Dynamic and Mobile
Federated Business Process Execution

A WebV2 Whitepaper

December 2003
Version 2.2

WebV2, Inc.
510 Logue Ave
Mountain View, CA 94043
telephone: (650) 941-5116
www.webv2.com
sales@webv2.com

1. Introduction

WebV2 enables highly flexible, dynamic, federated business process management. Today's centralized business process management systems can deliver great benefit to the enterprise but are often difficult to deploy in federated environments, costly to change, and poorly integrate the mobile user. WebV2 enables business process management systems to be cost effectively scaled across the extended enterprise by enabling:

- Intelligent mobile user business process participation,
- Federated, distributed business process execution,
- Loosely-coupled connectivity of business processes to applications, and
- Low barriers to insertion – complete BPM solution or complementary to existing BPMs.

WebV2 exploits loose-coupling to enable a flexible solution for executing dynamic business processes and connecting them to applications and users.

Loose-coupling is essential in business process management for supporting highly dynamic business processes (business processes that change) and for supporting business agility. Among the many definitions of loose coupling, we refer to the ability to develop, deploy, and execute business processes independently from the underlying issues of domain-independent processes, control flow, connectivity to applications and users, and system architecture. In particular, there are many domain-independent processes that support enterprise functionality, such as handling security, and finding services and applications. For example, it should be possible to dynamically:

- introduce new applications (with potentially new interfaces),
- change message transport protocols, and
- change security requirements and procedures

… all in a running system without changing the top-level business process.

In all cases, there are many interdependencies among the various components required for setting up and seamlessly executing business processes. Dynamic business process creation and management requires an underlying connectivity solution that supports dynamic connectivity and managing dynamic interdependencies.

Today's enterprise needs to effectively manage, adapt, and scale its business processes to new requirements. An effective Business Process Execution (BPE) solution must comprehend:

- The extended enterprise: Internal divisions, customers, partners, and suppliers.
- Disparate IT systems: Legacy systems, application servers, portals, Web Services, as well as existing EAI, B2Bi, and BPM solutions across different vendors.
- Changes in the underlying systems: Replacing old systems with new; coping with new standards and architectures; adding new partners.

- Changes in the business processes: Replacing old processes with more efficient ones; incorporating and/or integrating with processes of customers, partners, and suppliers; adding and/or changing subprocesses.
- Diverse enterprise and cross-enterprise topologies: Dealing with centralized as well as decentralized environments.

Current Business Process Management and Enterprise Application Integration systems are tightly coupled and are, consequently, very rigid and costly.

WebV2 introduces a cost-effective, flexible, loosely coupled runtime Business Process Execution solution that fulfills the above requirements. This solution is available either stand-alone or as part of an existing EAI/BPM framework.

2. Business Process Management

Current approaches

Business Process Management (BPM) refers to the coordination of a number of tasks in order to achieve certain business goals (cf. Figure 1). These tasks are carried out by different nodes in a network (intranet, Internet, or both), including enterprise applications and humans. Any given node may execute one or more different tasks. Often, the results of a task determine which tasks should be executed next. The results of a task may also need to be propagated to subsequent tasks. Typically, BPM solutions define which tasks should be executed, which nodes should execute the tasks, as well as the order and dependencies among the tasks. The execution of a task by an application node is triggered by a corresponding invocation to its API or data adapter.

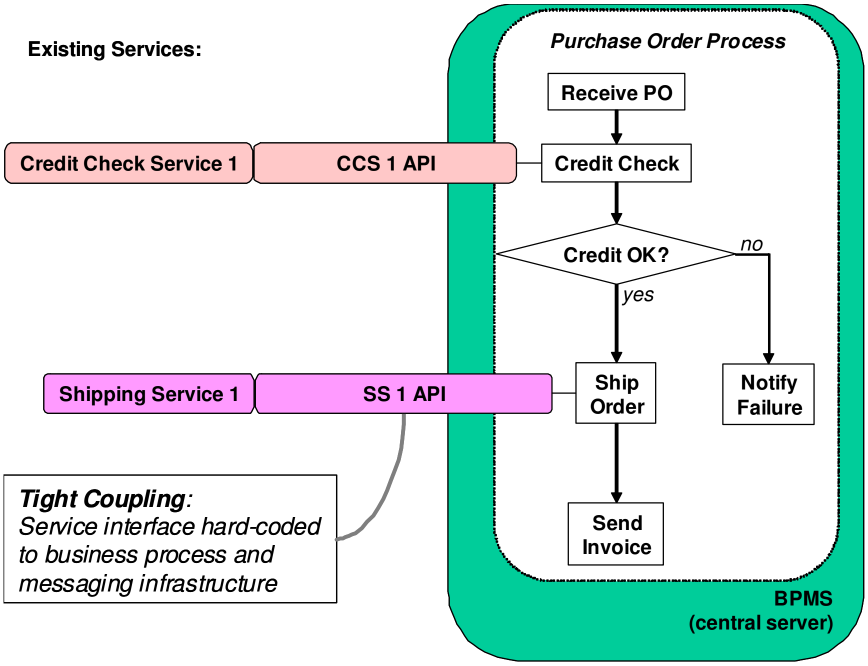


Figure 1: Traditional Business Process Management System

Business Process Management is an important component of Enterprise Application Integration, both within and across firewalls. The increasing adoption and use of open standards and protocols, such as XML and Web Services, is making it easier to provide the underlying data and transport integration infrastructures. Enterprises can now concentrate on the BPM issues. BPM is also closely related to Workflow, where Workflow systems concentrate more on human user involvement. However, many of the same problems exist, especially as BPM and Workflow solutions merge. Existing EAI vendors have been required to incorporate BPM approaches into their offerings; however, they are currently very tightly coupled to the specific EAI solution. On the other hand, independent BPM vendors need to deal with underlying connectivity issues, without having to rely on an expensive EAI solution.

A Business Process Management System (BPMS) is a software program that executes a business process by coordinating the execution of the tasks and supporting the transfer of task results to other tasks. As such, a BPM solution must define not only the tasks that should be executed and their dependencies (*what-to-do*), but also the procedures and control of the business processes (*how-to-do-it*). Traditionally, the *how-to-do-it* is implicitly hard-wired in the code that specifies *what-to-do*. This is particularly problematic in integrating with the external world and in supporting interaction among components. There is no high level model of interaction accessible to the business analysts. Rather, the interaction must currently be hardwired in low level code. This is especially the case when considering that there are many underlying domain-independent processes that are used repeatedly, and are currently hard-coded into business processes.

Currently, several languages have been proposed that support a more abstract representation of the business process (BPML, BPEL). However, current implementations supporting these languages are (partly by the very nature of the languages) still inherently linked to the system architecture, interfaces, and control of application interfaces.

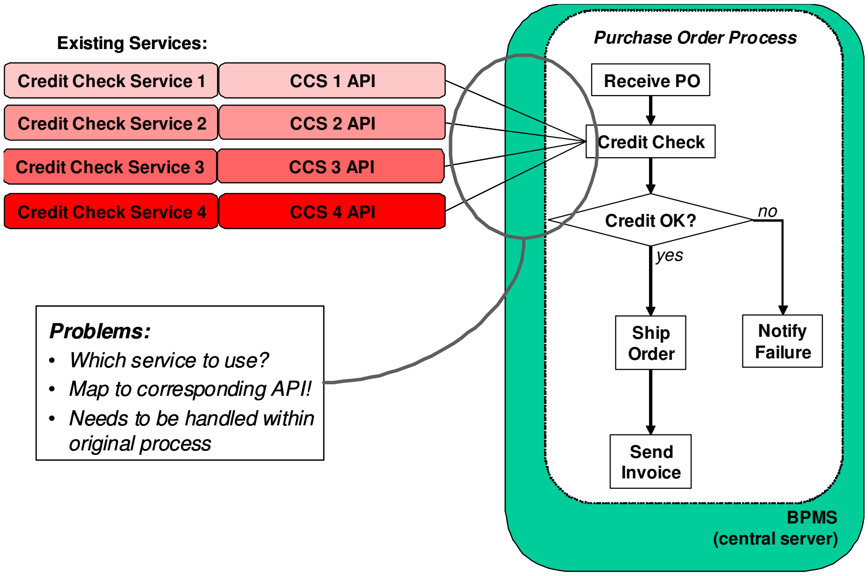


Figure 2: Adding new services to a BPMS

However, a number of problems arise if additional services capable of executing the same task are added, or if the API of a service changes (cf. Figure 2). Traditional BPM Systems need to be reprogrammed to take such changes into account, thereby altering the original business process. This can be very time consuming, error prone, and expensive.

3. WebV2's Business Process Execution Solution

Enabling loose-coupling through ProcessCouplers

WebV2's Business Process Execution solution allows abstracting the domain-specific details of a business process from the domain-independent mechanisms used to carry out the process. This provides a number of advantages:

- The business process owner can concentrate solely on defining the business process, and not worry about underlying technical architectural details.
- Underlying process components providing enterprise functionality can be provided as highly reusable patterns.
- New sub-processes can be added dynamically and independently from the original process. This allows for easily defining additional processes for selecting and negotiating among additional applications.
- Underlying services can be dynamically added, replaced, and/or changed within a running system and be immediately accessible – without changing the business process or changing any part of the existing runtime system. This greatly simplifies and largely automates change management, both during initial system development and deployment, as well as in ongoing change.

Furthermore, the resulting system is independent of the following IT choice points:

- Underlying message transport infrastructure (e.g. asynchronous or synchronous; point-to-point or broadcast; TCP/IP, HTTP, JMS, or other third-party offerings)
- Security and authentication procedures (e.g. use of Public Key Infrastructure)
- Transaction and logging procedures
- Architecture – where subsystems are actually deployed.

This means that any desired IT approach can be integrated (at any time) without changing the original business process.

WebV2's solution is achieved by logically wrapping participating nodes in the system with lightweight Java-based *ProcessCoupler*™ software (referred to as Process Couplers, cf. Figure 3). Process Couplers provide a runtime environment abstracting the node-specific interfaces and underlying enterprise functionality from the rest of the system.

The WebV2 Process Couplers have the following functionality:

- Explicit representation of the state and other contextual parameters of the process.
- Lightweight, local process engine for executing sub-processes and participating interaction patterns.
- Asynchronous or synchronous processes execution.
- Capability to reason dynamically about the service invocation, delaying, or delegating it as necessary.
- Capability to select the message transport mechanism dynamically from a customizable set of available mechanisms.

- Capability to determine dynamically the best means of finding and engaging the most appropriate service providers, based upon required functionality (broadcast, broker) and Quality of Service (QoS) preferences.
- Capability to automatically handle exceptions and service failures.

The communication and coordination framework among the WebV2 Process Couplers supports not only

- direct method invocation and
- stream-based messaging

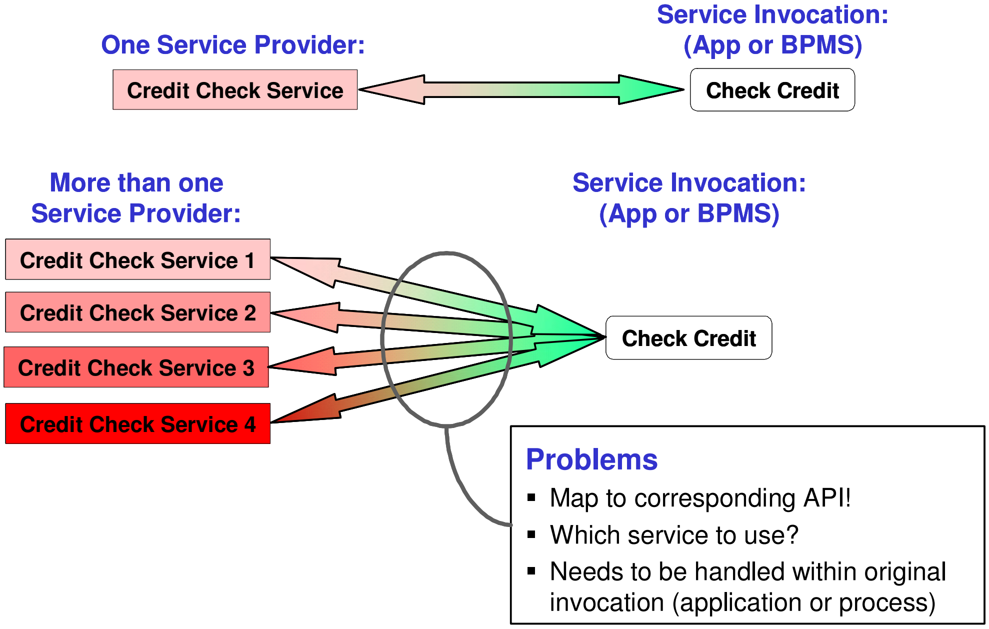
but also

- transferal of the process,
- transferal of the state of the process,
- high-level primitives acting upon the service request, and
- customer- or supplier-specific coordination and interaction protocols.

Invoking Several Services

Whether or not a service request is initiated from a BPMS or another application, problems arises when there are several service providers actually able to fulfill the service request:

- The service providers must be located
- It must be decided which service providers are to be used.
- The service invocation must be mapped to the interfaces of the different service providers
- All of this needs to be handled by the originating application or BPMS



Problems
- Map to corresponding API!
- Which service to use?
- Needs to be handled within original invocation (application or process)

Figure 3: Problem with Tightly Coupled Interfaces

With the Process Coupler approach, each service provider maps to a corresponding Process Coupler (cf. Figure 4). The Process Couplers use specialized Interaction Protocols, described in the next section, to automatically handle service location and interface mapping in a dynamically changing environment, independently from the original domain business process.

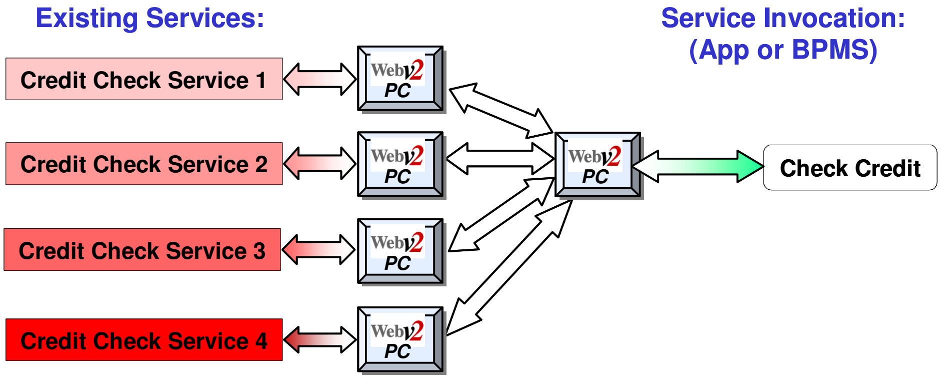


Figure 4: Process Couplers Support Loosely Coupled Interfaces

Enabling decentralized process execution

Furthermore, Process Couplers are actually able to directly execute a business process, without using a centralized BPMS! (cf. Figure 5) This grid-based alternative, or peer-to-peer approach, reduces reliance on expensive heavyweight servers and communication bottlenecks. It is automatically configuring and adaptable to any IT infrastructure.

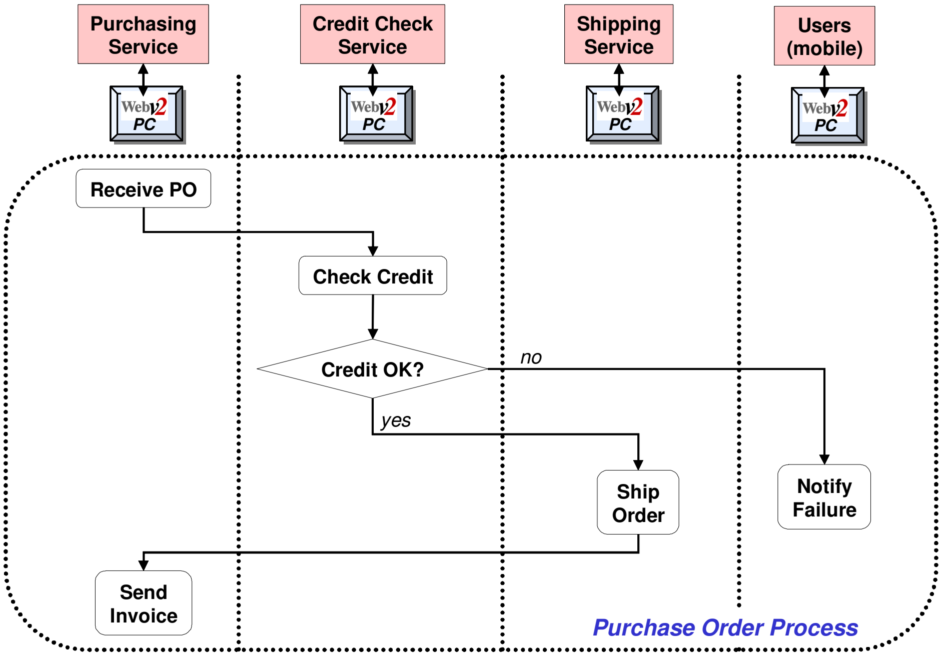


Figure 5: Decentralized Business Process Execution

4. Interaction Processes

Beyond request/response

A powerful feature of WebV2's process execution solution is the ability to support highly reusable application-independent sub-processes that can be used in conjunction with the original business process.

For example, suppose the original business process is designed for accessing a single service provider for a given task. The business may decide that it can gain significant advantage by selecting dynamically from a range of possible service providers, based on certain criteria, such as best price, quality of service, availability, etc. Without changing the original business process, the Process Couplers can dynamically (at run time) select which service to invoke. This is handled by the contracting interaction process (cf. Figure 6).

The interaction processes are built upon the semantically rich communication and coordination standards developed by the Foundation for Intelligent Physical Agents (FIPA).

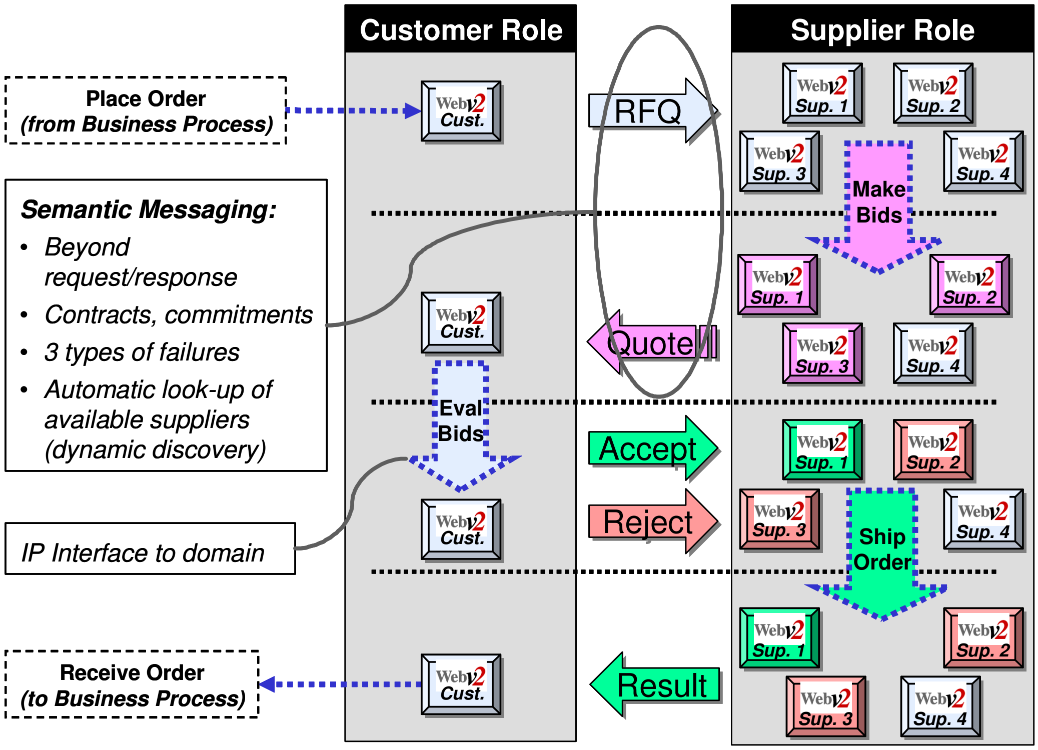


Figure 6: The Contracting Interaction Process

In particular, FIPA defines higher level message types that go beyond the usual request/response interaction. These message types include *agree* (to perform the task), *refuse* (to perform the task), *query* (for information), *inform* (information), *cfp* (call-for proposals), *bid*, *accept* (a bid), and *reject* (a bid). FIPA also specifies a means for representing possible sequences of such message types into Interaction Protocols (IPs), and a library of common IPs, such as *FIPA-request* (similar to client-server) or *FIPA-contract-net*.

Interaction processes allow for

- interaction about the task to be performed, before it is actually invoked,
- sharing of the context of the process, and
- incorporation of transaction, security, and monitoring processes, without changing underlying interfaces, arguments, or processes.

The rich set of communication forms and interaction processes allow business analysts to specify easily and rapidly the services they want and the way they want them delivered.

Furthermore, interaction processes can be dynamically nested, allowing seamless incorporation of new processes and contexts supporting enterprise functionality, such as those managing security, authentication, or transactions. This is accomplished without changing the underlying application services or their interfaces.

Interaction processes cover the following functionalities

- Enterprise functionality (e.g. security, two-phase commit)
- IT functionality (e.g. monitoring, control)
- Process execution strategies (e.g. round-robin, multi-server)
- Negotiation mechanisms (e.g. sealed-bid contracting, auction)
- Service discovery mechanisms (e.g. brokered, brokerless)
- Underlying exception-handling guidance

5. Mobilization of Business Processes

Occasionally Connected Computing (OCC) has been described as a key enabler for the real time enterprise. As mobility of workers and computing environments becomes increasingly more important to the enterprise, work is often done while offline, and must be seamlessly integrated with the enterprise when a connection is re-established. This requires the ability to support asynchronous connectivity to business processes.

WebV2's solution enables the asynchronous participation of applications and human users in business processes. As the solution is lightweight and based on pure Java it runs on heterogeneous platforms, including mobile devices (e.g. PocketPC). This enables easy mobilization of the enterprise and incorporation of mobile devices in the enterprise and its business processes.

A major component of business process automation and workflow solutions is the integration of the human user. Of increasing importance is the ability to support the human user in an asynchronous fashion, as required by mobile and occasionally connected environments. WebV2's Process Couplers can be installed directly on a user's mobile device. This solution provides the following features:

1) Users of mobile devices can be integrated with enterprise business processes in a manner that is significantly less difficult than standard portal-based solutions. All device-specific and user-specific files and information are stored and maintained only by the device. This allows for seamless scalability to thousands of users without any extra hardware resources. A burden with portal-based solutions is that the portal must maintain all communication and interface possibilities for each device and user (e.g. WAP, screen resolutions, …). This in turn inhibits scalability and maintenance.

2) The Process Coupler is aware of the process and its state and context – even while disconnected. The user can interact with the Process Coupler (and thereby in context of the process) while offline. Upon reconnection, the Process Coupler automatically continues the process. This is true even if the IP address of the device changes. Any change in the IP address of the device is automatically re-registered with the Process Coupler network. The Process Coupler also has direct access to any data stored on the user's device. Any delivered information and the display of the information can be tailored to the user's context.

3) Direct device-to-device (peer-to-peer) data and process interaction among users across mobile devices as well as interaction with applications on the fixed net.

4) Multi-modal user interaction via one or more of the following:

   - Existing standard application (e.g. Microsoft Excel): The Process Coupler uses back-end connectivity to the application, transmitting the data to the application, which uses its built-in interfaces for presentation to and interaction with the user.

   - Standard HTML browser: The Process Coupler acts as a local web server. Changing the user interface is as simple as replacing a few html files, and can be done while the Process Coupler is running.

- customized Java application: A dedicated graphical user interface can be built and deployed as a separately running application.

5) Interaction across Process Couplers and with net-based applications via Web Services standards. Process Couplers support UDDI registry, SOAP interaction, and participation in BPEL4WS-based processes.
6) Automatic update of run-time software upon change (from external repository). This can occur either by regular polling or pushing. This significantly reduces maintenance overhead.
7) Ability to interact with software while off-line; when back on-line software resumes interaction with fixed net.

WebV2 currently demonstrates the Process Couplers running on PocketPCs. All that is required is a Java runtime engine supporting the J2ME Personal Profile.

6. Embracing and Extending Web Services

The WebV2 solution embraces and extends current and planned Web Service standards. In particular, the solution leverages standards-based business process languages, such as BPEL4WS (Business Process Execution Language for Web Service) developed by IBM, Microsoft and others, and currently undergoing standardization by OASIS.

Web Services are supported by the WebV2 solution in the following ways:

- A Process Coupler can invoke an external web service to accomplish a task in the process.
- This allows a web service to be encapsulated by a Process Coupler as a back-end application in its own right.
- Process Couplers enable their applications to be offered as web services to the outside world.

BPEL provides mechanisms for representing a business process and for defining the interfaces a web service must provide in order to participate in the business process. Process Couplers are able to execute a process represented by BPEL.

Future work will entail enhancing web services in such a way to enable them to participate in a BPEL-specified business process independently from their specific interfaces. For example, the current BPEL spec describes what changes have to be made when introducing a different messaging paradigm:

> “The example makes the implicit assumption that the customer request can be processed in a reasonable amount of time, justifying the requirement that the invoker wait for a synchronous response (because this service is offered as a request-response operation). When that assumption does not hold, the interaction with the customer is better modeled as a pair of asynchronous message exchanges. In that case, the "sendPurchaseOrder" operation is a one-way operation and the asynchronous response is sent by invoking a second one-way operation on a customer "callback" interface. In addition to changing the signature of "sendPurchaseOrder" and defining a new portType to represent the customer callback interface, two modifications need to be made in the preceding example to support an asynchronous response to the customer. First, the service link type "purchaseLT" that represents the process-customer connection needs to include a second role ("customer") listing the customer callback portType. Second, the <reply> activity in the process needs to be replaced by an <invoke> on the customer callback operation.”
> *from: http://www.ibm.com/developerworks/library/ws-bpel/*
> *Copyright© 2002 BEA Systems, International Business Machines Corporation, Microsoft Corporation, Inc.*

With WebV2’s solution, no reprogramming is necessary, as the Process Couplers abstract the interfaces from the messaging and automatically handle the messaging whether synchronous or asynchronous.

7. Application and Deployment

Applicability of the WebV2 solution

The BPM area covers a broad spectrum of enterprise software and is closely related to Workflow and Enterprise Application Integration. WebV2's solution is equally applicable to these areas and is complementary to existing deployments.

Some sample scenarios are as follows:

- A company has an existing EAI or BPM solution and wishes to incorporate a new department or external partners or customers in its solution without having to reconfigure the entire solution. By using WebV2's Process Couplers, the company can integrate the additional applications quickly and at far less cost than any other approach.
- A company does not yet have an EAI or BPM solution, and wishes to install one realizing that changes will need to be made in the future. WebV2 will work together with existing EAI/BPM solution providers to install Process Couplers along with the EAI/BPM solution. This allows for rapid and efficient change management when the company wishes to expand or enhance its business process solution.

Process Couplers are lightweight (under 200KB of pure Java) and run on any Java Runtime Environment. Process Couplers are automatically configurable and dynamically updatable, drastically reducing maintenance costs over traditional approaches.

The following figures show some sample deployment scenarios.

Figure 7 shows deployment within an enterprise across several silos. There is also no need for enterprises to re-architect existing integration deployments. By associating Process Couplers with the corresponding hub, the deployed system is immediately integrated with the rest of the Process Coupler network. This provides an effective means for linking the islands of integration.

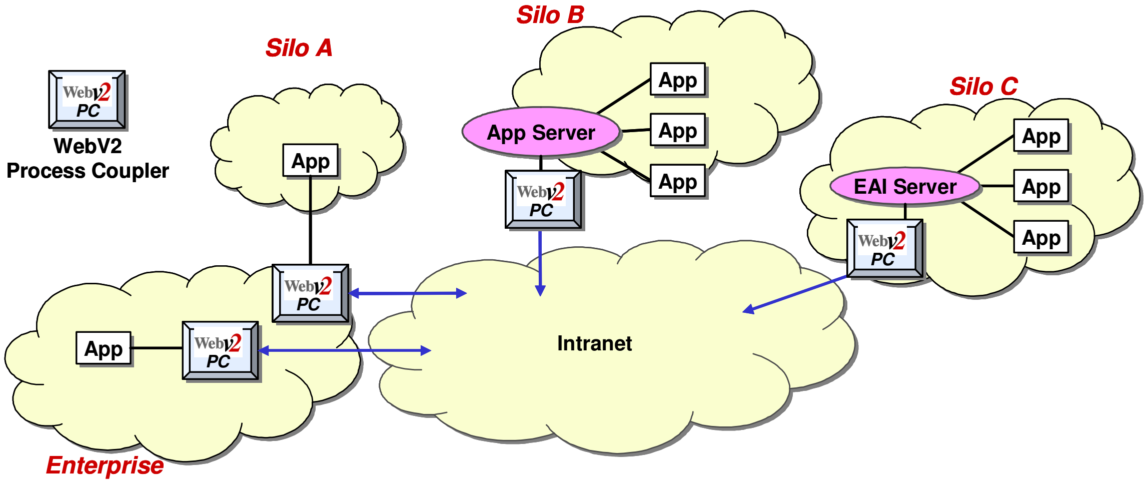


Figure 7: Process Coupler Deployment within the Enterprise

Figure 8 shows deployment across enterprises. It is not necessary to physically deploy Process Couplers at all participating enterprises – as long as the connection to the respective applications is available, Process Couplers may be hosted by an enterprise or by a third party.

Process Couplers run on mobile devices. Thus, users can be offered either a normal web-based interface for participation in the business process, or can participate directly through the PC or wireless device. Process Couplers support the asynchronous participation in processes, required by occasionally connected computing and mobility.

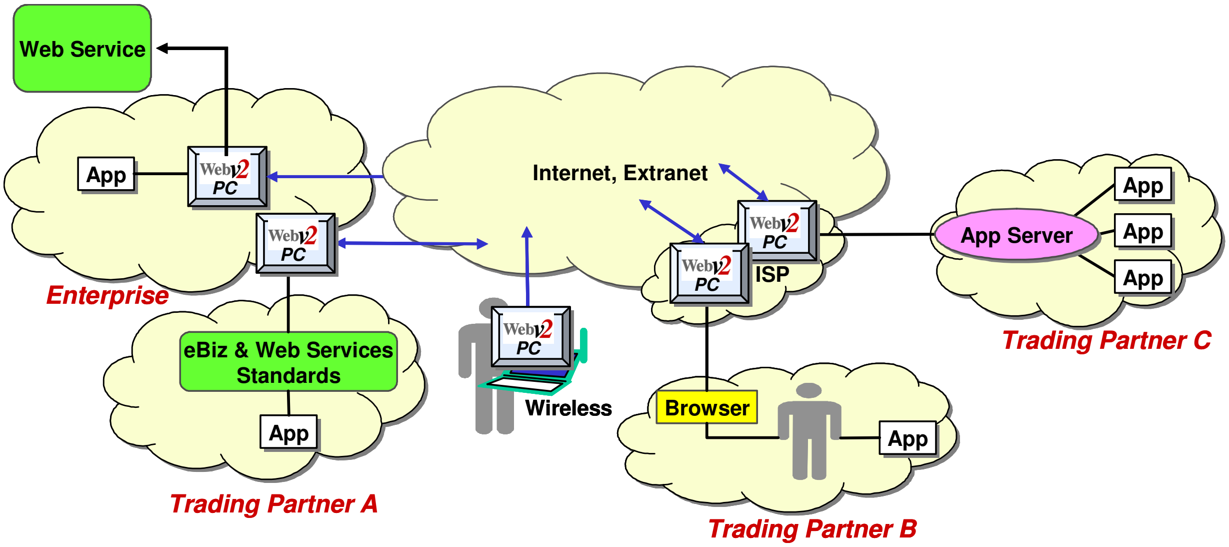


Figure 8: Process Coupler Deployment across Enterprises

8. Summary

Superior Return on Investment and lower TCO throughout the entire process lifecyle

WebV2 solves the problem of inflexible business process management through a loosely coupled architecture. WebV2 helps enterprises achieve greater business agility through more agile business process solutions. WebV2 provide significant benefit in:

- significantly reducing the cost of deploying and scaling processes across the extended enterprise,
- incremental deployment,
- dynamically connecting users, legacy apps, Web Services,
- dynamically changing business processes in running systems,
- enabling intelligent mobile user business process participation,
- transitioning to a services-oriented architecture and BPEL4WS, and
- easily integrating best-of-breed supporting applications.