

Project: FiloSAFE Enterprise Suite v.3
Employer: FILOSAFE Corporation, Canada
Term: August 2000-December 2005

1) Application of theory

a) Analysis of requirements

FiloSAFE v.3 is an extension of the previous versions in view of general features; however, its internal structure needed to be completely re-designed, since we found various mismatches with the older versions when we introduced collaboration features to the older versions.

FiloSAFE v.3 delivers secure online collaboration spaces that provide online storage, collaboration space, and remote application access over our own public key infrastructure (PKI) and duty-oriented access control.

The major advance of this version is how we handle the access of users to data. We defined a set of access rules to data based on a security policy, a variation on the Clark-Wilson model. All the access to data in FiloSAFE follow the definition of this access rule and users basically see and manipulate data depending on their duties in the organisation rather than with the binary access rule of superuser-and-user structure of most of commercial operating systems. This is what we call a duty-oriented access system.

b) Design and construction

We designed the system on the foundation of Java 2 Enterprise Edition (J2EE) by following its framework specification so that the system can deliver a scalable, reliable, modularised, and flexible service.

The architecture of this system has four tiers including a client-tier, web-tier, business logic, and enterprise information base. We provide two client interfaces: a web browser interface and a Java client interface. The web-tier is implemented with XML, XHTML, and servlets. The business logic is implemented in Enterprise Java Beans (EJB) including session beans and entity beans. The functions of the enterprise information base are performed by a relational database such as Oracle, SyBase, MS SQL Server, or MySQL. Since we need to support different databases according to the preference of the user, we have extracted common query commands which can be universally deployed in these databases so that we can minimize the portability gap.

As a benefit of Java, FiloSAFE enjoys easy portability to many systems; it currently works on HP-UX, Sun Solaris, Linux, and Microsoft Windows 2000/2003 server.

Since we deal with international clients, FiloSAFE currently supports three languages, English, Korean, and Mandarin Chinese. To achieve this, we designed the system following the rules of internationalisation and localisation defined in Java so that the locale-dependent part can be separated from the internal logic. When we need to support another language, we just need to fill up localisation files in the language.

c) Quality control

Since it is a complex system with several subsystems working together, the testing also requires a complex lattice of cases. In each version release, we apply a series of test cases and collect bugs. We also set up a bug reporting and analysing system to make bug fix easier and faster. This system is based on Bugzilla, an open-source defect tracking system, and we can keep track of each bug and see how a bug has been dealt with.

d) Asset management

We register each version of software source code and documentation of design and implementation. To keep all the revisions of them, we run a Concurrent Version System (CVS) server, and when we share a part of our code with international collaborators, we allow them to make a link to our CVS server from their CVS server.

As a standard J2EE application, FiloSAFE works with a Java application server and a database. To meet the various demands of users, we support both commercial and free application servers including WebLogic (BEA), WebSphere (IBM), Oracle 9i AS, Orion, and JBoss. The databases that we are supporting include Oracle 9i, SyBase, SQL Server (Microsoft), InterBase (Borland), and MySQL.

This creates a number of combinations and the versions for each combination need to be maintained. Apart from the revisions of the main module, we also control the versions of interface modules for each combination.

Since we release our software in an automatic installation manner, we package the client software with InstallAnywhere by Zero G and the server software with our own script for easy installation.

2) Practical experience

This system is currently used by financial firms, such as banks and insurance companies. We are testing this solution for a large corporation with over 10,000 concurrent users. We are verifying that our conceptual model works under practical situations in an organisation dealing with a large amount of data exchange.

3) Communication skills

Since this project was kept updated by end-users' requests, the communication between the end-user and our development team was crucial to complete the project successfully. It is very efficient and beneficial that we write down verbal agreements and discussions and get them confirmed by the user and the development team at every stage. Otherwise, we may suffer confusions and unexpected results.

4) Social implication of engineering

This product made the users realise a way to manage their data efficiently outside of their organisation without sacrificing data security. This realisation can be shared with other organisations that are in a similar situation, and this will make their work environment mobile and more flexible. A healthcare company based in New York, NY has deployed this solution to protect privacy of the patient information. This solution can be deployed in the biotech industry as an information repository infrastructure. A Korean bank is using this solution to share their information seamlessly with their financial consultants who travel very often; this solution demonstrates a safe way of building a mobile working environment.

Project: FiloSAFE v. 1 and v. 2

Employer: Iaso Research Centre, Korea

Term: February 2000-August 2000

1) Application of theory

a) Analysis of requirements

FiloSAFE was a secure online storage solution until its version 2, and it is a fully web-enabled service. It allows users to upload their files to online file storage through a web interface so that users can access files from anywhere and at anytime. It is a sort of a hard disk on the web. Owing to the limit of implementation time and lack of highly qualified developers, we could not extend FiloSAFE's functionality to the level of commercial collaboration solutions. It triggered the redesign of FiloSAFE in version 3.

b) Design and construction

We focused on building reliable uploading and downloading functions, since they are the most frequently used functions and file transfer errors could be critical. Another issue was to define a folder and file indexing mechanism to find a requested file in a given response time.

Since this service is designed for mass users, load balancing is an important issue. We designed a modularised architecture so that we could distribute function modules to multiple servers based on the load on each module.

c) Quality control

Since this was a public service, we established a group of beta testers and let them use our service aggressively. We found many bugs and received valuable advice from the beta testers.

2) Practical experience

Since FiloSAFE is designed as a public service, there were many points at which its reliable operation broke down. We could experience how large traffic can affect our system and how important a flexible system configuration and easy administration functions are.