DESCRIZIONE DELLE SPECIFICHE DI SICUREZZA NEGLI ACCORDI DI SERVIZIO

Versione 1.0

CNIPA Centro Nazionale per l'Informatica nella Pubblica Amministrazione

IBM SISTEMI INFORMATIVI An IBM Company

Nome doc.:	**SPCoop-Descrizione delle specifiche di sicurezza negli Accordi di Servizio_v1.0**	Versione:	**1.0**
Data emissione:	**5 Marzo 2009**	Stato:	**Pubblicato**

INDICE

1. PREFAZIONE

1.1. Autori

Redatto da:	Andrea Carmignani	RTI IBM-SI
Verificato da:	Nazzareno Ticconi	RTI IBM-SI
Revisione a cura di:	Stefano Fuligni	CNIPA
	Giovanni Olive	CNIPA
	Alfio Raia	CNIPA
	Alessandro Vinciarelli	CNIPA
Validato da:	Francesco Tortorelli	CNIPA

1.2. Modifiche Documento

Descrizione Modifica	Edizione	Data
Emissione 1^ bozza	0.1	01/06/2008
Inserimento Descrizione Requisiti di sicurezza	0.5	12/06/2008
Revisione contenuti	0.6	20/06/2008
Revisione contenuti	0.7	24/09/2008
Revisione ed Impaginazione finale	1.0	05/03/2009

1.3. Riferimenti

Codice	Titolo

1.4. Acronimi e Definizioni

Sigla	Descrizione
SAML	Security Assertion Markup Language
WSDL	Web Service Description Language
AdS	Accordo di Servizio

2. OBIETTIVI E CONTESTO DI RIFERIMENTO

Il *Sistema Pubblico di Connettività e Cooperazione (SPC)* si colloca nel contesto definito dal Decreto legislativo n° 82 del 7 marzo 2005, pubblicato in G.U. del 16 maggio 2005, n. 112, recante il "**Codice dell'amministrazione digitale**" (C.A.D.) e successive modifiche ed integrazioni. Esso istituisce il SPC, definendone gli obiettivi, le funzionalità ed il modello di governance.

Il processo di regolamentazione normativa del SPC è proseguito nel tempo, arrivando alla pubblicazione del Decreto del Presidente del Consiglio dei Ministri n.1 del 1 aprile 2008, pubblicato in G.U. del 21 giugno 2008, n. 144, recante le **"Regole tecniche e di sicurezza per il funzionamento del Sistema pubblico di connettivita"**, previste dall'art. 71, comma 1-bis, del C.A.D, con il quale viene definito il quadro tecnico di riferimento per lo sviluppo dei servizi SPC e le regole per il funzionamento e l'adesione ai servizi SPC.

Parallelamente, come previsto dal modello condiviso di cooperazione applicativa per la P.A. italiana *SPCoop*, è stato avviato e portato a termine lo sviluppo dei *Servizi Infrastrutturali di interoperabilità, cooperazione ed accesso (SICA)* e del centro di gestione per l'erogazione di tali servizi (CG-SICA), infrastruttura condivisa a livello nazionale che abilita l'interoperabilità e la cooperazione applicativa fra le Amministrazioni pubbliche nonché l'accesso ai servizi applicativi da queste sviluppati e resi disponibili su SPC.

L'evoluzione dello scenario di riferimento e la disponibilità di servizi di infrastruttura per la cooperazione applicativa, hanno reso necessaria la definizione e publicazione di una serie di documenti che specificassero in dettaglio le modalità tecniche per l'interoperabilità e la cooperazione applicativa e l'utilizzo dei servizi SICA, come peraltro prevista dalle succitate regole tecniche.

Gli ultimi documenti tecnici relativi al SPCoop rilasciati alla fine del 2005, infatti, definivano un livello di condivisione che consentiva sia la stabilità del modello nel tempo rispetto al contesto organizzativo e tecnologico di riferimento, sia i necessari gradi di libertà per la sua implementazione; ciò a scapito del dettaglio tecnico necessario, invece, nel momento in cui si fa riferimento ad una specifica implementazione del modello ed a specifici servizi infrastrutturali.

I seguenti documenti sono stati redatti dal Raggruppamento Temporaneo di Imprese (IBM-Sistemi Informativi), incaricato dello sviluppo e dell'implementazione del Centro di Gestione dei servizi SICA, con la supervisione del CNIPA, ed hanno origine dalla documentazione sviluppata nel corso del progetto e nella fase di collaudo dei servizi stessi.

L'insieme dei documenti prodotti specifica i modelli, le modalità, i dettagli tecnici di realizzazione, di gestione ed di utilizzo dei servizi SICA, le modalità di interfacciamento, le procedure qualificazione e gestione dei componenti infrastrutturali SPCoop, sulla base di quanto già previsto e definito nei documenti precedentemente condivisi e nel rispetto delle succitate regole tecniche.

Titolo Documento
1. Introduzione ai servizi SICA
2. Specifiche di nomenclatura in SPCoop
3. Specifiche di utilizzo del Servizio di Registro SICA
4. Modalità di funzionamento del Client SICA
5. Struttura dell'Accordo di Servizio e dell'Accordo di Cooperazione
6. Descrizione delle specifiche di sicurezza negli Accordi di Servizio
7. Aspetti di sicurezza applicativa nella cooperazione fra servizi
8. Modalità di funzionamento del Catalogo Schemi e Ontologie
9. Interfacce applicative tra Registro SICA generale e Registri SICA secondari
10. Modalità di Qualificazione del Registro SICA secondario
11. Modalità di Qualificazione della Porta di Dominio
12. Schema d'interoperabilità IndicePA
13. Guida ai servizi IndicePA
14. Modello di Gestione Federata delle Identità Digitali (GFID)
15. Modalità di accreditamento alla GFID
16. Modello di funzionamento dell'Indice dei Soggetti
17. Modello di funzionamento della Certification Authority

2.1. Scopi del documento

Obiettivo del presente documento è la definizione di linee guida per la descrizione delle specifiche di Sicurezza negli accordi di servizio. Il documento si focalizza in particolare sulla definizione delle modalità di descrizione dei requisiti di sicurezza applicativa di un servizio erogato in ambito SPCoop e di pubblicazione degli stessi nell'Accordo di Servizio.

Per la descrizione delle modalità per il trasferimento delle informazioni di sicurezza applicativa dal dominio fruitore al dominio erogatore in una interazione SPCoop, al fine di consentire l'attuazione dal lato del dominio erogatore del processo di autorizzazione all'accesso al servizio richiesto si rimanda al documento

Per una descrizione di un possibile esempio di descrizione delle specifiche di sicurezza relative all'infrastruttura si rimanda all'allegato "Allegato A: Esempi di Specifiche di Sicurezza per l'infrastruttura"

2.2. Note di lettura del documento

Nella definizione dei requisiti, delle specifiche e delle regole descritte nei documenti precedentemente indicati sono utilizzate le parole chiave DEVE, NON DEVE, OBBLIGATORIO,

VIETATO, DOVREBBE, CONSIGLIATO, NON DOVREBBE, SCONSIGLIATO, POTREBBE, OPZIONALE che devono essere interpretate in conformità con [RFC2119]. In particolare:

- DEVE, OBBLIGATORIO significano che la definizione è un requisito assoluto, la specifica deve essere implementata, la consegna è inderogabile.
- DOVREBBE, CONSIGLIATO significano che in particolari circostanze possono esistere validi motivi per ignorare un requisito, non implementare una specifica, derogare alla consegna, ma che occorre esaminare e valutare con attenzione le implicazioni correlate alla scelta.
- PUÒ, OPZIONALE significano che un elemento della specifica è a implementazione facoltativa.
- NON DOVREBBE, SCONSIGLIATO significano che in particolari circostanze possono esistere validi di motivi per cui un elemento di specifica è accettabile o persino utile, ma, prima di implementarlo, le implicazioni correlate dovrebbero essere esaminate e valutate con attenzione.
- NON DEVE, VIETATO significano che c'e proibizione assoluta di implementazione di un determinato elemento di specifica.

2.3. Note sul Copyright

3. TECNOLOGIE DI RIFERIMENTO

Obiettivo di questa sezione è illustrare e commentare le principali tecnologie su cui nel seguito si basano le specifiche di sicurezza degli accordi si servizio.

3.1. Security Assertion MarkUp Language (SAML)

Lo standard OASIS **Security Assertion Markup Language** (SAML) [1] ha l'obiettivo di fornire una sintassi per lo scambio di informazioni di sicurezza tra entità interagenti online basata su XML e le relative regole di interpretazione. Tali informazioni sono costituite da *asserzioni* scambiate tra *asserting party*, (le entità che le emettono) e *relying party* (le entità che le richiedono e che possono farne uso per gli scopi di autenticazione e autorizzazione).

La specifica SAML (giunta alla versione 2.0), è organizzata in modo modulare, e si basa sui alcuni "componenti" basilari quali:

- **asserzioni**: permettono di trasferire informazioni relative all'autenticazione e all'autorizzazione di utenti. La struttura delle asserzioni SAML è definita in [2], sez. 2, e nei documenti correlati (in particolare la specifica per la descrizione del contesto di autenticazione [3]).
- **Protocolli**: le asserzioni devono poter essere opportunamente scambiate tra le entità che interagiscono (per esempio un fornitore di servizi e un certificatore di identità). Per questo sono stati definiti opportuni protocolli per lo scambio di asserzioni, che avviene usualmente mediante un meccanismo di richiesta e risposta. I protocolli SAML sono definiti in [2], sez. 3.
- **Binding**: i protocolli stabiliscono la struttura delle informazioni che possono essere scambiate ma non determinano le specifiche modalità di trasporto. Per questo sono stati definiti opportuni *binding* che indicano come realizzare effettivamente in SAML lo scambio di informazioni di sicurezza attraverso determinati protocolli di trasporto (per esempio HTTP o SOAP). I binding SAML sono descritti in [5].
- **Profili**: un profilo identifica una particolare combinazione di tipologie di asserzioni, protocolli e binding SAML atta a supportare un determinato caso d'uso ritenuto rilevante per l'interazione online tra entità. In [4], sez. 4, si trova la descrizione dei principali profili SAML che riguardano principalmente l'interazione con l'utente via browser web.

Inoltre, la specifica SAML prevede opportuni meccanismi di estendibilità (descritti in dettaglio caso per caso nei documenti citati) così come l'uso di *metadati*, cioè costrutti aventi una struttura standardizzata per la descrizione di alcune informazioni caratterizzanti le entità SAML a supporto delle interazioni tra loro.

3.2. WS-Security

Lo standard OASIS **Web Services Security: SOAP Message Security** (WS-Security o WSS) definisce una serie di estensioni al protocollo SOAP mirate a garantire la protezione end-to-end dei messaggi scambiati nelle interazioni tra Web services. In particolare, i tre principali meccanismi forniti dallo standard (giunto alla versione 1.1) sono:

- la possibilità di veicolare **token di sicurezza** come parte di un messaggio SOAP,
- la tutela dell'**integrità** del messaggio SOAP,
- la tutela della **confidenzialità** del messaggio SOAP.

Il concetto alla base di WS-Security è quello di *security token*. I security token sono costrutti che possono essere usati all'interno dei messaggi SOAP (in particolare nell'header) per rappresentare e veicolare informazioni (di identità, oppure relative ai meccanismi di processing da eseguire per validare i token stessi). Lo standard permette di usare diversi tipi di token: binari, basati su XML (tra cui rientrano le asserzioni SAML), criptati e altri ancora.

Per la protezione di integrità e confidenzialità, WS-Security utilizza standard come XML-Signature e XML-Encription, applicandoli a determinate parti del messaggio SOAP.

Lo standard WS-Security fornisce meccanismi generali ed estendibili per la protezione dei messaggi SOAP. Per questo, WS-Security è alla base di diverse altre specifiche legate alla sicurezza nell'ambito Web service.

3.3. WS-Policy

La specifica **Web Services Policy** (WS-Policy) fornisce un linguaggio per la descrizione machine-readable dei requisiti e delle caratteristiche di un Web service. La specifica definisce in particolare una grammatica di base con cui poter dichiarare e combinare requisiti e caratteristiche espressi sotto forma di **policy**. Una policy può ad esempio descrivere i requisiti di sicurezza (autenticazione, autorizzazione, confidenzialità ecc.) che un client deve soddisfare per poter accedere a un servizio, oppure la messa in atto da parte del servizio di determinati meccanismi come il tracciamento dei messaggi applicativi. Per esempio, tramite una policy WS-Policy un Web service può comunicare di essere in grado di accettare asserzioni SAML 2.0.

WS-Policy non entra nel merito della rappresentazione delle specifiche caratteristiche di un servizio: a tal fine esistono diverse altre specifiche che illustrano e dettagliano le modalità di descrizione di policy relative a determinati ambiti applicativi (per esempio WS-SecurityPolicy, WS-SecureConversation e WS-ReliableMessaging).

WS-Policy in sé non specifica neanche il modo in cui le policy possono essere associate a un Web service: a tal fine esiste la specifica WS-PolicyAttachment, che indica i meccanismi per legare le policy di un servizio alla relativa descrizione WSDL, a diversi livelli di scope.

La specifica WS-Policy, giunta alla versione 1.5, è pensata per essere generale ed estendibile.

4. DESCRIZIONE DEI REQUISITI DI SICUREZZA APPLICATIVA NELL'ACCORDO DI SERVIZIO SPCOOP

In ambito SPCoop la descrizione delle caratteristiche di sicurezza di un servizio è pubblicata nell'Accordo di Servizio (cfr. [15], sez. 8). La versione attuale della specifica dell'Accordo di Servizio SPCoop elenca le sezioni inerenti la sicurezza da considerare (identificazione, autenticazione, autorizzazione, integrità, confidenzialità ecc.) indicando alcune delle tecnologie utilizzabili al fine di descrivere tali requisiti (es: WS-Policy, WS-SecurityPolicy).

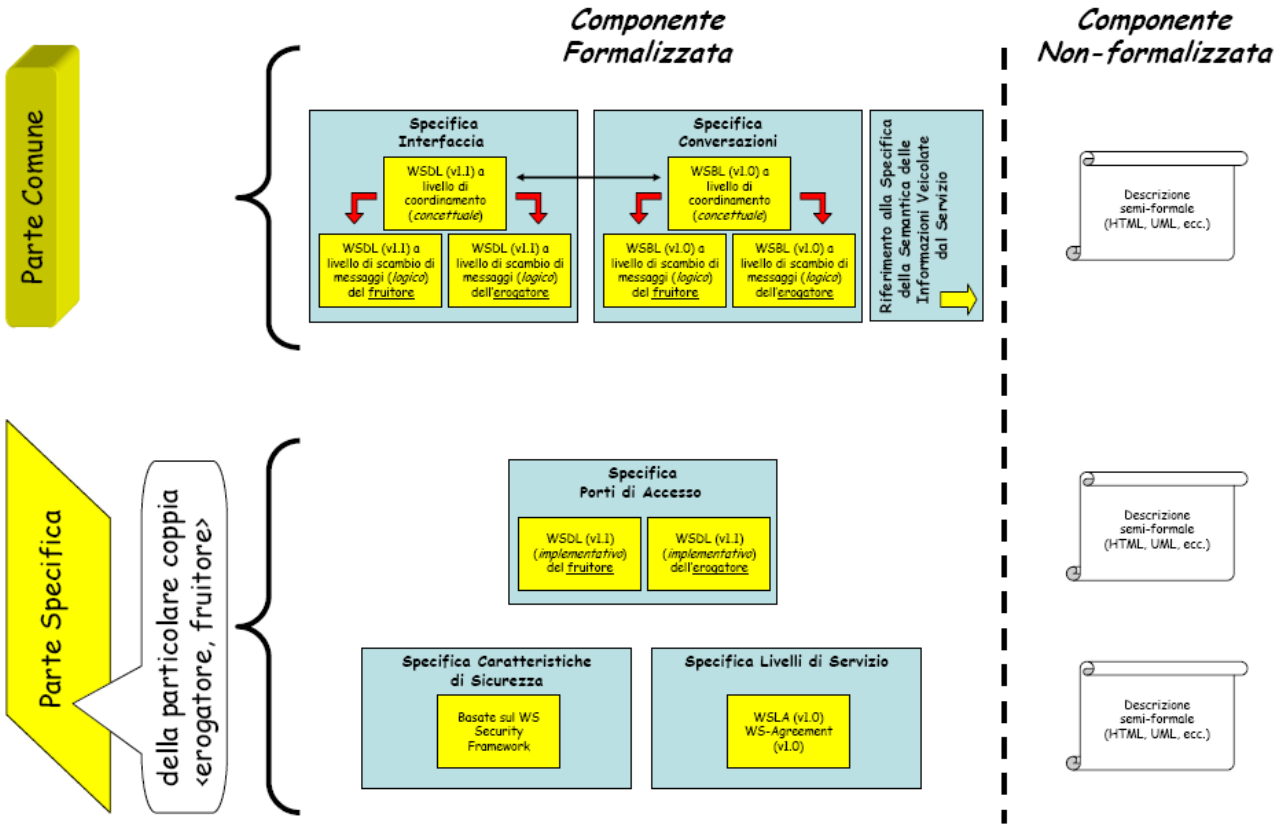

Figura 1. Struttura generale di un Accordo di Servizio SPCoop (cfr. [15])

In aggiunta, il documento sui servizi di sicurezza SPCoop (cfr. [13], sez. 4.2) si occupa di elencare e analizzare i requisiti di sicurezza comportati dall'erogazione e fruizione di un servizio applicativo in cooperazione applicativa.

Obiettivo del documento è illustrare delle linee guida per la la descrizione formale e "machine-readable" dei requisiti di sicurezza di un servizio SPCoop, tenendo in considerazione la compatibilità con l'attuale specifica SPCoop (ad esempio le indicazioni sul formato dell'Accordo di Servizio).

La soluzione proposta si basa sull'utilizzo della specifica WS-Policy unitamente ad alcune delle specifiche a essa correlate (es. WS-PolicyAttachment e WS-SecurityPolicy, cfr. sez. **Errore. L'origine riferimento non è stata trovata.**). che permettono la descrizione di policy di un servizio.

Questa scelta,in linea con le indicazioni CNIPA (cfr. [15], sez. 8), trova ulteriore fondamento nelle seguenti considerazioni:

1. la specifica WS-Policy fornisce una struttura semplice, generale, standard ed estendibile per descrivere le policy di un servizio;
2. la specifica WS-PolicyAttachment fornisce una modalità codificata e sufficientemente flessibile per associare policy WS-Policy a elementi della descrizione WSDL (versione 1.1 e 2.0) di un servizio (l'Accordo di Servizio prevede già l'utilizzo di WSDL, a diversi livelli di dettaglio, per la descrizione dei servizi SPCoop);
3. specifiche come WS-SecurityPolicy mettono a disposizione modalità consolidate ed estendibili per descrivere diversi aspetti dei requisiti di sicurezza di un servizio all'interno di policy WS-Policy.

Nel seguito di questa sezione vengono illustrate le caratteristiche di una policy basata su WS-Policy e il modo in cui essa si lega al WSDL di un servizio. Inoltre vengono illustrate le modalità con cui è possibile descrivere i requisiti di sicurezza applicativa di un servizio in ambito SPCoop per mezzo degli standard citati, complementandoli e integrandoli ove necessario.

Nel seguito si utilizza il prefisso fittizio `spcoopp`, "SPCoop Policy", per indicare un esempio di namespace in cui sono definiti i nuovi costrutti introdotti in ambito SPCoop.

4.1. Introduzione alle caratteristiche di una WS-Policy

Una **policy** WS-Policy può essere espressa mediante i costrutti base previsti dalla specifica: **`<wsp:Policy>`**, **`<wsp:ExactlyOne>`** e **`<wsp:All>`**, a cui in specifici casi si può aggiungere l'elemento **`<wsp:PolicyReference>`** (cfr. [19]) (il prefisso `wsp` indica il namespace previsto dalla specifica WS-Policy).

Gli elementi `<wsp:Policy>` e `<wsp:All>` sono equivalenti.

Un esempio di generica policy WS-Policy è il seguente:

```
<wsp:Policy xmlns:wsp="http://www.w3.org/ns/ws-policy">
 <wsp:All>
       [POLICY ASSERTION 1]
       [POLICY ASSERTION 2]
       <wsp:ExactlyOne>
             [POLICY ASSERTION 3]
             [POLICY ASSERTION 4]
       </wsp:ExactlyOne>
 </wsp:All>
</wsp:Policy>
```

Figura 2: Un esempio di WS-Policy

Una **policy assertion** rappresenta un requisito da soddisfare. Tutte le policy assertion contenute in un elemento `<wsp:All>` devono essere soddisfatte contemporaneamente, mentre in un elemento `<wsp:ExactlyOne>` si richiede che venga soddisfatta solamente una: ciò equivale a disporre di operatori "AND" e "OR" con cui combinare le policy assertion.

Nell'esempio descritto nella figura precedente, la policy richiede che siano soddisfatte insieme le policy assertion 1, 2 e 3 o 1, 2 e 4.

E' possibile che gli elementi `<wsp:ExactlyOne>` e `<wsp:All>` compaiano più volte in una policy ed essere annidati a piacere, permettendo così di esprimere requisiti complessi (ciò dipende anche da come è definita la struttura una certa policy assertion e dai punti di estensione che essa prevede, come spiegato nel seguito).

E' importante evidenziare che una policy assertion è espressa mediante elementi definiti da specifiche diverse da WS-Policy.

Nell'ambito WS-SecurityPolicy, per esempio, è definito l'elemento ***<sp:X509Token>*** (il prefisso sp indica il namespace previsto dalla specifica WS-SecurityPolicy) (cfr. [18], sez. 5.4.3): tale elemento, opportunamente inserito in una policy WS-Policy, costituisce una policy assertion, cioè un requisito (nello specifico relativo alla presenza di un token di sicurezza binario contenente un token X.509) che si vuole che venga rispettato per poter accedere al servizio. L'esempio che segue mostra una policy che fa uso della policy assertion appena descritta:

```
<wsp:Policy xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702">
      [...]
      <wsp:All>
```

```
            <sp:X509Token/>
            [...ALTRE POLICY ASSERTION...]
      </wsp:All>
      [...]
    </wsp:Policy>
```

Figura 3: Un esempio di utilizzo di WS-SecurityPolicy all'interno di una WS-Policy

Diverse policy assertion possono dunque essere combinate tra loro, poste in alternativa, annidate secondo le possibilità offerte dalla sintassi WS-Policy. L'annidamento di policy è particolarmente utile per dettagliare meglio il requisito legato a una determinata policy assertion, come nell'esempio che seguente:

```
    <wsp:Policy xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702">
      [...]
      <wsp:All>
            <sp:TransportToken>
                  <wsp:Policy>
                        <sp:HttpsToken>
                              <wsp:Policy />
                        </sp:HttpsToken>
                  </wsp:Policy>
            </sp:TransportToken>
      </wsp:All>
      [...]
    </wsp:Policy>
```

Figura 4: un esempio di policy strutturata.

L'esempio mostra che all'interno della policy assertion `<sp:TransportToken>` definita in WS-SecurityPolicy c'è la possibilità di inserire una **nested policy assertion** `<sp:HttpsToken>` che meglio precisa il requisito (richiedendo l'utilizzo di un ben preciso token di trasporto).

Analogamente, all'interno della policy assertion `<sp:HttpsToken>` la specifica prevede la possibilità di specificare ulteriori sotto-elementi relativi ad aspetti di autenticazione (per esempio `sp:HttpBasicAuthentication`, `sp:HttpDigestAuthentication` e `sp:RequireClientCertificate`) come nested policy. Nella policy di esempio questo ulteriore

livello di dettaglio non è richiesto, per questo è esplicitamente presente l'elemento `<wsp:Policy/>` (**empty policy**) che indica che in questo caso non ci sono requisiti addizionali relativamente alla policy assertion `<sp:HttpsToken>` (cfr. [19], sez. 2.9).

Mediante i costrutti WS-Policy è possibile referenziare policy definite esternamente (per esempio pubblicate e accessibili via web a un determinato URL, o contraddistinte da uno specifico URI) mediante l'elemento ***<wsp:PolicyReference>.***

È inoltre possibile specificare in un elemento di una policy assertion l'attributo ***wsp:Optional***, con valore "true", a indicare una capacità di supporto anziché un requisito: per esempio, in questo modo un servizio può segnalare nella propria policy di essere in grado di supportare un certo formato in input pur senza costringere i propri client ad adottarlo. Ecco un esempio in tal senso (frammento di policy):

```
<wsp:All>
 <sp:X509Token wsp:Optional="true" />
 [...]
</wsp:All>
```

Figura 5: un esempio di utilizzo dell'attributo wsp:Optional

Analogamente all'attributo `wsp:Optional`, è possibile specificare in un elemento di una policy assertion l'attributo ***wsp:Ignorable***, con valore "true", a indicare che quella policy assertion può essere ignorata.

Questa opzione può essere utilizzata a puro scopo informativo, per esempio per indicare nella policy che il servizio mette in atto una qualche forma di auditing (cosa che in sé non costituisce un vero e proprio requisito per l'accesso al servizio ma fornisce ai potenziali client un'informazione da poter valutare). Ecco un esempio in tal senso (frammento di policy; il namespace `log` in cui si suppone definita la policy assertion `Auditing` è fittizio):

```
<wsp:All>
 <log:Auditing wsp:Ignorable="true"/>
 [...]
</wsp:All>
```

Figura 6: Utilizzo dell'attributo wsp:ignorable

In generale, la definizione di una policy WS-Policy può prevedere opportuni punti di estendibilità.

4.2. Policy ed Accordo di Servizio SPCoop

Data o definita una policy WS-Policy, è necessario poterla associare opportunamente a un servizio. La specifica WS-PolicyAttachment descrive come associare una policy WS-Policy alla descrizione WSDL di un servizio. In particolare, nel caso di WSDL 1.1 l'associazione può avvenire a diversi livelli (scope):

- costrutto `service`,
- endpoint (costrutti `port`, `binding`, `portType`),
- costrutto `operation`,
- costrutto `message`.

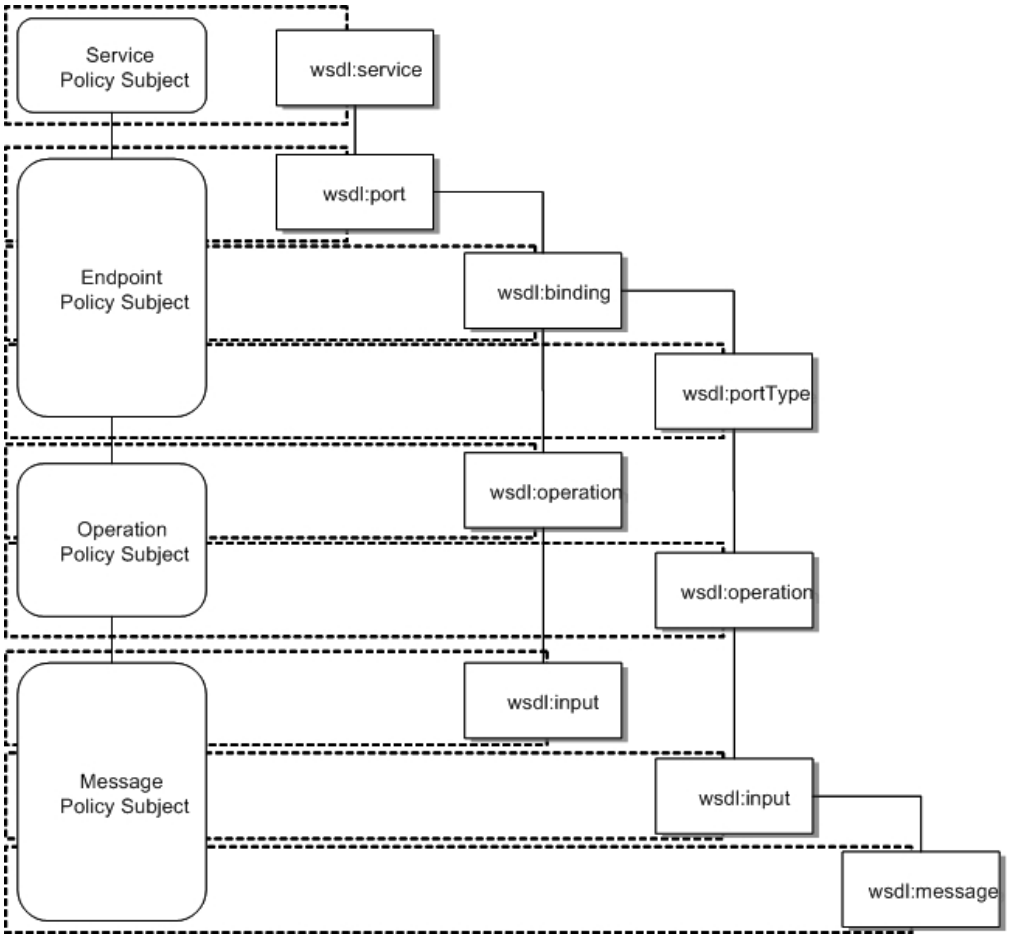

Figura 7. Possibili scope di una policy WS-Policy associata a un WSDL (cfr. [21])

A questi quattro diversi livelli, detti **policy subject**, le policy possono essere inserite direttamente nel WSDL oppure essere referenziate (mediante l'elemento `<wsp:PolicyReference>` oppure attraverso attributi `wsp:PolicyURIs`) (cfr. [21], sez. 4). In questo modo è possibile associare una (o più) policy WS-Policy alla descrizione WSDL del servizio al desiderato livello di granularità (le policy referenziate possono essere definite esternamente al file WSDL stesso).

Segue un esempio di associazione di una policy WS-Policy a una descrizione WSDL (adattato da [21]).

```
    <wsdl:definitions name="StockQuote"
targetNamespace="http://www.fabrikam123.example.com/stock/binding"
xmlns:tns="http://www.fabrikam123.example.com/stock/binding"
xmlns:fab="http://www.fabrikam123.example.com/stock"
xmlns:rmp="http://schemas.xmlsoap.org/ws/2005/02/rm/policy"
xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsoap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
      <wsp:Policy wsu:Id="X509EndpointPolicy">
            <sp:AsymmetricBinding>
                  <wsp:Policy>
                        <sp:IncludeTimestamp/>
                        <sp:OnlySignEntireHeadersAndBody/>
                  </wsp:Policy>
            </sp:AsymmetricBinding>
      </wsp:Policy>
      <wsp:Policy wsu:Id="SecureMessagePolicy">
            <sp:SignedParts>
                  <sp:Body/>
            </sp:SignedParts>
            <sp:EncryptedParts>
                  <sp:Body/>
            </sp:EncryptedParts>
      </wsp:Policy>
      <wsdl:binding name="StockQuoteSoapBinding" type="fab:Quote">
            <wsoap12:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
            <wsp:PolicyReference URI="#X509EndpointPolicy"
wsdl:required="true"/>
            <wsdl:operation name="GetLastTradePrice">
                  <wsoap12:operation
soapAction="http://www.fabrikam123.example.com/stock/Quote/GetLastTradePriceRequ
est"/>
                  <wsdl:input>
                        <wsoap12:body use="literal"/>
```

```
                        <wsp:PolicyReference URI="#SecureMessagePolicy"
wsdl:required="true"/>
                  </wsdl:input>
                  <wsdl:output>
                        <wsoap12:body use="literal"/>
                        <wsp:PolicyReference URI="#SecureMessagePolicy"
wsdl:required="true"/>
                  </wsdl:output>
            </wsdl:operation>
      </wsdl:binding>

     </wsdl:definitions>
```

In generale, la descrizione di una policy WS-Policy dovrebbe specificare a quali policy subject è associabile.

Dunque mediante l'uso della specifica WS-PolicyAttachment è possibile associare una (o più) policy WS-Policy alla descrizione WSDL 1.1 del servizio, per esempio una o più tra quelle contenute nell'Accordo di Servizio SPCoop.

L'approccio scelto sarà pertanto quello di inserire inline in uno (o più) dei WSDL previsti nell'Accordo di Servizio (logico, concettuale ecc. sia lato fruitore che lato erogatore) le opportune policy WS-Policy, ai diversi livelli di granularità possibili (service, endpoint ecc.).

Tale approccio permetterà la realizzazione di un WSDL autoconsistente, che potrà essere così oggetto di verifica sintattiche in fase di creazione dell'accordo di servizio (nel caso si utilizzi il Client SICA) o in fase di pubblicazione dell'accordo stesso nel Registro SICA Generale.

Quanto all'associazione delle policy ai WSDL attualmente previsti dall'Accordo di Servizio SPCoop, essa a seconda dei casi può avvenire verso i WSDL concettuali e/o logici (contenenti gli elementi `message`, `portType` e relative `operation`) presenti nella parte comune, o direttamente verso i WSDL implementativi (contenenti in più gli elementi `binding`) presenti nella parte specifica.

Riassumendo, le integrazioni all'attuale assetto dell'Accordo di Servizio potrebbero consistere in:

- per la parte specifica:
  - file WSDL implementativo erogatore corredato da policy;
  - file WSDL implementativo fruitore corredato da policy;

4.3. Descrizione dei Requisisti di Sicurezza di un Servizio

La specifica WS-Policy fornisce un framework per l'espressione e la combinazione di policy ma non si occupa di definire le specifiche policy assertion adatte ai diversi scopi.

Per questo esistono varie proposte, più o meno consolidate e recenti, che definiscono la struttura e la semantica delle policy assertion più adatte nell'ambito di determinati domini applicativi: WS-SecurityPolicy, WS-ReliableMessagingPolicy, WS-SecureConversation e altre ancora (in alcuni casi si tratta di specifiche più generali i cui costrutti possono costituire o servire a costruire policy assertion).

Ai fini della stesura delle presenti linee guida è stata condotta un'analisi dello stato dell'arte con l'obiettivo di identificare i principali costrutti già esistenti e "riusabili" ai fini della descrizione dei requisiti di sicurezza di un servizio SPCoop.

Tale analisi ha portato come risultato a identificare ed utilizzare un nucleo base di policy assertion già definite da specifiche consolidate (prima fra tutti **WS-SecurityPolicy**) e alla definizione di alcune policy assertion "ad hoc" per poter soddisfare le esigenze di descrizione dei requisiti in ambito SPCoop.

La definizione di nuove policy assertion ad hoc è stata considerata anche quale opzione semplificativa e nei casi in cui non vi poteva essere ragionevole confidenza circa l'adeguatezza dell'uso di costrutti esistenti.

E' importante sottolineare che, data la complessità del contesto tecnologico in cui ci si è mossi, non si può escludere a priori che alcuni requisiti di sicurezza di un servizio SPCoop possano essere descritti in modo alternativo a quello illustrato nel presente documento, facendo uso di altri costrutti esistenti o ad hoc. Analogamente, non si può escludere che possano nascere nei prossimi tempi specifiche contenenti policy assertion equivalenti a quelle introdotte.

Nello specifico, nel definire nuove policy ci si è attenuti in modo stringente alle linee guida indicate dal W3C (cfr. [22] W3C, Web Services Policy 1.5 – Guidelines for Policy Assertion Authors, W3C Working Group Note, 12 novembre 2007.).

Le policy assertion ad hoc che sono state definite possono rispondere, a seconda dei casi, a esigenze diverse: in alcune circostanze servono semplicemente a richiedere la presenza nel messaggio SOAP di un certo elemento (per esempio un sotto-elemento di una asserzione SAML 2.0), in altre pongono un vincolo più stringente in merito al contenuto (o al valore) di un sotto-elemento o attributo (per esempio il valore atteso del subject di una asserzione SAML).

Questa distinzione non risponde a un criterio generalizzabile, bensì supporta l'esigenza di "anticipare" in alcuni casi già a livello di policy determinati criteri di autorizzazione per l'accesso a un servizio SPCoop: un caso tipico in tal senso è rappresentato dalla dichiarazione dei valori ammessi per l'attributo di ruolo del soggetto che opera per conto dell'ente richiedente di un servizio in cooperazione applicativa.

L'approccio utilizzato per definire i requisiti di sicurezza per l'erogazione e la fruizione di un servizio applicativo SPCoop è rappresentato dal documento sui servizi di sicurezza (cfr. [13], sez. 4.2). riprendendo le caratteristiche di sicurezza previste in SPCoop (es: Integrità, confidenzialità,

Autenticazione ed Autorizzazione) e parte delle relative indicazioni da cui sono corredate, con l'obiettivo di utilizzarle quale spunto di partenza per valutare la completezza dell'approccio basato su policy WS-Policy.

4.3.1. Premessa su WS-Security Policy

La specifica WS-SecurityPolicy prevede la possibilità di usare opportune policy assertion per richiedere la presenza di un determinato elemento nell'header del messaggio SOAP, in particolare:

- **<sp:RequiredElements>**: contiene uno o più sottoelementi **<sp:XPath>** con cui indicare mediante sintassi XPath gli elementi che si richiede siano presenti nell'header (cfr. [18], sez. 4.3.1);
- **<sp:RequiredParts>**: contiene uno o più sottoelementi **<sp:Header>** con cui indicare gli header SOAP che si richiede siano presenti (cfr. [18], sez. 4.3.2).

Segue un esempio d'uso della policy assertion `<sp:RequiredElements>` (frammento di policy) che richiede la presenza dell'elemento `<xenc:ReferenceList>` nell'header del messaggio:

```
[...]
<sp:RequiredElements xmlns:sp="..." xmlns:wsse="..." xmlns:xenc="..." >
 <sp:xPath>/wsse:Security/xenc:ReferenceList</sp:xPath>
</sp:RequiredElements>
[...]
```

Figura 8: un esempio di utilizzo dell'elemento <sp:Requiredelements>

Questa è una modalità generale che permette, per mezzo di espressioni XPath, di descrivere dei requisiti su un qualunque elemento del messaggio SOAP purché relativo all'header, tuttavia questo approccio, oltre che parzialmente limitato, appare poco agevole per esprimere requisiti generali e articolati.

Per questo motivo l'approccio utilizzato nel seguito fa uso, quando possibile, degli altri costrutti di base messi a disposizione da WS-SecurityPolicy o da altre specifiche

4.3.2. Integrità

Tra le regole per il controllo dell'integrità dettate dal documento sui servizi di sicurezza SPCoop vi sono i seguenti (cfr. [13], sez. 4.2.4):

- *Il controllo di integrità dei messaggi scambiati DEVE essere implementato attraverso gli standard OASIS WS-Security [WS-Security 2004], in particolare gli standard W3C XMLSignature [XML Signature].*
- *La firma prevista da XML-Signature DEVE essere inserita all'interno dell'header block* `<wsse:Security>` *utilizzando l'elemento* `<ds:Signature>`.
- *Tutti gli elementi* `<ds:Reference>` *contenuti come sottoelementi di* `<ds:Signature>` *DEVONO contenere il riferimento ad una parte del messaggio SPCoop.*

La tutela dell'integrità di un messaggio è legata essenzialmente a meccanismi di firma. La specifica WS-SecurityPolicy mette a disposizione opportune policy assertion per porre il requisito che un determinato elemento del messaggio SOAP sia firmato, in particolare:

- **`<sp:SignedParts>`:** indica le parti del messaggio da firmare (cfr. [18], sez. 4.1.1) e può contenere zero o più sottoelementi (l'assenza di sottoelementi indica che tutti gli header e il body del messaggio devono essere firmati):
  - **`<sp:Body>`:** l'intero body del messaggio deve essere firmato;
  - **`<sp:Header>`:** permette di indicare l'header che deve essere firmato; può comparire più volte, e mediante l'attributo opzionale `Name` e l'attributo `Namespace` è possibile precisare a quale header ci si riferisce;
  - **`<sp:Attachments>`:** tutti gli attachment (SOAP with Attachments) del messaggio devono essere firmati;
- **`<sp:SignedElements>`:** contiene uno o più sottoelementi `<sp:XPath>` con cui indicare mediante sintassi XPath gli elementi che si richiede siano firmati (cfr. [18], sez. 4.1.2).

Inoltre, la specifica WS-SecurityPolicy mette a disposizione una policy assertion (cfr. [18], sez. 7.1) per specificare i requisiti sugli algoritmi da utilizzare (scopo di firma), in particolare:

- **`<sp:AlgorithmSuite>`:** contiene una nested policy in cui indicare sottoelementi come `<sp:Basic128Rsa15>`, `<sp:Basic256Sha256>` ecc. ciascuno dei quali corrisponde a una determinata combinazione di proprietà quali "encryption", "digest", "symmetric key signature" ecc. (per ulteriori dettagli si rimanda a [18], sez. 6.1).

In aggiunta è possibile usare la policy assertion **`<sp:Wss10>`** per specificare quali opzioni WS-Security 1.0 si richiede siano supportate (cfr. [18], sez. 9.1). È prevista anche l'analoga policy assertion **`<sp:Wss11>`** relativa a WS-Security 1.1 (cfr. [18], sez. 9.2).

Segue un esempio (frammento di policy) in cui si fa uso di alcune delle policy assertion appena citate.

```
    [...]
    <wsp:Policy>
     <sp:RequiredElements xmlns:sp="...">
           <sp:xPath>/wsse:Security/wsse:Signature</sp:xPath>
     </sp:RequiredElements>
     <sp:Wss10>
           <sp:MustSupportRefKeyIdentifier/>
           <sp:MustSupportRefEmbeddedToken/>
     </sp:Wss10>
     <sp:SignedParts>
           <sp:Body/>
           <sp:Header Name="Action"
Namespace="http://www.w3.org/2005/08/addressing"/>
           [...]
     </sp:SignedParts>
     <sp:AlgorithmSuite>
           <wsp:Policy>
                 <sp:Basic256/>
           </wsp:Policy>
     </sp:AlgorithmSuite>
     [...]
    </wsp:Policy>
    [...]
```

Figura 9: Esempio di utilizzo di policy assertion per definire i requisiti di integrità

Queste policy assertion si possono applicare ai seguenti policy subject:

- ***operation*** (valgono per tutti i message dell'operation);
- ***endpoint*** (valgono per tutti le operation dell'endpoint).

4.3.3. Confidenzialità

Il documento sui servizi di sicurezza SPCoop introduce una classificazione dei dati in ambito SPCoop che comprende cinque categorie (cfr. [13], sez. 4.2.5):

- R1: dato pubblico,
- R2: dato interno,
- R3: dato personale,
- R4: dato sensibile / dato giudiziario,
- R5: dato riservato.

Tra le regole per la riservatezza dettate dal documento sui servizi di sicurezza SPCoop abbiamo (cfr. [13], sez. 4.2.5):

- *I dati di livello R3, R4, R5 DEVONO essere cifrati ogni qualvolta sono scambiati attraverso messaggi SPCoop.*
- *I meccanismi di riservatezza a livello dei messaggi SPCoop DEVONO essere implementati attraverso gli standards OASIS WS-Security [WS-Security 2004], in particolare gli standards W3C XML-Encryption [XML-Encryption].*
- *Ogni certificato digitale emesso DEVE contenere gli elementi minimi previsti da WS-Security (Subject, Key Identifier, Serial Number ed Issuer) ed il profilo del certificato DEVE essere conforme al CPS previsto per lo scopo di certificazione.*
- *La cifratura prevista da XML-ENCRYPTION DEVE essere utilizzata all'interno del blocco* `<wsse:Security>` *del messaggio.*

La tutela della confidenzialità di un messaggio è legata essenzialmente a meccanismi di cifratura. La specifica WS-SecurityPolicy mette a disposizione opportune policy assertion per porre il requisito che un determinato elemento del messaggio SOAP sia cifrato, in particolare:

- **`<sp:EncryptedParts>`:** indica le parti del messaggio che devono essere cifrate (cfr. [18], sez. 4.2.1) e può contenere zero o più sottoelementi (l'assenza di sottoelementi indica che il body del messaggio deve essere cifrato):
  - **`<sp:Body>`**: l'intero body del messaggio deve essere cifrato;
  - **`<sp:Header>`:** permette di indicare l'header che deve essere cifrato; può comparire più volte, e mediante l'attributo opzionale `Name` e l'attributo `Namespace` è possibile precisare a quale header ci si riferisce;
  - **`<sp:Attachments>`**: tutti gli attachment (SOAP with Attachments) del messaggio devono essere cifrati;

- **<sp:EncryptedElements>:** contiene uno o più sottoelementi <sp:XPath> con cui indicare mediante sintassi XPath gli elementi che si richiede siano cifrati (cfr. [18], sez. 4.2.2).
- **<sp:ContentEncryptedElements>:** contiene uno o più sottoelementi <sp:XPath> con cui indicare mediante sintassi XPath gli elementi il cui contenuto si richiede che sia cifrato (cfr. [18], sez. 4.2.3).

In aggiunta, la specifica WS-SecurityPolicy mette a disposizione una policy assertion (cfr. [18], sez. 7.1) per specificare quali algoritmi possono essere utilizzati (in questo caso a scopo di cifratura), in particolare:

- **<sp:AlgorithmSuite>:** contiene una nested policy in cui indicare sottoelementi come **<sp:Basic128Rsa15>, <sp:Basic256Sha256>** ecc. ciascuno dei quali corrisponde a una determinata combinazione di proprietà quali "encryption", "digest", "symmetric key signature" ecc. (per ulteriori dettagli si rimanda a [18], sez. 6.1).

Segue un esempio (frammento di policy) in cui si fa uso di alcune delle policy assertion appena citate.

```
[...]
<wsp:Policy>
 <sp:EncryptedParts>
       <sp:Body/>
 </sp:EncryptedParts>
 <sp:AlgorithmSuite>
       <wsp:Policy>
             <sp:Basic256/>
       </wsp:Policy>
 </sp:AlgorithmSuite>
 [...]
</wsp:Policy>
[...]
```

Figura 10: Esempio di utilizzo di policy assertion per definire i requisiti di cifratura

Queste policy assertion si possono applicare ai policy subject operation (e in tal caso valgono per tutti i message dell'operation) ed endpoint (e in tal caso valgono per tutti le operation dell'endpoint).

4.3.4. Autenticazione e Identificazione

Tra le regole per l'autenticazione dettate dal documento sui servizi di sicurezza SPCoop vi sono le seguenti (cfr. [13], sez. 4.2.1):

- *I meccanismi di autenticazione a livello dei messaggi applicativi DEVONO essere implementati attraverso gli standards OASIS WS-Security [WS-Security 2004], in particolare lo standard W3C XML-Signature [XML Signature] e lo standard SAML V2.0, con l'uso esclusivo del X509 Token Profile [WSS TP X509].*
- *Per ogni messaggio da autenticare DEVE essere creato il contrassegno di sicurezza (security token), DEVE essere inserito nel messaggio e DEVE contenere informazioni relative alle credenziali dell'applicazione che richiede il servizio.*
- *DEVE essere presente l'elemento* `<wsse:SecurityTokenReference>` *all'interno dell'header* `<wsse:Security>`*, il subject Key Identifier (elemento* `<wsse:KeyIdentifier>`*), il binarySecurityToken, per rappresentare il certificato in formato X.509 v3 presente in binario all'interno di un elemento* `<wsse:BinarySecurityToken>`*, il serial number per identificare il certificato attraverso l'emittente.*

La specifica WS-SecurityPolicy definisce o cita diversi token di autenticazione da poter usare in una policy WS-Policy, quali per esempio:

- `<sp:BinarySecurityToken>` (definito da WS-Security);
- `<sp:UsernameToken>` (cfr. [18], sez. 5.4.1);
- `<sp:X509Token>` (cfr. [18], sez. 5.4.3);
- `<sp:KerberosToken>` (cfr. [18], sez. 5.4.4);
- `<sp:SamlToken>` (cfr. [18], sez. 5.4.8).

Nel seguito della sezione si approfondisce la policy assertion `<sp:SamlToken>` per valutarne l'adeguatezza a supporto dei meccanismi di trasferimento di informazioni di sicurezza applicativa nel contesto SPCoop basati su costrutti SAML.

Le principali caratteristiche definite per il Token SAML utilizzato in ambito SPCoop e descritte dalla struttura della policy assertion **`<sp:SamlToken>`** definita in WS-SecurityPolicy sono le seguenti:

- l'issuer del token, mediante un URI che ne rappresenti il nome logico (costrutto **<sp:IssuerName>**);
- elemento **<wst:Claims>** (mutuato da WS-Trust, il cui namespace è rappresentato dal prefisso `wst`) in cui indicare ciò che il token SAML deve asserire;
- elemento ***<sp:RequireKeyIdentifierReference>*** che permette di indicare, mediante nested policy, come deve essere referenziato il token SAML all'interno del messaggio SOAP. Nel caso SPC richiede l'uso dei costrutti previsti dal WS-Security SAML Token Profile 1.1 per i token SAML 2.0 (costrutto ***<sp:WssSamlV20Token11>***.
- attributo ***<sp:IncludeToken>*** (Opzionale), indica la modalità con cui si richiede che il token SAML sia incluso nel messaggio (per esempio può valere "AlwaysToRecipient" a indicare che deve essere incluso in tutti i messaggi spediti dal mittente al ricevente).

Per maggiori dettagli sulla sintassi e sull'uso del costrutto `<sp:SamlToken>` si rimanda alla specifica citata.

4.3.4.1. Elementi Opzionali Avanzati

La struttura base del costrutto `<sp:SamlToken>` definito in WS-SecurityPolicy permette di richiedere la presenza di un token avente determinate caratteristiche ma non sembra permettere di dettagliarne il contenuto in modo approfondito.

Una possibilità per entrare nel dettaglio del contenuto di un token SAML è sfruttare l'elemento ***<wst:Claims>*** per indicare ciò che il token SAML deve asserire[1]. Prendendo spunto dalla definizione di Claim base descritto all'interno di WS-Federation l'elemento **<wst:Claims>** ha la seguenti caratteristiche:

- prevede un attributo ***Dialect*** il cui valore è un URI che caratterizza la sintassi con cui esprimere i claim (nella specifica WS-Federation, per esempio, l'URI "http://schemas.xmlsoap.org/ws/2006/12/authorization/authclaims" indica i claim relativi ad autenticazione);
- può contenere più elementi ***<auth:ClaimType>,*** anche opzionali, contraddistinti da un attributo `Uri` che ne caratterizza il tipo (per esempio "http://schemas.xmlsoap.org/claims/EmailAddress");

[1] Le policy assertion ad hoc definite nel seguito potrebbero essere utilizzate a sé stante anche al di fuori dell'elemento `<wst:Claims>`.

  - ogni elemento **<auth:ClaimType>** può contenere elementi **<auth:Value>** per specificare il valore atteso per il claim;
- ciascun costrutto è pienamente estendibile.

Seguendo questo approccio diventa possibile esprimere requisiti sul contenuto del token SAML per mezzo dell'elemento **<wst:Claims>** e dei relativi elementi **<auth:ClaimType>** opportunamente definiti e valorizzati.

Per indicare che si tratta di requisiti che si riferiscono specificamente a una asserzione SAML 2.0, l'attributo `Dialect` dell'elemento `<wst:Claims>` presente all'interno dell'elemento `<sp:SamlToken>` assume il valore "urn:oasis:names:tc:SAML:2.0:assertion".

Occorre poi prevedere un tipo di `<auth:ClaimType>` per ciascun aspetto dell'asserzione SAML su cui si vuole descrivere. Per esempio:

- il subject del token SAML,
- le condizioni di validità del token SAML,
- la presenza di statement di autenticazione nel token SAML,
- la presenza di statement di attributo nel token SAML,
- la "forza" del meccanismo di autenticazione
- la presenza dell'elemento `<saml:Advice>` nel token SAML,

A tal fine, si introducono nuovi URI da usare come valori dell'attributo `Uri` di elementi `<auth:ClaimType>`, per esempio:

- "http://spcoop.gov.it/schemas/authn/saml/Subject",
- "http://spcoop.gov.it/schemas /authn/saml/Conditions",
- "http://spcoop.gov.it/schemas authn/saml/AuthnStatements",
- "http://spcoop.gov.it/schemas authn/saml/AttributeStatements",
- "http://spcoop.gov.it/schemas authn/saml/AuthnContext",
- "http://spcoop.gov.it/schemas/authn/saml/Advice",

In alcuni casi sarà sufficiente l'elemento `<auth:ClaimType>`, con il relativo attributo `Uri` opportunamente valorizzato, per esprimere il requisito. In altri casi occorrerà prevedere un sotto-elemento `<auth:ClaimValue>` per specificare il valore atteso per il claim.

Per esempio, il frammento di policy che segue esprime il requisito sulla presenza, all'interno del token SAML, di statement di autenticazione.

```
    [...]
    <sp:SupportingTokens>
     [...]
     <sp:SamlToken>
          <wst:Claims Dialect="urn:oasis:names:tc:SAML:2.0:assertion">
               <auth:ClaimType
Uri="http://www.spcoop.gov.it/schemas/authn/saml/AuthnStatements"/>
          </wst:Claims>
          [...]
     </sp:SamlToken>
     [...]
    </sp:SupportingTokens>
    [...]
```

Figura 11: Utilizzo dell'elemento <auth:ClaimType>

Per poter precisare più in dettaglio i valori che possono assumere gli specifici claim si può ricorrere invece che all'elemento <auth_ClaimType> a policy assertion definite ad hoc, come ad esempio:

- ***<spcoopp:Subject>:*** una policy assertion per il subject del token SAML,
- ***<spcoopp:Conditions>:*** una policy assertion per le condizioni di validità del token SAML;
- ***<spcoopp:AuthnContext>:*** una policy assertion per le "forza" del meccanismo di autenticazione utilizzato per ottenere il token SAML;
- altre policy assertion per richiedere la presenza di determinati costrutti nel token SAML.

4.3.4.1.1 Policy Assertion <spcoopp: Subject>

La policy assertion ***<spcoopp:Subject>*** può avere la seguente sintassi[2]:

[2] Per la definizione della sintassi delle policy assertion si utilizza la stessa notazione utilizzata nel documento di specifica WS-SecurityPolicy (cfr. [18], sez. 1.4.1). In estrema sintesi:
- "?" indica un elemento opzionale;

```
<spcoopp:Subject xmlns:spcoopp="..." ( wsp:Optional="true" )? ... >
        (
          <spcoopp:NameIDFormatUnspecified /> |
          <spcoopp:NameIDFormatEmailAddress /> |
          <spcoopp:NameIDFormatX509SubjectName /> |
          <spcoopp:NameIDFormatWindowsDomainQualifiedName /> |
          <spcoopp:NameIDFormatKerberosPrincipalName /> |
          <spcoopp:NameIDFormatEntityIdentifier /> |
          <spcoopp:NameIDFormatPersistentIdentifier /> |
          <spcoopp:NameIDFormatTransientIdentifier />
        ) ?
    <spcoopp:SubjectNameQualifier NameQualifier="xs:string" ... /> ?
        (
          <spcoopp:SubjectConfirmationMethodSaml20Bearer ... /> |
          <spcoopp:SubjectConfirmationMethodSaml20HolderOfKey ... /> |
          <spcoopp:SubjectConfirmationMethodSaml20SenderVouches ... />
        ) *
    <spcoopp:RequireSubjectConfirmationData ... /> ?
    ...
</spcoopp:Subject>
```

/spcoopp:Subject

Identifica la policy assertion relativa al subject del token SAML.

/spcoopp:Subject/spcoopp:NameIdFormat…

Elemento opzionale che specifica la policy relativa al NameID del subject del token SAML (i formati enumerati sono quelli previsti dalla specifica SAML 2.0).

/spcoopp:Subject/spcoopp:SubjectNameQualifier

- "*" indica una cardinalità di zero o più;
- "+" indica una cardinalità di uno o più;
- "|" indica un'alternativa;
- "(" e ")" indicano che gli elementi contenuti sono trattati come un gruppo rispetto a cardinalità o alternativa;
- "…" indica punti di estendibilità, cioè la possibilità di aggiungere attributi o altri elementi più o meno annidati senza violare la sintassi della policy assertion.

Elemento opzionale multiplo che specifica il requisito sul valore dell'attributo NameQualifier del NameID del subject del token SAML.

/spcoopp:Subject/spcoopp:SubjectNameQualifier/@NameQualifier

Attributo che specifica il valore atteso dell'attributo NameQualifier del NameID del subject del token SAML.

/spcoopp:Subject/spcoopp:SubjectConfirmationMethod…

Elemento opzionale multiplo che specifica il requisito sul metodo di conferma del subject del token, tra quelli previsti dalla specifica SAML 2.0.

/spcoopp:Subject/spcoopp:RequireSubjectConfirmationData

Elemento opzionale che specifica il requisito sulla presenza del sotto-elemento `<SubjectConfirmationData>` nel subject del token SAML.

La policy assertion `<spcoopp:Subject>` può essere utilizzata per precisare i claim di una policy assertion `<sp:SamlToken>` (anche più volte, nel caso in cui si vogliano indicare requisiti sulla presenza di più asserzioni con subject aventi determinate caratteristiche).

4.3.4.1.2 Policy Assertion <spcoop:Condiitons>

La policy assertion **`<spcoopp:Conditions>`** può avere la seguente sintassi:

```
<spcoopp:Conditions xmlns:spcoopp="..." ( wsp:Optional="true" )? ... >
 <spcoopp:NotBefore DateTime="xs:dateTime" ... /> ?
 <spcoopp:NotOnOrAfter DateTime="xs:dateTime" ... /> ?
 <spcoopp:RequireAudienceRestriction ... /> ?
 ...
</spcoopp:Conditions>
```

/spcoopp:Conditions

Identifica la policy assertion relativa alle condizioni di validità del token SAML.

/spcoopp:Conditions/spcoopp:NotBefore

Elemento opzionale che specifica il limite temporale inferiore di validità del token SAML.

/spcoopp:Conditions/spcoopp:NotBefore/@scpc:DateTime

Attributo che specifica il limite temporale inferiore di validità atteso del token SAML.

/spcoopp:Conditions/spcoopp:NotOnOrAfter

Elemento opzionale che specifica il limite temporale superiore di validità del token SAML.

/spcoopp:Conditions/spcoopp:NotOnOrAfter/@scpc:DateTime

Attributo che specifica il limite temporale inferiore di validità atteso del token SAML.

/spcoopp:Conditions/spcoopp:RequireAudienceRestriction

Elemento opzionale che specifica il vincolo sulla presenza del sotto-elemento `<AudienceRestriction>` nel token SAML.

La policy assertion `<spcoopp:Conditions>` può essere utilizzata per precisare i claim di una policy assertion `<sp:SamlToken>`.

Il frammento di policy che segue mostra un esempio di descrizione di requisiti sul subject e sui limiti di validità temporale del token SAML che fa uso delle due policy assertion descritte.

```
    [...]
    <sp:SupportingTokens>
     [...]
     <sp:SamlToken>
          <wst:Claims Dialect="urn:oasis:names:tc:SAML:2.0:assertion">
               <auth:ClaimType
Uri="http://www.cnipa.gov.it/schemas/authn/saml/Subject">
                    <auth:ClaimValue>
                         <spcoopp:Subject>
                              <spcoopp:NameIDFormatEmailUnspecified />
                              <spcoopp:SubjectNameQualifier
NameQualifier="http://idp.cnipa.gov.it/idp" />
                         </spcoopp:Subject>
                    </auth:ClaimValue>
               </auth:ClaimType>
               <auth:ClaimType
Uri="http://www.cnipa.gov.it/schemas/authn/saml/Conditions">
                    <auth:ClaimValue>
                         <spcoopp:Conditions>
                              <spcoopp:NotBefore DateTime="2008-04-
20T22:23:50.640Z" />
                              <spcoopp:NotOnOrAfter DateTime="2008-04-
21T15:11:50.640Z" />
                         </spcoopp:Conditions>
```

```
                    </auth:ClaimValue>
                </auth:ClaimType>
                [...]
            </wst:Claims>
            [...]
        </sp:SamlToken>
        [...]
    </sp:SupportingTokens>
    [...]
```

Figura 12: Esempio di utilizzo di policy assertion di tipo Subject e Conditions

4.3.4.1.3 Policy Assertion <Authncontext>

La policy assertion **<spcoopp:AuthnContext>** può avere la seguente sintassi:

```
    <spcoopp:AuthnContext xmlns:spcoopp="..." ( wsp:Optional="true" )? ... >
     <spcoopp:RequiredAuthnClass AuthnContextClassRef="xs:anyURI" ... /> *
     ...
    </spcoopp: AuthnContext >
```

/spcoopp:AuthnContext

Identifica la policy assertion relativa alla "forza" dell'autenticazione veicolata dal token SAML.

/spcoopp:RequiredAuthnClass

Elemento multiplo che specifica il vincolo sulla AuthnContextClass SAML attesa.

/spcoopp:RequiredAuthnClass/@AuthnContextClassRef

Attributo che specifica l'URI della AuthnContextClass SAML attesa (i valori ammessi sono quelli previsti dalla specifica SAML 2.0).

Il frammento di policy che segue esprime un requisito sui contesti di autenticazione attesi per il token SAML.

```
    [...]
```

```
    <spcoopp:AuthnContext>
     <spcoopp:RequiredAuthnClass
AuthnContextClassRef="urn:oasis:names:tc:SAML:2.0:ac:classes:Smartcard" />
     <spcoopp:RequiredAuthnClass
AuthnContextClassRef="urn:oasis:names:tc:SAML:2.0:ac:classes:SmartcardPKI" />
      [...]
    </spcoopp:Conditions>
    [...]
```

La policy assertion `<spcoopp:AuthnContext>` può essere utilizzata per precisare i claim di una policy assertion `<sp:SamlToken>`.

Lo schema XML che formalizza la sintassi delle policy assertion descritte è riportato in appendice

4.3.5. Attributi richiesti ai fini autorizzativi

Tra le regole per l'autorizzazione dettate dal documento sui servizi di sicurezza SPCoop vi sono le seguenti (cfr. [13], sez. 4.2.3):

- *I profili di autorizzazione POSSONO prevedere meccanismi gerarchici e POSSONO essere basati sui ruoli secondo i paradigmi RBAC (Role-Based Access Control).*
- *Lo scambio delle informazioni relative alle autorizzazioni DEVONO essere effettuate utilizzando lo standard SAML V 2.0, mediante chiavi crittografiche certificate a tale scopo da una o più infrastrutture a chiave pubblica accreditate e i cui certificati radice sono presenti nella TCL della Bridge CA.*

La specifica WS-SecurityPolicy non definisce specifiche policy assertion utilizzabili per esprimere policy di richiesta attributi a fini di autorizzazione. Per questo motivo si propone di introdurre una policy assertion **`<spcoopp:Attributes>`** .

La struttura della policy si può dividere in una parte base in grado di descrivere gli attributi necessari all'erogazione del servizio ed una avanzata con cui si può descrivere, se necessario, il mapping fra gli attributi ricevuti dal fruitore e quelli utilizzati dell'erogatore.

La policy assertion ha la seguente sintassi:

```
    <spcoopp:Attributes xmlns:spcoopp="..." ( wsp:Optional="true" )? ... >

        ( <spcoopp:Attribute Name="xs:string" ( NameFormat="..." )? ... >
```

```
            <spcoopp:AttributeValue ... /> ?
            ...
        </spcoopp:Attribute>  ) *

        ( <spcoopp:AttributeMapping ( wsp:Ignorable="true" )? ... >
          <spcoopp:SenderAttribute Name="xs:string" ( NameFormat="..." )? ... >
            <spcoopp:AttributeValue ... /> ?
            ...
          </spcoopp:SenderAttribute>
          <spcoopp:ReceiverAttribute Name="xs:string" ( NameFormat="..." )?
... >
            <spcoopp:AttributeValue ... /> ?
            ...
          </spcoopp:ReceiverAttribute>
        </spcoopp:AttributeMapping> ) *
      ...
    </spcoopp:Attributes>
```

dove gli elementi hanno il seguente significato:

/spcoopp:Attributes

Identifica la policy assertion relativa ai requisiti circa gli attribute statement presenti nel token SAML.

Parte Base

/spcoopp:Attributes/spcoopp:Attribute

Elemento multiplo che esprime il requisito su un singolo attributo atteso.

/spcoopp:Attributes/spcoopp:Attribute/@spcoopp:Name

Attributo che indica il nome atteso per l'attributo.

/spcoopp:Attributes/spcoopp:Attribute/@spcoopp:NameFormat

Attributo opzionale che indica il formato atteso per il nome dell'attributo.

/spcoopp:Attributes/spcoopp:Attribute/spcoopp:AttributeValue

Elemento opzionale che indica il valore atteso del nome dell'attributo.

Parte Avanzata

/spcoopp:Attributes/spcoopp:AttributeMapping

Elemento opzionale che descrive la corrispondenza tra i diversi nomi (e/o i corrispondenti valori) dati agli attributi tra chi invia e chi riceve il messaggio.

/spcoopp:Attributes/spcoopp:AttributeMapping/spcoopp:SenderAttribute

Elemento che descrive l'attributo per il mittente del messaggio.

/spcoopp:Attributes/spcoopp:AttributeMapping/spcoopp:SenderAttribute/@spcoopp:Name

Attributo che indica il nome dell'attributo per il mittente del messaggio.

/spcoopp:Attributes/spcoopp:AttributeMapping/spcoopp:SenderAttribute/@spcoopp:NameFormat

Attributo opzionale che indica il formato del nome dell'attributo per il mittente del messaggio.

/spcoopp:Attributes/spcoopp:AttributeMapping/spcoopp:SenderAttribute/spcoopp:AttributeValue

Attributo opzionale che indica il valore dell'attributo per il mittente del messaggio.

/spcoopp:Attributes/spcoopp:AttributeMapping/spcoopp:ReceiverAttribute

Elemento che descrive l'attributo per chi riceve il messaggio.

/spcoopp:Attributes/spcoopp:AttributeMapping/spcoopp:ReceiverAttribute/@spcoopp:Name

Attributo che indica il nome dell'attributo per chi riceve il messaggio.

/spcoopp:Attributes/spcoopp:AttributeMapping/spcoopp:ReceiverAttribute/@spcoopp:NameFormat

Attributo opzionale che indica il formato del nome dell'attributo per chi riceve il messaggio.

/spcoopp:Attributes/spcoopp:AttributeMapping/spcoopp:ReceiverAttribute/spcoopp:AttributeValue

Attributo opzionale che indica valore dell'attributo per chi riceve il messaggio.

Questa policy assertion può essere utilizzata all'interno di una policy assertion **`<sp:SamlToken>`**, per precisarne meglio i claim: a tal fine può comparire all'interno dell'elemento **`<wst:Claims>`** oppure dell'elemento **`<spcoopp:Subject>`**, qualora si volesse specificare a quale subject si riferiscono gli attributi.

Di seguito un esempio di policy relativa ad attributi utente (frammento di policy). Quanto esposto in figura 18 descrive l'utilizzo di entrambi le tipologie: *base*, per descrivere gli attributi attesi tipo ID utente,

Codice Fiscale e Cognome; *avanzata*, per fare il matching fra i ruoli inviati dal fruitore edi ruoli dell'erogatore.

Nell'esempio il ruolo inviato dal fruitore SPCoopISADM viene associato al ruolo interno dell'erogatore Admin_PA-SO (nell'esempio è il ruolo di amministratore per una data PA per l'indice dei soggetti). Similmente SPCoopRSADM viene associate con Admin_PA-PA (Amministratore PA per Amministrazioni in IPA) e SPCoopSQPDDADM con Admin_PA-QD (Amministratore PA per Servizio di Qualificazione Porta di Dominio)

```
    [...]
    <spcoopp:Attributes xmlns:spcoopp="http://www.cnipa.gov.it/schema"
xmlns:wsp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <spcoopp:Attribute Name="Codice_Amministrazione"/>
      <spcoopp:Attribute Name="ID_Utente"/>
      <spcoopp:Attribute Name="CodiceFiscale_Utente"/>
      <spcoopp:Attribute Name="Nome_Utente"/>
      <spcoopp:Attribute Name="Cognome_Utente"/>
      <spcoopp:Attribute Name="Ruolo"/>

      <spcoopp:AttributeMapping>
            <spcoopp:SenderAttribute Name="Ruolo">
                  <spcoopp:AttributeValue
xsi:type="xs:string">SPCoopISADM</spcoopp:AttributeValue>
            </spcoopp:SenderAttribute>
            <spcoopp:ReceiverAttribute Name="Ruolo">
                  <spcoopp:AttributeValue xsi:type="xs:string">Admin_PA-S
      O</spcoopp:AttributeValue>
            </spcoopp:ReceiverAttribute>
      </spcoopp:AttributeMapping>
      <spcoopp:AttributeMapping>
            <spcoopp:SenderAttribute Name="Ruolo">
                  <spcoopp:AttributeValue
xsi:type="xs:string">SPCoopRSADM</spcoopp:AttributeValue>
            </spcoopp:SenderAttribute>
            <spcoopp:ReceiverAttribute Name="Ruolo">
                  <spcoopp:AttributeValue xsi:type="xs:string">Admin_PA-
PA</spcoopp:AttributeValue>
```

```
            </spcoopp:ReceiverAttribute>
      </spcoopp:AttributeMapping>
      <spcoopp:AttributeMapping>
            <spcoopp:SenderAttribute Name="Ruolo">
                  <spcoopp:AttributeValue
xsi:type="xs:string">SPCoopSQPDDADM</spcoopp:AttributeValue>
            </spcoopp:SenderAttribute>
            <spcoopp:ReceiverAttribute Name="Ruolo">
                  <spcoopp:AttributeValue xsi:type="xs:string">Admin_PA-
QD</spcoopp:AttributeValue>
            </spcoopp:ReceiverAttribute>
      </spcoopp:AttributeMapping>
      <spcoopp:AttributeMapping wsp:Ignorable="true">
            <spcoopp:SenderAttribute Name="Ruolo">
                  <spcoopp:AttributeValue
xsi:type="xs:string">SPCoopSQRSAM</spcoopp:AttributeValue>
            </spcoopp:SenderAttribute>
            <spcoopp:ReceiverAttribute Name="Ruolo">
                  <spcoopp:AttributeValue xsi:type="xs:string">Admin_PA-
RG</spcoopp:AttributeValue>
            </spcoopp:ReceiverAttribute>
      </spcoopp:AttributeMapping>
    </spcoopp:Attributes>
    [...]
```

Figura 13: Esempio di utilizzo della policy assertion Attributes. Il frammento di policy riporta sia l'utilizzo della parte base e della parte avanzata

Lo schema XML che formalizza la sintassi della policy assertion appena descritta è riportato in appendice.

4.3.6. Tracciatura

Le regole per la registrazione degli eventi, l'ispezione e la tracciatura dettate dal documento sui servizi di sicurezza SPCoop riguardano solo le Porte di Dominio (cfr. [13], sez. 4.2.7) e non il comportamento dei servizi applicativi.

La specifica WS-SecurityPolicy non propone particolari costrutti per descrivere in una policy gli aspetti di tracciatura (audit). Per questo si propone di introdurre una policy assertion ***<spcoopp:Audit>*** che può avere la seguente sintassi:

```
    <spcoopp:Audit xmlns:spcoopp="..." ( wsp:Ignorable="true" )? ... >
      <wsp:Policy xmlns:wsp="..." >
        <spcoopp:Retention Duration="xs:duration" ... /> ?
        <spcoopp:ContactPoint Name="xs:string" ... /> ?
        <spcoopp:AuditedInfo Name="xs:string" ( Namespace="xs:anyURI" ) ? ...
/> *
        ...
      </wsp:Policy>
      ...
    </spcoopp:Audit>
```

/spcoopp:Audit

Identifica la policy assertion relativa all'effettuazione della tracciatura di informazioni. Se non si specificano nested policy, indica solo che il servizio effettua tracciatura.

/spcoopp:Audit/wsp:Policy

Indica la presenza di requisiti ulteriori.

/spcoopp:Audit/wsp:Policy/spcoopp:Retention

Elemento opzionale che specifica la politica di retention (cioè il periodo di tempo per cui le informazioni tracciate vengono conservate).

/spcoopp:Audit/wsp:Policy/spcoopp:Retention@spcoopp:Duration

Attributo che specifica il periodo di tempo per cui le informazioni tracciate vengono conservate.

/spcoopp:Audit/wsp:Policy/spcoopp:ContactPoint

Elemento opzionale con cui si segnala un recapito (per esempio e-mail) a cui rivolgersi per avere maggiori informazioni circa la politica di tracciatura.

/spcoopp:Audit/wsp:Policy/spcoopp:Retention@spcoopp:Name

Attributo che specifica il recapito.

/spcoopp:Audit/wsp:Policy/spcoopp:AuditedInfo

Elemento opzionale multiplo che contiene il dettaglio delle informazioni che vengono tracciate. Vi può essere più di un elemento di questo tipo.

/spcoopp:Audit/wsp:Policy/spcoopp:AuditedInfo/@Name

Attributo che caratterizza l'informazione tracciata mediante un nome.

/spcoopp:Audit/wsp:Policy/spcoopp:AuditedInfo/@Namespace

Attributo opzionale che caratterizza l'informazione tracciata mediante un namespace.

Di seguito un esempio di policy di tracciatura (frammento di policy).

```
[...]
<spcoopp:Audit xmlns:spcoopp="..." wsp:Ignorable="true">
  <wsp:Policy xmlns:wsp="...">
    <spcoopp:ContactPoint>logadmin@cnipa.gov.it</spcoopp:ContactPoint>
    <spcoopp:Retention>P30D</spcoopp:Retention>
    <spcoopp:AuditedInfo Name="RequestorIPAddress" />
    <spcoopp:AuditedInfo Name="Timestamp" />
    [...]
  </wsp:Policy>
</spcoopp:Audit>
[...]
```

E' importante notare che l'uso della policy assertion `<spcoopp:Audit>` rappresenta un tipico caso di policy "informativa", dove cioè l'informazione veicolata non costituisce un reale vincolo o requisito circa l'accesso al servizio bensì un insieme di dati che può essere valutato da chi intende accedere a tale servizio. Per questo motivo, per la policy `spcoopp:Audit` è naturale prevedere sempre l'uso dell'attributo `wsp:Ignorable` con valore "true".

Lo schema XML che formalizza la sintassi della policy assertion appena descritta è riportato in appendice.

5. BIBLIOGRAFIA

[1] OASIS Security Services (SAML) TC, Security Assertion Markup Language (SAML) V2.0 Technical Overview, Draft 15, 5 marzo 2008.

[2] OASIS Security Services (SAML) TC, Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, 15 marzo 2005.

[3] OASIS Security Services (SAML) TC, Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, 15 marzo 2005.

[4] OASIS Security Services (SAML) TC, Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, 15 marzo 2005.

[5] OASIS Security Services (SAML) TC, Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, 15 marzo 2005.

[6] OASIS Security Services (SAML) TC, Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, 15 marzo 2005.

[7] OASIS Security Services (SAML) TC, Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0, OASIS Standard, 15 marzo 2005.

[8] OASIS Security Services (SAML) TC, Errata Working Document for SAML V2.0, Working Draft, 24 marzo 2008.

[9] A Universally Unique IDentifier (UUID) URN Namespace (IETF RFC 4122), luglio 2005. http://www.ietf.org/rfc/rfc4122.txt

[10] IETF/W3C XML Signature WG, XML-Signature Syntax and Processing, W3C Recommendation, 12 febbraio 2002. http://www.w3.org/TR/xmldsig-core/

[11] W3C, XML Schema Part 2: Datatypes, W3C Recommendation, 2 maggio 2001.

[12] CNIPA, Sistema pubblico di cooperazione: Quadro Tecnico d'Insieme, versione 1.0, 14 ottobre 2005.

[13] CNIPA, Sistema pubblico di cooperazione: Servizi di Sicurezza, versione 1.0, 14 ottobre 2005.

[14] CNIPA, Sistema pubblico di cooperazione: Porta di Dominio, versione 1.0, 14 ottobre 2005.

[15] CNIPA, Sistema pubblico di cooperazione: Accordo di Servizio, versione 1.0, 14 ottobre 2005.

[16] CNIPA, Sistema pubblico di cooperazione: Busta di e-Gov, versione 1.1, 14 ottobre 2005.

[17] OASIS Web Services Security TC, Web Services Security: SAML Token Profile 1.1, OASIS Standard Incorporating Approved Errata, 1 novembre 2006.

[18] OASIS Web Services Security TC, WS-SecurityPolicy 1.2, OASIS Standard, 1 luglio 2007.

[19] W3C, Web Services Policy 1.5 – Primer, Working Group Note, 12 novembre 2007.

[20] W3C, Web Services Policy 1.5 – Framework, W3C Recommendation, 4 settembre 2007.

[21] W3C, Web Services Policy 1.5 – Attachment, W3C Recommendation, 4 settembre 2007.

[22] W3C, Web Services Policy 1.5 – Guidelines for Policy Assertion Authors, W3C Working Group Note, 12 novembre 2007.

[23]WS-I, Basic Profile 1.1 Second Edition, Final Version, 10 ottobre 2006.

WS-I, Basic Security Profile 1.0, Final, 30 marzo 2007

APPENDICE A: SCHEMI XML

Si riporta lo schema XML in cui sono definite le policy assertion e i costrutti ad hoc utilizzati nelle sezioni precedenti del documento.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:spcoopp="http://www.cnipa.gov.it/schema"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:wsp="http://docs.oasis-
open.org/ws-sx/ws-securitypolicy/200702"
targetNamespace="http://www.cnipa.gov.it/schema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <element name="Attributes" type="spcoopp:AttributesType"/>
  <complexType name="AttributesType">
    <sequence>
      <element name="Attribute" type="spcoopp:AttributeType"
minOccurs="0" maxOccurs="unbounded"/>
      <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      <element name="AttributeMapping"
type="spcoopp:AttributeMappingType" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <anyAttribute namespace="##any" processContents="lax"/>
  </complexType>
  <complexType name="AttributeType">
    <sequence>
      <element name="AttributeValue" type="xs:anyType"
nillable="true" minOccurs="0"/>
      <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </sequence>
    <attribute name="NameFormat" type="xs:anyURI" use="optional"/>
    <attribute name="Name" type="xs:string" use="required"/>
    <anyAttribute namespace="##any" processContents="lax"/>
```

```
      </complexType>
      <complexType name="AttributeMappingType">
            <sequence>
                  <element name="SenderAttribute" type="spcoopp:AttributeType"/>
                  <element name="ReceiverAttribute"
type="spcoopp:AttributeType"/>
                  <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <anyAttribute namespace="##any" processContents="lax"/>
      </complexType>
      <element name="Audit" type="spcoopp:AuditType"/>
      <complexType name="AuditType">
            <sequence>
                  <!-- <xs:element ref="wsp:Policy" minOccurs="0" /> -->
                  <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <anyAttribute namespace="##any" processContents="lax"/>
      </complexType>
      <element name="Retention" type="spcoopp:RetentionType"/>
      <complexType name="RetentionType">
            <sequence>
                  <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <attribute name="Duration" type="xs:duration" use="required"/>
            <anyAttribute namespace="##any" processContents="lax"/>
      </complexType>
      <element name="ContactPoint" type="spcoopp:ContactPointType"/>
      <complexType name="ContactPointType">
            <sequence>
                  <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <attribute name="Name" type="xs:string" use="required"/>
            <anyAttribute namespace="##any" processContents="lax"/>
```

```
      </complexType>
      <element name="AuditedInfo" type="spcoopp:AuditedInfoType"/>
      <complexType name="AuditedInfoType">
            <sequence>
                  <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <attribute name="Name" type="xs:string" use="required"/>
            <attribute name="Namespace" type="xs:anyURI" use="optional"/>
            <anyAttribute namespace="##any" processContents="lax"/>
      </complexType>
      <element name="Conditions" type="spcoopp:ConditionsType"/>
      <complexType name="ConditionsType">
            <sequence>
                  <element name="NotBefore" type="spcoopp:NotBeforeType"
minOccurs="0"/>
                  <element name="NotOnOrAfter" type="spcoopp:NotBeforeType"
minOccurs="0"/>
                  <element name="RequireAudienceRestriction"
type="spcoopp:RequireAudienceRestrictionType" minOccurs="0"/>
                  <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <anyAttribute namespace="##any" processContents="lax"/>
      </complexType>
      <complexType name="NotBeforeType">
            <sequence>
                  <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <attribute name="DateTime" type="xs:dateTime" use="required"/>
            <anyAttribute namespace="##any" processContents="lax"/>
      </complexType>
      <complexType name="NotOnOrAfterType">
            <sequence>
                  <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
```

```
            <attribute name="DateTime" type="xs:dateTime" use="required"/>
            <anyAttribute namespace="##any" processContents="lax"/>
      </complexType>
      <complexType name="RequireAudienceRestrictionType">
            <sequence>
                  <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <anyAttribute namespace="##any" processContents="lax"/>
      </complexType>
      <element name="Subject" type="spcoopp:SubjectType"/>
      <complexType name="SubjectType">
            <sequence>
                  <choice minOccurs="0">
                        <element name="NameIDFormatUnspecified"
type="xs:anyType"/>
                        <element name="NameIDFormatEmailAddress"
type="xs:anyType"/>
                        <element name="NameIDFormatX509SubjectName"
type="xs:anyType"/>
                        <element name="NameIDFormatWindowsDomainQualifiedName"
type="xs:anyType"/>
                        <element name="NameIDFormatKerberosPrincipalName"
type="xs:anyType"/>
                        <element name="NameIDFormatEntityIdentifier"
type="xs:anyType"/>
                        <element name="NameIDFormatPersistentIdentifier"
type="xs:anyType"/>
                        <element name="NameIDFormatTransientIdentifier"
type="xs:anyType"/>
                  </choice>
                  <element name="SubjectNameQualifier"
type="spcoopp:SubjectNameQualifierType" minOccurs="0"/>
                  <choice minOccurs="0" maxOccurs="unbounded">
                        <element name="SubjectConfirmationMethodSaml20Bearer"
type="xs:anyType"/>
                        <element
name="SubjectConfirmationMethodSaml20HolderOfKey" type="xs:anyType"/>
                        <element
name="SubjectConfirmationMethodSaml20SenderVouches" type="xs:anyType"/>
```

```
                </choice>
                <element name="RequireSubjectConfirmationData"
type="spcoopp:RequireSubjectConfirmationDataType" minOccurs="0"/>
                <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <anyAttribute namespace="##any" processContents="lax"/>
      </complexType>
      <complexType name="SubjectNameQualifierType">
            <sequence>
                <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <attribute name="NameQualifier" type="xs:string" use="required"/>
            <anyAttribute namespace="##any" processContents="lax"/>
      </complexType>
      <complexType name="RequireSubjectConfirmationDataType">
            <sequence>
                <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <anyAttribute namespace="##any" processContents="lax"/>
      </complexType>
      <element name="AuthnContext" type="spcoopp:AuthnContextType"/>
      <complexType name="AuthnContextType">
            <sequence>
                <element name="RequiredAuthnClass"
type="spcoopp:RequiredAuthnClassType" maxOccurs="unbounded"/>
                <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
            <anyAttribute namespace="##any" processContents="lax"/>
      </complexType>
      <complexType name="RequiredAuthnClassType">
            <sequence>
                <any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
            </sequence>
```

```
            <attribute name="AuthnContextClassRef" type="xs:anyURI"
use="required"/>
            <anyAttribute namespace="##any" processContents="lax"/>
      </complexType>
     </schema>
```