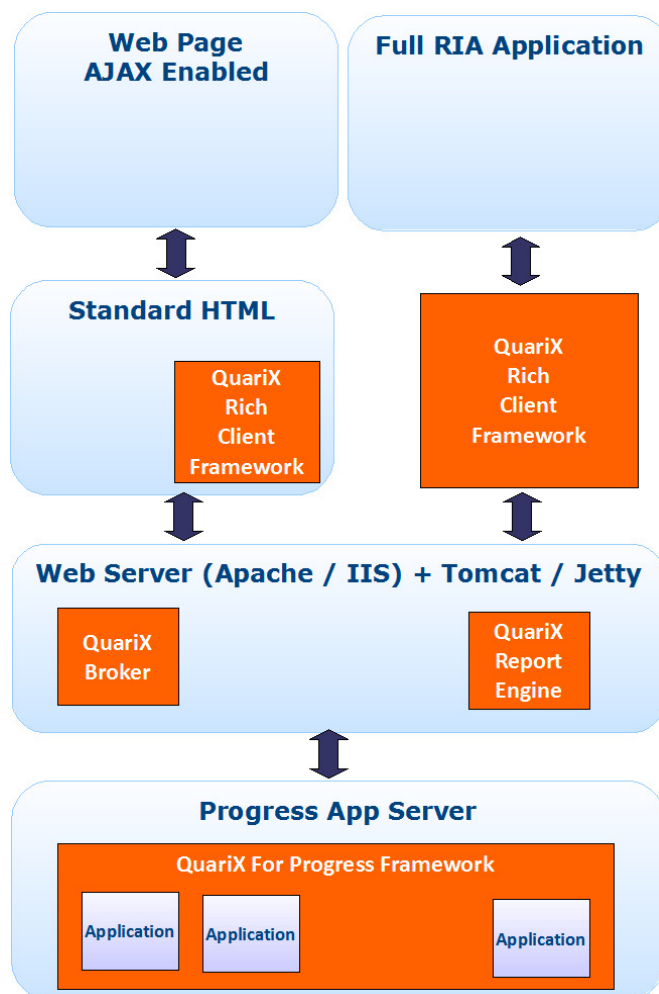


## Yonder QuariX Solution

The deployment of business applications has, generally, forced solution developers to choose between two models: **“thick”** desktop installed applications or **“thin”** web browser based applications. Thick applications are great for their robust features but are expensive to install and deploy. Because of that and for obvious mobility needs, businesses today prefer web-accessible applications. However, the traditional web applications are limited in functionality because HTML and HTTP, the technologies traditional applications rely on, were created for marking up documents and forms, not for authoring and deploying software solutions.

### QuariX framework



The permanent goal of such solutions is to speed-up access to information with productive applications while keeping costs low. The technology already existed in the web browser, especially in asynchronous JavaScript and XML (AJAX) thus eliminating the need for client side plug-ins, applets and software installation and maintenance.

With **QuariX**, an AJAX based framework, enterprises get the best of both worlds: the low cost of ownership of web applications and the rich graphical user interface features, speed and performance of thick clients.

Progress Application Server running in stateless mode sustains the backend, but other technologies are possible.



With QuariX you can develop the UI of your applications through direct XML widget declaration, the upcoming QuariX Eclipse IDE or an industry standard UML for functional design through the MDA/MDG module.

With the QuariX MDG Technology for Enterprise Architect rapid prototyping of applications becomes possible directly from UML use cases thanks to a forward engineering engine able to actually produce code from the abstract, human-elaborated specifications.

Improved rich application usability with no deployment or scalability hustles is what QuariX tries to accomplish in a world where geographically distributed tends to be the norm rather than the exception.

Quality in all processes. TSS and Nethrom are offering their customers all the comfort and security of a large software company. With a uniform and excellent back office operation, the customer can be sure he will be served in a professional way in all areas.

## QuariX reduces TCO

(Total Cost Of Ownership)

### Eliminates distribution and installation costs

- ◆ QuariX based applications are web based so they inherit the ability to be updated and maintained without distributing and installing software on potentially thousands of client computers. While inheriting all the benefits of the traditional web application QuariX inherits none of their drawbacks by capitalizing on modern AJAX technologies that allows for more interactive, complex and efficient applications

### Increased productivity

- ◆ Compared to development platforms that offers similar benefits, Java in our example, we calculated an almost 300% productivity increase
- ◆ Automation of CRUD operations
- ◆ Repository database for users profiles and preferences, authentication, application configuration and more
- ◆ Special tools based on Eclipse where designed and built in order to further increase productivity: a WYSIWYG user interface designer, an integrated rapid development environment and a graphical report designer
- ◆ With the QuariX MDG Technology for Enterprise Architect rapid prototyping of applications becomes possible directly from UML use cases thanks to a forward engineering engine able to actually produce code from the abstract, human-elaborated specifications.

### Lower support and future development cost

- ◆ QuariX is built on standard technologies and browser features that can perform as specified regardless of the operating system or OS version installed on a given client. Rather than creating clients for MS Windows, Mac OS X, Linux, and other operating systems, the application can be written once and deployed almost anywhere.
- ◆ The framework includes design-time or run-time rendering capabilities that can eliminate the need to create, compile, deploy, or maintain the source code for objects represented and stored in repository and also allows for easier porting and conversion of the application to new technologies that the future will demand.



## QuariX Business Case

Yonder provides QuariX based Rich Internet Application development to companies in a number of clearly defined steps.

### Step 1 – Requirements Analysis and Definition, System Assessment and Overview

Understanding the specific business and technology challenges our client is facing, gaining a comprehensive understanding of its situation is foundational to everything else we do. Understanding and documenting the most important components, functions and features, the security, hardware and software environment allows us to see the big picture of the project and the vision of the target product or system.

Deliverables: General system requirements and assessment report with specific recommendation for project development approach and estimations
---

### Step 2 – Functional Specification

Based on the general system requirements and other documents as input, in this stage the target is to **determine, define** and **document** what exactly the system must do and what are the premises for implementation. Streamlining the development process the functional specifications details what the finish product will do, how the user will interact with it and how will look like. By creating a blueprint of the product first, time and productivity are saved during the development stage, it will also enable to manage the expectations of the clients and management, as will be know exactly what to expect.

Deliverables: Functional specification and UI interface documents, project schedules, milestones and human resources
--

### Step 3 – Proof of Concept

Based on a small and relevant module of the application a Proof of Concept (POC) is to be implemented in order to help demonstrate how QuariX can be leveraged as an effective solution to the specific business problems as well as establish viability, technical issues and overall direction.



This gives the client the possibility of having a real feel of the application and streamlines the process needed for the complete project.

Deliverables: Proof of Concept functional application
---

#### **Step 4 – Iterative Application Development**

The goal of this phase is building the target system based on the specifications developed in the previous phases. Through the iterative development process slices of the deliverable business value (system functionality) is delivered into iterations, starting from the model through to the testing/deployment and acceptance.

Deliverables: Complete application deployment including sources and documentation
---

#### **Step 5 – Operation and Maintenance**

Once the product developed, delivered, deployed, and accepted, **Yonder** offers the Application Management Solution which guarantees the proper function of the software, improvements to the product or system and continuous maintenance.

### **Yonder QuariX Solution**

The **QuariX** Rich Client Framework is **Yonder's** solution to a new breed of IT solutions that battles the total cost of ownership of traditional “thick” applications or conventional web applications.

**Yonder** has the right mix of expertise, services and technology to assist our clients in application development, migration services or application maintenance. We understand that every client is unique and we are always prepared to adapt our methodology and technology to enable cost-effective services enabling our clients focus on their most critical asset: their clients and their needs.

**QuariX** is also an important component of **Yonder Transformation Solution**.