

Progetto interregionale ICAR
“Interoperabilità e Cooperazione Applicativa tra le Regioni”

Task INF-2: Gestione di Strumenti di Service
Level Agreement a livello interregionale
SPECIFICHE TECNICHE DEL SISTEMA
(INF2_SPE)

Versione 3.0

INDICE

1	Modifiche Documento.....	3
2	Introduzione	4
3	Contesto di riferimento	5
4	Presupposti al disegno architetturale	7
4.1	Definizione e registrazione dell'accordo di servizio	7
4.2	Caratteristiche dei parametri SLA.....	8
4.3	Sistemi di tracciatura e monitoraggio	11
5	Architettura generale e dominio di intervento	14
5.1	Specifica dei livelli di servizio.....	15
5.2	Sistema di tracciatura	17
5.3	Sistema di monitoraggio	19
6	Le componenti del sistema.....	20
6.1	Specifica dei livelli di servizio.....	20
6.2	Specifica del sistema di tracciatura	33
6.2.1	DBWSA – Database Web Service Agreement	33
6.2.2	Interfaccia di scrittura sul Sistema di Tracciatura.....	38
6.2.3	Interfaccia Operatore - Console di monitoraggio.....	44
6.3	Specifica del sistema di monitoraggio.....	48
6.3.1	ServiceTermState.....	49
6.3.2	ServiceGuaranteeTermState.....	53
7	Bibliografia.....	59
	Allegati.....	60

1 MODIFICHE DOCUMENTO

Descrizione Modifica	Edizione	Data
Prima redazione	1.0	4/1/2007
Riempimento dell'intero capitolo 6	1.1	30/1/2007
Varie modifiche al paragrafo 6.2	1.2	5/3/2007
Revisione generale; modifiche significative all'intero capitolo 6	2.0	8/8/2007
Revisione generale; modifiche significative al paragrafo 6.2.3	3.0	7/1/2009

2 INTRODUZIONE

Obiettivo del documento è di esplicitare le specifiche tecniche inerenti all'attività del task INF-2.

Le componenti elementari che contribuiscono al disegno complessivo della soluzione da implementare hanno una natura variegata: può trattarsi, infatti, della definizione di flussi organizzativi che governano e gestiscono il processo di cooperazione applicativa, delle regole tecniche che vincolano la partecipazione al processo medesimo, del modello architettuale dell'infrastruttura realizzata, dei formalismi adottati per l'uniformità dello SLA management e delle loro caratteristiche di espandibilità, delle infrastrutture tecnologiche che abilitano al monitoraggio dei livelli di qualità dei servizi resi disponibili.

L'individuazione e la specifica delle componenti oggetto del documento sono state influenzate da confronti con diverse impostazioni ed idee, quali quelle del Gruppo di lavoro per i Servizi di interoperabilità, cooperazione applicativa ed accesso del Sistema Pubblico di Connettività e Cooperazione in ambito CNIPA (SPCoop), le scelte architetture in ambito SLA management di web services adottate da alcune aziende software, le tecnologie e gli standard emergenti, in particolare in ambiente open source.

3 CONTESTO DI RIFERIMENTO

L'erogazione di servizi agli utenti è uno dei principi fondanti i progetti presentati nei piani di e-government.

Tale obiettivo si persegue attraverso la messa a disposizione di servizi on-line che consentono l'interazione non mediata tra l'utente e l'Amministrazione. Tale obiettivo è pienamente raggiunto solo in presenza di un efficace interscambio di dati ed eventi (che sui dati agiscono) tra le Amministrazioni.

L'interscambio e la cooperazione tra le Pubbliche Amministrazioni sono infatti prerequisito all'affidabilità e all'efficacia del risultato fornito all'utente finale, sia in termini di livello di aggiornamento delle informazioni fornite a valle di una sua richiesta, sia in termini di collaborazione dei soggetti coinvolti in un processo amministrativo per garantire un unico front end ai soggetti finali.

Tale situazione si realizza solo in presenza di una community network a livello interregionale che costituisca l'asse su cui creare l'infrastruttura organizzativa, logica e fisica a sostegno della cooperazione applicativa.

Il Sistema Pubblico di Connettività (SPC) e la sua componente relativa alla Cooperazione Applicativa, il Sistema Pubblico di Cooperazione (SPCoop), costituiscono il punto di riferimento per sviluppare ed estendere la Community Network Interregionale, in modo da renderla interoperabile anche con SPC, all'interno del quale cooperano applicazioni utilizzando servizi e accedendo a banche dati di diversi soggetti della Pubblica Amministrazione Locale.

Pur restando "opzionale" all'interno dell'attuale della specifica SPC sull'accordo di servizio, l'introduzione di strumenti comuni, in grado di effettuare un monitoraggio efficiente e costante dei livelli di servizi e delle funzionalità complessive costituisce un importante componente all'interno di SPCoop.

L'elemento atomico costituente lo scenario di riferimento è il dominio di un Ente all'interno di una rete regionale che interagisce con il dominio di servizi di altri Enti, anche appartenenti ad altre reti regionali, per attuare lo scambio di dati e servizi.

Il modello proposto si basa sulla federazione tra community network che comunicano tra loro, secondo lo scenario per la connettività delineato dal Sistema Pubblico di Connettività (SPC).

Vista l'eterogeneità dei domini di servizi presenti sulla rete e i diversi livelli di sviluppo dal punto di vista funzionale e tecnologico, la costituzione della comunità degli Enti deve avvenire attraverso il mascheramento del livello applicativo dei servizi dei singoli domini e delle specifiche modalità di interazione utilizzate all'interno delle singole comunità.

Al fine di consentire la minore invasività possibile, sia sull'assetto organizzativo dell'Ente coinvolto nel processo di interscambio, sia sul sistema informatico di back office che realizza il servizio applicativo, è stata formalizzata la presenza della porta di dominio, che intercetta e regola l'accesso ai singoli servizi del dominio, costituendo l'interfaccia tra le applicazioni richiedenti un servizio ed il dominio che tale servizio può espletare.

L'indagine si concentra invece sulle interazioni che avvengono, ad esempio, tra la componente applicativa di front end messa a disposizione da un web server ed i servizi di back end, appartenenti ad altri domini di servizi, invocati per il completamento dell'operazione.

In particolare, nel task INF-2 si trattano le questioni relative ai livelli di servizio garantiti nel corso delle interazioni tra servizi applicativi sopra delineate, al fine di definire una soluzione organizzativa e tecnologica che, implementando le funzioni di controllo della qualità e delle prestazioni dei servizi in Cooperazione Applicativa, garantisca alla comunità degli Enti un monitoraggio efficiente e costante dei livelli di servizio offerti, attraverso standard e strumenti comuni per la gestione di un sistema interregionale di Service Level Agreement.

4 PRESUPPOSTI AL DISEGNO ARCHITETTURALE

Come accennato nel capitolo 2, la progettazione della soluzione tecnica qui proposta è stata effettuata non solo in base all'analisi degli obiettivi proposti, dell'ambito di applicazione e dei vincoli presenti, ma anche sulla scorta di una duplice fase di indagine e confronto.

La prima fase ha comportato la ricerca e lo studio degli approcci, relativi allo SLA management associato all'erogazione di servizi applicativi, maggiormente analizzati ed utilizzati nell'attuale scenario IT, soprattutto in ambiente industriale e di resource sharing.

La seconda è il frutto di reiterati incontri tecnici con diversi, qualificati interlocutori: rappresentanti delle Regioni, in particolare delle capofila degli altri task infrastrutturali, e componenti del CNIPA.

Soprattutto l'interazione con CNIPA ha portato all'individuazione di alcune linee guida ulteriori rispetto a quanto già pubblicato nel gruppo di documenti "Sistema Pubblico di Cooperazione" e all'effettuazione di alcune significative variazioni progettuali.

Nell'espone le singole impostazioni architettoniche e tecnologiche daremo spazio all'illustrazione dei fattori che hanno contribuito a plasmarle.

4.1 Definizione e registrazione dell'accordo di servizio

I partecipanti al progetto sono Regioni ed altri Enti Pubblici. Tali soggetti non vanno alla ricerca di un erogatore "di mercato" in grado di offrire servizi rispondenti alle proprie esigenze e che sia dotato delle migliori caratteristiche relativamente a tempi, costi, esaustività, affidabilità e mole di risorse disponibili.

In una situazione reale, un'Amministrazione conosce il servizio del quale necessita e sa quale altro soggetto è in grado di erogarlo.

La conclusione di un accordo di servizio assume quindi il valore di stabilire una sorta di confine di responsabilità tra la Pubblica Amministrazione fruitrice e la Pubblica Amministrazione erogatrice del servizio in questione, e permette una tracciatura esplicita tra le competenze delle parti coinvolte.

La ricerca del soggetto erogatore e la formalizzazione dell'accordo di servizio alla base della collaborazione tra le due Amministrazioni avvengono quindi off-line, mediante trattative informali che si concludono con la registrazione di un accordo di servizio formale sul registro SICA.

Questa impostazione risulta perfettamente compatibile con quanto specificato al paragrafo "5.1 – Ciclo di vita dell'accordo di servizio" del documento "CNIPA, Sistema pubblico di cooperazione: accordo di servizio, 14 Ottobre 2005" e porterà all'utilizzo, nella fase di definizione dell'accordo di servizio, di un approccio di tipo "unilaterale" per la parte comune (concepita autonomamente dal soggetto erogatore)

e “concordato” per le parti specifiche (ideate in modo congiunto da soggetto erogatore e soggetto fruitore).

Essendo la registrazione dell’accordo di servizio sui registri SICA un processo che coinvolge attività di pertinenza dei Task INF-1 e INF-2, l’implementazione relativa andrà concordata tra i capofila dei task infrastrutturali in questione.

Sono già disponibili alcuni framework nati per gestire in modo formalizzato vari processi assimilabili allo SLA management.

In particolare sono stati oggetto di accurata indagine: WSLA - Web Services Level Agreement (progettato da IBM) e WS-Agreement (ideato nell’ambito del GGF Global Grid Forum), entrambi in forma di “proposte” e dunque soggetti a evoluzione.

Senza replicare l’analisi di dettaglio ed il confronto già riportati nel documento “INF-2 Linee progettuali”, ci si limita ad osservare che, da un punto di vista esclusivamente concettuale, il framework più adatto al nostro dominio di intervento è risultato, in prima istanza, WSLA.

Alcune importanti caratteristiche lo rendevano preferibile: maggiore accuratezza nella definizione dei parametri SLA e i loro algoritmi di calcolo, assenza di strutture e interfacce dedicate alla contrattazione e alla stipula on-line degli accordi tra le parti, maggiore semplicità.

Tuttavia l’abbandono da parte di IBM stessa di WSLA e, soprattutto, le indicazioni in tal senso di CNIPA, hanno portato alla scelta di WS-Agreement come formalismo di definizione e di monitoraggio dei parametri SLA.

Questo nuovo orientamento ha comportato una profonda revisione progettuale, non solo di quanto descritto in questo paragrafo, a causa dell’utilizzo di un formalismo dichiaratorio diverso, ma anche per la presenza in WS-Agreement di definizioni di interfacce per la gestione ed il monitoraggio degli accordi che hanno condizionato la progettazione dei componenti di cui al paragrafo 4.3.

Le considerazioni generali riportate in questo paragrafo hanno portato in modo naturale all’individuazione di un sottoinsieme del framework WS-Agreement da utilizzare nello sviluppo del progetto. Le caratteristiche di questo sottoinsieme saranno maggiormente dettagliate nel capitolo successivo.

4.2 Caratteristiche dei parametri SLA

In ambito SPCoop, un episodio di cooperazione applicativa che comporta collaborazione tra un servizio erogatore ed un servizio fruitore genera un processo complesso suddiviso in diverse macro fasi logiche.

Nel caso dell’invocazione di un servizio, la prima fase consiste nel colloquio di un servizio applicativo (non necessariamente un web service) con la Porta di Dominio di riferimento (messaggio applicativo), cui segue il colloquio tra le PDD di riferimento del servizio fruitore e del servizio erogatore (messaggio

SPCoop) ed infine il colloquio tra la PDD di riferimento del servizio erogatore ed il servizio erogatore medesimo (messaggio applicativo). Analogamente, con le fasi in ordine invertito, nel caso di risposta all'invocazione ricevuta.

Le caratteristiche qualitative associate all'erogazione di un servizio, se valutate dal momento della prima invocazione da parte del servizio fruitore fino al momento della ricezione dell'ultima risposta, finiscono per risultare una sorta di sommatoria delle caratteristiche qualitative di tutte le macro fasi eseguite in tutte le iterazioni necessarie al completamento dell'episodio di cooperazione applicativa.

Tenere conto di tutte queste rilevazioni, compresi eventuali malfunzionamenti o errori di qualcuno degli strati attraversati, risulta assai complesso, e ancor più lo diventa se si vuole cercare di distinguere le responsabilità degli endpoint (servizi applicativi erogatori ma anche fruitori) da quelle degli strati infrastrutturali coinvolti.

In questa prima fase del progetto si è manifestata la necessità limitare la complessità di questo scenario. La possibile semplificazione poteva essere realizzata grazie ad uno dei seguenti approcci:

1. limitazione al monitoraggio del “trasporto applicativo”, ovvero dare la possibilità di ricavare parametri SLA, anche molto complessi, a partire dai soli dati di cui è già prevista la tracciatura nell'apposito repository associato alle Porte di Dominio.
2. limitazione alla tracciatura e al monitoraggio “applicativo”, ovvero delegare ai servizi interfacciati dalla Porta di Dominio la scrittura dei dati necessari per la valutazione dei parametri SLA individuati.

Nella prima fase di analisi era stato progettato un approccio incrementale, che partiva con la realizzazione dei sistemi di tracciatura e monitoraggio secondo il primo approccio senza però escludere una successiva estensione al secondo.

La gestione SLA realizzata in tale modo avrebbe avuto vantaggi e limiti piuttosto evidenti.

Sarebbe stata pochissimo invasiva, sia per le Porte di Dominio, sia per i servizi applicativi da esse erogati, in quanto il sistema di tracciatura dei messaggi SPCoop è già previsto quale requisito della PDD e, a partire da tali tracciatore, sarebbe stato possibile creare un sistema di monitoraggio (vedi paragrafo successivo) basato su algoritmi e dotato di interfacce standard per ricostruire parametri SLA anche complessi. La scarsa invasività e la facilità di realizzazione rendevano particolarmente appetibile questa soluzione, soprattutto per la prima iterazione prevista dal ciclo di sviluppo.

D'altro canto, i parametri SLA disponibili sarebbero stati rappresentativi quasi esclusivamente della qualità dell'infrastruttura SPCoop medesima.

Anche mettendo in relazione le tracciatore di tutti i colloqui afferenti ad ogni singola erogazione di servizio, nessuna informazione avrebbe potuto essere ricavata sulle reali caratteristiche di funzionamento dei servizi applicativi gestiti dalle PDD. Le Porte di Dominio stesse esercitano un ruolo di incapsulamento, rendendo indistinguibili i tempi e gli eventuali disservizi dovuti alla comunicazione con i servizi erogati, al coinvolgimento di eventuali broker applicativi, alle elaborazioni di backend.

Nel caso particolare di interazione di tipo “singolo one-way”, la tracciatura dei soli messaggi SPCoop non avrebbe fornito alcuna informazione significativa sul buon esito dell'erogazione e sui tempi di risposta.

Infine, basare lo SLA management del progetto ICAR esclusivamente sulla struttura attuale della Porta di Dominio SPCoop avrebbe potuto comportare un ulteriore rischio. Infatti, poiché quanto sviluppato in questo progetto a nessun titolo potrebbe essere considerato standard SPC, un'evoluzione delle linee guida SPCoop, e in particolare di quanto riguarda la Porta di Dominio, rischierebbe di invalidare quanto qui progettato.

I vantaggi e i limiti del secondo approccio, ossia delle tracciature di tipo “applicativo”, sono complementari a quelle sopra esposte.

Delegare ai soli servizi applicativi l'onere della tracciatura avrebbe significato dover dotare da subito i servizi stessi di nuove funzionalità, mentre le Porte di Dominio, pur mantenendo il loro sistema di tracciatura e monitoraggio come da specifica “CNIPA, Sistema pubblico di cooperazione: porta di dominio, 14 Ottobre 2005”, non sarebbero state coinvolte in questa attività.

Per contro, poter utilizzare parametri SLA basati su metriche realmente “di risorsa”, scevre da tare dovute al trasporto, avrebbe potuto fornire misure dell'effettiva efficienza e qualità dei servizi esposti.

Volendo, in iterazioni successive, comprendere nel monitoraggio messo a disposizione anche i dati forniti dall'infrastruttura SPCoop, sarebbe stato sufficiente integrare quanto già registrato dal sistema di tracciatura delle Porte di Dominio.

Un vantaggio in grado di controbilanciare parzialmente i maggiori oneri gravanti sui servizi applicativi consisterebbe nella libertà che i soggetti erogatori di servizi avrebbero nell'individuare e indicare nell'accordo di servizio i parametri SLA, a patto di calcolare e fornire le misure necessarie al monitoraggio dei parametri medesimi.

Questa libertà dovrebbe ovviamente essere guidata, oltre che dal buon senso nell'individuazione di parametri SLA quantitativamente e qualitativamente ben bilanciati, da quanto fornito nella reference implementation prevista quale deliverable del progetto.

Successivamente a questa fase di analisi, che aveva portato alla scelta del primo approccio (monitoraggio del “trasporto applicativo”), sono intervenute altre riflessioni, scaturite soprattutto in seguito ad incontri di progetto e a sedute tecniche presso CNIPA.

In tali contesti è emerso un orientamento che tende a non attribuire al trasporto applicativo tra Porte di Dominio una reale valenza di indicatore qualitativo dell'erogazione di servizi end-to-end. Il dialogo tra PDD può cioè essere assimilato ad un invariante di infrastruttura, il cui funzionamento è già peraltro monitorato a carico delle PDD medesime.

Pare quindi preferibile, almeno in una prima fase del progetto, concentrarsi su un approccio del secondo tipo che, a fronte di innegabili carichi attribuiti ai servizi applicativi cooperanti, non prevede interazioni con le funzionalità espletate dalle Porte di Dominio ed è in grado di fornire dati maggiormente significativi, dipendenti dal funzionamento interno delle Amministrazioni coinvolte.

Nel recepire tali indicazioni, si è deciso di adottare un procedimento incrementale. Nella prima iterazione del progetto solo i servizi applicativi erogatori tratteranno le metriche relative ai parametri SLA da loro stessi dichiarati nell'accordo di servizio. Al termine della prima iterazione progettuale si valuteranno i responsi forniti secondo questo approccio, con la possibilità di ampliare le funzionalità di tracciatura, estendendole ai servizi fruitori ed eventualmente integrandole con le tracciate delle PDD.

4.3 Sistemi di tracciatura e monitoraggio

A valle dell'individuazione della natura dei dati oggetto di memorizzazione e a fondamento dello SLA management, occorre stabilire le caratteristiche del sistema di tracciatura che provvederà alla loro registrazione, la dislocazione e il tracciato del repository ad essi dedicato e le caratteristiche del sistema di monitoraggio dei parametri SLA dichiarati negli accordi di servizio.

Fermo restando che la progettazione dettagliata di queste componenti sarà esposta nei capitoli successivi, in questo paragrafo si analizzeranno i fattori che hanno condizionato alcune delle soluzioni messe a punto.

Una volta deciso che le metriche di interesse sono rilevate a cura dei servizi erogatori e riguardano il loro proprio funzionamento, restano sul tappeto alcune ipotesi teoricamente percorribili relativamente al luogo di memorizzazione di tali dati:

1. creazione di apposito repository presso i servizi erogatori
2. affiancamento del repository delle tracciate "applicative" a quello già previsto presso le Porte di Dominio di riferimento dei servizi erogatori
3. creazione di una terza parte di memorizzazione e monitoraggio

Ipotesi 1: la fase di registrazione delle tracciate risulterebbe semplificata, poiché non ci sarebbero vincoli tecnologici né su tali funzionalità né sulle caratteristiche del repository stesso. Ogni dominio dovrebbe realizzare una o più "black box", assicurando solo la disponibilità di interfacce standard, definite all'interno di questo progetto, per l'interrogazione delle misure registrate e la valutazione dei parametri SLA corrispondenti. Queste interfacce costituirebbero il sistema di monitoraggio e dovrebbero essere realizzate a loro volta come web service in tutto compatibili con le specifiche SPCoop.

A fronte della minimizzazione dell'invasività e del peso addossato ai servizi applicativi per la costruzione del sistema di tracciatura, questa soluzione comporta un aumento dello sforzo richiesto ai servizi applicativi stessi per la realizzazione del sistema di monitoraggio. Inoltre la disponibilità di tale sistema viene legata indissolubilmente a quella dei servizi applicativi medesimi.

Anche la dislocazione dei repository di raccolta dei dati sulle prestazioni dei servizi all'interno dei domini erogatori dei servizi stessi, con l'inevitabile conseguente mancanza di trasparenza dei processi di

tracciatura e di monitoraggio, potrebbe far nascere dubbi sull'affidabilità dell'intero sistema di Service Level Agreement.

Ipotesi 2: secondo quanto specificato nei requisiti PD_UR-8, PD_UR-9 e loro sotto requisiti (paragrafo “3.4 Requisiti Funzionali” del documento “CNIPA, Sistema pubblico di cooperazione: porta di dominio, 14 Ottobre 2005”), le Porte di Dominio sono tenute, a istituire un sistema di tracciatura, secondo un formato stabilito, dei messaggi SPCoop e dei messaggi diagnostici che sono scambiati negli episodi di cooperazione applicativa e di consentirne la consultazione. Tali requisiti sono stati pienamente recepiti da Regione Toscana, capofila del Task INF-1.

Il requisito PD_UR-1.3 (ibidem), stabilisce invece la necessità di tracciare anche i messaggi applicativi scambiati tra servizi applicativi e Porta di Dominio di riferimento, senza però indicare modalità o tracciato di tali registrazioni.

Questa soluzione prevederebbe dunque una particolare implementazione ed estensione del suddetto requisito, istituendo un repository avente un tracciato formalizzato adatto alla memorizzazione, non solo delle interazioni PDD – servizi applicativi, ma anche delle metriche di risorsa associate alle interazioni medesime e necessarie per il monitoraggio dei parametri SLA pubblicati nell'accordo di servizio.

L'implementazione della PDD dovrebbe dunque prevedere un siffatto repository, le interfacce di comunicazione per permettere ai servizi applicativi di registrare le tracciate previste e l'intero sistema di monitoraggio parametri SLA.

Questo approccio denuncia dunque un duplice punto debole: un significativo sovraccarico strutturale e funzionale per la Porta di Dominio ed una cospicua estensione, in nessun modo coperta da standard, dell'operatività ad essa esplicitamente attribuita dalle direttive SPC.

Ipotesi 3: il framework WSLA cita esplicitamente la possibilità di creare una “terza parte di misura e valutazione delle condizioni”. Anche al di fuori del suddetto formalismo, si potrebbe prevedere la creazione di un unico soggetto esterno ai domini cooperanti, dotato della fiducia di tutte le parti in causa, in grado di effettuare autonomamente misurazioni relative alle metriche previste e di fornire interfacce standard per la verifica dei parametri SLA basati su tali metriche.

Il nostro ambito di applicazione e le scelte operate fin qui non renderebbero possibile dar vita ad un'istituzione avente tutte le succitate caratteristiche.

In primo luogo perché, come esposto al paragrafo 4.2, dovranno essere i servizi applicativi erogatori gli agenti di misurazione e tracciatura.

Inoltre la creazione ed il mantenimento di un soggetto unico e centralizzato esterno comporterebbero difficoltà organizzative e non rispetterebbe il principio, adottato anche da Regione Toscana, capofila del Task INF-1, nella progettazione di registri SICA di secondo livello, della dislocazione dei dati presso il dominio titolare degli stessi.

Ipotesi 4: Come superamento di quanto esposto nell'ipotesi 3, si potrebbe, adottando un approccio ibrido, creare un componente destinato a svolgere il ruolo di “terza parte distribuita” per le sole funzionalità di monitoraggio, mentre il sistema di tracciatura e il repository ad esso associato resterebbero logicamente associati ai sistemi erogatori di servizi.

L'intero sottosistema potrebbe essere affiancato al registro SICA di secondo livello dei domini cooperanti (elemento peraltro non obbligatorio, le cui specifiche vengono fornite da Regione Toscana, capofila Task INF-1) oppure essere dislocato presso le Porte di Dominio degli Enti che si intendono dotare degli strumenti SLA qui descritti.

Le funzionalità rese disponibili da questo sottosistema saranno:

- un repository fisico in grado di memorizzare le tracciate secondo il formato previsto;
- un'interfaccia utilizzabile dai servizi applicativi appartenenti al dominio di riferimento per la registrazione delle tracciate applicative nel repository;
- un'interfaccia esposta come repertorio di web services compatibili SPCoop per il monitoraggio dei parametri SLA basati sulle tracciate depositate nel repository.

A valle della formulazione delle ipotesi di cui sopra, è intervenuta l'indicazione congiunta del project management e di CNIPA, circa la preferibilità del quarto approccio.

5 ARCHITETTURA GENERALE E DOMINIO DI INTERVENTO

I componenti architetture di pertinenza del Task infrastrutturale INF-2 si articolano lungo tre linee di intervento: specifica dei livelli di servizio negli accordi di servizio, sistema di tracciatura e sistema di monitoraggio.

Quanto qui esposto nasce sotto l'influenza delle analisi, delle considerazioni e delle scelte di cui al capitolo precedente.

L'architettura generale della soluzione adottata non può ignorare quanto proposto dal Task INF-1.

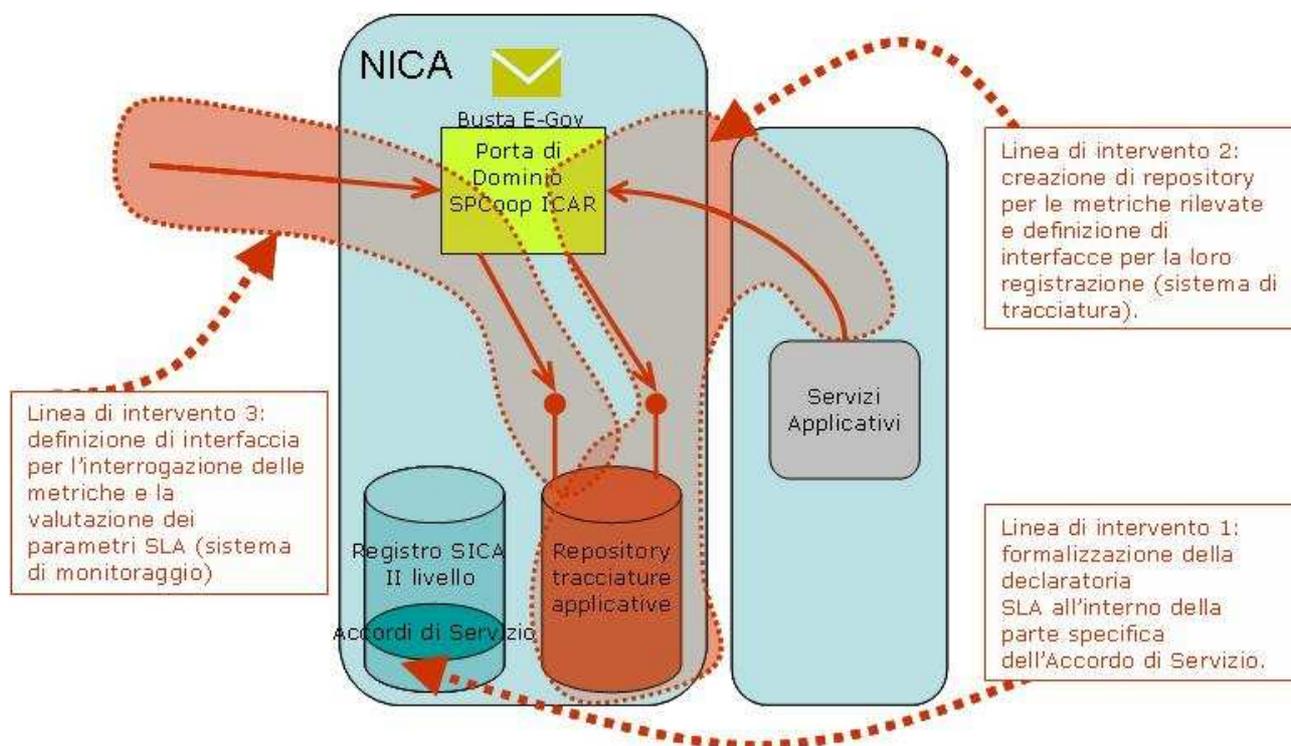
In effetti, le tre linee di intervento si appoggiano e si interfacciano con quanto progettato da Regione Toscana, capofila Task INF-1, che ha definito il complesso delle sue realizzazioni come NICA, “Nodo di Interconnessione per la Cooperazione Applicativa” (Regione Toscana, “INF-1: Realizzazione dell'Infrastruttura di base per l'Interoperabilità e la Cooperazione Applicativa a livello interregionale:

LINEE PROGETTUALI.” v.1.1.6).

In particolare le procedure di pubblicazione degli accordi di servizio sui servizi di registro SICA secondari e la sincronizzazione con il registro SICA generale sono parte dei deliverable del Task INF-1 stesso. Il compito legato alla prima linea di intervento del Task INF-2 è dunque limitato all'individuazione del formalismo da utilizzare nella dichiarazione dei parametri SLA associati ai servizi applicativi cooperanti, lasciando poi a quanto già realizzato da INF-1 i meccanismi di pubblicazione e sincronizzazione.

Le altre due linee di intervento, frutto delle considerazioni di cui ai paragrafi 4.2 e 4.3, godono di una maggiore autonomia, anche se quanto prodotto tiene conto dei principi adottati nel disegno del Task INF-1 e verranno realizzate integrazioni tra le interfacce operatore (console) proposte dai task infrastrutturali 1 e 2.

Alla luce di queste considerazioni, si possono schematizzare l'architettura generale dell'intervento nel seguente modo



La reference implementation prodotta come deliverable del progetto comprenderà esempi di tracciatura e monitoraggio di parametri SLA. Questi esempi, elaborati anche secondo le indicazioni raccolte in ambito progettuale, comprenderanno parametri derivanti da metriche di risorsa “di base”, utili cioè a fondare un sistema di gestione SLA semplice ma significativo ed in grado di evolvere verso forme di monitoraggio più sofisticate.

5.1 Specifica dei livelli di servizio

Come già indicato nel capitolo precedente, la registrazione degli accordi di servizio avviene in modalità off-line, in modo unilaterale o dopo accordi tra soggetto erogatore e soggetto fruitore. E' ammessa la registrazione in tempi distinti della parte comune e delle parti specifiche di uno stesso AS.

CNIPA sta elaborando in bozza un nuovo documento della suite “Sistema pubblico di cooperazione”, contenente le linee guida per la compilazione degli accordi di servizio.

Benché tale testo, nel suo stato attuale, non entri nei dettagli della specifica SLA, anche in ragione della sua opzionalità e dell'assenza di formalismi standard, prevede comunque un segnaposto nel “manifesto dell'AdS” (un documento XML utilizzato, con la funzione di elenco delle parti dell'accordo di servizio, per il dispatching dell'accordo di servizio stesso).

Quindi, una volta stabilite le caratteristiche e le delimitazioni del formalismo adottato in questo progetto, la definizione dei parametri SLA sarà collocata nel posto ad essa riservato in modo del tutto naturale.

Il framework adottato nel progetto ICAR è dunque WS-Agreement, ideato nell'ambito del GGF Global Grid Forum.

Alcune caratteristiche di questo formalismo sono rilevanti per la sua applicazione nel nostro ambito.

Per quanto riguarda la definizione dei parametri SLA, è dato come esplicito obiettivo (GGF, “Web Services Agreement Specification – 7/9/2006”, paragrafo 1.1.2 “Non-goals”) non specificare dettagli delle metriche associate ai parametri stessi come, per esempio, dove e in che modo misurarle. La validità dei parametri SLA (“GuaranteeTerm”) è definita da asserzioni (“ServiceLevelObjective”) basate sulle metriche di riferimento. La definizione del formalismo di tali asserzioni è esplicitamente definita come estranea al framework, tuttavia è possibile utilizzare sia formalismi personalizzati (struttura “CustomServiceLevel”) sia metriche composte basate su metriche di risorsa personalizzate (struttura “KPITarget”).

```
<wsag:GuaranteeTerm Obligated="wsag:ServiceRoleType">
  <wsag:ServiceScope ServiceName="xs:string">
    xs:any
  </wsag:ServiceScope>*
  <wsag:QualifyingCondition>...</wsag:QualifyingCondition?>
  <wsag:ServiceLevelObjective>...</wsag:ServiceLevelObjective>
  <wsag:BusinessValueList>...</wsag:BusinessValueList>
</wsag:GuaranteeTerm>
```

...

```
<wsag:ServiceLevelObjective>
  <wsag:KPITarget> </wsag:KPITarget> |
  <wsag:CustomServiceLevel> ... </wsag:CustomServiceLevel>
</wsag:ServiceLevelObjective>
```

Poiché, come più volte rimarcato, il framework WSLA è molto più orientato alla specifica formale degli algoritmi che, partendo da metriche elementari, portano alla composizione di metriche composte e SLO, si può ipotizzare l'innesto di alcune parti ad esso ispirate all'interno del formalismo WS-Agreement.

WS-Agreement prevede la negoziazione e la conclusione di accordi on-line. Una rilevante parte del formalismo XML proposto è dedicato alla definizione dell'elemento a struttura complessa “AgreementTemplate”. I soggetti che intendono usufruire di un servizio valutano le caratteristiche offerte all'interno di tale struttura e, eventualmente, viene creato un elemento “Agreement” a immagine del template esaminato.

Questa parte di formalismo non è utile ai nostri scopi e dunque tutti gli elementi afferenti all'”AgreementTemplate” non saranno utilizzati. Un procedimento del genere è autorizzato dal framework, che prevede esplicitamente (GGF, “Web Services Agreement Specification – 7/9/2006”,

paragrafo 1.1.1 “Requirements”) l’utilizzo indipendente di parti diverse della specifica e la creazione di accordi al di fuori dei meccanismi di negoziazione proposti.

WS-Agreement non è solo una specifica di linguaggio XML, ma definisce anche protocolli, basati su interfacce astratte, per la creazione di accordi ed il monitoraggio degli stessi on-line. Come appena precisato, la creazione di accordi di servizio on-line non è un nostro obiettivo, mentre lo sviluppo di un sistema di monitoraggio on-line dei parametri SLA individuati costituisce la terza delle nostre linee di intervento.

La descrizione di questo sottosistema occupa il paragrafo 5.3 di questo documento, tuttavia può essere utile, per maggiore completezza dell’analisi di WS-Agreement, anticipare che i web service di interrogazione delle metriche registrate e di verifica dei parametri SLA saranno disegnati rifacendosi alla logica del port type “wsag:AgreementState”, definito dal framework per l’indagine standardizzata degli stati di accordi, servizi e singolo parametro SLA. Nel nostro ambito, essendo l’agreement del tutto gestito a design time, saranno monitorati solo il servizio e i parametri SLA.

In conclusione, lasciando al capitolo 6 il dettaglio completo di quanto proposto, si evidenzia qui che la parte specifica di ogni accordo di servizio riservata alla specifica SLA sarà occupata dalla struttura XML di un “/wsag:Agreement”, definita dal formalismo WS-Agreement di cui allo schema <http://schemas.ggf.org/graap/2006/07/ws-agreement>.

Questo elemento conterrà le identificazioni necessarie del servizio coinvolto (“/wsag:ServiceScope”), dei parametri SLA (“/wsag:GuaranteeTerm”, “/wsag:GuaranteeTerm/wsag:ServiceLevelObjective”); a tal scopo utilizzerà le medesime chiavi utilizzate nelle altre parti dell’accordo di servizio.

5.2 Sistema di tracciatura

Come indicato nel capitolo precedente, nella prima iterazione del progetto solo i servizi applicativi erogatori tracceranno le metriche relative ai parametri SLA da loro stessi dichiarati nell’accordo di servizio.

Al termine di tale prima iterazione progettuale si valuteranno i responsi forniti secondo questo approccio, con la possibilità di ampliare le funzionalità di tracciatura, estendendole ai servizi fruitori ed eventualmente integrandole con le tracciate delle PDD.

La realizzazione del repository delle tracciate applicative costituisce parte integrante della reference implementation prevista quale deliverable del Task INF-2, tuttavia, in analogia con quanto previsto per il registro SICA secondario, l’adozione di questo prodotto non sarà obbligatoria per i domini cooperanti in ambito ICAR.

Se un dominio cooperante sceglierà di non dispiegare presso di sé il repository di cui sopra, dovrà comunque, per potersi avvalere degli strumenti di SLA management progettati, dotarsi di un altro

supporto di memorizzazione, gestione ed esposizione delle tracce applicative rispondente ai requisiti esposti in questo documento, con particolare riferimento a:

- interfaccia di scrittura delle metriche di risorsa;
- identificazione univoca delle tracce memorizzate;
- interfaccia di monitoraggio dei parametri SLA.

Individuata la natura dei dati oggetto di memorizzazione, occorre stabilire le caratteristiche del sistema di tracciatura che provvederà alla loro registrazione, la dislocazione e il tracciato del repository ad essi dedicato e le caratteristiche del sistema di monitoraggio dei parametri SLA dichiarati negli accordi di servizio.

Il disegno del repository delle tracciate deve tenere conto di due esigenze primarie: l'individuazione dell'episodio di cooperazione applicativa cui si riferisce la misura da registrare e l'identificazione della misura medesima.

Senza entrare in dettagli che saranno esplicitati in altro capitolo, la singola tracciatura memorizzata conterrà riferimenti a:

- accordo di servizio;
- fruitore ed erogatore del servizio;
- servizio;
- singola interazione fruitore-erogatore (per correlare metriche di risorsa diverse relative al medesimo episodio). Sarà di tipo “data e ora dell’inizio elaborazione”, in modo da poter definire eventuali finestre temporali;
- misura di metrica di risorsa elementare. Gli identificativi di queste misure corrisponderanno a quanto specificato, a seconda dell’approccio scelto, nella definizione di /wsag:KPITarget o nell’espressione complessa riportata in /wsag:ServiceLevelObjective/wsag:CustomServiceLevel. Entrambe queste tipologie di definizione non fanno parte della specifica WS-Agreement.

La metrica di risorsa rilevata associata all’identificativo complesso individuato sulla base delle indicazioni di cui sopra costituisce il dato elementare, da utilizzare secondo le regole di composizione di espressione complessa (approccio KPI) o l’espressione assertiva (approccio CustomServiceLevel) specificate nell’accordo di servizio.

Le interfacce messe a disposizione dal sistema di tracciatura consentiranno la registrazione delle singole misure di ogni metrica di risorsa prevista nell’accordo di servizio, fornendo la chiave necessaria per l’univoca identificazione dell’episodio di cooperazione applicativa di riferimento e della misura medesima.

5.3 Sistema di monitoraggio

Questo sistema sarà costituito da un'interfaccia, esposta come repertorio di web services compatibili SPCoop e dunque, a regime, dotato a sua volta di Accordo di Servizio, per il monitoraggio dei parametri SLA basati sulle tracciatore depositate nel repository di cui sopra.

Come già osservato, WS-Agreement definisce anche protocolli, basati su interfacce astratte, per il monitoraggio on-line degli accordi di servizio.

In particolare, l'implementazione dell'interfaccia di monitoraggio sarà realizzata ispirandosi alla medesima logica definita dal port type “wsag:AgreementState” fornite nel framework. Tale port type, mediante la definizione di una struttura basata su “Resource Properties”, consente l'interrogazione dello stato degli elementi “Agreement”, “ServiceTerm” e “GuaranteeTerm”.

Le verifiche sugli “Agreement” (accordo nel suo complesso) sono di scarso interesse nel nostro dominio di applicazione, a causa della stipula off-line degli accordi stessi. Dunque un'interrogazione sui parametri che corrispondono a quanto WS-Agreement definisce come “resource property wsag:AgreementState” potrebbe solo restituire invariabilmente il valore “Observed”, corrispondente al significato di accordo siglato ed in funzione.

Il monitoraggio sui parametri che corrispondono a quanto WS-Agreement definisce “ServiceTerm” (nella nostra accezione si tratta di un servizio) e “GuaranteeTerm” (singolo parametro SLA) realizza invece lo scopo del nostro intervento e sarà in grado di restituire i seguenti valori:

per i servizi (“ServiceTerm”):

- Not ready – servizio non utilizzabile;
- Ready – servizio pronto per l'esecuzione;
- Processing – servizio in esecuzione che sta processando una richiesta;
- Idle – servizio in esecuzione e in attesa;
- Completed – esecuzione terminata e non più utilizzabile.

per i parametri SLA (“GuaranteeTerm”):

- Fulfilled – parametro SLA soddisfatto;
- Violated – parametro SLA violato;
- NotDetermined – l'attività effettuata non consente di valutare il parametro SLA.

6 LE COMPONENTI DEL SISTEMA

6.1 Specifica dei livelli di servizio

Come già evidenziato, il framework adottato nel progetto ICAR per la dichiarazione dei parametri SLA nell'accordo di servizio è WS-Agreement, ideato nell'ambito del GGF Global Grid Forum.

Tuttavia il formalismo in questione non ha ancora visto la nascita di una versione stabile ed universalmente accettata. Questo ci obbliga ad utilizzare la bozza attualmente proposta da GGF, confidando che le eventuali nuove versioni siano compatibili con quella adottata.

Il reperimento stesso dell'XMLSchema associato a tale versione è attualmente problematico, essendo risultati spezzati tutti i link indicati all'interno del sito GGF stesso.

Tenendo presente questa impostazione di fondo, viene qui di seguito riportata la struttura di un documento XML corrispondente alle specifiche WS-Agreement, corredata da spiegazioni sul significato da attribuire agli elementi e da commenti circa le scelte di utilizzo o accantonamento di alcuni componenti.

Gli schemi XML da includere negli Accordi di Servizio, parte specifica, definizione parametri SLA potranno essere compilati a partire da questa struttura.

LEGENDA:

[in verde](#): dati da inserire a cura del compilatore;

[in azzurro](#): commenti;

[in magenta](#): elementi previsti dalla specifica ma non significativi nell'ambito di applicazione.

```
<wsag:Agreement
  xmlns:wsag="http://..."          locazione di agreement\_types.xsd
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xsi:schemaLocation="http://... agreement\_types.xsd http://www.w3.org/2001/XMLSchema
  XMLSchema.xsd" >
  <wsag:Name>
    "Nome Accordo XXX"                non obbl., nome dell'AdS nell'ambito del quale il file
                                         viene incluso
  </wsag:Name>
  <wsag:Context>
    <wsag:AgreementInitiator>

```

<p><code>wsa:EndpointReference</code></p>	<p>non obbl. Nel ns. caso, non essendoci template, sarebbe utile averlo, a meno di voler evitare di replicare qui dati già specificati in altra parte dell'AdS (specifica dell'interf. logica del fruitore).</p> <p>Può essere di qualsiasi tipo (anche URI), ma forse meglio adeguarsi a WS-Addressing. Si può decidere che è l'ind. del fruitore (Initiator = Fruitore, vedi <code>wsag:ServiceProvider</code>)</p> <p>Da esplodere.</p>
<p><code></wsag:AgreementInitiator></code></p>	
<p><code><wsag:AgreementResponder></code></p>	
<p><code>wsa:EndpointReference</code></p>	<p>non obbl., vedi sopra. Si può decidere che è l'ind. dell'erogatore (Responder = Erogatore, vedi <code>wsag:ServiceProvider</code>)</p> <p>Da esplodere.</p>
<p><code></wsag:AgreementResponder></code></p>	
<p><code><wsag:ServiceProvider></code></p>	
<p><code>“AgreementResponder”</code></p>	<p>non obbl., default=AgreementResponder; riflette la scelta fatta</p>
<p><code></wsag:ServiceProvider></code></p>	
<p>non inclusi elementi non obbl. relativi a template e expiration time</p>	
<p><code><xsd:xs:any/></code></p>	<p>possibilità di aggiungere 1 o più elementi custom</p>
<p><code></wsag:Context></code></p>	
<p><code><wsag:Terms></code></p>	
<p><code><wsag:All></code></p>	<p>elemento per la combinazione ricorsiva di elementi diversi</p>
<p><code><wsag:ServiceDescriptionTerm wsa:name="NomeTerm1" wsag:serviceName="Servizio1"></code></p>	<p>(zero o più)utilizzato per indicare il servizio sul quale ci si accorda. Si potrebbe evitare di replicare qui dati già specificati in altra parte dell'AdS (specifica dell'interfaccia concettuale)</p>
<p><code><xsd:xs:any/></code></p>	<p>possibilità di aggiungere 1 o più elementi custom, per es. riferimenti wsdl.</p>
<p><code></wsag:ServiceDescriptionTerm></code></p>	
<p><code><wsag:ServiceReference wsa:name="NomeSR1" wsag:serviceName="Service1"></code></p>	<p>(zero o più)utilizzato per indicare riferimenti, anche “domain specific” al servizio sul quale ci si accorda. Anche questi dati risulterebbero già specificati in altra parte dell'AdS.</p>
<p><code><xsd:xs:any/></code></p>	<p>possibilità di aggiungere 1 o più elementi custom, per es. un EndpointReference.</p>
<p><code></wsag:ServiceReference></code></p>	

- `<wsag:ServiceProperties wsag:Name="NomeSP1" wsag:ServiceName="Service1">`
 (zero o più)serve per dichiarare “variabili” associate al servizio e utilizzabili nella definizione nei ServiceLevelObjective.
 L’attributo “wsag:ServiceName” dovrà corrispondere al valore dell’attributo “nome” dell’elemento “servizio-spcoop” nel manifesto dell’AdS.
- `<wsag:VariableSet>`
`</wsag:VariableSet>`
 Questo approccio, basato su un insieme di variabili cui si attribuisce un nome e che vengono utilizzate negli SLO, sia stato recepito e probabilmente superato dall’introduzione di KPI e CustomServiceLevel. (il fatto che la definizione di SLO sia diversa dagli esempi riportati mi fa pensare che ci sia stata un’evoluzione...)
- `</wsag:ServiceProperties>`
- `<wsag:GuaranteeTerm` obbl.; ogni elemento corrisponde ad un parametro SLA. Ne possono essere elencati uno o più.
`wsag:Nome="Parametro1">` nome del parametro SLA
`wsag:Obligated="ServiceProvider">` riflette la nostra impostazione
- `<wsag:ServiceScope wsag:ServiceName="Service1">`
 specifica del servizio al quale si riferisce il “GuaranteeTerm”.
 L’attributo “wsag:ServiceName” dovrà corrispondere al valore dell’attributo “nome” dell’elemento “servizio-spcoop” nel manifesto dell’AdS.
 Questo è l’elemento chiave per il raccordo e l’accoppiamento tra la dichiarazione SLA ed il resto dell’AdS
 E’ possibile che un medesimo parametro SLA si applichi a più di un servizio. Allora gli elem. “ServiceScope” saranno più di uno
- `</wsag:ServiceScope>`
- `<wsag:QualifyingCondition>` condizioni, espresse in linguaggio “esterno”, che costituiscono
`</wsag:QualifyingCondition>` prerequisito per il parametro SLA. Non obbl.
- `<wsag:ServiceLevelObjective>` definisce le espressioni che stabiliscono il rispetto del parametro SLA associato al “GuaranteeTerm”. Tali espressioni possono utilizzare un formalismo di asserzione libero e totalmente estraneo alla specifica WS-Agreement (utilizzo di “CustomServiceLevel”) o, in alternativa, possono definire, sempre esternamente alla specifica, di una quantità misurabile associata ad un valore soglia (utilizzo di “KPITarget” vedi cap. 5.2).

Il contenuto degli elementi “wsag:CustomServiceLevel” sarà un’espressione complessa a valore booleano, basata su predicati logici facenti riferimento a metriche composte e valori di soglia.

Le metriche composte conterranno operatori definiti nel linguaggio e operandi, a loro volta eventualmente complessi, ma corrispondenti in ultima analisi a metriche di risorsa elementari.

Queste ultime saranno identificate univocamente da un nome in linguaggio naturale possibilmente esplicativo. Tale nome coinciderà con la relativa parte di chiave nel sistema di tracciatura, permettendo di identificare una singola traccia all’interno del repository (vedi paragrafo 5.2).

Quando l’espressione complessa contenuta in un elemento “wsag:CustomServiceLevel” assumerà valore “true” il parametro SLA relativo sarà da considerare rispettato (“Fulfilled”).

La definizione di parametri SLA relativi ad un servizio applicativo avviene dunque a partire dall’individuazione di metriche di risorsa aventi attinenza col servizio applicativo medesimo e di algoritmi in grado di costruire, basandosi su tali metriche elementari, il valore complessivo del parametro e, di conseguenza, la rispondenza con quanto pattuito.

Il completamento di questo processo logico di individuazione dei parametri SLA ha una duplice ricaduta sulle attività a carico degli erogatori dei servizi applicativi:

- le metriche di risorsa devono essere dichiarate come operandi terminali nella struttura dei “wsag:CustomServiceLevel” e costituiscono parte di chiave nel sistema di tracciatura;
- gli algoritmi devono essere esplicitati, utilizzando il linguaggio di asserzioni, nella struttura interna ai “wsag:CustomServiceLevel” da noi proposta e, in tale modo, guidano la realizzazione ed il funzionamento dei web services previsti nel sistema di monitoraggio.

Come accennato nel paragrafo 5.3, il framework WSLA di IBM prevedeva una consistente parte dedicata alla definizione di un linguaggio atto alla costruzione di metriche composte a partire da metriche di risorsa elementari.

Nell’elaborare un linguaggio adatto al nostro ambito applicativo, si è partiti dal sottoinsieme WSLA di cui sopra, mantenendo alcuni costrutti originali ma approdando però ad una specifica formale del tutto autonoma.

Una delle modifiche più significative consiste nell’introduzione di un nuovo elemento “BasicMetric”, destinato a rappresentare le metriche di risorsa. In WSLA gli elementi relativi a metriche di base contenevano generalmente un identificativo URI mediante il quale interrogare le misure residenti presso i domini cooperanti. Nel nostro contesto, invece, le misurazioni delle metriche di risorsa vengono via via depositate nel repository di tracciatura, e dunque è sufficiente dotare le metriche di risorsa solo di identificativo univoco per reperirle in fase di monitoraggio nel repository stesso.

Qui di seguito si pubblica il formato proposto per il linguaggio di asserzioni (XMLSchema).

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:complexType name="CustomServiceLevelType">
    <xs:sequence>
      <!-- valore soglia e operatore per testare il rispetto del parametro -->
      <xs:element ref="Threshold" />
      <!-- valore composto da testare -->
      <xs:element name="Function" type="FunctionType" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Threshold">
    <xs:complexType>
      <xs:attribute name="Operator" type="OperatorType" use="required" />
      <xs:attribute name="Value" type="xs:string" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="OperatorType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Greater" />
      <xs:enumeration value="GreaterEqual" />
      <xs:enumeration value="Less" />
      <xs:enumeration value="LessEqual" />
      <xs:enumeration value="Equal" />
      <xs:enumeration value="NotEqual" />
    </xs:restriction>
  </xs:simpleType>
  <!-- Funzioni per il calcolo di metriche composte (def. astratta) -->
  <xs:complexType name="FunctionType" abstract="true">
    <xs:attribute name="resultType" type="Type"/>
  </xs:complexType>
  <!-- Tipi dei valori di ritorno -->
  <xs:simpleType name="Type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="integer" />
      <xs:enumeration value="float" />
      <xs:enumeration value="double" />
      <xs:enumeration value="long" />
      <xs:enumeration value="byte" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

```

        <xs:enumeration value="boolean" />
        <xs:enumeration value="string" />
        <xs:enumeration value="time" />
    </xs:restriction>
</xs:simpleType>
<xs:element name="BasicMetric">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="Unit" type="xs:string" />
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<!-- Funzioni standard (estensioni del tipo astratto) -->
<!-- Funzioni base aritmetiche -->
<xs:complexType name="Plus">
    <xs:complexContent>
        <xs:extension base="FunctionType">
            <xs:sequence>
                <xs:element name="Operand" type="OperandType" minOccurs="2"
maxOccurs="2" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Minus">
    <xs:complexContent>
        <xs:extension base="FunctionType">
            <xs:sequence>
                <xs:element name="Operand" type="OperandType" minOccurs="2"
maxOccurs="2" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Divide">
    <xs:complexContent>
        <xs:extension base="FunctionType">
            <xs:sequence>

```

```

maxOccurs="2" />
        <xs:element name="Operand" type="OperandType" minOccurs="2"
        </xs:sequence>
    </xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="Times">
    <xs:complexContent>
        <xs:extension base="FunctionType">
            <xs:sequence>
                <xs:element name="Operand" type="OperandType" minOccurs="2"
maxOccurs="2" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Round">
    <xs:complexContent>
        <xs:extension base="FunctionType">
            <xs:sequence>
                <xs:choice>
                    <xs:element ref="BasicMetric" />
                    <xs:element name="Function" type="FunctionType" />
                </xs:choice>
                <xs:element name="Digits" type="xs:long" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- Funzioni Complesse -->
<xs:complexType name="Mean">
    <xs:complexContent>
        <xs:extension base="FunctionType">
            <xs:sequence>
                <!-- Elemento del quale si collezioneranno i valori sui quali poi calcolare la
media -->
                <xs:choice>
                    <xs:element ref="BasicMetric" />
                    <xs:element name="Function" type="FunctionType" />
                </xs:choice>
                <!-- Numero di valori o intervallo di tempo sui quali calc. la media -->

```

```

                <xs:element ref="Window" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Max">
    <xs:complexContent>
        <xs:extension base="FunctionType">
            <xs:sequence>
                <!-- Elemento del quale si collezioneranno i valori sui quali poi calcolare il
max -->
                <xs:choice>
                    <xs:element ref="BasicMetric" />
                    <xs:element name="Function" type="FunctionType" />
                </xs:choice>
                <!-- Numero di valori o intervallo di tempo sui quali calc. il max -->
                <xs:element ref="Window" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Min">
    <xs:complexContent>
        <xs:extension base="FunctionType">
            <xs:sequence>
                <!-- Elemento del quale si collezioneranno i valori sui quali poi calcolare il min
-->
                <xs:choice>
                    <xs:element ref="BasicMetric" />
                    <xs:element name="Function" type="FunctionType" />
                </xs:choice>
                <!-- Numero di valori o intervallo di tempo sui quali calc. il min -->
                <xs:element ref="Window" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="Sum">
    <xs:complexContent>
        <xs:extension base="FunctionType">
            <xs:sequence>

```

```

sommatoria -->
    <!-- Elemento del quale si collezioneranno i valori sui quali poi calcolare la
    <xs:choice>
        <xs:element ref="BasicMetric" />
        <xs:element name="Function" type="FunctionType" />
    </xs:choice>
    <!-- Numero di valori o intervallo di tempo sui quali calc. la sommatoria -->
    <xs:element ref="Window" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="Hits">
    <xs:complexContent>
        <xs:extension base="FunctionType">
            <xs:sequence>
                <!-- Elemento del quale si conteranno le occorrenze (senza riguardo al valore
contenuto) -->
                <xs:element ref="BasicMetric"/>
                <!-- eventuale valore soglia e operatore per decidere se contare le letture -->
                <xs:element ref="Threshold" minOccurs="0" />
                <!-- Numero di valori o intervallo di tempo relat. al quale contare le
rilevazioni; in questo caso se la Window è di tipo "Times" e non c'è "Threshold", il risultato è pleonastico -->
                <xs:element ref="Window" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<xs:complexType name="OperandType">
    <xs:sequence>
        <xs:choice>
            <xs:element ref="BasicMetric" />
            <xs:element name="Function" type="FunctionType" />
            <xs:element name="LongScalar" type="xs:long" />
            <xs:element name="FloatScalar" type="xs:float" />
        </xs:choice>
    </xs:sequence>
</xs:complexType>
<xs:element name="Window">
    <xs:complexType>
        <xs:choice>
            <xs:element name="Interval" type="IntervalTimeType" />

```

```
        <xs:element name="Times" type="xs:long" />
    </xs:choice>
</xs:complexType>
</xs:element>
<xs:simpleType name="IntervalTimeType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Year"/>
        <xs:enumeration value="SixMonths"/>
        <xs:enumeration value="ThreeMonths"/>
        <xs:enumeration value="Month"/>
        <xs:enumeration value="TwoWeeks"/>
        <xs:enumeration value="Week"/>
        <xs:enumeration value="Day"/>
        <xs:enumeration value="SixHours"/>
        <xs:enumeration value="Hour"/>
    </xs:restriction>
</xs:simpleType>
</xs:schema>
```

Di seguito si esplicitano alcune ulteriori considerazioni sul linguaggio di specifica dei parametri SLA, relative in particolare ai criteri di annidamento delle funzioni e alla definizione delle finestre di osservazione.

Lo schema XML che estende “wsag” utilizza, per la definizione di funzioni, un approccio ricorsivo, secondo il quale gli operandi di una funzione possono a loro volta essere funzioni, senza alcun vincolo sintattico sul numero di annidamenti.

Alcune di queste funzioni sono caratterizzate dalla presenza di un elemento “Window” che definisce la finestra (temporale o quantitativa) di osservazione all’interno della quale reperire i dati da utilizzare.

L’unione di questi fattori comporta la seguente conseguenza: una funzione avente associato un intervallo può contenere come operando un’altra funzione riferita ad altro intervallo e così via, senza limiti ai livelli di ricorsione.

Questo fatto, unito ad osservazioni di carattere generale, ha portato all’individuazione di almeno tre problemi:

1. al di là di un certo numero di annidamenti, la consistenza ed il reale significato delle finestre di osservazione vengono messi seriamente in forse;
2. la sintassi XML proposta rende possibile descrivere parametri SLA di fatto degeneri, in quanto privi di funzioni aventi elemento “Window”, e dunque senza nozioni sull’intervallo di osservazione delle metriche di risorsa utilizzate;

3. occorre specificare alcune regole base per l'interpretazione di eventuali di finestre di osservazione annidate.

Si è dunque deciso di creare le seguenti regole:

1. percorrendo l'alberatura dell'XML di ogni parametro SLA, ogni ramo la cui foglia è un elemento "BasicMetric" non può contenere più di due funzioni aventi elemento "Window" (in futuro si potrebbe a tre);
2. percorrendo l'alberatura dell'XML di ogni parametro SLA, ogni ramo la cui foglia è un elemento "BasicMetric" deve contenere almeno una funzione avente elemento "Window";
3. in caso di due funzioni annidate aventi elemento "Window" di tipo omogeneo (temporale o quantitativo), si richiede che la finestra più interna sia non più ampia di quella esterna. ù

I controlli sul rispetto di queste regole verranno effettuati dalla console di INF-1, al momento dell'associazione di un file wsag ad un accordo di servizio. L'associazione non sarà consentita qualora l'esame del contenuto del file segnali la violazione di una regola di compilazione. Questo comporterà, in un secondo tempo, l'impossibilità di dare inizio alla tracciatura e monitoraggio dell'occorrenza di servizio in questione.

E' inoltre necessario specificare i criteri che il sistema di monitoraggio adotterà per il calcolo dei parametri SLA che prevedano funzioni con intervalli annidate.

La finestra più esterna servirà per selezionare l'insieme delle tracciature da prendere in esame (un numero predeterminato se l'elemento "Window" è di tipo "Times", quelle registrate in un lasso di tempo se l'elemento "Window" è di tipo "Interval"). Per ognuna delle tracciature individuate, fisicamente composta da un record per ogni metrica di risorsa coinvolta nel calcolo, si applicherà la finestra più interna, come se la singola tracciatura fosse l'ultima da osservare.

Esempio per due finestre di tipo temporale:

Finestra più esterna: "week"

Finestra più interna: "day"

Procedura: vengono lette tutte le tracciature di tutte le "BasicMetric" coinvolte nella settimana di riferimento. Per ognuna di queste, a partire dalla più recente, si valuta la funzione più interna a partire dalle 24 ore precedenti. Ognuno di questi risultati parziali viene utilizzato per il calcolo della funzione più esterna. La valutazione della funzione interna si arresta quando le 24 ore precedenti alla tracciatura in esame corrispondono ad una data/ora esterna alla settimana di riferimento.

Esempio per due finestre di tipo quantitativo:

Finestra più esterna: "150"

Finestra più interna: "10"

Procedura: vengono lette le 150 tracciature di tutte le “BasicMetric” a partire dalla più recente. Per ognuna di queste, a partire dalla più recente, si valuta la funzione più interna basandosi su quella e le 9 precedenti. Ognuno di questi risultati parziali viene utilizzato per il calcolo della funzione più esterna. La valutazione della funzione interna si arresta quando le 9 tracciature precedenti a quella in esame non sono tutte comprese nelle 150 di riferimento.

Esempio per due finestre di tipo eterogeneo:

Finestra più esterna: “week”

Finestra più interna: “15”

Procedura: vengono lette tutte le tracciature di tutte le “BasicMetric” coinvolte nella settimana di riferimento. Per ognuna di queste, a partire dalla più recente, si valuta la funzione più interna basandosi su quella e le 14 precedenti. Ognuno di questi risultati parziali viene utilizzato per il calcolo della funzione più esterna. La valutazione della funzione interna si arresta quando le 14 tracciature precedenti a quella in esame non hanno tutte una data/ora compresa nella settimana di riferimento. Nel caso in cui la settimana di riferimento non contenga almeno 15 tracciature, il parametro SLA avrà valore “Not Determined”.

Esempio per due finestre di tipo eterogeneo:

Finestra più esterna: “80”

Finestra più interna: “hour”

Procedura: vengono lette le 80 tracciature di tutte le “BasicMetric” a partire dalla più recente. Per ognuna di queste, a partire dalla più recente, si valuta la funzione più interna a partire dall’ora precedente. Ognuno di questi risultati parziali viene utilizzato per il calcolo della funzione più esterna. La valutazione della funzione interna si arresta quando l’ora precedente alla tracciatura in esame contiene tracciature non comprese nelle 80 di riferimento. Nel caso in cui le 80 tracciature di riferimento siano tutte comprese in un intervallo temporale inferiore ad un’ora, il parametro SLA avrà valore “Not Determined”.

Va infine precisato che è possibile avere, al livello interno di annidamento, più di una funzione dotata di finestra di osservazione in parallelo (ad esempio due operandi di una somma a sua volta utilizzata dalla funzione “Max”).

In questo caso, ferme restando le due regole di cui sopra, il parallelismo richiede che il numero di valori utilizzato sia il minimo tra quelli potenzialmente forniti dalle funzioni in parallelo. Dunque, la funzione la cui finestra di riferimento individuerà il minimo numero di tracciature compatibili con l’intervallo al primo livello di annidamento, fisserà tale numero anche per le altre funzioni cooperanti al medesimo parallelismo.

Esempio di parallelismo annidato:

Finestra più esterna: “day”

Finestra più interna in parallelo numero 1: “10”

Finestra più interna in parallelo numero 2: “hour”

Procedura: vengono lette tutte le tracciature di tutte le “BasicMetric” coinvolte nel giorno di riferimento. Per ognuna di queste, a partire dalla più recente, si valuta la funzione più interna numero 1 basandosi su quella e le 9 precedenti e la funzione più interna numero 2 a partire dall’ora precedente. Ognuno di questi risultati parziali viene utilizzato per il calcolo della funzione più esterna. Quando la valutazione di una delle due funzioni in parallelo si arresta (vedere esempi precedenti), viene fermata anche la valutazione dell’altra. Nel caso in cui la settimana di riferimento non contenga almeno 10 tracciature, il parametro SLA avrà valore “Not Determined”.

Esempi di declaratorie di parametri SLA sono stati raccolti nel documento “Task INF-2: Gestione di Strumenti di Service Level Agreement a livello interregionale - Definizione dei Parametri soggetti a Monitoraggio SLA (INF2_PAR)”.

6.2 Specifica del sistema di tracciatura

Secondo le specifiche WS-Agreement il sistema di tracciatura deve permettere ai servizi applicativi interdominio di tracciare lo stato dell’occorrenza di Servizio (“ServiceTerm”) e lo stato dei parametri SLA (“GuaranteeTerm”) contenuti nel manifesto dell’Accordo di Servizio (vedi paragrafo 4.1).

Il sistema di tracciatura qui descritto prevede i seguenti sotto componenti:

- **DBWSA** - un database in cui sono memorizzate le misurazioni delle metriche di risorsa associate ai parametri SLA contenuti nell’Accordo di Servizio e un dato indicatore dello stato del servizio;
- **Interfaccia di scrittura sul Sistema di Tracciatura** - un componente che espone dei metodi verso i servizi applicativi erogatori, utilizzabili per:
 - o aggiornare lo stato dell’occorrenza di servizio (relativo ad una coppia erogatore-fruttore);
 - o inserire i dati necessari al sistema di tracciatura per generare gli SLA presenti nell’Accordo di Servizio.
- **Interfaccia Operatore** - per consentire interattivamente di:
 - o inserire nuove occorrenze di servizi.

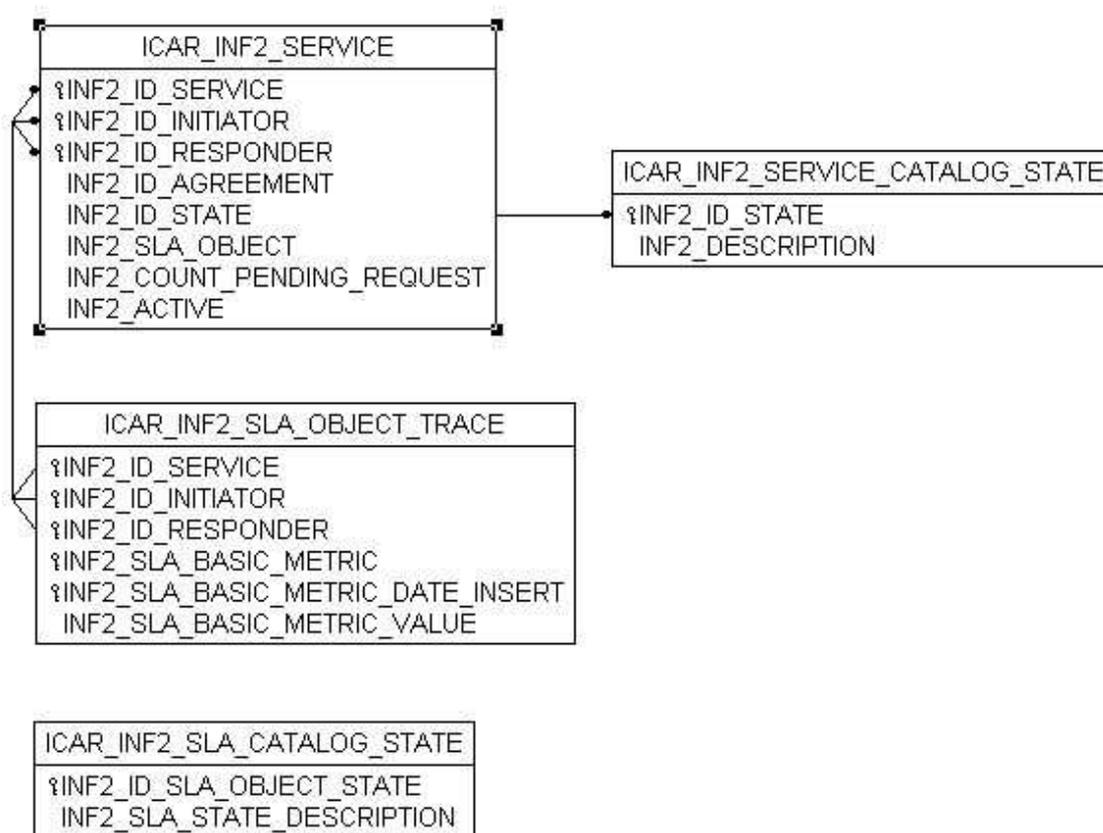
6.2.1 DBWSA – Database Web Service Agreement

La conformazione di questo database è disegnata per consentire di tener traccia di:

- lo stato delle occorrenze di servizio messe a disposizione dall’Ente Erogatore;

- le misure associate alle metriche di risorsa dichiarate negli Accordi di Servizio. Tali misure saranno realizzate in concomitanza con ogni erogazione del servizio relativo da parte dei servizi erogatori;
- gli algoritmi pubblicati nel repository degli Accordi di Servizio, per facilitare il calcolo dei parametri SLA da parte del sistema di monitoraggio;

Lo schema del database è il seguente:



Rispetto alla versione precedente di specifica, lo schema delle base dati è stato modificato. I cambiamenti maggiormente rilevanti riguardano la tabella ICAR_INF2_SERVICE.

E' stata modificata la chiave primaria della tabella, facendo sì che il sistema di tracciatura possa immediatamente individuare i record di interesse a partire dai dati passati dalla Porta di Dominio (identificatore servizio, identificatore fruitore, identificatore erogatore). La presenza in chiave dell'identificativo dell'erogatore garantisce inoltre compatibilità nei casi di servizi multierogatore.

E' stato aggiunto l'attributo INF2_COUNT_PENDING_REQUEST per consentire il corretto monitoraggio dello stato di un'occorrenza di servizio anche nel caso di erogazioni quasi simultanee al medesimo fruitore.

ICAR_INF2_SERVICE: destinata a contenere le rilevazioni dello stato di ogni occorrenza di servizio. Ad ogni erogazione di un servizio, lo stato relativo alla corrispondente occorrenza verrà aggiornato due volte, secondo la logica di seguito riportata.

All'inizio dell'elaborazione viene testato il valore di INF2_ID_STATE; in caso di valore uguale a "READY_IDLE" si effettua un update dello stato del servizio in questione a "READY_PROCESSING" e l'impostazione del contatore INF2_COUNT_PENDING_REQUEST a uno. In caso di valore "READY_PROCESSING" si procede al solo incremento del contatore.

Alla fine dell'elaborazione viene testato il valore di INF2_COUNT_PENDING_REQUEST; in caso di valore uguale a uno si effettua un update dello stato del servizio in questione a "READY_IDLE" e l'impostazione del contatore INF2_COUNT_PENDING_REQUEST a zero. In caso di valore maggiore di uno si procede al solo decremento del contatore.

Ogni record conterrà anche gli algoritmi, desunti dalla parte specifica di riferimento dell'AdS, per la valutazione dei parametri SLA, allo scopo di renderne più agile l'utilizzo.

Tabella: ICAR_INF2_SERVICE	
CAMPO	DESCRIZIONE
INF2_ID_SERVICE	Primary key. Identificatore del servizio (valore dell'attributo "nome" dell'elemento "servizio-spcoop" nel manifesto dell'AdS).
INF2_ID_INITIATOR	Primary key. Identificativo dell'Ente fruitore del Servizio.
INF2_ID_RESPONDER	Primary key. Identificativo dell'Ente erogatore del Servizio (valore dell'attributo "nome" dell'elemento "soggetto-referente" nel manifesto della parte comune dell'AdS).
INF2_ID_AGREEMENT	Identificatore dell'Accordo di Servizio (parte comune) cui il servizio erogato fa riferimento. (valore dell'attributo "nome" dell'elemento "accordo-servizio" nel manifesto dell'AdS). Questo dato viene ricavato in modo non ambiguo grazie ai tre identificatori parte di chiave.
INF2_ID_STATE	Identificatore dello stato dell'occorrenza di servizio

	(vedi tabella ICAR_INF2_CATALOG_STATE).
INF2_ID_SLA_OBJECT	Contiene l'intera sezione XML dell'AdS relativa a tutti i parametri SLA ("wsag:ServiceLevelObjective") associati all'occorrenza di servizio. Si tratta di un campo di servizio, che è utile affinché il sistema di monitoraggio non debba, ogniqualevolta venga richiesta la verifica di un parametro, accedere ai registri SICA per leggere gli algoritmi associati.
INF2_COUNT_PENDING_REQUEST	Contatore richieste in fase di elaborazione.
INF2_ACTIVE	Valore booleano che indica se l'occorrenza di servizio è attiva o no. Solo le occorrenze attive possono essere tracciate.

ICAR_INF2_SLA_OBJECT_TRACE: tabella cardine del sistema di tracciatura. Destinata a contenere i valori delle metriche di risorsa associate all'erogazione di un servizio tra due soggetti che, a tale scopo, hanno sottoscritto un Accordo di Servizio.

Tabella: ICAR_INF2_SLA_OBJECT_TRACE	
CAMPO	DESCRIZIONE
INF2_ID_SERVICE	Primary key, foreign key. Identificatore del servizio (valore dell'attributo "nome" dell'elemento "servizio-spcoop" nel manifesto dell'AdS).
INF2_ID_INITIATOR	Primary key, foreign key. Identificativo dell'Ente fruitore del Servizio.
INF2_ID_RESPONDER	Primary key, foreign key. Identificativo dell'Ente erogatore del Servizio (valore dell'attributo "nome" dell'elemento "soggetto-referente" nel manifesto della parte comune dell'AdS).
INF2_SLA_BASIC_METRIC	Primary key. Campo che contiene il nome della metrica di risorsa di cui si memorizza la misura. Si tratta del valore dell'elemento "BasicMetric" desunto dalla dichiarazione dei livelli di servizio presente nella parte specifica dell'AdS di riferimento.
INF2_SLA_BASIC_METRIC_DATE_INSERT	Primary key. Data e ora in cui è stata inserita

	la misura.
INF2_SLA_BASIC_METRIC_VALUE	Valore della misura della metrica di risorsa.

ICAR_INF2_CATALOG_STATE: destinata a contenere i valori dei possibili stati di un servizio erogato. Tali valori sono mutuati dalla specifica WS-Agreement.

Tabella: ICAR_INF2_SERVICE_CATALOG_STATE	
CAMPO	DESCRIZIONE
INF2_ID_STATE	Primary key. Identificatore dello stato del servizio
INF2_DESCRIPTION	<p>I possibili valori che descrivono lo stato di un servizio sono:</p> <ul style="list-style-type: none"> • NOT_READY: se il servizio in quel momento non è più erogato e quindi risulta non disponibile; • READY_IDLE : se il servizio è disponibile ed in attesa di una richiesta; • READY_PROCESSING: se il servizio è disponibile e sta processando una richiesta; • COMPLETED: se il servizio ha eseguito un'elaborazione e non è più utilizzabile.

ICAR_INF2_SLA_CATALOG_STATE: destinata a contenere i valori dei possibili stati di un parametro SLA previsto nella dichiarazione dei livelli di servizio della parte specifica dell'AdS. Tali valori sono mutuati dalla specifica WS-Agreement.

Tabella: ICAR_INF2_SLA_CATALOG_STATE	
CAMPO	DESCRIZIONE
INF2_ID_SLA_OBJECT_STATE	Primary key. Identificatore dello stato del parametro SLA.
INF2_SLA_STATE_DESCRIPTION	<p>I possibili valori che descrivono lo stato di un parametro SLA sono:</p> <ul style="list-style-type: none"> • NOT_DETERMINED stato non determinabile, per esempio a causa di mancanza di tracciature sufficienti. • FULFILLED parametro SLA è soddisfatto (il predicato costruito sulla base dell'elemento "Threshold" dalla dichiarazione dei livelli di servizio

	<p>presente nella parte specifica dell'AdS di riferimento restituisce valore "true")</p> <ul style="list-style-type: none"> • VIOLATED quando il parametro SLA non è soddisfatto (il predicato costruito sulla base dell'elemento "Threshold" dalla dichiarazione dei livelli di servizio presente nella parte specifica dell'AdS di riferimento restituisce valore "false")
--	--

6.2.2 Interfaccia di scrittura sul Sistema di Tracciatura

Nell'architettura proposta, il Sistema di Tracciatura dovrebbe avere come naturale collocazione l'interno del dominio di appartenenza dei servizi erogatori verso i quali espone le interfacce di scrittura.

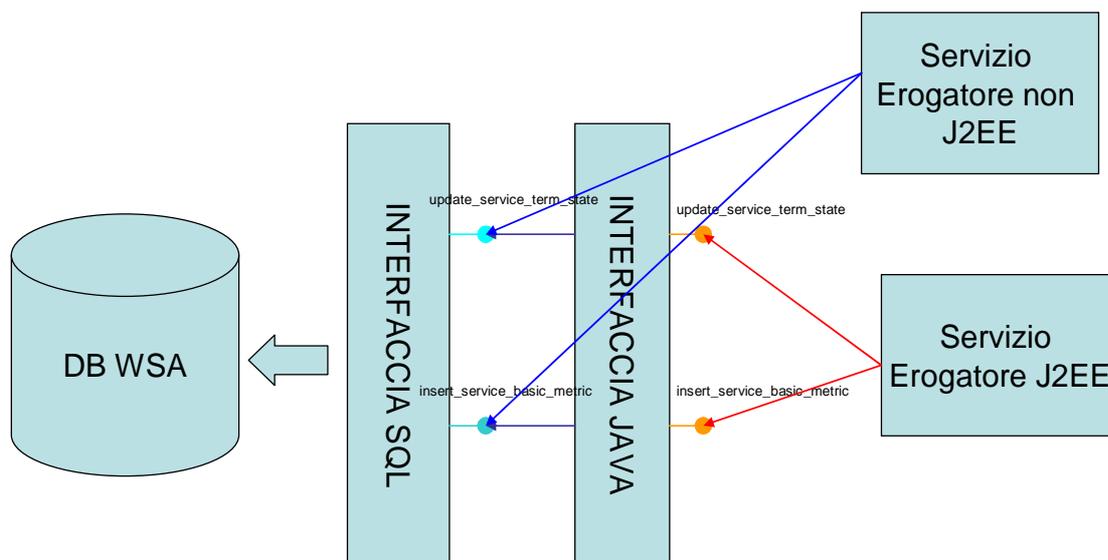
Potrebbe quindi non essere necessaria l'esposizione di interfacce in modalità Web Service (comunque descritte in coda al paragrafo) e, nell'ottica di velocizzare e semplificare l'interazione tra Sistema di Tracciatura ed i servizi erogatori, le interfacce esposte risulterebbero le seguenti:

- una libreria java attraverso la quale i servizi erogatori, sviluppati in analogia tecnologia, possono registrare il proprio stato e le misurazioni delle metriche di risorsa;
- un set di procedure SQL, atte a registrare stato dei servizi e misurazioni di metriche di risorsa, per i servizi erogatori che non utilizzano tecnologie J2EE. Questo approccio massimizza la velocità di accesso, ma vincola al tipo di RDBMS utilizzato. Non si esclude quindi l'evoluzione verso un'interfaccia di tipo EJB.

Entrambe le modalità di scrittura sono rese disponibili ai servizi erogatori per modificare il proprio stato o per inserire le rilevazioni delle metriche di risorsa, in modo tale che il Sistema di Monitoraggio sia in grado, attraverso gli opportuni servizi integrati sulla Porta di Dominio in modalità trasparente, di fornire lo stato delle occorrenze di servizio e lo stato dei parametri SLA previsti nell'accordo di servizio.

Di seguito vengono schematizzate le due possibili modalità di interazione tra servizi erogatori e sistema di tracciatura:

Interfaccia di scrittura del Sistema di Tracciatura



Anche l'interfaccia java utilizza, in modo mediato, l'interfaccia SQL disponibile per i servizi erogatori che non utilizzano tecnologie J2EE.

Nel seguito si pubblica l'interfaccia java verso il repository delle tracciature, per permettere ai servizi erogatori di:

- modificare lo stato dell'occorrenza di servizio sottoposto a tracciatura (secondo la logica descritta nel sottoparagrafo 6.2.1);
- inserire nel componente DBWSA le misure relative alle metriche di risorsa utilizzate per la definizione dei parametri SLA presenti nell'Accordo di Servizio corrispondente all'occorrenza di servizio in erogazione ("BasicMetric").

Metodo: update_service_term_state	
Descrizione	Metodo per l'aggiornamento dello stato del servizio erogato in riferimento ad uno specifico Accordo di Servizio.

SPECIFICHE TECNICHE – v3.0

	Attraverso questo metodo il servizio erogatore, mediante il Sistema di Monitoraggio, rende accessibile al servizio fruitore il suo stato in concomitanza con l'erogazione del servizio medesimo.
INPUT	Il metodo riceve in input: <ul style="list-style-type: none"> • Identificatore del Servizio • Identificatore del Fruitore • Identificatore dell'Erogatore • Stato del Servizio (il valore dello stato deve essere uno di quelli presenti nella tabella ICAR_INF2_SERVICE_CATALOG_STATE).
OUTPUT	Il metodo restituisce come output una coppia di valori (codice + descrizione) che descrivono l'esito positivo o negativo dell'operazione.
LOGICA	Il Sistema di Tracciatura aggiorna il campo INF2_ID_STATE della tabella ICAR_INF2_SERVICE per il record avente chiave (INF2_ID_SERVICE + INF2_INITIATOR + INF2_RESPONDER) uguale ai primi tre parametri forniti in input e campo INF2_ACTIVE = TRUE.

Metodo: insert_service_basic_metric	
Descrizione	Metodo per l'inserimento delle misure relative alle metriche di risorsa utilizzate nell'Accordo di Servizio corrispondente al servizio in erogazione ("BasicMetric"). Attraverso questo metodo il servizio erogatore, mediante il Sistema di Monitoraggio, rende accessibile al servizio fruitore lo stato dei parametri SLA ("ServiceLevelObjective") presenti nell'Accordo di Servizio di riferimento.
INPUT	Il metodo riceve in input: <ul style="list-style-type: none"> • Identificatore del Servizio • Identificatore del Fruitore • Identificatore dell'Erogatore • Nome della BasicMetrics • Data e ora Erogazione del Servizio • Valore della BasicMetrics
OUTPUT	Il metodo restituisce come output una coppia di valori (codice + descrizione) che descrivono l'esito positivo o negativo dell'operazione.
LOGICA	Il Sistema di Tracciatura inserisce nella tabella ICAR_INF2_SLA_OBJECT_TRACE un record avente chiave (INF2_ID_SERVICE + INF2_INITIATOR + INF2_RESPONDER +

	INF2_SLA_BASIC_METRIC + INF2_SLA_BASIC_METRIC_DATE_INSERT) e valore (INF2_SLA_BASIC_METRIC_VALUE) corrispondenti ai parametri forniti in input. L'inserimento è subordinato alla presenza di un'occorrenza di servizio attiva nella tabella ICAR_INF2_SERVICE avente chiave (INF2_ID_SERVICE + INF2_INITIATOR + INF2_RESPONDER).
--	--

Si riporta qui di seguito un esempio di utilizzo del Sistema di Tracciatura da parte di un servizio erogatore:

```

.....
// Creazione oggetto Sistema di Tracciatura
TraceSystem ts = new TraceSystem ();

// Aggiornamento dello stato del servizio
ts.update_service_term_state(Id_Service,IdFruitore,IdResponder,StateService.READY_PROCESSI
NG);

// ESECUZIONE DELLA RICHIESTA DA PARTE DEL SERVIZIO EROGATORE
.....

// Inserimento delle BasicMetrics sul Sistema di Tracciatura
Date date = new Date();
ts.insert_service_basic_metric(Id_Service,IdFruitore, IdResponder,"tempoRisposta",10,date);
ts.insert_service_basic_metric(Id_Service,IdFruitore,
IdResponder,"numeroRecordRicevuti",100,date);
ts.insert_service_basic_metric(Id_Service,IdFruitore, IdResponder,"numeroRecordScartati",
0,date);
ts.insert_service_basic_metric(Id_Service,IdFruitore, IdResponder,"numeroRecordInErrore",
5,date);
ts.insert_service_basic_metric(Id_Service,IdFruitore, IdResponder,"numeroRecordRestituiti",
5,date);

// Aggiornamento dello stato del servizio
ts.update_service_term_state(Id_Service,IdFruitore, IdResponder ,StateService.READY_IDLE);
.....
    
```

Infine, qualora il Sistema di Tracciatura dovesse essere dotato di interfacce di tipo Web Service, queste saranno le seguenti.

Il Sistema di Tracciatura espone un oggetto (Web Service) **TraceSystem** dotato di due metodi:

- **update_service_term_state**
- **insert_service_basic_metric**

La logica di questi due metodi è la medesima utilizzata negli approcci precedenti, ma invece di presentare un'interfaccia JAVA o SQL espone al servizio erogatore un'interfaccia WSDL.

WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://sistematracciatura.inf2.icar/" xmlns:ns1="http://sistematracciatura.inf2.icar/"
name="TraceSystem" targetNamespace="http://sistematracciatura.inf2.icar/">
  <types>
    <schema targetNamespace="http://sistematracciatura.inf2.icar/"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/">
      <include schemaLocation="./SistemaTracciatura.xsd"/>
    </schema>
  </types>
  <message name="update_service_term_stateRequest">
    <part name="richiesta" type="ns1:Request_update_service_term_stateType"/>
  </message>
  <message name="update_service_term_stateResponse">
    <part name="risposta" type="ns1:Response_TraceSystemType"/>
  </message>
  <message name="insert_service_basic_metricRequest">
    <part name="richiesta" type="ns1:Request_insert_service_basic_metricType"/>
  </message>
  <message name="insert_service_basic_metricResponse">
    <part name="risposta" type="ns1:Response_TraceSystemType"/>
  </message>
  <portType name="TraceSystemPortType">
    <operation name="update_service_term_state">
      <input name="update_service_term_stateRequest"
message="tns:update_service_term_stateRequest"/>
      <output name="update_service_term_stateResponse"
message="tns:update_service_term_stateResponse"/>
    </operation>
    <operation name="insert_service_basic_metric">
      <input name="insert_service_basic_metricRequest"
message="tns:insert_service_basic_metricRequest"/>
      <output name="insert_service_basic_metricResponse"
message="tns:insert_service_basic_metricResponse"/>
    </operation>
  </portType>
  <binding name="TraceSystemBinding" type="tns:TraceSystemPortType">
    <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="update_service_term_state">
      <soap:operation soapAction="" style="rpc"/>
      <input name="update_service_term_stateRequest">
        <soap:body use="literal" namespace="http://icar.inf2.sistematracciatura"/>
      </input>
    </operation>
  </binding>
</definitions>
```

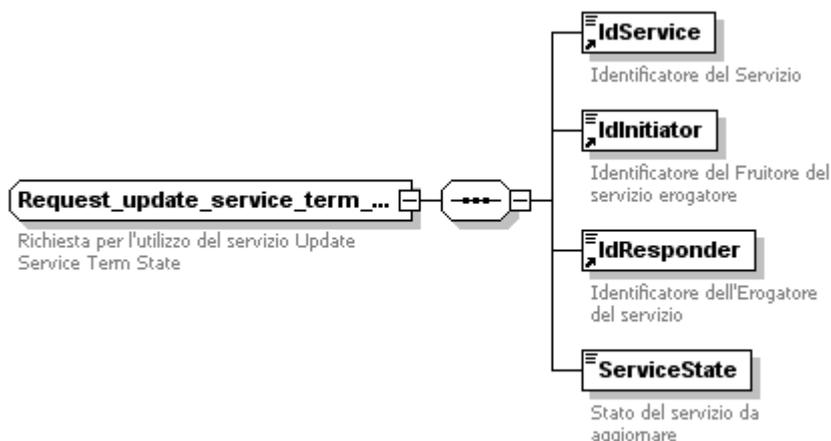
```

<output name="update_service_term_stateResponse">
  <soap:body use="literal" namespace="http://icar.inf2.sistematracciatura"/>
</output>
</operation>
<operation name="insert_service_basic_metric">
  <soap:operation soapAction="" style="rpc"/>
  <input name="insert_service_basic_metricRequest">
    <soap:body use="literal" namespace="http://icar.inf2.sistematracciatura"/>
  </input>
  <output name="insert_service_basic_metricResponse">
    <soap:body use="literal" namespace="http://icar.inf2.sistematracciatura"/>
  </output>
</operation>
</binding>
<service name="TraceSystem">
  <port name="TraceSystem" binding="tns:TraceSystemBinding">
    <soap:address location="TBD"/>
  </port>
</service>
</definitions>

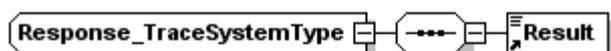
```

I tipi citati nel WSDL sono definiti dello schema SistemaTracciatura.xsd allegato. Il metodo **update_service_term_state** riceve in input un oggetto di tipo **Request_update_service_term_stateType** e restituisce un oggetto di tipo **Response_TraceSystemType**.

L'oggetto **Request_update_service_term_stateType** nello schema Sistema di Tracciatura è così definito:



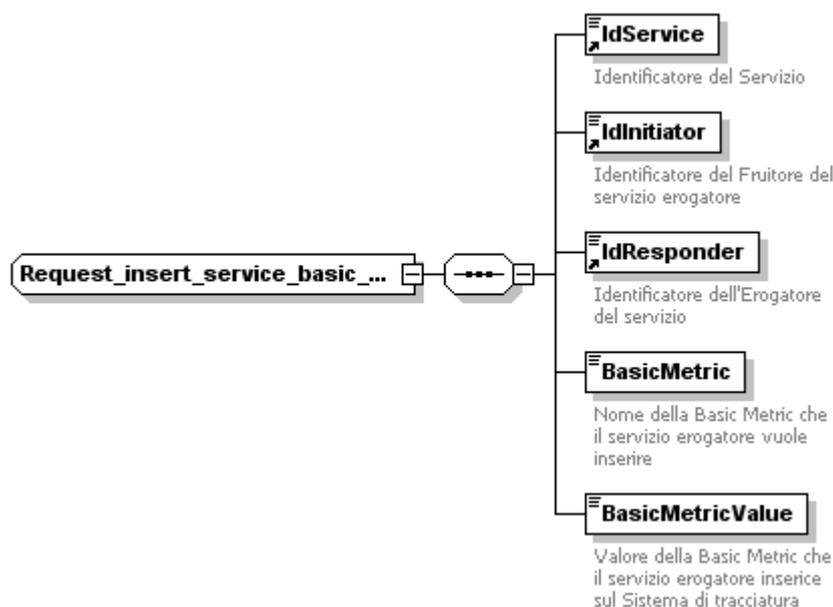
L'oggetto **Response_TraceSystemType** nello schema Sistema di Tracciatura è così definito:



Dove il nodo **Result** contiene una coppia di valori che descrive l'esito positivo o negativo dell'operazione codice numerico

Il metodo `insert_service_basic_metric` riceve in input un oggetto di tipo `Request_insert_service_basic_metricType` e restituisce un oggetto di tipo `Response_TraceSystemType`.

L'oggetto `Request_insert_service_basic_metricType` nello schema Sistema di Tracciatura è così definito:



6.2.3 Interfaccia Operatore - Console di monitoraggio

Poiché sia il sistema di tracciatura, sia il sistema di monitoraggio prevedono la realizzazione di un'interfaccia operatore, si è deciso di produrre un'unica "console di monitoraggio" che comprenderà le funzionalità elencate nei paragrafi 6.2 e 6.3, ovvero:

- inserimento delle occorrenze di servizi che, secondo l'AdS, devono fornire dei parametri SLA. Attivazione/disattivazione della tracciatura relativa ad un'occorrenza di servizio;
- monitorare lo stato delle occorrenze di servizio iscritte al Sistema di Tracciatura;
- monitorare, per ogni occorrenza di servizio, lo stato dei parametri SLA definiti nell'AdS attraverso opportune rappresentazioni grafiche;

E' opportuno sottolineare la peculiare caratteristica attribuita alla prima funzionalità di cui sarà dotato questo sottosistema. La funzionalità di inserimento delle occorrenze di servizio soggette a monitoraggio provocherà l'inserimento di un record sulla tabella "ICAR_INF2_SERVICE" per ognuna delle occorrenze medesime.

Contestualmente il campo “INF2_ID_SLA_OBJECT” sarà riempito col contenuto relativo al monitoraggio SLA dell’accordo di servizio pattuito per il servizio stesso, onde potere, in fase di monitoraggio, eseguire gli algoritmi previsti senza accedere reiteratamente ai registri SICA.

Dunque i momenti di pubblicazione della parte specifica degli accordi di servizio e di inserimento dei dati corrispondenti nel sistema di tracciatura risultano distinti.

La tracciatura delle metriche di risorsa associate ad un’occorrenza di servizio viene effettuata solo per le erogazioni che avvengono dopo l’inserimento del record relativo sul DBWSA.

A fronte della necessità di un doppio intervento degli operatori, si ottiene quindi la possibilità di differire l’inizio del monitoraggio dei parametri SLA di un servizio rispetto al momento della pubblicazione dell’accordo di servizio che lo norma, consentendo così di dare il via al monitoraggio dopo un primo periodo di erogazione che potrebbe considerarsi di “avvio e test”.

L’ulteriore funzionalità di attivazione/disattivazione è stata sviluppata per consentire di sospendere, eventualmente in modo temporaneo, la tracciatura di un’occorrenza di servizio (per esempio quando l’Accordo di Servizio ad essa relativa viene storicizzato).

Per mantenere unita la descrizione dell’intera interfaccia operatore, e renderne quindi più semplice la consultazione, questo paragrafo riporta anche le specifiche tecniche delle funzionalità afferenti al sistema di monitoraggio (secondo e terzo punto).

Per migliorare l’utilizzo delle interfacce operatore fornite dai task INF-1 e INF-2, le due console sono state integrate dotandole di un unico entry point.

Tale punto d’ingresso prevede l’inserimento da parte dell’operatore di una coppia utente/password, che viene intercettata dalla console INF-1.

A seconda del profilo dell’operatore, questi viene indirizzato sulle maschere operative della console INF-1 stessa (profilo “amministratore”) o su quelle della console INF-2 (profilo “operatore”).

Gli utenti di tipo “amministratore” potranno in seguito, dalla console INF-1, accedere direttamente alla console INF-2 avendo a disposizione tutte le funzionalità esistenti.

Gli utenti di tipo “operatore” sono invece caratterizzati dall’abilitazione ad operare solo sui servizi che coinvolgono, come erogatori o fruitori, uno o più soggetti (contenuti nella tabella “nica_registro.soggetti”) per i quali sono stati esplicitamente abilitati. Al momento dell’accesso alla console INF-2 le informazioni necessarie ad individuare tali soggetti vengono comunicate in background alla console stessa.

Le funzionalità presenti nella console INF-2, saranno esposte agli utenti suddivise in due parti distinte: “Monitoraggio servizi erogati” e “Monitoraggio servizi fruiti”.

Monitoraggio servizi erogati

In questa sezione gli operatori potranno effettuare le seguenti operazioni:

- dare inizio alla tracciatura di occorrenze di servizio (sistema di tracciatura).

Per effettuare questa operazione, riservata agli operatori con profilo “amministratore”, è necessario scegliere una delle terne servizio - erogatore - fruitore presentate (corrispondenti a tutte le occorrenze di servizio erogate dal dominio di appartenenza) e selezionare la voce “aggiungi”. L’associazione del file xml non è più necessaria, poiché viene effettuata una lettura dell’opportuno campo (wsag_sla) della tabella nica_registro.servizi_fruitori, facente parte dell’implementazione del NICA di INF-1.

Le tracciatore dell’occorrenza di servizio risultano così abilitate.

- attivare/disattivare la tracciatura di occorrenze di servizio (sistema di tracciatura).

Queste due operazioni, di utilizzo saltuario e riservate agli operatori con profilo “amministratore”, servono a sospendere la tracciatura di un’occorrenza di servizio quando l’accordo di servizio relativo cessa di essere valido o a riprendere la tracciatura precedentemente sospesa qualora la sospensione fosse stata erronea. Le operazioni si effettuano selezionando, rispettivamente, le voci “disattiva” o “attiva”.

- monitorare lo stato di occorrenze di servizio (sistema di monitoraggio).

Questa operazione consente di monitorare lo stato di un’occorrenza di servizio erogata dal dominio di appartenenza e già attivato (cfr. punti precedenti). La logica di interrogazione ed i risultati mostrati sono quelli descritti al sottoparagrafo 6.3.1 “ServiceTermState”.

Gli utenti con profilo “operatore” visualizzeranno i dati relativi a servizi aventi soggetti erogatori per i quali essi sono stati esplicitamente abilitati.

Per ottenere le informazioni occorre selezionare la voce “monitoraggio stato”. Vengono mostrate le terne servizio - erogatore - fruitore corrispondenti a tutte le occorrenze di servizio erogate dal dominio di appartenenza. I dati “stato” e “richieste in fase di elaborazione” rappresentano lo stato dell’erogazione del servizio ad un determinato fruitore (“READY_PROCESSING” se in fase di erogazione, “READY_IDLE” se in attesa) e il numero di erogazioni concorrenti in corso al medesimo fruitore.

- monitorare il rispetto dei parametri SLA pattuiti negli Accordi di Servizio (sistema di monitoraggio).

Questa operazione consente di verificare il rispetto dei parametri SLA associati ad un’occorrenza di servizio erogata dal dominio di appartenenza. La logica di interrogazione ed i risultati mostrati sono quelli descritti al sottoparagrafo 6.3.2 “ServiceGuaranteeState”.

Gli utenti con profilo “operatore” visualizzeranno i dati relativi a servizi aventi soggetti erogatori per i quali essi sono stati esplicitamente abilitati.

Per ottenere le informazioni occorre selezionare la voce “monitoraggio SLA servizi erogati”, quindi “verifica SLA”, inserire i parametri SLA che si desidera osservare mediante la voce “aggiungi SLA” e provocare la verifica selezionando la voce “verifica SLA”.

Vengono mostrati tutti i parametri SLA richiesti, e, per ognuno di questi, il risultato della verifica (“FULFILLED” se è stato rispettato quanto pattuito nell’AdS, “VIOLATED” in caso contrario, “NOT DETERMINED” se non ci sono tracciatore sufficienti per la valutazione). In caso di

parametro SLA “FULFILLED” o “VIOLATED”, vengono altresì mostrati i valori di soglia concordati ed il valore attuale calcolato.

- visualizzare statistiche sui parametri SLA pattuiti negli Accordi di Servizio (sistema di monitoraggio).

Questa operazione consente di visualizzare graficamente delle statistiche personalizzate sul rispetto dei parametri SLA associati ad un’occorrenza di servizio erogata dal dominio di appartenenza.

Gli utenti con profilo “operatore” visualizzeranno i dati relativi a servizi aventi soggetti erogatori per i quali essi sono stati esplicitamente abilitati.

Occorre selezionare la voce “monitoraggio SLA servizi erogati”, quindi “statistiche SLA”, selezionare il parametro SLA che si desidera osservare ed inserire periodo e frequenza di tali osservazioni. Il grafico viene generato dopo la selezione della voce “visualizza”.

Va notato che il periodo (determinato da “data inizio periodo” e “data fine periodo”) e la frequenza (determinata da “granularità della rilevazione”) della rilevazione statistica non hanno nessuna relazione con la finestra (temporale o quantitativa) di osservazione all’interno della quale reperire i dati per il calcolo del parametro SLA. Impostare, per esempio, come periodo un anno e come granularità un mese, significa provocare 13 calcoli del parametro SLA sottostante, aventi date di fine osservazione distanziate di un mese (con data di fine osservazione massima uguale a “data di fine periodo”). I tredici valori del parametro SLA così ottenuti saranno utilizzati nella rappresentazione grafica (istogramma con valore di soglia o areogramma a torta).

Monitoraggio servizi fruiti

In questa sezione gli operatori potranno effettuare le seguenti operazioni:

- monitorare lo stato di occorrenze di servizio (sistema di monitoraggio).

Questa operazione consente di monitorare lo stato di un’occorrenza di servizio fruita dal dominio di appartenenza. La logica di interrogazione ed i risultati mostrati sono quelli descritti al sottoparagrafo 6.3.1 “ServiceTermState” ed il reperimento dei dati avviene mediante l’utilizzo di tale web service messo a disposizione dal dominio che eroga il servizio.

Gli utenti con profilo “operatore” visualizzeranno i dati relativi a servizi aventi soggetti fruitori per i quali essi sono stati esplicitamente abilitati.

Per ottenere le informazioni occorre selezionare la voce “monitoraggio stato e parametri SLA”. Vengono mostrate le terne servizio - erogatore - fruitore corrispondenti a tutte le occorrenze di servizio fruita dal dominio di appartenenza. Selezionare la voce “monitoraggio stato”, impostare il nome del soggetto erogatore del servizio di monitoraggio (deve essere un soggetto del dominio che eroga il servizio da monitorare) e selezionare “verifica stato”. Il dato “stato” che sarà visualizzato rappresenta lo stato dell’erogazione del servizio al dominio di appartenenza (“READY_PROCESSING” se in fase di erogazione, “READY_IDLE” se in attesa).

- monitorare il rispetto dei parametri SLA pattuiti negli Accordi di Servizio (sistema di monitoraggio).

Questa operazione consente di verificare il rispetto dei parametri SLA associati ad un'occorrenza di servizio fruito dal dominio di appartenenza. La logica di interrogazione ed i risultati mostrati sono quelli descritti al sottoparagrafo 6.3.2 “ServiceGuaranteeState” ed il reperimento dei dati avviene mediante l'utilizzo di tale web service messo a disposizione dal dominio che eroga il servizio.

Gli utenti con profilo “operatore” visualizzeranno i dati relativi a servizi aventi soggetti fruitori per i quali essi sono stati esplicitamente abilitati.

Per ottenere le informazioni occorre selezionare la voce “monitoraggio stato e parametri SLA”, impostare il nome del soggetto erogatore del servizio di monitoraggio (deve essere un soggetto del dominio che eroga il servizio da monitorare), inserire i parametri SLA che si desidera osservare mediante la voce “aggiungi SLA” e provocare la verifica selezionando la voce “verifica SLA”.

Vengono mostrati tutti i parametri SLA richiesti, e, per ognuno di questi, il risultato della verifica (“FULFILLED” se è stato rispettato quanto pattuito nell'AdS, “VIOLATED” in caso contrario, “NOT DETERMINED” se non ci sono tracciature sufficienti per la valutazione). A regime, in caso di parametro SLA “FULFILLED” o “VIOLATED”, verranno altresì mostrati i valori di soglia concordati ed il valore attuale calcolato.

- visualizzare statistiche sui parametri SLA pattuiti negli Accordi di Servizio (sistema di monitoraggio).

Questa operazione consente di visualizzare graficamente delle statistiche personalizzate sul rispetto dei parametri SLA associati ad un'occorrenza di servizio fruito dal dominio di appartenenza.

Gli utenti con profilo “operatore” visualizzeranno i dati relativi a servizi aventi soggetti fruitori per i quali essi sono stati esplicitamente abilitati.

Occorre selezionare la voce “visualizza statistiche”, selezionare il parametro SLA che si desidera osservare, impostare il nome del soggetto erogatore del servizio di monitoraggio (deve essere un soggetto del dominio che eroga il servizio da monitorare) ed inserire periodo e frequenza di tali osservazioni. Il grafico viene generato dopo la selezione della voce “visualizza”.

Va notato che il periodo (determinato da “data inizio periodo” e “data fine periodo”) e la frequenza (determinata da “granularità della rilevazione”) della rilevazione statistica non hanno nessuna relazione con la finestra (temporale o quantitativa) di osservazione all'interno della quale reperire i dati per il calcolo del parametro SLA. Impostare, per esempio, come periodo un anno e come granularità un mese, significa provocare 13 calcoli del parametro SLA sottostante, aventi date di fine osservazione distanziate di un mese (con data di fine osservazione massima uguale a “data di fine periodo”). I tredici valori del parametro SLA così ottenuti saranno utilizzati nella rappresentazione grafica (istogramma con valore di soglia o areogramma a torta).

6.3 Specifica del sistema di monitoraggio

Il sistema di monitoraggio qui descritto prevede i seguenti sotto componenti:

- **Interfaccia applicativo del Sistema di Monitoraggio** - due web services, esposti tramite la Porta di Dominio:
 - **ServiceTermState**: permette di ricavare lo stato di un servizio relativo ad una coppia fruitore erogatore;
 - **ServiceGuaranteeTerm**: permette di ricavare lo stato di uno o più parametri SLA (“ServiceLevelObjective”) per un’occorrenza di servizio relativa ad un determinato Accordo di Servizio.
- **Interfaccia Operatore** - (descritta nel sottoparagrafo 6.2.3) per consentire interattivamente di:
 - monitorare lo stato di occorrenze di servizi;
 - monitorare lo stato degli SLA ad esse associati.

6.3.1 ServiceTermState

Questo servizio espone il metodo (Web Service) **getServiceTermState** che riceve in input un oggetto di tipo RequestServiceTermStateType e restituisce un oggetto di tipo ResponseServiceTermStateType.

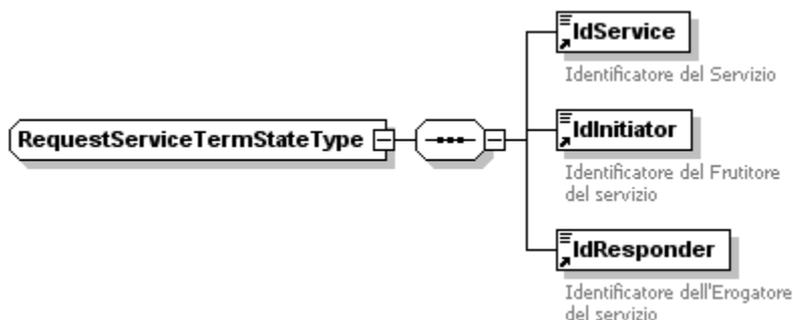
Questi tipi sono definiti nello schema “SistemaMonitoraggio.xsd” in allegato.

Richiesta del Servizio

Il servizio riceve un oggetto di tipo RequestServiceTermStateType che contiene:

- l’identificatore del Servizio (nodo **IdService**)
- l’identificatore del Fruitore (nodo **IdInitiator**)
- l’identificatore dell’Erogatore (nodo **IdResponder**)

Viene riportato di seguito lo schema XML che descrive un tipo RequestServiceTermStateType:



Logica del Servizio

Il servizio prende in carico la richiesta ed esegue i seguenti passi:

1. lettura dei parametri applicativi contenuti nell’oggetto ServiceTermStateRequest;
2. lettura dello stato del servizio nella tabella ICAR_INF2_SERVICE;

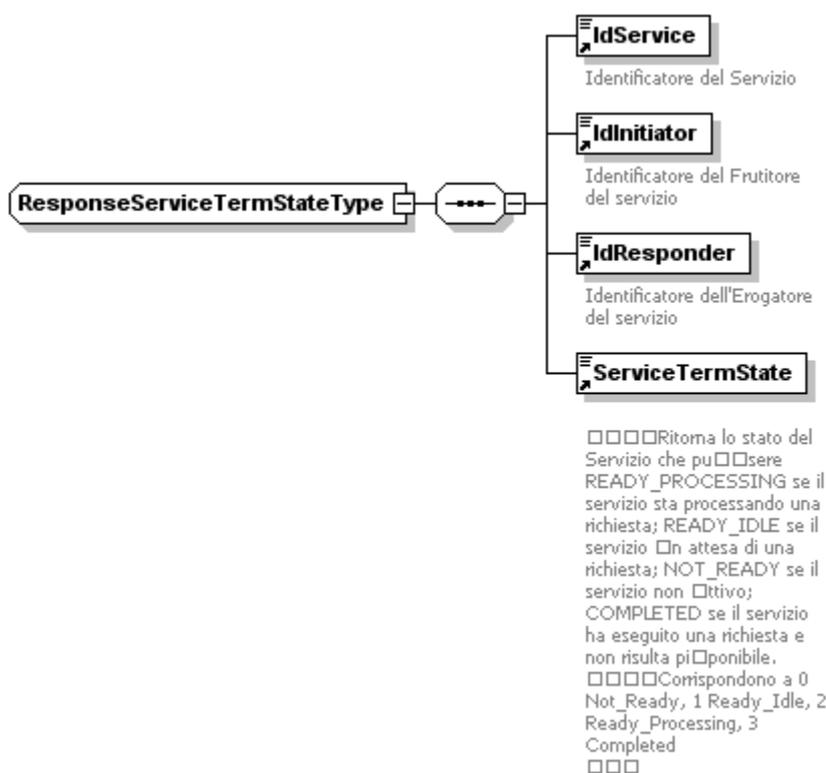
3. creazione dell'oggetto ServiceTermStateResponse;
4. invio dell'oggetto al chiamante.

Risposta del Servizio

Il servizio risponde inviando un oggetto di tipo **ResponseServiceTermStateType** che contiene:

- l'identificatore del Fruitore (nodo **IdInitiator**)
- l'identificatore dell'Erogatore (nodo **IdResponder**)
- l'identificatore del Servizio (nodo **IdService**)
- un oggetto **ServiceTermState** di tipo **ServiceTermStateType**

Lo schema xsd del tipo ResponseServiceTermType è il seguente:



Di seguito viene riportato il tipo ServiceTermState:

```
<xs:simpleType name="ServiceTermStateType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Ready_Processing"/>
    <xs:enumeration value="Ready_Idle"/>
    <xs:enumeration value="Not_Ready"/>
    <xs:enumeration value="Completed"/>
  </xs:restriction>
</xs:simpleType>
```

Accordo di Servizio

Per il servizio ServiceTermState è richiesta la standardizzazione secondo le direttive SPCoop delle interfacce messe a disposizione. La formalizzazione avviene all'interno dell'Accordo di Servizio associato.

Un Accordo di Servizio è in generale composto da una parte comune e tante parti specifiche quante sono le coppie di erogatori e fruitori associate al servizio sottostante. Per la natura del sistema di monitoraggio, il servizio ServiceTermState darà origine ad un AdS mono-erogatore / multi-fruitor.

Inoltre l'approccio scelto per la definizione della parte comune e delle parti specifiche dell'AdS è quello "unilaterale", per cui l'erogatore emana la parte comune dell'accordo e stila una sorta di template della parte specifica. Tale template, completo di tutto tranne l'esplicita indicazione del fruitore, costituisce guida per la creazione delle singole parti specifiche, che verranno registrate quando un fruitore desidererà usufruire effettivamente dei servizi resi disponibili dal sistema di monitoraggio del dominio desiderato.

Questa impostazione discende dal fatto che i servizi erogati dal sistema di monitoraggio medesimo saranno assolutamente identici per tutti i fruitori. In particolare, non si ritiene di dover associare a questi servizi garanzie su parametri SLA connessi all'erogazione stessa.

Di seguito si accludono i cinque file necessari per la composizione dell'Accordo di Servizio associato a ServiceTermState.

Parte comune - Specifica dell'Interfaccia - WSDL concettuale: descrizione WSDL a livello di scenario di coordinamento.



ServiceTermStateConcettuale.wsd

Parte comune - Specifica dell'Interfaccia - WSDL logico dell'erogatore: descrizione WSDL dell'erogatore a livello di scambio elementare di messaggi.



ServiceTermStateLogico.wsd

Parte comune - Specifica dell'Interfaccia - WSDL logico del fruitore: descrizione WSDL del fruitore a livello di scambio elementare di messaggi.

Omesso perché si tratta di servizio sincrono, per il quale non è prevista la presenza di web service lato fruitore. Per conformità con le indicazioni CNIPA, si includerà nell'AdS un file vuoto.

Parte specifica - Specifica Porti di Accesso - WSDL implementativo dell'erogatore: descrizione WSDL dell'erogatore a livello implementativo, comprendente URI e binding delle operazioni.



ServiceTermState.ws
dl

Parte specifica - Specifica Porti di Accesso - WSDL implementativo del fruitore: descrizione WSDL del fruitore a livello implementativo, comprendente URI e binding delle operazioni. Essendo un template, ci saranno dei “segnaposto” da compilare a cura dell'effettivo fruitore.

Omesso perché si tratta di servizio sincrono, per il quale non è prevista la presenza di web service lato fruitore. Per conformità con le indicazioni CNIPA, si includerà nell'AdS un file vuoto.

Caso d'uso

Il servizio ServiceTermState utilizzerà la modalità trasparente per integrazione con la Porta di Dominio sviluppata nel task infrastrutturale INF1.

Di seguito viene riportato un esempio di utilizzazione del Web Service

```
try {  
  
    // Istanziamento oggetto remoto ServiceTermState  
    ServiceTermStateLocator locator = new ServiceTermStateLocator();  
    String url_endpoint= http://icar/inf2/PD/getServiceTermState;  
    locator.setServiceTermStateEndpointAddress(url_endpoint);  
    ServiceTermState port = locator.getServiceTermState();  
  
    // Creazione dell'oggetto Richiesta  
    RequestServiceTermStateType request = new RequestServiceTermStateType();  
    // inserimento dei parametri dell'oggetto richiesta  
    request.setIdService("SERVIZIO1");  
    .....  
    // Chiamata dell'oggetto remoto  
    ResponseServiceTermStateType response = port.getServiceTermState(request);  
  
    //lettura dello stato del servizio  
    ServiceTermStateType stato = response.getServiceTermState();  
  
} catch (AxisFault e) {  
    System.out.println("Ricevuto SOAPFault applicativo");  
    System.out.println("Actor: "+e.getFaultActor());  
    System.out.println("Code: "+e.getFaultCode());  
    System.out.println("String: "+e.getFaultString());  
} catch (Exception e) {  
    System.out.println("ClientError: "+e.getMessage());  
    e.printStackTrace();  
}
```

6.3.2 ServiceGuaranteeTermState

Questo servizio espone il metodo (Web Service) **getServiceGuaranteeTermState** che riceve in input un oggetto di tipo **RequestGuaranteeTermStateType** e restituisce un oggetto di tipo **ResponseGuaranteeTermStateType**.

Il WSDL che identifica questo servizio è il seguente:

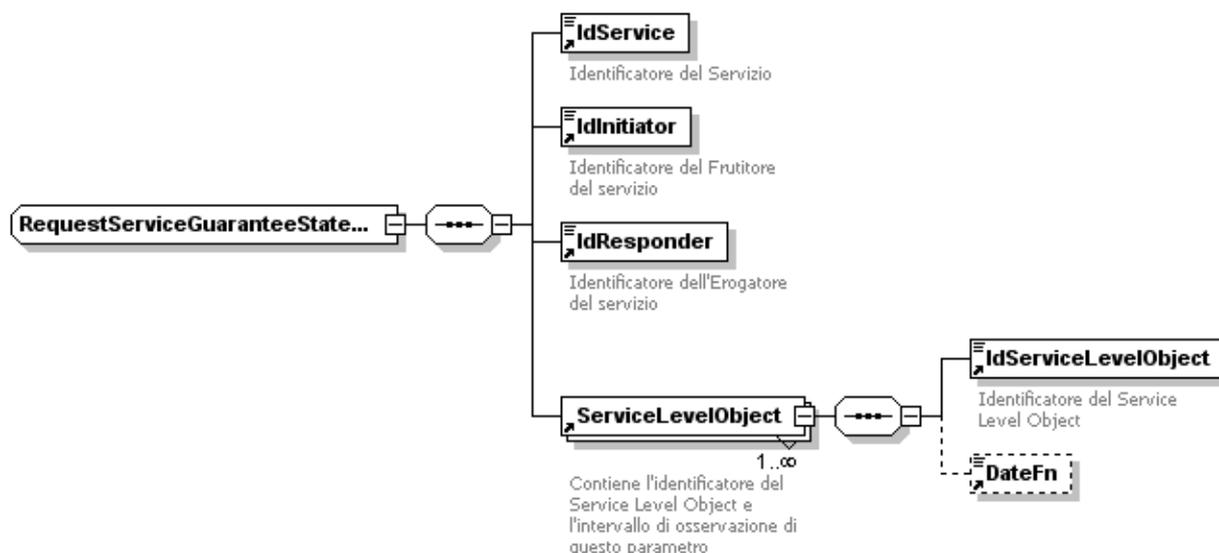
Questi tipi sono definiti nello schema “SistemaMonitoraggio.xsd” allegato.

Richiesta del Servizio

Il servizio riceve un oggetto di tipo **RequestGuaranteeTermStateType** che contiene:

- l'identificatore del Servizio (nodo **IdService**)
- l'identificatore del Fruitore (nodo **IdInitiator**)
- l'identificatore dell'Erogatore (nodo **IdResponder**)
- una lista **ServiceLevelObjective** che è composta da:
 - l'identificatore del Service Level Objective
 - una data di fine osservazione (**opzionale**)

Viene riportato di seguito lo schema XML che descrive un tipo **RequestGuaranteeTermStateType**:



Logica del Servizio

Il servizio prende in carico la richiesta ed esegue i seguenti passi:

1. lettura dei parametri applicativi contenuti nell'oggetto `GuaranteeTermStateRequest`;
2. lettura del campo **INF2_SLA_OBJECT**, interrogando la tabella **ICAR_INF2_SERVICE** con chiave uguale a `Identificatore del Servizio + Identificatore del Fruitore + Identificatore dell'Erogatore`. In questo modo si ricava:
 - l'algoritmo per il calcolo dei parametri SLA richiesti;
 - la finestra temporale o il numero di occorrenze su cui applicare l'algoritmo;
 - la soglia per determinare lo stato del Service Level Object.

(alternativa: interrogare il repository degli accordi di servizio)

3. lettura del campo `INF2_SLA_BASIC_METRIC_VALUE`, interrogando la tabella `ICAR_INF2_SLA_OBJECT_TRACE` con parte di chiave uguale a `Identificatore di Servizio + Identificatore del Fruitore + Identificatore dell'Erogatore + ogni nome di BasicMetric citata nell'algoritmo di calcolo e una finestra osservazione`, atta ad individuare un sottoinsieme dei record presenti nella tabella per la parte di chiave fornita.

Se nell'Accordo di servizio la condizione di osservazione ("Window") è tipo "Interval" e nella richiesta è presente la data di fine osservazione, la finestra osservazione sarà data da

- $\text{Data Iniziale} = \text{Data Finale della Richiesta} - \text{Intervallo di tempo definito nell'AdS}$
- $\text{Data Finale} = \text{Data Finale della Richiesta}$

Altrimenti, se non è specificata la data finale:

- $\text{Data Iniziale} = \text{Data di Sistema} - \text{Intervallo di tempo definito nel Ads}$
- $\text{Data Finale} = \text{Data di Sistema}$

Se nell'Accordo di servizio la condizione di osservazione ("Window") è tipo "Times" e nella richiesta è presente la data di fine osservazione, la finestra osservazione sarà data da:

- Numero di occorrenze dei parametri Basic Metric
- $\text{Data Finale} = \text{Data Finale della Richiesta}$

Altrimenti, se non è specificata la data finale:

- Numero di occorrenze dei parametri Basic Metric
- $\text{Data Finale} = \text{Data di Sistema}$

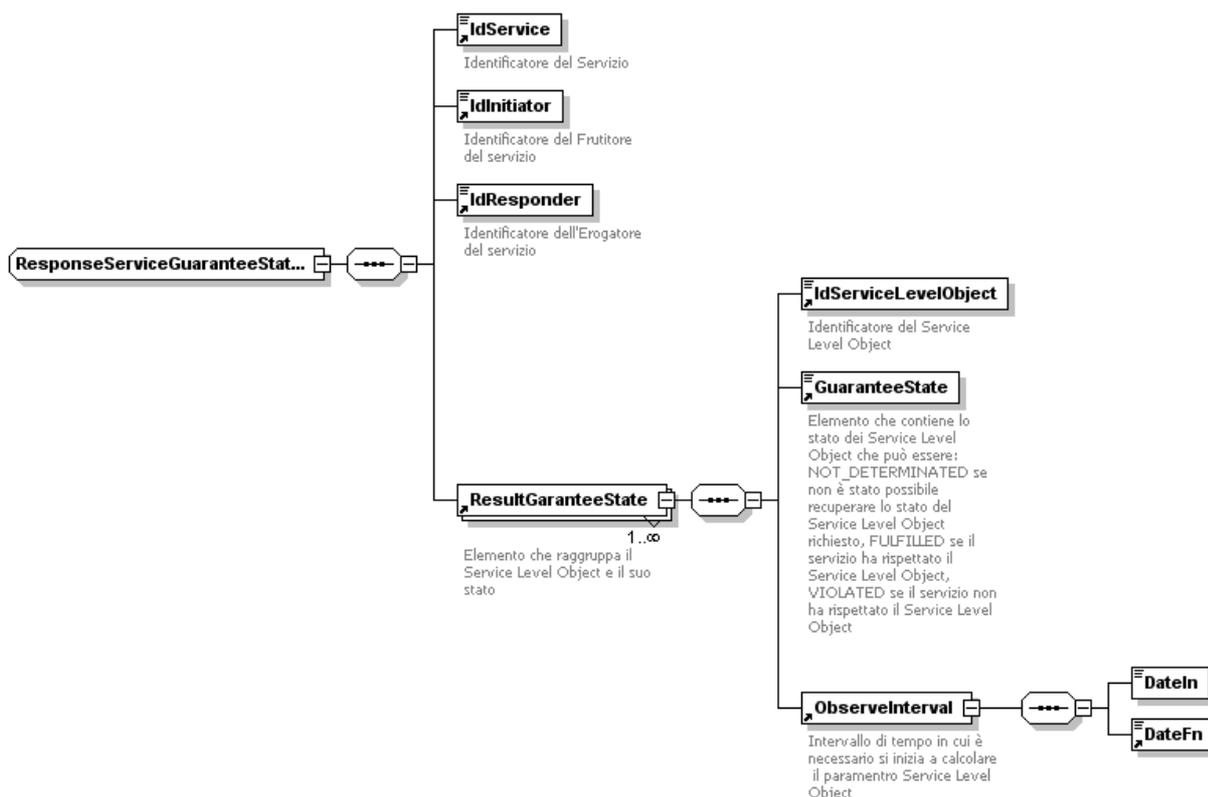
4. calcolo dello stato di ogni Service Level Object richiesto;
5. creazione dell'oggetto `ServiceGuaranteeStateResponse`;
6. invio dell'oggetto al chiamante.

Risposta del Servizio

Il metodo `getServiceGuaranteeTermState` (Web Service) invia al chiamante un oggetto di tipo `ResponseServiceGuaranteeTermStateType` che contiene:

- l'identificatore del Servizio (nodo **IdService**)
- l'identificatore del Fruitore (nodo **IdInitiator**)
- l'identificatore dell'Erogatore (nodo **IdResponder**)
- una lista di oggetti **ResultGaranteeState** che contengono:
 - l'identificatore del Service Level Object (**IdServiceLevelObject**);
 - un oggetto **GuaranteeState** di tipo **GuaranteeStateType**;
 - un oggetto **ObserveInterval** che indica il periodo temporale di osservazione.

Lo schema xsd del tipo `ResponseServiceGuaranteeTermStateType` è il seguente:



Di seguito viene riportato il tipo **GuaranteeStateType**:

```
<xs:simpleType name="GuaranteeStateType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="NotDetermined"/>
    <xs:enumeration value="FulFilled"/>
    <xs:enumeration value="Violated"/>
  </xs:restriction>
</xs:simpleType>
```

Accordo di Servizio

Per il servizio ServiceGuaranteeTermState è richiesta la standardizzazione secondo le direttive SPCoop delle interfacce messe a disposizione. La formalizzazione avviene all'interno dell'Accordo di Servizio associato.

Un Accordo di Servizio è in generale composto da una parte comune e tante parti specifiche quante sono le coppie di erogatori e fruitori associate al servizio sottostante. Per la natura del sistema di monitoraggio, il servizio ServiceTermState darà origine ad un AdS mono-erogatore / multi-fruitori.

Inoltre l'approccio scelto per la definizione della parte comune e delle parti specifiche dell'AdS è quello "unilaterale", per cui l'erogatore emana la parte comune dell'accordo e stila una sorta di template della parte specifica. Tale template, completo di tutto tranne l'esplicita indicazione del fruitore, costituisce guida per la creazione delle singole parti specifiche, che verranno registrate quando un fruitore desidererà usufruire effettivamente dei servizi resi disponibili dal sistema di monitoraggio del dominio desiderato.

Questa impostazione discende dal fatto che i servizi erogati dal sistema di monitoraggio medesimo saranno assolutamente identici per tutti i fruitori. In particolare, non si ritiene di dover associare a questi servizi garanzie su parametri SLA connessi all'erogazione stessa.

Di seguito si accludono i cinque file necessari per la composizione dell'Accordo di Servizio associato a ServiceGuaranteeTermState.

Parte comune - Specifica dell'Interfaccia - WSDL concettuale: descrizione WSDL a livello di scenario di coordinamento.



ServiceGuaranteeTermStateConcettuale.w

Parte comune - Specifica dell'Interfaccia - WSDL logico dell'erogatore: descrizione WSDL dell'erogatore a livello di scambio elementare di messaggi.



ServiceGuaranteeTermStateLogico.wsdl

Parte comune - Specifica dell'Interfaccia - WSDL logico del fruitore: descrizione WSDL del fruitore a livello di scambio elementare di messaggi.

Omesso perché si tratta di servizio sincrono, per il quale non è prevista la presenza di web service lato fruitore. Per conformità con le indicazioni CNIPA, si includerà nell'AdS un file vuoto.

Parte specifica - Specifica Porti di Accesso - WSDL implementativo dell'erogatore: descrizione WSDL dell'erogatore a livello implementativo, comprendente URI e binding delle operazioni.



ServiceGuaranteeTermState.wsdl

Parte specifica - Specifica Porti di Accesso - WSDL implementativo del fruitore: descrizione WSDL del fruitore a livello implementativo, comprendente URI e binding delle operazioni. Essendo un template, ci saranno dei “segnaposto” da compilare a cura dell'effettivo fruitore.

Omesso perché si tratta di servizio sincrono, per il quale non è prevista la presenza di web service lato fruitore. Per conformità con le indicazioni CNIPA, si includerà nell'AdS un file vuoto.

Caso d'uso

Il servizio ServiceGuaranteeState utilizzerà la modalità trasparente per l'integrazione con la Porta di Dominio sviluppata nel task infrastrutturale INF1.

Di seguito viene riportato un esempio di come può essere invocato il Web Service

```
try {
    // Istanzio l'oggetto remoto ServiceTermState
    ServiceGuaranteeStateLocator locator = new ServiceGuaranteeStateLocator();

    String url_endpoint= http://icar/inf2/PD/getServiceGuaranteeTermState;

    locator.setServiceGuaranteeStateEndpointAddress(url_endpoint);
    ServiceGuaranteeState port = locator.getServiceGuaranteeTermState();

    // Creazione dell'oggetto Richiesta
    RequestServiceGuaranteeTermStateType request =
        new RequestServiceGuaranteeTermStateType();
    // inserimento dei parametri dell'oggetto richiesta
    request.setIdService("SERVIZIO1");
    .....

    // Chiamata dell'oggetto remoto
    ResponseServiceGuaranteeTermStateType response =
        port.getServiceGuaranteeTermState(request);

    //lettura dello stato del servizio
    ServiceGuaranteeStateType stato = response.getServiceGuaranteeTermState();
} catch (AxisFault e) {
    System.out.println("Ricevuto SOAPFault applicativo");
    System.out.println("Actor: "+e.getFaultActor());
}
```

```
        System.out.println("Code: "+e.getFaultCode());
        System.out.println("String: "+e.getFaultString());
    } catch (Exception e) {
        System.out.println("ClientError: "+e.getMessage());
        e.printStackTrace();
    }
```

7 BIBLIOGRAFIA

- A. Raffio, C. Tziviskou Service Level Agreement Languages,
<http://www.elet.polimi.it/upload/ghezzi/PRIVATE/SLA-final.pdf>
- AA.VV. (Global Grid Forum) Web Services Agreement Specification (WS-Agreement), 2004-2006
http://forge.ggf.org/sf/docman/do/downloadDocument/projects.graap-wg/docman.root.current_drafts/doc6091
- CNIPA, Sistema pubblico di cooperazione: quadro tecnico d'insieme, 14 Ottobre 2005
- CNIPA, Sistema pubblico di cooperazione: termini e definizioni, 14 Ottobre 2005
- CNIPA, Sistema pubblico di cooperazione: esercizio e gestione, 14 Ottobre 2005
- CNIPA, Sistema pubblico di cooperazione: convenzioni di nomenclatura e semantica, 14 Ottobre 2005
- CNIPA, Sistema pubblico di cooperazione: servizi di sicurezza, 14 Ottobre 2005
- CNIPA, Sistema pubblico di cooperazione: servizi di registro, 14 Ottobre 2005
- CNIPA, Sistema pubblico di cooperazione: accordo di servizio, 14 Ottobre 2005
- CNIPA, Sistema pubblico di cooperazione: porta di dominio, 14 Ottobre 2005
- CNIPA, Sistema pubblico di cooperazione: busta di e-Gov, 14 Ottobre 2005
- Regione Toscana, INF-1: Realizzazione dell'Infrastruttura di base per l'Interoperabilità e la Cooperazione Applicativa a livello interregionale: Linee Architettoniche, versione 1.1.7
- Regione Toscana, INF-1: Specifiche Tecniche di Interfaccia, versione 1.1
- Regione Toscana, INF-1: Specifiche Accordo di Servizio, versione 1.0

ALLEGATI



SistemaTracciaturaN
ew.xsd



SistemaMonitoraggio
New.xsd