

On-the-fly Storage Management



Xoriant helps a Silicon Valley start-up provide on-demand online storage solutions to the consumers.

Background:

As a part of the current trend of cloud computing and software-as-a-service of the enterprise market, our client, a Silicon Valley based start-up founded in 2006, is delivering flexible, secure and limitless storage capabilities to mobile users. Remote storage has become an inexpensive and on-demand solution for the small and home businesses. Our client's product delivers an all-in-one file server solution that combines unlimited, on-demand file storage and workgroup collaboration capabilities with an automated, online backup. This product works on both Windows and Mac platforms.

Re-engineering by Xoriant:

Before the client engaged us, their website was based on a traditional web approach involving full-page refreshes on a single click. This resulted in significant lags and delays. This was a big turn-off for the site visitors. In addition, their feature quality, quantity and variety left a lot to be desired. This made their site much less attractive, resulting in low repeat visitor rates. Xoriant was engaged to resolve some of the issues and help create better modules that would enable enhanced user experience.

When we engaged with our client, we realized that the small client team had created code that lacked proper documentation, planning, and transparency. After a quick “code walk-through” with the client development team, we took on the onus of cleaning up and formalizing the existing code. Once this was realized, we proposed that the entire product be rebuilt using current web tools like Ajax, so that the user experience could be improved significantly and the product matched the market-trends of the Web 2.0 style of deployment.

While re-engineering the product, Xoriant delivered:

- A full-fledged web 2.0 based content management solution
- Features which include seamless desktop-web backups
- Windows-like file/folder explorer on the web
- Access control for maintaining security for the users/groups
- An Ajax-based site with webDAV for creating the file/folder structure that could be accessed over the web

We deployed a team with a skill-mix of Java, Ajax & Python technologies, implementing the web UI using DWR framework for Ajax that enables client-side javascript to directly interact with the server-side Java code. Python was used to build the desktop uploader application versions for both Windows and Mac.

Credit Card Integration:

Before Xoriant was engaged, the payment processing on this website was purely manual. To make it more industry standard, credit card capabilities were sorely needed. We designed, implemented and tested a fully functional credit card processing facility using which our client's users could pay for the service subscription using standard credit cards.

The UI part for credit card information was developed in JSP. To be able to reliably authenticate and authorize the credit card details, this payment module was integrated with PaySimple (www.paysimple.com) gateway.

Incremental, rapid, iterative prototyping:

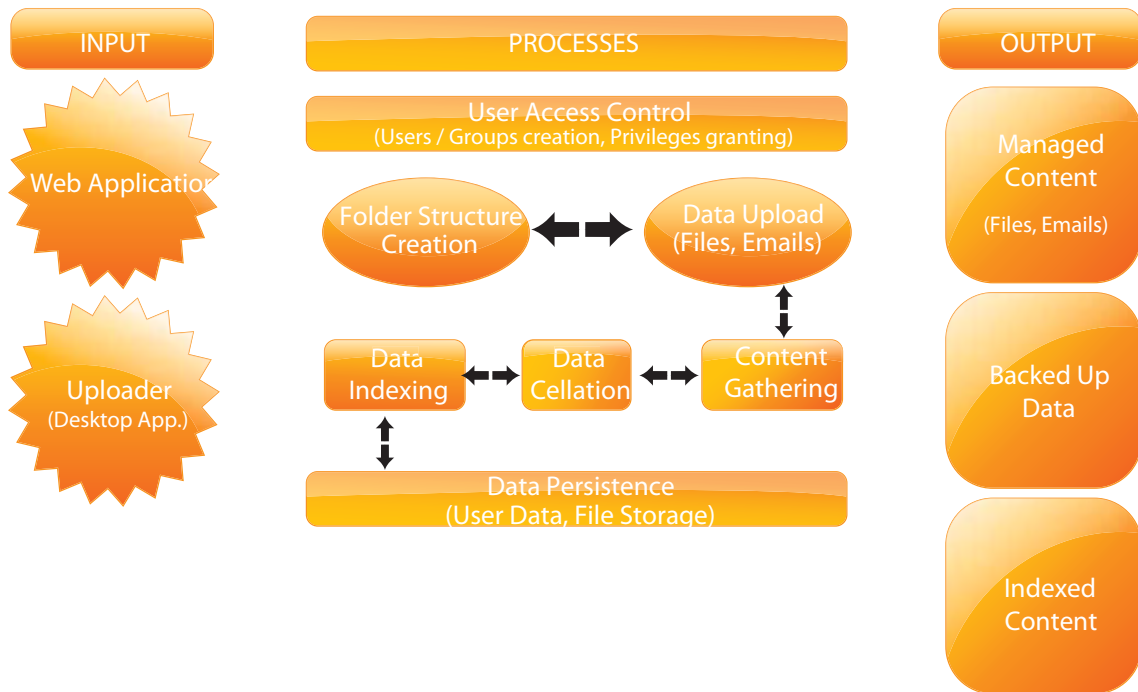
Since the product development was key to the client market launch, Xoriant adopted a weekly release methodology to create and implement a full suite of modules, patches, and features that responded to user feedback, and client integration requirements. In this, we achieved high success in seamlessly delivering the product user experience over IE, Safari, and Firefox, substantially increasing the client footprint.

In the course of the engagement, the Xoriant offshore team worked in synch with the client's U.S. development team to define the architecture, the overall feature sets, and the finer modules delivery framework. By overlapping time-zones, we could work in real time with the client to deliver the right technology within the right schedule and market launch plans.

Today, the client presents a robust, on-demand, user-adopted storage solution that is seeing significant success in the market place.

Client testimonial:

“I wanted to thank you for the great work you have been doing. In the frenzy of next release, bug fixes and everything required “yesterday,” sometimes we all get so caught up and not realize and appreciate the hard work being put in by the team. I wanted to express my appreciation and thanks to you.” The client CEO



The application is composed of 5 broad modules:

1. Web UI Module (Client Web)

- o The client website that is built on AJAX (DWR) Framework and Java.
- o It facilitates content management and user access activities online.
- o The web application is hosted on Tomcat application server and Apache is used as the web server.
- o This module communicates with the Server module and Database modules.

2. Desktop Module (Uploader)

- o This is the client-server application that is installed on the client's machine for seamlessly uploading content (files/emails) from the client machine over to the remote server (web).
- o The CherryPy Python Package is utilized to serve the HTML pages via the CherryPy Server. Python scripts are employed for database related activities like user authentication, content tag updates, and so on.
- o Periodically, this module
 - Extracts content from outlook and configured folders using a Content Crawler Service,
 - Creates a zip archive and transmits via SCP (secure copy) to a CrawlerConfigSync Servlet (Java) for processing and synchronization using a Polling Service.

3. Server Module

- o This is responsible for Creating and maintaining search indexes that help in quicker searches.
- o The components involved here are the Indexer and the Searcher services. These are deployed on top of the Spring framework.
- o The Indexer service is used for creating the search indexes.

4. Content Gatherer Module

- o This gathers data from all data sources for the indexer service to index the content.
- o This module employs the ContentCrawlerService and is built using Python and Java.

5. Database Module (LDAP)

- o This is responsible for persisting the user and module configuration information into an LDAP database.
- o The LDAP Factory acts as an interface between the business objects and the database.

All the content that is uploaded either via the web or the uploader (client-server app) is delegated to WebDAV. WebDAV is a protocol used for publishing and managing content to web servers.

Within the architecture, a host of design patterns have been utilized. The Observer and Command design patterns have been used in the Content Gatherer Module for polling and notification purposes. BusinessDelegate and ServiceLocator design patterns are used in the Web UI Module for locating and executing the appropriate business logic functions. Finally, the Façade design pattern has been used in the Web UI Module for executing the AJAX calls.

