Developing a technical architecture for Web-based enterprise software systems

How to bring consistency to infrastructures that support such development

Vispi Munshi

As Web-based enterprise software systems become a reality, the need to understand and plan for infrastructures that support these applications grows. Application developers need to understand the processes and the issues involved in developing this infrastructure so they can architect and design their applications accordingly. This article elaborates on the issues associated with developing a technical architecture for Web-based enterprise software systems.

During the last few years, most enterprise software vendors have announced and implemented plans to Web-enable their software. Two factors have driven this trend:

- First, by making applications available through a browser, companies have been able to drastically reduce the cost of deployment and maintenance.
- More importantly, user interfaces have become more simplified and more uniform, thus reducing much of the user training required during implementation.

As more and more applications are becoming Web-enabled, Information Systems (IS) groups are trying to develop technical architectures to support these Web-enabled applications within their organizations.

What is technical architecture?

Before I go into how the Web-enabling of applications affects technical architecture, I want to explain what I mean by technical architecture.

Technical architecture during the mainframe days was simple: There was one big computer (the mainframe) with dumb terminals attached to it. Then came the client-server era. Now we have a mini computer (the server) and several PCs connected to it through a local area...
network configuration. With advanced networking, distributed computing became affordable and widespread -- in large part because of the need to access enterprise applications from more than one location. Even system software (like databases and operating systems) that supported these applications had to address the issues of networks and distributed computing. Distributed computing raised a hoard of connectivity, performance, and security issues. There was a need to plan and structure the deployment of applications, as well as the need to do some long-term thinking in order to understand and plan for the infrastructure needs of the enterprise.

Today, the enterprise network is the key component of the enterprise technical infrastructure. The application infrastructure includes supporting hardware (servers, storage systems, and client PCs) and associated system software, such as the database and the operating system.

Developing a technical architecture

How can you bring consistency to the infrastructure that supports such applications? You should know who will plan and implement the architecture, learn what issues and roles are involved, and ask the right questions.

Knowing the players and the issues

Someone has to prepare the blueprint for the infrastructure and provide specifications for its components. The architects have to consider both the current and future requirements of both the enterprise and the application being deployed.

**Enterprise-level technical architects** plan the overall infrastructure of the company and lay down standards on how various applications are going to use it.

**Application technical architects** understand the overall network and how the infrastructure supporting their particular application can be most effectively leveraged. They also sort out any connectivity, performance, and security issues with the enterprise architects and their team before the deployment of the application.

Asking the right questions

So how does the application technical architect define the technical architecture for her application? What questions should she ask and to whom? Is there a method to follow? Most of the time, the application team focuses on development and the concerns of application users. The application architect has to understand the application and how it will be used before she can predict its impact on the enterprise technical architecture.

Initial questions would include:

- Who is going to use the application and where are they located?
- What kind of data traffic is the application going to generate?

Most of the answers will come from within the application team. But sometimes the application architect will have to turn to people outside the team. Questions such as "What bandwidth is
available between location X and location Y?” require the application technical architect to interact with people outside the application team. Such questions are best answered by people responsible for the enterprise network -- namely the enterprise technical architects.

**Dos and don'ts for the technical architect**

It is the duty of the application architect to correct any wrong assumptions that the application team might make with regard to the enterprise technical architecture. The application architect also has to ensure that the application and its supporting infrastructure (such as databases and operating systems) follow the norms and standards of the enterprise architecture. Digressions from the established norms are often costly in terms of maintenance, as they might require skills that the current staff may not have. However, sometimes these digressions are necessary in order to support new application features. When this is the case, the application architect has to properly understand the digression and convey and discuss the implications with both the application team and the people responsible for the enterprise architecture.

For each component of the application, the application architect has to provide accurate specifications for the application team and others to use. Sizing a server is a common example of the application architect's tasks. She also has to test connectivity from all locations and resolve the issues before the application rollout. Performance issues are also a concern for the application architect: A proper understanding of the application's data flows can provide her with insights and enable her to predict any bottlenecks. Deciding whether to amend the application or the enterprise architecture is a delicate issue that can often lead to conflicts.

In addition to her technical role, the application architect acts as an arbitrator between the application team and those responsible for enterprise architecture. The application architect defines and conveys the agreed upon deliverables to the enterprise team and also follows up on their progress. This helps to avoid any major impact on the project schedule. Typically, changes to the enterprise infrastructure take a longer time to complete, as they go through a detailed and slow approval process. The architect is expected to create and support the environments necessary for project activities like prototyping, development, and testing. Another important responsibility of the application architect is to provide the project management team with a technical plan that clearly defines the deliverables.

**Developing a blueprint**

Enterprise Resource Planning

The term *Enterprise Resource Planning* was coined in the early 1990s. ERP is a software solution that integrates information and business processes to enable information entered once into the system to be shared throughout an organization. While ERP had its origins in manufacturing and production planning systems, the scope of ERP offerings expanded in the mid-1990s to include other *back office* functions such as finance, human resources, and sales and marketing. The range of functionality of ERP systems has further expanded in recent years to include other applications, such as supply chain management, customer relationship management, and sales force automation. Common examples of ERP systems available include SAP and Peoplesoft.

Figure 1 illustrates the high-level technical architecture of a typical enterprise application.
In Figure 1, the application is supported by two servers at the main location. Both locations B and C also have a server. While users at location A are directly connected to the server at the main location, other users connect through the Internet and firewall to the server at the main location. Most of the time, such users are mobile users. The application architect has to determine which location will have an application server on site. Factors that determine such decisions include: number of users at the location, bandwidth availability between the location in question and the main location, and expected response time of the users. This highly simplified example explains key considerations in determining the technical architecture for a given application. Depending on the application, the application architect has to sketch out high-level, detailed architectures as appropriate.

After discussing the process of developing a technical architecture, I want to look at how the process has evolved and what new issues have arisen because of the Web-enabling of enterprise applications.

**Client-server days**

Early client-server applications were designed mostly for departmental use. Hence the data traffic generated by the application was mostly within the local area network of the user's own department. Very rarely was the application used far from its home location. The application technical architect's role was mostly linked to performance issues of the application. Connectivity
and security were very rarely major concerns of the application architect. Most of the time the database administrator (DBA) played the role of technical architect in the application team, as the database was typically the cause of any application performance issues. The need for an enterprise architecture was small.

The enterprise network developed as companies began feeling the need to connect their sites. However, the applications were still localized. Enterprise applications like Enterprise Resource Planning (see sidebar) created the need for connectivity across the enterprise.

**Impact of the Web**

We all know how the Internet and the Web have had a major impact on life during the last decade. Enterprise applications have also been affected: They have had to change and evolve to meet the needs of a connected world, and continue to do so today. Most companies had already invested huge sums of money in their enterprise network and applications. New applications like e-mail and intranets also challenged the enterprise infrastructure.

So, how has the Web-enabling of the applications affected the role, responsibilities, and methods of the application technical architect?

To begin understanding this, you have to grasp a simple but profound change in the nature of how applications are accessed -- from locations outside the company network.

This means that company data is going to flow outside the company, which raises security concerns. The architect has to determine if application use can be restricted to within a company firewall. If not, then she has to investigate security-related technologies like encryption. Often, because of security issues, the application has to be re-engineered so that only some of its functionality is accessible from outside the firewall. The architect now has to resolve browser issues with the application team and also with the application vendor. She has to ensure that the application can be accessed in a user-friendly way from most common Web browsers without any loss of major functionality.

Connectivity issues are also more likely to be complex and, to be resolved, may require intervention from people outside the application team. If the application is going to be accessed from outside the firewall, then one can expect difficult connectivity issues. The application architect now has to worry about not only the performance of her application, but also how the data traffic of her application is going to affect the performance of other applications. The enterprise network becomes a truly shared network as enterprise applications like ERP are going to utilize it in an online manner. This means it’s important to watch out for any huge downloads or the transfer of data from one location to another.

The application architect has to understand the data flows of the application and, if possible, suggest changes to the application infrastructure. This might mean the addition of a new server at a particular location (which often requires approval from project and IS management, as it frequently affects the budget).
The future

The Web-enabling of enterprise applications has certainly brought the role of the technical architect to the forefront and increased her responsibilities. For rollouts of applications such as ERP, a team of technical architects is commonly formed to better manage these responsibilities. Heading the team is a very senior IS executive, who is respected and heard by the project's functional, technical, and management teams. This executive also has clout with enterprise-level IS teams that are responsible for the upkeep of the enterprise network, standards, and other infrastructure concerns. The application technical architecture team also requires specialists with in-depth knowledge of the hardware and software platforms and protocols being used by the application.

With the advent of e-commerce applications, platforms and technologies are changing at a fast pace. Architects have to plan for the long term by selecting appropriate platforms and technologies with which to deploy and support their applications.

The increasing impact of XML on enterprise software is also noteworthy. After Web-enablement, the adoption of XML seems to be the most likely significant impact on enterprise software technical architectures. XML technology is the common denominator across all leading technology platforms. Enterprise software vendors are trying to integrate their products with other leading applications, and XML is the glue for that integration. Technical architects will have to understand the impact of XML and plan for it effectively.

Technical architecture has been evolving rapidly over the last decade. The major catalysts of change have been the advent of enterprise applications that require enterprise-wide connectivity and the emergence of Web-based applications that require connectivity from beyond the enterprise network. This increased complexity requires forward thinking, planning, and methodical implementation. Technical architecture as a subject has not received the attention that is usually afforded to technologies themselves, and the IS community as a whole needs to give more recognition to this subject.
Related topics

- Check out ITtoolbox which provides excellent resources on a multitude of subjects like ERP, data warehousing, Linux, Java, and CRM. The site also includes e-mail-based discussion groups for most of the subjects it offers.
- Investigate EACommunity, a resource that includes articles, white papers, and more on enterprise architecture. It also provides e-mail alerts when new material is posted.
- Read about enterprise technologies such as ERP, CRM, and SCM on TechnologyEvaluation.com.
- Find more about developer technologies such as Java and XML on Builder.com.
- For more on XML-related technologies, visit the XML zone here on developerWorks.