JavaOne℠

Sun's 2002 Worldwide Java Developer Conference℠

FIPER: The Federated S2S Environment

Michael Sobolewski
Senior Computer Scientist
GE CR&D
sobol@crd.ge.com

Overall Presentation Goal

Learn how to architect and build Service-to-Service (S2S) environments using RMI, Jini™, Rio, Java™ Servlet, and JDBC™ technologies

Learning Objectives

- As a result of this presentation, you will be able to:
  - Design S2S environments using RMI, Jini™ and Rio technologies
  - Recognize network-centric vs. server-centric frameworks
  - Understand the job/task/context/method paradigm
  - Recommend a distributed megaprogramming system

Speaker's Qualifications

- Dr. Mike Sobolewski is a senior computer scientist at GE Corporate Research and Development (CR&D)
- Published more than 70 publications in computer science and SW engineering
- OO technologist with 10 years of experience in C/C++, 15 years on Smalltalk, and 6 years using Java™ technology
- Architect and lead developer for more than 20 Java technology-based and Web-based projects at GE CR&D, recently for the FIPER project at GE CR&D

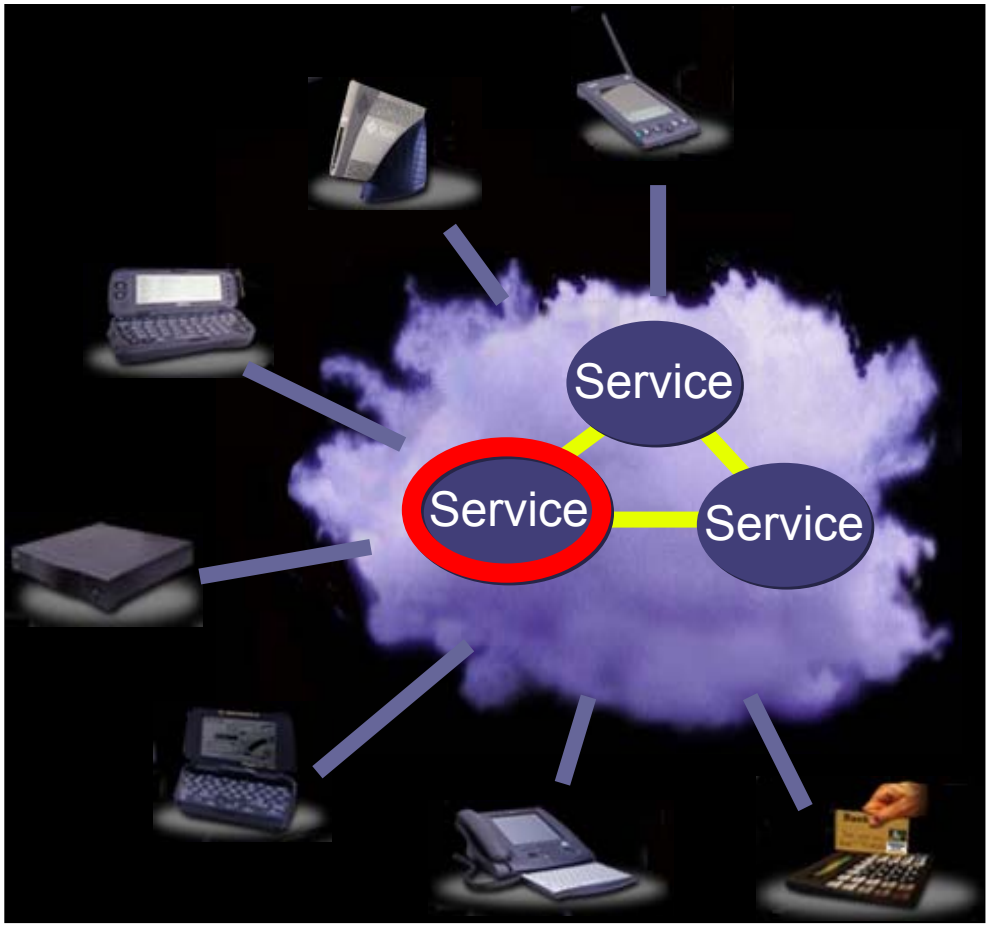
Distributed Megaprogramming

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

Presentation Agenda

- Distributed Megaprogramming
- Job/Task/Context/Method Paradigm
- FIPER S2S Environment
- FIPER Functional Architecture
- Design Issues (UML diagrams)
- Summary

Federated Intelligent Product EnviRonment (FIPER) Vision

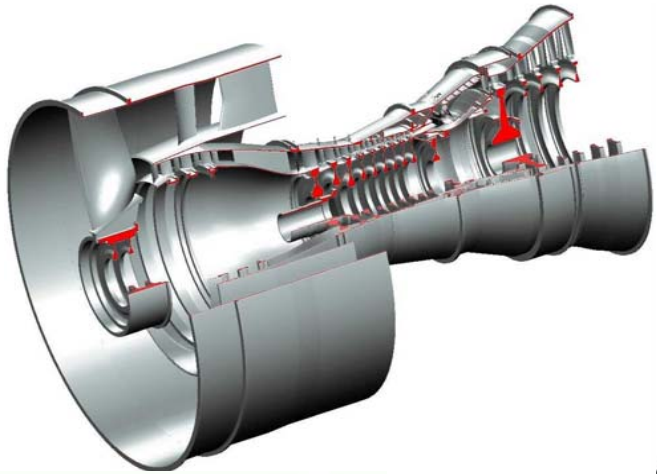


Federated S2S environment to ...

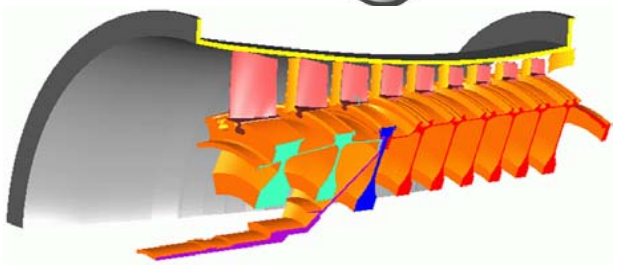
- Service — Build new services
- Service — Convert legacy apps to dynamic FIPER services (J2EE™ technology)
- Assemble FIPER services together (RMI, Jini, Rio technologies)
- Create modern clients accessing services

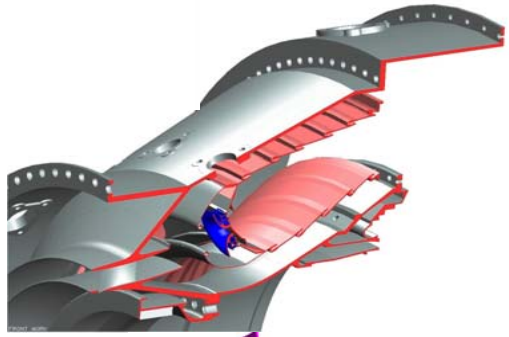
FIPER Megaprogramming Domain

System Design

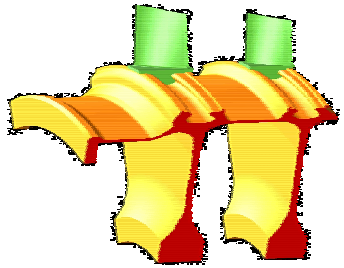


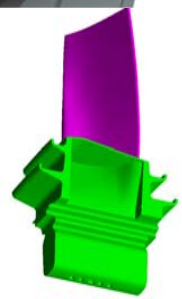
Subsystem Design





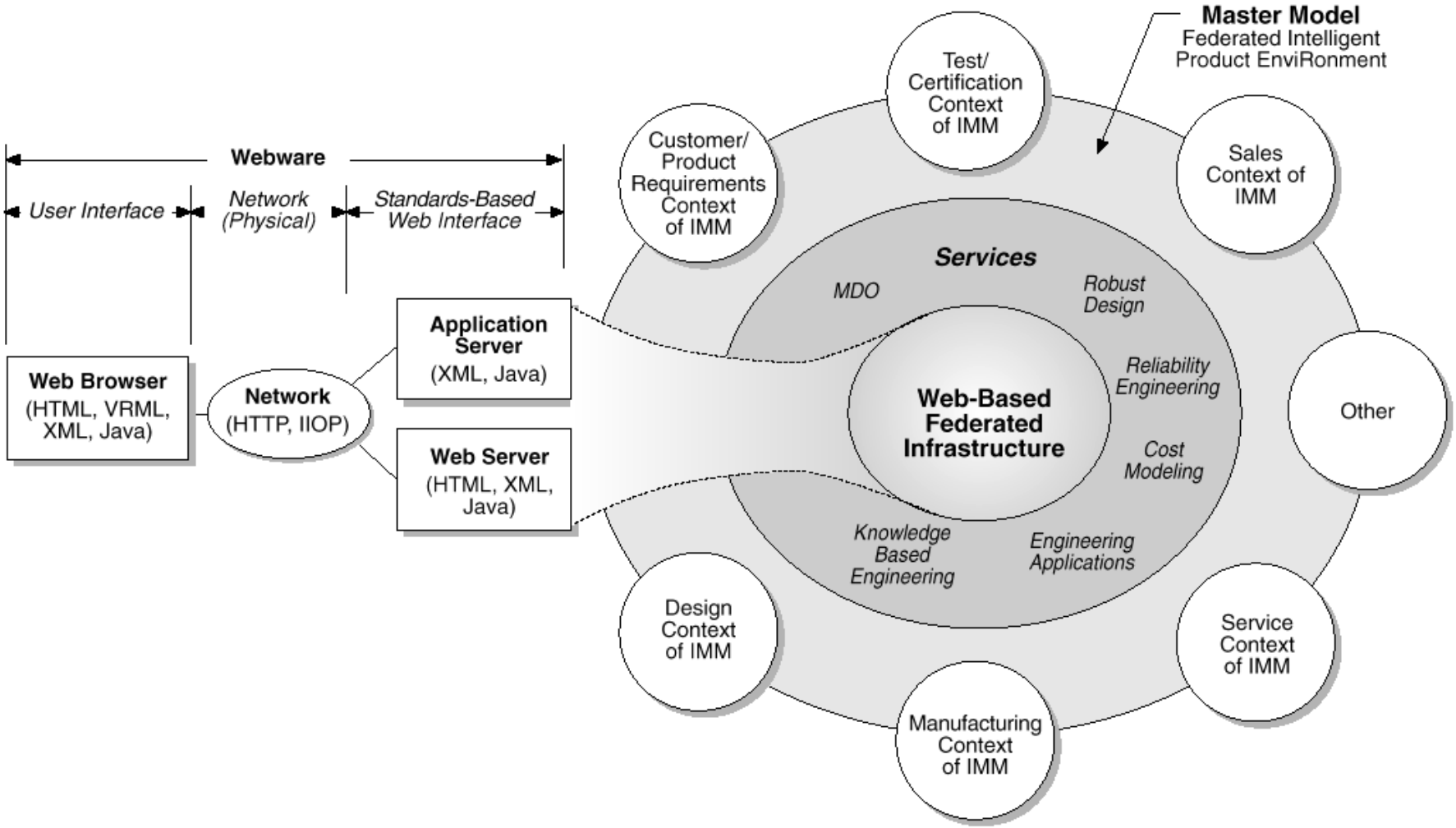
Component Design





JavaOne

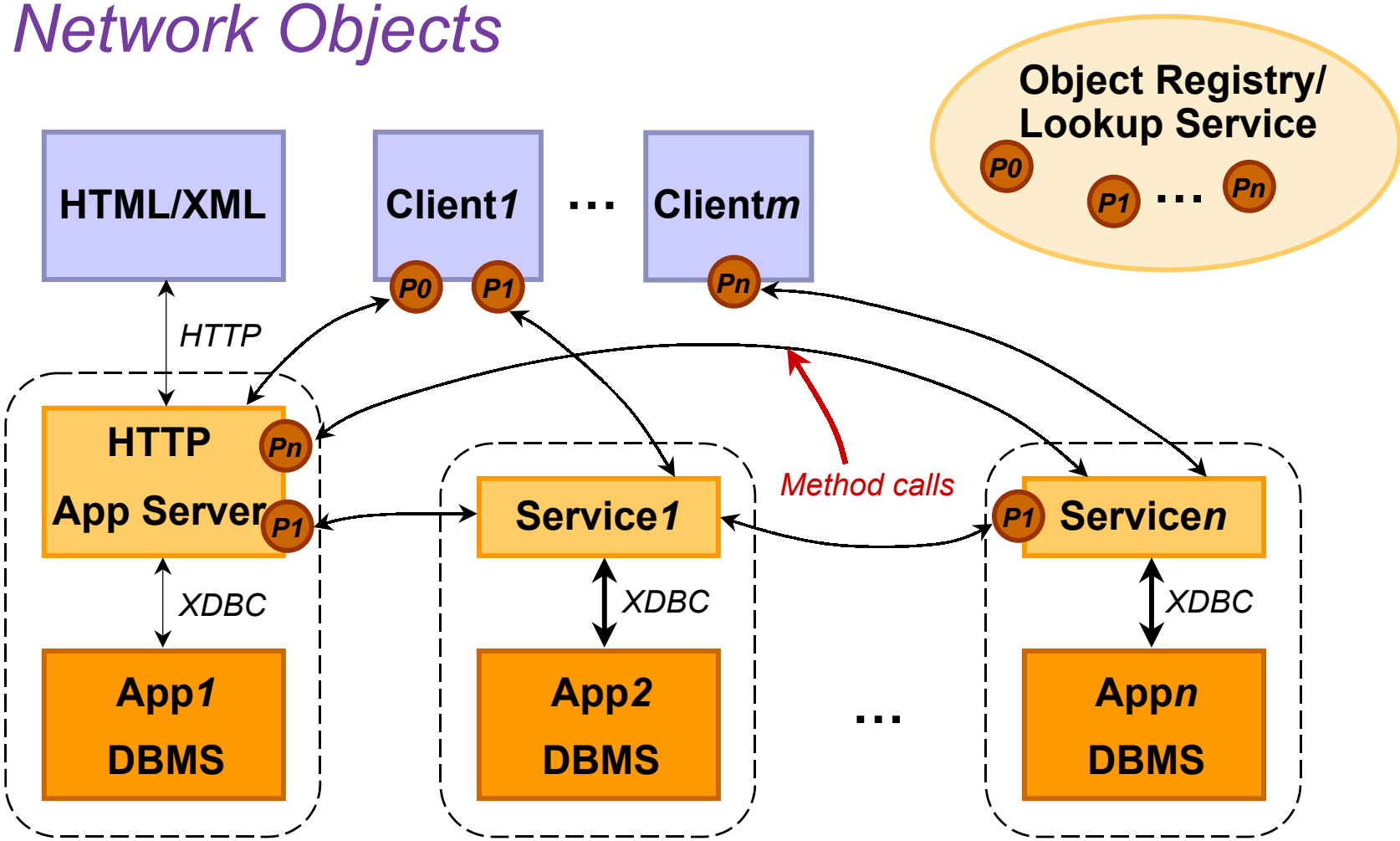
Everything Is on the Network
Everything on the Network Is a Service



Service Provider As a Network Object

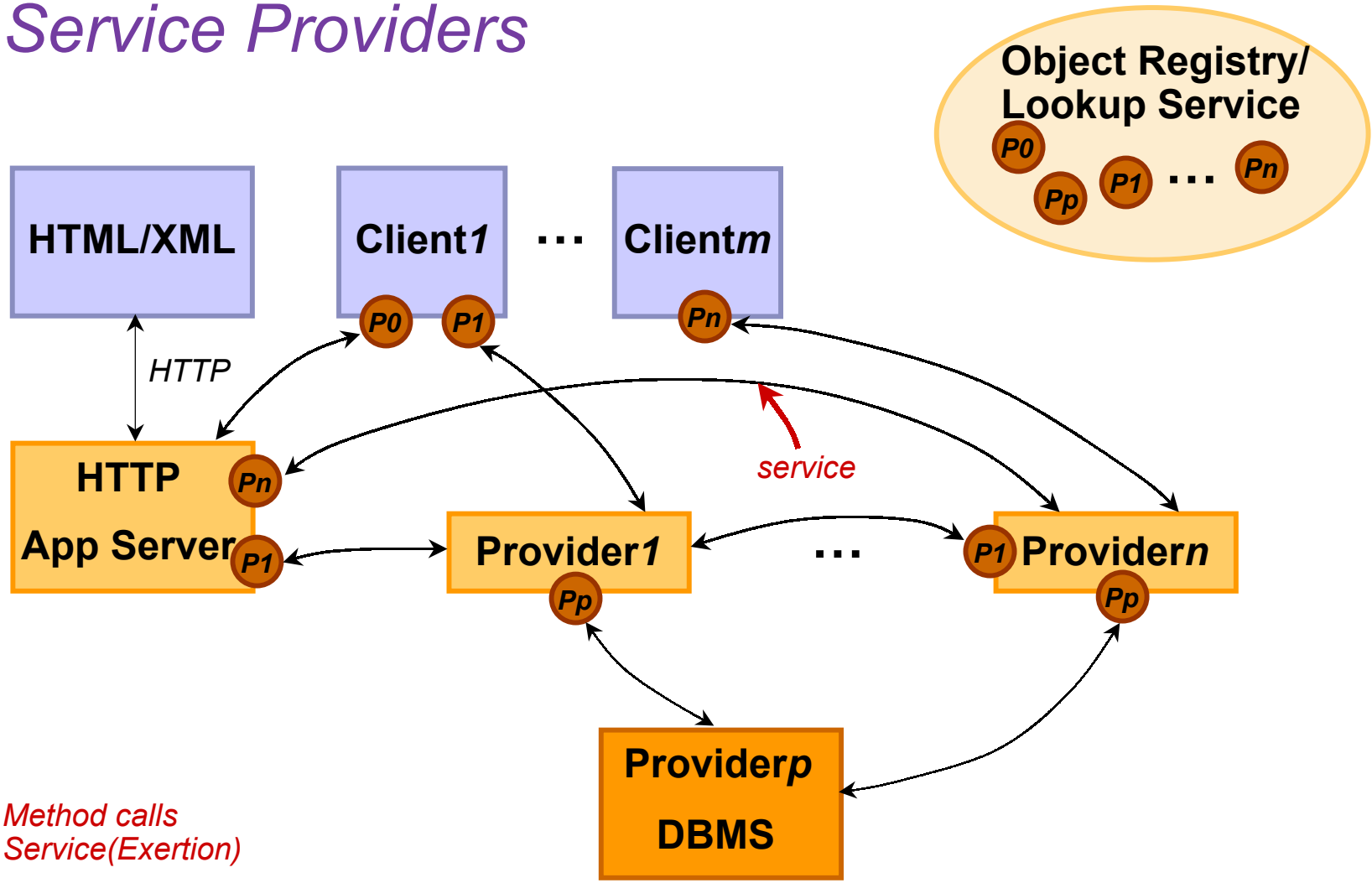
Service-Oriented Computing

Network Objects



Service-to-Service (S2S)

Service Providers



Applying OO Techniques to the Network

- 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 Java™ technology 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

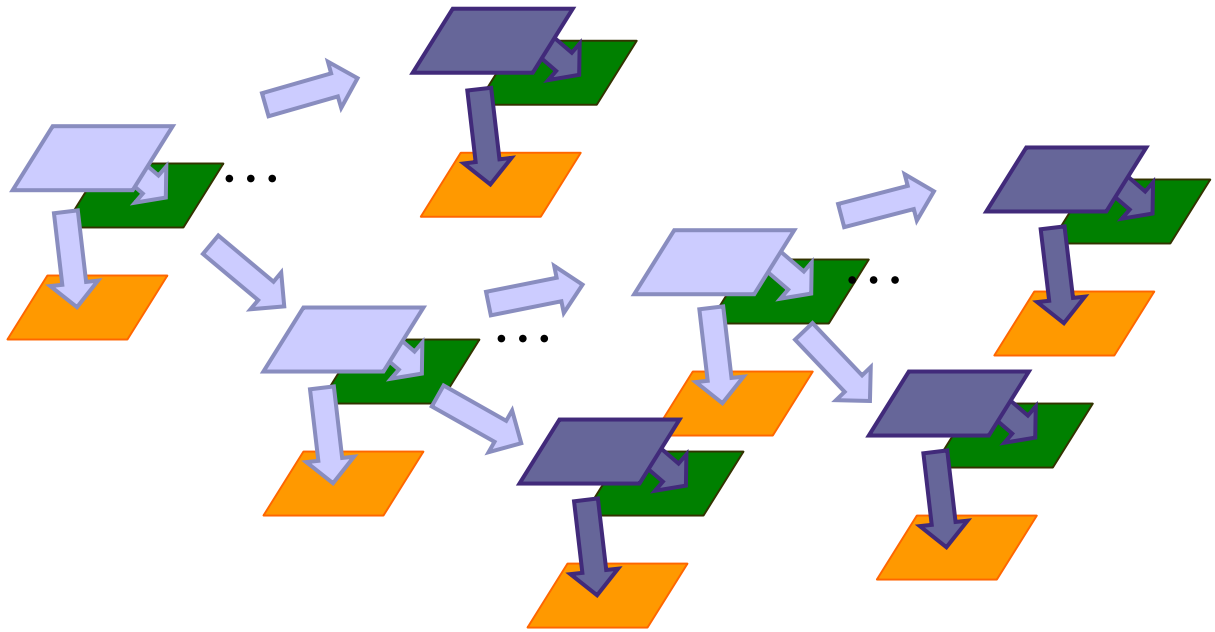
- All services implement this interface:

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

    // Monitoring methods
    ...

}
```

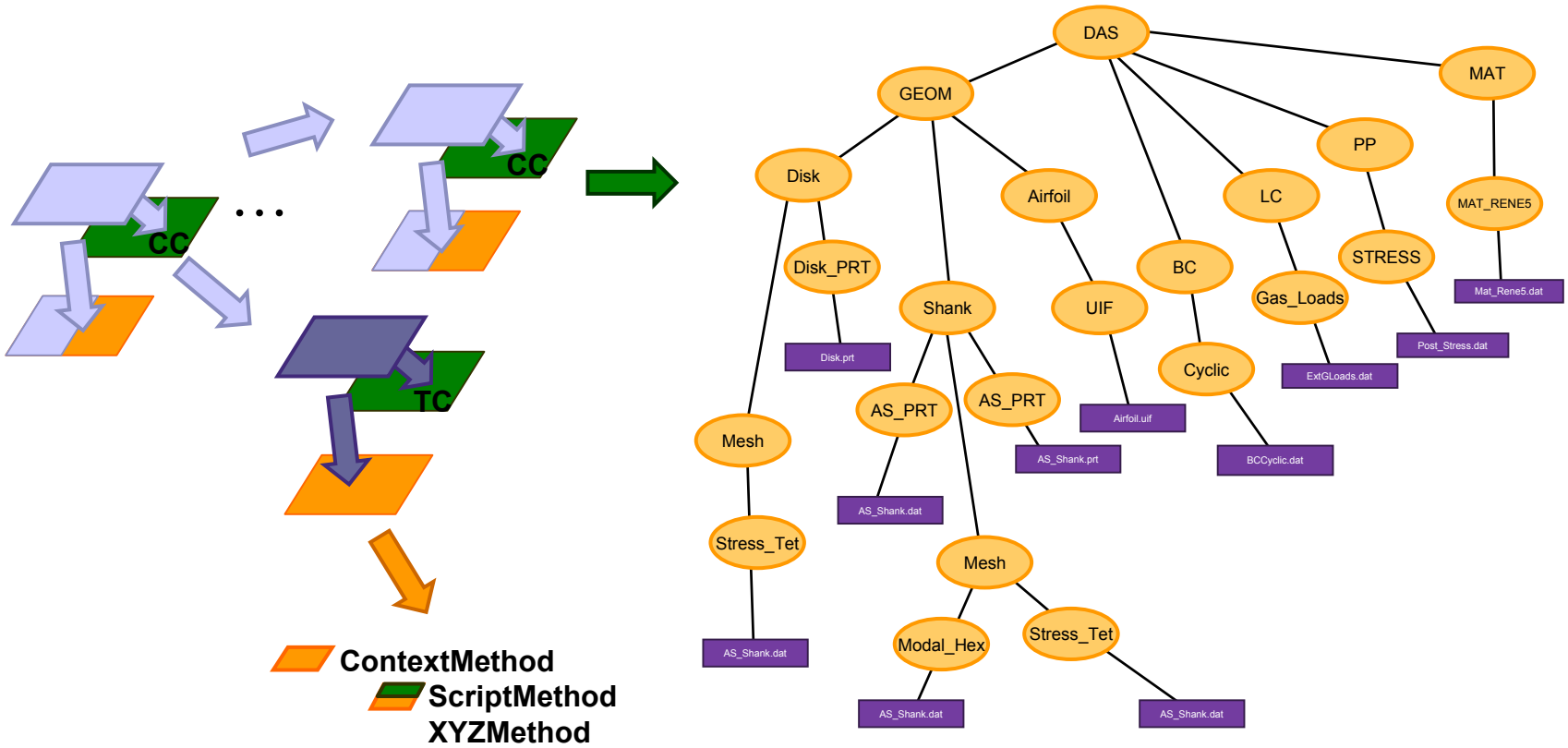
Job/Task/Context/Method Paradigm



Job Task Task Context Context Method

JavaOne

Contexts and Task Methods



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

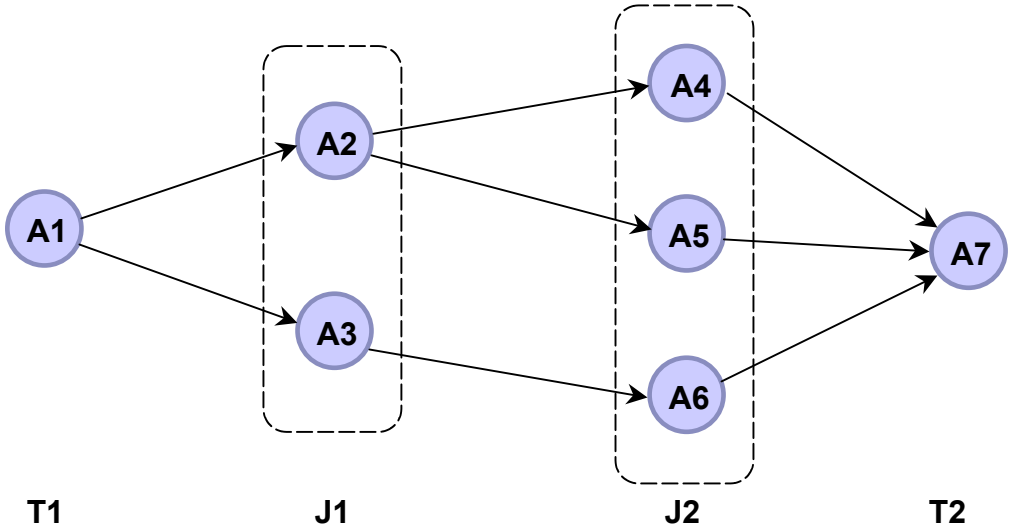
Method type: preprocess, process, postprocess, append

TC – Task Context, CC – Control Context

Job Task Task Context Context Method

JavaOne

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

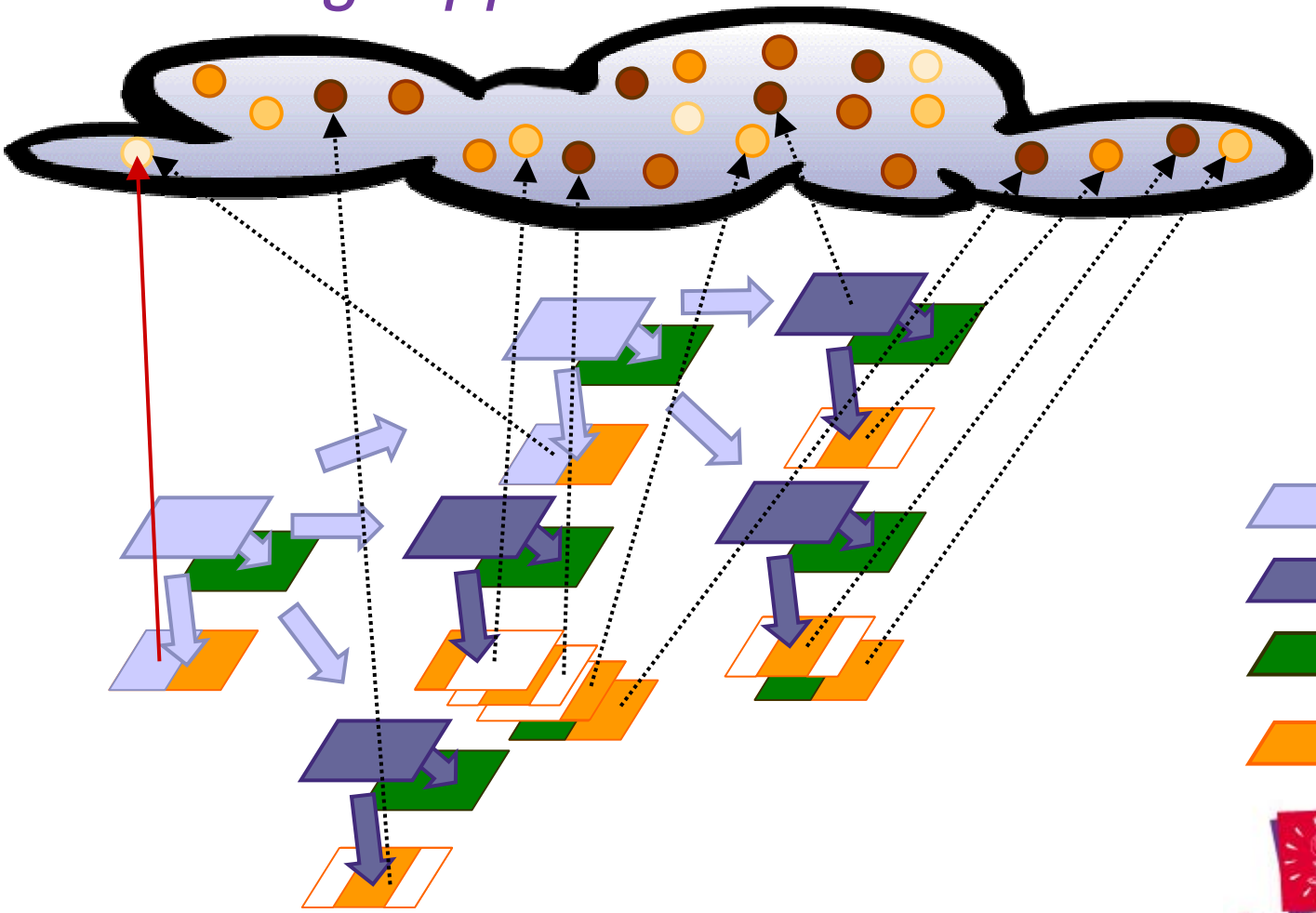
Bi-directional aggregation

Control strategy separated

Workflow defined implicitly

Service Binding

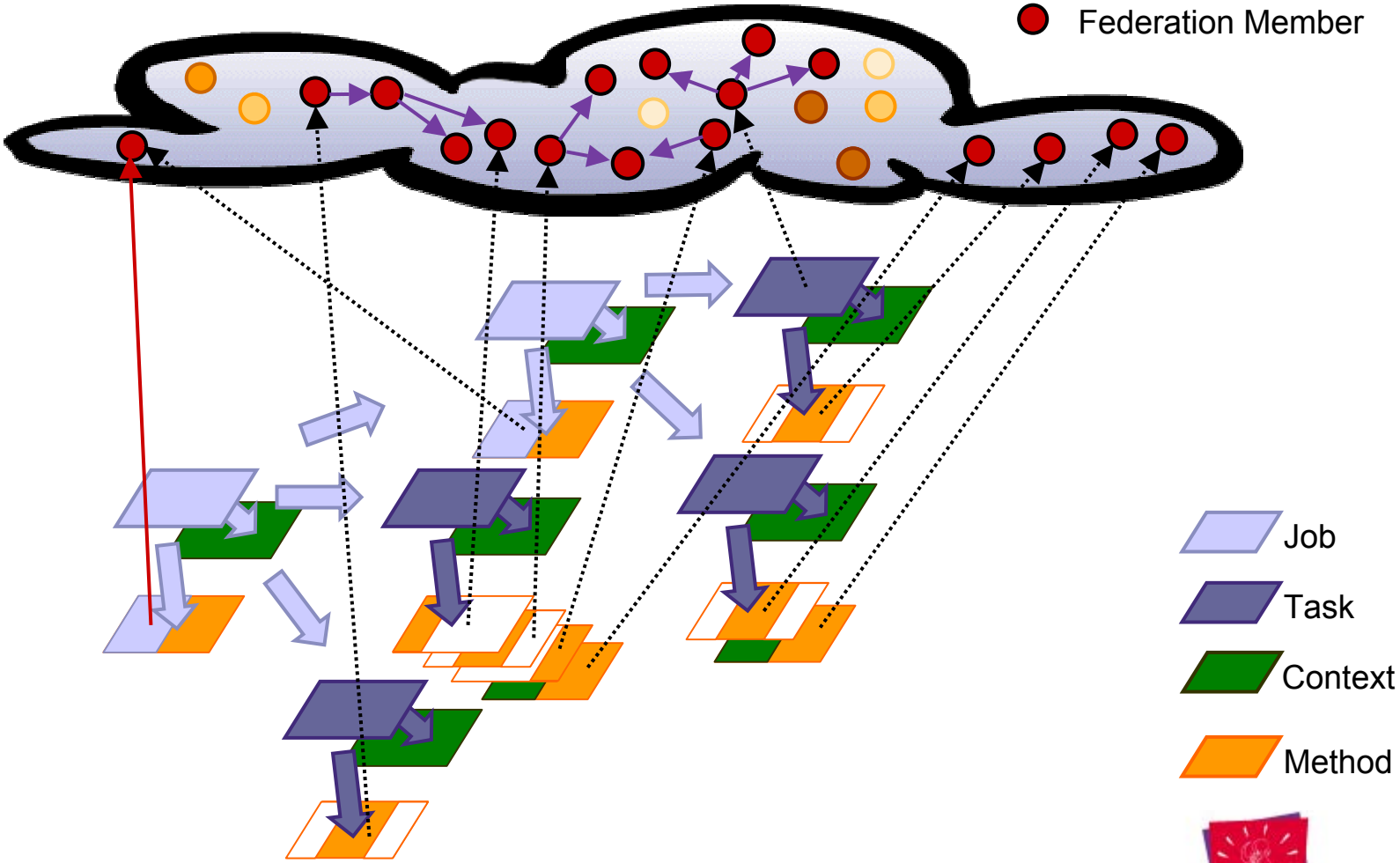
Job as a Megaapplication



Method type: preprocess, process, postprocess, append

JavaOne

Federation of Services as a Job Runtime Environment



Method type: preprocess, process, postprocess, append

JavaOne

Question

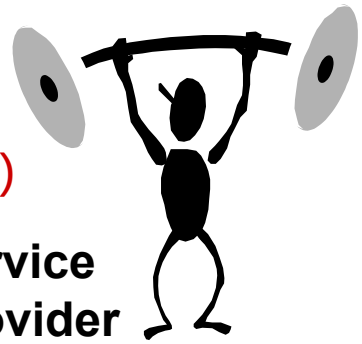
What does it mean to be a service?

Answer

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

Service Requestor

`service(exertion)`



Service Provider

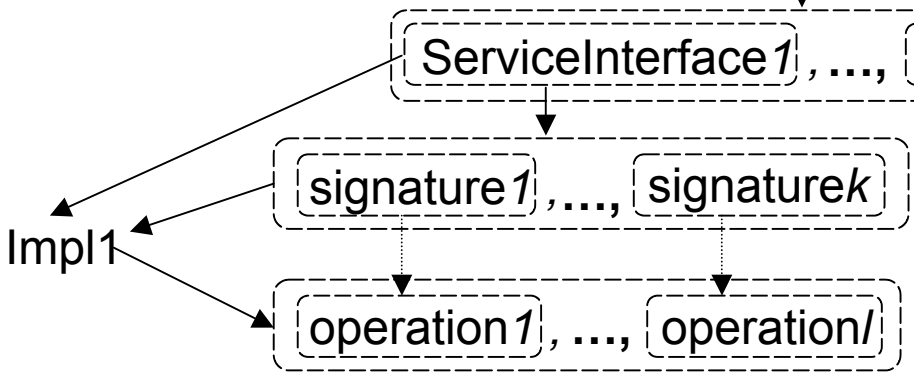
If accepted

then

`exertion.exert()`

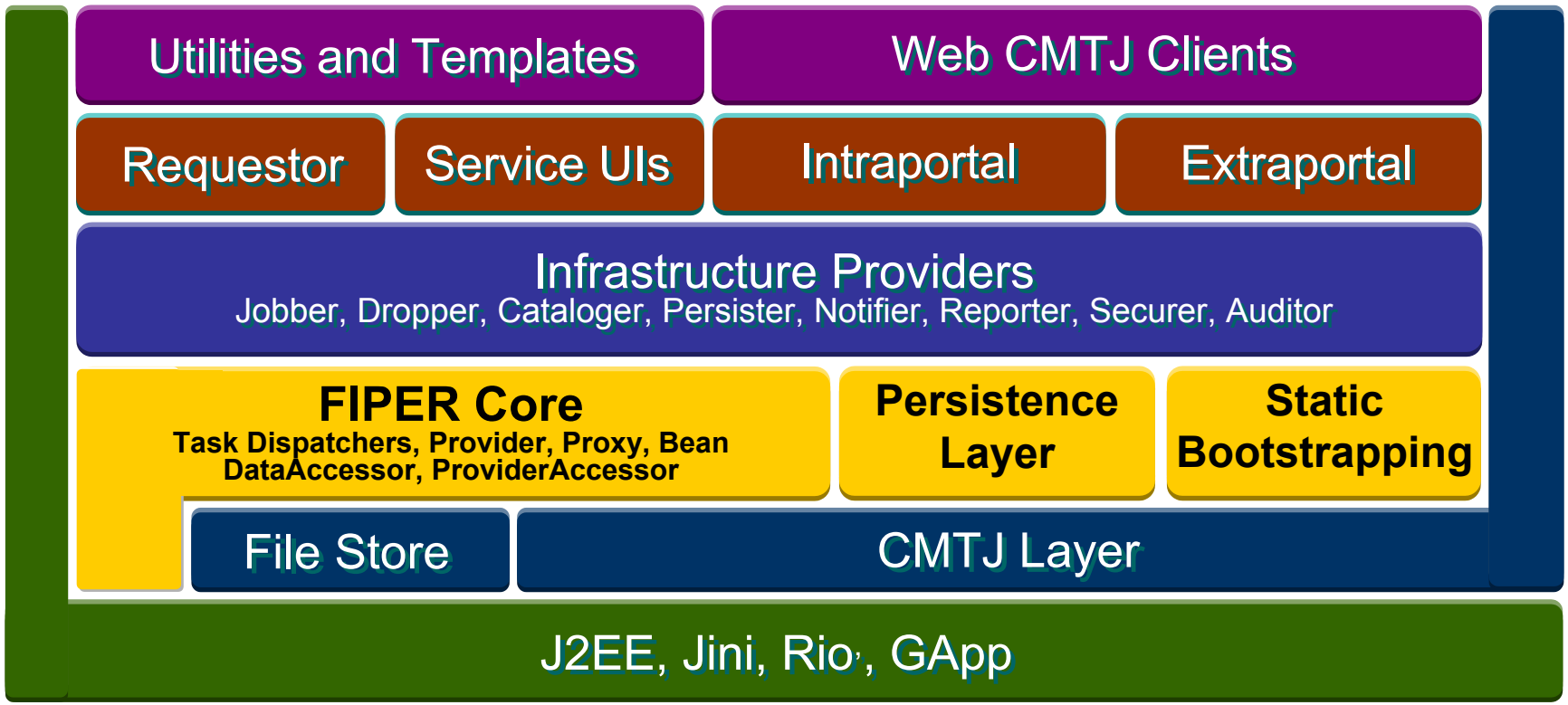
else

forward to a relevant service provider



JavaOne

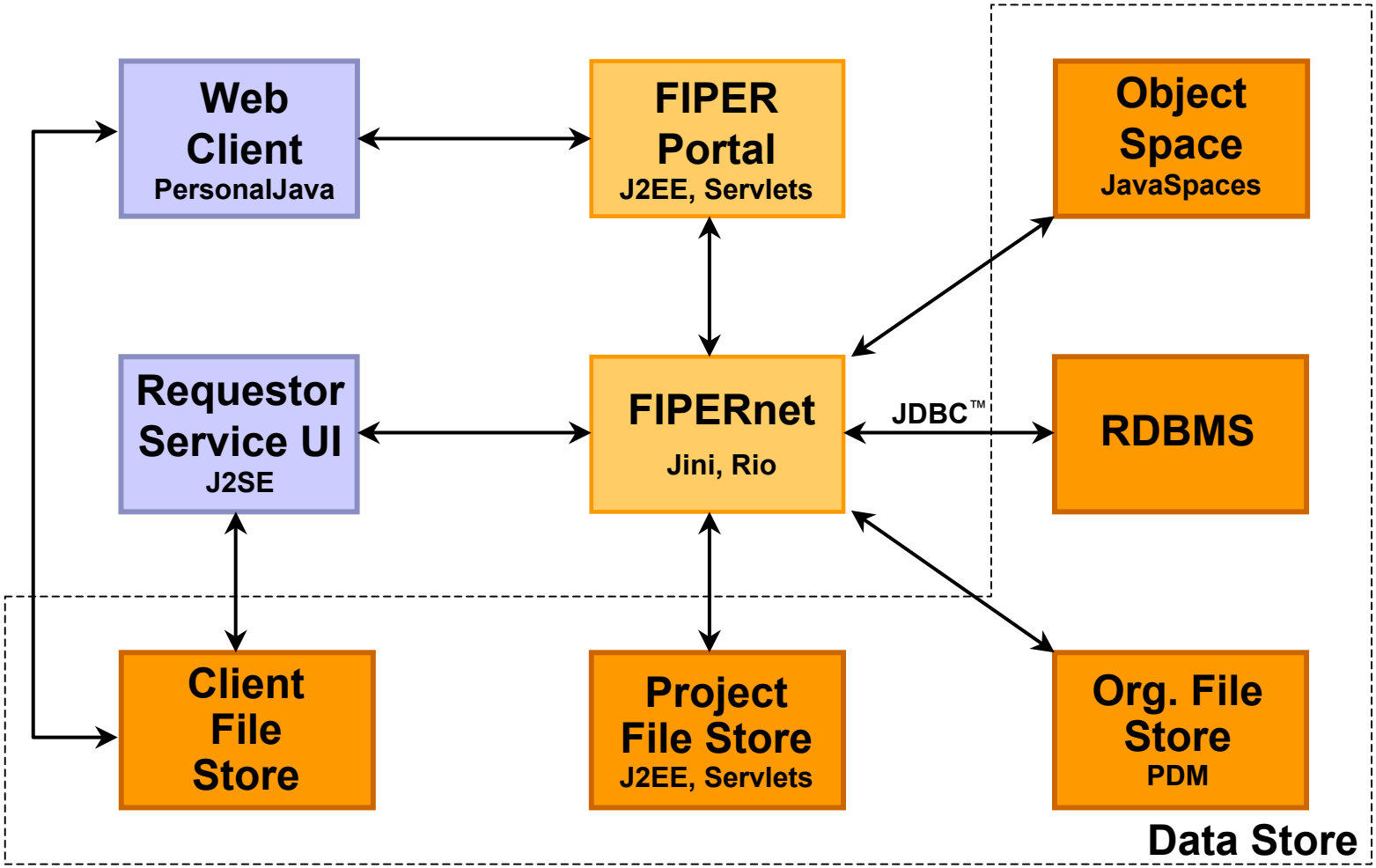
FIPER Functional Architecture Overview



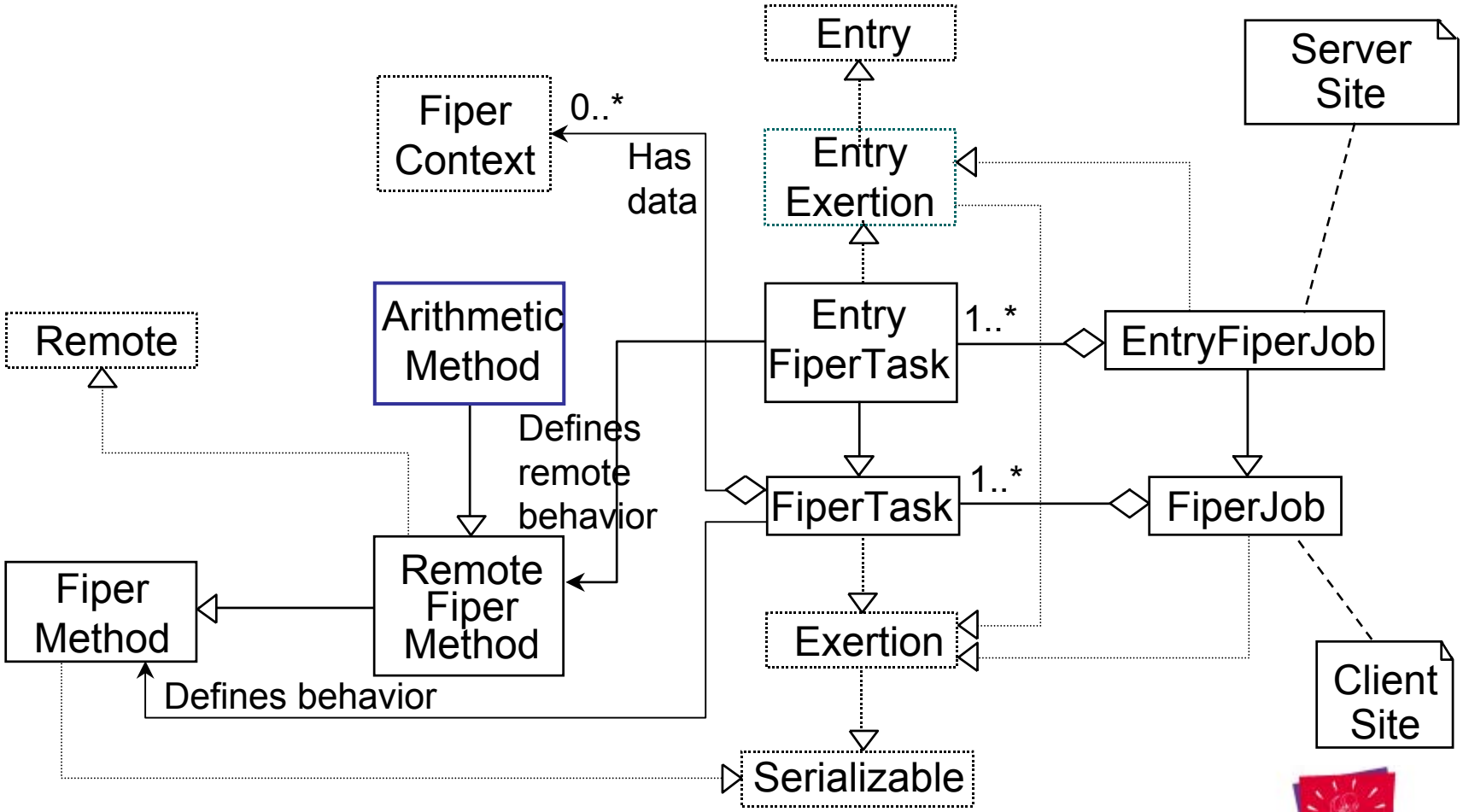
Dynamic Capability Trend

JavaOne

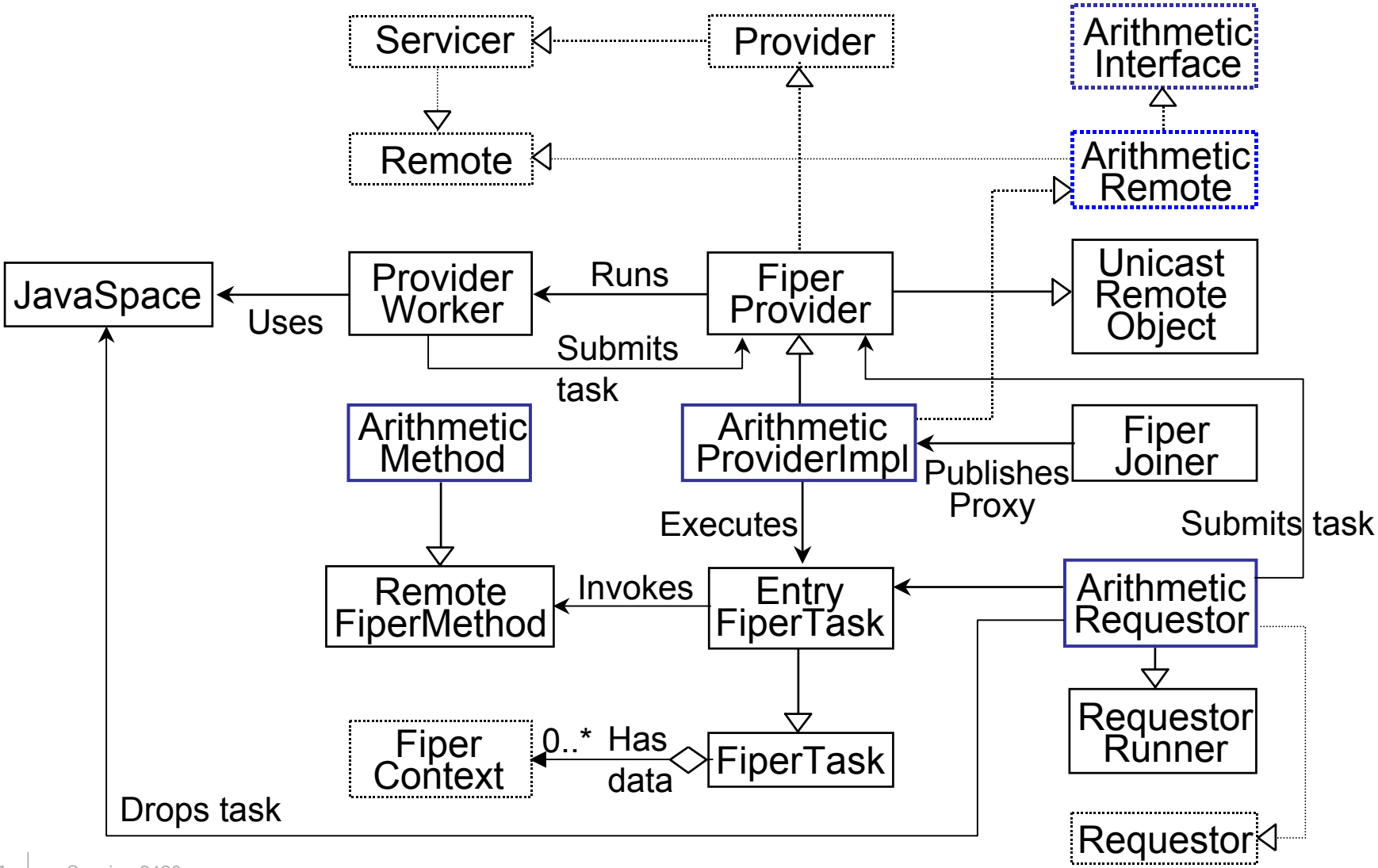
FIPER Organizational Architecture



Context/Method/Task/Job



Task Execution

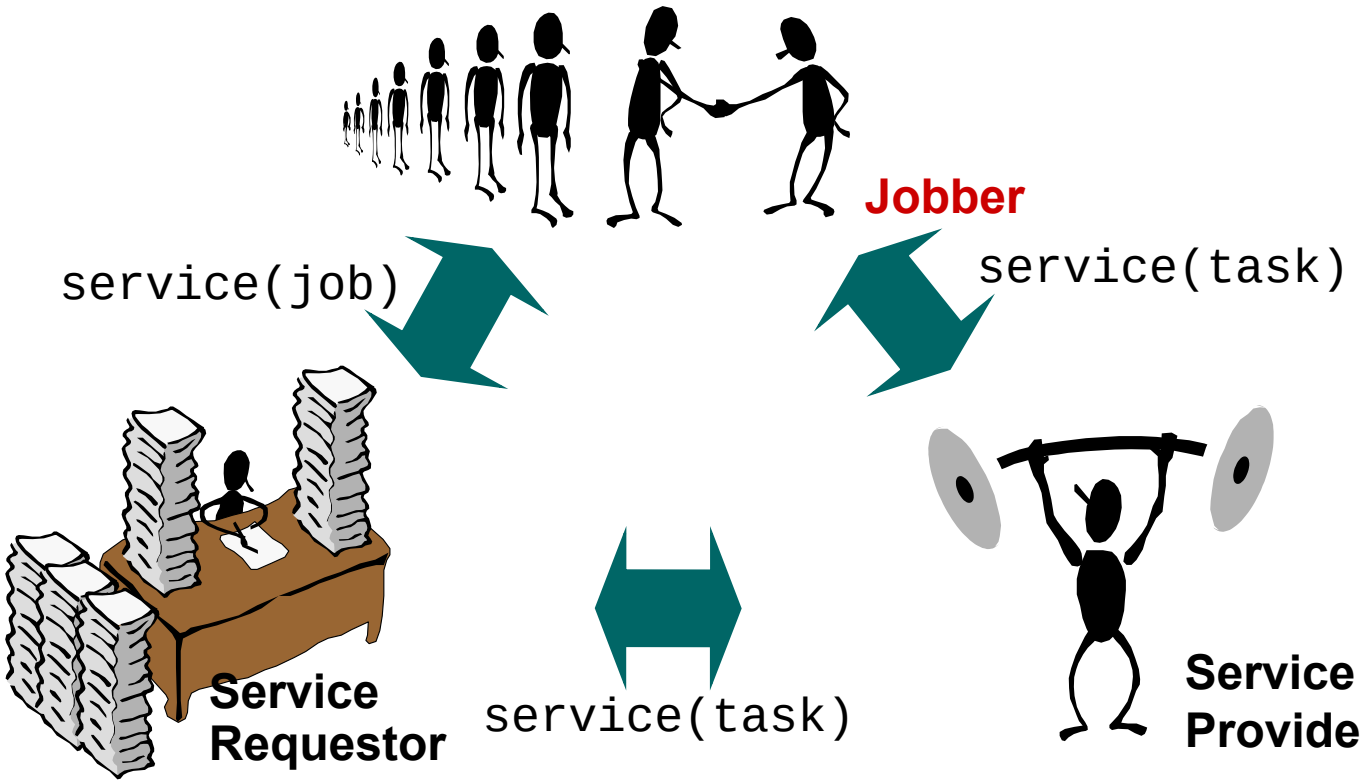


Question

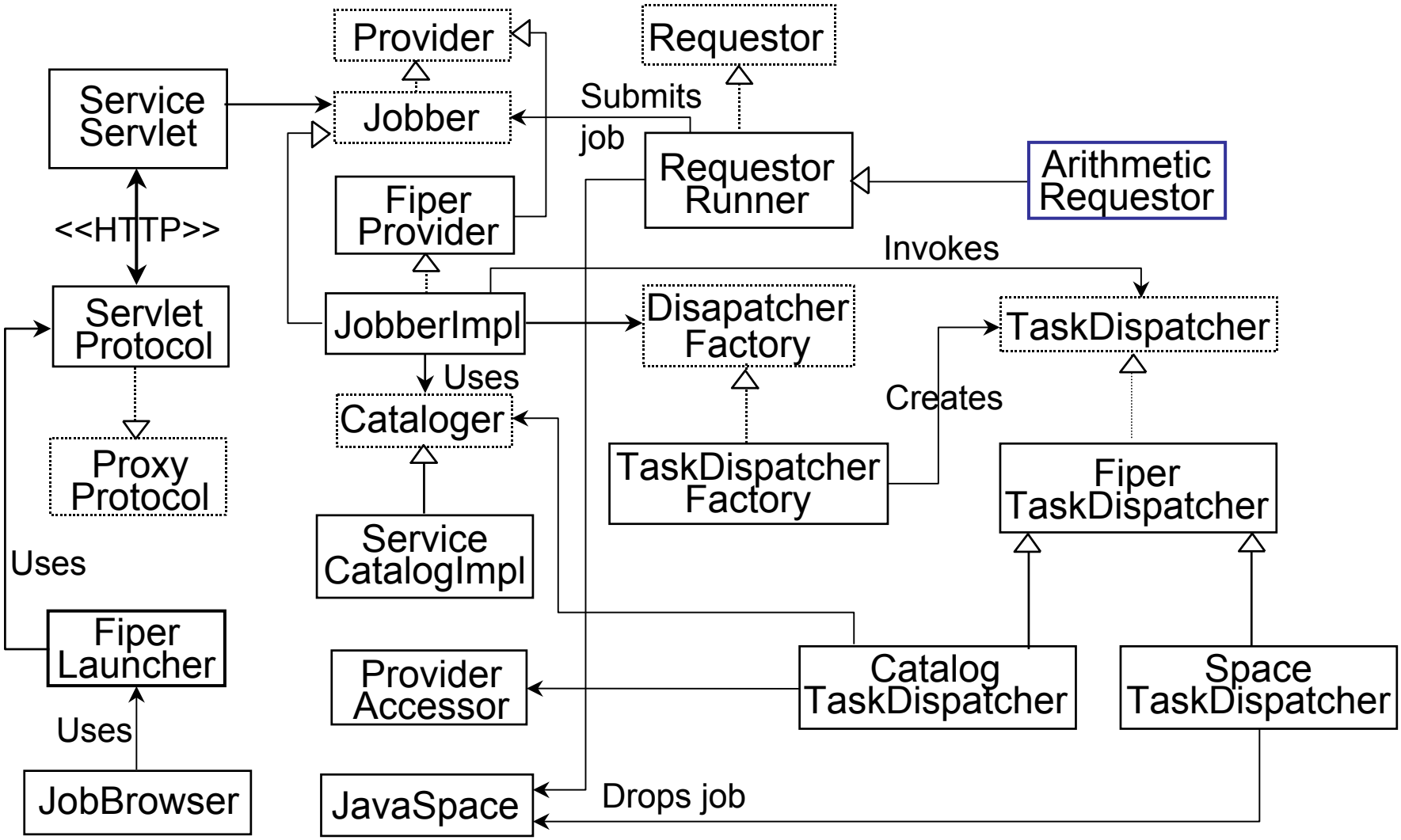
Does FIPER use service brokers?

Answer

A FIPER service broker is called a jobber.



Job Execution



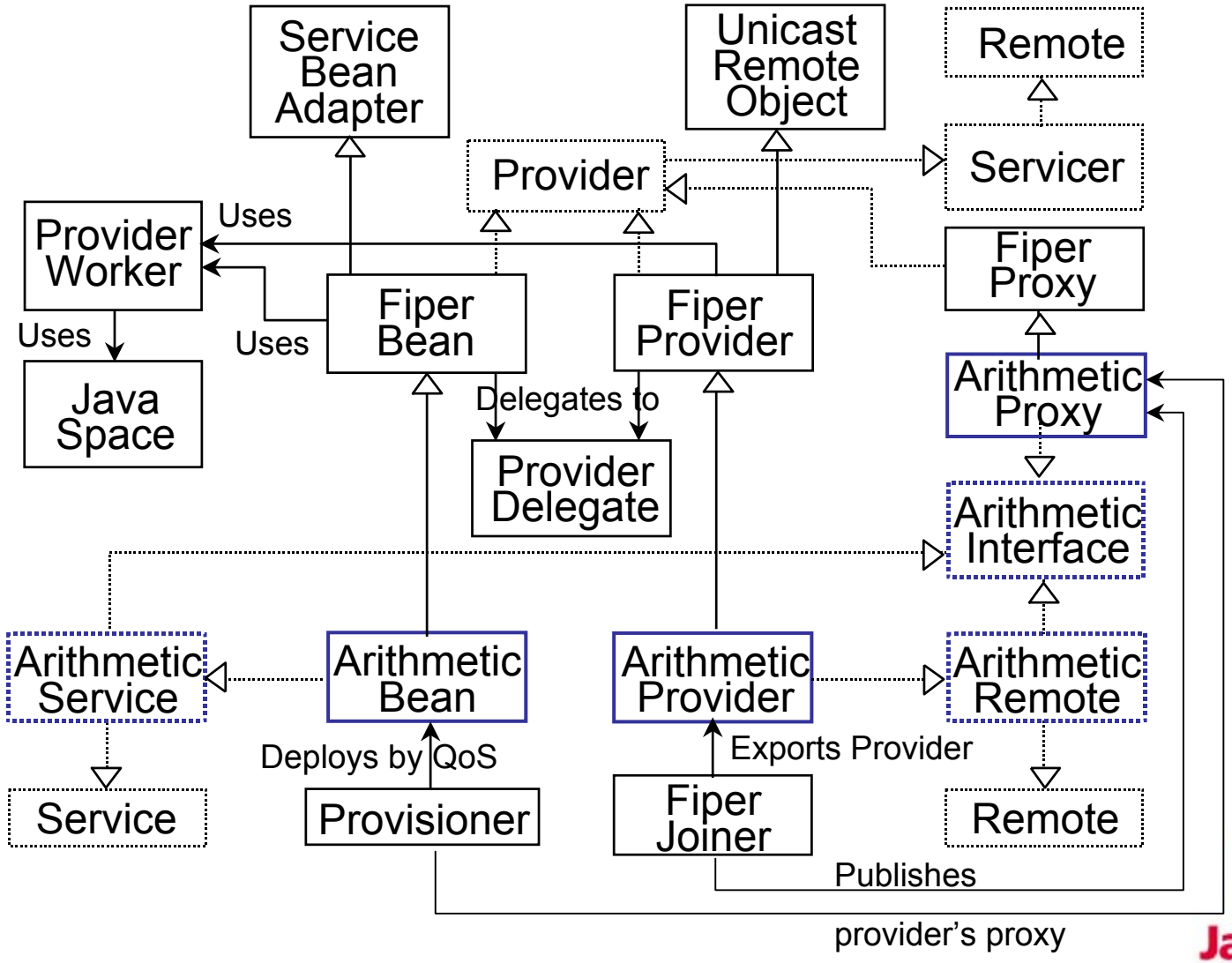
Service access: Direct, Catalog, Space

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

FIPER Smart Proxy
BL on Client and Server



JavaOne

Provider Bootstrapping

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

Mobile Code

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

FIPER Runtime Environment

Domain specific:

- Providers
- Requestors
- ServiceUIs

Infrastructure:

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

FIPER

- Websters
- Cybernodes
- Provisioners
- Lincolns

Rio

Web Server/App Server

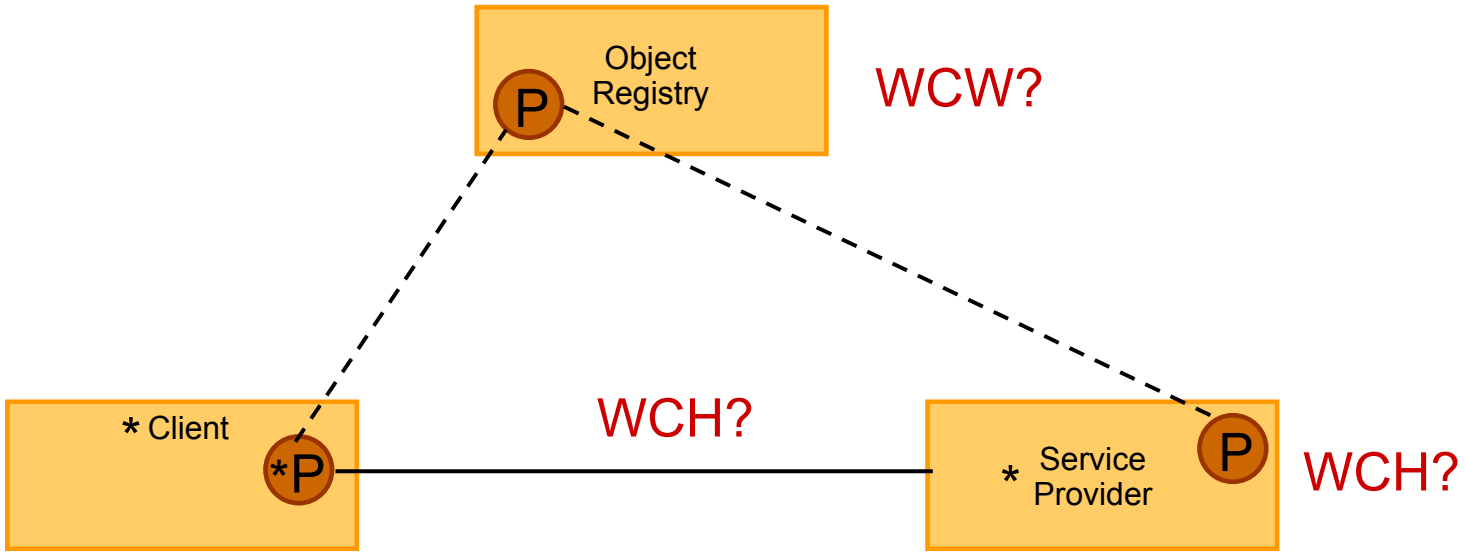
Summary (CNb)3

FIPER'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, S-N)
- Web Centricity
  - Interportals/Extraportals to services with thin web clients (applets/servlets)

Summary (CNb)3

FIPER's N^3



- Co-location Neutrality
- Protocol Neutrality
- Implementation Neutrality

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

JavaOne

Summary (CNb)3

Architecture Qualities b^3

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

Conclusion

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

Q&A

JavaOne℠

Sun's 2002 Worldwide Java Developer Conference™

BEYOND BOUNDARIES

Session 2420