


**Interoperability and Service-Oriented Architectures:
Meeting the Demands of Systems
of Systems**


Tricia Oberndorf
Director, Dynamic Systems

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

 **Software Engineering Institute** | **CarnegieMellon** © 2007 Carnegie Mellon University

Agenda

Introduction to the Software Engineering Institute (SEI)
Basics of Interoperability
Service-Oriented Architectures

 **Software Engineering Institute** | **CarnegieMellon** Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University 2

The Software Engineering Institute

Federally Funded Research and Development Center, created in 1984

Sponsored by the U.S. Department of Defense, operated by Carnegie Mellon University

The SEI advances software and related disciplines to ensure the development and operation of systems with predictable and improved cost, schedule, and quality.

Washington, DC



Pittsburgh, PA



Frankfurt, Germany



Software Engineering Institute

CarnegieMellon

Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University

3

Agenda

Introduction to the Software Engineering Institute (SEI)

Basics of Interoperability

Service-Oriented Architectures



Software Engineering Institute

CarnegieMellon

Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University

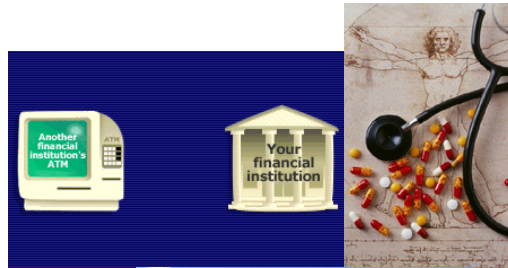
4



Interoperability Needs are Everywhere

Industry and Government

- Defense
- Healthcare
- Financial networks
- Retail
- Tax collection
- ...



What's the expected gain?

- Information superiority
- Efficiencies
- Competitive advantage



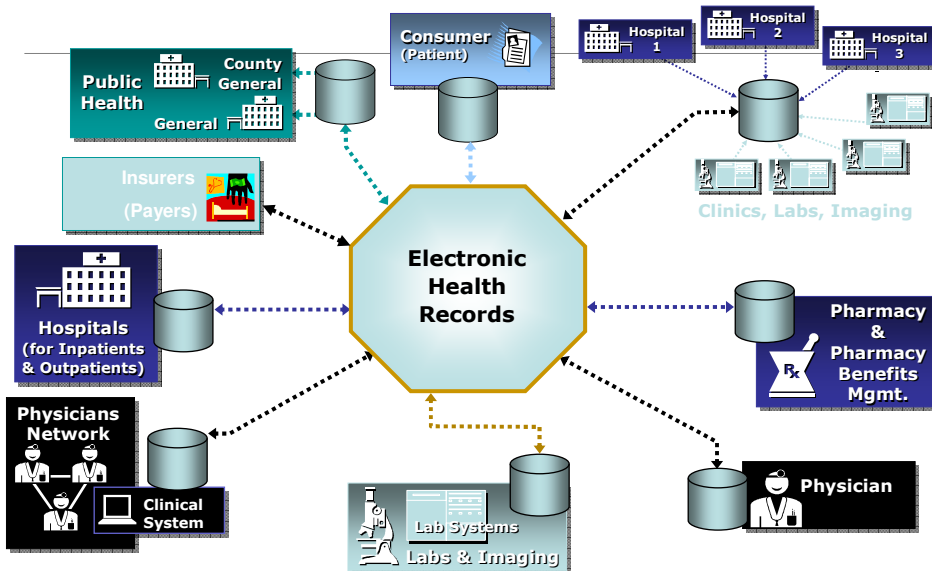
Software Engineering Institute

CarnegieMellon

Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University

5

The Healthcare Electronic Health Record (EHR) Concept



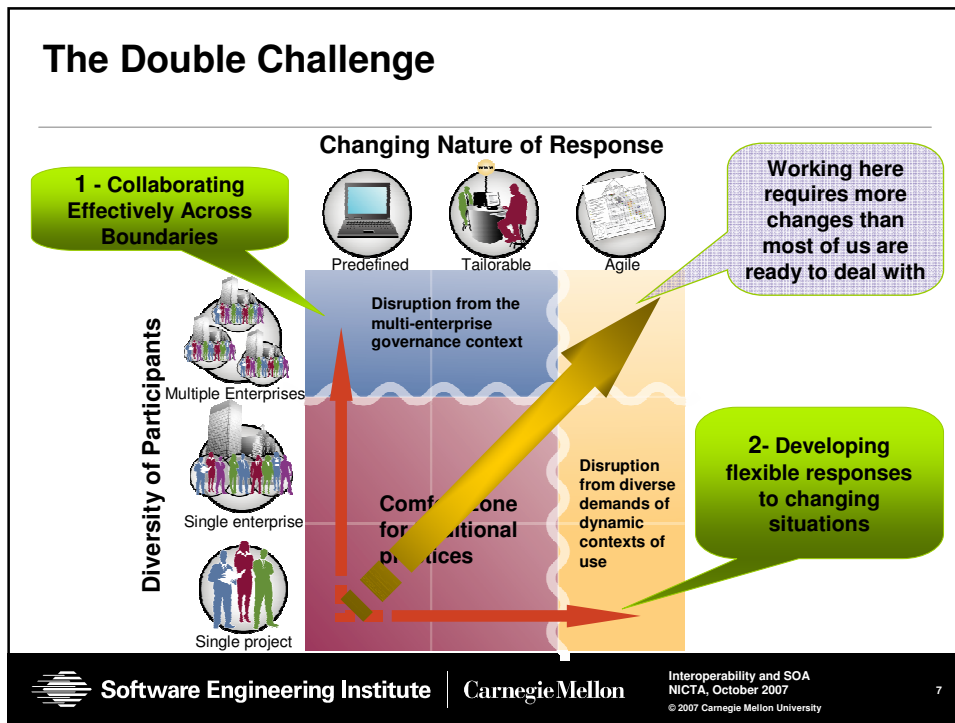
Software Engineering Institute

CarnegieMellon

Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University

6





Necessary Conditions for Migration to Net-centric Operations

1. A **legal and business framework** exists that encourages the acquisition of SoSs.
2. **Engineering practices** exist that are sufficient to create and evolve SoS.
3. A **social and cultural environment** is in place that motivates the collaborative behavior.
4. An **infrastructure** is in place that supports the NCO vision
5. A **technology base** exists that can deliver the realization of the vision
6. **Governance procedures** (e.g., how to sustain/operate tomorrow and in the future) for Systems of Systems are well understood and routinely practiced.

Agenda

Introduction to the Software Engineering Institute (SEI)

Basics of Interoperability

Service-Oriented Architectures



Software Engineering Institute

CarnegieMellon

Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University

9

What is SOA?

Service-oriented architecture is a way of designing systems that enables

- Cost-efficiency
- Agility
- Adaptability
- Leverage of legacy investments



Software Engineering Institute

CarnegieMellon

Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University

10



Services

Services are reusable components that represent business tasks.

- Customer lookup
- Credit card validation
- Weather
- Hotel reservation



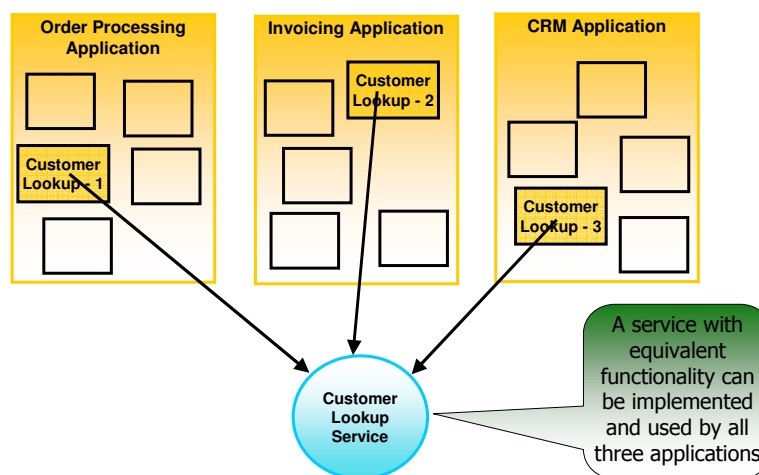
Services can be

- Globally distributed across organizations
- Reconfigured into new business processes

Service interface definitions are well-defined first-class artifacts (ideally) available in a service repository.



Services and Cost-Efficiency



Services and Agility

The diagram illustrates two applications, 'Order Processing Application' and 'Course Management Application', each represented by a yellow box. Below them are five services represented by blue circles: 'Customer Lookup Service', 'Credit Check Service', 'Item Lookup Service', 'Inventory Check Service', and 'Room Availability Service'. Arrows show that the Order Processing Application uses the Customer, Credit, and Inventory services, while the Course Management Application uses the Credit, Item, and Room Availability services. Two callout boxes provide context: one says 'The new application can use available services.' and another says 'New services can be used by other applications as well.'

Software Engineering Institute | CarnegieMellon

Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University

13

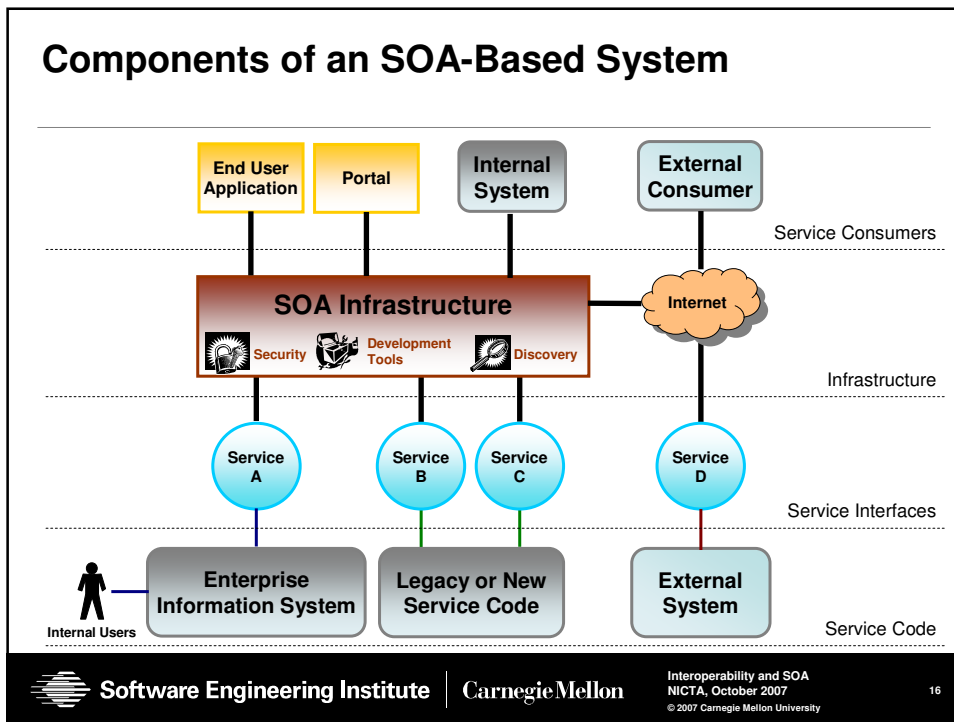
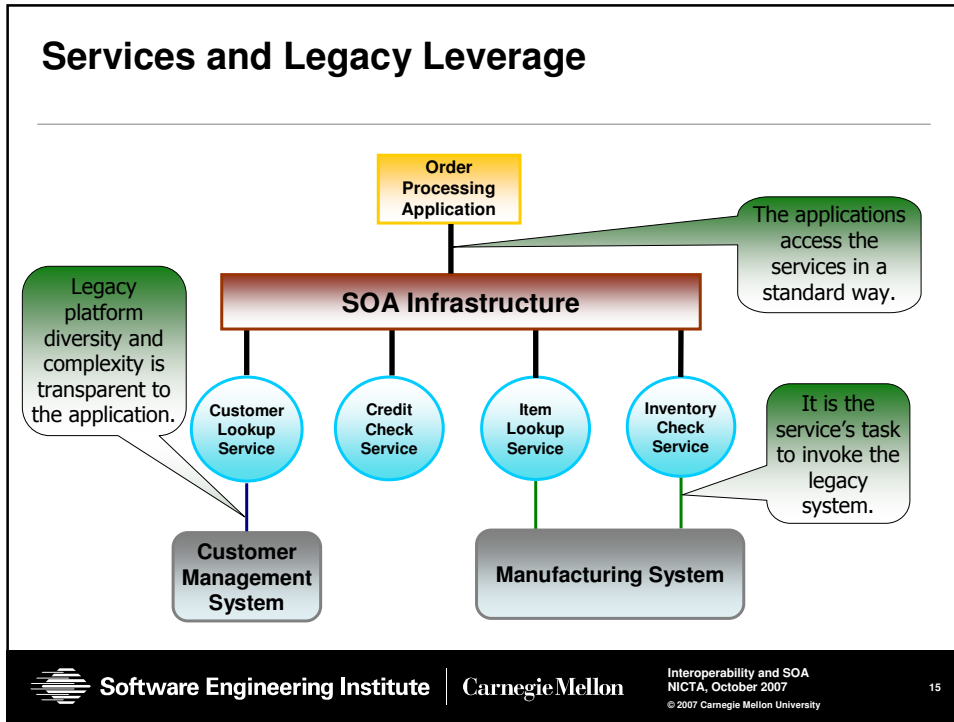
Services and Adaptability

The diagram shows the 'Order Processing Application' (yellow box) connected to four services (blue circles): 'Customer Lookup Service', 'Credit Check Service', 'Item Lookup Service', and 'Inventory Check Service'. A central brown box labeled 'SOA Infrastructure' acts as a mediator between the application and the services. A callout box states 'The SOA Infrastructure provides a standard communication mechanism between applications and services.' Another callout box notes 'Changes in services have potentially no impact on existing applications that use them.' A 'NEW' starburst icon is placed above the 'Item Lookup Service' circle.

Software Engineering Institute | CarnegieMellon

Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University

14

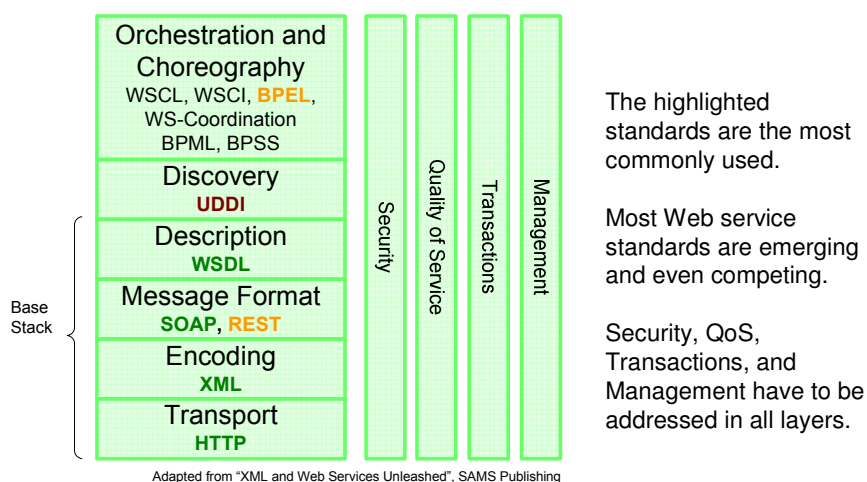


Some Common Misconceptions about SOA₁

- SOA provides the complete architecture for a system.
 - *SOA is an architectural pattern/style/paradigm and not the architecture of the system itself.*
- All legacy systems can be easily integrated into an SOA environment.
 - *Upfront hands-on analysis on the technical feasibility and return on investment must be performed to avoid last minute surprises.*
- SOA is all about technology.
 - *SOA is a shift in technology and the organizational governance model.*
- SOA is all about standards, and standards are all that is needed.
 - *There are many standards; not all are mature enough yet.*
- The use of standards guarantees interoperability in an SOA environment.
 - *Interoperability needs agreement on both syntax and semantics.*



Web Service Protocol Stack



Adapted from "XML and Web Services Unleashed", SAMS Publishing

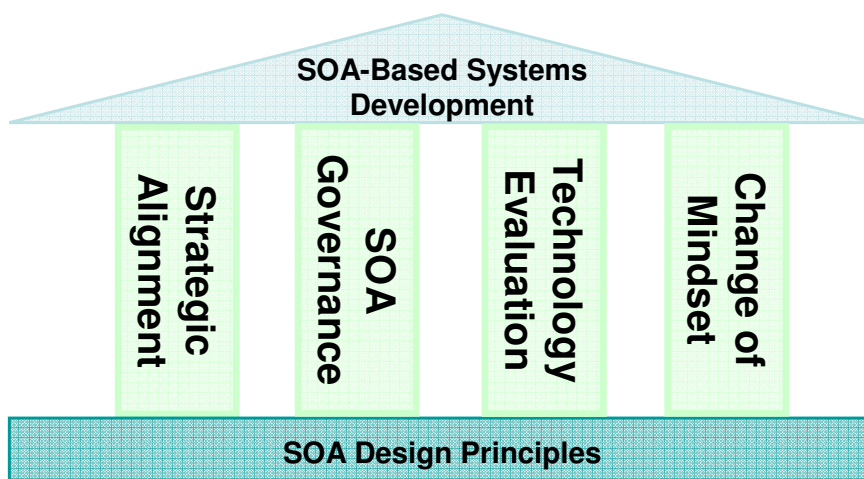


Some Common Misconceptions about SOA₂

- It is very easy to develop applications based on services.
 - It is *relatively* easy to build applications and services that work with a particular infrastructure ... but designing a “good” service may not be that easy.
- A service registry allows service binding dynamically at runtime.
 - Current technologies have not advanced to the point that this is possible in production environments.
- Testing SOA-based systems is no different than testing any other type of system.
 - Testing service consumers, as well as the services themselves, is challenging for various reasons.
- Everything in an SOA-based system has to be a service.
 - A service-oriented approach might not make sense for the whole system.



Pillars of SOA-Based Systems Development



T-CheckSM

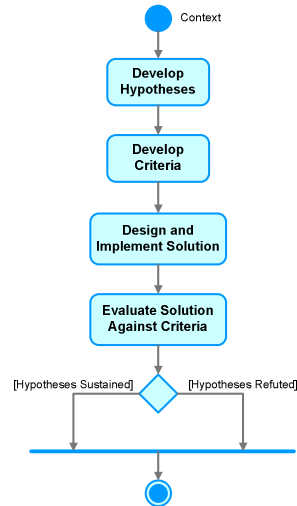
Experiment, situated in a specific context, with the goal of providing a “technology sanity check”

The approach

1. Formulate hypotheses about the technology
2. Examine these hypotheses against very specific criteria through experimentation

Extremely efficient

- Focus on implementing the simplest experiment to validate technology claims



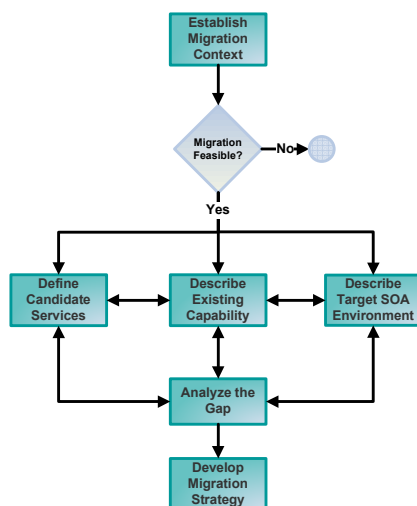
Software Engineering Institute

CarnegieMellon

Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University

21

Service Migration And Reuse Technique (SMART)



SMART analyzes the viability of reusing legacy components as the basis for services by answering these questions:

- Does it make sense to migrate the legacy system to services?
- What services make sense to develop?
- What components can be mined to derive these services?
- What changes are needed to accomplish the migration?
- What migration strategies are most appropriate?
- What are the preliminary estimates of cost and risk?



Software Engineering Institute




CarnegieMellon

Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University

22



Three Elements of SMART

Process 	Service Migration Interview Guide (SMIG) 	Artifacts 
Gathers information about <ul style="list-style-type: none"> • Goals and expectations of migration effort • Candidate services • Legacy components • Target SOA environment Analyzes gap between legacy and target state	Guides discussions in initial SMART activities	<ul style="list-style-type: none"> • Stakeholder List • Characteristics List • Migration Issues List • Business Process-Service Mapping • Service Table • Component Table • Notional SOA-Based System Architecture • Service-Component Alternatives • Migration Strategy



SOA Research Agenda IRAD

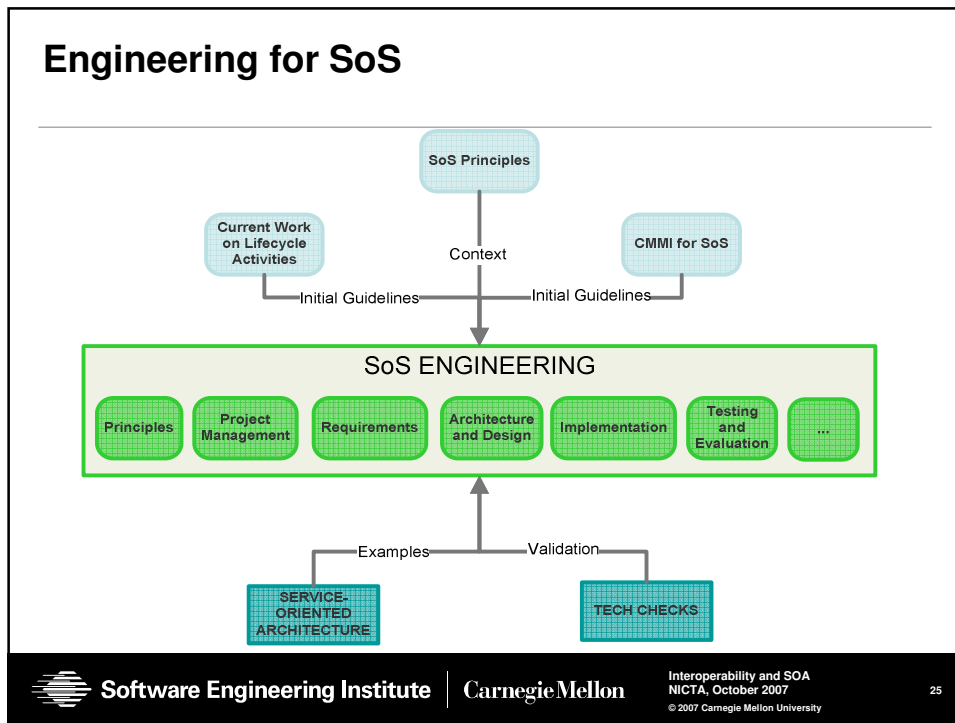
Purpose

- Develop a long-term research agenda for service-oriented architecture
- Determine if there is a community of interest to carry out the research agenda

Strategy

- Leverage an international team of researchers
- Develop initial taxonomy of research issues
- Determine current state of the practice, state of the art, open research issues
- Get broad input from research community
 - CSMR, March 2007
 - I-ESA – March 2007
 - ICSE, May 2007





References

Pointers to Technologies and Standards

- ISIS Guide to Interoperability: <http://www.sei.cmu.edu/isis/isis-guide.htm>

SMART: The Service Migration and Reuse Technique

- <http://www.sei.cmu.edu/publications/documents/05.reports/05tn029.html>
- *SMART: Analyzing the Reuse Potential of Legacy Components in a Service-Oriented Architecture Environment*. Proceedings of the 2007 AIAA Infotech@Aerospace Conference.

T-Check

- Process: <http://www.sei.cmu.edu/publications/documents/05.reports/05tn025.html>
- Applications:
 - Web Services: <http://www.sei.cmu.edu/publications/documents/06.reports/06tn021.html>
 - OWL-S (OWL Web Ontology Language for Services): <http://www.sei.cmu.edu/publications/documents/06.reports/06tn018.html>
 - MDA (Model-Driven Architecture): <http://www.sei.cmu.edu/publications/documents/05.reports/05tn022.html>

Software Engineering Institute | CarnegieMellon
 Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University
26

Contact Information

Tricia Oberndorf

(412) 268-6138

po@sei.cmu.edu

ISIS: Dennis Smith

dbs@sei.cmu.edu

SoS Eng: Grace Lewis

glewis@sei.cmu.edu

Software Engineering Institute
4500 5th Ave.
Pittsburgh, PA 15213-3890

www.sei.cmu.edu/programs/ds/index.html



Software Engineering Institute

CarnegieMellon

Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University

27



Software Engineering Institute

CarnegieMellon



Software Engineering Institute

CarnegieMellon

Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University

28



Backup/Additional Slides



SOA Infrastructure

Set of technologies that bind service consumers to services

- Products, standards and protocols that support communication
 - Typically message-based document exchanges
 - Web Services (HTTP, SOAP, WSDL)
 - Message-oriented middleware (i.e. IBM Websphere MQ)
 - Publish/subscribe (i.e. Java Messaging Service — JMS)
 - CORBA ...
- Infrastructure services available to service providers and/or service consumers
 - Security, discovery, data transformation, ...
- Development, deployment and management tools and guidelines



Service Consumers

Clients for the functionality provided by the services

- End-user applications
- Internal systems
- External systems
- Composite services

Consumers programmatically bind to services.



Software Engineering Institute

CarnegieMellon

Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University

31

Challenges for Service Consumers

Available services might not meet functional and non-functional requirements.

Services may change or disappear without notification.

Tools and programs provided by the infrastructure may conflict with development environment.

Services may not be semantically correct from the consumer's point of view.

Services coming from different organizations can have inconsistencies between them.

End-to-end testing would require test instances of all services to be available.



Software Engineering Institute

CarnegieMellon

Interoperability and SOA
NICTA, October 2007
© 2007 Carnegie Mellon University

32



Challenges for Service Developers

If consumer requirements are not understood, services may never be used.

The effort to translate legacy data types into data types that can be transmitted in messages can be greater than expected.

If dealing with proprietary SOA environments, there may be

- Multiple constraints imposed on developed services
- Dependencies on tools and programs provided by the infrastructure that are in conflict with development tools

Guidance for using Service-Level Agreements (SLAs) is often not clear.

- Benefits of SLAs are not well quantified.



Challenges for Infrastructure Developers

Changes in standards and products used in the infrastructure may have a large impact on its users.

- Especially emerging standards

Effort for development, support, and training for the use of tools and infrastructure may be underestimated.

