SPD-2021-2022

Servizi di supporto alle Piattaforme Digitali. Anni 2021 - 2022

01.01 - Assistenza, supporto applicativo, manutenzione e gestione dell'applicativo

D01.01.03 – Manuale di integrazione Access Manager

Regole tecniche e modalità da seguire per l'interfacciamento di un generico sistema informativo regionale con il componente Access Manager (AM SPID-RAS).

Data documento:	22.11.2022
File:	SPD-2021-2022_D01.01.03 - Manuale di integrazione Access Manager_v07.00.docx
Versione:	07.00
Redazione:	Sardegna IT
Validazione tecnica e rilascio:	Sardegna IT
Approvazione:	**Regione Autonoma della Sardegna** Assessorato degli Affari generali, personale e riforma della Regione Direzione generale degli Affari generali e società dell'informazione Servizio dei sistemi informativi regionali e degli enti

SPD-2021-2022

Servizi di supporto alle Piattaforme Digitali. Anni 2021 - 2022

01.01 - Assistenza, supporto applicativo, manutenzione e gestione dell'applicativo

D01.01.03 – Manuale di integrazione Access Manager

Autori

Autori del documento	*Struttura di appartenenza*	*Parte del documento*	*Note*
Giorgio Follesa	Sardegna IT	Tutte	Redazione
Nicola Bulla	Sardegna IT	Capitolo 4.4	Redazione
Antonio Pes	Sardegna IT	Tutte	Revisione
Francesco Micheli	Sardegna IT	Tutte	Validazione tecnica per rilascio a RAS
Monica Faedda	Sardegna IT	Tutte	Rilascio a RAS

Storia delle modifiche

Vers.Rev.	*Data*	*Autore/i*	*Descrizione modifiche*
00.01	17/01/2017	Giorgio Follesa	Prima impostazione struttura e scrittura paragrafi
00.02	31/01/2017	Nicola Bulla	Integrazione paragrafo integrazione OAuth2
00.03	31/01/2017	Antonio Pes	Revisione
00.04	13/02/2017	Francesco Micheli	Revisione
00.05	23/02/2017	Nicola Bulla	Revisione integrazione OAuth2
00.06	28.02.2017	Giorgio Follesa	Revisione Integrazione SAML
01.00	01.03.2017	Francesco Micheli	Revisione e rilascio
02.00	06.03.2020	Francesco Micheli	Inserimento Capitolo 5 e nuovo rilascio
03.00	31.08.2020	Giorgio Follesa	Rivisto il paragrafo sull'autenticazione con OAuth2, aggiunti paragrafi appendice con esempio OAuth2 e corrette le liste numeriche
04.00	16.10.2020	Giorgio Follesa	Aggiunte informazioni sugli attributi di contesto
05.00	21.01.2021	Giorgio Follesa	Aggiunte informazioni su CIE e nuovi set di attributi (si vedano i paragrafi 4.3.1 e 4.3.2 nell'attuale versione 05.00)
05.01	25.02.2022	Antonio Pes Giorgio Follesa	Inserimento paragrafo 1.1.1 Aggiornamento Tabella 6 Aggiornamento Tabella 8
05.02	21.04.2022	Giorgio Follesa	Aggiunta paragrafi 2.1.3 e 4.3.3 per autenticazione EIDAS Aggiunti i nuovi attributi riguardanti il domicilio fisico dell'utente
		Francesco Micheli	Validazione per rilascio a RAS
06.00	26.04.2022	Monica Faedda	Rilascio a RAS Aggiornamento Codice deliverable
07.00	22.11.2022	Monica Faedda	Rilascio a RAS Aggiornamento Tabella 6

SPD-2021-2022

Servizi di supporto alle Piattaforme Digitali. Anni 2021 - 2022

01.01 - Assistenza, supporto applicativo, manutenzione e gestione dell'applicativo

D01.01.03 – Manuale di integrazione Access Manager

Acronimi e definizioni

Acronimi	*Definizioni*
AC	Access Manager
AgID	Agenzia per l'Italia Digitale (D.L. n. 83/2012)
CAD	Codice dell'Amministrazione Digitale (D.L.vo 7 marzo 2005, n. 82)
CIE	Carta d'identità elettronica
D.L.vo / D.lgs	Decreto Legislativo
D.P.C.M.	Decreto Presidente del Consiglio dei Ministri
eIDAS	electronic IDentification Authentication and Signature (Regolamento UE n° 910/2014)
IdM-RAS	(Sistema di) Identity Management delle Regione Sardegna
IdP	Identity Provider
OAuth	Open standard for Authorization
PA/PPAA	Pubblica Aministrazione/Pubbliche Amministrazioni
SAML	Security Assertion Markup Language
SP	Service Provider
SPID	Sistema Pubblico di Identità Digitale (D.P.C.M. 24 ottobre 2014)
SSO	Single Sign On
TS-CNS	Tessera Sanitaria – Carta Nazionale dei Servizi

Sommario

1 Introduzione 5
1.1 Scopo del documento 5
1.1.1 Note esplicative sulla versione 06.00 del documento 5
1.2 Riferimenti 5
1.2.1 Normativa 5
1.2.2 Avvisi 5
2 Documenti tecnici di Linee guida tecniche per l'attuazione dello SPID 6
2.1 Infrastruttura SPID 6
2.1.1 Gestori dell'identità digitale 6
2.1.2 Integrazione CIE 7
2.1.3 Integrazione EIDAS 7
2.1.4 Fornitori di servizi 7
2.1.5 Gestori degli attributi qualificati 7
3 Access Manager 7
3.1 Definizione 7
3.2 Architettura Access Manager 10
4 Integrazione di un Servizio RAS 10
4.1 Autenticazione 11
4.2 Single Logout 16

4.3 Integrazione SAML 18
4.3.1 Attributi utente 18
4.3.2 Attributi CIE 23
4.3.3 Attributi EIDAS 23
4.3.4 Attributi di contesto 24
4.3.5 Integrazione servizio Java con Spring (3.x o 4.x) 24
4.3.6 Migrazione servizi da IdM-RAS a Access Manager 25
4.4 Integrazione OAuth2 27
4.4.1 Attributi utente 27
4.4.2 Configurazione dei servizi 27
5 Procedura amministrativa 28
6 Appendice 32
6.1 XML Spring Security 32
6.2 Esempio chiamate OAuth2 41
6.2.1 Login 41
6.2.2 Logout 44

Indice delle tabelle

Tabella 1 – Accesso utente generico servizio RAS 12
Tabella 2 – Flusso con autenticazione SAML 14
Tabella 3 – Flusso con autenticazione OAuth 16
Tabella 4 – Flusso Single Logout 17
Tabella 5 - Elenco attributi utente (base) 19
Tabella 6 - Set attributi del Metadata RAS 23
Tabella 7 - Tabella di conversione attributi 26
Tabella 8 - Sinossi della procedura Tecnica e Amministrativa 32

Indice delle figure

Figura 1 - Infrastruttura SPID 6
Figura 2 – Implementazione Access Management, situazione transitoria 8
Figura 3 - Implementazione Access Management, situazione a regime 9
Figura 4 - Architettura generale del sistema e interfacce esposte 10
Figura 5 – Funzionamento autenticazione 11
Figura 6 – Funzionamento autenticazione con integrazione SAML 13
Figura 7 – Funzionamento autenticazione con integrazioneOAuth 15
Figura 8 – Funzionamento Single Logout 17

SPD-2021-2022

Servizi di supporto alle Piattaforme Digitali. Anni 2021 - 2022

01.01 - Assistenza, supporto applicativo, manutenzione e gestione dell'applicativo

D01.01.03 – Manuale di integrazione Access Manager

1 Introduzione

1.1 Scopo del documento

Il presente documento ha la finalità di illustrare il funzionamento della infrastruttura SPID e del sistema di *middleware* della Regione Autonoma Sardegna (d'ora in poi *Access Manager* o AC o AM) e definire le modalità l'interfacciamento con i sistemi informativi regionali (Service Provider).

1.1.1 Note esplicative sulla versione 06.00 del documento

Il presente documento, per le parti specificate nella "Storia delle modifiche", aggiorna quello che era stato precedentemente rilasciato nell'ambito del progetto SPD-2020 con il seguente codice deliverable:

D01.02.05.A - Manuale di integrazione Access Manager_v05.00.

Con la finalità di registrare le successive revisioni che si sono rese necessarie per l'aggiornamento/evoluzione del sistema AM, al documento viene attribuito progressivo di versione 06.00, ma con diverso codice deliverable D01.01.03 associato alla linea di attività 01.01 prevista nell'incarico SPD-2021-2022.

1.2 Riferimenti[1]

1.2.1 Normativa

- Decreto della Presidenza del Consiglio dei Ministri 24 ottobre 2014
- Regolamento SPID: regole tecniche
- Determinazione AGID pubblicazioni AVVISI SPID

1.2.2 Avvisi

- Avviso n. 1 - Gestione della sicurezza del canale di trasmissione;
- Avviso n. 2 - Collaborazione tra IDP e SP per problematiche di accesso a SPID;
- Avviso n. 3 - Chiarimenti sulla gestione delle sessioni SSO e meccanismo di Single logout;
- Avviso n. 4 - Livelli di servizio minimo per funzionalità omogenee;
- Avviso n. 5 - Regolamento recante le Regole Tecniche - Errata Corrige;
- Avviso n. 6 - Note sul dispiegamento di SPID presso i gestori dei servizi;
- Avviso n. 01-2018 – Note per il dispiegamento del Login EIDAS presso le Pubbliche Amministrazioni
- Avviso n. 02-2018 – Note per il dispiegamento del Login EIDAS presso gli IDP SPID.

[1] Si veda la documentazione pubblicata su http://www.agid.gov.it/agenda-digitale/infrastrutture-architetture/spid

2 Documenti tecnici di Linee guida tecniche per l'attuazione dello SPID

2.1 Infrastruttura SPID

Il funzionamento dello SPID si basa sulla relazione di "fiducia" garantita all'interno della federazione tramite l'intermediazione dell'Agenzia per l'Italia Digitale (AgID) che gestisce l'accreditamento dei gestori dell'identità digitale (Identity Provider), dei fornitori di servizi (Service Provider) e dei gestori degli attributi qualificati.

L'AgID gestisce inoltre il Registro SPID (definito nel DPCM 24 ottobre 2014, specificato nei contenuti e regole di iscrizione nel Modello di Convenzione, richiamato per le parti tecniche all'interno delle linee guida) nel quale vengono pubblicate tutte le entità aderenti al patto di fiducia e pertanto abilitate a operare nell'ambito della federazione.

Le modalità di funzionamento delle entità SPID sono basate sullo standard SAML v2.0.

La figura seguente illustra in sintesi il funzionamento dell'infrastruttura SPID.

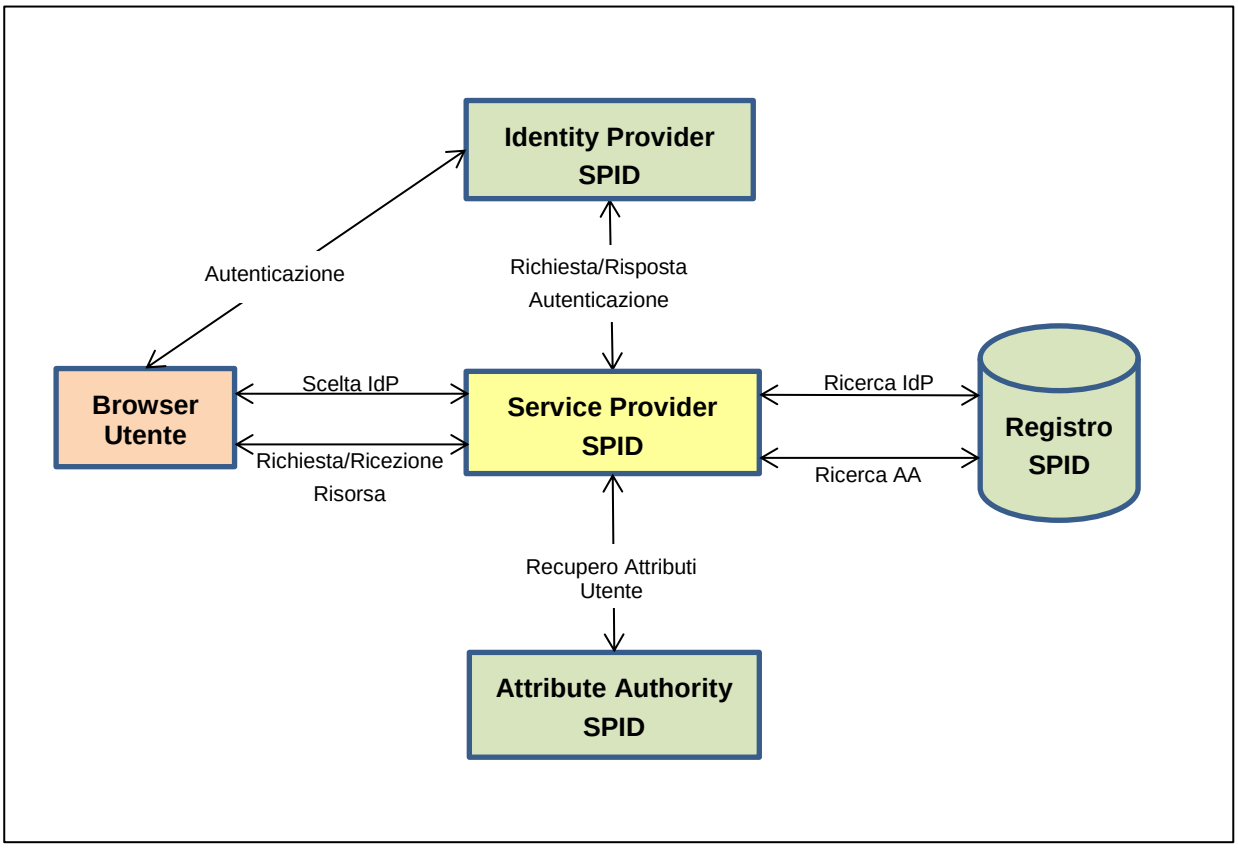


Figura 1 - Infrastruttura SPID.

Di seguito si forniscono gli elementi principali che descrivono le entità dell'infrastruttura (DPCM 24 ottobre 2014, art. 1 – Definizioni).

2.1.1 Gestori dell'identità digitale

I Gestori dell'identità digitale (Identity Provider - IdP) forniscono il servizio di autenticazione mediante la gestione dell'identità digitale e degli attributi dell'utente (identificativi e secondari, per la cui migliore definizione si fa rinvio alla Tabella attributi pubblicata sul portale AgID http://www.agid.gov.it/agenda-digitale/infrastrutture-architetture/spid).

2.1.2 Integrazione CIE

A partire dal 2021 è stata resa disponibile su Access manager la possibilità di autenticazione mediante CIE (carta di identità elettronica). L'identity provider che permette l'utilizzo del documento elettronico viene gestito ed erogato dal Ministero dell'interno (https://www.cartaidentita.interno.gov.it/) e utilizza il protocollo SAML[2]. Tutte le interazioni fra access manager e l'identity provider CIE sono identiche a quelle previste per SPID riportate in seguito. L'autenticazione mediante CIE verrà abilitata su ogni singolo servizio a seguito di verifiche di corretto funzionamento in quanto il gruppo di attributi restituito è un sotto insieme di quello SPID.

2.1.3 Integrazione EIDAS

Nel 2021 è stata resa disponibile su Access manager la possibilità di autenticazione mediante identità eIDAS. Il gateway nazionale (https://www.agid.gov.it/it/piattaforme/nodo-eidas-italiano) , tramite il pulsante "Login with eIDAS", consente l'interoperabilità transfrontaliera delle identità digitali (eID) fra gli stati membri dell'Unione Europea.

Per tale integrazione, sia con riferimento alla procedura tecnica che a quella amministrativa di federazione dell'SP Regione Aurtonoma Sardegna all'ambiente di Quality Assurance (QA) che a quello di Produzione eIDAS, si richiamano i seguenti avvisi:

- Avviso n. 01-2018 – Note per il dispiegamento del Login EIDAS presso le Pubbliche Amministrazioni,
- Avviso n. 02-2018 – Note per il dispiegamento del Login EIDAS presso gli IDP SPID,

consultabili all'indirizzo https://www.eid.gov.it/abilita-eidas e a cui si rinvia per i relativi dettagli.

2.1.4 Fornitori di servizi

I fornitori dei servizi (Service Provider - SP) inoltrano le richieste di identificazione dell'utente ai gestori dell'identità digitale specificando nella richiesta il livello di autenticazione e il set di attributi dell'utente (richiesti per l'accesso allo specifico servizio).

2.1.5 Gestori degli attributi qualificati

I Gestori degli attributi qualificati (Attribute Authority) attestano il possesso e la validità di attributi qualificati (su richiesta dei fornitori di servizi e previo consenso degli utenti interessati), quali ad es. qualifiche/abilitazioni professionali e poteri di rappresentanza.

3 Access Manager

3.1 Definizione

Con il termine *Access Manager* si intende l'infrastruttura applicativa che permette la gestione della fase di autenticazione per tutti i servizi applicativi erogati dalla Regione Sardegna (in qualità di Service Provider). In particolare, l'Access Manager si occupa della gestione di tutti i processi e le direttive dettati dal CAD, seguendo le regole tecniche del progetto SPID. Il componente rende questa gestione completamente trasparente per tutti i servizi che vi si interfacceranno.

Si precisa che, in fase di analisi e progettazione, assunta la completa migrazione dei servizi regionali sullo SPID, è stata prevista una fase transitoria in cui i sistemi informativi risulteranno agganciati contemporaneamente sia all'IdM-RAS che a SPID in modo da garantire agli utenti la possibilità del doppio accesso. A regime, i componenti Local Proxy e Local Authority Registry dell'IdM-RAS verranno dismessi e l'autenticazione IdM-RAS dovrà essere disabilitata. Pertanti l'Access Manager è stato dotato di apposito modulo per l'accesso tramite TS-CNS in modo da garantire tale tipologia di accesso anche ai servizi non integrati precedentemente con IdM-RAS.

[2] Specifiche contenute nel documento Accesso ai servizi in rete mediante la CIE 3.0 - Manuale operativo per le pubbliche amministrazioni e i soggetti privati (CIE-ManualeOperativoperifornitoridiservizi.pdf).

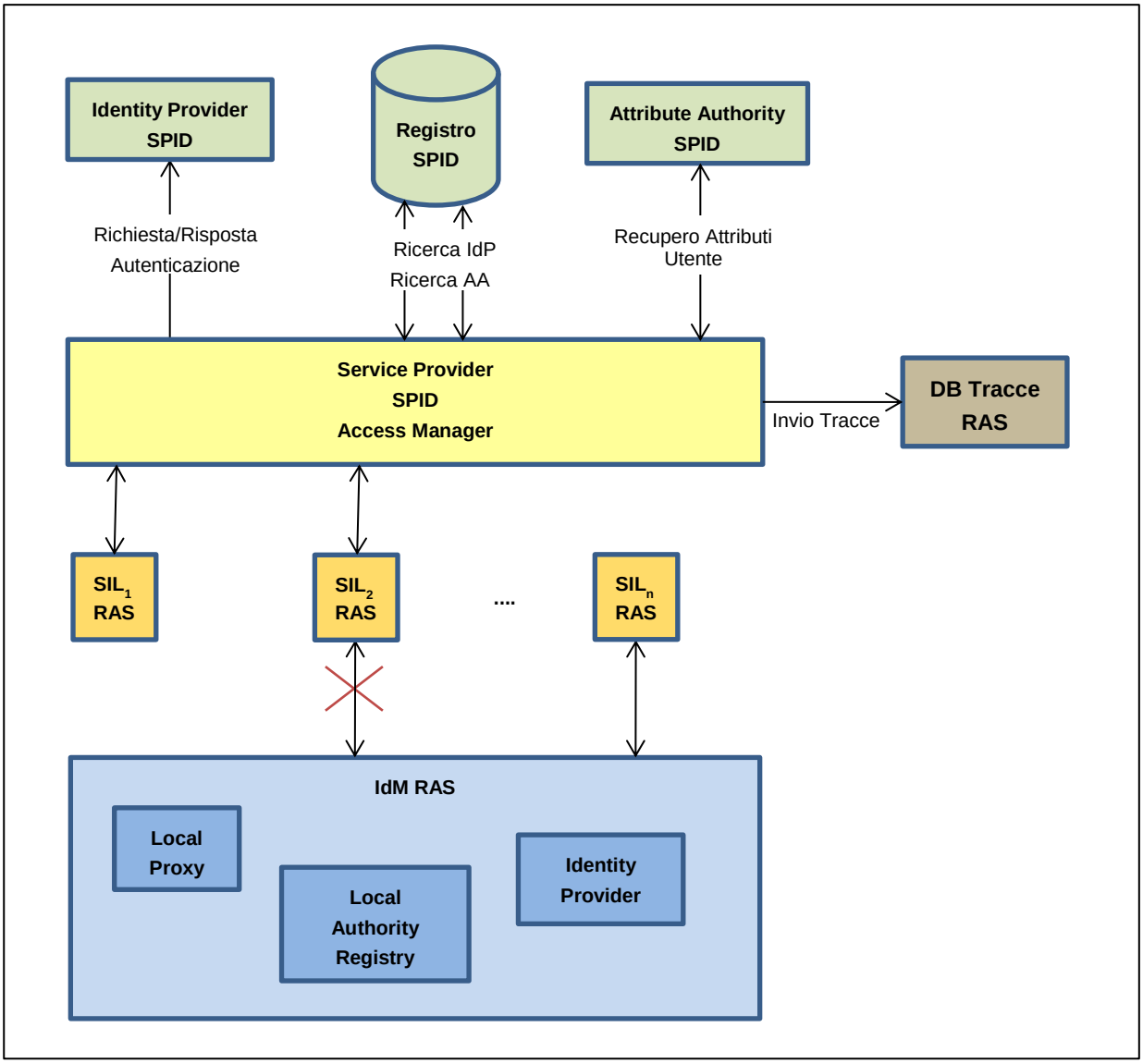


Figura 2 – Implementazione Access Management, situazione transitoria

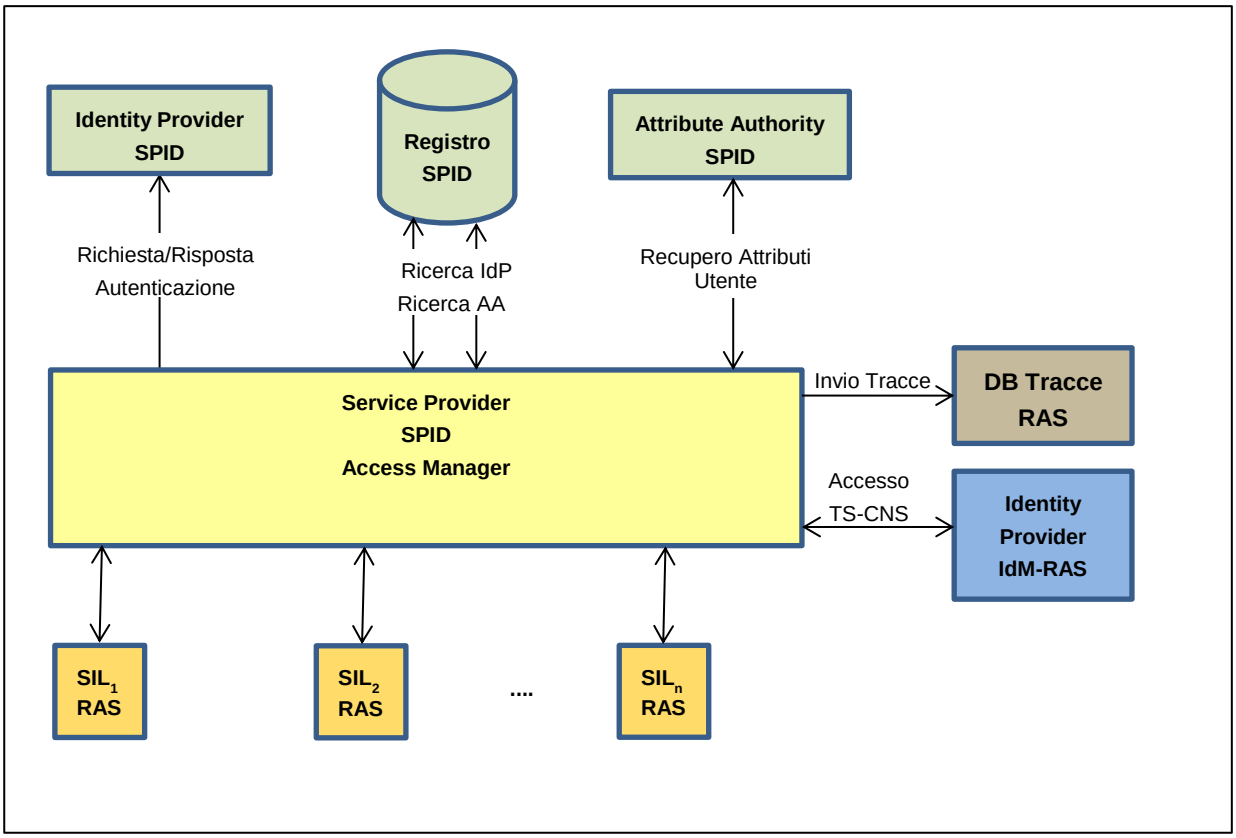


Figura 3 - Implementazione Access Management, situazione a regime

3.2 Architettura Access Manager

L'utilizzo dell'Access Manager centralizzato permette di gestire in due distinte fasi l'autenticazione di ogni utente ed assume, a seconda delle fasi, un ruolo differente:

1. nella fase di dialogo con i servizi RAS, assume il ruolo di Identity Provider,
2. nella fase di interazione con gli Identity Provider SPID, assume il ruolo di Service Provider.

Attualmente sono implementati i due protocolli **SAML** e **OAuth** entrambi nella versione 2.0.

Inoltre l'Access Manager fornisce agli utenti la possibilità di autenticarsi, non solo con credenziali SPID, ma anche mediante le credenziali rilasciate nell'ambito del sistema IdM-RAS.

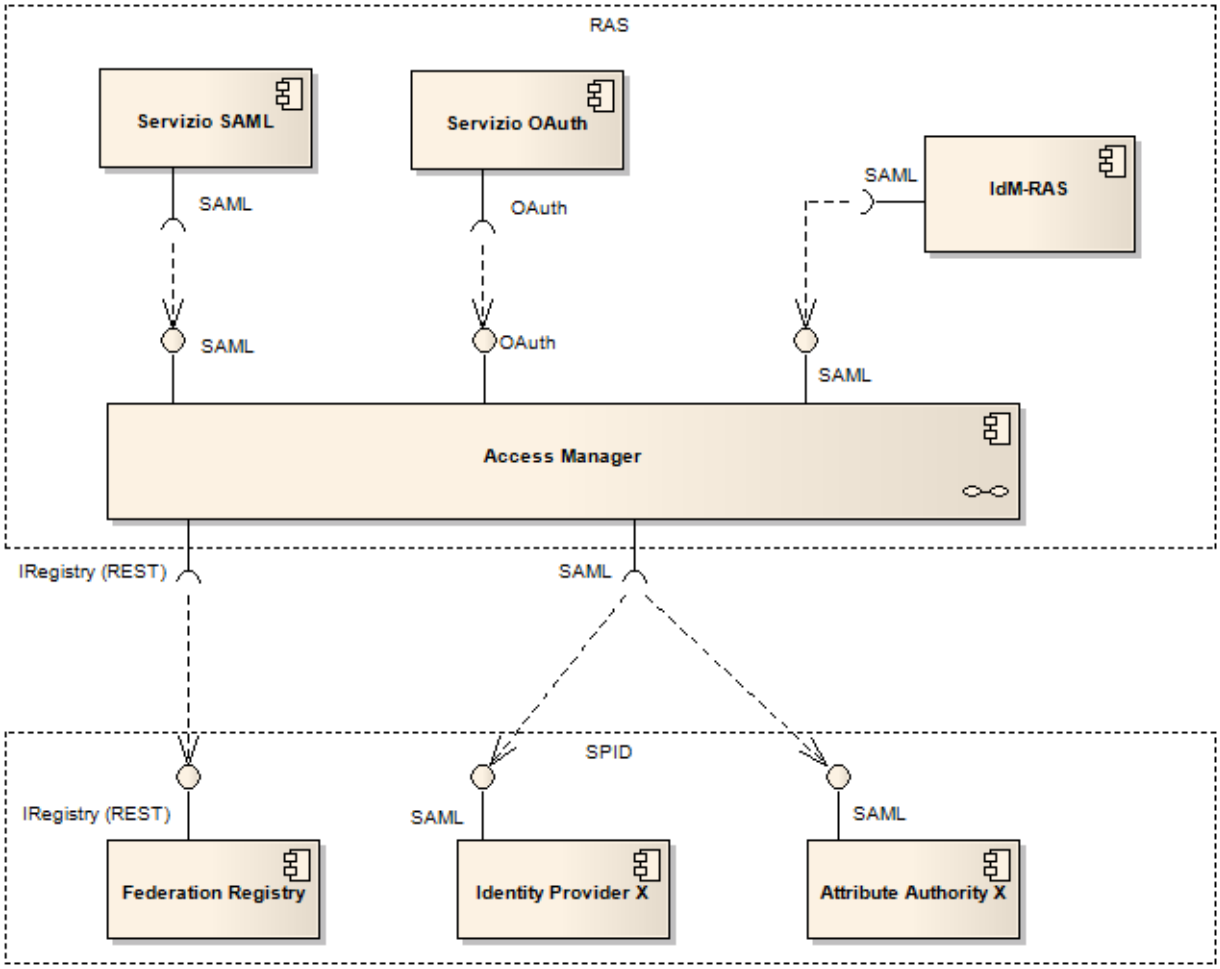


Figura 4 - Architettura generale del sistema e interfacce esposte

In questo caso emulerà il Local Proxy presente nell'architettura del sistema IdM-RAS.

4 Integrazione di un Servizio RAS

In questo capitolo sono analizzate le fasi di autenticazione (login) e logout gestite dall'Access Manager e, in particolare, vengono descritte le interazioni tra gli attori coinvolti nei processi e i protocolli utilizzati.

4.1 Autenticazione

L'Access Manager permette all'utente di autenticarsi mediante TS-CNS, SPID e, inizialmente (fase transitoria), mediante IdM-RAS.

Per comprendere meglio il comportamento dell'Access Manager durante tutto il processo di autenticazione, vengono di seguito descritti i passi seguiti durante l'accesso di un utente ad un generico servizio protetto RAS mediante browser web e senza specificare il protocollo di interfacciamento fra servizio e Access Manager.

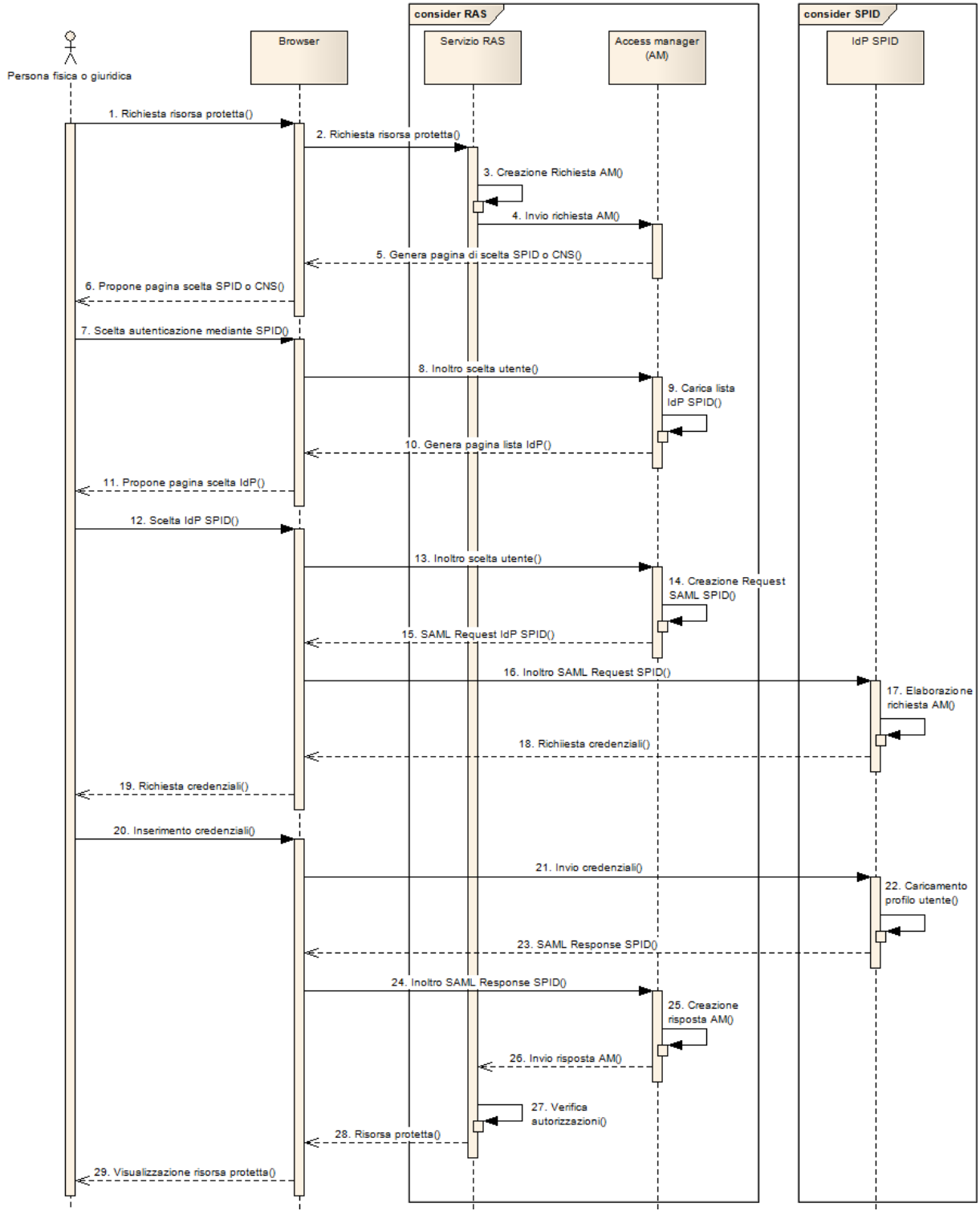


Figura 5 – Funzionamento autenticazione

Step	*Descrizione*
1 – 2	L'utente richiede mediante browser l'accesso ad una risorsa protetta erogata da un Servizio RAS
3	Il Servizio Ras non riconoscendo l'utente genera la richiesta per l'Access Manager
4	Il Servizio Ras invia la richiesta all'Access manager
5 – 6	L'Access Manager genera e invia all'utente mediante browser la pagina di scelta tra autenticazione mediante CNS, SPID, CIE o IdM RAS (per un periodo transitorio)
7 – 8	L'utente sceglie mediante browser l'autenticazione mediante SPID
9	L'Access Manager carica la lista degli Identity Provider SPID disponibili
10 – 11	L'Access Manager genera e invia all'utente mediante browser la pagina di scelta dell'IdP SPID corretto
12 – 13	L'utente sceglie mediante browser un IdP SPID
14	L'Access Manager genera la richiesta SAML per l'IdP SPID scelto
15 – 16	L'Access Manager inoltra all'IdP SPID la richiesta SAML tramite browser
17	L'IdP SPID verifica i dati contenuti nella richiesta
18 – 19	L'IdP SPID genera e propone attraverso il browser una pagina di richiesta credenziali all'utente
20 – 21	L'utente inserisce le credenziali mediante il browser
22	L'IdP SPID verifica le credenziali e carica i dati del profilo utente
23 – 24	L'IdP SPID invia la Response SAML contenente i dati dell'utente all'Access Manager mediante browser
25	L'Access Manager verifica la Response arrivata. Con i dati in essa contenuti genera la risposta per il Servizio RAS
26	L'Access Manager invia la risposta al Servizio RAS
27	Il Servizio RAS verifica le autorizzazioni dell'utente
28 – 29	Il Servizio RAS restituisce la risorsa protetta mediante browser

Tabella 1 – Accesso utente generico servizio RAS

I punti 3, 4, 25 e 26 riassumono genericamente le interazioni fra Access Manager e Servizio RAS.

Il numero di passi e i messaggi che vengono scambiati variano in base al protocollo scelto.

In caso di utilizzo di protocollo SAML, la sequenza è la seguente:

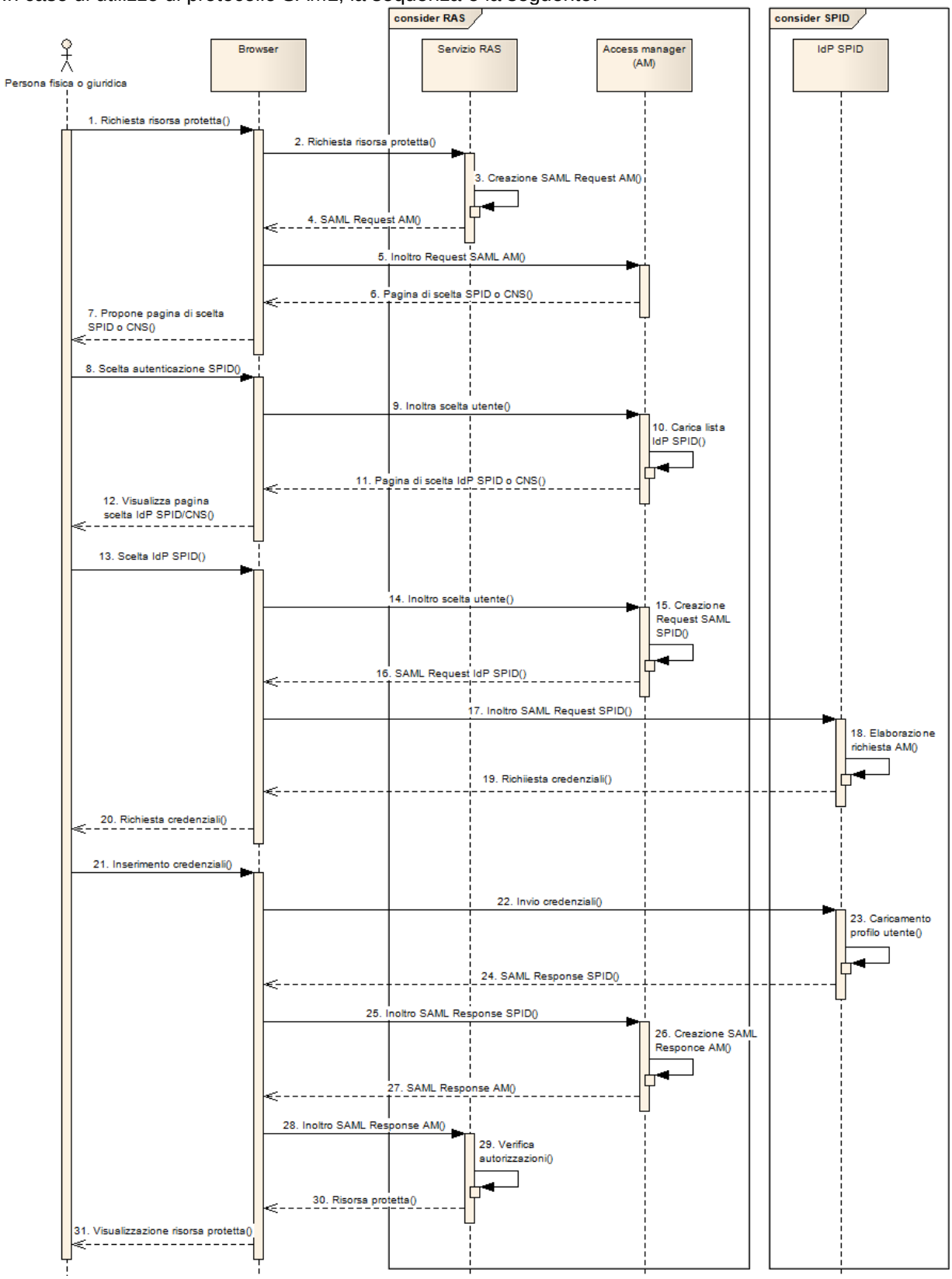


Figura 6 – Funzionamento autenticazione con integrazione SAML

Step	*Descrizione*
1 – 2	L'utente richiede mediante browser l'accesso ad una risorsa protetta erogata da un Servizio RAS
3	Il Servizio Ras non riconoscendo l'utente genera la SAML Request per l'Access Manager
4 – 5	Il Servizio Ras invia la richiesta SAML all'Access manager mediante browser
6 – 7	L'Access Manager genera e invia all'utente mediante browser la pagina di scelta tra autenticazione mediante CNS, SPID o CIE
8 – 9	L'utente sceglie mediante browser l'autenticazione mediante SPID
10	L'Access Manager carica la lista degli Identity Provider SPID disponibili
11 – 12	L'Access Manager genera e invia all'utente mediante browser la pagina di scelta dell'IdP SPID corretto
13 – 14	L'utente sceglie mediante browser un IdP SPID
15	L'Access Manager genera la richiesta SAML per l'IdP SPID scelto
16 – 17	L'Access Manager inoltra all'IdP SPID la richiesta SAML tramite browser
18	L'IdP SPID verifica i dati contenuti nella richiesta
19 – 20	L'IdP SPID genera e propone attraverso il browser una pagina di richiesta credenziali all'utente
21 – 22	L'utente inserisce le credenziali mediante il browser
23	L'IdP SPID verifica le credenziali e carica i dati del profilo utente
24 – 25	L'IdP SPID invia la Response SAML contenente i dati dell'utente all'Access Manager mediante browser
26	L'Access Manager verifica la Response arrivata. Con i dati in essa contenuti genera la Response SAML per il Servizio RAS
27 – 28	L'Access Manager invia la Response SAML al Servizio RAS mediante browser
29	Il Servizio RAS verifica le autorizzazioni dell'utente
30 – 31	Il Servizio RAS restituisce la risorsa protetta mediante browser

Tabella 2 – Flusso con autenticazione SAML

Invece, con l'utilizzo del protocollo OAuth (per cui si rinvia ai dettagli riportati nel paragrafo 4.4), l'Access Manager viene suddiviso logicamente in Authorization Server e Resource Server, riportando alcune variazioni al sistema OAuth.

Nella prima fase l'Authorization Server non fornisce direttamente la pagina di login all'utente, ma lo reindirizza, secondo le scelte fatte dallo stesso, all'Identity Provider SPID.

Al rientro, i dati di profilo vengono conservati dal sistema per renderli disponibili mediante Resource Server.

In caso di utilizzo di protocollo OAUTH, la sequenza è la seguente:

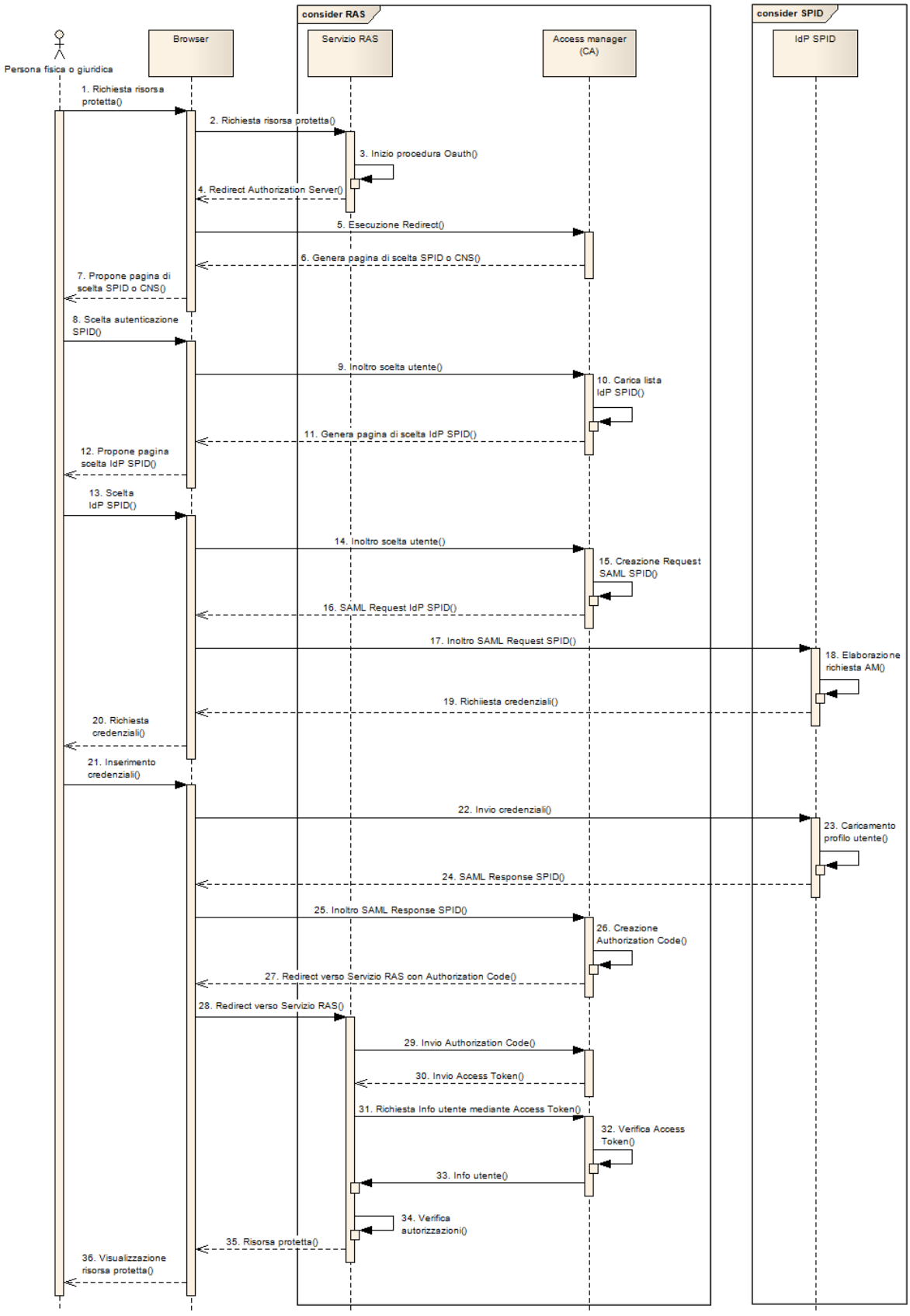


Figura 7 – Funzionamento autenticazione con integrazioneOAuth

Step	*Descrizione*
1 – 2	L'utente richiede mediante browser l'accesso ad una risorsa protetta erogata da un Servizio RAS
3	Il Servizio Ras non riconoscendo l'utente inizia l'autenticazione OAuth
4 – 5	Il Servizio Ras ridirige l'utente verso l'Authorization Server dell'Access manager mediante browser
6 – 7	L'Access Manager genera e invia all'utente mediante browser la pagina di scelta dell'IdP SPID corretto
8 – 9	L'utente sceglie mediante browser l'autenticazione mediante SPID
10	L'Authorization Server dell'Access Manager carica la lista degli Identity Provider SPID disponibili
11 – 12	L'Authorization Server genera e invia all'utente mediante browser la pagina di scelta tra autenticazione mediante CNS o uno degli IdP SPID
13 – 14	L'utente sceglie mediante browser un IdP SPID
15	L'Access Manager genera la richiesta SAML per l'IdP SPID scelto
16 – 17	L'Access Manager inoltra all'IdP SPID la richiesta SAML tramite browser
18	L'IdP SPID verifica i dati contenuti nella richiesta
19 – 20	L'IdP SPID genera e propone attraverso il browser una pagina di richiesta credenziali all'utente
21 – 22	L'utente inserisce le credenziali mediante il browser
23	L'IdP SPID verifica le credenziali e carica i dati del profilo utente
24 – 25	L'IdP SPID invia la Response SAML contenente i dati dell'utente all'Access Manager mediante browser
26	L'Access Manager verifica la Response arrivata. Conserva i dati in essa contenuti per alimentare il Resource Server OAuth e Genera l'Authorization Code per il Servizio RAS
27 – 28	L'Access Manager effettua un redirect verso il Servizio RAS mediante browser passando l'Authorization Code
29	Il Servizio RAS invia l'Authorization Code attraverso le API messe a disposizione dall'Authorization Server dell'Access Manager
30	L'Authorization Server dell'Access Manager restituisce l'Access Token al Servizio RAS
31	Il Servizio RAS invia la richiesta del profilo utente passando l'Access Token attraverso le API messe a disposizione dal Resource Server dell'Access Manager
32	Il Resource Server dell'Access Manager verifica il Token ricevuto e carica le informazioni dell'utente
33	L'Access Manager restituisce le informazioni di profilo dell'utente al Servizio RAS
34	Il Servizio RAS verifica le autorizzazioni dell'utente
35 – 36	Il Servizio RAS restituisce la risorsa protetta mediante browser

Tabella 3 – Flusso con autenticazione OAuth

4.2 Single Logout

La funzionalità di logout permette di chiudere in modo completo e sicuro la propria sessione di lavoro.

Estendendo il concetto all'interno di un sistema governato da *single sign on*, l'operazione effettuata in un contesto applicativo comporta la chiusura di tutte le sessioni di lavoro aperte nelle altre applicazioni/servizi utilizzati dall'utente durante la stessa sessione globale ovvero realizzando quello che è definito come **Single Logout**.

Nel modello Access Manager – SPID descritto in questo documento, la sessione globale di Single Sign on viene mantenuta dall'Identity provider SPID che ha riconosciuto l'utente. Vi è, inoltre, una sessione locale

all'Access Manager che gestisce la chiusura delle sessioni degli applicativi riferiti allo stesso service provider esposto al sistema SPID.

Più in dettaglio l'azione di Single Logout scaturita dalla richiesta dell'utente ad un qualsiasi servizio facente parte della sessione globale segue il flusso descritto qui di seguito:

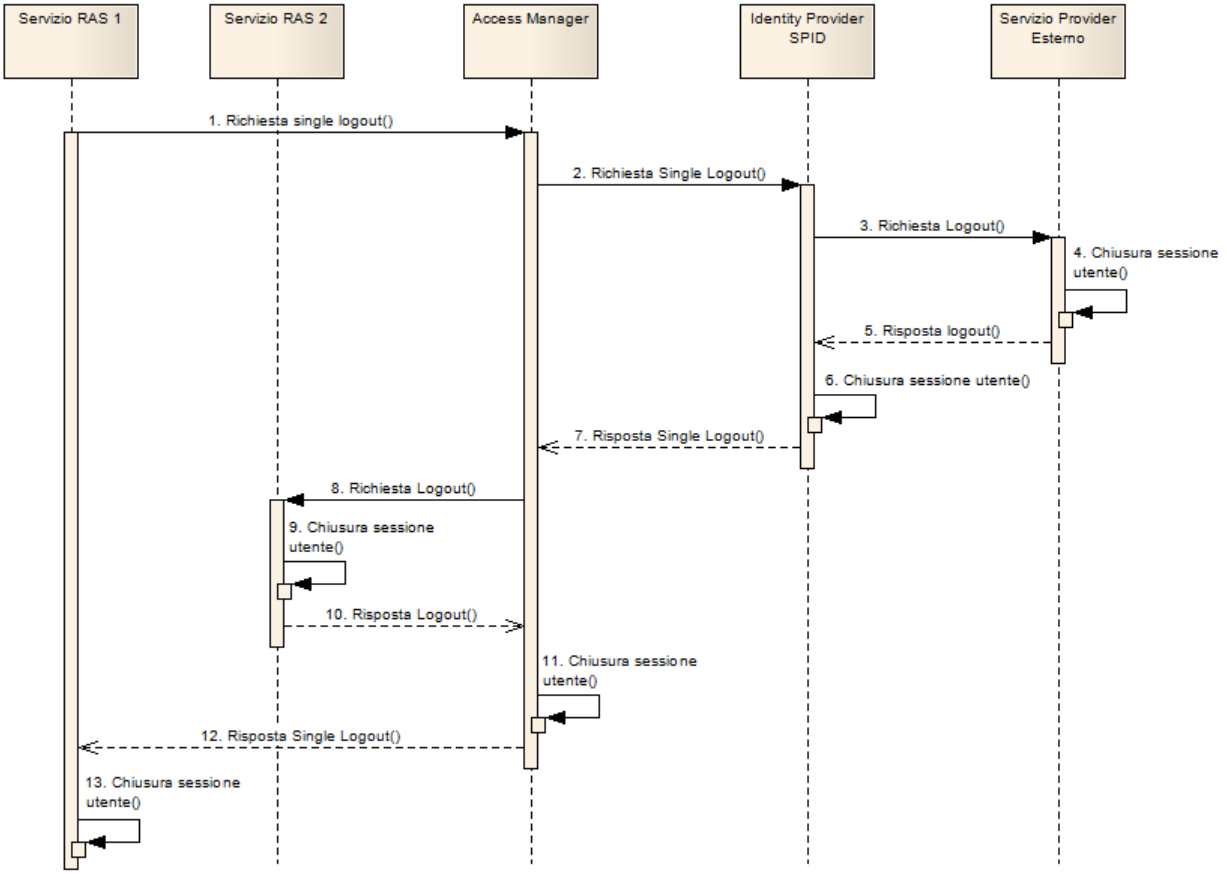


Figura 8 – Funzionamento Single Logout

Step	*Descrizione*
1	Il Servizio RAS 1, su richiesta dell'utente, avvia il processo di Single Logout inviando la richiesta all'Access Manager
2	L'Access Manager invia la richiesta di Single Logout all'Identity provider SPID associato alla sessione locale dell'utente
3 – 4 – 5	L'Identity Provider SPID invia la richiesta di Logout a tutti i Service provider associati alla Sessione Globale . Tutti i Service Provider contattati chiudono la sessione associata all'utente.
6	L'Identity provider chiude la Sessione Globale dell'utente
7	L'Identity provider invia all'Access Manager la risposta di avvenuta chiusura della Sessione Globale
8 – 9 – 10	L'Access Manager invia la richiesta di Logout a tutti i Servizi RAS associati alla Sessione Locale . Tutti i Servizi RAS contattati chiudono la sessione associata all'utente.
11	L'Access Manager provider chiude la Sessione Locale dell'utente
12	L'Access Manager invia al Servizio RAS 1 la risposta di avvenuta chiusura della Sessione Locale
13	Il Servizio RAS 1 chiude la sessione di lavoro dell'utente

Tabella 4 – Flusso Single Logout

4.3 Integrazione SAML

L'Access manager implementa il protocollo SAML in versione 2.0. Nella fase di login e di logout dovranno essere utilizzati rispettivamente i profili Web Browser SSO e Single Logout.

La richiesta di autenticazione da parte del servizio da integrare può essere inviata mediante binding *HTTP Redirect,* in cui la richiesta viene inserita all'interno dell'Url di connessione all'Access manager, o *HTTP POST,* in cui la richiesta viene inviata all'Access manager mediante POST.

La relativa risposta XML verrà inviata solo tramite il binding *HTTP POST*.

Tutti i messaggi scambiati fra le parti devono essere firmati e inviati da un servizio conosciuto dal sistema. Per questo motivo, per ogni servizio da integrare, verranno forniti dal gestore dell'Access Manager un identificativo e relativi certificati di firma riconosciuti dal sistema.

Per quanto riguarda il servizio di ricezione delle richieste di logout da parte dell'Access manager il binding utilizzato sarà di tipo *SOAP*.

4.3.1 Attributi utente

L'Access Manager, una volta ottenuti i dati dall'IdP scelto dall'utente, li elabora e li restituisce al servizio richiedente.

Nella tabella seguente vengono raccolti gli attributi di base disponibili:

Nome attributo	*Descrizione*	*Note*
fiscalNumber	Codice fiscale	
name	Nome	
familyName	Cognome	
gender	Sesso	
dateOfBirth	Data di nascita	
placeOfBirth	Comune di nascita	
countyOfBirth	Provincia di nascita	
address	Indirizzo di residenza completo contenente via, numero civico, cap, comune e provincia tutti separati solo da uno spazio	L'attributo è stato sostituito e suddiviso negli attributi domicileStreetAddress, domicilePostalCode, domicileMunicipality, domicileProvince e domicileNation. Verrà rimosso prossimamente.
domicileStreetAddress	Indirizzo di residenza – via e numero civico	
domicilePostalCode	Indirizzo di residenza – codice postale	
domicileMunicipality	Indirizzo di residenza – comune	
domicileProvince	Indirizzo di residenza – provincia	
domicileNation	Indirizzo di residenza – nazione	
streetAddress	Indirizzo di residenza – via	I campi vengono tenuti per retro compatibilità. L' attributo houseNumberAddress sarà sempre vuoto mentre gli atri saranno valorizzati rispettivamente con gli attributi domicileStreetAddress, domicileMunicipality, domicileProvince e domicilePostalCode
houseNumberAddress	Indirizzo di residenza – numero civico	
cityAddress	Indirizzo di residenza – comune	
countyAddress	Indirizzo di residenza – provincia	
postalCodeAddress	Indirizzo di residenza – CAP	
mobilePhone	Numero cellulare	

Nome attributo	*Descrizione*	*Note*
digitalAddress	Indirizzo PEC	
email	Indirizzo E-Mail	
idCard	Dati riguardanti il documento di identità contenente tipo di documento (valori possibili *cartaIdentita, passaporto, patenteGuida; patenteNautica; librettoPensione, patentinoImpTermici, portoArmi, tesseraRiconoscimento*), numero documento, ente emettitore (senza spazi es:*regioneLombardia*), data emissione, data scadenza	Campo disponibile solo per utenze SPID
typeIdCard	Documento di identità – tipo Valori possibili 3. *cartaIdentita* 4. *passaporto* 5. *patenteGuida* 6. *patenteNautica* 7. *librettoPensione* 8. *patentinoImpTermici* 9. *portoArmi* 10. *tesseraRiconoscimento*	Campi disponibili solo per utenze SPID. I valori vengono ottenuti dal parsing del campo “idCard”
numberIdCard	Documento di identità – numero	
issuerIdCard	Documento di identità – emettitore	
issueDateIdCard	Documento di identità – data emissione	
expiringDateIdCard	Documento di identità – data scadenza	
ivaCode	Partita iva	Campo disponibile solo per utenze professionali SPID e solo per le persone giuridiche
companyName	Ragione o denominazione sociale	Campo disponibile solo per utenze professionali SPID e solo per le persone giuridiche
companyFiscalNumber	Codice fiscale persona giuridica	Campo disponibile solo per utenze professionali SPID e solo per le persone giuridiche
registeredOffice	Indirizzo completo sede legale contenente via, numero civico, cap, comune e provincia tutti separati solo da uno spazio	Campo disponibile solo per utenze SPID e solo per le persone giuridiche
streetRegisteredOffice	Indirizzo sede legale – via	Campo disponibile solo per utenze professionali SPID e solo per le persone giuridiche. I valori vengono ottenuti dal parsing del campo “registeredOffice”
houseNumberRegisteredOffice	Indirizzo sede legale – numero civico	
cityRegisteredOffice	Indirizzo sede legale – comune	
countyRegisteredOffice	Indirizzo sede legale – provincia	
postalCodeRegisteredOffice	Indirizzo sede legale – cap	
spidCode	Codice univoco dell’identità SPID	Campo disponibile solo per utenze SPID
expirationDate	Data scadenza identità	Campo disponibile solo per utenze SPID

Tabella 5 - Elenco attributi utente (base)

*Nota: *Tutte le date restituite negli attributi sono in formato aaaa-mm-gg e le province vengono sempre indicate con l’abbreviazione a due lettere.*

L'Access Manager restituisce il sotto insieme minimo di questi attributi in base alle esigenze del servizio richiedente.

Per ragioni di privacy tale set minimo dovrà includere solo gli attributi necessari al corretto funzionamento del singolo servizio e verrà concordato in fase di integrazione dello stesso.

Di seguito si espone la tabella contenente tutti i set configurati sul sistema.

Potranno essere implementati, dopo esplicita autorizzazione della Regione Sardegna, set di attributi diversi da quelli già disponibili per i servizi che lo richiederanno.

Set	*Attributi*
0	spidCode name familyName fiscalNumber email expirationDate
1	spidCode name familyName placeOfBirth countyOfBirth dateOfBirth gender companyName registeredOffice fiscalNumber ivaCode idCard mobilePhone email address domicileStreetAddress domicilePostalCode domicileMunicipality domicileProvince domicileNation digitalAddress expirationDate
2	name familyName fiscalNumber spidCode
3	name familyName fiscalNumber spidCode email
4	name familyName fiscalNumber spidCode email address domicileStreetAddress domicilePostalCode domicileMunicipality domicileProvince domicileNation

Set	*Attributi*
5	name familyName fiscalNumber spidCode email address domicileStreetAddress domicilePostalCode domicileMunicipality domicileProvince domicileNation placeOfBirth countyOfBirth dateOfBirth
6	name familyName fiscalNumber spidCode email address domicileStreetAddress domicilePostalCode domicileMunicipality domicileProvince domicileNation placeOfBirth countyOfBirth dateOfBirth gender
7	fiscalNumber spidCode
8	spidCode name familyName fiscalNumber email mobilePhone digitalAddress
9	spidCode name familyName fiscalNumber email mobilePhone expirationDate
10	spidCode name familyName fiscalNumber email mobilePhone placeOfBirth countyOfBirth dateOfBirth expirationDate

Set	*Attributi*
11	spidCode name familyName fiscalNumber email mobilePhone placeOfBirth countyOfBirth dateOfBirth expirationDate gender
12	spidCode expirationDate name familyName gender fiscalNumber placeOfBirth countyOfBirth dateOfBirth email mobilePhone address domicileStreetAddress domicilePostalCode domicileMunicipality domicileProvince domicileNation digitalAddress
13	name familyName fiscalNumber
14	fiscalNumber
15	name familyName fiscalNumber email
16	spidCode name familyName placeOfBirth countyOfBirth dateOfBirth companyName registeredOffice fiscalNumber ivaCode idCard mobilePhone email address domicileStreetAddress domicilePostalCode domicileMunicipality domicileProvince domicileNation digitalAddress expirationDate

Set	*Attributi*
17	spidCode name familyName placeOfBirth countyOfBirth dateOfBirth companyName registeredOffice fiscalNumber ivaCode email address domicileStreetAddress domicilePostalCode domicileMunicipality domicileProvince domicileNation digitalAddress
18	name familyName placeOfBirth dateOfBirth fiscalNumber

Tabella 6 - Set attributi del Metadata RAS[3]

In futuro potranno essere aggiunti degli attributi forniti da specifiche attribute authority e saranno comunicati ai servizi interessati non appena verranno resi disponibili.

4.3.2 Attributi CIE

L'Identity provider CIE può restituire solo un sottoinsieme degli attributi disponibili con gli IdP SPID:

- fiscalNumber
- name
- familyName
- dateOfBirth

L'utilizzo di questo identity provider potrà essere attivato su qualsiasi servizio previa verifica di corretto funzionamento.

4.3.3 Attributi EIDAS

L'Identity provider EIDAS è un gateway che permette l'autenticazione dei cittadini delle nazioni europee federate col sistema. Per queste identità l'attributo di riconoscimento univoco non è più il codice fiscale (*fiscalNumber*) ma lo spid code (*spidCode*). Gli unici set disponibili sono l'"eIDAS Natural Person Minimum Attribute Set" con gli attributi:

- spidCode
- name
- familyName
- dateOfBirth

e l'"eIDAS Natural Person Full Attribute Set" con gli attributi

- spidCode
- name
- familyName

[3] Vedasi su http://www.regione.sardegna.it/servizispid/ la sottosezione Metadati Regione Sardegna > Metadata Service Provider Regione Sardegna (https://login.regione.sardegna.it/saml/metadata?entityid=accessmanager.spid.regione.sardegna.it)

- dateOfBirth
- placeOfBirth
- address
- gender

4.3.4 Attributi di contesto

Oltre agli attributi utente provenienti dagli IdP o dalle attribute autority verranno restituiti anche tre attributi di contesto. Questi permetteranno al servizio chiamante di capire quale Identity provider è stato scelto dall'utente e saranno sempre presenti.

Nome attributo	*Descrizione*	*Note*
idpSelected	Entity Id dell'IdP scelto	
idpTypeSelected	Tipologia dell'IdP scelto	Tipologie possibili: - *SPID* : Identity provider autorizzato su SPID - *IDM* : IdM-RAS - *CIE* : servizio CIE, IdP del ministero dell'interno - *EIDAS* : servizio gateway per la gestione degli utenti della comunità europea - *AM-CNS* : autenticazione mediante CNS su Access Manager
authenticationMethodLevel	Livello autenticazione	Valori possibilli: - https://www.spid.gov.it/SpidL1 - https://www.spid.gov.it/SpidL2 - https://www.spid.gov.it/SpidL3

4.3.5 Integrazione servizio Java con Spring (3.x o 4.x)

Per i progetti che utilizzano il modulo spring security è stata creata una libreria ad hoc "Spring Security Access Manager" che sfrutta il protocollo SAML2 attraverso l'estensione "Spring Security SAML v2 library" (spring-security-saml2-core).

La versione 1.0.2 attualmente disponibile è compatibile sia con spring security versione 3 e versione 4.

Per la corretta integrazione è necessario impostare il file xml che riporta le configurazioni del modulo spring security l'entry point di autenticazione sul tag http e inserire tutti i filtri utili alla libreria.

Al suo interno vengono configurati i riferimenti:

- ai server Access Manager

```
<bean class="org.opensaml.saml2.metadata.provider.HTTPMetadataProvider">
    <constructor-arg>
        <value type="java.lang.String"> indirizzo metadati access manager </value>
    </constructor-arg>
    <constructor-arg>
        <value type="int">5000</value>
    </constructor-arg>
    <property name="parserPool" ref="parserPool"/>
</bean>
```

- al file dei metadati del servizio,

```
<bean class="org.springframework.security.saml.metadata.ExtendedMetadataDelegate">
    <constructor-arg>
        <bean class="org.opensaml.saml2.metadata.provider.FilesystemMetadataProvider">
            <constructor-arg>
                <value type="java.io.File">path file metadata servizio</value>
            </constructor-arg>
                <property name="parserPool" ref="parserPool"/>
        </bean>
    </constructor-arg>
    <constructor-arg>
```

```
            <bean class="org.springframework.security.saml.metadata.ExtendedMetadata"/>
        </constructor-arg>
    </bean>
```

- al certificato utilizzato per la firma delle asserzioni e il nome del servizio stesso (hostedSPName)

```
    <bean id="keyManager" class="org.springframework.security.saml.key.JKSKeyManager">
            <constructor-arg value="path keystore firma"/>
            <constructor-arg type="java.lang.String" value="password keystore firma"/>
            <constructor-arg>
                <map>
                    <entry key="alias chiave firma" value="password chiave firma"/>
                </map>
            </constructor-arg>
            <constructor-arg type="java.lang.String" value="alias chiave firma"/>
    </bean>
```

Copia precompilata del file xml di cui sopra verrà fornita come parte del kit associato che verrà fornito insieme alla presente guida.

Per quanto riguarda il codice da implementare per l'integrazione, tutti i dati del profilo provenienti dall'Access Manager vengono memorizzati su una istanza della classe

org.springframework.security.saml.SAMLCredential

presente all'interno dell'Authentication del ServiceContext.

Attraverso la classe

it.sardegnait.base.detailservice.UserAttributeUtil

contenente il metodo getAttributeFirstValue(<nome campo>) è possibile recuperare i valori degli attributi. I nomi dei campi sono presenti nella stessa classe mediante delle costanti.

È inoltre possibile sviluppare una classe che implementa

org.springframework.security.saml.userdetails.SAMLUserDetailsService.

Questo bean deve essere configurato e passato a *userDetails* del bean *samlAuthenticationProvider* all'interno del file di configurazione sopra descritto.

Di questa classe verrà richiamato il metodo loadUserBySAML() a cui verrà passato l'oggetto SAMLCredential e su cui può essere inserito il codice per la gestione dell'integrazione con i dati dell'applicazione.

Per il logout deve essere richiamato il link <nome_webapp>/saml/logout?local=false. Per attività di test è possibile mettendo il parametro local a true effettuare un logout solo a livello applicativo senza richiamare il servizio di single logout dell'Access Manager e in cascata quello dell'Identity Provider scelto dall'utente.

Come esempio verrà fornito un servizio di test su cui sono implementate tutte le classi sopra descritte.

4.3.6 Migrazione servizi da IdM-RAS a Access Manager

Negli ultimi anni la maggior parte delle applicazioni web per la Regione Sardegna sono state integrate con il sistema di autenticazione IdM-RAS. Questo servizio, come l'Access Manager sfrutta il protocollo SAML.

Di seguito vengono riportate le modifiche da effettuare per integrare le applicazioni che attualmente utilizzano l'IdM-RAS per l'autenticazione degli utenti.

4.3.6.1 Modifica attributi

Gli attributi di base restituiti dall'Access Manager, come visto sopra, sono diversi da quelli restituiti finora dall'IdM-RAS. Di seguito viene riportata la tabella di conversione:

Nome attributo Access Manager	*Nome attributo IdM-RAS*	*Descrizione*
fiscalNumber	codiceFiscale	Codice fiscale
name	nome	Nome
familyName	Cognome	Cognome

Nome attributo Access Manager	*Nome attributo IdM-RAS*	*Descrizione*
gender	Sesso	Sesso
dateOfBirth	dataNascita	Data di nascita
placeOfBirth	luogoNascita	Comune di nascita
countyOfBirth	provinciaNascita	Provincia di nascita
streetAddress	indirizzoResidenza	Indirizzo di residenza – via
houseNumberAddress	nrCivicoResidenza	Indirizzo di residenza – numero civico
cityAddress	cittaResidenza	Indirizzo di residenza – comune
countyAddress	Provincia	Indirizzo di residenza – provincia
postalCodeAddress	capResidenza	Indirizzo di residenza – CAP
mobilePhone	Cellulare	Numero cellulare
email	emailAddress	Indirizzo E-Mail
-	telefono	Numero telefono

Tabella 7 - Tabella di conversione attributi

*Nota: *Tutte le date restituite negli attributi sono in formato aaaa-mm-gg e le province vengono sempre indicate con l'abbreviazione a due lettere.*

4.3.6.2 Migrazione servizio Java su Spring Framework

Per l'interfacciamento dei servizi java che si basano su Spring Framework integrati con l'IdM-RAS, è stata utilizzata finora l'estensione spring-security_idm-ras per Spring Security che si basa su una versione preliminare della libreria spring-security-saml2-core.

Vista la similarità delle librerie di partenza, per entrambe è spring-security-saml2-core ma in versione diversa, le modifiche da fare si possono ridurre a quelle elencate di seguito:

1. Eliminare le dipendenze verso le librerie spring-security_idm-ras e spring-security-saml2-core
2. Aggiungere la dipendenza alla libreria "Spring Security Access Manager" descritta sopra
3. Inserire il nuovo file di configurazione per spring security simile all'esempio precedente e fornito ad inizio attività
4. Modificare tutti i path e le varie configurazioni riguardanti il certificato di firma, l'id del servizio e l'indirizzo dell'Access Manager, come riportato nel paragrafo 4.3.2
5. Modificare l'implementazione dell'interfaccia SAMLUserDetailsService presente sul codice per gestire i nuovi attributi descritti in precedenza

Una volta effettuate queste modifiche l'applicazione è pronta per interfacciarsi con l'Access Manager.

4.3.6.3 Migrazione servizio Java su ras-sp-integration-kit

Per l'interfacciamento dei servizi java già integrati con l'IdM-RAS senza Spring Framework è stata utilizzata finora la libreria ras-sp-integration-kit che si basa sulle librerie del progetto ICAR. Per permettere l'interfacciamento con l'Access Manager si dovranno seguire i passi elencati di seguito:

1. Sostituire la libreria ras-sp-integration-kit con la nuova versione fornita
2. Modificare su web.xml :
    - Sostituire i dati del certificato di firma nei parametri keystorePath, keystoreAlias e keystorePassword
    - Modificare se necessario il parametro entityID con l'id fornito per l'AM

- Modificare il parametro localProxyMetadataProviderURL con l'url dei metadati riferito all'ambiente dell'AM fornito
- Modificare il valore del parametro identifyingAttribute con fiscalNumber

3. Sostituire il file dei metadati dell'applicativo con quello fornito

4.4 Integrazione OAuth2

L'Access Manager implementa il protocollo OAUTH in versione 2.0 con Grant Authorization Code. Questo tipo di Grant utilizza un codice che viene passato dopo il riconoscimento dell'utente e deve essere usato dal client per la richiesta del token.

Gli indirizzi che l'applicazione client deve utilizzare per la fase di autenticazione sono:

- *<indirizzo_access_manager>/oauth/authorize* : chiamata in GET che inizia la fase di autenticazione. Parametri richiesti:
  - client_id : id del client
  - redirect_uri : uri di callback verso il servizio client
  - response_type : valorizzato a 'code'
  - scope : valorizzato a 'read'
  - state : valore identificativo della sessione client
- *<indirizzo_access_manager>/oauth/token* : chiamata in POST da client per la richiesta del token di autenticazione. La chiamata deve avvenire con basic authentication contenente client_id e secret forniti. Parametri richiesti:
  - grant_type : sempre valorizzato 'authorization_code'
  - code : codice ricevuto nella callback di ritorno al client come parametro 'code'
  - redirect_uri: uri di callback verso il servizio client
- *<indirizzo_access_manager>/SSOServiceOAuth2* : Chiamata in REST che, dato il token, restituisce i dati dell'utente in formato JSON

Il valore di *<indirizzo_access_manager>* varia in base all'ambiente con cui ci si sta integrando:

- https://accessmanager.demo.sardegnait.it/access-manager per gli ambienti di sviluppo e demo
- https://login.regione.sardegna.it/ per gli ambienti di produzione

Per la propagazione del logout il client deve effettuare una chiamata in REST con method DELETE verso l'indirizzo Access Manager *<indirizzo_access_manager>/revoke-token* passando nell'header http 'Authorization' il bearer token. In questo modo Access Manager provvedeeà alla cancellazione del token associato al servizio, alla propagazione della richiesta di logout verso l'IdP scelto in precedenza dall'utente e alla rimozione dei dati dell'utente dalla sessione.

4.4.1 Attributi utente

L'Access Manager, una volta ottenuti i dati dall'IdP scelto dall'utente, li elabora e li restituisce al servizio richiedente attraverso un oggetto JSON contenente lo userid (di tipo stringa) e una mappa degli attributi di base indicati nella tabella 2 aventi come chiave il nome dell'attributo, inviato come risposta alla chiamata REST effettuata in precedenza dal servizio.

4.4.2 Configurazione dei servizi

Per il riconoscimento di ogni servizio occorre che alcune informazioni corrispondano con quelle archiviate nel database di Access Manager.

Lato client occorre configurare i parametri relativi a:

1. Client id (ex: "oauth2client"): nome utilizzato per identificare il client;
2. ClientSecret (ex: "123456789"): password da associare al client;
3. AccessTokenUri (<nome_webapp>/oauth/token) url di access manager da richiamare per ottenere il toklen di riconoscimento del client;
4. UserAuthorizationUri(<nome_webapp>/oauth/authorize) url di access manager per la verifica dell'autorizzazione del client;

5. GrantType("authorization_code"): tipo di grant, utilizzare esclusivamente “authorization_code”;
6. Scope("read"): scope, utilizzare esclusivamente “read”.

4.4.2.1 Integrazione servizio Java con Spring (3.x o 4.x)

Per i progetti java che utilizzano il framework Spring è possibile effettuare la configurazione come riportato nel seguente file di esempio:

```
@Configuration
@EnableOAuth2Client
protected static class ResourceConfiguration {
  @Value("${accessManager.uri}")
  private String accessManagerUri;//<nome webapp>
  @Bean
  public OAuth2ProtectedResourceDetails provaClient() {
    AuthorizationCodeResourceDetails details = new AuthorizationCodeResourceDetails();
    details.setClientId("oauth2client");
    details.setClientSecret("123456789");
    details.setAccessTokenUri(accessManagerUri + "/oauth/token");
    details.setUserAuthorizationUri(accessManagerUri + "/oauth/authorize");
    details.setGrantType("authorization_code");
    details.setScope(Arrays.asList("read"));
    return details;
}
```

5 Procedura amministrativa

Il processo di integrazione di un servizio con l’Access Manager, parallelamente all’attività propriamente tecnica, richiede l’esecuzione di una procedura amministrativa nei confronti di RAS i cui dettagli sono descritti nel presente capitolo.

In particolare, ogni volta che si integra un servizio con l’Access Manager occorre:

1. Compilare e trasmettere al referente regionale per lo SPID e RUP (Ing. Gualtiero Asunis) un file di registro contenente le informazioni necessarie ad AgID per monitorare lo sviluppo del sistema ed il grado di attuazione del disposto normativo che lo ha introdotto in termini di servizi integrati;
2. Effettuare l’aggiornamento dell’elenco dei servizi regionali che utilizzano lo SPID (http://www.regione.sardegna.it/servizispid/) ogniqualvolta viene associato nuovo un applicativo/o modificato uno precedentemente inserito (attenzione non modificare l’impostazione del file che riporta dei campi selezionabili).

Si consideri che:

- il servizio RAS competente per l’Access Manager deve ricevere il file compilato/aggiornato almeno **una settimana prima della data prevista di pubblicazione**,
- possibilmente, l’aggiornamento deve essere anticipato al referente Regionale ed eventualmente condiviso in anteprima prima dell’invio formale per eventuali dettagli e/o chiarimenti e/o richieste,
- È auspicabile, ove possibile, che l’invio venga effettuato **dal soggetto che ha la titolarità del sistema da integrare** o quantomeno anticipare che, se si procederà diversamente, l’invio a referente sarà

effettuato in base a delega da parte del soggetto che gestisce il sistema o ne coordina la realizzazione/gestione per conto del titolare.
- In via generale, Sardegna IT ha ruolo di supportare i diversi soggetti coinvolti nella compilazione del registro.

Separatamente e contestualmente all'avvio della procedura propriamente tecnica, vengono inviate al titolare del servizio richiedente l'integrazione i dettagli sulla procedura amministrativa e la documentazione da compilarsi con le relative istruzioni per trasmissione ad AA.GG[4].

Nella tabella successiva[5], si riportano brevemente i passi propedeutici e consecutivi al rilascio in produzione dell'integrazione con il sistema Access Manager.

[4] I dettagli della procedura sono esplicitati nel documento SPD-2021-2022 - D01.03.01 – Nuova procedura amministrativa integrazione AM-RAS.docx

[5] Attenzione, alcune informazioni e i riferimenti indicati nella tabella potrebbero variare in ragione di modifiche/aggiornamenti intervenuti nel corso del tempo in ragione di specifiche esigenze di RAS. Pertanto si deve fare riferimento alla documentazione aggiornata relativa alla procedura amministrativa che, tempo per tempo, viene trasmessa al titolare del servizio richiedente l'integrazione con l'AM-RAS per il suo avvio.

<table>
<tr><th>Quando</th><th>Che cosa</th><th>Chi</th><th>Come</th><th>Note operative</th></tr>
<tr><td rowspan="2">Avvio</td><td>Richiesta della documentazione amministrativa (registro) per l'integrazione con il sistema AM-RAS</td><td>Titolare del servizio/sistema da integrare</td><td>Contatta per email il Referente Regionale per lo SPID/RUP della Convezione RAS-AgID</td><td>Invia una comunicazione ad AA.GG per informarla dell'esigenza di avviare la procedura di integrazione con l'AM-RAS per il servizio di cui è titolare e richiedendo il file "Servizi SPID_aaaa-mm-gg.ods".</td></tr>
<tr><td>Richiesta della documentazione tecnica</td><td>Titolare del servizio/sistema da integrare e/o il suo supporto tecnico</td><td>Contatta Sardegna IT inviando una email a supporto.login@regione.sardegna.it
Come indicato nelle note tecniche pubblicate su http://www.regione.sardegna.it/servizi spid/</td><td>Richiesta a Sardegna IT della documentazione tecnica aggiornata per l'integrazione con il sistema AM-RAS.
Sardegna IT, fornisce in risposta:
- la documentazione e le ulteriori informazioni preliminari necessarie per l'integrazione ed eventuali modalità operative,
- le informazioni per raggiungere gli ambienti di demo/stage,
- i riferimenti da contattare per la gestione delle fasi successive.</td></tr>
<tr><td rowspan="2">Definizione e parametri di base</td><td>Scelta del livello di sicurezza SPID da configurare per il sistema oggetto di integrazione</td><td>Titolare del servizio/sistema da integrare e/o il suo supporto tecnico</td><td rowspan="2">Contatta i riferimenti del supporto tecnico forniti in sede di avvio</td><td>Il richiedente, con l'eventuale supporto di Sardegna IT (area tecnica) che illustra le differenze intercorrenti a livello applicativo tra i livelli SPID, individua il livello minimo di credenziale SPID (1, 2, 3) da impiegare per il servizio specifico.
A titolo di riferimento si veda l'Avviso nr. 4 Agid
https://www.agid.gov.it/sites/default/files/repository_files/documentazione/spid-avviso-n4-livelli-servizio-minimo-funz-omogenee.pdf</td></tr>
<tr><td>Scelta degli attributi utente da utilizzare per il sistema oggetto di integrazione</td><td>Titolare del servizio/sistema da integrare e/o il suo supporto tecnico</td><td>Il richiedente indica e motiva gli attributi utente da adottarsi per il sistema di riferimento. Sardegna IT fornisce eventuale supporto per la migliore comprensione degli attributi. Gli attributi richiesti per ciascun servizio devono essere identificati/selezionati considerando quelli strettamente indispensabili per l'erogazione del servizio.
<table>
<tr><th>Attributo</th></tr>
<tr><td>Codice identificativo SPID</td></tr>
<tr><td>Nome</td></tr>
<tr><td>Cognome</td></tr>
<tr><td>Luogo di nascita</td></tr>
<tr><td>Data di nascita</td></tr>
<tr><td>Sesso</td></tr>
<tr><td>Ragione o denominazione sociale</td></tr>
<tr><td>Sede legale</td></tr>
<tr><td>Codice fiscale</td></tr>
<tr><td>Partita IVA</td></tr>
<tr><td>Documento d'identità</td></tr>
<tr><td>Numero di telefono mobile</td></tr>
<tr><td>Indirizzo di posta elettronica</td></tr>
<tr><td>Domicilio fisico</td></tr>
<tr><td>Domicilio digitale</td></tr>
<tr><td>Data di scadenza identità</td></tr>
<tr><td>Provincia di nascita</td></tr>
</table></td></tr>
</table>

Quando	Che cosa	Chi	Come	Note operative
				Gli attributi selezionabili riportati nel file "**Servizi SPID_aaaa-mm-gg.ods**" sono altresì riportati nel Metadata Service Provider Regione Sardegna pubblicato su http://www.regione.sardegna.it/servizispid/ alla fine della pagina sotto la voce "**Metadati Regione Sardegna**". **N.B.** I set di attributi con indice 99 e 100 presenti nel tracciato XML di cui al link https://login.regione.sardegna.it/saml/metadata?entityid=accessmanager.spid.regione.sardegna.it (rispettivamente minimum dataset e full dataset eIDAS) sono riservati alla sola procedura di identificazione elettronica tramite eIDAS, pertanto non sono utilizzabili per l'autenticazione tramite identità SPID.
Test in ambiente di demo/stage	Esecuzione di prove preliminari di integrazione in ambiente di demo/stage sulla base dei parametri individuati	Titolare del servizio/sistema da integrare e/o il suo supporto tecnico	Contatta i riferimenti del supporto tecnico forniti in sede di avvio	Il richiedente, sulla base della documentazione e delle informazioni fornite dal supporto tecnico di Sardegna IT in fase di avvio, procede con l'esecuzione degli interventi tecnici necessari per l'integrazione del sistema con la piattaforma AM-RAS. Il supporto tecnico: - riscontra gli esiti dal lato dell'AM-RAS, - fornisce le ulteriori informazioni utili per completare i test di integrazione negli ambienti specificati. **N.B.** Se, per esigenze specifiche del servizio, le strutture richiedenti necessitano di una combinazione di attributi utente non compresa tra quelle disponibili (v. Errore. L'origine riferimento non è stata trovata.), Sardegna IT predispone le attività interne per l'aggiornamento del metadata della regione Sardegna e la sua successiva trasmissione ad AgID da parte di RAS AA.GG per la relativa verifica e approvazione alla messa in esercizio degli aggiornamenti introdotti.
Prima di andare in produzione	Compilazione "Servizi SPID_aaaa-mm-gg.ods".	Titolare del servizio/sistema da integrare	Compilazione del file di registro ods precedentemente richiesto e sua trasmissione via PEC al Referente Regionale per lo SPID/RUP della Convenzione	Compilazione del registro "**Servizi SPID_aaaa-mm-gg.ods**" e sua trasmissione formale via pec all' **Assessorato degli affari generali, personale e riforma della Regione** Direzione generale dell'innovazione e Sicurezza IT Servizio Sistemi - Settore Piattaforme Digitali Via Posada 1 09122 CAGLIARI - CA *c.a. ing. Nicoletta Sannio* *c.a. Ing. Gualtiero Asunis* *PEC:* innovazione@pec.regione.sardegna.it La trasmissione dovrà avvenire preferibilmente entro i 15 giorni lavorativi dalla data prevista/fissata per la messa in produzione. Il registro dovrà essere compilato secondo le istruzioni contenute nel foglio "Guida_alla_compilazione" ed in tutte le sue parti (Foglio Servizi e PA_Utilizzatrici). Il file dovrà essere salvato nello stesso formato originale,

Quando	Che cosa	Chi	Come	Note operative
				aggiungendo nel nome la data di trasmissione dello stesso. La data di avvio servizio (messa in esercizio) dovrà essere definita considerando le tempistiche necessarie all'eventuale aggiornamento del metadata di regione Sardegna e alla sua approvazione da parte di AgID che saranno state valutate nella fase precedente. Il registro dovrà essere inviato con nota di accompagnamento, distinguendo se si tratta di integrazione di un nuovo servizio o di aggiornamento di un servizio già integrato. A tal fine è predisposto un esempio da utilizzarsi come riferimento per i contenuti della nota di accompagnamento (vedasi allegati).
Messa in esercizio del servizio integrato	Inserimento del nuovo servizio nell'elenco dei servizi pubblicato nella pagina https://www.regione.sardegna.it/servizispid/	Titolare del servizio/sistema che è stato integrato con l'AM-RAS	Invio di una richiesta (Ticket) a supporto portali di Sardegna IT supporto.portali@sardegnait.it	In conformità con gli altri brevi pubblicati su https://www.regione.sardegna.it/servizispid/ , la richiesta di inserimento del nuovo servizio nell'elenco indicato dovrà contenere: - un titoletto (nome del servizio) e il link all'applicativo (lo stesso indicato nel registro); - un breve (max 500 caratteri inclusi spazi); - l'indicazione della tipologia di utente che accede al sistema mediante autenticazione (cittadini/operatori, entrambi); - il Livello SPID associato, come riportato nel registro. Se non ancora in esercizio alla data di invio della richiesta, dovrà essere riportata l'indicazione di mantenere in stand by la pubblicazione nell'elenco sino ad avvenuto rilascio in produzione che sarà riscontrato internamente dal supporto tecnico di Sardegna IT agli operatori del portale istituzionale. **N.B.** Seguire la stessa modalità di richiesta sia per l'aggiornamento dei dati di servizi già integrati che per la cancellazione.
	Aggiornamento pagina https://www.regione.sardegna.it/servizispid/	Sardegna IT	Esecuzione ticket	Ad esecuzione avvenuta, segue relativa informativa via email a Titolare del servizio/sistema e a Referente Regionale per lo SPID/RUP della Convenzione

Tabella 8 - Sinossi della procedura Tecnica e Amministrativa

6 Appendice

6.1 XML Spring Security

```
<?xml version="1.0" encoding="UTF-8" ?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:sec="http://www.springframework.org/schema/security"
```

```
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:jee="http://www.springframework.org/schema/jee"
      xmlns:p="http://www.springframework.org/schema/p"
      xmlns:util="http://www.springframework.org/schema/util"
      xmlns:tx="http://www.springframework.org/schema/tx"
      xmlns:aop="http://www.springframework.org/schema/aop"
      xmlns:mvc="http://www.springframework.org/schema/mvc"
      xmlns:context="http://www.springframework.org/schema/context"
      xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd
      http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-4.3.xsd
     http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-4.3.xsd
      http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.3.xsd
      http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.3.xsd
      http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd
      http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd
      http://www.springframework.org/schema/security http://www.springframework.org/schema/security/spring-security-4.1.xsd">

    <!-- Enable auto-wiring -->
    <context:annotation-config/>


    <!-- Scan for auto-wiring classes in spring saml packages -->
    <context:component-scan base-package="org.springframework.security.saml"/>


    <!-- Unsecured pages -->
    <sec:http pattern="/favicon.ico" security="none"/>
    <sec:http pattern="/img/**" security="none"/>
    <sec:http pattern="/js/**" security="none"/>
    <sec:http pattern="/js-i18n/**" security="none"/>
    <sec:http pattern="/css/**" security="none"/>
    <sec:http pattern="/ckeditor/**" security="none"/>
    <sec:http pattern="/login.htm**" security="none"/>
    <sec:http pattern="/redirect.jsp" security="none"/>

    <!-- Security for the administration UI -->
    <sec:http pattern="/saml/web/**" use-expressions="false">
        <sec:access-denied-handler error-page="/saml/web/metadata/login"/>
        <sec:form-login login-processing-url="/saml/web/login" login-page="/saml/web/metadata/login" default-target-url="/saml/web/metadata"/>
        <sec:intercept-url pattern="/saml/web/metadata/login" access="IS_AUTHENTICATED_ANONYMOUSLY"/>
        <sec:intercept-url pattern="/saml/web/**" access="ROLE_ADMIN"/>
        <sec:custom-filter before="FIRST" ref="metadataGeneratorFilter"/>
    </sec:http>

    <!-- Secured pages with SAML as entry point -->
    <sec:http entry-point-ref="samlEntryPoint" use-expressions="false" >
        <sec:csrf disabled="true"/>
```

```
        <sec:intercept-url pattern="/SAMLExample/**" access="ROLE_LOGGED"/>
        <sec:custom-filter before="FIRST" ref="metadataGeneratorFilter"/>
        <sec:custom-filter after="BASIC_AUTH_FILTER" ref="samlFilter"/>
        <sec:custom-filter position="CONCURRENT_SESSION_FILTER" ref="concurrencyFilter" />
    </sec:http>


    <!-- Filters for processing of SAML messages -->
    <bean id="samlFilter" class="org.springframework.security.web.FilterChainProxy">
        <sec:filter-chain-map request-matcher="ant">
            <sec:filter-chain pattern="/saml/login/**" filters="samlEntryPoint"/>
            <sec:filter-chain pattern="/saml/logout/**" filters="samlLogoutFilter"/>
            <sec:filter-chain pattern="/saml/metadata/**" filters="metadataDisplayFilter"/>
            <sec:filter-chain pattern="/saml/SSO/**" filters="samlWebSSOProcessingFilter"/>
            <sec:filter-chain pattern="/saml/SSOHoK/**" filters="samlWebSSOHoKProcessingFilter"/>
            <sec:filter-chain pattern="/saml/SingleLogout/**" filters="samlLogoutProcessingFilter"/>
            <sec:filter-chain pattern="/saml/discovery/**" filters="samlIDPDiscovery"/>
        </sec:filter-chain-map>
    </bean>

    <!-- Handler deciding where to redirect user after successful login -->
    <bean id="successRedirectHandler" class="org.springframework.security.web.authentication.SavedRequestAwareAuthenticationSuccessHandler">
        <property name="defaultTargetUrl" value="/"/>
    </bean>

    <!-- Handler deciding where to redirect user after failed login -->
    <bean id="failureRedirectHandler"
class="org.springframework.security.web.authentication.SimpleUrlAuthenticationFailureHandler">
        <property name="useForward" value="true"/>
        <property name="defaultFailureUrl" value="/error.jsp"/>
    </bean>


    <!-- Handler for successful logout -->
    <bean id="successLogoutHandler" class="org.springframework.security.web.authentication.logout.SimpleUrlLogoutSuccessHandler">
        <property name="defaultTargetUrl" value="/"/>
    </bean>
    <sec:authentication-manager alias="authenticationManager">
        <!-- Register authentication manager for SAML provider -->
        <sec:authentication-provider ref="samlAuthenticationProvider"/>
        <!-- Register authentication manager for administration UI -->
        <sec:authentication-provider>
            <sec:user-service id="adminInterfaceService">
                <sec:user name="admin" password="admin" authorities="ROLE_ADMIN"/>
            </sec:user-service>
        </sec:authentication-provider>
```

```
    </sec:authentication-manager>

    <!-- Logger for SAML messages and events -->
    <bean id="samlLogger" class="org.springframework.security.saml.log.SAMLDefaultLogger"/>

    <!-- Central storage of cryptographic keys -->
    <bean id="keyManager" class="org.springframework.security.saml.key.JKSKeyManager">
        <constructor-arg value="path keystore firma"/>
        <constructor-arg type="java.lang.String" value="password keystore firma"/>
        <constructor-arg>
            <map>
                <entry key="alias chiave firma" value="password chiave firma"/>
            </map>
        </constructor-arg>
        <constructor-arg type="java.lang.String" value="alias chiave firma"/>
    </bean>

    <!-- Entry point to initialize authentication, default values taken from properties file -->
    <bean id="samlEntryPoint" class="org.springframework.security.saml.SAMLEntryPoint">
        <property name="defaultProfileOptions">
            <bean class="org.springframework.security.saml.websso.WebSSOProfileOptions">
                <property name="includeScoping" value="true"/>
                <property name="assertionConsumerIndex" value="1"/>
                <property name="allowCreate" value="true"/>
                <property name="nameID" value="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"/>
                <property name="binding" value="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"/>
                <property name="relayState" value="relaystate_${springsec.hostedSPName}"/>
                <property name="authnContexts">
                    <list>
                        <value>urn:oasis:names:tc:SAML:2.0:ac:classes:SpidL1</value>
<!--                        <value>urn:oasis:names:tc:SAML:2.0:ac:classes:SpidL2</value>
                        <value>urn:oasis:names:tc:SAML:2.0:ac:classes:SpidL3</value>-->
                    </list>
                </property>
            </bean>
        </property>
    </bean>

    <!-- IDP Discovery Service -->
    <bean id="samlIDPDiscovery" class="org.springframework.security.saml.SAMLDiscovery">
        <property name="idpSelectionPath" value="/WEB-INF/security/idpSelection.jsp"/>
    </bean>

    <!-- Filter automatically generates default SP metadata -->
    <bean id="metadataGeneratorFilter" class="org.springframework.security.saml.metadata.MetadataGeneratorFilter">
```

```
        <constructor-arg>
            <bean class="org.springframework.security.saml.metadata.MetadataGenerator">
                <property name="extendedMetadata">
                    <bean class="org.springframework.security.saml.metadata.ExtendedMetadata">
                        <property name="idpDiscoveryEnabled" value="true"/>
                    </bean>
                </property>
            </bean>
        </constructor-arg>
    </bean>

    <!-- The filter is waiting for connections on URL suffixed with filterSuffix and presents SP metadata there -->
    <bean id="metadataDisplayFilter" class="org.springframework.security.saml.metadata.MetadataDisplayFilter"/>

    <!-- Configure HTTP Client to accept certificates from the keystore for HTTPS verification -->
    <!--
    <bean class="org.springframework.security.saml.trust.httpclient.TLSProtocolConfigurer">
        <property name="sslHostnameVerification" value="default"/>
    </bean>
    -->

    <!-- IDP Metadata configuration - paths to metadata of IDPs in circle of trust is here -->
    <bean id="metadata" class="org.springframework.security.saml.metadata.CachingMetadataManager">
        <constructor-arg>
            <list>
                <bean class="org.springframework.security.saml.metadata.ExtendedMetadataDelegate">
                    <constructor-arg>
                        <bean class="org.opensaml.saml2.metadata.provider.FilesystemMetadataProvider">
                            <constructor-arg>
                                <value type="java.io.File">path file metadata servizio</value>
                            </constructor-arg>
                            <property name="parserPool" ref="parserPool"/>
                        </bean>
                    </constructor-arg>
                    <constructor-arg>
                        <bean class="org.springframework.security.saml.metadata.ExtendedMetadata">
                        </bean>
                    </constructor-arg>
                </bean>
                <bean class="org.opensaml.saml2.metadata.provider.HTTPMetadataProvider">
                    <constructor-arg>
                        <value type="java.lang.String">indirizzo metadati access manager</value>
                    </constructor-arg>
                    <constructor-arg>
                        <value type="int">5000</value>
```

```
                    </constructor-arg>
                    <property name="parserPool" ref="parserPool"/>
                </bean>
            </list>
        </constructor-arg>
        <!-- OPTIONAL used when one of the metadata files contains information about this service provider -->
        <property name="hostedSPName" value="entityID servizio"/>
        <!-- OPTIONAL property: can tell the system which IDP should be used for authenticating user by default. -->
        <property name="defaultIDP" value="accessmanager.regione.sardegna.it"/>
    </bean>

    <!-- SAML Authentication Provider responsible for validating of received SAML messages -->
    <bean id="samlAuthenticationProvider" class="org.springframework.security.saml.SAMLAuthenticationProvider">
        <!-- OPTIONAL property: can be used to store/load user data after login -->
        <property name="userDetails" ref="UserDetailsService" />
    </bean>

    <bean id="UserDetailsService" class="it.sardegnait.base.detailservice.CommonUserDetailsSSOService" />

    <!-- Provider of default SAML Context -->
    <bean id="contextProvider" class="org.springframework.security.saml.context.SAMLContextProviderImpl"/>

    <!-- Processing filter for WebSSO profile messages -->
    <bean id="samlWebSSOProcessingFilter" class="org.springframework.security.saml.SAMLProcessingFilter">
        <property name="authenticationManager" ref="authenticationManager"/>
        <property name="authenticationSuccessHandler" ref="successRedirectHandler"/>
        <property name="authenticationFailureHandler" ref="failureRedirectHandler"/>
    </bean>


    <!-- Processing filter for WebSSO Holder-of-Key profile -->
    <bean id="samlWebSSOHoKProcessingFilter" class="org.springframework.security.saml.SAMLWebSSOHoKProcessingFilter">
        <property name="authenticationManager" ref="authenticationManager"/>
        <property name="authenticationSuccessHandler" ref="successRedirectHandler"/>
        <property name="authenticationFailureHandler" ref="failureRedirectHandler"/>
    </bean>
    <!-- Logout handler terminating local session -->
    <bean id="logoutHandler"          class="org.springframework.security.web.authentication.logout.SecurityContextLogoutHandler">
        <property name="invalidateHttpSession" value="false"/>
    </bean>

    <!-- Override default logout processing filter with the one processing SAML messages -->
    <bean id="samlLogoutFilter" class="it.sardegnait.springframework.security.saml.SAMLLogoutFilter">
        <constructor-arg index="0" ref="successLogoutHandler"/>
        <constructor-arg index="1" ref="logoutHandler"/>
```

```
    <constructor-arg index="2" ref="logoutHandler"/>
</bean>

<!-- Filter processing incoming logout messages -->
<!-- First argument determines URL user will be redirected to after successful global logout -->
<bean id="samlLogoutProcessingFilter" class="it.sardegnait.springframework.security.saml.SAMLLogoutProcessingFilter">
    <constructor-arg index="0" ref="successLogoutHandler"/>
    <constructor-arg index="1" ref="logoutHandler"/>
</bean>

<!-- Class loading incoming SAML messages from httpRequest stream -->
<bean id="processor" class="org.springframework.security.saml.processor.SAMLProcessorImpl">
    <constructor-arg>
        <list>
            <ref bean="redirectBinding"/>
            <ref bean="postBinding"/>
            <ref bean="artifactBinding"/>
            <ref bean="soapBinding"/>
            <ref bean="paosBinding"/>
        </list>
    </constructor-arg>
</bean>

<!-- SAML 2.0 WebSSO Assertion Consumer -->
<bean id="webSSOprofileConsumer" class="org.springframework.security.saml.websso.WebSSOProfileConsumerImpl"/>

<!-- SAML 2.0 Holder-of-Key WebSSO Assertion Consumer -->
<bean id="hokWebSSOprofileConsumer" class="org.springframework.security.saml.websso.WebSSOProfileConsumerHoKImpl"/>

<!-- SAML 2.0 Web SSO profile -->
<bean id="webSSOprofile" class="org.springframework.security.saml.websso.WebSSOProfileImpl"/>

<!-- SAML 2.0 Holder-of-Key Web SSO profile -->
<bean id="hokWebSSOProfile" class="org.springframework.security.saml.websso.WebSSOProfileConsumerHoKImpl"/>
<!-- SAML 2.0 ECP profile -->
<bean id="ecpprofile" class="org.springframework.security.saml.websso.WebSSOProfileECPImpl"/>
<!-- SAML 2.0 Logout Profile -->
<bean id="logoutprofile" class="it.sardegnait.springframework.security.saml.websso.SingleLogoutProfileImpl"/>

<!-- Session Registry -->
<bean id="sessionRegistry" class="org.springframework.security.core.session.SessionRegistryImpl"/>

<bean id="concurrencyFilter" class="org.springframework.security.web.session.ConcurrentSessionFilter" >
    <constructor-arg ref="sessionRegistry"/>
```

```
    </bean>

    <!-- Bindings, encoders and decoders used for creating and parsing messages -->
    <bean id="postBinding" class="org.springframework.security.saml.processor.HTTPPostBinding">
        <constructor-arg ref="parserPool"/>
        <constructor-arg ref="velocityEngine"/>
    </bean>

    <bean id="redirectBinding" class="org.springframework.security.saml.processor.HTTPRedirectDeflateBinding">
        <constructor-arg ref="parserPool"/>
    </bean>

    <bean id="artifactBinding" class="org.springframework.security.saml.processor.HTTPArtifactBinding">
        <constructor-arg ref="parserPool"/>
        <constructor-arg ref="velocityEngine"/>
        <constructor-arg>
            <bean class="org.springframework.security.saml.websso.ArtifactResolutionProfileImpl">
                <constructor-arg>
                    <bean class="org.apache.commons.httpclient.HttpClient">
                        <constructor-arg>
                            <bean class="org.apache.commons.httpclient.MultiThreadedHttpConnectionManager"/>
                        </constructor-arg>
                    </bean>
                </constructor-arg>
                <property name="processor">
                    <bean class="org.springframework.security.saml.processor.SAMLProcessorImpl">
                        <constructor-arg ref="soapBinding"/>
                    </bean>
                </property>
            </bean>
        </constructor-arg>
    </bean>


    <bean id="soapBinding" class="org.springframework.security.saml.processor.HTTPSOAP11Binding">
        <constructor-arg ref="parserPool"/>
    </bean>
    <bean id="paosBinding" class="org.springframework.security.saml.processor.HTTPPAOS11Binding">
        <constructor-arg ref="parserPool"/>
    </bean>

    <!-- Initialization of OpenSAML library-->
    <bean class="org.springframework.security.saml.SAMLBootstrap"/>

    <!-- Initialization of the velocity engine -->
    <bean id="velocityEngine" class="org.springframework.security.saml.util.VelocityFactory" factory-method="getEngine"/>
```

```
    <!-- XML parser pool needed for OpenSAML parsing -->
    <bean id="parserPool" class="org.opensaml.xml.parse.StaticBasicParserPool" init-method="initialize">
        <property name="builderFeatures">
            <map>
                <entry key="http://apache.org/xml/features/dom/defer-node-expansion" value="false"/>
            </map>
        </property>
    </bean>

    <bean id="parserPoolHolder" class="org.springframework.security.saml.parser.ParserPoolHolder"/>

</beans>
```

6.2 Esempio chiamate OAuth2

6.2.1 Login

Prima chiamata da parte dell'applicativo client mediante browser in get:

```
GET https://accessmanager.demo.sardegnait.it/access-manager/oauth/authorize?client_id=oauth2client.debug&redirect_uri=http://localhost:8080/oauth2client/client_oauth2&response_type=code&scope=read&state=7OoS6r HTTP/1.1
Host: accessmanager.demo.sardegnait.it
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate, br
Referer: http://localhost:8080/oauth2client/index.html
Connection: keep-alive
Cookie: JSESSIONID=EEAC4BD8185262D77019D8A0A15A1980
Upgrade-Insecure-Requests: 1
```

Response Access Manager:

```
HTTP/1.1 302 302
Date: Thu, 09 Jul 2020 13:19:35 GMT
Set-Cookie: JSESSIONID=678B51FA10BABF3B57AF1029A2D4732E;path=/access-manager/;Secure;HttpOnly
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Location: /access-manager/saml/discovery?returnIDParam=idp&entityID=accessmanager.spid.regione.sardegna.it
Content-Length: 0
Access-Control-Allow-Origin: *
```

```
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/plain
```

Questa dà origine alla fase di autenticazione.

Finito il riconoscimento dell'utente Access manager rimanda all'applicazione client mediante redirect con verso l'indirizzo di callback:

```
HTTP/1.1 302 302
Date: Thu, 09 Jul 2020 13:19:40 GMT
Cache-Control: no-store
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Location: http://localhost:8080/oauth2client/client_oauth2?code=phHG2v&state=7OoS6r
Content-Language: it-IT
Content-Length: 0
Access-Control-Allow-Origin: *
Keep-Alive: timeout=5, max=95
Connection: Keep-Alive
Content-Type: text/plain
```

A questo punto l'applicazione client richiede il token :

```
POST https://accessmanager.demo.sardegnait.it/access-manager/oauth/token HTTP/1.1
Authorization: Basic b2F1dGgyY2xpZW50LmRlYnVnOjEyMzQ1Njc4OQ==
Accept: application/json, application/x-www-form-urlencoded
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache
Pragma: no-cache
User-Agent: Java/11.0.6
```

```
Host: accessmanager.demo.sardegnait.it
Connection: keep-alive
Content-Length: 115

grant_type=authorization_code&code=phHG2v&redirect_uri=http%3A%2F%2Flocalhost%3A8080%2Foauth2client%2Fclient_oauth2
```

Response Access Manager contenente il token:

```
HTTP/1.1 200 200
Date: Thu, 09 Jul 2020 13:19:41 GMT
Cache-Control: no-store
Pragma: no-cache
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Type: application/json;charset=UTF-8
Access-Control-Allow-Origin: *
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Length: 109

{"access_token":"2c43e666-fe4b-4e59-bc84-0615e2c184a3","token_type":"bearer","expires_in":179,"scope":"read"}
```

Attraverso il token l'applicazione richiede i dati dell'utente riconosciuto:

```
GET https://accessmanager.demo.sardegnait.it/access-manager/SSOServiceOAuth2 HTTP/1.1
Authorization: bearer 2c43e666-fe4b-4e59-bc84-0615e2c184a3
Accept: application/json, application/*+json
User-Agent: Java/11.0.6
Host: accessmanager.demo.sardegnait.it
Connection: keep-alive
```

E ottiene

```
HTTP/1.1 200 200
Date: Thu, 09 Jul 2020 13:19:42 GMT
Set-Cookie: JSESSIONID=39EDD282585D49F9E5F42EE890546665;path=/access-manager/;Secure;HttpOnly
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Type: application/json;charset=UTF-8
Access-Control-Allow-Origin: *
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Length: 432

{"attributeMap":{"placeOfBirth":"Cagliari","address":"Via bb111 11 00100 ROMA RM","gender":"M","dateOfBirth":"1970-05-
01","countyAddress":"RM","postalCodeAddress":"00100","mobilePhone":"3213434343","streetAddress":"Via
bb111","familyName":"Follesa","name":"Giorgio","cityAddress":"ROMA","countyOfBirth":"CA","fiscalNumber":"FLLGRG70E01H501Q","houseNumberAddress"
:"11","email":"test1234@sardegnait.it"},"username":"FLLGRG70E01H501Q"}
```

6.2.2 Logout

Richiesta Client

```
DELETE https://login.regione.sardegna.it/revoke-token HTTP/1.1
Authorization: bearer 5ee1ba0b-1c69-4416-a24c-ee0aceeee85a
Cache-Control: no-cache
Pragma: no-cache
User-Agent: Java/1.8.0_171
```

```
Host: login.regione.sardegna.it
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
```

Response Access Manager

```
HTTP/1.1 200 200
Date: Mon, 09 Sep 2019 12:55:50 GMT
Server: Apache/2.4.29 (Ubuntu)
Set-Cookie: JSESSIONID=5F956B123D4E2304F41BE18CF2BAE2B6.worker1;path=/;Secure;HttpOnly
Cache-Control: no-cache, no-store, must-revalidate, private
Pragma: no-cache
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Length: 0
Keep-Alive: timeout=5, max=98
Connection: Keep-Alive
```