

# single sign-on capability

Technology that provides corporate users with access to all of their information through a single portal website.

- case study: a growing internet service provider** .....3
- login service architecture** .....4
  - requirements for a login server .....4
  - login server components .....5
- hp solutions** .....6
  - solution components on NonStop servers .....6
  - solution components on hp servers running the HP-UX operating system .....7
  - solution components on ProLiant servers .....7
  - eBOSS software .....8
  - hp NonStop data transformation engine software .....9
- the role of the hp ZLE framework in single login solutions** .....11

This paper relates HP’s experience with architectures in which a login server controls user (or customer) access to relevant applications and websites. It begins with a case study, which sets the stage for the general discussion that follows.

A business objective for many institutions is to construct a capability that enables users to access all of their relationships with the institution through a single portal website that requires only a single login, regardless of the separate applications that may invoke. Typically, each application is constructed to require its own login, often with unique login ID and password. The burden is usually placed on the user to remember and enter these passwords for each function or application they invoke. A single login portal would remove that burden from the user and facilitate their access to those applications that are authorized to them. This single login capability could also lead to providing the institution with a unified view of each user's interactions, improving the quality of the "real-time" service. Figure 1 illustrates one example of a login server architecture.

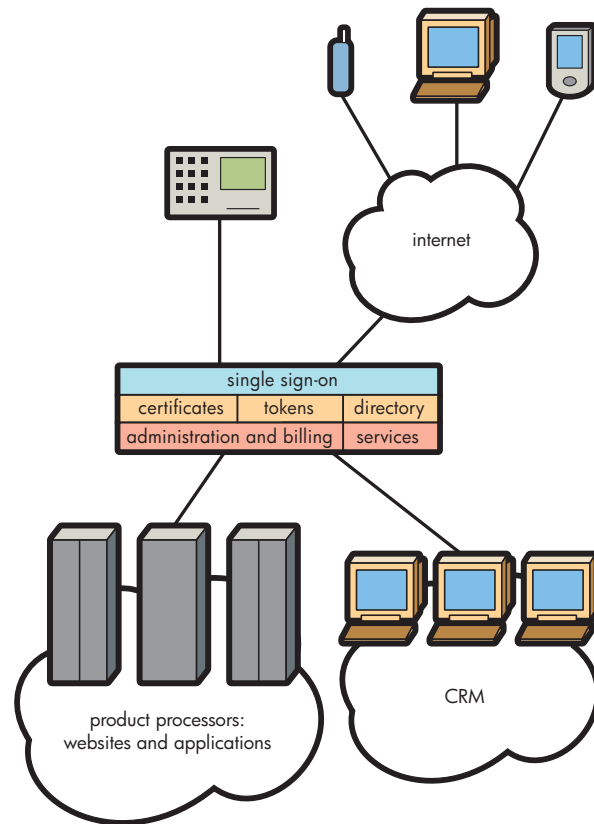


Figure 1. High-level architecture for single sign-on capability.

## case study: a growing internet service provider

HP implemented a login server with single sign-on capability for a major U.S. Internet service provider (ISP) facing rapid growth. This ISP's subscribers were spending more time online, resulting in higher revenues for subscriptions, advertising, and product sales. In response to market growth, the ISP defined the following strong growth requirements of its infrastructure:

- Expand servers according to the business demand without outages for forklift upgrades
- Expand databases with the business, without expensive replications
- Let servers expand applications transparently without expensive reconfigurations
- Expand without allowing planned outages for growing capacity

In addition, with tens of millions of subscribers, high rates of peak concurrent use, and millions of sessions per day, the ISP was extremely sensitive to server performance and manageability requirements for key customer-facing applications. Given these growth requirements, the ISP adopted an architecture based on the HP NonStop server platform as the basis for its login service, following a short proof-of-concept test. The decision to choose NonStop servers was based on several unique advantages of this platform, including

- The highest level of system and application availability in the industry
- The highest level of system scalability to cater to the higher transaction volumes, with future growth at the least possible cost and with the least system disruption
- A unique single system image (SSI) capability

By utilizing the technology of this platform for its critical application, the ISP was able to provide a fully fault-tolerant infrastructure for access to Internet-based services. This ensured the highest possible customer service levels. The SSI capability enabled the organization to physically distribute system components while operating the entire network as a single logical application.

## login service architecture

Figure 2 presents a high-level view of the architecture for a general login service. The service includes authorization and billing components, both of which must be executed reliably to enable user access to other applications. The architecture uses a lightweight login service that relies on the billing server being fault tolerant to avoid complex logic at login. The login service is similar to the stand-in processor of an automated teller machine: It needs a read-only authorization data file, which must be streamlined for high-volume use. The authorization file is updated on a regular basis, with an exception method available for interim crisis updates. The login server also generates alerts programmatically via Simple Network Mail Transmission Protocol (SNMTP) or JMail, for example, to notify operators of unusual circumstances.

The login service maintains the customer state in the billing database. Therefore, this database must be transactionally oriented and available on a 24 x 7 basis. The use of three copies of the database provides extra security for disaster recovery and for offline accounting and reporting functions. The primary database is the only one transactionally protected. The others are near-real-time copies of the database that are asynchronously fed from the primary server. Asynchronous feeds are used to populate the other databases to avoid slowing down the primary processing and avoid consuming high levels of processing power, while still keeping the databases current with the login activity.

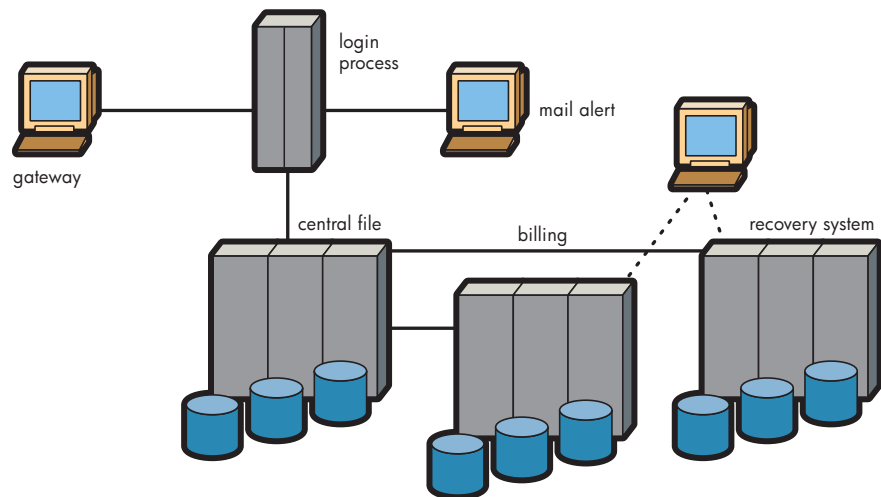


Figure 2. High-level architecture for a login service.

### requirements for a login server

There are several natural requirements for a single sign-on login server:

- Availability
- Scalability
- Manageability

For a single sign-on system to be successful, the service must be available continuously. Even planned downtime in such a system is unacceptable. Given the global nature of many corporations, there is no good time to take the service offline.

If the service is successful, then the number of hits a site receives is likely to increase by orders of magnitude, as customers are enrolled and become comfortable with the facility. The service must be capable of adding capacity in a planned fashion without requiring outages or serious reprogramming. To limit growth in the cost of operations as the scale and complexity of the service increases, the entire service must be manageable and auditable as a single, stable configuration.

### login server components

The major components needed for a login server to achieve single sign-on capability include

- Web server
- Java™ applet and servlet support
- Security authorization application
- Security audit logging capability
- Validation repository (database)

Figure 3 illustrates such a login server. The architecture could also integrate Wireless Application Protocol (WAP) and the tracking of customer usage data for customer relationship management (CRM) purposes.

The login server validates the customer login using an applet for the secure dialogue. Once validated, the customer is presented with a menu of available choices (using a Java servlet application running on the login server). Using typical Web technologies, the server redirects the customer to the already constructed website chosen with the addition of a security authorization token that identifies the user to the website application. In addition to providing the authorization token and directing user traffic, the menu servlet application maintains a security audit log for internal use.

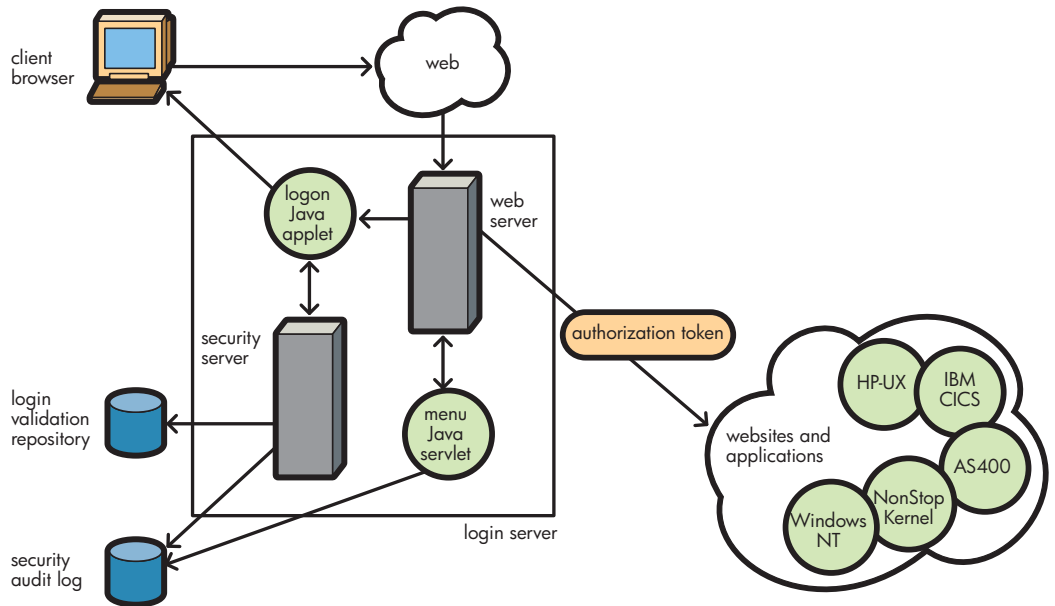


Figure 3. Architecture for a single sign-on login server.

## hp solutions

HP offers three server families as potential solutions for a login server:

- HP ProLiant servers running the Microsoft® Windows NT Server, Windows 2000, or Linux operating system
- HP servers running the HP-UX operating system
- HP NonStop servers

HP also offers all of the components needed to construct a login server, using any of these server families or a combination of them. Each server has its unique strengths, which may lead an institution to prefer a particular solution. NonStop servers offer the highest availability at the application and database level, including database and application management, which position them for high-volume front-end services. As discussed earlier, a major U.S. ISP uses this family of servers as its front-end login platform with a custom application.

HP servers running the HP-UX operating system offer a fusion of product, services, and solutions that demonstrates HP's "adaptive infrastructure" vision. HP servers range from single processor servers to high-end servers, such as the HP Superdome server. Many institutions already use HP-UX systems with Oracle® databases for fast response applications.

ProLiant servers offer the lowest entry cost and have a vast array of tools and components available. Many application service providers (ASPs) have adopted ProLiant servers as their Web-facing front-end systems.

### **solution components on NonStop servers**

The HP NonStop server platform and components meet the requirements of a login server fully. The platform has a long and venerable track record in applications that have the same availability, scalability, and manageability requirements, including telecommunications companies, ISPs, stock exchanges, and major money center banks. NonStop server databases support a mixed workload environment with predictable service levels. That is, they consistently sustain transactional inserts, batch extracts, online transaction processing (OLTP), and massively parallel queries against the same tables concurrently without degrading service levels.

Single sign-on solutions built with NonStop servers include the following components:

- HP iTP WebServer software
- HP NonStop Server for Java software (combined with either BEA's WebLogic Server or Iona's E2A software for Enterprise JavaBeans capability)
- eBOSS security application software (distributed by USA Software)
- Enscribe logging database and validation repository

## **solution components on hp servers running the HP-UX operating system**

HP servers running the HP-UX operating system support the requirements of a login server by providing complete solutions for clustered high availability to eliminate single points of failure. Adaptive infrastructure allows a system manager to optimize the allocation of resources between competing workloads; run multiple operating systems, including different platforms for test, development, and deployment; and scale resources to optimize application performance. For this solution, the application server may run on one set of UNIX® based servers from HP, while the database is implemented with others.

The Superdome server platform is the high-end UNIX server in the HP server product family. It's designed to deliver the performance, availability, capacity, security, and manageability needed for the "mission-critical Internet" and other compute-intensive applications. Featuring advanced partitioning, multiple operating system capabilities, and several innovative pricing options, Superdome servers provide high availability with the highest performance characteristics, leveraging the Intel Itanium microprocessor, while supporting a wide range of industry-standard middleware and applications.

Single sign-on solutions built with servers running the HP-UX operating system include the following components:

- Netscape Web server and Lightweight Directory Access Protocol (LDAP) validation repository
- Java Virtual Machine (JVM)
- Oracle Relational Database
- BEA WebLogic Server J2EE application integration middleware
- BEA Tuxedo software, which manages the application, database server, legacy system, and interface
- HP OpenView management software

## **solution components on ProLiant servers**

ProLiant servers provide a substantial measure of support for the requirements of login servers, especially with a Distributed Internet Server Array (DISA) approach. In DISA, multiple Web servers are used in a round-robin fashion to achieve high availability and scalability. Further separating functional layers in a DISA server array offers for additional modularity and scalability. HP Insight Manager XE software provides a basis for managing the complex DISA server array in a cost-effective fashion.

A single sign-on solution built with ProLiant servers includes the following components:

- Microsoft Internet Information Server (IIS) and BizTalk integration middleware
- TIBCO, NEON, and Faire-Isaac application integration middleware
- Oracle or Microsoft SQL Server relational database
- BEA WebLogic Server software

## eBOSS software

eBOSS software, distributed by USA Software, provides access control, security management, and auditing of applications in an open architecture. The software also enables a single sign-off capability, even with multiple applications. This heterogeneous access control product is the extension of the BOSS security product for the NonStop server platform. Figure 4 depicts the architecture for a portal constructed with eBOSS on NonStop servers.

eBOSS software allows access from a Web browser to a variety of target systems and applications that could reside on platforms such as mainframes or on systems running Windows NT, Windows 2000, UNIX, or Linux operating systems. eBOSS still uses the NonStop server platform as the secure, fault-tolerant platform “glue” that holds all of this open architecture together. The Windows NT Server system exemplifies an external application that could be integrated with eBOSS security for a single login service. Product highlights include

- Context checking (regardless of application)
- Ability of the BOSS Manager to allow or disallow access to any application on the BOSS Menu to a set of users
- Ability to completely deny users’ ability to visit a website outside of what they are allowed via the eBOSS Menu
- Complete audit trails
- Secure access to applications that run on Windows NT, Windows 2000, Sun Microsystems, Linux, and UNIX (including HP-UX) operating systems; the Internet (HTTP applications); and website authorizations
- Secure access to applications running on a NonStop server from a user’s Web browser by way of the Internet

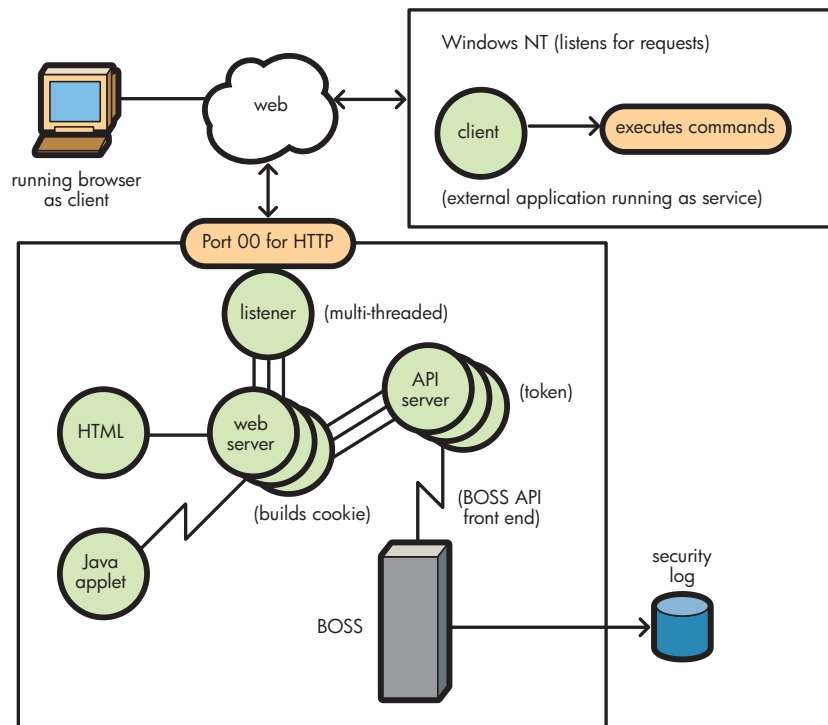


Figure 4. Architecture for a portal constructed with eBOSS software on a NonStop server.

## hp NonStop data transformation engine software

HP NonStop Data Transformation Engine (DTE) software provides a high-performance and cost-effective alternative to writing or generating programs that translate diverse data models used by different applications into a common data format. It is based on Mercator technology, which is recognized in the industry as being best of breed in the category of data transformation products. Mercator worked closely with HP to customize the Mercator Enterprise Broker software and deliver powerful data transformation technology across a heterogeneous business environment. NonStop DTE software consists of Design Studio, Platform API, adapters, and servers. Figure 5 shows how these component groups are interrelated.

NonStop DTE software is a powerful data integration tool that has the ability to grasp the structure, content, and semantics of any application data, as well as completely transform the data, without the need for programming. It is a complete data integration offering, including automatic data transformation tools and more than 100 predefined rules for dynamically acting on the content of data inputs and outputs. NonStop DTE software can be deployed as either a NonStop Tuxedo service or a NonStop CORBA object. The NonStop server cluster provides scalability, high availability, and load balancing. This ensures that real-time translation of data does not become a bottleneck in online, integrated application environments.

NonStop DTE software consists of mapping tools and repositories used to create data integration maps, along with the actual engine, which performs the transformation, taking maps as input. The development process is easy and includes three steps.

1. The first step is to create transformation maps by associating input data with a type tree.
2. The second step is to map input data to output data. Optional input data processing (for example, validation rules) can be specified. Rules can be combined and nested in ways that provide robust retry, restart, and recovery capabilities when bad data is encountered.
3. The third step is to execute integration maps. During this step, the engine dynamically retrieves the map for incoming data, performs the transformation, and sends the translated data to the receiving applications.

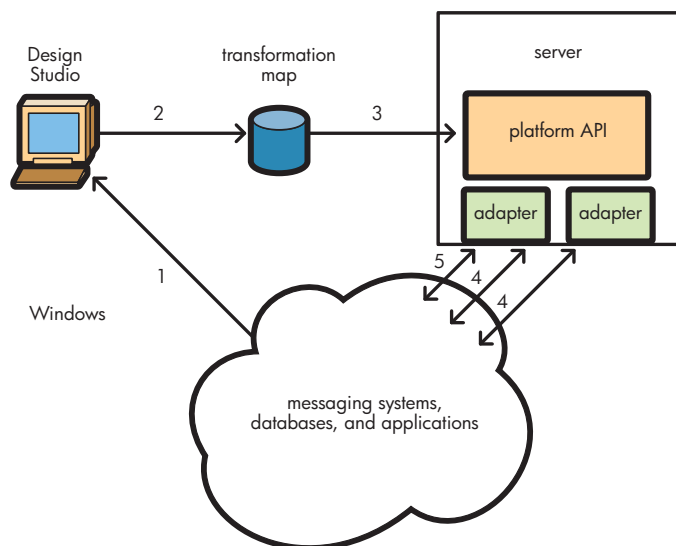


Figure 5. Components of NonStop DTE software.

A map object can represent either a simple transformation (common to many applications) or a full mapping solution. With transformation rules, you can go beyond simple reformatting and construct an entirely new business object. If required, you can create rules to route newly formed business objects to appropriate targets based on the data content. When dealing with complex objects, you can develop rules to filter, scrub, calculate, validate, parse, substitute, expand, change character set, or otherwise convert data from multiple inputs to multiple outputs. You can compute data, look up something in a database, sort, extract, merge, or use any of dozens of predefined functions to create output.

You can also map an output to another output. For example, you can map an output that was generated at the top of an output, to the bottom of an output—supporting the need to summarize, tally, or validate what has been created. You can also map an output that was generated in one output, to another output. For example, you can generate a message as one output and archive it as another output.

Multiple input sources and output targets can be used in a single map to provide any-to-any mapping in terms of content. Support is designed for any-to-many, many-to-any, and many-to-many data sources and targets.

## **the role of the hp ZLE framework in single login solutions**

In addition to the architecture described, HP's Zero Latency Enterprise (ZLE) framework may also leverage the single login solution by enhancing and extending legacy systems and processes without reengineering to maintain a virtual company structure while providing a real-time unified customer view. This view may be extended beyond existing systems and processes to address new business opportunities as well. For example, in the initial phase of an ongoing project, one HP client implemented a ZLE framework to integrate existing point solutions that address 22 distinct, critical data sources to provide a single real-time view across all customer touchpoints.

The single login solution could be augmented by the addition of application integration adapters to capture and maintain corporatewide transaction data in high-performance operational data stores (ODSs). The ZLE framework is noninvasive. Existing applications and systems continue, with the framework providing a parallel and continually available information source that is up to date, with all customer events from all customer touchpoints. Applications continue to process work as usual while a copy of the transaction or event is captured in real time and updated to the ODS.

For additional information on the HP ZLE framework, visit <http://zle.nonstop.compaq.com>.

For more information, go to [www.hp.com/go/nonstop](http://www.hp.com/go/nonstop).

March 2003, first published 2001. Intel and Itanium are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries and are used under license. Microsoft, Windows, and Windows NT are U.S. registered trademarks of Microsoft Corporation. UNIX is a registered trademark of The Open Group. Oracle is a registered U.S. trademark of Oracle Corporation, Redwood City, California. Java is a U.S. trademark of Sun Microsystems, Inc. All other product names mentioned herein may be trademarks of their respective companies. HP shall not be liable for technical or editorial errors or omissions contained herein. The information is subject to change without notice. The warranties for HP products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

5981-6526EN

©2003 Hewlett-Packard Development Company, L.P.

