

WebServices
identity Trust Management AG
Customer Services



API Reference
Version 2.10c

Index

1	Changelog	3
2	General Information	5
2.1	Contact and Credentials for Service Integration	5
2.2	Basic definitions	5
2.3	Typical flow	5
2.4	Redirects	6
2.5	Webhooks	6
2.6	Technical Data	7
3	Handling orders	8
3.1	Creating Orders	8
3.2	Checking the status of orders	16
3.3	Retrieving identification results in JSON format.....	21
3.4	Retrieving identification results as a PDF.....	25
3.5	Retrieving signed documents	27
3.6	Retrieving signing hashes	28
3.7	Requesting video file provisioning	29
3.8	Retrieving video binary files after provision	30
3.9	Retrieving audio binary files	31
3.10	Deleting identification data.....	32
4	Working with incomplete Orders	33
4.1	Setting up a Time Out	33
4.2	Retrieving full status information for all orders	33
4.3	Cancelling orders	36
5	Customer-side eSign processing	37
5.1	Process order and start signing.....	37
5.2	Confirm signing	38
5.3	Request a new TAN	39
6	Re-using identifications to sign documents	40
7	Service Monitoring.....	41
7.1	Checking liveness	41
8	Additional functions	42
8.1	Requesting the list of banks that support identity Giro	42
8.2	Checking availability of designated video agents	43
8.3	Checking sign-me user status.....	43
9	Appendix	45
9.1	IP-Ranges currently used	45
9.2	Supported browsers.....	45
9.3	Technical requirements for Video Ident	45
9.4	Simplified flow of a video identification	46
9.5	Simplified Video + eSign flow	47

1 Changelog

Version	Date	Editor	Changes
2.0	2017-03-06	Markus Herzog	initial new Version
	2017-03-16	Markus Herzog	New Method getVoiceFiles
2.01	2017-04-26	Markus Herzog	New Method getVideoQueueStatus Added Status 89 to ExtendedStatus List
2.02	2017-05-17	Markus Herzog	Added Object RedirectURL on putOrder Changed putOrder.Add possible Options
2.03	2017-05-30	Markus Herzog	New Method getShops
2.031	2017-07-31	Markus Herzog	New Method getVideoFileBinary Added Object WebhookURL on putOrder
2.032	2017-08-18	Markus Herzog	New Method getESignHash
2.032b	2017-11-20	Markus Herzog	Added new Type for Additional Field Country ISO on putOrder
2.032c	2018-02-12	Markus Herzog	Error in Definition getIdentData->IDCard->Type corrected
2.032d	2018-03-20	Markus Herzog	Added Object eSignHash on putOrder
2.04	2018-05-28	Markus Herzog	New Method checkSignmeUser
2.05		Markus Herzog	Added Product AutoID
2.06	2018-10-04	Erkan Ciftci	Removed no longer supported endpoints; removed deprecated putOrder elements; added several descriptions and sign-me features
2.06b	2018-11-29	Erkan Ciftci	Minor corrections and clarifications
2.07	2018-10-09	Markus Herzog	new endpoint, added error codes on some errors
2.08	2019-01-04	Markus Herzog	Internal Processing changes
2.09	2019-01-21	Markus Herzog	Added Element AllowedProductSet to putOrder Added Field DateOfIssue to IdentData->IDCard
2.09a	2019-03-01	Erkan Ciftci	Approved Version
2.09b	2020-05-20	Timo Tegtmeier	Minor clarifications and extensions
2.09c	2020-06-20	Timo Tegtmeier	Added information about e-mail notifications
2.10	2020-08-23	Timo Tegtmeier	Editorial overhaul, added new PoS functionality
2.10a	2020-08-24	Timo Tegtmeier	Added eID steering and mobile phone verification
2.10b	2020-09-08	Timo Tegtmeier	Added eidRestrictedIdentifier to getIdentData result

2.10c	2020-09-10	Timo Tegtmeier	Added information about WebHook retry scheme, added simplified flow diagrams for video identification and eSigning
--------------	------------	----------------	--

2 General Information

This document describes the usage of **identity.TM's** Customer API. Intended audience are developers and system integrators.

2.1 Contact and Credentials for Service Integration

Throughout the technical implementation you will be assisted by **identity.TM**. itsupport@identity.tm is the contact for any kind of technical and procedural questions. If you have not already received backend credentials for the services in this document, please apply for some by contacting **identity.TM**.

2.2 Basic definitions

User Natural person to be identified, also referred to as “signee” in eSign context
OrderID **identity.TM's** internal unique reference to an order

Datatype Codes

O	Object
A	Array
DS	Date string, with format YYYY-MM-DD
DTS	Date time string according to ISO 8061 (i.e. 2013-07-15T18:00:00+02:00)
S	String
N	Number
B	Boolean
PN	International phone number without blanks (i.e. +491701234567) (MSISDN)
CC	The ISO-Code of the country (ISO 3166) e.g. DE, AT, NL

Abbreviations

Len	Length
Type	Datatype, abbreviated
Req	Element is required
Occ	Element occurs
Y	Yes
N	No
Dep	Depends on parameters/conditions

2.3 Typical flow

The identification of a natural person is started by creating an identification order. As a result, a URL will be returned. Typically, the user will now be redirected to that URL to choose between different identification methods (dubbed *Verfahrensseite* or landing page).

The user will then carry out the identification. In the meantime, you can lean back and wait for a result to come.

After the identification has been conducted, the user will typically be redirected to you (in the rare case that no redirect is desired, the user will be redirected to a friendly page on **identity.TM** side that thanks him for conducting the identification).

Once a final judgment on the result of an identification has been made, you will be notified asynchronously via a webhook, that a result is now available. You would then make API calls to query the status of that specific identification order and query the identification data, if available. Identification data will be available in case the identification has been conducted positively or the identification is suspicious to fraud.

Once all data has been retrieved, you will then make an API call to delete the identification data.

In case the user does not finish an identification, a configurable time out will occur, that invalidates and deletes the order. A webhook will be triggered to indicate that condition.

2.4 Redirects

By default, the user will be redirected to the designated URL when the identification is finished from an end-user perspective. Keep in mind, that this moment will most probably not be the moment in which the order is finally processed, so no result data might be available. You must wait for the `OnFinal` webhook to be triggered, before retrieving results.

Alternatively, your account can be configured to keep the user on a spinner page on **identity.TM** side until the order is completed. In this case, data will be available when the redirect happens. Please contact itsupport@identity.tm if you would like to have such a spinner or more information on this topic.

2.5 Webhooks

Like the redirect URLs, the webhook URLs can also be set up within each `putOrder` request (`WebhookURL` element) that is being made. If these URLs do not change for each order, it is possible to configure them on a per account level.

The placeholders `[Ref]` and/or `[OrderID]` can be used to make the URL dynamic. `[Ref]` will be replaced by your internal reference number as specified when creating the order while `[OrderID]` will be replaced with **identity.TM's** unique OrderID of the order.

By default, Webhooks will be GET requests. This can be changed on a per account level to either POST or PUT if necessary.

2.5.1 Recommendations

Do not try to run longer operations in the code that receives the webhook. Webhooks will be executed synchronously and further order processing will be blocked until a webhook call has been finished. The operation should not take longer than 5 seconds and the webhook call will be terminated after 10 seconds.

2.5.2 Retry scheme

In case a webhook is not replied with a HTTP status code 200 or 202, the system will try to deliver the webhook again. The first retry will be started after approximately 86,4 seconds after the first try.

From	To	Interval
86.4s	864s	Once every minute
864s	12h	Once every hour
12h	6d	Once at night

2.5.3 Example URLs

```
https://www.example.com/identity/callback/AfterVideo?orderId=[OrderID]&ref=[Ref]  
https://www.example.com/identity/callback/OnFinal?orderId=[OrderID]&ref=[Ref]
```

2.6 Technical Data

Service Kind

RESTful Webservice over HTTPS (HTTP 1.1 Status Codes)
JSON Data in HTTP Body

API Base URL

https://customer.identity.tm/api/2.10/

Content-Type

application/json

Authentication

HTTP Basic Authentication
with username (CustomerID) and password (CustomerCode)

TLS

Version 1.2

3 Handling orders

3.1 Creating Orders

This central method is used to create any kind of new order for identifications and/or qualified electronic signatures for natural persons.

3.1.1 Requirements

The requested products must be associated with and enabled on your account by **identity.TM**.

3.1.2 Method and URL

| **[PUT]** <https://customer.identity.tm/api/2.10/putOrder>

3.1.3 JSON Request Body

Element	Parent	Type	Len	Req	Description
Ref	[ROOT]	S	100	N	Your internal reference number to the order. This field has no meaning for identity and can be used to associate an order with an ID on your systems.
Product	[ROOT]	N		Y	Specifies the product 0 = Identification excluding Video and autoID methods 12 = Identification including Video method, but excluding autoID 15 = eSign 16 = autoID
Add	[ROOT]	N		Y	Specifies certain additional product behaviors. 0 will be the correct choice in most cases. 0 = normal behavior 8 = if sign-me: generate account 16 = if sign-me: use exist. account 32 = no acquisition by phone 4096 = License ID Check 8192 = License Check (without ID) 131072 = Record Video

					4194304 = PoS Service
eIDSettings	[ROOT]	O		N	Settings for eID, no effect if user selects different method
VerifyMobileNumber	eIDSettings	B		N	Enables mobile phone number verification
EnableOCR	eIDSettings	B		N	Enables OCR to gather document number, issuing authority and date of issue of identification document.
CaptureImages	eIDSettings	B		N	Capture images of the legitimation document; if EnableOCR is set, images will always be taken
NoVideoFallback	eIDSettings	B		N	If eID fails, no fallback to Video will be offered.
getTargetLink	[ROOT]	B		N	Set to true to get the TargetURL in response (otherwise only the user will be informed via e-mail)
getVideoHashes	[ROOT]	B		N	Set to true to get the video system hash in the response if Product is 12 or 15.
getGiroLink	[ROOT]	S		N	IBAN or BIC of supported Bank to get giro redirect url
RedirectURL	[ROOT]	O		N	
Success	RedirectURL	S	150	N	Redirect to URL after success
Abort	RedirectURL	S	150	N	Redirect to URL after failure
WebhookURL	[ROOT]	O		N	
AfterVideo	WebhookURL	S	150	N	Webhook to trigger when the user leaves the video system. Cannot be used together with AfterIdent. Ident data will not necessarily be ready when triggered.
AfterIdent	WebhookURL	S	150	N	Webhook to trigger when the identification has been conducted. Cannot be used together with AfterVideo. Ident

					data will not necessarily be ready when triggered.
OnFinal	WebhookURL	S	150	N	Webhook to trigger when the order has been finally processed
Phone	[ROOT]	PN	50	N	Phone number of your customer
Mobile	[ROOT]	PN	50	N	Mobile number of your customer
Email	[ROOT]	S	50	Y	Email address of your customer
Address	[ROOT]	O		Y	Mandatory if user is allowed to use products Courier or Shop
Firstname	Address	S	50	Y	Physical Address data of your customer (Contact / Delivery)
Surname	Address	S	50	Y	
CompanyName	Address	S	50	N	
Street	Address	S	50	N	
ZipCode	Address	S	5	N	
City	Address	S	100	N	
Country	Address	S	3	Y	The ISO-Code of the country (ISO 3166) e.g. DE, AT, NL
IdentData	[ROOT]	O		N	The fields you provide, will be checked. Empty fields will be picked if possible.
Firstname	IdentData	S	50	Dep	Please give us notice if you plan to omit any of these fields.
Surname	IdentData	S	50	Dep	
Street	IdentData	S	100	Dep	
ZipCode	IdentData	S	5	Dep	
City	IdentData	S	100	Dep	
Country	IdentData	S	2	Dep	The ISO-Code of the country (ISO 3166) e.g. DE, AT, NL
Birthdate	IdentData	DS		Dep	Date of birth
Birthplace	IdentData	S	100	Dep	Place of birth
Birthname	IdentData	S	75	N	Surname at birth. Will only be determined, if field is neither null nor omitted
Nationality	IdentData	S	50	Dep	Nationality

Title	IdentData	S	50	Dep	Title
Artistname	IdentData	S	50	Dep	Artist name / alias
Gender	IdentData	S		Dep	female/male
Email	IdentData	S	50	N	Mobile number of your customer
Mobile	IdentData	PN	50	N	Email address of your customer
CollectedBy	IdentData	S	100	Dep	PoS Service only
IDCard	IdentData	O		No	Data about the ID document
CardNo	IDCard	S	50	Dep	ID document number
PlaceOfIssue	IDCard	S	100	Dep	ID document place of issue
DateOfIssue	IDCard	DS		Dep	ID document date of issue
ValidUntil	IDCard	DS		Dep	ID document valid until
Images	IDCard	O		Dep	PoS Service only
Front	Images	S		Dep	Base64 encoded JPEG image of the front side of ID Card
Back	Images	S		Dep	Base64 encoded JPEG image of the front side of ID Card
DriverLicence	IdentData	O		Dep	If Driver's License Check with Add 4096 and/or 8192, provide data if available
LicenceNo	DriverLicence	S	50	Dep	
Classes	DriverLicence	S	50	Dep	
DateOfIssue	DriverLicence	DS		Dep	
Additional	IdentData	O		N	If you need special attributes, you can add here for Courier/Shop up to 2, Video up to 10 items. <detail> can have a length of 50
<detail>	Additional	O		N	
Type	<detail>	N			1 = String, 2 = Date, 3 = Bool, 4 = Country ISO 3166
Label	<detail>	S	100		Readable label for the element
Value	<detail>	S	150		

LegitimationDocument	[ROOT]	A		N	Individual legitimation documents in case default is not sufficient
Document	LegitimationDocument	A	50		The readable description shown to the agent / employee processing the order
eSignPDF	[ROOT]	O/A		N	Document(s) designated for QES, use array for multiple documents.
Data	eSignPDF	S		N	Base64 encoded PDF (PDF must be unprotected)
Filename	eSignPDF	S	50	N	Filename
SignaturePosX	eSignPDF	N		N	The units are PDF Points equivalent to 1/72 inch. The lower/left corner of the document is the origin X=0 and Y=0.
SignaturePosY	eSignPDF	N		N	
SignatureHeight	eSignPDF	N		N	Only applicable if SignaturePos is also set. The units are PDF Points equivalent to 1/72 inch.
SignatureWidth	eSignPDF	N		N	
SignaturePage	eSignPDF	N		N	Page to place the signature on
SignatureFieldName	eSignPDF	S		N	Use the given predefined signature field in the PDF
eSignHash	[ROOT]	O/A		N	
Hash	eSignHash	S		N	Base64 Hash of prepared PDF
Filename	eSignHash	S	50	N	Filename
SignaturePosX	eSignHash	N		N	The units are PDF Points equivalent to 1/72 inch. The lower/left corner of the document is the origin X=0 and Y=0.
SignaturePosY	eSignHash	N		N	
SignatureHeight	eSignHash	N		N	Only applicable if SignaturePos is also set. The units are PDF Points equivalent to 1/72 inch.
SignatureWidth	eSignHash	N		N	
MultiSignOptions	[ROOT]	O		N	

FirstOrderID	MultiSignOptions	S	14		To reference an eSign order to the master. Contract from first Order will be signed by each.
SignParameters	MultiSignOptions	A		N	Signature parameters for file(s) given with the first order
Filename	SignParameters	S	50	N	Filename given with first order
SignaturePosX	eSignPDF	N		N	The units are PDF Points equivalent to 1/72 inch. The lower/left corner of the document is the origin X=0 and Y=0.
SignaturePosY	eSignPDF	N		N	
SignatureHeight	eSignPDF	N		N	Only applicable if SignaturePos is also set. The units are PDF Points equivalent to 1/72 inch.
SignatureWidth	eSignPDF	N		N	
SignaturePage	eSignPDF	N		N	Page to place the signature on
SignatureFieldName	eSignPDF	S		N	Use the given predefined signature field in the PDF
IdentDataSigned	[ROOT]	S		N	Previously generated identification data for signing purposes
EmailNotifications	[ROOT]	O		N	Send final processing mails, overrides customer settings
OnPositiveIdent	EmailNotifications	S	150	N	Send a mail to this address in case of a final positive status
OnNegativeIdent	EmailNotifications	S	150	N	Send a mail to this address in case of a final negative status
OnTimeOut	EmailNotifications	S	150	N	Send a mail to this address in case of a final timeout status
OnLeave	EmailNotifications	S	150	N	Send a mail to this address in case the user leaves the identification process

3.1.4 Response Data

Element	Parent	Type	Len	Occ	Description
OrderID	[ROOT]	S	20	Y	Our internal ID of the Order, always unique
Ref	[ROOT]	S	100	Y	Your internal ID of the Order, can be set multiple times
TargetURL	[ROOT]	S	100	Dep	URL to redirect your customer
VideoHashes	[ROOT]	O		Dep	
ShortCode	VideoHashes	S	5	Dep	Reference number to start Video identification in identity App on iOS/Android
LongCode	VideoHashes	S	20	Dep	
Error	[ROOT]	S		Dep	In case of an error

3.1.5 HTTP Codes

- 202 Order accepted, JSON document in body
- 400 Bad request, error description in body
- 401 Authentication failed, please check username and password

3.1.6 Example Request

```
{
  "Ref": "Sign-me_Account_Creation_Test",
  "Product": 15,
  "Add": 8,
  "getTargetLink": true,
  "getVideoHashes": true,
  "WebhookURL": {
    "OnFinal": "https://webhook.site/1234567_890?OnFinal=[OrderID]"
  },
  "RedirectURL": {
    "Success": "https://www.example.com/redirect/positive?OrderID=[OrderID]",
    "Abort": "https://www.example.com/redirect/negative?OrderID=[OrderID]"
  },
  "Email": "e.mustermann@example.com ",
  "Mobile": "+491711234567",
  "Address": {
    "Firstname": "Erika",
    "Surname": "Mustermann",
    "Street": "Heidestr. 17",
    "ZipCode": "43000",
    "City": "Köln",
    "Country": "DE"
  },
  "IdentData": {
    "Firstname": "Erika",
    "Surname": "Mustermann",
    "Street": "Heidestr. 17",
    "ZipCode": "43000",
    "City": "Köln",
    "Birthdate": "1964-08-12",
    "Birthplace": "Berlin",
    "Additional": {
      "idCardDateOfIssue": {
        "Type": 2,
        "Label": "Date of Issue",
        "Value": "2010-01-11"
      }
    }
  }
}
```

3.1.7 Example Response

```
{
  "OrderID": "99921269855041",
  "Ref": "Just_a_test",
  "TargetURL": "https://www.identity.tm/status/37C4B60A72A1C6C74FC038282CED500B",
  "VideoHashes": {
    "ShortCode": "SHU8T",
    "LongCode": "a9PPz-eFcNBifsk--bNJ"
  }
}
```

3.2 Checking the status of orders

When an order has been processed, it is of course necessary to check the final status of the order. This usually happens after the OnFinal webhook.

Regarding the status codes, we differentiate between *final* and *non-final* codes. All codes not listed in chapter 3.2.1.1 and chapter 3.2.1.2 are non-final. Furthermore, we differentiate between *general* and *extended* status code. General status codes will be returned in any variant of the `getStatus` functions, while the extended ones need to be specifically requested.

3.2.1 List of status codes

3.2.1.1 Final status codes for orders without QES

Status Code	Description
6	Positive identification
17	Negative identification
22	Manually terminated
89	Fraud Suspicion
99	TimeOut

3.2.1.2 Final status codes for orders with QES

Status Code	Description	ESignPDF/Hash Availability
84	Document(s) signed	Yes
17	Negative identification	No
22	Manually terminated	No
89	Fraud Suspicion	No
99	TimeOut	No
100	TimeOut after identification	No

3.2.1.3 Full List of all status codes

Status codes formatted **bold** will be returned by the default variant of `getStatus` and `getAllStatus`. To get all status codes an order has ever had, you need to call the `ExtendedList` variant of the respective function.

Kind	Definition EN Long	Definition DE Short
3	Order was loaded by courier	Beladen
6	Identification positively conducted	Zustellung
7	E-mail has been sent	EMail
9	Order on the way to the central HUB	Ausgang Depot
10	Order resides in the depot	Eingang Depot
12	Customer could not be reached by phone	TNE
14	Document sent to client	Retour Z
15	Customer was not present at appointment and was notified	Na
16	Identification refused by customer	Av
17	Identification negatively conducted	In
19	Appointment conducted with customer	Tv
20	Incorrect address in order	FA
21	The processing was aborted	Abbruch
22	The processing was terminated	Retour
23	Document verified	Vertrag OK
24	Document erroneous in postprocessing	Vertrag fehlerhaft
25	Order initialized	Daten erfasst
26	Follow-up scheduling	TV Wiedervorlage
33	Giro Forwarding	Giro Weiterleitung
34	Giro Feedback	Giro - Timeout
49	eID IDapp link created	eID Ausweisapp Link gebaut
50	eID IDapp request	eID Ausweisapp Request
51	eID data gathered	eID Daten erfasst
54	Support	Kundenanfrage bearbeitet

55	Customer could not be reached by phone	2xTNE
56	Phone number is incorrect	Datenkorrektur
57	Identification not possible, customer was repeatedly not present at appointments	AND
58	Identification refused by customer by phone	telefonischer AV
63	Cancellation	Storno
64	Appointment cancellation	TV Storno
66	Order is physically on the way to client	RetourZ Sendung
68	Change of procedure to Shop	Verfahrenswechsel zu Shop
69	Change of procedure to Home (Courier)	Verfahrenswechsel zu Home
70	Spoken on answering machine	Anrufbeantworter besprochen
71	SMS sent	SMS gesendet
73	Change of procedure to Video	Verfahrenswechsel zu Video
74	Change of procedure to eiD	Verfahrenswechsel zu eiD
75	Video identification conducted	Video Ident durchgeführt
77	Video identification in waiting room	Video Ident im Warteraum
78	Video identification call has begun	Video Ident Call begonnen
79	Video identification call has been left	Video Ident Call verlassen
80	Identification data ready for pick-up	Identdaten abholbereit
81	Change of procedure to Giro	Verfahrenswechsel zu Giro
83	Document signature request	Dokument SignaturRequest
84	Document signed	Dokument signiert
89	Fraud attempt by customer	Betrugsverdacht
94	eSign Account created	eSign Konto angelegt
95	eSign Account verified	eSign Konto verifiziert
96	eSign Account already exists	eSign Konto ist schon existent
97	Images were sent to Video	Images an Videosystem
98	Change of procedure to AutoID	Verfahrenswechsel zu AutoID

99	TimeOut	TimeOut
100	TimeOut after Ident	TimeOut nach Ident
101	eSign documents verified and hashes generated	eSign Dokumente bestätigt/Hash erzeugt
102	eSign document preview	eSign Dokumente Preview

3.2.2 Method and URL

[GET] <https://customer.identity.tm/api/2.10/getStatus/OrderID>
 [GET] <https://customer.identity.tm/api/2.10/getStatus/OrderID/ExtendedList>

3.2.3 Variants

Variant	Description
<i>Default</i>	Return only general status codes
ExtendedList	Returned extended status codes too

3.2.4 Response Data

Element	Parent	Type	Len	Occ	Description
OrderID	[ROOT]	S	20	Y	Requested OrderID
Ref	[ROOT]	S	100	Y	Corresponding Ref
Executed	[ROOT]	DTS		Y	Timestamp at which the request has been executed
Status	[ROOT]	A		Y	Array of status codes
Kind	Status	N		Y	Status code
Time	Status	DTS		Y	Timestamp at which status has been set
Text	Status	S		Dep	Explanatory text

3.2.5 HTTP Codes

200 OK, JSON document in body
 400 Bad request, error description in body
 401 Authentication failed, please check username and password
 404 Order not found

3.2.6 Example Request

[GET] <https://customer.identity.tm/api/2.0/getStatus/99921269855041/ExtendedList>

3.2.7 Example Response

```
{
  "OrderID": "99921269855041",
  "Ref": "Sign-me_Account_Creation_Test",
  "Executed": "2018-09-05T11:06:21+02:00",
  "Status": [
    {
      "Kind": 25,
      "Time": "2018-09-05T10:53:59+02:00"
    },
    {
      "Kind": 71,
      "Time": "2018-09-05T10:54:00+02:00",
      "Text": "signme Passwort"
    },
    {
      "Kind": 94,
      "Time": "2018-09-05T10:54:00+02:00"
    },
    {
      "Kind": 77,
      "Time": "2018-09-05T10:56:37+02:00"
    },
    {
      "Kind": 78,
      "Time": "2018-09-05T10:56:45+02:00"
    },
    {
      "Kind": 71,
      "Time": "2018-09-05T10:59:11+02:00",
      "Text": "TAN"
    },
    {
      "Kind": 75,
      "Time": "2018-09-05T11:06:21+02:00"
    },
    {
      "Kind": 6,
      "Time": "2018-09-05T11:06:21+02:00"
    },
    {
      "Kind": 23,
      "Time": "2018-09-05T11:06:21+02:00"
    },
    {
      "Kind": 71,
      "Time": "2018-09-05T10:59:11+02:00"
    },
    {
      "Kind": 95,
      "Time": "2018-09-05T11:09:51+02:00"
    }
  ]
}
```

3.3 Retrieving identification results in JSON format

The usual next step after an OnFinal webhook and the `getStatus` request is the `getIdentData` request with the subsequent logic.

If required, your account can be configured to return the data encrypted.

3.3.1 Requirements

The order must be finally processed. For a list of final status see chapters 3.2.1.1 and 3.2.1.2.

3.3.2 Method and URL

```
[GET] https://customer.identity.tm/api/2.10/getIdentData/OrderID
[GET] https://customer.identity.tm/api/2.10/getIdentData/OrderID/IncludeInitialData
[GET] https://customer.identity.tm/api/2.10/getIdentData/OrderID/IncludeIdentifyMethod
[GET] https://customer.identity.tm/api/2.10/getIdentData/OrderID/Signed
```

3.3.3 Variants

Variant	Description
Default	Normal variant, response as described below
IncludeInitialData	Include initial data as given to the <code>putOrder</code> request
IncludeIdentifyMethod	Include an additional element in the response "IdentifyMethod"
Signed	Return signed and encrypted version of the identification data for later re-use

3.3.4 Response Data

Element	Parent	Type	Len	Occ	Description
OrderID	[ROOT]	S	20	Y	Requested OrderID
Ref	[ROOT]	S	100	Y	Corresponding Ref
IdentifyMethod	[ROOT]	S		Dep	Method in which the identification occurred: Courier, Shop, Video, eID, Giro, SmartID, autoID
Identified	[ROOT]	DTS		Y	Time of identification The typo in the name is kept for backward compatibility
UserIp	[ROOT]	S		Dep	IP address of user while conducting the identification. Will only be collected and

					returned, if configured on your account.
IdentData	IdentData	O		Y	
Type	IdentData	N		Dep	1 = Checked Data 2 = Initial Data from putOrder
Firstname	IdentData	S	50	Y	
Surname	IdentData	S	50	Y	
Street	IdentData	S	100	Y	
ZipCode	IdentData	S	5	Y	
City	IdentData	S	100	Y	
Country	IdentData	S		Dep	
Birthdate	IdentData	DS		Y	
Birthplace	IdentData	S	100	Y	
Birthname	IdentData	S	75	Dep	Only if requested
Nationality	IdentData	S	50	Dep	
Title	IdentData	S	50	Dep	
Artistname	IdentData	S	50	Dep	
Gender	IdentData	S		Dep	female/male
Mobile	IdentData	S	50	Dep	If verified with TAN
Email	IdentData	S	100	Dep	If verified with TAN
IDCard	IdentData	O		Dep	
CardNo	IDCard	S	50	Dep	
eidRestrictedIdentifier	IDCard	S		Dep	Restricted Identifier (dienst- und kartenspezifisches Kennzeichen - DKK); only if eID was used to identify user
PlaceOfIssue	IDCard	S	100	Dep	
ValidUntil	IDCard	DS		Dep	
Type	IDCard	S		Dep	IDCard/Passport or LegitimationDocument provided in putOrder
DriverLicence	IdentData	O		Dep	If enabled in putOrder
LicenceNo	DriverLicence	S	50	Dep	If enabled in putOrder

Classes	DriverLicence	S	50	Dep	
DateOfIssue	DriverLicence	DS		Dep	
Additional	IdentData			Dep	If set in putOrder
<detail>	Additional	S	150	N	If set in putOrder
Images	IdentData	A		Dep	If enabled for this request
Image	Images	S			Base64 encoded Data
Filename	Image	S	100		Filename for the data

3.3.5 HTTP Codes

- 200 OK, JSON document in body
- 400 Bad request, error description in body
- 401 Authentication failed, please check username and password
- 404 Order not found

3.3.6 Example Request

| **[GET]** <https://customer.identity.tm/api/2.10/getIdentData/99921269855041/IncludeInitialData>

3.3.7 Example Response

```
{
  "Identified": "2018-09-05T11:06:21+02:00",
  "IdentData": [
    {
      "Type": 1,
      "Firstname": "Erika",
      "Surname": "Mustermann",
      "Street": "Heidestr. 17",
      "ZipCode": "43000",
      "City": "Köln",
      "Birthdate": "1964-08-12",
      "Birthplace": "Berlin",
      "Nationality": "DE",
      "Gender": "female",
      "Mobile": "+4915114084205",
      "IDCard": {
        "CardNo": "T22000129",
        "PlaceOfIssue": "Stadt Köln",
        "ValidUntil": "2020-10-31",
        "Type": "IDCard"
      }
    },
    {
      "Type": 2,
      "Firstname": "Erika",
      "Surname": "Mustermann",
      "Street": "Heidestr. 17",
      "ZipCode": "43000",
      "City": "Köln",
      "Birthdate": "1964-08-12",
      "Birthplace": "Berlin",
      "Nationality": "DE",
      "IDCard": {
        "CardNo": null,
        "PlaceOfIssue": null,
        "ValidUntil": "1970-01-01"
      }
    }
  ]
}
```


3.4 Retrieving identification results as a PDF

The same `getIdentData` logic as described in chapter 3.3 applies here too. The content is basically the same except that a PDF will be returned, and all images/screenshots are included by default.

3.4.1 Requirements

Same as `getIdentData`, additionally this endpoint must be enabled by **identity.TM**.

3.4.2 Method and URL

| [GET] <https://customer.identity.tm/api/2.10/getIdentDataPDF/OrderID>

3.4.3 HTTP Codes

- 200 OK, PDF in body
- 400 Bad request, error description in body
- 401 Authentication failed, please check username and password
- 404 Order not found

3.4.4 Example Request

| [GET] <https://customer.identity.tm/api/2.10/getIdentDataPDF/99921269855041>

3.4.5 Example Response

Resultat IDENTprüfung

Auftrag: 99921267754340

Referenz: Shop GWG

Angabe	Geliefert	Geprüft
Nachname	Mustermann	Mustermann
Vorname	Erika	Erika
Strasse	Heidestr. 17	Heidestr. 17
PLZ	51147	51147
Ort	Köln	Köln
Geburtsdatum	12.08.1964	12.08.1964
Geburtsort	Berlin	Berlin
Nationalität		DE
Künstlername		
Titel		
Ausweisnummer		T22000129
AusstellungsLand		
AusstellungsOrt		Stadt Köln
Gültig bis		31.10.2020

Lichtbild und Merkmale gemäß Ausweisdokument mit der zu identifizierenden Person verglichen und Identität festgestellt.

Resultat positiv am 03.09.2018 08:52
Identifizierer
Prüfer 1872

06.09.2018 16:10 (1/3)

3.5 Retrieving signed documents

Endpoint for fetching processed/signed eSign PDF documents

3.5.1 Requirements

eSignPDF element was set in putOrder and final status is 84 (see chapter for 3.2.1.2 status details).

3.5.2 Method and URL

| [GET] <https://customer.identity.tm/api/2.10/getESignPDF/OrderID>

3.5.3 Response Data

The response will either be a binary PDF in case the order only contains a single PDF, or a JSON documents otherwise.

Element	Parent	Type	Len	Occ	Description
PDFs	[ROOT]	A		Y	Array of documents
Base64Data	PDFs	S		Y	Base64 encoded PDF data

3.5.4 HTTP Codes

- 200 OK, PDF stream in body if only one document, otherwise a JSON document
- 400 Bad request, error description in body
- 401 Authentication failed, please check username and password
- 404 Order not found

3.6 Retrieving signing hashes

Endpoint for fetching processed/signed eSign hashes.

3.6.1 Requirements

eSignHash element was set in putOrder and final status is 84 (see chapter for 4.2.1.2 status details).

3.6.2 Method and URL

| [GET] <https://customer.identity.tm/api/2.10/getESignHash/OrderID>

3.6.3 Response Data

Element	Parent	Type	Len	Occ	Description
Hashes	[ROOT]	A		Y	Array of hashes
<Name>	Hashes	S		Y	Hash value, <Name> was given in putOrder request

3.6.4 HTTP Codes

- 200 OK, JSON document in body
- 401 Authentication failed, please check username and password
- 400 Bad request, error description in body
- 404 Order not found

3.7 Requesting video file provisioning

In previous versions of our API, one could download a video file of the identification process immediately after retrieving the identification data.

This behavior is no longer recommended, as specific products do not support that. It is therefore necessary to request video file provisioning with a call to this function first.

Once the video file is available, the given callback URL will be called with a GET request to indicate video file availability.

3.7.1 Requirements

The desired products must be associated with and enabled on your account by **identity.TM**.

3.7.2 Method and URL

[POST] <https://customer.identity.tm/api/2.10/getVideoFileBinaryAsync/OrderID>

3.7.3 JSON Request Body

Element	Parent	Type	Len	Req	Description
callbackUrl	[ROOT]	S		Y	Array of documents

3.7.4 HTTP Codes

- 200 OK
- 400 Bad request, error description in body
- 401 Authentication failed, please check username and password
- 404 Order not found

3.8 Retrieving video binary files after provision

This file is provided, when identification was conducted in video with a recording of the session from the agent's point of view.

3.8.1 Requirements

This file is provided, when final status from video is 6 or 89 and the recording requirement was set either in the Add element of the `putOrder` request or globally configured in the account for legal (e.g. German AML) compliance by **identity.TM**.

Furthermore, if you did not request provisioning of the file or did not wait for the callback to happen before calling this function, you might get a placeholder video instead.

3.8.2 Method and URL

| **[GET]** <https://customer.identity.tm/api/2.10/getVideoFileBinary/OrderID>

3.8.3 HTTP Codes

- 200 OK, binary video file (video/mp4 or video/webm) in body
- 400 Bad request, error description in body
- 401 Authentication failed, please check username and password
- 404 Order not found

3.9 Retrieving audio binary files

In some countries it is sufficient, to have an audio recording instead of a video file. In these cases, the configuration will be made by **identity.TM** for the given account. An audio recording cannot be requested with the putOrder request.

3.9.1 Method and URL

| [GET] <https://customer.identity.tm/api/2.10/getVoiceFiles/OrderID>

3.9.2 HTTP Codes

200 OK, JSON document in body
401 Authentication fails, please check username and password
400 Bad request, error description in body
404 Order not found

3.9.3 Example Request

| [GET] <https://customer.identity.tm/api/2.10/getVoiceFiles/99920006C38842>

3.9.4 Example Response

```
| {  
  "Voicefiles": [  
    { "Filename": "sound_4348.mp3", "Base64Data": "\\uQBAAAAA....." }  
  ]  
}
```

3.10 Deleting identification data

By default, **identity.TM** will delete all result files 7 days after fulfilment. Nevertheless, it would be best, to delete the result files after they have been collected completely and successfully.

Some confusion exists on when to use `delIdentData` and when to use `cancelOrder`. As a rule of thumb, use `cancelOrder` before the `OnFinal` webhook has been triggered and `delIdentData` after that point.

3.10.1 Method and URL

| **[DELETE]** `https://customer.identity.tm/api/2.10/delIdentdata/OrderID`

3.10.2 HTTP Codes

202 Accepted, data deleted
400 Bad request, error description in body
401 Authentication failed, please check username and password
404 Order not found or already deleted
406 Invalid order state

3.10.3 Example Request

| **[DELETE]** `https://customer.identity.tm/api/2.10/delIdentdata/99920006C38842`

4 Working with incomplete Orders

4.1 Setting up a Time Out

Please contact itsupport@identity.tm in order to set a general time out for incomplete orders.

4.2 Retrieving full status information for all orders

Endpoint to get all status information about all orders. It is possible to filter status data by sending an appropriate JSON document in the request body.

In case no period is specified, this function will only return information about orders that have not been reported before.

4.2.1 Method and URL

```
[POST] https://customer.identity.tm/api/2.10/getAllStatus  
[POST] https://customer.identity.tm/api/2.10/getAllStatus/ExtendedList
```

4.2.2 Optional JSON Request Body

Element	Parent	Type	Len	Req	Description
Period	[ROOT]	O		N	If this element exists, returned data will not be marked as already queried
Begin	Period	DTS		Dep	Status from
End	Period	DTS		Dep	Status until
Paging	Period	O		N	For Paging the result if Result >= 1000 Elements
Elements	Paging	N		N	<= 1000
Page	Paging	N		N	

4.2.3 Response Data

At max 1000 status elements will be returned at once.

Element	Parent	Type	Len	Occ	Description
Order	[ROOT]			Y	
OrderID	Order	S	20	Y	The OrderID
Ref	Order	S	100	Y	Your internal number to the order
Delivered	Order	DTS		Dep	DateTime if Order is delivered/executed ISO 8601
Status	Order	A		Y	
Kind	Status	N		Y	See Description of kind of status (at the end of this Document)
Time	Status	DTS		Y	ISO 8601
Text	Status	S	255	Dep	Hints to a status
TVSlot	Order	O		Dep	Current TV – just in Status 19
SlotBegin	TVSlot	DTS		Y	Begin of Slot ISO 8601
SlotEnd	TVSlot	DTS		Y	End of Slot ISO 8601

4.2.4 HTTP Codes

- 200 OK, JSON data in body
- 304 No new status information available since last call
- 400 Bad request, error information in body
- 401 Authentication failed

4.2.5 Example Request

```
{
  "ExtendedList": true,
  "Period": {
    "Begin": "2016-01-01T00:00:00+01:00",
    "End": "2016-01-15T00:00:00+01:00",
    "Paging": {
      "Elements": 200,
      "Page": 2
    }
  }
}
```

4.2.6 Response Example

```
[
  {
    "OrderID": "9992120505XXXX",
    "Ref": "TestOrder"
  },
  {
    "OrderID": "9992000BE9XXXX",
    "Ref": "Test",
    "Executed": "2018-05-28T16:47:00+02:00"
  },
  {
    "OrderID": "9992121170XXXX",
    "Ref": "TestOrder",
    "Executed": "2018-05-30T16:05:52+02:00"
  },
  {
    "OrderID": "9992121171XXXX",
    "Ref": "TestOrder",
    "Executed": "2018-05-30T16:08:52+02:00"
  },
  {
    "OrderID": "9992000BEBXXXX",
    "Ref": "ReferenceNo.12345",
    "Executed": "2018-06-01T09:01:00+02:00"
  }
]
```

4.3 Cancelling orders

If an open order shall not be processed anymore, you can cancel it with this endpoint. The order data will not be deleted, but the user will not be able to proceed with his identification or signature.

4.3.1 URL

| **[POST]** `https://customer.identity.tm/api/2.10/cancelOrder/OrderID`

4.3.2 HTTP Codes

202 Accepted
400 Bad request, error information in body
401 Authentication failed
404 Order not found
406 Order not in a cancellable state

4.3.3 Example Request

| **[POST]** `https://customer.identity.tm/api/2.10/cancelOrder/99920006C38842`

5 Customer-side eSign processing

All these functions need to be enabled for your account.

If you desire a deeper integration of the signing process into your corporate design and want to provide the necessary function on your side, these functions will enable to you carry out the whole eSign process.

5.1 Process order and start signing

Use this function to start the signing process. A TAN will be sent to the signee.

5.1.1 Requirements

This functionality must be enabled for your account.

5.1.2 Method and URL

| **[POST]** <https://customer.identity.tm/api/2.10/requestSign/OrderID>

5.1.3 HTTP Codes

202 Accepted
400 Bad request, error description in body
401 Authentication failed, please check username and password
404 Order not found or already deleted
406 Invalid order state
410 Order has already been processed

5.1.4 Example Request

| **[POST]** <https://customer.identity.tm/api/2.10/requestSign/99920006C38842>

5.2 Confirm signing

Sign the document. To do so, the TAN previously sent need to be provided.

5.2.1 Requirements

This functionality must be enabled for your account.

5.2.2 Method and URL

[POST] <https://customer.identity.tm/api/2.10/confirmSign/OrderID>

5.2.3 JSON Request Body

Element	Parent	Type	Len	Req	Description
tan	[ROOT]	S	6	Y	The TAN the user received.

5.2.4 HTTP Codes

- 202 Accepted
- 400 Bad request, error description in body
- 401 Authentication failed, please check username and password
- 404 Order not found or already deleted
- 410 Order has already been processed
- 412 Singing has not been started, call `requestSign` first
- 429 Too many tries with a wrong TAN, order canceled

5.2.5 Example Request

[POST] <https://customer.identity.tm/api/2.10/confirm/99920006C38842>

```
{  
  "tan": "123456"  
}
```

5.3 Request a new TAN

In case the TAN has not been received by the signee, you can request to send a new TAN to the signee.

5.3.1 Requirements

This functionality must be enabled for your account.

5.3.2 Method and URL

| **[POST]** <https://customer.identity.tm/api/2.10/requestResendSignTan/OrderID>

5.3.3 HTTP Codes

- 202 Accepted
- 400 Bad request, error description in body
- 401 Authentication failed, please check username and password
- 404 Order not found or already deleted
- 410 Order has already been processed
- 412 Singing has not been started, call `requestSign` first

5.3.4 Example Request

| **[POST]** <https://customer.identity.tm/api/2.10/requestResendSignTan/99920006C38842>

6 Re-using identifications to sign documents

If your account is eSign-enabled and is using Namirial, it is possible to store identification data and re-use it later to avoid a going through a whole new identification process.

To be able to do so, a two-step approach is used:

1. Retrieve and store a signed version of the identification data
2. Provide this identification data when placing a new order for the same user

To retrieve the signed version of the identification data you need to call the “Signed”-variant of `getIdentData` (`getIdentData/OrderID/Signed`) as described in chapter 3.3 and store the result body.

When you now want to sign a document and re-use the identification, you provide the stored data in the element `IdentDataSigned` inside the body of the `putOrder` call.

Before re-using the given data, it will be checked, if

- It has not been tampered and is signed correctly,
- The given data is sufficient to be used for signing purposes,
- The identity document used for the identification is still valid.

In case the data is considered valid for signing purposes, the user will immediately be redirected to the signing process without a further identification, otherwise the user will be re-identified.

7 Service Monitoring

7.1 Checking liveness

To check the liveness of our service, you can call a special function without side effects.

7.1.1 Method and URL

| **[GET]** <https://customer.identity.tm/api/2.10/serverStatus>

7.1.2 HTTP Codes

200 Service available
500 Service not available
None Service not available

8 Additional functions

8.1 Requesting the list of banks that support identity Giro

8.1.1 Method and URL

| [GET] <https://customer.identity.tm/api/2.10/getBankList>

8.1.2 Response Data

Root object is an array, information about the banks are returned in objects within that array.

Element	Parent	Type	Len	Req	Description
bic	[ROOT]	N	11	Y	BIC (Bank Identifier Code)
blz	[ROOT]	N	8	Y	German BLZ (Bank-Leit-Zahl)
name	[ROOT]	S	255	Y	Name of the bank

8.1.3 HTTP Codes

200 OK, JSON document in body

400 Bad request, error description in body

401 Authentication failed, please check username and password

8.1.4 Example Request

| [GET] <https://customer.identity.tm/api/2.10/getBankList>

8.1.5 Example Response

```
[
  {
    "bic": "WELADED1OPR",
    "blz": 16050202,
    "name": "Spk Ostprignitz-Ruppin"
  },
  {
    "bic": "NOLADE21RDB",
    "blz": 21450000,
    "name": "Spk Mittelholstein"
  },
  {
    "bic": "NOLADE21WHO",
    "blz": 22250020,
    "name": "Spk Westholstein"
  }
]
```

8.2 Checking availability of designated video agents

In case you are creating orders for video identifications and you wish to check prior how the availabilities of the designated agents are in a given moment, you can apply for a separate endpoint by contacting itsupport@identity.tm. This service may make sense if you are using different identity providers and would like to implement a load balancer between those.

8.3 Checking sign-me user status

This method is only applicable if you wish to create an eSign order for sign-me (D-Trust). Before creating the order, you can check prior the user status in the sign-me system in order to find out, if the given user already has a sign-me account and if he is able to sign with a qualified electronic signature (QES). Both elements in the response need to be true in order to create a plain signature Order (Add:16 in `putOrder`). Otherwise the account needs to be created and verified (Add:0 or 8 in `putOrder` – The Add value 0 represents the default setting with creation/verification of the account plus PDF document(s) that need(s) to be signed – With the Add value 8 it is the same with the exception of the PDF and signature part).

8.3.1 Requirements

Your account must be configured for eSign using sign-me.

8.3.2 Method and URL

| [POST] <https://customer.identity.tm/api/2.10/checkSignmeUser>

8.3.3 JSON Request Body

Element	Parent	Type	Len	Req	Description
Email	[ROOT]	String		Yes	The sign-me username (e-mail)

8.3.4 Response Data

Element	Parent	Type	Len	Occ	Description
UserExists	[ROOT]	B		Y	The sign-me account exists
UserIsAbleToSign	[ROOT]	B		Y	The sign-me account is qualified and able to sign

8.3.5 HTTP Codes

- 200 OK, JSON document in body
- 400 Bad request, error description in body
- 401 Authentication failed, please check username and password

8.3.6 Example Request

```
[POST] https://customer.identity.tm/api/2.10/checkSignmeUser
{
  "Email": "e.mustermann@example.com"
}
```

8.3.7 Example Response

```
{
  "UserExists": false,
  "UserIsAbleToSign": false
}
```

9 Appendix

9.1 IP-Ranges currently used

Range	Network name
212.211.189.136 - 212.211.189.143	ID8_NET
212.211.206.240 - 212.211.206.247	identity-NET
212.211.181.64 - 212.211.181.71	idvos_NET
212.211.191.96 - 212.211.191.103	ID8_NET

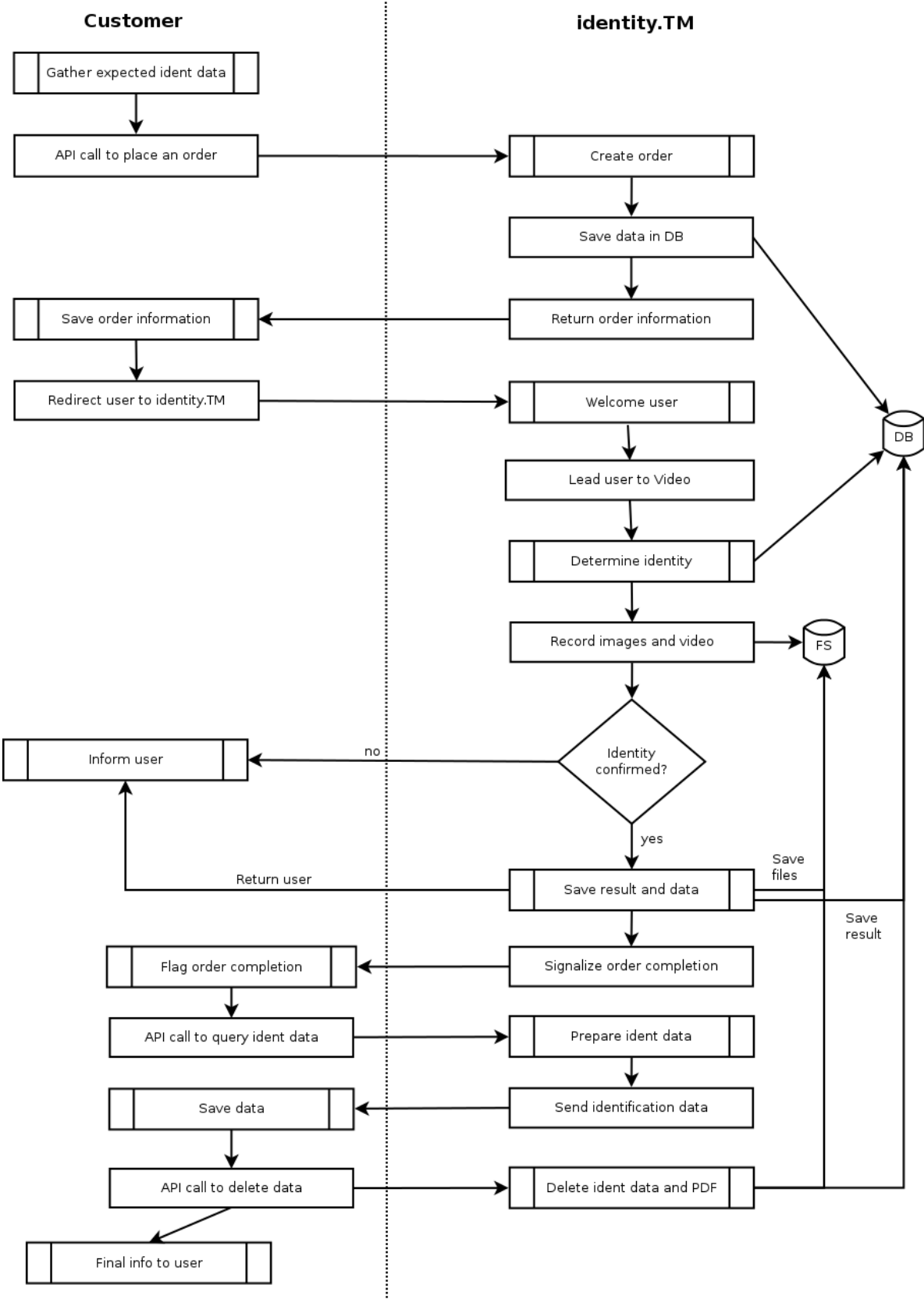
9.2 Supported browsers

- **Desktop:** Chrome, Opera, Firefox, Safari (macOS only) – latest two versions
- **Mobile:** Android, Chrome - latest versions (official)

9.3 Technical requirements for Video Ident

- A supported browser, alternatively mobile apps for Android and iOS can be used
- Bandwidth: Minimum 0,5 MB/s up/down
- Camera: Minimum resolution 640 x 480 px
- Network requirements:
 - Minimum: The minimum Requirement is that TCP port 443 is open. Some firewall/proxy rules only allow for SSL traffic over port 443. You will need to make sure that non-web traffic can also pass over this port. TLS1.2
 - Better Experience: In addition to the minimum requirements being met, we also recommend that UDP port 3478 is open. TLS1.2
 - Best Experience: For the best possible experience, we recommend that UDP ports 1025 - 65535 be open. TLS1.2
- WebRTC: Outbound TCP, non-SSL web traffic on port 443 and the following domains must be accessible:
 - *.tokbox.com (static IP blocks also available)
 - static.opentok.com
 - enterprise.opentok.com
 - api.opentok.com
 - anvil.opentok.com
- WebSocket: In some situations, WebSocket connections are blocked over port 80. In this case a secure SSL connection using WSS over port 443 should successfully connect. The destinations and ports used by Pusher clients are as follows:
 - ws://ws.pusherapp.com on port 80
 - wss://ws.pusherapp.com on port 443

9.4 Simplified flow of a video identification



9.5 Simplified Video + eSign flow

