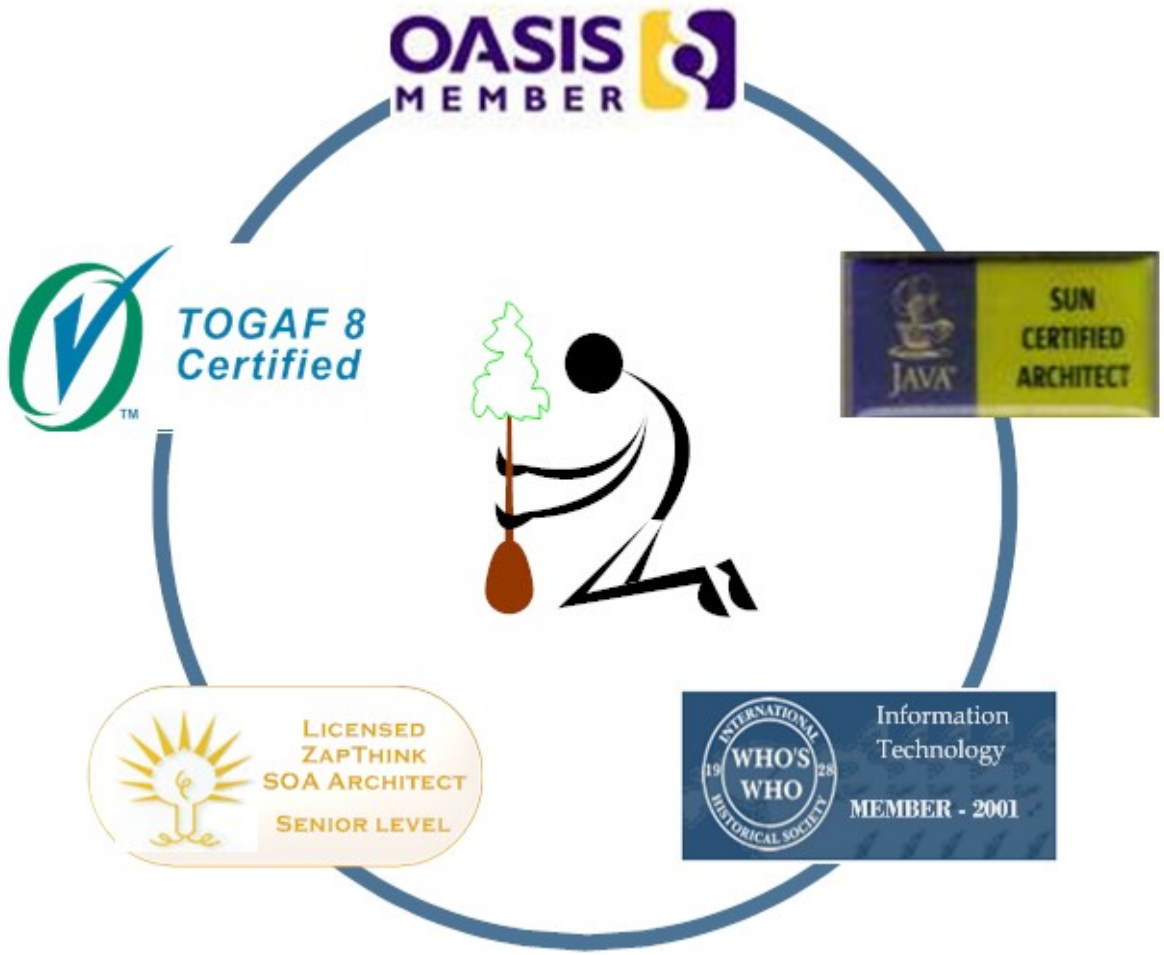
Real Life SOA

by Michael Poulin,
enterprise-level solution architect

London QCon 2009
Conference Mar. 11-13

Introduction



Michael Poulin works as an enterprise-level solution architect consulting in the UK Financial Industry for the last 3 years.
Previously, he worked in the same capacity in the USA.

Contact e-mail:
michael.poulin@consultant.com

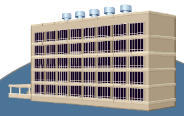
ALWAYS READY TO HELP YOU

Agenda

- Consequences of business changes
- Design solutions and lessons learnt in the Financial Industry:
  - Changes of service behavior in the execution context (policy influence)
  - UI for Business Service (Conciliator Pattern) – changes in between
  - Handling of changes via service reuse (Types of Reuse)
- Domain Service-Oriented Modelling (DOSOM)
- Food for thought (ideas to take away)

The World of Changes

FINANCE
- Changes in investment tactics
- Changes in the cash flows



ORGANISATION
- Changes in goals and strategies
- Change of plans
- Change of priorities
- Changes in resources

MARKET
- Changes in rules and regulations
- Changes in partnership

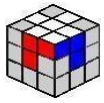
SOCIAL AND DEMOGRAPHIC ENVIRONMENT
- Changes in approach to employment
- Changes in the employment morale
- Change of available expertise

"There is nothing permanent except change"

- Heraclitus, 6th Century BC

"Enterprises that can rapidly configure new business models based on changing market dynamics will have a distinct advantage over enterprises lacking the technical and business process ability to alter business models on the fly"

- Tony Murphy, author of "Achieving Business Value From Technology"



Environment is changing faster and on a larger scale than technology and its organisation can handle

Flexibility is the Key for Efficiency

Flexibility in adopting changes is the fundamental mechanism for reaching Efficiency

Service Orientation, when applied across Business and Technology, provides for maximum flexibility in an organisation

Maximum Flexibility: adaptation of changes with

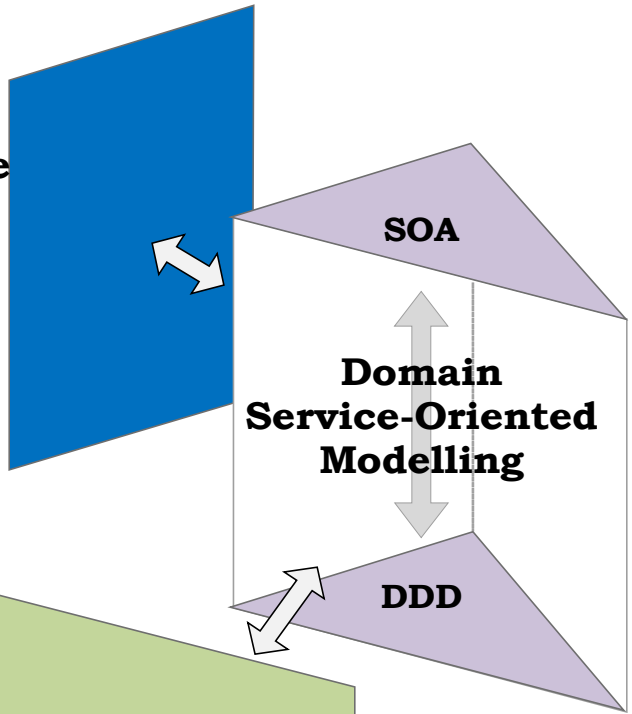
- minimum **imp**lementation cost
- minimum **inv**estments into the follow-up maintenance and modifications of surrounding environment
- minimum **t**ime-**to**-**m**arket

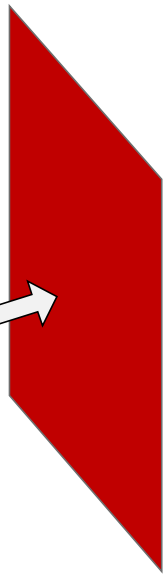
max ('flexibility') = min { ∑ (IMP+INV+T2M) }

SO Solutions Designed for Changes

Service in the Execution Context

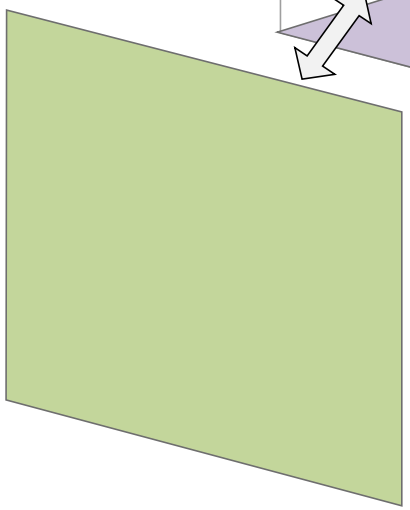
- Business context
- Technical context





Service Reuse 'by extension'

- Operation as container
- Composite message schema
- Optional elements



UI for Business Services

- Business Services UI
- Business Process UI
- Business UI Aggregation
- User Experience and Business logic

SO Principles help to deal with 'Change'

Principles of Service-Orientation

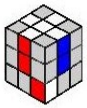
- Service reusability
- Service contract
- Service loose coupling
- Service abstraction
- Service composability
- Service autonomy
- Service statelessness
- Service discoverability

http://ServiceOrientation.org

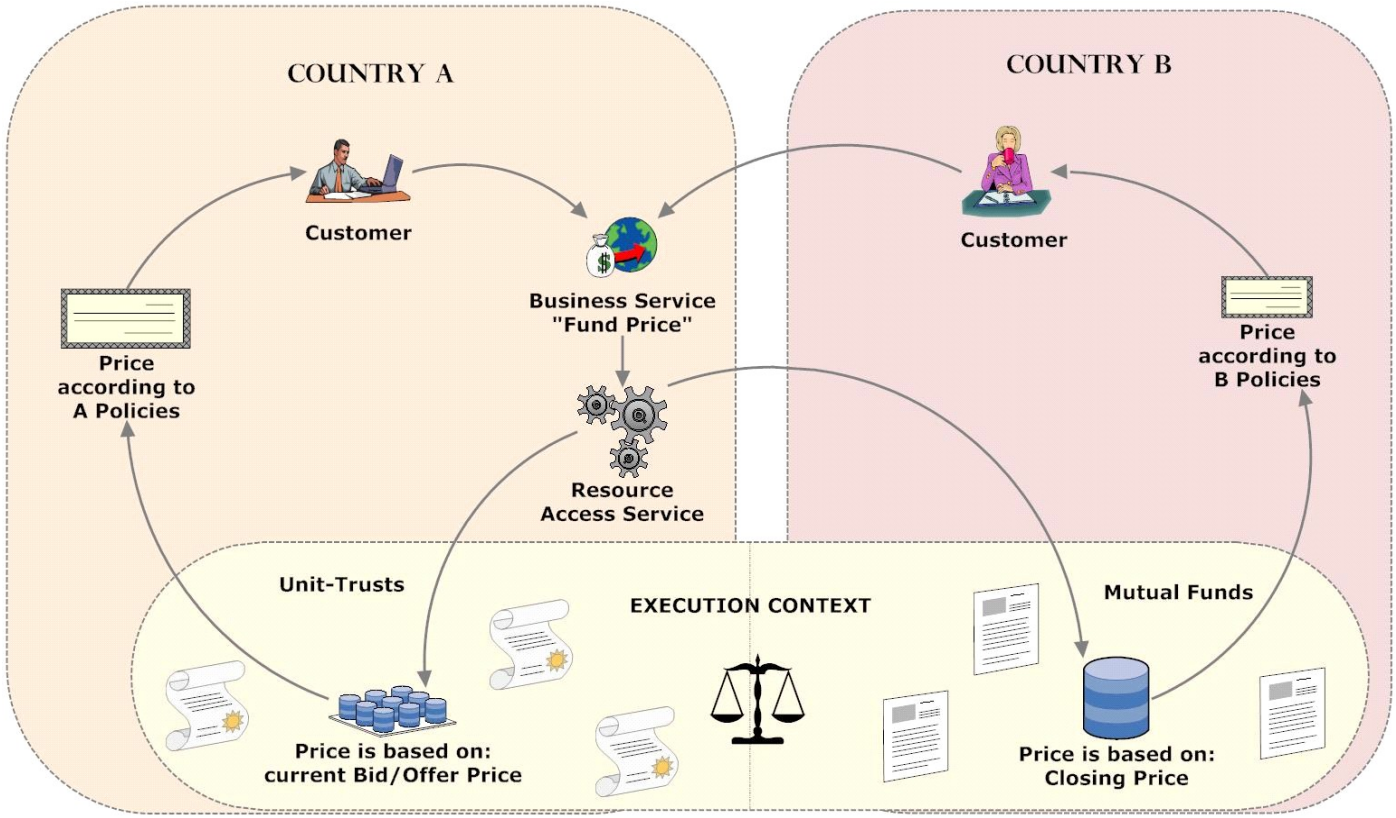
★ Service-Orientation Design Principles

- **Service Composability:** helps providing for the most important mechanism of adaptability to changes - via flexible service compositions
- **Service Autonomy:** helps to define the level of business functionality that should handle the change by itself, internally
- **Standardized Service Contracts:** helps in the announcement of changes in the service functionality, in the service Real World Effect (result), or in the Execution Context
- **Service Abstraction:** helps to adopt changes in the service functionality or in the service Real World Effect
- **Service Loose Coupling:** helps to adopt change in the Execution Context and service body (implementation)
- **Service Reusability:** helps to accommodate a spectrum of changes via 'reuse by extension'
- **Service Discoverability:** helps to support multiple versions of the service compositions

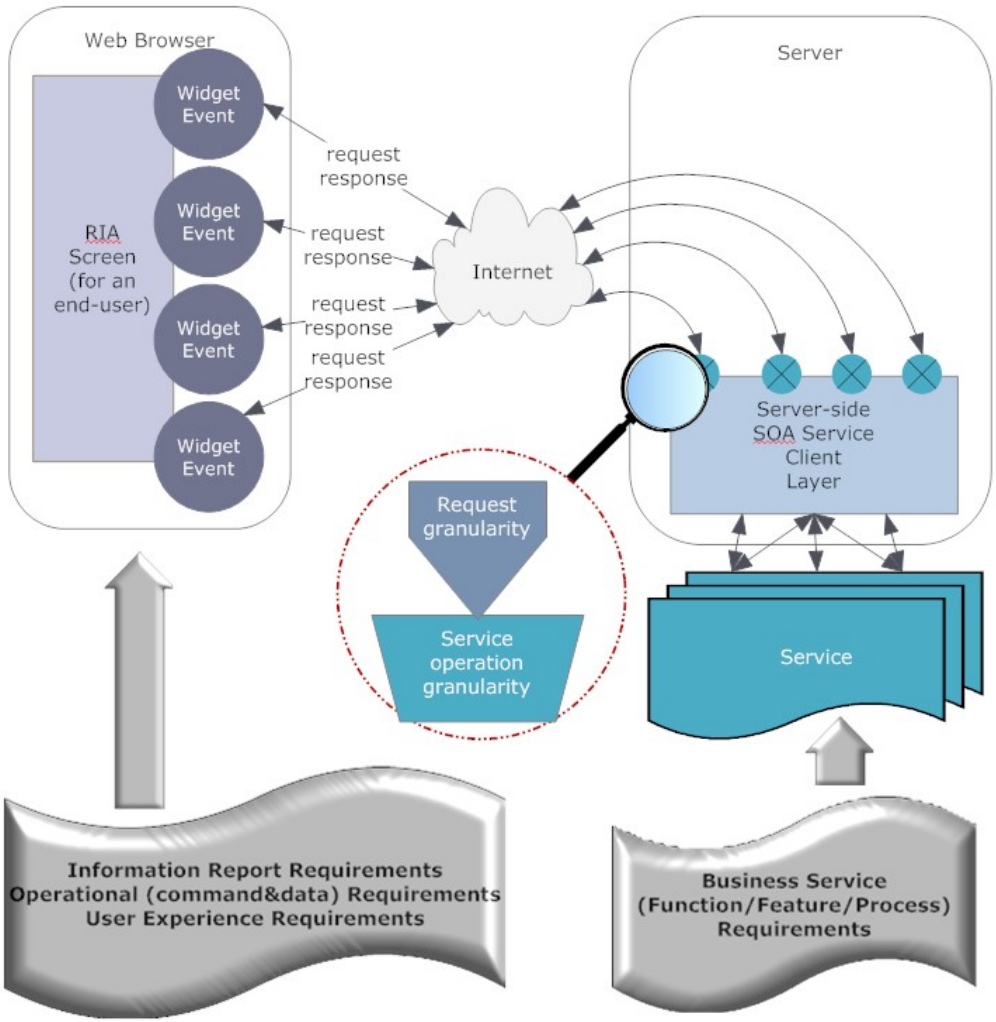
Change in service Execution Context



Service Execution Context is a set of technical and business infrastructure elements, process entities, policy assertions, and agreements that forms a path between those with needs and those with capabilities

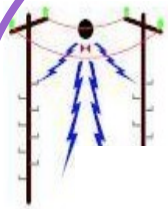


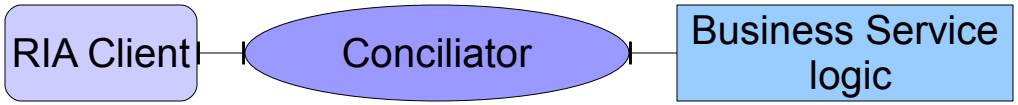
UI for Business Service
(change in User Experience vs. change in Business Logic)

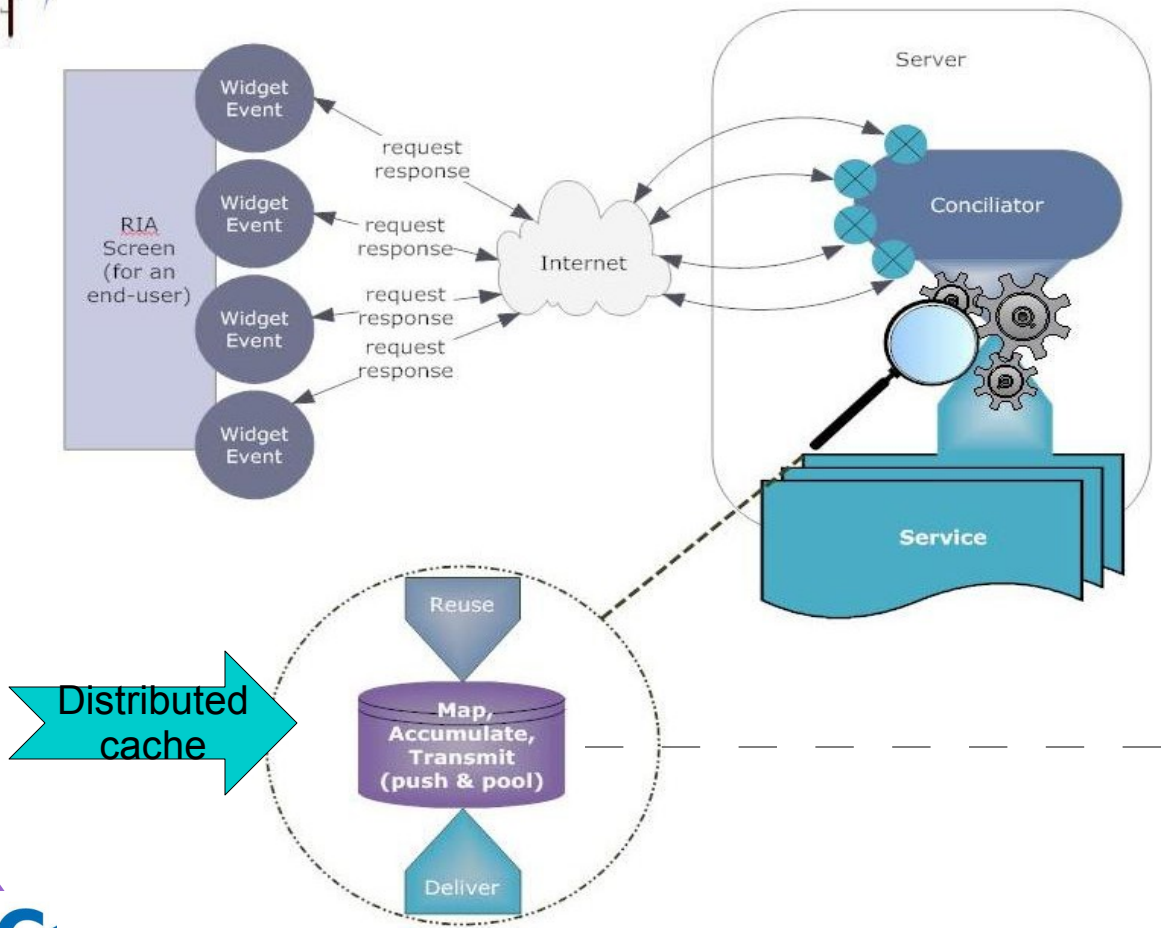


- The major mismatch between RIA and SOA is in
  - the fine-grained operations in RIA
  - the coarse-grained operations of business SOA services

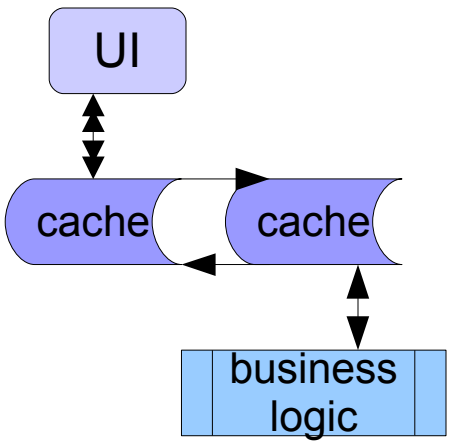
UI for Business Service - 2





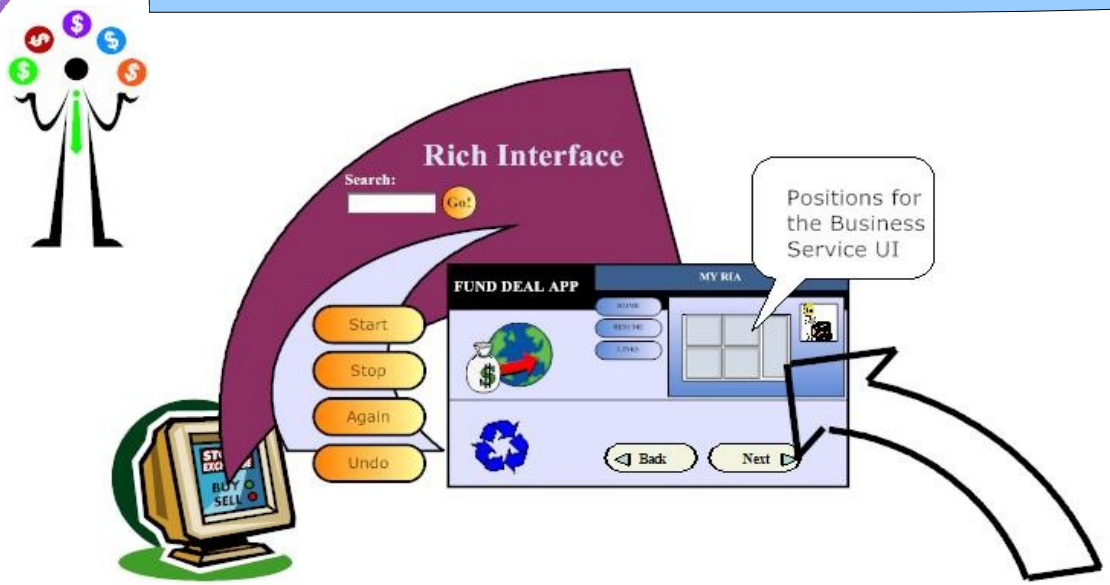


- Collaboration-Conciliator Pattern:
  - balances functionality and granularity
  - transforms data



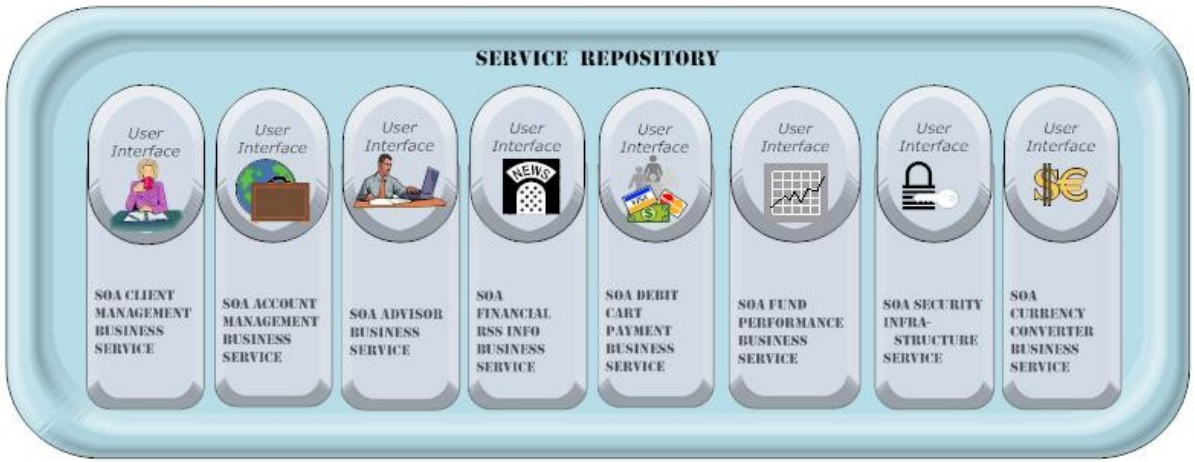
http://www.soapatterns.org/ui_mediator.asp

UI for Business Service - 3



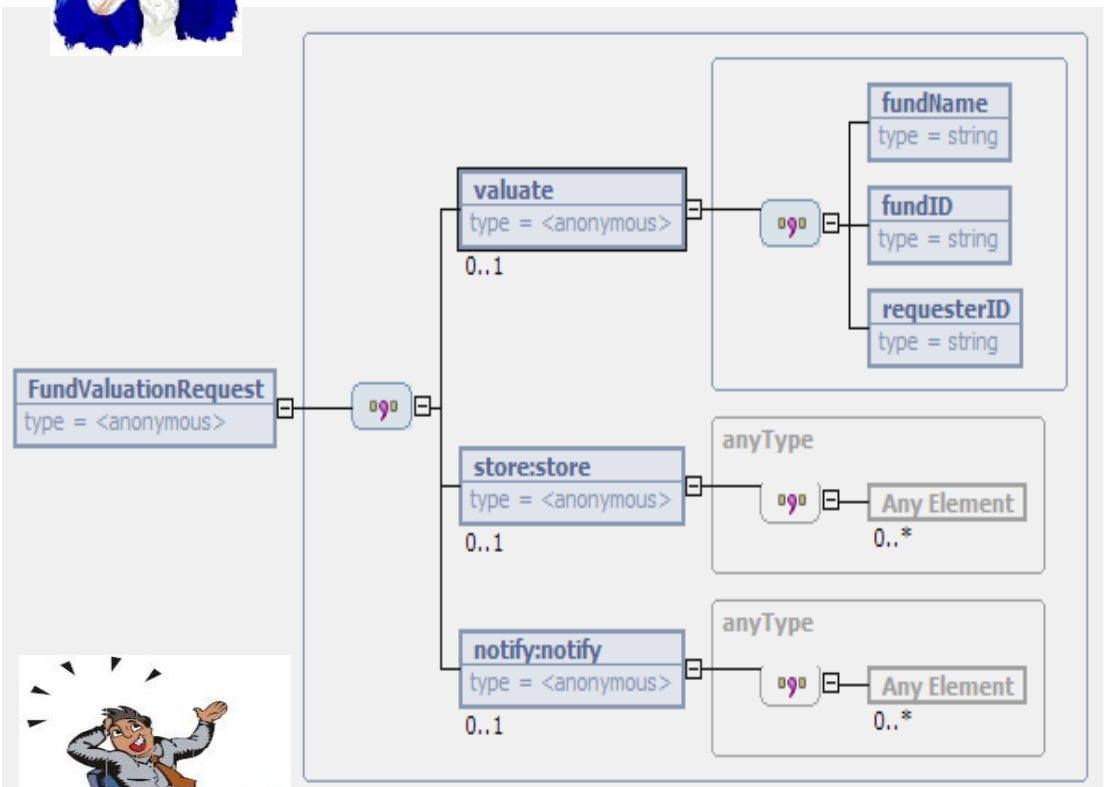
- Composite or Aggregate Service: RIA Client is an explicit aggregation of the UIs of Business Services

- Business Process (User Journey): RIA Client's UI is combined with the UIs of used Business Services



Change Handling via Service Reuse

- I do not want to hear about a new service version if I did not ask for it



- Do not worry, you will not see it until you ask for it

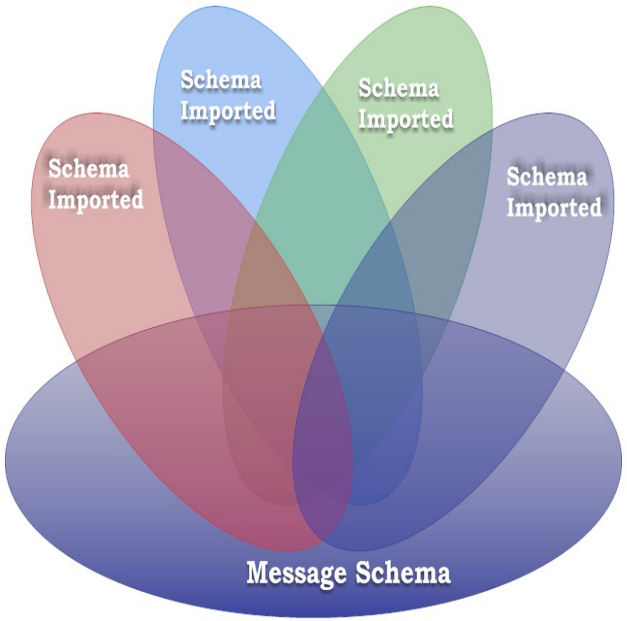
Reuse 'by extension' – keeping users happy

Reuse '*as is*' vs. reuse '*by extension*'

- 'as is': minimum service flexibility but the simplest
- 'by extension': high level of service flexibility but not trivial

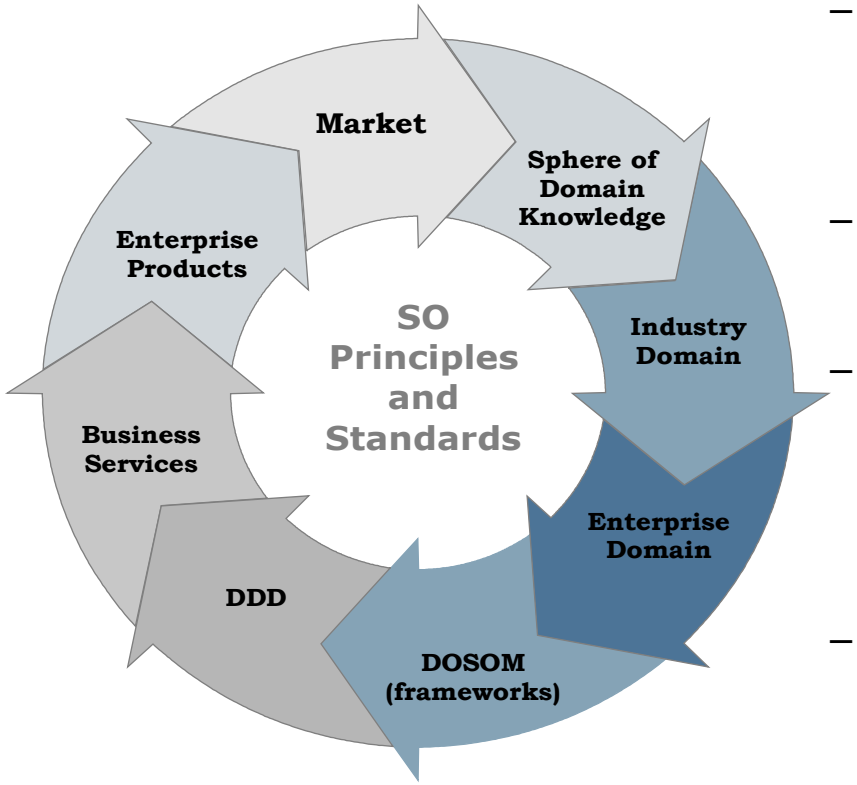
How to reuse service 'by extension':

- Define *operations* of the service interface
- Define in/out messages for the *operations*
- Separate consumer activities from the *operations*
- Specify each activity within its own namespace or schema and import it into the message
- Define each activity as optional (*minOccurs="0"*)



With reuse 'by extension', we can extend the *messages* by adding/removing activities as needed preserving backward compatibility for existing users

Domain Service-Oriented Modelling



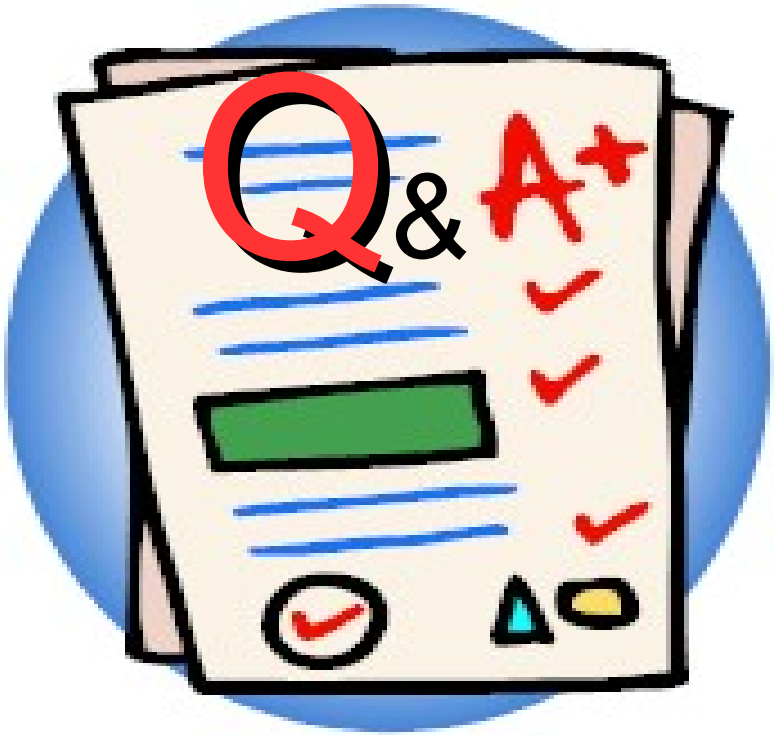
DOSOM© :

- a combination of Domain-Driven Design and Model-Driven Architecture (MDA) in the sphere of Service Orientation
- a domain-specific model that preserves service-oriented principles
- a domain-agnostic approach that targets domain-specific business tasks at the model level with no technical constraints for the model realisation
- the form of a seamless stream of inheritable Domain Models within boundaries of Business Services

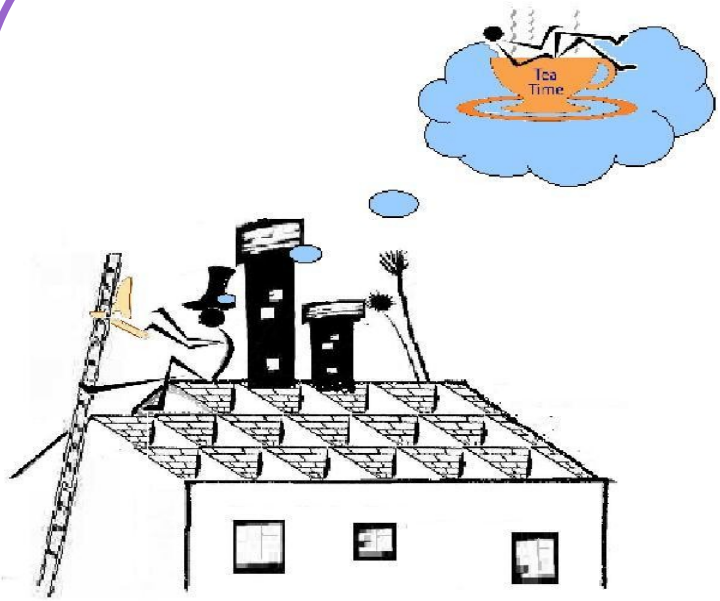
Food for Thought

- Service Orientation is a solution for gaining maximum efficiency in the market through collaboration between Business and Technology
- A Business Service is a Service, which realises business task, feature, function or business process, or a combination of them
- Services ought to be designed for changes
- Service collaboration is the instrument for change adoption with minimal investments, efforts, and time-to-market
- Service behavior depends on the Execution Context
- Domain Service-Oriented Modelling is the way for business-oriented service design

At Your Service....



My Publications



- Sys-Con Media: http://michaelpoulin.sys-con.com
- ebizQ, BLOG - Service-Oriented Solutions: http://www.ebizq.net/blogs/service_oriented

Book-in-printing: **Ladder to SOE**