ITS_DC_SLG06_ISAM LINEE GUIDA PER INTEGRAZIONE DI APPLICAZIONI

	Struttura aziendale	Responsabile
Redatta da:	**DIDT/CTO/SCR**	
Condivisa con:	**DIDT/CISO**	
Approvata da:	**DIDT**	

EXECUTIVE SUMMARY

Obiettivi del documento	• **Linee guida per integrazione di applicazioni con ISAM Verify**

INDICE DEI CONTENUTI

1. SCOPO E CAMPO DI APPLICAZIONE

Il presente documento descrive le linee guida ad uso degli sviluppatori per l'integrazione di applicazioni (gestite e/o sviluppate dalla Direzione IT e Digital Transformation di ASPI ed in uso in ASPI e/o Società controllate e Società Terze sulla base di Contratti di Service IT) con l'Identity Provider ASPI (IBM Security Verify Access, in precedenza ISAM), ai fini di autenticazione e autorizzazione degli utenti e/o dei client che accedono alle applicazioni stesse.

2. CONTROLLO ACCESSI

Le presenti linee guida aderiscono alle policies stabilite dal CISO ASPI a recepimento degli standard in materia[1].

L'implementazione di queste linee guida vuole perseguire molteplici scopi:

- ottenere un controllo di gestione degli accessi integrato, in termini di autenticazione e autorizzazione, per web application on-prem, su cloud privato, su cloud pubblici, per componenti mobile
- assicurare il Single Sign-On per accessi che insistono sulle stesse istanze IdP
- permettere la federazione di identità
- applicare l'autenticazione a più fattori.

Riguardo le policies adottate per l'*MFA* e le possibili modalità di applicazione si rimanda al paragrafo dedicato.

Si fa notare che la maschera di login presentata all'utente è quella propria di ISAM, condivisa quindi per tutte le applicazioni integrate e non quella proprietaria dell'applicazione stessa.

Si ricorda inoltre, nel caso di dispositivi al di fuori del dominio ASPI, di importare fra quelle attendibili la *CA Autostrade*, che firma (salvo casi di utilizzo di certificati pubblici) i certificati in uso a ISAM, al fine di evitare alert di sicurezza.

In ottemperanza alle policies aziendali di revoca di abilitazione ad una applicazione per mancato utilizzo[2], inoltre, tramite i log di accesso a ISAM vengono anche tracciati[3] i *Last Login* degli utenti alle applicazioni per successiva elaborazione. In caso di integrazione con ISAM, dunque, gli sviluppatori delle applicazioni sono esonerati dalla registrazione dei last login nelle apposite tabelle.

[1] https://dwisrv.cto-utilities.prod.aws.autostrade.it/en/Public/wiki-security/abb-repository/sysint-abb
[2] ..procedure
[3] Al momento gestiti per use case Giunzione e Federazione; non ancora implementato per OAuth

Le applicazioni in questione devono essere preliminarmente censite nel Catalogo Applicazioni Software (GCS) e ivi riportare criteri di Autenticazione e Autorizzazione coerenti con le modalità di integrazione prescelte, sulla base di queste linee guida. La verifica della veridicità e della coerenza delle informazioni riportate è a carico del RIA dell'applicativo stesso.

Queste linee guida si riferiscono alle applicazioni censite in GCS con criterio di Autenticazione denominati:

- "ISAM intranet/internet" (use case delle "giunzioni" specificato nel seguito del documento)

- "Federazione" (use case SAML specificati nel seguito del documento)

- "ISAM Mobile" (use case OAuth specificato nel seguito del documento).

Applicazioni censite con altri criteri di Autenticazione non rientrano nello scope di questo documento. L'integrazione con ISAM resta comunque raccomandata.

L'IdP ASPI in oggetto (ISAM) ha in carico l'autorizzazione degli utenti nel caso in cui l'applicazione si censita in GCS con criterio di Autorizzazione denominato:

- "Gruppi LDAP"

- "Ruolo DIG" (nei casi in cui il ruolo aggregato includa gruppi LDAP).

In questi casi il RIA ha in carico l'esplicita richiesta di configurazione dei relativi filtri (ACL).

Si raccomanda l'utilizzo di criteri autorizzativi gestiti centralmente[4] (quali i gruppi LDAP), evitando l'utilizzo di configurazioni locali negli applicativi ad opera degli amministratori delle applicazioni.

Nel modello di controllo accessi implementato è possibile far uso, insieme ai gruppi autorizzativi menzionati, di ulteriori attributi utente (che possono concorrere o meno alla definizione del profilo di accesso alla applicazione), oltre quelli anagrafici, trasferiti da ISAM all'applicativo:

- userid (pari alla matricola)
- Nome Cognome
- Mail
- Attributi di tipo organizzativo (da concordare).

[4] Modello di autorizzazione su base ruolo autorizzato dal Gestore Delegato

3. ARCHITETTURA E AMBIENTI

ISAM è costituito da:

- Policy Server, l'ambiente di configurazione
- LDAP server, directory che contiene utenti e gruppi
- WebSeal Server, il reverse proxy utilizzato per accedere alle applicazioni. È l'elemento che colloquia con il server di autenticazione. Sul server WebSeal si definiscono gli oggetti - detti giunzioni - che rimappano gli indirizzamenti verso le risorse interne.

Sono disponibili per l'integrazione ambienti di PROD e di NO PROD:

Produzione

- Istanza intranet **www1.gruppo.autostrade.it**
- Istanza internet **www2.autostrade.it**

Quality

- Istanza intranet **qa.gruppo.autostrade.it**
- Istanza internet **dmzqa.autostrade.it**

Test

- Istanza intranet **intranettst.gruppo.autostrade.it**
- Istanza internet **dmztst.autostrade.it**

Gli ambienti intranet sono raggiungibili solo da rete locale o tramite VPN.

Gli ambienti internet sono raggiungibili da internet e anche da rete locale o VPN.

Per ogni ambiente è bene definire l'utilizzo di una sola istanza, esposta o meno su internet.

Il SSO sarà attivo per applicazioni integrate sulla stessa istanza.

La definizione delle utenze è distinta tra gli ambienti di Produzione/Quality e di Test, pur mantenendo la stessa definizione degli userid. Questo vuol dire che le password tra Produzione/Quality e Test non sono allineate e l'utente deve utilizzare le credenziali specifiche dell'ambiente in cui opera. I gruppi autorizzativi sono definiti e popolati in maniera indipendente nei diversi ambienti, anche se è consigliato mantenere l'allineamento nella definizione dei gruppi (non degli utenti associati).

4. USE CASE

Di seguito si riportano i possibili casi di utilizzo delle diverse configurazioni supportate per l'integrazione.

Giunzioni

Questa configurazione è possibile per integrare applicazioni web on prem o su cloud privato.

L'integrazione avviene tramite reverse proxy, quindi l'applicativo deve demandare l'autenticazione a ISAM, senza riproporre nuovamente una pagina di login (SSO).

Se l'applicativo deve gestire utenti e gruppi autorizzativi, deve essere in grado di utilizzare gli headers http (iv-user; iv-groups).

Si richiede il supporto della Mutua Autenticazione TLS (MTLS).

Si noti che l'utilizzo della giunzione è l'unico tra quelli citati in questo documento che non richiede la comunicazione diretta tra client e server.

Saml2.0

Con protocollo SAML è possibile integrare applicazioni web su SaaS IaaS PaaS esterni, oppure privati nel caso in cui non sia possibile utilizzare una giunzione.

Se il sistema da integrare prevede anche una componente mobile, si può invece optare per la configurazione in Oauth anche la parte web.

Le saml request devono essere firmate.

Oauth2.0

Sono possibili diverse implementazioni di Oauth in relazione ai casi d'uso.

1. Client credentials (machine-to-machine)

Rientrano in questa casistica le applicazioni client che accedono a risorse per proprio conto (sono esse stesse Resource Owner) o che comunque *non richiedono il consenso dell'utente*.

Ad esempio: procedure batch, cron job e altre applicazioni eseguite lato server senza alcuna interazione utente. Rientrano altresì nella medesima casistica anche altre applicazioni che devono accedere a risorse non sotto la protezione di utenti.

2. **Authorization code PKCE**

Rientrano in questo scenario le applicazioni client di tipo "native" o *"public"* e quindi per loro natura non affidabili nel custodire credenziali in modo sicuro.

Esempio tipico di questi client sono le applicazioni mobile e le single page application.

L'estensione "PKCE" del pattern Authorization Code svolge il compito di garantire il corretto rilascio in sicurezza dell'access token al Client anche in caso di fraudolenta acquisizione dell'Authorization Code tramite intercettazione durante gli scambi (con redirezioni) tra l'Authorization Server, il Client e lo User Agent.

In questo caso si deve autenticare l'utente su client mobile.

Si utilizza in alcuni casi per webapp che non supportano saml.

3. **Authorization code**

Rientrano in questa casistica le applicazioni client che operano per conto di un utente e che, essendo in esecuzione su un server, hanno la possibilità di custodire credenziali di accesso in modo sicuro (*"confidential"* client).

Possibili esempi di tali applicazioni sono le web application o in generale applicazioni erogate nel contesto di framework come Spring Boot o .Net.

La definizione di "confidential" client per implementazione senza estensione pkce deve essere approvata da CISO.

Per qualsiasi di queste configurazioni è necessario preliminarmente il censimento della applicazione nel catalogo software aziendale.

Il criterio di autenticazione/autorizzazione dichiarato in GCS deve essere coerente con le configurazioni richieste.

Eventuali richieste di eccezione all'adozione degli standard sopra definiti in relazione al caso d'uso devono essere inoltrate a CISO.

4. PARAMETRI DI CONFIGURAZIONE

Si descrive in questo paragrafo il dettaglio dei flussi nei differenti casi e si forniscono le informazioni utili alle configurazioni.

Giunzioni

Siamo nel caso di utilizzo del WebSeal come reverse proxy.

La chiamata https del browser viene prima intercettata da WebSeal e quindi ridiretta alla risorsa /context / della applicazione. È quindi necessario impostare una pagina di ingresso, nel caso in cui non vi sia una "welcome page" predefinita.

È preferibile che il context root della applicazione coincida col codice applicativo utilizzato per la definizione della giunzione (si parla di giunzione "trasparente").

Per configurare la giunzione è necessario sia definito nel DNS l'alias di servizio con cui il reverse proxy contatterà il backend.

Il server di backend deve esporre il servizio in HTTPS. Il certificato presentato deve essere emesso dalla CA ASPI e l'entità per cui è stato emesso deve corrispondere all'alias DNS del backend.

Per permettere la mutua autenticazione (MTLS) sul backend deve essere importato il certificato cui ISAM si presenta al backend.

Ogni pagina richiamata dal browser durante la navigazione transita dal webseal.

Le istruzioni per recuperare le stringhe utente di interesse sono del tipo:

String user = request.getHeader("iv-user"); // Restituisce l'utente loggato su ISAM

String groups = request.getHeader("iv-groups"); // Restituisce i gruppi associati all'utente (il separatore è ";")

Si noti che durante la sessione dell'utente possono intervenire:

- Timeout ISAM per inattività (2h)
- Timeout ISAM per durata (12h).

Devono poi essere gestiti dall'applicativo i redirect di logout e di notauth, come specificato di seguito.

Il logout dell'applicazione deve rimandare al logout di ISAM, tramite la pagina[5]:

https://{{AMBIENTEISAM}}/pkmslogout

Nel caso in cui la autorizzazione non sia gestita da ISAM ma dall'applicazione, se l'utente, già autenticato, non è profilato sull'applicazione a cui tenta di accedere, l'applicazione stessa deve rimandare l'utente alla pagina:

https:// {{AMBIENTEISAM}}/notauth.html..

A livello operativo, la richiesta di configurazione di una giunzione deve essere inoltrata tramite ticket alla coda IAM, allegando il template in allegato, opportunamente compilato.

220413 Modulo Richiesta Giunzioni.xls

L'url di accesso alle applicazioni per gli utenti è del tipo[6]

https://www1.gruppo.autostrade.it/{{CODICEAPP}}

dove CODICEAPP è il codice applicazione (come censito in GCS).

[5] Specificare {{AMBIENTEISAM}} come da indicazioni nel paragrafo 3.
[6] Nell'esempio l'ambiente è produzione intranet. Url analoghi sono da considerarsi per ogni ambiente.

La pagina di login del webseal è mostrata in figura.

Saml2.0

L'implementazione del protocollo rispetta lo standard v2.0 (https://www.rfc-editor.org/rfc/rfc7522).

Come raccomandato dalla fondazione OWASP[7], i messaggi Saml2.0 devono transitare tramite canali sicuri (e.g., TLS v1.2 o superiori). In particolare, si richiede che il Resource Server e l'Authentication Server siano muniti di Certificati (e.g., X509).

Si riporta di seguito un saml di esempio.

```
<samlp:Response xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
                xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
                xmlns:xs="http://www.w3.org/2001/XMLSchema"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

Destination="https://account.mysite.com/organizations/5c40d1e9-e979-4e16-
bf88-bb985c438ff4/saml2/login/24e90579-c3a8-4bf6-bdf3-6d7eb68405ab"
                ID="FIMRSP_88e13e6a-0183-1c6f-a55c-b45a6f4d395e"
                IssueInstant="2022-09-29T10:53:26Z"
                Version="2.0"
                >
    <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">https://www2.autostrade.it/XXX/sps/XXX/saml20</saml:Issuer>
    <ds:Signature Id="uuid88e13e6b-0183-1d82-8954-b45a6f4d395e">
        <ds:SignedInfo>
            <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            <ds:SignatureMethod
Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
```

[7] https://cheatsheetseries.owasp.org/cheatsheets/SAML_Security_Cheat_Sheet.html

```
<ds:Reference URI="#FIMRSP_88e13e6a-0183-1c6f-a55c-b45a6f4d395e">
                <ds:Transforms>
                    <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
                    <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#">
                        <xc14n:InclusiveNamespaces
xmlns:xc14n="http://www.w3.org/2001/10/xml-exc-c14n#"
                                                    PrefixList="samlp saml xs
ds xsi"
                                                    />
                    </ds:Transform>
                </ds:Transforms>
                <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />

<ds:DigestValue>nLA6EnqBObpWW8yZufQb8bWtMn1LQfKUtBzpJBIQrLo=</ds:DigestValue
>
            </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>VWgsnds</ds:SignatureValue>
        <ds:KeyInfo>
            <ds:X509Data>
                <ds:X509Certificate>MIIH</ds:X509Certificate>
            </ds:X509Data>
        </ds:KeyInfo>
    </ds:Signature>
    <samlp:Status>
        <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"
/>
    </samlp:Status>
    <saml:Assertion ID="Assertion-uuid88e13e54-0183-16b4-a7f9-b45a6f4d395e"
                    IssueInstant="2022-09-29T10:53:26Z"
                    Version="2.0"
                    >
        <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">https://www2.autostrade.it/XXX/sps/XXX/saml20</saml:Issuer>
        <ds:Signature Id="uuid88e13e56-0183-148e-8b36-b45a6f4d395e">
            <ds:SignedInfo>
                <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
                <ds:SignatureMethod
Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
                <ds:Reference URI="#Assertion-uuid88e13e54-0183-16b4-a7f9-
b45a6f4d395e">
                    <ds:Transforms>
                        <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
                        <ds:Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                            <xc14n:InclusiveNamespaces
xmlns:xc14n="http://www.w3.org/2001/10/xml-exc-c14n#"
                                                        PrefixList="saml xs
xsi"
                                                        />
```

```
</ds:Transform>
                    </ds:Transforms>
                    <ds:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />

<ds:DigestValue>5r45giXQXWPjuFLNqiER5ectzwWFsfxdwQQRlhka7nk=</ds:DigestValue
>
                </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue>DtrNsqTds:SignatureValue>
            <ds:KeyInfo>
                <ds:X509Data>
                    <ds:X509Certificate>MIIH==</ds:X509Certificate>
                </ds:X509Data>
            </ds:KeyInfo>
        </ds:Signature>
        <saml:Subject>
            <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress">mario.rossi@dominio.com</saml:NameID>
            <saml:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                <saml:SubjectConfirmationData NotOnOrAfter="2022-09-
29T10:58:26Z"

Recipient="https://account.mysite.com/organizations/5c40d1e9-e979-4e16-bf88-
bb985c438ff4/saml2/login/24e90579-c3a8-4bf6-bdf3-6d7eb68405ab"
                                                    />
            </saml:SubjectConfirmation>
        </saml:Subject>
        <saml:Conditions NotBefore="2022-09-29T10:48:26Z"
                         NotOnOrAfter="2022-09-29T10:58:26Z"
                         >
            <saml:AudienceRestriction>

<saml:Audience>https://account.mysite.com/organizations/5c40d1e9-e979-4e16-
bf88-bb985c438ff4/saml2</saml:Audience>
            </saml:AudienceRestriction>
        </saml:Conditions>
        <saml:AuthnStatement AuthnInstant="2022-09-29T10:53:26Z"
                             SessionIndex="uuid94db289b-11dc-421e-bacb-
d53c24a8affe"
                             SessionNotOnOrAfter="2022-09-29T11:53:26Z"
                             >
            <saml:AuthnContext>

<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:Password</
saml:AuthnContextClassRef>

<saml:AuthenticatingAuthority>https://intranettst.gruppo.autostrade.it/XXX/s
ps/XXX/</saml:AuthenticatingAuthority>
            </saml:AuthnContext>
        </saml:AuthnStatement>
        <saml:AttributeStatement>
            <saml:Attribute Name="givenname"
```

```
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                              >
                <saml:AttributeValue
xsi:type="xs:string">Mario</saml:AttributeValue>
            </saml:Attribute>
            <saml:Attribute Name="emailaddress"

NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                              >
                <saml:AttributeValue
xsi:type="xs:string">utente.prova@austostrade.it</saml:AttributeValue>
            </saml:Attribute>
            <saml:Attribute Name="surname"

NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                              >
                <saml:AttributeValue
xsi:type="xs:string">Rossi</saml:AttributeValue>
            </saml:Attribute>
        </saml:AttributeStatement>
    </saml:Assertion>
</samlp:Response>
```

In caso di customizzazioni è necessario concordare e specificare gli attributi utente opzionali da inserire.

A livello operativo, è necessario che Identity Provider e Service Provider forniscano al partner il proprio metadata saml. Una volta importati su entrambi i provide si potrà procedere con i test di accesso utente.

In occasione dei rinnovi dei certificati, bisogna pianificare, in accordo con i referenti applicativi e con congruo anticipo, una sessione congiunta per rendere operativo il change minimizzando il disservizio.

Oauth2.0

Come specificato nel RFC 6749[8], i messaggi OAuth2.0 devono transitare tramite canali sicuri (e.g., TLS v1.2 o superiori). In particolare, in terminologia OAuth2.0, si richiede che il Service Provider e l'Identity Provider siano muniti di Certificati (e.g., X509).

A livello operativo, è sempre necessario richiedere al team IAM la configurazione dei parametri client_id (o client_id e client_secret) per l'applicativo in oggetto.

Di norma, la nomenclatura prevede per il client_id il richiamo ai codici applicazione sorgente e destinazione.

[8] https://www.rfc-editor.org/rfc/rfc6749

Si riportano di seguito gli endpoint specifici di ogni ambiente

TEST INTRANET

- Token: https://intranettst.gruppo.autostrade.it/mga/sps/oauth/oauth20/token
- Introspect: https://intranettst.gruppo.autostrade.it/mga/sps/oauth/oauth20/introspect
- Authorize: https://intranettst.gruppo.autostrade.it/mga/sps/oauth/oauth20/authorize
- Metadata: https://intranettst.gruppo.autostrade.it/mga/sps/oauth/oauth20/metadata/{{CODICEAPP}}
- JKWS: https://intranettst.gruppo.autostrade.it/mga/sps/oauth/oauth20/jwks/{{CODICEAPP}}

TEST DMZ

- Token: https://dmztst.autostrade.it/mga/sps/oauth/oauth20/token
- Introspect: https://dmztst.autostrade.it/mga/sps/oauth/oauth20/introspect
- Authorize: https://dmztst.autostrade.it/mga/sps/oauth/oauth20/authorize
- Metadata: https://dmztst.autostrade.it/mga/sps/oauth/oauth20/metadata/{{CODICEAPP}}
- JKWS: https://dmztst.autostrade.it/mga/sps/oauth/oauth20/jwks/{{CODICEAPP}}

QA INTRANET

- Token: https://qa.gruppo.autostrade.it/mga/sps/oauth/oauth20/token
- Introspect: https://qa.gruppo.autostrade.it/mga/sps/oauth/oauth20/introspect
- Authorize: https://qa.gruppo.autostrade.it/mga/sps/oauth/oauth20/authorize
- Metadata: https://qa.gruppo.autostrade.it/mga/sps/oauth/oauth20/metadata/{{CODICEAPP}}
- JKWS: https://qa.gruppo.autostrade.it/mga/sps/oauth/oauth20/jwks/{{CODICEAPP}}

PROD INTRANET

- Token: https://www1.gruppo.autostrade.it/mga/sps/oauth/oauth20/token
- Introspect: https://www1.gruppo.autostrade.it/mga/sps/oauth/oauth20/introspect
- Authorize: https://www1.gruppo.autostrade.it/mga/sps/oauth/oauth20/authorize
- Metadata: https://www1.gruppo.autostrade.it/mga/sps/oauth/oauth20/metadata/{{CODICEAPP}}
- JKWS: https://www1.gruppo.autostrade.it/mga/sps/oauth/oauth20/jwks/{{CODICEAPP}}

PROD INTERNET

- Token: https://www2.autostrade.it/mga/sps/oauth/oauth20/token
- Introspect: https://www2.autostrade.it/mga/sps/oauth/oauth20/introspect
- Authorize: https://www2.autostrade.it/mga/sps/oauth/oauth20/authorize
- Metadata: https://www2.autostrade.it/mga/sps/oauth/oauth20/metadata/{{CODICEAPP}}
- JKWS: https://www2.autostrade.it/mga/sps/oauth/oauth20/jwks/{{CODICEAPP}}

Si descrivono nel dettaglio i flussi di interesse.

1. Client credentials (machine-to-machine)

I client che rientrano in questa casistica devono selezionare il Grant Type "Client Credentials" del protocollo OAuth2 (https://tools.ietf.org/html/rfc6749#section-4.4).

L'endpoint[9] per la richiesta dei token è raggiungibile con:

https:// {{AMBIENTEISAM}}/mga/sps/oauth/oauth20/token

con i parametri

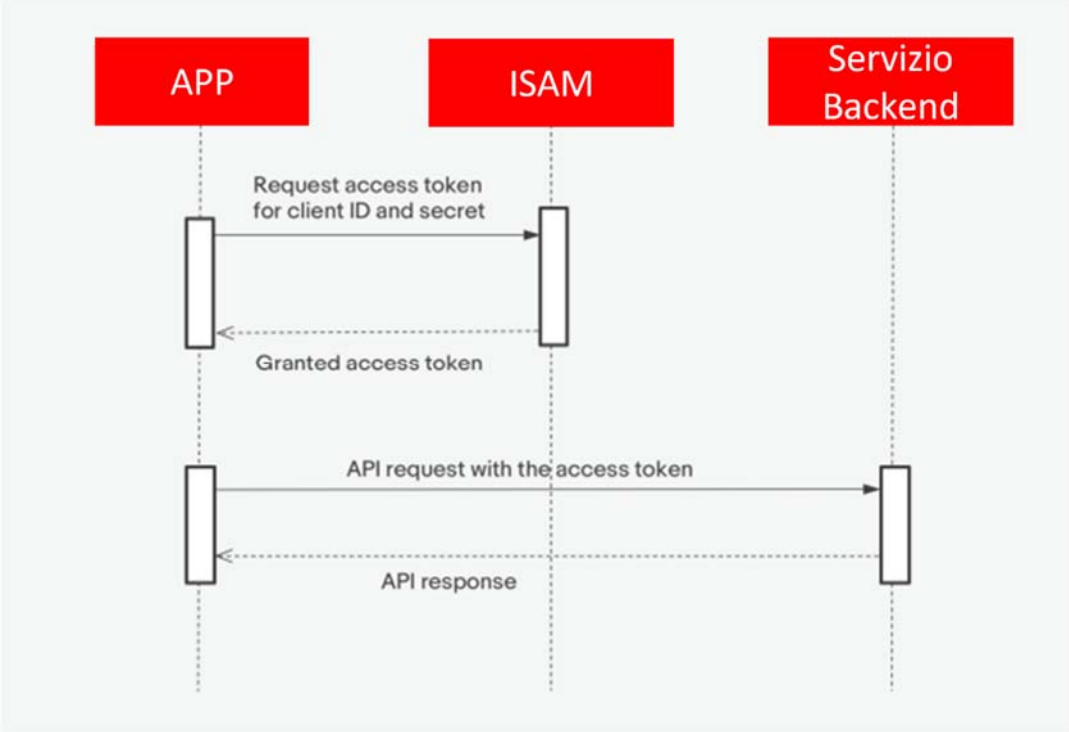
client_id=*xxxxx*

grant_type=client_credentials

Il dettaglio dei passi relativi a questo Grant Type è riportato qui di seguito:

- Il client effettua la richiesta dell'access token all'authorization server tramite il grant_type "client_credentials". Invia per l'autenticazione i propri client_id e client_secret.
- L'authorization server autentica il client e, in caso di esito positivo, restituisce l'access token (o il jwt) al client.
- Il client invoca l'operazione protetta utilizzando l'access token ottenuto al passo precedente.
- Il server (o il gateway in caso di interfaccia tramite api gateway) verifica la validità del token invocando l'endpoint Introspect esposto dall'authorization server (o verificando la firma del jwt)
- Il Resorce Owner elabora la richiesta, eventualmente effettuando ulteriori validazioni dei dati del token.
- La risposta dell'applicazione viene inviata al client.

[9] Analogo endpoint definito per ogni ambiente isam



È possibile - e preferibile - richiedere ad ISAM un token JWT, in luogo di un access token, specificando l'attributo scope=openid, in modo da ottenere un token firmato e sostituire la chiamata introspect con la verifica della firma del token (che non richiede interrogazione di ISAM).

Salvo diverse specifiche, il parametro audience è pari al client-id.

Di seguito, alcune chiamate di esempio.

- **CHIAMATA D'ESEMPIO PER OTTENERE ACCESS TOKEN**

```
curl --location --request POST 'https://dmztst.autostrade.it/mga/sps/oauth/oauth20/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'client_id={{CLIENT_ID}}' \
--data-urlencode 'client_secret={{SECRET}}' \
```

- **RISPOSTA D'ESEMPIO**

HTTP STATUS: 200 OK

Body:


```
{
    "access_token": "fC92uDwHYt7SZOhDlkNr",
    "scope": "",
    "token_type": "bearer",
    "expires_in": 3599
}
```


- **CHIAMATA D'ESEMPIO PER OTTENERE ACCESS TOKEN + JWT**

```
curl --location --request POST 'https://dmztst.autostrade.it/mga/sps/oauth/oauth20/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'client_id={{CLIENT_ID}}' \
--data-urlencode 'client_secret={{SECRET}}' \
--data-urlencode 'scope=openid'
```

- **RISPOSTA D'ESEMPIO**


```
{
    "access_token": "ye1iJa4nOCaCR49o1ur8",
    "scope": "openid",
    "id_token": "YEGa0vpkg5Cww",
    "token_type": "bearer",
    "expires_in": 3599
}
```

- **CHIAMATA D'ESEMPIO PER INTROSPECT DI UN ACCESS TOKEN**

```
curl --location --request POST
'https://dmztst.autostrade.it/mga/sps/oauth/oauth20/introspect' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'token={{ACCESSTOKEN_DA_VERIFICARE}}' \
--data-urlencode 'client_id= ={{CLIENT_ID}}' \
--data-urlencode 'secret==={{SECRET}}'
```

- **RISPOSTA D'ESEMPIO**

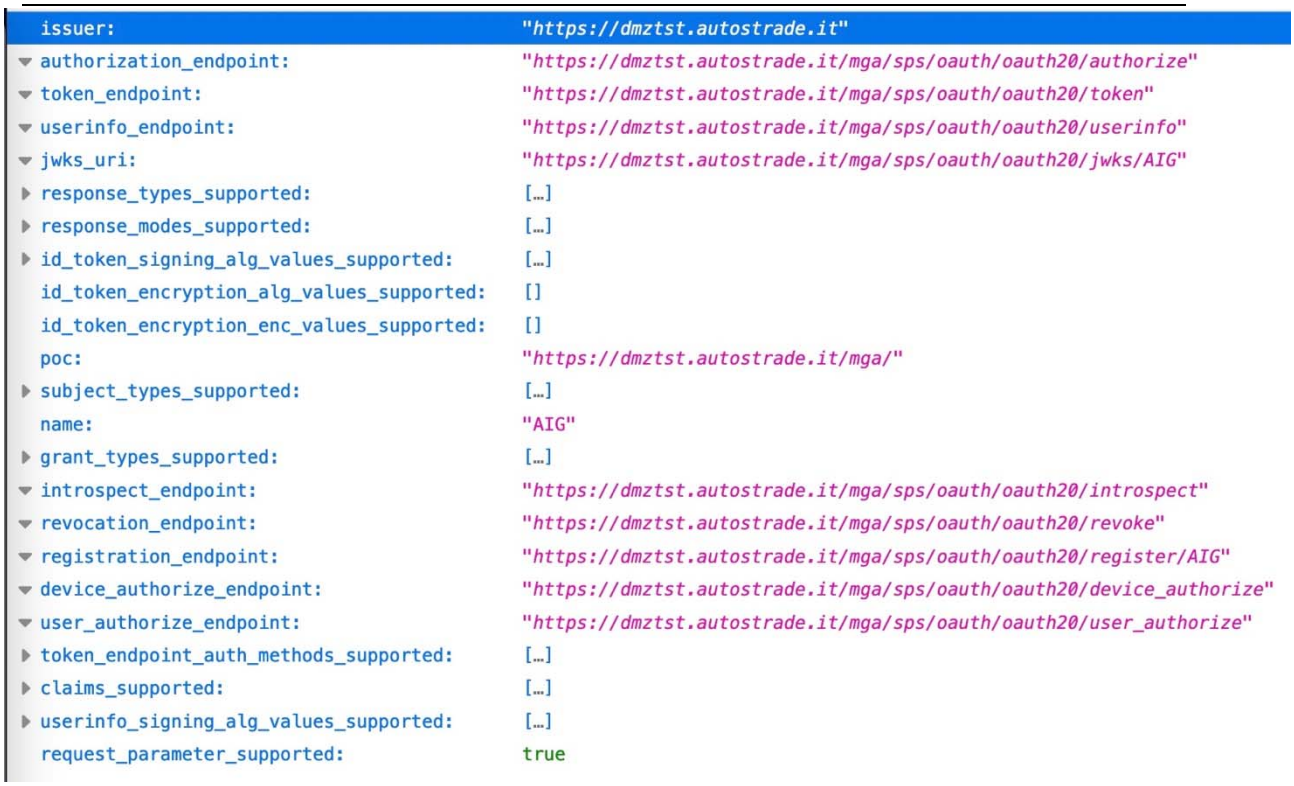
```
{
    "scope": "openid",
    "active": true,
    "token_type": "bearer",
    "exp": 1666111362,
    "iat": 1666107762,
    "client_id": "{{CLIENT_ID}}",
    "username": "{{CLIENT_ID}}"
}
```

2. **Authorization code PKCE**

I client che rientrano in questa casistica devono selezionare il Grant Type "Authorization Code (PKCE)" del protocollo OAuth2 (https://tools.ietf.org/html/rfc7636).

Una volta censita l'applicazione, verrà fornito l'url del metadata OAuth che contiene tutti gli endpoint ed i certificati necessari.

In Figura, metadata di esempio.



issuer:	"https://dmztst.autostrade.it"
authorization_endpoint:	"https://dmztst.autostrade.it/mga/sps/oauth/oauth20/authorize"
token_endpoint:	"https://dmztst.autostrade.it/mga/sps/oauth/oauth20/token"
userinfo_endpoint:	"https://dmztst.autostrade.it/mga/sps/oauth/oauth20/userinfo"
jwks_uri:	"https://dmztst.autostrade.it/mga/sps/oauth/oauth20/jwks/AIG"
response_types_supported:	[…]
response_modes_supported:	[…]
id_token_signing_alg_values_supported:	[…]
id_token_encryption_alg_values_supported:	[]
id_token_encryption_enc_values_supported:	[]
poc:	"https://dmztst.autostrade.it/mga/"
subject_types_supported:	[…]
name:	"AIG"
grant_types_supported:	[…]
introspect_endpoint:	"https://dmztst.autostrade.it/mga/sps/oauth/oauth20/introspect"
revocation_endpoint:	"https://dmztst.autostrade.it/mga/sps/oauth/oauth20/revoke"
registration_endpoint:	"https://dmztst.autostrade.it/mga/sps/oauth/oauth20/register/AIG"
device_authorize_endpoint:	"https://dmztst.autostrade.it/mga/sps/oauth/oauth20/device_authorize"
user_authorize_endpoint:	"https://dmztst.autostrade.it/mga/sps/oauth/oauth20/user_authorize"
token_endpoint_auth_methods_supported:	[…]
claims_supported:	[…]
userinfo_signing_alg_values_supported:	[…]
request_parameter_supported:	true

Il dettaglio dei passi relativi a questo Grant Type è riportato qui di seguito.

• Il Client, seguendo le modalità indicate nella RFC-7636 sez. 4.1, genera la chiave Code Verifier. Quindi, in accordo a quanto indicato nella RFC-7636 sez. 4.2, genera la chiave Code Challenge, utilizzando come metodo di hashing "SHA256".

• Il Client richiede l'authorization code all'Authorization Server inviando la chiave Code Challenge tra i parametri della richiesta. Il “code challenge_method” da indicare è quello riferito al punto precedente e quindi si deve utilizzare il valore "S256".

• Superata la validazione, l'Authorization Server redirige il client all'IdP per l'autenticazione.

• L’utente inserisce le proprie credenziali per effettuare l'autenticazione.

• Il Client esegue l'operazione di autenticazione sull'IdP, ottenendo in risposta l'Authorization Code.

• Il Client effettua la richiesta dell'access token all'Authorization Server, con grant type "authorization_code", inviando la chiave Code Verifier tra i parametri della richiesta.

• L'Authorization Server usa il Code Challenge, precedentemente ricevuto, per validare il Code Verifier. Superata la validazione, restituisce l'access token al Client.

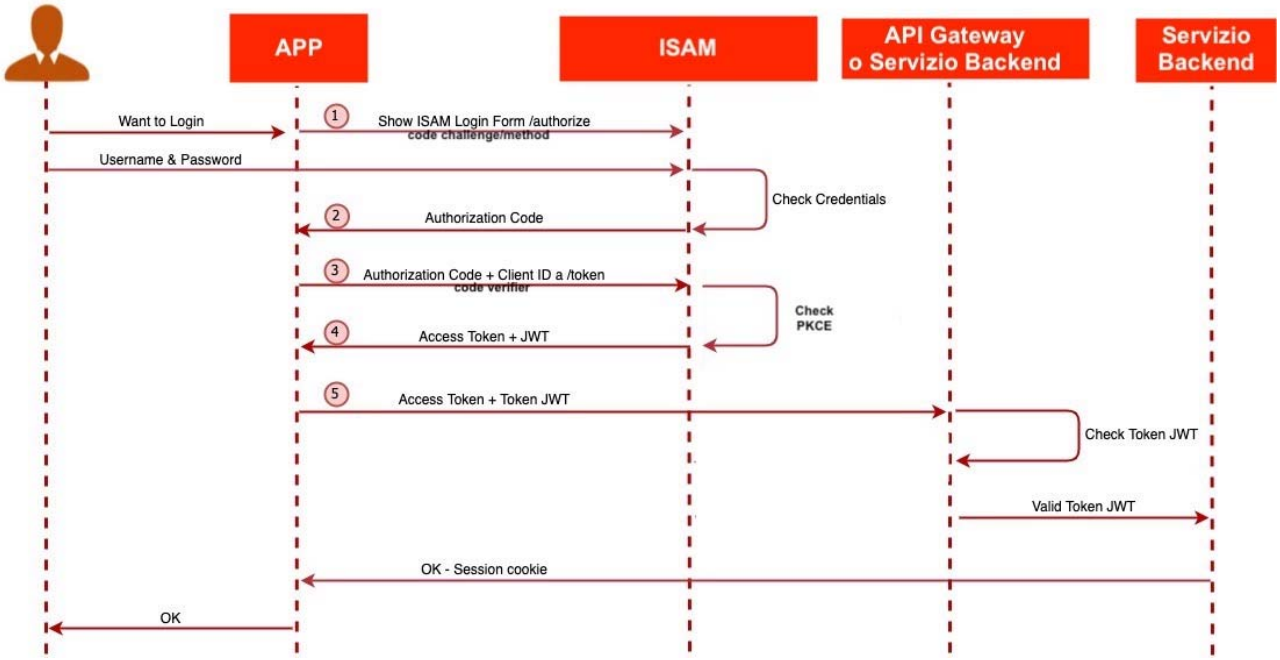
• Il Client invoca l'operazione protetta sul Servizio backend (o sull'API Gateway) utilizzando l'access token ottenuto al passo precedente.

• L'API Gateway inoltra la richiesta all'applicazione che eroga il servizio, inoltrando i dati di autorizzazione (o il token originale) per gli aspetti di validazione che competono l'erogatore.

• La risposta dell'applicazione viene inviata all'API Gateway e da quest'ultimo al client.

In caso di scadenza dell'access token, il Client ne può ottenere uno nuovo effettuando una nuova richiesta all'Authorization Server mediante l'authorization code e il Code Verifier già utilizzato precedentemente.

Si riporta in Figura lo schema delle interazioni nel caso di Client OAuth di tipo “public”.



Il flusso descritto viene implementato con le chiamate riportate di seguito.

La chiamata che dà inizio al flusso authorization_code è così composta:

https://dmztst.autostrade.it/mga/sps/oauth/oauth20/authorize?response_type=code&state=2020&client_id={{CLIENT_ID}}&scope=openid&redirect_uri=https%3A%2F%2Foauth.pstmn.io%2Fv1%2Fcallback&

code_challenge={{CODE_CHALLENGE}}&code_challenge_method=S256

Oltre a client_id, code_challenge. code_challenge_method, nella chiamata è specificato uno "state".

Questo parametro sarà poi presentato al client per permettere di accoppiare l'authorization code ricevuto e la richiesta di autenticazione effettuata.

Questo parametro deve cambiare per ogni chiamata.

L'utente visualizzerà la consueta pagina di login isam.

Una volta fatto il login, l'utente riceverà un redirect http 302 verso l'endpoint dell'app client con authorization code e state in query string:

https://{{APP-URL}}/v1/callback?**state=2020**&code=m7xUa79kPK6hw8sjcWBSjuA74n5ANS

A questo punto l'applicazione usa l'authorization code ricevuto, in associazione con client_id, client_secret e code_verifier, per farsi staccare da isam un access token, un refresh token ed un jwt.

La chiamata è la seguente:

```
POST /mga/sps/oauth/oauth20/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: PostmanRuntime/7.26.5
Accept: */*
Cache-Control: no-cache
Postman-Token: 1030eb21-56fd-4b1d-8076-93e19fc44cfb
Host: {{AMBIENTEISAM}}
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 165

grant_type=authorization_code&code=m7xUa79kPK6hw8sjcWBSjuA74n5ANS&redirect_uri=https%3A%2
F%2Foauth.pstmn.io%2Fv1%2Fcallback&client_id={{CLIENT_ID}}&client_secret={{CLIENT_SECRET}
}&code_verifier={{code_verifier}}

HTTP/1.1 200 OK
```

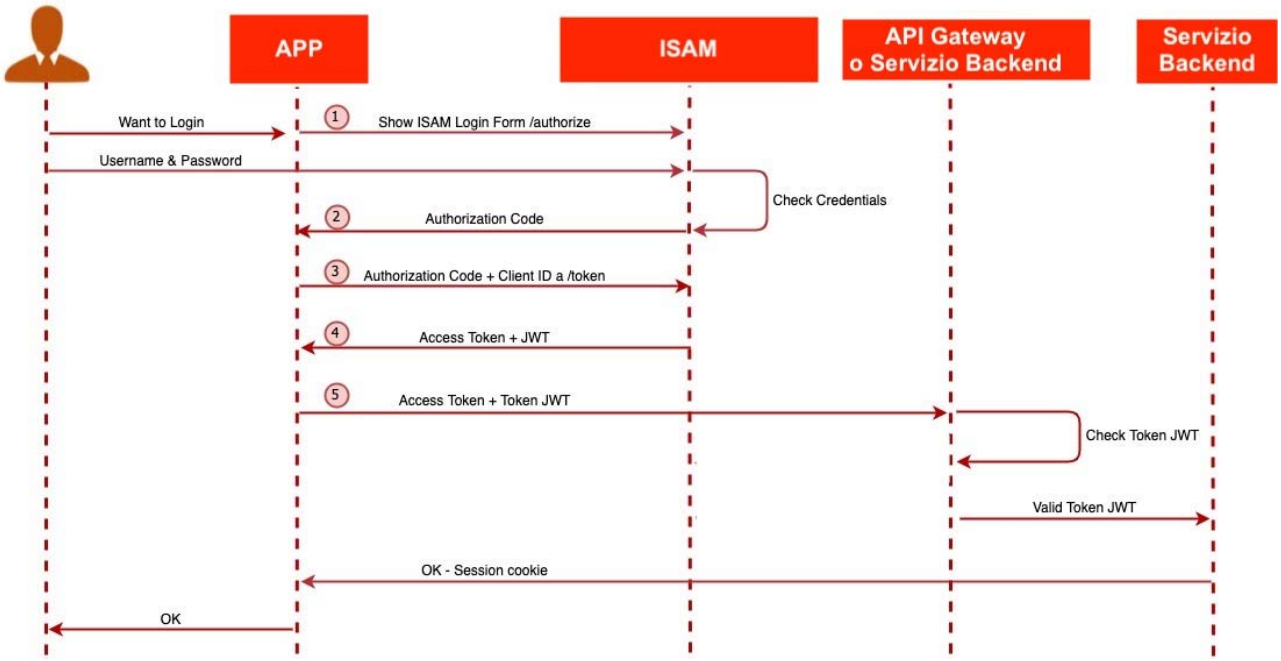
Risposta di esempio:

{"**access_token**":"M2JUSPJtLy7InRwZrg9pK3bmL","**refresh_token**":"E8AgUwtVYlOhaQ0EsLnfAsclNEjLx9tPydkyS8av","scope":"openid","**id_token**":"eyJraWQiOiJoZ0lPNGpZWDFENGctRU5WRUE4c2l4QURxSjV2N1RmZVRfRHo1cW8tNnFnIiwiYWxnIjoiUlMyNTYifQ.eyJydF9oYXNoIjoia0RNeFlUZ1NSUVMyYkU5Z1VvRWVwQSIsImNvZGljZWZpc2NhbGUiOiJTVFJQUUw2NVQyNkY4MzlPIiwicHJvdG9jb2xsbyI6IjEwMDEiLCJpYXQiOjE2MDUxMDUyMzgsImlzcyI6Imh0dHBzOi8vd2VidG9rZW50ZXN0Lm5vdGFyaWF0by5pdC8iLCJhdF9oYXNoIjoiV1JoVkR ","token_type":"bearer","expires_in":3599}

L'app potrà utilizzare l'access token o il jwt ricevuto per accedere alle risorse protette.

3. Authorization code

Nel caso "confidential client" il flusso è semplificato come riportato in figura.



Il flusso descritto viene implementato con le chiamate riportate di seguito.

La chiamata che dà inizio al flusso authorization_code è così composta:

https://dmztst.autostrade.it/mga/sps/oauth/oauth20/authorize?response_type=code&state=2020&client_id={{CLIENT_ID}}&scope=openid&redirect_uri=https%3A%2F%2Foauth.pstmn.io%2Fv1%2Fcallback

Nella chiamata è specificato il parametro "state".
Questo parametro sarà poi presentato al client per permettere di accoppiare l'authorization code ricevuto e la richiesta di autenticazione effettuata.

Questo parametro deve cambiare per ogni chiamata.

L'utente visualizzerà la pagina di login isam.

Una volta fatto il login, l'utente riceverà un redirect http 302 verso l'endpoint dell'app client con authorization code e state in query string:

https://{{APP-URL}}/v1/callback?**state=2020**&code=m7xUa79kPK6hw8sjcWBSjuA74n5ANS

A questo punto l'applicazione usa l'authorization code ricevuto, in associazione con client_id e client_secret, per farsi staccare da isam un access token, un refresh token ed un jwt.

La chiamata è la seguente:

```
POST /mga/sps/oauth/oauth20/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: PostmanRuntime/7.26.5
Accept: */*
Cache-Control: no-cache
Postman-Token: 1030eb21-56fd-4b1d-8076-93e19fc44cfb
Host: {{AMBIENTEISAM}}
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 165

grant_type=authorization_code&code=m7xUa79kPK6hw8sjcWBSjuA74n5ANS&redirect_uri=https%3A%2
F%2Foauth.pstmn.io%2Fv1%2Fcallback&client_id={{CLIENT_ID}}&client_secret={{CLIENT_SECRET}
}

HTTP/1.1 200 OK
```

Risposta di esempio:

{"**access_token**":"M2JUSPJtLy7InRwZrg9pK3bmL","**refresh_token**":"E8AgUwtVYlOhaQ0EsLnfAsclNEjLx9tPydkyS8av","scope":"openid","**id_token**":"eyJraWQiOiJoZ0lPNGpZWDFENGctRU5WRUE4c2l4QURxSjV2N1RmZVRfRHo1cW8tNnFnIiwiYWxnIjoiUlMyNTYifQ.eyJydF9oYXNoIjoia0RNeFlUZ1NSUVMyYkU5Z1VvRWVwQSIsImNvZGljZWZpc2NhbGUiOiJTVFJQUUw2NVQyNkY4MzlPIiwicHJvdG9jb2xsbyI6IjEwMDEiLCJpYXQiOjE2MDUxMDUyMzgsImlzcyI6Imh0dHBzOi8vd2VidG9rZW50ZXN0Lm5vdGFyaWF0by5pdC8iLCJhdF9oYXNoIjoiV1JoVkR "},"token_type":"bearer","expires_in":3599}

L'app potrà utilizzare l'access token o il jwt ricevuto per accedere alle risorse protette.

Si noti che di default nel JWT sarà presente la matricola dell'utente ed il client_id dell'applicazione.

È possibile modificare i claims del JWT per inserire altri attributi utente, come ad esempio:

- Nome
- Cognome
- Mail
- Sede
- Tronco
- Società
- Gruppi LDAP associati all'utente.

Nei casi Authorization code, con o senza PKCE, si può applicare una acl al client-id, per permettere solo ad utenti in un determinato gruppo di ottenere il relativo token.

La configurazione deve essere esplicitamente richiesta.

Si osservi infine la gestione di timeout e refresh token (authorization flow) descritta di seguito.

ISAM rilascia un access token e un refresh token.

L'access token ha durata di 1 ora, ma viene automaticamente rinnovato presentando un refresh token valido.

Il refresh token a sua volta ha validità di 12 ore.

Quando l'app tenta l'accesso ad una risorsa protetta, presenta l'access token:

- Se l'access token è valido, l'accesso è permesso
- Se l'access token non è valido, viene richiesto un nuovo access token presentando un refresh token.

A questo punto:

- Se il refresh token è valido, viene rilasciato (in modo trasparente all'utente) un nuovo access token con cui avere accesso alla risorsa protetta.
- Se il refresh token non è valido, l'utente deve ripetere l'autenticazione, al fine di ottenere nuovamente dei token validi.

4. MULTI FACTOR AUTHENTICATION

Il requisito di MFA sugli accessi è espresso da CISO a livello di singola applicazione, in base alla classificazione business critical, alla tipologia di dati trattati, al criterio di esposizione su internet.

L'enforcement con MFA viene quindi applicato secondo tali indicazioni e di default sulle applicazioni esposte in internet, salvo diverse considerazioni.

L'MFA viene gestito da ISAM per le applicazioni protette da ISAM stesso.

È possibile gestire la distribuzione dell'OTP via mail, via sms o tramite push notification, secondo quanto convenuto per le singole applicazioni.

La protezione con MFA è trasparente per gli applicativi protetti con ISAM.