Inland Revenue
Te Tari Taake

Inland Revenue

Build pack: Intermediation service

Date: 13/04/2018
Version: v1.00

UNCLASSIFIED

Contents

1 Overview 4
1.1 This solution 4
1.2 Intended audience 4
1.3 Supported onboarding packs 4
1.3.1 TDS Overview and Transition build pack—onboarding section 4
1.4 Related build packs 5
1.4.1 Transaction data services overview and transition build pack 5
1.4.2 Identity and access services build pack 5
1.5 Prerequisites 5
2 Solution design 6
2.1 Architecture 6
2.2 Service scope 6
2.3 Messaging 6
2.4 Security 8
3 Operations 11
3.1 RetrieveClientList 12
3.1.1 Request 12
3.1.2 Response 13
4 End points, schemas and WSDLs 14
4.1 End points 14
4.2 Schemas 15
4.3 WSDLs 15
5 Responses 16
5.1 Generic gateway response codes 16
5.2 Generic intermediation response codes 18
6 Processing flows 19
6.1 Generic use cases 19
6.1.1 RetrieveClientList 19
7 Appendix A—Glossary 20
8 Appendix B—Change log 23

List of tables and figures

Table 1: Prerequisites 5
Figure 1: Flow of data from user to Inland Revenue 6
Figure 2: SOAP request 7
Figure 3: SOAP response 7
Table 2: Ciphers and key strengths 8
Table 3: End points 10
Figure 4: Schema aliases 11
Table 4: Software platform data fields 11
Figure 5: RetrieveClientList operation structure 12
Table 5: RetrieveClientList request data fields 12
Figure 6: RetrieveClientList response structure 13
Table 6: RetrieveClientList response data fields 13
Table 7: Sliced data end points 14
Table 8: Unsliced data end points 14
Table 9: Production end points 14
Figure 7: WSDL naming conventions 15
Table 10: Generic Gateway Service response codes 17
Table 11: Intermediation response codes 18

1 Overview

1.1 This solution

Inland Revenue has a suite of digital services available for consumption by our service providers that supports efficient, electronic business interactions with Inland Revenue. The Intermediation service described in this build pack document forms part of a suite of Gateway Services.

Intermediation is the process of linking an intermediary (such as a tax agent) to an individual or organisation so the intermediary can act on their behalf for tax purposes. While the Intermediation Gateway Service will eventually provide this linking feature, initially it will only provide the ability to retrieve tax agent links already established in myIR.

This build pack document is intended to provide the technical information required to support the end-to-end onboarding of the Intermediation service. It describes the architecture of the technical solution, schemas, end points, sample payloads to use in non-production environments, and also its interaction with other build packs that cover different aspects of Gateway Services. The associated onboarding documents (see sections 1.3 and 1.4, below) describe the end-to-end business-level solution, of which this build pack forms part.

1.2 Intended audience

The solution outlined in this document is intended to be used by technical teams and development staff. It describes the technical interactions, including responses, provided by the Intermediation service.

The reader is assumed to have a suitable level of technical knowledge in order to understand the information provided. A range of technical terms and abbreviations are used throughout this document, and while most of these will be understood by the intended readers, a glossary is provided at the end of this document.

1.3 Supported onboarding packs

Before using this build pack, ensure the relevant onboarding pack has been consulted to provide business-level context. The Inland Revenue onboarding pack listed below is supported by this build pack, and is available on Inland Revenue's Gateway Services GitHub site: https://github.com/InlandRevenue/Gateway-Services

1.3.1 TDS Overview and Transition build pack—onboarding section

The onboarding section of the Transaction Data Services (TDS) build pack provides a guide for how consumers can onboard the various TDS components. It gives details of prerequisites, set-up requirements, testing, contact lists and more. It is intended to help organisations start using the TDS solution as quickly and easily as possible.

1.4 Related build packs

The following Gateway Services build packs complement this one and can be found at Inland Revenue's Gateway Services GitHub site: https://github.com/InlandRevenue/Gateway-Services.

1.4.1 Transaction data services overview and transition build pack

The Transaction data services overview and transition build pack was created to support service providers in their transition from Tax Agent Web Services to the use of TDS. It provides an overview of TDS, describes the data that will be made available through the services and processes, as well as giving use cases for how these services will be employed.

This TDS build pack also includes definitions for the Intermediation data bulk file. This data set will reflect the same data accessible by the Intermediation real time service defined in this build pack.

1.4.2 Identity and access services build pack

The Identity and Access (IAS) services build pack describes the operations provided under Identity and Access services, which is another part of the Gateway Services suite. These services are used to authenticate access.

1.5 Prerequisites

Party	Requirement	Description
Inland Revenue	Provide the Inland Revenue public certificate for mutual TLS	Inland Revenue's public X.509 certificate to support TLS will be provided as part of connectivity testing
Service provider	Acquire a X.509 certificate from a certificate authority for the Test and Production environments	This is required when using mutual TLS with cloud-based service providers

Table 1: Prerequisites

2 Solution design

2.1 Architecture

Inland Revenue is offering a suite of web services in order to facilitate interactions via software packages. The Gateway Services suite will be used by approved software providers to facilitate everything from registration activities, filing returns, making payments and other service offerings in order to allow customers to interact with Inland Revenue.

The diagram below illustrates the flow of data from the tax agent to Inland Revenue.

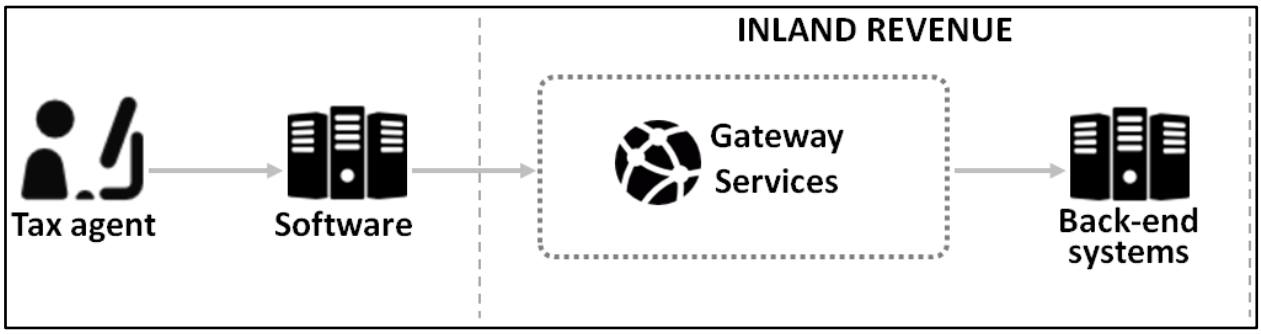

Figure 1: Flow of data from user to Inland Revenue

The WSDLs for the Gateway Services define an 'any' XML request and response structure, which then relies on a group of XSDs to define the data structure of those requests and responses. Each request and response type will define a lower, 'wrapper' element.

Any malformed XML will instantly be rejected by the Gateway Services prior to any schema validation.

2.2 Service scope

The Intermediation service supports the following operation:

- **RetrieveClientList:** This operation is used to retrieve client lists belonging to a tax agency as well as the clients in each list.

2.3 Messaging

All SOAP messages require a SOAP header containing **Action:** parameters, as well as a SOAP body containing a structured XML payload. Please refer to the WSDL (section 4.3 of this document) for the correct addresses.

The online WSDLs for the Gateway Services define an 'any' XML request and response structure, which then relies on a group of XSDs to define the data structure of those requests and responses. Each request and response type will define a lower, 'wrapper' element. To simplify analysis and code generation, a development-oriented version of the WSDL and XSDs is provided with the build pack that has the 'any' elements replaced with relevant types.

The Gateway Services allow the consumption of any structured XML payload but will be validated against the Inland Revenue-published XSDs.

This is a late-binding validation, performed after authentication has been reviewed. The message structure of these services is a simple request/response. The XML request will be checked for well-formed XML before the schema validation. Responses to these requests will be in XML format as well, and will be defined in the same schemas that define the requests. Any XML submissions in the SOAP body that do not meet the provided schemas will not be accepted by the Gateway Services. Incorrect namespaces will also fail validation against the published schemas.

Example SOAP request structure

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
        xmlns:int="https://services.ird.govt.nz/GWS/Intermediation/"
        xmlns:ret=https://services.ird.govt.nz/GWS/Intermediation/:types/RetrieveClientListRequest
        xmlns:a="http://www.w3.org/2005/08/addressing">
    <soap:Header/>
    <soap:Header>
        <a:Action>https://services.ird.govt.nz/GWS/Intermediation/Intermediation/RetrieveClie
        ntList</a:Action>
    </soap:Header>
    <soap:Body>
        <int:RetrieveClientList>
            <int:RetrieveClientListRequestMsg>
                <ret:RetrieveClientListRequestWrapper>
                    <!-- Intermediation Fields -->
                </ret:RetrieveClientListRequestWrapper>
            </int:RetrieveClientListRequestMsg>
        </int:RetrieveClientList>
    </soap:Body>
</soap:Envelope>
```

Figure 2: SOAP request

Example SOAP response structure

```
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope"
        xmlns:a=http://www.w3.org/2005/08/addressing
        xmlns:int1="https://services.ird.govt.nz/GWS/Intermediation/"
        xmlns:int2="urn:www.ird.govt.nz/GWS:types/Intermediation.v1"
        xmlns:b=https://services.ird.govt.nz/GWS/Intermediation/:types/RetrieveClientListResponse
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
    <s:Header>
        <a:Actions:mustUnderstand="1">
            https://services.ird.govt.nz/GWS/Intermediation/Intermediation/RetrieveClientList
        </a:Action>
    </s:Header>
    <s:Body>
        <int1:RetrieveClientListResponse>
            <int1:RetrieveClientListResult>
                <b:RetrieveClientListResponseWrapper>
                    <int:retrieveClientListResponse >
                        <!-- Response fields -->
                    </int:retrieveClientListResponse>
                </b:RetrieveClientListResponseWrapper>
            </int1:RetrieveClientListResult>
        </int1:RetrieveClientListResponse>
    </s:Body>
</s:Envelope>
```

Figure 3: SOAP response

2.4 Security

Gateway Services requests are access-controlled using an OAuth token that identifies the user making the request. Users will authenticate using their Inland Revenue myIR credentials. For instructions on how to acquire an OAuth token, review the Identity and Access build pack. For TDS Real Time web service requests, an OAuth access token is required in the HTTP header.

Authorisation for using the Gateway Services is defined in the permissions set in myIR. Permissions will reflect those granted in myIR. For example, if a user does not have permission to file a return online, they will not be able to file a return via Gateway Services either. This applies to users who are granted access as staff inside an organisation or as staff in a tax agency.

The Gateway Services use an HTTPS transport layer, with HTTP1.1 transport protocol supported.

The Gateway Services also use the SOAP version 1.2 protocol.

The SOAP service contract is published using WSDL version 1.1.

Transport layer encryption is mandatory and Gateway Services generally use the TLS version 1.2 specification.

Inland Revenue requires the following ciphers and key strengths to be used:

Encryption:	Advanced Encryption Standard (AES)	FIPS 197	256-bit key
Hashing:	Secure Hash Algorithm (SHA-2)	FIPS 180-3	SHA-256

Table 2: Ciphers and key strengths

There will be two end points, which are summarised in the bullet points below (the table immediately afterwards provides more detail):

1. There is an end point to which service providers' centralised **cloud** locations can connect. This will involve mutual TLS certificates that need to be exchanged during the onboarding phase. On the cloud end point, Inland Revenue has controls to shield service providers from issues caused by heavy usage by other providers.

2. For service providers connecting from **desktops,** there is a separate end point that does not use mutual TLS. For this service, certificates do not need to be exchanged during onboarding. On the desktop end point, Inland Revenue has less ability to shield consumers of the service from heavy usage by others.

	End point for cloud-based connections	End point for desktop connections
Purpose	• This is the default end point to connect software providers to the Gateway Services	• Additional end point provided to facilitate connecting from desktops which may have high volumes of sources addresses, transient DHCP addresses, not realistically associated with client-side TLS certificates, not individually onboarded to setup certificate trust

	End point for cloud-based connections	End point for desktop connections
Client application type	• Cloud applications	• Desktop/native applications • For connecting from multiple decentralised clients
Constraints	• Only for source locations with client-side TLS certificates • On the cloud end point Inland Revenue has controls to shield software providers from issues caused by heavy usage from other providers	• Less scalable • Subject to tighter security controls • On the desktop end point Inland Revenue has less ability to shield consumers of the service from heavy usage by others • OAuth2 refresh tokens will not be offered to desktop clients
Mutual TLS	• Inland Revenue explicitly trusts the certificate the software provider uses as client for Mutual TLS connections and uses it to identify the software provider in conjunction with the web service identification below	• Server-side certificates only
Minimum TLS version	• 1.2	• 1.0(+)
URL	• Contains …/gateway/..	• Contains …/gateway2/..
Port	• 4046	• 443 (Default https port)
Web service consumer identification	• Each software provider is given a software platform ID during onboarding. This ID type is "Customer ID" and is independent of the end point • To be identified in web service calls each cloud application will be given identity and access system client_id/client_secret credentials during onboarding to allow it to get OAuth tokens to call this end point • The mutual TLS certificate is used to identify the service provider	• Each software provider is given a software platform ID during onboarding. This ID type is "Customer ID" and is not specific to the end point • Desktop clients will be given different identity and access system client_id/client_secret credentials to cloud application clients

	End point for cloud-based connections	End point for desktop connections
Firewalling in production	• No IP address restrictions • Access limited by certificate enrolment	• No IP address restrictions
Firewalling in non-production environments	• No IP address restrictions • Access limited by certificate enrolment	• Firewalled—IP whitelisting required

Table 3: End points

Delegated permissions: The services will allow the retrieval of all of the data for a customer to whom the calling user (as represented by the OAuth token) has access. There may be additional accounts to which this identity does not have access—those will not be mentioned. An error will be returned if an account is targeted by the requestor but the user does not have permission. This access will depend on the delegation permissions set up in myIR. If the token represents a user in a tax agency or other intermediary, then the agent-client linking is also considered.

3 Operations

IMPORTANT: *The schemas listed here are subject to change. For the authoritative definitions, please refer to the information provided on the Inland Revenue Gateway Services GitHub site—* https://github.com/InlandRevenue/Gateway-Services

The structures of all Gateway Service operations are intended to produce the most efficient requests and responses. Any common structures and fields will be used across many schemas and tax types through an intentional inheritance method. The section below describes the structure of each operation and the scenarios in which certain fields will be used in XML requests and responses.

This section contains these schema aliases:

- Cmn: Common.v1.xsd
- Int: Intermediation.v1.xsd

All requests and responses live in the Intermediation.xsd.

All operations for the Intermediation service will contain two standard header fields: **softwareProviderData** and **identifier**.

The **identifier** field is common across all gateway services but refers to different parties in different services. In all cases it is the party with delegated permissions to whom an OAuth token is provided. If the value cannot be resolved to a known context, or if it can but the provided OAuth token does not have the necessary delegated permissions, then the error code 4 "unauthorised delegation" is returned. Please refer to individual operations for the nature of the identifier expected in this parameter in any given context.

For example:

```
<cmn:softwareProviderData>
      <cmn:softwareProvider>SoftwareProvider</cmn:softwareProvider>
      <cmn:softwarePlatform>SoftwarePlatform</cmn:softwarePlatform>
      <cmn:softwareRelease>v1</cmn:softwareRelease>
</cmn:softwareProviderData>
<cmn:identifier IdentifierValueType="IRD">012345678</cmn:identifier>
```

Figure 4: Schema aliases

Field	Description
softwareProvider	The company that developed the software
softwarePlatform	The software package that is making the request
softwareRelease	The version of the software package
IdentifierValueType	The ID type being submitted—can be IRD
identifier	The value submitted for this field should contain only digits, with no dashes. IRD numbers that are eight digits must be padded with a leading zero.

Table 4: Software platform data fields

Proper use:

- The only softwareProviderData fields into which users will be able to enter information are the ones that were provided to Inland Revenue at the time of onboarding.
- The identifier is that of the tax agency on whose behalf the operations are being performed.

Example scenario:

- Tax agency with IRD 898989898 wants to retrieve a client list
  - Tax agent with access to agency calls /Intermediation/RetrieveClientList/ with <cmn:identifier IdentifierValueType="IRD">898989898</cmn:identifier>

3.1 RetrieveClientList

The RetrieveClientList operation will be used to retrieve all or some of a tax agency's clients. There is also an option to retrieve all of a tax agency's clients of a given account type.

3.1.1 Request

```
<int:retrieveClientListRequest
      xmlns:int="urn:www.ird.govt.nz/GWS:types/Intermediation.v1"
      xmlns:cmn="urn:www.ird.govt.nz/GWS:types/Common.v1">
          <cmn:softwareProviderData>
                  <cmn:softwareProvider>…</cmn:softwareProvider>
                  <cmn:softwarePlatform>…</cmn:softwarePlatform>
                  <cmn:softwareRelease>…</cmn:softwareRelease>
          </cmn:softwareProviderData>
          <cmn:identifier IdentifierValueType= "IRD">123123123</cmn:identifier>
          <!--Optional-->
          <int:filterAccountType>FBT</int:filterAccountType>
          <!--Optional-->
          <int:filterClientListID>123154150</int:filterClientListID>
</int:retrieveClientListRequest>
```

Figure 5: RetrieveClientList operation structure

Field	Required	Description
Identifier	Required	This field will always have of IdentifierValueType IRD (see note below)
AccountType	Not used	This field is not used in Intermediation (optional field in Common.v1.xsd but not in figure above)
filterAccountType	Optional	This is used to limit the resulting accounts to only this specified account type
filterClientListID	Optional	This filter will be used to only retrieve one client list. This can be used in conjunction with filterAccountType.

Table 5: RetrieveClientList request data fields

NOTE: The IRD number submitted in the identifier field will **ALWAYS** be that of the tax agency and NOT that of the client list. If a single client list is desired be sure to use the filterClientListID field.

3.1.2 Response

```
<agency agencyID="123154150" agencyIDType="IRD">
    <clientList clientListId="123154150" clientListIdType="IRD">
        <client>
            <clientIRD>123154134</clientIRD>
            <clientAccountType>GST</clientAccountType>
        </client>
        <client>
            <clientIRD>123154126</clientIRD>
            <clientAccountType>FBT</clientAccountType>
        </client>
    </clientList>
    <clientList clientListId="123154452" clientListIdType="IRD">
        <client>
            <clientIRD>123163915</clientIRD>
            <clientAccountType>GST</clientAccountType>
        </client>
    </clientList>
</agency>
```

Figure 6: RetrieveClientList response structure

Field	Required	Description
agencyID	Required	This attribute of the <agency> tag will display the ID of the agency
clientListID	Required	This attribute of the <clientList> tag will display the ID of the client list
clientIRD	Required	Client's account IRD number
clientAccountType	Required	Client's account type

Table 6: RetrieveClientList response data fields

4 End points, schemas and WSDLs

IMPORTANT: *The end points, schemas and WSDLs listed here are subject to change. For the authoritative definitions, please refer to the information provided on the Inland Revenue Gateway Services GitHub site:* https://github.com/InlandRevenue/Gateway-Services

4.1 End points

The end points for the Digital Test Environment XZT (Sliced data) are as follows:

Service	Environment	URL
Authentication	Cloud	https://q.services.ird.govt.nz
	Desktop/native app	https://q.services.ird.govt.nz
Gateway Services	Cloud	https://xzt.services.ird.govt.nz:4046/gateway/gws/intermediation/
	Desktop/native app	https://xzt.services.ird.govt.nz/gateway2/gws/intermediation/

Table 7: Sliced data end points

The end points for the Digital Test Environment XZS (Unsliced data) are as follows:

Service	Environment	URL
Authentication	Cloud/desktop/native apps	https://q.services.ird.govt.nz
Gateway Services	Cloud	https://xzs.services.ird.govt.nz:4046/gateway/gws/intermediation/
	Desktop/native app	https://xzs.services.ird.govt.nz/gateway2/gws/intermediation/

Table 8: Unsliced data end points

The end points for Production are as follows:

Service	Environment	URL
Authentication	Cloud/desktop/native apps	https://services.ird.govt.nz:443
Gateway Services	Cloud	https://services.ird.govt.nz:4046/gateway/gws/intermediation/
	Desktop/native app	https://services.ird.govt.nz/gateway2/gws/intermediation/

Table 9: Production end points

4.2 Schemas

All schemas for the Intermediation service import a common.v1.xsd which has some data types specific to Inland Revenue. This common.v1.xsd will be used in other gateway services outside of the /Intermediation/ namespace so it must be kept up-to-date, without numerous redundant versions remaining.

The Intermediation.xsd creates data types to be used within the RetrieveClientList operation. It also includes the request and response root elements for the supported operation.

4.3 WSDLs

The Intermediation Gateway Service has one WSDL, which has a target namespace of https://services.ird.govt.nz/GWS/Intermediation/ and can be found at

https://services.ird.govt.nz/GWS/Intermediation/?singleWsdl.

As explained in the 'Solution design' section of this document, the online WSDLs have 'any' elements underneath the wrapper elements. There is a development version of the WSDL available with this build pack that replaces the 'any' element with an imported reference to the schema to facilitate initial development and testing. To consume the actual service, the binding will need to be done at the hosted end point. However, for initial development this static WSDL can be used.

All WSDL messages follow this naming convention:

```
Intermediation_RetrieveClientList_InputMessage and
Intermediation_RetrieveClientList__OutputMessage.

<wsdl:portType name="Intermediation">
            <wsdl:operation name="RetrieveClientList">
<wsdl:service name="Intermediation">
```

Figure 7: WSDL naming conventions

5 Responses

The response message from the Gateway Services will always include a status code and status message. These values will describe any successes or failures of a web service call. Following the status message will be the response data for the given operation.

5.1 Generic gateway response codes

The following response codes are common to all gateway service calls. This service applies framework security validation as follows, and the descriptions in the table below reflect that:

- RetrieveClientList operation–Customer Level Validation

Standard codes	Standard message	Description	Customer Level validation
-1	An unknown error has occurred	This is generally what will be returned for internal errors that are not due to the service request	Y
0	(Success)	Standard success code is 0	Y
1	Authentication failure	General authentication failure status. Could be the result of inability to validate security token.	Y
2	Missing authentication token(s)	Tokens were not included in the HTTP header as expected	Y
3	Unauthorised access	Access is not permitted for the requester to use the gateway services. Access could not be confirmed due to the failure of OAuth token validation. This could be due to invalid format of the token or infrastructure being unavailable.	Y
4	Unauthorised delegation	Access is not permitted for the requester to perform this operation for the submitted identifier. This code will be returned in any of these situations: • The submitted cmn:identifier has an invalid value. • The identifier type (IdentifierValueType attribute on cmn:identifier) supplied is invalid. • All the values above are valid but the provided OAuth token does not have delegated access to that customer or	Y

Standard codes	Standard message	Description	Customer Level validation
		account. This service operation will NOT return this code if the filterAccountType supplied does not exist for that identifier, or the OAuth token does not have access for that specific account.	
5	Unauthorised vendor	The vendor code provided does not permit access—has not been onboarded for this operation	Y
6	Authentication Expired	Token authentication has expired and needs to be refreshed. Note this will only be provided for a token that has been successfully used before. For an expired unused token response 3 above will be returned.	Y
(7)	(Account Type not supported)	Please note this service operation, unlike some other services, will **never** return the 7 response code for unsupported account types. An invalid or unsupported filterAccountType value might result in a 103 code as per the table below.	N/A
20	Unrecognised XML request	This could be due to the external sender sending in incorrect XML or it could be due to bad/poor/missing configuration	Y
21	XML request failed validation	The external requestor submitted XML that is not formatted according to our defined schemas.	Y
(none)	(non xml)	In some scenarios where the request message does not have a well formed XML structure or is not valid or does not adhere to the SOAP protocol formats, the framework generates a parsing exception that is not wrapped in XML nor has a response status code.	Y
(none)	(SOAP fault) UnAuthorised	When maximum concurrency has been exceeded by the service provider this SOAP fault will be returned	Y

Table 10: Generic Gateway Service response codes

5.2 Generic intermediation response codes

The following response codes are specific to intermediation gateway service calls:

Standard codes	Standard message	Description
100	Could not extract data from XML payload	Could not extract data from XML payload
101	Tax agency IRD is not valid	The provided IRD number does not belong to a tax agency
102	No client lists available for agent	The requesting agent has not been delegated access to any client lists
103	No results for requested parameters	There are no client accounts for the request parameters provided

Table 11: Intermediation response codes

6 Processing flows

6.1 Generic use cases

This section will convey how Inland Revenue intends the Gateway Services to be used. These are base cases and Inland Revenue recognises that these services can be used in many different ways to satisfy business needs.

Additional information on use cases is available in the Transaction Data Services Overview Build Pack.

6.1.1 RetrieveClientList

Scenario 1:

1) The user representing the intermediary wishes to retrieve a list of all clients linked to that intermediary.
2) The user invokes the 'RetrieveClientList' operation to attempt to retrieve a list of clients that belong to the intermediary.
3) Upon receipt of the request, the intermediation services will ensure the requestor has permission to make the request on behalf of the intermediary.
4) After validation the client list will be returned.

7 Appendix A—Glossary

Acronym/term	Definition
Authentication	The process that verifies the identity of the party attempting to access Inland Revenue
Authorisation	The process of determining whether a party is entitled to perform the function or access a resource
Build Pack	Details the technical requirements and specifications, processes and sample payloads for the specified activity
Client	As used in this build pack client generally refers to the party licensing and using the software intermediary/software provider's software
Credentials	Information used to authenticate identity, for instance an account username and password.
Customer	A Customer is the party who is a tax payer or a participant in the social policy products that are operated by Inland Revenue. The Customer might be a person (an "individual") or a non-individual entity such as a company, trust, society etc. Practically all of the service interactions with Inland Revenue are about a Customer (eg their returns, accounts, entitlements etc) even though these interactions might be undertaken by an Intermediary such as a tax agent on their behalf.
Encryption	Cryptographic transformation of data (called "plaintext") into a form (called "cipher text") that conceals the data's original meaning to prevent it from being known or used. If the transformation is reversible, the corresponding reversal process is called "decryption", which is a transformation that restores encrypted data to its original state. [RFC 2828]
End points	A term used to describe a web service that has been implemented.
GWS	Gateway Services—the brand name for the suite of web services that Inland Revenue is providing. The Software intermediation service is a Gateway Service.
HTTP, HTTPS	Hyper Text Transmission Protocol (Secure)—the protocol by which web browsers and servers interact with each other. When implemented over TLS1.2 HTTP becomes HTTPS.
IP	Internet Protocol—the principal communication protocol in the Internet protocol suite for relaying datagrams across networks
NZISM	NZ Information Security Manual—the security standards and best practices for Government agencies. Maintained by the NZ Government Communications Security Bureau (GCSB).
OAuth 2.0	OAuth 2.0 is an industry-standard protocol for authorisation
Pattern	A constraint on data type values that require the string literal used in the data type's lexical space to match a specific pattern
Payloads	This refers to the data contained within the messages that are exchanged when a web service is invoked. Messages consist of a header and a payload.

Acronym/term	Definition
Schemas	An XML schema defines the syntax of an XML document, in particular of a payload. The schema specifies what a valid payload (such as a GST return) must/can contain, as well as validating the payload.
SHA	Secure Hashing Algorithm. There is a family of these that provide different strengths. SHA-2 is currently favoured over SHA-1, which has been compromised.
Software provider	The organisation developing the software connecting to Inland Revenue gateway services (also known as software intermediary, software developer or service provider)
Software provider software	A client application is an operating instance of software that is deployed in one or more sites. A number of deployment patterns are possible: 1. A single cloud based instance with multiple tenants and online users, 2. An on premise instance (such as an organisation's payroll system) 3. A desktop application with an online user. This is the computer software that contains interfaces to consume the services that Inland Revenue exposes. Software is developed and maintained by a software developer and subsequently deployed as one or more client applications.
SFTP	Secure File Transport Protocol. SFTP 3.0 is used.
Solution	The technology components, systems and interface specifications constituting the Tax Agent Web Services capability which enables integration and communication across the gateway channel between Inland Revenue and tax agents for the purpose of providing the service
SOAP	Simple Object Access Protocol—a set of standards for specifying web services. Gateway Services uses SOAP version 1.2.
SSL	Secure Sockets Layer certificates—used to establish an encrypted connection between a browser or user's computer and a service or website
START	Simplified Taxation and Revenue Technology—Inland Revenue's new core tax processing application. It is an implementation of the GenTax product from FAST Enterprises.
Tax agent	A tax agent who is formally registered as such with Inland Revenue
TDS	Transaction Data Services
TLS1.2	Transport Layer Security version 1.2—the protocol that is observed between adjacent servers for encrypting the data that they exchange. Prior versions of TLS and all versions of SSL have been compromised and are superseded by TLS1.2.
URL	Universal Resource Locator—also known as a web address
User	The user referred to in this document is the user of the software provider accounting or tax package. This user needs delegated permissions on customer tax accounts (potentially via a tax agency or other intermediary) in order to use TDS. The web logon used in

Acronym/term	Definition
	eServices needs to be used in making Inland Revenue queries. This web logon must be granted permission there to access customer accounts.
WSDL	Web Service Definition Language—an XML definition of a web service interface
X.509 certificate	An international standard for encoding and describing a digital certificate. In isolation a public key is just a very large number, the X509 certificate to which it is bound identifies whose key it is, who issued it, when it expires etc. When a counterparty's X509 digital certificate is received, the recipient takes their public key out of it and store the key in their own keystore. The recipient can then use this key to encrypt and sign the messages that they exchange with this counterparty.
XIAMS	External IAMS—an instance of IAMS that authenticates and authorises access by external parties, for example customers, trading partners etc, as opposed to internal parties such as staff.
XML	eXtensible Mark-up Language—a language used to define a set of rules used for encoding documents in a format that can be read by humans and machines
XSD	XML Schema Definition—the current standard schema language for all XML data and documents

8 Appendix B—Change log

This table lists all changes that have been made to this build pack document since version 0.5 was created.

Version	Date of change	Document section	Description
1.0	13/04/2018	2.4 Security	• Reworded delegation note at end of section
		Entire document	• Cosmetic/formatting changes
0.82	06/03/2018	Entire document	• Cosmetic/formatting changes
0.81	23/02/2018	5.1 Generic gateway response codes	• Expanded some generic error code descriptions, and added others
		Glossary	• Replaced glossary
		Table of contents	• Added figure and table captions and indexes
0.8	31/01/2018	2 Solution design	• Updated definitions and XML formatting
		3 Operations	• Updated operation information and formatting
		5 Responses	• Updated response structures and attributes
	27/11/2017	2.4 Security	• Updated information
	27/11/2017	4.1 End points	• URLs updated
0.5	24/11/2017	All	• Draft created