



Symphony helped a Hong Kong based software development firm specializing in trade financing technology solution to build a Trade and Supply Chain Finance software using Java Open Source technology that helped achieve on-demand access to working capital funding in banking operations with platform independent deployment.

### The Client

The client is a Hong Kong based software development firm backed by a team of banking and financial experts. They focus on offering exclusive software services to a leading global bank based out of Hong Kong to develop, support, and implement Trade Finance technology solution for bank operations across Asia.

### The Challenges

Trade Finance technology allows banks to service global open account supply chain financing by dynamically monitoring and managing importer and exporter trade flows, credit exposures, financing and payments through multiple funding sources. The global bank's objective was to introduce trade-financing technology across Asia. The global bank entered into an alliance with the client to build an enterprise-scale Trade and Supply Chain Finance software to support automation of Asset Based Lending, with powerful integration capability.

As a startup firm, the client preferred not to build and maintain in-house software development team. Hence, they sought to outsource to a third-party software R&D expert and focus on utilizing their domain (banking and financial) expertise to support in the development of the software.

### The Solution

As a software R&D partner to the client, Symphony's solution included designing three-tier distributed architecture for effective load distribution, and developing a service-oriented web based application using J2EE technologies. The technology stack included Java Open Source components like Spring, Hibernate, Acegi and

## End-to-end Product Development

### Platform

Java Open Source

### Industry / Domain

Banking / Financial Services

### Engagement Key Points

- Peak Team Size – 50
- SDLC: Agile
- Development Model: Custom MDD Model
- Release cycle – short iteration cycle

### Technology and Tools Used

- Web layer: Spring MVC, JSP 2.0 Tags, JSP, Display Tag, Cewolf( Charting)
- Service Layer: Spring, Acegi Security
- ORM: Hibernate
- Scheduling: Quartz
- Database: Oracle 9i/10g
- Web Server: Tomcat
- Configuration management and version control : SVN
- Build automation : Maven
- Continuous build process : Cruise Control
- Unit testing : JUnit
- Test automation: Watij, TestComplete
- IDE : Eclipse
- Project and defect management : ProNav(Digite)

### Major Product Features

- Supplier and Buyer Management
- Funding Request Management
- Real time Funding
- Administration
- Risk Management and Credit Scoring
- Management of Accounting standards



Quartz. The distributed application consists of:

#### WEB LAYER

- Spring Model View Controller as web component helped to integrate with the business services
- Creating all UI components using JSP2.0 tags

#### SERVICE LAYER

- The business logic was contained in POJO classes, and managed by Spring IOC (Inversion of Control) container
- Spring AOP based transactions were used with annotations
- Asynchronously run application logic was modeled as scheduled job using Quartz
- Acegi was used to provide Role based access control feature

#### DATABASE LAYER

- Hibernate was used as ORM solution to support multiple databases which helps avoid using native SQL Queries
- HQL (Hibernate Query Language) was used to handle complex queries and avoid usage of Stored Procedures

#### APPLICATION/WEB SERVER

- The application supports configuration based on deployment on various servers
  - Lightweight Spring container allowed deployment across lightweight web servers (Apache Tomcat, Jetty) with its own database connection pool and transaction handling capabilities.
  - For deployment on Application Servers such as Websphere or JBoss, the application uses the servers features such as, database connection pooling and 2-phase commit

- Superior architecture help achieved product scalability – support high volume horizontal scaling as well as vertical scaling
- Ease of product deployment
  - For on-demand access
  - By reusability of existing IT infrastructures

## The Benefits

Symphony's product development expertise achieved the following benefits for the client:

- Significant reduction in development cost