# MANUAL DE SEGURETAT

# ÍNDEX

# 1. INTRODUCCIÓ

En les dues seccions següents es fa un exercici a mode de resum executiu de tots els aspectes que formen part de la disciplina de configuració i aplicació del paradigma de seguretat establert pel servei de l'API Manager Transversal de CTTI.

## 1.1. Objecte

El model de seguretat i la seva aplicació és una activitat transversal dintre dels objectius de la present iniciativa. Els principals objectius del model de seguretat que s'ha definit són:

- Poder **publicar APIs de forma segura** garantint, en tot moment, l'accés a les mateixes alhora que es manté la integritat i la seguretat dels backoffices que exposin els seus serveis/ operacions/dades mitjançant la solució API Connect és un dels objectius clau del projecte de l'API Manager.
- Poder **oferir de forma segura, ben gestionada i auditable**, la possibilitat a la ciutadania i als àmbits de la Generalitat i altres organismes relacionas, **el consum d'APIs publicades** que poden esdevenir fonts comunes d'operacions i d'informació.
- **Establir la configuració, les fonts d'usuaris de confiança i els diversos graus d'accés** (rols/ permisos) **a les diverses eines tecnològiques** que s'empraran en aquesta iniciativa (SaaS IBM API Connect, on-premise IBM API Gateways, SaaS IBM Portals dels 4 catàlegs, KeyCloak GICAR/Corporatiu, KeyCloak/Justicia, etc.) és un dels aspectes claus que tracta el document.
- **Establir les polítiques de publicació de les APIs** i, en concret, recomanar que tots els plans de consum tinguin la obligatorietat de l'aprovació de les sol·licituds de subscripció a les APIs esdevindrà un dels instruments de control bàsics i principals que ens facilitin la gestió i el coneixement, en tot moment, de les fonts de consum de les APIs publicades i el seu grau de consum (item important per a aspectes de facturació del consum d'una API publicada).
- **Permetre als responsable del servei Cloud / CTTI i el responsable de l'àmbit / codi de diàleg publicador (d'APIs) prendre decisions com ara revocar subscripcions i/o bloquejar consum inapropiats de certes APIs** també és una funcionalitat rellevant i que està íntimament emparentada amb la implantació del model seguretat establert pel projecte.

Tots els paràgrafs anteriors componen, a grans trets, les necessitats de seguretat associades a la present iniciativa.

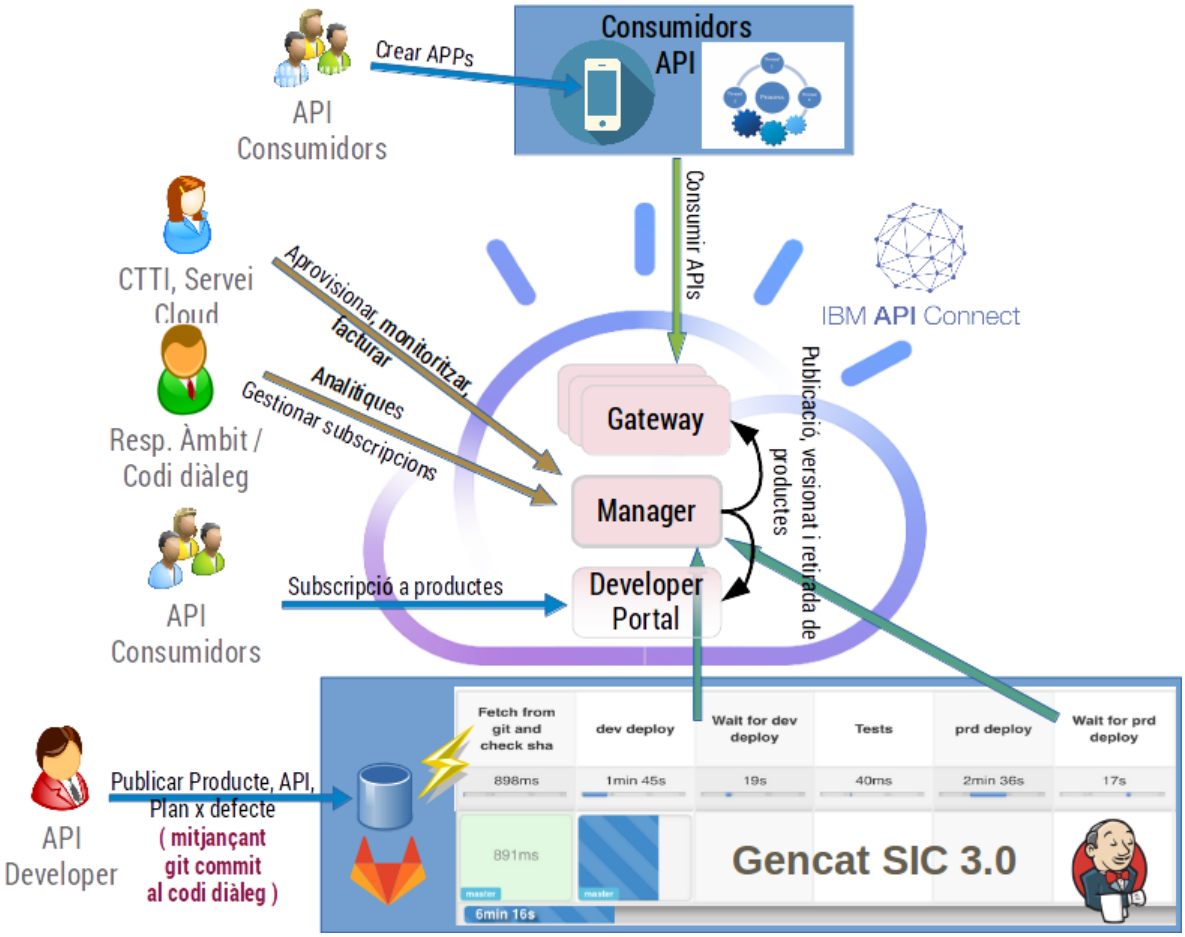
Aquest document aporta les respostes a les qüestions sobre el "què i el com" ho farem.

### 1.2. Abast

Estructurem el present document incidint en els aspectes claus relatius a l'aplicació del model de seguretat i la seva utilització per a tots els actors involucrats:

- **Actors involucrats (perfils):** Es descriuen tots els actors involucrats en tots els aspectes relacionats amb el servei de publicació i consum d'APIs.
  - CTTI.
  - Oficina tècnica API Manager Transversal.
  - Proveïdors de nous gateways on-premise.
  - Usuaris publicadors d'APIs (d'un codi de diàleg).
  - Usuaris dels portals de consumidors d'APIs (un portal per a cada catàleg).
  - Usuaris finals (corporatius o externs) que utilitzin SPAs i/o altre programari que inclou el consum d'una API amb protecció d'accés addicional del tipus KeyCloak(Corporatiu), protecció d'accés del tipus KeyCloak(propi d'un departament) i/o un altre tipus de protecció.
  - Usuaris responsables en la facturació del consum de les APIs.
- **Eines/Serveis**: Descripció de les eines i funcionalitats que caldrà tenir en compte per a la gestió dels permisos associats a les mateixes.
- **Tasques/operatives:** Tasques identificades que estiguin subjectes a un o més mecanismes de seguretat i que es componen d'un conjunt d'accions a dur a terme.
- **Accions identificades:** Tota tasca (operativa) està composada per una o més accions que cal portar a terme emprant una eina. Aquí és on s' incidirà en aspectes de configuració per a securitzar l'accés a aquestes accions dintre de les diverses eines que componen el parc tecnològic de la present solució.

## 2. DESCRIPCIÓ DE LA SOLUCIÓ




Elements principals de la solució:

| Element | Descripció |
|---|---|
| GitLab SIC 3.0 o GitHub Enterprise SIC+ | Repositori de codi on els col·laboradors desaran la **definició del fitxer aca i dels YAMLs de descripció d'un producte i de l'API** que es publicarà |
| Jenkins SIC 3.0 o GitHub Enterprise SIC+ | **Eina CI/CD** amb la que el responsable del codi de diàleg (on es produeix l'API) pot dur a terme totes les **operatives relatives al cicle de vida del binomi API (versió) - producte (versió)** |
| API Connect (SaaS) | Programari SaaS d'IBM (desplegat al servei cloud públic d'IBM que permet fer totes les **tasques de gestió addicionals incloses en la gestió del cicle de vida del binomi api (versió) - producte (versió)** |
| API Gateway (on-premise/AWS/Azure) | **Gateway de servei encarregat de controlar l'accés a les APIs publicades i establir les comunicacions** (i validar la seguretat) **abans de passar les peticions als backoffices involucrats** i destinataris finals de les operacions que es publiquen a API Connect. La infraestructura de Gateways segueix el mateix |

| | |
|---|---|
| | esquema de clústers i balancejadors en On-premise (CPD2), AWS i Azure, i està prevista la seva incorporació a CPD4 en els pròxims mesos. |
| Portal del desenvolupador (per catàleg) | Per a cadascun dels catàlegs aprovisionats es crea un **portal per posar a disposició dels usuaris** (futurs consumidors dels productes publicats sota els espais CD<codis de diàleg>) **la possibilitat de subscriure's al consum de les APIs** |

### 2.1. Actors involucrats

En les diferents eines participen diversos tipus d'usuaris/perfils:

| Nom | Descripció |
|---|---|
| CTTI | Àmbit Generalitat **responsable del servei** d'API Manager |
| OT API Manager | **Responsable del support i l'operació** de la plataforma i de la gestió dels gateways de AWS i Azure |
| Responsable APIs de l'Àmbit | **Responsable de totes les APIs** que es publiquen sota el seu patrocini |
| Release Manager CD | **Release Manager associat a un codi de diàleg.** Mitjançant l'auto-aprovisionament poden donar permisos als usuaris desenvolupadors sobre els pipelines que s'aprovisionen per a cada CD i sobre el repositori del GitLab(SIC 3.0)/GitHub(SIC+) associat al CD |
| Proveïdor d'APIs (lot Ax) | Usuaris Gencat amb permisos al GitLab(SIC 3.0)/GitHub(SIC+) per a l'execució de pipelines i del desplegament de les APIs |
| Consumidor d'APIs (lot Ax) | Release Manager/developers associats a un CD **responsable del consum d'una o N APIs publicades** |
| CPD2 | **Responsable de l'aprovisionament** i la gestió **dels gateways** de CPD2 que formen part de la plataforma |

### 2.2. Tasques

A continuació es mostren les tasques que requereixen de gestió de permisos:

| **Activitats a realitzar** | | | **Permisos involucrats** | | | |
|---|---|---|---|---|---|---|
| **Disciplina** | **Acció** | **Eina/automatisme /sub-acció** | **Oficina Tècnica APIM/SIC** | **Responsable Àmbit** | **Release Manager** | **Proveïdor /Consumidor** |
| **Desenvolupar / Publicar APIs** | | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | **Preparar projecte publicació d'APIs** | GitLab SIC 3.0 / Jenkins SIC 3.0 / GitHub SIC+ / API Connect (SaaS) | Crear el GitLab / GitHub associat al CD.<br><br>Crear el fitxer aca.yml dins del repo en cas de fer servir GitLab.<br><br>Crear els pipelines / workflows associades al CD.<br><br>Assignar permisos al responsable d'àmbit i al Release Manager per a que puguin gestionar els usuaris del projecte al GitLab / GitHub.<br><br>Assignar permisos al Release Manager per a que pugui executar les pipelines / workflows associades al CD.<br><br>Aprovisionar el CD als catàlegs pertinents i afegir els permisos pel | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | responsable d'àmbit i el Release Manager del CD. | | | |
| | **Crear API / Producte** | Instal.lar toolkit local a l'ordinador dels desenvolupadors | | | | Permisos per a instal·lar el toolkit gratuït de l'API Connect en l'ordinador local |
| | **Entregar API / Producte** | GitLab SIC 3.0 / GitHub SIC+<br><br>Pujar al GitLab/GitHub la definició del producte / API com a YAMLs | | Validar accés al GitLab / GitHub associat al codi de diàleg | Assignar permisos als usuaris desenvolupadors per a poder fer commit al GitLab / GitHub associat al codi de diàleg (en modalitat autogestió de projectes) | Validar que pot pujar i fer commit al repositori del GitLab / GitHub associat al codi de diàleg |
| | **Publicar Producte** | Jenkins SIC 3.0 / GitHub SIC+ / API Connect (SaaS)<br><br>Execució del pipeline/workflow de publicació del producte | En cas de no fer servir el SIC, l'oficina tècnica podrà desplegar els productes des de la consola de gestió a petició del desenvolupador | Validar els permisos d'execució dels jobs al Jenkins / GitHub | Validar els permisos d'execució dels jobs al Jenkins / GitHub<br><br>Executar el pipeline/workflow de publicació al Jenkins / GitHub | Permisos de lectura per a veure el resultat de l'execució dels jobs |

| | | | | | | |
|---|---|---|---|---|---|---|
| | **Validació governance automàtica al pipeline** | Jenkins SIC 3.0 / GitHub SIC+ — Compuerta de qualitat que valida automàticament les APIs i Productes contra els rulesets sic-corporate-api (bloc API-SEC entre altres) i sic-corporate-product (bloc PRD-PLA entre altres) abans del desplegament. | La OFT no intervé en la validació automàtica; només revisa manualment casos d'excepció que el pipeline no cobreix. | Rep informació dels findings a través dels logs del pipeline (SIC 3.0) o de comentaris a la Pull Request (SIC+). | Rep informació dels findings a través dels logs del pipeline (SIC 3.0) o de comentaris a la Pull Request (SIC+). | Corregeix els findings de severitat error per poder desplegar. Els findings de severitat warn són recomanacions no bloquejants. |
| | **Gestió del Cicle de Vida del Producte** | Jenkins SIC 3.0 / GitHub SIC+ / API Connect (SaaS) | En cas de no fer servir el SIC, l'oficina tècnica podrà gestionar el cicle de vida dels productes des de la consola de gestió a petició del desenvolupador | Validar els permisos d'execució dels jobs al Jenkins / GitHub | Executar els pipelines / workflows associats a la gestió del cicle de vida del producte | Permisos de lectura per a veure el resultat de l'execució dels jobs |

| **Consum d' APIs** | | | | | | |
|---|---|---|---|---|---|---|
| | **Subscripció al portal de consum d'APIs (veure manual de consum d'APIs)** | Portal del desenvolupador | | | | L'usuari executa l'acció d'autoregistre als portals involucrats |
| | **Creació d'una APP consumidora de Productes** | Portal del desenvolupador | | | | L'usuari crea tantes APPs com necessiti per a fer el consum d'APIs (productes) |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | | Cada App té associat el seu **clientID** |
| | **Subscripció a un producte** | Portal del desenvolupador | | | | L'usuari solicita la subscripció a un producte publicat i el vincula a una de les seves Apps (**clientID**) |
| | **Aprovació de la subscripció d'una producte** | API Connect (SaaS) | L'equip de suport de l'oficina tècnica d'APIM (previa aprovació del responsable d'àmbit del producte publicat), entra a l'eina API Connect (SaaS), s'adreça al catàleg/espai pertinent i accepta subscripció | El responsable d'àmbit del producte publicat rep email amb la sol·licitud de subscripció. Entra a l'eina API Connect (SaaS), s'adreça al catàleg/espai pertinent i accepta la subscripció | El Release Manager del producte publicat rep email amb la sol·licitud de subscripció. Entra a l'eina API Connect (SaaS), s'adreça al catàleg/espai pertinent i accepta la subscripció | L'usuari reb mail de confirmació de l'acceptació de la subscripció El producte passa a estar operatiu al gateway associat al catàleg/espai a on està publicat el producte |
| | **Denegació de la subscripció d'un producte** | API Connect (SaaS) | L'equip de suport de l'oficina tècnica d'APIM (previa decisió del responsable d'àmbit del producte publicat), entra a l'eina API Connect (SaaS), | El responsable d'àmbit del producte publicat rep email amb la sol·licitud | El Release Manager del producte publicat rep email amb la sol·licitud de | L'usuari rep mail de denegació de l'acceptació de la subscripció. Es rebutja |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | s'adreça al catàleg/espai pertinent i rebutja subscripció | de subscripció. Entra a l'eina API Connect (SaaS), s'adreça al catàleg/espai pertinent i rebutja la subscripció | subscripció. Entra a l'eina API Connect (SaaS), s'adreça al catàleg/espai pertinent i rebutja la subscripció | la subscripció al producte. |
| | **Eliminació de la subscripció d'un producte** | API Connect (SaaS) | L'equip de suport de l'oficina tècnica d'APIM (prèvia decisió del responsable d'àmbit del producte publicat), entra a l'eina API Connect (SaaS), s'adreça al catàleg/espai pertinent i elimina la subscripció | El responsable d'àmbit del producte publicat rep email amb la sol·licitud de subscripció. Entra a l'eina API Connect (SaaS), s'adreça al catàleg/espai pertinent i elimina la subscripció | El Release Manager del producte publicat rep email amb la sol·licitud de subscripció. Entra a l'eina API Connect (SaaS), s'adreça al catàleg/espai pertinent i elimina la subscripció | L'usuari esborra la subscripció feta des del portal de desenvolupador |

| **Operació de la plataforma** |
|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| | **Configuració proveïdor OAuth2 (KeyCloak /VALid, etc.)** | API Connect (SaaS) | L'oficina tècnica realitzarà la configuració del nou proveïdor d'OAut2 a API Connect. Les dades del proveïdor les ha de proveir l'equip proveïdor d'APIs, bé el responsable o els desenvolupadors. S'autoritza l'ús del nou servei OAuth2 a tots els espais necessaris. | | | L'usuari client de l'API és el responsable de programar l'enviament, a les peticions, de la capçalera Authorization amb el token ( "bearer" ) OAuth2 que representa la identitat de l'usuari Gencat que està fent la petició a l'API protegida després d'haver-se identificat amb el SSO / OAuth del proveïdor de seguretat (siqui aquest GICAR, Justicia o d'altres) |
| | **Instal·lar certificats** | API Connect (SaaS) / API Gateway (on-premise) | L'oficina tècnica podrà configurar certificats a la plataforma, bé per ser usats pels Gateways, o bé perquè siguin usats | Haurà de proveir a l'oficina tècnica dels certificats que | Haurà de proveir a l'oficina tècnica dels certificats que | Haurà de proveir a l'oficina tècnica dels certificats que |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | per una API que requereixi trucar a un backend que necessiti autenticació per certificat. Per al primer cas, serà l'oficina qui obtingui aquests certificats de Ciberseguretat. En el segon cas, loficina haurà de rebre els certificats adequats de lequip proveïdor. | requereixin fer servir els seus APIs | requereixin fer servir els seus APIs | requereixin fer servir els seus APIs |
| | **Registre Gateways a API Connect** | API Conect (SaaS) | Un cop el proveïdor de infraestructura marca els gateways aprovisionats com a vàlids i els registren com a gateway disponibles a la solució, l'oficina tècnica passa a utilitzar els nous gateways al catàleg/espai als que estan destinats | | | |

## 3. Model de seguretat

Recordem els principals actors de la sol·lució.

| Nom | Descripció |
|---|---|
| CTTI | Àmbit Generalitat **responsable del servei** d'API Manager |
| OT API Manager | **Responsable del support i l'operació** de la plataforma |
| Responsable APIs de l'Àmbit | **Responsable de totes les APIs** que es publiquen sota el seu patrocini |
| Release Manager CD | **Release Manager associat a un codi de diàleg.** Mitjançant l'auto-aprovisionament poden donar permisos als usuaris desenvolupadors sobre els pipelines que s'aprovisionen per a cada CD i sobre el repositori del GitLab(SIC 3.0)/GitHub (SIC+) associat al CD |

| | |
|---|---|
| Proveïdor d'APIs (lot Ax) | Usuaris Gencat amb permisos al GitLab(SIC 3.0)/GitHub(SIC+) per a l'execució de pipelines/workflows i del desplegament de les APIs |
| Consumidor d'APIs (lot Ax) | Release Manager/developers associats a un CD **responsable del consum d'una o N APIs publicades** |
| CPD2 | **Responsable de l'aprovisionament** i la gestió **dels gateways** de CPD2 que formen part de la plataforma |

### 3.1. Model de rols/permisos establerts

Actualment, se segueix el següent model de rols per a l' organització proveïdora CTTI a la plataforma d' API Manager:

- **Owner.**

  Té associats tots els rols de la plataforma, contractat al servei Cloud d'IBM. L'usuari identificat amb rol Owner tindrà assignat el Rol admin a totes les organitzacions, catàlegs i espais poden realitzar qualsevol acció disponible (actual i futura) del SaaS API Connect.

  Aquest rol el té una única persona, que es correspon amb el **responsable per part de CTTI de la plataforma** d'API Manager.

- **OT.**

  Les persones que són **membre de l'oficina tècnica d'API Manager** compten amb aquest rol i s'encarreguen de **mantenir i administrar la plataforma**. L'oficina tècnica ha de poder convidar a IBM API Connect (CTTI) als usuaris (amb rol) **Responsable APIs de l'Àmbit** o **Release Manager** en la fase 0 dels projectes (CDs) que vulguin disposar de la nova funcionalitat de desplegament d'APIs al API Connect.

  Aquest rol es dona mitjançant l'ús del grup d'accés **CTTI – API Manager OT** de l'**IAM d'IBM Cloud**, el qual té associats tots els rols de la plataforma menys el rol d'**Owner** a totes les organitzacions / catàlegs / espais i permet realitzar totes les activitats dins de l'eina API Connect relatives a l'organització, configuració de proveïdors de seguretat third-party OAuth2, creació de noves organitzacions, catàlegs i espais...

- **Developer.**

  Aquest rol es dona mitjançant l'ús del grup d'accés **CTTI – API Manager Developer** de l'**IAM d'IBM Cloud**, el qual té associat el rol de ***Viewer*** de la plataforma. CTTI només permet als desenvolupadors (**Proveïdor d'APIs**) l' accés a la plataforma en **mode de lectura**, per tal d' evitar que puguin modificar desenvolupaments i configuracions d' altres proveïdors. Aquest rol permet accedir als codis de diàleg a on tenen APIs/productes publicats.

- **Community Manager.**

  Aquest rol es dona a un **responsable d'un projecte (Responsable APIs de l'Àmbit/Release Manager)** en el cas que aquest projecte requereixi de la necessitat **d'administrar les seves subscripcions**. Per a això, es creen dos grups d'accés a l'IAM amb el nom **CTTI – CDXXXX – PRE i CTTI – CDXXXX – PRO**, on les **XXXX** seran substituïdes pel codi d'aplicació corresponent, de manera que coincideixi amb el nom de l'espai al qual es donarà accés. Aquest grup d' accés dóna el rol de **Community Manager** de la plataforma només dins l' **espai** corresponent. Així poden accedir a les funcionalitats de gestió de les subscripcions i les analítiques de consums.

Els consumidors (**Consumidor d'APIs**) es registraran als portals del desenvolupador, unint-se a una organització consumidora. Dins d' aquesta organització, podran tenir el rol d' **Owner, Administrador,**

**Desenvolupador o Visor**, rols que en aquest cas tenen **permisos diferents** als anteriorment definits, atès que només aplicaran dins del **Developer Portal**.

- Els usuaris de l'organització consumidora amb rol de **Visor** podran veure, però no modificar res del contingut dins de l'organització consumidora.
- Els usuaris amb rol de **Desenvolupador** podran crear aplicacions amb les quals subscriure' s als productes.
- Els usuaris amb rol d' **Administrador *o* Owner** podran administrar els membres i els seus permisos, alhora que les aplicacions i subscripcions.

## 3.2. Frameworks i fluxos d' autenticació i autorització suportats per API-M

### 3.2.1. Conceptes, estàndards i consideracions de seguretat

API Connect permet la gestió i ús de diferents fluxos d'autenticació i autorització per poder securitzar el consum de les APIs exposades. A l'hora de dissenyar els fluxos d'autenticació i autorització per al consum d'APIs de la plataforma, es tindran en compte diversos conceptes, estàndards i consideracions de seguretat, alhora que els casos d'ús de CTTI.

- Les APIs s' han d' exposar sempre mitjançant el protocol **HTTPS**, mai amb HTTP.
- Per defecte, les APIs seran securitzades mitjançant fluxos d'autorització **OAuth2.0/OpenID Connect**, l'estàndard a nivell d'indústria per a l'autorització, més l'enviament del **client-id** d'IBM. Es buscarà utilitzar, sempre que es pugui, els IDPs i servidors OAuth2 externs que conformen **GICAR** o altres àrees del **CTTI** (Keycloak, Pasarela GICAR-VALid, Azure B2C, etc.), integrant-les prèviament amb la plataforma d'API Connect (caldrà configurar-lo a la plataforma i associar-lo a tots els espais a on es sol·licita el seu ús). En cas que sigui necessari usar un proveïdor OAuth2 diferent als existents en CTTI, es generarà i usarà un nadiu a la plataforma.
- En el cas que una API no pugui ser securitzada amb OAuth2.0/OIDC, el nivell mínim de seguretat que ha de tenir serà l'enviament d'una API Key de tipus **client-id** d'IBM, per tal de no comptar amb cap API sense cap tipus de seguretat i per poder mantenir posteriorment una bona traçabilitat de les trucades i els usuaris/aplicacions associades.
- L'ús de **Mutual TLS** serà, en principi, opcional, sent el seu ús recomanat per reforçar la seguretat en certs casos concrets (vegeu punt de Mutual TLS més endavant per clarificar tots els possibles casos d'ús). **NOTA:** aquesta consideració es fa encara que, amb la versió actual de CTTI, la implementació de Mutual TLS no és possible. Actualment el CTTI fa servir el Secrets Manager d'IBM Cloud i amb la versió d'instàncies reservades d'IBM no és compatible encara.
- Es requerirà que totes les APIs estiguin associades amb, mínim, un pla que limiti el seu consum, per impedir la sobrecàrrega dels servidors de backend i que puguin ser fruits d' atacs DDOS. Els valors màxims dels plans seran acordats per l' oficina tècnica d' API Manager.

Al seu torn, és recomanable per bones pràctiques complementar els fluxos definits amb l' establiment d' altres mesures de seguretat addicionals per a les APIs, com la implementació de CORS o validació de les peticions d' entrada.

**Validació automàtica al pipeline.** La major part dels principis de seguretat exposats en aquest apartat es **validen de manera automàtica al pipeline corporatiu del SIC** com a part de la compuerta de qualitat prèvia al desplegament, mitjançant el bloc **API-SEC** del ruleset corporatiu sic-corporate-api. Concretament:

- **api-sec-01-root-security-defined-not-empty** (error) — La propietat security a nivell root ha d'estar definida i contenir com a mínim un requisit. No s'admet ni el camp absent ni un array buit.

- **api-sec-02-oas2-securitydefinitions-defined** i **api-sec-02-oas3-securityschemes-defined** (error) — Cal declarar com a mínim un esquema de seguretat (securityDefinitions a OAS 2 o components.securitySchemes a OAS 3).
- **api-sec-03-operation-security-not-empty** (error) — Cap operació pot declarar security: [], ja que això eliminaria la seguretat heretada del root i deixaria l'operació sense protecció.
- **api-sec-04-oas2-no-basic-auth** i **api-sec-04-oas3-no-basic-auth** (warn) — Es desaconsella l'ús d'esquemes basic als securityDefinitions (OAS 2) o securityScheme (OAS 3). L'estàndard corporatiu del SIC és OAuth / clientId-clientSecret.

El detall funcional complet de les regles és al **Disseny funcional (ERQ) corporatiu dels rulesets sic-corporate-api i sic-corporate-product**.

### 3.2.2. OAuth 2.0

El marc d'autorització d'OAuth 2.0 admet diversos fluxos (o concessions) diferents. Els fluxos són formes de recuperar un token d' accés.

Dins de cadascun dels fluxos admesos per OAuth 2.0, la idea dels rols(actors) forma part de l'especificació central del marc d'autorització OAuth2.0. Aquests defineixen els components essencials d' un sistema d' OAuth 2.0, essent els següents:

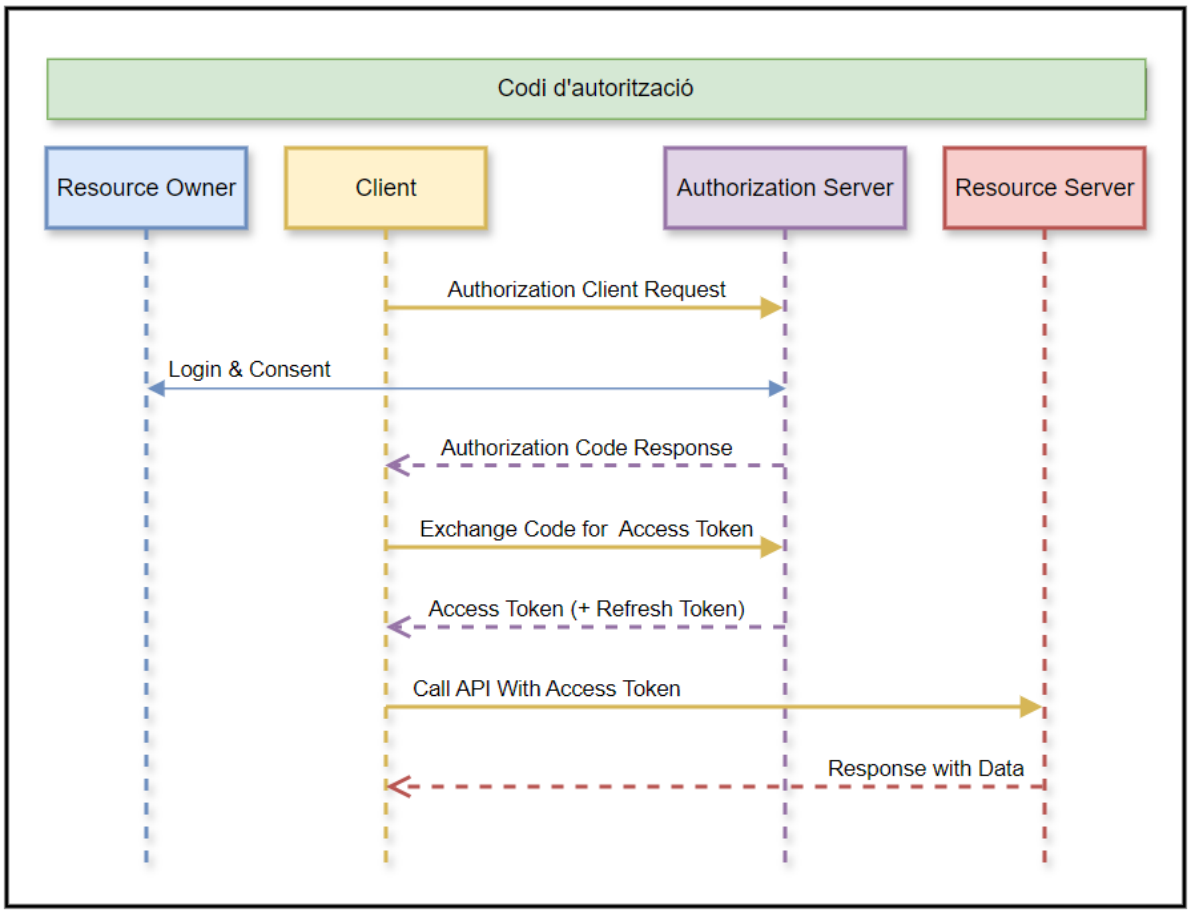
- **Resource Owner**: L'entitat (usuari o sistema) capaç de concedir accés a un recurs protegit. Quan l' entitat és una persona, se' l refereix com a usuari final.
- **Client**: El client és l' aplicació que realitza peticions als recursos protegits en nom del Resource Owner, comptant amb la seva autorització, és a dir, posseint el token d' accés corresponent.
- **Authorization Server**: Aquest servidor rep les sol·licituds de tokens d'accés del client i emet els corresponents tokens d'accés un cop el propietari del recurs s'ha autenticat i ha donat el seu consentiment. El servidor d' autorització exposa dos punts de connexió: el punt de connexió d' autorització, que maneja l' autenticació interactiva i el consentiment de l' usuari, i el punt de connexió de token, que està involucrat en una interacció de màquina a màquina.
- **Resource Server:** Un servidor que conté els recursos protegits de l' usuari i és capaç d' acceptar i respondre a peticions de recursos protegits mitjançant l' ús de tokens d' accés. Accepta i valida el token d' accés del client i li retorna els recursos adequats.

A continuació, s' exposen els diagrames de comunicació entre les diferents parts involucrades dels principals fluxos OAuth2.0 i els que es recomana utilitzar dins de l' organització del CTTI:

#### 3.2.2.1. Authorization Code Flow

Els clients confidencials i públics utilitzen el tipus de concessió del codi d' autorització per intercanviar un codi d' autorització per un token d' accés. Una vegada que l'usuari torna al client a través de la URL de redireccionament, l'aplicació obtindrà el codi d'autorització de la URL i el farà servir per sol·licitar un token d'accés. Per a més informació, consulteu el següent link.

No obstant això, **es recomana que tots els clients** utilitzin l 'extensió PKCE amb aquest flux per proporcionar una millor seguretat.




Les etapes compreses a nivell general dins del funcionament del flux OAuth **"Authorization code"** serien les següents:

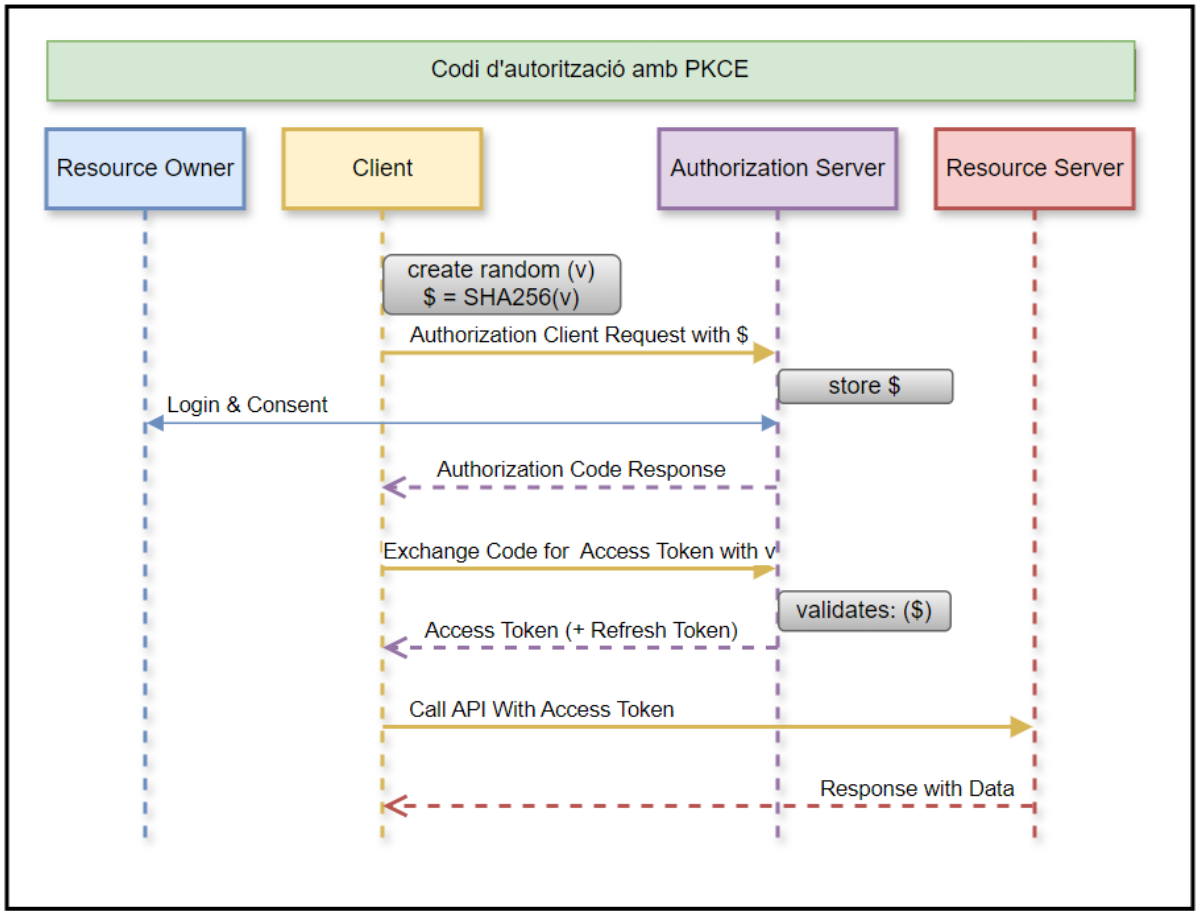
- El **client** inicia el flux sol·licitant accés a un subconjunt de dades de l'usuari, especificant quin tipus de concessió desitja utilitzar i quin tipus d'accés.
- Es demana a **l'usuari** que iniciï sessió en el **servidor d'autorització** i de la seva consentiment per a l'accés sol·licitat.
- Si l' **owner** dóna el seu consentiment, el **servidor d' autorització** redirecciona el **client** proveint-li un **codi d' autorització**.
- **L'aplicació client** demana un **token d'accés** únic que demostra que té permís de l'usuari per accedir a les dades sol·licitades. Per a això, s' autentica amb el **servidor d' autorització** i proporciona el **codi d' autorització** rebut en el pas previ.
- **L'aplicació client**, si la petició és correcta, obté un **token d'accés** (opcionalment pot rebre a més un **token de refresc**) i utilitza aquest token d'accés per realitzar trucades al **servidor de recursos protegits** i obtenir les dades rellevants.

#### 3.2.2.2. Authorization Code Flow amb PKCE

PKCE és un reforç de seguretat del flux Authorization Code que preveu atacs CSRF i d' injecció de codi d' autorització, i que permet que les aplicacions utilitzin el flux del codi d' autorització en clients que siguin públics o no siguin de confiança.

Per aconseguir això, es realitza un treball de configuració previ al flux i alguna verificació al final d' aquest per utilitzar de manera efectiva un secret generat dinàmicament. Això és el que determina una

diferenciació significativa respecte al flux original, ja que no és segur tenir un secret fix (i estàtic) en un client públic (com una aplicació SPA en un navegador). Per a més informació, consulteu el següent link.




Les etapes compreses a nivell general dins del funcionament del flux OAuth **"Authorization code flow amb PKCE"** serien les següents:

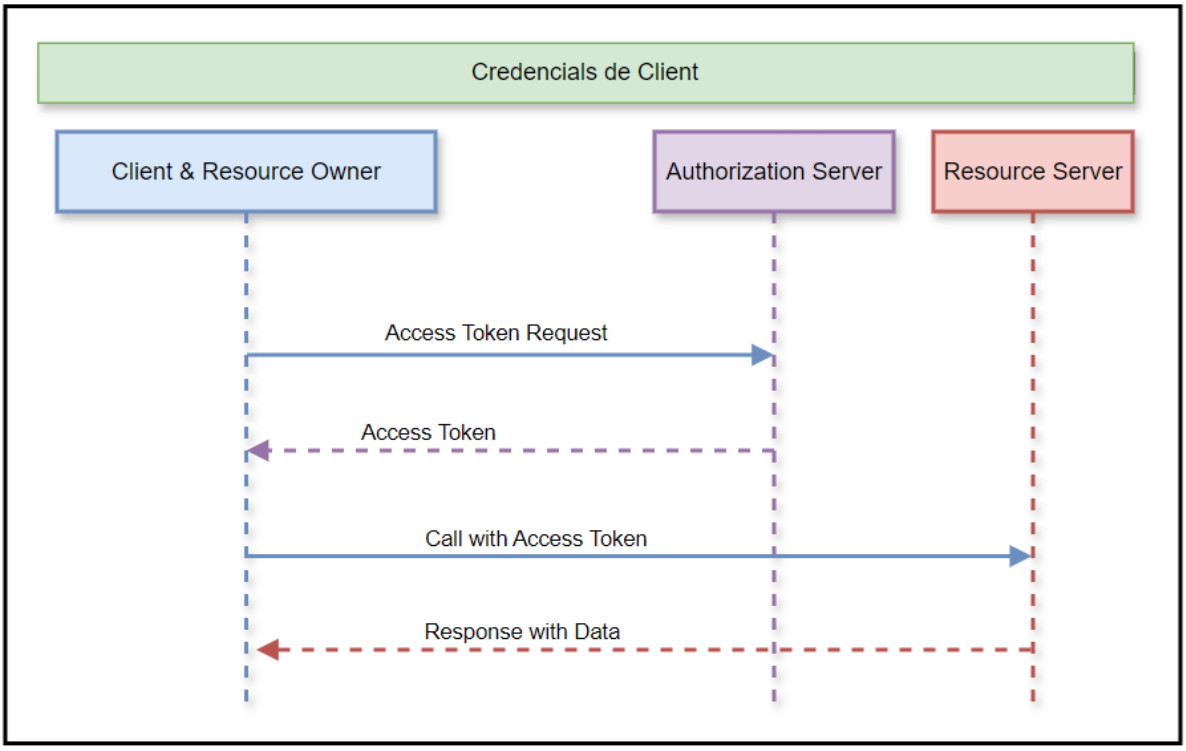
- El **client** crea i registra la informació secreta denominada **"code_verifier"**, i després calcula el **"code_challenge"** en funció del **"code_verifier"**. El seu valor pot ser **"code_verifier"** o el **hash SHA-256 de "code_verifier"**, però s'ha de preferir el hash criptogràfic.
- El **client** inicia el flux sol·licitant accés a un subconjunt de dades de l'usuari, especificant quin tipus de concessió desitja utilitzar i quin tipus d'accés. El **client** envia **"code_challenge"** i **"code_challenge_method"** opcional (una paraula clau que representa el text original o hash SHA-256) al **servidor d'autorització** juntament amb els paràmetres de sol·licitud d'autorització normals, i aquest els guardarà i registrarà per a comprovacions posteriors.
- Es demana a l'**usuari** que inicií sessió en el **servidor d'autorització** i doni explícitament el seu consentiment per a l'accés sol·licitat.
- Si l' **owner** dóna el seu consentiment, el **servidor d' autorització** redirecciona el **client** proveint-li un **codi d' autorització**.
- **L'aplicació client** demana un **token d'accés** únic que demostra que té permís de **l'usuari** per accedir a les dades sol·licitades. Per a això, s'autentica amb el **servidor d'autorització** i proporciona el **codi d'autorització** rebut en el pas previ, juntament amb el **"code_verifier"**, el qual serà verificat pel **servidor d'autorització** calculant el **"code_challenge"** en funció

d'aquest **"code_verifier"** i verificant si coincideix amb el **"code_challenge"** enviat originalment.

- **L'aplicació client**, si la petició és correcta, obté un **token d'accés** (opcionalment pot rebre a més un **token de refresc**) i utilitza aquest token d'accés per realitzar trucades al **servidor de recursos protegits** i obtenir les dades rellevants.

#### 3.2.2.3. Client Credentials Flow

Els clients utilitzen el tipus de concessió de credencials de client per obtenir un token d' accés fora del context d' un usuari. Amb aplicacions de **màquina a màquina (*M2M*)**, com CLI o serveis que s'executen en el seu back-end sense interacció d'usuari, el sistema autentica i autoritza l'aplicació en lloc d'un usuari. Les aplicacions *M2M* utilitzen el flux de credencials de client, en el qual transmeten el seu ID de client i el seu secret de client per autenticar-se i obtenir un token. Per a més informació, consulteu el següent link.




Les etapes compreses a nivell general dins del funcionament del flux OAuth **"Client Credentials flow"** serien les següents:

- El **client** envia les **credencials d' aplicació** al **servidor d' autorització** OAuth.
- El **servidor d' autorització** OAuth valida les credencials de l' aplicació.
- El **servidor d'autorització** OAuth respon al **client** amb un **token d'accés** generat si aquestes credencials són correctes.
- **L'aplicació client** utilitza aquest **token d'accés** per realitzar trucades a **recursos protegits** i obtenir les dades rellevants.

### 3.2.3. Tipus de clients

Decidir quin flux és adequat per a cada cas d' ús depèn principalment del tipus d' aplicació, però també influeixen altres paràmetres, com el nivell de confiança per al client o l' experiència que tinguin els

usuaris. A la *RFC 6749 The OAuth 2.0 Authorization Framework - Section 2.1* s'especifiquen els tipus de client que poden existir. Es mostra a continuació un breu resum:

- **Client públic**
  - Clients incapaços de mantenir la confidencialitat de les seves credencials (per exemple, clients que s'executen en el dispositiu utilitzat pel propietari del recurs, com una aplicació nativa instal·lada o un lloc web aplicació basada en navegador) i incapaç de protegir l'autenticació del client per qualsevol altre mitjà.
- **Client confidencial**
  - Clients capaços de mantenir la confidencialitat de les seves credencials (per exemple, client implementat en un servidor segur amb accés restringit a les credencials del client), o que estan capacitats de realitzar l'autenticació del client per altres mitjans.

Això dóna lloc als següents perfils de clients públics i confidencials generals:

- **Aplicacions natives**
  - Existents en una varietat de dispositius, inclosos computadores d' escriptori, telèfons, tauletes, etc., en general, un client públic.
- **Aplicacions basades en agents d' usuari**
  - Aplicacions basades en JavaScript i aplicacions SPA per als nostres propòsits, en general, un client públic.
- **Aplicacions web**
  - Arquitectura d' aplicacions web tradicional, normalment, un client confidencial.

### 3.2.4. Quin flux OAuth 2.0 he d'implementar en el meu cas?

Cal tenir en compte les següents consideracions a l' hora d' escollir el flux idoni:

- Clients públics versus confidencials.
- Si es posseeix o controla l' aplicació davant un tercer que posseeixi o controli l' aplicació.
- Tipus d' aplicació i client: aplicació web, aplicació nativa, aplicació basada en agent d' usuari, i si són públics o confidencials.
- Identitat de l'usuari final o de l'aplicació (o tots dos).

A continuació, s' exposen en una **matriu** les **casuístiques identificades i els fluxos òptims** a implementar:

| **Tipus d' aplicació** | **Tipus de client** | **Flux OAuth2.0 a implementar** | **Cas d' ús** | **Informació addicional** |
|---|---|---|---|---|
| Aplicació web (amb component dedicat del costat del servidor) | Confidencial | *Authorization Grant Code* | Autenticació d'usuaris finals i accés per part del client a un recurs (possiblement propietat de l'usuari final).<br>Un exemple d'aplicació dins de CTTI pròpia d'aquesta categorització seria:<br>*Aplicació 0434-Configuració de tràmits (GSIT)* | *RFC 6749 - Section 1.3.1* |

| | | | | |
|---|---|---|---|---|
| Aplicació nativa d' escriptori o mòbils | Públic | *Authorization Grant Code amb PKCE* | Autenticació d'usuaris finals i accés per part del client a un recurs (possiblement propietat de l'usuari final), on no es pot assegurar la confidencialitat per part del client públic.<br>Un exemple d'aplicació dins de CTTI propi d'aquesta categorització seria:<br>*2998 – Execució Penal* | *RFC 8252: OAuth 2.0 for Mobile and Native Apps* |
| *Single Page App (SPA)* (amb component dedicat del costat del client) | Públic | *Authorization Grant Code amb PKCE* | Autenticació d'usuaris finals i accés per part del client a un recurs (possiblement propietat de l'usuari final), on no es pot assegurar la confidencialitat per part del client públic. | *OAuth 2.0 for Browser-Based Apps draft-ietf-oauth-browser-based-apps-07- section 6.2* |
| *Single Page App (SPA)* (amb component dedicat del costat del servidor) | Confidencial | *Authorization Grant Code* | Autenticació d'usuaris finals i accés per part del client a un recurs (possiblement propietat de l'usuari final).<br>Un exemple d'aplicació dins de CTTI propi d'aquesta categorització seria:<br>*3563 - Banca online de l'ICF* | *OAuth 2.0 for Browser-Based Apps draft-ietf-oauth-browser-based-apps-07- section 6.3* |
| Tot tipus d' aplicació que pugui mantenir la confidencialitat del secret del client, no necessiti l' autenticació de l' usuari final i necessiti accedir a un recurs de tercers | Confidencial | *Client Credentials* | L' aplicació o la identitat del sistema s' autentiquen i després accedeixen a un recurs en un servidor de recursos.<br>Un exemple d'aplicació dins de CTTI propi d'aquesta categorització seria:<br>*Dades Obertes (Consulta i Consulta NTI)* | *RFC 6749 - Section 4.4* |

A continuació, es mostra un diagrama de selecció, que pot ajudar un desenvolupador a obtenir de manera fàcil el flux de seguretat que ha d' implementar.

Primer, es mostrarà el diagrama de selecció final que s'haurà de fer servir una vegada sigui possible implementar Mutual TLS a les Instàncies Reservades d'IBM API Connect. Fins aleshores, s'haurà de fer servir el segon diagrama de selecció, el qual treu de la foto l'opció de poder fer servir Mutual TLS.

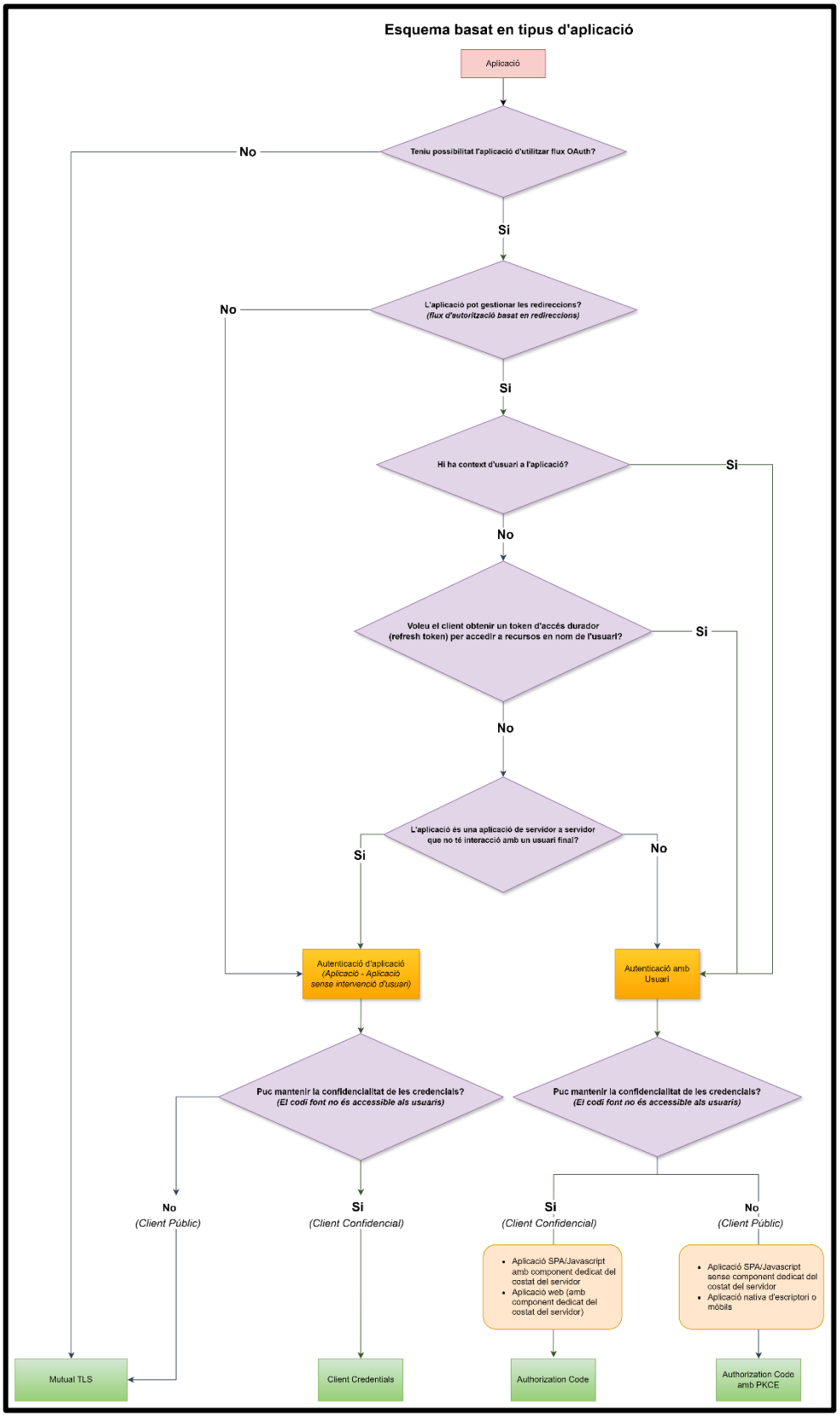



***Diagrama de selecció final, quan sigui possible la implementació de Mutual TLS.***

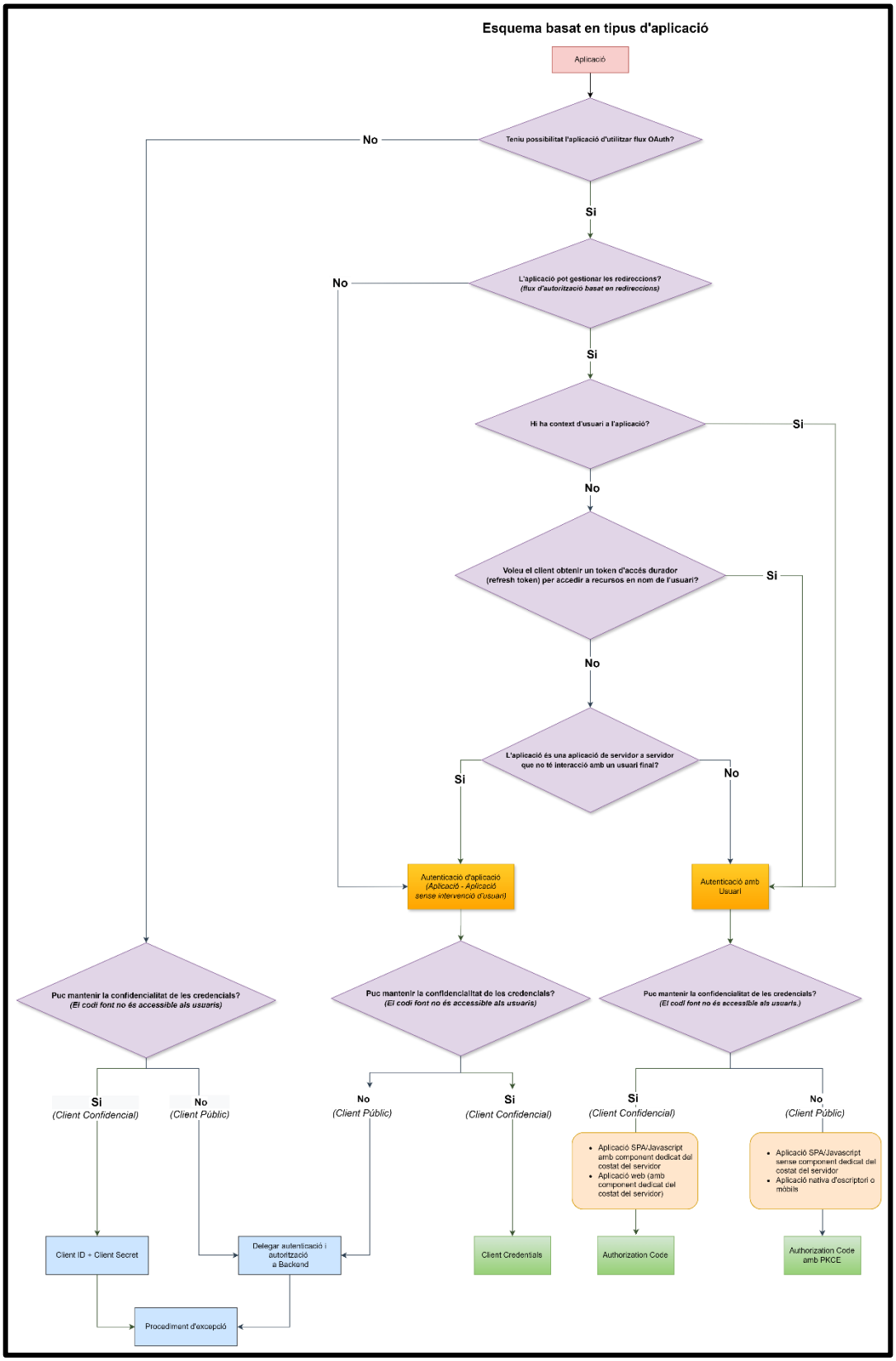



***Diagrama de selecció provisional, mentre no sigui possible la implementació de Mutual TLS.***

### 3.2.5. Ús de Mutual TLS

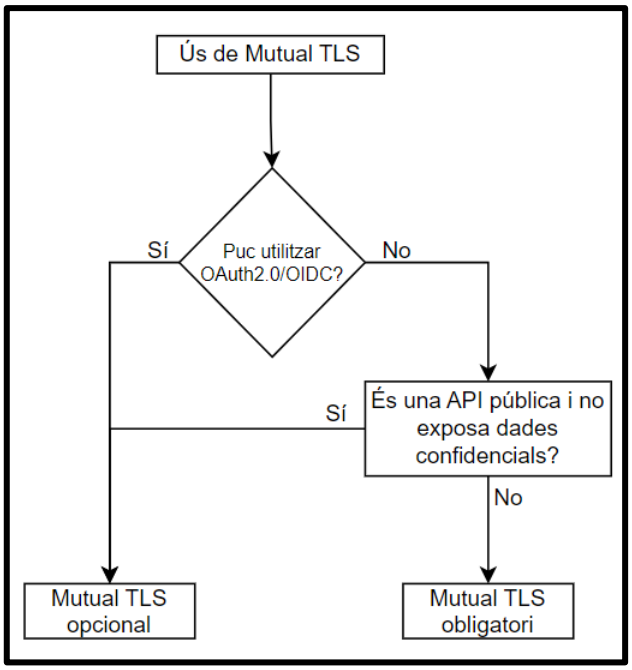
L' autenticació **mTLS** o l' autenticació mútua basada en certificats es refereix a dues parts que s' autentiquen entre si mitjançant la verificació del certificat digital proporcionat perquè ambdues parts estiguin segures de la identitat de les altres. En termes de tecnologia, es refereix a un **client** (navegador web o aplicació client) que s'autentica en un **servidor** (lloc web o aplicació de servidor) i aquest servidor també s'autentica davant el client mitjançant la **verificació del certificat** de clau pública o certificat digital emès per les autoritats de certificació **(CA)**.

El que es comenta en els paràgrafs següents està basat en la suposició que la plataforma suporta Mutual TLS. Atès que, a dia de l' escriptura de la versió del document, les Instàncies Reservades d' API Connect van amb la configuració de **Secrets Manager** i aquesta no suporta la implementació de Mutual TLS, no s' ha de tenir en compte actualment. Un cop sigui possible realitzar la seva implementació, es recomana seguir el que s' exposa a continuació.

Per configurar aquesta autenticació a la plataforma, **caldrà proveir el certificat corresponent** en la configuració de l' aplicació registrada, així com **activar l' esmentada autenticació a nivell d' API**, especificant si el certificat vindrà en una **capçalera** o a nivell de **TLS client**. Per a la generació dels certificats, s' haurà de seguir els estàndards de seguretat de CTTI, havent de **generar-se** un certificat que contingui el **mateix certificat CA root o intermedi** que el de les Gateways.

Com es va esmentar en les consideracions inicials, el mètode de securització **estàndard** de les APIs serà l'ús d'un **flux OAuth2.0/OIDC**, amb la qual cosa l'ús de **Mutual TLS** com a mesura de seguretat serà, en principi, **opcional i complementari**, podent aplicar-se com **a reforç de l'autenticació**. El seu ús com a reforç o complement dependrà de l' anàlisi prèvia de requeriments i necessitats particulars del projecte corresponent, essent més convenient la seva aplicació en els casos en què les APIs estiguin securitzades amb un nivell menor de seguretat.

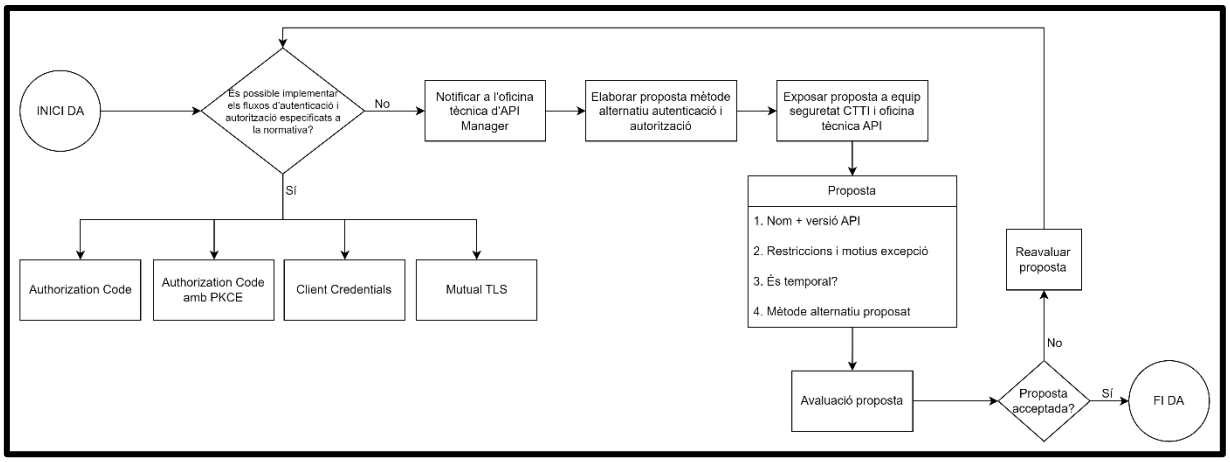
L'ús de Mutual TLS es considerarà obligatori només en el cas que l'API implementada no sigui una **API pública** que exposi dades no confidencials i, per tant, hagi de ser securitzada amb una major seguretat mínima que **client-id** o **client-id + client-secret** però, per restriccions de tercers, no es pugui implementar cap dels fluxos **OAuth2.0/OIDC**. Per exemple, una aplicació **legacy** que té implementat ja un flux de seguretat amb Mutual TLS i s'integra d'aquesta manera amb totes les aplicacions externes, no podent modificar el seu estàndard.



### 3.2.6. Procediment d' excepció

És possible que, a causa de restriccions imposades, generalment per terceres parts que requereixin integrar-se amb algun dels serveis exposats per l' organització, sigui impossible la implementació i ús dels fluxos d' autenticació i autorització contemplats, per la qual cosa aquesta funció d' autenticació i autorització s' hauria de buscar delegar els serveis del *back-end*. Per això, s' estableix un procediment d' excepció, que compta amb una sèrie de passos a donar per finalment rebre la validació per l' equip de seguretat quant al mètode alternatiu d' autenticació i autorització proposat.

A continuació, es descriuen els **passos** a prendre:



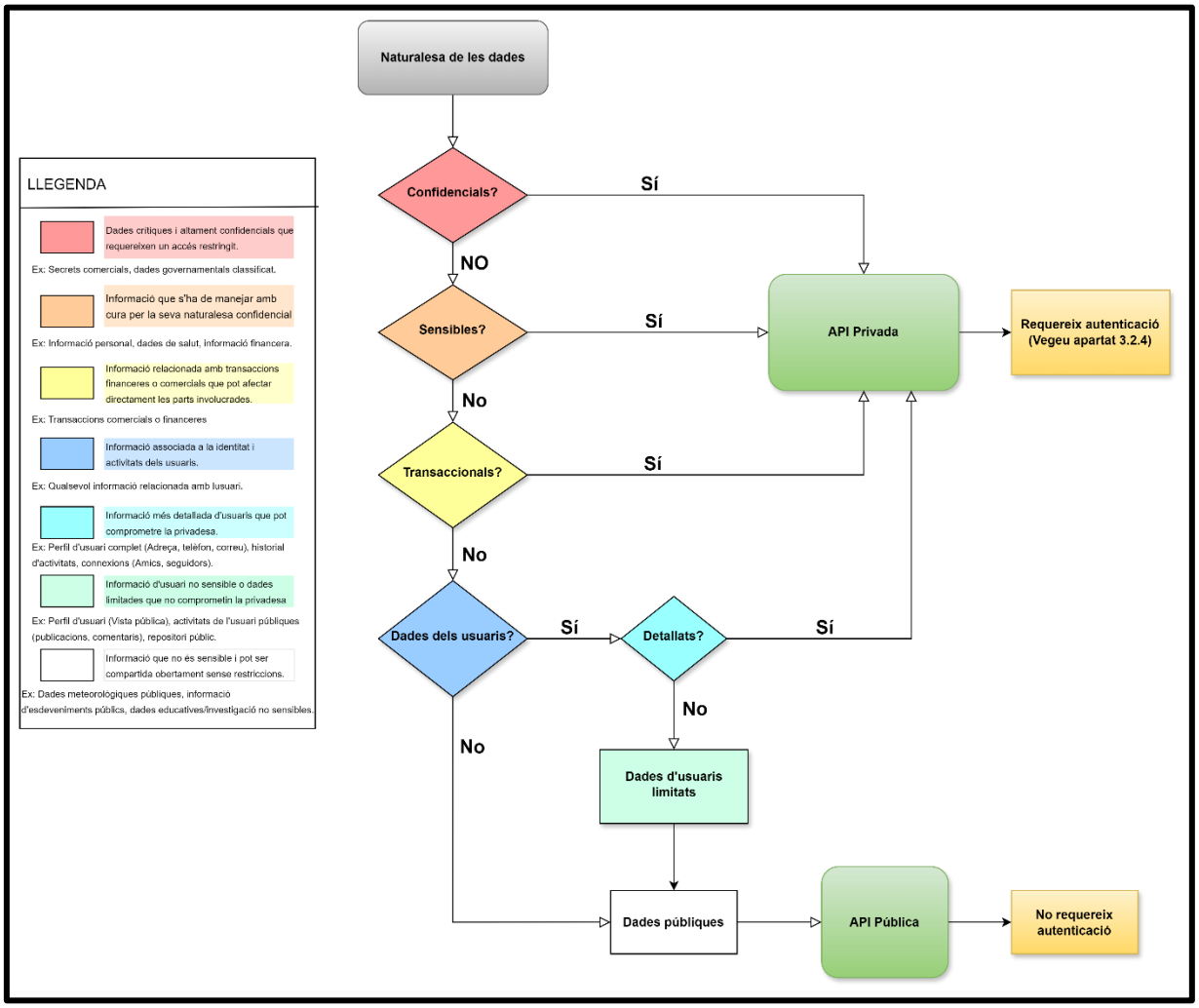

1. Primer, l' equip proveïdor ha d' analitzar, durant la realització del DA, si hi ha alguna **possibilitat d' implementar els fluxos d' autenticació i autorització** existents. Si hi ha la possibilitat que s'adaptin els fluxos actuals de l'aplicació tercera al *framework OAuth2.0/OIDC* o, en el seu defecte, *Mutual TLS*, s'ha de fer.
2. Si finalment es troben **restriccions insalvables** per alguna de les parts involucrades, l' equip proveïdor ho ha de comunicar a l **' oficina tècnica d' API Manager** i elaborar una proposta d' implementació de seguretat en el servei de backend. Cal tenir en compte en la solució proposada l' ús complementari i obligatori d' una API Key de tipus **client-id** a la capa d' API Manager.
3. Posteriorment, **l' equip proveïdor** ha d' exposar aquesta proposta alternativa de seguretat a l **' equip de seguretat de CTTI** i l **' oficina tècnica de l' API Manager** usant els procediments actuals que compta CTTI per a això, indicant si serà un mètode temporal fins que es pugui alinear la seguretat amb els fluxos definits o no. Es pot requerir, si fos necessari, realitzar sessions entre l' equip proveïdor i els equips de CTTI necessaris.
4. S **' avaluarà** la solució presentada i es prendrà una decisió.
    a. **Acceptada**. Un cop s' accepti la solució per part dels equips corresponents, quedarà reflectida al DA i es podrà procedir a la seva implementació.
    b. **Rebutjada**. Reavaluar la proposta i revisar el motiu pel qual ha estat rebutjada, revisant novament la possibilitat d' adaptació als mètodes definits en la normativa.

**Interacció amb la validació governance del pipeline.** El procediment d'excepció descrit en aquest apartat aplica a casos en què no és possible implementar cap dels fluxos OAuth 2.0/OIDC ni Mutual TLS. Cal tenir en compte que la validació automàtica del pipeline (bloc API-SEC del ruleset sic-corporate-api) exigeix declarar com a mínim un esquema de seguretat i **no admet security: []** a nivell d'operació. Per tant, encara que el proveïdor implementi un mètode alternatiu al backend, la definició OpenAPI de l'API ha d'incloure com a mínim l'esquema de seguretat mínim acceptable (típicament el X-IBM-Client-Id de subscripció). En cas de necessitat de bypass complet de la validació governance, cal escalar-ho també a l'oficina tècnica d'API Manager per gestionar l'excepció al ruleset.

### 3.2.7. Categorització de les APIs (públiques o privades)

En el proper apartat, explorarem la classificació de les APIs en base a la naturalesa de les dades que manegen, brindant una guia essencial per determinar la necessitat d'autenticació. Aquesta classificació permet identificar si una API específica ha de ser pública, sense requerir autenticació per part de l' usuari, o privada, cas en el qual se seguirà el diagrama detallat a la secció 3.2.4 per establir les mesures de seguretat corresponents.

Per simplificar la presa de decisions, presentem un diagrama de selecció que facilita als desenvolupadors identificar quin tipus d'API implementar segons la naturalesa de les dades, optimitzant així la configuració i garantint la seguretat adequada en cada cas.




## 3.3. Configuració de plans de consum

Com es comenta en el punt anterior, es requerirà que totes les APIs estiguin associades amb, mínim, un pla que limiti el seu consum, per impedir la sobrecàrrega dels servidors de backend i que puguin ser fruits d' atacs DDOS. De cara a estandarditzar els plans de subscripció dels productes que poden usar els desenvolupadors dins de CTTI, s'han definit i acordat els següents tiers, amb els següents límits màxims per a cada tier:

| Pla | Límit màxim Rate | Límit màxim Burst |
|---|---|---|

| | | |
|---|---|---|
| Default Plan | 100/h | 10/min |
| Bronze | 1000/h | 100/min |
| Silver | 2000/h | 200/min |
| Gold | 4000/h | 400/min |

Per tant, els projectes dins de CTTI hauran d'acollir-se a un d'aquests plans. Els límits establerts són els límits màxims per a aquest tier, podent cada tier tenir un valor entre 1 i el valor màxim que apareix a la taula per a cada límit.

**Validació automàtica al pipeline.** Els valors dels tiers i el format estricte dels límits es validen de manera automàtica al pipeline corporatiu del SIC pel bloc **PRD-PLA** del ruleset sic-corporate-product. Concretament:

**Estructura i format (error bloquejant):**

- prd-pla-01-plans-present-not-empty (error) — El Producte ha de declarar com a mínim un pla.
- prd-pla-03-approval-required (error) — Cada pla ha de declarar explícitament el camp approval.
- prd-pla-05-rate-limits-exists (error) — Cada pla ha de declarar la secció rate-limits amb com a mínim un rate-limit.
- prd-pla-06-rate-limit-per-hour (error) — El format del rate-limit ha de seguir N/1hour.
- prd-pla-07-no-unlimited-rate (error) — No s'admet el valor unlimited com a rate-limit.
- prd-pla-08-hard-limit-true (error) — Cada rate-limit ha de declarar explícitament hard-limit: true.
- prd-pla-13-burst-limits-exists (error) — Cada pla ha de declarar la secció burst-limits.
- prd-pla-14-burst-limit-per-minute (error) — El format del burst-limit ha de seguir N/1minute.

**Topalls corporatius i nomenclatura (warn recomanació):**

- prd-pla-02-name-tier-period-pattern (warn) — El nom del pla ha de seguir el patró {tier} o {tier}-{funcional} amb tier a default-plan, bronze, silver o gold.
- prd-pla-04-approval-false-notice (warn) — Avís informatiu quan un pla declara approval:false.
- prd-pla-09 a prd-pla-12 (warn) — Topalls màxims de rate-limits per tier (100/h, 1000/h, 2000/h i 4000/h respectivament per default-plan, bronze, silver i gold).
- prd-pla-15 a prd-pla-18 (warn) — Topalls màxims de burst-limits per tier (10/min, 100/min, 200/min i 400/min respectivament per default-plan, bronze, silver i gold).

El detall funcional complet de les regles és al **Disseny funcional (ERQ) corporatiu dels rulesets sic-corporate-api i sic-corporate-product**.

Els plans, per tant, hauran de seguir la següent nomenclatura:

- **Nom:** el nom del pla haurà de començar per un dels quatre tiers i estar en minúscules, separant cada paraula per guions. En cas que es vulgui afegir alguna cosa més que el nom del tier, s'haurà d'introduir a continuació, després d'un guió. **Exemples:** gold-auth, default-plan-users, silver-external-clients, etc.

- **Títol:** títol haurà de ser igual que el nom, cada paraula començar en majúscula i ha de contenir espais entre cada paraula en cas d' existir.
- **Descripció:** descripció del pla amb prou detall per ser fàcilment entendible, si és necessària. També es pot deixar amb un valor igual al títol.
  En cas de necessitar-ho, podran sol·licitar una excepció, via petició Jira a la OFT.

```
plans:
  gold:
    title: Gold
    description: Gold
```

Regles validadores relacionades del ruleset sic-corporate-product: prd-pla-02-name-tier-period-pattern (warn) — el nom del pla ha de seguir el patró {tier} o {tier}-{funcional} amb tier a default-plan, bronze, silver o gold.

Es recomana que continguin tots els plans aprovació de subscripcions, per evitar que qualsevol usuari registrat als portals pugui subscriure' s a un producte sense l' aprovació del **Release Manager** o **Responsable d' àmbit** corresponent.

## 4. Conclusions

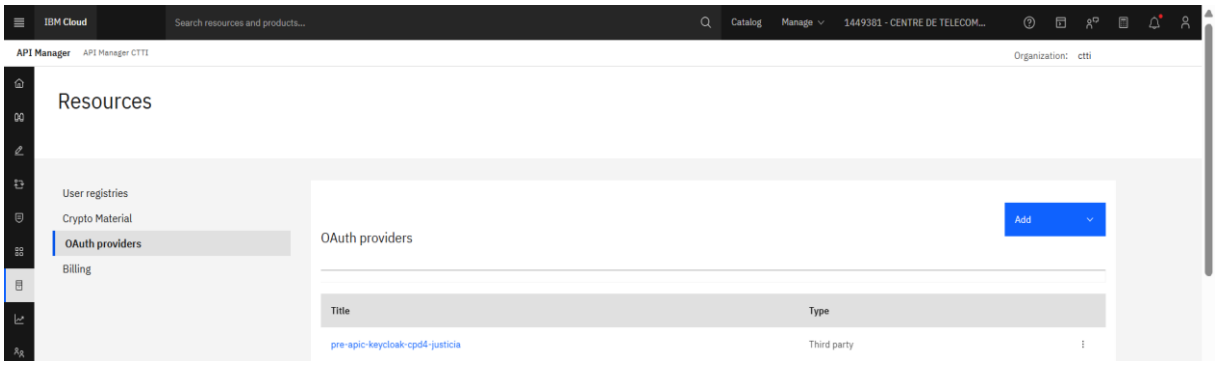
Control·lar la forma (plans de consum i requisits d'aprovació) i la seguretat que aplica a les APIs, conjuntament amb la integració de proveïdors OAuth2 com ara GICAR/Keycloak, ens permet tenir el control de quins usuaris efectius consumeixen les APIs publicades en aquesta nova plataforma.

## 5. Annex

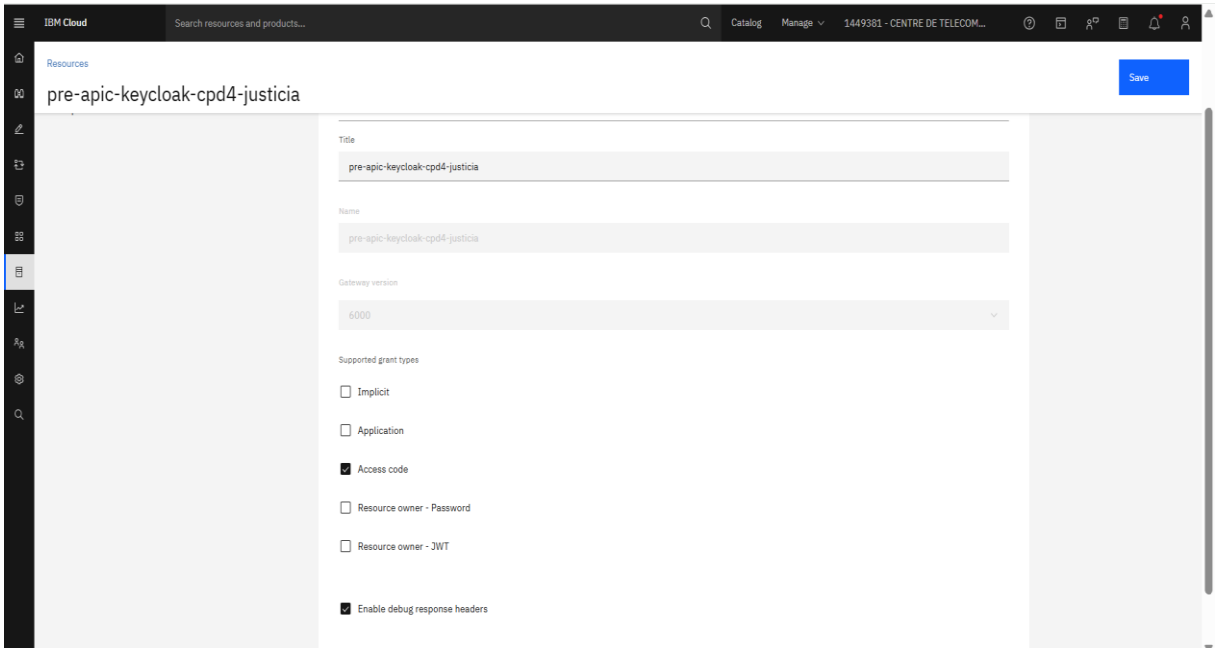
### 5.1. Configuració proveïdor OAuth KeyCloak

A continuació, es mostra com configurar un proveïdor OAuth a la plataforma que fa ús del **Keycloak de Justícia**. Aquests passos són necessaris sempre que es necessiti configurar un proveïdor OAuth extern que no hagi estat configurat encara.

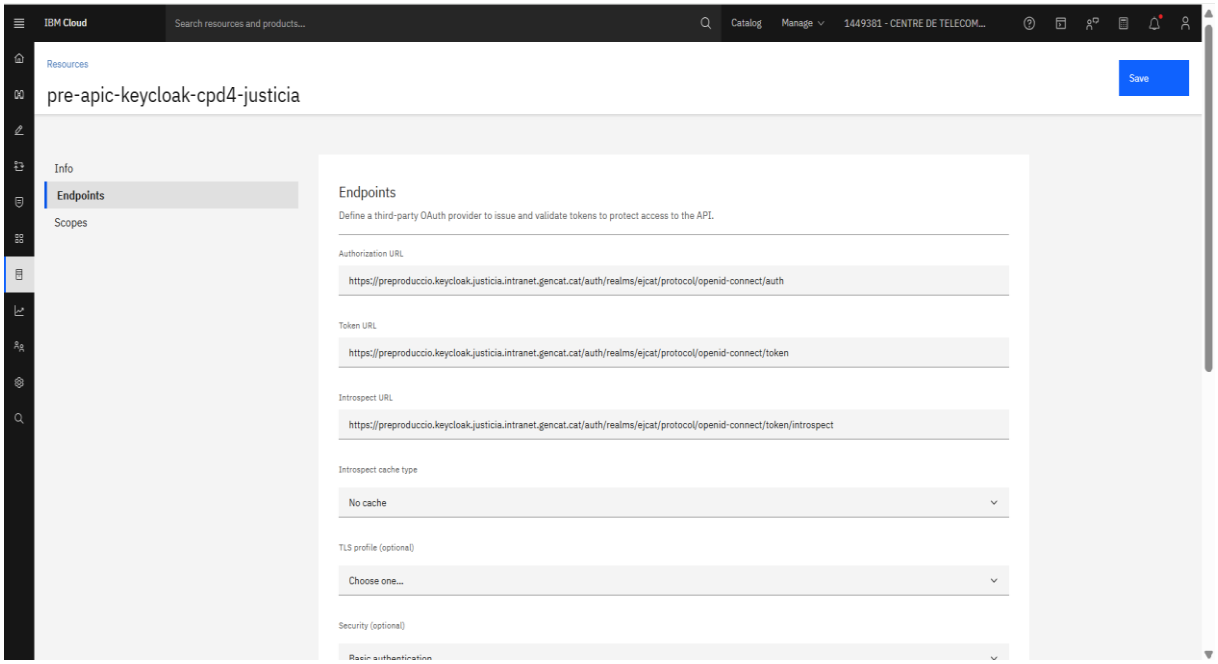
Primer, a la secció d' **OAuth providers**, es polsa a **Add** per afegir un nou proveïdor, seleccionant en aquest cas a **Third party OAuth provider**.

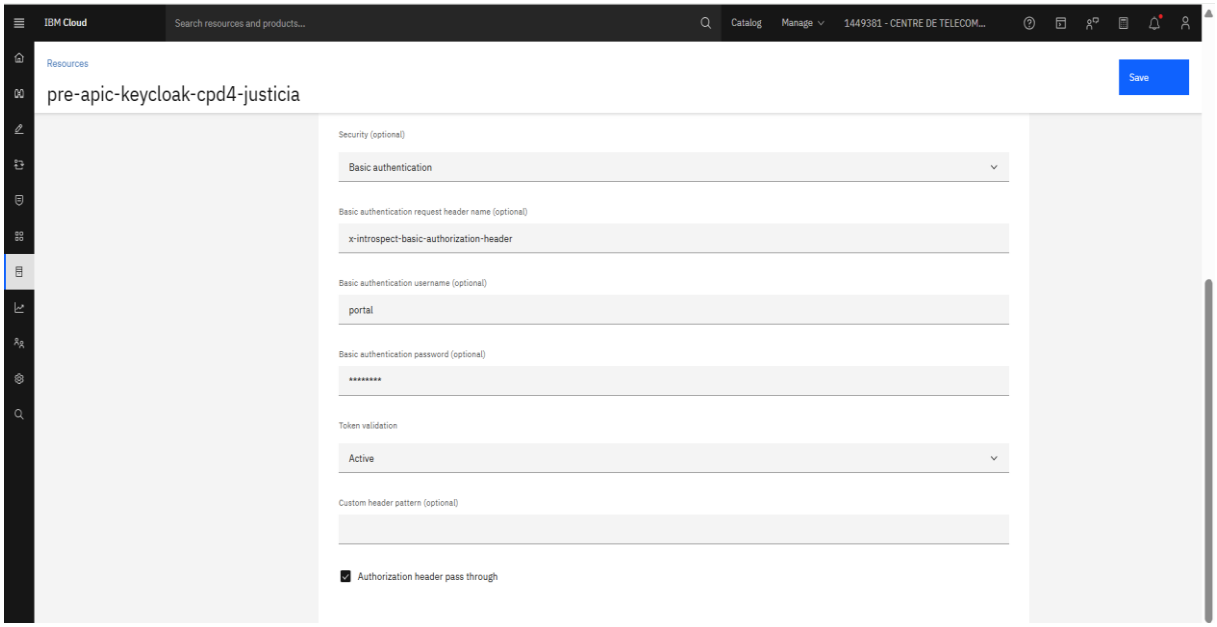



El **proveïdor/desenvolupador** haurà d'informar de quin flux es farà ús per a aquest proveïdor, sent els 2 recomanats en aquest document **Application (Client Credentials)** i **Access code (Authorization Code amb i sense PKCE)**.

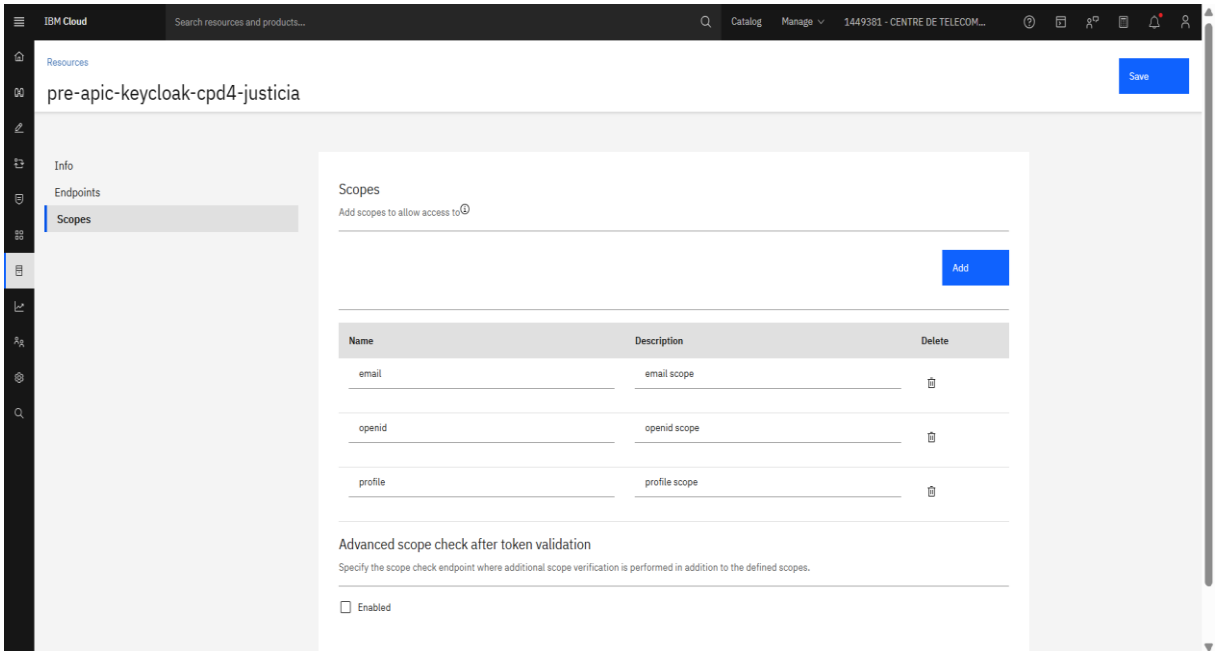



El proveïdor/desenvolupador haurà de proveir les **urls d'authorization, token i introspection** del proveïdor OAuth, juntament amb dades de connexió que es requereixin (**credencials de crida a introspecció** si es requereixen, **perfil TLS** a usar per connectar-se al proveïdor si es requereix, **configuració de caixet** si es requereix, etc.). Al seu torn, haurà de confirmar si es vol passar de forma automàtica el token al backend, que requereix activar el check **d'Authorization header pass through**.



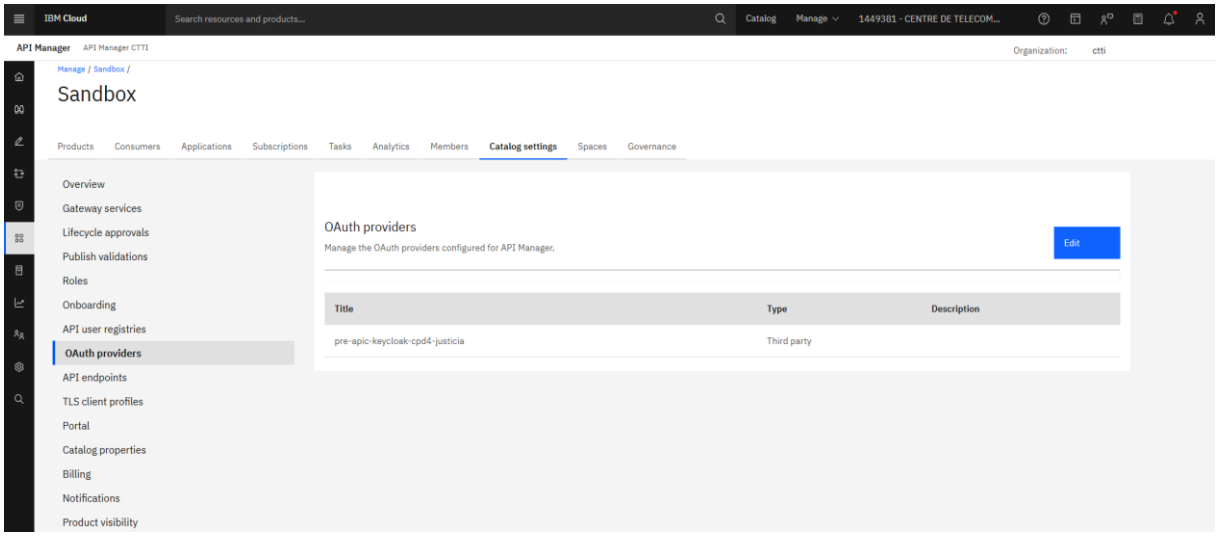


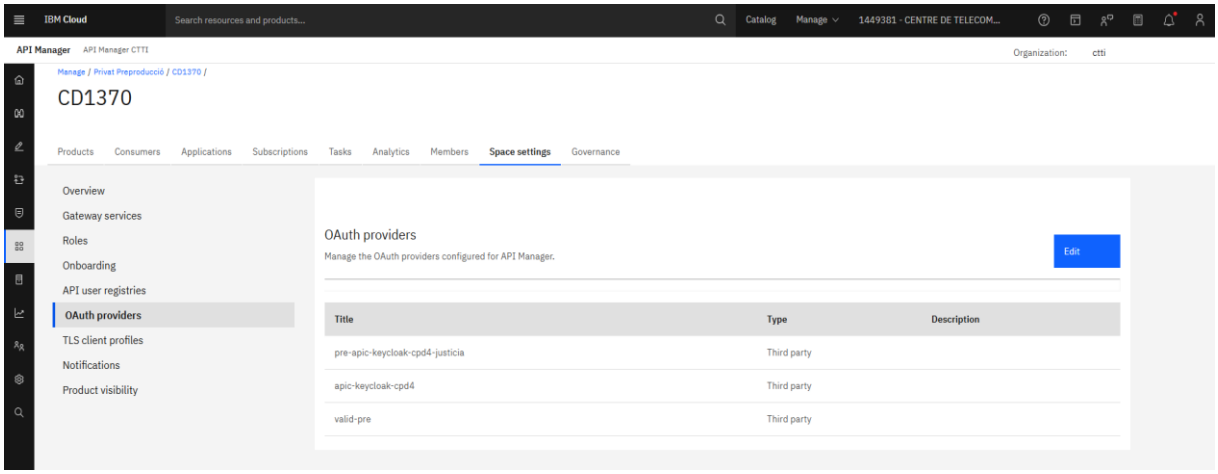

Finalment, el **proveïdor/desenvolupador** haurà de proveir les dades necessàries de **scopes** que es requereixin.




Després, l' oficina tècnica de l' API Manager ha d' activar aquest proveïdor OAuth en els espais corresponents on hagi de ser usat.

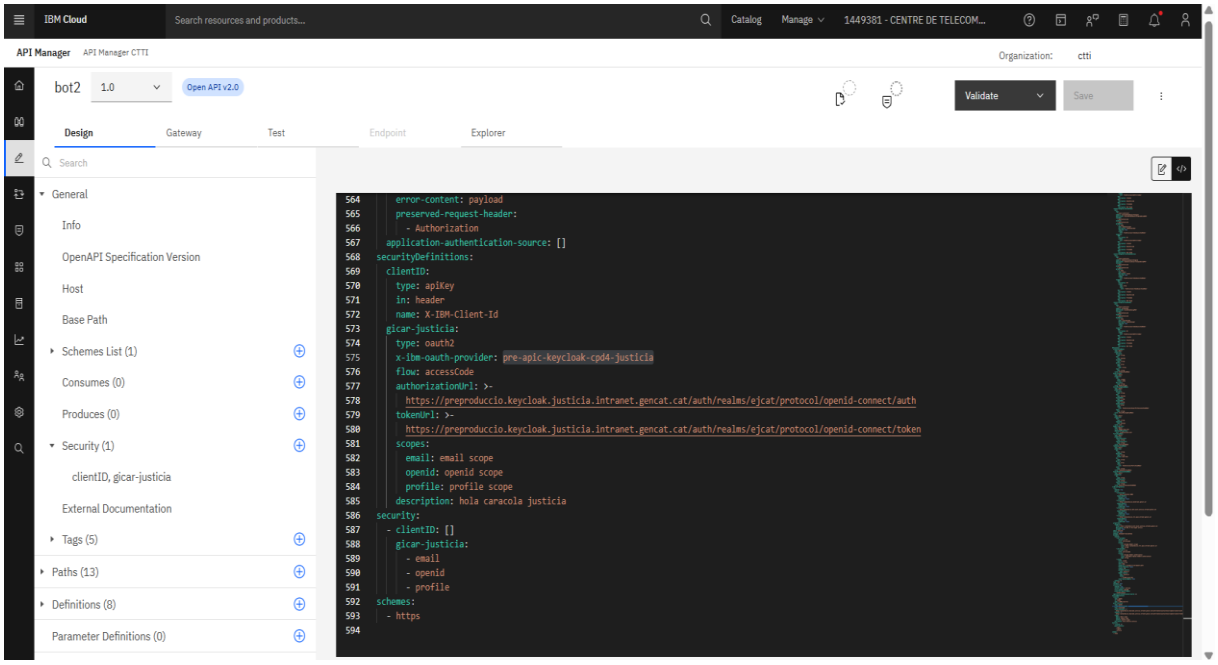
A continuació, es posen exemples de activació d'un proveïdor OAuth2 dins d'un espai (codi de diàleg) i del catàleg de Sandbox.




Un cop tenim activat el proveïdor third-party d'OAuth2 al CD, ja podem publicar APIs que requereixin d'aquest nivell addicional de seguretat en aquest espai, atès que no deixarà publicar-les abans.

bot2 és l'exemple d'una API que requereix l'autenticació amb el KeyCloak de Justicia.




L'api bot2 publicada amb 2 requeriments de seguretat:

1. X-IBM-Client-Id és el requeriment estàndard de seguretat mínim acceptable (implica subscripció vàlida al producte).
2. Autorització validada mitjançant el proveïdor de seguretat que està disponible dins l'espai on es publica el producte/API.

### 5.2. Configuració KeyCloak al codi client

Les SPAs, APPs de mòbil i altres programaris que desitgin consumir una operació publicada a una API amb seguretat OAuth2 d' un proveïdor configurat a Keycloak hauran de fer l'inicialització de la llibreria keycloak i fer el SSO/OAuth per a obtenir el token Bearer que enviaran a la capçalera d'autorització (Authorization) de les seves peticions als gateways.

Exemple d'ús del keycloak javascript adapter:

Cap problema, et passo el codi del HTML (en groc el que s'ha de posar per inicialitzar el Keycloak Javascript Adapter):

```
<html>
        <head>

      <script src="dist/keycloak.js" type="text/javascript">
        </script>
        </head>

   <body>
        <h1>Aplicació de proves Keycloak GICAR</h1>

      <div>
        L'usuari <b id="subject"></b> ha fet aquesta petició.
        <p><b>Detalls de l'usuari (extrets de <span id="profileType"></span>):</b></p>
```

```
        <p>Nom d'usuari (NIF): <span id="username"></span></p>
        <p>Email: <span id="email"></span></p>
        <p>Nom complet: <span id="name"></span></p>
        <p>Nom: <span id="givenName"></span></p>
        <p>Cognoms: <span id="familyName"></span></p>
                        <p><b>--------------------------------</b></p>
                        <p><b>Altres dades:</b></p>
                        <p>Access Token: <span id="tokenbearer"></span></p>
                        <p>Resposta User Info (extreta de resposta a l'API): <span
id="respostaUserInfo"></span></p>
                        <p>Resposta api manager (extreta de resposta a l'API): <span
id="respostaApiManager"></span></p>

        </div>

      <script type="text/javascript">
    var keycloak = Keycloak('http://localhost:3000/keycloak.json');
        var loadData = function () {
        document.getElementById('subject').innerHTML = keycloak.subject;
        if (keycloak.idToken) {
        document.getElementById('profileType').innerHTML = 'IDToken';
        document.getElementById('username').innerHTML =
keycloak.idTokenParsed.preferred_username;
        document.getElementById('email').innerHTML = keycloak.idTokenParsed.email;
        document.getElementById('name').innerHTML = keycloak.idTokenParsed.name;
        document.getElementById('givenName').innerHTML =
keycloak.idTokenParsed.given_name;
        document.getElementById('familyName').innerHTML =
keycloak.idTokenParsed.family_name;
                                document.getElementById('tokenbearer').innerHTML =
keycloak.token;
        } else {
        keycloak.loadUserProfile(function() {
                document.getElementById('profileType').innerHTML = 'Account Service';
                document.getElementById('username').innerHTML =
keycloak.profile.username;
                document.getElementById('email').innerHTML = keycloak.profile.email;
                document.getElementById('name').innerHTML = keycloak.profile.firstName
+ ' ' + keycloak.profile.lastName;
                document.getElementById('givenName').innerHTML =
keycloak.profile.firstName;
                document.getElementById('familyName').innerHTML =
keycloak.profile.lastName;

document.getElementById('tokenbearer').innerHTML = keycloak.token;
        }, function() {
                document.getElementById('profileType').innerHTML = 'Failed to retrieve
user details. Please enable claims or account role';
        });
```

```
        }


      var url =
'https://preproduccio.endpointma.autenticaciogicar4.extranet.gencat.cat/realms/gicarcpd4/protocol/openid-connect/userinfo';
        var req = new XMLHttpRequest();
        req.open('GET', url, true);
        //req.setRequestHeader('Accept', 'application/json');
                        req.setRequestHeader('Accept', 'application/x-www-form-
urlencoded');
        req.setRequestHeader('Authorization', 'Bearer ' + keycloak.token);
        req.onreadystatechange = function () {
        if (req.readyState == 4) {
                if (req.status == 200) {
                //var users = JSON.parse(req.responseText);
                //var html = '';
                //for (var i = 0; i < users.length; i++) {
                //        html += '<p>' + users[i] + '</p>';
                //}
                document.getElementById('respostaUserInfo').innerHTML =
req.responseText;
                                                        //document.write(req.responseText);
            console.log('finished loading data');
                }
        }
        }
        req.send();


                        //var url2 =
'https://preproduccio.ctti.apim.intranet.gencat.cat/ctti/privat-pre/api/v1/actuator/health';
                        //var url2 =
'http://limsunsangnim.efile.kr:8080/~weedscut/javascript/AjaxInAction/ch11/server_src/content/headers.jsp';
                        //var url2 =
'http://preproduccio.bot.ejcat.justicia.intranet.gencat.cat/api/v1/actuator/health';
                        //var url2 = 'https://catfact.ninja/fact';
                        var url2 =
'https://preproduccio.ctti.apim.intranet.gencat.cat/ctti/privat-pre/cat-fact/';
        var req2 = new XMLHttpRequest();
        req2.open('GET', url2, true);
        //req.setRequestHeader('Accept', 'application/json');
                        req2.setRequestHeader('Accept', 'application/x-www-form-
urlencoded');
        req2.setRequestHeader('Authorization', 'Bearer ' + keycloak.token);
                        req2.setRequestHeader('x-ibm-client-id',
'c8c2c4409c0e6b56e6a6262c8d69a215');
                        //req2.setRequestHeader('Origin', 'http://localhost:3000/index2.html');
```

```
        req2.onreadystatechange = function () {
        if (req2.readyState == 4) {
                if (req2.status == 200) {
                //var users = JSON.parse(req.responseText);
                //var html = '';
                //for (var i = 0; i < users.length; i++) {
                //        html += '<p>' + users[i] + '</p>';
                //}
                 document.getElementById('respostaApiManager').innerHTML =
req2.responseText;
                                                        //document.write(req.responseText);
                console.log('finished loading data api manager');
                }
        }
        }
        req2.send();



   };
        var loadFailure = function () {
        document.getElementById('customers').innerHTML = '<b>Failed to load data.
Check console log</b>';
        };
        var reloadData = function () {
        keycloak.updateToken(10)
              .success(loadData)
                .error(function() {
                document.getElementById('customers').innerHTML = '<b>Failed to load
data.  User is logged out.</b>';
                });
        }
    keycloak.init({ onLoad: 'login-required' }).success(reloadData);
</script>

<br><br>
<button onclick="reloadData()">Reload data</button>
        </body>
</html>
```

Podeu trobar més informació al respecte els clients existents a: https://www.keycloak.org/