Journal of Technological Innovations

Est. 2020

Enterprise Data Integration Architecture

Naveen Muppa

10494 Red Stone Dr

Collierville, Tennessee

Abstract:

This paper focuses on the Functional and Technical design of the CIH architecture and all the relevant publications and subscriptions.

Keywords: The data hub is an integration architecture that helps determine effective mediation of semantics (governance and sharing) by providing a unified and efficient data exchange infrastructure to collect and connect data across applications, enterprises and ecosystems.

Below are the Key functions of the Data Hub:

- Orchestrate Data Integration across multiple sources – on-premises, Cloud through a governed publication/subscription hub.
- Connectivity to ingest all data - Any Type, Any Scale, From Any Source
- Flexible, agile, and efficient architecture that reduces dependency on source system and eliminates the traditional point-to-point integration approach, by reuse and consistencies.
- Foundation for future Analytics and MDM capabilities: advanced analytics, machine learning, big data, etc
- Centralized monitoring and alerting for improved governance. Technical Capabilities:
- Data Integration Tools (Bulk/Batch, Real time)
- Application Integration (API)
- Common Data Models
- Data Governance Controls
- Persistent Storage Technology Components:
- Informatica Cloud Data and App Integration Platform (iPaaS)
- Informatica Cloud Integration Hub (CIH)
- Data Store in Azure (Blob, Relational/NoSQL)

1. Introduction

The design and development of Enterprise Data Integration architecture is vital role in any organization. While it is undergoing application modernization for ongoing business data integration needs, the essential design has remained the same. The data integration capability essentially involves a one-way flow of data from key 'core' systems:

The data flow from thecore systems to numerous other 'consuming' applications within the enterprise. The implementation for these data flows were done via data to database replicas of these systems maintained in a central operational data. Point-to-point integrations move data from one application to each consuming application. This architecture and point-to-point integrations are difficult to maintain as it becomes challenging during this application modernization.

The vision is to have an architecture that is decoupled, agile, modern, and promotes reusability for futuristic digital transformation. Build a data integration platform using Informatica CIH to support the different patterns of Integration such as ETL (Extract, Transform and Load), Pub-Sub (Publication Subscription model) and real-time/ service-based integration and batch requirements.

2.Informatica Intellegent Cloud Services(Iics)

IICS platform is built for the future to provide complete end-to-end data management in a uniform, non-siloed approach. IICS unifies existing Informatica cloud service offerings and expands into a full suite of cloud data management services over time. It has been tailored to fit the current project needs. Below are the important components and their functions for the proposed solution.

Service	Functions
Application Integration	Provides API based integration, event processing and service and process orchestration. Integrated with API Integration Console to manage APIs and Business Processes
Data Integration	High Performance, cloud data warehouse for huge volumes of data using advanced transformations, pushdown optimization, synchronization, replication, asset management, pre-configured templates and mappings.
Integration Hub	Provides a modernized hub-based approach for data integration and application synchronization. Boost productivity and data availability with publish-subscribe integration pattern.
Administrator	Provides organization management capabilities across all cloud services including user, security, licenses, run time and connection management.
Monitor	Analyses state of orchestration and deployment activities across all cloud services and provides in-depth actionable insights.

Table: Glossary of services and functions

3. Key Components In The Architecture

Pub/Sub Pattern:

- Publication Process:

The publication process includes retrieving the data from the publisher, running the publication mapping, and writing the data to the relevant topic in the publication repository. After the publication process ends, each subscriber consumes the published data according to the schedule and the filter that you define when you create the subscription. In addition, in our process after every publication a Mapping task subscription will be triggered to load the Topic table's data into Datastore.

** As part of this solution to increase reusability and to mitigate any future risks if we add/remove any attributes – Mappings are considered instead of DSS tasks.

The following flow shows the main stages of the management and Application publication process for publications:

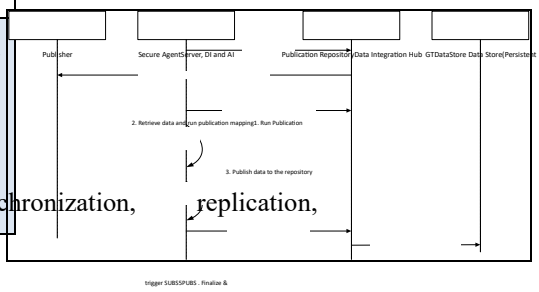

Figure1 Processes for Publication are not used downstream, and which are only for application specific.

The following flow shows the main stages of how integrated within the

The subscription process includes retrieving the required data from the Data Integration Hub publication repository, running the subscription mapping, and writing the data to one or more subscriber targets. Data Integration Hub keeps the data in the publication repository until the retention period of the topic expires.

As soon as all the Topics and topic tables are published, all the data from these topics and topic tables will be subscribed into the Data Store, a persistent database on Azure.

For all the applications that are being configured to run the APIs for Subscription, the tasks will retrieve data from Datastore but not from Data Hub. For all the real time feeds, Informatica Data Integration or Application integration tasks (mappings, task flows, etc.) are considered, but subscription Power Exchange CDC comes into consideration as all the real time feeds are dependent on the condensed files for capturing few attributes, which will be consumed by other applications.'

The following image shows the main stages of the process for each subscription:

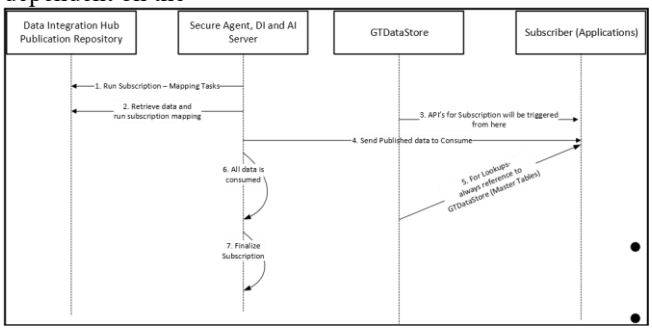

Figure 2 Processes for Subscription

- Bound Subscription will subscribe the data through mapping task from Topics to the Applications.
- A Publisher for Full load will be loaded from Integration Tasks without Topics/Topic Tables Datastore to Full Load Topics

(ETL Pattern):

Informatica Data Integration or Application integration tasks (mappings, task flows, etc.) will be developed and configured to run on the source data and the target data is staged in the *Data Store*

- Subscription Process: the DT Data Store Stage is process:

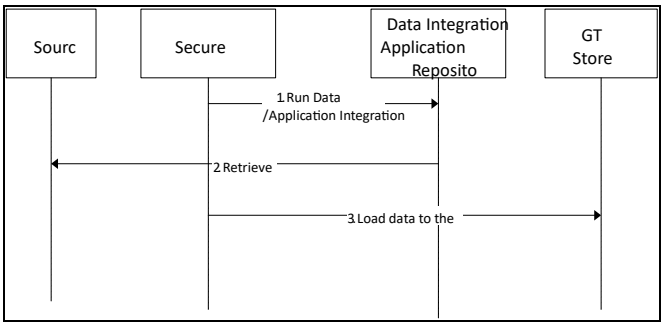

Figure 3 Processes for Topic

Integration Tasks for Real Time applications:

Integration or Application integration tasks (mappings, task flows, etc.) are considered, but subscription Power Exchange CDC comes into consideration as all the real time feeds are dependent on the condensed files for capturing few attributes, which will be consumed by other applications.'

4.Handling Cdc & Full Load Process

All CDC logic will be built from Source to CDC Topics in their Publication.

Bound Subscription will subscribe that data through a Mapping task from Topics to Datastore

- Unbound Subscription will subscribe that data from Topics.

Stage database on Azure for further processing. This route will be taken for application data which

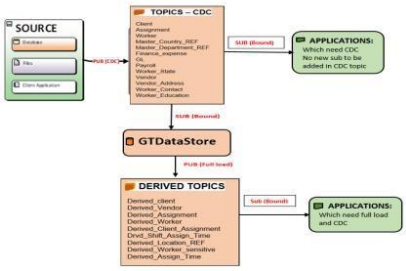

Figure 4 Processes for Full Load Topics

5.Initialize And Generate Automatic Parameter File

- This task takes the process name and application name parameter and executes the batch job to call the stored procedure, which in turns does the following activities.
- While inserting a record in Process table, it checks for the below conditions and picks up the respective values to insert:
- Checks for IsOverrideETLProcessID (default is 0), for each task from TaskParameter table.
- If IsOverrideETLProcessID=0, then Identity column in Process Table: ProcessID= ETLProcessID in Process table.
- If IsOverrideETLProcessID>0, it picks up the mentioned ETLProcessID from the TaskParameter table.

- Checks for the override parameter value (IsOverride) for each task from TaskParameter table (default is always 0, 1 means override)
- If IsOverride=0, then the latest TaskStartDate for that Task from Task Table where Status='Success' is updated into TaskParameter Table as SourceExtractStartDate. And Then, SourceExtractEndDate=getdate ().
- If IsOverride=1, it picks up the mentioned SourceExtractStartDate and SourceExtractEndDate from the TaskParameter table.
- Generates the dynamic parameter file by merging static file and the output of TaskParameter table • For PUB/SUB MCT's, the variables are set at the mapping level itself. (SourceExtractStartDate, SourceExtractEndDate, ModifiedDate, ETLProcessID)

6.Delta Detection

When the data in a source system is frequently updated, it is necessary to capture the updated information to the target extracts. However, due to high volume and load window, it is desirable to consider only the updated delta information, rather than reloading the entire source table. Usually, there is aModified timestamp column in the source table. This column can be used to filter the source records, based on the last source extract end time of that task.

This is implemented using a common framework component to generate a dynamic parameter file for each task flow, based on the previous successful execution status in the Task table, Stored procedure picks the SourceExtractEndDate for that task and updates the TaskParameter table SourceExtractStartDate = Task.SourceextractendDate and SourceExtractEndDate = GETDATE.

7.Folder Structure

The below Project structure facilitates the coordination and implementation of data hub using different services in IICS. Its main reason is to create an environment that fosters interactions among the team members with a minimum number of disruptions, overlaps and conflict. The structure of the directories created in the repositories always follows the below hierarchy.

- Application Name

- API (Process, Process Objects, etc.)
- ETL (Mappings, Mapping tasks, DSS tasks, etc.)
- Orchestration
- Publisher (All mappings that are in the Publications for the application)
- Subscriber (All mappings that are in the Subscriptions for the application)

Figure 6 Folder structure **Folder structure in integration hub:**

In Integration hub, assets like Application, Topic, Publisher, and subscribers are managed.
Folders in CIH are predefined based on Applications, Topics, Publications and Subscriptions.
There is no provision to update folder structure manually.

7.Version Control

Version control is not available in Informatica Cloud. All developers must save their changes religiously and always make sure to take backups regularly.
Run team is working on a solution for Version Control which is out of scope for this project.
The below is the proposed solution for code management. Even though it is not in scope of this project, below is the future roadmap for handling code version management in IICS as the tool doesn't provide any version control.

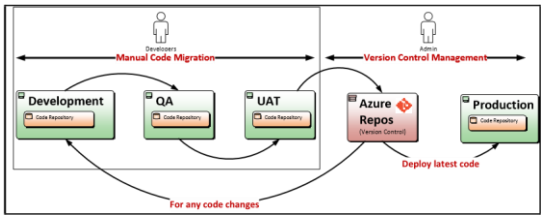

Figure 5 Deployment Flow

[1] https://www.informatica.com/blogs/welcome-toinformatica-intelligent-cloud-services.html

[2] https://now.informatica.com/IICS-Cloud-DataIntegration-Services-onDemand.html

8.References

The following section summarizes the links to external resources that this document references. The aim of this section is to make it easier for you to add links to your own documentation.