BANCA D'ITALIA EUROSISTEMA

BANCO DE ESPAÑA Eurosistema

BANQUE DE FRANCE EUROSYSTÈME

DEUTSCHE BUNDESBANK EUROSYSTEM

T2S Non Repudiation of Origin (NRO)

Upgrade of non-repudiation service for U2A(CR-0722)

Technical Document

This document is outdated following the allocation of CR-722 to R6.0. Please refer instead to the 'ESMIG qualified configurations document available at https://www.ecb.europa.eu/paym/target/t2s/profuse/html/index.en.html

Author	4CB
Version	*1.8.5*
Date	*18/08/2021*
Status	final
Classification	
Accessible	
Classified until	

History of releases

RELEASE	DATE	ISSUES	STATUS[1]
1.0.	30/01/2015	First draft – as explained in PMG telco on 8 December 2014, relevant assessments parts, including NSP-related parts, are still ongoing and might result in noticeable changes	draft
1.0	28/02/2015	Final version – as explained in PMG telco on 8 December 2014, relevant assessments parts, including NSP-related parts, are still ongoing and might result in noticeable changes	final
1.1	09/03/2015	Updated version – results of ongoing assessment activities are included	final
1.2	27/03/2015	Updated version – issues raised during the PMG workshop on 11/03/2015 and respective clarifications are included	final
1.3	10/08/2015	Updated version – new qualified configuration.	
1.4	21/12/2015	Updated version – clarifications from user testing	
1.5	21/01/2016	Updated version – clarifications from user testing	
1.6	16/02/2017	Updated version – changes to recommended configurations	
1.7 – 1.7.1	28/11/2019	Updated version – details of Ascertia solution included (communication flows, customer impact/actions, security aspects)	
1.7.2	02/02/2020	Updated version – section 2.6.3. Recommended Configurations revised to reflect OMG/CRG comments	
1.7.3	13/02/2020	Updated version – included comments / answers as per Q&A file	
1.8.0	01/02/2021	Updated version – additional checks included (section 2.6.2)	
1.8.1	19/02/2021	Updated version – additional clarifications related to Citrix environment and Ascertia solutions	
1.8.2	04/03/2021	Added clarifications about gosign desktop client TLS certificate renewal procedure	
1.8.3	08/04/2021	Additional clarifications related to Citrix terminal server and Ascertia GoSign Desktop solution	
1.8.4	16/07/2021	Add section in the annex concerning the GSD multi-user solution	
1.8.5	20/08/2021	Minor corrections plus integrations to the annex related to the GSD multi-user solution	

Reference documents

REFERENCE		OBJECT
1	T2S-0466-BFD	CR for implementation of non-repudiation of origin for U2A
2	MOP	Manual of Operational Procedures V1.0

[1] Status value : Draft, Open, Final, Dismissb

1. Introduction

1.1. Context of the document

This document describes the general technical framework to help the customer to implement the requested non-repudiation of origin functionality (NRO) as enhanced security requirement. The content of the document is the guideline for the T2S participants and informs about the prerequisites to implement and use the NRO-functionality.

This document reflects the status at the time of preparation. Subsequent changes are possible, especially as relevant assessment parts are still ongoing.

After the deployment of CR-466 in T2S, the general URD requirement that "… a very high level of security is requested in terms of confidentiality, authentication, integrity, access control and non-repudiation of the T2S information" was eventually fulfilled so that its originator could not deny changes performed via the T2S Graphical User Interface (GUI).

After consultation of the various bodies (CRG, ISSG and Legal experts), the CSDs opted for the use of a digital signature for critical / user-selected transactions only via X.509 standard certificates stored in a secure USB device (e.g. USB token) or accessible via a secret PIN code.

Current solution is based on Java applet that are recalled by the web application screens via Java Script (embedded in the html pages). Applet allows calling the APIs of the USB token, which allows on turn the generation of the digital signature (PKCS#11 standard protocol).

After several years from its development, such technology (based on NPAPI - Netscape Plugin API plugin) has been considered not safe enough andwhile also showsing performance problems and considering in comparison with the possibilities offered by plug-in free technologies (e.g. HTML5).

This led to:

- Browser vendors to phase out NPAPI support starting from 2013; IE11 is the only browser still supporting Java applet, but for IE 11 the support is limited. For Windows 7 it will end 14/01/2020 and for Windows 10 in 14/10/2025;
- Oracle to announce that applet support may be removed any time in Java SE8 (as of March 2019) even if there are currently no plans to remove the components required to run applets. It

After a thorough investigation of the market solutions, it was decided to extend the adoption of the Ascertia solution (ADSS server + ADSS Go>Sign Desktop) that is currently used in T2 Internet Access and Contingency Network context and that has already proved to be reliable, stable and responsive from a performance point a view.

1.2. Definition of NRO

NRO provides the recipient with the evidence NRO which ensures that the originator will not be able to (i) deny having sent the U2A instruction; or (ii) claim having sent the U2A instruction with a different content.The evidence of origin is generated by the originator and held by the recipient.

1.3. Network Service Provider responsibility

The procedures related to the management of the certificate, signature and PIN - issuance, distribution, revocation, locking after failed attempts, credential management, etc. - are individual and specific to each network service provider (NSP) and the description can be found in the technical documentation attached to the contract signed by each Directly Connected Actor and the selected NSP. In general, the rules and procedures applicable to the authentication certificates provided by the selected NSP will apply also to the digital signing certificates. However, there will be no interference between both certificates.

The certificate to be used for the digital signature are issued by the NSP and are stored on the same device used for storing the user authentication certificate.. In any case, the authentication and the signing certificate will have the same DN as well as the same validity date.

Information regarding the availability of e-token and detailed procedures for handling e-tokens, being part of NSPs' responsibilities, will be provided by NSPs.

Furthermore, there will be the possibility to store signing/authentication certificate on remote HSM devices in order to replace use of USB/e-token and simplify certificates management. This is planned to be available as of June, 2022 according to NSP Tender. (Implementation details to be provided by NSPs accordingly). Readers Guide

The following matrix provides a consolidated overview of the responsibility of the T2S participants.

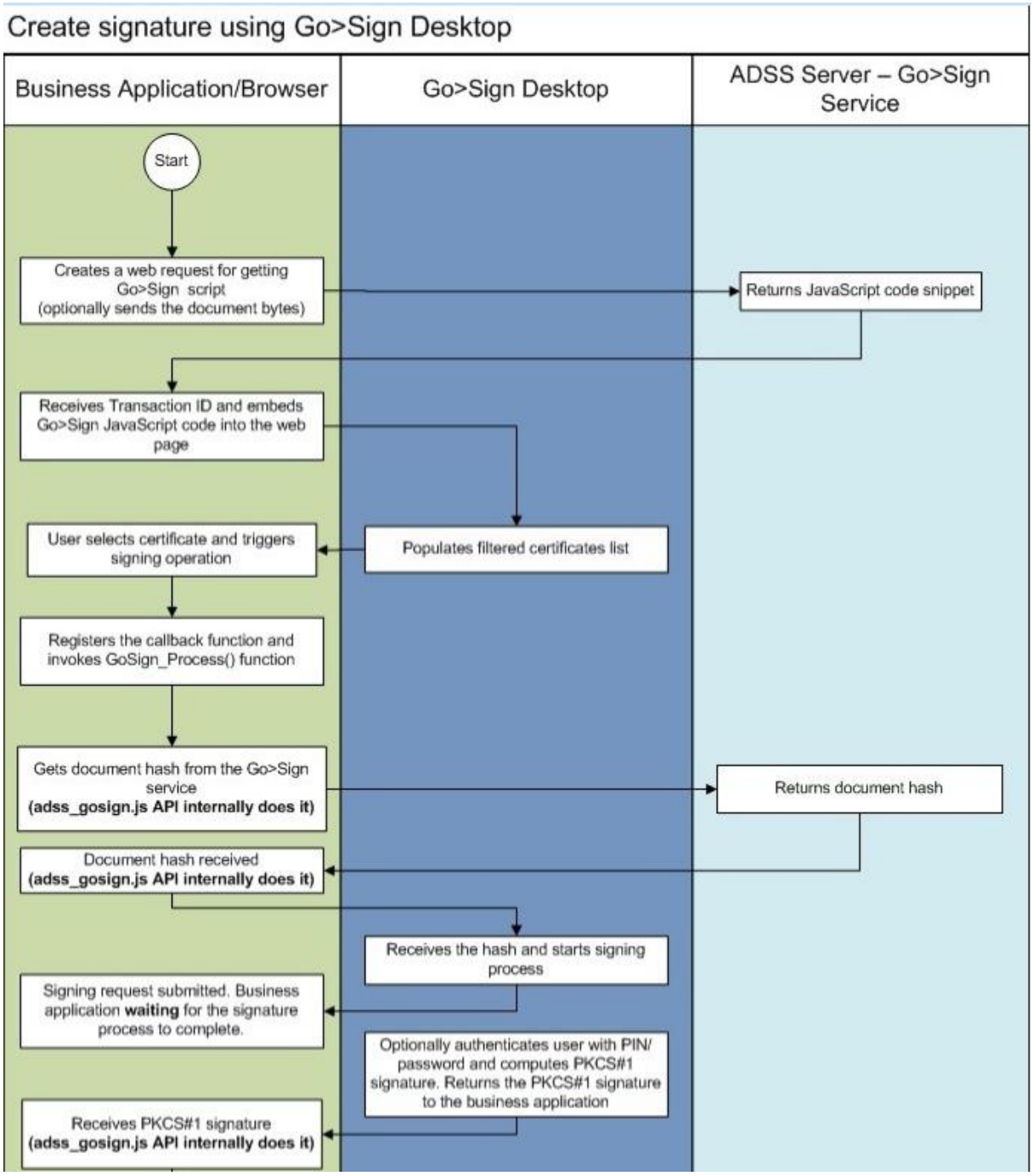
		T2S-participants tasks	
Techn. doc. chapter	**Issue**	**SWIFT**	**SIA-COLT**
1.3	digital signature (row may be removed)	Upload the second certificate to the existing e-token (if not already done)	Check availability of the second certificate on the existing e-token
2.6.3.	technical requirements	Ensure fulfilment of compatibility matrix supported by the NSPs	Ensure fulfilment of compatibility matrix supported by the NSPs
	system set up and updates	Ensure fulfilment of the NSP recommendations	Ensure fulfilment of the NSP recommendations

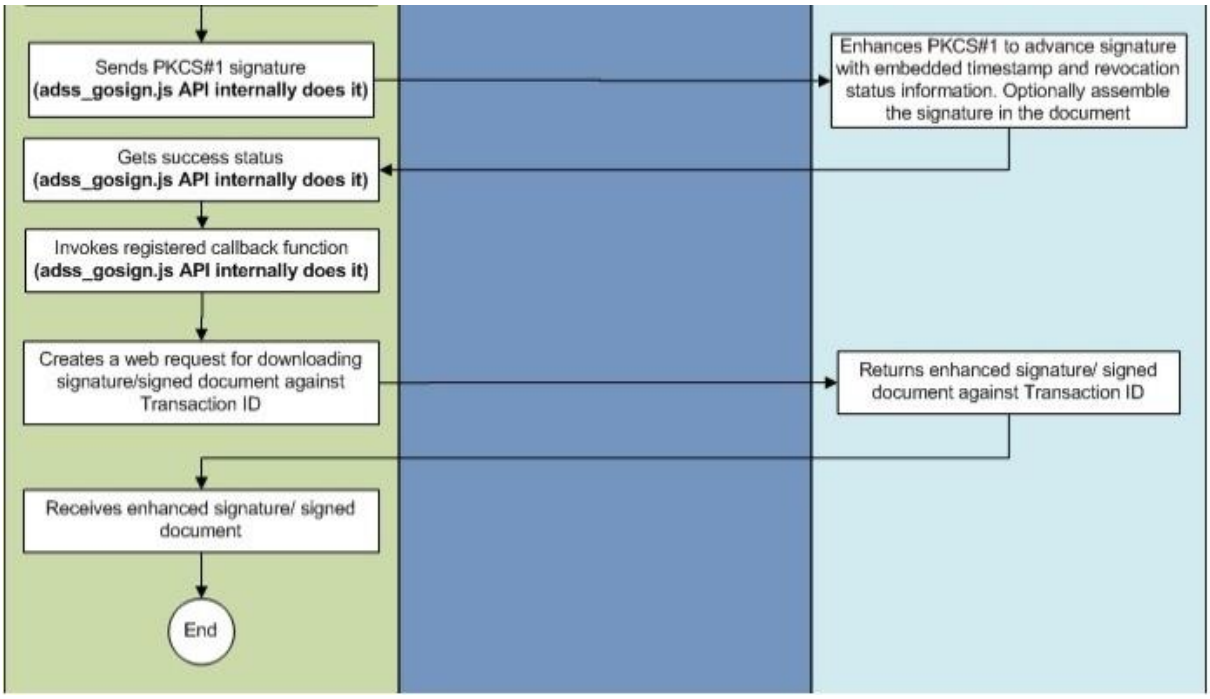
2. Process Overview

2.1. Business Process Overview (without error handling)

The following business workflow shows the interactions of the actors in the U2A signature procedure and their confirmation by the Ascertia Digital Signature Server (ADSS) and serves to establish a better understanding between functional and technical procedures. The same flow applies to both Ascertia Applet and Ascertia GoSign Desktop client; as the IBM CBT Applet signature flow is not compatible with the future GoSign Desktop client one, also the IBM CBT Applet will be replaced with the Ascertia equivalent in order to allow smoother customer migrations.

The T2S GUI will select the applet or the desktop client according to the browser used to access so , for example, one user with IE11 will use Ascertia Applet; other browsers will instead be using automatically the Ascertia Desktop client.


Create signature using Go>Sign Desktop
Business Application/Browser
Go>Sign Desktop
ADSS Server – Go>Sign Service
Start
Creates a web request for getting Go>Sign script (optionally sends the document bytes)
Returns JavaScript code snippet
Receives Transaction ID and embeds Go>Sign JavaScript code into the web page
User selects certificate and triggers signing operation
Populates filtered certificates list
Registers the callback function and invokes GoSign_Process() function
Gets document hash from the Go>Sign service (adss_gosign.js API internally does it)
Returns document hash
Document hash received (adss_gosign.js API internally does it)
Receives the hash and starts signing process
Signing request submitted. Business application waiting for the signature process to complete.
Optionally authenticates user with PIN/ password and computes PKCS#1 signature. Returns the PKCS#1 signature to the business application
Receives PKCS#1 signature (adss_gosign.js API internally does it)



2.2. Process Description (without error handling)

In order to perform a digital signature, the user needs a private key associated with a public certificate stored in a portable device (e-token) accessible by entering a PIN- code. As defined by the NSPs, the PIN code applies to the token, i.e. it is the security password for accessing the token and its content and as such, the same PIN is used for accessing all certificates stored within the token.This PIN will be then also used for signing the selected instructions initiated by the user (other than for authentication).

1) The user enters an instruction into the input fields of the U2A screen and clicks the submit-button. Working in multiple windows is not supported2.
2) The screen adds the signing area (a screen area containing the signing applet) to the current screen, loads and initializes the applet. The applet appears on the screen, allowing for scrolling and viewing the complete data entered on the screen while signing.
3) The screen passes the message to be signed to the applet which verifies that the message to be signed is present and correct.
4) If check is ok, the applet requests a list of available certificates from the driver then displaying the signature one to the user. Steps 2 to 4 happen without user interaction in a short time span.
5) The user presses the “ok/sign” button of the applet which will call the token driver in order to establish the PKCS#11 session.

[2] Cf. UHB Version 2.1, page 47.

6) PIN is requested by the applet/token driver to the user (as first log-in step during PKCS #11 session establishment with the token). This is required for every signing request (in order to authenticate the application/applet to the token).
7) Once correct PIN is entered in the popup window, the applet finally establishes a PKCS #11 session with the key token, which is in turn necessary to elaborate message signature. The user can move the popup window free over his screen, i.e. the position of the popup windows is selectable.

   Only screens used for performing sensitive U2A functions (as defined by the user in CR-0466) require entering the PIN.

   The signing of the instruction is performed either in 2-eyes mode or in 4-eyes mode. The choice depends on the set-up of the T2S-GUI for the participant. If the U2A function is set up in 2-eyes mode, the user who submits the request has to sign his request.

   If the U2A function is set up in 4-eyes mode, the user who submits the request has to sign the initial request and the second user has to confirm with his respective signature.
8) The applet can finally elaborate message signature (through PKCS #11 standard functions/API available on the token), build the signed message and then transfer the signed request back to the U2A screen. As expected, the private key used for cryptographic calculation never leaves the token.
9) The screen sends the signed request to the T2S server, where a Digital Signature Server (DSS) will verify the signature as well as the validity of signature certificate.
10) DSS performs the above mentioned verification activities.
11) If both the signature and the certificate are valid, the request is forwarded to the back-end processing and the request is stored (T2S-environment).
12) Meanwhile the server sends the confirmation of the acceptance of the signature to the user.
13) During the SoD-processing the signed requests are extracted and transferred to the legal archiving module (LEA) for archiving.

2.3. Business Process Overview (with error handling)

As there are many different error conditions, a graphic description is not deemed adequate.

2.4. Process Description (with error handling)

The following error handling is focused on the user side and is described up to the second signature for transferring the sensitive U2A functions. Network security features are not described. For more details on login credentials see chapter 1.3. (NSP responsibility).

T2S does not provide any reporting of logging information as in the current design defined in CR-0466 error information is not transferred from the client to the server. All scenarios that do not result in a signed message are of no consequence to the backend - regardless of the cause of the error prohibiting signing (e.g. no or wrong PIN entered, certificate not shown by the applet,

etc.), no processing happens as no message is signed. Thus, so far no possibility to archive or log errors on the client side exists.[3]

In case the input message is eventually missing or not well formatted, the applet will not start and will raise an error blocking in the application. User may need to open a ticket in this case.

In case user certificate is expired the applet will not show it thereby stopping the signature process. A revoked but not yet expired certificate will be eventually identified on server side during message verification; the DSS module will report accordingly the error to T2S back-end application which will, in turn, track this in the log files. It has to be noted that in compliance with A2A traffic, and according with user requirements, erroneous messages are not transferred to LEA.

In case certificate is valid, the user is demanded for PIN-entry into the popup window:

- With a positive PIN-entry a handover starts the request to the third party applet.
- With a negative PIN-entry the response leads to a request for a second attempt.
- After “wrong PIN threshold” reached, message signature process fails and the U2A screen will inform user that the instruction was not processed. Subject to further detailing by NSPs, the token will then be locked and the customer has to activate an ad-hoc procedure with NSPs to unlock it or eventually request a new one – please refer to NSPs documentation for additional details.

Remarks: the minimum PIN length and its restrictions and the maximum number of failed attempts before lock off are described in chapter 2.4.1. You must be aware that the NSP´s have different specifications.

In case no certificate is available for signing the customer may need to verify correct token installation and restart the signature process.

In case required PIN input field is left empty, a message will be displayed to the user in an applet pop-up dialog window.

Finally, in case signing is cancelled (i.e. user cancels the PIN entry) or fails for technical reasons, an error will be returned by the token driver to the applet. Appropriate mechanism will be put in place to inform the user accordingly.

As described in UHB V2.0, after 10 minutes of inactivity in the live-environment T2S will automatically log you out.

[3] The requirement to report login information was not raised during the scope and content definition of the CR-0466. This clarification is included as a PMG workshop (11/03/2015) follow-up.

2.4.1. Token login credential policies

The main credential policies are summarized in the following table:

	SWIFT	SIA/COLT
Minimum PIN Length	6	4
PIN restrictions	Alphanumeric. At least two different characters must be used	Any number of ASCII characters
Maximum number of consecutive failed attempts before the token is locked	5	3

Signatures contained in U2A instructions are created by the originator and verified by the recipient (T2S) at runtime. In case of unsuccessful verification of a signature at runtime, an error message must be generated. The user session must be terminated and an alert must be sent to the standard technical monitoring system allowing the post treatment.

2.5. Restrictions

2.5.1. Trusted Archiving

The non-repudiation functionality is obtained by associating the digital signature technique with a trusted archiving system, i.e. legal archiving, to be able in case of legal action to provide evidence and to prove that an event actually occurred in the past.

The digital signature will be stored in the inbound communication table of the Interface Domain, which is itself stored at the start of the day in the legal archiving.

The retrieval of archived transactions from legal archiving is under the responsibility of T2S Service Desk by request and described in the Manual of Operational Procedures V1.0., Chapter 3.4.4. The maximum time-frame to get the requested archived data will be three days.

No dedicated customer queries and reports on NRO related static and dynamic data will be possible.

2.6. Technical requirements and recommendations

2.6.1. Go>Sign Desktop Client Requirements

The client invocation on T2S user side will be triggered by T2S application (via Javascript) with the first attempt to sign an instruction (and each time the user needs to sign one) and is transparent to users.
ADSS Go>Sign Desktop relies on TLS communication only with the T2S application (port 8782). This communication is secured using a TLS server certificate having hostname:

client.go-sign-desktop.com.

Therefore, the local client machine must be able to resolve this FQDN (Fully Qualified Domain Name complete domain name for a specific computer, or host, on the internet) to itself.
In order to achieve this, the standard procedure foresees that the Go>Sign Desktop Installer automatically adds the entry :

127.0.0.1 client.go-signdesktop.com

in the Operating System host file to register the client.go-sign-desktop.com as a local domain (Windows OS: C:\Windows\System32\Drivers\etc\hosts).
This will add the FQDN client.go-sign-desktop.com to resolve to IP address 127.0.0.1.
The default value client.go-sign-desktop.com *must not be changed*. The TLS server certificate will be self-signed and different for each workstation where the client will be installed. Once loaded into Windows OS, it is expected to be found in the Root CA keyring (i.e. and not in the personal certificate keyring).

32 bit and 64 bit client version will be made available through dedicated links accessible by CSDs and their communities.
The T2S users have to ensure that the security settings of their institutions, i.e. firewalls, allow for installation of the applet/desktop client as well as for code signing certificate revocation check, if not generally disabled. By default, User Account Control Settings (UAC) are enabled in windows. ADSS Go>Sign Desktop needs user permissions to make changes on the installing device. Windows always prompt a dialog to get the user permissions if user granted the permissions then ADSS Go>Sign Desktop would be installed on the device.

2.6.2. Other technical requirements

Here following a list of items to be checked before starting the test sessions or in case of exceptions that may block the testing. (APP meaning applet; GSD meaning GoSignDesktop client; ALL meaning checks to be done in any case).

- (ALL) In case of certificate exceptions in the browser during first interaction with new Ascertia infrastructure: add DSS host certificates (possibly sent by the T2S/NSP infrastructure) in browsers keyring (IE11, Chrome, Firefox ecc). Host names following for information:

SIA TST	t2s-tst-dss.sia-colt.target-ssp.eu
SIA CRT	t2s-crt-dss.sia-colt.target-ssp.eu
SIA PRD	t2s-dss.sia-colt.target-ssp.eu
SWIFT TST	t2s-tst-dss.ssp.swiftnet.sipn.swift.com
SWIFT USR	t2s-usr-dss.ssp.swiftnet.sipn.swift.com
SWIFT PRD	t2s-dss.ssp.swiftnet.sipn.swift.com

The same above URL may need to be added to the browsers trusted sites.

- (ALL) In case of CORS issue during first interaction with new Ascertia infrastructure, please temporarily disable CORS checks both in Chrome and/or FF

a. FF --> https://addons.mozilla.org/es/firefox/addon/access-control-allow-origin/ + Toggle ON

b. Chrome --> "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" --disable-web-security --user data-dir="C:\.....\Chrome"

- (APP) Check if the Java Plug-in is enabled from IE11 option and enable it in case of need. Close IE11 and access the page again.

- (GSD) Check windows host file for the definition 127.0.0.1 client.go-sign-desktop.com

- (GSD) Check installation / install "client.go-sign-desktop.com" certificate in Chrome and Firefox explicitly (or check first if it is in all browsers keyring after GSD client installation)

https://client.go-sign-desktop.com:8782

- Check GSD client running before start using T2S GUI both on the Windows tray and by using the following URL in a browser:

  https://client.go-sign-desktop.com:8782/gosign-desktop

Above URL may also need to be added in the browser trusted sites in order to work properly. Expected response: “go sign desktop is running”.

2.6.3. Qualified Configurations

The Eurosystem has qualified a specific subset of the NSPs compatibility matrix. These configurations have been extensively tested and support on them is guaranteed.

APPLET USAGE		
NSP	SWIFT	SIA-COLT
OS	Windows 10[4]	
Browser	Microsoft Internet Explorer 11[5] Mozilla Firefox v52 ESR	
JRE[6] Version	1.8.0_66 or higher. (The SWIFT solution for T2S U2A is not compatible with 64-bit version of Java)	1.8.0_66 or higher. (following the availability of the CISCO SSL VPN Gateway software)

DESKTOP CLIENT USAGE[7]		
NSP	SWIFT	SIA-COLT
OS	Windows 10	
Browser	, Google Chrome 26.0+, Firefox 20.0+	

4 Other OSs are supported by the Ascertia solution but are not included in the qualified configuration; 4CB will provide support to the maximum extent possible eventually.

5 The compatibility view mode may need to be disabled.

These cryptographic key stores, used to access the signing keys, are supported:

- MS CAPI/CNG (Windows)
- PKCS#11 for hardware-based tokens

Standard desktop client package can be either installed on traditional desktop machines or on virtual desktop, provided through Citrix Virtual Desktop technology. Installation notes are covered in section 3.1.
Desktop client package installation notes for Terminal Server environment (i.e. Citrix Terminal Server, Windows Remote Desktop protocol) are instead covered in section 3.2.

The 4CB will ask customers running a software version lower than that qualified to upgrade to a qualified version in order to proceed with problem investigation.

The 4CB will investigate issues experienced by customers while running a software version higher than that qualified: If the root cause is linked to the specific software version, then the 4CB will attempt to find a workaround (which may involve customers downgrading their software to a qualified version). The 4CB will evaluate whether a fix for the issue can be included in a future T2S GUI release.
Customers using totally or partially different system components or versions than those mentioned are then responsible to verify the full compatibility with the T2S GUI in the test environments and the system. 4CB will in any case, provide support to the maximum extent possible for checking / testing alternative configurations in order to support troubleshooting process.
The local customer system set-up, i.e. adaption of the local firewall and security policies in order to enable the client installation, HTTPS communication, access to the certificates on the USB tokens from the client machines (either physical or remote workstations). is out of control of T2S and is under the responsibility of T2S participants (that may also need to involve their internal IT Dept. as well as external providers, in case of product specific issues).

2.6.4. Logging and trouble shooting

ADSS Go>Sign Desktop application has two log levels. First informational, which is for normal use, and second, debug, which should only be used when investigating performance issues, functionality problems, etc. For Windows OS and go>sign client configuration, users can view ADSS Go>Sign Desktop application logs at:

C:\Users\[User_Name]\AppData\Roaming\Ascertia\Go-Sign-Desktop\logs

For troubleshooting, if emptying Java cache has not already solve the issue, the customer should follow the below-mentioned indications in order to collect relevant information for analysis:

- Taking step-by-step screenshots providing a brief description of each step,
- Setting the trace level to '5' in the applet JAVA Console,(only for IBM Java applet configuration)
- Activating in the web browser console (open in IE using F12 key, in Firefox using crtl+shift+J) the network information collection (network tab) and providing both network statistics and copy of the "console" tab.
- Send the GoSignDesktopLog.txt

According to some customers' experience:

- Relevant URLs have to be added not only in the trusted sites under IE settings/Security, but also in the local intranet area (in case of IE browser);
- One token at time is connected to a workstation during signing operation

Configuration of the following windows parameters could be activated on specific workstations in order to improve responsiveness of instruction signature[8]:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\AFD\Parameters]
"DefaultSendWindow"=dword:0000fc00
"DefaultReceiveWindow"=dword:0000fc00
```

2.6.5. Change logging level for ADSS Go>sign

By default, ADSS Go>Sign Desktop logging level is set to INFO. To enable detailed debug logging, follow these instructions:

1. Go to ADSS Go>Sign Desktop installation path → C:\Program Files\Ascertia\Go-Sign-Desktop\app\conf\
2. Edit the gosign_desktop.properties file using a suitable text editor.
3. Change the value of the property GOSIGN_DESKTOP_LOG_LEVEL from INFO to DEBUG and save the file.
4. Stop ADSS Go>Sign Desktop application → right click ADSS Go>Sign Desktop application icon and select the option Quit.
5. Start ADSS Go>Sign Desktop application → Start Menu

[8] When a PC is connected to a network, in the local PC settings a limit is set on the capacity to upload files on this network. This buffer to upload could be by default rather limited, making the upload of big files for a user more time-consuming. An IT administrator could change these local settings in order to allow overall greater responsiveness of sending and receiving traffic over the network. We recommend testing this on a test workstation first.

2.7. Risk assessment

In line with the Information Security Risk Treatment Plan (ISRTP) PPSA-06, the successful implementation of CR-0466 (now CR722) provided an adequate mitigation of the risk identified. Since T2S Release 1.1 was in production the risk was decreased to a level that has been considered acceptable according to the Eurosystem ORM Risk Management Policy.

Risk Treatment Plan ID	Description of the: (i) Proposed Action Plan; or (ii) Risk Acceptance Proposal		Residual Risk								Planned Implementation Date	Status
			Before				After [1]					
		Threat Id	Likelihood	Business	Reputation	Financial	Likelihood	Business	Reputation	Financial		
PPSA-06	Implementation of U2A signature (CR-0466 for release 1.1)	T52	1	1	3	5	1	1	3	2	Q1 2016	Started in Q2 2014

3. Annex

3.1. GoSign Desktop (GSD) Client – Desktop Installation Guide

3.1.1. GoSign Desktop Client download

The below URLs allow the download of the installation manual plus the desktop client, version 6.6.0.14 for both Windows 32 and Windows 64 bit OS.

SIA links:

https://t2s-eac-gui.sia-colt.target-ssp.eu/ICMWeb/Ascertia/ADSS-Go-Sign-Desktop-Installation-Guide.pdf
(official Ascertia installation guide)
https://t2s-eac-gui.sia-colt.target-ssp.eu/ICMWeb/Ascertia/ADSS-Go-Sign-Desktop-Win32.msi
(32bit Win package)
https://t2s-eac-gui.sia-colt.target-ssp.eu/ICMWeb/Ascertia/ADSS-Go-Sign-Desktop-Win64.msi
(64bit Win package)

https://t2s-utest-gui.sia-colt.target-ssp.eu/ICMWeb/Ascertia/ADSS-Go-Sign-Desktop-Installation-Guide.pdf
(official Ascertia installation guide)
https://t2s-utest-gui.sia-colt.target-ssp.eu/ICMWeb/Ascertia/ADSS-Go-Sign-Desktop-Win32.msi
(32bit Win package)
https://t2s-utest-gui.sia-colt.target-ssp.eu/ICMWeb/Ascertia/ADSS-Go-Sign-Desktop-Win64.msi
(64bit Win package)

SWIFT links:

https://t2s-eac-gui.ssp.swiftnet.sipn.swift.com/ICMWeb/Ascertia/ADSS-Go-Sign-Desktop-Installation-Guide.pdf
(official Ascertia installation guide)
https://t2s-eac-gui.ssp.swiftnet.sipn.swift.com/ICMWeb/Ascertia/ADSS-Go-Sign-Desktop-Win32.msi
(32bit Win package)
https://t2s-eac-gui.ssp.swiftnet.sipn.swift.com/ICMWeb/Ascertia/ADSS-Go-Sign-Desktop-Win64.msi
(64bit Win package)

https://t2s-utest-gui.ssp.swiftnet.sipn.swift.com/ICMWeb/Ascertia/ADSS-Go-Sign-Desktop-Installation-Guide.pdf
(official Ascertia installation guide)
https://t2s-utest-gui.ssp.swiftnet.sipn.swift.com/ICMWeb/Ascertia/ADSS-Go-Sign-Desktop-Win32.msi
(32bit Win package)
https://t2s-utest-gui.ssp.swiftnet.sipn.swift.com/ICMWeb/Ascertia/ADSS-Go-Sign-Desktop-Win64.msi
(64bit Win package)

3.1.2. GoSign Desktop Client installation manual

Installation as well as uninstall procedures are described in the attached document from Ascertia. The NRO technical document remains the reference for qualified configurations plus checks and troubleshooting indications.

ADSS-Go-Sign-Desktop-Installation-Gui

3.1.3. GoSign Desktop Client – TLS certificate renewal instructions

In case / Once the TLS GoSign desktop client certificate will need to be renewed , please execute the following steps in order to proceed with renewal process:

1) Stop Go>Sign Desktop

2) Go to Go>Sign Desktop installation directory i.e “C:\Program Files\Ascertia\Go-Sign-Desktop\app\conf“ and remove the file 'gosign.keystore'

3) Restart Go>Sign Desktop and it will add the new self-signed certificate and remove the previous one from windows keystore.

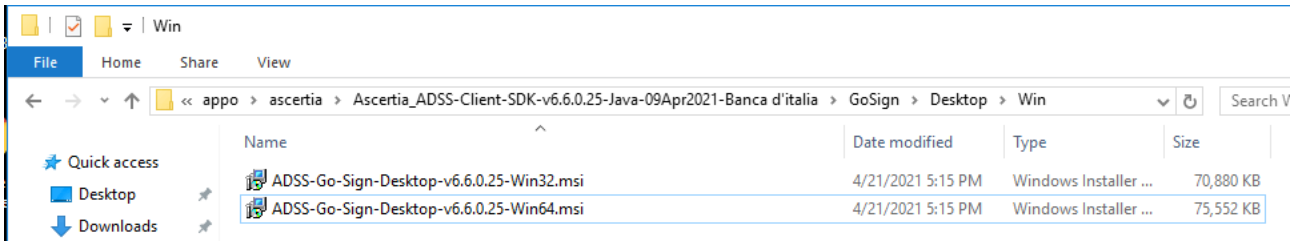
Procedure may be verified in advance and possibly adapted, in case specific tools for installation / sw distribution have been adopted on customer side.

3.2. GoSign Desktop (GSD) Client – Terminal server Installation Guide

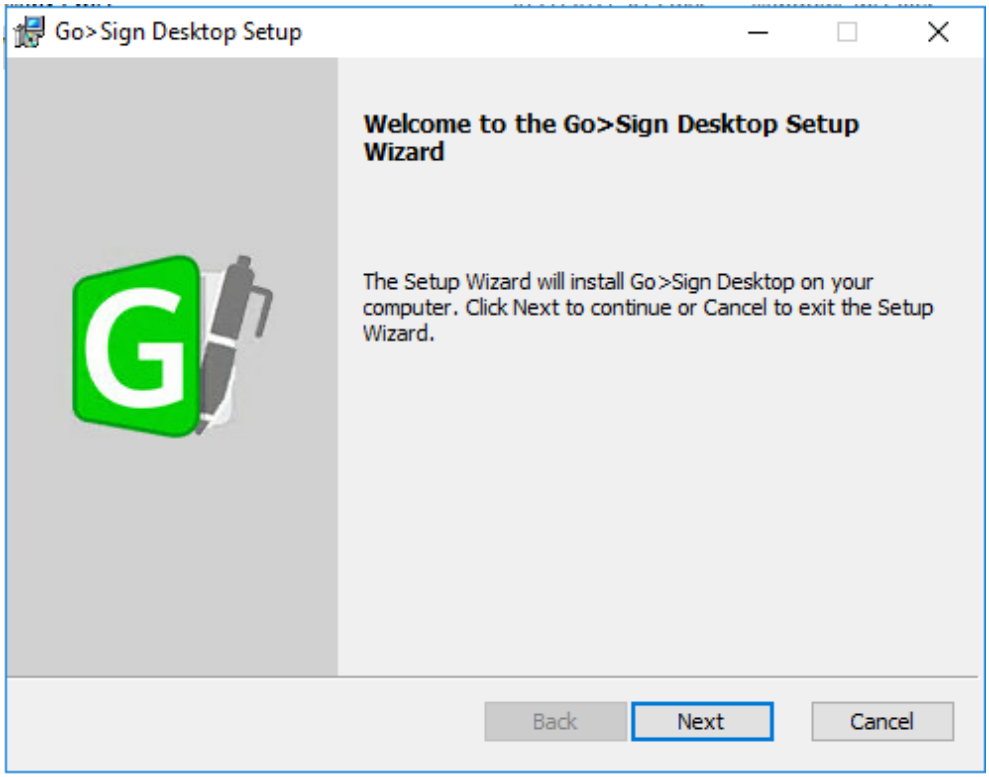
Installation steps are reported in the following paragraph; they may subject to further changes / improvements in order to simplify the overall process.

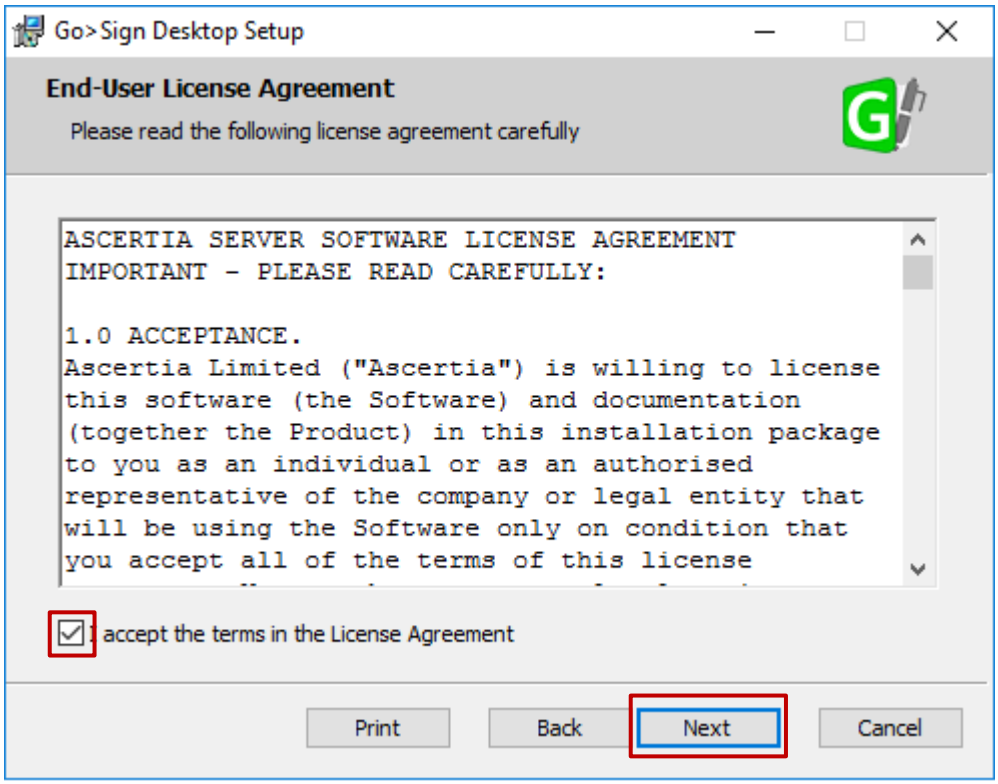
3.2.1. Setup GSD single user client

- Open Command prompt as Administrator
- Execute command: chgusr /install
- then run ADSS-Go-Sign-Desktop-v.6.6.0.xx-win64.msi installation package (6.6.0.14 package can be used)

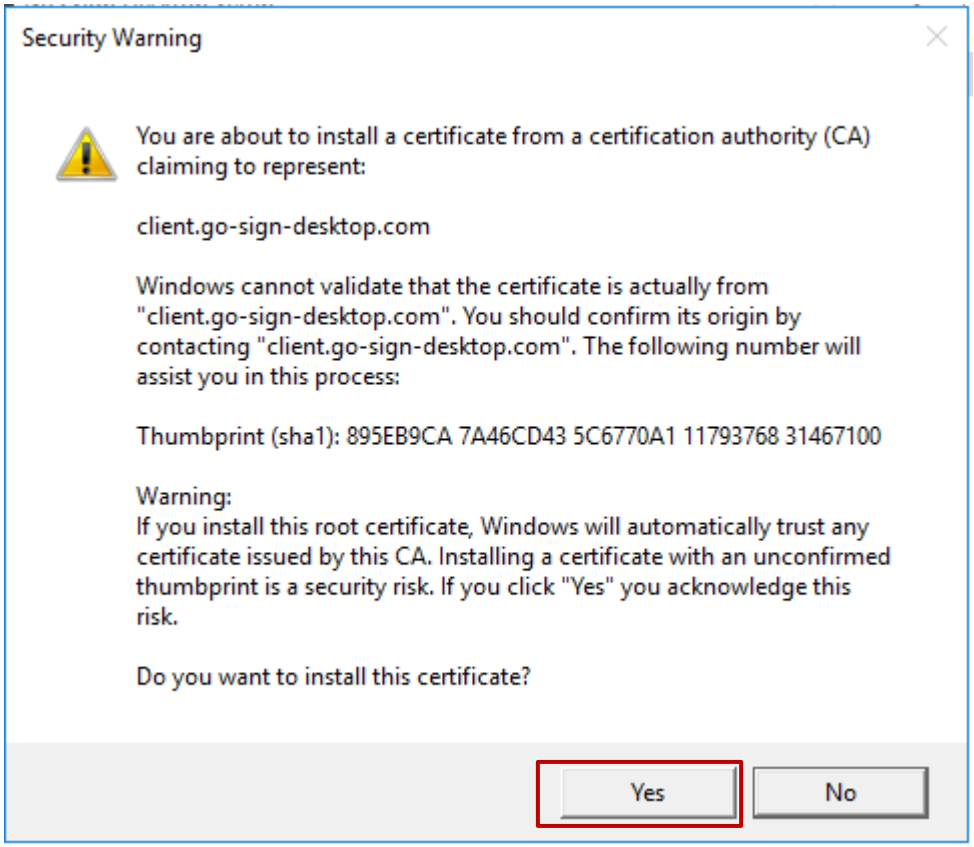


Click Next and accept End User License Agreement

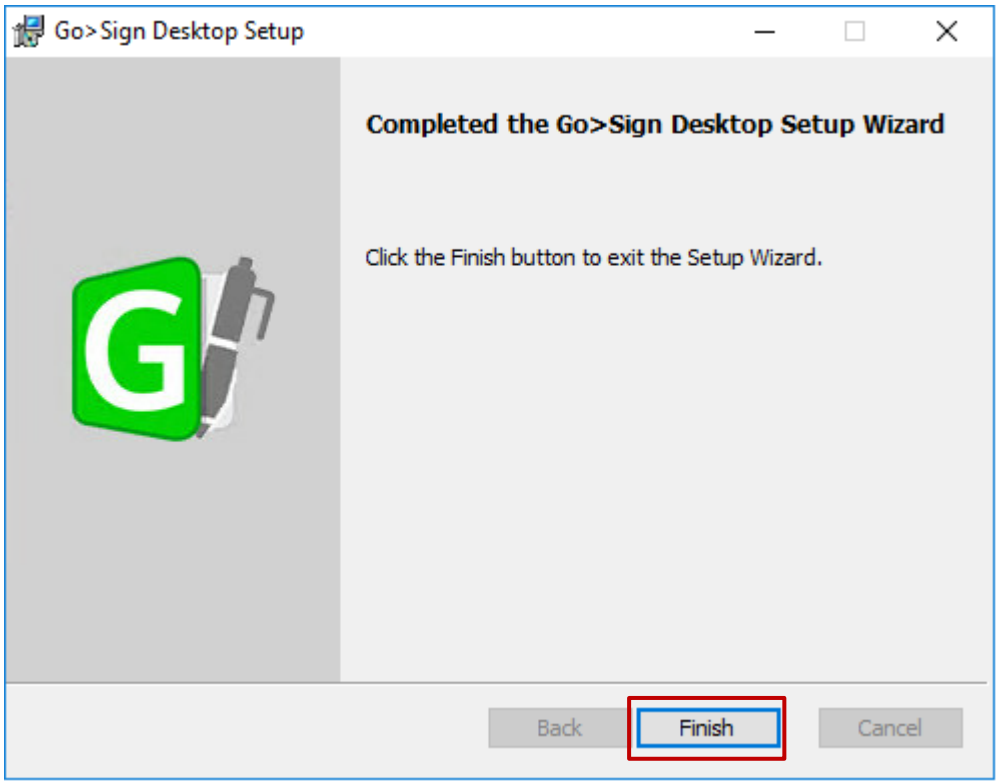




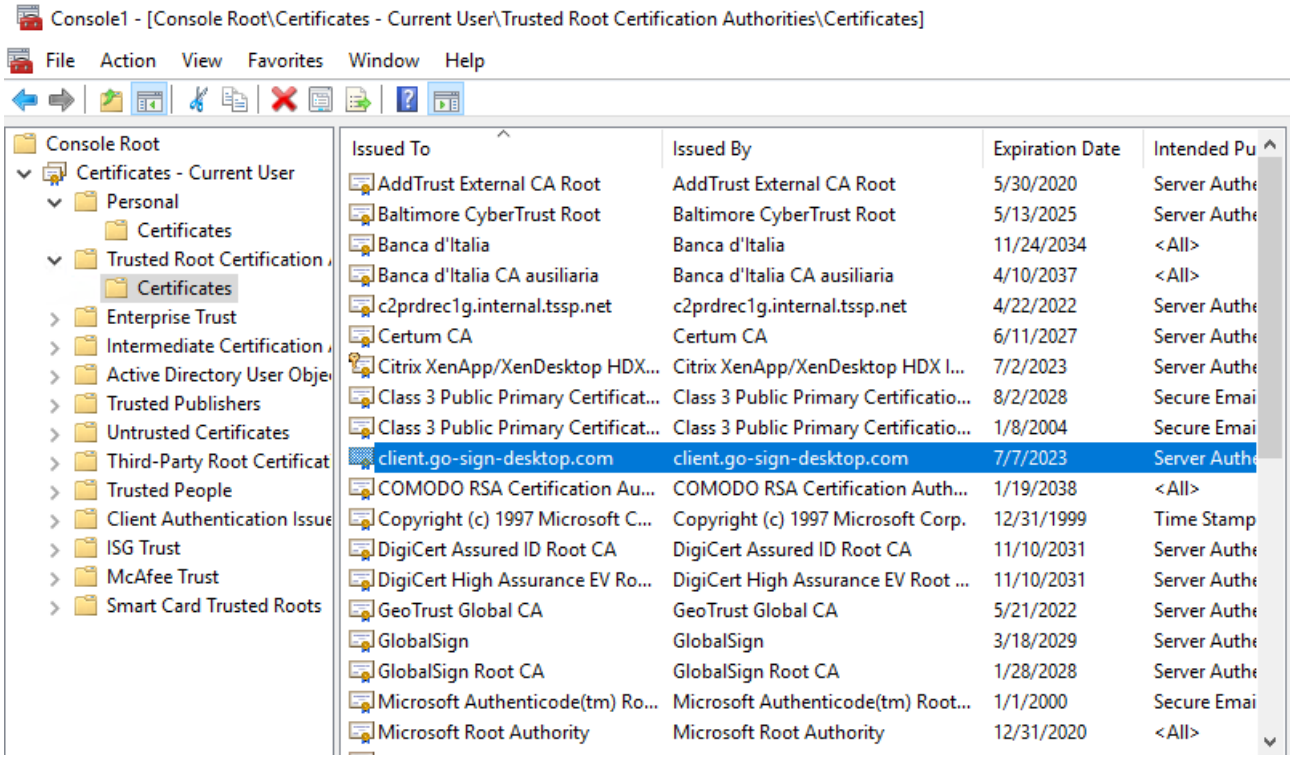
Accept to install certificate:



Select Finish:



Check that certificate client.go-sign-desktop.com is imported in User Certificate store by running *certmgr.msc* tool:



3.2.2. Download and copy GSD executable into GSD client installation path

Download the updated code for terminal server environments from one of the following URLs:

EAC

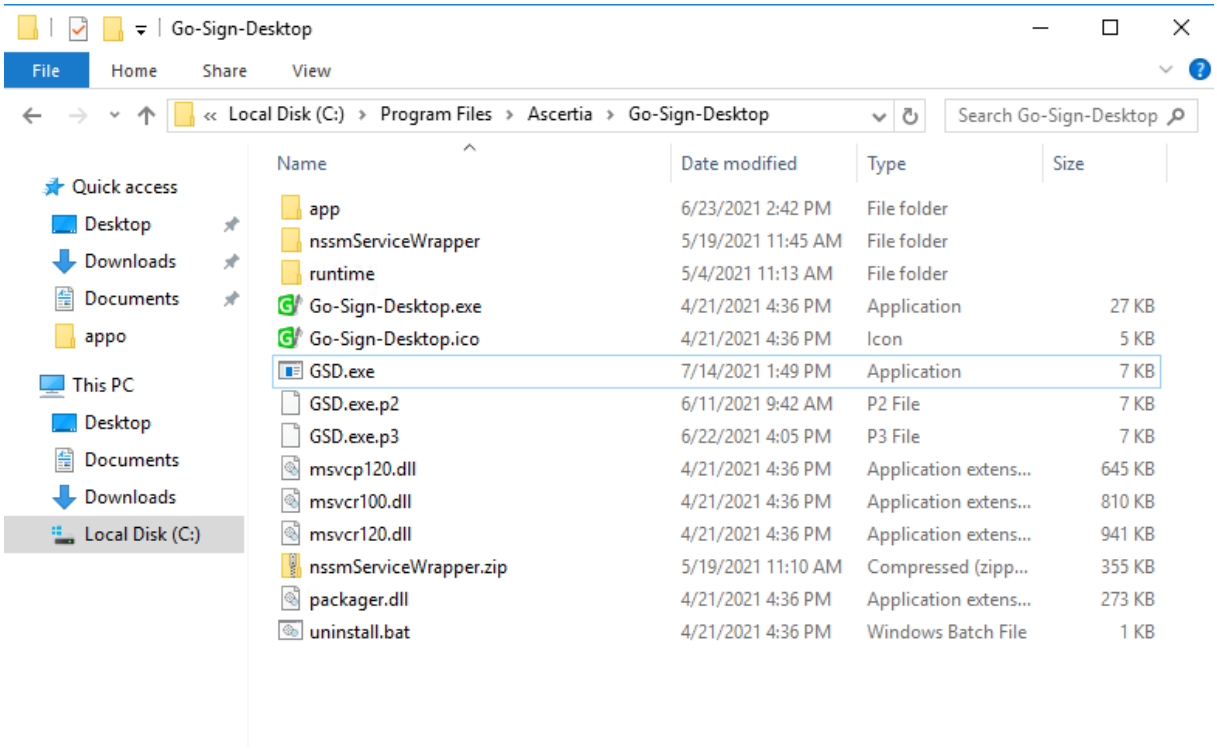
- https://t2s-eac-gui.sia-colt.target-ssp.eu/ICMWeb/Ascertia/ADSS-Go-Sign-Multi-User
- https://t2s-eac-gui.ssp.swiftnet.sipn.swift.com/ICMWeb/Ascertia/ADSS-Go-Sign-Multi-User

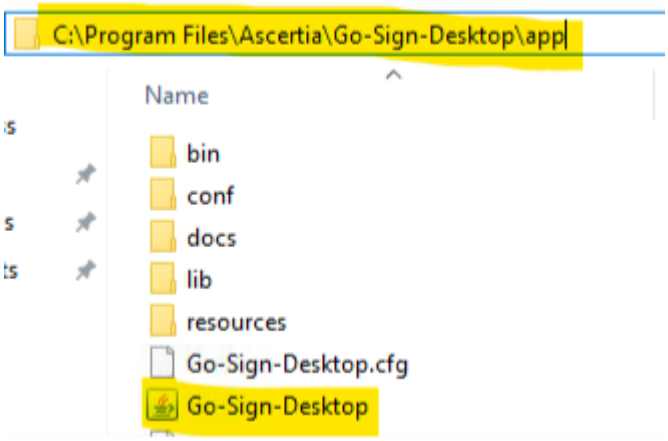
UTEST

- https://t2s-utest-gui.sia-colt.target-ssp.eu/ICMWeb/Ascertia/ADSS-Go-Sign-Multi-User
- https://t2s-utest-gui.ssp.swiftnet.sipn.swift.com/ICMWeb/Ascertia/ADSS-Go-Sign-Multi-User

And

1) copy the GSD exe file into GSD installation directory
2) replace the existing Go-Sign-Desktop.jar file with the one provided in the above package
3) see below screenshots as reference





3.2.3. Update go-sign-desktop.properties file

Following two lines have to be added to the go-sign-desktop.properties file (C:\Program Files\Ascertia\Go-Sign-Desktop\app\conf directory):

GOSIGN_DESKTOP_INSTALLATION_MODE = MULTI_USER
GOSIGN_DESKTOP_LOG_MODE = info

3.2.4. Configure Go-sign-desktop as Windows Service

Current implementation foresees to have Go>Sign Desktop running as a Windows Service via a Service Wrapper which wraps the Go>Sign Desktop executable binary into a Windows Service.(via NSSM opens source software https://nssm.cc/).

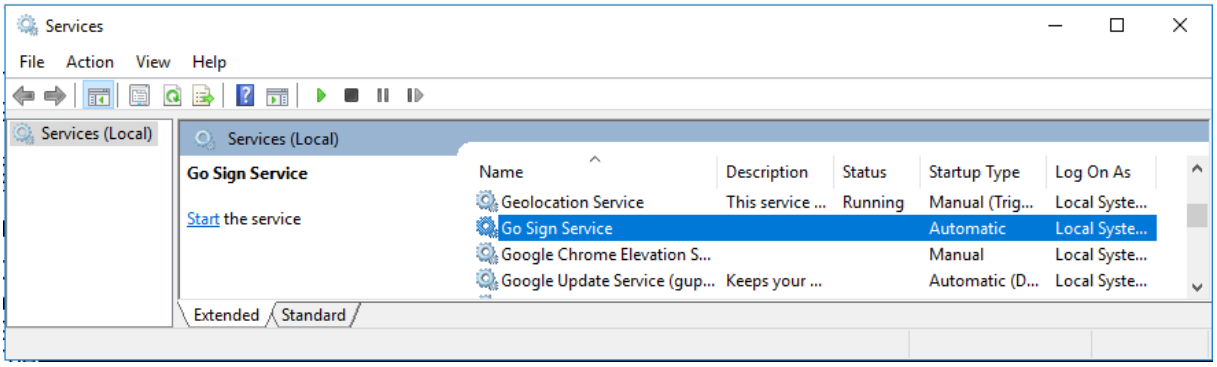
To configure service:

1) Extract archive content “nssmServiceWrapper.zip” to the C:\Program Files\Ascertia\Go-Sign-Desktop\ nssmServiceWrapper

2) Run Command Prompt with Administrator permissions and enter the following line with respect to the real location of Go>Sign Desktop:

```
“C:\Program Files\Ascertia\Go-Sign-Desktop\nssmServiceWrapper\win64\nssm.exe" install "Go Sign Service" "C:\Program Files\Ascertia\Go-Sign-Desktop\Go-Sign-Desktop.exe"
```

3) Open Services.msc and locate the Go Sign Service just installed:



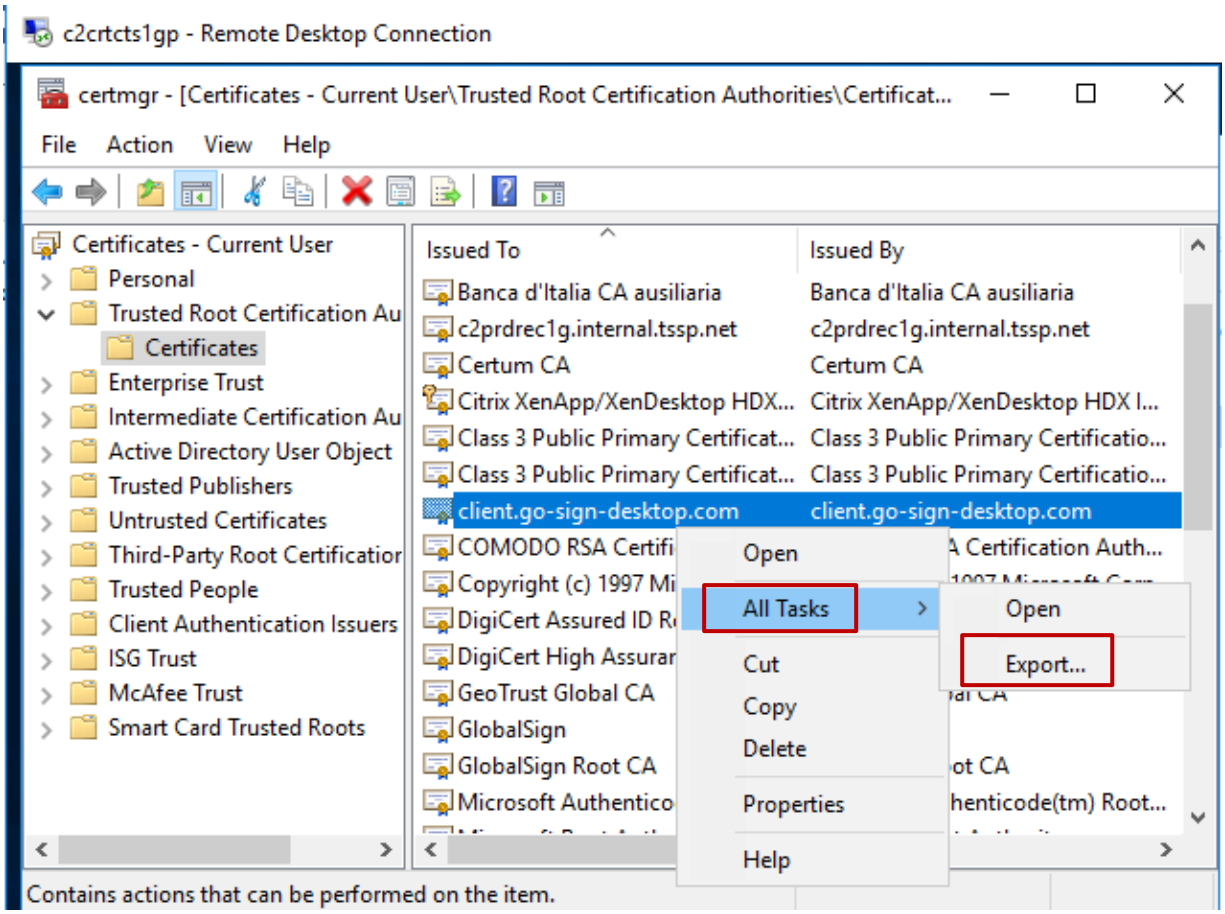
- Right-click Go Sign Service and choose Properties.
- Ensure “Startup type” is set to Automatic.
- Open the Log On tab and change “Log on as” to this account, then specify the account NETWORK SERVICE and leave the password blank

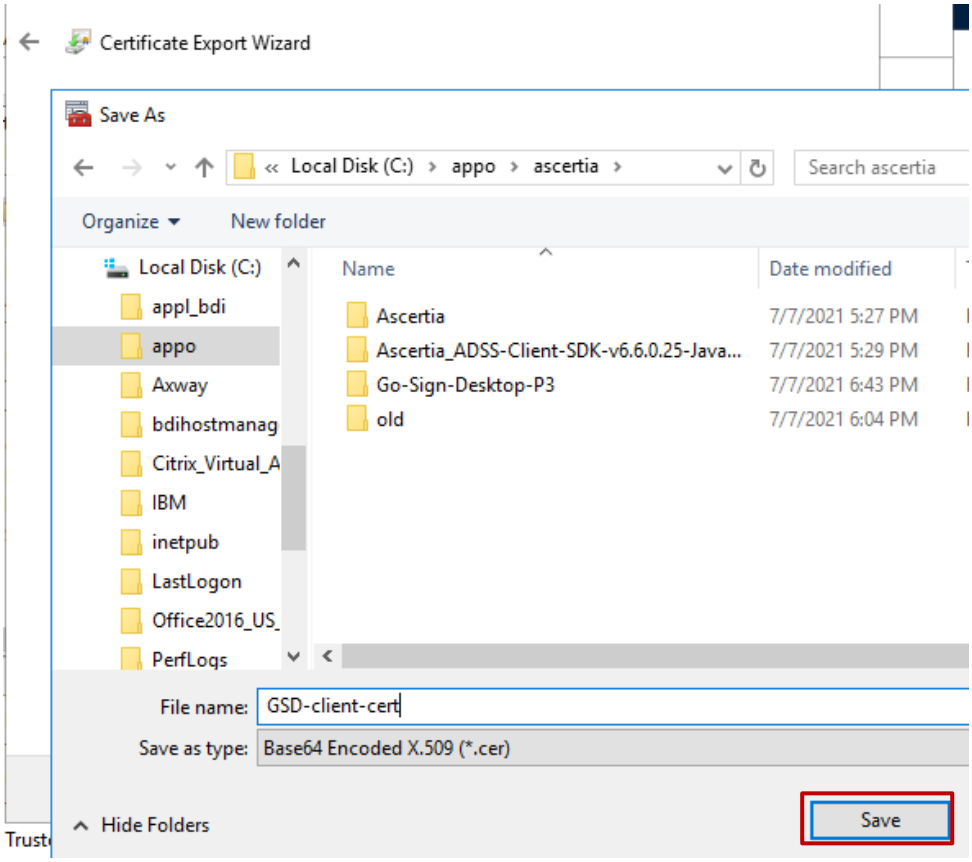
3.2.4.1. Import client.gosign certificate into Network Service user keystore

Run *certmgr.msc* tool and import directly the client certificate exported from the Current User Store.
Alternatively, GSD.exe file should be started from a “Network Service” user environment command prompt in order to start the client.gosign certificate (e.g. psexec tool can be used for such scope).

3.2.4.2. Connect client.gosign into GSD java keystore

Export new client.gosign-desktop certificate from the user store in DER format (certmgr.msc tool):


c2crtcts1gp - Remote Desktop Connection
certmgr - [Certificates - Current User\Trusted Root Certification Authorities\Certificat...
File Action View Help
Certificates - Current User
Personal
Trusted Root Certification Au
Certificates
Enterprise Trust
Intermediate Certification Au
Active Directory User Object
Trusted Publishers
Untrusted Certificates
Third-Party Root Certificatior
Trusted People
Client Authentication Issuers
ISG Trust
McAfee Trust
Smart Card Trusted Roots
Issued To
Issued By
Banca d'Italia CA ausiliaria
c2prdrec1g.internal.tssp.net
Certum CA
Citrix XenApp/XenDesktop HDX...
Class 3 Public Primary Certificat...
Class 3 Public Primary Certificat...
client.go-sign-desktop.com
COMODO RSA Certifi
Copyright (c) 1997 Mi
DigiCert Assured ID R
DigiCert High Assurar
GeoTrust Global CA
GlobalSign
GlobalSign Root CA
Microsoft Authentico
Open
All Tasks
Cut
Copy
Delete
Properties
Help
Open
Export...
Contains actions that can be performed on the item.


Certificate Export Wizard
Save As
Local Disk (C:) › appo › ascertia ›
Search ascertia
Organize
New folder
Local Disk (C:)
appl_bdi
appo
Axway
bdihostmanag
Citrix_Virtual_A
IBM
inetpub
LastLogon
Office2016_US_
PerfLogs
Name
Date modified
Ascertia
Ascertia_ADSS-Client-SDK-v6.6.0.25-Java...
Go-Sign-Desktop-P3
old
7/7/2021 5:27 PM
7/7/2021 5:29 PM
7/7/2021 6:43 PM
7/7/2021 6:04 PM
File name: GSD-client-cert
Save as type: Base64 Encoded X.509 (*.cer)
Hide Folders
Save

Update ‘cacerts’ file located in ‘C:\Program Files\Ascertia\Go-Sign-Desktop\runtime\jre\lib\security’ with the certificate Go>Sign Desktop app trusted when it ran for the first time. The following keytool command to be used:

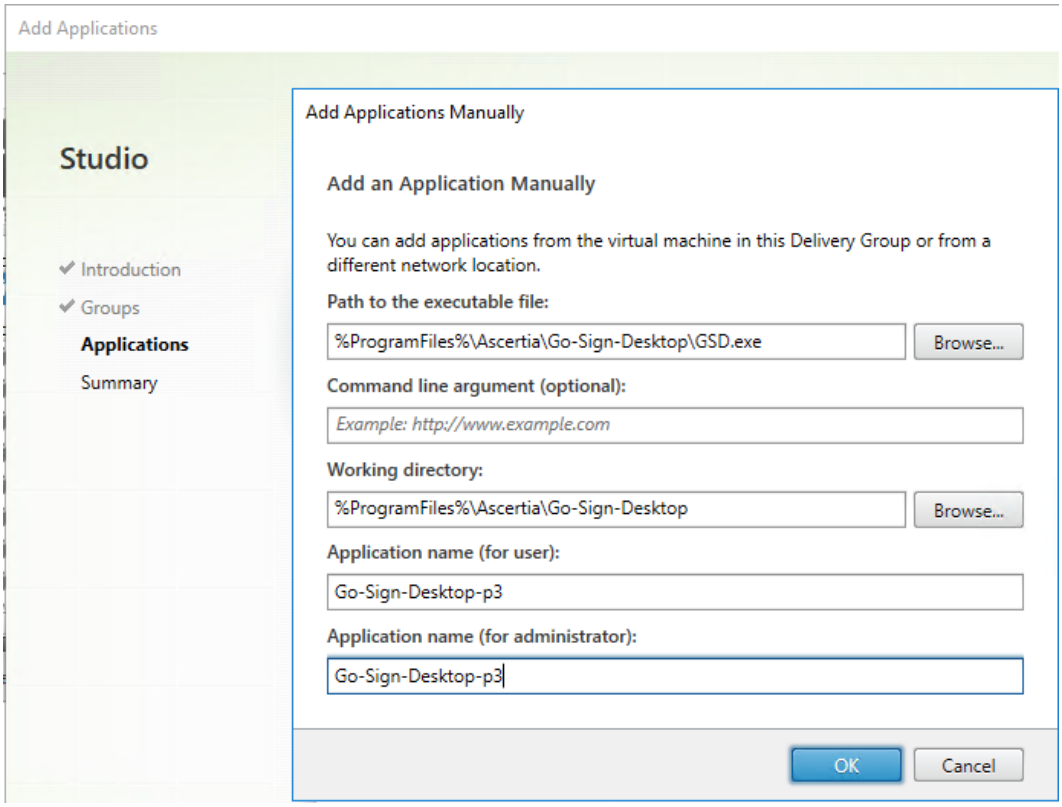
```
keytool -import -alias gosign -file exported-cert-der.cer -keystore "C:\Program Files\Ascertia\Go-Sign-Desktop\runtime\jre\lib\security\cacert"
```

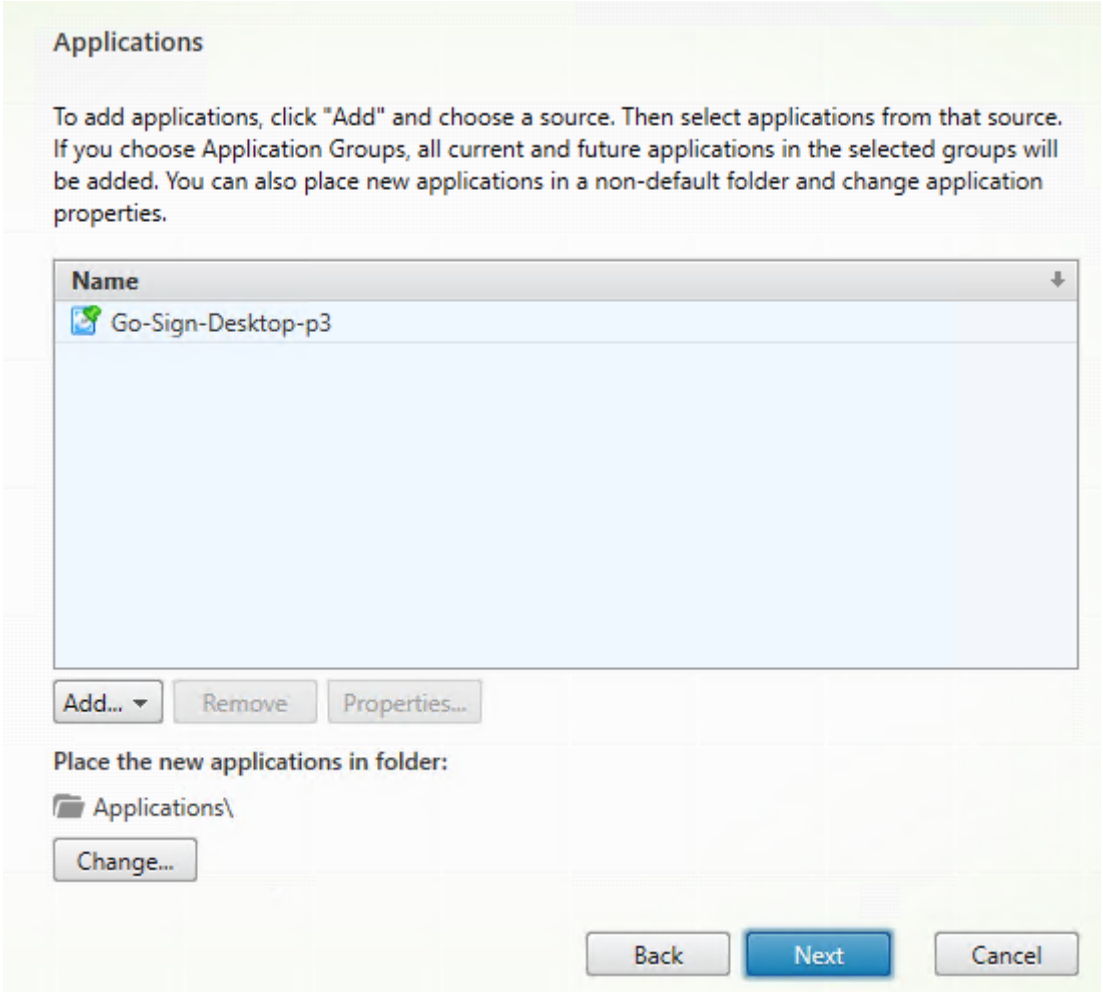
Copy C:\Program Files\Ascertia\Go-Sign-Desktop\runtime\jre\lib\security\cacert from the first server to the other terminal servers (if present).

3.2.5. Publish applications GSD.exe and Chrome in Citrix Farm

3.2.5.1. Publishing GSD from Citrix Studio

From Citrix Studio console Add a new Application, assign Delivery Group and “add an application manually” as described:





To be repeated for all the relevant users.

3.2.5.2. User preliminary actions

From Citrix Studio console Add a new Application, assign Delivery Group and “add an application manually” as described before using GSD each user must launch one time GSD application on Citrix Portal in order to add in user profile the right registry keys and save the user configuration.
Then open browser session.