

# Issues with Service Oriented Architectures

---

SPA Architecture Day  
22 January 2005



## Agenda

---

- Considerations
- Definitions
- Service Oriented Architectures
- Technologies
- Web Services
- Issues
- Applicability



## Disclaimer

---

- The views are those of the author alone
- Sometimes he even disagrees with himself

## To Begin...

---

- A service is not the same as a web service
- A component is not the same as a service
- Service oriented architectures don't need web services
- Web services are not the same as SOAP
- Service oriented architectures are not the same as object oriented architectures
- Forget RPC

## Definitions

---

- What is a service?
- What is a service oriented architecture?
- The literature is abysmally inconsistent
- We need a shared definition

## Service

---

- A service is a function that is well-defined, self-contained, and does not depend on the context or state of other services
- It is a promise, not an explanation
- “Generally better if it is stateless”

## Service Oriented Architectures

---

- Logical not physical view, typically business operations
- Message orientation: independence from how
- Description orientation: metadata with semantics
- Granularity: few operations with large and complex messages
- Generally network oriented
- Platform neutral messages

## Why?

---

- Loose coupling
- Platform independence
- Tool support
- Infrastructure availability
- Vendor support
- Because we have to...

## Choosing a Service Architecture

---

- Are web services the only game in town?
- CORBA
- Jini
- What can we learn?

---



## CORBA

---

- Abstraction through IDL
- Not proprietary
- Implementation independent
- Metadata repository
- Coarse granularity
- Network oriented
- Platform neutral through IIOP



---




## CORBA Issues

---

- Difficult
- Interoperability
- Service and infrastructure availability
- Cost
- Apparent encouragement of RPC model
- Lack of design patterns
- Internet unfriendly

---



## Jini

---

- Different view
  - Independence from protocol,  
dependence on presence of Java VM
- Abstraction through Java interface
- Implementation independent
- Metadata repository
- Coarse granularity possible
- Network oriented
- Platform neutral through VM



---



## Jini Issues

---

- Difficult
- Vendor and developer commitment
- VM availability
- Apparent encouragement of RPC model
- Lack of design patterns
- Lack of security
- Internet unfriendly

---

digitalsteps

## Lessons for Web Services

---

- Design patterns are essential
- Network latency impacts design
- Services will take time to stabilise
- Interoperability will continue to be a problem
- Service evolution requires careful thought
- Internet friendliness is important
- Leasing is A Good Thing™

---

digitalsteps

## Definition of a Web Service

---

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

**Web Services Architecture, W3C Working Group**



## Web Services

---

- Add Internet protocols
- XML, SOAP, WSDL, and UDDI
  - SOAP transport: HTTP, FTP, SMTP, RMI/IIOP...
- Currently lacking...
  - Security
  - Deployment
  - Logging
  - Failover and load balancing
  - Maintenance



## SOAP RPC

---

- A Bad Thing™
- Encapsulates application semantics
- Reduces extensibility
- Weakest information content
- Strong typing
- Ironic that this is the origin of SOAP

## Dynamic Discovery

---

- Index, registry, P2P
- UDDI: index or registry?
- Semantics generally *agreed* between parties
  - Seems unlikely to lead to dynamic binding
  - Optional component of web services architecture
- Does define the key metadata
- Compare interface matching in Jini lookup
- Web Services Inspection Language

## Problems with Web Services

---

- Inappropriate modelling techniques
  - Start with shared understanding
  - UML is not sufficient (or appropriate?)
  - Over extending existing paradigms
- Inappropriate application
- Inadequate interoperability
- WSDL is not appealing

---

digitalsteps

## Big Issues

---

- Security
  - Authentication is the most basic problem
- Transactions
  - Doing it
  - Distributing it
- Design
- WS-I is likely to be fraught with difficulties

---

digitalsteps

## Realisation Problems

---

- Weak typing
- Lack of design patterns
- Lack of modelling techniques
- Limitations of current implementations
- Lack of interoperability
  - Consider authentication of NT web services

## Take a REST

---

- Representation State Transfer, Fielding
- Emphasis on the message, not the method
- Simple set of invocations
  - CRUD
- Complex document exchange
- Uniform interface semantics, essentially those of the HTTP protocol
- Uniform operations: GET, PUT, DELETE

## Security

---

- Authentication Mechanisms
- Authorization
- Data Integrity and Data Confidentiality
- Integrity of Transactions and Communications
- Non-Repudiation
- End-to-End Integrity and Confidentiality of Messages
- Integrity and confidentiality in the presence of intermediaries

## Applicability

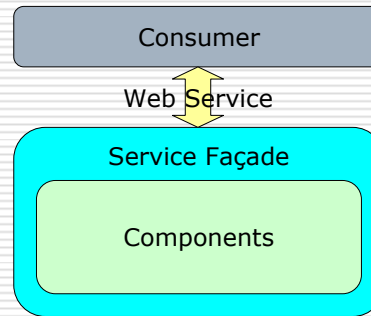
---

- J2EE
- Legacy integration
- Enterprise integration

## SOA and J2EE

---

- Remote interfaces are the work of the devil
- What is an EJB for?
- Pattern: "They're my components"
- Out of the box access



[digitalsteps](#)

## Legacy Integration

---

- Readily applicable
- Generally internal: more control and options
- Avoid  $n^2$  problem
- Simple interfaces assist extensibility
- Do you need guarantees?

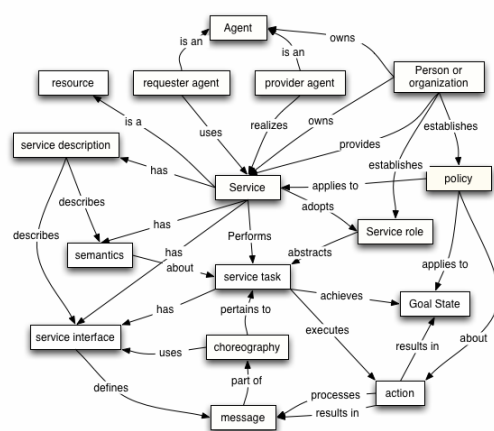
[digitalsteps](#)

## Enterprise Integration

- Poor man's EDI?
- Quality of Service
- Security
- Message semantics
- Did I mention the Internet?

digitalsteps

## Big Picture



©Web Services  
Architecture, W3C Note

digitalsteps

## A Broader Role?

---

- Application to general architectures?
- Principles apply
- Matter of definition

---

digitalsteps

## Questions?

---

[peter.black@digitalsteps.com](mailto:peter.black@digitalsteps.com)

---

digitalsteps