La protezione dei dati

Claudio Cappello
Co-fondatore
Project Manager
4Solid s.r.l.

Relatore

Claudio Cappello

Co-fondatore e Project Manager presso 4Solid srl

Prima di iniziare l'attività di Project Manager ho seguito, per oltre 2 decenni, lo sviluppo di software in diversi campi applicativi e per molteplici clienti facendo uso delle più disparate tecnologie.

Fin dall'inizio della creazione della Piattaforma IoT, che rappresenta uno dei prodotti di 4Solid, ho coordinato le diverse fasi operative del suo sviluppo mediante applicazione di metodologie Agile di cui sono convinto sostenitore.

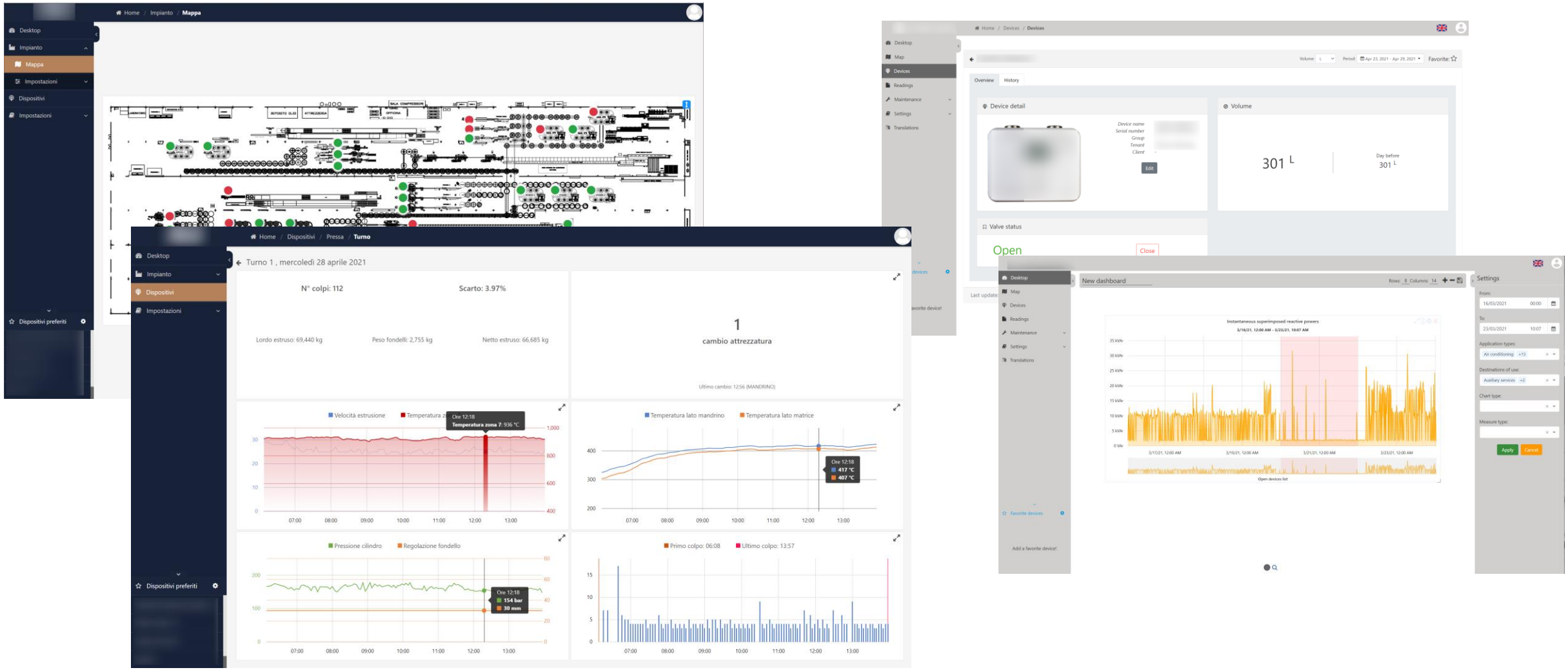
L'attenzione alla protezione dei dati è stato uno dei principali valori della Piattaforma a cui abbiamo dato particolare attenzione nelle diverse fasi implementative e che tuttora rimane una costante nell'evoluzione del prodotto.

Contatto: claudio.cappello@4solid.it

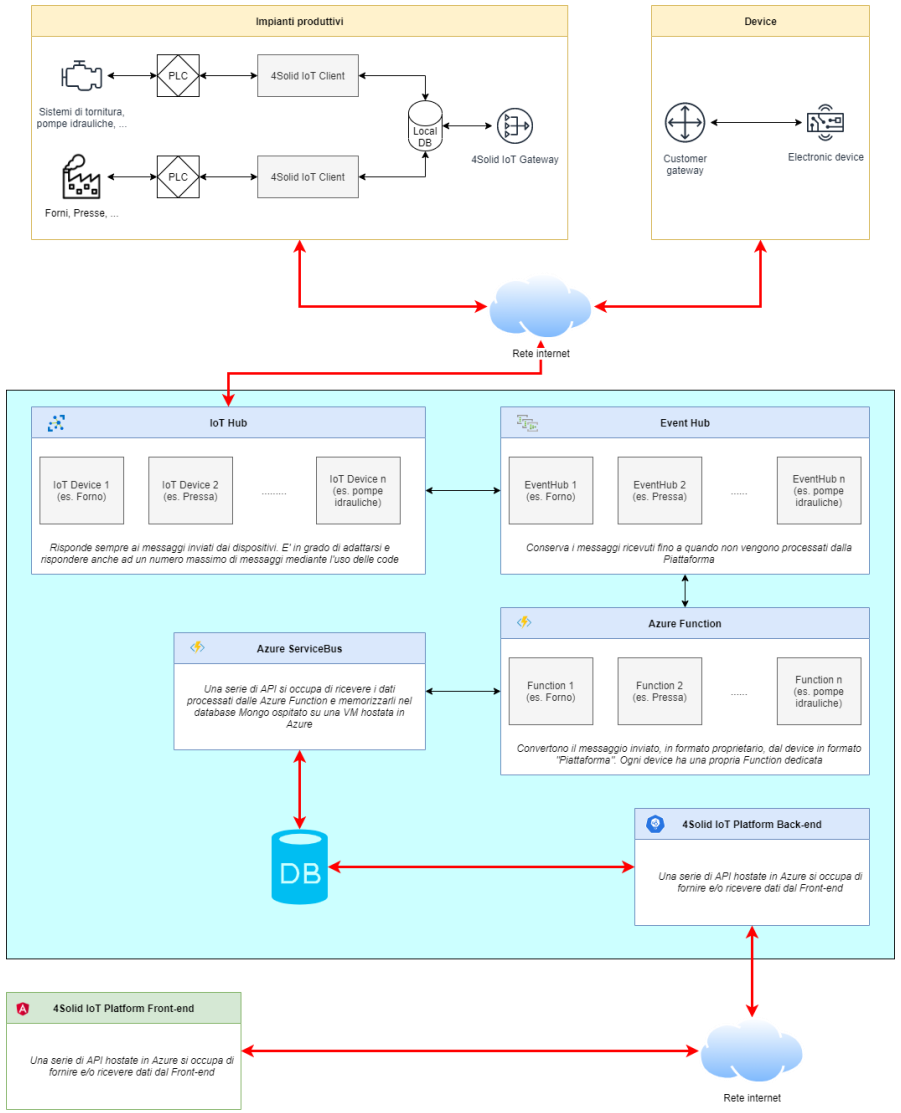
Argomenti trattati

- Dal Device al Cloud, come garantire la sicurezza nella trasmissione del dato: SAS token-based authentication e individual X.509 certificate authentication
- La sicurezza nell'accesso ai database on-Cloud tramite l'uso di API e Azure Key Vault
- Uso di Token e politiche di Refresh Token per l'accesso alle API
- Tecnologie di Blockchain per la certificazione e autenticazione del dato
- Question & Answer

Piattaforma IoT 4Solid



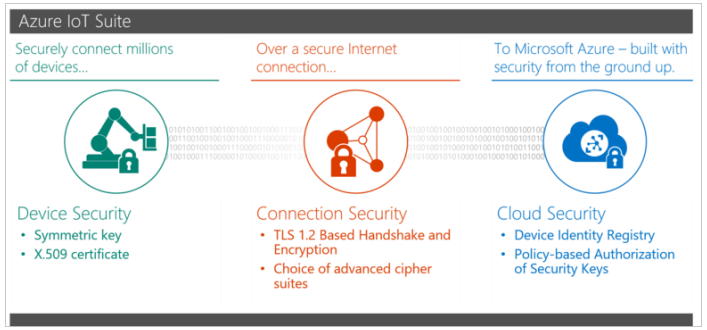
Architettura della Piattaforma IoT



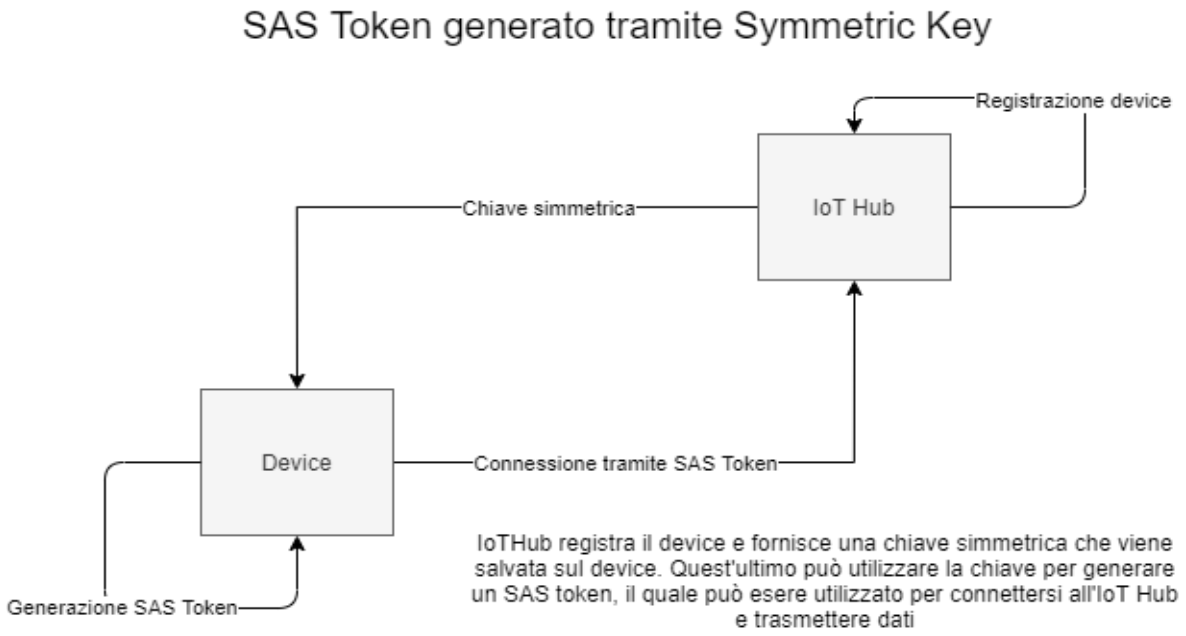
SAS Token Authentication e X.509 Certificate

- Dobbiamo sempre garantire la sicurezza dei dati inviati dai dispositivi alla Piattaforma IoT e conseguentemente all'infrastruttura Azure IoT su cui si basa. Possiamo distinguere 3 aree fondamentali nelle quali garantire questa sicurezza:
  - **Sicurezza del dispositivo: proteggere il dispositivo IoT mentre viene distribuito in circostanze normali;**
  - *Sicurezza delle connessioni*: garantire che tutti i dati trasmessi tra il dispositivo IoT e l'hub IoT siano riservati e a prova di manomissione;
  - *Sicurezza del cloud*: fornire un mezzo per proteggere i dati durante il trasferimento e l'archiviazione nel cloud.

- Esistono 2 possibili metodi da utilizzare per garantire la «**Sicurezza del dispositivo**»:
  - Utilizzo di una **chiave simmetrica** posseduta da ogni dispositivo e utilizzata per generare un Token SAS, quest'ultimo usato per comunicare con l'IoT Hub. Ogni chiamata all'IoT hub viene autenticata associando la chiave simmetrica
  - Utilizzo di un **certificato X.509** (sul dispositivo) e una chiave privata per autenticare il device nell'IoT Hub, garantendo che la chiave privata sul dispositivo non sia mai nota all'esterno. Consente di autenticare un dispositivo IoT a livello fisico come parte del processo di creazione della connessione TLS

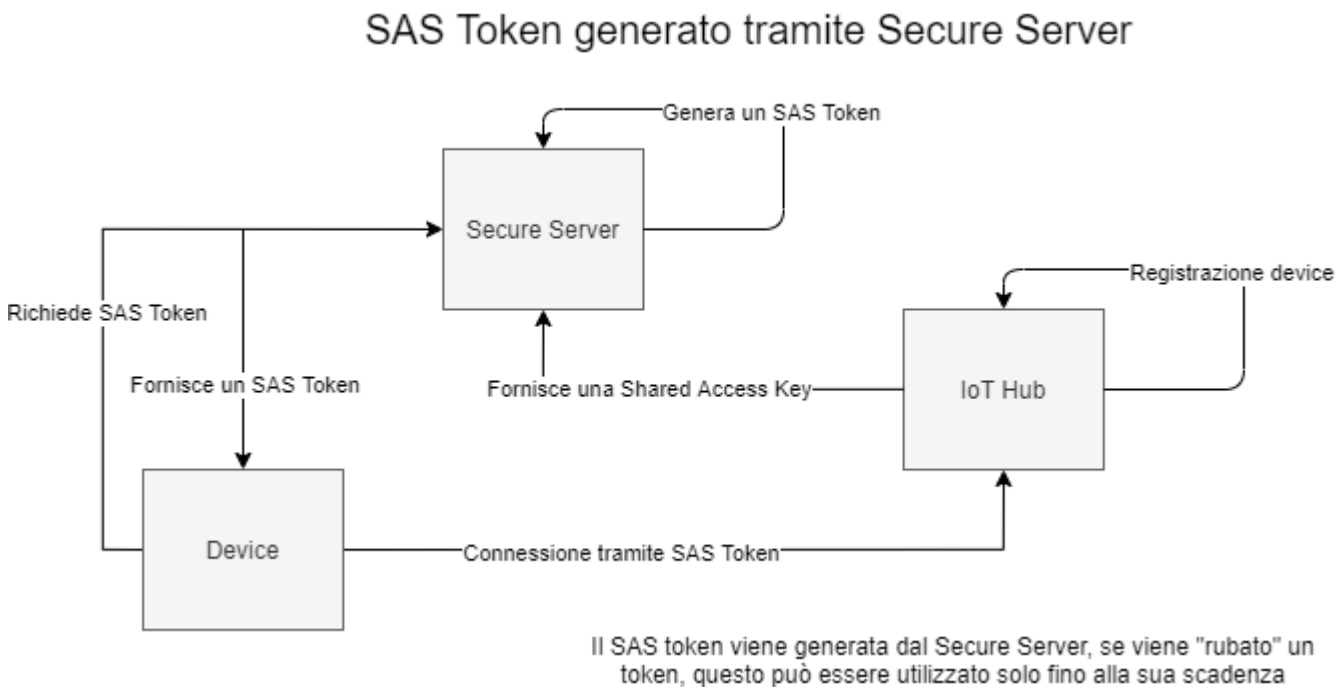


- Se utilizziamo le chiavi simmetriche possiamo avvalerci di 2 pattern:
  - **Chiave simmetrica salvata sul dispositivo**, il quale la utilizza per generare token utilizzati per connettersi all'IoT Hub e inviare messaggi



- *Vantaggi*: facilità nel deployment dei dispositivi
- *Svantaggi*: se il device viene compromesso si possono generare Token SAS validi fino alla disabilitazione del device

- **Token SAS fornito da un sistema intermediario «Secure Server»**, il quale fornisce un Token senza la necessità di salvare la chiave simmetrica sul dispositivo



- *Vantaggi*: se il device viene “compromesso” si può utilizzare il Token SAS solo fino alla sua scadenza
- *Svantaggi*: dobbiamo assicurarci che esista un flusso di autenticazione, diverso di quello dell’IoT Hub, fra dispositivo e Secure server, quindi aumento nella complessità del sistema

- Autenticazione del dispositivo basata su un **certificato di autenticazione X.509**
- La chiave privata viene archiviata in modo sicuro nel dispositivo e non può essere individuata all'esterno del dispositivo. Il certificato x.509 contiene informazioni sul dispositivo (ad esempio l'ID dispositivo) e altri dettagli dell'organizzazione. Una firma del certificato viene generata utilizzando la chiave privata.
- L’autenticazione di più dispositivi verso l’IoT Hub viene garantito dall’uso di un Certificato acquistato da un’autorità che garantisce la legittimità dei dispositivi
- Registrare un certificato permette l’autenticazione di più dispositivi

Azure Key Vault

- I dati sensibili (stringhe di connessione) non vengono più embeddati nel codice o in file di configurazione (appSettings.json), nemmeno se criptati

- **Azure Key Vault** (**AKV**) offre una serie di soluzioni che permettono di salvaguardare gli aspetti che devono rimanere segreti nella nostra applicazione, per archiviarli, controllarli ed accedervi in modo sicuro una volta archiviati su Azure:
  - **Gestione dei secret**: archiviazione e controllo dell'accesso a token, password, certificati, chiavi API
  - **Gestione delle chiavi**: creazione e controllo delle chiavi di crittografia usate per crittografare i dati
  - **Gestione dei certificati**: gestione e distribuzione dei certificati

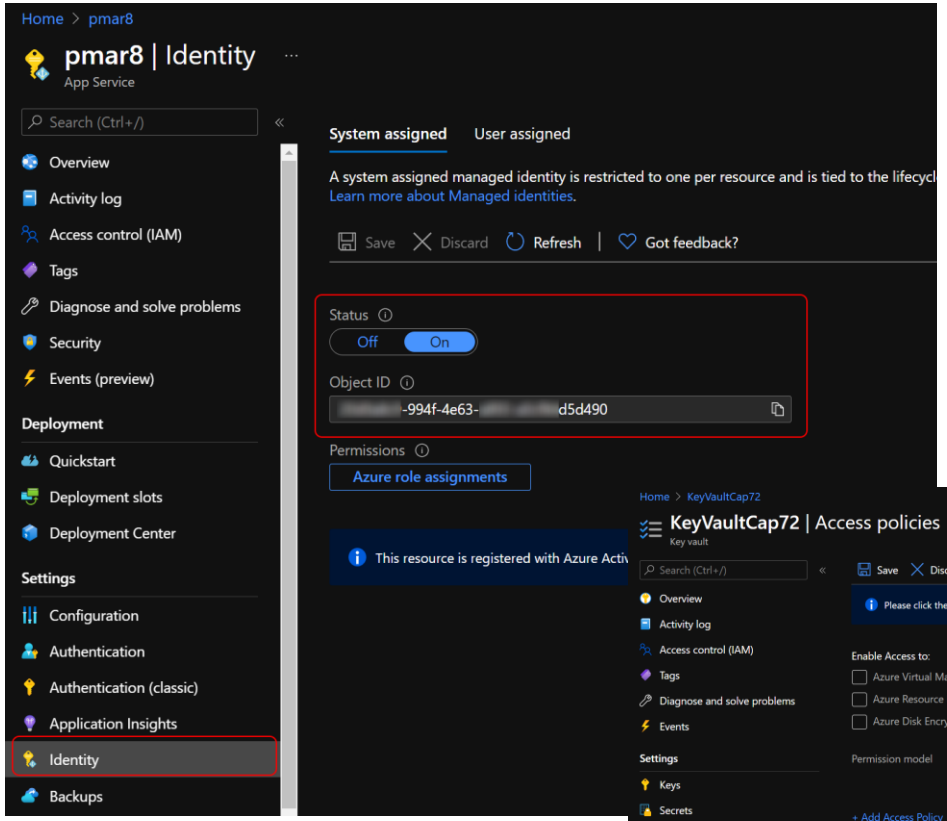
- AKV utilizza **Hardware Security Module** (HSM) certificati FIPS140-2 livello 2. Si tratta di reti di computer sicure in grado di eseguire operazioni di crittografia

- Abbiamo utilizzato i Secret per archiviare e controllare l'accesso a password e altre informazioni indispensabili alle API per accedere ai database e fornire i dati alla Piattaforma

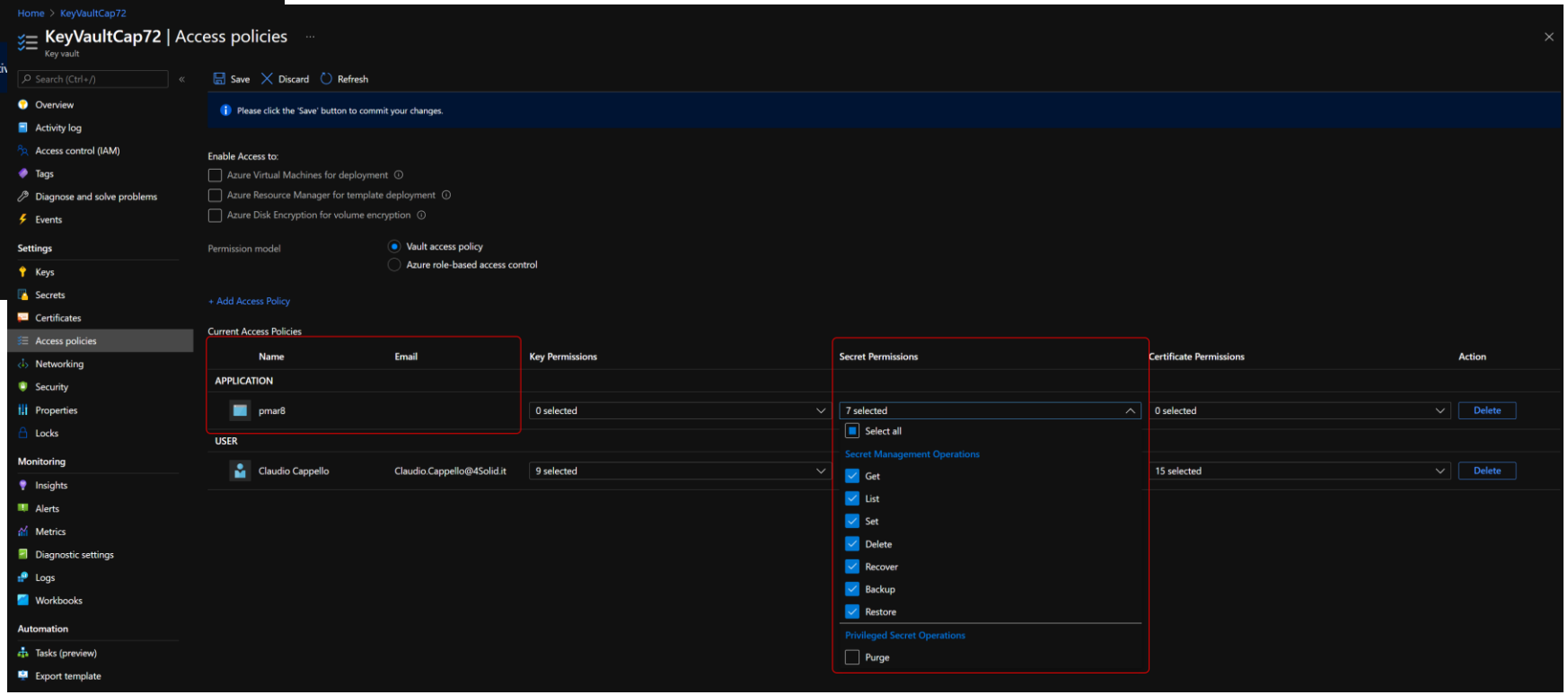
- Abbiamo 2 modalità di controllo degli accessi al AKV:
  - Registrare gli App Service con Azure Active Directory per garantire l'accesso a risorse (in questo caso AKV) protette dal Azure AD
    A system assigned managed identity is restricted to one per resource and is tied to the lifecycle of this resource. You can grant permissions to the managed identity by using Azure role-based access control (Azure RBAC). The managed identity is authenticated with Azure AD, so you don't have to store any credentials in code);

  - Utilizzare utenti configurati nel Azure AD a cui vengono fornite autorizzazioni di accesso al AKV
    User assigned managed identities enable Azure resources to authenticate to cloud services (Azure Key Vault) without storing credentials in code. This type of managed identities are created as standalone Azure resources and have their own lifecycle. A single resource (Virtual Machine) can utilize multiple user assigned managed identities. Similarly, a single user assigned managed identity can be shared across multiple resources (Virtual Machine)

- Sfruttiamo le funzionalità built-in messe a disposizione da Azure per garantire sicurezza e controllo degli accessi a informazioni sensibili



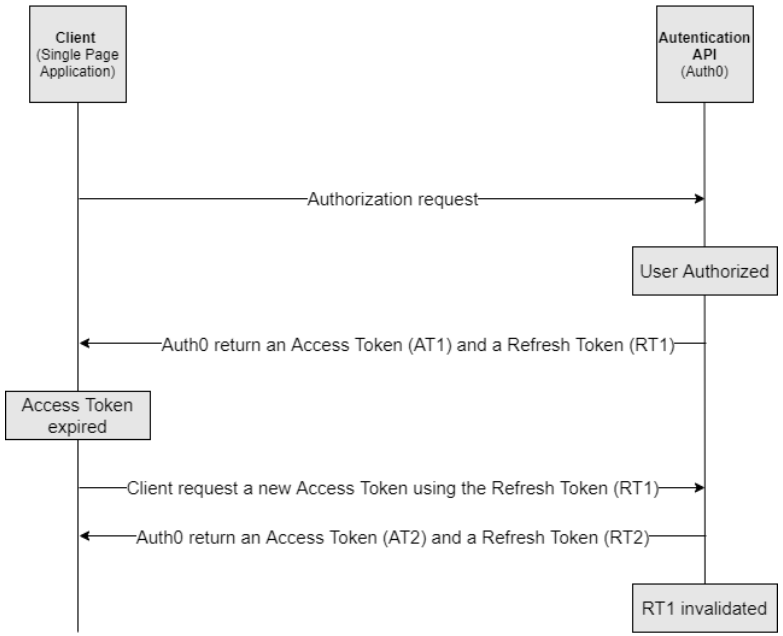

```
public async Task OnGetAsync()
{
    try
    {
        AzureServiceTokenProvider azureServiceTokenProvider = new AzureServiceTokenProvider();

        KeyVaultClient keyVaultClient = new KeyVaultClient(new KeyVaultClient.AuthenticationCallback(azureServiceTokenProvider.KeyVaultTokenCallback));

        var secret = await keyVaultClient.GetSecretAsync(GetKeyVaultSecretsEndpoint()).ConfigureAwait(false);

        Message = secret.Value;
    }
    catch (KeyVaultErrorException keyVaultException)
    {
        Message = keyVaultException.Message;
    }
}
```



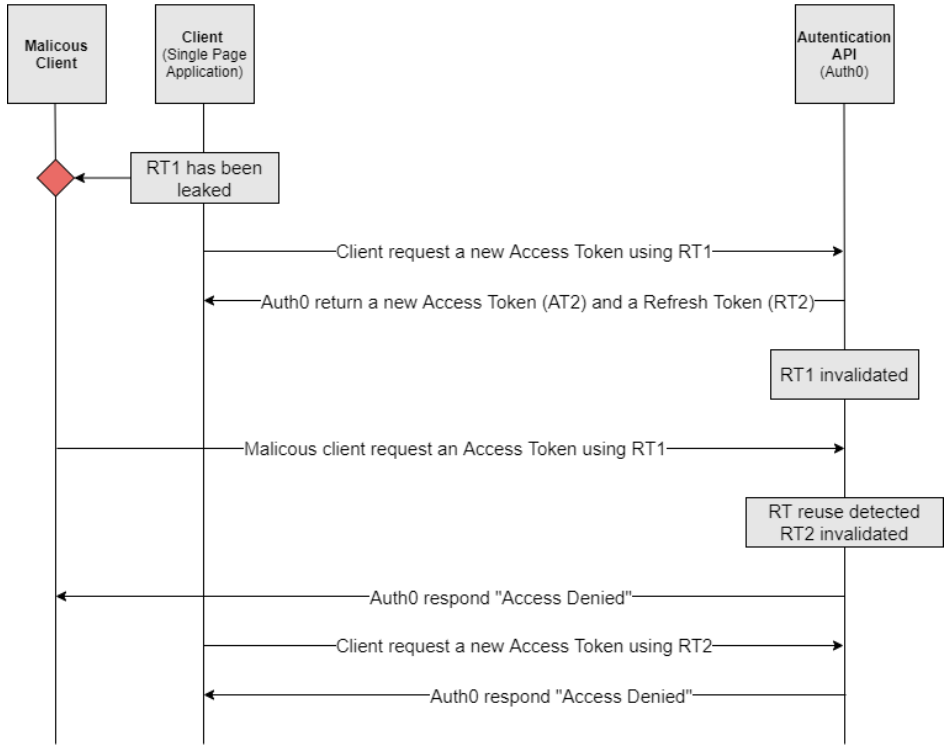
Token e Refresh Token

- L’accesso alla piattaforma avviene tramite form di Autenticazione (e-mail, password)
- Il servizio di autenticazione basato su Autho rilascia un Token Bearer di brevissima dura
- Insieme al Token viene rilasciato una SessionId (permette login su dispositivi diversi utilizzando medesime credenziali) e un Refresh Token (long time experation), utilizzabili one-time per il rinnovo del Token stesso (Refresh Token Rotation)
- Il Token viene salvato nel local storage del client **previa criptazione**
- Ogni API della piattaforma è protetta e l’accesso è consentito solamente con utilizzo di Token validi **(ogni chiamata all’API passa alla stessa un Token che viene controllato mediante un middle-ware che ne verifica la validità)**
- Se un Refresh Token non risulta più valido vengono automaticamente invalidati tutti i Refresh Token della «stessa famiglia». Tutte le successive richieste alle API richiedono nuova Autenticazione dell’utente. Questo meccanismo funziona indipendentemente da quale dei Client (Malicous o Legitimate) richiede per primo un nuovo Token. Appena viene individuato un «riuso» del Refresh Token viene richiesta una nuova autenticazione
- JWT token diventa ancora più sicuro quando combinato con JSON Web Signature (JWS) per validarne il contenuto e con JSON Web Encryption (JWE) per «oscurare» le informazioni al client

Access Token e Refresh Token



Automatic reuse detection



Blockchain

- Stiamo introducendo l'uso delle Blockchain nella Piattaforma IoT al fine di:
  - Utilizzare il registro distribuito presente nei sistemi di Blockchain in quanto a prova di manomissione dei dati e aumentare la fiducia tra le parti coinvolte (chi invia il dato e chi lo riceve)
  - Utilizzare la Blockchain per archiviare i dati IoT aggiungendo un ulteriore livello di sicurezza che dovrebbe essere aggirato da eventuali hacker per accedere alla rete e ai dati
  - Utilizzare il robusto livello di crittografia fornito dalla Blockchain per rendere impossibile la sovrascrittura dei dati esistenti
  - Garantire quando il dato è stato inviato e quando è stato ricevuto, nonché garantirne l'integrità e la possibilità di tracciarne il percorso
  - Poter identificare dove e quando può esserci stato una perdita di dati e intraprendere azioni correttive rapide
  - Garantire, tramite la tecnologia di registro distribuita, che al crescere del numero di dispositivi connessi il sistema sia in grado di scalare rapidamente garantendo l'elaborazione di un maggior numero di transazioni