TOWARDS A PATTERNS BASED SOA IN THE ENTERPRISE

Ayal Rosenberg, CTO
PDEV Pty Ltd.
ayal@pdev.co.za
http://www.pdev.co.za
MISSION STATEMENT

ARCHITECT AN ENTERPRISE

SOA APPLICATION

ARCHITECT AN ENTERPRISE
“What do I mean by the term ‘enterprise application’? I can’t give a precise definition, but I can give some indications of my meaning.” – Fowler [PEAA]

- Persistent Data
- A lot of data
- Concurrent access to data
- Lots of user interface screens
- Integration with other Enterprise applications.
ENTERPRISE - DEFINITION

- “Enterprise level software is software which provides business logic support functionality for an organization, typically in commercial organizations, which aims to improve the organization’s productivity and efficiency.” - http://en.wikipedia.org/wiki/Enterprise_software
“Enterprise class business solutions, the kind that companies bet their business on, are often extremely complex and must perform well against high expectations. Not only must they be highly available and scalable in the face of unpredictable usage, they must also be malleable and predictable in response to rapidly changing business requirements.” - Enterprise Solution Patterns Using Microsoft .NET
“Enterprise class business solutions, the kind that companies bet their business on, are often extremely complex and must perform well against high expectations. Not only must they be highly available and scalable in the face of unpredictable usage, they must also be malleable and predictable in response to rapidly changing business requirements.” - Enterprise Solution Patterns Using Microsoft .NET
“Design is a funny word. Some people think design means how it looks. But of course, if you dig deeper, it’s really how it works.” – Steve Jobs 1996 interview with Wired.
“Design is a funny word. Some people think design means how it looks. But of course, if you dig deeper, it’s really how it works.” – Steve Jobs 1996 interview with Wired.
“SOA is an architectural style that supports service orientation. Service orientation is a way of thinking in terms of services and service-based development and the outcomes of services. An architectural style is the combination of distinctive features in which architecture is performed or expressed,” - OG


“In order to promote better understanding and to give the IT industry an opportunity to make progress, it is important to come to a single, generally agreed definition of SOA.”
‘The OASIS SOA TC will develop a Reference Model for SOA. This is primarily to address SOA being used as a term in an increasing number of contexts and specific implementations. Sometimes, the term is used with differing – or worse, conflicting – understanding of implicit terminology and components.’ – OASIS http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm
“A service-oriented application is simply an aggregation of services into a single logical, cohesive application, much as object-oriented application is the aggregation of objects. The application itself may expose the aggregate as a new service, much like an object can be composed of smaller objects.” – Juval Lowy Programming WCF Services
THE APPLICATIONS

- Loan applications.
- Interact with Credit Bureau.
- Existing legacy systems. Algorithms and rules in legacy. Legacy issues the loans.
- Each application a long-running workflow.
- UI must be Windows Forms application using CAB.
- SOA must be achieved using WCF.
BASE LINE ARCHITECTURE

EXTERNAL SERVICES

LEGACY

SERVICES

CAB UI
MISSION STATEMENT

ARCHITECT AN ENTERPRISE

SOA APPLICATION

SOA APPLICATION

ARCHITECT AN ENTERPRISE
CLIENT #1: PAYLOAD

- COBOL
- ORACLE
- FLAT FILE

- BROKER REPOSITORY
- APPLICATION REPOSITORY
- BIZ-TALK LEGACY BROKER
- DOMAIN & DATA ACCESS
- SERVICES

EXTERNAL SERVICES

Collate and gather reference data

CAB UI
CLIENT #1: PAYLOAD

- COBOL
- ORACLE
- FLAT FILE
- BROKER REPOSITORY
- APPLICATION REPOSITORY
- BIZ-TALK LEGACY BROKER
- DOMAIN & DATA ACCESS
- SERVICES
- EXTERNAL SERVICES
- CAB UI

Loan Process workflow
CLIENT #1: PAYLOAD

- COBOL
- ORACLE
- FLAT FILE
- BROKER REPOSITORY
- BIZ-TALK LEGACY BROKER
- APPLICATION REPOSITORY
- DOMAIN & DATA ACCESS
- SERVICES
- CAB UI

Send payload
CLIENT #2: TRANSACTIONAL

- SAP
- DOMAIN & Legacy Broker
- SERVICES
- EXTERNAL SERVICES
- CAB UI
Client 1

- Simple UI – Client side processing
- Distinct server side domain model
- 5 Developers
- 8 Months
- 2 million Rands
Client 2

- Decorative UI – no Client side processing
- No server side domain model
- 30 Developers, 5 testers, 3 project managers
- 10 Months
- 40 million Rands
LESSONS

- Know the technology and use it.
- Boundaries are explicit!!!!
- Refactor to services.
- Services indeed are autonomous!
- Service metrics are vital to SOA design
- Design fundamental drives services
“In preparing for battle, I have always found that plans are useless but planning is indispensable.”

- General Dwight Eisenhower
“I believe that computer programming is a practical skill and I agree with Polya that ‘we acquire any practical skill by imitation and practice.’ For immediate applicability, don’t forget the deep insight of Tom Duff: ‘Whenever possible steal code.!!!’ "

- Jon Bentley  *The Most Beautiful Code I Never Wrote*
Service Metrics - 1

- Results in more specialized, loosely-coupled, fine-tuned, reusable contracts.
- Results in contracts with fewer operations.
- Balance cost of implementing contract against cost of integrating them into cohesive application.
- Too granular no good. Too large also no good.
Service Metrics - 2

- Well designed system has not too many but not too few services.
- Avoid single operation services.
- Optimal number of services between three and five.
- 12 or more bad – 20 not allowed.
Rules of thumb and generic metrics are only tools to help gauge and evaluate a particular design. There is no substitute for domain expertise and experience.
“If I’d my customers what they wanted, they’d have said a faster horse.” – Henry Ford.
“Don’t listen to your customers! They don’t know what they want!” – Steve Jobs.
“Actually this book is build on the premise: that good code matters. I have seen too much ugly code make too much money to believe that quality of code is either necessary or sufficient for commercial success or wide spread use.”

Kent Beck – *Implementation Patterns*