WHY IS ARCHITECTURE IMPORTANT?

Architecture is the art or science of building, including plan, design and construction. As it relates to physical structures has perhaps been best illustrated by such ancient marvels as the Stonehenge or the Pyramids, however, architecture as it relates to technology is still an extremely young field. With the proliferation of computers and the advent of distributed computing, technology architecture started to gain recognition as a discipline within corporate America in the 1980s. As the computer industry matured — mainframes giving birth to minis and micros, and network interconnectivity starting to play a more significant role, choices in technology were not so simple. It became more challenging for a large company to remain true to one vendor, whether for hardware, software, or telecommunications. As computing infrastructures in medium- to large-size companies became increasingly diverse, interoperability problems were the first to surface. Often, when two vendors’ products were used, they did not work. Sometimes vendors wanted to constrain customers to their products (in certain cases this might have been true), or it was a situation of products not operating by the same standards.

Good architecture becomes the first means by which technologists ensure compatibility between tech-
nology platforms. This leads to the inevitable question: Why is architecture important? To illustrate its importance, consider a large company: what might happen if each business unit wanted to make their own technology choices, independent of each other? Under this scenario, each unit does their own R&D, implements, administers, and supports their own specific solution. What may result is different platforms, technologies, and industry standards. Costs may be significantly more than if the whole organization provided a standard solution. With individual solutions, there may be compatibility and interoperability issues, not to mention eventual replacement and unwind costs.

The kinds of security issues that are being resolved today involve solutions that affect the entire enterprise. For example, take the use of a Public Key Infrastructure (PKI), which is used to provide digital certificates to entities to ensure strong authentication. This type of a solution will only work if all businesses in the organization participate. Other reasons involve risks associated with making a bad choice — one that might expose a vulnerability from inadequate security — going to a technology that is based on proprietary not “open” standards, using a vendor that cannot provide adequate support or just goes out of business. For example, by standardizing on APIs, application portability is achieved; by standardizing on protocols or tokens (X.509 Certificates, Kerberos Tickets, any defined data structure) one achieves implementation interoperability; by standardizing on all of the above, one achieves implementation replaceability of products. In fact, in terms of bang for the buck for information risk management, architecture will always be the best choice, as illustrated in Exhibit 1.

WHAT IS ARCHITECTURE AND HOW IS IT CREATED?
In general, the objective of architecture is to provide a framework that allows products to be assembled quickly and seamlessly permitting the old and the new to interoperate. Client/server, legacy, and distributed systems must bond into a uniform processing machine on a network that enables ubiquitous communications. The enterprise architecture should provide a vehicle for information sharing and promote component reuse. The enterprise architecture can be discussed on four different levels: business, information, application, and technical architectures. These are considered to be horizontal architectures, while security architecture is a vertical architecture contributing to each of these efforts. Security in the business architecture concerns system ownership, business risks, correct deployment of security services in accordance with business risks, and separation of duties. Security in the information architecture concerns information risk management and data risk assessment. Information risk assessments should be conducted on all data, which should be classified according to sensitivity or business confidentiality. Any inherent risk in
the technology, as well as operation risk, should also be assessed. Security in the application architecture includes placement of mechanisms such as identification and authentication, access control, audit, cryptographic, nonrepudiation, and administration. It is at the application layer we talk about APIs. This article focuses on the technical architecture.

An enterprise architecture should be based on a company’s information risk and security policies, but should be tempered by business requirements. The security and risk policies are normally determined by policy committees and can have input from other corporate entities such as audit, compliance, and a corporate communications function. The policies should be technology independent and high level in nature. Based on the policies, more specific technical standards can be written to guide the implementation of the policies. The technical standards should address key technologies and industry standards, but still attempt to be product independent. Both the policies and standards provide input into the architecture. From the architecture, specific products can be determined. Exhibit 2 illustrates the architecture creation process:

**EXHIBIT 1 — Where the Risk Dollar Should Be Spent**

- The cost of security is rising exponentially as technology becomes more sophisticated and ubiquitous
- Allocation of risk dollars must be carefully targeted where they yield the most benefit
- Businesses must decide where they are prepared to accept risk

---

**THE IMPORTANCE OF BUSINESS REQUIREMENTS**

All architectures must be based on business requirements, which can be different for each organization. They can also vary tremendously from unit to unit within the same organization. The first step in designing any architecture is to collect business requirements. This may seem simple,
but it can be the most challenging step. Either by interview or by asking
units to submit requirements immediately raises questions about how the
requirements should be articulated. Simply put, business people talk
about requirements in business terms and security professionals talk
about requirements in terms of security services. The seven security ser-

vices are defined in Exhibit 3.

In general, the two perspectives of business requirements are quite
different. Businesspeople want to know what products meet their busi-
ness needs while satisfying the “policy” people. Their concerns often do
not involve (nor do they care particularly) about the information risk lev-
el of the data or how each security service is provided.

These two perspectives are illustrated in Exhibit 4.

When talking to business people, one might hear requirements such
as the following:

**EXHIBIT 3 — Security Services**

2. Authentication. Verifying the identity of an individual or entity in order to determine
   the right to access specific categories of information or services.
4. Integrity. The prevention of unauthorized modification or destruction of information.
5. Nonrepudiation. The process that prevents someone from denying initiation and/or
   receipt of a message/transaction.
6. Audit. The performance of monitoring and independent review of system records,
   operational procedures and system activities to test for the adequacy of controls, and
   compliance with policy.
7. Availability. Ensuring that information resources and/or computing services are usable
   by authorized persons or programs when needed.
• We need to exchange documents/files with our clients, prospective clients, our business partners and suppliers in a secure way.
• We need to be able to give our clients the ability to place orders and receive confirmations remotely over the Internet.
• We need to enable our portfolio managers to remotely manage their clients’ portfolios.
• We would like to be required to login only once to our company’s systems via the Web and able to access all of our company’s internal databases without having to login a second time.
• We would like to provide for end-to-end security (from client to back-end data) when performing financial transactions over the Internet.

Both business and security professionals agree, however, that there are fundamental technology business drivers. A framework of solid business controls will help minimize financial losses and damage to reputation. Facilitating secure communications between clients and the enterprise will enable new business and permit business to be conducted location independently. An architectural strategy will aid the building of global business, increase market share, and reduce technology cost, while improving availability of information. At the same time, confidentiality and the integrity of client data can be maintained.

The business requirements must be translated into security requirements before the architecture is built. Solutions for classes of require-
ments can be found, which can be categorized on a high, medium, or low information risk level of data. Examples of each of the risk levels of data is given in Exhibit 5.

The security technologist goes through more steps to arrive at the desired product solution. He starts with the business requirements and does an information risk assessment of the business data. This determines whether the data is high-, medium-, or low-risk and which security services are required (e.g., strong authentication, nonrepudiation). The risk assessment then translates into security requirements that define the technical architecture. Out of the technical architecture comes the product set solutions.

THE PURPOSE OF A DEMILITARIZED ZONE (DMZ)

Taken from the military term, a demilitarized zone or DMZ, is used to refer to any kind of screened subnetwork placed between an internal network (i.e., a corporate network) and the Internet. The screening of the subnet is generally achieved by a dual firewall architecture, which can include elements such as bastion hosts and choke routers as well as commercial firewall boxes. The purpose of the firewalls is to provide controlled access to/from the DMZ from both the Internet as well as the corporate or trusted network. A DMZ architecture creates three distinct areas to which access is controlled by the rules set by the firewalls. A typical firewall architecture for a DMZ is illustrated in Exhibit 6.

The left area labeled “Internet” provides no protection to systems located in it. This area is defined as accessible to and by the general public. All systems and data located in the Internet area are assumed to be insecure and potentially compromised. Any information placed on systems in

<table>
<thead>
<tr>
<th>Business Requirement</th>
<th>Risk Level of Data</th>
<th>Access Control</th>
<th>Authentication</th>
<th>Confidentiality</th>
<th>Integrity</th>
<th>Non-Repudiation</th>
<th>Audit</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>View/Query non-sensitive, non-client</td>
<td>L</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>information</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sales/Marketing information</td>
<td>L</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client query of research data</td>
<td>M</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client use of analytical tools</td>
<td>M</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View/Query client information</td>
<td>H</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Update client information</td>
<td>H</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
the Internet area must be low risk, that is, not be sensitive, critical, confidential or proprietary in nature. The Internet area is typically used for public Web services or any type of public application services a company may want to provide to the Internet community. The middle area, labeled “DMZ,” is used to house systems that can provide data and application services to clients of the company via the Internet. Any client application placed in the DMZ affects other systems that reside in the DMZ in terms of overall risk. If one application is compromised in the DMZ, others are exposed to increased risk. These considerations must be taken into account when designing DMZ systems and determining the viability and potential liability of client data that reside there. However, in general, both the Internet and DMZ areas are to be considered part of the Internet; only low risk data may be transmitted through or stored in these areas without additional security controls. The right area, labeled “Internal,” represents the corporate or trusted network. All systems and hosts in this area are considered to be fully protected and secured according to company security policies and standards.

The three areas are controlled by the external and internal firewalls, which protect and restrict access to/from the DMZ. The primary role for the external firewall is to provide controlled access by providing packet filtering rules as well as additional authentication to systems located in the DMZ. This firewall is configured to allow only those systems and services that reside in the DMZ to pass into company network. The internal firewall has a dual function. The first is to protect the Internal network from any unauthorized external access. The second is to protect those hosts and applications residing in the DMZ from any unauthorized access by internal company users. To meet additional future needs, the DMZ can be divided into several segments to provide additional segregation between applications and areas pertaining to different clients.
Some general rules for configuring a firewall are:

- “That which is not explicitly allowed is denied.” All services that pass through a company’s firewall systems must be specifically allowed.
- No network packet is allowed to enter or leave the company network without first being inspected by the firewall.
- Only those services that can be secured will be allowed to traverse the firewall.
- All inbound and outbound traffic from the corporate network must be explicitly allowed and defined by source and destination addresses, protocol, port, and service type. All other undefined traffic is denied.
- All inbound network traffic from an external source first must be initiated by a host from within the internal network before the network packet is allowed to traverse the firewall. All inbound traffic that has not been initiated from within the internal network is denied.

AN EXAMPLE OF A DEMILITARIZED ZONE ARCHITECTURE

So what does an architecture really look like? It will have an implementation model that shows technical components (such as browsers, servers, and certification authorities), how these components talk to one another, and where they physically reside, whether on an internal network, a DMZ, or the Internet. It will contain the characteristics of the model and discuss the flow of a typical message through the architecture. It also should include what level of information risk the data flowing through it can have and how the model addresses the information security services. An example of a security architecture for a DMZ configuration is shown in Exhibit 7.

Exhibit 7 represents a three-tier architecture that can support high risk data. The heart of the architecture is end-to-end security, achieved between Tiers 1 and 3. Additionally, transport layer security is achieved between Tiers 1 and 2 and between Tiers 2 and 3. The path of a request through the architecture is as follows:

1. A secure transport layer is established between Tiers 1 and 2 using Transport Layer Security (SSL). This establishes mutual identification and authentication as well as data confidentiality and integrity.
2. The browser sends a request to the Web server in the DMZ.
3. The request is digitally signed and enciphered by the user.
4. The Web server passes or rejects the request based on the identity established by the transport layer security.
5. If accepted, another secure transport layer is established between Tiers 2 and 3, also using SSL (or some other transport layer security protocol).
6. The request is passed to an application server (Oracle or Sybase), which signature verifies and deciphers the request. Note that there may be other required steps for processing not indicated on this diagram.

7. The application server processes the request and signs and enciphers the response.

8. The response is passed back to the Tier 2 Web server and then to the Web browser, both over a secure transport layer links.

9. The Web browser signature verifies and deciphers the response, thereby achieving end-to-end security.

With this model, mutual authentication, confidentiality, integrity, and access control are achieved from end to end. As this model represents a security architecture view, products must be found or developed that provide these functionalities. The marketplace is quite close to providing commercial off-the-shelf (COTS) products to fit this model, but in some cases, some custom development must be undertaken. In general, implementing a COTS products will ready the architecture more quickly, but building custom components will give an organization some competitive advantage.

HOW TO MAKE AN ARCHITECTURE SUCCESSFUL
Creating a master project plan is always advisable when undertaking such a monumental task as developing and implementing a security ar-
architecture. There are tasks that need to happen serially and those that need to happen in tandem. High-level tasks of such a project plan would most likely include the following:

1. Assemble a team of cross-business representatives.
   • The team must include both technologists and business people, which gives a more complete perspective.
   • This group should make all the major decisions, thereby inherently giving buy-in to any conclusions.
   • Funding should come in part from each business unit and in part from the central infrastructure group.
2. Determine the scope of the project.
   • Decide if the architecture will be for the Internet, an intranet, or an extranet, or all of the above.
   • Decide if the architecture is to last one or five years.
   • Decide if the architecture is to be based on any given infrastructural choices on which the organization has already made a sizable investment.
3. Gather the cross-business requirements.
   • Gather the requirements in business terms, helping the team to focus on like categories of requirements.
   • Translate the business requirements into security terms, because the products must meet the security requirements.
4. Design a security architecture that meets the security requirements.
   • Base solutions on open standards: Internet Engineering Task Force (IETF), Object Management Group (OMG), the OpenGroup, and other standards that are on a standards track to ensure that the architecture is interoperable and compatible with other technologies and products.
   • Base solutions on official corporate technology policies and standards.
   • Choose one or two businesses with a current and pressing need for the architecture and work with them to develop a solution that will fit their needs.
4. Develop an implementation plan for the architecture.
   • Develop two or three business-specific implementation architectures that map to the general architecture.
   • Identify all common components and services to be reused. These components and services may already be in existence or may have to be built.
   • Evaluate and select the actual products to implement the architecture and custom build any components that are not COT products.
   • Discuss the implications of deploying a technology that is not one of the core infrastructural choices. Discuss any unwind costs or technical transition/migration issues.
• Develop specific implementation road maps (timelines to implement specific technologies) for any infrastructure service or reusable component.

• Develop critical path diagrams (Gantt Charts) to show which tasks must occur in tandem, and which serially. Include all start dates and end dates to determine windows of availability for products and components.

The final document should also include a discussion of the legalities of privacy as well as the export of encryption (international vs. domestic). Products that use encryption are subject to specific export controls by the U.S. Department of Commerce (DOC). The controls center around the length of the encryption key and who is the designated user of the encryption (such as name and country). These names must be registered/verified with the DOC twice per year. Export controls also vary depending on requesting institution. In the past, financial institutions have been given more flexibility. In general, these laws may change quickly because financial services organizations are being redefined.

There also may be issues with data privacy laws covering the monitoring of individuals in Germany, Scandinavia, Singapore and some other countries as well. Other issues may surface with the forthcoming European Privacy Directive. For example, France does not permit the use of any encryption unless it is with permission. There may be restrictions from the People’s Republic of China and Russia as well as some type of “process” requirement from Singapore. These restrictions/laws need to be investigated, and if possible, attorneys should determine the most current laws and regulations.

“SELLING” THE ARCHITECTURE
Measurable benefits must be identified to cost justify the architecture. Show dollar values where at all possible. Generally, the benefits fall into three categories:

1. Business enablements. Typical business enablements that may result from a security architecture might be:
   • secure financial transactions over the Internet,
   • single sign-on to multiple Web servers,
   • secure communications with clients via E-mail or FTP,
   • strong authentication of clients from any location or any machine, and
   • secure remote client portfolio management by salesforce.

2. Cost savings. Typical cost savings might be:
   • electronic delivery of services (sales, fulfillment, and support),
   • reuse of infrastructure services, and
• reuse of software components.

3. Cost avoidance: This category is the hardest to quantify, but include:
• Elimination of redundant architectural efforts
  • R&D and tools evaluation
  • analysis of security risks and countermeasures performed by information risk and security professionals
• Risk reduction resulting from single architectural effort
  • information risk
  • audit findings
• Centralization of resources (architectural implementation, required infrastructure, security administration, security monitoring, standards research and analysis)
• Unwind costs of bad technology choices: nonstandard, noninteroperable, etc.

The final architecture document should include a matrix that maps the solutions back to the original business requirements to close the loop. This is critical: it shows how the architecture meets the requirements on a one-by-one basis. Now, to really sell the architecture, put on a roadshow with a fancy presentation illustrating all the key benefits; getting buy-in from key senior players. Nothing sells like how to save money!!

Carol A. Siegel is a vice president of J P Morgan’s Internet & Security Architecture group. Carol is responsible for setting Morgan’s strategic direction with respect to Internet security, and most recently has led a major Internet security architectural development effort at Morgan. Carol has an undergraduate degree in Systems Analysis Engineering and an MBA in Computer Applications.

Note
1. Webster’s.