The Bank of Tokyo-Mitsubishi UFJ, Ltd.

Architecture Document

Version 1.0

Publish Date: Jan 2, 2014

Project Name: Calypso

Project ID: 0641

TABLE OF CONTENTS

REVISION HISTORY

Revision	Release Date	Author	Changes in progress
0.1	January 2, 2013	Julio Cesar	First Draft.
0.2	January 30, 2014	Jonas Araújo	Add Architectural Flows for Integrations(Phase 01)
1.0	February 24, 2014		Signed Off by the client (Eduardo Novo)

1. SUMMARY

This document describes the Architecture Guidelines that the Calypso implementation project should follow.

The integration spreadsheet (appendix B) contains a list of integrations identified during the preparation phase. A high level view of integrations for other phases is also shown.

A brief section explaining Calypso architecture is show.

2. INTEGRATION PHASES

Integrations will be done according to the following product prioritization done by BTMU:

- Phase 1 – Derivatives
  - Futures
  - Swaps
  - NDF
- Phase 2 – Fixed Income
- Phase 3 - FX

The diagram below shows the systems that should interface with calypso that were gathered during the Calypso Preparation Phase (see Appendix B for Integration Spreadsheet) for the three phases of the Calypso project:

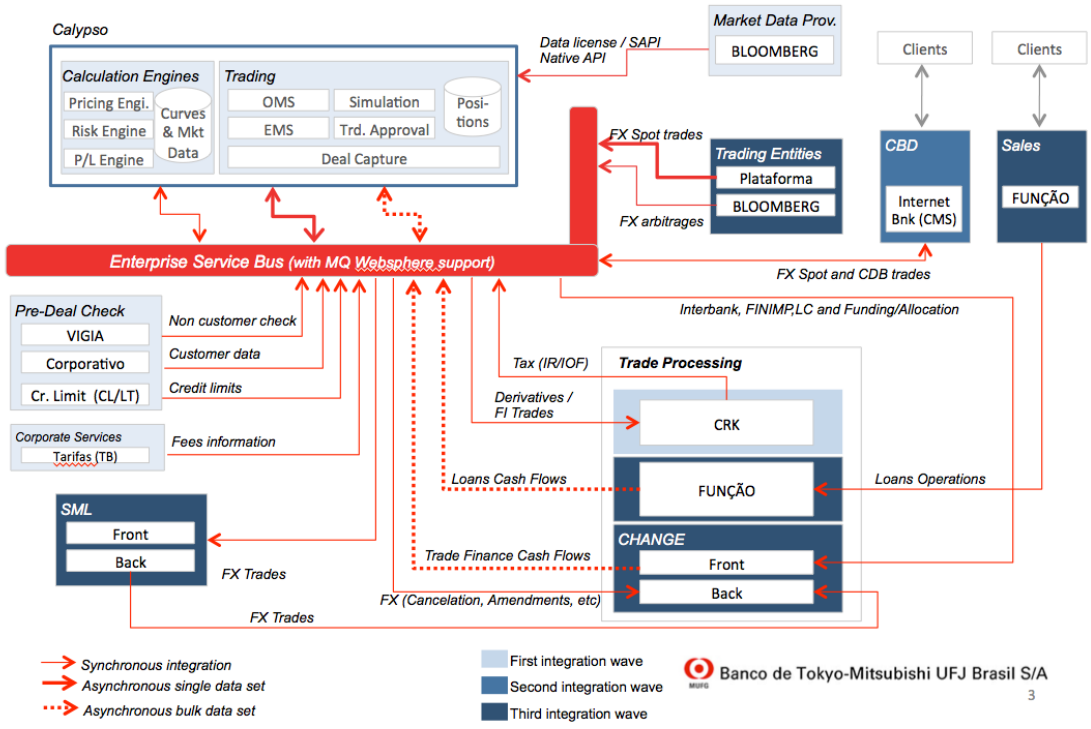

Figure 1 - Integrations for all phases

3. SOA GOVERNANCE

3.1 ENTERPRISE SERVICE BUS (ESB)

All communication between the systems should go through the Enterprise Service Bus (ESB). The preferred communication channel (connectors) between the systems and the ESB are:

1. Web Services (SOAP/HTTP)
2. MQ Queue (Websphere MQ)

All point-to-point interfaces should be avoided and the existing ones should be migrated to use the ESB.

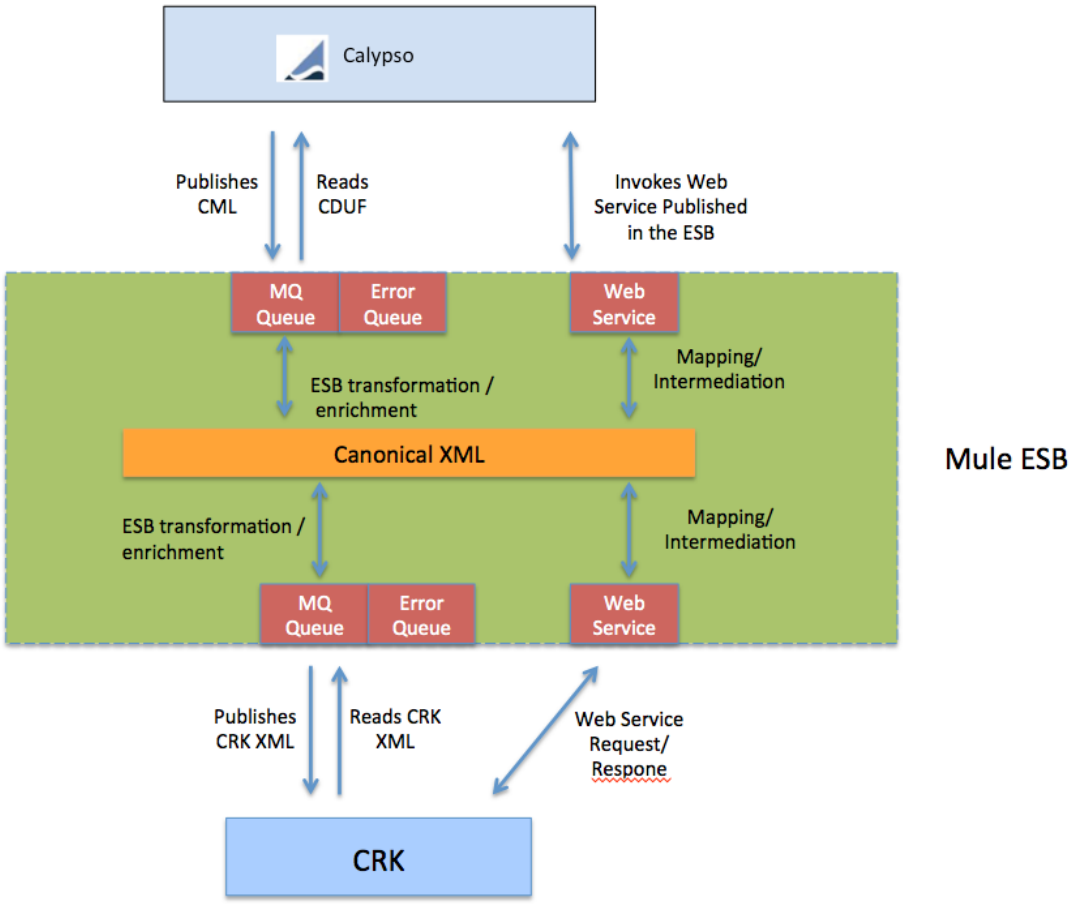

Figure 2 General Flow for Legacy Systems

3.2 ESB MESSAGE FORMAT: CANONICAL MODEL

The ESB message format is XML. All XML Messages flowing through the ESB should follow the XML Canonical Model. (See Figure 2 General Flow for Legacy Systems)

1. The definition of the XML Canonical Model should be done by IT once Business defines which information is necessary across all the institution.
2. A source system must be defined. The source system (i.e. the system that is the source of the information that will be provided to all services in the bus. As an example, Corporativo is the source system for all Client Data) defines which information will be available in the canonical model.
3. After defining which information will be available to all services, the XML Format is defined.
4. The XML Canonical Model is published.

The systems interacting with the ESB should preferably publish all information in the XML Canonical Model.

New systems should adopt the canonical model. Legacy systems will take benefit of ESB transformations/enrichments to convert to the XML Canonical model.

3.3 ERROR HANDLING

3.3.1 Websphere MQ

Systems that are interfacing using a MQ Queue, should log the errors on a separate error Queue. This way, every queue must have an associated error queue.

3.3.2 Calypso

Calypso will monitor integrations errors in the Task Station. Every integration with Calypso will have a separate tab in the Task Station where errors/warnings/messages can be seen.

3.4 SECURITY

3.4.1 Phase 1

For Phase 1, ESB security of Web Services is not a requirement. Security of the services will be done by restricting in the firewall of the bank which servers (by IP address) in production are allowed to access the Web Services published in the ESB.

3.4.2 Future Phases (Phase 2 and 3)

For future phases (phases 2 and 3) authentication of Web Services in the ESB via WS-Security or Spring Security is planned.

4. MIDDLEWARE INTEGRATION FLOWS

4.1 PHASE 1

For phase 1 we have the following system integrations with Calypso:

- Corporativo System (Entities and Client Data)
- CRK (Trades and Taxes)
- ILMNET System
- Credit Lines (CL) system

Since the systems need to communicate with the ESB, each system has a middleware integration flow with mocks to simplify the understanding. The idea of those flows is to serve as a reference for the development phase. The final flows for the integrations can be different so for more details and updated flows please see the Analysis and Design document of each integration.

4.1.1 Corporativo System (Entities and Client Data):

The purpose of this integration is to load all Client/Entity data of the Corporativo System into Calypso, the data format will be in XML and the integration will transport the data with the support of MQ Queues through the middleware.

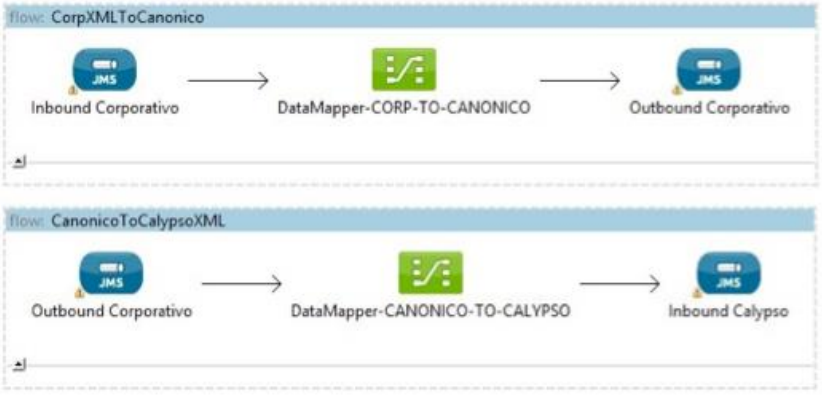

Figure 3 Mule Flow for Corporativo System

4.1.2 CRK Back Office System (Trades):

The purpose of this integration is to send trades entered in Calypso to the Back Office System i.e. CRK, the data format will be in XML and the integration will transport the data with MQ Queues through the middleware.

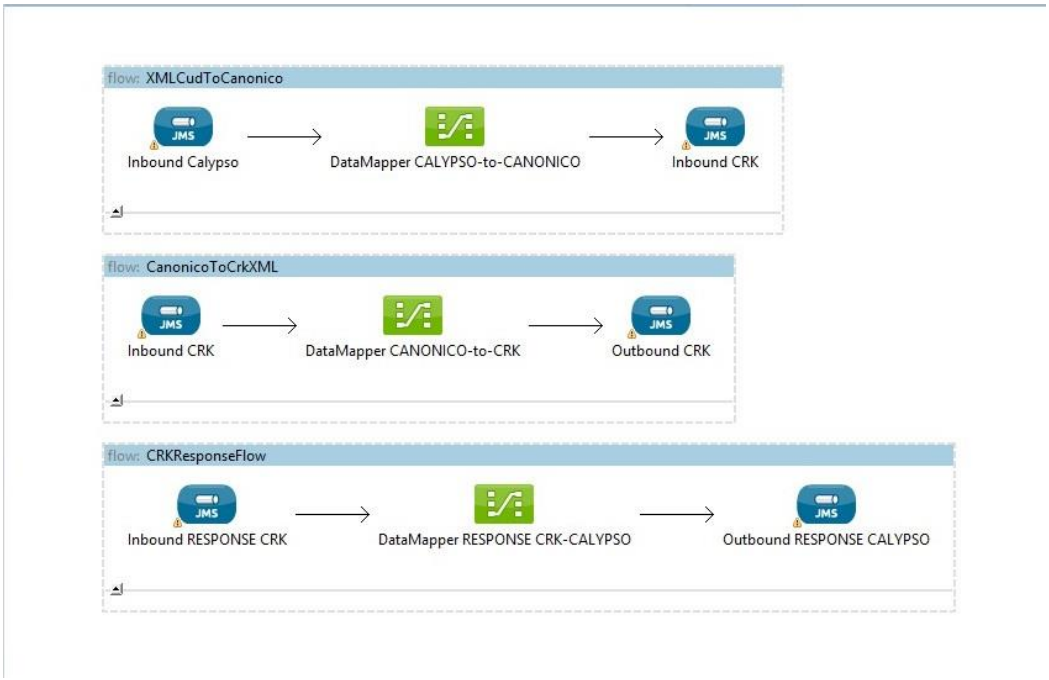

Figure 4 Mule Flow for CRK System (Trades)

4.1.3 CRK Taxes Simulation (Brazilian Tax Simulation):

The purpose of this integration is to provide to the Calypso System information about Taxes for a specific Trade. This integration will work via a call to a web service available on the CRK System. Calypso System will invoke the web service through the middleware (Mule ESB).

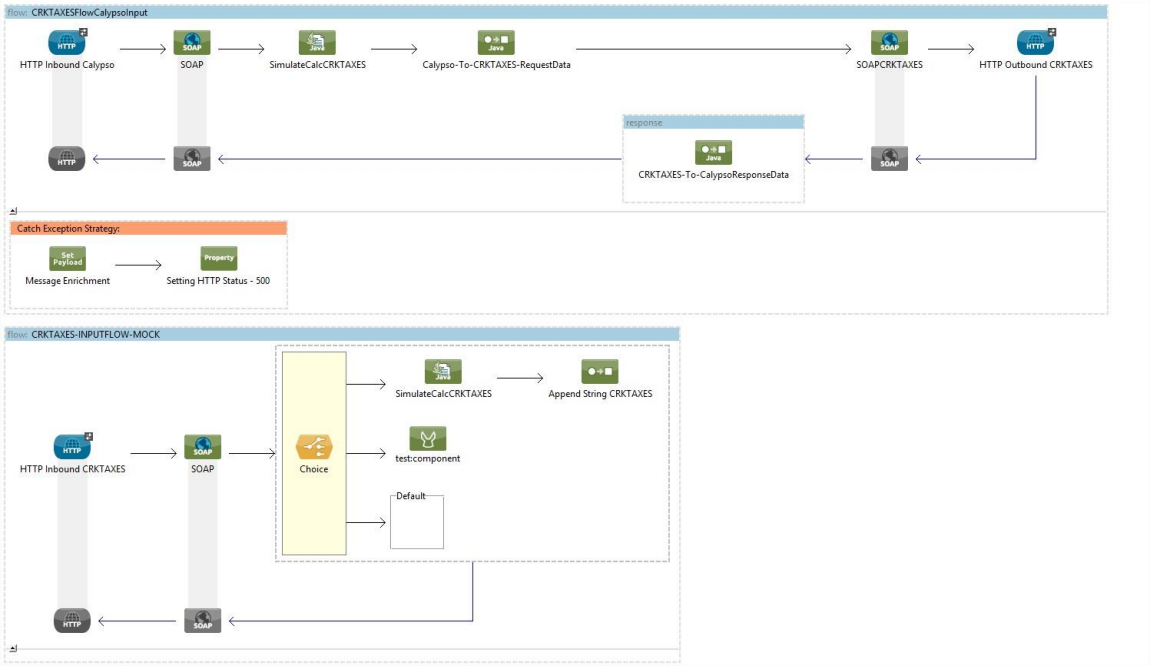

Figure 5 Mule flow for CRK Taxes Simulation

4.1.4 ILMNET System

The purpose of this integration is to calculate the Potential Exposure measure of a trade. The integration will work by a web service on the ILMNET System and the Calypso System will request this web service through the middleware.

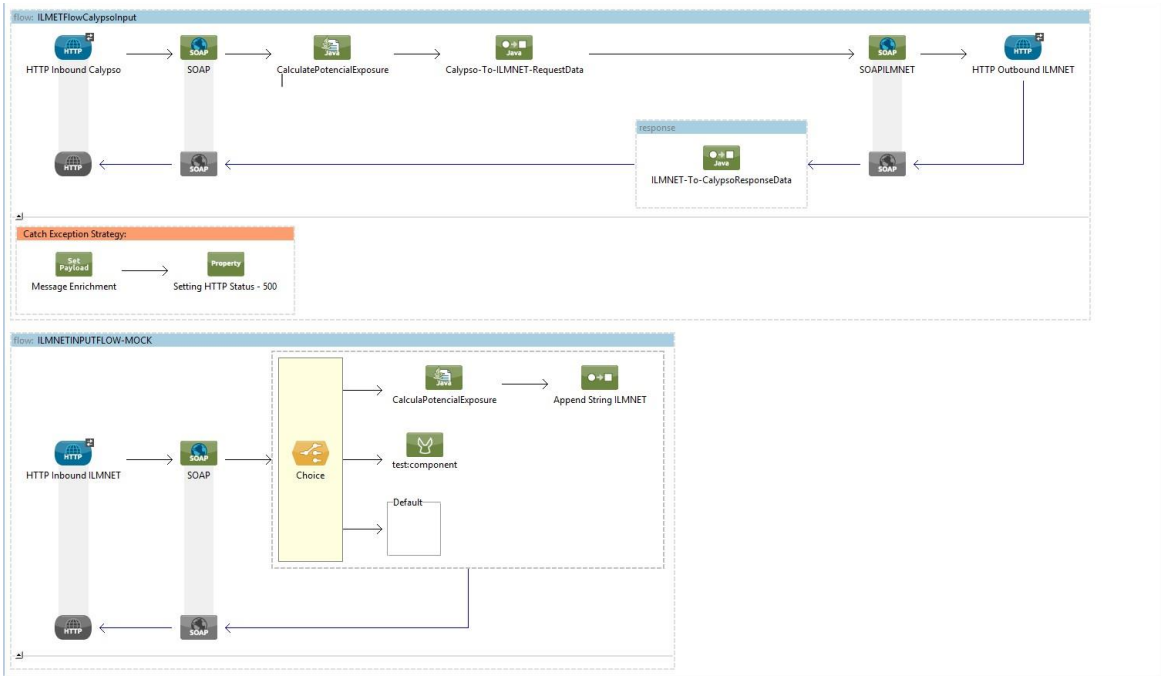

Figure 6 Mule Flow for ILMNET System

4.1.5 CL System (Credit Lines/Legal Lending Limits):

The purpose of this integration is to query and update the CL System with the new trades of the Calypso System. The integration will be live and work via a web service on the CL System, the Calypso System will request this web service through the middleware after the process of save a new trade.

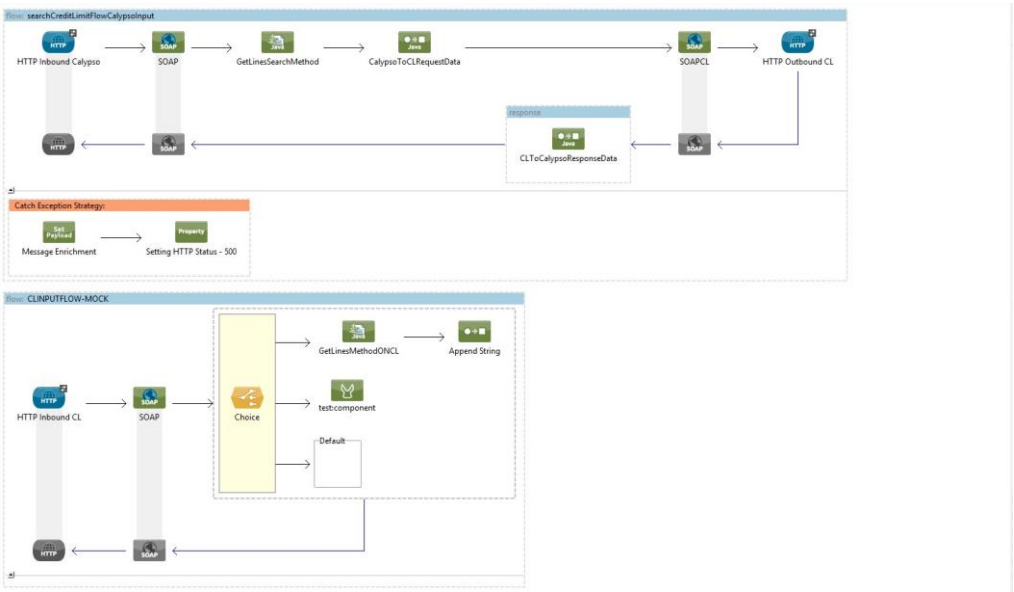

Figure 7 Mule Flow for CL system

5. CALYPSO ARCHITECTURE

The architecture of Calypso is divided into five components: the database component, the server component comprising the Data Server and the Event Server, the engine component, and the Authentication Server.

- **Data Server**: The Data Server is the lone bridge of communication between the database and the rest of the Calypso system. It hides the underlying relational structure of the database so the user, or developer, only has to work with the Java classes. The Data Server has other responsibilities. It implements the audit facilities. Since it's object-based, it's easy to provide field-by-field differences for users and to compare a current object version with an older one. It can cache objects it has loaded or stored to speed retrieval times; it implements the access control mechanism; and it applies workflow rules.
- **Event Server:** Server responsible to propagate and manage all events that are consumed in calypso. It supports a standard publish and subscribe model.
- **Authentication Service:** Responsible to manage authentication in Calypso.
- **Engines**: Engines are independent programs that implement business logic. Engines are multi-threaded and are designed, by definition, to subscribe to persistent events. Each engine follows a simple execution model: get an event

from the event server, process the event - this may require loading data via the Data Server - and then save the results. When they need to load or save data, they use RMI to call the appropriate method on the Data Server.

A typical interaction with Calypso is described below.

0. Engines are configured in Calypso to subscribe for some events
1. User books a trade in Calypso
2. Data Server authenticates user with the Authentication Service. This is done through a token saved for every user session.
3. Data Server publishes to Event Server the event associated (PSEventTrade) for booking a trade. The Event Server receives events and then forwards them on to event subscribers (engines and applications)
4. Upon receiving an event, subscribers (engines) typically execute various methods to retrieve critical data from the event object itself or from the database.

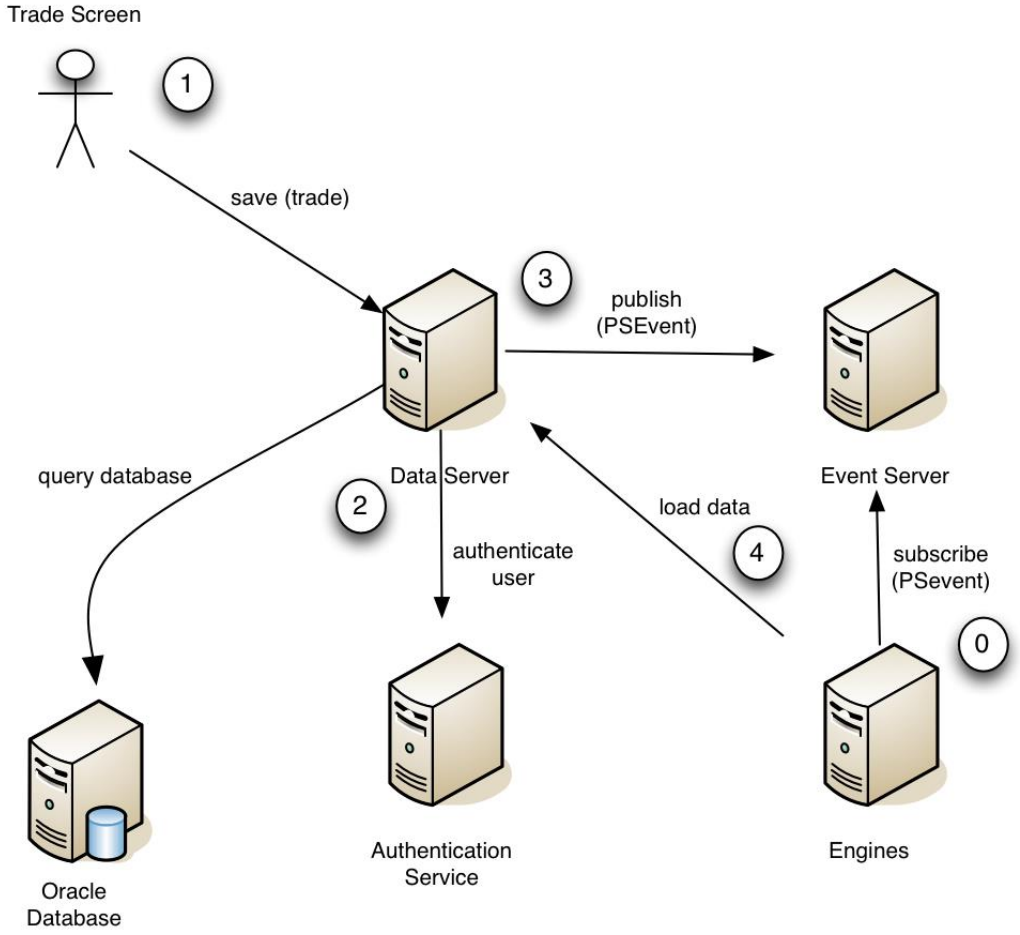

Figure 8 - Typical Calypso Interaction

6. APPENDIX A—ACRONYMS AND ABBREVIATIONS

Acronyms & Abbreviations

SOA	Service Oriented Architecture

7. APPENDIX B— INTEGRATION SPREADSHEET

BTMU_IntegrationMatrix.xls