Connecting a Portal to Enterprise Applications

Colin White
Founder and President, BI Research
Shared Insights Portals, Collaboration & Content Conference
Boston, November 2006

About Colin White

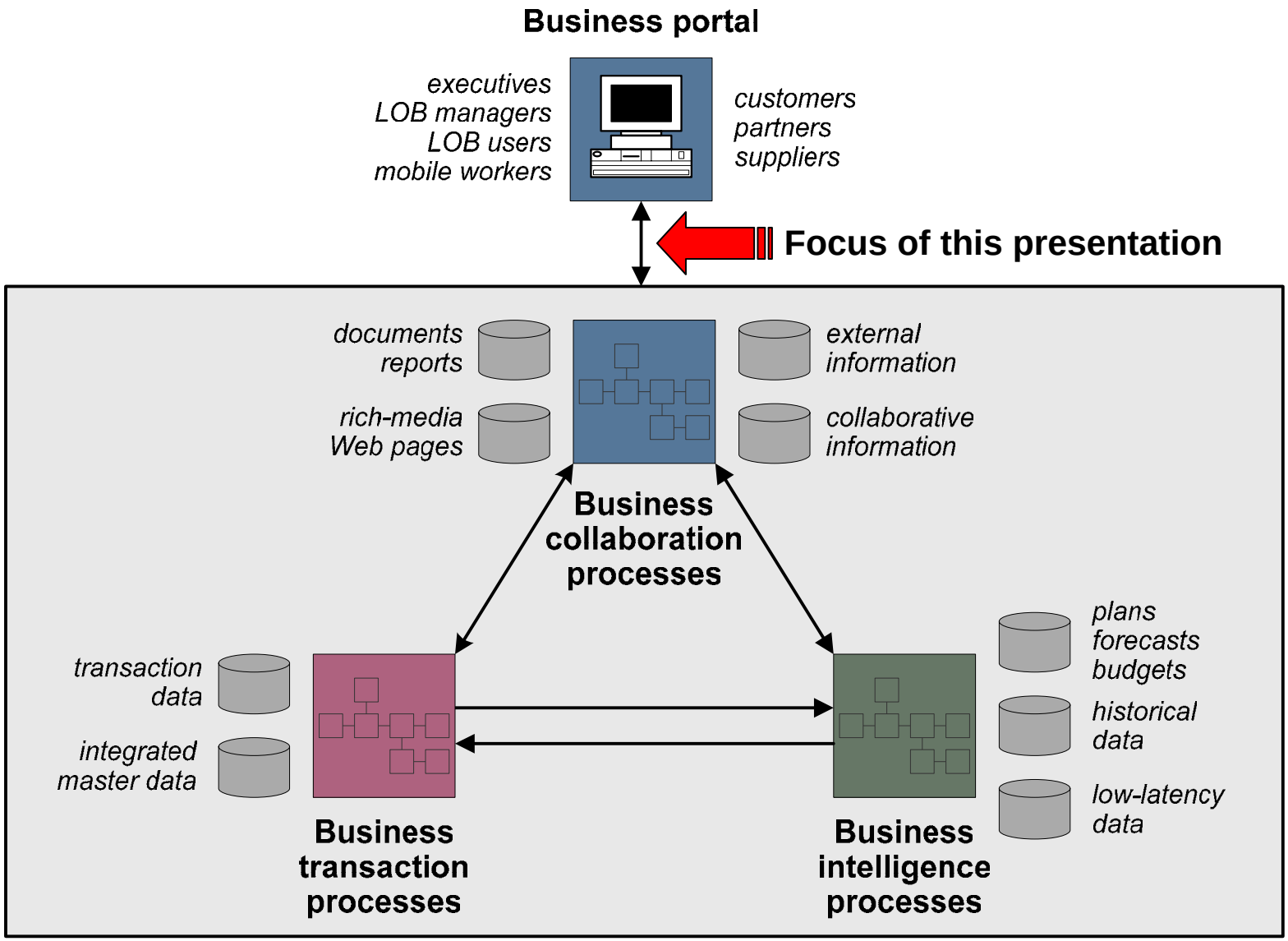
Colin White is the founder and president of BI Research. He is well known for his in-depth knowledge of leading-edge business intelligence and enterprise business integration technologies, and how they can be integrated into an IT infrastructure for supporting the smart and agile business. With over 36 years of IT experience, he has consulted for dozens of companies throughout the world and is a frequent speaker at leading IT events. Colin has written numerous articles and papers on BI and business integration and writes for the Business Intelligence Network and leading industry journals

e-Mail: info@bi-research.com
Web: www.bi-research.com

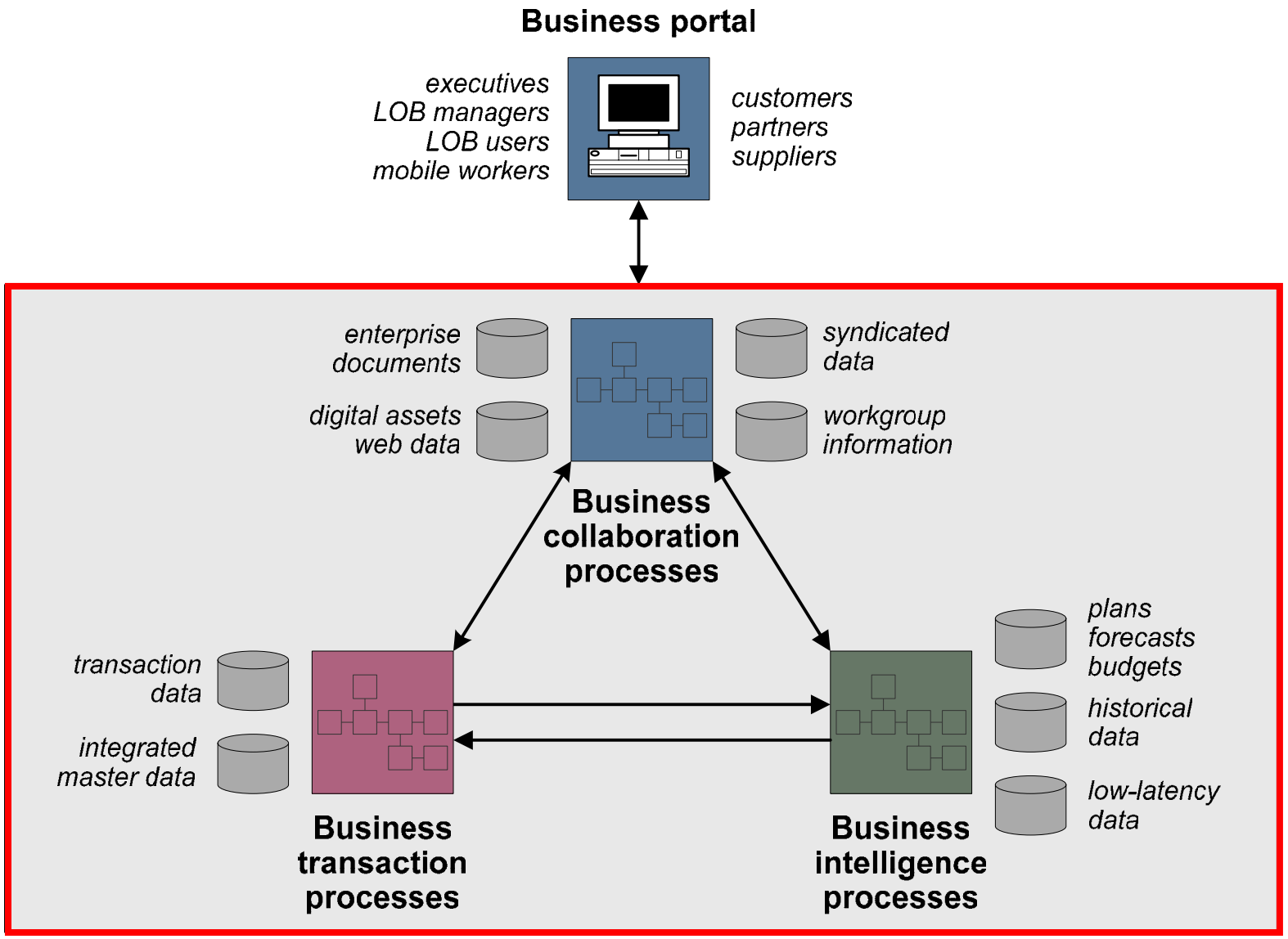
Topics

- Connecting a Portal to Custom Built and Packaged Applications
  - User interaction integration
  - Data integration
  - Application integration
  - Process integration
- Understanding the Role of Composite Applications
- A Status Report on Portal-Related Standards
- The Importance of a Services-Oriented Architecture and Web Services in Portal Integration

The Portal and Enterprise Application Environment



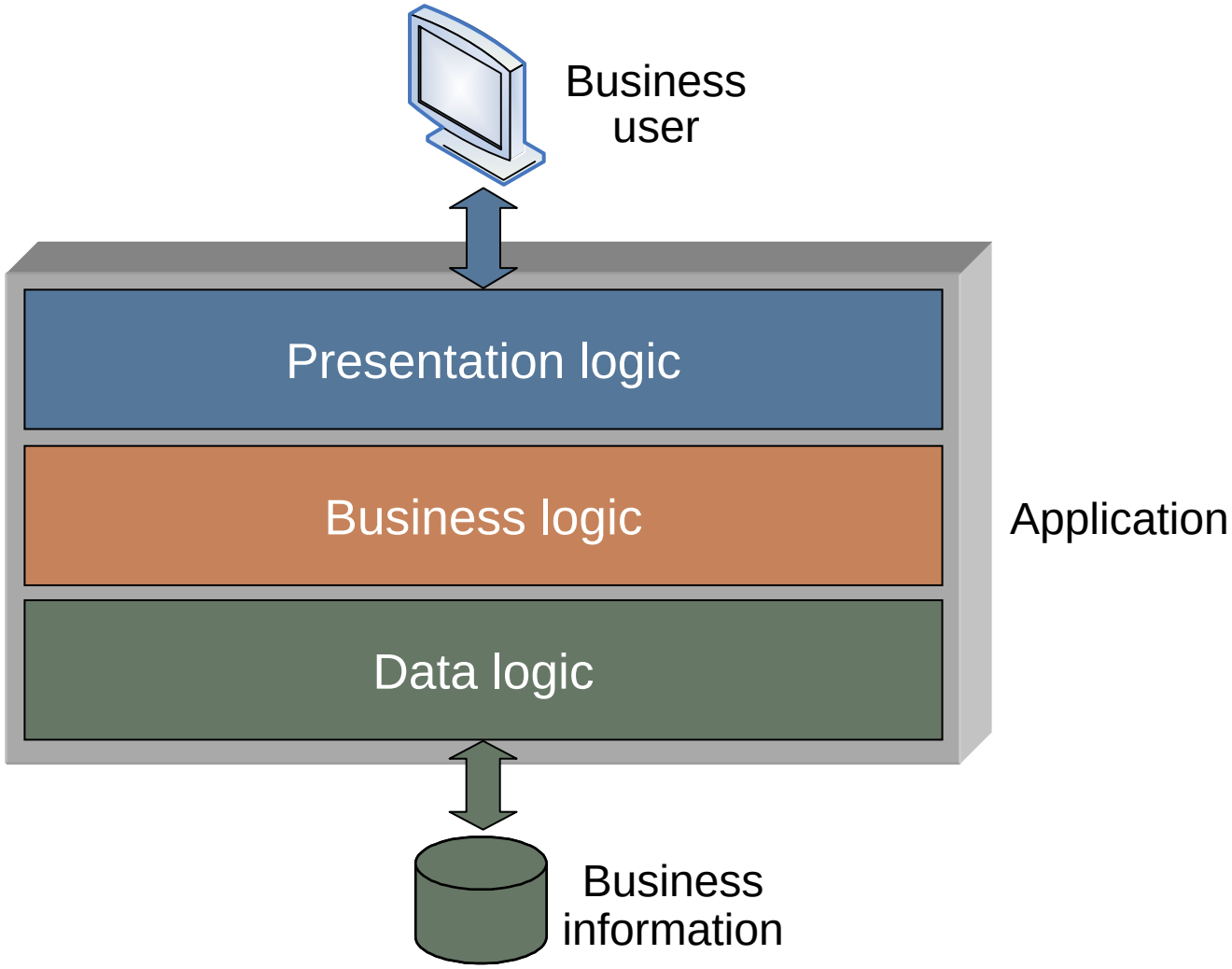
Enterprise Application Environment



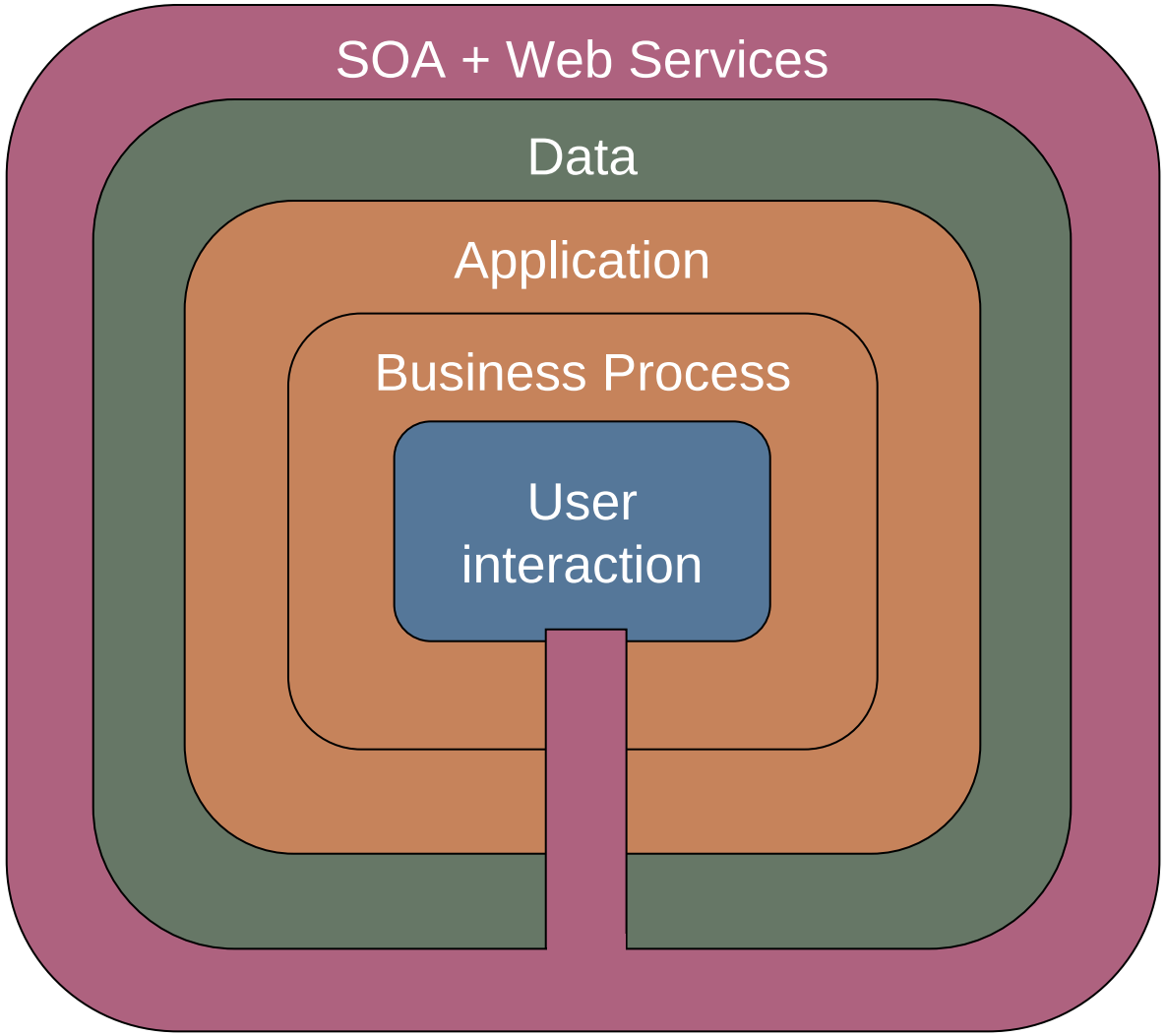
Types of Enterprise Application

- Business Transaction (BTx) Applications
  - Legacy applications
  - Packaged applications
  - External outsourced and partner/supplier applications
- Business Intelligence (BI) Applications
  - BI and planning tools
  - Packaged analytic applications
- Collaborative Applications
  - Collaboration tools
  - Content management (CM) applications
    - Document management
    - Records management
    - Web content management
    - Digital asset management
    - Workgroup content management

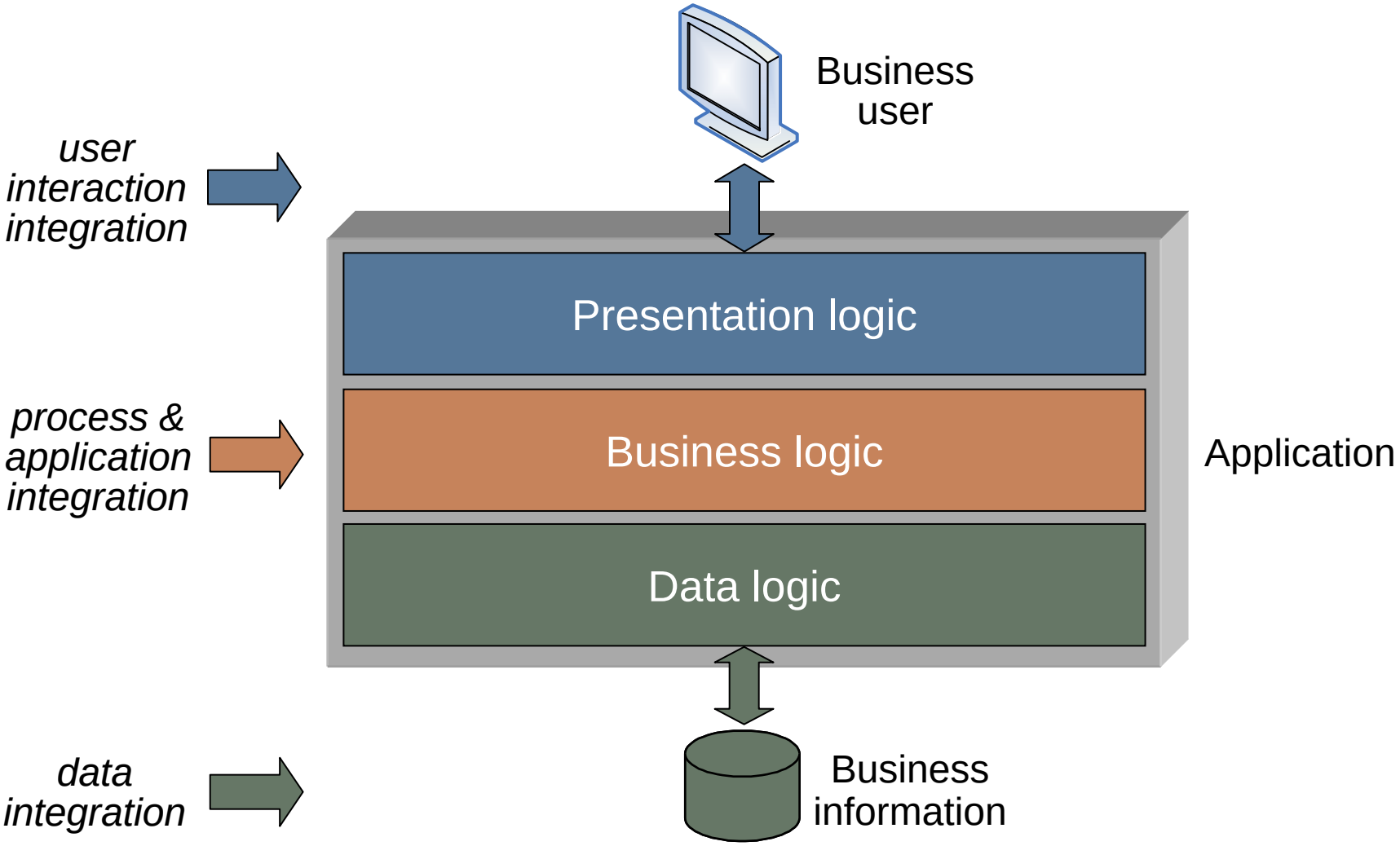
Application Architecture



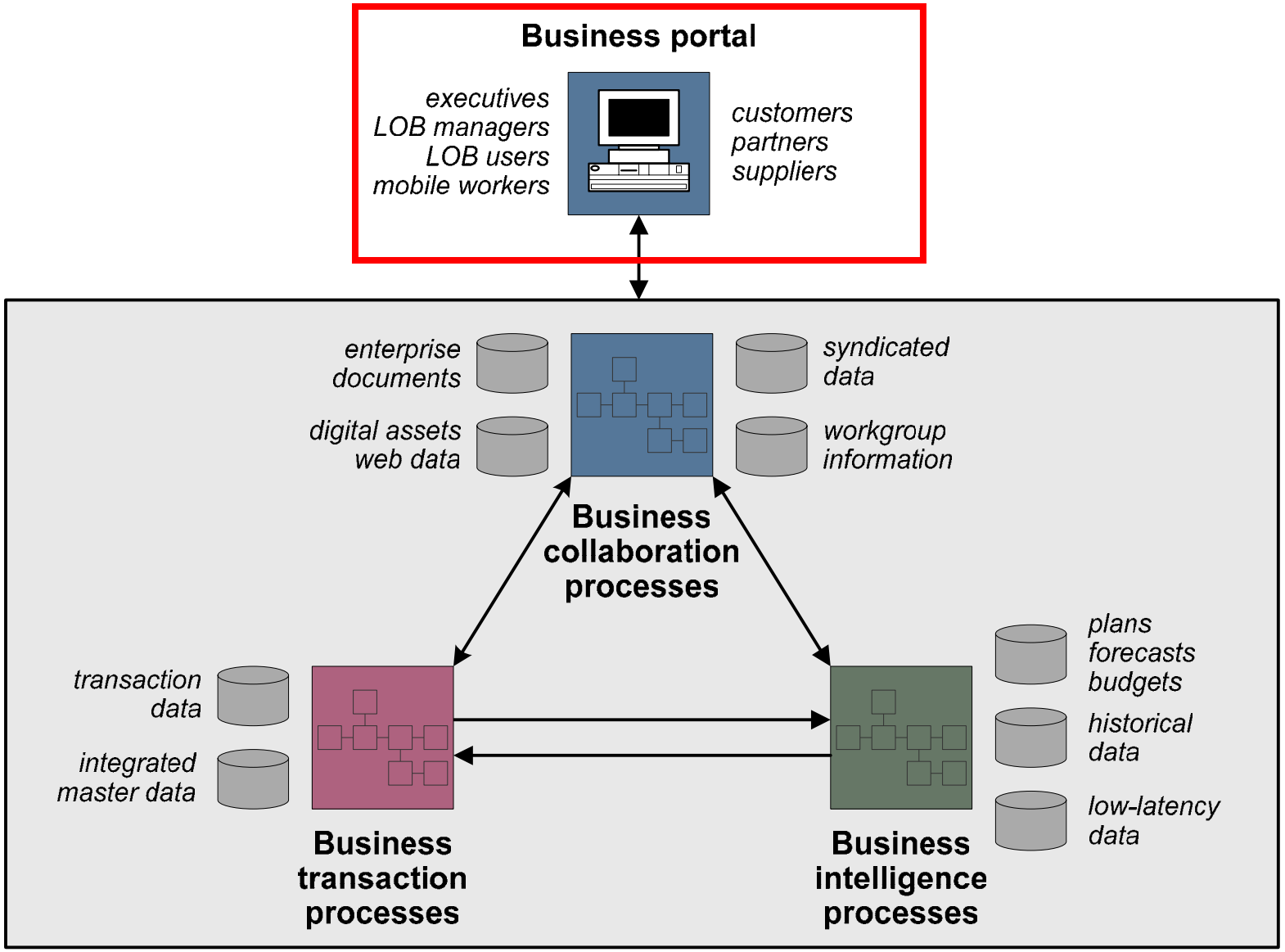
Types of Integration



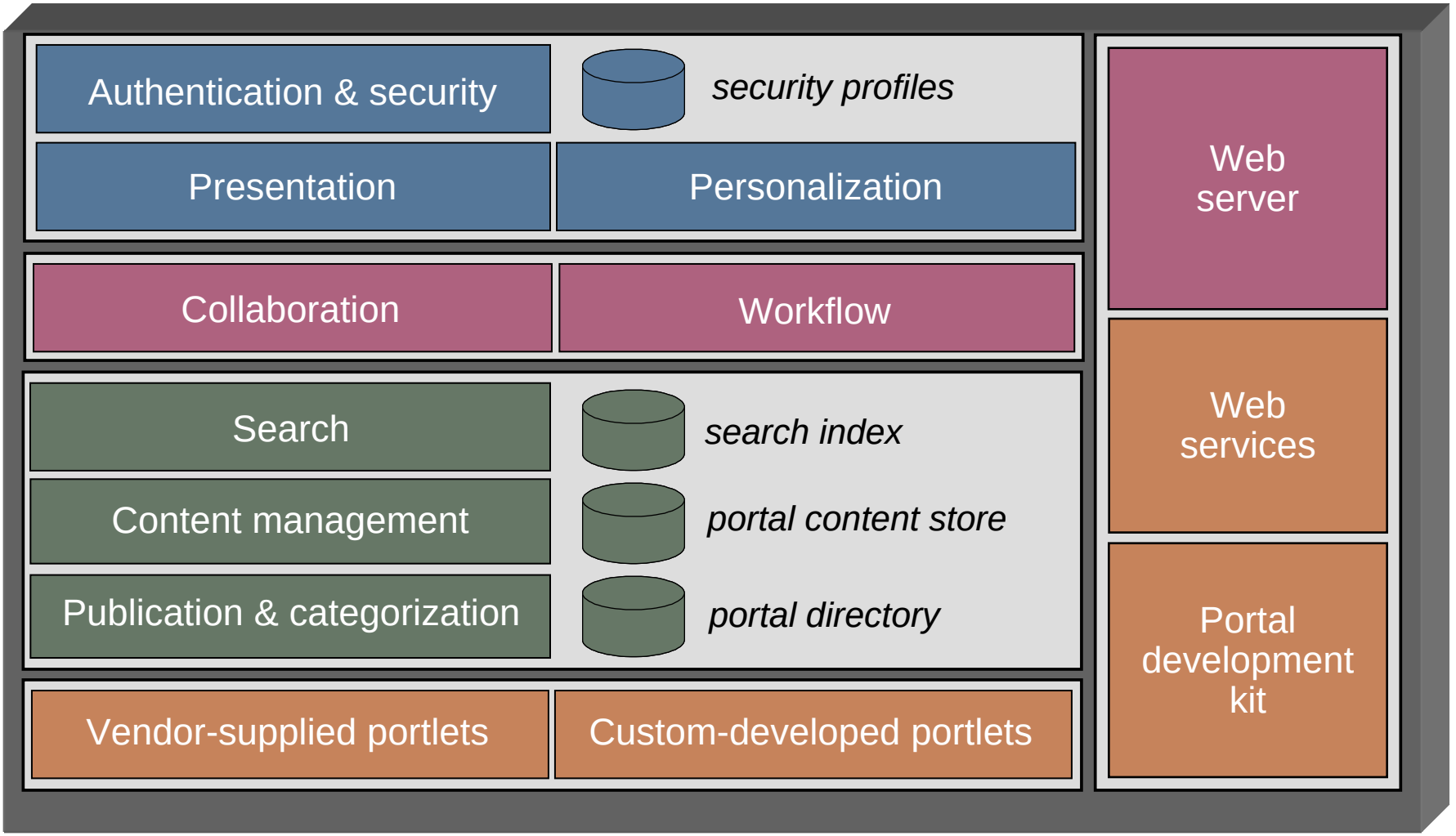
Application Architecture Revisited



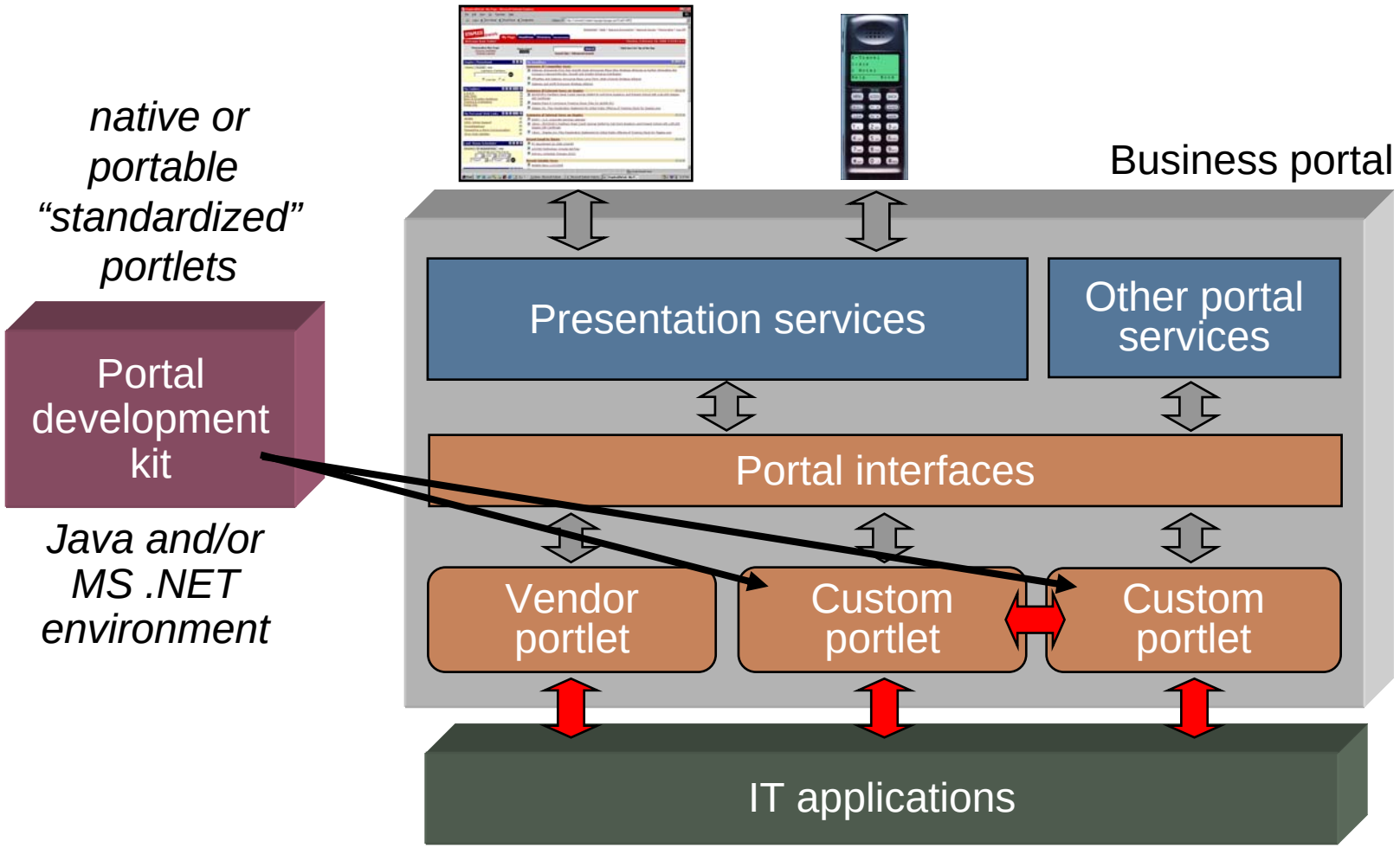
Business Portal Environment



Business Portal Architecture



Integrating a Portal into the Application Environment

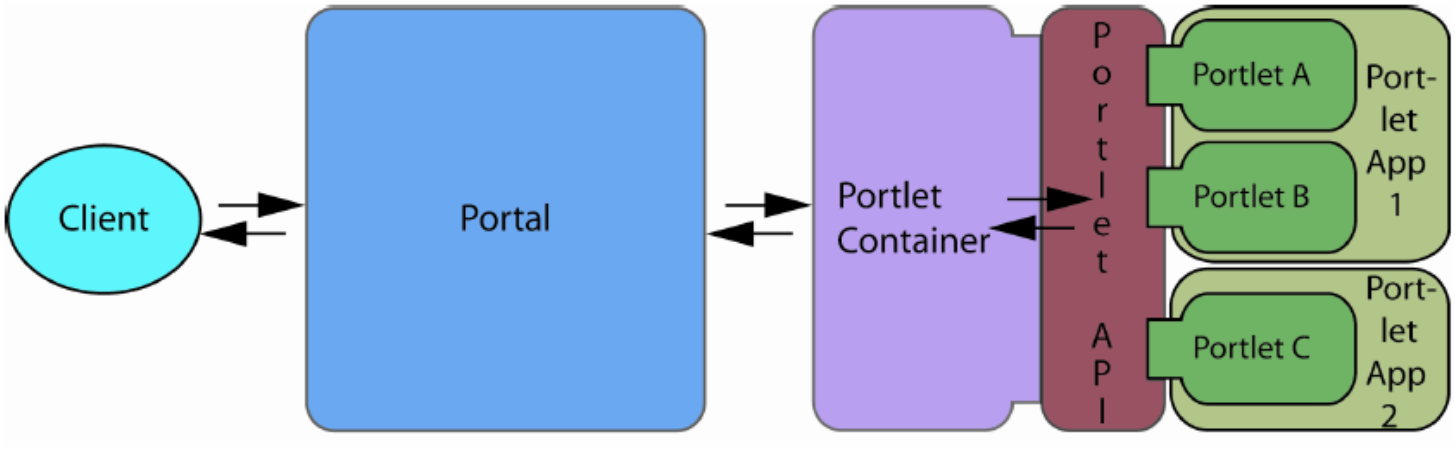


Some Definitions

- Portlets are Web components that generate content fragments (typically pieces of markup such as HTML, XML, or WML)
- By aggregating the content generated by many portlets, a portal is able to generate a single user interface that integrates business processes, applications and data
- A portlet container is responsible for the initialization, request processing, and destruction of portlets
- Portlets have the ability to dispatch requests to servlets, JSPs, and other standard Web resources
- Portlets, servlets, JSPs, and other Web resources are bundled into a specialized Web application called a portlet application
- *Source: www.developer.com/java/web/article.php/3547186*

Portal Standards: Java Portlet Specification (JSR 168)

- Establishes a standard Java portlet API to ensure interoperability between portlets and a portal – approved October 2003

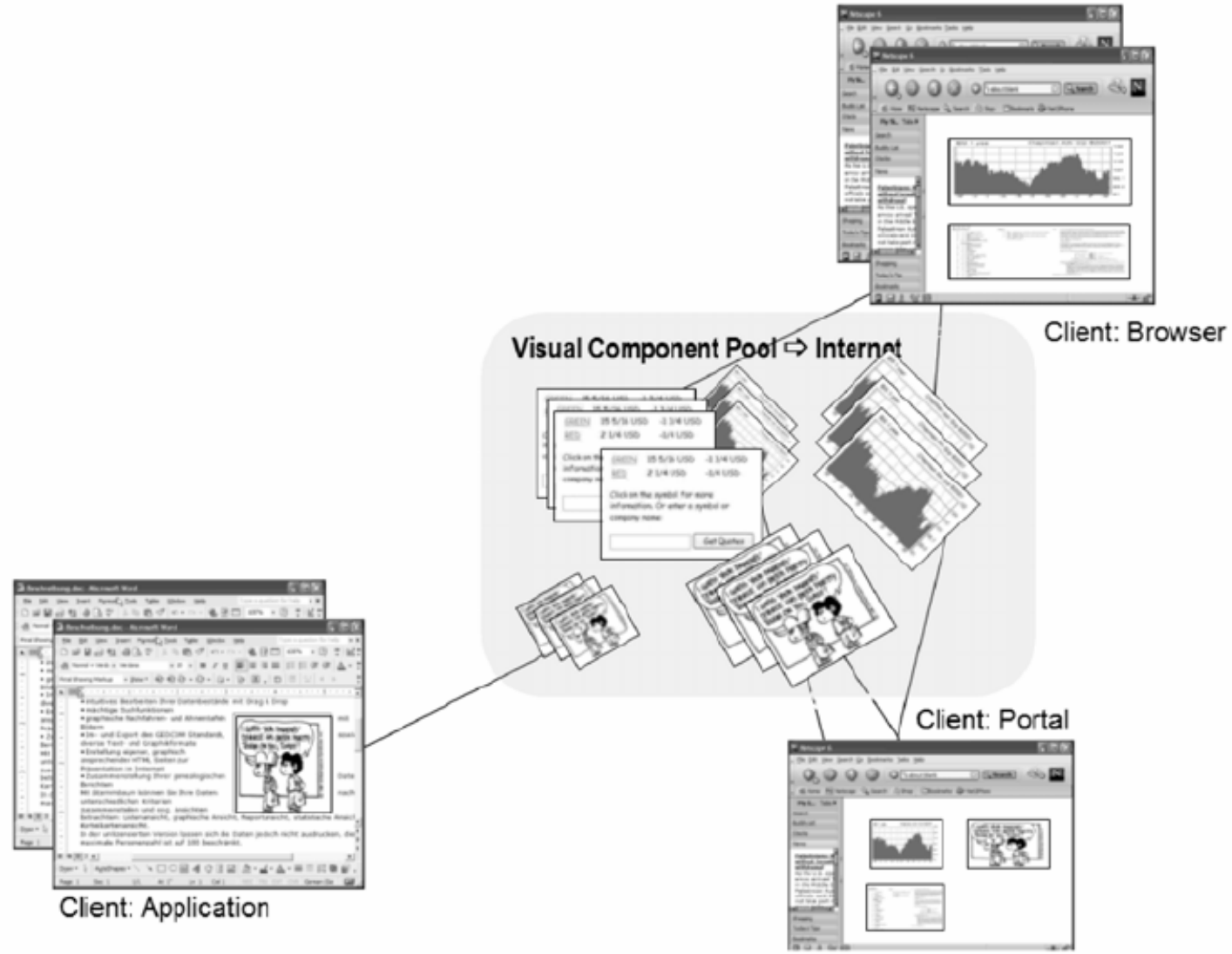


- Developed and governed by the Java Community Process (see *jcp.org/en/jsr/detail?id=168*)
- Java only (specification leaders were IBM and Sun)
- Reference implementation provided by IBM – (portal.apache.org/pluto)
- Compliance test suite available from Sun
- Supported by many commercial and open source portals

Portal Standards: JSR 286

- Represents JSR 168 V2
  - Started November, 2005
  - Planned schedule: early public draft October 2006, final public draft February 2007, approved by May 2007
- Developed and governed by the Java Community Process (see *jcp.org/en/jsr/detail?id=286*)
- Adds functionality in the following areas:
  - Coordination: adds portlet events and sharing of session data and rendering parameters between portlet applications
  - Support for WSRP V2 (see following slides)
  - Improved support for web frameworks (JavaServer Faces, Struts, etc.)
  - Support for AJAX (for asynchronous rendering of portlets)

Web Services for Remote Portlets (WSRP) Vision



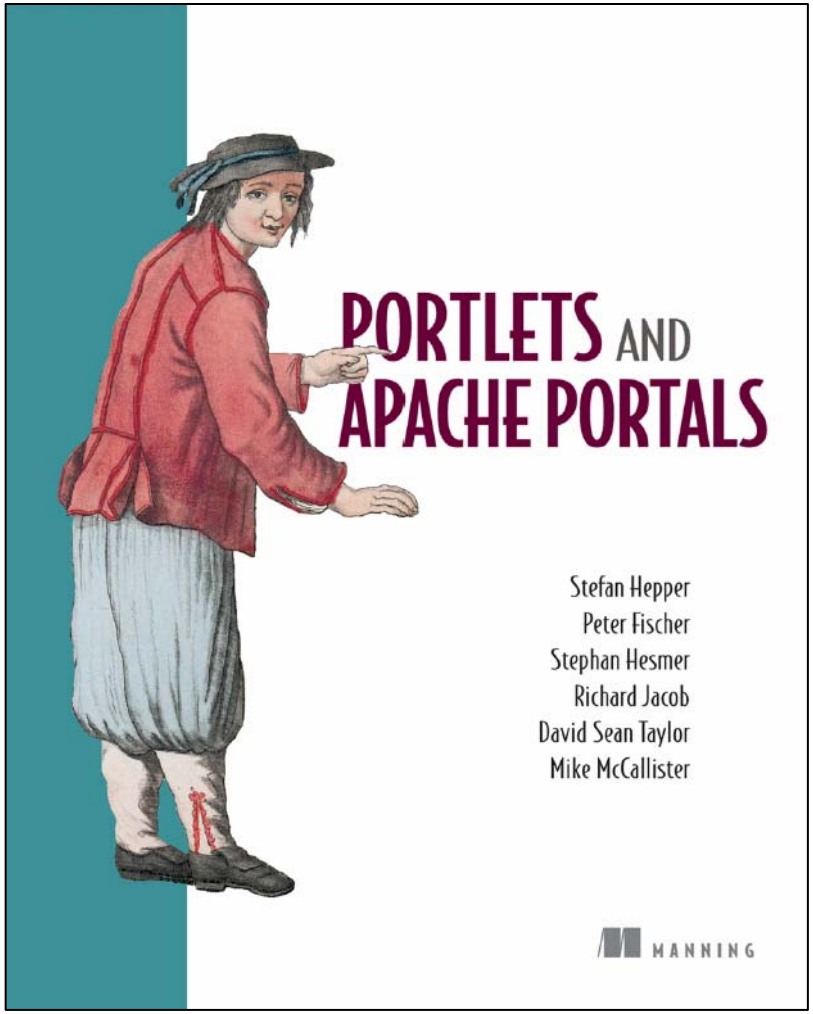
Portal Standards: WSRP

- WSRP enables a portal to aggregate content produced by portlets running on remote systems and that use different programming environments (e.g., J2EE and .NET)
- WSRP is based on web services and describes the generation of markup fragments for aggregation into portals and the handling of user interactions with that markup
- WSRP is defined by the OASIS standards committee (*oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp*)
- WSRP V1 approved September 2003
- WSRP V2 is in development – adds features such as coordination
- A WSRP service can be incorporated into a JSR 168 compliant portal using a JSR 168 proxy portlet
- Supported by many commercial and open source portals

WSRP Versus JSR 168/286

- WSRP is a web services protocol that allows application components (e.g., portlets) to work within another application (e.g., a portal server)
  - Means that portlet developers do not have to build custom user interface code for each different portal server
- JSR 168/286 is a Java API that allows Java-based portlets to work with other portal servers (including WSRP-compliant applications)
  - Means that software vendors do not have to build custom portlets for each Java-based portal server
- Currently benefits mainly vendors, but customers will also benefit from:
  - Less complexity and reduced training requirements
  - More open and less proprietary portal components

Everything You Ever Wanted to Know About Portlets



Excellent (and free) 436 page reference book: www.manning.com/hepper/

Content Management Standard: JSR 170

- JSR 170 was created to establish a standard Java API to content repositories
  - Developed and governed by the Java Community Process (*jcp.org/en/jsr/detail?id=170*)
  - Approved June 2005
  - Technology leader is Day Software
- "The API should be a standard, implementation independent, way to access content on a granular level within a content repository"
- "A content repository implements *content services* such as: author based versioning, full textual searching, fine grained access control, content categorization and content event monitoring. It is these content services that differentiate a content repository from a data repository"
- "The API will focus on transactional read/write access, binary content (stream operations), textual content, full-text searching, filtering, observation, versioning, handling of hard and soft structured content"
- Supported by several open source and some commercial CM vendors

Approaches to Building a Portal - 1

- Open Source Approach
  - Open source Java-Based Web application server running on a Linux operating system
  - + Open source portal server (e.g., Apache Jetspeed, eXo, JBoss, Liferay)
  - + Open source database system (e.g., MySQL)
  - + Open source collaboration (e.g., Mozilla, OpenOffice), content management (e.g., Apache JackRabbit), search (e.g., Lucene)
- Commercial Application Infrastructure Platform
  - Web-based application and portal development environment
  - May also support application integration, business process management, etc.
  - Provided by an infrastructure vendor such as BEA, IBM, Microsoft, Oracle, TIBCO, SAP, etc.

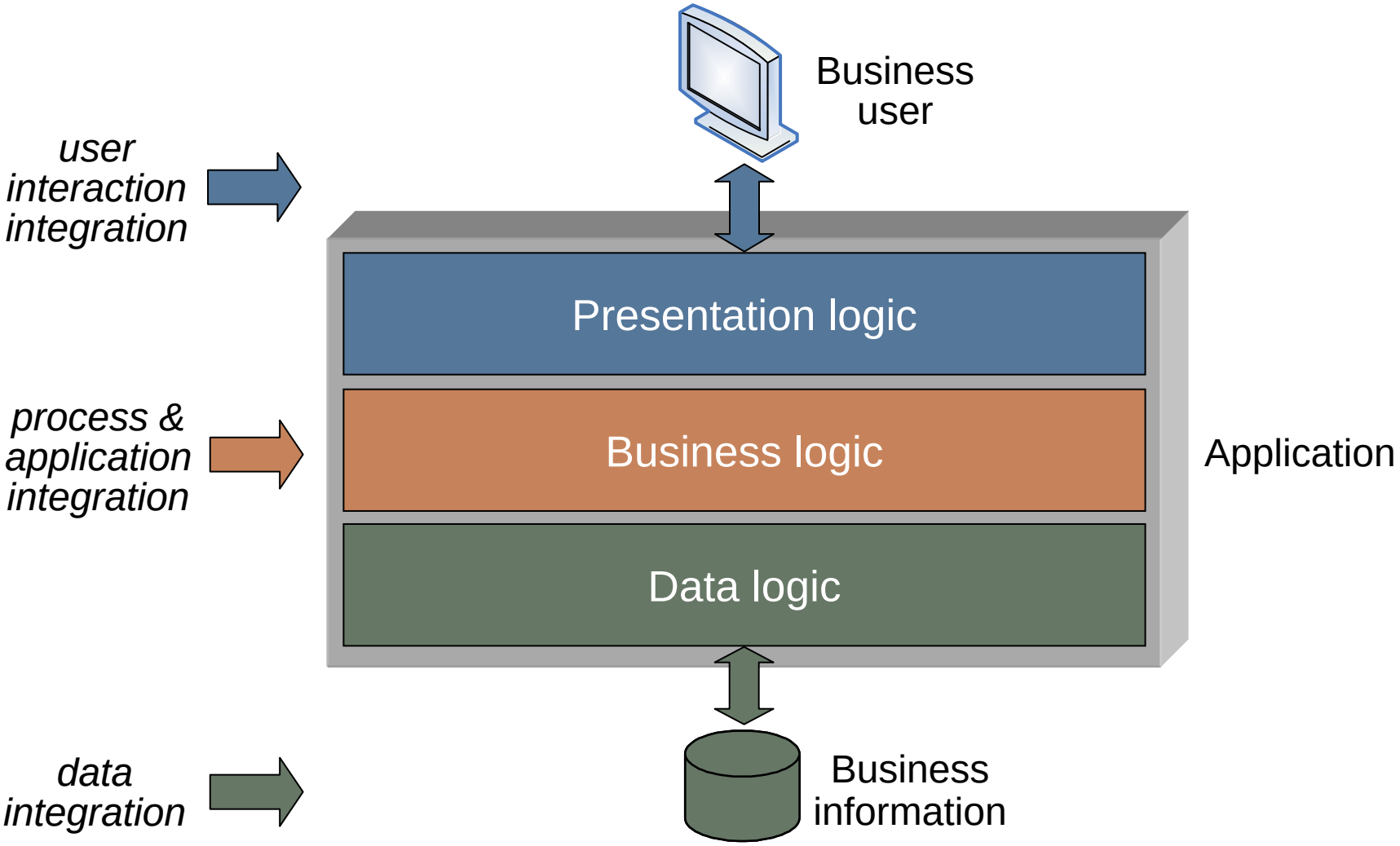
Approaches to Building a Portal - 2

- Out-of-the-Box Solution
  - Stand-alone portal, e.g.,
    - Vignette (Portal, Builder, Business Integration Studio)
    - Microsoft (Windows SharePoint Services and SharePoint Portal Server)
  - Vendor may offer additional capabilities such as content management, collaboration, etc.
- Packaged Application Portal
  - The Portal is integrated with vertical or horizontal application packages, e.g., Oracle Applications, SAP mySAP applications
  - May be a business transaction package, e-business package, or business intelligence package/tool
  - Often deployed using the vendor's application infrastructure platform

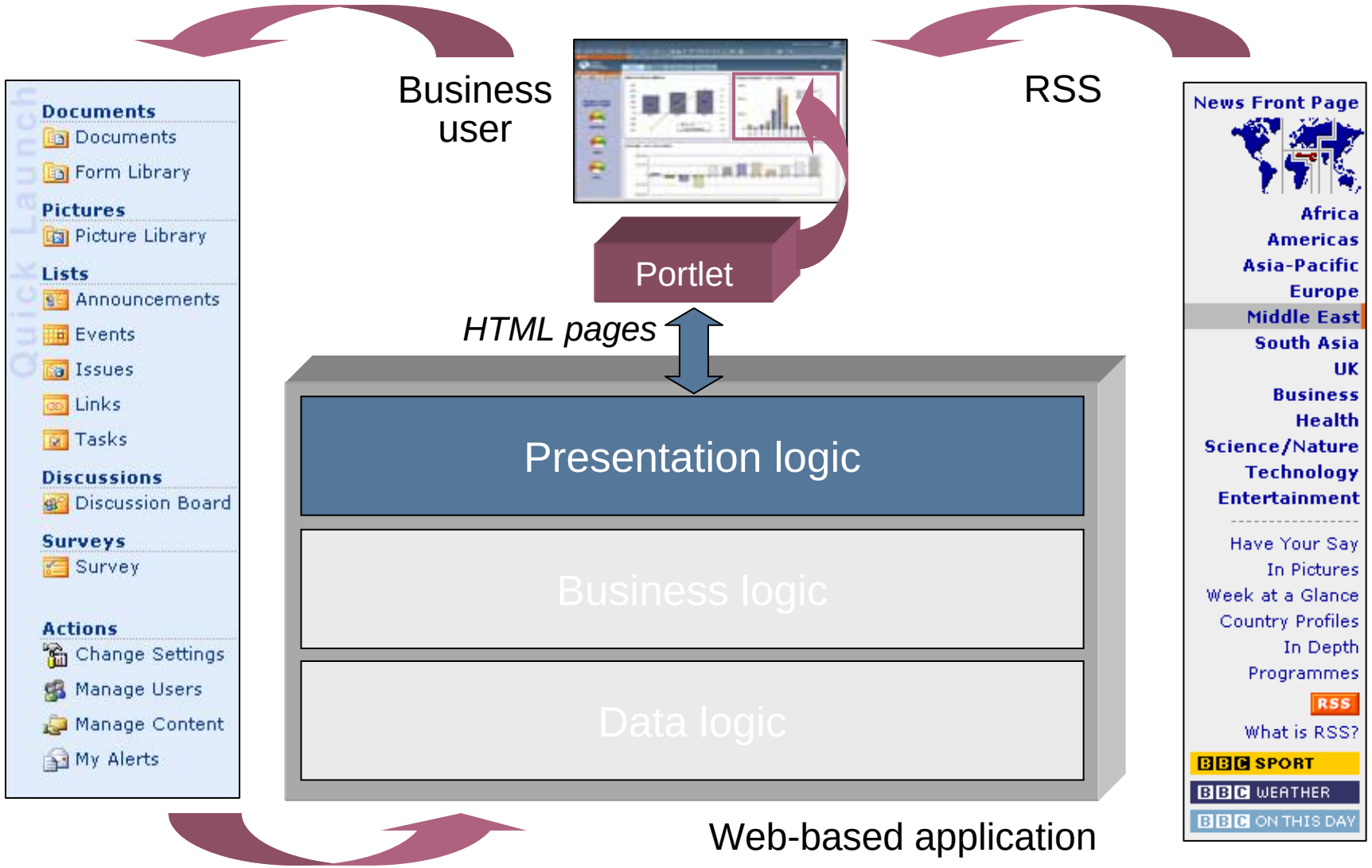
Choosing the Right Portal Development Approach

- Approach chosen will depend on:
  - Amount of development work required – the cost of this may offset savings of using custom-built or open source solutions
  - Existing IT environment and application development strategy
  - Best-of-breed versus single platform/vendor company strategy
- Some companies may deploy several different portals
  - Can be connected together in a federated/distributed architecture
  - Metadata management becomes the key issue with a federated/distributed approach
- Open interfaces, standards and a service-oriented architecture (SOA) are important for ease of integration regardless of the approach used

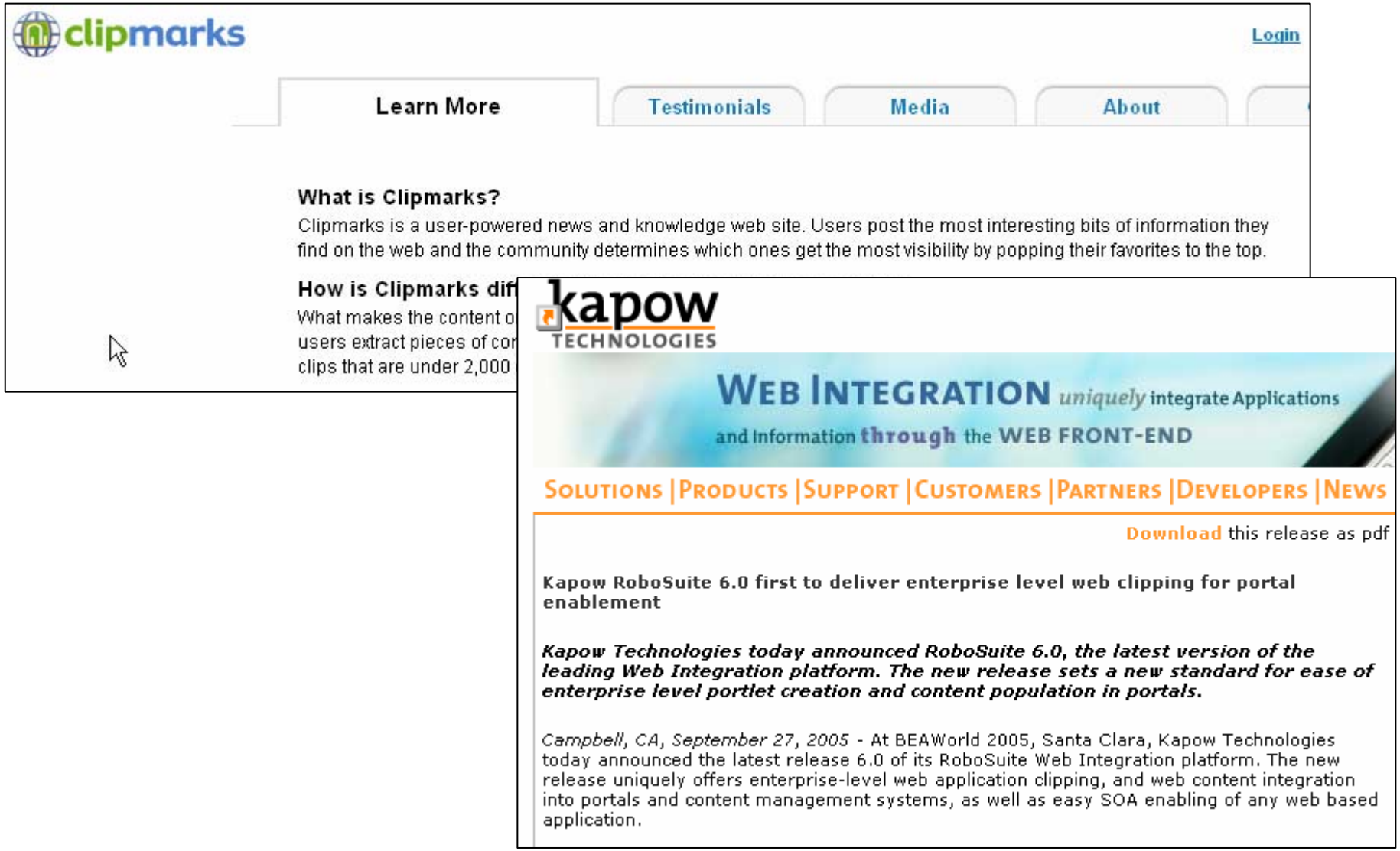
Application Architecture Revisited



User Interaction Integration: Quick Launch, Web Clipping, RSS



Third-Party Web Clipping Products



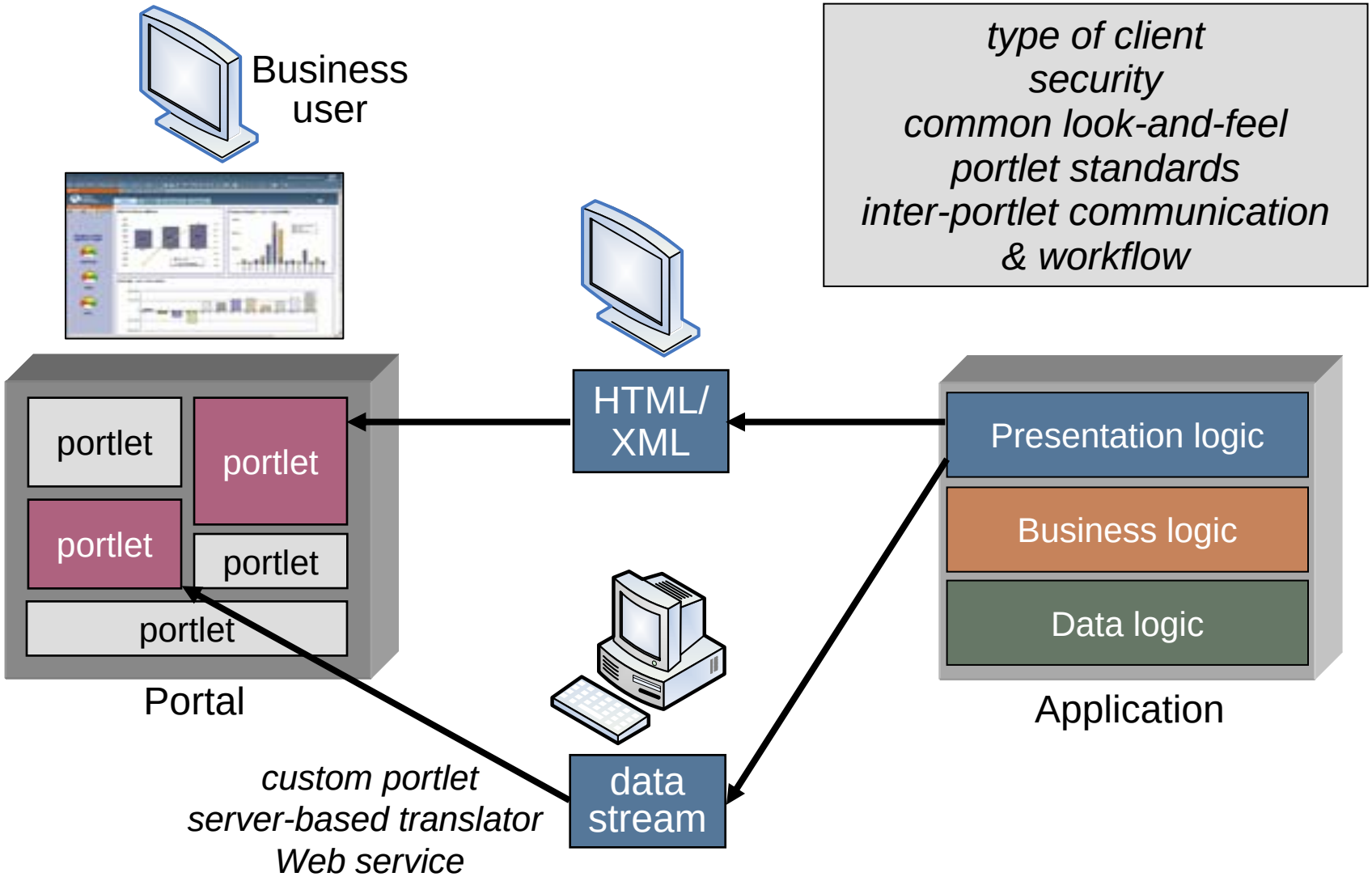
RSS in the Enterprise

RSS in the Enterprise: Application Examples

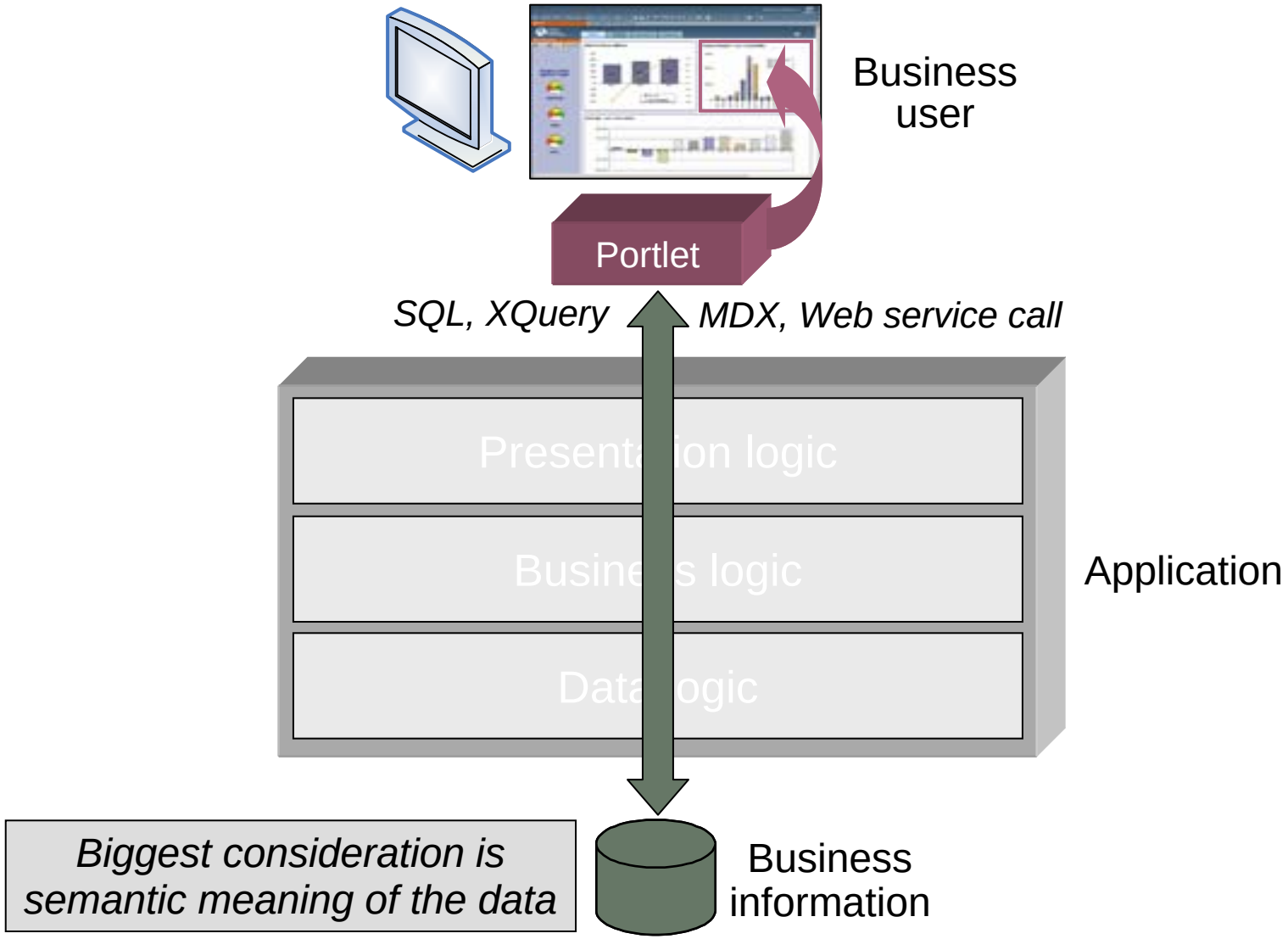
- Enterprise wide content distribution, e.g., reports
- Direct mass communication with employees
- Filtering industry news for relevancy and making it available to appropriate teams and employees in need of specific content
- Targeted and secure in-house communications between employees and teams
- RSS feed publishing to communicate with prospects, customers, business partners, the media and other target audiences.
- Republishing syndicated content on corporate sites

Source: marketingstudies.net

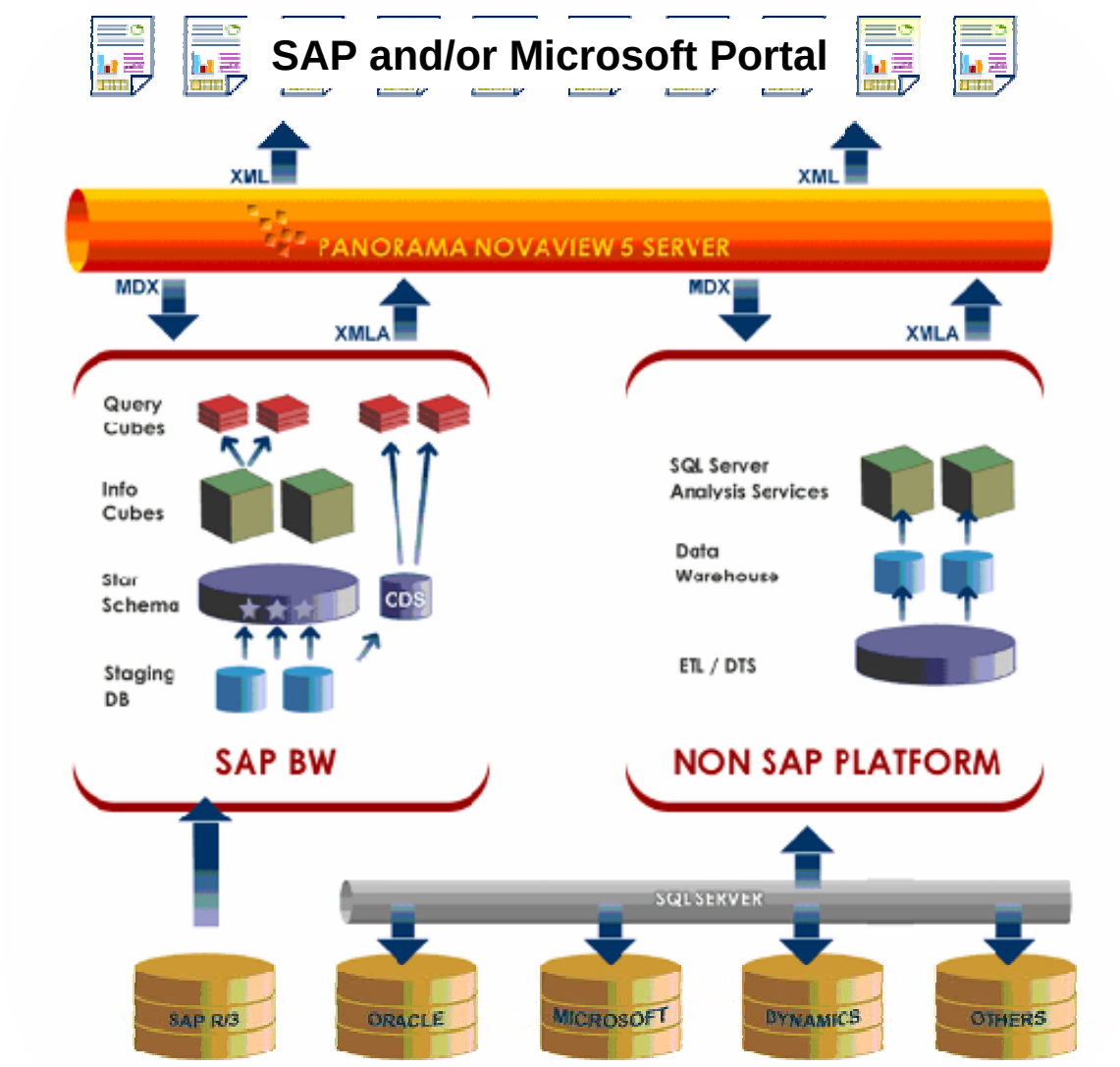
User Interaction Integration: Considerations



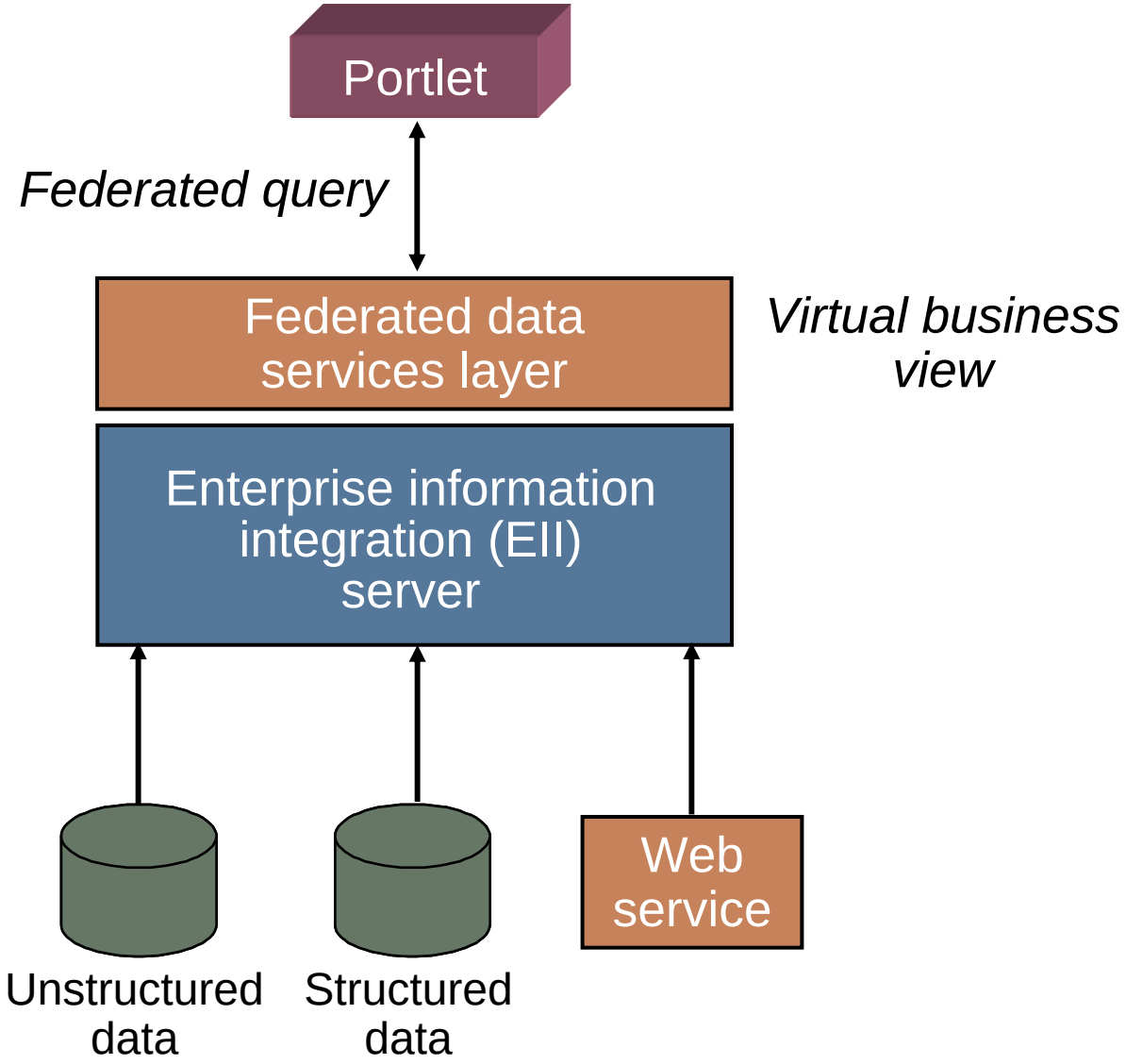
Data Integration



Product Example: Panorama Software



Data Integration: Federated Data Access

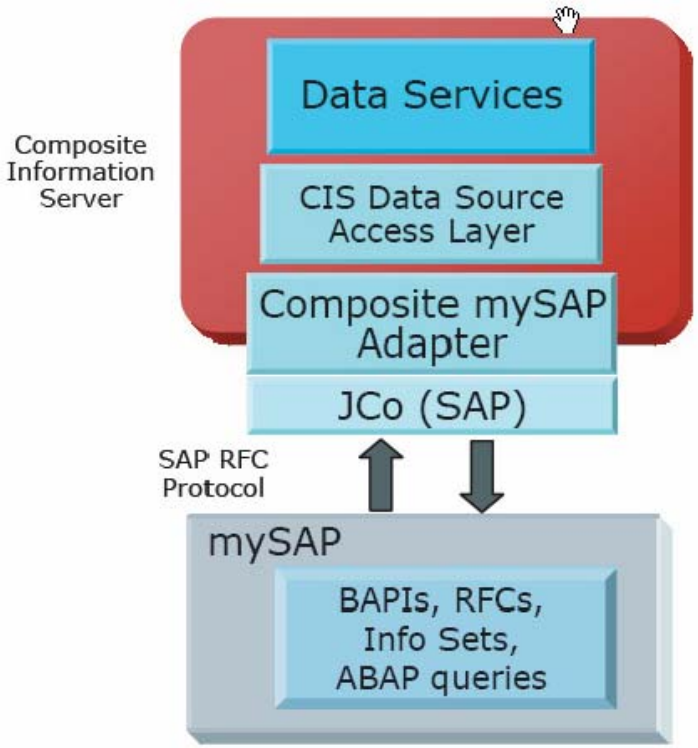


EII and EII-Like Product Examples

- DBMS Vendors
  - IBM WebSphere Information Server (Federation Server)
  - Sybase (acquired Avaki)
  - RDBMS materialized views, virtual cubes, distributed data features
- EAI vendors
  - BEA AquaLogic Data Services Platform
  - webMethods Enterprise Services Platform
- EII and BI vendors
  - Actuate Information Objects (acquired Nimble Technology)
  - Business Objects Data Federator (acquired Medience)
  - Composite Software Information Server
  - iWay Software Enterprise Data Hub
  - MetaMatrix Enterprise
  - …

Accessing Application Package Data: Composite CAV

CAVs for mySAP



The Composite architecture allows you to query and combine common mySAP data structures using certified interfaces. The complexities of the mySAP APIs are resolved by the Composite mySAP adapter, allowing you to transparently access both the data and metadata in a relational format. Then the Composite Information Server allows you publish the mySAP content as data services, readily consumable as ODBC/JDBC or Web services.

SPECIFICATIONS

Composite Application Views:

Application Sources

- mySAP 4.6, 4.7, 5.0

Other CAVs:

- Salesforce.com
- Siebel eBusiness Aplications 7.x
- Oracle E-Business Suite 11.5.8 ,11.5.9, 11.5.10

Composite Information Server:

Software Requirements

- Composite Information Server 3.5 or higher

Access Interfaces

- SOAP 1.1, JDBC 2.0, ODBC 2.x

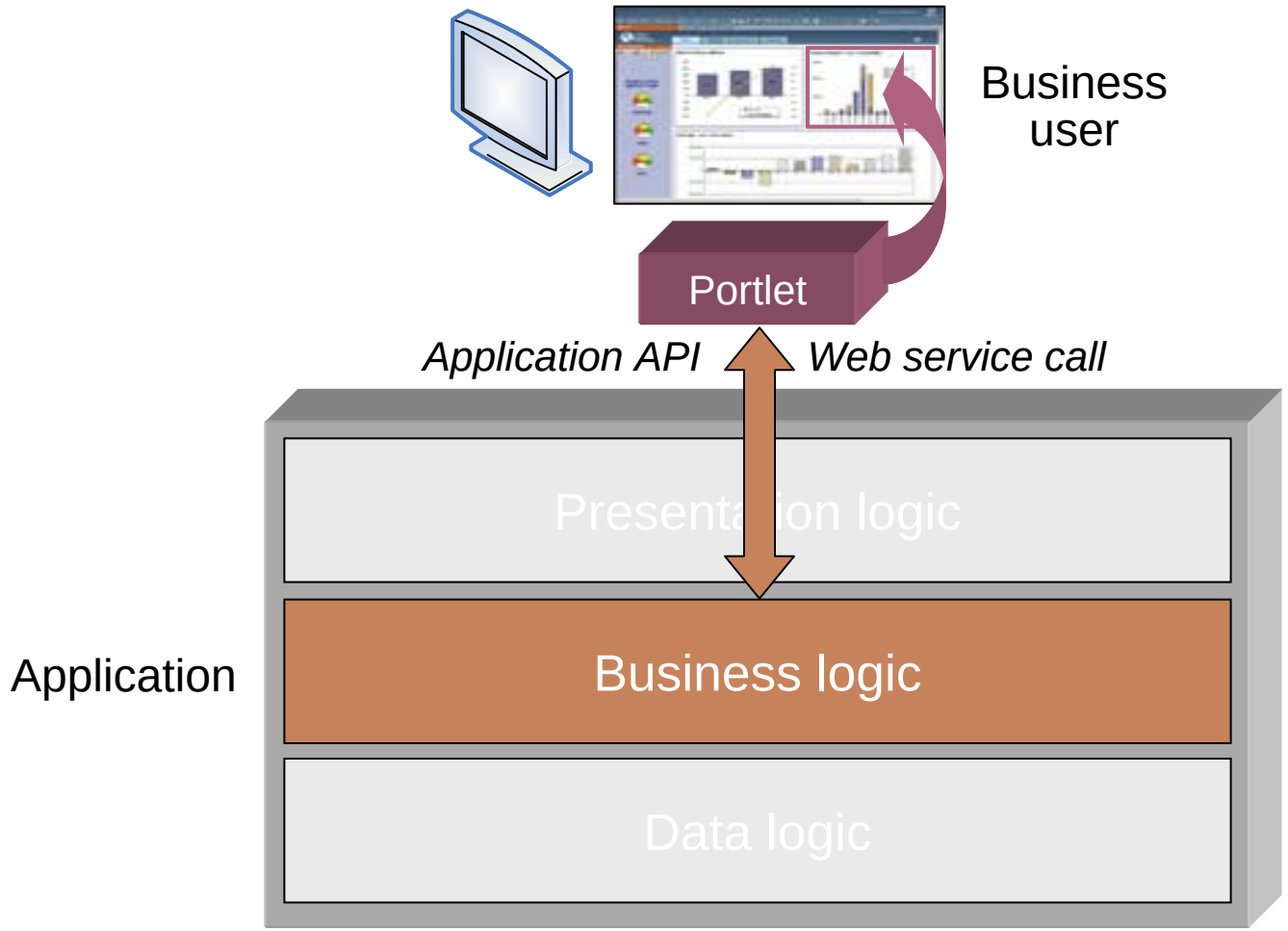
Standards

- JDK 1.4, J2EE 1.3, JAAS, JNDI
- WSDL 1.1, SOAP 1.1, XSLT 1.0, XML Schema 1.0, SAX and DOM
- SQL 99
- Unicode support

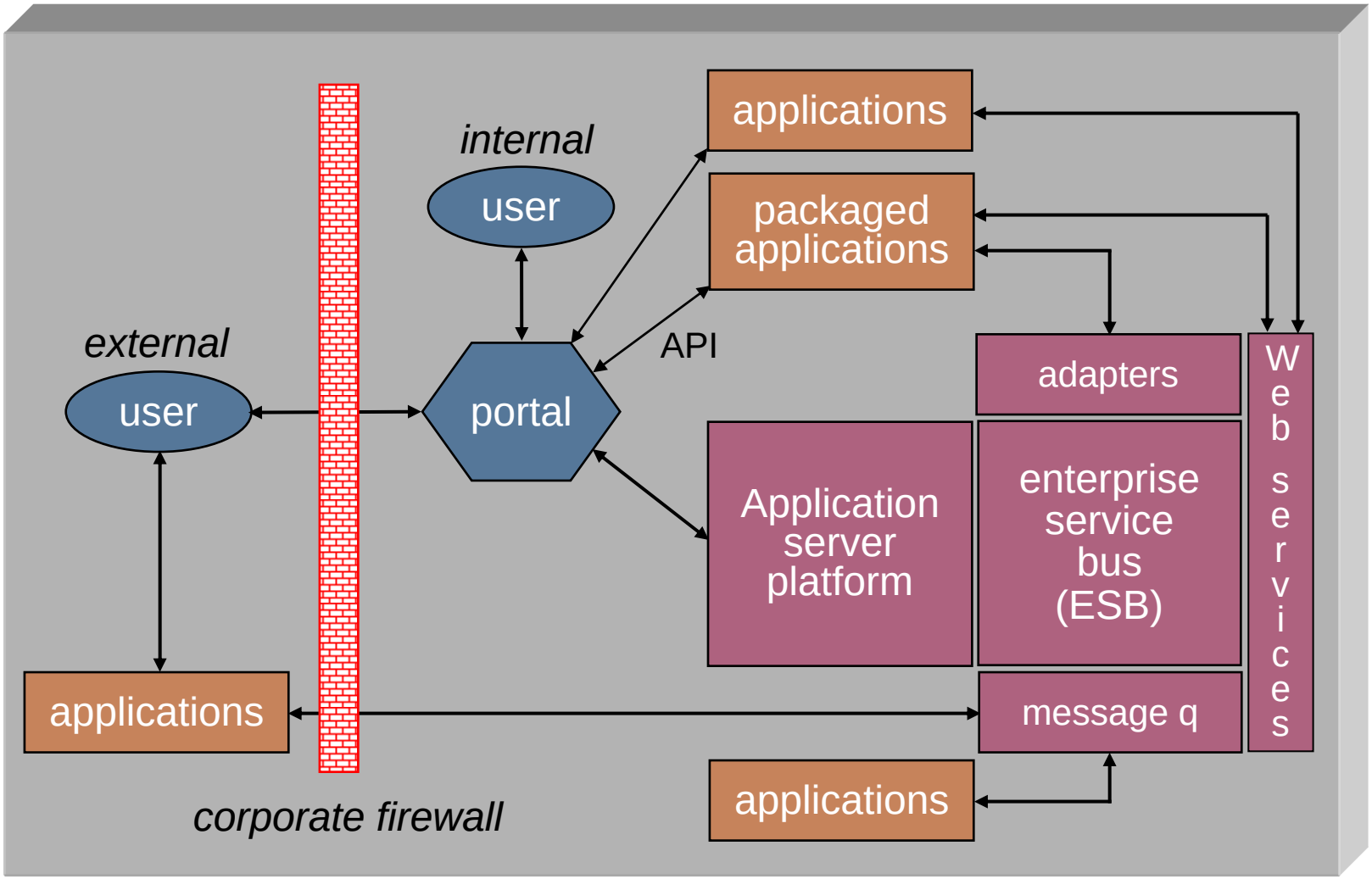
Data Integration: Accessing Documents (Examples)

- Web URL plus viewer
  - Microsoft Office products
  - Adobe PDF
  - Microsoft Office 2007 XML (OpenXML) - ECMA
  - OpenDocument Format (ODF) - OASIS
  - XML with industry vocabulary, e.g., XBRL, RSS
- Content management system API and protocol
  - Native API
  - Standard API (JSR 170)
  - WebDAV (IETF HTTP extension)

Application Integration

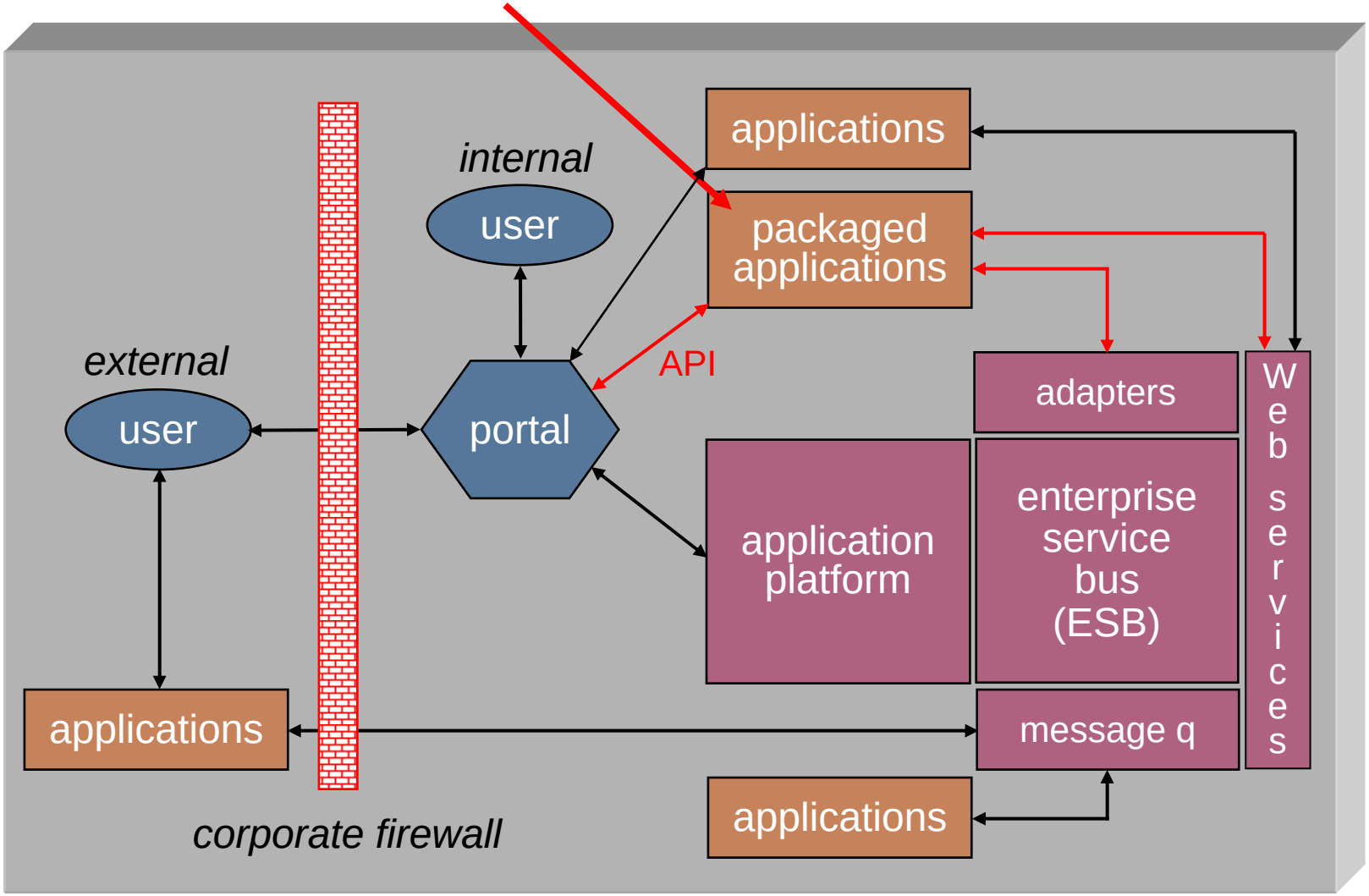


Approaches to Application Integration

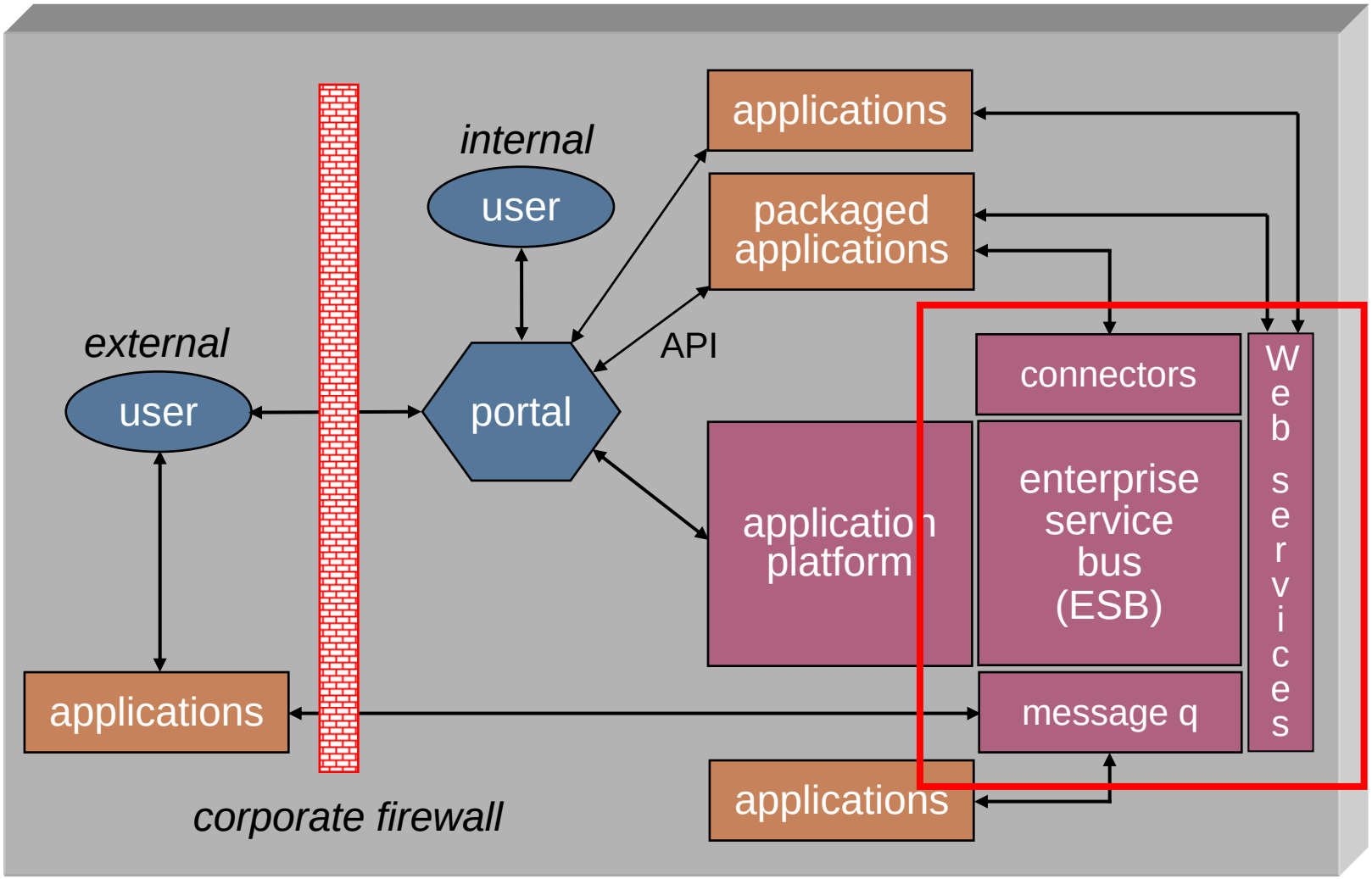


Application Integration: Packaged Applications

May be supported by portal frameworks, ESB adapters or Web services



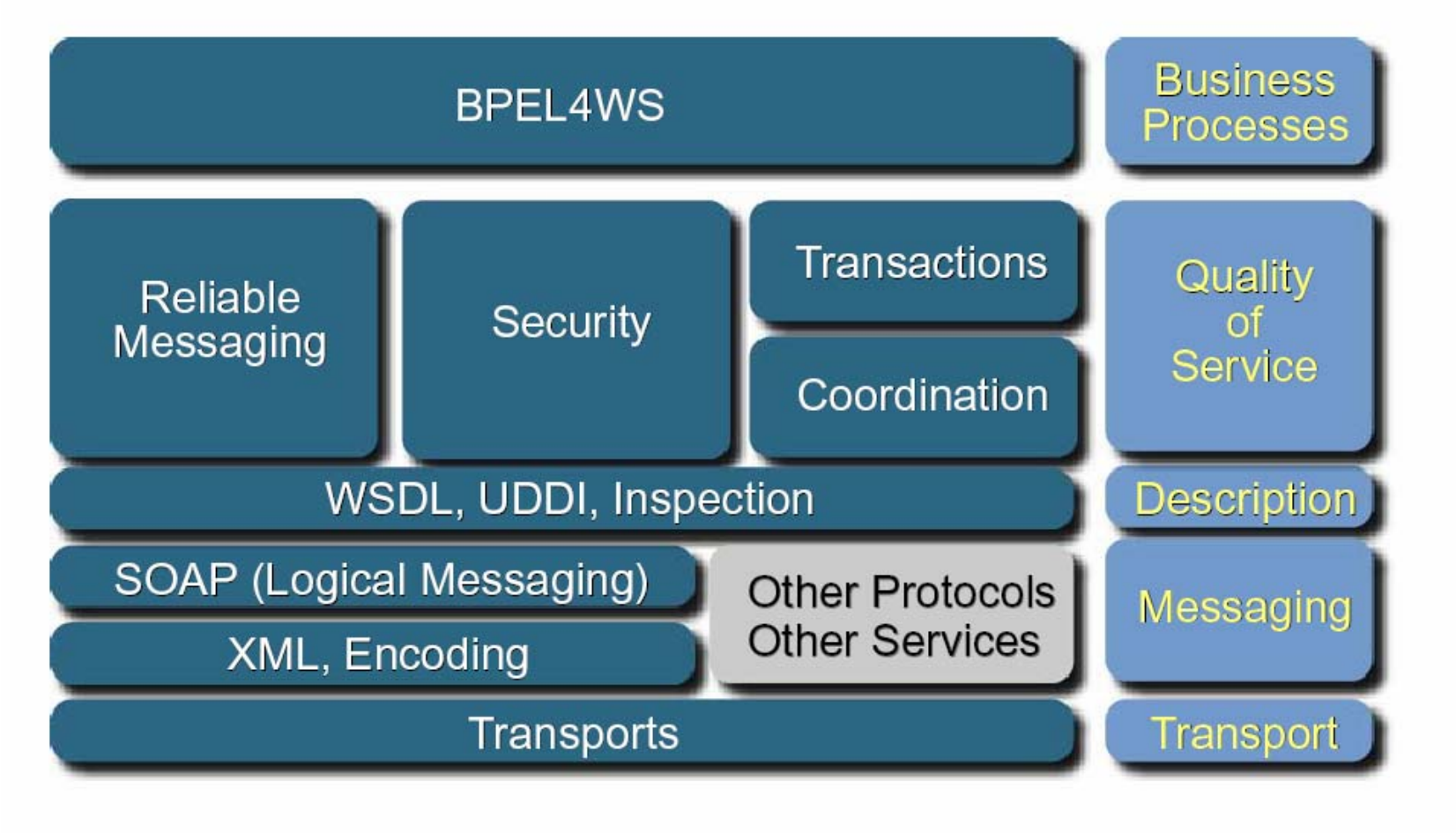
Application Integration: Services Oriented Architecture



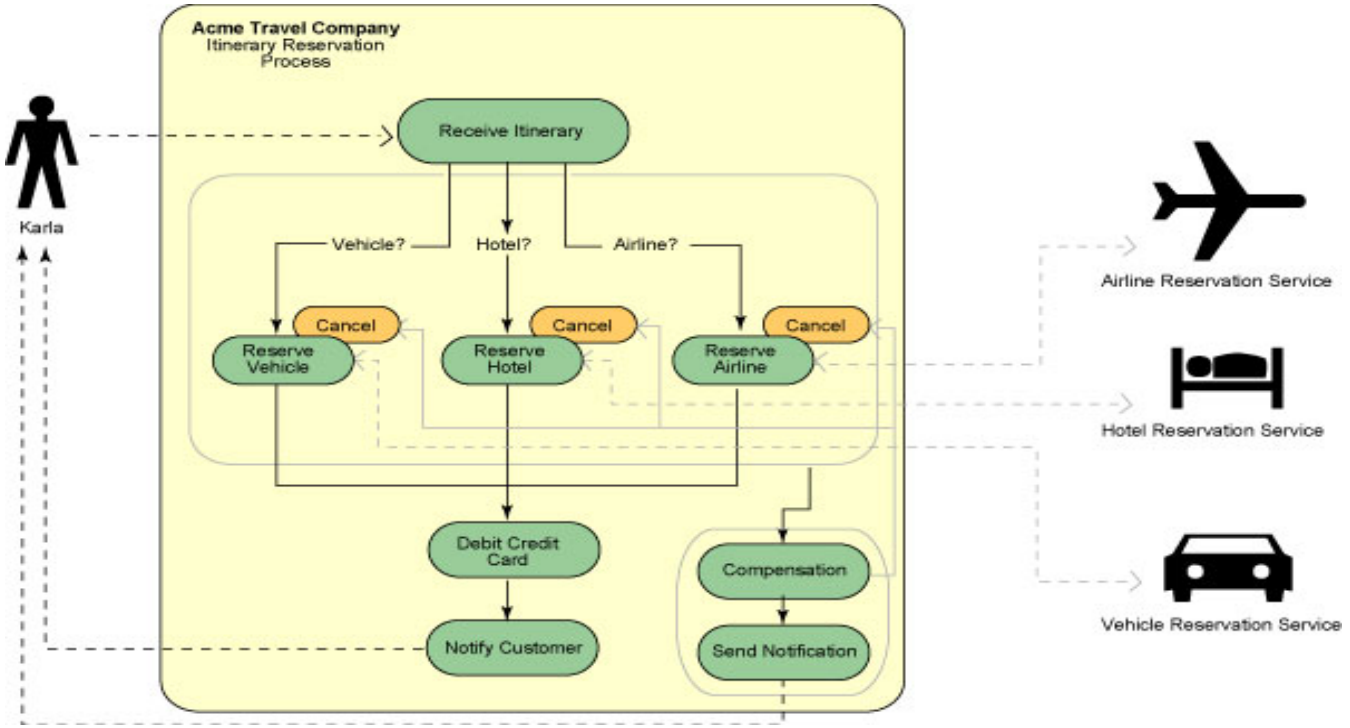
What is a Services-Oriented Architecture (SOA)?

- An SOA supports a network of loosely-coupled *components* (e.g., applications, shared services) that can be interconnected using common and open standards
- Early attempts at supporting an SOA were proprietary and complex to implement
- The advent of Web services and XML-based protocols has made a SOA more viable
  - Services and protocols are easier to implement
  - Based on open standards
- Existing applications and content can be wrapped and presented as a Web service, which supports an orderly migration to a SOA
- However, the use of Web services is not a prerequisite, could use JCA connectors, for example

The Role of Web Services

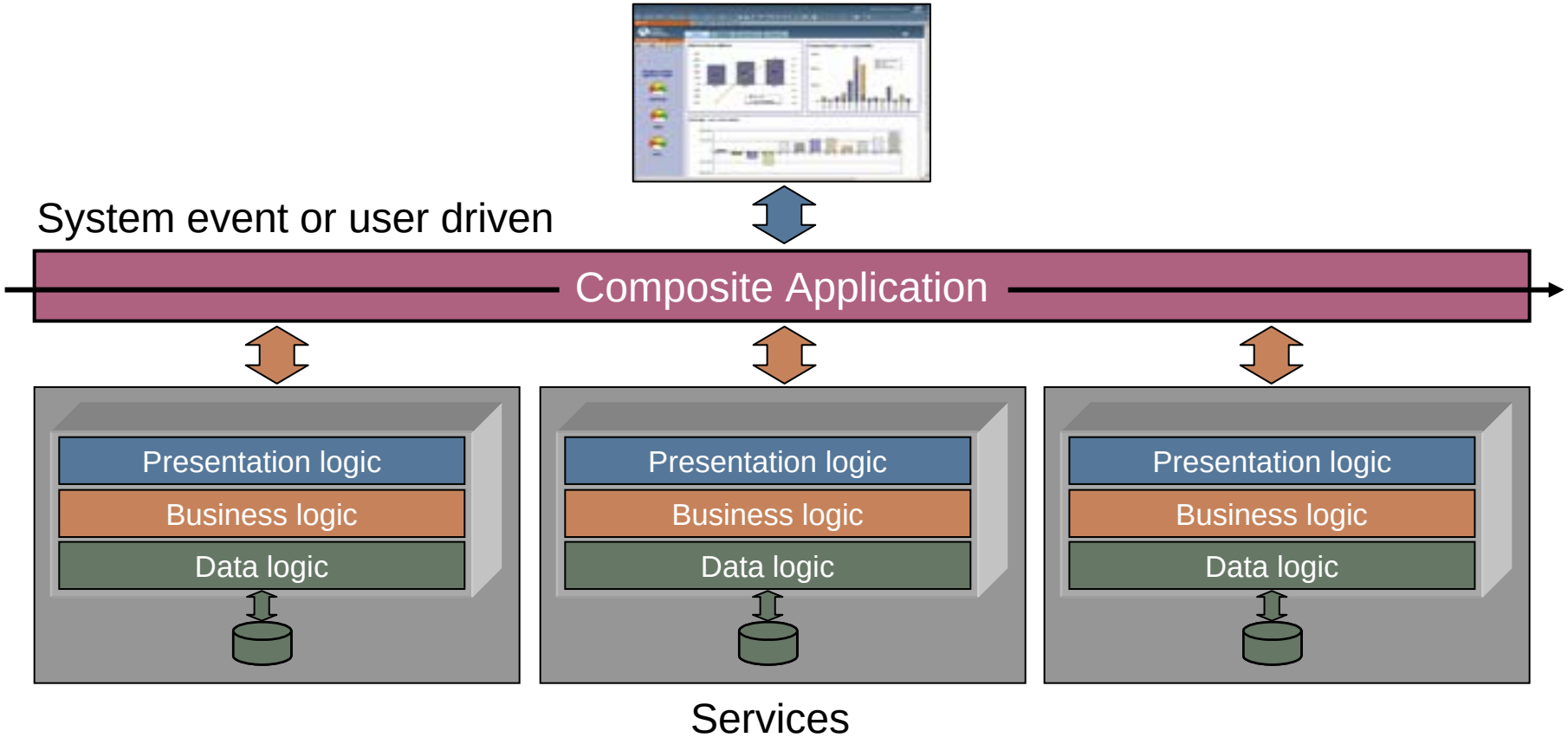


An Example of What Are We trying to Do With a SOA



- Defines the integration of reservation services for airline, hotel and car rental partners in a business process
- Defines how activities can be externalized as Web services
- Coordinates activity of multiple Web services
- Dynamically links to services from multiple providers at run-time based on data derived from process flow itself (e.g., the airline the customer wishes to use)

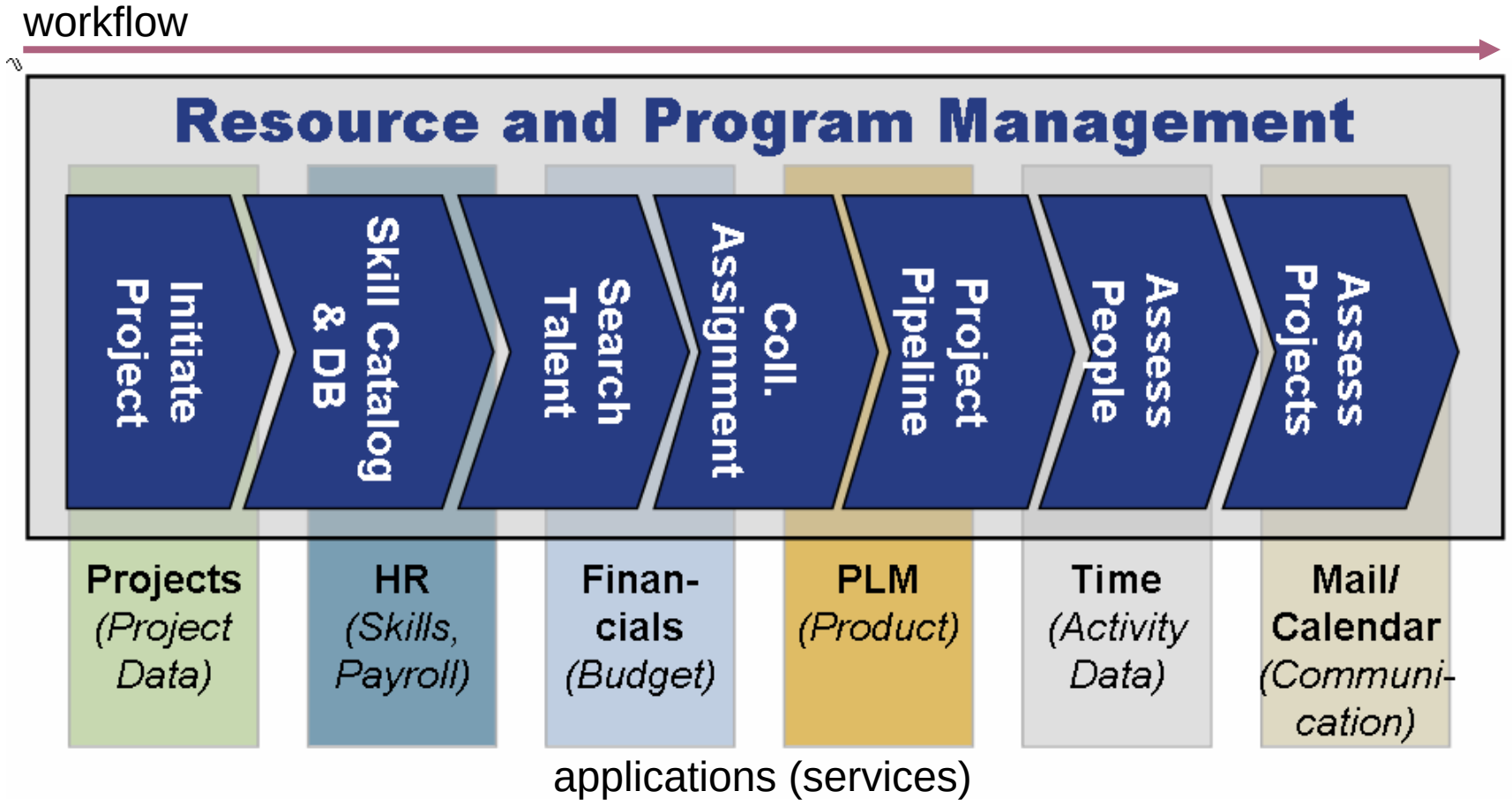
Process Integration: Composite Applications



Could be of a series of interconnected portlets or a business process workflow

Involves competing frameworks (e.g., OASIS WS-CAF) and standards covering context, coordination, and transaction management

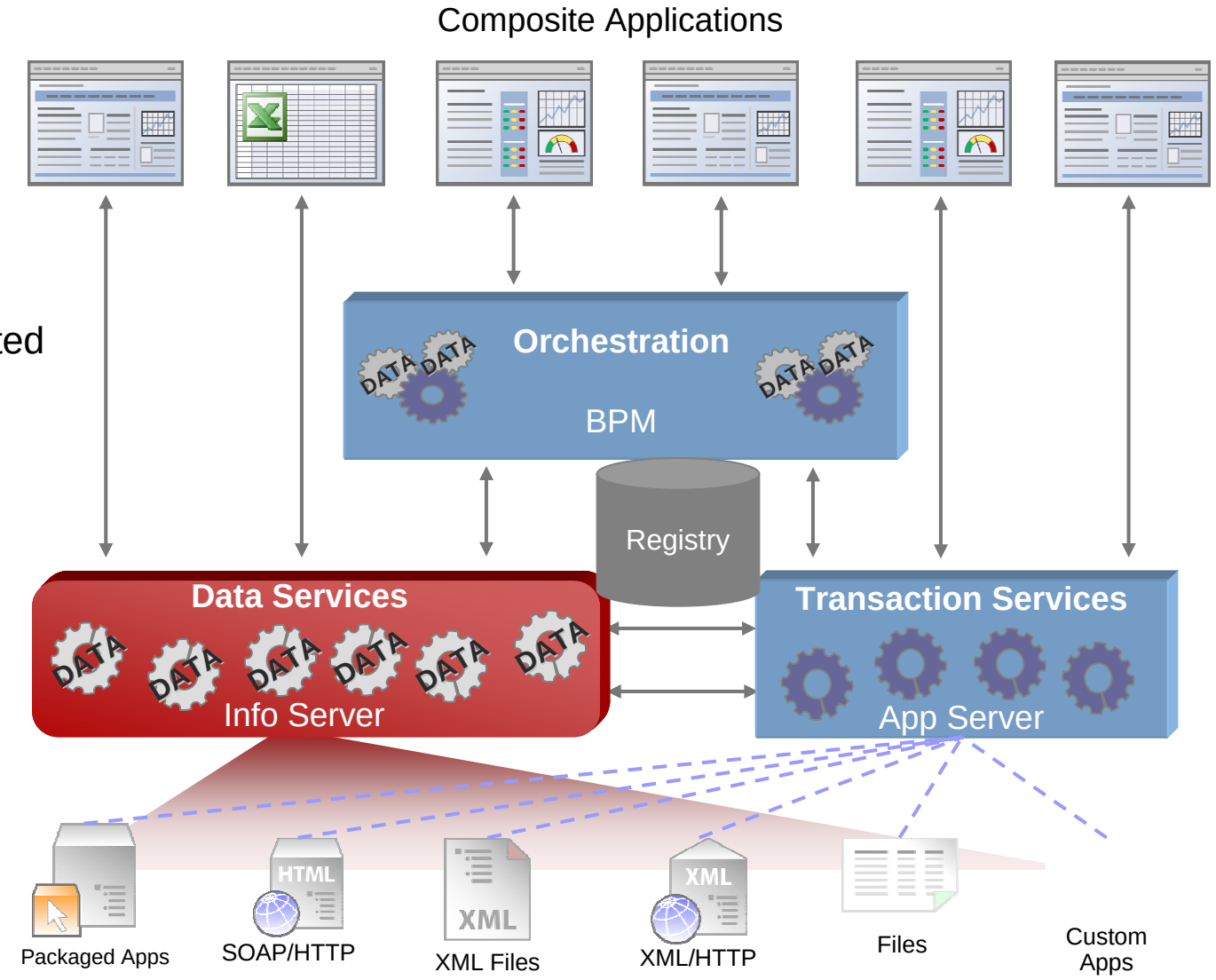
Composite Application Example: SAP xRPM



Supported by the SAP Composite Application Framework (provides a model-driven architecture, object access layer, collaborative context, user interface patterns and guided procedures)

Data versus Application versus Process Integration

1) Data services stand alone
2) Transaction services stand alone
3) Data services integrated with transaction services
4) Data services with orchestration using a business process management tool
5) Data services WSDL stored in a UDDI registry



Source: Composite Software

Business Integration Approaches: Considerations - 1

- User interaction integration
  - Three types: Web, desktop GUI, legacy terminal
  - Web interface (HTML web clipping) easy to do, other interaction interfaces more difficult
  - Limited control over user interface operations
  - Provides a basic level of integration
- Business process integration
  - Requires documented and supported application or services interface
  - Ideally suited to business process workflows, composite applications and a service-oriented architecture (SOA)
  - Good control over user interface operations and look-and-feel
  - Best overall solution

Business Integration Approaches: Considerations - 2

- Application integration
  - Similar considerations to business process integration
  - Dividing line between application and business process integration becoming “fuzzy”
- Data integration
  - Many “standardized” database and file interfaces (e.g., ODBC, JDBC), protocols, and technologies (e.g., enterprise information integration, EII)
  - Requires documented data model
  - Easy to implement and provides good control over user interface operations
  - Best suited to information retrieval and search rather than data entry or update

Take-Aways

- Many different approaches to developing a portal and accessing underlying business content
- Often the choice is between software costs versus development costs, but corporate software strategy is a major factor
- Organizations have a plethora of business content sources, and may have more than one portal
- Open interfaces and standards improve interoperability and reduce development and maintenance effort
- Web services hold much promise for supporting a SOA and accessing legacy applications
- Standards like JSR 168/286, JSR 170 and WSRP are very important, but their use may require a trade off between functionality and portability