Information Quality & Service Oriented Architecture

Presentation for the MIT IQ Industry Symposium

July 17, 2007

Dave Becker

The MITRE Corporation

MITRE

Problem

- SOA is a new paradigm for how applications and systems are delivered and operated

- Questions:
  - How should IQ be viewed as a set of services in an SOA?
  - How would IQ services be constructed and operate?

- Approach:
  - Identify distinctive features of SOA as a design style
  - Identify distinctive features of The Architecture of Data Quality
  - Select an architecture paradigm for which SOA is ideally suited, and for which DQ is a natural application
  - Characterize features of The Architecture of DQ as services in the target SOA problem area that meet the objectives of SOA as a design style

SOA as a Design Style

What is Service Oriented Architecture (SOA)?

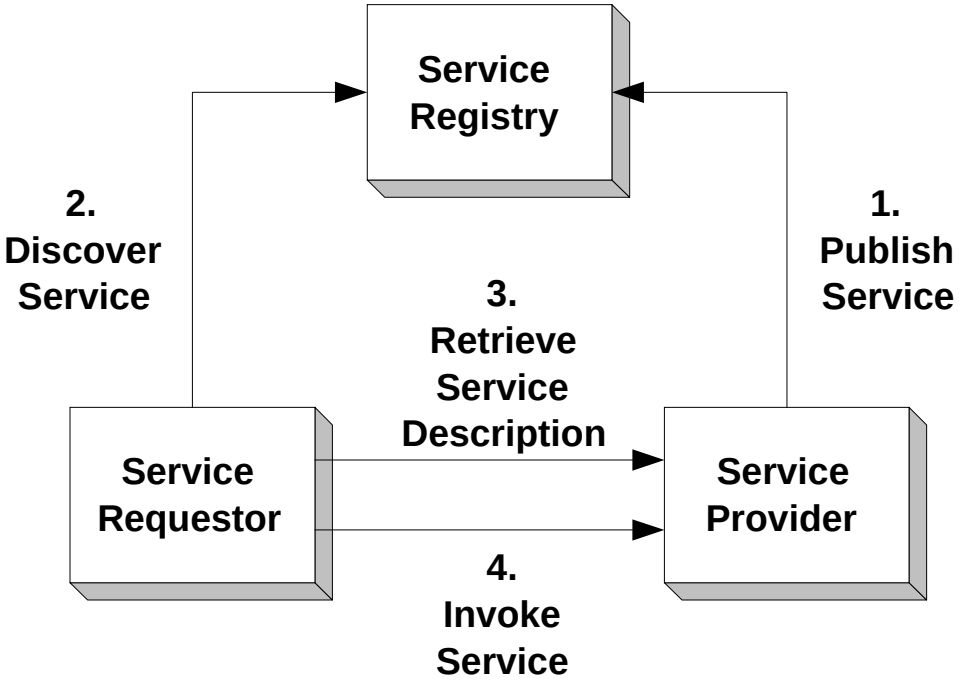
Some Definitions: [1]

- Service – A unit of work done by a service provider to achieve desired end results for a service consumer
- Software Agents – Both service provider and service consumer are roles played by software agents on behalf of their owners”
- SOA – An architectural style whose goal is to achieve the minimum amount of dependency between the software agents acting as service providers and service consumers.

Web Services as a Specialization of SOA

Traditional Basic Web Services Components

- Service Provider
  - Implements service
  - Publishes availability
  - Makes available on internet
  - Services requests
- Service Requestor
  - Discovers web service
  - Retrieves description
  - Sends request
  - Processes response
- Service Registry
  - Houses services descriptions
  - Adds new web services
  - Finds existing services



Some Key SOA Architectural Characteristics

- Separation of Concerns – the process of breaking the primary objectives of an application or capability into distinct features that overlap in functionality as little as possible. Consuming a provided service is usually cheaper and more effective than doing the service

- Modularity – the quality of an application whereby it is composed of individually distinguishable pieces or parts, each of which performs some particular function or concern, and all of which operate in concert to achieve the higher level objectives of the application

- Loose Coupling – the attribute of two or more collaborating components where there are a minimal number of dependencies between the components such that changes is one would affect the operation other

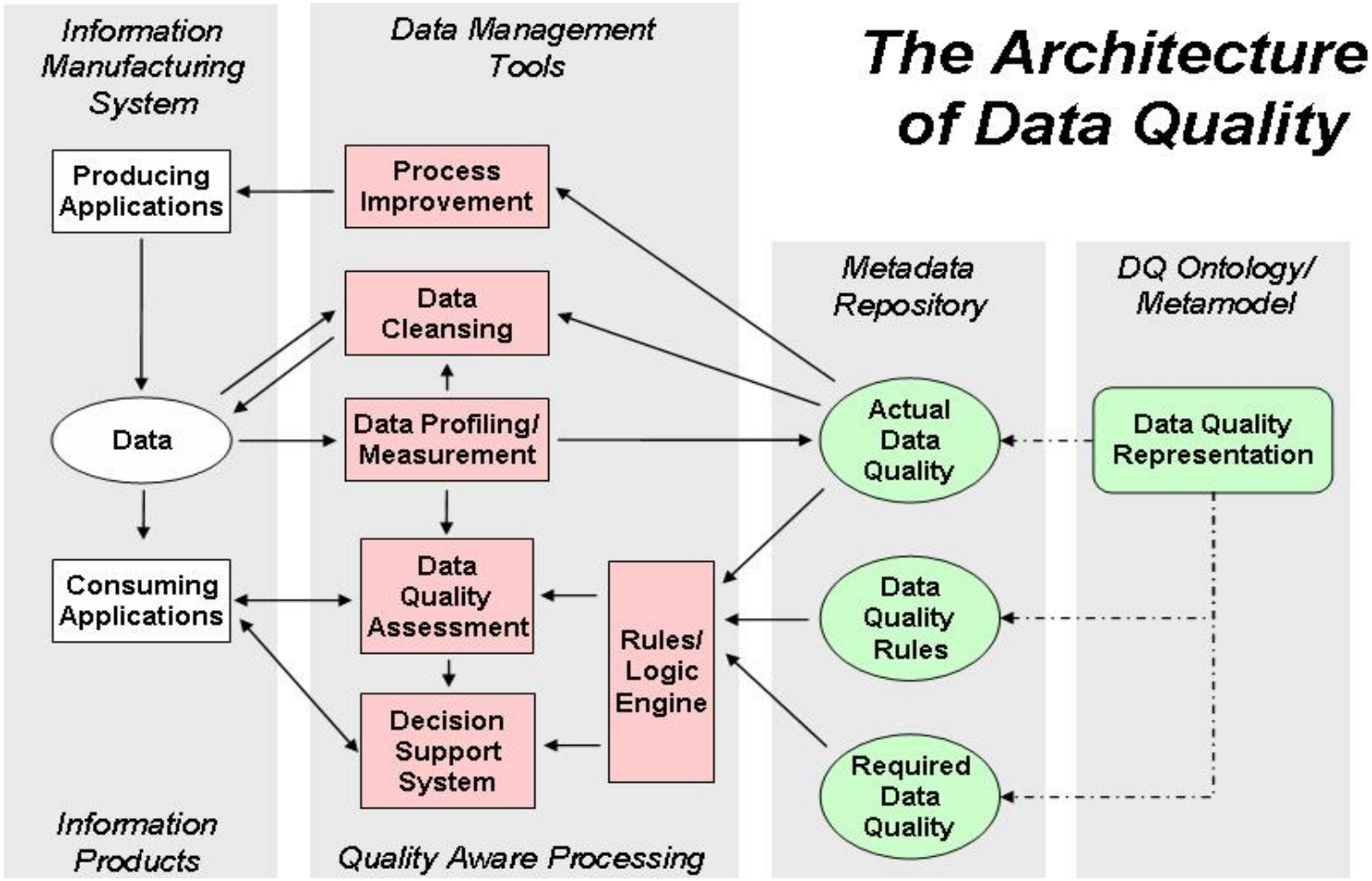
- Encapsulation – the combination of data with the instructions for manipulating the data into a single package that can be accessed through a separate interface

Some Key SOA Architectural Characteristics

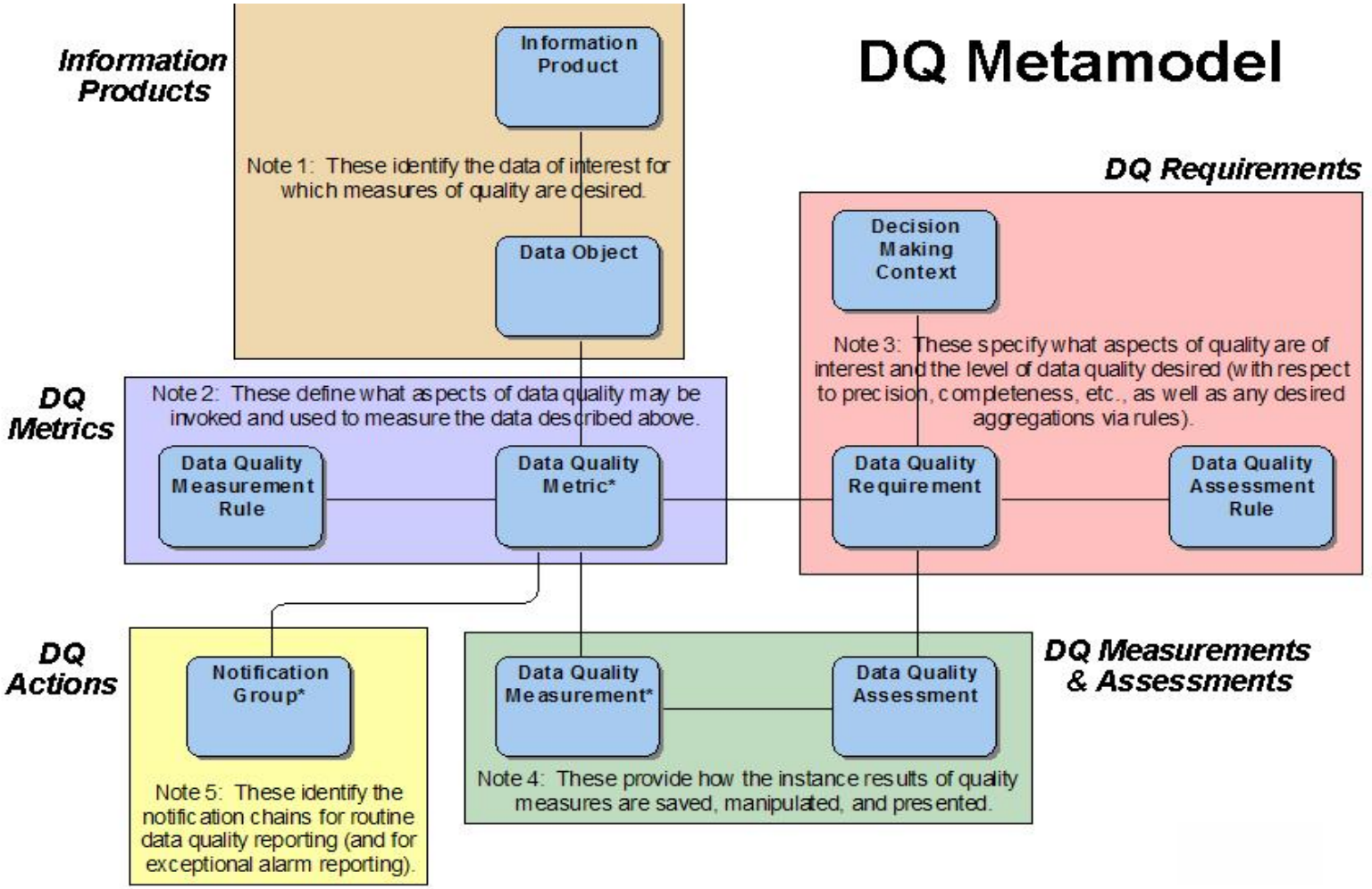
- Interfaces – entry points that provides access to a system, and prescribes the system's behavior. SOA implements a small set of simple interfaces maintained separately from the implementation and available for discovery.
- Messages – descriptive messages containing information to be exchanged whose structure and vocabulary are constrained by an extensible schema delivered through the interfaces
- Reuse – the act of using a component over and over again to accomplish the higher level objectives of an application or capability
- Composability – the ability to select components and assemble them in various ways that can meet the objectives of an application or capability

The Architecture of Data Quality

High-Level Systems Architecture



High-Level Data Architecture



DQ Architectural Features

- Information Manufacturing System
  - Consists of producing applications which generate data in the form of information products which are consumed by other applications
  - All DQ processing is a side activity to the mainline information manufacturing system and should be dealt with as an overhead activity
- COTS – Use commercial off-the-shelf tools as much as possible to implement components of the architecture
  - Data profiling
  - DQ measurement and assessment
  - DQ management & quality aware processing
  - Systems and data are often tightly coupled and proprietary
- Legacy System Error Processing
  - Many legacy systems generate error reports & files
  - These can be used to provide basic DQ measurement information

DQ Architectural Features

- Shared Information Space
  - COTS and legacy systems can expose the data profiling and data quality measurement and assessment information (DQ data) they generate
  - DQ data can then be captured in a publicly accessible, shared information space

- Metadata – Data about data
  - DQ data is metadata & DQ management is metadata management
  - A shared information space storing metadata is called a metadata repository (MDR)
  - Structure and vocabulary of a database is usually specified in an ontology or data model
  - Structure and vocabulary of metadata can be defined in a metadata model (metamodel)
  - Since a data model can be used to automatically generate a database, a metamodel can be used to automatically generate an MDR

DQ Architectural Features

- Types of Data Quality – There are multiple DQ dimensions which must be defined, measured separately, aggregated as necessary and stored in the MDR
  - Accuracy
  - Precision/Certainty
  - Completeness/Brevity
  - Consistency/Validity
  - Timeliness
  - Lineage/Pedigree
  - Community Feedback / Confidence
  - Others
    - As needed
    - As metrics and measurement tools can be defined
- Business Rules – definitions, operations and constraints that must be applied for a business to achieve its goals
  - Database constraints and domain violations
  - DQ metric calculations
  - DQ assessment computations
  - Business rules need to be defined and managed external to software as much as possible where they can be processed and applied by emerging rules engine technology

DQ Architectural Features

- DQ Measurement vs DQ Assessment
  - DQ measurement identifies characteristics intrinsic to the data
  - DQ requirements are intrinsic to the usage context
  - DQ Assessment compares DQ measurements to DQ requirements
  - Multiple assessments possible for a single set of measurements
- Multiple-Use - The same DQ Metadata can be used for many purposes
  - Data Cleansing
  - Data Quality Improvement
  - Process Management
  - Continuous Process Improvement (Six Sigma)
  - Analytics & Business Intelligence
  - Decision Support

DQ Architectural Features

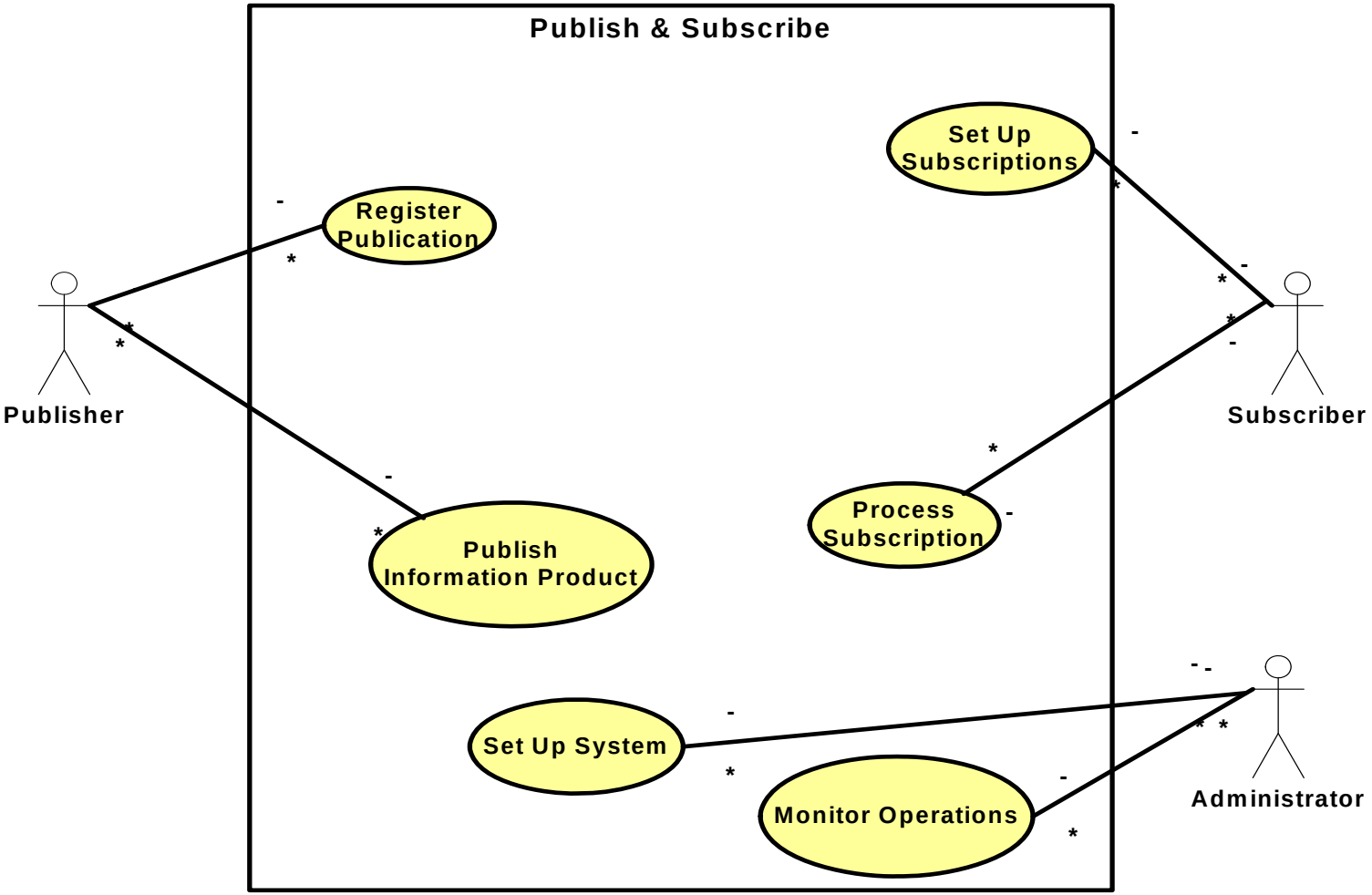
- DQ Actions
  - Use the data as is
  - Send out notifications
  - Obtain corroboration from other sources
  - Exclude bad data
  - Clean up bad data
  - Correct the current data operations
  - Correct the process & data
  - Change the subsequent processing steps
  - Include quality assessments in the downstream processes
  - …

Publish & Subscribe – A Type of SOA Architecture

Publish & Subscribe – Architectural Paradigm

- As an alternative to multiple point-to-point services
- Data is published one time
  - Data is published using a publication service
  - Data is published to a shared information space
- Users can discover and set up a subscription to published data
- As data is published, the system matches published data to active subscriptions
  - Matches will cause either the data to be pushed to a subscriber, or a notification of the availability of the data to be sent to the subscriber

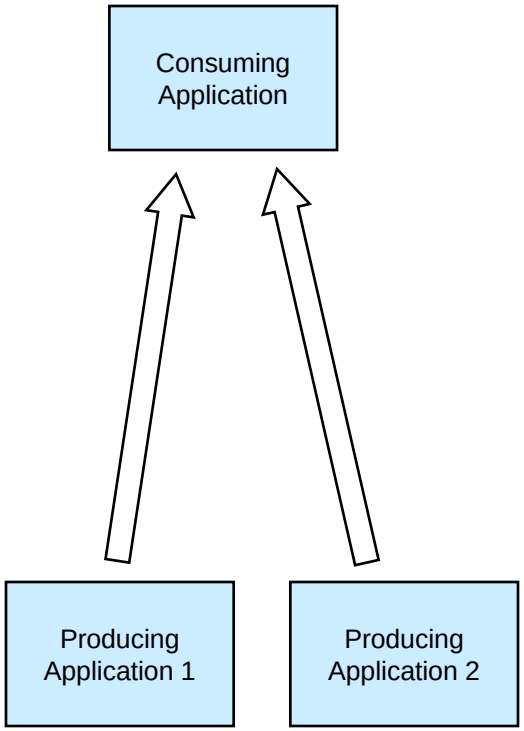
Publish & Subscribe – Use Case Diagram:



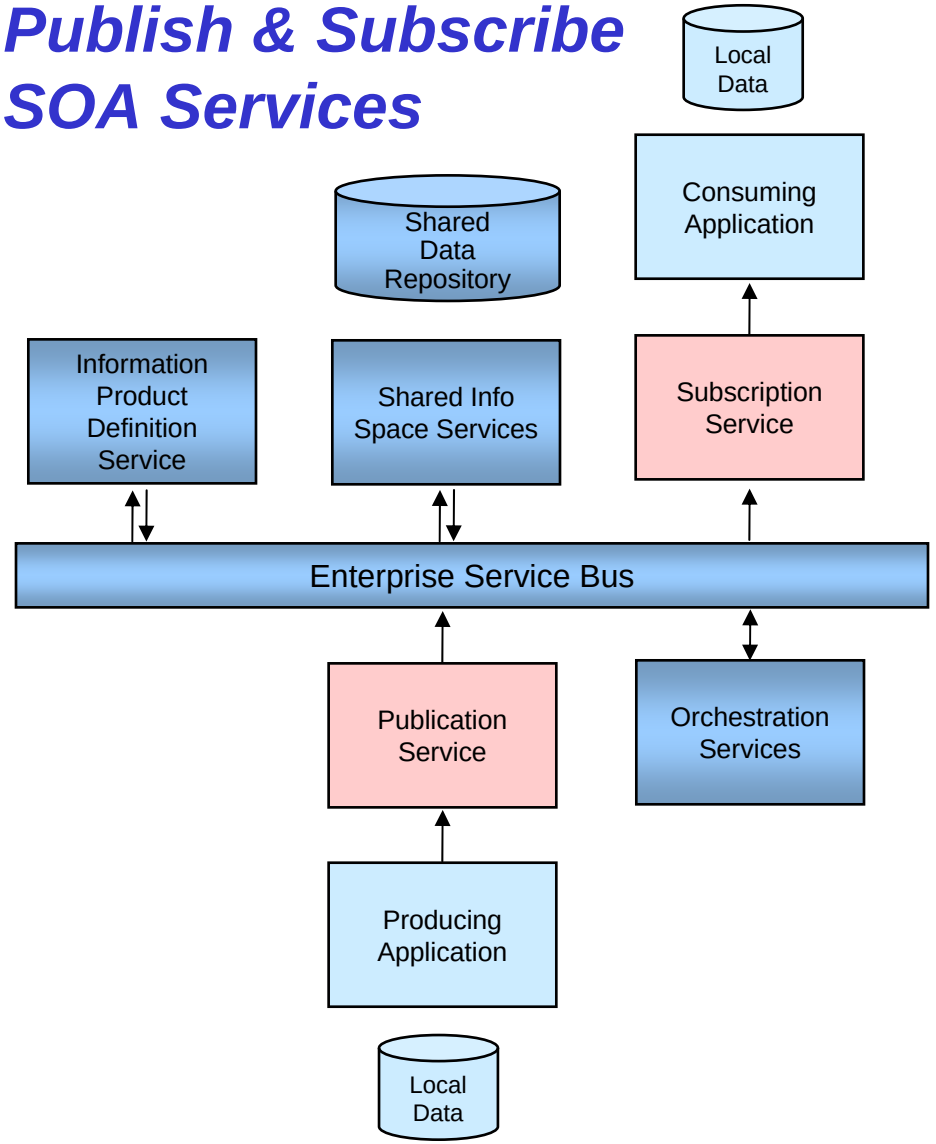
Use Cases:

1. Set Up Publication – Register a an information product as available for subscribers
2. Set Up Subscription – Set up a subscription to an available information product
3. Publish Information Product – Publish an information product to a shared information space
4. Process Subscription – Match up instances of published information products to subscriptions for that information product, and either push the data to the subscriber or send a notification of its availability
5. Set Up System – Register all services in a services registry, and set up the orchestrations to invoke them
6. Monitor Operations – Present a view of the current operating state of the system

As-Is Point-to-Point Interfaces



Publish & Subscribe SOA Services



MITRE

Publish & Subscribe – Web Services

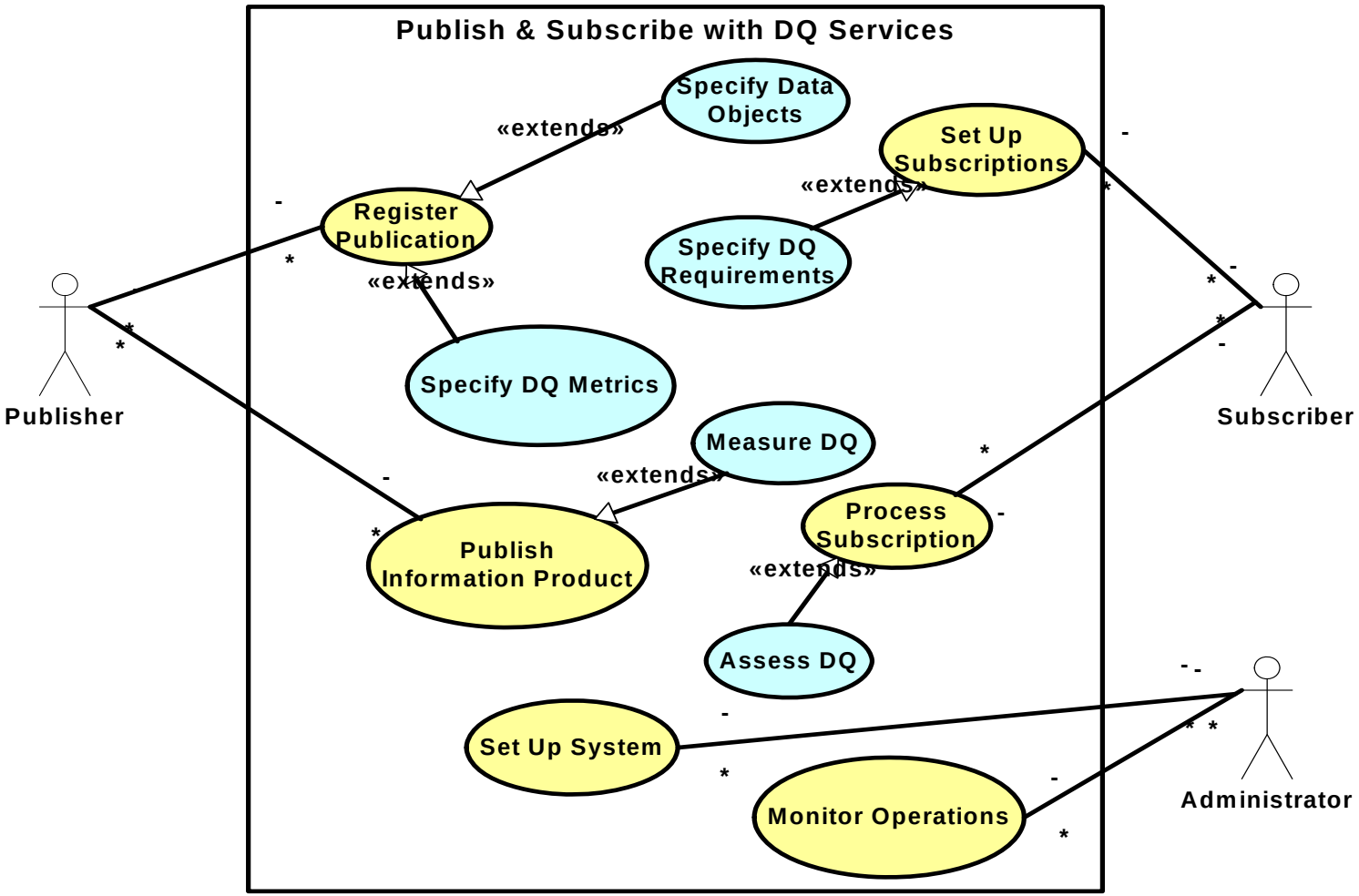
- Information Product Definition Service
- Publication Service
- Subscription Service
- Infrastructure Services
  - Shared Information Space Services
  - Orchestration Service
  - Enterprise Service Bus (ESB)

DQ Services in a SOA Publish & Subscribe Architecture

Publish & Subscribe with SOA Services

- Need to be able to specify data objects and their metrics when a publication is registered
- Need to specify usage contexts and the DQ requirements for those contexts when a subscription is set up
- Need to perform DQ Measurements on published data
- Need to perform a DQ assessment for each subscription to an instance of published data

Publish & Subscribe with DQ – Use Case Diagram :



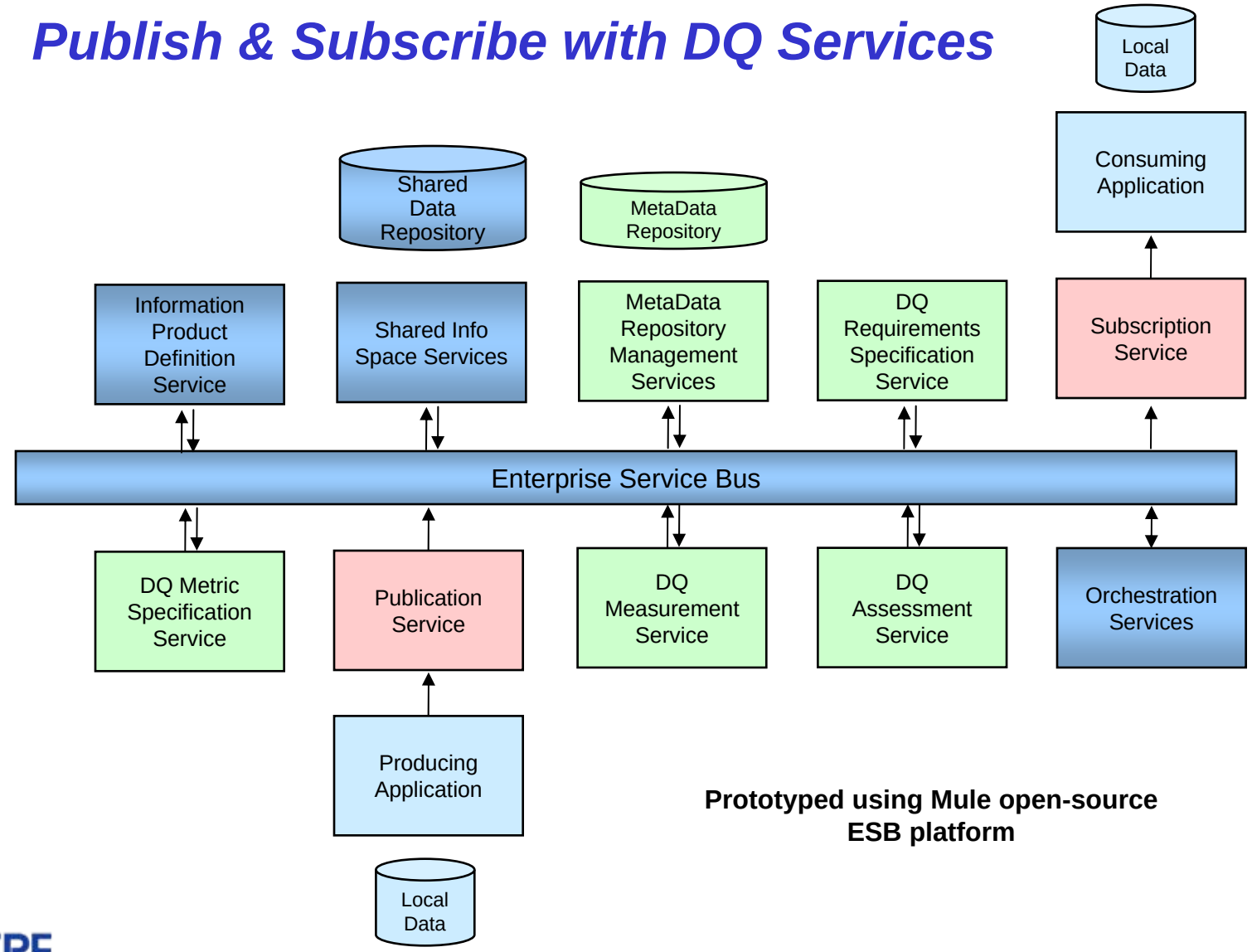
Use Cases:

1. Specify Data Objects – Specify the information products and the information manufacturing system including the data objects for which DQ must be measured
2. Specify DQ Metrics – Specify the relevant DQ metrics for each data object
3. Specify DQ Requirements – Whenever a subscription is set up (which constitutes a separate usage context), also specify the DQ requirements for each data object metric
4. Measure DQ – Whenever an information product is published, perform the necessary DQ profiling and measurement for each data object metric
5. Assess DQ– When the system attempts to match up published data to subscriptions, compare the actual DQ as measured to the DQ requirements, and generate actions as necessary

Publish & Subscribe – DQ Aware Web Services

- Information Product Definition Service
- DQ Metric Definition Service
- DQ Measurement Services
- DQ Requirements Specification Service
- DQ Assessment Service
- Infrastructure Services
  - Metadata Repository Management Services
  - Orchestration Service (Reconfigured)

Publish & Subscribe with DQ Services



Prototyped using Mule open-source ESB platform

MITRE

Conclusions

Conclusions:

- SOA services can be defined for the major architectural objectives of DQ processing
- DQ services can be built and used in key SOA architectural paradigms
- The DQ services must be properly orchestrated with existing services to achieve the desired effects
- Aggregate services that implement composable applications might need to be modified to actively incorporate the DQ capabilities
- DQ services are an overhead activity, the costs and benefits of which must be evaluated for each usage situation

References

References:

1. “What Is Service-Oriented Architecture”, Hao He, September 30, 2003, O’Reilly Media Inc., “http://webservices.xml.com”
2. Wikipedia, Wikimedia Foundation, Inc.