

# A Business Guide to Enterprise Mashups

JackBe Corporation

April, 2008

## Table of Contents

Everything is Personal.....	1
Personalization Enablers.....	2
An Architecture for Personalization .....	3
Defining a Mashup.....	4
Enterprise Mashup Principles.....	5
The Enterprise Mashup Ecosystem .....	6
Mashups: Making SOA Business-Relevant.....	7
Mashups: Making Portals Dynamic .....	8
Mashups: Making RIA Development Smarter .....	8
Case Study: Enterprise Mashups in Action .....	8
JackBe's Presto Solution for Enterprise Mashups .....	9
Mashups Are About Business Results .....	12

# A Business Guide to Enterprise Mashups

## Everything is Personal

About a hundred years ago, Henry Ford began mass producing cars and famously said “You can have a Model T in any color, as long as it’s black”. Thus began the automotive mass production era at the expense of little or no consumer personalization. At the time, this made perfect sense. To mass produce you had to keep costs down by addressing the needs of the masses and not the needs of the individual.

Fast forward a hundred years and we have an automobile industry which prides itself on consumer customization. You can buy a car in a vast array of colors, with countless features to choose from. And it won’t be long before you can design your own car that painted in your own custom created color and styling.

Imagine a short way into the future. You see a commercial about the Lexus *Web Series* automobile. You can create your Lexus by creating a completely custom color, and choose the styling from over ten thousand different body styling permutations. You create your custom car where no two cars look exactly alike. You do this from the comfort of your Web browser.

Unfortunately, radical personalization and mass production are at odds with each other. For now, you have to find a good body shop to paint a custom color and create the custom styling. In other words, if you want real, intimate personalization, you need to do it yourself.

The IT industry isn’t much different from the automobile industry. IT enables big business processes, large integration projects and features that target the many, not the few. They target the masses and not the individual. Why? Like the automotive industry, the cost of supporting individual needs makes it prohibitive.

But we are quickly becoming a society where everyone wants and expects complete customization. We are the Do-it-Yourself (DIY) nation. And this demand isn’t limited to consumer goods. Consumers are taking their DIY expectations into their enterprises. Corporate employees are demanding customization even if they have to ‘DIY’.

There is, of course, a right way and a wrong way to DIY. Practically everyday we perform DIY 'integration' by taking data from big ERP and CRM systems, email, documents, and web sites and copy, paste, and edit this data manually so we can make more informed decisions. Then we 'collaborate' by emailing the spreadsheet to anyone interested and perpetuate the copy/paste/edit cycle.

We have taken DIY into unproductive territory. We've reached the limits to what we can do this way. In a March 2005 report, 'The Hidden Costs of Information Work', IDC noted that an organization with 1,000 employees spends over \$10 million annually in reformatting and recreating information. And as the Web is bringing business data to our fingertips at an increasingly rapid rate, we are now forced to continue to integrate more with less, collaborate and make decisions in a fraction of the time with vast amounts of more data. We've reached the breaking point.

## Personalization Enablers

The business user is demanding IT open their IT assets to customers and partners to create an open and accessible environment the same way the public Web provides consumers. But, wait! Enterprises can't do that. They can't just open their most cherished assets to the world. In fact, they can. A combination of new technologies, ubiquitous Web access, and open standards have created an opportunity to empower end users to address their information needs. And enterprises *can* make use of these trends in a secure and governed manner so customers and partners can benefit from the assets as much them. In the enterprise the IT walls are falling.

In 2004 the world was stunned by the unique interaction capability in Google Maps. For the first time, we were able to drag around a map in a Web browser and have it behave much like a desktop application. The Web page didn't go away, it *stayed*. Google Maps became hugely popular not because of its amazing capabilities, but because it was so natural to interact with a Web page the way we interact with desktop applications. This

### RIA: Rich Internet Applications

'RIA', or Rich Internet Applications, is the umbrella term for the browser technologies that provide desktop-like capabilities within a browser. Ajax (Asynchronous Java and XML) is a set of standards-based technologies built into all current generation browsers that can be used to build RIAs; Ajax is open and not owned by anyone. Other technologies can also be used to create RIAs. The granddaddy of RIA, Flash/Flex is proprietary and owned by Adobe. New-comers include Microsoft's SilverLight, Sun's JavaFX and Adobe's AIR. All of these are proprietary, but represent innovation and advancing capabilities in the browser.

### SOA

'SOA', or Service Oriented Architecture, is not a technology but an architectural style. SOA is a server-based architectural style which embraces loosely-coupled standards-based 'services' that encapsulate business logic and/or data. SOA services can be combined to support new business functionality. While many SOA standards have been proposed and/or approved, common standards include RSS, REST, WSDL, SOAP, UDDI and WS-I.

was the coming-out party for Ajax and the beginning of the Rich Internet Application (RIA) technology trend (see the 'RIA: Rich Internet Applications sidebar for more information). The subsequent explosive growth in RIAs/Ajax confirmed an unmet demand to have browser applications behave dynamically like desktop applications.

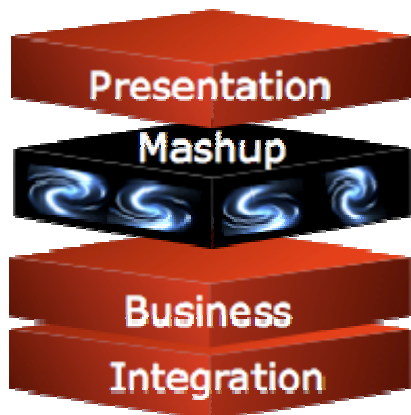
But users needed more. They wanted the same level of access to corporate data as they get from Web data. And the Web was built to provide fast, simplified data and content access. To meet this demand, a standards-based architectural approach, Service Oriented Architecture (SOA), emerged to provide the backbone for web-based and datacenter-managed data and applications (see the 'SOA' sidebar for more information). While SOA initially focused on intra-application integration, today it is rapidly focusing on exposing functionality to end users.

It's this combination of RIA, providing desktop-like interaction, and SOA, providing standardized access to business functionality and data, that forms the foundation of a solution for radical business personalization. But, as the saying goes, 'bricks don't build a house by themselves'.

## An Architecture for Personalization

Enterprise software is and will always be about sound architecture. And as all good architects know, whether working in bricks-and-mortar or software, layers make for a sound architecture. Today's IT architectures are commonly grouped into three tiers:

Presentation, Business and Integration.



The Business tier, being responsible for the business functionality necessary for the presentation tier, sends and receives data to/from the Integration tier and provides the necessary business functionality and data transformations required for the application in the Presentation tier. The logic required to transform the data in the Business tier is known as 'black-box' integration code. This code is specialized, time consuming to create, and usually tied to a specific application.

When IT decides when to integrate new services into Business Tier, they do so with a significant amount of planning, development, integration and testing. These so-called "big" IT integrations extract, transform, normalize and route data using sophisticated tools like ESBs, BPMs and BPEL engines. As you may imagine, developing and maintaining the integration code becomes increasingly expensive and time-consuming.

Today's dynamic users cannot wait or afford black-box integration. A sound architectural solution to this problem is to provide a new tier to accommodate the user's increasing ad-hoc and dynamic data integrations needs. This new tier sits between the Presentation tier and the Business tier, allowing users to take advantage of RIA/Ajax and SOA technologies for DIY solutions in an enterprise-friendly manner. When deployed in this way, we have taken a big step toward Enterprise Mashups.

## Defining a Mashup

The Wikipedia provides contains the following definition of 'Mashups': '*A mashup is a website or application that combines content from more than one source into an integrated experience*'. This is a useful basic definition, but let's use a more complete enterprise-relevant description:

*A mashup is a user-driven micro-integration of Web-accessible data.*

While short, this definition contains a number of important points worth considering;

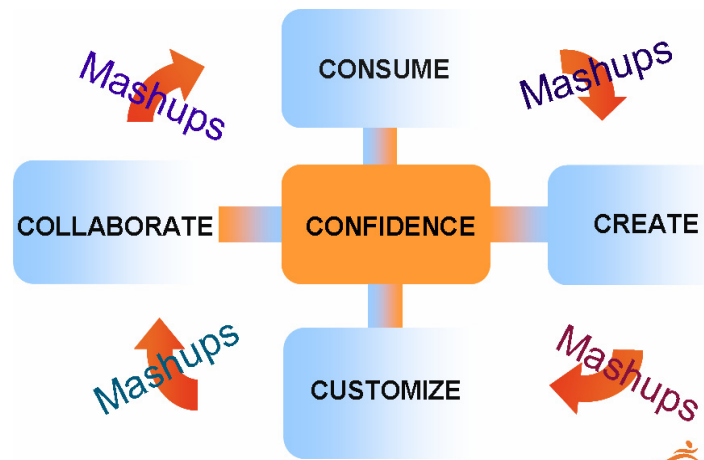
- **“User-driven”** – Mashups are executed for the user, not the by black-box back-end integration systems such as ESB, BPM, BPEL, etc. In this sense, a mashup must be completed by the users themselves. Without this guiding principle, we are merely sending the users back to IT for more development.
- **“Micro-integration”** – Think of a user taking data from multiple sources and copying it into Excel. While these workers are really good at know how to combine data, they are often frustrated with the time and effort it takes to accomplish this manual feat. And as these users typically deal with small amounts of knowledge-oriented information (as opposed to IT-managed applications that typically deal with large amounts of transactional information), these are called “micro-integrations”. There are five key micro-integration patterns worth noting, examined here through simple examples:
  - **Data merging:** Merging multiple RSS feeds into a single feed.
  - **Data feeding:** Feeding a customer list in Salesforce.com into an SAP order system to see pending orders for those customers.
  - **Data joining:** Joining the top consumer electronic stock gainers with market data to identify stock fluctuations according to product sales.
  - **Data filtering:** Getting a list of customers who have offices in a specific zip code.
  - **Data annotations:** Adding latitude/longitude to a customer list to plot main offices on a map.
- **“Web-accessible”** – Returning to the concept of RIA/Ajax and SOA, mashups are built upon data that could reasonably be displayed quickly in a web browser or, more precisely, data that doesn't require too much manipulation for the user to make sense of it. Standards-based interface/communication technologies such as WSDL, REST and RSS can help make this possible.

These elements describe what a mashup is but not its *usage*. That is left to the user, whether that user is an intelligence analyst performing an evaluation of a terrorist hotspot or a securities trader completing an analysis of an interesting market. More importantly, the way a user interacts with a mashup makes it distinct from IT-centric integrations. Users *dynamically* create and interact with mashups. The net effect is that IT doesn't prescribe the integration, they only need to govern it.

## Enterprise Mashup Principles

To effectively adopt enterprise mashups, organizations also need a better way to identify and categorize all of the important capabilities that enterprises typically require of their enterprise mashup software. To help identify these capabilities, there is a simple but powerful set of principles called the “Five Cs”:

- Consume** - Users need to consume public and private Web-enabled SOA-style services and mashups on demand. The minimum set of consumable SOA-style services should include WSDL, REST, RSS and SQL Databases.
- Create** - Users need to create new enterprise mashups from existing SOA-style services and mashups, preferably in a visual editor. Web 2.0 style tagging and metadata search should also be supported.
- Customize** - Users need to customize (e.g. filter, annotate, etc.) existing mashups and create variants which can be published visually in standardized user-interfaces such as portals. Versioning of mashups is also preferred.
- Collaborate** - Users need to tag, describe, publish, and share their mashups with others in their community.
- Confidence** - Users need to be confident that all mashup consumption, creation, customization and collaboration occur in a secured and governed environment. This includes proper authentication and authorization of access to all mashups. All enterprise application security requirements that have been established by IT must be adhered to.



The fifth “C”, Confidence, is what truly differentiates *consumer* mashups from *enterprise* mashups. Unlike browser-based consumer mashups, enterprise mashups must run in the datacenter to provide the necessary security, governance, performance, and

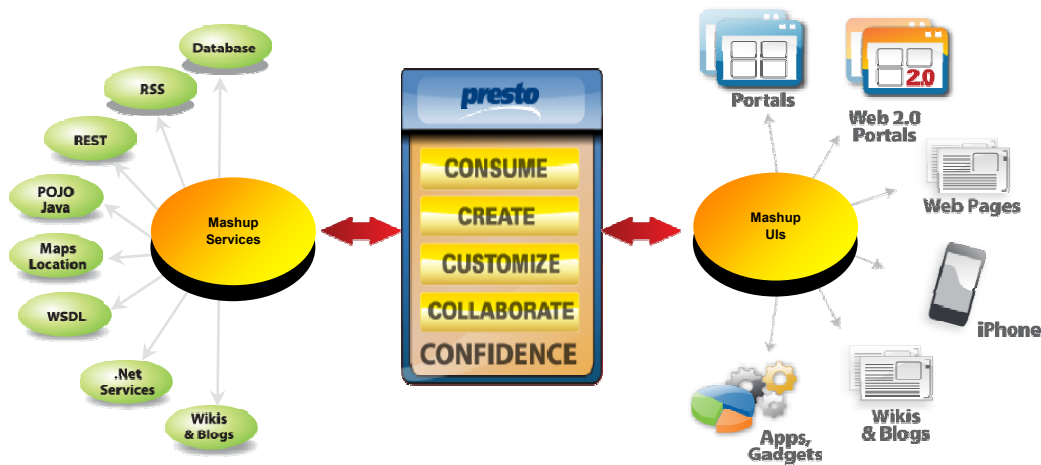
scalability necessary for enterprises. IT's security and governance infrastructure must provide the necessary entitlements to meet the enterprise's security and governance standards, governing and securing user's access to all mashups and to underlying services that are within the enterprise. And because the "sea of services" or "enterprise service cloud" include third-party services (like partner's and public services), the services must be governed at the access point versus within/at the service itself. Therefore, IT must have a secure extra-enterprise mashup management infrastructure and tools which allows them to govern the Five Cs, much like many enterprises do for their Service Oriented Architecture initiatives.

## The Enterprise Mashup Ecosystem

With a basic understanding of mashups and where they fit within the typical enterprise IT architecture, we also need to consider that mashups certainly do not live alone. A complete enterprise mashup ecosystem can be described that includes relevant third-party actors and important elements of the mashup solution itself. The five significant elements to an effective enterprise mashup ecosystem include:

1. Web-enabled source services (WSDL, RSS, REST, SQL Database, etc)
2. Mashup Services
3. A Mashup-enabling platform
4. Mashup interfaces (RSS, mashlets, etc.)
5. Consuming presentation platforms (Portals, Wikis/Blogs, Web pages, etc)

A conceptual mashup ecosystem would therefore look like this:



These components working together create a complete Enterprise Mashup Ecosystem. Just like the World Wide Web, the Enterprise Mashup Ecosystem is made up of suppliers of data, consumers of data and applications.

Most notable among these ecosystem components are SOAs, one of the most active suppliers of services to be mashed, as well as Rich Internet Applications (RIAs) and portals, active consumers of mashups. These are three areas where mashups can bring tangible value to already productive IT initiatives. Each is worth considering in detail

### ***Mashups: Making SOA Business-Relevant***

SOAs can be greatly enhanced with judicious use of enterprise mashups. SOAs live in the proverbial basement, with seemingly little relevance to the day-to-day issues of the enterprise executive. Mashups can bring SOA to the business user, unlocking the vast potential of SOA-driven process and SOA-tapped data sources.

As a result, the ROI of SOA is difficult, at best, to define and measure. The press and blogosphere is filled with SOA implementers/analysts discussing the ROI of SOA and the idea that stand-alone SOA efforts are DOA? Look no further than the [recent commentary](#) from SOA expert David Linthicum, the [Nucleus Report on SOA ROI](#), and the subsequent commentary from ZDNet's [Joe McKendrick](#) and IT advisors [Neil Macehiter and Neil Ward-Dutton](#).

While these experts differ on issues like the importance of SOA ROI, how to calculate SOA ROI (if at all), and why we don't have more/better of it, they all seem to agree on one thing: 'Enterprise-wide support for SOA hinges on the ability to demonstrate value to the business at large — more growth, revenue opportunities, and all that good stuff.'

SOA is plumbing. Nice, shiny, efficient plumbing to be sure, but still plumbing. And your average business dude/dudette (think sales manager, marketing director, finance officer, or customer support rep) could care less about it. In fact, if they think about it all, they probably just hope it stays right where it is: out of sight and running quietly. These same business folk probably appreciate the marble floors, wood-paneled doors, and brass fixtures that surround this plumbing much more. In other words, they like that bit of 'stuff' that actually frames the plumbing and brings it to life.

SOAs need to change this inward-focused quality. To paraphrase Macehiter/Ward-Dutton in their recent note, '[More big vs small thinking: SOA vs BPM](#)', IT must focus on where the real *business value* of SOA lies. That means it needs help. Macehiter/Ward-Dutton point out that BPM can help distill some SOA value up to the business level. And as one of the early implementers of Enterprise Ajax, JackBe knows from extensive first-hand experience that Ajax makes a great service consumer.

To this collection of SOA-complimenting tools we must add the enterprise mashup. JackBe has found in its enterprise mashup implementations that they can actually drag the SOA out of the proverbial basement and onto the end-user's desks. It's not only highly visible, but it's user-driven, giving IT a way fulfill the promise of SOA and enhance that elusive SOA ROI.

There are a number of very real synergies from the mashup/SOA combination. Just a few of them include:

- Mashups can help create normalized 'virtual' services from sources that haven't been 'SOAed' yet. It's no secret that SOA efforts can take years. Until the formal SOA magic has been applied, a quick, standardized service can help users get started earlier than otherwise.
- Mashups let users 'right-size' the granularity of services. Now IT doesn't have to guess/study/analyze whether a service offers data that is 'too specific', 'too general', 'too dated', or 'too cold'.
- Mashups let users share their resulting services, making them a part of the service-generating network. Now IT doesn't have to do it alone.
- Mashups let end-users visualize the SOA in graphs, charts, tables, maps, etc. Instead of hoping the aging corporate portal has a place/way to get services visualized in the way(s) the users want, users can each do it themselves to meet their own ever-changing needs.
- Mashups let users join in data from outside the enterprise. Today's SOA efforts are largely inwardly-focused. But users often want to include external data in their work. Mashups don't care and good mashup software makes the actual location of a data service irrelevant.

### ***Mashups: Making Portals Dynamic***

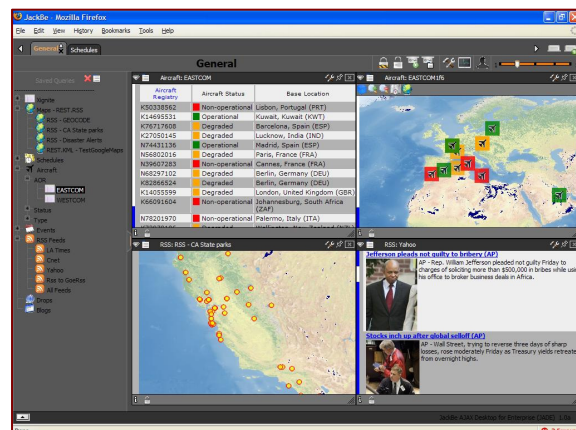
Mashups can portals dynamic, like the real world. Original portals were built for content. Mashups can bring a dynamic, data-based, application-like, user-driven capability to any portal. This column/article discusses the basic premise and an architectural overview of such a solution, including relevant standards and pitfalls.

### ***Mashups: Making RIA Development Smarter***

Unsurprisingly, Rich Internet Application (RIA) toolsets are great at 'display' and lousy at 'data'. With a proper approach mashups can deliver better data faster to the RIA. This column/article discusses the basic premise and an architectural overview of such a solution, including relevant standards and pitfalls.

## **Case Study: Enterprise Mashups in Action**

Nothing helps understand the impacts and benefits of an emerging technology like a real-world production example. Luckily, the US Defense Intelligence Agency (DIA) has taken a futuristic approach to intelligence gathering powered by enterprise mashup technology.



The Defense Intelligence Agency is a Department of Defense combat support agency and an important member of the United States Intelligence Community. With over 11,000 military and civilian employees worldwide, DIA is a major producer and manager of foreign military intelligence. The DIA provides military intelligence to warfighters, defense policymakers and force planners, in the Department of Defense and the Intelligence Community, in support of U.S. military planning and operations and weapon.

In today's intelligence community the need to be technologically driven correlates directly to the centralization of information among various government agencies. The need for rich and effective collaboration and integration solutions that enable communication flow throughout government departments are essential to building mission critical applications. Based on these needs, the DIA deployed 'Overwatch' (see picture), a Virtual Operating Center application that integrates multiple intelligence sources into a desktop like intelligence asset dashboard for the real-time analysis of data. Under time constraints, DIA needed a fully functioning technological platform that would present a browser based interactive application complying with government security standards.

A typical intelligence-gathering process begins with a user selecting an Area of Responsibility such as a Combatant Command, which limits all intelligence to that specific AOR. The user then selects the type of intelligence such as a facility, personnel, naval vessel, aircraft, event or travel, to further limit intelligence to that specific type. Additional filtering can be specified depending on the type. Once filtered, the resulting intelligence asset can be updated and comments made for rapid information sharing to occur. Users can save and share sets of information with other secured, authorized users.

The DIA chose a JackBe-based enterprise mashup solution as it eliminated significant development and deployment time. Overwatch was designed to address the DIA's information gathering and sharing challenges by empowering the end user to quickly paint a picture of situational awareness across various intelligence data sources, using a paradigm of drag-and-drop and bookmarking that is used for future private briefings.

## **JackBe's Presto Solution for Enterprise Mashups**

JackBe has developed an enterprise mashup platform, Presto, to deliver on the principles and architecture of enterprise mashups. The Presto Platform empowers application developers and users to create, customize and share enterprise Mashups for faster decisions and better business results. Presto provides a dynamic integration approach to building enterprise mashup applications faster, leveraging internal and external data, while meeting the toughest enterprise security and governance requirements. Presto provides enterprise mashups delivered to the user in 3 clicks versus 3 months.

The Presto family of products includes:

## Presto Server

**Service Virtualization, Security and Collaboration:** Presto Server delivers a service virtualization solution, creating source-neutral services that are easy to find; easy to combine; and easy to share. Presto Server is the glue that connects users to mashups, enabling today's Enterprise 2.0 applications to work efficiently in the browser. Presto Server includes:

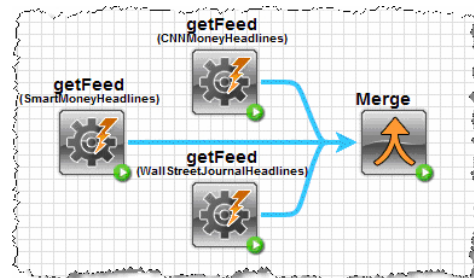
Popular Tags	
Address	(3)
Amazon	(4)
Analysis	(1)
Business	(4)
dapper	(1)
eBay	(1)
eCommerce	(7)
Finance	(5)
Gadgets	(1)
Geocode	(2)
Headlines	(1)
Jobs	(1)
Local	(1)
Maps	(2)
News	(11)
Payment	(1)
PayPal	(1)
Quotes	(3)
REST	(14)
RSS	(10)
Search	(8)
SOAP	(2)
SOAP-Document	(1)
SOAP-RPC	(1)
Software	(2)
Startups	(2)
Stocks	(4)
Strikelron	(2)
Technology	(4)
Traffic	(2)
Travel	(2)
Trends	(1)
Trip	(1)
Weather	(2)
WSDL	(3)
Yahoo!	(1)
Zip Code	(2)

- **Mashup Engine** provides dynamic runtime processing for mashups.
- **Service Access Engines** provides access to all popular data sources including RSS/ATOM, Database, REST, WSDL/SOAP..
- **Virtualization Management** enables disparate mashup sources to be exposed while obfuscating complexity from the user.
- **Enterprise Mashup Markup Language (EMML)** provides an open, declarative XML-based dynamic mashup language for rapid mashup creations.

## Presto Mashup Composers

**Simplified Mashup and Mashlet Creation:** Presto provides three unique tools to enable business and technical users to create mashups.

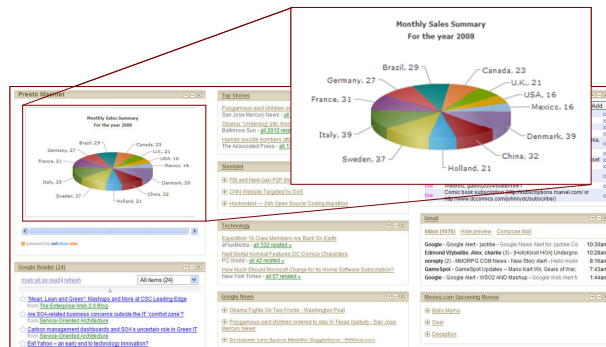
- **Presto Wires** is a browser-based drag-and-drop visual mashup composition tool for IT and business users.
- **Presto Mashup Studio** is an Eclipse plug-in providing programmers complete control to design, test, debug and deploy mashups.
- **Presto Mashlets** are enterprise widgets that give mashups a portable, sharable user interface. Presto ships with 20 different visualization mashlets and can be enhanced with many more.



## Presto Mashup Connectors

**Integration with Your Infrastructure:** The family of Presto Mashup Connectors connects your Presto mashups to standards-based third-party tools that enable faster, safer mashups. Through interfaces like WSRP and GIF, JackBe has created bi-directional connectors from Presto to standards-compliant enterprise applications including HP SOA Systinet, Oracle Fusion, Microsoft Excel and other important enterprise tools.

- The **Presto Connector for Excel** allows users to consume mashups into Excel and publish spreadsheets as mashup services directly from Excel.
- The **Presto Connector for HP SOA Systinet** automates the sharing of SOA governance information between HP Systinet and Presto, ensuring mashups are first-class SOA citizens.
- The **Presto Connector for Portals** lets users publish Mashlets directly to leading portal platforms including IBM, BEA and Oracle.
- The **Presto Connect APIs** provides mashup connectivity to for Java, JavaScript, and C# developers.
- The **Presto Event Connector** enables mashups to subscribe or publish to JMS queues.

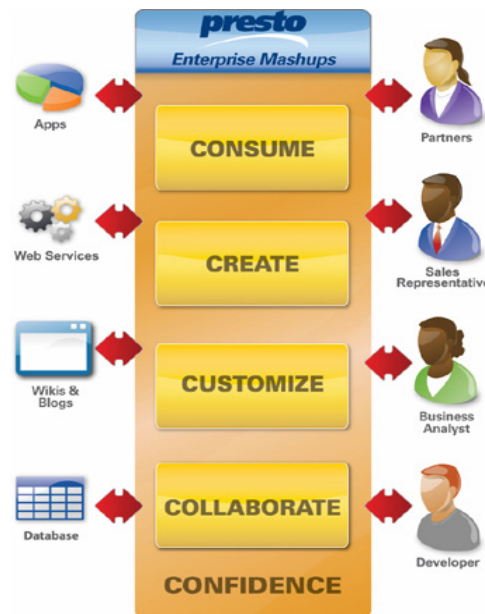


A Sales Mashlet in an Enterprise Portal

JackBe's Presto uniquely meets the 5Cs of enterprise mashups. Presto empowers application developers and users to **consume** information service, **create** new information answers, **customize** the results to their need, **collaborate** with peers by sharing the results in an efficient way, and doing so within a secure and governed framework of **confidence**.

The possible applications of Presto-driven Enterprise Mashups are numerous. For example, the ad-hoc nature of mashups provide a perfect fit for:

- **Enhanced customer self service applications**, delivering integrated views of information that customers expect to see on line.
- **Enhanced supply chain management applications** achieving better visibility across key customers and suppliers
- **Richer enterprise portal experiences** serving up corporate data sources, web based data, and data from many disparate sources directly to the user.



For more information about JackBe and JackBe's Enterprise Mashup Platform, Presto, go to <http://www.jackbe.com>.

## **Mashups Are About Business Results**

While enterprise mashups are relatively new, the growing influence of user-empowering 'Web 2.0' trends and technologies truly underscore the value enterprise mashups can have for solving critical business problems quickly and easily. Enterprises must approach enterprise mashups as a valuable asset that has the potential to add significantly to the bottom line.

And it is very important to remember that a critical success factor for the delivery of enterprise mashups to end-users is leveraging the correct technology for the job. While existing application development environments may seem like a viable option, the unique characteristics of mashups discussed here call for technology that addresses enterprise mashups specifically. Moreover, it does not replace, only enhances, existing enterprise assets such as SOAs, portals, and rich internet application development tools.

By leveraging mashup technology, you'll find that it quickly pays for itself in both developer productivity, as well as positive impacts on operational efficiencies, and support of dynamic high-growth initiatives. Enterprise mashup are indeed the wave of the future, and approached correctly with the proper technology, have huge ROI potential.