



V.B.G.

Virtual Business Gate

Cooperazione applicativa ed interoperabilità

V 2.10

Ed 1.0 01/12/2011



VBG è un sistema software per la gestione delle segnalazioni certificate di inizio attività e dei procedimenti di autorizzazione e concessione on-line.

Il sistema è stato realizzato da INIT srl su incarico del Consorzio SIR Umbria nell'ambito del progetto Ri_Umbria, il progetto di riuso della Regione Umbria.

Il sistema VBG è a disposizione in riuso a tutte le Pubbliche Amministrazioni che ne fanno richiesta al Consorzio SIR Umbria.

Maggiori informazioni sono reperibili all'indirizzo www.sir.umbria.it/riumbria



SOMMARIO

1	Definizioni	5
2	Introduzione	5
3	L'architettura Centro Servizi Regione Umbria per la Cooperazione Applicativa (CSRUCA)	6
4	Il Sistema Territoriale di Comunicazione (STC)	8
4.1	Ente con soluzione VBG completa (Frontoffice e Backoffice)	9
4.2	Ente con soluzione VBG Frontoffice e Backoffice di terze parti	11
5	Funzionalità del sistema STC	15
6	Funzionalità del nodo NLA	16
7	Utilizzo del servizio STC	17
7.1	Autenticazione al sistema	17
7.2	Verifica messaggi in ingresso	17
7.3	Notifica attività	18
7.4	Richiesta pratica	20
7.5	Richiesta allegato	21
8	Gestione errori del servizio STC	23
9	Utilizzo del servizio NLA	25
9.1	Richiesta pratica	25
9.2	Inserimento pratica	26
9.3	Inserimento attività	27
9.4	Richiesta allegato	28
10	Diagrammi di sequenza	29
10.1	Notifica attività	29
10.2	Richiesta pratica	29
10.3	Richiesta allegato	30
11	Integrazione con infrastruttura di cooperazione applicativa regionale (CSRUCA) e sistema STC	31
11.1.1	Comunicazione Punto-Punto	31
11.1.2	Comunicazione Mediata	31



12	Modalità di Interfacciamento	32
12.1	la chiamata dal sil fruitore verso il NAL	32
12.1.1	Gestione delle informazioni del metodo <i>callNAL</i>	32
12.1.2	Gestione delle informazioni degli altri metodi	34
12.1.3	Gestione degli errori nella risposta al SIL chiamante	35
12.2	La chiamata dal NAL verso il SIL erogatore	35
12.3	Esempio di invocazione del web service di integrazione	35
13	WSDL STC.....	38



1 Definizioni

Nel seguente documento si intende per:

1. CSRUCA
Centro Servizi Regione Umbria per la Cooperazione Applicativa
2. Porta di Dominio
La Porta di Dominio è un elemento concettuale che svolge la funzione di proxy per l'accesso alle risorse applicative del Dominio.
3. Porta Applicativa
Ruolo assunto da una Porta di Dominio nell'ambito di un processo di collaborazione applicativa. Assume tale ruolo una Porta di Dominio che, a seguito della ricezione di un messaggio di richiesta, proveniente da un'altra Porta di Dominio (Porta Delegata), invia al mittente un messaggio di risposta.
4. Porta Delegata
Ruolo assunto da una Porta di Dominio nell'ambito di un processo di collaborazione applicativa. Assume tale ruolo una Porta di Dominio che origina una richiesta di servizio destinato ad un'altra Porta di Dominio (Porta Applicativa).
5. Proxy Applicativo
Modulo software incaricato della gestione dell'integrazione tra la componente di comunicazione (Porta di Dominio) e la componente applicativa.
6. STC - Sistema Territoriale di Comunicazione
E' un sistema che permette a più applicativi presenti su un territorio di comunicare tra loro; per territorio si intende ad esempio un comune e per applicativi si intendono i vari software di Back-End e Front-End in uso presso l'ente comune, la comunicazione è territoriale perché potrebbe coinvolgere anche Back-End di enti terzi come ASL, Arpa, VVF...
7. NLA - Nodo Locale Applicativo
E' una componente che è in relazione uno ad uno con gli applicativi presenti nel territorio, in una comunicazione tra nodi NLA sarà sempre presente un nodo mittente ed un nodo destinatario. Un nodo può parlare la stessa lingua (formato xml) con qualsiasi nodo facente parte della rete STC. Un nodo NLA può rappresentare un mittente o un destinatario a seconda dei casi.

2 Introduzione

Il presente documento costituisce una guida di riferimento per la descrizione dell'integrazione tra i sistemi dei singoli Comuni e VBG utilizzando l'infrastruttura di cooperazione applicativa regionale (CSRUCA) e la tecnologia di comunicazione realizzata per VBG (STC).



3 L'architettura Centro Servizi Regione Umbria per la Cooperazione Applicativa (CSRUCA)

La Cooperazione Applicativa definisce un modello logico ed una serie di specifiche per la realizzazione di una infrastruttura tecnologica che consenta a sistemi ed organizzazioni eterogenee di interoperare in modalità standard.

L'esigenza è di far comunicare secondo modalità definite e condivise le diverse Amministrazioni Centrali e Locali che molto spesso gestiscono in maniera indipendente dati relativi agli stessi soggetti.

Proprio per tale motivo e rispettando le regole tecniche previste dal Sistema Pubblico di Connettività, nonché seguendo i progetti a livello nazionale, la Regione Umbria si è dotata, nel corso degli ultimi anni, di un'infrastruttura di cooperazione applicativa regionale, denominata Centro Servizi Regionale della Cooperazione Applicativa (CSRUCA).

L'architettura del Centro Servizi Regionale ha i seguenti obiettivi:

- Favorire l'interoperabilità tra i Sistemi Informativi degli Enti della Pubblica Amministrazione della Regione dell'Umbria;
- Favorire il dispiegamento di regole comuni per la gestione della sicurezza;
- Consentire il monitoraggio degli scambi telematici.

Gli attori principali di tale processo sono i Sistemi Informativi Locali che comunicano attraverso le Porte di Dominio, le quali, a loro volta, costituiscono i componenti che formalizzano la comunicazione telematica sia per quanto riguarda il trasporto che la sicurezza.

Infine, un apposito accordo tra erogatore e fruitore del servizio, denominato Accordo di Servizio, serve per regolare l'interazione tra le parti.

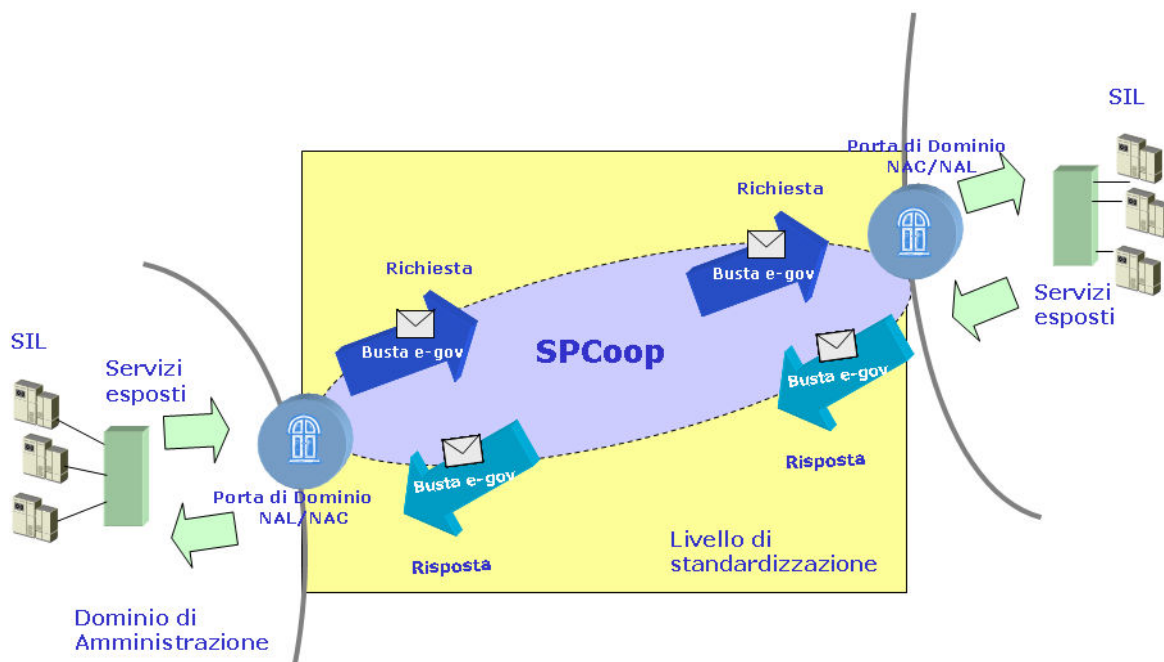


Fig.1 Modello concettuale del sistema di cooperazione applicativa

L'Accordo di Servizio, in particolare, ha il ruolo di determinare:

- Il profilo di collaborazione;



- Le modalità con cui viene gestita la sicurezza;
- I livelli di servizio (SLA).

Per disaccoppiare il Sistema Informativo Locale (SIL) dalla componente di comunicazione viene utilizzata la logia del proxy applicativo che ha in carico le logiche di integrazione ma non quelle applicative che rimangono di competenza del SIL.

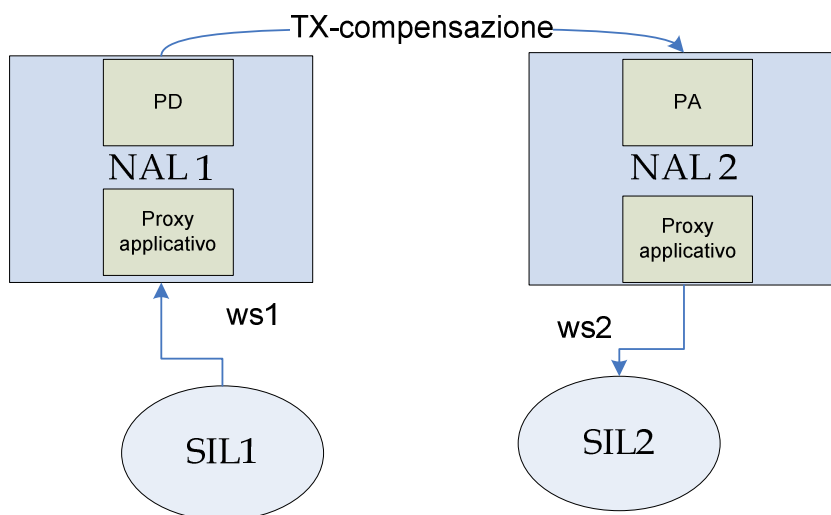


Fig.2 Proxy applicativo – comunicazione punto-punto

In conclusione si ha:

- La Porta di Dominio sovrintende alla comunicazione;
- Il proxy applicativo si occuperà della gestione della parte applicativa;
- La Porta Applicativa, insieme ai proxy applicativi, costituisce un Nodo Applicativo Locale (NAL);
- All'integratore spetta la realizzazione del proxy applicativo.



4 Il Sistema Territoriale di Comunicazione (STC)

Il sistema STC è una componente tecnologica che rappresenta la rete degli applicativi presenti in un determinato territorio.

Il sistema STC è un orchestratore che fornisce la possibilità di far colloquiare tra loro gli applicativi presenti all'interno di un ente comunale (es. SUAP, edilizia, commercio, altri uffici), di estendere la comunicazione, tramite l'infrastruttura di cooperazione della Regione dell'Umbria, verso altre amministrazioni terze (ASL, VVF, Arpa ...) e verso enti di livello superiore (Provincia e Regione).

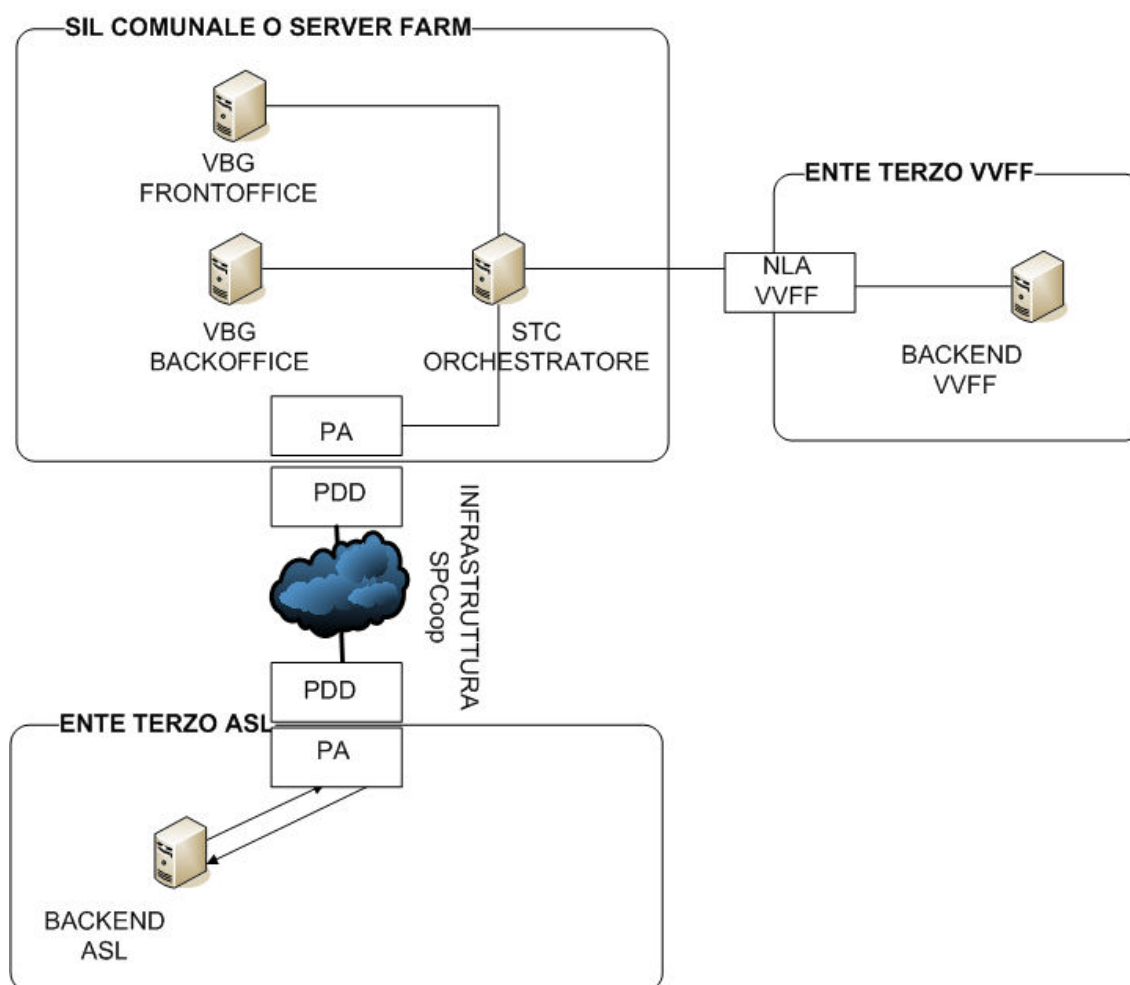
I nodi satellite di STC, denominati NLA, rappresentano delle "plug-in" configurate per tradurre il "linguaggio" (tracciati XML), utilizzato per le comunicazioni interne alla rete STC, nel "linguaggio" utilizzato dal Back-End che il nodo rappresenta.

Qualora la comunicazione avvenga verso un ente esterno il nodo si appoggerà ad una struttura di comunicazione predisposta allo scopo (Infrastruttura CSRUCA). Ogni nodo NLA che si integra alla rete STC sarà sistematicamente in grado di comunicare con tutti gli altri nodi della rete.

La logica di comunicazione vede sempre un nodo mittente e un nodo destinatario.

Sono prospettati i seguenti scenari di cooperazione:

4.1 Ente con soluzione VBG completa (Frontoffice e Backoffice)



In questo caso l'ente possiede la soluzione completa per gestire l'interoperabilità tramite STC. Di seguito viene mostrata una lista di possibili nodi NLA che potrebbero fare parte di una rete STC:

- Nodo di Back-End del SUAP/COMMERCIO (VBG BACKOFFICE)
 - Nodo di Back-End dell'Edilizia (VBG BACKOFFICE)
 - Nodo di Front-End di presentazione domanda e interazione on-line del cittadino (VBG FRONTOFFICE)
 - Nodo di Back-End di VVFF
- La comunicazione avviene direttamente mediante la rete STC e le specifiche definite dal presente documento.

- Nodo di Back-End di ASL (PDD)
- In questo caso la comunicazione è mediata tramite porta di dominio e infrastruttura CSRUCA. STC dell'ente dialoga con la PDD dell'ASL mediante il componente PA che instrada il messaggio imbustandolo con le specifiche SPCoop. La comunicazione tramite PDD deve essere stabilita con precisi accordi di servizio tra gli enti che partecipano a questa modalità e va configurata. I contenuti dei messaggi applicativi aderiscono ai



formati xsd specificati dal presente documento e la comunicazione avviene in modalità sincrona.



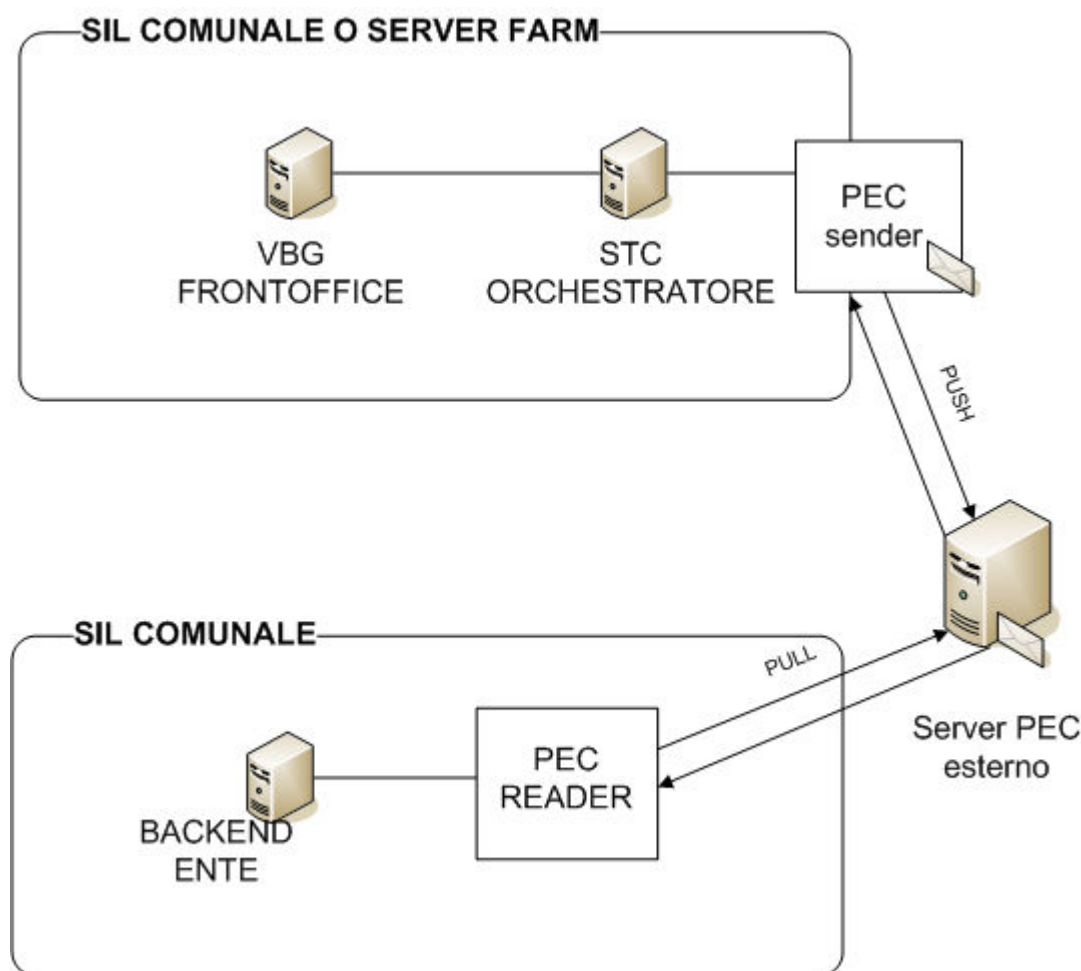
4.2 Ente con soluzione VBG Frontoffice e Backoffice di terze parti

In questo caso l'ente fruisce unicamente del servizio di frontoffice ed utilizza un proprio back end di gestione pratiche.

Sono possibili i seguenti scenari di integrazione tra frontoffice e back office dell'ente.

Cooperazione applicativa ed interoperabilità tramite PEC

In questo scenario l'ente possiede un back end di terze parti e utilizza VBG come frontend.



In questo caso VBG frontoffice è configurato per inviare i messaggi verso un nodo (PEC SENDER) che inbusta le informazioni ed invia ad una cartella configurata un messaggio secondo specifiche DPR 160.

Quello che dovrebbe fare l'integratore del back end di terze parti è scrivere una procedura (PEC READER) che processa la casella mail pec dell'ente ed elabora i messaggi pervenuti secondo specifiche DPR 160.



Il contenuto del messaggio PEC avrà il seguente formato:

Nell'oggetto del messaggio i seguenti campi:

"SUAP: <identificativo sportello destinatario> - <codice fiscale impresa> - <denominazioneimpresa>"

Nel corpo del messaggio i seguenti campi:

"SUAP: <identificativo SUAP destinatario>"

"Pratica: <codice pratica SUAP>"

"Impresa: <codice fiscale impresa> - <denominazione impresa>"

"Richiesta <tipologia richiesta> "

In allegato:

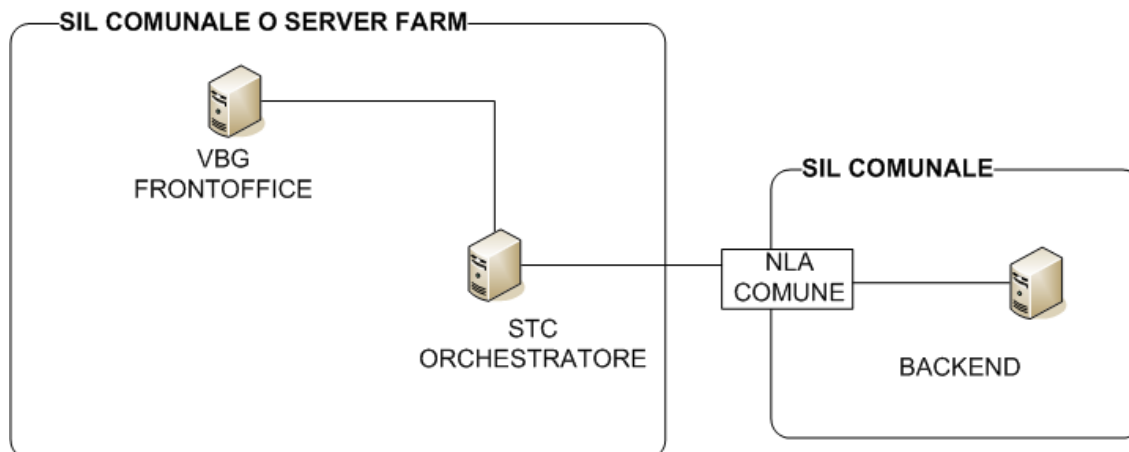
I file accessori che compongono la domanda

Il campo <tipologia richiesta> assumerà il valore "domanda"

Tra gli allegati della pratica ci sarà "<Codice-pratica>.MDA.xml" che conterrà un xml rispettante la struttura xsd definita nel metodo "inserimento Pratica" presente nella sezione NLA del documento relativo all'STC.

Cooperazione applicativa ed interoperabilità tramite STC

Questa ipotesi prevede la cooperazione applicativa mediante chiamate dirette a STC.

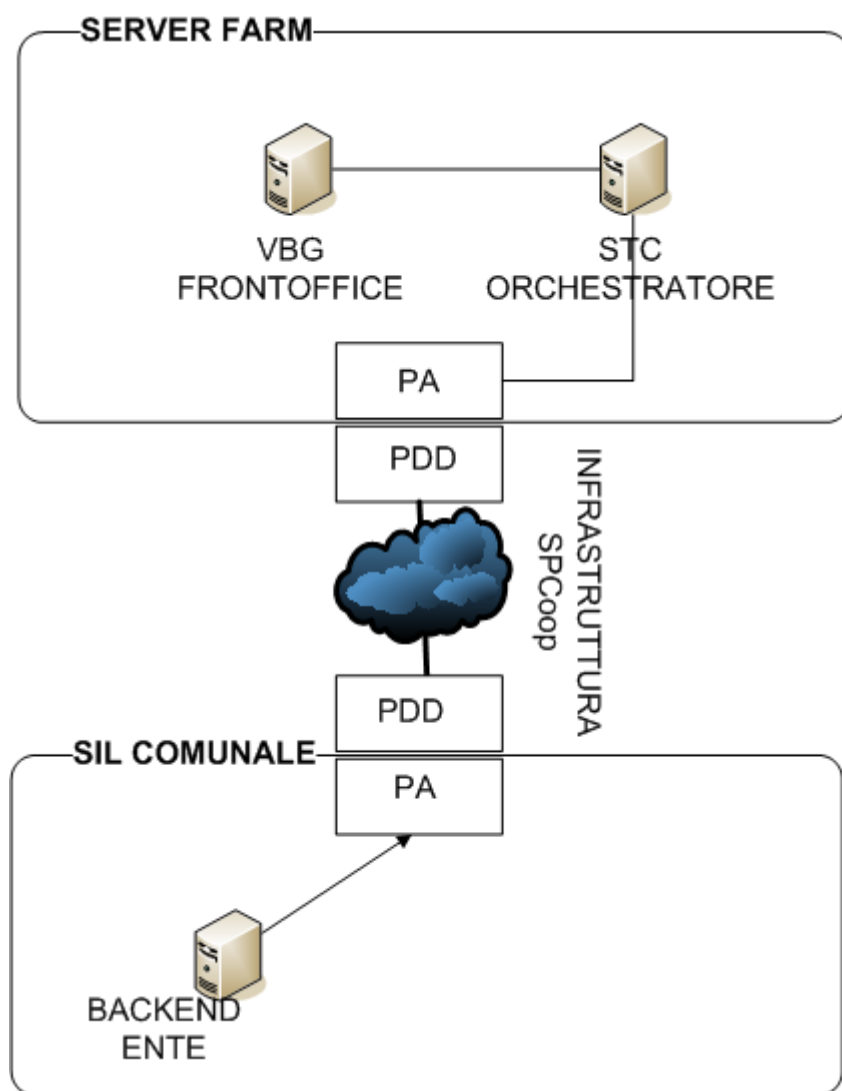


Quello che dovrebbe fare l'integratore del back end di terze parti è scrivere un nodo NLA secondo i formati e le specifiche del presente documento in grado di elaborare i messaggi pervenuti dal VBG Frontend.



Cooperazione applicativa ed interoperabilità tramite PDD

Quello che dovrebbe fare l'integratore del back end di terze parti è scrivere una proxy applicativo in grado di elaborare i messaggi dall'infrastruttura SPCoop secondo i formati e le specifiche del presente documento in grado di elaborare i messaggi pervenuti dal VBG Frontend.





5 Funzionalità del sistema STC

Il sistema STC espone i seguenti Web Service utilizzabili dal generico nodo NLA; un nodo NLA non necessariamente deve essere in grado di consumare tutti i metodi, dipende dalla natura del nodo.

- **Login:** metodo per l'autenticazione del nodo NLA
Per ogni nodo NLA sono rilasciate delle credenziali.
- **CheckToken:** metodo per la verifica della validità della chiave (token) che è stata generata durante l'operazione di login.
- **ListaPratiche:** viene utilizzato dal nodo mittente per ricevere, dal nodo destinatario, una lista di pratiche che rispettano determinati filtri; questo metodo viene utilizzato da un nodo di Front-End che tramite la sua area riservata permette al cittadino di vedere lo stato delle sue pratiche, in accoppiata con RichiestaPratica effettua una vera e propria visura della pratica.
- **RichiestaPratica:** ritorna la visura di una pratica indicando gli estremi della pratica del nodo destinatario della richiesta.
- **RichiestaPraticaCollegata:** ritorna la visura di una pratica indicando dagli estremi della pratica del mittente. Ad esempio dal SUAP si vuole conoscere lo stato del procedimento edilizio che è stato attivato in fase di avvio di un procedimento complesso.
- **NotificaAttività:** metodo per l'invio della richiesta di esecuzione di una particolare attività nel nodo NLA destinatario; per attività si intende una richiesta di parere o semplicemente una comunicazione.
- **AllegatoBinario:** i messaggi relativi alle "pratiche" e alle "attività" possono fare riferimento a degli allegati i quali vengono comunicati solo per riferimento e possono essere "scaricati" o visualizzati attraverso questo metodo.



6 Funzionalità del nodo NLA

Due nodi NLA non possono comunicare direttamente tra loro, ma lo fanno sempre tramite STC e Porte di Dominio; in pratica si ha:

1. il nodo mittente fa una richiesta a STC che la inoltra al nodo destinatario (eventualmente tramite porta di dominio).
2. Gli identificativi univoci di pratiche e attività sono gestiti dal sistema STC in modo tale che il nodo NLA mittente può utilizzare i suoi identificativi per richiedere informazioni sulle pratiche collegate del destinatario.

Di seguito la lista dei metodi web che un nodo NLA deve implementare per entrare a far parte delle rete STC:

- **RichiestaPraticaNLA:** questo metodo ritorna i dettagli di una pratica.
- **InserimentoPraticaNLA:** questo metodo inserisce una pratica nell'applicativo che è rappresentato dal nodo, come risposta vengono tornati il numero assegnato alla pratica e l'eventuale protocollo.
- **InserimentoAttivitaNLA:** il metodo inserisce una attività nell'applicativo che il nodo rappresenta, l'attività può essere ad esempio "richiesta parere" da parte del mittente.
- **AllegatoBinarioNLA:** il metodo ritorna il file binario richiesto.

Ognuno dei metodi ha tra gli elementi dei messaggi di input la stringa **token** con la quale verificare la validità del messaggio inviato utilizzando il web method **CheckToken** esposto dal sistema STC.



7 Utilizzo del servizio STC

7.1 Autenticazione al sistema

Il web method **Login** consente l'autenticazione al sistema STC e, di seguito, è riportata la segnatura come indicata nel WSDL relativo:

```
<wsdl:operation name="Login">
  <wsdl:input message="tns:LoginRequest" name="LoginRequest" />
  <wsdl:output message="tns:LoginResponse" name="LoginResponse" />
</wsdl:operation>
```

Il messaggio di input è caratterizzato dal seguente schema XSD:

Gli elementi username e password sono quelli forniti durante la registrazione del nodo NLA al sistema STC.

```
<element name="LoginRequest">
  <complexType>
    <sequence>
      <element name="username" type="string"/>
      <element name="password" type="string"/>
    </sequence>
  </complexType>
</element>
```

Il metodo Login resituisce il seguente messaggio:

il messaggio è composto da un elemento booleano che indica l'esito dell'autenticazione (true: autenticazione riuscita, false: autenticazione negata) e da una stringa che rappresenta il token rilasciato a seguito di un'autenticazione riuscita.

```
<element name="LoginResponse">
  <complexType>
    <sequence>
      <element name="result" type="boolean"/>
      <element name="token" type="string"/>
    </sequence>
  </complexType>
</element>
```

7.2 Verifica messaggi in ingresso

Questo web method può essere invocato dal nodo NLA ogni volta che il nodo riceve una chiamata ai web service esposti. Ognuno dei messaggi inviati conterrà la voce token in modo tale da permettere all'NLA chiamato di verificare la validità del messaggio inviato.

Il web method è così costituito:

```
<wsdl:operation name="CheckToken">
<wsdl:input
```



```

      message="tns:CheckTokenRequest"
      name="CheckTokenRequest" />
<wsdl:output
      message="tns:CheckTokenResponse"
      name="CheckTokenResponse" />
</wsdl:operation>

```

Il messaggio di input è rappresentato dal seguente schema XSD:

```

<element name="CheckTokenRequest">
  <complexType>
    <sequence>
      <element name="token" type="string"/></element>
    </sequence>
  </complexType>
</element>

```

Il messaggio di risposta è rappresentato dal seguente XSD:

result=true: verifica positiva del token, il messaggio è attendibile
 result=false: verifica negativa del token, il messaggio non è attendibile

```

<element name="CheckTokenResponse">
  <complexType>
    <sequence>
      <element name="result" type="boolean"/></element>
    </sequence>
  </complexType>
</element>

```

7.3 Notifica attività

Il web method NotificaAttività permette l'invio da parte di un NLA mittente di una richiesta di esecuzione di una determinata attività nell'NLA di destinazione.

Ad esempio può essere invocato da un nodo che rappresenta un SUAP il quale vuole notificare l'avvio di un procedimento edilizio al nodo destinatario dell'edilizia.

La segnatura del metodo è la seguente:

```

<wsdl:operation name="NotificaAttività">
  <wsdl:input
      message="tns:NotificaAttivitàRequest" name="NotificaAttivitàRequest" />
  <wsdl:output
      message="tns:NotificaAttivitàResponse" name="NotificaAttivitàResponse" />
</wsdl:operation>

```

Il messaggio di input è caratterizzato dal seguente schema XSD:

```

<element name="NotificaAttivitàRequest">

```



```

<complexType>
  <sequence>
    <element name="token" type="string"/>
    <element name="sportelloMittente"
      type="types:SportelloType">
    </element>
    <element name="sportelloDestinatario"
      type="types:SportelloType">
    </element>
    <element name="datiAttivita"
      type="types:DettaglioAttivitaType">
    </element>
    <element name="rifPraticaDestinatario"
      type="types:RiferimentiPraticaType" maxOccurs="1" minOccurs="0">
    </element>
  </sequence>
</complexType>
</element>

```

Il messaggio è composto dal token rilasciato a seguito della chiamata al metodo Login, dall'identificativo dell'NLA mittente (elemento sportelloMittente), dall'identificativo dell'NLA destinatario (elemento sportelloDestinatario) e dal dettaglio dell'attività (elemento datiAttivita).

L'identificazione dell'NLA è contenuta nell'elemento SportelloType, costituito dal seguente XSD:

```

<complexType name="SportelloType">
  <sequence>
    <element name="idEnte" type="string"/>
    <element name="idSportello" type="string"/>
  </sequence>
</complexType>

```

Gli elementi idEnte ed IdSportello rappresentano gli identificativi univoci di un nodo NLA e sono inseriti in STC in fase di registrazione del nodo NLA.

I dati inviati nell'elemento datiAttivita corrispondono ai dati relativi all'attività eseguita nell'NLA mittente tramite i quali l'NLA destinatario è in grado di individuare l'attività da eseguire sul proprio sistema.

Il messaggio di risposta è rappresentato dallo schema XSD seguente:

```

<element name="NotificaAttivitaResponse">
  <complexType>
    <choice minOccurs="1">
      <element name="dettaglioattivita"
        type="types:RiferimentiAttivitaType" minOccurs="0" maxOccurs="1">
      </element>
      <element name="dettaglioErrore" type="types:ErroreType"
        minOccurs="0" maxOccurs="unbounded">
      </element>
    </choice>
  </complexType>
</element>

```



```
</complexType>
</element>
```

Il messaggio di risposta può essere quindi composto o dai riferimenti dell'attività inserita nell'NLA destinatario o da una lista di errori che motivano l'errata esecuzione del metodo.

7.4 Richiesta pratica

STC mette a disposizione il web method RichiestaPratica che permette ad un NLA mittente di richiedere i dettagli di una pratica che risiede in un NLA destinatario.

La segnatura del metodo è la seguente:

```
<wsdl:operation name="RichiestaPratica">
  <wsdl:input
    message="tns:RichiestaPraticaRequest" name="RichiestaPraticaRequest" />
  <wsdl:output
    message="tns:RichiestaPraticaResponse" name="RichiestaPraticaResponse" />
</wsdl:operation>
```

Il messaggio di input è caratterizzato dal seguente schema XSD:

```
<element name="RichiestaPraticaRequest">
  <complexType>
    <sequence>
      <element name="token" type="string"/>
      <element name="sportelloMittente"
        type="types:SportelloType"/>
      </element>
      <element name="sportelloDestinatario"
        type="types:SportelloType"/>
      </element>
      <element name="rifPratica"
        type="types:RiferimentiPraticaType"/>
      </element>
    </sequence>
  </complexType>
</element>
```

Come si vede dallo schema sono presenti i tre elementi che caratterizzano tutti i messaggi scambiati tra NLA e STC (tranne la Login e la CheckToken): token, sportelloMittente e sportelloDestinatario.

L'elemento distintivo è l'idPratica che identifica la pratica ricercata nell'NLA di destinazione.

In risposta a tale input avremo il seguente messaggio (schema XSD):

```
<element name="RichiestaPraticaResponse">
  <complexType>
    <sequence>
      <element name="dettaglioPratica"
        type="types:DettaglioPraticaType" maxOccurs="1" minOccurs="0"/>
      </element>
    </sequence>
  </complexType>
</element>
```



```

        <element name="dettaglioErrore" type="types:ErroreType"
            minOccurs="0" maxOccurs="unbounded">
        </element>
    </sequence>
</complexType>
</element>

```

Il messaggio riporta il dettaglio della pratica richiesta. (per i dettagli dei tipi si rimanda al capitolo "STC wsdl del servizio").

7.5 Richiesta allegato

Il web method AllegatoBinario permette di recuperare eventuali files collegati ad una pratica

La segnatura del metodo è la seguente:

```

<wsdl:operation name="AllegatoBinario">
    <wsdl:input
        message="tns:AllegatoBinarioRequest" name="AllegatoBinarioRequest" />
    <wsdl:output
        message="tns:AllegatoBinarioResponse" name="AllegatoBinarioResponse" />
</wsdl:operation>

```

Il messaggio di input è rappresentato dal seguente schema XSD:

```

<element name="AllegatoBinarioRequest">
    <complexType>
        <sequence>
            <element name="token" type="string"></element>
            <element name="sportelloMittente"
                type="types:SportelloType">
            </element>
            <element name="sportelloDestinatario"
                type="types:SportelloType">
            </element>
            <element name="riferimentiAllegato"
                type="types:RiferimentiAllegatoType">
            </element>
        </sequence>
    </complexType>
</element>

```

Il messaggio di output è rappresentato dal seguente schema XSD:

```

<element name="AllegatoBinarioResponse">
    <complexType>
        <sequence>
            <element name="mimeType" type="string" minOccurs="0"
                maxOccurs="1" />
            <element name="binaryData" type="base64Binary" />
            <element name="fileName" type="string" />
        </sequence>
    </complexType>

```



</element>



8 Gestione errori del servizio STC

La notifica degli errori riscontrati dal sistema STC al nodo NLA chiamante avviene sempre sul canale http tramite l'invio di un envelope SOAP contenente come unica parte dell'elemento body, l'elemento fault.

L'elemento fault contiene solamente due tipi di codici:

SOAP-ENV:Client, indica un errore relativo al messaggio inviato dal client del servizio (nodo NLA).

SOAP-ENV:Server, indica un errore interno al sistema STC

Esempio di errore di tipo SOAP-ENV:Client

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
  <faultcode>SOAP-ENV:Client</faultcode>
  <faultstring xml:lang="en">Validation error</faultstring>
  <detail>
    <spring-ws:ValidationError
      xmlns:spring-ws="http://springframework.org/spring-ws">
      cvc-complex-type.2.4.b: The content of element 'q0:CheckTokenRequest' is not complete. One of
      '{"http://sigepro.init.it/rte":token}' is expected
    </spring-ws:ValidationError>
  </detail>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Esempio di errore di tipo SOAP-ENV:Server

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
  <faultcode>SOAP-ENV:Server</faultcode>
  <faultstring xml:lang="en">Internal error</faultstring>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```





9 Utilizzo del servizio NLA

9.1 Richiesta pratica

NLA attraverso questo web method ritorna i dati di una pratica. Deve essere utilizzato il tag “rifPratica” per rintracciare una pratica nel DB di NLA, si consiglia di seguire le ricerche utilizzando i tag nel seguente ordine:

1. rifPratica.idPratica
2. rifPratica.numeroPratica e rifPratica.dataPratica
3. rifPratica.numeroProtocolloGenerale e rifPratica.dataProtocolloGenerale

Il tag “altriDati” sarà popolato a seconda degli accordi tra NLA e potrebbe contenere informazioni aggiuntive per recuperare una pratica.

Il metodo può tornare una sola pratica, nel caso in cui i filtri selezionino più pratiche si consiglia di generare un errore.

Segnatura metodo

```
<wsdl:operation name="RichiestaPraticaNLA">
  <wsdl:input
    message="tns:RichiestaPraticaNLRequest" name="RichiestaPraticaNLRequest" />
  <wsdl:output
    message="tns:RichiestaPraticaNLResponse" name="RichiestaPraticaNLResponse" />
</wsdl:operation>
```

XSD messaggio di input

```
<element name="RichiestaPraticaNLRequest">
  <complexType>
    <sequence>
      <element name="token" type="string"/>
      <element name="sportelloMittente" type="types:SportelloType"/>
      <element name="sportelloDestinatario" type="types:SportelloType"/>
      <element name="rifPratica" type="types:RiferimentiPraticaType"/>
    </sequence>
  </complexType>
</element>
```

XSD messaggio di output

```
<element name="RichiestaPraticaNLResponse">
  <complexType>
    <sequence>
      <element name="dettaglioPratica" type="types:DettaglioPraticaType"
maxOccurs="1" minOccurs="0"/>
      <element name="dettaglioErrore" type="types:ErroreType" minOccurs="0"
maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
```



9.2 Inserimento pratica

Questo metodo viene utilizzato per comunicare all’NLA di inserire una nuova pratica nel suo DB, nonostante il sistema STC sia in grado di conoscere tutte le pratiche che sono transitate in esso si consiglia di utilizzare gli elementi “dettagliPratica” e “rifPraticaDestinatario” per ricercare la pratica evitando un doppio inserimento.

La richiesta del doppio inserimento potrebbe avvenire per le pratiche che non sono presenti in STC sin dalla loro creazione e per sopraggiunti errori sulla rete di comunicazione.

Per una corretta gestione dei messaggi è consigliabile consultare la sezione 6.5

Segnatura metodo

```
<wsdl:operation name="InserimentoPraticaNLA">
  <wsdl:input
    message="tns:InserimentoPraticaNLAResponse" name="InserimentoPraticaNLAResponse" />
  <wsdl:output
    message="tns:InserimentoPraticaNLAResponse"
    name="InserimentoPraticaNLAResponse" />
</wsdl:operation>
```

XSD messaggio di input

```
<element name="InserimentoPraticaNLAResponse">
  <complexType>
    <sequence>
      <element name="token" type="string"/>
      <element name="sportelloMittente"
        type="types:SportelloType"/>
      <element name="sportelloDestinatario"
        type="types:SportelloType"/>
      <element name="dettaglioPratica"
        type="types:DettaglioPraticaType"/>
      <element name="rifPraticaDestinatario" type="types:RiferimentiPraticaType"
        maxOccurs="1" minOccurs="0"/>
    </sequence>
  </complexType>
</element>
```

XSD messaggio di output

```
<element name="InserimentoPraticaNLAResponse">
  <complexType>
    <choice minOccurs="1">
```



```

minOccurs="0">
    <element name="dettaglioPratica" type="types:RiferimentiPraticaType"
    </element>
    <element name="dettaglioErrore" type="types:ErroreType" minOccurs="0"
maxOccurs="unbounded">
    </element>
  </choice>
</complexType>
</element>

```

9.3 Inserimento attività

Segnatura metodo

```

<wsdl:operation name="InserimentoAttivitaNLA">
  <wsdl:input
    message="tns:InserimentoAttivitaNLAResult" name="InserimentoAttivitaNLAResult"
  />
  <wsdl:output
    message="tns:InserimentoAttivitaNLAResponse"
    name="InserimentoAttivitaNLAResponse" />
</wsdl:operation>

```

XSD messaggio di input

```

<element name="InserimentoAttivitaNLAResult">
  <complexType>
    <sequence>
      <element name="token" type="string"></element>
      <element name="sportelloMittente"
        type="types:SportelloType">
      </element>
      <element name="sportelloDestinatario"
        type="types:SportelloType">
      </element>
      <element name="datiAttivita"
        type="types:DettaglioAttivitaType">
      </element>
      <element name="rifPraticaMittente" type="types:RiferimentiPraticaType"
maxOccurs="1" minOccurs="0"></element>
    </sequence>
  </complexType>
</element>

```

XSD messaggio di output

```

<element name="InserimentoAttivitaNLAResponse">
  <complexType>
    <choice minOccurs="1">
      <element name="dettaglioAttivita" type="types:RiferimentiAttivitaType"
minOccurs="0">

```



```

        </element>
        <element name="dettaglioErrore" type="types:ErroreType" minOccurs="0"
maxOccurs="unbounded">
        </element>
    </choice>
</complexType>
</element>

```

9.4 Richiesta allegato

Segnatura metodo

```

<wsdl:operation name="AllegatoBinarioNLA">
    <wsdl:input
        message="tns:AllegatoBinarioNLAResponse" name="AllegatoBinarioNLAResponse" />
    <wsdl:output
        message="tns:AllegatoBinarioNLAResponse" name="AllegatoBinarioNLAResponse" />
</wsdl:operation>

```

XSD messaggio di input

```

<element name="AllegatoBinarioNLAResponse">
    <complexType>
        <sequence>
            <element name="token" type="string"/></element>
            <element name="sportelloMittente"
                type="types:SportelloType">
            </element>
            <element name="sportelloDestinatario"
                type="types:SportelloType">
            </element>
            <element name="riferimentiAllegato"
type="types:RiferimentiAllegatoType"/></element>
        </sequence>
    </complexType>
</element>

```

XSD messaggio di output

```

<element name="AllegatoBinarioNLAResponse">
    <complexType>
        <sequence>
            <element name="mimeType" type="string" minOccurs="0" maxOccurs="1" />
            <element name="binaryData" type="base64Binary" />
            <element name="fileName" type="string" />
        </sequence>
    </complexType>
</element>

```



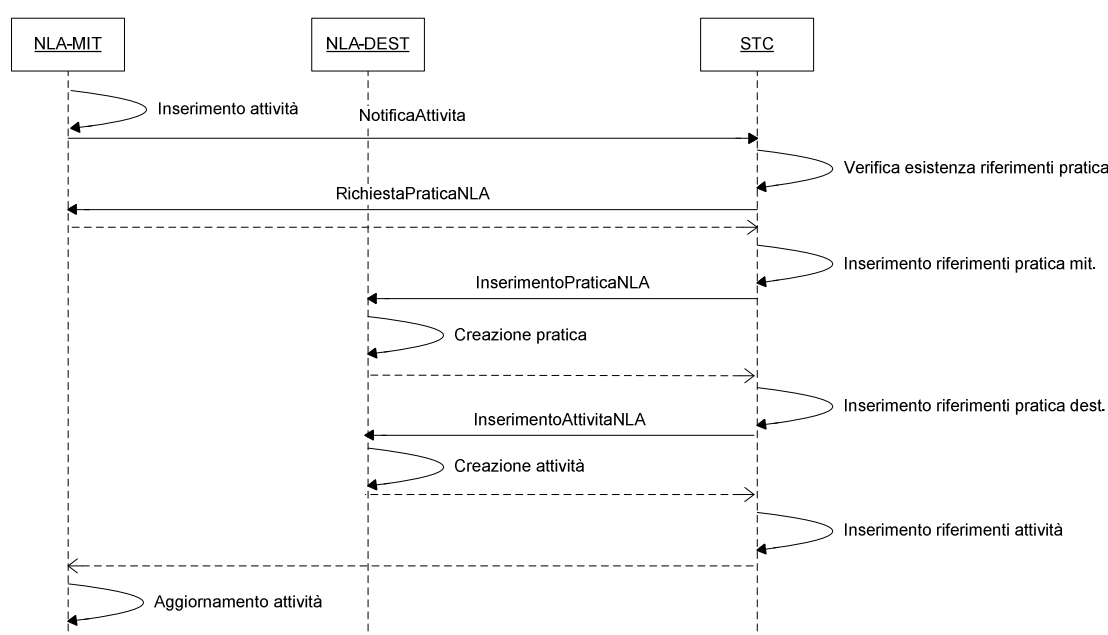
10 Diagrammi di sequenza

Nei diagrammi seguenti è evidenziata l'interazione tra due nodi NLA, nodo mittente (NLA-MIT) e nodo destinatario (NLA-DEST) mediata dal sistema STC.

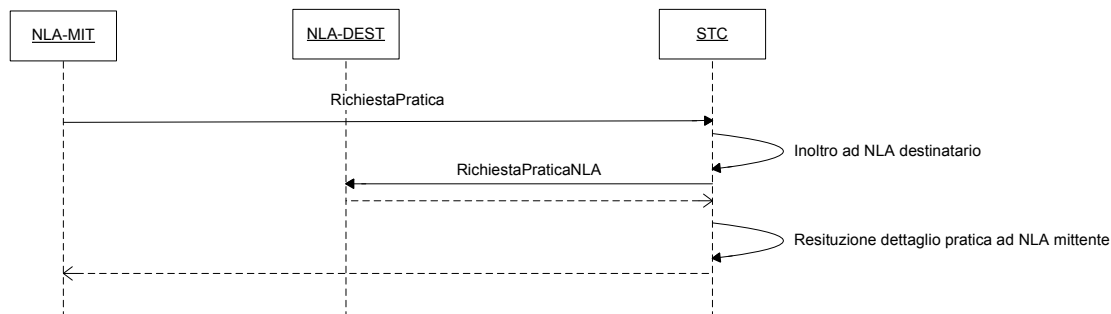
Nei diagrammi non è evidenziata la fase di login che deve precedere ognuna delle chiamate in modo da recuperare un token valido.

Per ognuna delle chiamate effettuate da STC verso NLA è possibile utilizzare il metodo CheckToken per la validazione del mittente prima del processamento del messaggio.

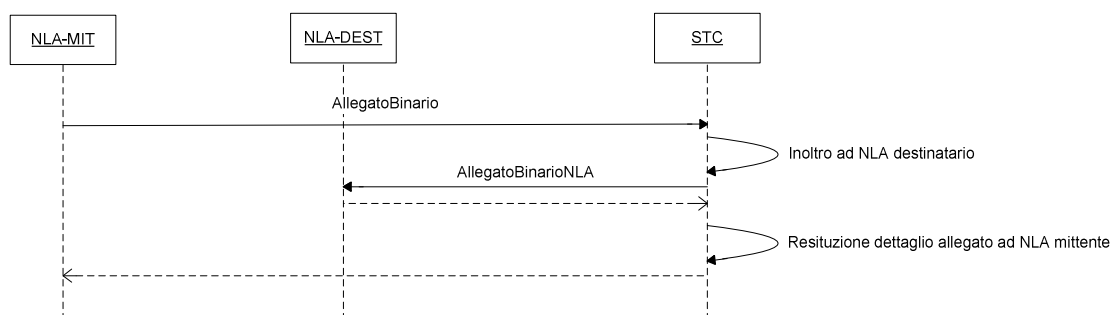
10.1 Notifica attività



10.2 Richiesta pratica



10.3 Richiesta allegato





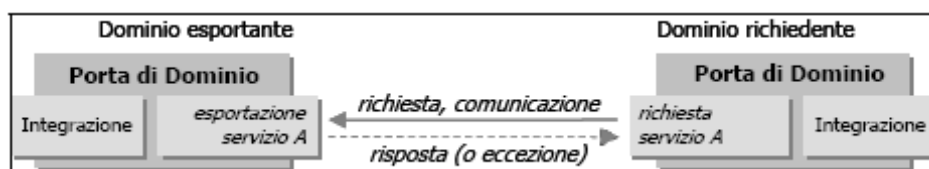
11 Integrazione con infrastruttura di cooperazione applicativa regionale (CSRUCA) e sistema STC

Tramite il STC e le Porte di Dominio, la comunicazione tra due NLA può avvenire in modalità punto-punto.

11.1.1 Comunicazione Punto-Punto

La comunicazione punto/punto prevede, essenzialmente, un tipo di collaborazione in modalità sincrona.

In questo caso il SIL fruitore si aspetta dal SIL erogatore la risposta applicativa nel payload (si osservi che la risposta può anche essere un errore applicativo).



Va sottolineato che il SIL erogante deve inserire eventuali errori applicativi (non di trasporto) nel corpo della busta. Questi errori possono essere restituiti immediatamente (profilo di collaborazione sincrono) e dopo un certo tempo (profilo di collaborazione asincrona) su richiesta di chi ha chiesto il servizio (profilo di collaborazione asincrona asimmetrica) o su chiamata all'indietro di chi aveva il compito di erogare il servizio (profilo di collaborazione asincrona asimmetrica).

Nel caso della comunicazione punto/punto l'accordo di servizio ha:

- 1 componente non formalizzata
- 1 componente formalizzata (tecnica)

Va aggiunta una nota relativamente ad una particolare modalità di comunicazione:

- la trasmissione in sequenza

La trasmissione in sequenza è un caso particolare di collaborazione e quindi in questo caso l'identificativo idMessaggioCapostipite deve essere valorizzato.

In questo caso il SIL che trasmette messaggi deve fornire anche il numero di sequenza. Le porte garantiscono la consegna certa ed in ordine anche se l'ordine di invio non è coerente con la sequenza assegnata.

Tale modalità non viene supportata se si usa come mediatore il gestore eventi.

11.1.2 Comunicazione Mediata

La comunicazione mediata avviene mediante le logiche proprie del mediatore.

Nel caso del Gestore Eventi viene supportata la comunicazione di tipo sincrono. Quindi:

- Un SIL pubblica, N SIL si sottoscrivono



La comunicazione SIL-SIL avviene due passi:

- Dapprima il SIL pubblicante pubblica chiamando il servizio apposito con modalità sincrona
- Poi il GE notifica ad uno degli N sottoscrittori anche qui usando la modalità sincrona

In questo caso esistono:

- 1+N accordi di servizio formalizzati (1 tra il SIL pubblicatore ed il GE e gli altri tra il GE e ciascuno degli N SIL sottoscrittori)
- N accordi di servizio NON formalizzati tra il SIL pubblicatore e gli N SIL sottoscrittori

12 Modalità di Interfacciamento

Un integratore:

- può chiamare il Centro Servizi per richiedere un servizio
- essere chiamato dal Centro Servizi per erogare un servizio

12.1 la chiamata dal sil fruitore verso il NAL

Attualmente il Centro servizi espone alcuni metodi utili ad un integratore per l'esecuzione di un servizio:

- **callNAL**. Viene utilizzato per invocare un servizio
- **getUUID**. Viene invocato per ottenere un identificativo di busta valida da inserire nella chiamata callNAL
- **getRiscontro**. Viene invocato per conoscere se il messaggio di richiesta è stato effettivamente consegnato al SIL destinatario
- **getTimeStamp**. Viene invocato per ottenere un timestamp dal NAL.

12.1.1 Gestione delle informazioni del metodo *callNAL*

La callNAL prevede i seguenti parametri:

```
byte[] callNAL (
String mittente,           // id del SIL mittente (obbligatorio)
String destinatario,      // id del SIL destinatario (obbligatorio)
String servizio,          // id del servizio (obbligatorio)
String azione,            // id dell'azione (opzionale)
Double sequenza,          // progressivo di sequenza (opzionale)1
String idMessaggio,       // id del messaggio (obbligatorio)
String idMessaggioCapostipite, // id capostipite (opzionale)2
String idMessaggioCorrelato, // id correlato3 (opzionale)
```

¹ E' obbligatoria se il profilo di collaborazione esige la spedizione in sequenza

² È obbligatorio se il profilo di collaborazione esige la spedizione in sequenza e si intende dare inizio alla sequenza

³ E' obbligatorio quando si invoca il servizio correlato in una comunicazione asincrona simmetrica o asimmetrica



```

byte[] payload,           // messaggio da inoltrare (opzionale)
String soggetto,         // id soggetto richiedente il servizio (opzionale)4
String password,         // psw soggetto richiedente il servizio (opzionale)
String ruolo,            // ruolo soggetto richiedente il servizio (opzionale)
byte[] certificato       // certificato del soggetto richiedente il servizio (opzionale)5
)

```

Il primo gruppo di informazioni è costituito dai dati base. Essi sono:

- mittente
- destinatario
- servizio
- azione

Sono tutti obbligatori. I valori dei vari campi vengono stabiliti in fase di redazione dell'accordo di servizio.

Il secondo gruppo di informazioni si riferisce all'identificazione ed alla correlazione di messaggi.

La valorizzazione delle informazioni dipende dal tipo di profilo di collaborazione scelta. La tabella seguente mostra le diverse possibilità:

Tipo di profilo	IdMessaggio	Sequenza	IdMessaggio Capostipite	IdMessaggioCorrelato
Sincrono	Obbligatori. E' l'ID della busta. Deve essere procurato invocando il metodo getUUID.	Obbligatorio	Facoltativo. Diventa obbligatorio in caso di collaborazione (la sequenza è una caso particolare di collaborazione) E' l'ID della busta spedita per prima e che da inizio alla collaborazione/sequenza. Durante tutta la collaborazione quindi IdMessaggio (di inizio collaborazione) = IdMessaggioCapostipite	-
Asincrono simmetrico	Obbligatorio. E' l'ID della busta. Deve essere procurato invocando il metodo getUUID.	Obbligatorio in caso di invio in sequenza. La sequenza è data da un numero che segue l'ordine logico di trasmissione	Obbligatorio. Per poterlo trasmettere durante tutta la durata della collaborazione il SIL deve avere qualche meccanismo di <i>retention</i> delle informazioni	Obbligatorio quando si chiama il servizio correlato. E' l'identificativo della busta corrispondente alla richiesta precedente ed alla quale il SIL sta attualmente rispondendo. Per poterlo trasmettere il SIL deve avere qualche meccanismo di <i>retention</i> delle informazioni correlate.
Asincrono Asimmetrico	Obbligatorio. E' l'ID della busta. Deve essere	Obbligatorio in caso di invio in sequenza. La	Obbligatorio. Per poterlo trasmettere durante tutta la durata della	Obbligatorio quando si chiama il servizio correlato.. E' l'identificativo della busta corrispondente alla

⁴ Soggetto, password e ruolo devono essere obbligatoriamente specificati se l'accordo di servizio prevede autenticazione/autorizzazione. L'accordo specifica quali delle informazioni devono essere dichiarate. Potrebbero essere in alternativa al certificato

⁵ Al momento è il certificato richiesto dal GE. Non viene passato bensì è quello che si trova nel keystore della porta e quindi viene prelevato dal contesto della porta stessa



Tipo di profilo	IdMessaggio	Sequenza	IdMessaggio Capostipite	IdMessaggioCorrelato
	procurato invocando il metodo getUUID.	sequenza è data da un numero che segue l'ordine logico di trasmissione	collaborazione il SIL deve avere qualche meccanismo di <i>retention</i> delle informazioni	richiesta precedente per la quale il SIL sta attualmente chiedendo risposta. Per poterlo trasmettere il SIL deve avere qualche meccanismo di <i>retention</i> delle informazioni

Per ciò che concerne il payload (cioè il contenuto applicativo del messaggio) la tabella seguente mostra le diverse possibilità:

Tipo di profilo	Payload (parametro in ingresso)	Payload (byte[] restituito dalla callNAL)
Sincrono	Obbligatorio	Il payload è quello inserito dal SIL destinatario (payload applicativo) ⁶
Asincrono simmetrico	Obbligatorio	Il payload è vuoto ⁷ .
Asincrono Asimmetrico	Obbligatorio	Il payload è vuoto ⁸

Quando la richiesta riscontra dei problemi interni viene sollevato un SOAP fault in modo da evitare al richiedente l'analisi di eventuali, specifici, messaggi di errore.

Per ciò che riguarda infine le informazioni di sicurezza:

- soggetto
- ruolo
- password
- certificato

la loro gestione viene rimandata allo stabilizzarsi della gestione della sicurezza in regione Umbria, ma dovrà essere prevista.

12.1.2 Gestione delle informazioni degli altri metodi

getUUID() Il metodo restituisce un UUID = IDunivoco di busta secondo specifiche CNIPA

⁶ Si noti che in caso di comunicazione mediata il payload non è quello del SIL ma quello del sistema mediatore. In questo caso i messaggi di risposta vengono generati dal sistema mediatore. Al paragrafo "Messaggi Generati dal Gestore Eventi" si può vedere i messaggi generati dal Gestore Eventi a fronte dei servizi da questo esposti.

⁷ In questa prima fase

⁸ In questa prima fase



A seconda dei casi e dell'opportunità il SIL inviante può avere utilità a memorizzare tale ID.

La sequenza di chiamata è:

```
getUUID()
  callNAL(....)
```

getRiscontro (UUID) restituisce un timestamp (gg-mm-yyyy-hh:ss)

Questo metodo consente di verificare lo stato di un invio asincrono. Viene utilizzato per i messaggi a consegna affidabile, cioè con gestione del riscontro che viene utilizzato nei profili di collaborazione asincrona oppure nei casi di consegna fatta con profilo sincrono ma in cui si passi attraverso un gestore eventi.

Nel caso in cui il messaggio non è ancora stato consegnato, il timestamp assume un valore di default pari a "0000000000". Nel caso in cui è stato consegnato, il timestamp è valorizzato.

getRiscontro () restituisce un timestamp (gg-mm-yyyy-hh:ss) che può essere utilizzato dal SIL, ad es. per sincronizzarsi con il NAL.

12.1.3 Gestione degli errori nella risposta al SIL chiamante

In generale viene fatta una distinzione a secondo che il servizio sia sincrono o asincrono.

Quando la conversazione è di tipo asincrono, dal momento che non ci si aspetta una risposta applicativa, se si è verificata qualche anomalia si riceve un SOAP Fault.

12.2 La chiamata dal NAL verso il SIL erogatore

La chiamata dal NAL verso il SIL avviene da parte del proxy collocato sul NAL erogatore (tipo **_E_in_out**).

Poiché il proxy è realizzato a cura dell'integratore i dettagli implementativi sono lasciati all'integratore stesso. Il proxy può quindi chiamare il proprio sistema usando una POST http, un web service, una connessione JDBC, un messaggio SMTP, ecc.

Si tenga presente quanto già detto in precedenza:

- Il proxy di tipo **_E_in_out** deve restituire TRUE se la consegna al SIL da parte del proxy applicativo è avvenuta con successo. Si ricorda che si parla di consegna e non di elaborazione. Il risultato dell'elaborazione (sia essa avvenuta con successo o meno) va consegnata dal SIL mediante altra via;
- In caso contrario deve restituire FALSE.

12.3 Esempio di invocazione del web service di integrazione

Di seguito è riportato un esempio di codice Java per l'invocazione di un servizio di cooperazione applicativa mediante il web service di integrazione.

La classe NalClient contiene un main che effettua la chiamata. Nel costruttore della classe viene inizializzato il client del web service Axis di integrazione denominato "Nal" che espone i servizi callNal() e getUUID().

```
public class NalClient
  implements INal
```



```
{
    INal binding;

    /**
     * Costruttore
     */
    public NalClient(String address)
        throws Exception
    {
        NalServiceLocator nalSrv = new NalServiceLocator();
        nalSrv.setNalEndpointAddress(address);
        nalSrv.setNalWSDDServiceName("Nal");
        binding = (INal) nalSrv.getNal();
    }

    /**
     * Metodo per la richiesta di un identificativo unico nell'ambito del NAL
     * @return string contenente l'identificativo
     */
    public String getUUID()
        throws Exception
    {
        return binding.getUUID();
    }
}

public static void main(String[] args)
    throws Exception
{
    //url della porta di dominio (porta applicativa)
    String nalUrl =
        "http://baschi.spcoop.pa.umbria.it:7777/gateway-baschi/services/Nal";

    String mittente = "ComuneBaschi";
    String destinatario = "CentroServiziRegioneUmbria";
    String servizio = "PubblicaEventiEcografico";
    String azione = "ecograficoMsgtratto";

    NalClient cl = new NalClient(nalUrl);
    String idMessaggio = cl.getUUID();
    double sequenza = -1.0;

    String idMessaggio = "";
    String idMessaggioCapostipite = null;
    String payload = /* XML del contenuto applicativo */
    String soggetto = "";
    String password = "";
    String ruolo = "";
    String certificato = "";

    byte[] resp =
        cl.callNal(mittente, destinatario, servizio, azione, sequenza,
            idMessaggio, idMessaggioCapostipite,
            idMessaggioCollaborazione, payload.getBytes(), soggetto,
            password, ruolo, certificato.getBytes());
}
```



...



13 WSDL STC

Gli xsd aggiornati sono visualizzabili al seguente URL:

<http://vbg.gruppoinit.it/stc/services/stc.wsdl>

<http://vbg.gruppoinit.it/stc/services/nla.wsdl>