# Service Oriented Architecture

## Impact on Information Quality

# John Walsh - Personal


**GROUP 1 software**


# Key Concepts

Software functionality is a re-usable service that can be discovered and accessed

Re-usable software services are “loosely coupled”

- Published interface with defined I/O that can be easily accessed and utilized by software that adheres to the interface

# How does SOA differ?

Previous approaches to integration of applications and information quality projects and programs were:

1. Local
2. ETL & EAI
3. DCOM & CORBA

# Local Integrations

Data quality tools, applications, and data are all on the same server

Data quality tools are “tightly integrated” into applications using code

- Tight integration: interface, communication layer and business logic all integrated

Not practical for enterprise deployments

No sharing of investments and expertise

Limited to capabilities of one data quality tool

# Traditional Middleware

Extract, Transform & Load (ETL) &
Enterprise Application Integration (EAI)

Files and transactions are sent to single server(s) where processes are designed, tested and implemented using suite of vendor's products

Limited to capabilities of "tightly" integrated data quality tools

Not designed to leverage use of other investments

# DCOM, CORBA

Distributed, heterogeneous application integration:

- Distributed Common Object Model (DCOM)
- Common Object Request Broker Architecture (CORBA)

Data can be sent to data quality applications on different servers

Applications had to be “tightly” integrated

- **Lots of coding, specific skill levels, difficult to modify**

Peer-to-peer, not process oriented

Not designed for re-use (hard wired)

# Comparison of Integration Approaches

| | Local | Middleware | DCOM | CORBA |
|---|---|---|---|---|
| **Type of Coupling** | Tight | Tight | Tight | Tight |
| **Integration Constraints** | Language/Server<br>Custom Adapters | Location of Products<br>Custom Adapters | Windows<br>Custom Adapters | IDL<br>Custom Adapters |
| **Resources: Skills & Knowledge**<br>**•Application**<br>**•Platform**<br>**•Languages** | <br>High<br>High<br>High | <br>High<br>Moderate<br>Moderate | <br>High<br>High<br>High | <br>High<br>High<br>High |
| **Types of Integration** | Direct | Centralized | Peer-to-Peer | Peer-to-Peer |
| **Processing Overhead** | Minimal | Minimal to High | Minimal | Minimal |
| **Cross Integration Efforts** | Not Available | Stand Alone | High | High |
| **Change Management** | High | Moderate | High | High |
| **Product (s)** | One | Suite of Software | Windows + 1 - N | IDL, OMG + 1 - N |
| **Training** | High | High to Moderate | High | High |

# Comparison of Integration Approaches

| | Middleware | DCOM | CORBA | SOA –Web Services |
|---|---|---|---|---|
| **Type of Coupling** | Tight | Tight | Tight | Loose |
| **Integration Constraints** | Proprietary Software<br>Custom Adapters | Windows<br>Custom Adapters | IDL<br>Custom Adapters | SOAP/HTTP/XML<br>Standard Adapters |
| **Resources: Skills & Knowledge**<br>**•Application**<br>**•Platform**<br>**•Languages** | <br>Moderate<br>Moderate<br>Moderate | <br>High<br>High<br>High | <br>High<br>High<br>High | <br>Minimal<br>Minimal<br>Moderate |
| **Type of Integration** | Process Oriented – Centralized | Peer-to-Peer | Peer-to Peer | Process Oriented – Distributed |
| **Processing Overhead** | High to Minimal | Minimal | Minimal | **High** |
| **Cross Integration Efforts** | High – Stand Alone | High | High | Moderate |
| **Change Management** | Moderate | High | High | Moderate |
| **Product (s)** | Suite of Software – One Vendor | Windows + 1-N | IDL,OMG + 1-N | **5-20 WS + 1-N** |
| **Training** | High to Moderate | High | High | High |

# Generic Benefits to SOA

- Faster integrations – Higher Productivity
- Faster projects – Greater Business Agility
- Operational efficiencies
  - Avoid duplication of effort, software licenses

# Specific SOA Benefits for IQ

Build a corporate library of shared software functionality and processes

- Track, monitor and report on results
  - Feedback loop for best practices
- IQ domain experts to collaborate
- Balance between departmental needs and corporate standards

# Why is SOA Important?

Provides for departmental and enterprise information quality utilizing all of the diverse vendor and in-house technologies

# Optimized Service Oriented Architecture (OSOA™)

- Address the weaknesses in SOA-WS
  - High Processing Overhead
  - High Total Ownership Costs
    - Large number of products
    - Acquisition Costs
    - Training
    - Maintenance
  - Standards Only Approach
    - Not applicable for all integration

# Comparison of SOA & OSOA™

| | SOA – Web Services | OSOA™ - ROME® |
|---|---|---|
| **Type of Coupling** | Loose | Loose |
| **Integration Constraints** | SOAP/HTTP/XML<br>Standard Adapters | **None - Open** |
| **Resources: Skills & Knowledge**<br>**•Application**<br>**•Platform**<br>**•Languages** | <br>Minimal<br>Minimal<br>Moderate | <br>Minimal<br>Minimal<br>**Minimal** |
| **Type of Integration** | Process Oriented | Process Oriented |
| **Processing Overhead** | High | **Minimal** |
| **Cross Integration Efforts** | Moderate | **Minimal** |
| **Change Management** | Moderate | **Minimal** |
| **Product (s)** | 5 -20 WS + 1-N | **One + 1-N** |
| **Training** | High | **Minimal** |

# Summary

SOA has many advantages over previous approaches integration of applications and information quality projects and programs.

OSOA™ offers even more!