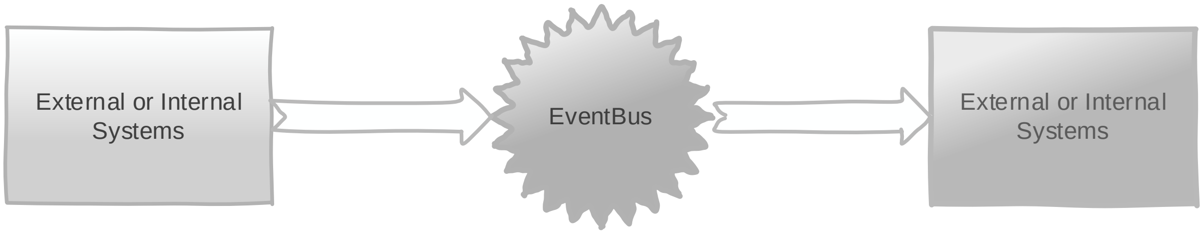
IXN EventBus

Product Description

The EventBus is a self-contained system that is capable of importing, updating and exporting system data to and from other systems. The EventBus orchestrates business process workflow into steps that can be configured to replace manual processes. The EventBus gives a menu of process steps that are either in common with all process or unique to a specific process.



IXN's EventBus provides data exchange services by offering an integration engine to push or pull data through interface services between enterprise systems. The EventBus architecture includes a framework for exporting, transforming, and loading (ETL) data. Custom programs can be included and the EventBus integrates with the SQL Server Integration Service (SSIS), custom web services, Secure File Transfer Protocol (SFTP) services, and other products.

The EventBus manages requests received in a queue, orchestrating system processing based upon events in a framework across distributed applications and micro services.

The EventBus Design Pattern

The term EventBus describes a mechanism that allows different components to communicate with each other without knowing about each other. A component can send an Event to the EventBus without knowing who will pick it up or how many others will pick it up. Components can also listen to Events on an EventBus, without knowing who sent the Events. That way, components can communicate without depending on each other. Also, it is very easy to substitute a component. As long as the new component understands the Events that are being sent and received, the other components will never know.

So what exactly is a component here? Well, a component could be anything. In the Android operating system (OS), EventBuses, are Java Objects. They send Events and they also listen to Events. Methods within a web service could communicate events to the EventBus, or they could receive transactions triggered by events on the EventBus.

And what are Events? Well, they are basically the messages that get sent and received by the components. Typically, they contain everything that the receiver needs to know in order to process the Event. A system clock also creates events so the EventBus can do tasks based upon daily or calendar events. Everything else about EventBuses is pretty much implementation dependent. Typically, Java EventBuses require the sender of an Event to create an Event Object, which can be filled with data for the receiver. The sender then calls EventBus.send(event). The receiver has to implement a certain interface with an nEvent(Event e) method, that gets called by the EventBus.

A.J. BOGGS & COMPANY

The EventBus makes it easy to build new applications by coordinating the workflows in a reliable way to coordinate components and step through the functions of an application.

EventBus functions automatically trigger and track each step, and retries when there are errors, so your application executes in order and as expected. It logs the state of each process, so when things do go wrong, you can diagnose and debug problems quickly. You can change and add steps without even writing code (in the Job Workflow Table), so you can easily evolve your application and innovate faster.

The data can be managed in several ways:

i. Data format can be custom though these standards formats are preferred:
   a. XML
   b. JSON
   c. CSV
   d. PDFs
   e. Word or Excel
   f. Fixed Field Length (or other custom types)

ii. Data Triggers:
   a. Scheduled sends (typically nightly or weekly)
   b. Queued records sent on ping (hit the web service with the “Get Updates” web method to pull down newly queued records, based on a send criterion)
   c. Complete or editing of completed records (Typical)
   d. Workflow triggers such as a record status, such as particular roles approving, deleting, editing, or archiving a record.

iii. The ways data can be sent:
   a. Web services over HTTPS
   b. Files placed on an sFTP site (customers or provided by the IXN data center)
   c. Manually pull the data via the web site user interface

iv. Data mapping to provide transformations of records to new forms or send to new systems
   a. Provide a drag and drop interface to configure the transformation mapping

v. Importing Methods:
   a. FTP
   b. XML
   c. JSON
   d. Manually uploaded via the website
   e. Import Excel or CSV files for batch uploads

List of EventBus Web Methods

The following operations are supported by the IXN EventBus:

AuthenticateUser - This function verifies the user name and password.

ChangeStatusId - This function changes the status of a record based on form result ID.

DeleteAssessmentFormResultId – This function logically deletes records by the form result ID.

GetDataFormResultID – This function gets the data for a record by form result ID.

GetPdfReport - This function returns the PDF data for a record with the given form record.

GetFormSchema - This function returns the XML data schema for the requested form.

GetFormResultIdsFromTackingNumber – This function returns the formResultId of a record by Tracking #.

GetStatusId - This function gets the status of the record by form result ID.

GetTrackingNumberFormResultId – This function gets the tracking number for a record by form result ID.

GetUpdates - This function gets form data in XML format that has been added to ActivityRequest table since the last time the GetSisUpdates web method was polled.

GetFormEditURL – This function returns a link directing the user to the edit screen of the website for the formResultID specified.

a. The link will only be returned by this method if web service authenticates successfully and write access is permitted.
b. The link is only valid for 5 minutes. If the user does not redirect within that time frame this method will need to be called again.
c. After successful redirect the user may use the application under a normal session which may last up to 4 hours. The link may expire after 5 minutes, the session does not.

IsCompleteFormResultId – This function checks to see if the form is completed.

SaveDataGetKeys - This function saves the form record.

VerifyHost - This function checks to see if the web service is accepting requests.

GetUserInfo - Retrieves the user's info for a particular UserId.

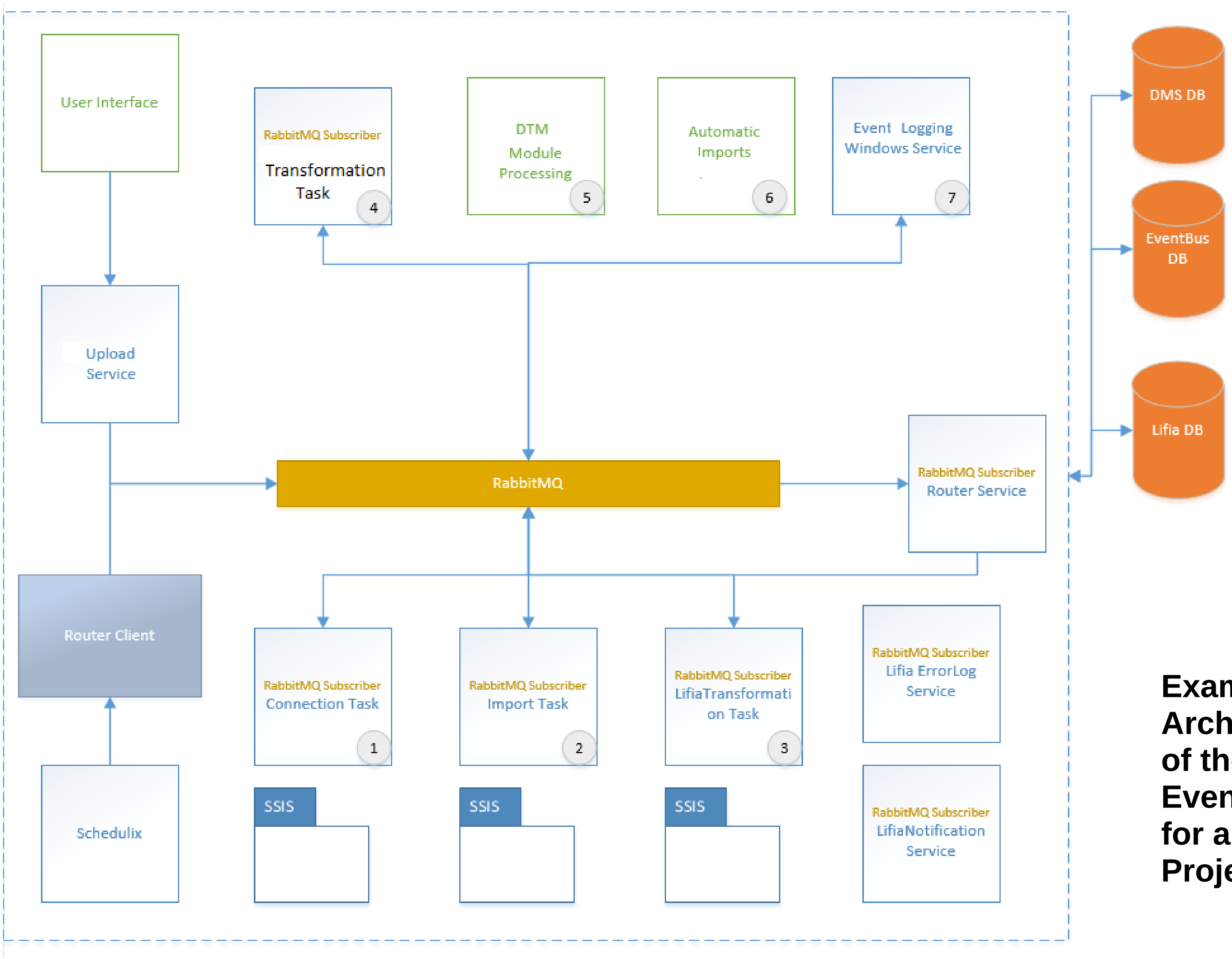
CreateUser - Creates a new user.

UpdateUser - Changes the user's records.

DeleteUser – Removes a user from active.

GetAllUsers - Retrieves the list of users for a particular user, based on the requesting user's level of access.

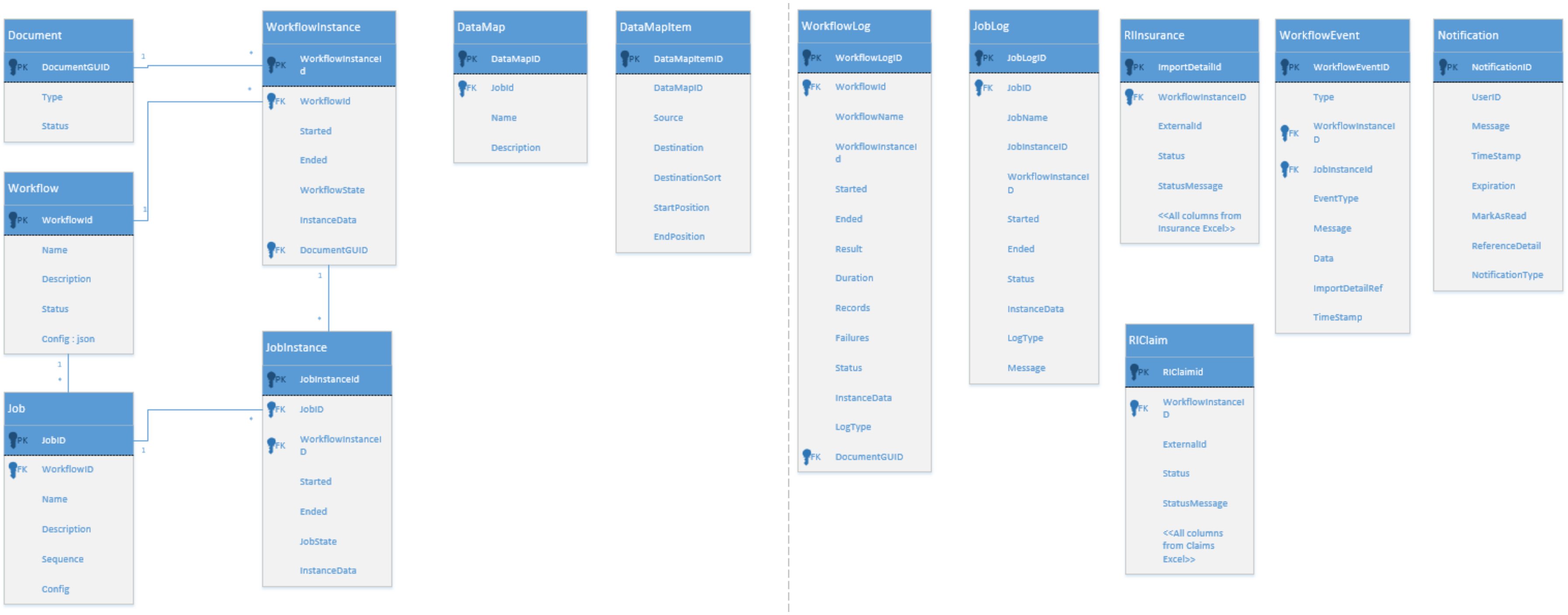
The Form Result IDs and Tracking Numbers are part of the DXF system for forms management.



Example Architecture of the EventBus for a Similar Project

EventBus Data Dictionary

Import & Job Tables UML Database Entity Relationship Diagram



Document	
PK	DocumentGUID
	Type
	Status

Workflow	
PK	WorkflowId
	Name
	Description
	Status
	Config : json

Job	
PK	JobID
FK	WorkflowID
	Name
	Description
	Sequence
	Config

WorkflowInstance	
PK	WorkflowInstanceId
FK	WorkflowId
	Started
	Ended
	WorkflowState
	InstanceData
FK	DocumentGUID

JobInstance	
PK	JobInstanceId
FK	JobID
FK	WorkflowInstanceID
	Started
	Ended
	JobState
	InstanceData

DataMap	
PK	DataMapID
FK	JobId
	Name
	Description

DataMapItem	
PK	DataMapItemID
	DataMapID
	Source
	Destination
	DestinationSort
	StartPosition
	EndPosition

WorkflowLog	
PK	WorkflowLogID
FK	WorkflowId
	WorkflowName
	WorkflowInstanceId
	Started
	Ended
	Result
	Duration
	Records
	Failures
	Status
	InstanceData
	LogType
FK	DocumentGUID

JobLog	
PK	JobLogID
FK	JobID
	JobName
	JobInstanceID
	WorkflowInstanceID
	Started
	Ended
	Status
	InstanceData
	LogType
	Message

RIInsurance	
PK	ImportDetailId
FK	WorkflowInstanceID
	ExternalId
	Status
	StatusMessage
	<<All columns from Insurance Excel>>

RIClaim	
PK	RIClaimid
FK	WorkflowInstanceID
	ExternalId
	Status
	StatusMessage
	<<All columns from Claims Excel>>

WorkflowEvent	
PK	WorkflowEventID
	Type
FK	WorkflowInstanceID
FK	JobInstanceId
	EventType
	Message
	Data
	ImportDetailRef
	TimeStamp

Notification	
PK	NotificationID
	UserID
	Message
	TimeStamp
	Expiration
	MarkAsRead
	ReferenceDetail
	NotificationType

EventBus Security Features

The EventBus incorporates security features as integral components of its designed capabilities. These include compliance with HIPAA security standards and adherence with OWASP recommendations for software system security.

Local Assessment Data Replication for Custom Reporting

Software can be provided to install in the customers' data center to provide a local copy of the data on a SQL Server, simplifying creating custom reports. Data can be to automatically replicated or queued to be sent on demand. Security features ensure HIPAA compliance. The IXN support team, a business unit of A.J. Boggs & Company, is available to assist with the required setup and configuration of EventBus systems.

Other EventBus Features

The EventBus incorporates security features as integral components of its designed capabilities. These include compliance with HIPAA security standards and adherence with OWASP recommendations for software system security.

1. *Connection Service*

Allows the EventBus to connect using a variety of protocols and methodologies to receive and deliver data to the EventBus or to send data from the EventBus.

2. *Database Service*

Allows the EventBus to extract or load data into any type of database.

3. *Router Service*

Allows the EventBus to initiate predefined workflows on a scheduled or real-time basis.

4. *Data Factory Service*

Allow the EventBus to separate a batch of records into individual records for upstream processing or to create a batch of records from a collection of individual records.

5. *Transformation Service*

Allows the EventBus to transform any record from its current format to a different record format.

6. *Event Service*

Allows the EventBus to process all logging events or any special events required. This feature allows the configuration of notifications to occur through email or user interface updates.

7. Enterprise Scheduler

A scheduler allows the EventBus to run processes on a scheduled basis based on the business need for any given business need.

8. EventBus User Interface (UI)

The front end (UI) of the EventBus provides users the systems needed to configure and monitor performance statistics of data exchanges and other processes supported by the EventBus.

a. Performance Dashboard

A web page that presents different indicators that will help to evaluate and monitor the performance of the EventBus data transformation and transfer.

b. User Manager

The EventBus includes a user administration subsystem (UAS) in which is configured for each EventBus implementation so that Admin users can create different types of users with various roles and permissions. The EventBus UAS is used to manage user contact information, groups, permissions, roles, user names, and passwords.

c. Notifications

The EventBus offers notification methods for users to receive notices of specific events, such as successful interface transactions or data transfer errors. Users can choose their preferred method of notification and other parameters to receive a notification.

d. Audit Trail

The EventBus can track events in audit trail records that can be provided to applications for reporting.

9. Reporting Service

Generates reports based on any activity within the EventBus. For example: Quantity of transferred records; Number and type of data transfer errors; Data transformation performance (transaction velocity times); and Database of archived transactions and other data sets.

For more information, contact IXN at info@ixn.com or 1-877-IXN-4IXN.