hp zero latency enterprise

a white paper from hp

zero latency enterprise architecture

Latency, or the inability to react *immediately* to business stimuli, is at the root of practically every business misadventure—from poor customer service to missed selling opportunities to consumer fraud. The solution? *Zero latency enterprise*, or ZLE. A zero latency enterprise is a company that has removed latency from its operations so that business events that occur anywhere in the organization can immediately trigger appropriate actions across all other parts of the enterprise and beyond. Operational inconsistencies are eliminated, and users gain real-time access to real-time consolidated information. Bottom line: the entire enterprise becomes more responsive and competitive.

The challenge lies in becoming a zero latency enterprise. A business must somehow integrate enterprise data, synchronize it, and route that information across the enterprise—all in real time. Not too long ago, this was an impossible order to fill. Now it's not only feasible, but companies are achieving it in multiple industries.

This white paper introduces the concept of the zero latency enterprise, discusses what it means to be a zero latency enterprise, and describes the architectural challenges involved. It concludes with a description of the extensible architecture that HP provides to enable companies to *see* their business and *know* their customers as they are right now and *act* on that information without delay to achieve a competitive advantage.

ZLE: the vision

Gartner was among the first to crystallize a vision of the zero latency enterprise as the "instantaneous awareness and appropriate response to events across an entire enterprise." In this vision, as one system in the enterprise tracks a business event, other systems are alerted in real time that the event has occurred so that appropriate and timely actions can take place.

The effects of instantaneous awareness and response are profound and global—touching on customers, suppliers and trading partners, as well as a company's overall competitiveness and profitability.

achieving up-to-the-second business views

ZLE technology employs a technique opposite to that of batch technology to integrate systems. Instead of waiting for daily updates, it updates and synchronizes systems and people in real time. In a zero latency customer relationship management (CRM) environment, for example, that means that when a customer browses a company's

website and then contacts that company's call center, the customer service representative has all the information necessary to make meaningful recommendations. He or she has access to a *complete*, *up-to-the-second* view of the customer. The clickstream information gleaned from the website visit has already been synthesized and made available. Key personnel can even be notified should the customer have a poor experience.

leveraging real-time integration across the enterprise

Another example scenario for zero latency operations is a supply chain management environment. Here the challenge is to integrate multiple internal and external systems that are often based on dissimilar platforms running different enterprise resource planning (ERP) software. ZLE technology provides a standards-based way to integrate systems and processes across and beyond the enterprise in real time. Thus, planners gain a consolidated, up-to-the-second view of orders, schedules, and the movement of products inside and outside of the company. And orders are fulfilled faster and more reliably with information integrated from suppliers, shippers, and receivers.

Because receipt settlements are evaluated more rapidly, order-to-cash cycles pick up speed. Furthermore, customer service improves with shorter lead times and the ability to offer more customized products and services.

rationalizing systems and processes

The ZLE environment allows systems in a merger or acquisition scenario to be integrated in real time and rationalized. Rather than force an immediate and wholesale consolidation of duplicate systems and processes, companies can integrate duplicate systems in their current form by placing a ZLE architecture on top of them. So, in a merger between company A and company B, systems can share a common view of the total enterprise.

In order entry systems, for example, as an order occurs in one organization's system, it can be reflected in the systems of the other. The enterprise benefits immediately from systems integration, allowing consolidation on a single system to proceed at an intelligent pace.

becoming a zero latency enterprise

The examples in this section represent just a few applications of the zero latency enterprise. Today it is entirely feasible for a company to become a zero latency enterprise in some—or almost all—of its operations. Indeed, the benefits of a ZLE environment are waiting to be reaped:

- Instant personalized marketing
- Reduced exposure to fraud, attrition, and other risks
- Faster time to market
- New applications that leverage real-time information from across the enterprise

The next section reviews the techniques that can be used to build a ZLE environment.

ZLE: the challenge

Zero latency enterprise is not about single products or monolithic solutions. Rather, it is about real-time enterprisewide integration of data and processes. This real-time integration is both the overriding goal of ZLE and its greatest challenge.

Of the two types of integration that exist in the IT world—consolidation of data and integration of processes—consolidation of data is the easier task. In fact, data warehousing initiatives have been used successfully for years to achieve this goal. Integrating processes and the applications that drive them, on the other hand, is considerably more complex. Consequently, it is not realistic to set a goal of integrating all data and all processes. The objective, rather, is to integrate systems so that they coexist gracefully yet still operate independently.

traditional application integration techniques

Traditionally, there have been three primary approaches to integrating systems and applications:

- Batch technology
- Transactional technology
- Messaging technology

batch technology

Batch technology is the most basic and simple way to achieve integration. Many existing systems are batch-oriented and can only be integrated with other systems through batch updates. Indeed, batch must be taken into account in any ZLE environment given the ubiquity of batch systems. Real time it is not; nevertheless, batch is attractive because it is easy to recover to a point in time.

transactional technology

Transactional technology provides a second integration technique, enabling applications to interact directly with one another on a transaction-by-transaction basis through a common set of methods and protocols. These common protocols, which are well defined, include J2EE, Tuxedo, CORBA, and Java™.

Transactional integration is based on a request/response model, in which one application can initiate a transaction with another and then receive a response based on whatever was executed. This model is attractive in that it enables transaction boundaries to be enforced. And it can coordinate transactions across systems, ensuring that each system completes its transaction(s) or not.

The transaction model is mature and well suited to high-volume, integrated environments. It leverages transaction processing monitors and application servers, which are optimized to handle numerous users and transactions in set time periods.

messaging technology

The third approach to integration is through messaging technologies. A proliferation of prebuilt application software has spurred the development of enterprise application integration (EAI) toolsets for noninvasive, standards-based integration. In fact, use of EAI has grown coincident with the overall enterprise computing market.

EAI toolsets use a publish/subscribe model to propagate messages asynchronously between applications. Like e-mail, publish/subscribe is a "fire and forget" messaging model. A job is considered finished once the message is delivered. Unlike the transactional approach of request/response, there is no waiting to see if a transaction is completed or not.

EAI toolsets provide a variety of essential functions, including

- Capturing events
- Transforming data in proprietary formats into a standard message format using wrappers or adapters[1]
- Routing the message to a message hub and on to the appropriate subscribing applications
- Transforming the data (that is, applying it) to a native format on the receiving end

Many EAI toolsets have higher-level functions as well that include business process flow and business-to-business (B2B) types of services. Business process flow capabilities provide a *business process view* of how information moves between systems, as opposed to a data flow or *message view*. Users can even develop a graphical picture of their business process. This view allows users to do things that a simple message-oriented view does not. For example, several days into a long-running transaction, the business process flow engine can initiate a follow-up activity, such as "check the progress of the order" while the transaction is still in flight.

EAI B2B functions support commerce between enterprises. These include security services, logging and auditing, management of trading partners, and other functions to extend integration outside the enterprise firewall.

why traditional techniques don't work effectively
The problem with these three approaches to integration is that they do not address data integration effectively. Data integration cannot be resolved simply by passing messages or integrating transactions. Data contains semantics and attributes, such as relationships and hierarchies, that must be merged to achieve a consolidated view. Hence, it must be materialized and maintained in a common and consistent format, which is beyond the scope of traditional synchronous messaging and transactional integration.

warehousing approach to data integration

The most common model for data integration is the data warehouse, which has a database designed for specific needs. An abundance of tools support the model, which is typically driven by metadata to handle changes rapidly. So, designing a new data model and loading data into it to achieve an integrated, subject-oriented view is relatively straightforward.

Though a proven and effective model for business intelligence and other tasks, the data warehouse does not fully address the challenge of real-time integration in a ZLE context. The problem with the data warehouse construct from a ZLE standpoint is that it is historically focused on helping companies understand and analyze their businesses. Thus, a data warehouse provides a long-term, integrated view, but not the up-to-the-second operational view required for ZLE.

[1] Adapters have always played a central role in EAI. Java connectors will be useful for new applications, as they will relieve the need for adapters. Yet, in the real world with its myriad existing applications, adapters will continue to be leveraged extensively for EAI.

Further, while data warehousing solves the data integration problem, up to a point, it does nothing to address the issues associated with integrating applications. It is not possible to interface applications through the data warehouse, nor is it feasible to build new applications on top of it.

Other deficiencies of the data warehouse model in a zero latency environment include

- *High latency:* Data is typically appended in batch mode at the end of a processing cycle.
- *Lack of atomic data:* Data warehouses contain detailed data that is typically summarized; the transaction-level data needed in a ZLE environment is lacking.
- *Optimized for complex queries, not high-volume user access*: A data warehouse is perfect for handling large, complex queries against aggregated sets of data, but not for large-scale user access (that is, it shows trends but not everything about a customer up to the second).
- *Unidirectional:* A data warehouse is a repository for information extracted from other systems, but it is not appropriate for propagating change to other operational systems (that is, it is an information "sink," not an active information broker).

when integration becomes the project

When building new enterprise solutions, most companies discover that integration becomes the project. Enterprise environments are constantly evolving as systems are added and subtracted. In large enterprises, therefore, it is not unusual for 80 percent of development efforts to go into the effort of integration, leaving just 20 percent for actually developing new services and capabilities. The ZLE architecture uses a hub approach that can substantially ease integration efforts.

EAI approach to data integration

Data warehousing is not the only model for data integration. The data integration shortcomings of transactional and messaging technologies aside, EAI technology can provide a "virtual" data integration approach based on either a request/response or a publish/subscribe model.

request/response model

With a request/response model, a system can remotely access operational data on another system. The remote system reaches in using synchronous or asynchronous communications and retrieves the data it needs. In this way, a remote point of sale (POS) system can request and receive a credit card authorization. The problem is that the accessed system has to stay online constantly, or the environment breaks down. Similarly, the accessed system has to have processor cycles available to process and respond to all of the incoming requests. As a result, the request/response model necessitates a tightly coupled integration that enforces dependencies between systems.

In addition, issues may arise concerning different service levels among systems that need to communicate. Older systems were designed commonly for large batch windows, while most new systems are not. This inherent mismatch can be the source of numerous problems.

publish/subscribe model

With a publish/subscribe approach, data can be pushed to all interested parties. The results of a transaction, such as "create a new order," can be published to any number of pertinent systems. The problem with this approach, however, is that all systems need to keep copies of all relevant data. Without a centralized repository or message store (which is not implicit in this approach), the synchronization problem and associated overhead are immense. And as new services are built on top of such an environment, data replication and redundancy only grow.

To compound the problem, some types of information are voluminous, such as call detail records. These records, as well as other types of transactional data, do not lend themselves to multiple copies.

Consequently, while data integration using EAI works on a small to medium scale, it is impractical in a large-scale environment that has to publish all the necessary data to all interested parties. It is difficult to build on this model, and you cannot count on it in a large-scale transaction environment that requires subsecond response times. Further, there is no message archiving, reporting, or auditing. Without a message store, messages are published and then forgotten.

an inclusive approach: the hp ZLE framework

A zero latency enterprise depends on comprehensive and efficient data *and* application integration. As discussed, traditional methods to achieve this integration, including data warehousing and EAI, only deliver parts of a solution. Zero latency operations require a new type of unifying architecture for integrating, synchronizing, routing, caching, and transacting in real time—one that rests on top of existing system environments and pulls them together in a noninvasive, standards-based fashion.

HP has developed such an architecture and a set of supporting tools. The HP ZLE framework is a data-oriented architecture that centers on a real-time ZLE data store. A new construct, the ZLE data store is a "hot cache" of data from across the enterprise. Rather than using EAI toolsets to keep applications synchronized, the ZLE framework uses them and other technologies to keep remote applications and their data in sync with the ZLE data store. And it uses the ZLE data store to keep remote applications in the ZLE environment synchronized with one another.

The utility of the ZLE data store is that it delivers, at any given time, a single, consolidated view of all pertinent data from all applications in the enterprise. Data is redundant between each application and the ZLE data store, but not between all applications. At the same time, the structure of the data in the ZLE data store is changed so that it is subject-oriented, persistent, and can be updated in real time.

The following section describes how, with the ZLE framework, HP and its ZLE independent software vendor (ISV) partners provide an end-to-end extensible architecture that

- Integrates data and business processes from across the enterprise in real time to support real-time actions
- Provides a rule-driven environment for flow of information
- Brings new scalability, availability, and independent performance to enterprise functions
- Enables new types of applications, based on real-time data and application integration

Rather than using EAI toolsets to keep applications synchronized, the ZLE framework uses them and other technologies to keep remote applications and their data in sync with the ZLE data store. And it uses the ZLE data store to keep remote applications in the ZLE environment synchronized with one another.

hp ZLE framework

The HP ZLE framework is a multilevel architecture centered on a virtual hub, called the *ZLE core*, that caches and routes information. Within the ZLE core resides a persistent *ZLE data store* and a variety of technologies to keep the data store in sync with all the systems it integrates.

The ZLE core represents a state-of-the-art EAI environment. As shown in figure 1, *enterprise applications* such as SAP, Siebel, and other enterprise systems attach to the core via a variety of application and technology *adapters*. These adapters convert proprietary messages to the standards used by the ZLE core.

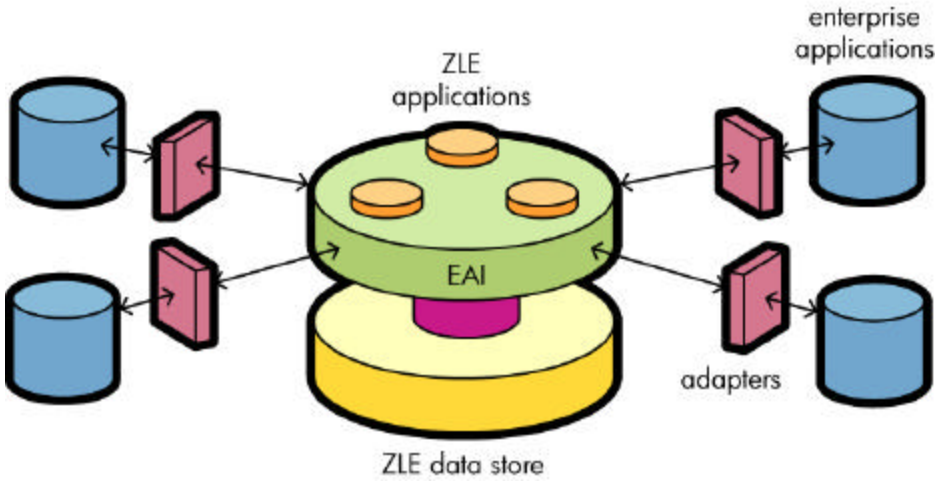

figure 1. the ZLE framework combines the strengths of a ZLE data store and enterprise application integration (EAI) to create a seamless, enterprisewide architecture for real-time information

The ZLE data store caches the information that runs in and out of the core, maintains and enriches it, and acts as an intelligent message store and queue to manage delivery of information to the appropriate applications. Data in the ZLE data store can be mined and used for business intelligence purposes, either through bulk extracts to data mining tools and data marts or by direct queries against the data store. The data mining and other knowledge discovery applications that pull data from the data store can also contribute models back to the ZLE core, where they can be leveraged to direct business processes.

Finally, specialized *ZLE applications*, such as personalization and fraud management, reside on top of the ZLE core to directly access its real-time data and leverage its real-time integration and other services.

As an operational solution with real-time, all-the-time integration as its primary goal, the ZLE framework depends on platform robustness and open standards. For that reason, the ZLE core's services and capabilities run on the HP NonStop™ server, a platform proven to provide the parallel performance, scalability, and availability requisite for business-critical applications. A framework based on CORBA and Java or Tuxedo technologies is available; the CORBA framework can interface to Tuxedo applications and vice versa. Companies can select the framework with which they are most familiar and comfortable.

defining the ZLE data store

The physical and logical center of the ZLE core is the ZLE data store, which is a new construct. Based on an operational data store (ODS) type of model, it is essentially a database front end for new and existing applications attached to the ZLE core (see figure 2).

A traditional data warehouse is a back-end database built for after-the-fact analysis purposes. In contrast, the ZLE data store resides logically in front of the other applications in the ZLE environment that use the data store as the basis for their real-time integration. In this role, the ZLE data store functions as a

- Hot enterprise cache
- State engine
- Real-time data warehouse
- Relational message store

The ZLE data store resides logically in front of the other applications in the ZLE environment that use the data store as the basis for their real-time integration.

hot enterprise cache

By caching data in the ZLE data store, the ZLE framework enables applications and systems to be loosely coupled. This is an advantage over closely integrated systems, which are limited in terms of scalability and availability (that is, they often cannot function without each other or scale independently). For example, an e-store may need to access a credit system to check a customer rating before completing a transaction. Caching data in a central repository loosens application bonds without disrupting integration. There is "breathing room," so that faults in one application don't cascade over the entire architecture, and each system can scale at its own pace.

As a hot enterprise cache, the ZLE data store must be able to update operational data continuously in real time. For this reason, continuous database availability is essential. The ZLE core uses the NonStop SQL database, which currently underpins most of the world's stock exchanges, the majority of POS and ATM networks, and other critical large-scale applications that cannot afford even a second of downtime. Designed as an extension of the NonStop platform, the NonStop SQL database inherits that platform's massive linear scalability, parallelism, and continuous availability.

The NonStop SQL database is also designed for online manageability, including performing updates and reorganizations online. For example, maintenance can be performed on a table while it is being read, written to, and updated.

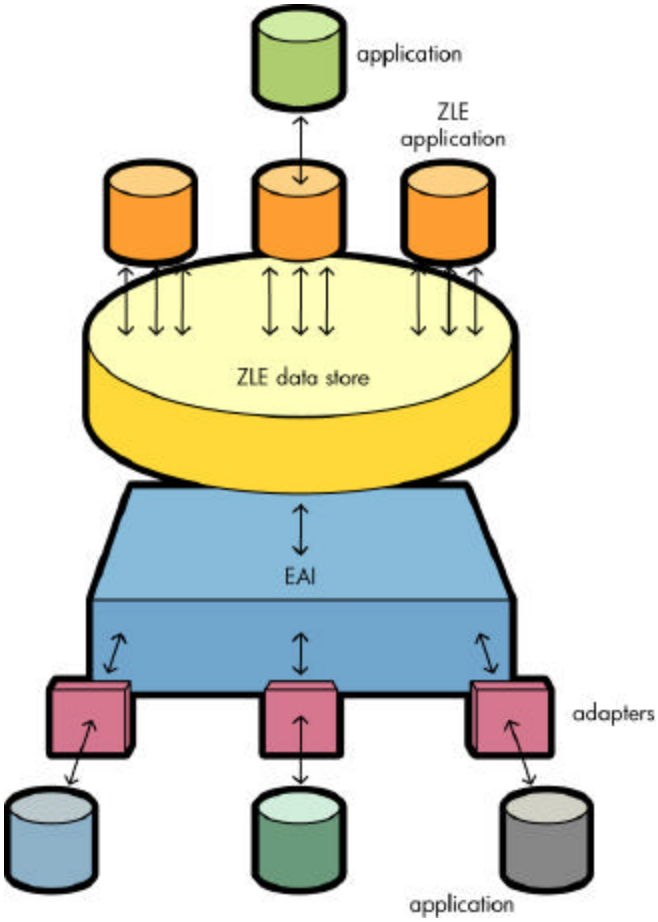

figure 2. the ZLE data store is a database front end for caching and distributing real-time data as well as a platform for new types of ZLE applications

state engine

The second aspect of the ZLE data store is that it represents the business state of the enterprise—it is a single place to go to see the state of the company at a given moment. One of the standard functions of a database in an online transaction processing (OLTP) environment is to act as a state engine to keep track of long-running transactions. The ZLE data store does this, but even more significant, it expresses the state of the business itself.

With this single, consolidated database in place, new ZLE applications can be developed that reside on top of the data store and leverage that current state of the business view. These applications can include new types of applications for fraud management, customer relationship management (CRM), and more.

It is important to note that because this business state information is valuable to many different systems, the database, by necessity, becomes a hotspot of activity, with all different kinds of activity taking place simultaneously. State data can be

- Updated as transactional systems send updates to the data store
- Accessed by customer service applications
- Queried and extracted to other operational and analytic systems at the same time

It is imperative that large extractions and queries do not affect customer service access and that all three of these activities in turn do not affect updates. So, the database management system must possess a robust mixed-workload capability. In this regard, the NonStop SQL database has proven itself, including the ability to perform reads/writes on widely varying priority levels while protecting workloads from one another. The key lies in the software's ability to manage diverse workloads so that they do not affect one another.

real-time data warehouse

The ZLE data store can also function as a real-time data warehouse. Sometimes, it can replace an enterprise data warehouse (but not data marts). Or it can coexist with a data warehouse. In any event, the ZLE data store is well suited to feeding downstream data marts with subsets of its real-time data.

The ZLE data store can function as a real-time data warehouse largely because the NonStop SQL database resolves the differing needs of OLTP and business intelligence. For example, indexing is addressed by a new technique called *MultiDimensional Access Method (MDAM)*. Through the use of MDAM, the NonStop SQL database allows a single database to support business intelligence (which uses many indexes) and OLTP (which does not) concurrently. The NonStop SQL database also successfully addresses the need for different aggregation and service levels.

relational message store

Finally, the ZLE data store functions as a relational message store. A message store is an essential element in a publish/subscribe environment. As messages move through the ZLE core, they are maintained in the relational database that underpins the ZLE data store. Each message—whether it represents a POS transaction, a call detail record, or a credit or debit transaction—is stored and managed in a single, centralized place from which it can be routed to multiple subscriber systems.

A publish/subscribe feature built into the NonStop SQL database enables subscriptions to be created in SQL as select statements, with the output streamed to the subscribing application. You can install subscriptions into tables and wait for events to occur.

Through the NonStop SQL database, publish/subscribe enables the messaging function to leverage the database platform's parallelism, partitioning, and built-in manageability to ensure an efficient flow of subscriber messages. Priority, first-in/first-out, guaranteed, and once-and-only-once delivery is supported. In addition, the publish/subscribe feature is in the disk process and is thus extremely high performance.

While other publish/subscribe products on the market contain a data store for their message queues, they are not relational and do not allow the end user to access these data stores through standard SQL syntax.

breaking the reliability dependencies

An important advantage of the ZLE data store construct is that dependencies between requesting and responding applications are largely severed—the responding application can go down, but current or near-current information is still available in the ZLE data store. Customers can still get answers, supply chain applications can stay current, and so forth. At a time when customer-facing applications like call centers and Web self-service applications strive for

24 x 7 operations, they can achieve this availability without depending on the production systems that feed them data. The way it works is that changes made to the data are applied to the ZLE data store and stacked on reliable queues to wait for the responding system to become available.

exploiting the ZLE data store

There are a number of ways in which the ZLE data store can be exploited to add value to the ZLE architecture.

return on cache—adding value to EAI

One key advantage of the ZLE data store is the way in which it can be used to control and reconcile messages as they move through the ZLE core. This can be done in two ways:

- Lookup and enrichment of messages
- Breaking direct links between systems

lookup and enrichment of messages

Messages can be controlled and reconciled by lookup and enrichment of messages and data as they flow between systems via the ZLE data store. For example, an inventory transaction originating in system A and propagated to system B may require a cross-reference to show how items map to each other. That cross-reference information can be cached in the data store and appended as needed. Another message may require a customer identifier, which can also be kept in the data store to enrich messages as they flow through.

Another example might involve a B2B customer looking to make an online purchase. As the ZLE architecture pulls together current inventory and pricing information, it can enrich it with personalized customer-specific data from the ZLE data store regarding special offers on related products—information that is invisible to the inventory system.

breaking direct links between systems

The second added value is how the ZLE data store, acting as a cache for persistent information, breaks the direct links between systems. The ZLE data store replicates data *once*—in a single repository that is shared by all other applications. Consequently, all applications have a single, consistent place to go for information, removing the need to keep duplicate data on multiple remote systems and to contact individual production systems for this information. For example, as items change in a system, the data store publishes the updates. Lookups are directed to the data store, not to the originating systems, which keeps them from being slowed down by repeated requests. Result: minimal latency and reduced system overhead.

It is important to note that since transaction data doesn't typically get changed, it can all reside in one place. State data, however, has to go out to everyone. Siebel, for example, has its own database that keeps demographic information and so on. But does the database have to contain all the transactions? No. A Siebel application can go to the ZLE data store, which keeps everyone's copy of state data in sync, to access transaction data.

intelligently reengineering and controlling redundancy

Another key advantage of the ZLE data store concept is its support of intelligent reengineering efforts. In managing transitions resulting from mergers, platform migration, and system consolidation, the challenge is often one of controlling redundant data. This is where ZLE integration excels.

In a transition scenario, the ZLE data store can serve as a control agent by acting as a platform for integrated views accessible by different systems. With a common data model enforced across different systems, companies can migrate at a managed pace.

In addition to a consolidating mechanism, the ZLE data store can function as a new system of record for the multiple systems integrated via the ZLE framework. Hence, new integrated functions can be built on top of the integration architecture—including functions that extend existing investments.

driving real-time analytics

The ZLE data store is operational by nature. Nevertheless, it can provide data to data marts and other downstream entities for aggregation and analysis. Thus, knowledge discovery applications like OLAP and data mining programs can derive significant value from the ZLE data store.

The relationship between the ZLE data store and downstream analytic applications is highly symbiotic. The ZLE data store enables data mining, OLAP, and other analytic programs to access and analyze real-time, current state information as well as historical information. Yet, even as bulk data is derived and delivered from the data store to analytic applications, these applications can subsequently feed back models and information gleaned from the data that can then be leveraged in real time by the ZLE core and applications (see figure 3). For example, a model that extrapolates a particular group of customers' purchasing likelihood based on past buying habits and recent purchases can be leveraged by the ZLE core's scoring service. In this way, customer interactions can be scored in real time or by a specialized personalization application.

It is also possible to query the ZLE data store directly through an open interface such as Open Database Connectivity (ODBC) or Java Database Connectivity (JDBC). The mixed-workload capability of the NonStop SQL database keeps such queries from affecting the performance of core ZLE tasks.

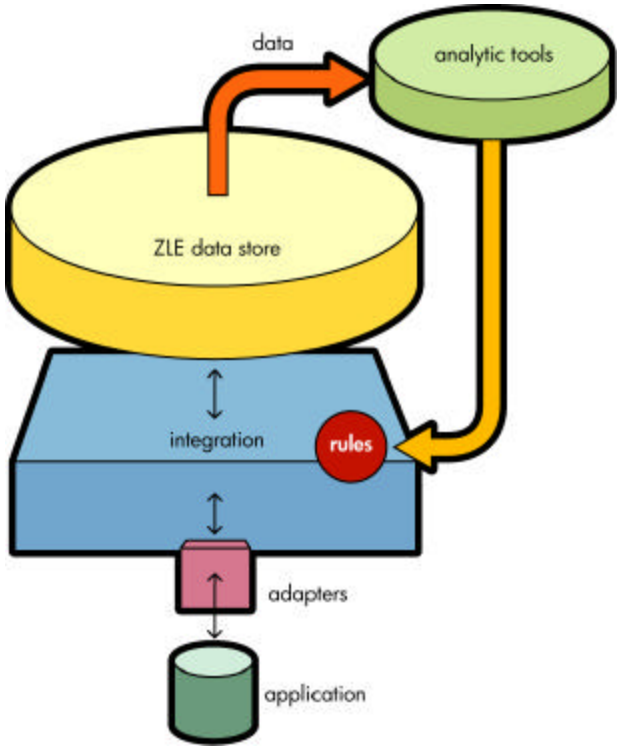

figure 3. the ZLE data store enables historical analysis and the application of rules and learned knowledge to the present environment

enabling unique ZLE applications

Another very powerful use of the ZLE data store is as a platform for launching unique ZLE applications. These applications interface directly with the ZLE data store and build on its real-time, consolidated data to provide enterprise services that otherwise would not be possible.

Unlike enterprise applications, which are standalone applications with their own database such as Siebel, ZLE applications are specialized programs that provide business-level functions. They require the substructure of the ZLE core, with its data store and services, in order to work. Functionally, ZLE applications "customize" the core for specific purposes.

ZLE applications are able to draw information from the data store and make updates to it, thus spreading their value throughout the enterprise. A perfect example of a ZLE application is a personalization application that drives recommendations out to customer touch points based on up-to-the-second views of each customer. Because of the real-time nature of the ZLE data store, personalization can be based on long-term as well as recent interactions—including those currently in process.

ZLE applications are discussed in detail in the following section.

leveraging the ZLE core

The ZLE core exists to serve. As discussed, key enterprise systems—from ERP to legacy OLTP and database systems to Web application servers—can interface or integrate with one another via the ZLE core. Business intelligence applications can draw data

from the ZLE core and then feed back resultant models. And specialized ZLE applications can be built on top of the ZLE core—and its ZLE data store—and use its native services.

This section examines the new types of ZLE applications that are being created by HP and its ZLE ISV partners and customers and describes how those applications leverage the ZLE core's real-time environment and application integration capabilities. ZLE applications developed by HP include the Interaction Manager and the Customer Manager.

hp interaction manager

The HP Interaction Manager is a set of objects designed to provide personalization, selling recommendations, and fraud detection services to requesting applications within a ZLE framework.

Integrating with the ZLE core, the Interaction Manager captures customer interactions as they occur and flow through the core. It leverages the core's rules and scoring services and real-time data cache and delivers recommendations based on the most current data to any or all customer touch points and delivery channels. The information it leverages as it makes its decisions can span from data created years and months ago to just seconds ago. So, recommendations can reflect a customer's traditional preferences as well as immediate interests.

On a technical level, the Interaction Manager "sessionizes" data in real time. This means that as interactions occur, they are assigned to a session. All subsequent clickstreams, offers, and purchases are grouped into this common session in real time, so they can be looked at holistically. Armed with real-time information, the Interaction Manager can react quickly to changes in customer behavior.

The Interaction Manager supports interaction with three types of customers: known, anonymous, and ambiguous. Known customers are those with unique identifiers and positive forms of identification; that is, they are registered with you by name. Anonymous customers may possess unique identifiers but lack positive forms of identification; that is, you may recognize their password or have a cookie to identify them, but you do not know who the person actually is. Ambiguous customers have unique identifiers, but the identifier can be resolved to more than one customer, such as anyone in a particular family sharing an account of some sort.

hp customer manager

The HP Customer Manager addresses, in real time, the need for superior-quality customer information for CRM. It does this by maintaining the integrity of customer tables in the ZLE data store and by invoking best-of-breed de-duplication and database hygiene software from leading ISVs such as Trillium and Acxiom. The Customer Manager provides a set of I/O routines to access pertinent data in the ZLE data store so it can be de-duplicated and cleansed and ensures that accurate data is available to remote systems.

An example of the Customer Manager at work would begin with a change being made to a customer file by a remote application. As the message is published to the ZLE core, the core calls on the Customer Manager to verify, de-duplicate, and cleanse the data. The Customer Manager invokes algorithms for name matches and so on and determines whether it is dealing with an existing or a new customer. If it is a known customer, the

Customer Manager pushes updates out to other systems that maintain customer files, so that all customer records stay current, and the information they contain stays in sync.

hp ZLE development kit

To enable companies to build ZLE applications, HP provides a comprehensive ZLE Development Kit (ZDK). The kit is both a toolset and a facility for sharing intellectual property and software that can subsequently be customized to particular ZLE environments.

The kit enables development using industry-standard development environments and deployment under CORBA or Tuxedo. It contains generic templates that provide reusable code sets for rapid time to market. The code sets cover basic functions to which a developer can add specific business logic. A scripting wizard, meanwhile, generates CORBA objects or Tuxedo servers.

In addition to templates and the scripting wizard, the kit provides example business scenarios that are specific to particular industries and functions.

ZLE applications from ZLE ISVs

HP ZLE ISVs that have already developed ZLE applications include Protagona (formerly Recognition Systems), MicroStrategy, and ACI. They include functionality for detecting fraud, maximizing the value of marketing campaigns, and electronic payments.

- *Narrowcaster:* Uses MicroStrategy software running against the ZLE core's database to score for fraud and then pushes (narrowcasts) information out to users' phones, pagers, and so on, as notable events occur within the ZLE framework. For example, a clerk or call center rep is notified that a current transaction has been identified as possessing a high risk of fraud before the transaction is completed.
- *Campaign management:* Enables Protagona's widely deployed campaign management software to directly leverage the huge volumes of constantly refreshed data in the ZLE data store to manage and fine tune campaigns in real time.
- *E-payment:* Fuses the power of the ACI Commerce Framework and the HP ZLE framework to provide banks with ZLE-enabled functionality such as
  - Customer authentication and transaction authorization based on up-to-the-second customer-centric views
  - Multichannel push capabilities to leverage up-sell and cross-sell opportunities at ATMs, websites, and mobile phones
  - Push capabilities for enhanced fraud protection, transaction settlement, and exception management

inside the ZLE core

This section examines what happens inside the ZLE core. It discusses the kind of information contained in the ZLE data store and how this hot cache of information is kept in sync with the applications and systems it integrates and serves.

kinds of data

The ZLE data store is based on a cluster-aware relational database management system—the NonStop SQL database running on the NonStop platform. In its role as a real-time enterprise cache, it contains all three of the primary types of data found in the relational database world:

- *State data:* current value information, such as a customer's current account balance, an inventory balance, credit ratings, and so on
- *Event data:* detailed transaction or interaction-level data, such as call detail records, credit card transactions, Internet or wireless interactions, and so on
- *Lookup data:* data such as control and lookup tables that is not modified by transactions or interactions

So, how do you synchronize this data? Lookup data is the easiest to deal with, as it is largely static. When it does change, those changes are typically made in batch.

Event data presents a log of events that have occurred, such as call detail or order records. Once captured, event data is never modified but is propagated from one point to another. So the challenge in a ZLE environment is not to keep event data in sync, but to propagate it reliably and in real time to the data store and its subscribers.

State data is substantially more complex than event data. It represents the current state of objects or subjects and is modified by transactions. This kind of data presents considerable challenges. For example, should a customer record reside on multiple systems, any changes to the record must be reflected in all of them. The problem is that state data does not always map from one system to another. Synchronization is therefore a significant task, requiring tools and architectures that enable

- Bidirectional synchronization of events
- Complex transformations of data
- Rules for governing synchronization, avoiding collisions, establishing loops, and so on

methods of updating

The ZLE data store is based on an operational data store (ODS) model, yet it differs in many ways from a traditional ODS. The typical ODS is volatile (that is, changes are not appended as they are in a data warehouse built for historical views). Further, in the typical ODS, detailed transactions (event data) are kept elsewhere, such as in an enterprise data warehouse.

In contrast, the ZLE data store is both volatile and subject-oriented, and it contains event data. It is also real-time focused. Batch requirements can be accommodated, but the main task is to propagate data as quickly and efficiently as possible to support real-time integration. This hybrid message store, real-time data warehouse, and integrated business state engine construct is maintained by a mix of state-of-the-art HP and third-party technologies, including messaging, transactional, and batch (see figure 4).

- *Messaging technologies:* A set of off-the-shelf application integration suites is provided to support asynchronous publish/subscribe, routing, and other EAI-type services.
- *Transactional technologies:* OLTP-type technologies are provided to support a synchronous request/response model that enables large numbers of users and

applications to directly access data. These capabilities are built on top of proven application server technologies.

- *Batch technologies:* Batch services are supported to extract and transform data for analysis and to move data in and out of legacy batch systems.

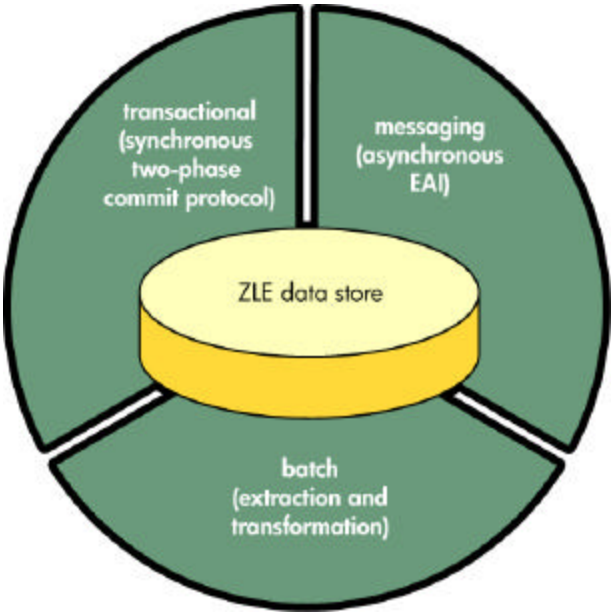

figure 4. the ZLE data store maintains data through a dynamic combination of batch, messaging, and transactional technologies

batch maintenance

Of the three methods of updating, batch is the most straightforward. It is employed to move large amounts of data into and out of analytic applications as well as batch-oriented operational systems, such as billing systems. Though it is not a real-time solution, it is necessary to support, preferably in a state-of-the-art fashion.

The ZLE framework offers two software services for high-performance batch maintenance of the ZLE data store. The HP Mining Integration Facility is a GUI-based service that is used to profile and transform data to be mined for business intelligence purposes (see sidebar). Users profile live data by running queries against it to determine its attributes. You can then extract and move the profiled data to the data mining environment, as well as send the resulting models back to the ZLE data store. The Mining Integration Facility makes full use of NonStop SQL features such as disk process sampling and parallelism. It also supports direct parallel transfers to SAS, along with direct access to the SAS model repository.

The high-performance insert service does the heavy lifting. It is designed to spread the workload across hundreds of processors, if necessary. Built on either a CORBA or a Tuxedo framework, this service supports efficient parallelization and bulk insertion of batch and minibatch (small batch insertions throughout a day) updates to a ZLE data store based on the NonStop SQL database. It is optimized to handle thousands of rows per second of random inserts as well as upserts and has been tested handling as many as 70,000 rows per second.

In addition to these HP services, industry-standard products from leading third-party extraction, transformation, and load (ETL) vendors provide the ability to efficiently move

batch data into and out of the ZLE data store. This includes popular products from Informatica and Savant. Thus, the HP ZLE framework enables companies to leverage off-the-shelf ETL and data integration software they may already have deployed.

leveraging live data for real-time intelligence

The HP Mining Integration Facility is a set of software tools developed by HP that

- Profiles and transforms data so it can be mined
- Moves transformed data into a mining environment, where models can be built
- Deploys models back into other services within the ZLE core

With a GUI, users can run series of queries directly against the live data in the ZLE data store to determine what attributes may be valuable in a data mining scenario. For example, data can be profiled to discover the distribution of customer ages, which in turn can help determine whether age might be an appropriate factor to use as a buying pattern predictor.

Once a corrected data mining case set has been derived, and the data has been transformed to make it conducive to a mining operation, the Mining Integration Facility allows the user to batch transfer this data, in parallel, directly into SAS running on a variety of UNIX® platforms.

Tree-based models can now be built within SAS, and those models can be deployed via the Mining Integration Facility directly back into the ZLE architecture to be used by ZLE core services such as the unified rules service and the scoring service.

message-based maintenance

Messaging, or asynchronous, technologies are used to keep volatile state data within the ZLE data store in sync with other applications in real time. This is a standard EAI tool suite task. HP works closely with EAI toolset vendors so customers can leverage these tools and complement them with the capabilities HP provides.

An example of messaging technologies in action might start with a change being made to a customer file in a Clarify or Siebel application. An industry-standard EAI toolset, such as that provided by SeeBeyond, can be used to transform the data and route it to the ZLE core. There the data can be enriched with customer identifier data cached in the ZLE data store and then routed to perhaps an SAP system. After an adapter transforms the data to the SAP format, the change to the customer data can be made in the SAP system.

Through work with its ZLE partners, HP ensures that companies can take full advantage of the leading EAI toolsets and publish/subscribe messaging solutions on the market, including tools from vendors such as IBM, SeeBeyond, Talarian, TIBCO, and webMethods. Two specific examples are NonStop SeeBeyond and Talarian.

HP offers a unique version of the SeeBeyond toolset, called NonStop SeeBeyond. This product is the result of a deep port of the SeeBeyond e*Gate solution to the NonStop platform. Consequently, companies benefit from the NonStop platform's continuous

availability, linear scalability, and high-performance, high-reliability transaction processing environment when engaging in high-volume, business-critical ZLE operations.

Talarian's publish/subscribe environment running on the NonStop platform provides companies with the industry's highest levels of performance, reliability, and scalability for publish/subscribe messaging. In addition to high-performance publish/subscribe, the Talarian solution provides a Java Message Service (JMS) application program interface (API) for persistent storage in a JMS environment, as well as robust queuing with IBM WebSphere MQ support. The Talarian solution leverages the NonStop platform to its fullest, including taking full advantage of the parallel attributes of the NonStop SQL database acting as a relational message store.

transactional services

The third technique for maintaining the ZLE data store involves using transactional services in support of request/response. A series of transactional services are provided that run on top of a choice of application server environments, including HP's J2EE application server and CORBA- and Tuxedo-based transaction monitors. These core services include services for business rules, transformations, personalization, and more. They are highly integrated with a great deal of synergy among them.

advantages of application server hosting

Through the agency of the application server environment, many remote enterprise applications and their users can request information and services from the ZLE core. The emphasis is on handling high volumes of transactions that require extremely short response times.

While most EAI messaging software supports request/response, a transactional application server model provides robust hosting benefits. In the ZLE framework, this includes the ability to run parallel tasks that are dynamically load balanced across nodes on a NonStop server. It also means the ability to monitor transactions and restart them in the event of failure, manage transaction boundaries, manage queues, and so on, in a robust fashion.

A transactional application server environment provides a common standards-based framework with which to interface ZLE core services and to interact with other enterprise applications, like Web servers, ERP applications, and legacy systems and databases. Because the application server utilizes a multi-threaded architecture, it can handle huge numbers of concurrent users while providing consistent service levels.

role of adapters—actional control broker technology

Like the asynchronous approach, the transactional approach requires some sort of adapter—in this case, one that allows an application to call the ZLE core explicitly and get a response that can be used as a part of its transaction. For example, a customer service representative changes an address using a Siebel application and receives an up-sell recommendation that then displays on the screen. A noninvasive adapter can capture such changes but would be hard pressed to allow the application to hang and wait for a response.

HP ZLE ISV partner Actional provides lightweight, high-performance adapters that enable direct request/response access to CORBA objects, Enterprise JavaBeans (EJBs), and Tuxedo servers. These adapters integrate applications at the data level and enable them to interface with the ZLE core without a mediation layer. Request/response performance is thus an order of magnitude faster than with conventional EAI methods.

The utility of Actional adapters is that their underlying Control Broker technology works like a client making a call into an object. So, they appear to the application to be just one more component. For example, to a Siebel or a Broadvision application, the ZLE framework to which it connects via an Actional adapter looks like another Siebel or Broadvision component. But, in actuality, the application made a high-performance call directly into the ZLE core.

Actional provides a broad range of *application* and *technology* adapters, plus a toolkit for developing *custom* adapters. The difference between these three types of adapters is fairly straightforward:

- Application and custom adapters normalize messaging between the ZLE core and either standard packaged or custom enterprise applications.
- Technology adapters enable interoperability with non-application-specific environments such as an IBM CICS queue, HP Pathway, databases, Web application servers, and so on.

While application adapters interface with a program's existing APIs, technology adapters are used when there is no built-in interface. Accordingly, data can be pulled directly from an IBM WebSphere MQ message queue or an Oracle® database table. Application adapters can be constructed so that no modifications are required to the source application, and all security mechanisms stay intact. Some adapters are targeted for interchange of data with business networks, like SWIFT for financial and Healthnet for healthcare.

high-performance data transformation service—NonStop DTE

As with asynchronous messaging, it is also necessary to perform data message transformations in a transactional environment. HP NonStop Data Transformation Engine (DTE) software consists of best-of-breed data transformation software for automating transformation.

Based on Mercator Enterprise Broker software from Mercator Software with scalability enhancements from HP, NonStop DTE runs as a NonStop CORBA or NonStop Tuxedo object—gaining all the reliability and scalability that this implies. Fully mapping differences in message syntax, semantics, and values, NonStop DTE efficiently transforms and integrates data in real time from any number of diverse sources, including data in XML, EDI, SWIFT, HL7, and other crucial e-business data formats. Integration is based on open standards and is entirely noninvasive:

- No modifications to applications are required.
- No new middleware is needed—NonStop DTE leverages CORBA or Tuxedo middleware for data transport.
- NonStop DTE provides native integration with relational databases.

In addition, NonStop DTE leverages advanced metadata-driven transformation technology to substantially reduce the time, effort, and cost of creating and maintaining durable application interfaces across heterogeneous environments. Metadata importers convert application formats to graphical form, while powerful graphical tools eliminate the need for hand coding.

unified rules service—blaze advisor rules engine from hnc software
A key core service, the unified rules service is designed to allow rules to be executed within a high-performance rules engine that can be called by other applications in a ZLE framework. The functions of the service include

- Enabling the easy creation of business rules using a graphical user interface or a syntax such as a declarative English-language sentence
- Finding and applying the most applicable business rule on the occurrence of an event and getting to the desired data or answer as rapidly as possible

The ZLE core uses the inference-based Blaze Advisor rules engine from HNC Software. This engine uses intelligent, nonsequential methods to find the appropriate business rule quickly and efficiently, regardless of the complexity of the rules or the size of the rules set. Consequently, the number of rules a company maintains has minimal impact on the performance of the rules service.

Rules and policies are kept in a centralized logical location, next to the message stream, and can be called by any object within the transactional framework of the ZLE core. This position allows the rules service to drive workflow more easily, facilitate rapid changes to the rules, and ensure consistent application of business rules across multiple systems (call centers, e-commerce sites, POS applications, and so on).

In addition, new ZLE applications that leverage the rules service can be written. An integrated scoring service uses the rules service and the Mining Integration Facility to facilitate the importation of models for personalization, profiling, and other CRM-type tasks.

scoring service
The scoring service is a peer of the unified rules service. It is designed to take the output of a data mining model (created using data from the ZLE data store) and execute against it to score a customer in real time. For example, it can be used to determine which offer a customer is most likely to accept and in which situation while the customer is still online with a call center or on a website.

The Mining Integration Facility can feed the scoring service. This facility allows models to be built, tested, and then exported to an SAS model repository. A user can browse the repository using a Web-based GUI, select which model(s) to deploy, and then automatically deploy that model into the ZLE data store.

When an application, such as the Interaction Manager, calls into the scoring service, the service immediately executes against the model. The scoring service can leverage appropriate historical information (past browsing patterns, purchases, and so forth) residing in database tables in the ZLE data store as well as information from the current interaction. It then scores the customer's likelihood to accept a given offer and provides that score back to the Interaction Manager in real time.

ZLE in action: real time in the real world

The ZLE framework is architected to perform myriad functions concurrently and in real time. This section details how all of its components coalesce to handle key business functions—and some of the most complex transactions imaginable—in a zero latency fashion. The focus is on a retail organization with both brick-and-mortar and Web operations. The total environment includes a call center powered by a CRM application

such as that provided by Siebel or Clarify, a POS system in each of the stores, and a Web environment based on software from a company such as Broadvision.

the art of doing it all—in real time

The goal is to create a ZLE data store in the logical center of this distributed environment to

- Track customer requirements and actions in real time
- Deliver specialized ZLE services on top of the data store that will personalize and increase the value of customer interactions across all channels (see figure 5)

Industry-standard EAI software is used to maintain an integrating link between all systems with customer files and the ZLE data store. The systems integrate and synchronize with the centralized data store, instead of directly with one another, while EAI techniques are used to synchronize volatile state data.

The integration is loosely coupled, avoiding the pitfalls of dependency between systems. When data and services are required, remote applications call on the ZLE core residing on the fault-tolerant NonStop platform and not directly on each other. So, data redundancy is controlled, other systems and applications do not take up the processor cycles of individual systems, and continuous availability is ensured. As data changes are pushed through this integrated environment—from one system to another—what emerges is a moving snapshot of the state of the business, its customers, and its opportunities for driving new revenues.

As data changes are pushed through this integrated environment—from one system to another—what emerges is a moving snapshot of the state of the business, its customers, and its opportunities for driving new revenues.

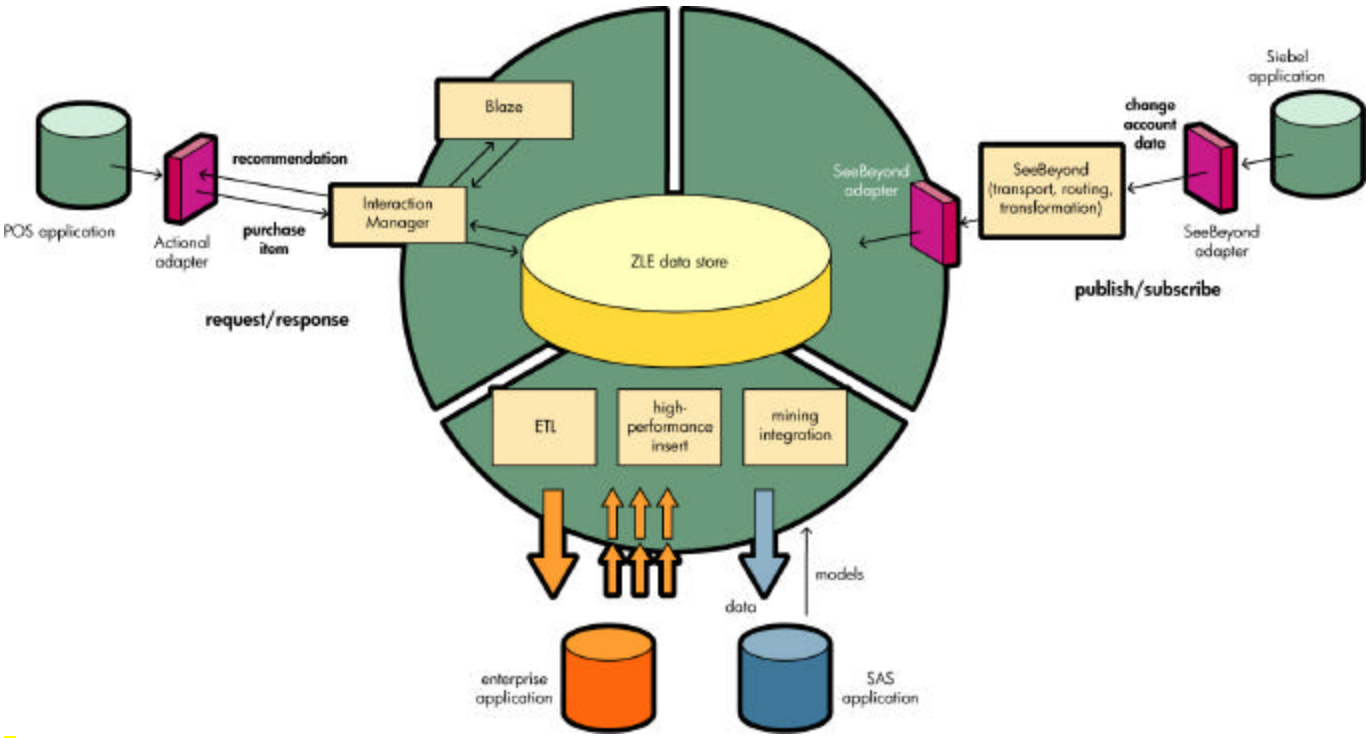

figure 5. the ZLE data store tracks all interactions with customers in real time and enables services that leverage real-time data to better personalize the customer's experience

taking it to the core

Now for the step-by-step details of a zero latency enterprise in action. To start, adapters provided by SeeBeyond connect remote systems to the ZLE core. The ZLE data store can call on the specialized services of the Customer Manager to de-duplicate and cleanse customer data when it enters the database. So, changes to customer data initiated by the call center application can be captured, transformed, and routed via the Customer Manager, which uses software from Trillium or Acxiom to discover whether the customer exists in the ZLE data store.

spotting predictable behaviors

As already noted, state information about each customer is synchronized within the ZLE framework with EAI technologies. But it may also be necessary to synchronize information with batch systems. POS systems, for example, are traditionally batch oriented, so large batch updates must be accommodated. These systems can update the ZLE data store at regular intervals (for example, every 20 minutes) using the high-performance insert service.

Meanwhile, batch services can be used to send bulk data to an SAS application for data mining. For example, the Mining Integration Facility can profile live data in the ZLE data store, determine what attributes are of interest, establish a case set, and leverage batch services to load the data directly into SAS. Once models predicting customer behavior are created, they can be deployed back into the ZLE data store for use by the unified rules and integrated scoring services.

leveraging real-time knowledge

The retailer may want to develop a model for up-selling or cross-selling and then deploy it directly into the transactional portion of the ZLE core. For this purpose, a high-performance application adapter from Actional can be embedded directly into the Broadvision application that underpins the retailer's Web operations. There it is treated as just another Broadvision object as it passes data about customer contacts and interactions back to the ZLE core in real time. The Interaction Manager, which functions as a CORBA object, leverages this information to

- Access the ZLE data store for data on old and current interactions
- Feed the data through the unified rules and scoring services
- Recommend an offer to the customer while the interaction is in progress

The centralized ZLE core can develop highly effective recommendations while handling an immense volume of transactions in real time on top of a state-of-the-art transactional application server. Yet, as far as the downstream applications such as Broadvision are concerned, they simply invoked a local object and received an instantaneous response.

Additional functionality that can be invoked includes narrowcasting and campaign management. For example, should a back-ordered item come into stock or a fraud event occur, MicroStrategy's narrowcasting application running against the ZLE data store can immediately push out an alert, notifying a subscriber via e-mail, phone, or pager. Similarly, Protagona's campaign management application can execute and manage highly effective campaigns based on real-time feedback flowing through the ZLE core.

taking it to a new level

This example scenario reflects a level of integration that goes far beyond traditional EAI:

- It keeps state data synchronized.
- It includes batch interfaces in and out of operational systems and analytic applications.
- It offers unique services on top of a real-time data store—services that provide a higher level of functionality than previously possible—to all the applications and systems in the business.

extending beyond the enterprise

The benefits of a ZLE environment are not limited to a single organization. It can be advantageous to expose services and transactions to other companies as well, such as trading partners. B2B ZLE requires the use of standards such as XML and SOAP to communicate with other organizations, market exchanges, and so on. The challenge is how to manage the exposure to other companies, including identifying the types of services a partner needs to access and what security mechanisms to employ.

The ZLE data store construct is invaluable in this scenario. It provides a company with the ability to audit all records of messages and transactions inside and outside of the enterprise. It can manage all trading partner and configuration information in a centralized store. And it can handle all validation tasks. Just as important is the loosely coupled integration of systems enabled by the ZLE framework. Loose coupling plays a significant role in protecting a company's core information in a B2B environment.

Many security and protocol services are built into the platforms that HP provides for ZLE solutions. HP ZLE partners, meanwhile, provide software for B2B partner management. This includes managing trading partner profiles, enveloping XML- or EDI-based messages, and reporting and auditing services. SeeBeyond e*Xchange and TIBCO ActiveExchange are two high-profile products that can be leveraged in this scenario with a ZLE framework.

ZLE web services

Extending ZLE beyond the enterprise means more than just offering B2B interfaces to other companies. Increasingly, it means offering services through a set of standards called Web services. Through support of Web services, companies can advertise in a Web services directory the standards that are required to access their services and then allow access through the advertised interfaces.

The ZLE core, as a real-time integration hub, provides access to legacy systems and real-time transactions. This makes it a logical gateway for offering access to compound services (a compound service might provide numerous services but is exposed as a single service) via support for Web services. In this way, an outside enterprise can access data, transactions, ZLE applications, process flows, workflows, and so on. The ZLE data store is ideally suited to supply elevated levels of availability for such compound services without any strain on back-end databases and systems.

addressing platform requirements

The ZLE framework presupposes a set of demanding platform and technology requirements. If these requirements were not so rigorous, the ZLE solution would fall short of the goal of recognizing events in real time and triggering immediate, appropriate actions across the enterprise.

As a start, a ZLE platform needs to handle a comprehensive mix of data, messaging, and hybrid functions concurrently while protecting workloads. Other major demands on the platform include support for 24 x 7 operations and seamless scalability.

The NonStop platform provides parallel execution of key functions to ensure real-time operations. The ability to support a single system image across multiple nodes is built into the platform, while its cluster architecture and associated NonStop SQL database and application server environment enable massive scalability, extreme performance (even under peak loads), and unrivaled availability for 24 x 7 operations.

Finally, the ZLE framework's open architecture enables core services and plug-in applications to be based on best-of-breed solutions from leading ZLE ISVs. This, in turn, ensures the strongest possible support for the full range of data, messaging, and hybrid demands.

conclusion

ZLE is all about achieving competitive advantage. It keeps information from myriad enterprise systems refreshed, ready for consolidation, and available for real-time deployment in decision-making, CRM, inventory management, and marketing campaigns. The ZLE framework captures this vision by

- Combining and hot-caching information from across the enterprise in real time

- Providing a unifying rule-driven, workflow-informed architecture for the flow of information
- Breaking down the performance and availability dependencies between systems
- Enabling new types of applications that could not exist without real-time data and application integration

The ZLE framework is a work in progress. It consists of a data-centric architecture that is unique to HP and not biased towards any particular technology. It is constructed to leverage best-of-breed tools into customized, integrated solutions for application and data integration. And it leverages all of HP's considerable skills in real-time, all-the-time computing.

Traditional database constructs and EAI functionality cannot come close to supporting zero latency operations. But the ZLE architecture offered by HP and its partners executes all of the transactions that define ZLE—including massive numbers of transactional inserts and updates hitting against the same database tables concurrently. It is designed specifically to perform different types of functions and process different kinds of workloads in parallel and around the clock—no other vendor can claim the same concurrent mixed-workload capabilities. And it provides the added functionality of specialized ZLE applications from HP and best-in-class ISVs.

Finally, as a pioneer of zero latency computing, HP provides a proven blueprint, or design methodology, for achieving a ZLE environment. This blueprint begins with defining the value proposition and extends through building the high-level architecture and implementation. In addition, HP's dedicated and highly expert HP Services organization is at the disposal of ZLE customers anywhere in the world.

The result is not just a one-stop shop—but the only shop—for unleashing the incredible business potential of ZLE.

For more information, go to www.hp.com/go/zle.

Printed in the U.S.A. 01-1060