

web analytics tools



Xoriant helps a web analytics company retool and build new components for better intelligence reporting to end-customers.

The Client

Our client is a Silicon Valley based eCommerce solutions provider, with comprehensive portfolio of solutions which improve the effectiveness of ecommerce and e-marketing activities of small to medium online retailers. One of our client's solutions allows the e-marketers to precisely measure the effectiveness of their promotions as well as their creative campaigns, thus minimizing the guesswork. This solution enables an online retailer to present the most precise targeted content to the customers and automatically keep track of which combination of choices leads to the greatest "conversions." Our client's solution enables e-marketers to run different combinations of creatives using a simple, easy-to-use web interface. Marketers can simply login to the hosted web site, define their tests, and then check the results, examining the best combinations of offers that yield the highest business metrics.

Client Challenges

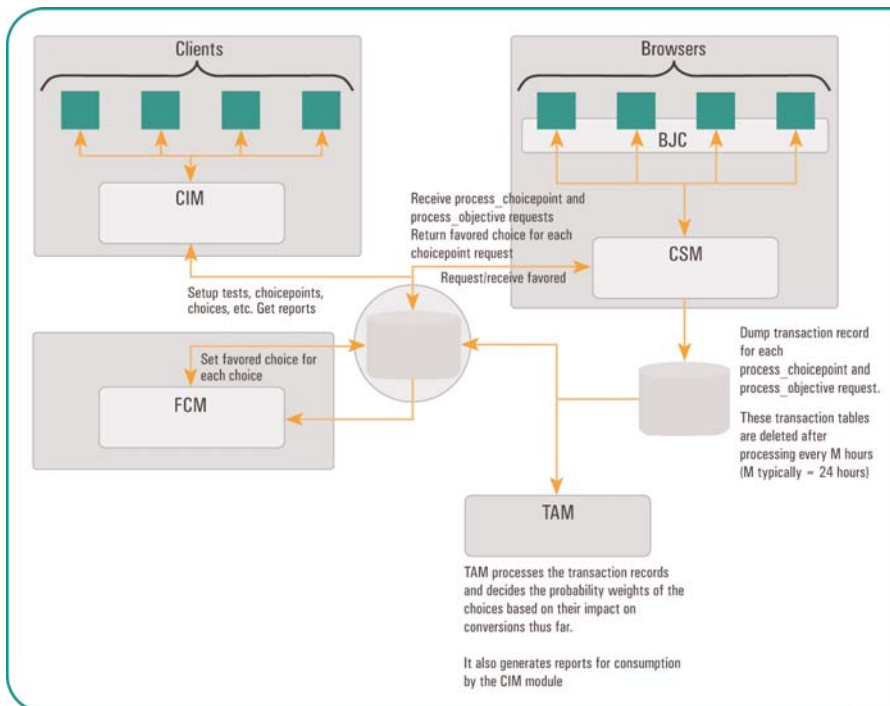
Our client's current solution had good data gathering capabilities and high class usability, but lacked a comprehensive and flexible decision making module. Our client's customers required information that would enable their decisions about the image and the text to be posted on a website for better hits. Arriving at a comprehensive decision model required robust analysis of multiple and changing permutations of different decision parameters. The project required modeling and programming the input process of these parameters and designing and implementing the decision making algorithms, which would then be adaptable to incorporate the changing parameters.

Xoriant Solution:

Xoriant team architected, designed, and built a fully functional system with high scalability requirements using Microsoft Internet product suite. Xoriant team spent considerable time in

studying the current systems, and different use cases to understand how typical users leveraged the different product features to make decisions. Our teams developed the domain and statistical expertise required to build the product. Based on the observations, we worked with the client to build an exhaustive requirements document. Following this, a multi-talented team performed classic product engineering tasks encompassing architecture, design, development and testing. Our approach included:

- Detailed real time participation with the client, to understand the actual needs and concept behind the product.
- A mix of both client server and web based architecture that achieved mutually orthogonal demands of ubiquity and network performance on one side and security and connectivity on the other side. The seamless interoperability between MS-Windows client-server and .Net internet environments made this mixed architecture functional.



- Partner approach with experts in design and architecture followed by product engineering by the Xoriant development team.

Architecture/Data flow diagram

Some of the highlights of the architecture were:

- **Layered approach:** The solution was broken down into different layers of class libraries to facilitate scalability and speed. There was a unique set of class libraries at the database level that provided a standardized access point to other modules and guaranteed efficient usage of processing power. Another set of class libraries was built to encapsulate the business logic. These libraries acted as links between the database server and the client and made the system flexible enough to accommodate any changes in business rules.
- **Use of dynamic handling of javascript calls:** Calls to any javascript files were routed to the ASP.Net handler process instead of the web server. This allowed us to echo out dynamic javascript code to the client browser to generate specific content.
- **Caching:** The javascript content that was sent to the client browser was cached at the server to minimize the database calls required by the request from the same client during a session. This also helped in serving a unique creative to client throughout the session that in turn helped in avoiding confusion on user's end.
- **Windows services for resource hungry processes:** We took out the routines requiring higher processor power and implemented them as Windows services. The services were executed during the off peak hours when the server had more resources available. This also allowed us to

make any changes to the processing routine without endangering user data or without affecting the performance.

- **Queuing of data:** In order to minimize the time taken by the server to respond to client's request for a javascript file, we implemented a queue where by the data fetched by the user request was put in a queue before sending it to the database. This saved a few milliseconds of each request being served by the server and greatly helped in increasing overall responsiveness of the server calls. We designed our own queue classes leveraging the built in .Net functionality that saved potential investment in a separate queue server.

- **JavaScript:** Echoing out a unique javascript code to be included on client's webpages that even allowed other third party javascripts to be included was a challenging task. Our engineers came up with unique javascript code that was easy to integrate in the client's web pages.

Implementation approach

The initial development was done on Xoriant servers in India. This helped our client to visualize the product and suggest changes at early changes. This resulted in major cost savings to our client in terms of logistics.

Client Benefits

Our client's solution was enriched by a decision-making module with a proven and tested algorithm that helped our client's e-marketing customer plan the emarketing strategy more definitively and profitably. While keeping the usability of the solution in tact, we were able to implement very complex statistical algorithms to increase the accuracy of the decisions, while completely hiding the complexity from the user.