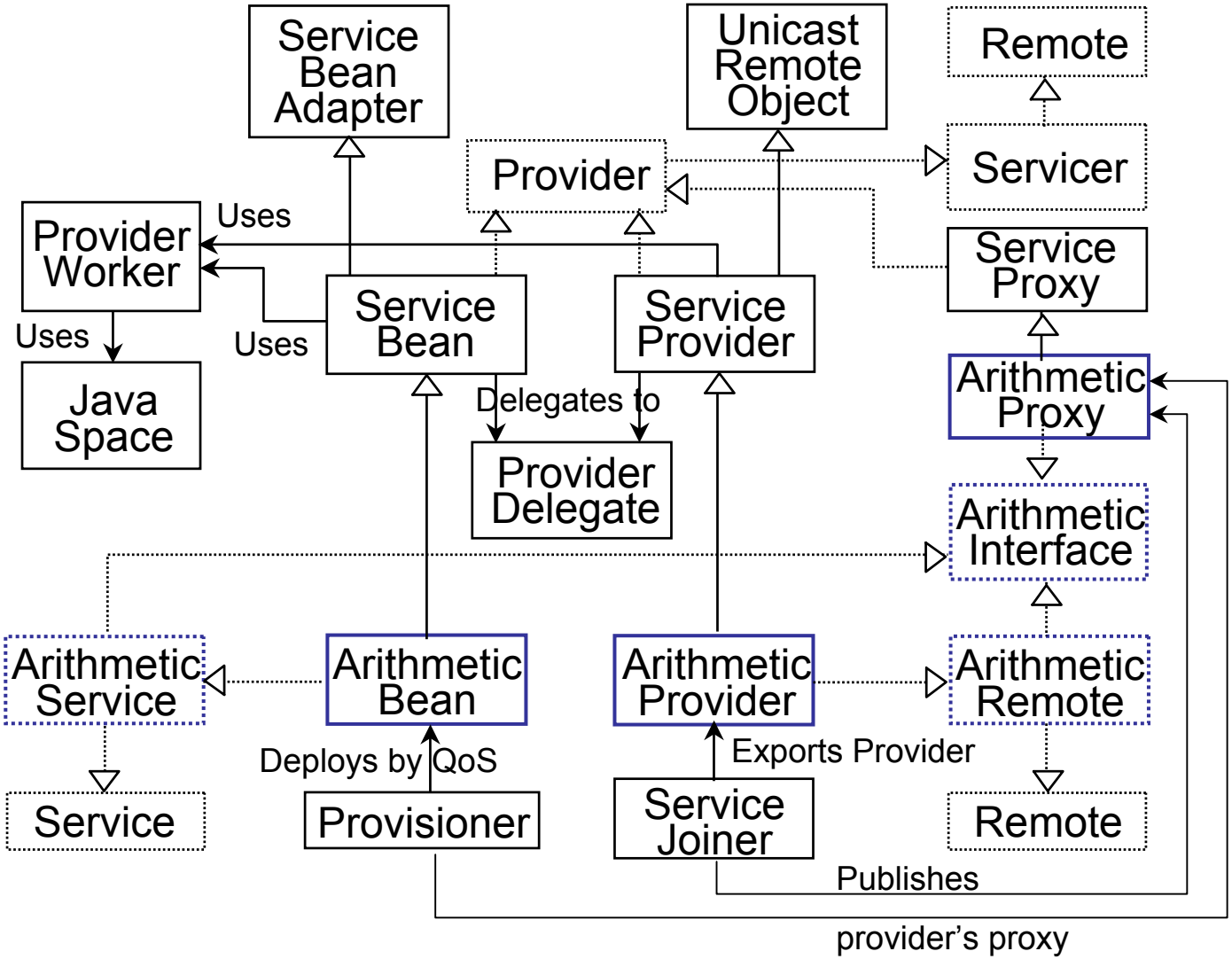
Self-organized service providers allocated to best resource

Dynamic QoS provisioning grid

Compute resource management grid



SORCER Smart Proxy
BL on Client and Server



Provider Bootstrapping

Bootstrapping Type	Server Type	NDS	Technology
java SorcerJoiner -sProviderClass	RMI server (JRMP/IIOP)	JNDI/RMI Reg JNDI/LDAP	RMI/CORBA
java SorcerJoiner -pProviderClass	Service provider (Jini)	LUS	Jini
java SorcerJoiner -pProviderClass:ProxyClass	Service provider with smart proxy (Jini)	LUS	Jini
Provisioning (Rio)	JSB	LUS	Rio/Jini

Mobile Code

- SORCER Code Mobility has many forms
  1. Proxies
  2. Exertions
  3. Task Methods
  4. Agents
  5. Service Beans (JSBs)
  6. Service UIs

SORCER Runtime Environment

Domain specific:

- Providers
- Requestors
- ServiceUIs

Infrastructure:

- Jobbers
- Droppers
- Catalogers
- Persisters
- Notifiers
- Service UIs

SORCER

- Websters
- Cybernodes
- Provisioners
- Lincolns

Rio

Web Server/App Server

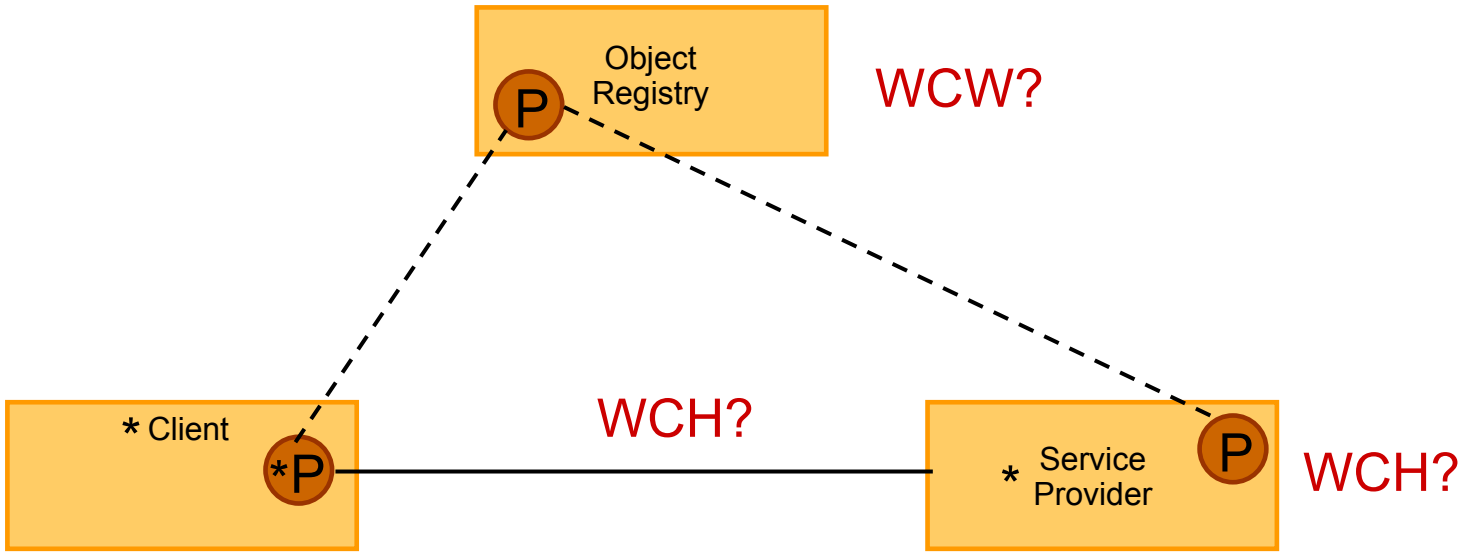
Summary (CNb)3

SORCER's C^3

- Service Centricity
  - everything is a service, each represented as an object on the network identified by type
- Network Centricity
  - services discover each other
  - the service is the network (N-1, 1-1, 1-N, S2S)
- Web Centricity
  - Interportals/Extraportals to services with thin web clients (applets/servlets)

Summary (CNb)3

SORCER's N^3



- Co-location Neutrality
- Protocol Neutrality
- Implementation Neutrality

* Business logic
WCH/W – Who cares how/where?

Summary (CNb)3

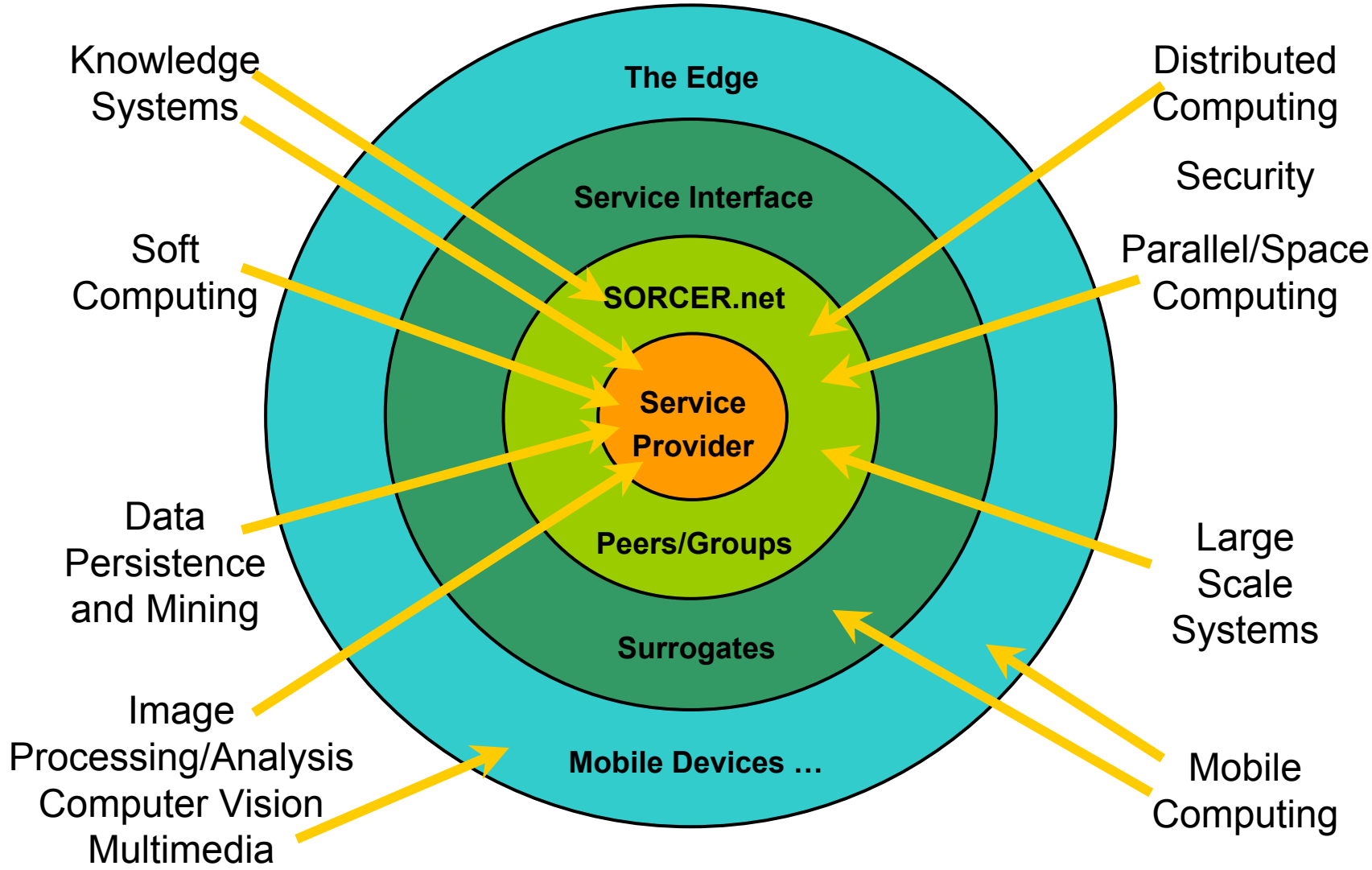
Architecture Qualities b^3

- Accessibility
  - Web Centricity, standalone clients, agents
- Adaptability
  - Mobile Code
- Scalability
  - Network Centricity, Federated Services

Everyone Can Contribute

TEXAS TECH UNIVERSITY Computer Science

Pervasive SORCER.net



Conclusion

- Jini™ and Rio technologies enable federated S2S, platform independent, real world megaprogramming environments.
- A SORCER job is a distributed megaapplication executed in a federated S2S environment.

Q&A

Michael Sobolewski
sobol@cs.ttu.com