

## Mashups versus Portals

It is important to differentiate a mashup from simple (external site) data embedding. For instance, if I have a site that allows to see a video in YouTube I do not have a mashup, that is simple embedding. **Mashups are intended to add value to the final user, processing information consumed by any API/Web Service and displaying it in an integrated experience.**

When the whole concept of Portals first come out, the whole idea was that we could draw content from various sources using little Portlets, and personalize it to what we want. Portals are providing a platform to leverage off these services. If you think of Facebook as a platform, and all the widgets/gadgets as what makes the overall content, you have a mashup. Through a portal-like interface you have brought in content from many sources: Web Services (SOAP, XML-RPC), REST, RSS Feeds and even Screen Scraping. Google also had its own portal-like solution with iGoogle (aka Google Start Page). iGoogle was a Portal platform that left you add and configure your own widgets/gadgets (aka Portlets) to do what you want. However, Google shutdown iGoogle after 8 year of usage in 2013 because the browsers are evolved to act as an application platform and its own Chrome browser could fulfill all the tasks what iGoogle did.

Both Mashups and portals are content aggregation technologies. Portals are an older technology designed as an extension to traditional dynamic Web applications, in which the process of converting data content into marked-up webpages is split into two phases: generation of markup "fragments" and aggregation of the fragments into pages. Each markup fragment is generated by a "portlet", and the portal combines them into a single webpage. Why it is different from the mashups that portal technology defines a complete event model covering reads and updates. Portal technology is about server-side, presentation-tier aggregation. A request for an aggregate page on a portal is translated into individual read operations on all the portlets that form the page.

In 2009, Gartner introduced the concept of the portal-less portal or the "lean portal". Gartner defines a portal as

a Web software infrastructure that provides interaction with relevant information assets (for example, information/content, applications and business processes), knowledge assets and human assets by select targeted audiences, delivered in a highly personalized manner."

It's built using modern Web 2.0 technologies, such as AJAX, widgets, representation state transfer (REST) and WOA/SOA approaches. Lean Portal offerings from vendors like IBM WebSphere, Oracle WebCenter and Liferay replace the traditional container-oriented portal model while maintaining the main purpose of a portal — providing a personalized point of access that allows customers to find relevant information, read about business processes and reach people. According to Gartner, organizations who opted for a Lean Portal found that it delivered more than 80% of the required functionality within months of launching, without compromising security or advanced integration requirements.

While both portals and mashups are supporting integration of content provided by other websites and presentation, portals are offering more things like navigation between portlets, access control and personalization. The list of the advanced features are:

- Single Sign-On — enterprise portals can provide single sign-on capabilities between their users and various other systems. This requires a user to authenticate only once.
- Access Control — the ability for portal to limit specific types of content and services users have access to. Access control lists manage the mapping between portal content and services over the portal user base.

- Integration — the connection of functions and data from multiple systems into new components/portlets with an integrated navigation between these components.
- Business process management (BPM) and a mechanism for providing or integrating with workflow and BPM tools or platforms.
- Federation — the integration of content provided by other portals, typically through the use of WSRP or similar technologies.
- Customization — Users can customize the look and feel of their environment. Customers who are using EIPs can edit and design their own web sites which are full of their own personality and own style; they can also choose the specific content and services they prefer.
- Personalization — Personalization is more about matching content with the user. Based on a user profile, personalization uses rules to match the "services", or content, to the specific user. To some degree, you can think of the two like this: customization is in hands of the end user, personalization is not.
- Support for multichannel and multi-device delivery.

Moreover, not just these aspects are made the separation between them, the following table from the crowdsourced Wikipedia also highlights the primary differences:

	Portal	Mashup
<b>Classification</b>	Older technology, extension of traditional Web server model using well-defined approach	Uses newer, loosely defined "Web 2.0" techniques
<b>Philosophy/approach</b>	Approaches aggregation by splitting role of Web server into two phases: markup generation and aggregation of markup fragments	Uses APIs provided by different content sites to aggregate and reuse the content in another way
<b>Content dependencies</b>	Aggregates presentation-oriented markup fragments (HTML, WML, VoiceXML, etc.)	Can operate on pure XML content and also on presentation-oriented content (e.g., HTML)
<b>Location dependencies</b>	Traditionally, content aggregation takes place on the server	Content aggregation can take place either on the server or on the client
<b>Aggregation style</b>	"Salad bar" style: Aggregated content is presented 'side-by-side' without overlaps	"Melting Pot" style - Individual content may be combined in any manner, resulting in arbitrarily structured hybrid content
<b>Event model</b>	Read and update event models are defined through a specific portlet API	<a href="#">CRUD</a> operations are based on <a href="#">REST</a> architectural principles, but no formal API exists
<b>Relevant standards</b>	Portlet behavior is governed by standards <a href="#">JSR 168</a> , <a href="#">JSR 286</a> and <a href="#">WSRP</a> , although portal page layout and portal functionality are undefined and vendor-specific	Base standards are XML interchanged as <a href="#">REST</a> or Web Services. <a href="#">RSS</a> and <a href="#">Atom</a> are commonly used. More specific mashup standards such as <a href="#">EMML</a> are emerging.

[Prev](#)

Mashups

[Up](#)
[Home](#)
[Next](#)

Cloud Computing