GOPAS

ADFS and Web Application Proxy

Ing. Ondřej Ševeček | GOPAS a.s. |
MCSM:Directory2012 | MCM:Directory2008 | MVP:Enterprise Security | CEH | CHFI | CISA |
ondrej@sevecek.com | www.sevecek.com |

ADFS intranet scenarios

GOPAS

Web application authentication

- Windows authentication
  - Kerberos, NTLM
  - SSO under domain account
  - RSO under any other account or from the internet
  - web server domain member
- Forms based authentication
  - custom login/credentials
  - cookies (URL bound, lifetime)
- ADFS authentication
  - redirect to ADFS server and back
  - cookies for ADFS and web

GOPAS

ADFS motivation

- Single authenticating server
  - trusted account store
  - trusted connection
  - credentials never “typed” into insecure web services
- Web services easy handling of tokens
  - no worry about security
  - just a signed piece of XML/JSON

GOPAS

Standard web-based authentication

- Active Directory Federation Services (ADFS)
- HTTP server providing several web based authentication mechanisms
  - Active Directory (ADDS)
  - Active Directory Lightweight Directory Services (ADLDS)
  - any third party
- Produces claims or cookies in various formats
  - WS-Trust or SAML-Token for active clients
  - WS-Federation (also called SAML 1.0/1.1) and SAML 2.0 for passive clients
  - OAuth for semi-passive clients
- Required by Office365/AzureAD for on-premises hybrid deployments

GOPAS

Active vs. passive clients

- Passive clients
  - do not understand XML/SOAP/??? by them
  - Internet Explorer, Chrome, FireFox, …
  - java script, HTTP 302 redirects
- Active clients
  - web service knowledgeable clients
  - Active Sync, Outlook, Word, …

GOPAS

ADFS version history

Version	OS	Notes	Support
ADFS 1.0	Windows 2003 R2	included runs in IIS	SAML 1.1
ADFS 1.1	Windows 2008 Windows 2008 R2	included runs in IIS	SAM 1.1 tokens
ADFS 2.0	Windows 2008 Windows 2008 R2	download runs in IIS	SAML 2.0 tokens
ADFS 2.1	Windows 2012	included runs in IIS	device registration
ADFS 3.0	Windows 2012 R2	included direct hosting on HTTP.SYS TLS SNI support PowerShell only config (plus HTML/Javascript) OAuth implicit grant	multifactor auth password change /adfs/probe
ADFS 4.0	Windows 2016	admin delegation own certification authority for device registration http to https redirection with WAP http publishing with WAP OAuth full OAuth and HTTP basic authentication with WAP	Azure MFA Microsoft Passport

ADFS certificates

- TLS HTTPS certificate
  - TCP 443, 49433
  - signs ECDH or encrypts RSA key exchanges
  - should be trusted by all clients
- Service communication certificate
  - by default the same as TLS certificate
  - encrypts SOAP message
  - must be trusted by all clients
- Token signing certificate
  - signs SAML/OAuth tokens
  - must be trusted by all servers as per thumbprint
- Token decryption certificate
  - decrypts SAML/OAuth tokens received from claim providers
  - must be trusted by all servers as per thumbprint

GOPAS

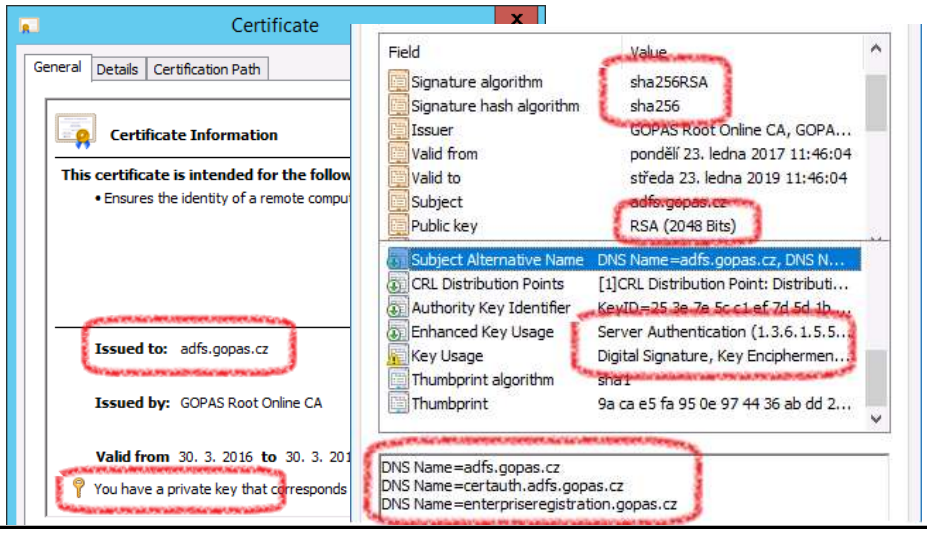
Note: TLS certificate subject names

- Subject
  - just a single name for backward compatibility
  - CN=adfs.gopas.cz
- Subject Alternative Name (SAN)
  - *.gopas.cz
    - wildcard rules them all :-)
    - does not match subname.name.gopas.cz
  - adfs.gopas.cz
    - at least the ADFS public name
  - enterpriseregistration.gopas.cz
    - if device registration is required
  - enterpriseregistration.sevecek.eu
    - if device registration is required for other user UPN suffixes
  - certauth.adfs.gopas.cz
    - with Windows 2016 no need to use TCP 49443 for certificate authentication

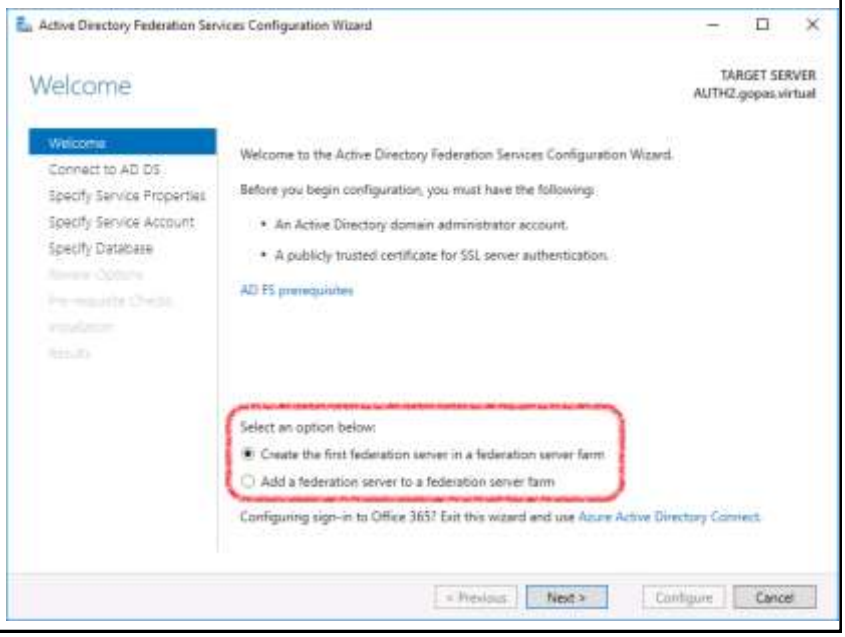
GOPAS

ADFS installation #1

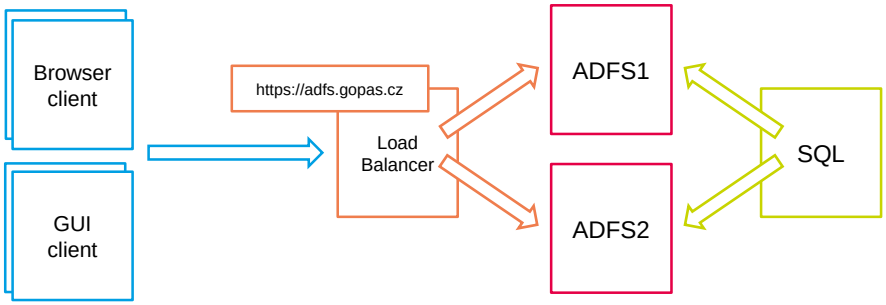
- Buy a public name from public CA
- SHA256, RSA 2048, EKU = Server Authentication
- TLS certificate key usage = Key Encipherment (TLS 1.0) and/or Digital Signature (requires TLS 1.1+)
- Service communication certificate key usage = Key Encipherment



ADFS installation #2

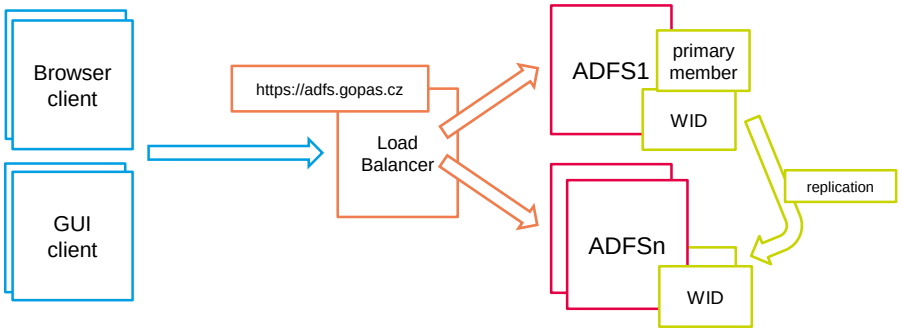


ADFS farm behind a load balancer with a common SQL server



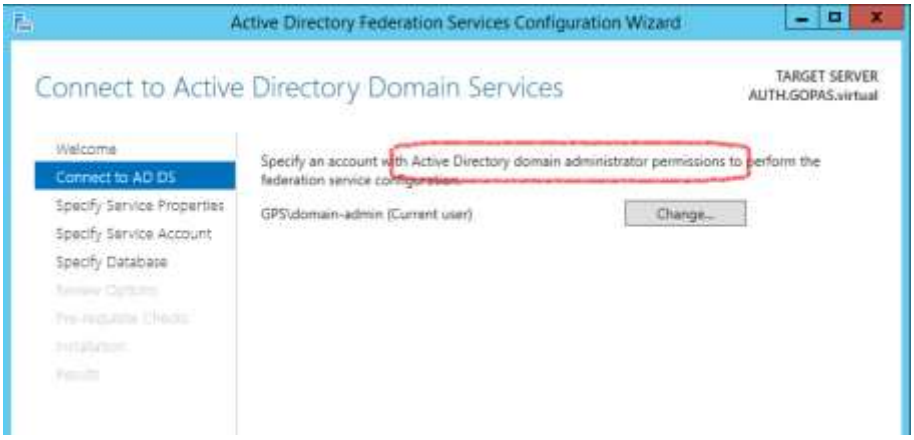
GOPAS

ADFS farm behind a load balancer with individual secondary WID instances



GOPAS

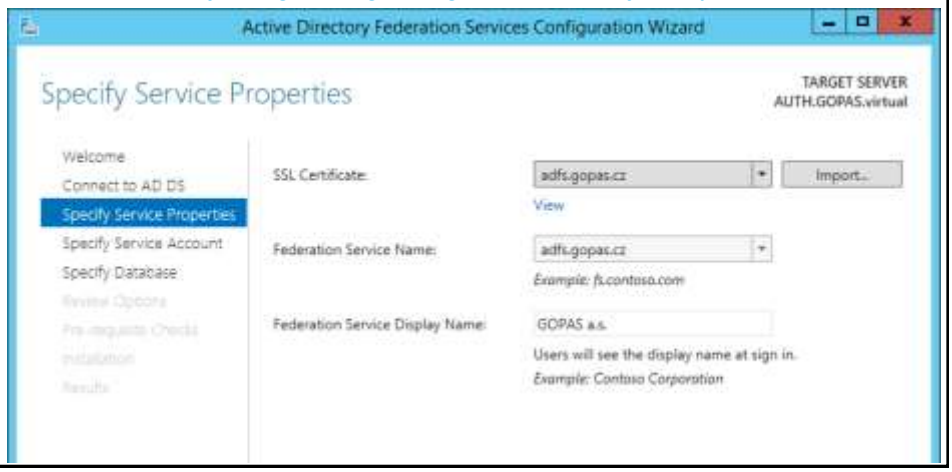
ADFS installation #3



GOPAS

ADFS installation #4

- Certificate template must NOT be Key Storage Provider
  - certutil -repairstore my *
- the best Key Usage is Digital Signature and Key Encipherment



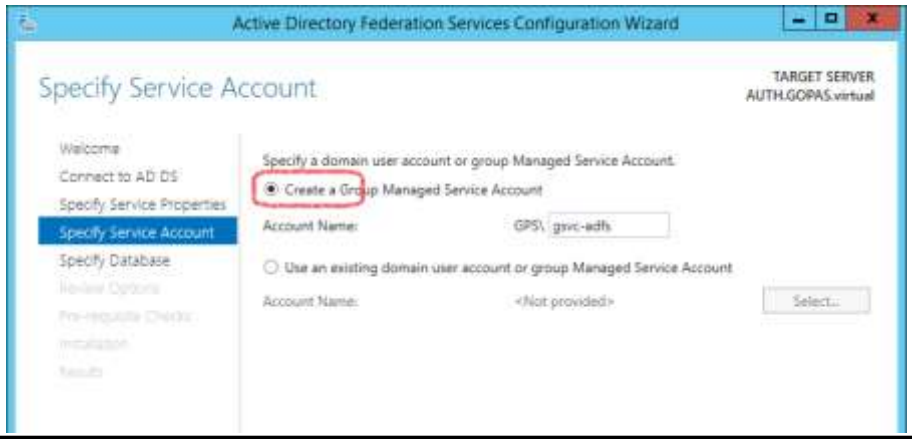
ADFS service communication certificate notes

- Key Storage Provider (CNG, KSP)
  - works fine for ADFS by default
  - does not work when enabling some endpoints
  - e.g. WS-Trust 2005 : password : message
- Digital signature key usage is sufficient by default
  - but if enabling some endpoints they need Key Encipherment
  - e.g. WS-Trust 2005 : password : message

GOPAS

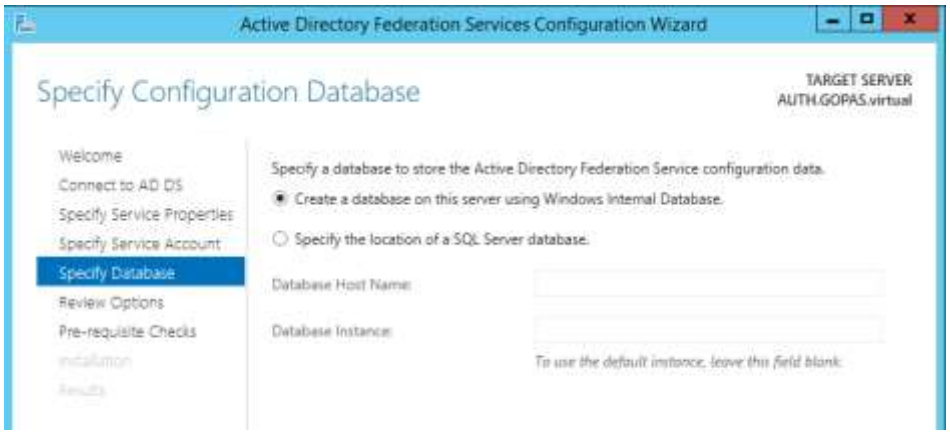
ADFS installation #5

- AD DFL must be Windows 2012+
- AD Key Distribution Service (KDS) must be provisioned
  - Add-KdsRootKey -EffectiveTime ([DateTime]::Now.AddDays(-1))



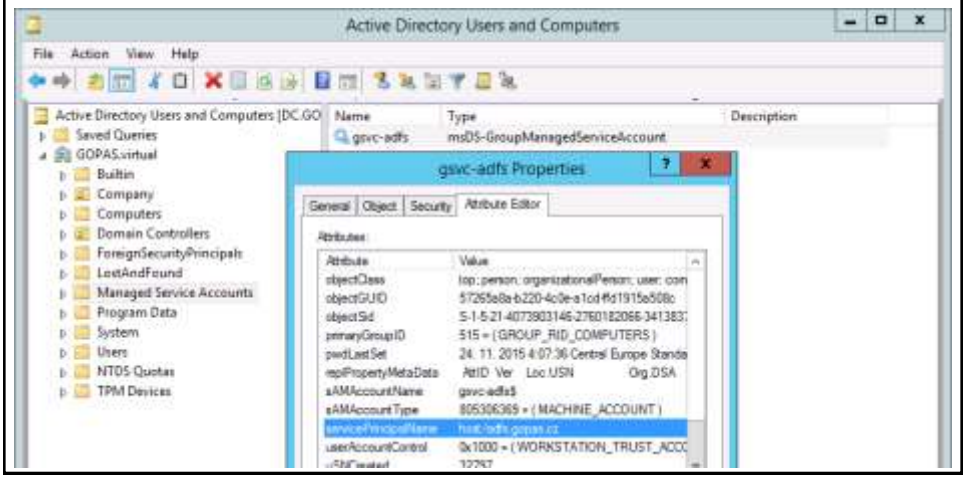
ADFS installation #6

- WID supports up to 5 ADFS servers and 60 000 users with more than 100 relying parties
- WID supports up to 30 ADFS servers with less than 100 relying parties
- Requires sysadmin in full SQL
  - dbcreator and securityAdmin are not sufficient

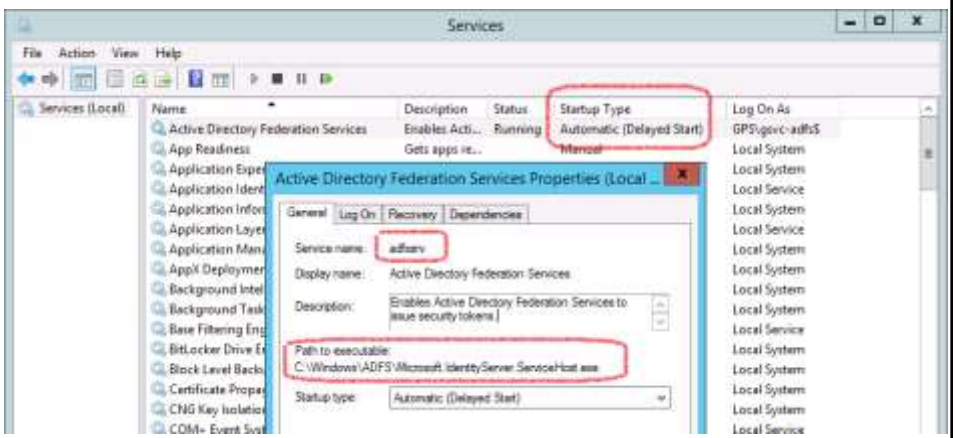


ADFS installation #7

- servicePrincipalName = host/adfs.gopas.cz
  - SOAP clients ask for host/SPN instead of http/SPN
- msDS-SupportedEncryptionTypes = RC4, AES



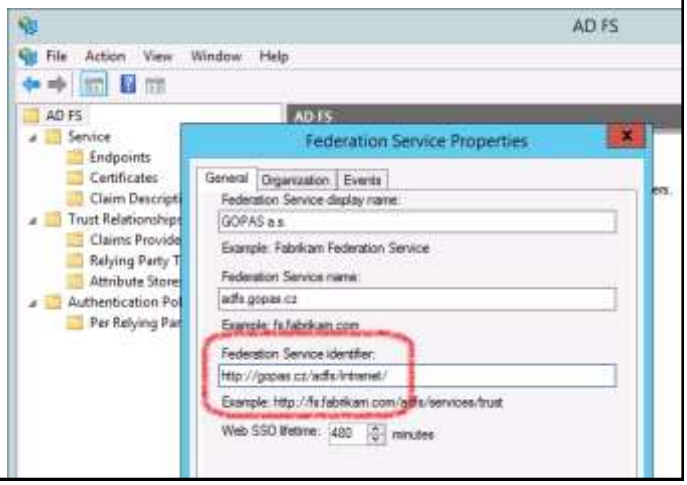
ADFS installation #8



GOPAS

ADFS installation #9

- URI: http://sevecek.com/2016-01/adfs/intranet
- URN: urn:oid:1.3.6.1.4.1.25005.7.3
- URN: urn:fdc:sevecek.com:201601:adfs-intranet

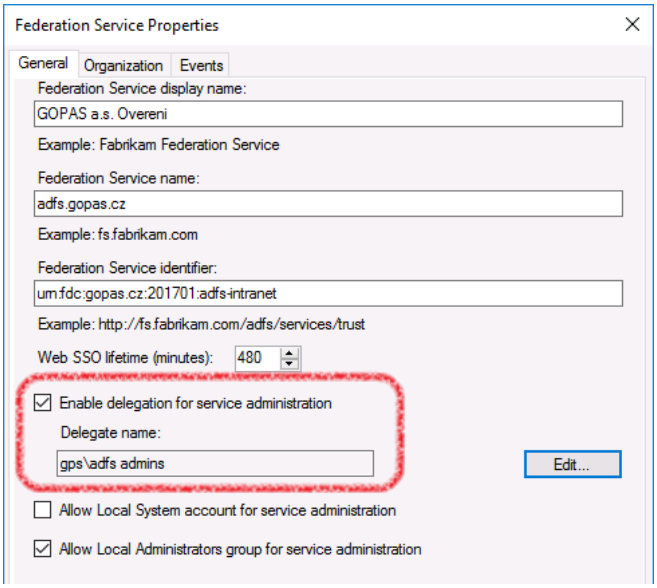


Note: Claim type URNs

- URI: http://sevecek.com/2016-01/adfs/intranet/myClaimType
- URN: urn:oid:1.3.6.1.4.1.25005.7.3/myClaimType
- URN: urn:fdc:sevecek.com:201601:adfs-intranet/myClaimType

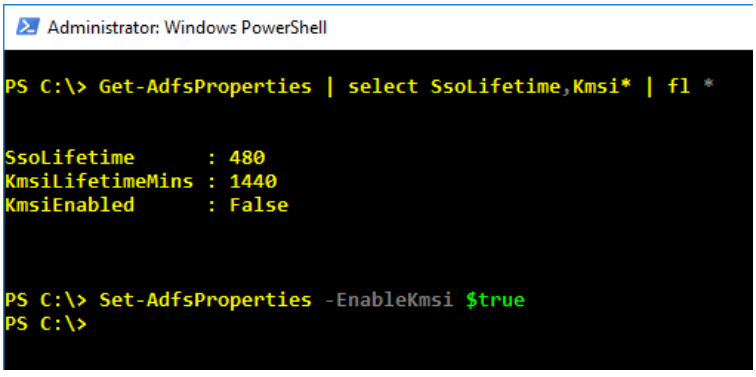
GOPAS

ADFS installation #10 (2016+)

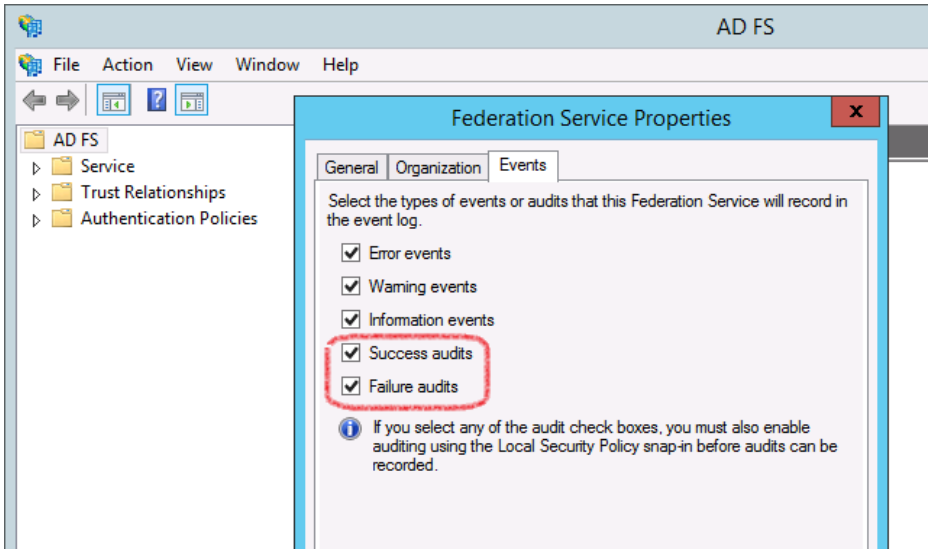


ADFS installation #11

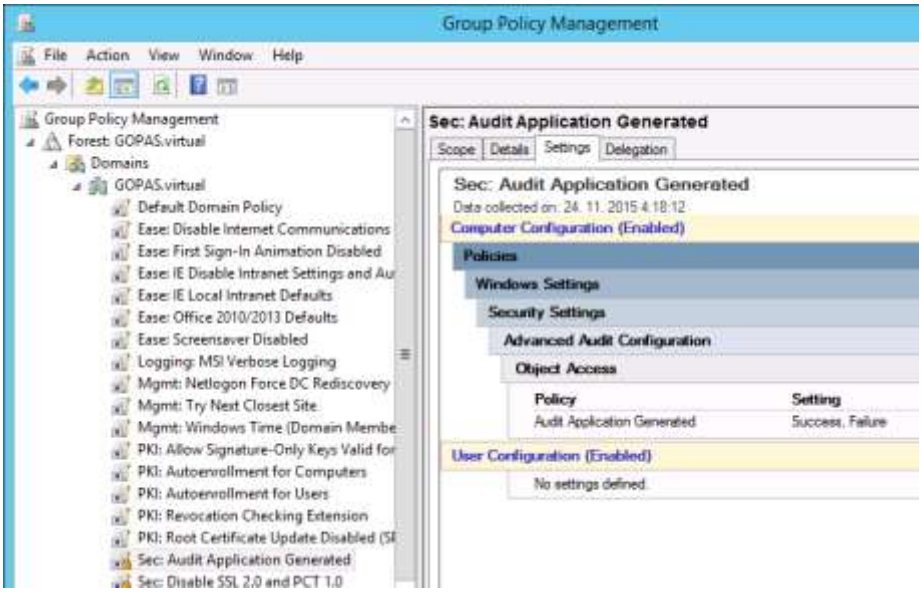
- SsoLifetime
  - lifetime of the MSISAuth session cookie by default
- KsmiLifetimeMins
  - lifetime of the MSISAuth persistent cookie if when KMSI enabled



ADFS installation #12

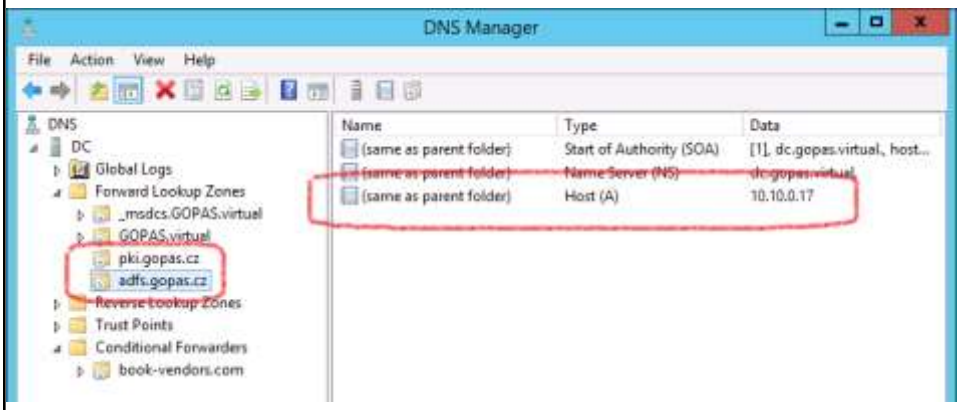


ADFS installation #13



ADFS installation #14

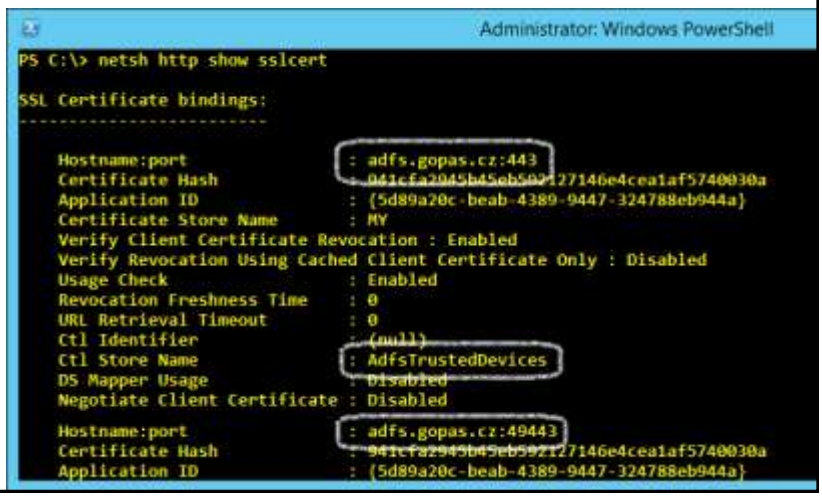
- do not forget about certauth.adfs.gopas.cz



GOPAS

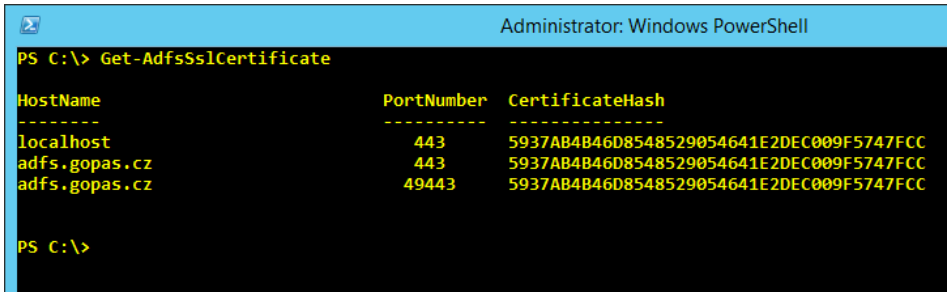
ADFS installation #15

- Port TCP 49443 - client certificate authentication
- AdfsTrustedDevices - ADFS proxy (WAP) TLS client trust



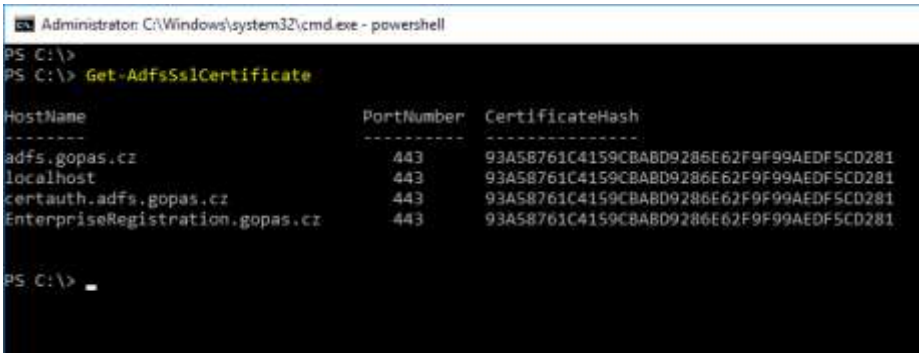
ADFS installation #16

- Get-AdfsSslCertificate, Set-AdfsSslCertificate
  - netsh http show sslcert
  - appId = {5d89a20c-beab-4389-9447-324788eb944a}
- by default it is the same as the Service communication certificate, but might be changed separately
  - ensure the service-communications certificate is the same



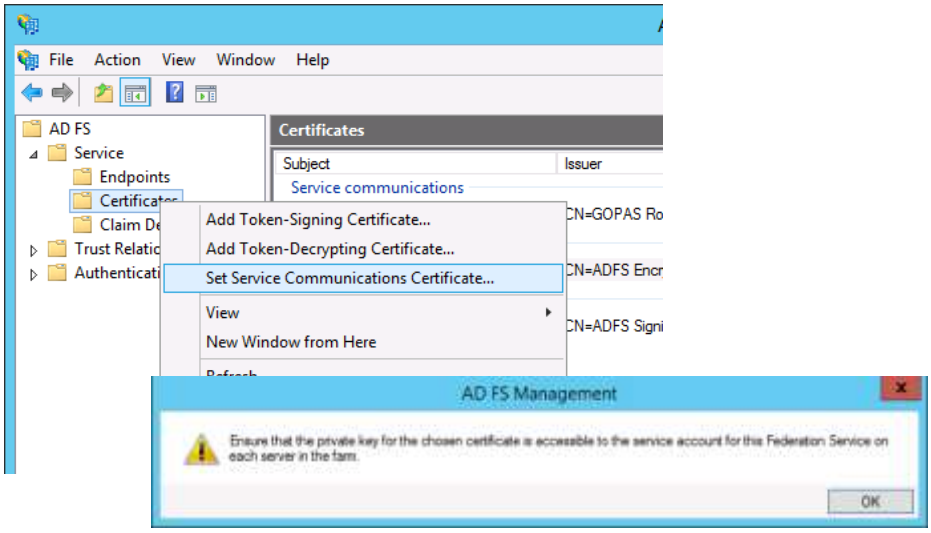
ADFS installation #17

- TLS client certificate authentication since Windows 2016 can use port 443
  - requires certauth.adfs.gopas.cz subject name (rather SAN)
- Set-AdfsAlternateTlsClientBinding -Thumbprint
  - use if certificate changed later (updates HTTP.SYS UrlAcl as well)
  - after the change you can update it with Set-WebApplicationProxySslCertificate on WAP



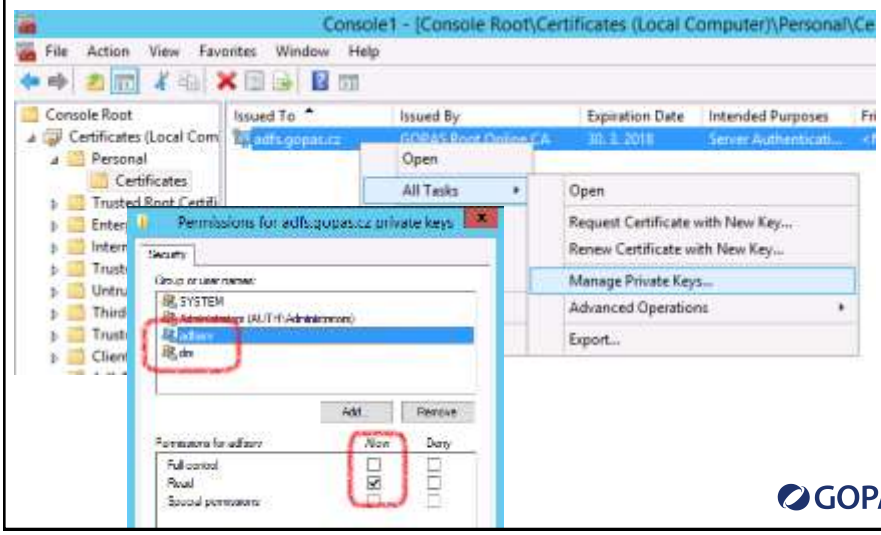
ADFS installation #18

- Renewing/replacing server authentication certificate



ADFS installation #19

- Assign read private key for NT SERVICE\ADFSSRV and NT SERVICE\DRS
- DRS removed on Windows 2016

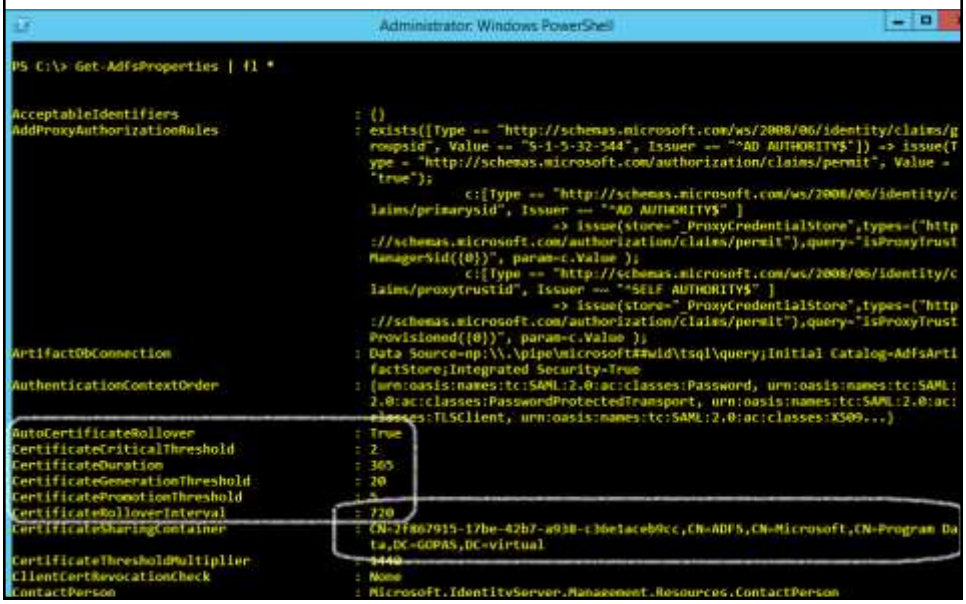


GOPAS

ADFS installation #20

- Service communications certificate
  - used to signing WS-Fed/SAML2 tokens and possibly encrypting SOAP messages
- Token signing certificate
  - self-signed certificate for signing issued tokens
  - on web application part trusted per thumbprint
- Token decrypting certificate
  - self-signed decrypts tokens issued by other AD FS servers
  - imported into the other ADFS and used explicitly
- Get-AdfsCertificate, Set-AdfsCertificate

ADFS installation #21



ADFS installation #22

```
Administrator: Wind
PS C:\>
PS C:\> Set-AdfsProperties -CertificateDuration 730
PS C:\>
PS C:\> Update-AdfsCertificate -Urgent
PS C:\>
```

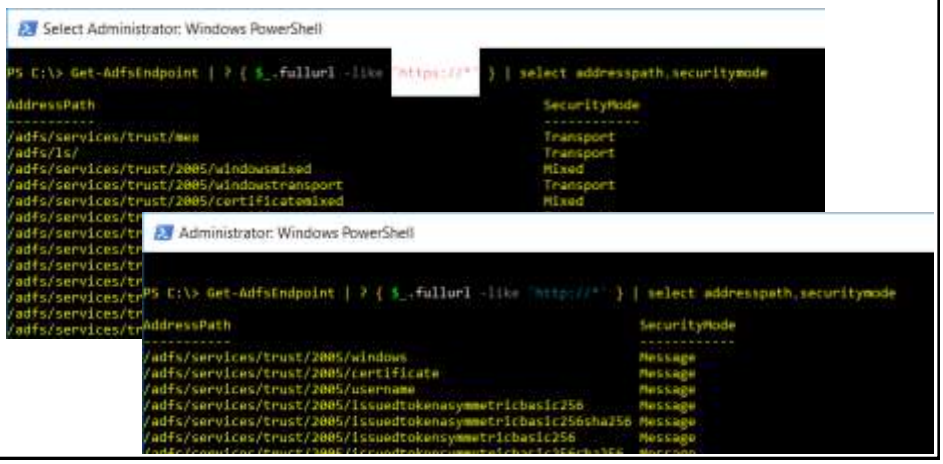
GOPAS

ADFS installation #23

```
DisplayName                                 : GOPAS a.s.
ExtendedProtectionTokenCheck                : Allow
FederationPassiveAddress                    : /adfs/ls/
HostName                                    : adfs.gopas.cz
HttpPort                                    : 80
HttpsPort                                   : 443
Identifier                                  : http://adfs.gopas.cz/adfs/services/trust
LogLevel                                    : {Errors, FailureAudits, Information, Verbose...}
MonitoringInterval                          : 1440
NetTcpPort                                  : 1501
NtlmOnlySupportedClientAtProxy              : False
OrganizationInfo                            :
PreventTokenReplays                         : False
ProxyTrustTokenLifetime                     : 21600
ReplayCacheExpirationInterval               : 60
SignedSamlRequestsRequired                  : False
SamlMessageDeliveryWindow                   : 5
SignSamlAuthnRequests                       : False
SsoLifetime                                 : 480
PersistentSsoLifetimeMins                   : 10080
PersistentSsoEnabled                        : True
PersistentSsoCutoffTime                     : 1. 1. 0001 1:00:00
KmsiEnabled                                 : False
LoopDetectionEnabled                        : True
LoopDetectionTimeIntervalInSeconds          : 20
LoopDetectionMaximumTokensIssuedInInterval  : 5
SendClientRequestIdAsQueryStringParameter   : False
WIASupportedUserAgents                      : {MSAuthHost/1.0/In-Domain, MSIE 6.0, MSIE 7.0, MSIE 8.0...}
ExtranetLockoutThreshold                    : 2147483647
ExtranetLockoutEnabled                      : False
ExtranetObservationWindow                   : 00:30:00
```

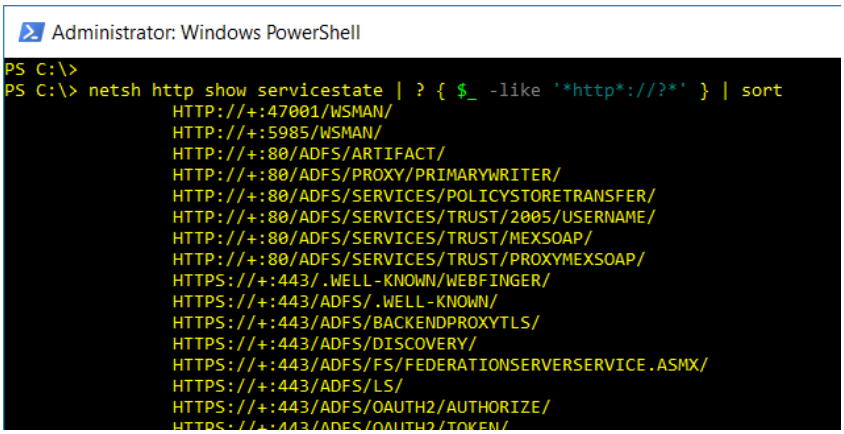
ADFS installation #24

- Transport and Mixed endpoints use HTTPS
- SOAP Message security does not require HTTPS



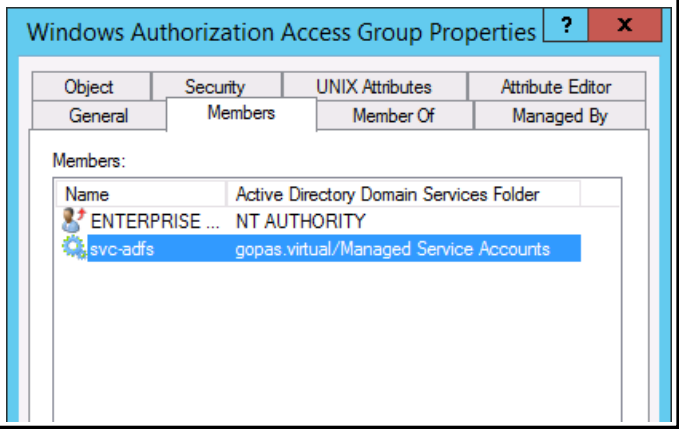
ADFS installation #25

- Enabling/disabling endpoints register them in HTTP.SYS



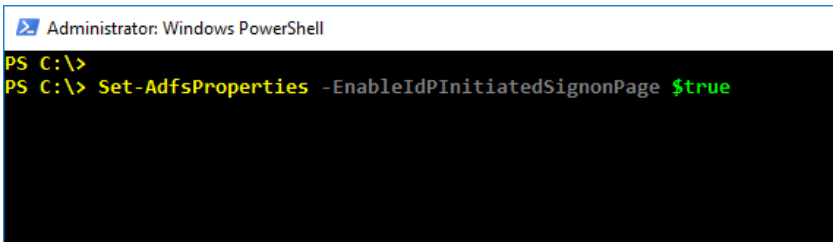
ADFS installation #26

- ADFS service account must be member of Windows Authorization Access Group (WAAG) in order to read tokenGroups attributes from AD and use Kerberos S4U service
  - by default all Authenticated Users are members of Pre-Windows 2000 Compatible Access which limits the need for WAAG



ADFS installation #27

- On Windows 2016, enable IDPinitiatedSignOn page to test authentication
  - Set-AdfsProperties -EnableIdPInitiatedSignonPage $true



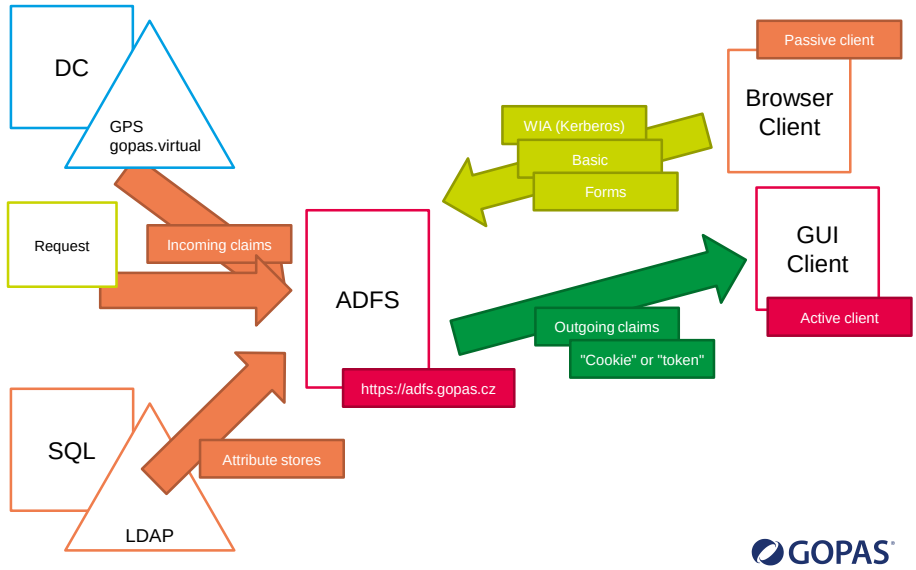
GOPAS

ADFS configuration notes

- Must be Domain Admins member to install ADFS
  - some stupid customer requirement
- Installer account must be sysadmin in DB if using full SQL
- ADFS service account gets servicePrincipalName
  - Domain Admins can write it, does not require self registration
- Creates and AD container
  - CN=Program Data,CN=Microsoft,CN=ADFS,CN=CertificateSharingContainer,DC=x
  - self-signed certificate private keys are stored here
- NETSH HTTP SHOW SSLCERT
- NETSH HTTP SHOW SERVICESTATE | findstr :443
- ADFS service account must be member of WAAG if user attributes are to be used as filters on incoming claims

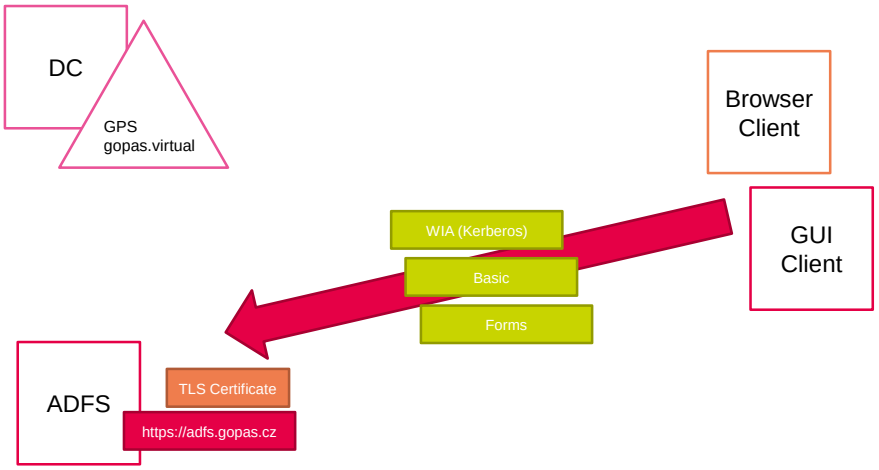
GOPAS

Simple ADFS terminology



GOPAS

ADFS internal testing



GOPAS

Testing ADFS from browser

- F12 developer toolbar (IE/Edge/Chrome)
  - does not show authentication headers
- Fiddler with TLS inspection
- Ctrl-Shift-DEL clear cookies (only)

GOPAS

Testing ADFS from browser (metadata)

- http://adfs.gopas.cz/adfs/probe
  - anonymous,
  - returns 200 OK, Server: Microsoft-HTTPAPI/2.0
- https://adfs.gopas.cz/federationmetadata/2007-06/federationmetadata.xml
  - requires **SNI**
  - SAML 2.0 metadata
  - anonymously available
  - digitally signed with XMLDSIG (similar to PKCS#7)
- https://adfs.gopas.cz/adfs/services/trust/mex
  - requires **SNI**
  - WS-Trust metadata
  - anonymously available
- https://adfs.gopas.cz/adfs/fs/federationserverservice.asmx
  - requires **SNI**, anonymous
  - ADFS1.0 web service metadata
- https://adfs.gopas.cz/adfs/ls
  - requires **SNI**, anonymous, returns error HTML with **illustration.png**
- https://adfs.gopas.cz/adfs/ls/idpinitiatedsignon

GOPAS

Quick ASCII, Base64 and URL reference

%3D =	%26 &	%2F /	%3F ?	
%3A :	%2B +	%3C <	%20 space	

```
[Reflection.Assembly]::LoadWithPartialName('System.Web')
[Web.HttpUtility]::UrlDecode( ' ') # from GET/POST params
[Web.HttpUtility]::UrlEncode( ' ')

[Web.HttpUtility]::HtmlDecode( ' ') # from HTLM FORM field
[Web.HttpUtility]::HtmlAttributeEncode( ' ')

[Web.HttpUtility]::ParseQueryString((New-Object Uri
'https://.../?a=1&b=2&c=3').Query)

# decoding SAMLP
[Text.Encoding]::ASCII.GetString(([Convert]::FromBase64Strin
g(([Web.HttpUtility]::UrlDecode( ' '))))))
```

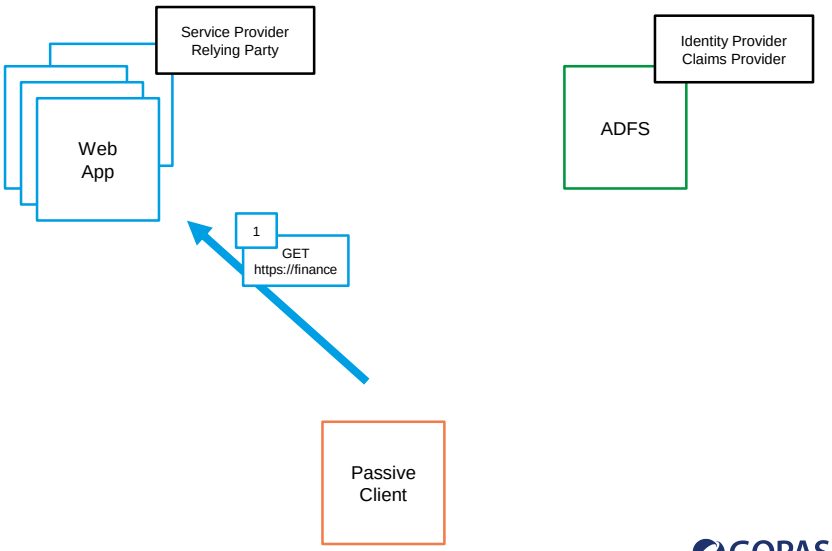
GOPAS

HTTP cookies generally

- Name=Value; Name=Value; ...
- Path=/subPath
  - limited to a subpath
- Domain=.gopas.cz
  - can enable cookie from a subdomain to go to other thirt-level subdomains
- Expires=23-May-2015 22:13:08 GMT
  - denotes persistent cookie
- Max-Age=[seconds]
  - expirations in browser are not enforced
  - servers expire cookies themselves
- Cleaning up a cookie = set empty value + expire
- HttpOnly
  - cannot be used by JavaScript
- Secure
  - requires HTTPS

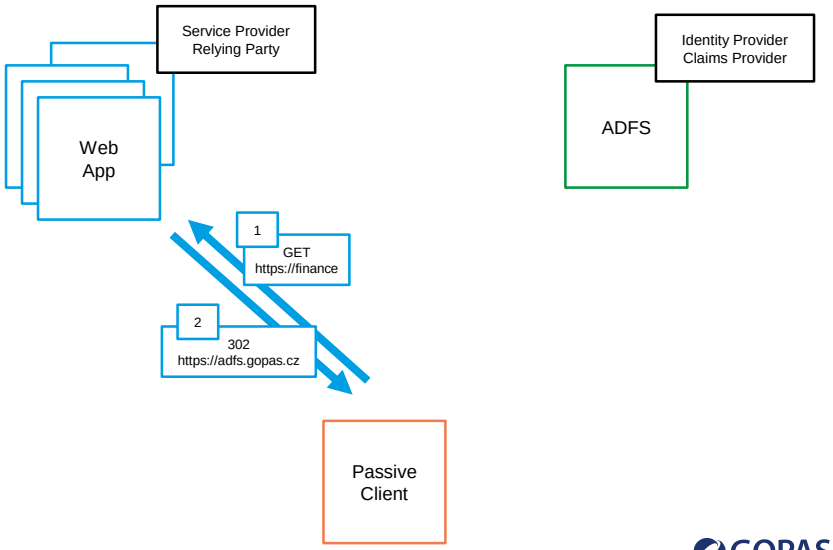
GOPAS

Passive client authentication



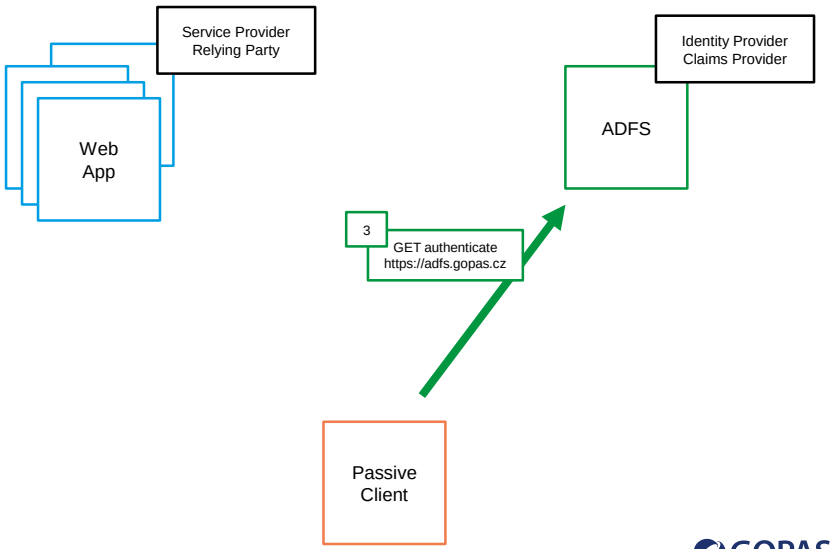
GOPAS

Passive client authentication



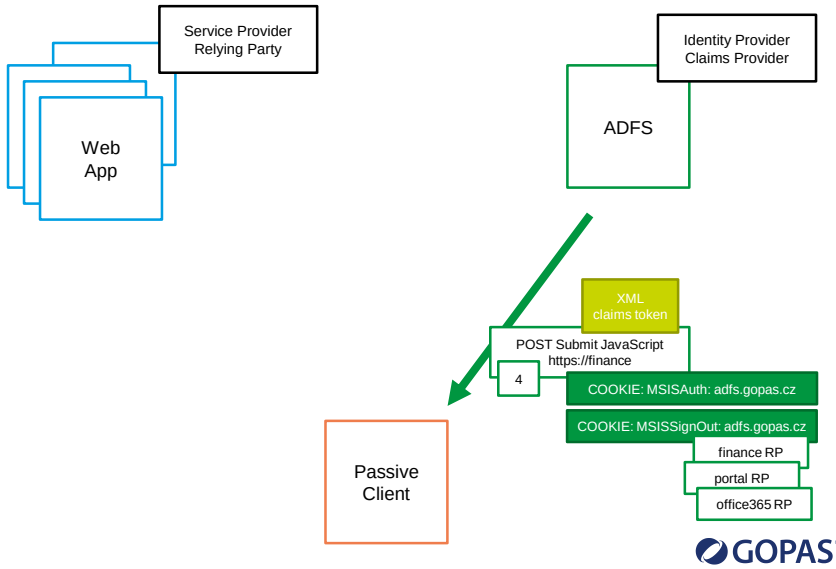
GOPAS

Passive client authentication

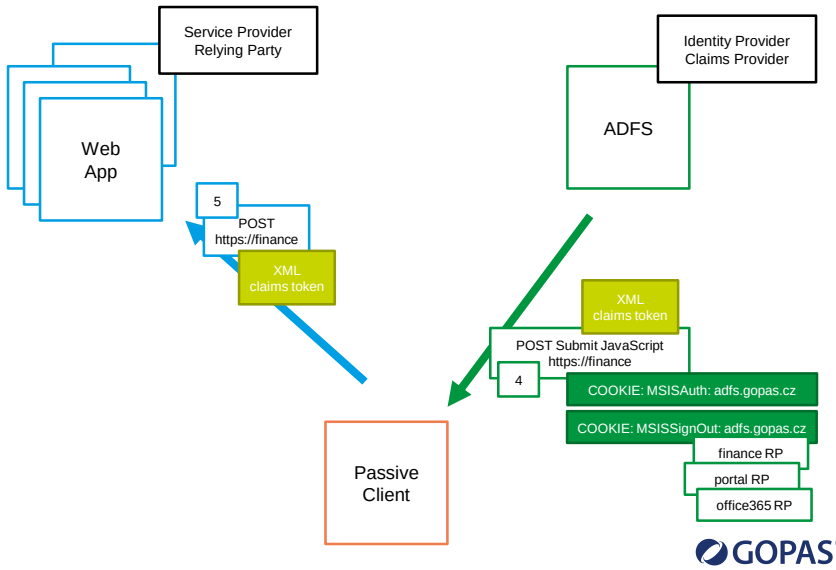


GOPAS

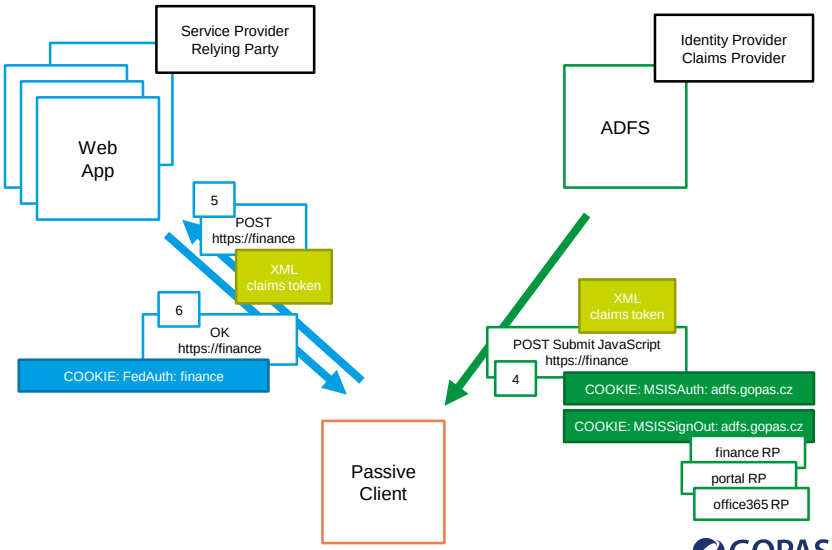
Passive client authentication



Passive client authentication



Passive client authentication



GOPAS

Testing ADFS from browser (authentication)

- https://adfs.gopas.cz/adfs/ls/idpinitiatedsignon.aspx
- https://adfs.gopas.cz/adfs/ls/idpinitiatedsignon
  - manually initiated from browser
  - file extension does not matter on 2012 R2+

- https://adfs.gopas.cz/adfs/ls?wa=wsignin1.0&wtrealm=https://portal.gopas.cz
  https://adfs.gopas.cz/adfs/ls/wia?wa=wsignin1.0&wtrealm=urn:fdc:sevecek.com:finance
  - WS-Federation passive sign-in URL, you receive SAML1.1 token
  - target 302 redirect configured as: WS-Federation Passive Endpoints on the Endpoints tab as Default
  - wtrealm = one of the relying party Identifiers

- https://adfs.gopas.cz/adfs/ls?wa=wsignin1.0&wtrealm=urn:fdc:sevecek.com:finance&wreply=https://portalinternal.gopas.cz
  - wreply = non-default target 302 redirect configured as: WS-Federation Passive Endpoints on the Endpoints tab

GOPAS

URI elements

- wtrealm
  - processed by the ADFS to determine relying party identifier for which the request came
- wreply
  - processed by the ADFS as the desired back redirection
  - must match one of the Trusted URLs on the Endpoints tab
- wctx, wct
  - values ignored by ADFS and just passed from requests to replies
  - storing client application context values
- wauth
  - &wauth=urn:oasis:names:tc:SAML:1.0:am:password (FBA)
  - &wauth=urn:federation:authentication:windows (WIA)
  - &wauth=urn:ietf:rfc:2246 (TLS client certificate)
  - &wauth=http://schemas.microsoft.com/claims/multipleauthn (request multifactor auth)
- whr
  - home realm claims provider explicitly named in URL
  - AD AUTHORITY, urn:fdc:books, ...

GOPAS

Standards

Name	What	How	Notes
WS-Federation	transport 302/POST redirects	passive clients	WIF (Windows Identity Foundation)
WS-Trust	transport	active SOAP clients	
SAML-P, SAMLP, SAML protocol	transport 302/POST redirects	passive client active SOAP clients	ADFS 3.0 no NETFX support
SAML 1.0	XML token format	used by WS-Federation urn:oasis:names:tc:saml:1.0:assertion	ADFS 1.0
SAML 1.1	XML token format	used by WS-Federation urn:oasis:names:tc:saml:1.0:assertion	ADFS 1.1
SAML 2.0	XML token format	used by SAMLP urn:oasis:names:tc:saml:2.0:assertion	ADFS 2.0
OAuth	transport + token format	active SOAP clients tokens JWT (JSON Web Token)	ADFS 3.0

Testing ADFS from browser (authentication)

- https://adfs.gopas.cz/adfs/ls?SAMLRequest=deflatedBase64request
  - SAML2.0 sign-in URL, returns SAML2.0 token
  - configured as: SAML Assertion Consumer Endpoints on the Endpoints tab
  - you can decode the "invalid" Base64 online at https://idp.ssocircle.com/sso/toolbox/samlDecode.jsp
    - <samlp:AuthnRequest ...
- https://adfs.gopas.cz/adfs/oauth2/authorize?response_type=code&client_id=11111111-2222-3333-4444-123456789012&redirect_uri=https://portal.gopas.cz&resource=https://portal.gopas.cz
  - OAuth sign-in URL, returns OAuth token, only for active clients
  - configured as: no endpoint plus use Get-AdfsClient and Add-AdfsClient

GOPAS

Note: MSISAuth cookie

- may be persistent if KMSI enabled on FBA authentication
  - 8 hours session vs. 24 hours persistent
- encrypted by ADFS farm wide encryption key
- contains only SAM login of the user
  - AD lookup is always performed by any ADFS farm member
  - uses Kerberos S4U
  - always updates group membership and attributes' store attributes
  - if UPN is changed, user is still logged-on
  - if SAM is changed, new logon dialog appears

GOPAS

Note: FedAuth cookie

- Encrypted by ASP.NET machine encryption keys
- By default stores the whole token (claims)
  - immune against farm member restart
  - shared among farm members
  - big
- Minimizing the cookie size
  - server-side session security token caching
  - implementing cache based on SessionSecurityTokenCache

GOPAS

Note: cookie encryption on web server farms

```
<system.web>
  <machineKey
    validationKey="SomeSHA1Key"
    decryptionKey="SomeAESKey"
    validation="SHA1"
    decryption="AES"
  />
</system.web>
```

```
byte[] utf8encoded = Encoding.UTF8.GetBytes(text);
byte[] protected = MachineKey.Protect(utf8encoded, "salt");
string urlEncoded = HttpServerUtility.UrlTokenEncode(protected);

// MachineKey.Unprotect(protected, "salt")
```

GOPAS

ADFS SSO cookie and RP token lifetime

- Default ADFS SSO cookie is 480 minutes (session) or 24 hours (if persistent)
- Set-AdfsRelyingPartyTrust -TokenLifetime [minutes]
  - default = 0 = 60 minutes!!

Testing ADFS from browser (sign-out)

- WS-Federation passive sign-out
  - https://adfs.gopas.cz/adfs/ls/?wa=wsignout1.0
    - cleaning up relying party trusts requires sign-out cookie to be generated during logon and always sent back: MSISSignOut
  - https://portal.gopas.cz/?wa=wsignoutcleanup1.0
    - cleans up the sign-out cookie on the claims-aware web site
  - https://portal.gopas.cz/?wa=wsignoutcleanup1.0&wreply=https://adfs.gopas.cz/adfs/ls/?wa=wsignout1.0
    - both in a single URL
- SAMLP 2.0 logout request
  - https://adfs.gopas.cz/adfs/ls/?SAMLRequest=deflatedBase64request
    - <samlp:LogoutRequest ...

GOPAS

Office365 passive client examples WS-Fed

- Metadata
  - https://nexus.microsoftonline-p.com/federationmetadata/2007-06/federationmetadata.xml
- Target endpoints for passive client redirection
  - https://login.microsoftonline.com/login.srf
- Relying party identifiers allowed by Microsoft for WS-Fed
  - login.windows.net (invalid wtrealm format)
  - urn:federation:MicrosoftOnline
  - https://login.windows.net
  - microsoftonline.com (invalid wtrealm format)
- Passive WS-Fed login URI
  - https://adfs.gopas.cz/adfs/ls/?wa=wsignin1.0&wtrealm=urn:federation:MicrosoftOnline
- Signout-cookie MSISSignOut
  - signoutCleanup;urn:federation:MicrosoftOnline
  - Microsoft+Office+365+Identity+Platform
  - https://login.microsoftonline.com/login.srf
  - https://login.microsoftonline.com/login.srf

GOPAS

Office365 passive client examples SAML 2.0

- Metadata
  - https://nexus.microsoftonline-p.com/federationmetadata/saml20/federationmetadata.xml
- Target endpoints for passive client redirection
  - https://login.microsoftonline.com/login.srf
- Passive SAML login URI
  - https://adfs.gopas.cz/adfs/ls/?SAMLRequest=fVHfS8MwEH4X%2FB9C3rM1XbtuRzsYDmHgVJz44Iuk3QUDbVJzqcz%2F3nSizJe9fr%2Fuu7uSVNf2sB7Cu33CjwEpsGPXWoITUfHBW3CKDIFVHRKEBvbr3R2kkwR674JrXMvPLJcdigh9MM5ytt1U%2FK2QhayTOhVyqVOR1SoVC73MRJrVOebzutASo5RowK2loGyoeJrITCRSzJJnOQe5gFn%2BytkLeoq5kZ4knK1%2F59w4S0OHfo%2F%2B0zQx5IDHikfF6vgKsXJsDKd4vxp7azygV6MRdqbxjpwOD7Y1FsvpufbP3MN9XHK7eXStab7YrfOdCpdvMCLmIPRJCv3YmwLawKcx9mfM%2F4dE%2BBs%3D
  - <samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" ID="_7171b0b2-19f2-4ba2-8f94-24b5e56b7f1e" IssueInstant="2014-01-30T16:18:35Z" Version="2.0" AssertionConsumerServiceIndex="0" >
    <saml:Issuer>urn:federation:MicrosoftOnline</saml:Issuer>
    <samlp:NameIDPolicy Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"/>
    </samlp:AuthnRequest>

GOPAS

Updating ADFS singing certificate in Office365/Azure/Intune

- Import-Module MSOnline
- Get-Credential
- Connect-MSOLService
- Get-MSOLFederationProperty

- Set-MsolADFSContext -Computer localhost -LogFile
  - only LAN connection
  - PS remoting must be enabled on the ADFS server
    - Enable-PSRemoting -force on Windows 2008 R2-
- Update-MSOLFederatedDomain

GOPAS

Modern Authentication for Office 2013

- Install all updates!!!
- Enable on client

```
HKCU\Software\Microsoft\Office\15.0\Common\Identity
EnableADAL = DWORD = 1
Version = DWORD = 1
```

- Enable on Exchange Online

```
$cred = Get-Credential; Connect-MsolService -Cred $cred; Import-Module (Import-PSSession $(New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri https://ps.outlook.com/powershell/ -Credential $cred -Authentication Basic -AllowRedirection) -DisableNameChecking) -Global -DisableNameChecking
Set-OrganizationConfig -OAuth2ClientProfileEnabled:$true
```

- Enable endpoints on ADFS server
  - adfs/services/trust/mex
  - adfs/services/trust/2005/windowstransport (enabled by default)
  - adfs/services/trust/13/windowstransport (could be used instead)
  - if non enabled, Outlook uses WS-Federation redirection with web page GUI
    - plus can perform MFA
- Cleaning the cache
  - delete the whole Identity key

```
Remove-Item HKCU:\Software\Microsoft\Office\15.0\Common\Identity -Recurse
```

GOPAS

Authenticating into SharePoint

```
$domain = 'gopas.cz'
$realm = "urn:fdc:$($domain):201609:sharepoint:intranet"
$signIn = "https://adfs.$domain/adfs/ls"
$certFile = '\\dc\public\adfs-{0}-#01.cer' -f $domain.Replace('.', '-')
$idClaim = "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
$idClaimName = 'EmailAddress'

$signCert = New-Object System.Security.Cryptography.X509Certificates.X509Certificate2($certFile)
New-SPTrustedRootAuthority -Name "$domain ADFS Token Signing Certificate" -Certificate $signCert

$map1 = New-SPClaimTypeMapping -IncomingClaimType $idClaim -IncomingClaimTypeDisplayName
$idClaimName -SameAsIncoming
$map2 = New-SPClaimTypeMapping -IncomingClaimType
"http://schemas.microsoft.com/ws/2008/06/identity/claims/role" -IncomingClaimTypeDisplayName
"Department" -SameAsIncoming
$map3 = New-SPClaimTypeMapping -IncomingClaimType "urn:fdc:sevecek.com:201801:claims/city" -
IncomingClaimTypeDisplayName "City" -SameAsIncoming

$claims = @($map1, $map2, $map3)

$ap = New-SPTrustedIdentityTokenIssuer -Name "$domain ADFS Provider" -Description "$domain ADFS
User Authentication" -realm $realm -ImportTrustCertificate $signCert -ClaimsMappings $claims -
SignInUrl $signIn -IdentifierClaim $idClaim
```

GOPAS

SharePoint cookies

- Sliding cookie expiration 50 minutes before RP token expires

```
$sts = Get-SPSecurityTokenServiceConfig
$sts.LogonTokenCacheExpirationWindow = (New-TimeSpan -Min 50)
$sts.Update();
```

- Using session cookies instead of persistent ones (requires Office applications re-authentication)

```
$sts = Get-SPSecurityTokenServiceConfig
$sts.UseSessionCookies = $true
$sts.Update()

iisreset
```

GOPAS

Authenticating into SharePoint

- WS-Federation endpoint
  - https://sp.gopas.cz/_trust
- SP built-in sign-out
  - https://sp.gopas.cz/_layouts/15/SignOut.aspx
- WS-Fed sign-out from SP only
  - https://sp.gopas.cz/_trust/?wa=wsignoutcleanup1.0
- WS-Fed sign-out from ADFS and all apps
  - https://sp.gopas.cz/_trust/?wa=wsignout1.0

GOPAS

Testing ADFS from browser with Fiddler

- Get-AdfsProperties
- by default requires extended protection for WIA
- Set-AdfsProperties -ExtendedProtectionTokenCheck None

GOPAS

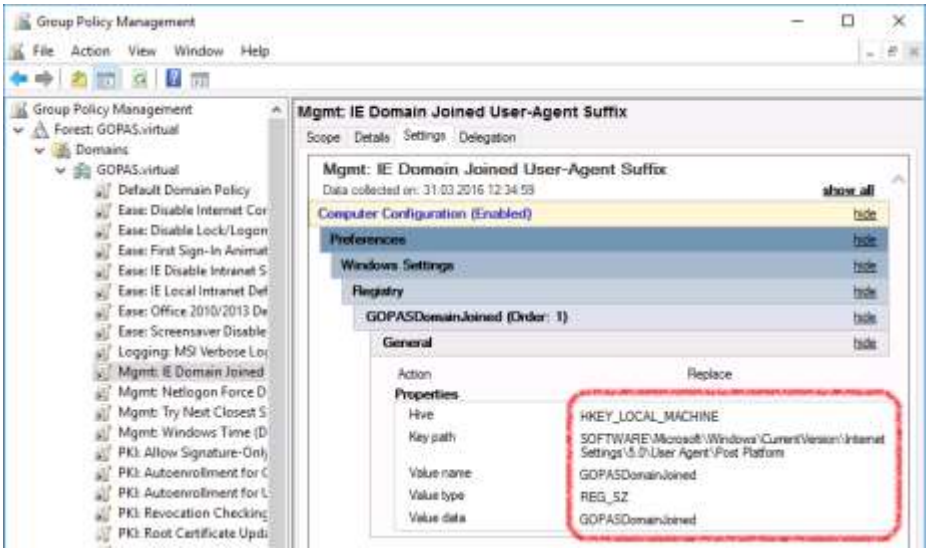
Testing ADFS from browser with FireFox or Chrome or Edge

- Firefox
  - disable extended protection for WIA
  - type 'about:config', filter for 'ntlm', add 'adfs.gopas.cz' to 'network.automatic-ntlm-auth.trusted-uris' setting
- FireFox, Chrome, Edge
  - WIASupportedUserAgents

ADFS 3.0 (2012 R2)	ADFS 4.0 (2016)
MSIE MSAuthHost/1.0/In-Domain Trident/7.0 MSIPC Windows Rights Management Client	MSAuthHost/1.0/In-Domain MSIE 6.0 MSIE 7.0 MSIE 8.0 MSIE 9.0 MSIE 10.0 Trident/7.0 MSIPC Windows Rights Management Client MS_WorkFoldersClient =~Windows\s*NT.*Edge

GOPAS

WIA for domain joined computers only



Testing ADFS from a GUI client

- use Fiddler to decrypt HTTPS
- use Windows Identity Foundation to request active responses
  - cannot produce SAML 2.0 (SAML-Protocol) cookie based responses

GOPAS

Azure MFA

- Requires
  - Azure AD Premium
  - or Intune (Mobile Device Management - MDM)
- Users register at:
  - https://aka.ms/MFAsetup
- App
  - Azure Authenticator

GOPAS

Initialize Azure MFA on Windows 2016

```
$tenant = 'sevecekeu201710.onmicrosoft.com'
$admin = "admin@$tenant"

# Note: this one identifies the AzureMFA service in MSOL
$appId = '981f26a1-7f43-403b-a875-f8b09b8cd720'

$selfSignedCert = New-AdfsAzureMfaTenantCertificate -TenantId $tenant

Connect-MsolService -Cred (Get-Credential $admin)

New-MsolServicePrincipalCredential -AppPrincipalId $appId -Type Asymmetric -
Usage Verify -Value $selfSignedCert

Set-AdfsAzureMfaTenant -TenantId $tenant -ClientId $appId
```

GOPAS

Requiring MFA for pre-2016 relying parties which do not use the new Access Control Policies

```
Get-AdfsRelyingPartyTrust

# Note: Require MFA for all requests
Set-AdfsRelyingPartyTrust -AdditionalAuthenticationRules '
=> issue(
Type =
"http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmethod",
Value = "http://schemas.microsoft.com/claims/multipleauthn"
);
'

# Note: Require MFA for both intranet and extranet
Set-AdfsRelyingPartyTrust -AdditionalAuthenticationRules '
c:[Type == "http://schemas.microsoft.com/ws/2012/01/insidecorporatenetwork", Value
== "false"]
=> issue(Type =
"http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmethod",
Value = "http://schemas.microsoft.com/claims/multipleauthn");

c:[Type == "http://schemas.microsoft.com/ws/2012/01/insidecorporatenetwork", Value
== "true"]
=> issue(Type =
"http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmethod",
Value = "http://schemas.microsoft.com/claims/multipleauthn");'
```

GOPAS

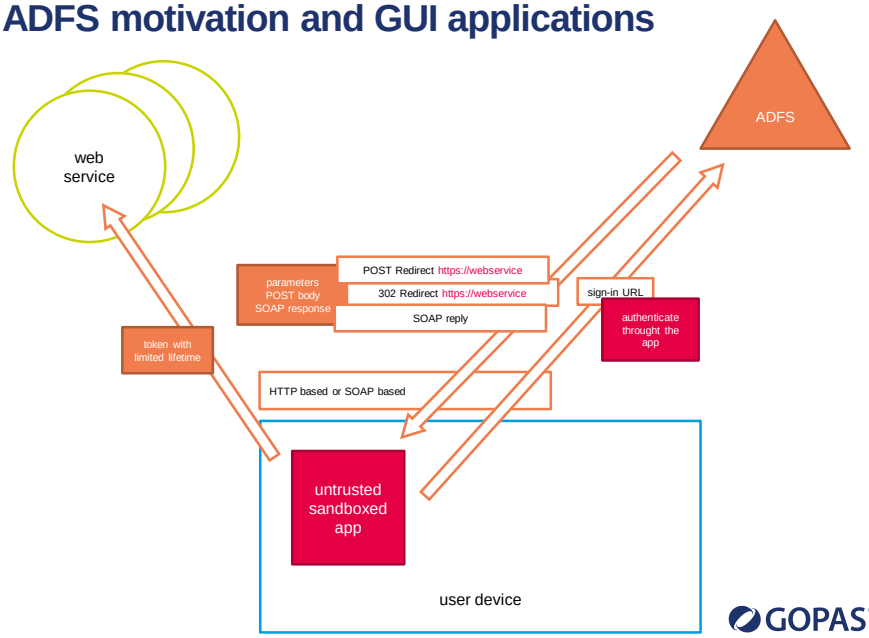
Disable double MFA pop-ups on Azure accounts which require MFA

```
Set-MsolDomainFederationSettings -DomainName gopas.cz -SupportsMfa $true

# Note: forward the claim
# http://schemas.microsoft.com/claims/authnmethodsreferences
# with value
# http://schemas.microsoft.com/claims/multipleauthn
#
# or you get a Loopback Detection error event 364
```

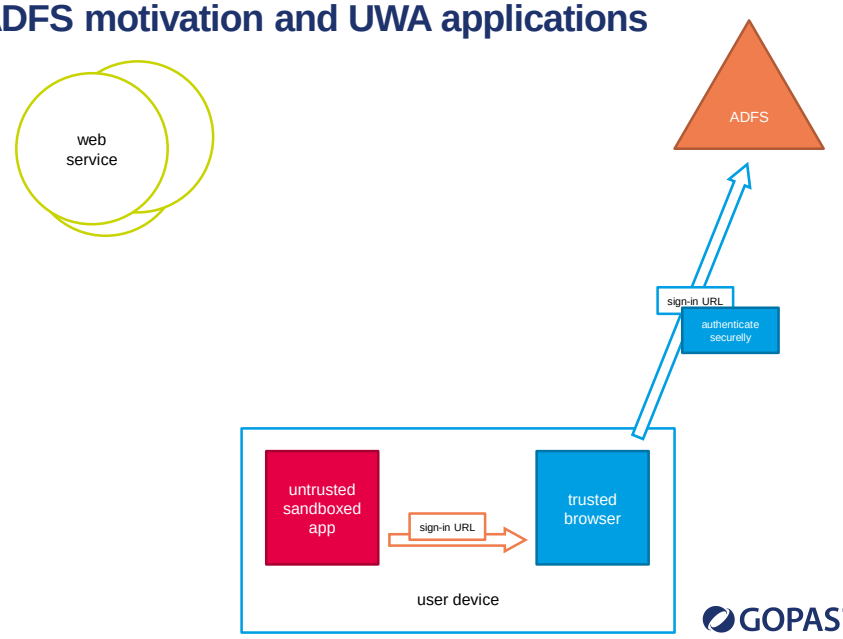
GOPAS

ADFS motivation and GUI applications



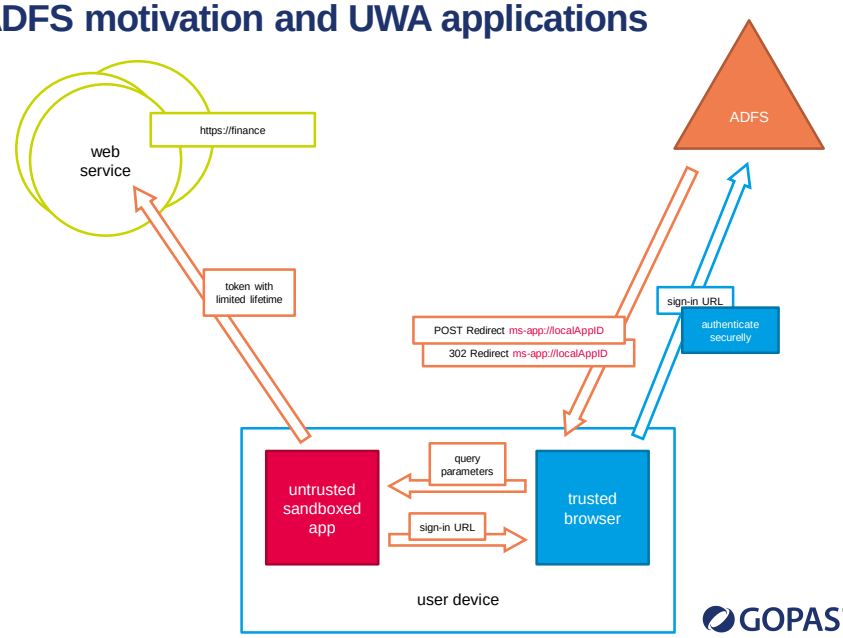
GOPAS

ADFS motivation and UWA applications



GOPAS

ADFS motivation and UWA applications



GOPAS

Claim rules

GOPAS

Claim members

- Type
  - http://schemas.xmlsoap.org/claims/UPN
  - urn:fdc:gopas.cz:201701:adfs/someClaim
- Issuer
  - AD AUTHORITY (primarySid, groupSid, ...)
  - LOCAL AUTHORITY (authenticationinstant, client cert thumbprint, subject, san, ...)
  - SELF AUTHORITY
  - urn:fdc:gopas.cz:201606:adfs-intranet
- OriginalIssuer
- Value
- ValueType
  - http://www.w3.org/2001/XMLSchema#string
  - http://www.w3.org/2001/XMLSchema#base64Binary
  - http://www.w3.org/2001/XMLSchema#date
  - http://www.w3.org/2001/XMLSchema#dateTime

- everything is case sensitive by default

- more claims of the same type can usually be generated and kept
  - things such as UPN, Name, windowsaccountname can have more items
  - except for NameID claim
  - exactly duplicate claims are removed

GOPAS

Claim rules (basics)

```
# general format to add outgoing claim (no OR operator exists)
# == equals, =~ match regex, $ end of string, ^ start of string,
(?i) ignore case

c1:[] && c2:[] => issue( ... c1.Value);

c1:[ Type == "...", Value == "..." ] &&
c2:[ Type == "...", Value =~ "..." ] &&
c3:[ Type == "..." ]
=> issue( Type = "...", Value = "..." + c1.Value )


issue( Issuer = c1.Issuer, OriginalIssuer = "..." )
Type == ".../multivalue", Value =~ "oneValue|secondValue"
Type == ".../ip", Value =~ "10\.10\.+"
```

GOPAS

Claim rules (basics)

```
# unconditional condition :-)
c1:[ ]

# copy the claim into outgoing claims
=> issue( claim = c1 );
```

GOPAS

Claim rules (aggregates)

```
# claim rules trigger for each individual incoming claim of the
same type, possibly issuing more claims

exists() && exists() && not exists() => issue()

exists([ ... ])
exists([
Type == "http://schemas.microsoft.com/2012/...", Value == "..."
       ])

count([ ... ]) >= 2
count([ Type == "http://contoso.com/proxyAddresses" ]) >= 2
count([ Type == "...", Value == "..." ]) >= 2
```

GOPAS

Claim rules (examples)

```
# user coming over proxy
exists([Type ==
"http://schemas.microsoft.com/2012/01/requestcontext/claims/x-ms-
proxy"])

# specific authentication endpoint
exists([Type ==
"http://schemas.microsoft.com/2012/01/requestcontext/claims/x-ms-
endpoint-absolute-path", Value ==
"/adfs/services/trust/2005/usernamemixed"])

# passive endpoint
exists([Type ==
"http://schemas.microsoft.com/2012/01/requestcontext/claims/x-ms-
endpoint-absolute-path", Value == "/adfs/ls/"])

# group membership by SID
exists([Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid",
Value == "S-1-5-21-domain-RID"])
```

GOPAS

Claim rules (examples)

```
# deny request
=> issue(Type =
"http://schemas.microsoft.com/authorization/claims/deny", Value =
"true");

# test for claim issuer
=> c:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsacc
ountname", Issuer == "AD AUTHORITY"]
```

GOPAS

Claim rules (advanced)

```
# Value expressions

= "..." + c1.Value + " ..."
= regexreplace(c1.Value, "...", "...")

# add claim among incoming claims to allow further processing
=> add(...)
```

GOPAS

Claim rules for Office365

```
c:[Type ==
"http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname"]
 => issue(store = "Active Directory", types =
("http://schemas.xmlsoap.org/claims/UPN",
"http://schemas.microsoft.com/LiveID/Federation/2008/05/ImmutableID"),
query = "samAccountName={0};userPrincipalName,objectGUID;{1}",
param = regexreplace(c.Value, "(?<domain>[^\\]+)\\(?<user>.+)", "${user}"),
param = c.Value);

c:[Type ==
"http://schemas.microsoft.com/LiveID/Federation/2008/05/ImmutableID"]
 => issue(Type =
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier",
Value = c.Value,
Properties["http://schemas.xmlsoap.org/ws/2005/05/identity/claimproperties/fo
rmat"] = "urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified");
```

GOPAS

Claim rules for Office365 (password expiration)

```
c1:[Type == "http://schemas.microsoft.com/ws/2012/01/passwordexpirationtime"]

⇒ issue(
store = "_PasswordExpiryStore",
types = (
  "http://schemas.microsoft.com/ws/2012/01/passwordexpirationtime",
  "http://schemas.microsoft.com/ws/2012/01/passwordexpirationdays",
  "http://schemas.microsoft.com/ws/2012/01/passwordchangeurl"
        ),
query = "{0};",
param = c1.Value
);
```

GOPAS

Additional chaotic notes

GOPAS

Old ADFS 2.0 notes

- IIS web site can use HTTP redirection to speed up login URL
  - disable redirection on the ADFS application
- Powershell ADFS module needs manual import
  - Add-PSSnapIn Microsoft.Adfs.PowerShell

GOPAS

ADFS certificates additional notes

- CRL checks
  - in order to be able to revoke the ADFS signing certificates
- Self-signed certificates
  - private keys stored in AD
  - auto rollover enabled, but must be trusted by the other party
  - Set-AdfsProperties -CertificateDuration
  - Update-AdfsCertificate -Urgent
  - Set-ADFSRelyingPartyTrust -EncryptionCertificateRevocationCheck -SigningCertificateRevocationCheck

GOPAS

ADFS farm member synchronization

- Get-ADFSSyncProperties
  - Get-ADFSSyncConfiguration on ADFS 2.0
- Preferred to use Windows Internal Database on each farm member separately
  - can use remote SQL server
- Secondaries sync from primary ADFS read/write server over HTTP or HTTPS
  - by default once per 5 minutes
  - http://primaryadfs.gopas.virtual/adfs/services/policystoretransfer (SOAP) or WCF net.tcp://primaryadfs.gopas.virtual:1500
  - Set-ADFSSyncProperties -PollDuration

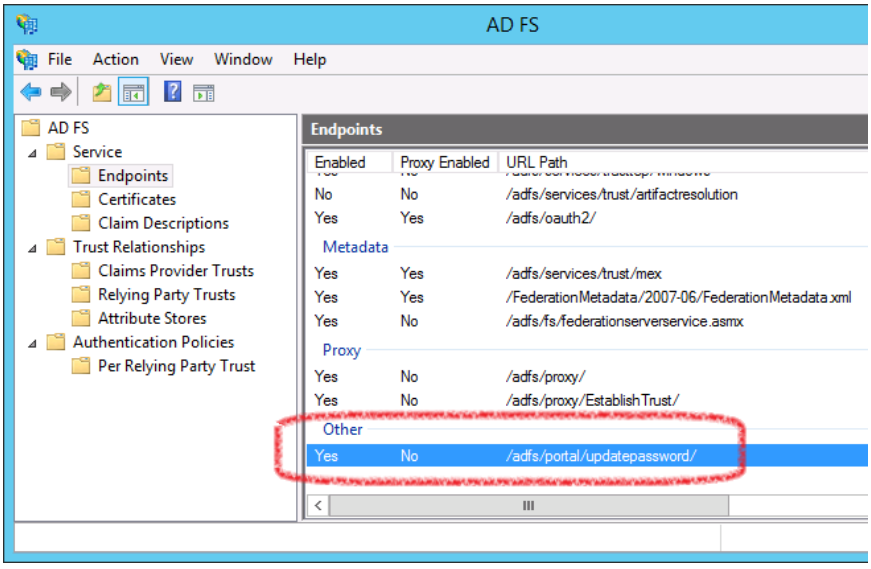
GOPAS

ADFS farm certificate requirements

- Single SSL certificate thumbprint is stored in configuration
  - all ADFS servers must use the same TLS certificate and the same Service Communication certificate
- In case of WIA through WAP with Extended protection for authentication enabled
  - the WAP servers must use the same TLS certificates as the back-end ADFS servers

GOPAS

Password Change portal for intranet clients?



ADFS web pages visual customizations

- Cannot customize clientLogon.aspx nor discoverClientRealm.aspx on ADFS 3.0 anymore
- Set-AdfsGlobalWebContent
- Set-AdfsWebTheme
  - CompanyName, Logo, Illustration, StyleSheet
  - ErrorPageDescriptionText, ErrorPageAuthorizationErrorMessage
  - ErrorPageSupportEmail
- Custom themes
  - New-AdfsWebTheme -Name myOwn -SourceName default
  - Set-AdfsWebConfig -ActiveTheme

GOPAS

ADFS home real discovery (HRD) pages customizations

- Add UPN suffixes for easier startup
  - Set-AdfsClaimsProviderTrust -TargetName thePartner -OrganizationalAccountSuffix 'sevecek.com'
- Disable HRD for intranet locations
  - Set-AdfsProperties -IntranetUseLocalClaimsProvider $true

GOPAS

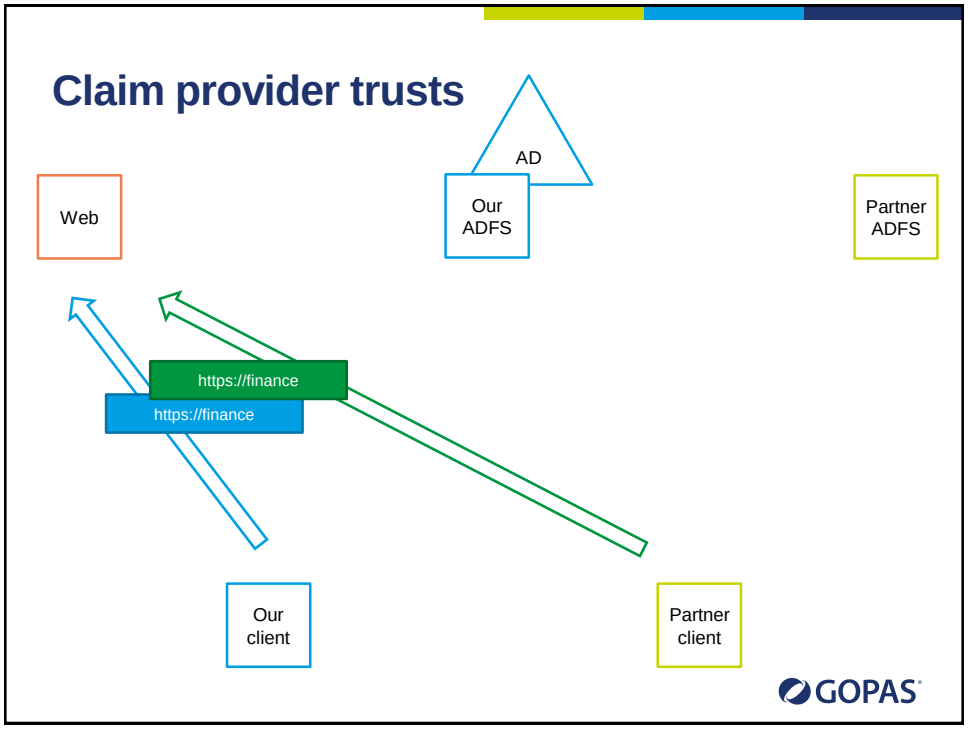
Alternative attribute stores

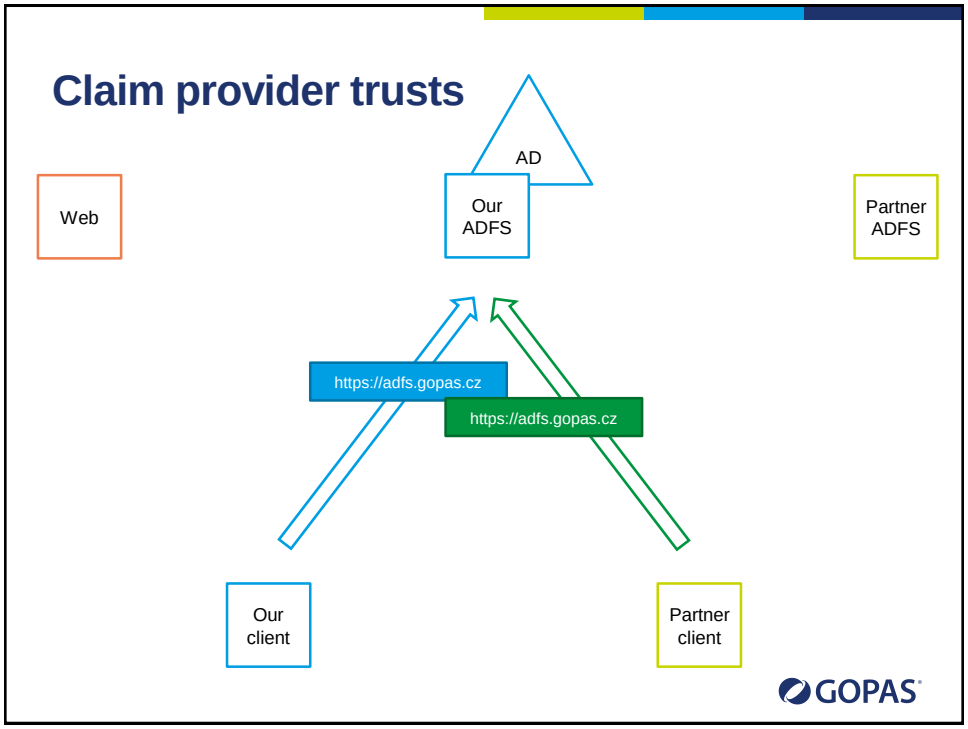
- LDAP connection string
  - LDAP://localhost:11111/cn=Users,o=GOPAS
  - ADFS authenticates against ADLDS with its service account
- SQL connection string
  - Server=GPS-DATA;Database=PartnerAccounts;Integrated Security=True;Encrypt=True

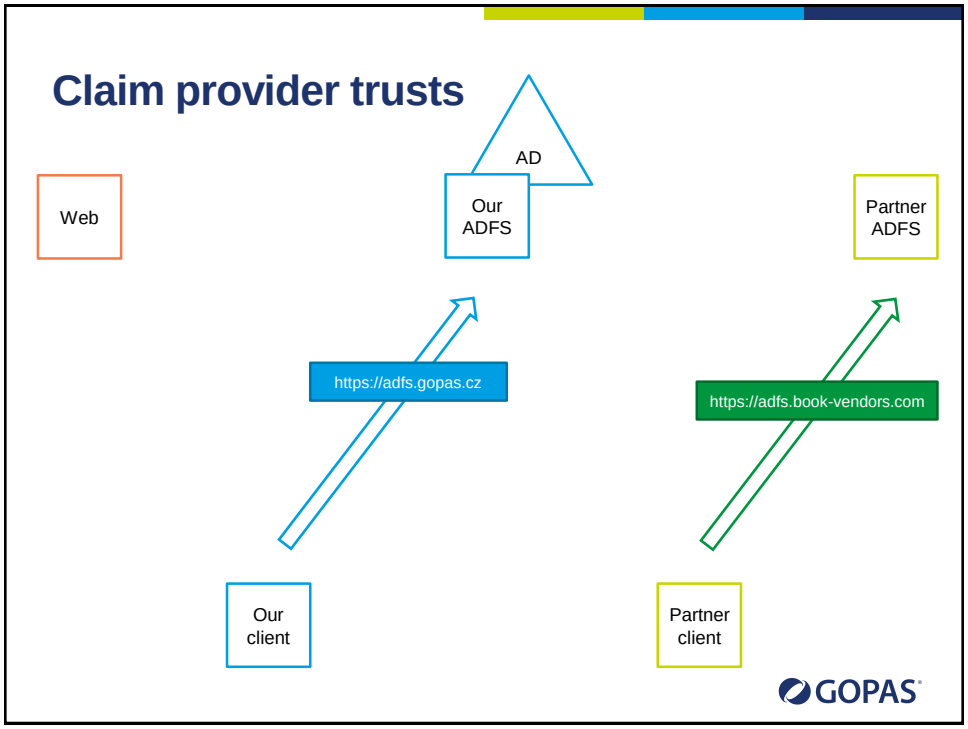
GOPAS

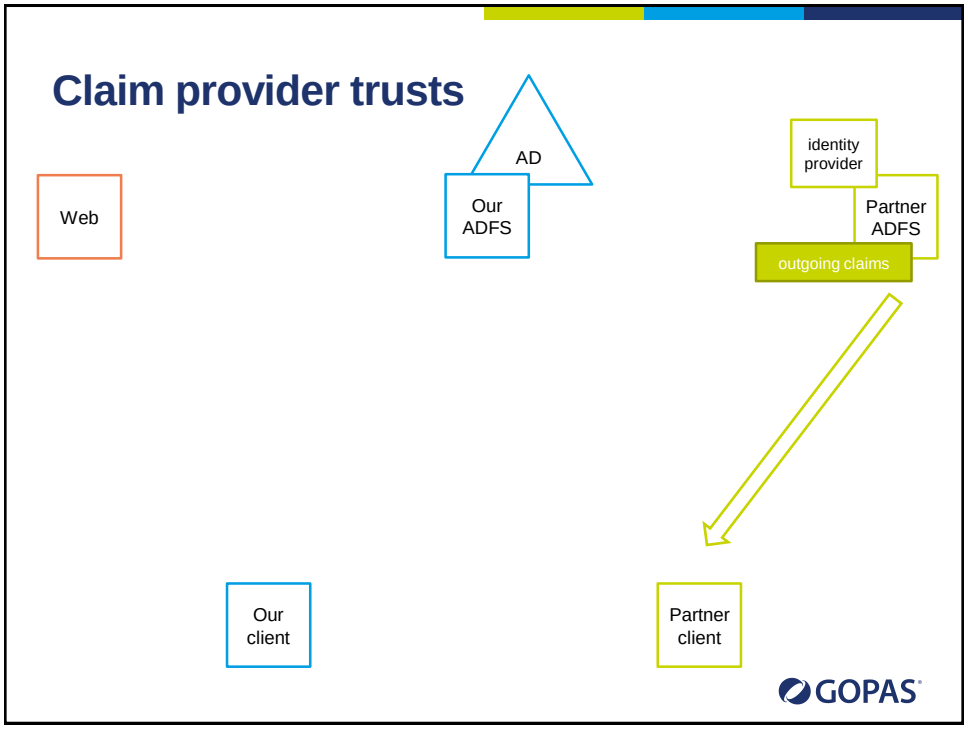
Third-parties aka claim providers

GOPAS

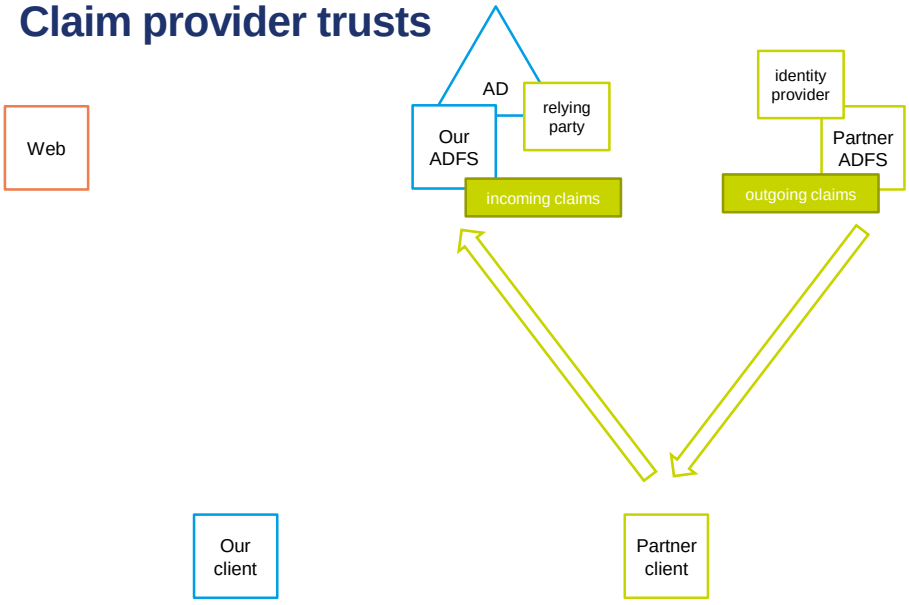

Claim provider trusts
AD
Web
Our ADFS
Partner ADFS
https://finance
https://finance
Our client
Partner client
GOPAS


Claim provider trusts
AD
Web
Our ADFS
Partner ADFS
https://adfs.gopas.cz
https://adfs.gopas.cz
Our client
Partner client
GOPAS


Claim provider trusts
Web
AD
Our ADFS
Partner ADFS
https://adfs.gopas.cz
https://adfs.book-vendors.com
Our client
Partner client
GOPAS

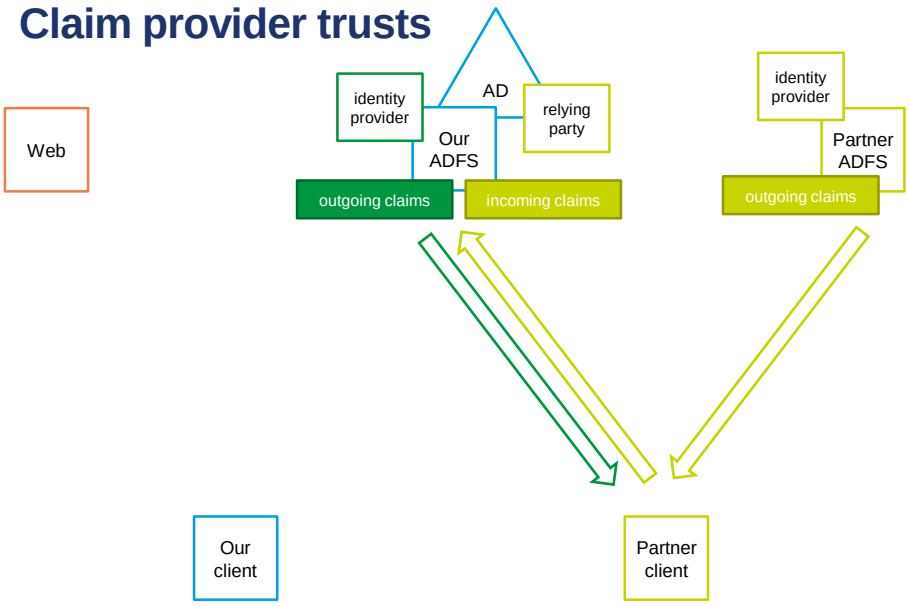

Claim provider trusts
Web
AD
Our ADFS
identity provider
Partner ADFS
outgoing claims
Our client
Partner client
GOPAS

Claim provider trusts



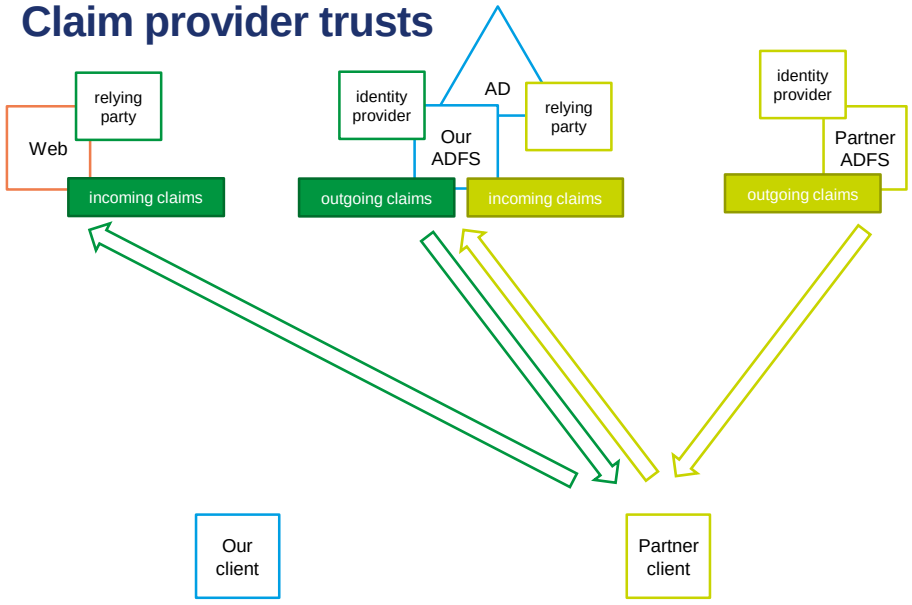
GOPAS

Claim provider trusts



GOPAS

Claim provider trusts



GOPAS

ADFS extranet scenarios with WAP reverse HTTPS proxy

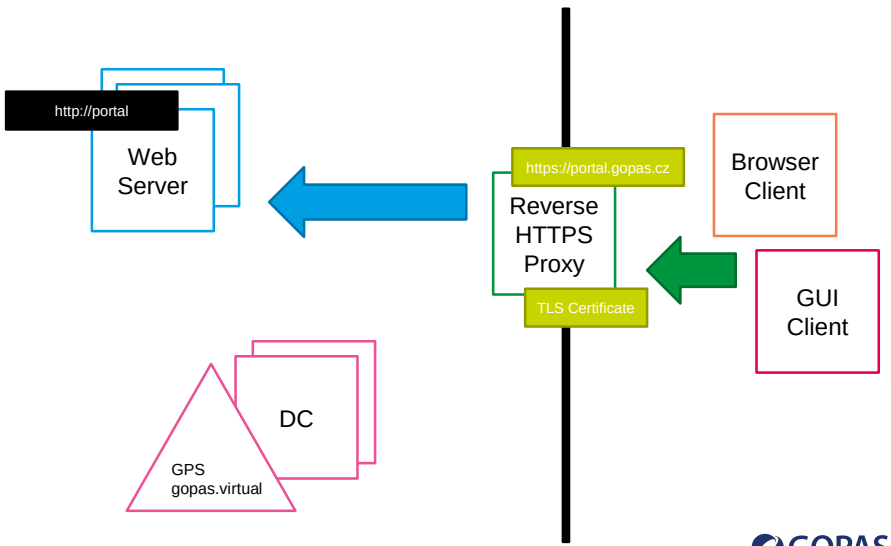
GOPAS

Motivation

- TMG (ISA) discontinued
  - TCP/IP/ICMP/IPSec/etc. inspection fully replaced with Windows Firewall
  - intrusion prevention filters included in Windows Defender and Microsoft Security Essentials
  - problematic expansion of reverse HTTPS publishing
- Secure reverse HTTPS publishing
  - Windows authentication at network perimeter
  - Forms-based (cookie) authentication with non-browser fallback to Basic and/or persistent cookie

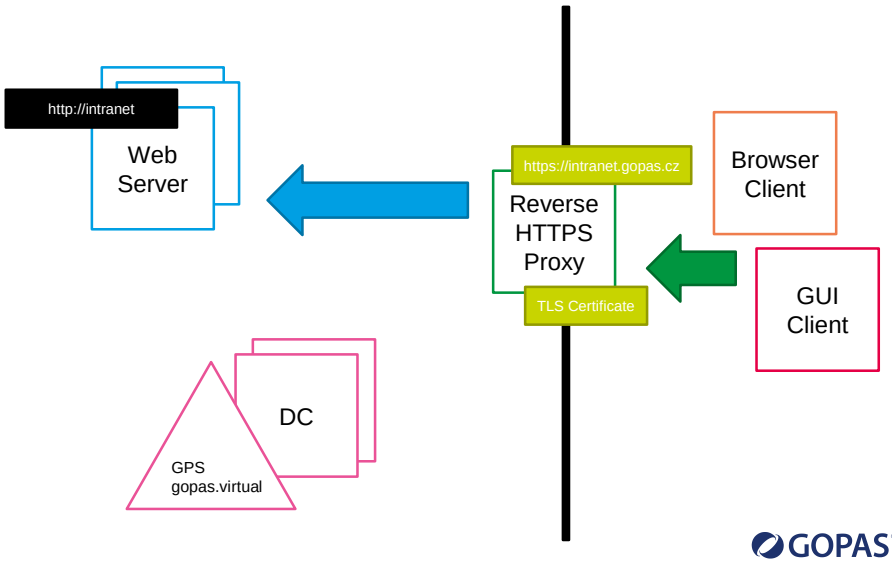
GOPAS

Principal scenario - Windows authentication web applications (internal HTTP or HTTPS)



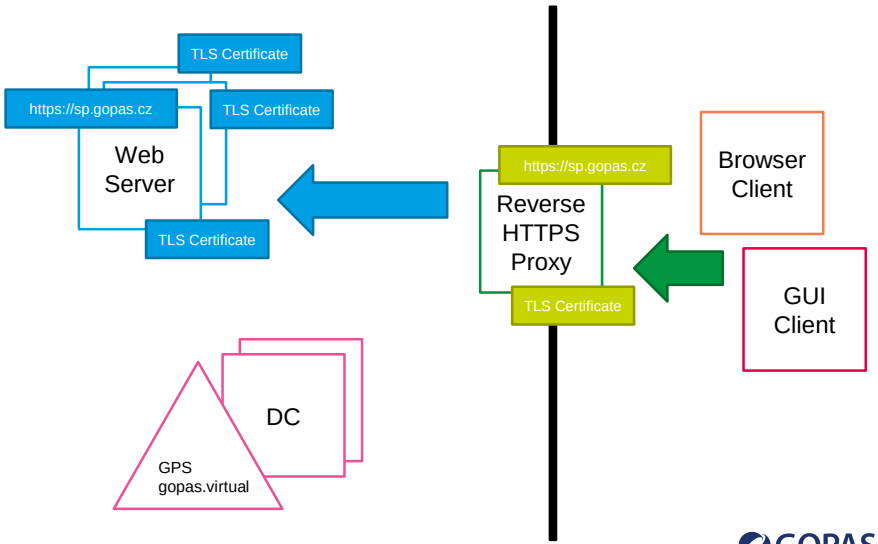
GOPAS

Principal scenario - Windows authentication web applications (SharePoint, Exchange, AAM)



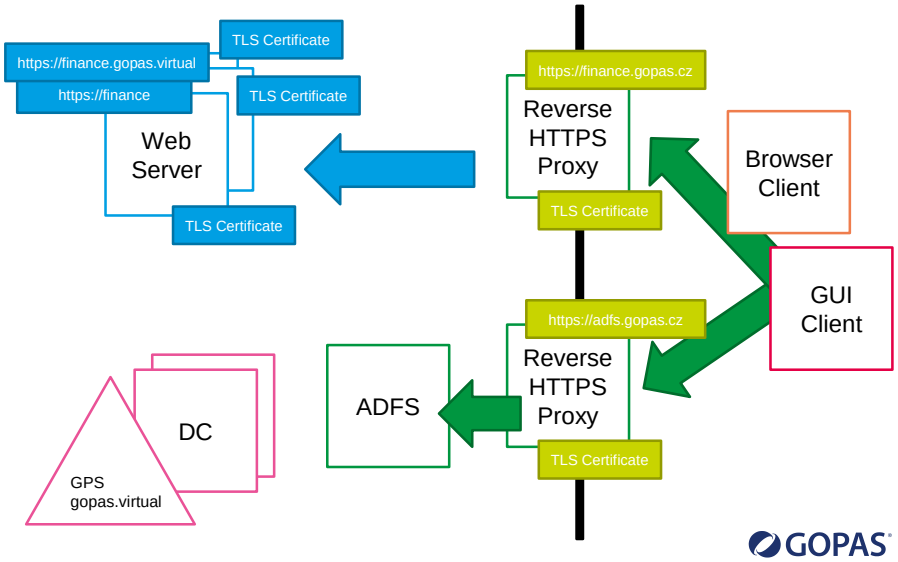
GOPAS

Principal scenario – Windows/claims web applications (no AAM necessary)

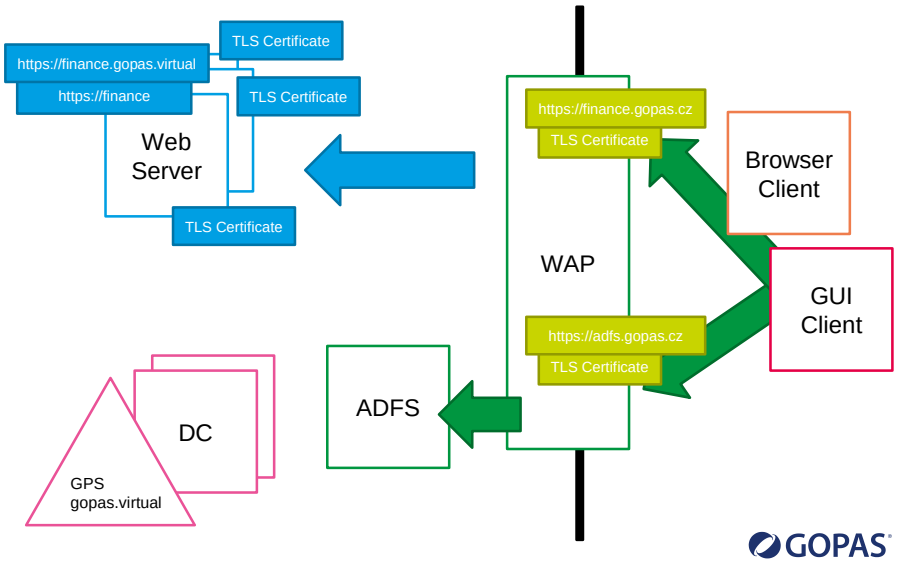


GOPAS

Principal scenario – claims aware web applications (internal HTTP or HTTPS)



Principal scenario – claims aware web applications (internal HTTP or HTTPS)



Another bit of motivation

- SharePoint
- not everything requires authentication
- HTTP level protocol exploits
  - many many many IIS modules to pass

GOPAS

Reverse HTTPS proxy general requirements

- Require HTTPS from client
  - possibly redirect to secure traffic
  - rather do not redirect to discourage HTTPS strip
  - minimize number of public TLS certificates
- Decrypt HTTPS at the perimeter
  - possibly inspect, define rules or extend with third-party
  - translate external URI to internal host names and paths
  - forward different host header
- Authenticate users at the perimeter
  - Windows authentication against Active Directory
  - allow other authentication databases if necessary
- Forward user credentials to the application
  - Windows authentication (WIA) delegation with Kerberos
  - claims with Windows Identity Foundation

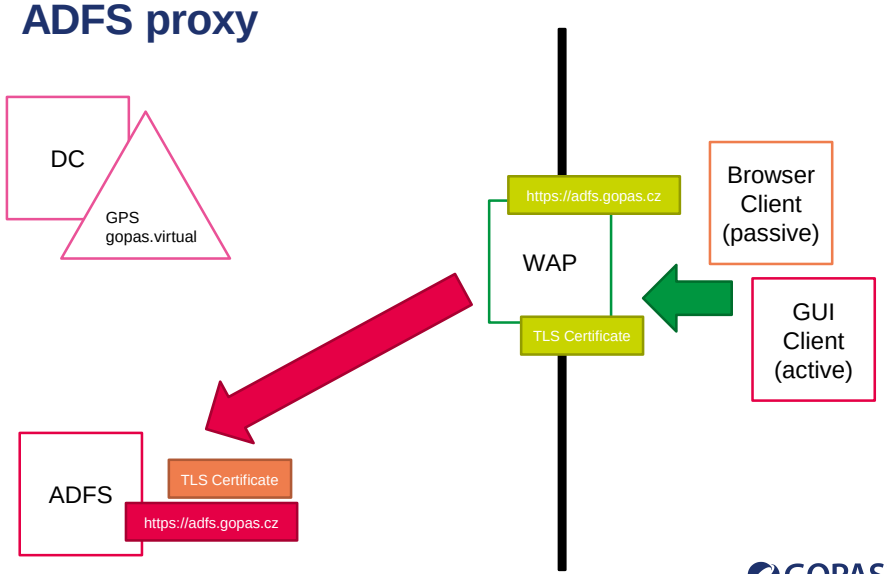
GOPAS

Web Application Proxy

- Require HTTPS from client
  - possibly redirect to insecure traffic (since 2016 only)
  - rather do not redirect to discourage HTTPS strip
  - minimize number of public TLS certificates
- Decrypt HTTPS at the perimeter
  - possibly inspect, define rules or extend with third-party
  - translate external URI to internal host names and paths
  - forward different host header
- Authenticate users at the perimeter
  - Windows authentication against Active Directory
  - allow other authentication databases if necessary
- Forward user credentials to the application
  - Windows authentication delegation with Kerberos
  - claims with Windows Identity Foundation
- TLS SNI as a bonus over TMG
  - plus Extended Protection for Authentication (NTLM mutual authenitcation)

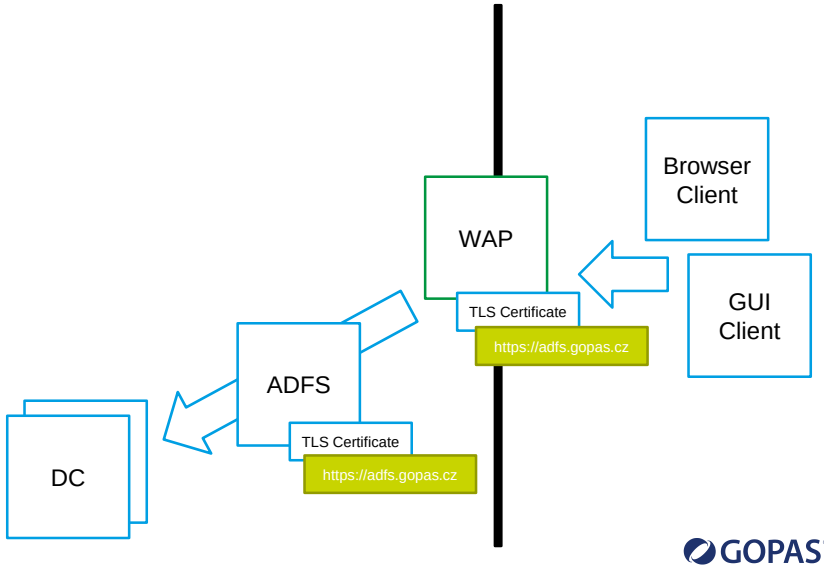
GOPAS

ADFS public access with WAP acting as an ADFS proxy



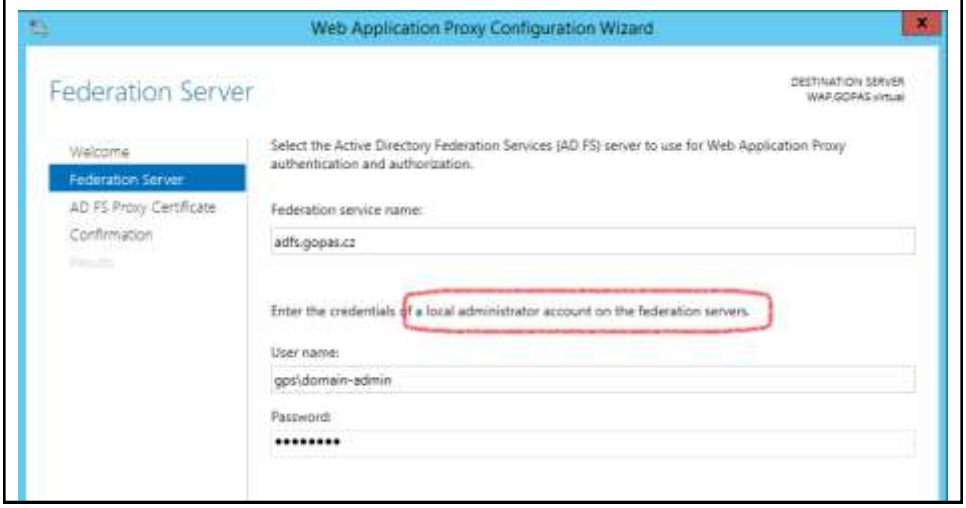
GOPAS

Publishing ADFS through WAP



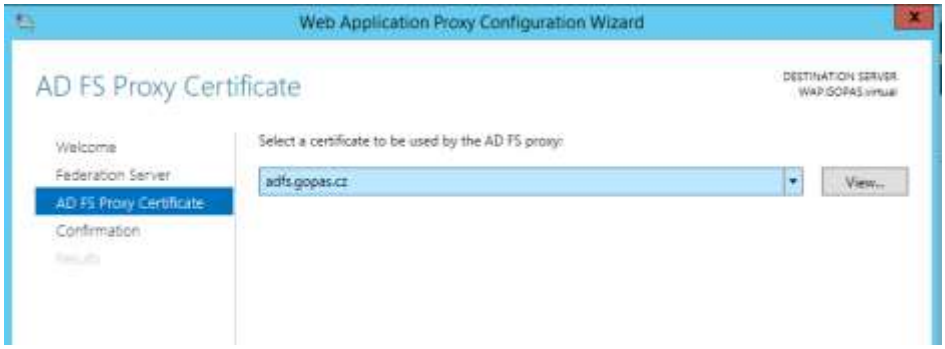
WAP installation #1

- Admin$ on the primary ADFS server
  - after installation HTTPS:443 only (TLS client cert.auth.)



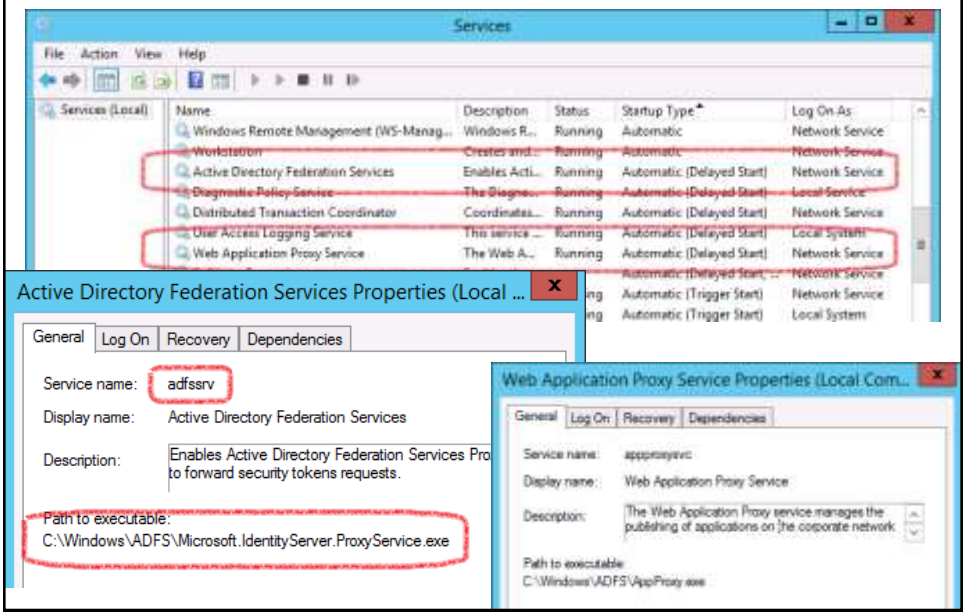
WAP installation #2

- Note: will use Kerberos even for SMB
  - SPN host/adfs.gopas.cz

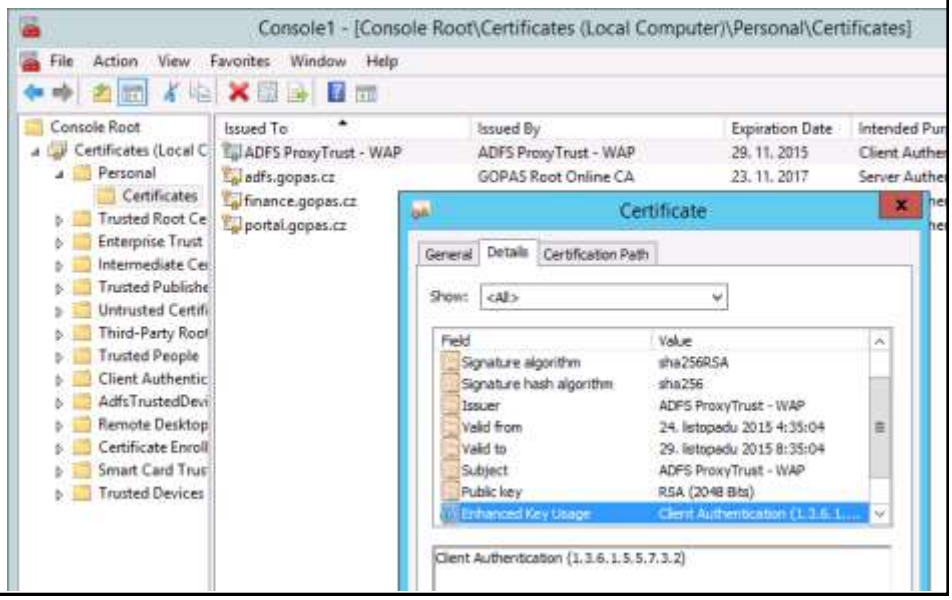


GOPAS

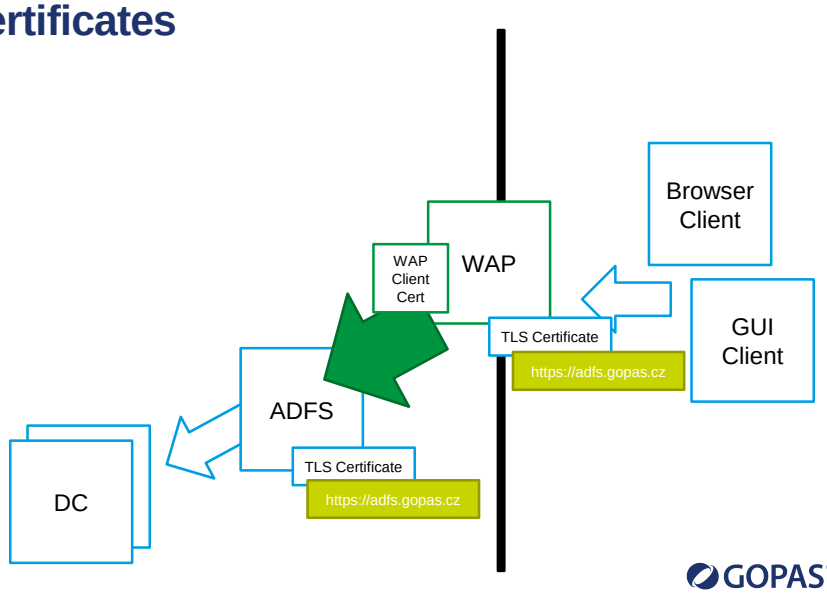
WAP installation #3



WAP installation #4 (WAP own client certificate)
5 (2012 R2) or 14 (2016+) days validity

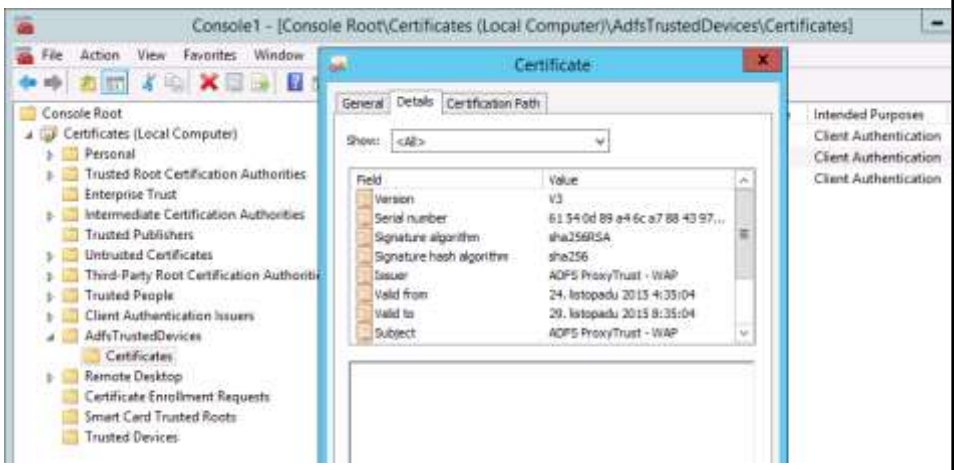


WAP to ADFS client authentication certificates



GOPAS

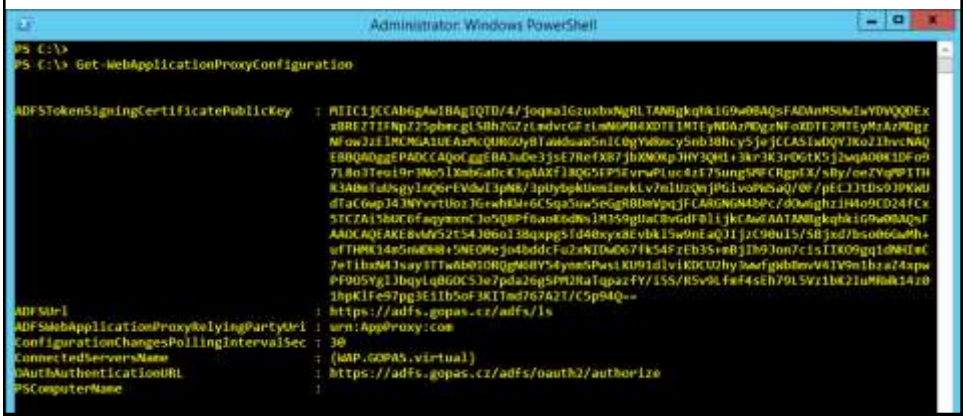
WAP installation #5 (WAP trusted on ADFS server)



GOPAS

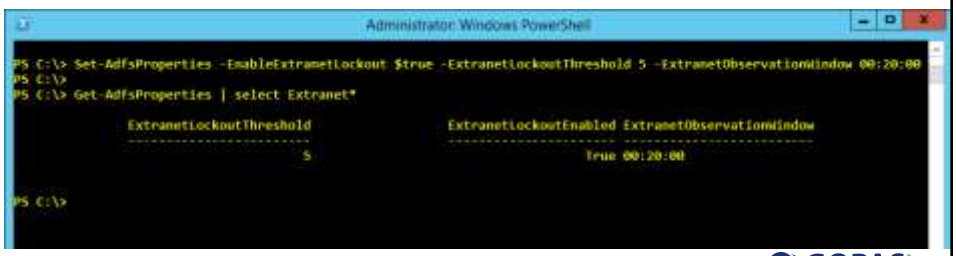
WAP installation #6 (ADFS certificates and URL on WAP)

- Get-WebApplicationProxyConfiguration
  - ADFSTokenSigningCertificatePublicKey, ConfigurationChangesPollingIntervalSet = 30
  - uses ADFS endpoints: https://adfs.gopas.cz/adfs/Proxy/webapplicationproxy/... and https://adfs.gopas.cz/adfs/Proxy/EstablishTrust



WAP installation #7 (ADFS 2012 R2 extranet lockout)

- Prevent external account lockout with lower threshold than what is on AD
  - Set-AdfsProperties -EnableExtranetLockout
  - checks only AD account for badPwdCount, badPasswordTime (prefers PDC if available)



GOPAS

WAP certificate notes

- Its own self-signed TLS client certificate
- Validates the ADFS TLS server certificate
- Does not use or validate the ADFS token-signing or token-decryption certificates when doing ADFS proxy
- Does validate ADFS token-signing certificates for published web applications
  - updates automatically from federation metadata
  - Set-WebApplicationProxyConfiguration -ADFSTokenSigningCertificatePublicKey

GOPAS

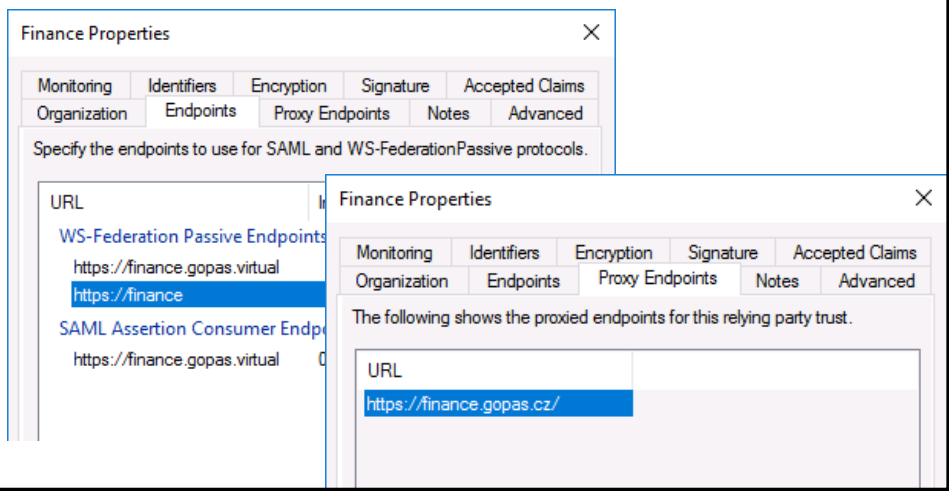
How ADFS knows what is internal and what is an external client

- ADFS proxy must forward requests with x-ms-proxy and x-ms-endpoint-absolute-path
  - you cannot simply proxy internal WAP-ADFS communication with Fiddler, because it is mutually authenticated
- Any reverse web proxy supported, not just WAP

GOPAS

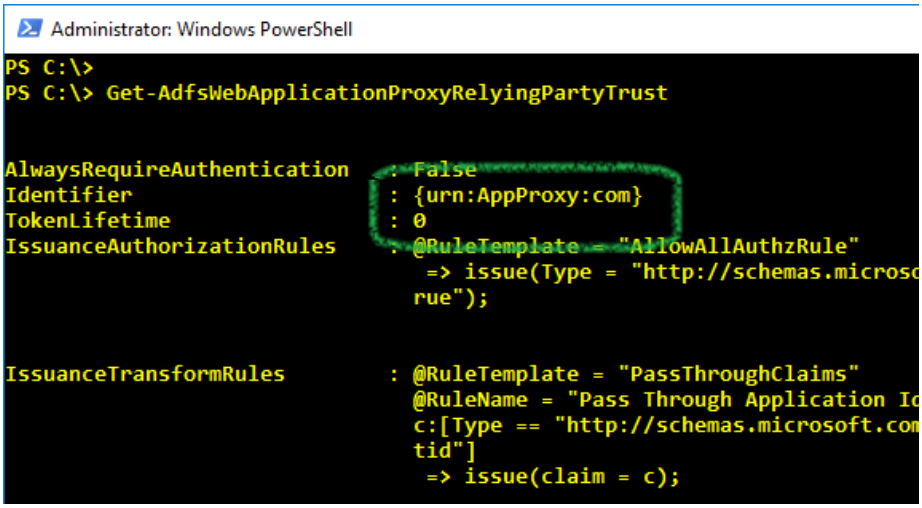
WAP published application with different host name

- ADFS itself generates correct absolute URL into the POST FORM ACTION as long as the wreply parameter is "valid"



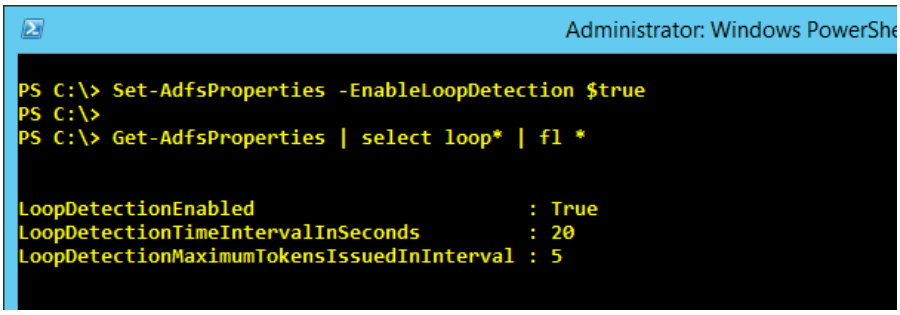
Hidden WAP relying party and EdgeAccessCookie timeout (default 60 minutes)

- Set-AdfsWebApplicationProxyRelyingPartyTrust -TokenLifetime



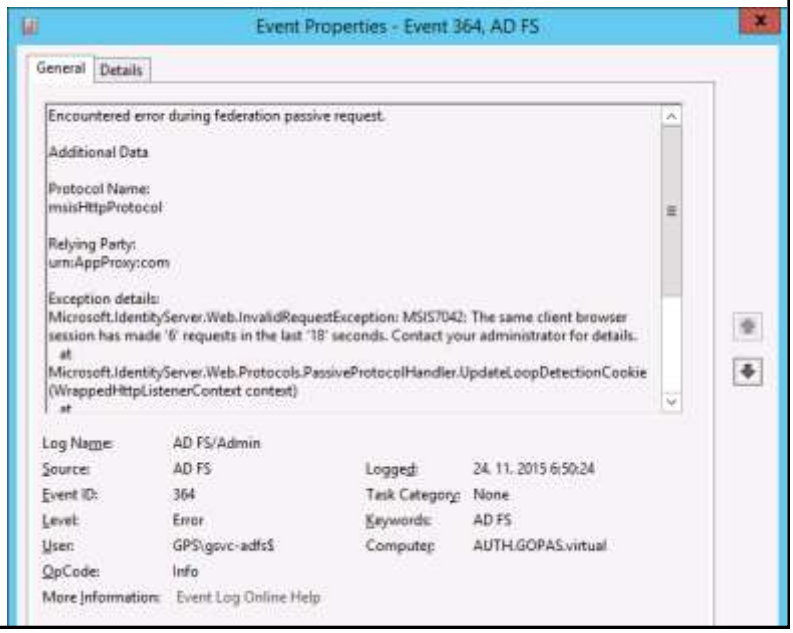
WAP publishing #8 (ADFS loopback detection)

- Set-AdfsProperties -EnableLoopDetection



GOPAS

WAP publishing #9 (testing ADFS loop detection)



WAP publishing #10 (persistent cookies)

- Set-AdfsProperties -EnableKmsi
  - "keep me signed in"
- Set-AdfsProperties -KmsiLifetimeMins
- Set-AdfsProperties -PersistentSsoCutoffTime <DateTime>
  - if you want to make all persistent cookies issued before the datetime invalid

GOPAS

WAP publishing #11 (enable password change page)

- Enable updatepassword endpoint
  - https://adfs.gopas.cz/adfs/portal/updatepassword

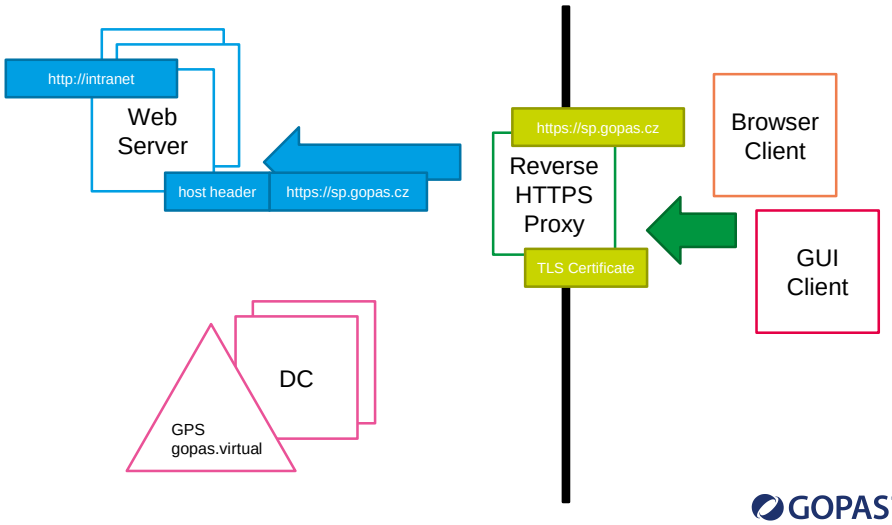
GOPAS

Publishing SharePoint

- Best practice to run internal SP web on public name since the very start
  - SharePoint must know the host name that client uses
- Running SharePoint on internal name
  - WAP should always forward with the external host header
  - WAP cannot define different host header for a different internal name/IP translation
  - WAP must use HOSTS or internal DNS records

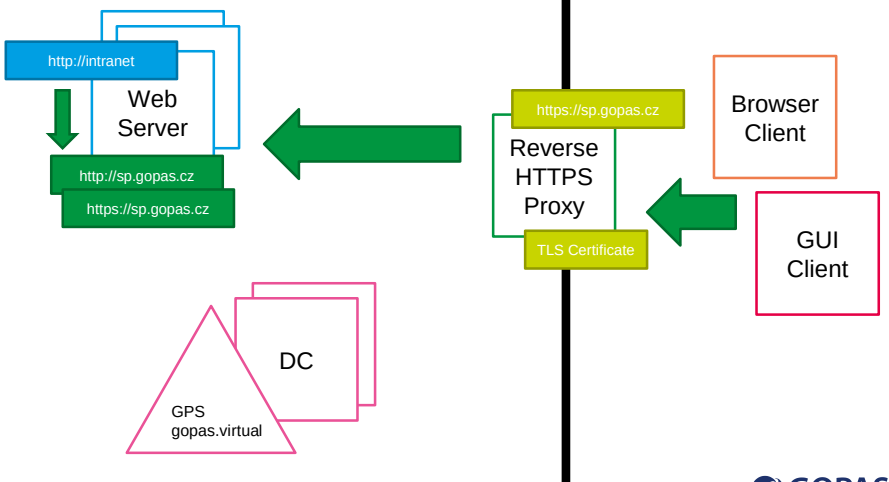
GOPAS

Scenario for SharePoint publishing ok if non-host header web binding or the same public/private host header (maybe AAM)



GOPAS

Extend web application first (maybe AAM) for host header web binding



GOPAS

WAP for WIA applications with Kerberos delegation

GOPAS

Wait. First make Kerberos work internally

AppPool	Kernel Mode Authentication	AD Account for Kerberos
Local System (SYSTEM)	on/off	GPS\WFE$
Local Service (NT AUTHORITY\Local Service)	on/off	no Kerberos
Network Service (NT AUTHORITY\Network Service)	on/off	GPS\WFE$
ApplicationPoolIdentity (IIS APPPOOL\apppool)	on/off	GPS\WFE$
GPS\svc-iis-canteen	on	GPS\WFE$
GPS\svc-iis-finance	off	GPS\svc-finance
SPS\sp-intranet-web	off	GPS\sp-intranet-web

GOPAS

Wait. First make Kerberos work internally

- Web server WFE
- Web application accessible at http://portal
- Application pool running under ApplicationPoolIdentity
- IIS Windows Authentication enabled, Kernel Mode Authentication enabled
- DNS name portal.gopas.virtual = A

- Set servicePrincipalName (SETSPN) on WFE
  - http/portal
  - http/portal.gopas.virtual

GOPAS

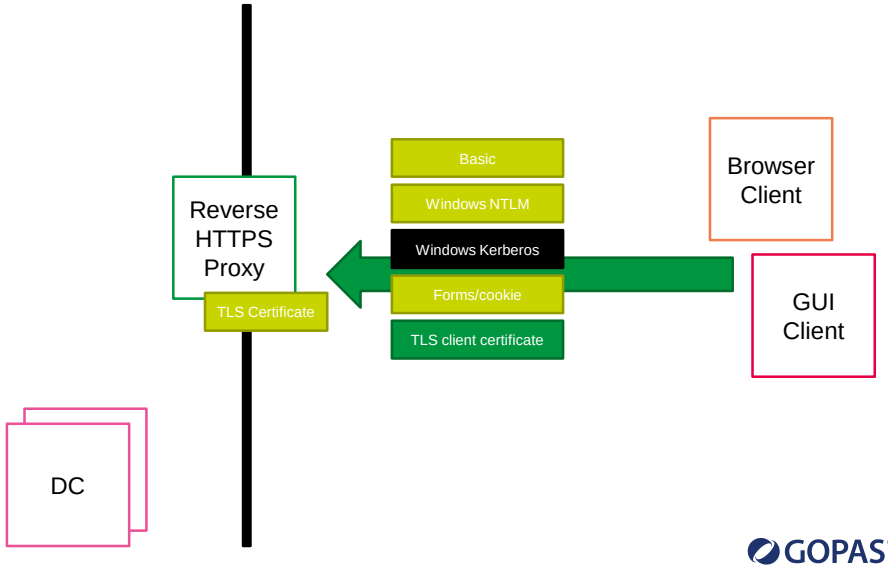
Wait some more. Yet make Kerberos work internally even for SharePoint

- Web server SP
- Web application accessible at http://intranet
- Application pool running under sp-intranet-web
- IIS Windows Authentication enabled, Kernel Mode Authentication disabled
- DNS name intrnaet.gopas.virtual = A

- Set servicePrincipalName (SETSPN) on sp-intranet-web
  - http/intranet
  - http/intranet.gopas.virtual

GOPAS

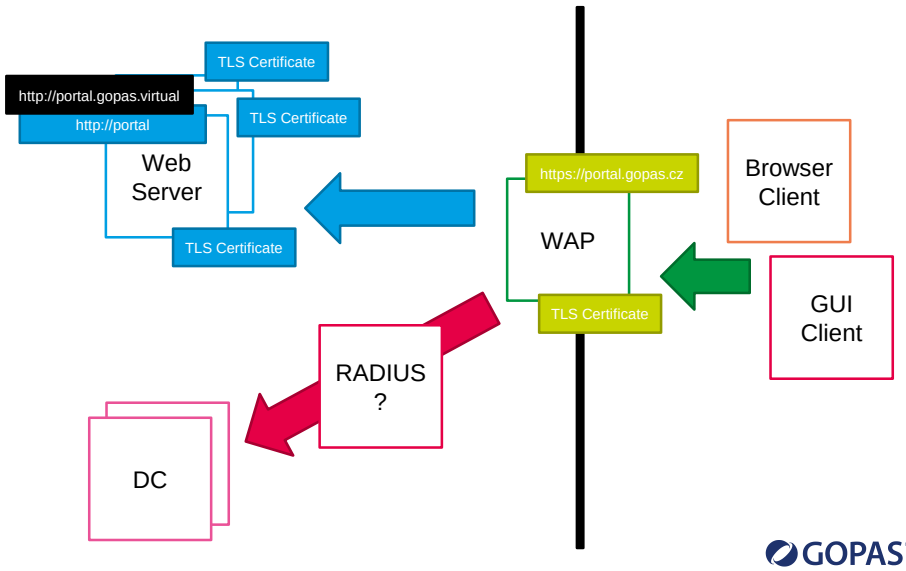
External authentication



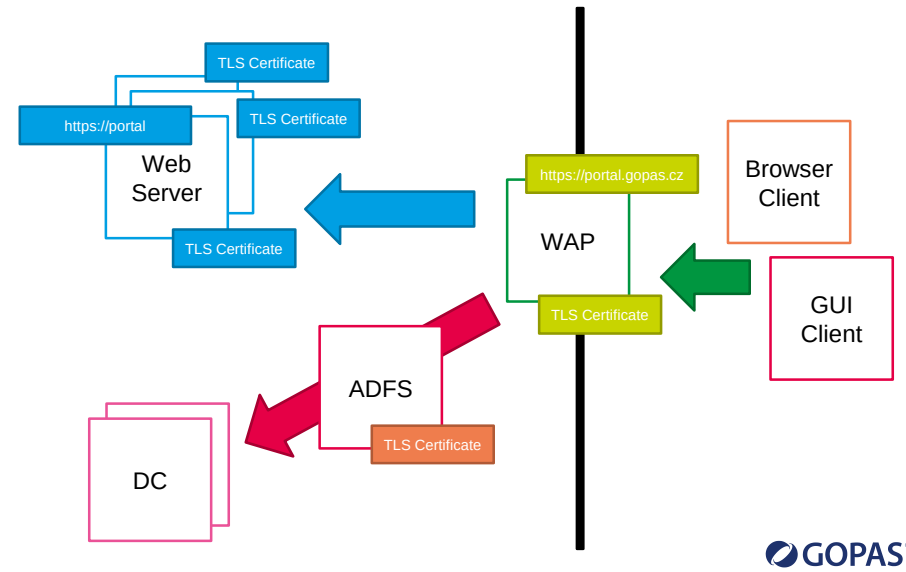
External authentication challenges

External authentication	Facts	Internal forwarding	Notes
Basic	plain-text TLS encrypted no SSO	easy	no browser sign-out no timeout non-browser clients
Windows NTLM	SSO	Kerberos constrained delegation	complicated sensitive
Windows Kerberos	not possible without direct contact with DC	Kerberos constrained delegation	impossible except for KDC Proxy
Forms/cookie	plain-text no SSO session vs. persistent cookie	easy claims SAML token	sign-out timeout browser clients
TLS client certificate	safe against password guessing safe against HTTP exploits	Kerberos constrained delegation claims SAML token	only for "partners" can use smart-cards both clients

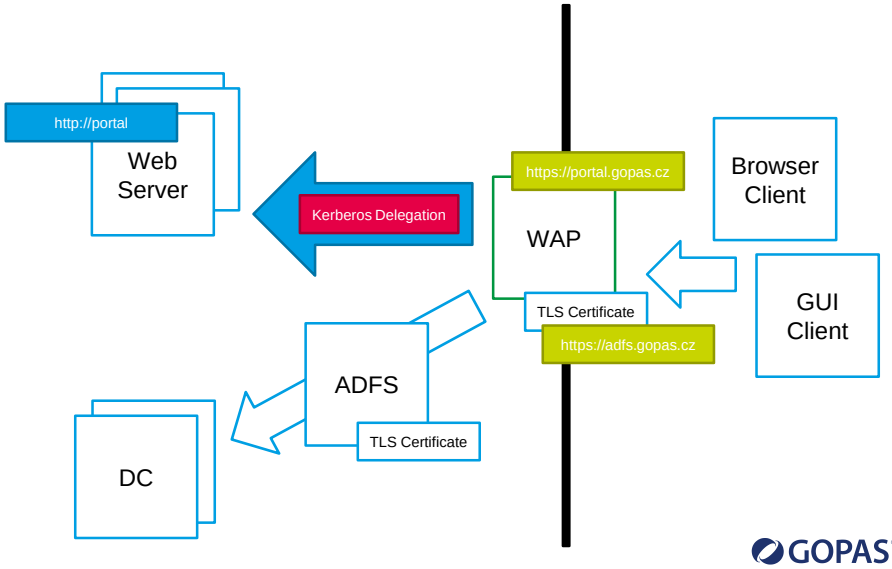
Scenario with an authentication server



Scenario with ADFS authentication server



Publishing simple WIA web application



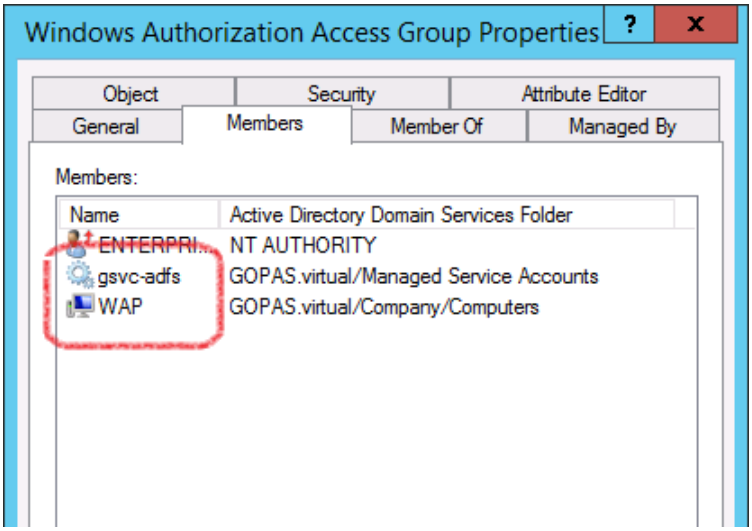
GOPAS

Kerberos delegation requirements

- Kerberos working internally WAP-WEB
  - http/portal
  - http/portal.gopas.virtual
  - or any arbitrary SPN specified in the WAP configuration
- Kerberos delegation for WAP server
  - Trust this computer to specified services only, Use any authentication protocol
  - WAP member of Windows Authorization Access Group (WAAG)
  - restart WAP machine

GOPAS

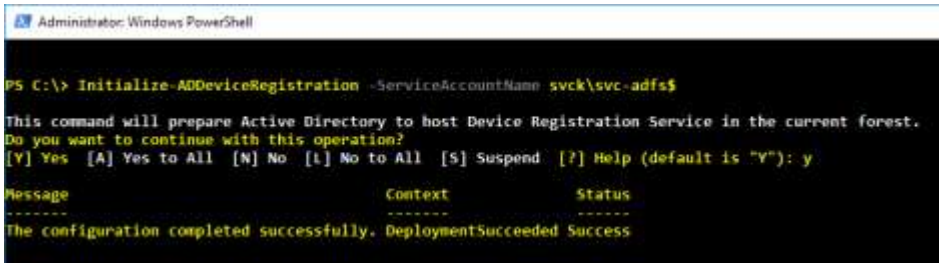
ADFS and WAP AD requirements



Workplace join aka Device registration

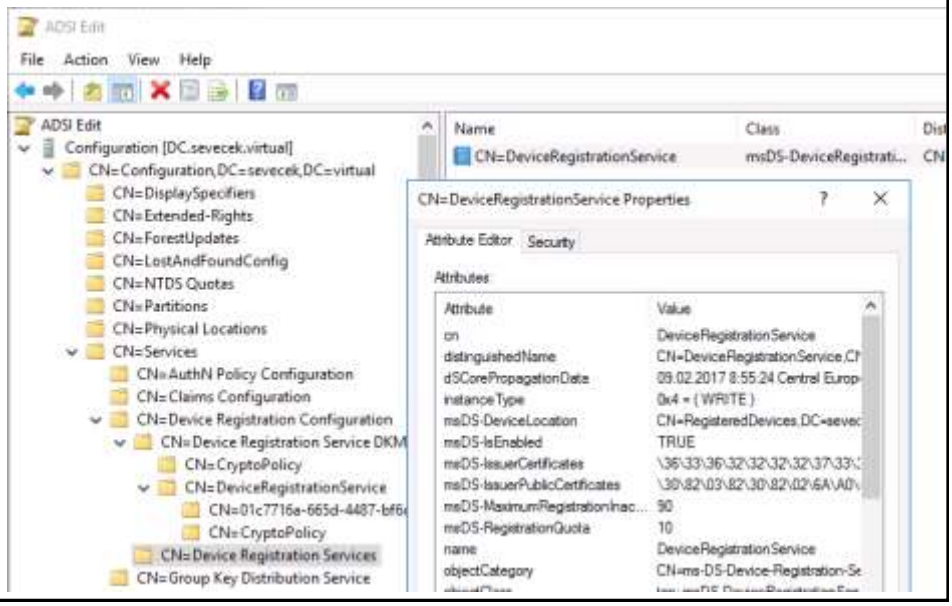
GOPAS

Initialize-ADDeviceRegistration



GOPAS

CN=DeviceRegistrationService,CN=Device Registration Services,CN=Device Registration Configuration,CN=Services,CN=Configuration



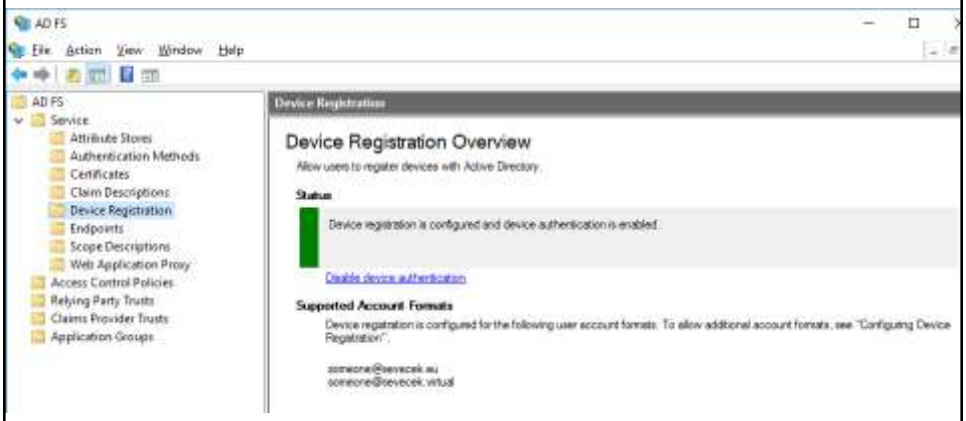
Enable-AdfsDeviceRegistration

```
Enable-AdfsDeviceRegistration

Set-AdfsGlobalAuthenticationPolicy -DeviceAuthenticationEnabled
$true
```

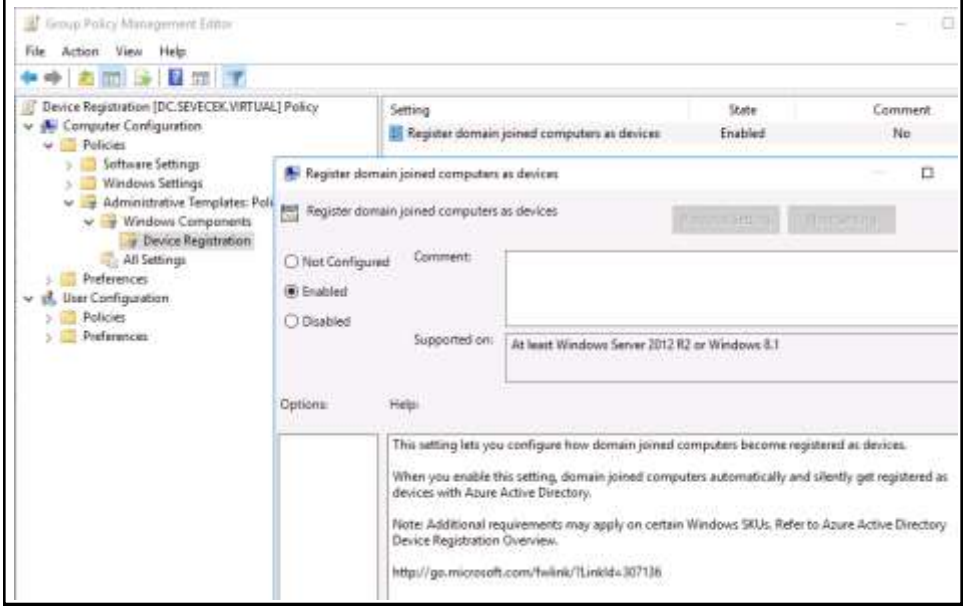
GOPAS

Device registration enabled in ADFS



GOPAS

Device Registration - Register domain joined computers as devices



OAuth 2.0

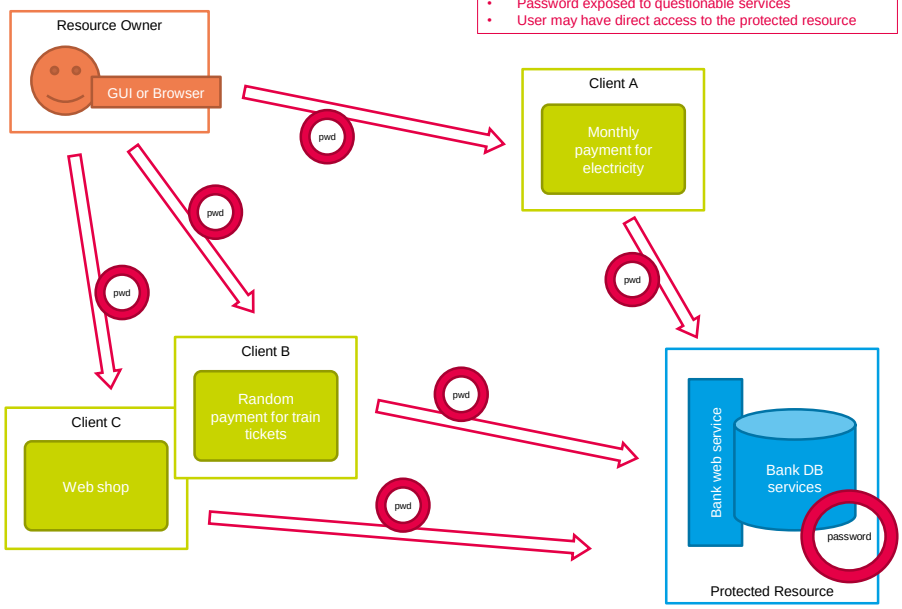
GOPAS

Basic motivation

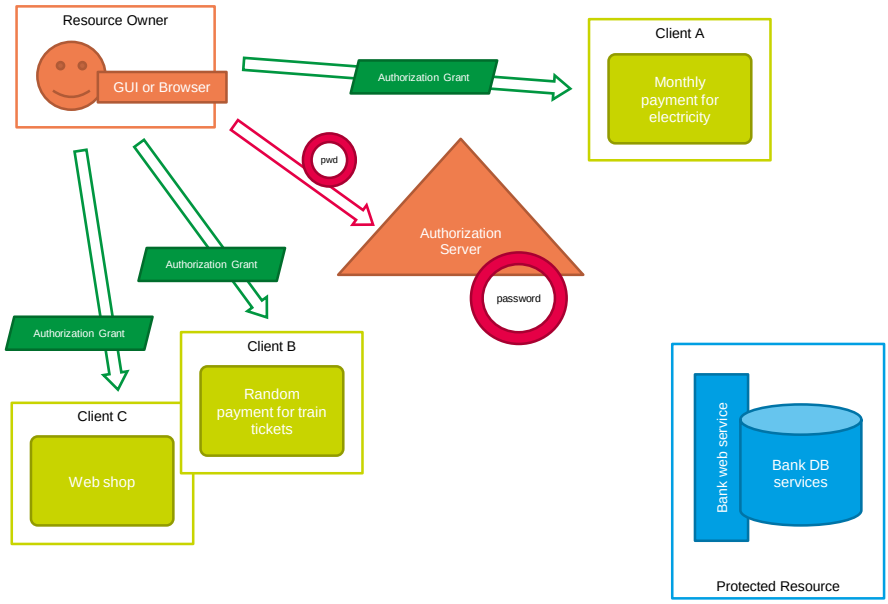
- Just another redirection protocol
  - "implicit grant"
- Different token format JWT
  - JSON Web Tokens
  - simpler and smaller
- Refresh tokens
  - issue a new access token based on a previously obtained refresh token

More complex motivation

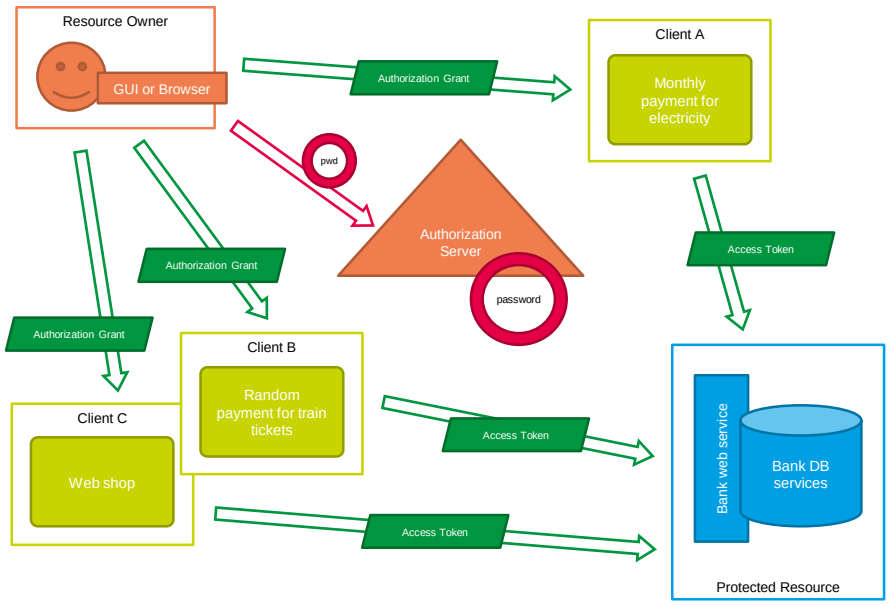
- Too permanent password which cannot be changed easily
- Cannot revoke single client
- Password exposed to questionable services
- User may have direct access to the protected resource



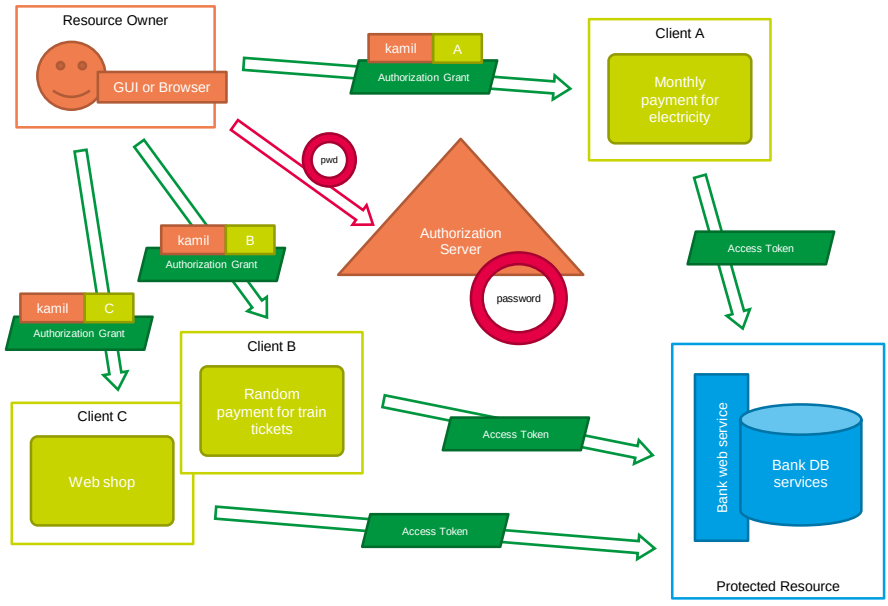
More complex motivation



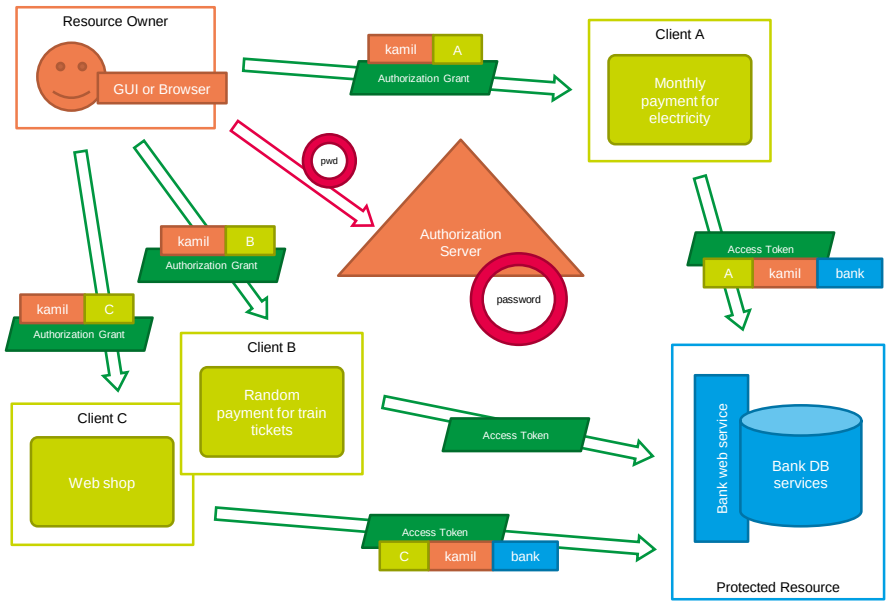
More complex motivation



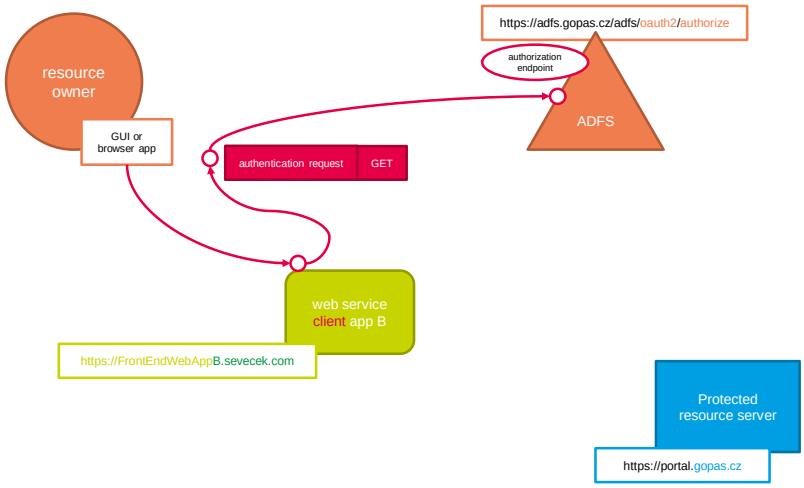
More complex motivation



More complex motivation

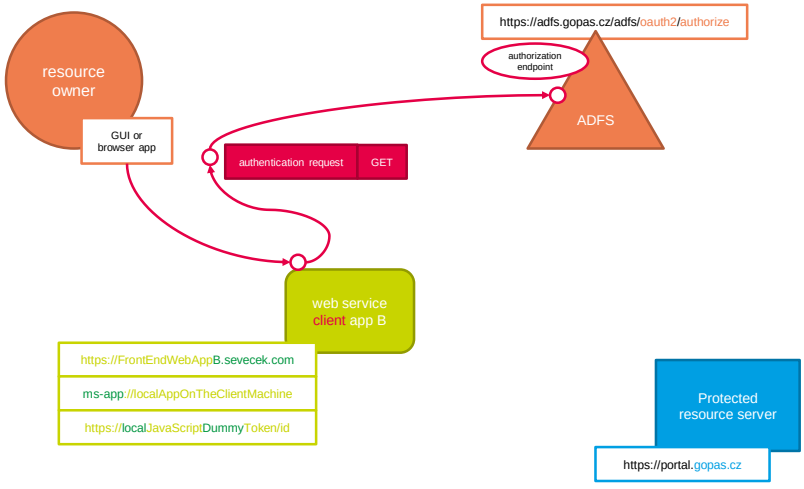


Complex redirection with confidential client



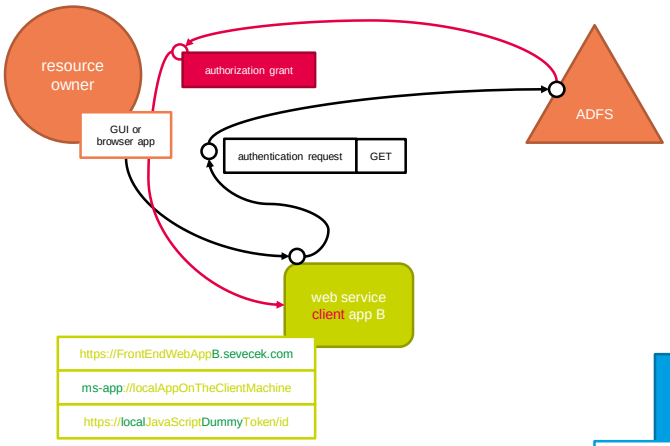
GOPAS

Complex redirection with confidential or public client



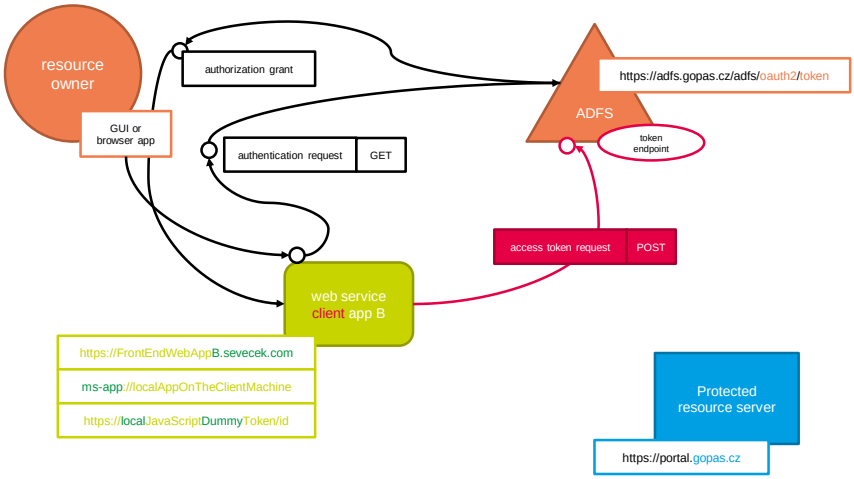
GOPAS

Complex redirection with confidential or public client



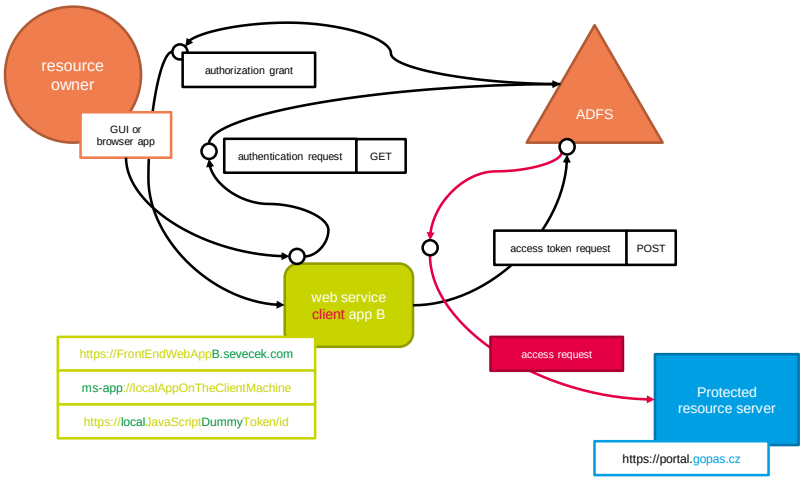
GOPAS

Complex redirection with confidential or public client



GOPAS

Complex redirection with confidential or public client



GOPAS

Client types

- Confidential
  - server application which can protect its own credentials
  - usually using the authorization grant
- Public
  - mobile application on the resource owner device
  - usually using implicit grant (just like WS-Fed or SAML-P)
  - resource owner has access to the client credentials
  - native application - GUI, sand-boxed or not
  - user-agent based application - JavaScript in browser

GOPAS

The authorization requests

GET
https://adfs.gopas.cz/adfs/oauth2/authorize?response_type=code&client_id=12345678-2222-3333-4444-123456789012&redirect_uri=ms-app://localAppOnTheClientMachine&resource=https://backEndSharedWebService.gopas.cz

GET
https://adfs.gopas.cz/adfs/oauth2/authorize?response_type=code&client_id=87654321-2222-3333-4444-123456789012&redirect_uri=https://FrontEndWebAppB.sevecek.com&resource=https://backEndSharedWebService.gopas.cz

GET
https://adfs.gopas.cz/adfs/oauth2/authorize?response_type=code&client_id=01010101-2222-3333-4444-123456789012&redirect_uri=https://localJavaScriptDummyToken/id&resource=https://backEndSharedWebService.gopas.cz

GOPAS

The authorization grant requests

POST https://adfs.gopas.cz/adfs/oauth2/token
grant_type=authorization_code&client_id=12345678-2222-3333-4444-123456789012&redirect_uri=https://portal.gopas.cz&code=[code]

POST https://adfs.gopas.cz/adfs/oauth2/token
grant_type=authorization_code&client_id=87654321-2222-3333-4444-123456789012&redirect_uri=https://backEndSharedWebService&code=[code]

POST https://adfs.gopas.cz/adfs/oauth2/token
grant_type=authorization_code&client_id=01010101-2222-3333-4444-123456789012&redirect_uri=https://portal.gopas.cz&code=[code]

GOPAS