



**REGIONE BASILICATA**  
**DIPARTIMENTO INFRASTRUTTURE,**  
**OO.PP. E MOBILITA'**

**Ufficio Difesa del Suolo di Potenza**

**SPECIFICHE TECNICHE INTERFACCIA**

“Informatizzazione dell’iter procedurale e dei controlli  
relativi alle costruzioni in zona sismica”

Allegato 1.B



INF-1: Specifiche tecniche di Interfaccia

# **INF-1: Specifiche Tecniche di Interfaccia**

*Versione 1.1*

INDICE

<i>Modifiche Documento</i> .....	3
<i>Introduzione</i> .....	4
<i>L'uso del Componente di Integrazione della Porta di Dominio</i> .....	4
<i>Modalità d'uso trasparente dei Servizi</i> .....	5
<i>Uso del Servizio IntegrationManager</i> .....	5
<i>Contenuti delle Risposte Applicative</i> .....	6
<i>La gestione degli errori nell'Interazione tra Servizio Applicativo e Porta Delegata</i> .....	6
<i>Gestione degli errori nell'Interazione tra Porta Applicativa e Servizio Applicativo</i> .....	8
<i>Ulteriori aspetti di integrazione nei Profili Asincroni</i> .....	9
<i>Allegato A: Interfaccia WSDL del Servizio IntegrationManager</i> .....	10
<i>Allegato B: Schema risposta OK della PdD</i> .....	20

## Modifiche Documento

Descrizione Modifica	Edizione	Data
Prima versione	1.0	12/01/2007
Limitata al profilo Oneway la gestione degli errori nell'Interazione tra Porta Applicativa e il Servizio Applicativo	1.1	20/02/2007

## Introduzione

Nella specifica SPCoop, un dominio è definito come il confine di responsabilità di un ente o soggetto amministrativo e racchiude al suo interno tutte le applicazioni da esso gestite. Il confine applicativo del Dominio è rappresentato dalla Porta di Dominio (PdD), attraverso la quale devono transitare tutte le comunicazioni da e verso il dominio.

Le Porte di Dominio si parlano tra di loro scambiandosi richieste e risposte in un formato parte della specifica SPCoop, denominato busta eGov. Tuttavia il formato della busta non è parlato nativamente dalle applicazioni, pertanto la Porta di Dominio deve anche occuparsi di convertire le richieste applicative nel formato busta eGov. Facendo riferimento a questa problematica, i compiti della Porta di Dominio vengono solitamente classificati in due componenti: il componente di cooperazione, che riguarda la comunicazione tra le Porte di Dominio e quello di integrazione, che riguarda la comunicazione tra i Servizi Applicativi dell'Ente e la Porta di Dominio.

Il componente di integrazione si differenzia a sua volta in due diversi moduli: la porta delegata e la porta applicativa. In particolare la porta delegata è utilizzata come proxy per l'accesso al servizio destinazione, mentre la porta applicativa deve essere in grado di gestire la consegna dei contenuti delle buste eGov ricevute al corrispondente servizio applicativo interno al dominio destinazione.

Differentemente da quanto avviene per il componente di cooperazione (busta eGov), per il componente di integrazione la specifica SPCoop si limita a presentare un esempio di massima di una sua possibile realizzazione. Al fine di uniformare le interfacce del livello applicativo di ICAR verso il task INF-1, in questo documento vengono quindi presentate le interfacce esposte dalle Porte di Dominio verso i Servizi Applicativi.

## L'uso del Componente di Integrazione della Porta di Dominio

Le interazioni tra i servizi applicativi e le porte delegate e applicative delle Porte di Dominio che li ospitano devono avvenire tramite richieste SOAP su protocollo https. Ogni richiesta del Servizio Applicativo deve essere autenticata dalla Porta tramite certificati X.509.

In generale le interazioni tra i servizi applicativi e la porta di dominio possono avvenire in due modi:

1. modalità trasparente: prevede che il servizio applicativo utilizzi (in caso di porta delegata) o esponga (in caso di porta applicativa) le interfacce applicative native dei servizi, esattamente come registrate negli accordi di servizio; in tal caso la Porta di Dominio agisce come un proxy trasparente con funzionalità di imbustamento e sbustamento eGov dei messaggi applicativi; utilizzando questa modalità, gli applicativi potranno continuare ad operare esattamente come se stessero interagendo direttamente con il servizio applicativo dell'altro Ente;
2. uso del Servizio "Integration Manager" della PdD: prevede di utilizzare le interfacce di un apposito web service di Integrazione, messo a disposizione dalla Porta di Dominio per la spedizione e/o la ricezione di messaggi applicativi da parte dei servizi applicativi del proprio Dominio di Servizi.

## Modalità d'uso trasparente dei Servizi

Questa modalità prevede che il servizio applicativo utilizzi (in caso di porta delegata) o esponga (in caso di porta applicativa) le interfacce applicative native dei servizi, così come registrate negli accordi di servizio; in tal caso la Porta di Dominio agisce come un proxy SOAP trasparente con funzionalità di imbustamento e sbustamento eGov dei messaggi applicativi; utilizzando questa modalità, gli applicativi potranno quindi continuare ad operare esattamente come se stessero interagendo direttamente con il servizio applicativo dell'altro Ente.

L'invocazione della porta delegata in modalità trasparente può essere realizzata tramite gli strumenti del linguaggio di programmazione nativo del servizio applicativo, utilizzando ad esempio stub creati tramite il proprio ambiente di sviluppo Web Services (ad esempio wsdl2java in Axis), facendo riferimento direttamente al WSDL del servizio erogatore registrato nell'accordo di servizio. In questo caso la principale modifica rispetto all'invocazione dell'effettivo servizio destinazione sarà la URL utilizzata per l'invocazione http, che dovrà essere quella corrispondente alla porta delegata del servizio esposta dalla PdD.

## Uso del Servizio IntegrationManager

Prevede di utilizzare le interfacce di un apposito web service di Integrazione, messo a disposizione dalla Porta di Dominio per la spedizione e/o la ricezione di messaggi applicativi da parte dei servizi applicativi del proprio Dominio di Servizi. L'interfaccia WSDL completa dell'Integration manager è disponibile in allegato "A". A mero titolo descrittivo si mostra di seguito l'interfaccia esposta dal Web Service, espressa in linguaggio java:

```
public interface IntegrationManager {
    public String[] getAllMessagesId()
    public String[] getAllMessagesIdByService(String servizio, String
azione)
    public SPCoopMessage getMessage(String idEGov)
    public SPCoopMessage getAndDeleteMessage(String idEGov)
    public SPCoopMessage getMessageByReference(String riferimentoMsg)
    public SPCoopMessage getAndDeleteMessageByReference(String
riferimentoMsg)
    public SPCoopMessage getNextMessage()
    public SPCoopMessage getNextAndDeleteMessage()
    public void deleteMessage(String idEGov)
    public void deleteAllMessages()
    public SPCoopMessage invocaPortaDelegata(String portaDelegata,
SPCoopMessage msg)
}
```

L'oggetto SPCoopMessage contiene l'ID del messaggio, il servizio e l'azione SPCoop e un array di byte che rappresenta il messaggio SOAP vero e proprio:

```
public class SPCoopMessage implements java.io.Serializable {
    public java.lang.String getID();
    public void setID(java.lang.String ID);
    public java.lang.String getDestinatario();
    public void setDestinatario(java.lang.String destinatario);

    public java.lang.String getServizio();
}
```

```
public void setServizio(java.lang.String servizio);

public java.lang.String getAzione();
public void setAzione(java.lang.String azione);

public byte[] getMessage();
public void setMessage(byte[] message);
}
```

## Contenuti delle Risposte Applicative

Nel caso del profilo Sincrono, le risposte (ricevute dalla PdD nel caso di invocazioni della porta delegata o inviate dalla PdD nel caso di invocazioni dei Servizi Applicativi abbinati alle Porte Applicative), sono direttamente generate dai Servizi Applicativi erogatori e trasparentemente restituiti dalla PdD ai servizi fruitori. La stessa cosa succede per i profili Asincroni Simmetrico e Asimmetrico, che corrispondono concettualmente ad una coppia di interazioni sincrone correlate tra loro.

Nel caso del profilo Oneway, invece, la PdD mittente restituisce la risposta al servizio fruitore immediatamente, senza attendere di ricevere una risposta dal servizio applicativo erogatore del servizio. In questo caso il servizio fruitore riceverà quindi una risposta appositamente creata dalla PdD e conforme allo schema xsd in allegato B.

## La gestione degli errori nell'Interazione tra Servizio Applicativo e Porta Delegata

In questo caso il Servizio Applicativo invoca un servizio della PdD. In funzione del fatto che si usi la modalità di invocazione trasparente o i servizi dell'IntegrationManager, cambia il modo in cui le condizioni di errore vengono gestite dal servizio applicativo.

Nel caso in cui si usino i servizi dell'IntegrationManager le condizioni di errore saranno restituite all'interno di una eccezione gestita dal servizio IntegrationManager, il cui formato è rappresentato di seguito in linguaggio java.

```
public class SPCoopException {
    public java.lang.String getCodiceEccezione();
    public void setCodiceEccezione(java.lang.String codiceEccezione);

    public java.lang.String getDescrizioneEccezione();
    public void setDescriptionEccezione(java.lang.String descrizioneEccezione);

    public java.lang.String getIdentificativoFunzione();
    public void setIdentificativoFunzione(java.lang.String
                                           identificativoFunzione);

    public java.lang.String getIdentificativoPorta();
    public void setIdentificativoPorta(java.lang.String identificativoPorta);

    public java.lang.String getOraRegistrazione();
    public void setOraRegistrazione(java.lang.String oraRegistrazione);

    public java.lang.String getTipoEccezione();
    public void setTipoEccezione(java.lang.String tipoEccezione);
}
```

}

Nel caso in cui si utilizzi la modalità trasparente, sarà invece possibile, in caso di eccezione SOAPFault, testare il campo FaultActor per riconoscere e gestire i casi di errore dovuti all'interazione con la porta di dominio (valore "it.icar.pdd") da quelli puramente applicativi. Nel frammento di codice seguente vediamo, a titolo di esempio, come gestire questa situazione nel caso specifico del linguaggio java usando Axis 1.4 come web-services engine.

```
try {
    HelloWSServiceLocator locator = new HelloWSServiceLocator();
    locator.setHelloWorldEndpointAddress("http://pdd/icar/PD/GetDate");
    HelloWS port = locator.getHelloWorld();
    String msg = port.getDate();
} catch (AxisFault e) {
    if("it.icar.pdd".equals(e.getFaultActor())){
        System.out.println("Ricevuto Messaggio di Errore Applicativo[" +
            e.getFaultCode() + "]:");

        System.out.println(e.getFaultString());
    }else{
        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();
}
```

Come evidenziato nell'esempio, il particolare codice di eccezione generato dalla Porta di Dominio potrà essere ottenuto tramite il campo FaultCode del messaggio di Fault SOAP.

In entrambi gli approcci, in caso di errore, la Porta di Dominio restituirà gli stessi codici di errore, elencati nella tabella seguente.

FaultCode	FaultString
ICAR_IT_500	Porta di Dominio Temporaneamente non Disponibile
ICAR_IT_401	Porta Delegata Inesistente
ICAR_IT_402	Autenticazione Fallita
ICAR_IT_403	Pattern Ricerca Porta Delegata Non Valido
ICAR_IT_404	Autorizzazione Fallita
ICAR_IT_405	Servizio SPCoop abbinato alla Porta Delegata Inesistente
ICAR_IT_406	Nessun Messaggio disponibile per il Servizio Applicativo
ICAR_IT_407	Messaggio Richiesto Inesistente
ICAR_IT_4XX	Generico errore client



I servizi applicativi sono tenuti a gestire tutti gli errori generati dalla PdD, gestendo i messaggi che hanno generato errore per una successiva spedizione. In particolare, in caso di errore ICAR\_IT\_500, il servizio applicativo è tenuto a sospendere temporaneamente per un tempo calcolato in maniera random, la spedizione di messaggi, in modo da favorire il decongestionamento dell'infrastruttura.

## Gestione degli errori nell'Interazione tra Porta Applicativa e Servizio Applicativo, nel caso di profilo Oneway.

Nel caso in cui un servizio applicativo eroghi un servizio SPCoop con profilo OneWay, e si verifichi un errore nell'invocazione del servizio applicativo da parte della Porta di Dominio, si pone il problema di saper distinguere tra gli errori applicativi, da interpretare comunque come un successo nella consegna del messaggio, e gli errori propri della transazione di consegna, da interpretare quindi come un fallimento nella consegna del messaggio al Servizio Applicativo. Solo in quest'ultimo caso, la PdD dovrà infatti effettuare successivi tentativi di riconsegna del messaggio.

Perché quindi la PdD sia in grado di distinguere tra queste due situazioni, è necessaria la collaborazione del Servizio Applicativo, il quale dovrà permettere di distinguere tra i codici di errore applicativi e quindi da ignorare (nel caso di profili asincroni) o da propagare al fruitore (nel caso di profili sincroni) da codici di errore relativi alla gestione del messaggio e per i quali sarà necessaria quindi una successiva rispedizione.

A tal fine si richiede che i servizi applicativi invocati da una Porta Applicativa relativa ad un servizio SPCoop con profilo Sincrono debbano restituire in caso di fallimento nella gestione del messaggio, un SOAP Fault con FaultActor avvalorato a "it.icar.pdd" e codici di errore come da tabella seguente.

FaultCode	FaultString
ICAR_IT_500	Servizio Applicativo Temporaneamente non Disponibile
ICAR_IT_402	Autenticazione Fallita
ICAR_IT_404	Autorizzazione Fallita
ICAR_IT_4XX	Errore Client Generico

Una PdD conforme alla specifica ICAR dovrà quindi rispedire il messaggio se e solo se si verifica uno dei seguenti casi durante la sottomissione del messaggio:

1. errore generale di invocazione del servizio, quindi una eccezione relativa al canale di trasporto http (ad esempio server non raggiungibile);
2. restituzione di un soap fault con FaultActor='it.icar.pdd' e FaultCode=ICAR\_IT\_500.

Nota: questo stesso problema non si pone nel caso degli altri profili (quindi sia per il profilo Sincrono che per le due richieste in gioco nei due profili Asincroni), in cui la risposta ottenuta dal server dovrà comunque essere girata alla Porta di Dominio in attesa sincrona.

## Ulteriori aspetti di integrazione nei Profili Asincroni

I profili di collaborazione Asincroni, sia Simmetrici che Asimmetrici, al contrario degli altri profili previsti in SPCoop (Oneway e Sincrono) richiedono due interazioni, una richiesta di servizio ed una risposta del servizio che viene ritornata tramite una nuova interazione. Le buste eGov prodotte sono quindi quattro (due per ogni interazione) e vengono correlate tra di loro attraverso l'identificativo eGov.

Questo fatto comporta che i servizi applicativi devono necessariamente scambiare con la porta di dominio l'informazione relativa l'id eGov, necessaria per la correlazione tra la richiesta e la successiva risposta, ad essa correlata.

Nel caso si usi il servizio di IntegrationManager della Porta di Dominio è possibile utilizzare le interfacce di questo servizio per lo scambio dell'id eGov necessario per la correlazione.

Nel caso in cui si utilizzi la modalità trasparente per l'invocazione dei servizi Asincroni, sarà invece necessario utilizzare l'header 'SPCoop.ID' del trasporto http per lo scambio dell'id eGov necessario per la correlazione, dovendo l'interfaccia applicativa rimanere quella originale del servizio applicativo invocato.

## Allegato A: Interfaccia WSDL del Servizio IntegrationManager

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions          targetNamespace="http://services.pdd.openspcoop.org"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://services.pdd.openspcoop.org"
xmlns:intf="http://services.pdd.openspcoop.org"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wSDLsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!--WSDL created by Apache Axis version: 1.4
Built on Apr 22, 2006 (06:55:48 PDT)-->
  <wsdl:types>
    <schema                    elementFormDefault="qualified"
targetNamespace="http://services.pdd.openspcoop.org"
xmlns="http://www.w3.org/2001/XMLSchema">
      <element name="getMessage">
        <complexType>
          <sequence>
            <element name="in0" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
      <element name="getMessageResponse">
        <complexType>
          <sequence>
            <element name="getMessageReturn" type="impl:SPCoopMessage"/>
          </sequence>
        </complexType>
      </element>
      <complexType name="SPCoopMessage">
        <sequence>
          <element name="ID" nillable="true" type="xsd:string"/>
          <element name="azione" nillable="true" type="xsd:string"/>
          <element name="destinatario" nillable="true" type="xsd:string"/>
          <element name="message" nillable="true" type="xsd:base64Binary"/>
          <element name="servizio" nillable="true" type="xsd:string"/>
        </sequence>
      </complexType>
      <complexType name="SPCoopException">
        <sequence>
          <element name="codiceEccezione" nillable="true" type="xsd:string"/>
          <element name="descrizioneEccezione" nillable="true" type="xsd:string"/>
          <element name="identificativoFunzione" nillable="true" type="xsd:string"/>
          <element name="identificativoPorta" nillable="true" type="xsd:string"/>
          <element name="oraRegistrazione" nillable="true" type="xsd:string"/>
          <element name="tipoEccezione" nillable="true" type="xsd:string"/>
        </sequence>
      </complexType>
      <element name="fault" type="impl:SPCoopException"/>
      <element name="deleteMessage">
        <complexType>
          <sequence>
            <element name="in0" type="xsd:string"/>
          </sequence>
        </complexType>
      </element>
      <element name="deleteMessageResponse">

```

```
<complexType/>
</element>
<element name="getAllMessagesId">
  <complexType/>
</element>
<element name="getAllMessagesIdResponse">
  <complexType>
    <sequence>
      <element maxOccurs="unbounded" name="getAllMessagesIdReturn"
type="xsd:string"/>
    </sequence>
  </complexType>
</element>
<element name="getAllMessagesIdByService">
  <complexType>
    <sequence>
      <element name="in0" type="xsd:string"/>
      <element name="in1" type="xsd:string"/>
    </sequence>
  </complexType>
</element>
<element name="getAllMessagesIdByServiceResponse">
  <complexType>
    <sequence>
      <element maxOccurs="unbounded" name="getAllMessagesIdByServiceReturn"
type="xsd:string"/>
    </sequence>
  </complexType>
</element>
<element name="getAndDeleteMessage">
  <complexType>
    <sequence>
      <element name="in0" type="xsd:string"/>
    </sequence>
  </complexType>
</element>
<element name="getAndDeleteMessageResponse">
  <complexType>
    <sequence>
      <element name="getAndDeleteMessageReturn" type="impl:SPCoopMessage"/>
    </sequence>
  </complexType>
</element>
<element name="getMessageByReference">
  <complexType>
    <sequence>
      <element name="in0" type="xsd:string"/>
    </sequence>
  </complexType>
</element>
<element name="getMessageByReferenceResponse">
  <complexType>
    <sequence>
      <element name="getMessageByReferenceReturn" type="impl:SPCoopMessage"/>
    </sequence>
  </complexType>
</element>
<element name="getAndDeleteMessageByReference">
  <complexType>
```

```

    <sequence>
      <element name="in0" type="xsd:string"/>
    </sequence>
  </complexType>
</element>
<element name="getAndDeleteMessageByReferenceResponse">
  <complexType>
    <sequence>
      <element
        name="getAndDeleteMessageByReferenceReturn"
type="impl:SPCoopMessage"/>
    </sequence>
  </complexType>
</element>
<element name="getNextMessage">
  <complexType/>
</element>
<element name="getNextMessageResponse">
  <complexType>
    <sequence>
      <element name="getNextMessageReturn" type="impl:SPCoopMessage"/>
    </sequence>
  </complexType>
</element>
<element name="getNextAndDeleteMessage">
  <complexType/>
</element>
<element name="getNextAndDeleteMessageResponse">
  <complexType>
    <sequence>
      <element name="getNextAndDeleteMessageReturn" type="impl:SPCoopMessage"/>
    </sequence>
  </complexType>
</element>
<element name="deleteAllMessages">
  <complexType/>
</element>
<element name="deleteAllMessagesResponse">
  <complexType/>
</element>
<element name="invocaPortaDelegata">
  <complexType>
    <sequence>
      <element name="in0" type="xsd:string"/>
      <element name="in1" type="impl:SPCoopMessage"/>
    </sequence>
  </complexType>
</element>
<element name="invocaPortaDelegataResponse">
  <complexType>
    <sequence>
      <element name="invocaPortaDelegataReturn" type="impl:SPCoopMessage"/>
    </sequence>
  </complexType>
</element>
<element name="sendRispostaAsincrona">
  <complexType>
    <sequence>
      <element name="in0" type="xsd:base64Binary"/>
      <element name="in1" type="xsd:string"/>
    </sequence>
  </complexType>
</element>

```

```

    </sequence>
  </complexType>
</element>
<element name="sendRispostaAsincronaResponse">
  <complexType>
    <sequence>
      <element name="sendRispostaAsincronaReturn" type="xsd:string"/>
    </sequence>
  </complexType>
</element>
</schema>
</wsdl:types>
<wsdl:message name="getNextMessageResponse">
  <wsdl:part element="impl:getNextMessageResponse" name="parameters"/>
</wsdl:message>
<wsdl:message name="getMessageRequest">
  <wsdl:part element="impl:getMessage" name="parameters"/>
</wsdl:message>
<wsdl:message name="getAndDeleteMessageResponse">
  <wsdl:part element="impl:getAndDeleteMessageResponse" name="parameters"/>
</wsdl:message>
<wsdl:message name="sendRispostaAsincronaRequest">
  <wsdl:part element="impl:sendRispostaAsincrona" name="parameters"/>
</wsdl:message>
<wsdl:message name="invocaPortaDelegataRequest">
  <wsdl:part element="impl:invocaPortaDelegata" name="parameters"/>
</wsdl:message>
<wsdl:message name="deleteAllMessagesResponse">
  <wsdl:part element="impl:deleteAllMessagesResponse" name="parameters"/>
</wsdl:message>
<wsdl:message name="getAllMessagesIdByServiceRequest">
  <wsdl:part element="impl:getAllMessagesIdByService" name="parameters"/>
</wsdl:message>
<wsdl:message name="getAllMessagesIdByServiceResponse">
  <wsdl:part element="impl:getAllMessagesIdByServiceResponse"
name="parameters"/>
</wsdl:message>
<wsdl:message name="getMessageResponse">
  <wsdl:part element="impl:getMessageResponse" name="parameters"/>
</wsdl:message>
<wsdl:message name="sendRispostaAsincronaResponse">
  <wsdl:part element="impl:sendRispostaAsincronaResponse"
name="parameters"/>
</wsdl:message>
<wsdl:message name="invocaPortaDelegataResponse">
  <wsdl:part element="impl:invocaPortaDelegataResponse" name="parameters"/>
</wsdl:message>
<wsdl:message name="getNextMessageRequest">
  <wsdl:part element="impl:getNextMessage" name="parameters"/>
</wsdl:message>
<wsdl:message name="getNextAndDeleteMessageRequest">
  <wsdl:part element="impl:getNextAndDeleteMessage" name="parameters"/>
</wsdl:message>
<wsdl:message name="deleteMessageRequest">
  <wsdl:part element="impl:deleteMessage" name="parameters"/>
</wsdl:message>
<wsdl:message name="getAllMessagesIdResponse">
  <wsdl:part element="impl:getAllMessagesIdResponse" name="parameters"/>
</wsdl:message>

```

```

<wsdl:message name="getNextAndDeleteMessageResponse">
  <wsdl:part
name="parameters" />
    element="impl:getNextAndDeleteMessageResponse" />
</wsdl:message>
<wsdl:message name="getAndDeleteMessageRequest">
  <wsdl:part element="impl:getAndDeleteMessage" name="parameters" />
</wsdl:message>
<wsdl:message name="SPCoopException">
  <wsdl:part element="impl:fault" name="fault" />
</wsdl:message>
<wsdl:message name="getAndDeleteMessageByReferenceResponse">
  <wsdl:part
name="parameters" />
    element="impl:getAndDeleteMessageByReferenceResponse" />
</wsdl:message>
<wsdl:message name="deleteAllMessagesRequest">
  <wsdl:part element="impl:deleteAllMessages" name="parameters" />
</wsdl:message>
<wsdl:message name="deleteMessageResponse">
  <wsdl:part element="impl:deleteMessageResponse" name="parameters" />
</wsdl:message>
<wsdl:message name="getAndDeleteMessageByReferenceRequest">
  <wsdl:part
name="parameters" />
    element="impl:getAndDeleteMessageByReference" />
</wsdl:message>
<wsdl:message name="getMessageByReferenceResponse">
  <wsdl:part
name="parameters" />
    element="impl:getMessageByReferenceResponse" />
</wsdl:message>
<wsdl:message name="getAllMessagesIdRequest">
  <wsdl:part element="impl:getAllMessagesId" name="parameters" />
</wsdl:message>
<wsdl:message name="getMessageByReferenceRequest">
  <wsdl:part element="impl:getMessageByReference" name="parameters" />
</wsdl:message>
<wsdl:portType name="IntegrationManager">
  <wsdl:operation name="getMessage">
    <wsdl:input message="impl:getMessageRequest" name="getMessageRequest" />
    <wsdl:output
name="getMessageResponse" />
      message="impl:getMessageResponse" />
    <wsdl:fault message="impl:SPCoopException" name="SPCoopException" />
  </wsdl:operation>
  <wsdl:operation name="deleteMessage">
    <wsdl:input
name="deleteMessageRequest" />
      message="impl:deleteMessageRequest" />
    <wsdl:output
name="deleteMessageResponse" />
      message="impl:deleteMessageResponse" />
    <wsdl:fault message="impl:SPCoopException" name="SPCoopException" />
  </wsdl:operation>
  <wsdl:operation name="getAllMessagesId">
    <wsdl:input
name="getAllMessagesIdRequest" />
      message="impl:getAllMessagesIdRequest" />
    <wsdl:output
name="getAllMessagesIdResponse" />
      message="impl:getAllMessagesIdResponse" />
    <wsdl:fault message="impl:SPCoopException" name="SPCoopException" />
  </wsdl:operation>
  <wsdl:operation name="getAllMessagesIdByService">
    <wsdl:input
name="getAllMessagesIdByServiceRequest" />
      message="impl:getAllMessagesIdByServiceRequest" />
  </wsdl:operation>
</wsdl:portType>

```

```

    <wsdl:output          message="impl:getAllMessagesIdByServiceResponse"
name="getAllMessagesIdByServiceResponse" />
    <wsdl:fault message="impl:SPCoopException" name="SPCoopException" />
  </wsdl:operation>
  <wsdl:operation name="getAndDeleteMessage">
    <wsdl:input          message="impl:getAndDeleteMessageRequest"
name="getAndDeleteMessageRequest" />
    <wsdl:output          message="impl:getAndDeleteMessageResponse"
name="getAndDeleteMessageResponse" />
    <wsdl:fault message="impl:SPCoopException" name="SPCoopException" />
  </wsdl:operation>
  <wsdl:operation name="getMessageByReference">
    <wsdl:input          message="impl:getMessageByReferenceRequest"
name="getMessageByReferenceRequest" />
    <wsdl:output          message="impl:getMessageByReferenceResponse"
name="getMessageByReferenceResponse" />
    <wsdl:fault message="impl:SPCoopException" name="SPCoopException" />
  </wsdl:operation>
  <wsdl:operation name="getAndDeleteMessageByReference">
    <wsdl:input          message="impl:getAndDeleteMessageByReferenceRequest"
name="getAndDeleteMessageByReferenceRequest" />
    <wsdl:output          message="impl:getAndDeleteMessageByReferenceResponse"
name="getAndDeleteMessageByReferenceResponse" />
    <wsdl:fault message="impl:SPCoopException" name="SPCoopException" />
  </wsdl:operation>
  <wsdl:operation name="getNextMessage">
    <wsdl:input          message="impl:getNextMessageRequest"
name="getNextMessageRequest" />
    <wsdl:output          message="impl:getNextMessageResponse"
name="getNextMessageResponse" />
    <wsdl:fault message="impl:SPCoopException" name="SPCoopException" />
  </wsdl:operation>
  <wsdl:operation name="getNextAndDeleteMessage">
    <wsdl:input          message="impl:getNextAndDeleteMessageRequest"
name="getNextAndDeleteMessageRequest" />
    <wsdl:output          message="impl:getNextAndDeleteMessageResponse"
name="getNextAndDeleteMessageResponse" />
    <wsdl:fault message="impl:SPCoopException" name="SPCoopException" />
  </wsdl:operation>
  <wsdl:operation name="deleteAllMessages">
    <wsdl:input          message="impl:deleteAllMessagesRequest"
name="deleteAllMessagesRequest" />
    <wsdl:output          message="impl:deleteAllMessagesResponse"
name="deleteAllMessagesResponse" />
    <wsdl:fault message="impl:SPCoopException" name="SPCoopException" />
  </wsdl:operation>
  <wsdl:operation name="invocaPortaDelegata">
    <wsdl:input          message="impl:invocaPortaDelegataRequest"
name="invocaPortaDelegataRequest" />
    <wsdl:output          message="impl:invocaPortaDelegataResponse"
name="invocaPortaDelegataResponse" />
    <wsdl:fault message="impl:SPCoopException" name="SPCoopException" />
  </wsdl:operation>
  <wsdl:operation name="sendRispostaAsincrona">
    <wsdl:input          message="impl:sendRispostaAsincronaRequest"
name="sendRispostaAsincronaRequest" />
    <wsdl:output          message="impl:sendRispostaAsincronaResponse"
name="sendRispostaAsincronaResponse" />
    <wsdl:fault message="impl:SPCoopException" name="SPCoopException" />
  </wsdl:operation>

```



```
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="IntegrationManagerSoapBinding"
type="impl:IntegrationManager">
  <wsdlsoap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="getMessage">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="getMessageRequest">
      <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="getMessageResponse">
      <wsdlsoap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="SPCoopException">
      <wsdlsoap:fault name="SPCoopException" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="deleteMessage">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="deleteMessageRequest">
      <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="deleteMessageResponse">
      <wsdlsoap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="SPCoopException">
      <wsdlsoap:fault name="SPCoopException" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="getAllMessagesId">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="getAllMessagesIdRequest">
      <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="getAllMessagesIdResponse">
      <wsdlsoap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="SPCoopException">
      <wsdlsoap:fault name="SPCoopException" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="getAllMessagesIdByService">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="getAllMessagesIdByServiceRequest">
      <wsdlsoap:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="getAllMessagesIdByServiceResponse">
      <wsdlsoap:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="SPCoopException">
      <wsdlsoap:fault name="SPCoopException" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="getAndDeleteMessage">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="getAndDeleteMessageRequest">
      <wsdlsoap:body use="literal"/>
    </wsdl:input>
```

```
<wsdl:output name="getAndDeleteMessageResponse">
  <wsdlsoap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="SPCoopException">
  <wsdlsoap:fault name="SPCoopException" use="literal"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="getMessageByReference">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="getMessageByReferenceRequest">
    <wsdlsoap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="getMessageByReferenceResponse">
    <wsdlsoap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="SPCoopException">
    <wsdlsoap:fault name="SPCoopException" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="getAndDeleteMessageByReference">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="getAndDeleteMessageByReferenceRequest">
    <wsdlsoap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="getAndDeleteMessageByReferenceResponse">
    <wsdlsoap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="SPCoopException">
    <wsdlsoap:fault name="SPCoopException" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="getNextMessage">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="getNextMessageRequest">
    <wsdlsoap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="getNextMessageResponse">
    <wsdlsoap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="SPCoopException">
    <wsdlsoap:fault name="SPCoopException" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="getNextAndDeleteMessage">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="getNextAndDeleteMessageRequest">
    <wsdlsoap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="getNextAndDeleteMessageResponse">
    <wsdlsoap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="SPCoopException">
    <wsdlsoap:fault name="SPCoopException" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="deleteAllMessages">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="deleteAllMessagesRequest">
    <wsdlsoap:body use="literal"/>
  </wsdl:input>
  </wsdl:operation>
```

```
</wsdl:input>
<wsdl:output name="deleteAllMessagesResponse">
  <wsdlsoap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="SPCoopException">
  <wsdlsoap:fault name="SPCoopException" use="literal"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="invocaPortaDelegata">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="invocaPortaDelegataRequest">
    <wsdlsoap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="invocaPortaDelegataResponse">
    <wsdlsoap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="SPCoopException">
    <wsdlsoap:fault name="SPCoopException" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="sendRispostaAsincrona">
  <wsdlsoap:operation soapAction=""/>
  <wsdl:input name="sendRispostaAsincronaRequest">
    <wsdlsoap:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="sendRispostaAsincronaResponse">
    <wsdlsoap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="SPCoopException">
    <wsdlsoap:fault name="SPCoopException" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="IntegrationManagerService">
  <wsdl:port
    name="IntegrationManager"
    binding="impl:IntegrationManagerSoapBinding"
    <wsdlsoap:address
location="http://localhost:8080/openspcoop/IntegrationManager"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

## Allegato B: Schema risposta OK della PdD

```
<xsd:schema targetNamespace=http://www.spcoop.it/pdd/services/PD
  elementFormDefault="qualified" attributeFormDefault="unqualified">
<xsd:element name="RispostaOK">
  <xsd:complexType>
    <xsd:attribute name="ID" type="xsd:string" use="required"/>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
```