

---

# FORTÉ TO J2EE

---

## MIGRATION

---

A White Paper from  
**Goldstone Technologies Ltd.**



[www.goldstonetech.com](http://www.goldstonetech.com)



## Contents

Contents .....	2
Introduction – Situation before Forté .....	3
Advent of Forté .....	3
Web Services – Future Technology .....	5
Goldstone - Forté to J2EE Migration Approach .....	9
Conclusion .....	11
Contact Information .....	11
Appendix 1 .....	12
Appendix 2 .....	15
References: .....	15

## Introduction – Situation before Forté

Creating competitive advantage requires a lot more than the operational streamlining made possible by application packages. To keep a competitive advantage, enterprises must create software components and assemble them into the highly tailored applications that drive profitability.

Delivering new applications in a timely fashion, however, has become increasingly difficult, due to the infrastructure issues of networked environments. According to a recent survey by The Standish Group, a leading high tech market research firm, organizations have been spending twice as much effort solving software infrastructure issues as they have implementing the needed business functionality. The Industry average project success rate has been: Failed 40%, Challenged 33% and Succeeded 27%.

To deliver successful applications consistently and in a timely fashion, IT organizations needed more powerful tools that allowed developers to concentrate on creating the needed business functionality instead of wrestling with complex software infrastructure issues. To capture market opportunities, businesses had to move more quickly and forcefully. It was realized that strategic advantage comes from software applications that allow firms to compete with Internet speed and Enterprise power. In order to survive the competition firms required to deliver what customers wanted, when they wanted it, and at a very competitive price. It obviously meant that software had to present a unified customer relationship, a single business image - even if the business operations involved many systems. Organizations needed highly integrated systems, which at the same time had to remain flexible so as to accommodate the next wave of changes and opportunities.

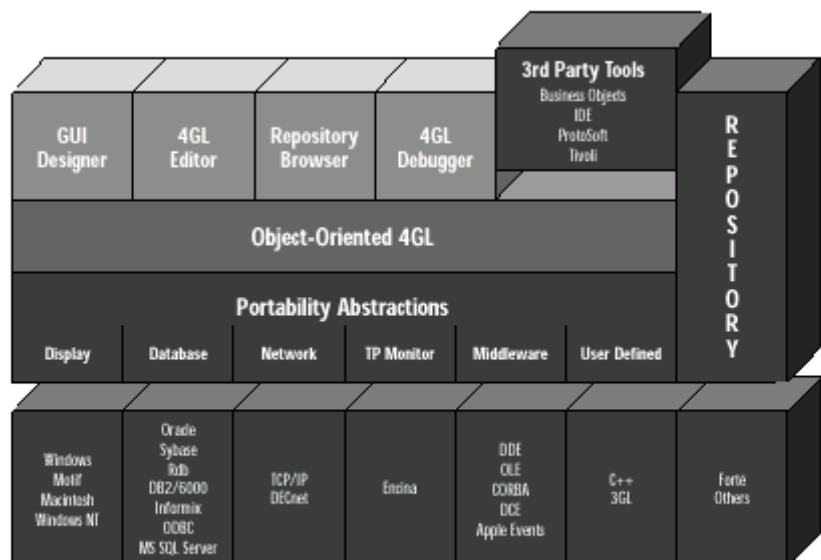
## Advent of Forté

Forté is an integrated set of products that deliver a complete solution for building, integrating, automating, and deploying enterprise networked applications. Specific product offerings include Forté 4GL, Forté Fusion, Fusion Conductor, Forté Express, Forté WebEnterprise, Forté Adapter Series.

With the advent of Forté – there was a dramatic revision in the Project Success Rate. 47% of Forté customers completed the project in time and budget. Another 27% finished within two months of the target date. Forté follows a unique and comprehensive approach without using different tools to solve each part of the development, integration and deployment challenge. Forté eliminates the need for buying and using separate client development tools, server development tools, middleware, object request brokers, repositories, and application management tools. This approach has proven effective in dramatically shortening development time, reducing development risk, providing greater business flexibility and decreasing costs.

The Forté environment fosters a style of development and integration that collects user feedback and minimizes the chances of failure. With Forté solving most of the infrastructure challenges, projects are shorter, learning curves less steep, and developers more productive than with other development and integration approaches.

Technologically, Forté pioneered solutions that make it easy to develop, integrate and deploy networked applications. This is done by insulating developers from the low-level infrastructure issues of software deployment. Forté environment takes the application definitions and automatically generates a scalable, reliable, open and manageable distributed system that facilitates integrating entire business processes by spanning packaged applications and departmental technologies. The applications can have more features and better performance, delivered to meet an aggressive deadline.



The Forté architecture provides an abstraction layer

*Forté Meets Technical Requirements*

- High programmer productivity
- Scalability and reliability
- Component-based multi-tier architecture
- Internet and large-scale internal deployments
- Open systems APIs and multivendor standards
- Integration with application packages, electronic feeds or legacy systems

*Forté Satisfies User Needs*

- Faster Time to Market
- Reduced Risk
- Cost Effectiveness
- Proven Solution
- Long - Term Value
- Flexibility

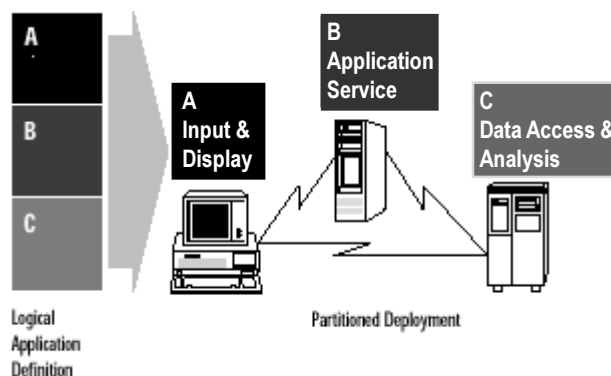
*Forté helps in solving various challenges:*

**The Enterprise Application Challenge**

The Forté Application Environment includes tools and infrastructure that enables developers to easily build new application components, integrate external application components and automate entire multi-step business processes. Forté also includes a component deployment environment with a powerful set of pre-integrated services for enterprise levels of scalability, reliability, and manageability.

**Speed in Application Development**

Forté's object-oriented development environment includes middleware services, transactional semantics, an integrated debugger, and a shared repository for the most effective team development. Forté applications use components and Platform independent Partitioning.

**Simplifying Application Deployment**

Forté's open runtime environment ensures successful deployment with enterprise levels of scalability, reliability and manageability. Load balancing, compiled code, parallel processing, component consistency, performance monitoring, software distribution etc are a few typical features that help in simplifying application deployment.

**Enterprise Application Integration**

Forté's Application Integration Features support multiple application adapters, integration techniques and solve the critical requirements for a successful EAI solution. The Forté Adapter Series enables consistent and extensible integration through an architecture that is based upon open technologies and industry standards like XML for data interchange and HTTP messaging protocol for transport of XML messages, making Web-based applications a natural extension of Forté integration solutions.

## Web Services – Future Technology

The next generation of distributed computing has arrived. Over the past few years, XML has enabled heterogeneous computing environments to share information over the World-Wide Web. It now offers a simplified means by which to share process as well. From a technical perspective, the advent of **Web Services** has been a natural evolution of XML application from structured representation of information to structured representation of inter-application messaging.

Web Services could be defined as 'A collection of functions that are packaged as a single entity and published to the network for use by other programs. Web services are building blocks for creating open distributed systems, and allow companies and individuals to quickly and cheaply make their digital assets available worldwide'.

Web services must be reliable, highly available, fault-tolerant, scalable, and must perform at acceptable levels, which is the basic need of any other enterprise application. **J2EE** (Sun Microsystems) and **.NET** (Microsoft) are evolutions of existing application server technology used to build such enterprise applications easily.

J2EE is an industry standard led by Sun Microsystems and was designed to simplify complex problems with the development, deployment, and management of multi-tier enterprise solutions. It is not a Product and describes agreements between applications and the containers in which they run. Microsoft.NET is a product suite that enables organizations to build smart, enterprise-class web services. .NET is a product strategy, whereas J2EE is a standard to which products are written. This white paper will not attempt to compare J2EE and .Net. Several such comparisons have been made by the software industry in the past involving several features like time to market, support for existing systems, maturity of the platform, portability etc.

## Relevance of Forté in current scenario

The Forté User community has been watching the development of Web Services as the future technology. While it is widely agreed that Forté has been a robust and stable product, however there is also an underlying threat about its uncertain future. In the likelihood of Sun stopping support of Forté, the current deployed applications will not be flexible to accommodate any new changes and future opportunities. Also, Forté is a proprietary technology whereas J2EE is not. As Sun continues to support the open technologies, there is a doubt cast on how much effort Sun would spend on a proprietary technology such as Forté rather than its strong hold Java.

The biggest question that is faced today is "Where is Forté going in the future?" This has led the Forté user community to explore the various options to move the Forté based applications to other viable and dependable technologies, which would support the strong features of Forté and also assure to protect the investment for future web enhancements. The obvious choice has been J2EE on several merits comparable with Forté.

*J2EE helps in solving various challenges:*

### The Enterprise Application Challenge

The Java Enterprise Technologies with the J2EE standards provide user interfacing, data enabling, communications enabling, system assurance services, web enabling and application enabling which constitute the critical requirement for any Enterprise Application architecture technology. Enterprise System development has benefited very much with the use of components model, which is one of the core underpinnings of Java Enterprise development and the J2EE. Features such as transaction processing, security, object persistent, resource pooling etc have helped in scalability and performance for Enterprise Applications.

### Speed in Application Development

J2EE Enterprise Java Beans (EJB) Component Model serves as the core way in which distributed Java Enterprise Applications can be rapidly built. Component based model enables Enterprise Software to be developed more rapidly with high reliability. Components have a coarse-grained nature that facilitates more rapid application development. EJB clients talk to EJBs using standard interface patterns and also operate inside their own J2EE application client container environment, which aids development and makes implementation of server components a simple task. J2EE servers allow fast development of complex applications capable of withstanding many millions of transactions per second and many hundreds of millions of users.

## **Simplifying Application Deployment**

Leading Application servers today such as WebSphere, iPlanet, Weblogic etc support J2EE and are built with features comparable to Forté in deployment of complex applications. They provide functionality such as load balancing and failover through clustered services, messaging mechanics using Java Messaging Service (JMS), concurrency achieved through Entity beans, easy-to-deploy feature by its Deployment Wizard etc, which ensure successful deployment of enterprise level applications.

## **Enterprise Application Integration**

Prior to the advent of Web Services, Enterprise Application Integration was very difficult due to differences in programming languages and middleware used within organizations. The interoperability was cumbersome and painful. With Web Services, any application can be integrated so long as it is Internet-enabled. The foundation of J2EE is XML messaging over standard web protocols such as HTTP. The service's operations are generally invoked using SOAP (Simple Object Access Protocol) and ebXML, which is emerging as a very important standard for eBusiness collaboration. ebXML is a suite of XML specifications and related processes and behavior designed to provide an e-infrastructure for B2B collaboration and integration. SOAP offers an XML format for representing parameters and return values over HTTP. This is a very lightweight communication mechanism which helps integrate businesses, departments, or applications quickly and cost-effectively by easing interoperability greatly.

### **Forté to J2EE Migration – A good long term business decision**

J2EE is platform-agnostic, running on a variety of hardware and operating systems, such as Win32, UNIX, and Mainframe systems. The portability is an absolute reality today because the Java Runtime Environment (JRE), on which J2EE is based, is available on any platform. ebXML, a very important standard for eBusiness collaboration is experiencing broad adoption from around the world. This includes HL7 (Health Care), OTA (Open Travel Alliance), RosettaNet, OAG (Open Applications Group), GCI (Global Commerce Initiative), and DISA (Data Interchange Standards Association). The support of ebXML in J2EE has further reinforced its credibility across the world. Increasing number of consulting firms, software vendors, application servers like WebSphere, iPlanet, Weblogic etc support J2EE and also have features comparable with Forté.

Here are a few features of Java, which reflect its maturity compared with Forté in these hot areas.

### **Java Application Server**

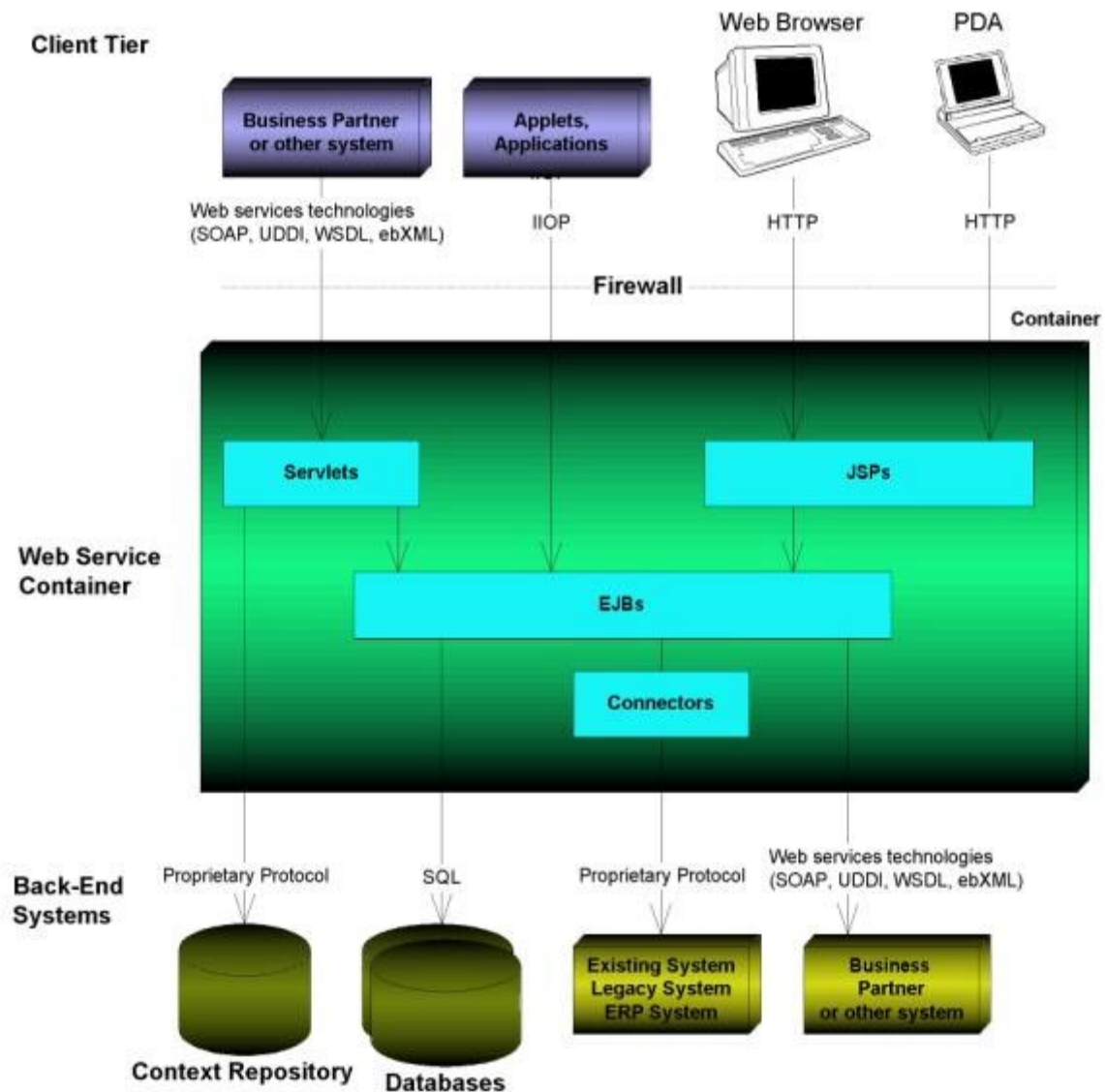
Application Servers like Weblogic, WebSphere, iplanet etc have provided a full-featured application server platform for J2EE. They also provide the basic functionality such as the load balancing and failover through clustered services, messaging mechanics using JMS, easy-to-deploy feature by its Deployment Wizard etc.

### **Enterprise Java Bean (EJB)**

EJB is designed to hook together legacy enterprise systems covered by Java wrappers to form the back-end business logic of enterprise application. Basically, EJB is an extension to Remote Method Invocation (RMI) and includes many new features such as transaction processing, security, object persistence, and resource pooling. For example, Entity beans are persistent across client session and the lifetime of the server.

### **Java Messaging Service (JMS)**

Apart from providing the basic messaging mechanism as Forté does, it goes further to extend this mechanism to the persistent level. The body of messages can be of any type such as text or object, which are stored in the database. If the subscriber is inactive or dead, the message is kept for it until either the message subscriber is activated or the leasing time for the message expires. Its delegation mechanism is also far more flexible, and supports programming patterns such as centralized event managers, event chaining, and so forth.



*Developing web services with J2EE*

### Java Server Page (JSP)

JSP is another technology for accessing server-side Java component directly in HTML pages. Java Servlet overcomes the traditional CGI's "bottleneck". Instead of shutting down at the end of each request, the servlet can remain loaded in memory, ready to handle subsequent requests from Web clients, and it also has the capability of cookie manipulation and session tracking. The important thing is that the servlet can use Java Naming and Directory Service (JNDI), Java IDL etc to perform whatever tasks are necessary to fulfill the request.

### Remote Method Invocation (RMI)

RMI can extend the internal garbage-collection mechanisms of the standard Java VM to provide distributed garbage collection of remotely exported objects. The activation subsystem in RMI provides you with two basic features: the ability to have remote objects instantiated (activated) on-demand by the client requests, and the ability for remote object references to remain valid across server crashes, making the references persistent. It could be used to upload new code to the server, enabling on-the-fly maintenance and upgrades of the services running there.

### **Java Database Service (JDBC)**

JDBC includes support for a variety of enterprise-development activities. For example, it lets you use JNDI for connecting to a database, making it possible to obtain the name of a database from a name service, rather than using a hard coded name. Another key feature is the ability to treat the results of a database query as a Java Beans component. It also offers connection-pooling functionality, so that your application can have a cache of open database connection. It also provides support for distributed transactions utilizing Java transaction Service (JTS).

### **Java IDL**

Sun builds in Java IDL (CORBA) into its Java 2 platform. Apart from the full-fledged CORBA functionality, Java IDL also provides 15 basic CORBA services defined by OMG, namely, Naming, Security, Event, Persistent Object, Life Cycle, Concurrency Control, Externalization, Relationship, Transaction, Query, Licensing, Property, Time, Trading and Collection.

Thus, Java plus CORBA brings a real distributed, interoperable, enterprise-wide, platform and language-independent solution to the enterprise computing communities.

### **Java Naming and Directory Service (JNDI)**

As its name implies, JNDI doesn't just deal with naming services. JNDI also encompasses directory services. The directory service allows the association of attributes with objects while a naming service does not. The JNDI API is general enough to work with any name or directory server. Thus, with a directory service, you can access the attributes of objects and search for objects based on their attributes. You can use JNDI to access directory service like Lightweight Directory Access Protocol (LDAP), Novell Directory Services (NDS) directories and RMI registry.

### **Future Java: Jini**

Jini has the potential to revolutionize the way we build networked software. Jini moves us from a world in which the "system" is the individual networked device, to one so-called "cluster" and "community" (federating and connecting Jini communities into larger structures) in which the "system" is spontaneous, self-heal networking, and the collection of all these devices working together. It will be leading us from Small-Office/Home-Office (SOHO) or Remote-Office/Home-Office (ROHO), all the way up to Enterprise-wide Network Services (ENS) and possibly entire Internet itself.

Jini adds a number of incremental changes to Java to extend it to this new world of lightweight distributed computing, but doesn't require developers to invest in learning entirely new programming paradigms from ground up.

Like Forté, J2EE is largely used to configure relatively static pathway between enterprise software components. The fairly static computing patterns will be replaced by the dynamic computing paradigm based on mobile code, strongly typed interfaces and separation of interface and implementation, code generation on-the-fly, administration-free (zero-administration) setting and installation, self-healing, spontaneous networking.

Sun, along with its partners and the wider Jini community of users, is working to define a set of common interfaces for printers, scanners, cellular telephones, storage services, and other common network devices and services.

Two years ago, one might have said that Forté was the best solution for the enterprise computing, but things have changed dramatically. An application or a product today has little commercial importance if it cannot be seamlessly integrated with Internet, e-commerce, B2B applications etc. Apparently, Forté is out of the Internet race. It is well known that Java is playing a leading role in the enterprise computing, Internet and e-commerce applications. J2EE with Java, Java Enterprise APIs, support form several application servers and enormous third party support has emerged as the technology for future.

## Goldstone - Forté to J2EE Migration Approach

Goldstone Technologies with strong expertise in Forté and J2EE has been working towards providing a smooth transition for organizations that intend to migrate their existing Forté applications to J2EE. With proven Quality processes in place, Goldstone follows a comprehensive Software Development Life Cycle (SDLC) process and extensively uses its own proprietary suite of tools to automate the migration process at every stage of SDLC to ensure swift, smooth and reliable transition.

The migration approach will completely eliminate Forté from the production environment and will not have any application integration with the existing Forté application. It also provides scope for the user to enhance the existing application with additional features. Goldstone's tool based approach in conjunction with the onsite offshore business model can show savings up to 50% over conventional migration processes.

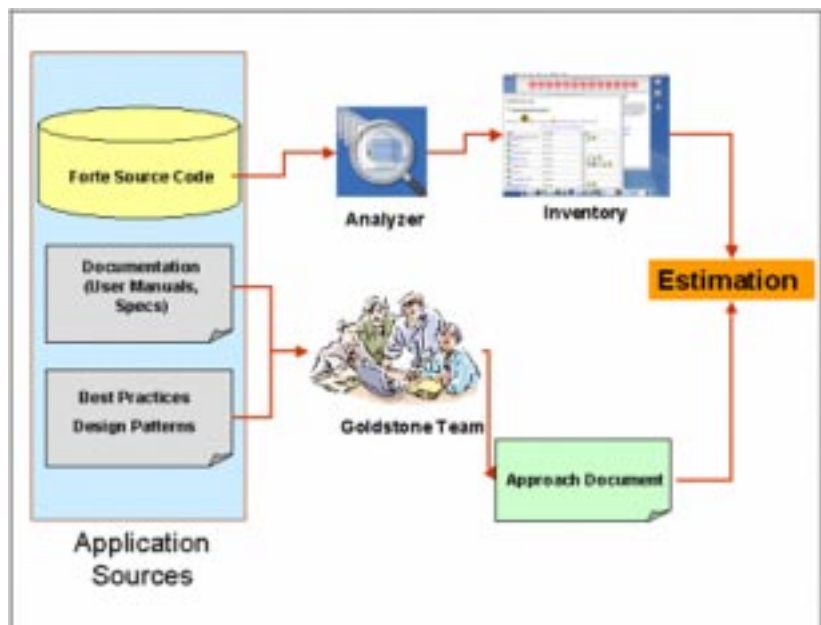
### Forté to J2EE - 6 steps

These steps are best applicable to a Forté application, which has followed the Forté best practices and tiered approach in its application development process. In other cases, the conventional SDLC process is followed.

#### STEP 1 – Planning and Decision Making

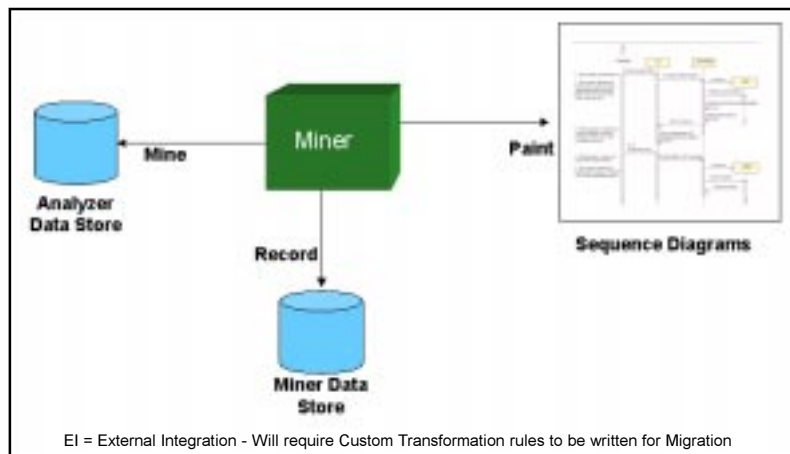
This step helps in having the basic understanding, sizing and estimating the effort required for migrating the Forté application.

Goldstone **F2J Analyzer™ Tool** is used to analyze the Forté source code of the customer's application and create application inventory. The application inventory is used in analyzing the application for its complexity and size for migration effort estimation. The business functionality, technical requirements and performance requirements of the application are analyzed at this stage. Based on the analysis, Goldstone would be able to specify timelines for the migration process. The proposal with the estimates, project plan, deliverables and acceptance criterion is submitted at the end of this step. The estimate also includes any extra functionality to be included in the reengineered application. This step is executed onsite at the client's facility by senior consultants from Goldstone.



#### STEP 2 – Analysis and Design

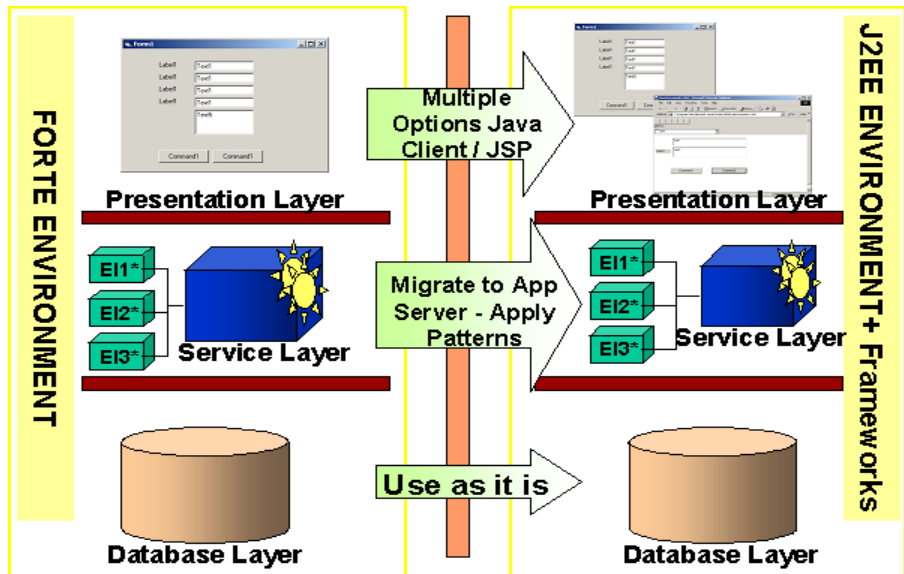
Having identified the detailed requirements from the assessment phase, Goldstone uses its **F2J Miner™ Tool** for Reverse Engineering of Forté applications, which involves the extraction of program specifications from the existing Forté application and generates application flow diagrams (sequence diagrams). These flow diagrams help understand the design of the Forté application, aid programmers to understand the program flow and testers in the Verification and Validation Mechanism for the converted Java Code, thereby reducing the overall migration effort.



**Architecture Design:** Goldstone maintains a repository containing different patterns of designs and architectures that are compliant to J2EE standards. Goldstone after identifying the features and flow diagrams of the existing Forté application proposes the best suited J2EE compliant architecture maintained in the repository.

**GUI Design:** Goldstone uses its **F2J GUI Designer™ Tool** to convert the Forté presentation layer to the equivalent in J2EE.

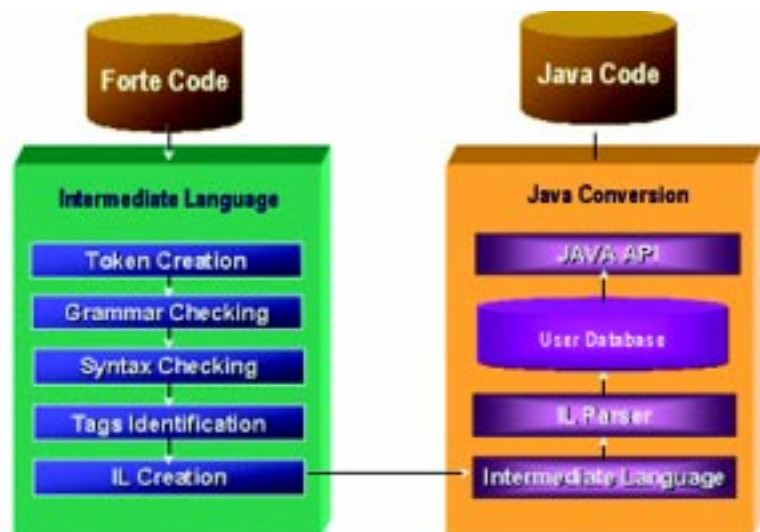
Once the architecture design and the GUI designs are carried out, mapping and transformation rules for the Forté application are created, deployment and test strategies are designed and the transition plan is finally prepared. There is no time spent on the database design and the complete database design in the existing Forté application is reused. Proof of concept is carried out with the client to get the final approval.



\* - EI = External Integrations - Will require Custom Transformation rules to be written for migration

### STEP 3 – Development

Goldstone defines the transformation rules and mapping rules for converting the Forté code to Java code. Most parts of this step are automated, including GUI conversion, using Goldstone **F2J Code Converter™ Tool** which takes inputs from Forté **F2J Analyzer™ Tool** and generates Java code as the output. It converts the Forté classes including UI classes and service objects to Java classes and EJB components respectively. It also provides statistics on the generated Java code. While the **F2J Code Converter™ Tool** takes inputs from the **F2J Analyzer™ Tool**, any part of the code that cannot be converted using the **F2J Code Converter™ Tool** at this step is developed manually using the specifications generated.



Enhancements required in the application are coded and whatever minimal coding that is required for GUI conversion is done parallel to the auto code conversion. External integration and report integration rules are finally coded to integrate external applications and reports respectively.

### STEP 4 – Testing

The complete application is tested using the test cases created at the Analysis and Design stage of the application. After the mapping is completed successfully, the number of classes, methods, frameworks and environment features that are converted are identified to resolve logical errors, functional deficiencies and performance bottlenecks.

### **STEP 5 – Transition Management**

The application that is migrated is deployed at the client's location based on the deployment environment. The migrated application is run parallel with the existing application to allow the client to compare and contrast the migrated application with the existing application and also avoid service downtime. Defect tracking and rectification mechanism is implemented during the transition phase to ensure minimum service downtime of the application.

### **STEP 6 – Post Transition Consulting**

Goldstone provides post deployment technical support and maintenance of the migrated application with dedicated resources. The support could be either onsite or offshore based on the complexity, mission criticality of the application and any specific requirements of the customer.

### **Conclusion**

It is widely agreed that Forté has been a robust and stable product for Enterprise Applications. However, there is also an underlying threat about its uncertain future. In the likelihood of Sun stopping support for Forté, the current deployed applications will not be flexible to accommodate any new changes and future opportunities. J2EE has emerged as future technology and already received wide acceptance based on several merits comparable with Forté. The Forté user community has already begun to move the Forté based applications to J2EE, which assures to protect their investment for future.

Leveraging its successful onsite-offshore business model, Goldstone is helping its multi-platform clients migrate their Forté based applications to J2EE. Goldstone has achieved its success in Forté based applications through its experience in diverse environments including Windows, Unix, Linux, OPEN VMS and Solaris. Goldstone has expertise in platforms ranging from Intel machines to IBM mainframes to Sun Sparc systems. With strong processes, expertise, quality and availability of resources in both Forté and J2EE, Goldstone has been a preferred service provider by Forté users to help them achieve their business goals in the future by ensuring swift, smooth and reliable transition from Forté to J2EE.

### **Contact Information**

**Goldstone Technologies Ltd**

[www.goldstonetech.com](http://www.goldstonetech.com)

[solutions@goldstonetech.com](mailto:solutions@goldstonetech.com)

## Appendix 1

### Comparison Matrix between Forté and J2EE Technologies

Capability	Forté	J2EE
Application Types	Client/Server Distributed Web applications	Client/Server Distributed Web applications Embedded applications Wireless applications
Application Server/Component Container	Components are managed by Forté Environment Manager services	Application Servers like WebSphere, Web Logic, iPlanet, Oracle 9iAs
External Integrations	C bridge CORBA COM	JNI CORBA
Middle Tier Development	Forté Service Objects	Enterprise Java Beans
Supported Back ends	Oracle Ingres Informix DB2 Sybase Sybase CT-Lib ODBC compliant databases	Oracle Informix MS SQL Server DB2 Generic SQL MySQL JDBC-ODBC compliant database
Naming Service	Forté Naming Service	JNDI
Transaction Support	Transactional property set at Object level.	Transaction support through Component Transaction Monitors. Transactional properties set through Transaction attributes that specify how business objects participate in transaction.
Object Visibility	Shared objects are Environment Visible Non-shared objects are User Visible	Entity beans are Environment Visible Session beans are User Visible
Object State mation Retention	Depends on the Dialog Duration set (Message, Transaction, Session)	Stateless bean (Message level) Infor- Stateful Bean (Transaction, Session)
Mail Support	Requires integration with external mail systems	Allows EMAIL operations in a cross-platform, cross-protocol fashion

Capability	Forté	J2EE
Hot Deployment *	Not available	Available and supported by the Container
User Community	Small	Large
Key Application Domains	Telecomm Finance Banking Insurance	Several Vertical Domains
Application Development and Deployment tools provided by vendors	Single vendor	Multiple vendors
Internationalization	Support through the Display library and Message Catalogs Support for recognition of Japanese characters through TOOL code	Support through Resource Bundle class and properties file

\* In an application runtime environment where multiple server components are installed, the possibility of bringing down a server component and bringing it up after modification of the properties/code without affecting the other deployed server components.

## Mapping of Forté GUI controls to Java controls

<b>FORTÉ</b>	<b>JAVA</b>
Push Button	Jbutton
Picture Button	Jbutton with image
Toggle Field	JtoggleButton
Scroll Bar	JScroll Bar
Outline Field	JTree (TreeNode)
Tree View	Jtree
List View	Jtable
Picture Field	JLabel with image
ArrayField	Jtable
Tab Folder	JtabbedPane
Text Field	JText Area
Text Edit Field	JText Area
Data Field	JText Field
FillinField	JCombo Box
Radio List	JCheckBox, JButtonGroup
Scroll List	JScroll Pane
Drop List	Jlist
Picture Graphic	Utilising JFrame
Text Graphic	Jlabel
OLE Field	No corresponding control
Active X Field	No corresponding control*
Domain Widget	No corresponding control

\* ActiveX fields use third party controls for re-usability. If the re-usable components are available in Java then direct mapping of ActiveX components to Java is possible.

## **Appendix 2**

### **References:**

1. Sun Microsystems: [www.sun.com/forte/](http://www.sun.com/forte/)
2. Forté Technology Series from Forté Software, Inc (Sun Microsystems)
3. J2EE vs. Microsoft. NET, a comparison of building XML based web services for Sun Microsystems:  
[www.theserverside.com](http://www.theserverside.com)
4. Forté User Forum: <http://lists.cuenet.com/forte-users/>
5. Building Java Enterprise Systems with J2EE, Authored by Parrone, Venkata SR Krishna & R Chaganti. Samtech Media Publications.