AssureID™ Connect
Web Service Integration Guide

Version 2.16.12
October 2020

October 2020

Acuant Inc.
6080 Center Drive
Suite 850
Los Angeles, CA 90045

Contents

AssureID Connect Web Service

The AssureID Connect Web Service provides the following features:

- Captures, classifies, and authenticates identification documents against a database of over 3,400 supported document types
- Stores and retrieves individual documents
- Provides secure communication over SSL/TLS with user-based authentication and authorization
- Supports a wide range of document scanners and imaging devices from purpose-built ID scanners to flatbed scanners and mobile phones
- Enables uploading individual document resources including image files, binary data files and results of contactless chip authentications
- Automatically detects and crops a document from within the uploaded image
- Transfers data via XML or JavaScript Object Notation (JSON)
- Integrates into existing or new applications with the included .NET SDK or the REST API
- Includes the AssureID Connect Service Workbench application, which allows integrators to easily test the service using previously captured data

Using AssureID Connect

This section provides details and examples of how to use the AssureID Connect API to process individual document data capture and authentication transactions. For specifics on using the C# wrapper and REST API interfaces, see the .NET Web SDK and REST API sections in this document. See the AssureID Connect Workflows for more information.

POST document instance

The document instance is created by posting the document instance and obtaining a document *instance identifier*. Rather than posting all document images, only the images that are required to classify the document (for example, identify the type of document) are posted. At a minimum, this is the white light source image of one side of the document, but may also include the white light source image of the reverse side of the document when images of both sides of the document have been captured.

Posting images typically requires the following distinct operations:

- Capture the image
- Compress the image
- Post (upload) the compressed image

These operations are largely independent and are constrained by different resource types, therefore they can be overlapped for significant performance gains. Image capture tends to be constrained by device I/O, image compression is CPU-driven, and the posting of the image is limited by network bandwidth. See Image Requirements and Recommendations for more information.

Note Non-imaging data (such as magnetic stripe, machine readable zone) is very small compared to imaging data; therefore, it can be beneficial to post this information prior to initiating document classification provided that the data is already available. In some cases, this information can be used to speed up the classification process. If the data is not immediately available, it should be posted after classification has been completed.

GET document classification

Getting the document classification will initiate the classification process within the service and return the results of the classification process, including document type, issuer, class, and series. It will also return which additional image types (such as nearinfrared and ultraviolet) are supported to allow for optimization of subsequent captures and transmission so that no images are unnecessarily transmitted.

During classification, the presentation of the document may also be determined. That is, if an image was posted as the front side of the document but is actually the back side of the document, the **PresentationChanged** property will indicate that the document presentation (for example, sides) were reversed during classification. If this occurs, all subsequent image posts must account for this presentation change.

For example, if a white/front image was posted for classification, but it classified as white/back, the remaining white image should be posted as white/front to properly account for the presentation change.

POST document image

All remaining images are then posted to the document instance.

GET image metrics

If requested, image metrics can be calculated on the images. This is recommended for mobile document capture. The following metrics are available:

Metric	Description
GlareMetric	A value between 0 and 100 (a score of 50 and above is considered a pass)
HorizontalResolution	Horizontal resolution of the image
IsCropped	Indicates whether the image is cropped by the web service
IsTampered	Indicates whether the image has been tampered with
SharpnessMetric	A value between 0 and 100 (a score of 50 and above is considered a pass)
VerticalResolution	Vertical resolution of the image

PUT document image

An image can be updated using the document instance created in the POST document instance step. For example, if the image posted in the POST document image step fails the sharpness metric with a score of 40, then you should post a new image in the PUT document image step and then evaluate the image metrics on the updated image.

POST document data

Any non-imaging data that is to be included is posted to the document instance.

POST any non-imaging data

After all data is posted, the client checks and waits for the GET document classification operation to complete. After the operation is complete, the results can be retrieved. This operation will perform any remaining processing, including data capture and authentication that had not yet been performed on the server.

GET corrected image

During classification, AssureID Connect identifies the type of a document as well as its presentation and orientation. When posting images to AssureID Connect, one side of the document must be specified as the front, but this may not actually be the front of the document. If the **PresentationChanged** property has been set, this means that the presentation of the document has been corrected, indicating that the document was presented with the back side specified as the front side and vice versa. Orientation refers to whether the top edge of the document was at the top of the captured image. If the document was rotated to correct the orientation, the **OrientationChanged** property will be set. If either of these two properties are set, additional action may be required by the application to ensure that displayed images or images stored external to AssureID Connect reflect the corrected orientation and/or presentation.

To obtain images with corrected presentation and orientation, an application can retrieve each image from AssureID Connect, using the image URIs provided in the document result. It is also possible to retrieve images using the Web SDK.

Manual document review

Acuant offers an optional service called **Acuant Review** that provides a manual review of a document if the authentication results are not trusted or under high security requirements. This service requires a subscription and the client application must provide the email and password authentication (same as used for any other Acuant service) to access this service.

> **Note** Acuant Review requires that the store PII option be enabled on a subscription to use this service.

The review transaction requires the client application to provide a document **instanceID**, the **subscriptionID,** and the **type** (currently only one **Type** is supported: **1 = Identity Documents**)

Example

```
{
  "SubscriptionId": "7da88b7d-04b7-478b-8979-a07451c24d15",
  "InstanceId": "08e9c696-6d72-4b2e-8ecc-f85aa3ddef22",
  "Type": 1
}
```

The review results will include the **reviewTime** (time to review, in milliseconds) and the **reviewStage** code that identifies the value for the result. See the AssureID Connect transaction API documentation for complete information and values for all codes.

Barcode-only mode

The **Barcode** enumeration was added to support 2D barcode-only read mode. It decodes and parses the barcode from an image of the back of a North American (U.S./Canada) ID. If a 2D barcode string is specified, any images are ignored. If 2D barcode string is *not* specified, AssureID will look first for a Front/White image, then a Back/White image. The first image from which it is able to extract a barcode will be the data processed.

> Note The AssureTec.AssureID.Web.SDK.Document returned by AssureID will include all data fields available in the 2D barcode. AssureID will not classify the document; therefore, the returned classification will be Unknown. No authentication tests will be run.

To process a barcode, use **AssureTec.AssureID.Web.SDK.DocumentProcessMode.Barcode**. To use this mode, callers must post either a parsed 2D barcode data string, or an image that contains a 2D barcode, for example:

```
POST /Document/Instance, DocumentSettings.ProcessMode.Barcode
POST /Document/{instanceId}/Data, DocumentDataType.Barcode2D, string
GET /Document/{instanceId}
(optional) DELETE /Document/{instanceId}
```

or

```
POST /Document/Instance, DocumentSettings.ProcessMode.Barcode
POST /Document/{instanceId}/Image, DocumentSide.Front|Back,
   LightSource.White, image
GET /Document/{instanceId}
(optional) DELETE /Document/{instanceId}
```

Web Service authentication

The AssureID Connect Web Service uses the HTTP Basic authentication scheme as defined in RFC 2617, HTTP Authentication: Basic and Digest Access Authentication. Basic authentication works as follows:

- For requests to Web Services that require authentication, the server returns 401 (Unauthorized). The response will include a WWW-Authenticate header, indicating the server supports Basic authentication.
- The client sends another request, with the client credentials in the Authorization header. The credentials are formatted as the string "*name:password,*" base64-encoded. The credentials are not encrypted.

Because the credentials are sent unencrypted to the server, the resources contained within the AssureID Connect Web Services are only accessible using the HTTPS protocol.

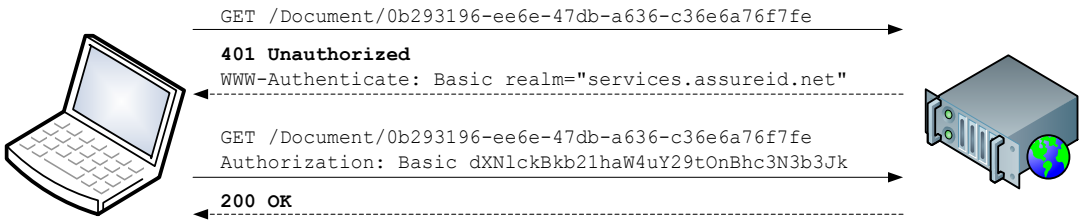


Figure 1. HTTP Basic authentication

Basic authentication is performed within the context of a *realm*. The server will include "services.assureid.net" as the realm in the WWW-Authenticate header.

Programmability

The following example demonstrates one approach on how a C# client can easily access a secured AssureID Connect Web Services Basic Authentication resource.

Example

```
using System;
using System.Net;

using AssureTec.AssureID.Web.SDK;

try
{
    //--- Create web service client.
    Uri serverAddress = new Uri("https://services.assureid.net");
    NetworkCredential credential = new NetworkCredential("user@domain.com", "password");
    AssureIDServiceClient client = new AssureIDServiceClient(serverAddress, credential);

    //--- Get document from web service.
    Guid instanceId = new Guid("0b293196-ee6e-47db-a636-c36e6a76f7fe");
    Document = client.GetDocument(instanceId);

    //--- Do additional work.
}
catch (WebException e)
{
    Console.WriteLine(e.Message);
}
```

Language support

All API calls now support the **Accept-Language** header, which lets you display authentication details (responses) in a specified language. AssureID supports the following languages:

▪ Arabic (ar)	▪ French (fr)	▪ Portuguese (pt)
▪ Czech (cs)	▪ German (de)	▪ Russian (ru)
▪ Dutch (nl)	▪ Greek (el)	▪ Spanish (es)
▪ English (en)	▪ Italian (it)	▪ Turkish (tr)

The following example shows the Accept-Language header for the languages English (en) and German (de).

Example

```
GET http://connect-service-host/AssureIDService/Document/3701e605-b28f-480b-8354-
3f0b308646e1 HTTP/1.1
Authorization: Basic XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Accept-Language: en
Accept: application/json
Host: connect-service-host

GET http://connect-service-host/AssureIDService/Document/3701e605-b28f-480b-8354-
3f0b308646e1 HTTP/1.1
Authorization: Basic XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Accept-Language: de
Accept: application/json
Host: connect-service-host
```

Character set support

AssureID supports optical character recognition (OCR) of the following character sets:

▪ Arabic	▪ Japanese
▪ Cyrillic	▪ Korean
▪ Chinese	▪ Thai
▪ Hebrew	▪ Vietnamese
▪ Baltic	▪ Turkish

Integration tools

Two sets of programming tools are currently available that enable integration with AssureID Connect:

- The .NET Web SDK is a .NET-based assembly that encapsulates the REST API and provides an easy-to-integrate set of classes for interacting with the AssureID Connect web service.
- The REST API is the actual AssureID Connect Web Services interface. Developers may choose to integrate directly with this interface and not utilize the .NET Web SDK.

.NET Web SDK

The AssureID Connect Web Services Integration Kit (SDK) includes a set of .NET classes that encapsulate the AssureID Connect Web Services and provide an easy-to-use interface. For more information, see the *AssureID Connect SDK (.NET) Reference (AssureTec.AssureID.Web.SDK.NET)* included in the kit.

REST API

Developers also have the option of interfacing directly to the AssureID Connect REST Web Service. If using this interface, it is the responsibility of the developer to properly set up request structure, content, and authentication as well as parse the returned responses. For more information, see the *AssureID Connect SDK (REST) Reference (AssureTec.AssureID.Web.SDK.REST)* included in the kit.

Image requirements and recommendations

The minimum requirement for processing a document with AssureID Connect is posting the white (light source) front (side) image of the document. There are specific imaging requirements that must be met to ensure that a document is correctly and accurately processed. These include cropping, resolution, and compression. The minimum and maximum accepted image size is defined by AssureID Connect in terms of pixel dimensions. The minimum pixel width and height is 100 x 100 pixels. AssureID Connect will accept images independent of a specified resolution, or in the absence of accurate resolution information. While AssureID Connect still benefits from accurate image resolution reporting, it will assess the resolution during classification.

Image color depth requirements

The following table describes the image color depth requirements for each illumination source.

Note All images should be 24-bit RGB, except Near-infrared images, which should be 8-bit grayscale.

Table 1. Image color depth requirements

Illumination source	Color depth requirement
Visible (white)	24-bit RGB color
Near-infrared	8-bit grayscale
Ultraviolet-A	24-bit RGB color

Image naming conventions

The following table describes the naming convention for each illumination source for images used with the *i-D*entify Emulator.

Table 2. Image naming conventions

Illumination source	Naming convention
Visible (white)	White
Near-infrared	NearInfrared
Ultraviolet-A	UltravioletA
Ultraviolet-A Back	Back-UltravioletA
Visible Back	Back-White
Near-infrared Back	Back-NearInfrared

Image file formats

The following table describes the supported image file formats and their respective file extensions.

Table 3. Supported image file formats

Format	File extension
Windows Bitmap	bmp
PNG	png
TIFF	tiff
JPEG	jpeg
JPEG-XR	vnd.ms-photo
JPEG 2000	jp2

Image resolution

Although the minimum supported ***physical*** resolution is 300 DPI (dots per inch), higher resolutions can improve document classification and OCR accuracy. For document authentication, Acuant *strongly* recommends using a minimum resolution of 600 DPI. If the resolution is not specified, AssureID will determine the resolution during classification.

Note If you call **GET Classification**, the resolution in the metrics is an estimation based on the expected size you provided. If you call **GET Metrics** after calling **GET Classification**, the resolution in the metrics is based on the resolution/size determined during classification.

The original images from an imaging device should not be resized or scaled to a smaller size, unless the resolution is very high (greater than 800 DPI). Images with a resolution greater than 800 DPI should be scaled down in size and resolution *before* compression to optimize processing performance. The aspect ratio of the image should not be changed, and the horizontal and vertical resolution of the image should ideally be within 1% of each other.

Note The AssureID Connect photo printing and substrate printing tests require a high-quality image to be effective. Submitting a low-resolution document image may result in these tests being skipped.

Glare and sharpness metrics

Glare metrics describe the amount of *glare* on the image and is a measure of light uniformity across the document. Acuant recommends a score of 50 or greater to be considered an image with no glare or an acceptable level of glare. Sharpness metrics describe the *sharpness* of the image and is a measure of edge contrast across the document. A score of 0 means that the image is very blurry (not sharp) and a score of 100 means that the image is very sharp. By default, Acuant recommends a sharpness grade of at least 50 or greater to be considered an image with an acceptable level of sharpness.

Image compression

Do *not* excessively compress an image. Overcompression may degrade data capture and authentication performance. Use the following recommended minimum compressed image file sizes that correspond to the applicable light source.

Note The recommended image compression for an image using a visible (white) light source has been increased (from 250 KB minimum/300-600 KB recommended to 500 KB minimum/600-1200 KB recommended) in order to perform additional authentication tests.

Table 4. Image compression recommendations

Light Source	Minimum	Recommended
Visible	500 KB	600-1200 KB
Near-Infrared	200 KB	250-500 KB
Ultraviolet	125 KB	150-300 KB
Coaxial	125 KB	150-300 KB

Note The AssureID Connect photo printing and substrate printing tests may be affected by compression, therefore you should avoid overcompressing an image. Submitting an overcompressed image may result in these tests being skipped.

Image cropping

Use the following guidelines for image cropping.

- Use **None** if the images are already tightly cropped.
- Use **Always** if the images are uncropped.

Acuant recommends that you *only* use **Automatic** as a last resort. The only scenario where it may provide better results than **None** or **Always** is if you have an ID1-sized document where one side is cropped and the other side is not. Note that the **Automatic** setting will generally result in an unnecessarily high **Unknown** rate and will cause inconsistent behavior for non-ID1-sized documents. If possible, you will get better results if you run one transaction set to **Always**, and then if you get an Unknown result, retry with **None**.

The following table describes the settings for image cropping.

Table 5. Image cropping modes

Cropping mode	Description
Always	Cropping is *always* performed on posted images. **Note** This will result in additional processing time if you submit images that are already cropped.
None	No cropping is performed on posted images. This mode should be used when the images are already tightly cropped to the document when posted. (**DEFAULT**)
Automatic	Automatically determines whether image cropping is required on each image and performs the cropping and calculates image resolution when necessary. **Note** Acuant does *not* recommend using the **Automatic** setting.

If the cropping mode is set to **None** and the expected image size is not specified as ID1 or ID3, then AssureID will determine the resolution. Acuant recommends setting the cropping mode to **Always**.

Note Acuant does *not* recommend using the **Automatic** setting

Examples

The following are examples of uncropped, properly cropped, and poorly cropped ID3 size sample (passports) documents.

Figure 1. Uncropped images

The following figure illustrates an overcropped image. Notice that the MRZ is cut off and the left side is too tightly cropped.

Figure 2. Properly cropped image

Cropping an image

The following procedure describes how to crop an image, using an editor such as such as *Irfanview*.

Crop an image

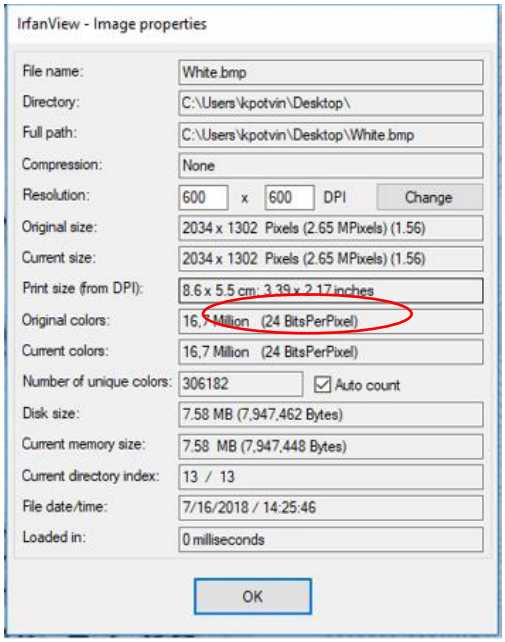
1. Open the image:

2. Select the area to be cropped.

Note **For passports *only*, capture the data page.**

3. From the **Select** menu, click **Image > Information**:
4. Edit the image properties to set the correct resolution.

Note Many images are set with a *default* resolution of 72 DPI or 96 DPI. This *must* be changed to the *actual* resolution, based on the number of pixels that are in the saved cropped image.



To determine the actual resolution:

1. Divide the number of pixels in the document by the dimensions (size) of the document type (ID1 or ID3):

ID1	3.37”W x 2.125”H
ID3	4.921”W x 3.456”H

2. Round up decimals to a whole number.
   For example, for the sample image:

2034 ÷ 3.37	=	603 dpi
1302 ÷ 2.125	=	612 dpi

5. Select **Change**.
6. Click **OK** to apply.
7. **Save** the image using the proper image naming convention.

Enumerations

This section provides a definition of all enumeration values used by AssureID Connect. See the *AssureID Connect SDK Reference Guide (.NET)* help file for more information.

- AuthenticationResult
- AuthenticationSensitivity
- ChipAuthenticationResult
- ChipAuthenticationType
- ChipDataGroup
- ClassificationMode
- CroppingExpectedSize
- CroppingMode
- DocumentClass
- DocumentDataSource
- DocumentDataType
- DocumentElement
- DocumentProcessMode
- DocumentSide
- DocumentSize
- DocumentStatus
- GenderType
- IssuerType
- LightSource
- SensorType

AuthenticationResult

```
public enum AuthenticationResult
{
    Unknown = 0,
    Passed = 1,
    Failed = 2,
    Skipped = 3,
    Caution = 4,
    Attention = 5
}
```

AuthenticationSensitivity

```
public enum AuthenticationSensitivity
{
    Normal = 0,
    High = 1,
    Low = 2
}
```

ChipAuthenticationResult

```
public enum ChipAuthenticationResult
{
    NotPerformed = 0,
    Passed = 1,
    Failed = 2
}
```

ChipAuthenticationType

```
public enum ChipAuthenticationType
{
    ActiveAuthentication = 0,
    BasicAccessControl = 1,
    ChipAuthentication = 2,
    PassiveAuthentication = 3,
    SupplementalAccessControl = 4,
    TerminalAuthentication = 5
}
```

ChipDataGroup

```
public enum ChipDataGroup
{
    COM = 0,
    DG1 = 1,
    DG2 = 2,
    DG3 = 3,
    DG4 = 4,
    DG5 = 5,
    DG6 = 6,
    DG7 = 7,
    DG8 = 8,
    DG9 = 9,
    DG10 = 10,
    DG11 = 11,
    DG12 = 12,
    DG13 = 13,
    DG14 = 14,
    DG15 = 15,
    DG16 = 16,
    CardAccess = 17,
    CVCA = 18,
    SOD = 19
}
```

ClassificationMode

```
public enum ClassificationMode
{
    Automatic = 0,
    Manual = 1
}
```

CroppingExpectedSize

```
public enum CroppingExpectedSize
{
    None = 0,
    ID1 = 1,
    ID3 = 3
}
```

CroppingMode

```
public enum CroppingMode
{
    None = 0,
    Automatic = 1,
    Always = 3
}
```

DocumentClass

```
public enum DocumentClass
{
    Unknown = 0,
    Passport = 1,
    Visa = 2,
    DriversLicense = 3,
    IdentificationCard = 4,
    Permit = 5,
    Currency = 6,
    ResidenceDocument = 7,
    TravelDocument = 8,
    BirthCertificate = 9,
    VehicleRegistration = 10,
    Other = 11
    WeaponLicense = 12,
    TribalIdentification = 13,
    VoterIdentification = 14
    Military = 15
    ConsularIdentification = 16
}
```

DocumentDataSource

```
public enum DocumentDataSource
{
    None = 0,
    Barcode1D = 1,
    Barcode2D = 2,
    ContactlessChip = 3,
    MachineReadableZone = 4,
    MagneticStripe = 5,
    VisualInspectionZone = 6,
    Other = 7
}
```

DocumentDataType

```
public enum DocumentDataType
{
    Barcode2D = 0,
    MachineReadableZone = 1,
    MagneticStripe = 2
}
```

DocumentElement

```
public enum DocumentElement
{
    Unknown = 0,
    None = 1,
    Photo = 2,
    Data = 3,
    Substrate = 4,
    Overlay = 5
}
```

DocumentProcessMode

```
public enum DocumentProcessMode
{
    Default = 0,
    CaptureData = 1,
    Authenticate = 2
    Barcode = 3
}
```

DocumentSide

```
public enum DocumentSide
{
    Front = 0,
    Back = 1
}
```

DocumentSize

```
public enum DocumentSize
{
    Unknown = 0,
    ID1 = 1,
    ID2 = 2,
    ID3 = 3,
    Letter = 4,
    CheckCurrency = 5,
    Custom = 6
}
```

DocumentStatus

```
public enum DocumentStatus
{
    None = 0,
    Classified = 1,
    Complete = 2,
    Error = 3
}
```

GenderType

```
public enum GenderType
{
    Unspecified = 0,
    Male = 1,
    Female = 2,
    Unknown = 3
}
```

IssuerType

```
public enum IssuerType
{
    Unknown = 0
    Country = 1,
    StateProvince = 2,
    Tribal = 3
    Municipality = 4
    Business = 5
    Other = 6
}
```

LightSource

```
public enum LightSource
{
    White = 0,
    NearInfrared = 1,
    UltravioletA = 2,
    CoaxialWhite = 3,
    CoaxialNearInfrared = 4
}
```

SensorType

```
public enum SensorType
{
    Unknown = 0,
    Camera = 1,
    Scanner = 2,
    Mobile = 3
}
```

API changes by release

This section contains a detailed list of changes to the AssureID Connect Web Services API by release, including any impact on compatibility. The following interface-impacting enhancements and changes were made to the AssureID Connect Web Services API.

API changes in version 2.14.2

.NET

.NET SDK member	Parameter	Description of changes
AssureIDServiceClient	**PUT DocumentData** method	**New**. Enables the re-capture and re-upload of data from the back side of a document that contains a barcode (1D or 2D), MRZ, or magstripe.

REST

REST API member	Parameter	Description of changes
PUT Document/ INSTANCEID/Data	**DocumentData** method	**New.** Enables the re-capture and re-upload of data from the back side of a document that contains a barcode (1D or 2D), MRZ, or magstripe.

API changes in version 2.9.0

The following interface-impacting enhancements and changes were made in this version of the **AssureID Connect .NET SDK**:

.NET SDK member	Parameter	Description of changes
GetDocumentFieldImage method	**quality**	**New.** Enables a user to specify the quality of an image to accept RANGE is 1-100. **DEFAULT** is **50**.
GetDocumentImage method	**quality**	**New.** Enables a user to specify the quality of an image to accept RANGE is 1-100. **DEFAULT** is **50**.
UserAuthorization class		**New**. Describes an authorized user's activity for a Document
AssureIDSessionClient	**GetDocumentActivity** method	**New**. Returns authorized user activity for a Document

The following changes and additions are included this version of the AssureID Connect REST API:

REST API member	Parameter	Description of changes
GET Document/ INSTANCEID/Field/Image	**quality**	**New.** Enables a user to specify the quality of an image to accept RANGE is 1-100. DEFAULT is 50.
GET Document/ INSTANCEID/Image	**quality**	**New.** Enables a user to specify the quality of an image to accept RANGE is 1-100. DEFAULT is 50.
GET Document/Instance /Activity		**New** endpoint. Returns an authorized user's activity for a Document

API changes in version 2.8.2

There were no API changes in version 2.8.2.

API changes in version 2.8.1

The following interface-impacting enhancements and changes were made in version 2.8.1 of the **AssureID Connect .NET SDK**:

SDK class	SDK member	Description of changes
DocumentType	IssuerType	**New** property that identifies the issuing authority type for the document

The following interface-impacting enhancements and changes were made in version 2.8.1 of the **AssureID Connect REST API**:

REST method	REST element	Description of changes
DocumentType	IssuerType	**New** property that identifies the issuing authority type for the document

API changes in versions 2.6.0 – 2.7.3

There were no API changes in versions 2.6.0 to 2.7.3.

API changes in version 2.5.0

The following interface-impacting enhancements and changes were made in version 2.5.0 of the **AssureID Connect .NET SDK**:

SDK class	SDK member	Description of changes
DocumentClass	Military	**New** document class. An identity document issued by a government agency to grant or restrict access to military information, locations, or privileges.

The following interface-impacting enhancements and changes were made in version 2.5.0 of the

AssureID Connect REST API:

REST method	REST element	Description of changes
DocumentClass	Military	**New** document class. An identity document issued by a government agency to grant or restrict access to military information, locations, or privileges.

API changes introduced in versions 2.1.1 – 2.4.0

There were no API changes in versions 2.1.1 to 2.4.0.

API changes in version 2.1.0

The following REST API method was added in version 2.1.0.

REST method	REST element	Description of changes
PUT Document/Instance	Image	**New** method. Replaces an image in a document instance. **Note** The image must first be posted before it can be replaced.

API changes in version 2.0.2

There were no API changes in version 2.0.2.

API changes in version 2.0.1

The AssureID Connect SDK was modified to add the optional calculation of metrics for sharpness and glare. This feature allows users to determine the quality of a captured image, on a scale of 0 – 100 (with 100 being the best quality), after an image is posted but before completing the transaction. You can use these metrics to evaluate whether the captured images are of sufficient quality. Acuant recommends that you accept images that yield a quality factor of at *least* 50, however, you may choose to modify this threshold to better suit your application's specific requirements. A setting of 50 means that you would want to recapture images that are determined to be of insufficient quality (below 50).

Note The use of these metrics do *not* have any bearing on the document result. It provides the user with a set of metrics to evaluate the image quality *before* completing a transaction.

.NET SDK member	Parameter	Description of changes
DocumentImage class	**GlareMetric** property	New. If specified, indicates the relative amount of glare detected on an image, on a scale of 0 (obscured by glare) to 100 (no glare detected).
	SharpnessMetric property	**New**. If specified, indicates the relative sharpness of this image, on a scale of **0** (blurry) to **100** (sharp).
AssureIDSessionClient	**GetDocument** method	**Modified**. Returns metrics via the **DocumentImage** class.
AssureIDSessionClient	**GetDocumentImage Metrics** method	**New** parameter. Returns metrics via the **DocumentImage** class.
AssureIDSessionClient	PostDocumentImage method	**New** parameter. If specified, calculates the quality metrics of sharpness and glare on an image. **True |** False (default)

The following changes and additions are included this version of the AssureID Connect REST API:

REST API member	Parameter	Description of changes
POST DocumentImage	metrics	New metrics parameter. Calculates the amount of glare or sharpness present on an image. True | **False** (DEFAULT)
GET DocumentImageMetrics		**New** endpoint. Returns any available metadata, including the quality metrics of sharpness and glare, if requested for a previously posted image.
GET Document		Modified to include returning any available metadata, including the quality metrics of sharpness and glare, if requested for a previously posted image.

API changes in versions 1.9.1 — 1.9.9

There were no API changes from version **1.9.1** to **1.9.9**.

API changes in version 1.9.0

The following interface-impacting enhancements and changes were made in version 1.9.0 of the AssureID Connect .NET SDK:

SDK class	SDK member	Description of changes
DocumentSettings	ProcessMode.Barcode	New enumeration. Only the document's barcode is processed.
DocumentClass	WeaponLicense TribalIdentification	**New** document types.

SDK class	SDK member	Description of changes
	VoterIdentification	

The following interface-impacting enhancements and changes were made in version 1.9.0 of the AssureID Connect REST API:

REST method	REST element	Description of changes
POST Document/Instance	ProcessMode.Barcode	New enumeration. Only the document's barcode is processed.
DocumentClass	WeaponLicense TribalIdentification VoterIdentification	**New** document types.

All API calls now support the **Accept-Language** header, which enables you to display authentication details (responses) in a specified language.

API changes in version 1.8.8

The **Barcode** enumeration was added to support 2D barcode-only read mode.

API changes in version 1.8.7

There were no API changes from version 1.8.6 to 1.8.7.

API changes in version 1.8.6

API changes in version 1.8.6 were made to detect document size if cropping is enabled.

SDK Class	SDK Member	Description of Changes
CroppingMode	ImageCroppingMode	**New** enumeration
CroppingExpectedSize	ImageCroppingExpectedSize	**New** enumeration

API changes in version 1.8.5

API changes were made in version 1.8.5 to improve automatic document image cropping support. These changes are backward-compatible such that existing applications written using the .NET or REST APIs will continue to function without modification. Modification will be required to take advantage of new functionality.

SDK Class	SDK Member	Description of Changes
CroppingMode		**New** enumeration value **Always**

API changes in version 1.8.4

There were no API changes from version 1.8.3 to 1.8.4.

API changes in version 1.7.0

API changes from version 1.6.0 to 1.7.0 were made to support automatic document image cropping. These changes are backward-compatible such that existing applications written using the .NET or REST APIs will continue to function without modification. Modification will be required to take advantage of new functionality.

The following interface-impacting enhancements and changes were made in release 1.7.0 of the AssureID Connect .NET SDK:

SDK Class	SDK Member	Description of Changes
CroppingMode	ImageCroppingMode	**New** enumeration
CroppingExpectedSize	ImageCroppingExpectedSize	**New** enumeration
DocumentSettings	ImageCroppingMode	**New** property
	ImageCroppingExpectedSize	**New** property
AssureIDSessionClient	PostDocumentImage	**Modified**. Image may now be uncropped without accurate resolution specified if Automatic cropping was enabled when the document instance was created.

The following interface-impacting enhancements and changes were made in release 1.7.0 of the AssureID Connect REST API:

REST Method	REST Element	Description of Changes
POST Document/Instance	ImageCroppingMode	**New** enumeration
	ImageCroppedExpectedSize	**New** enumeration
POST DocumentImage	Image	**Modified**. Image may now be uncropped without accurate resolution specified if Automatic cropping was enabled when the document instance was created.

API changes in version 1.6.0

There were no API changes from version 1.5.0 to version 1.6.0.

API changes in version 1.5.0

The AssureID Connect Web Services API 1.5.0 is generally backward-compatible with the version 1.4.0 API. However, the AssureID Connect Web Services client SDK is not backward-compatible due to some of the below-listed enhancements and changes that break object serialization and deserialization.

Web Services Operation	What changed?	Comments
GET DocumentTypes	Number of query string arguments changed	**New SubscriptionId** query string argument. If supplied, the value of this argument must be set to a valid subscription identifier that is obtained using the new GET Subscriptions operation.
GET Subscriptions	New operation	New

> **Note** Applications that were compiled against the AssureID Connect Web Services API 1.4.0 may not need to be recompiled; those that were compiled against the AssureID Connect client SDK 1.4.0 must be recompiled. In all cases, some minor source code changes may be necessary because of the improvements made to the software.

The following interface-impacting enhancements and changes were made in release 1.5.0 of the AssureID Connect Web Services API:

Client SDK Member	What changed?	Comments
Document class	Improved	New **ProcessMode** read-only property that yields a **DocumentProcessMode** enumeration value. Added a **Subscription** read-only property that yields a **Subscription** object
DocumentClassification class	Improved	New **PresentationChanged** read-only property
DocumentDataSource enumeration	Improved	**New Other** member
DocumentField class	Improved	**New DataSource** read-only property that yields a **DocumentDataSource** enumeration value
DocumentProcessMode enumeration	Added	**New** enumeration
DocumentSettings class	Improved	Modified. Renamed the ManualDocumentTypeId read/write property to ManualDocumentType. **Note** The value of this property must be set to a valid **DocumentType** object that is obtained from the service using the GET **DocumentType** operation.
		New ProcessMode read/write property containing a **DocumentProcessMode** enumeration value for

Client SDK Member	What changed?	Comments
		specifying how a document shall be processed. Added a **SubscriptionId** read/write property for specifying the subscription that shall be used for document processing. **Note** The value of this property must be set to a valid **Subscription** identifier (Guid) that is obtained from the service using the GET **Subscriptions** operation.
Subscription class	Added	**New** class

API changes in version 1.4.0

The AssureID Connect Web Services API and client SDK 1.4.0 are *not* binary compatible with version 1.3.0 of Web Services API and client SDK. Applications that were compiled against the AssureID Connect Web Services API or client SDK 1.3.0 should be recompiled. Some source code changes may be necessary because of the improvements made to the software.

Note The serialization namespace of all data returned from the AssureID Connect Web Services and the client SDK 1.4.0 data contracts has been changed to **http://services.assureid.net/2014/09**.

The following interface-impacting enhancements and changes were made in version 1.4.0 of the AssureID Connect Web Services API:

Web Services Operation	What changed?	Comments
GET DocumentChipData	Added	**New** operation
POST DocumentChipData	Added	**New** operation
GET DocumentData	Number of query string arguments	**Modified**. Removed the **DocumentDataSubType** query string argument. It was used to specify contactless chip data items. The new GET **DocumentChipData** operation can be used for this purpose.
POST DocumentData	Number of query string arguments	**Modified**. Removed the **DocumentDataSubType** query string argument. It was used to specify contactless chip data items. The new **POST DocumentChipData** operation can be used for this purpose.
GET DocumentImage	Query string argument	**Modified**. Renamed the **DeviceLight** query string argument to **LightSource**.

Web Services Operation	What changed?	Comments
POST DocumentImage	Query string argument	**Modified**. Renamed the DeviceLight query string argument to **LightSource**.
POST DocumentInstance	New operation	**New** operation.
GET DocumentLog	New operation	**New** operation.

Client SDK Member	What changed?	Comments
AssureIDServiceClient class	Improved	**New**. Added a **ServiceAddress** read-only property.
ChipAuthentication class	Changed	**Modified** the return type of the **Result** read-only property from **AuthenticationResult** to **ChipAuthenticationResult**
DeviceInfo class	Improved	Added a constructor overload accepting **SensorType** argument and added new **SensorType** read-only property
DeviceType class	Improved	**New** constructor overload accepting **SensorType** argument and added new **SensorType** read-only property
Document class	Changed/enhanced	**Modified**. Renamed the **Id** read-only property to InstanceId. Added an **AuthenticationSensitivity** read-only property
DocumentClassification class	Changed	Removed the **GenericIssuerCode** and **GenericIssuerName** read-only properties **Note** Use the read-only **Type.Issuer** and **Type.IssuerName** properties instead.
DocumentDataField class	Improved	**New IsImage** read-only property.
DocumentField class	Improved	**New IsImage** read-only property.
DocumentImage class	Changed	**Modified** the return type of the **Light** read-only property from the **DeviceLight** enumeration to the **LightSource** enumeration.
DocumentImageType class	Changed	**Modified** the return type of the **Light** read-only property from the **DeviceLight** enumeration to the **LightSource** enumeration.
ChipAuthenticationResult enumeration	Added	**New** enumeration

Client SDK Member	What changed?	Comments
ChipAuthenticationType enumeration	Changed/enhanced	**New**. Added the following members: **PassiveAuthentication** **SupplementalAuthentication** **Removed** the following members: **CountrySignerRevocation** **DocumentSignerRevocation** **DocumentSignerValidation** **HashValueValidation** **SignatureValidation**
ChipDataGroup enumeration	Added	**New** enumeration.
DeviceLight enumeration	Removed	**Deleted**. This enumeration is obsolete. Use the LightSource enumeration instead.
DocumentDataSubType enumeration	Removed	**Deleted**. This enumeration is obsolete. Use the **ChipDataGroup** enumeration instead.
DocumentDataType enumeration	Changed	**Deleted** the **ContactlessChip** member.
LightSource enumeration	Added	**New** enumeration.

HTTP Status codes

The following table lists the HTTP status codes and which API calls use these codes.

Table 6. HTTP status codes

Code	Summary	Description	API Calls
200	OK	The requested resource was found and resolved.	All API calls using GET
201	Created	The requested resource was created. The Location header contains the URL of the created resource.	All API calls using POST
204	No Content	The requested resource was updated or deleted.	All API calls using PUT or DELETE
401	Unauthorized	The caller could not be authenticated (typically due to invalid credentials).	All
403	Forbidden	The referenced or default subscription may be inactive.	All
404	Not Found	The resource, document, or document data could not be found.	All
409	Conflict	An attempt was made to submit the same document image, data, or chip data multiple times (for example, the Front/White image may only be submitted once per transaction).	**PostDocumentImage, PostDocumentChipData, PostDocumentChipAuthentication**
430	Required service operation argument not specified	One or more of the required arguments to the service operation were not specified.	**GetDocumentData, GetDocumentFieldData, GetDocumentFieldImage, PostDocumentChipAuthentication, PostDocumentChipData, PostDocumentData, PostDocumentImage, PostDocumentInstance**
431	Invalid or unsupported service operation argument	One or more of the arguments supplied to the service operation are invalid or unsupported.	**PostDocumentImage**
432	Document settings incomplete	The supplied document settings do not contain all of the required information necessary for processing. At a minimum, a subscription and a document capture device that includes manufacturer and model names are required. If manual classification is requested, a valid document type is also required.	**PostDocumentInstance**
433	Resource identifier not in expected format	The identifier used to reference a specific resource was not supplied in the expected format.	All

Code	Summary	Description	API Calls
434	Subscription not found	The referenced subscription was not found or is temporarily unavailable.	**PostDocumentInstance**
435	Subscription not specified	Multiple subscriptions were found for the authenticated user and one must be specified to invoke this service operation.	**PostDocumentInstance**
436	Manual classification document type not found.	The referenced document type used for manual classification was not found or is temporarily unavailable.	**PostDocumentInstance**
437	Document complete or in error state.	The status of the referenced document is complete or in an error state; additional images and/or data cannot be posted to the document and no further processing may be performed.	**All except PostDocumentInstance**
438	Document image load failure.	The document image could not be loaded from the supplied stream due to a format, compatibility or integrity issue.	**PostDocumentImage, PutDocumentImage**
439	Invalid or unsupported document image pixel depth.	The pixel depth of the document image is invalid or unsupported. The following image pixel depths are supported: **Lighting** / **Pixel Depth** Visible / 24-bit RGB Ultraviolet / 24-bit RGB Near-Infrared / 8-bit Grayscale	**PostDocumentImage, PutDocumentImage**
440	Document image size is outside the acceptable range.	The width and/or height of the document image must not be less than 100 pixels.	**PostDocumentImage, PutDocumentImage**
441	Document image resolution is outside the acceptable range.	The resolution of the document image must be greater than 96 DPI and less than 1,500 DPI.	**PostDocumentImage, PutDocumentImage**
442	Document image resolution difference between each axis is outside the acceptable range.	The resolution difference between the horizontal and vertical axis of the document image must not exceed 5%.	**PostDocumentImage, PutDocumentImage**
500	Internal Server Error.	An internal error occurred. See system log.	All
503	Service Unavailable.	The website may be down, it may not be configured properly, or it may have insufficient credentials to run the website.	All

AssureID Connect document workflows

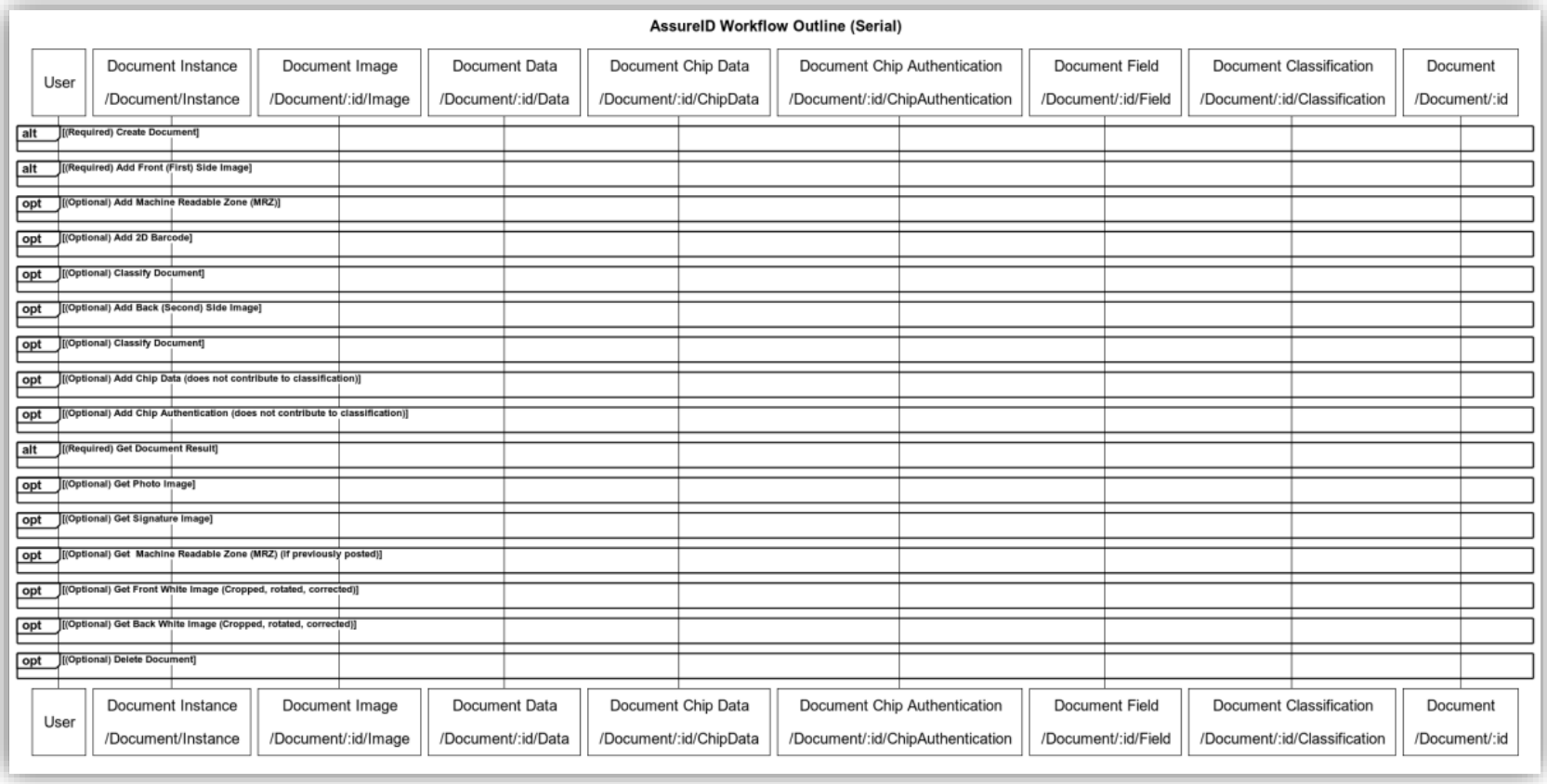
This section describes the AssureID Connect REST API document workflows.

- AssureID Connect Serial Workflow
- AssureID Connect Parallel Workflow

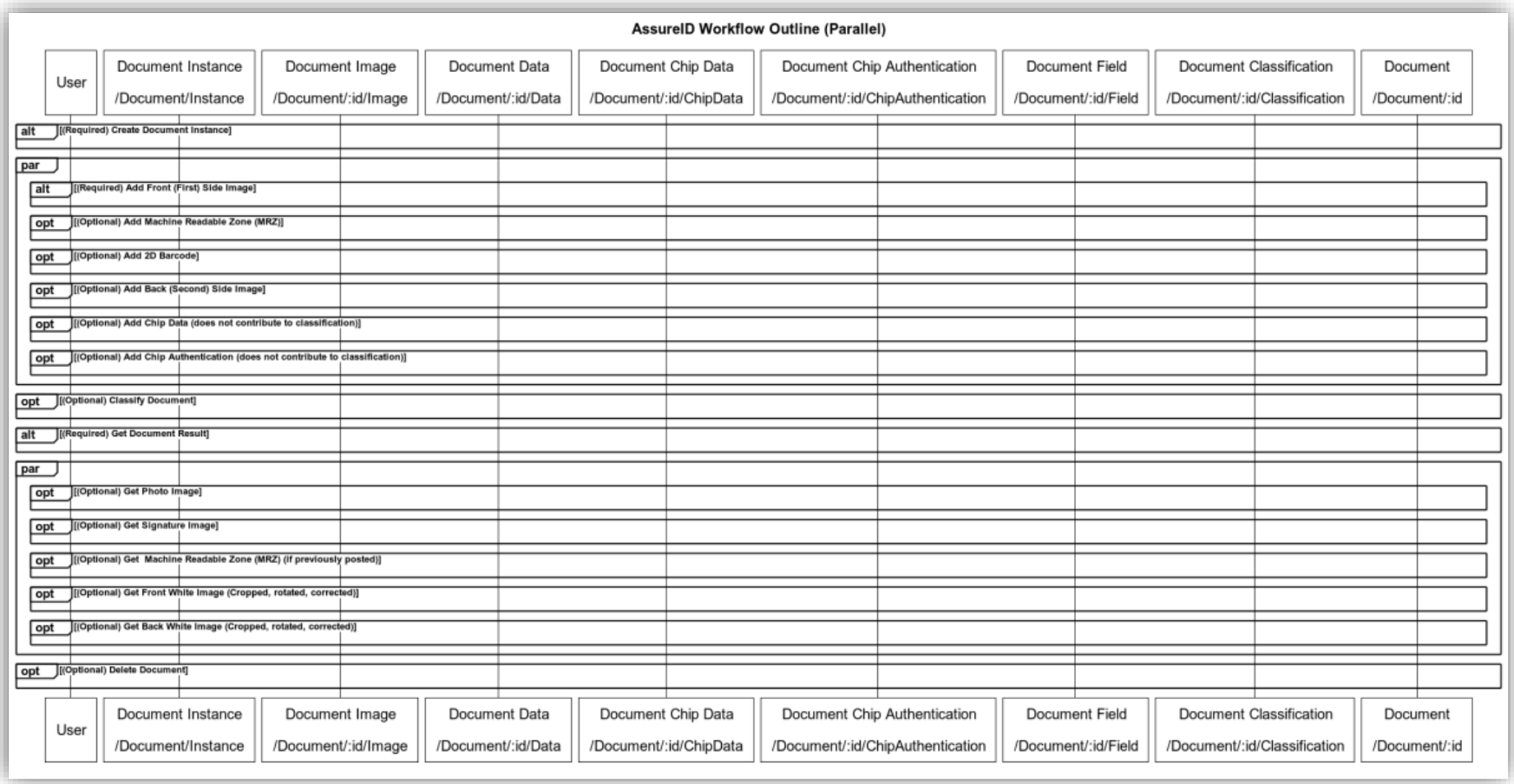
The AssureID Connect API offers the following recommended building blocks.

- Create Document (**REQUIRED**)
- Add Front (First) Side Image (**REQUIRED**)
- Add 2D Barcode (OPTIONAL)
- Add Machine Readable Zone (MRZ) (OPTIONAL)
- Classify_Document (OPTIONAL)
- Add Back (Second) Side Image (OPTIONAL)
- Classify Document (OPTIONAL)
- Add Chip Data (does not contribute to classification) (OPTIONAL)
- Add Chip Authentication (does not contribute to classification) (OPTIONAL)
- Get Document Result (**REQUIRED**)
- Get Document Component Results (OPTIONAL)
- Get Photo Image (OPTIONAL)
- Get Signature Image (OPTIONAL)
- Get Machine Readable Zone (MRZ) (if previously posted) (OPTIONAL)
- Get Front White Image (Cropped, rotated, corrected) (OPTIONAL)
- Get Back White Image (Cropped, rotated, corrected) (OPTIONAL)
- Other Fields (OPTIONAL)
- Delete_Document (OPTIONAL)

AssureID Serial workflow

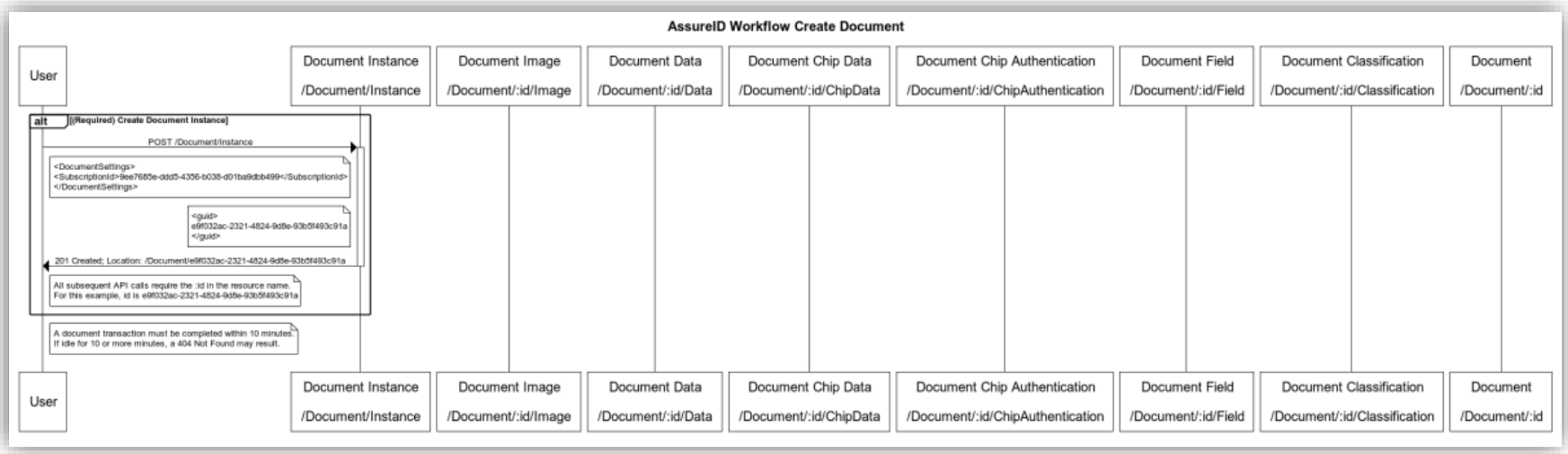


AssureID Parallel workflow



Create Document workflow

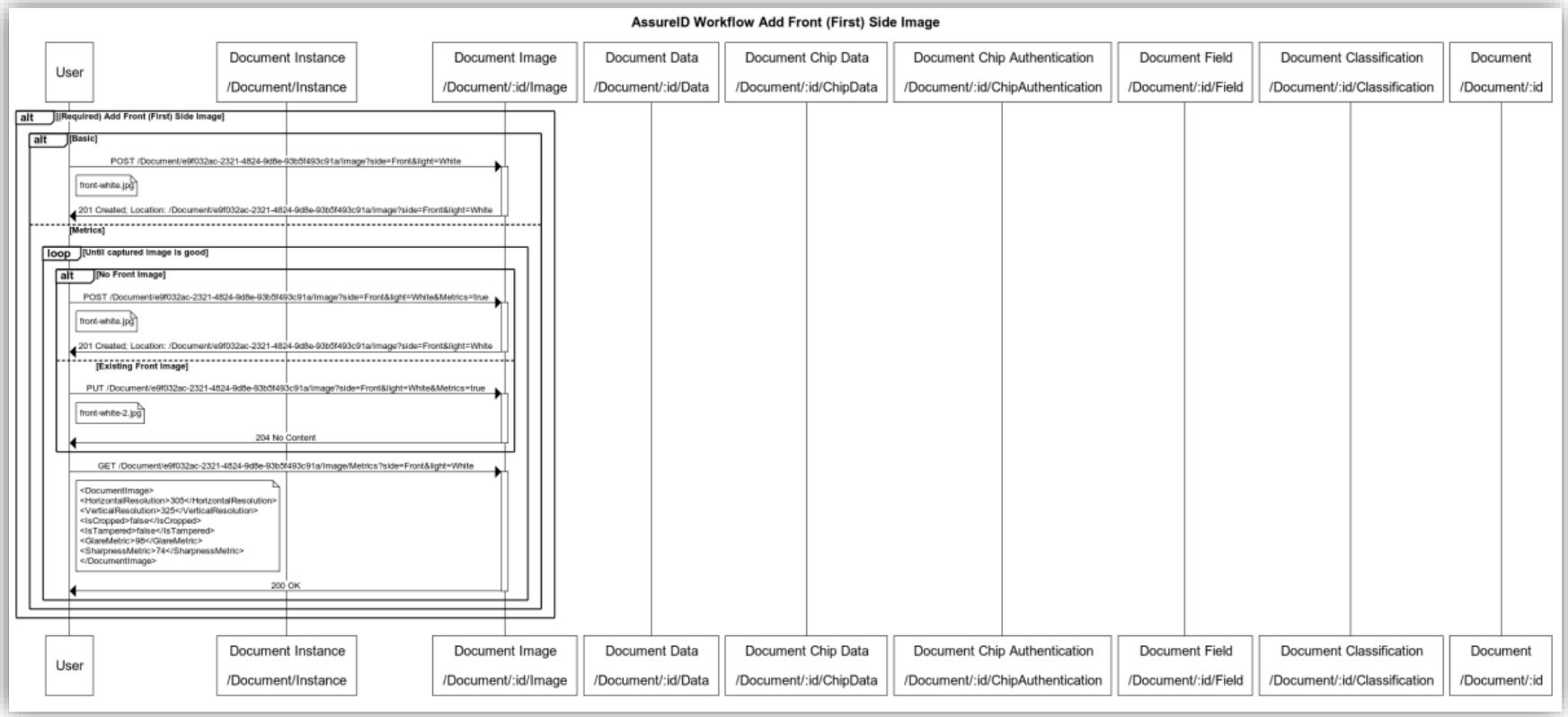
Create Document *must* be the first building block in an AssureID Connect transaction, and can't be run in parallel with any other building block.



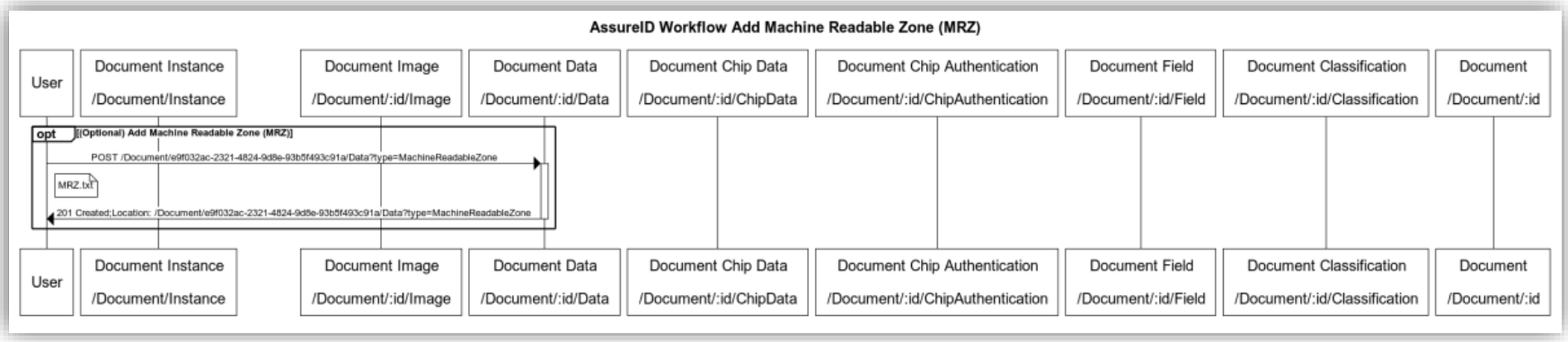
Adding information for the initial classification

Adding the front (first) side image and optionally the Machine Readable Zone (MRZ) may be done in parallel and both should be completed prior to classifying the document.

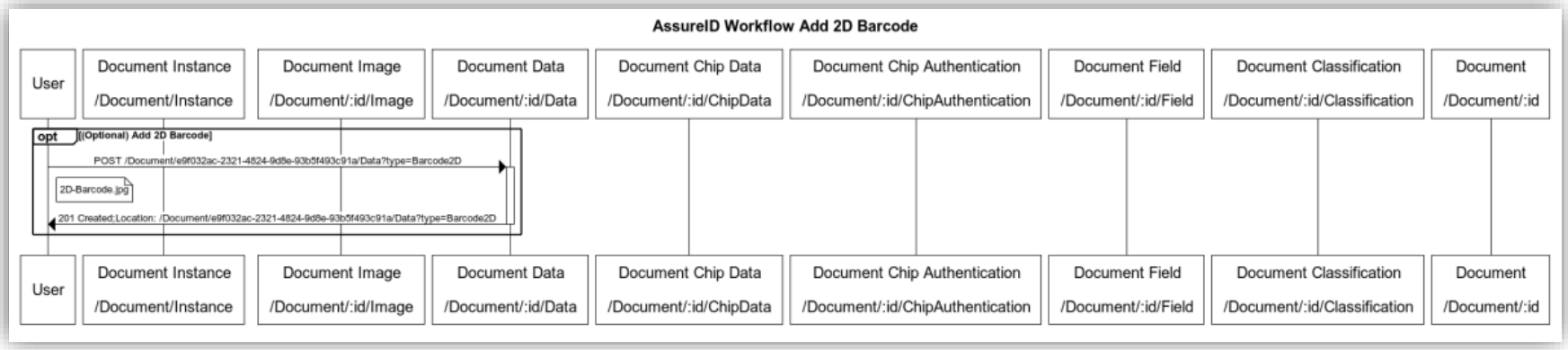
Add Front image



Add Machine Readable Zone (MRZ) *(optional)*

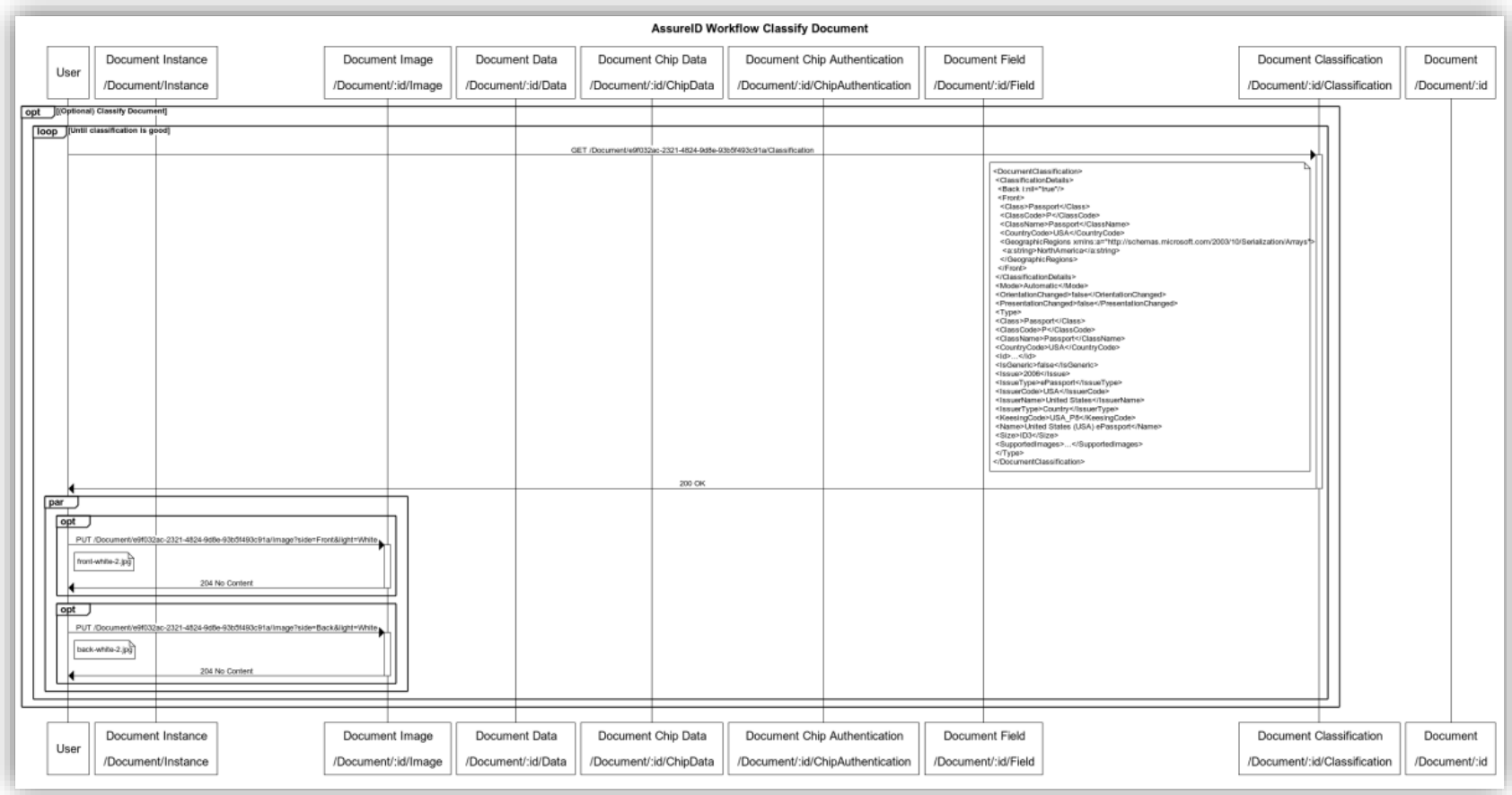


Add 2D Barcode *(optional)*



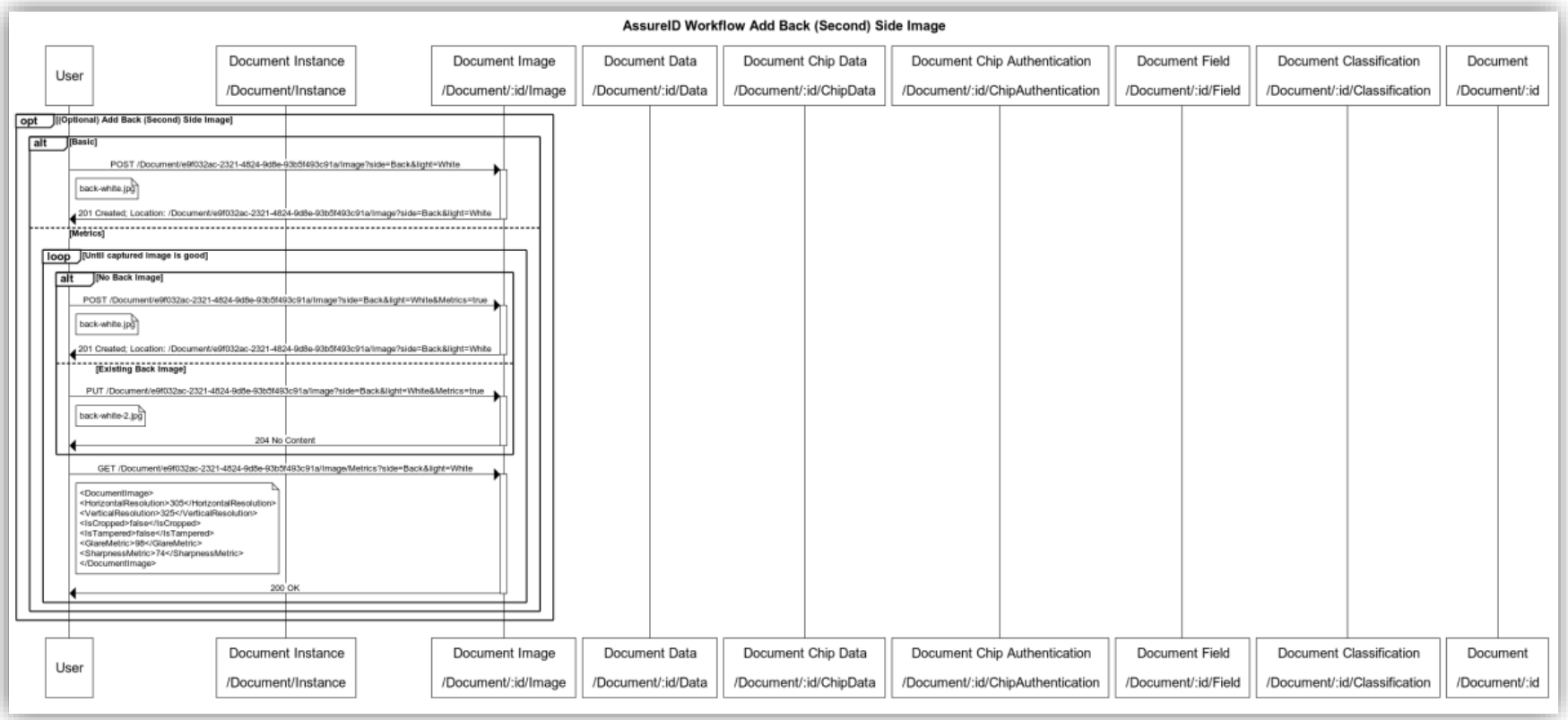
Classify Document *(optional)*

Classifying the document is an optional step and if performed, must be done ***after*** all the information (images, MRZs) are complete.



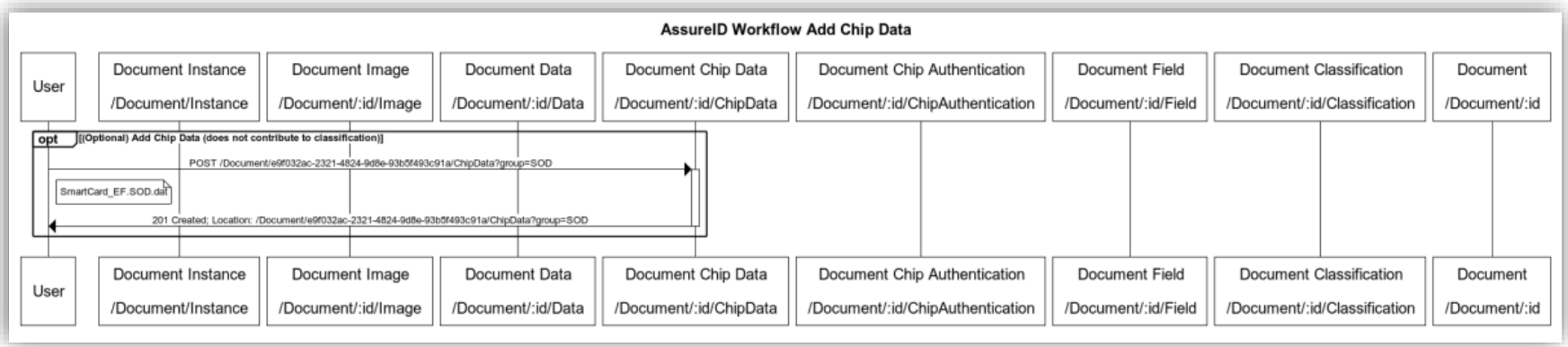
Add Back image *(optional)*

Adding the back (second) side image is optional, and should be done ***after*** classifying. If you skipped the earlier Classify Document step, then this can be done in parallel with adding the front and the MRZ.



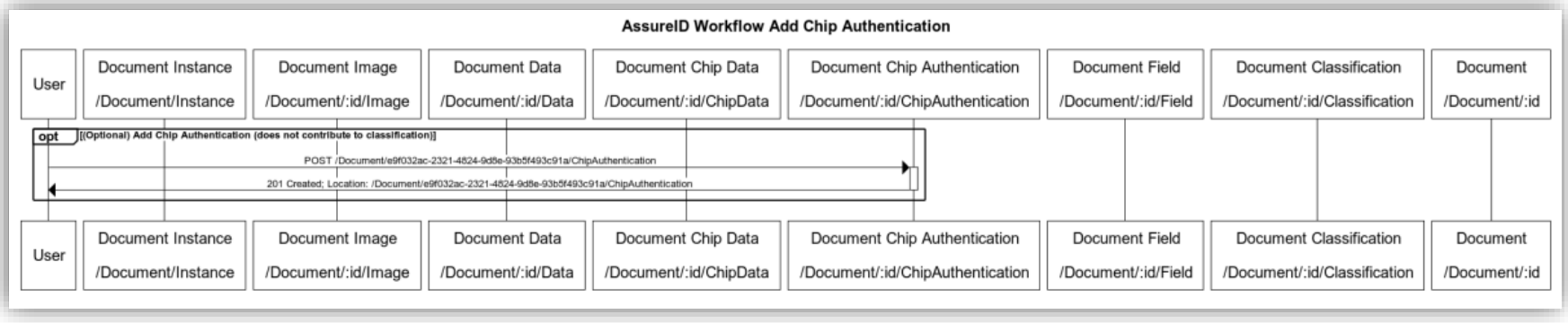
Add Chip data *(optional)*

Adding chip information, while ***not*** contributing to classification, does contribute to authentication, and may be done in parallel.



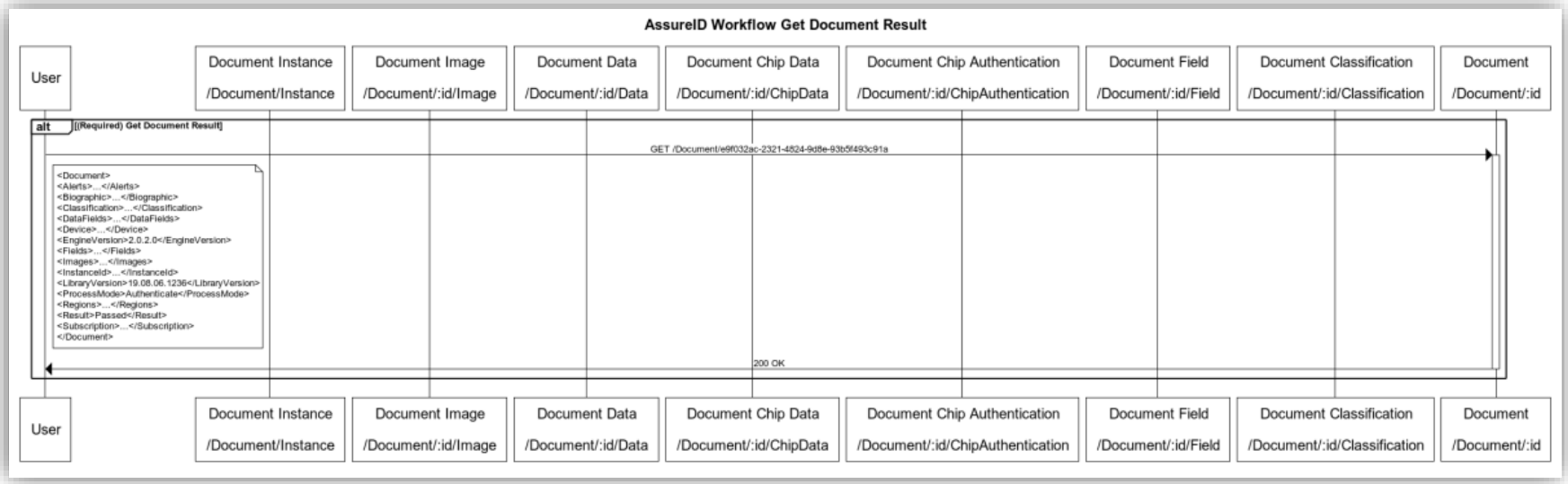
Add Chip authentication *(optional)*

This applies to devices that support chip authentication (validates the device itself).



Get Document result

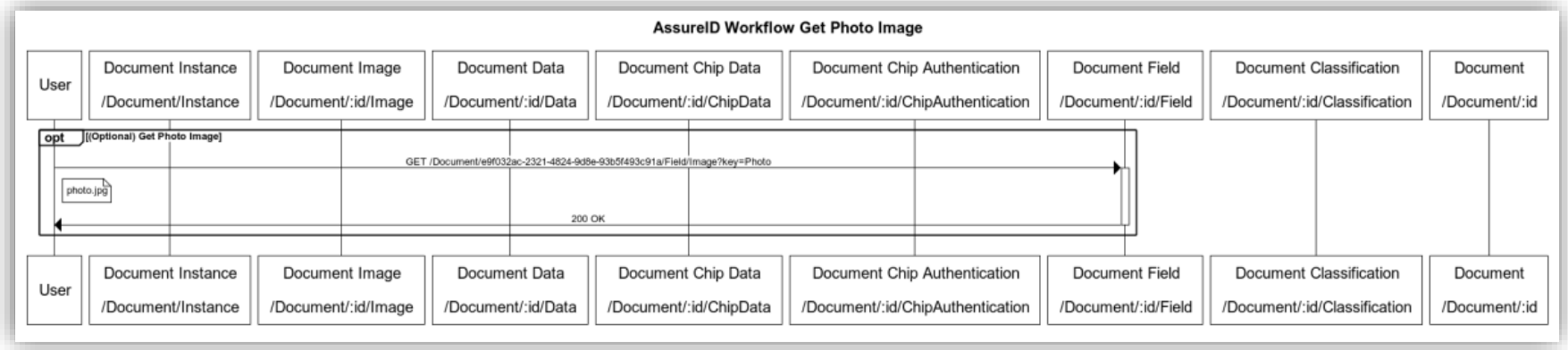
Getting the document result must be done after *all* of the preceding blocks are complete.



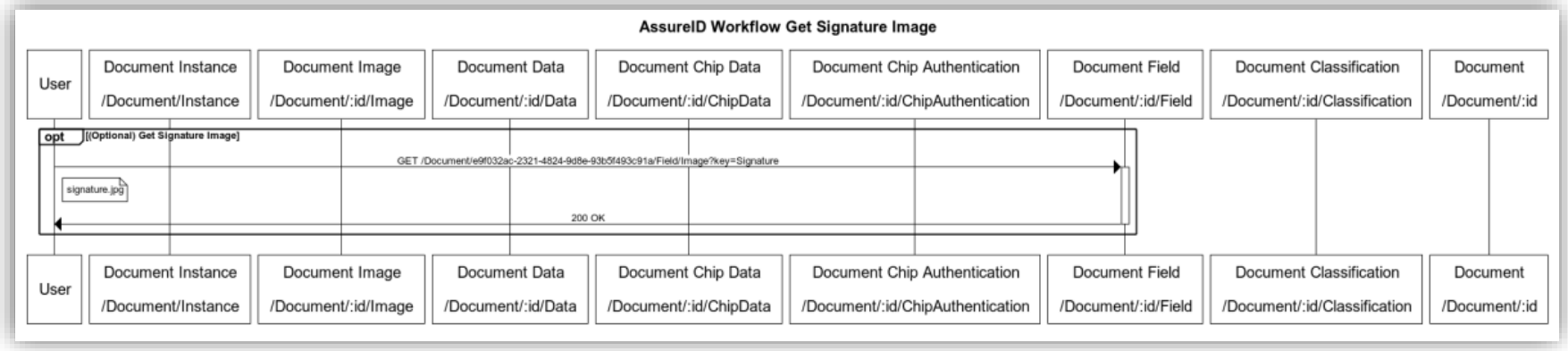
Get Document Component results

Getting the document component results must be done after getting the document result block is complete, and may be done in parallel.

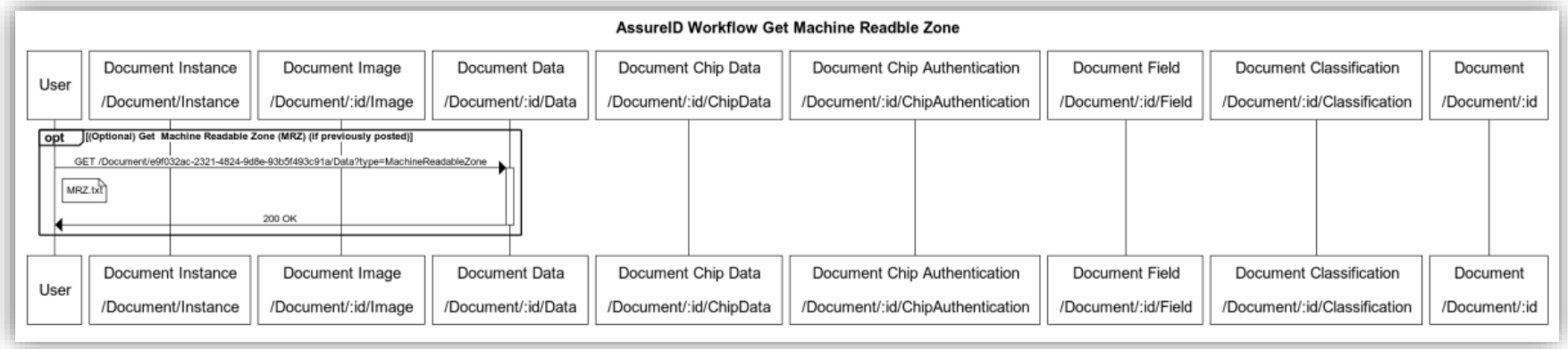
Get Photo image *(optional)*



Get Signature image *(optional)*

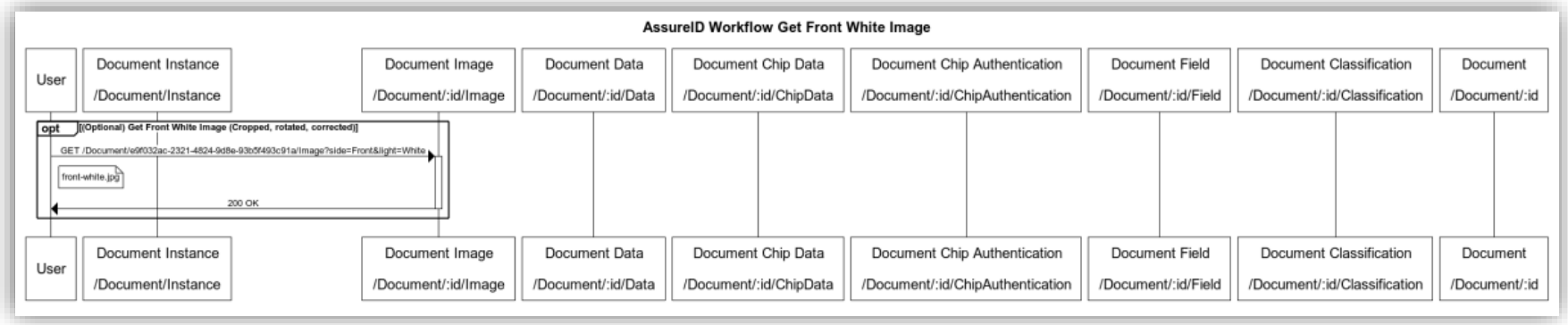


Get Machine Readable Zone* (MRZ) *(optional)*



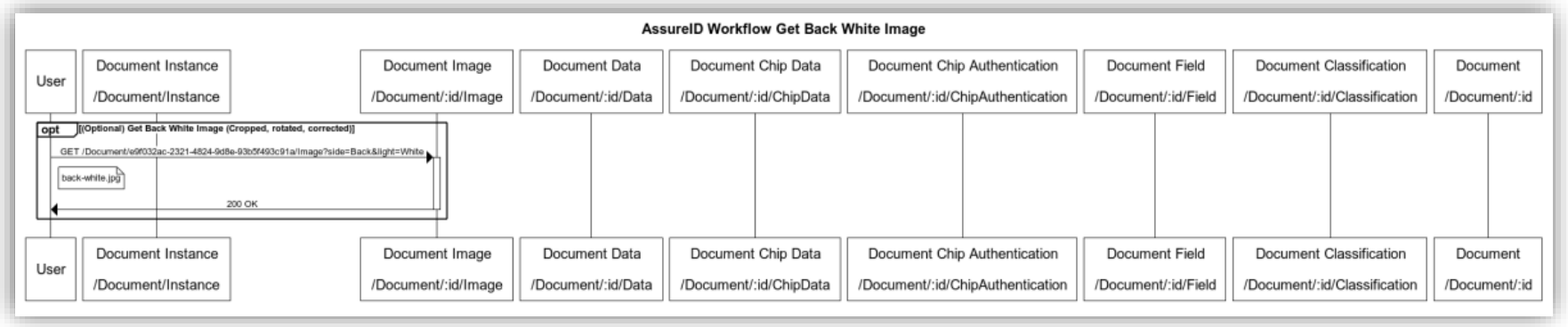
*If previously posted

Get Front White image* *(optional)*



*Cropped, rotated, and corrected

Get Back White image* *(optional)*



*Cropped, rotated, and corrected

Other Fields

The AssureID Connect API has several additional field images that can be obtained. These include **Fingerprint, VIZ Photo, VIZ Signature,** and **VIZ Fingerprint**. The request format mirrors that of the photo or signature above; only the key changes.

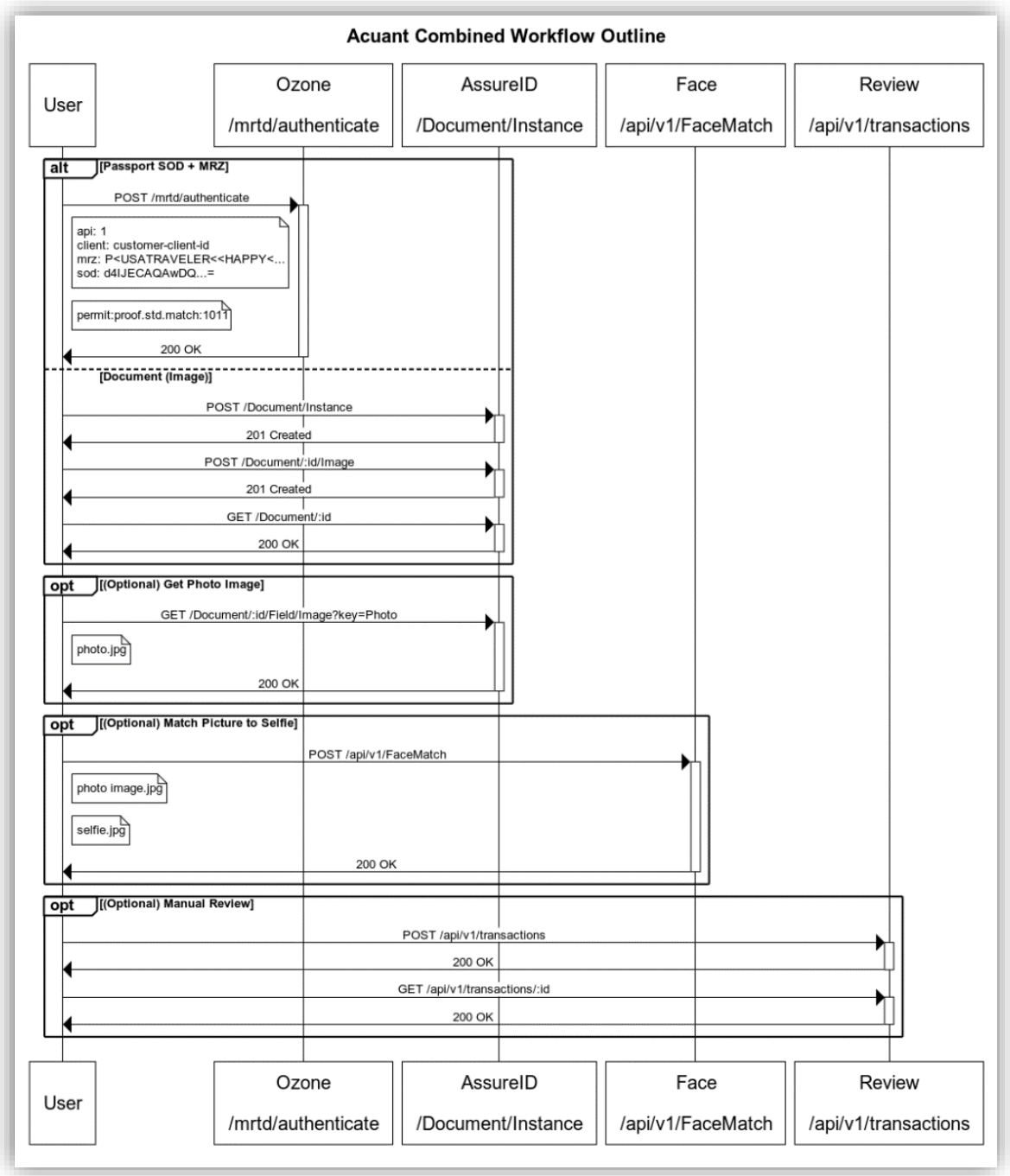
Delete Document

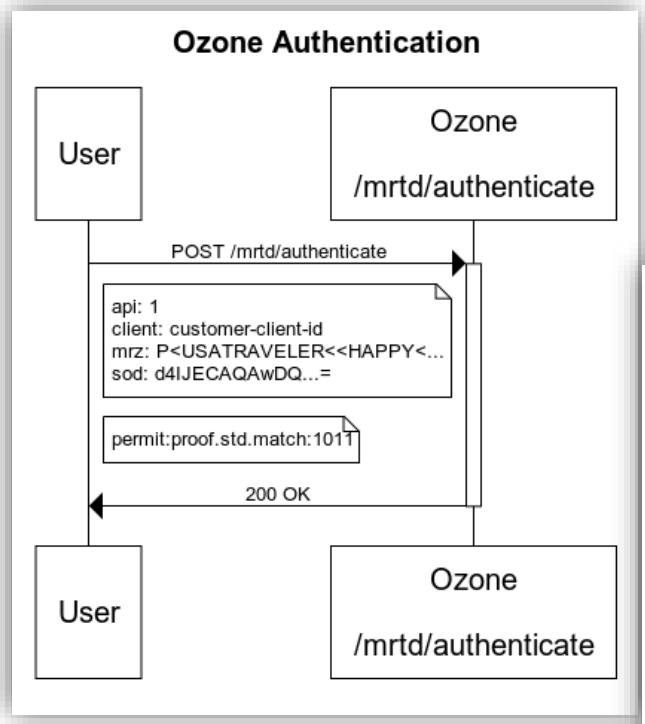
Deleting the document must be done after all of the preceding blocks are complete. After deletion is completed, all subsequent requests for the document will return **404 - Not Found**. For subscriptions with:

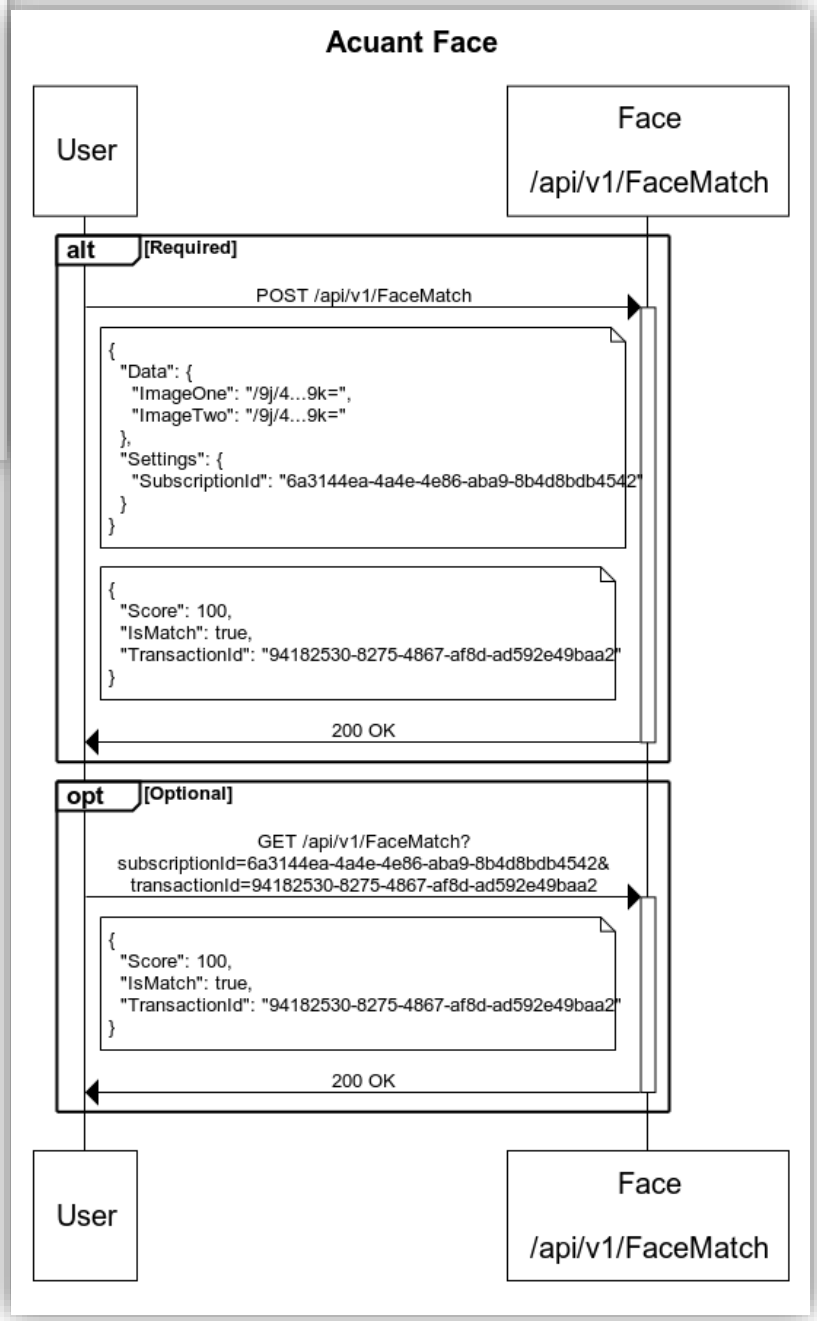
If StorePII =	then the results, images, and PII are retrievable:
true	Indefinitely or until the document is deleted via this API
false	For 60 seconds or until the document is deleted via this API

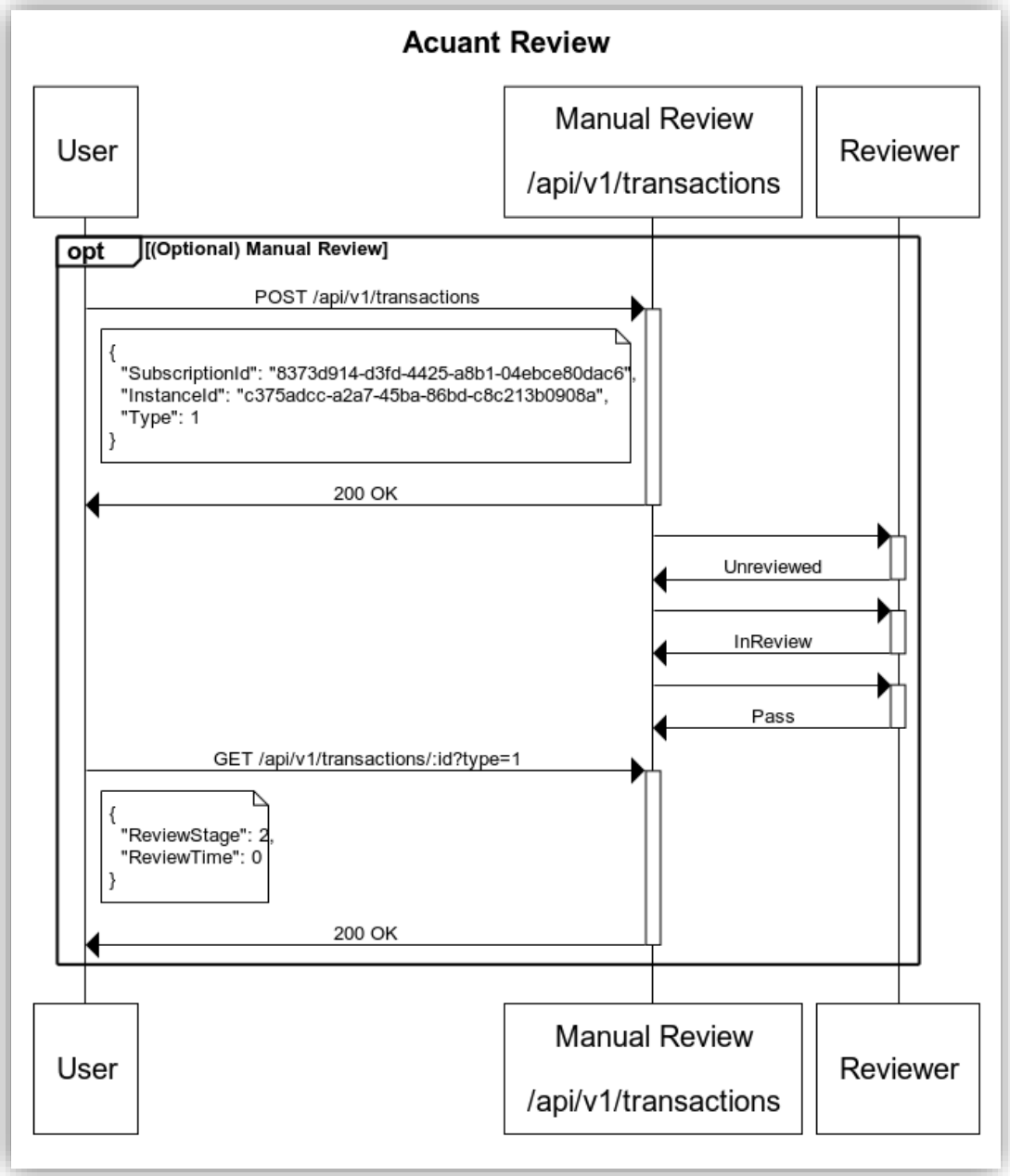
Acuant combined workflow

The following diagram shows the Acuant combined workflow, incorporating **Acuant Ozone**, **Acuant Face**, and **Acuant Review**.




Ozone Authentication
User
Ozone
/mrtd/authenticate
POST /mrtd/authenticate
api: 1
client: customer-client-id
mrz: P<USATRAVELER<<HAPPY<...
sod: d4IJECAQAwDQ...=
permit:proof.std.match:1011
200 OK
User
Ozone
/mrtd/authenticate


Acuant Face
User
Face
/api/v1/FaceMatch
alt
[Required]
POST /api/v1/FaceMatch
{
"Data": {
"ImageOne": "/9j/4...9k=",
"ImageTwo": "/9j/4...9k="
},
"Settings": {
"SubscriptionId": "6a3144ea-4a4e-4e86-aba9-8b4d8bdb4542"
}
}
{
"Score": 100,
"IsMatch": true,
"TransactionId": "94182530-8275-4867-af8d-ad592e49baa2"
}
200 OK
opt
[Optional]
GET /api/v1/FaceMatch?
subscriptionId=6a3144ea-4a4e-4e86-aba9-8b4d8bdb4542&
transactionId=94182530-8275-4867-af8d-ad592e49baa2
{
"Score": 100,
"IsMatch": true,
"TransactionId": "94182530-8275-4867-af8d-ad592e49baa2"
}
200 OK
User
Face
/api/v1/FaceMatch


Acuant Review
User
Manual Review
/api/v1/transactions
Reviewer
opt
[(Optional) Manual Review]
POST /api/v1/transactions
{
"SubscriptionId": "8373d914-d3fd-4425-a8b1-04ebce80dac6",
"InstanceId": "c375adcc-a2a7-45ba-86bd-c8c213b0908a",
"Type": 1
}
200 OK
Unreviewed
InReview
Pass
GET /api/v1/transactions/:id?type=1
{
"ReviewStage": 2,
"ReviewTime": 0
}
200 OK
User
Manual Review
/api/v1/transactions
Reviewer