d

Mediation Architecture Document

Contents

[Document Purpose 2](#_Toc422132485)

[System Overview 3](#_Toc422132486)

[Data Flow 3](#_Toc422132487)

[Technical Aspects 4](#_Toc422132488)

[Data Flow Stages 5](#_Toc422132489)

[Processing Parts 7](#_Toc422132490)

[Receive Adapter 7](#_Toc422132491)

[CDR Parser 7](#_Toc422132492)

[CDR Transformer 7](#_Toc422132493)

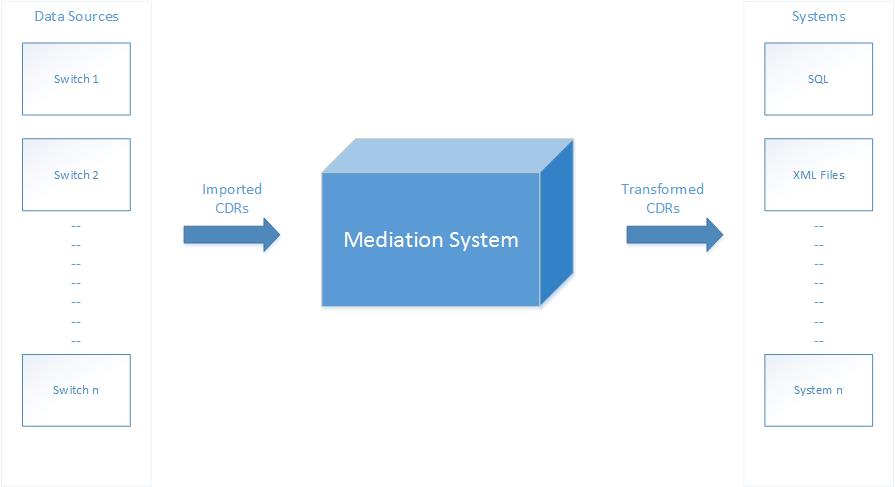
[Send Adapter 7](#_Toc422132494)

# Document Purpose

The purpose of this document is to put a technical architecture of the new Mediation System.

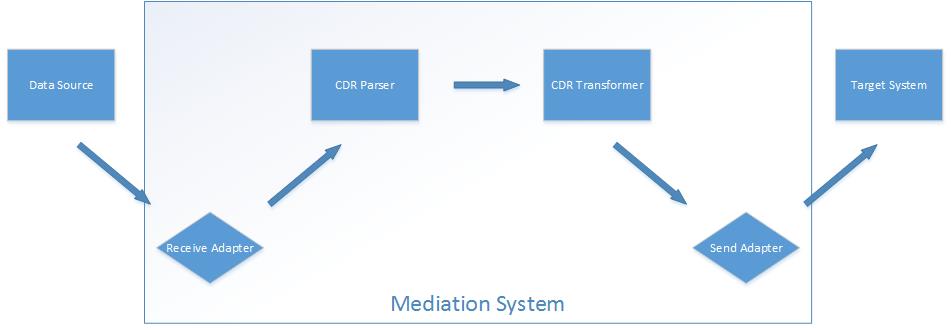
# System Overview

The Mediation System is a middleware system that would connect to one or multiple data sources, collect CDRs, transform them to specific format, and deliver them to one or more systems (e.g. databases, files…).



## Data Flow

The data flow in the Mediation system is depicted in the following diagram:



|  |  |
| --- | --- |
| Part | Description |
| Data Source | The switch/gateway to import CDRs from |
| Receive Adapter | This component shall be responsible of importing/receiving CDRs from the data source(s) and submitting them to the Mediation database |
| CDR Parser | This component shall be responsible of parsing the imported/received CDRs and converting them to system-readable format. It shall be also responsible of aggregating related CDR records and producing the final meaningful CDR record (in case of long open session or call for example) |
| CDR Transformer | This component shall be responsible of transforming the CDRs to the desired format before delivering it to the end system(s) |
| Send Adapter | This component shall be responsible of collecting ready CDRs from the Mediation database and send them to the end system |
| Target System | The end system which would eventually receive imported CDRs |

## Technical Aspects

The system shall have the following technical aspects:

|  |  |
| --- | --- |
| Feature | Description |
| Dynamic | The system data flow shall be configuration-driven starting from the connection to the data source(s), manipulating the imported data, until delivering the data to the target system(s) |
| Extensible | The system shall deal with abstract definitions of almost all parts of the data flow (e.g. receive adapters, CDR parsers, CDR Mappers, send adapters…). And it shall be possible to build and implement parts in a flexible manner |
| Performance | The system shall be highly performing especially in term of data flow speed |
| Data Validation and Accuracy | The system shall comply with all required data validation and ensure data accuracy and integrity |
| Control and Visualization | The system shall give a set of visual tools to visualize the system business processes and to take actions |
| Troubleshooting | The system shall implement logging and tracing in an efficient way and shall give visual tools in order to maintain the system and troubleshoot any issues. In addition it shall generate alerts and notifications in case of errors and warnings |
| Security | The system shall ensure secure access to all system modules and data |

The below sections in this document will give a detailed technical design for the Mediation system

# Data Flow Stages

As discussed in the above Data Flow section, the Mediation system consists of main data flow where CDRs are imported and processed by multiple parts:

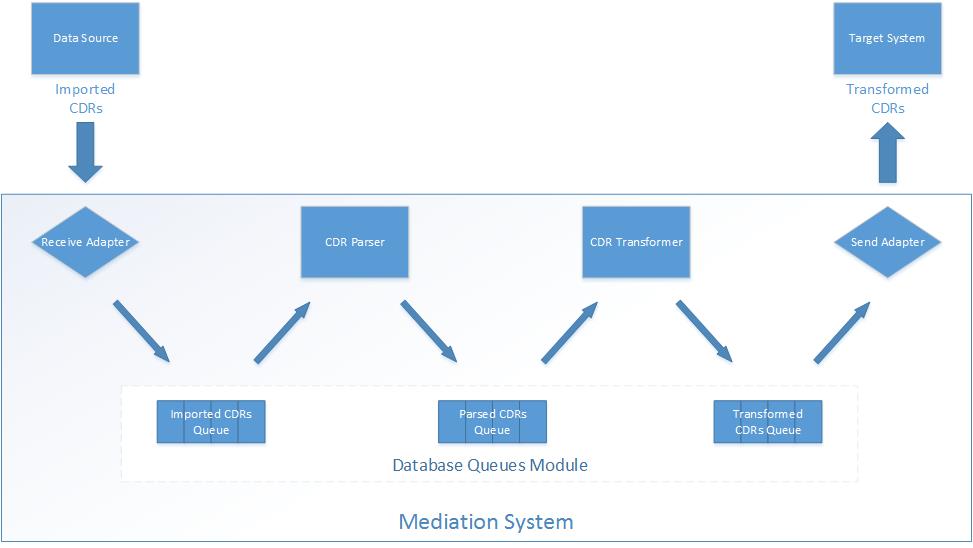
* Receive Adapter
* CDR Transformer
* CDR Mapper
* Send Adapter

Each one of these parts is considered as a separate and independent technical component which job is to take a batch of CDRs, do whatever it is configured to do on the batch, and pass the batch to the next stage. From here on, we will call each part as Processing Part

In order to implement such CDR staging, the Mediation system will include a database Queues module which shall be responsible of:

* Enable persistent storage of CDRs received from one processing part
* Deliver the CDRs to the next processing part

By following this approach, we would have each database queue representing a specific stage of the CDRs



Until now we can identify the followings within the Mediation system:

* 3 stages:
  + **Imported CDRs stage**: represented by “Imported CDRs Queue”
  + **Parsed CDRs stage**: represented by “Parsed CDRs Queue”
  + **Transformed CDRs stage**: represented by “Transformed CDRs Queue”
* 4 processing parts:
  + **Importing/Receiving CDRs**: represented by the “Receive Adapter”
  + **Parsing CDRs**: represented by the “CDR Parser”
  + **Transforming CDRs**: represented by the “CDR Transformer”
  + **Sending CDRs**: represented by the “Send Adapter”

The system shall allow:

* Creating as many instances as needed of the CDR stages (i.e. database Queues)
* Dynamically configuring as many instances as needed of the processing parts

# Processing Parts

The following sections discusses the technical characteristics of each processing part. This consists of:

* Receive Adapter
* CDR Parser
* CDR Transformer
* Send Adapter

## Receive Adapter

The Receive adapter is the part responsible of importing/receiving CDRs from external data sources and submitting them to the Mediation system. It is the first step in the Mediation data flow. The job of the receive adapter is summarized as follow:

* Connect to the data source (configured on the adapter level)
* Retrieve CDRs in batches
* Submit each CDR batch to the instance of “Imported CDRs Queue” (configured on the adapter level)

The Receive Adapter doesn’t do any CDR manipulation. It submits the CDRs to the Queue in the same format they are received.

## CDR Parser

## CDR Transformer

## Send Adapter