zapthink white paper

PROCESSING XML ON THE NETWORK

THE IMPORTANCE OF THE DATA LAYER

ZapThink, LLC • 11 Willow Street, Suite 200 • Waltham, MA 02453 • info@zapthink.com • www.zapthink.com

PROCESSING XML ON THE NETWORK

THE IMPORTANCE OF THE DATA LAYER

December 2002

Analysts: Ron Schmelzer and Jason Bloomberg

Abstract

The dramatic increase of Web Services and other XML traffic on today's enterprise networks presents serious security, management, and performance considerations for the IT manager. At the root of the problem: existing network devices such as routers, firewalls, and load balancers operate at the packet level, rather than at the data or content level. As a result, existing network infrastructures are entirely unable to provide the data-level security that enterprises need. To meet this need, vendors are offering a variety of solutions with a confusing array of features. These products fall into two broad, overlapping categories: network appliances and XML gateways.

Table of Contents

I. The Importance of the Data Layer

XML and Web Services traffic on the network will dramatically increase over the next few years. ZapThink estimates that XML currently represents only 2% of all network traffic in 2002. However, this figure is expected to increase to almost 25% of all LAN network traffic by 2006. In most cases, this increased XML traffic raises serious security questions. After all, if XML goes right through the Web ports on the firewall, even SSL—the industry standard way of securing Web traffic—is inadequate for preventing unauthorized XML traffic on the network.

As a result, many security vendors have jumped into the market for data-level security products. Some of the newer products are hardware devices, while others are software. There seem to be as many kinds of solutions as there are vendors—each one seems to have a different product strategy. ZapThink has cut through this noise and made sense out of the data-level processing market. Before we can explain the market, however, we must explain what data-level security and processing is.

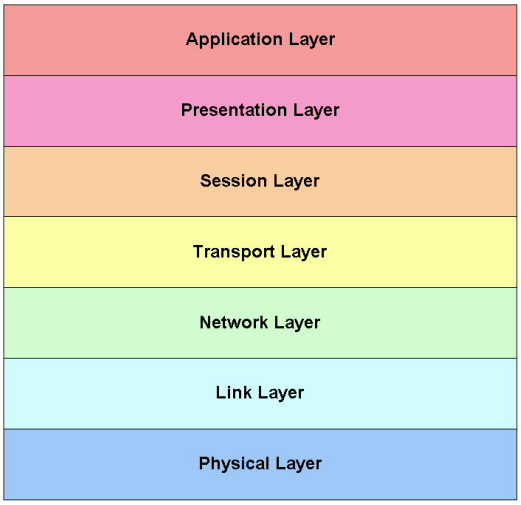
1.1. Network Level Processing vs. Application-Level Processing

To understand what the data level is, it's important to understand the basics of network processing. Network processing is often referred to as part of a seven-layered network model, called the Open Systems Interconnect (OSI) stack. The OSI stack, as shown in Figure 1 below, encapsulates the following layers of networking technology:

- *Physical* – Represents the lowest level of network connectivity – that of the wire or physical connection that links machines together
- *Link* – Provides the necessary protocol framing to get two systems to successfully communicate.
- *Network* – Identifies a machine as part of an overall network architecture and provides mechanisms for connecting multiple systems together.
- *Transport* – Provides a means for transporting messages over the network and connecting multiple networks together into an overall network.

- *Session* – Specifies protocols and messaging schemes that will operate together for the fulfillment of a task in a session.
- *Presentation* – Identifies the messaging and protocol specifications necessary to display and represent information to the user
- *Application* – The highest level of the networking stack, the application level protocol specifies a "complete" networking application.

Figure 1: OSI Network Stack

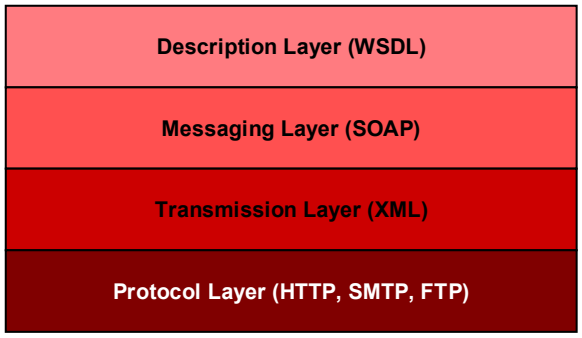


Network devices of all sorts operate at various levels of the OSI model. Many vendors target the middle of the OSI stack with a plethora of familiar devices including gateways, routers, switches, hubs, bridges, firewalls, and proxies. In effect, all of these kinds of equipment are network Intermediaries in that they facilitate communications between systems, but aren't the end systems themselves. In the end, networking devices that follow the OSI model are oblivious to the specific application architecture running on the network, but instead concern themselves with the mechanics of communication.

From the perspective of sending data over the network, far too much functionality is included in the application layer.

The problem with the aging OSI model is the application layer. From the perspective of networking, the single application layer is sufficient to describe communications with applications. However, from the perspective of sending data over the network, far too much functionality is included in the application layer. As a result, the application layer actually consists of a number of individual data layers, some of which are shown in Figure 2:

Figure 2: Data Layers within the Application Layer



The fact that XML and the Web Services standards must operate at the application level of the OSI stack poses a challenge, because all data-specific network traffic appears to be the same to lower-level devices. As such, companies need a new class of network devices to become aware not only of the network ports and IP addresses, but also of the content itself that is traveling across the network. In this regard, current firewall, router, proxy, and switch solutions are inadequate. Instead of being simply network and IP-aware, these solutions must be *data-aware*. More specifically, they need to be XML-aware. They need to be able to inspect and understand XML traffic as it flows across the network and perform some sort of activity on the traffic, as policies dictate.

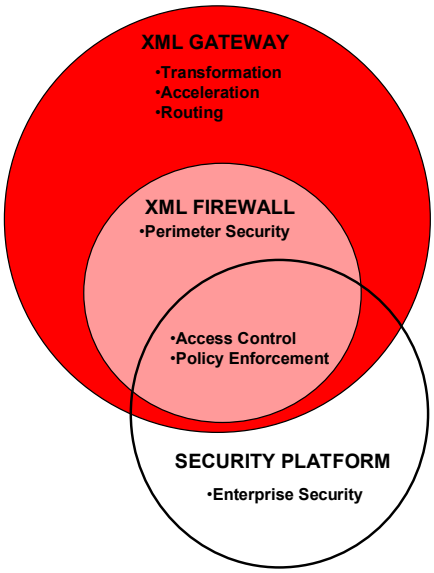
Data-level processing involves looking inside the content of the network traffic, and making authentication, authorization, and other decisions for that content.

Data-level processing, therefore, goes beyond the packet-level processing offered by traditional firewalls, routers, and load balancing solutions that operate at the middle of the OSI stack. For example, data-level security involves looking inside the content of the network traffic, and making authentication and authorization decisions for that content. Data-level security also includes support for confidentiality—using encryption to keep messages secret. Other data-level processing operations include critical management, performance improvement, and transformation operations. Any company that wishes to exchange XML messages or Web Services calls across their network perimeter must understand how to implement data-level processing.

II. Three Approaches to Data-Level Processing

At the core of the data-level processing market is a class of solution known as an *XML firewall*. XML firewalls basically provide perimeter security to an enterprise that is content-aware: they can inspect XML traffic and make routing and blocking decisions based on the content of the messages. However, while there are some vendors with XML firewall products on the market, many vendors position their XML firewall products as parts of other solutions, causing confusion in the marketplace, as shown in Figure 3:

Figure 3: Data-Level Processing Market Segments



Some vendors provide data-level processing by offering XML gateways, while others include XML firewall functionality as part of a Web Services security platform. To make matters worse, each of these categories can overlap, as well. This report explains what each of these categories means, and why an enterprise would choose to purchase one over another.

2.1. XML Firewalls

A *firewall* is a device that acts as a message intermediary, inspecting the traffic that attempts to pass, and either allows or rejects the traffic based on a range of criteria. Traditional hardware firewalls inspect traffic on the packet level, and as such, cannot reject traffic based on the structured content (for example, XML-formatted content) it contains. Software XML firewalls are capable of understanding the content that pass through them and appropriately managing that traffic.

These firewalls typically operated by inspecting SOAP message headers. Even for the part of the SOAP message not intended for the firewall itself, the software firewall may be able to decrypt part or the entire message and policy decisions based upon the secure contents of the message.

XML firewalls typically offer several of the following features:

- XML message inspection (including SOAP messages)
- Malicious attack protection and intrusion detection
- Access control, including authentication and authorization capabilities on an accept or reject basis
- Decryption and encryption capabilities
- Transaction auditing capability

Some XML firewalls consist of software, while other vendors implement XML firewall functionality in hardware. In either case, XML firewalls are examples of a broader category of product called an XML gateway.

Some XML firewalls consist of software, while other vendors implement XML firewall functionality in hardware. In either case, XML firewalls are examples of a

broader category of product called an XML gateway, or XML proxy. XML gateways are discussed in the next section, and hardware solutions are covered in section 3.1.

2.2. XML Gateways

"XML gateway" describes applications that monitor network traffic for XML content and perform some activity on that traffic as dictated by pre-defined business rules.

An appropriate term to describe the evolving role of XML-aware intermediaries is *XML gateway*. "XML gateway" describes applications that monitor network traffic for XML content and perform some activity on that traffic as dictated by pre-defined business rules. XML gateways can be either hardware or software solutions that actively listen for XML traffic on a corporate network and either pass the traffic along unmodified or perform some action on its content.

XML gateways add one or more of the following pieces of functionality:

- *Message Routing* –Transport and relaying of XML and Web Services messages from one network segment to another based on the context and content of the message.
- *Security* – Enforcement of corporate security policies on all processed XML traffic including data and schema validation, encryption, authentication, authorization, identity management, privacy, and rights management operations. Access control on a gateway often includes more of a decision-making process than the "accept or reject" capabilities of an XML firewall.
- *Performance Enhancement and Acceleration* – Operations to optimize XML traffic including compression, caching, and other packaging steps.
- *Message Transformation* – In-stream transformation of one XML message type to other XML message types or other formats such as EDI or other text formats.
- *Monitoring and Management* – Activities and operations centered on obtaining an accurate picture of the flow of XML and Web Services in a network and work to improve the overall performance of XML to meet necessary business needs. Includes notification, alerting, and fault management for Web Services, including the management of SOAP faults.

XML gateways can help to establish security policies in a comprehensive manner that is consistent with overall corporate network. XML gateways can monitor XML traffic for specific patterns and apply content-level security features such as encryption down to the element level, digital signatures, and authorization features. XML gateways can work with other security solutions by applying different security technologies based on the destination or nature of the XML documents. For example, Web Services messages bound for external networks may need to have all of their data encrypted, while those messages bound for internal destinations can simply have the appropriate digital signatures.

XML gateways can identify security risks across multiple packets and point-to-point message exchanges, making them far superior to packet-level security detection methods.

Current TCP/IP-based firewalls are wholly inadequate at providing data-level security. XML exchanges can span multiple partners, require interaction among many entities, and integrate with corporate security policies for access control and authorization, thus posing additional security challenges. XML gateways can identify security risks across multiple packets and point-to-point message exchanges, making them far superior to packet-level security detection methods.

Also, because XML gateways are implemented as network-filtering devices, data centers can implement the latest XML security technologies in a transparent

manner, thereby eliminating the requirement for developers to apply new technologies or patch existing implementations in order to be compliant with the latest standards. XML gateways in this way provide a "shell of security" over existing applications, sheltering developers from the need to keep track of or even implement XML security or privacy technologies. See ZapThink's *XML Proxies: XML-aware Network Appliances and Firewalls* report for more information.

2.2.1. Achieving a Positive ROI with an XML Gateway

In networked, distributed computing systems, there are fundamentally two types of entities: those that are providing functionality (servers or service providers), and those that are requesting functionality (clients or service requesters). In a typical non-proxy system, service requesters send messages directly to providers, which respond with the results of the requested operation. While this client/server approach works in limited volume environments, the direct request-response model does not scale well. For example, if all email systems worked by directly sending email between senders and receivers, the service would fail under high email loads. Email servers function as aggregation points for inbound and outbound messages, optimizing the network flow and handling service outages.

At a certain threshold, centralizing messaging functions in an XML-aware intermediary such as an XML gateway greatly improves performance and scalability.

In much the same manner, XML messaging will experience tremendous scalability issues if all communication happens on a point-to-point basis. At a certain threshold, centralizing messaging functions in an XML-aware intermediary such as the XML gateway greatly improves performance and scalability. Highly scalable distributed systems that are predominantly message-based require flexible buffering of messages and routing, based not only on message parameters such as origin, destination, and priority but also on the state of the system measured by parameters such as the availability and load of its nodes as well as network traffic information. In this model, it is the role of transparent intermediaries to handle the messaging and operation workload. Based on various message characteristics, the XML gateway can send the message directly to the end point, respond with a pre-cached result, or batch messages to a server for batch response.

XML proxies and network appliances can also allow users to implement XML and Web Services solutions without having to frequently modify those applications to comply with various corporate XML policies. These policies may affect the security, management, performance, and vocabulary features of the XML documents. For example, an enterprise may stipulate that all XML messages bound for outside the network must be compressed, digitally signed, and compliant with ebXML specifications. Rather than recoding all XML and Web Services applications to be compliant with this policy, the XML gateway can apply security, performance enhancement, and transformation rules to all outbound-only traffic, while leaving all behind-the-firewall traffic alone in its original format. In this way, XML gateways can not only help applications comply with corporate XML policies, but can also bring different XML and Web Services implementations into conformance with a single XML methodology.

Another major driver to implementation of XML gateways is the need to simplify the process of integrating with external partners. Since XML gateways can normalize a company's XML "footprint" to the outside world, they can be used very effectively to help reduce the amount of time that it takes to communicate with new business partners. In effect, XML gateways act as B2B integration applications, but without the complex business process management and semantic integration capabilities. Naturally, more sophisticated integration-specific applications such as those offered by Service-Oriented Integration (SOI)

techniques must handle these capabilities (see ZapThink's *Service-Oriented Integration* Report for more information). However, XML gateways can help to simplify the process of exposing SOI solutions to actual end points. Since XML gateways can accept XML traffic from multiple SOI solutions and XML-producing applications, they can serve as aggregation points for XML traffic and thus simplify external XML integration.

As XML becomes an increasingly important part of the corporate IT fabric, IT administrators, managers, and CIOs will want to wring more value out of their XML traffic.

Finally, users wish to gain increasingly more value out of the XML documents and traffic on the network. As XML becomes an increasingly important part of the corporate IT fabric, IT administrators, managers, and CIOs will want to wring more value out of their XML traffic. For example, they may want more in-depth reporting and auditing of XML traffic, message-tracing facilities, billing and metering functions, and all sorts of value-added XML features that are currently in the imaginations of vendors and end users. XML gateways can provide the "tap point" into the XML traffic stream where they can add these functions. Without such an XML-aware intermediary, adding these new features becomes an expensive and custom-coded endeavor.

2.3. Web Services Security Platforms

The concept of a *platform* is a set of technologies that underlie a variety of applications and services and provide several basic services to those applications and services. It is unusual, therefore, to find a special use platform, because by their very nature, platforms serve multiple purposes. As a result, many existing IT platforms provide certain security features, and many of those also support Web Services.

Web Services security platforms offer enterprise-wide security capabilities, and typically share many of the following features:

- A security policy engine that coordinates security and privacy policies across multiple systems.
- XML network management capabilities, including monitoring and reporting features. Platforms also often integrate with existing network management tools (typically via Web Services interfaces).
- A trust server that might manage keys and certificates, and serve as a registration and assertion server.
- Adapters and connectors for integration with various other systems.
- Many also have support for SAML tokens.

Web Services security platforms also typically contain XML firewalls as part of their overall solution. See ZapThink's *XML and Web Services Security* report for more information on Web Services security platforms.

III. The Hardware vs. Software Decision

There are two basic approaches to implementing an XML gateway solution: either as a software application or as a hardware device. When implemented as software, an XML gateway must have a runtime environment as well as a box to run on. Some XML gateways run as standalone applications on top of their own integrated runtime engines, but most take advantage of application servers or Web Servers to provide their runtime environments. In any case, software architects and developers are typically involved when selecting, installing, and configuring a software XML gateway.

Hardware devices take a different implementation approach. Such network appliances are self-contained, and fit into racks in the data center. As a result, the data center manager typically manages them, and the software developer does not necessarily need to know that the device is present. In other configurations, the network appliance provides a Service on the network that the developer can access via standard Web Service protocols. This section explores the advantages of the network appliance approach in more detail.

3.1. Hardware XML Network Appliances

There are two primary ways that enterprises can implement XML gateway solutions: in dedicated hardware (XML network appliances) or in specialized software (software XML gateways). These approaches are not mutually exclusive; there are distinct scenarios in which hardware implementations are most appropriate, other scenarios where software is best, and yet others where combinations of hardware and software solutions are optimal.

The trend in network devices is to move from custom-built software applications to dedicated hardware devices that are built as a "black box" in which specific optimizations have been made for a particular network protocol layer or application and can fit within traditional data-center rack environments.

The trend in network devices is to move from custom-built software applications to dedicated hardware devices that are built as a "black box" in which specific optimizations have been made for a particular network protocol layer or application and can fit within traditional data-center, IT rack environments. There are a number of scenarios where hardware XML network appliances are the most appropriate implementation of XML gateway solutions:

- *Performance* – Network appliance solutions can make use of specialized hardware and optimize their software solutions to make use of this hardware to drastically improve performance over pure software XML gateways. In particular, there exist a variety of hardware solutions for accelerating encryption, parsing, and transformation operations.
- *Deployment flexibility* – Network appliances can typically be installed in a transparent, in-line mode between the standard firewall and the servers on the internal network, in shared service mode, acting as an addressable coprocessor to the network, or in proxy mode, actively intercepting and processing XML traffic as it passes into the network.
- *Controlled installation environment* – IT personnel can preconfigure network appliance solutions so that they are ready to install by simply plugging the equipment into the appropriate rack. As a result, IT shops can control their installation process, making sure that they have properly configured certain security, routing, transformation, and management features prior to installation.
- *Centralized installation in IT environment* – IT personnel can install network appliances in a central network operations environment, allowing developers to pass off device administration and maintenance responsibilities to IT administrators. In addition, administrators can manage network appliances as they manage other devices on the network rack, with technologies such as SNMP, and provide backup, power, cooling, and appropriate equipment facilities.
- *Different purchasing cycle* – IT managers purchase and account for network appliances differently than software. They can be depreciated over many years, and can fit into global IT purchasing budgets in addition to divisional or departmental IT budgets. This allows greater flexibility in selling network appliance solutions by offering a choice of customer: either the application developer or the central IT administrator.

- *Simple channel sale* – Reseller channels that currently sell network equipment and associated products can easily sell network appliances as well. Online and retail stores can sell these devices, as well as value-added channels such as VARs and systems integrators.

3.2. Achieving Internal Goals with a Network Appliance

Every company is ostensibly looking for a positive ROI on any IT investment, but there are many direct and indirect factors that make up the return on such an investment. The deployment flexibility of network appliances in particular can potentially meet the various needs of different people within the IT organization:

- The CIO or executive in charge of the data center is looking for straightforward *ROI*. If a network can solve problems that are costing the organization money, and can solve them fast enough to justify the cost of the appliance, then the executive will recommend the purchase of the appliance. For such executives, the pain points can include insufficient data-level security, as well as levels of XML traffic that are high enough to form a bottleneck within the existing architecture.
- The network administrator above all else is looking for *simplicity*. The admin will be concerned about what changes must be made in the existing data center environment, and what resources will be required to manage the new device. This person will typically favor network appliances that can be placed in shared service mode, because of the minimal impact on the network topology.
- The application architect wants *transparency*. The architect realizes that to use a network device in proxy mode, many configuration changes may need to be made. Likewise, a network appliance in a shared service configuration requires application code to take advantage of its capabilities. As a result, the architect typically favors a network appliance in transparent in-line mode.
- The application developer, however, desires *application functionality*. They have design time problems to solve, and a network appliance in shared service mode that the developer can offload XML processing tasks to can solve those problems.

Naturally, which configuration a network appliance ends up in depends upon the needs of the particular organization. The fact that network appliances often offer most or all of the functionality of a software XML firewall, and then also adds performance and deployment flexibility improvements as well, can make a network appliance the favored choice of many IT organizations.

The fact that network appliances often offer most or all of the functionality of a software XML firewall, and then also adds performance and deployment flexibility improvements as well, can make a network appliance the favored choice of many IT organizations.

IV. Conclusions: Which Approach is Right for You?

There are a number of business drivers that are spurring the adoption of XML gateway, firewall, and network appliance solutions. These drivers include:

- The need to secure XML and SOAP traffic crossing the enterprise perimeter.
- The need to manage increased volume of XML network traffic.
- The desire to establish a consistent XML usage policy across the enterprise.

- The desire to simplify external (B2B) integration.
- The desire to continually value-add XML and Web Services implementations on the network.

Naturally, what type of solution is appropriate depends upon the particular circumstances in the enterprise. Here are some pointers to keep in mind when deciding which approach is right for you.

4.1. How to Select an XML Gateway

The box on the next page contains some straightforward decision points that can help you decide which XML gateway approach is appropriate in your situation.

If your primary concern is malicious or unauthorized Web Services or other XML traffic entering your network from external Sources…
- You should consider an XML/Web Services firewall.

If you must establish enterprise-wide IT security policies…
- You should consider a Web Services security platform.

If you are trying to eliminate an XML processing a bottleneck somewhere in your network…
- You should consider an XML gateway that offers performance improvements in parsing, translation, routing, and other XML processing operations.

If you want to implement data-level processing in a simple and transparent manner without user intervention…
- You should consider a network appliance that can run in shared service or transparent in-line mode.

If you want to provide corporate application developers XML and Web Services processing capabilities such as XML transformation or encryption/decryption as a Service on the network…
- You should consider an XML gateway that can run in shared service mode.

If you want to utilize an XML gateway in your data center…
- You should consider a network appliance implementation.

4.2. Summary

The rapid increase of XML and Web Services traffic on the network heralds enormous benefits to the organizations that make use out of these standard technologies, but these technologies come with their own risks and drawbacks. The demands on optimizing the performance of the XML data and applying enterprise-wide XML policies are increasing daily. More and more organizations are seeking to find solutions that can transparently monitor XML traffic on the network and apply business rules or corporate IT policies such as security, routing, performance, management, transformation, or end-point connection provisioning. Furthermore, XML is creating new exposures--security, performance, and otherwise--that corporate IT environments must appropriately deal with.

In order to process high volumes of XML content on the network, both hardware and software devices must able to understand not only network protocols, but also the XML-based content traveling on these protocols. However, current network technologies are not capable of meeting this demand, thus requiring new categories of XML-aware applications. As a result, vendors like **Forum Systems** have brought full-featured network appliances and XML gateways to market that provide the functionality and flexibility that today's XML-intensive IT infrastructures require.

Copyright, Trademark Notice, and Statement of Opinion

About ZapThink, LLC

ZapThink is an IT market intelligence firm that provides trusted advice and critical insight into XML, Web Services, and Service Orientation. We provide our target audience of IT vendors, service providers and end-users a clear roadmap for standards-based, loosely coupled distributed computing – a vision of IT meeting the needs of the agile business.

ZapThink's role is to help companies understand these IT products and services in the context of SOAs and the vision of Service Orientation. ZapThink provides market intelligence to IT vendors who offer XML and Web Services-based products to help them understand their competitive landscape and how to communicate their value proposition to their customers within the context of Service Orientation, and lay out their product roadmaps for the coming wave of Service Orientation. ZapThink also provides implementation intelligence to IT users who are seeking guidance and clarity into how to assemble the available products and services into a coherent roadmap to Service Orientation. Finally, ZapThink provides demand intelligence to IT vendors and service providers who must understand the needs of IT users as they follow the roadmap to Service Orientation.

ZapThink's senior analysts are widely regarded as the "go to analysts" for XML, Web Services, and SOAs by vendors, end-users, and the press. They are in great demand as speakers, and have presented at conferences and industry events around the world. They are among the most quoted industry analysts in the IT industry.

ZapThink was founded in October 2000 and is headquartered in Waltham, Massachusetts. Its customers include Global 1000 firms, public sector organizations around the world, and many emerging businesses. ZapThink Analysts have years of experience in IT as well as research and analysis. Its analysts have previously been with such firms as IDC and ChannelWave, and have sat on the working group committees for standards bodies such as RosettaNet, UDDI, CPExchange, ebXML, EIDX, and CompTIA.

Call, email, or visit the ZapThink Web site to learn more about how ZapThink can help you to better understand how XML and Web Services impact your business or organization.

ZAPTHINK CONTACT:
ZapThink, LLC
11 Willow Street
Suite 200
Waltham, MA 02453
Phone: +1 (781) 207 0203
Fax: +1 (786) 524 3186
info@zapthink.com

ZapThink, LLC • 681 Main Street, Suite 2-11 • Waltham, MA 02451 • info@zapthink.com • www.zapthink.com