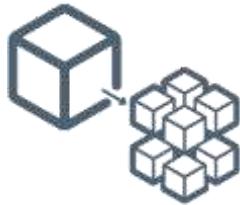


# Taller Desarrollo de Configuraciones APIC

## Entrenamiento VCSOFT

APIC 2018.4



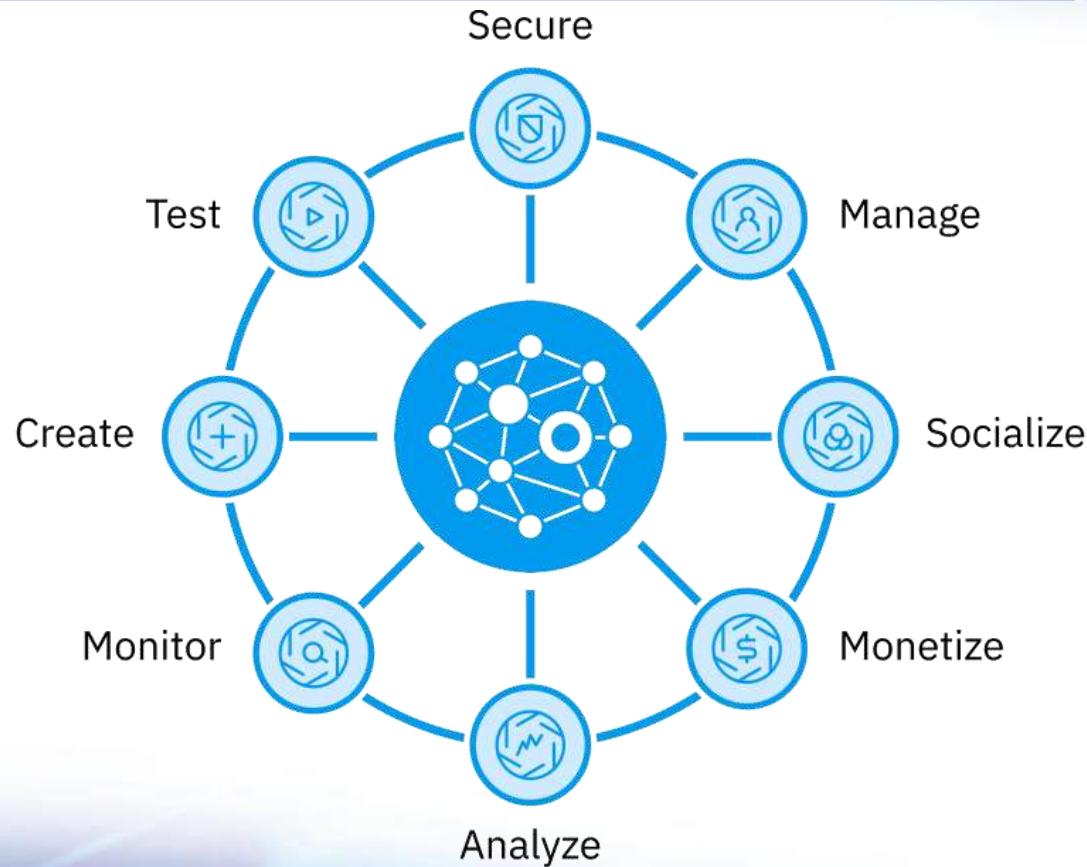
# Taller Desarrollo de Configuraciones APIC

<b>1</b>	<b>Nombre</b>	<b>Conceptos, Administración y Operación APIC</b>
2	Duración:	20 horas
3	Alcance de la Capacitación:	Registro de APIs y Productos en de la plataforma IBM API Connect 2018.
4	Requisitos que deben cumplir los participantes	<ul style="list-style-type: none"><li>▪ Conocimientos básicos de REST y servicios API</li><li>▪ Conocimiento en Contratos API con Swagger</li><li>▪ Conocimientos básicos en Javascript.</li><li>▪ Conocimientos de seguridad de plataformas y seguridad de información.</li><li>▪ Conocimiento en conceptos de monitoreo, operación y creación de scripts de control de plataformas.</li></ul>

# Taller Desarrollo de Configuraciones APIC

Temario	Desarrollo de Configuraciones APIC
Contexto	<ul style="list-style-type: none"><li>▪ Qué es APIC. Que es una plataforma API manager</li><li>▪ Componentes</li><li>▪ Comparación con otros productos</li></ul>
Flujo Desarrollo API	<ul style="list-style-type: none"><li>▪ Flujo de registro y publicación de API</li><li>▪ Actores</li></ul>
Componentes y Herramientas	
Definición de API	<ul style="list-style-type: none"><li>▪ OpenAPI / Swagger</li></ul>
Productos y API	<ul style="list-style-type: none"><li>▪ Ciclo de vida producto y API</li><li>▪ Políticas y Assembly en API</li><li>▪ Control de Plan</li></ul>
Modelo Seguridad	<ul style="list-style-type: none"><li>▪ OAuth2</li><li>▪ JOSE: JWE/JWS</li><li>▪ JWT</li></ul>

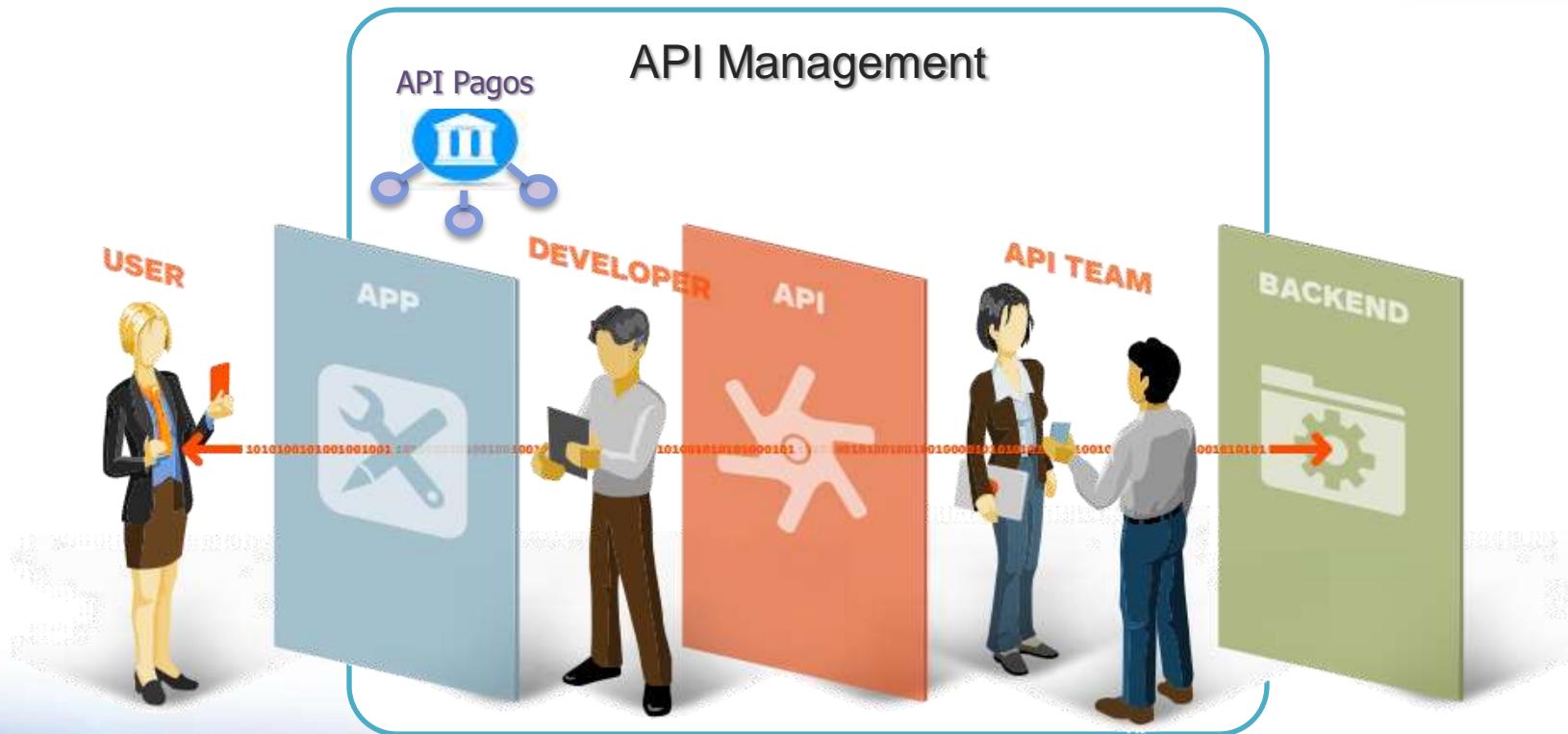
**Contexto  
APIConnect**





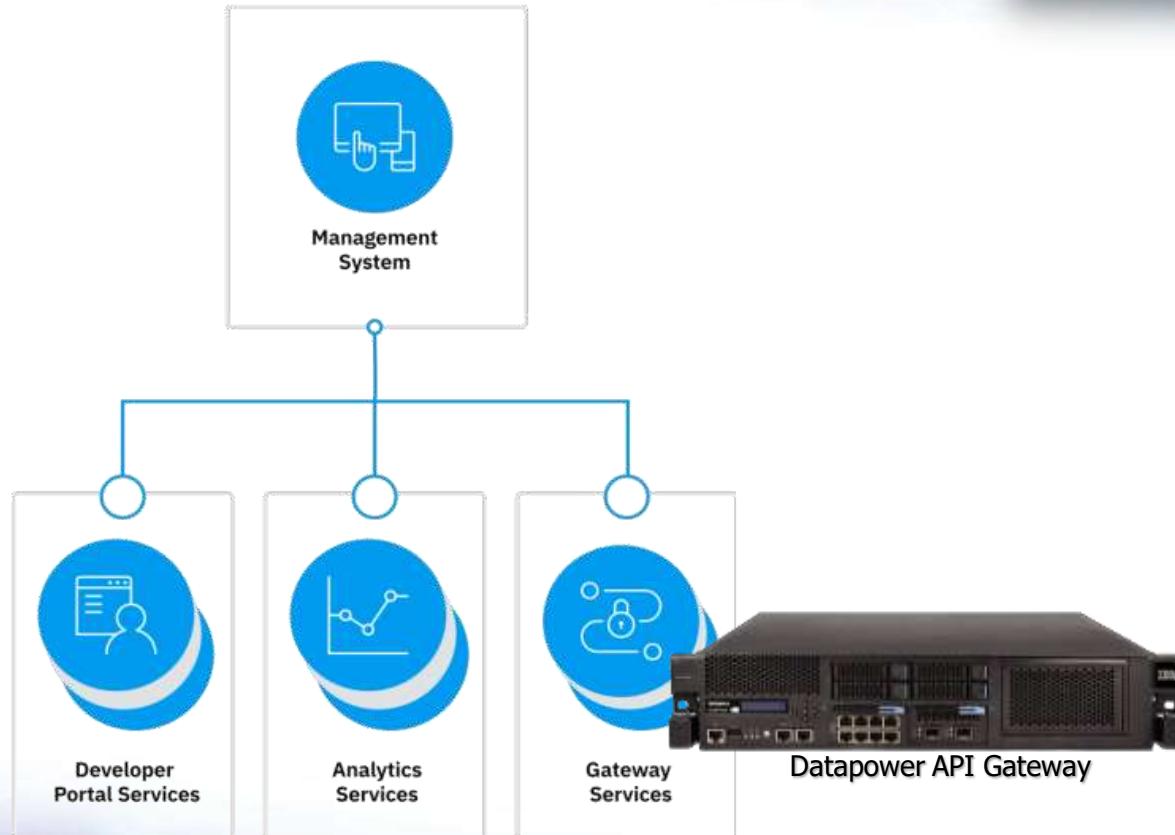


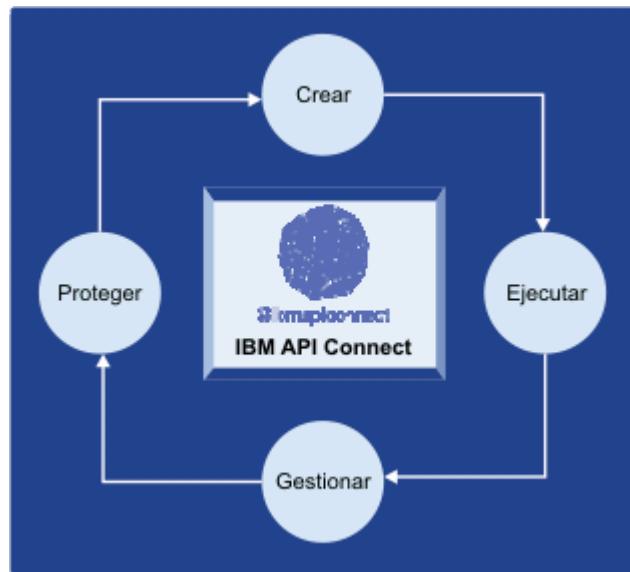
# ¿Cuál se ajusta mejor a mis necesidades reales?

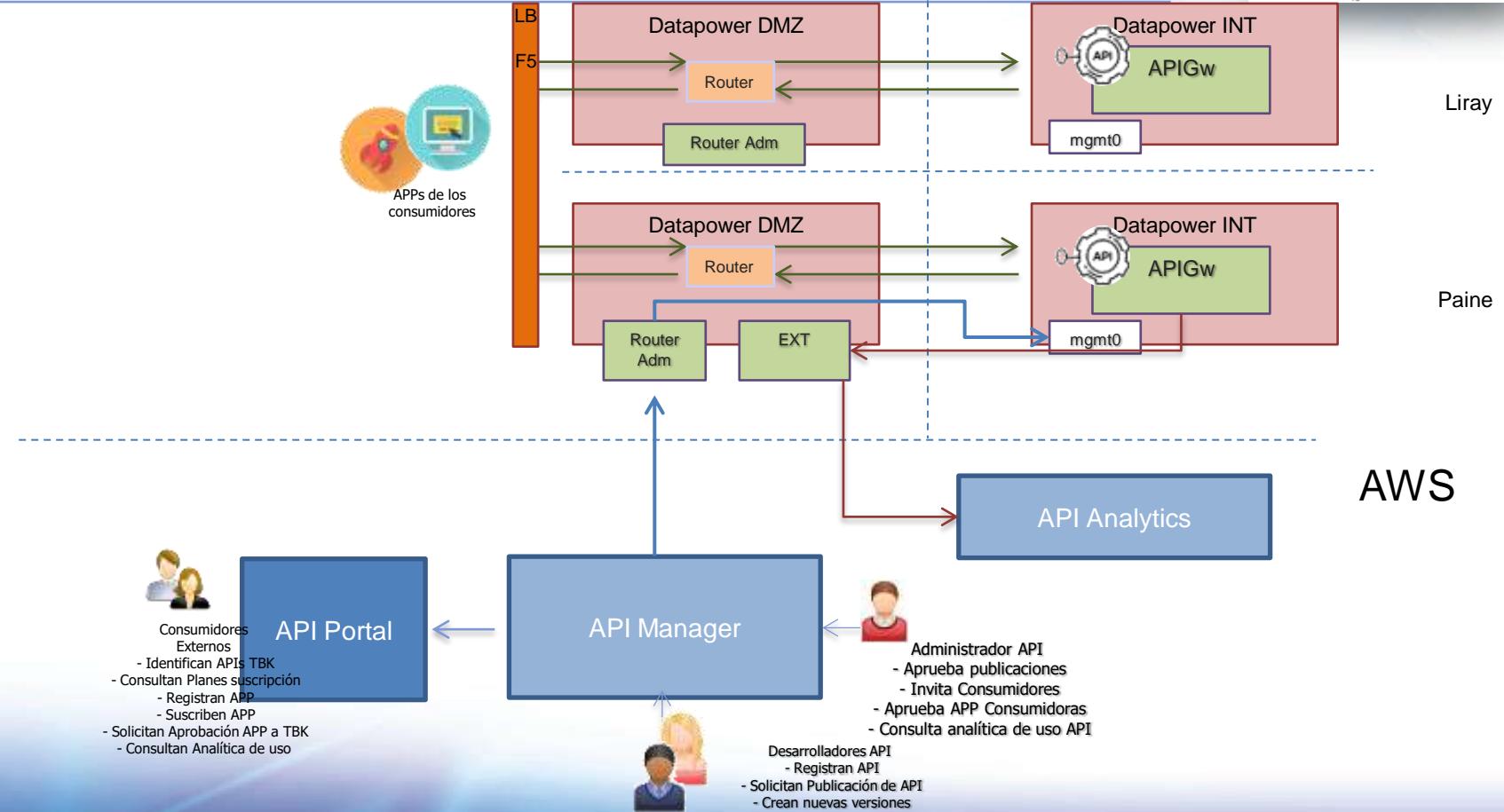


**ECOSISTEMA**

# Componentes APIC







# Cloud Manager

IBM API Connect  
**Cloud Manager**

- [Home](#)
- [Provider Organizations](#)
- [Resources](#)
- [Topology](#)
- [Members](#)
- [Settings](#)

Welcome to the Cloud Manager

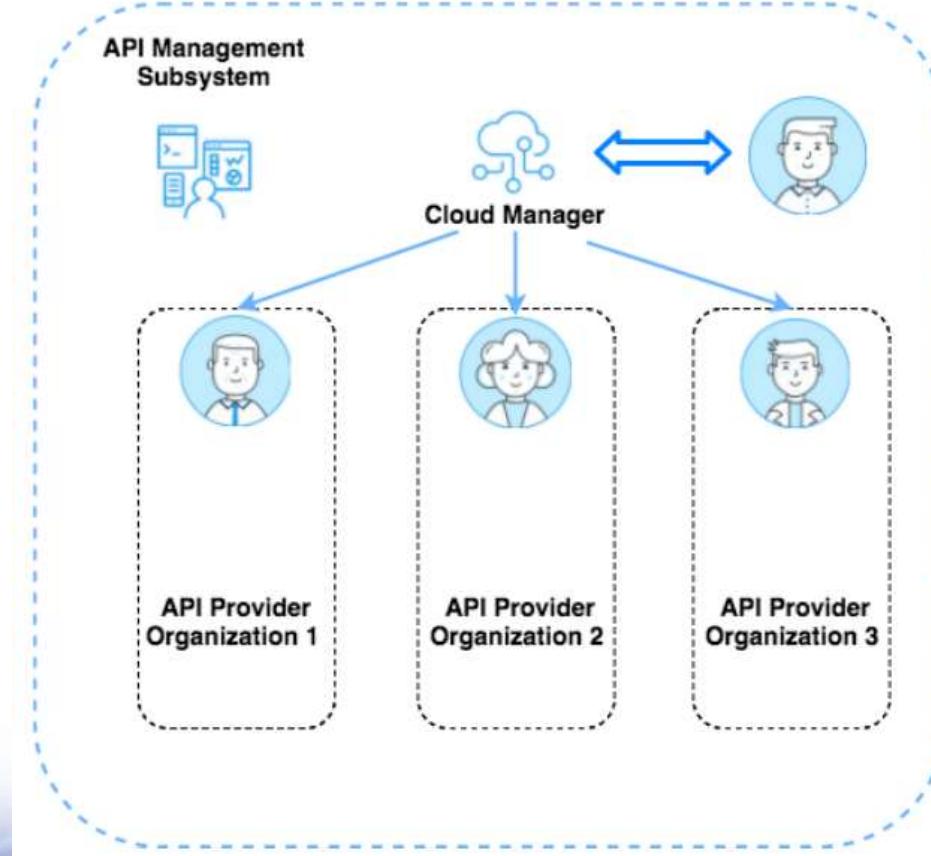
**Configure Cloud**  
Edit settings for user registries, roles, endpoints, and more.

**Configure Topology**  
Manage availability zones and services

**Manage Resources**  
Configure user registries, TLS, OAuth providers and email servers

**Manage Organizations**  
Create and manage API provider organizations and owners

# Cloud Manager



# API Manager

IBM API Connect  
**API Manager**

Welcome to the API Manager

Organization Transbank

Home Develop Manage Resources Members Settings

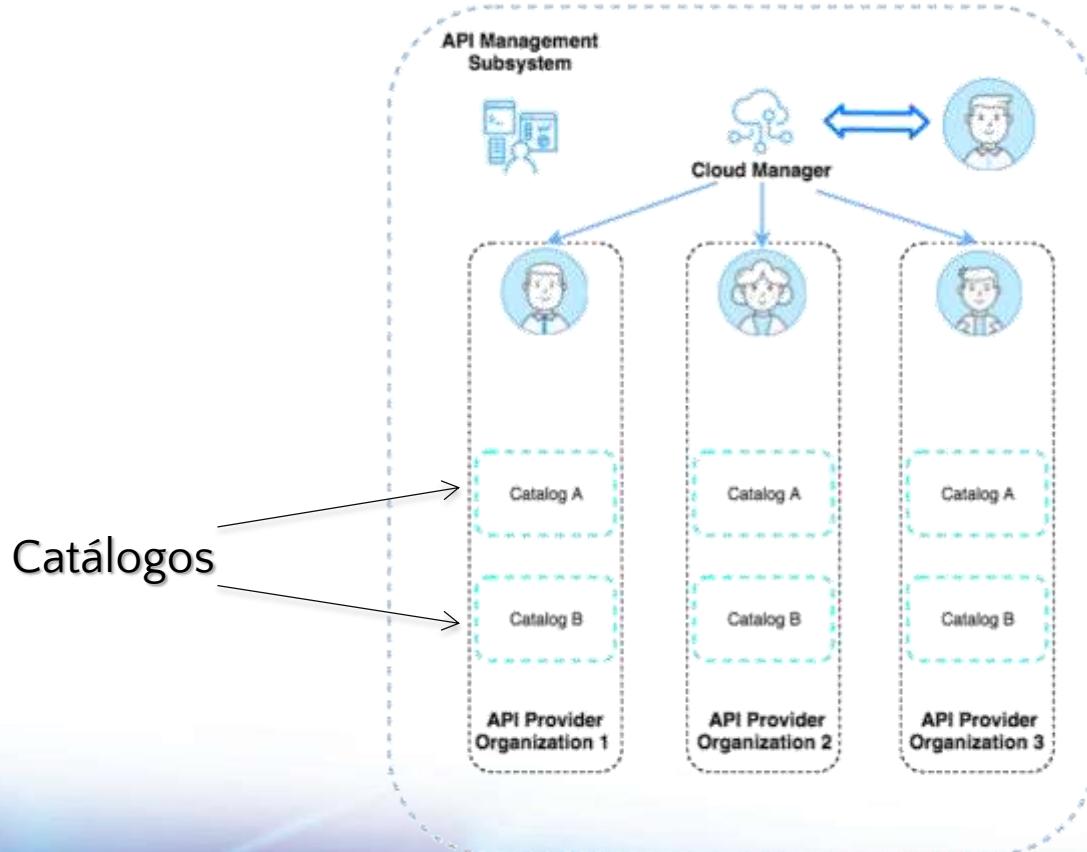
 Develop APIs and Products  
Edit, assemble, secure and test APIs.  
Package APIs using products for publishing to consumers.

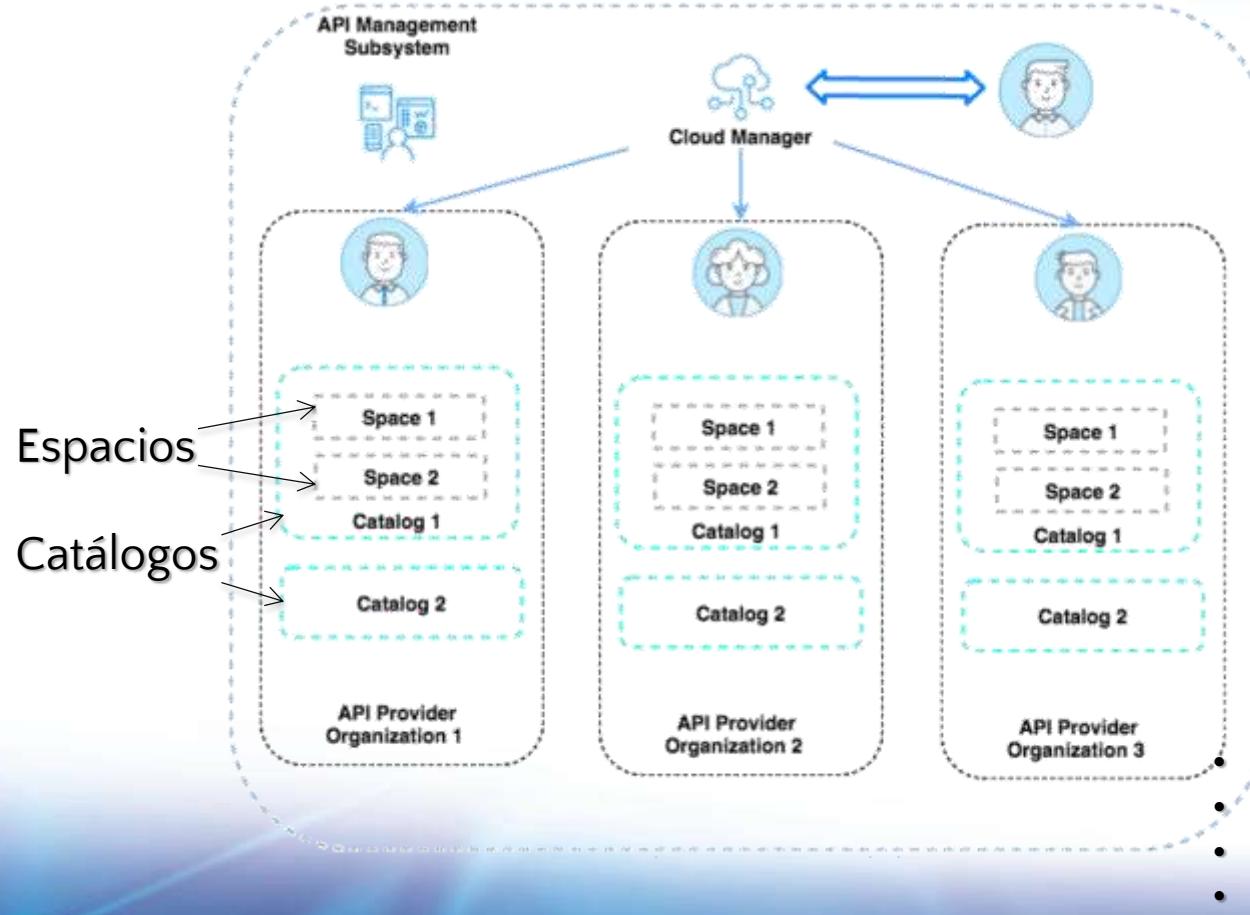
 Manage Catalogs  
Manage active APIs and consumers

 Manage Resources  
Configure user registries, OAuth providers and TLS

 Manage Settings  
Edit settings for roles, notifications and more.

# Entidades gobernadas en API Manager





# APIs & Products

## APIs and Products

### TITLE

### TYPE



Abonos-1.1.0

API (REST)



Laboratorio\_UDP-1.0.0

API (REST)



Tbk-Test-Ventas-1.0.0

API (REST)



Test-Jaime-Bernal-1.0.1

API (REST)



Test Ventas-1.0.0

API (REST)



Ventas-1.1.0

API (REST)

APIs

Productos



TestPiloto-1.0.0

Product



Ventas auto product-1.1.0

Product

# Definición API

Develop abonos 1.1.0 ▾

Design Source Assemble

Stopped No Errors Save

**API Setup**

Security Definitions

Security

Paths

Definitions

Properties

Target Services

Categories

Activity Log

**Info**  
Enter the API summary details.

**Title**  
Abonos.

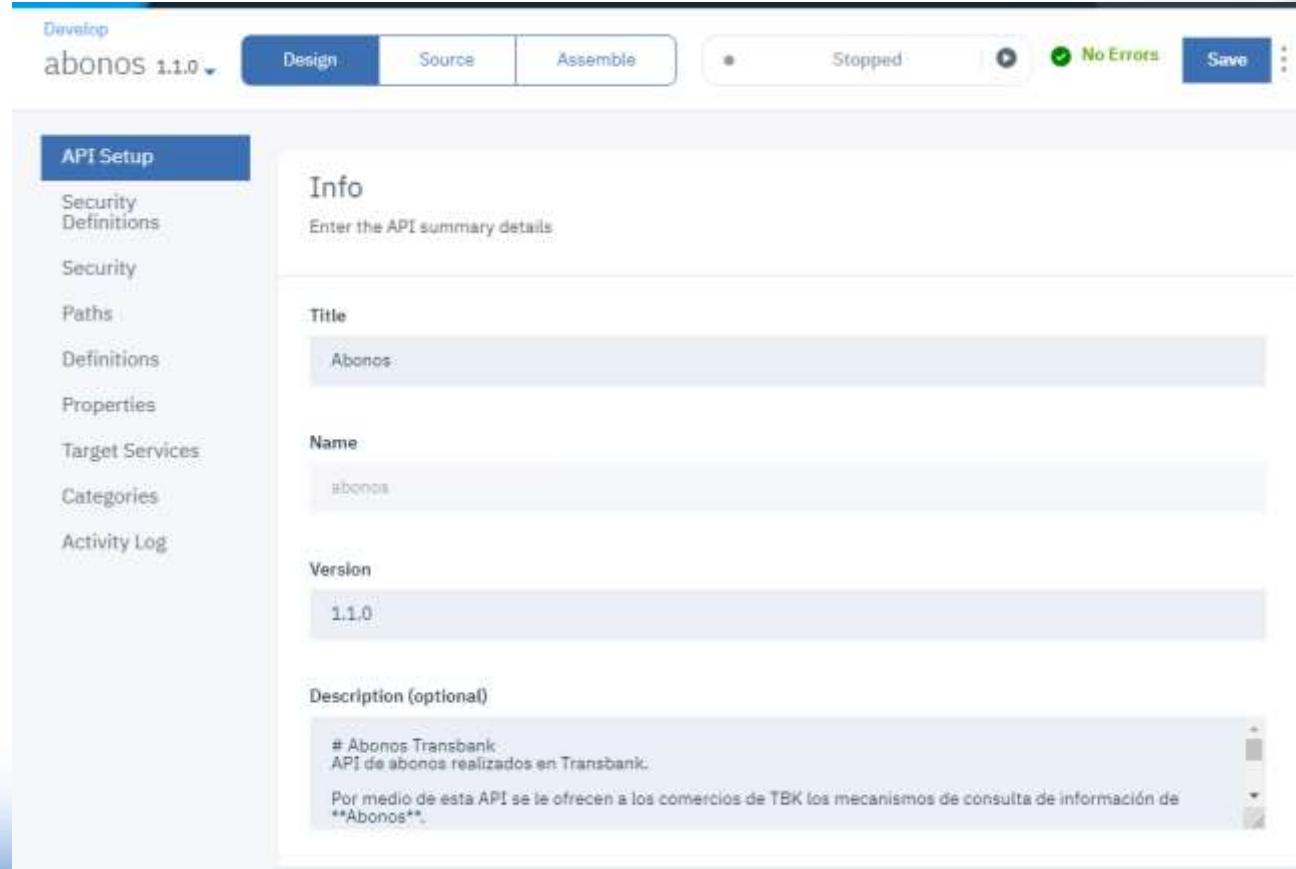
**Name**  
abonos

**Version**  
1.1.0

**Description (optional)**

# Abonos Transbank.  
API de abonos realizados en Transbank.

Por medio de esta API se le ofrecen a los comercios de TBK los mecanismos de consulta de información de \*\*Abonos\*\*.



# Definición Producto

Develop testpi... 1.0.0 ▾

Design Source Save

**Product Setup**

Visibility

APIs

Plans

Categories

**Info**

Title: TestPiloto

Name: testpilot

Version: 1.0.0

Summary (optional)

Contact

This screenshot shows a user interface for defining a product. At the top, there's a navigation bar with 'Develop' and 'testpi... 1.0.0'. Below it, tabs for 'Design' and 'Source' are visible, along with a 'Save' button. On the left, a sidebar lists 'Product Setup' (selected), 'Visibility', 'APIs', 'Plans', and 'Categories'. The main area is titled 'Info' and contains fields for 'Title' (set to 'TestPiloto'), 'Name' (set to 'testpilot'), and 'Version' (set to '1.0.0'). There's also a 'Summary (optional)' field which is currently empty. At the bottom, there's a 'Contact' section.

# API Developer Portal

IBM API Connect  
Developer Portal

API Products Apps Blogs Forums Support

Organisation: 1234567890-1 1



**Brace yourselves.  
APIs are coming.**

Explore, subscribe to and be creative with our APIs.  
We can't wait to see what you come up with!

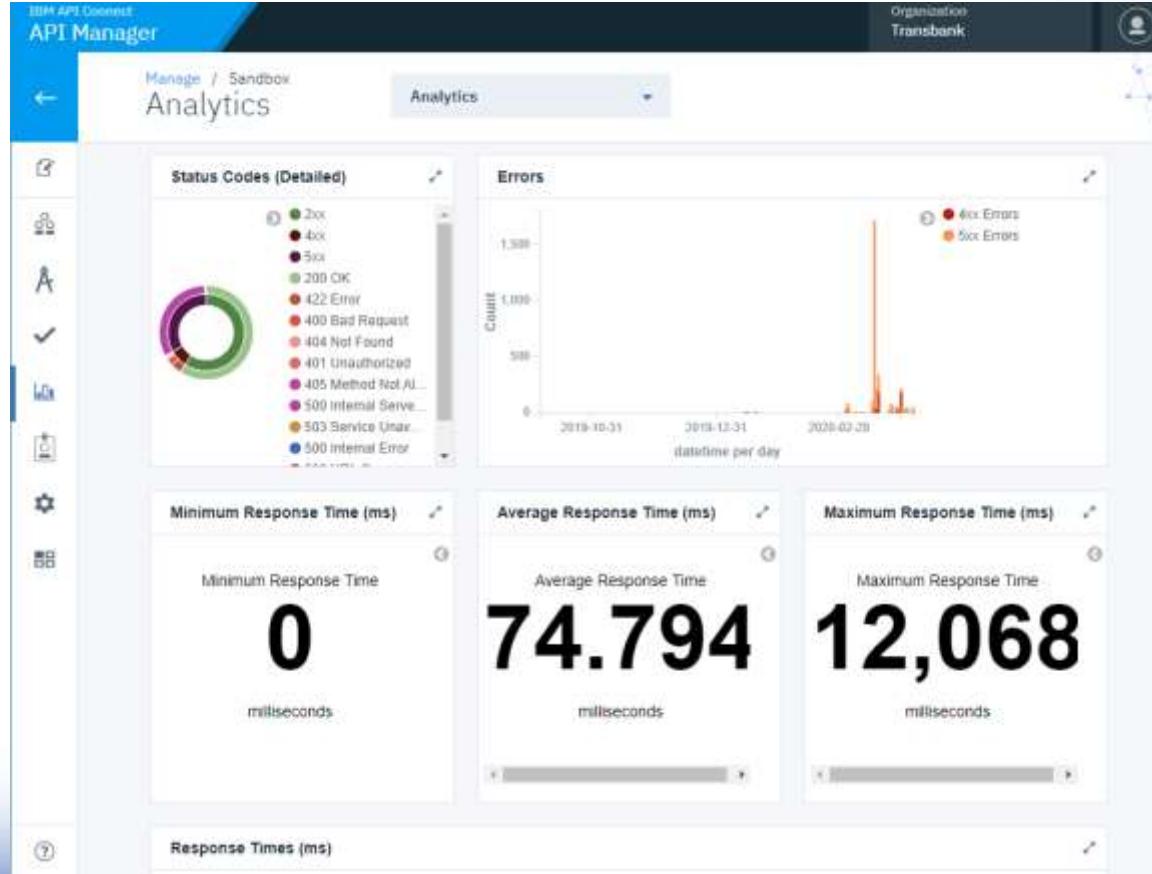
[Explore API Documentation](#)

API Products [See all products](#)



Abonos auto  
product 1.1.0

# API Analytics @ API Manager



# API Analytics @ API Developer Portal

IBM API Connected  
Developer Portal

API Products Apps Blogs Forums Support

Organization 1234567890-1 1

Applications

## APP-RUT-1234567890-1

Dashboard Subscriptions

Description

APP para pruebas de un comercio con nombre 1234567890-1

API Stats 5:16 pm - 5:17 pm

30 secs 1 min 30 mins 1 hr 1 day 7 days 30 days

Response Time

Time: 5:16:06 pm, 5:16:16 pm, 5:16:26 pm, 5:16:36 pm, 5:16:46 pm, 5:16:56 pm, 5:17:06 pm

Total Calls (Last 30 days) 451 calls

Total Errors (Last 30 days) 301 errors

Average Response Time (Last 30 days) 285.08 ms

# API Designer

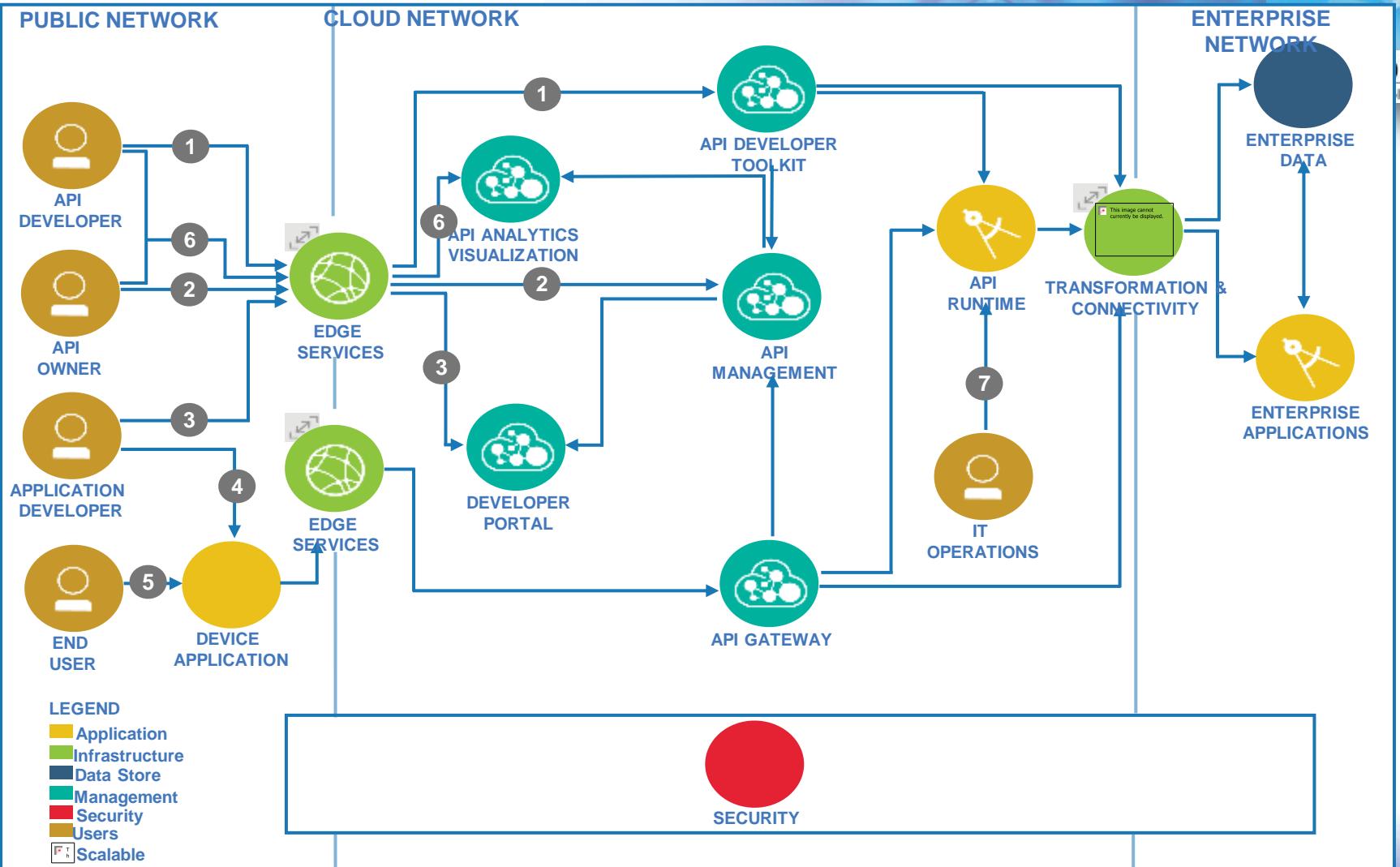
The screenshot shows the IBM API Connect API Designer interface. The top navigation bar includes 'API Connect' (with a gear icon), 'API Designer', 'Edit', and 'View'. Below the bar, the title 'IBM API Connect' and 'API Designer' are displayed, along with 'Switch Cloud Connection' and 'Organization Transbank' buttons. A user profile icon is also present.

The main area is titled 'Develop' and contains a section titled 'APIs and Products'. A blue 'Add' button is located in the top right corner of this section. The table lists the following items:

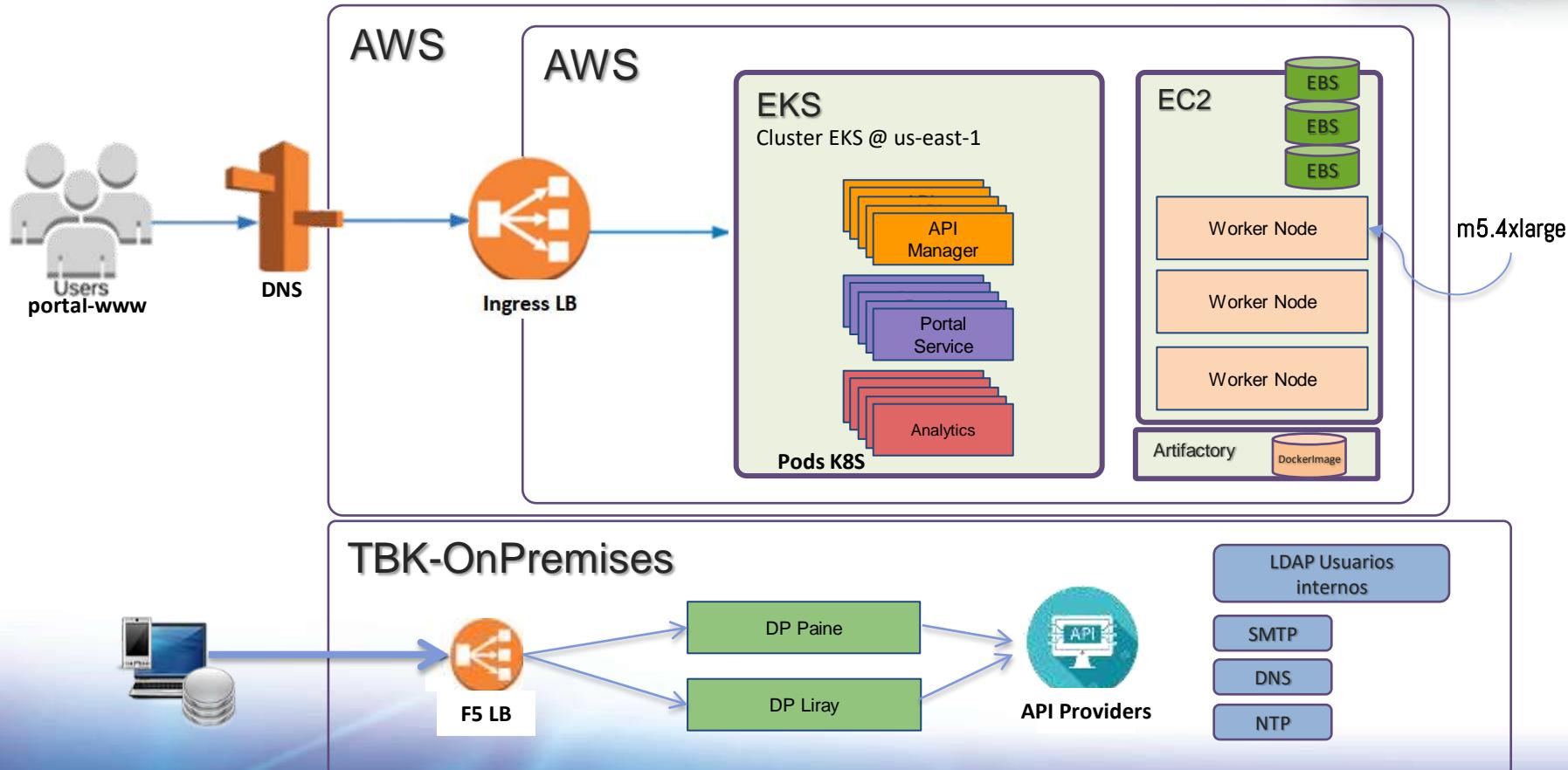
TITLE	TYPE	LAST MODIFIED
Abonos-1.1.0	API (REST)	Today at 1:00 PM
Test Ventas-1.0.0	API (REST)	Last Tuesday at 5:44 PM
Ventas-1.1.0	API (REST)	Today at 1:00 PM
Abonos auto product-1.1.0	Product	Last Thursday at 9:12 AM
Ventas auto product-1.1.0	Product	Last Wednesday at 8:14 AM

At the bottom of the table, there are pagination controls: 'Items per page' set to 10, a page indicator '1 of 1 pages', and navigation arrows for 'Previous', 'Next', and 'Last'.

Other visible elements include a sidebar on the left with icons for 'APIs', 'Products', and 'Cloud Connections', and a bottom navigation bar with a question mark icon and 'Cookie preferences' link.



# Topología alto nivel APIC



# Flujo Desarrollo API



API provider actions  
(Provider organization)

Manage and monitor the service in the cloud

Create and manage provider organization

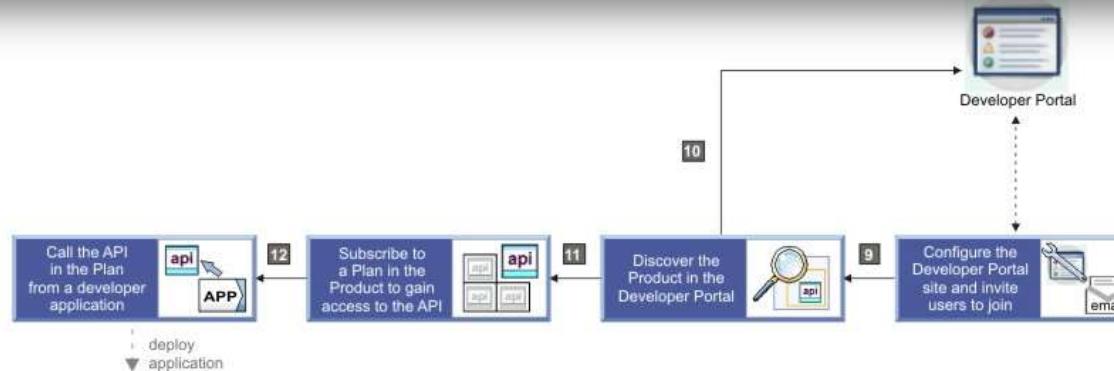
Create a draft in a provider organization

## Acción: 1

Propietario de nube

Administrador de nube

Los requisitos mínimos de una solución local de API Connect constan de un servidor de gestión que gestione y analice las API, un servidor de pasarela que dirija el tráfico del API, un servidor de análisis para analizar las API y un servidor que aloje el Portal de desarrollador. Como propietario o administrador de la nube se pueden agrupar un conjunto de servidores de gestión, de análisis, de pasarela y de Portal de desarrollador para crear *clústeres* a fin de equilibrar la carga y aislar el tráfico. Un clúster tiene una única dirección de red a través de la cual puede acceder a sus prestaciones.



API consumer actions  
(Consumer organization)



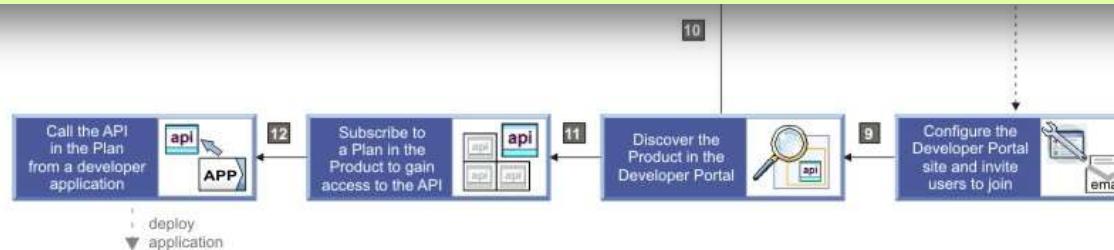
API provider actions  
(Provider organization)



## Acción: 2



Gestor de organización Propietario organización Una vez lista la infraestructura, los gestores y los propietarios de organización pueden gestionar *organizaciones* de usuarios que crean las API, las aplicaciones de proveedores y los Productos asociados. Los usuarios pertenecen a una o más organizaciones de proveedores y trabajan tanto individualmente como colectivamente en las API o en las aplicaciones que pertenecen a la organización. Los equipos de proyecto, departamentos, y divisiones de empresa son todos ejemplos de grupos de usuarios que podrían ser miembros de la misma organización de proveedores en API Connect.



API consumer actions  
(Consumer organization)

# Flujo API

Acción: 3 4 5



transbank.

DFT  
solutions



API provider actions  
(Provider organization)

Manage  
monitor the  
in the c

Create  
manage p  
organiza

Create a d  
in a pro  
organiz

Call the API  
in the Plan  
from a developer  
application



12

Subscribe to  
a Plan in the  
Product to gain  
access to the API



11

Discover the  
Product in the  
Developer Portal



9

Configure the  
Developer Portal  
site and invite  
users to join



API consumer actions  
(Consumer organization)

deploy  
application

Desarrollador de API

Una vez definido como usuario en una organización de proveedores y asignado permisos de acceso, un desarrollador de API (al que se le podría asignar más de un rol) puede diseñar desarrollar y probar las API, y asociarlas a Planes y Productos. Como desarrollador de API, se pueden especificar valores de política para limitar el uso de las API expuestas por el Plan. Se puede definir una política de cuota única que se aplique a todos los recursos de API a los que se accede a través del plan o se pueden definir políticas de cuota independientes para recursos de API concretos. También se pueden definir políticas en recursos de API para configurar prestaciones como, por ejemplo, la seguridad, el registro cronológico, el direccionamiento de peticiones a servicios de destino y la transformación de datos de un formato a otro. Estas políticas controlan aspectos del procesamiento en la Pasarela durante la gestión de una invocación de API y son los bloques de construcción de los flujos de ensamblaje. Mientras desarrollan y mantienen las API, también se pueden crear destinos de despliegue llamados *Catálogos* a efectos de pruebas y producción. Cada Catálogo se asocia a un Portal de desarrollador y a un punto final concretos. Si se tienen privilegios administrativos se puede restringir el acceso de despliegue a un Catálogo y requerir acciones como, por ejemplo, la aprobación del despliegue de nuevas versiones de API.





1

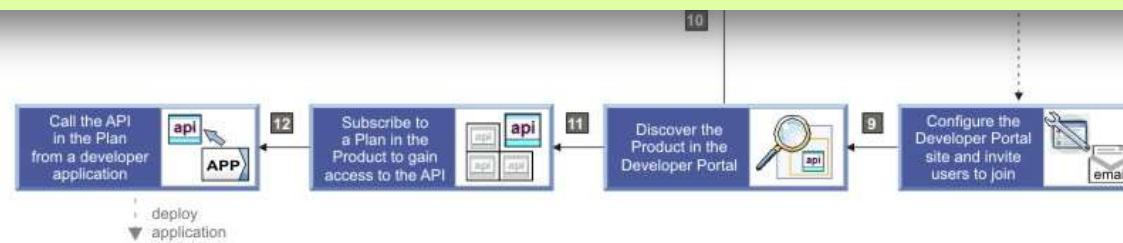
## Acción: 7 8



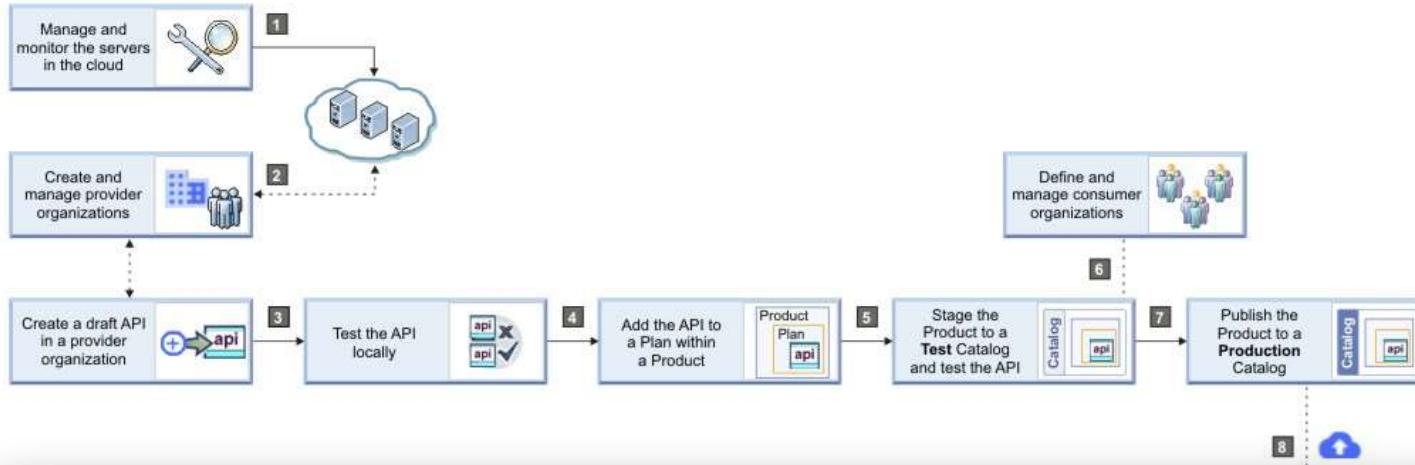
**Administrador de API** Una vez creadas y probadas las API satisfactoriamente, un Administrador de API publica uno o varios *Productos* para exponer las API en el Portal de desarrollador para su descubrimiento y uso. Las API se incluyen en un *Plan*, que está contenido en un Producto, antes de ser publicadas y pueden ser publicadas en una o más organizaciones de consumidores, restringiendo así la visibilidad de las mismas. Solo los desarrolladores de aplicaciones de la organizaciones especificadas podrán ver el API en el Portal de desarrollador y obtener claves de aplicación para acceder a ella. El administrador de API también se encarga de gestionar el ciclo de vida de los Productos y sus API asociadas y utiliza la analítica para realizar un seguimiento del uso de API y determinar si un API está cumpliendo su finalidad.



API provider actions  
(Provider organization)



API consumer actions  
(Consumer organization)



## Acción: 9

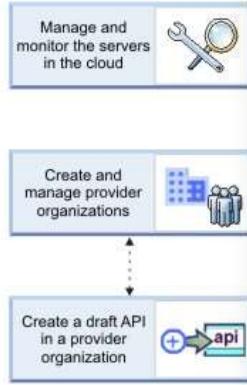
## Comercio!

**Propietario org. consumidores** Una vez creada una organización de consumidores, el Propietario de la organización de consumidores nombrado puede invitar a otros usuarios a unirse a la organización de consumidores para que puedan acceder al Portal de desarrollador y utilizar las API expuestas a la organización de consumidores. El Propietario de la Organización de consumidores u otro usuario con el correspondiente acceso también pueden configurar el sitio de Portal de desarrollador; por ejemplo, personalizar su aspecto, crear y controlar foros, entradas de blog y configurar blogs.



API consumer actions  
(Consumer organization)

# Flujo API



Acción: 10 11 12

**Desarrollador aplicac.** Una vez publicado un Producto, los desarrolladores de aplicaciones autorizados obtendrán acceso a sus API registrando aplicaciones para acceder a los Planes de dicho Producto. Un desarrollador de aplicaciones utiliza el Portal de desarrollador para buscar una API necesaria, suscribirse a su Plan asociado e incluir el API en una aplicación que posteriormente se pueda desplegar en un dispositivo.

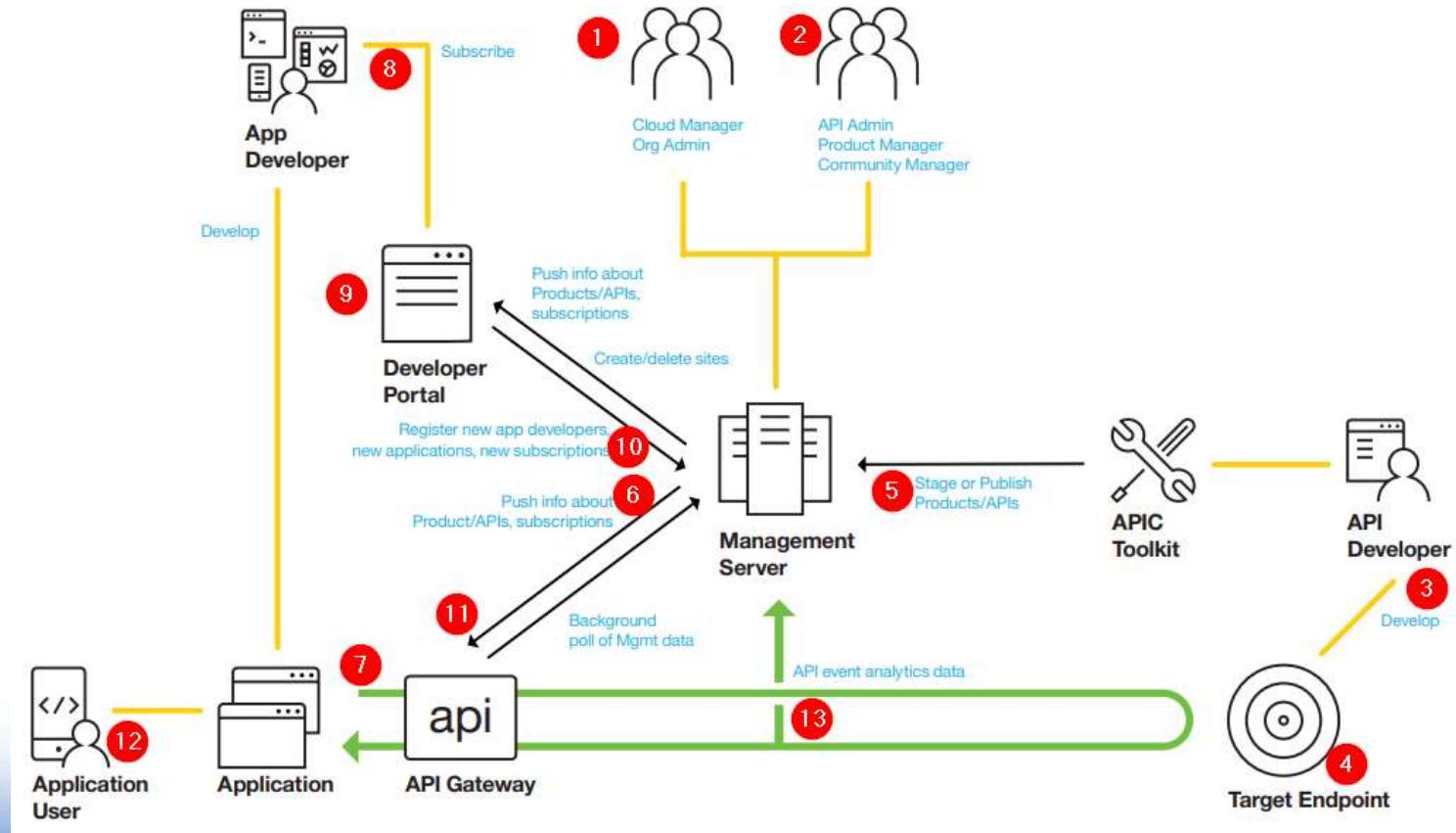


comercio

Cuando el API se invoca desde la aplicación desplegada en un dispositivo, un ejemplo de flujo de solicitudes/resuestas de las interacciones en tiempo de ejecución de API Connect podría ser el siguiente:

1. El usuario del dispositivo abre la aplicación, que emite la petición de API.
2. La pasarela y el entorno de ejecución del API manejan la solicitud (y realizan el equilibrado de carga y la validación de seguridad en todas las peticiones de API) y el API de ejecución:
  - a. La Pasarela valida las políticas de acceso a la Gestor de API e invoca el API.
  - b. El entorno de ejecución del API ejecuta el API y obtiene la carga útil de datos procedente del sistema de fondo (backend).
  - c. La respuesta del API se devuelve a la Pasarela.
  - d. La Pasarela reenvía la respuesta a la aplicación invocadora.
  - e. La Pasarela notifica métricas de uso a la Gestor de API.
  - f. La pasarela notifica datos de análisis al servidor de análisis.





## **Usuarios y Roles**

## Digital transformation demands a new architecture

<p><b>Client-Tier</b> Mobile, IoT, Web</p> 	 <p>Apps</p>
<p><b>Interaction Services Layer</b></p> 	<p><b>What's needed is the <b>Interaction Services Layer</b></b></p> <ul style="list-style-type: none"> <li>• Designed for a microservices architecture</li> <li>• Non-blocking, event-driven I/O to remain lightweight</li> <li>• Efficient in the face of data-intensive real-time applications</li> <li>• Supports massive concurrency</li> <li>• Designed for hybrid cloud deployment</li> <li>• Seamless communication between front-end and back-end systems</li> <li>• Simplified &amp; comprehensive API lifecycle to Create, Run, Manage and Secure APIs</li> </ul>  <p>IBM API Connect</p>
<p><b>Middle-Tier</b></p> 	<p>Traditional SOA infrastructure designed for internal integration does not cut it for real-time external interactions</p>
<p><b>Enterprise Applications &amp; Data Back-end</b></p> 	<p>Need for simplified discovery and secure reuse of Systems of Record via APIs</p>



## Four Primary Roles



**DIANA**  
API DEVELOPER

- Create and test APIs from existing services
- Add value with Gateway policies
- Stage completed APIs into Catalogs



**OLIVIA**  
API OPERATIONS

- Approves subscriptions that require authorization
- Monitors the Operational Analytics dashboards
- Creates custom analytics reports



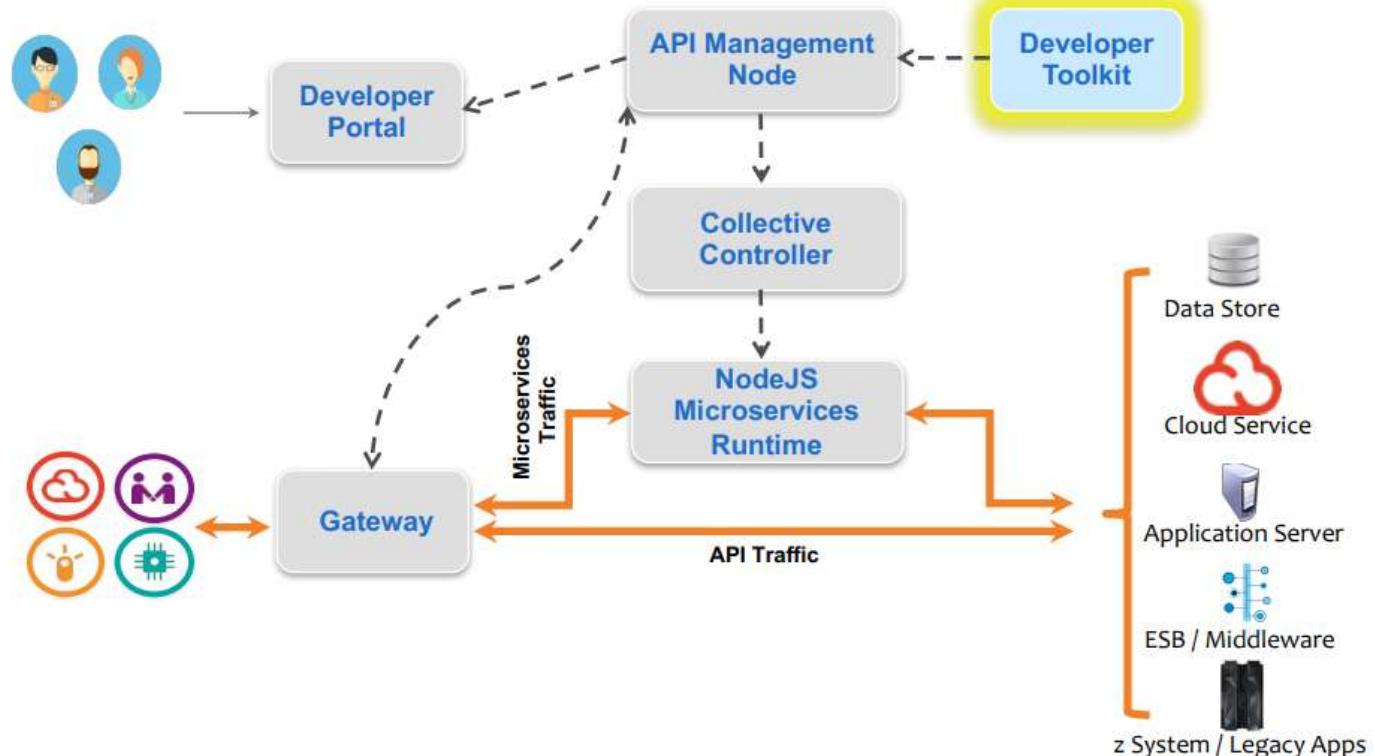
**SHAWN**  
API PRODUCT  
MANAGER

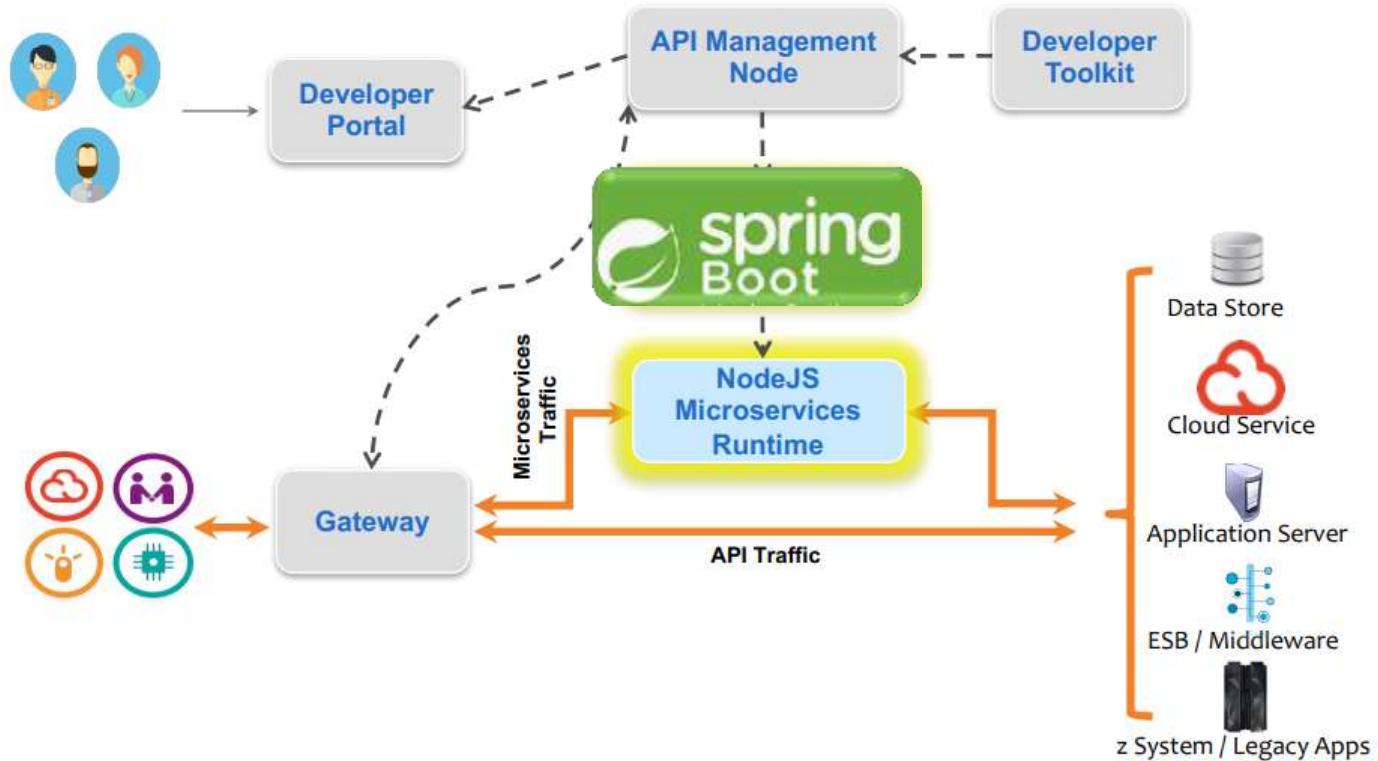
- Bundle APIs into consumable Products
- Create Plans and determine SLAs
- Publish Products and Plans into developer Portal

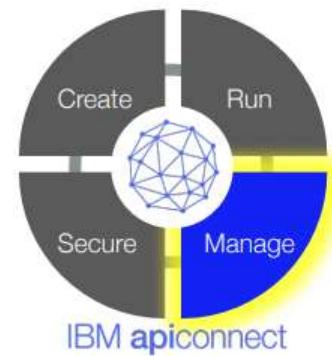


**DEREK**  
3<sup>rd</sup>-PARTY  
DEVELOPER

- Reviews and tests available API Products
- Registers apps in the Developer Portal
- Subscribes to a specific Plan for an API Product



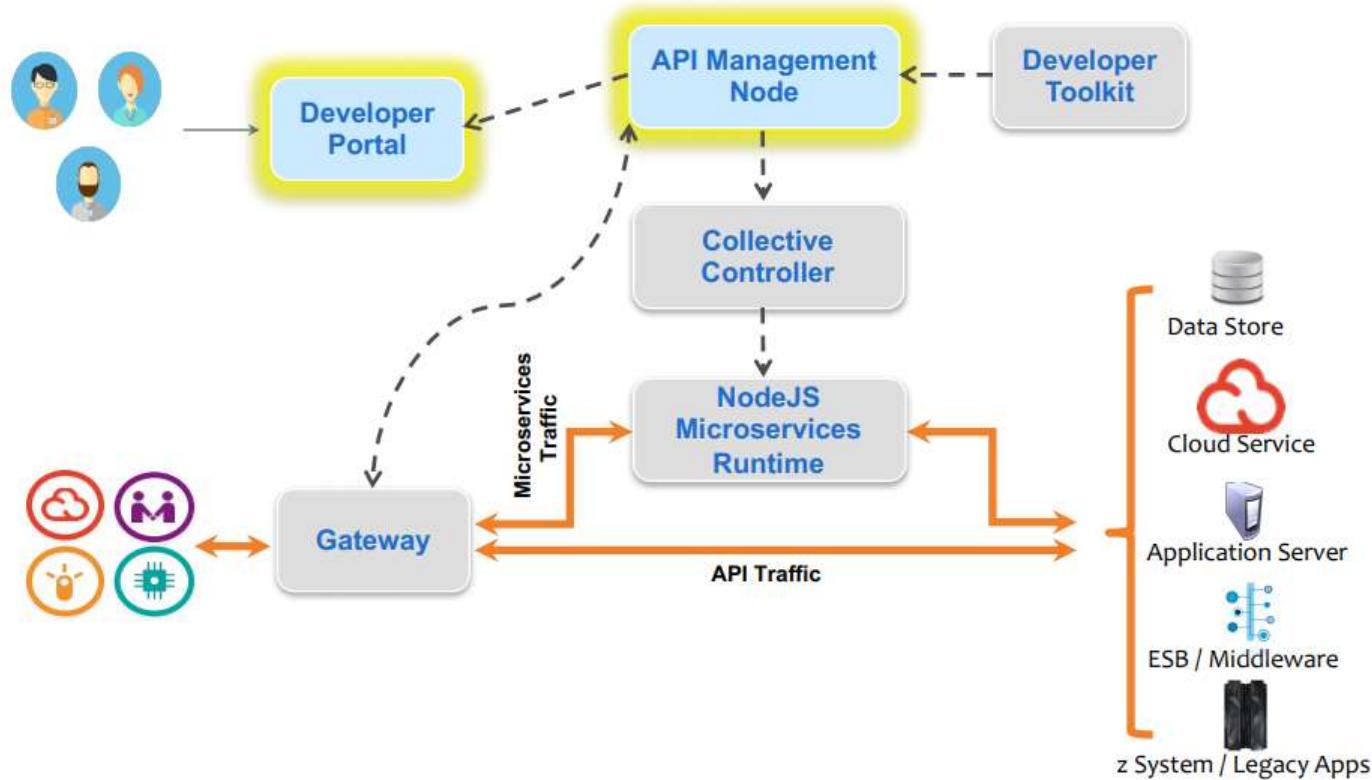


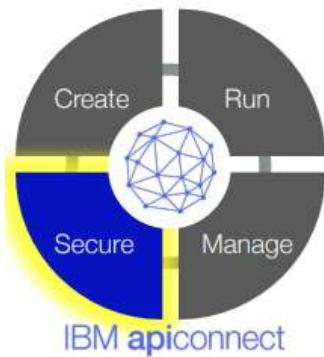


**SHAWN**  
API  
PRODUCT  
MANAGER

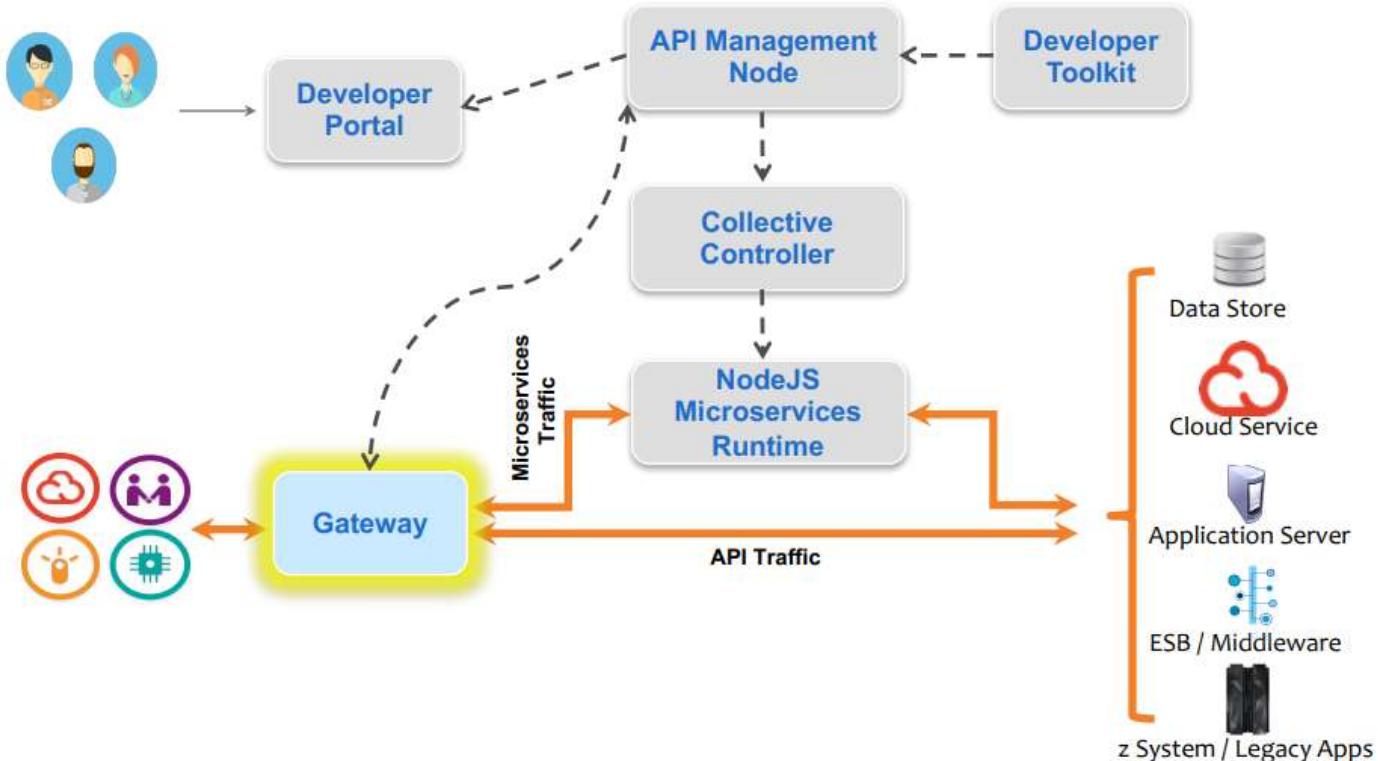


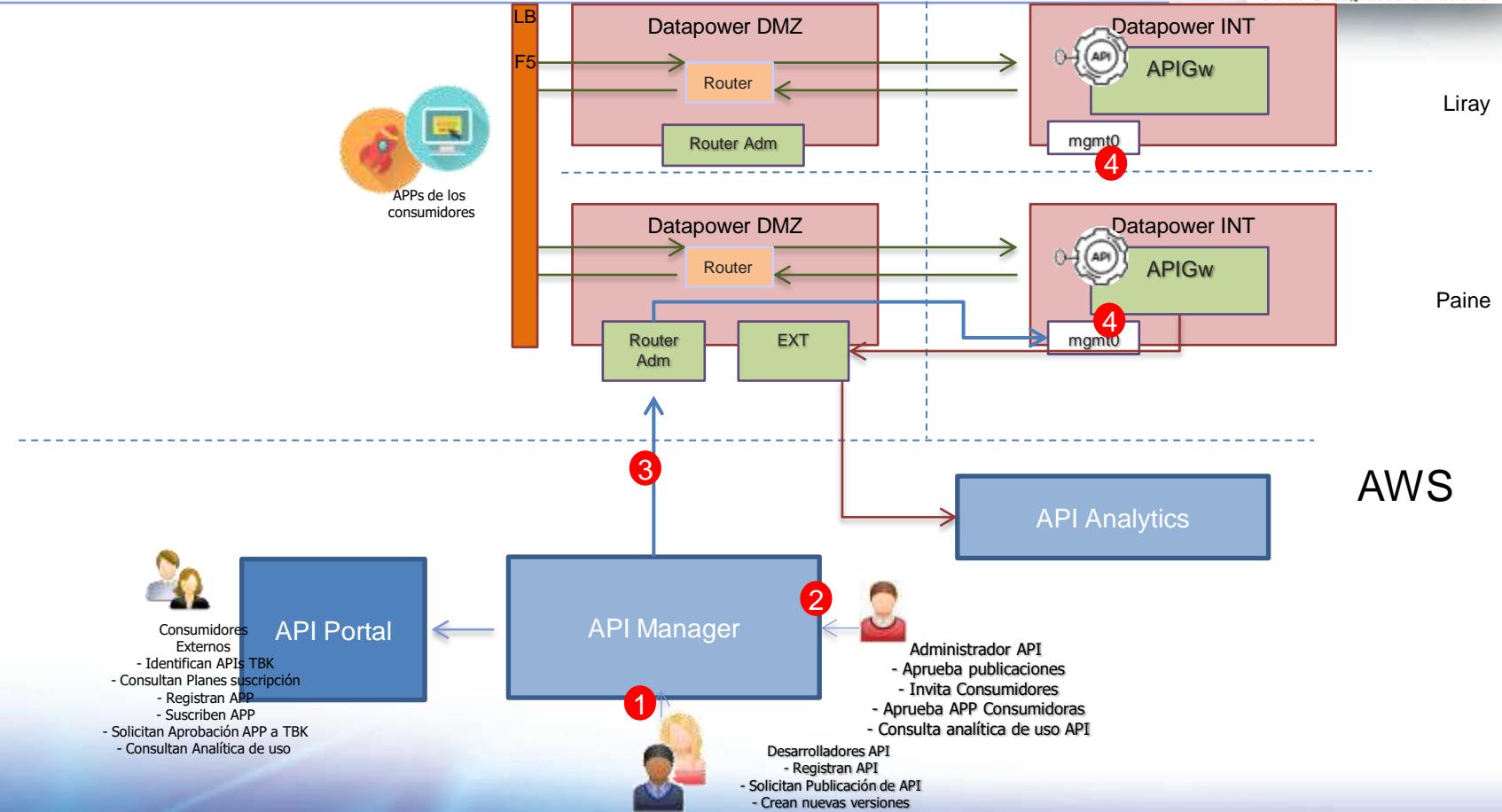
**DEREK**  
3rd-PARTY  
DEVELOPER

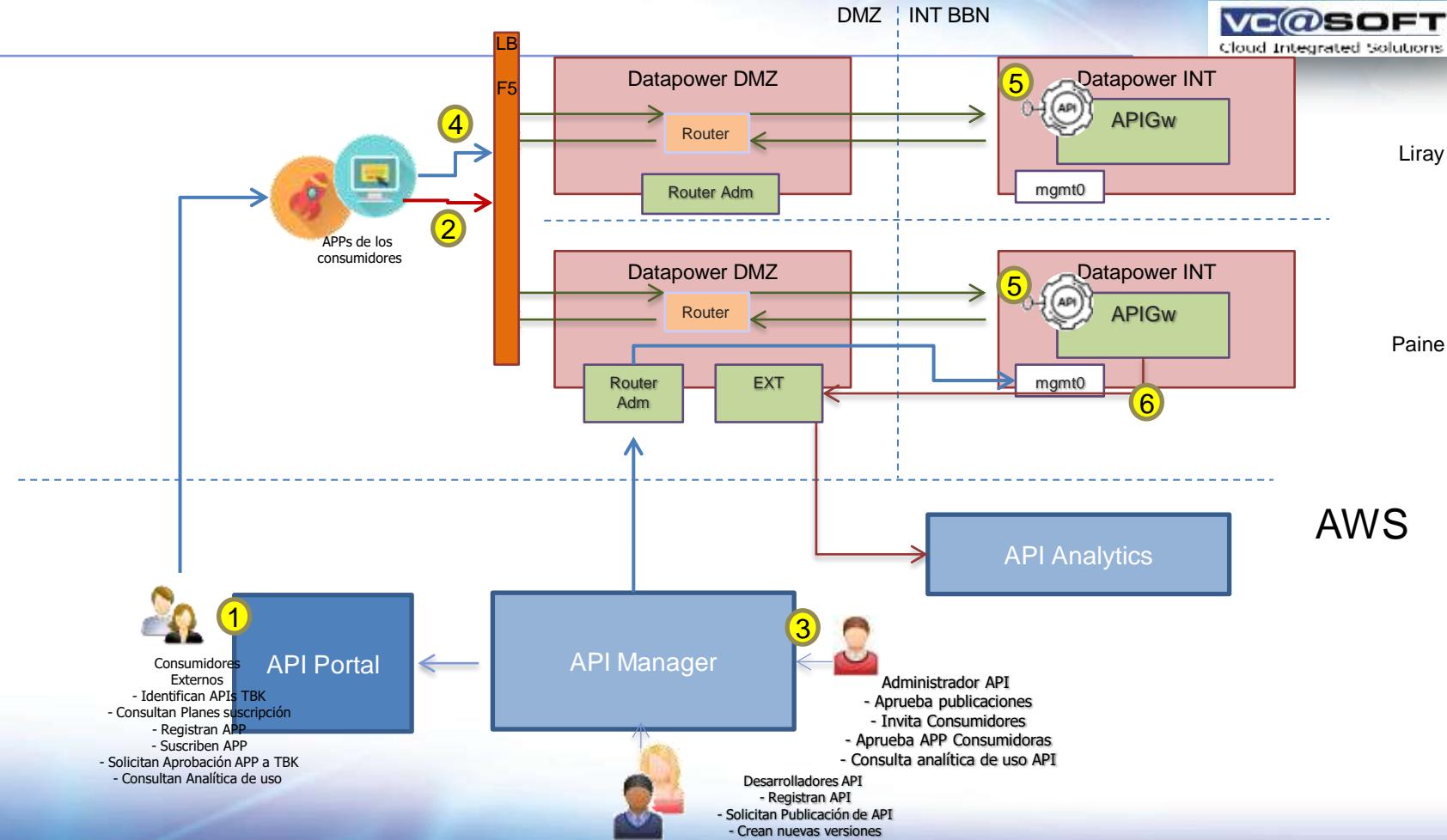




**OLIVIA**  
API OPERATIONS







# **APIC**

## **Componentes y Herramientas**

# Componentes

IBM APIC		
Cloud Manager	<a href="https://cloud.apic.dev1.tbk.cl">https://cloud.apic.dev1.tbk.cl</a>	
API Manager	<a href="https://apim.apic.dev1.tbk.cl">https://apim.apic.dev1.tbk.cl</a>	
API Gateways	IBM DataPower	
Developer Portal	<a href="https://developer.apic.dev1.tbk.cl/transbank/sandbox/">https://developer.apic.dev1.tbk.cl/transbank/sandbox/</a>	
API Analytics		
The developer toolkit	<a href="https://apim.apic.dev1.tbk.cl">https://apim.apic.dev1.tbk.cl</a>	

# API Designer

apic-slim.exe  
apic.exe  
api\_designer-win.exe

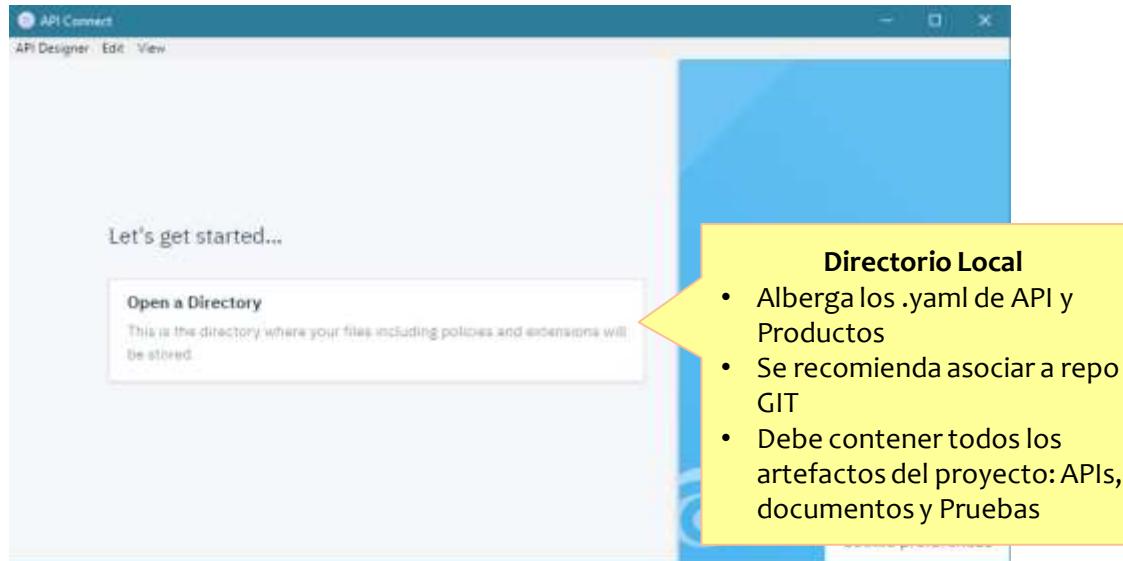
The screenshot shows the IBM API Connect API Designer application window. The title bar includes 'API Connect', 'API Designer Edit View', 'IBM API Connect', 'API Designer', 'Switch Cloud Connection', 'Organization: Transbank', and a user icon. The main menu bar has 'Develop' selected, followed by 'test-v... 1.0.0', 'Design', 'Source', 'Assembly', and status indicators 'Running' and 'No Errors'. A 'Save' button is also present.

The left sidebar navigation menu is open, showing the 'API Setup' tab is selected, along with other options like 'Security Definitions', 'Paths', 'Definitions', 'Properties', 'Target Services', 'Categories', and 'Activity Log'. The main content area is titled 'Info' with the sub-instruction 'Enter the API summary details'. It contains fields for 'Title' (set to 'Test Ventas'), 'Name' (set to 'test-ventas'), 'Version' (set to '1.0.0'), and a 'Description (optional)' field containing the following text:

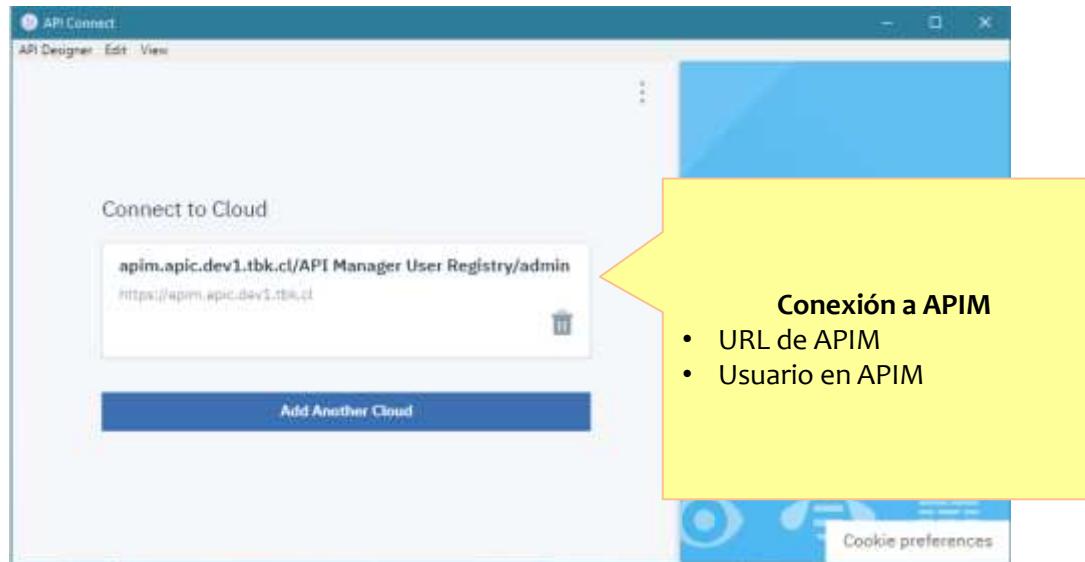
# Transacción Financiera Transbank  
API de transacciones financieras realizados en Transbank.  
Por medio de esta API se le ofrecen a los comercios de TBK los mecanismos de consulta de información de  
\*\*Transacción Financiera\*\*.

At the bottom right of the main content area, there is a 'Cookie preferences' link.

- Herramienta local de diseño de API



- Herramienta local de diseño de API



# APIC Designer

The screenshot displays the APIC Designer application interface, which is a cloud-based API management tool. The main window shows a dashboard with a "Welcome to the API Designer" message and two large cards: "Develop API and Products" and "Explore APIs". The left sidebar includes icons for Home, Help, and Logout, along with a search bar and a "Develop" section. The main content area has tabs for "API Connect", "IBM API Connect", and "API Designer", with the "API Designer" tab currently selected. The "API Designer" tab has sub-options for "Edit", "View", and "Switch Cloud Connection" (set to Transbank). The "Develop" section contains a table for managing APIs and products, with columns for TITLE, TYPE, LAST MODIFIED, and a blue "Add" button. The table shows one entry: "Product".

TITLE	TYPE	LAST MODIFIED
	Product	

Add API

- From target service**  
Create a REST proxy that routes all traffic to a target API via service endpoint.
- From existing OpenAPI service**  
Create a REST proxy based upon an OpenAPI described target service.
- From existing WSDL service (SOAP proxy)**  
Create a SOAP proxy based upon a WSDL described target service.
- From existing WSDL service (REST proxy)**  
Create a REST proxy based upon a WSDL described target service.
- New OpenAPI**  
Compose a new REST proxy by defining paths and operations.

Import

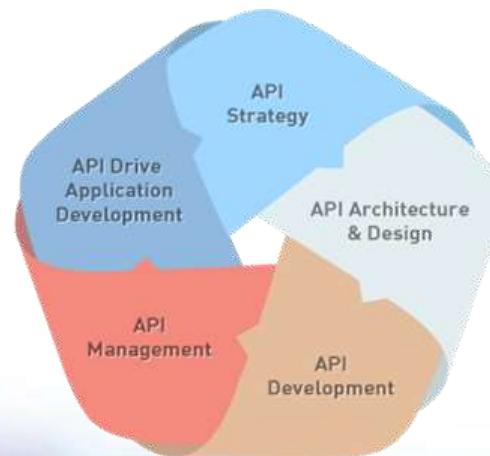
- Existing OpenAPI**  
Use an existing definition of a REST proxy or SOAP API.

## ■ Escenarios de creación de APIs

- ❖ **Proxy:** Envía las invocaciones a un target existente
- ❖ **Basado en Contrato:** Permite cargar el contrato OpenApi del API y a partir de esta definición crear la configuración de invocaciones
- ❖ **Proxy SOAP:** En Base a un WSDL crea un SOAP proxy que invoca al SOAP existente
- ❖ **REST proxy SOAP:** En Base a un WSDL crea un REST con mapeo directo al SOAP.
- ❖ **En Blanco:** Define el OpenAPI desde cero
- ❖ **Importar un API de otro entorno APIC:** Carga el archivo .yaml de la definición APIC.

# ¿Cuál escenario adoptar?

- Contrato Inicial (Contract First)
- El desarrollo en APIC no es el único elemento en el ciclo de vida del API
  - ❖ El contrato es un artefacto que evoluciona desde la estrategia hasta la implantación
  - ❖ Cada actor requiere responsabilizarse por diferentes vistas del Contrato



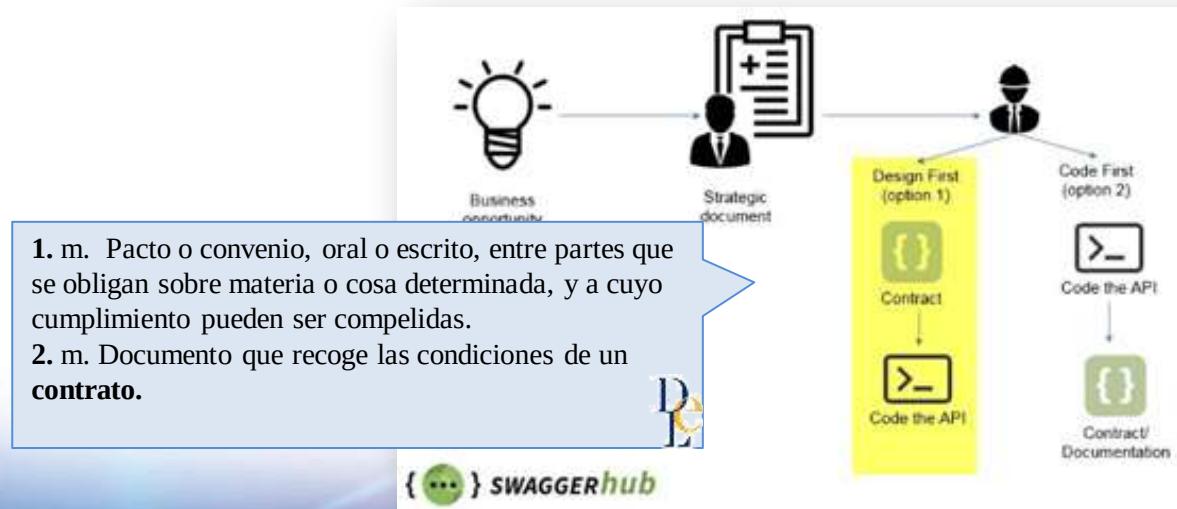
# Contrato en varios ciclos de vida

- APIC o APIC designer posiblemente no son la mejor herramienta para iniciar el diseño del contrato OpenAPI



## ■ Code First vs Contract First

- ❖ API es un Producto, no un webservice ni una interface
- ❖ Code First plantea que el API es el subproducto del desarrollo: El código se crea primero y el sub-producto es el API.
- ❖ Contract-First piensa que se requiere un acuerdo previo.

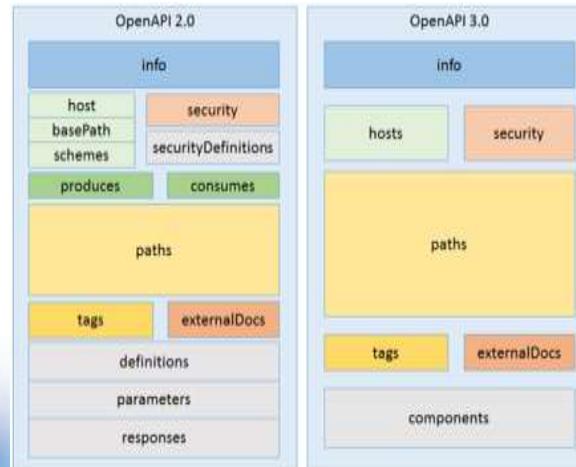


# ¿Cuál Contrato?

- El estándar de facto: OpenAPI



- Versiones 2 y 3



# Diseño de OpenAPI en APIC Designer

- Nombre del API como es entendido por el negocio.
- Se debe mantener una estructuración de nombre consistente entre APIs pero el foco principal es lograr un entendimiento claro para los consumidores
- Es importante que sea el área de **arquitectura** el que segregá responsabilidades y define el alcance del API. (el negocio no entiende necesariamente las implicaciones de granularidad y tamaño del API)
- Esta decisión de alcance de API se basa más en un análisis de necesidades de los consumidores y de DDD (Domain Driven Design) que al entendimiento de capacidad que pueda plantear el proveedor y que puede plantear APIs con granularidades que no facilitan el uso o el gobierno del API.

- Un API puede exponer un agregado de entidades por lo que no es necesario que el nombre del API sea una Entidad.
- Un API es más un contexto delimitado con la unidad interna de entidades que esta unidad requiera.
- No es necesariamente una entidad con su CRUD
  - ❖ No es obligatorio que el API exponga todo el modelo CRUD de una entidad
  - ❖ Definir que el nombre del API sea una entidad (sustantivo) y agregar todas las operaciones posibles en el API puede plantear un modelo de API de un tamaño inconveniente para su gobierno.
- Es preferible plantear un modelo CQRS donde se propone una segregación entre
  - ❖ "comandos" que modifican el recurso,
  - ❖ consultas que tratan el recurso como un elemento inmutable. Esto plantea que el nombre del API podría representar esta separación incluyendo estos dos tipos de APIs.

- **Nombre técnico:** Se utiliza una extensión propia de IBM (APIC) para registrar el nombre técnico del API
- **Descripción de API:** Es información que ayuda al potencial consumidor a entender si este API es el que le ofrece la capacidad esperada. Es un texto orientado a los usuarios del API, no es la misma documentación del webservice proveedor que describe aspectos de implementación o de conocimiento interno. Adicional a la información específica de aclaración del servicio, este texto es un lugar apropiado para hacer referencia a las condiciones de seguridad que se deben cumplir, a los compromisos asumidos al invocar el servicio, y a documentos de diseño que ayudan a contextualizar la invocación que deberá realizar este consumidor.
- **Términos de Servicio:** El área legal de TBK deberá documentar las condiciones de servicio y se podrán registrar en el Portal de desarrollador para que puedan ser consultadas por los comercios.
- **Contacto:** Nombre, URL e email de contacto que el consumidor de API podrá utilizar para comunicarse al interior de TBK buscando apoyo en su consumo de API. Es importante que este mail si sea mantenido por un actor que responda a requerimientos de apoyo. Se debe definir si existirán mails por área de dominio de APIs
- **Licencia:** URL y nombre de licencia de uso del API.
- **Versión de la API.** Se adopta numeración de versionamiento semántico

- OpenAPI ofrece las siguientes Definiciones de Seguridad

## Definiciones de Seguridad

- ❖ Autorización/Autenticación por APIKey/API secret
- ❖ Autorización Basic
- ❖ Autorización Oauth2

- Sigue los flujos del framework OAUTH2
- Es estándar de facto para autorización delegada de usuarios finales (cuando se necesita confiar que el dueño final de la cuenta accede a delegar la autorización)
- Se describe más adelante

- Solicita una identidad y una contraseña en el Header Authorization
- El tipo de contenido del header Authorization en “Basic”
- La identidad y contraseña se envían concatenadas con “:”
- El resultado se envía Base64 para evitar corrupciones por cambio de juego de caracteres
- Podría ser APIKEY:SECRET

- El request del API debe incluir en Header el valor de APIKey asignado a la Aplicación consumidora.
- Para escenarios confidenciales se debe solicitar el Secret
- El nombre de los Header será:
  - X-Client-Id
  - X-Secret-Id

- La semántica de las operaciones se acogen a las definiciones de métodos HTTP: solo se utiliza
  - ❖ POST para creación,
  - ❖ GET, HEAD y OPTIONS para consultas,
  - ❖ PATCH y POST para Modificaciones,
  - ❖ DELETE para inactivaciones o eliminaciones
- Las operaciones deben cumplir que
  - ❖ GET, HEAD y OPTIONS son métodos seguros (que puede ser invocado sin causar efectos colaterales) y
  - ❖ GET, HEAD, OPTIONS PUT y DELETE son métodos idempotentes (que puede ser invocado varias veces obteniendo el mismo resultado).

	Seguro	Idempotente
GET	Sí	Sí
HEAD	Sí	Sí
OPTIONS	Sí	Sí
PUT	No	Sí
DELETE	No	Sí
POST	No	No
PATCH	No	No

- Si se requiere que los métodos POST y PATCH sean idempotentes se deberá utilizar un header identificador para idempotencia, este header apoya que la implementación del Servicio (no APIC) pueda comprobar si se trata de una solicitud nueva o una ya invocada. Esta implementación es una capacidad funcional y está fuera del alcance de APIC.
  - ❖ Nombre de Header: **Idempotency-Key**
- Siempre se deben definir los valores de
  - ❖ OperationId (nombre técnico de la operación),
  - ❖ resumen (summary) y
  - ❖ TAG.

- Definiciones de entidades que utiliza el OpenAPI
- Utilizan Schema JSON
- Las definiciones base son entidades, por lo tanto se describen con un sustantivo (no verbos)
- Los mensajes asociados a Request y Response de cada operación son tradicionalmente agregados de "definiciones" de tipo Entidad, por lo que se recomienda especificar de forma individual estas definiciones. Se recomienda utilizar un estándar de nombramiento como
  - ❖ <nombre-mensaje>\_Rq,
  - ❖ <nombre-mensaje>\_Rs,
  - ❖ <nombre-entidad>\_Rq,
  - ❖ <nombre-entidad>\_Rs,

The screenshot displays the Apicurio interface for managing APIs. On the left, a sidebar navigation includes Dashboard, APIs (selected), and Settings. The main content area shows the details for the 'Abonos' API.

**APIs > Abonos**

**Abonos**

**Abonos Transbank**  
API de abonos realizados en Transbank.  
Por medio de esta API se le ofrecen a los comercios de TBK los mecanismos de consulta de información de **Abonos**.

**Mecanismos de control de acceso y autorización**  
Este API obliga que el co...

Created on Mar 10, 2020  
Created by tbk-vcsoft@vc-soft.com  
3 Other Collaborator(s)

Tags: **Abonos**

**Edit API** | **Preview Documentation**

**Top Contributors**

tbk-vcsoft@...	157 of 300 edits
jaimevale@...	77 of 300 edits
jbernal@vc-...	72 of 300 edits

**Activity Log**

- jbernal@vc-soft.com changed the value of the example for content-type 'application/json' for the Response at location /paths/{abonos/ventas}/get/responses[200].  
14 hours ago
- jbernal@vc-soft.com changed the value of property 'example' at location /definitions[VentasRs].  
14 hours ago
- jbernal@vc-soft.com changed the value of property 'example' at location /definitions[Venta].  
14 hours ago
- + jbernal@vc-soft.com added a new Schema Property named 'totalAbonoDolares' at location /definitions[venta].  
14 hours ago
- + jbernal@vc-soft.com added a new Schema Property named 'totalAbonoPesos' at location /definitions[venta].  
14 hours ago
- i jbernal@vc-soft.com changed the type of the Schema Property named 'data' at location /definitions[VentasRs]/properties[data].  
14 hours ago
- + jbernal@vc-soft.com added a new Schema Property named 'movimientos' at location /definitions[venta].  
14 hours ago

# Elementos en OPENAPI 2.x

## PATHs

- REST no es un estándar, es un estilo de arquitectura
- No hay una autoridad que defina el estándar de paths en REST
  - Por lo que hay muchos estándares
- Colecciones: Sustantivos plural

APIs > Abonos > Editor

## Abonos

Search everything...

Design Source

- Paths [4]   
/abonos/por-cuentas  
/abonos/totales-proximos  
/abonos/totales-realizados  
/abonos/ventas

- Data Types [15]   
 \_error  
 \_links  
 \_meta  
 \_pagination  
 \_verificationError  
 MontoAbono  
 Movimiento  
 TotalesPorCuenta  
 TotalesPorCuentaRS  
 TotalesProximos  
 TotalesProximosRS  
 TotalesRealizados  
 TotalesRealizadosRS  
 Venta  
 VentasRS

- Responses   
No reusable responses have been created. Add a response

- INFO   
Version 1.1.0   
Description

**Abonos Transbank**

API de abonos realizados en Transbank.

Por medio de esta API se le ofrecen a los comercios de TBK los mecanismos de consulta de información de **Abonos**.

Mecanismos de control de acceso y autorización

Este API obliga que el comercio cuente con las siguientes condiciones:

- Cuenta activa en el portal de APIs de Transbank.
- Aplicación registrada en el portal de APIs en estado aprobado.
- Presentar el identificador único de aplicación (Cliente).

Versión Justificación

1.0.0 Version inicial creada en base al documento inicial de Req. funcionales.  
Modificaciones definidas en documento "Especificación de Requerimientos Funcionales - P0002839\_API Transbank V2.0"

Autor  
VCsoft Chile  
jaimev@vcsoft.com

Consumes (Inputs)  
**application/json**

Produces (Outputs)  
**application/json**

- CONTACT  
Name  
Transbank S.A.

CONSUMER DATA STANDARDS

Search

Introduction

**Standards**

- Principles
- Versioning
- URI Structure
- HTTP Headers
- HTTP Response Codes
- Protocol Conventions
- Common Field Types
- Regulation
- ID Persistence
- Extensibility
- Security Profile
- Consumer Experience
- Banking APIs
- Common APIs
- Schemas
- Admin APIs
- Authorisation Scopes
- Non-functional Requirements
- Known Issues
- Change Log
- Archives

Consumer Data Standards Home  
CDS Standards vs OpenAPI  
Open Report  
Swagger (JSON)  
Swagger (YAML)

## Standards

These standards represent **version 1.2.0** of the high level standards. See the [versioning section](#) for more information.

Note that, in this proposal, the key words: **MUST**, **MUST NOT**, **REQUIRED**, **SHALL**, **SHALL NOT**, **SHOULD**, **SHOULD NOT**, **OPTIONAL** are to be interpreted as described in

### Principles

The following principles, classified as Outcome P, Data Right.

#### Outcome Principles

These principles articulate qualitative outcomes:

**Principle 1: APIs are secure**  
The API definitions will consider and incorporate breach but also additional concerns of inadvertent outcome that the API standards must seek

**Principle 2: APIs use open standards**  
In order to promote widespread adoption, open

**Principle 3: APIs provide a good customer experience**  
The API definitions will consider and incorporate that are simple and enticing to use.

**Principle 4: APIs provide a good developer experience**



Home > Current Working Groups > Financial-grade API (FAPI) WG

## Financial-grade API (FAPI) WG

About Charter Status Repository

### What is the Financial-grade API (FAPI) WG?

#### News

- 2019-04-03 New bi-weekly issues call starting today on the Atlantic time schedule.
- 2019-04-01 FAPI Conformance tests and Self-certifications are now available. To see the list of results, click [here](#).

Overview

Type to search:

HTTP API Design

Introduction

Foundations

Separate Concerns

Require Secure Connections

Require Versioning in the Accepts

Support Etags for Caching

Provide Request-Id for Introspection

Divide Large Responses Across R...

Requests

Accept serialized JSON in request...

Available for online reading and in multiple formats at [github](#).

## HTTP API Design Guide

This guide describes a set of HTTP+JSON API design practices, originally extracted from work on the [Heroku Platform API](#). This guide informs additions to that API and also guides new internal APIs at Heroku. We hope it's also of interest to API designers outside of Heroku.

Our goals here are consistency and focusing on business logic while avoiding design boilerplate. We're looking for a good, consistent, well-documented way to design APIs, not necessarily the only/ideal way.

We assume you're familiar with the basics of HTTP+JSON APIs and won't cover all of the fundamentals of those in this guide.

## Abonos

Search everything...  q

Paths (4)

- /abonos/por-cuentas
- /abonos/totales-proximos
- /abonos/totales-realizados
- /abonos/ventas

Data Types (15)

- ↳ \_error
- ↳ \_links
- ↳ \_meta
- ↳ \_pagination
- ↳ \_verificationError
- ↳ MontoAbono
- ↳ Movimiento
- ↳ TotalesPorCuenta
- ↳ TotalesPorCuentaRs
- ↳ TotalesProximos
- ↳ TotalesProximosRs
- ↳ TotalesRealizados
- ↳ TotalesRealizadosRs
- ↳ Venta
- ↳ VentasRs

Responses

No reusable responses have been created. Add a response

/abonos/ventas

Design   Source

> QUERY PARAMETERS

> HEADER PARAMETERS

> OPERATIONS (1)

Get

Put

Post

Delete

Options

Head

Patch

INFO

Summary

No Summary

Operation ID

obtenerMovimientoTransaccion

Description

Consulta el listado de las ventas que componen un **Abono**

Obtiene el listado de ventas en la fecha especificada, adicionalmente se pueden filtrar códigos de comercio.

El RUT se obtiene de la organización dueña de la aplicación que realiza el consumo

El resultado se entrega paginado

Tags

Abonos

Consumes (Inputs)

↳ application/json

Produces (Outputs)

↳ application/json

# Estructuras de gobierno

- Se han planteado diferentes estructuras para lograr información de gobierno (contexto, metadata, cabeceras, etc.)
- También se han propuesto modelos de apoyo a HATEOAS
  - ❖ Para facilitar la referenciación de recursos por hypelinks
- JSON:API

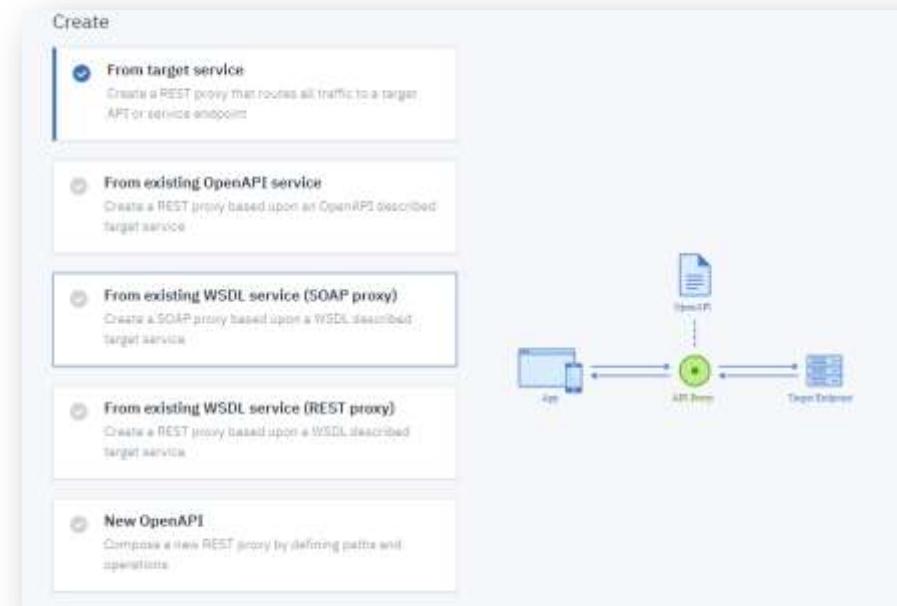
Openbanking UK

```
1  HTTP/1.1 201 Created
2  x-jws-signature: V2hhdB3ZSBnb3QgaGVyZQ0K..aXMgZmFpbHVyZS
3  x-fapi-interaction-id: 93bac548-d2de-4546-b106-880a5018460
4  Content-Type: application/json
5
6  {
7      "Data": {
8          "ConsentId": "7290",
9          "Status": "AwaitingAuthorisation",
10         "CreationDateTime": "2017-06-05T15:15:13+00:00",
11         "StatusUpdateDateTime": "2017-06-05T15:15:13+00:00",
12         "Initiation": {
13             "InstructionIdentification": "ANSM023",
14             "EndtoEndIdentification": "FRESCO.21302.GFX.37",
15             "InstructedAmount": {
16                 "Amount": "20.00",
17                 "Currency": "GBP"
18             },
19             "DebtorAccount": {
20                 "SchemeName": "UK.OBIE.SortCodeAccountNumber",
21                 "Identification": "11280001234567",
22                 "Name": "Andrea Smith"
23             },
24             "CreditorAccount": {
25                 "SchemeName": "UK.OBIE.SortCodeAccountNumber",
26                 "Identification": "08080021325698",
27                 "Name": "Bob Clements"
28             },
29             "RemittanceInformation": {
30                 "Reference": "FRESCO-037",
31                 "Unstructured": "Internal ops code 5120103"
32             }
33         },
34         "Risk": {
35             "PaymentContextCode": "PartyToParty"
36         }
37     }
38 }
```

# Tipos de Definiciones API en APIC

## ■ Tipos APIs en APIC

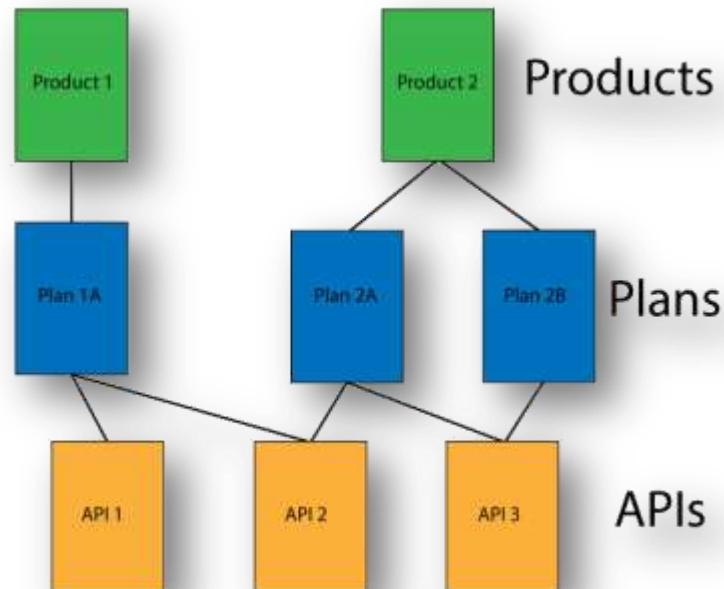
- ❖ Creación de una API de proxy REST a partir de un servicio de destino
- ❖ Creación de una API de proxy REST a partir de un servicio OpenAPI existente
- ❖ Creación de una definición de OpenAPI REST
- ❖ Creación de una API de proxy REST a partir de un servicio WSDL existente
- ❖ Adición de una API REST mediante la importación de un archivo de OpenAPI
- ❖ Creación de una API de proxy SOAP a partir de un servicio WSDL existente



# Productos y APIs

# Productos

- Elemento que los consumidores pueden suscribir para invocar el API



- API
  - ❖ Creación de una definición de API
  - ❖ Edición de una definición de API
  - ❖ Activación de una API
  - ❖ Prueba de una API con la herramienta de prueba
  - ❖ Despliegue de una API
  - ❖ Publicación de una API
  - ❖ Descarga de una definición de API
  - ❖ Eliminación de una definición de API

- Productos
  - ❖ Creación de un borrador de producto
  - ❖ Download de draft de producto
  - ❖ Edición de draft de producto
  - ❖ Despliegue de producto
  - ❖ Publicación de producto
  - ❖ Creación de una versión nueva del producto
  - ❖ Eliminación de un producto

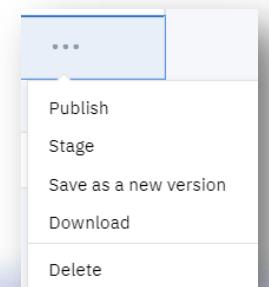
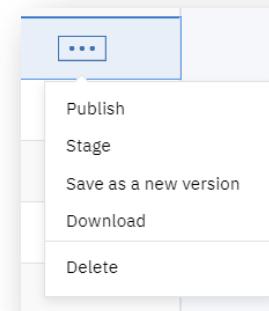
## Develop

### APIs and Products

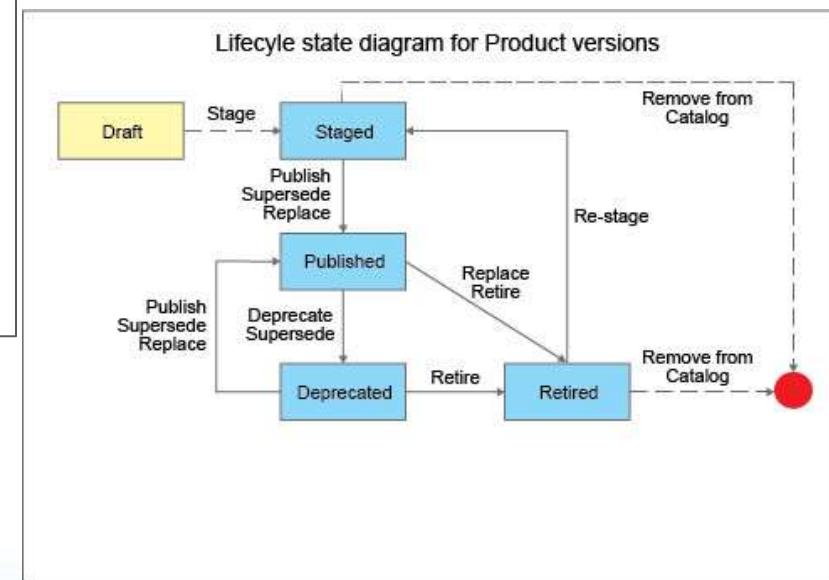
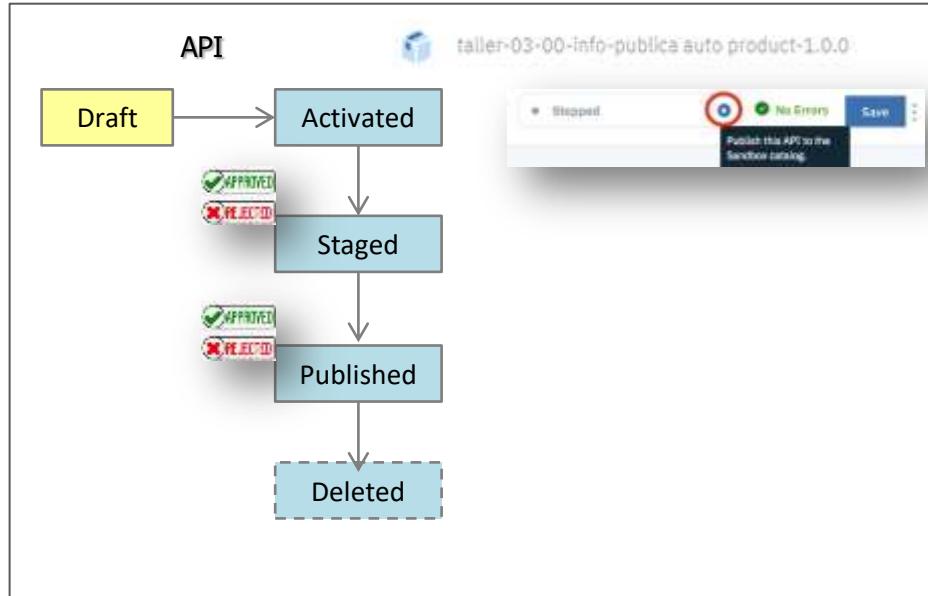
TITLE	TYPE
Abonos-1.1.0	API (REST)
Taller-01-00_where-is-my-ip-1.0.0	API (REST)
taller-03-00-info-publica-1.0.0	API (REST)
test-01-00	API (REST)
Inv-Ganancias-1.0.1	API (REST)
Test Version-1.0.0	API (REST)
Ventas-1.1.0	API (REST)
	Product
Taller-01-00_ProducAuto-1.0.0	Product
Taller-01-00_where-is-my-ip auto product-1.0.0	Product
taller-03-00-info-publica auto product-1.0.0	Product
test-1.0.0	Product

APIs

Productos

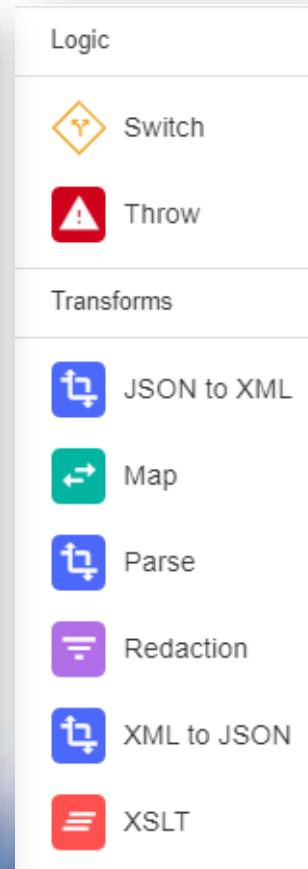
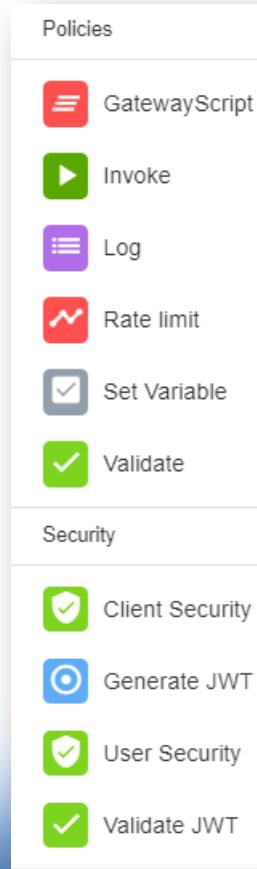


# Operaciones sobre Drafts

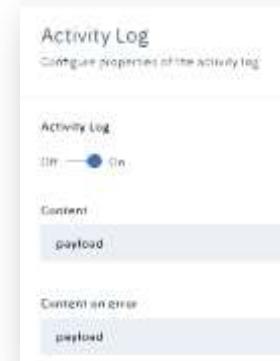


# Políticas ejecutables dentro de API

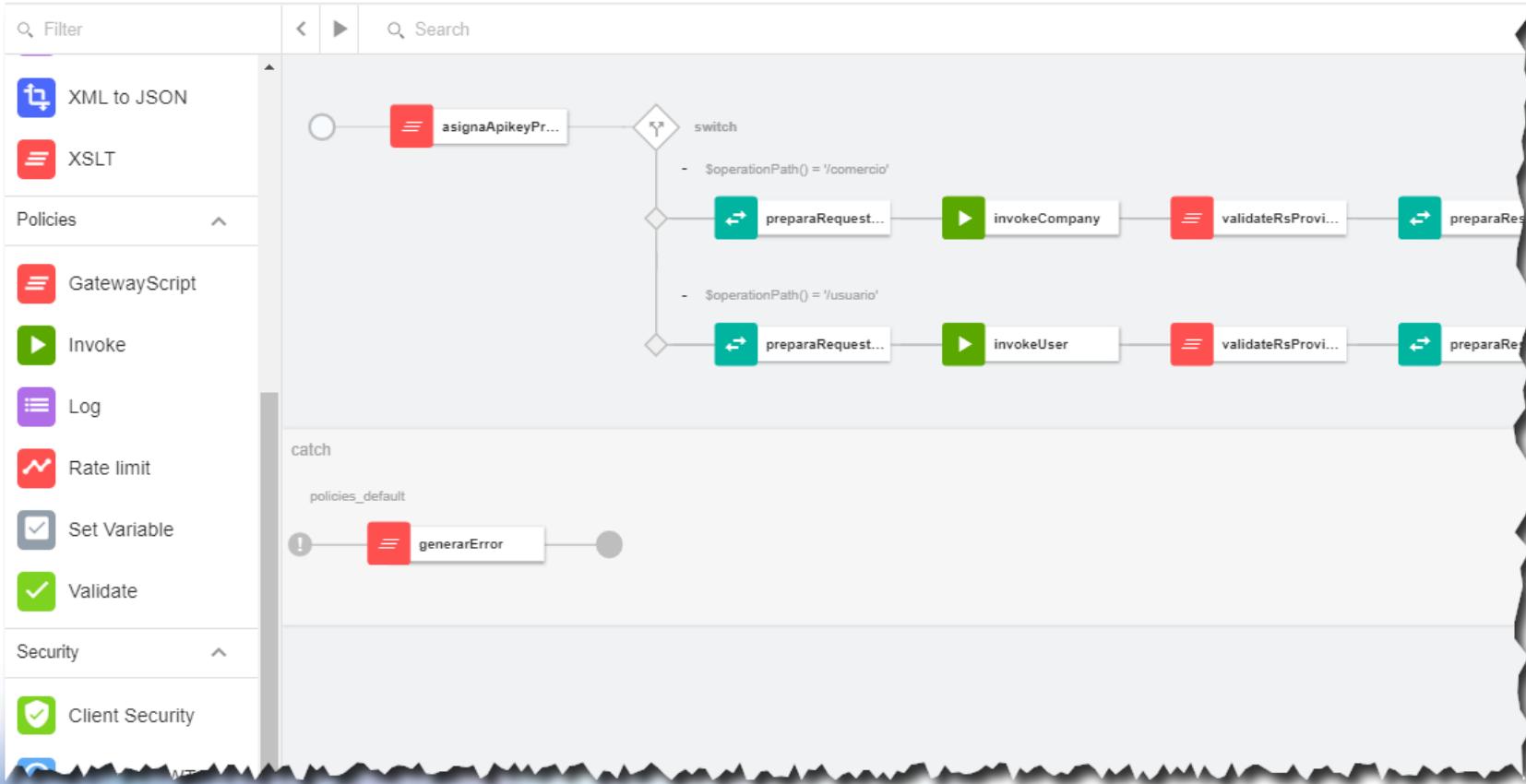
Para definir en el Assembly



Global para todo el flujo:



# Assembly Canvas



- Definidas en Swagger
  - ❖ Control Basic
    - No se acostumbra en API, pero se encuentra en todas las implementaciones de frameworks cliente/server.
  - ❖ Control APIKey
    - Obliga que las invocaciones se verifiquen con un identificador único asociado con la app suscrita al servicio.
  - ❖ Control OAuth2
    - Estándar de facto de autorización de APIs
    - Tradicionalmente asociado a delegación de autorización de usuarios finales

# OAuth2



## OAuth 2.0

- [OAuth 2.0 Framework](#) - RFC 6749
  - OAuth Scope
- [OAuth Grant Types](#)
  - Authorization Code
  - PKCE
  - Client Credentials
  - Device Code
  - Refresh Token
  - Legacy: [Implicit Flow](#)
  - Legacy: [Password Grant](#)
- [Client Types](#) - Confidential and Public Applications
- [Bearer Tokens](#) - RFC 6750
- [Threat Model and Security Considerations](#) - RFC 6819
- [OAuth Security Best Current Practice](#)

## Mobile and Other Devices

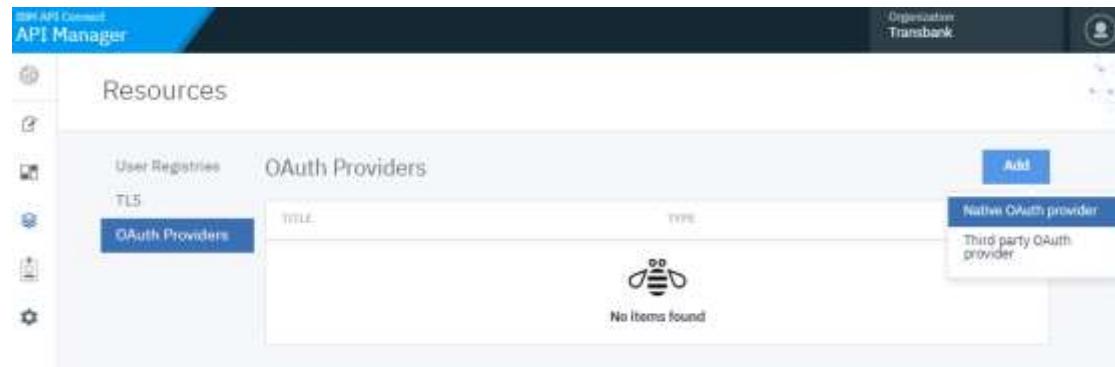
- [Native Apps](#) - Recommendations for using OAuth with native apps
- [Browser-Based Apps](#) - Recommendations for using OAuth with browser-based apps (e.g. an SPA)
- [Device Authorization Grant](#) - OAuth for devices with no browser or no keyboard

## Token and Token Management

- [Token Introspection](#) - RFC 7662, to determine the active state and meta-information of a token
- [Token Revocation](#) - RFC 7009, to signal that a previously obtained token is no longer needed
- [JSON Web Token](#) - RFC 7519

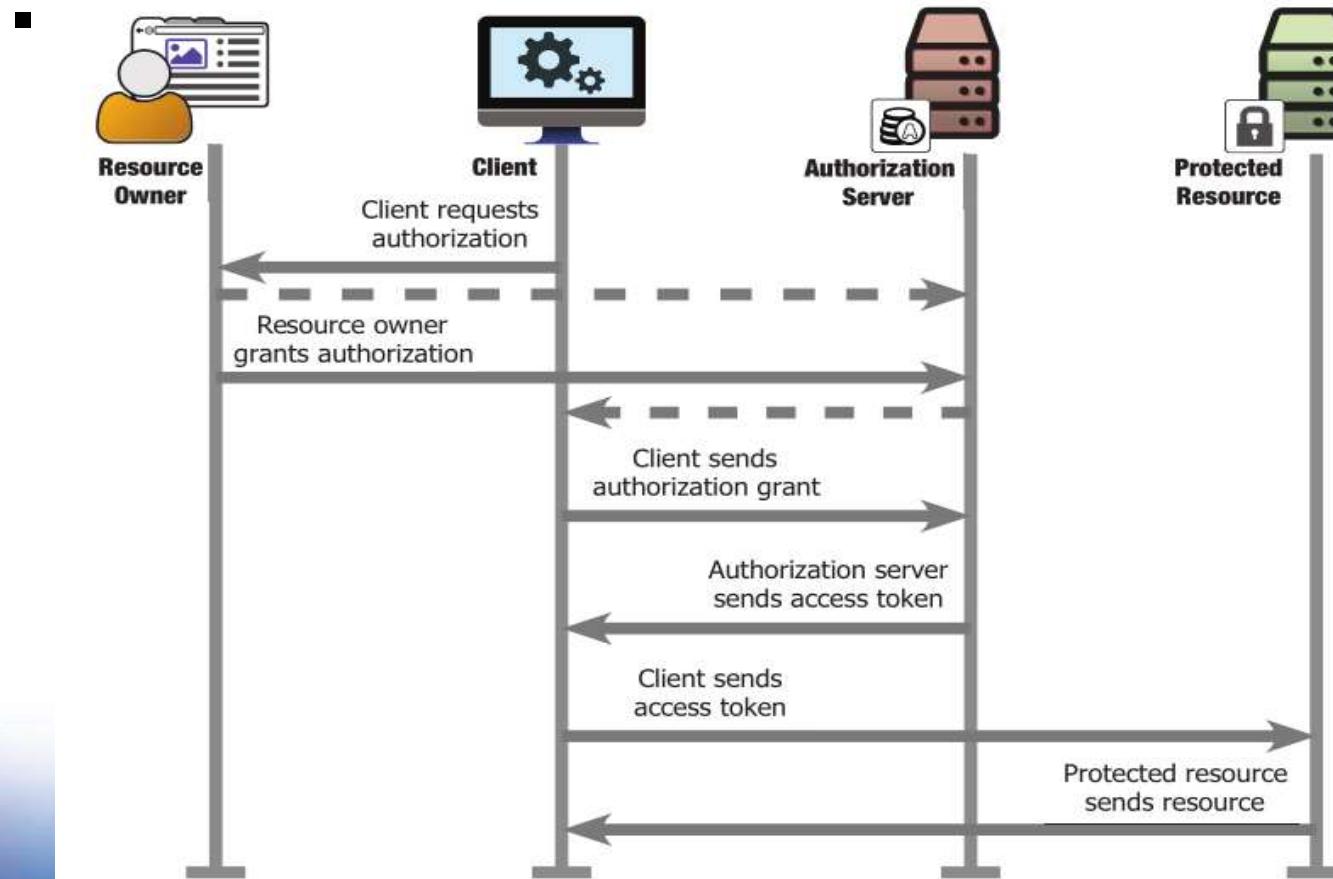
# Proveedor OAuth

- APIC puede ofrecer un proveedor OAuth
  - ❖ Pero depende de un servidor de autorización externo
  - ❖ Se pueden adoptar modelos de 2 elementos (2 legged) o 3 elementos
  - ❖ Resources > Oauth Provider > Add



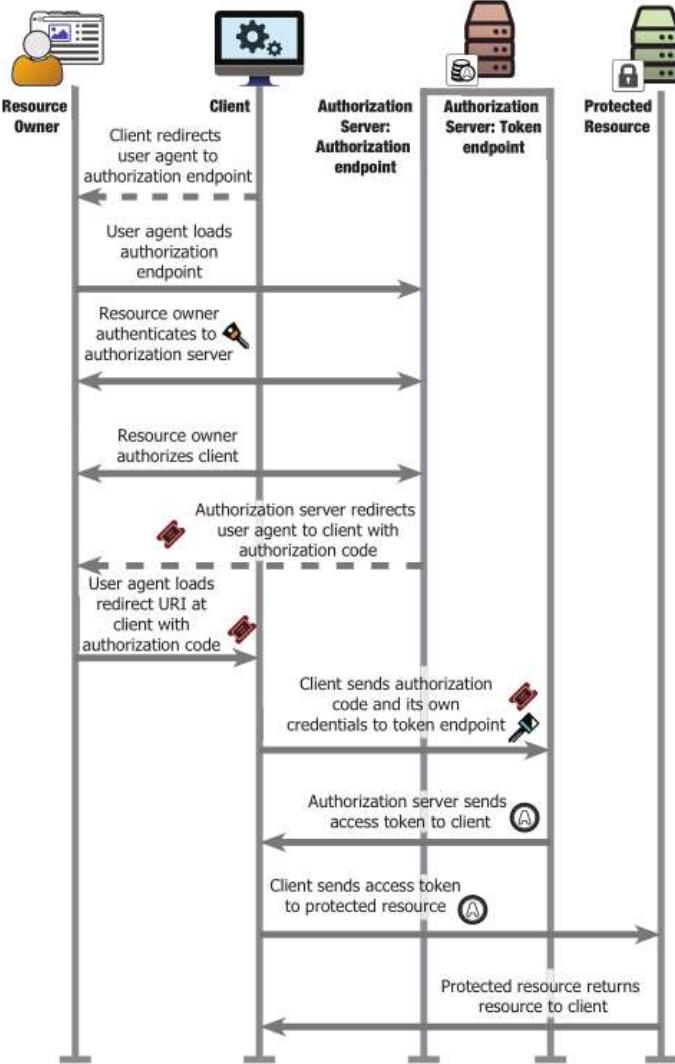
- Grants (flujos)
  - ❖ Authorization Code
  - ❖ Application
  - ❖ Password
- Scopes
  - ❖ Se recomienda una estructura como
  - ❖ Colección-subcolección:verbo:rol
    - abonos-totales:read:app

# OAuth2: visión simplificada



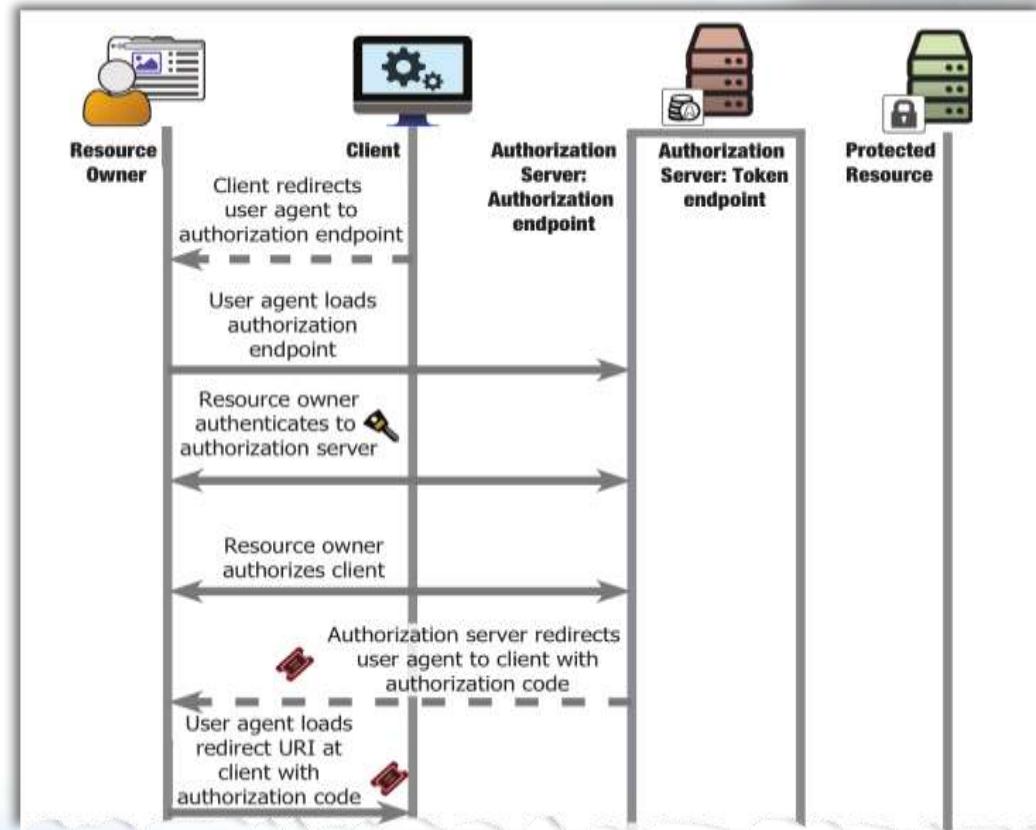
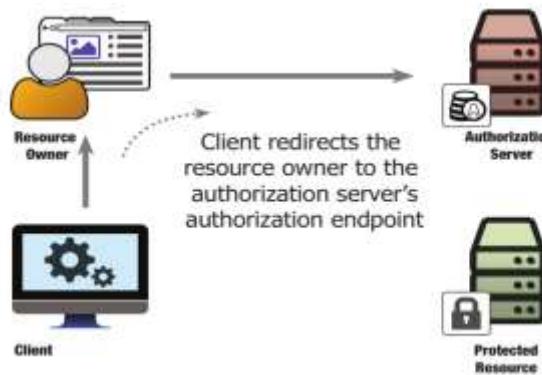
- Se logra que el cliente obtenga un mecanismo acceso a recursos del usuario
  - ❖ Obtiene un **access token**
  - ❖ Obtiene un mecanismo de renovación del **access token**
- Qué no es objetivo
  - ❖ Un protocolo de autenticación
  - ❖ Un mecanismo de delegación Usuario-Usuario
  - ❖ Un mecanismo de procesar la autorización: Se ofrece un “scope” pero no se conocen roles ni permisos de autorización.
  - ❖ No define un formato de Token
  - ❖ No se definen mecanismos criptográficos (por eso se requiere JOSE)

# Conversación OAuth



# Authorization Grant

- (1) El cliente identifica que requiere un **access token**
- (2) Redirecciona al usuario hacia el Authorization Server



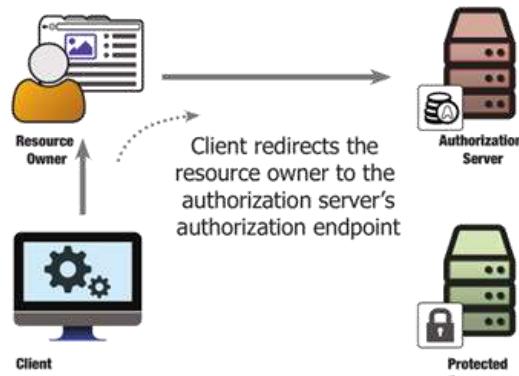
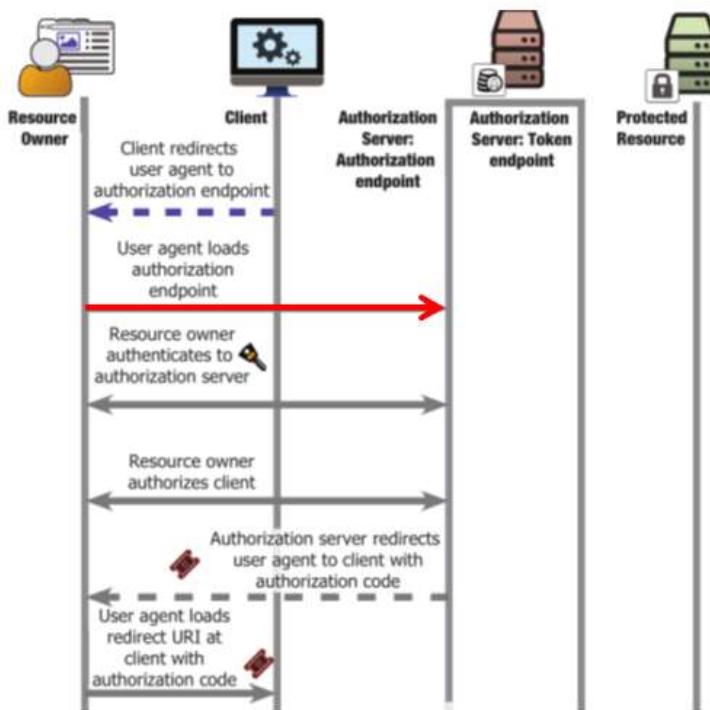
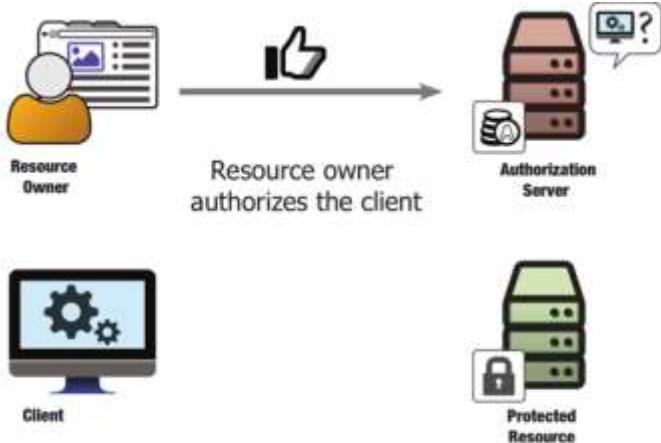


Figure 2.2 Sending the resource owner to the authorization server to start the process

```
HTTP/1.1 302 Moved Temporarily
x-powered-by: Express
Location: https://auth-server.banco.com:9001/authorize?response_type=code&scope=foo&client_id=oauth-client-1&redirect_uri=http%3A%2F%2Fclient.com%3A9000%2Fcallback&state=Lwt50DDQKUB8U7jtfLQCVGDL9cnmwH1
Vary: Accept
Content-Type: text/html; charset=utf-8
Content-Length: 444
Date: Fri, 31 Jul 2015 20:50:19 GMT
Connection: keep-alive
```



```
GET /authorize?response_type=code&scope=foo&client_id=oauth-client-1&redirect_uri=http%3A%2F%2Fclient.com%3A9000%2Fcallback&state=Lwt50DDQKUB8U7jtfLQCVGDL9cnmwHH1 HTTP/1.1
Host: auth-server.banco.com:9001
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:39.0)
Gecko/20100101 Firefox/39.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Referer: http://client.com:9000/
Connection: keep-alive
```

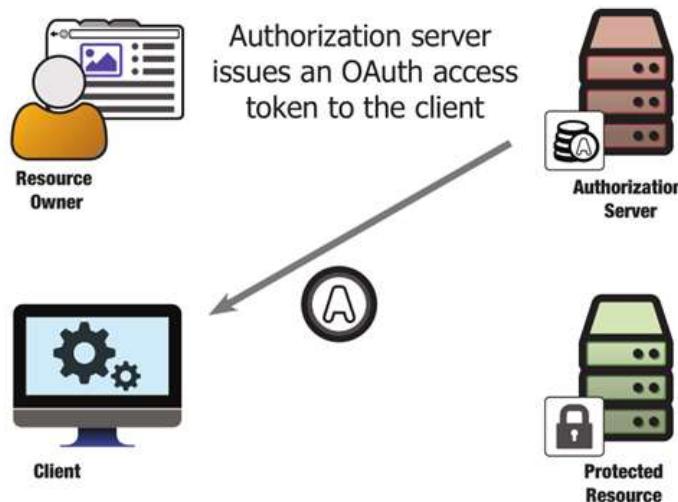


HTTP 302 Found

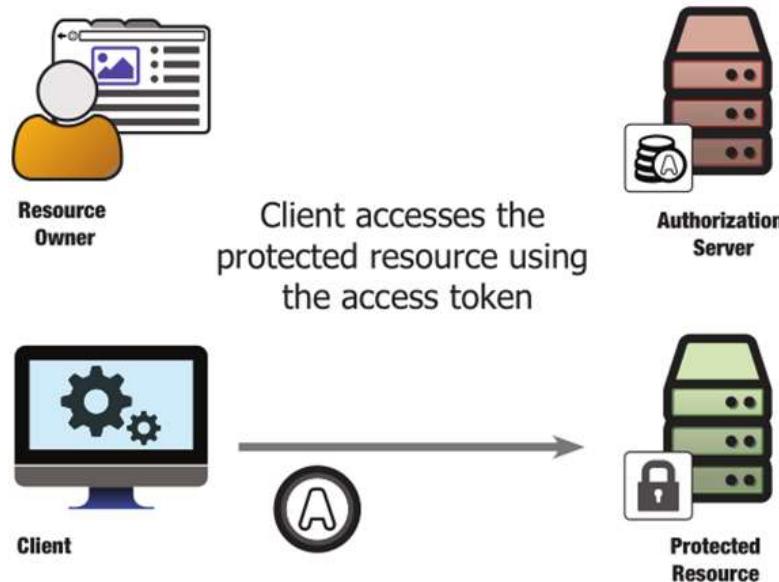
Location:

[https://client.com:9000/oauth\\_callback?code=8V1pr0rJ&state=Lwt50DDQKUB8U7jtflQCVGDl9cnmwHH1](https://client.com:9000/oauth_callback?code=8V1pr0rJ&state=Lwt50DDQKUB8U7jtflQCVGDl9cnmwHH1)

```
GET /oauth_callback?code=8V1pr0rJ&state=Lwt50DDQKUB8U7jtflQCVGDl9cnmwHH1 HTTP/1.1
Host: client.com:9000
```

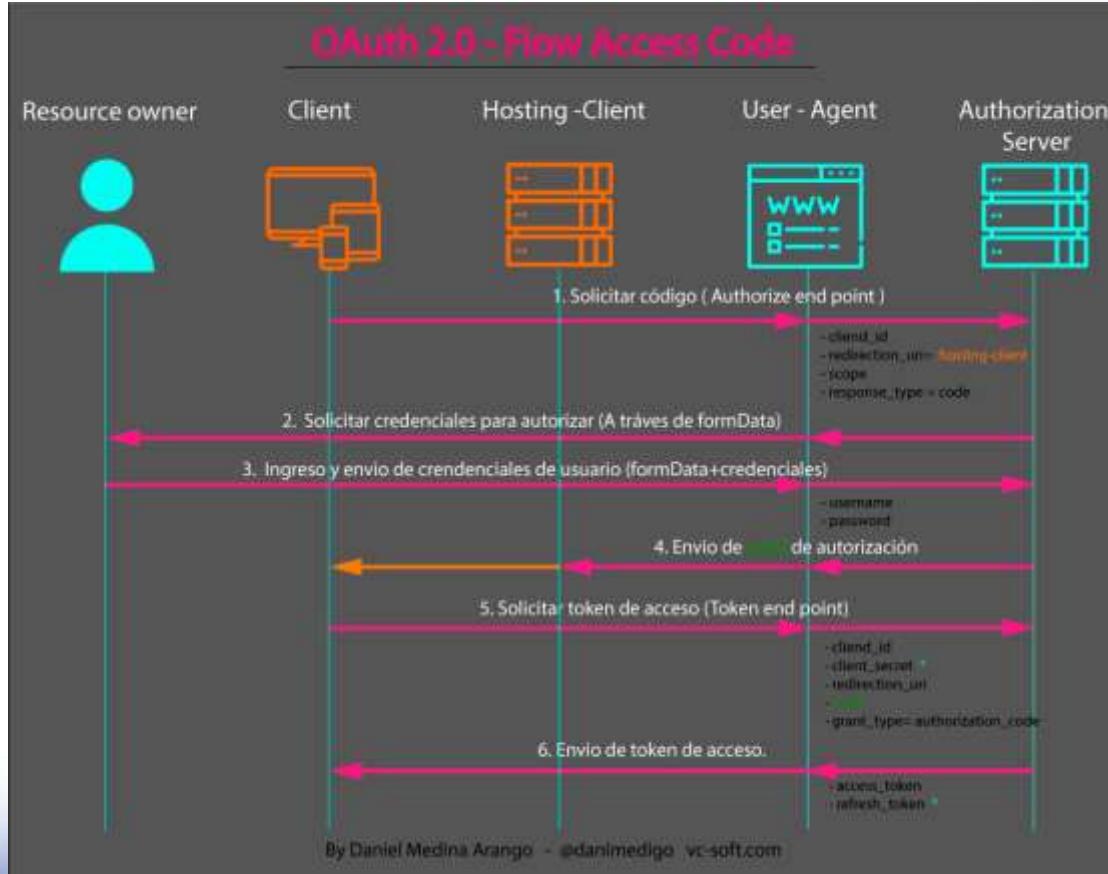


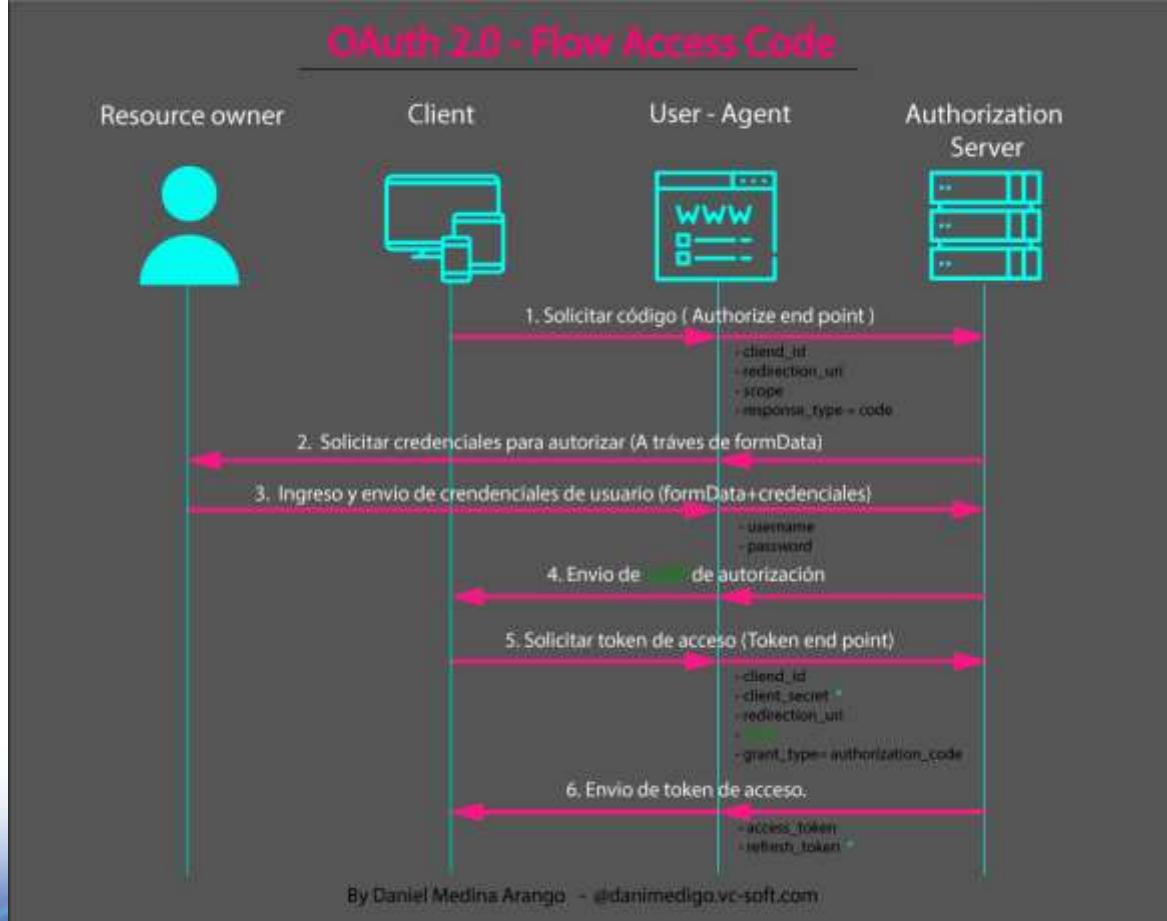
```
HTTP 200 OK
Date: Fri, 31 Jul 2015 21:19:03 GMT
Content-type: application/json
{
  "access_token": "987tghjkiu6trfghjuytrghj",
  "token_type": "Bearer"
}
```



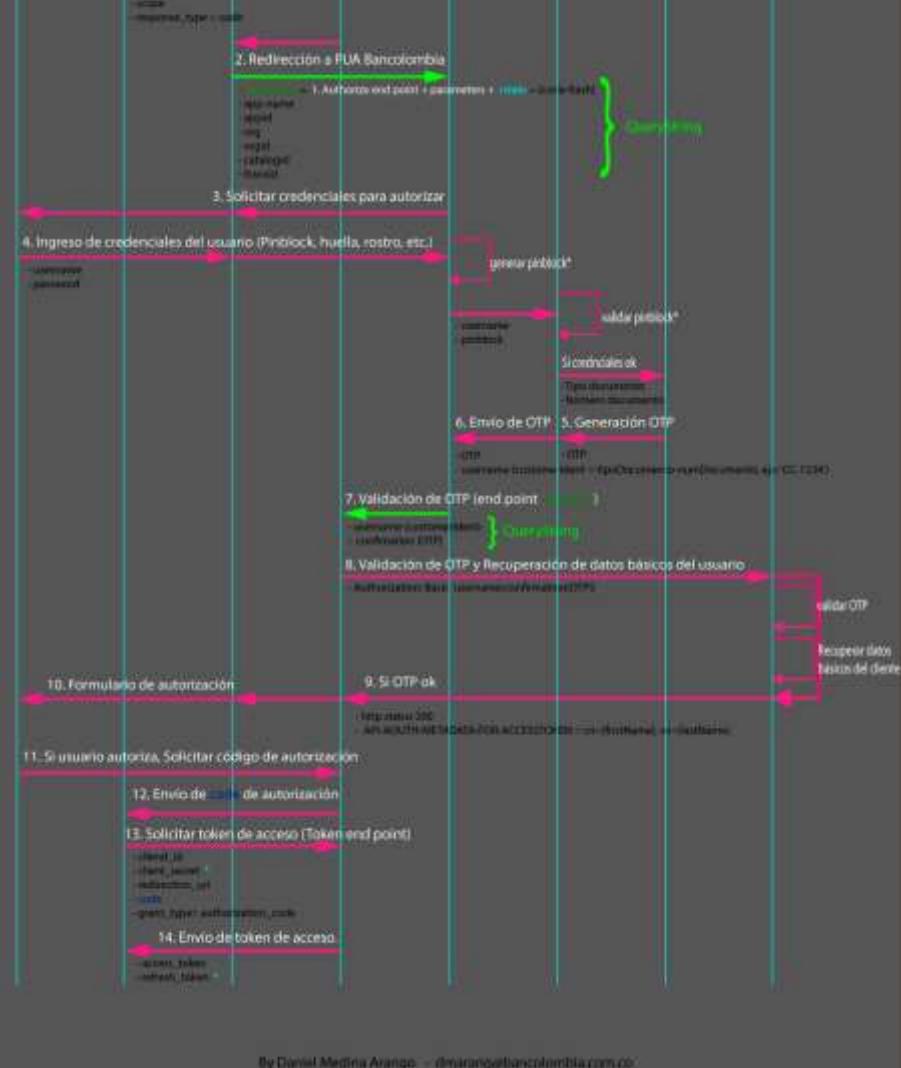
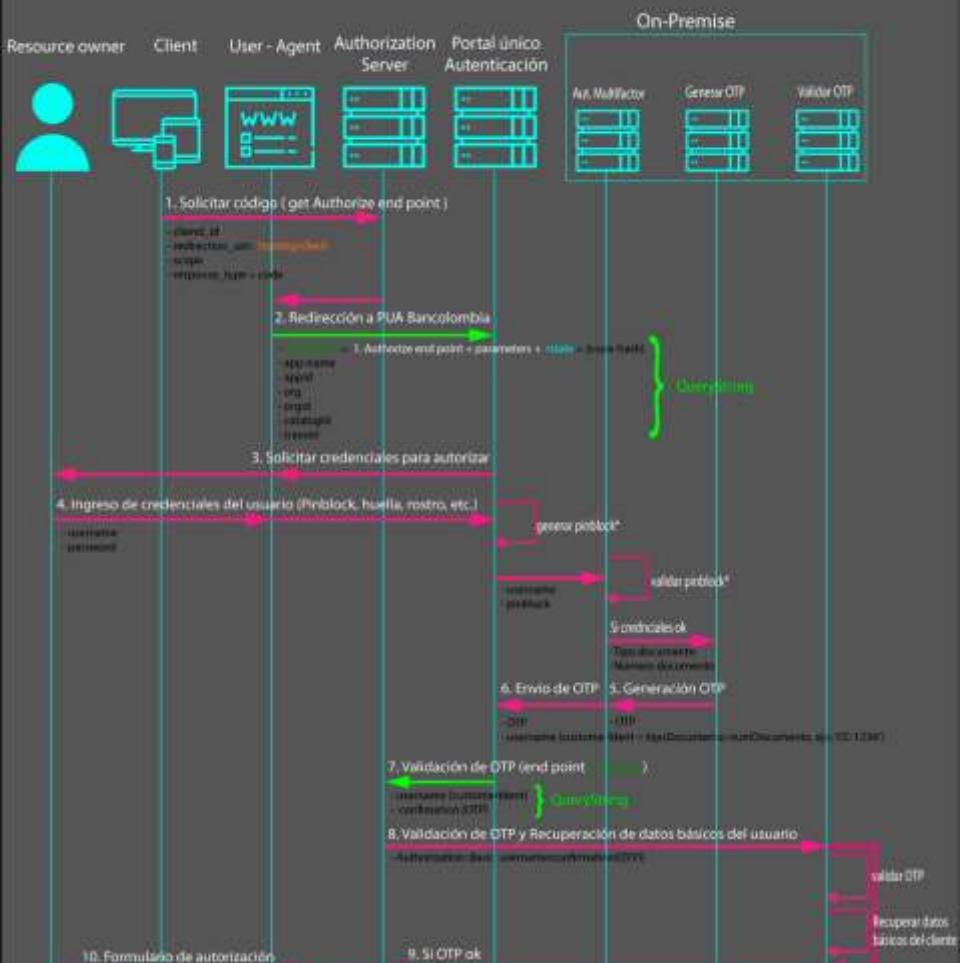
```
GET /resource HTTP/1.1
Host: resource.com:9002
Accept: application/json
Connection: keep-alive
Authorization: Bearer 987tghjkiu6trfghjuytrghj
```

# Authorization Code: IBM Access Code



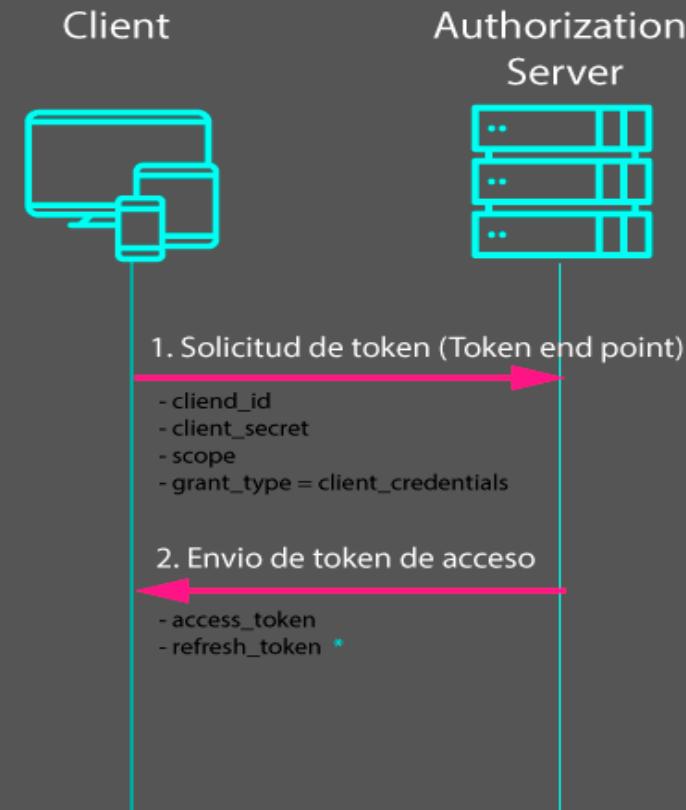


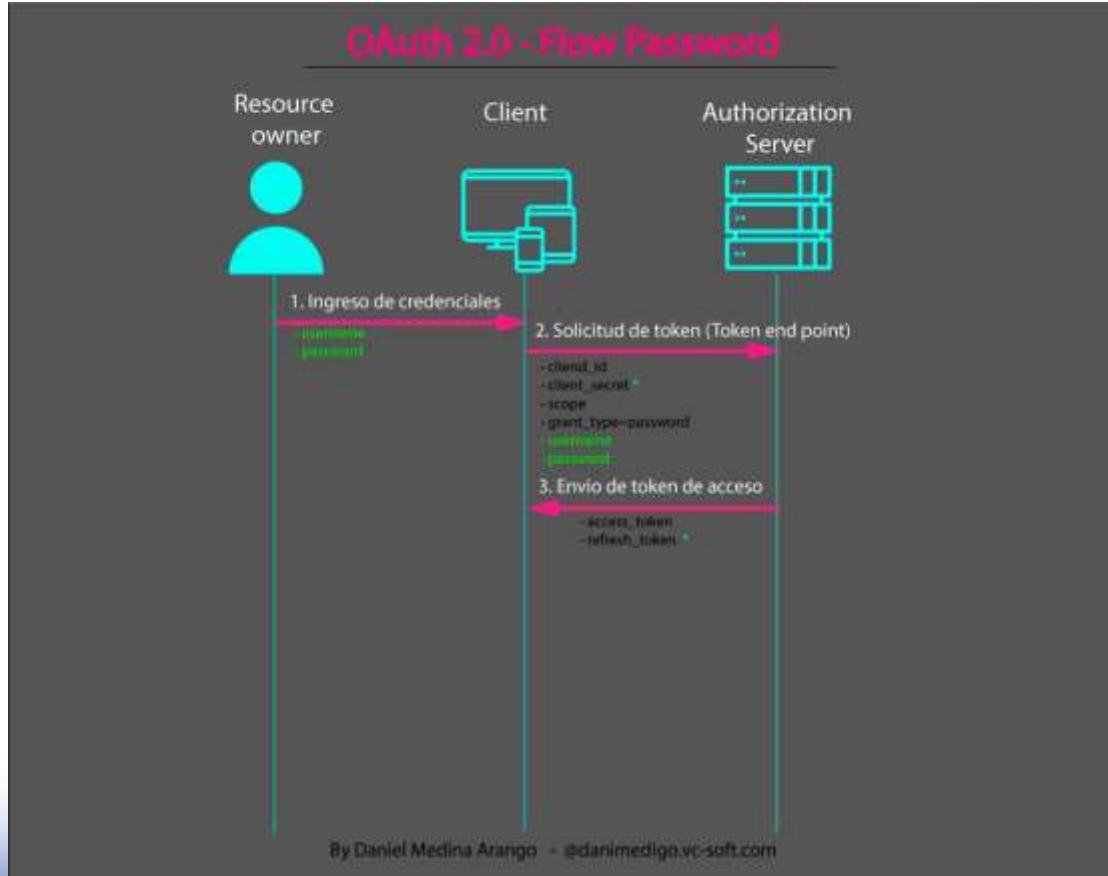
## APC - OAuth 2.0 - Access Code - form redirect (Portal único Autenticación)



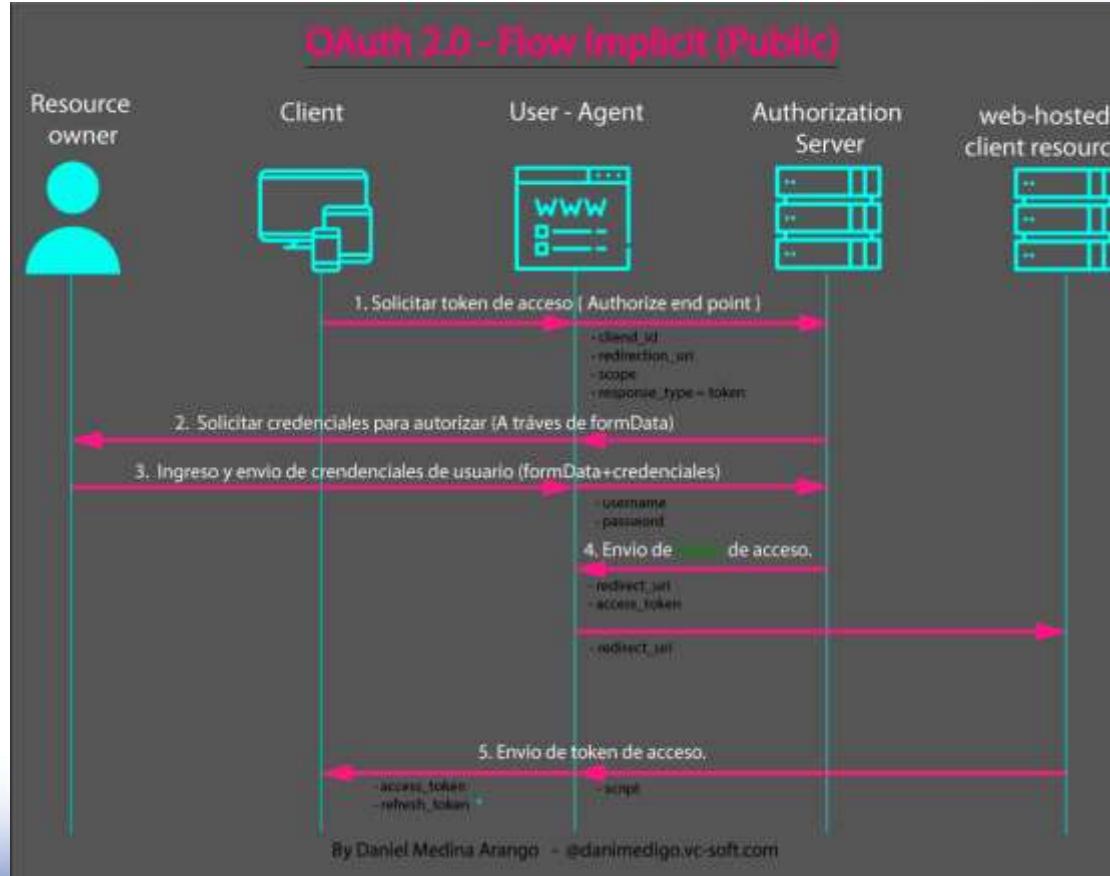
**OAuth2**  
**Otros Flujos**

## OAuth 2.0 - Flow Application





# Implicit



# OAuth 2.0

## Flujos de seguridad OAuth 2.0 - API Connect

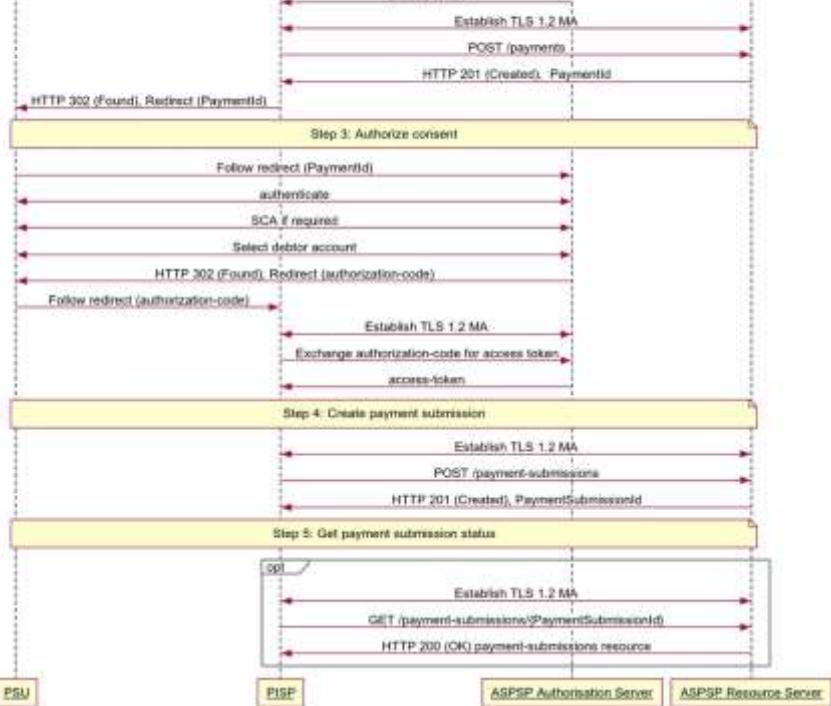
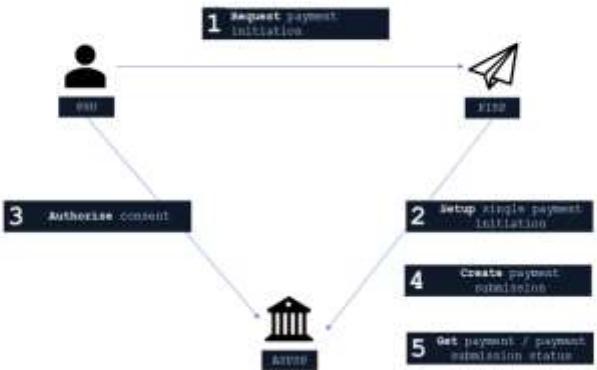
Tipo	Flujo	Grant_type	Parámetros en formData	Parámetros en QueryString	Parámetros en Header	End Point	Observaciones
Confidential	Application	client_credentials	<ul style="list-style-type: none"> <li>▪ <a href="#">scope</a></li> <li>▪ <a href="#">grant_type</a></li> </ul>		<ul style="list-style-type: none"> <li>• client_id</li> <li>• client_secret</li> </ul>	(POST) <a href="#">Token_Endpoint_Url</a> .	Se realiza la solicitud enviando en la cabecera 'Authorization' como autenticación tipo 'Basic' los datos de client_id y client_secret. Se envían en formData los demás campos.
Confidential and Public	Password	password	<ul style="list-style-type: none"> <li>▪ <a href="#">client_id (solo para Public).</a></li> <li>▪ <a href="#">username</a></li> <li>▪ <a href="#">password</a></li> <li>▪ <a href="#">scope</a></li> <li>▪ <a href="#">grant_type</a></li> </ul>		Solo aplica para tipo confidential: <ul style="list-style-type: none"> <li>• client_id</li> <li>• client_secret*</li> </ul>	(POST) <a href="#">Token_Endpoint_Url</a> .	Se realiza la solicitud para obtener el token de acceso, se envían los datos de acuerdo al flujo.
	Access code	authorization_code	<ul style="list-style-type: none"> <li>▪ <a href="#">client_id (solo para Public).</a></li> <li>▪ <a href="#">code</a></li> <li>▪ <a href="#">redirect_uri</a></li> <li>▪ <a href="#">grant_type</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">response_type*</a></li> <li>• redirect_uri</li> <li>• scope</li> <li>• client_id</li> </ul>	Solo aplica para tipo confidential: <ul style="list-style-type: none"> <li>• client_id</li> <li>• client_secret*</li> </ul>	1. (GET) <a href="#">Authorization_Endpoint_URL</a> . 2. (POST) <a href="#">Token_Endpoint_URL</a> .	Código de autorización para obtener el token de acceso.  1. Se debe realizar la solicitud del "code", a través de la <a href="#">Authorization_Endpoint_URL</a> . Envío los <a href="#">parámetros</a> en QueryString. 2. Se debe invocar el <a href="#">Token_Endpoint_URL</a> para obtener el Access token. Envío los <a href="#">parámetros</a> en header y/o formData según sea tipo Confidential o formData.
	N/A	refresh_token			<ul style="list-style-type: none"> <li>• client_id</li> <li>• client_secret*</li> <li>• grant_type</li> <li>• refresh_token</li> </ul>	(POST) <a href="#">Token_Endpoint_Url</a> .	Se hace la solicitud para renovar el token, enviando el token de renovación.
	Implicit				<ul style="list-style-type: none"> <li>• <a href="#">response_type*</a></li> <li>• <a href="#">redirect_uri</a></li> <li>• <a href="#">scope</a></li> <li>• <a href="#">client_id</a></li> </ul>	(GET) <a href="#">Authorization_Endpoint_URL</a> .	Se obtiene un token de acceso. Se invoca solamente el <a href="#">Authorization_Endpoint_URL</a> para obtener el Access token.

\* Solo necesario en tipo confidencial. Se debe utilizar la cabecera de Authorization para enviar las credenciales del cliente.

\* El valor para flujo Access code es "code" y para flujo implicit es "token".

**JOSE  
JWT**

# Ejemplo de Uso: OpenBanking UK



## REQUEST Header Value

x-fapi-financial-id
x-fapi-customer-last-logged-time
x-fapi-customer-ip-address
x-fapi-interaction-id
Authorization
Content-Type
Accept
x-idempotency-key
x-jws-signature

## RESPONSE Header Value

Content-Type
x-jws-signature
x-fapi-interaction-id
Retry-After

## Specification

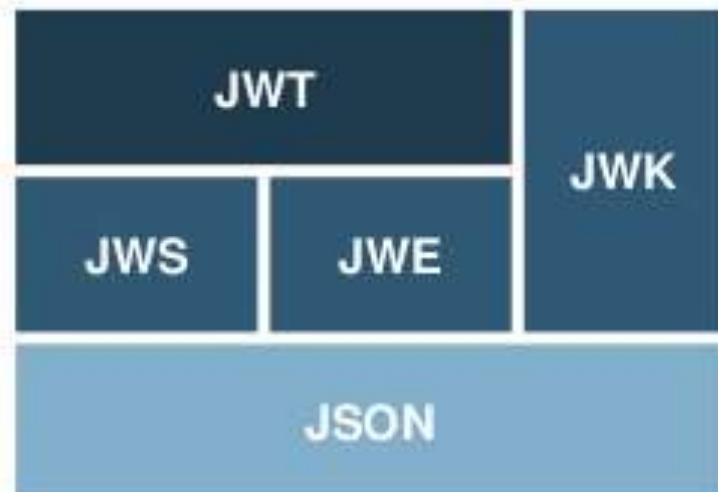
The TPP **must** sign the HTTP body of each API request that has an HTTP body. (e.g. GET requests do not have an HTTP body and are not signed).  
The ASPSP **must** sign the HTTP body of each API response that it produces which has an HTTP body.  
The ASPSP **should** verify the signature of API requests that it receives before carrying out the request. If the signature fails validation, the ASPSP  
The ASPSP **must** reject any API requests that should be signed but do not contain a signature in the HTTP header with a 400 (Bad Request) error.  
The TPP **should** verify the signature of API responses that it receives.

## Sample JOSE Header

```
{  
  "alg": "RS512",  
  "kid": "90210ABAD",  
  "b64": false,  
  "http://openbanking.org.uk/iat": 1501497671,  
  "http://openbanking.org.uk/iss": "C=UK, ST=England, L=London, O=Acme Ltd.",  
  "crit": [ "b64", "http://openbanking.org.uk/iat", "http://openbanking.org.uk/iss" ]  
}
```

- Ws-Security para JSON?
  - ❖ JSON Object Signing and Encryption (**JOSE**)
- Diferencias
  - ❖ API: info en URL (Resource Reference) o en payload
    - URL safe
      - base64url
    - Dos formas de serialización JSON y Compacta JWT
  - ❖ SOAP: info en <Envelope>

- JavaScript Object Signing and Encryption (JOSE)
  - JSON Web Signature (JWS)
    - A way of representing content secured with a digital signature or MAC using JSON data structures and base64url encoding
  - JSON Web Encryption (JWE)
    - Like JWS but for encrypting content
  - JSON Web Key (JWK)
    - JSON data structures representing cryptographic keys
  - JSON Web Algorithms
    - Defines the use of cryptographic algorithms and identifiers for JWS, JWE and JWK
- JSON Web Token (JWT)
  - A compact URL-safe means of representing claims/attributes to be transferred between two parties
  - A JWT is a JWS and/or a JWE with JSON claims as the payload



- JWT format

- ❖ Mecanismo para representar información segura incluyendo firma digital verificable (o MAC)

- ❖ Formato: **{header}.{payload}.{signature}**

- ❖ **header**

- "alg": Algorithm

- HMAC, RSA, RSA-PSS and ECDSA
    - Unsigned/none (controversy!)
    - Extensible

- "kid": Key ID
    - "jku": JWK Set URL
    - "jwk": JSON Web Key
    - "x5u": X.509 URL
    - "x5t": X.509 Thumbprint
    - "x5c": X.509 Certificate Chain
    - "typ": Type

# JSON Web Token ( JWT )

ALGORITHM HS256

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJz
dWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpvag4gR
G9lIiwiYWRtaW4iOnRydWV9.TJVA950rM7E2cBab3
0RMHrHDcEfjoYZgeFONFh7HgQ
```

Decoded EDIT THE PAYLOAD AND SECRET (ONLY HS256 SUPPORTED)

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

PAYOUT: DATA

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "admin": true  
}
```

VERIFY SIGNATURE

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  secret  
)  secret base64 encoded
```

# Componentes

## ■ JWA: JSON Web Algorithms

Internet-Draft		JSON Web Algorithms (JWA)		January 2015
alg Param Value	Digital Signature or MAC Algorithm	Implementation Requirements		
HS256	HMAC using SHA-256	Required		
HS384	HMAC using SHA-384	Optional		
HS512	HMAC using SHA-512	Optional		
RS256	RSASSA-PKCS-v1_5 using SHA-256	Recommended		
RS384	RSASSA-PKCS-v1_5 using SHA-384	Optional		
RS512	RSASSA-PKCS-v1_5 using SHA-512	Optional		
ES256	ECDSA using P-256 and SHA-256	Recommended+		
ES384	ECDSA using P-384 and SHA-384	Optional		
ES512	ECDSA using P-512 and SHA-512	Optional		
PS256	RSASSA-PSS using SHA-256 and MGF1 with SHA-256	Optional		
PS384	RSASSA-PSS using SHA-384 and MGF1 with SHA-384	Optional		
PS512	RSASSA-PSS using SHA-512 and MGF1 with SHA-512	Optional		
none	No digital signature or MAC performed	Optional		

alg Param Value	Key Management Algorithm	More Header Params	Implementation Requirements
RSA1_5	RSAES-PKCS1-V1_5	(none)	Recommended+
RSA-OAEP	RSAES OAEP using default parameters	(none)	Recommended+
RSA-OAEP-256	RSAES OAEP using SHA-256 and MGF1 with SHA-256	(none)	Optional
A128KW	AES Key Wrap with default initial value using 128 bit key	(none)	Recommended
A192KW	AES Key Wrap with default initial value using 192 bit key	(none)	Optional
A256KW	AES Key Wrap with default initial value using 256 bit key	(none)	Recommended
dir	Direct use of a shared symmetric key as the CEK	(none)	Recommended
ECDH-ES	Elliptic Curve Diffie-Hellman Ephemeral Static key agreement using Concat KDF	"epk", "apu", "apv"	Recommended+
ECDH-ES+A128KW	ECDH-ES using Concat KDF and CEK wrapped with "A128KW"	"epk", "apu", "apv"	Recommended
ECDH-ES+A192KW	ECDH-ES using Concat KDF and CEK wrapped with "A192KW"	"epk", "apu", "apv"	Optional
ECDH-ES+A256KW	ECDH-ES using Concat KDF and CEK wrapped with "A256KW"	"epk", "apu", "apv"	Recommended
A128GCMKW	Key wrapping with AES GCM using 128 bit key	"iv", "tag"	Optional

<https://tools.ietf.org/html/draft-ietf-jose-json-web-algorithms-40>

# Componentes

## JWK: JSON Web Key

```
{  
  "kty": "EC",  
  "kid": "bilbo.baggins@hobbiton.example",  
  "use": "sig",  
  "crv": "P-521",  
  "x": "AHKZLLOsCOzz5cY97ewNUajB957y-C-U88c3v13nmGZx6sYl_oJXu9  
    A5RkTKqjqvjyekWF-7ytDyRXYgCF5cj0Kt",  
  "y": "AdymlHvOiLxXkEhayXQnNCvDX4h9htZaCJN34kfmC6pV50hQHiraVy  
    SsUdaQkAgDPrwQrJmbnX9cwlGfP-HqHZR1"  
}
```

Figure 1: Elliptic Curve P-521 Public Key

```
{  
  "kty": "oct",  
  "kid": "e18c0ae5-4d9b-471b-bfd6-eef314bc7037",  
  "use": "sig",  
  "alg": "HS256",  
  "k": "h3txIZ2uNSNkbQfbtTNWbpdmhkVBF3G-Onbc6mxCcYg"  
}
```

Figure 5: HMAC SHA-256 Symmetric Key

```
{  
  "kty": "EC",  
  "kid": "bilbo.baggins@hobbiton.example",  
  "use": "sig",  
  "crv": "P-521",  
  "x": "AHKZLLOsCOzz5cY97ewNUajB957y-C-U88c3v13nmGZx6sYl_oJXu9  
    A5RkTKqjqvjyekWF-7ytDyRXYgCF5cj0Kt",  
  "y": "AdymlHvOiLxXkEhayXQnNCvDX4h9htZaCJN34kfmC6pV50hQHiraVy  
    SsUdaQkAgDPrwQrJmbnX9cwlGfP-HqHZR1",  
  "d": "AAhRON2r9cqXX1hg-RoI6R1tX5p2rUAYdmpHZoC1XNM56KtscriX6zb  
    KipQrCW9CGZH3T4ubpnoTKLDYJ_ff3_rJt"  
}
```

Figure 2: Elliptic Curve P-521 Private Key

# Componentes

## ■ JWS: JSON Web Signature

It's a dangerous business, Frodo, going out your door. You step onto the road, and if you don't keep your feet, there's no knowing where you might be swept off to.

SXTigJlzIGEgZGFuZ2Vyb3VzIGJ1c2luZXNzLCBGcm9kbywgZ29pbmcgb3V0IH  
lvdXIgZG9vci4gWW91IHN0ZXAgb250byB0aGUgcm9hZCwgYW5kIGlmIHlvdSBk  
b24ndCBzZWVwIHlvdXIgZmVldCwgdGhlcmXigJlzIG5vIGtub3dpbmcd2hlc  
UgeW91IG1pZ2h0IGJIIHNSZXB0IG9mZiB0by4

```
{"alg":"RS256",  
 "kid":bilbo.baggins@hobbiton.example  
}
```

JWS Protected Header JSON

eyJhbGciOiJSUzI1NiIsImtpZCI6ImJpbGJvLmJhZ2dpbnNAaG9iYml0b24uZX  
hhbXBsZSJ9

eyJhbGciOiJSUzI1NiIsImtpZCI6ImJpbGJvLmJhZ2dpbnNAaG9iYml0b24uZX  
hhbXBsZSJ9  
.  
SXTigJlzIGEgZGFuZ2Vyb3VzIGJ1c2luZXNzLCBGcm9kbywgZ29pbmcgb3V0IH  
lvdXIgZG9vci4gWW91IHN0ZXAgb250byB0aGUgcm9hZCwgYW5kIGlmIHlvdSBk  
b24ndCBzZWVwIHlvdXIgZmVldCwgdGhlcmXigJlzIG5vIGtub3dpbmcd2hlc  
UgeW91IG1pZ2h0IGJIIHNSZXB0IG9mZiB0by4

Figure 11: JWS Signing Input

UTF-8 y base64

"RS256" = (RSASSA- PKCS1-v1\_5 with SHA-256)

eyJhbGciOiJSUzI1NilsImtpZCI6ImJpbGJvLmJhZ2dpbnNAaG9iYml0b  
24uZXhhbXBsZSJ9

base64

MRjdkly7\_-oTPTS3AXP4liQIGKa80A0ZmTuV5MEmaHoxnW2e5CZ5N1KtainoFmK  
ZopdHM1O2U4mwzJdQx996ivp83xuglII7PNDi84wnB-BDkoBwA78185hX-Es4J  
IwmDLJK31fWRa-XtL0RnltuYv746iYTh\_qHRD68BNt1uSNCrUCTJDt5aAE6x8w  
W1Kt9eRo4QPocSadnHXFxnt8Is9UzpERV0ePPQdLuW3IS\_de3xyIrDaLGdj1uP  
xAUh6L2aXic1U12podGU0KLUQSE\_oI-ZnmKJ3F4uOZDnd6QZWJushZ41Axf\_f  
cIe8u9ipH84ogoree7vjbU5y18kDquDg

Figure 12: JWS Signature, base64url-encoded  
"RS256" (RSASSA- PKCS1-v1\_5 with SHA-256) y base64

## JWS (2):

### Compact Serialization

```
eyJhbGciOiJSUzI1NiIsImtpZCI6ImJpbGJvLmJhZ2dpbnNAaG9iYml0b24uZX
hhbXBsZSJ9
.
SXTigJlzIGEgZGFuZ2Vyb3VzIGJ1c2luZXNzLCB Gcm9kbywgZ29pbmcgb3V0IH
lvdXIgZG9vci4gNW91IHN0ZXAgb250byB0aGUgcm9hZCwgYW5kIGlmIHlvdSBk
b24ndCBzWVwI HlvdXIgZmVldCwgdGhlcmXigJlzIG5vIGtub3dpbmcd2hlc
UgeW91IG1pZ2h0IGJ1IHN3ZX B0IG9mZiB0by4
.
MRjdkly7_-oTPTS3AXP41iQIGKa80A0ZmTuV5MeAhoxnW2e5CZSN1KtainoFmK
ZopdHM1O2U4mwzJdQx996ivp83xuglII7PNDi84wnB-BDkoBwA78185hX-Es4J
IwmDLJK31fWRa-XtL0RnltuYv746iYTh_qHRD68BNt1uSNCrUCTJDt5aAE6x8w
W1Kt9eRo4QPocSadnHXFxnt8Is9UzpERV0ePPQdLuW3IS_de3xyIrDaL GdjluP
xUAhb6L2aXic1U12podGU0KLUQSE_oI-ZnmKJ3F4uOZDnd6QZWJushZ41Axf_f
cIe8u9ipH84ogoree7vjbU5y18kDquDg
```

Figure 13: JWS Compact Serialization

### JSON Serialization

```
{
  "payload": "SXTigJlzIGEgZGFuZ2Vyb3VzIGJ1c2luZXNzLCB Gcm9kbywg
Z29pbmcgb3V0IHlvdXIgZG9vci4gNW91IHN0ZXAgb250byB0aGUgcm9h
ZCwgYW5kIGlmIHlvdSBkb24ndCBzWVwI HlvdXIgZmVldCwgdGhlcmXi
gJlzIG5vIGtub3dpbmcd2hlc UgeW91IG1pZ2h0IGJ1IHN3ZX B0IG9m
ZiB0by4",
  "signatures": [
    {
      "protected": "eyJhbGciOiJSUzI1NiIsImtpZCI6ImJpbGJvLmJhZ2
        dpbnNAaG9iYml0b24uZXhhbXBsZSJ9",
      "signature": "MRjdkly7_-oTPTS3AXP41iQIGKa80A0ZmTuV5MeAh
        oxnW2e5CZSN1KtainoFmKZopdHM1O2U4mwzJdQx996ivp83xuglII
        7PNDi84wnB-BDkoBwA78185hX-Es4J IwmDLJK31fWRa-XtL0Rnlt
        uYv746iYTh_qHRD68BNt1uSNCrUCTJDt5aAE6x8wW1Kt9eRo4QPo
        cSadnHXFxnt8Is9UzpERV0ePPQdLuW3IS_de3xyIrDaL GdjluPxU
        Ahb6L2aXic1U12podGU0KLUQSE_oI-ZnmKJ3F4uOZDnd6QZWJush
        Z41Axf_f cIe8u9ipH84ogoree7vjbU5y18kDquDg"
    }
  ]
}
```

Figure 14: General JWS JSON Serialization

# Ejemplo

```
{ [
  "payload": "eyJpc3MiOiJqb2UiLA0KICJleHAIojEzMDA4MTkzOD",
  "signatures": [
    {
      "protected": "eyJhbGciOiJSUzI1NiJ9",
      "header": {
        "kid": "2014-06-29"
      },
      "signature": "cC4hiUPoj9Eetdgtv3hF80EGrhkB"
    },
    {
      "protected": "eyJhbGciOiJFUzI1NiJ9",
      "header": {
        "kid": "e909097a-ce81-4036-9562-d21d2992db0d"
      },
      "signature": "DtEhU3ljbEg8L38VWAfUAqOyKAM"
    }
  ]
}
```

The JWS payload not necessarily needs to be a JSON payload,  
it can be of any content type

- Formato

{header}.{EncryptedKey}.{InitializationVector}.{CipherText}.{AuthenticationTag}

## Headers

- "alg": Algorithm (key wrap or agreement)
- "enc": Encryption Method (Authenticated Encryption only)
- "zip": Compression Algorithm
  - "DEF" for the DEFLATE Compressed Data Format from RFC 1951 is currently the only one
- "kid", "jku", "jwk", "x5u", "x5t", "x5c", etc..

Content Encryption Algorithm	JWE "enc" Parameter Values
Authenticated encryption with Advanced Encryption Standard (AES) in Galois/Counter Mode (GCM)	A128GCM, A192GCM and A256GCM
Authenticated encryption with an AES-CBC and HMAC-SHA2 composite	A128CBC-HS256, A192CBC-HS384 and A256CBC-HS512

Key Management Algorithm	JWE "alg" Parameter Values
Direct encryption with a shared symmetric key	dir
RSAES-PKCS1-V1_5 key encryption	RSA1_5
RSAES using OAEP key encryption	RSA-OAEP and RSA-OAEP-256
AES key wrap	A128KW, A192KW and A256KW
AES GCM key encryption	A128GCMKW, A192GCMKW and A256GCMKW
Elliptic Curve Diffie-Hellman Ephemeral Static key agreement using Concat KDF	ECDH-ES
Elliptic Curve Diffie-Hellman Ephemeral Static key agreement using Concat KDF with AES key wrap	ECDH-ES+A128KW, ECDH-ES+A192KW and ECDH-ES+A256KW
PBES2 with HMAC SHA-2 and AES key wrapping	PBES2-HS256+A128KW, PBES2-HS384+A192KW and PBES2-

## ■ JSON Web Encryption

### JWS Compact Serialization (String)

```
BASE64URL(UTF8(JWS Protected Header)) || '.' ||
BASE64URL(JWS Payload) || '.' ||
BASE64URL(JWS Signature)
```

### JWS JSON Serialization (JSON object)

```
{
  "payload": "<payload contents>",
  "signatures": [
    {"protected": "<integrity-protected header 1 contents>",
     "header": "<non-integrity-protected header 1 contents>",
     "signature": "<signature 1 contents>"},
    ...
    {"protected": "<integrity-protected header N contents>",
     "header": "<non-integrity-protected header N contents>",
     "signature": "<signature N contents>"}]
```

# Componentes

## JWE: JSON Web Encryption

```
{  
  "kty": "RSA",  
  "kid": "Frodo.baggins@hobbiton.example",  
  "use": "enc",  
  "n": "maxhbnsBt0d3CNrKuprUE6n91YcregDNLXhieTAwclJ8HnPU9XIYegT  
  HVHqjxKDSiP21-F5j7spg61wgdaq2yhrnvXhYlvcM7RfgKxnNx_xAHx  
  6f3y7s-MHPSNEuPC21hGUAKRAI0EHv9IeyPMp141BlOp9t5fS9m5U  
  MraAllhrd-osQGPjleLldeTwx-ZTHu3C60Pu_L1I6hKrn9bwuJa4c  
  RSBDzppgbay7ASgsjCubtYiaNITHScuHxprUdJZKUfaMzV8NOKPA60P14oy  
  p8adjvM242Azb3BnXaS5sEZhaueTxv284eZOAjIyh2e_VoIKV9msnDr7VA  
  VotGlu9Q",  
  "e": "AQAB",  
  "d": "Kn9tgohf1TVi8uPh5b9TnwyHuG5dK6RE0uFd1pCgnJN7ZE1963R7w  
  bQ1pLAHmpIbhTztfrheoAniRv1NCiqXaH_qS461xiDTpdntEPmqckSyO  
  5jMAj17-CLBvhpYYounfV2esgMoVaPRMYT9TW63hNmBaIs7USZ_hlg6  
  OemV9vItIT3fucjSM86NffFa0lE0n3r2fppgPG6lrd6fpLc90eq-qEP  
  1GFULInRdnDe-PBq8kWn3KHINATEgrQAgTTgBB5-3VD0Fgkfgbn1PW  
  miuXo080p19KDIrr_wcc6fg14ns9Ke6REsvhGPHf2afjHg5y_Fd2v  
  pzj85bQO",  
  "p": "20wQm243f0TmQ8IKuJU595Eh2ni2ZAS5h172MinUE3sdTVKSLtA  
  eekX9vbZuluXhdVMhM6nKCI_2iNk8ZBavLYHLB_G21aXf9-unynEpJsh  
  7HHTk1LpVAz00x12dVlijoxAduNn3hiEfJrjZL265710H-aJQQ1DQoQo103  
  2vFnJ",  
  "q": "te6LY4-w7IyaqhM1ExujjMpkTAT1TeRbv0VQnFLY2xInhrhIdw1Q93_V  
  F099aPIESe1jaZnu-61K1e-q17mtCPozKfVtUfYz5HJ_XY2kflexJ1N  
  91h2H9w5piskZpeI5-GPHCC6gR1ko1q_idn_qxyusfvw7Wx1ISVFQfK8  
  dEtB",  
  "dp": "UfYKcl_ord92vC0PzwLSp1hg4l3-25wl48wiwbpzOyTgd2xHTH  
  QmjpFA1ZBq-zf9Rmg3XkDrFs9rkdxPtAsL1kYdeCT5c125fkdg317JV  
  RDoilnX7x2Kdh8ERCrer4zXItuT1_KsX7NU51vMQj4bTw2eTx11psf  
  loOrYu",  
  "dq": "1EGc0-QFpepdH8Fwd7miFyeXdm0kXJBCogChY6YKuJH6c_p8Le9Mb  
  pFKESzExALN11ehf3B6eGB151z_ayU12t0q282znolpa#FvYmNt827A  
  CfzI07qNh7RIPader-103tkvXAaBau_9vs5rs-7#HtskVrxSlvYJ14  
  TkXjHE",  
  "qi": "kc-1z20qofaZCr510t0vtREKsVqaAvhQiqaTq6L-Mj5dsCmRxx6vZ  
  1XYYxRTE1n_Aaqjqajk1jeGlxTTthD81gaoFgBlaAr5uR1hQpqsC7  
  G17CF1D2kB7MTQn6EshYzfxm88eI0896Rzuuh0eLef09mkDcIyPrBxx  
  2b0_nM"
```

Figure 73: RSA 2048-Bit Key, in Jwk Format

- "RSA1\_5" = (RSAES-PKCS1-v1\_5) key encryption algorithm
- "A128CBC-HS256" = (AES-128-CBC-HMAC-SHA-256) content encryption algorithm.

3qyTVhIwt5juqZUKpf8qpvauuB956ME3L2Rt-8qXK5o

Figure 74: Content Encryption Key, base64url-encoded

AES symmetric key as the Content Encryption Key (CEK)

bdb5sTkVwhATqfHsxB0DayA

Figure 75: Initialization Vector, base64url-encoded

laLxI0j-nLH-\_BgLOXMozKxmy9gffy2gTdqvzfTih]Buuxg0V7yk1wClnQePF  
vG2K-pvSlWc9BRlazDrn50RcRai\_3TDON395H3c62tIouJJ4XaRvYHFjZT2G  
Xfz8YAImc91Tfk0WXC2F5Xbb71ClQ1DDH151tlph77f2ff7xiSxh9oSewYrcG  
TSLUeeCt36r1Kt3OSj7EyBQXoZ1N7IxbyhMAfgIe7Mv1rOT0I5I8NQqeXXW8V1  
zNmoxaGMy3YnGir5Wf6Qt2nBq4qDaPdnaAuuGUGEeceli01wx1BpyIfgvfj0h  
MBs9M8XL223Fg47x1GsMXdfuY-4jaqVw

Figure 76: Encrypted Key, base64url-encoded

Performing the key encryption operation over the CEK (Figure 74) with the RSA key (Figure 73) results in the following Encrypted Key:

## ■ JWE(2)

```
{
  "alg": "RSA1_5",
  "kid": "frodo.baggins@hobbiton.example",
  "enc": "A128CBC-HS256"
}
```

Figure 77: JWE Protected Header JSON

eyJhbGciOiJSU0ExXzUiLCJraWQiOiJmcm9kby5iYWNraW5zQGhvYmJpdG9uLmV4Ym1wbGU1LCJlbmMiOiJBMTI4Q0JDLUHTMjU2In0

Figure 78: JWE Protected Header, base64url-encoded

"You can trust us to stick with you through thick and "  
 "thin\xe2\x80\x93 to the bitter end. And you can trust us to "  
 "keep any secret of yours\xe2\x80\x93 closer than you keep it "  
 "yourself. But you cannot trust us to let you face trouble "  
 "alone, and go off without a word. We are your friends, Frod

Figure 72: Plaintext Content



- o Cek (Figure 74);
- o Initialization Vector (Figure 75); and
- o The Protected Header (Figure 77) as authenticated data

0fys\_TY\_na7F8du5fXLiYdHaA2dxUjD67ieF7fcVbI862jhJvG24\_FMVsiGc\_r  
 aa0HnLQ6s1P2sv3Xz1lpll\_o5wR\_Rs5zr582-wmI3jvo0mkpEEnlDm2vDu\_k80  
 WzJv7eZVEqjNkdjVzFhpPiyoQU286LopRc2vbVbK4d0KPdMTjPPEm#qcaGeTMZV  
 yeSuvf5k59yJ2xRu5vWFf6KrNtmRdZ8R4mD0jHSrM\_s8uwIFcqjt4r5GX8TKai0  
 zT5CbL5Qlw3sRc7u\_hg@yKvDiRytEAEs3vZkcflkP6nbXdc\_PkMdhiS-ohP78T2  
 06\_7uInMghFeX4ctHg7Ve1Hg1t93JHdEQ15\_V9UN1rhXlrYu-0fVMkZAKX3W  
 i7lxA6BP430m

Figure 79: Ciphertext, base64url-encoded

kvKuFBXHe5mQr4lqgojbAUG

Figure 80: Authentication Tag, base64url-encoded

Ciphertext

Authentication Tag

## JWE (3): Compact Serialization

```

eyJhbGciOiJSU0ExXzUiLCJraWQiOiJmcm9kby5iYWdnaw5zQGhvYmJpdG9uLm
V4YW1wbGUiLCJ1bmMiOiJBMTI4Q0JDLUhTMjU2In0
.
laLxI0j-nLH-_BgLOXMoKxmy9gffy2gTdvqzfTihJBuuzxg0V7yk1wClnQePF
vG2K-pvSlWc9BRIazDrn50RcRai__3TDON395H3c62tIouJ4xaRvYHFjZTZ2G
Xfz8YAImc91Tfk0WCXC2F5Xbb71ClQ1DDH151tlpH77f2ff7xiSxh9oSewYrcG
TSLUeeCt36r1Kt30sj7EyBQXoZ1N7IxbyhMAfgIe7Mv1rOT0I5I8NQqeXXw8V1
zNmoxaGMny3YnGir5Wf6Qt2nBq4qDaPdnaAuuGUGEeceli01wx1BpyIfgvfjOh
MBs9M8XL223Fg47x1GsMXdfuY-4jaqVw

bbd5sTkYwhAIqfHsx8DayA
.

0fys_TY_na7f8dwSfXLiYdHaA2DxUjD67ieF7fcVbIR62JhJvGZ4_FNVSiGc_r
aa0HnLQ6s1P2sv3Xzl1p1l_o5wR_RsSzsR8Z-wnI3Jvo0mkpEEEn1DmZvDu_k80
WzJv7eZVEqiWKdyVzFhPpiyQU28GLOpRc2VbVbK4dQKPdNTjPPEmRqcaGeTWZV
yeSuVf5k59yJZxRuSvWFF6KrNtmRdZ8R4mD0jHSrM_s8uwIFcq4r5GX8TKaI0
zT5CbL5Qlw3sRc7u_hg0yKV0iRytEAEs3vZkcflkP6nbXdc_PkMdNS-ohP78T2
06_7uInMGhFeX4ctHG7Ve1HGiT93JfWDEQi5_V9UN1rhXNrYu-0fVmKZAKX3VW
i7lzA6BP430m
.

kvKuFBXHe5mQr4lqgobAUg

```

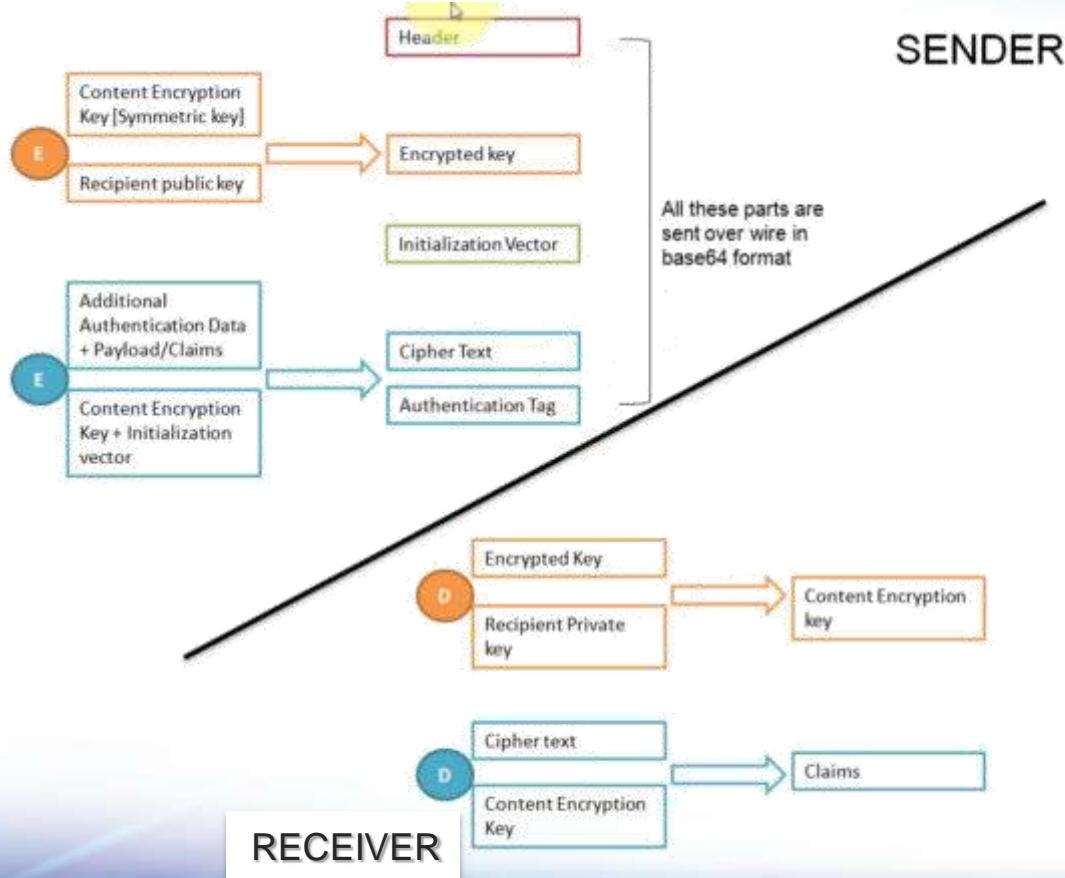
Figure 81: JWE Compact Serialization

## JSON Serialization

```
{
  "recipients": [
    {
      "encrypted_key": "laLxI0j-nLH-_BgLOXMoKxmy9gffy2gTdvqzf
TihJBuuzxg0V7yk1wClnQePFvG2K-pvSlWc9BRIazDrn50RcRai_
_3TDON395H3c62tIouJ4xaRvYHFjZTZ2GXfz8YAImc91Tfk0WX
C2F5Xbb71ClQ1DDH151tlpH77f2ff7xiSxh9oSewYrcGTSLUeeCt
36r1Kt30sj7EyBQXoZ1N7IxbyhMAfgIe7Mv1rOT0I5I8NQqeXXw8V1
V1zNmoxaGMny3YnGir5Wf6Qt2nBq4qDaPdnaAuuGUGEeceli01wx
1BpyIfgvfjOhMBs9M8XL223Fg47x1GsMXdfuY-4jaqVw"
    }
  ],
  "protected": "eyJhbGciOiJSU0ExXzUiLCJraWQiOiJmcm9kby5iYWdnaw
5zQGhvYmJpdG9uLmV4YW1wbGUiLCJ1bmMiOiJBMTI4Q0JDLUhTMjU2In
0",
  "iv": "bbd5sTkYwhAIqfHsx8DayA",
  "ciphertext": "0fys_TY_na7f8dwSfXLiYdHaA2DxUjD67ieF7fcVbIR62
JhJvGZ4_FNVSiGc_r_aa0HnLQ6s1P2sv3Xzl1p1l_o5wR_RsSzsR8Z-wn
I3Jvo0mkpEEEn1DmZvDu_k80WzJv7eZVEqiWKdyVzFhPpiyQU28GLOpRc
2VbVbK4dQKPdNTjPPEmRqcaGeTWZVyeSuVf5k59yJZxRuSvWFF6KrNtm
RdZ8R4mD0jHSrM_s8uwIFcq4r5GX8TKaI0zT5CbL5Qlw3sRc7u_hg0y
KV0iRytEAEs3vZkcflkP6nbXdc_PkMdNS-ohP78T206_7uInMGhFeX4c
tHG7Ve1HGiT93JfWDEQi5_V9UN1rhXNrYu-0fVmKZAKX3VWi7lzA6BP4
30m",
  "tag": "kvKuFBXHe5mQr4lqgobAUg"
}
```

Figure 82: General JWE JSON Serialization

```
{ [
  "protected": "eyJlbmMiOiJBMTI4Q0JDLUhTMjU2In0",
  "unprotected": [
    "jku": "https://server.example.com/keys.jwks"
  ],
  "recipients": [
    {
      "header": {
        "alg": "RSA1_5",
        "kid": "2011-04-29"
      },
      "encrypted_key": "UGhIOguC7IuEvf_NPVaXsGMoL0mwvc1GyqlI9XShH59_i8J0PH5ZZyNfGy2xGd"
    },
    {
      "header": {
        "alg": "A128KW",
        "kid": "7"
      },
      "encrypted_key": "6KB707dM9YTigHtLvtgWQ8mKwboJW3of9locizkDTHzBC2IlrT1o0Q"
    }
  ],
  "iv": "AxY8DCtDaGlsbGljb3RoZQ",
  "ciphertext": "KDlTtXchhZTGufMYm0YGS4HffxPSUrfrmqCHXaI9wOGY",
  "tag": "Mz-VPPyU4RlcuYv1IwIvzw"
}]
```



# OpenId Connect



# OpenIdConnect OIDC

- OpenID Connect (OIDC) is built on top of the OAuth 2.0 protocol and focuses on identity assertion.
  - ❖ OIDC provides a flexible framework for identity providers to validate and assert user identities for Single Sign-On (SSO) to web, mobile, and API workloads.
- [https://raw.githubusercontent.com/ibm-apiconnect/openid/master/oidc\\_1.0.0.yaml](https://raw.githubusercontent.com/ibm-apiconnect/openid/master/oidc_1.0.0.yaml)
  - ❖ Scopes: openid
- ID Token:
  - ❖ The primary extension that OpenID Connect makes to OAuth 2.0 to enable End-Users to be Authenticated is the ID Token data structure

```
{ "token_type": "bearer",
  "access_token": "<sanitized>",
  "expires_in": 3600,
  "scope": "weather openid",
  "refresh_token": "<sanitized>",
  "id_token": "<sanitized>"
}
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
    "access_token": "S1AV32hkKG",
    "token_type": "Bearer",
    "refresh_token": "8xLOxBtZp8",
    "expires_in": 3600,
    "id_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWdkazcifQ.ewogImlzc
        yI6ICJodHRwOi8vc2VydVmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ4Mjg5
        NzYxMDAxIiwKICJhdWQiOiAiczzCaGRSa3F0MyIsCiAibm9uY2UiOiAibi0wUzz
        fV3pBMk1qIiwKICJleHAIoiAxMzExMjgxOTcwLAogImlhdCI6IDEzMTEyODA5Nz
        AKfQ.ggW8hz1EuVLuxNuuiJKX_V8a_OMXzR0EHR9R6jgdqrOOF4dagU96Sr_P6q
        Jp6IcmD3HP99Obi1PRs-cwh3LO-p146waJ8IhehcwL7F09JdijmBqkvPeB2T9CJ
        NqeGpe-gccMg4vfKjkM8FcGvnzZUN4_KSP0aAp1tOJ1zZwgjxqGByKHiOtX7Tpd
        QyHE5lcMiKPXFElQILVq0pc_E2DzL7emopWoaoZTF_m0_N0YzFC6g6EJbOEoRoS
        K5hoDalrcvRYLSrQAZZKf1yuVCyixEoV9GfNQC3_osjzw2PAithfubEEBLuVVk4
        XUVrWOLrLl0nx7RkKU8NXNHq-rvKMzqg"
}
```

Claim	Description	
iss	Issuer of the JWT.	Quién creó el token
sub	Subject that the JWT is representing	Respecto a quién
aud	Audience for the JWT	Para quién se creó
exp	Time the JWT is set to expire	Segundos desde 1970
nbf	Time the JWT is valid from (not-before)	Cuando inicia
iat	Timestamp when the JWT was issued (issued-at)	Cuando fue creado
jti	Unique identifier for the JWT (JWT ID)	random

# ID token structure

Legacy

OIDC-conformant

```
{  
  "sub": "auth0|alice",  
  "iss": "https://vcsoft.auth0.com/",  
  "aud": "123",  
  "exp": 1482809609,  
  "iat": 1482773609,  
  "email": "alice@example.com",  
  "email_verified": true,  
  "https://app.example.com/favorite_color": "blue"  
}
```

GET /oauth2/authorize

POST /oauth2/authorize

POST /oauth2/token

POST /oauth2/revoke

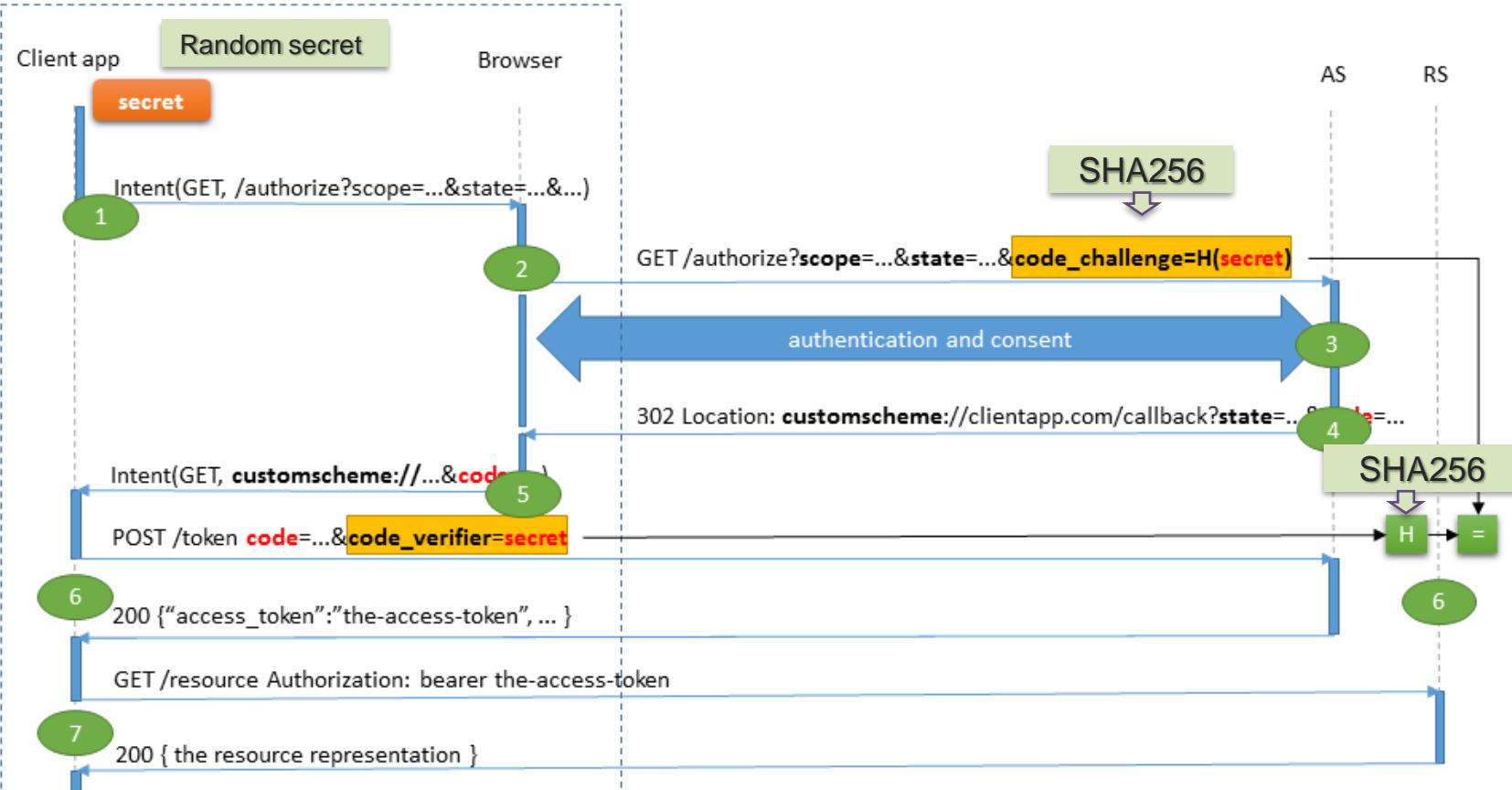
POST /oauth2/introspect

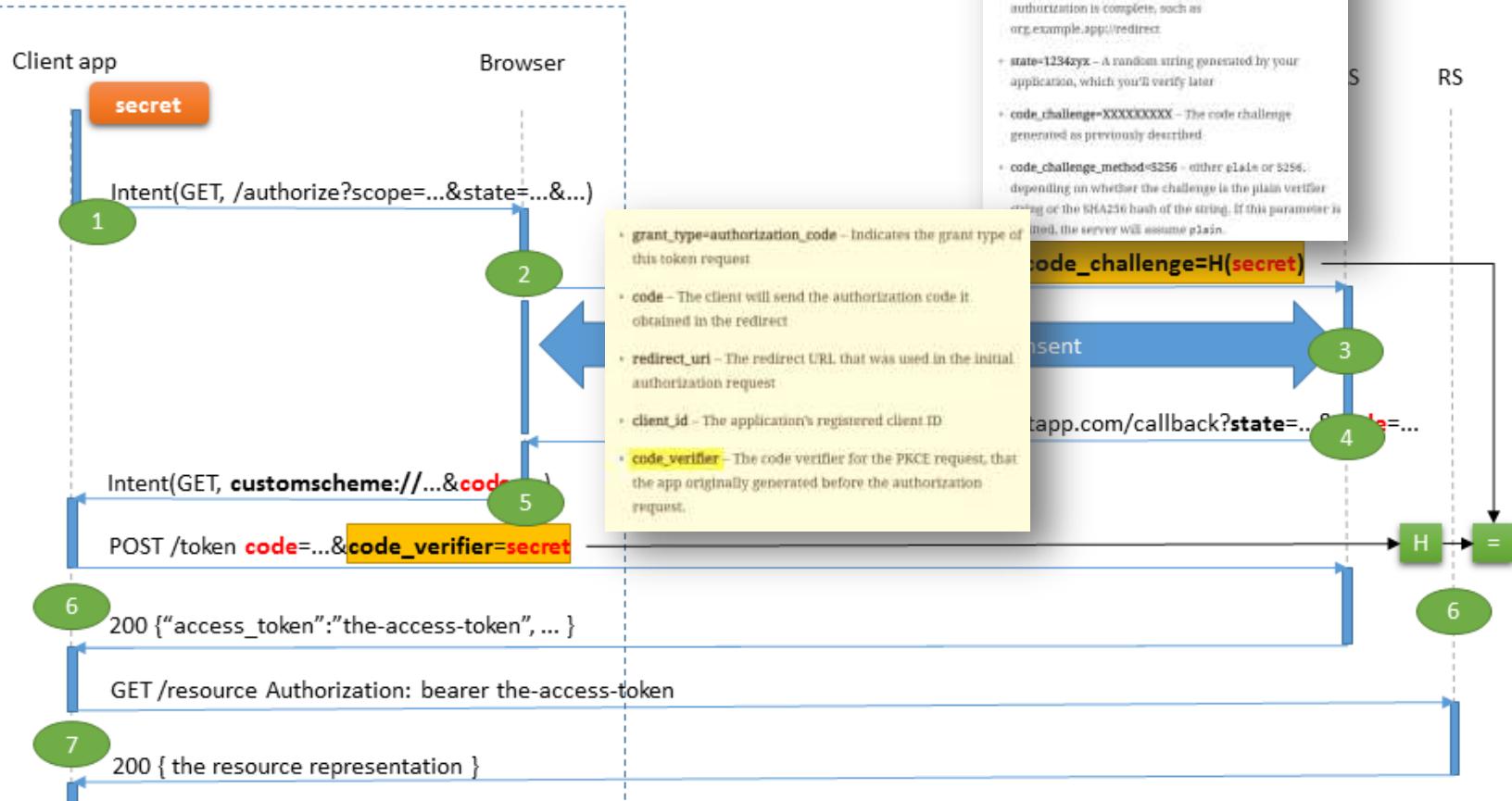
GET /oauth2/issued

DELETE /oauth2/issued

```
[  
 {  
 "clientId": "6975765558067200",  
 "clientName": "Alex Nelson",  
 "owner": "septowa",  
 "scope": "ikcone",  
 "issuedAt": "behwig",  
 "expiredAt": "zilav",  
 "refreshTokenIssued": false,  
 "misInfo": "faudcide",  
 "consentedOn": "jivecag",  
 "appId": "8586019806904320",  
 "org": "ceusfen",  
 "orgId": "6102866097864704",  
 "catalog": "zimakiip",  
 "catalogId": "1514907262517248"  
 }
```

# PKCE





# API Experts

**VC@SOFT**  
API Integrated Solutions

**VCSOFT Argentina S.A.**  
Luis María Campos 134 10°  
Tel.: (54-11) 5258 2374  
Cel:(54 9 11) 3236 9775  
Buenos Aires, Argentina

**VCSOFT S.A.S.**  
Calle 100 No 17A - 36 Of. 601  
Tel: (57-1) 621 9348  
Cel: 315 343 3919  
Bogotá, Colombia

**VCSOFT Chile S.A.**  
Francisco Antonio Encina 1781  
Tel: (56-2) 28171983  
Cel:(56-9) 6288 5114  
Santiago, Chile