



VetEnvoy[®] Services

API & Architectural Approach

Version 1.50

What's New

Version 1.5

- Update made to reflect the introduction of a different live server for North America (see section "VETENVOY TEST SITE AND LIVE SITE"). Also, a minor change to section "Microchip Registration service" section, to reflect that the Microchip Registration service in North America is currently not supporting "Live Processing".

Version 1.41

- Update to Microchip Registration Service, to i) make it compliant with version 1.3 of the Vet-XML Microchip Registration schema, and ii) to provide associated new errors codes with the live-processing facility.

Version 1.40

- Marketing lead generation service

Version 1.39

- Significant update to ordering service including notifications and dispatched messaging

Version 1.38

- Update to ordering service to include partial orders
- Clarification of error returned if the microchip database provider is not compatible

Version 1.37

- New Dietary Advice service
- Lab service now has the facility for an attachment to be returned
- Case History service now includes an optional initial request
- New ID service for animals
- New Catalogue and Ordering services
- Microchip service now includes new failure codes
- Live processing feature in microchip service

Version History

Version	
1.50	12/09/2011
1.41	24/03/2011
1.40	18/07/2010
1.39	10/06/2010
1.38 text update	26/03/2010
1.38	16/03/2010
1.37 text update	03/02/2010
1.37 correction	27/11/2009
1.37	03/11/2009
1.36	02/03/2009

1.35	15/02/2009
1.34	02/12/2008
1.33	06/11/2008
1.32	13/10/2008
1.31	09/09/2009
1.3	05/03/2008
1.2	21/02/2008

Contents

1.	INTRODUCTION	7
2.	METHOD OF COMMUNICATION	8
3.	AUTHENTICATION AND SECURITY	9
4.	CONCEPT OF 'CONVERSATION'	10
4.1	Starting the Conversation	10
4.2	Conversation Types.....	11
5.	POLLING (OBTAINING NOTIFICATIONS).....	12
5.1	Performing a Poll (i.e. Poll Request)	13
5.2	Obtaining the Poll contents (i.e. Poll Response)	14
6.	CONTENT LENGTH & TYPE	16
6.1	Content Length	16
6.2	Content Type.....	16
7.	RETURN CODES AND ERROR REPORTING.....	17
8.	LIVE PROCESSING	18
9.	ENCODING REQUIREMENTS.....	19
10.	VETENVOY TEST SITE AND LIVE SITE	20
10.1	Production Test site.....	20
10.2	Commercial 'LIVE' server	20
11.	SERVICE WORKFLOWS AND STATUS CODES.....	21
11.1	Insurance Claim service	22
11.2	Lab Report service	26
11.3	Microchip Registration service	30
11.4	Benchmark Data Delivery service.....	34
11.6	Case History Transfer service.....	37

11.7	Dietary Advice Facilitation service	42
11.8	eCatalogue service.....	47
11.9	Product Ordering service	50
11.10	Marketing Lead Generation service	57
11.11	SMS service	60
12.	ANCILLARY SERVICES	62
12.1	Attachment service	63
12.2	Query service.....	65
12.3	Batch service	68
12.4	Animal ID service	70
13.	APPENDIX A: FULLY WORKED EXAMPLE	71
13.1	Starting a New Claim (obtaining a Claim GUID).....	72
13.2	Uploading (Sending) the Claim	73
13.3	Starting an Attachment (obtaining an Attachment GUID).....	74
13.4	Uploading (Sending) the Attachment	75
13.5	Downloading (Receiving) the Claim	76
13.6	Confirming Claim Download	77
13.7	Downloading (Receiving) the Attachment.....	78
13.8	Confirming Attachment Download	79
13.9	Acknowledging Attachment download	80
13.10	Setting Claim Status to Processing.....	81
13.11	Starting a Query (obtaining a Query GUID).....	82
13.12	Uploading (Sending) the Query.....	83
13.13	Downloading (Receiving) the Query	84
13.14	Confirming Query Download.....	85
13.15	Uploading (Sending) the Query Reply	86
13.16	Downloading (Receiving) the Query Reply	87
13.17	Confirming Query Reply Download	88
13.18	Acknowledging the Query Reply download.....	89

13.19	Setting Claim Status to Settled	90
13.20	Acknowledging Claim as Settled	91
14.	APPENDIX B: CODE EXAMPLES	92
14.1	Polling VetEnvoy	92
14.2	Uploading XML to VetEnvoy	94
14.3	Downloading XML from VetEnvoy	96
14.4	Uploading an Attachment to VetEnvoy	98
14.5	Downloading an Attachment from VetEnvoy	100
15.	APPENDIX C: VETENVOY ERROR CODES	102
15.1	Bad Request (400)	102
15.2	Unauthorised (401)	102
15.3	Not found (404)	102
15.4	Conflict (409)	102
15.5	Internal server error (500)	102
15.6	Service Unavailable (503)	103
16.	APPENDIX E: TRANSFORMING THE LAB REPORT XML	104
17.	APPENDIX F: NULLIFYING A CONVERSATION	105
17.1	Nullifying	106
17.2	Acknowledging the Nullification	107
18.	LATEST PARTNER STATUS	108
19.	APPENDIX D: VETENVOY TEST TOOL	109

1. Introduction

This document provides an overview of the software architecture for the VetEnvoy services and the technique for communicating with them.

Disclaimer

The use of this VetEnvoy API is totally at the discretion of each party who decides to implement it. LiveTime 24/7 Ltd will not be held responsible for any damages of any nature arising from implementation and use of the API.

2. Method of Communication

The technique for communicating with VetEnvoy is through sending and receiving **XML messages over secure HTTP (i.e. HTTPS)**. In essence, as explained in the following sections, requests to VetEnvoy use the HTTP methods POST and GET sent to specific URIs. The technique is **easy to implement, highly flexible (easy to update), scalable, efficient (minimises bandwidth usage), resilient and secure**.

The software architecture for the VetEnvoy services is based on an architectural style known as **REST (i.e. Representational State Transfer)**. (A recommended reference on REST is: *“RESTful Web Services”* by Leonard Richardson and Sam Ruby, published by O'Reilly 22 May 2007, ISBN 978-0596529260.)

3. Authentication and Security

Security is a key design feature of VetEnvoy. When communicating with VetEnvoy, authentication details need to be sent as custom headers in the request. These headers should be sent as *name value* pairs. The required details will depend on which party is sending the request:

- **Vet (Practice):**
 - UserId** and **UserPassword**. The values for these will be unique for the particular veterinary practice.
 - VendorPassword**. This is the password for the particular vendor currently providing the Practice Management System (PMS) for the veterinary practice. It is required to ensure that only the Practice's current Vendor can send messages via VetEnvoy on their behalf.
 - RecipientId**. This is needed to uniquely identify the intended recipient. It is required only when information (such as the claim XML or lab report XML) is being sent to another user. It is NOT needed when simply polling VetEnvoy (see section 5).
- **Service Provider*:**
 - UserId** and **UserPassword**. The values for these will be unique for the particular service provider.
 - RecipientId**. See comments above against "Vet (Practice)".

** i.e. Insurance Company, Laboratory Company or Microchip Registration Database Provider.*

If these authentication headers are not provided, or if the values supplied are not recognised or supported, VetEnvoy will reject the request on security grounds.

Please note, a **SenderId** customer header (again as a *name value* pair) is returned in the response whenever relevant (i.e. when information, such as the claim XML or lab report XML, is being received from another user).

4. Concept of ‘Conversation’

An important concept with VetEnvoy is the term “conversation”. A conversation is considered to be a complete end-to-end workflow process (e.g. from insurance claim submission through to claim settlement). It involves at least two parties sending and receiving a series of messages (including data where relevant) through VetEnvoy during this process.

4.1 Starting the Conversation

To start a new conversation (e.g. to start a new insurance claim), the first step is to ask VetEnvoy to create a unique ID to represent it. This ID will then be used to uniquely identify the process (e.g. insurance claim) from its start to finish – i.e. from first sending the claim off to finally settling it. The ID will be in the form of a GUID (i.e. a unique 32-character string). This will be returned by VetEnvoy in XML format (in the entity body), for example:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<c id="41F8A092-EE5D-4F20-A06F-1A837839AC7E" />
```

The syntax of the XML structure is explained in Section 5. The GUID that is returned is used to represent the conversation at each step in the workflow (e.g. insurance claim) process and should be provided in the URI that all subsequent requests are sent to (see Sections 11 and 13 for further details).

4.2 Conversation Types

The different types of conversation are shown below. The conversation type is represented by a 4-digit code, as shown. The first two digits represent the primary service (e.g. Insurance Claim or Lab Report) while the second two represent the ancillary service (e.g. Query or Attachment, as associated with an Insurance Claim).

- 0100 – Insurance Claim
- 0101 – Query on a Claim
- 0102 – Attachment on a Claim
- 0200 – Lab Report
- 0300 – Microchip Registration
- 0301 – Query on a Microchip Registration
- 0400 – Benchmark data delivery
- 0401 – Query on a Benchmark data delivery
- 0500 – SMS Out
- 0600 – Case History
- 0700 – Dietary Advice
- 1000 – Catalogue
- 1100 – Product Ordering
- 1101 – Query on an Order
- 1300 – AnimalId
- 1500 – Marketing lead generation

5. Polling (Obtaining Notifications)

Another important concept with VetEnvoy is “polling”. To find out if any material is waiting to be collected, the recipient needs to poll VetEnvoy periodically. A poll interval of between 10 and 30 seconds is recommended.

In addition to telling if there is anything waiting on polling it, VetEnvoy will also tell if the stage in the workflow process has been moved on by the other party (for instance, with insurance claims, if a claim has been downloaded by the Insurance Company, or if it has been settled by them).

If the UserId utilised (see Section 3) represents one or more users in one or more practices then it is imperative that the poll is done and managed centrally on behalf of the group it represents.

5.1 Performing a Poll (i.e. Poll Request)

To perform a poll, an "HTTP GET" should be sent to:

<http://www.vetenvoy.org/poll/{ref}>

As indicated, a poll reference (as represented by “ref” in the URI above) needs to be provided:

- If polling VetEnvoy for the very first time ever, the value for this should be set to “0” (i.e. zero): <http://www.vetenvoy.org/poll/0>
- On all subsequent occasions, the value should be set to that as provided by VetEnvoy in the response of the immediately preceding poll (see Section 5.2 for details).

If the poll reference is not provided, VetEnvoy will reject the poll.

5.2 Obtaining the Poll contents (i.e. Poll Response)

Structure and Syntax of Poll XML

On polling VetEnvoy, the response obtained will contain XML (in the entity body). This XML will contain one or both of the following **elements**:

- p represents “Poll”
- c represents “Conversation” (e.g. an insurance claim or lab report, or a query or attachment relating to either of these).

It may also contain the following **attributes**:

- t represents the conversation “Type” (see section 4)
- id represents the “Conversation ID” (i.e. a unique 32-character string representing the conversation – see section 4)
- pid represents the “Parent Conversation ID” (i.e. if an insurance claim has a query or an attachment associated with it, the “pid” will be the conversation id of the associated ‘parent’ insurance claim)
- s represents the “Stage” in the conversation (i.e. the stage in the workflow process); from this, it is possible to work out what the next step in the workflow should be ([see Section 11 for details and a full description of these stages](#))
- ref See Section “Poll Reference” on page 15 below for an explanation of this.

Receiving an Empty Poll (i.e. with no content)

If VetEnvoy has *nothing* to report (i.e. there is nothing awaiting collection and there has been no change in the workflow process), the following XML will be passed back:

```
<?xml version="1.0" encoding="UTF-8" ?>
<p ref="{ref}">
```

Receiving a Poll with Content

If there *is* something to report, the XML will be sent in the following format (for example). One or more blocks will be returned, as shown, each representing a unique conversation (i.e. in this example, an insurance claim and an attachment associated with it).

```
<?xml version="1.0" encoding="UTF-8" ?>
<p ref="{ref}">
  <c t="0100" id="{GUID}" s="12002" />
  <c t="0102" pid="{GUID}" id="{GUID}" s="33000" />
</p>
```

Poll Reference

As shown in the previous section, VetEnvoy passes a poll reference back in the response as an attribute in the XML (represented by “ref” in the XML). **This critical value is a numerical marker representing the “last time” that a change in workflow process occurred against the particular user account in VetEnvoy. This value should be sent back to VetEnvoy in the request of the very next poll. It should be sent in the poll URI.**

This cycle of VetEnvoy providing the poll reference in the last poll response and then the calling user system passing it back in the very next poll request allows VetEnvoy to work out whether anything has changed between polls, sending information back only if so. In this way, VetEnvoy can avoid redundancy (i.e. sending the same information back over and over again), and in doing so this makes the system very responsive and ensures it is highly scalable.

Correct Poll Usage

- **The latest poll reference as provided by VetEnvoy should be set as such in the calling system only if ALL information as provided in the corresponding last poll has been properly saved (or acted upon) by the calling system. It is recommended that this is done within a transaction.**
- If the previous poll reference (i.e. not the latest) is sent up to VetEnvoy in the poll, VetEnvoy will return only the information that it originally sent when that particular poll reference was initially used. That is, it will not return any new information that may have come in to it since. Indeed, to obtain any new information, the correct (i.e. latest) poll reference as last returned by VetEnvoy must be supplied in the current poll.
- **If the UserId utilised (see Section 3) represents one or more users in one or more practices then it is imperative that the poll is done and managed centrally on behalf of the group it represents.**
- Should the calling user system ever lose track (for whatever reason) of the last valid poll reference that it sent to VetEnvoy and need to find out what this was, it should make a HTTP request (GET) to the following URL:

<http://www.vetenvoy.org/Poll/LastRef>

VetEnvoy will respond with XML containing the particular value. For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<p ref="18037">
```

6. Content Length & Type

6.1 Content Length

When any data is sent up to VetEnvoy in the entity body (i.e. message body) – such as when XML data or an attachment is uploaded – the content length of the data needs to be specified (in bytes). This should be done using the standard HTTP header, “Content-Length”, e.g:

Content-Length: 718

If the Content-Length is not specified, VetEnvoy will assume it is zero (i.e. no content). If the specified Content-Length is incorrect, VetEnvoy will abort the response and return an error.

Please note, an arbitrary limit of 10MB (Live Server) and 5MB (Test Server) has been set with regards to the content length. If an attempt is made to send a file that is larger than this arbitrary limit, VetEnvoy will instantly close the connection and the request will be terminated. The sending system should be prepared to handle this failure gracefully.

See also related Section 6.2.

6.2 Content Type

When any data is sent up to VetEnvoy in the entity body (i.e. message body) – such as when XML data or an attachment (see section 12.1) is uploaded – the content type of the data needs to be specified. By providing this information, the recipient of the data will be able to properly identify what type of file (data) it is. To specify the data type, the standard HTTP header, “Content-Type”, should be used. For example:

Content-Type: text/xml; charset=utf-8

Other standard values include:

Content-Type: application/pdf

Content-Type: application/msword

Content-Type: image/gif

Content-Type: image/jpeg

Content-Type: image/png

Content-Type: text/html; charset=utf-8

If the Content-Type is not specified even though the Content-Length (see section 6.1) is not zero, VetEnvoy will abort the response and return an error.

A list of common content types can be found at:

http://en.wikipedia.org/wiki/Internet_media_type .

A full list is available at: <http://www.iana.org/assignments/media-types>

See also related Section 6.1.

7. Return Codes and Error Reporting

If an error occurs during any part of the request process, VetEnvoy will return standard HTTP status codes (see Section 15). Where possible, VetEnvoy will also provide further details of the error as an XML message. The error XML will be in the following format:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<error>Encoding specified does not match encoding used</error>
```

If the request is successful, VetEnvoy will return in the response an HTTP "200 OK".

8. Live Processing

Certain services include real-time communication (i.e. live processing) with certain service providers, with instant feedback being provided on messages sent. In these instances, feedback from the service provider will be communicated immediately in the response through a pair of custom headers, **ResultCode** and **ResultDetails**.

As when processing the Poll XML (as discussed in Section 5.2), the ResultCode header represents the current “Stage” in the conversation (i.e. the stage in the workflow process). From this, it is possible to work out what the next step in the workflow is, and will allow any notifications to be handled immediately without having to poll for the latest status code ([see Section 11 for details and a full description of these stages](#)). Similarly, the ResultDetails header will provide some further details that could be presented to the user to help resolve any issue or to confirm success.

Please visit our website (www.vetenvoy.com) for latest information on which Partners and Suppliers support live processing.

9. Encoding Requirements

While VetEnvoy will accept XML using any encoding method, for bandwidth considerations UTF-8 is recommended. If no encoding is specified in the XML, UTF-8 is assumed.

If an encoding is specified in the XML that does not match the actual encoding used on the bytes sent, VetEnvoy will reject the message.

10.VetEnvoy Test Site and Live site

There are separate VetEnvoy sites for testing and for live usage. Details on these are explained and provided below.

For the purposes of explaining the functionality of the VetEnvoy services in this API document, the <https://www.vetenvoy.org> server address (i.e. the production test server running over HTTPS) has been assumed.

For live usage, the server addresses as specified below (i.e. the commercial 'live' servers running over HTTPS) can be substituted throughout. With these, **the appropriate 'live' user accounts and passwords would need to be deployed for VetEnvoy to accept and validate the requests.**

10.1 Production Test site

www.vetenvoy.ORG

This is intended solely for testing out the implementation and deployment of the VetEnvoy services. The user accounts associated with this server are simply test accounts and are different from those as issued for use with the commercial 'live' servers (see below). The production test server can be accessed over either HTTP or HTTPS. Please note, the certificate used to enable HTTPS has been issued by VeriSign and as such will be automatically trusted by the computer on which the vendor or supplier software runs. **For security reasons, we recommend that HTTPS is used when connecting to this site.**

10.2 Commercial 'LIVE' servers

www.vetenvoy.NET (UK) | www.vetenvoyinc.info (North America)

These are intended solely for making real-world (live) transactions. The user accounts associated with these servers are real live accounts and are different from those as issued for use with the production test server.

For security reasons, the commercial 'live' servers can be accessed only using SSL over HTTPS. The certificates used to enable this have been issued by recognised parties (e.g. VeriSign) and as such should be automatically trusted by the computer on which the vendor or supplier software runs.

11. Service Workflows and Status Codes

This section details the workflows and status codes for each service in the VetEnvoy suite. A detailed example can be found in Section 13 (Appendix A: Fully Worked Example), which describes an end-to-end conversation using the Insurance Claim service. This example also includes usage of the Query ancillary service and Attachment ancillary service.

Ancillary Services

The Query, Attachment and Batch services are all ancillary services (see Section 12). These services can be used only in relation to an existing conversation on a compatible primary service. For example, the insurance claim service allows calls into the Attachment and Query services at appropriate points in its workflow. The Query service can also be called at certain points during the Ordering service using the same techniques (and workflow and status codes).

Optional Sections

Some services contain optional parts. For example, the Lab Report service includes an optional lab request part, which can be sent by the practice to the Lab company ahead of the results being returned. In this case, if the Lab has the facility to accept requests (see www.vetenvoy.com for details of partner capabilities), this allows the results to be delivered on the same conversation, thereby enabling the PMS to reliably and easily link them to the relevant animal record.

Live Processing (Real-time Communication)

Live processing (see section 8) allows real-time communication on certain services with certain service providers (see www.vetenvoy.com for details of partner capabilities). For example, when sending microchip registrations to AVID's or Petlog's registration databases in the UK, instant feedback (i.e. in the same request-response cycle) is given on whether the registration was successful or not (and reasons why if not).

11.1 Insurance Claim service

There are a number of steps in the Insurance Claim workflow process, as follows:

1. Sending and receiving a claim
2. Sending and receiving an attachment associated with the claim (*optional*)
3. Making a query on a claim and replying to it (*optional*)
4. Setting the status of a claim to “processing”, “settled”, “declined” or “nullified”.

The stages involved in this service are detailed in the following table. For each of the stages, the action(s) that can be carried out next are provided along with the parameters required to make the request (e.g. “URI” and “Method”). For each action, the stage the conversation will move to, on success or failure, is also given.

A query can be started during the insurance claim service from status code 13000 to 19000. After the insurance claim has been settled or declined, any existing queries can still be completed but new ones cannot be started.

An attachment can be started during the service from status code 11000 to 19000 (NB the claim must be uploaded before any attachments are sent). After the insurance claim has been settled or declined, any existing attachments can still be completed but new ones cannot be started.

This service uses the Insurance Claim schema as defined by the Vet-XML Consortium. The latest version is available at <http://www.vetxml.org/StandardDataFormat.aspx>.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate GUID	https://www.vetenvoy.org/Claim/NewConversationId	GET	PMS	11000	
11000	GUID for new insurance claim has been generated	- Upload claim to Vet Envoy	https://www.vetenvoy.org/Claim/{id}	POST	PMS	12002 (13000)	12001
12001	Submitted claim contains invalid XML	- Re-upload claim to Vet Envoy, this time correctly formatted	https://www.vetenvoy.org/Claim/{id}	POST	PMS	12002 (13000)	12001
12002	Submitted claim is awaiting collection by Insurance Company	Not applicable	Not applicable				
13000	Claim available on VetEnvoy for downloading by Insurance Company	- Download claim from VetEnvoy	https://www.vetenvoy.org/Claim/{id}	GET	INS CO	13001	
13001	An attempt to download the claim has been made by the Insurance Company, however VetEnvoy has not yet received confirmation that this was successful.	- Confirm download of claim	https://www.vetenvoy.org/Claim/{id}/ConfirmDownload	POST	INS CO	13002	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
13002	Download of claim has been confirmed by the Insurance Company	- Set status of claim to "Processing" (see section 13.10)	https://www.vetenvoy.org/Claim/{id}/UpdateStatus/ClaimProcessing	POST	INS CO	19000	
		- Set status of claim to "Declined"	https://www.vetenvoy.org/Claim/{id}/UpdateStatus/ClaimDeclined	POST	INS CO	19002	
		- Set status of claim to "Settled" (see section 13.19)	https://www.vetenvoy.org/Claim/{id}/UpdateStatus/ClaimSettled	POST	INS CO	19001	
19000	Claim is being processed by Insurance Company	- Set status of claim to "Declined"	https://www.vetenvoy.org/Claim/{id}/UpdateStatus/ClaimDeclined	POST	INS CO	19002	
		- Set status of claim to "Settled" (see section 13.19)	https://www.vetenvoy.org/Claim/{id}/UpdateStatus/ClaimSettled	POST	INS CO	19001	
19001	Claim has been settled	- Acknowledge the claim as being settled	https://www.vetenvoy.org/Claim/{id}/Acknowledge	POST	PMS	19003	
19002	Claim has been declined	- Acknowledge the claim as being declined	https://www.vetenvoy.org/Claim/{id}/Acknowledge	POST	PMS	19003	
-999	Claim has been nullified	- Acknowledge the claim as being nullified	https://www.vetenvoy.org/Claim/{id}/Acknowledge	POST	INS CO	-1000	
-1000	Claim is nullified	Not applicable	Not applicable				

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
19003	Insurance claim service has ended.	Not applicable	Not applicable				

PMS = Practice Management System; INS CO = Insurance Company.

11.2 Lab Report service

There are a number of steps in the Lab Report workflow process, as follows:

1. Making a Lab Service requisition (*optional*)
2. Sending and receiving the Lab Report (as a whole or in batches)
 - VetEnvoy can be requested to transform the lab report into a particular version of the XML by providing an extra HTTP header (see section 16)
3. Acknowledging that the Lab Report (as a whole or batches thereof) has been delivered.

The stages involved in this service are detailed in the following table. For each of the stages, the action(s) that can be carried out next are provided along with the parameters required to make the request (e.g. “URI” and “Method”). For each action, the stage the conversation will move to, on success or failure, is also given.

Batch uploads can be started at status code 46000 (See section 12.3). After the status has moved on to 46001 or 45003, existing batch uploads can still be completed but new ones cannot be started.

An attachment can be started during the service from status code 41000 to 49000 (NB the lab result must be uploaded before any attachments are sent). After the status has moved on to 46001 or 45003, any created attachments can still be completed but new ones cannot be started.

This service uses the Lab Report schema as defined by the Vet-XML Consortium. The latest version is available at <http://www.vetxml.org/StandardDataFormat.aspx>.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate GUID	https://www.vetenvoy.org/Lab/NewConversationId	GET	PMS or LAB CO	41000	
41000	GUID for new Lab Service Requisition, or Lab Report, has been generated	- Upload Lab Service Requisition to Vet Envoy	https://www.vetenvoy.org/Lab/{id}	POST	PMS	42002 (43000)	42001
		- Upload Lab Report to Vet Envoy	https://www.vetenvoy.org/Lab/{id}	POST	LAB CO	44002 (45000)	44001
		- Begin the Lab Report batch upload	https://www.vetenvoy.org/Lab/{id}/Batch/Begin	POST	LAB CO	46000	
42001	Submitted Lab Service Requisition contains invalid XML	- Re-upload Lab Service Requisition to Vet Envoy, correctly formatted	https://www.vetenvoy.org/Lab/{id}	POST	PMS	42002 (43000)	42001
42002	Lab Service Requisition awaiting collection	Not applicable	Not applicable				
43000	Lab Service Requisition available on VetEnvoy for downloading	- Download Lab Service Requisition from VetEnvoy	https://www.vetenvoy.org/Lab/{id}	GET	LAB CO	43001	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
43001	An attempt to download the Lab Service Requisition has been made by the LAB CO, however VetEnvoy has not yet received confirmation that this was successful	- Confirm download of Lab Service Requisition	https://www.vetenvoy.org/Lab/{id}/ConfirmDownload	POST	LAB CO	43002	
43002	Download of Lab Service Requisition has been confirmed by LAB CO	- Upload Lab Report to Vet Envoy	https://www.vetenvoy.org/Lab/{id}	POST	LAB CO	44002 / (45000)	44001
		- Begin the Lab Report batch upload (Skip to 46000)	https://www.vetenvoy.org/Lab/{id}/Batch/Begin	POST	LAB CO	46000	
44001	Submitted Lab Report contains invalid XML	- Re-upload Lab Report to Vet Envoy, correctly formatted	https://www.vetenvoy.org/Lab/{id}	POST	LAB CO	44002 (45000)	44001
44002	Lab Report awaiting collection	Not applicable	Not applicable				
45000	Lab Report available on VetEnvoy for downloading	- Download Lab Report from VetEnvoy*	https://www.vetenvoy.org/Lab/{id}	GET	PMS	45001	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
45001	An attempt to download the Lab Report has been made by the PMS, however VetEnvoy has not yet received confirmation that this was successful	- Confirm download of Lab Report	https://www.vetenvoy.org/Lab/{id}/ConfirmDownload	POST	PMS	45002	
		- Download Lab Report from VetEnvoy	https://www.vetenvoy.org/Lab/{id}	POST	PMS	45001	
45002	Download of Lab Report has been confirmed by PMS	- Acknowledge the Lab Report as being successfully delivered	https://www.vetenvoy.org/Lab/{id}/Acknowledge	POST	LAB CO	45003	
46000	Lab Report batch upload process has begun	- End the batch upload	https://www.vetenvoy.org/Lab/{id}/Batch/End	POST	LAB CO	46001	
46001	Lab Report batch upload process has ended	- Acknowledge the Lab Report as being successfully delivered	https://www.vetenvoy.org/Lab/{id}/Acknowledge	POST	PMS	45003	
45003	Lab Report service ended.	Not applicable	Not applicable				

PMS = Practice Management System; LAB CO = Lab Company.

11.3 Microchip Registration service

There are a number of steps in the Microchip Registration workflow process, as follows:

1. Sending and receiving the Microchip Registration
2. Accepting or rejecting the registration. Steps 1 and 2 happen in real-time with database providers with live processing feedback (the AVID and Petlog databases in the UK currently support this – see www.vetenvoy.com for latest details).
3. Acknowledging that the Microchip Registration has been accepted (or rejected).

The stages involved in this service are detailed in the following table. For each of the stages, the action(s) that can be carried out next are provided along with the parameters required to make the request (e.g. “URI” and “Method”). For each action, the stage the conversation will move to, on success or failure, is also given.

This service uses the Microchip schema as defined by the Vet-XML Consortium. The latest version is available at <http://www.vetxml.org/StandardDataFormat.aspx>.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate GUID	https://www.vetenvoy.org/Chip/NewConversationId	GET	PMS	51000	
51000	GUID for new registration has been generated	- Upload microchip registration to Vet Envoy	https://www.vetenvoy.org/Chip/{id} (a)	POST	PMS	52002 or 54000 (b)	52001 or 54001 – 54108 (b)
<p>(a) With the service in the UK, when the microchip XML is uploaded a custom header, MicrochipNumber, must be provided so that VetEnvoy can deliver the registration to the correct database provider (the AVID and Petlog databases in the UK currently supported – see www.vetenvoy.com for latest details). The request will be rejected if this is not specified. If the microchip needs to be delivered to a database provider (in the UK) that is not currently compatible with VetEnvoy, then the upload will fail with an HTTP 400 (Bad Request) error and the following error message will be returned: “The specified microchip number does not resolve to a current supported database provider”. As in the UK the database provider is generally determined from the microchip number (first 3 characters), the RecipientID header is not required on this service in the UK. (NB: with the service in North America, as Live Processing is not currently supported, the MicrochipNumber header is <i>not</i> required, but the RecipientID header <i>is</i> required.)</p> <p>(b) With the service in the UK, if the database provider supports Live Processing (the AVID and Petlog databases in the UK currently do – see www.vetenvoy.com for latest details), the microchip registration will be processed by them in real-time and feedback will be given immediately in the response as a pair of custom headers (see Section 8). This will provide the practice with instant confirmation of either successful registration or any errors. The PMS should test for the existence of these custom headers to see if Live Processing has occurred.</p> <p>The status code will still be communicated in the poll XML as normal.</p>							
52001	Submitted microchip registration contains invalid XML	- Re-upload registration to Vet Envoy, this time correctly formatted	https://www.vetenvoy.org/Chip/{id}	POST	PMS	52002 (53000)	52000-52001
52002	Microchip registration awaiting collection	Not applicable	Not applicable				

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
53000	Registration available on VetEnvoy for downloading	- Download microchip registration from VetEnvoy	https://www.vetenvoy.org/Chip/{id}	GET	MRDP	53001	
53001	An attempt to download the registration has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm download of registration	https://www.vetenvoy.org/Chip/{id}/ConfirmDownload	POST	MRDP	53002	
53002	Download of registration has been confirmed	<ul style="list-style-type: none"> - Accept registration - Reject as microchip number already submitted - Reject as microchip database does not recognise sender - Reject as microchip number invalid - Reject as data is invalid / does not comply to schema 	https://www.vetenvoy.org/Chip/{id}/UpdateStatus/Accepted https://www.vetenvoy.org/Chip/{id}/UpdateStatus/Duplicate https://www.vetenvoy.org/Chip/{id}/UpdateStatus/InvalidSender https://www.vetenvoy.org/Chip/{id}/UpdateStatus/InvalidChipNo https://www.vetenvoy.org/Chip/{id}/UpdateStatus/InvalidData	POST	MRDP	54000 54100 54101 54102 54103	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
54000	Microchip registration has been accepted	- Acknowledge the registration as being accepted	https://www.vetenvoy.org/Chip/{id}/Acknowledge	POST	PMS	54003	
54100 / 54101 / 54102 / 54103 / 54104 / 54105 / 54106 / 54107 / 54108	Registration rejected: Duplicate registration Sender not recognised Invalid chip number Invalid data Implant Date Error PinNo Error General SQL Error Registration database currently unavailable (try again later) Duplicate registration and owner is different.	- Acknowledge the registration as being rejected	https://www.vetenvoy.org/Chip/{id}/Acknowledge	POST	PMS	54003	
54003	Registration service ended	Not applicable	Not applicable				

PMS = Practice Management System; MRDP = Microchip Registration Database Provider.

11.4 Benchmark Data Delivery service

There are a number of steps in the Benchmark workflow process, as follows:

1. Sending and receiving Benchmark data
2. Making a query and replying to it (*optional*)
3. Setting the status of the benchmark data to “processed”
4. Acknowledging the processing of the delivered data.

The stages involved in this service are detailed in the following table. For each of the stages, the action(s) that can be carried out next are provided along with the parameters required to make the request (e.g. “URI” and “Method”). For each action, the stage the conversation will move to, on success or failure, is also given.

A query can be started during the benchmark service from status code 63000 to 63002. After the benchmark has been processed created queries can still be completed but new ones cannot be started.

Please note, an arbitrary limit of 10MB (Live usage) and 5MB (Testing) has been set with regards to the content length. If an attempt is made to send a file that is larger than this arbitrary limit, VetEnvoy will instantly close the connection and the request will be terminated. The sending PMS should be prepared to handle this failure gracefully.

This service uses the Benchmark schema as defined by the Vet-XML Consortium. The latest version is available at <http://www.vetxml.org/StandardDataFormat.aspx>.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate GUID	https://www.vetenvoy.org/Benchmark/NewConversationId	GET	PMS	61000	
61000	GUID for new benchmark has been generated	- Upload benchmark to Vet Envoy	https://www.vetenvoy.org/Benchmark/{id}	POST	PMS	62002 (63000)	62001
62001	Submitted benchmark contains invalid XML	- Re-upload benchmark to Vet Envoy, this time correctly formatted	https://www.vetenvoy.org/Benchmark/{id}	POST	PMS	62002 (63000)	62001
62002	Submitted benchmark is awaiting collection by Benchmark Company	Not applicable	Not applicable				
63000	Benchmark available on VetEnvoy for downloading by Benchmark Company	- Download benchmark from VetEnvoy	https://www.vetenvoy.org/Benchmark/{id}	GET	BEN CO	63001	
63001	An attempt to download the benchmark has been made by the Benchmark Company, however VetEnvoy has not yet received confirmation that this was successful.	- Confirm download of benchmark	https://www.vetenvoy.org/Benchmark/{id}/ConfirmDownload	POST	BEN CO	63002	
63002	Download of benchmark has been confirmed by the Benchmark Company	- Set status of Benchmark to "Processed"	https://www.vetenvoy.org/Benchmark/{id}/UpdateStatus/BenchmarkProcessed	POST	BEN CO	69000	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
69000	Benchmark data has been processed by Benchmark Company	- Acknowledge the benchmark data as being processed	https://www.vetenvoy.org/Benchmark/{id}/Acknowledge	POST	PMS	69003	
69003	Benchmark service has ended	Not applicable	Not applicable				

11.6 Case History Transfer service

There are a number of steps in the Case History workflow process, as follows:

1. Sending and receiving a Case History Requisition (*optional*) (*i.e. out-of-hours emergency request*)
2. Sending and receiving a Case History
3. Making a query on a Case History and replying to it (*optional*)
4. Sending and receiving a Case History Reply (as a whole or in batches) (*optional*)
5. Acknowledging the end of the conversation.

The stages involved in this service are detailed in the following table. For each of the stages, the action(s) that can be carried next out are provided along with the parameters required to make the request (e.g. “URI” and “Method”). For each action, the stage the conversation will move to, on success or failure, is also given.

To assist with the identification of an animal with a Case History, a call can be made to the Animal ID service (see section 12.4) to obtain a unique ID (global unique identifier) which can be included in the case history XML.

Batch uploads can be started at status code 93002 (See section 12.3). After the status has moved on to 96001 or 96003, existing batch uploads can still be completed but new ones cannot be started.

This service uses the Case History schema as defined by the Vet-XML Consortium. The latest version is available at <http://www.vetxml.org/StandardDataFormat.aspx>.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate GUID for sending case history	https://www.vetenvoy.org/History/NewConversationId	GET	PMS1	91000	
		- Generate GUID for sending case history requisition	https://www.vetenvoy.org/History/NewConversationId/Requisition	GET	PMS2	90100	
90100	GUID for new history requisition has been generated	- Upload requisition to VetEnvoy	https://www.vetenvoy.org/History/{id}	POST	PMS2	90202 (90300)	90201
90201	Submitted requisition contains invalid XML or missing data	- Re-upload requisition to VetEnvoy	https://www.vetenvoy.org/History/{id}	POST	PMS2	90202 (90300)	90201
90202	Submitted requisition is awaiting collection by PMS1	Not applicable	Not applicable				
90300	History requisition available on VetEnvoy for downloading by PMS1	- Download requisition from VetEnvoy	https://www.vetenvoy.org/History/{id}	GET	PMS1	90301	
90301	An attempt to download the requisition has been made by PMS1, however VetEnvoy has not yet received confirmation that this was successful.	- Confirm download of requisition	https://www.vetenvoy.org/History/{id}/ConfirmDownload	POST	PMS1	90302	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
90302	Download of requisition has been confirmed by the PMS1	- Upload history to VetEnvoy	https://www.vetenvoy.org/History/{id}	POST	PMS1	92002 (93000)	92001
91000	GUID for new case history has been generated	- Upload history to VetEnvoy	https://www.vetenvoy.org/History/{id}	POST	PMS1	92002 (93000)	92001
92001	Submitted history contains invalid XML or missing data	- Re-upload history to VetEnvoy	https://www.vetenvoy.org/History/{id}	POST	PMS1	93000	92001
92002	Submitted history is awaiting collection by PMS	Not applicable	Not applicable				
93000	History available on VetEnvoy for downloading by PMS2	- Download history from VetEnvoy	https://www.vetenvoy.org/History/{id}	GET	PMS2	93001	
93001	An attempt to download history has been made by the PMS2, however VetEnvoy has not yet received confirmation that this was successful.	- Confirm download of history	https://www.vetenvoy.org/History/{id}/ConfirmDownload	POST	PMS2	93002	
		- Download history from VetEnvoy	https://www.vetenvoy.org/History/{id}	GET	PMS2	93001	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
93002	Download of history has been confirmed by the PMS2	- End conversation	https://www.vetenvoy.org/History/{id}/End	POST	PMS2	93003	94001
		- Upload reply	https://www.vetenvoy.org/History/{id}	POST	PMS2	94002 (95000)	
		- Begin batched reply	https://www.vetenvoy.org/History/{id}/Batch/Begin	POST	PMS2	96000	
93003	Case history ended	- Acknowledge End	https://www.vetenvoy.org/History/{id}/Acknowledge	POST	PMS1	95003	
94001	Submitted reply contains invalid XML or missing data	- Re-upload reply to VetEnvoy	https://www.vetenvoy.org/History/{id}	POST	PMS2	94002 (95000)	
94002	Submitted reply is awaiting collection by PMS1	Not applicable	Not applicable				
95000	History reply available on VetEnvoy for downloading by PMS1	- Download history reply from VetEnvoy	https://www.vetenvoy.org/History/{id}	GET	PMS1	95001	
95001	An attempt to download reply has been made by PMS1, however VetEnvoy has not yet received confirmation that this was successful.	- Confirm download of reply	https://www.vetenvoy.org/History/{id}/ConfirmDownload	GET	PMS1	95001	
		- Download reply from VetEnvoy	https://www.vetenvoy.org/History/{id}	GET	PMS1	95001	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
95002	Download of reply has been confirmed by PMS1	- Acknowledge	https://www.vetenvoy.org/History/{id}/Acknowledge	POST	PMS2	95003	
96000	Batch upload process has begun	- End the batch upload	https://www.vetenvoy.org/History/{id}/Batch/End	POST	PMS2	96001	
96001	Batch upload process has ended	- Acknowledge the reply as being successfully delivered	https://www.vetenvoy.org/History/{id}/Acknowledge	POST	PMS1	95003	

PMS1 = Practice Management System sending case history records;

PMS2 = Practice Management System receiving history records - PMS2 can optionally perform an initial request and/or return updates to PMS1;

11.7 Dietary Advice Facilitation service

There are a number of steps in the Dietary Advice workflow process, as follows:

1. Sending and receiving a Dietary Advice requisition
2. Sending and receiving a Dietary Advice reply (as a whole or in batches) (*optional*)

OR

3. Sending and receiving a Dietary Advice requisition rejection (*optional*)
4. Acknowledging the end of the conversation.

The stages involved in this service are detailed in the following table. For each of the stages, the action(s) that can be carried out next are provided along with the parameters required to make the request (e.g. “URI” and “Method”). For each action, the stage the conversation will move to, on success or failure, is also given.

Batch uploads can be started at status code 117000 (See section [12.3]). After the status has moved on to 117001 or 115003, existing batch uploads can still be completed but new ones cannot be started.

This service uses the Dietary Advice schema as defined by the Vet-XML Consortium. The latest version is available at <http://www.vetxml.org/StandardDataFormat.aspx>.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate GUID	https://www.vetenvoy.org/DietaryAdvice/NewConversationId	GET	PMS	111000	
111000	GUID for new dietary advice requisition has been generated	- Upload dietary advice requisition to Vet Envoy	https://www.vetenvoy.org/DietaryAdvice/{id}	POST	PMS	112002 (113000)	112001
112001	XML uploaded is invalid	- Re-Upload dietary advice requisition to Vet Envoy	https://www.vetenvoy.org/DietaryAdvice/{id}	POST	PMS	112002 (113000)	
112002	Submitted dietary advice requisition is awaiting collection	Not applicable	Not applicable				
113000	Dietary advice requisition available on VetEnvoy for downloading	- Download the dietary advice requisition from Vet Envoy	https://www.vetenvoy.org/DietaryAdvice/{id}	GET	NUT CO	113001	
113001	An attempt to download the dietary advice requisition has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm the download of the dietary advice requisition	https://www.vetenvoy.org/DietaryAdvice/{id}/ConfirmDownload	POST	NUT CO	113002	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
113002	Download of dietary advice requisition has been confirmed	- Nutritional Company ends conversation	https://www.vetenvoy.org/DietaryAdvice/{id}/End	POST	NUT CO	113003	
		- Nutritional Company uploads results	https://www.vetenvoy.org/DietaryAdvice/{id}	POST	NUT CO	114002 (115000)	114001
		- Begin the batch upload	https://www.vetenvoy.org/DietaryAdvice/{id}/Batch/Begin	POST	NUT CO	117000	
		- Nutritional Company rejects requisition (upload rejection XML)	https://www.vetenvoy.org/DietaryAdvice/{id}/Reject	POST	NUT CO	116002	116001
113003	Dietary advice requisition has ended.	- Practice acknowledge end of conversation	https://www.vetenvoy.org/DietaryAdvice/{id}/Acknowledge	POST	PMS	115003	
114001	Submitted dietary advice reply contains invalid XML	- Re-upload reply to VetEnvoy, correctly formatted	https://www.vetenvoy.org/DietaryAdvice/{id}	POST	NUT CO	114002 (115000)	114001
114002	Submitted dietary advice reply awaiting collection by practice	Not applicable	Not applicable				
115000	Reply available on VetEnvoy for downloading	- Download reply from VetEnvoy	https://www.vetenvoy.org/DietaryAdvice/{id}	GET	PMS	115001	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
115001	An attempt to download the dietary advice reply has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm download of reply	https://www.vetenvoy.org/DietaryAdvice/{id}/ConfirmDownload	POST	PMS	115002	
115002	Download of reply has been confirmed by PMS	- Acknowledge the reply as successfully delivered	https://www.vetenvoy.org/DietaryAdvice/{id}/Acknowledge	POST	NUT CO	115003	
116001	Submitted rejection contains invalid XML	- Re-upload rejection to VetEnvoy, correctly formatted	https://www.vetenvoy.org/DietaryAdvice/{id}	POST	NUT CO	116002	116001
116002	Submitted rejection awaiting collection by practice	Not applicable	Not applicable				
116003	Rejection available on VetEnvoy for downloading	- Download rejection from VetEnvoy	https://www.vetenvoy.org/DietaryAdvice/{id}	GET	PMS	116004	
116004	An attempt to download the rejection has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm download of rejection	https://www.vetenvoy.org/DietaryAdvice/{id}/ConfirmDownload	POST	PMS	116005	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
116005	Download of rejection has been confirmed by PMS	- Acknowledge the rejection as successfully delivered	https://www.vetenvoy.org/DietaryAdvice/{id}/Acknowledge	POST	NUT CO	115003	
117000	Batch upload process has begun	- End the batch upload	https://www.vetenvoy.org/DietaryAdvice/{id}/Batch/End	POST	NUT CO	117001	
117001	Batch upload process has ended	- Acknowledge the reply as successfully delivered	https://www.vetenvoy.org/DietaryAdvice/{id}/Acknowledge	POST	PMS	115003	
115003	Dietary advice service ended	Not applicable	Not applicable				

PMS = Practice Management System; NUT CO = Nutritional Company.

11.8 eCatalogue service

There are a number of steps in the eCatalogue workflow process. These steps apply when sending a full catalogue or a catalogue update.

If a catalogue publisher has multiple catalogues then these should be differentiated by using the “reference” attribute in the XML to provide a unique identifier for the catalogue. The “issuedate” attribute should be used to provide a date stamp for the catalogue to identify the most recent version. If a catalogue publisher wants to provide updates to a catalogue, the “type” attribute can be used. This attribute can contain either “F” for full or “U” for update to indicate to the receiving PMS whether the XML represents a full catalogue (a new catalogue or a complete revision of the catalogue) or a partial update.

To allow partial updates it is important that the “SupplierCode” field for each item in the XML is a unique value across a specific catalogue. This will enable a PMS system to locate the item being updated and replace the old information with the new data being provided.

1. Catalogue publisher uploads a catalogue (or part of it) for distribution
 - **A custom HTTP header ‘Description’ must be specified when uploading a catalogue (see table below).**
2. Catalogue publisher uploads a list of intended recipient practice(s)
3. Practice management systems download the catalogue material
 - **Before downloading the catalogue a HEAD request can be made to receive details about the catalogue (see table below). If the PMS is not configured to automatically download catalogues then this information could be presented to the user so that they can decide whether to download the catalogue.**

The stages involved in this service are detailed in the following table. For each of the stages, the action(s) that can be carried out next are provided along with the parameters required to make the request (e.g. “URI” and “Method”). For each action, the stage the conversation will move to, on success or failure, is also given.

This service uses the Product Ordering - Catalogue schema as defined by the Vet-XML Consortium. The latest version is available at <http://www.vetxml.org/StandardDataFormat.aspx>.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate GUID	https://www.vetenvoy.org/Catalogue/NewConversationId	GET	CAT	141000	
141000	GUID for new catalogue has been generated	- Upload catalogue to VetEnvoy	https://www.vetenvoy.org/Catalogue/{id} The custom header “Description” must be provided to give details about the catalogue being sent.	POST	CAT	142002	142001
141001	XML uploaded is invalid	- Re-upload catalogue to VetEnvoy, correctly formatted	https://www.vetenvoy.org/Catalogue/{id}	POST	CAT	142002	
142002	Submitted catalogue is awaiting distribution	- Upload catalogue distribution list to VetEnvoy	https://www.vetenvoy.org/Catalogue/{id}	POST	CAT	145003	143001
143001	XML uploaded is invalid	- Re-upload catalogue distribution list to VetEnvoy, correctly formatted	https://www.vetenvoy.org/Catalogue/{id}	POST	CAT	145003	
145003	Catalogue delivery has ended.	Not applicable	Not applicable				
Successful upload of a catalogue distribution list will automatically generate a collection of “Delivery” child conversations consisting of one conversation for each intended recipient. Each recipient will then receive notification of a waiting catalogue in their poll and can download the catalogue on the child conversation (see below).							

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
144000	Submitted catalogue is awaiting collection	- Download catalogue details (HTTP headers only)	https://www.vetenvoy.org/Catalogue/{pid}/Delivery/{id} As with a download, the “SenderId” header will be available to identify the sender. In addition, two extra headers, “CurrentBytes” and “Description”, will be sent which give the size of the catalogue and the description as provided by the sender on upload.	HEAD	PMS	144000	
		- Download catalogue	https://www.vetenvoy.org/Catalogue/{pid}/Delivery/{id}	GET	PMS	144001	
144001	An attempt to download the catalogue has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm the download of the catalogue	https://www.vetenvoy.org/Catalog/{pid}/Delivery/{id}/ConfirmDownload	POST	PMS	144003	
		- Re-download the catalogue	https://www.vetenvoy.org/Catalogue/{pid}/Delivery/{id}	GET	PMS	144001	
144003	Catalogue delivery has ended.	Not applicable	Not applicable				

PMS = Practice Management System; CAT = Catalogue publisher.

11.9 Product Ordering service

There are a number of steps in the Ordering workflow process, as follows:

1. Sending and receiving an order
2. Sending notifications and available actions to the practice, and the practice replying with their decisions (*optional*) – see '*Notification*' below
3. Sending and receiving dispatch information – see '*Dispatched*' below
4. Finalising the order process and ending the conversation.

The stages involved in this service are detailed in the following table. For each of the stages, the action(s) that can be carried out next are provided along with the parameters required to make the request (e.g. "URI" and "Method"). For each action, the stage the conversation will move to, on success or failure, is also given.

This service and its child services use the Product Ordering schemas (Order, OrderItem, Notification, NotificationAction and Dispatched) as defined by the Vet-XML Consortium. The latest versions are available at <http://www.vetxml.org/StandardDataFormat.aspx>.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate GUID	https://www.vetenvoy.org/Order/NewConversationId	GET	PMS	121000	
121000	GUID for new order has been generated	- Upload order to VetEnvoy	https://www.vetenvoy.org/Order/{id}	POST	PMS	122002 (123000)	121001
121001	XML uploaded is invalid	- Re-Upload order to VetEnvoy	https://www.vetenvoy.org/Order/{id}	POST	PMS	122002 (123000)	
122002	Submitted order is awaiting collection	Not applicable	Not applicable				
123000	Order available on VetEnvoy for downloading	- Download the order from VetEnvoy	https://www.vetenvoy.org/Order/{id}	GET	SUPP	123001	
123001	An attempt to download the order has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm the download of the order	https://www.vetenvoy.org/Order/{id}/ConfirmDownload	POST	SUPP	123002	
123002	Download of order has been confirmed by Supplier	<ul style="list-style-type: none"> - Begin notification flow - Begin dispatch flow - End order 	<p>See 'Notification' below.</p> <p>See 'Dispatched' below.</p> <p>https://www.vetenvoy.org/Order/{id}/End</p>	POST	SUPP	129003	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
129003	Order service has ended.	- Acknowledge the claim as being ended	https://www.vetenvoy.org/Order/{id}/Acknowledge	POST	PMS	129004	
129004	Order has ended	Not applicable	Not applicable				
An order can be nullified by the practice until the supplier has downloaded it.							
-999	Order has been nullified	- Acknowledge the claim as being nullified	https://www.vetenvoy.org/Order/{id}/Acknowledge	POST	SUPP	-1000	
-1000	Order is nullified	Not applicable	Not applicable				

Notifications

The following table describes the notification child service that can be called as part of the ordering service above. This service allows the supplier to notify the practice about changes or issues with their order and for the practice to tell the supplier how they would like to proceed.

The supplier sends Notification xml and the practice responds with NotificationAction xml.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate GUID	https://www.vetenvoy.org/Order/{pid}/Notification/NewConversation	GET	SUPP	131000	
131000	GUID for new notification has been generated	- Upload notification to VetEnvoy	https://www.vetenvoy.org/Order/{pid}/Notification/{id}	POST	SUPP	132002 (133000)	131001
131001	XML uploaded is invalid	- Re-Upload notification to VetEnvoy	https://www.vetenvoy.org/Order/{pid}/Notification/{id}	POST	SUPP	132002 (133000)	
132002	Submitted notification is awaiting collection	Not applicable	Not applicable				
133000	Notification available on VetEnvoy for downloading	- Download the notification from VetEnvoy	https://www.vetenvoy.org/Order/{pid}/Notification/{id}	GET	PMS	133001	
133001	An attempt to download the notification has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm the download of the notification	https://www.vetenvoy.org/Order/{pid}/Notification/{id}/ConfirmDownload	POST	PMS	133002	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
133002	Download of notification has been confirmed by PMS	- Send notification action to the supplier	https://www.vetenvoy.org/Order/{pid}/Notification/{id}	POST	PMS	134002 (135000)	134001
134001	XML uploaded is invalid	- Re-Send notification action to the supplier	https://www.vetenvoy.org/Order/{pid}/Notification/{id}	POST	PMS	134002 (135000)	134001
134002	Submitted notification action is awaiting collection	Not applicable	Not applicable				
135000	Notification action available on VetEnvoy for downloading	- Download the notification action from Vet Envoy	https://www.vetenvoy.org/Order/{pid}/Notification/{id}	GET	SUPP	135001	
135001	An attempt to download the notification action has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm the download of the notification action	https://www.vetenvoy.org/Order/{pid}/Notification/{id}/ConfirmDownload	POST	SUPP	135002	
135002	Download of notification action has been confirmed	- Acknowledge the notification action as being downloaded	https://www.vetenvoy.org/Order/{pid}/Notification/{id}/Acknowledge	POST	PMS	139003	
139003	Notification service has ended.	Not applicable	Not applicable				

Dispatched

The following table describes the dispatched child service that can be called as part of the ordering service above. This service allows the supplier to notify the practice about items that have been dispatched along with delivery details.

The supplier sends Dispatched xml.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate GUID	https://www.vetenvoy.org/Order/{pid}/Dispatched/NewConversationId	GET	SUPP	141000	
141000	GUID for new dispatched message has been generated	- Upload dispatched message to VetEnvoy	https://www.vetenvoy.org/Order/{pid}/Dispatched/{id}	POST	SUPP	142002 (143000)	141001
141001	XML uploaded is invalid	- Re-Upload dispatched message to VetEnvoy	https://www.vetenvoy.org/Order/{pid}/Dispatched/{id}	POST	SUPP	142002 (143000)	
142002	Submitted dispatched message is awaiting collection	Not applicable	Not applicable				
143000	Dispatched message available on VetEnvoy for downloading	- Download the dispatched message from VetEnvoy	https://www.vetenvoy.org/Order/{pid}/Dispatched/{id}	GET	PMS	143001	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
143001	An attempt to download the dispatched message has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm the download of the dispatched message	https://www.vetenvoy.org/Order/{pid}/Dispatched/{id}/ConfirmDownload	POST	PMS	143002	
143002	Download of dispatched message has been confirmed by PMS	- Acknowledge the dispatched message action as being downloaded	https://www.vetenvoy.org/Order/{pid}/Dispatched/{id}/Acknowledge	POST	SUPP	149003	
149003	Dispatched service has ended.	Not applicable	Not applicable				

PMS = Practice Management System; SUPP = Supplier.

11.10 Marketing Lead Generation service

There are a number of steps in the Lead Generation workflow process, as follows:

1. Sending marketing lead data
2. Ending the service

OR

3. Sending policy details (*optional*)
4. Acknowledging the end of the conversation.

The stages involved in this service are detailed in the following table. For each of the stages, the action(s) that can be carried out next are provided along with the parameters required to make the request (e.g. “URI” and “Method”). For each action, the stage the conversation will move to, on success or failure, is also given.

This service uses the Marketing Lead schema as defined by the Vet-XML Consortium. The latest version is available at <http://www.vetxml.org/StandardDataFormat.aspx>.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate GUID	https://www.vetenvoy.org/Lead/NewConversationId	GET	PMS	151000	
151000	GUID for new marketing lead has been generated	- Upload marketing lead to Vet Envoy	https://www.vetenvoy.org/Lead/{id}	POST	PMS	152002 (153000)	152001
152001	XML uploaded is invalid	- Re-Upload marketing lead to Vet Envoy	https://www.vetenvoy.org/Lead/{id}	POST	PMS	152002 (153000)	
152002	Submitted marketing lead is awaiting collection	Not applicable	Not applicable				
153000	Marketing lead available on VetEnvoy for downloading	- Download the marketing lead from Vet Envoy	https://www.vetenvoy.org/Lead/{id}	GET	INS CO	153001	
153001	An attempt to download the marketing lead has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm the download of the marketing lead	https://www.vetenvoy.org/Lead/{id}/ConfirmDownload	POST	INS CO	153002	
153002	Download of marketing lead has been confirmed	- Insurance Company ends conversation	https://www.vetenvoy.org/Lead/{id}/End	POST	INS CO	153003	154001
		- Insurance Company uploads policy details	https://www.vetenvoy.org/Lead/{id}	POST	INS CO	154002 (155000)	

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
153003	Marketing lead has ended.	- Practice acknowledge end of conversation	https://www.vetenvoy.org/Lead/{id}/Acknowledge	POST	PMS	155003	
154001	Submitted policy details contains invalid XML	- Re-upload policy details to VetEnvoy, correctly formatted	https://www.vetenvoy.org/Lead/{id}	POST	INS CO	154002 (155000)	154001
154002	Submitted policy details awaiting collection by practice	Not applicable	Not applicable				
155000	Policy details available on VetEnvoy for downloading	- Download policy details from VetEnvoy	https://www.vetenvoy.org/Lead/{id}	GET	PMS	155001	
155001	An attempt to download the policy details has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm download of policy details	https://www.vetenvoy.org/Lead/{id}/ConfirmDownload	POST	PMS	155002	
155002	Download of policy details has been confirmed by PMS	- Acknowledge the policy details as successfully delivered	https://www.vetenvoy.org/Lead/{id}/Acknowledge	POST	INS CO	155003	
155003	Marketing lead service ended	Not applicable	Not applicable				

PMS = Practice Management System; INS CO = Insurance Company.

11.11 SMS service

This service allows users to send SMS messages directly from their Practice Management System (PMS).

There are two steps in the SMS workflow process, as follows:

1. Sending the SMS XML
2. Acknowledging the delivery of the SMS or re-uploading where there is a failure.

The stages involved in this service are detailed in the following table. For each of the stages, the action(s) that can be carried out next are provided along with the parameters required to make the request (e.g. “URI” and “Method”). For each action, the stage the conversation will move to, on success or failure, is also given.

This service uses the SMS Message (Outbound) schema as defined by the Vet-XML Consortium. The latest version is available at <http://www.vetxml.org/StandardDataFormat.aspx>.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate GUID	https://www.vetenvoy.org/SmsOut/NewConversationId	GET	PMS	71000	
71000	GUID for new SMS has been generated	- Upload SMS to Vet Envoy	https://www.vetenvoy.org/SmsOut/{id}	POST	PMS	79000	72001 / 79001
72001	Submitted SMS contains invalid XML or missing data	- Re-upload SMS to Vet Envoy	https://www.vetenvoy.org/SmsOut/{id}	POST	PMS	79000	
79000	SMS sent	- Acknowledge the SMS as being sent	https://www.vetenvoy.org/SmsOut/{id}/Acknowledge	POST	PMS	79003	
79001	SMS failed	- Re-upload SMS to Vet Envoy	https://www.vetenvoy.org/SmsOut/{id}	POST	PMS	79000	72001 / 79001
79003	SMS service has ended	Not applicable	Not applicable				

12. Ancillary Services

The following services are ancillary services which are used with a compatible primary service. The Insurance Claim, Benchmark Data Delivery and Product Ordering services all support the Query ancillary service. Insurance Claims support the Attachment ancillary service. The Lab Report service provides the Batch ancillary service to allow results to be uploaded over multiple parts.

The workflow process and status codes used for these services are the same irrespective of the primary service.

12.1 Attachment service

There are two steps in the Attachment workflow process, as follows:

1. Sending and receiving an attachment
2. Acknowledging the end of the conversation.

The stages involved in this service are detailed in the following table. For each of the stages, the action(s) that can be carried out next are provided along with the parameters required to make the request (e.g. "URI" and "Method"). For each action, the stage the conversation will move to, on success or failure, is also given.

This service can send any binary file.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party		
-1	Conversation has not started	- Generate GUID	https://www.vetenvoy.org/{Service}/{pid}/Attachment/NewConversationId	GET	SEN	31001	
31000	GUID for new attachment has been generated	- Upload attachment to Vet Envoy	https://www.vetenvoy.org/{Service}/{pid}/Attachment/{id}	POST	SEN	32002	
32002	Submitted attachment is awaiting collection	Not applicable	Not applicable				
33000	Attachment available on VetEnvoy for downloading	- Download the attachment from Vet Envoy	https://www.vetenvoy.org/{Service}/{pid}/Attachment/{id}	GET	REC	33001	
33001	An attempt to download the attachment has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm the download of the attachment	https://www.vetenvoy.org/{Service}/{pid}/Attachment/{id}/ConfirmDownload	POST	REC	33002	
33002	Download of attachment has been confirmed	- Acknowledge the attachment as being downloaded	https://www.vetenvoy.org/{Service}/{pid}/Attachment/{id}/Acknowledge	POST	SEN	33003	
33003	Attachment service has ended.	Not applicable	Not applicable				

SEN = Sender; REC = Recipient.

12.2 Query service

There are a number of steps in the Query workflow process, as follows:

1. Sending and receiving a query
2. Sending and receiving a reply
3. Acknowledging the end of the conversation.

The stages involved in this service are detailed in the following table. For each of the stages, the action(s) that can be carried out next are provided along with the parameters required to make the request (e.g. “URI” and “Method”). For each action, the stage the conversation will move to, on success or failure, is also given.

This service uses the Message (for queries) schema as defined by the Vet-XML Consortium. The latest version is available at <http://www.vetxml.org/StandardDataFormat.aspx>.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate GUID	https://www.vetenvoy.org/{Service}/{pid}/Query/{id}/NewConversationId	GET	SEN	21000	
21000	GUID for new query has been generated	- Upload query to Vet Envoy	https://www.vetenvoy.org/{Service}/{pid}/Query/{id}	POST	SEN	22002 (23000)	21001
21001	XML uploaded is invalid	- Re-Upload query to Vet Envoy	https://www.vetenvoy.org/{Service}/{pid}/Query/{id}	POST	SEN	22002 (23000)	21001
22002	Submitted query is awaiting collection	Not applicable	Not applicable				
23000	Query available on VetEnvoy for downloading	- Download the query from Vet Envoy	https://www.vetenvoy.org/{Service}/{pid}/Query/{id}	GET	REC	23001	
23001	An attempt to download the query has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm the download of the query	https://www.vetenvoy.org/{Service}/{pid}/Query/{id}/ConfirmDownload	POST	REC	23002	
23002	Download of query has been confirmed	- Send a reply to the query	https://www.vetenvoy.org/{Service}/{pid}/Query/{id}	POST	REC	24002 (25000)	24001
24001	XML uploaded is invalid	- Re-Send a reply to the query	https://www.vetenvoy.org/{Service}/{pid}/Query/{id}	POST	REC	24002 (25000)	24001
24002	Submitted reply is awaiting collection	Not applicable	Not applicable				

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
25000	Reply available on VetEnvoy for downloading	- Download the reply from Vet Envoy	https://www.vetenvoy.org/{Service}/{pid}/Query/{id}	GET	SEN	25001	
25001	An attempt to download the reply has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm the download of the reply	https://www.vetenvoy.org/{Service}/{pid}/Query/{id}/ConfirmDownload	POST	SEN	25002	
25002	Download of reply has been confirmed	- Acknowledge the reply as being downloaded	https://www.vetenvoy.org/{Service}/{pid}/Query/{id}/Acknowledge	POST	REC	25003	
25003	Query service has ended.	Not applicable	Not applicable				

SEN = Sender; REC = Recipient.

12.3 Batch service

There are two steps in the Batch workflow process, as follows:

1. Sending and receiving some batched data
2. Acknowledging the end of the conversation.

The stages involved in this service are detailed in the following table. For each of the stages, the action(s) that can be carried out next are provided along with the parameters required to make the request (e.g. “URI” and “Method”). For each action, the stage the conversation will move to, on success or failure, is also given.

This service uses the schema as defined for the parent service.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate GUID	https://www.vetenvoy.org/{Service}/{pid}/Batch/{id}/NewConversationId	GET	SEN	81000	
81000	GUID for new batch has been generated	- Upload batch to Vet Envoy	https://www.vetenvoy.org/{Service}/{pid}/Batch/{id}	POST	SEN	82002	81001
81001	XML uploaded is invalid	- Re-Upload batch to Vet Envoy	https://www.vetenvoy.org/{Service}/{pid}/Batch/{id}	POST	SEN	82002	
82002	Submitted batch is awaiting collection	Not applicable	Not applicable				
83000	Batch available on VetEnvoy for downloading	- Download the batch from Vet Envoy	https://www.vetenvoy.org/{Service}/{pid}/Batch/{id}	GET	REC	83001	
83001	An attempt to download the batch has been made, however VetEnvoy has not yet received confirmation that this was successful	- Confirm the download of the batch	https://www.vetenvoy.org/{Service}/{pid}/Batch/{id}/ConfirmDownload	POST	REC	83002	
83002	Download of batch has been confirmed	- Acknowledge the batch as being downloaded	https://www.vetenvoy.org/{Service}/{pid}/Batch/{id}/Acknowledge	POST	SEN	83003	
83003	Batch service has ended.	Not applicable	Not applicable				

SEN = Sender; REC = Recipient.

12.4 Animal ID service

The Animal ID service creates a unique ID (guaranteed unique identifier) for use in services such as the Case History Transfer service (see section 11.5). With that service, a unique ID is created by making a GET request, as described below, and the ID returned can be included in the AnimalID field of the Case History XML. The PMS receiving the case history XML can then use this ID to determine if they have handled a referral for this animal before. With such repeat referrals, the receiving practice would know not to create a new animal record, as they would already have the animal recorded in the system under the unique ID.

Work-flow stage (code)	Description	Next possible action(s)					
		Action	URI (for HTTP Request)	HTTP method	Party	Success	Failure
-1	Conversation has not started	- Generate Animal ID	https://www.vetenvoy.org/AnimalId/NewId	GET	Any user	An ID is returned	Standard HTTP failure codes

13. Appendix A: Fully Worked Example

The insurance Claim service, including dealing with a query and sending an attachment, will be used as an example of how to implement VetEnvoy.

There are a number of steps in the Insurance Claim workflow process, as follows:

1. Sending and receiving a claim
2. Sending and receiving an attachment associated with the claim (*optional*)
3. Making a query on a claim and replying to it (*optional*)
4. Setting the status of a claim to “processing”, “settled”, “declined” or “nullified”.

With each of these steps, the parameters required to make the request (e.g. “URI”, “Method” and, where relevant, “Entity body”) are provided along with the response that can be expected.

The principles used within this example can be applied to each and every other service within the VetEnvoy suite.

To help test the implementation of the suite of services, the “VetEnvoy Test Tool” can be used to imitate each and any step in the workflow process. See section 19 for details on how to obtain this facility.

13.1 Starting a New Claim (obtaining a Claim GUID)

This is undertaken by the Practice Management System (PMS).

Please note, “id” below represents a GUID (see section 4) that uniquely identifies the particular insurance claim being started.

Request

URI: <https://www.vetenvoy.org/Claim/NewConversationId>

Method: GET

Response

200 OK

```
<?xml version="1.0" encoding="UTF-8" ?>
<c id="{id}" />
```

Following this initial step in the workflow, the next step is for the PMS to upload the insurance claim to VetEnvoy (see next section [13.2]).

13.2 Uploading (Sending) the Claim

This is undertaken by the Practice Management System (PMS).

Request

Entity body:

```
<?xml version="1.0" encoding="UTF-8" ?>
<InsuranceClaim xmlns="http://www.vetxml.org/schemas/InsuranceClaim" version="1.09">
  <Identification>...</Identification>
  <InfoFromPolicyHolder>...</InfoFromPolicyHolder >
  <InfoFromVet>...</InfoFromVet >
</InsuranceClaim>
```

URI: <https://www.vetenvoy.org/Claim/{id}>

Method: POST

Headers: The Guid of the Insurance Company to whom this claim is being sent must be specified as an additional header **RecipientId** (see section [3]).

The Content Length and Content Type of the claim being uploaded must also be specified (see sections [6.1] and [6.2]).

Please note, a claim that has already been uploaded to VetEnvoy can be replaced by re-uploading it (using this same technique) at any point provided it has not already been downloaded by the Insurance Company. Once it has, the PMS will need to resort to either nullifying the claim (see section 17) or sending a relevant query to the Insurance Company (functionality not yet documented) to deal with it.

Response

200 OK

Following this step in the workflow, the next step is for the Insurance Company to download the awaiting claim from VetEnvoy (see section [13.5]).

If one or more attachments or queries are to accompany the claim, then that process can be started by the PMS at this point too (see sections [12.1] and [12.2] for details).

13.3 Starting an Attachment (obtaining an Attachment GUID)

This is undertaken by the Practice Management System (PMS).

Please note, “id” below represents a GUID (see section 4) that uniquely identifies the particular attachment being started.

Request

URI: <https://www.vetenvoy.org/Claim/{id}/Attachment/NewConversationId>

Method: GET

Response

200 OK

```
<?xml version="1.0" encoding="UTF-8" ?>
<c id="{attachmentid}" />
```

Following this initial step in the workflow, the next step is for the PMS to upload the attachment to VetEnvoy (see next section [13.4]).

13.4 Uploading (Sending) the Attachment

This is undertaken by the Practice Management System (PMS).

Request

Entity body:

Attachments allow and binary data to be transferred as well as xml.

URI: <https://www.vetenvoy.org/Claim/{id}/Attachment/{attachmentid}>

Method: POST

Headers: The Guid of the Insurance Company to whom this attachment is being sent must be specified as an additional header **RecipientId** (see section [3]).

The Content Length and Content Type of the attachment being uploaded must also be specified (see sections [6.1] and [6.2]).

Response

200 OK

Following this step in the workflow, the next step is for the Insurance Company to download the awaiting claim and attachment from VetEnvoy (see next section [13.5]).

13.5 Downloading (Receiving) the Claim

This is undertaken by the Insurance Company's system.

Request

URI: <https://www.vetenvoy.org/Claim/{id}>

Method: GET

Response

200 OK

Entity body:

```
<?xml version="1.0" encoding="UTF-8" ?>
<InsuranceClaim xmlns="http://www.vetxml.org/schemas/InsuranceClaim" version="1.09">
  <Identification>...</Identification>
  <InfoFromPolicyHolder>...</InfoFromPolicyHolder>
  <InfoFromVet>...</InfoFromVet>
</InsuranceClaim>
```

Please note, a **SenderId** customer header (as a *name value* pair) will also be returned in the response.

Following this step in the workflow, the next step is for the Insurance Company to confirm to VetEnvoy that they have successfully downloaded the claim (see next section [13.6]).

13.6 Confirming Claim Download

This is undertaken by the Insurance Company's system.

Request

URI: <https://www.vetenvoy.org/Claim/{id}/ConfirmDownload>

Method: POST

Response

200 OK

Following this step in the workflow, there are three possible steps that the Insurance Company can take next:

- Set the status of the claim to "Processing" (see section 13.10)
- Set the status of the claim to "Declined"
- Set the status of the claim to "Settled" (see section 13.19)

13.7 Downloading (Receiving) the Attachment

This is undertaken by the Insurance Company's system.

Request

URI: <https://www.vetenvoy.org/Claim/{id}/Attachment/{attachmentId}>

Method: GET

Response

200 OK

Entity body:

Any binary data as denoted by the Content Type specified in the HTTP headers.

Please note, a **SenderId** customer header (as a *name value* pair) will also be returned in the response.

Following this step in the workflow, the next step is for the Insurance Company to confirm to VetEnvoy that they have successfully downloaded the attachment (see next section [13.8]).

13.8 Confirming Attachment Download

This is undertaken by the Insurance Company's system.

Request

URI: <https://www.vetenvoy.org/Claim/{id}/Attachment/{attachmentId}/ConfirmDownload>

Method: POST

Response

200 OK

13.9 Acknowledging Attachment download

This is undertaken by the Practice Management System (PMS). It is the final step in the Attachment ancillary workflow process.

Request

URI: <https://www.vetenvoy.org/Claim/{id}/Attachment/{attachmentid}/Acknowledge>

Method: POST

Response

200 OK

13.10 Setting Claim Status to Processing

This is undertaken by the Insurance Company's system.

Request

URI: <https://www.vetenvoy.org/Claim/{id}/UpdateStatus/ClaimProcessing>

Method: POST

Response

200 OK

Following this step in the workflow, there are two possible steps that the Insurance Company can take next (as applicable), as follows:

- Set the status of the claim to “Declined”
- Set the status of the claim to “Settled” (see section 13.19)

13.11 Starting a Query (obtaining a Query GUID)

This is undertaken by the Insurance Company in this example however the Practice Management System (PMS) could also initiate a Query.

Please note, "id" below represents a GUID (see section 4) that uniquely identifies the particular query being started.

Request

URI: <https://www.vetenvoy.org/Claim/{id}/Query/NewConversationId>

Method: GET

Response

200 OK

```
<?xml version="1.0" encoding="UTF-8" ?>
<c id="{queryid}" />
```

Following this initial step in the workflow, the next step is for the PMS to upload the query to VetEnvoy (see next section [13.12]).

13.12 Uploading (Sending) the Query

This is undertaken by the Insurance Company.

Request

Entity body:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Message xmlns="http://www.vetxml.org/schemas/Message" version="1.00">
  <From>...</From>
  <Subject>...</Subject>
  <Priority>...</Priority>
  <Body>...</Body>
</Message>
```

URI: <https://www.vetenvoy.org/Claim/{id}/Query/{queryid}>

Method: POST

Headers: The Guid of the Practice Management System to whom this query is being sent must be specified as an additional header **RecipientId** (see section [3]).

The Content Length and Content Type of the query being uploaded must also be specified (see sections [6.1] and [6.2]).

Response

200 OK

Following this step in the workflow, the next step is for the Practice Management System to download the query from VetEnvoy (see next section [13.13]).

13.13 Downloading (Receiving) the Query

This is undertaken by the Practice Management System

Request

URI: <https://www.vetenvoy.org/Claim/{id}/Query/{queryId}>

Method: GET

Response

200 OK

Entity body:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Message xmlns="http://www.vetxml.org/schemas/Message" version="1.00">
  <From>...</From>
  <Subject>...</Subject>
  <Priority>...</Priority>
  <Body>...</Body>
</Message>
```

Please note, a **SenderId** customer header (as a *name value* pair) will also be returned in the response.

Following this step in the workflow, the next step is for the Practice Management System to confirm to VetEnvoy that they have successfully downloaded the query (see next section [13.14]).

13.14 Confirming Query Download

This is undertaken by the Practice Management System.

Request

URI: <https://www.vetenvoy.org/Claim/{id}/Query/{queryId}/ConfirmDownload>

Method: POST

Response

200 OK

Following this step in the workflow, the next step is for the Practice Management System to reply to the query (see next section [13.15]).

13.15 Uploading (Sending) the Query Reply

This is undertaken by the Practice Management System.

Request

Entity body:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Message xmlns="http://www.vetxml.org/schemas/Message" version="1.00">
  <From>...</From>
  <Subject>...</Subject>
  <Priority>...</Priority>
  <Body>...</Body>
</Message>
```

URI: <https://www.vetenvoy.org/Claim/{id}/Query/{queryid}>

Method: POST

Headers: The Guid of the Insurance Company to whom this query is being sent must be specified as an additional header **RecipientId** (see section [3]).

The Content Length and Content Type of the query being uploaded must also be specified (see sections [6.1] and [6.2]).

Response

200 OK

Following this step in the workflow, the next step is for the Insurance Company to download the query reply from VetEnvoy (see next section [13.16]).

13.16 Downloading (Receiving) the Query Reply

This is undertaken by the Insurance Company.

Request

URI: <https://www.vetenvoy.org/Claim/{id}/Query/{queryId}>

Method: GET

Response

200 OK

Entity body:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Message xmlns="http://www.vetxml.org/schemas/Message" version="1.00">
  <From>...</From>
  <Subject>...</Subject>
  <Priority>...</Priority>
  <Body>...</Body>
</Message>
```

Please note, a **SenderId** customer header (as a *name value* pair) will also be returned in the response.

Following this step in the workflow, the next step is for the Insurance Company to confirm to VetEnvoy that they have successfully downloaded the query (see next section [13.17]).

13.17 Confirming Query Reply Download

This is undertaken by the Insurance Company.

Request

URI: <https://www.vetenvoy.org/Claim/{id}/Query/{queryId}/ConfirmDownload>

Method: POST

Response

200 OK

13.18 Acknowledging the Query Reply download

This is undertaken by the Practice Management System (PMS). It is the final step in the Query ancillary workflow process.

Request

URI: <https://www.vetenvoy.org/Claim/{id}/Query/{queryid}/Acknowledge>

Method: POST

Response

200 OK

13.19 Setting Claim Status to Settled

This is undertaken by the Insurance Company's system.

Request

URI: <https://www.vetenvoy.org/Claim/{id}/UpdateStatus/ClaimSettled>

Method: POST

Response

200 OK

Following this step in the workflow, the next step is for the PMS to tell VetEnvoy that it is aware that the claim has been settled (see next section [13.20]).

13.20 Acknowledging Claim as Settled

This is undertaken by the Practice Management System (PMS). It is the final step in the claim workflow process in the event that the claim was settled or declined by the Insurance Company beforehand.

Its purpose is to provide closure of the claim (and subsequent archiving in VetEnvoy). In effect, through this step the PMS can tell VetEnvoy that they are aware that the Insurance Company has either settled the claim or declined the claim and that no further communication is required with regards to it.

NB: Once this step is completed successfully, no further communication is possible between the parties with regard to the claim. That is, the claim will be treated as closed by VetEnvoy.

Request

URI: <https://www.vetenvoy.org/Claim/{id}/Acknowledge>

Method: POST

Response

200 OK

This is a final step in the insurance claim workflow process (i.e. the claim by this point is now finished or closed).

14. Appendix B: Code Examples

The following code samples show generally how to consume the VetEnvoy services via calls to the methods exposed through the VetEnvoy API. All examples are written in the Microsoft C# programming language in the Visual Studio 2005 environment.

14.1 Polling VetEnvoy

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Net;
using System.IO;

namespace VetEnvoy.Samples
{
    public class CodeExample
    {
        public static bool Poll(Guid userId, Guid userPassword, Guid vendorPassword, string pollRef, out string xml)
        {
            try
            {
                // Create the web request
                HttpWebRequest request = WebRequest.Create("https://www.vetenvoy.org/Poll/" + pollRef) as HttpWebRequest;
                request.Method = "GET";

                // Add authentication to request
                request.Headers.Add("UserId", userId.ToString());
                request.Headers.Add("UserPassword", userPassword.ToString());
                request.Headers.Add("VendorPassword", vendorPassword.ToString());
            }
        }
    }
}
```

```

        // Set the content length
        request.ContentLength = 0;

        // Get response
        using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
        {
            using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
            {
                xml = reader.ReadToEnd();
            }
            return true;
        }
        // We don't have an http 200
        catch (WebException ex)
        {
            xml = "";
            return false;
        }
        // Catch other errors
        catch (Exception ex)
        {
            xml = "";
            return false;
        }
    }
}

```

14.2 Uploading XML to VetEnvoy

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Net;
using System.IO;

namespace VetEnvoy.Samples
{
    public class CodeExample
    {
        public static bool Upload(Guid userId, Guid userPassword, Guid vendorPassword, Guid conversationId, Guid recipientId,
            string xmlToSend, out string xml)
        {
            try
            {
                // Create the web request
                HttpWebRequest request = WebRequest.Create("https://www.vetenvoy.org/Claim/" + conversationId.ToString())
                    as HttpWebRequest;
                request.Method = "POST";

                // Add authentication to request
                request.Headers.Add("UserId", userId.ToString());
                request.Headers.Add("UserPassword", userPassword.ToString());
                request.Headers.Add("VendorPassword", vendorPassword.ToString());

                // Add recipient id as sending xml to another user via VetEnvoy
                request.Headers.Add("RecipientId", recipientId.ToString());

                // Create a byte array of the data we want to send
                byte[] byteData = UTF8Encoding.UTF8.GetBytes(xmlToSend);

                // Set the content length
                request.ContentLength = byteData.Length;
            }
        }
    }
}
```

```

        // Set the content type
        request.ContentType = "text/xml; charset=utf-8";

        // Write data
        using (Stream postStream = request.GetRequestStream())
        {
            postStream.Write(byteData, 0, byteData.Length);
        }

        // Get response
        using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
        {
            using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
            {
                xml = reader.ReadToEnd();
            }
            return true;
        }
        // We don't have an http 200
        catch (WebException ex)
        {
            xml = "";
            return false;
        }
        // Catch other errors
        catch (Exception ex)
        {
            xml = "";
            return false;
        }
    }
}

```

14.3 Downloading XML from VetEnvoy

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Net;
using System.IO;

namespace VetEnvoy.Samples
{
    public class CodeExample
    {
        public static bool Download(Guid userId, Guid userPassword, Guid vendorPassword, Guid conversationId, out string xml)
        {
            try
            {
                // Create the web request
                HttpWebRequest request = WebRequest.Create("https://www.vetenvoy.org/Claim/" + conversationId.ToString())
                    as HttpWebRequest;
                request.Method = "GET";

                // Add authentication to request
                request.Headers.Add("UserId", userId.ToString());
                request.Headers.Add("UserPassword", userPassword.ToString());
                request.Headers.Add("VendorPassword", vendorPassword.ToString());

                // Set the content length
                request.ContentLength = 0;

                // Get response
                using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
                {
                    using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
                    {
                        xml = reader.ReadToEnd();
                    }
                }
            }
        }
    }
}
```



```

        return true;
    }
    // We don't have an http 200
    catch (WebException ex)
    {
        xml = "";
        return false;
    }
    // Catch other errors
    catch (Exception ex)
    {
        xml = "";
        return false;
    }
}
}
}

```

14.4 Uploading an Attachment to VetEnvoy

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Net;
using System.IO;

namespace VetEnvoy.Samples
{
    public class CodeExample
    {
        public static bool UploadAttachment(Guid userId, Guid userPassword, Guid vendorPassword, Guid conversationId,
            Guid attachmentId, Guid recipientId, string filePath, out string xml)
        {
            try
            {
                // Create the web request
                string uri = "https://www.vetenvoy.org/Claim/" + conversationId.ToString() + "/Attachment/" +
                    attachmentId.ToString();
                HttpWebRequest request = WebRequest.Create(uri) as HttpWebRequest;
                request.Method = "POST";

                // Add authentication to request
                request.Headers.Add("UserId", userId.ToString());
                request.Headers.Add("UserPassword", userPassword.ToString());
                request.Headers.Add("VendorPassword", vendorPassword.ToString());

                // Add recipient id as sending data to another user via VetEnvoy
                request.Headers.Add("RecipientId", recipientId.ToString());

                // Create a byte array of the data we want to send
                byte[] byteData = File.ReadAllBytes(filePath);

                // Set the content length
                request.ContentLength = byteData.Length;
            }
        }
    }
}
```

```

        // Set the content type
        request.ContentType = "application/pdf";

        // Write data
        using (Stream postStream = request.GetRequestStream())
        {
            postStream.Write(byteData, 0, byteData.Length);
        }

        // Get response
        using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
        {
            using (StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.UTF8))
            {
                xml = reader.ReadToEnd();
            }
            return true;
        }
        // We don't have an http 200
        catch (WebException ex)
        {
            xml = "";
            return false;
        }
        // Catch other errors
        catch (Exception ex)
        {
            xml = "";
            return false;
        }
    }
}

```

14.5 Downloading an Attachment from VetEnvoy

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Net;
using System.IO;

namespace VetEnvoy.Samples
{
    public class CodeExample
    {
        public static bool DownloadAttachment(Guid userId, Guid userPassword, Guid vendorPassword, Guid conversationId,
            Guid attachmentId, out string filePath)
        {
            try
            {
                // Create the web request
                string uri = "https://www.vetenvoy.org/Claim/" + conversationId.ToString() + "/Attachment/" +
                    attachmentId.ToString();
                HttpWebRequest request = WebRequest.Create(uri) as HttpWebRequest;
                request.Method = "GET";

                // Add authentication to request
                request.Headers.Add("UserId", userId.ToString());
                request.Headers.Add("UserPassword", userPassword.ToString());
                request.Headers.Add("VendorPassword", vendorPassword.ToString());

                // Set the content length
                request.ContentLength = 0;

                filePath = "C:\\\" + attachmentId.ToString();
            }
        }
    }
}
```

```

// Get response
using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
{
    using (Stream responseStream = response.GetResponseStream())
    {
        // Read 4k at a time and write to disk
        byte[] buffer = new byte[4096];

        using (FileStream file = new FileStream(filePath, FileMode.Create))
        {
            int bytesRead;
            while ((bytesRead = responseStream.Read(buffer, 0, buffer.Length)) > 0)
            {
                file.Write(buffer, 0, bytesRead);
            }
        }
    }
    return true;
}
// We don't have an http 200
catch (WebException ex)
{
    filePath = "";
    return false;
}
// Catch other errors
catch (Exception ex)
{
    filePath = "";
    return false;
}
}
}

```

15. Appendix C: VetEnvoy Error Codes

15.1 Bad Request (400)

This is obtained in the following circumstances:

- The GUID(s) specified in the URI are not valid for the service being called.
- The request must contain some content.
- This action cannot be performed at this stage in the conversation.
- The URI, method or recipient specified is not valid for the particular user at this stage in the conversation.
- The XML uploaded is badly formed. Please check its syntax before re-sending it.
- The RecipientId specified is invalid.
- The RecipientId specified is not valid for this conversation.
- The request header Content-Type is required when uploading data to VetEnvoy.
- The maximum upload size for VetEnvoy is currently {x} KB.

15.2 Unauthorised (401)

This is obtained in the following circumstances:

- UserId or Password is missing
- UserId or Password is not a valid GUID
- VendorPassword is not a valid GUID
- UserId is incorrect
- VendorPassword is required and has not been supplied
- VendorPassword is invalid
- Password is invalid
- Authentication failed.

15.3 Not found (404)

This is obtained in the following circumstances:

- The service requested is not valid
- The conversation ID (i.e. GUID) as specified is not recognised by VetEnvoy
- The XML requested is not available
- The PDF requested is not available.

15.4 Conflict (409)

This may occur when two users attempt to “speak” on the same conversation at exactly the very same time.

15.5 Internal server error (500)

Miscellaneous (non-specific) server error.

15.6 Service Unavailable (503)

This may be obtained if a service is offline for scheduled maintenance.

16. Appendix E: Transforming the Lab Report XML

The VetEnvoy API optionally allows the custom header “**SchemaVersion**” to be specified when requesting the download of the Lab Report results (see section [11.2]). When this header is provided, VetEnvoy will first attempt to transform the report to the version of the Vet-XML schema as specified in the header prior to delivering the report.

The process works as follows:

1. VetEnvoy looks for a “**version**” attribute (i.e. `version="1.0"` or `version="1.2"`) in the opening tag of the XML (lab or lab batch) as sent by the lab company.
2. VetEnvoy also looks to see if the custom header “**SchemaVersion**” has been sent by the PMS during the call for downloading the report.
3. If the above two values are both present and of the permitted values, VetEnvoy will do the transformation. Please note, if either the version attribute or the custom header is missing, or if either of the values supplied is not permitted, VetEnvoy will simply ignore them and return the XML in its original state (i.e. it will not transform the report).

Currently the supported values are 1.0, 1.2 and 1.3. VetEnvoy supports converting upwards from 1.0 to 1.2 or 1.0 to 1.3 and downwards from 1.3 to 1.2 or 1.2 to 1.0.

17. Appendix F: Nullifying a Conversation

The technique to nullify a conversation is the same with each of the VetEnvoy services. The steps involved are shown over the following pages.

In essence, when a conversation is nullified, this means that the party requesting the nullification (e.g. the PMS in relation to insurance claims and Product Orders) wants the communication to cease immediately and no further activity to take place on it (i.e. to effectively cancel it). Needing to do this may be necessary, for instance, if an insurance claim was originally sent by mistake, if it is no longer relevant, or if it is withdrawn by the policy holder.

Please note, currently the nullification facility is active only with the Insurance Claim service, Product Ordering service, and Microchip Registration service.

To help test the implementation of the nullification service, the “VetEnvoy Test Tool” can be used to imitate each and any step in the process. See section 19 for details on how to obtain this facility.

17.1 Nullifying

This is undertaken by the Practice Management System (PMS) in relation to insurance claims and Product Ordering. It can be done at any point during an active claim or product order (provided it has not already been settled or declined by the Insurance Company or accepted by the Supplier).

Please note, with regard to the URI provided below on this page and the next:

- “**Service**” should be substituted with the actual name of the relevant service (i.e. “Claim” for insurance claim and “Order” for product ordering).
- “**id**” represents a GUID (see section 4) which is used to uniquely identify the particular insurance claim or order.

Request

URI: <https://www.vetenvoy.org/{Service}/{id}/Nullify>

Method: POST

Response

200 OK

Following this step in the workflow, the next step is for the recipient to tell VetEnvoy that it is aware that the insurance claim or product order has been nullified (see section 17.2).

17.2 Acknowledging the Nullification

This is undertaken by the Insurance Company in relation to insurance claims and by the Supplier with respect to Product Ordering.

As a final step in the insurance claim or product ordering workflow process, its purpose is to provide closure of the claim or product order (and subsequent archiving in VetEnvoy). In effect, through this step the intended recipient can tell VetEnvoy that they are aware that the intended sender has nullified the claim or order (see section 17.1) and that no further communication is required with regards to it.

NB: Once this step is completed successfully, no further communication is possible between the parties with regard to the claim or order. That is, the claim or order will be treated as closed by VetEnvoy.

Request

URI: <https://www.vetenvoy.org/{Service}/{id}/Acknowledge>

Method: POST

Response

200 OK

This is a final step in the workflow process (i.e. the claim or order by this point is now finished or closed).

18. Latest Partner Status

Please visit our website (www.vetenvoy.com) for latest information on which Partners and Suppliers are supporting VetEnvoy and have implemented it.

19. Appendix D: VetEnvoy Test Tool

The “VetEnvoy Test Tool” has been created by LiveTime 24/7 Ltd for use by PMS Software Vendors and other service providers (e.g. Insurance Companies, Lab Companies and Microchip Registration Database Providers) to facilitate their testing of the implementation of the VetEnvoy API.

To request a copy of this tool, please contact Mike Fletcher at LiveTime 24/7 Ltd on mike.fletcher@livetime247.co.uk or 07989 940416.