ProKarma

# Choosing an Enterprise Architecture Platform: J2EE versus Tuxedo

## INTRODUCTION

Developers are faced with important decisions when designing an enterprise computing platform. The process should always start with a requirements analysis. The analysis should describe what the system should do so that the developers and the client can come to an agreement on the necessary functions of the system. This can include business concepts, domain descriptions, user stories, and user interface mock-ups. Next, analysts will determine problem domain models. This analysis can be used to understand the business processes that will be handled by the platform. Technical and implementation details aren't included in this phase in order to keep the model ideal.

Once the requirements and problem domains have been established, an architecture can be designed to define and map out the system. The architecture includes the structure, interfaces, and communication mechanisms. This can be broken down between the enterprise-wide system and the individual applications that make up the system. Once the pieces have been established, engineers must determine the specific development platform that will be used. This white paper will consider two different approaches: Java2 Enterprise Edition (J2EE) and Tuxedo with C++.

## J2EE

Java 2 Enterprise Edition (J2EE) is Oracle's enterprise Java computing platform. The platform's design is based on modular components running on an application server. The platform includes architecture specifications that define standards at different layers of the enterprise applications. A collection of component technologies, a J2EE platform can include the following components:

- Enterprise JavaBeans (EJB), the server-side component architecture for modular construction of enterprise applications, which provides a standard system for regulating backend code.
- Javaserver Pages (JSP) for creating dynamic web pages and user interfaces.
- Java Servlets, a language for extending capabilities of a server.
- Java Database Connectivity (JDBC) defines how a client may access a database.
- RMI/IIOP enables the platform to run remotely.
- JavaMail, a framework for building mail and messaging applications.
- Java2 EE Connector Architecture (JCA) for connecting multiple applications.
- Java Message Service (JMS) for sending messages between applications.
- Java Transaction Service (JTS) and Java Transaction API (JTA) for transaction management between applications.
- Java Naming and Directory Interface (JNDI) which makes data searchable by name.

The applications are developed based upon the design patterns and blueprints available from Oracle. EJB specifies a standard framework for developing reusable business components and JMS specifies a messaging framework for communication between distributed resources using P2P and Pub/Sub. Web-tier services are provided by JSP and Java Servlets. JSP provides a clean separation between presentation and logic. JSP Tag Libraries provide reusable web components.

ProKarma

# Choosing an Enterprise Architecture Platform: J2EE versus Tuxedo

## Advantages of J2EE

Platform independent and flexible, J2EE is a mature standard that is supported by over 30+ vendors. The system is scalable and provides a wide range of flexibility in regards to component parts, integration with existing information systems, and security.

The EJB standard defines the framework for developing business components, ensuring consistency throughout the platform. Security, transaction, persistence, and naming and directory services are provided by the container, which means that developers can leave the complicated "plumbing" to the application server and concentrate on writing business logic. Open standards-based development provides more flexibility for reuse of business components.

JMS provides a messaging standard for asynchronous communications. Web applications can be developed using the standards and deployed on multiple application servers, allowing for a flexible, scalable system. JSP and Servlets provide an efficient alternative to CGI-based Web applications.

## Disadvantages of J2EE

The largest disadvantage of J2EE is that Java is the only language supported. Retraining IT staff in Java can incur significant costs when adopting and implementing the new system. J2EE applications using EJB for developing business components need to be designed properly. Without advanced skills in Java and J2EE, project development could be hindered or problematic. Infrastructure costs for development tools, app servers, and other relevant tools may increase from existing costs, though the returns can be greater if the platform is used to its full potential. Specifications for the different component technologies are continually undergoing changes which can be difficult to keep up with. Differences in implementation and deployment between application server vendors can prevent seamless reuse and sharing of business components.

## Guidelines for J2EE Development

It is a misconception that Enterprise JavaBeans (EJB) is synonymous with J2EE. EJB provides a framework for developing transactional business components. However, if an application does not do any transactional updates across multiple resources, EJB might be overkill. Successful J2EE-based applications are developed without using EJB. Business logic can be written as JavaBeans that are not distributed.

Using Entity Beans to represent persistent business objects is useful but can be expensive, due to system scalability and performance issues. Simple database-oriented applications that perform querying without updates do not need to use Entity Beans. Session Beans could use data access objects or wrap JDBC calls. Developers should be careful about "marshalling" remote reference to Entity Beans.

# TUXEDO

Also offered by Oracle, Tuxedo is used as a transaction-processing middleware for building distributed client/server applications, and as an application server for C, C++, and Cobol applications. A highly reliable and robust platform, Tuxedo provides scalability, high availability, and load balancing of enterprise services. The platform

ProKarma

Choosing an Enterprise Architecture Platform: J2EE versus Tuxedo

manages transactions across multiple resource managers using the XA/Open standard, and is available on multiple operating systems, including UNIX Flavors, Win32, and Linux. Tuxedo provides a messaging infrastructure for asynchronous communication between services.

### Advantages of Tuxedo/C++

Tuxedo is highly respected in the industry and is used for mission critical applications with high transaction throughput requirements. The platform provides load balancing and data-driven application partitioning to provide better scalability. Development using C++ could provide better performance since intermediary byte code translation isn't necessary. Tuxedo provides a framework for developing stateless business services. Database connection pooling, transaction management, and 2PC using XA are all supported by the platform.

### Disadvantages of Tuxedo/C++

The platform for business services is provided by only one vendor, limiting options and flexibility. Sharing Tuxedo services with other non-Tuxedo-based applications requires additional software, such as JOLT for Tuxedo for Java clients. Business services are not automatically portable to other platforms. Unlike Java, the business services code needs to be recompiled for another platform. Reusing business services involves purchasing the Tuxedo platform, unlike EJB components which can be deployed in any J2EE application server. Declarative transactions are not available. Transaction logic has to be hand-coded into the business logic. ATMI API is proprietary and ties the application's business logic to the vendor.

## CONCLUSION

Ultimately, developers need to analyze the needs of the enterprise thoroughly before choosing a platform. The architecture platform should be based on the requirements of the problem domains and the constraints. If high transaction throughput with transactions spanning multiple resources is a requirement, Tuxedo/C++ is a viable choice. If the enterprise requires a simple database-oriented application that retrieves data and does some business calculations, a JSP/Servlet-based architecture using regular JavaBeans for business logic is viable. If managed transactions are a requirement of the system, an EJB should be used with the J2EE platform. Problem solving remains the most important tool of the developer. When choosing architecture for enterprises, one has to remember that there is no one architecture solution that addresses all problems.

## ABOUT PROKARMA

ProKarma delivers integrated technology and business process outsourcing solutions for over 150 global leaders in a wide range of industries and markets. ProKarma is co-headquartered in Portland, Oregon and Omaha, Nebraska, with sales and delivery centers in the United States, India, Argentina and Peru. ProKarma was selected as a Global Services 100 Provider for 2012 and ranked as the fastest growing IT services company in America by Inc. 500.