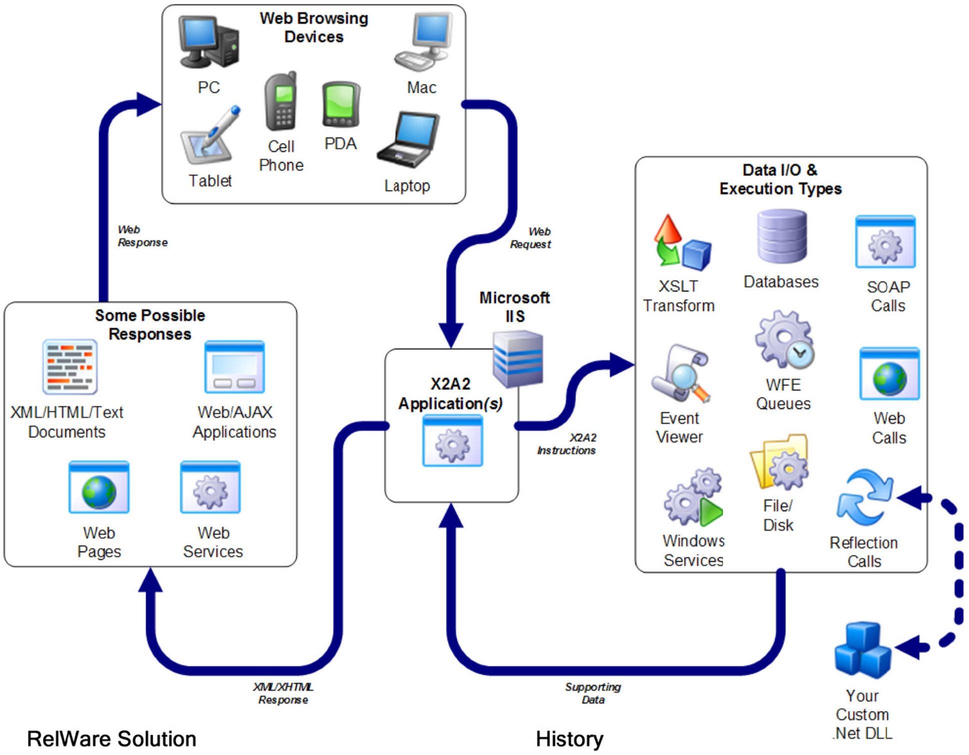
X2A2

“Vision is the Ability to See What is Possible Before it Becomes Obvious.”

-RelWare Development

Reduce Your Programming Complexities by Moving to Our XML/XSL Application Architecture



RelWare Solution

The X2A2 is a lightweight, yet powerful application engine that is the core of our Web and workflow engines. It was built using XML from the ground up, and has been RelWare's primary development platform for the past four years.

RelWare's constant commitments to technology and excellence arrived on this patent-pending programming methodology in 2003. We realized then that XML was the data medium of the future, and had already known that the Web would continue to be the delivery mechanism.

The X2A2 merges these into a powerful, flexible engine, with built-in and external connectors that give them the ability to do almost anything.

History

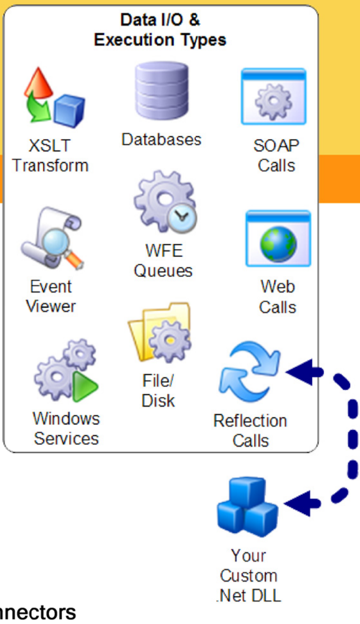
Originally called the X2A, or XML/XSL Application, we wrote a single ASP page that programmatically was able to parse XML configuration files and perform different actions based on different URL parameters.

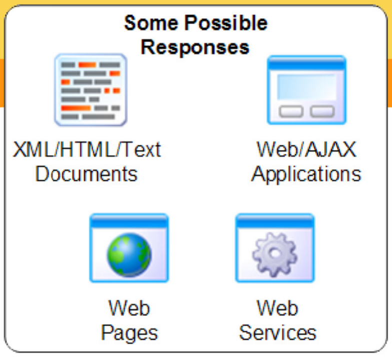
While simple in concept, the X2A allowed us to build an intelligent form system based on it. The X2A ran a live CPOE *(Computerized Physician Order Entry)* system, using XML files to store intelligent forms and their supporting data *(Quizlets)*.

The X2A was entirely redeveloped in .Net *(1.0, 2.0, and now 3.5)*, as a complete engine and named X2A2, as it is no longer a single application, but is now truly an application architecture.

OneRecord™

RelWare®





Web Browsing Devices
PC
Mac
Cell Phone
PDA
Tablet
Laptop

Connectors

The connector (as well as its related mappers) gives the X2A2 its ability to interact with the outside world. Every X2A2 application can be configured to use (or not) any of the available connectors. Those connectors are well defined endpoints of information that are accessed using Mappers.

Connectors are methods of performing actions which most likely come from an external source. When executed, data is returned from the connector to the application as XML for further processing or for display.

The most familiar of our Connectors would be our Database mapper, which provides the ability for an X2A2 request to run stored procedures or straight SQL, either "hard-coded" or programmatically generated.

Obviously, databases are only the start. The X2A2 has File/Disk connectors, which provide basic file I/O operations. Queue connectors allow reading/writing of Microsoft Message Queues, which can trigger asynchronous events. Event Viewer connectors can read and write to Microsoft Event Logs, which can then be further monitored via SCOM. Windows Service mappers allow X2A2 applications to start, stop or pause other services, which is priceless in an enterprise environment. Of course, the X2A2 also has SOAP and HTTP mappers for making both formatted SOAP calls, or simple Get/POST commands.

The real innovation, however, lies within the Reflection Class mapper, which provides an X2A2 application with the ability to call functions within your own hand-crafted .Net DLL.

Response Types

The X2A2 is an XML/XSLT engine; however, it has several potential uses and front-ends. Obviously, this leads to the initial thought of webpages, but that is only the first step.

At the root, the X2A2 is an XML processing engine. So, obviously, XML documents are one possible result. However, this alone provides several potential uses. It can act as an XML document repository, providing different stored or dynamic XML documents based on your desired scheme.

This leads to the next obvious evolution, which is a Web service, where XML payloads are the expected result of different "function-calls". These are simply URL *(or POST data)* that are given to your X2A2 application for processing.

Webpages as well as Web applications become the next step up in complexity. With these, the X2A2 uses the URL parameters to determine which page and what "flavor" of page *(or page portion)* should be displayed.

X2A2 applications can configure their requests to return HTML, XML, or another MIME data-type. Using AJAX calls for XML payloads, in conjunction with standard HTML Web requests, allows for the building of complex AJAX applications in X2A2.

Further, with the continued adoption of XML as a standard data-storage platform, more and more document types *(e.g. Microsoft Word®, and Microsoft Excel®)* can be generated on-the-fly by a real-time Web request.

Performance, Scalability and Stability

Performance and Scalability were two of the primary motivators for us in creating the X2A2 engine. We wanted to build an environment that would allow us to scale up or scale out, but would also provide enough performance in even a small low-end server.

Performance counters were in the design from the beginning, and those counters enabled us not only to actively monitor performance, but help us continually maintain our goal of sub-second response times.

A key feature of our connectors and mappers is that they provide a base level of consistency testing that is performed on initialization, allowing the application to verify its stored procedures before it even starts.

Our engine utilizes unit tests as part of the core build, allowing us the freedom to move from one .Net version to another with utmost confidence that no functionality was ever lost.

About RelWare Technology

At RelWare, we realized in 1998 that the Internet, and specifically, Web-based technology was the future. We built our company with the continued motto: "Every application is a Web application." This meant that anytime we looked at writing a new application, we first asked ourselves, "Can I write this as a Web application?" What we soon found was the answer was invariably "Yes". . .every time.

It has been 10 years since we started, and we are still saying "Yes". . .every time.

Rel Ware®
www.relware.com
877-735-9273 toll free