

PORTALS, SOA AND COMPOSITE APPLICATIONS

Together they are greater than the sum of their parts

GLINTECH Tech Sheet

Q. How can businesses become more flexible and have relief from the pressure of adapting to changing market demands without having to rewrite existing software applications?

A. By building composite applications out of existing business IT assets and providing users the ability to create through a user-centric Portal.

COMPOSITE APPLICATIONS

The term 'composite application' was coined when German software developer SAP AG announced its xApps software and Composite Application Framework. Composite applications reuse functionality from existing applications by breaking them down to components, separating out the business logic, and then joining selected elements back together again for presentation to the user as a single (hence 'composite') application.

SOA: KEY TO COMPOSITE APPLICATIONS

The de-coupling of existing applications and business processes into web services as part of a shift towards Services Oriented Architecture (SOA) has enabled the rapid assembly of new applications from existing IT assets. This decoupling presents the need of a new integration layer that bridges the various systems on which each of the services run. At this layer resides the Composite Business Logic that makes up the building blocks of a composite application. Figure 1 shows how the Composite Business Logic fits into the technology stack.

PORTALS FOR COMPOSITE APPLICATIONS

Portals are a single point of access to multiple business processes and applications and have the ability to complete logical business processes, regardless of the underlying application architecture. Portals are ideal candidates for the delivery of composite applications because of their inherent nature of application aggregation via portlets. Portlets can be written to expose the functionality of web services and through the use of inter-portlet communication, or wiring, can quickly be assembled into composite applications.

Wiring enables interaction and data exchange between two or more portlets making them 'context aware'. The advantage of using portlets is that they can be created as standalone building blocks serving specific functions and data stores. Wiring allows portlet specific data and events to be sent from one portlet to others, which can dynamically drive the data and functions of the other portlets. By building composite applications, the developer can eliminate manual and repetitive data lookups and improve efficiencies. If the portlets

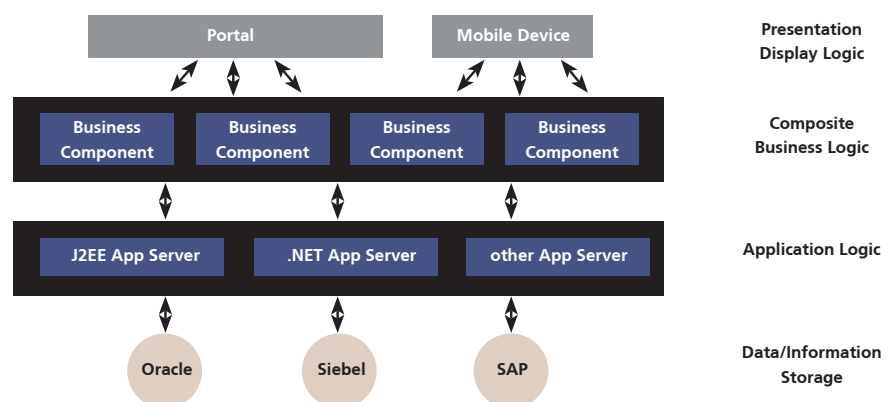
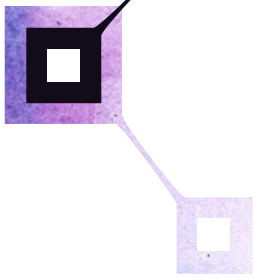


Figure 1. Composite Business Logic in the technology stack.



are designed with careful consideration, it would be feasible for business user to build their own composite applications by simply wiring the correct parts together.

To illustrate the portlet-wiring concept, let us consider a composite application made up of four portlets. The first is a city selector portlet, which allows you to select a city from a list in a data store. The other three: map, weather and news rely on a city as the input parameter that directs each to bring back the related information on the chosen city. When the user selects a city, a message containing the selected city is sent to each of the other portlets and they all subsequently render their content based upon that input. Figure 2 shows how the wiring of portlets produces a composite application.

USER-CENTRIC AND COST EFFECTIVE

Combining composite application development with portal delivery enables the creation of user-centric composite applications that allow end users to participate productively. As portals provide content and functionality based on pre-defined user roles, any changes to business processes via way of new composite applications can easily or even automatically be deployed to the end user. In fact, users may have the ability to click their way to new composite applications themselves by installing and wiring the various parts together.

Composite applications are not developed; they are assembled, which means they take significantly less time to construct than coding from scratch and they require less testing. This enables businesses to build IT systems from existing ones and to quickly adapt to new initiatives and market changes at less cost than was previously possible.

In summary, composite applications are made up of decoupled business processes that are reassembled to form new applications. Portals are complimentary in that they provide a user-centric framework for the delivery of composite applications with the use of portlets. In addition, the ability for inter-portal communication or wiring adds the ability for non-programmers to assemble composite applications. The benefits realised by businesses include the ability to build new systems quickly and easily allowing then the freedom to change their business processes to adapt to new initiatives.

Together, Portals, SOA and composite applications, provide a cost- effective, flexible development solution. Together, they are greater than the sum of their parts.

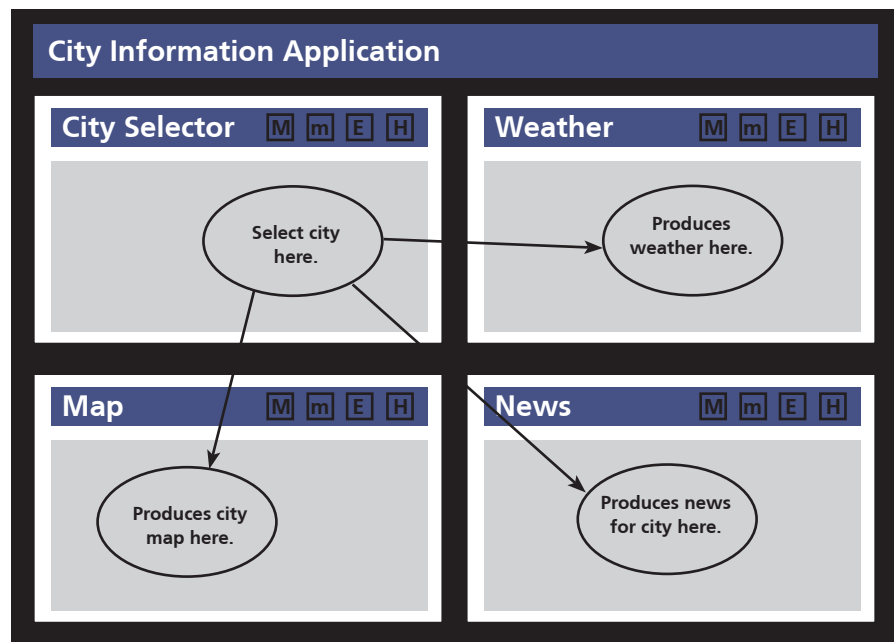


Figure 2. Portlet wiring to produce a Composite Application