stone bond technologies

White Paper

Agile Integration Software Essentials

A model for next-generation consolidated integration infrastructure

An Enterprise Enabler® white paper from Stone Bond Technologies

Stone Bond Technologies, L.P.

1021 Main Street Suite 1550

Houston, TX 77002

713-622-8798

www.stonebond.com

Title

It's all about the underlying philosophy:

Ensure that the technology behind the scenes automates virtually everything that is done more than a couple of times by a person configuring integrations.

Primary Indicators of the possibility of being an AIS

1. Native Transformation Engine:

It must transform data a) from multiple disparate sources at once, and b) in the native source formats.

2. Single IDE:

A single integrated Development Environment must cross the entire scope of functionality for all integration patterns and must incorporate the run-time engines.

Stone Bond Technologies, L.P.

The Compelling Need for Agile Integration

We live at an ever-increasing rate of change, particularly in information technology. Just twenty years ago cell phones barely existed, and it is clear that cell phones have made a huge impact on corporate success. If there are any companies not partaking, they are certainly not competing. How can CIOs think their business should still be run on the integration products of 20 years ago just because big software companies provide them? Why do they cling to the ballast of decades-old technologies? Agile integration is not just compelling or interesting. In a very short time businesses that cling to ancient infrastructures with their layered façades of agility will not be able to compete with truly agile competitors.

Agile Integration Software has come of age and is being adopted by companies and CIOs that have grown weary of the classic products and technologies that with every new feature become more complex, require more specialized skills, and take longer and longer to yield results. They are freed up to leverage agile integration infrastructures that in turn dramatically reduce Tech Debt and open new opportunities.

Why Does Agile Integration Seem So Elusive?

The important shift in thinking about what constitutes Agile Integration is that it's not about the features that an integration platform has; it's about what's missing. It's the completeness of the solution. That is: "Is anything critical NOT present?" Looking at it from that perspective, the trite and simplistic car analogy comes to mind. You have to have four wheels, an engine, and some kind of fuel if you are going to have a working car. If just one wheel is missing, it can't operate like a car; if there's no engine, it won't operate like a car. If there's no fuel, it won't do the job of a car. So it's not a car.

We have tended to list the features of AIS, and while several integration platforms boast some or even most of these features, the perspective of having a short list of absolutely essential capabilities or features, can quickly eliminate the poseurs.

The Quick Test for Agile Integration Software

It's pretty easy to determine a quick first pass using the two primary qualifiers.

1 **The product must have a transformation engine that aligns and transforms data a) from multiple disparate sources at once and b) in their native formats.** Without this, complex integrations get little assistance from the platform itself, but are accomplished by extensive custom coding. A streamlined, high-performance data virtualization solution is impossible. Writing back to the sources becomes cumbersome at best, and live, real-time end user interaction with the endpoint simply cannot be effective. Older one-to-one transformation engines do not satisfy the needs; XSLT transformation engines also do not meet the two criteria, because all data must be converted to XML before it is transformed, and the XML output must then be converted to the destination format. Each of those conversions, to and from XML are effectively additional full transformations that generally must be accomplished with custom coding.

2 **There must be a single Integrated Development Environment (IDE)** that crosses the entire scope of functionality for all integration patterns. This IDE must incorporate the run-time engines in order to be able to design, develop, test, deploy, and monitor in the same environment. Leaving the environment for anything dramatically reduces the speed of implementation and of change, which is essential for agility, time to value, and minimizing tech debt.

If just one feature is missing... you can't have AGILITY

	IDE	EAI/ETL/ SOA/DV	App-Comms	Transfor-mation Engine	Virtual Federation	DV with writeback	Data Workflow	Change manage-ment	Impact	Why?
Single IDE (with runtime engines)	✗	✓	✓	✓	✓	✓	✓	✓	5X - 10X increase in implementation time; Instant tech debt	Need to move among different tools, leave and set up separate environments for testing
EAI/ETL/SOA/DV	✓	✗	✓	✓	✓	✓	✓	✓	No reuse of metadata and rules across patterns. More development time, maintenance, tech debt	No sharing of metadata across patterns. Must learn multiple tools, redundant coding, configuration, installation
Intelligent Endpoint Connectors (AppComms)	✓	✓	✗	✓	✓	✓	✓	✓	No schema discovery. No full CRUD. Also the inability to switch control at run time with tranformation engine	Only intelligent connectors discover schemas even in customized applications.
Native Transformation Engine	✓	✓	✓	✗	✓	✓	✓	✓	More time configuring, custom programming, more processing at run time	Must convert all data to some central format, e.g., XML, before transformation, then from XML to destination format
Virtual Data Federation (for all patterns)	✓	✓	✓	✓	✗	✓	✓	✓	More design, configuring, maintenance, tech debt. Write-back may not be possible	Staging required. What could be one transformation becomes multiple transformations
Data Virtualization with Writeback	✓	✓	✓	✓	✓	✗	✓	✓	No possibility of interactive use of data virtualization	Cannot support end user input from endpoint application to feed back to sources
Complex Data Workflow Logic	✓	✓	✓	✓	✓	✓	✗	✓	Cant do realisitic event-driven ETL scenarios.	No validation, error handling, parallel data flow
Pro-active Monitoring for Change	✓	✓	✓	✓	✓	✓	✓	✗	Risk of catastrophic impact from changes in endpoint schemas. Risk whenever reusable metadata is modified	Not taking advantage of the patented pro-active metadata monitoring built into Enterprise Enabler
With All Features	✓	✓	✓	✓	✓	✓	✓	✓	**Agile Integration Software** (Enterprise Enabler®)	

What about the Cloud?

Any platform that meets the other requirements can naturally be hosted in the cloud. Transaction-based usage can easily be captured by customer. Security considerations are handled within the product. Contrary to some promotions, the fact that an application is hosted on the cloud does not imply agility.

What about Big Data?

Big Data is one of the beneficiaries of Agile Integration Software, as the efficiencies it produces allow for scalability and high speed processing. In fact AIS is the platform of choice to merge, align and prepare data, and send it to a Hadoop or other specialized Big Data consumer.

What about Master Data Management (MDM)?

MDM is a natural extension of AIS, since it is a single place where reusable components of all kinds are managed. Enterprise Masters, the data bi-directional data virtualization services can be cataloged as MDM.

Now, if it Passed the Quick Test, Look in More Detail

There are more capabilities that must be present to ensure that you will not be surprised with a roadblock instead of smooth sailing.

1 Single, Comprehensive Integrated Development Environment (IDE)

Any person with a role in developing, deploying, monitoring, or modifying integrations should be logging into a single tool that allows them to do everything they need without ever leaving the environment.

Value

There is a significant saving of up to 90% in time and related Tech Debt to implement and maintain the integration. It also means that any point within a data flow is aware of and can be influenced by the status or operations elsewhere. This holistic view provided at design time is also carried through at run time, resulting in an agile "fabric" that enables secure free flowing of data and awareness of state throughout the enterprise.

Without It

In the absence of an IDE, an integration requires using a multitude of independent tools that do not share metadata, promoting a disconnected collection of independent integration components. Developing a solution means leaving the development environment(s) and setting up separate test scenarios, and going back and forth as the debugging is accomplished. Specialized skills are necessary for using some tools. In the absence of an IDE, Tech Debt abounds.

2 All integration patterns: EAI, ETL, SOA, DV (Data Virtualization)

Value

By handling all patterns, there is no need to first determine which tool to use, or in many cases to definitively determine which category fits the work you want to do. Almost all applications of ETL, for example, benefits from virtual data federation, which eliminates staging and all the overhead associated with it.

Without It

Before you do anything, you must make a decision as to which category your integration falls in. Often it is not as clear as you would like. If your solution requires any features of other patterns, more than likely they will need to be handled by custom coding. Once you make the determination, you will use the tool that is designed for that particular pattern. For example, if it looks like a data virtualization problem, you select the DV tool. Too bad you will not be able to leverage that work to use the same business rules for an ETL data load, or share the metadata and validation rules. You'll have to learn multiple tools and maintain multiple skill sets. This kind of environment is prohibitive to establishing an agile overall integration infrastructure design.

3 Intelligent Endpoint Connectors (AppComms™)

Intelligent Connectors differ from classic adapters in that they do not take data from sources and convert it to the format and mapping required by the destination. Instead, they have a layer that has intimate knowledge of how to interact with the endpoint data source, discovering schemas even for instances of the endpoint that have been customized. They know how to read, write, update and delete data in that endpoint, how to handle security, and how communication with the endpoint can be optimized for different situations. Beyond that, AppComms know how to interpret all of this and hand off in tandem with the transformation engine as it orchestrates across multiple sources and destinations.

Value

This type of intelligent connectivity accelerates the development time and reduces the level of knowledge required to configure integrations by presenting them with options and the particulars of the specific instance of the endpoint that they will be working with. At run time it enhances performance by working seamlessly in lock-step with the transformation engine.

Without It

You may have multiple steps to program in order to get the source data into a central format and then back out. If there are custom field that have been added, you will need to modify whatever the assumptions are about the endpoint. Performance is impacted and flexibility for things like end user awareness are not possible.

4 Native Transformation Engine

A native transformation engine interacts with the endpoint applications in their native formats, using a separate reusable connectivity and discovery layer that knows how to perform complete CRUD (create, read, update, delete) functions. The transformation engine orchestrates across these Application Communicators to access data from the multiple sources as necessary to federate, filter, and transform the data according to the configured rules, then orchestrates via the destination AppComm™. All of this is done in a single efficient streamlined execution.

Value

The streamlined architecture of data merging and transformation inherently delivers optimum performance. Additionally, with fewer moving parts, it greatly reduces the time and complexity of configuring integrations and of modifying them later.

Without It

Data transformation is done using ad hoc custom code, a point-to-point transformation engine that cannot handle multiple sources without staging, or all source data must first be transformed into a standard format such as XML before it can be processed by the transformation engine. Then it must be transformed from the XML to the destination's required format. This is a huge Tech Debt producer. All the tech debt you accumulate works against agility in both the short term development investment and the long term difficulty of applying change.

5 Virtual Data Federation (for all patterns)

These days when you hear about virtual data federation it is in the context of the new Data Virtualization space, where data is pulled directly from the sources, aligned, and delivered on-demand to a calling program. The same virtual federation is needed by all but the simplest integration problems, like synchronization, where data is physically moved without manipulation.

Value

Agility is about change, about changing data, and about rapid implementation. The requirement of aligning and transforming data from totally different sources is the reality of nearly all integrations. If you can do that very easily, without defining a data model and without staging the data anywhere, you eliminate latency and always have the most current data for that solution. This is desirable for ETL, DV, SOA and MDM services, as well as EAI. The products for older patterns do not generally incorporate virtual federation in the toolsets without considerable effort.

Without It

How agile can your final solution be if you have to align data from multiple sources by putting it first in a staging database? The data is all transformed going in, and then transformed again going out. Implementation is tedious and time consuming, and support over time is cumbersome. Data cannot be delivered live, so there will always be some amount of latency. Without virtual federation, agility is handicapped.

6 Data Virtualization with Write-Back

Standard Data Virtualization delivers federated data live on demand. The third wave of DV, as embodied Stone Bond's Enterprise Enabler, is also end-user aware and delivers end user input directly back to the sources live. Transaction rollback and security around reads and writes is imposed.

Value

The write-back capability for Data Virtualization turns the standard read-only reporting and BI uses into interactive consoles where end users can work with data that is accessed live from multiple completely different sources, aligned meaningfully, and presented via (for example) a browser screen. The user updates, corrects, or adds data that is sent directly back to the sources.

Without It

Data Virtualization is for reporting only.

7 Complex Data Workflow Logic

Historically associated with ETL patterns, data workflow logic establishes when and under what conditions integration is executed. In reality a powerful Data Logic Workflow engine is a composite application that drives potentially complex combinations of data flows, data cleaning, notifications, error handling logic and most anything else you can think of. In an Agile Integration Software, the singularity of the IDE and the interplay of the engines mean that there is a powerful awareness across the entire integrated environment through visibility and shared global variables.

Value

Unlike the typical manifestation of any of the integration patterns, Agile Integration heavily leverages the data workflow throughout the solution. The same metadata that is used for a DV can be leveraged for an ETL, and the workflow can be used to trigger the flow. These reusable processes can be inserted anywhere in any pattern, for cleaning, notification, special processing, and many other things.

Without It

Without a Data Workflow engine, it is impossible handle the range of demands of complex integrations. Inserting logic at various places in a data stream would otherwise be done with custom code. do ETL, which is a requirement for AIS. One simply cannot do ETL, which is a required capability of AIS, without a Data Workflow engine.

8 Pro-active Monitoring for Change

This can be thought of as the cornerstone, or perhaps the guardian of agility. Agility is about change; change is inevitable; change is the future. Enterprise Enabler's patented Integration Integrity Process monitors and manages change in metadata/integration components during design/development time and at deployment, confirming no conflicts. It monitors external endpoints for schema changes that are touched by integrations. It handles detection, impact analysis, notification, and reconciliation.

Value

Changes can be applied without lengthy study of potential impact. Reuse of metadata is encouraged and safe even with changes applied. Potentially catastrophic impact of changes in endpoint applications and data bases are averted. A watch for new fields in an application means you will know immediately so you can take advantage of new information if desired.

Without It

Rarely will anyone admit it, but catastrophic effects of undetected change in a runtime environment do happen and take often take days to determine the root cause. When developing, fear of changing metadata detracts from maximizing reusability. You won't know when new data is available that you may benefit from. When deploying new integrations, you must have time-consuming mandatory processes and cross-checks before rolling out any integrations.

Conclusion

When all these capabilities characterizing Agile Integration Software are met, integration no longer is daunting. Instead of the expensive limiting critical path factor, it becomes a true enabler of every IT project. Your whole IT department will double their productivity.

Architects can stop struggling with decisions of tradeoffs across options, each of which incurs overhead of different forms.

The CIO can think about new ways to help lead the company in responding to opportunities and market changes.

The State of Agile Integration Software

In the course of researching and preparing this white paper it has become clear that there is not really a class of Agile Integration Software. Surprisingly, and lamentably, Stone Bond's Enterprise Enabler software is the only product on the market that meets the criteria for AIS. Without at least a handful of products on the market, the awareness an market definition falls to a single player.

About Stone Bond

Stone Bond Technologies, L.P. is the provider of Enterprise Enabler® ,the leading business integration software for data virtualization, federation and orchestration delivering the industry's fastest time-to-value. As the only integration solution to scale both up and down to meet business needs, Stone Bond provides software solutions that help businesses improve their bottom line by streamlining interaction among business systems. To learn more about Stone Bond Technologies please visit, www.stonebond.com.

