Enterprise Asset Management Proposal

Created by Jason Azim

Table of Contents

S.No	Slide Name	Slide No.
1.	Cover Slide	1
2.	Table of Contents	2
3.	Objective	3
4.	Enterprise Asset Management - Overview	4
5.	Design Choices	5
6.	Case Study	6
7.	Custom Solution – Physical Architecture	7
8.	EAM – Application Proposal	8 - 13
11.	Chain of Responsibility	14
12	Managed Extensibility Framework (MEF)	15
13	ESG Integration	16 - 23
15.	References	24
16.	Questions	25

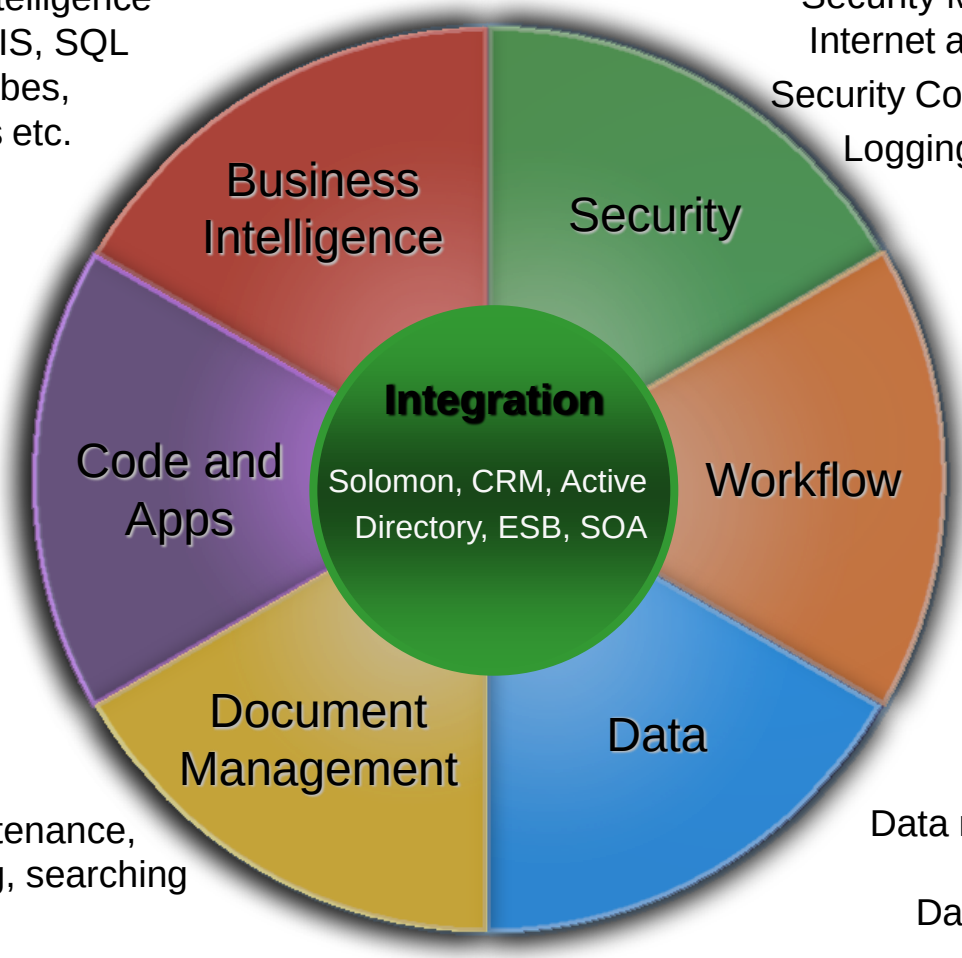
Objective:

1. Process improvement.
2. Integration with cloud.
3. Develop Prototype.
4. Evaluate third party tools and products.

Enterprise Asset Management (EAM) – Solution Overview

Business Intelligence includes SSIS, SQL Reports, Cubes, Dashboards etc.

Security Manager, Encryption, Internet and extranet security, Security Components and tools. Logging & Exception Mgmt.



Source Code, Application inventory, Deployment, process improvement.

Windows Workflow, Task Manager, Queue, Cache and orchestration.

Document maintenance, routing, indexing, searching and processing.

Data modeling and design, Data Dependency, Data Interface mapping.

Design Choices

Solution	Effort (Complexity)	Cost
Enterprise Service Bus (ESB)	At least 4 Months	High
Business Process Manager (BPM)	3 Months	Medium
Custom Solution – Open Source	5 Months	Low

Note:- None of these solutions is a zero custom code solution. They all involve some coding. Also the installation of the systems is fast but configuring and re-mapping data sources is time consuming.

EAM Architecture Types

1) Enterprise Service Bus (ESB) based solution. ESB are open source or commercial. The word middleware means the same thing.

2) Business Process Manager (BPM) based solution. This solution is built using orchestrations and a task processing engine.

3) Custom solution. We can build our own solution using an open source software or Workflow engine that will be customized to fit our needs.

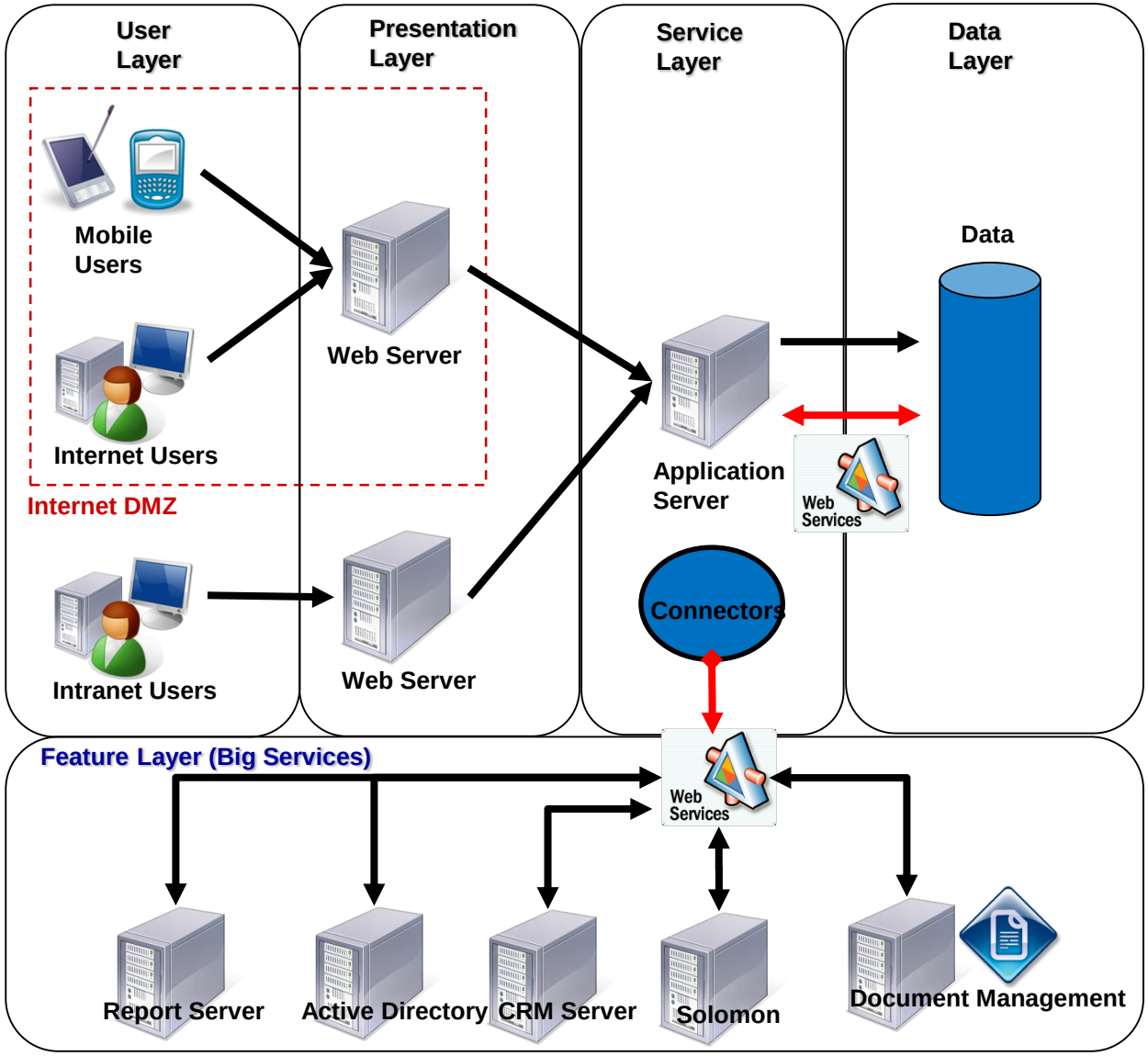
Case Studies (Listed in increasing Cost)

1. Enterprise Service Bus
    1. CenterPoint Energy (Maximo, FileNet, DataPower, CentraSite,TIBCO, SAP CRM, SQL Server)
    2. Toyota (IBM WebSphere, WebMethods, Oracle)
    3. BP (Business Objects, WebMethods)
    4. Spectra Energy (Maximo, FileNet, Sonic MQ, Historian, SQL Server)
    5. El Paso Energy (Maximo, LiveLink, Webmethods, SQL Server)
2. Business Process Manager
    1. Clear Channel Communications (Biztalk)
    2. Cameron (SAP, K2.NET)
    3. Harris County Hospital District (Innersystems, Oracle, SharePoint)
3. Custom Solution
    1. NACE – Custom (Microsoft CRM, SQL Server)
    2. IFCO Systems – Custom (Microsoft CRM, Solomon)

Evaluation Criteria and other Considerations:

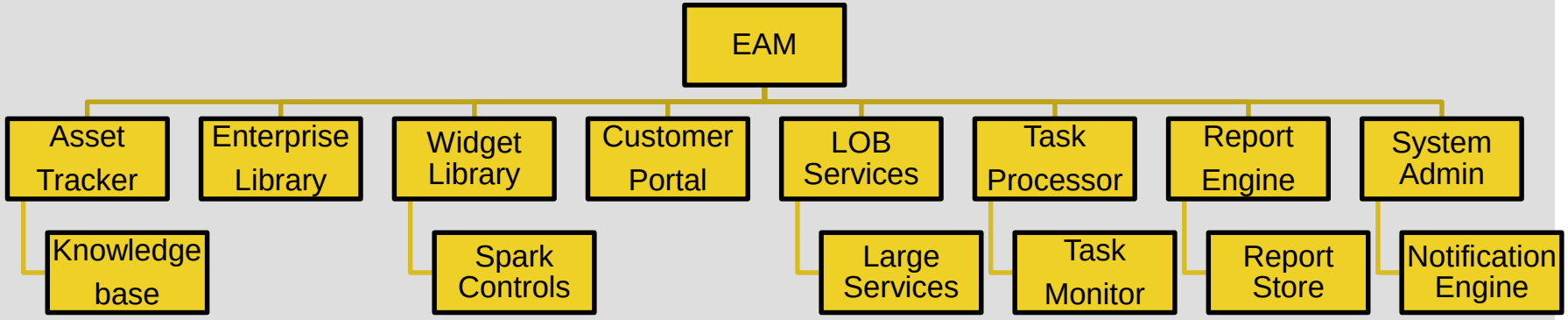
Vendors	Solution Type	Cost	Complexity (Effort)	Function
IBM Microsoft SAP Software AG Progress Software	1. ESB 2. BPM 3. Custom	1. High 2. Medium 3. Low	1. High 2. Medium 3. Low	Real Time Business App SCADA

Custom Solution - Physical Architecture

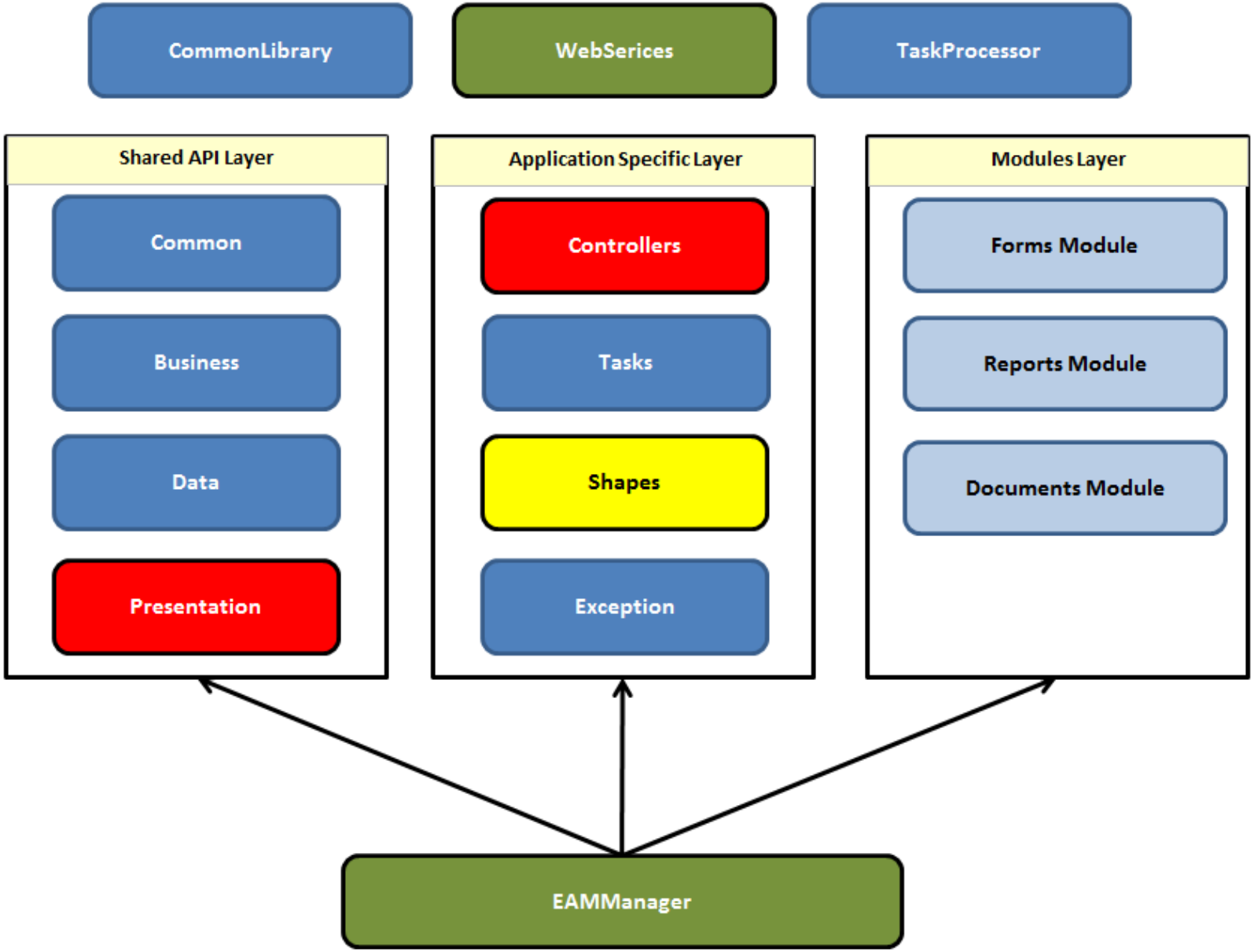


Custom Solution - Application Proposal

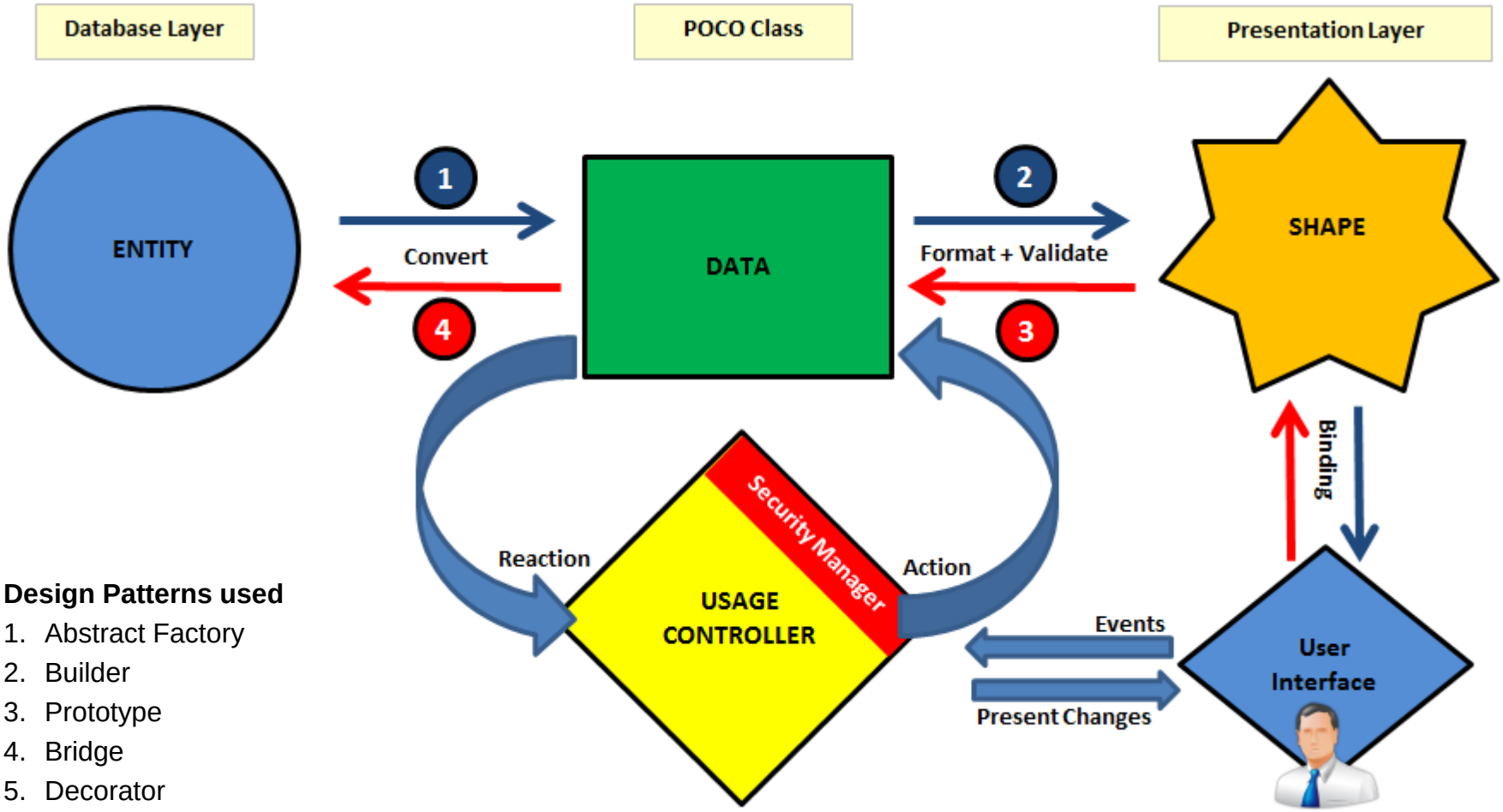
- AssetTracker – Track application, server, database etc. inventory.
- Knowledgebase – store documentation for systems and processes.
- Enterprise Library – Common Library of shared components.
- Customer Portal – Customer Extranet web site.
- LOB Services – Shared WebServices for applications. Big data services.
- Task Processor – Backend batch processing application.
- Task Monitor – Monitor health, integrity of systems. Compiles Alerts.
- ReportEngine – Store and run reports. Includes all BI stack.
- SystemAdmin – Provides Admin screens for Web, WebServices & Tasks.



Enterprise Library



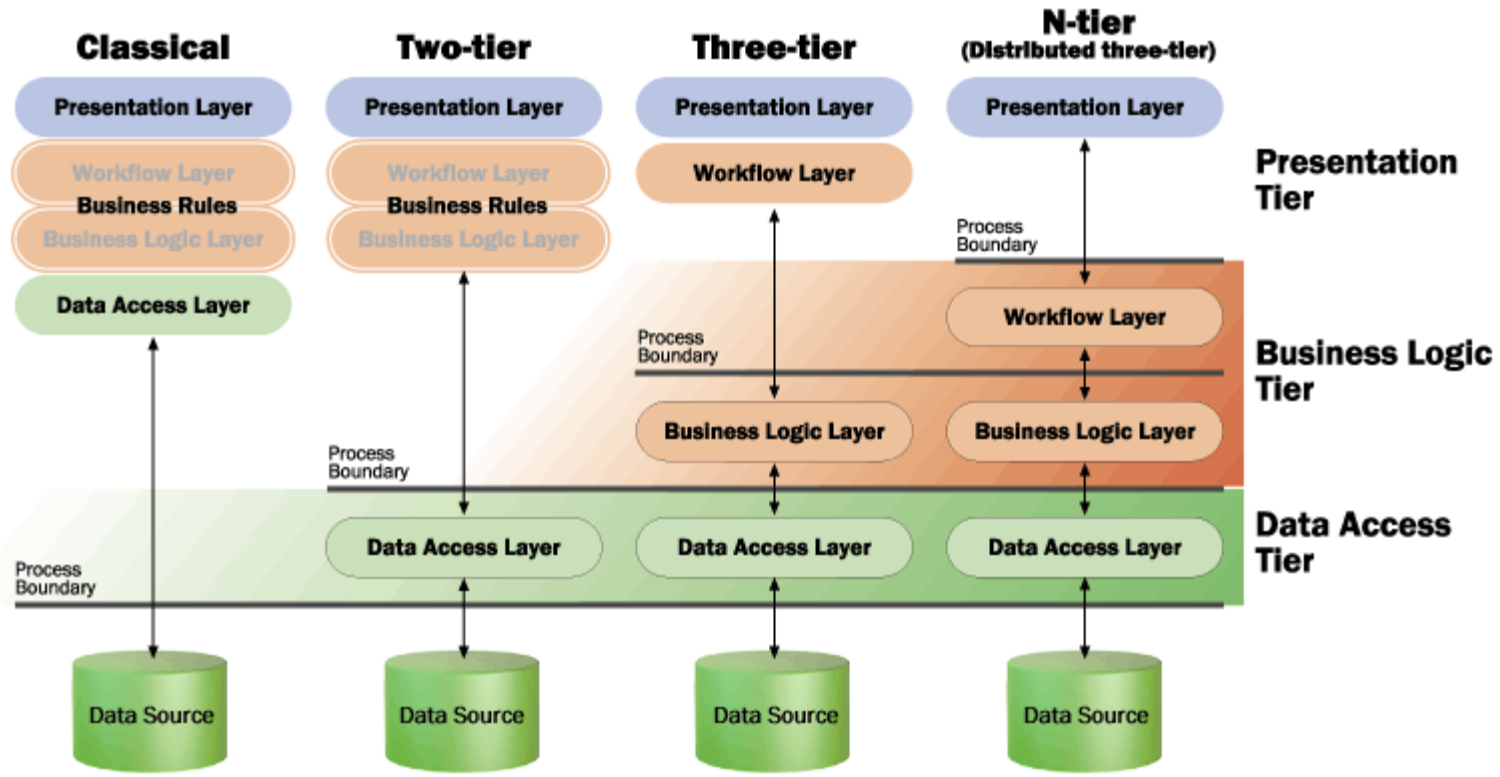
Customer Portal - Architecture



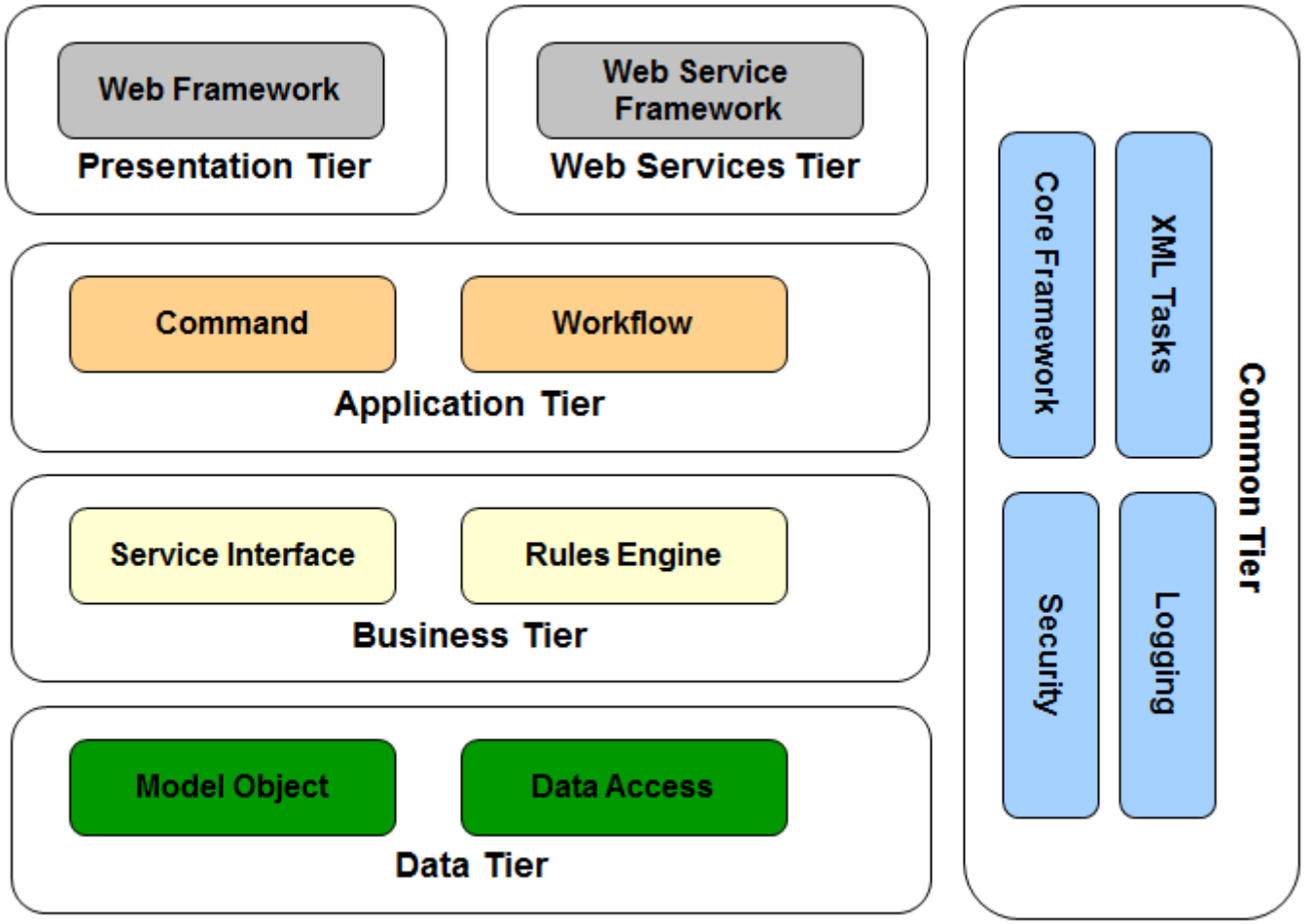
Design Patterns used

1. Abstract Factory
2. Builder
3. Prototype
4. Bridge
5. Decorator
6. Observer
7. Command

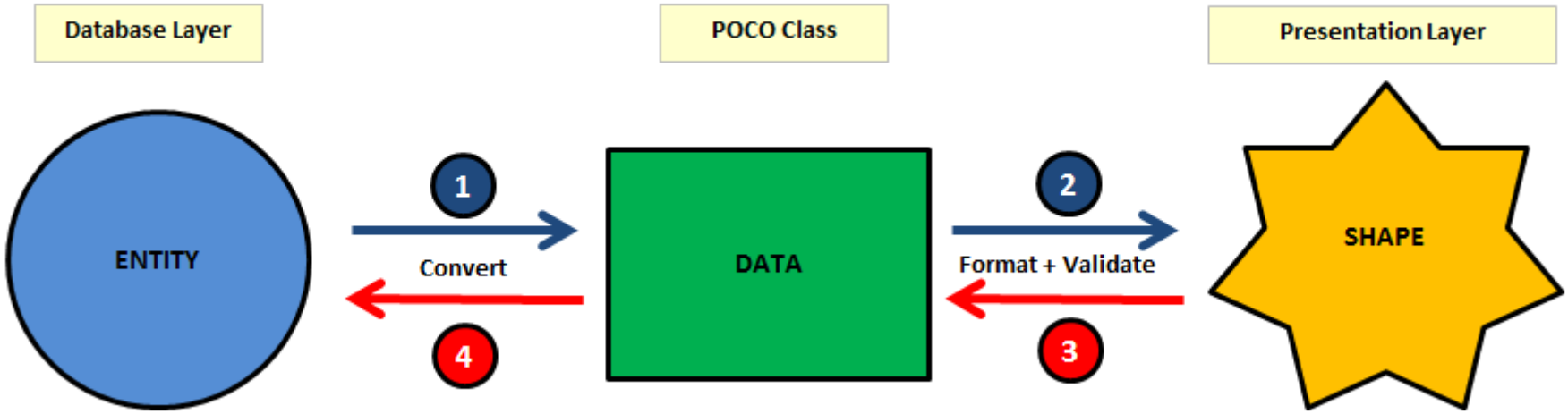
N-Tier Architecture



Enterprise Services - SOA Architecture



Middle Tier Design



Design Patterns used

1. Abstract Factory
2. Builder
3. Prototype
4. Decorator
5. Observer

Chain of Responsibility Design Pattern

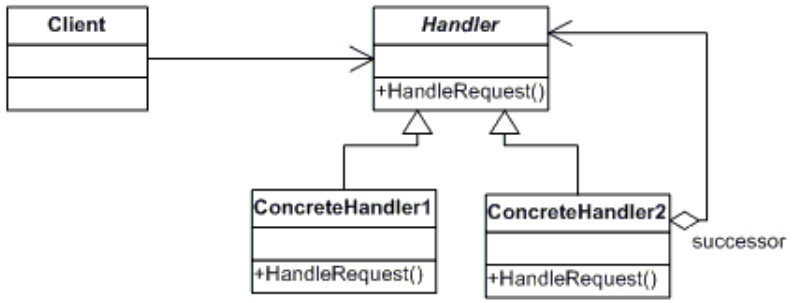
- definition
- UML diagram

---

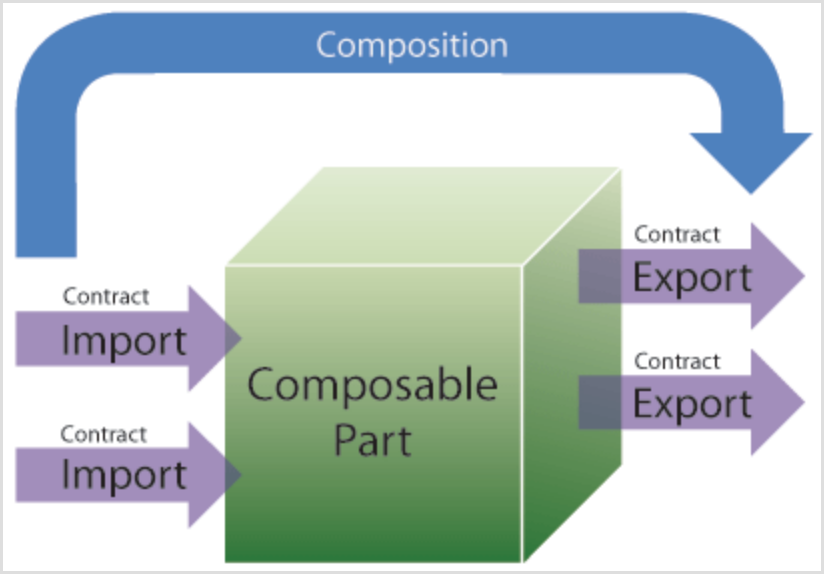
definition

The **Chain** of **Responsibility Pattern** describes how we handle a single request by a **chain** of multiple handler objects. The request has to be processed by only one handler object from this **chain**. However, the determination of processing the request is decided by the current handler. If the current handler object is able to process the request, then the request will be processed in the current handler object; otherwise, the current handler object needs to shirk **responsibility** and push the request to the next **chain** handler object. And so on and so forth until the request is processed.

UML class diagram



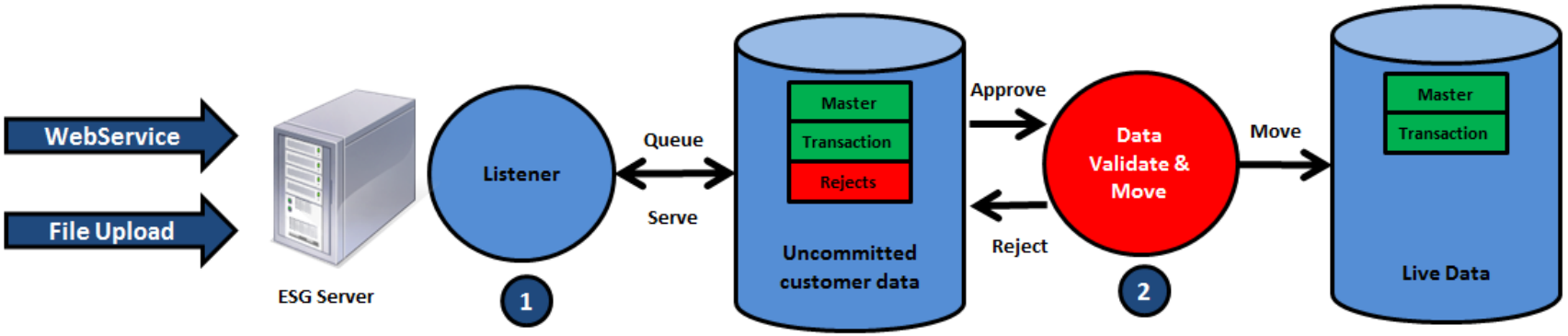
Managed Extensibility Framework



Microsoft®
patterns & practices
proven practices for predictable results

1. Build Pluggable Components that fly into place at runtime.
2. Allow dynamic configuration of object dependencies at runtime.
3. Bundle and Organize components into modules and packages.
4. Provide separation of concerns i.e. separate objects into UI, Reports, etc.
5. Provide object mapping.
6. Separate objects based on functionality and nature.
7. Build shared components, avoid duplication of code.

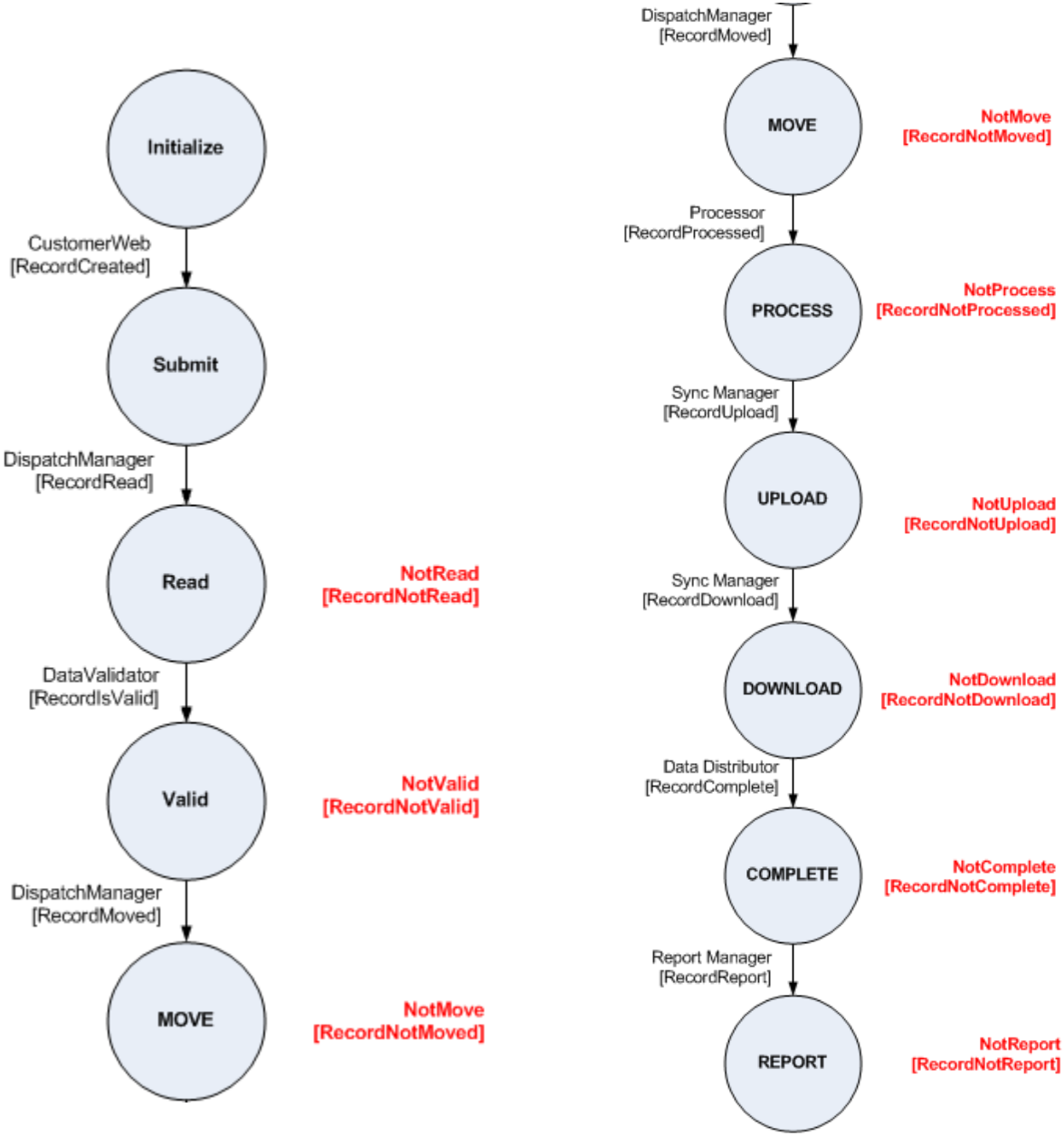
Cloud Transactions in Action



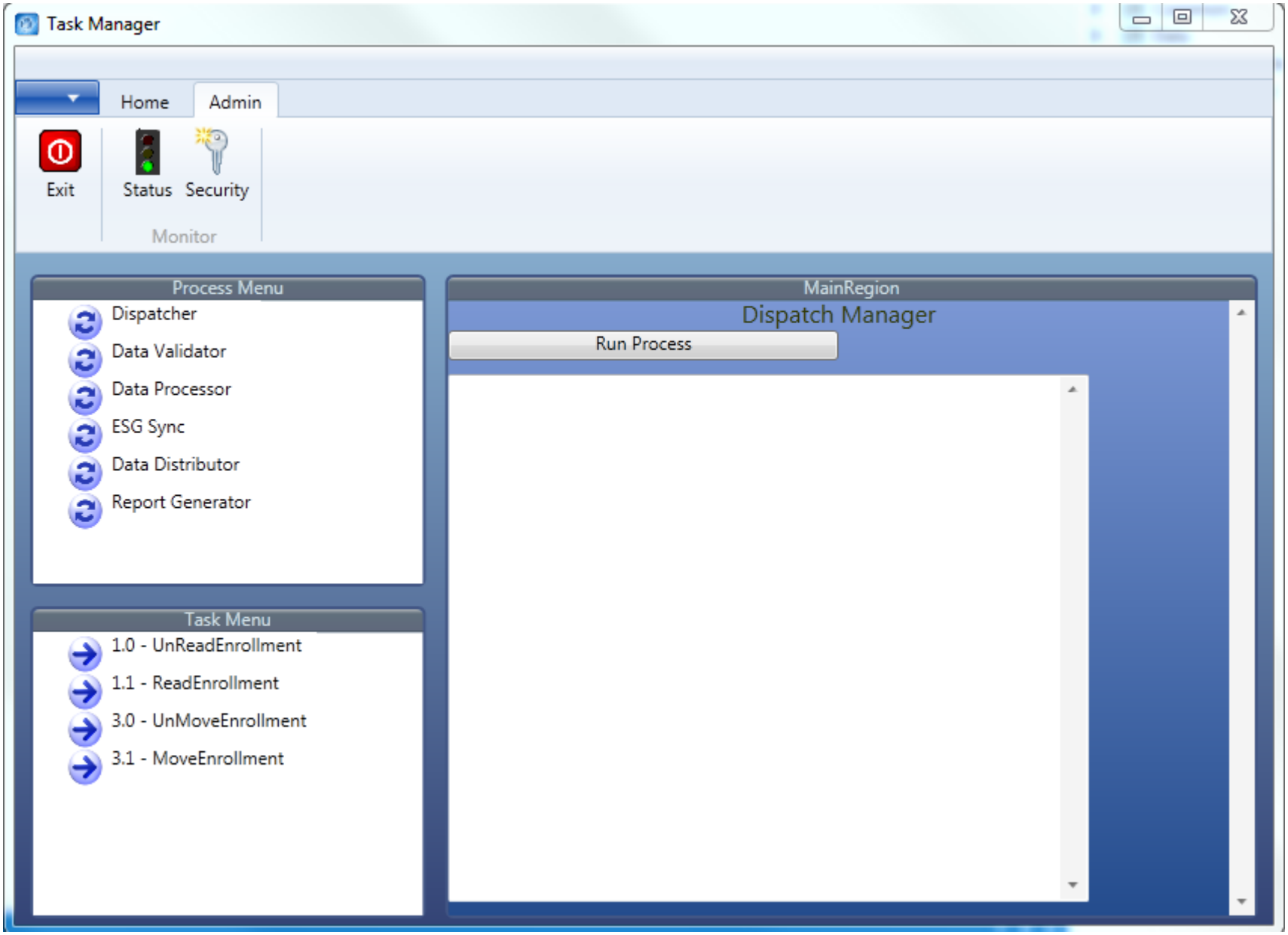
The above diagram is a closer look at the Cloud integration server under the hood:

1. Cloud Fabric has a listener that receives data via an upload enrollment webservice or an upload excel file in csv format. It can read one record or a batch file. It does some preliminary validation and saves the data to an intermediate data store.
2. Cloud Fabric has some other processes running that validate and move the data. If the data fails validation then it is rejected. The validated data makes it to the Live data store.

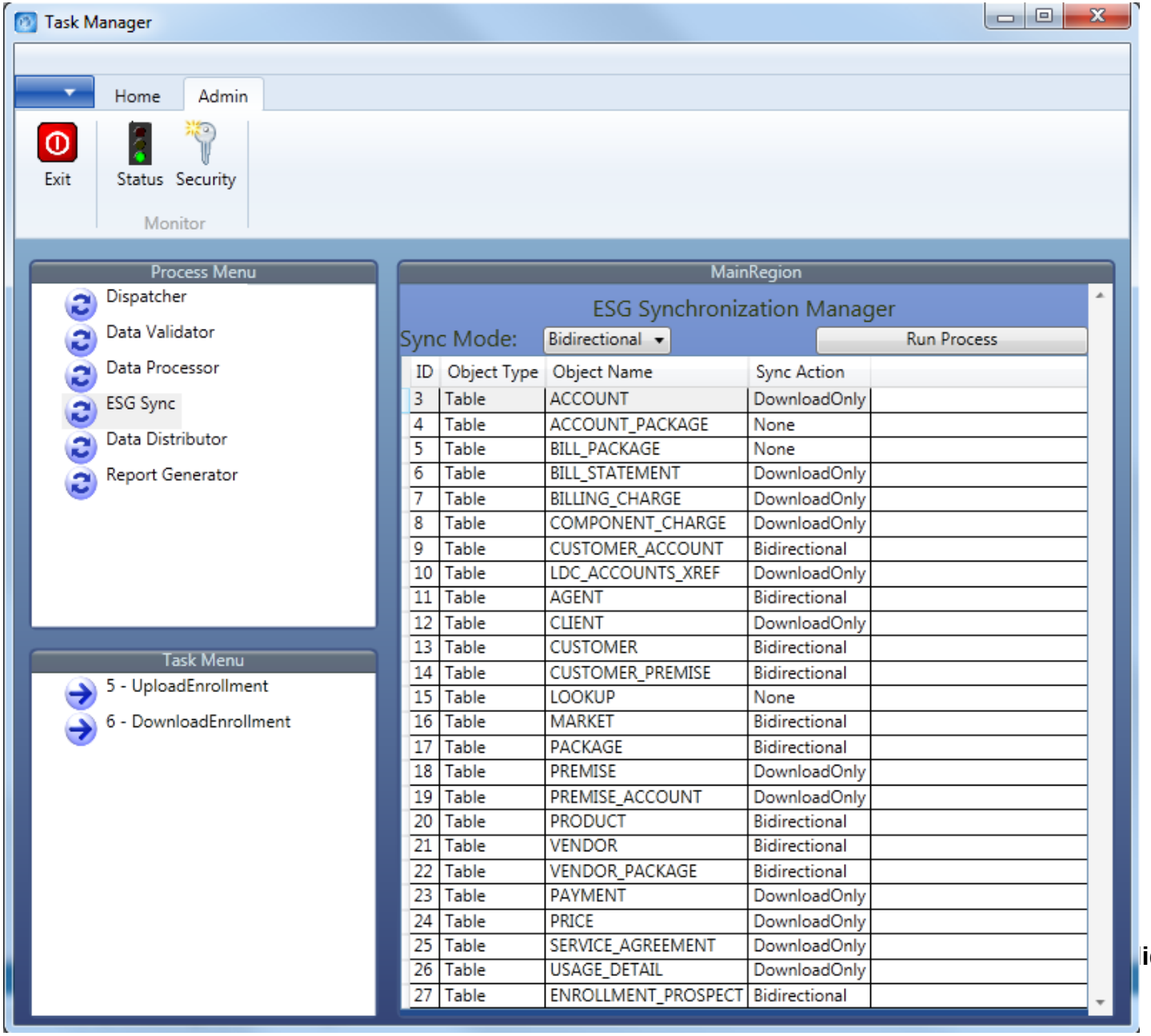
Integration Workflow



TaskManager in Dispatch Mode



TaskManager in Sync Mode



References - Further Reading

Widget Demo - http://dropthings.omaralzabir.com/

Controls - http://www.devexpress.com/

Windows 8 - http://droptiles.com/

SharePoint - http://www.brightstarr.com/

Infopath Forms

Ideablade - http://www.ideablade.com/

IdeaBlade - http://cocktail.ideablade.com/

Wijmo Widgets - http://wijmo.com/widgets/

Windows Service Bus

Big Data (Large Volume) WebServices

WebServices written using Windows Communication Foundation using REST for data messaging. The services are module based and dynamically configurable using Managed Extensibility Framework and the PRISM Composite Application Guidance Design Pattern.

Questions?