Composition of SOA based Applications

2008-01-23
Magnus Larsson
Agenda

• Introduction

• Provisioning of stable core services

• Agile composition of service based applications

• Summary
Providing one Service is easy...

```java
@WebServiceRef
private AbcService service;

Abc port = service.getAbcPort();
List<Pojo> list = port.find("foo");
```

```java
@WebService
public class Abc {
    public List<Pojo> find(String arg) {
        ...
    }
}
```
Upscaled provisioning gives some concerns...
Upscaled provisioning gives some concerns...
Upscaled provisioning gives some concerns...
Upscaled provisioning gives some concerns...

We need a stable core of services!!!
Stable core services challenges...

Holistic guidelines

Where are my services?

Service message structures

Service life cycles

Service interoperability
How to bring the values of SOA to the End Users?

End Users expects:

1. Fast development of new GUI
2. Quick adoption to changes in business processes

⇒ Agile development of Composite Applications

WSDL and XML Schema
Sample application - SOA Technology

- Seeking for innovation on how SOA can meet the End Users...
  
  ...we found one vendor driving innovation in this area
Sample application - SOA Technology

• Seeking for innovation on how SOA can meet the End Users...

  ...we found one vendor driving innovation in this area

• ...that is SAP
  – SAP NetWeaver Composition Environment (CE) 7.1
    • Built on top of Java EE 5.0
    • Tooling based on Eclipse 3.3
    • Support for the latest WS-* specs
    • Contains tools in all layers supporting agile development of composite applications based on a stable core of services

• Conceptual demonstrations will be based on SAP NW CE 7.1
Agenda

- Introduction

- **Provisioning of stable core services**
  1. Service interoperability
  2. Service message structures
  3. Service life cycles
  4. Where are my services?
  5. Holistic guidelines

- Agile composition of service based applications

- Summary
1. Interoperability in a heterogeneous environment

- Java EE
- MS .Net
- Legacy IBM iSeries and z/OS
- COTS SAP, Siebel, JD Edwards…
1. WS – Death Star...
1. WS – Death Star...

**WS-I Basic Profile**

...one specification to rule them all...

[www.ws-i.org](http://www.ws-i.org)

“WS-I is an open industry organization chartered to promote Web services interoperability across platforms, operating systems and programming languages.”
1. WS – I profiles

- Basic Profile
  - WS-Secure Conversation
  - WS-Security Policy
  - XML Signature
  - Kerberos profile
  - XCBF profile
- Basic Security Profile
  - WS-Federation
  - WS-Trust
  - XML Encryption
  - X509 profile
- Reliable Security Profile
  - WS-Authorization
  - WS-Privacy
  - Username profile
  - SAML profile
Agenda

• Introduction

• **Provisioning of stable core services**
  1. Service interoperability
  2. **Service message structures**
  3. Service life cycles
  4. Where are my services?
  5. Holistic guidelines

• Agile composition of service based applications

• Summary
2. Service message structures

![Diagram showing service message structures with ArtNo CHAR(10), PartNo LONG, and SkuId CHAR(18)]
2. Service message structures

We need a standardized system independent message format!

ArtNo CHAR(10)
PartNo LONG
SkuId CHAR(18)
2. Canonical Message Formats
2. Canonical service message structures
Agenda

• Introduction

• **Provisioning of stable core services**
  1. Service interoperability
  2. Service message structures
  3. **Service life cycles**
  4. Where are my services?
  5. Holistic guidelines

• Agile composition of service based applications

• Summary
3. Service life cycles

The only thing constant is...
3. Service life cycles

The only thing constant is change!
3. Service life cycles

- Services need to evolve and change over time
  - Requires new versions of services
  - Both backward compatible and non-compatible versions

- Important to establish a common version policy for services
  - Canonical Message Formats can provide a common solution

**NOTE:** Sample XML mechanisms to use in WSDL

- XML Schema Any - elements to define “extension points”
- Version number in XML namespaces
Agenda

• Introduction

• Provisioning of stable core services
  1. Service interoperability
  2. Service message structures
  3. Service life cycles
  4. Where are my services?
  5. Holistic guidelines

• Agile composition of service based applications

• Summary
4. Where are my services?

How do I get an overview of existing services?
- What version exists?
- Who are using what versions?

How can I search for services?

How do I find the proper End Points?
4. Service Registry

- A central repository for discovery and governance of services

- Development – Service Consumption
  - Development tool typically integrate with Service Registry to create client proxies

- Run Time – Deployment Information
  - Client proxies uses the Service Registry in runtime to lookup the actual service End Point

- Standards: UDDI and WSIL

- **DEMO:** Create a Web Service proxy using the Service Registry!
Agenda

- Introduction

- **Provisioning of stable core services**
  1. Service interoperability
  2. Service message structures
  3. Service life cycles
  4. Where are my services?
  
  **5. Holistic guidelines**

- Agile composition of service based applications

- Summary
5. Holistic guidelines

- **SOA Governance**
  - Central organization that guide and support the usage of SOA
  - Context given by existing Enterprise Architecture

- **Promotes**
  - Business value of SOA benefits
  - Proper use/re-use of services
  - Safeguard SOA projects
  - Accelerate learning curve

“Governance refers to the processes an enterprise puts in place to make sure things are done right — that is, in accordance with best practices, architectural principles, legal and industry regulations and other factors.”

“SOA Governance refers to the process used to govern adoption and implementation of SOA”

- Anne Thomas Mannes, Burton Group
Agenda

- Introduction
- Provisioning of stable core services
- Agile composition of service based applications
- Summary
How to bring the values of SOA to the End Users?

Agile development of Composite Applications
1. Web Services aware GUI tools
2. Tool support for defining workflows
What is a Composite Application?

- Workflow
- Presentation
- Application Logic
- Backend

Composite Application

Workflow 1

View 1

Local Service 1

View 2

Core Service 1

Core Service 2

Core Service 3

Core Service 4

Cadec 2008 - Composition of SOA based Applications, Slide 33
Copyright 2008, Callista Enterprise AB
Sample application - SOA Technology

Composite Application

Workflow

Presentation

SAP NW Guided Procedure

SAP NW Visual Composer
Web Services aware Web GUI tool - overview

- SAP NetWeaver Visual Composer
  - Creates GUI fragments based on Web Services
    - Creates default GUI from WS Req. and Resp. messages
    - Fully customizable using WYSIWYG - editor

- Parameter passing
  - Input and output to the GUI fragment - component
  - Also inside the component, e.g. between WS calls and UI elements
Web Services aware Workflow tool - overview

- SAP NetWeaver Guided Procedure
  - Defines workflows as blocks of actions
  - Actions calls
    - GUI fragment
    - Web Service

- Well defined parameter passing
  - Access to GUI actions are controlled by Roles
  - Workflow engine coordinate user activities
Sample application

Project Manager
- Identify Missing Skill

Employees
- Assign project members on course

Team Manager
- Book course event
- Approve course booking
- Receive notification

getAllProjects()
getProjectsByNameDesc()getEmployeesByProject()
identifyMissingSkillList()
getCourseBySkill()
getEventsForCourse()registerForEvent()

ProjectMgmService

SkillMgmService

CourseBookingService
Sample application - SOA Technology

SAP NW CE 7.1
Windows XP

ProjectMgmService
MS .Net 3.0 – WCF
Windows XP

SkillMgmService
SUN Glassfish 2.0
Mac OS 10.5

CourseBookingService
SAP Java EE 5.0
Windows XP
Demo Web Services aware Web GUI tool

- Identify Missing Skill
- Assign project members on course
- Book course event
- Approve course booking
- Receive notification

- Project Manager
- Employees
- Team Manager

- ProjectMgmService
- SkillMgmService
- CourseBookingService

- getAllProjects()
- getProjectsByNameDesc()
- getEmployeesByProject()
- identifyMissingSkillList()
- getCourseBySkill()
- getEventsForCourse()
- registerForEvent()
Agenda

- Introduction
- Provisioning of stable core services
- Agile composition of service based applications

- Summary
Summary - Provisioning of stable core services

- Use Canonical Message Formats to simplify usage of services
- Use WS-I Profiles to ensure Service interoperability
- Use a Service Registry to find your services
- Use a common version policy of Services for life cycle management
- SOA Governance to guide and control
Summary - Provisioning of stable core services

- Use Canonical Message Formats to simplify usage of services
- Use WS-I Profiles to ensure Service interoperability
- Use a Service Registry to find your services
- Use a common version policy of Services for life cycle management
- SOA Governance to guide and control
Summary - Provisioning of stable core services

- Use Canonical Message Formats to simplify usage of services
- Use WS-I Profiles to ensure Service interoperability
- Use a Service Registry to find your services
- Use a common version policy of Services for life cycle management
- SOA Governance to guide and control
Summary - Provisioning of stable core services

- Use Canonical Message Formats to simplify usage of services
- Use WS-I Profiles to ensure Service interoperability
- Use a Service Registry to find your services
- Use a common version policy of Services for life cycle management
- SOA Governance to guide and control
Summary - Provisioning of stable core services

Use Canonical Message Formats to simplify usage of services

Use a common version policy of Services for life cycle management

Use WS-I Profiles to ensure Service interoperability

Use a Service Registry to find your services

SOA Governance to guide and control
Summary - Provisioning of stable core services

SOA Governance to guide and control

Use a Service Registry to find your services

Use Canonical Message Formats to simplify usage of services

Use WS-I Profiles to ensure Service interoperability

Use a common version policy of Services for life cycle management
Summary – SOA reaching out to the End Users

- Agile development of Composite Applications
  - Requires
    - Stable core services
    - Web Services aware GUI and Workflow tools
Questions?