Relocation Industry - Legacy Modernization

Dileep was the **lead architect** on the Perot Systems team to help Cartus realize the vision of its legacy modernization initiative. He worked closely with Scott Mitchell (Director Architecture - Cartus), and Robert Biely (Technical Lead - Bearing Point) to develop the key design of the Financial Re-engineering effort, which was the most intricate project in the overall Legacy Modernization program. Dileep managed Perot Systems' onsite and offshore integration team to deliver the detailed technical designs and the associated code on time and within budget.

Client

Cartus is the industry leader in global mobility and employee relocation support to organizations worldwide. Annually, Cartus assists over 100,000 transferees spanning across 140 countries, in the corporate, government, military and membership organizations with its range of cost-effective relocation services. This is achieved by leveraging an elaborate service provider network spread across five continents.

Problems

Based on a 90 day analysis, Cartus personnel found that there business systems were not supporting the business operational model for future growth. Specifically, they found:

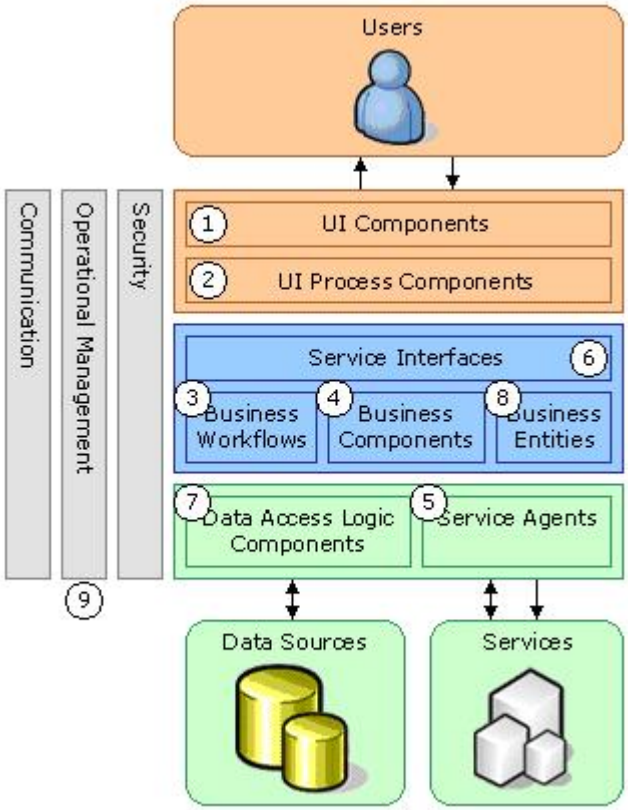
1) To avoid expensive system changes, Cartus counselling and resource center groups were using email, tasking etc to manage the workflow manually.
2) Databases used by different Cartus business systems were out of sync, required duplicate entry, data scrubbing and manual inspection for accuracy.
3) Off the shelf applications were avoided in Cartus because of the efforts required to integrate them with legacy applications e,g A MS Access system was developed to calculate VAT tax instead of implementing a third party VAT application and integrating that with the legacy financial application.
4) Legacy applications and technology tools were not suitable for extending the Cartus business processes to its service providers to reduce the administrative overhead.
5) Legacy financial applications required extensive workarounds to support new international requirements e,g the multi-currency requirements of one their major clients were fullfilled by a complicated process which required multiple client ids for the same client.
6) Offline databases, which were costly to maintain, were used for productivity measurement, metrics, dashboard reporting, decision support and service provider management reporting.

Solution

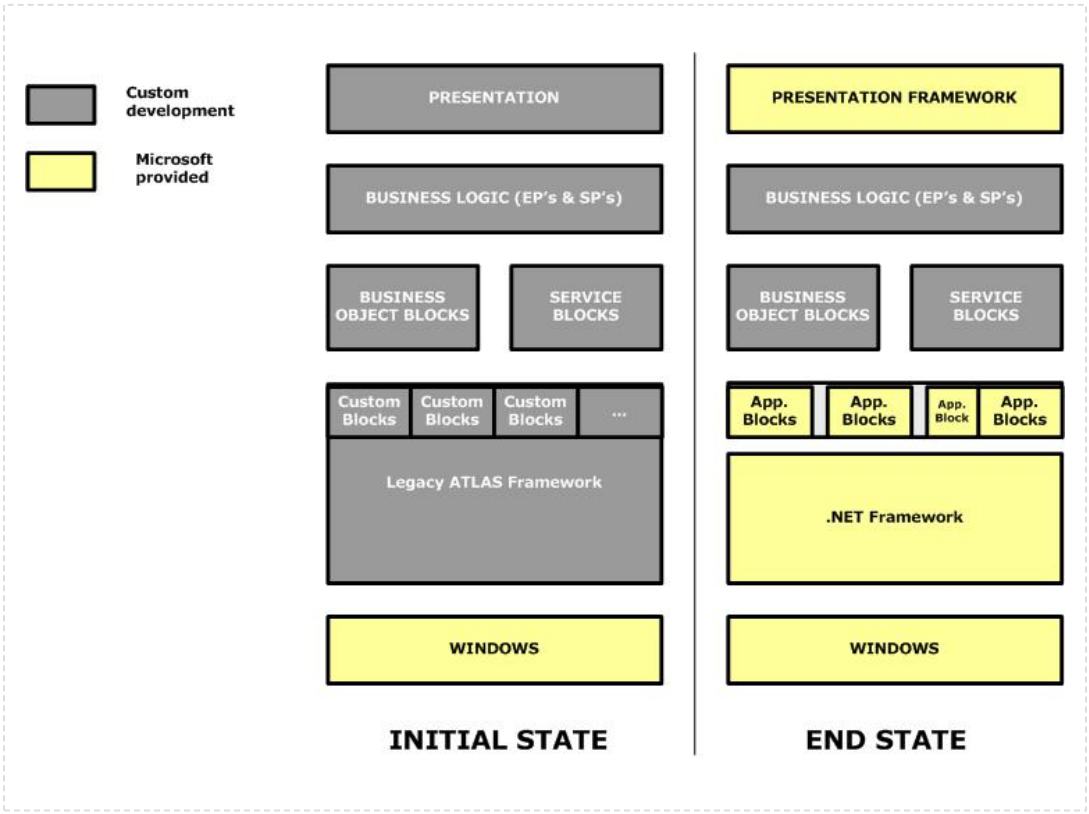
After the analysis, the Cartus team decided to have a single web-based application, that will have a role-based security for all members of the virtual team servicing a customer from any location, policy-driven tasking and workflow, access to customer file related email and attachments, and integration with telecom systems.

The highlights of the solution are:

1. *Use of .NET as the standard application platform.*
   .NET platform was chosen because Cartus already had developed a web application on Microsoft platform. A move to Java platform
   was considered cost prohibitive.



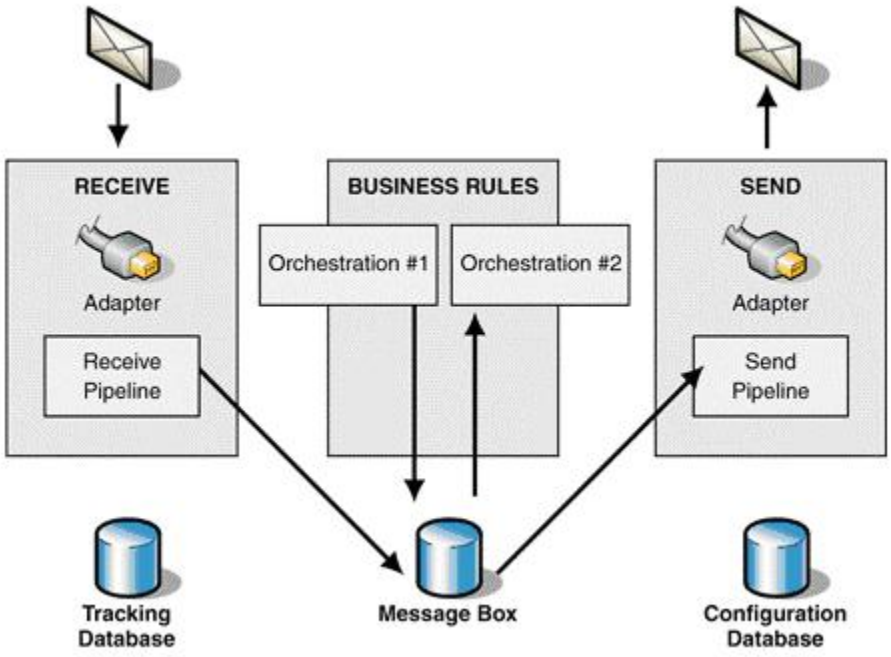
Microsoft Application Blocks were used so that the entire development team could follow the best practices.



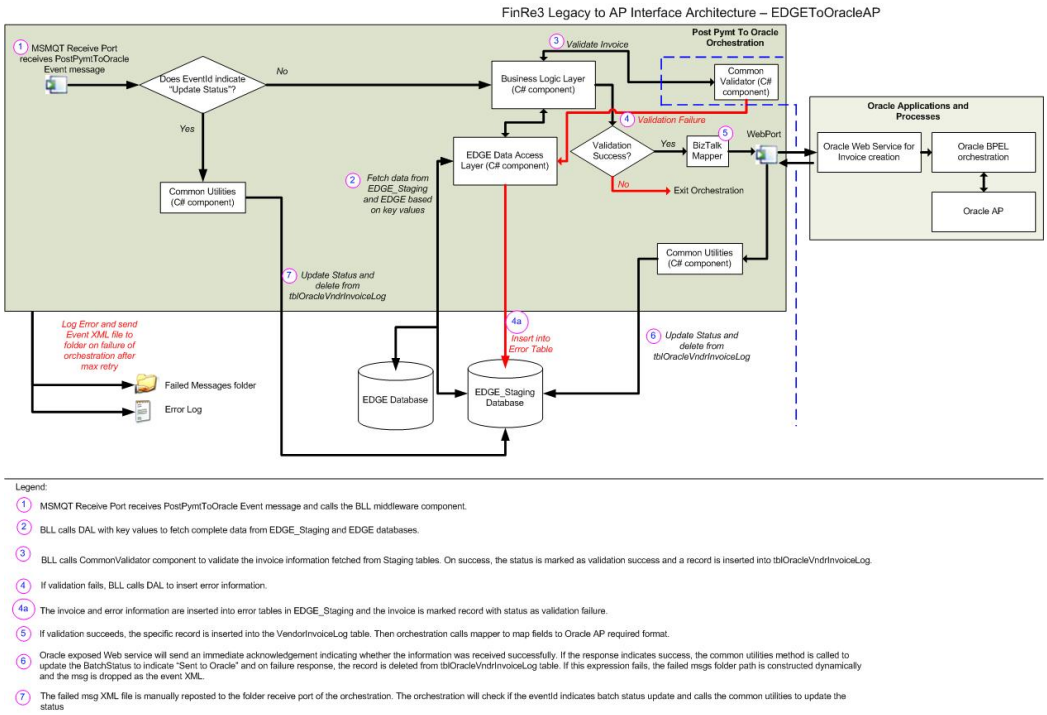
2. Use of BizTalk 2004 for Enterprise Application Integration and Business to Business system interaction.

MSMQ was used for message queuing. BizTalk acted as the hub and offered features to read messages sent from the originating

system into queues, and process them using the orchestrations to interact with one or more target systems.



Example Integration Design for the Financial Re-engineering project



Legend:

1. MSMQT Receive Port receives PostPymtToOracle Event message and calls the BLL middleware component.
2. BLL calls DAL with key values to fetch complete data from EDGE_Staging and EDGE databases.
3. BLL calls CommonValidator component to validate the invoice information fetched from Staging tables. On success, the status is marked as validation success and a record is inserted into tblOracleVndrInvoiceLog.
4. If validation fails, BLL calls DAL to insert error information.
4a. The invoice and error information are inserted into error tables in EDGE_Staging and the invoice is marked record with status as validation failure.
5. If validation succeeds, the specific record is inserted into the VendorInvoiceLog table. Then orchestration calls mapper to map fields to Oracle AP required format.
6. Oracle exposed Web service will send an immediate acknowledgement indicating whether the information was received successfully. If the response indicates success, the common utilities method is called to update the BatchStatus to indicate "Sent to Oracle" and on failure response, the record is deleted from tblOracleVndrInvoiceLog table. If this expression fails, the failed msgs folder path is constructed dynamically and the msg is dropped as the event XML.
7. The failed msg XML file is manually reposted to the folder receive port of the orchestration. The orchestration will check if the eventId indicates batch status update and calls the common utilities to update the status

The receive port of 'Post Payment to Oracle' Orchestration reads the key values of invoice data from the event message sent to its MSMQT based receive location by RAS aggregate orchestration or Home Sale aggregate orchestration. C# business logic components and the data access components will retrieve the complete invoice information from EDGE_Staging database and pass it to the common validator component to perform business validations. On successful validation, the data is transformed into the Oracle format using the BizTalk mapper. This formatted data is sent to the web port to consume a web service provided by Oracle AP. Oracle AP web service will return an immediate acknowledgement for the web method call. If this response indicates error to accept data by Oracle AP, the batch status in the staging table is updated to indicate error in oracle web service and the record in tblOracleVndrInvoiceLog is deleted. If the response indicates success, the batch status is updated to indicate that the data was sent to Oracle AP.

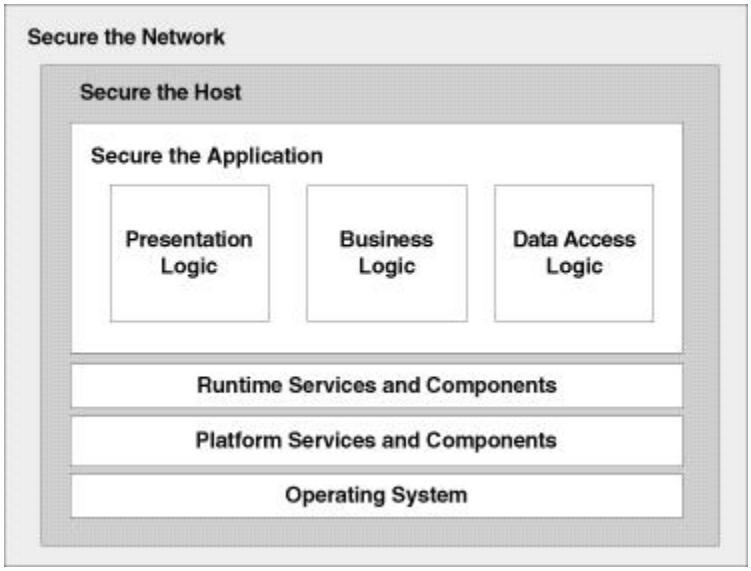
In case of validation errors, the key details will be written into an error table in EDGE_Staging database. The orchestration will drop the errant event XML into a Failed Messages folder. In case of errors other than validation errors, the errant event XML will be dropped into the Failed Messages folder

by the orchestration and the error information will be logged into an error log file.

3. Use of defense in depth strategy

Specifically, security is handled at the following levels:

- Robust network-level intrusion detection devices
- Locked-down server configuration (Windows Server 2003)
- Robust Antivirus Strategy
- Web sites protected by RSA ClearTrust web agent (ISAPI filter)
    - Authentication (against Active Directory for employees, ClearTrust for others)
    - Authorization is achieved by granting “entitlements” to various entities in ClearTrust
        - Organizations
        - Applications
        - User Roles
    -
- API-based entitlement checking against ClearTrust to display only authorized navigational links
- Search parameters constrained by user/role
- Page-level “ownership checking” to prevent unauthorized context-switching
- Field-level security providing role-level protection for sensitive data
- Encryption of data for sensitive information


Secure the Network
Secure the Host
Secure the Application
Presentation Logic
Business Logic
Data Access Logic
Runtime Services and Components
Platform Services and Components
Operating System