Portal Integration Scenarios

Applies to:

Article applies to SAP Enterprise Portal.

Summary

The most common business case for portal implementation is the need to provide a common platform with a consistent look-and-feel for access to different applications within the Organization's intranet or through the web for access to Customers. This document discusses the content aggregation / integration scenario's that are feasible and approach that should be adopted under different scenarios. It covers each approach in detail and provides recommendations and best practices. The approach is not specific to any product and can be generally adopted for almost all portals that support the different alternatives for aggregation / integration.

Author: Rakhi Gupta

Company: Tata Consultancy Services Ltd.

Created on: 1st April 2007

Table of Contents

Introduction 3

1. Linking 4

2. Data Integration 4

3. Integration through Web Services 4

4. Integration through XML 5

5. PORTLETS (JSR 168) based integration 5

6. “WSRP” BASED INTEGRATION 6

7. Data Integration with KAPOW 6

Disclaimer and Liability Notice 8

Introduction

Following are the Possible Mechanisms and the Situation that Can Be Used to integrate Content in a Portal:

Launching applications as URL's from the Portal – This should be used only for quick fix and temporary solutions.

Data Integration – This should be used when data from the backend needs to be web enabled or the users of the new interface are different than the present users of the application. Typical case is SAP GUI transactions that need to be enabled for Self-Service through the portal.

Integration through Web services – This should be used to provide sub-aspects of the applications in the portal e.g. Human Interaction Workflows.

Integration through XML – This should be used for linking other external data sources e.g. travel information, news etc.

Portlets (JSR168) based Integration – This should be selected, in case the applications contain a strict distinction between presentation logic and application logic. Here, the remote portlets should be preferred to the classical portlets. If complex portlets and Web service for remote portlets are used on much frequented pages, the effect on the performance of must be evaluated. If there are performance related problems, corresponding contents should be displayed on another site connected through a link.

WSRP based Integration - This is the preferred integration solution for applications with sufficient bandwidth and response times between portal and the application.

Data integration through Kapow – This approach can be selected as integration scenario in case a web application with the scenario of application integration needs to be integrated in the portal, but it is not immediately possible.

1. Linking

Linking is not integration because the entire application functionality, starting from data up to the layout lies in the application itself. However, it is taken into consideration because linking, in specific cases, is used for a loose connection and, in addition, is a cost-effective variant. The single sign-on functionality needs to be ensured in case applications are required to be linked.

2. Data Integration

Data integration is used for those applications, which do not have an individual application logic or presentation logic or the application logic or presentation logic should not be used. Typically, data integration takes place when data needs to be integrated directly from databases, data warehouses or from files. The information can be displayed using reports.

Following technologies are feasible for data integration:

JDBC (transactional and non-transactional)

CSV files (non-transactional)

XML files (non-transactional)

SAP IDOC (transactional and non-transactional)

The following important considerations need to be kept in mind:

In case data in the portal needs to be changed, the synchronization and transaction safety should be ensured.

It is important during data integration that the data sources have stable output formats or, otherwise, the data source formats are consistently decoupled from the portal-specific formats.

It must be checked and documented before the integration, how the data sources should be accessed and which general conditions are connected with the data source. It should be determined, whether the access should be transactional (read and write accesses) or non-transactional and whether the data source fulfill the logical and technical pre-requisites as specified in the table.

When it is a file based integration (CSV/XML) - One storage location (directory) must be provided for files for each application.

3. Integration through Web Services

The integration of web service is different from the integration with the help of Web service for remote portlet (“WSRP”), because Web services do not provide any presentation logic only business logic. It is recommended to integrate Web services when human interaction in IT-supported business flows becomes necessary.

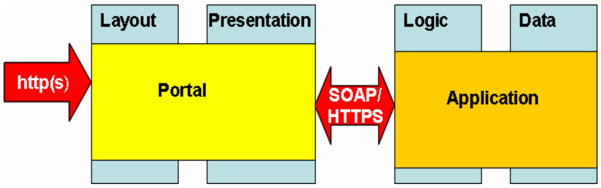

Figure 1: Integration through Web Services

Integration of Web services is relatively simple however the following considerations need to be kept in mind:

It is resource-intensive.

There is a risk of underestimating the interface dynamics. This dynamic emerges when Web services are created in a very fine-granular manner and it results in many interfaces. This gives rise to the change frequency and the related maintenance expense.

Integration of an application via web services requires a service platform (generally consists of registry, lifecycle management, policy management and runtime environment) that helps the application in providing its services.

Registry and runtime environment are contained in each portal product. Ensure that Policy and lifecycle management are also covered by the portal product, in case too many web services are used.

The providing application, depending upon the use case, must support the service standards in a company in order to provide web services (safety standards, transaction handling and policy use).

A clean separation must be present in service-fit structures, which can be provided using SOAP interfaces in order to provide the (sub) services of an application at all.

Both target portal for integration and source application must support standards WS-Policy (used policies characteristic) and WS-Security (exposure to certificates, ...).

4. Integration through XML

Integration of XML presents a special variant of data integration. Standardised XML data format is used for integration of XML. Http/https is used for the communication.

Applications that need to be connected via XML-integration are mostly connected via http in order to ensure that, generally, the integration is relatively simple.

The application should allow for the state-less character of the transport log (HTML), due to this reason, the deployment is recommended in typical request-respond scenarios (e.g. via http GET).

5. PORTLETS (JSR 168) based integration

In a portal server portlets help in displaying the contents. This display is independent of the information source, from where information originates. In this context, the portlets are used in different integration scenarios for display.

Under this approach there are three options:

Applications or their presentation layer can be completely created in the portal

Data of external sources can be displayed through data integration e.g. through JDBC portlet.

Outputs of the existing applications can be displayed through front end integration e.g. Clipping or through URL-Portlet.

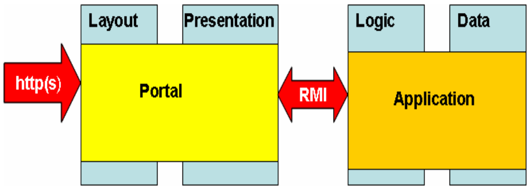

Figure 2: JSR168 based integration

As compared to other approaches portlet integration has the advantage of being able to leverage portal functions on the content of integrated applications, such being able to apply personalization.

6. “WSRP” BASED INTEGRATION

“Web service for remote portlets (WSRP)" presents a relatively new standard for portal integration. The significant difference in comparison to portlets lies in the provisioning of presentation logic by service providers, i.e. the application that needs to be integrated. This leads to the fact that the application structure is not relevant from the point of view of target Portal, which again leads to a significantly better decoupling. The target portal acts as a consumer where as the source portal acts as a producer.

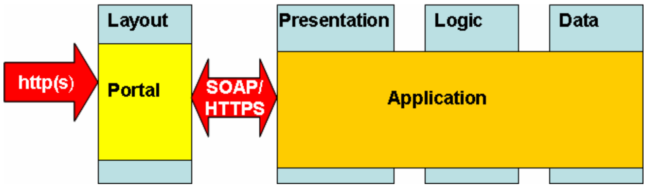

Figure 3: “WSRP” Based integration

The following points need to be kept in mind:

A need for consistent application of standards, in order to be able to integrate applications as remote portlets because the entire application lies outside the portal platform.

Applications with complex presentation logic can also be integrated as a result of the extensive decoupling of applications.

A strict decoupling of the presentation logic from the application logic is not absolutely necessary from the point of view of target portal.

An important consideration is sufficient bandwidth between the target portal platform and the integrated applications.

As compared to portlets, significant advantages are mainly in the presentation logic outside the portal, which reduces the performance influence.

7. Data Integration with KAPOW

The Kapow approach to integration requires applications to be integrated using their presentation layer.

The following considerations need to be kept in mind:

Logins for this application must be possible with the help of reverse proxy single sign-on.

Changes in the integrated applications affecting the display must be maintained in the Kapow tool or the portal.

Data is extracted from the linked websites, which can then be displayed in the portal along with their individual presentation logic.

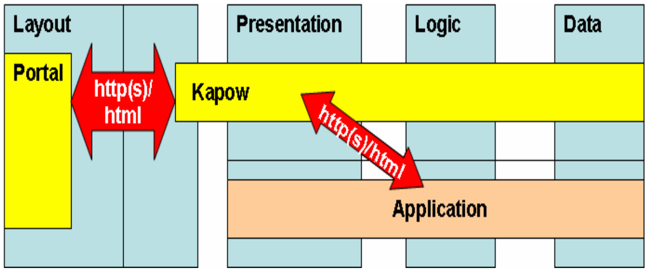

Figure 4: KAPOW based integration

Data integration with the help of Kapow is suitable for one-time migration scenarios and as temporary solution for application integration when an efficient access to other application interfaces is not possible.

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.