



Stingray™ Solutions

At Qwest: Java helps speed up new product introductions

Highlights

Qwest Communications: A multi-billion dollar diversified telecommunications carrier

Business Challenge: Improve time to market for rollout of new backend IT systems and services

Solutions: Reengineer legacy terminal applications with Java-based data access tool

Hardware/Software:

- Two UltraSparc Enterprise four-way servers
- Stingray software
- Solaris

Key Business Results:

- Enabling integration via Java programming language for mainframe/client interoperability
- New, flexible network orientation supplements existing IT infrastructure
- CICS more accessible via Java-enabled front end
- Reliability, scalability and performance of Java-enabled mainframe access solution

Delivering innovative, integrated and cost-effective solutions has helped build Qwest Communications into one of the largest telecommunications companies in the world. New product revenue, growth in customer access lines and continued demand for custom calling services helps the company maintain its leadership position in the fiercely competitive telecommunications industry. New product divisions, including wireless PCS, interLATA long distance, expanded data network services and video, are expected to add billions of dollars in revenues over the next five years.

However, to effectively penetrate these new markets, Qwest needed to dramatically reduce the lead-time for introducing new products and services. Its legacy Bellcore customer and network information systems, some of which were at least 30 years old, posed formidable obstacles to introducing new products and services to its customer base. Qwest required a new solution to help it keep pace with the new time-to-market requirements. Today, the company leverages existing resources using a Java-based legacy data access tool and hence, is able to speed up new product rollouts.

Taking a leadership position

With customers anxious for faster Internet connectivity, taking a leadership position in the rollout of Digital Subscriber Line (DSL) data services was a priority. While the company data network was in place and ready to go, new product deployment also meant having the back end IT provisioning systems in place to support the new service rollout. Provisioning systems manage all critical customer and network information required to set up a customer with a new service, encompassing inventory, telephone lines, and circuits. With the introduction of each new customer service, there are new business rules for the provisioning of that service, which generates the need to develop a new application.

Qwest's back end, IBM MVS-based, IMS transaction systems predate the break up of AT&T. These systems contain decades-worth of customer and network-related information. When the company decided to inaugurate DSL service, it had a two-month development window to put its new back end systems in place. The company's IT team required a solution that would shorten development time while also providing scalability and reliability.

Until the DSL project, Qwest's IT team had used PERL to develop provisioning systems. As a sequential scripting language, development time using PERL takes anywhere from three to four months, and the code is not easily maintained.

There were additional shortcomings to using PERL, according to Kevin Sun, system architect at Qwest. PERL applications written on a different Unix operating system would have to be ported to operate on Sun Microsystems's Solaris operating environment. Furthermore, the PERL applications had scalability limitations. Facing a tight deadline for the DSL service rollout, it was clear that using PERL was not an option.

Another alternative, C or C++, was also out of the question because typical application development cycles required even more time - six months, in many cases. Further drawbacks included the need for highly skilled developers, and the difficulty of reading and maintaining the code.

Given these obstacles, the IT group looked at Java. "We started to evaluate and prototype Java before the DSL project was on the table," said system architect Sun. "Java's object orientation made it an inviting alternative." For starters, it meant that the code would be more maintainable, easier to read, and extensible.

Finally, productivity would be high. Developers would not have to reinvent the wheel, constantly reprogramming the same functions or logic, for a number of reasons. First, Java's object-oriented nature encouraged reuse. "We could reuse the legacy access code in future applications or in the same project," said Sun. Additionally, the rich JDK class libraries eliminated the need to write many routine functions, such as visual features for painting end user screens.

The Java Solution

As a strategic solution, Qwest selected Stingray as the tool for legacy system access because Stingray allowed their developers to create Java based applets and applications to communicate with any mainframe application through 3270 emulation. Essentially, Stingray enables corporations to link the Web browser to data locked away in corporate mainframes.

The Stingray product was chosen because of its simple recording and Java code generation model. Qwest's developers simply used the Stingray SDK's Terminal Recorder to connect to the mainframe, and record sessions that encapsulated the screen navigation and data exchanges for transactions with their 3270 applications. The developers then used the SDK's Code Builder to generate Java classes from the recorded sessions. These classes, known as Legacy Business Objects™ are 100% Java and enable access to the recorded transactions in either client or server side implementations. According to Sun, the product was easy to use, provided a simple interface and was reliable.

The next issue was getting the right platform in place. "Moving forward, we knew it was time to change development platforms," said Sun. Qwest was hitting the wall with its existing server, which couldn't support a symmetrical multiprocessor architecture.

The company purchased two Sun Microsystems UltraSparc Enterprise servers, each configured with four CPUs and the Solaris operating system. Qwest's Sun reported that the overall performance of the Sun servers was better than their previous server.

Additionally, Qwest relied on Sun Microsystems to get the best out of Java. "Aligning ourselves with Sun and Solaris enabled us to get the latest Java technology and a better performing Java." said Kevin Sun. He noted that the native threading capability in the Java version resulted in fast performance.

While the Sun Microsystems and Stingray solution provided Qwest with the development architecture it needed to move forward, the IT team required staffing resources to meet the impending internal project deadline. Qwest turned to Sun Professional Services (Sun PS) to provide software design, Java coding support, Java code review and Java technology mentoring. Aided by Sun PS engineers, Qwest created the DSL provisioning system in six weeks.

The Results

The result was a faster, more compact, easier to maintain system. The Java application required only half the number of lines of code compared to what they would have written using PERL, and delivered ten times the performance. Qwest's Sun attributed the performance advantage, in part, to Java's thread-safe feature, which

allows programmers to start multiple concurrent sessions to the legacy system, thereby running multiple processes.

There were additional substantial benefits as well:

- Qwest reaped savings in development costs because Java shortens the development cycle and eliminates the need to purchase third-party libraries
- They also realized business savings, literally hundreds of thousands of dollars, by being able to program the business rules into the application rather than having to write these functions into the applications manually
- Code reusability
- The ability to service customers faster
- Application scalability

Using a Java solution allowed Qwest to achieve its desired goal. “The business can get its products to market ASAP, to generate revenue, and customers get the services they want.” said Sun. And, thanks to the Sun Microsystems Solaris platform, Qwest can be assured that no matter how successful their new DSL businesses become, they have a computing environment that will scale with their growth.