Managing CA Gen in a Service Oriented Architecture (SOA)

Gary Donoghue
IET UK

Information Engineering Technology

EDGE EMEA 2009
October 11-13, 2009
Amsterdam

Before we begin…

> Some caveats…

- This presentation outlines 1st generation web services being deployed in the world today
  - WSDL, XML, SOAP
- SOA 2.0 is not considered
  - WS-* extensions
- Dependent on future CA Gen support and overall strategy for Web Services

Who are IET?

> UK Based Company

> Exclusively develop products to complement CA Gen

> 20+ years of constant development and innovation ***using*** CA Gen

> Products used by 150+ CA Gen sites world wide

> TECHNOLOGY PARTNER

Information Engineering Technology

Presentation Outline…

> What is SOA?

> How does CA Gen support SOA?

> How can we manage change within a CA Gen SOA?

What is SOA?

> Service Oriented Architecture

An approach for building distributed computing systems based on encapsulating reusable business functions as SERVICES that can be easily accessed in a loosely coupled fashion

> Maturity of concepts outlined, within CA Gen, by CBD

- Consumer focused
- Platform independent
- Capable of realising potential of application, not just code, reuse

CA Gen Web Services Support

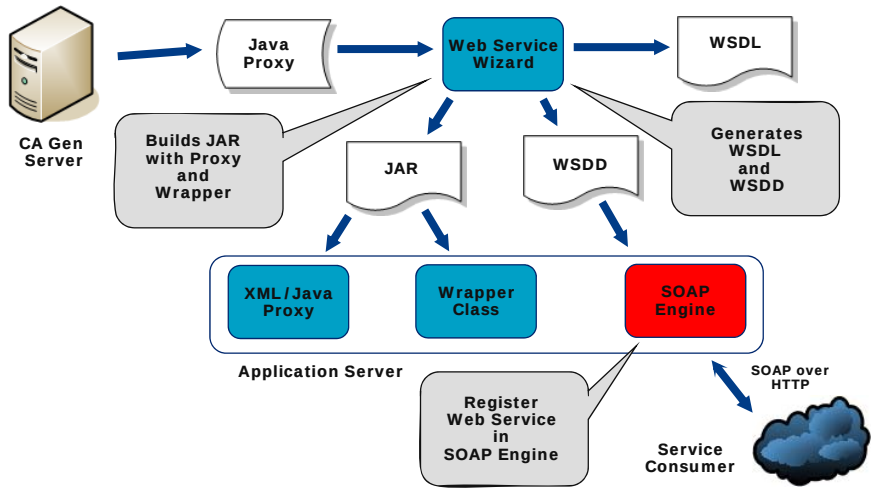
> Web Services Provisioning

- Server Procedure Step (modular, self-contained)
- JAVA Proxy, .NET Proxy, EJBs, Customised Web Service Interfaces
- Web Service Wizard Toolset Plug-in
- Register Web Service in SOAP engine (e.g. UDDI)

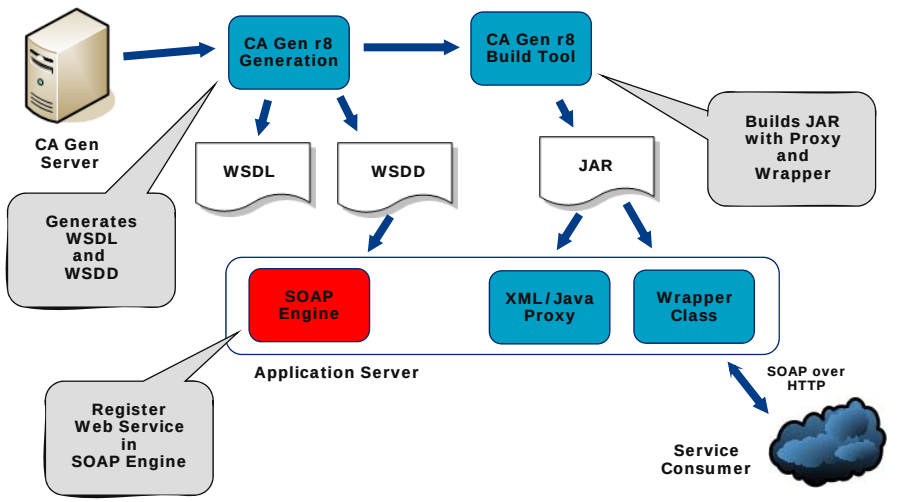
> Web Services Consumption

- Achieved by writing external calls (via EABs) to invoke Web Service
- 3rd Party Tools available
- *or* Gen r8 Web Service Runtime Router for EJBs

CA Gen Web Service Provisioning (Pre-Gen r8 JAVA Proxy)



CA Gen Web Service Provisioning (Gen r8 .NET Proxy and EJBs)



Managing Change in a CA Gen SOA

"Change is inevitable – except from a vending machine"

Robert C. Gallagher

SOA Perspectives on Change

> **SOA Governance**

- Defines activities related to ***exercising control*** over the Services deployed within a SOA
- Portfolio management (what do we have and where?)
- Lifecycle management (changes to services)
- Standards compliancy (for consistency)
- Performance

> **= SOA Trust**

- Increasingly seen as critical to SOA success

SOA Challenges of Change

> **SOA Governance** seen as increasingly critical
  - Particularly control of 'core' services
  - Lack of regulation, compliance and standards
  - Lack of change impact decreases trust
    - Impacts reuse, increases complexity, maintenance
  - Lack of change impact affects stability

> Weaknesses threaten SOA long term success and practicality
  - Consuming services requires acceptance of a degree of risk

GuardIEn Overview

> Change and Configuration Management designed specifically for CA Gen
  - Integrated Version and Change Control, Release Management, Automated Deployment

> HE and all CSE platforms

> 20 years of continual development

> Developed with CA Gen

> Readying for CA Gen r8

GuardIEn Overview

> GuardIEn maintains CA Gen object control

> CA Gen core objective to generate **all** application code from business logic stored within the repository

> Er. Almost. Alas there has **always** been a need to store some objects outside of the repository

- External AB source code
- JCL, scripts
- Copybooks
- Documentation, Help Files

CA Gen 'Externals' are increasing…

> Complexity and interaction 'breeds' externals…

- Bitmaps, icons, C header files

> As has opening up CA Gen via the Proxies…

- HTML, Java, VB, C#, Office

> External objects playing an increasingly critical role within the structure/future of CA Gen application development

- Exemplified by XML
- **Web Services, WSDLs via SOA**

GuardIEn *with* XOS can help

> Define and version control external objects, including check-in/check-out capability

> Manage associations between CA Gen and external objects to provide **complete** impact analysis

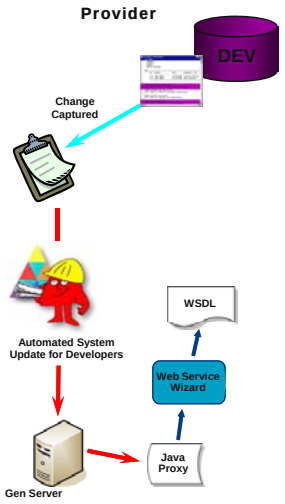
> Enables synchronisation of CA Gen and associated externals across application development stages

> Enables visibility of change(s) **across disparate** application development teams

CA Gen Web Service Provision & Consumption via GuardIEn

> Develop and 'Publish' a Web Service
  - Using CA Gen

CA Gen Web Service Provision via GuardIEn



> Standard Gen development process to build the Pstep(s) that will be exposed as operations within Service

- Change control and deployment automated by GuardIEn except…
- Web Service Generation workstation based manual process
  - Including packaging of operations into Service
- From Gen r8 for .NET and EJBs
  - Fully automated

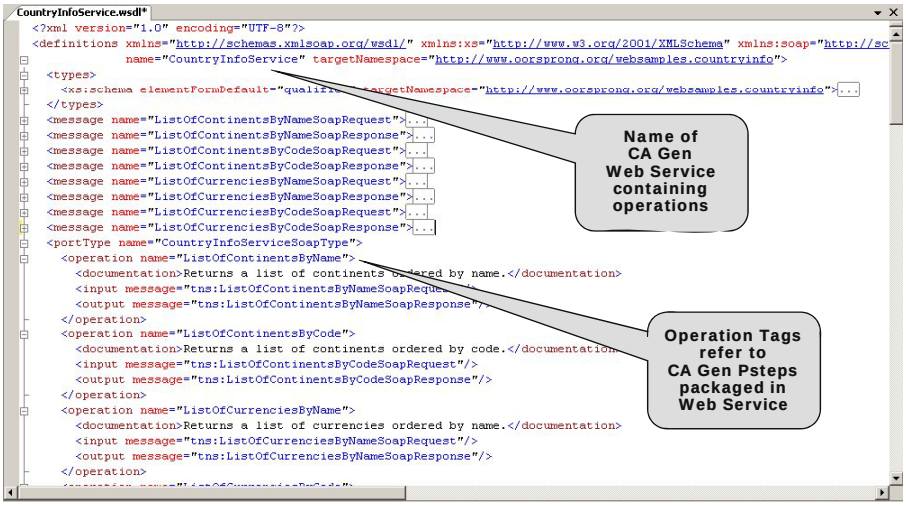
WSDL (Web Services Definition Language)

> The 'contract' between the service provider and the consumer

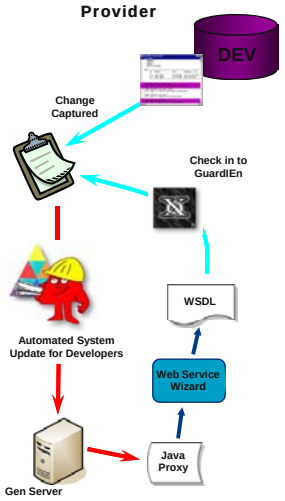
> A WSDL XML document unambiguously describes

- What the service does
- Where it resides
- How to call (invoke) it
- What information it requires
- What results you will receive back

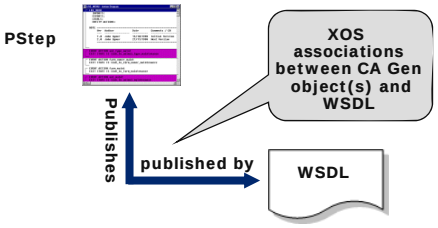
WSDL (Web Services Definition Language)



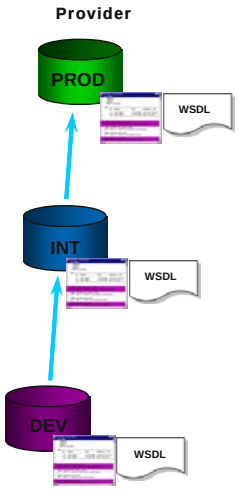
CA Gen Web Service Provision via GuardIEn



- XOS manages **XML associations** at check-in
- XOS ***parses*** uploaded XML and 'associates' with referencing CA Gen object(s) automatically



CA Gen Web Service Provision via GuardIEn



> Once 'checked-in' WSDL usage can then be easily assessed
  - Via Pstep(s) to which it relates
> GuardIEn System Updating synchronises promotion and deployment of CA Gen object, WSDL and related objects forward
  - Ensures WSDL deployment occurs securely and consistently alongside related CA Gen migration flow

CA Gen Web Service Provision via GuardIEn

> Other elements of Web Service also 'staged'…
  - .jar file to DEPLOY directory and…deploy
  - Deploy/undeploy WSDD to SOAP Engine
> System Update process executes WSDD
  - Informs Application Server that Web Service is now ready for use
> Register service into Registry server (e.g. MS UDDI)

CA Gen Web Service Provision & Consumption via GuardIEn

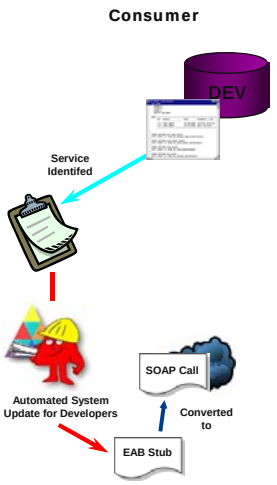
> Develop and 'Publish' a Web Service
- Using CA Gen

> Consume Web Service
- Locate Service
- Build Interface

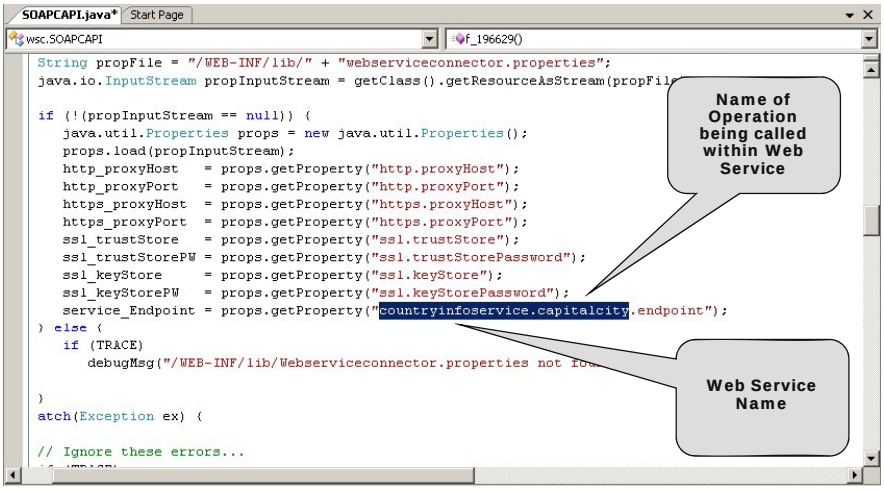
CA Gen Web Service Consumption via GuardIEn

> WSDL catalogue reviewed
- 3rd Party tools to automate consumption within CA Gen model
- EABs created **per** operation call
- CA Gen Abs/Psteps **USE** EAB to invoke service operation

> EAB stub controlled and deployed automatically via GuardIEn
- Java changes made on workstation to build Web Service call
- E.g. via COOL:Profs WebServiceConnector



JAVA EAB

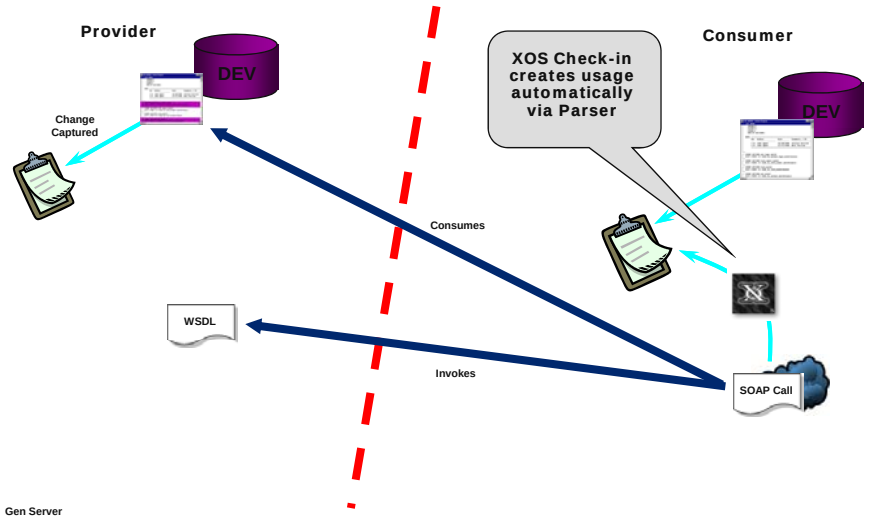


CA Gen Web Service Consumption via GuardIEn

> XOS manages **EAB associations** at check-in
  - EAB already connected to Psteps/Abs that USE service

> XOS ***parses*** uploaded JAVA and 'associates' with…
  - Pstep NAME from operation within Service
  - WSDL containing Service calls

GuardIEn provides full usage



CA Gen Web Service Consumption via GuardIEn

> GuardIEn System Updating synchronises promotion and deployment of operation consumption forward through lifecycle
  - (Re)generation into each environment or…
  - …staging of Java executables

> Change Management Pre-Requisites
  - If used can ensure Service provided before Consumer can attempt to consume it!

> Optional migration of Test Harness
  - Typically not required into Production!

CA Gen Web Service Provision & Consumption via GuardIEn

> Develop and 'Publish' a Web Service
- Using CA Gen

> Consume Web Service
- Locate Service
- Build Interface

> Manage Web Service Change
- Implications, Challenges

CA Gen Web Service Enhancement

> Change required to web service
- Assess current deployment & test schedules
- Assess impact of change
  - New Operations, Operation Changes, Internal Logic
- **May** require offer of **new** Service 'Version'
  - Entirely new Web Service

> Existing consumers **must** still function
- Loose coupled design allows backward compatibility
- Consumer has choice to take latest Service offering

CA Gen Web Service Enhancement via GuardIEn

> GuardIEn Impact Analysis/Usage facilities allow easy assessment of extent of change

> GuardIEn Change Control provides, for CA Gen projects, cross application 'associations' and impact

- Fast assessment of existing Service consumption
- Assists decision about deploying new Web Service *or* revision to existing

> GuardIEn System Updating provides mechanism for automating deployment of new changes consistently through development lifecycle

Some Conclusions…

> GuardIEn with XOS provides **'technical'** layer impact analysis and version control of change

- From Development to Production launch
- For Internal and External elements of CA Gen SOA application

> GuardIEn with XOS automates deployment

- Ensures consistency of implementation into each environment
  - Between CA Gen components and externals
- Speeds implementation

> Scope for further automation though…

Wrapping it up…

> SOA is a rapidly evolving ***amalgam of technologies*** that aim to simplify the consumption and reuse of application business logic
  - Broadening the scope of reuse (for CA Gen) beyond CBD

> SOA Governance emerging as increasingly important to longer term success
  - Impact analysis and change control
  - Longer term maintenance

> GuardIEn can play its part
  - For CA Gen projects

Questions?