


X-RBAC

Specifica XML-based per il controllo dell'accesso di documenti XML nei Web-Services

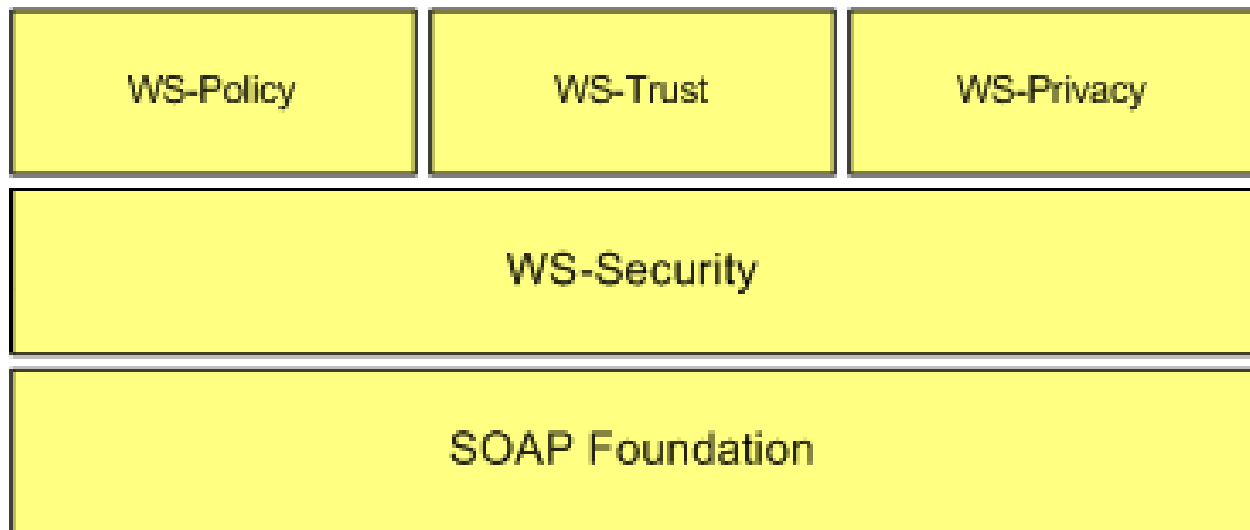


WS-Security Stack

WS-Security è un componente che può essere utilizzato con altre Web-Service extensions e con altri protocolli di alto livello al fine di supportare un ampio insieme di modelli di sicurezza.

WS-Security Stack

Il modello offerto da WS-Security fornisce le basi per altre specifiche di sicurezza:



WS-Security Stack

Posizionati sopra WS-Security, abbiamo un livello di specifiche che comprende:

WS-Policy: framework che fornisce un modello e la corrispondente sintassi per descrivere le policy di un Web-Services

WS-Trust: definisce come si possa istituire un rapporto di fiducia tra service requester e service provider

WS-Privacy: specifica come definire le politiche sulla privacy

WS-Policy

- WS-Policy fornisce un modello generico, e la corrispondente sintassi, utile a definire un insieme di base di costrutti che possono essere utilizzati da altre specifiche per descrivere un ampio insieme di requisiti, preferenze e concessioni che uno specifico Web-Service può avere la necessità di gestire.

X-RBAC

X-RBAC è un framework basato sulla specifica di politiche RBAC (Role-Based Access Control), espresse attraverso XML, che consente di implementare un modello per il controllo degli accessi nei Web-Services e che si posiziona al livello di WS-Policy.

Web-Services e XML

La crescente richiesta di condivisione di contenuti e di servizi on-line ha portato al massiccio sviluppo di Web-Services per lo scambio di documenti tra data repositories. XML, grazie alle proprie caratteristiche, stà giocando un ruolo sempre più importante nella rappresentazione delle informazioni scambiate.

Problematiche di sicurezza

Il servizio di condivisione di contenuti on-line, operato da questi Web-Services, introduce però un insieme di nuove problematiche, legate alla disseminazione di dati sensibili, che non vengono considerate dai modelli di sicurezza tradizionali.

Problematiche di sicurezza

Il nuovo modello di sicurezza deve:

- basarsi sui contenuti
- essere dipendente dal contesto
- gestire l'eterogeneità degli oggetti
- gestire l'eterogeneità dei soggetti
- permettere la specifica di politiche con una granularità superiore a quella dell'intero documento

Basato sui contenuti

Il modello deve garantire la possibilità di richiedere che l'accesso alle informazioni sia ristretto sulla base dei contenuti.

Esempio: in un'azienda ospedaliera deve essere possibile garantire accesso selettivo alle informazioni sui pazienti in base al ruolo svolto dagli operatori.

Dipendente dal contesto

Il modello deve catturare tutte le informazioni, rilevanti per la sicurezza, relative al contesto e utilizzarle nelle proprie decisioni per il controllo degli accessi.

Esempio: un Web-Service che distribuisce librerie potrebbe rendere disponibili tali risorse solo agli utenti appartenenti ad un determinato dominio.

Eterogeneità degli oggetti

Il modello deve garantire la possibilità di specificare policy su oggetti di varia natura: definizioni di documenti (XML schema), istanze di documenti e loro componenti, oggetti concettuali (come i cluster).

Eterogeneità dei soggetti

Il modello deve permettere di gestire in modo ottimale soggetti eterogenei. Non conosciamo a priori le caratteristiche, le necessità o il numero degli utenti.

Inoltre il loro profilo, caratteristiche e qualifiche, può essere dinamico e può cambiare nel tempo richiedendo una modifica delle autorizzazioni.

Esempio: un utente può spostarsi da un dominio ad un altro.

Granularità

Il modello deve dare la possibilità di esprimere policy che garantiscano un livello di granularità più fine dell'intero documento, e che permettano di far riferimento anche alle specifiche di documenti e a raggruppamenti logici (cluster) di schema XML, istanze XML e loro elementi e attributi.

Il modello RBAC (richiami)

Il modello scelto come base per lo sviluppo del framework è RBAC (Role-Based Access Control) al quale sono state aggiunte delle estensioni.

Il modello RBAC si basa su: utenti, ruoli, permessi.

Il modello RBAC (richiami)

Gli utenti sono i soggetti che accedono al sistema.

I ruoli rappresentano le funzioni che un utente può svolgere nel sistema.

I permessi sono le autorizzazioni che i diversi ruoli hanno sugli oggetti del sistema.

Quando un utente ricopre un ruolo eredita tutte le autorizzazioni assegnate al ruolo stesso.

Estensioni al modello RBAC

Il modello adottato include anche:

- gerarchie dei ruoli (il senior eredita da junior)
- separazione delle competenze (separation of duties, evita conflitti tra ruoli)
- sensibilità al contesto (locazione)
- sensibilità ai dati della sessione (durata, login_time, login_date, ...)

Estensioni al modello RBAC

- specifica in base ha i contenuti su 4 livelli: concettuale, schema, istanza e elemento

Raggruppare le informazioni in cluster concettuali riduce la complessità del processo di specifica e di amministrazione della sicurezza.

Specifica degli elementi X-RBAC

X-RBAC si basa sull'utilizzo di 5 elementi:

- user credentials (XUS Xml User Sheet)
- roles (XRS Xml Role Sheet)
- permissions (XPS Xml Permission Sheet)
- user-to-role mapping (XURM)
- permission-to-role mapping (XPRM)

User credentials

Per valutare un particolare utente si utilizza il concetto di credenziale. Una credenziale è un insieme di attributi legati ad un soggetto che sono ritenuti rilevanti ai fini della sicurezza. Credenziali con struttura simile sono raggruppate in un unico tipo. XCredTypeDef è un XML schema che permette di definire nuovi tipi di credenziali.

Istanza di XCredTypeDef

```
<XCredTypeDef>
  <credential_type cred_type_id="C100">
    <type_name>Nurse</type_name>
    <attribute_list>
      <attribute_name type="integer">age</attribute_name>
      <attribute_name type="string">field</attribute_name>
      <attribute_name type="integer">level</attribute_name>
      <attribute_name type="string">status</attribute_name>
    </attribute_list>
  </credential_type>
</XCredTypeDef>
```

Istanza di XCredTypeDef per definire la struttura della credenziale di tipo Nurse

XUS

In un XML User Sheet vengono inserite tutte le credenziali di un utente.

Ogni credenziale contiene un identificatore, che indica il tipo della credenziale, e un insieme di coppie attributo-valore, che verranno utilizzate per valutare se tale credenziale permette l'acquisizione di un ruolo.

XUS

```
<XUS>
  <user user_id="john">
    <user_name>John</user_name>
    <cred_type cred_type_id="C100">
      <type_name>Nurse</type_name>
      <cred_expr>
        <age>30</age>
        <field>ophthalmology</field>
        <level>6</level>
        <status>single</status>
      </cred_expr>
    </cred_type>
    <max_roles>2</max_roles>
  </user>
</XUS>
```

Credenziali dell'utente John

Tipo di credenziale

Coppie attributo-valore

Cardinalità: max numero di ruoli che John può ricoprire

Roles

I ruoli vengono creati dagli amministratori. Ogni ruolo ha associate un insieme di credenziali che gli utenti assegnati ad esso devono possedere.

XRS

```
<XRS>
<roles>
<role role_id="R100">
  <role_name>Nurse</role_name>
  <senior>Eye_Doctor</senior>
  <cardinality>8</cardinality>
</role>
<role role_id="R200">
  <role_name>Eye_Doctor</role_name>
  <DSD_Role_Set_id>DSD1</DSD_Role_Set_id>
  <junior>Nurse</junior>
  <senior>Eye_Surgeon</senior>
  <cardinality>6</cardinality>
</role>
</roles>
<DSD_Role_Sets>
<DSD_Role_Set DSD_Role_Set_id="DSD1"
  DSD_cardinality="1">
  <DSD_Role>Eye_Doctor</DSD_Role>
  <DSD_Role>Eye_Surgeon</DSD_Role>
</DSD_Role_Set>
</DSD_Role_Sets>
</XRS>
```

Nome del ruolo

Cardinalità: max numero di utenti che possono ricoprire questo ruolo

Padre e figlio nella gerarchia dei ruoli

Dinamic (Static) Separation of Duty

Ogni ruolo può avere due parti opzionali:

- **SSD_Role_Set (Static Separation of Duty)**

Indica che non possono essere assegnati più di n dei ruoli indicati ad uno stesso utente

- **DSD_Role_Set (Dinamic Separation of Duty)**

Indica che non possono essere attivati simultaneamente da uno stesso utente più di m dei ruoli indicati

Permissions

I permessi definiscono quali operazioni possono essere effettuate sugli oggetti del sistema. Gli oggetti possono essere:

- **cluster** (identificati da id)
- **schema** (identificati da id)
- **instance document** (identificati da id)
- **document element** (identificati con XPath)

XPS

```
<XPS>
  <permission perm_id="P1">
    <object_type>Schema</object_type>
    <object_id>XS101</object_id>
    <operation>all</operation>
  </permission>
  <permission perm_id="P2">
    <object_type>Instance</object_type>
    <object_id>XI100</object_id>
    <operation>all</operation>
  </permission>
  <permission perm_id="P3">
    <object_type>Element</object_type>
    <object_id>/EyeCareMedicalHistory/Patient/Name</object_id>
    <operation>read</operation>
  </permission>
</XPS>
```

Definizione dei permessi P1

Tipo di oggetto a cui si riferisce

ID dell'oggetto

Permessi

Elemento identificato con XPath

XPS

Le operazioni che possono essere eseguite su un oggetto sono:

- **read** - lettura delle informazioni
- **write** - modifica delle informazioni
- **navigate** - possibilità di navigarvi attraverso
- **all** - tutte le precedenti

XURM

```
<XURM>
<urm urm_id='URM1' />
  <role_name>Eye_Doctor</role_name>
  <cred_type>Nurse</cred_type>
  <conditions><condition>
    <mode value='AND'>
      <predicate>
        <operation>gt</operation>
        <parameter1>level</parameter1>
        <parameter2>5</parameter2>
      </predicate>
      <predicate>
        <operation>lt</operation>
        <parameter1>age</parameter1>
        <parameter2>80</parameter2>
      </predicate>
    </mode>
  </condition></conditions>
</urm>
</XURM>
```

Definizione delle credenziali necessarie per appartenere al ruolo Eye_Doctor

Una possibilità (l'unica in questo caso) è quella di avere una credenziale di tipo Nurse

In cui level sia maggiore di 5 AND age minore di 80

XPRM

```
<XPRM>
  <prm prm_id='PRM1'>
    <role_name>Eye_Doctor</role_name>
    <permissions>
      <perm_id>P3</perm_id>
    </permissions>
  </prm>
  <prm prm_id='PRM2'>
    <role_name>Dispenser</role_name>
    <permissions>
      <perm_id>P4</perm_id>
    </permissions>
  </prm>
</XPRM>
```

Assegnazione dei permessi al ruolo Eye_Doctor

Al ruolo vengono assegnati i permessi specificati nel XPS con id P3

XUS, XPS, XRS, XURM, XPRM

Mantenere le specifiche degli utenti, dei ruoli e dei permessi indipendenti dalla loro associazione permette una progettazione e una amministrazione indipendente delle politiche di controllo.

Un utente ha accesso alle risorse in base al ruolo che gli viene assegnato dallo XURM e ai permessi che l'XPRM assegna a tale ruolo.

Esempio

```
<EyeCareMedicalHistory>
  <Patient id='1'>
    <Name>Jason</Name>
    <Age>64</Age>
    <History>
      <Disease>Glaucoma</Disease>
      <Date_Operated>12/09/78</Date_Operated>
      <Dues>15000</Dues>
    </History>
  </Patient>
  <Patient id='2'>
    <Name>Mary</Name>
    <Age>29</Age>
    <History>
      <Disease>Cataract</Disease>
      <Date_Operated>12/09/78</Date_Operated>
      <Dues>15000</Dues>
    </History>
  </Patient>
</EyeCareMedicalHistory>
```

Esempio

A John verrebbe assegnato il ruolo di Eye_Doctor in quanto egli ha credenziali di tipo Nurse, ha meno di 80 anni e un livello superiore a 5.

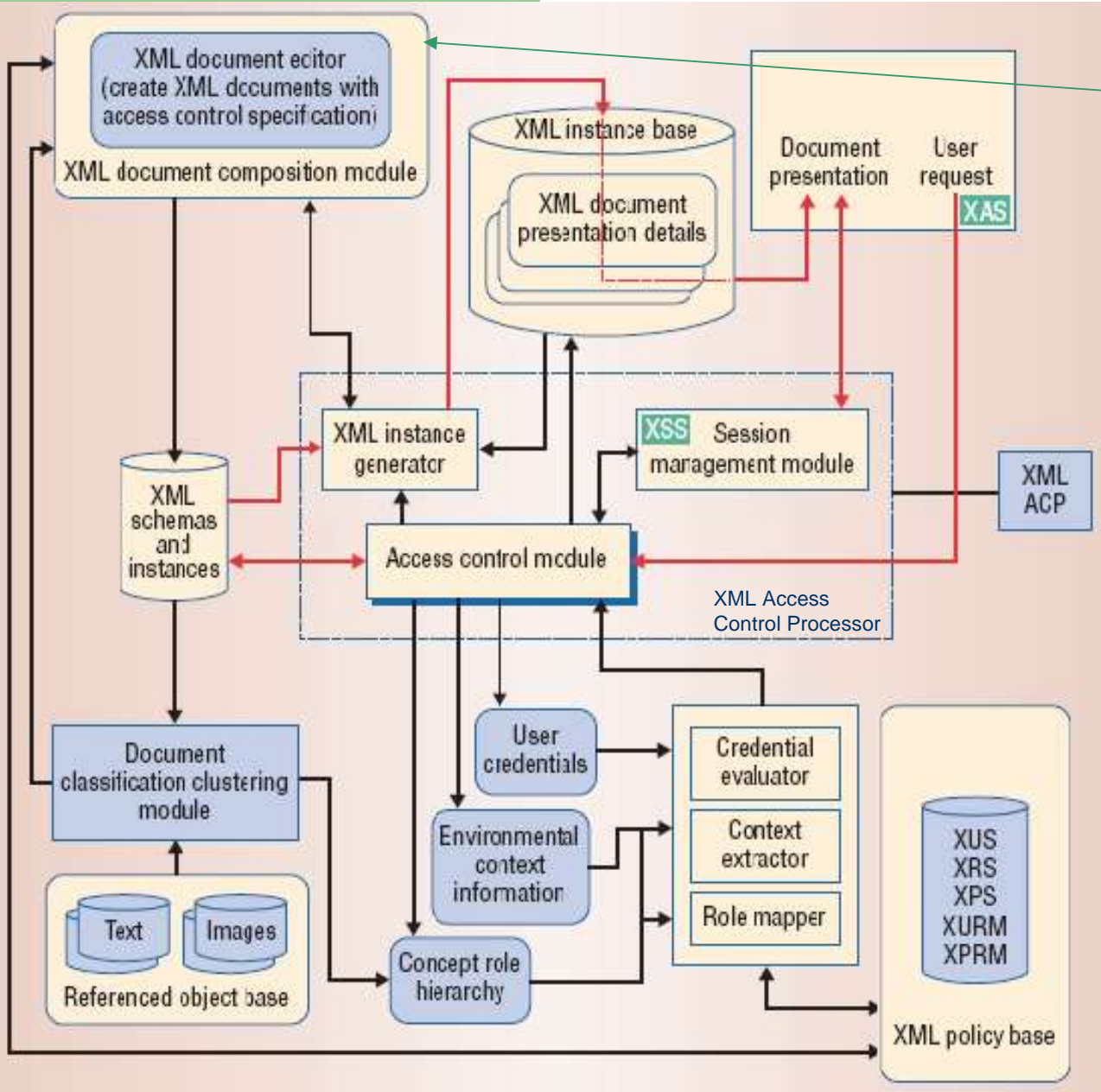
Al ruolo Eye_Doctor verrebbero assegnati i permessi espressi in P3.

John avrebbe quindi il diritto di leggere il contenuto dell'elemento identificato dal percorso XPath:

“/EyeCareMedicalHistory/Patient/Name”.

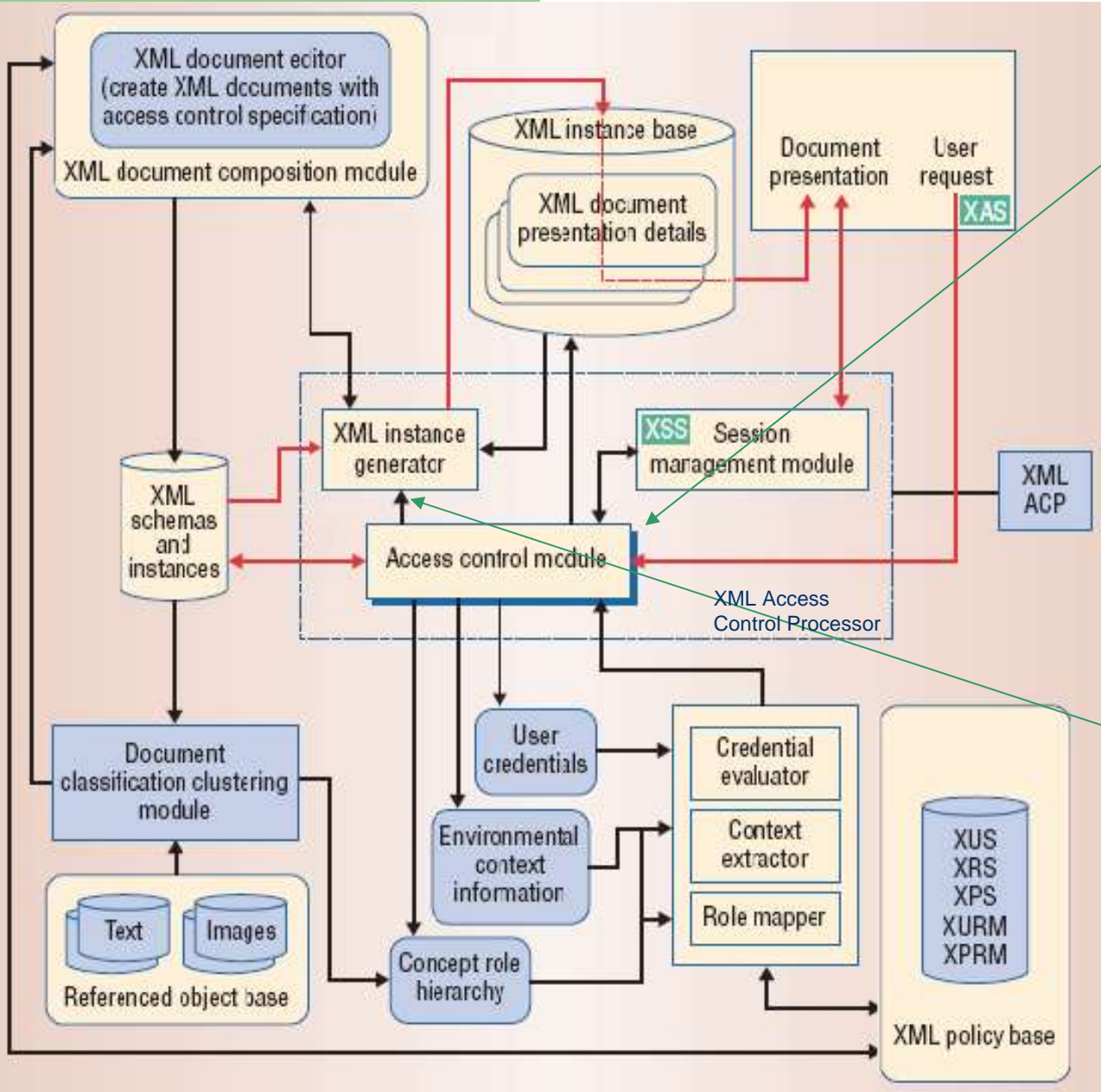
Architettura Software

L'architettura software proposta per una applicazione Web-Service-enabled, che si occupi della disseminazione sicura di documenti XML, basato sul framework X-RBAC è la seguente:



XML Document Composition Module (XDCM)

Fornisce la principale interfaccia per comporre gli schema degli elementi RBAC e per amministrare le policy.



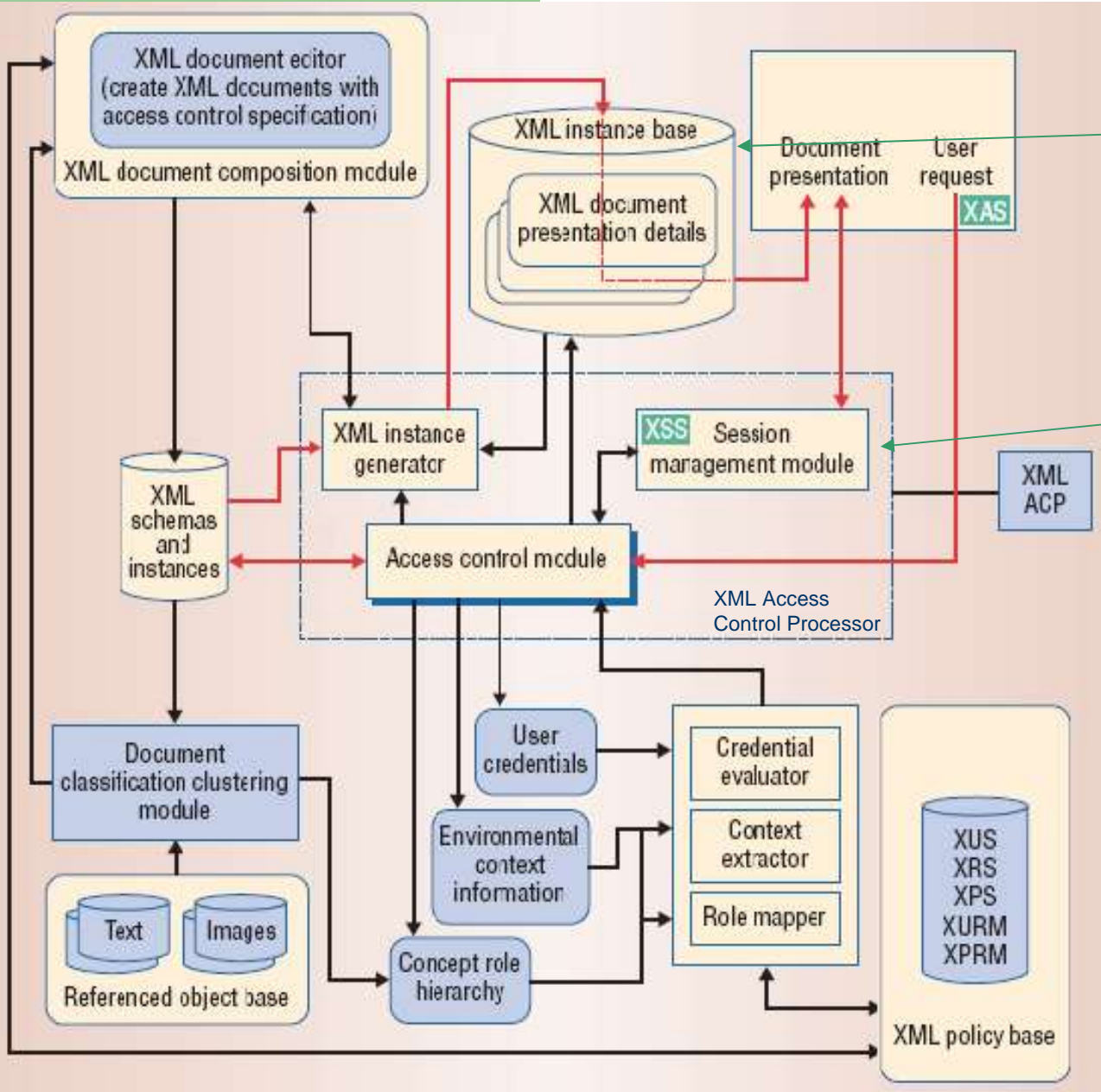
Access Control Module (ACM)

Componente chiave dell'architettura.

Si interfaccia con gli altri moduli e con i repository per prendere le decisioni necessarie sulle autorizzazioni.

XML Instance Generator (XIG)

Prende le informazioni su sui diritti di accesso dall'ACM e crea la vista XML di un oggetto in base ad essi

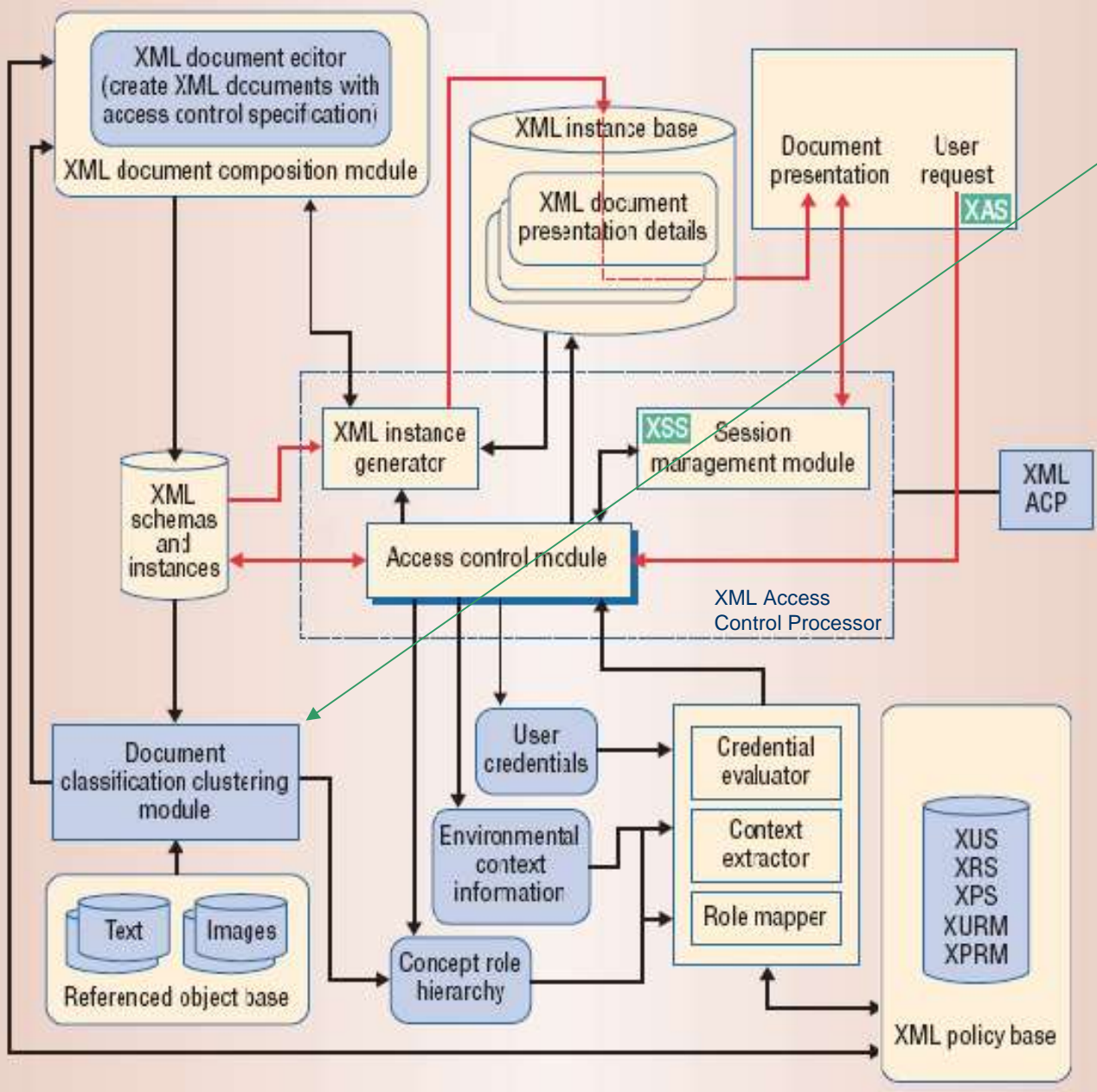


XML Instance Base (XIB)

Memorizza le viste prodotte dallo XIG

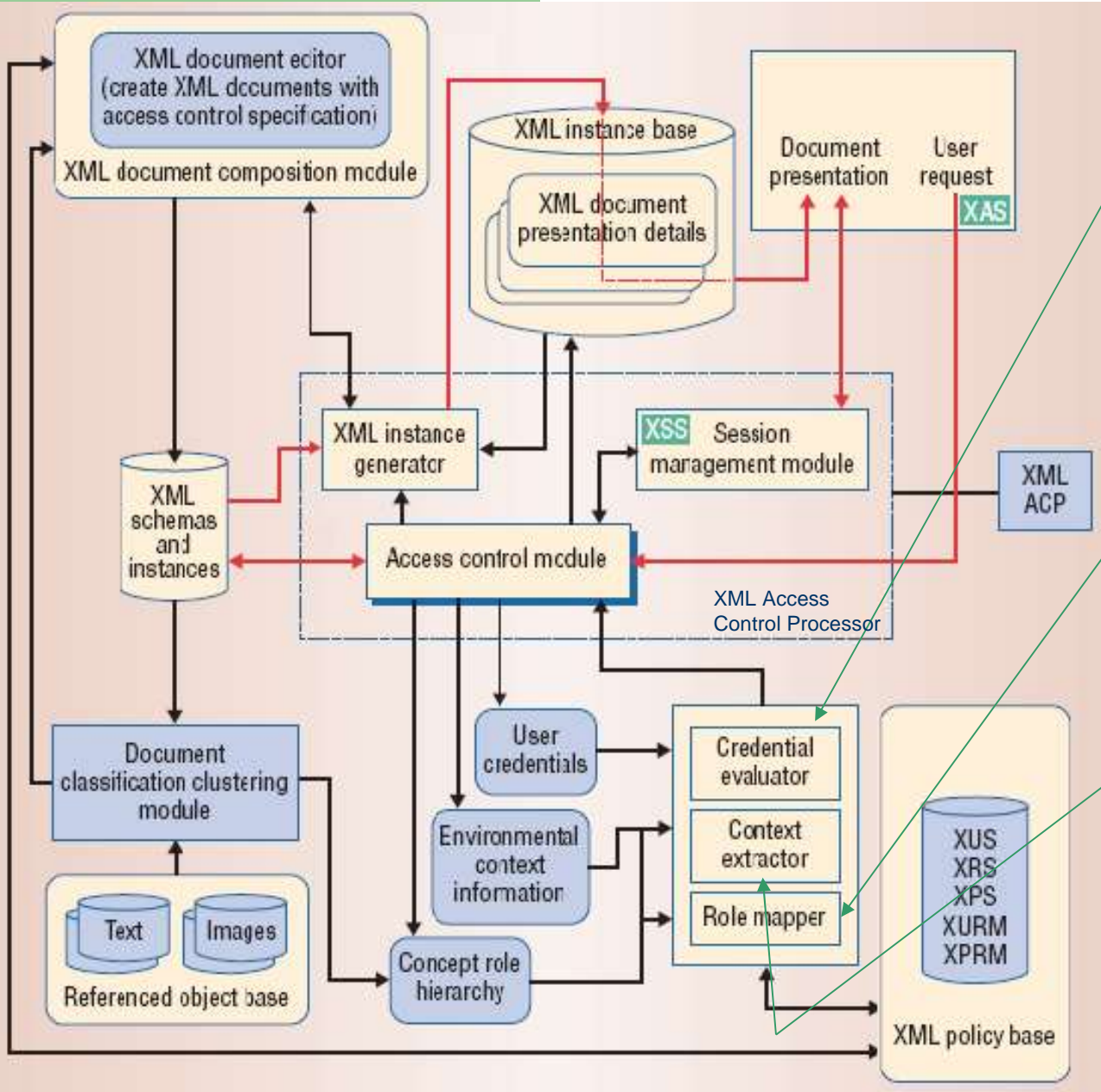
Session Management Module (SMM)

Monitora le attività svolte in una sessione e cattura quelle rilevanti, utili per l'aggiornamento delle credenziali e la modifica dei diritti di accesso futuri. Il sistema memorizza tali informazioni nell'XSS (XML Session Sheet)



Document Classification Clustering Module (DCM)

Gestisce la classificazione di nuovi documenti nei cluster esistenti e permette di creare o eliminare clustering.



Credential Evaluator Module (CEM)

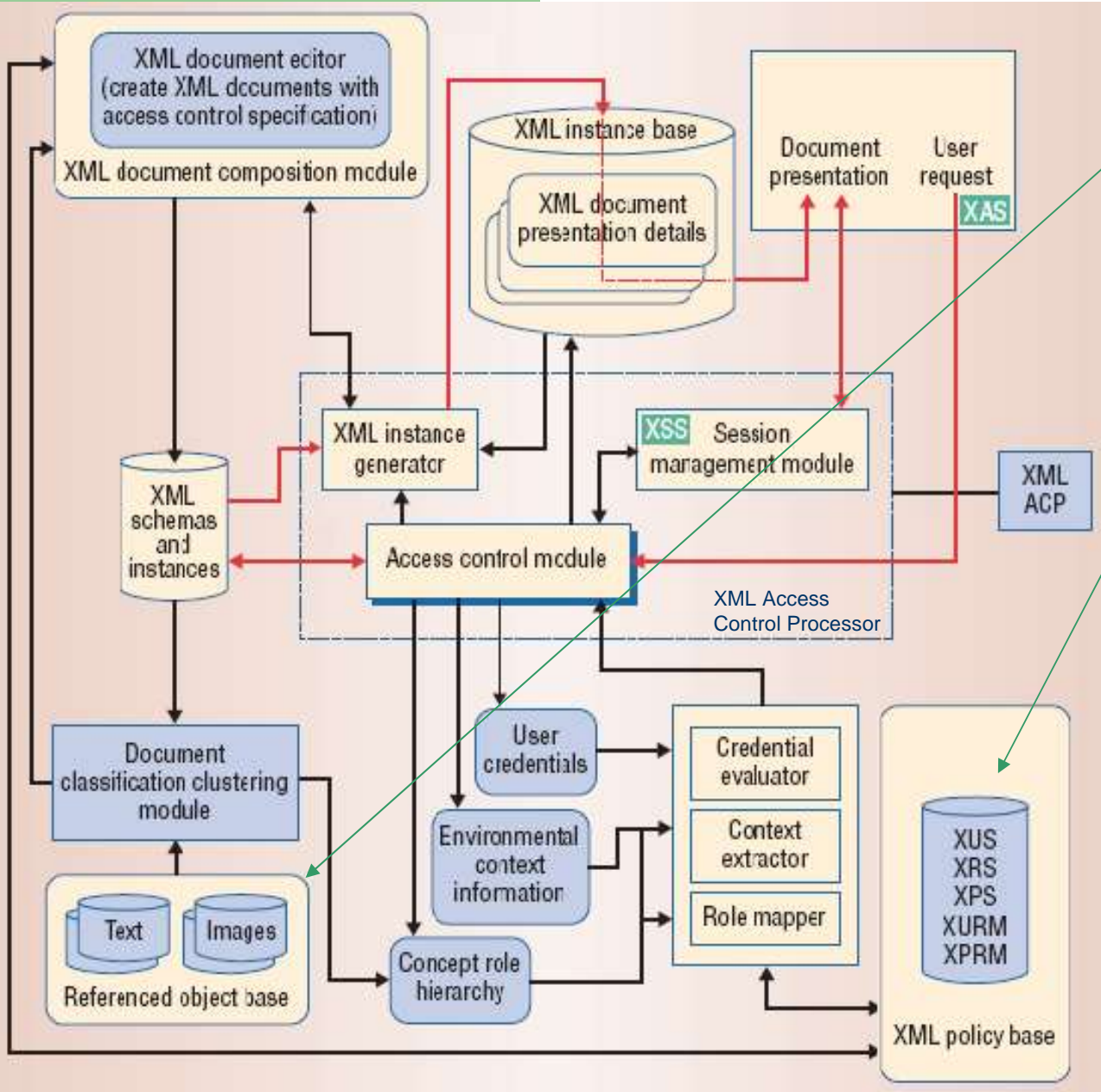
Valuta le credenziali che gli vengono inviate da ACM e assegna l'utente ad un tipo di credenziale.

Role Mapper

Collabora con il CEM per assegnare un ruolo ad un utente.

Context Extractor

Valuta le informazioni relative al contesto e comunica all'ACM eventuali decisioni, da esse derivanti, sugli accessi

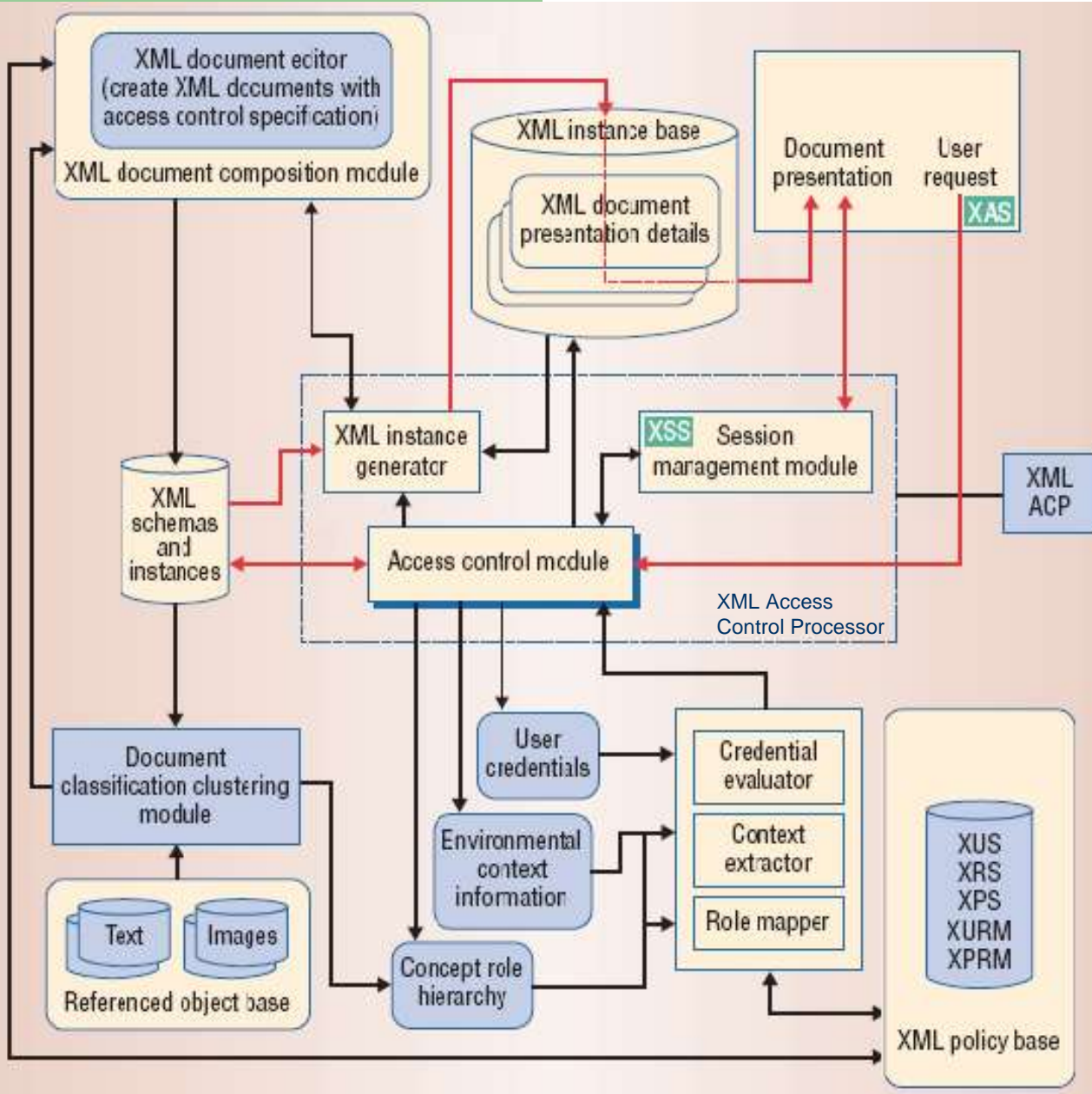


Referenced Object Base

Oggetti presenti nel sistema.

XML Policy Base

Contiene tutti i documenti XML relativi alle policy composti dall'XDCM.



L'utente invia una richiesta (XAS – Xml Access Sheet) all'ACM contenente le credenziali e le richieste di accesso.

ACM genera, collaborando con gli altri moduli, un insieme di autorizzazioni, e identifica le sorgenti XML.

XIG genera la vista a partire dalle sorgenti e dalle autorizzazioni.

Il risultato viene fornito all'utente.