



## **ATLAS**

The Advanced Transaction Look-up and Signaling Platform

Version 1.0

January 23, 2003

This document is the property of VeriSign Global Registry Services, Inc. It may be used by recipient only for the purpose for which it was transmitted and will be returned upon request or when no longer needed by recipient. It may not be copied or communicated without the prior written consent of VeriSign Global Registry Services.

## **COPYRIGHT NOTIFICATION**

Copyright© 2001, VeriSign, Inc. All rights reserved.

## **VERISIGN GLOBAL REGISTRY SERVICES PROPRIETARY INFORMATION**

This document is the property of VeriSign, Inc. Information contained herein may include trade secrets and confidential information belonging to VeriSign. Unauthorized disclosure without the express written consent of VeriSign, Inc. is prohibited. It may be used by recipient only for the purpose for which it was transmitted and will be returned upon request or when no longer needed by recipient. It may not be copied or communicated without the prior written consent of VeriSign, Inc.

## **DISCLAIMER AND LIMITATION OF LIABILITY**

VeriSign, Inc. has made efforts to ensure the accuracy and completeness of the information in this document. However, VeriSign, Inc. makes no warranties of any kind (whether express, implied or statutory) with respect to the information contained herein. VeriSign, Inc. assumes no liability to any party for any loss or damage (whether direct or indirect) caused by any errors, omissions or statements of any kind contained in this document. Further, VeriSign, Inc. assumes no liability arising from the application or use of the product or service described herein and specifically disclaims any representation that the products or services described do not infringe upon any existing or future intellectual property rights. Nothing herein grants the reader any license to make, use, or sell equipment or products constructed in accordance with this document. Finally, all rights and privileges related to any intellectual property right described in this document are vested in the patent, trademark, or service mark owner, and no other person may exercise such rights without express permission, authority, or license secured from the patent, trademark, or service mark owner.

VeriSign Inc. reserves the right to make changes to any information herein without further notice.

## **NOTICE AND CAUTION**

### **Concerning U.S. Patent or Trademark Rights**

The inclusion in this document, the associated on-line file, or the associated software of any information covered by any patent, trademark, or service mark rights will not constitute nor imply a grant of, or authority to exercise, any right or privilege protected by such patent, trademark, or service mark. All such rights and privileges are vested in the patent, trademark, or service mark owner, and no other person may exercise such rights without express permission, authority, or license secured from the patent, trademark, or service mark owner.

This publication was created using Microsoft® Word 2000 for Windows® by Microsoft Corporation. Microsoft is a registered trademark and Windows is a trademark of Microsoft Corporation.

### **Version 1.0**

January 23, 2003

VeriSign® Global Registry Services  
21345 Ridgeway Circle  
Dulles, VA 20166-6503  
E-mail: [info@verisign-grs.com](mailto:info@verisign-grs.com)  
Internet: <http://www.verisign-grs.com>

# CONTENTS

<b>Introduction</b>	1
A. What is ATLAS?	1
B. ATLAS Architectural Elements	1
C. ATLAS Performance Capabilities	2
D. Why VeriSign Developed ATLAS	2
1. <i>The Domain Name (Web Address) Market</i>	2
2. <i>ATLAS is Born</i>	2
E. What Types of Problems does ATLAS Solve?	2
1. <i>Characteristics of an ATLAS Problem</i>	2
2. <i>ATLAS as Directory Infrastructure</i>	3
<b>Examples of ATLAS Applications</b>	4
A. Example #1: Teleblock	4
1. <i>Market Drivers</i>	4
2. <i>Problem</i>	4
3. <i>Solution</i>	4
B. Example #2: Dynamic Location and Routing	4
1. <i>Market Drivers</i>	4
2. <i>Problem</i>	5
3. <i>Solution</i>	5
<b>Vision for the Future</b>	5
<b>Conclusion</b>	6
<b>Appendix: Technical Overview of ATLAS</b>	7
A. ATLAS Architectural Elements	7
1. <i>Data Extraction Process</i>	7
2. <i>Data Distribution Process</i>	7
3. <i>Look-up Engines</i>	7
B. Data Extraction Process	7
1. <i>Initial Send Files (ISF's)</i>	8
C. Data Validation	8
1. <i>Validation Process</i>	8
D. ATLAS Look-up Engine	9
1. <i>Protocol Engines (PE's)</i>	9
2. <i>Look-up Engine (LUE's Subcomponent)</i>	9
3. <i>SuperPacket Protocol</i>	10
4. <i>Minimum Unit (MU)</i>	10
E. Data Distribution	10
1. <i>Update Process</i>	11
2. <i>Checkpoints</i>	11
F. Monitoring	12

## INTRODUCTION

The purpose of this document is to provide the reader with a broad understanding of ATLAS, which is VeriSign's global platform for deploying the next generation of directory services, including its potential applications, and technical operation. It is hoped that this document will provide enough of an understanding of ATLAS in specific, and directories in general, to stimulate the reader to conceptualize new ATLAS applications.

### WHAT IS ATLAS?

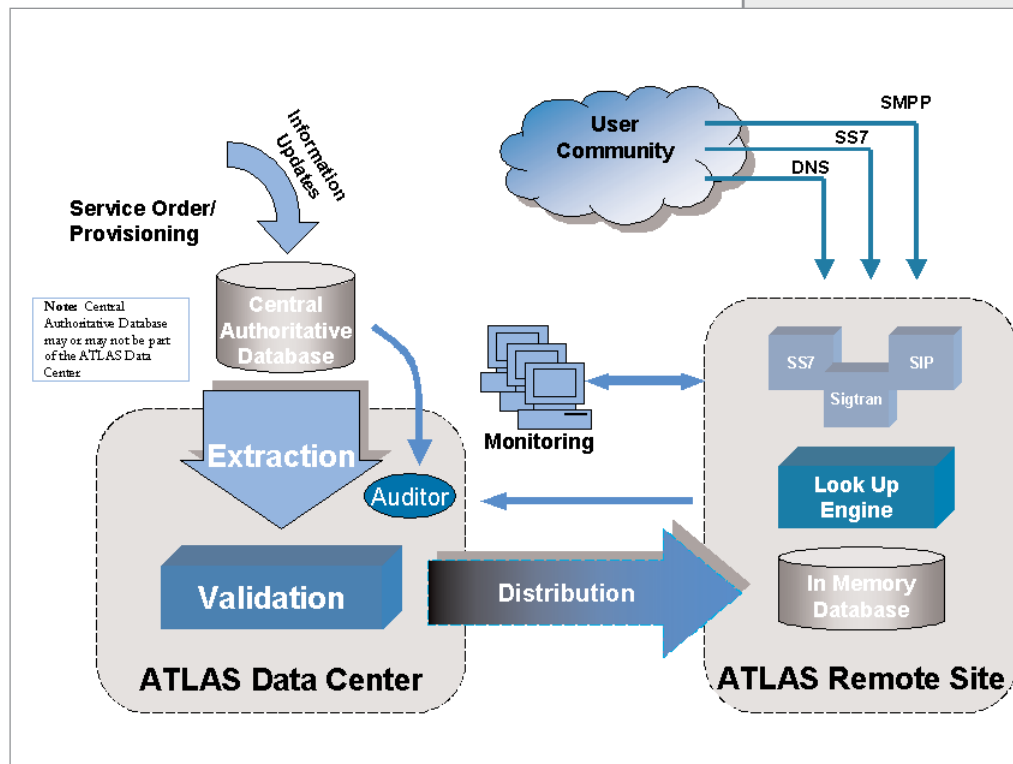
The Advanced Transaction Look-up and Signaling platform, or "ATLAS," is the next generation platform currently used to handle over seven billion queries daily for the .com and .net domain name registries—the largest, fastest-growing directories of their type. VeriSign has developed ATLAS to be a real-time distributed directory system that is designed to support other business and technical requirements based on a wide variety of protocols and database applications. ATLAS provides "dial tone" availability for distributed directory applications, such as those required by financial services, health care and government organizations, where the right answer is critical, every time, on time.

### ATLAS ARCHITECTURAL ELEMENTS

ATLAS consists of three architectural elements:

- a data extraction process which connects to an authoritative database and extracts, validates and prepares the data for distribution;
- the data distribution process that insures that the data in the authoritative database is faithfully reproduced in the distributed lookup engines;
- and distributed lookup engines that maintain local copies of the data to provide fast, efficient retrieval of the information.

The architectural elements are explained in detail in the appendix of this document.



## ATLAS PERFORMANCE CAPABILITIES

The performance capabilities of ATLAS are staggering. It is capable of answering more than 100 billion look-ups per day across datasets of more than 500 million records. Its globally distributed look-up servers can receive incremental updates in less than 60 seconds—even over limited bandwidth media. The multi-protocol provisioning (RRP, EPP, WWW, LDAP, etc.) and lookup (DNS, SS7, Sigtran, SIP, OCSP, etc.) capabilities of ATLAS combine to make it an extremely flexible, distributed database platform.

## WHY VERISIGN DEVELOPED ATLAS?

### The Domain Name (Web Address) Market

VeriSign developed ATLAS out of necessity. During the dot com boom, domain names were being registered at a blistering pace, growing from 900,000 new registrations per quarter in Q1 1999 to 5.6 million new registrations per quarter in Q2 2000. As new Web addresses were added, the number of people trying to access the information located at Web addresses also increased. Hence, the number of queries or load on the master servers or the global top-level domain servers (*gtlds*) grew as the Internet grew. To support this increasing load, VeriSign continued to add very expensive, state-of-the-art equipment including IBM M80 servers and EMC storage arrays.

As the dot com bubble began to burst in mid-2001, the volume of new Web address registrations decreased drastically. Between Q2 2000 and Q4 2001, new registrations dropped from 5.6 million to 2.3 million per quarter. Since the VeriSign Registry receives a flat fee of \$6 per year for each .com and .net Web address registered, their revenue growth began to decline. Further, even though the absolute number of domain names leveled off, the growing popularity of the Internet caused the number of queries on *gtd* servers to increase (a service for which VeriSign does not receive revenue) at a rate of 100 percent every 12 months. Costs were increasing and revenue growth was decreasing for VeriSign's registry business—not a sunny business pattern. While the outlook was nowhere near fatal to VeriSign, it was obvious that infrastructure adjustments were necessary to support the emerging business trend.

### ATLAS is Born

There are two key variables in any business equation—revenue and cost. Since market forces beyond VeriSign's control drove revenue, VeriSign focused on controlling capital and operating expenses while still providing the world-class service for which they are known. The VeriSign engineers' design goal was to develop an infrastructure that could scale massively to handle increasing amounts of data and numbers of queries without a commensurate increase in cost. ATLAS is the fruit of their labor.

## WHAT TYPES OF PROBLEMS DOES ATLAS SOLVE?

Although ATLAS was initially designed to support DNS infrastructure, the protocol also has applicability outside of the world of DNS. ATLAS is especially well suited in applications where distributed data look-ups are essential. Where a data look-up means a binary query (is this exact string in the list or not) on a data record that is relatively small in size (200 bytes or less). This is opposed to a conventional database which is optimized to search for items that have some relation to the string requested and manage large record sizes well (Oracle, DB2, etc.).

### **Characteristics of an ATLAS Problem**

In identifying the characteristics of an ATLAS problem it is helpful to begin with gaining an understanding of ATLAS' primary attributes. ATLAS is able to:

- Scale to handle the largest look-up problems while avoiding proportionate increases in costs
- Maintain globally distributed look-up databases with real-time updates
- Process data adds, deletes and modifications instantaneously
- Communicate using multiple data provisioning and resolution protocols

Using the proceeding as a decision framework, ATLAS addresses two classes of problems. First, situations where the speed of a data look-up and the capacity of the system is the primary concern, and/or where data needs to be replicated locally in a number of geographies or systems. Second, it addresses situations that require the highly scaleable nature of ATLAS and its ability to communicate via multiple protocols.

### **ATLAS as Directory Infrastructure**

The ATLAS infrastructure is well suited in applications that rely on directory services because they typically have relatively small record sizes and need to be widely available to a diverse set of clients. A well-known example of a widely-used directory is the Internet DNS system. The VeriSign *gtld* servers are essentially large directories that map domain names to IP addresses. When a person enters a Web site into their browser, a query is sent to their local DNS server for an answer. If the local DNS server cannot answer the query, it forwards it up a chain of DNS servers until it reaches one of the VeriSign *gtld* servers where the question is then answered. VeriSign has identified other directory opportunities that can utilize ATLAS for infrastructure. The following section provides two examples for illustration purposes.

## EXAMPLES OF ATLAS APPLICATIONS

### EXAMPLE #1: TELEBLOCK

#### Market Drivers

The telemarketing industry generates more than \$600 billion in revenue for U.S. companies and is growing at a rate of eight percent annually. The over six million Americans employed as telemarketers generate the two-to-three calls the average household receives each day. Many households view the increasing frequency and timing of these calls as troublesome.

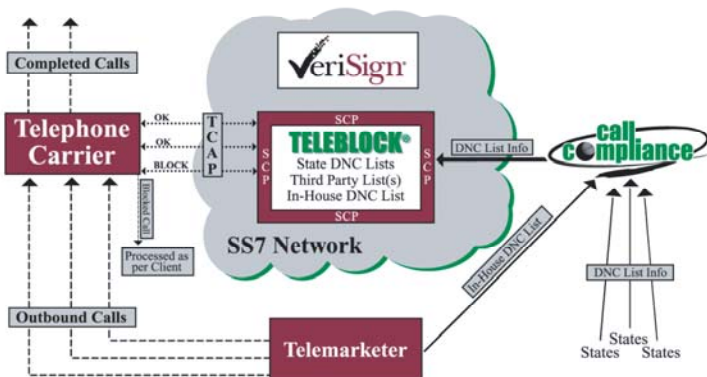
#### Problem

From the preceding statistics, it is clear that telemarketing is big business in the United States. However, the intrusive nature of telemarketing tactics has also angered many people and yielded demands for greater privacy. Currently, 25 states have passed legislation to adopt do-not-call lists and 24 more are in the process. This is in addition to the Telephone Consumer Protection Act (TCPA) of 1991, under which the FCC is administering and promulgating regulations. Under the regulations, telemarketing firms are required to maintain a centralized do-not-call list of people who do not wish to receive telephone solicitations.

#### Solution

VeriSign Telecommunication Services partnered with Call Compliance, Inc. to develop the TeleBlock service which is being offered to major carriers (AT&T, Sprint, etc.) that serve the telemarketing industry. The service is based on the ATLAS infrastructure and performs automatic blocking of do-not-call lists. A diagram of the service is shown at left.

The process works such that when a telemarketer dials a phone number, the call travels to their carrier's host switch and triggers a query to a specialized directory called a Service Control Point (SCP) that checks to see if the telephone number is on the do-not-call list. If the number is on the list, the marketer receives a message indicating that the call cannot be completed. If the telephone number is not on the list, the call is completed.



ATLAS is the infrastructure for the SCP shown in the prior diagram. As usage of the service grows, VeriSign will be able to easily scale the system to accommodate the telemarketing community's needs.

### EXAMPLE #2: DYNAMIC LOCATION AND ROUTING

#### Market Drivers

As the diversity of methods by which people communicate continues to increase, so does the challenge of knowing how and where to connect the parties (known as "presence").

Mobile phones that are capable of roaming worldwide are becoming more commonplace; Session Initiation Protocol (SIP) telephones (phones that connect using the Internet rather than conventional phone lines) are beginning to migrate from corporate campuses into general use; instant messaging (IM) is becoming an integral part of the day-to-day life of the younger generation. All of these methods require a new network fabric to be built to support the convergence of technologies.

**Problem**

The new network fabric will likely consist of an enhanced version of an SCP or some SCP analog. SCPs are the traffic cops of the telecommunications network. Just as DNS servers route end users to a Web site by providing their browsers with an IP address, so do SCPs route calls to the desired services. Since many of the new services are at least in part IP-based, a new type of SCP or directory is required. The number of queries that these new directories answer will also increase significantly. The sum of adding together IM traffic, landline traffic, and wireless traffic will place a greater burden on the new SCPs.

**Solution**

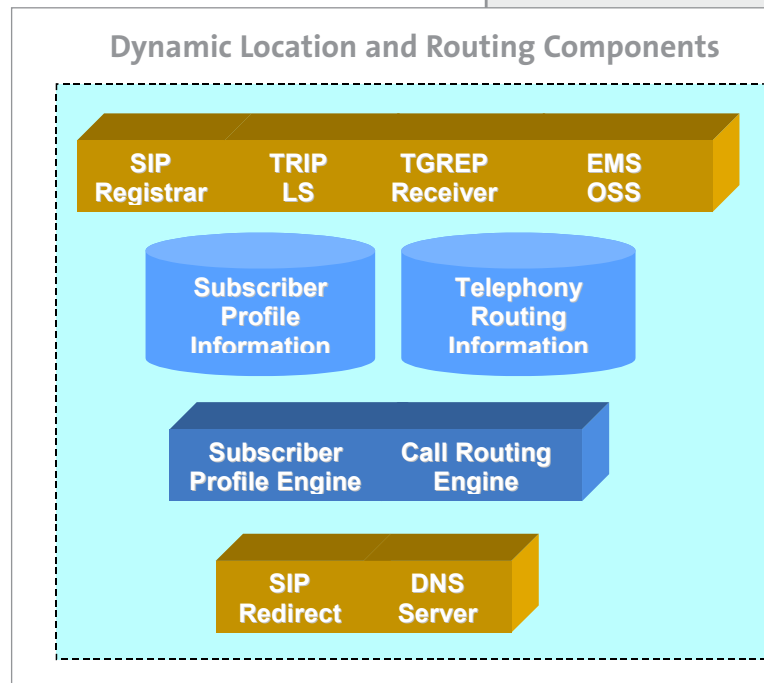
VeriSign has customized ATLAS to perform the functions of an SCP at speeds in orders of magnitude greater than the current SCPs. Standard SCPs can sustain a query rate of approximately 3,000 queries per second while an ATLAS SCP can sustain a query rate of over 50,000 queries per second. Additionally, an ATLAS SCP is capable of speaking to network devices using a variety of languages due to the number of different protocols it can speak. The real-time data synchronization feature of ATLAS also makes the next generation ATLAS SCP compelling. The following diagram shows the components of an ATLAS SCP solution.

**VISION FOR THE FUTURE**

As the world increasingly becomes an information economy, connecting fragmented islands of information is going to be crucial to achieving the promise the new economy holds for the future. Everyday, databases churn away, touching every aspect of our lives, whether it be a credit card transaction, phone call, identity authorization, etc. Locating data pertinent to a transaction in a fast, efficient secure way will be essential to maintaining an efficiently run information-based economy.

VeriSign stands at the doorway, ready to enable those transactions with its experience and expertise at managing some of the largest directories in the world. They tie together millions of Web pages through the DNS system and millions of people through VeriSign telecommunications services. Web Services, digital ID and others are around the corner. VeriSign will soon become a vital part of the fabric of the information network.

The top bar represents the various ways that a service can be provisioned into the DLR. The subscriber profile contains service subscription information indicating the services that each customer is subscribed to and how they want them delivered. The telephony routing information contains the addressing information for each service. The subscriber profile engine and call routing engine can route calls based on the service and the customer's preferences to any IP-enabled device.



## CONCLUSION

ATLAS is a very flexible, robust distributed directory platform that may have applicability in a number of industries. It is our hope that this document provides the reader with a basic understanding of how the innovative capabilities of ATLAS may apply to their industry.

If you would like to obtain more information about ATLAS or convey a new concept, please contact Brian Matthews, Vice President of Directory Services for VeriSign at [bmatthews@verisign.com](mailto:bmatthews@verisign.com).

## APPENDIX: TECHNICAL OVERVIEW OF ATLAS

### ATLAS ARCHITECTURAL ELEMENTS

ATLAS consists of three architectural elements, the data extraction process which connects to the authoritative database and extracts, validates and prepares the data for distribution; The data distribution process that insures that the data in the authoritative database is faithfully reproduced in the distributed look-up engines; and the distributed look-up engines that maintain local copies of the data to provide fast, efficient retrieval of the information. The following paragraphs discuss each element in greater detail.

#### Data Extraction Process

The authoritative database is a centralized database that contains all of the information the customer wishes to distribute to its remote sites. This database is an Online Transaction Processing database that may be built on Oracle, DB2, SQL Server, or any database that has the capability of uniquely identifying the commit order of all transactions within the database.

The Data Extraction process is responsible for extracting changes made to the central database as they occur, formatting the output into the Send File format, and validating the data in preparation for distributing to the look-up sites.

#### Data Distribution Process

The data distribution mechanism is highly reliable, secure, and operationally fault-tolerant and supports continuous incremental update of the look-up engines from the authoritative database. This simple and robust allows data to be updated at a high rate and in large amounts.

#### Look-up Engines

The look-up engines are read-only database sites that are geographically distributed based on the customers needs and that support extreme query scalability and various types of data structures. These read-only sites are highly optimized for search performance while simultaneously providing support for high volume incremental updates from the authoritative database.

### DATA EXTRACTION PROCESS

As changes are made to the central database, the Data Extraction Process reads the changes and records them in "sendfiles." Sendfiles are text files that contain a header, a checksum, and a sequence of commit-ordered add, modify or delete transactions. Sendfiles are generated every 15 seconds or so, and are identified by monotonically increasing serial numbers. They are ultimately sent to the look-up engines, which apply the changes to their copy of the database.

In the case of DNS, an extraction daemon creates the sendfiles by recognizing when updates are made to the authoritative hub's transaction table. The extraction daemon periodically scans that table for new transactions of an appropriate type. To create a valid sequence of transactions, the extraction daemon needs to properly order the transactions—an update to a domain cannot precede the addition of the domain. This ordering is based on an Oracle feature called the System Change Number (SCN), an integer that records the commit order of all transactions in the database.

As mentioned earlier, ATLAS can work with any database that provides a method to determine the order in which transactions are committed to the database.

### **Initial Send Files (ISFs)**

An initial sendfile is a text file that represents the complete state of the authoritative database at a given point in time. Since send files only record changes to the authoritative database, initial sendfiles must be sent to the look-up engines to initialize or re-initialize the engine. ISFs share the sendfile format, but contain only "add" operations. Initial sendfiles are created daily but the look-up engines are restarted on fresh ISFs only once a week. ISFs are created daily to ensure that any failure of the sendfile stream would result in no worse than 24-hour latency. ATLAS customers have the flexibility of determining how frequently they wish to distribute ISFs.

Once the ISF is distributed, the ISF notes the most recent transaction that is included in the ISF, along with the sendfile that contains that transaction. Given those two pointers, a process can determine where exactly in the sendfile stream it should begin applying changes.

## **DATA VALIDATION**

Before updates are distributed to the look-up engines (LUE), we must ensure that the changes are applied correctly and that they will not cause operational or data integrity issues. The validation process seeks to duplicate the operational environment of the look-up engines as closely as possible using the same hardware, same software, and same procedures. In fact, the data validators are essentially captive look-up engines with additional verification capabilities.

The primary validator mimics the look-up engine at the remote sites, and is therefore always running on the same ISF, refreshed at the same frequency as the remote LUEs. Since a mechanical failure of the validator would stop the update process, we also run a backup validator over the same ISF. The system also generates daily ISFs for emergency recovery purposes, so we ensure that all sendfiles are correctly applied to the daily ISF, in addition to the constellation ISF. This is the job of the third validator. Finally, a fourth and final validator is used as a backup.

### **The Validation Process**

The validation coordinator monitors the status of the four validators and communicates with the validators through the authoritative database. Validators record their successes and failures in the database, which the validation coordinator monitors. When a validator starts, it first validates the ISF itself. This involves loading the ISF into an in-memory database, then performs a line-by-line comparison of each entry currently resident in the in-memory database to its counterpart in the central database.

Discrepancies can occur because the validation process for an ISF may end many hours after the process started depending on the size of the data in the authoritative hub. It may take several hours to generate the ISF, additional time to move and load it, and several more hours to validate every entry. In the meantime, thousands of sendfiles may have been generated, each noting changes to the database described by the ISF. Thus discrepancies are expected, and indeed can be numerous. The task of the validator is to ensure that they

are benign—that any differences it finds are accounted for in sendfiles it has not yet processed. If any unaccountable discrepancies are identified, the ISF is not approved.

Once the ISF has been approved, the validator begins processing sendfiles as it receives them. To minimize the number of discrepancies, the validator processes "chunks" of sendfiles, from the oldest sendfile not yet processed to the most recent sendfile available. It applies all the sendfiles to the in-memory database, validates them, then approves or fails the entire chunk at once. If the validator falls behind (as it does when validating an ISF, for example), the number of sendfiles in a chunk will be large. Once it has caught up, however, chunks will normally consist of a single sendfile.

Note that the validator does not perform the ISF validation process for each sendfile - that would take far too long. Instead, it verifies only that the domains and lookup engine referenced in the sendfiles are correct.

### **ATLAS Look-up Engine**

The ATLAS look-up engine is designed to provide extremely high response rates, scaling to enormous numbers of requests. This is made possible by two key features: a high speed, shared memory database, and a two-tiered architecture that lowers the number of packets the look-up machine receives. The protocol engines receive the look-up requests and aggregate those into a "SuperPacket" and farm off the request to the look-up engine. The following paragraphs describe each component of the look-up engine and the protocol through which the components communicate.

#### **Protocol Engines (PEs)**

To maximize the amount of work the look-up engine can process, a bank of front-end "protocol engines" (PE) actually receive the look-up requests, validate them, and combine several small messages into one larger "SuperPacket" message (described later in this document). This packaging cuts down the number of network interrupts experienced by the look-up engine, leaving more CPU cycles free to process requests.

The farms of protocol engines handle the nuances of the wire-line protocol, immediately responding to invalid look-ups and multiplexing valid look-ups to the look-up engines. After receiving the query results from the look-up engines, the protocol engines de-multiplex the raw responses along with any data required to return the response (i.e. original IP address and port number) that were packed with the original multiplexed queries as opaque data. The PEs construct and return valid wire-line responses.

#### **Look-up Engine Subcomponent**

The shared memory database is a fully transactional database, but requires no OS-level locks during updates. As sendfiles arrive at the look-up engine, they are processed by an update daemon, which applies the changes to the shared memory database. If the look-up engine were to refer to one of the records being updated, it would find the record marked "dirty" and restart the query.

The back-end look-up engines are powerful, large-memory servers that store the entire lookup record set in-memory. By virtue of "SuperPacket protocol" multiplexing, each LUE spends much less of its CPU capacity on network transport processing, which dramatically increases the query throughput of each machine. Since all the data is held in memory, each

LUE is identical and can service any request—no complicated load splitting algorithm is required with its associated data management headaches.

Note that the look-up engines have a read-only view of the shared memory database—if they crash, the database remains intact. The look-up engines simply restart and reconnect.

#### **SuperPacket Protocol**

A SuperPacket is a collection of look-up requests or responses. PE's combine several small look-up messages into one larger SuperPacket message so that the look-up engine only has to process a single network packet. The same process occurs, in reverse, for the responses.

The "SuperPacket" protocol is used to transmit multiplexed requests to the look-up engines from the PEs to the LUEs and the LUEs send multiplexed responses back. The SuperPacket protocol is optimized for easy parsing of request and response data. Apart from the core function of multiplexing requests and responses, a critical part of the SuperPacket protocol is preserving opaque data originally inserted by the LUE in the response back to the PE. This allows for all PEs to be stateless, greatly simplifying the protocol engine software implementation.

#### **Minimum Unit (MU)**

Protocol engines are combined with look-up engines, networking gear, and monitoring equipment to form a Minimum Unit, or "MU." An MU is a balanced, functionally complete set of equipment that can be replicated for additional performance—as the name implies, an MU is the minimum unit of scalability. The MU specification includes everything from specific product choices to wiring and rack layout diagrams.

An MU includes two look-up machines, each of which has a dedicated bank of PEs feeding it requests. To simplify the behavior and recovery of the system, PEs do not fail over to the other look-up engine if they lose contact with their own look-up engine. Instead, the PEs signal the load balancer that they are unable to process queries, and load is transferred to the remaining PEs.

The divide-and-conquer approach of separately optimizing the hardware architectures for network transport and look-up, linked together by the multiplexing SuperPacket protocol, results in the lowest cost configuration that absolutely minimizes the number of identical, stateful servers. Though the PEs are small and numerous to minimize cost, each is identical and stateless, as close to a disposable server as possible to simplify administration. Higher-powered servers can be used for PEs to reduce box count if administration of so many small machines proves more challenging than currently envisioned.

#### **Data Distribution**

Once the validation process has approved a sendfile or ISF, it must be distributed to the remote sites. This can be particularly challenging depending on the geographical locations of the sites due to the possibility of poor quality network connections between the central hub and overseas sites. These network issues tend to have a particularly bad effect on TCP transfers, since TCP backs off the sending rate in response to errors.

To improve the situation, files are transferred through a UDP-based protocol that transmits

at fixed rates, regardless of the rate of acknowledgements. This technique appears to improve long-haul transmission rates dramatically.

Distribution is the job of two programs, the sender and the receiver. The sender contains many threads, each of which transmits files from a single queue to a specific destination address and port. That destination address and port is serviced by a receiver, which also has multiple threads. Each sender thread links to a single receiver thread, and vice-versa. In each sender-to-receiver link, files are transferred one at a time, in order, sequentially.

At the central distribution server, the sender has a single thread per remote node receiving sendfiles. It also has a second thread per node receiving ISFs—the two streams are separated to ensure that the small sendfiles won't have to wait behind a large ISF.

ISFs are very large, but they may change relatively little day-to-day. When transmitting ISFs, therefore, the distribution system uses the `xdelta` utility to calculate the difference between the current ISF and the previous one, and only transmits the difference. The receiver reverses the process, applying the difference to the previous ISF received to produce the new file.

The sender and receiver are used whenever sendfiles have to be transferred between machines, anywhere in the system—even within the same data center.

#### **Update Process**

The update processes are responsible for creating and maintaining the shared memory database. To create a shared memory database, operators load an ISF using a utility called `isfLoader`. The ISF loader is responsible for applying all the actions listed in the ISF, then running through the sendfiles until it locates the last transaction represented in the ISF. Having done that, it applies all the changes present in that sendfile after the transaction, and marks the database up-to-date with respect to the given sendfile. Thus it leaves the database on a sendfile boundary.

Once the database has been created, it is the responsibility of the update daemon to keep it current. As new sendfiles arrive, the update daemon applies the changes to the database, and marks the database as up-to-date with that sendfile. If the update daemon experiences any problems processing the sendfile, it rolls back all of the actions from that sendfile. Thus for purposes of recovery, the sendfile is the minimal transactional unit. In the normal course of operation, however, individual transactions within the sendfile are the minimal transactional unit.

The update daemon is designed such that it never leaves the shared memory database in a corrupted state, no matter how or when it fails. All memory modifications are logged before and after the fact to ensure that, should the daemon fail, it can always roll back any changes when it restarts.

#### **Checkpoints**

Checkpoints are binary images of the shared memory database that are periodically written to disk by the checkpoint daemon. Checkpoints are another method for initializing the shared memory database.

Checkpoints are typically used to quickly re-initialize the look-up engine database. Re-initializing with the ISF loader requires processing the ISF and then having the update daemon process all the sendfiles received since the ISF was generated. Since sendfiles are generated every 15 seconds, and ISFs are typically generated weekly, that means you might have as many as 40,000 sendfiles to

work through before becoming current. Initializing the database from a checkpoint is typically much faster than the ISF process.

### **Monitoring**

To enable continuous monitoring of the system, all ATLAS servers send periodic status messages to a designated address. These messages identify the server by machine, server type, process ID and name, and also contain some statistics relevant to the particular type of server.

In addition, each ATLAS node includes a number of traffic monitoring processes, which scan packets as they enter and leave the network, and characterize the number of look-ups and their content. These network monitors transmit summaries of their findings to a designated address.

The status messages are sent to one or more repeaters, which store pertinent facts in a database and provide digests or complete feeds of the information to subscribing clients. One notable client is the Heads-Up Display, which provides a graphical depiction of the real-time status of ATLAS components.

