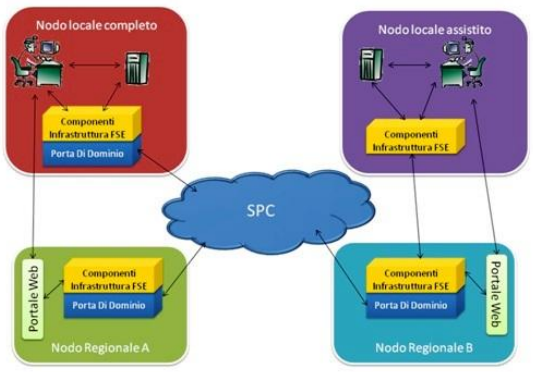
Presidenza del Consiglio dei Ministri
Dipartimento per la digitalizzazione della Pubblica Amministrazione e l'innovazione tecnologica

Consiglio Nazionale delle Ricerche
Dipartimento delle Tecnologie dell'Informazione e delle Comunicazioni

Progetto
"Infrastruttura Tecnologica del Fascicolo Sanitario Elettronico"



InFSE:

Infrastruttura tecnologica del Fascicolo Sanitario Elettronico

Linee guida

eGov 2012 – Obiettivo Salute

Luglio 2012

Indice

Presentazione del documento 9

1 Prefazione 10

2 Sintesi 11

3 Obiettivi del documento 12

3.1 Versioning 13

4 Termini e acronimi 14

5 Contesto internazionale e nazionale 17

5.1 Healthcare Services Specification Project (HSSP) 17

5.1.1 Retrieve, Locate and Update Service (RLUS) 17

5.1.2 Entity Identification Service (EIS) 18

5.2 Integrating the Healthcare Enterprise (IHE) 19

5.3 European Patients Smart Open Services (epSOS) 20

5.4 Sperimentazione di un sistema per l'Interoperabilità europea e nazionale delle soluzioni di fascicolo sanitario elettronico: componenti Patient Summary e ePrescription (IPSE) 20

6 Requisiti dell'Infrastruttura del FSE 22

7 Modello architetturale di InFSE 24

7.1 Descrizione del modello 24

7.2 Architettura multi-layers 26

8 Architettura delle componenti di InFSE 27

8.1 Interfaccia di Accesso 27

8.1.1 Attori e ruoli 28

8.1.2 Casi d'uso 30

8.1.2.1 Casi d'uso per la gestione dei documenti 30

8.1.2.2 Casi d'uso per la gestione degli eventi 30

8.1.2.3 Casi d'uso per la gestione degli indici ai documenti 33

8.1.2.4 Casi d’uso per la gestione delle federazioni 35
8.1.3 Architettura della componente 36
8.1.4 Integrazione con il Sistema Pubblico di Connettività 40
8.1.5 Scenari d’uso 40
8.1.5.1 Creazione documento 40
8.1.5.2 Aggiornamento documento 42
8.1.5.3 Recupero documento 44
8.1.5.3.1 Recupero documento in ambito regionale 44
8.1.5.3.2 Recupero documento in ambito extra-regionale 44
8.1.5.4 Registrazione topic 47
8.1.5.5 Sottoscrizione topic 49
8.1.5.6 Creazione topic 51
8.1.5.7 Creazione gerarchia 53
8.1.5.8 Archivazione gerarchia 56
8.1.5.9 Notifica di un evento 58
8.2 Registro Indice Federato 60
8.2.1 Modello concettuale 61
8.2.2 Attori e ruoli 62
8.2.3 Casi d’uso 62
8.2.4 Tecnologie di riferimento 65
8.2.5 Architettura della componente 65
8.2.6 Modello dati 69
8.2.7 Scenari d’uso 74
8.2.7.1 Ricerca di documenti in ambito regionale 74
8.2.7.2 Ricerca federata di documenti in ambito extra-regionale 74
8.2.7.3 Notifica di eventi tra registri 77
8.2.8 Integrazione con il Sistema Pubblico di Connettività 79

8.3 Gestore Gerarchico degli Eventi 80
8.3.1 Modello concettuale 80
8.3.2 Modello gerarchico degli eventi 81
8.3.3 Attori e ruoli 84
8.3.4 Casi d'uso 86
8.3.5 Architettura della componente 89
8.3.6 Integrazione con il Sistema Pubblico di Connettività 94
8.3.7 Scenari d'uso 94
8.3.7.1 Registrazione topic 94
8.3.7.2 Registrazione gerarchia 96
8.3.7.3 Sottoscrizione Topic 97
8.3.7.4 Sottoscrizione gerarchia 99
8.3.7.5 Creazione topic 101
8.3.7.6 Creazione gerarchia 102
8.3.7.7 Archiviazione gerarchia 104
8.3.7.8 Notifica di un evento 105
8.4 Gestore dei Documenti 107
8.4.1 Attori e ruoli 108
8.4.2 Casi d'uso 108
8.4.3 Architettura della componente 109
8.4.3.1 Integrazione con il Sistema Pubblico di Connettività 110
8.4.4 Scenari d'uso 110
8.4.4.1 Caricamento documento 110
8.4.4.2 Aggiornamento documento 111
8.4.4.3 Recupero documento 112
8.5 Gestore delle Politiche di Accesso 113
8.5.1 Fase di autenticazione 115

8.5.2 Fase di identificazione 116
8.5.3 Fase di autorizzazione 117
8.5.4 Componenti previste 118
8.5.5 Integrazione dei PEP con le componenti di InFSE 121
8.5.6 Politiche per il controllo degli accessi 122
8.5.7 Standard XACML 123
8.5.8 Adattamento al dominio InFSE 128
8.5.8.1 Context Handler 128
8.5.8.2 Attribute Finder 128
8.5.8.3 Policy Finder 128
8.5.9 Metodo di autorizzazione basato sul ruolo 129
8.5.9.1 Ruoli: prima parte delle credenziali di autorizzazione 129
8.5.9.2 Proprietà dei ruoli 132
8.5.9.3 Contesto: seconda parte delle credenziali di autorizzazione 133
8.5.9.4 Modalità di autorizzazione ed elementi delle regole 134
Bibliografia 135

Indice delle figure

Figura 1. Modello architetturale di InFSE 25
Figura 2. Architettura multi-layers di InFSE 26
Figura 3. Attori e ruoli 29
Figura 4. Casi d'uso per la gestione dei documenti 30
Figura 5. Casi d'uso per il produttore di eventi 32
Figura 6. Casi d'uso per il consumatore di eventi 32
Figura 7. Casi d'uso per il produttore di metadati 34
Figura 8. Casi d'uso per il consumatore di metadati 34
Figura 9. Casi d'uso per il manager della federazione 35
Figura 10. Interfacce IEvent e IBrokerFederation 37
Figura 11. Interfaccia IDocument 38
Figura 12. Interfacce IEntry e IRegistryFederation 39
Figura 13. Scenario Creazione documento 41
Figura 14. Scenario Aggiornamento documento 43
Figura 15. Scenario Recupero documento in ambito regionale 45
Figura 16. Scenario Recupero documento in ambito extra-regionale 46
Figura 17. Scenario Registrazione topic 48
Figura 18. Scenario Sottoscrizione topic 50
Figura 19. Scenario Creazione topic 52
Figura 20. Scenario Creazione gerarchia 55
Figura 21. Scenario Archivazione gerarchia 57
Figura 22. Scenario Notifica di un evento 59
Figura 23. Modello concettuale del Registro Indice Federato 61
Figura 24. Casi d'uso per il produttore di metadati 63
Figura 25. Casi d'uso per il consumatore di metadati 64
Figura 26. Casi d'uso per un nodo registro 64
Figura 27. Livelli di federazione del Registro Indice Federato 67
Figura 28. Interfacce IMetadataMgt, IQueryMgt, IEventMgt e IRegistryFederationMgt 68
Figura 29. Interfaccia ed architettura della componente MetadataLifecycleMgt 71
Figura 30. Interfaccia ed architettura della componente QueryMgt 72
Figura 31. Interfaccia ed architettura della componente FederationMgt 73
Figura 32. Scenario Ricerca di documenti in ambito regionale 75
Figura 33. Scenario Ricerca federata di documenti in ambito extra-regionale 76
Figura 34. Scenario Notifica di eventi tra registri 78

Figura 35. Modello concettuale del Gestore Gerarchico degli Eventi 81
Figura 36. Modello gerarchico degli eventi 83
Figura 37. Attori e ruoli 85
Figura 38. Casi d'uso per il produttore di eventi 88
Figura 39. Casi d'uso per il consumatore di eventi 88
Figura 40. Casi d'uso per il gestore della federazione 89
Figura 41. Interfacce supportate dai nodi broker 90
Figura 42. Interfaccia ed architettura della componente PublisherRegistrationMgt 91
Figura 43. Interfaccia ed architettura della componente SubscriptionMgt 92
Figura 44. Interfaccia ed architettura della componente NotificationBrokerMgt 93
Figura 45. Scenario Registrazione topic 95
Figura 46. Scenario Registrazione gerarchia 96
Figura 47. Scenario Sottoscrizione topic 98
Figura 48. Scenario Sottoscrizione gerarchia 100
Figura 49. Scenario Creazione topic 101
Figura 50. Scenario Creazione gerarchia 103
Figura 51. Scenario Archiviazione gerarachia 104
Figura 52. Scenario Notifica di un evento 106
Figura 53. Casi d'uso per il Gestore dei Documenti 108
Figura 54. Interfaccia IDocumentMgt 109
Figura 55. Interfaccia ed architettura della componente DocumentMgt 109
Figura 56. Scenario Caricamento documento 110
Figura 57. Scenario Aggiornamento documento 111
Figura 58. Scenario Recupero documento 112
Figura 59. Componenti del Gestore delle Politiche di Accesso 120
Figura 60. Architettura di riferimento XACML 125
Figura 61. Modello del linguaggio XACML 127
Figura 62. Ruoli previsti in InFSE 130

Indice delle tabelle

Tabella 1. Descrizione dei ruoli previsti in InFSE.....132

Presentazione del documento

Parte I – Contesto di riferimento

Nella prima parte del documento è definito il contesto di riferimento

Capitolo 1: Prefazione

Capitolo 2: Sintesi

Capitolo 3: Obiettivi del documento
L'obiettivo è di indicare le linee guida di riferimento per InFSE.

Capitolo 4: Termini e acronimi

Capitolo 5: Contesto internazionale e nazionale.
Illustrazione dei progetti nazionali e internazionali per la standardizzazione delle soluzioni tecnologiche.

Parte II – Infrastruttura tecnologica del Fascicolo Sanitario Elettronico

Nella seconda parte del documento sono definite le linee guida per l'Infrastruttura tecnologica del Fascicolo Sanitario Elettronico

Capitolo 6: Requisiti dell'Infrastruttura del FSE
Definizione dei requisiti funzionali dell'Infrastruttura tecnologica del Fascicolo Sanitario Elettronico.

Capitolo 7: Modello architetturale di InFSE
Descrizione del modello architetturale di InFSE orientato ai servizi (Service-Oriented Architecture, SOA).

Capitolo 8: Architettura delle componenti di InFSE
Descrizione delle componenti dell'Infrastruttura tecnologica del Fascicolo Sanitario Elettronico.

1 Prefazione

Il presente documento definisce le linee guida di riferimento per garantire l'interoperabilità tra le soluzioni di fascicolo sanitario elettronico del cittadino (FSE) in corso di realizzazione e/o di studio a livello territoriale.

Le predette linee guida non sono da intendersi prescrittive per le Regioni e le Province Autonome che hanno realizzato, stanno realizzando o intendono realizzare soluzioni di FSE per i loro cittadini, ma costituiscono un primo iniziale risultato del dibattito in atto nell'ambito del Tavolo permanente per la Sanità Elettronica delle Regioni e delle Province Autonome, un patrimonio informativo comune su cui costruire per favorire una prossima convergenza verso soluzioni condivise.

2 Sintesi

Lo sviluppo di una soluzione nazionale di Fascicolo Sanitario Elettronico (FSE) comporta la progettazione di un modello architetturale dell'infrastruttura tecnologica, nella quale sono definiti sia i meccanismi per la raccolta e disponibilità di documenti e dati sanitari in formato digitale, sia i servizi di supporto ai processi sanitari.

L'iniziativa per la realizzazione dell'Infrastruttura tecnologica del Fascicolo Sanitario Elettronico (Progetto InFSE), realizzata nell'ambito di una collaborazione tra il Dipartimento per la digitalizzazione della pubblica amministrazione e l'innovazione tecnologica e il Dipartimento Tecnologie dell'Informazione e delle Comunicazioni del Consiglio Nazionale delle Ricerche (CNR), è finalizzata a sostenere il processo di costruzione e diffusione di una infrastruttura federata di FSE del cittadino, condivisa a livello nazionale e allineata allo scenario internazionale, con particolare attenzione ai temi della sicurezza e della privacy.

Scopo di InFSE è quello di offrire un'infrastruttura tecnologica che consenta a tutti i cittadini e agli operatori sanitari autorizzati di accedere ai documenti sanitari di loro competenza, ovunque essi siano localizzati nel territorio nazionale e nel rispetto della tutela della privacy, e di facilitare la gestione dell'evoluzione dello stato, nel tempo, dei processi sanitari. Le informazioni sanitarie possono essere rese disponibili sia per gli usi primari (come l'assistenza e l'emergenza), che per gli usi secondari, cioè per finalità amministrative e di governo.

L'Infrastruttura tecnologica del FSE dovrà garantire la compatibilità con le soluzioni architetturali già sviluppate presso alcune Regioni e Province Autonome, in una visione generale orientata verso un unico modello di Infrastruttura federata, condivisa a livello nazionale e allineata allo scenario internazionale. Inoltre, il modello deve recepire i requisiti infrastrutturali necessari alla interoperabilità funzionale e semantica oggetto dei progetti nazionali ed europei in tema di FSE.

Per rispettare tutti questi requisiti, il modello architetturale di InFSE deve rispecchiare, nel suo complesso, un'architettura orientata ai servizi (Service-Oriented Architecture, SOA), la quale, a sua volta, deve essere posta al di sopra delle infrastrutture tecnologiche del Sistema Pubblico di Connettività (SPC) per la cooperazione applicativa tra Pubbliche Amministrazioni.

3 Obiettivi del documento

Il presente documento ha l'obiettivo di indicare le linee guida di riferimento per l'implementazione dell'Infrastruttura tecnologica del Fascicolo Sanitario Elettronico (InFSE).

Il Fascicolo Sanitario Elettronico (FSE) è qui inteso come l'insieme di documenti e dati digitali di tipo sanitario e socio-sanitario (prescrizioni, referti, patient summary, etc.) inerenti allo stato di salute dei cittadini prodotti all'atto dei loro rapporti con i diversi attori del Servizio Sanitario Nazionale (SSN).

Scopo di InFSE è quello di offrire un'infrastruttura tecnologica che consenta a tutti i cittadini e agli operatori sanitari autorizzati di accedere ai documenti sanitari di loro competenza, ovunque essi siano localizzati nel terriorio nazionale e nel rispetto della tutela della privacy, e di facilitare la gestione dell'evoluzione dello stato nel tempo dei processi sanitari. Nel rispetto della normativa vigente, le informazioni sanitarie devono essere rese disponibili sia per gli usi primari, quali assistenza ed emergenza, che per gli usi secondari, ossia per scopi amministrativi e di governo.

L'Infrastruttura tecnologica del FSE deve essere compatibile con le soluzioni architetturali locali già sviluppate in una visione orientata verso un unico modello di infrastruttura federata. Inoltre, essa deve recepire i requisiti infrastrutturali necessari alla interoperabilità funzionale e semantica oggetto dei progetti nazionali ed europei.

Allo scopo di rispettare i requisiti descritti, il modello architetturale di InFSE deve rispecchiare, nel suo complesso, un'architettura orientata ai servizi (Service-Oriented Architecture, SOA), la quale, a sua volta, deve essere posta al di sopra delle infrastrutture tecnologiche del Sistema Pubblico di Connettività (SPC) per la cooperazione applicativa tra Pubbliche Amministrazioni.

Questo documento presenta il modello architetturale di InFSE, che prevede nodi di primo livello (nodi regionali) e nodi di secondo livello (nodi locali), e le sue componenti base. Per ognuna di tali componenti, sono descritte le funzionalità offerte e la loro architettura.

3.1 Versioning

Titolo	InFSE – Linee guida
Data	30/07/2012
Versione	1.2
Stato	DEF

Storia delle principali revisioni:

Versione	Status	Data	Descrizione Modifica
1.2	DEF	30/07/2012	Aggiornati alcuni diagrammi di sequenza. Estesa la descrizione delle componenti. Commenti minori.
1.1	DEF	26/10/2010	Aggiunti i capitoli "Presentazione del documento"e "Sintesi". Altre correzioni minori sparse.
1.0	Prima versione	15/06/2010	Prima versione rilasciata.

4 Termini e acronimi

Acronimo	Termine
CDA	Clinical Document Architecture
DICOM	Digital Imaging and COmmunications in Medicine
ebXML	Electronic Business eXtensible Markup Language
EIS	Entity Identification Service
epSOS	European Patients Smart Open Services
FSE	Fascicolo Sanitario Elettronico
HIMSS	Healthcare Information and Management Systems Society
HL7	Health Level Seven
HSSP	Healthcare Services Specification Project
IHE	Integrating the Healthcare Enterprise
InFSE	Infrastruttura tecnologica del Fascicolo Sanitario Elettronico
IPSE	Sperimentazione di un sistema per l'Interoperabilità europea e nazionale delle soluzioni di fascicolo sanitario elettronico: componenti Patient Summary e ePrescription
MTOM	Message Transmission Optimization Mechanism
OMG	Object Management Group

PA	Pubblica Amministrazione
PAP	Policy Administration Point
PDD	Porta di Dominio
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point
RFP	Request for Proposal
RLUS	Retrieve, Locate and Update Service
RSNA	Radiological Society of North America
SAML	Security Assertion Markup Language
SFM	Service Functional Model
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol
SSN	Servizio Sanitario Nazionale
SPC	Sistema Pubblico di Connettività
SSO	Single Sign-On
STM	Service Technical Model
TSE	Tavolo di Sanità Elettronica

URI	Universal Resource Identifier
XACML	eXtensible Access Control Markup Language
XCA	Cross-Enterprise Community Access
XDS	Cross-Enterprise Document Sharing
XEIS	Cross Domain Entity Identification Service
XML	eXtensible Markup Language
XOP	XML-binary Optimized Packaging
WSDL	Web Services Description Language
WS-MEX	Web Services Metadata Exchange

5 Contesto internazionale e nazionale

5.1 Healthcare Services Specification Project (HSSP)

Il progetto HSSP [1] nasce nel gennaio 2005 dalla collaborazione di Health Level Seven (HL7) con Object Management Group (OMG). Gli obiettivi principali del progetto sono quelli di incentivare l'adozione e l'uso di servizi "plug-and-play" standardizzati da parte dei vendors di prodotti software sanitari e di facilitare lo sviluppo di un insieme di interfacce standard implementabili che supportino specifiche di servizi condivise.

Le specifiche seguono uno specifico processo di stesura di seguito descritto.

Il ruolo principale di HL7 nella formulazione delle specifiche è quello di identificare i servizi candidati per la standardizzazione, definire i requisiti funzionali ed i criteri di conformità in forma di specifiche Service Functional Model (SFM) e adottare queste ultime come standard HL7 ballottati.

A partire dalle SFM specificate da HL7, OMG ha il compito di sviluppare le Requests for Proposals (RFPs), che sono alla base del processo di standardizzazione di OMG. Questo processo consente ai vendors di proporre soluzioni che soddisfino i requisiti obbligatori e opzionali espressi nella RFP, lasciando flessibilità di progettazione ai proponenti e flessibilità di implementazione agli utenti dello standard.

Il risultato di tale collaborazione è una RFP Submission, nota nel processo HSSP come Service Technical Model (STM).

In sintesi, le SFM di HL7 specificano i requisiti funzionali di un servizio, le RFP di OMG specificano i requisiti tecnici di un servizio ed una STM rappresenta il modello tecnico risultante.

Nell'ambito del progetto HSSP, sono state attivate una serie di attività. Quelle di maggior rilievo sono descritte in sintesi nei prossimi paragrafi.

5.1.1 Retrieve, Locate and Update Service (RLUS)

L'obiettivo del progetto RLUS è quello di fornire una serie di interfacce attraverso cui i sistemi informativi possono accedere alle informazioni e gestirle. Tali interfacce permettono di localizzare, accedere e aggiornare i dati sanitari, indipendentemente

dalla strutturazione dei dati, dagli aspetti di sicurezza o dai meccanismi di delivery sottostanti. RLUS infatti non intende rimpiazzare i sistemi o le implementazioni esistenti, ma creare un'interfaccia standard per un layer orientato ai servizi al fine di esporre le risorse di una organizzazione.

Le componenti di sistema più importanti sono le seguenti:

- *Organizational Resource Registry*: contiene informazioni per la localizzazione di informazioni sanitarie;
- *Local Resource Service (RLUS)*: funge da mediatore per le transazioni di locazione, recupero e aggiornamento;
- *Cross-Organizational RLUS*: rappresenta un'implementazione tra più organizzazioni RLUS;
- *Organizational Resource Repository*: memorizza Resource Sources;
- *Resource Source*: rappresenta informazioni, dati ed altri oggetti di interesse ai consumatori RLUS;
- *User System*: rappresenta un qualsiasi sistema informativo sanitario che interagisce con l'utente e si interfaccia con le componenti RLUS.

5.1.2 Entity Identification Service (EIS)

Il progetto EIS ha come obiettivo quello di produrre specifiche tecniche che definiscono le interfacce di servizio per l'identificazione univoca dei pazienti nell'ambito di diversi sistemi informativi gestiti da una singola organizzazione o da più organizzazioni cooperanti. Inoltre, le specifiche prodotte nell'ambito del progetto EIS forniscono le basi per identificare univocamente altre entità all'interno del dominio, tra cui provider individuali e istituzionali, dispositivi medicali, etc.

Il servizio EIS, quindi, fornisce funzionalità per definire, aggiornare e gestire l'identità delle entità. Inoltre, il servizio EIS si interfaccia con i sistemi informativi presenti all'interno delle strutture sanitarie, quali ad esempio Hospital Information System (HIS), Radiology Information System (RIS), Cardiology Information System (CIS), etc.

Un'istanza del servizio EIS che opera su più domini prende il nome di Cross Domain Entity Identification Service (XEIS) e permette di associare una singola entità inter-dominio a più identificatori locali a domini provvisti di servizi EIS. XEIS gestisce anche l'accesso e la gestione degli identificatori dei domini locali.

5.2 Integrating the Healthcare Enterprise (IHE)

IHE [2] è un'iniziativa internazionale nata nel 1998 dalla collaborazione di Radiological Society of North America (RSNA) e Healthcare Information and Management Systems Society (HIMSS) atta a supportare l'integrazione di sistemi informativi sanitari sulla base di standard esistenti (soprattutto DICOM e HL7).

IHE definisce una serie di profili di integrazione, ognuno dei quali mira a risolvere problematiche relative ad uno specifico dominio. In questo contesto, i profili di integrazione più interessanti sono Cross-Enterprise Document Sharing (XDS) e Cross-Community Access (XCA).

Il profilo XDS fornisce un insieme di linee guida per la creazione di un'infrastruttura tecnologica per la condivisione di documenti clinici all'interno di un *affinity domain*. Un affinity domain è un insieme di strutture sanitarie che decidono di cooperare sulla base di infrastrutture e politiche condivise. Questo profilo è gestito da un insieme di attori: più *Document Source*, che producono i documenti clinici, più *Document Repository*, che memorizzano i documenti clinici in maniera persistente, un *Document Registry*, che indicizza i documenti clinici, più *Document Consumer*, che reperiscono i documenti clinici, un *Patient Identity Source*, che assegna un identificatore univoco degli assistiti e mantiene le informazioni anagrafiche. Le interazioni tra gli attori avvengono sotto forma di transazioni. In particolare, la seconda versione del profilo, denominata XDS.b, utilizza gli standard più recenti (ebXML 3.0, SOAP 1.2, MTOM/XOP).

Il profilo XCA definisce un insieme di linee guida per lo scambio documentale tra differenti *community*. Una community è un insieme di strutture sanitarie che condividono politiche e protocolli di comunicazione. Rispetto a XDS, questo profilo aggiunge due tipi di attori: *Initiating Gateway*, che gestisce tutte le transazioni in uscita da una community inoltrandole ad altre community, e *Responding Gateway*, che gestisce tutte le transazioni in ingresso ad una community consegnandole agli attori interni. È interessante notare che il profilo XCA consente di far interoperare tra loro community eterogenee, a prescindere se esse siano basate o meno sul profilo XDS.

5.3 European Patients Smart Open Services (epSOS)

Il progetto epSOS [3] ha lo scopo di progettare, realizzare e valutare una infrastruttura di servizio a supporto dell'interoperabilità transfrontaliera tra i sistemi di Fascicolo Sanitario Elettronico in Europa.

Il consorzio è composto da 47 beneficiari provenienti da 20 Stati Membri dell'Unione Europea e 3 Stati non appartenenti all'Unione Europea.

Il progetto epSOS è finalizzato a garantire la continuità di cura ai pazienti che si trovano all'estero, mirando a risolvere una serie di problemi riguarduanti in particolar modo l'erogazione di farmaci importanti che un paziente ha perso o dimenticato, la comunicazione di situazioni cliniche a medici che parlano una lingua straniera, la diagnosi di malattie e la prescrizione di opportuni farmaci attraverso una sintetica conoscenza della storia del paziente.

In particolare, l'obiettivo di epSOS è quello di assicurare che le soluzioni nazionali di Fascicolo Sanitario Elettronico possano interoperare tra di loro, permettendo a professionisti sanitari di accedere ai dati di un paziente da un altro Paese nel proprio linguaggio, usando differenti tecnologie e sistemi.

L'idea è quella di sviluppare un framework tecnologico per l'eHealth ed un'infrastruttura ICT al fine di permettere un accesso sicuro alle informazioni sanitarie dei pazienti localizzate in Paesi distinti, realizzando in particolare servizi per la gestione di Patient Summary e di ePrescription.
Al fine di facilitare l'interoperabilità dei Patient Summary e delle ePrescription, è prevista la possibilità di trasferire sia documenti in forma strutturata che in forma destrutturata.

5.4 Sperimentazione di un sistema per l'Interoperabilità europea e nazionale delle soluzioni di fascicolo sanitario elettronico: componenti Patient Summary e ePrescription (IPSE)

Il progetto IPSE [4] nasce da un Accordo tra il Dipartimento per la digitalizzazione della Pubblica Amministrazione e l'innovazione tecnologica, il Ministero della Salute, la Regione Lombardia, in qualità di coordinatore, la Regione Abruzzo, la Regione Molise, la Regione Emilia Romagna, la Regione Toscana, la Regione Umbria, la

Regione Veneto, la Regione Autonoma della Sardegna, la Provincia Autonoma di Trento e l'Agenzia Regionale della Sanità della Regione Autonoma Friuli Venezia Giulia.

Il principale obiettivo del progetto è stato quello di garantire il raccordo con quanto previsto nell'ambito del progetto epSOS, assicurando l'interoperabilità interregionale dei sistemi di sanità elettronica attraverso la condivisione di una definizione comune dei servizi di Patient Summary e di ePrescription, in grado di definire soluzioni compatibili con le realizzazioni già esistenti o in fase di sviluppo e con quanto previsto a livello europeo.

La fase di sperimentazione ha coinvolto opportuni siti pilota, individuati a partire da specifici criteri di selezione, comprendenti l'esistenza di infrastrutture di FSE e l'utilizzo di standard aperti.

6 Requisiti dell'Infrastruttura del FSE

Il modello architetturale dell'Infrastruttura tecnologica del FSE deve essere rispondente a specifiche esigenze progettuali, le più importanti delle quali sono elencate di seguito.

1. *Consentire la localizzazione e la disponibilità delle informazioni sanitarie*
   Le informazioni sanitarie degli assistiti devono essere sempre accessibili da utenti autorizzati, ovunque esse siano memorizzate.
2. *Supportare adeguatamente i processi sanitari*
   L'Infrastruttura deve permettere agli attori del SSN di seguire i processi di cura dei propri assistiti, notificando opportuni eventi clinici ad ogni loro occorrenza.
3. *Supportare la natura federata e decentralizzata del SSN*
   Al fine di rispettare l'autonomia delle Regioni, l'Infrastruttura non deve essere basata su logiche centralizzate, ma su modelli federati.
4. *Consentire una facile integrazione con sistemi e infrastrutture preesistenti*
   L'approccio alla progettazione dell'Infrastruttura deve essere di tipo bottom-up, ossia deve mirare a rendere interoperabili i sistemi e le infrastrutture regionali di sanità elettronica.
5. *Essere basato su standard aperti e sulle tecnologie dell'Internet del futuro*
   L'uso di standard aperti e delle tecnologie dell'Internet del futuro facilita l'interoperabilità delle infrastrutture e dei sistemi regionali e minimizza gli investimenti.
6. *Presentare caratteristiche di scalabilità e modularità*
   L'infrastruttura deve essere modulare e scalabile, al fine di consentirne uno sviluppo incrementale e distribuito.
7. *Fornire caratteristiche di affidabilità*
   La progettazione dell'Infrastruttura deve tenere in considerazione criteri di dependability, che la rendano fault-tolerant e priva di single-point-of-failure.
8. *Fornire adeguate caratteristiche prestazionali*
   L'Infrastruttura deve offrire opportune proprietà prestazionali in termini di accessibilità ai documenti e dati sanitari.

9. *Garantire un elevato livello di sicurezza*

   L'Infrastruttura deve essere capace di gestire gli aspetti di identificazione ed autenticazione degli utenti e delle componenti e di accesso ai documenti.

10. *Essere basato sul Sistema Pubblico di Connettività (SPC)*

    Le interazioni tra le componenti infrastrutturali appartenenti a domini differenti devono avvenire attravero le regole di cooperazione applicativa previste dal Sistema Pubblico di Connettività.

11. *Essere conforme alle indicazioni del Garante della Privacy*

    L'Infrastruttura deve rispettare le normative vigenti in materia di riservatezza e accesso ai dati contenuti nel Fascicolo Sanitario Elettronico.

7 Modello architetturale di InFSE

7.1 Descrizione del modello

Il FSE di un cittadino consiste in una composizione di tutti i documenti sanitari e socio-sanitari inerenti al suo stato di salute. L'Infrastruttura del FSE deve garantire la consultazione di tali documenti, e anche la possibilità di gestire l'evoluzione temporale dello stato sia dei documenti sia dei processi sanitari.

L'Infrastruttura tecnologica del FSE deve integrare tra loro tutte le strutture che a vario titolo concorrano alla produzione (e/o alla consultazione) di eventi concernenti l'interazione del singolo assistito con il SSN. In Figura 1 è descritto il modello architetturale di alto livello di InFSE, attraverso il quale le strutture sanitarie interagiscono tra loro mediante opportuni moduli software (inglobati nel blocco *Componenti Infrastruttura FSE* in figura), localizzati presso le strutture sanitarie o le Regioni.

L'architettura prevede dei nodi di primo livello (nodi regionali) e nodi di secondo livello (nodi locali).

I *nodi regionali* contemplano la presenza di tutte le componenti infrastrutturali del FSE e sono in grado di garantire tutte le funzionalità necessarie al reperimento e alla gestione delle informazioni. I *nodi locali* possono essere funzionalmente equivalenti ai nodi regionali (nodi locali completi) o prevedere solo alcune componenti infrastrutturali (nodi locali assistiti).

Il modello architetturale prevede l'adozione del Sistema Pubblico di Connettività per le comunicazioni interregionali. Conseguentemente, per ogni nodo regionale è previsto il collegamento mediante una Porta di Dominio (PDD), mentre un nodo locale può esporsi sia direttamente mediante una propria PDD, o indirettamente attraverso un nodo regionale.

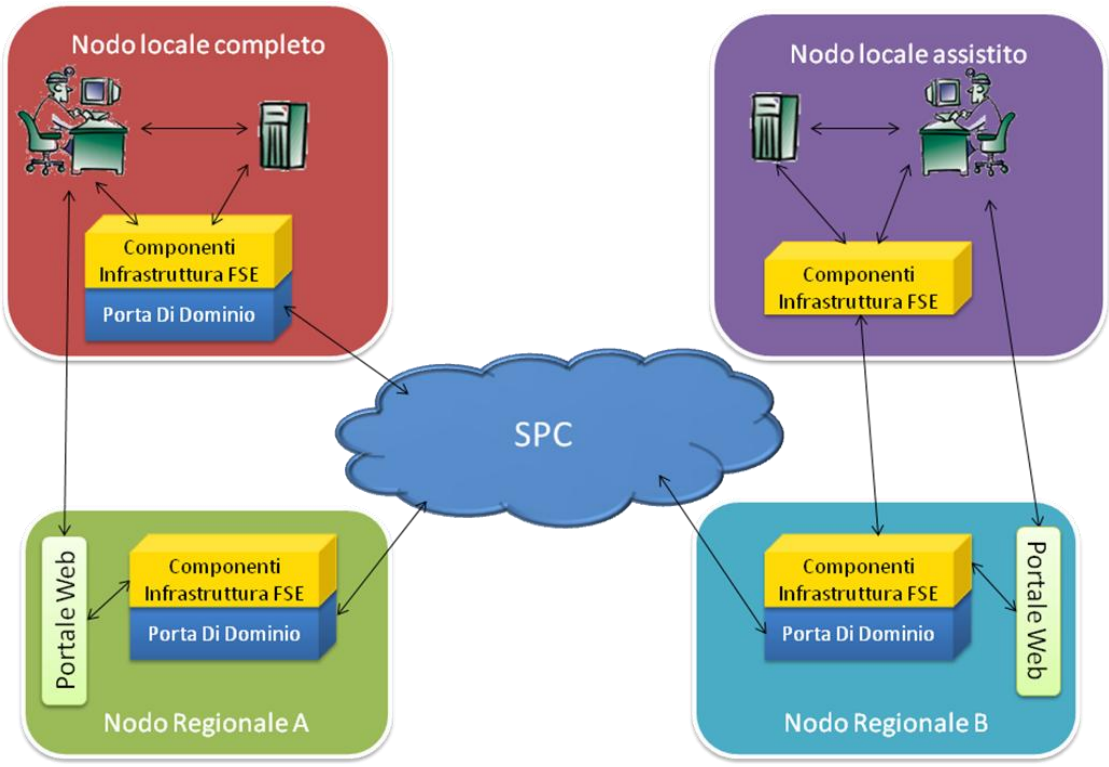


Figura 1. Modello architetturale di InFSE

7.2 Architettura multi-layers

Il modello architetturale di InFSE rispecchia, nel suo complesso, un'architettura orientata ai servizi (Service-Oriented Architecture, SOA). Tale architettura è organizzata in più livelli (layers) ed è mostrata in Figura 2.

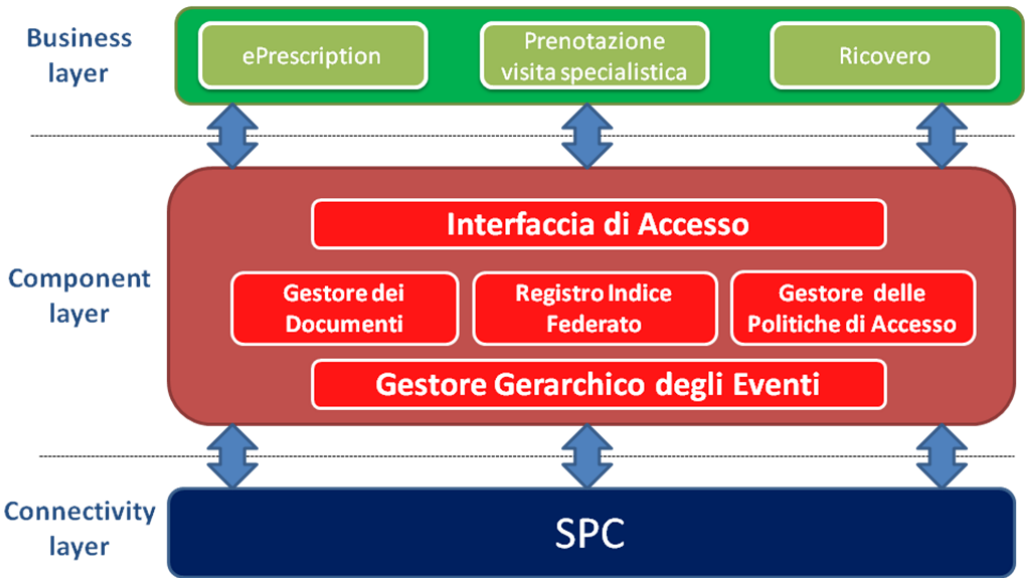


Figura 2. Architettura multi-layers di InFSE

Il *Connectivity layer* è rappresentato dal Sistema Pubblico di Connettività per la cooperazione applicativa tra le Pubbliche Amministrazioni mediante busta eGov. In particolare, tutte le interazioni interregionali devono avvenire attraverso una Porta di Dominio.

Il *Component layer* è costituito dalle componenti infrastrutturali del FSE, che sono descritte in dettaglio nel seguito del documento.

Infine, il *Business layer* definisce i servizi di supporto ai processi sanitari quali, ad esempio, l'ePrescription, la prenotazione di una visita specialistica, il Patient Summary, etc.

8 Architettura delle componenti di InFSE

8.1 Interfaccia di Accesso

Questa componente funge da interfaccia all'Infrastruttura del Fascicolo Sanitario Elettronico. Essa deve essere presente presso ogni nodo regionale e, opzionalmente, presso i nodi locali.

L'*Interfaccia di Accesso* di un nodo regionale è la componente che riceve tutte le richieste avanzate dagli attori regionali (es. Medico di Medicina Generale) e dalle *Interfacce di Accesso* dei nodi delle altre Regioni o Province Autonome.

A fronte di una richiesta, l'*Interfaccia di Accesso* del nodo regionale inizia ad orchestrare una serie di interazioni con le altre componenti dell'Infrastruttura al fine di soddisfare la richiesta.

Le installazioni presso i nodi locali hanno principalmente lo scopo di costituire il punto di accesso all'Infrastruttura per i sistemi informativi locali.

In tal caso, quindi, se da un lato l'*Interfaccia di Accesso* di un nodo locale offre i servizi di accesso all'Infrastruttura agli attori attivi presso il nodo locale, dall'altro ha il compito di intercettare gli eventi prodotti presso le componenti legacy del nodo locale che possono essere di interesse per l'Infrastruttura stessa.

Questi ultimi saranno perciò notificati ai registri ed ai nodi broker regionali.

8.1.1 Attori e ruoli

È possibile identificare i seguenti attori e ruoli negli scenari di interazione con la componente:

- *DocumentProducer* – È l'entità capace di produrre nuovi documenti o aggiornare documenti preesistenti;
- *DocumentConsumer* – È l'entità capace di consumare documenti mantenuti nel FSE;
- *Publisher* – È l'entità capace di pubblicare nuovi eventi per conto di un produttore;
- *NotificationProducer* – È l'entità capace di produrre nuovi eventi;
- *Subscriber* – È l'entità capace di sottoscrivere eventi per conto di un consumatore;
- *NotificationConsumer* – È l'entità capace di consumare eventi;
- *FederationManager* – È l'entità capace di gestire le federazioni;
- *Event Producer* – È l'entità capace sia di pubblicare che di produrre nuovi eventi;
- *Event Consumer* – È l'entità capace di sottoscrivere e consumare eventi;
- *Entry Producer* – È l'entità capace di produrre nuove entry per i registri;
- *Entry Consumer* – È l'entità capace di consumare entry dei registri;
- *Registry Node* – È un nodo capace di entrare in federazione con altri registri;
- *Broker Node* – È un nodo capace di entrare in federazione con altri nodi brokers.

Le relazioni fra attori e ruoli sono rappresentate in Figura 3.

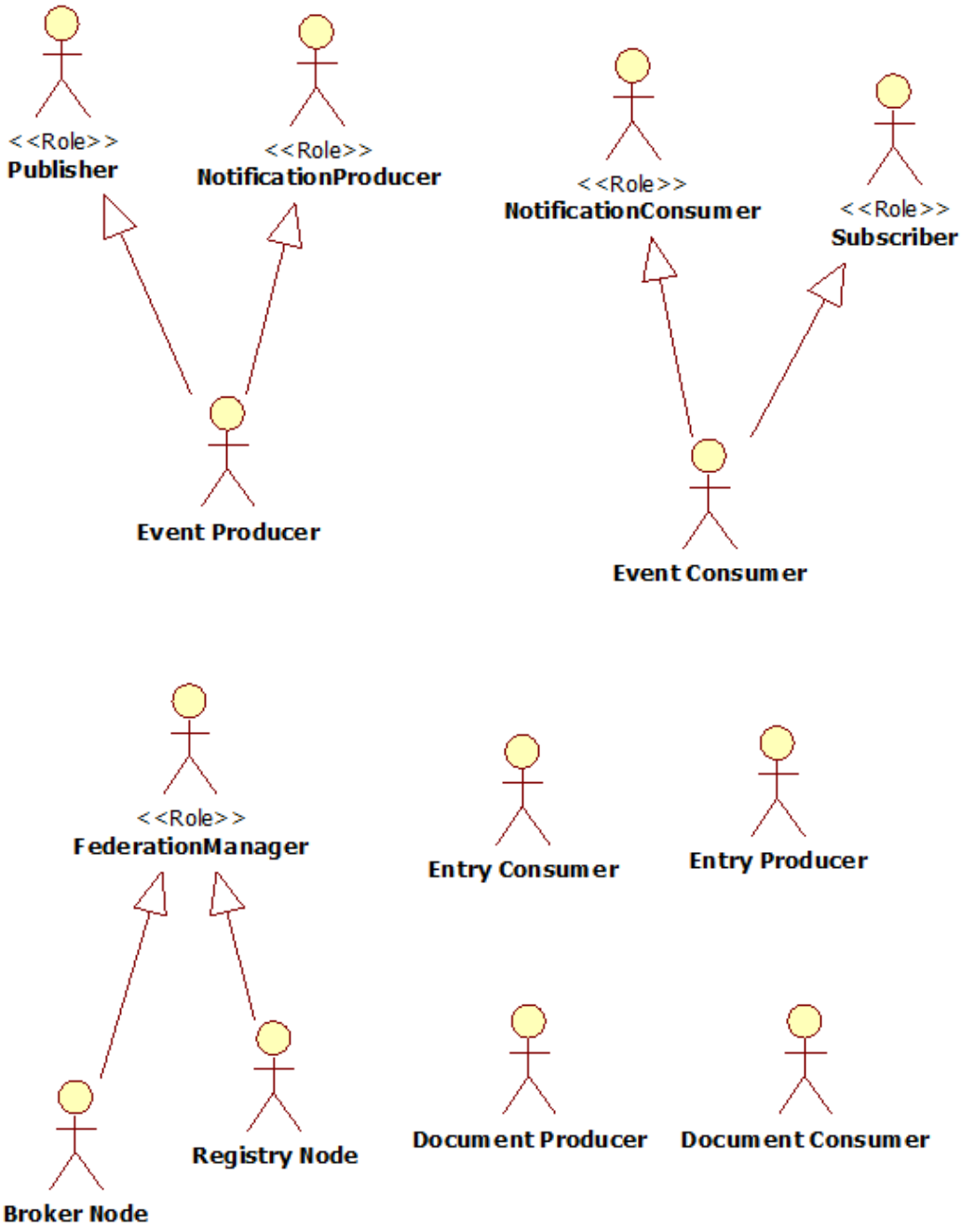


Figura 3. Attori e ruoli

8.1.2 Casi d'uso

Si definiscono i seguenti casi d'uso per la componente *Interfaccia di Accesso*.

8.1.2.1 Casi d'uso per la gestione dei documenti

- *AddDocument* – È la funzionalità che permette di aggiungere un nuovo documento al fascicolo;
- *UpdateDocument* – È la funzionalità che permette di aggiornare un documento preesistente. L'aggiornamento può riguardare sia lo stato che la versione del documento;
- *RetrieveDocument* – È la funzionalità che permette di ottenere un documento dal gestore.

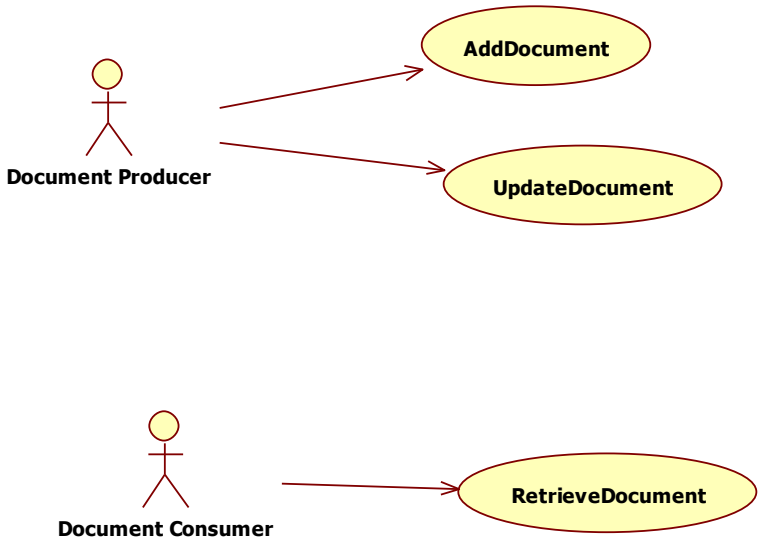


Figura 4. Casi d'uso per la gestione dei documenti

8.1.2.2 Casi d'uso per la gestione degli eventi

- *Register* – È la funzionalità che permette la pubblicazione di uno o più topic (categoria che correla logicamente un insieme di eventi). Attraverso questa funzionalità un publisher comunica al fascicolo la possibilità di generare nuovi eventi;
- *RegisterHierarchy* – È la funzionalità che consente la registrazione di un'intera gerarchia da parte di un producer;

- *DestroyRegistration* – È la funzionalità che permette l'eliminazione di una registrazione;
- *CreateTopic* – È la funzionalità che permette di creare una nuova classe di eventi (topic);
- *ArchiveTopic* – È la funzionalità che permette di archiviare una classe di eventi. Dal momento dell'archiviazione, il topic non è più disponibile per la notifica di nuovi eventi; piuttosto, è ancora possibile ottenere gli eventi archiviati;
- *CreateHierarchy* – È la funzionalità che permette la creazione di una nuova gerarchia;
- *ArchiveHierarchy* – È la funzionalità che permette di archiviare un'intera gerarchia e tutti gli eventi prodotti ad essa correlati;
- *CreateAssociation* – È la funzionalità che consente la creazione dinamica di un'associazione tra topic all'interno di una gerarchia;
- *AddContentFilter* – È la funzionalità che consente la creazione di un filtro sul contenuto degli eventi per una maggiore selettività nella consegna di eventi;
- *RemoveAssociation* – È la funzionalità che permette la rimozione di un'associazione tra topic all'interno di una gerarchia;
- *Notify* – È la funzionalità che permette la notifica di un evento al gestore;
- *GetAllTopics* – È la funzionalità che permette di ottenere la lista di tutti i topic accessibili;
- *GetAllHierarchies* – È la funzionalità che permette di ottenere le gerarchie attive attraverso opportuni parametri di filtraggio;
- *Subscribe* – È la funzionalità che permette la sottoscrizione di uno o più topic;
- *SubscribeHierarchy* – È la funzionalità che permette la sottoscrizione di un'intera gerarchia;
- *Renew* – È la funzionalità che permette di rinnovare una sottoscrizione;
- *Unsubscribe* – È la funzionalità che permette la rimozione dall'elenco dei sottoscrittori;
- *GetCurrentMessage* – È la funzionalità che permette di ottenere l'ultimo messaggio notificato per un determinato topic;
- *RetrieveEvents* – È la funzionalità che permette di ottenere tutti gli eventi notificati in relazione ad un determinato topic;
- *RetrieveHierarchy* – È la funzionalità che permette di ottenere tutti gli eventi notificati in relazione ad una determinata gerarchia.

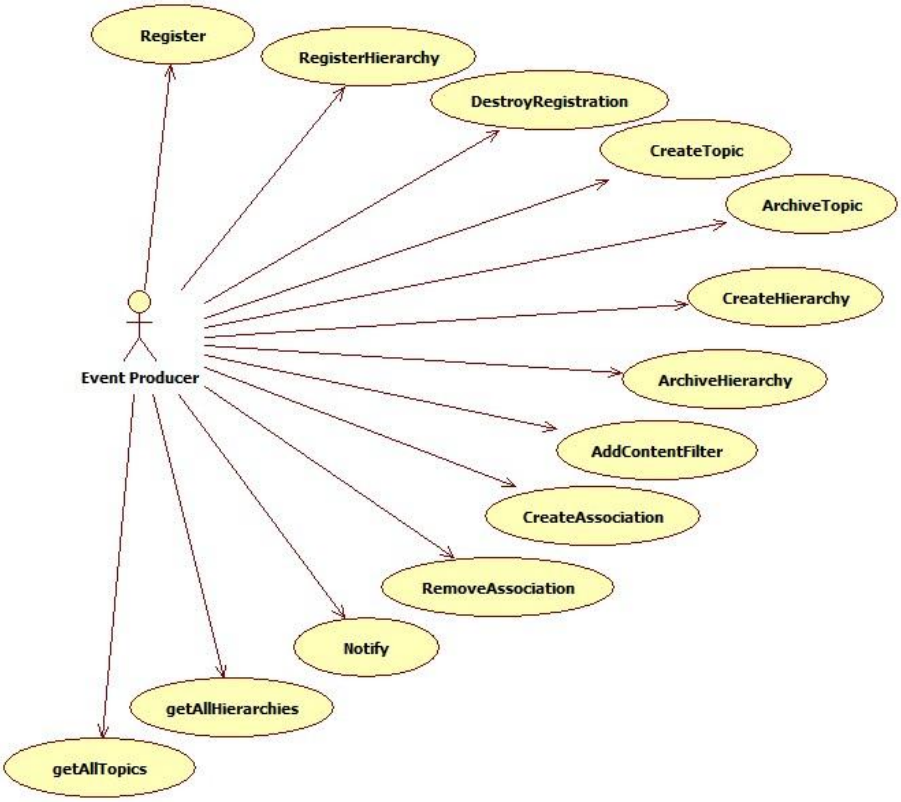


Figura 5. Casi d'uso per il produttore di eventi

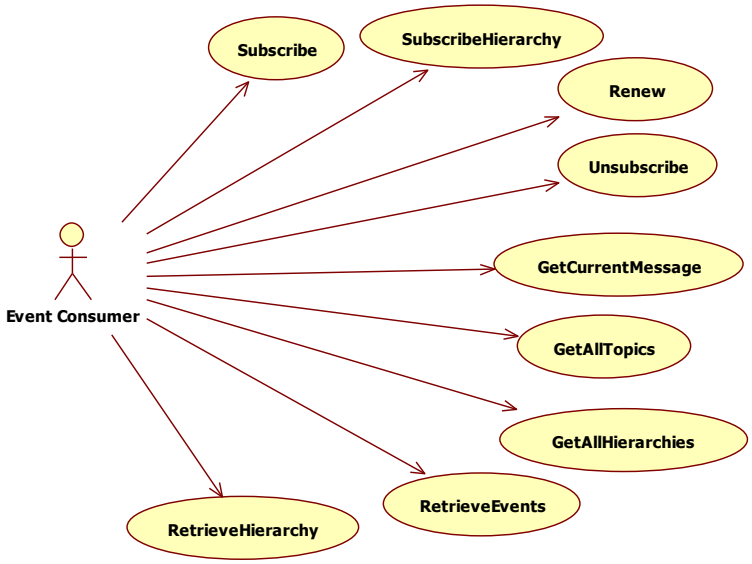


Figura 6. Casi d'uso per il consumatore di eventi

8.1.2.3 Casi d'uso per la gestione degli indici ai documenti

- *RegisterEntry* – È la funzionalità che consente di inviare metadati inerenti ad uno o più documenti ad un registro;
- *UpdateEntry* – È la funzionalità che permette l'aggiornamento dei metadati inerenti ad uno o più documenti;
- *ApproveEntry* – È la funzionalità che permette di approvare i metadati presenti in un registro sulla base di specifici criteri;
- *DeprecateEntry* – È la funzionalità che permette di specificare che particolari metadati sono obsoleti;
- *UndeprecateEntry* – È la funzionalità che permette di specificare che particolari metadati non sono più obsoleti;
- *RemoveEntry* – È la funzionalità che permette di rimuovere specifici metadati da un registro (deve essere utilizzata solo in particolari condizioni, ad es. per eliminare metadati errati);
- *RelocateEntry* – È la funzionalità che permette di rilocare specifici metadati in un altro registro;
- *SubscribeEvent* – È la funzionalità che permette di sottoscrivere l'interesse a ricevere eventi relativi alla sincronizzazione dei registri;
- *UnsubscribeEvent* – È la funzionalità che permette la rimozione di una sottoscrizione;
- *Query* – È la funzionalità che permette di sottoporre query ad uno specifico registro o ad una federazione di registri.

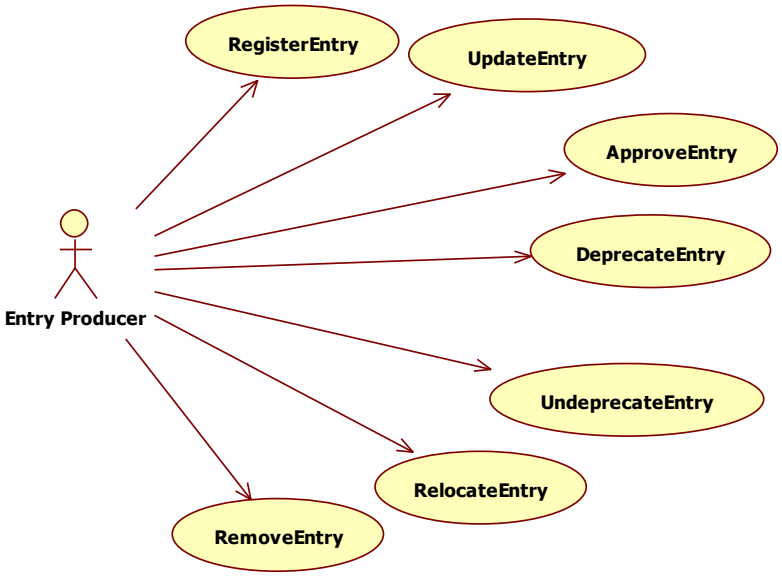


Figura 7. Casi d'uso per il produttore di metadati

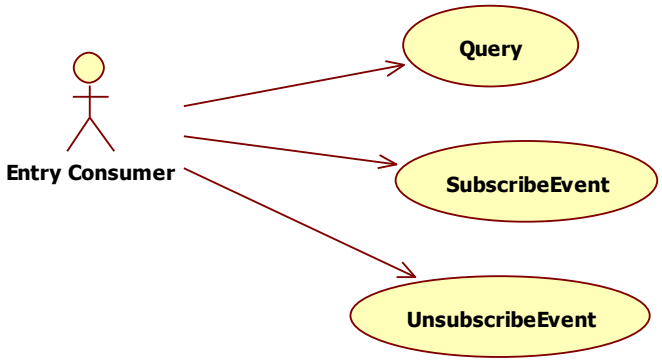


Figura 8. Casi d'uso per il consumatore di metadati

8.1.2.4 Casi d'uso per la gestione delle federazioni

- *CreateFederation* – È la funzionalità che consente la creazione di una nuova federazione;
- *JoinFederation* – È la funzionalità che consente ad un nodo l'ingresso in federazione;
- *Connect* – È la funzionalità che permette ad un nodo di connettersi logicamente ad un altro nodo della federazione;
- *LeaveFederation* – È la funzionalità che consente ad un nodo di uscire dalla federazione;
- *DissolveFederation* – È la funzionalità che permette la rimozione logica di una federazione;
- *SetCoordinator* – È la funzionalità che permette di specificare il nodo coordinatore all'interno di una federazione;
- *RouteEvent* – È la funzionalità che consente la propagazione di un evento tra i nodi della federazione di broker.

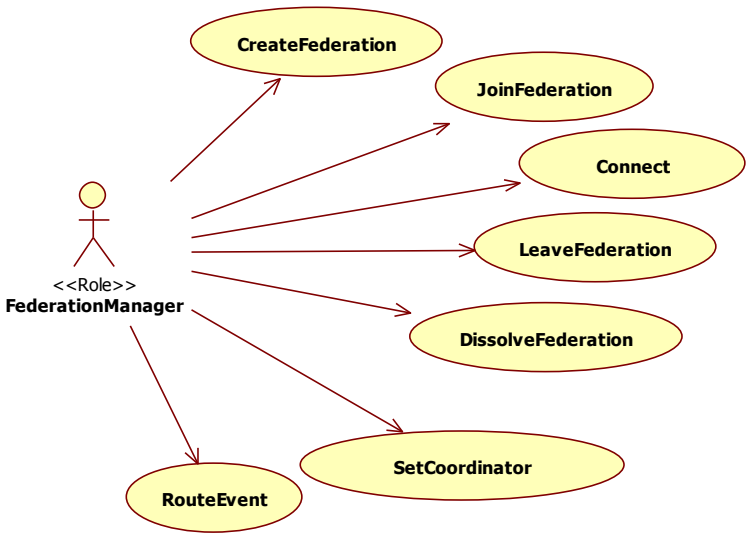


Figura 9. Casi d'uso per il manager della federazione

8.1.3 Architettura della componente

L'*Interfaccia di Accesso* fornisce i meccanismi di accesso all'Infrastruttura del Fascicolo Sanitario Elettronico.

Essa supporta le seguenti interfacce di sistema:

- *IDocument* è l'interfaccia per l'accesso ed il caricamento di documenti nel FSE;
- *IEvent* è l'interfaccia per la gestione degli eventi, classi di eventi e gerarchie di classi;
- *IEntry* è l'interfaccia per la gestione dei riferimenti ai documenti nel registro indice federato;
- *IBrokerFederation* è l'interfaccia per la gestione della federazione di nodi broker di eventi;
- *IRegistryFederation* è l'interfaccia per la gestione della federazione di registri.

L'architettura delle interfacce e le loro dipendenze dalle interfacce di livello business delle altre componenti dell'Infrastruttura sono mostrate in Figura 10, Figura 11 e Figura 12.

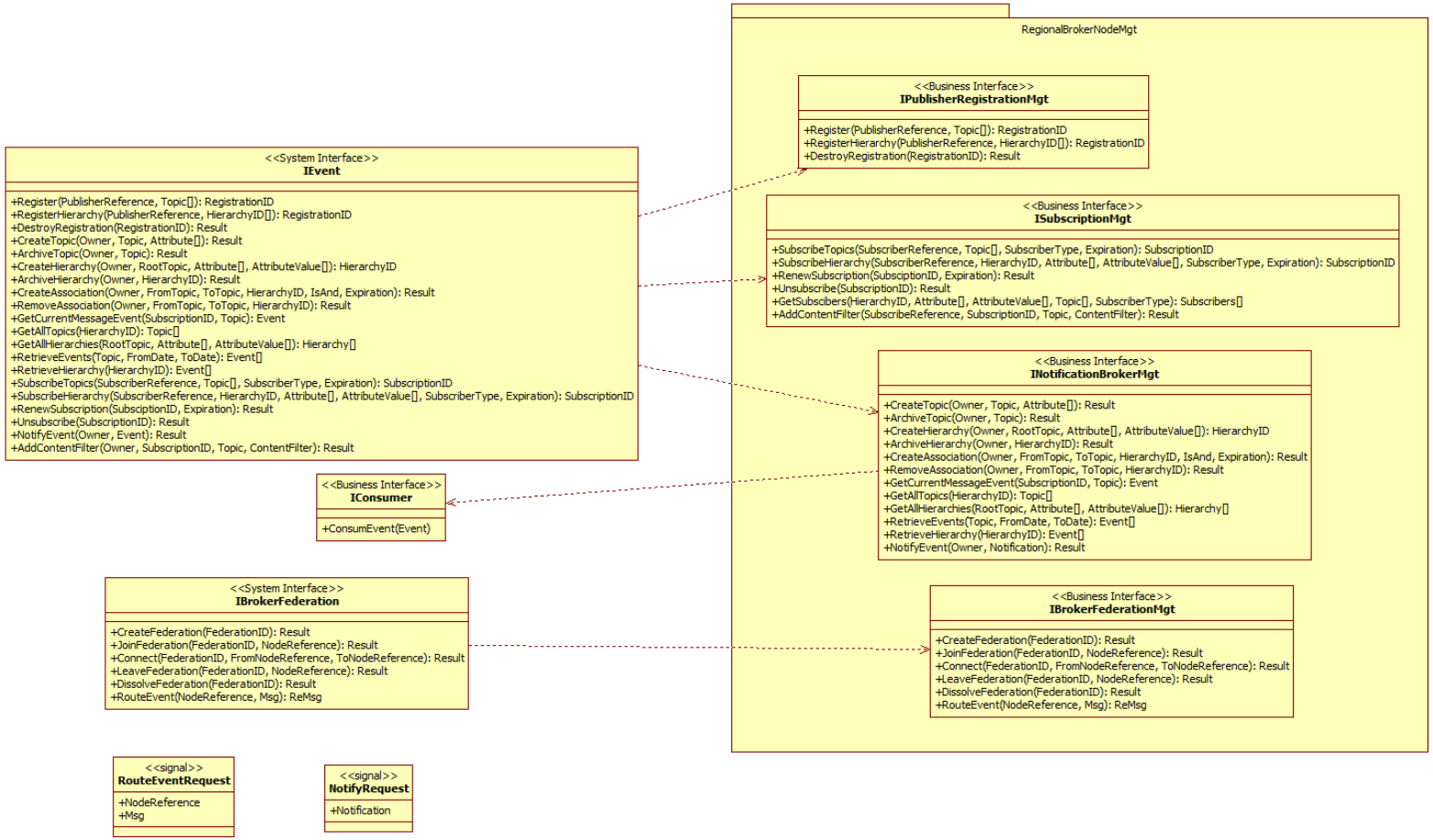


Figura 10. Interfacce IEvent e IBrokerFederation

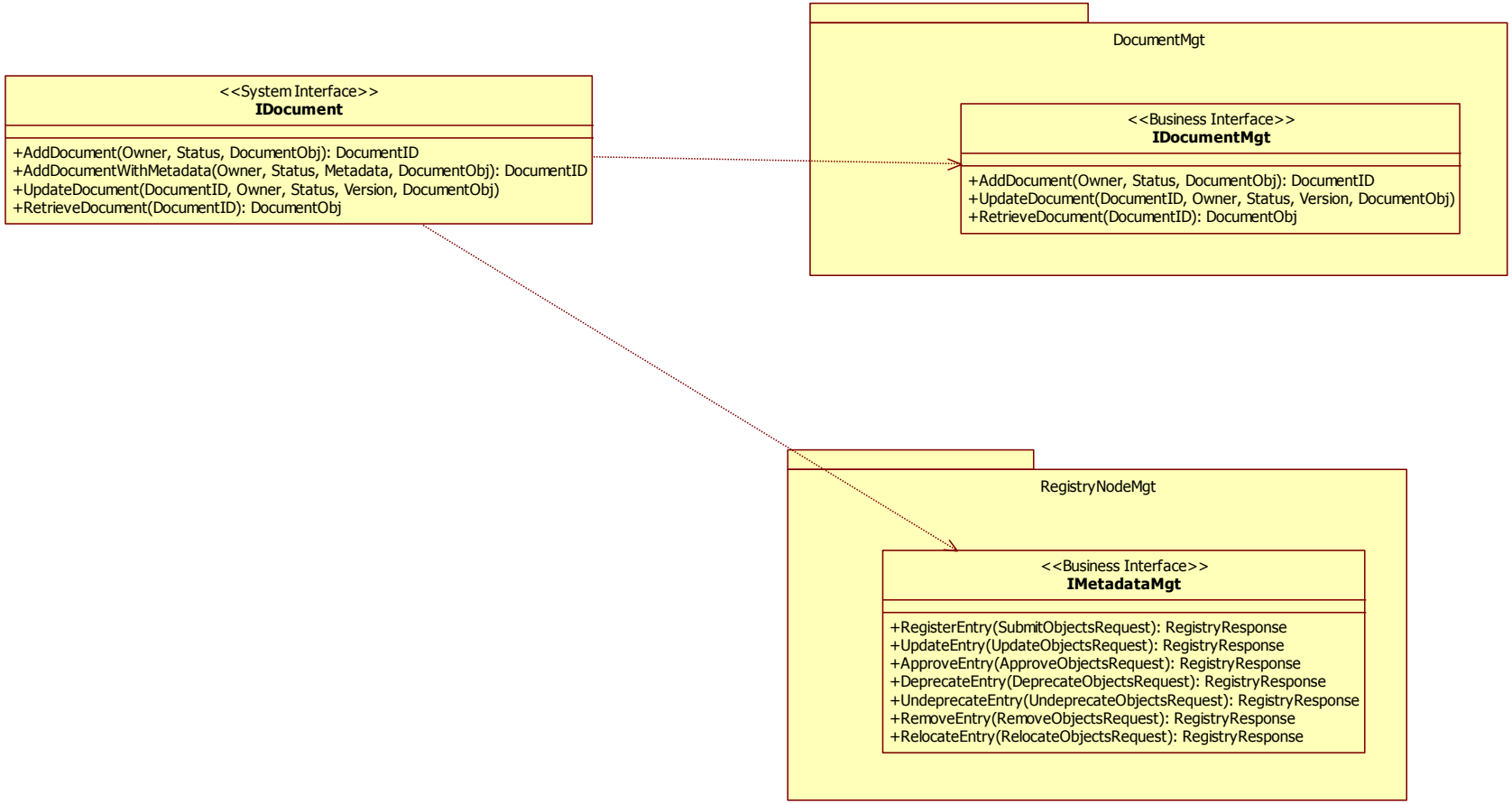


Figura 11. Interfaccia IDocument

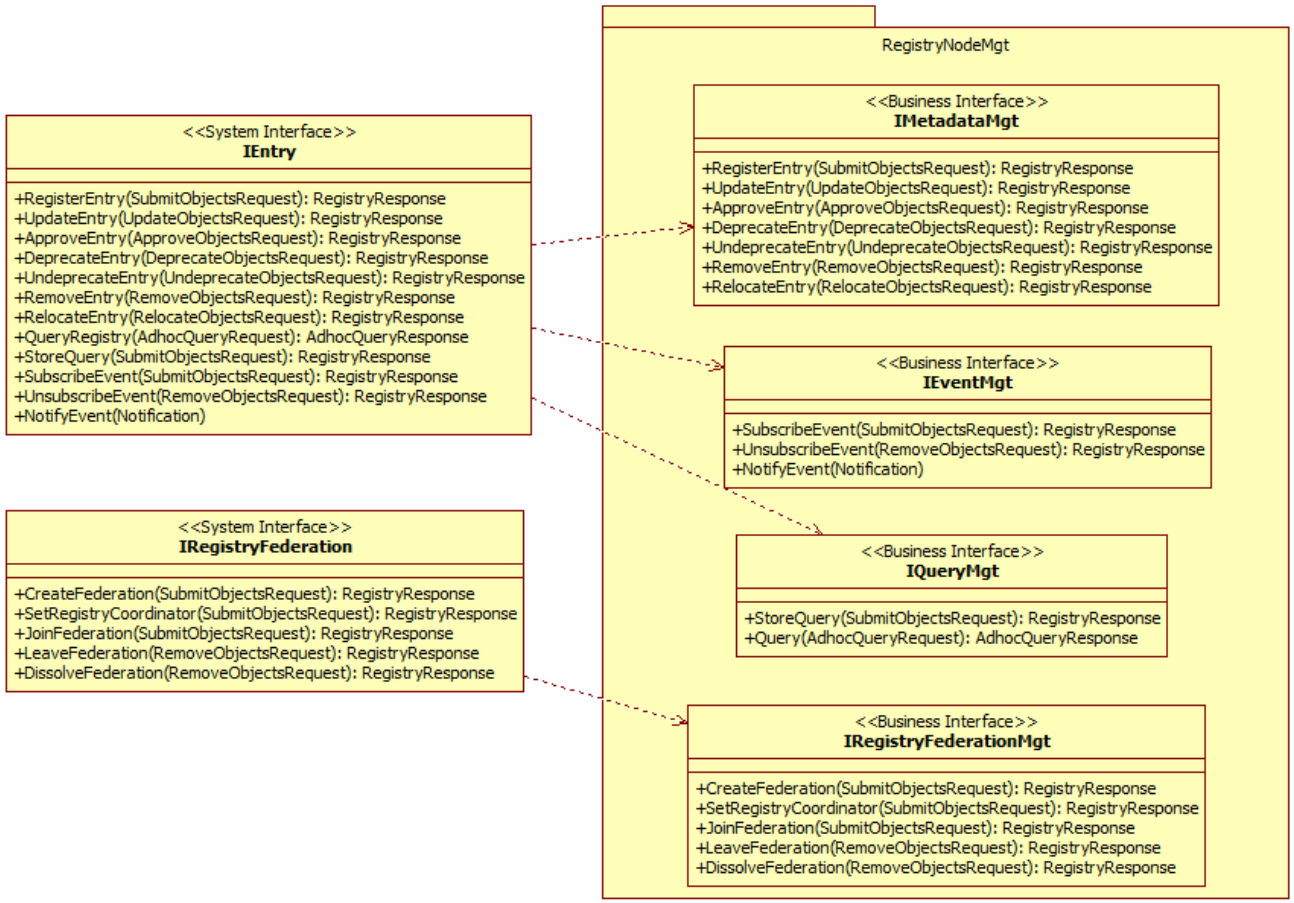


Figura 12. Interfacce IEntry e IRegistryFederation

8.1.4 Integrazione con il Sistema Pubblico di Connettività

Tutte le funzionalità dell'*Interfaccia di Accesso* sono esposte su Porta di Dominio SPC. In particolare, l'interfaccia *IDocument* deve essere esposta da ogni nodo regionale per permettere l'elaborazione di richieste di recupero documenti provenienti da domini extra-regionali, mentre l'interfaccia *IEntry* può essere esposta per gestire la notifica di eventi tra registri.

8.1.5 Scenari d'uso

Di seguito si riportano i principali scenari di interazione con la componente.

8.1.5.1 Creazione documento

Un produttore può creare un nuovo documento per il FSE. L'operazione di creazione, richiesta all'*Interfaccia di Accesso*, comporta l'inserimento del documento in uno dei *Gestori dei Documenti* regionali, la creazione di una entry nel *Registro Indice Regionale* (o in uno dei registri nel caso di federazione locale) e la notifica di un evento alla federazione attraverso il *Nodo Broker Regionale*.

1. Il produttore richiede la creazione di un nuovo documento per il FSE;
2. L'interfaccia *IDocument* seleziona il *Gestore dei Documenti* deputato a memorizzare il nuovo documento e ne richiede l'archiviazione fornendo il documento stesso come parametro d'ingresso;
3. Il *Gestore dei Documenti* restituisce l'identificativo univoco associato al nuovo documento;
4. L'interfaccia *IDocument* richiede la registrazione di una nuova entry nel registro regionale che può essere inoltrata alla federazione nel caso il documento riguardi un assistito di un'altra regione;
5. Il registro restituisce l'esito dell'inserimento;
6. L'interfaccia *IDocument* richiede la notifica di un evento, relativo all'inserimento di nuovo documento, all'interfaccia *IBrokerFederationMgt* del nodo broker regionale;
7. Viene restituito l'ID del nuovo documento.

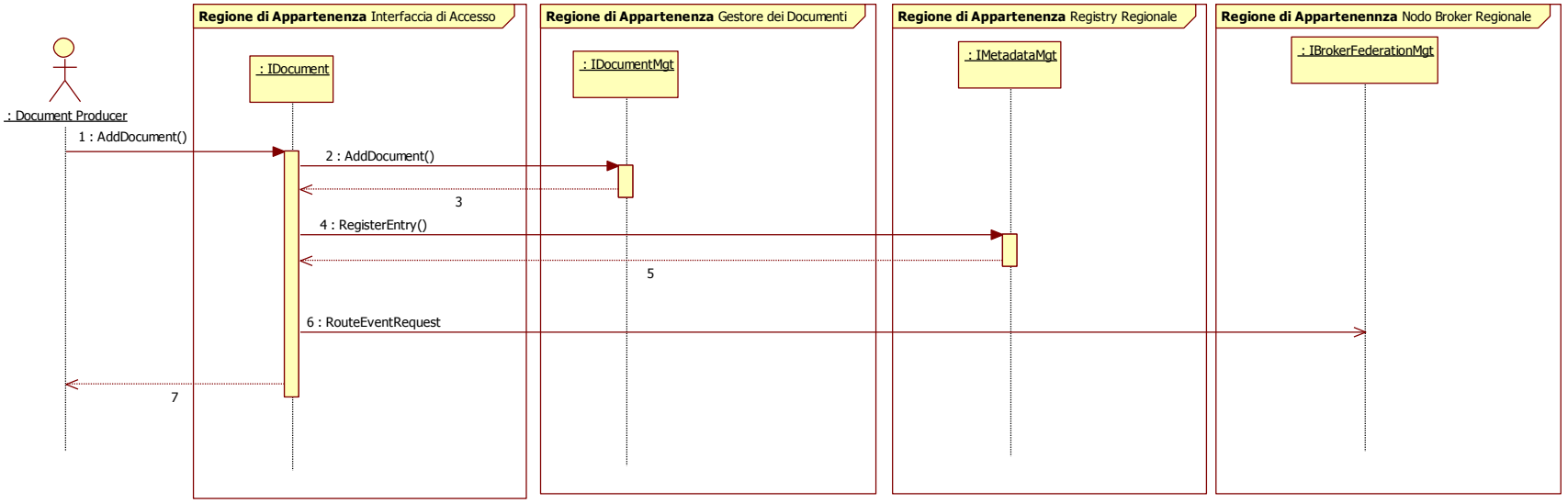


Figura 13. Scenario Creazione documento

8.1.5.2 Aggiornamento documento

Un produttore può richiedere l'aggiornamento di un documento preesistente. L'aggiornamento può riguardare lo stato del documento oppure la sua versione.

1. Il produttore richiede l'aggiornamento di un documento del FSE;
2. L'interfaccia *IDocument* seleziona il *Gestore dei Documenti* deputato a memorizzare il nuovo documento e ne richiede l'aggiornamento;
3. L'interfaccia *IDocument* riceve l'esito dell'aggiornamento da parte del *Gestore dei Documenti*;
4. L'interfaccia *IDocument* richiede l'aggiornamento della entry del *Registro Regionale*;
5. L'interfaccia *IDocument* riceve l'esito dell'aggiornamento dei metadati da parte del *Registro Regionale*;
6. L'interfaccia *IDocument* può richiedere la notifica di un evento, relativo all'aggiornamento del documento, all'interfaccia *IBrokerFederationMgt* del nodo broker regionale;
7. Il controllo viene restituito al produttore del documento.

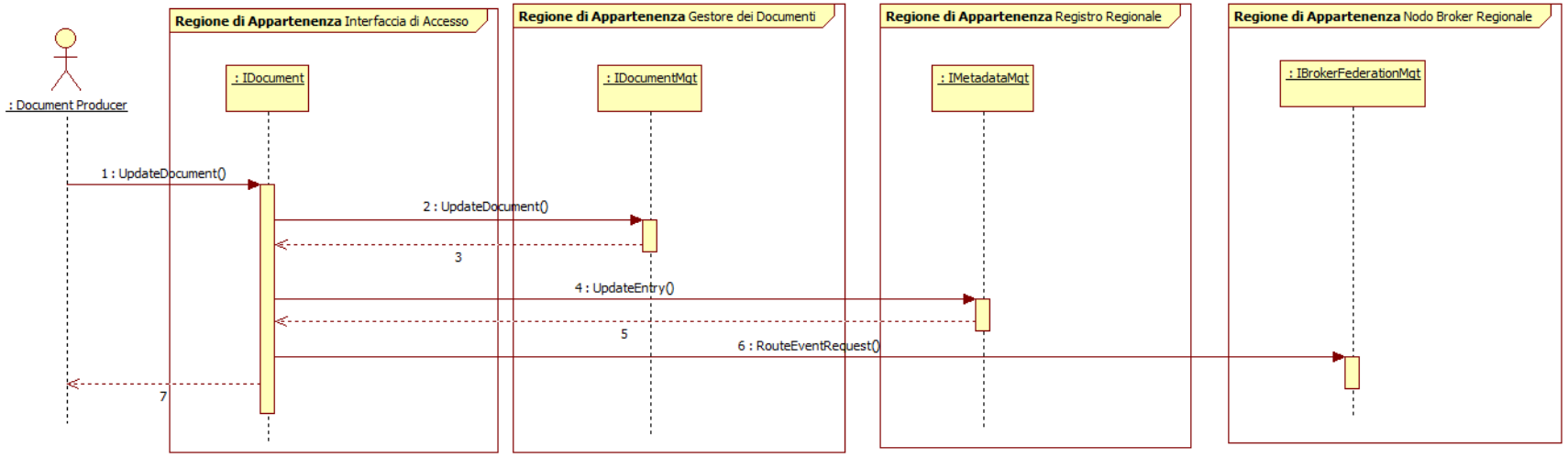


Figura 14. Scenario Aggiornamento documento

8.1.5.3 Recupero documento

Un consumatore può richiedere un documento attraverso l'interfaccia *IDocument*. In generale, è possibile distinguere due casi: il documento è disponibile presso la Regione dove viene inviata la richiesta; il documento è disponibile presso una Regione differente.

8.1.5.3.1 Recupero documento in ambito regionale

1. Il consumatore richiede un documento disponibile nel FSE fornendo il suo identificativo, il quale può essere ottenuto a valle di una operazione di ricerca presso la federazione di registri;
2. L'interfaccia *IDocument* richiede il documento al *Gestore dei Documenti* che lo mantiene, che può essere identificato a partire dall'identificativo fornito;
3. Il *Gestore dei Documenti* restituisce il documento richiesto;
4. L'interfaccia *IDocument* inoltra il documento al richiedente.

8.1.5.3.2 Recupero documento in ambito extra-regionale

1. Il consumatore richiede un documento disponibile nel FSE fornendo il suo identificativo, il quale può essere ottenuto a valle di una operazione di ricerca presso la federazione di registri;
2. L'interfaccia *IDocument* invia la richiesta all'interfaccia *IDocument* esposta dalla Regione che ha in carico il documento, che può essere identificata a partire dall'identificativo fornito;
3. L'interfaccia *IDocument* della Regione contenente il documento richiede quest'ultimo al *Gestore dei Documenti* che lo mantiene, che può essere identificato a partire dall'identificativo fornito;
4. Il *Gestore dei Documenti* restituisce il documento richiesto;
5. L'interfaccia *IDocument* della Regione contenente il documento inoltra il documento all'interfaccia *IDocument* richiedente;
6. L'interfaccia *IDocument* inoltra il documento al consumatore.

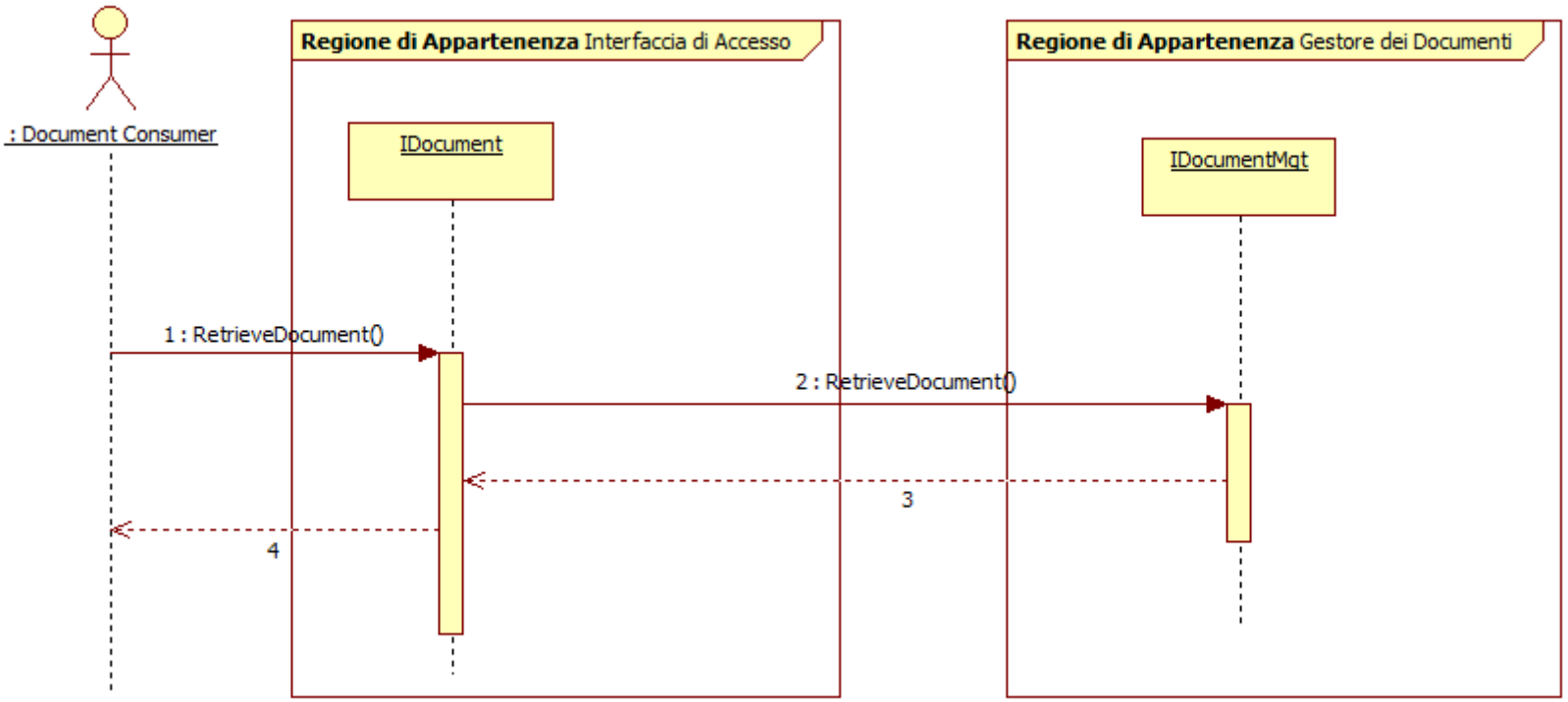


Figura 15. Scenario Recupero documento in ambito regionale

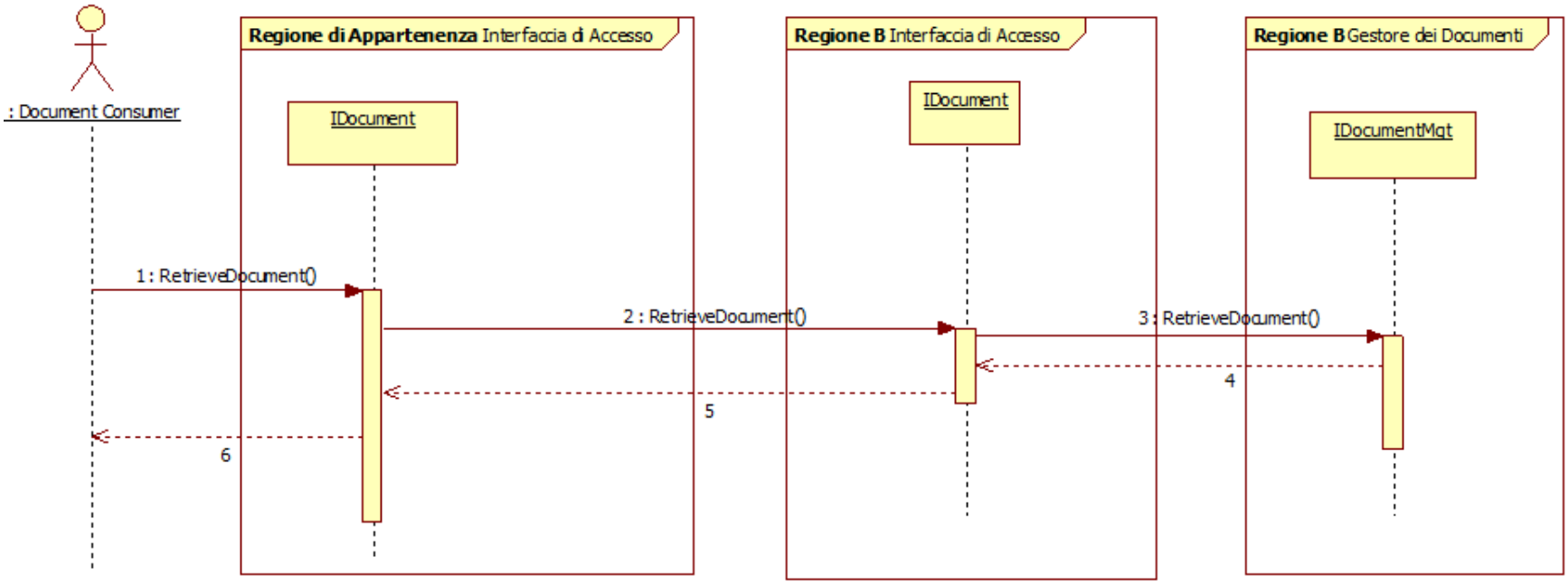


Figura 16. Scenario Recupero documento in ambito extra-regionale

8.1.5.4 Registrazione topic

Un produttore può registrare uno o più topic al fine di notificare eventi appartenenti ad essi. La registrazione riguarda topic preesistenti ed avviene attraverso l'interfaccia di sistema *IEvent*.

1. Il produttore di eventi richiede l'elenco di tutti i topic attraverso l'interfaccia *IEvent*. Ogni nodo regionale mantiene l'elenco completo di tutti i topic e di tutte le gerarchie disponibili;
2. L'interfaccia richiede l'elenco dei topic al nodo broker regionale;
3. Il nodo broker regionale restituisce l'elenco dei topic;
4. Il servizio restituisce l'elenco dei topic al produttore;
5. Il produttore richiede la registrazione di una serie di topic disponibili;
6. L'interfaccia *IEvent* inoltra la richiesta al servizio di registrazione dell'interfaccia *IPublisherRegistrationMgt*;
7. Il servizio crea una nuova registrazione e ne restituisce l'identificativo;
8. L'interfaccia *IEvent* inoltra l'identificativo della registrazione al produttore.

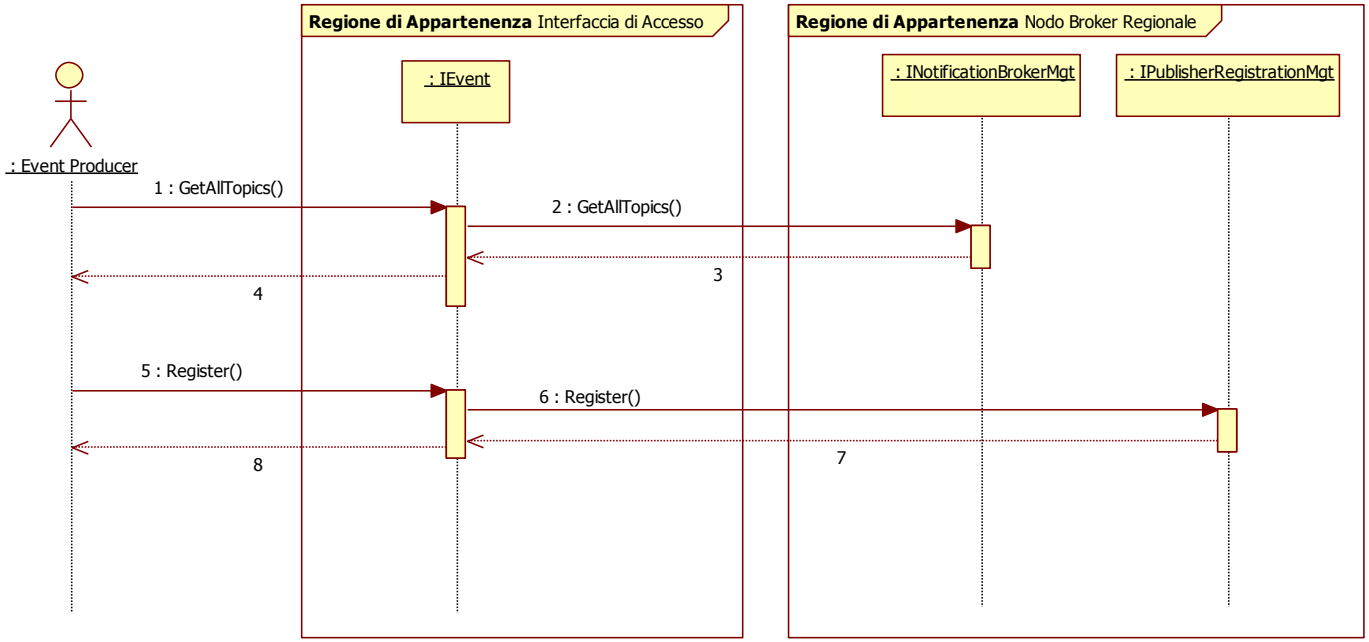


Figura 17. Scenario Registrazione topic

8.1.5.5 Sottoscrizione topic

Un consumatore può sottoscriversi ad uno o più topic.

1. Il consumatore di eventi richiede l'elenco di tutti i topic attraverso l'interfaccia *IEvent*. Nella richiesta può specificare o meno la gerarchia di interesse; in quest'ultimo caso, otterrà l'elenco di tutti i topic;
2. L'interfaccia *IEvent* inoltra la richiesta al nodo broker regionale che mantiene tutti i topic e le gerarchie attive;
3. Il nodo broker regionale restituisce l'elenco dei topic all'interfaccia *IEvent*;
4. L'interfaccia *IEvent* inoltra l'elenco dei topic al consumatore;
5. Il consumatore richiede la sottoscrizione di un elenco di topic, ed indica il tipo di sottoscrittore: push o pull. Nel primo caso il consumatore espone un'interfaccia *IConsumer* che viene invocata dal gestore degli eventi nel caso sia pubblicato un evento di interesse. Nel secondo caso, è il consumatore a richiedere uno o più eventi pubblicati di suo interesse per mezzo di apposite funzioni offerte dall'interfaccia *IEvent*. Eventulamnete, è possibile indicare anche la data di scadenza della sottoscrizione;
6. La sottoscrizione, a differenza di una registrazione, deve essere propagata all'interno della federazione; quindi, l'interfaccia *IEvent* richiede il servizio di routing dell'interfaccia *IBrokerFederationMgt* del nodo broker regionale;
7. Il servizio di routing analizza il messaggio e richiede la sottoscrizione all'interfaccia *ISubscriptionMgt* locale;
8. L'interfaccia *ISubscriptionMgt* locale restituisce l'identificativo univoco nazionale della sottoscrizione;
9. La funzionalità *RouteEvent* inoltra il messaggio di sottoscrizione (inclusivo dell'identificativo generato) ad un altro nodo della federazione;
10. Il nuovo nodo effettua una memorizzazione locale della nuova sottoscrizione;
11. L'identificativo della sottoscrizione viene restituito all'interfaccia *IEvent* della regione di appartenenza;
12. L'intrerfaccia *IEvent* restituisce al consumatore l'identificativo della nuova sottoscrizione.

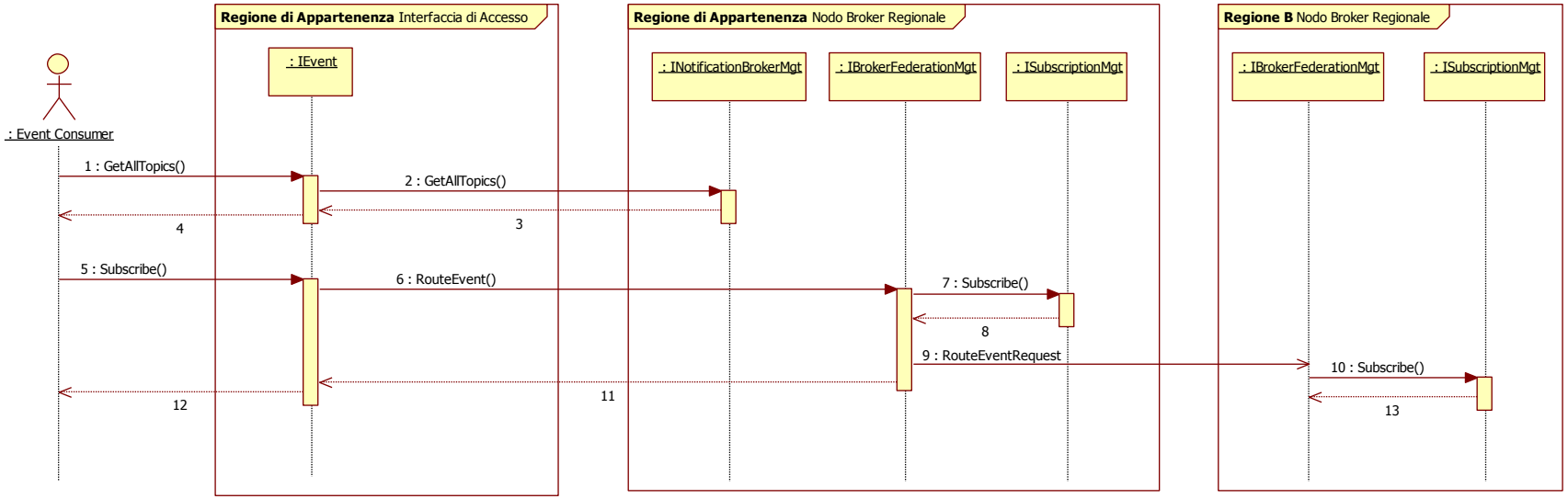


Figura 18. Scenario Sottoscrizione topic

8.1.5.6 Creazione topic

Un produttore, o altro attore autorizzato, può creare nuove classi di eventi (topic).

1. Il produttore richiede la creazione di un nuovo topic all'interfaccia *IEvent*;
2. L'interfaccia *IEvent* richiama il servizio di routing degli eventi dell'interfaccia *IBrokerFederationMgt*;
3. Il servizio di routing dell'interfaccia *IBrokerFederationMgt* della regione corrente analizza il messaggio ricevuto, quindi, richiede la creazione di un nuovo topic all'interfaccia *INotificationBrokerMgt* del nodo locale.
4. Il servizio di routing dell'interfaccia *IBrokerFederationMgt* riceve conferma dell'avvenuta creazione;
5. Il servizio propaga la richiesta di creazione di un nuovo topic ai nodi broker della federazione connessi logicamente;
6. L'interfaccia *IEvent* resituisce l'esito della creazione del topic;
7. Ogni nodo broker regionale che riceve la richiesta attraverso il proprio servizio di routing analizza il messaggio ricevuto, quindi richiede la creazione del topic all'interfaccia *INotificationBrokerMgt* locale ed eventualmente propaga la richiesta ad altri nodi regionali;
8. Il produttore riceve l'esito della richista;
9. Il servizio di routing dell'interfaccia *IBrokerFederationMgt* riceve conferma dell'avvenuta creazione.

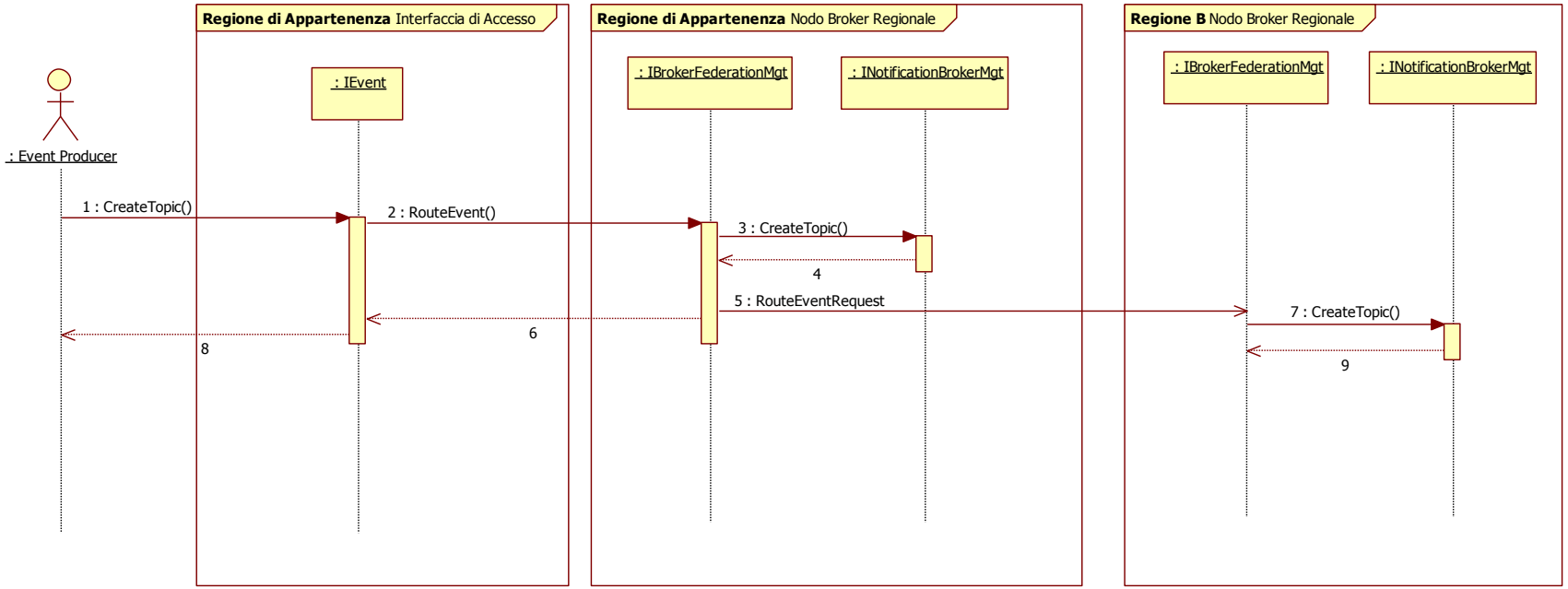


Figura 19. Scenario Creazione topic

8.1.5.7 Creazione gerarchia

Un produttore, o altro attore autorizzato, può creare nuove gerarchie di eventi.

La creazione di una nuova gerarchia avviene individuando un topic di riferimento (root), quindi una serie di attributi della gerarchia e di loro eventuali valori.

Successivamente, il produttore aggiunge ulteriori topic alla gerarchia.

1. Il produttore di eventi richiede l'elenco di tutti i topic attraverso l'interfaccia *IEvent*. Ogni nodo regionale mantiene l'elenco completo di tutti i topic e di tutte le gerarchie disponibili;
2. L'interfaccia richiede l'elenco dei topic al nodo broker regionale;
3. Il nodo broker regionale restituisce l'elenco dei topic;
4. Il servizio restituisce l'elenco dei topic al produttore;
5. Il produttore richiede la creazione di una nuova gerarchia all'interfaccia *IEvent* indicando la *RootTopic* ed, eventualmente, una lista di attributi e loro valori;
6. L'interfaccia *IEvent* inoltra la richiesta alla federazione di nodi broker attraverso l'interfaccia *IBrokerFederationMgt*;
7. Il servizio di routing degli eventi analizza il messaggio ricevuto e richiede la creazione di una nuova gerarchia all'interfaccia *INotificationBrokerMgt*;
8. L'interfaccia restituisce l'identificativo univoco nazionale della gerarchia creata;
9. Il servizio di routing degli eventi inoltra il messaggio di creazione di una nuova gerarchia, arricchito dell'identificativo prodotto, al successivo nodo della gerarchia;
10. Ogni altro nodo della gerarchia crea un'istanza locale in seguito al messaggio ricevuto;
11. Il servizio di routing degli eventi del nodo broker della regione di appartenenza completa l'operazione di creazione restituendo l'identificativo della gerarchia all'interfaccia *IEvent*;
12. L'identificativo viene quindi inoltrato dall'interfaccia *IEvent* al produttore;
13. Successivamente, il produttore può completare la gerarchia aggiungendo ulteriori topic legati tra loro per mezzo di associazioni. Le associazioni possono essere di tre tipi: AND, AND temporizzate e OR;
14. La richiesta di associazione di nuovi topic viene inoltrata dall'interfaccia *IEvent* alla gerarchia attraverso il servizio di routing degli eventi;
15. Il servizio di routing richiede la creazione dell'associazione per l'istanza locale della gerarchia;

16. Il servizio di routing inoltra il messaggio;
17. Viene creata l’associazione per tutte le istanze locali agli altri nodi broker.

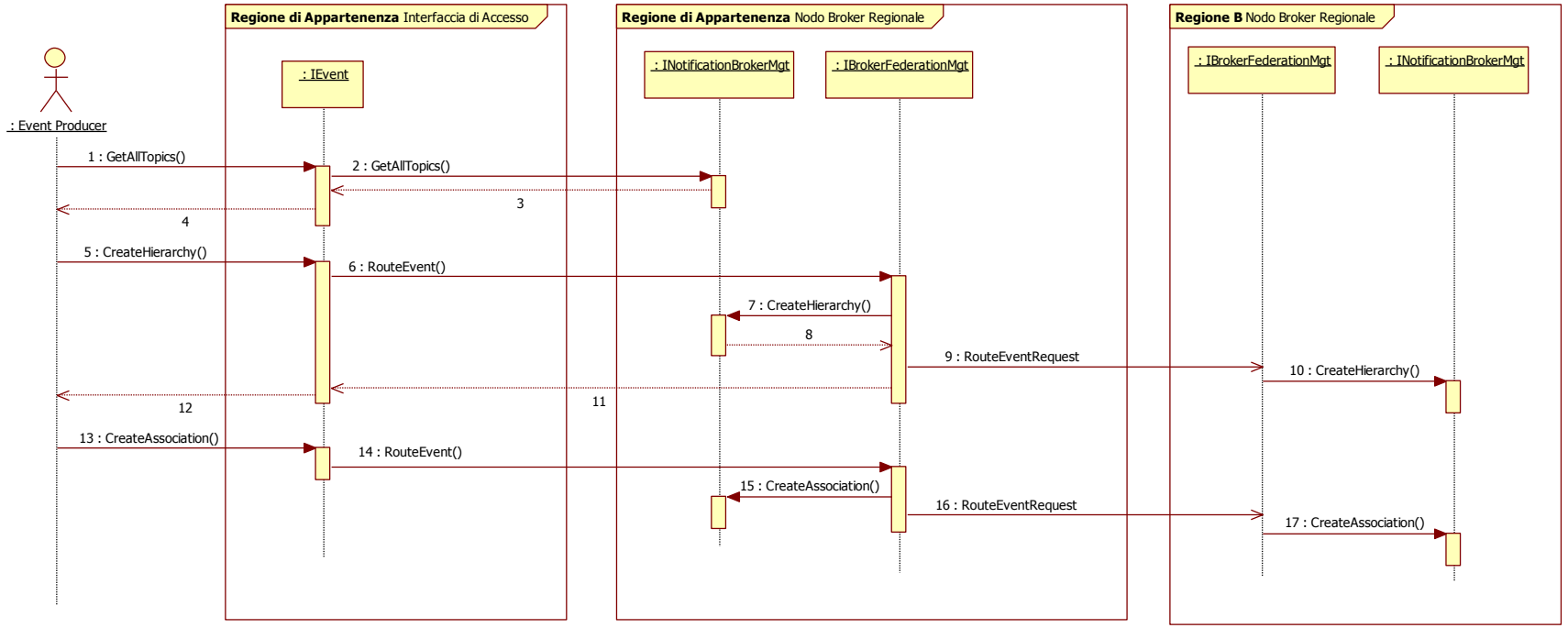


Figura 20. Scenario Creazione gerarchia

8.1.5.8 Archivazione gerarchia

Il proprietario della gerarchia può richiederne l'archiviazione. L'operazione di archiviazione rende la gerarchia indisponibile per la creazione e notifica di nuovi eventi, ma consente il recupero degli eventi associati.

1. Il produttore richiede l'archiviazione della gerarchia all'interfaccia *IEvent*;
2. Il servizio inoltra la richiesta alla federazione di broker attraverso l'operazione *RouteEvent*;
3. L'operazione *RouteEvent* analizza la richiesta ed invoca l'operazione di archiviazione dell'interfaccia *INotificationBrokerMgt*;
4. L'operazione di routing degli eventi inoltra la richiesta alla federazione (nell'esempio al nodo broker della regione B);
5. L'operazione di routing degli eventi restituisce il controllo all'interfaccia *IEvent*;
6. L'interfaccia *IEvent* restituisce il controllo al *Producer*;
7. I nodi broker delle altre Regioni, in maniera asincrona, aggiornano lo stato delle gerarchie locali.

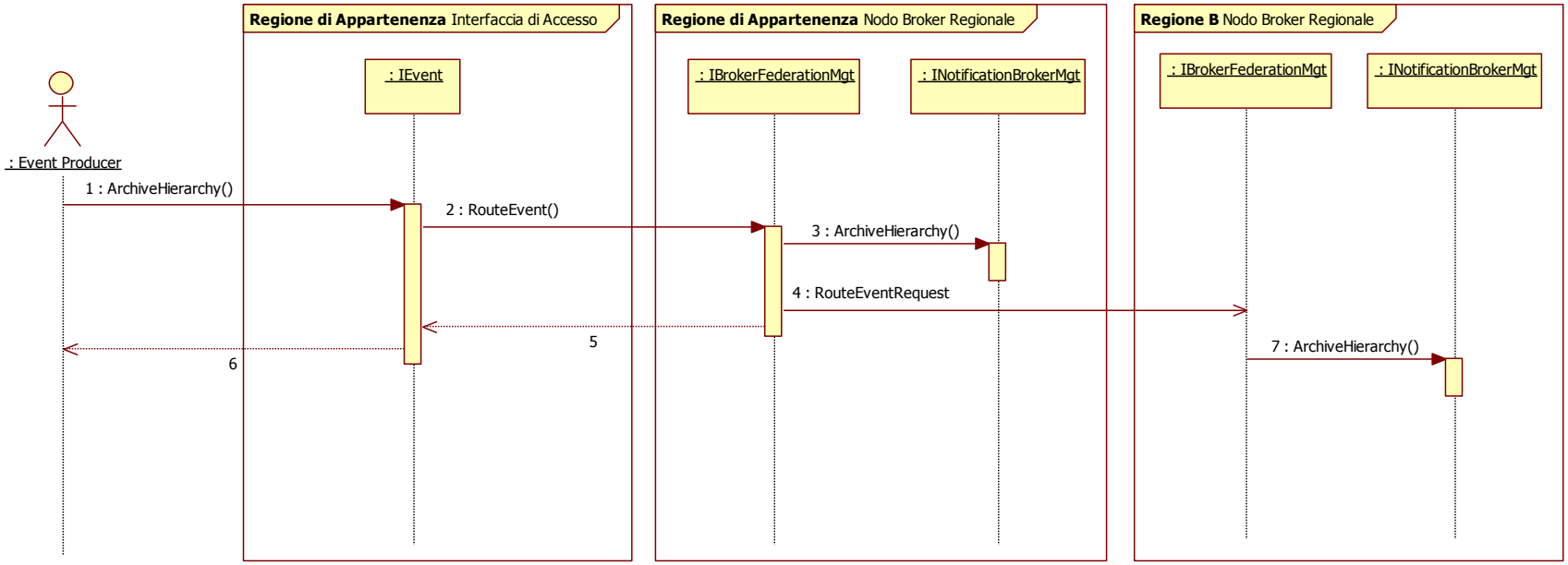


Figura 21. Scenario Archivazione gerarchia

8.1.5.9 Notifica di un evento

L'operazione di notifica di un evento da parte di un produttore riguarda un evento di un determinato topic, oppure di un topic di una gerarchia.

1. Il produttore richiede il servizio di notifica all'interfaccia *IEvent*;
2. Il servizio inoltra la richiesta alla federazione di broker;
3. Il servizio richiede la notifica ai consumatori locali attraverso l'interfaccia *INotificationBrokerMgt* (che richiederà l'elenco dei sottoscrittori di tipo CONSUMER all'interfaccia *ISubscriptionMgt*);
4. Il servizio di routing richiede all'interfaccia *ISubscriptionMgt* l'elenco dei nodi della federazione sottoscrittori (sottoscrittori di tipo BROKER);
5. L'elenco dei nodi viene restituito al richiedente;
6. Il servizio propaga alla federazione l'evento;
   6.1. Opzionalmente, se sottoscrittori di tipo PUSH sono stati registrati per il topic su cui l'evento è stato pubblicato, allora l'interfaccia *INotificationBrokerMgt* invoca un'istanza dell'interfaccia *IConsumer* disponibile presso il consumatore per notificare l'evento.

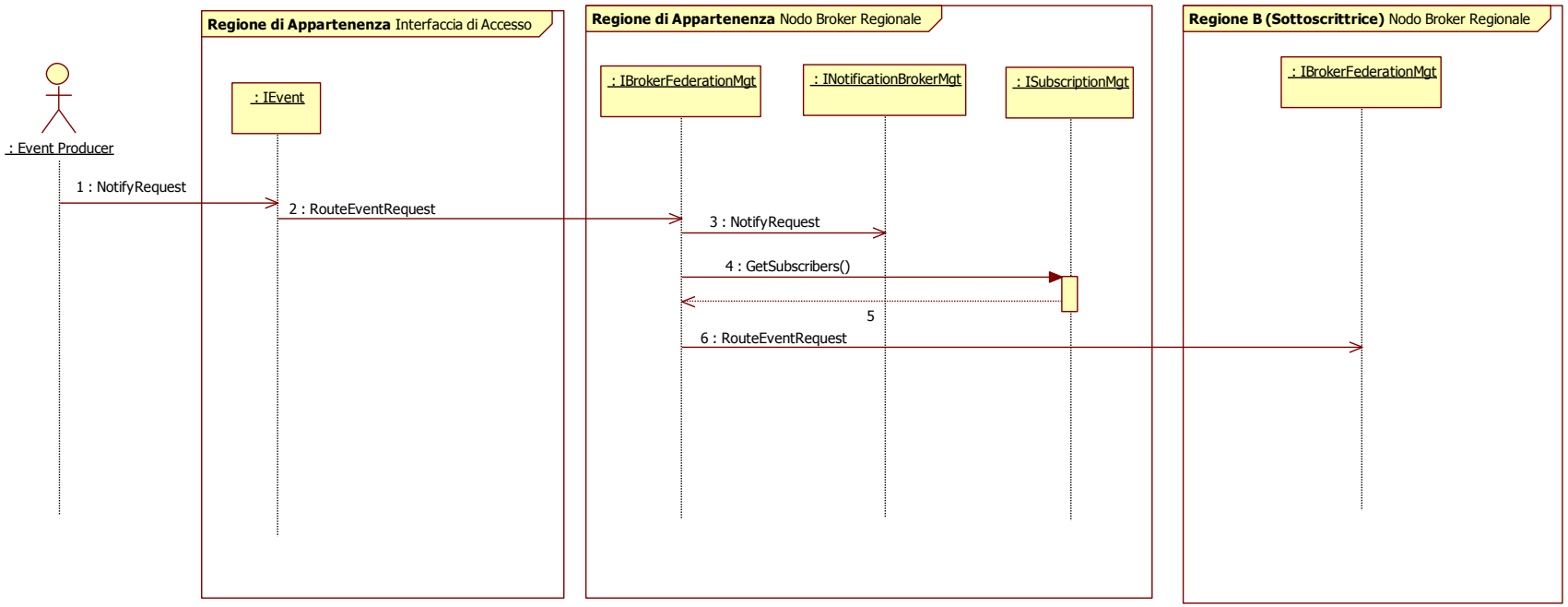


Figura 22. Scenario Notifica di un evento

8.2 Registro Indice Federato

Il *Registro Indice Federato* è una componente atta a memorizzare una serie di informazioni (metadati) inerenti ai documenti sanitari archiviati nei repository, al fine di facilitarne la ricerca e la localizzazione. Tali informazioni possono riguardare la tipologia del documento, l'identificativo del paziente, l'autore del documento, l'organizzazione incaricata della custodia del documento, l'identificativo del documento, etc. In particolare, l'identificativo del documento deve permettere di identificare la Regione contenente il documento, il *Gestore dei Documenti* dispiegato all'interno della Regione e associato ad uno specifico repository ed il documento stesso.

Questa componente può fungere anche da indice dei servizi, ossia i metadati possono rappresentare anche indirizzi (ad es. URI), mediante i quali è possibile localizzare i servizi esposti dai nodi locali. La struttura dei metadati è definita da uno specifico modello informativo opportunamente verticalizzato per il dominio sanitario italiano [7].

Ogni volta che un documento sanitario è generato, oppure ne viene modificato lo stato, il *Registro Indice Federato* deve essere opportunamente aggiornato. In tal modo, questa componente può anche gestire lo stato dei documenti (ad es. una ricetta può passare dallo stato 'prescritto' allo stato 'erogato').

Il *Registro Indice Federato* è una componente composta da un insieme di interfacce che interagiscono con i registri secondo un modello federato. Gli utenti possono accedere al sistema attraverso qualsiasi registro e, in generale, ognuno di questi mantiene solo le informazioni inerenti al dominio di pertinenza. Pertanto, è possibile ottenere i metadati inerenti ai documenti disponibili presso i domini regionali interrogando i corrispondenti registri regionali mediante apposite interfacce del Registro Indice Federato, eseguendo in tal modo una ricerca federata.

Inoltre, i registri membro della federazione possono essere allineati tra di loro mediante un meccanismo di notifica degli eventi basato sul paradigma publish/subscribe, anche allo scopo di gestire la ridondanza dei metadati.

Va sottolineato che l'interoperabilità tra le interface ed i registri legacy può essere raggiunta realizzando uno specifico wrapper che sia in grado di effettuare il mapping tra i protocolli di comunicazione ed i modelli informativi.

8.2.1 Modello concettuale

Il modello concettuale riportato in Figura 23 descrive le principali entità coinvolte.

Registry è l'elemento centrale del modello e gestisce le *EntrySet*, le quali rappresentano l'insieme di metadati associati ad un singolo documento. Ogni *EntrySet* può essere composta da una o più *EntryClass*, le quali possono essere caratterizzate da *EntryAttribute*.

Una federazione di *Registry* è rappresentata dall'entità *Federation*.

L'entità *EventSubscription* rappresenta una sottoscrizione ad uno o più eventi di interesse, mentre l'entità *Query* rappresenta un insieme di criteri per la ricerca di specifici documenti.

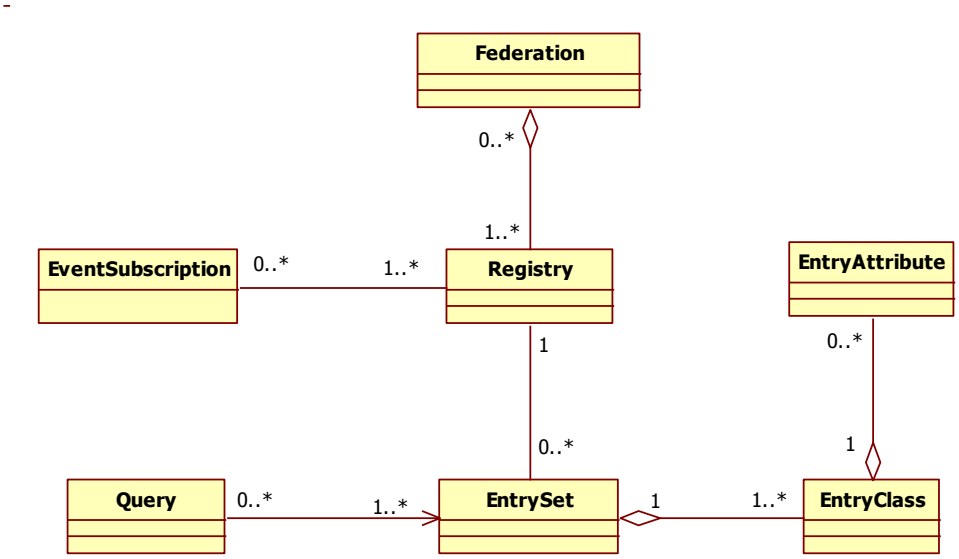


Figura 23. Modello concettuale del Registro Indice Federato

8.2.2 Attori e ruoli

È possibile identificare i seguenti attori negli scenari di interazione con il componente:

- *Entry Producer* – È l'entità capace di produrre nuove entry per i registri;
- *Entry Consumer* – È l'entità capace di consumare entry dei registri;
- *Registry Node* – È un nodo capace di entrare in federazione con altri registri.

8.2.3 Casi d'uso

Si definiscono i seguenti casi d'uso:

- *RegisterEntry* – È la funzionalità che consente di inviare metadati inerenti ad uno o più documenti ad un registro;
- *UpdateEntry* – È la funzionalità che permette l'aggiornamento dei metadati inerenti ad uno o più documenti;
- *ApproveEntry* – È la funzionalità che permette di approvare i metadati presenti in un registro sulla base di specifici criteri;
- *DeprecateEntry* – È la funzionalità che permette di specificare che particolari metadati sono obsoleti;
- *UndeprecateEntry* – È la funzionalità che permette di specificare che particolari metadati non sono più obsoleti;
- *RemoveEntry* – È la funzionalità che permette di rimuovere specifici metadati da un registro (deve essere utilizzata solo in particolari condizioni, ad es. per eliminare metadati errati);
- *RelocateEntry* – È la funzionalità che permette di rilocare specifici metadati in un altro registro;
- *SubscribeEvent* – È la funzionalità che permette di sottoscrivere l'interesse a ricevere eventi sulla base di specifici criteri, al fine di supportare la federazione di registri mediante notifica di eventi;
- *UnsubscribeEvent* – È la funzionalità che permette la rimozione di una sottoscrizione;
- *Query* – È la funzionalità che permette di sottoporre una query ad uno specifico registro o, alternativamente, di invocare una stored query disponibile presso

un registro;

- *CreateFederation* – È la funzionalità che permette di creare una nuova federazione;
- *JoinFederation* – È la funzionalità che permette di aggiungere un nuovo registro alla federazione;
- *SetRegistryCoordinator* – È la funzionalità che permette di specificare il registro coordinatore all'interno di una federazione;
- *LeaveFederation* – È la funzionalità che permette di rimuovere un registro dalla federazione;
- *DissolveFederation* – È la funzionalità che permette di rimuovere una federazione;
- *QueryFederation* – È la funzionalità che permette di sottoporre query ad una federazione di registri.

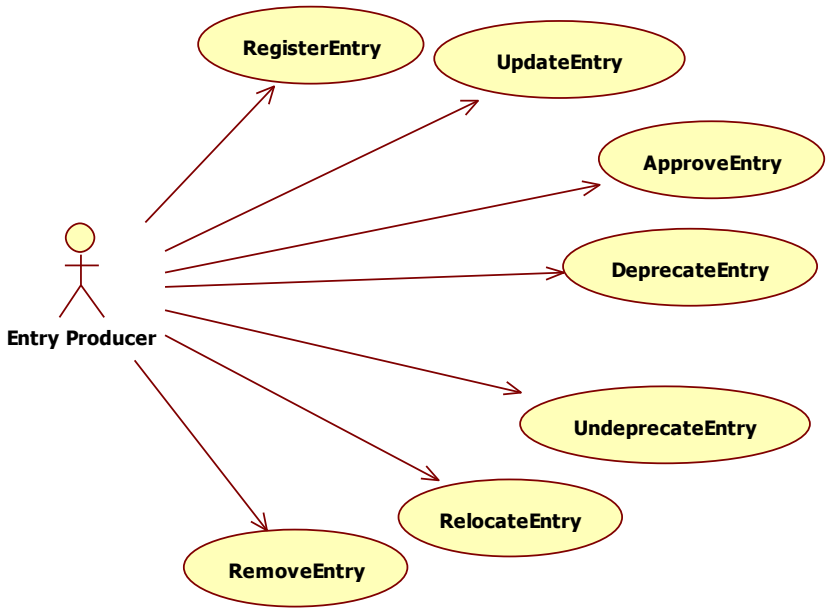


Figura 24. Casi d'uso per il produttore di metadati

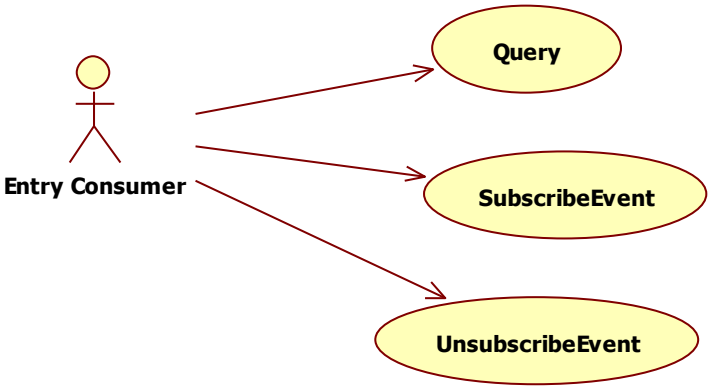


Figura 25. Casi d'uso per il consumatore di metadati

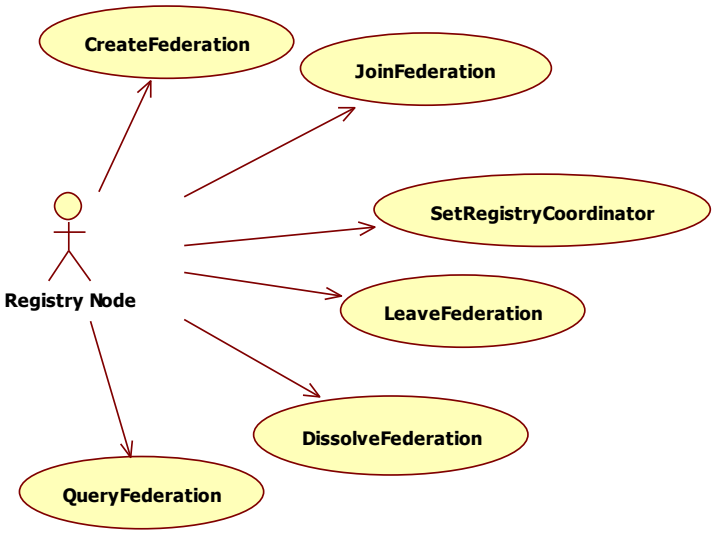


Figura 26. Casi d'uso per un nodo registro

8.2.4 Tecnologie di riferimento

La tecnologia più indicata per lo sviluppo del *Registro Indice Federato* è ebXML ver. 3.0 o superiore [8][9], in quanto essa è in grado di gestire grandi quantità di informazioni complesse in maniera sicura. In aggiunta, le specifiche ebXML Registry Services e ebXML Registry Information Model ver. 3.0 o superiore definiscono una serie di servizi ed un modello informativo di riferimento, mediante i quali è possibile condividere i metadati in una federazione, inviare notifiche di eventi basate sul contenuto, gestire l'identità federata tra i registri della federazione e governare l'accesso ai registri mediante politiche basate sul ruolo.

8.2.5 Architettura della componente

Il *Registro Indice Federato* è una componente distribuita e federata secondo una logica peer-to-peer, che si estende su tutto il territorio nazionale.

Ogni singola Regione (o Provincia Autonoma) può prevedere la presenza di più registri, ubicati presso i nodi locali, quali ad esempio le ASL, e presso il nodo regionale (o provinciale).

Ogni nodo locale deve far riferimento al proprio registro, detto *Registro Locale*. I nodi locali privi di un registro possono far riferimento ai registri di altri nodi locali o del nodo regionale.

Ogni nodo regionale deve prevedere la presenza di almeno un registro, il *Registro Regionale*. Il nodo regionale può prevedere la presenza di più *Registri Regionali*, ma è importante che solo uno di questi funga da interfaccia per le altre Regioni, il *Registro Regionale di Riferimento*. Evidentemente, nel caso in cui una Regione preveda la presenza di un unico *Registro Regionale*, questo coincide con il *Registro Regionale di Riferimento*. Si noti che, in assenza di ambiguità, in questo documento il *Registro Regionale di Riferimento* è indicato semplicemente come *Registro Regionale*.

L'architettura del *Registro Indice Federato* può essere suddivisa in più livelli:

- *livello 2*: a questo livello fanno parte le interfacce della componente che interagiscono con i *Registri Locali*, i quali possono essere federati tra di loro mediante un modello peer-to-peer, anche allo scopo di gestire la ridondanza dei dati; il livello 2 è opzionale;
- *livello 1*: a questo livello appartengono le interfacce della componente che

interagiscono con il *Registro Regionale*, il quale è connesso con i *Registri Locali*, se presenti, secondo un modello super-peer; è comunque possibile che vi siano più *Registri Regionali*: in questo caso, tali registri possono essere federati tra di loro, al fine di gestire la ridondanza dei dati ed il load balancing, anche se è importante che ce ne sia uno di riferimento; il livello 1 è necessario;

- *livello 0*: questo livello rappresenta la federazione dei *Registri Regionali di Riferimento* su base nazionale, i quali sono interconnessi secondo un modello peer-to-peer; il livello 0 è necessario.

Nella prossima figura sono mostrati i livelli di federazione del *Registro Indice Federato*.

La federazione di livello 0 può essere basata su SPC, ossia i *Registri Regionali di Riferimento* possono interoperare tra di loro attraverso le Porte di Dominio.

Le informazioni contenute nei *Registri Regionali* possono essere di due tipologie:

1. possono rappresentare l'intero dataset di metadati, eventualmente ridondato rispetto a quello contenuto nei *Registri Locali* (se presenti);
2. possono rappresentare un dataset di metadati minimale, tra cui i riferimenti ai *Registri Locali*, i quali memorizzano le informazioni complete.

Al fine di minimizzare l'interazione interregionale tra le interfacce che interagiscono con i registri, ogni *Registro Regionale di Riferimento* deve contenere le informazioni riguardanti i propri assistiti. Di conseguenza, tutti gli eventi clinici occorsi in una Regione differente da quella in cui un assistito risiede possono essere notificati dal *Registro Regionale* della Regione in cui è occorso l'evento clinico al *Registro Regionale* della Regione di residenza. Quest'ultimo può contenere una delle seguenti tipologie di informazione:

1. l'intero dataset di metadati;
2. un dataset di metadati minimale, tra cui i riferimenti al *Registro Regionale di Riferimento* extra-regionale.

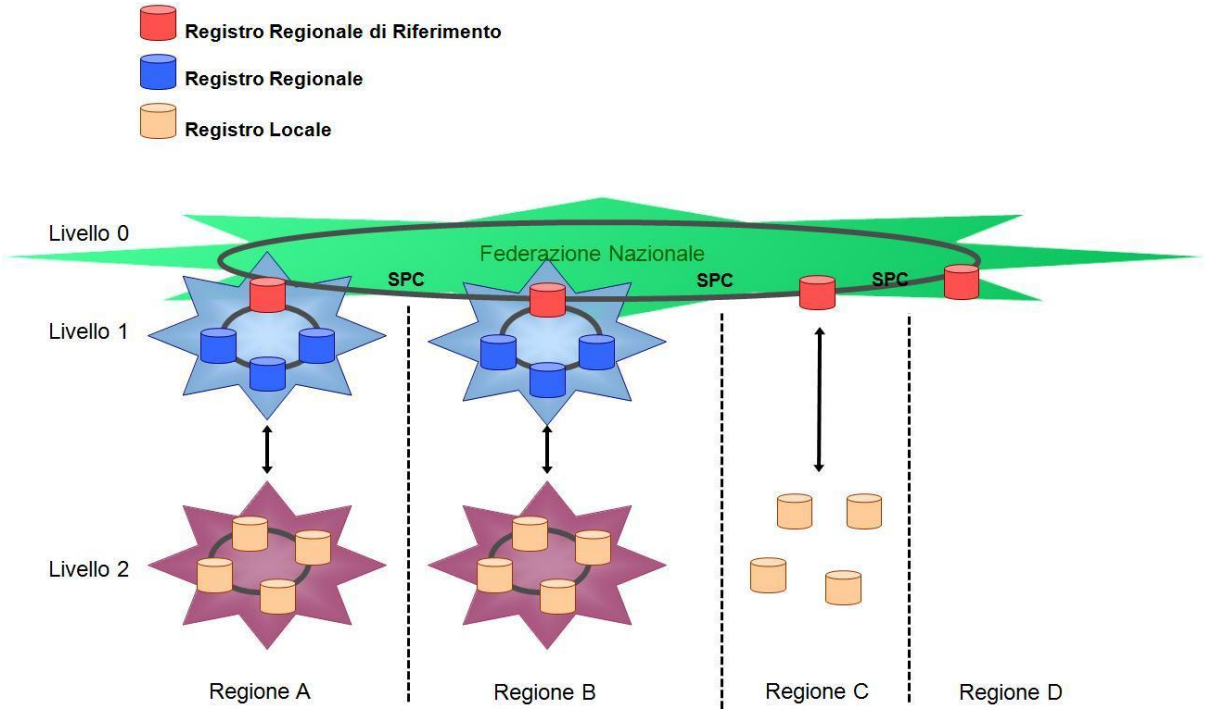


Figura 27. Livelli di federazione del Registro Indice Federato

Secondo tale modello, la ricerca delle informazioni sanitarie di un cittadino per gli usi primari è estremamente facilitata, in quanto, nella maggior parte dei casi, sono reperibili presso la Regione di residenza del cittadino stesso.

Invece, le ricerche federate si rendono necessarie per gli usi secondari.

Presso ogni registro possono essere dispiegate le interfacce *IMetadataMgt, IQueryMgt*, *IEventMgt* e *IRegistryFederationMgt* riportate nella prossima figura.

L'interfaccia *IMetadataMgt* supporta le operazioni per gestire il ciclo di vita dei metadati (memorizzazione, aggiornamento, etc.).

L'interfaccia *IQueryMgt* consente di sottoporre query ad uno specifico registro o ad una federazione di regsitri.

L'interfaccia *IEventMgt* permette la notifica di eventi tra i registri della federazione.

L'interfaccia *IRegistryFederationMgt* offre funzionalità per gestire la federazione localmente presso ogni registro.

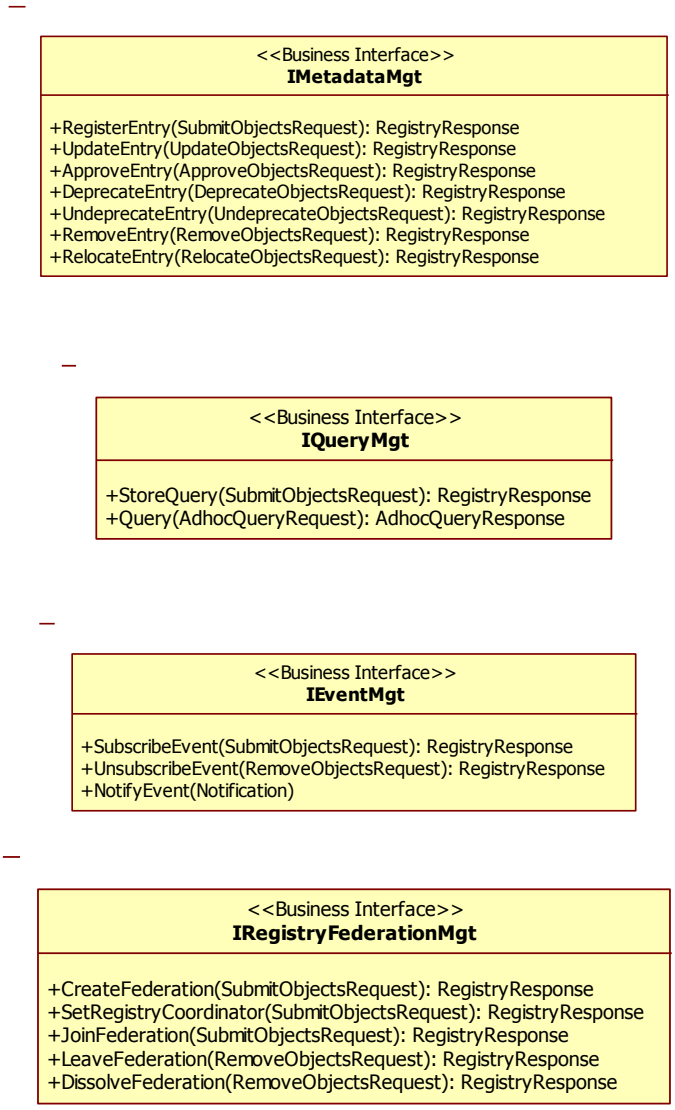


Figura 28. Interfacce IMetadataMgt, IQueryMgt, IEventMgt e IRegistryFederationMgt

8.2.6 Modello dati

Nella prossima figura è mostrata un'architettura che prevede la realizzazione di tre componenti fisici: il gestore del ciclo di vita dei metadati e delle notifiche, il gestore delle query ed il gestore della federazione.

Di seguito è illustrato il modello dei dati, il quale è conforme al modello informativo definito nelle specifiche ebXML Registry Services e ebXML Registry Information Model ver. 3.0 o superiore.

- *RegistryPackage* – Rappresenta l'insieme di metadati che caratterizzano un documento ed è organizzato in un insieme di *RegistryObject*. È conforme alla classe RegistryPackage definita in ebXML.
- *RegistryObject* – Rappresenta una generica classe. È conforme alla classe RegistryObject definita in ebXML.
- *Federation* – Rappresenta una federazione di registri. È conforme alla classe Federation definita in ebXML.
- *Registry* – Rappresenta uno specifico registro. È conforme alla classe Registry definita in ebXML.
- *Subscription* – Definisce l'interesse di un utente a ricevere notifiche per specifici tipi di eventi. È conforme alla classe Subscription definita in ebXML.
- *Notification* – Comprende il contenuto dell'evento da notificare.
- *AdhocQuery* – Contiene una query espressa in una specifica sintassi. È conforme alla classe AdhocQuery definita in ebXML.
- *SubmitObjectsRequest* – Rappresenta la richiesta di inviare metadati in un registro. È conforme alla classe SubmitObjectsRequest definita in ebXML.
- *UpdateObjectsRequest* – Rappresenta la richiesta di aggiornare metadati in un registro. È conforme alla classe UpdateObjectsRequest definita in ebXML.
- *ApproveObjectsRequest* – Rappresenta la richiesta di approvare metadati in un registro. È conforme alla classe ApproveObjectsRequest definita in ebXML.

- *DeprecateObjectsRequest* – Rappresenta la richiesta di rendere obsoleti metadati di un registro. È conforme alla classe DeprecateObjectsRequest definita in ebXML.
- *UndeprecateObjectsRequest* – Rappresenta la richiesta di rivalidare specifici metadati di un un registro. È conforme alla classe UndeprecateObjectsRequest definita in ebXML.
- *RemoveObjectsRequest* – Rappresenta la richiesta di rimuovere metadati da un registro. È conforme alla classe RemoveObjectsRequest definita in ebXML.
- *RelocateObjectsRequest* – Rappresenta la richiesta di rilocare metadati da un registro ad un altro. È conforme alla classe RelocateObjectsRequest definita in ebXML.
- *RegistryResponse* – Rappresenta la risposta di un registro ad una richiesta. È conforme alla classe RegistryResponse definita in ebXML.
- *AdhocQueryRequest* – Rappresenta la richiesta di sottoporre una query ad un registro, la quale può essere propagata ad una federazione. È conforme alla classe AdhocQueryRequest definita in ebXML.
- *AdhocQueryResponse* – Rappresenta la risposta di un registro ad una query. È conforme alla classe AdhocQueryResponse definita in ebXML.
- *ResponseOption* – Consente di specificare il formato dei risultati di una query. È conforme alla classe ResponseOption definita in ebXML.

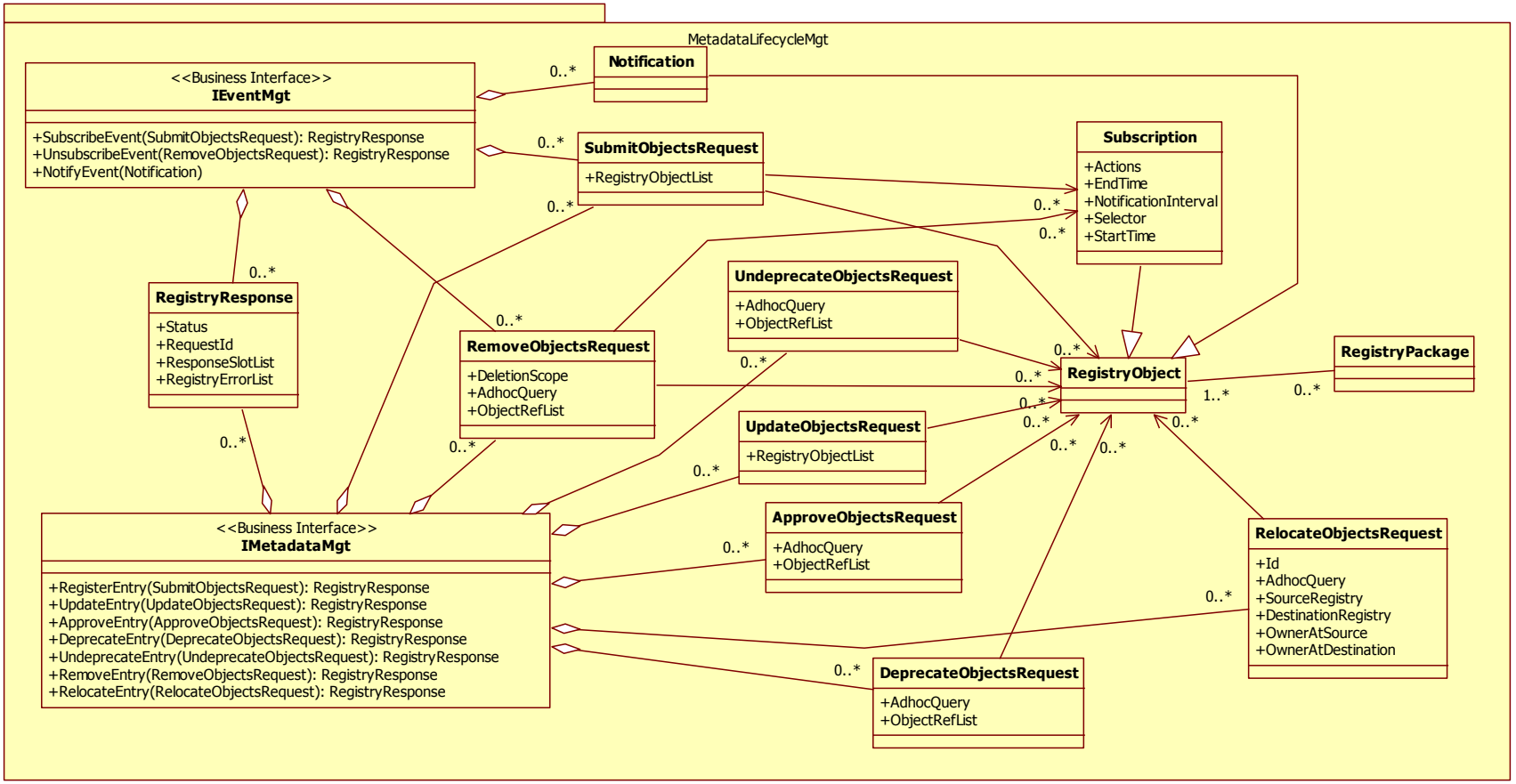


Figura 29. Interfaccia ed architettura della componente MetadataLifecycleMgt

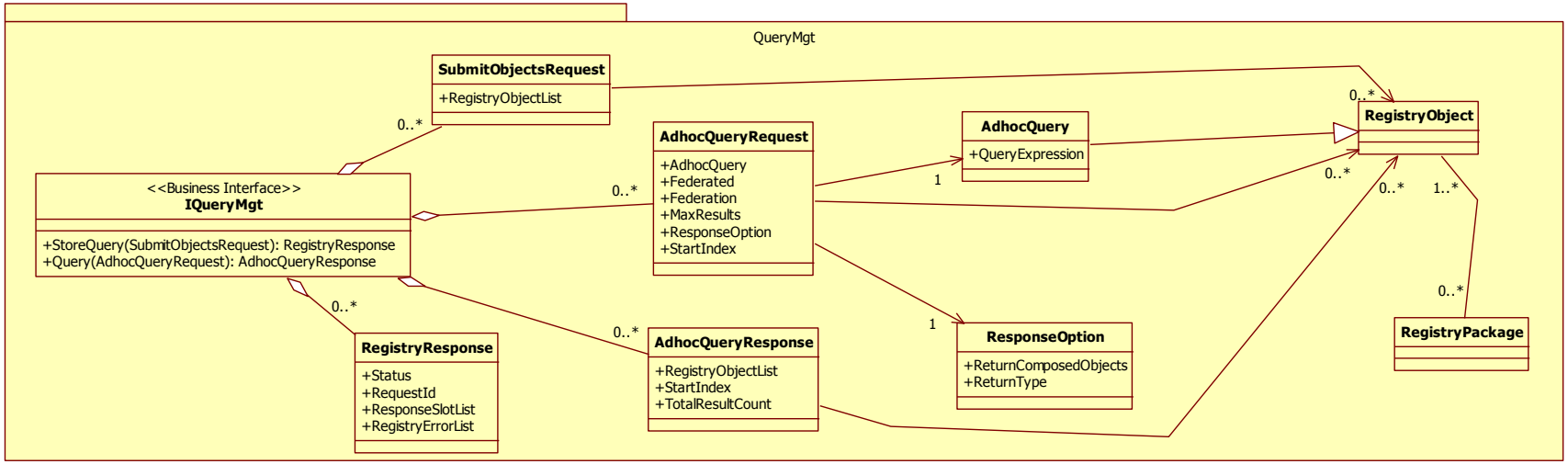


Figura 30. Interfaccia ed architettura della componente QueryMgt

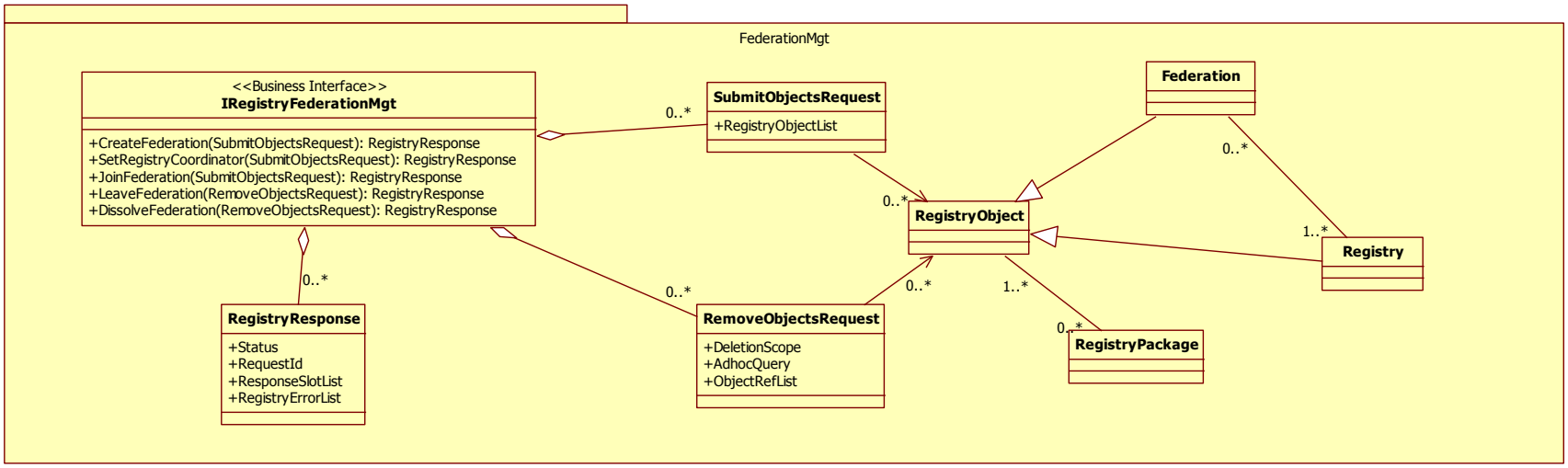


Figura 31. Interfaccia ed architettura della componente FederationMgt

8.2.7 Scenari d'uso

8.2.7.1 Ricerca di documenti in ambito regionale

Lo scenario fa riferimento al caso in cui viene sottoposta una query al fine di ricercare specifici documenti disponibili nel dominio regionale.

1. Il richiedente invia una query all'interfaccia *IEntry*;
2. L'interfaccia *IEntry* inoltra la richiesta all'interfaccia *IQueryMgt* del registro regionale;
3. L'interfaccia *IQueryMgt* restituisce i risultati all'interfaccia *IEntry*;
4. L'interfaccia *IEntry* restituisce i risultati al richiedente.

8.2.7.2 Ricerca federata di documenti in ambito extra-regionale

Lo scenario fa riferimento al caso in cui viene sottoposta una query al fine di ricercare specifici documenti disponibili nella federazione nazionale.

1. Il richiedente invia una query all'interfaccia *IEntry*;
2. L'interfaccia *IEntry* inoltra la richiesta all'interfaccia *IQueryMgt* del registro regionale;
3. L'interfaccia *IQueryMgt* propaga la query a tutte le interfacce *IQueryMgt* della federazione esposte da ogni dominio regionale;
4. L'interfaccia *IQueryMgt* riceve tutti i risultati dalle interfacce *IQueryMgt* invocate;
5. L'interfaccia *IQueryMgt* aggrega i risultati e li restituisce all'interfaccia *IEntry*;
6. L'interfaccia *IEntry* restituisce i risultati al richiedente.

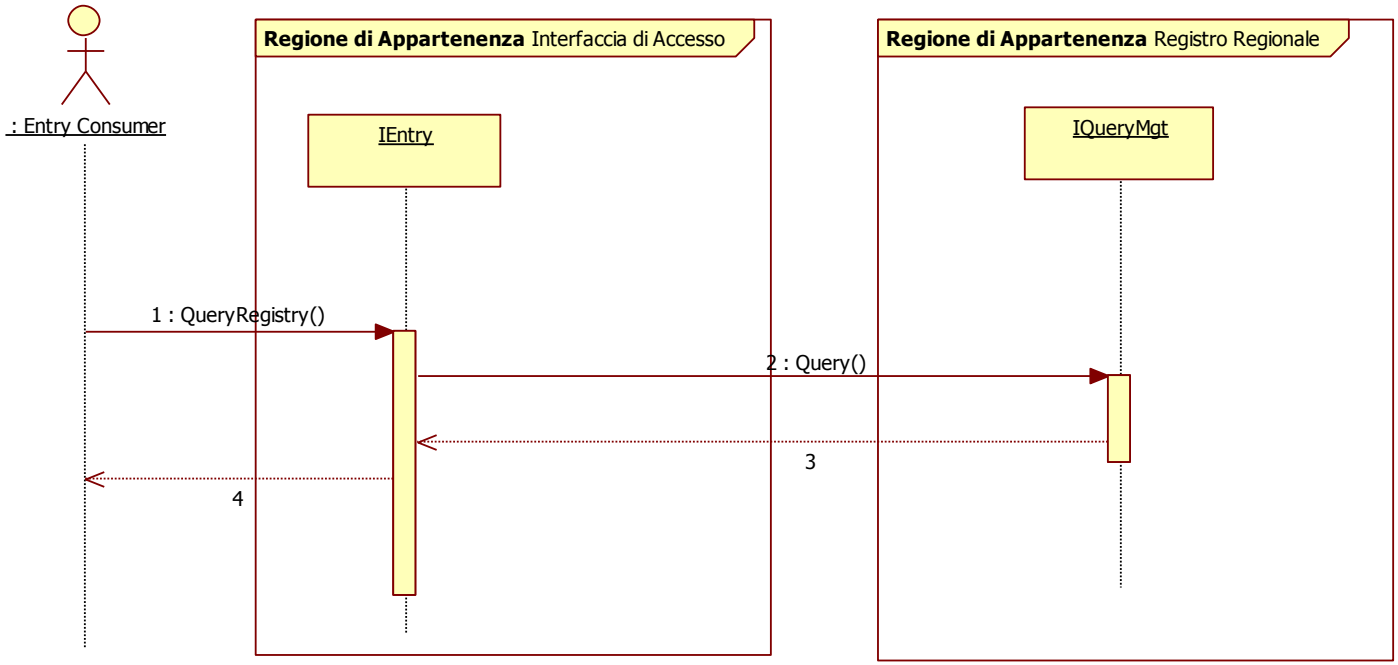


Figura 32. Scenario Ricerca di documenti in ambito regionale

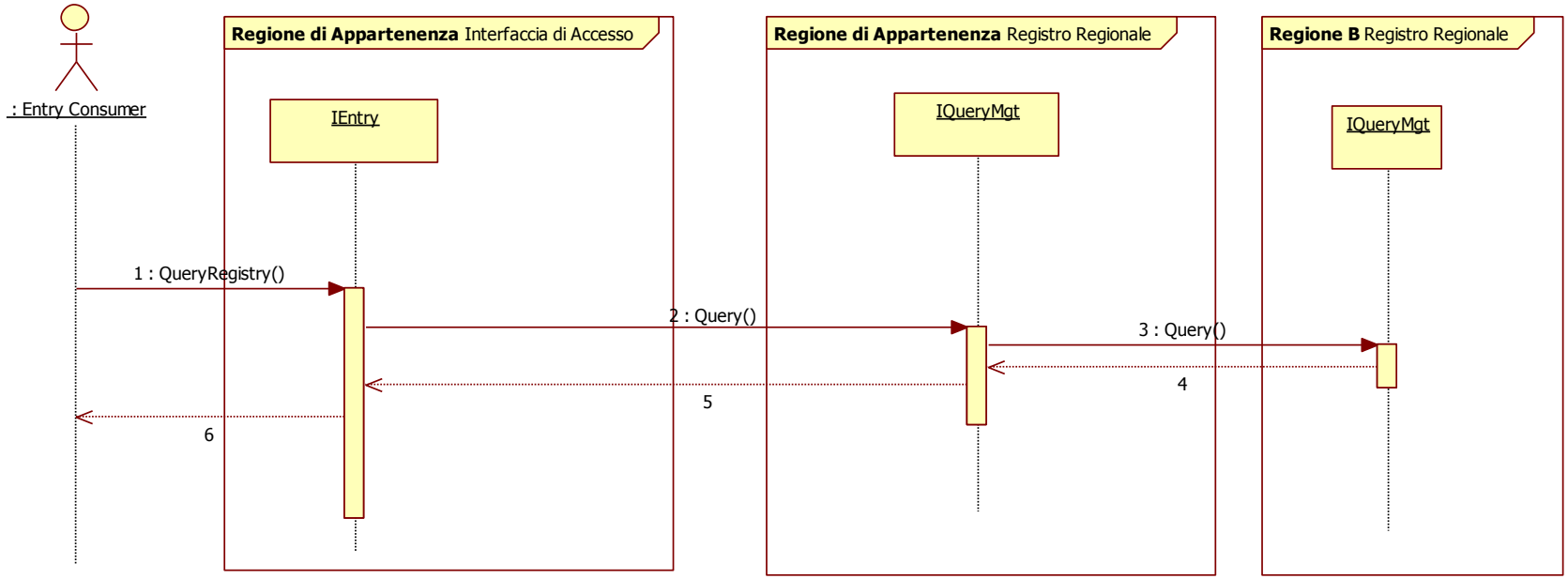


Figura 33. Scenario Ricerca federata di documenti in ambito extra-regionale

8.2.7.3 Notifica di eventi tra registri

Lo scenario descrive le modalità di federazione tra i registri mediante la notifica di eventi. In particolare, si fa riferimento al caso in cui, dopo la produzione o l'aggiornamento di un documento sanitario in una Regione ospite, il registro di quest'ultima notifichi opportuni metadati alla Regione di appartenenza del cittadino. Si presuppone che il registro della Regione di appartenenza abbia già inviato una richiesta di sottoscrizione al registro della Regione ospite.

1. Il *Registro Regionale* della Regione ospite invia la notifica di un nuovo evento, contenente i metadati inerenti ad uno specifico documento, alle interfacce *IEntry* delle Regioni che hanno manifestato l'interesse a ricevere eventi di questo tipo;
2. L'interfaccia *IEntry* inotra l'evento all'interfaccia *IEventMgt*;
3. L'interfaccia *IEventMgt* della Regione sottoscrittrice riceve ed analizza l'evento, quindi richiede l'aggiornamento dei metadati locali all'interfaccia *IMetadataMgt*.
4. L'interfaccia *IEventMgt* riceve l'esito dall'interfaccia *IMetadataMgt*.

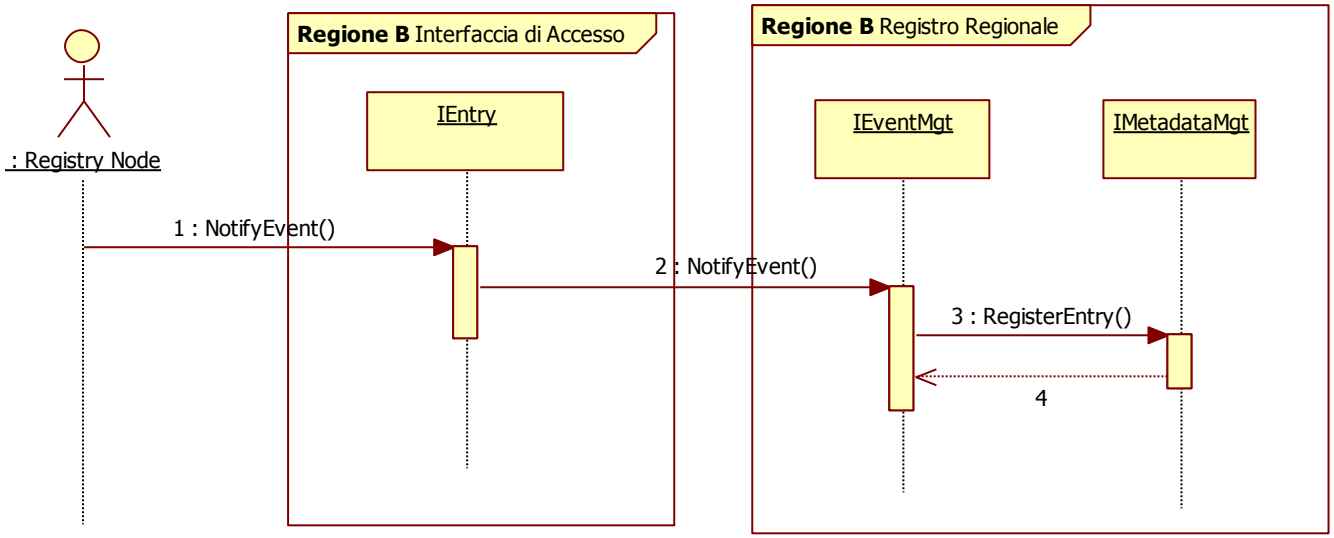


Figura 34. Scenario Notifica di eventi tra registri

8.2.8 Integrazione con il Sistema Pubblico di Connettività

Le funzionalità del *Registro Indice Federato* possono essere esposte su Porta di Dominio SPC. In particolare, l'interfaccia *IQueryMgt* deve essere esposta da ogni nodo regionale per permettere l'elaborazione di richieste di ricerca documenti provenienti da domini extra-regionali, mentre l'interfaccia *IEventMgt* può essere esposta per consentire la gestione di eventi contenenti metadati inerenti a documenti caricati o aggiornati presso domini extra-regionali, i quali devono essere memorizzati nel registro regionale.

8.3 Gestore Gerarchico degli Eventi

Il *Gestore Gerarchico degli Eventi* è una componente opzionale che effettua il routing e la notifica degli eventi sanitari a tutti gli attori interessati (medici di medicina generale, medici specialisti, etc.). Il modello adottato è il publish/subscribe basato su broker.

La gestione degli eventi nell'Infrastruttura del FSE si estende, in generale, su una vasta area geografica: ciò comporta una soluzione di tipo distribuito del sistema di notifica degli eventi. Il broker deve quindi necessariamente essere distribuito e decentralizzato, ossia realizzato attraverso una federazione di broker locali. Gli utenti possono accedere al sistema attraverso qualsiasi broker. Ogni nodo regionale o nodo locale completo può prevedere l'esistenza di un nodo broker.

Allo scopo di rendere più efficiente la gestione e la notifica degli eventi, si adotta un modello gerarchico di classificazione degli eventi stessi. In questo modo, risultano notevolmente semplificate le operazioni di sottoscrizione degli insiemi di eventi di interesse, nonché la notifica stessa.

Infine, va sottolineato che l'interoperabilità con le soluzioni esistenti è garantita da semplici operazioni di collegamento tra le relative componenti di brokering degli eventi.

8.3.1 Modello concettuale

Il modello concettuale riportato in Figura 35 descrive le principali entità coinvolte. I *Consumer* ed i *Producer* sono associati, rispettivamente, ad una o più *Subscription* ed una o più *Registration*.

Ogni nodo *Broker* della federazione (*BrokerFederation*) gestisce una o più gerarchie (*Hierarchy*) di classi (*Topic*) di eventi (*Event*). Ogni classe, così come ogni gerarchia, può essere caratterizzata da attributi (*Attribute*).

Sia le sottoscrizioni che le registrazioni possono riguardare singole classi di eventi oppure intere gerarchie.

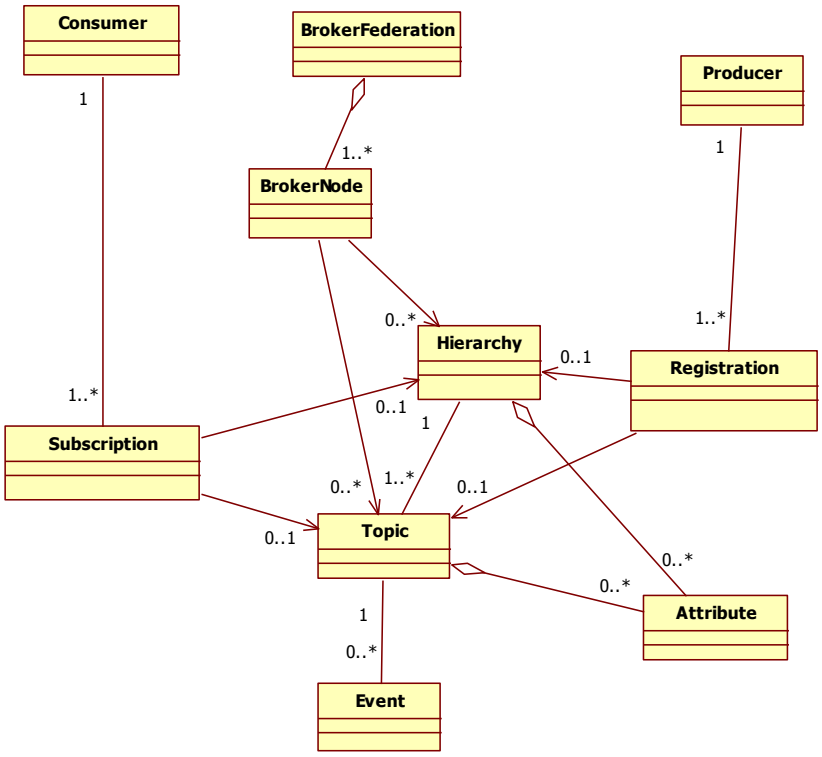


Figura 35. Modello concettuale del Gestore Gerarchico degli Eventi

8.3.2 Modello gerarchico degli eventi

La gerarchia di classi di eventi presentata in Figura 36 si compone di due namespace separati. Il primo, R1, contiene oltre alla root class quattro classi aggiuntive (A, B, C e D) di eventi, ognuna eventualmente caratterizzata da propri attributi.

Nella gerarchia la classe B estende le classi A e C, quindi specifica i propri attributi (*AttributeB1* ed *AttributeB2*). Conseguentemente, il consumatore *Consumer1* che sottoscrive l'attributo *AttributeC1* della classe C, sottoscriverà automaticamente anche le classi B e D della gerarchia. Differentemente, il consumatore *Consumer2* che sottoscrive l'intero topic della classe B o uno degli attributi *AttributeB1* o *AttributeB2*, riceverà esclusivamente gli eventi di questa classe.

È infine opportuno sottolineare che le classi C delle due gerarchie sono distinte ed individuate attraverso namespace differenti.

Le gerarchie di classi di eventi sono definite dinamicamente. In tal senso, il *Gestore Gerarchico degli Eventi* deve fornire meccanismi per istanziare una gerarchia, identificando una root topic, e definire le relazioni tra classi di eventi. Questi

meccanismi consentono di definire, in funzione delle esigenze specifiche, oppurtune relazioni tra eventi solitamente scorrelati tra loro. È, ad esempio, il caso in cui si voglia definire uno specifico percorso di cura che prevede una certa sequenza di esami clinici abitualmente non correlati tra di loro.

La possibilità di definire gerarchie ad hoc consente l'aggregazione logica di documenti sanitari che potranno in seguito essere reperiti dal fascicolo attraverso un'unica operazione. A tale scopo, il *Gestore Gerarchico degli Eventi* può memorizzare in maniera persistente le gerarchie dinamicamente definite.

Il *Gestore Gerarchico degli Eventi* si realizza attraverso un'architettura federata secondo il modello peer-to-peer. Per questo motivo ogni nodo deve supportare opportune interfacce per la sottoscrizione e notifica degli eventi, nonché i metodi per la gestione delle gerarchie, ed infine i servizi per la gestione della federazione di broker.

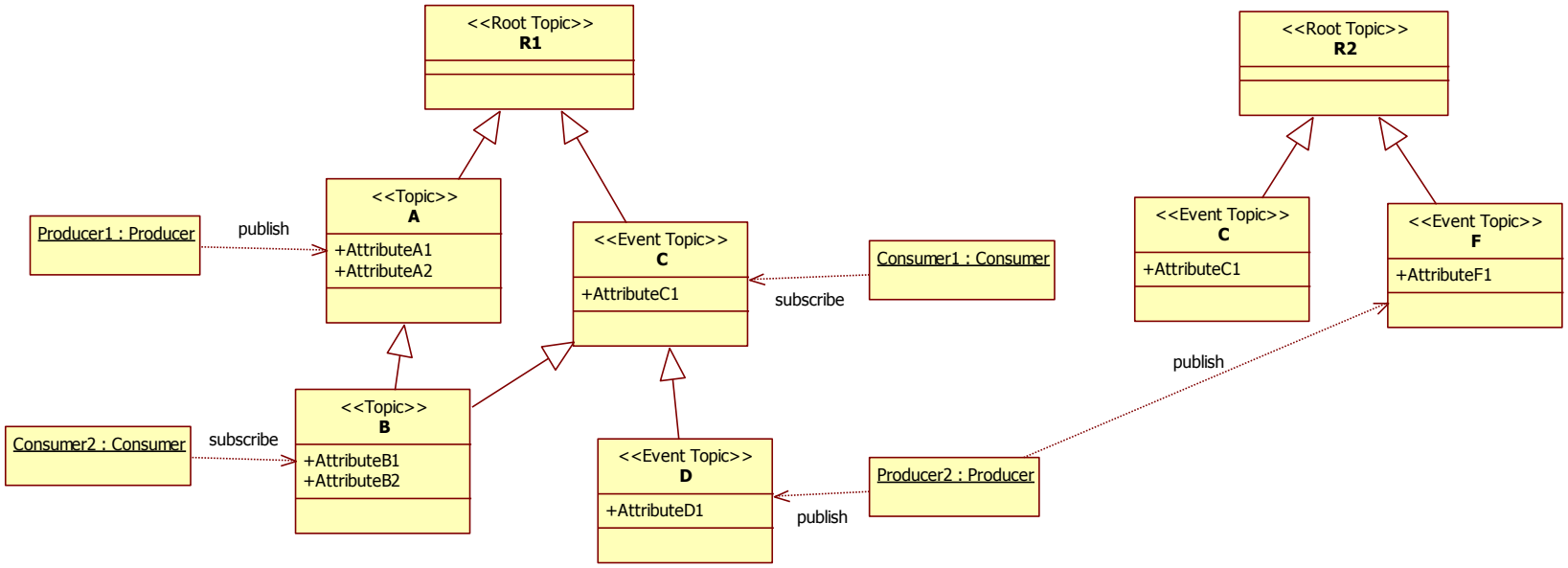


Figura 36. Modello gerarchico degli eventi

8.3.3 Attori e ruoli

È possibile identificare i seguenti attori e ruoli negli scenari di interazione con il componente:

- *Publisher* – È l'entità capace di pubblicare nuovi eventi per conto di un produttore;
- *NotificationProducer* – È l'entità capace di produrre nuovi eventi;
- *Subscriber* – È l'entità capace di sottoscrivere eventi per conto di un consumatore;
- *NotificationConsumer* – È l'entità capace di consumare eventi;
- *FederationManager* – È l'entità capace di modificare la federazione di broker;
- *Event Producer* – È l'entità capace sia di pubblicare che di produrre nuovi eventi;
- *Event Consumer* – È l'entità capace di sottoscrivere e consumare eventi;
- *Broker Node* – È un nodo capace di entrare in federazione.

Le relazioni fra attori e ruoli sono rappresentate in Figura 37.

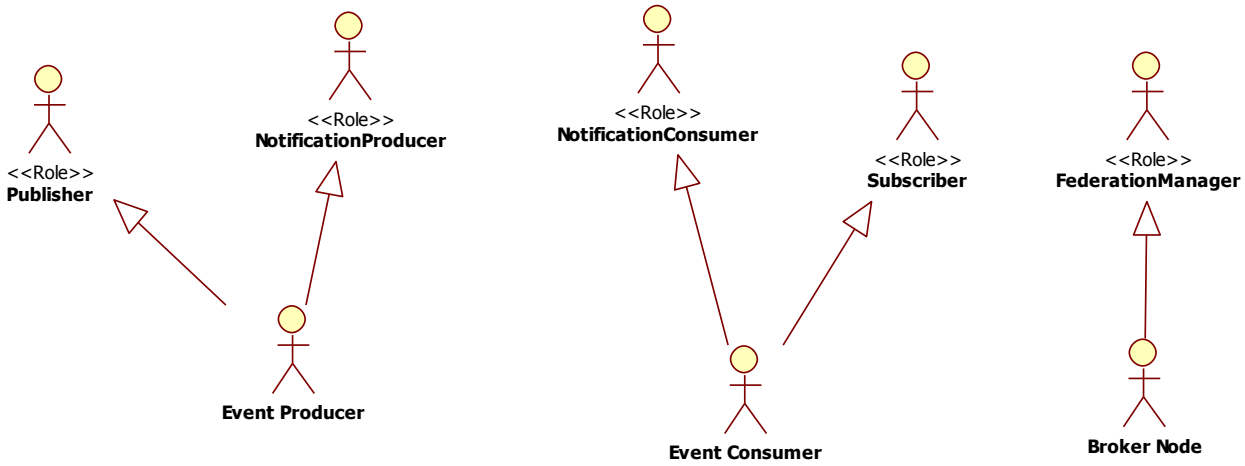


Figura 37. Attori e ruoli

8.3.4 Casi d'uso

Si definiscono i seguenti casi d'uso:

- *Register* – È la funzionalità che permette la pubblicazione di uno o più topic. Attraverso questa funzionalità un publisher comunica al gestore degli eventi la possibilità di generare nuovi eventi;
- *RegisterHierarchy* – È la funzionalità che consente la registrazione di un'intera gerarchia da parte di un producer;
- *DestroyRegistration* – È la funzionalità che permette l'eliminazione di una registrazione;
- *CreateTopic* – È la funzionalità che permette di creare una nuova classe di eventi;
- *ArchiveTopic* – È la funazionalità che permette di archiviare una classe di eventi. Dal momento dell'archiviazione, il topic non è più disponibile per la notifica di nuovi eventi; piuttosto, è ancora possibile ottenere gli eventi archiviati;
- *CreateHierarchy* – È la funzionalità che permette la creazione di una nuova gerarchia;
- *ArchiveHierarchy* – È la funzionalità che permette di archiviare un'intera gerarchia e tutti gli eventi prodotti ad essa correlati;
- *CreateAssociation* – È la funzionalità che consente la creazione dinamica di un'associazione tra topic all'interno di una gerarchia;
- *RemoveAssociation* – È la funzionalità che permette la rimozione di un'associazione tra topic all'interno di una gerarchia;
- *Notify* – È la funzionalità che permette la notifica di un evento al gestore;
- *GetAllTopics* – È la funzionalità che permette di ottenere la lista di tutti i topic accessibili. La lista può variare in funzione dei diritti e dei ruoli dell'attore richiedente;
- *GetAllHierarchies* – È la funzionalità che permette di ottenere tutte le gerarchie attive;
- *Subscribe* – È la funzionalità che permette la sottoscrizione di uno o più topic;
- *SubscribeHierarchy* – È la funzionalità che permette la sottoscrizione di un'intera gerarchia;
- *Renew* – È la funzionalità che permette di rinnovare una sottoscrizione;
- *Unsubscribe* – È la funzionalità che permette la rimozione dall'elenco dei sottoscrittori;

- *GetCurrentMessage* – È la funzionalità che permette di ottenere l'ultimo messaggio notificato per un determinato topic;
- *RetrieveEvents* – È la funzionalità che permette di ottenere tutti gli eventi notificati in relazione ad un determinato topic;
- *RetrieveHierarchy* – È la funzionalità che permette di ottenere tutti gli eventi notificati in relazione ad una determinata gerarchia;
- *CreateFederation* – È la funzionalità che consente la creazione di una nuova federazione;
- *JoinFederation* – È la funzionalità che consente ad un nodo l'ingresso in federazione;
- *Connect* – È la funzionalità che permette ad un nodo di connettersi logicamente ad un altro nodo della federazione;
- *LeaveFederation* – È la funzionalità che consente ad un nodo di uscire dalla federazione;
- *DissolveFederation* – È la funzionalità che permette la rimozione logica di una federazione;
- *RouteEvent* – È la funzionalità che consente la propagazione di un evento tra i nodi della federazione.

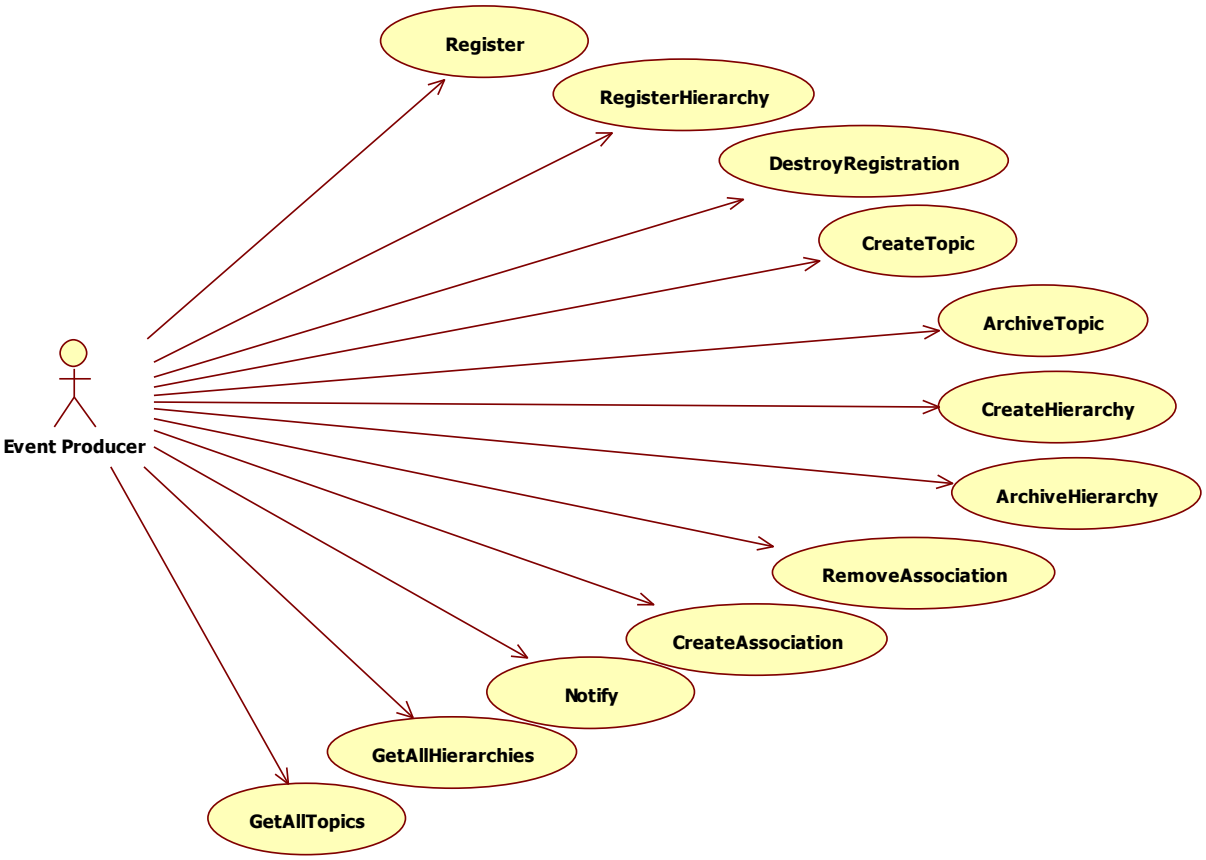


Figura 38. Casi d'uso per il produttore di eventi

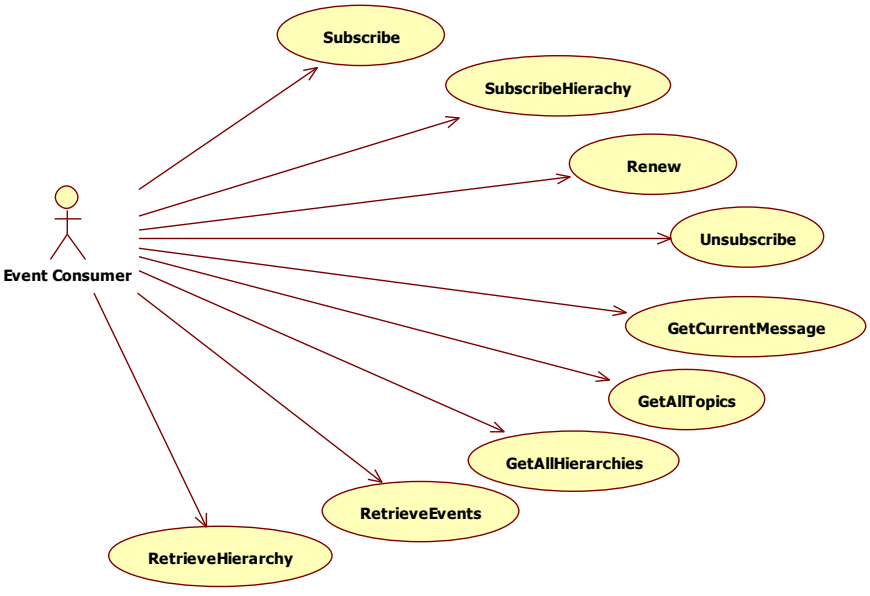


Figura 39. Casi d'uso per il consumatore di eventi

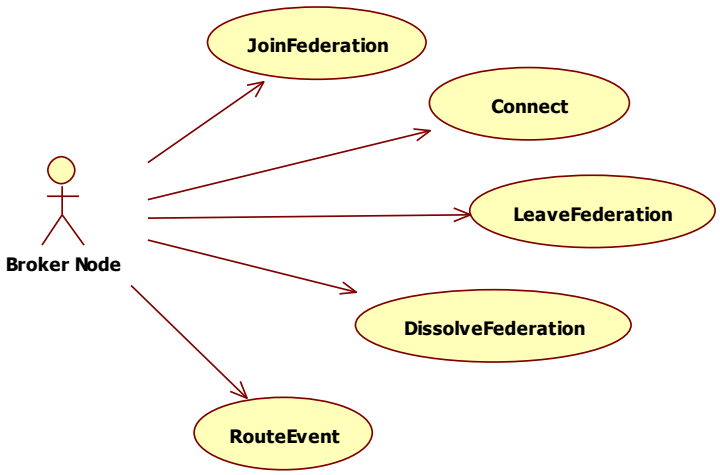


Figura 40. Casi d'uso per il gestore della federazione

8.3.5 Architettura della componente

Il *Gestore Gerarchico degli Eventi* si realizza attraverso un'architettura federata secondo il modello peer-to-peer. Per questo motivo ogni nodo supporta le interfacce *IPublisherRegistrationgMgt*, *ISubscriptionMgt*, *INotificationBrokerMgt* e *IBrokerFederationMgt* riportate in Figura 41.

L'interfaccia *IPublisherRegistrationgMgt* offre servizi per la gestione delle registrazioni da parte dei publisher. *ISubscriptionMgt*, invece, gestisce le sottoscrizioni effettuate dai consumer. L'interfaccia *INotificationBrokerMgt* supporta i meccanismi per la notifica e la gestione delle strutture dati, topic e gerarchie. Infine, *IBrokerFederationMgt* è l'interfaccia per la gestione delle federazioni di nodi broker.

Il modello informativo e l'architettura delle sotto-componenti è riportata in Figura 41, Figura 42, Figura 43 e Figura 44. Il *Gestore Gerarchico degli Eventi* consiste di tre sotto-componenti: il gestore delle registrazioni, il gestore delle sottoscrizioni ed il gestore delle notifiche e della federazione.

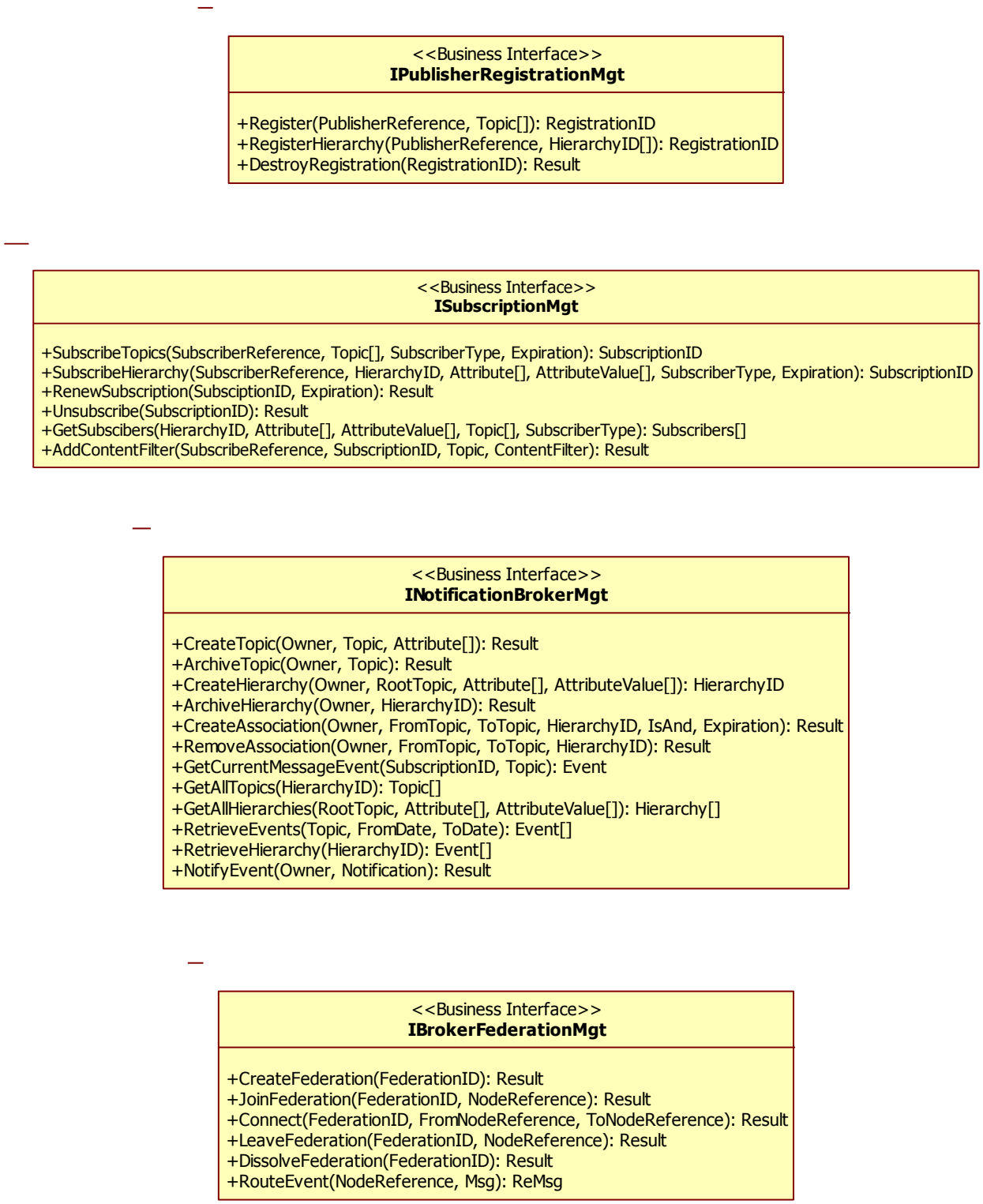


Figura 41. Interfacce supportate dai nodi broker

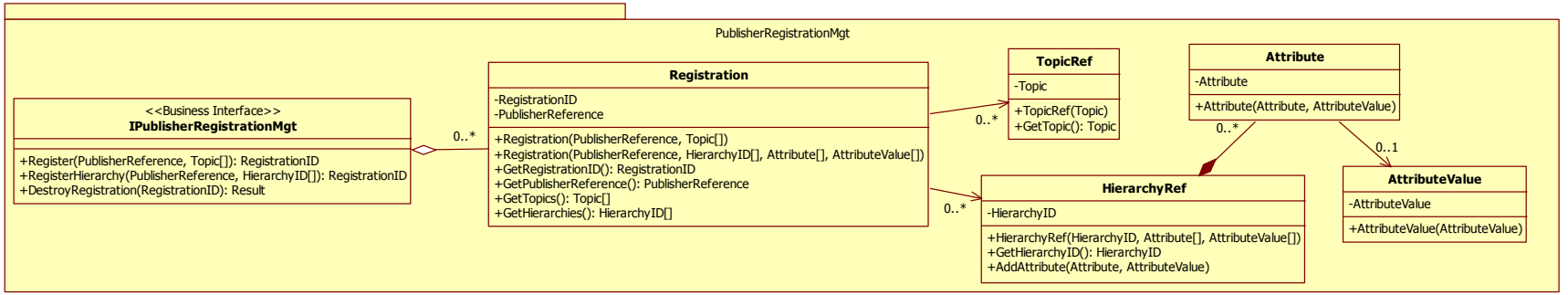


Figura 42. Interfaccia ed architettura della componente PublisherRegistrationMgt

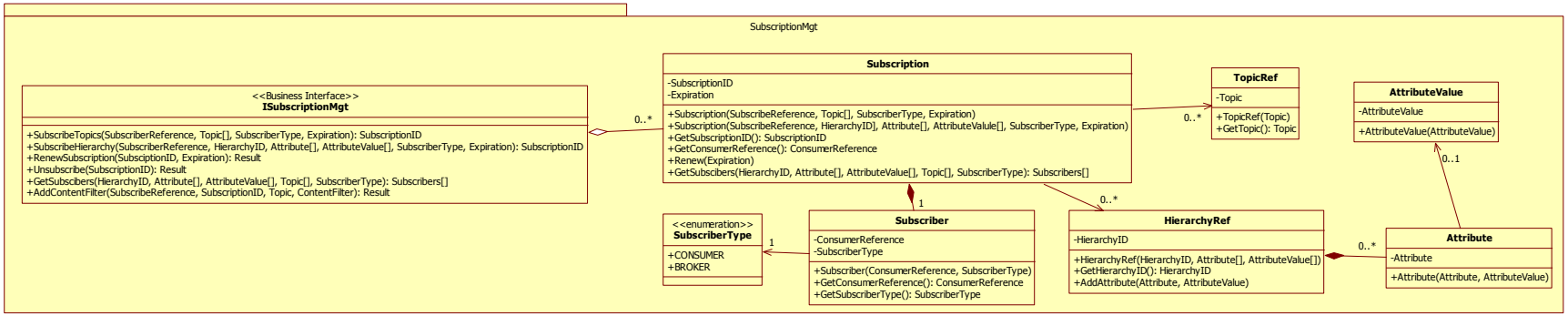


Figura 43. Interfaccia ed architettura della componente SubscriptionMgt

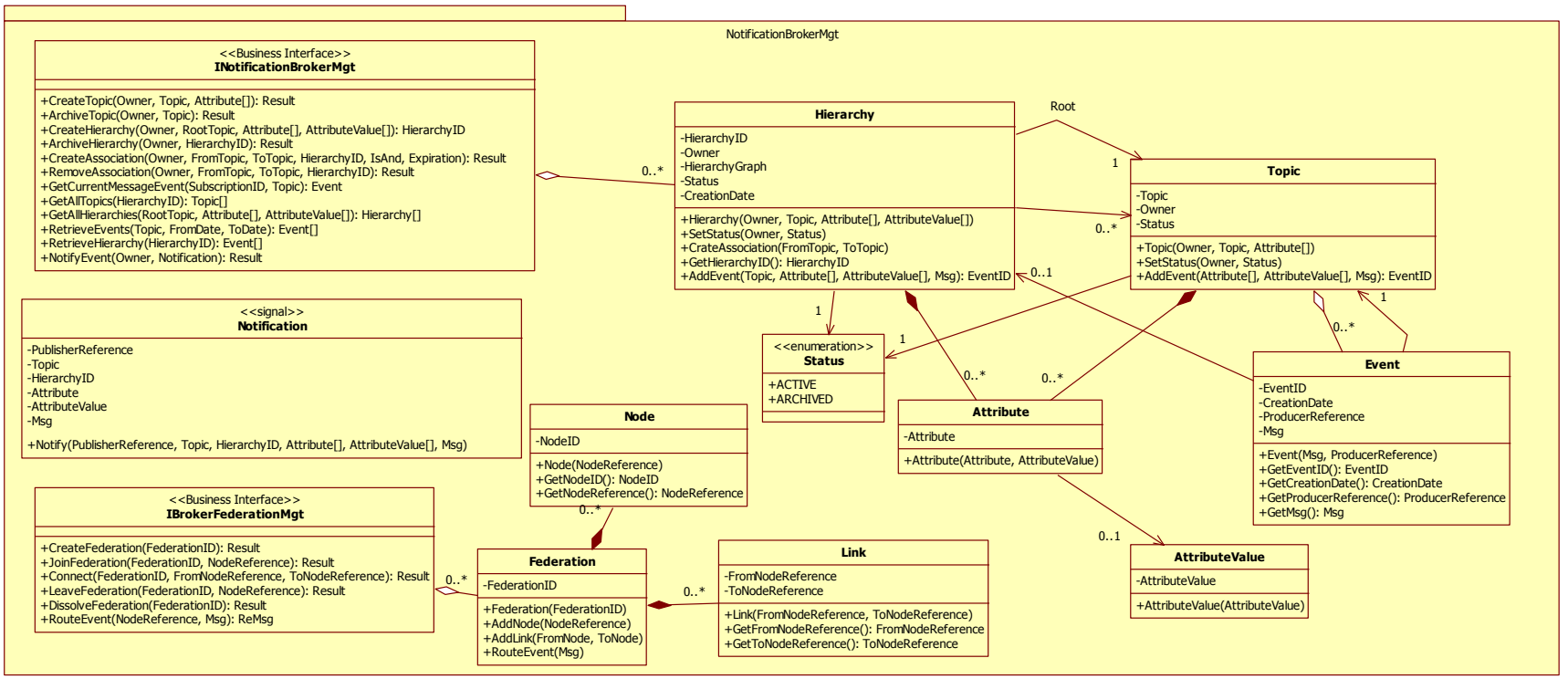


Figura 44. Interfaccia ed architettura della componente NotificationBrokerMgt

8.3.6 Integrazione con il Sistema Pubblico di Connettività

Le funzionalità specificate precedentemente possono essere esposte su Porta di Dominio al fine di consentire le interazioni tra i nodi broker regionali, nel rispetto delle norme vigenti in merito alla cooperazione applicativa tra Pubbliche Amministrazioni.

In particolare, le funzionalità che potranno essere escluse da tale processo sono quelle relative alla gestione delle registrazioni (interfaccia *IPublisherRegistrationiMgt*) e la funzionalità *GetSubscriber* (interfaccia *INotificationBrokerMgt*).

Le prime sono escluse in quanto ad essere propagate tra i diversi nodi della federazione saranno esclusivamente le sottoscrizioni; inoltre, ogni nodo broker regionale può ottenere la lista di sottoscrittori (siano essi consumatori finali o altri nodi broker regionali) direttamente dalle proprie strutture dati interne.

8.3.7 Scenari d'uso

Di seguito si riportano i principali scenari di interazione con il *Gestore Gerarchico degli Eventi*.

8.3.7.1 Registrazione topic

Un produttore può registrare uno o più topic al fine di notificare eventi appartenenti ad essi. La registrazione riguarda topic pre-esistenti; invece, la creazione di nuovi topic può avvenire attraverso la funzionalità *CreateTopic* dell'interfaccia *INotificationBrokerMgt*.

1. Il produttore di eventi richiede l'elenco di tutti i topic attraverso l'interfaccia *INotificationBrokerMgt*;
2. Il produttore ottiene i risultati;
3. Il produttore richiede la registrazione di una serie di topic disponibili;
4. Il servizio crea una nuova registrazione;
5. La registrazione aggiunge le referenze ai topic registrati;
6. Le referenze vengono associate alla registrazione;
7. La registrazione viene restituita all'interfaccia *IPublisherRegistrationMgt*;
8. L'interfaccia *IPublisherRegistrationMgt* richiede l'identificativo della nuova registrazione;
9. L'interfaccia *IPublisherRegistrationMgt* ottiene l'identificativo della nuova registrazione;
10. L'interfaccia *IPublisherRegistrationMgt* restituisce al produttore l'identificativo della nuova registrazione.

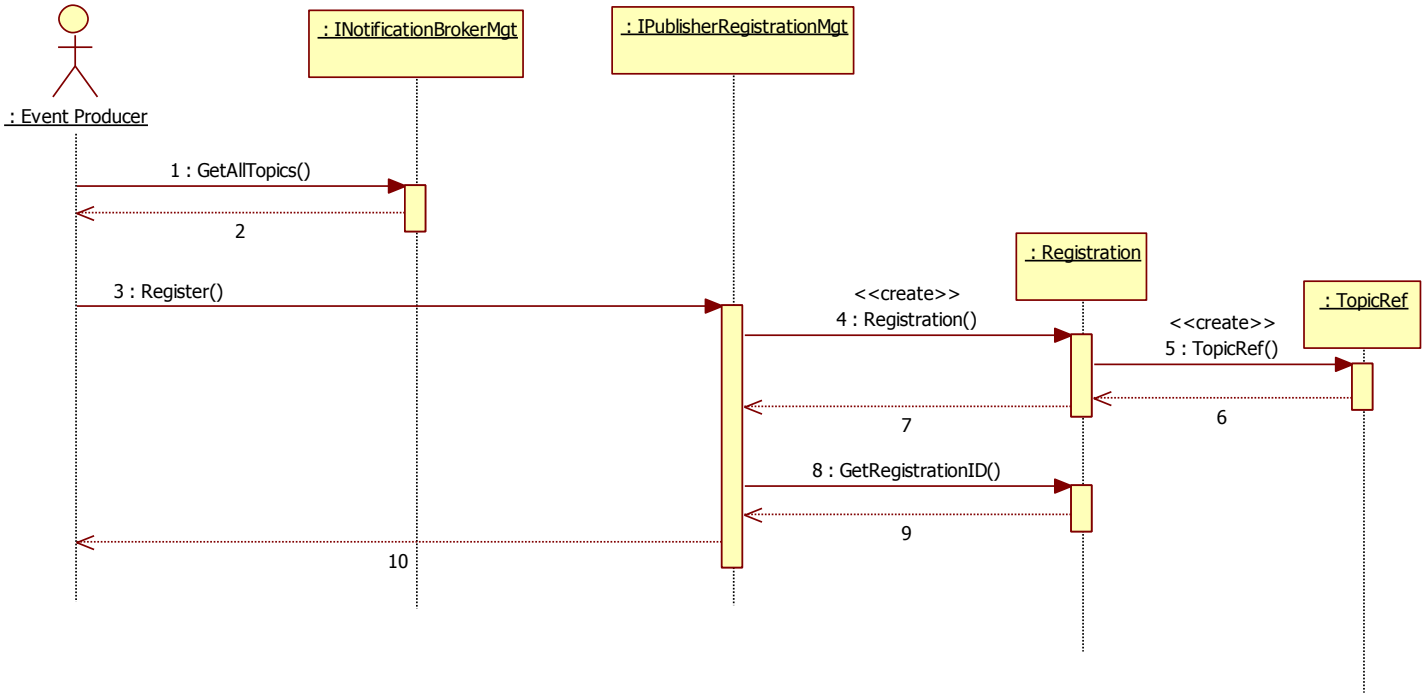


Figura 45. Scenario Registrazione topic

8.3.7.2 Registrazione gerarchia

Un produttore può registrare una o più gerarchie al fine di notificare eventi appartenenti ad esse.

1. Il produttore di eventi richiede l'elenco di tutte le gerarchie attive attraverso l'interfaccia *INotificationBrokerMgt*;
2. Il produttore ottiene i risultati;
3. Il produttore richiede la registrazione di una serie di gerarchie disponibili;
4. Il servizio crea una nuova registrazione;
5. La registrazione aggiunge le referenze alle gerarchie registrate;
6. Le referenze vengono associate alla registrazione;
7. La registrazione viene restituita all'interfaccia *IPublisherRegistrationMgt*;
8. L'interfaccia *IPublisherRegistrationMgt* richiede l'identificativo della nuova registrazione;
9. L'interfaccia *IPublisherRegistrationMgt* ottiene l'identificativo della nuova registrazione;
10. L'interfaccia *IPublisherRegistrationMgt* restituisce al produttore l'identificativo della nuova registrazione.

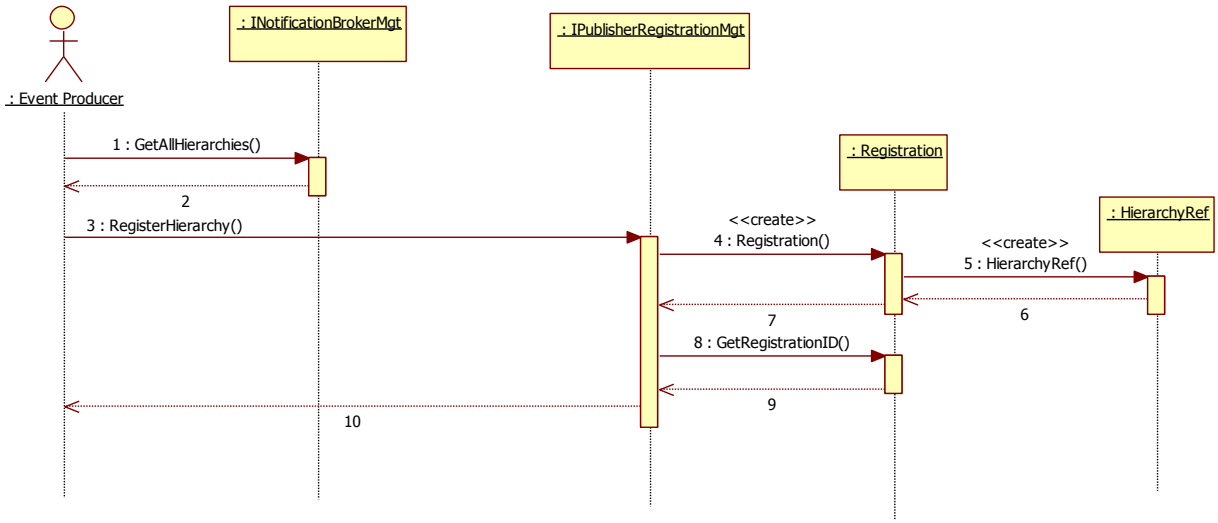


Figura 46. Scenario Registrazione gerarchia

8.3.7.3 Sottoscrizione Topic

Un consumatore può sottoscrivere uno o più topic.

1. Il consumatore di eventi richiede l'elenco di tutti i topic attraverso l'interfaccia *INotificationBrokerMgt*. Nella richiesta può specificare o meno la gerarchia di interesse; in quest'ultimo caso, otterrà l'elenco di tutti i topic;
2. Il consumatore ottiene i risultati;
3. Il consumatore richiede la sottoscrizione di una serie di topic disponibili;
4. Il servizio crea una nuova sottoscrizione;
5. La sottoscrizione aggiunge le referenze ai topic sottoscritti;
6. Le referenze vengono associate alla sottoscrizione;
7. La sottoscrizione viene restituita all'interfaccia *ISubscriptionMgt*;
8. L'interfaccia *ISubscriptionMgt* richiede l'identificativo della nuova sottoscrizione;
9. L'interfaccia *ISubscriptionMgt* ottiene l'identificativo della nuova sottoscrizione;
10. L'interfaccia *ISubscriptionMgt* restituisce al consumatore l'identificativo della nuova sottoscrizione.

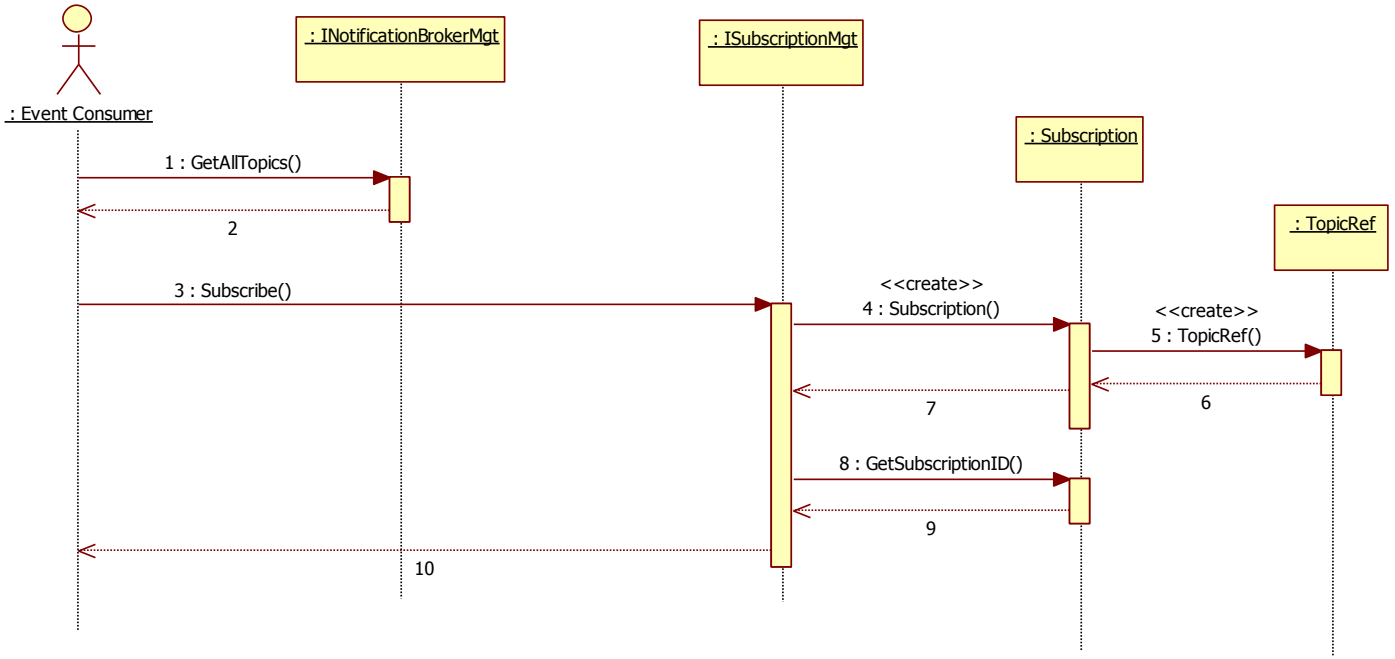


Figura 47. Scenario Sottoscrizione topic

8.3.7.4 Sottoscrizione gerarchia

Un consumatore può sottoscrivere una gerarchia o parte di essa. La sottoscrizione di un'intera gerarchia avviene specificando l'identificativo della gerarchia stessa; invece, una sotto-gerarchia può essere sottoscritta aggiungendo degli attributi (caratteristici di topic della gerarchia stessa) ed eventualmente i loro valori.

1. Il consumatore di eventi richiede l'elenco di tutte le gerarchie attraverso l'interfaccia *INotificationBrokerMgt*. Nella richiesta può specificare una serie di filtri (attributi e loro valore);
2. Il produttore ottiene i risultati;
3. Il consumatore richiede la sottoscrizione di una gerarchia o parte di essa attraverso la specifica di attributi e loro valori;
4. Il servizio crea una nuova sottoscrizione;
5. La sottoscrizione aggiunge le referenze alla gerarchia;
6. La sottoscrizione aggiunge le referenze agli attributi ed ai loro valori;
7. Le referenze vengono associate alla sottoscrizione;
8. La sottoscrizione viene restituita all'interfaccia *ISubscriptionMgt*;
9. L'interfaccia *ISubscriptionMgt* richiede l'identificativo della nuova sottoscrizione;
10. L'interfaccia *ISubscriptionMgt* ottiene l'identificativo della nuova sottoscrizione;
11. L'interfaccia *ISubscriptionMgt* restituisce al consumatore l'identificativo della nuova sottoscrizione.

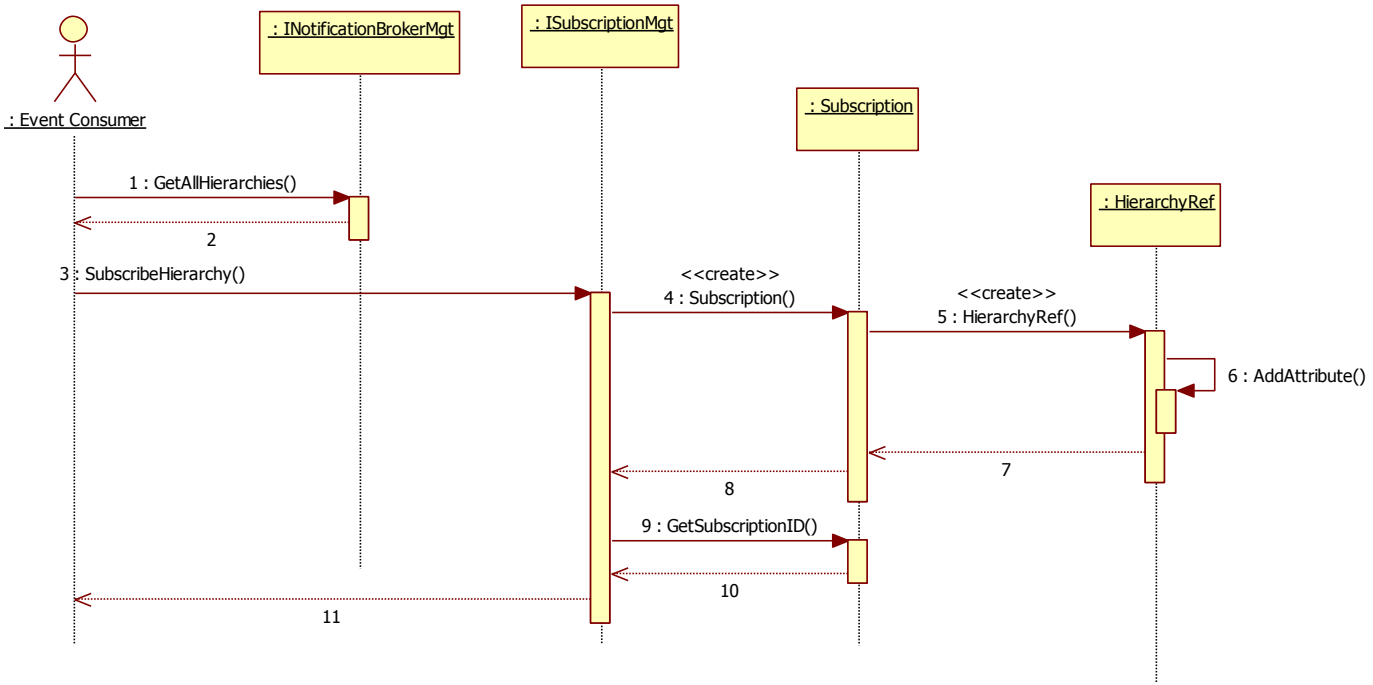


Figura 48. Scenario Sottoscrizione gerarchia

8.3.7.5 Creazione topic

Un produttore, o altro attore autorizzato, può creare nuove gerarchie di eventi.

1. Il produttore richiede la creazione di un nuovo topic all'interfaccia *INotificationBrokerMgt*;
2. Il servizio crea un nuovo topic con una lista di attributi;
3. Il topic aggiunge gli attributi;
4. Gli attributi vengono associati al topic;
5. Il topic viene restituito all'interfaccia *INotificationBrokerMgt*;
6. L'interfaccia *INotificationBrokerMgt* restituisce l'esito al produttore.

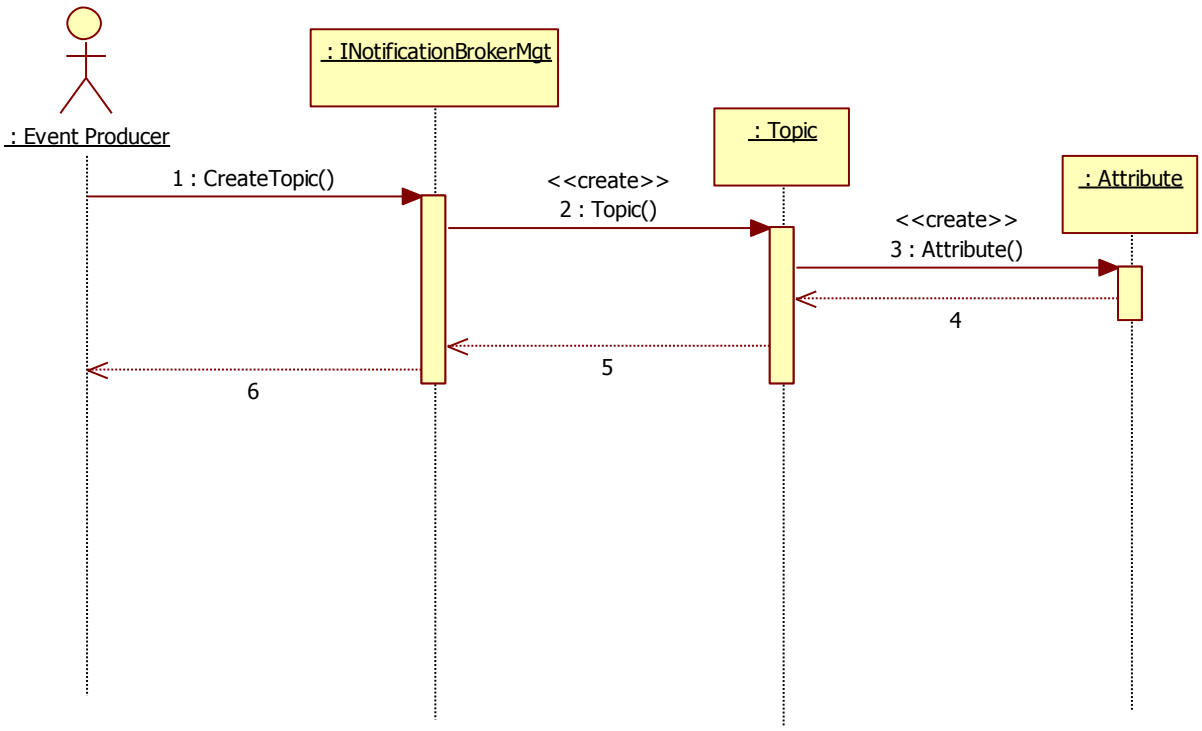


Figura 49. Scenario Creazione topic

8.3.7.6 Creazione gerarchia

Un produttore, o altro attore autorizzato, può creare nuove gerarchie di eventi.

La creazione di una nuova gerarchia avviene individuando un topic di riferimento (root), quindi una serie di attributi della gerarchia e di loro eventuali valori.

Successivamente, il produttore aggiunge ulteriori topic alla gerarchia.

1. Il produttore richiede la creazione di una nuova gerarchia all'interfaccia *INotificationBrokerMgt* indicando la *RootTopic* ed, eventualmente, una lista di attributi e loro valori;
2. Il servizio crea una nuova gerarchia;
3. La gerarchia aggiorna la propria *RootTopic*, la lista di attributi e valori, e setta lo stato ad ACTIVE. Da questo momento in poi la gerarchia è attiva per la produzione di eventi;
4. La gerarchia viene restituita all'interfaccia *INotificationBrokerMgt*;
5. L'interfaccia *INotificationBrokerMgt* richiede l'identificativo alla nuova gerarchia;
6. L'interfaccia *INotificationBrokerMgt* ottiene l'identificativo della nuova gerarchia;
7. Il controllo viene restituito al produttore;
8. Il produttore aggiunge un nuovo topic specificando la relazione con un topic già presente nella gerarchia.
9. L'interfaccia *INotificationBrokerMgt* associa il topic con la gerarchia specificata;
10. L'associazione viene restituita all'interfaccia *INotificationBrokerMgt*;
11. L'associazione viene restituita al produttore.

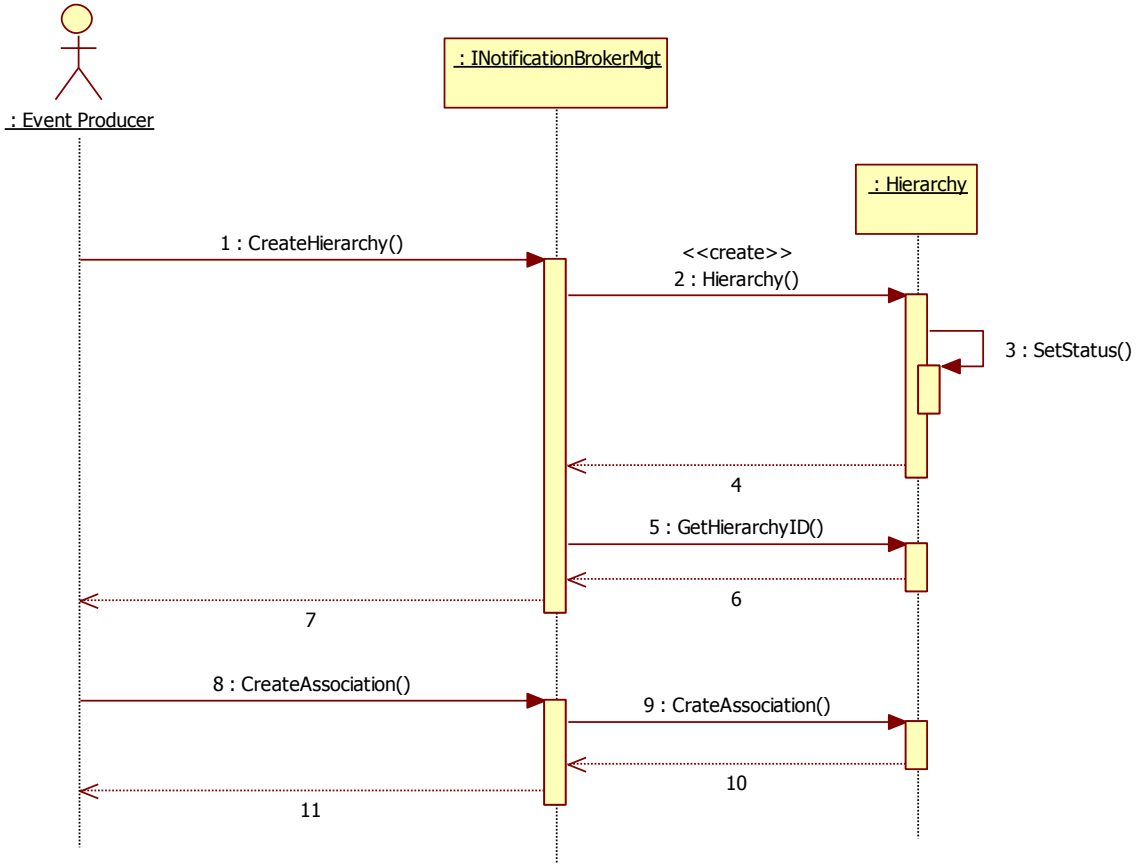


Figura 50. Scenario Creazione gerarchia

8.3.7.7 Archiviazione gerarchia

Il proprietario della gerarchia può richiederne l'archiviazione. L'operazione di archiviazione rende la gerarchia indisponibile per la creazione e notifica di nuovi eventi, ma consente il recupero degli eventi associati.

1. Il produttore richiede l'archiviazione della gerarchia;
2. Il servizio richiede la modifica dello stato della gerarchia;
3. La gerarchia aggiorna il proprio stato dopo aver verificato le credenziali del produttore.
4. L'interfaccia *INotificationBrokerMgt* restituisce l'esito al produttore.

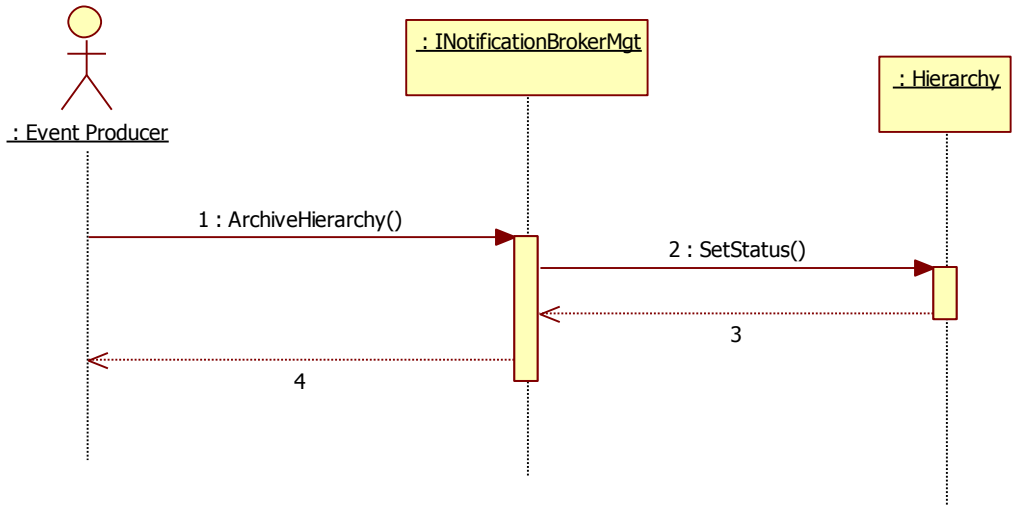


Figura 51. Scenario Archiviazione gerarachia

8.3.7.8 Notifica di un evento

L'operazione di notifica di un evento da parte di un produttore riguarda un evento di un determinato topic, oppure di un topic di una gerarchia.

1. Il produttore richiede il servizio di notifica all'interfaccia *INotificationBrokerMgt*;
2. L'interfaccia *INotificationBrokerMgt* analizza la richiesta;
3. L'interfaccia *INotificationBrokerMg* richiede la creazione di un nuovo evento;
4. La richiesta viene inoltrata al topic;
5. La richiesta crea un nuovo oggetto *Event*;
6. L'oggetto *Event* viene restituito al topic;
7. L'oggetto *Topic* richiede l'identificativo del nuovo evento;
8. L'identificativo viene restituito all'oggetto *Topic*;
9. L'identificativo viene restituito all'oggetto *Hierarchy*;
10. L'identificativo viene restituito all'interfaccia *INotificationBrokerMgt*;
11. L'interfaccia *INotificationBrokerMgt* richiede la lista di sottoscrittori di tipo CONSUMER all'interfaccia *ISubscriptionMgt*;
12. L'interfaccia *INotificationBrokerMgt* ottiene la lista di consumatori;
13. L'interfaccia *INotificationBrokerMgt* notifica l'evento ai consumatori locali;
14. L'interfaccia *INotificationBrokerMgt* invia la notifica ai consumatori propagandola alla federazione di broker.

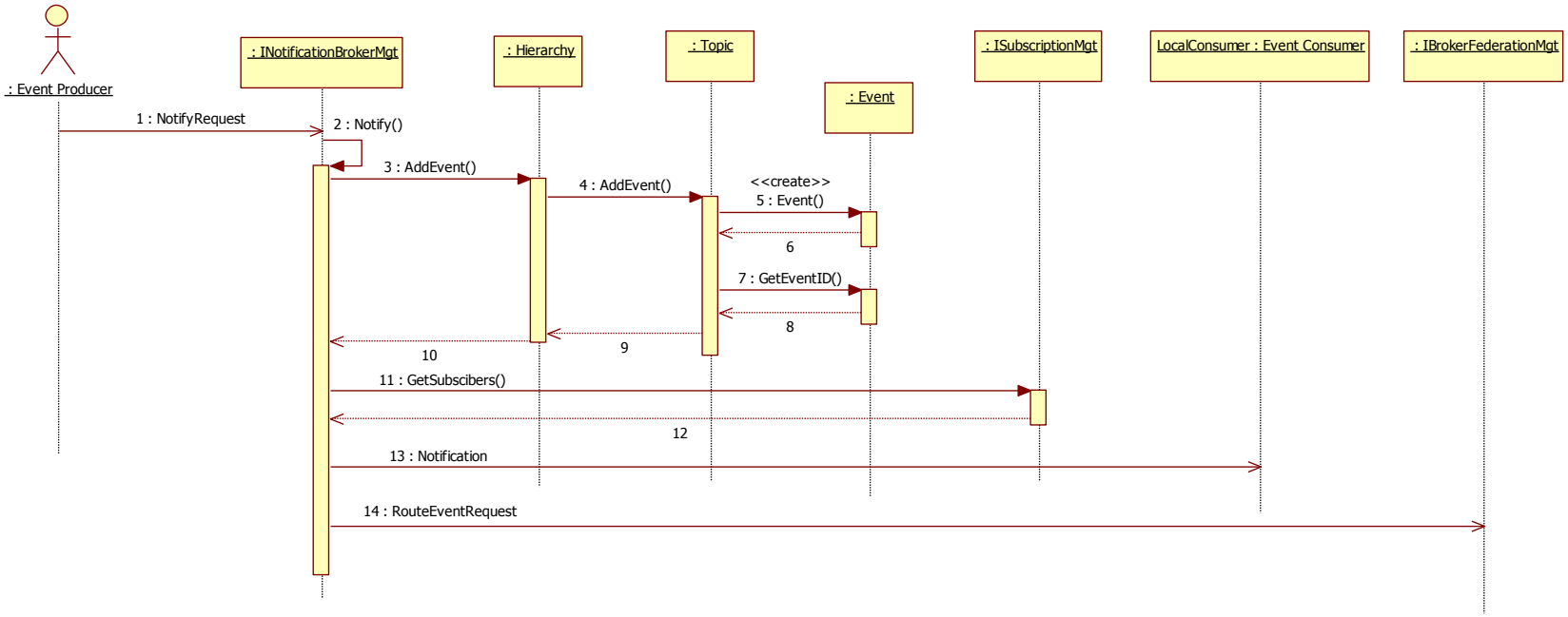


Figura 52. Scenario Notifica di un evento

8.4 Gestore dei Documenti

Il *Gestore dei Documenti* deve memorizzare in maniera persistente, affidabile e sicura i documenti creati da un utente autorizzato ad ogni occorrenza di un evento sanitario di un assistito. Tale memorizzazione deve avvenire all'interno di opportuni repository. Questi ultimi devono essere localizzati presso i nodi locali o regionali. È fondamentale che queste componente siano ad alta affidabilità e che siano in grado di conservare i documenti sanitari garantendone disponibilità, consistenza e non ripudio dei dati.

Si noti che l'attuale legislazione farebbe necessariamente propendere per una memorizzazione sui nodi locali. Tuttavia requisiti di alta affidabilità (disponibilità h24, no single point-of-failure) richiederebbero necessariamente meccanismi di replicazione dei dati.

Le principali operazioni che il *Gestore dei Documenti* deve offrire sono le seguenti:

- reperimento di uno o più documenti sanitari disponibili in un repository a partire da un riferimento;
- archiviazione di un documento sanitario all'interno di un repository.

I documenti sanitari, preferibilmente, devono essere strutturati secondo lo standard HL7-CDA Rel. 2.0, anche se il *Gestore dei Documenti* deve essere capace di gestire altri formati.

Ogni nodo dell'Infrastruttura, sia esso locale che regionale, può interagire con i repository, attraverso le componenti *Gestore dei Documenti*.

Il *Gestore dei Documenti* può quindi essere installato presso il nodo regionale o presso i nodi locali.

8.4.1 Attori e ruoli

È possibile identificare i seguenti attori e ruoli negli scenari di interazione con la componente:

- *DocumentProducer* – È l'entità capace di produrre nuovi documenti;
- *DocumentConsumer* – È l'entità capace di acquisire documenti preesistenti.

8.4.2 Casi d'uso

Si definiscono i seguenti casi d'uso:

- *AddDocument* – È la funzionalità che permette di richiedere la memorizzazione di un nuovo documento al gestore;
- *UpdateDocument* – È la funzionalità che permette di aggiornare un documento preesistente;
- *RetrieveDocument* – È la funzionalità che permette di ottenere un documento dal gestore.

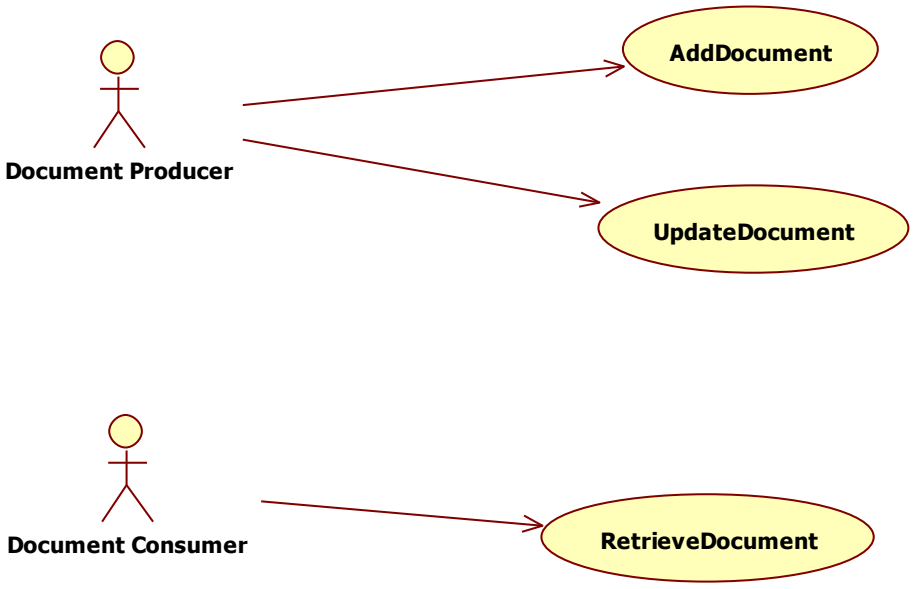


Figura 53. Casi d'uso per il Gestore dei Documenti

8.4.3 Architettura della componente

Il *Gestore dei Documenti* si realizza attraverso un'unica sotto-componente di gestione dei documenti.

Le interfacce e l'architettura della sotto-componente sono descritte in Figura 54 ed in Figura 55.

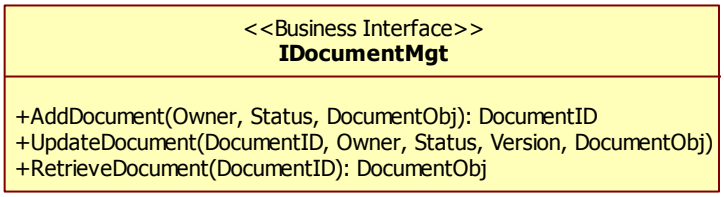


Figura 54. Interfaccia IDocumentMgt

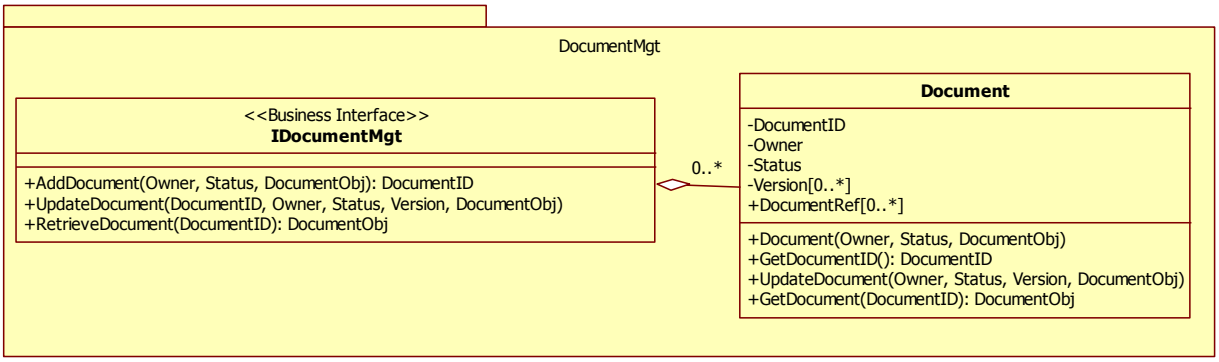


Figura 55. Interfaccia ed architettura della componente DocumentMgt

8.4.3.1 Integrazione con il Sistema Pubblico di Connettività

Il *Gestore dei Documenti* non deve necessariamente esporre servizi su Porta di Dominio; piuttosto, l'accesso alle funzionalità è garantito attraverso la componente *Interfaccia di Accesso*.

8.4.4 Scenari d'uso

Di seguito si riportano i principali scenari di interazione con il *Gestore dei Documenti*.

8.4.4.1 Caricamento documento

Un produttore può inserire un nuovo documento.

1. Il produttore richiede il caricamento di un nuovo documento attraverso l'interfaccia *IDocumentMgt*;
2. L'interfaccia *IDocumentMgt* crea una nuova istanza di documento;
3. L'interfaccia *IDocumentMgt* ottiene l'istanza di documento;
4. L'interfaccia *IDocumentMgt* richiede l'ID della nuova istanza di documento;
5. L'interfaccia *IDocumentMgt* ottiene l'ID della nuova istanza di documento;
6. L'interfaccia *IDocumentMgt* restituisce al produttore l'identificativo della nuova istanza di documento.

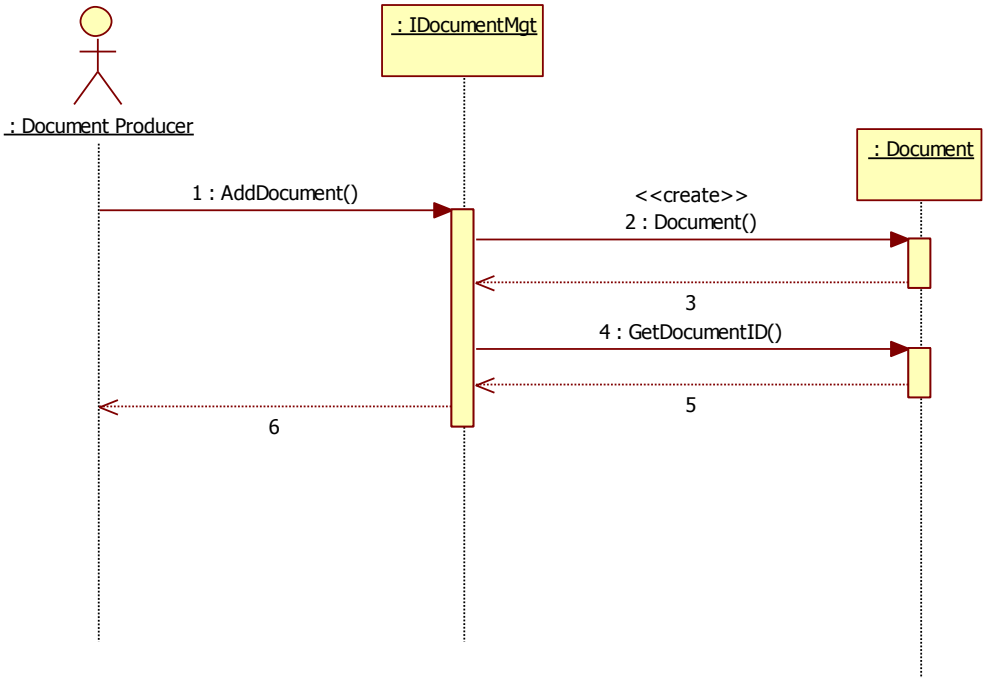


Figura 56. Scenario Caricamento documento

8.4.4.2 Aggiornamento documento

Un produttore può aggiornare un documento preesistente. L'aggiornamento può riguardare la versione del documento, in tal caso il gestore gestirà il versioning mantenendo le versioni precedenti, oppure lo stato del documento stesso.

1. Il produttore richiede l'aggiornamento di un documento preesistente attraverso l'interfaccia *IDocumentMgt*;
2. L'interfaccia *IDocumentMgt* richiede l'aggiornamento all'istanza di documento;
3. Il controllo viene restituito all'interfaccia *IDocumentMgt*;
4. Il controllo viene restituito al produttore.

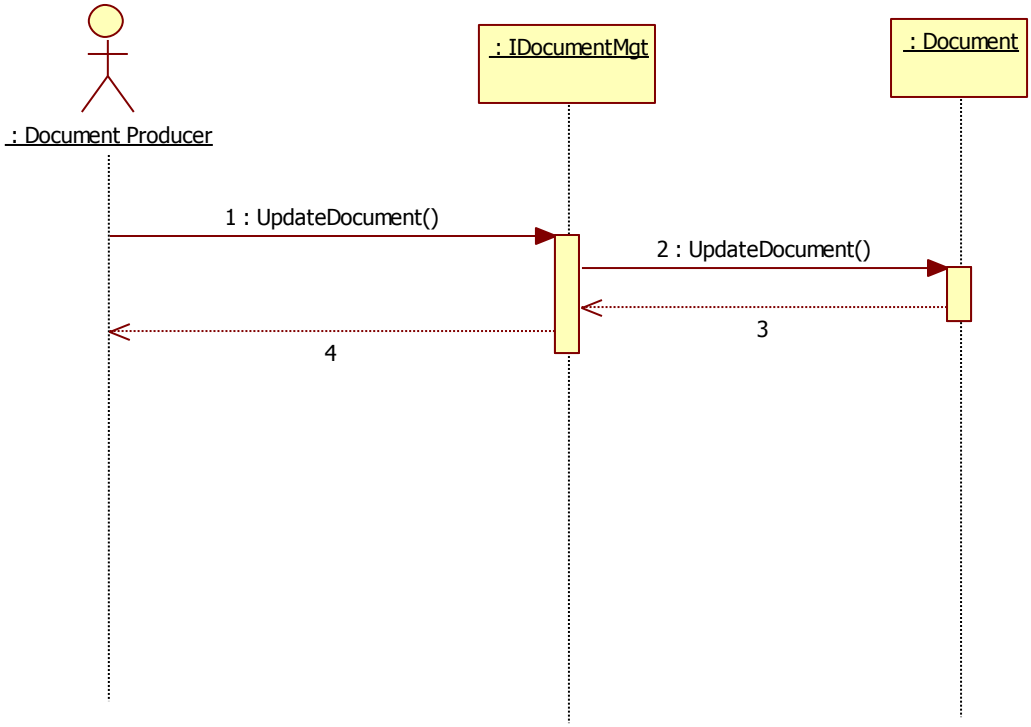


Figura 57. Scenario Aggiornamento documento

8.4.4.3 Recupero documento

Un consumatore può richiedere un documento attraverso il suo identificativo.

1. Il consumatore richiede il documento attraverso l'interfaccia *IDocumentMgt*;
2. L'interfaccia *IDocumentMgt* ricerca l'istanza di documento e richiede il documento in formato elettronico;
3. L'interfaccia *IDocumentMgt* ottiene il documento in formato elettronico;
4. L'interfaccia *IDocumentMgt* restituisce il documento al consumatore.

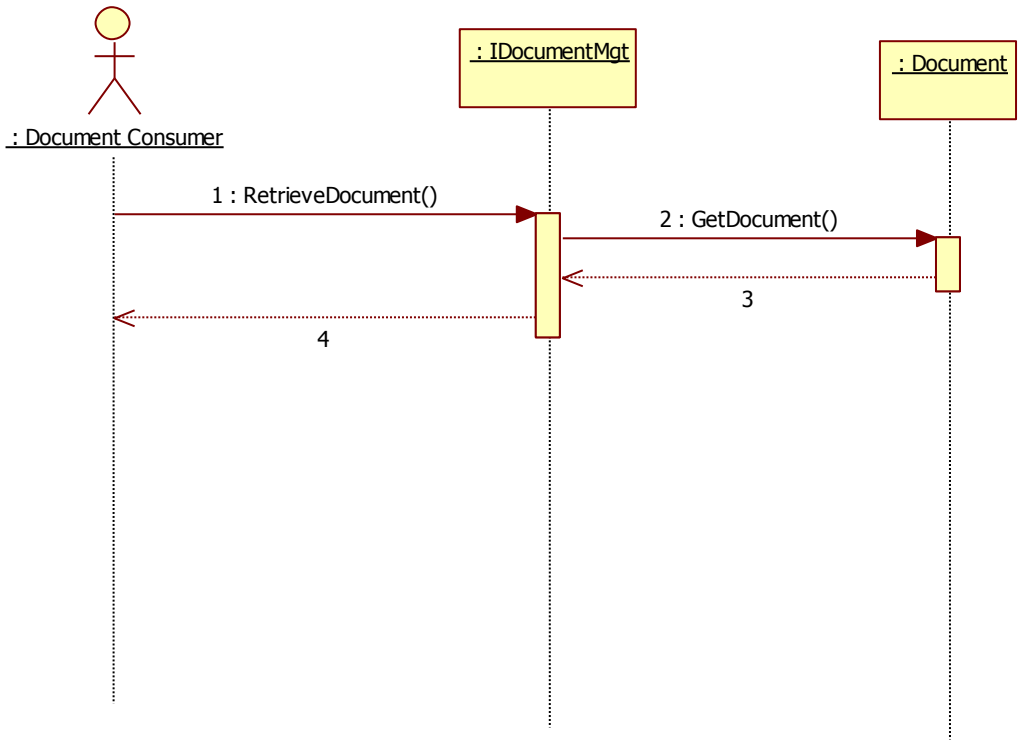


Figura 58. Scenario Recupero documento

8.5 Gestore delle Politiche di Accesso

Il *Gestore delle Politiche di Accesso* è responsabile degli aspetti generali di sicurezza sia per i servizi infrastrutturali che per quelli di tipo applicativo. L'approccio adottato segue il paradigma *security as a service*, tipico delle architetture orientate ai servizi, per l'implementazione di un Single Sign-On (SSO).

Di seguito, si è fatto riferimento a quanto indicato nel progetto SPC [11] e nel progetto ICAR [12]. In particolare, sono stati approfonditi i seguenti aspetti:

1. *Processo di autorizzazione*, ovvero le funzionalità e le interfacce che espongono servizi a supporto della fase di decisione; è richiamato a valle della fase di identificazione per autorizzare l'accesso ai servizi una volta conosciuti la totalità dei parametri necessari; allo scopo, questa fase comprende l'eventuale produzione di asserzioni necessarie alla fase di autorizzazione, nel caso in cui quelle prodotte nella fase di identificazione non fossero sufficienti.
2. *Processo di interazione*, ovvero i diversi messaggi che sono scambiati nella comunicazione con l'Infrastruttura (nel seguito del paragrafo anche indicata come *erogatore*) da parte di un sistema che funge da applicazione utente o che operi in modalità di cooperazione applicativa (nel seguito del paragrafo nominato anche *fruitore*).

Quest'ultimo aspetto è fondamentale, perché l'infrastruttura di sicurezza deve riguardare sia le comunicazioni interne ad un dominio (ad es. fruitore MMG/PLS), sia quelle tra domini SPC. In quest'ultimo caso lo scenario di interazione cambia, poiché si è in presenza di un'interazione tra due servizi applicativi offerti da due distinti Service Provider (fruitore ed erogatore), ma potrebbe essere comunque subordinato al trasferimento di opportune informazioni di autenticazione e autorizzazione utente. All'interno di ciascun dominio, quindi, sarebbe opportuno implementare e gestire le misure di sicurezza, in modo coerente con l'infrastruttura di sicurezza tra domini definita da SPC.

Indipendentemente dal dispiegamento dell'architettura di InFSE, è necessario proteggere le componenti della stessa (quali ad es. l'*Interfaccia di Accesso* ed il *Registro Indice Federato*) attraverso uno strato specifico di sicurezza, che a sua volta si compone di tre macro-componenti, rispettivamente per le funzioni di *Autenticazione*, *Identificazione* ed *Autorizzazione*.

Il Service Provider di InFSE prevede quindi l'esposizione dei servizi in modo da attivare le tre macro-componenti sopra descritte. Inoltre, deve consentire anche di modificare o configurare le componenti da proteggere in maniera tale da sfruttare tali servizi di sicurezza.

Oltre a ciò, è necessaria la presenza di specifici ulteriori entità, sia per il reperimento delle informazioni opportune al funzionamento delle macro-componenti, sia per la gestione delle identità federate così come indicato nel modello *Gestione Federata Identità SPC*.

Nel presente capitolo sono trattati i principali aspetti semantici per la gestione della

sicurezza in InFSE, in particolare relativi alla progettazione delle componenti coinvolte nella fase di Autorizzazione (ad es., il Policy Enforcement Point), deputate a fornire le decisioni di autorizzazione all'accesso ai servizi dell'Infrastruttura InFSE. Le componenti sono previste nel modello architetturale complessivo della sicurezza e sono richiamate in genere a valle della fase di identificazione, ed il loro compito è implementare le policy previste per il particolare dominio e/o contesto di invocazione del servizio di InFSE.

Le scelte architetturali di InFSE hanno evidenziato l'esigenza di adottare a pieno il modello XACML (soprattutto per la flessibilità di recuperare asserzioni in alcuni casi non presenti nella richiesta di servizio, ad esempio la relazione medico-paziente dal punto di vista organizzativo e dal punto di vista temporale), affiancato da una gestione articolata del controllo degli accessi basato sul ruolo (RBAC).

Dal punto di vista prettamente tecnologico, l'affermarsi del modello XACML ha risolto le problematiche di autorizzazione in tutte le componenti; peraltro, nell'ambito del Sistema Pubblico di Connettività (SPC), oramai è maturo l'assioma che debba essere possibile il passaggio e la relativa interpretazione delle politiche di sicurezza da un domino all'altro (anche attraverso le Porte Di Dominio); rimangono comunque da consolidare gli aspetti semantici dei ruoli e quelli dichiarativi delle regole da applicare, ovvero l'insieme delle policy di autorizzazione che si intendono applicare in un particolare dominio.

Questi aspetti sono fondamentali per garantire l'interoperabilità tra più sistemi di FSE che comprendono informazioni inerenti allo stesso cittadino. Infatti, in assenza di una formalizzazione di ruoli e di regole, i diversi richiedenti coinvolti nella cura ed assistenza del cittadino (ad es., il MMG o il medico specialista) appartenenti a domini diversi (Regioni, ASL), non riuscirebbero ad attingere in maniera unica alle informazioni, poiché la regola impiegata per uno stesso ruolo, ad esempio "Medico di Pronto Soccorso", potrebbe non essere facilmente interpretabile tra i diversi domini.

È evidente che ciò impatterebbe sull'Infrastruttura InFSE che, viceversa, ha proprio l'obiettivo della "ricomposizione".

Nella gestione della sicurezza, in generale, occorre quindi prevedere non solo entità che mirano alla federazione delle identità, ma anche altre che rendono possibile l'interoperabilità delle politiche di accesso, a garanzia di sicurezza fra tutti i domini coinvolti.

L'interoperabilità delle politiche di accesso passa da:

- una terminologia condivisa dei ruoli possibili del richiedente (ad es. Assistito, Medico di Medicina Generale, Pediatra di Libera Scelta, Guardia Medica, Medico di Reparto, Specialista Ambulatoriale, Medico di Emergenza, Farmacista, Operatore CUP, Operatore Accettazione, Operatore Amministrativo, etc.);
- una serie di entità (ad es. Attribute Authority) con interfaccia standard, in grado di restituire attributi anche non presenti nelle prime asserzioni di accesso (ad es. codice fiscale, assistito al quale afferisce il servizio/documento oggetto di autorizzazione, tipo di documento che viene trattato, relazione fra l'utente o l'operatore che fa la richiesta e l'assistito a cui essa si riferisce, struttura di appartenenza del richiedente, etc.);

- una descrizione aperta e condivisa delle regole (ad es., se il ruolo dell'operatore è "Medico di Reparto", l'assistito soggetto delle informazioni accedute deve essere ricoverato in quel momento nella struttura dell'operatore stesso).

L'autorizzazione per l'accesso al servizio sarà quindi richiesta essenzialmente sulla base del ruolo dell'operatore, comunque correlato da altre informazioni necessarie per l'autorizzazione, come ad esempio l'assistito, il tipo di documento, il livello di riservatezza, la struttura organizzativa che detiene il documento e così via.

8.5.1 Fase di autenticazione

La fase di autenticazione ha come scopo quello di determinare un cosiddetto *trusted context environment* fra client (fruitore) e fornitore del servizio (erogatore). Tale fase si realizza attraverso l'invio di credenziali che rappresentano l'identità da parte del fruitore e da parte dell'erogatore in modo che entrambi possano attestare la corrispondente attendibilità.

La verifica dell'identità di un utente è una proprietà di sicurezza molto importante in quanto, in generale, i diritti di accesso alle risorse concessi all'utente dipendono dall'identità o da fattori (o attributi, come ad esempio il ruolo) connessi all'identità stessa. Inoltre, la verifica dell'identità è anche importante per il logging degli eventi, in quanto ad ogni evento che viene registrato deve essere associato l'identità del soggetto che lo ha generato.

Nel caso dell'Infrastruttura InFSE, l'autenticazione del soggetto che accede ai referti medici memorizzati nei vari archivi assume una importanza fondamentale. Infatti i dati che sono registrati dall'Infrastruttura sono sensibili, e devono poter essere letti solo da personale medico autorizzato per scopi di prevenzione, diagnosi e cura del paziente. Quindi, occorre che l'identità dei soggetti che accedono ai dati sia verificata dal sistema. A tal proposito, risulta necessario assegnare a questi ultimi appropriati diritti di lettura, modifica e cancellazione dei dati, tenendo conto della finalità per cui vi accedono.

Il processo di autenticazione deve adottare la modalità forte, con l'uso ad esempio di una smart card e di un PIN segreto.

L'autenticazione debole, con l'uso di sole credenziali di tipo username e password per utenti con permessi limitati, è possibile solo per una fase transitoria limitata nel tempo, ed è comunque sconsigliata.

L'autenticazione forte, invece, prevede l'utilizzo di due o più mezzi di autenticazione contemporaneamente: ad esempio, è possibile sfruttare una smart card di tipo CNS (in genere contenente due certificati di tipo X.509, uno per l'autenticazione e l'altro per la firma digitale) o CSE (Carta Sanitaria Elettronica). Quest'ultima, in particolare, è una smart card appositamente introdotta allo scopo di permettere l'accesso al FSE, contenente i dati anagrafici e identificativi per l'assistenza sanitaria dell'utente, con l'obiettivo principale di permettere l'autenticazione forte dell'utente: è previsto,

infatti, che essa sostituirà l'attuale Tessera Sanitaria Nazionale, la Tessera europea di assicurazione malattia e il Tesserino del codice fiscale, includendo anche un certificato digitale conforme agli standard della Carta Nazionale dei Servizi, necessario per autenticare il proprietario, e un codice personale di accesso segreto (PIN).

Dopo aver accertato l'autenticazione dell'erogatore, il fruitore del servizio può richiamare l'endpoint applicativo corredando un token (Username, X.509) nel rispetto della specifica WS-Security. Ciò può consentire all'erogatore del servizio di verificare l'attendibilità del fruitore. In alcuni casi, è possibile anche coinvolgere un soggetto terzo in grado di generare una chiave temporanea e sfruttare quest'ultima per la cifratura del canale.

Durante il processo di autenticazione, di conseguenza, è necessario invocare i servizi esposti da entità che fungono da *Certification Authority (CA)* per la validazione dei certificati digitali.

8.5.2 Fase di identificazione

La fase di identificazione ha lo scopo di determinare l'identità del fruitore (utente) ed eventualmente proporre tutte e sole le operazioni consentite. Le informazioni necessarie per l'identificazione, ovvero il set minimo di informazioni utili per identificare un soggetto sulla base delle credenziali fornite, vengono indicate nella descrizione del servizio esposto dall'erogatore.

In questo caso, il client che intende accedere al servizio offerto dal Service Provider richiede a quest'ultimo la descrizione della interfaccia esposta e le policy di accesso: invia una richiesta (ad es., *GetMetadata()*, sfruttando il protocollo WS-MEX), e ottiene una risposta (ad es. l'interfaccia WSDL e le WS-Policy); tra le WS-Policy ottenute, saranno elencate anche le policy di sicurezza.

Il fruitore potrebbe già essere in possesso delle asserzioni necessarie. Nel caso in cui il fruitore fosse parzialmente o totalmente impossibilitato a fornirle nella forma richiesta dall'erogatore (il Service Provider), si rende necessaria la presenza di ulteriori attori. In maniera conforme a quanto stabilito nell'ambito del progetto ICAR, il fruitore può essere veicolato verso una triade di entità: *Identity Provider (IdP)*, *Profile Authority (PA)*, *Attribute Authority (AA)*.

Nel semplice caso in cui l'erogatore non necessita di particolari asserzioni per gestire le politiche di sicurezza, l'autenticazione può coincidere con l'identificazione. In questo caso, infatti, l'erogatore può invocare una componente dell'IdP, detto *Security Token Service (STS)*, responsabile di rilasciare token di tipo SAML [13] a valle dell'inoltro, ad esempio, di un certificato digitale di tipo X.509 per l'autenticazione dell'operatore. Il STS consente, inoltre, di generare un token SAML per i sistemi esterni che non sono in grado di fornirlo autonomamente o che non l'hanno ereditato da precedenti accessi.

Nei casi più complessi, l'IdP accede alle entità AA al fine di ricavare ulteriori informazioni, come quelle necessarie per identificare il fruitore e per assegnare i ruoli che questo assume nel dominio sanitario. Queste informazioni sono utili per il processo di autorizzazione.

I basamenti informativi, come le Anagrafiche Sanitarie, le Anagrafiche Operatori, le Correlazioni Cittadino-Operatore Sanitario e così via, sono Attribute Authority SAML (come specificato in SAML 2.0) che possono essere ottenuti mediante interrogazioni basate sul protocollo AttributeQuery.

Nei casi di gestione avanzata delle asserzioni e di gestione delle identità federate, la componenente che viene coinvolta prioritariamente è il PA, che svolge il ruolo di mediatore tra i diversi IdP (eventualmente presenti in altri domini) e le diverse AA (eventualmente presenti in altri domini) per la composizione del portafoglio di asserzioni per i cosiddetti Profili di Autorizzazione.

Il fruitore può a questo punto utilizzare il portafoglio di asserzioni appena costruito per accedere ad uno dei servizi protetti dell'erogatore.

8.5.3 Fase di autorizzazione

Il processo di autorizzazione è il processo decisionale tramite il quale vengono verificati i diritti dei soggetti che richiedono l'accesso alle varie componenti dell'Infrastruttura FSE per reperire o modificare i dati memorizzati. Il processo di autorizzazione viene eseguito dopo le fasi di autenticazione e di identificazione, nelle quali è stata invece verificata l'identità del soggetto richiedente l'accesso ed è stato costruito il portafoglio di asserzioni che il soggetto intende utilizzare nella fase di autorizzazione, e dopo aver effettuato la verifica dell'autenticità delle credenziali contenute nel portafoglio di asserzioni. Infatti, data la criticità dei dati memorizzati dall'Infrastruttura del FSE, è necessario che l'accesso a qualsiasi componente del sistema, sia per reperire direttamente dati sanitari che per reperire metadati, venga regolato da un processo di autorizzazione che verifica se il soggetto richiedente ha effettivamente il diritto di compiere il tipo di accesso richiesto. Ad esempio, considerando il caso d'uso di acquisizione di documenti clinici, la fase di autorizzazione deve essere eseguita per ogni richiesta ricevuta dai *Registri Regionali*, perché anche i metadati sono da considerare informazioni sensibili, e devono essere acceduti solo dai soggetti aventi diritto. Il processo di autorizzazione utilizza le informazioni che sono presenti nella richiesta ricevuta dal soggetto richiedente.

Il livello di autorizzazione implementa le regole di policy per consentire l'accesso ai servizi applicativi di back-end.

Da un punto di vista delle politiche di sicurezza, la fase di autorizzazione, intesa come attribuzione, sospensione e revoca dei profili di accesso ai soggetti, ha lo scopo di garantire il controllo degli accessi alle risorse: i profili di accesso sono predisposti in relazione alle operazioni consentite, secondo i tempi previsti, relativamente ad insiemi di dati definiti e secondo le altre modalità ritenute necessarie.

In questo contesto, il controllo degli accessi può essere effettuato a partire dall'associazione di uno specifico ruolo a ciascun fruitore che richiede l'accesso ai servizi, secondo il modello RBAC (Role-Based Access Control). Quindi, una volta accertata l'identità del soggetto, vengono verificate le credenziali che ne attestano il ruolo (o i ruoli). Ad ogni ruolo vengono associati un insieme di diritti dalla politica di sicurezza e vengono garantiti ai soggetti che dimostrano di possedere tale ruolo.

8.5.4 Componenti previste

Dal punto di vista funzionale, il processo di gestione della sicurezza si può distinguere nelle tre fasi predette, mentre trasversalmente vi sono le attività che mirano alla federazione delle identità e alla garanzia di sicurezza delle comunicazioni che intercorrono fra tutti le componenti dell'architettura attraverso la gestione dell'integrità e del non ripudio, della confidenzialità e della privatezza.

Lo strato di sicurezza, una volta ricevuta la richiesta di sevizio, esegue le operazioni di autenticazione e di autorizzazione e provvede alla costruzione del portafoglio di asserzioni. Di conseguenza, le diverse fasi hanno anche il compito di costruire un portafoglio di asserzioni SAML 2.0 (se queste non sono del tutto complete) relative all'operatore ed eventualmente all'assistito a cui si riferisce il servizio (ciò a supporto delle politiche di autorizzazione), nonché eventualmente dei processi business della componente che sta fornendo il servizio invocato.

È quindi plausibile l'ipotesi che il portafoglio di asserzioni SAML ricevuto sia sufficiente: in tal caso, esso viene solo verificato e passato ai servizi di back-end.

Questo scenario si riconosce per la struttura dell'header di sicurezza che contiene le asserzioni SAML.

Questa struttura del messaggio è normalmente quella ricevuta da un altro dominio in cooperazione applicativa, ma può anche provenire dallo stesso dominio da parte di un fruitore/client che ha provveduto autonomamente a costruire il portafoglio di asserzioni.

Il *Gestore delle Politiche di Accesso* di InFSE è una componente federata e prevede una serie di componenti.

Le componenti presenti a livello di Service Provider sono descritti di seguito.

- Componente di autenticazione: stabilisce un context trusted environment tra fuitore ed erogatore;
- Componente di identificazione federata (Federation Gateway in SPC, Local Proxy in ICAR Task-INF3): è la componente che, dal punto di vista del Service Provider, si comporta da proxy verso l'infrastruttura di autenticazione federata.
- Policy Enforcement Point (PEP): si interpone tra il fruitore ed il servizio, intercetta tutte le richieste e garantisce il soddisfacimento del requisito di autorizzazione, interagendo con il PDP.
- Policy Decision Point (PDP): è la componente che effettivamente esegue il processo decisionale valutando le politiche di sicurezza e le richieste di accesso e decide se un accesso deve essere consentito oppure negato.

Le componenti "esterne" al Service Provider, principalmente accedute dalla componente di identificazione federata, sono:

- Identity Provider (IdP): è l'entità incaricata della gestione delle regole di identificazione. Si occupa dell'identificazione ed è innanzitutto un Security Token Service con il compito di generare, validare e rinnovare i token di sicurezza; si occupa altresì di interrogare gli AA per ottenere tutti gli attributi da inserire nella asserzione.

- Profile Authority (PA): è l'entità incaricata della gestione e manutenzione dei profili utente e può essere interrogata anche remotamente; il profilo è composto da n-ple strutturate, ad esempio, nel seguente modo: Nome Attributo, Valore Attributo, Riferimento logico dell'Authority in grado di validare l'attributo.

- Attribute Authority (AA): è l'entità atta a gestire le informazioni sugli operatori, gli assistiti e le loro correlazioni temporali; si identifica con i basamenti e viene interrogata mediante un'operazione di AttributeQuery; sulla base degli attributi necessari per la fase di autorizzazione presenti nell'elemento AttributeConsumingService dei propri metadati, fornisce come risposta una asserzione SAML con i valori degli attributi richiesti.

- Certificate Authority (CA): questa entità ha lo scopo di gestire i certificati.

Le componenti precedenti vengono indicate come "entità validatrici". Ulteriori componenti "esterne" al Service Provider, necessarie al funzionamento delle componenti precedenti, sono:

- Authority Registry;

- Attribute Authority Registry;

- Authority Registry Service;

- Attribute Authority Registry Service;

- Policy Administration Point, per la conservazione delle policy.

Per quanto riguarda l'esecuzione del processo di autorizzazione si rende necesario ricorrere alle due componenti Policy Enforcement Point (PEP) e Policy Decision Point (PDP), come descritto di seguito ed illustrato in Figura 59.

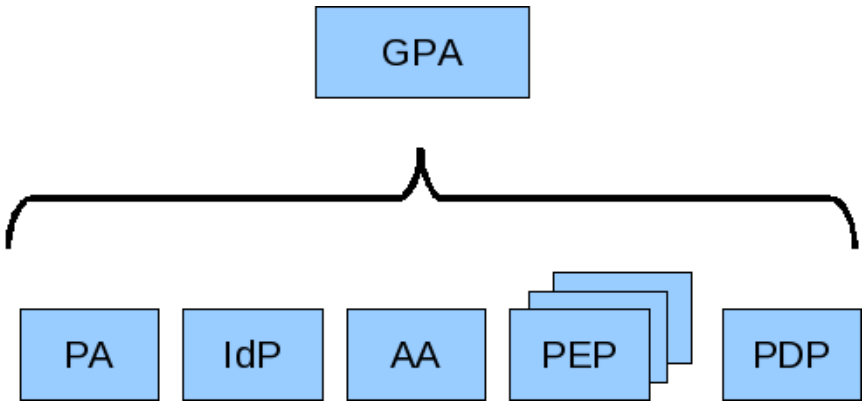


Figura 59. Componenti del Gestore delle Politiche di Accesso

- *Policy Enforcement Point* (PEP): si occupa di "intercettare" le richieste di accesso rilevanti dal punto di vista della sicurezza della risorsa a cui il PEP è associato. Una volta intercettata la richiesta, l'accesso viene sospeso ed il PEP invoca il PDP per compiere il processo decisionale e determinare se l'accesso è autorizzato oppure no. Il PEP estrae dalla richiesta tutti i dati relativi al soggetto richiedente ed all'operazione di accesso richiesta, li inoltra al PDP, ed aspetta l'esito della valutazione della politica di sicurezza. Tra questi dati ci sono le credenziali contenenti le asserzioni che attestano gli attributi del soggetto richiedente. Il PEP si occupa di verificare la veridicità di queste credenziali prima di inoltrarle al PDP. Se la risposta del PDP è positiva, l'esecuzione dell'operazione di accesso è consentita, ed il PEP la esegue. Altrimenti, se la risposta del PDP è negativa, l'operazione richiesta è vietata, ed il PEP non la esegue. Per ogni risorsa da proteggere è necessario inserire un PEP che sia in grado di intercettare le richieste di accesso indirizzate a quella risorsa. Quindi, nell'architettura di un nodo regionale, è necessario prevedere almeno un PEP per le interfacce di ogni componente InFSE. Le tecniche che è possibile adottare per l'integrazione dei PEP con le componenti dell'Infrastruttura del FSE sono descritte di seguito.

- *Policy Decision Point* (PDP): è la componente dell'architettura che esegue il processo decisionale per valutare se un dato soggetto ha il diritto di compiere una operazione di accesso ad una risorsa. Per eseguire il processo decisionale, il PDP valuta la politica di sicurezza utilizzando i dati relativi all'operazione di accesso richiesta e al soggetto richiedente (portafoglio di asserzioni), i quali sono stati propagati dal PEP. Il risultato del processo decisionale deve essere: "operazione permessa" oppure "operazione vietata". Un PDP può servire uno o più PEP. Nell'Infrastruttura del FSE, in ogni nodo regionale è presente un PDP, che applica la politica di sicurezza definita nella specifica Regione e che viene utilizzato da tutti i PEP associati alle componenti di quel nodo. La politica di sicurezza esprime i diritti di accesso che la speciica Regione assegna ad ogni ruolo per le componenti di quel nodo, e viene espressa tramite un apposito linguaggio, come ad esempio XACML.

I PEP ed il PDP dello stesso nodo regionale interagiscono tra loro tramite messaggi per eseguire il processo di autorizzazione. Le interazioni tra PEP e PDP possono essere implementate sfruttando lo standard SAML versione 2.0. In questo caso, viene utilizzato il costrutto <*AuthzDecisionQuery*> per formulare il messaggio di richiesta di autorizzazione che il PEP invia al PDP. In risposta, il PDP invia una asserzione SAML di autorizzazione, *Authorization Decision Assertion*, che contiene l'elemento <*AuthzDecisionStatement*>, il quale specifica l'esito del processo di autorizzazione (ad es., permit/deny/indeterminate). In alternativa, SAML 2.0 Profile of XACML v2.0 descrive come i protocolli e le asserzioni definite dallo standard SAML possono essere estesi per interfacciare il PEP con un PDP basato su XACML. Ad esempio, il costrutto <*AuthzDecisionQuery*> viene ridefinito dal costrutto <*XACMLAuthzDecisionQuery*> e viene utilizzato dal PEP per richiedere l'autorizzazione ad un PDP XACML. Similmente, il nuovo costrutto <*XACMLAuthzDecisionStatement*>, che estende il costrutto <*AuthzDecisionStatement*>, viene utilizzato dal PDP per comunicare al PEP il risultato del processo di autorizzazione. I nuovi costrutti definiti in SAML 2.0 Profile of XACML v2.0 estendono gli originali definiti dallo standard SAML, permettendo di trasmettere dati supplementari, come ad esempio XACML Request Context nel caso di <*XACMLAuthzDecisionQuery*> e XACML Response Context nel caso di <*XACMLAuthzDecisionStatement*>. Request Context e Response Context hanno lo scopo di contenere attributi XACML.

8.5.5 Integrazione dei PEP con le componenti di InFSE

Come appena descritto, ciascuna componente dell'architettura InFSE deve essere protetta da un sistema di autorizzazione che controlla il diritto del soggetto di eseguire l'operazione richiesta. Ciascuna componente dell'architettura delega al Policy Decision Point del nodo regionale a cui appartiene il compito di controllare i diritti di accesso dei soggetti richiedenti l'accesso. A tale scopo, le componenti dell'Infrastruttura del FSE devono integrare i Policy Enforcement Point, che contattano il PDP per verificare i diritti di esecuzione di ogni richiesta di accesso ricevuta dalla componente.

L'integrazione del PEP nelle componenti dell'Infrastruttura InFSE può avvenire attraverso differenti tecniche, tra cui:

- Componenti predisposte;
- Web Service Handler Chain;
- Wrappers.

Un esempio della prima soluzione, cioè la configurazione di componenti già predisposte, è dato dallo standard OASIS ebXML Registry 3.0 [7], che può essere adottato per l'implementazione dei Registri Regionali. Tale standard risulta essere predisposto per l'integrazione di un PEP ed un PDP XACML per la gestione delle politiche di accesso.

Se invece la componente da integrare nell'Infrastruttura del FSE è implementata

come servizio Web, può essere facilmente adottata la seconda soluzione per l'integrazione del PEP. Infatti, il motore che gestisce i servizi Web permette di ispezionare ed eventualmente modificare le richieste di servizio SOAP ricevute prima di avviare il servizio Web richiesto. Questo meccanismo si chiama Web Service Handler Chain. Un Web Service Handler è una componente che riceve in ingresso la richiesta di servizio Web, la elabora, e restituisce come risultato la richiesta rielaborata. Il primo handler della catena intercetta la richiesta ricevuta dal soggetto, la elabora e la restituisce al secondo handler della catena. Gli handler intermedi della catena ricevono la richiesta dall'handler precedente e la restituiscono all'handler successivo. L'ultimo handler invoca il servizio Web. Per l'integrazione della componente nell'Infrastruttura del FSE è possibile aggiungere alla catena di handler già presenti un nuovo handler che si occupa di invocare il Policy Decision Point al fine di richiedere la valutazione della politica di sicurezza per la richiesta in ingresso, e che esegue effettivamente la richiesta solo se riceve l'autorizzazione dal PDP.

Per l'integrazione del PEP nel caso di sistemi legacy di gestione sanitaria regionale nell'Infrastruttura del FSE (ogni struttura ha un proprio sistema personalizzato per la gestione dei documenti), la soluzione basata sull'utilizzo di wrappers è quella più idonea. Il wrapper non è altro che un servizio Web sviluppato ad hoc che sostituisce la componente stessa. Il wrapper riceve la richiesta di servizio in luogo del sistema legacy, effettua la fase di autorizzazione, e se la richiesta è autorizzata contatta la componente originale usando il protocollo necessario per l'esecuzione dell'operazione richiesta.

8.5.6 Politiche per il controllo degli accessi

La fase di autorizzazione valuta la politica per il controllo degli accessi del nodo regionale per verificare i diritti di accesso del soggetto richiedente sulle componenti dell'Infrastruttura. Nel caso dell'Infrastruttura del FSE, la fase di autorizzazione è molto importante, perché non tutti i soggetti aventi diritto all'accesso al Fascicolo Sanitario Elettronico hanno in realtà gli stessi diritti sui dati; un corretto controllo di accesso può ad esempio restringere l'accesso ad informazioni private o non consentirne modifiche.

Nel contesto di riferimento, il controllo sugli accessi viene effettuato sulla base del ruolo del soggetto che richiede l'accesso, secondo il modello RBAC. Ogni soggetto può possedere più ruoli, ma quando formula la richiesta di accesso tramite l'*Interfaccia di Accesso*, deve scegliere quali dei suoi ruoli utilizzare. Quindi, all'atto della preparazione della richiesta di accesso, una volta accertata l'identità del soggetto, vengono prodotte le credenziali che attestano il ruolo (o i ruoli) che il soggetto possiede da parte delle autorità che garantiscono tali ruoli al soggetto. Nella fase di autorizzazione, le asserzioni vengono estratte dalle richieste, verificate ed utilizzate per la valutazione delle politiche di sicurezza per la determinazione del relativo set di diritti di accesso.

Se, ad esempio, il soggetto che richiede l'accesso ai dati sanitari è il paziente stesso, nessun certificato di attributo viene incluso nella richiesta. Al contrario, se il soggetto è un medico, questo seleziona il profilo con cui accedere al servizio, in modo da

allegare alla richiesta le asserzioni attestanti gli attributi associati al profilo scelto. Ad esempio, il profilo di Medico di Medicina Generale può essere certificato dall'asserzione che attesta il ruolo di "Medico di Medicina Generale", emessa dall'ASL dove il medico presta servizio.

I moderni sistemi di autorizzazione sono basati su politiche di sicurezza che vengono espresse tramite appositi linguaggi, i quali permettono di definire condizioni più o meno complesse che tengono conto di vari fattori che devono essere valutati per decidere l'accesso. Più il linguaggio utilizzato per esprimere la politica di sicurezza è espressivo, più facile è esprimere con tale linguaggio una politica di autorizzazione che modelli i requisiti di sicurezza del sistema da proteggere.

La soluzione più adeguata ai requisiti dello scenario in studio, perciò, è quella di definire le politiche di accesso ai dati memorizzati dall'Infrastruttura del FSE tramite un linguaggio per politiche di sicurezza flessibile ed espressivo, come l'eXtensible Access Control Markup Language (XACML).

8.5.7 Standard XACML

eXtensible Access Control Markup Language (XACML) [15] è uno standard definito dal consorzio OASIS basato su XML, che definisce un linguaggio per esprimere politiche di sicurezza per il controllo degli accessi. L'ultima versione di XACML disponibile è la 3.0, sviluppata nel 2009, anche se attualmente la versione più usata dello standard è la 2.0. Lo standard XACML definisce sia un linguaggio che una architettura di riferimento. Inoltre, SAML 2.0 Profile of XACML v2.0 definisce i costrutti per il protocollo di comunicazione tra PEP e PDP XACML.

Lo standard XACML definisce una architettura di riferimento per l'implementazione di un sistema di autorizzazione:

- Policy Enforcement Point (PEP): effettua il controllo sugli accessi, intercettando le richieste di accesso alle risorse, inoltrando le relative richieste di decisione al PDP. Il PEP può includere nella richiesta di decisione gli attributi del soggetto, della risorsa e dell'ambiente. Una volta eseguito il processo decisionale da parte del PDP, il PEP si occupa di fare rispettare le decisioni di autorizzazione, eseguendo la richiesta di accesso solo se la decisione è positiva, ed occupandosi dell'esecuzione di eventuali obbligazioni richieste dal PDP;
- Policy Information Point (PIP): ha la funzione di reperire gli attributi del soggetto, della risorsa e dell'ambiente;
- Policy Decision Point (PDP): è il motore decisionale del sistema, che valuta le policy applicabili e produce la decisione di autorizzazione per l'esecuzione dell'azione sulla risorsa richiesta. La valutazione della policy può richiedere gli attributi del soggetto, della risorsa o dell'ambiente;
- Policy Administration Point (PAP): si occupa della produzione delle policy e dei gruppi di policy (policy set), che vengono conservate in un policy repository e vengono passate al PDP per la valutazione.

Le interazioni tra le componenti dell'architettura di riferimento di XACML in caso di una richiesta di accesso sono descritte di seguito:

- Il soggetto (access requester) esegue una richiesta per l'accesso ad una risorsa.
- Il PEP intercetta la richiesta prima che venga servita.
- Il PEP inoltra la richiesta al Context Handler.
- Il Context Handler traduce la richiesta dal formalismo del particolare dominio in una richiesta XACML e la inoltra al PDP.
- Il PDP esegue gli algoritmi di decisione che valutano le politiche precedentemente caricate tramite il PAP.
- Il PDP può contattare il Context Handler per richiedere il valore di alcuni attributi del soggetto, della risorsa o di ambiente necessari alla valutazione della politica.
- Il Context Handler contatta i Policy Information Point (PIP) per ottenere tutti gli attributi mancanti.
- Il PIP restituisce gli attributi raccolti al Context Handler
- Il Context Handler riporta gli attributi al PDP che può continuare la valutazione della policy per produrre una decisione.
- Il PDP restituisce la decisione al Context Handler.
- Il Context Handler restituisce la decisione al PEP che esegue l'accesso richiesto solo se la decisione è positiva.
- Qualora la decisione contenga delle obbligazioni, queste vengono passate dal PEP all'Obligation Service che si occupa della loro esecuzione.

Le interazioni tra le componenti descritte sono mostrate in Figura 60.

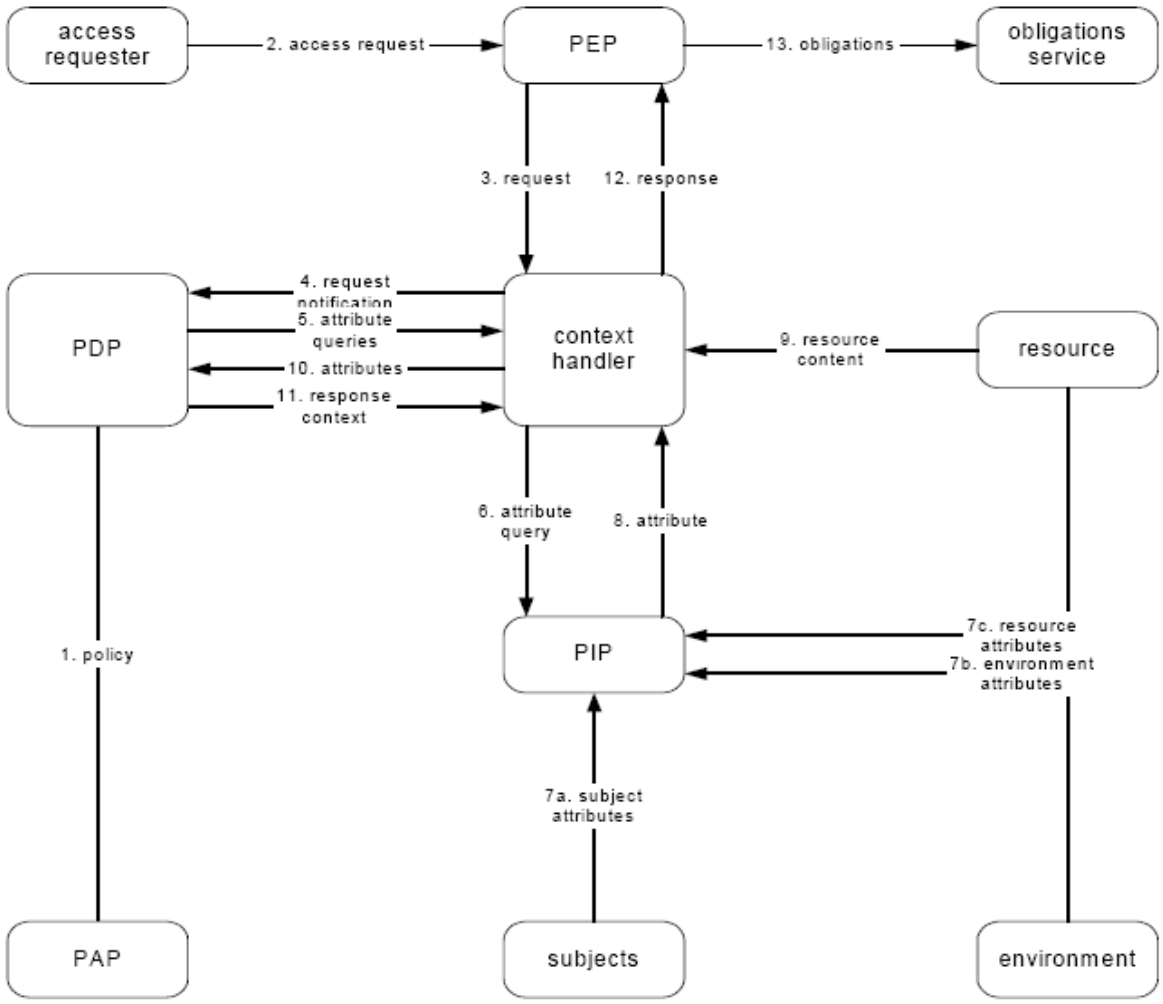


Figura 60. Architettura di riferimento XACML

Il SAML 2.0 Profile of XACML v2.0 [14] estende il protocollo di richiesta/risposta definito da SAML 2.0 per supportare le comunicazioni con un PDP XACML. In particolare, alcuni dei costrutti dallo standard SAML vengono estesi dal SAML 2.0 Profile of XACML v2.0 per supportare lo scambio di informazioni addizionali. I costrutti estesi definiti da questo standard sono i seguenti:

- <*XACMLAuthzDecisionQuery*>: viene utilizzato dal PEP per richiedere ad un PDP XACML l'esecuzione della procedura di autorizzazione per una certa richiesta d'accesso. Rispetto al costrutto SAML <*AuthzDecisionQuery*> che estende, esso permette di trasmettere il Request Context che contiene attributi XACML;
- <*XACMLAuthzDecisionStatement*>: estende il costrutto SAML <*AuthzDecisionStatement*> e viene utilizzato dal PDP per comunicare al PEP il risultato del processo di autorizzazione, includendo il Response Context;
- <*XACMLPolicyQuery*>: viene utilizzato per richiedere una o più policy XACML al

Policy Administration Point;

- *<XACMLPolicyStatement>*: viene utilizzato dal PAP in una SAML Response per rappresentare una o più policy XACML, oppure può essere utilizzato in una asserzione SAML come formato per immagazzinare le policy in un Policy Repository.

XACML permette di definire, in maniera dichiarativa, le politiche di autorizzazione che si intendono applicare in un particolare dominio: esso permette di definire un insieme di regole riguardanti una risorsa, e fornisce un set di algoritmi per combinarne il risultato; al momento di decidere se concedere l'accesso, queste regole vengono valutate ed il loro risultato deve essere combinato secondo l'algoritmo scelto. Ad esempio, se l'algoritmo scelto è "Deny overrides" ("il rifiuto ha la precedenza"), basta che una sola delle regole applicabili alla richiesta di accesso dia risultato negativo (accesso rifiutato), che l'accesso viene rifiutato, anche se tutte le altre regole applicabili hanno dato risultato positivo (cioè avrebbero consentito l'accesso).

Inoltre, XACML supporta in modo nativo la gestione degli attributi dei soggetti, delle risorse ed ambientali. Infatti, le regole di una politica possono anche includere la valutazione di attributi, come ad esempio il ruolo del soggetto, e quindi risulta naturale definire le politiche di accesso necessarie nel contesto dell'Infrastruttura del FSE, dove è necessario assegnare diritti diversi sulla stessa risorsa a seconda del ruolo del soggetto.

Gli elementi costituenti del linguaggio XACML, mostrati in Figura 61, sono i seguenti:

- *Policy (<Policy>)*: rappresenta le politiche di accesso, che sono costituite da un target, un insieme di regole (rule), un algoritmo per la combinazione di regole ed un insieme di obbligazioni. Gli algoritmi di combinazione sono necessari nelle politiche che contengono più regole, perché la loro valutazione potrebbe restituire risultati contrastanti, ed è necessario combinare tali risultati per produrre un'unica decisione finale.
- *Target (<Target>)*: definisce il soggetto a cui si applica la policy o la regola di cui il target fa parte, la risorsa e l'azione su tale risorsa. Tali condizioni servono per definire se la policy o la regola è applicabile alla richiesta di accesso corrente. Se tutte le condizioni del target sono soddisfatte, allora viene applicata la rispettiva policy o regola che decide sull'accesso alla risorsa.
- *Rule (<Rule>)*: rappresenta le regole elementari contenute nelle politiche, che sono costituite da un target, un effetto (o decisione) e una condizione. Il target, come indicato, indica le entità alla quale la regola deve essere applicata. Ad esempio, una regola può avere come target un determinato soggetto ed una determinata risorsa, e quindi si applica agli accessi di quel soggetto a quella risorsa. L'effetto di una regola (attributo <*Effect*>) deve essere specificato obbligatoriamente, ed indica la decisione associata alla regola quando essa è verificata (ad es., Permit, Deny). Le condizioni di una regola (elemento <*Condition*>, figlio dell'elemento <*Rule*>), invece, sono facoltative e sono espressioni booleane che restringono l'applicabilità della regola. Le condizioni possono riguardare la valutazione di attributi del

soggetto, della risorsa o dell'ambiente. Se il risultato della valutazione degli attributi è positivo, allora la regola è applicabile e la decisione che ne consegue è data dal valore del campo effetto. Altrimenti la regola risulta non applicabile, e non contribuisce alla decisione.

- *Policy Set (<PolicySet>)*: è costituito da un target, un insieme di politiche, un algoritmo di combinazione delle decisioni prodotte dalle politiche ed un insieme di obbligazioni. Ad un livello superiore viene ripetuto ciò che accade per le regole all'interno di una policy.
- *Algoritmi di Combinazione*: hanno lo scopo di combinare i risultati ottenuti dalle regole applicabili di una policy per ottenere la decisione finale della policy stessa, ed i risultati delle varie policy per ottenere la decisione del *Policy Set*.

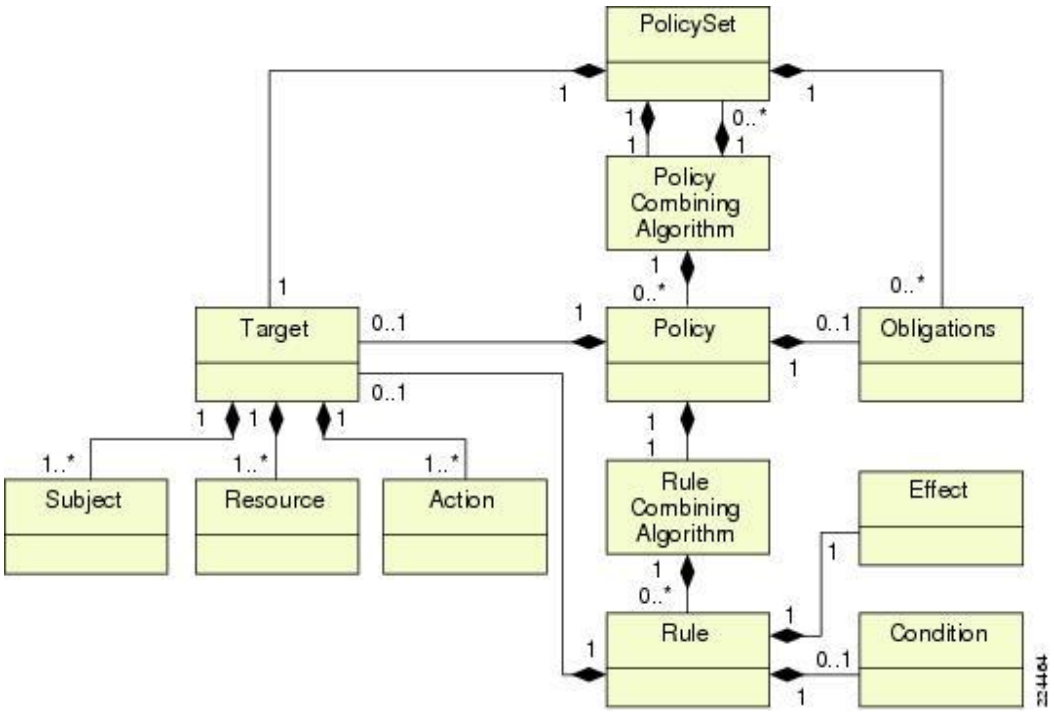


Figura 61. Modello del linguaggio XACML

Lo standard XACML ha diverse caratteristiche che ne rendono vantaggioso l'utilizzo per implementare il sistema di autorizzazione nell'Infrastruttura InFSE.

Prima di tutto, XACML supporta in modo nativo la gestione degli attributi dei soggetti, perché consente di inserire nelle politiche di accesso delle condizioni che valutano gli attributi dei soggetti per definire l'applicabilità della regola. Dato che nel contesto dell'Infrastruttura InFSE è necessario assegnare diritti diversi sulla stessa risorsa a seconda del ruolo del soggetto e ad altri attributi che eventualmente possiede, l'adozione di XACML è sicuramente vantaggiosa, in quanto permette di esprimere queste policy nativamente.

Dal punto di vista architetturale, il modello di riferimento di XACML può venire facilmente adottato nell'Infrastruttura seppur con qualche semplificazione. Ad esempio, il PEP ed il Context Handler potrebbero essere fusi in un'unica componente, che potrebbe adottare SAML 2.0 Profile of XACML v2.0 per la comunicazione con il PDP.

8.5.8 Adattamento al dominio InFSE

8.5.8.1 Context Handler

Il Context Handler è la componente che orchestra il flusso XACML e traduce la richiesta dal particolare dominio applicativo in una sintassi XACML.

Ad ogni policy di dominio è necessario fornire il corretto Context Handler. Infatti questa componente è dipendente dal dominio applicativo e non può essere generalizzata. Per poter realizzare un Context Handler occorre estendere la classe *AbstractContextHandler* e fornire l'implementazione del metodo:

public abstract RequestCtx getRequestCtx(Object request)

Questo metodo ha l'obbligo di effettuare la traduzione tra il formalismo del domino applicativo ed il formalismo XACML.

8.5.8.2 Attribute Finder

Si possono implementare diversi *Attribute Finder* (PIP), ad esempio per interrogare dei servizi di Attribute Authority, oppure per ottenere attributi applicando delle espressioni XPath ad un documento XML.

Per creare un nuovo *Attribute Finder* è necessario estendere la classe *AttributeFinderModule* ed implementare il metodo:

public EvaluationResult findAttribute(EvaluationCtx context)

Questo metodo deve restituire gli attributi richiesti.

8.5.8.3 Policy Finder

I *Policy Finder* sono gli oggetti deputati a ricavare le policy che si applicano sulle singole richieste. I policy finder sono basati sul profilo XACML Role-Based Access Control.

La componente PAP rappresenta il repository delle policy XACML e ne permette l'amministrazione.

8.5.9 Metodo di autorizzazione basato sul ruolo

La ricomposizione delle informazioni sanitarie e socio-sanitarie distribuite è un processo di conoscenza che prevede numerosi attori, appartenenti a varie organizzazioni, che creano "valore" partecipando alle diverse fasi del processo. Se si considerano i numerosi attori e le regole implicate legate al contesto per ciascuno, l'insieme di diritti di accesso per ogni singola combinazione attore-contesto è considerevole.

Il metodo di autorizzazione è un elemento chiave per fornire agli stakeholder partecipanti a questa ricomposizione processi dominio-dominio sicuri che ispirino fiducia reciproca (trust), ovvero la fiducia che gli attori attivi e/o consumatori ripongono nei confronti di altri attori.

Il metodo di autorizzazione e controllo degli accessi ai dati e alle risorse basato sul ruolo, come alternativa ai tradizionali metodi di controllo degli accessi, associa dei *Ruoli* a ogni attore che abbia bisogno di interagire con il sistema. Ogni ruolo definisce una determinata serie di diritti di base che l'attore di quel ruolo può esercitare, ed in genere i ruoli vengono assegnati sulla base delle responsabilità inerenti alle mansioni svolte nell'organizzazione.

Dalle considerazioni precedenti, nel caso dell'Infrastruttura InFSE, tale approccio deve essere affiancato da una definizione rigorosa dei *ruoli*, del *contesto* della richiesta e delle *regole* da applicare, perché l'esigenza è che le autorizzazioni siano applicate in maniera coerente, non necessariamente uniforme, in ogni dominio e/o contesto dei vari sistemi infrastrutturali locali sottesi da InFSE.

8.5.9.1 Ruoli: prima parte delle credenziali di autorizzazione

I ruoli previsti in InFSE sono riportati nella figura seguente.

La prima parte delle credenziali di autorizzazione (ruolo) è assegnata sulla base dell'identità dell'attore e di altre caratteristiche di identità (ad es., organizzazione di accesso). Il ruolo è assegnato da entità certificate e certificatrici dell'attributo richiesto.

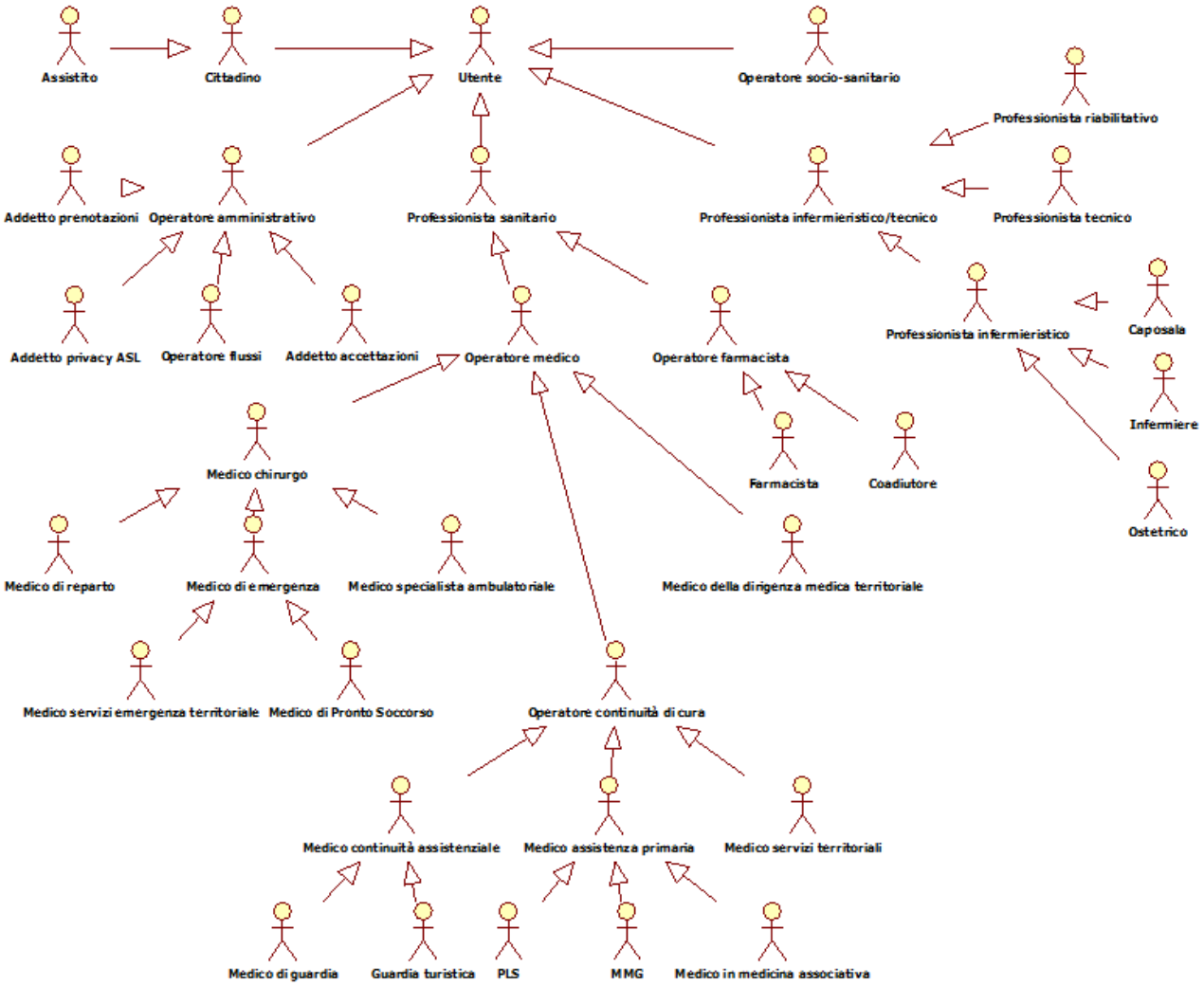


Figura 62. Ruoli previsti in InFSE

Nella tabella seguente è riportata una prima descrizione.

Ruolo	Acronimo	Descrizione
Utente	UTE	Utente generico, ruolo padre astratto
Cittadino	CIT	Utente riconosciuto da una anagrafe demografica (ad es., INA-SAIA)
Assistito	ASS	Utente iscritto all'anagrafe assistibili
Operatore socio-sanitario	OSS	Operatore generico secondo l'Accordo Stato-Regioni del 22/02/2001
Operatore amministrativo	AMM	Operatore operante nelle strutture socio-sanitarie con mansioni amministrative
Addetto prenotazioni	APR	Operatori incaricati di ricevere e gestire le richieste di prenotazione delle prestazioni socio-sanitarie

Addetto accettazioni	ADT	Operatori incaricati di gestire l'accettazione delle prestazioni ambulatoriali o l'intero processo di accettazione, degenza e dimissione degli assistiti nelle strutture di cura
Addetto privacy ASL	APV	Soggetto operante nella ASL con l'incarico di gestione del consenso, nonché gestione e trattamento della privacy
Operatore flussi	OFL	Operatore amministrativo per flussi erogati (Regione, ASL, MEF, MdS, INPS)
Professionista infermieristico/tecnico	PIT	Professionista infermieristico/tecnico generico, ruolo padre astratto
Professionista tecnico	PTC	Professionista tecnico (area tecnico/diagnostica, area tecnico/assistenziale, etc.)
Professionista riabilitativo	PRI	Professione area sanitaria-riabilitativa
Professionista infermieristico	PIN	Operatore paramedico generico, ruolo padre astratto
Infermiere	INF	Infermiere addetto alla assistenza (anche infermieri pediatrici)
Caposala	CAP	Caposala addetto al controllo dei servizi di reparto
Ostetrico	OST	Ostetrico
Professionista sanitario	PSA	Operatore che svolge attività di prevenzione, assistenza, cura o riabilitazione
Operatore farmacista	OFR	Operatore farmacista generico, ruolo padre astratto
Farmacista	FAR	Operatore addetto alla farmacia
Coadiutore	COA	Soggetto operante nella farmacia in collaborazione con il titolare
Operatore medico	OMD	Operatore medico generico, ruolo padre astratto
Medico chirurgo	MCH	Medico specialista abilitato alla erogazione di prestazioni sanitarie ospedaliere in regime di ricovero, presso strutture pubbliche
Medico di reparto	MRP	Medico di reparto ospedaliero
Medico specialista ambulatoriale	MSA	Medico specialista abilitato alla erogazione di prestazioni sanitarie specialistiche erogate in regime ambulatoriale (ambulatori, poliambulatori, consultori familiari o strutture di erogazione specialistiche private accreditate)
Medico di emergenza	MEM	Medico adibito all'emergenza
Medico di Pronto Soccorso	MPS	Medico del servizio di emergenza ospedaliera (Pronto Soccorso)
Medico servizi emergenza territoriale	MET	Medico del servizio di emergenza sanitaria (servizio 118)
Medici della dirigenza medica territoriale	MDT	Medici abilitati alle funzioni dirigenziali
Operatore continuità di cura	OCC	Operatori sanitari generici che assicurano la continuità di cura, ruolo padre astratto
Medico continuità assistenziale	MCA	Medico convenzionato preposto alle attività di continuità assistenziale.
Medico servizi territoriali	MST	Medico convenzionato preposto alle attività dei servizi territoriali

Medico assistenza primaria	MAP	Medico convenzionato per l'erogazione dell'assistenza primaria che, ai fini della scelta del medico da parte del cittadino, appartiene ad uno specifico ambito territoriale (comuni, gruppi di comuni, distretti)
Medico di Medicina Generale	MMG	Medico di famiglia
Pediatra di Libera Scelta	PLS	Medico di Assistenza Primaria con specializzazione in Pediatria
Medico in medicina associativa	MMA	Medico che opera in forma di medicina associativa o di medicina di rete con altri MMG/PLS
Medico di guardia	GMD	Medico per la continuità assistenziale nelle guardie mediche
Guardia turistica	GTU	Medico per la continuità assistenziale nelle guardie turistiche

Tabella 1. Descrizione dei ruoli previsti in InFSE

Nella risoluzione delle policy, a tali ruoli viene applicato un assioma di propagazione, rispetto alla gerarchia dei ruoli, ovvero la verifica positiva di una policy per un ruolo a un livello della gerarchia è condizione sufficiente per validare l'attivazione positiva della policy per tutti i ruoli che sono nella gerarchia di nodi figli del ruolo per cui la policy stessa è definita (ovviamente a condizione di identico contesto).

Per esempio, la policy che stabilisce che un utente nel ruolo "Operatore amministrativo" può accedere a qualsiasi documento se egli è l'autore del documento richiesto, è valida e viene automaticamente applicata per ogni ruolo nella gerarchia di cui "Operatore amministrativo" è il ruolo padre (ad es., "Addetto prenotazioni"). Tale assioma è conseguenza della semantica di inclusione nella gerarchia dei ruoli, come mostrato in figura, per la quale ogni utente classificato come appartenente al ruolo X è anche classificato (as-is) come appartenente al ruolo padre di X.

8.5.9.2 Proprietà dei ruoli

Per ogni ruolo è importante definire proprietà opzionali legate ai seguenti aspetti:

- Area di default: indica dove un attore svolge di default le sue funzioni (ospedale, studio medico);
- Ruoli in conflitto: indica l'eventuale ruolo con cui si entrerebbe in conflitto, nel caso di una allocazione dinamica, ma anche nell'ambito di porte di accesso che consentono il rilascio delle credenziali attraverso una scelta manuale dei ruoli da una lista predefinita;
- Limiti di tempo: indica l'eventuale periodo temporale limite in cui il ruolo definito è valido.

8.5.9.3 Contesto: seconda parte delle credenziali di autorizzazione

Il controllo del contesto è estremamente importante per determinare la seconda parte delle credenziali di autorizzazione (contesto), ed in InFSE è fondamentale che gli operatori siano autorizzati sulla base di tali dati, tratti da diverse fonti, in modo da svolgere controlli sulla coerenza della richiesta. Ciò coinvolge attributi legati alla richiesta (come il tipo di documento acceduto, l'assistito relativo, momento temporale, etc.), ma anche al dominio di appartenenza (ad es., Regione) e/o di localizzazione (ad es., sito o reparto), che potrebbero, soprattutto nell'ultimo caso, modificare dinamicamente il ruolo assegnato (ad es., da "Medico di Medicina Generale" a "Medico di Medicina Generale di altra Regione").

L'ultimo aspetto è legato al fatto che i diritti di accesso potrebbero "scalare", ovvero essere ridotti, se l'attore non opera nel suo dominio di default. Peraltro, in questo caso, potrebbe essere necessario richiedere al dominio di origine ulteriori attibuti dell'attore per determinare i suoi privilegi nel dominio erogatore, differentemente da quelli necessari per un attore con un ruolo analogo, ma appartenente al domino erogatore. Di conseguenza, la determinazione delle informazioni di contesto ha il duplice scopo di:

1. completare gli elementi per prendere la decisione finale di autorizzazione;
2. evolvere un ruolo statico basato sulle credenziali ad una allocazione dinamica determinata durante la sessione.

È possibile distinguere, quindi, per un particolare attore ruoli statici e ruoli dinamici.

- I primi, come già indicato, sono definiti all'atto del censimento dell'attore nelle anagrafiche (registrazione) e vengono assegnati sulla base delle credenziali dello stesso.
- I secondi sono definiti dal contesto e determinano quale sottoinsieme può essere attivato durante la sessione.

Per quanto rigarda il contesto di confidenza dei documenti, la normativa vigente per la composizione dei documenti in formato HL7-CDA Rel. 2.0 impone che il livello di riservatezza da applicare ai documenti sia espresso tramite i valori che compongono il sistema di codifica internazionale "Confidentiality". Per le applicazioni nel realm italiano, l'accesso ai documenti registrati è consentito, in accordo con il rilascio del consenso generale alla fruizione dello stesso FSE, di default al paziente (per i propri documenti) ed all'autore (per i propri documenti). Si riportano i possibili valori della codifica "Confidentiality":

- V (Very restricted): l'accesso è regolato da precise norme secondo quanto disposto dal Garante della Privacy in accordo con il consenso espresso; nel realm italiano questo coincide con il consentire l'accesso al solo paziente ed al solo autore del documento;
- R (Restricted): l'accesso è consentito solo a chi ha attualmente una relazione di

cura con il paziente; una relazione di cura può essere la relazione che lega l'operatore medico MMG/PLS al paziente mediante scelta attiva (ossia l'operatore medico MMG/PLS è il medico di famiglia del paziente), ovvero la relazione con un operatore sanitario che ha in cura un paziente per necessità legate alla sua patologia (ad esempio, un operatore Medico di Reparto appartenente al reparto di ricovero dell'assistito);

- N (Normal): è il livello di riservatezza normale, ovvero possono accedere ai documenti solo coloro che sono autorizzati secondo le regole definite per l'accesso.

8.5.9.4 Modalità di autorizzazione ed elementi delle regole

Una volta associata una sessione all'attore con un determinato ruolo (dopo la fase di autenticazione e di identificazione), viene determinato il contesto d'uso ed eventualmente ulteriori informazioni da entità distribuite in rete per questo compito, in modo da determinare:

- l'autorizzazione all'accesso al servizio;
- l'autorizzazione all'esecuzione dell'operazione richiesta;
- l'autorizzazione al trattamento del gruppo di dati correlati all'operazione.

Infine, si illustrano in maniera sintetica gli elementi necessari per la determinazione delle regole che disciplinano l'accesso degli operatori ai documenti indicizzati:

1. Tipo di documento (ad es., Prescrizione Farmaceutica, Patient Summary);
2. Livello di confidenza del documento;
3. Autore del documento;
4. Relazione esistente tra l'attore e l'assistito;
5. Relazione esistente tra l'autore del documento e l'assistito;
6. Relazione tra l'attore e l'autore del documento (se il ruolo dell'operatore è medico sostituto o delegato rispetto al MMG che è il medico curante dell'assistito);
7. Tipo di consenso rilasciato (ad es., presenza dell'assistito, consenso al trattamento dei dati, etc.);
8. Possibilità di assunzione di responsabilità (ad es., per i medici di Pronto Soccorso); in questo caso, è possibile accedere ad una risorsa solo a seguito di registrazione di assunzione di responsabilità o, qualora il paziente disponga di una CNS/CIE, con la presenza del paziente;
9. Tipo di operazione sul documento;
10. Struttura di appartenenza (se il ruolo dell'operatore è Medico di reparto, l'assistito è ricoverato in questo momento nella struttura dell'operatore).

Bibliografia

[1] Healthcare Services Specification Project (HSSP), http://hssp.wikispaces.com/

[2] Integrating the Healthcare Enterprise (IHE), http://www.ihe.net/

[3] European Patients Smart Open Services (epSOS), http://www.epsos.eu/

[4] Sperimentazione di un sistema per l'Interoperabilità europea e nazionale delle soluzioni di fascicolo sanitario elettronico: componenti Patient Summary e ePrescription (IPSE), http://www.progettoipse.it

[5] Una politica per la Sanità Elettronica, Tavolo di lavoro permanente di Sanità Elettronica delle Regioni e delle Province Autonome, http://www.innovazionepa.gov.it/media/566294/tse-politica%20condivisa%20per%20la%20sanit%20elettronica.pdf

[6] Strategia architetturale per la Sanità Elettronica, Tavolo di lavoro permanente di Sanità Elettronica delle Regioni e delle Province Autonome, http://www.innovazionepa.gov.it/media/566290/tse-ibse-strategia_architetturale-v01.00-def.pdf

[7] InFSE – Modello informativo dei metadati

[8] ebXML Registry Services and Protocols Specification (ebRS), version 3.0.1, http://www.oasis-open.org/committees/download.php/23648/regrep-3.0.1-cd3.zip

[9] ebXML Registry Information Model (ebRIM), version 3.0.1, http://www.oasis-open.org/committees/download.php/23648/regrep-3.0.1-cd3.zip

[10] Web Services Notification (WSN), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn

[11] Sistema Pubblico di Connettività (SPC), http://www.cnipa.gov.it/site/it-it/Attivit%C3%A0/Sistema_Pubblico_di_Connettivit%C3%A0_(SPC)/

[12] Interoperabilità e Cooperazione Applicativa fra le Regioni (ICAR), http://www.progettoicar.it/ViewCategory.aspx?catid=222a18445d9d4d71af6fd2a6e43f7ef0

[13] Security Assertion Markup Language (SAML), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[14] SAML 2.0 profile of XACML 2.0 docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-profile-spec-os.pdf

[15] eXtensible Access Control Markup Language (XACML), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml