



Turbo-Charging Dynamic Web Sites with Akamai EdgeSuiteSM

Table of Contents

Introduction3

Next-Generation E-Business Applications 4

Dynamic Sites Delivered from The Edge5

Business Benefits11

A Technical Primer 12

About Akamai 16



Introduction

Content Delivery Networks (CDN) provide users with fast and reliable delivery of Web content, streaming media, and transaction processing across the Internet. As the leading provider of secure, outsourced e-business services and software, Akamai Technologies maintains the largest CDN with edge servers in more than 13,000 locations worldwide. Akamai provides solutions that optimize Web site performance, deliver broadcast-caliber streaming media, and provide interactive application services.

As content delivery pathways between the server and the user continue to become more congested, problems such as sites that load slowly, are only partially delivered, or crash during delivery are increasing. A content delivery network solves these problems while reducing infrastructure costs and enhancing the company's bottom line. Akamai's research has shown that Web sites using a CDN can increase click-throughs by 20%, reduce abandonment rates by 10-15%, and increase order completion by 15%.

CDNs resolve performance problems related to Web server processing delays and Internet delays. Users requesting popular Web content may well have those requests served from a location much closer to them (a local Network Provider's data center), rather than from much farther away at the original Web server. By serving content requests from a server much closer to the user, a quality CDN can reduce the likelihood of overloaded Web servers and Internet delays. A CDN can deliver rich, compelling content while increasing customer loyalty and strengthening your company's brand.

Akamai Technologies offers a distributed architecture for e-business applications that leverages network intelligence and economies of scale to deliver the type of reliability, scalability, performance, and availability required for mission-critical e-business systems. As of October, 2001, Akamai's network consists of approximately 13,000 servers in 1,000 leading carriers in more than 63 countries. These include hosting, backbones, broadband, satellite, and subscriber/access networks. This highly proven network platform is already the workhorse of the mission-critical infrastructures of 1,200 of the world's leading e-businesses and supports average traffic levels of multiple gigabits per second.

Until recently, content delivery networks were typically focused on delivering static content, but Akamai can also accelerate dynamic and personalized content. EdgeSuite represents Akamai's next-generation content delivery service, leveraging its core technology and industry-leading content delivery network for the distribution of an entire Web site—including static, dynamic, embedded objects, and HTML. EdgeSuite was specifically designed to seamlessly interact with the content management systems and application servers that generate highly dynamic content. This document describes how content providers can integrate EdgeSuite into their existing infrastructure to serve personalized and dynamic content from the distributed server closer to their end users.

Next-Generation E-Business Applications

Several years ago, the Web was seen by many companies mainly as a new way to publish corporate information. As these companies' Web sites grew, the problem of managing this increasingly dynamic content on their sites grew exponentially, and the first content management applications emerged. Application servers were also developed to handle all application operations between Web servers and a company's back-end business applications, legacy systems, and databases. Because these applications cannot process HTTP requests and generate HTML, the application server works as a translator—allowing, for example, a customer with a browser to search an online retailer's database for pricing information.

Application servers and content management systems occupy a large chunk of computing territory between database servers and the end user (often referred to as middleware). There are many reasons for having an intermediate player in this connection—among other things, a desire to decrease the size and complexity of client programs, the need to cache and control the data flow for better performance, and a requirement to provide security for both data and user traffic.

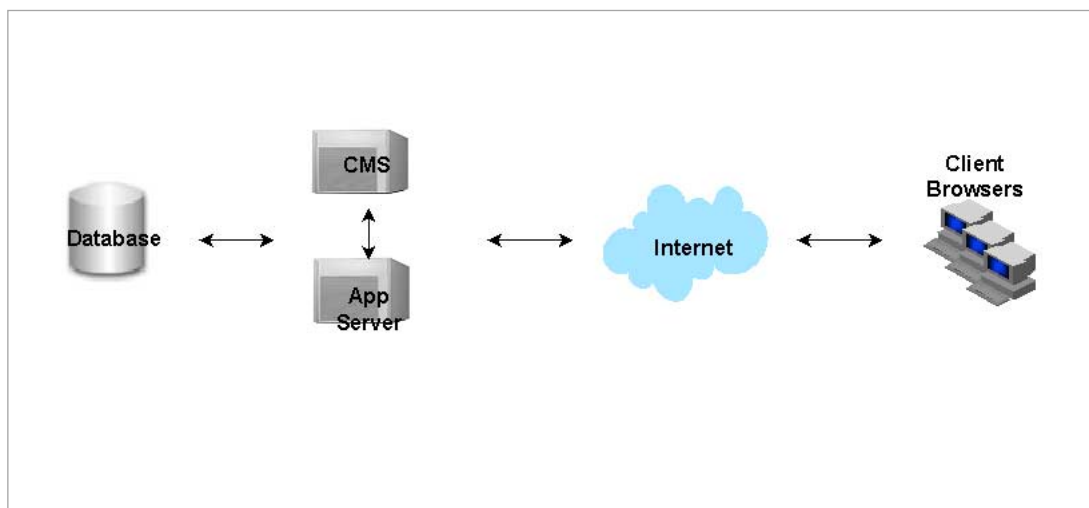


Figure 1: Traditional Data Center Infrastructure

Today's and tomorrow's e-businesses face the challenge of providing more and more access to back-end systems in order to streamline their operations by providing Web-based applications to their customers, suppliers, and partners. The business processes that must come together to drive this new generation of online applications are more complex than ever before. Far from the HTML and static pages of years past, the new breed of applications depends on hundreds, if not thousands, of discrete content fragments. The content involved now feeds dynamic, personalized applications.

Within a given data center, the primary Web site components include a router, reverse proxy caches, switches, local load balancers, Domain Name Service (DNS) server, Web servers, application servers, database servers, and storage, firewalls, and access lines. The typical architecture of a hosted e-business infrastructure is best depicted in three tiers:

- **Content Generation Tier:** The content generation tier is typically centrally maintained in an enterprise data center or a hosting facility. Its primary function is for application coordination and communication to generate the information that is to be presented to the end user based on a set of business rules. It typically includes application servers, directory and policy servers, data servers, transaction servers, and storage management.

- **Integration Tier:** Between the application tier and the content delivery tier is a simple integration layer that provides HTTP-based connectivity between the e-business applications of the content generation tier and the content delivery tier. In the distributed architecture, it consists of a single or few Web servers serving as HTTP communication gateways.
- **Content Delivery Tier:** A massively distributed, fully outsourced edge network is the foundation of a successful content delivery tier. The ubiquity of this network allows content to be stored and delivered from the edge of the Internet within a few hops of the user, resulting in as much as an order of magnitude improvement in download performance. The scale of the edge network also minimizes the need for a large, horizontal origin site because it acts like a giant reverse proxy cache, fielding requests and serving content in lieu of the origin server.

With the introduction of the Akamai EdgeSuite service, companies now can improve the performance and reliability of their site by distributing content to the edge of the Internet, closer to the end user. Users of content management systems and application servers immediately think about their site infrastructure though and ask: What about dynamically generated content? Can that be cached on the edge? And if so, how?

Dynamic Sites Delivered from The Edge

Many sites, especially transaction-intensive e-commerce and highly personalized sites, utilize application servers and back-end database and content management systems to create a dynamic site. EdgeSuite allows you to separate this content generation, which takes place in your data center, from content assembly and delivery, which can take place at the edges of the Internet. For the purpose of this document, we will illustrate how EdgeSuite works in conjunction with such a site, by using the example of an online retailer for electronic products.

For example, when a user accesses the site and searches for “handhelds,” that request is sent to the application server. The application server performs a database query and assembles the page based on the return values and other common page components, such as navigation menu, logos, and advertisements. The user then receives the assembled page containing product images, product descriptions, and advertising (see Figure 2).



Figure 2: Workflow without EdgeSuite

The next time the page is accessed, the same steps need to happen, which introduces unnecessary latency in content assembly and content delivery. Sometimes the page might be cached within the application server's internal cache, in which case the request would still have to be satisfied from the origin server, requiring a full round-trip from browser to origin server and back and requiring additional computational processes on the application server, necessitating more CPU and memory usage.

On the other hand, if the site in our example is served using EdgeSuite, all initial hits go to the Akamai edge servers. The edge server then checks its internal cache to see if the page has been requested before and is still stored in cache. The rules for cacheability of content are defined by the content provider via a metadata configuration file.

In our example the company assigned the page a Time To Live (TTL) of one day, since there are only infrequent changes to the inventory for handhelds. The first time a user requests the page it will be assembled by the application server as described in Figure 2. Since the page has a TTL of one day, it will be stored on the Akamai edge servers for that time period, so that all subsequent requests for that page can be served from a server closer to the user (Figure 3).

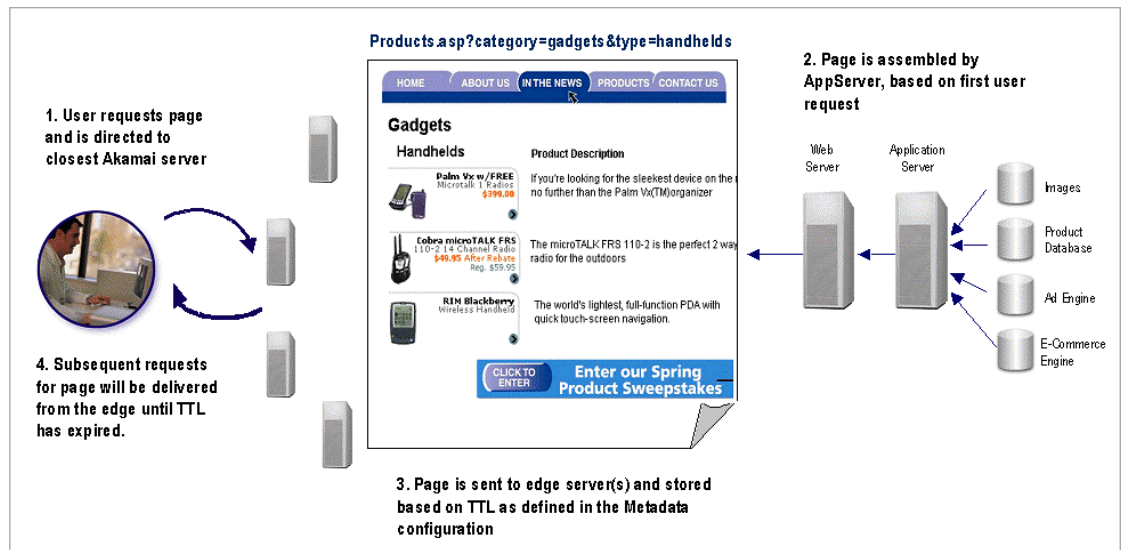


Figure 3: Workflow with EdgeSuite

Although the page was created dynamically, the entire page can still be stored on the Akamai network, since product images and description are common components and don't vary from user to user. Even though it was assembled for an individual user, there are no user-specific components (i.e. personalization through cookies) on the page that might prohibit it from being cached. It is a unique page that can be cached on the edge.

Many Web destinations, mainly portal sites, use personalization to create a unique user experience. The look and feel and content of such a site is determined by an individual's preferences, geographic location, gender, etc. By nature these sites heavily rely on application servers and/or content management systems and the use of cookies to create a dynamic and personalized user experience. The majority of pages on these sites are considered "un-cacheable," and content distribution from the edge of the Internet seems impossible.

Even serving truly uncacheable content through Akamai is generally faster and more reliable than having end users go directly from their browser to the origin Web servers. The origin site maintains persistent connections to a finite number of Akamai edge servers, rather than trying to do this with millions of individual browsers. A persistently maintained connection between the origin server and Akamai speeds up requests, making them more reliable and less variable in performance, and offloading quite a bit of CPU and memory from origin servers. Performance improvements result from keeping the connection open between the edge and the origin server with data flowing through it, thereby avoiding the overhead associated with setting up a separate connection for every browser request.

Furthermore, Akamai can use data compression to reduce further the amount of traffic between the origin server and edge server. If the requesting browser supports compression, the Akamai edge server will send compressed content to the user. In case the browser doesn't support compression, the edge server will decompress the content and send it to the browser uncompressed. Akamai's servers can also forward or process most commonly used technologies employed for personalization, such as user agents, cookies, and geographic location.

EdgeSuite contains the ability to dynamically assemble content at the edge of the Internet on the Akamai servers. To provide this capability, Akamai, together with leaders in the application server and content management system industry, developed Edge Side Includes or ESI. Edge Side Includes is a simple markup language used to define Web page fragments for dynamic assembly at the edge. Dynamic assembly can improve site performance by caching the objects that comprise dynamically generated HTML pages at the edge of the Internet, close to the end user. Businesses can design and develop the business logic to form and assemble the pages, using the ESI language within their HTML development environment. Today, ESI provides businesses with the following capabilities:

- **Inclusion:** The central ESI feature is the ability to fetch and include files to comprise a Web page, with each file subject to its own configuration and control—its own specified Time-To-Live in cache, revalidation instructions, and so forth.
- **Environmental variables:** ESI supports the use of a subset of standard CGI environment variables such as cookie information. These variables can be used inside ESI statements or outside ESI blocks.
- **Conditional inclusion:** ESI supports conditional processing based on Boolean comparisons or environmental variables.
- **Exception and error handling:** ESI allows specification of alternative pages and supports default behavior such as serving default HTML in the event that an origin site or document is not available. Further, it provides an explicit exception-handling statement set.

Instead of being assembled by the application/Web server in the main data center, the application/Web server would send the page template and the content fragments to an Akamai edge server where the page would be assembled. Each content fragment can have its own cacheability profile to manage the “freshness” of the content (Figure 4). Once a user requests a page (template), the edge server will look in its cache for the included fragments and assemble the HTML on-the-fly. If a fragment has expired or is not stored on the edge server, EdgeSuite will contact the origin server via an optimized connection to retrieve the new/missing fragment. The two main benefits of this process are:

1. Faster loading pages, because pages are assembled closer to the end user, instead of on the origin server.
2. Reduced traffic on the application/Web server, because more requests can be satisfied on the edge and smaller pieces of content are being transmitted between the origin server and edge server.

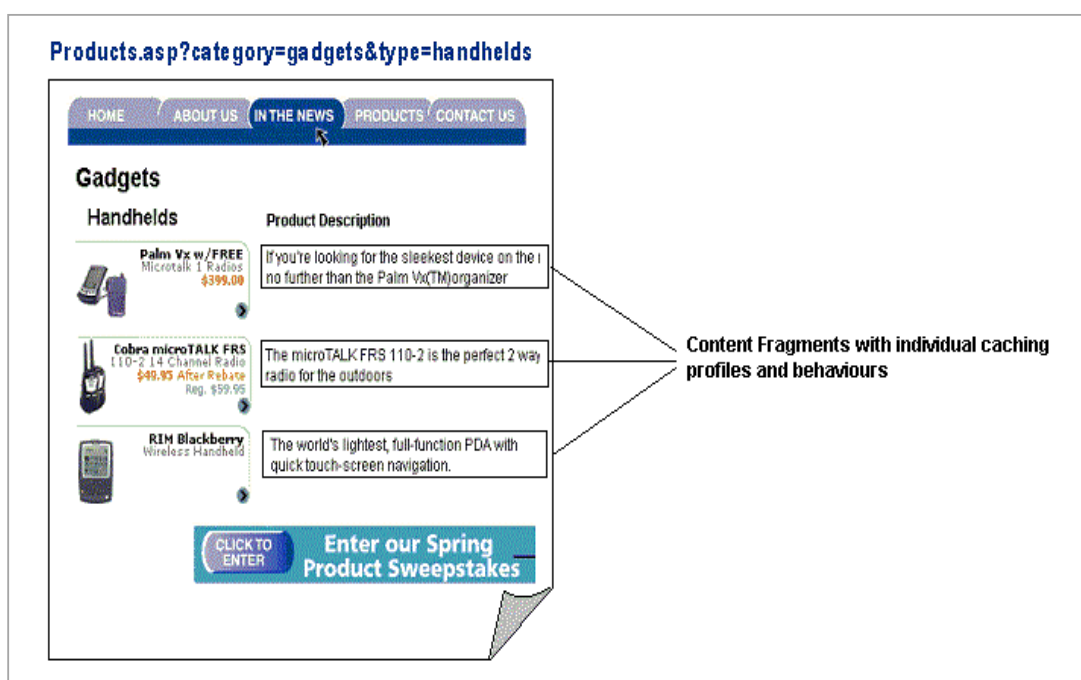


Figure 4: Dynamic Page with ESI Fragments

The integration of ESI into the leading content management systems and application servers, gives businesses great flexibility in choosing the best deployment model for an application. Web applications that use ESI can be deployed in an intranet environment where the content is being assembled on the local application server or it can be scaled to a global audience on an extranet or the Internet by simply using EdgeSuite (Figure 5). Since both the application server and the Akamai edge server understand the ESI language and content management protocol, applications can be deployed in a flexible and transparent manner, without requiring any changes to the application itself and with the benefits of reduced complexity and infrastructure costs.

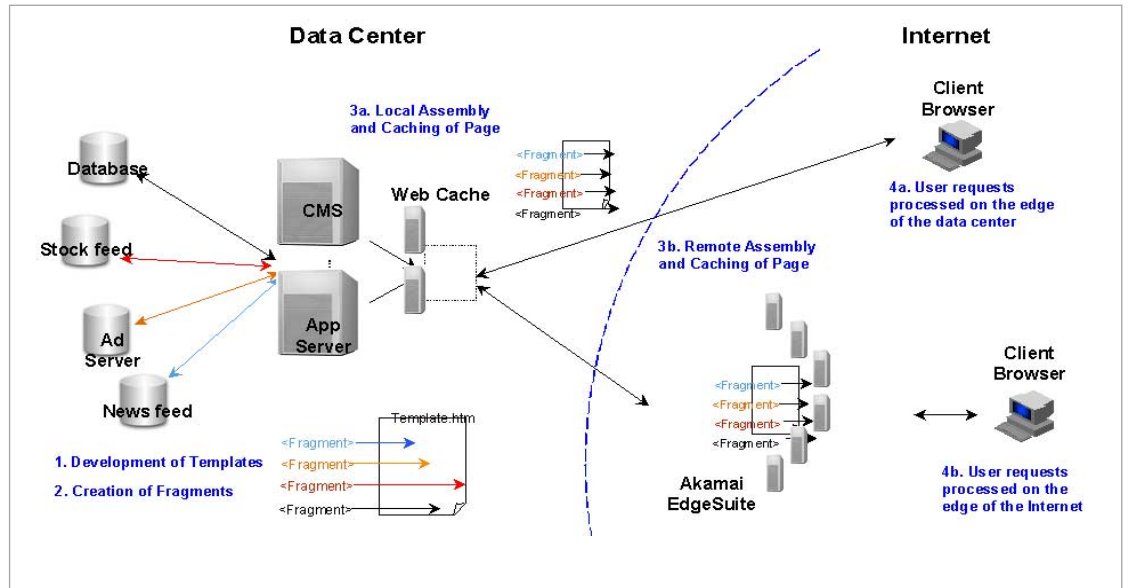


Figure 5: Flexible Content Deployment with ESI

ESI also includes a content invalidation protocol as part of the specification. With the invalidation protocol, we create a standard for publishing systems, application servers, reverse proxies, etc., to send invalidation messages to the delivery network to overwrite the metadata associated with any fragment residing on the edge servers. In this way businesses are able to control the proliferation and purging of content just as if it were residing on their local Web servers, and yet still utilize the benefits of a distributed content delivery network.

Integrated into a robust content management system, the invalidation protocol allows for seamless purging of out-of-date or undesired content from the delivery network. This can be integrated via a database call, via a script, or using a variety of other methods that can then be integrated into the site administration process used by the content server. To the customer, the process can be invisible. When a catalog item is flagged in their database, it is no longer viewable on the site, even though that site is now served from thousands of content delivery servers spread across the hundreds of networks that form the Internet (Figure 6).

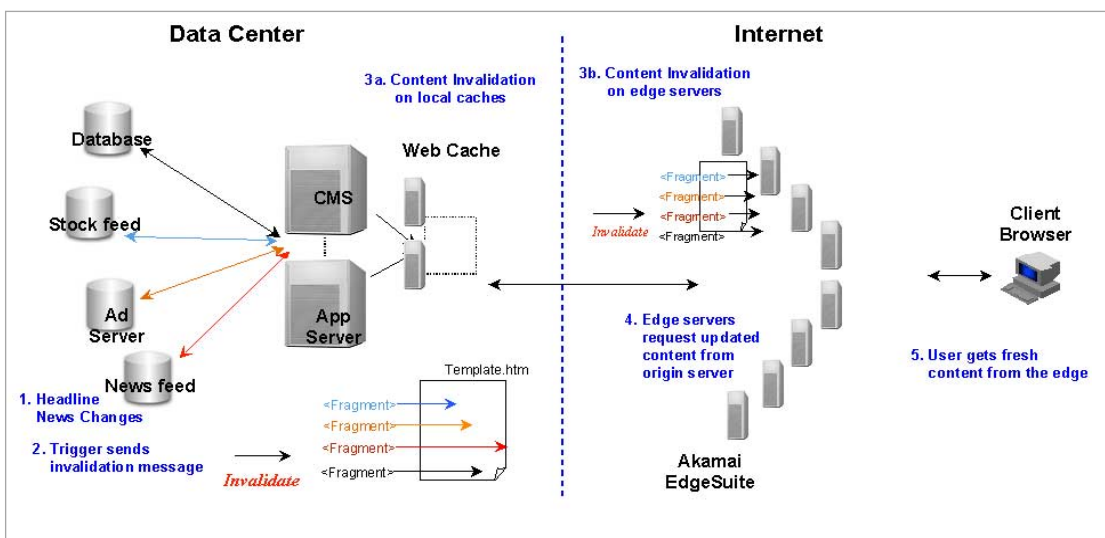


Figure 6: Transparent Content Management with ESI

Using the Akamai EdgeSuite service along with the ESI language allows you to cache, distribute, and assemble individual content fragments on the edges of the Internet. Web sites with highly dynamic content that may seem uncacheable are really simply combinations of cacheable content. By utilizing the ESI language pioneered by Akamai and now adopted by the leading application server and content management system vendors, e-businesses can dynamically assemble personalized and dynamic content on the edges of the Internet just as they do in their own data center.

The ESI language is quickly gaining popularity as more and more application vendors are adopting it and will be integrating the capability to generate and/or process ESI tags into their products. Leading application server and content management system vendors, such as ATG, BEA, IBM, Interwoven, Oracle, OpenMarket, Sun, and Vignette, along with Akamai, have co-authored or supported the ESI specification and ESI for Java (JESI). As part of this standardization effort, ESI will constantly evolve to also include XSLT processing capabilities and fragment-level security on the edge. For more information on ESI, please visit <http://www.esi.org>.

Business Benefits

EdgeSuite enhances the reliability, performance, and scalability of all sites—even those that rely heavily on dynamically generated content and personalization. EdgeSuite provides the content provider with a variety of options to deliver dynamic content and take advantage of the following benefits:

- **Instant Global Expansion of IT Infrastructure with Minimized Risk:** The global distribution of the Akamai network platform in 63 countries and more than 1,000 networks instantaneously places your vital Web content and applications close to your international audience.
- **Scalability on Demand that Grows with Your Business:** Site mirroring still doesn't solve the performance problems associated with the Internet. The fact that content is mirrored on a few networks does not make it easily accessible to a majority of the users of the Internet. Internet users are spread across thousands of ISP networks, and Web performance is nonlinear when compared to IP hops—and exponential as compared to network hops.
- **100% Reliability and Availability Insurance for Mission-Critical Applications:** Akamai's globally distributed architecture fundamentally improves reliability and availability by removing two potential single points of failure: the Internet and the central data center. Additionally, the Akamai network is self-healing and extremely redundant—overcoming the failure of individual ISP networks or the congestion of individual routes in the Internet by finding the most optimal route to serve content to users.
- **Improved Revenues and Customer Retention through Increased Performance:** The performance of Web applications that run in a distributed architecture increase substantially. A content delivery network avoids performance problems introduced by the Internet by locating and caching content near the end users. Also, moving dynamic content assembly to the thousands of servers at the edge of the network eliminates the central performance bottleneck of the application server's page assembly engines personalizing content for all users.
- **Simplicity, Uniformity, and Reduced Cost:** Akamai's edge network reduces the load on the originating site by serving static and dynamic content. Caching frequently requested content at the edge of the network decreases bandwidth requirements at the origin site. In addition, the global content delivery network allows you to extend that centralized application infrastructure into new locations by offering a uniform platform for new devices and applications.
- **Focus on Your Core Competence with Outsourced Services:** Using an outsourced global content and application distribution network eliminates the need to negotiate separate contracts with different network providers. In addition, outsourcing allows you to deploy your skilled and scant IT resources on the things that you do best—creating the content and business applications that enable your competitive advantage.
- **Gain from The Investments Made by Akamai and Its Ecosystem of Partners:** Akamai continues to make substantial investments in our network platform, giving our customers the ability to leverage the cutting edge of Web application technologies. Akamai also works closely with a wide variety of industry-leading technology vendors, system integration companies, and network partners to constantly increase the functionality and reach of our suite of solutions.

Akamai EdgeSuite is the perfect solution for even the most dynamic Web applications that e-businesses develop today to successfully compete tomorrow.

A Technical Primer

ESI Philosophy and Features

From its inception, ESI was designed to work well with the most dynamic pages present in today's Web sites. The following ESI features make it easy to build highly dynamic Web pages with ESI:

- **Coexistence of Cacheable and Uncacheable Content:** it is easy to combine cacheable and uncacheable content on the same page
- **Separation of Page Assembly Logic and Delivery:** a way to separate the complex logic required to select the content itself from the delivery of that content
- **Recursive ESI:** a way to perform ESI processing on components themselves
- **Logic on the Edge:** ESI features that perform certain personalization and conditional processing on the edge.

Co-existence of Cacheable and Un-cacheable Content

The ESI language recognizes the fact that many pages have dynamic and often uncacheable content. By breaking up Web pages into individual components, each with different cache policies, ESI makes it easy to speed up the delivery of dynamic pages. Only those components that are uncacheable or need updating will be requested from the origin server on every request. This results in a considerable speed improvements.

Page Logic and Delivery Separation

When people refer to a Web page as being “completely dynamic,” what they usually mean is that the logic that governs page generation is complex and cannot be easily expressed in ESI. However, this does not mean that the HTML fragments themselves cannot be cached by our edge servers. Even if every page is built anew for every user, the building blocks of the page are often shared between users, and can be easily cached. ESI provides a convenient way to do this.

Recursive ESI

Recursive ESI is exactly what is required to separate page logic from content delivery. Any ESI fragment can, in turn, contain other fragments. In particular, an uncacheable dynamic fragment can contain `<esi:includes>` to point to cacheable sub-fragments.

Logic on the Edge

A lot of personalization can happen on the edge. The `<esi:choose>` tag allows content providers to include different fragments based on:

- User Agent and other header values
- Cookie values
- User's location (EdgeScape feature)
- User's connection speed (EdgeScape feature)

The last two are particularly important. Akamai EdgeScape is a unique service that allows content providers to personalize Web content based their users' locations and connection speeds.

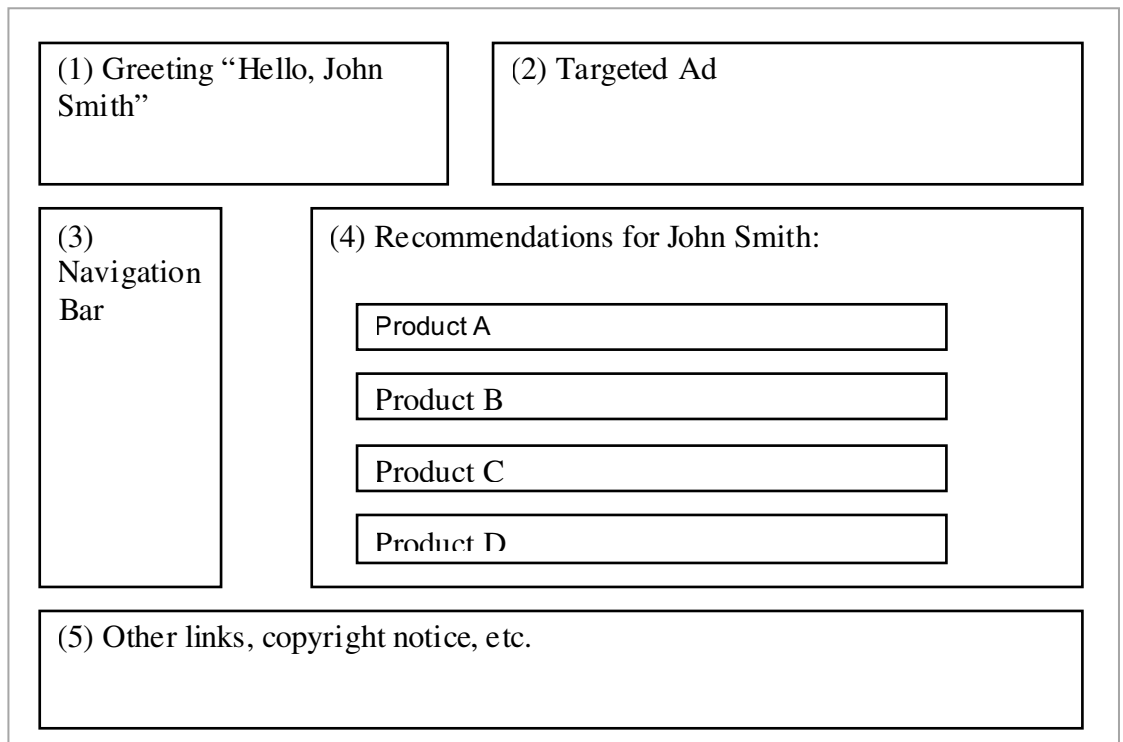
Finally, cookie-based variables can be substituted into the text of the page, which makes many previously uncacheable personalized pages easily deliverable from the edge.

Dynamic Page Example

We will illustrate how all of the above features can be used on a highly dynamic Web page from a fictional e-commerce site.

The example page from this e-commerce site will contain:

- A personalized greeting generated by a personalization engine
- A targeted advertisement generated by an ad serving technology
- A navigation bar and a footer generated by a content management system
- Several product recommendations generated by a CRM application



Please note the following elements:

- Static content (3) and (5): navigation bars, links, copyright notice.
- Personalized greeting (1)
- Targeted ad (2) that depends on user’s location
- Recommendations for the user (4), made on the basis of complicated analysis by the site’s collaborative filtering engine.

As one can see, most of the content on this page is personalized and dynamically generated. Yet it can be successfully delivered from an edge server using ESI.

The ESI version

This is what the ESI version of the page may look like:

```
<html>
<body>
<!--personalized greeting (1) -->
Hello ${HTTP_COOKIE{"username"}}

<!--targeted ad (2) -->
<esi:choose>
<esi:when test="${GEO{country_code}} == 'US' ">
    <esi:include src=us_ad.html/>
</esi:when>
<esi:when test="${GEO{country_code}} == 'Canada' ">
    <esi:include src=canada_ad.html/>
</esi:when>
<esi:otherwise>
    <esi:include src=generic_ad.html/>
</esi:otherwise>
</esi:choose>

<!-- Static navigation bar (3) -->
<a href=...> <a href=...> <a href=...> <a href=...>

<!-- Personalized recommendations (4) -->
<esi:include src=recommendations.html/>

<!--Static links, copyright, etc (5) -->
<a href=...> <a href=...> <a href=...> <a href=...>
Copyright 2001, etc.
</body>
</html>
```

It's important to notice the following:

- Coexistence of Cacheable and Uncacheable Content: static (3 and 5) and dynamic (1, 2, and 4) blocks are combined on the same page. Static blocks become a part of the template and dynamic blocks are included using various ESI commands.
- Recursive ESI, Page Logic, and Delivery Separation: to see how this works, we need to look closer into block (4) recommendations.html. This block is uncacheable, and the request is tunneled back to the origin server. What's returned, however, is not the full HTML block but the list of references to the recommended products:

```
<esi:include src="products/A.html" />
<esi:include src="products/B.html" />
<esi:include src="products/C.html" />
<esi:include src="products/D.html" />
```

Note that each of the product descriptions is cacheable. In fact, it is likely the total number of products recommended to all the users can be easily cached on the edge. The logic to generate the recommendations resides at the origin server, but the actual HTML is cached and delivered from the edge server. Also note that because requests for uncacheable fragments like recommendations.html are always tunneled to the origin, they can be used to update the session state information. Therefore, user recommendations may depend on previous pages visited.

- **Logic on the Edge:** fragments (1) and (2) illustrate Logic on the Edge:

In fragment (1), the value of cookie “username” is substituted into the body of the page to produce a personalized greeting.

Fragment (2) illustrates personalization on the edge and an ESI conditional for which advertisement to include, which is dependent on the user’s geographic location. If the user is from the USA, the us_ad.html fragment is included. If the user is from Canada, then canada_ad.html is included. Otherwise, a generic ad is shown. The Akamai EdgeScape service automatically makes information about user’s location available to the content providers. Of course, us_ad.html, canada_ad.html, and generic_ad.html can all be cached on the network.

Even though most of this example Web page is generated dynamically, the majority of the fragments making up the page can be cached and delivered from the edge. The amount of data that has to be retrieved from the origin site is very small. This results in a significant performance improvement for the end user and a reduction of infrastructure required to deliver the site.

ESI Integration

ESI is being developed by industry-leading application vendors to be easily integrated into enterprise data centers. To that regard, a JavaServer Pages library that interprets JSP tags into ESI tags has been developed. This allows for businesses that are already using the JSP within their system processes to more easily utilize the benefits of the ESI language.

The JESI custom tag library allows a user to:

- Separate a JSP page into ESI fragments without moving sections of content into separate files
- Execute certain JSP content only when desired
- Specify metadata in JSP files

In addition, the ESI language has been natively integrated into the Oracle 9i Application Server and will soon be integrated into Oracle’s JDeveloper J2EE development tool set. These integrations allow a developer to preview, test, and proliferate their content within the application server development environment that they are already familiar with. ESI Integration into other industry-leading development tools will soon follow.

Conclusion

ESI is well suited for delivering highly dynamic and personalized pages. It was explicitly designed to support modern web sites, and it has a variety of features for delivering dynamic content.

About Akamai

Akamai is the leading provider of secure, outsourced e-business infrastructure services and software. These services and software enable companies to reduce the complexity and cost of deploying and operating a uniform Web infrastructure while ensuring unmatched performance, reliability, scalability, and manageability. Akamai's services and world-class customer care give businesses a distinct competitive advantage and provide an unparalleled Internet experience for their customers. Akamai's intelligent edge platform for content, streaming media, and application delivery comprises more than 13,000 servers in more than 1,000 networks in 63 countries. With headquarters in Cambridge, Massachusetts, Akamai provides services to companies worldwide. For information on Delivering a Better InternetSM, visit www.akamai.com.

© 2001 Akamai Technologies, Inc. All Rights Reserved. Reproduction in whole or in part in any form or medium without express written permission is prohibited. Akamai is a registered trademark and service mark. EdgeSuite and the Delivering a Better Internet slogan are Akamai service marks. All other trademarks contained herein are the property of their respective owners. Akamai believes that the information in this publication is accurate as of its publication date; such information is subject to change without notice.

AKAMWP-TC1201



Akamai Technologies, Inc.

US Headquarters

500 Technology Square, Cambridge, MA 02139

Tel 617.444.3000

Fax 617.444.3001

US toll free 877.4AKAMAI

(877.425.2624)

European Headquarters

Heisenbergbogen 2, 85609 Dornach, Germany

Tel +49.89.94.00.60

www.akamai.com