# Interoperabilità e Cooperazione Applicativa delle Regioni

Azione di transizione ai nuovi standard

## Deliverable di Progetto

Versione 1.8

## Sommario

## Sommario

# 1. Finalità e riferimenti del documento

Scopo del documento è descrivere le risultanze delle attività svolte, relativamente al task INF-1, nell'ambito dell'azione interregionale di transizione ai nuovi standard, come previsto per la terza annualità.

## 1.1. Riferimenti

Il documento congiunto di fine annualità, prodotto congiuntamente dai tre task, contiene un'ampia sezione di analisi del Contesto, alla quale si rimanda per un corretto inquadramento delle attività qui descritte.

# 2. Introduzione

Il tema dell'interoperabilità, nell'ambito normativo italiano, ha riguardato nell'ultimo decennio l'adozione su scala nazionale del sistema SPCoop. Tale sistema è incentrato sul protocollo di comunicazione SOAP, esteso con un'intestazione personalizzata chiamata Busta eGov.

Nel 2018 AGID ha pubblicato la prima bozza delle linee guida del "Modello di Interoperabilità per le Pubbliche Amministrazioni" che prospettano uno scenario di migrazione da SPCoop ad un nuovo modello di interoperabilità basato su solidi standard internazionali. Dopo un lungo lavoro di gestazione, il 9 settembre del 2020, con la determinazione n. 406/2020, AGID ha poi pubblicato la prima versione delle nuove specifiche di interoperabilità, nella forma di «Linee di Indirizzo sull'interoperabilità tecnica»:

- https://trasparenza.agid.gov.it/archivio28_provvedimenti-amministrativi_0_122346_725_1.html

Il lavoro sulle specifiche è poi proseguito, portando in un documento ad-hoc le tematiche specifiche relative agli standard di sicurezza alla base dell'interoperabilità delle API, prima incluse nelle specifiche di ModI. Questo nuovo documento focalizzato sia sugli algoritmi e i cypher di sicurezza utilizzabili, sia sulle modalità d'uso dei certificati X509 per stabilire il trust tra le diverse parti cooperanti, ha poi portato alla definizione di una nuova linea guida sulla sicurezza dell'interoperabilità delle API. Dopo il consueto periodo di consultazione, entrambe le specifiche sono state portate al rango di 'linea guida', con Determinazione n. 547 del 1 ottobre 2021, avente ad oggetto l' Adozione delle "Linee guida Tecnologie e standard per la sicurezza dell'interoperabilità tramite API dei sistemi informatici" e delle "Linee guida sull'interoperabilità tecnica delle Pubbliche Amministrazioni".

Infine, con la determinazione n. 627/2021, AGID ha adottato le "Linee Guida sull'infrastruttura tecnologica della Piattaforma Digitale Nazionale Dati per l'interoperabilità dei sistemi informativi e delle basi di dati" ai sensi dell'articolo 50-ter, comma 2 del CAD. Queste Linee guida hanno l'obiettivo di favorire la

condivisione delle banche dati attraverso API e semplificarne l'accesso attraverso la costruzione di un catalogo delle API.

Sul versante europeo, il Parlamento Europeo ha approvato il regolamento eIDAS a luglio del 2014, prevedendo la sua piena efficacia dal 1 luglio del 2016. Il regolamento tratta le tematiche dell'identità elettronica e dei cosiddetti "servizi fiduciari" al fine di assicurare il valore legale delle transazioni elettroniche a livello europeo.

Di fronte alla forte evoluzione in essere, le Regioni si sono attivate nell'ambito di questo progetto per elaborare un modello condiviso che, conformemente con le proposte nazionali ed europee, soddisfi pienamente le esigenze delle Regioni.
Il principale obiettivo del task INF-1 del progetto è quello di analizzare le varie necessità emergenti nei diversi contesti regionale, nazionale ed europeo, al fine di cercare una convergenza che eviti di dover applicare modalità e realizzare infrastrutture diverse per l'accesso ad ognuna di esse.
L'obiettivo è quindi quello di individuare e sperimentare un modello di riferimento ed una rispettiva architettura di dispiegamento che assicurino:

1. la conformità al nuovo modello di interoperabilità proposto da AGID;
2. l'interoperabilità con i protocolli di interoperabilità previsti a livello Europeo;
3. il pieno supporto degli standard di mercato.

Nella figura che segue viene mostrata la prima versione del modello di riferimento per l'interoperabilità proposto in ICAR e finalizzato a descrivere le modalità con cui i

3 diversi layer regionale, nazionale ed europeo possano interagire.

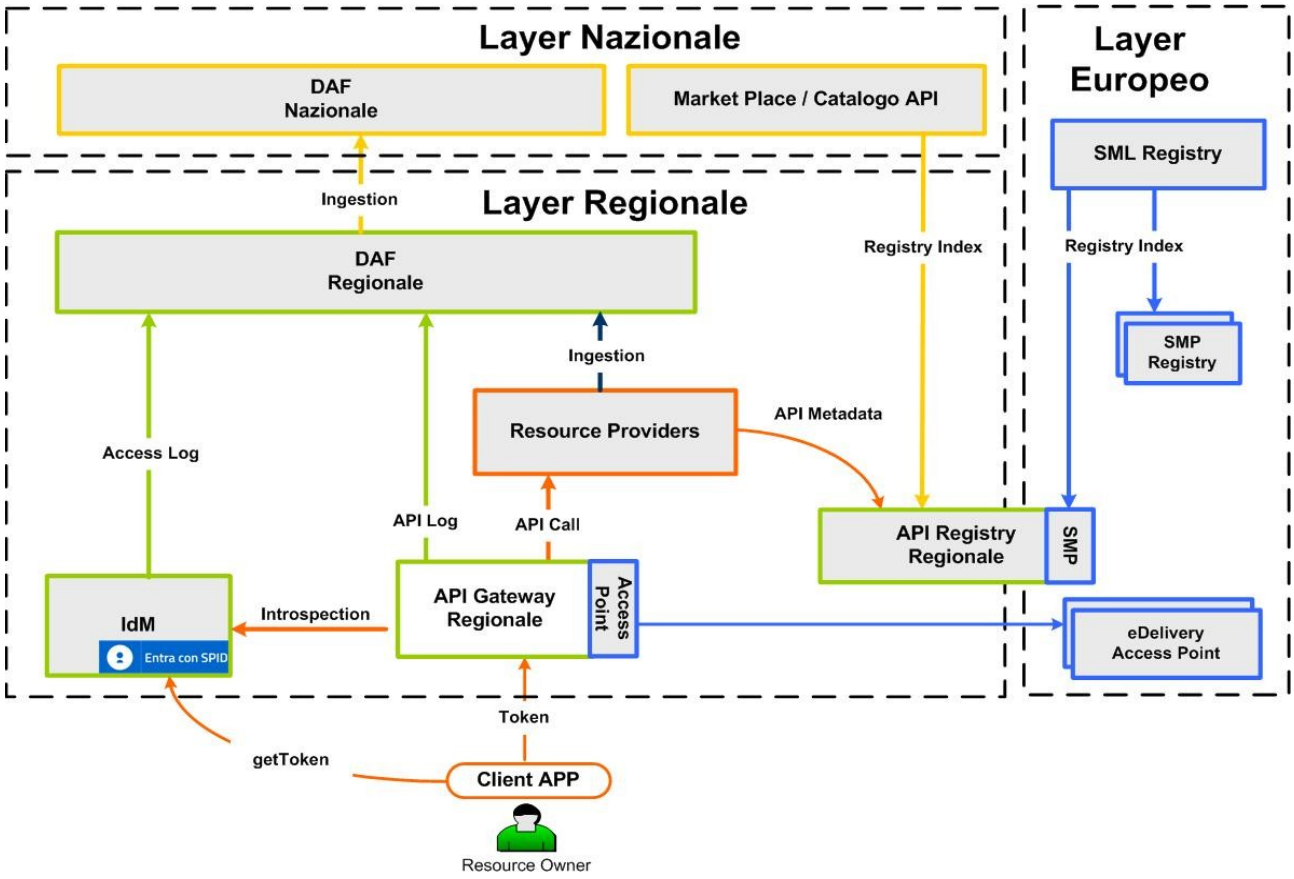



***Figura 1: Modello di Interoperabilità Applicativa in ICAR***

Il modello individua le seguenti componenti di base per le infrastrutture regionali:

- l'Identity Manager che presiede al rilascio ed alla verifica delle diverse credenziali di accesso utilizzate dagli applicativi a livello regionale, interagendo con gli IdP SPID per la fase di autenticazione e delega da parte dell'utente;
- l'API Gateway Regionale, che verifica ed autorizza tutte le richieste in arrivo dalle applicazioni client verso gli API Provider interni al dominio regionale;
- l'API Registry, che censisce tutte le API erogate dal dominio regionale, mettendo a disposizione dei potenziali fruitori tutti i metadati necessari per l'accesso.
- il DAF regionale, un cluster Big Data che raccoglie tutti i dati di interesse prodotti dall'ecosistema regionale.

Il livello regionale interagisce poi con i livelli nazionali ed europeo, in accordo alle specifiche di interoperabilità di questi due domini.
Per quanto riguarda il livello nazionale, facendo riferimento alle attuali bozze delle linee guida sinora circolate, i principali impatti riguardano:

- la necessità di implementare sull'API Gateway i profili di interoperabilità e di sicurezza previsti dalla specifica Modi;
- la necessità di dotare l'API Gateway delle caratteristiche necessarie all'interoperabilità con la Piattaforma Digitale Nazionale Dati (PDND). Con i recenti aggiornamenti normativi risulta evidente che questa piattaforma assumerà un ruolo sia per quanto riguarda la catalogazione delle API, sia per la messa disposizione degli strumenti di security al fine di supportare l'autenticazione per l'accesso alle API.

Per quanto riguarda il livello europeo, facendo riferimento alla specifica eDelivery, i principali interventi prevedono l'estensione dell'API Gateway in modo da potersi qualificare come "Access Point" eDelivery e l'estensione del Registro regionale, in modo da potersi qualificare come "SMP Registry" eDelivery.

# 3. Modello applicativo di riferimento

Al fine di realizzare delle soluzioni infrastrutturali di interesse per l'interoperabilità dei sistemi informativi delle Regioni italiane, è necessario partire dall'analisi delle comuni esigenze in termini di specifici requisiti di interoperabilità. Tali requisiti saranno descritti tramite opportuni **casi d'uso** che possano rappresentare in modo chiaro le diverse esigenze emerse nei tanti progetti già gestiti dalle Regioni italiane. Rifacendoci quindi all'approccio proposto nelle linee guida di interoperabilità recentemente rilasciate da AGID, per ogni caso d'uso individuato saranno proposti opportuni **pattern di interoperabilità**, che rispondano alla definizione proposta da AGID di: "possibile schema di risoluzione di un caso d'uso, espresso come una serie di linee guida che raccomandano come utilizzare una specifica tecnologia od approccio, e permette eventualmente di risolvere eventuali ambiguità/punti non adeguatamente definiti in alcune tecnologie possibili con cui le PA possono interoperare".

## 3.1. Lo scenario

Per consentire l'individuazione dei casi d'uso, e quindi lo studio dei pattern di interoperabilità relativi, definiamo inizialmente lo scenario di riferimento nel quale si inquadrano tutte le interazioni tra gli attori coinvolti.

In Figura 2 è illustrato lo scenario applicativo di riferimento per il task INF-1.

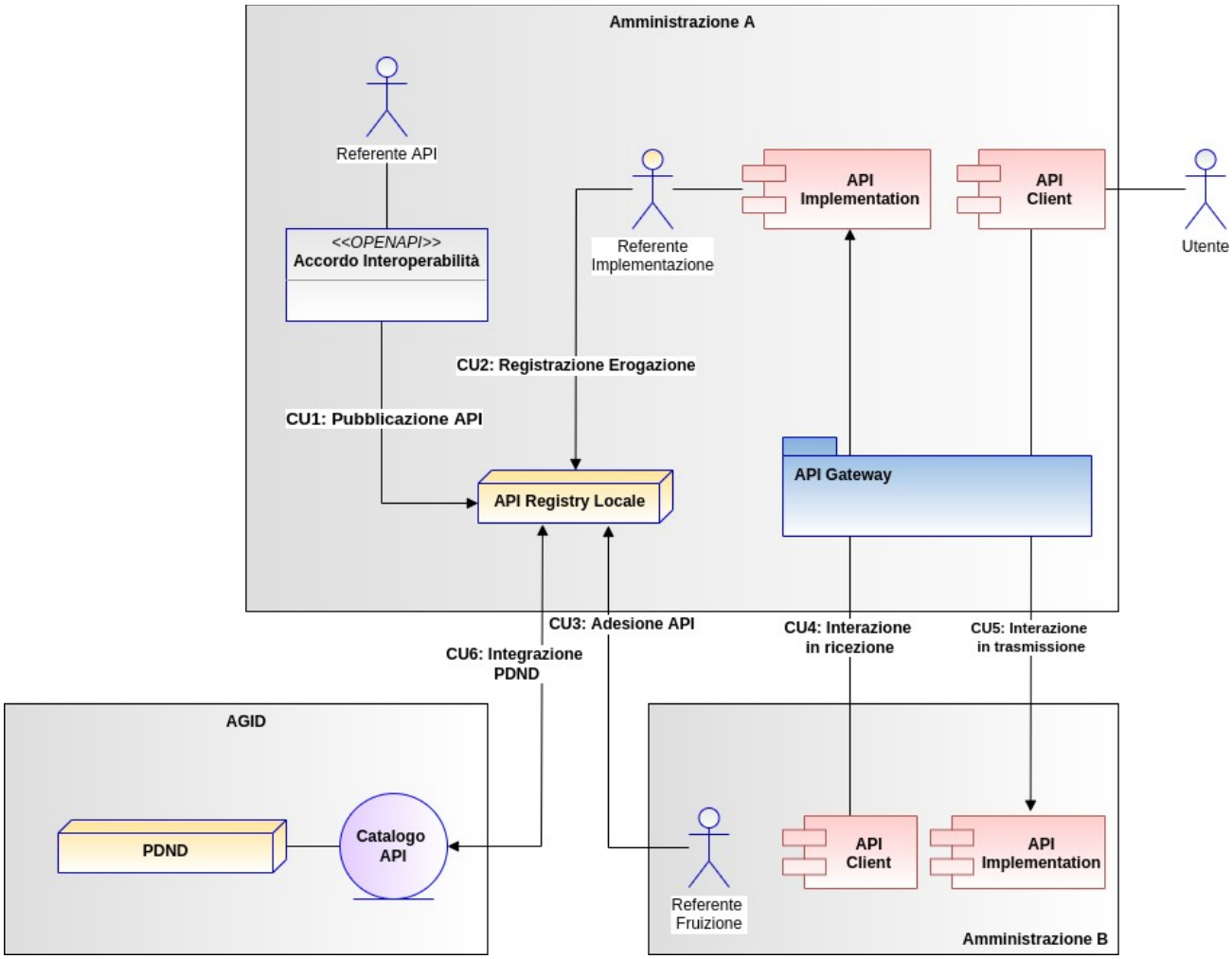



***Figura 2 - Scenario applicativo di riferimento per INF-1***

Lo scenario di riferimento si compone dei seguenti domini interoperanti:

- L'ente (Amministrazione A) cui si fa riferimento per l'espressione delle esigenze tipiche e quindi dei casi d'uso. In questo contesto lo si può considerare un ente regione;
- L'ente interlocutore (Amministrazione B) che interagisce con l'Amministrazione A tramite le API condivise;

- La PDND in qualità di repository centrale delle API dispiegate sul territorio e fornitura degli strumenti di base per l'accesso alle stesse.

Nell'ambito della singola amministrazione locale troviamo i seguenti elementi:

- API Registry Locale, una versione locale del Catalogo API integrata ai meccanismi supportati dalla PDND.
- Il Referente API, individuo che si occupa di formulare la descrizione formale di una determinata API, in accordo a un modello denominato "Accordo di Interoperabilità", al fine di pubblicarla nell'API Registry e renderla disponibile, sia per le implementazioni, sia per le sottoscrizioni ad opera dei client fruitori.
- L'API Gateway, applicazione cui spetta fornire le funzionalità di API management. Funge da mediatore di tutto il traffico dei servizi, tra applicazioni erogatrici e client fruitori, mettendo in atto le misure previste nei descrittori delle API presenti nel registro.
- Le applicazioni del dominio, sia nel caso di API implementation che client, con i relativi referenti tecnici.

## 3.2. I casi d'uso

Lo scenario descritto nella sezione precedente evidenzia le interazioni principali che sono relative ai casi d'uso individuati. La metodologia adottata prevede che:

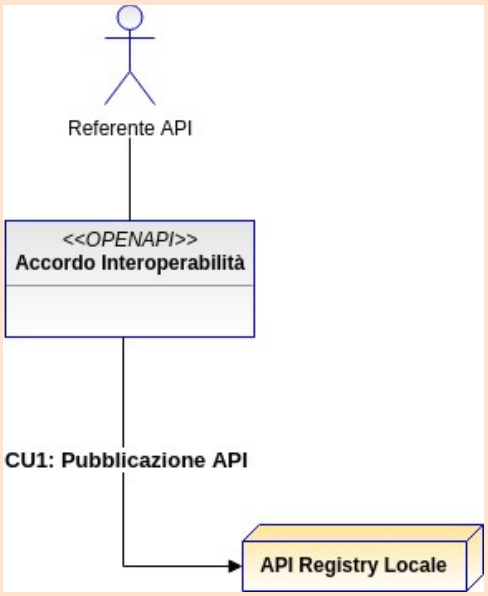
- I casi d'uso vengano descritti in dettaglio, facendo esplicito riferimento alle esigenze degli attori coinvolti.
- Per ogni singolo caso d'uso si proceda alla definizione dei requisiti che devono essere soddisfatti per il raggiungimento degli obiettivi del caso d'uso.
- Per ciascun caso d'uso possono essere proposti più pattern di interoperabilità, ciascuno dei quali rappresenta una possibile soluzione del caso d'uso, conforme ai requisiti espressi.

I casi d'uso e i pattern presentati in questa versione del documento sono da intendersi come una proposta iniziale e saranno soggetti a completamento ed evoluzione nel corso del progetto.

A partire dallo scenario applicativo INF-1 sono stati individuati i seguenti casi d'uso:

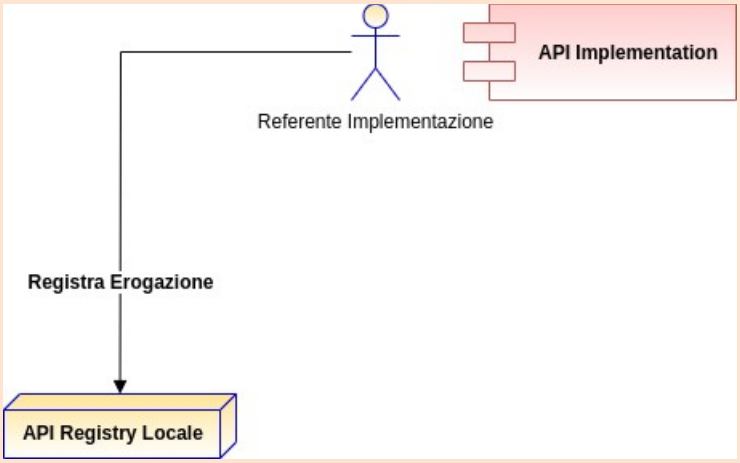
- **CU1: Pubblicazione di una API**
- **CU2: Registrazione di una API Implementation**
- **CU3: Adesione (Subscription) di una API Implementation**
- **CU4: Interazione applicativa in ricezione da domini esterni**
- **CU5: Interazione applicativa in trasmissione verso domini esterni**
- **CU6: Integrazione PDND**

Diamo nel seguito una descrizione dei casi d'uso individuati, per ciascuno dei quali saranno elencati i requisiti funzionali e i pattern di interoperabilità proposti.

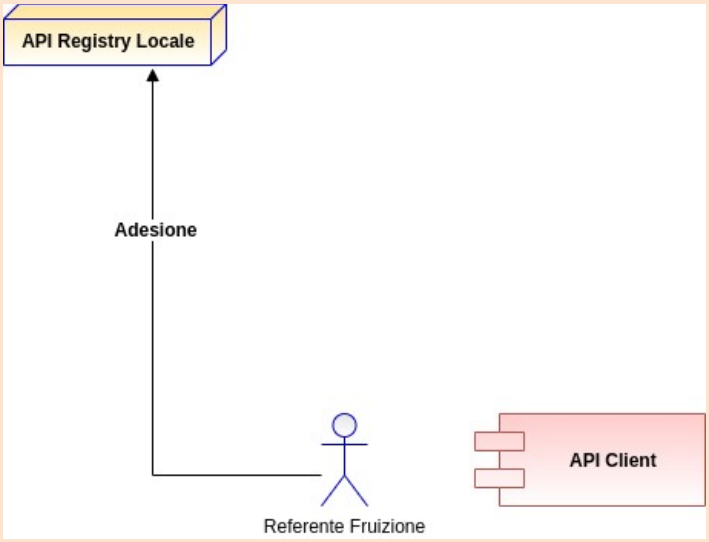
| CU1 - PUBBLICAZIONE DI UNA API | |
|---|---|
| **DESCRIZIONE** | Il Referente API produce l'accordo di interoperabilità da sottomettere per la pubblicazione sul Registro API Locale. |
| **ATTORI** | • Referente API<br>• API Registry Locale |
| **SCHEMA DI DETTAGLIO** |  |

| | |
|---|---|
| **REQUISITI** | • **CU1-R1**: Descrizione interoperabile dell'interfaccia delle API<br>• **CU1-R2**: Definizione dei criteri di autenticazione, con specifica della sicurezza sul canale, anche differenziati per singola risorsa<br>• **CU1-R3**: Definizione dei criteri di autorizzazione, con specifica degli scope richiesti, anche differenziati per singola risorsa |
| **PATTERN** | **CU1-P1: Pattern OpenAPI 3 per le API REST**<br>Questo pattern di interoperabilità soddisfa interamente i requisiti del caso d'uso nell'ambito della descrizione formale di API REST.<br>Come descritto più in dettaglio in seguito, l'adozione dello standard OpenAPI 3, ed in particolare dei nuovi strumenti mirati alla sicurezza degli scambi, consente di formalizzare gli aspetti di autenticazione ed autorizzazione delle richieste pervenute al servizio. |

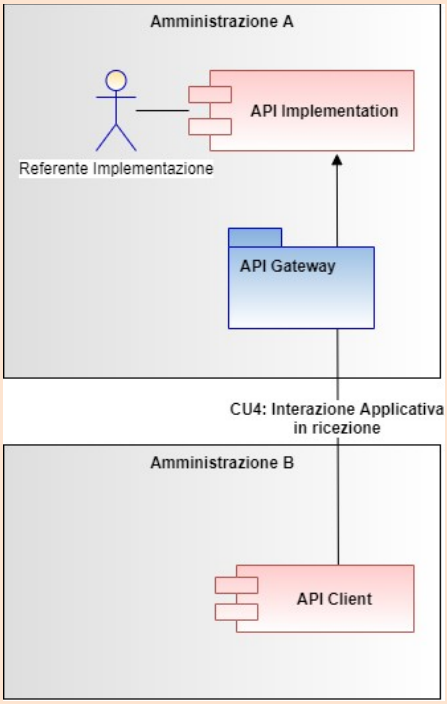
| CU2 - REGISTRAZIONE DI UNA API IMPLEMENTATION | |
|---|---|
| **DESCRIZIONE** | Il Referente dell'applicazione erogatrice formalizza sul registro API l'erogazione di una data API per il proprio ente. |
| **ATTORI** | • Referente Implementazione<br>• API Registry Locale |

| | |
|---|---|
| **SCHEMA DI DETTAGLIO** |  |
| **REQUISITI** | Da definire |
| **PATTERN** | Da definire |

| **CU3 - ADESIONE (SUBSCRIPTION) DI UNA API IMPLEMENTATION** | |
|---|---|
| **DESCRIZIONE** | Il Referente dell'applicazione client formalizza sul registro API l'adesione ad una data erogazione per il proprio ente. |
| **ATTORI** | • Referente Fruizione<br>• API Registry Locale |

| | |
|---|---|
| **SCHEMA DI DETTAGLIO** |  |
| **REQUISITI** | Da definire |
| **PATTERN** | Da definire |

| **CU4 - INTERAZIONE APPLICATIVA IN RICEZIONE DA DOMINI ESTERNI** | |
|---|---|
| **DESCRIZIONE** | L'applicazione client, del dominio dell'ente interlocutore, invoca il servizio sottoscritto, nel dominio dell'ente, attraverso la mediazione dell'API Gateway. |
| **ATTORI** | • API Implementation<br>• API Client |

| | |
|---|---|
| **SCHEMA DI DETTAGLIO** |  |
| **REQUISITI** | • **CU4-R1**: Sicurezza Canale<br>  ○ Identificazione delle amministrazioni coinvolte<br>  ○ Confidenzialità degli scambi<br>• **CU4-R2**: Sicurezza Messaggio<br>  ○ Identificazione degli applicativi coinvolti (mittente/destinatario dei messaggi)<br>  ○ Integrità dei messaggi scambiati<br>  ○ Identificazione messaggi duplicati<br>• **CU4-R3**: Non ripudio delle comunicazioni<br>  ○ Tracciatura dei dati di contesto relativi a richieste e risposte (con identità degli interlocutori)<br>  ○ Persistenza delle evidenze dei contenuti scambiati |
| **PATTERN** | Sulla base dei requisiti individuati nel caso d'uso vengono proposti i |

seguenti due pattern di interoperabilità:

- **CU4-P1: Erogazione Modi**
- **CU4-P2: Erogazione OAuth – Client Credentials**

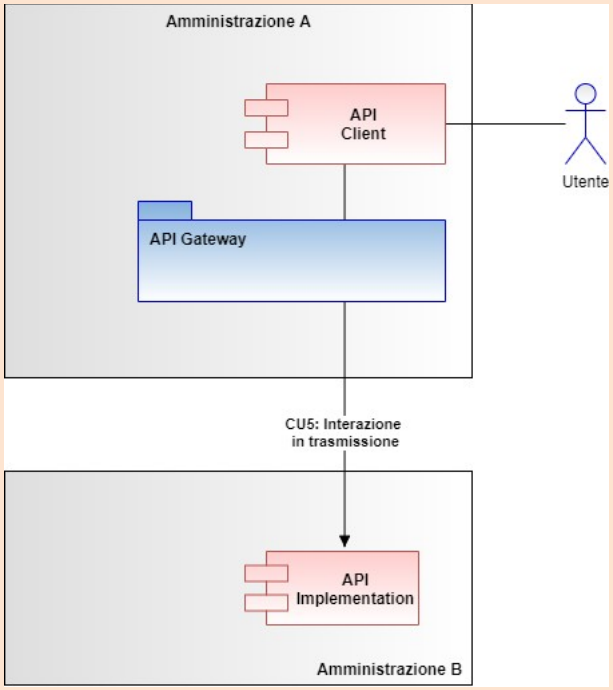
I due pattern sono entrambi calati nel contesto delle interazioni tra due sistemi applicativi, senza il coinvolgimento dell'utente. Possono essere adottati in contesti differenti, sulla base dei requisiti richiesti. Si devono pertanto tenere presenti le caratteristiche fondamentali che distinguono i due pattern:

- Il pattern "Erogazione Modi", pensato in maniera specifica per le interazioni Application-to-Application, soddisfa interamente la lista dei requisiti elencati in precedenza.
- Il pattern "Erogazione OAuth – Client Credentials":
  - se associato al protocollo SSL con mutua autenticazione, soddisfa i requisiti sulla sicurezza del canale.
  - per quanto riguarda la sicurezza messaggio, consente di soddisfare il solo requisito sull'autenticazione degli interlocutori (identificazione tramite token).
  - per quanto riguarda il non ripudio delle comunicazioni, il requisito non è soddisfatto in quanto non prevede le evidenze sui contenuti scambiati.

| **CU5: INTERAZIONE APPLICATIVA IN TRASMISSIONE VERSO DOMINI ESTERNI** | |
|---|---|
| **DESCRIZIONE** | L'applicazione client, interna al dominio dell'ente, invoca il servizio sottoscritto, nel dominio dell'ente interlocutore, attraverso la mediazione dell'API Gateway. |
| **ATTORI** | • API Implementation |

- API Client
- Utente

**SCHEMA DI DETTAGLIO**




**REQUISITI**

- **CU5-R1**: Sicurezza Canale
  - Identificazione delle amministrazioni coinvolte
  - Confidenzialità degli scambi
- **CU5-R2**: Sicurezza Messaggio
  - Identificazione degli applicativi coinvolti (mittente/destinatario dei messaggi)
  - Integrità dei messaggi scambiati
  - Identificazione messaggi duplicati
- **CU5-R3**: Non ripudio delle comunicazioni
  - Tracciatura dei dati di contesto relativi a richieste e risposte (con identità degli interlocutori)

- Persistenza delle evidenze dei contenuti scambiati

- **CU5-R4**: Autenticazione e Consenso dell'utente possessore dei diritti di accesso a risorse protette lato server
  - Identificazione dell'utente
  - Propagazione dei dati identificativi dell'utente contestualmente a quelli già previsti per gli applicativi

**PATTERN**

Sulla base dei requisiti individuati nel caso d'uso vengono proposti i seguenti pattern:

- **CU5-P1: Fruizione Modi con identificazione Utente**

  Il pattern prevede che un applicativo interno al dominio dell'ente effettui delle invocazioni di servizio su delega di un utente il quale possiede i diritti di accesso alle relative risorse protette sul server.

  Se il servizio gestore delle risorse protette fosse locale al dominio dell'ente, la relativa invocazione avverrebbe tramite autorizzazione basata sul token rilasciato al client tramite consenso dell'utente. Lo scenario attuale riguarda invece il caso in cui il servizio erogatore risieda su un dominio amministrativo esterno a quello dell'ente. Pertanto sarà necessario che la comunicazione tra i due domini rispetti la normativa vigente in materia di interoperabilità tra sistemi della PA. Questo si traduce nell'utilizzo dell'API Gateway in grado di reperire sia le informazioni di identità dell'utente sia quelle del client mantenendo inalterata la logica di invocazione lato client. Una volta acquisita la richiesta l'API Gateway si occupa di instaurare la comunicazione con il dominio esterno erogatore in accordo alla specifica Modi, garantendo inoltre la propagazione delle necessarie

informazioni di identità dell'utente di origine al fine di consentire i necessari controlli autorizzativi al sevizio destinatario.

Riguardo le rimanenti caratteristiche di sicurezza, legate a questo scambio, valgono le considerazioni già evidenziate per il pattern **CU4-P1: Erogazione Modi.**

- **CU5-P2: Fruizione Modi con identificazione dell'origine compatibile con il framework di sicurezza ANPR**

  Questo pattern ricalca il medesimo flusso già descritto in quello precedente con la differenza che si prevede l'invio al Service Provider destinatario di un pacchetto informativo integrativo che certifichi un set di dati relativi all'effettiva origine della richiesta applicativa: utente, postazione, ufficio, ente, ecc.

  In tale direzione è stato preso in esame il framework di sicurezza adottato dai comuni per l'integrazione con ANPR, nel caso di e-service SOAP. Tale framework è stato analizzato nell'ottica dell'integrazione sui flussi di comunicazione ModI, selezionando le informazioni aggiuntive e quelle ridondanti. Il pattern fornisce una soluzione specifica per il contesto del progetto ICAR.

- **CU5-P3: Fruizione OAuth2 con Asserzione JWT firmata con X.509**

  Il pattern prevede l'accesso ad una risorsa protetta nel dominio dell'erogatore, per cui è necessario approvvigionarsi di un access token, in standard OAuth2, che certifichi l'identità dell'applicazione client in base alla relazione di trust stabilita fra i due domini coinvolti.

L'autenticazione dell'applicazione chiamante si basa sull'uso di un certificato X.509. Per ottenere l'access token è necessario autenticarsi sul dominio erogatore inviando un'asserzione JWT firmata con la chiave privata associata al certificato in trust, in accordo allo standard RFC 7523 (https://datatracker.ietf.org/doc/html/rfc7523#section-2.2).

L'API Gateway predispone l'asserzione JWT ed effettua, per conto del client reale, la negoziazione del token sul dominio erogatore. Dopo che l'Authorization Server ha validato l'asserzione viene rilasciato l'access token, con al suo interno l'identità rilevata dell'applicazione chiamante. L'identificativo del client autenticato viene riportato nel claim "azp" (Authorized Party) che, da specifica JWT, rappresenta l'entità destinataria del token rilasciato. In fase di elaborazione dell'erogatore, il valore del claim "azp" risulta essenziale per l'autenticazione del client richiedente e l'applicazione delle relative politiche di autorizzazione.

Questo pattern corrisponde a quello proposto dal Ministero del Lavoro, nell'ambito del progetto di migrazione del NCN-CO, dove i client fruitori ottengono l'access token direttamente dall'Identity Cloud Service di MLPS autenticandosi tramite un JWT firmato con la coppia di chiavi associata al certificato in trust con MLPS.

L'API Gateway invia all'erogatore l'access token JWT sull'header Authorization, in accordo alla specifica MODI, consentendo a quest'ultimo di autenticare il client tramite i dati presenti nei claim opportuni.

- **CU5-P4: Fruizione API con token rilasciato da PDND**

  Il pattern prevede l'accesso ad una risorsa protetta nel

dominio dell'erogatore, per cui è necessario approvvigionarsi di un access token, in standard OAuth2, che certifichi l'identità dell'applicazione client.

I domini coinvolti sono entrambi aderenti alla PDND, inoltre è in essere un accordo di interoperabilità che prevede che l'ente interno acceda all'API sulla base di una finalità condivisa e autorizzata.

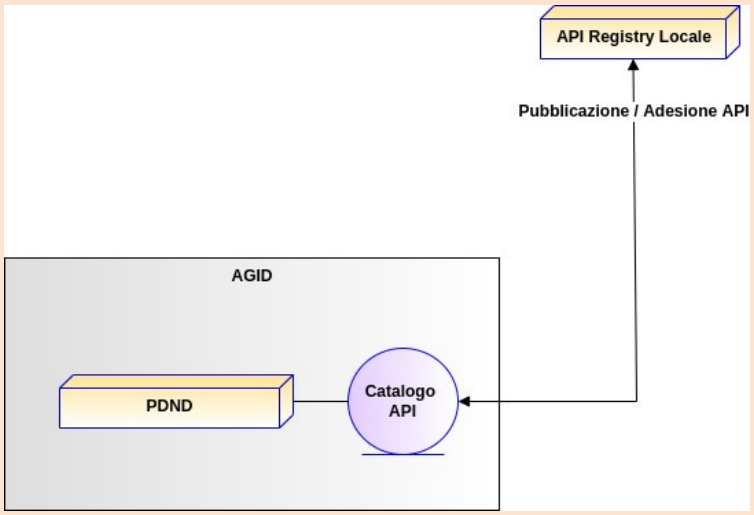
I profili di emissione del token autorizzativo sono quelli stabiliti nella RFC 6749 (Framework OAuth 2.0). L'ente, con la stipula dell'Accordo di Interoperabilità, ha preventivamente provveduto a registrare il client utilizzatore, caricando il relativo materiale crittografico necessario all'identificazione dello stesso in fase di negoziazione del token.

L'autenticazione del client sulla PDND si basa quindi sull'uso di un certificato X.509. Per ottenere l'access token è necessario autenticarsi inviando un'asserzione JWT firmata con la chiave privata associata al certificato in trust, in accordo allo standard RFC 7523. L'asserzione inviata per richiedere il token deve comprendere le seguenti informazioni a beneficio della PDND:

- Indicazione del Client Fruitore per cui si richiede l'emissione dell'Access Token.
- Indicazione dell'Accordo di interoperabilità che abilità il Client Fruitore all'accesso all'e-service dell'Erogatore.
- Indicazione della finalità associata dal Fruitore all'Accordo di interoperabilità entro cui il Client Fruitore si impegna ad utilizzare la risposta dell'e-service dell'Erogatore.

L'API Gateway predispone l'asserzione JWT ed effettua, per conto del client reale, la negoziazione del token sulla PDND.

| | |
|---|---|
| | Dopo che la PDND ha rilasciato l'access token, con al suo interno l'identità rilevata dell'applicazione chiamante, l'API Gateway invia all'erogatore l'access token JWT sull'header Authorization, in accordo alla specifica MODI, consentendo a quest'ultimo di autenticare il client tramite i dati presenti nei claim opportuni. |

| CU6 – INTEGRAZIONE PDND | |
|---|---|
| **DESCRIZIONE** | L'API Registry Locale sincronizza i dati di registrazione e adesione alle singole API, tramite i meccanismi supportati dalla PDND. |
| **ATTORI** | • API Registry Locale<br>• PDND |
| **SCHEMA DI DETTAGLIO** |  |

| | |
|---|---|
| **REQUISITI** | Da definire |
| **PATTERN** | Da definire |

## 3.3. I Pattern di Interoperabilità

In questa sezione sono descritti in dettaglio i pattern citati nelle precedenti schede dei casi d'uso.

### 3.3.1. CU1-P1: OpenAPI 3 per le API REST

Il presente pattern definisce una modalità per ottenere una descrizione formale interoperabile di API REST in conformità ai requisiti espressi per il caso d'uso "Pubblicazione di una API".
Vediamo come il pattern risponde ai requisiti espressi nel caso d'uso:

- **CU1-R1: Descrizione interoperabile dell'interfaccia delle API**

  Le regole tecniche per la stesura del descrittore dell'interfaccia sono quelle previste dallo standard OpenAPI 3, aderente alle specifiche delle linee guida AGID.

- **CU1-R2: Definizione dei criteri di autenticazione, con specifica della sicurezza sul canale, anche differenziati per singola risorsa**

  Definizione dei criteri di accesso tramite la sezione "components/securitySchemes" prevista dalla specifica OpenAPI. È possibile utilizzare gli schemi:

  - http

- apiKey
- Oauth2
- OpenIdConnect

```yaml
components:
  securitySchemes:

    BasicAuth:
      type: http
      scheme: basic

    BearerAuth:
      type: http
      scheme: bearer

    ApiKeyAuth:
      type: apiKey
      in: header
      name: X-API-Key

    OpenID:
      type: openIdConnect
      openIdConnectUrl: https://example.com/.well-known/openid-configuration

    OAuth2:
      type: oauth2
      flows:
        authorizationCode:
          authorizationUrl: https://example.com/oauth/authorize
          tokenUrl: https://example.com/oauth/token
          scopes:
            read: Grants read access
            write: Grants write access
            admin: Grants access to admin operations
```

Le definizioni di securitySchema prodotte possono essere riferite per l'utilizzo tramite la sezione "security", con la possibilità di specificarla globalmente per tutte le risorse/operazioni, oppure nel contesto di una singola risorsa/operazione.

Esempio per HTTP Bearer Auth:

```yaml
openapi: 3.0.0
...

# 1) Define the security scheme type (HTTP bearer)
components:
  securitySchemes:
    bearerAuth:            # arbitrary name for the security scheme
      type: http
      scheme: bearer
      bearerFormat: JWT    # optional, arbitrary value for documentation purposes

# 2) Apply the security globally to all operations
security:
  - bearerAuth: []         # use the same name as above
```

- **CU1-R3: Definizione dei criteri di autorizzazione, con specifica degli scope richiesti, anche differenziati per singola risorsa.**

  Definizione degli scope richiesti sulle singole risorse OpenAPI.

  I criteri di autorizzazione sono esprimibili sulla base degli scope necessari al client per l'esecuzione di determinate operazioni. Gli scope sono richiesti solo per gli schemi OAuth2 e OpenID.

  Gli scope richiesti possono essere definiti globalmente oppure a livello della singola risorsa/operazione, attraverso la sezione "security".

```yaml
security:
  - OAuth2:
      - scope1
      - scope2
  - OpenId:
      - scopeA
      - scopeB
  - BasicAuth: []
```

  Nel caso di OAuth2 gli scope espressi sono un sottoinsieme di quelli dichiarati nella sezione *securitySchemes*. Nel caso di OpenID gli scope non vengono dichiarati poiché sono risolti dinamicamente tramite la discovery URL definita sempre nella sezione *securitySchemes* con l'attributo "openIdConnectUrl".

  Per la definizione degli scopes è bene distinguere tra due distinti criteri di concessione delle autorizzazioni:

  - **Scope di Profilo Utente**

Requisiti sulla base dei quali saranno autorizzate le richieste di accesso, a determinate risorse, in funzione dei ruoli o in genere attributi relativi al profilo dell'utente (resource owner). Ad esempio: operazione consentita se l'utente ha il ruolo "Medico" oppure ha l'attributo ComuneResidenza con valore "Milano".

- **Scope di Delega Utente**

  Requisiti sulla base dei quali saranno autorizzate le richieste di accesso, a determinate risorse, in funzione di una specifica operatività delegata dall'utente (resource owner): Ad esempio: "Aggiornamento dati anagrafici" oppure "Lettura posizione lavorativa".

### 3.3.2. CU4-P1: Erogazione Modi

Questo pattern descrive il caso dell'erogazione di un servizio in accordo alla normativa Modi di AGID.

Le principali caratteristiche dello scenario sono:

1. L'erogazione del servizio verso i client è fornita attraverso il profilo di sicurezza canale IDAC02. Quindi è necessario che i domini coinvolti abbiano attuato il trust dei reciproci certificati SSL al fine di garantire la mutua autenticazione dei sistemi.
2. L'autenticazione degli applicativi, effettivamente mittente e destinatario della comunicazione, è garantita tramite l'applicazione del profilo di sicurezza messaggio IDAR02 (IDAS02 per SOAP) con l'integrazione del profilo IDAR03/IDAS03 a garanzia dell'integrità del payload scambiato.
3. Tracciatura a norma degli scambi effettuati, sia riguardo la comunicazione di richiesta che quella di risposta.
4. Conservazione delle tracce, complete delle evidenze di trasmissione, per il supporto al non ripudio.

L’elenco seguente descrive la sequenza degli scambi realizzati in questo pattern di interoperabilità:

1. Il Client Modi esterno invia la richiesta di servizio conforme alle specifiche Modi.
2. Il Reverse Proxy termina la comunicazione SSL occupandosi dell'autenticazione del client tramite il repository dei certificati "trusted".
3. Il Reverse Proxy, attuando le eventuali politiche di load balancing, inoltra la richiesta pervenuta, insieme al cerificato SSL ricevuto, all'API Gateway.
4. L'API Gateway, tramite il certificato SSL ricevuto, identifica il dominio mittente attuando eventuali politiche di autorizzazione.
5. L'API Gateway effettua la validazione del token Modi, per la sicurezza messaggio, tramite il repository dei certificati degli applicativi "trusted". Tali controlli sono volti all'autenticazione dell'applicativo mittente ed alla verifica dell'integrità del payload.
6. L'API Gateway effettua il tracciamento dei dati inerenti la richiesta con tutti i dati estratti nel corso dell'elaborazione effettuata. La traccia della richiesta comprende anche il token Modi originale inviato dal mittente (ai fini del non ripudio).
7. L'API Gateway inoltra la richiesta al servizio erogato (API Modi Implementation). La richiesta inviata è corredata da un elemento contenente i dati di integrazione che comprende le informazioni ricavate dal token Modi, accessibili in maniera semplice dal servizio. In opzione è possibile configurare l'inoltro al servizio del token originale.
8. Il servizio erogatore elabora la richiesta.
9. Il servizio erogatore invia la risposta all'API Gateway.
10. L'API Gateway produce il token Modi da inserire nella risposta avvalendosi di un Keystore relativo agli applicativi interni.

11. L'API Gateway effettua il tracciamento dei dati inerenti la risposta, mantenendo la correlazione alla richiesta ed includendo il token prodotto al passo precedente.
12. L'API Gateway inoltra la risposta, completa di token Modi, al Reverse Proxy.
13. Il Reverse Proxy chiude la comunicazione HTTPS girando la risposta al Client Modi.

### 3.3.3. CU4-P2: Erogazione OAuth - Client Credentials

Questo pattern descrive il caso dell'erogazione di un servizio accessibile alle applicazioni Client tramite il processo di autorizzazione "Client Credentials", previsto nell'ambito del protocollo OAuth2. In questo caso, gli applicativi client richiedono l'accesso a risorse protette per cui è necessario esibire un token di autorizzazione rilasciato da un Authorization Server.

Le principali caratteristiche di questo scenario sono:

- La fase di approvvigionamento del token, richiesto dai client all'Authorization Server, tramite le relative credenziali.
- La validazione base del token, da parte del Reverse Proxy, al fine di scartare le richieste con token non validi, evitando di inoltrarle all'API Gateway.
- La richiesta di ulteriori informazioni all'Authorization Server, da parte dell'API Gateway, tramite i servizi di introspection e user info.
- La tracciatura a norma degli scambi, in ingresso e uscita, comprensiva dei dati di autorizzazione acquisiti dai token.

L’elenco seguente descrive la sequenza degli scambi realizzati in questo pattern di interoperabilità:

1. Il Client richiede l'access token all'Authorization Server fornendo le proprie credenziali (ClientID e Secret).
2. L'Authorization Server verifica le credenziali del Client.
3. L'Authorization Server restituisce l'access token al Client.

4. Il Client invia la richiesta all'endpoint del servizio includendo l'access token ottenuto al passo precedente.
5. Il Reverse Proxy riceve la richiesta del Client ed effettua l'handshake SSL.
6. Il Reverse Proxy verifica la validità del token OAuth incluso nella richiesta del Client.
7. Il Reverse Proxy, superata la validazione del token, inoltra la richiesta all'API Gateway.
8. L'API Gateway effettua la validazione del token OAuth.
9. L'API Gateway richiede eventuali informazioni aggiuntive all'Authorization Server, tramite i servizi di Introspection e UserInfo.
10. L'Authorization Server restituisce le informazioni aggiuntive richieste.
11. L'API Gateway effettua il tracciamento della richiesta.
12. L'API Gateway inoltra la richiesta al servizio erogatore completando il messaggio originale con i dati di integrazione, insieme di informazioni relative al contesto di indirizzamento al servizio, per un più rapido accesso da parte del destinatario.
13. L'applicativo erogatore effettua i propri controlli di autorizzazione avvalendosi del token originale o, in alternativa, dei dati di integrazione forniti dall'API Gateway.
14. Superati i propri controlli di autorizzazione, l'applicativo erogatore elabora la richiesta.
15. L'applicativo erogatore restituisce la risposta applicativa.
16. L'API Gateway effettua il tracciamento della risposta.
17. L'API Gateway inoltra la risposta applicativa, in risposta al Reverse Proxy, includendo i dati di integrazione a beneficio del Client.
18. Il Reverse Proxy chiude la comunicazione HTTPS girando la risposta al Client.

### 3.3.4. CU5-P1: Fruizione Modi con Identificazione Utente

Questo pattern descrive il caso della fruizione di un servizio, esposto da un dominio amministrativo esterno, in accordo alla normativa Modi di AGID, assicurando l'identificazione dell'utente che origina la richiesta tramite l'uso di token.

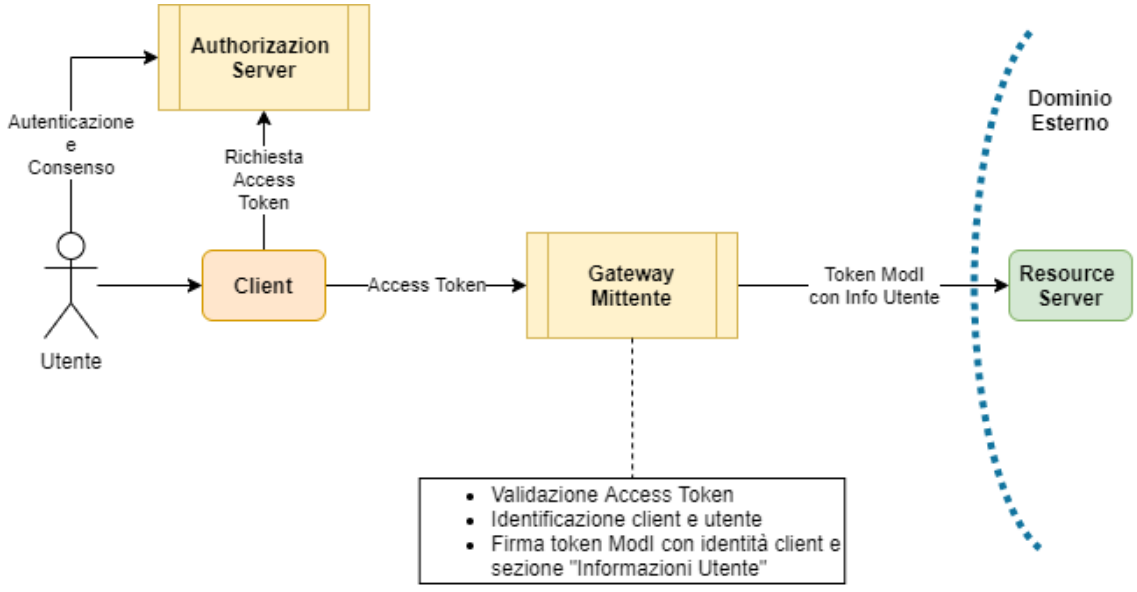
Le principali caratteristiche dello scenario sono:

- L'accesso ai client è consentito solo previa presentazione di un access token, il quale viene rilasciato previa autenticazione e consenso da parte dell'utente coinvolto. L'access token viene utilizzato dal gateway per identificare l'utente ed autenticare l'applicativo client chiamante.
- La fruizione del servizio prevede l'uscita dal dominio attraverso il profilo di sicurezza canale IDAC02. Quindi è necessario che i domini coinvolti abbiano attuato il trust dei reciproci certificati SSL al fine di garantire la mutua autenticazione dei sistemi.
- L'autenticazione degli applicativi, effettivamente mittente e destinatario della comunicazione, è garantita ai fini dello scambio tra i due domini tramite l'applicazione del profilo di sicurezza messaggio IDAR02 (IDAS02 per SOAP) con l'integrazione del profilo IDAR03/IDAS03 a garanzia dell'integrità del payload scambiato.
- Tracciatura a norma degli scambi effettuati, sia riguardo la comunicazione di richiesta che quella di risposta.
- Conservazione delle tracce, complete delle evidenze di trasmissione, per il supporto al non ripudio.

L'elenco seguente descrive la sequenza degli scambi realizzati in questo pattern di interoperabilità:

4. Il Client invia la richiesta di servizio all'API Gateway dopo aver negoziato l'access token con l'intervento dell'utente coinvolto nell'operazione.

- L'API Gateway, tramite il token ricevuto, identifica il client, attuando eventuali politiche di autorizzazione, quindi estrae il profilo dell'utente dal contenuto del token.
- L'API Gateway, tramite l'identità del client, utilizza le relative credenziali SSL per produrre il token Modi, comprensivo del profilo utente, da inviare al dominio destinatario.
- L'API Gateway effettua il tracciamento dei dati inerenti la richiesta con tutti i dati estratti nel corso dell'elaborazione effettuata. La traccia della richiesta comprende anche il token Modi originale inviato al destinatario.
- L'API Gateway inoltra la richiesta al servizio erogato, tramite il Reverse Proxy, e riceve la risposta cui vengono applicate analoghe elaborazioni a quelle già descritte nel caso dell'erogazione.




### 3.3.5. CU5-P2: Fruizione Modi con identificazione dell'origine compatibile con il framework di sicurezza ANPR

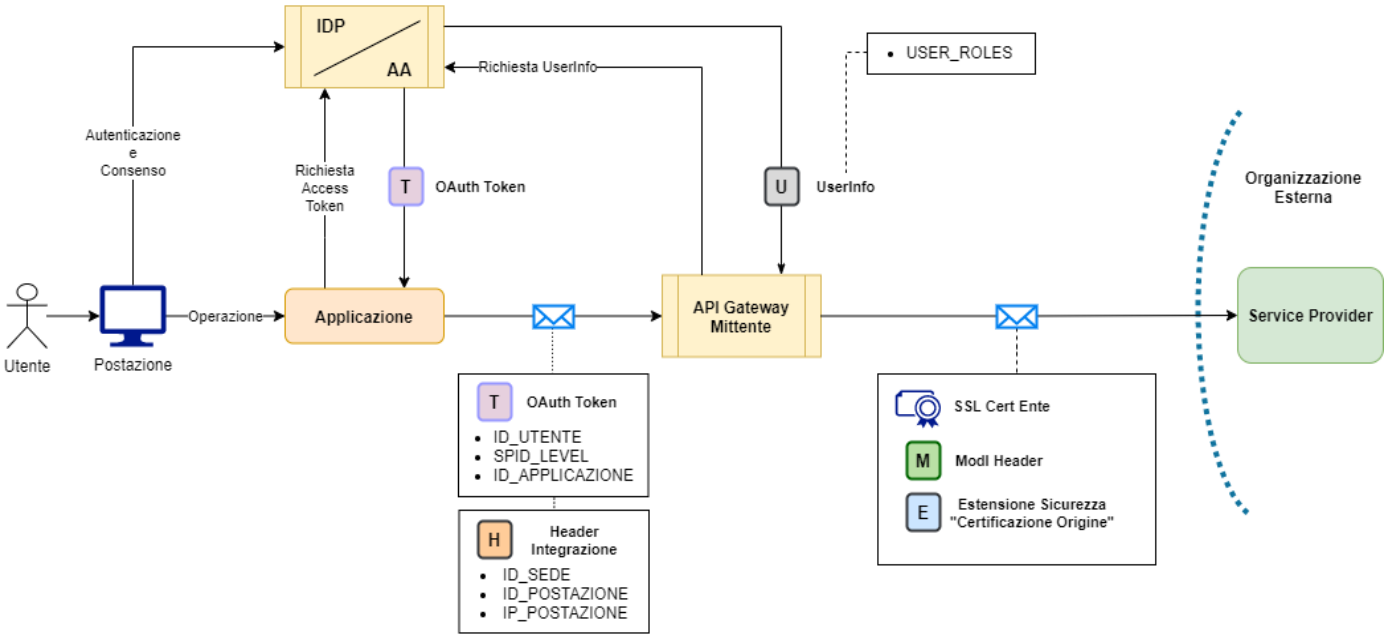
Questo pattern descrive il caso della fruizione di un servizio, esposto da un dominio amministrativo esterno, in accordo alla normativa Modi di AGID, assicurando la

trasmissione di un elemento aggiuntivo di sicurezza, compatibile con il framework già adottato per ANPR, per la certificazione dell'identità dell'utente e contesto di origine della richiesta applicativa.

Il pattern nasce dall'esigenza di adeguare il formato del messaggio previsto in ModI alle esigenze di alcuni servizi, in particolare facendo riferimento all'esperienza del progetto ANPR, di ottenere informazioni dettagliate sull'origine della richiesta ricevuta da un Ente terzo (id_sede, id_postazione, id_applicazione, id_utente, livello di autenticazione effettuato dall'utente, ruolo assegnato all'utente rispetto al servizio richiesto).

Le principali caratteristiche dello scenario sono simili al pattern descritto nella sezione precedente con i seguenti elementi in integrazione, prevedendo però un'estensione dell'header ModI per consentire l'integrazione dei dati aggiuntivi di identificazione dell'origine della richiesta. Il formato dell'estensione si caratterizza in due modalità distinte a seconda che l'interazione sia basata su SOAP oppure REST.

La figura seguente descrive le caratteristiche principali del pattern, evidenziando gli elementi specifici relativi al framework di sicurezza.




Gli elementi che caratterizzano questo pattern sono i seguenti:

- L’utente, dalla propria postazione, richiede l’operazione all’applicazione del dominio dell’ente.
- L’applicazione richiede all’IDP/AA il token Oauth per l’accesso al servizio tramite l’API Gateway Mittente. Tale richiesta causa le redirezione del flusso di navigazione dell’utente sul processo di autenticazione.
- L’utente si autentica con SPID e fornisce il consenso per l’esecuzione dell’operazione che lo riguarda.
- L’applicazione ottiene il token e lo utilizza per effettuare la richiesta all’API Gateway Mittente. Il token Oauth conterrà claim specifici relativi all’identità dell’utente e applicazione chiamante certificate dall’IDP/AA con il processo di autenticazione.
- L’applicazione fornisce ulteriori dati, relativi al contesto di invocazione (sede, ufficio, postazione) tramite un header di integrazione stabilito e condiviso con l’API Gateway Mittente.
- L’API Gateway Mittente estrae i dati identificativi dal token fornito e quelli eventualmente presenti nell’header di integrazione. Utilizza, ove previsto, la funzionalità “UserInfo” per richiedere all’IDP/AA ulteriori attributi relativi al contesto di autenticazione, ad esempio i ruoli posseduti dall’utente.
- L’API Gateway Mittente effettua la richiesta al Service Provider, in accordo al protocollo ModI, inserendo nella richiesta le informazioni aggiuntive necessarie all’identificazione dell’origine, con una modalità che dipende dal protocollo di servizio adottato:
    - Nel caso SOAP, nell’ambito dell’header WS-Security già previsto da ModI, le informazioni vengono inserite nel formato di un’asserzione SAML.
    - Nel caso REST, nell’ambito del JWT già previsto da ModI, le informazioni vengono inserite come claims integrativi.
- I dati identificativi che vengono trasmessi al dominio destinatario sono il risultato di quanto già previsto dall’header ModI nativamente e di

un’ulteriore estensione che recepisce lo schema di sicurezza ANPR. La seguente tabella presenta nel complesso lo schema di sicurezza e il dettaglio dei dati contenuti:

| Dato di sicurezza | Fonte | Collocazione | Descrizione |
|---|---|---|---|
| **Issuer** | API Gateway Mittente | Estensione Sicurezza (Token SAML o JWT) | Identificativo di chi certifica i dati trasmessi. Tipicamente è l’identificativo dell’API Gateway |
| **Id_Ente** | API Gateway Mittente | Header standard ModI | Identificativo dell’ente mittente. Tale dato viene ricavato dall’API Gateway mittente tramite l’identità dell’applicazione ed utilizzato per la selezione del certificato SSL in uscita |
| **Id_Sede** | Applicazione Mittente | Estensione Sicurezza (Token SAML o JWT) | Identificativo dell’AOO o UO dell’ente da cui origina la richiesta. Tale informazione, a seconda dei contesti di applicazione, potrebbe essere desunta dal certificato di firma previsto a livello dai profili di sicurezza a livello del messaggio, oppure inserita |

| Dato di sicurezza | Fonte | Collocazione | Descrizione |
|---|---|---|---|
| | | | appositamente in altri casi. |
| **Id_Postazione** | Applicazione Mittente | Estensione Sicurezza (Token SAML o JWT) | Identificativo della postazione dell'utente che ha dato origine all'operazione, se applicabile. |
| **Ip_Postazione** | Applicazione Mittente | Estensione Sicurezza (Token SAML o JWT) | Indirizzo IP della postazione dell'utente. |
| **Id_Utente** | Oauth Token (IdP) | Estensione Sicurezza (Token SAML o JWT) | Identificativo dell'utente che ha superato il processo di autenticazione. |
| **Livello_SPID** | Oauth Token (IdP) | Estensione Sicurezza (Token SAML o JWT) | Livello SPID utilizzato per il processo di autenticazione. |
| **Id_Applicazione** | Oauth Token (IdP) | Header standard ModI | Identificativo dell'applicazione che esegue l'operazione in delega per l'utente. Tipicamente si tratta del CliendID utilizzato per il riconoscimento |

| Dato di sicurezza | Fonte | Collocazione | Descrizione |
|---|---|---|---|
| | | | dell'applicazione in Oauth. |
| **Ruoli Utente** | UserInfo (IdP) | Estensione Sicurezza (Token SAML o JWT) | Elenco dei ruoli posseduti dall'utente e assegnati dall'IDP/AA al termine dell'autenticazione. |

L'esempio seguente mostra l'asserzione SAML prodotta dall'API Gateway mittente, dopo aver recuperato tutte le informazioni sopra indicate, per l'integrazione all'header ModI sulla richiesta in uscita.

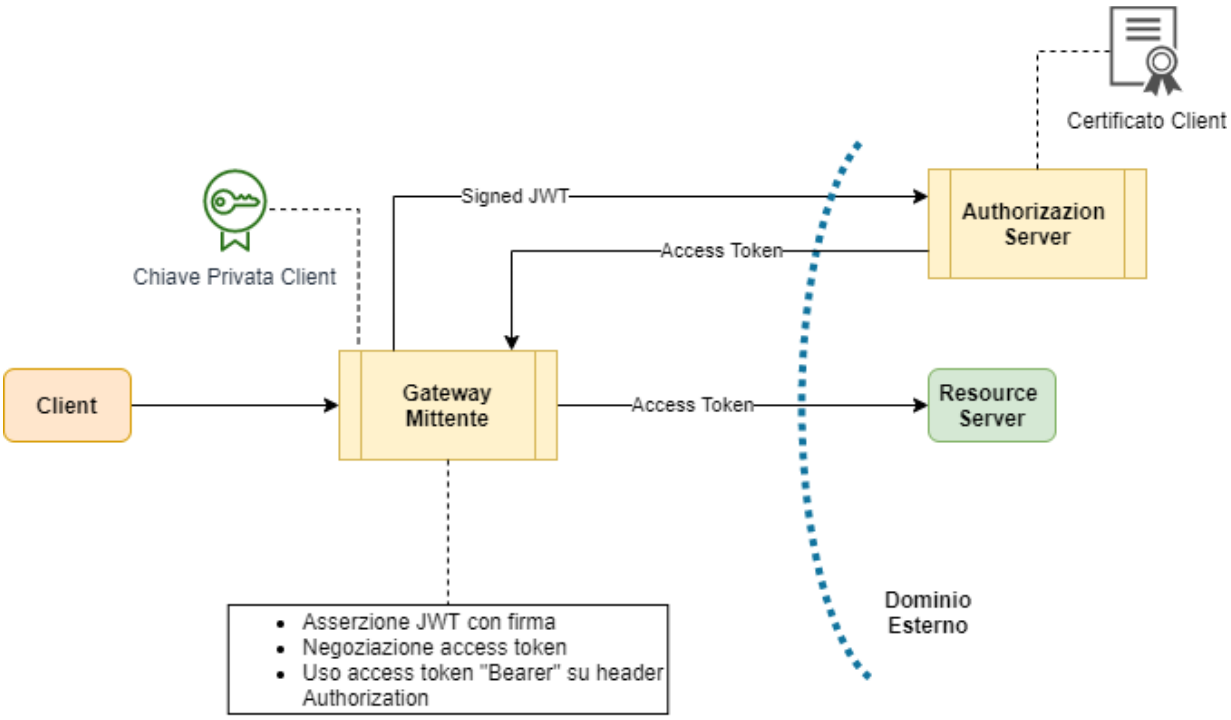
```
<saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ID="_00103c0b-5f9f-44d2-a695-
519c5133d34a" IssueInstant="2020-07-02T15:53:53.753Z" Version="2.0"
xsi:type="saml2:AssertionType">
      <saml2:Issuer>GW-RT</saml2:Issuer>
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
            ...
      </ds:Signature>
      <saml2:Subject>
            <saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified">ID_UTENTE</saml2:NameID>
            <saml2:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                  <saml2:SubjectConfirmationData NotBefore="2020-07-
02T15:53:52.895Z" NotOnOrAfter="2020-07-02T15:58:52.895Z"/>
            </saml2:SubjectConfirmation>
      </saml2:Subject>
      <saml2:Conditions NotBefore="2020-07-02T15:53:53.763Z" NotOnOrAfter="2020-07-
02T15:58:53.763Z"/>
      <saml2:AuthnStatement AuthnInstant="2020-07-02T15:53:52.895Z"
SessionNotOnOrAfter="2020-07-02T15:58:52.895Z">
            <saml2:AuthnContext>
      <saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:SpidL2</
saml2:AuthnContextClassRef>
            </saml2:AuthnContext>
      </saml2:AuthnStatement>
      <saml2:AttributeStatement>
            <saml2:Attribute Name="IdSede"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
                  <saml2:AttributeValue
xsi:type="xsd:string">ID_SEDE</saml2:AttributeValue>
            </saml2:Attribute>
            <saml2:Attribute Name="IdPostazione"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
                  <saml2:AttributeValue
xsi:type="xsd:string">ID_POSTAZIONE</saml2:AttributeValue>
            </saml2:Attribute>
            <saml2:Attribute Name="IpPostazione"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
                  <saml2:AttributeValue
xsi:type="xsd:string">IP_POSTAZIONE</saml2:AttributeValue>
            </saml2:Attribute>
            <saml2:Attribute Name="IdApplicazione"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
                  <saml2:AttributeValue
xsi:type="xsd:string">ID_APPLICAZIONE</saml2:AttributeValue>
            </saml2:Attribute>
            <saml2:Attribute Name="Role"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
                  <saml2:AttributeValue
xsi:type="xsd:string">UTENTE_RUOLO_1</saml2:AttributeValue>
                  ...
                  <saml2:AttributeValue
xsi:type="xsd:string">UTENTE_RUOLO_N</saml2:AttributeValue>
      </saml2:AttributeStatement>
</saml2:Assertion>
```

### 3.3.6. CU5-P3: Fruizione OAuth2 con Asserzione JWT firmata con X.509

Questo pattern descrive il caso della fruizione di un servizio, esposto da un dominio amministrativo esterno, per il cui accesso è necessario utilizzare un token di autenticazione rilasciato dall'erogatore. Si tratta di uno scenario proposto dal Ministero del Lavoro e successivamente incluso tra i pattern di interoperabilità della specifica MODI.

Le principali caratteristiche dello scenario sono:

1. Condivisione con il dominio erogatore del certificato client ai fini del trust sull'authorization server.
2. Negoziazione dell'access token in carico all'API Gateway, il quale si occupa di predisporre l'asserzione JWT firmata con la chiave privata associata al certificato X.509 client da inviare all'authorization server per l'autenticazione.
3. Rilascio dell'access token all'API Gateway contenente i dati identificativi del client. Per il passaggio dell'identità del client, vero destinatario del token rilasciato, viene valorizzato il claim "azp" (rappresenta l'applicativo autorizzato nella specifica JWT). Il token è firmato dall'authorization server del dominio erogatore che l'ha rilasciato.
4. Utilizzo dell'access token da parte dell'API Gateway che lo invia secondo lo schema conforme a ModI nella successiva richiesta diretta all'API che gestisce la risorsa protetta.
5. Dopo aver validato il token, l'erogatore verifica l'identità del client, presente nel claim "azp" del token e quindi applica i relativi criteri di autorizzazione e conseguente rilascio degli scope richiesti.




### 3.3.7. CU5-P4: Fruizione API con token rilasciato da PDND

Questo pattern descrive il caso della fruizione di un servizio, esposto da un dominio amministrativo esterno, per il cui accesso è necessario utilizzare un token di autenticazione rilasciato dalla PDND.

Le principali caratteristiche del pattern sono:

1. **Invio della richiesta da parte del client all'API Gateway**

   L'applicazione client invia la richiesta al servizio esterno utilizzando l'endpoint messo a disposizione dall'API Gateway. È necessaria l'autenticazione dell'applicazione client sull'API Gateway, il quale dovrà completare l'identificazione al fine di poter selezionare le corrette credenziali necessarie per la negoziazione del voucher sulla PDND. L'autenticazione dell'applicazione client può avvenire tramite i classici sistemi: https, http-basic o apikey.

2. **Negoziazione del voucher sulla PDND**

L'API Gateway, dopo aver identificato l'applicazione chiamante internamente, utilizza le relative credenziali per negoziare il voucher sulla PDND.

3. **Invocazione del servizio con utilizzo del voucher**

   L'API Gateway utilizza il voucher ottenuto dalla PDND nella successiva richiesta al Servizio Erogato (Resource Server) che gestisce la risorsa protetta.

4. **Elaborazione della richiesta da parte del servizio erogato**

   Il Servizio Erogato valida il voucher fornito, tramite il certificato della PDND, e quindi, dopo aver acquisito l'identità dell'applicazione client, applica i relativi criteri di autorizzazione e di conseguenza gli scope richiesti.