



IBM Software Group

# IBM WebSphere Partner Gateway V6.1 Advanced and Enterprise Editions

## *Architecture overview*



@business on demand.

© 2007 IBM Corporation  
Converted to video July 16, 2014

This presentation covers the architecture and components of WebSphere® Partner Gateway V6.1 and some of the differences between V6.0 and V6.1.

## Agenda

- High level overview of architecture and components
- Basic document flow
- Incoming and outgoing EDI document flow
- Extensibility
- Architecture changes from V6.0 and V6.1
- Summary



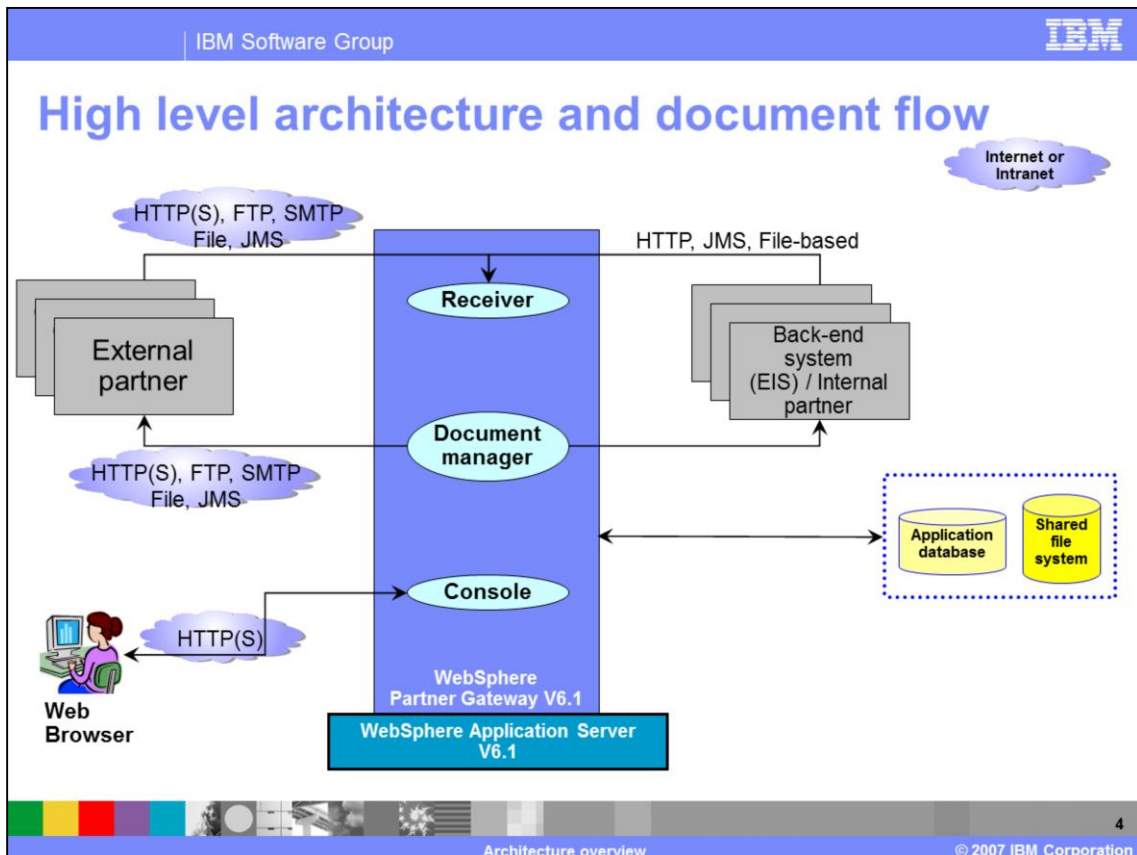
The agenda includes a high level overview of architecture and components, followed by some basic flows and then some Electronic Data Interchange or EDI flows. The extensibility of the product is also discussed, along with the architecture differences between V6.0 and V6.1.

## Section

# *Architecture*



The first section provides details on the architecture and components of the hub.



The slide shows the high level architecture and high-level view of the document flow through the WebSphere Partner Gateway hub components. This architecture is similar to V6.0.

There are three major WebSphere Partner Gateway components, namely the Receiver, Document Manager and Console.

Receiver is the front end to WebSphere Partner Gateway. It accepts and stores documents from partners or back end system for further processing by WebSphere Partner Gateway.

Document Manager retrieves stored data, processes and routes it to both partners and enterprise applications. Additionally, this component performs packaging, validation, logging and other document related functions.

Console provides a view of all business-to-business interactions, allows creation and maintenance of various partner data, profiles, certificates and provides a user interface for the hub administration

WebSphere Partner Gateway requires a shared File system such as Network Area Storage (NAS) used for common storage of documents.

A Database is used to store hub configuration data, meta data, document status, activity logs, temporary store and other information. The Console and the Document manager interacts with the database to save or retrieve information.

Documents can be sent by the external trading partners or internal trading partners (some back end system). Once inside the hub, the document is processed based on the configuration of the hub for that type of document. The processed documents are then sent to the trading partners or to the back end systems.

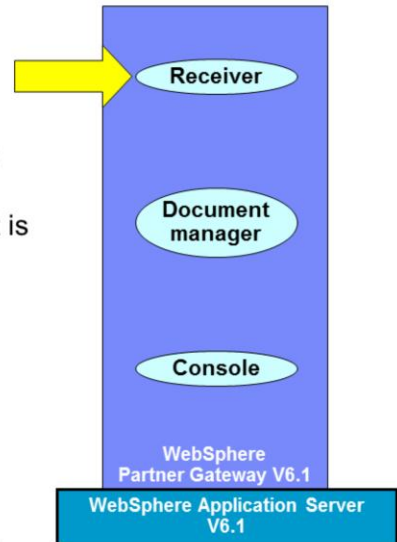
WebSphere Partner Gateway is a loosely coupled component architecture which enables very high scalability and high availability topologies.

WebSphere Partner Gateway can be used alone to provide business-to-business connectivity to a partner. You might also choose to deploy it with other WebSphere Business Integration offerings to provide tighter integration with your enterprise applications. It supports JMS connectivity, HTTP/S, FTP, FTP Scripting and File-based between the gateway and its external or internal partners

Not shown on this page, WebSphere Partner Gateway V6.1 also provides a graphical tool called the Data Interchange Service (DIS) client. It allows you to create the data dictionary, validation maps, transformation maps for the EDI, ROD and XML documents.

## Receiver

- Accepts documents from community participants and from back-end systems and stores them
  - ▶ Receives a document over a supported transport protocol
  - ▶ Determines if a synchronous response document is required for keeping the connection open
- Writes the document and metadata relating to the document to shared file services
- Records any transport-specific data (like the source IP address and certificate information about the SSL connection)
- Completes any transport-specific technical acknowledgment (like sending a 200 response to an HTTP POST)



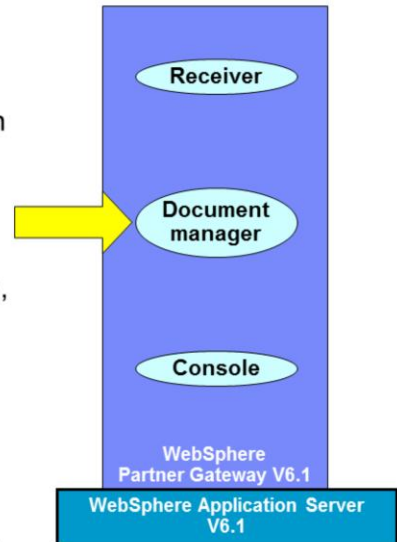
A high level view of the functions of the Receiver is shown here. As the name suggests, the job of the receiver is to receive documents and prepare them for the next stage of the document manager. It receives documents from trading partners or back end systems over many supported protocols and packages.

The received document and meta data are placed in the common shared file system and the document manager is notified of the new arrival. The receiver handles any transport level functions.

## Document manager

Retrieves stored data, processes and routes the data, both to external partners and to enterprise systems

- Saves the inbound document to the non-repudiation store and message store
- Un-packages the data and performs authentication and authorization
- Processes the data to the destination format performing tasks such as validation, transformation, and packaging
- Saves the destination document to the non-repudiation store and message store
- Delivers the document to the intended target destination using JMS queue, file directory, HTTP, FTP or other supported mechanism
- Additional tasks include protocol acknowledgments and retries, event processing, alert processing, and resends



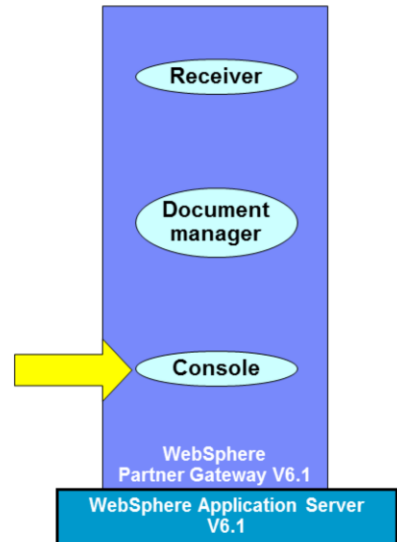
The document manager is the brain of the hub. This is where document processing occurs. When notified of a new document by the receiver, the document manager retrieves the document from the common storage, processes it and routes it to the intended target. The administrator configures the document manager to perform the appropriate actions, such as validation, transformation, or just pass through, on the document.

Protocol specific actions for received documents such as un-packaging including decryption, signature verification, and acknowledgement are performed.

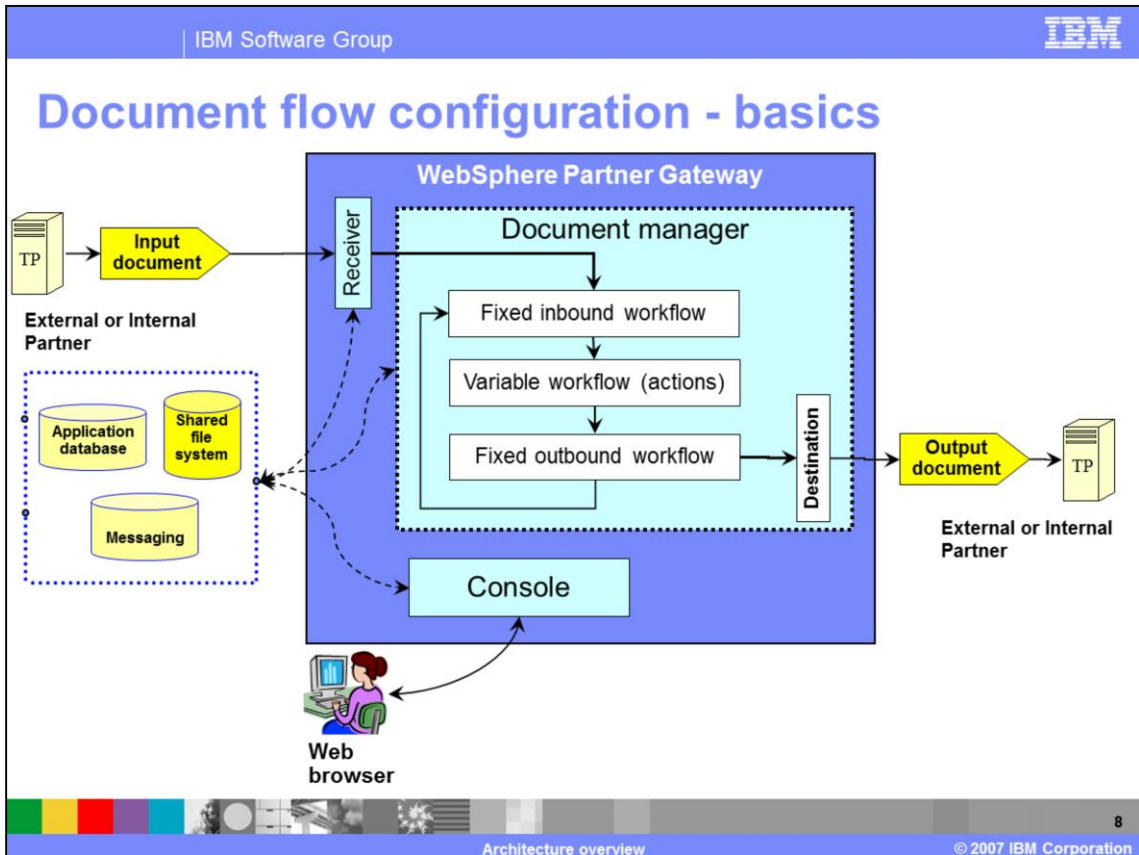
Protocol specific actions for sending documents such as packaging including encryption, signatures, and retries are also performed.

## Console

- The community console is used for configuring, administering, and monitoring trading community activities, and responding to events
- The users are primarily the hub administrator and the internal and external partners
- Console provides role-based access control to the various features and views
- Features of the console include:
  - ▶ Screens for configuring the hub community and for managing partners.
  - ▶ Tools for monitoring business-process events and exceptions.
  - ▶ Detailed reports and analysis on business process, trend, and exception activity
  - ▶ Tools to troubleshoot document processing
  - ▶ The ability to drill down to events and raw documents



The Console is the user interface for the Community Operator, Internal Partner, or the External Partner to configure. It is also used to administer and monitor the trading activities. Based on the type of user logged in to the console, appropriate functions are exposed based on the user's role. The console also provides an interface to a set of tools that allow you to view the documents that have flowed through the hub and provides troubleshooting for failed documents.



This graphic provides more detail about the basic flow within WebSphere Partner Gateway components, especially the Document Manager.

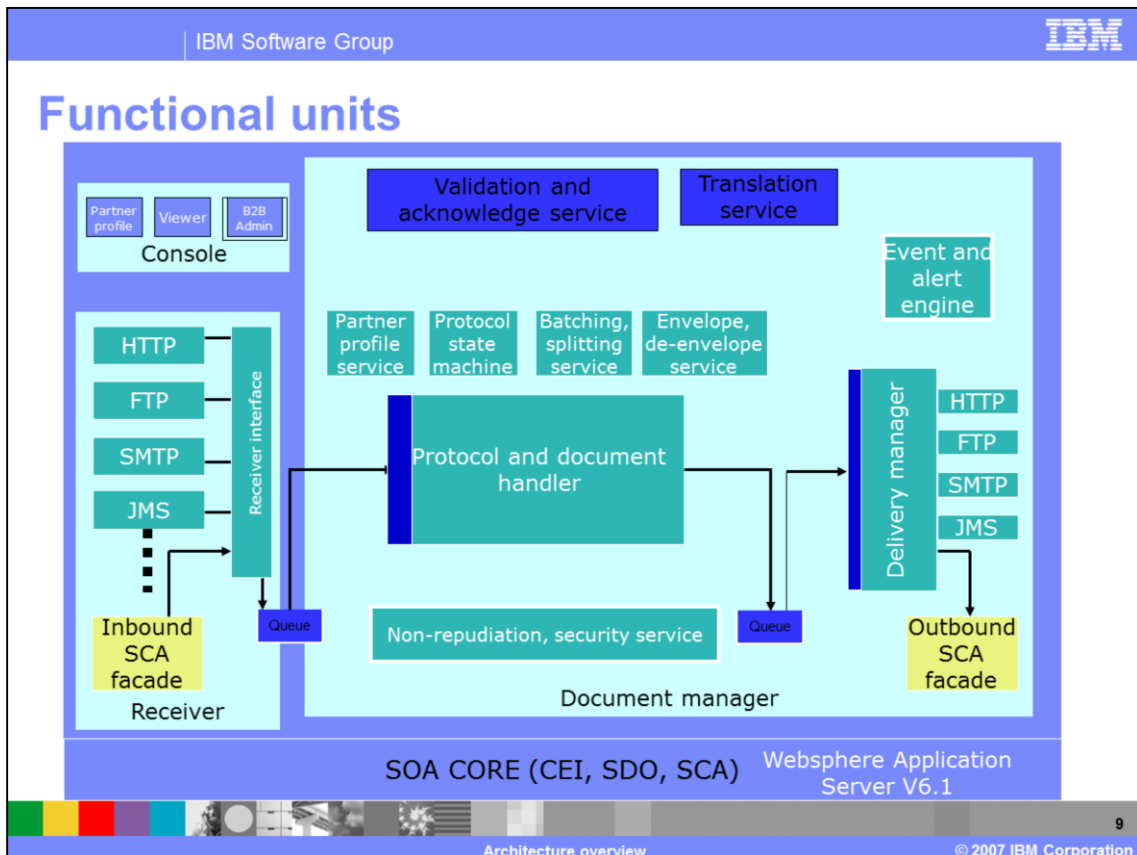
There are three different workflows for every document passing through the document manager. First, the fixed inbound workflow is used for a fixed set of work actions that apply to all incoming documents. For example, unpackaging the incoming document like an AS2 packaged document.

Second, the variable workflow is, as the term implies, variable and is based on the actions specified by the administrator on the specific incoming document. Examples are Validation and XML Translation that apply to EDI and XML document flows.

Third, the Fixed outbound workflow is used for a fixed set of work actions that apply to all outgoing documents. For example, packaging the outgoing document to an AS2 packaged document.

In some document flows, especially involving EDI documents, intermediate documents are created by the hub and then flowed back through the document manager for further processing. Each flow of any document through the hub represents a connection between the source and the target trading partners, including the intermediate ones.





This diagram shows the functional units of WebSphere Partner Gateway. The Console provides the user interface for partner profile management, administration of business-to-business capabilities of the hub and partners, different viewers like the event viewer, document viewer and so on.

The Receiver on the left side of the picture is the gateway into the hub. It supports many different protocols like HTTP, FTP, SMTP, JMS, File based and so on. In WebSphere Partner Gateway V6.0.2, there was support for inbound service component architecture, or SCA façade to allow integration from WebSphere Process Server to WebSphere Partner Gateway.

The Document Manager is where the documents are processed. It contains all the logic to process, validate, transform and send the documents to the destination. It contains the functions for EDI enveloping and de-enveloping. It performs the non-repudiation and security services. It is the one that contains the event and alert engine that sends out user control alerts for specific events. The Delivery manager of the Document manager sends the output document to the target destination through the supported transport protocols such as HTTP/s, FTP, SMTP, JMS, File based and so on. Just like the inbound SCA façade, there is an outbound SCA façade that allows WebSphere Partner Gateway to send documents to WebSphere Process Server.

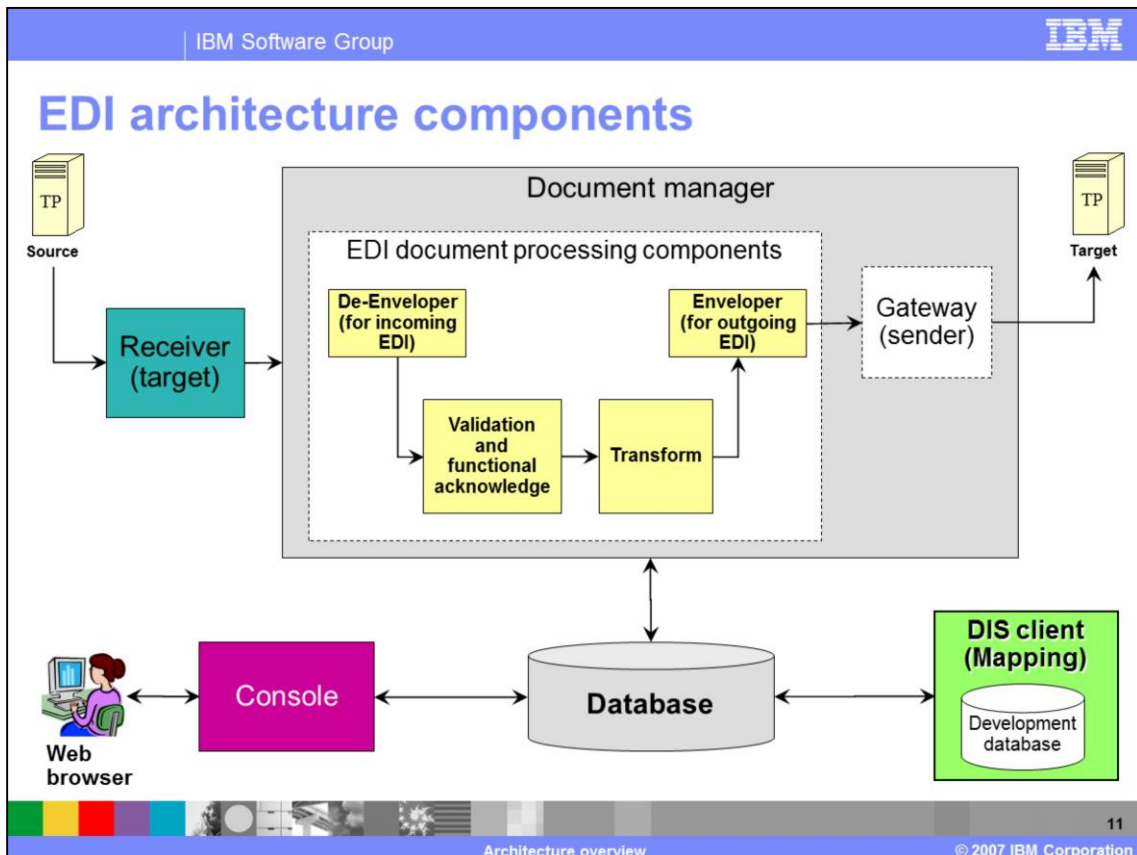
The underlying WebSphere Application Server V6.1 provides the SOA core functionality like the Service Data Object (SDO), Common Event Infrastructure (CEI) and SCA.

Different presentations touch on the various functional units described in this page.

## Section

# ***Components and architecture to support native EDI document processing***

This section will provide a high level view of the architecture and components to support native EDI document processing.



The main components involved in native processing of EDI documents within the document manager are shown here in relation to the rest of the WebSphere Partner Gateway architecture and components. The main EDI related components De-Envelope, Validators, Translator and Envelope

The De-Envelope for incoming EDI Interchange to de-envelope EDI documents and extract the EDI transactions contained in them.

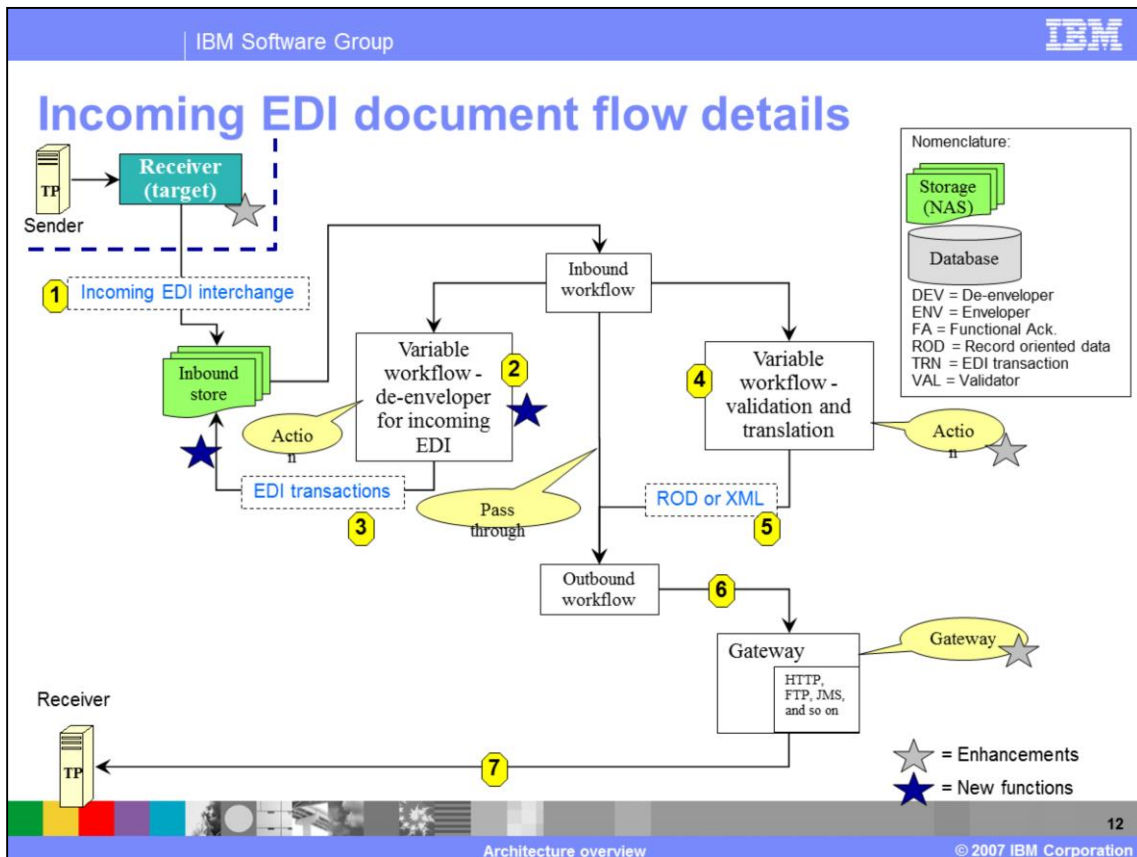
The Validators to validate the EDI documents and transactions as they enter the hub or as they are generated from the maps.

The Translator for translation to and from EDI, based on the maps.

The Envelope for EDI output used to envelope any generated EDI transactions into an EDI interchange before sending to the target trading partner.

The DIS client tool is used to create the necessary maps to and from EDI transaction. The maps are imported in the WebSphere Partner Gateway database directly from the DIS client or using the command line tool called bcgDISImport.

Other presentations cover the details of these components and different flow use cases involving EDI documents.



New capability in WebSphere Partner Gateway provides native processing of EDI documents for validation, transformation and other EDI specific functions. This graphic shows the details of an incoming EDI that can be converted to ROD or XML. For the Incoming EDI document, the typical steps are as follows:

In step 1, the receiver receives the incoming EDI document and places it in the common storage for the document manager

In step 2, the EDI interchange is de-enveloped and EDI transactions extracted.

In step 3, the EDI transactions are fed back through the document manager.

In step 4, the EDI transactions are validated and transformed to the required output format, which can be XML, ROD or EDI.

In step 5, the generated ROD or XML is shown.

In step 6, 7, and 8, if the transformed data is ROD or XML, they are sent to the target trading partner gateway and then to the target trading partner.

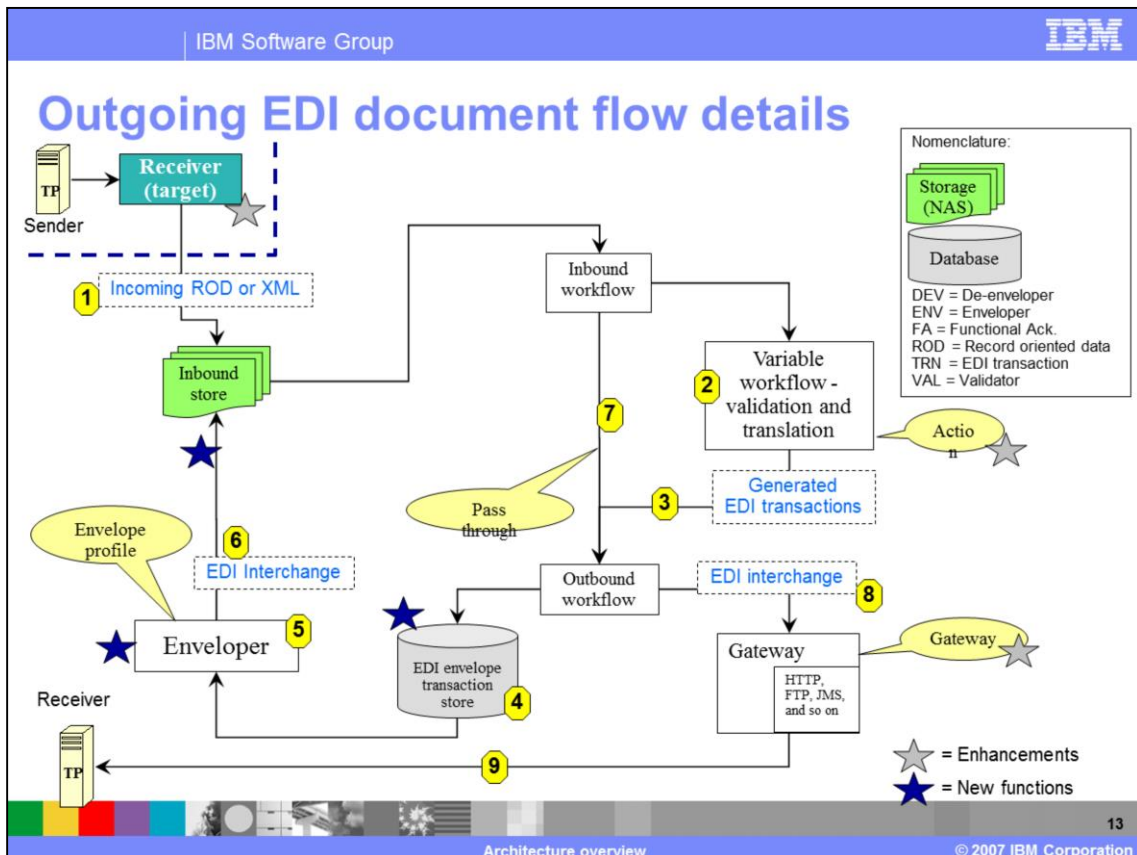
If the EDI acknowledgements (TA1 and Functional) need to be sent, they are sent by the document manager. The flow here does not cover the EDI acknowledgements.

In the Fixed Inbound Workflow, the Protocol Process step EDI Handler will be used to identify an EDI Interchange.

The EDI Handler is modified for handling Interchanges and Transactions. Interchanges will either be coming from a partner or internally as output from EDI enveloper. Transactions will be coming from Interchanges that have been de-enveloped. Some of these transactions may be Functional Acknowledgments (FA) that require additional special handling.

The Variable work flow is also referred to as the action. This is where the administrator specified the action to be performed on the document, like De-enveloping, validation, or Transformation in the case of an EDI document. The De-envelope action splits the EDI Interchange into its

transactions. The Transformation action then transforms the EDI transactions into the appropriate output.



This graphic shows the details of an incoming ROD or XML document converted to an EDI interchange. The steps of generating an EDI document and sending it to the target trading partner are as follows:

- In step 1 - The receiver receives the incoming ROD or XML document and places it in the common storage for the document manager
- In step 2 – The EDI interchange is de-enveloped and the EDI transactions extracted. ROD or XML documents are transformed into EDI transactions and validated.
- In step 3 - The EDI transactions are fed back through the document manager.
- In step 4 - The EDI transaction must enveloped before being sent to the target trading partner. The enveloper runs from a scheduler. Between scheduled runs of the enveloper, the EDI transactions being generated are stored in the EDI Envelope transaction store, ready to be enveloped.
- In step 5 - At the scheduled time, the enveloper is invoked, enveloping the EDI transaction
- In step 6 - An EDI interchange is generated.
- In step 7 - The EDI interchange is sent back through the document manager for any packaging and other processing needed. The EDI passes through the document manager without any action.
- In step 9 - The output EDI interchange with the appropriate packaging is sent to the gateway. Finally, the EDI interchange is sent to the target trading partner.

## Section

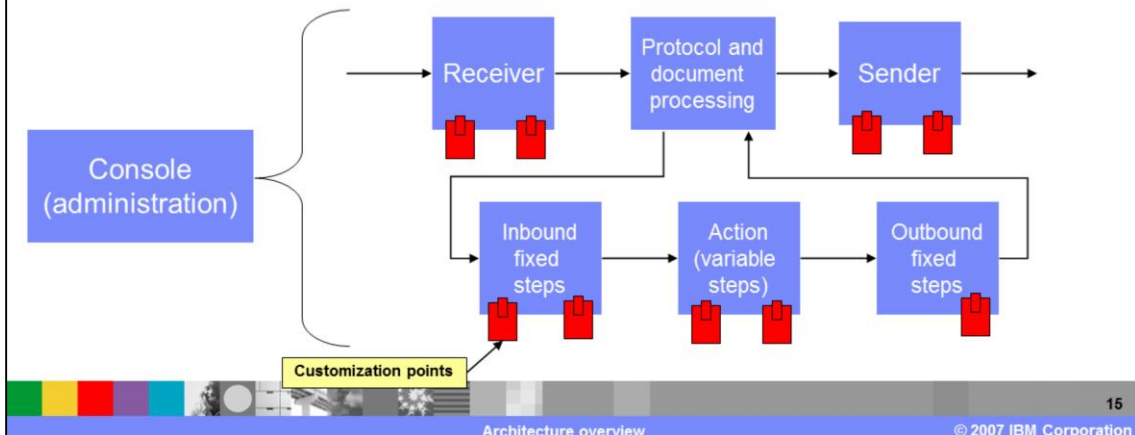
# ***Extensible architecture***



This section shows the extensibility of the architecture.

## Extensible architecture through exits (handler)

- Provide interface to perform custom functions through user 'exits' or "handlers" – multiple customization points
- Custom processing function such as validation and transformation
- Enable addition of custom transports, packages and protocols
- Documented for customers, services and partners



WebSphere Partner Gateway provides a mechanism for adding custom functions called exits or handlers at different stages of document processing. It includes support for many pre-defined packages, protocols and document types. With the pluggable extensibility of the architecture, new custom packages, protocols or document types can be added by the hub administrator.

Listed below are some examples of architecture extension through the use of exits and handlers.

One example is for Encryption, Validation and Transformation,

As WebSphere Partner Gateway receives and processes a document, a 'call out' to any program can be easily made as part of the processing. These User Exits or handlers can be used for example for document encryption and decryption, and validation, or transformation, or both.

Another example is for Custom Transports, Packages and Protocols.

WebSphere Partner Gateway is extensible to enable custom Transports, Packages and Protocols to be added within the framework of WebSphere Partner Gateway runtime and Community Console based administration. There are a number of exit points that enable the addition of custom listeners for additional Transports, parsers for additional Packages and Protocols. In this way the rest of the infrastructure of WebSphere Partner Gateway can be used in conjunction with these new capabilities.

For example, a new Transport could deliver a document, such as EDI X12 that would be processed in the normal way through the rest of the system. Also, AS2 might be used to deliver a new Protocol payload document. So the combination of standard and custom components can be used to extend and reuse all the capabilities of WebSphere Partner Gateway in a seamless manner.



## Section

# ***Architecture changes from V6.0 to V6.1***



Next section discusses the Architecture changes from WebSphere Partner Gateway V6.0 to V6.1

## Deployment changes

- WebSphere Partner Gateway V6.1 components (Console, Receiver and Document manager) are J2EE applications installed in single server or Network Deployment cell of WebSphere Application Server V6.1
  - ▶ Provides several new options of having different topologies
- WebSphere Partner Gateway V6.1 can now exploit the capabilities of WebSphere Application Server Network Deployment, namely the clustering and central point of administration
- Use of embedded messaging instead of WebSphere MQ
  - ▶ Reduces a product dependency.
  - ▶ Distributed has separate messaging cluster, enabling HA.
  - ▶ Requires a separate messaging (MAS) database.
- Other – Help system is now embedded in the hub console



WebSphere Partner Gateway V6.1 components, namely, the Receiver, Console and the Document Manager are J2EE applications running in WebSphere Application Server V6.1 or WebSphere Application Server V6.1 Network Deployment Cell.

These hub Components can now take better advantage of underlying WebSphere Application Server infrastructure, including the clustering capabilities. It also provides you with several new options for your deployment topologies. These topologies are discussed in the Install and Topology presentation.

The other change is the use of embedded messaging within the WebSphere Application Server for internal messaging needed between the hub components. In V6.0, WebSphere MQ provided that messaging function. WebSphere Partner Gateway V6.1 does not require WebSphere MQ for internal messaging. You can still use WebSphere MQ for external connection.

WebSphere Application Server Network Deployment capability of messaging cluster provides high availability for WebSphere Partner Gateway messaging.

Messaging cluster is now used for High Availability. By adding WebSphere Partner Gateway Messaging Application Servers (MAS) additional backup machines are available. Note that only one MAS server is actively processing messages at a time. A separate messaging database is required for persistence and sharing among the multiple hub components. The install

presentation goes into more details of the deployment changes.

## Messaging in WebSphere Application Server - Background

- Service integration bus is the messaging infrastructure within the application server
- Messaging Engines are used to put and get messages from the service integration bus
- Destinations or queues are where messages are put or where messages are obtained
- The physical location of a Destination or queue is associated with specific messaging engine instance
- A messaging engine can be associated with a cluster instead of a specific application server and one of the cluster member runs the messaging engine
- WebSphere Application Server Network Deployment clusters and operating system (OS) HACMP™ clusters are not the same thing



The Platform messaging within WebSphere Application Server is exploited for internal messaging between hub components.

The Service Integration Bus is the infrastructure for the messaging. Messaging engines running within the application server put and get the message on the destinations or queues within the service integration bus.

The messaging engine is can also be associated with a cluster within the Network Deployment cell. One of the messaging engines in the cluster is active at any given time and in case of fail over, another messaging engine on another server of the cluster can take over the messaging function.

Clustering in WebSphere Application Server Network Deployment is different than the operating system HACMP clusters. For OS HACMP, servers are configured independently and then added to a cluster. For WebSphere Application Server cluster nodes are added to a cluster and then configured with the same applications. They are clusters of cloned application servers, which is different than OS clusters.

## Embedded messaging to WebSphere MQ analogy

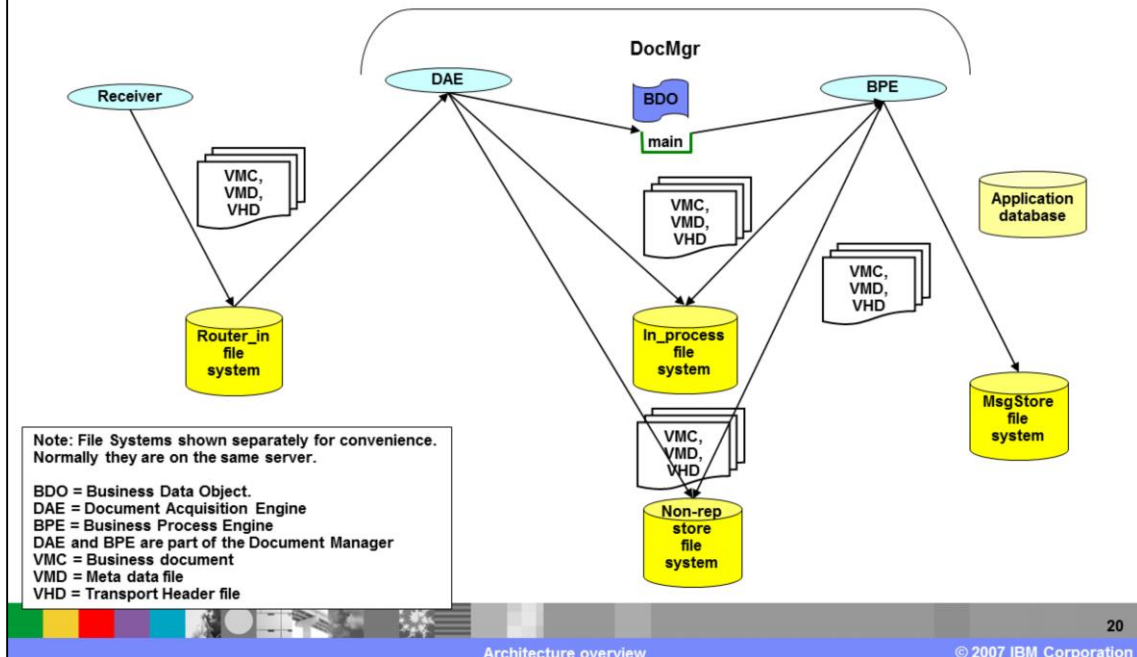
- Messaging engine is similar to a WebSphere MQ QManager
- Destinations or queues are similar to WebSphere MQ Queues
- Service integration bus is similar to the underlying “network” that WebSphere MQ uses for moving messages around to the various queues



Some of the related terminology between the WebSphere MQ and the embedded messaging within WebSphere Application Server is shown here.

The Messaging Engine is similar to WebSphere MQ QManager. The destinations and queues in service integration bus are similar to WebSphere MQ Queues. Finally, the service integration bus is similar to the entire WebSphere MQ infrastructure.

## V6.0 runtime data flow architecture receiver → BPE



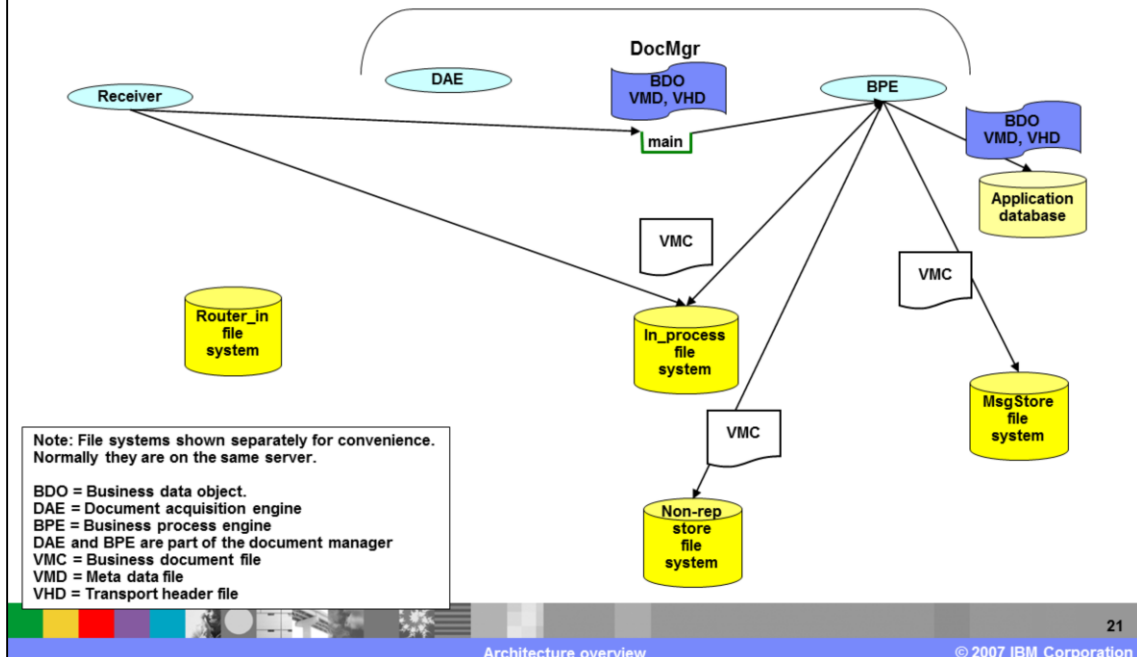
High level characteristics of the hub V6.0 data flow during runtime from the Receiver to the BPE are as shown on this page.

The Receiver creates three files, VMC for the business document, VMD for the Meta data and VHD for the Transport header data and writes them into the “router\_in” directory. The 3 files, VMC, VMD and VHD are file extensions.

The DAE retrieves the three files, Non-repudiates the Source, writes three files to the “in\_process” directory and creates BDO into the main queue.

The BPE retrieves BDO from queue, Non-repudiates the Source certificates, writes three files to message store, processes three files, Non-repudiates the Target.

## V6.1 runtime data flow architecture receiver → BPE



The high level characteristics of the WebSphere Partner Gateway V6.1 data flow during runtime from the Receiver to the BPE are as shown on this page.

The Receiver creates one file (VMC – business document) in the “in\_process” directory and creates BDO (also contains VMD and VHD data) in the main queue. The BOD contains the VMD and VHD data as opposed to putting them in the files in V6.0.

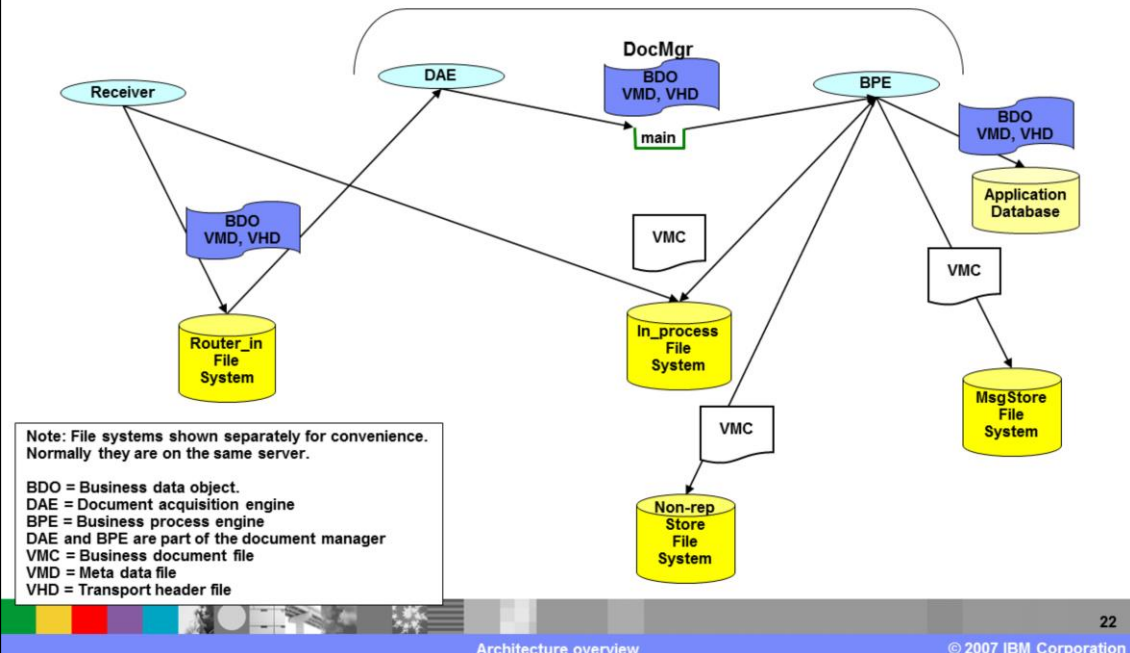
DAE is not used in this process. BPE retrieves BDO from queue, Non-repudiates Source document and certificates, writes VMC files to message store, processes one file, Non-repudiates Target, writes BDO to database.

The main differences between V6.0 and V6.1 is that in V6.1 DAE is now bypassed by the Receiver. In V6.1 there is a single file, namely the VMC file for the business data. Lastly, in V6.1, the BDO contains the VMD and VHD data, and is stored into the database instead of the VMD and VHD files.



## V6.1: runtime data flow architecture – Alternative receiver → BPE

Used for Receiver to queue problems  
and backward compatibility

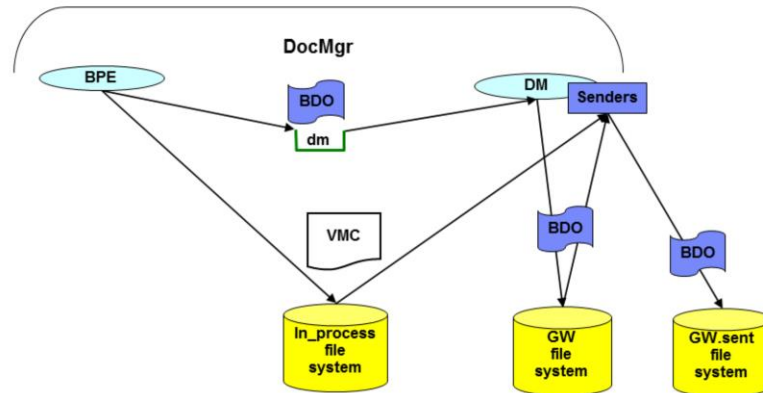


If either the main queue cannot be written to by the Receiver or the `bcg.global.common.introduce.document.transport` attribute is set to `FileSystem`, V6.1 behavior goes back to V6.0 kind of behavior. In this case, the Receiver still writes a single VMC file to the `in_process` directory, still creates the BDO but puts the BDO into the `router_in` directory. The DAE retrieves the BDO and writes to the main queue. The BPE processing remains the same.

For backward compatibility, you can set the attribute as specified.



## V6.0 runtime data flow architecture BPE → DM



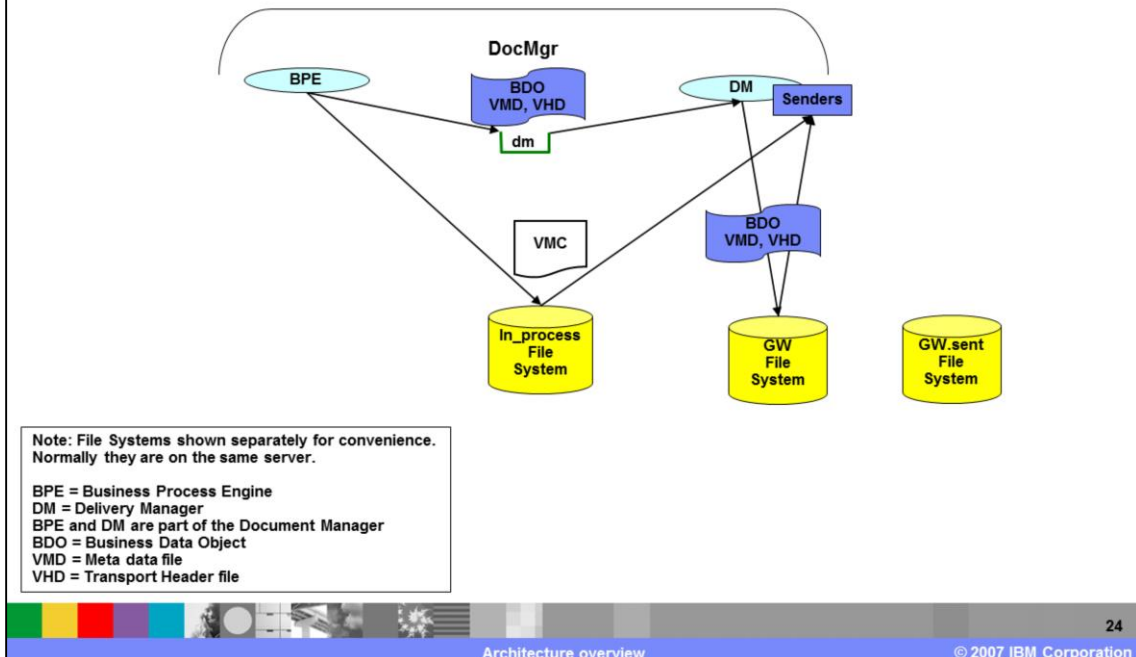
Note: File systems shown separately for convenience. Normally they are on the same server.

BPE = Business process engine  
 DM = Delivery manager  
 BPE and DM are part of the document manager  
 BDO = Business data object  
 VMD = Meta data file  
 VHD = Transport header file

High level characteristics of WebSphere Partner GatewayV6.0 data flow during runtime from the BPE to Delivery Manager (DM) are shown on this page.

The BPE writes the BDO to the Delivery Manager queue. DM retrieves the BDO from the queue and writes the BDO to the GW folder. DM Senders read the BDO from the Gateway, GW, folder, retrieve the business document, sends the business document, writes the BDO to the GW.sent folder.

## V6.1 runtime data flow architecture BPE → DM



High level characteristics of WebSphere Partner GatewayV6.1 data flow during runtime from the BPE to DM are shown on this page.

The BPE writes the BDO to the Delivery Manager queue. DM retrieves the BDO from the queue and writes the BDO to the Gateway, GW, folder. DM Senders read the BDO from the GW folder, retrieve the business document and sends the business document.

The main difference is that the GW.sent folder is no longer used.

# Section

## ***Summary***

This section provides a summary of this presentation.

## Summary

- Architecture and components of WebSphere Partner Gateway
- Main components of the product and their roles in the document flow
- Architecture extension points support functions and protocols that may be needed by the users of hub
- V6.1 architecture changes include:
  - ▶ The Receiver, Document Manager and Console are J2EE applications
  - ▶ Exploitation of WebSphere Application Server Network Deployment
  - ▶ New deployment and installation options
    - These can be installed in single server/single machine
    - Multiple instances in single server/multiple machine
    - Multiple server/multiple machine to provide deployment flexibility
  - ▶ Data flow changes
    - Receiver performs general role of the DAE
    - VMD and VHD files eliminated
    - GW.sent folder no longer used

In summary, this presentation covered the basic architecture and the components of WebSphere Partner Gateway V6.1. It also included the architecture changes from V6.0 to V6.1.

## Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

HACMP IBM WebSphere

J2EE and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2007. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.