

Informatica Cloud Application Integration: A Capabilities Overview

The shift to the cloud requires a combined
data and application integration strategy

ABOUT INFORMATICA

Digital transformation changes our expectations: better service, faster delivery, greater convenience, with less cost. Businesses must transform to stay relevant. The good news? Data holds the answers.

As the world's leader in enterprise cloud data management, we're prepared to help you intelligently lead—in any sector, category or niche. To provide you with the foresight to become more agile, realize new growth opportunities or even invent new things. With 100% focus on everything data, we offer the versatility you need to succeed.

We invite you to explore all that Informatica has to offer—and unleash the power of data to drive your next intelligent disruption. Not just once, but again and again.

Table of Contents

Overview	4
Informatica Intelligent Cloud Services	5
Integration Cloud—A Differentiated Integration iPaaS	6
Informatica Cloud Application Integration (CAI)	7
ESBs and Informatica Cloud Application Integration	16
A Cloud Application Integration Example	18
Conclusion	23

Overview

The shift to the cloud compels architects to develop a combined data and application integration strategy that considers how on-premises and cloud application and data services co-exist and integrate to fulfill the role they were deployed for.

A comprehensive integration strategy must consider various co-existence and integration aspects:

- Data consumption: How applications consume data whether locally and without latency, on-demand and interactively from an external data provider or service, via a data hub, or through streaming listeners
- Data and application services: How data is exposed as a service
- Data propagation: How data propagates—for example, via data set synchronization, replication, store and forward or publish/subscribe, streaming, messaging, or event-based propagation via service-oriented API requests

An application integration strategy needs to also consider how business and data service APIs are provided and consumed as the means used to:

- Propagate business events triggered in one application to others—for instance, order fulfillment
- Give users the ability to interactively access application data and business services residing in the cloud and/or on-premises without having to replicate data
- Consume application and data services using synchronous and asynchronous means of interactions
- Integrate business processes across a set of loosely coupled applications—for example, order to cash
- Integrate partners

Informatica Intelligent Cloud Services (IICS) offers the means with its integration platform as a service (iPaaS) to integrate and offer data and application services deployed on-premises and in the cloud.

Informatica Intelligent Cloud Services

Integration Cloud, a component of Informatica Intelligent Cloud Services (IICS), is offered as an iPaaS that provides near universal access to application data regardless of its location, format, or origin and integrates applications and application processes regardless of where they are deployed. Integration Cloud provides the means to integrate and deliver:

- The right data, of the highest quality, at the right time
- Data to the right place, whether on-premises or in the cloud
- Data to the right consumer, whether it is a business user or an application
- Data in the right way, ensuring it is secure and protected

Integration Cloud provides the ability to move and migrate existing enterprise business applications to public and private cloud solutions as well as allowing for continued co-existence with on-premises applications and systems. It supports on-going co-existence integration needs as businesses shift some or all applications to cloud solutions over time.

Integration Cloud, which can be adopted in a modular fashion or implemented in whole based on need, helps customers manage:

- Data distribution that ensures it is available locally to the application that consumes it
- Data propagation that moves and processes data feeds as data sets or events
- Data services that expose data as a service
- Event discovery that gleans events from data sources
- Event processing that reacts to events as they are discovered or take place
- Data and business services that provide, consume, and orchestrate data as it integrates applications and systems in real time using service-based API interaction
- Process integration and management that executes within a diverse environment and integrates loosely coupled application and business processes

Integration Cloud lets you address your application and data integration needs using a variety of integration patterns:

- API creation and consumption
- Service orchestration (request/response or straight through processing)
- Process automation and integration (including long-running business processes requiring asynchronous responses)
- Message-based integration (publish/subscribe)
- Business-to-business integration
- Data synchronization and replication
- Managed file transfer
- Bulk and batch data integration and transformations of data sets
- Handling of structured and unstructured data

Integration Cloud is a comprehensive iPaaS that lets lines of businesses address their cloud and data and application integration needs. Informatica originally targeted integration for applications and has gradually enhanced its platform to where it is the most complete and comprehensive offering available today.

Integration Cloud—A Differentiated Integration iPaaS

Integration Cloud provides the means for your cloud and on-premises applications to coexist. This iPaaS enables access to data wherever it resides— ‘in the cloud and on-premises—delivering trustworthy data while meeting your company’s security and compliance standards.

Integration Cloud shares the same foundation as Informatica’s on-premises products, providing unparalleled advantages over competing solutions. It differentiates itself through a large set of capabilities including:

- Comprehensive support for cloud-to-cloud, cloud-to-on-premises, and on-premises-to-on-premises integration for data, service, and process integration scenarios and patterns
- Flexibility to choose any environment and move workloads from on-premises to cloud and vice versa, depending on application, processing, or other characteristics
- Shared metadata and definitions, as well as interoperable and reusable integrations across cloud and on-premises
- Flexibility of design environment so data and application integration designers can leverage the cloud or on-premises tool of their choice
- Self-service consumption by lines of businesses and departments, while still allowing centralized governance by integration competency centers
- Data management services, including data replication, data quality, master data management, address validation, data masking, and test data management
- Secure agent technology with auto-updates for secure access to on-premises applications and middleware platforms for cloud-to-on-premises integrations
- SDK and APIs to embed and extend the platform
- Broad, secure, and universal connectivity (on-premises and cloud), including SaaS, on-premises systems and database, message formats, B2B libraries, big data, social networks, unstructured data, devices, and more

Informatica Cloud Application Integration (CAI)

Informatica's event-driven and service-oriented application integration capabilities encompass event processing, service orchestration, and process management. These are built on Informatica's business process management technology. Its use within Integration Cloud, embedded within the Cloud Secure Agent, makes it possible to create and consume APIs, orchestrate data services and business services, integrate processes, and offer data and applications services inside and outside an organization.

Informatica's cloud application integration capabilities are ideally suited for service-oriented integration when you need:

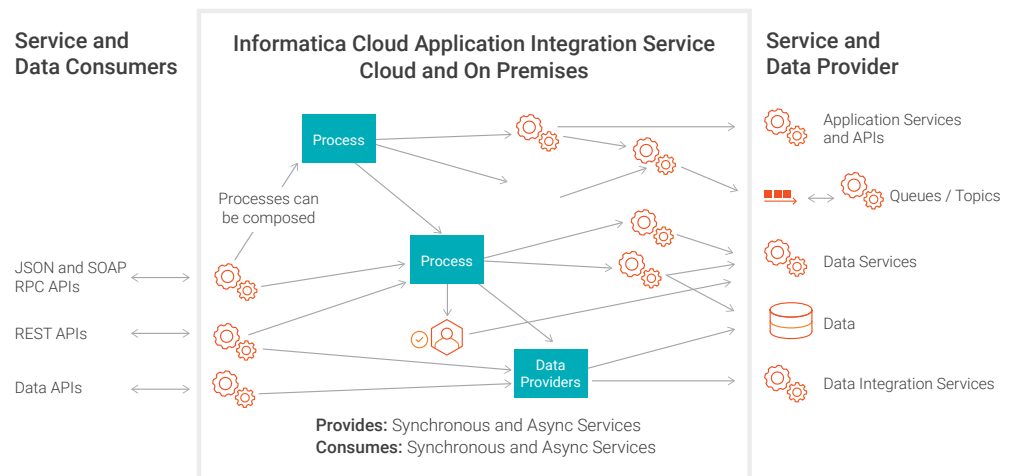
- Long-running transactions that maintain state
- Short-running or transactional system integration processes requiring integration sequences, different execution paths, or composite transactions
- Rich semantics for parallel execution
- Timers and event triggers
- Rich event, fault, and error-handling systems that control how and what to compensate through automated compensation to roll back a transaction if all required steps are not completed successfully

Transaction orchestration that spawns across different companies, business units, products, or services to realize horizontal business integration processes, such as an order-to-cash process. Visibility during execution into what is or isn't happening and which processes are in progress in order to manage escalations, timeouts, and schedules.

Other capabilities include:

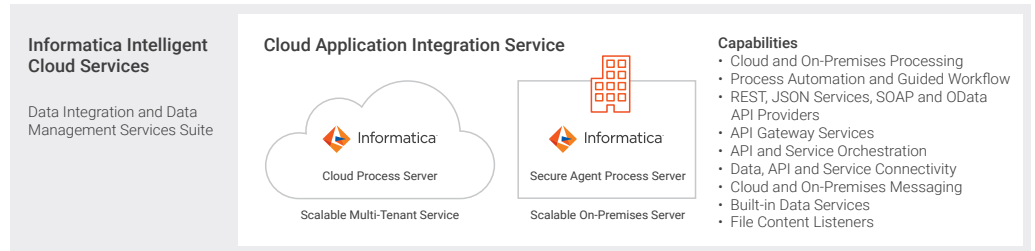
- Screenflow for user task automation, workflow, and interactive access to data
- Content-based routing, transformations to or from XML and non-XML types, and decryption/encryption, signature validation, authorization, and more.

The platform's architecture makes it ideally suited for event-driven integration such as that depicted here.



Components

Informatica Cloud Application Integration (CAI) lets customers expose business services at cloud or on-premises service endpoints accessible via REST (XML/JSON – the server receives either formats, and the content-type HTTP header is used to control what the server responds with or sends – J JSON/RPC, and SOAP, and as message-oriented services and consumers. This section describes the components of CAI's service-oriented architecture, including the Cloud Process Server, the Cloud Secure Agent's embedded Process Server, and the technologies and capabilities of the platform.



- Integrate your apps, your data, and your processes in real time
- Expose your APIs to your applications and partners
- Automate your business processes and workflows



Process Server

Process Server is a run-time and process management engine that scales to meet the needs of the cloud and enterprises of any size. Execution is carried out by Process Server. Process Server provides several sophisticated features to ensure business continuity and can be deployed as a cluster in failover mode to ensure high availability.

When deployed within Cloud Application Integration, Process Server is used to securely partition users into discrete tenants, or IICS organizations. With this multitenant architecture, each IICS organization (or tenant) shares hardware and software resources but has its own private and secure access to CAI's Process Server.

Process Server has been built to support nonstop operation of composite business applications.

You can:

- Configure and enforce runtime behavior of an orchestration using standard policies
- Perform server-based runtime message correlation
- Perform automated service invocation retry if a service is temporarily unavailable
- Offer endpoint management capabilities to easily deploy an orchestration in one environment or another or deal with a change in topology
- Suspend a running process to handle bad data that would otherwise have unnecessarily failed a transaction and then correct the problem

Process Console carries out these functions and configures Process Server.

Process Console

Process Console provides a central location from which to manage and configure Process Server instances and its deployed resources whether in the cloud or embedded within Secure Agent. Process Console provides a means to schedule processes and deploy new or updated processes.

Process Console lets tenants perform root cause analysis if a process exception occurs and then take corrective actions. Process rewind—a process exception management feature—offers the ability to visually rewind a process to a specific activity and redo the work without having to invoke any of the built-in compensation logic, giving organizations unprecedented flexibility in managing and running in-flight processes.

Process Designer

Cloud users demand an easy-to-use web interface for creating their integrations and automating processes. Process Designer provides unparalleled ease of use for citizen developers to create and deploy processes to the cloud's and Secure Agent's Process Server. Process Designer is intended to be used by a technical power user—an automation designer—who may or may not be a developer, but knows the business processes and services used to accomplish them. This designer is designed to be easy to use yet powerful and expressive to create any business process.

A key guiding principle behind Process Designer is ease of use. This is exemplified by features that free the user from the tedium of having to lay out process activities by hand. Instead, steps are automatically linked for the user. Users can select step types such as Decisions, Services, Parallel Paths, and iteration constructs to accomplish their process.

If a user, for example, creates a Decision step with multiple possibilities, branches will then automatically be created for those possibilities. The same is true of Parallel Path steps, where parallel branches grow on the canvas that correlate with the desired parallel activities to be performed. When done, the user simply saves and publishes the process definition, and the service is automatically created, deployed, and ready to be invoked as a REST (XML/JSON), a JSON/RPC, and a SOAP service. No other vendor has this type of capability or can claim this level of ease of use.

Creating a Service Definition to invoke from a process is as simple as using a form to specify input/output parameters, endpoint information, and test connection information and then saving and publishing the service connection. Once saved, the service definition is automatically incorporated as part of the services used in the process and others that want to consume this definition. Swagger, WSDL/XML Schema and OData introspection documents are automatically created for users.

To satisfy data integration orchestration needs, a specialized version of Process Designer is offered that provides the means to orchestrate data synchronization, mapping configuration templates, and others. Customers benefit from the ability not only to serialize and handle errors in a resilient manner but also to process data ingestion, for example, in parallel or conditionally.

Process Developer

Development teams must often work on multiple projects including Java, service-based development, and orchestration. They shouldn't have to necessarily adopt a new development tool each time they switch between projects. For this purpose, Informatica also offers Process Developer, a rich Eclipse-based IDE intended for developers, that incorporates the BPMN, BPEL, and BPEL Extensions for People (BPEL4People) standards. Its optimized and easy-to-use features make it easy for developers to create business process applications quickly. And because these applications are based on industry standards, enterprises' business logic is freed from proprietary orchestration engines.

Process Developer can:

- Make it easy for architects and developers to work collaboratively with business analysts by offering as a notation BPMN for modeling and implementing business processes. Process Designer makes use of BPMN notation as well
- Expose the full power of BPMN, enabling designers to control every aspect of the diagram. Process Developer promotes modeling best practices while being significantly easier to use. Structured activities can be dragged and dropped from a palette onto the canvas, significantly reducing the amount of time required to model a BPEL process
- Allow users to perform service discovery and provides the ability to manage service references to help users deal with changes of service definitions
- Orchestrate services defined using Web Services Definition Language (WSDL) interfaces or allowing designers to start with XML schema or XML fragments if this is all that is available
- Incorporate non-web service-based assets through a WSDL interface facade, enabling designers to leverage existing JMS, REST (XML/JSON), JSON/RPC, and Java-based assets. As such, these are used as if they were services, each having a distinctive binding
- Simulate local processes or remote debugging, allowing designers to save simulations and test data, which can then be used to generate unit tests and test suites to perform scenario testing
- Use wizard-based deployment to execute new orchestrations and updates to Process Server or Secure Agent's Embedded Process Server

Secure Agent

Cloud Secure Agent is a key component of Informatica's secure solution. Secure Agent can be installed on-premises or in the cloud depending on connectivity needs. It acts as a container for various services such as the Channel Service that manages communication to and from the cloud service, the Data Integration Service that processes data sets using mapping and data synchronization tasks, and the Process Server Service that processes execution and event-processing on-premises.

Communication between the Secure Agent and IICS is carried out through a Secure Channel that the agent initiates. This is depicted here as an example of how the Secure Agent facilitates data integration among a local database and Salesforce CRM and Force.com.

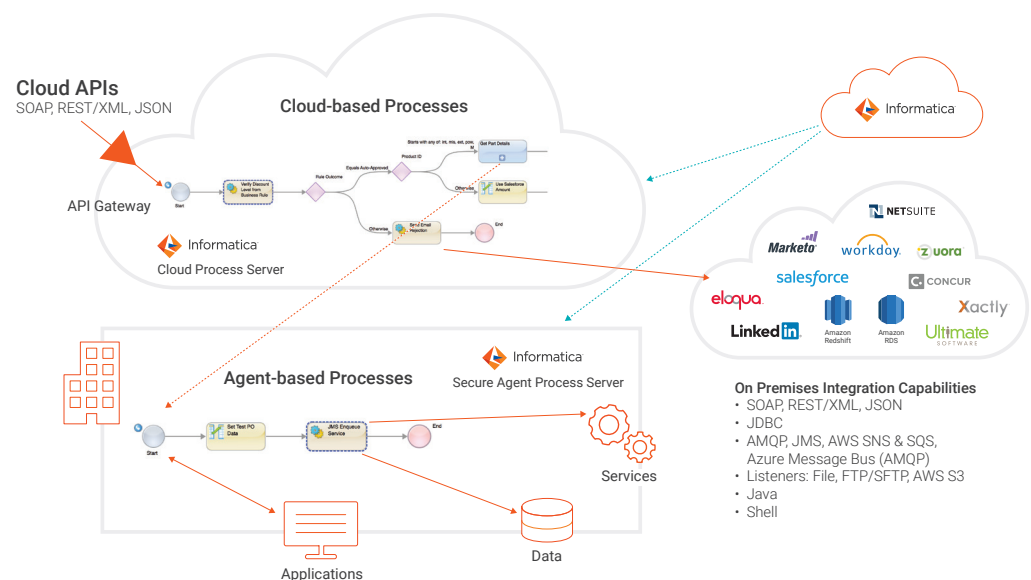
Secure Agent is used both for data integration as well as service and application integration. When licensed, Process Server is automatically installed on Secure Agent. Process Server deployed to Secure Agent is built on the same technology that runs in the cloud service in multitenant mode. This provides customers the ability to deploy process contributions to the cloud or Secure Agents.

Secure Agent can be installed in different configurations. For data integration payloads, a cloud runtime environment is offered to process data integration payloads by infrastructure managed by Informatica. When hosted by customers, agents can be grouped as Agent Groups to process in round-robin manner data and application integration workloads across agents of a group. Customers can also cluster Process Server instances of an Agent Group to provide a high-availability and fault-tolerant configuration. Clustering should be considered when processing long-running processes. This typically calls for automatically failing over process instance to another node in the event of a failure of a node.

Cloud and On-Premises Interaction

Incoming service (i.e., API) requests to a cloud-deployed process (depicted here) can originate from a cloud or on-premises consumer over JSON RPC and SOAP and REST (XML/JSON). These either initiate a new process or represent a callback or some event that the process is waiting to receive. An API gateway is offered to secure and apply various access policies to provider APIs.

Invoking cloud-based services (for example, Salesforce or NetSuite) employs the security mechanism offered by that service such as a SOAP endpoint's WS-Security username token or HTTP Basic Authentication. Invoking services on-premises is performed via a secure channel between a process instance running in Integration Cloud's CAI Process Server and an agent-based Process Server. Calls from Integration Cloud to the Secure Agent can be carried out only via the Cloud Process Server through a mutually authenticated session to ensure fully-secured access to on premise systems.



REST (XML/JSON) or JSON/RPC services exposed by customers are secured using HTTPS Basic-Auth or handled by third party OAuth providers. SOAP services exposed by customers are secured using Basic-Auth at the HTTPS layer. Additional forms of authentication are available via WS-Security in the form of WS-Security tokens. The Username, X.509, and SAML token formats are supported.

Based on its process definition, the Cloud Process Server receives and invokes service consumers and providers deployed to the Cloud. It also processes requests destined to on-premises service providers and responds synchronously to these using the HTTPS over TLS connection established by the service consumer.

Cloud to Secure Agent communication is performed using a Secure Channel created by the Secure Agent's Channel Service. Calls from Integration Cloud to Secure Agent can be carried out only by Integration Cloud through a mutually authenticated session.

Customers deploy process definitions and manage process instances from the Cloud Application Integration Process Console. Process administrators log in as a tenant and are granted access to tenant-specific data and configuration information. The same console used to access process definitions running in the cloud is used to access process definitions running on Secure Agents.

Access to Process Console provides customers access to transient data flowing through Integration Cloud. This provides customers access to variable data (for example, inputs and outputs to the process and services calls) of running process instances and completed or faulted process instances.

Process Console access to deploy process definitions or access to process instances are secured by an IICS username and password that customers manage in the IICS user and group store. SAML support is also offered.

Connectivity

Customers using Process Designer benefit from rich connectivity options:

Service connectors

- Allow customers to build REST (XML/JSON, JSON/RPC, or SOAP) service integration using a simple form. If the service offers a WSDL or Swagger interface document, the Service Connector can be created by importing the interface document.
- Allow customers to import and configure prebuilt business and data service definitions as reusable assets.

Data service connectors

- Afford customers with JDBC, OData, SAP Table Reader, SAP BAPI, Workday, and NetSuite capable of a variety of CRUD operations.
- Messaging services
- Provide built-in JMS, AMQP (includes Azure Service Bus), and Amazon Web Services SNS/SQS messaging services for queue and topic processing.
- File content listeners/writers
- Deliver data sets or discrete events that arrive on the file system, S3, FTP/s as well as the ability to generate and transfer file content to these targets

Service Steps (or Automated Steps) BPEL Definitions

Create reusable services built with Process Developer by developers that are directly consumable by Process Designer and make it possible to expose native Java integration among other uses.

Built-in services

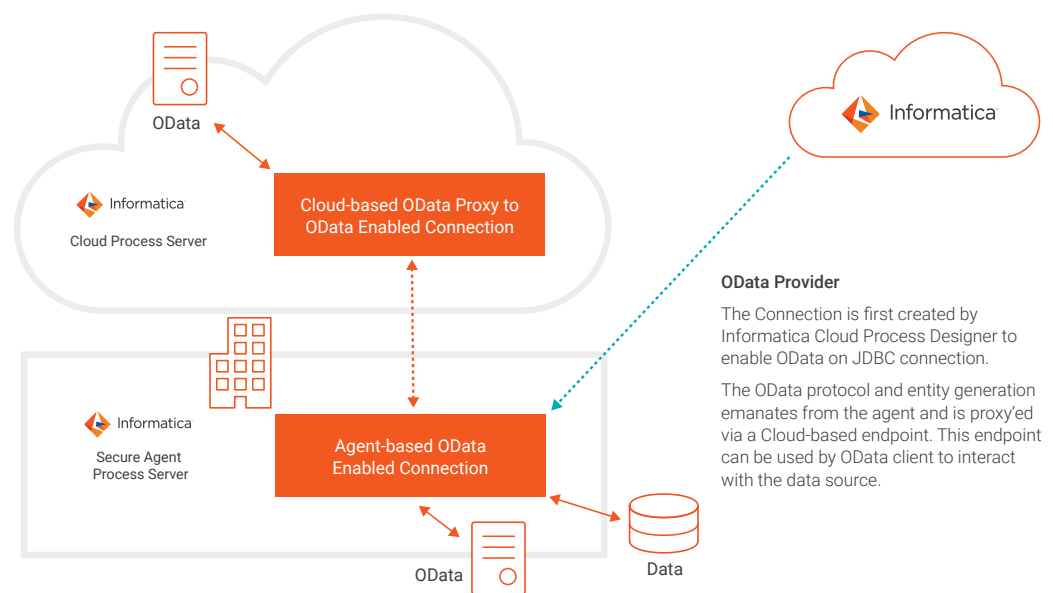
Deliver a data access service for direct SQL or stored procedure execution

Provide email services

Supply shell services to execute shell scripts and utilities

OData provider

Enables OData access to internal data sources such as those available via JDBC, Salesforce, and the SAP Table Reader. This allows OData clients such as Salesforce Lightning Connect to access OData streams across the Web and on-premises.



Interfaces and Protocols

CAI capabilities integrate people, processes, and services by leveraging industry standards. Services—whether exposed as SOAP, REST/XML, JSON, JMS/AMQP, or Java classes—are exposed to developers at design time as a service, thereby eliminating binding details to the underlying technology that implements these “services.”

Informatica’s services platform provides rich support for service interfaces and protocols. This is a natural result of supporting standards at its core. BPEL, the foundational component, is layered on top of and extends the WSDL service definition model. A common service interface is used to interact with several implementation types (for example, web services, REST, JSON, JMS/AMQP, and Java). Developers don’t need to be concerned about this abstraction. They simply use it.

Integrating with a service requires only a Swagger or WSDL interface. Importing this interface will create a service connector. If an interface is not available, then generating the service connector is as simple as using a form to specify input/output parameters, endpoint information, and test connection information and then saving and publishing the service connection. Once saved, the service definition is automatically incorporated as part of the services that can be used in the process and others that want to consume this definition. Swagger, WSDL/XML Schema, and OData introspection documents are automatically created for service consumers for any application or data service created within CAI.

The Informatica Process Definition (IPD) generated by Process Designer offers a simple abstraction over BPEL. The deployment of an IPD automatically generates the BPEL definition.

Message Exchange Patterns

A variety of message exchange patterns are available with CAI, which makes it possible to implement any multi-cloud and on-premises solution. These include:

- Synchronous request/reply
- One-way fire and forget
- Asynchronous request/reply
- Queuing and publish and subscribe
- Reliable SOAP message delivery with WS-Reliable Messaging

These message exchange patterns are available in the cloud and on the agent. The cloud–agent communication is automatically managed for developers.

Versioning

To isolate process versions and their artifacts, Process Designer and Process Developer package a process’s contents into an SCA “contribution.” Contributions are deployed to CAI’s Process Server or can be specifically deployed to a Secure Agent’s Process Server.

Process versioning and migration capabilities let you deploy multiple versions of a process. Processes currently running continue to run with the definition they started with, with new instances using the last deployed version of the process definition. You can also terminate or migrate pre-existing process instances to the latest version.

When using Process Developer, the developer needs only deploy a single contribution, all contained components are deployed automatically as a set—or for example, WSDL, XSD, and HTML as well as process definitions. Process Designer uses the same contribution mechanism but isolates the user from having to manage packaging, which is taken care of for the user. Contributions make it easy for developers to:

- Automatically manage versioning of the contribution and its artifacts
- Delete all old process instances and old resources by deleting the contribution
- Maintain your own resources so that they do not collide with those of other developers
- Roll back the current contribution to an earlier version

To support this, Process Server's Resource Catalog is versioned, implying that multiple versions of processes, WSDL, XSD, and POJO can be deployed and operate concurrently. At runtime, this ensures that the artifacts deployed by a contribution are the only ones it can access. The contribution's deployment log and Process Console's contribution detail page make it easy to understand dependencies and the artifacts that make up the contribution.

Monitoring

Process Server running in multi-tenant mode provides administration functions and monitoring that are used by IICS's Operations staff to manage its multi-tenant environment. It is used by tenants to access from a single cloud location instance details of processes executing on a Secure Agent's Process.

Process Console gives you visibility into built-in monitors, including:

- Process Monitoring's active process, alarm queue, and receive queue
- Secure Agent Process Server Monitoring's engine statistics and deployment logs

ESBs and Informatica Cloud Application Integration

Some ask how an enterprise service bus (ESB) compares to Informatica iPaaS integration capabilities. Succinctly:

- An ESB does a good job of routing messages between applications and services
- Informatica iPaaS offering is intended for event-driven and service-oriented application integration capabilities that encompasses event processing, service orchestration and process management. It makes it possible to create and consume APIs, orchestrate data services and business services, integrate processes, and offer data and applications services within and outside of an organization. It is better suited for service and event-based processing use cases for reasons described in this section.

ESB Routing and Transformation

An ESB's primary role is propagating data among endpoints using adapters (web services, FTP, File, JDBC, etc.) and protocols (HTTP, JMS), while enriching and transforming it using XSL and domain value mappings.

With an ESB, you can route service requests through a single proxy in a manner similar to a gateway. An ESB will typically perform its routing decisions based on message headers. Acting as an untyped service proxy—a proxy that works based on headers, without knowing or caring about the operations being called—an ESB might perform decryption, signature validation, authorization, and other tasks without having any hard-coded understanding of the types represented in the body of the message.

Contrasting ESB message routing and Informatica Cloud Application Integration functions

CAI's Process Server can invoke the same endpoints that an ESB offers, using similar communication mechanisms and patterns. Using Process Server, messages are received from an end system and processed. Process Server natively supports SOAP, REST, and JSON/RPC services, JMS (queues/topics), AMQP (queues/topics) (e.g. Azure Service Bus, RabbitMQ, ActiveMQ), AWS SNS/SQS, SQL Data Access, Shell Command Execution, and plain old Java Objects (POJO) as a means of interacting with systems. A variety of message exchange patterns are commonly used.

Process Server supports stateful and stateless execution, synchronous and asynchronous message exchanged patterns, and long running processes (with built-in fault recovery, compensation, and rewind), as well as offering built-in correlation. Process Server in Cloud and running on a Secure Agent offers enterprise performance and scale needed for mission-critical deployments through clustering and load-balancing.

Like many ESBs, Process Server transforms incoming data using XPath, XQuery, and JavaScript for data manipulation. ESBs and Process Server can both digitally sign and encrypt/decrypt messages.

ESB technology and Process Server both support dynamic endpoint selection. Routing can be controlled within the process using the payload's data to perform a routing decision. The caller's identity can also be used to make routing decisions, or endpoints can be statically assigned through configured through URN indirection.

Unlike ESBs, Process Server provides rich semantics that ESBs do not, such as parallel execution and `forEach/while/repeat` until constructs. Exceptions are caught and the developer has the means to control how and what to compensate. Timers and event triggers are built in along with the associated event handlers.

Most importantly, unlike ESBs, processes can be stateless or be fully stateful. This means for example, not only can you process an order using a long-running process and handle asynchronous callbacks, but you can also update the order information, request the status of the order, and cancel the order. This type of functionality would need to be built into endpoints. With a stateful process, the process holds and manages the state of the order.

ESBs and CAI's Process Server can be combined to leverage each of their respective strengths when building applications. You can use a pre-existing ESB to implement message routing and transformations and message-level monitoring. And, you can use Process Server to build complex business process applications using services, some of which accessed by or exposed from the ESB. Essentially one can regard an ESB as a source of web service endpoints that the CAI service orchestrates by sending and receiving messages to the ESB.

That said, CAI does not require an ESB. The service supports a wide variety of application and service endpoints: RESTful services, RPC services (JSON and SOAP), JMS/AMQP queues and topics, SQL DB access, Plain Old Java Objects, command shell utilities, and EJBs. If you already have access to the systems and services you need, you can develop your business process applications and integrations with the CAI service.

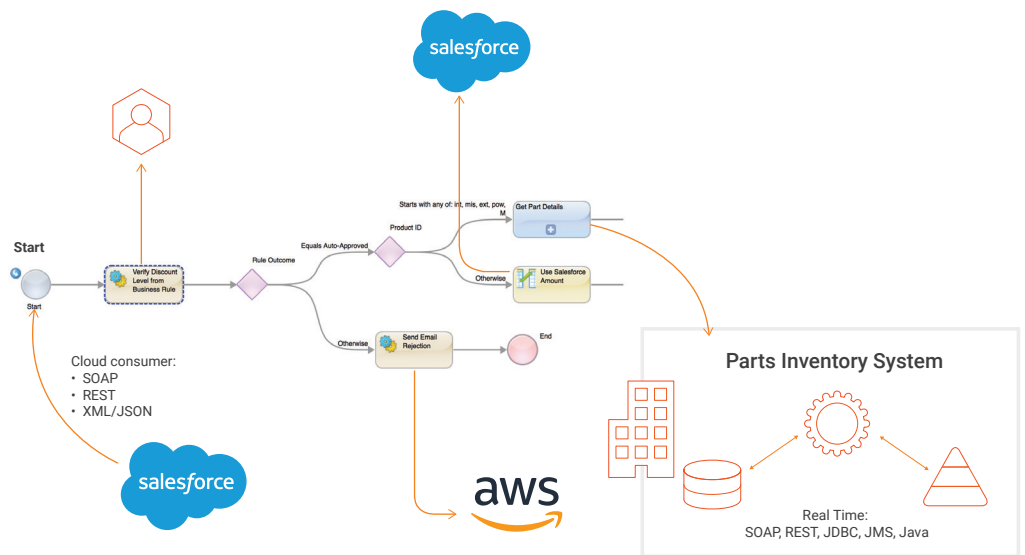
Summarizing, Informatica's Cloud Application Integration capabilities are better-suited for service-oriented integration than ESBs particularly when you need:

- Long-running transactions that maintain state
- Short-running or transactional system integration processes requiring integration sequences, different execution paths, or composite transactions
- Rich semantics for parallel execution
- Timers and event triggers
- Rich event, fault, and error handling systems controlling how and what to compensate through automated compensation to roll back a transaction if all required steps are not completed successfully
- The ability to orchestrate transactions that spawn across different companies, business units, different products, or services to realize horizontal business/integration processes, such as an order-to-cash process
- Visibility into what is going on during execution such as knowing what is happening and what is not happening; reporting on processes that are in progress, not just individual requests; and managing escalations, timeouts, and schedules

A Cloud Application Integration Example

To make this more concrete, let's look at an example. This example demonstrates how an order submitted by a service API consumer (for example, a website) invokes CAI implemented using a process to:

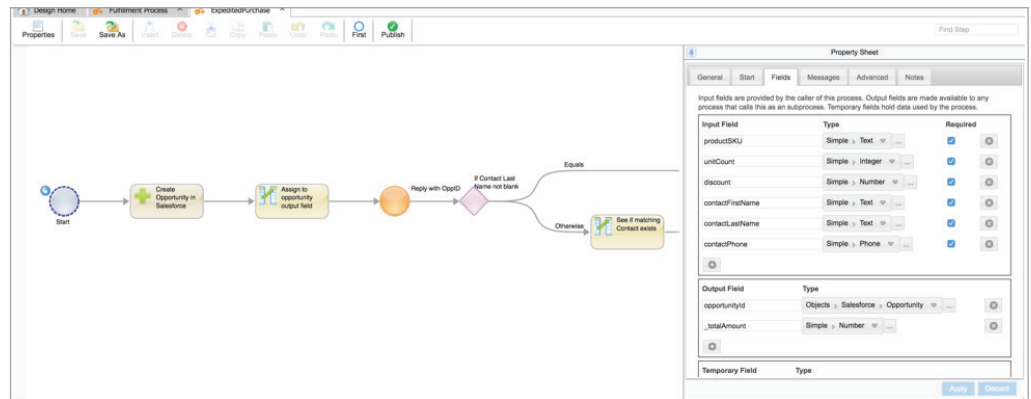
1. Pre-process an order (a part order in this case) by first creating an Opportunity object in the CRM (Salesforce in this case)
2. Registers information about the individual the website is placing an order on behalf of
3. Invoke a fulfillment process (depicted below) that
 - a. Invokes a rule service to determine if the website's proposed discount is appropriate
 - a. Based on the type of part, obtains part pricing and availability information from Salesforce or an inventory database
 - a. Initiates the fulfillment of the order using the Shipping service



This orchestration is initiated by the website that send a JSON/RPC request to the ExpeditedPurchase service. The API takes as input a CRM Account from the URL (e.g. [CAIS URL]/ExpeditedPurchase/id/001F0000013oHSKIA2) and the body of the JSON request contains to request shown here.

```
{
  "productSKU": "int1782",
  "unitCount": 1,
  "discount": 10,
  "contactFirstName": "John",
  "contactLastName": "Snow",
  "contactPhone": "781-555-1212"
}
```

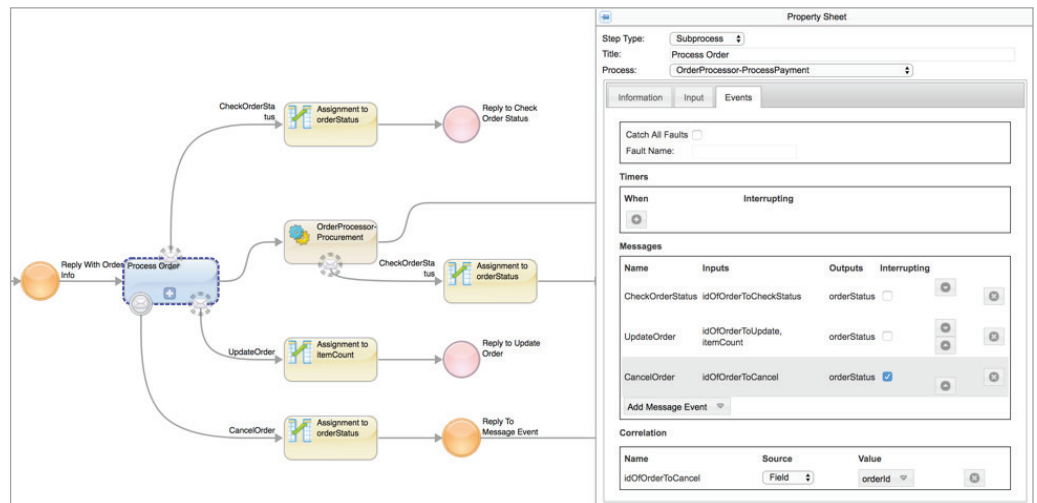
To process this request, the designer of the orchestration defined a simple set of input fields to match the content of the request. As a second step, an opportunity is created in Salesforce, the created Opportunity ID obtained from Salesforce, and the ID is returned to the caller as shown below.



The Opportunity ID returned to the client (i.e. the website) will be returned as

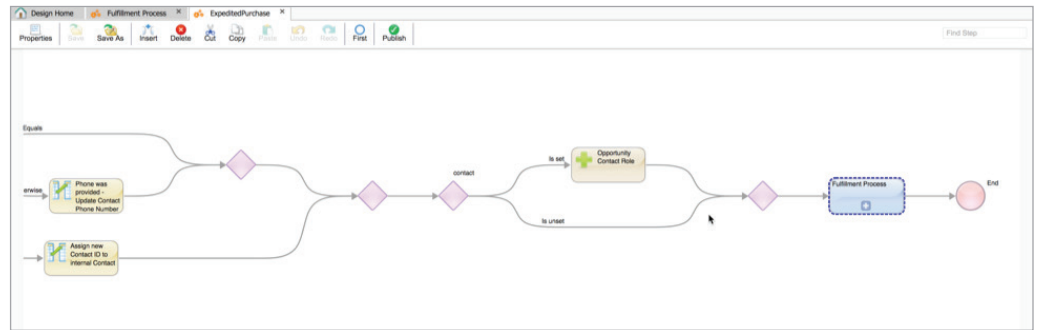
```
{
  "opportunityId": "0060G00000pczXpQAI"
}
```

The use of IDs for example is useful to correlate callbacks. For example, an orderId might be used to process the cancellation of an order as depicted here.



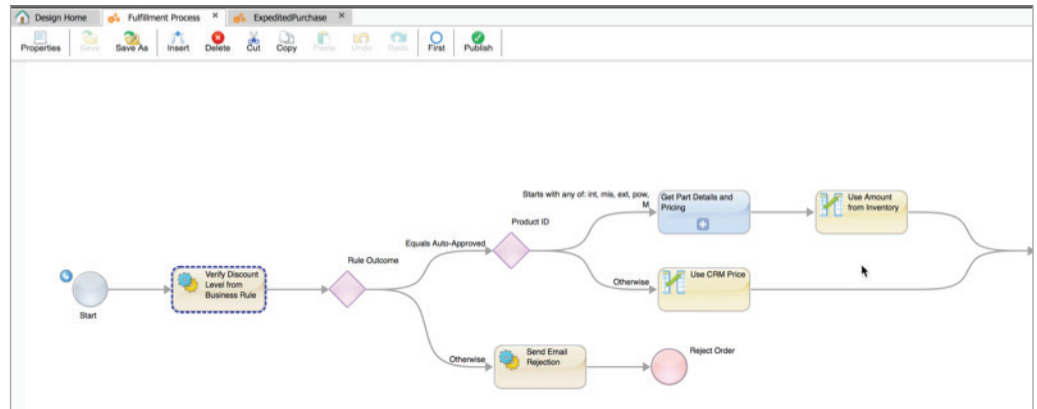
Once returned, the ExpeditedPurchase process continues its pre-processing, and updates contact information in Salesforce before finally proceeding to the fulfillment phase of the order process.

The Fulfillment Process selected here (in blue) is invoked for this purpose.



The Fulfillment Process has three primary tasks:

1. Verify that the discount is appropriate
2. Determine pricing and availability either via the CRM or an inventory database
3. Complete the order fulfillment by invoking the Shipping service

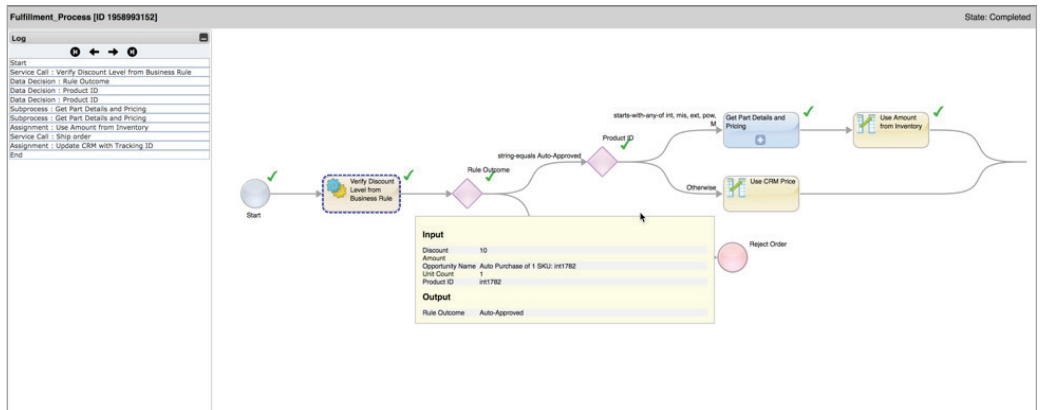


The Process Console depicted below shows the instance of the Fulfillment_Process with process ID 1958993152. You'll note that several processes instances (for example, AutoApprovalDetermination, GetPartsDetails, and Order) were instantiated. This demonstrates reuse of services (that is, process orchestrations) that can be orchestrated in a variety of ways, in this case by the Fulfillment_Process.

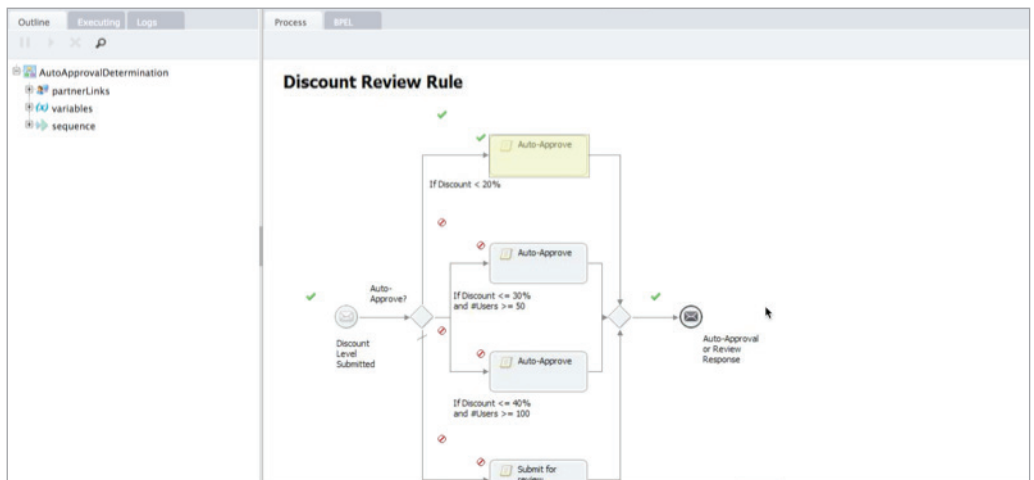
ID	Title	Ver.	Start Date	End Date	State	Tenant	Group
1958993792	Order	10.0	2017-06-02 19:57	2017-06-02 19:57	Completed	000063	Fulfillment
1958993664	OrderProxy	20.0	2017-06-02 19:57	2017-06-02 19:57	Completed	000063	
1958993536	GetPartDetails	10.0	2017-06-02 19:57	2017-06-02 19:57	Completed	000063	RT.User
1958993408	AutoApprovalDetermination	10.0	2017-06-02 19:57	2017-06-02 19:57	Completed	000063	Fulfillment
1958993280	AutoApprovalDeterminationProxy	22.0	2017-06-02 19:57	2017-06-02 19:57	Completed	000063	
1958993152	Fulfillment_Process	127.0	2017-06-02 19:57	2017-06-02 19:57	Completed	000063	RT.User
1958993024	ExpeditedPurchase	97.0	2017-06-02 19:57	2017-06-02 19:57	Completed	000063	RT.User
195855904	OppToOrder	4.0	2017-05-30 00:27		Running	000063	RT.User
1958622080	OppToOrder	3.0	2017-05-30 00:22		Running	000063	RT.User
1958354432	Get_Approval_Answer	17.0	2017-05-26 19:44		Running	000063	RT.User
1958354304	Opportunity_to_Order	42.0	2017-05-26 19:44		Running	000063	RT.User
1958354048	Get_Approval_Answer	17.0	2017-05-26 19:44		Running	000063	RT.User
1958353920	Opportunity_to_Order	42.0	2017-05-26 19:44		Running	000063	RT.User
1957535872	Opportunity_to_Order	42.0	2017-05-16 22:59		Running	000063	RT.User
1957528704	Opportunity_to_Order	42.0	2017-05-16 21:03		Running	000063	RT.User

The Process Console provide the execution details of the ExpeditedPurchase service. The Process Detail View describe inputs, outputs and the paths of execution taken to complete the process. Timing information of each step and the ability to rewind a suspected process to an earlier state is offered by the Advanced View.

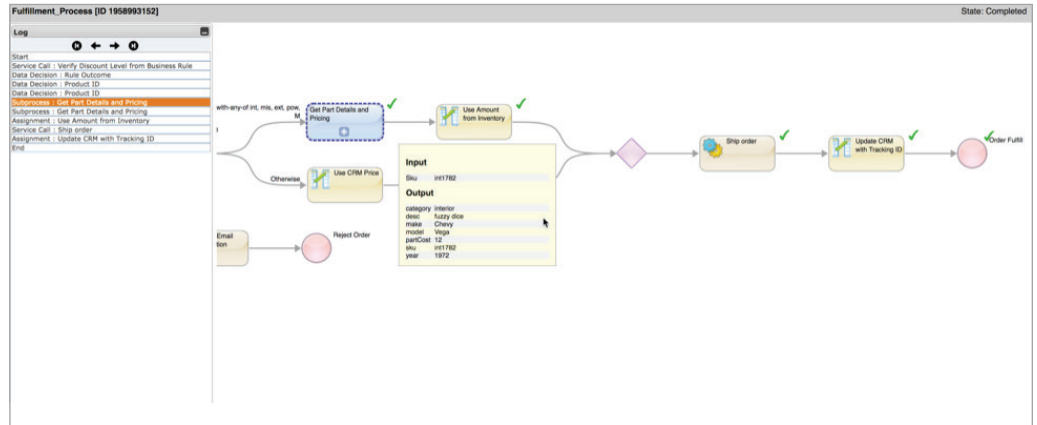
As part of its orchestration, the Fulfillment_Process process invoked the "Verify Discount Level from Business Rule" service (an orchestration) to determine if the discount is approved.



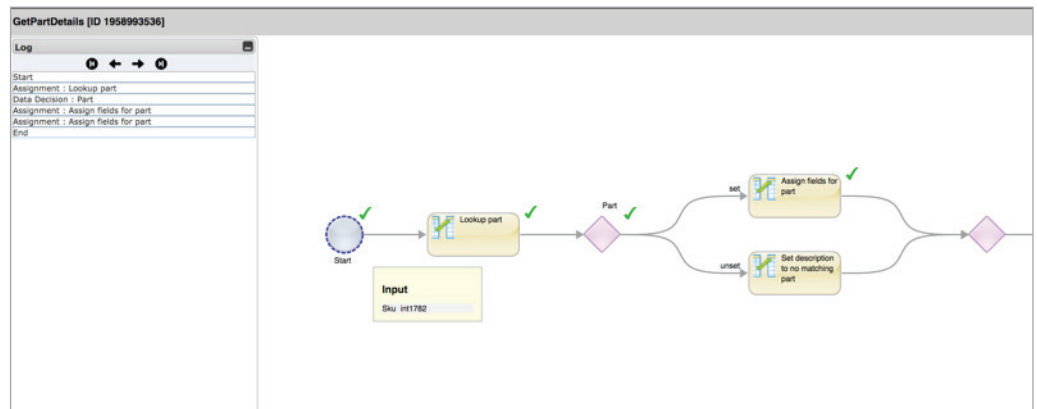
Albeit simplistic, the "Discount Review Rule" implemented using Process Developer (an Eclipse-based process) returns a decision. A rules engine would typically fulfill this role. This demonstrates how Process Designer- and Process Developer-based processes can be used together.



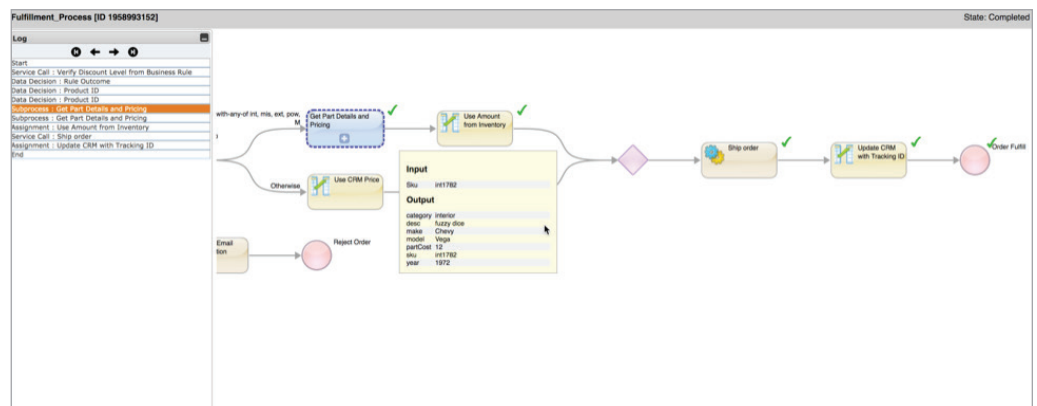
The Product ID conditional branch that make its decision on the type of part (e.g. int1782, the productSku property in the message shown at the beginning of this example) requires a lookup for pricing information from an inventory database. The “Get Parts Details” service (a process) is used for this purpose. It returns pricing and parts detail information as shown below.



To perform its lookup, the “Get Parts Details” service makes use of the JDBC connector to lookup the part’s details using a simple select statement.



The final phase of the orchestration consists of invoking the Shipping service using as input the shipping and parts information obtained from the CRM and the inventory database.



Conclusion

Informatica Intelligent Cloud Services (IICS) supports the next generation of integration platform as a service (iPaaS) integration patterns. Cloud Application Integration (CAI), offered by IICS, provides a unified development environment and a breadth of functionality that guarantees superior ease of use, including a forms-based Service Connector tool that integrates any API with ease, advanced orchestration design capabilities, and ease of deployment.

Unlike traditional ESB-based solutions, CAI manages the state of orchestrations and business processes for you—system-to-system interactions, be they synchronous, asynchronous, long running, or short-running. It makes it easier to define and operate sophisticated and truly reliable business processes and integrations that give you a competitive advantage.

If you are struggling to succeed with your application integration projects using traditional ESB or similar methods, [contact us](#) to learn how Cloud Application Integration can help your organization.



Worldwide Headquarters 2100 Seaport Blvd., Redwood City, CA 94063, USA Phone: 650.385.5000, Toll-free in the US: 1.800.653.3871
www.informatica.com linkedin.com/company/informatica twitter.com/Informatica

IN09_1120_3407

© Copyright Informatica LLC 2017. Informatica, Informatica Intelligent Cloud Services, and the Informatica logo and are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners. The information in this documentation is subject to change without notice and provided "AS IS" without warranty of any kind, express or implied.