Client Data Integration

Version: 2.0

Date: 2nd July 2021

CONFIDENTIAL

Table of Contents

Overview

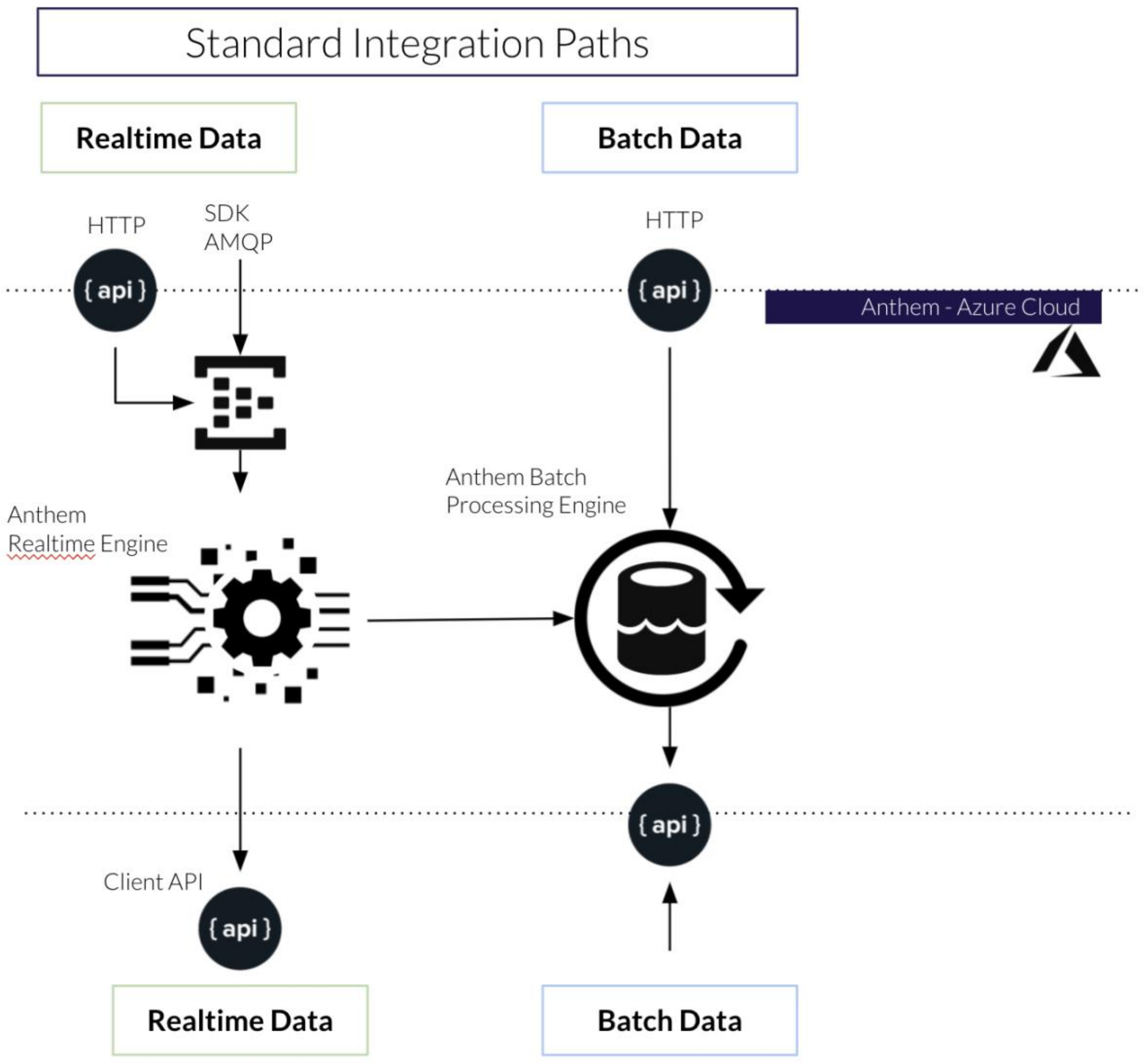
This document summarises the options available to integrate with Future Anthem, for both data ingress and egress.

Future Anthem offers Standard as well as Client Bespoke integration options for either Real-time or Batch Data.

This document also outlines the options available to provide large datasets to Future Anthem on a one-off or irregular basis.

Standard Integrations

The standard Integration of Future Anthem products with client systems is via API's:



Real-time Data Integration

Future Anthem's Standard Real-time Data Integration uses Azure Event Hubs, which provide clients with a widely supported standard technology to integrate with Anthem's Real-time Engine.

Sending Events to Anthem

Azure Event Hubs supports sending event data via:

1. REST API
2. AMQP 1.0
3. SDK for many different programming languages

Sending Events via REST API

Request

Method	Request URI
POST	https://{**clientEndpointId**}-fane01.servicebus.windows.net/{**eventTypeName**}/messages?timeout=60&api-version=2014-01

Parameter	Example	Description
clientEndpointId	casinomagic	Future Anthem provides the **clientEndpointId** for the integration.
eventTypeName	spins	The client can set the **eventTypeName** to match the client's terminology

Request Headers

Content-Type	application/vnd.microsoft.servicebus.json
Authorization	Future Anthem provided SharedAccessSignature token
x-ms-retrypolicy	NoRetry

Request Body

JSON formatted event data. Either a single event or a list of batched events.
All JSON payloads have to be String escaped and passed as the value to the “Body” key

JSON Schema

```
{
   "$schema": "http://json-schema.org/draft-07/schema",
   "$id": "http://example.com/example.json",
   "type": "array",
   "title": "Real-time Event Upload",
   "description": "Schema definition for real-time events uploaded to Future Anthem",
   "default": [],
   "examples": [
      [
         {
            "Body": "{\"GameId\":\"1\",\"PlayerId\":\"123\",\"SpinId\": 1,\"Amount\": 0.25}"
         }
      ]
   ],
   "additionalItems": true,
   "items": {
      "$id": "#/items",
      "anyOf": [
         {
            "$id": "#/items/anyOf/0",
            "type": "object",
            "title": "Event Schema",
            "description": "The schema of a single event",
            "default": {},
            "examples": [
               {
                  "Body": "{\"GameId\":\"1\",\"PlayerId\":\"123\",\"SpinId\": 1,\"Amount\": 0.25}"
               }
            ],
            "required": [
               "Body"
            ],
            "properties": {
               "Body": {
                  "$id": "#/items/anyOf/0/properties/Body",
                  "type": "string",
                  "title": "Event Body Schema",
                  "description": "The event Payload",
                  "default": "",
                  "examples": [
                     "{\"GameId\":\"1\",\"PlayerId\":\"123\",\"SpinId\": 1,\"Amount\": 0.25}"
                  ]
               }
            },
            "additionalProperties": true
         }
      ]
   }
}
```

JSON (single event)	JSON (batch events)
[{“Body”:”{\"Key1\":\"Value1\"}”}]	[{“Body”:”{\"Key1\":\"Value1\"}”}, {“Body”:”{\"Key1\":\"Value2\"}”}]

- Future Anthem will translate the client's data schema to the internal data schema used by Anthem’s Real-time Engine
- The event data provided can follow any schema and can be nested
- Clients should provide their schema documentation, allowing Anthem to configure their internal schema transformations

Response

The response includes:

- an HTTP status code
- a set of response headers
- a response body

- If the request is successful, the response body is empty
- If the request isn't successful, the body contains an error code and error message

Response Code	Description
201	Created
401	Authorization failure
500	Internal Error

Example

```
POST /spins/messages?timeout=60&api-version=2014-01 HTTP/1.1
Host: https://casinomagic-fane01.servicebus.windows.net
Authorization: SharedAccessSignature sr=casinomagic-
fane01.servicebus.windows.net&sig=casinomagic-token&se=1657122931&skn=upload
Content-Type: application/vnd.microsoft.servicebus.json

[{"Body": "{\"GameId\":\"1\",\"PlayerId\":\"123\",\"SpinId\": 1,\"Amount\": 0.25}"}]
```

```
curl --location --request POST 'https://casinomagic-
fane01.servicebus.windows.net/spins/messages?timeout=60&api-version=2014-01' \
--header 'Content-Type: application/vnd.microsoft.servicebus.json' \
--header 'Authorization: SharedAccessSignature sr=https%3A%2F%2Fcasinomagic-
fane01.servicebus.windows.net%2Fspins&sig=casinomagic-token&se=1657122931&skn=upload' \
--data-raw '[{"Body": "{\"GameId\":\"1\",\"PlayerId\":\"123\",\"SpinId\": 1,\"Amount\": 0.25}"}]'
```

Sending Events via AMQP 1.0

AMQP integration is documented as part of Azure's Service Bus:
https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-amqp-overview

Sending Events via SDK

The Azure SDK exists for all major programming languages:

- Python: https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-python-get-started-send
- .NET
  https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-dotnet-standard-getstarted-send
- Java
  https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-java-get-started-send
- JS
  https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-node-get-started-send
- Others
  https://docs.microsoft.com/en-us/azure/event-hubs/

Receiving Events from Anthem

- To receive real-time events, an API should be provided for Anthem to call
- Anthem integrates with the Client API's based on their provided specifications and requirements

Anthem supports:

- POST method to APIs
- Static IPs for IP whitelisting
- SSL with self-signed client certificates

If clients do not have an API available we offer bespoke integrations, such as hosting a webhook - for examples see the section on bespoke integration options.

Batch Data Integration

Irregular

For provisions of large datasets (e.g. an initial set of historic data), Anthem's standard pattern is to enable client's to upload via Azure Storage Explorer {GUI}.

Other options are:

- AzCopy
- Pulling data from client environment (e.g. from S3 bucket)
- SFTP can be made available on request

Regular

Regular data provision is handled via API.

Upload

Request

Method	Request URI
POST	https://{**clientEndpointId**}-fane01.azurewebsites.net/api/upload/{**date**}/{**filename**}?code={**token**}

Parameter	Example	Description
clientEndpointId	casinomagic	Future Anthem provides the **clientEndpointId** for the integration.
date	2021-01-01	The corresponding date for the uploaded data in YYYY-MM-DD format
filename	spins_01.csv	The target filename of the upload. Reusing the filename allows data to be overwritten
token	Abc123Z==	Future Anthem provides the required API key for Authorization

Request Headers

Content-Type	multipart/form-data

Body

Binary multipart form data

Response

Response Code	Description
200	Success
401	Authorization failure
500	Internal Error

Example

```
POST /api/upload/2021-01-01/spins_01.csv?code=ABC123Z== HTTP/1.1
Host: https://casinomagic-fane01.azurewebsites.net
Content-Length: 198
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

----WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name=""; filename="spins_01.csv"
Content-Type: text/csv

(data)
----WebKitFormBoundary7MA4YWxkTrZu0gW
```

```
curl --location --request POST 'https://casinomagic-fane01.azurewebsites.net/api/upload/2021-01-
01/spins01.csv?code=ABC123Z==' \
--form '=@"/path/to/file/spins-048046c6b0cf-c000.csv"'
```

Download

Request

Method	Request URI
GET	https://{**clientEndpointId**}-fane01.azurewebsites.net/api/download/{**date**}/{**templateId**}?code={**token**}

Parameter	Example	Description
clientEndpointId	casinomagic	Future Anthem provides the **clientEndpointId** for the integration.
Date	2021-01-01	The corresponding date for the uploaded data in YYYY-MM-DD format
templateId	0	The template ID specifies the clients configured output format
token	Abc123Z==	Future Anthem provides the required API key for Authorization

Body

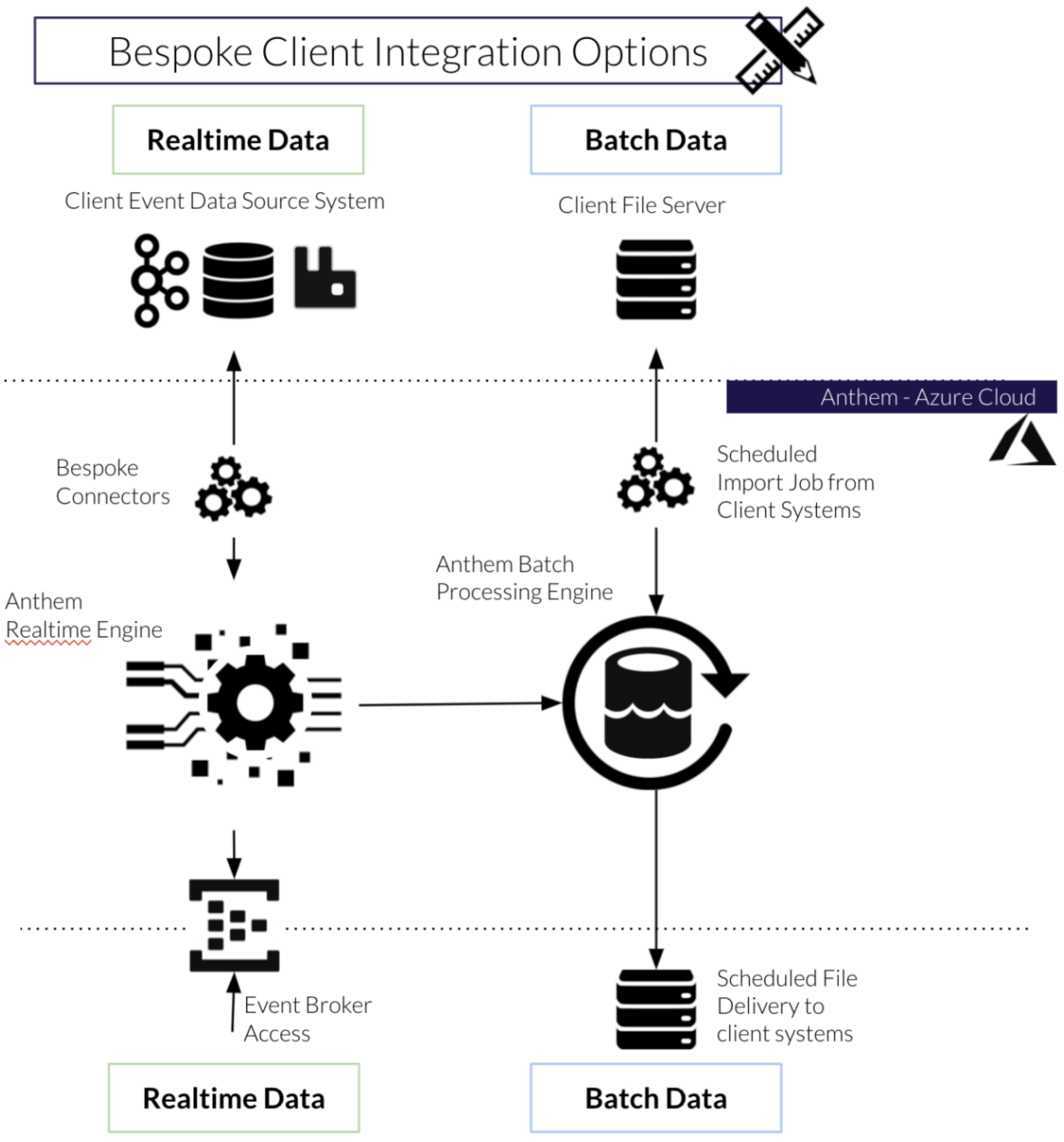
Returned file based on the configured templateId which includes:

- Client’s desired naming convention
- Date and time formatting
- File types such as csv, json
- Client specific business logic

Bespoke Integrations

Where the standard integration options are not suitable, Anthem will deliver a bespoke integration based on client requirements.

Future Anthem has built a range of successful client integrations which required bespoke solutions.


Bespoke Client Integration Options
Realtime Data
Batch Data
Client Event Data Source System
Client File Server
Anthem - Azure Cloud
Bespoke Connectors
Scheduled Import Job from Client Systems
Anthem Batch Processing Engine
Anthem Realtime Engine
Event Broker Access
Scheduled File Delivery to client systems
Realtime Data
Batch Data

Real-time Data

Sending Events to Anthem

Future Anthem can ingest real-time data directly from client systems via a bespoke connection.

Future Anthem has successfully integrated clients via a direct connection to:

- Client Kafka brokers to consume real-time feeds from Kafka
- Client RabbitMQ brokers
- Various databases and DWH implementations to consume real-time feeds via change data capture. Examples include Postgres and Snowflake

Anthem's bespoke connectors support:

- Static IPs for IP whitelisting
- SSL with self-signed certificates
- Authentication via username and password or via certificates

Receive Events from Anthem

- Clients can receive real time data from Future Anthem's real-time engine without providing an API
- Clients can connect to Future Anthem's systems to poll data directly via a bespoke connection

Future Anthem supports:

- Hosting a client webhook
- Providing a message broker to connect to e.g. Kafka or AMQP
- Static IPs for IP whitelisting
- Authentication via username and password or certificates

Batch Data

Providing files to Anthem

Future Anthem can ingest files with batch data directly from client file servers or blob storage, and can also host a file server or blob storage on a client's behalf.

Future Anthem has successfully integrated clients via:

- SFTP
- Blob storage in Azure and AWS S3
- Ingestion on time schedules
- Ingestion triggered by notifications

Future Anthem can deliver files directly to client systems using a wide range of bespoke integration options.

Future Anthem supports:

- SFTP
- Blob Storage in all major cloud providers such as Azure and AWS S3
- Uploading data via client hosted APIs