



**PROCEDURA APERTA
PER L' ESTENSIONE DEL SISTEMA DI CONTROLLO DEGLI
ACCESSI ALLE ZTL NELLA CITTÀ DI BERGAMO E LA
MANUTENZIONE DEI VARCHI ESISTENTI**

SPECIFICA WEBSERVICES SIRIOWEB

ALLEGATO II

SISTEMA SIRIO[®] City

Specifica di interscambio dati
tra SirioWeb e sistemi esterni

Motu **S21**

Sommario

1.	Riepilogo delle modifiche	3
2.	Lista delle abbreviazioni	4
3.	SCOPO DEL DOCUMENTO	5
4.	AMBITO DEL PROGETTO	5
5.	INTRODUZIONE ALL'ARCHITETTURA	6
6.	CONFIGURAZIONI INIZIALI	7
7.	PRINCIPI BASE DI FUNZIONAMENTO DEI VARCHI	7
8.	REQUISITI.....	8
8.1.	REQUISITI FUNZIONALI	8
8.2.	REQUISITI TECNICI.....	8
9.	SPECIFICA DEI SERVIZI	9
9.1.	SERVIZI PER LISTE E CICLI	10
9.1.1.	LISTSERVICE - GETWORKCYCLES	10
9.1.2.	LISTSERVICE - GETWHITELIST	12
9.1.3.	LISTSERVICE – FINDINLISTS	14
9.1.4.	LISTSERVICE - GETBLACKLIST.....	16
9.2.	SERVIZI DI INSERIMENTO	18
9.2.1.	VIOLATIONSSERVICE – INSERTSIGNAL.....	18
9.2.2.	ALARMSSERVICE - INSERTALARMS.....	20
10.	RIFERIMENTI	21

1. Riepilogo delle modifiche

N.	Versione	Stato	Data	Autore	Approvato	Tipo di modifica
1	1.0a	Rilasciato	11/12/2013	Lorenzo Tramuto		Bozza specifica
2	1.0b	Rilasciato	12/12/2013	Lorenzo Tramuto	Andrea Mennini	Emissione specifica
3	1.01	Elaborato	24/11/2014	Lorenzo Tramuto	Lauro Zuffi	Modifica servizi Web
4	1.02	Elaborato	31/12/2014	Stefano Morelli	Lauro Zuffi	Aggiornamento ad emissione WSDL
5	2.0	Rilasciato	08/01/2015	Lorenzo Tramuto	Andrea Mennini	Emissione
6	2.0	Rilasciato	08/11/2022	Lorenzo Tramuto	Stefano Cattabriga	Cambio formattazione

Tabella 1 Riepilogo delle modifiche.

Riferimento per l'amministrazione di stato e versione:

Stato:

Elaborato ("Processed") il documento è in corso di elaborazione

Rilasciato ("Released") il documento è stato verificato e rilasciato dal controllo qualità; può essere modificato solo se viene aggiornato il numero di versione.

Versioni:

Presentano due fasi. I documenti accettati ricevono il successivo numero intero di versione.

00-01, 00-02 ecc. versioni non rilasciate, con stato "**Elaborato**"

01 prima versione rilasciata con stato "**Rilasciato**"

01-01, 01-02 ecc. versioni che integrano la versione 01-00 e hanno stato "**Elaborato**"

02 seconda versione rilasciata con stato "**Rilasciato**"

2. Lista delle abbreviazioni

Abbreviazione	Descrizione
CBO	Commercial Back Office – Programma di gestione permessi del Comune
CPS	Charge Point Server
FTPS	File Transfer Protocol Secured
MCTC	Motorizzazione Civile Trasporti in Concessione
OBO	Operational Back Office – Programma di gestione tecnologica della ZTL
OCR	Optical Character Recognition
PM	Polizia Municipale
PRA	Pubblico Registro Automobilistico
MCTC	Motorizzazione civile
SCP	Secure CoPy
SFTP	SSH File Transfer Protocol
SSH	Secure SHell
VDR	Vehicle Detection and Registration
XML	eXtensible Markup Language
XSD	Xml Schema Definition
XSLT	Xml Stylesheet Language Trasformation
ZTL	Zona a Traffico Limitato
WSDL	Web Services definition Language
SOAP	Simple Object Access Protocol
ACK	segnale di Acknowledge emesso in risposta alla ricezione di un'informazione completa
PMV	Pannello a messaggio variabile

3. SCOPO DEL DOCUMENTO

Questo documento descrive i requisiti dei servizi di interscambio dati tra SirioWeb, applicativo per il controllo degli accessi veicolari nelle zone a traffico limitato della città, di proprietà della ditta MotuS21 SRL ed un sistema esterno di controllo accessi installato su qualsiasi piattaforma compatibile con la tecnologia di scambio dati adottata su SirioWeb.

Viene fornita una introduzione all'architettura implementata, la specifica funzionale e tecnica dei servizi esposti da SirioWeb e una descrizione dettagliata su come tali servizi devono essere invocati.

I capitoli e i contenuti possono essere quindi riassunti nei seguenti punti:

- Introduzione ai servizi disponibili
- Introduzione all'architettura d'interscambio dati
- Requisiti funzionali e tecnici
- Specifica dei servizi remoti

4. AMBITO DEL PROGETTO

In questa integrazione i servizi di interscambio dati sono messi a disposizione dall'applicativo SirioWeb al fine di consentire a sistemi terzi:

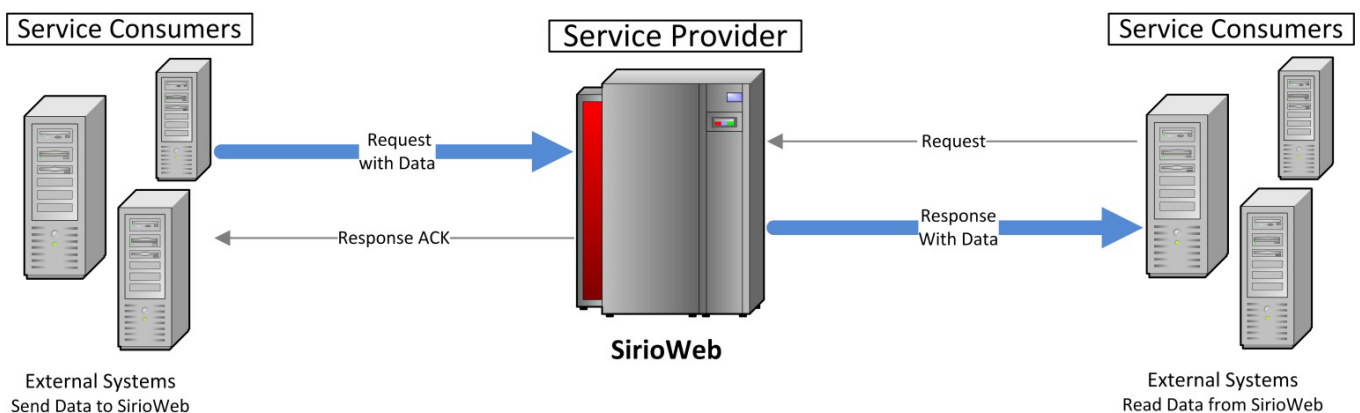
- Il recupero dei dati relativi ai cicli di lavoro dell'impianto di controllo ZTL per la programmazione oraria dell'attivazione o la disabilitazione di ogni varco, ovvero:
 - Calendari runtime limitatamente ai 15 giorni successivi al momento della generazione delle informazioni
 - Cicli di lavoro associati ai calendari
 - Indicazioni di stati di emergenza per pannelli a messaggio variabile
- L'inserimento su SirioWeb di dati ed immagini relativi alle segnalazioni di transito recepiti dal sistema esterno e delle semplici informazioni sui transiti per fini statistici;
- La trasmissione su SirioWeb e l'inserimento nella base dati di allarmi, eventi e stati generati e indicazioni di pannelli a messaggio variabile gestiti dal sistema software esterno per la diagnostica dell'intero impianto;
- Il recupero delle liste bianche, cioè le targhe connesse ad un permesso valido di accesso in ZTL;
- Il recupero delle liste nere, cioè targhe da tenere sotto stretta osservazione per l'eventuale invio immediato del passaggio;

L'architettura di interscambio dati scelta è quella per cui SirioWeb fornisce una interfaccia per l'esposizione di servizi web remoti che possono essere richiesti da un software esterno, sia nel caso di dati in uscita che in ingresso da e verso SirioWeb; nessun movimento di dati con l'applicativo esterno è scatenato da SirioWeb o da uno dei suoi moduli applicativi.

I sistemi esterni devono essere in grado di sviluppare la logica di interscambio e di interoperabilità per quanto riguarda l'invocazione dei servizi esposti da SirioWeb, la gestione dei flussi di dati in input, la gestione dei flussi di dati in output, la trascodifica dei dati tra SirioWeb ed il sistema software esterno.

5. INTRODUZIONE ALL'ARCHITETTURA

In questa sezione viene descritta l'architettura di interscambio dati tra SirioWeb ed i sistemi esterni. Nella seguente figura viene descritta l'architettura di come interagiscono i sistemi coinvolti nell'interscambio dati.



Il flusso dei dati previsto è bidirezionale e sincrono, SirioWeb è il *provider* di servizi mentre i sistemi esterni sono i *consumer* dei servizi esposti da SirioWeb.

Ognuno dei sistemi coinvolti nell'interscambio dati ha uno strato applicativo in cui vengono descritte le interfacce e i servizi di decodifica che permettono l'integrazione tra i sistemi. Lato provider deve essere implementato uno strato che espone le interfacce (WSDL) dei servizi, mentre lato consumer deve essere implementato uno strato di decodifica per il risultato delle richieste di servizio.

L'interscambio dati avviene via protocollo SOAP HTTP (o HTTPS), o altro protocollo di scambio dati (FTP, SFTP), questo permette di ovviare alle problematiche legate alle limitazioni imposte dai *firewall* sulle connessioni TCP/IP permesse; ciò implica la necessità di effettuare connessioni verso le porte TCP 80/8443 (o altre porte concordate) del server SirioWeb ospitante i servizi.

I dati veicolati all'interno dei messaggi SOAP devono rispettare lo standard XML e seguono delle strutture ben definite su files di schema XSD messi a disposizione da MotuS21 S.r.l.

La trattazione che segue nel presente documento riguarda esclusivamente la descrizione del protocollo di scambio dati mediante SOAP http/HTTPS.

Per quanto riguarda le informazioni da veicolare dal sistema esterno verso SirioWeb, queste saranno direttamente inviate ai servizi web appositi esposti dal service provider nei messaggi SOAP di request.

I servizi esposti sono di tipo sincrono e a seguito di un messaggio request di invio dati, seguirà sempre un messaggio SOAP response con l'esito dell'operazione di elaborazione ed eventuale persistenza dei dati.

Una architettura con questa tipologia permette quindi un controllo delle informazioni ricevute ed una risposta descrittiva per eventuale correzione di dati errati restituita al sistema esterno inviante.

6. CONFIGURAZIONI INIZIALI

La corretta integrazione del Sistema SirioWeb con altri software esterni richiede una configurazione iniziale di alcune entità esplicitamente considerate conosciute e condivise tra i sistemi eterogenei.

Esempio di queste entità sono in particolare l'identificativo dell'anagrafica di posizionamento geografico dei varchi, i codici che identificano la classe dimensionale dei veicoli.

7. PRINCIPI BASE DI FUNZIONAMENTO DEI VARCHI

Elenchiamo i principi base di funzionamento dei varchi per i quali i protocolli saranno successivamente progettati; qualora i varchi non supportino tali principi di funzionamento sarà onere del fornitore virtualizzare tali funzionalità attraverso dispositivi o applicazioni intermedie che sopperiscano alla mancanza di tali requisiti.

- Nel funzionamento normale è disponibile una connettività sostanzialmente continua, persistente o a richiesta del chiamante tra il Posto Centrale e i Varchi Remoti
- Un'unità remota di controllo varco opera ed agisce sostanzialmente in modalità realtime e la bufferizzazione delle segnalazioni avviene solo a scopo di archivio temporaneo in attesa che siano disponibili le condizioni di connettività verso il Posto Centrale.
- Ogni transito anche qualora non sanzionabile lascia una traccia informativa a fini statistici; tale informazione sarà aggregata solo nel processo elaborativo del Posto Centrale.
- Qualora le informazioni inerenti alle politiche sanzionatorie non siano disponibili sull'unità remota la politica che si attiverà di default sarà l'archiviazione di tutte le segnalazioni per l'invio al Posto Centrale che provvederà successivamente al processamento dei transiti; pertanto non è previsto l'invio di informazioni per l'elaborazione off-line o di eventi remoti sulle unità remote.

- Le informazioni fornite all'unità remota devono garantire un'adeguata autonomia di funzionamento in merito al processo sanzionatorio.
- L'unità remota è in grado di notificare il suo stato di funzionamento in modalità non sollecitata; tale notifica di stato/allarmi può essere completa o contenere solo le variazioni di stato.
- Le unità remote con funzionalità avanzate sono in grado di gestire le informazioni del Posto Centrale in merito alle politiche di sanzionamento in esso definite.
- Qualora la comunicazione varco – Posto Centrale avvenga su rete pubblica deve essere garantita riservatezza ed integrità delle informazioni.
- Ogni sistema remoto deve essere sincronizzato all'orologio del Posto Centrale attraverso protocollo standard NTP a meno che non siano disponibili risorse indipendenti che però devono garantire pari allineamento.
- Il colloquio tra varco di terze parti e il Posto Centrale SirioWeb può avvenire direttamente e l'utilizzo di dispositivi, gateway, posti centrali, db centrali di terze parti costituisce una opzione non vincolante che il fornitore di tale unità remota assume per sua scelta e comodità.

8. REQUISITI

8.1. REQUISITI FUNZIONALI

R1 - Flusso dei dati unidirezionale bidirezionale e simmetrico: SirioWeb espone dei servizi che vengono consumati da sistemi esterni. Ad ogni messaggio ricevuto (request) corrisponde sempre un messaggio inviato (response).

R2 – Servizi di recupero: i servizi di recupero devono permettere la ricerca delle informazioni in base a dei filtri. I campi filtro possono essere obbligatori oppure opzionali.

R3 – Response: i servizi di inserimento restituiscono al chiamante una informazione relativamente ad ogni messaggio ricevuto ed ai dati in esso contenuti. E' previsto che i servizi operino con modalità sincrona.

R4 - Gestione dell'errore: i servizi devono restituire un Fault in seguito ad un errore. Un servizio può restituire uno o più Fault in base alle diverse situazioni di errore. Nel caso di errore semantico sulle informazioni ricevute, il service provider inserisce una descrizione di errore dettagliata nel messaggio di response in modo da consentire un logging più accurato per la diagnostica dell'eventuale problema e la possibilità di correzione dei dati da parte del sistema esterno.

8.2. REQUISITI TECNICI

R5 – Conformità a SW-I Basic Profile: i servizi esposti con protocollo SOAP sono conformi alle specifiche SW-I Basic Profile. Il documento "SW-I Basic Profile" [SWBP] è una guida all'uso degli standard fondamentali, in particolare SOAP, WSDL, XML, XML Schema e HTTP per garantire l'interoperabilità tra diversi software funzionanti anche su sistemi ed

ambienti eterogenei.

R6 – Stile di interoperabilità Document/literal wrapped: per la descrizione dei file WSDL dei servizi esposti è utilizzato lo stile *Document/literal wrapped*. Questo stile obbliga allo scambio di messaggi SOAP conformi alle specifiche SW-I Basic Profile e permette l'interoperabilità di software funzionanti su piattaforme di diversa concezione e framework di sviluppo.

R7 – Sicurezza: i servizi possono garantire la sicurezza a livello di trasporto usando i protocolli HTTPS ed SFTP, così da garantire trasferimenti riservati di dati nel web. E' supportato anche l'eventuale mutuo scambio di certificati digitali in modo da permettere l'autenticazione reciproca tra provider e consumer (mutua autenticazione).

R8 – Trasmissione binaria: il servizio di ricezione delle segnalazioni è compatibile con la tecnologia MTOM/XOP [SWMX] che consente di comprimere del 33% la dimensione delle immagini inserite nei documenti XML come dati binari codificati in base64. Nel file allegato (signal_raw.txt) è presentato un esempio di come siano trasmessi i dati in questa modalità di trasmissione)

Nel caso il client non attivi tale modalità di comunicazione le immagini saranno trasmesse con la consueta codifica base64.

9. SPECIFICA DEI SERVIZI

Nel presente capitolo vengono descritti i servizi, le interfacce esposte da SirioWeb e un manuale d'uso per il chiamante (consumer). Il pattern che viene utilizzato per questa descrizione si suddivide nelle seguenti voci:

- Nome servizio: nome del servizio esposto da SirioWeb, raggruppa diverse operazioni.
- Operation name: nome dell'operazione invocata per uno specifico servizio.
- Descrizione: una breve descrizione del servizio.
- Invocazione: Parametri da impostare per l'invocazione del metodo ed istruzioni su come può essere chiamato dal Consumer.
- WSDL: file WSDL di interfaccia contenente la struttura da rispettare per i documenti XML usati nei messaggi request e response del servizio.
- Response tag: Nome dell'elemento XML usato nel body del messaggio response SOAP ritornato verso il sistema esterno.
- Esempio: un esempio semplificato di messaggi Request e Response.

SirioWeb espone un'unica interfaccia WSDL per ogni servizio (più metodi per ogni servizio).

Ogni operazione deve essere richiamata utilizzando il servizio apposito; questo significa che per tutti i servizi web riportati nei paragrafi successivi l'invocazione va eseguita usando come endpoint:

`https://addressServerSirioWeb:8443/ JSirio-JSirioEjb/[Nome Servizio]`

9.1. SERVIZI PER LISTE E CICLI

9.1.1. LISTSERVICE - GETWORKCYCLES

Nome servizio	ListService
Operation Name	GetWorkCycles
Descrizione	<p>Il servizio permette al chiamante di recuperare le informazioni sui cicli di lavoro, e quindi del comportamento da seguire da parte di ogni varco remoto, già presenti e configurati in SirioWeb.</p> <p>Le informazioni rilasciate da SirioWeb descrivono le fasi temporali in cui ogni varco deve lavorare in modalità sanzionatoria o di solo conteggio statistico, e queste hanno una struttura dati tale da dare evidenza, con risoluzione di 15 minuti per volta, del comportamento configurato per i successivi 15 giorni dal momento in cui viene effettuata la richiesta. Informazioni riguardo interventi temporanei al di fuori dei normali cicli di lavoro sono evidenti mediante l'uso degli attributi "emergencyState".</p> <p>Tipi di informazioni esposte:</p> <ul style="list-style-type: none"> • Ciclo di lavoro di una generica giornata. • Calendari (limitatamente ai successivi 15 giorni rispetto alla data della richiesta) utilizzati per ogni varco con l'associazione del ciclo di lavoro da utilizzare in quel giorno. <p>Nota: è possibile che siano presenti più "records" riferiti allo stesso giorno ma con intervalli orari di validità diversi.</p>
Invocazione	<p>E' possibile specificare il tag opzionale List<GateID> per avere le informazioni su un sottoinsieme specifico di varchi; qualora non sia specificato espressamente l'elenco dei varchi nella richiesta il sistema produrrà le informazioni per tutto l'elenco dei varchi registrati.</p> <p>La response contiene per ogni segmento ripetuto WorkCycle la lista di segmenti Band che specificano con risoluzione temporale di 15 minuti se in quell'intervallo di tempo l'eventuale varco collegato al ciclo di lavoro è attivo o meno.</p>
WSDL	list.wsdl

Response tag	cycleExport
Esempio Request	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:lis="http://ws.ext.sirio/listService"> <soapenv:Header/> <soapenv:Body> <lis:getWorkCycles> <!--Optional:--> <gateList> <!--Zero or more repetitions:--> <gateId>1</gateId> <gateId>2</gateId> </gateList> </lis:getWorkCycles> </soapenv:Body> </soapenv:Envelope> </pre>
Esempio Response	vedi allegato cycle.xml (in fase di rieditazione)

9.1.2. LISTSERVICE - GETWHITELIST

Nome servizio	ListService
Operation Name	GetWhiteLists
Descrizione	<p>Il servizio permette al chiamante di recuperare l'elenco della lista bianca, cioè le informazioni dei permessi di autorizzazione al transito in ZTL.</p> <p>Le informazioni contenute nel messaggio di risposta riguardano le targhe con permesso di accesso e la loro validità temporale, le fasce orarie con risoluzione temporale di 15 minuti di validità del permesso in lista bianca, l'associazione tra il tipo di giorno e le fasce orarie configurate su SirioWeb che compongono la definizione di un contrassegno orario.</p> <p>Queste informazioni sono incrociate con il tipo giorno configurato nel calendario per riconoscere se un transito in un preciso momento della giornata ha una autorizzazione valida in lista bianca.</p>
Invocazione	<p>La richiesta delle targhe in lista bianca deve essere schedulata sul sistema di terze parti secondo intervalli temporali concordati con il cliente.</p> <p>La lista bianca delle targhe con permesso è filtrata per i soli permessi in corso di validità, senza quelli già cessati o scaduti, ed è relativa ad ogni preciso varco remoto (corrispondenza con la configurazione di SirioWeb dei contrassegni di zona).</p> <p>E' possibile specificare il tag opzionale List<GateID> per avere le informazioni dei permessi su un sottoinsieme specifico di varchi; qualora non sia specificato espressamente l'elenco dei varchi nella richiesta il sistema produrrà le informazioni per tutto l'elenco dei varchi registrati.</p> <p>La risposta corrispondente all'invocazione fornisce tutte le informazioni necessarie per verificare l'autorizzazione al transito.</p> <p>Sostanzialmente le principali informazioni fornite sono:</p> <p>Elenco dei varchi per i quali sono richieste le informazioni di autorizzazione dei transiti con associato il corrispondente calendario.</p> <p>Elenco delle zone di autorizzazione con associato l'elenco dei varchi per i quali tale autorizzazione è valida.</p> <p>Calendario con l'elenco dei tipi giorno da utilizzare nella valutazione dell'autorizzazione e relativi periodi di validità giornalieri.</p>

	<p>Elenco dei contrassegni orari con la fascia oraria da utilizzare in base al tipo giorno.</p> <p>Elenco delle fasce orarie su cui valutare l'autorizzazione al transito. Elenco dei permessi dei veicoli contenenti:</p> <p>permissionTime: codice del contrassegno orario. Corrisponde ad uno dei codici ricercabile nella sezione dei comportamenti orari inseriti nel medesimo messaggio di response</p> <p>permissionZone: codice del contrassegno zona. Corrisponde ad uno dei codici ricercabile nella sezione dei comportamenti di zona inseriti nel medesimo messaggio di response</p> <p>plate: targa del veicolo che possiede il permesso</p> <p>start: data di inizio validità del permesso</p> <p>stop: data di fine validità del permesso</p>
WSDL	list.wsdl
Response tag	wlExport
Esempio Request	<pre> <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:lis="http://ws.ext.sirio/listService"> <soapenv:Header/> <soapenv:Body> <lis:getWhiteLists> <!--Optional:--> <gateList> <!--Zero or more repetitions:--> <gateId>1</gateId> <gateId>2</gateId> </gateList> </lis:getWhiteLists> </soapenv:Body> </soapenv:Envelope> </pre>
Esempio Response	vedi allegato list.xml (in fase di rieditazione)

9.1.3. LISTSERVICE – FINDINLISTS

Nome servizio	ListService
Operation Name	<p>FindInLists</p> <p>Viene reso disponibile un ulteriore servizio che fa invece una ricerca nelle varie liste a fronte di una specifica richiesta del varco. Questo nel caso in cui la tecnologia periferica non preveda un trasferimento completo delle liste oltre che ad esempio per far fronte nel tempo a ricerche in Database molto più complessi anche per scopi diversi dalla sola ricerca della presenza o meno di una autorizzazione all'accesso.</p>
Descrizione	<p>Il servizio permette al chiamante di effettuare una ricerca nelle liste bianche, speciali e nere di SirioWeb, di una targa relativa ad un transito tracciato da un varco remoto.</p> <p>Il metodo controlla la validità dell'eventuale permesso rilasciato per la targa del veicolo, considerando le informazioni di validità temporale, di zona, del tempo di grazia e del ciclo di lavoro configurato per ogni singolo varco.</p>
Invocazione	<p>I dati richiesti nel messaggio di richiesta riguardano:</p> <ul style="list-style-type: none"> • targa • varco di rilevazione • data ed ora del transito <p>Il risultato sarà inerente alla presenza in lista bianca, speciale o nere, e se il passaggio è avvenuto in un momento valido legato al permesso rilasciato.</p> <p>In base a tale esito il varco archiverà il transito per fini sanzionatori o esclusivamente per fini statistici; la restituzione del valore true indicherà che l'archiviazione è a fini sanzionatori.</p>
WSDL	list.wsdl
Response tag	archiveTransit
Esempio Request	<pre><soapenv:Body> <lis:findInList> <!--Optional:--> <plate>AB123DF</plate></pre>

	<pre> <gateId>1</gateId> <!--Optional:--> <dateTime>2014-12-30T12:30:00</dateTime> </lis:findInList> </soapenv:Body> </soapenv:Envelope> </pre>
Esempio Response	<pre> <env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"> <env:Header/> <env:Body> <ns1:findInListResponse xmlns:ns1="http://ws.ext.sirio/listService"> <archiveTransit>true</archiveTransit> </ns1:findInListResponse> </env:Body> </env:Envelope> </pre>

9.1.4. LISTSERVICE - GETBLACKLIST

Nome servizio	ListService
Operation Name	GetBlackLists
Descrizione	<p>Il servizio permette al chiamante di recuperare l'elenco della lista nera, cioè le targhe da tenere sotto osservazione. Su SirioWeb è possibile configurare l'invio di una email con l'avvertimento immediato dell'avvenuto transito, sotto un qualsiasi varco del veicolo con targa presente in lista nera.</p> <p>Qualora non esistano delle particolari politiche di gestione dei transiti riscontrati in lista nera sui varchi, sarà necessario gestire i transiti sul varco come transiti ordinari da sanzionare, inviando cioè al posto centrale sia le informazioni testuali sul transito che le immagini associate, indipendentemente dal fatto che tale transito sia avvenuto in un periodo di non sanzionabilità del varco stesso o che sia stato identificato come in lista bianca. Sarà poi onere del posto centrale attivare tutte le azioni di notifica in merito alla segnalazione in lista nera appena il transito sarà trasmesso al posto centrale.</p>
Invocazione	<p>Nel messaggio di request sarà presente un elemento <i>blackListPlate</i> per ogni record in lista nera da esportare verso il sistema esterno.</p> <p>Le informazioni di ogni elemento <i>blackListPlate</i> in lista nera sono:</p> <p>plate: targa da tenere sotto osservazione begin: data di inizio periodo osservazione end: data di fine periodo di osservazione</p>
WSDL	list.wsdl
Response tag	BlackList
Esempio Request	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:lis="http://ws.ext.sirio/listService"> <soapenv:Header/> <soapenv:Body> <lis:getBlackLists/> </soapenv:Body> </soapenv:Envelope></pre>

**Esempio
Response**

```

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Header/>
  <env:Body>
    <ns1:getBlackListsResponse
xmlns:ns1="http://ws.ext.sirio/listService">
      <blackList>
        <blackListPlate>
          <plate>AB123AB</plate>
          <begin>2014-12-01</begin>
        </blackListPlate>
        <blackListPlate>
          <plate>DF123DF</plate>
          <begin>2014-12-07</begin>
          <end>2014-12-27</end>
        </blackListPlate>
        <blackListPlate>
          <plate>DR456FG</plate>
        </blackListPlate>
      </blackList>
    </ns1:getBlackListsResponse>
  </env:Body>
</env:Envelope>
  
```

9.2. SERVIZI DI INSERIMENTO

9.2.1. VIOLATIONSSERVICE – INSERTSIGNAL

Nome servizio	ViolationsService
Operation Name	InsertSignal
Descrizione	<p>Il servizio permette al chiamante di inserire su SirioWeb una o più segnalazioni di transito, o di far incrementare i dati statistici di conteggio traffico se i transiti sono effettuati in momenti in cui un varco risulta, da configurazione, funzionante in modalità non sanzionatoria.</p>
Invocazione	<p>All'interno del segmento devono essere presenti tutti i tag valorizzati con le informazioni sul transito o eventuale violazione. Dalla specifica sul file XSD è possibile verificare la tipologia dei dati e quali di questi sono opzionali o obbligatori.</p> <p>Per ogni violazione è possibile inserire più immagini di diverso tipo tra loro correlate:</p> <ul style="list-style-type: none"> • Immagine transito per OCR (typeImage=OCR) • Immagine di dettaglio della targa (typeImage=PLATE) • Immagine del veicolo in transito (typeImage=FULL) • Immagine di contesto (typeImage=CONTEXT) <p>Sono codificate ed inseribili le seguenti categorie dimensionali di veicoli: "UNKNOWN" "MOTORBIKE" "MOTOCYCLE" "AUTOVEHICLE" "BUS" "TRUCK"</p> <p>Le nazionalità attualmente codificate sono presenti nel WSDL.</p> <p>Nella response del messaggio ci sarà l'esito dell'operazione di inserimento. Se l'esito è negativo sarà presente anche una descrizione testuale dell'errore ed un codice numerico per l'identificazione univoca del problema.</p> <p>L'inserimento dei dati sul transito senza le relative immagini sarà considerato dalla piattaforma SirioWeb come dato da gestire solamente a livello di incremento dei contatori statistici sul traffico registrato.</p>
WSDL	signal.wsdl
Response tag	SignalResult
Esempio Request	Vedi allegati signal.xml e signal_raw.txt
Esempio Response	<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

	<pre> <env:Header/> <env:Body> <insertSignalResponse xmlns="http://ws.ext.sirio/signalInsert"> <run>true</run> <error>0</error> <description/> </insertSignalResponse> </env:Body> </env:Envelope> </pre>
	<p>error = 0 → inserimento corretto Run = true -> il Sistema esterno può continuare ad inoltrare transiti verso SirioWeb</p>

9.2.2. ALARMSERVICE - INSERTALARMS

Nome servizio	AlarmsService
Operation Name	InsertAlarms
Descrizione	<p>Il servizio permette al chiamante di comunicare alla piattaforma centrale SirioWeb gli allarmi, gli eventi e lo stato complessivo di ogni varco remoto collegato all'impianto di telcontrollo del traffico.</p> <p>Il sistema centrale permette di registrare sia i cambi di stato su singolo evento, sia di ricevere sempre l'intero array di stato complessivo del varco, registrando comunque attivazione e rientro di ogni singolo allarme.</p> <p>Tra i dati sullo stato del varco è possibile gestire anche la notifica del messaggio mostrato nel pannello a messaggio variabile (PMV), se presente.</p> <p>E' onere del sistema terzo esterno schedulare l'invio verso SirioWeb di tali informazioni ad intervalli temporali regolari, con frequenza configurabile dal lato varco remoto.</p> <p>In questo modo risulta implementata anche una funzione di tipo "Alive notify"</p>
Invocazione	<p>La ricezione di questi messaggi può essere effettuata o periodicamente con lo stato generale del varco in un determinato momento, piuttosto che al momento del cambio di stato di un allarme, quindi sia al momento in cui si verifica un problema su un varco, sia quando questo rientra.</p> <p>Nella response del messaggio ci sarà l'esito dell'operazione di inserimento. Se l'esito è negativo sarà presente anche una descrizione testuale dell'errore ed un codice numerico per l'identificazione univoca del problema.</p>
WSDL	alarm.wsdl
Response tag	AlarmsResponse
Esempio Request	vedi allegato alarm.xml
Esempio Response	<pre> </ns1:insertAlarmResponse> </env:Body> </env:Envelope> error = 0 → inserimento corretto Run = true; al momento è l'unico tipo di valore restituito </pre>

10. RIFERIMENTI

[SWBP] <http://www.ws-i.org/deliverables/workinggroup.aspx?wg=basicprofile>

[SWSP] <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>

[SWMX] <http://www.w3.org/TR/soap12-mtom/>

[SWMX] http://en.wikipedia.org/wiki/Message_Transmission_Optimization_Mechanism