QUANDIS

Business solutions powered by qbo

Quandis Military Search Integration Guide

Quandis Data Services

June 2016

v 1.0.3

Notice

Quandis has used its best effort in preparing this document. Quandis makes no representations, or warranties with respect to the accuracy or completeness of this document and specifically disclaims any implied warranties or merchantability or fitness for a particular purpose. Quandis shall have no liability to any party wherever located because of any use or attempted use of this document

Table of Contents

QMS Integration Guide

Revision History

Date	Version	Description
2015-10-22	1.0.0	Created.
2015-10-27	1.0.1	Product Notification updates
2015-11-11	1.0.2	Added Parsing and Permutation Logic
2016-06-21	1.03	Endpoint Update

Overview

This document serves as a guide for automating the integration of Quandis Military Search (QMS) using the Quandis Data Services Platform. It describes transmission protocols utilized, the QMS Military workflow, Xml messaging and schema requirements. It will also outline general usage and common error types.

It is written for both prospective and new system integrators, consultants and other technical professionals interested in integrating with QMS.

Further information about QMS, including latest information, schemas, registration and the software development kit is all available via the web. Please contact services.support@quandis.com or the Quandis web site (available at http://www.quandis.com/).

Definitions

The following terms are referred to in this document.

Term	Definition
Client, Clients	Organizations requesting military search orders
Endpoint	Web service or HTTP receiver that is exposed by an organization to receive inbound data
QBO	Acronym for Quandis Business Objects. Generally refers to any module or component that has been created by Quandis.

Data Transmission

QMS sends and receives data via the public internet using HTTPS (2048 bit) protocol as the standard protocol. VPN connectivity is supported under special circumstances. QMS supports the following protocols:

- SOAP (using HTTPS)
- HTTP POST (RESTful)
- sFTP

Preferred Transmission Patterns

QMS supports an event driven transmission model whereas the preference is for the originating system to push or invoke the target end point whenever there is a need to transmit data. This is opposite from the source system polling the target system endpoint to check if data is available at any given time. As a result, we require our partners to expose an end point such as a SOAP web service or HTTPS receiver. The end point is generally public accessible with security controlled by IP restriction and BASIC authentication.

Service Connectors

QMS supports the above protocols as standard connectivity. In some scenarios, there is a requirement to connect to an existing web service or platform using a combination of protocols to satisfy non-standard connectivity requirements for a partner. QMS can also implement Service Connectors that will facilitate this. The usage of Service Connectors is evaluated on a per implementation basis.

Additional Security Models

The following table describes the various protocols and security models used by QMS:

Protocol	Security Model	Notes
HTTPS (SOAP,REST)	BASIC Authentication using HTTPS. See: http://www.quandis.com/CodeSample.DCSWebService.aspx http://www.faqs.org/rfcs/rfc2617.html	Primary security model used by QMS
sFTP	No additional security	

FTP Exchange Model

The FTP file exchange model uses secure FTP to exchange data. Due to the nature of the FTP file exchange, integrations are addressed on an individual basis. Please refer to Appendix B for common implementation models.

QMS Environments

The following table outlines the location of UAT and Production host values. Unless otherwise stated, all URL relative paths are the same for production with only the host value changed.

QMS Environment	Host	Notes
UAT	https://uatscra.quandis.net/	All references in user guide point to UAT
Production	https://scra.quandis.com/	

Endpoints

During the lifecycle of a military search several events occur on the client and Quandis systems. A subset of these events requires communication of data between the various systems to facilitate order processing.

QMS Specific Endpoints

The following endpoints are exposed by QMS for military search events:

Purpose	QMS Endpoint
Order Placement	https://uatscra.quandis.net/Military/Military.ashx/Search
Order Pickup (Polling)	https://uatscra.quandis.net/Military/Military.ashx /Pop
Product	https://uatscra.quandis.net/Military/Military.ashx /Product
Request Validation	https://uatscra.quandis.net/Military/Military.ashx /Validate

Client Specific Endpoints

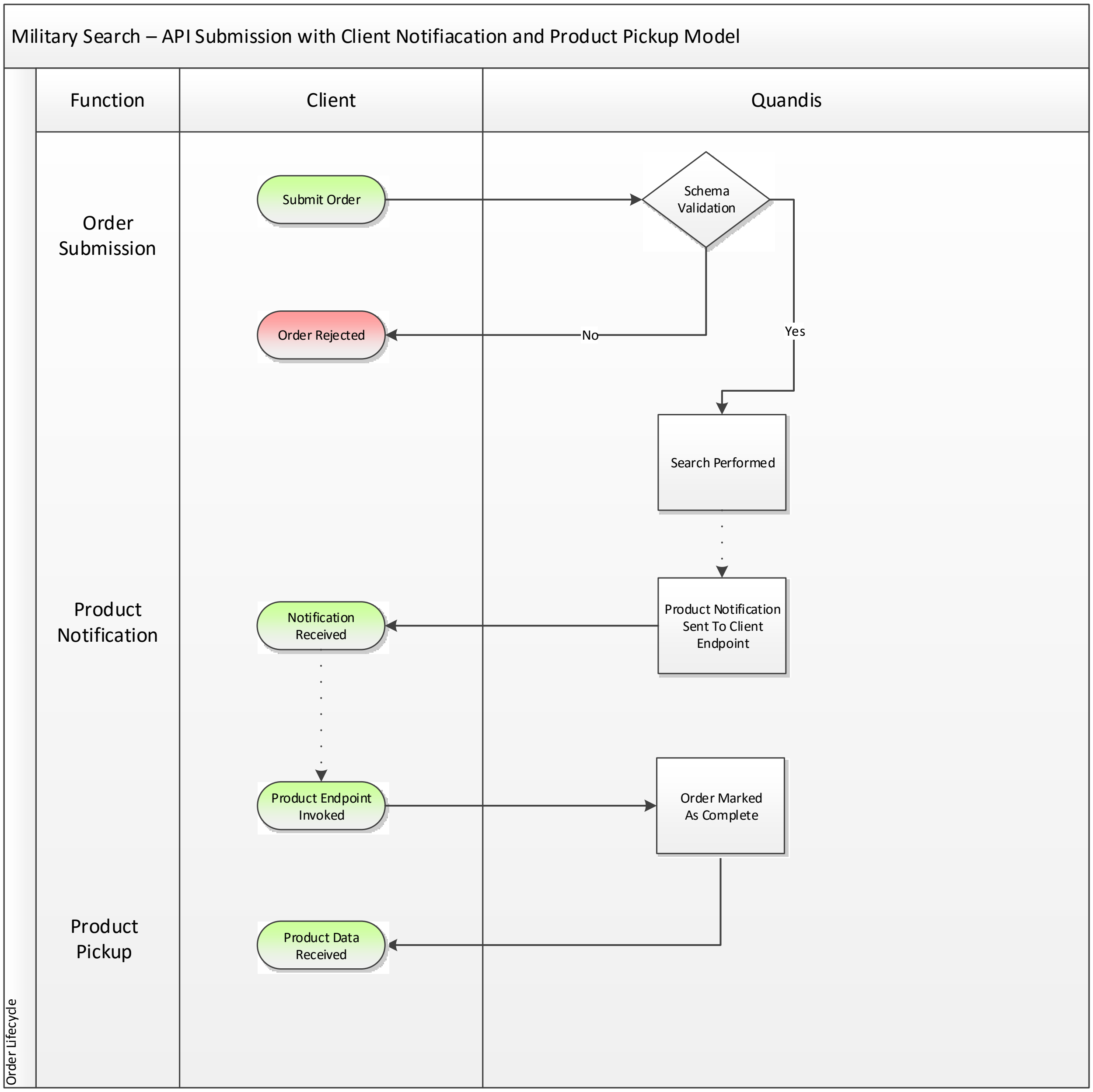
The following table outlines the client endpoints utilized by QMS for military search events:

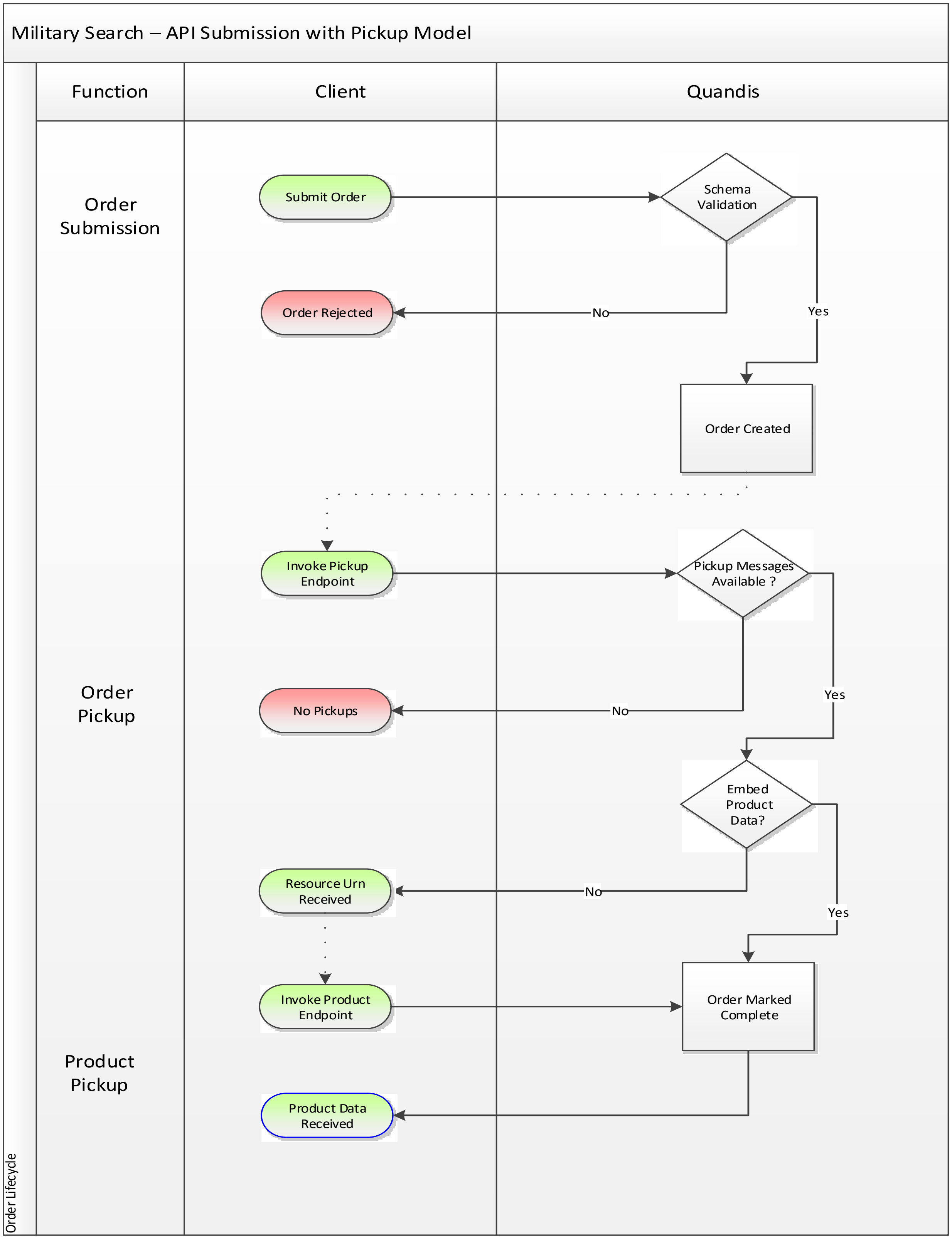
QMS Event	Client Endpoint
Product Data (Non Pickup)	Client Receives Data. Only used for non-pickup scenarios when QMS posts military response data into the client system Client must provide endpoint that utilizes the supported protocols.

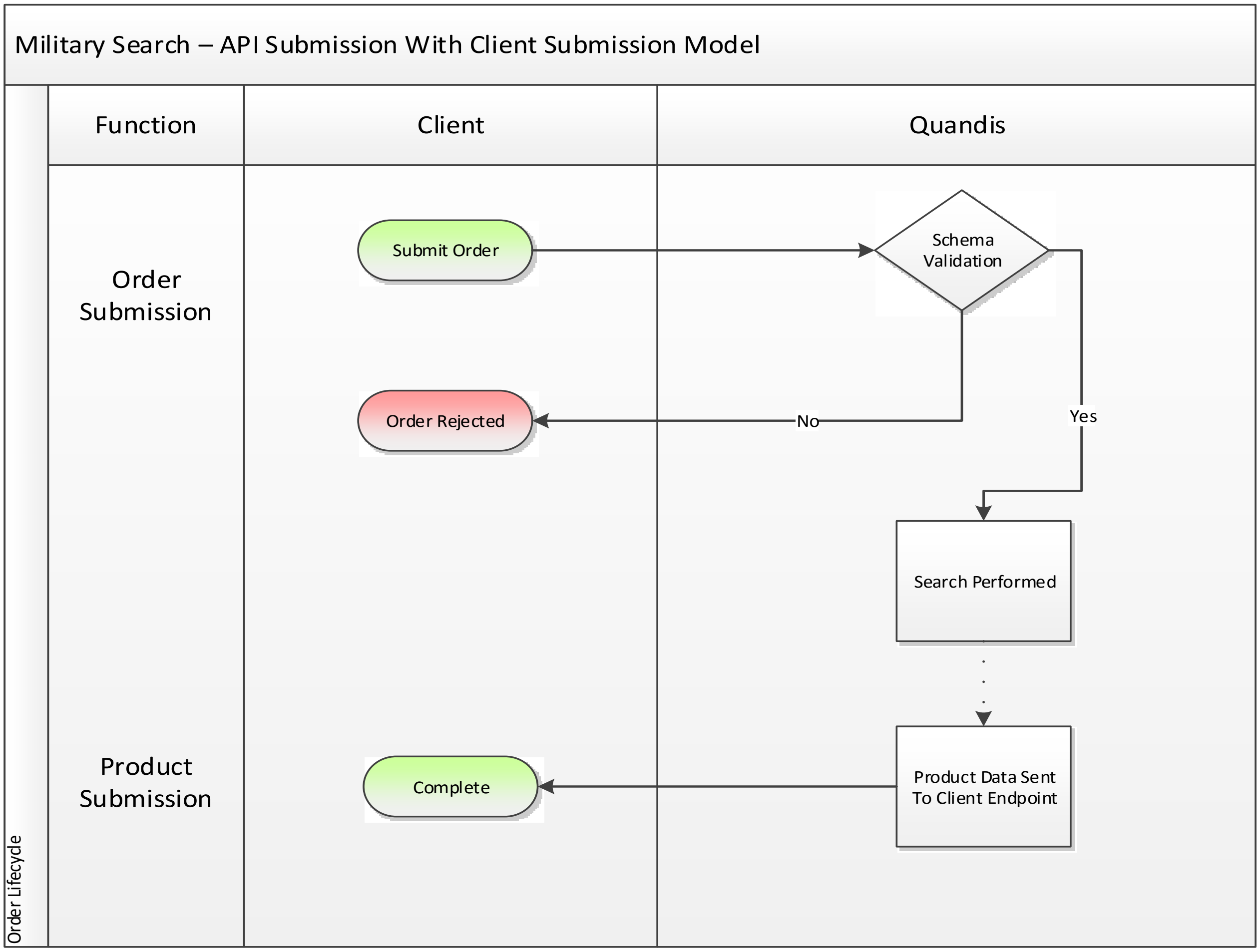
Military Search Workflow

Workflow Diagram

The following diagrams depict the various QMS process follows. For each arrow, there is a corresponding Xml message that is sent from party to party. The solid lines represent required messages.




Military Search – API Submission with Pickup Model
Function
Client
Quandis
Order Submission
Submit Order
Schema Validation
Order Rejected
No
Yes
Order Created
Order Pickup
Invoke Pickup Endpoint
Pickup Messages Available ?
No Pickups
No
Yes
Embed Product Data?
Resource Urn Received
No
Yes
Product Pickup
Invoke Product Endpoint
Order Marked Complete
Product Data Received
Order Lifecycle



Xml Message Types for Military Search Workflow Steps

The following table describes the various Xml message types used for military search and what stage in the workflow they are utilized:

Workflow Step	Participant(s)	Schema(s) Utilized	Triggering	Endpoint(s) Utilized
Order Placement	Client	Active Duty Search.xsd		QMS Endpoint
Synchronous Acknowledgement	Client	QMS.Acknowledgement.xsd	Synchronous acknowledgement returned during order placement with QMS	N/A

Product Data / Product Notification	QMS	ActiveDuty.Response.Pickup.xsd	Order processed	QMS Endpoint
Order Pickup	Client	QMS Pickup.xsd	By client on schedule	QMS Pickup

Key Military Search Workflow Steps

The military search process contains key workflow steps or events. The following provides additional detail on these steps.

Order Submission

Order submission originates from the client system. Orders are submitted by using any of the supported protocols. QMS will evaluate the format of the message using schema validation and instantly return a success or failure in the form of a QMS acknowledgment. In the diagram above, this is represented as a separate message but is part of order submission

Product Data

When the order has been processed on QMS, product data will be submitted to the client endpoint or remain ready for pickup. There are variants on how product can be submitted to the client:

Client Submission

Under the client submission model, QMS will deliver the product data to the client's configured endpoint.

Client Notification / Product Pickup

QMS has the ability to notify the client product is ready for consumption and the client must invoke the product endpoint to obtain product data. Upon order completion, QMS will invoke the client endpoint submitting a Product Notification. The Product Notification contains a ResourceUrn binding to the product endpoint for subsequent product download. The client reserves the right to download the resource on their schedule. The ResourceUrn contains a reference to the Product endpoint and will return the Product Data format. This is the preferred approach for processing orders that contain multiple searches.

Order Pickup

QMS exposes a stand-alone endpoint which allows clients to poll for completed orders. Product data is retrieved by invoking the endpoint. Upon invoking the pickup endpoint, each complete order is represented as a message in the pickup response and marked as received. Once the pickup list includes a given order, it is automatically removed from the list and will no longer be present in subsequent calls. Users have the option to embed product data in the pickup response or obtain a Resource Urn to access the product data at a later time.

Pickup Model Options

The pickup endpoint accepts the following optional parameters:

Parameter	Description
MessageCount	Allows for multiple messages to be downloaded at once. Allowed values: 1-5. Default value is 1 which implies one message will be returned when the pickup endpoint is invoked.
PayloadType	Embeds product content or product pointer in pickup message. Enumerated: Content, ResourceUrn. Default is Content. See below for best practices.

Pickup Model Best Practices

The frequency is which to invoke the pickup method should be set by a schedule in accordance with given SLA requirements. Since the pickup method is a polling method, it will return a message list with a count ranging from zero to the number of available messages. The following table is suggests best practices for invoking the pickup endpoint:

Message Count Returned	Frequency
0	If no messages are available, use the next run date on the predefined schedule performing the polling.
Greater than 0	If messages are available, invoke the pickup method as soon as the previous call completes and repeat until no messages are available to ensure all messages are retrieved.

The pickup model response allows for embedding of product data as a convenience or emitting a ResourceUrn which allows product to be accessed in a subsequent call. PayloadType usage guidelines are:

Payload Type	Recommended Usage
Content (Embedded Product)	Smaller payloads, smaller volume and when content is NOT processed during the pickup call. We recommend saving a copy of the content and processing at a later time.
ResourceUrn (Product Pointer)	Larger payloads (due to multiple searches), larger volume with ability to use separate process to invoke the Product endpoint to download and process product a separate process. Also note the product can be

	obtained subsequent times by accessing ResourceUrn without invoking pickup method.

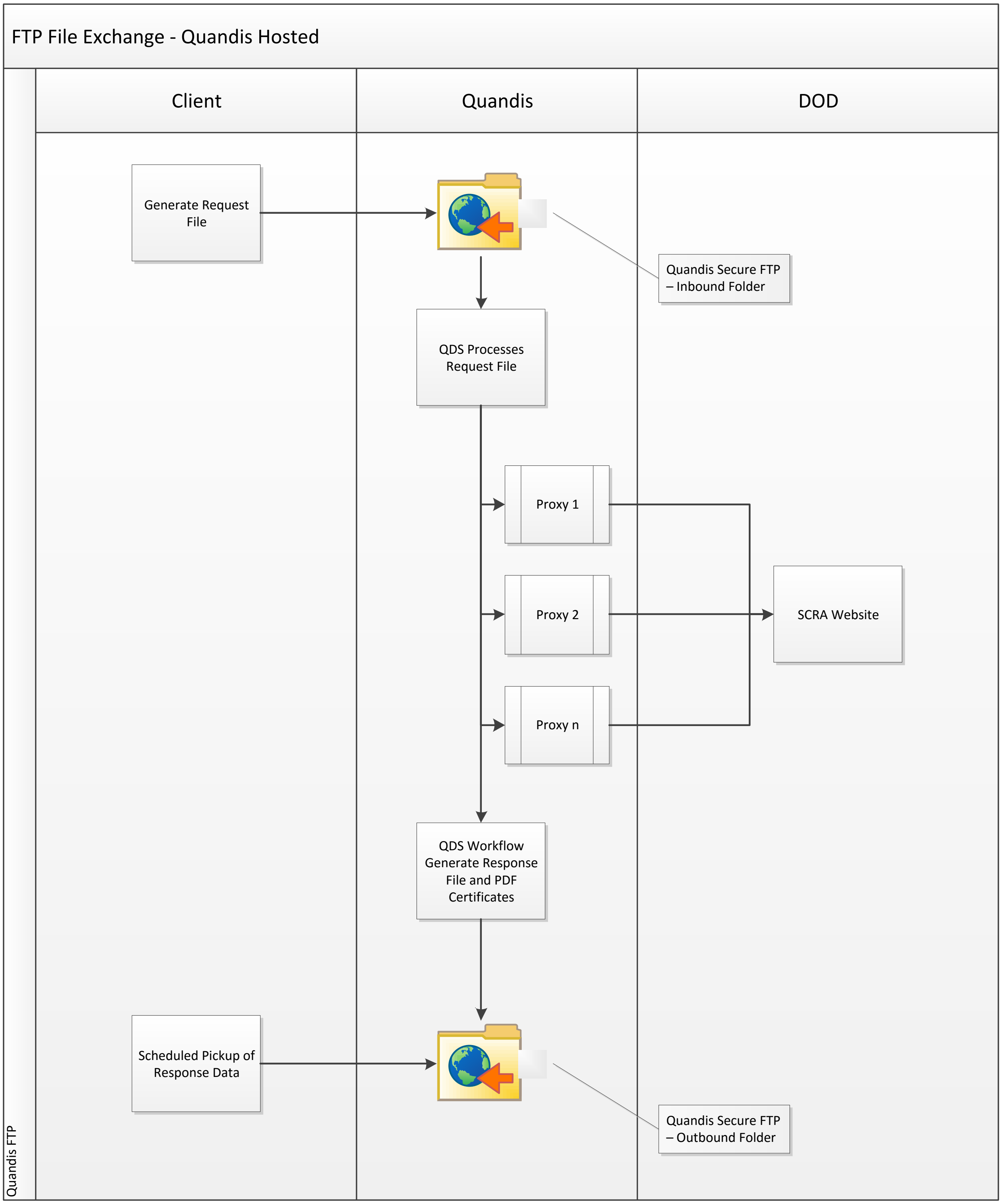
Practical Example: We suggest not to set MesageCount to *5* with PayloadType to *Content* and process data (saving file content, saving data to DBMS) during the pickup call ***in favor of*** MessageCount to *5* with PayloadType to *ResourceUrn* with separate calls to the product resource.

Appendix A: FTP Exchange Models

Military search data can also be exchanged using secure FTP. In most cases, each implementation is slightly different. The following examples are the most common integration scenarios.

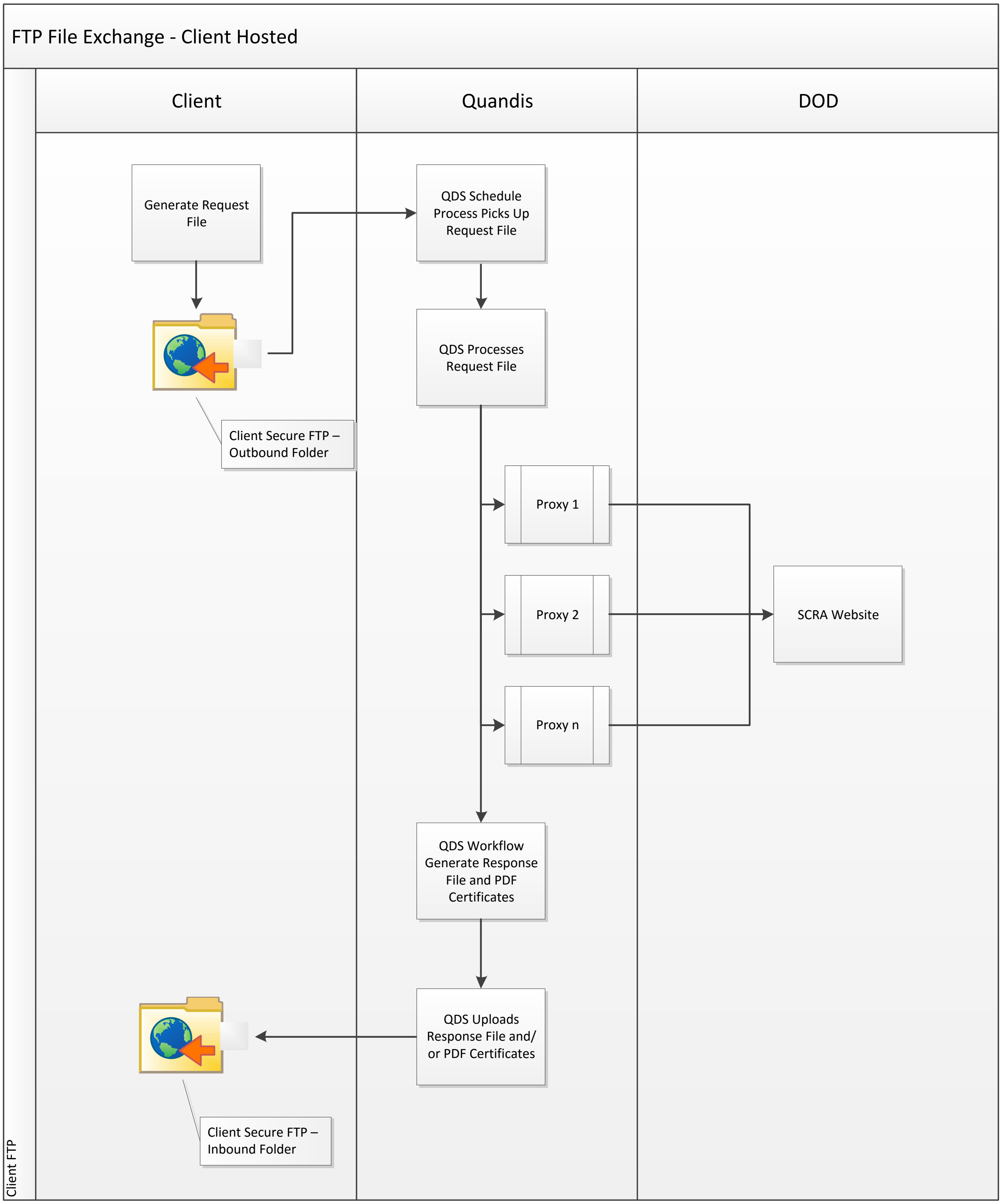
Quandis Hosted FTP

The following model outlines the FTP file exchange process where Quandis hosts the FTP server.


FTP File Exchange - Quandis Hosted
Quandis FTP
Client
Quandis
DOD
Generate Request File
Quandis Secure FTP – Inbound Folder
QDS Processes Request File
Proxy 1
Proxy 2
Proxy n
SCRA Website
QDS Workflow Generate Response File and PDF Certificates
Scheduled Pickup of Response Data
Quandis Secure FTP – Outbound Folder

Client Hosted FTP

The following model outlines the FTP file exchange process where the client hosts the FTP server.


FTP File Exchange - Client Hosted
Client
Quandis
DOD
Client FTP
Generate Request File
QDS Schedule Process Picks Up Request File
QDS Processes Request File
Client Secure FTP – Outbound Folder
Proxy 1
Proxy 2
Proxy n
SCRA Website
QDS Workflow Generate Response File and PDF Certificates
QDS Uploads Response File and/ or PDF Certificates
Client Secure FTP – Inbound Folder

Appendix B: Parsing and Permutation Logic

QMS currently supports two permutation models: Simple and Advanced. With each model, subject data is parsed and prepared before permutation logic is applied. The following tables outline each models' respective parsing and permutation features.

The following definitions are applicable to name parsing and permutations:

Term	Definition
Name Part	Individual word which is part of name. For example last name *Jones Smith* contains two name parts: 1) *Jones*, 2) *Smith*
Name Prefix	Prefix to a name that It may signify either veneration, an official position or a professional or academic qualification. For example the name *Mrs. Jane Doe* the prefix is *Mrs*.
Name Suffix	Suffix following last name which provides more information about the subject. For example, last name *Jones, Jr* contains suffix *Jr*
Name Segment	Quandis term used to identify a name embedded in a name part. For example the last name *Jane Doe aka John Doe* contains two name segments

Simple Model

Parsing features apply only to last name of subject.

Parse Features

Parse Feature	Description
Known Word Removal	Removes known string patterns from last name such as *Estate, AKA, deceased, etc.* See Standard Known Words under Simple Permutation Enumerations. Additional words can be added to removal list.
Suffix Removal	Removes known names suffixes from last name. See Name Suffixes under Simple Permutation Enumerations.
Space Normalization	Reduces multiple spacing between name parts to a single space. Removes any leading and trailing spaces next to hyphens.
Word Scrubbing	Removes all non-word characters from last name including numbers and any type of whitespace with the exception of Non Removed Characters. See Simple Permutation Enumerations.

Permutation Features

Permutation Feature	Description
Force Hyphen	Generates additional name variations only on two part last name delimited by a space replacing the space with a hyphen. For example last name *Smith Jones* would also generate a *Smith-Jones* permutation.
Include Original	Includes original name provided along with any name variations generated.
Transpose	Transposes name parts to generate all unique variations. For example transposing *Doe Smith* would generate *Smith Doe*. Transposing occurs on any number of name parts provided.
Name Limit Count	Limits the amount of names that are permutated. For example if a name contains 3 parts and the limit is set to two parts, the right 2 names will be used as the permutation basis.
Toggle Middle Name	Ability to generated name variations with and without middle name when middle name is provided. Middle name toggle generates variations after transposing of name parts.

Simple Permutation Enumerations

Enumerated Item	Description
Name Suffixes	jr, sr, i, ii, iii, iv, v, vi, vii, viii, ix, x, xi, xii, xiii, xiv, xv
Standard Known Words	aka, a/k/a, nka, n/k/a,fka ,f/k/a ,(deceased),deceased
Non Removed Characters	Space, hyphen, apostrophe

Advanced Model

Parsing features apply to first, middle and last names of the subject. Segment parsing is only applicable to last name of subject.

Parse Features

Parse Feature	Description
Known Word Removal	Removes known string patterns from last name such as *Estate, AKA, deceased, etc.* See Standard Known Words under Advanced Permutation Enumerations. Additional words can be added to removal list.
Prefix Removal	Removes known names prefixes from first name. See Name Prefixes under Advanced Permutation Enumerations.
Suffix Removal	Removes known names suffixes from last name. See Name Suffixes under Advanced Permutation Enumerations.
Space Normalization	Reduces multiple spacing between name parts to a single space. Removes any leading and trailing spaces next to hyphens.
Word Scrubbing	Removes and/or substitutes all non-word characters from names including numbers and any type of whitespace with the exception of Non Removed Characters. See Advanced Permutation Enumerations.
Segment Parsing	Splits last name based on presence of Segment Delimiter and creates name aliases. Used when multiple full names are passed in last name. For example, last name *Smith AKA John A. Doe Jr* will generate two name segments: 1) last name *Smith* 2) *John A. Doe Jr* where the left most name is the primary segment and the subsequent names are aliases. Segments are evaluated to ensure that no duplicate alias are produced by redundant segments.

Permutation Features

Permutation Feature	Description
Force Hyphen	Generates additional name variations only on two part last name delimited by a space replacing the space with a hyphen. For example last name *Smith Jones* would also generate a *Smith-Jones* permutation.
Include Original	Includes original name provided along with any name variations generated. When segment parsing is used and the last name generates more than 1 segment the original last name is no longer used in searching.
Transpose	Transposes name parts to generate all unique variations. For example transposing *Doe Smith* would generate *Smith Doe*. Transposing occurs on any number of name parts provided. This feature can be disabled on a per order basis.
Name Limit Count	Limits the amount of names that are permutated. For example if a name contains 3 parts and the limit is set to two parts, the right 2 names will be used as the permutation basis.

Toggle Middle Name	Ability to generated name variations with and without middle name when middle name is provided. Middle name toggle generates variations after transposing of name parts.
Mixed Case Split	Detects and splits names based on letter case to generate multiple name parts. For example, last name *MacDonald* would generate *Mac* and *Donald*.

Advanced Permutation Enumerations

Enumerated Item	**Description**
Name Prefixes	ms, miss, mrs, mr, master, rev, fr, dr, atty, prof, hon, pres, gov, coach, ofc, msgr, sr, br, supt, rep, sen, amb, treas, sec, pvt, cpl, sgt, adm, maj, capt, cmdr, lt, col, gen
Name Suffixes	jr, sr, i, ii, iii, iv, v, vi, vii, viii, ix, x, xi, xii, xiii, xiv, xv
Standard Known Words	deceased,spouse,of,unknown,trust,trustee,individually,and,as,the,personal,representative,estate,life,tenant
Segment Delimiters	aka , a/k/a, a/ka/, fka , f/k/a, nka , n/k/a, of the estate of
Non Removed Characters	Space, hyphen, apostrophe.
Character Substitution	Tick (`) is replaced by apostrophe. For example O`Neill is modified to O'Neill.

Appendix C: Frequently Asked Questions

Q: How can QMS prove they submitted our exact data to the DOD?

A: Some clients request a "screen shot" of the data being submitted to the SCRA website as "proof of submission". There are two problems with this request:

1. Such a screen shot does not constitute proof (i.e. it can be easily fabricated)
2. The "screen shot" is required for human users, but is never actually used by automation

Why a screen shot does not constitute proof

One can easily bring up the DoD website, key in data, and print the page. This does not mean that the user actually clicks on the "Lookup" button. As a result, this "proof" can be easily fabricated.

Why the screen shot is never actually used by automation

"Web pages" are HTML rendered by a browser generally geared for human eyes. It does nice things like prompt people to enter the correct data fields, and provide a Lookup button to tell the browser to transmit data back to the SCRA servers. When a user clicks on the "Lookup" button, the user's browser **just transmits the data fields** to the SCRA servers.

Quandis' technology **just transmits the data fields** to the SCRA servers, without ever requesting the HTML page that prompts users to enter data fields. Not only is such a page useless to the automation software, requesting the HTML from the SCRA website with every search would actually slow the search down.

Solution

Proof of submission means proof that the **SCRA website actually received the data** Quandis claims to have submitted. The DoD recognizes the need to provide an **audit trail** in case questions arise about the validity of data in the future. To solve this, DoD provides a ReportID, which is a unique identifier that can be used to reconcile with the DoD at a future date. The ReportID cannot be faked, and is proof both that Quandis submitted the data, and that the SCRA website received the data.

Should a court require that a bank "prove" the DoD listed a service member as active (or not), one must simply contact the DoD with the ReportID, and the DoD can provide the data that was submitted. This is the only fool-proof audit trail.

Q: We have our own xml formats we use. Can QMS adopt these xml formats?

A: Yes. QMS can transform our native QBO formatted data into your organization's format. This is extra effort and will impact the implementation timeline.

Q: We already expose various end points to accept data. Can QMS tap into these endpoints?

A: Yes. However there are two considerations: 1) if the current endpoint is not supported by QMS standard protocols then a custom Service Connector will need to be built. 2) Your current endpoint may require a certain xml format to correctly function. When QMS consumes a custom endpoint for a Service Connector, we generally transform the data from the QBO format to the required endpoint format. This is extra effort and will impact the implementation timeline.

Q: Is data delivered to Quandis in a secure manner?

A: Yes, clients typically transmit batch orders as Excel files. The Excel files may be encrypted with PKI and/or transmitted securely via sFTP, FTPS, or HTTPS. Once on the Quandis network, the request files are processed and individual borrower names and SSNs are stored in a secure SQL Server (with SSNs being symmetrically encrypted at rest). Data is transmitted between Quandis and the SCRA website over HTTPS.

Q: Are results received from Quandis in a secure manner?

A: Yes, result files are built on the Quandis platform, and may be encrypted with PKI and/or transmitted securely via sFTP, FTPS or HTTPS to the client. We can email clients with a secure hyperlink to download the result file. Each search includes a PDF document containing the image of the SCRA website results. Neither this image nor the aggregate results file contain SSNs, though they usually contain borrower information and loan numbers, so continue to be treated securely.

Q: How do I know Quandis employees will not compromise my data?

A: Military search is a completely automated process; Quandis employees are not involved with the mechanics of a military search. Quandis system administrators have access to the servers on which the data is processed, and are governed by our Security Policy and Procedures document.

Q: Is the data that you process secure?

A: The request and result files are encrypted and/or transmitted via secure channels (sFTP, FTPS, or HTTPS). All data within Quandis for processing is stored in a secure SQL Server database, with SSNs being encrypted at rest. SQL backups are encrypted, and stored on RAID media in our alternate production data center. RAID media is physically destroyed upon disposal. We do not use tape backups or off-site storage of our backups. All access to RAID media is tightly controlled with a minimum of two-factor authentication, and employees with access are trained in accordance with our Security Policy and Procedures document.

Q: How long do you retain the data?

A: We retain all data for 6 months by default. However, we can configure on a client-by-client basis different retention periods to comply with client retention policies. We recommend 6 months ensuring we can answer any questions that may arise concerning your data. The monitoring product requires that we retain the data for at least the duration of monitoring.

Q: Is your system scalable? How many searches can you perform?

A: Quandis maintains two production SAS-70 type II certified data centers: one in CA and one in VA. Our Military Search primary hosting location in our CA data center, with nightly backups transmitted to our VA datacenter. Each runs on load balanced web servers with SQL Server 2008 for the DBMS. The limiting factor in volume is the DOD's SCRA website. We typically process 15K orders per hour. We can auto-scale multiple proxy servers to distribute the requests bound for the SCRA website, ensuring we remain in compliance with their IP-based request thresholds.

Q: What are the various statuses of a search? What do they mean?

A: There are 4 possible statues: Active Duty, Discharged, Not Found and Unable To Process.

- Active Duty - Indicates Active Duty information was located based on SSN or birth date and last name. This is considered a match.
- Discharged - Indicates Active Duty information was located based on SSN or birth date and last name. The contact shows a discharge date.
- Not Found - Indicates SCRA has no information based on SSN or birth date and last name. This is considered a no hit.
- Unable To Process - Indicates an error was encountered when searching SSN or birth date and last name. SCRA generally returns this error when the SSN appears to be in a fictitious pattern such as 999999999 or 111111111. In a very small percentage of cases the SCRA will return this status with what appears to be a valid SSN and last name.
- Undetermined - Multiple individuals located with the same last name which share the same birth date. Generally speaking, this type of match requires further investigation.

Q: What are the various service branches returned by the search?

A: Air Force, Army, Coast Guard, Marine Corps, Navy, NOAA, Public Health Service