SYSTEM INTERFACING GUIDELINES

FOR

PORT COMMUNITY SYSTEM (PCS)

Release No 1.0 – March, 2007

CrimsonLogic Pte Ltd
31 Science Park Road, The Crimson, Singapore 117611, Main: (65) 6887 7888, Fax: (65) 6778 5277,
www.crimsonlogic.com.sg

PREPARED BY

ROLE	NAME	SIGNATURE	DATE (DD.MM.YYYY)
Project Leads	GOURAV ARORA, PANG TECK CHUN, ASHOK KUMAR, VENKATESAM, RAMESH KUMAR		09.03.2007
Project Manager	SURESH KUMAR KANTHOLY		

REVIEWED BY

ROLE	NAME	SIGNATURE	DATE (DD.MM.YYYY)
Project Manager	SURESH KUMAR KANTHOLY		16.03.2007

APPROVALS BY

DESIGNATION	NAME	SIGNATURE	DATE (DD.MM.YYYY)
PROJECT MANAGER	SURESH KUMAR KANTHOLY		16.03.2007
IPA PROJECT MANAGER	RAJEEV PURI		

DISTRIBUTION LIST *(Applicable only if hardcopies are distributed)*

DOCUMENT CONTROL NUMBER	VERSION NUMBER	RECIPIENT NAME	DATE (DD.MM.YYYY)
1	1.0		

RELEASE HISTORY

RELEASE NUMBER	ASSOCIATED CHANGE REQUEST NUMBER	DATE (MM,YYYY)	BRIEF SUMMARY OF CHANGES
Draft		March, 2007	First cut

CONTENTS

1. INTRODUCTION 5

1.1 PURPOSE AND SCOPE OF THIS DOCUMENT 5
1.2 REFERENCES 5
1.3 DEFINITIONS AND ABBREVIATIONS 6

2. PCS INTERFACES 7

2.1 OVERVIEW 7
2.2 USER TYPES AND INTERFACES 8
2.2.1 High Volume Users 8
2.2.2 General Users 8
2.2.3 Web Online Users 8

3. MESSAGING SYSTEM INTERFACES 9

3.1 MHUB MESSAGING SYSTEM 9
3.2 MHUB GATEWAYS/INTERFACES 10
3.2.1 FTP (Intelligence Services) 11
3.2.2 FTP Over SSL 11
3.2.3 RTFTP 11
3.2.4 ebXML 11
3.2.5 EDIINT AS2 12
3.2.6 SMTP 12
3.2.7 Web Services 12
3.3 CLIENT SYSTEM REQUIREMENTS 13
3.3.1 Client System/software 14
3.4 METHODS OF MESSAGE EXCHANGE 14
3.4.1 Application Interface 15
3.4.2 Scheduled message submission and retrieval 15
3.4.3 Manual message submission and retrieval 16
3.5 MESSAGING INTERFACE IMPLEMENTATION 17
3.6 BUSINESS SCENARIO ILLUSTRATED 18
3.7 COMMUNICATION NETWORK REQUIREMENTS 20

4. WEB SERVICES 21

4.1 OVERVIEW OF WEB SERVICES 21
4.2 WEB SERVICE TERMINOLOGIES/COMPONENTS 21
4.2.1 WSDL 21
4.2.2 Sample WSDL 23
4.2.3 SOAP Request from Client 24
4.2.4 SOAP Response 24
4.3 PCS WEB SERVICES 26
4.4 PCS AS SERVICE PROVIDER 26
4.5 PCS WEB SERVICE INTERFACE IMPLEMENTATION 27
4.6 PCS AS SERVICE REGISTRY 29
4.7 PCS AS SERVICE CONSUMER 29
4.8 WEB SERVICES SECURITY (WS-SECURITY: SECURESOAP) 30
4.8.1 HTTPS/SSL 30
4.8.2 Digital Signature 30
4.9 DEVELOPING WEB SERVICES 30

5. WEB USER INTERFACE 32

5.1 PCS WEB 32
5.2 SYSTEM REQUIREMENTS 32

6. BUSINESS DOCUMENTS 33

6.1 MESSAGE FORMAT 33
6.1.1 XML Schema 33

1. INTRODUCTION

1.1 PURPOSE AND SCOPE OF THIS DOCUMENT

Centralized Port Community System (PCS) is an initiative by Indian Ports Association (IPA) intended to provide a single window system for the Port communities in India to securely exchange the documents and information electronically with their stakeholders involved in the maritime transport and logistics chain including the trading partners and government agencies. It also expected to provide global visibility and access to the central database to all its stakeholders through internet based interfaces.

This document provides guidelines for PCS stakeholders on interfacing their system with PCS. The target audience for this document is the systems administrator/engineers of stakeholders, PCS project team members and the technical working group.

This document assumes that the target audience is familiar with concepts like messaging system, synchronous and asynchronous communication, enterprise application integration, XML and HTTP.

1.2 REFERENCES

The following documents have been referred during the preparation of this document:

- RFP No. IPA/ EDP/EDI/WBPCS/2005 for Centralized web-based Port Community System dated December 2005
- CrimsonLogic Proposal for IPA/ EDP/EDI/WBPCS/2005 dated March 6, 2006
- Work Order No IPA/EDP/EDI/WBPCS/2005/WO dated 12th Oct 2006
- Systems Requirements Specification for PCS release 2.2 in Feb 2007

1.3 DEFINITIONS AND ABBREVIATIONS

Abbreviation	Description
API	Application Programming Interface
BOM	Bill Of Material
CFS	Container Freight Station
CHA	Custom House Agent
CSV	Comma Separated Value file
CWG	Container Working Group
DC	Data Centre
DR	Disaster Recovery
EAI	Enterprise Application Integration
ebMS	EbXML Message Service
EC	Electronic Commerce
EDI	Electronic Data Interchange
EDIINT	EDIINT (EDI over INTernet)
XML	eXtensive Markup Language
SOAP	Simple Object Access Protocol
FTP	File Transfer Protocol
WSDL	Web Services Description Language

2. PCS INTERFACES

2.1 OVERVIEW

PCS provides a complete gamut of services to its stakeholders including centralized intelligent messaging and translation, real time track and trace, online registration and electronic payment services.

Since the IT infrastructure capabilities and the platforms in use vary to a large extent among different stakeholders, PCS offers a range of interfaces to access PCS, exchange business documents and avail any of the PCS services. The various interfaces provided or supported by PCS can be grouped as following:

1. Messaging System Interfaces
2. Web User Interface
3. Web Services

Depending on the IT adoption and resource availability in the organization, the stakeholders can use one or more interfaces provided to access PCS. For example, a Shipping Agent with reasonably good IT and communication systems can have the software interfaces to exchange the documents to PCS while a Customs Agent with just a PC and an Internet access can go for the Web Interface to access PCS.

A Port with a sophisticated IT and communication infrastructure and In-house systems can have Direct Application Interface with PCS availing the messaging gateways and Web Services provided by PCS to exchange documents with its clients and also to publish some of its services to the clients via Web Services.

2.2 USER TYPES AND INTERFACES

The choice of interface depends mainly on the transaction volume and the IT infrastructure the stakeholder has. Based on the IT Infrastructure and resource availability, the stakeholders can be grouped to three types broadly.

4. High volume users
5. General users
6. Web online users

2.2.1 HIGH VOLUME USERS

Major Ports with the transaction volumes quite high, the ports that exchange on an average between 8,000 and 15,000 documents (messages) with other stakeholders a day approximately, can be included in this group. These users are presumed to have sophisticated IT infrastructure and In-housel systems to manage their operations. They could use either the Direct Application Interfaces or System Interfaces for message exchange with PCS.

2.2.2 GENERAL USERS

Major or minor Ports and other stakeholders exchange documents up to 8,000 per day can be considered in this group. These users would have good IT infrastructure and internal systems to manage the operations. This group of users avail the System Interfaces for message exchange and web online interface for some of the interactions with PCS.

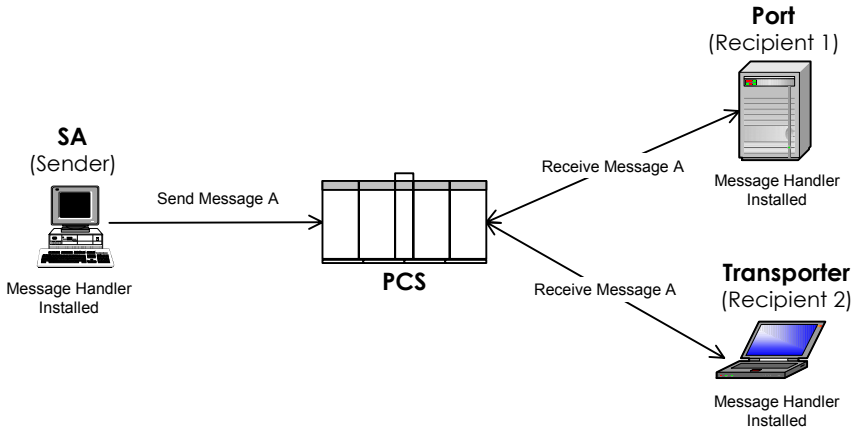
2.2.3 WEB ONLINE USERS

The Shipping and Customs House Agents, CFS and other stakeholders who do not have any or minimum IT infrastructure would fall in this group. Even minor Ports where vessel traffic and transactions are not high can also be considered in this category. They would use Web User Interface to access PCS.

3. MESSAGING SYSTEM INTERFACES

3.1 MHUB MESSAGING SYSTEM

MHub is an Internet-based messaging platform which allows business partners to carry out electronic document exchange in a highly secure manner. As part of PCS messaging solution, MHub handles electronic document exchange between stakeholders, the message routing and forwarding. It enables pre-defined trading partners to request for the submission and retrieval of document-based messages.

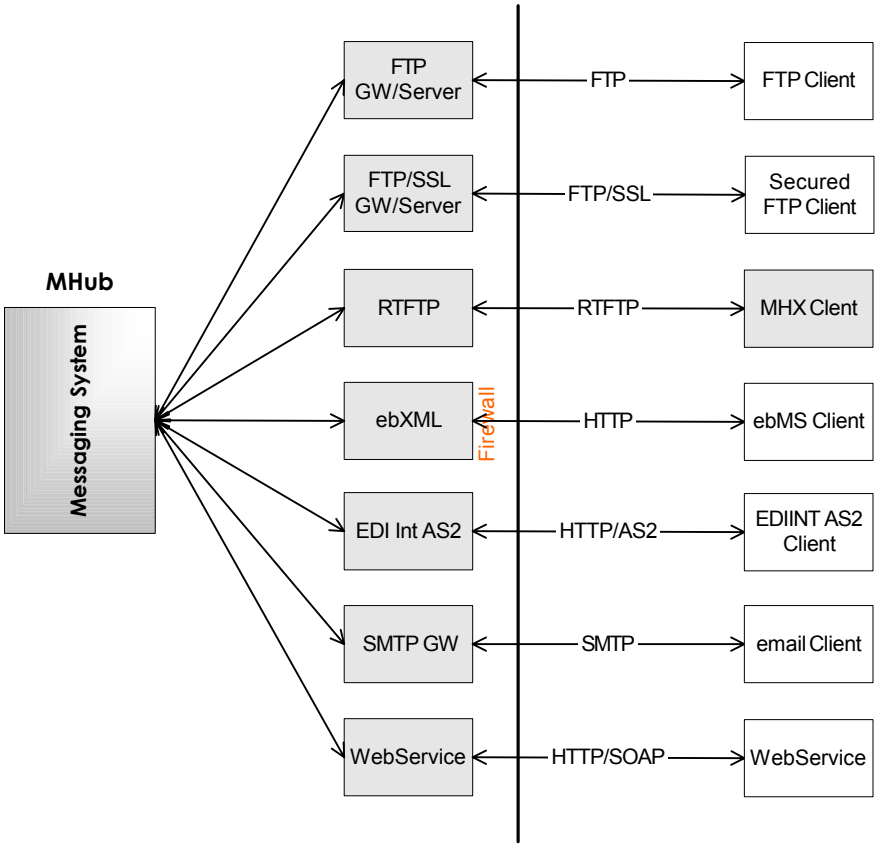


MHub provides a variety of interfaces/gateways for the stakeholders to exchange messages with their business partner. The stakeholder would choose the interfaces based on their in-house system interfacing requirements, IT infrastructure and the messaging protocols preferred and the transaction volume.

3.2 MHUB GATEWAYS/INTERFACES

MHub supports the standard messaging protocols like HTTP, SMTP and FTP and has inbuilt gateways to handle most of the international messaging standards. Some of the Interfaces/Gateways provided by MHub are:

- FTP
- FTP over SSL
- MHAccess (MHX)
- SMTP
- ebXML
- EDIINT AS2
- Web Service



The following section explains about the various messaging interfaces in detail.

3.2.1 FTP (INTELLIGENCE SERVICES)

Stakeholders can deposit their messages for their trading partner onto a designated folder in PCS FTP server using their FTP client. Whenever a message is deposited in the FTP server, FTP Gateway fetches this message from FTP server and deposits the message into the respective Message Hub mailbox.

In response, when the message is received into the mailbox, FTP Gateway will retrieve the message and move it to the FTP server's user directory allocated for the recipient.

FTP Gateway handles EDIFACT, X12, XML, Binary and Text file formats. FTP Gateway retrieves and submits messages from MHub at the user's preferred time and at varied preferred intervals based on pre-defined criteria, such as, document type.

3.2.2 FTP OVER SSL

FTP/SSL gateway works in the same way as FTP gateway but uses SSL (Secured Socket Layer) to transmit the messages. The users can use any secured FTP clients to send messages to or retrieve from PCS MHub.

3.2.3 RTFTP

RTFTP receives messages sent from the MHX Client installed on the stakeholders' system and sends the messages in response to the request from the MHX Client.

MHX client is a communication program (messaging client) from CrimsonLogic that allows messages to be transferred between a user's system and Message Hub. MHX client uses its proprietary RTFTP for messaging. MHX can be invoked using a command-line script which can be configured for execution by a scheduler (program) or a server.

3.2.4 EBXML

The ebXML messaging provides the message packaging, routing and transport facilities for the ebXML infrastructure. It defines the message enveloping and header document

schema used to transfer ebXML messages over a communications protocol such as HTTP or SMTP and the behavior of software sending and receiving ebXML messages.

The ebXML messaging service is defined as a set of layered extensions to the base Simple Object Access Protocol [SOAP] specification with security and reliability features necessary to support international electronic business.

MHub includes ebXML gateway that can receive and forward ebXML messages from stakeholders.

3.2.5 EDIINT AS2

EDIINT (EDI over INTernet) protocol is a MIME based data exchange protocol either communicating through AS2-HTTP. EDIINT is specially designed to allow structured data transmission over the Internet and focuses on MIME, security (encryption and signatures) and business process exchange requirements like acknowledgement (MDN), Non-repudiation of receipt (NRR).

MHub supports EDIINT and has own gateway for EDIINT AS2-HTTP. The stakeholders need to have their own application that can send and receive messages using AS2 –HTTP to PCS. The stakeholders systems should be able to support the handshaking protocol via SSL, required for proper AS2 message communications between PCS and their systems.

3.2.6 SMTP

One of the simplest means to interface with PCS Message hub is to send messages through SMTP gateway. Stakeholders can send messages as email attachments to MHub and MHub SMTP gateway would separate the message from the email and deposit to recipients MHub mail box. SMTP gateway uses SMTP protocol for message exchange. It will zip the header, footer, body and the attachment and deposit in the recipient's mailbox.

3.2.7 WEB SERVICES

Web Service uses standardized XML messaging system and platform independent. Web Service use SOAP or XML-RPC protocols over HTTP.

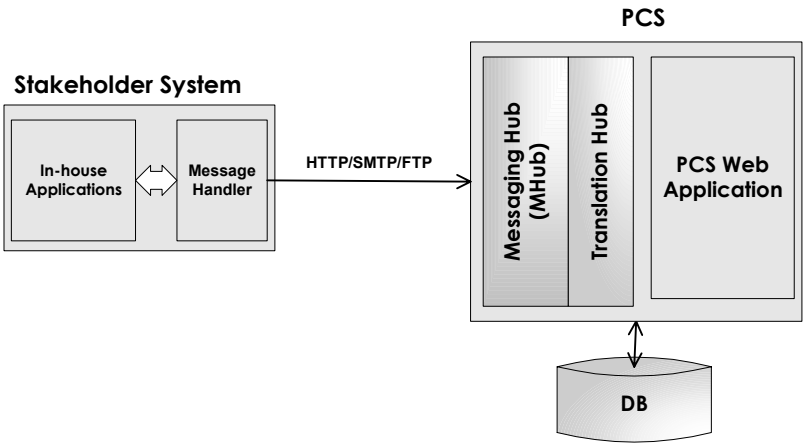
M-Hub provides Web Services for the following functionalities:

- Retrieve Message Web Service
- Submit Message Web Service
- List Messages Web Service
- Delete Message Web Service

M-Hub web service interface can handle SOAP request and response for clients systems connecting to M-Hub. The SOAP request formulated by client systems should comply with WS Basic Profile 1.0

3.3 CLIENT SYSTEM REQUIREMENTS

Client (Stakeholder) needs to have a Message Handler in their system to send or receive messages from/to PCS as shown in the diagram below. This Message Handler can be a software program developed in-house, an application module part of in-house system or a third party messaging client for the gateway they choose for interfacing with MHub.



For all the MHub Interfaces/Gateways (except MHAccess), the registered stakeholders can use any client of their choice that can support the gateway and the protocol they choose to interface with PCS and send and receive the electronic documents from PCS

messaging system. This gives the flexibility to the stakeholder to interface with Mhub according to their IT resource and capability.

MHAccess is CrimsonLogic proprietary interface and required to use MHAccess client from CrimsonLogic to exchange messages.

3.3.1 CLIENT SYSTEM/SOFTWARE

SNo	Interface	Messaging Client	System
1	FTP	Any FTP client	refer the respective software documentation
2	FTP/SSL	Any Secured FTP Client	refer the respective software documentation
3	MHAccess	MHAccess client (CrimsonLogic)	o Pentium III 600Mhz, 256 MB RAM, 50Mb of free hard disk space o MS Windows 98 / Me / XP / NT / 2000, Internet Explorer (IE) 5.5 or above
4	EbXML	MSH for ebMS	refer the respective software documentation
5	EDIINT AS2	AS2 –HTTP client	refer the respective software documentation
6	SMTP	Email Client	refer the respective software documentation
7	Web Service GW	Web Service	refer the respective software documentation

3.4 METHODS OF MESSAGE EXCHANGE

Stakeholders can interface with PCS Mhub according to their IT resource and capability as mentioned earlier. The Interface can be:

- An application Interface
- A scheduled message submission and retrieval
- Manual message submission and retrieval

3.4.1 APPLICATION INTERFACE

Users can have seamless integration with the PCS and exchange messages from their in-house applications if the in-house applications support messaging through the standard protocols or gateways mentioned in section 3.2.

If the in-house systems do not support messaging and user still would like to have an application interface with PCS from their in-house applications, they need to develop a message handler, a software program on their IT platform, in order to interface with PCS from their application system. The message handler should wrap the EDI or XML data files from their in-house systems to a message for the desired gateway/protocol and submit to the gateway and also, to receive the incoming messages from the gateway and pass it to the application for its further processing.

User also can achieve the real time submission or retrieval without human intervention by installing a third party messaging client that supports scripting (command line execution). The in-house systems can make system calls to submit and receive messages from their application directly.

3.4.2 SCHEDULED MESSAGE SUBMISSION AND RETRIEVAL

Another way to exchange messages with PCS without manual intervention is to use a scheduler to submit or receive messages in regular intervals via a third part messaging client installed at users system.

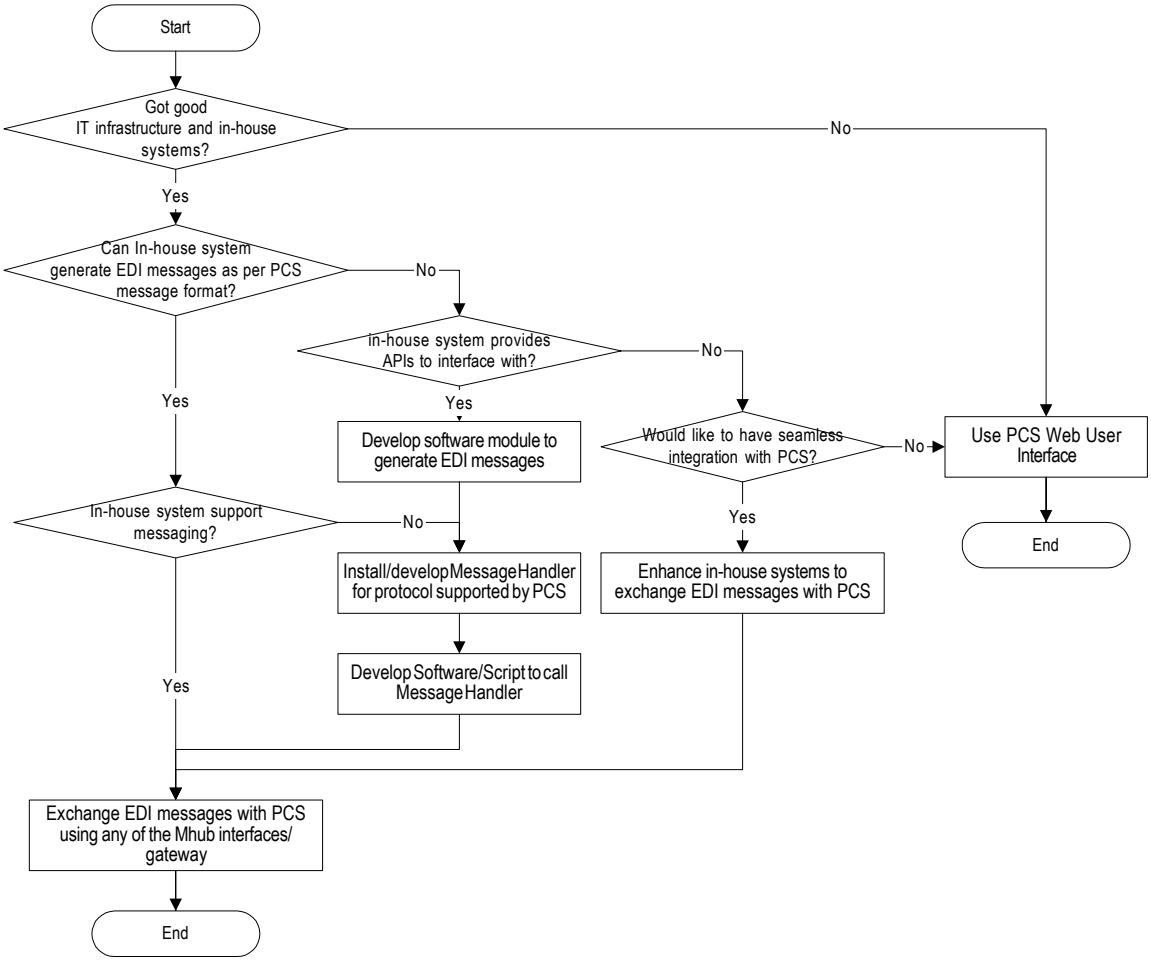
The in-house applications can put the EDI/XML data files to be sent to a designated folder in the system and the Scheduler can call a script that picks up the messages from the designated folder and invokes the messaging client to submit these messages to the gateway. The script can invoke the messaging client again to receive the messages to the user from PCS from the gateway into another designated folder, after sending out the messages are done or can have another scheduler to receive the incoming messages.

3.4.3 MANUAL MESSAGE SUBMISSION AND RETRIEVAL

Stakeholders also can use the third party messaging clients' user interfaces to do a batch submission/retrieval of the messages. This mode is manual and the user need to start the software and submit or receive the messages by clicking the send/receive buttons or executing send/receive commands.

3.5 MESSAGING INTERFACE IMPLEMENTATION

Interfacing with PCS Messaging Hub is straight forward if the stakeholders' in-house systems are open and support the messaging. But it's not always and can't expect all stakeholders got the ideal system. The following flow chart helps the stakeholders to identify the steps in preparation to interface with PCS Message Hub:



3.6 BUSINESS SCENARIO ILLUSTRATED

PCS messaging is illustrated with the help of examples with some of the business scenarios below:

Example 1: Shipping Agent (SA) submitting IGM* to Customs and Port

Note: * In the current system Shipping Agents submit the IGM to Customs and Customs forward IGM to Port. We took an ideal case where more than 2 stakeholders are involved, just for an example.

Scenario – 1: Shipping Agent SA1 with in-house system that generates IGM in Customs, Port or PCS format messages.

1. In-house system of SA1 generates IGM either in Customs, Port or PCS format and drops it in a designated folder required by the message handler.

2. Message Handler installed in SA1 system picks up the IGM from the folder and sends it to PCS

3. PCS receives the IGM, translates the IGM to PCS format if it not in PCS format, extract relevant information from IGM and insert the data to PCS Database.

4. If SA1 IGM format is not in Customs format, PCS translates the IGM to the Customs format and sends it to the Customs server.

5. If SA1 IGM format is not in Port format, PCS also translates the IGM to the format Port requires and sends translated IGM to port server.

Scenario – 2: Shipping Agent SA2 without any in-house system.

1. SA1 access PCS Web and enters the IGM details or upload IGM message in Customs, Port or PCS format to PCS via PCS Web.

2. PCS translates the IGM to the Customs format and sends it to the Customs server.

3. PCS also translates the IGM to the format Port requires and sends translated IGM to Port server.

Example 2: Port sending Tally Report to Shipping Agent and Customs.

1. Port Terminal staff enters the container discharged from the vessel onto Port Operating System (In-house system) of Port A

2. Port Operating System generates Tally Report (UN/EDIFACT or any structured format) and copies the file in a designated folder.

3. Message Handler installed in Port A system picks up the Tally Report file from the folder and sends it to PCS

4. PCS receives the Tally Report, extract relevant information required and insert the extracted data to PCS Database.

5. PCS sends the Tally Report in default PCS format to the Shipping Agent.

6. If Port A Tally Report format is different from that of customs, PCS translates** the Tally Report to the format Customs required using the map and sends translated tally to Customs server.

Note: ** Translation requires a map to translate from one format to another format and the map depending on the in-house format need to be developed by the Port A using the Transwork software provided.

3.7 COMMUNICATION NETWORK REQUIREMENTS

Since PCS is an Internet based solution, the only communication link required to access PCS is internet access. Depending on the transaction volume of the stakeholder, the link can be a broadband, shared or dedicated leased line with the local ISP (Internet Service Provider).

For the high volume users, stakeholders like Port, where the service is critical and the availability and accessibility of PCS is crucial, it is advised to have a back up line with another ISP in order to assure business continuity.

The general guideline for choosing the bandwidth for the internet access for PCS is given below. Please note that the suggestions below are based on rough estimation of the number of messages and other assumptions like the bandwidth available to the stakeholders is the same that the stakeholder originally subscribed with the service provider and the bandwidth is used for PCS mainly.

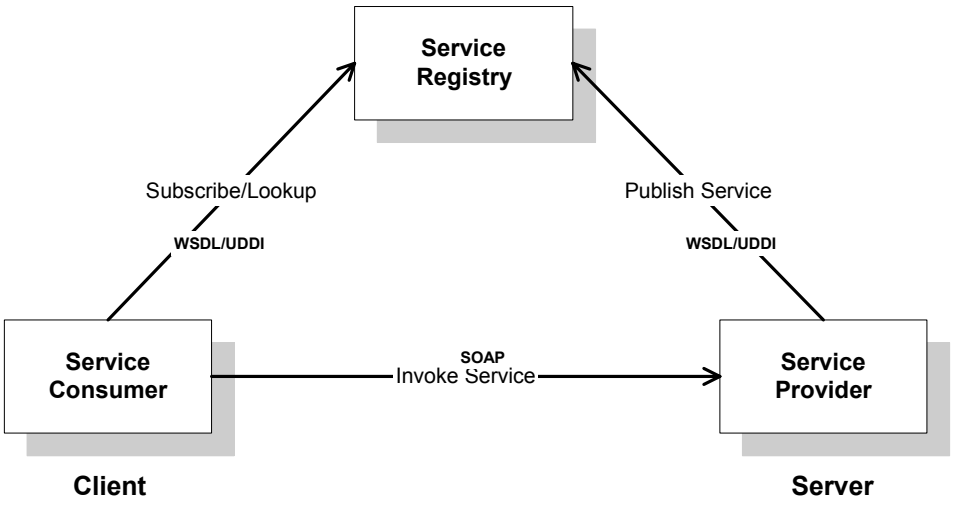
User Type	Description	Internet Access	Back up
High Volume users	Major ports or other stake holder exchanges approximately 8000 to 15000 messages a day	512 kbps leased line (dedicated)	256 kbps
Normal users	Major or minor ports or any other stakeholders exchanges less than 8000 messages a day	256 kbps	128 kbps
Web online users	Stakeholders whom the transaction volume is quite low (in hundreds a day) or accessing PCS only through web online	Broadband (256 kbps)	

Note: Performance of the local ISP should be evaluated before choosing the service provider as the PCS service availability and speed depends heavily on the Internet Service provided by the ISP.

4. WEB SERVICES

4.1 OVERVIEW OF WEB SERVICES

Web Services are the new breed of web applications that can be described, published, located and invoked over the Internet. Web services facilitate direct interfacing of systems/applications hosted in different platforms in different geographical locations. It makes use of XML messaging. The Web Services architecture has three distinct roles; service provider, service Consumer and Service Registry.



4.2 WEB SERVICE TERMINOLOGIES/COMPONENTS

The terminologies or components used in Web Services are briefly explained below taking one of the PCS service, finding the container location in the port yard, as an example.

4.2.1 WSDL

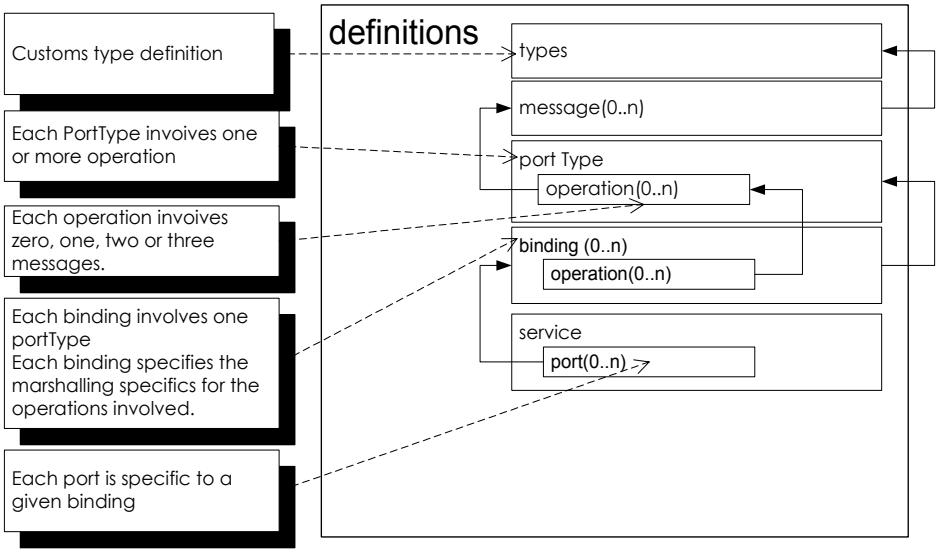
WSDL (Web Services Description Language) is used by the Service Provider to describe the service that is being provided. WSDL includes information on all publicly available

functions in the service, data type for the XML messages, address for locating the specified service and the binding information for the transport protocol to be used.

These descriptions are published in the Service Registry (UDDI) where they can be searched. A Service Requestor uses the Find operation to retrieve the WSDL definition of a service that it wants to use. Once the service requestor has obtained the WSDL definition, it has enough information to invoke the service.

The terms used in WSDL are summarized below:

1. **Services** are accessible through one or more **Ports**
2. **Ports** are bound to a **portType**.
3. Multiple **binding** are possible per **portType**
4. **portType** gives abstract definition of a service



One WSDL document contains one or more services. A service contains zero or more port definitions (service endpoints), and each port definition contains a specific protocol extension.

4.2.2 SAMPLE WSDL

The following is a sample WSDL file for a simple Container Tracking Service. Please note that it is just a sample file given as example for the readers to understand the concept and it does not represent the actual WSDL for the service provided by PCS.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="ContainerTrackingService"
targetNamespace="http://www.pcs.com/wsdl/ContainerTrackingService.wsdl"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://www.pcs.com/wsdl/ContainerTrackingService.wsdl"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<message name="findContainerRequest">
<part name="containerID" type="xsd:string"/>
</message>
<message name="findContainerResponse">
<part name="location" type="xsd:string"/>
</message>

<portType name="ContainerPortType">
<operation name="findContainer">
<input message="tns:findContainerRequest"/>
<output message="tns:findContainerResponse"/>
</operation>
</portType>

<binding name="ContainerBinding" type="tns:ContainerPortType">
<soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="findContainer">
<soap:operation soapAction=""/>

<input>
<soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:pcs:containerTrackingService"
use="encoded"/>
</input>
<output>
<soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:pcs:containerTrackingService"
use="encoded"/>
</output>

</operation>
</binding>

<service name="ContainerTracking_Service">
<documentation>WSDL File for Container Tracking Service</documentation>
<port binding="tns:ContainerBinding" name="ContainerPort">
<soap:address location="http://localhost:8080/soap/servlet/rpcRouter"/>
</port>

</service>
</definitions>
```

4.2.3 SOAP REQUEST FROM CLIENT

Client systems can locate a web service and invoke any of the available function using the WSDL of the service published. The client system prepares a SOAP request based on the information gathered from the WSDL and sends the SOAP request to the Web Service.

Following is a sample soap request from client to the web service as per the WSDL published given above.

```
POST /servlet/rpcrouter HTTP/1.0
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: 494

<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<SOAP-ENV:Body>
<ns1:findContainer
xmlns:ns1="urn:pcs:containerTrackingService"
SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-encoding/">
<containerID xsi:type="xsd:string">con1234567</zipcode>
</ns1:findContainer>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

4.2.4 SOAP RESPONSE

The Web Service receives the SOAP request, generates the SOAP response with the information requested and returns the response SOAP to the client.

Here is a sample SOAP response from the web service to the client that sent the SAOP request as mentioned in 4.2.3

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://www.w3.org/2001/09/soap-envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
<ns1:findContainerResponse
xmlns:ns1="urn:pcs:containerTrackingService"
SOAP-ENV:encodingStyle="http://www.w3.org/2001/09/soap-encoding/">
<return location xsi:type="xsd:int">DF008</return>
</ns1:findContainerResponse>
```

```
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

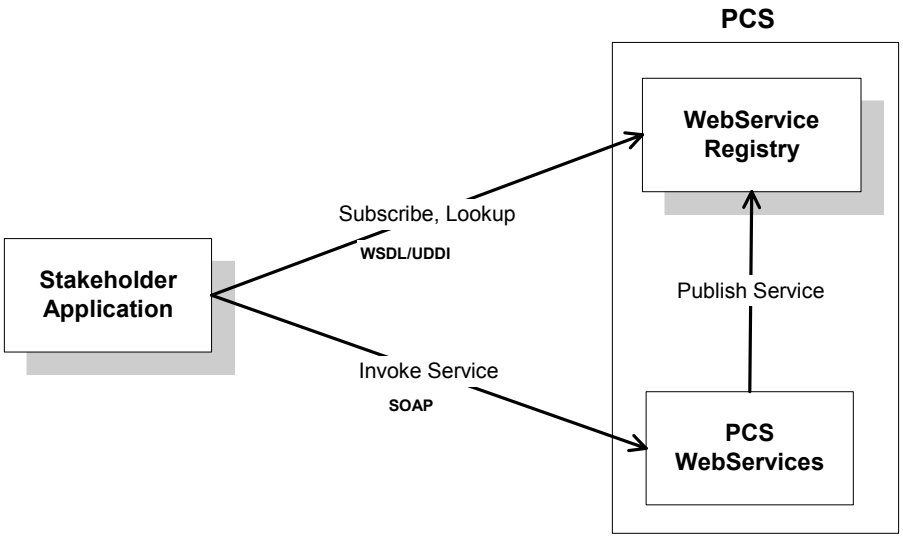
4.3 PCS WEB SERVICES

PCS provides web service interfaces and platform for ports and other stakeholders in order to help stakeholders' in-house applications to integrate directly with PCS and takes all the above roles in various scenarios depends on the service provided. So the services provided by PCS can be grouped according to the roles:

- PCS as service Provider
- PCS as service Registry
- PCS as service Consumer

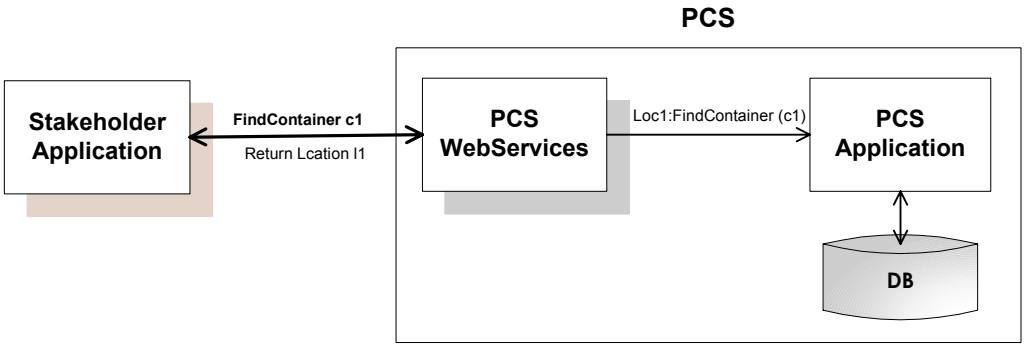
4.4 PCS AS SERVICE PROVIDER

As service provider, PCS exposes many of its services as web service interfaces for the stakeholder in-house applications to consume. PCS Web Services are published on PCS Web Registry hosted by PCS for stakeholders. The stakeholder application has to find/subscribe for the services and then use the service information and binding templates to invoke the service.



Web Service can be synchronous or asynchronous service depending on the requirements. Some of the web services provided by PCS like finding container location

in the port/CFS yard are synchronous. In synchronous services, the client invokes a service and wait for the response to the request. Synchronous services are best suited when the service can process the request in a small amount of time or the client applications require an immediate response to a request.



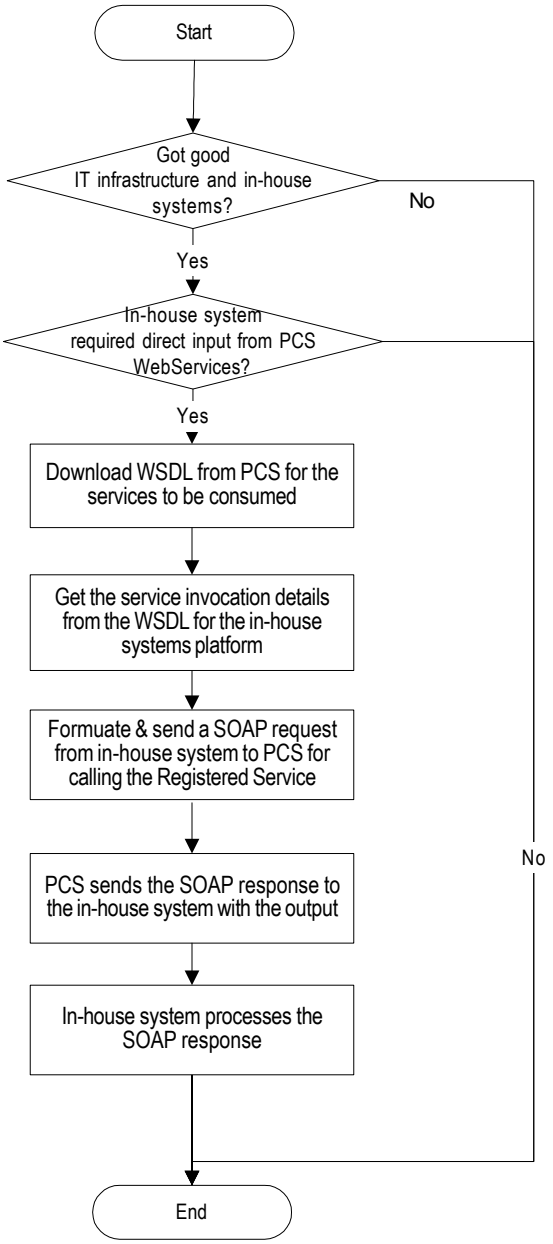
Web services that rely on synchronous communication normally use RPC-oriented approach for synchronous services.

4.5 PCS WEB SERVICE INTERFACE IMPLEMENTATION

As service consumer, Stakeholders who wish to consume PCS Web Services are required to download the PCS Web Services WSDL via PCS Web Services.

All the stakeholders have to perform their integration with PCS Web Services through the services detailed in the WSDL. This process of integration is external system-specific and level of automation is left flexibility to the owner of these external systems.

The activities involved in the preparation to interface with PCS Web Services are shown as a flow chart below:



The stakeholders then formulate the SOAP request (depending on their technology used for e.g. J2EE or .Net) to PCS for calling the registered Web Services.

PCS Web Services will then formulate the SOAP response back to the external application, which should take care of processing the SOAP response.

4.6 PCS AS SERVICE REGISTRY

PCS allows the Ports to publish their web services in PCS so that their users can find the web services a port provides and then interface with the service from their systems or portal if they wish to.

PCS as service registry makes use of database storage to maintain service registry. This registry contains the information related to the services registered on the PCS Web services including the pointer to an external specification and an address for invoking the web service. PCS Service Registry allows the stakeholders to search web services registered and to find the details of the services.

PCS service registry can be used to:

- Register web services
- Search for registered services and to view service details
- Activate, de-activate, de-register and manage services

4.7 PCS AS SERVICE CONSUMER

PCS access some of the web services from the Ports like resource availability in order to complete the business transaction or to show it through the PCS Portal. This can be a published service in the PCS web service registry or some of the services ports developed for interfacing with PCS.

External Systems that wish to publish their web services for PCS modules to interface with, can simply upload their WSDL to the PCS Web Service Gateway (WSG). WSG will perform all the SOAP request validation and extraction and SOAP response generation, bridging the communication between internal PCS modules and external web-services aware systems.

4.8 WEB SERVICES SECURITY (WS-SECURITY: SECURESOAP)

For Web Services interfacing, a set of security-related, non-functional requirements have been identified. The PCS Web Services Gateway follows the OASIS standard WS-security. WS-Security describes enhancements to SOAP messaging to provide quality of protection through message integrity, message confidentiality, and single message authentication.

4.8.1 HTTPS/SSL

The communication between the Web services consumer, the web services gateway and the web services producer should not be viewed by a third party as it travels on the Internet. This is ensured by the use of HTTPS/SSL transport security.

4.8.2 DIGITAL SIGNATURE

The Web services gateway should be able to determine from whom the message was coming and be able to verify that the sender was who the sender claimed to be. Similarly, the Web services publisher must be able to ensure that the party invoking the web service is well identified. This requirement is addressed through the use of digital signatures and digital certificates.

When using a digital certificate approach, the Web service requester must have a digital certificate which has been signed by a trusted certificate authority (CA). The requester will use this certificate to assert their identity and will digitally sign the SOAP message so that the requester's identity and the message's integrity can be verified. This also ensures the provider that the data being transmitted was not tampered with.

4.9 DEVELOPING WEB SERVICES

The following steps can be adopted for stakeholders who want to have their services published as Web Services via PCS.

1. Select internal functionalities of internal systems that are to be exposed as web Services.

2. Depending on the underlying technology of the internal system, develop the Web Services methods / operations that will invoke these internal functionalities.

3. Construct the WDSL file that defines the list of Web Services exposed.
   (The WSDL file is a XML file that contains the details of each Web Service, including the operation names, the data types to be passed in etc. The WSDL file produced should comply with WS-I Basic Profile 1.0 so that only standardized data types defined in the profile is used.)

4. Upload the WSDL file via the PCS Web Services Gateway (WSG) portal, which hosts the PCS Web Service Registry for all Web Services.

5. Upon a successful upload to PCS, the publisher has made available to the rest of the stakeholders the Web Services operations that it can accept for invocations.

5. WEB USER INTERFACE

5.1 PCS WEB

PCS provides web user interface for the stakeholders who don't have enough IT infrastructures to directly interface to PCS from their systems. PCS web is used by the category of users mentioned Web Online uses in the beginning of this document.

PCS web has web forms for its online users to fill and submit and options to receive and view any of the messages used in PCS to/from its trade partner.

PCS web also provides web upload and download for the Web online users who do not have in-house systems but have some means to prepare the structured files for the PCS messages.

PCS web doesn't require any infrastructure other than a PC with browser and an internet connection, making it possible for any stakeholder to access PCS. PCS web can be accessed even from any cyber café anywhere anytime.

5.2 SYSTEM REQUIREMENTS

PCS Web doesn't require any IT infrastructure other than a PC with browser and an internet connection as mentioned above.

- PC: Pentium III 600 Mhz or above with minimum 256 MB RAM
- OS: Microsoft Windows 98/ Me/ XP/ NT /2000 with Internet Explorer 5.5 or above
- Internet: Internet connection with Bandwidth 256 KB or above

6. BUSINESS DOCUMENTS

6.1 MESSAGE FORMAT

PCS follows the message formats published by IPA as base format and the message structure from customs for customs related documents as customs standard.

The complete PCS message formats including the message formats from IPA and Customs with the changes if any would be compiled and published by PCS shortly as PCS message format release 1.0.

6.1.1 XML SCHEMA

In order to standardize the messaging, PCS uses XML as the default message type and translates the EDI documents/messages in other types like UN/EDIFACT, X12 or any structured format used by Ports or Customs.