



Business Gateway Developer pack

Architecture overview

Version 1.0

Information Systems
Seaton Court
2 William Prance Road
Plymouth
Devon
PL6 5WS



Contents

1. Introduction.....	3
1.1. Purpose.....	3
1.2. Business Gateway overview	3
2. Architecture overview	4
2.1. Web service implementation.....	4
2.1.1. Process flow	5
2.2. Example request/response flow	5
2.3. Poll requests	8
2.4. Security	9
2.5. Interoperability	9
2.6. Reliable messaging.....	9
2.6.1. Use of unique message IDs	10
2.6.2. Message is lost in transit from the client.....	10
2.6.3. Message is lost in transit from the gateway	10
2.6.4. Resubmission of a failed request.....	11



1. Introduction

1.1. Purpose

The purpose of this document is to provide an overview of the externally facing Business Gateway architecture to enable a third party to understand the key principles involved in integrating with Business Gateway services offered by Land Registry.

1.2. Business Gateway overview

Business Gateway software exposes Land Registry business services as web services using SOAP, an industry-proven standard for communication between systems using XML.

The key features of Business Gateway are:

- adoption of open standards – e-GIF, HTTP, SOAP, WSDL, XML
- XML-based web services
- XML schemas that comply with the standards set by W3C, e-GIF and OSGRE
- provision of secure access to specific Land Registry services over the internet
- mutual SSL authentication between Business Gateway and client using certificate based authentication
- requests will be queued by Business Gateway when Land Registry systems are unable to process requests due to working hour limitations
- synchronous responses to requests when Land Registry systems are available
- reliable messaging facilitated by the use of a unique message ID
- multi-language enabled – facilitated via locale parameter passed in the SOAP header.

Figure 1 shows the basic principles for integration with the Business Gateway solution.

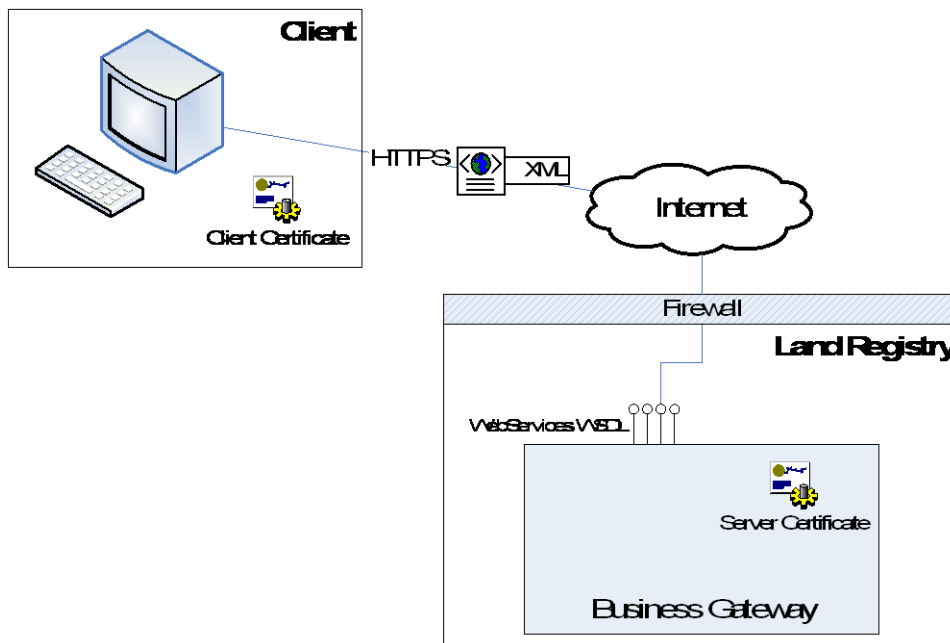


Figure 1: Solution overview

2. Architecture overview

2.1. Web service implementation

Business Gateway is presented as a collection of web services. An example integration between a service consumer and a service provider is shown figure 2. The example uses a generated proxy class to allow the client application to consume a web service as if it were a call to a native method. The proxy is generated from the web services described by the WSDL endpoint.

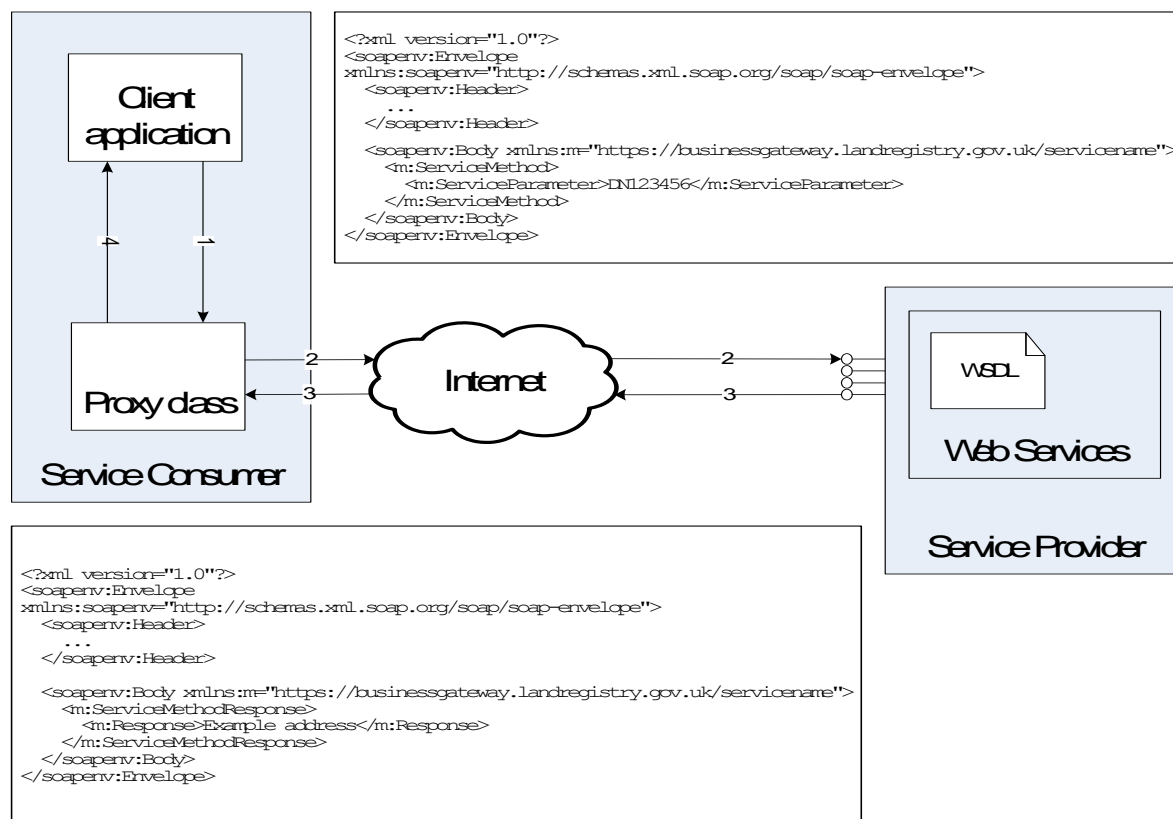


Figure 2: Web services overview

2.1.1. Process flow

1. The client application calls the proxy method for the desired web service.
2. The proxy class serialises the parameters into a SOAP message and sends it over HTTP to the SOAP endpoint as defined in the WSDL..
3. The web services engine de-serialises the received message into an object that is used by the service provider application in a method call. The result of the method call is then serialised into a SOAP message by the web services engine and returned to the service consumer.
4. The proxy class de-serialises the response message and returns the web service call result to the client application in a format that is suitable for native consumption.

2.2. Example request/response flow

Business Gateway is implemented as a set of web services. It is designed to be highly available and is exposed as a set of pull-for-data interactions. The gateway does not push data to client systems, as this would require client systems to be highly available.

The gateway operates using the following model:

- request received in Land Registry business hours, processed automatically – result and response returned synchronously
- request received in Land Registry business hours, requires manual processing – acknowledgement response returned



- request received out of Land Registry business hours – acknowledgement response returned with a suggested time for poll back
- polling request received – response returned if available. If response is not yet available then initial acknowledgment returned.

Using this model, each service offered by the gateway will have a corresponding service to allow for a poll request. A response will consist of an acknowledgment, a result, or a rejection.

For each request the following is required:

- client certificate - provided by Land Registry to create an SSL connection
- user ID and password - to authenticate the user
- locale for the response language (currently only 'en' supported)
- unique message ID - generated by the client to allow for reliable messaging.

Figure 3 shows the basic flow for Business Gateway behaviour upon receiving a service request.

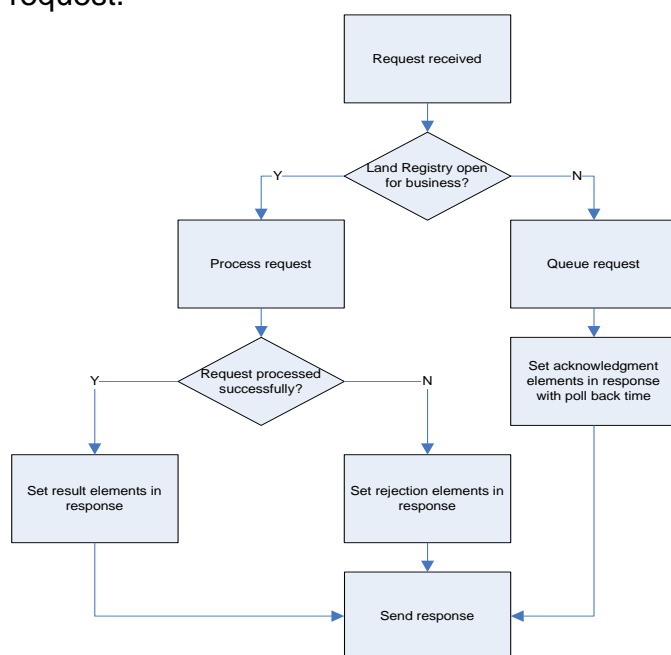


Figure 3: Business Gateway request processing flow

Figure 4 shows an example service request. The scenario covers the case where a secure SSL connection has already been created and a service request is being sent for the first time.

The scenario covers the following generic cases.

- SOAP fault due to XML request message failing validation.
- Rejection response due to authorisation failure.
- Result response as request processed successfully.
- Acknowledgement response as request unable to be processed currently.
- Rejection response as request failed processing.

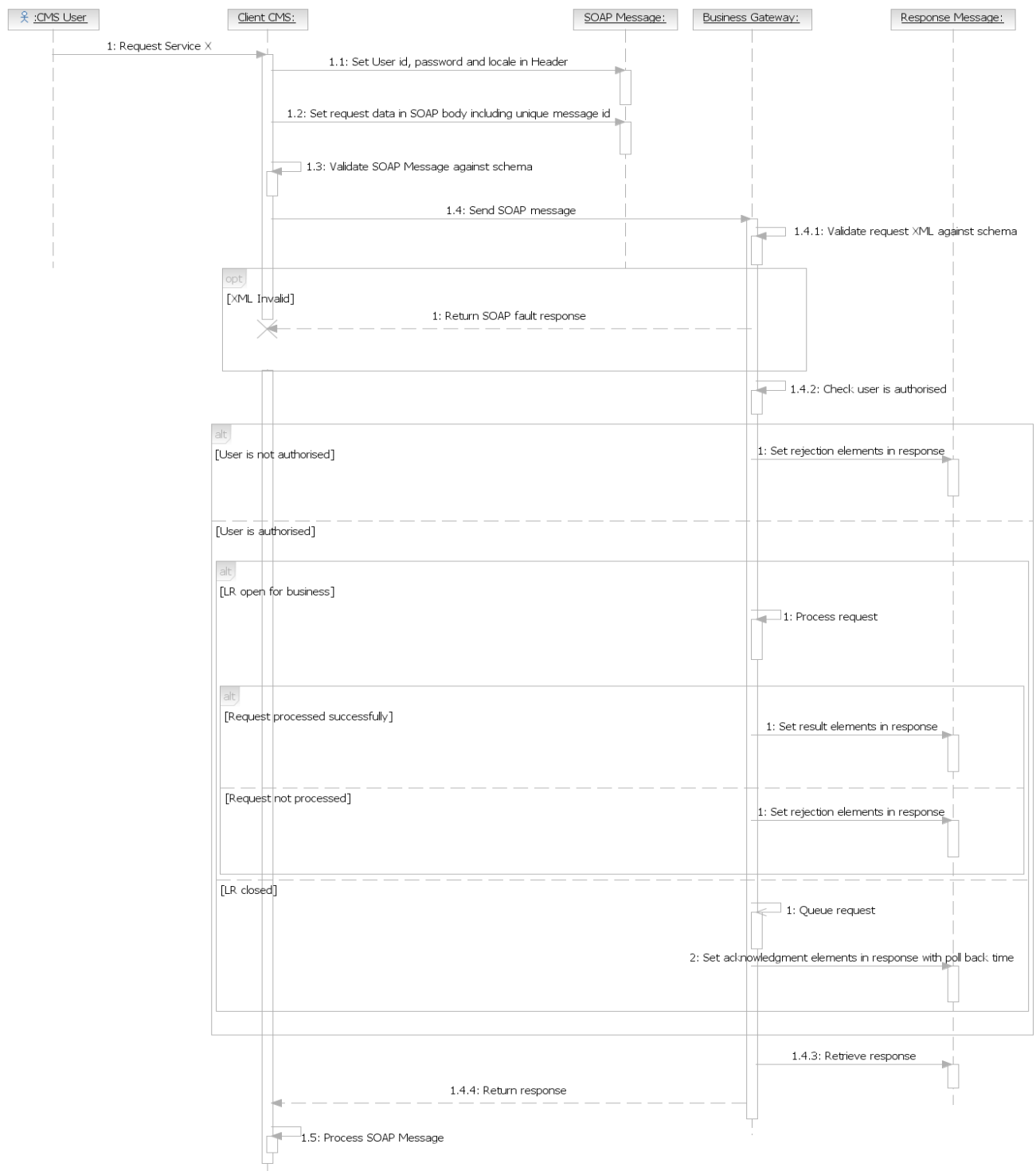


Figure 4: Example service request flow



2.3. Poll requests

When a request has received an acknowledgment response, the client is required to poll the gateway for a response. This can be automated by the client using the Outstanding Request Service. Figure 5 shows the basic flow for Business Gateway behaviour upon receiving a poll request.

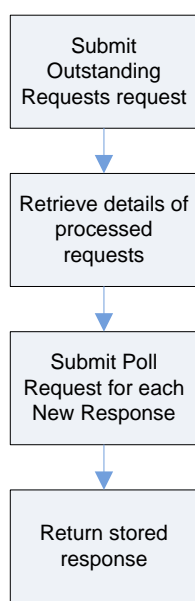


Figure 5: Business Gateway poll request processing flow

The following scenario demonstrates the polling flow for polling

1. A request requiring Land Registry processing is received out-of-hours, or is sent for manual processing
2. The request is stored by the gateway.
3. A response is sent back to the client with the unique message ID that the client has provided and relevant details (If the request was submitted out of hours, then the response has a predicted processing time. If the request goes for Manual Processing, then the response will contain the ABR).
4. The gateway has a scheduler that will run during working hours. The scheduler will fetch pending requests, re-authenticate the user, and then process the request. After the request is processed the response is stored in the gateway for retrieval.
5. The client sends an Outstanding Requests request. The gateway returns details of applications with a new response.
6. The client submits poll requests for each message returned by the Outstanding Request service.
7. The gateway verifies that the user ID of the current request belongs to the same organisation of the original request. The gateway sends the processed response stored in the gateway.
8. If a poll request is made for an application that does not have a new response yet, the gateway will return the previous response/acknowledgement.



2.4. Security

The solution enforces two levels of authentication. The first level of authentication will be established using digital certificates and the second level will be using user ID and password. Authorisation is via user ID and password. Transport level encryption will be applied over a HTTPS connection.

2.5. Interoperability

All web services exposed via the gateway are WS-I and Basic profile 1.1 compliant. Web services invocations using SOAP document literal style will be supported. Any CMS client using a programming language that supports web services such as Java, .net or C++ should be able to consume the WSDL files exposed by the gateway.

2.6. Reliable messaging

Web services requests over HTTP/HTTPS internet protocols are not truly reliable and are known for losing messages between request and response. A reliable messaging technique is implemented, to ensure messages are not lost between the client software and the gateway.

Figure demonstrates a normal style of interaction between a client and the gateway where messaging is assumed to be reliable.

Figure 7 shows the scenario where a message is sent to the gateway but is lost in transit.

Figure 8 shows the final scenario where a message is received by the gateway but the response fails to reach the client.

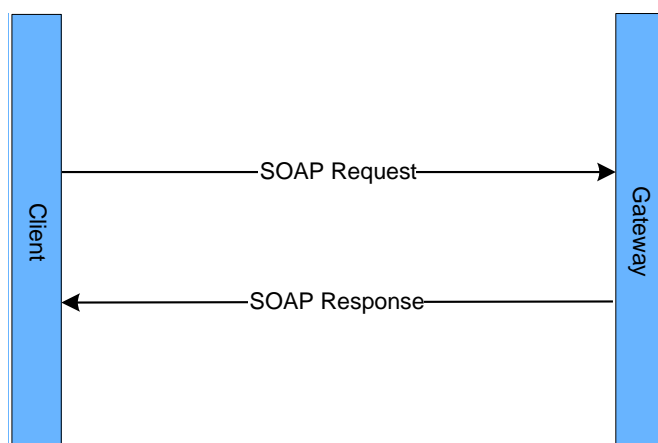


Figure 6: Normal SOAP request response processing



2.6.1. Use of unique message IDs

To facilitate reliable messaging, Business Gateway relies on the client producing a unique message ID that is sent with every request. The message ID is used to retrieve a response back from Business Gateway in the case where the original response has been lost (possibly due to network problems).

The unique combination of message ID, user ID and message type will be used by Business Gateway software internally to ensure that every request is unique in the system. If two requests are made with the same message ID from different CMS systems, although the message ID is same the user ID will be different. This forms a unique key and the message is therefore uniquely identified in the system.

2.6.2. Message is lost in transit from the client

The client code sends a SOAP request containing a unique message ID and should wait for a response. If the client does not receive a response within a reasonable period, it should send the request again with the same message ID. The client code can continue this pattern until it receives a response from the gateway.

The client code must ensure that all message requests have a unique message ID.

It will be responsibility of the gateway to correlate the SOAP request and response with the same message ID.

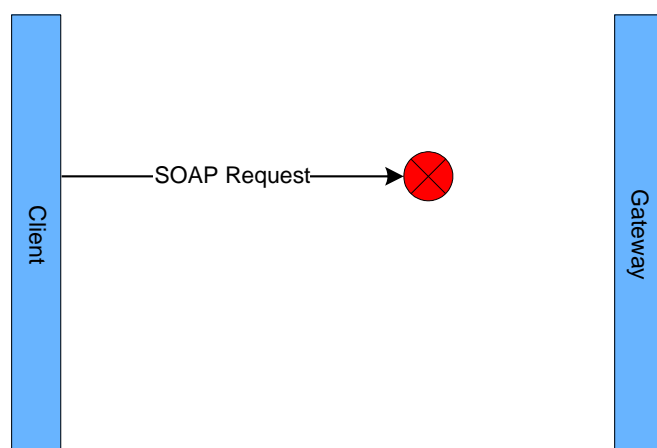


Figure 7: SOAP request is lost

2.6.3. Message is lost in transit from the gateway

The gateway will process the request for the first time and store the response. If a subsequent request is received with the same message and user ID, due to a message being lost in transit between the gateway and the client, then the response will be re-sent using the data stored in the gateway. This will ensure that the initial request will be processed only once.

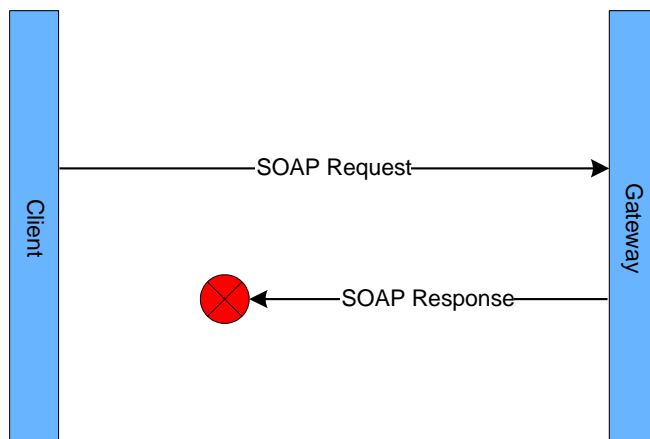


Figure 8: SOAP response is lost

2.6.4. Resubmission of a failed request

If a response contains a rejection or a SOAP fault then the client code should be able to handle the error condition, make necessary corrections and resubmit the request with a new messaging ID. Message IDs should **not** be re-used for resubmission. A SOAP fault will only be returned as a result of a system exception, for example a schema validation failure. Any rejections that come from the gateway in terms of a business rule validation failure will be sent as a response containing a rejection reason.

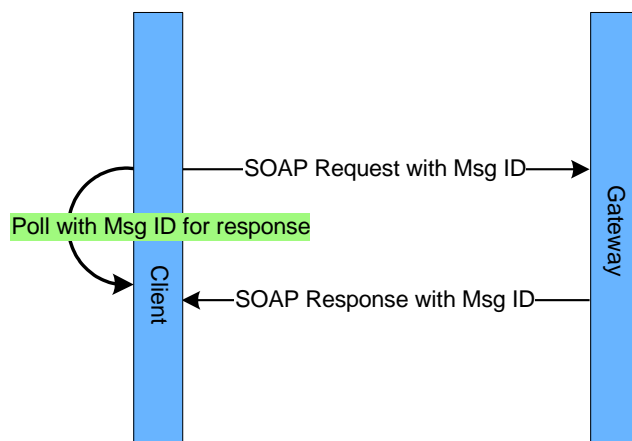


Figure 9: SOAP reliable messaging