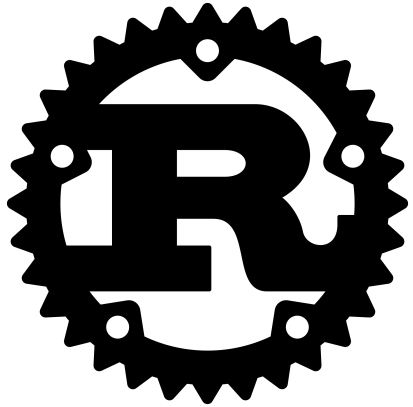
Secure Photo Hub

Rust REST API per memorizzare e gestire in maniera sicura foto e albums

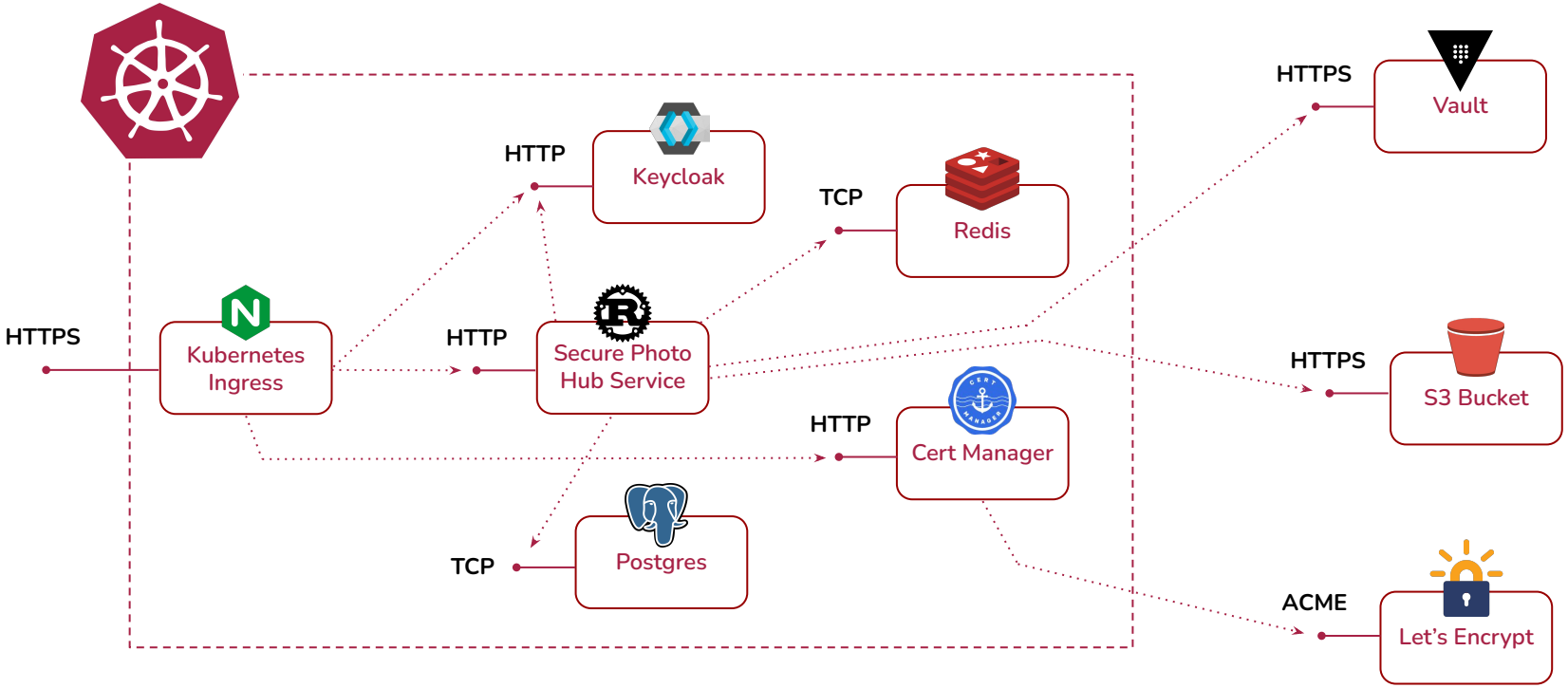
Vincenzo Tramo
N97000433

24 Febbraio, 2025
Università di Napoli, Federico II
Secure Systems Design



https://github.com/vtramo/secure-photo-hub

Architettura



Descrizione applicazione e ruoli

Gli utenti della REST API hanno la possibilità di caricare, gestire, visualizzare e modificare le proprie immagini all'interno di album.

Gli utenti possono assegnare dei livelli di visibilità alle foto e agli album (*PRIVATE*, *SHARED*, *PUBLIC*).

Esistono due ruoli: *BasicUser* e *Administrator*.

Un Administrator può visualizzare ogni risorsa, indipendentemente dalla visibilità.

Un BasicUser può visualizzare tutte le risorse pubbliche, ma non quelle private degli altri utenti. Può eseguire ogni operazione sulle proprie risorse private.

OAuth2, OPENID e Keycloak

Main goals: confidentiality, availability, integrity of data, accountability, and authenticity.

Utilizzo di *OAuth2*, *OpenID Connect* e *Keycloak* per rendere sicura le REST API.

Keycloak viene anche utilizzato per *centralizzare le politiche di controllo degli accessi*, sfruttando una combinazione di *RBAC (Role-Based Access Control)* e *ABAC (Attribute-Based Access Control)*, insieme a OAuth2 e OpenID Connect.

La REST API è scritta nel linguaggio di programmazione *Rust* e implementa *da zero* la parte client OAuth2/OpenID Connect, nonché l'interfaccia di comunicazione con Keycloak per prendere decisioni di autorizzazione di accesso alle risorse (conosciuto anche come *Policy Enforcer*). Per questo motivo, è stato necessario progettare interfacce di programmazione ben organizzate per forzare le politiche di accesso alle risorse in modo efficace.

Scopo degli altri componenti

- ***Kubernetes Ingress (nginx)***: fornisce un punto di accesso centralizzato per il cluster Kubernetes, esponendo un endpoint HTTPS sicuro e gestendo il routing del traffico verso i vari servizi interni del cluster, in base a regole configurabili.

- ***Redis***: sistema di cache in-memory basato su key-value, utilizzato per memorizzare in modo rapido e sicuro le sessioni degli utenti, migliorando le performance.

- ***Postgres***: database relazionale utilizzato come supporto per l'applicazione, memorizzando riferimenti e metadati delle immagini anziché le immagini stesse, garantendo una gestione efficiente e scalabile dei dati strutturati.

- ***Cert-manager con Let's Encrypt***: cert-manager è uno strumento di automazione per la gestione dei certificati SSL/TLS in Kubernetes. Integrazione con Let's Encrypt permette di ottenere, rinnovare e gestire automaticamente certificati SSL/TLS gratuiti per i domini, garantendo una comunicazione sicura tra i servizi esposti nel cluster Kubernetes e gli utenti finali.

Scopo degli altri componenti

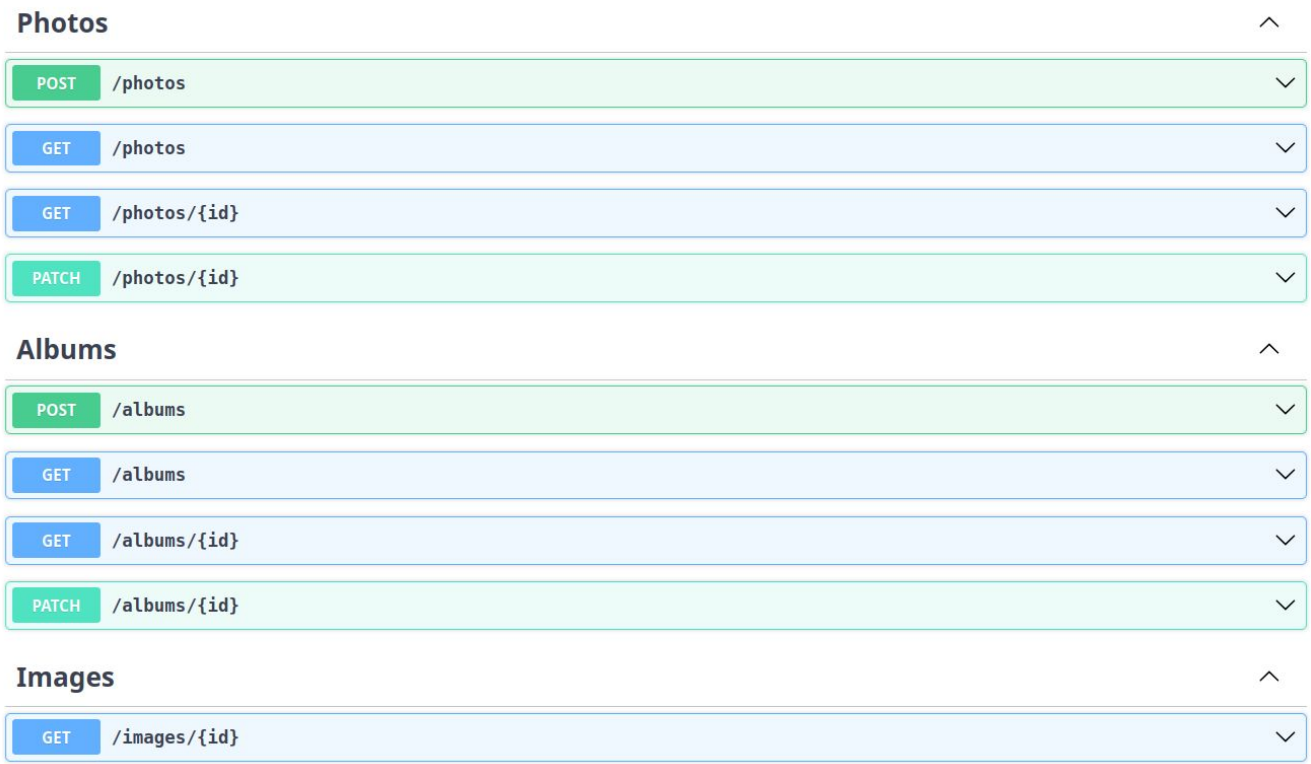
- ***S3 Bucket:*** utilizzato per memorizzare le immagini in modo sicuro, garantendo la crittografia dei file sia in transito che a riposo.

- ***Vault:*** utilizzato per memorizzare e fornire i secrets all'applicazione al momento dell'avvio, previa autenticazione dell'applicazione stessa. Vault non è presente internamente nel cluster. Separare Vault dal cluster riduce i rischi legati a un potenziale compromesso del cluster stesso. Se Vault fosse all'interno del cluster, un attacco che compromette il cluster potrebbe compromettere anche la sicurezza dei secrets e delle credenziali gestite da Vault. Inoltre, mantenendo Vault esterno, si riduce la superficie di attacco e si aumenta la resilienza complessiva, poiché Vault può essere configurato per essere più sicuro e indipendente dal ciclo di vita del cluster Kubernetes.

Endpoint esposti dall’applicazione

L’applicazione Secure Photo Hub espone degli endpoint REST per la creazione, visualizzazione e modifica di foto e album.

La risorsa Image è l’immagine vera e propria, mentre la risorsa Photo è un riferimento all’immagine e arricchisce quest’ultima con dei metadati come titolo, dimensione, descrizione e così via…

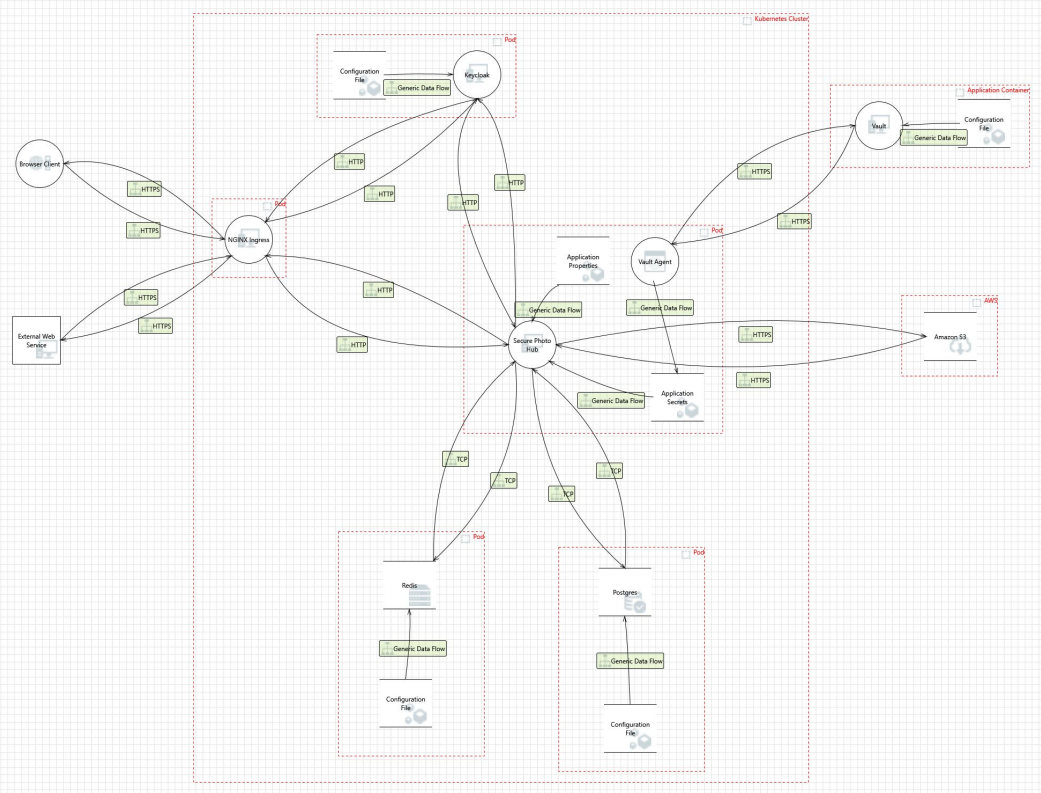


https://github.com/vtramo/secure-photo-hub/blob/main/openapi.yaml

Struttura della presentazione

1. Threat Model (con Microsoft Threat Modeling Tool)
2. Famiglia di controlli:
    - **IA**
    - **AC** (insieme a Policy Enforcers di Keycloak e gestione secrets Vault)
    - **AU**
    - **SC**
3. Come sono state gestite le configurazioni dell'applicazione
    - OIDC, Redis, AWS S3, Postgres
4. Sequence Diagrams flussi di autenticazione (OAuth2)
    - Access Token validation flow
    - Authorization Code flow
5. Sequence Diagrams flussi di autorizzazione (Policy Enforcers)
6. Come è stato creato e configurato in maniera sicura AWS S3

Microsoft Threat Modeling Tool



IA-2 IA-8 IDENTIFICATION AND AUTHENTICATION

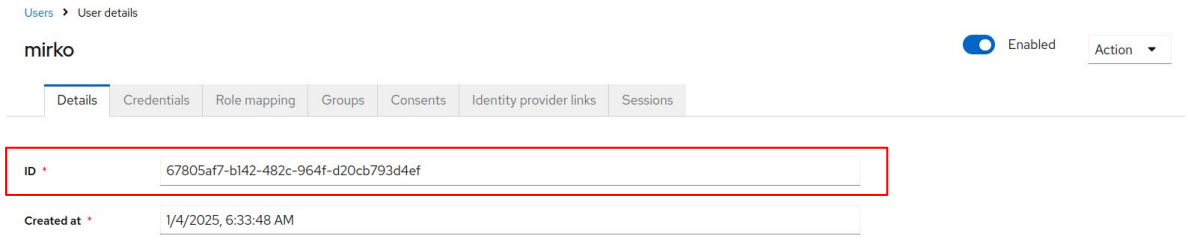
(and non-organizational users)

Control: Uniquely identify and authenticate organizational users and associate that unique identification with processes acting on behalf of those users.

- **Identificazione univoca:** Ogni utente, sia esso interno o esterno all'organizzazione, riceve un identificativo univoco generato da Keycloak al momento della registrazione. Questo identificativo garantisce che ogni utente sia univocamente riconoscibile all'interno del sistema.

- **Autenticazione e autorizzazione:** L'identità di ciascun utente viene verificata attraverso i processi di autenticazione e autorizzazione forniti dai protocolli OAuth2 e OpenID Connect.

- **Tracciamento azioni:** Tutte le azioni tracciate eseguite da un utente nel sistema sono collegate al suo identificativo univoco.



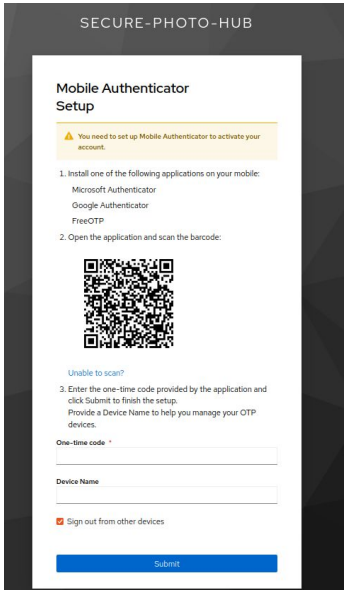

```
CREATE TABLE photos(
   id uuid NOT NULL,
   PRIMARY KEY(id),
   title TEXT NOT NULL,
   description TEXT NOT NULL,
   visibility visibility NOT NULL,
   owner_user_id uuid NOT NULL,
   tags TEXT[] NOT NULL DEFAULT '{}'::text[],
   category TEXT NOT NULL DEFAULT '',
   album_id uuid,
   image_id uuid NOT NULL
       REFERENCES images(id)
       ON DELETE CASCADE,
   is_deleted boolean NOT NULL DEFAULT false,
   created_at timestamptz NOT NULL DEFAULT NOW()
);
```

IA-2 IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS)

Control Enhancements:

(1) IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS) | MULTI-FACTOR AUTHENTICATION TO PRIVILEGED ACCOUNTS

Implement multi-factor authentication for access to privileged accounts.



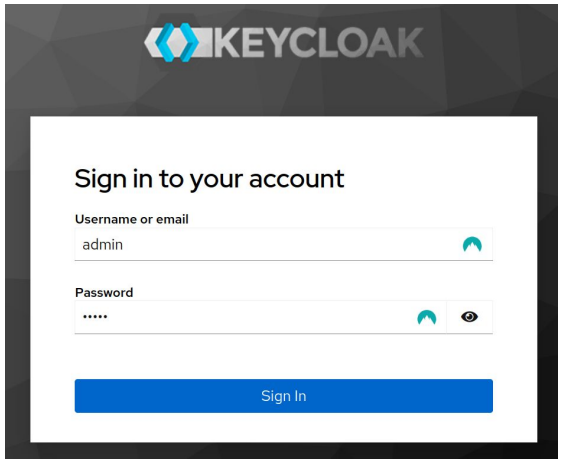
E’ stato configurato Keycloak per abilitare TOTP (*Time-Based One-Time Password*) soltanto per gli utenti con ruolo Administrator

- in base alla presenza o meno di un attributo utente *enable-totp*

IA-6 AUTHENTICATION FEEDBACK

Control: Obscure feedback of authentication information during the authentication process to protect the information from possible exploitation and use by unauthorized individuals.

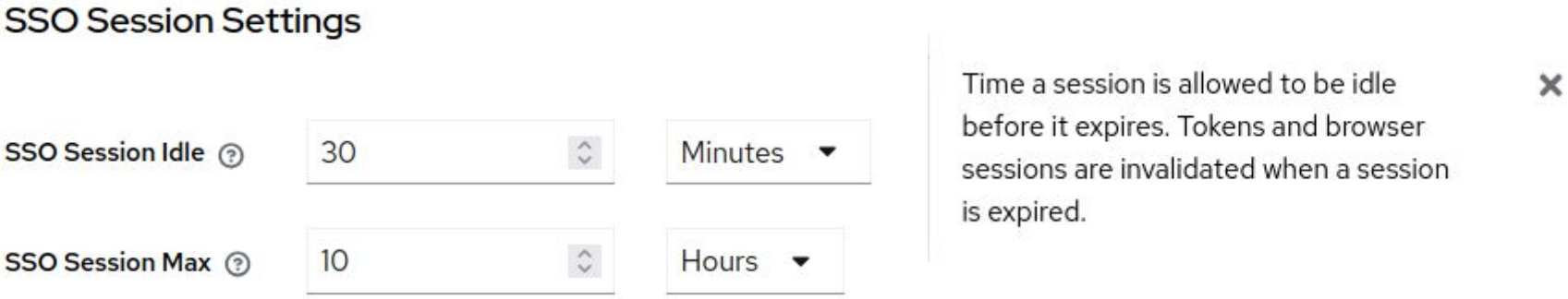
- **Protezione del feedback di autenticazione:** Keycloak protegge le informazioni di autenticazione (come password, codici PIN o altri dati sensibili) durante il processo di inserimento, per evitare che possano essere sfruttate da individui non autorizzati. Questo viene fatto oscurando o limitando la visualizzazione di tali informazioni di solito tramite degli asterischi.



IA-11 RE-AUTHENTICATION

Control: Require users to re-authenticate when [*Assignment: organization-defined circumstances or situations requiring re-authentication*].

- **Riautenticazione dopo 30 minuti di inattività:**
  - Dopo 30 minuti di inattività, sia l'Access Token che il Refresh Token scadono.
  - L’applicazione che agisce per conto dell’utente deve richiedere all'utente di riautenticarsi per generare nuovi token validi.



AC-3 **ACCESS ENFORCEMENT**

Control: Enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

Il sistema utilizza **Keycloak** per:

- proteggere le risorse tramite policy di autorizzazione granulari e differenti meccanismi di controllo degli accessi:
  - RBAC (Role-Based Access Control)
  - ABAC (Attribute-Based Access Control)
- gestione centralizzata di risorse, policy e permessi
- *Policy Decision Point* (PDP) centralizzato

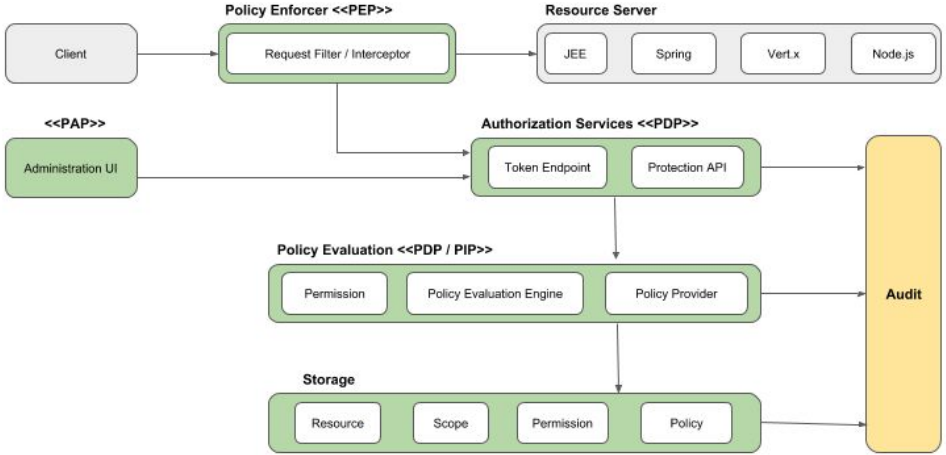
Queste funzionalità sono fornite da Keycloak sotto il nome di **Authorization Services**

AC-3 ACCESS ENFORCEMENT

Control: Enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

Keycloak Authorization Services Architecture:

- Il sistema Secure Photo Hub implementa un **Policy Enforcer (PEP)** personalizzato che comunica con Keycloak per forzare le decisioni di autorizzazione lato Resource Server



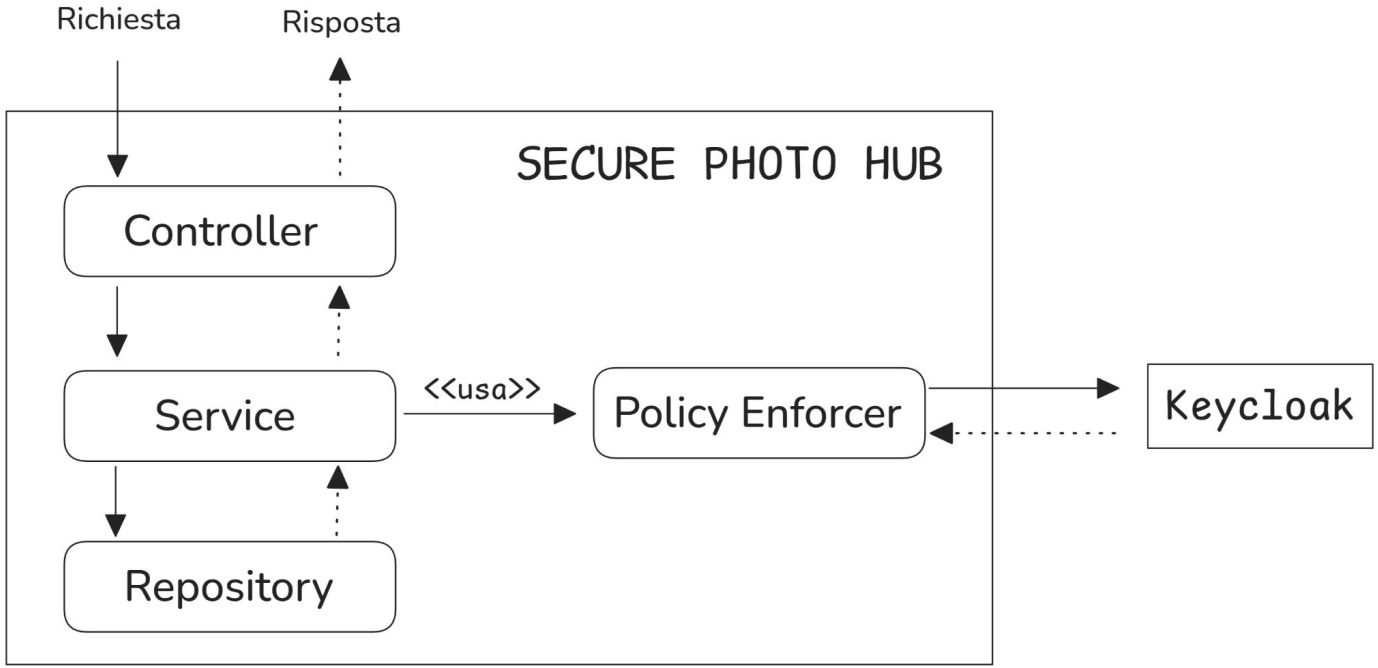
https://www.keycloak.org/docs/latest/authorization_services/index.html#_overview_architecture

AC-3 ACCESS ENFORCEMENT

Control: Enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

Secure Photo Hub invia le informazioni di autorizzazione a Keycloak, che si occupa di prendere le *decisioni relative al controllo degli accessi*.

Successivamente, Secure Photo Hub *forzerà* tali decisioni.



AC-3 **ACCESS ENFORCEMENT**

Control: Enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

Secure Photo Hub è composto da tre componenti che interagiscono con Keycloak per gestire e prendere decisioni relative all'accesso:

- *ImagePolicyEnforcer*
- *PhotoPolicyEnforcer*
- *AlbumPolicyEnforcer*

La strategia adottata prevede la creazione di un Policy Enforcer dedicato per ciascuna tipologia di risorsa.

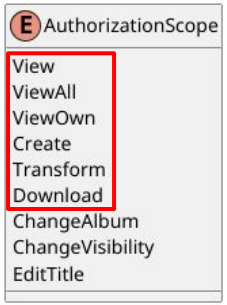
AC-3 ACCESS ENFORCEMENT

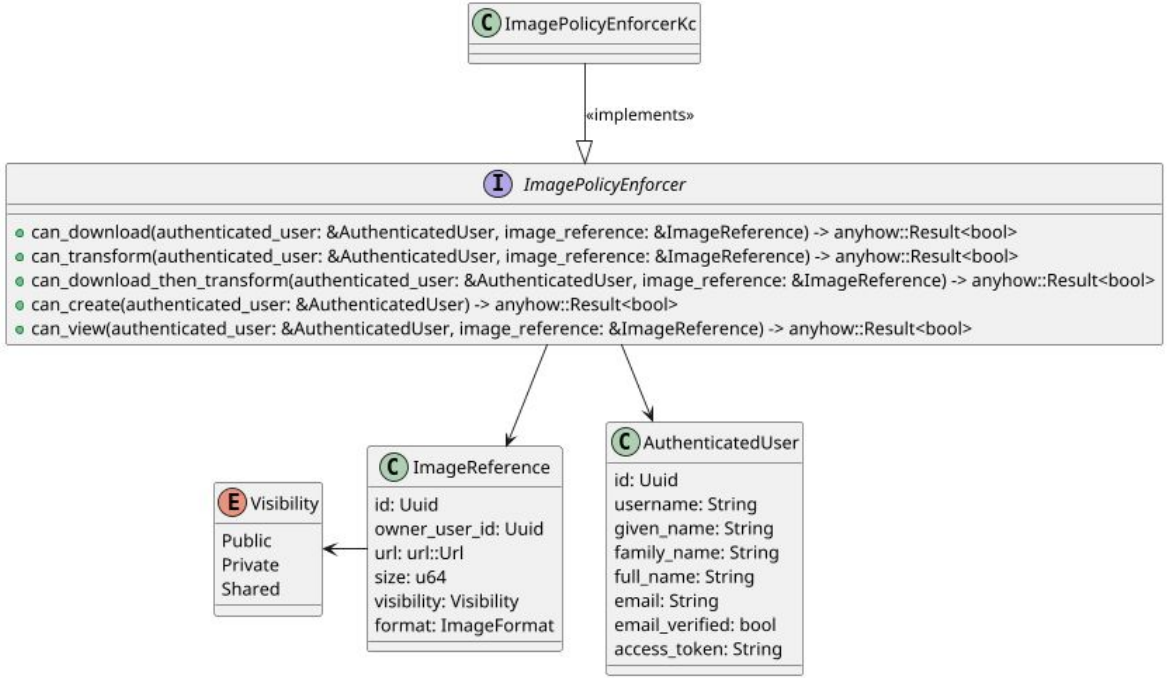
Control: Enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

Implementazione Policy Enforcers:

- *ImagePolicyEnforcer*
- *PhotoPolicyEnforcer*
- *AlbumPolicyEnforcer*

Azioni disponibili sulla risorsa IMAGE





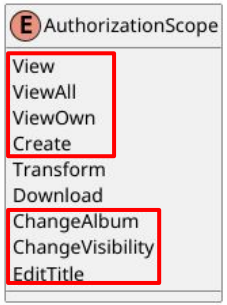
AC-3 ACCESS ENFORCEMENT

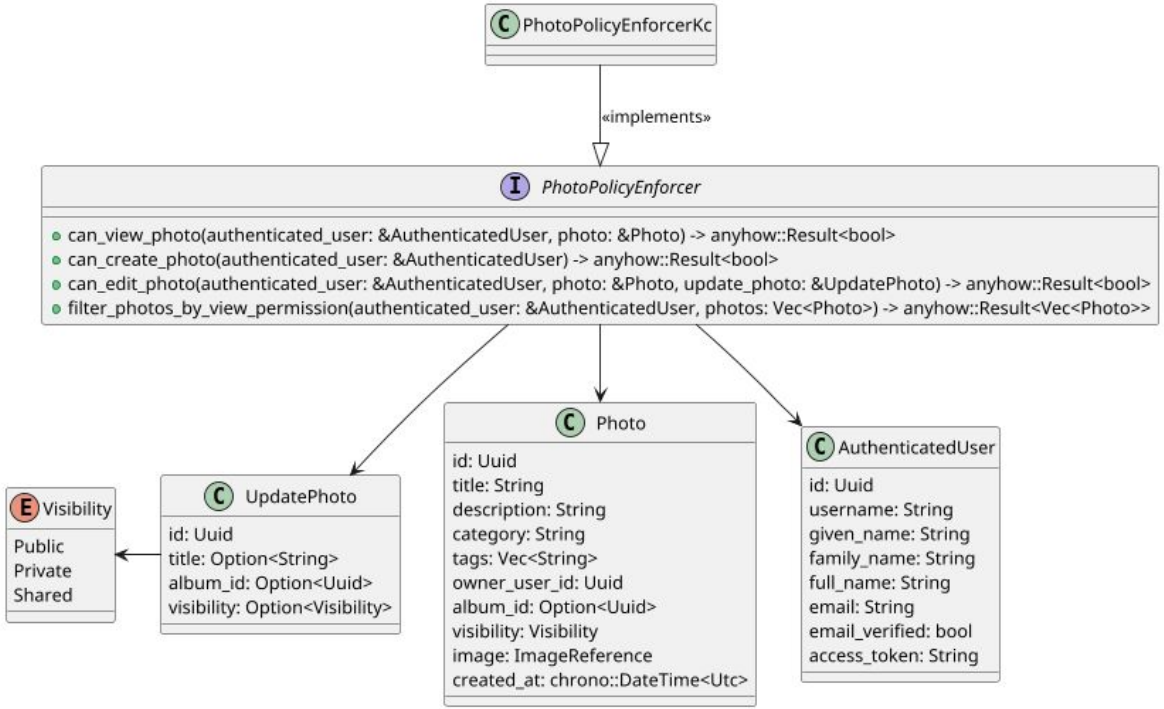
Control: Enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

Implementazione Policy Enforcers:

- *ImagePolicyEnforcer*
- *PhotoPolicyEnforcer*
- *AlbumPolicyEnforcer*

Azioni disponibili sulla risorsa PHOTO





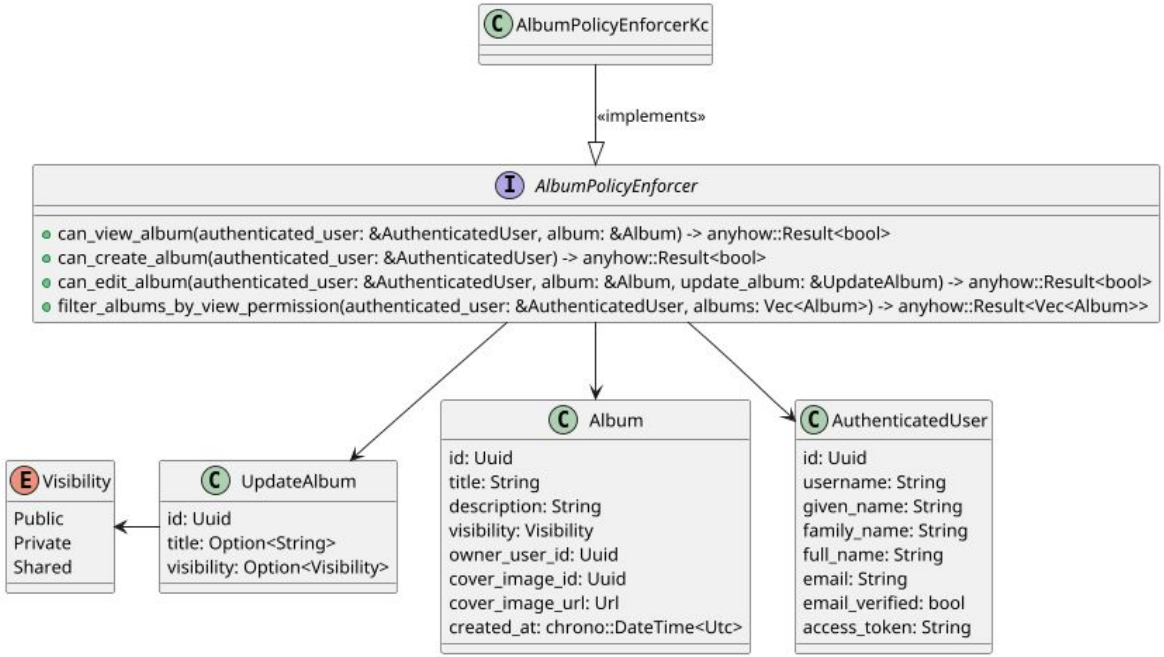
AC-3 ACCESS ENFORCEMENT

Control: Enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

Implementazione Policy Enforcers:

- *ImagePolicyEnforcer*
- *PhotoPolicyEnforcer*
- *AlbumPolicyEnforcer*

Azioni disponibili sulla risorsa ALBUM



AC-3 ACCESS ENFORCEMENT

Control: Enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

(1) ACCESS ENFORCEMENT | RESTRICTED ACCESS TO PRIVILEGED FUNCTIONS

[Withdrawn: Incorporated into AC-6.]

SI

(3) ACCESS ENFORCEMENT | MANDATORY ACCESS CONTROL

NO

Enforce [*Assignment: organization-defined mandatory access control policy*] over the set of covered subjects and objects specified in the policy, and where the policy:

- **(a) Is uniformly enforced across the covered subjects and objects within the system;**
- **(b) Specifies that a subject that has been granted access to information is constrained from doing any of the following;**
  - **(1) Passing the information to unauthorized subjects or objects;**
  - **(2) Granting its privileges to other subjects;**
  - **(3) Changing one or more security attributes (specified by the policy) on subjects, objects, the system, or system components;**
  - **(4) Choosing the security attributes and attribute values (specified by the policy) to be associated with newly created or modified objects; and**
  - **(5) Changing the rules governing access control; and**
- **(c) Specifies that [*Assignment: organization-defined subjects*] may explicitly be granted [*Assignment: organization-defined privileges*] such that they are not limited by any defined subset (or all) of the above constraints.**

(4) ACCESS ENFORCEMENT | DISCRETIONARY ACCESS CONTROL

NO

Enforce [*Assignment: organization-defined discretionary access control policy*] over the set of covered subjects and objects specified in the policy, and where the policy specifies that a subject that has been granted access to information can do one or more of the following:

- **(a) Pass the information to any other subjects or objects;**
- **(b) Grant its privileges to other subjects;**
- **(c) Change security attributes on subjects, objects, the system, or the system's components;**
- **(d) Choose the security attributes to be associated with newly created or revised objects; or**
- **(e) Change the rules governing access control.**

SI

(7) ACCESS ENFORCEMENT | ROLE-BASED ACCESS CONTROL

Enforce a role-based access control policy over defined subjects and objects and control access based upon [*Assignment: organization-defined roles and users authorized to assume such roles*].

Le politiche di sicurezza vengono forzate *anche* utilizzando i ruoli di un utente:

- BasicUser
- Administrator etc..

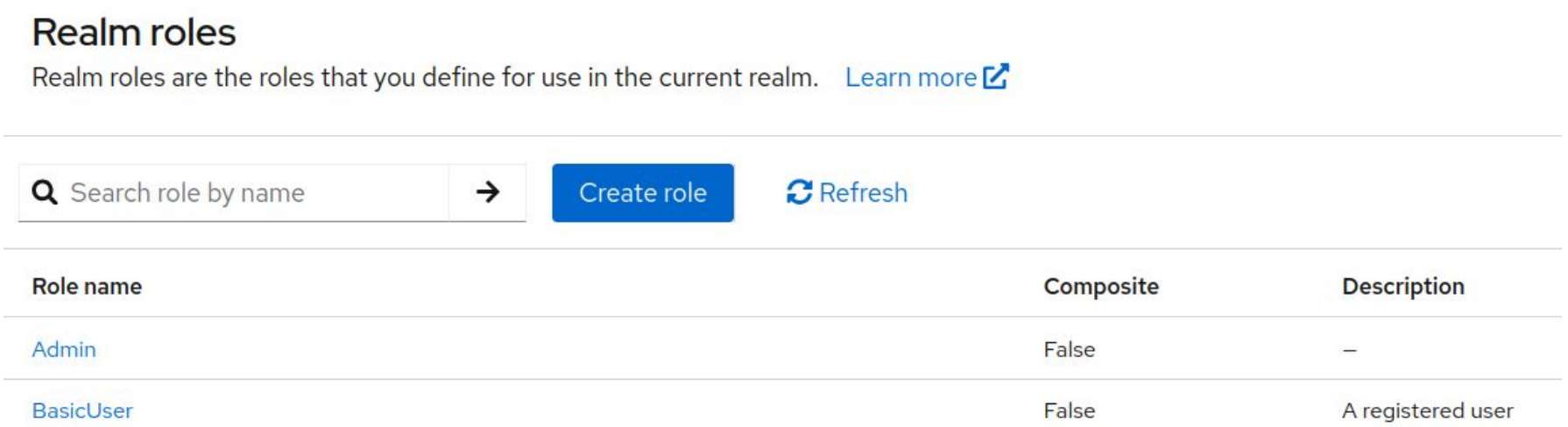
AC-3 ACCESS ENFORCEMENT

SI

Control: Enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

(7) ACCESS ENFORCEMENT | ROLE-BASED ACCESS CONTROL

Enforce a role-based access control policy over defined subjects and objects and control access based upon [*Assignment: organization-defined roles and users authorized to assume such roles*].



AC-3 **ACCESS ENFORCEMENT** **SI**

Control: Enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

(8) ACCESS ENFORCEMENT | REVOCATION OF ACCESS AUTHORIZATIONS

Enforce the revocation of access authorizations resulting from changes to the security attributes of subjects and objects based on [*Assignment: organization-defined rules governing the timing of revocations of access authorizations*].

- **Centralizzazione politiche di sicurezza:** Il sistema centralizza le politiche di sicurezza tramite Keycloak, permettendo modifiche rapide e immediate ai permessi di accesso per singoli utenti o gruppi.

- **Tempistiche di revoca:** Le modifiche vengono applicate istantaneamente, garantendo un controllo preciso sugli accessi alle risorse. Tuttavia, una volta che le immagini vengono scaricate, non è più possibile revocare l'accesso, poiché il controllo sulle risorse è perso al di fuori del sistema.

AC-3 ACCESS ENFORCEMENT

SI

Control: Enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

(11) ACCESS ENFORCEMENT | RESTRICT ACCESS TO SPECIFIC INFORMATION TYPES

Restrict access to data repositories containing [*Assignment: organization-defined information types*].

Discussion: Restricting access to specific information is intended to provide flexibility regarding access control of specific information types within a system. For example, role-based access could be employed to allow access to only a specific type of personally identifiable information within a database rather than allowing access to the database in its entirety. Other examples include restricting access to cryptographic keys, authentication information, and selected system information.

Related Controls: CM-8, CM-12, CM-13, PM-5.

- Con Keycloak, è possibile **implementare controlli di accesso granulari**, consentendo l'accesso solo a specifici tipi di informazioni. Grazie alla flessibilità del sistema, qualora sorgesse la necessità di restrizioni più dettagliate, l'implementazione risulterebbe semplice e rapida, soddisfacendo pienamente il requisito di limitare l'accesso a un sottoinsieme specifico di dati, invece di consentire l'accesso all'intero insieme.

AC-3 **ACCESS ENFORCEMENT** **SI**

Control: Enforce approved authorizations for logical access to information and system resources in accordance with applicable access control policies.

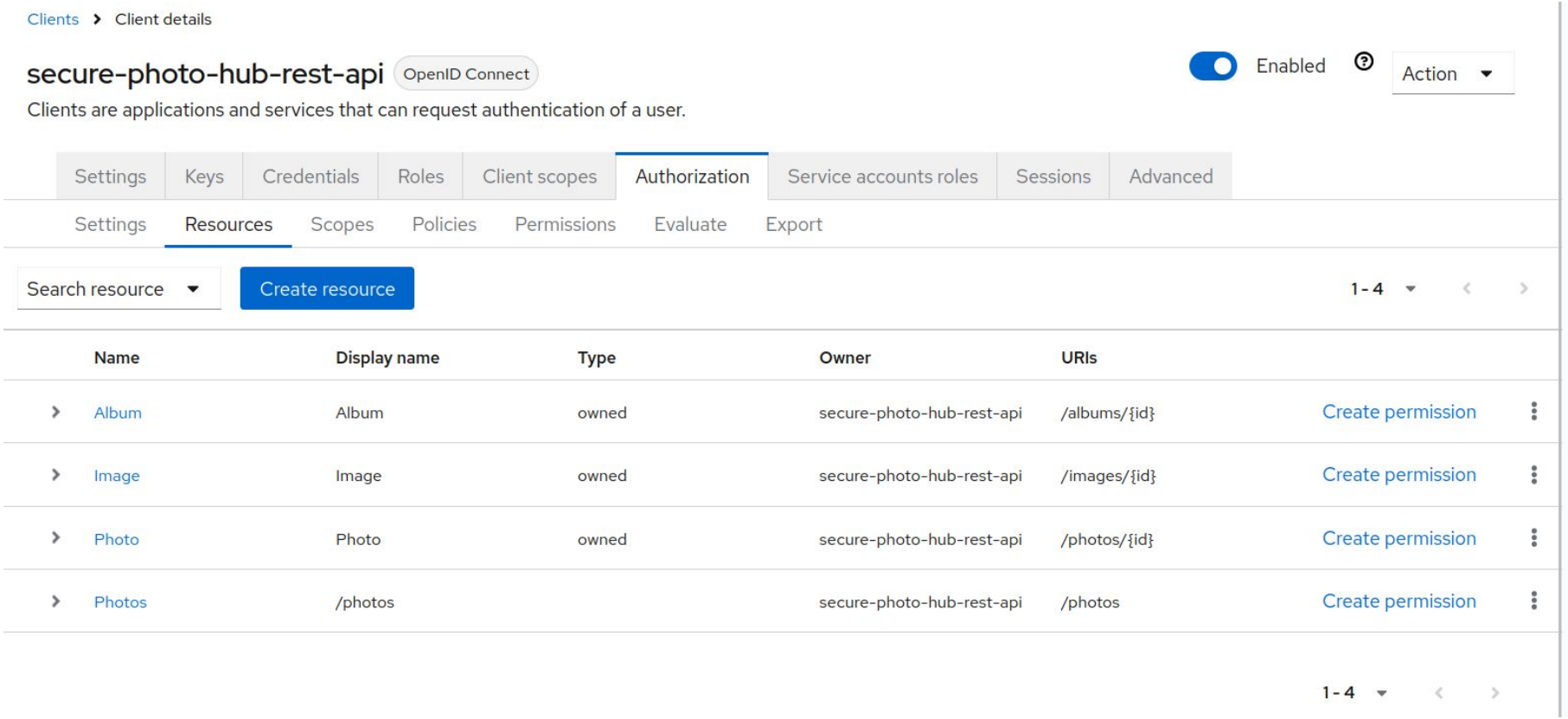
(13) ACCESS ENFORCEMENT | ATTRIBUTE-BASED ACCESS CONTROL

Enforce attribute-based access control policy over defined subjects and objects and control access based upon [*Assignment: organization-defined attributes to assume access permissions*].

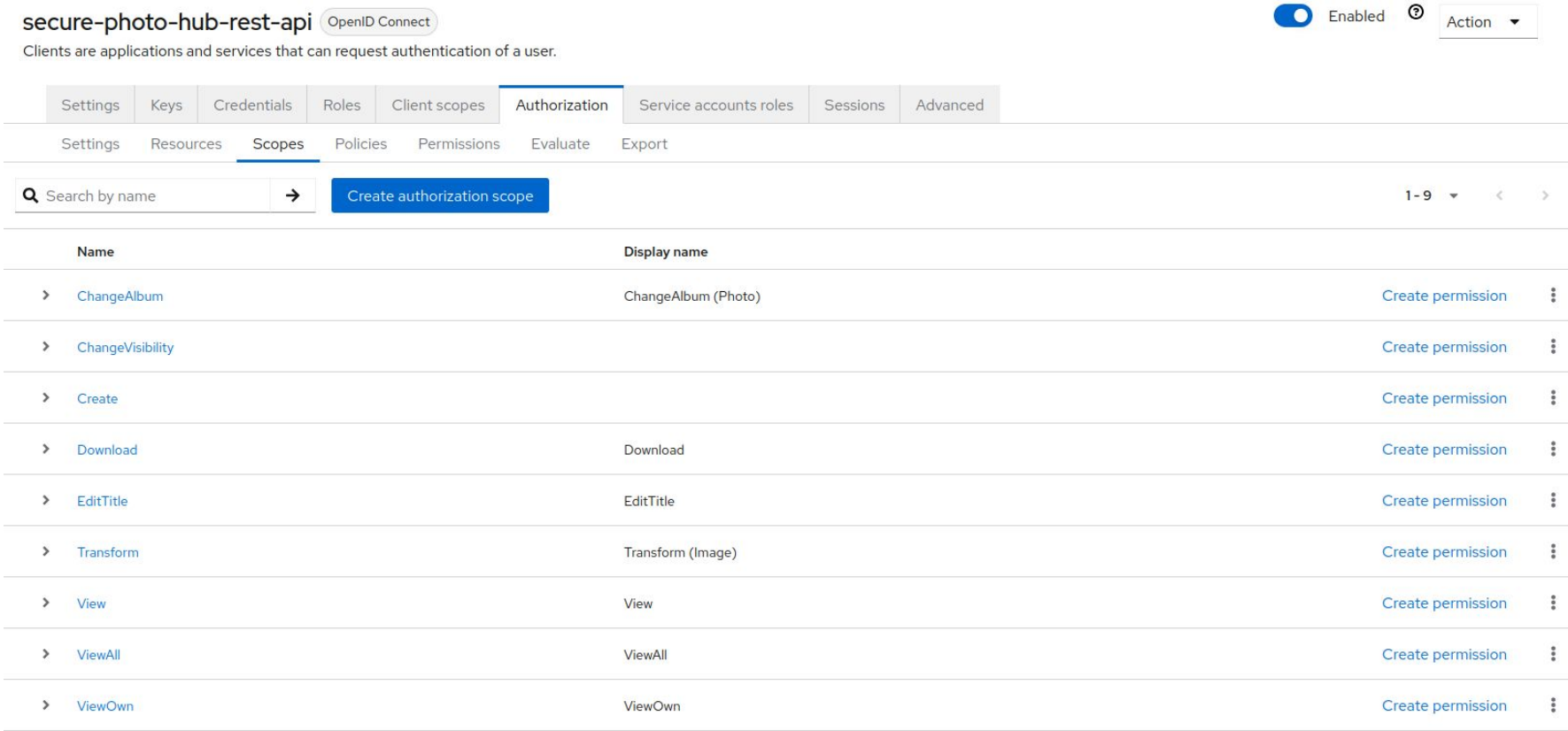
Discussion: Attribute-based access control is an access control policy that restricts system access to authorized users based on specified organizational attributes (e.g., job function, identity), action attributes (e.g., read, write, delete), environmental attributes (e.g., time of day, location), and resource attributes (e.g., classification of a document).

- Il controllo è implementato in quanto su Keycloak sono già presenti politiche di sicurezza che prendono **decisioni di accesso** basate su *action attributes*, *environmental attributes* e *resource attributes*. Di conseguenza, l'access control mechanism del sistema implementa anche una **Attribute-based Access Control Policy (ABAC)**.

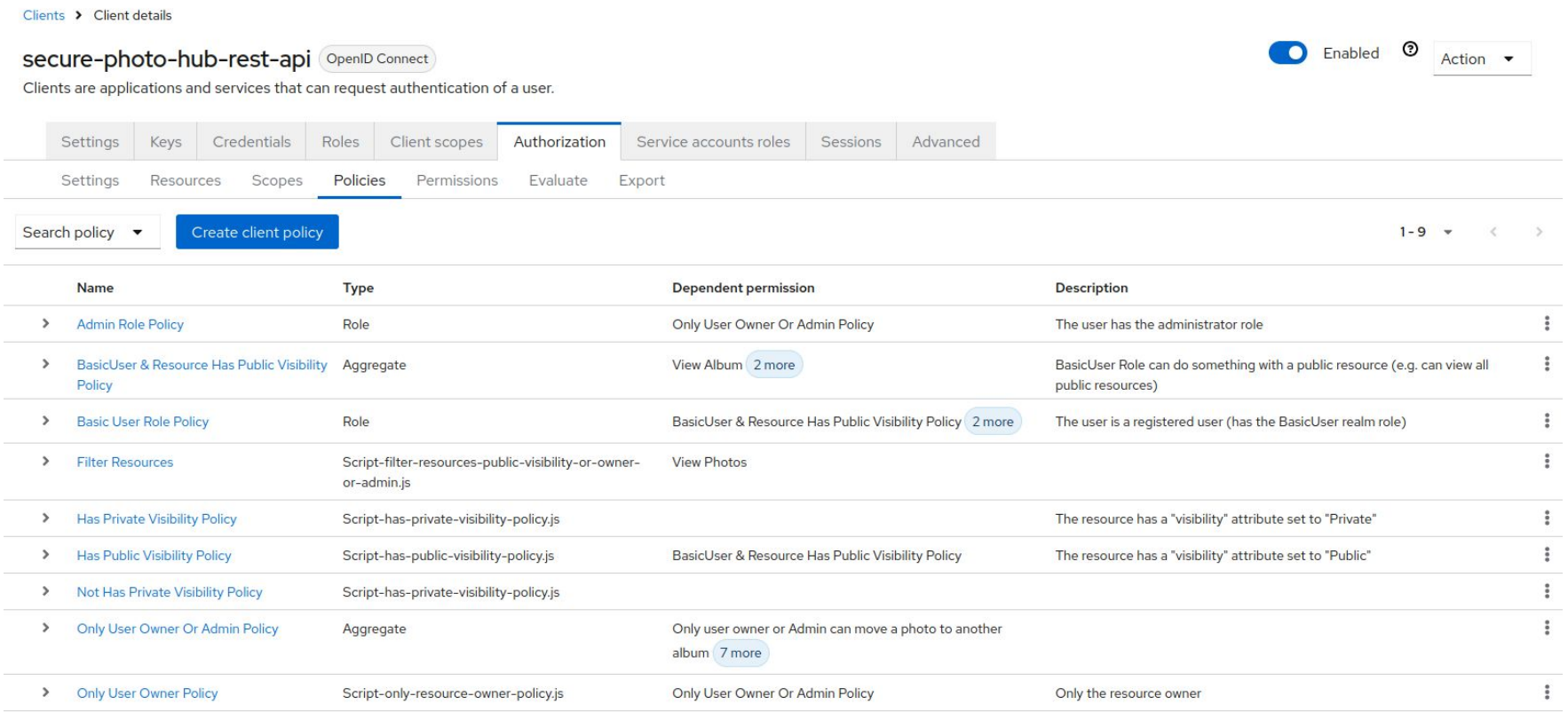
Keycloak Authz Services - Creazione risorse



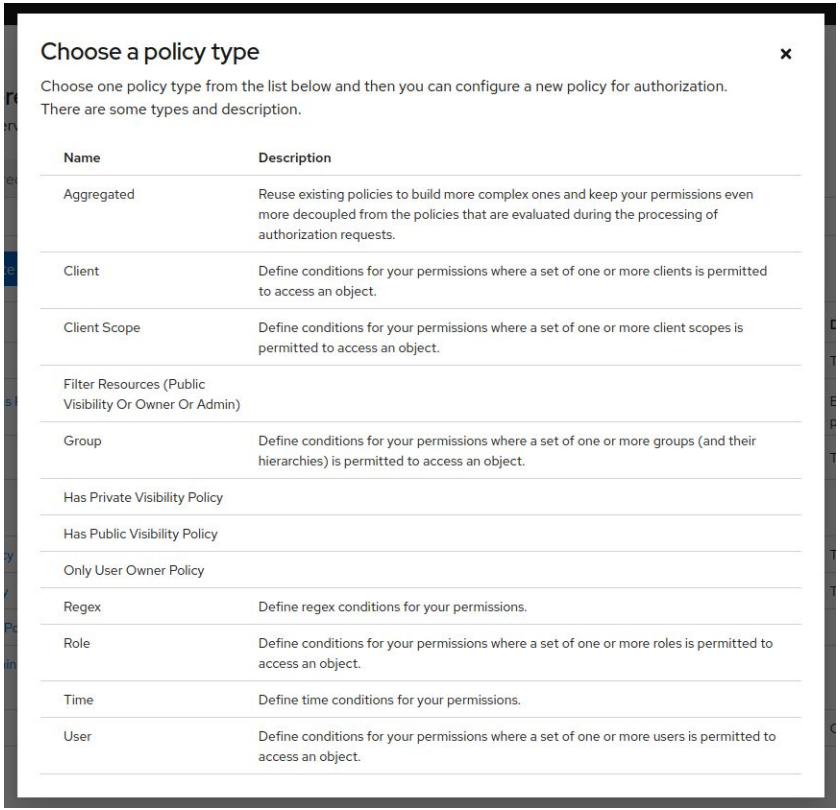
Keycloak Authz Services - Creazione scopes



Keycloak Authz Services - Creazione policies



Keycloak Authz Services - Creazione policies

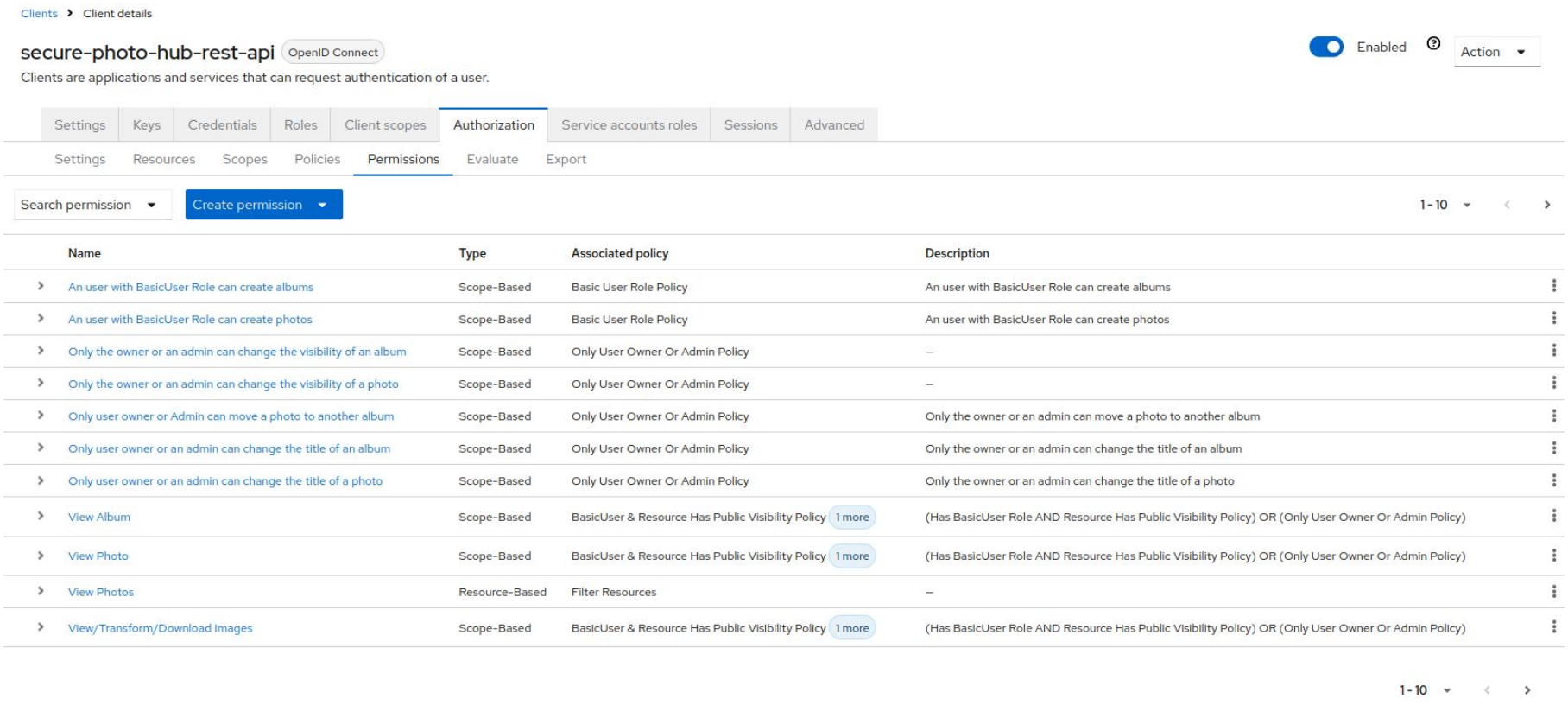


Javascript-based policy:

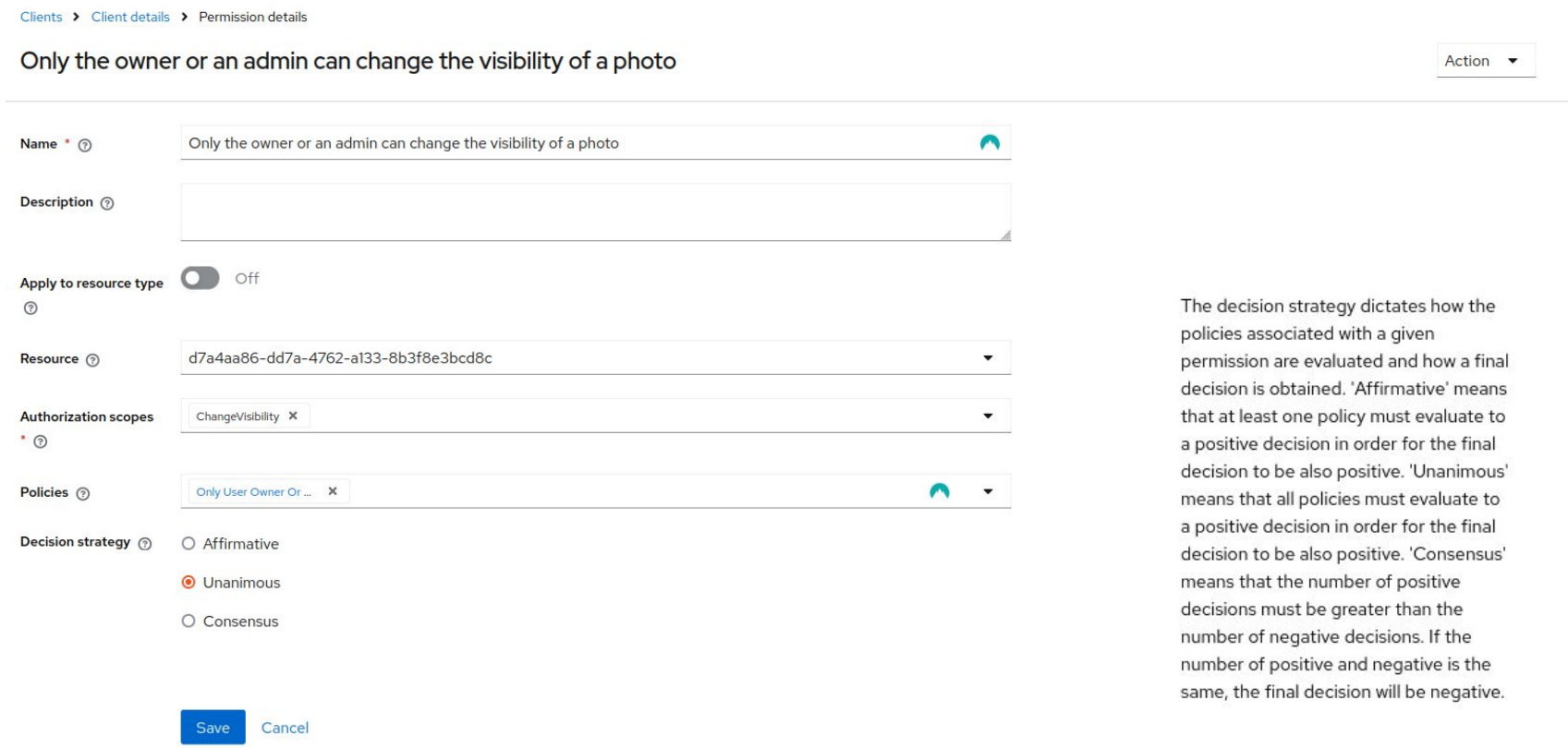
```
var context = $evaluation.getContext();
var identity = context.getIdentity();
var identityAttributes = identity.getAttributes();
var sub = identityAttributes.getValue('sub').asString(0);

var contextAttributes = context.getAttributes();
var resourceOwnerAttr = contextAttributes.getValue('resourceOwner');
var resourceOwner = resourceOwnerAttr ? resourceOwnerAttr.asString(0) : null;

if (sub === resourceOwner) {
       $evaluation.grant();
}
```

“Soltanto il proprietario della risorsa può eseguire [*actions*...]”

Keycloak Authz Services - Creazione permissions



Keycloak Authz Services - Creazione permissions



AC-4 INFORMATION FLOW ENFORCEMENT

Control: Enforce approved authorizations for controlling the flow of information within the system and between connected systems based on [*Assignment: organization-defined information flow control policies*].

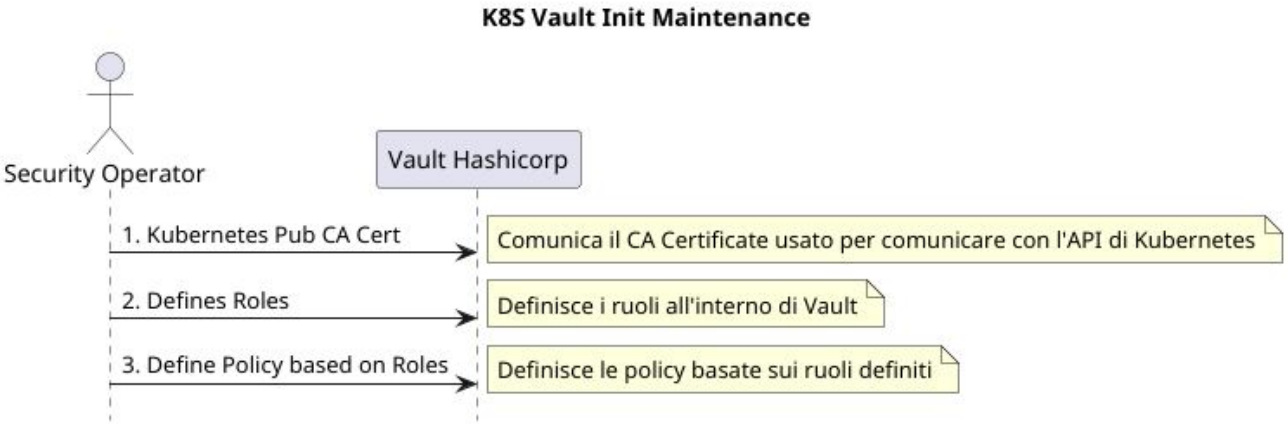
- **Flow Control Policies:**
  - Trasmissione di informazioni tra:
    - **NGINX Ingress e l'esterno:**
      - il traffico che esce dal cluster o entra nel cluster tramite NGINX Ingress **deve** viaggiare su **HTTPS** per garantire la sicurezza delle informazioni
      - le **comunicazioni interne al cluster** (tra i vari servizi) sono consentite in **chiaro**, dato che avvengono all'interno di un ambiente protetto
    - **Vault Agent e Vault:**
      - Vault è situato esternamente al cluster e la comunicazione **deve** avvenire su **HTTPS** per garantire che i dati sensibili siano protetti durante il trasferimento
    - **Cert Manager e Let's Encrypt:**
      - **deve** avvenire su un canale protetto
  - In generale, per tutte le comunicazioni esterne il traffico deve essere protetto.

Gestione dei secrets - Vault

- **Vault** è utilizzato per la gestione sicura dei secrets
- Vault si trova all'**esterno del cluster Kubernetes**
- Le informazioni sensibili come:
  - client-id, client-secret di Keycloak
  - AWS Key, AWS Secret, Bucket Name etc...
  - Database username & password

  devono essere rese disponibili all'applicazione Secure Photo Hub al momento dell'avvio
- Al momento dell'avvio, Vault Agent esegue prima dell'applicazione per comunicare in modo sicuro con Vault, ottenere i secrets e renderli accessibili all'applicazione
- È necessario configurare correttamente Vault e il cluster Kubernetes per garantire una comunicazione sicura tra i due

Gestione dei secrets - Vault



Gestione dei secrets - Vault (creazione secrets e ruolo)

```
vault secrets enable -path=secure-photo-hub kv-v2
vault kv put -mount=secure-photo-hub application-secrets yaml=@application-secrets.yaml
```

Creazione secrets

```
vault policy write secure-photo-hub-kv-ro - <<EOF
 path "secure-photo-hub/data/application-secrets" {
    capabilities = ["read", "list"]
 }
EOF

vault auth enable kubernetes
vault write auth/kubernetes/config \
    token_reviewer_jwt="$SA_JWT_TOKEN" \
    kubernetes_host="$K8S_HOST" \
    kubernetes_ca_cert="$SA_CA_CRT" \
    issuer="https://kubernetes.default.svc.cluster.local"
vault write auth/kubernetes/role/secure-photo-hub-role \
    bound_service_account_names=vault-auth \
    bound_service_account_namespaces=default \
    token_policies=secure-photo-hub-kv-ro \
    ttl=24h
```

Gestione dei secrets - Vault (creazione secrets e ruolo)

```
vault secrets enable -path=secure-photo-hub kv-v2
vault kv put -mount=secure-photo-hub application-secrets yaml=@application-secrets.yaml
```

```
vault policy write secure-photo-hub-kv-ro - <<EOF
 path "secure-photo-hub/data/application-secrets" {
    capabilities = ["read", "list"]
 }
EOF
```

Creazione policy: è possibile soltanto leggere i secrets

```
vault auth enable kubernetes
vault write auth/kubernetes/config \
    token_reviewer_jwt="$SA_JWT_TOKEN" \
    kubernetes_host="$K8S_HOST" \
    kubernetes_ca_cert="$SA_CA_CRT" \
    issuer="https://kubernetes.default.svc.cluster.local"
vault write auth/kubernetes/role/secure-photo-hub-role \
    bound_service_account_names=vault-auth \
    bound_service_account_namespaces=default \
    token_policies=secure-photo-hub-kv-ro \
    ttl=24h
```

Gestione dei secrets - Vault (creazione secrets e ruolo)

```
vault secrets enable -path=secure-photo-hub kv-v2
vault kv put -mount=secure-photo-hub application-secrets yaml=@application-secrets.yaml

vault policy write secure-photo-hub-kv-ro - <<EOF
 path "secure-photo-hub/data/application-secrets" {
    capabilities = ["read", "list"]
 }
EOF
```

```
vault auth enable kubernetes
vault write auth/kubernetes/config \
    token_reviewer_jwt="$SA_JWT_TOKEN" \
    kubernetes_host="$K8S_HOST" \
    kubernetes_ca_cert="$SA_CA_CRT" \
    issuer="https://kubernetes.default.svc.cluster.local"
vault write auth/kubernetes/role/secure-photo-hub-role \
    bound_service_account_names=vault-auth \
    bound_service_account_namespaces=default \
    token_policies=secure-photo-hub-kv-ro \
    ttl=24h
```

Configurazione comunicazione Vault e cluster Kubernetes

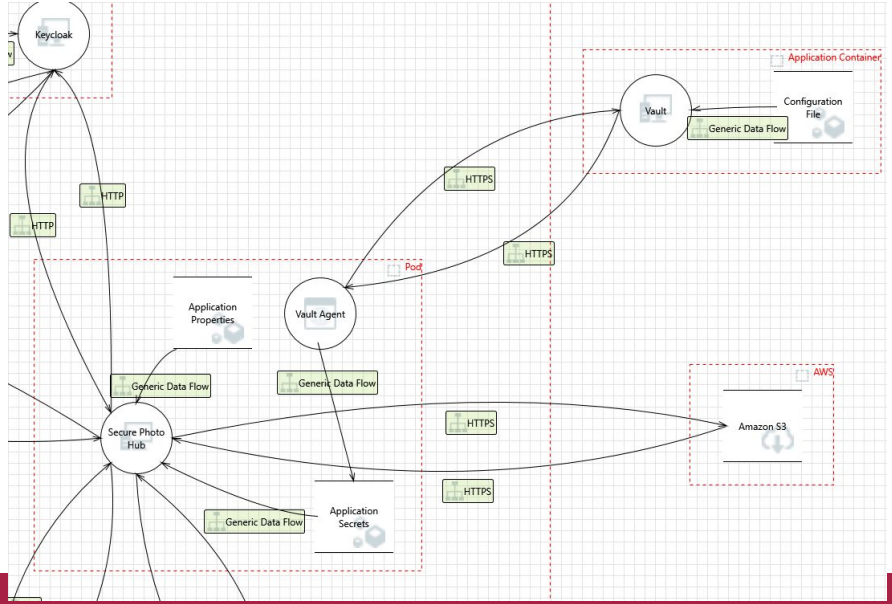
Gestione dei secrets - Vault (creazione secrets e ruolo)

```
vault secrets enable -path=secure-photo-hub kv-v2
vault kv put -mount=secure-photo-hub application-secrets yaml=@application-secrets.yaml

vault policy write secure-photo-hub-kv-ro - <<EOF
 path "secure-photo-hub/data/application-secrets" {
    capabilities = ["read", "list"]
 }
EOF

vault auth enable kubernetes
vault write auth/kubernetes/config \
    token_reviewer_jwt="$SA_JWT_TOKEN" \
    kubernetes_host="$K8S_HOST" \
    kubernetes_ca_cert="$SA_CA_CRT" \
    issuer="https://kubernetes.default.svc.cluster.local"
vault write auth/kubernetes/role/secure-photo-hub-role \
    bound_service_account_names=vault-auth \
    bound_service_account_namespaces=default \
    token_policies=secure-photo-hub-kv-ro \
    ttl=24h
```

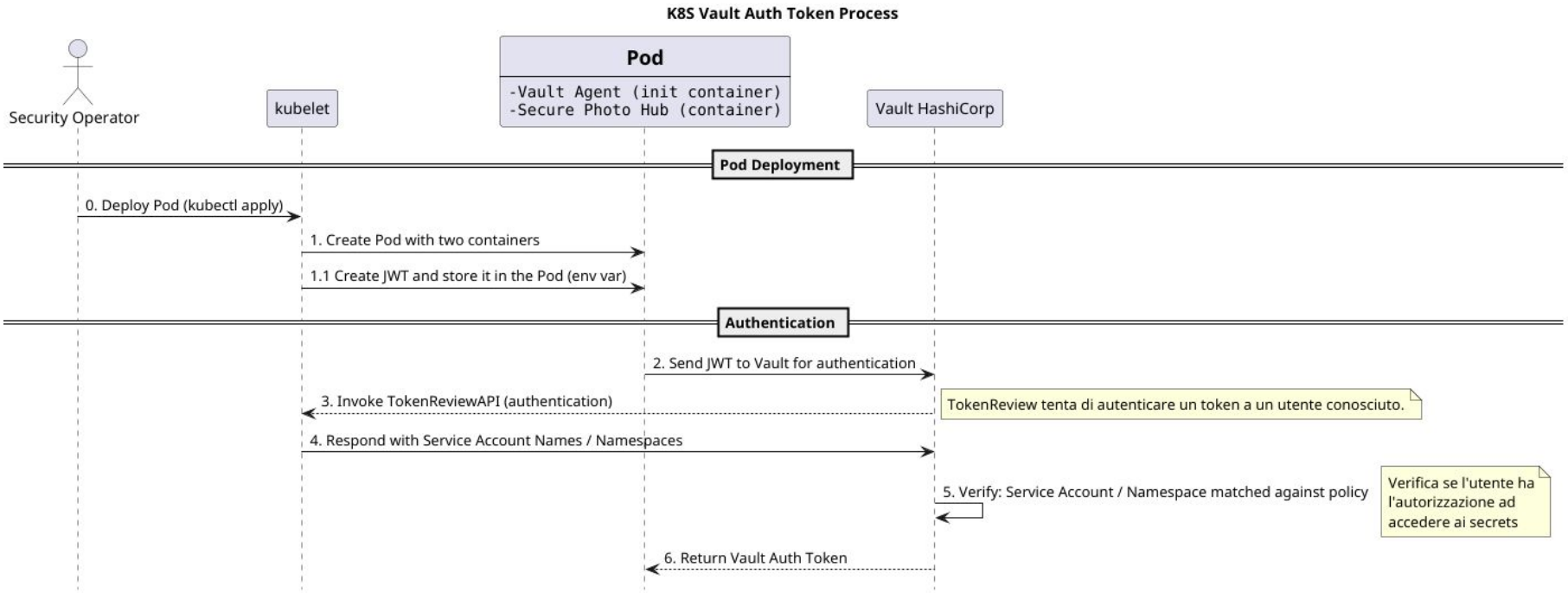
Creazione ruolo l'autenticazione K8S (secure-photo-hub-role) e successivo bind della policy precedentemente creata

Gestione dei secrets - Vault (creazione secrets e ruolo)

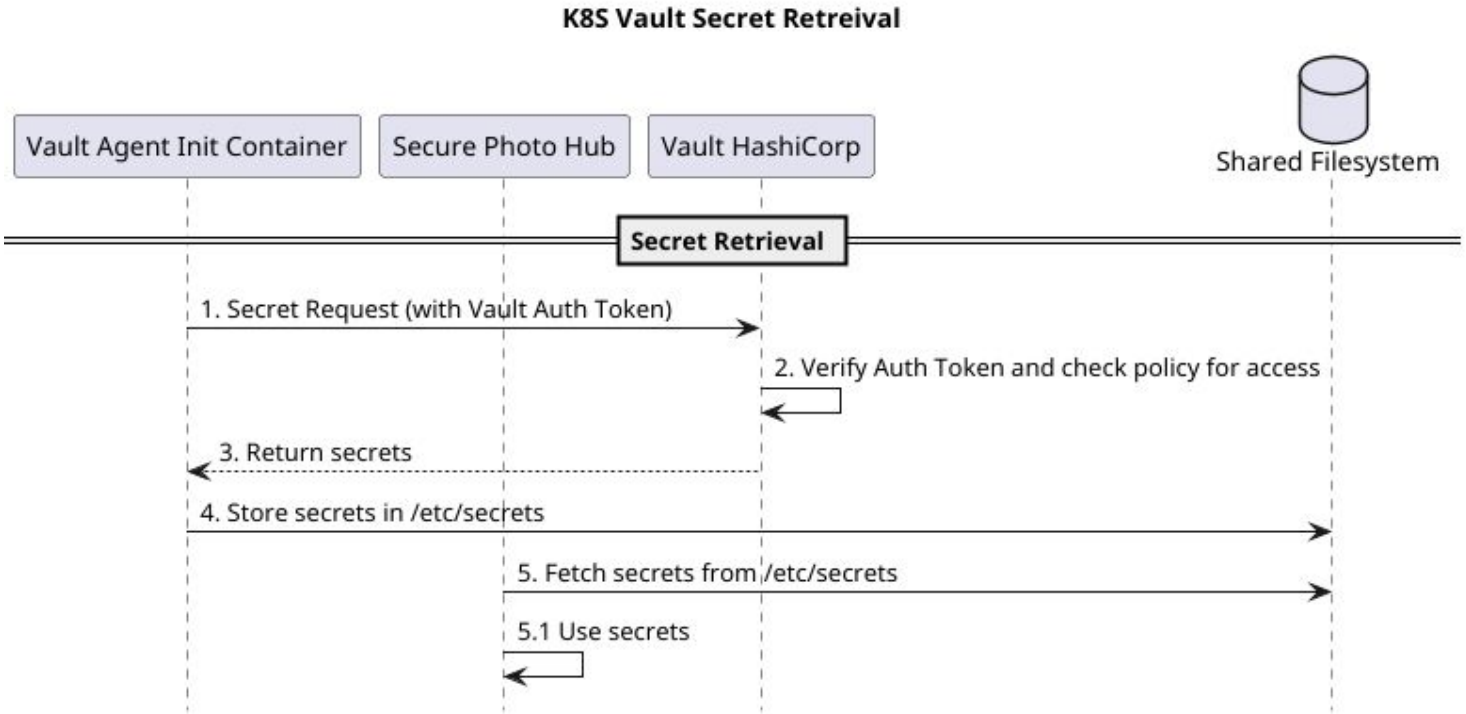
- Al momento dell'avvio del Pod contenente l'applicazione Secure Photo Hub:
  - viene eseguito prima un *init container* **Vault Agent**
  - che comunica in maniera sicura con Vault
  - per prelevare i secrets
  - e renderli disponibili a Secure Photo Hub



Gestione dei secrets - Vault



Gestione dei secrets - Vault



AC-5 SEPARATION OF DUTIES

- Divisione dei compiti e delle responsabilità per evitare conflitti di interessi e l'abuso di privilegi.

- Il principio della *separazione dei compiti* può essere forzato nel sistema tramite Keycloak attraverso la gestione dei ruoli, gruppi, permessi e politiche

- Sono stati creati ruoli, politiche e permessi in maniera tale da evitare di assegnare troppo potere a un singolo utente.

- Principalmente questo viene raggiunto attraverso *fine-grained authorization*.

AC-6 **LEAST PRIVILEGE**

Control: Employ the principle of least privilege, allowing only authorized accesses for users (or processes acting on behalf of users) that are necessary to accomplish assigned organizational tasks.

- Il principio del *least privilege* viene implementato con gli stessi meccanismi di sicurezza utilizzati per implementare il principio di separation of duties.

- Il principio del privilegio minimo viene applicato anche ai processi di sistema, garantendo che i processi operino a **livelli di privilegio non superiori a quelli necessari** per eseguire le funzioni per le quali sono stati progettati.

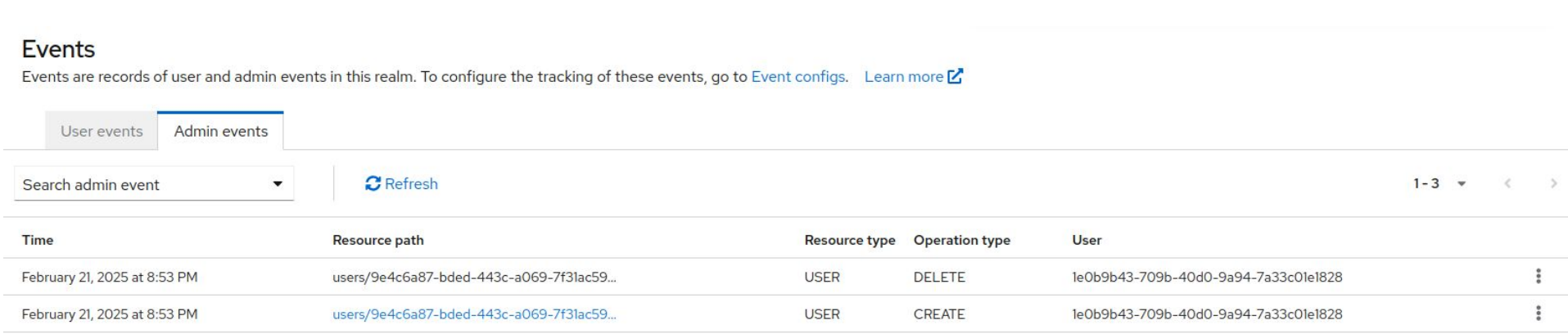
AC-6 LEAST PRIVILEGE

Control: Employ the principle of least privilege, allowing only authorized accesses for users (or processes acting on behalf of users) that are necessary to accomplish assigned organizational tasks.

Control Enhancements:

(9) LEAST PRIVILEGE | LOG USE OF PRIVILEGED FUNCTIONS

Log the execution of privileged functions.



AC-6 LEAST PRIVILEGE

Control: Employ the principle of least privilege, allowing only authorized accesses for users (or processes acting on behalf of users) that are necessary to accomplish assigned organizational tasks.

Control Enhancements:

(10) LEAST PRIVILEGE | PROHIBIT NON-PRIVILEGED USERS FROM EXECUTING PRIVILEGED FUNCTIONS

Prevent non-privileged users from executing privileged functions.

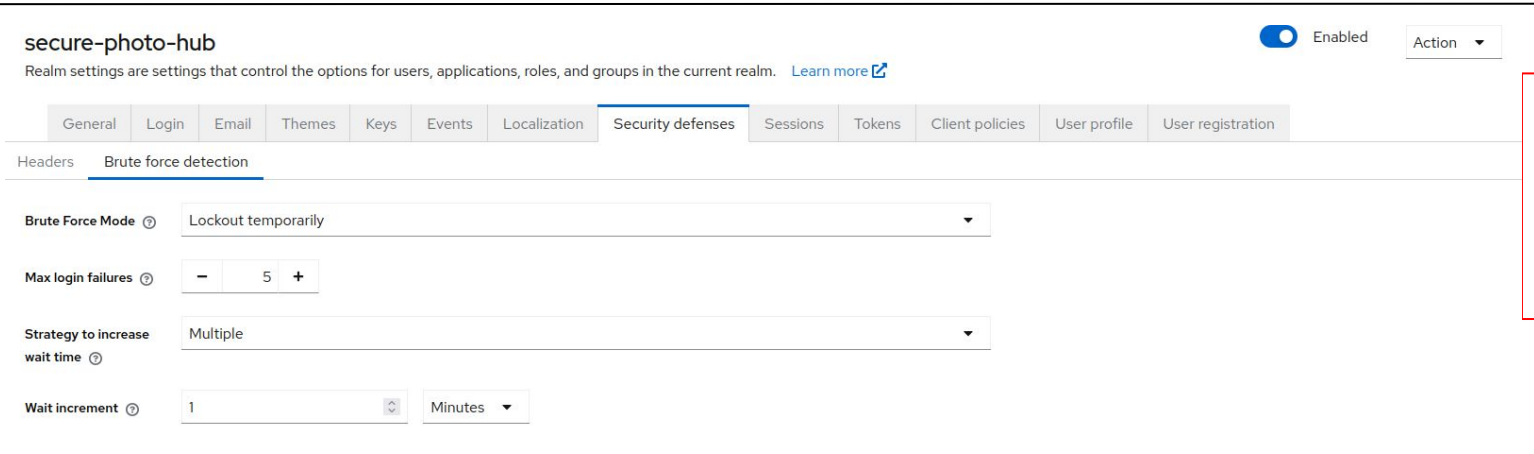
- Il sistema implementa il Control Enhancement (10) in quanto un utente non-privilegiato non ha nessun modo per eseguire funzioni privilegiate.

AC-7 UNSUCCESSFUL LOGON ATTEMPTS

Control:

a. Enforce a limit of [*Assignment: organization-defined number*] consecutive invalid logon attempts by a user during a [*Assignment: organization-defined time period*]; and

b. Automatically [*Selection (one or more): lock the account or node for an* [*Assignment: organization-defined time period*]; *lock the account or node until released by an administrator; delay next logon prompt per* [*Assignment: organization-defined delay algorithm*]; *notify system administrator; take other* [*Assignment: organization-defined action*]] when the maximum number of unsuccessful attempts is exceeded.



Dopo 5 login failures, l'account viene bloccato momentaneamente.

AC-14 PERMITTED ACTIONS WITHOUT IDENTIFICATION OR AUTHENTICATION

Control:

a. Identify [*Assignment: organization-defined user actions*] that can be performed on the system without identification or authentication consistent with organizational mission and business functions; and

b. Document and provide supporting rationale in the security plan for the system, user actions not requiring identification or authentication.

- Chiunque ha il permesso di visualizzare foto o album PUBBLICI

AC-24 ACCESS CONTROL DECISIONS

Control: [*Selection: Establish procedures; Implement mechanisms*] to ensure [*Assignment: organization-defined access control decisions*] are applied to each access request prior to access enforcement.

Discussion: Access control decisions (also known as authorization decisions) occur when authorization information is applied to specific accesses. In contrast, access enforcement occurs when systems enforce access control decisions. While it is common to have access control decisions and access enforcement implemented by the same entity, it is not required, and it is not always an optimal implementation choice. For some architectures and distributed systems, different entities may make access control decisions and enforce access.

L'applicazione Secure Photo Hub ha all'interno dei componenti che si occupano esclusivamente delle ***Access Control Decisions*** (tramite i *Policy Enforcers* e Keycloak). L'applicazione, quindi, separa nettamente le decisioni di accesso dall'***Enforcement***. Precisamente, chi utilizza i Policy Enforcers sarà responsabile di prendere la decisione finale su come agire, ossia sarà lui a *forzare* l'access control decision.

AC-24 ACCESS CONTROL DECISIONS

(1) ACCESS CONTROL DECISIONS | TRANSMIT ACCESS AUTHORIZATION INFORMATION

Transmit [*Assignment: organization-defined access authorization information*] using [*Assignment: organization-defined controls*] to [*Assignment: organization-defined systems*] that enforce access control decisions.

L’applicazione invia le informazioni di autorizzazione a Keycloak, che funge da punto centrale per le Access Control Decisions. Le informazioni trasmesse comprendono anche attributi relativi a una o più risorse, essenziali per valutare correttamente le decisioni di accesso.

AU-2 EVENT LOGGING

Events

Events are records of user and admin events in this realm. To configure the tracking of these events, go to Event configs. Learn more

User events | Admin events

Search admin event | Refresh | 1 - 3

Time	Resource path	Resource type	Operation type	User
February 21, 2025 at 8:53 PM	users/9e4c6a87-bded-443c-a069-7f31ac59...	USER	DELETE	1e0b9b43-709b-40d0-9a94-7a33c01e1828
February 21, 2025 at 8:53 PM	users/9e4c6a87-bded-443c-a069-7f31ac59...	USER	CREATE	1e0b9b43-709b-40d0-9a94-7a33c01e1828

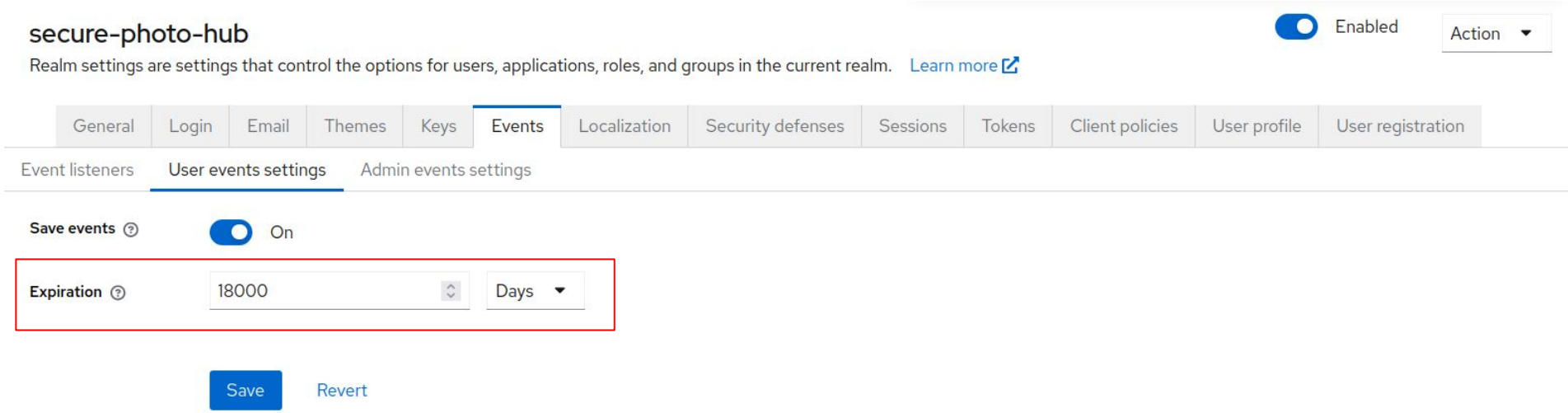
User events | Admin events

Search user event | Refresh | 1 - 10

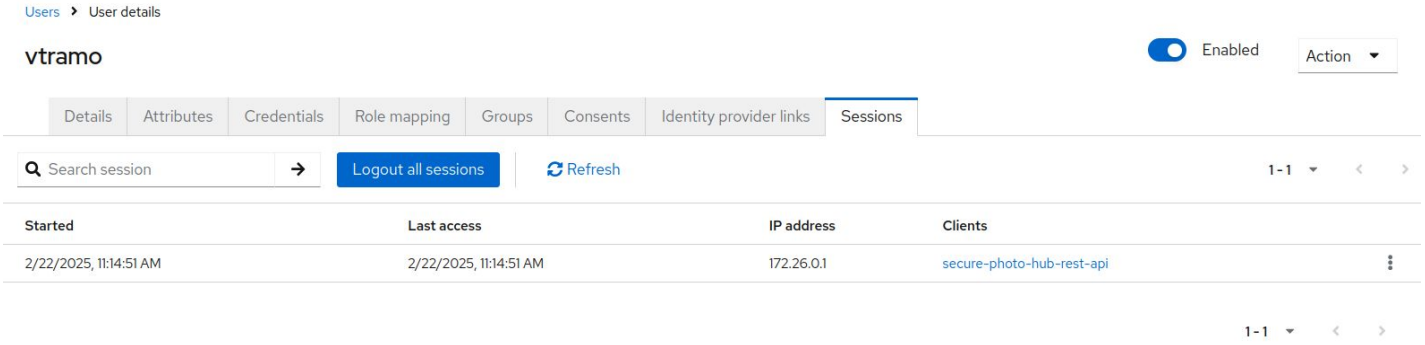
Time	User	Event saved type	IP address	Client
February 22, 2025 at 10:07 AM	08bb3d22-fdce-49f8-9d51-2e71bdb33b8e	CLIENT_LOGIN	172.26.0.1	secure-photo-hub-rest-api
February 22, 2025 at 10:02 AM	08bb3d22-fdce-49f8-9d51-2e71bdb33b8e	CLIENT_LOGIN	172.26.0.1	secure-photo-hub-rest-api
February 22, 2025 at 9:56 AM	08bb3d22-fdce-49f8-9d51-2e71bdb33b8e	CLIENT_LOGIN	172.26.0.1	secure-photo-hub-rest-api
February 22, 2025 at 9:56 AM	08bb3d22-fdce-49f8-9d51-2e71bdb33b8e	CLIENT_LOGIN	172.26.0.1	secure-photo-hub-rest-api
February 22, 2025 at 9:55 AM	08bb3d22-fdce-49f8-9d51-2e71bdb33b8e	CLIENT_LOGIN	172.26.0.1	secure-photo-hub-rest-api
February 22, 2025 at 9:51 AM	a5607f8e-fbff-4e8f-98b6-0f194a8b594c	CODE_TO_TOKEN	172.26.0.1	secure-photo-hub-rest-api
February 22, 2025 at 9:51 AM	a5607f8e-fbff-4e8f-98b6-0f194a8b594c	LOGIN	172.26.0.1	secure-photo-hub-rest-api
February 22, 2025 at 9:50 AM	a5607f8e-fbff-4e8f-98b6-0f194a8b594c	CODE_TO_TOKEN	172.26.0.1	secure-photo-hub-rest-api
February 22, 2025 at 9:50 AM	a5607f8e-fbff-4e8f-98b6-0f194a8b594c	LOGIN	172.26.0.1	secure-photo-hub-rest-api
February 22, 2025 at 9:48 AM	08bb3d22-fdce-49f8-9d51-2e71bdb33b8e	CLIENT_LOGIN	172.26.0.1	secure-photo-hub-rest-api

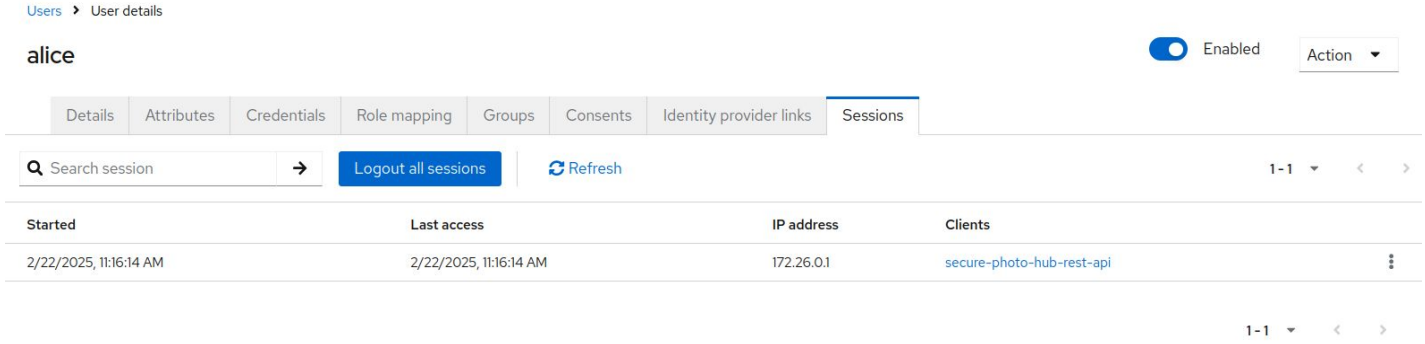
AU-11 AUDIT RECORD RETENTION

Control: Retain audit records for [*Assignment: organization-defined time period consistent with records retention policy*] to provide support for after-the-fact investigations of incidents and to meet regulatory and organizational information retention requirements.



AU-14 SESSION AUDIT





SC-8 TRANSMISSION CONFIDENTIALITY AND INTEGRITY

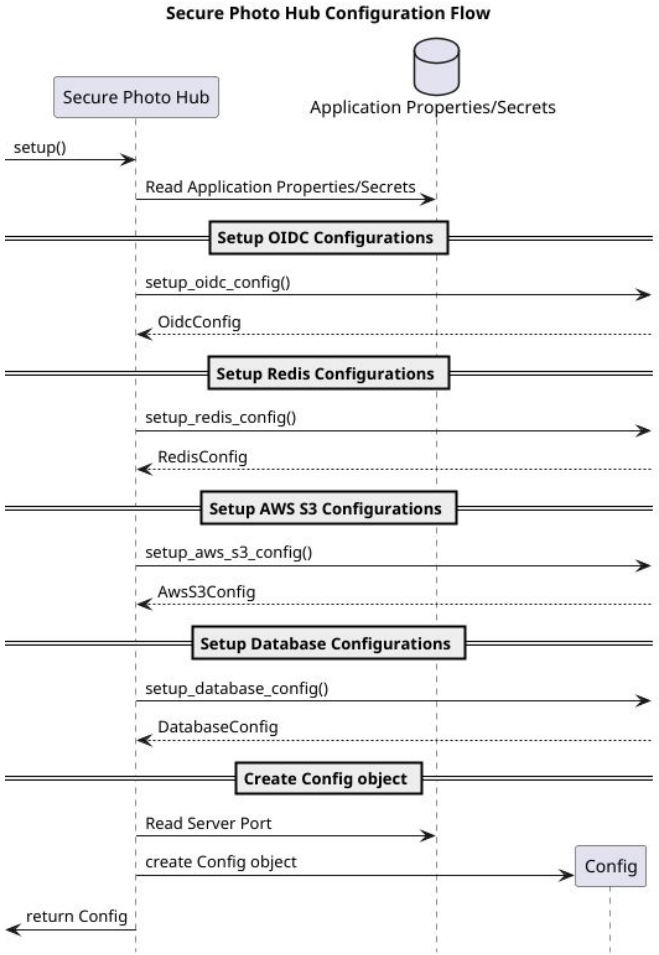
Control: Protect the [*Selection (one or more): confidentiality; integrity*] of transmitted information.

Tutte le informazioni trasmesse, sia in uscita che in entrata dal cluster, vengono protette tramite HTTPS, assicurando riservatezza e integrità durante il trasferimento

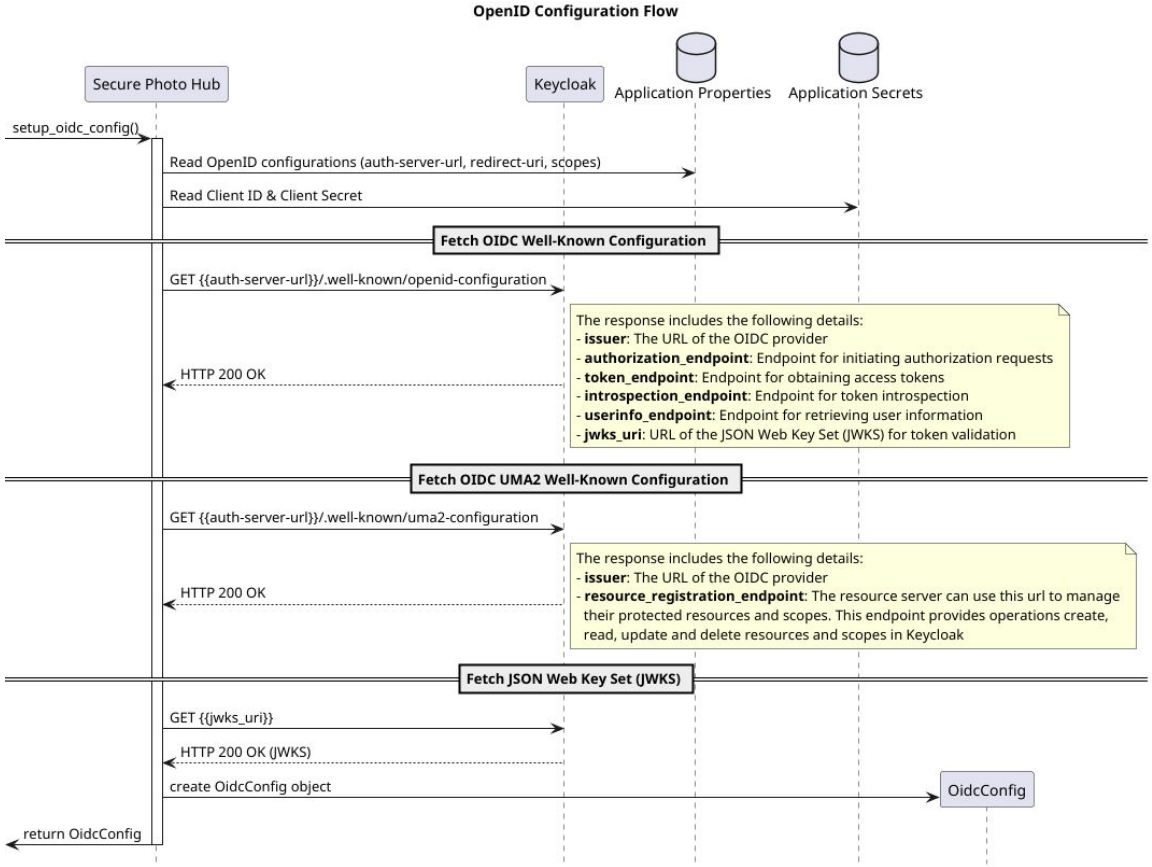
System Configuration

Il servizio Secure Photo Hub all’avvio:

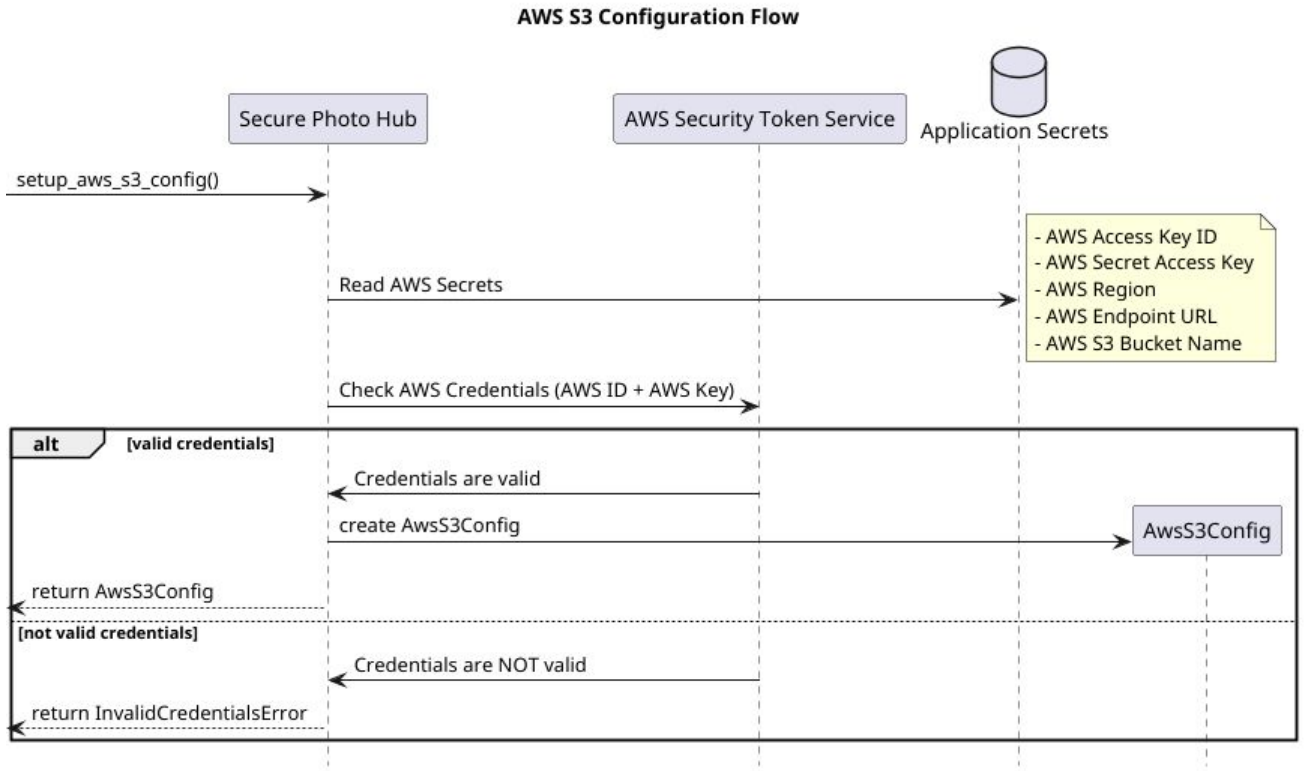
1. Legge application properties & secrets
2. Configura OAuth2 e OIDC
3. Configura Redis
4. Configura AWS S3
5. Configura la comunicazione con il database
6. Crea un oggetto *Config* con tutte le configurazioni del sistema



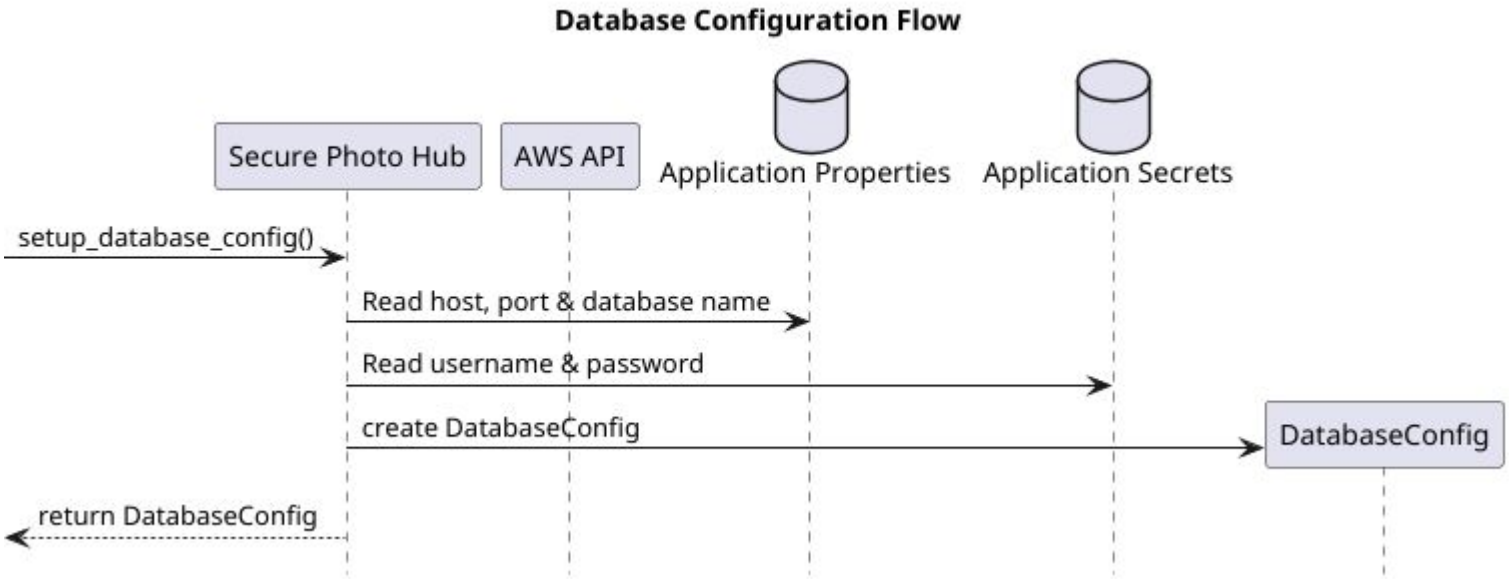
OIDC Configuration



AWS S3 Configuration



Database Configuration



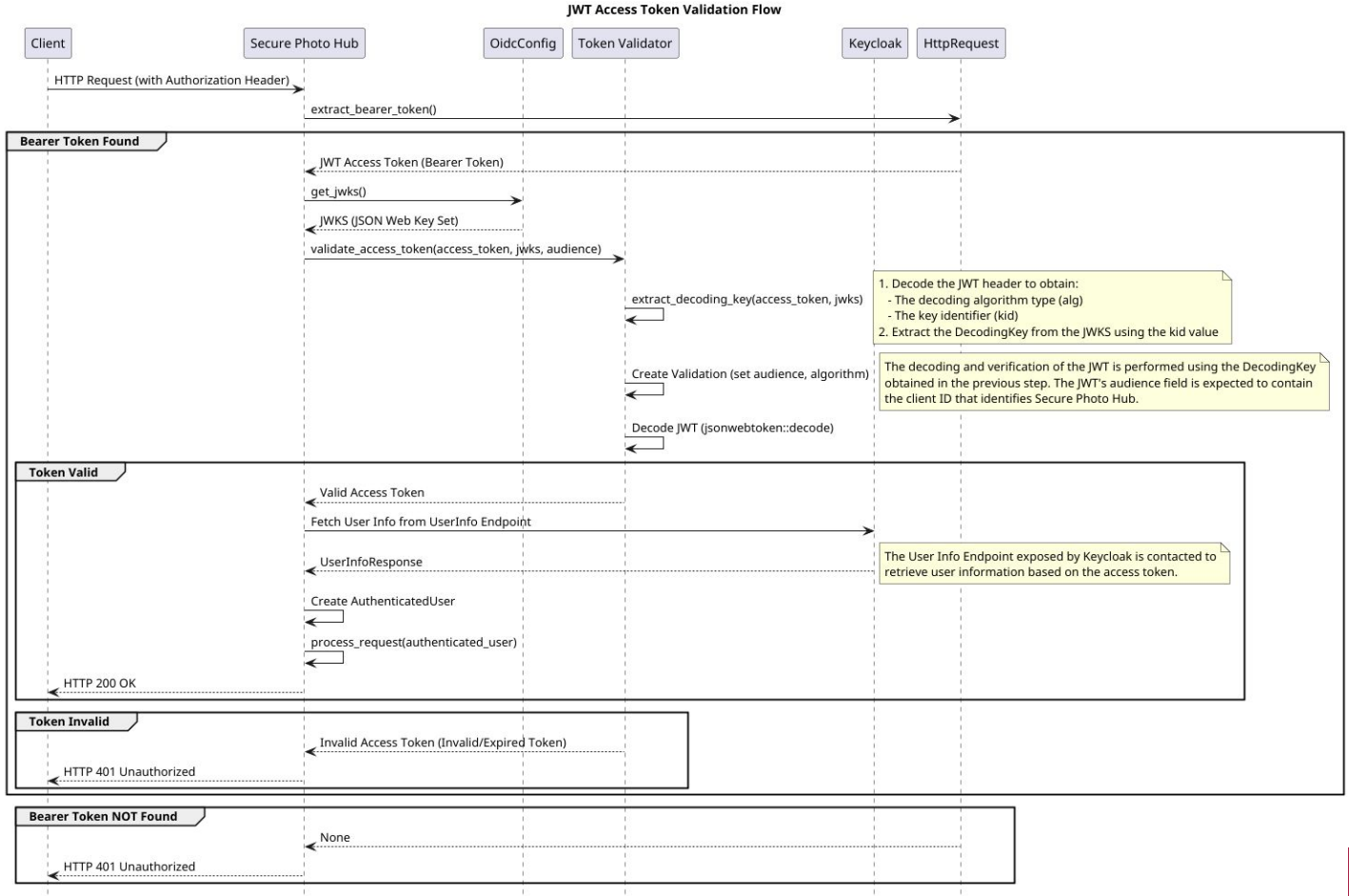
OIDC - Chi può interagire con Secure Photo Hub?

Secure Photo Hub può essere utilizzato sia da un browser che da un servizio web generico (essendo una REST API). In base al tipo di client, viene applicato un meccanismo di autenticazione differente:

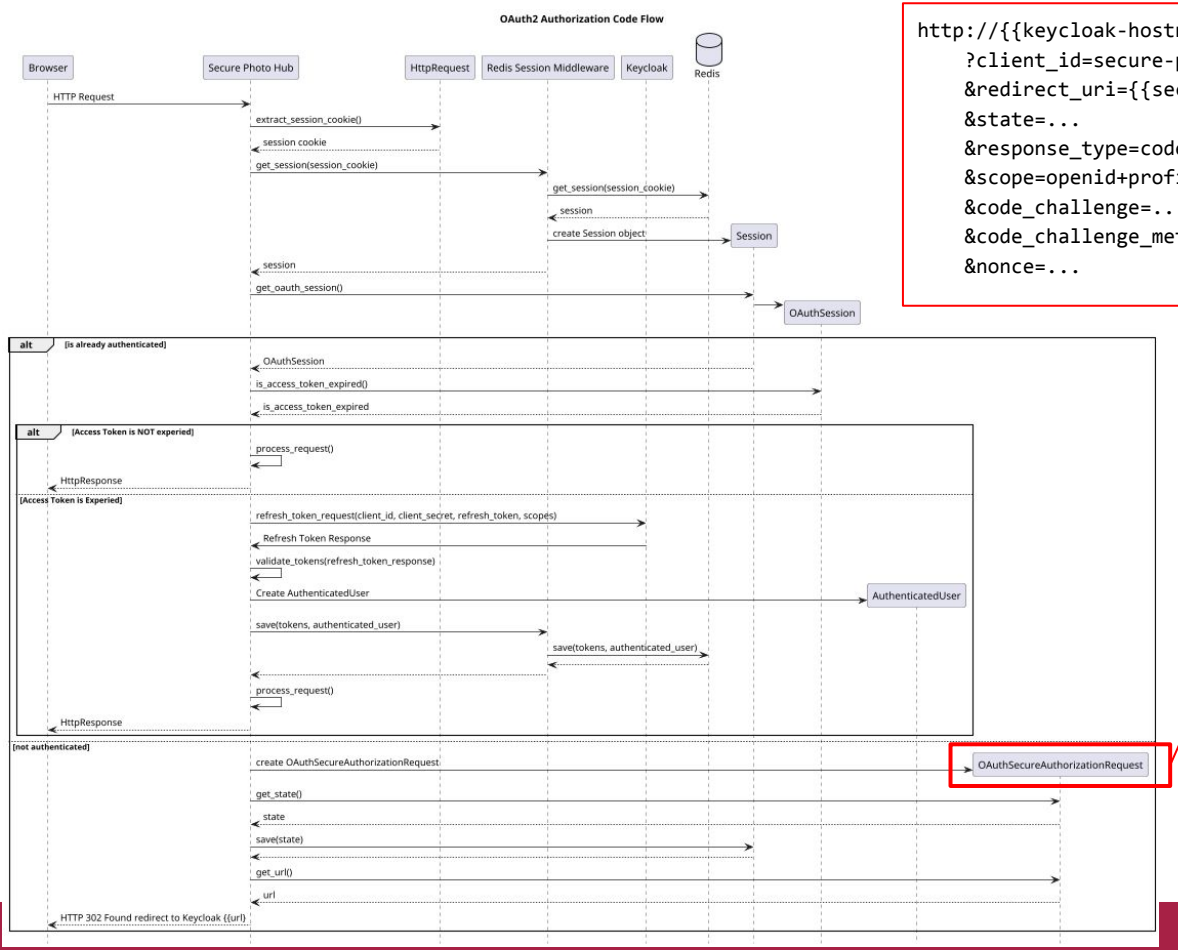
- **Servizio Web:** ad esempio una applicazione Java. Se il client è un servizio web, quest'ultimo deve includere un Bearer Token valido, generato da Keycloak, nell'header Authorization della richiesta HTTP.
- **Browser:** se l'interazione avviene tramite un browser e l'utente non è autenticato, verrà automaticamente reindirizzato su Keycloak (Authorization Server) per completare il processo di autenticazione.

Ogni richiesta HTTP verso Secure Photo Hub viene elaborata da due middleware di autenticazione, ciascuno progettato per gestire i due casi descritti in precedenza. Se entrambi i middleware falliscono, la richiesta viene respinta con **HTTP 401 Unauthorized**.

OIDC - Access Token Validation Flow (Servizio Web)



OIDC - Authorization Code Flow (Browser) 1/2




```
http://{{keycloak-hostname}}/realms/secure-photo-hub/protocol/openid-connect/auth
    ?client_id=secure-photo-hub-rest-api
    &redirect_uri={{secure_photo_hub_endpoint}}
    &state=...
    &response_type=code
    &scope=openid+profile+email+offline_access+secure-photo-hub-audience
    &code_challenge=...
    &code_challenge_method=S256
    &nonce=...
```

OIDC - Authorization Code Flow

```
http://{{keycloak-hostname}}/realms/secure-photo-hub/protocol/openid-connect/auth
    ?client_id=secure-photo-hub-rest-api
    &redirect_uri={{secure_photo_hub_endpoint}}
    &state=...
    &response_type=code
    &scope=openid+profile+email+offline_access+secure-photo-hub-audience
    &code_challenge=...
    &code_challenge_method=S256
    &nonce=...
```

- **/realms/secure-photo-hub/protocol/openid-connect/auth:** è l'Authorization Endpoint di Keycloak usato per interagire con il Resource Owner e ottenere un Authorization Grant (in questo caso Authorization Code Grant). Secure Photo Hub reindirizza l'utente non autenticato su questo endpoint.
- **client_id:** è l'identificatore del client (che identifica Secure Photo Hub su Keycloak).
- **redirect_uri:** dopo che l'Authorization Server ha terminato la sua interazione con il Resource Owner, l'Authorization Server reindirizza lo user-agent del Resource Owner sul redirection endpoint di Secure Photo Hub (Resource Owner)
- **state:** un valore opaco usato da Secure Photo Hub per mantenere lo stato tra la richiesta e la callback (redirect). L'Authorization Server (Keycloak) include questo valore quando reindirizza l'utente su Secure Photo Hub

OIDC - Authorization Code Flow

```
http://{{keycloak-hostname}}/realms/secure-photo-hub/protocol/openid-connect/auth
    ?client_id=secure-photo-hub-rest-api
    &redirect_uri={{secure_photo_hub_endpoint}}
    &state=...
    &response_type=code
    &scope=openid+profile+email+offline_access+secure-photo-hub-audience
    &code_challenge=...
    &code_challenge_method=S256
    &nonce=...
```

- **scope:** gli scopes, in generale, sono meccanismi usati dal framework OAuth 2.0 per limitare l'accesso di un'applicazione all'account di uno utente. Tuttavia, questi possono essere usati anche per altri fini (con OpenID e Keycloak).
- **code_challenge & code_challenge_method:** questi parametri fanno parte del security layer PKCE (Proof Key for Code Exchange) che risiede sopra l'Authorization Code Grant per garantire che gli Authorization Codes non possano essere rubati o riutilizzati (vedi prossime slide per maggiori dettagli).
- **nonce:** questo parametro fa parte di OpenID Connect ed è opzionale. Verrà incluso dell'ID Token che Keycloak genererà successivamente. Secure Photo Hub potrà quindi verificare che l'ID Token sia nuovo, mitigando i Replay Attacks. Il nonce è un valore opaco di 8 byte generato in maniera random e codificato con Base64 URL.

OIDC - Authorization Code Flow

```
http://{{keycloak-hostname}}/realms/secure-photo-hub/protocol/openid-connect/auth
    ?client_id=secure-photo-hub-rest-api
    &redirect_uri={{secure_photo_hub_endpoint}}
    &state=...
    &response_type=code
    &scope=openid+profile+email+offline_access+secure-photo-hub-audience
    &code_challenge=...
    &code_challenge_method=S256
    &nonce=...
```

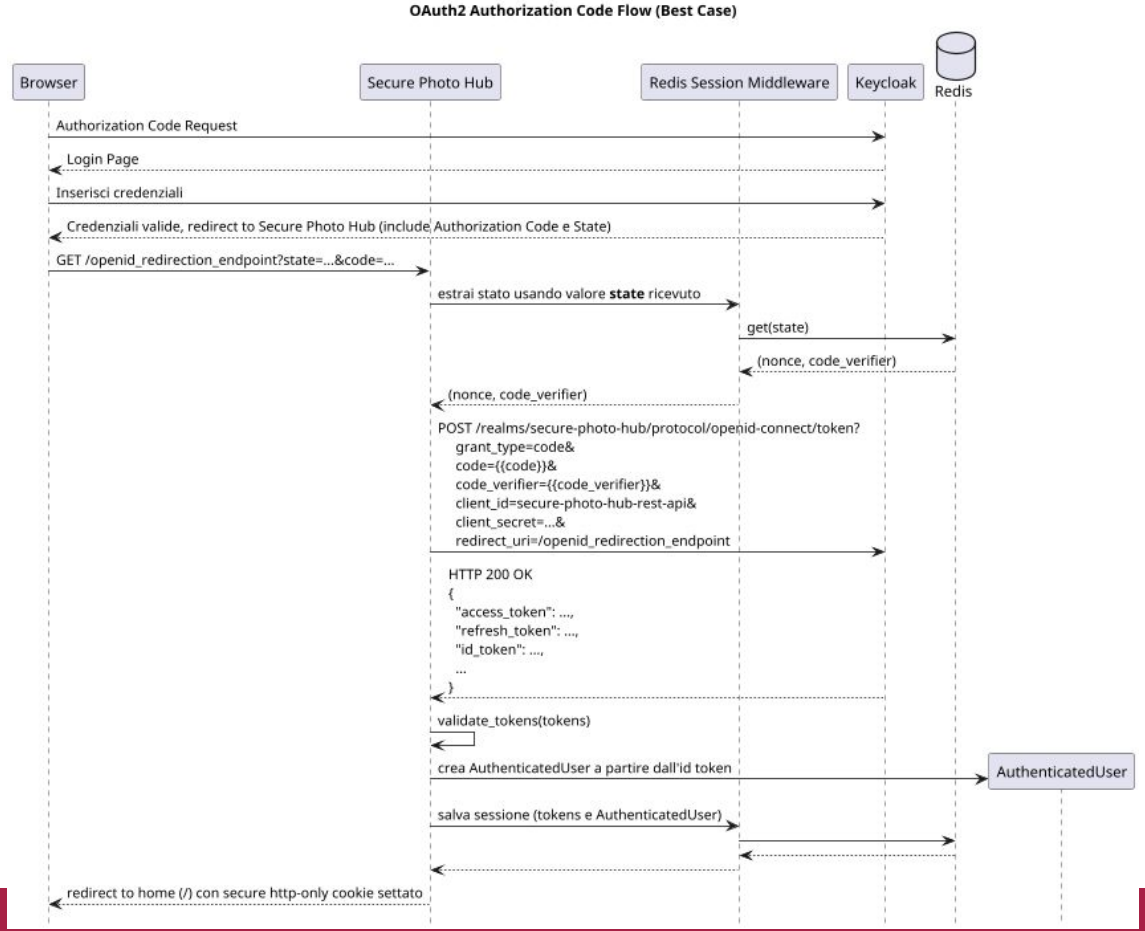
- I valori **state**, **code_challenge** e **nonce** devono essere memorizzati in una **server-side session**. Le sessioni vengono memorizzate in una mappa associativa sul server, con una chiave rappresentata da un cookie. Per garantire la sicurezza, è essenziale che i cookie siano:
  - **HTTP Only cookies**: un codice JavaScript malizioso nel browser non può leggere il valori dei cookies
  - **Secure cookies**: I cookies devono viaggiare su una connessione sicura (HTTPS)

OIDC - Authorization Code Flow (PKCE)

```
http://{{keycloak-hostname}}/realms/secure-photo-hub/protocol/openid-connect/auth
    ?client_id=secure-photo-hub-rest-api
    &redirect_uri={{secure_photo_hub_endpoint}}
    &state=...
    &response_type=code
    &scope=openid+profile+email+offline_access+secure-photo-hub-audience
    &code_challenge=...
    &code_challenge_method=S256
    &nonce=...
```

- **PKCE (Proof Key for Code Exchange)** è un livello di sicurezza che si trova sopra l'Authorization Code Grant per garantire che gli Authorization Codes non possano essere rubati o riutilizzati:
  1. L'applicazione genera una chiave segreta (chiamata *code verifier*) e crea un digest con SHA-256.
  2. L'applicazione invia quindi l'hash al server OAuth, che lo memorizza.
  3. Successivamente, quando l'applicazione riceve i token dal server OAuth, l'applicazione invierà al server la chiave segreta e il server OAuth verificherà che l'hash della chiave segreta fornita corrisponda al valore fornito in precedenza.
- PKCE previene i CSRF e Authorization Code Injection Attacks.
- Non è obbligatorio, ma lo dovrebbe essere quando il Client è un Public Client e non un Confidential Client. Secure Photo Hub è un Confidential Client, ma nonostante ciò, è stato implementato questo ulteriore livello di sicurezza per la sua semplicità di implementazione.

OIDC - Authorization Code Flow (Browser) 2/2



Flussi di autorizzazione per l'accesso alle risorse

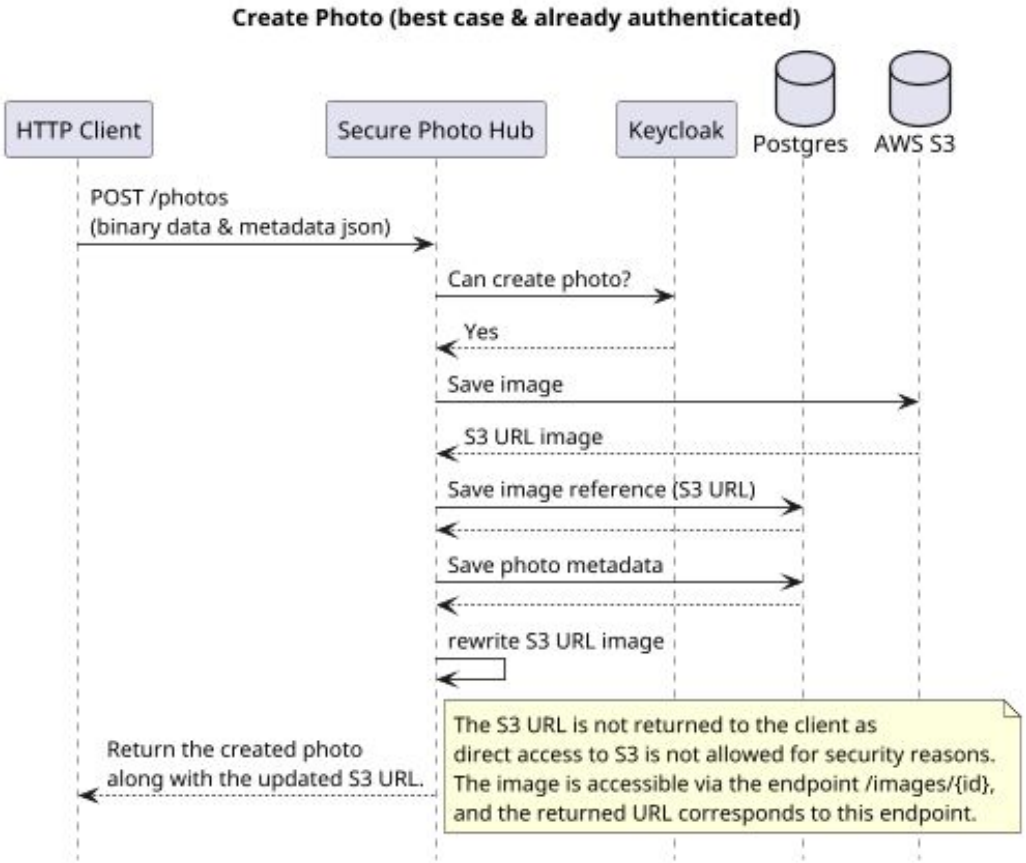
- Secure Photo Hub, prima di accedere a qualsiasi risorsa, interagisce con Keycloak per verificare se l'utente autenticato ha i permessi necessari per eseguire azioni sulla risorsa richiesta.

- L'applicazione utilizza i Policy Enforcers precedentemente descritti per comunicare con Keycloak.

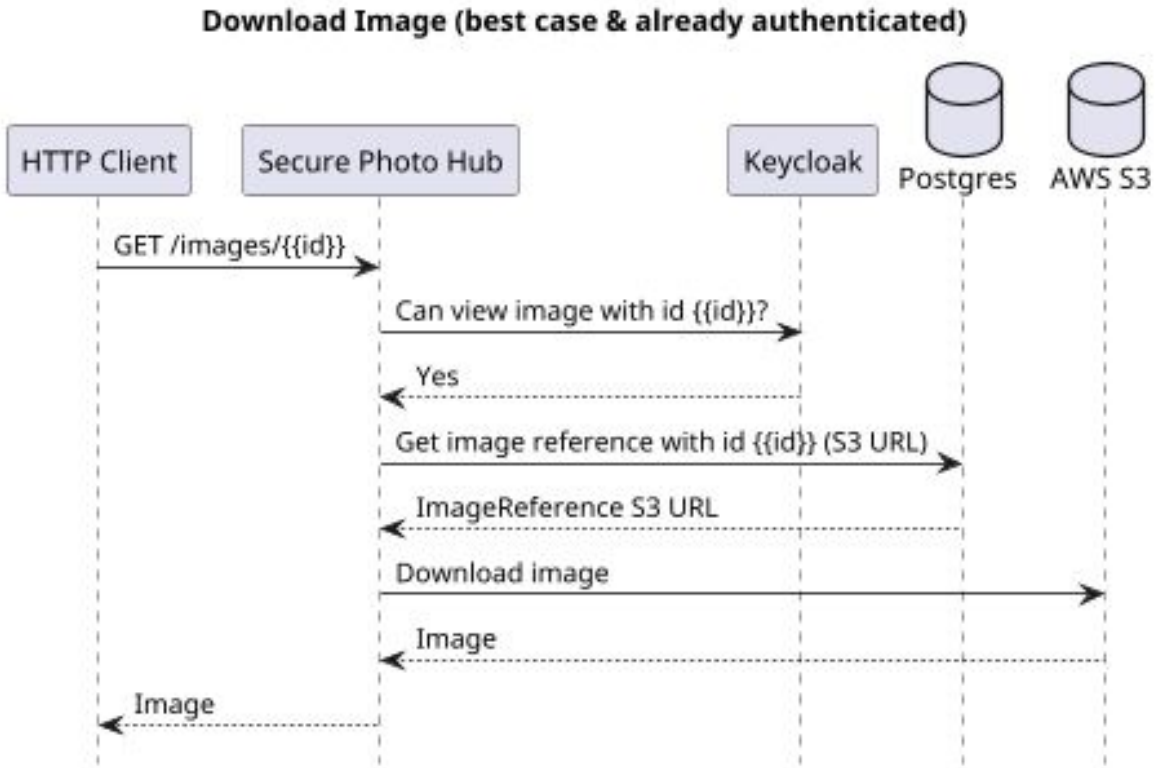
- Nelle prossime slide, saranno presentati dei sequence diagrams che evidenziano questi processi, assumendo che l'utente sia già autenticato e che il flusso di scambio informazioni avvenga senza errori.

- Tutti gli altri endpoint seguono un flusso simile

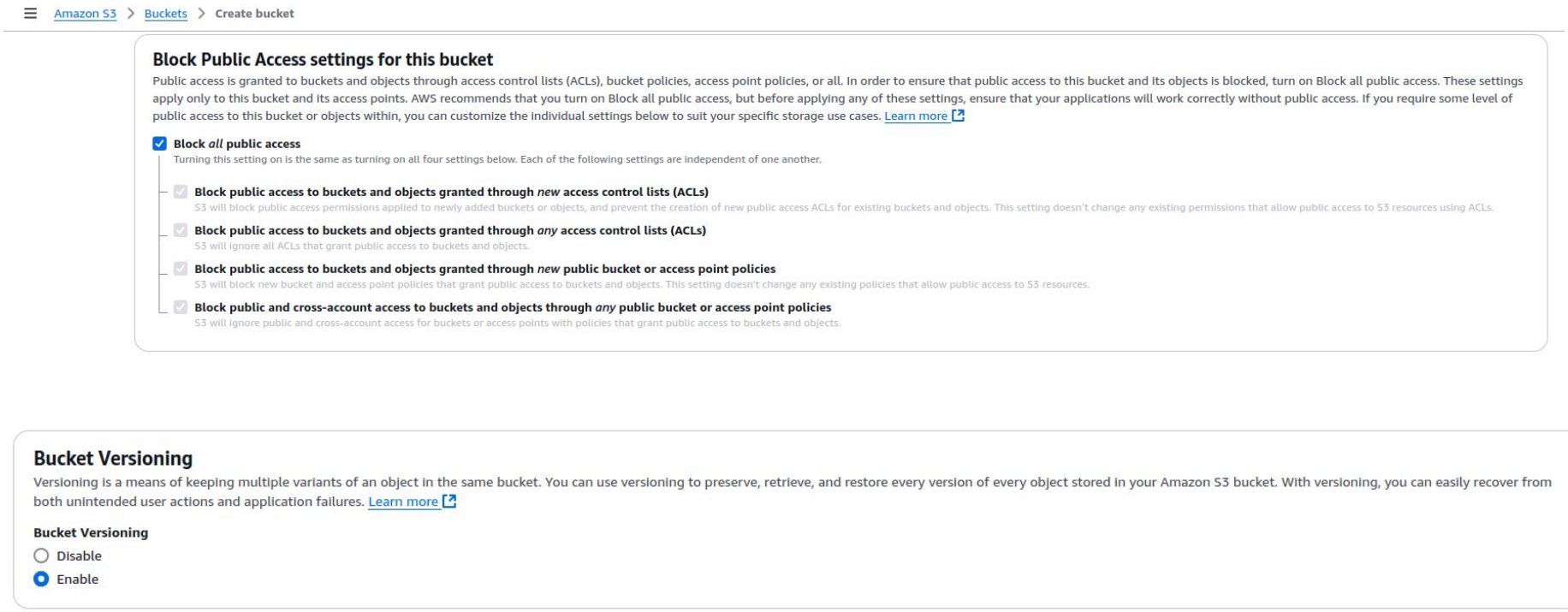
Flussi di autorizzazione per l'accesso alle risorse



Flussi di autorizzazione per l'accesso alle risorse



AWS S3 - Creazione Bucket



AWS S3 - Creazione Bucket

Default encryption Info

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type | Info

- (•) Server-side encryption with Amazon S3 managed keys (SSE-S3)
- ( ) Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- ( ) Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
  Secure your objects with two separate layers of encryption. For details on pricing, see **DSSE-KMS pricing** on the **Storage** tab of the Amazon S3 pricing page.

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more

- ( ) Disable
- (•) Enable

▼ Advanced settings

Object Lock

Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. Object Lock works only in versioned buckets. Learn more

- ( ) Disable
- (•) Enable
  Permanently allows objects in this bucket to be locked. Additional Object Lock configuration is required in bucket details after bucket creation to protect objects in this bucket from being deleted or overwritten.

ⓘ Object Lock works only in versioned buckets. Enabling Object Lock automatically enables Versioning.

⚠ **Enabling Object Lock will permanently allow objects in this bucket to be locked**
After you enable Object Lock for a bucket, you can't disable Object Lock or suspend Versioning for that bucket. Learn more about Using Object Lock

- [ ] I acknowledge that enabling Object Lock will permanently allow objects in this bucket to be locked.

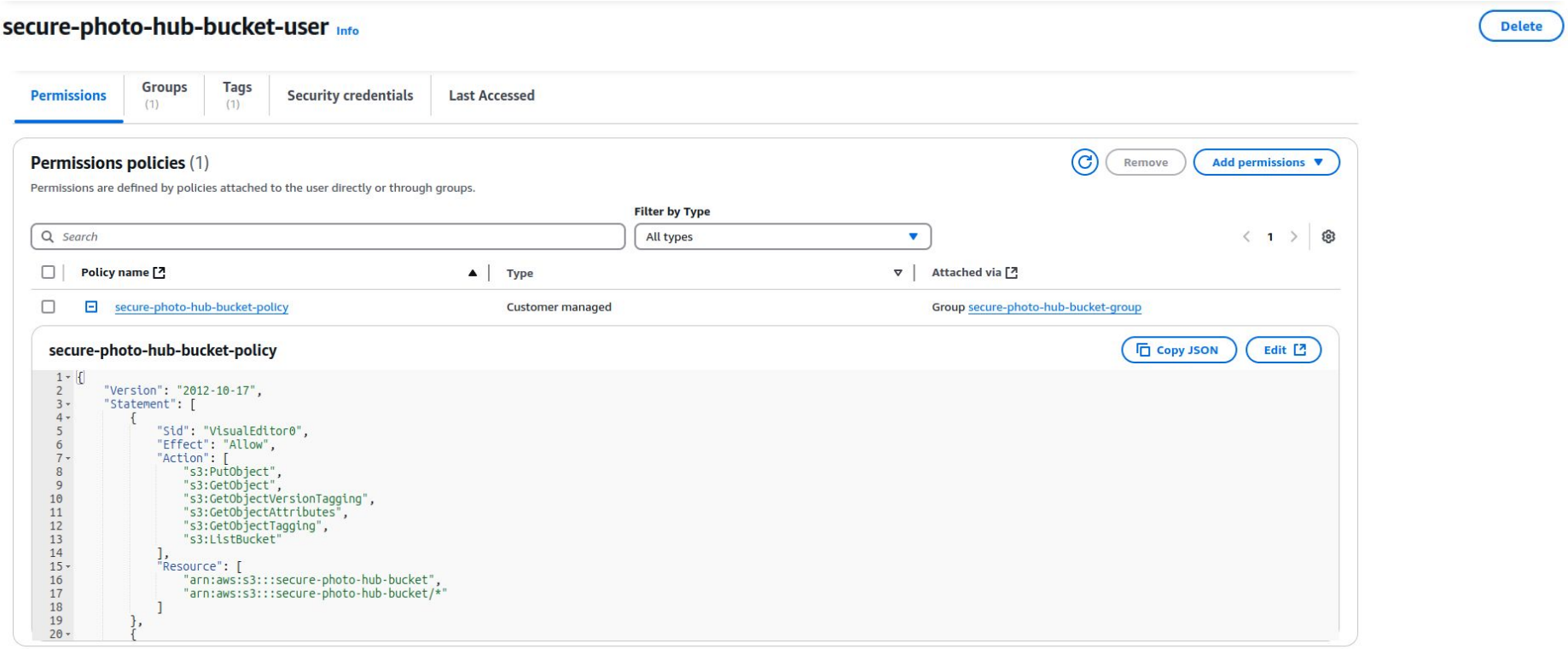
AWS S3 - Creazione Policy

```
{
 "Version": "2012-10-17",
 "Statement": [
   {
     "Sid": "VisualEditor0",
     "Effect": "Allow",
     "Action": [
       "s3:PutObject",
       "s3:GetObject",
       "s3:GetObjectVersionTagging",
       "s3:GetObjectAttributes",
       "s3:GetObjectTagging",
       "s3:ListBucket"
     ],
     "Resource": [
       "arn:aws:s3:::secure-photo-hub-bucket",
       "arn:aws:s3:::secure-photo-hub-bucket/*"
     ]
   },
   {
     "Effect": "Deny",
     "NotAction": [
       "s3:PutObject",
       "s3:GetObject",
       "s3:GetObjectVersionTagging",
       "s3:GetObjectAttributes",
       "s3:GetObjectTagging",
       "s3:ListBucket"
     ],
     "NotResource": [
       "arn:aws:s3:::secure-photo-hub-bucket",
       "arn:aws:s3:::secure-photo-hub-bucket/*"
     ]
   }
 ]
}
```

Consente le azioni di lettura e scrittura (PutObject, GetObject, ListBucket, ecc.) sul bucket secure-photo-hub-bucket e sui suoi oggetti.

Blocca tutte le azioni non elencate nel primo statement sul bucket e sui suoi oggetti.

AWS S3 - Creazione utente con permessi



Grazie per l’attenzione

https://github.com/vtramo/secure-photo-hub