CHUBB DIGITAL EMBEDDED

Integration overview

Abstract

This document provides an overview for partners
on how to integrate with Chubb Digital platforms and services

This document is part of a series which provide useful information
to our partners in terms of our capabilities and how to make use of them.

Document Version Control

Revision History

Revision Date	Version	Updated by	Summary of Changes
28/09/2022	1.0	Le Roi Beukes leroi.beukes@chubb.com	Created Document
25/10/2022	1.1	Le Roi Beukes Leroi.beukes@chubb.com	Updated with more information on the authentication flow Updated with partner on-boarding overview

CHUBB®

Chubb provides various ways for partners to integrate, our main integration channel is via the use of API's.

Chubb's api endpoints are exposed and hosted in the cloud – we make us of an API Management Gateway to provide our partners with a secure and stable platform.

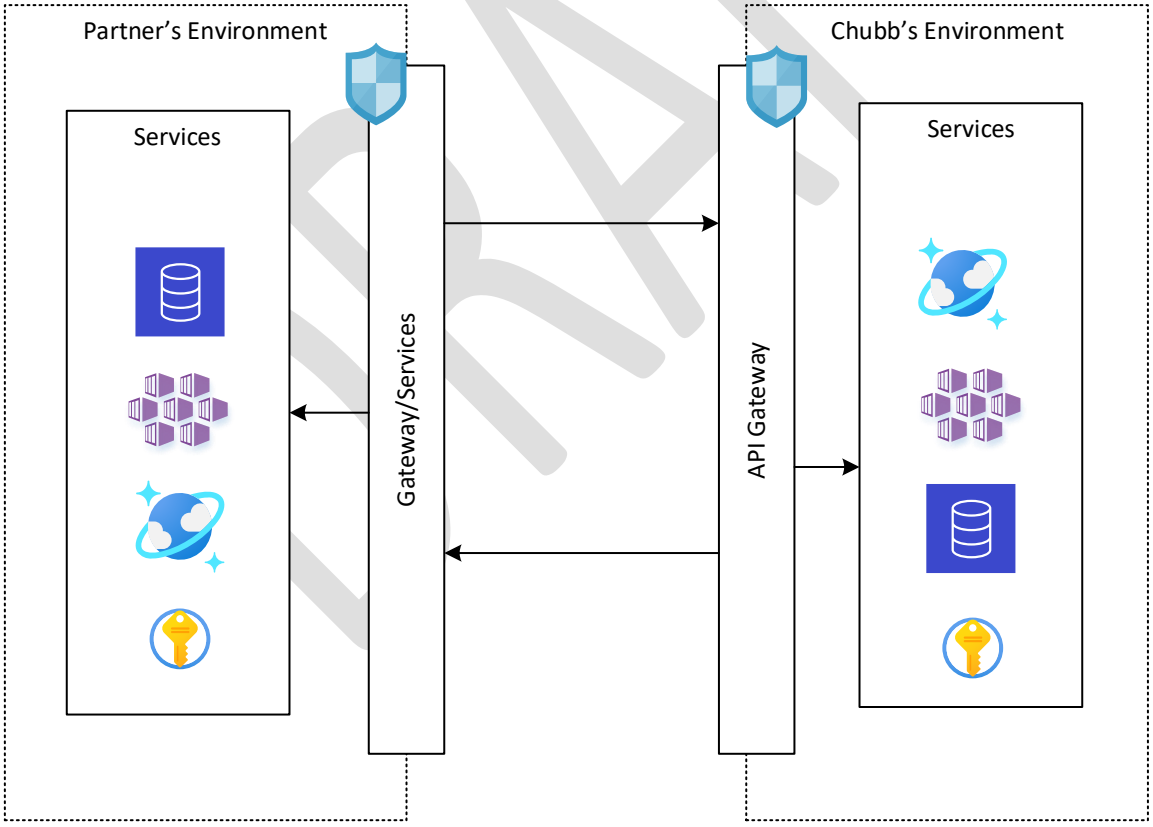
Partner development teams are required to be on-boarded to our platform before they can start making use of our services.

On-boarding is a quick and easy process provided via a self-service portal, should the team not be able to make use of the self-service portal , Chubb partner teams would be able to assist to do it on their behalf.

The portal, respective API endpoints and backing services are fully cloud native and supported by not just our own dedicated cloud and network engineering teams , but also by our designated cloud service provider.

Chubb is committed to ensuring high availablity and scability targets for our partners.

High Level Integration Architecture



CHUBB®

Environments

- Development (DEV)
- User Acceptance Test (UAT)
- Production (PROD)

Chubb provides multiple environments to support the software development life-cyle.

Each environment has its own dedicated instance of the api gateway and backing services.

NB:

Endpoints might differ slightly between the environments but these are a rarity and would be communicated early in the process if it applies.

Gateway endpoints:

Development	https://apacsit.chubbdigital.com
UAT	https://apacuat.chubbdigital.com
Production	https://apac.chubbdigital.com

API Studio Portal :

https://developer.chubb.com/

Developer Sign-on & Subscription

The APIM provides a single sign-on (SSO) for partner developers to start making use of our services.

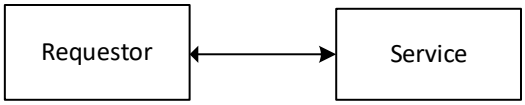
Processing Modes

Chubb supports three main modes of processing requests:

- Synchronous
- Asynchronous
- Asynchronous with callback

Synchronous

This is the standard request/response mode of processing where a request is made to an endpoint and the requestor waits for a response



Some endpoints utilise only this mode and in those cases the services are turned to respond in an agreed time (pursuant to any SLA) – generally speaking Chubb aims to provide very fast response times for these services.

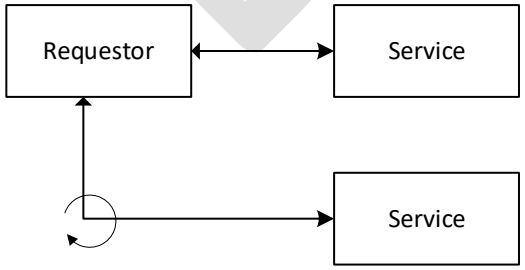
NB – Optional auto switching to asynchronous mode:

For some endpoints that may start out as using a synchronous processing mode , an optional time window could be provided as part of the request – where upon reached it would automatically return and switch to an asynchronous processing mode should a result not yet be available.

Asynchronous

This is a more modern processing approach where a request is made to an endpoint and an identifier is returned to the requestor together with an indication of where to check on the status of the request.

The requestor then periodically 'polls' another service to check on the status of the request and/or retrieve the result.



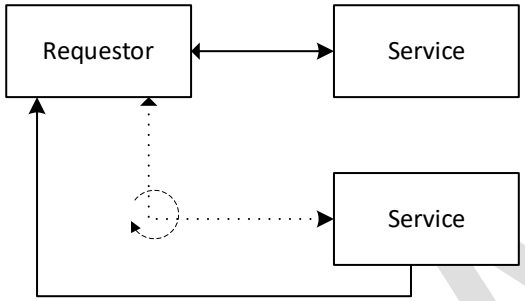
This mode can be specified on certain endpoints and am recommended for transactional requests.

Asynchronous with callback

This is a variation of the Asynchronous processing approach where a request is made to an endpoint and as part of the request an endpoint is provided where the requestor can be notified that a request has been completed.

In addition to this an identifier is also returned to the requestor together with an indication of where to check on the status of the request should they want to check before being notified.

The requestor will be 'called' when the request has been completed and provided with either the result or an indication of where to fetch the result from.



This mode can be specified on certain endpoints and am recommended for high volume transactional requests.

Schemas

Chubb endpoints support a variety of schema's out of the box, from minimal to full-fledged – we always aim to make the developer burden less for our partners by not requiring unnecessary fields or information in our schema's.

We manage all of our schemas by way of a centralized schema registry where all schema's are stored and a governance process is in place to ensure consistency and/or updates.

All schema's are versioned and indicated as part of the endpoint request, and whilst we provide a fairly generic list of schema's, we also have the capability to support BYOS (bring your own schema) – if for whatever reason the partner cannot make use of our existing schema's or they are not extensible enough to accommodate them or a partner is unable to consume our schema , we support the importing / definition of the partner's schema.

These would be on a case-by-case basis and form part of our schema registry going forward for the particular region where it was requested.

Life Cycle

Chubb's endpoints are structured around the insurance sales and servicing life cycle.

Chubb provides everything necessary for our partners to be able to sell insurance to their customers.

Each step of the life-cycle is depicted as a category of endpoints and exposed on our gateway.

1. **Discovery**
    a. These endpoints provide a partner with enough information to know which products are available to sell to their customers, together with product details like description, benefit levels, coverage, premium, etc.
    b. Some products which are simple may even include static pricing and hence not require the usage of the Pricing category

2. **Pricing**
    a. These endpoints provide a partner with the ability to request dynamic pricing for a specific product by providing the necessary parameters (in the form of question/answer pairs and/or other input)

3. **Sale**
    a. These endpoints provide a partner with the ability to capture a sale for a customer and issue an insurance policy.
    b. Chubb has a policy of payment before coverage, thus (depending on the product), the sale endpoints may also act as an orchestration to capture payment on behalf of the partner.
    c. Depending on the overall business integration model, payment details need not be provided.

4. **Payment**
    a. These endpoints provide a partner with the ability to manage payment on behalf of their customers, in the event of credit card payment – Chub is fully PCI compliant.
    b. We provide support for the full payment life cycle (capture,void,auth,settle,enquire) by making use of our integrations with various payment gateway providers.
    c. Chubb also provides for PCI compliant mechanisms to capture a customer's account number, including providing the provider with a secure iframe to tokenise the customer's account number before sending the tokenised card number to Chubb.
    d. Chubb also provides the ability to make use of a hosted payment page to collect the customer's payment in the event that the partner is unable to do so for whatsoever reason.

5. **Fulfilment**
    a. These endpoints provide a partner with ability to request and or instruct Chubb to send out fulfilment requests such as policy documents.

6. **Servicing**
    a. These endpoints provides a partner with the ability to facilitate customer and policy servicing requests such updating of personal information, retrieving a list of policies and performing searches on customers etc.

7. **Claim**
    a. These endpoints provides a partner with the ability to facilitate the lodging of first notification of loss (FNOL) for their customers and to also make enquiries on the status of claims already lodged.

On-boarding and the road to production

New partner on-boarding process

The following process is followed when on-boarding a partner:

1. NDA's signed
2. Third party cyber risk assessment
3. Third party cloud governance risk assessment
4. Build permit review

Once done, the provisioning of credentials would be enabled and a partner may start their development/integration testing.

Partner engagement process

Once the on-boarding process has been completed , then the engagement with the delivery team starts.

1. Introduction meetings
2. BA draws up requirements
3. Tech teams discuss collaboration methods (email / IM etc.)
4. Tech teams have initial meeting to discuss expectations and refine requirements
5. Development starts at both ends
6. Integration testing is completed
7. UAT test cases are drawn up and agreed upon

Road to production

1. UAT Testing
2. UAT Sign-off received
3. CAB Approval
4. Deployment prep
5. Production go-live
6. Post-production tech live verification (Tech LV)
7. Post-production business live verification (Business LV)

Partner Connect

Partner connect is our online portal where partners are able to explore and interact with our embedded experiences.

Some key features :

- Documentation
- Interactive playground
- SDK’s
- Community Boards
- FAQ
- Support
- Telemetry
- Insights
- Provision of credentials

As part of the engagement process , partner connect is meant to accelerate the development cycle and assist in providing additional information as required without having to involve the tech teams.

Integration

Message transfer and structure

- All requests and responses are JSON-formatted and UTF-8 encoded.

Authentication

Authentication is by way of bearer token and would be requested before each and every call is made to an endpoint.

Bearer tokens are represented as JSON Web Token (JWT) tokens -

JSON Web Token (JWT) is a JSON-based open standard (RFC 7519) for passing claims between parties in a web application environment.

The tokens are designed to be compact, URL-safe and usable especially in the web browser single sign-on (SSO) context.

JWT claims can be typically used to pass the identity of an authenticated user between an identify provider and a service provider, or any other type of claims as required by the business process.

The tokens can also be authenticated and encrypted.

Chubb uses JWT tokens to ensure that all requests to the APIs are from authorised users and they have the necessary access to perform the requested operation.

Mime Types

An Accept header is required for all requests, for example:

Accept: application/json

A Content-Type header must be given when sending data to the API (using POST), for example:

Content-Type: application/json

Payload Security

Payload can optionally be encrypted using PGP or other form of encryption.

Should this be required a custom HTTP header needs to be included in the request – more details can be provided upon request.

Authentication

Authentication is requested by making an initial POST request to the 'token' endpoint.

As part of the request the following headers must be present and have valid values (as obtained from the portal or provided by a Chubb representative)

App_ID	Obtained from portal / provided by Chubb
App_Key	Obtained from portal / provided by Chubb
Resource	Obtained from portal / provided by Chubb
apiVersion	Obtained from portal / provided by Chubb

Typical Authentication Flow