White Paper

Inspiring New Patterns or Data Integration – Mixed-Time Federation and Virtualization

Pamela Szabó
Chief Technology Officer
Stone Bond Technologies
June 2012

The Inspiration

A new paradigm for agile data integration has given rise to formerly impossible architectures and solutions for data handling. One of these is now commonly referred to as “federation and virtualization,” which facilitates the ability to bring together disparate data meaningfully for end users to view and drill down. Even more powerfully, users can interact with the data so that changes are posted back to the original sources. As with all that is new, the reality of this “writeback” capability brings questions about the impact of this new paradigm on the corporate infrastructure. This paper presents some background on federation and virtualization’s opportunities, and describes one specific architecture pattern that addresses one of these points of discussion.

Forget EAI and ETL Patterns

The maturing of data virtualization for real-time interaction is bringing important fresh opportunities to light with respect to application and data integration. Business solution designers are discovering that they can think outside the proverbial box (“EAI or ETL?”) because many of the historic impediments of standard integration models have been removed. IT architects are translating those new business opportunities into streamlined architecture patterns, eliminating staging databases, and easing the amount of effort to make them a reality.

With live data federation now coming to the forefront, virtualization of those federated sources at endpoint applications means that end users are able to access data live from multiple databases and systems, aligned just the way they need it to make decisions or perform their specific duties. The idea of federation and virtualization is that data never actually moves anywhere.

For example, Enterprise Enabler® (EE), the integration system, is triggered by the end user upon opening or refreshing a web page. EE executes the instructions that were configured earlier for this particular use. The instructions are called metadata, and include everything necessary for the engine to reach into multiple systems, on premise or in the cloud (SaaS), extract specific sets of data, transform and align them into the format that is meaningful for that user or role. This data is sent directly to the screen without saving it anywhere. Once the screen is refreshed or the application is closed, it is gone. Nevertheless, the user can see and interact with this “virtual” data, drilling down to more detail, which also is accessed virtually.

Enterprise Enabler also provides the ability for that user to edit data on the screen and have it passed back and written to the originating application(s) in the correctly transformed format. This capability is called “Bi-Directional Federation and Virtualization.”

Inspiring New Patterns … and Fears

Having this technology available means that new use cases and solution patterns can be seriously contemplated for the first time. For example, historically, the only way to combine data from on-premise systems with SaaS data was to move the relevant on-premise data to the cloud, which can pose a formidable risk for an enterprise. Instead, with EE, the data remains at the sources, with no copies ever needed. Another important use case is the ability to define role- or user-based screens that constitute that person’s entire interaction with multiple backend systems.

A five minute out-of-box web part configuration in SharePoint, for example, eliminates the awkward log in and navigation within SAP, and aligns data from other systems on the same page, for full read and write. Any dashboard can immediately be converted from an information-only display to a console where the user’s data and decisions can be captured and fed back to the appropriate systems or databases.

On the other hand, one quickly sees that this new paradigm could open a "Pandora's Box" of things that keep architects, risk managers, data security and governance experts, and dinosaur programmers awake at night. As it turns out, the greatest concern, end users writing back live to the sources, is unfounded.

Actually, write-back is only enabled when specific fields of specific endpoint applications are configured for write-back. Further levels of control are accomplished by passing the user's credentials through to the source systems for authorization of all CRUD (create, read, update delete) functions. No worry in the end on this front.

Can this model bring backend systems to their knees?

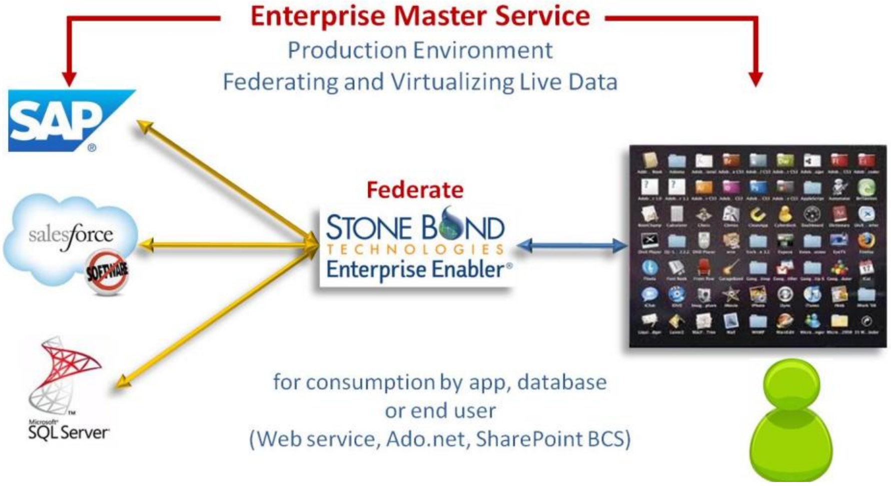
We now have a different perspective on live data for end users. With data federation, users are getting the current information from every source each time their screen is refreshed, as opposed to pulling data from a central repository such as a data warehouse. Data is always fresh and aligned, instead of yesterday's snapshot. As more users and uses are configured and rolled out, there are increased hits to the backend systems. Refresh speed is generally limited, not by the integration itself, but by the response time of source applications, and, as is the case with SharePoint, by the user interaction application screen.

Suppose you have 1000 users hitting your backend system via integration for federated virtual access and write-back. The good thing is that users are getting the most current data. The bad thing is that the backend systems may not be able to keep up. What is the best way to solve this dilemma? Use data from the data warehouse instead? Limit the number of users? The answer needs to be in the context of preserving the real-time federation and virtualization while accommodating the time-demand perspective.

Importance of the Time Dimension

The frequency of change of any particular set of data varies considerably. Some change constantly, real time. Some may only be updated at the end of the day. We also have weekly, monthly, and yearly changes. Incoming transactions must be captured real time as they come in. This is certainly not a new concept, and is key to frequencies set for data synchronization and data warehouse design and updates. It's not just about the frequency of change of the data, but sometimes the delta of change in a typical timeframe and, of course, as well as the criticality of having the latest value.

Enterprise Enabler addresses this issue by federating live or real-time data with data that doesn't change often or doesn't change value enough to be critical, by managing the latter with a scheduled or event-driven cache update.


Enterprise Master Service
Production Environment
Federating and Virtualizing Live Data
SAP
salesforce
SOFTWARE
SQL Server
Federate
STONE BOND TECHNOLOGIES
Enterprise Enabler
for consumption by app, database
or end user
(Web service, Ado.net, SharePoint BCS)

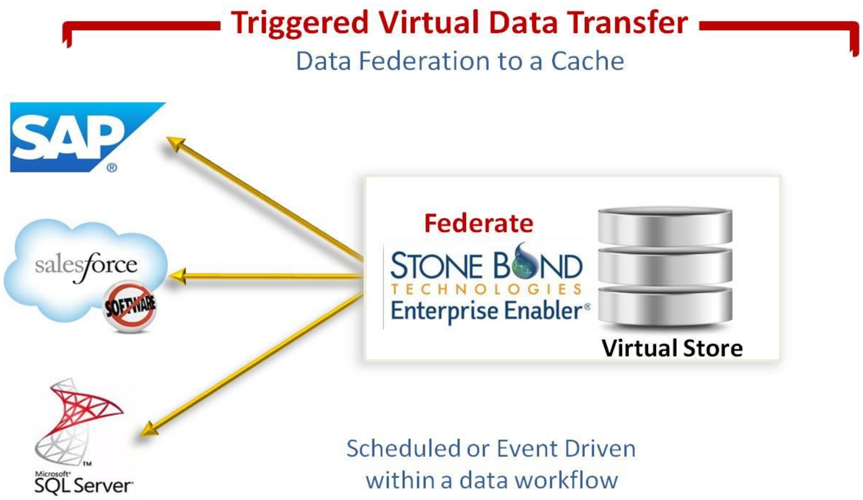
Typical Federation and Virtualization Pattern

Federation and Virtualization with Multi-Time Persistence

As described above, federation and virtualization taken together imply real-time data access, alignment, and virtual delivery. Using a quickly configured complex federation model that federates "on-the-fly" cached federated data with other live sources, Enterprise Enabler handles multi-time persistence.

The diagram on the previous page is of a typical federation that is made available virtually via multiple modes with Enterprise Master Services. In this case, data is being accessed live from SAP on premise, Salesforce.com in the cloud, and from an on-premise SQL Server database. When one of the consuming applications (e.g., SharePoint BCS, ADO.Net object read, or a web service) initiates the call, Enterprise Enabler reads the selected data from each source, aligns and transforms the data in native mode, returning it to the calling application. Full bi-directional CRUD functionality for write-back to the sources is available if configured for it.

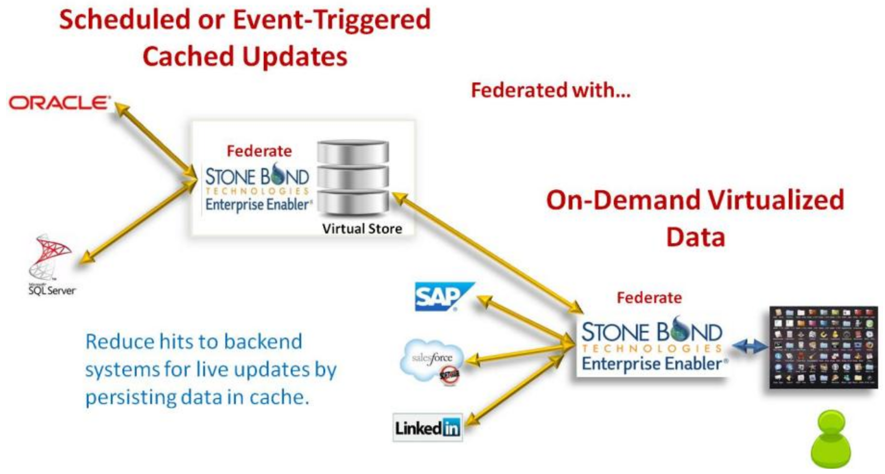
Using this pattern, each application is touched at each refresh of the screen (i.e., each invocation of the Enterprise Master Service). Many hits by many users can impact the performance of some backend systems.



Pattern for Persisting Data in Enterprise Enabler's "Virtual Store" Cache

The above pattern echoes a federated data load, without sending the data to a destination. Instead, the data is persisted in a cache for use as a source whenever needed. These are typically use-specific, as opposed to being designed for general use, as is a data warehouse. The data persisted in the Virtual Store is federated live by EE's transformation engine, to any schema desired. The Virtual Store then becomes a source to provide data to an Enterprise Master Service (pull) or to another triggered or scheduled integration pushing data to a destination application or database. In the design of any specific use of this pattern, the updates to various fields in the cache are made in optimum time frames or events for best "right-time" data persistence.

These Virtual Store definitions, along with the data access and federation, can be defined using Enterprise Enabler in a matter of minutes, and can include data validation "on-the-fly" as well as configurable notifications on any condition. In fact, all of the composite application building features of EE's Process Designer and engine (data workflow), are available in this pattern.


Scheduled or Event-Triggered Cached Updates
Federated with...
ORACLE
Federate
STONE BOND TECHNOLOGIES Enterprise Enabler
Virtual Store
SQL Server
On-Demand Virtualized Data
SAP
Federate
STONE BOND TECHNOLOGIES Enterprise Enabler
salesforce
LinkedIn
Reduce hits to backend systems for live updates by persisting data in cache.

Pattern for Federation and Virtualization with Multi-Time Persistence

In the pattern depicted above, the potential for impact on performance of backend systems is mitigated, while still taking advantage of live data federation and virtualization. This scenario persists data that doesn't change often in the Virtual Store, so that the backend is not hit each time an end user needs it. Data is pushed to the cache either at various intervals, as appropriate, or triggered by its change at the source. The sources for other fields are accessed and federated live upon the user's request. Secure write-back is handled inherently for the on-demand Enterprise Master Service, and is configurable for the persisted data.

This type of integration can be configured in a couple of hours using Enterprise Enabler. The rapid implementation time allows architects and analysts to focus on the business case and solution design, and not on the months of programming to put together a solution.

Conclusion

This model of "federated federations" with multi-time persistence is only one scenario for using Enterprise Enabler to federate and virtualize data to streamline corporate integrations. Looking forward, as the concepts are better understood, this new paradigm of live, bi-directional federation and virtualization will stoke the imagination and creativity of architects. Truly integrated agile businesses will gain competitive advantage and reduce overall IT costs as they leave the man-power-intensive EAI and ETL in their wake.

The information contained in this document represents the current view of Stone Bond Technologies on the issue discussed as of the date of publication.

This White Paper is for information purposes only.

Stone Bond Technologies may have patents, patent applications, trademark, copyright or other intellectual property rights covering the subject matter of this document. Except as expressly provided in any written license agreement from Stone Bond Technologies LP, the furnishing of this document does not give you any license to these patents, trademarks, copyrights or intellectual property.

Stone Bond Technologies, L.P.
1020 Main Street Suite 1550
Houston, TX 77002
713-622-8798