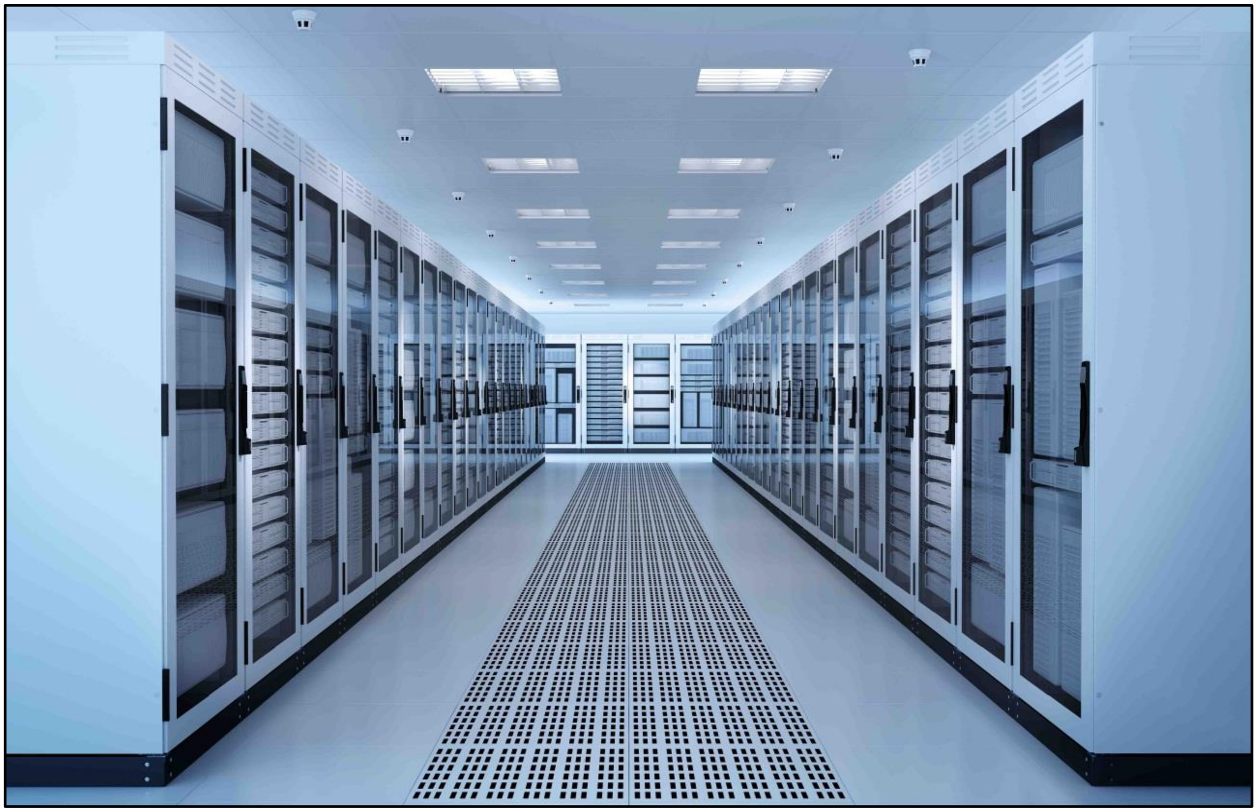
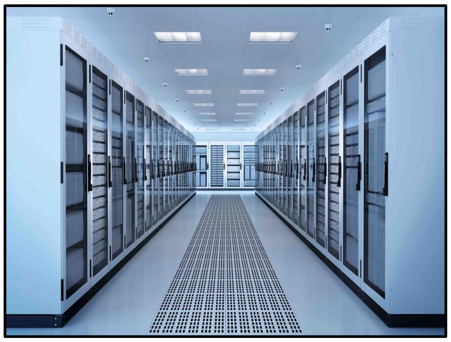
HPE Shadowbase Streams for Application Integration

A Gravic, Inc. White Paper



Executive Summary



The HPE Shadowbase product suite (built by Gravic, sold by HPE) distributes real-time critical data and information to target databases and to application environments throughout the enterprise. Uses for Shadowbase products include achieving high or continuous availability for business continuity environments by synchronizing data changes between active or passive redundant systems, integrating applications, feeding data warehouses and real-time business intelligence facilities, and driving ETL (extract, transform, and load) utilities.

HPE Shadowbase Streams, a member of the HPE Shadowbase software solutions suite, uses change data capture (CDC) technology to stream data generated by one application to other applications, enabling low-latency, real-time data distribution between heterogeneous systems and applications. *Shadowbase Streams* for data integration and *Shadowbase Streams* for application integration provide the facilities for integrating existing applications at the data level or event-driven level in order to create new and powerful functionality, supporting the Real-Time Enterprise (RTE).

Using *Shadowbase Streams*, changes made in any database can be quickly and easily integrated into other data environments to keep that target information synchronized. The changes can be made in real-time or can be batched for periodic snapshot or *micro-batch* updating.

Additionally, using *Shadowbase Streams*, applications that once were isolated can now interoperate in an event-driven fashion in real-time. Critical data generated by one application is distributed and acted upon immediately by other applications, enabling the implementation of Event-Driven Architectures (EDA).

Shadowbase Streams supports many models for data distribution, including maintaining a remote database of critical data; sending critical data to client applications or servers directly via queues, interprocess messages, TCP/IP sessions, or client APIs; publishing data to applications that have subscribed to the data; and responding to poll queries from client applications.

Shadowbase Streams is extensible. It allows the user to embed custom processing logic into the data-processing path. It readily filters, transforms, and adapts data from one application or database environment into the protocol or format required by another application or database environment, all without requiring any changes to the existing application that is generating the data.

Shadowbase Streams modernizes legacy applications by integrating diverse applications across the enterprise so that new and valuable services may be generated to enhance competitiveness, to reduce costs or to increase revenue, to satisfy regulatory requirements, and to generally improve the user experience.

Shadowbase Streams simplifies the integration effort by providing a set of adapters to more quickly and easily integrate various environments. For example, adapters are available for Kafka, IBM MQ, flat-files, and JSON formats, with more being developed in every release.

In this paper, several customer production examples of *Shadowbase Streams* for application integration follow a description of how *Shadowbase Streams* works.

Note that this paper focuses on using *Shadowbase Streams* for application integration. For data integration needs, please refer to the companion document titled *HPE Shadowbase Streams for Data Integration*.

Table of Contents

Table of Figures

HPE Shadowbase Streams for Application Integration

HPE Shadowbase Streams (built by Gravic, sold by HPE) is a powerful and flexible facility that enables diverse applications to interoperate with each other at the event-driven level (called *application integration*) and at the data level (called *data integration*). This capability is provided even for legacy applications that were never intended nor designed to work together. Shadowbase Streams is a member of the HPE Shadowbase product suite and focuses on distributing information in real-time throughout the enterprise, enabling the provision of a Real-Time Enterprise (RTE).[1] It rapidly delivers information where and when it is needed without customer application modification. The application's responsibility is simply to make functional use of the data delivered by Shadowbase Streams.

With application integration, events generated by a source application are sent in real-time to a target application for processing, and is known as an Event-Driven Architecture (EDA).[2] Target application processing may or may not result in a reply back to the source application. With data integration, Shadowbase Streams maintains a copy in real-time of selected data from a source system's database in a target system's database or environment. Target system applications may use this local copy of data changes from the source application for expanded functionality and to improve response times.

In this white paper, the use of Shadowbase Streams for application integration is described. A separate white paper, *HPE Shadowbase Streams for Data Integration* is available on the Shadowbase website.

The Data Continuum

From onset to action, there is value in data. The Data Continuum, shown in Figure 1, represents data's typical lifecycle:

1. **Create** – From the Edge, through IoT and other means, data is measured and tracked.
2. **Store** – Once identified, the data is written to a database or other form of storage.
3. **Analyze** – Analytics programs inspect the data, scanning for anomalies and other valuable information.
4. **Learn** – The data is translated into meaningful information stating facts, key insights, and accurate metrics.
5. **Act** – Strategic analysis of the information creates specific and advantageous actions.

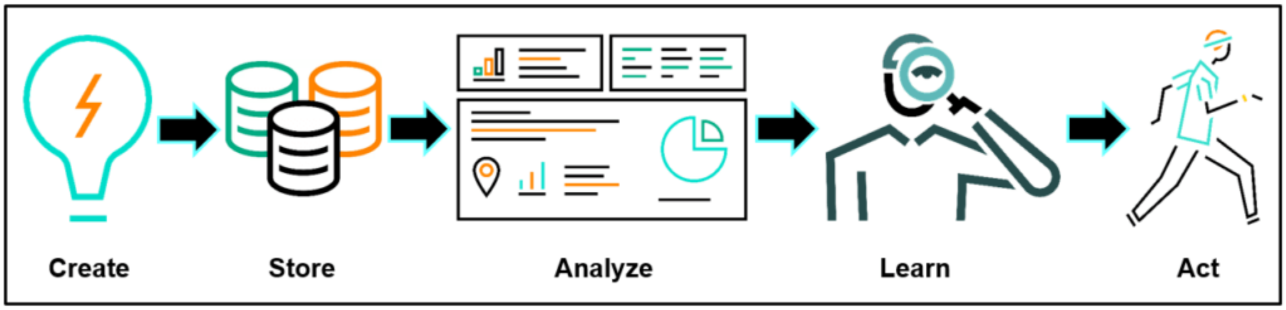


Figure 1 – The Data Continuum

Companies that record and analyze data about their business can analyze their performance, which can help determine the next action to take.

The Need to Integrate Applications

Over time, the number of legacy applications developed to support an enterprise's operations grows significantly. Many of these applications are silo applications that recognize inputs from and deliver outputs to hard-coded interfaces only. The only way to share information generated by one such application with another is via virtual tape (or magnetic tape images on disk), or by file transfer, typically processing the data in batch mode.

[1] Real-Time Enterprise is a concept in business systems design focused on ensuring organizational responsiveness.
[2] Event-Driven Architecture is a software architecture pattern promoting the production, detection, consumption of, and reaction to events.

This mode of interoperation (or complete lack thereof) is unacceptable in today's 24x7 always-online environments. Applications that were never intended to work with each other now must cooperate closely in real-time. For instance, a new customer-facing online store application must have immediate access to inventory changes from a back-end inventory program that was written long before online stores were even imagined.

Using today's modern technologies, application programs can be written so that they can cooperate with other yet-unknown applications to be created in the future. For instance, the Service Oriented Architecture (SOA) creates a platform on which applications can be built as a set of services. The services provide their own encapsulated data and processing functions but expose a standard discoverable interface that can be used by any process to invoke these services, via a ubiquitous communications protocol (such as HTTPS). Applications are built by calling (composing) these services to provide critical functionality. Thus, new applications can easily use the services exposed by existing applications.

Such is not the case with many applications in use today, and certainly not with legacy stovepipe applications. Their functionalities are buried deep in their proprietary code, accessible only via unknown and arcane proprietary interfaces. In fact, for many of the older applications in use today, the code is lost, or the programmers who wrote and understood the applications have long ago moved on. There is no easy way to leverage the (often considerable) functionality of such applications to build new business services. If a company is fortunate enough to be able to continue maintenance on a legacy application, the cost and risk of modifying that application to provide new functionality is likely to be significant.

What is needed then is a simple method for integrating existing (often legacy) applications with new applications that are being created. Fulfilling this need is the challenge that Shadowbase Streams for application integration solves. Shadowbase Streams provides the mechanism to expose events and data generated by a source application to a target application without software modification to either.

Methods for Integrating Applications

In order to interoperate with each other, applications may be integrated at either the service-layer or at the event-layer.

Service-Level Integration

With service-level integration, special interfaces expose server system data or application services to external client applications, as shown in Figure 2. One way to accomplish this exposure is via specially designed agents that reside on the server system. For instance, a database agent may provide access to the target database via ODBC, JDBC, OLE, XML, or ADO.NET.

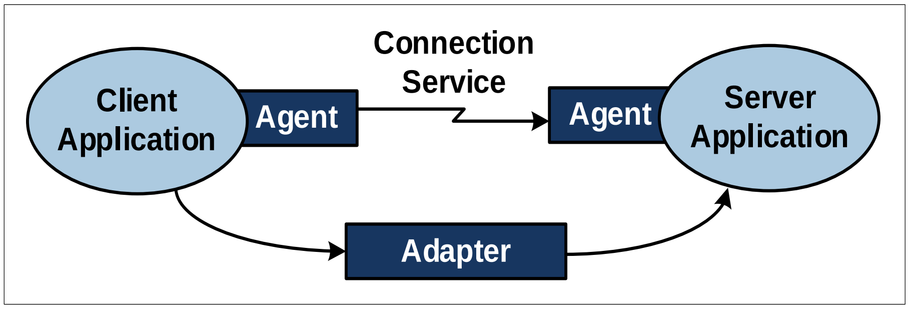


Figure 2 – Service-Level Integration

An application agent may use application metadata to define the application services that will be exposed and to provide access to these services via XML, SOA, JCA, or 3GL APIs. Often, a complementary agent that allows client applications to invoke the services available from the server system is provided for the client system.

An alternate implementation involves designing a special adapter to connect well-known applications. For instance, an adapter might supply a Customer Relationship Management system (CRM, the client system in this case) with the services of a name/address scrubber (the server system) to rationalize data into a standard form. Adapters are typically limited to well-known, popular applications and protocols.

Event-Level Integration

Event-level integration enables Event-Driven Architectures, which deliver services based on changes made to the source system's database. Event-level integration avoids the necessity of developing special agents or adapters for each application that is to be integrated. Rather, it monitors source application events in real-time and sends them immediately to the target application so that action can be taken on the event by the target.

Event-level integration is the method employed by Shadowbase Streams for application integration. It makes use of the fact that in most applications, events are evidenced by changes made to the application's database. As shown in Figure 3, event-level integration monitors these database changes. If a change represents an event of interest, the event data is reformatted as required by the target application by the event transmitter and is sent to the target application for processing.

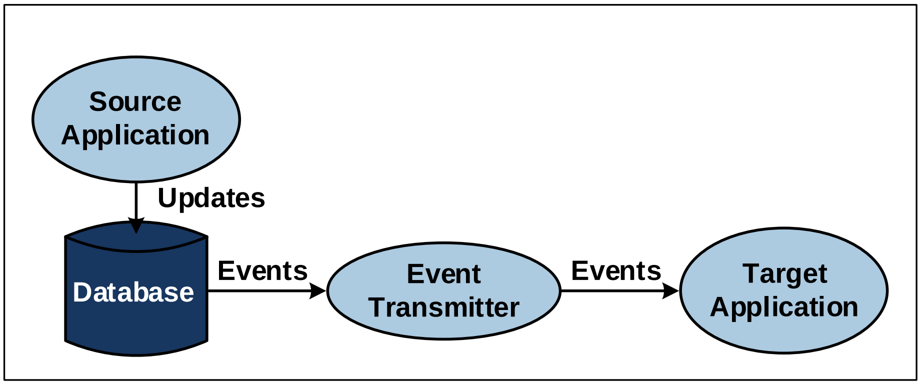


Figure 3 – Event-Level Integration

There must be some means for the event transmitter to pass the event to the target application, via a mechanism that the target application already provides (such that the target application does not need to be modified). Legacy applications often expose their services to other applications by providing interfaces that other applications can use to submit events on which action is to be taken. This interface might be a proprietary interface that implements an application programming interface (API) called by a source application to provide event information. Alternatively, messaging middleware may be used to send the target application event notifications. ODBC or JDBC may also be used to submit event data to a file or a database on the target system. It is via such an interface that the transmitter sends events to the target application. Shadowbase Streams is an event transmitter.

Why Integrate at the Event Level?

Using Shadowbase Streams to integrate applications at the event level offers significant advantages over service-level integration:

- *Performance* – With event-level integration, a target application is directly handed the information it needs as soon as it is available. With service-level integration, the remote system must often be queried periodically to get event data. Polling for database changes is inefficient and adds a time delay to the processing of events.

- *Invasiveness* – Service-level integration generally requires that the client and server applications be modified to support the agents or adapters needed to integrate the applications. Furthermore, with respect to adapters, a suitable adapter must be implemented for each pair of applications to be integrated. With event-level integration, there is no need to make application code changes or to implement specialized modules. The event transmitter handles the interface tasks.

- *Seamlessness* – Service-level integration inserts processing functions into the flow of the events or data changes which may impact response times. Event-level integration works under-the-covers. Events are replicated asynchronously from the source database's transaction log. The source application is unaware that replication is taking place.

- *Security* – Integrating at the event level means the event transmitter is the only function that needs access to the source information, and all source information is routed through it. Integrating at the service level means many more application-level components are potentially involved in handling the event data, thereby opening up more points for security breaches to occur.

HPE Shadowbase Streams for Application Integration Architecture

HPE Shadowbase Streams for application integration uses the Shadowbase data replication engine for change data capture and as well as the event transmitter. For many target environments, the Shadowbase architecture provides an adapter to more quickly and easily integrate with that target environment, including Kafka, IBM MQ, flat-files, and JSON format.

We first describe the architecture and characteristics of the engine and then explain how it is used to implement application integration.

The HPE Shadowbase Data Replication Engine

The HPE Shadowbase data replication engine transmits data changes from a source application's database to a target application's event interface. As shown in Figure 4, it is driven by the transaction log of the source database. The transaction log is created by the transaction manager on the source system. For instance, in HPE NonStop systems, the transaction log is the Transaction Management Facility (TMF) Audit Trail. In Oracle systems, Shadowbase Streams uses Oracle Log Miner to retrieve the events from the Redo log. For DB2, Shadowbase Streams uses the IBM Data Event Publisher and MQ Series to retrieve the events from the DB2 Journal. For Sybase, Shadowbase Streams uses Sybase replication to feed the database changes into the Shadowbase engine. For SQL Server, Shadowbase Streams uses database triggers to capture the database changes.

The transaction manager records every change made to the source database in its transaction log so that the database can be reconstructed should it become corrupted for any reason. The log is therefore an instant source of database changes generated by many (most) source applications. By following the transaction log, Shadowbase replication can read every database change in real-time as it is applied to the source database. It can select those changes that represent events of interest, format them according to the requirements of the target application, and pass them to the target application. If there is no transaction log maintained by the source database environment, Shadowbase replication can either receive database changes from triggers added to the source database, or it can follow a change log created by the source application. Shadowbase replication comprises a Collector process on the source system and a Consumer process on the target system (for most situations). The Collector follows the source system's database changes as recorded in the transaction log, selecting those that represent events of interest to the target system. It sends these events to the Consumer via a communication channel. The Consumer formats the events and passes them to the target application.

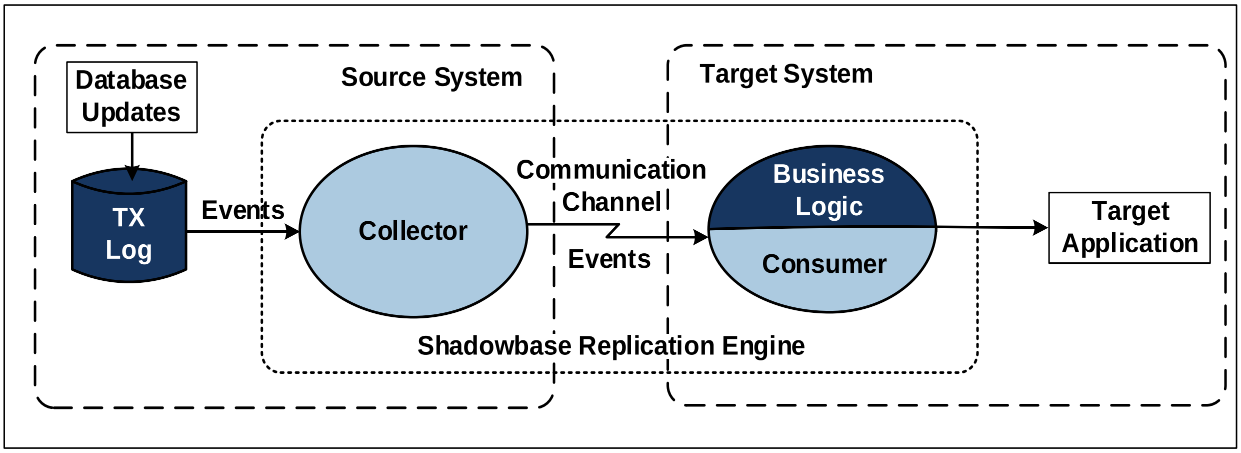


Figure 4 – The HPE Shadowbase Data Replication Engine

Shadowbase replication performs whatever data transformation is required to format the database changes as read from the transaction log into usable events for the target application. Fields can be remapped from a source schema to a target schema. Data values can be reformatted. Fields can be eliminated, combined, or aggregated. Related data can be added from a local database. In short, any enrichment of the data required by the target application can be performed by Shadowbase technology.

This powerful feature is made possible by *user exits* in the Consumer. User exits can be customized by the implementer to provide whatever data transformation is necessary to meet the requirements of the target application. There are two classes of user exits:

- *Scripts* perform the more common data transformation functions such as schema remapping, data selection, field-format conversion, field splits and merges, and aggregations. The scripts are text files of transformation directives read by Shadowbase software at startup and are interpreted/executed for each database object that flows through the Shadowbase Consumer.

- *Code modules* are executed by Shadowbase software for each data change. The code modules are created by the user and are bound into the Shadowbase Consumer. They run at (optimized) compiled code speed and can execute more complex data transformations such as adding data from local databases.

With user exits, any database change can be transformed into an event that is meaningful to the target application. The HPE Shadowbase data replication engine is heterogeneous. It supports a wide variety of source and target systems and databases, as shown in Figure 5. The Shadowbase engine can pass events from any supported source system to any supported target system.[3]

A key feature of Shadowbase software is that unlike other data replication technologies, there are no disk queuing points in the transfer path (unless they are optionally added). Therefore, Shadowbase technology imposes a minimum time delay (or latency) on the delivery of events to the target application from the time that the events are available in the transaction log. Latency times are measured in sub-seconds.

The HPE Shadowbase data replication engine is the heart of Shadowbase Streams for application integration. Figure 5 shows a physical representation of the internal architecture of Shadowbase software, whereas Figure 6 shows a logical view. In this figure, a source system is generating database changes which are delivered to target applications as events by Shadowbase Streams via a variety of mechanisms.

As the source application changes its database, the transaction manager captures the changes in its transaction log. Shadowbase Streams uses powerful change data capture (CDC) technology to follow the transaction log and select those database changes that represent meaningful events to the target system (the changes to select are defined to Shadowbase Streams via configuration parameters). Shadowbase Streams then scrubs and cleanses, reformats, and enriches the changes to deliver events that meet the needs of the target application.

The target application will present an interface to receive external events. It is to this interface that Shadowbase Streams will pass the events. Via user exits, Shadowbase Streams can be configured to drive any target application interface (hence the target application will not typically require modification to receive events via Shadowbase Streams). Typical application interfaces include:

- a work queue from which the target application can fetch events
- a work file from which the target application can read events
- messaging middleware, such as IBM MQ Series, which accepts events and passes them to the target application (via its own mechanisms, for which the target application is written)
- other queuing and messaging environments, such as Kafka
- flat-files in various formats (CSV, fixed-position, JSON format, etc.)
- a specific API that can be called to supply events (e.g., a procedure call), such as an AWS API
- TCP/IP sessions established with the target application over which events can be passed
- published events to which the target application can subscribe

Shadowbase Streams can deliver events simultaneously to multiple target systems and applications. A separate HPE Shadowbase data replication engine is configured for each system. Each system can receive its own subset of events formatted according to its specific needs. Alternatively, a single Collector can monitor

[3] Please check with Shadowbase Support for the latest set of supported source and target platforms.

the source database for events and can send individually selected events to multiple Consumers, or one Consumer for each target system or environment. Each Consumer will format the events for its particular target applications. It is the power of the Shadowbase user exits that allows Shadowbase Streams to deliver events to multiple heterogeneous target applications running on diverse systems via a variety of event-reception interfaces.

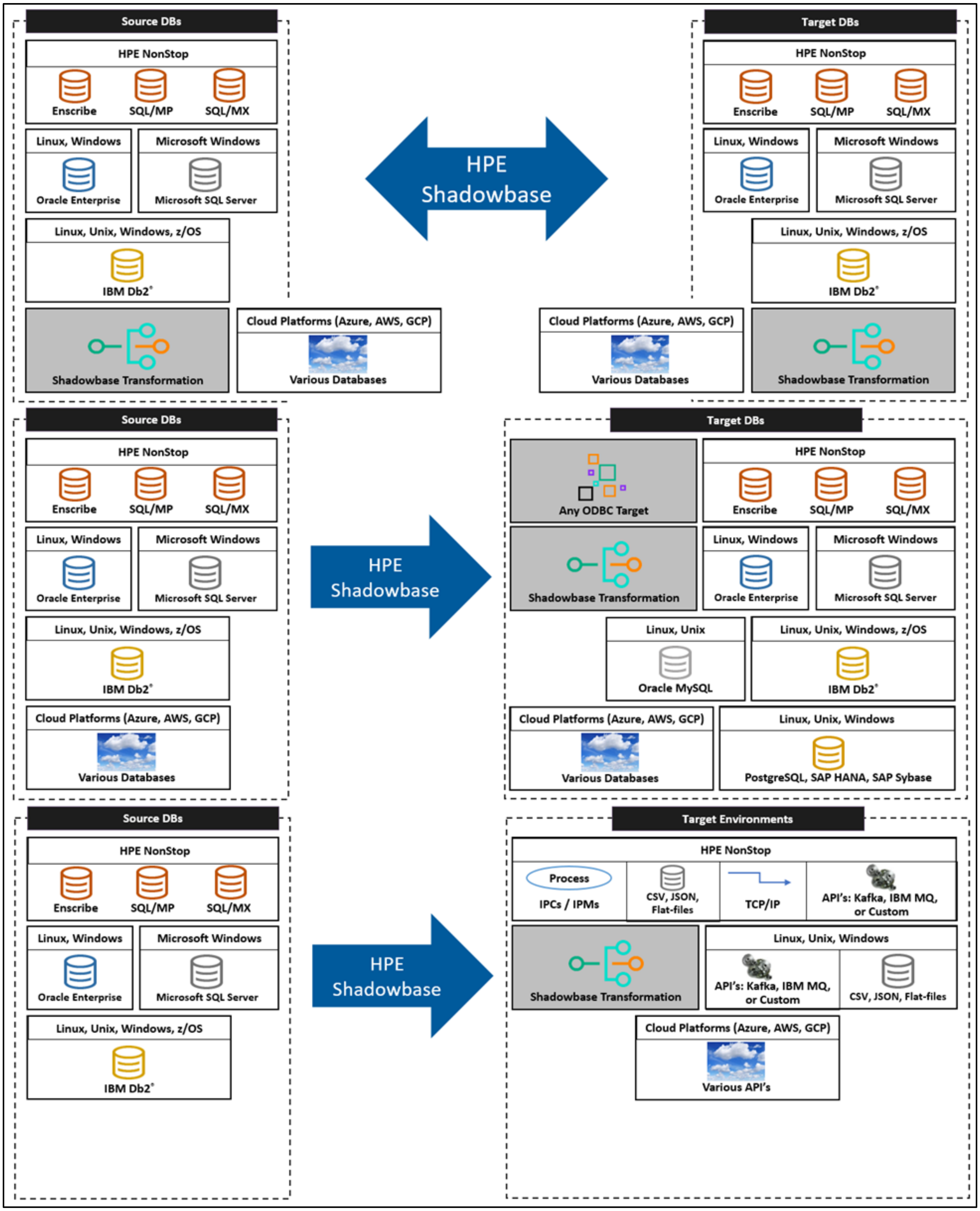


Figure 5 – HPE Shadowbase Supported Platforms, Databases, and Operating Environments

HPE Shadowbase Streams for Application Integration

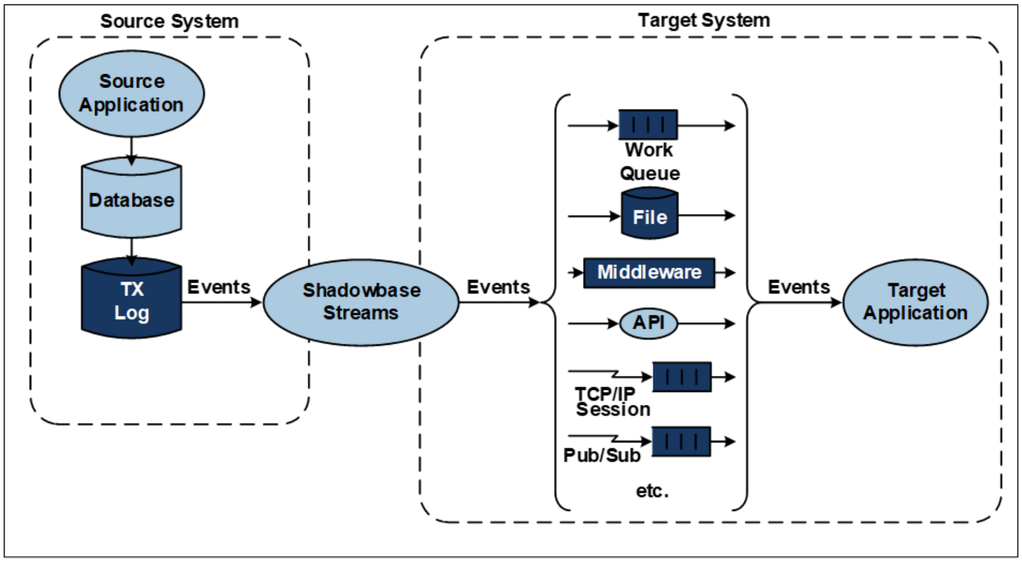


Figure 6 – HPE Shadowbase Streams for Application Integration

Benefits of HPE Shadowbase Streams

Using Shadowbase Streams to integrate applications provides many benefits to the enterprise, enabling the business value inherent in isolated applications to easily be unlocked and leveraged to deliver new business services, as an RTE.

- Shadowbase Streams is efficient. It is event-driven and acts only upon selected data-change events as they happen.
- Applications can react to events in real-time as they occur. There is no need to periodically poll for pertinent events, a process that introduces significant latency and overhead into event processing.
- Because events are processed in real-time, there is no more working with stale data. The reaction to events is timely and the data is always up to date.
- No application code changes are required to monitor and to deliver selected events to a target application. There is no need to build costly and complex agents and adapters.
- Shadowbase Streams is heterogeneous. It interoperates between a variety of applications, databases, and platforms from different vendors. A source system generating events can act as a hub. It can send events to multiple other systems. Each system can receive its own set of events formatted according to the requirements of its own target applications, on a schedule that matches the target application’s needs.
- Shadowbase Streams can publish events as they occur. Different target applications can subscribe to the subset of events that they would like to receive.
- Shadowbase Streams is flexible. It can be configured in a variety of architectures to provide a wealth of application integration options, and processing can be customized to meet specific application requirements.
- Shadowbase Streams supports both asynchronous and synchronous replication. With asynchronous replication, change data is sent to the target system after the changes have been made on the source system. In rare circumstances, it is possible for data to be lost in the event of a failure. For some applications lost data is not a problem, but for others, the data is critical and must not be lost. Shadowbase Zero Data Loss (ZDL), a future technology, uses synchronous replication to solve this problem. No data is changed on the source system unless the data has been safe-stored on the target system, ensuring no data loss, no matter what the failure. Asynchronous replication also allows for the possibility of data collisions,[4] which may be unacceptable for some applications.

[4] In an active/active architecture, a data collision is where the same data field is simultaneously updated on two (or more) copies of a database. When replication of that update between those copies occurs, each update will overwrite the other, resulting in an inconsistent database.

> Shadowbase synchronous replication also solves this problem with another future technology, Shadowbase ZDL+, preventing the data collision from occurring in the first place. Shadowbase Streams with synchronous replication is the solution for the most mission-critical applications, where data loss and/or data collisions cannot be tolerated.[5]

Data Transformation

Besides providing the distribution network to deliver change data in real-time to other databases and applications, HPE Shadowbase solutions include powerful capabilities to transform that change data into whatever format is required by the target database/application; the data may be aggregated, disaggregated, filtered, and transformed. The following methods are available:

- **SBMAP** – a scripting "language" that can be used to inform Shadowbase software how to transform source data into target data formats. SBMAP is powerful, sophisticated, and extensible.
- **SBDDLUTL** – a utility that reads an HPE Enscribe DDL record definition and produces a "flattened" (normalized) DDL structure along with an SQL CREATE TABLE statement for the selected target SQL environment. This capability simplifies the replication of unstructured Enscribe data into structured SQL databases. SBDDLUTL includes features to allow manipulation of the source fields when creating the target columns, including dropping and/or renaming fields, transforming field data, and normalizing the non-normalized Enscribe data.
- **User Exits** – enable the inclusion and execution of customized user logic (program code) at various points in the Shadowbase replication stream and provide capabilities that are more complex than the scripting language. User Exits are extremely flexible, enabling almost any kind of data transformation, and can also perform specific field/column-level encryption and data tokenization.
- **DBS Mapping** – a scripting "language" for target-side Other Server platforms, including these capabilities: drop all events for a target table; drop columns and/or certain events for a target table; convert updates to inserts; concatenate (text) columns; and reformat and convert/replace characters.

There are a number of parameter settings that can be set in Shadowbase software to assist with data transformation, (i.e., to convert non-printable binary data to spaces in a character field). However, these settings work on the data record at an aggregate level, and not at a field level. SBMAP and the other methods described above allow for much finer detail of data transformation.

Case Studies[6]

The following case studies of customer production deployments demonstrate how Shadowbase Streams for application integration may be used to integrate diverse applications. Shadowbase Streams is well-suited to this task. It is heterogeneous and is able to replicate data between a variety of systems and databases.

HPE Shadowbase Streams Enables Real-Time Fraud Management

HPE Shadowbase Streams Monitors Securities Prices for Wealth Management Clients

The brokerage operations unit of a major North American bank provides specialized trading services for its wealth management clients. It accepts securities trading orders from its clients and monitors several stock exchanges to track trading activities. By doing so, the brokerage searches for opportunities to maximize its clients' positions before making the trades.

The brokerage's trading and back office applications run on an HPE NonStop system. Shadowbase Streams monitors the changes made to the application's trading database and selects those changes that represent buy/sell orders. The orders are sent as order events to a work file that holds the order events for processing by the brokerage's Stock Watch System, a Windows application using Microsoft SQL Server. In this implementation, the Security Watch List Work File is implemented as an HPE NonStop Enscribe QUEUE file, allowing the Watch List Monitor Process to post 'read' requests that queue until an event is committed by Shadowbase Streams. This architecture avoids the file polling or messaging overhead that would otherwise be needed for Shadowbase Streams to alert the Watch List Monitor Process that additional work is available for it to process.

[5] For further information, visit Shadowbase ZDL and ZDL+.
[6] For further information, visit: Application Integration use cases and Data Integration use cases.

The work file is tracked by the Watch List Monitor Process that reads order events from the work file and feeds them to an IBM MQSeries queue. The Stock Watch System fetches the trade requests from the IBM MQSeries queue and makes trading recommendations in real-time for the wealth management clients based on current market activity.

Shadowbase Streams sends the trade requests to the Stock Watch System via the work file and IBM MQSeries because this is the way in which the Stock Watch application was originally implemented. Alternatively, Shadowbase Streams could have directly delivered the requests to the Stock Watch application and avoided queuing delays (and the cost of IBM MQSeries software licensing).

HPE Shadowbase Streams **Integrates Online Store Purchases with Suggestions for Upselling**

Summary

HPE Shadowbase Streams is a member of the HPE Shadowbase suite of products (built by Gravic, sold by HPE). Shadowbase Streams for application integration and for data integration provide the facilities for integrating existing applications, services, or database environments in order to create new functionalities for the enterprise, typically without the need to modify existing application code.

Using Shadowbase Streams, applications that once were isolated now can interoperate in real-time to provide services that were not possible before integration. Critical data generated by one application is distributed immediately to other applications or database environments.

Shadowbase Streams supports many models for data distribution, including maintaining a remote database of critical data, sending critical data to client applications directly via queues or client APIs, publishing data to applications that have subscribed to the data, or responding to poll queries from client applications.

HPE Shadowbase Streams modernizes legacy applications by integrating diverse applications across the enterprise so that new and valuable services that were not possible in the past can be created to enhance competitiveness, to reduce costs or increase revenue, to satisfy regulatory requirements, and to generally improve the user experience.

International Partner Information

Global

Hewlett Packard Enterprise
6280 America Center Drive
San Jose, CA 95002
USA
Tel: +1.800.607.3567
www.hpe.com

Japan

High Availability Systems Co. Ltd
MS Shibaura Bldg.
4-13-23 Shibaura
Minato-ku, Tokyo 108-0023
Japan
Tel: +81 3 5730 8870
Fax: +81 3 5730 8629
www.ha-sys.co.jp

Gravic, Inc. Contact Information

17 General Warren Blvd.
Malvern, PA 19355-1245
USA
Tel: +1.610.647.6250
Fax: +1.610.647.7958
www.shadowbasesoftware.com
Email Sales: shadowbase@gravic.com
Email Support: sbsupport@gravic.com

Hewlett Packard Enterprise Business Partner Information

Hewlett Packard Enterprise directly sells and supports Shadowbase Solutions under the name ***HPE Shadowbase***. For more information, please contact your local HPE account team or visit our website.

Copyright and Trademark Information

NOTICE: Shadowbase Zero Data Loss (ZDL) does not guarantee that you will not lose any data; all user warranties are provided solely in accordance with the terms of the product License Agreement. Each user's experiences will vary depending on its system configuration, hardware and other software compatibility, operator capability, data integrity, user procedures, backups and verification, network integrity, third party products and services, modifications and updates to this product and others, as well as other factors. Please consult with your supplier and review our License Agreement for more information.