

SharePoint Server mit ADFS – SAML-Authentifizierung einrichten



WISSEN

Grundlagen, Architektur und alle Praxisbeiträge rund um ADFS an einem Ort.

› [Active Directory Federation Services](#)

BERATUNG

SharePoint-Anbindung prüfen, Identitätsclaim richtig wählen, Migration ohne Rechtsverlust planen.

› [Consulting zu ADFS](#)

SCHULUNG

Im Labor: Token Issuer bauen, People Picker zähmen, Identitätsclaim absichtlich falsch wählen.

› [ADFS-Schulungen](#)

Es gibt Konfigurationen, die verzeihen dir alles. Die SAML-Anbindung von SharePoint Server an ADFS gehört nicht dazu. Der eigentliche Aufbau ist in einer Stunde erledigt: eine Relying Party in ADFS, ein Token Issuer in SharePoint, ein Zertifikat, zwei Claim-Mappings. Die Anmeldung funktioniert, alle sind zufrieden. Und dann, zwei Jahre und eine Umbenennung der E-Mail-Domäne später, stellt jemand fest, dass man den Identitätsclaim nicht einfach umstellen kann – und dass SharePoint ab diesem Moment jeden Benutzer für einen Fremden hält.

Dieser Beitrag zeigt dir den Aufbau Schritt für Schritt: den Relying Party Trust in ADFS, den Trusted Identity Token Issuer in SharePoint, das Claim-Mapping mit E-Mail oder UPN als Identitätsclaim, das Zonen-Design und den Zertifikatswechsel. Danach geht es um die beiden Stellen, an denen in der Praxis die meisten Projekte nachträglich wehtun: den People Picker, der ungeprüft jede Eingabe akzeptiert, und die Migrationsfälle beim Wechsel des Identitätsclaims. Die ADFS-Grundlagen setzen wir voraus – sie stehen auf der Übersichtsseite [Active Directory Federation Services](#). Alles rund um Farmen, Dienste und Betrieb von SharePoint selbst findest du auf der Seite [SharePoint](#).

FAKTEN – SharePoint und ADFS in fünf Sätzen

SharePoint Server 2013, 2016, 2019 und Subscription Edition unterstützen die Anmeldung über einen vertrauenswürdigen Identitätsanbieter wie ADFS. In SharePoint heißt das Objekt dafür SPTrustedIdentityTokenIssuer.

Technisch läuft die Anmeldung über passives WS-Federation mit SAML-1.1-Token. SAML 2.0 im Sinne des SAML-P-Protokolls spricht der Token Issuer nicht – wenn jemand „SharePoint mit SAML 2.0“ bestellt, lohnt eine Rückfrage.

Die Subscription Edition kann zusätzlich OpenID Connect – auch gegen ADFS. Für bestehende Umgebungen bleibt der SAML-Weg der Normalfall.

Der Identitätsclaim wird beim Anlegen des Token Issuers festgelegt. Set-SPTrustedIdentityTokenIssuer hat dafür keinen Parameter.

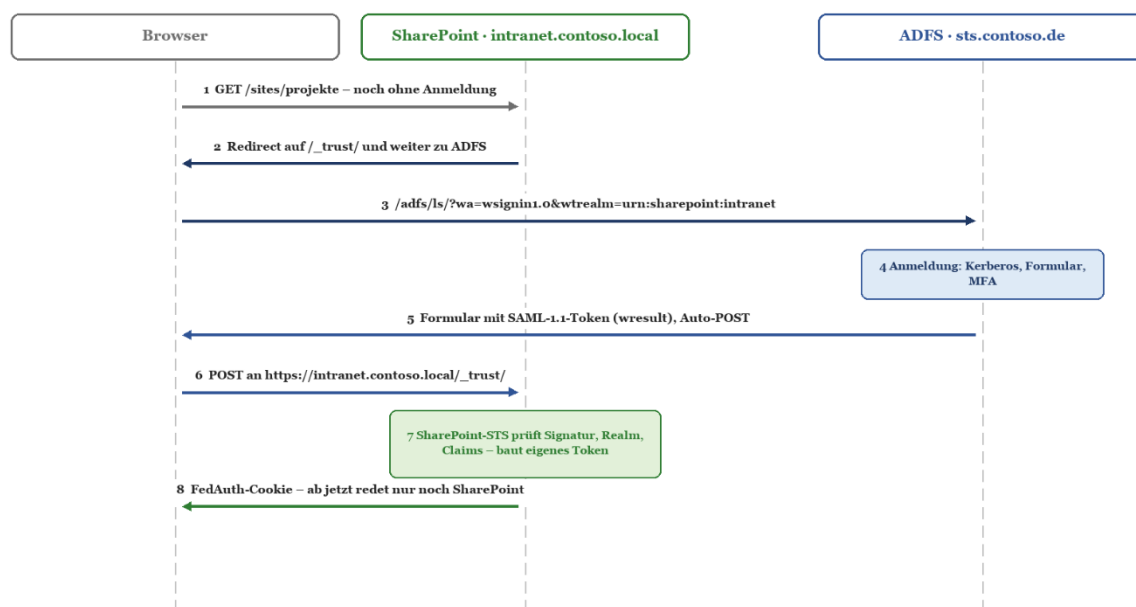
Ohne zusätzlichen Claims Provider löst der People Picker jede Eingabe auf, auch wenn es den Benutzer gar nicht gibt.

Was SharePoint unter „SAML“ versteht – und wie die Zonen aussehen

Bevor du die erste Zeile PowerShell tippst, lohnt ein Blick auf den Ablauf. SharePoint ist bei dieser Anbindung eine Relying Party wie jede andere, nur eine mit eigenem Security Token Service. Der Browser landet ohne Anmeldung auf einer Seite, SharePoint schickt ihn über den Endpunkt `/_trust/` zu ADFS, ADFS meldet den Benutzer an und liefert ein SAML-1.1-Token per Formular-POST zurück an `/_trust/`. Der SharePoint-STS prüft Signatur, Realm und Claims, baut daraus ein eigenes Token und setzt das FedAuth-Cookie. Ab diesem Moment ist ADFS raus, bis das Cookie abläuft.

Skizze 1: So meldet sich ein Benutzer über ADFS an SharePoint an

WS-Federation als Transport, SAML 1.1 als Token – und am Ende ein Cookie von SharePoint selbst



ADFS sieht den Benutzer genau einmal pro Sitzung. Danach lebt alles vom FedAuth-Cookie und vom SharePoint-eigenen Token. Wer im ADFS-Log nach jedem Seitenaufruf sucht, sucht vergeblich.

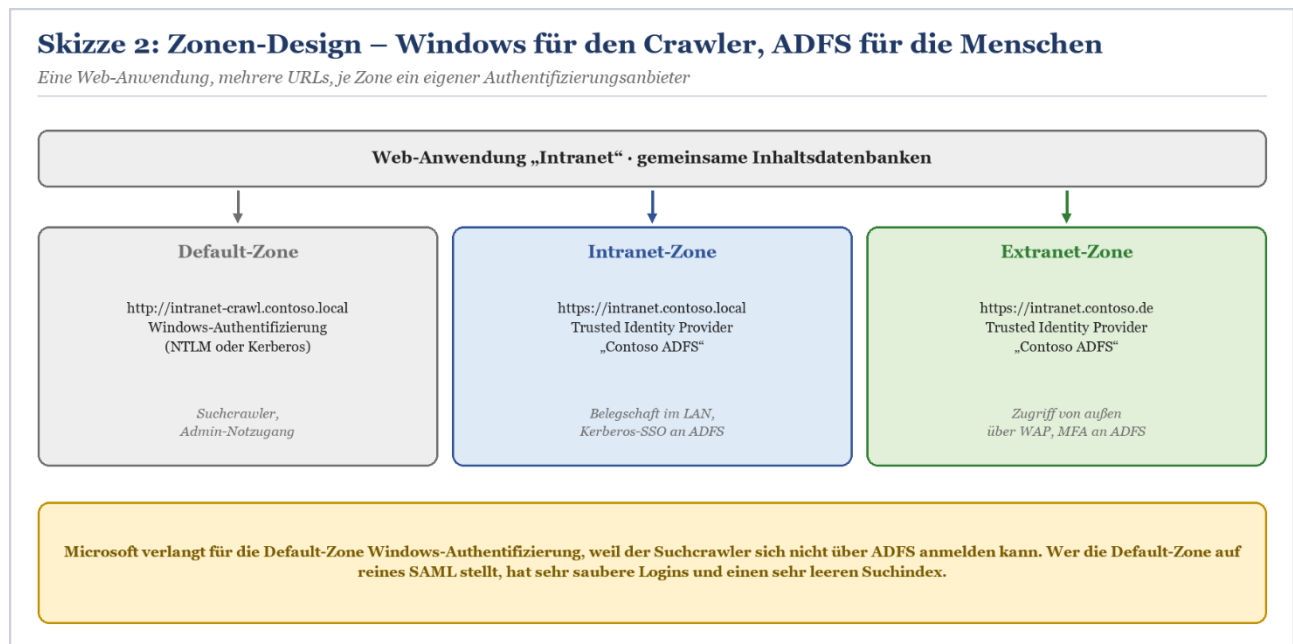
Skizze 1: Der Anmeldeablauf – ADFS sieht den Benutzer einmal, danach übernimmt SharePoint.

Das hat zwei praktische Folgen. Erstens: Was SharePoint mit dem Benutzer anfängt, entscheidet allein der Inhalt des Tokens. SharePoint fragt nicht im Active Directory nach, ob es Anna Berger wirklich gibt, ob sie in einer Gruppe ist oder wie sie heute heißt. Es glaubt dem Token. Zweitens: Alles, was danach kommt – Berechtigungen, Gruppenmitgliedschaften, „Geändert von“ in jeder Dokumentbibliothek – hängt an einer Zeichenkette, die SharePoint aus dem Identitätsclaim baut. Merke dir diesen Satz, er kommt im letzten Kapitel mit voller Wucht zurück.

Das Zonen-Design: Default-Zone bleibt Windows

Eine SharePoint-Webanwendung kann bis zu fünf Zonen haben, jede mit eigener URL und eigenen Authentifizierungsanbietern. Microsoft schreibt für die ADFS-Anbindung eine Regel ausdrücklich

vor: In der Default-Zone muss Windows-Authentifizierung aktiviert sein, weil der Suchcrawler sich nicht über ADFS anmelden kann. Außerdem muss die URL, die über ADFS angemeldet wird, per HTTPS laufen. Daraus ergeben sich zwei gängige Muster: Entweder die Default-Zone bekommt Windows und ADFS gleichzeitig, dann sehen Benutzer beim Anmelden eine Auswahlseite. Oder du erweiterst die Webanwendung um eine zusätzliche Zone, die nur ADFS kennt.



Skizze 2: Default-Zone mit Windows für Crawler und Notzugang, weitere Zonen mit ADFS für die Belegschaft.

Muster	Vorteil	Nachteil	Passt, wenn ...
Default-Zone mit Windows und ADFS	eine URL für alle, kein zusätzlicher IIS-Site	Auswahlseite beim Anmelden, Benutzer klicken gern das Falsche	die Umstellung schrittweise läuft und beide Welten eine Weile parallel leben
Erweiterte Zone nur mit ADFS	Benutzer sehen nur ADFS, saubere Trennung	zusätzliche URL, Alternate Access Mappings, zweiter IIS-Site	ADFS der einzige Weg für Menschen sein soll – der häufigste Zielzustand
Default-Zone nur mit ADFS	auf dem Papier am einfachsten	Suche crawlt nicht, Microsoft rät ausdrücklich ab	nie

Tabelle 1: Zonen-Muster für die ADFS-Anbindung

TIPP – Der Windows-Zugang ist auch dein Rettungsboot

Lass dir in der Default-Zone mindestens ein Windows-Konto als Websitesammlungsadministrator stehen. Wenn ADFS einmal hustet, das Signaturzertifikat abläuft oder jemand die Relying Party versehentlich deaktiviert, kommst du über diesen Weg noch in die Websites – und musst die Fehlersuche nicht über das Telefon des Vorstands erledigen.

Schritt 1: Die Relying Party in ADFS anlegen

Auf der ADFS-Seite ist SharePoint eine klassische WS-Federation-Relying-Party. Metadaten veröffentlicht SharePoint für diesen Zweck nicht, also legst du den Trust von Hand an. Wie das grundsätzlich funktioniert und worauf du bei manuell angelegten Trusts achten musst, beschreibt der Beitrag [Relying Party Trust anlegen – per Metadaten und von Hand](#). Für SharePoint brauchst du drei Dinge: einen Bezeichner, der später als Realm in SharePoint auftaucht, den WS-Federation-Endpunkt und eine Zugriffssteuerungsrichtlinie.

Bezeichner und Endpunkt

Der Bezeichner ist eine frei wählbare URI. Bewährt hat sich ein URN, der die Webanwendung beschreibt und nicht die URL – also etwa urn:sharepoint:intranet. URLs ändern sich, URNs nicht, und du wirst dankbar sein, wenn der Bezeichner eine Umbenennung des Hostnamens überlebt. Der Endpunkt ist die URL der Webanwendung mit /_trust/ am Ende, in Microsofts Anleitung mit /_trust/default.aspx. Beides funktioniert, wichtig ist nur, dass die URL zur Zone gehört, die über ADFS angemeldet werden soll.

```
# Auf einem ADFS-Server der Farm, als Administrator
$name           = "SharePoint Intranet"
$identifier      = "urn:sharepoint:intranet"
$endpoint       = "https://intranet.contoso.local/_trust/"

Add-AdfsRelyingPartyTrust -Name $name `
    -Identifier $identifier `
    -WSFedEndpoint $endpoint `
    -AccessControlPolicyName "Permit everyone" `
    -Enabled $true

Get-AdfsRelyingPartyTrust -Name $name |
    Format-List Name, Identifier, WSFedEndpoint, AccessControlPolicyName
```

Listing 1: Relying Party für SharePoint anlegen – ab ADFS unter Windows Server 2016 mit Zugriffssteuerungsrichtlinie

Microsofts eigene Anleitung arbeitet noch mit einer Autorisierungsregel, die allen den Zugriff erlaubt. Das stammt aus der Zeit von ADFS 2.0 und funktioniert weiterhin. Auf aktuellen Farmen ist die Zugriffssteuerungsrichtlinie der lesbarere Weg – und wenn nur bestimmte Gruppen auf SharePoint dürfen oder von außen MFA verlangt wird, regelst du das dort und nicht in SharePoint. Wie das geht, steht im Beitrag [Access Control Policies in ADFS – Zugriffsregeln ohne Claim-Rule-Akrobatik](#).

Claim Rules: Was SharePoint im Token braucht

SharePoint braucht mindestens einen Claim: den Identitätsclaim. Fast jede Umgebung will zusätzlich Gruppen als Rollen-Claims, damit Berechtigungen nicht für jeden Benutzer einzeln vergeben werden müssen. Eine einzige LDAP-Regel erledigt beides. Wir stellen hier E-Mail, UPN und die Gruppen aus. Welcher davon der Identitätsclaim wird, entscheidest du erst in SharePoint – ausgestellt werden dürfen ruhig beide.

```
# Claim-Typen aus .NET holen statt lange URIs abzutippen
$mailType = [System.Security.Claims.ClaimTypes]::Email
$upnType  = [System.Security.Claims.ClaimTypes]::Upn
$roleType = [System.Security.Claims.ClaimTypes]::Role
$winType  = [System.Security.Claims.ClaimTypes]::WindowsAccountName

$rules = @"
@RuleTemplate = "LdapClaims"
@RuleName = "AD-Attribute für SharePoint"
c:[Type == "$winType", Issuer == "AD AUTHORITY"]
=> issue(store = "Active Directory",
    types = ("$mailType", "$upnType", "$roleType"),
    query = ";mail,userPrincipalName,tokenGroups(fullDomainQualifiedNames);{0}",
    param = c.Value);
"@

$rules # einmal ansehen: Die Variablen sind jetzt durch die vollständigen Claim-Typ-URIs
ersetzt
Set-AdfsRelyingPartyTrust -TargetName "SharePoint Intranet" -IssuanceTransformRules $rules
```

Listing 2: E-Mail, UPN und Gruppen (als CONTOSO\Gruppenname) an SharePoint ausstellen

Die Gruppen kommen als Rollen-Claims im Format Domäne\Gruppenname. Das ist lesbar und für den People Picker praktisch, hat aber einen Haken: Benennst du eine Gruppe um, ändert sich der

Claim, und die Berechtigung in SharePoint zeigt ins Leere. Wer das vermeiden will, stellt statt des Namens die Gruppen-SID aus – lesbarer wird es dadurch nicht, stabiler schon. Die Regelsprache dahinter erklärt der [ADFS Claim Rules Leitfadens](#).

WARNUNG – Benutzer ohne E-Mail-Adresse

Wenn du E-Mail als Identitätsclaim nimmst, muss jedes Konto, das sich an SharePoint anmelden soll, das Attribut mail gefüllt haben. Dienstkonten, frisch angelegte Benutzer und Konten aus Altdomänen haben es oft nicht. ADFS stellt dann schlicht keinen E-Mail-Claim aus, die Anmeldung an ADFS klappt, und SharePoint lehnt das Token ab, weil der Identitätsclaim fehlt. Für den Benutzer sieht das aus wie „SharePoint kaputt“, für ADFS wie ein erfolgreicher Login. Prüfe die Attribute vorher – und nicht erst, wenn das Helpdesk-Telefon klingelt.

E-Mail oder UPN als Identitätsclaim?

Das ist die wichtigste Entscheidung dieses Beitrags, und sie wird erstaunlich oft nebenbei getroffen. Entscheidend ist nicht, welcher Wert heute schöner aussieht, sondern welcher Wert sich in den nächsten zehn Jahren am seltensten ändert, immer gefüllt ist und eindeutig bleibt. Keiner der Kandidaten ist perfekt.

Kriterium	E-Mail (emailaddress)	UPN (upn)
Für Benutzer lesbar	ja, alle kennen ihre Adresse	meist ja, wenn UPN = E-Mail; sonst verwirrend
Immer gefüllt	nein – Konten ohne Postfach haben oft keine	ja, jedes Benutzerkonto hat einen UPN
Ändert sich bei Namenswechsel	ja	ja, wenn der UPN aus dem Namen gebildet wird
Ändert sich bei neuer Mail-Domäne	ja – Umfirmierung, Fusion, Marken-Relaunch	nur, wenn der UPN-Suffix mitzieht
Eindeutigkeit	durch Exchange in der Regel gewährleistet	im Forest garantiert
Spätere Migration nach Entra ID	gut, sofern mail und UPN gleich sind	gut, Entra ID denkt in UPNs
Microsofts Anleitung nutzt	E-Mail	–

Tabelle 2: E-Mail oder UPN – keine Wahl ist perfekt, aber eine ist meistens besser

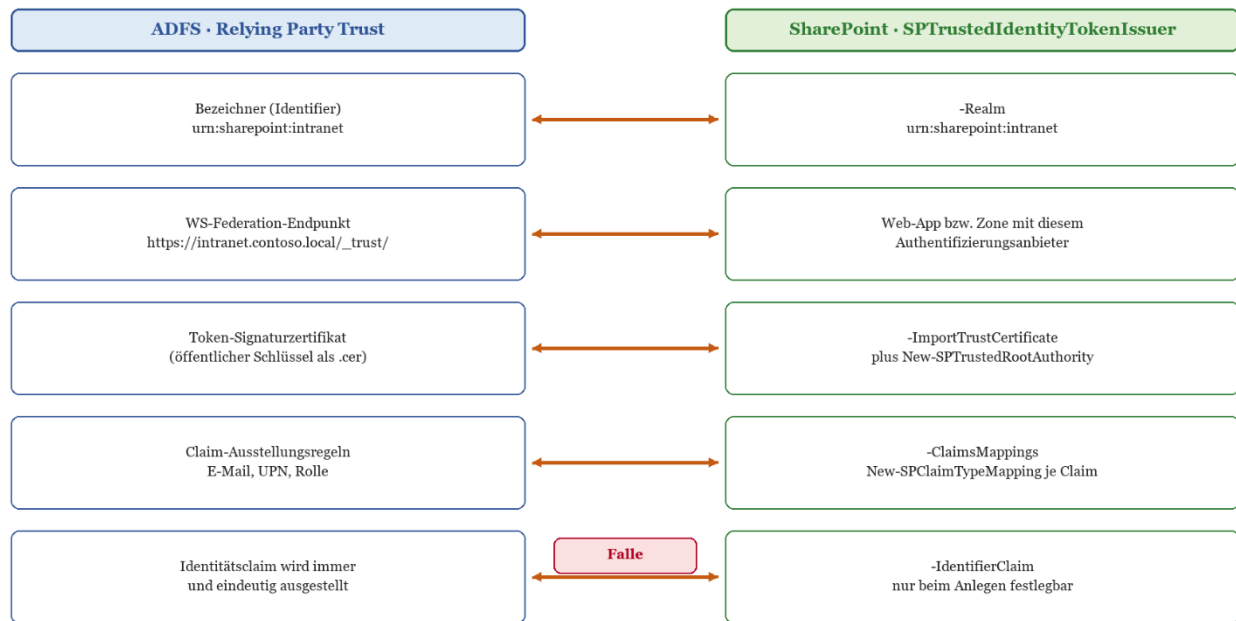
Unsere Empfehlung aus der Praxis: Wenn UPN und primäre E-Mail-Adresse in deiner Umgebung identisch sind und das auch bleiben sollen, nimm den UPN. Er ist immer gefüllt, im Forest eindeutig und passt zu Entra ID, falls SharePoint später einmal einen anderen Identitätsanbieter bekommt. Sind UPN und E-Mail unterschiedlich – etwa contoso.local als UPN-Suffix –, nimm die E-Mail, weil niemand im People Picker „aberger@contoso.local“ eintippen will. Und in beiden Fällen gilt: Schreib die Entscheidung mit Begründung ins Betriebshandbuch. Der Mensch, der in fünf Jahren fragt, warum das so ist, könntest du sein.

Schritt 2: Den Trusted Identity Token Issuer in SharePoint anlegen

Auf der SharePoint-Seite brauchst du das öffentliche Token-Signaturzertifikat von ADFS, eine Zuordnung der Claim-Typen und den Token Issuer selbst. Alle Befehle laufen in der SharePoint-Verwaltungsshell auf einem Server der Farm. Welche Werte auf beiden Seiten exakt zueinander passen müssen, zeigt Skizze 3.

Skizze 3: Was in ADFS und SharePoint exakt zusammenpassen muss

Fünf Paare – ein einziger Tippfehler reicht für eine Fehlerseite



Der unterste Pfeil ist der teuerste: Den Identitätsclaim kannst du später nicht per Set-Cmdlet ändern. Wer ihn wechseln will, baut den Token Issuer neu und migriert jede Berechtigung.

Skizze 3: Fünf Wertepaare zwischen ADFS und SharePoint – der Identitätsclaim ist das teuerste.

Zertifikate exportieren und vertrauen

SharePoint prüft die Signatur jedes Tokens. Dafür bekommt der Token Issuer den öffentlichen Schlüssel des Token-Signaturzertifikats. Zusätzlich muss SharePoint der Zertifikatskette vertrauen – und zwar über seinen eigenen Speicher für vertrauenswürdige Stammzertifizierungsstellen, nicht über den Windows-Zertifikatspeicher. Ist das Signaturzertifikat selbstsigniert, wie bei ADFS mit AutoCertificateRollover üblich, trägst du das Zertifikat selbst dort ein. Stammt es von einer Zertifizierungsstelle, trägst du Stamm- und Zwischenzertifikate ein.

```
# Auf dem ADFS-Server: primäres Token-Signaturzertifikat als .cer exportieren
$ts = Get-AdfsCertificate -CertificateType Token-Signing | Where-Object IsPrimary
[System.IO.File]::WriteAllBytes("C:\Temp\adfs-signing.cer", $ts.Certificate.Export("Cer"))

# In der SharePoint-Verwaltungsshell: Zertifikat laden und als vertrauenswürdig eintragen
$signingCert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate2("C:\Temp\adfs-signing.cer")
New-SPTrustedRootAuthority -Name "ADFS Token-Signing sts.contoso.de" -Certificate $signingCert
```

Listing 3: Token-Signaturzertifikat exportieren und in SharePoint vertrauen (selbstsigniertes Zertifikat)

Claim-Mappings und Token Issuer

Jeder Claim, den SharePoint aus dem Token übernehmen soll, braucht ein Mapping mit New-SPClaimTypeMapping. Claims ohne Mapping ignoriert SharePoint. Der Identitätsclaim ist dabei kein eigener Typ, sondern eines dieser Mappings, das du beim Anlegen per -IdentifierClaim zum Identitätsclaim erklärst. Microsoft weist ausdrücklich darauf hin, den Schalter -UseDefaultConfiguration nicht zu verwenden, weil er die Identität der Benutzer intern auf eine Weise setzt, die später unerwartete Nebenwirkungen hat.

```
$upn = New-SPClaimTypeMapping -IncomingClaimType ([System.Security.Claims.ClaimTypes]::Upn) `
-IncomingClaimTypeDisplayName "UPN" -SameAsIncoming
```

```
$email = New-SPClaimTypeMapping -IncomingClaimType ([System.Security.Claims.ClaimTypes]::Email)
`
    -IncomingClaimTypeDisplayName "E-Mail" -SameAsIncoming
$role = New-SPClaimTypeMapping -IncomingClaimType ([System.Security.Claims.ClaimTypes]::Role)
`
    -IncomingClaimTypeDisplayName "Rolle" -SameAsIncoming

New-SPTrustedIdentityTokenIssuer -Name "Contoso ADFS" `
    -Description "ADFS-Farm sts.contoso.de" `
    -Realm "urn:sharepoint:intranet" `
    -ImportTrustCertificate $signingCert `
    -ClaimsMappings $upn, $email, $role `
    -SignInUrl "https://sts.contoso.de/adfs/ls/" `
    -IdentifierClaim $upn.InputClaimType
```

Listing 4: Token Issuer mit UPN als Identitätsclaim – diese eine Zeile ganz unten ist die Entscheidung für Jahre

WICHTIG – Der Name des Token Issuers steht in jeder Berechtigung

Der Wert von -Name ist nicht nur eine Beschriftung. SharePoint baut ihn in jede Benutzer- und Gruppenidentität ein, die über diesen Anbieter vergeben wird – etwa i:05.t|Contoso ADFS|anna.berger@contoso.de bei E-Mail als Identitätsclaim. Wähle einen neutralen, dauerhaften Namen. „ADFS-Test-Klaus“ ist auch nach zehn Jahren noch in jeder Berechtigungsliste zu sehen.

Webanwendung oder Zone zuweisen

Der Token Issuer existiert jetzt farmweit, benutzt ihn aber noch niemand. Du weist ihn einer Zone zu – entweder beim Erweitern der Webanwendung um eine neue Zone oder an einer bestehenden Zone. Das folgende Beispiel erweitert die Webanwendung um die Intranet-Zone mit ADFS als einzigem Anbieter. Danach bindest du in IIS das HTTPS-Zertifikat an den neuen Site und prüfst in der Zentraladministration die Alternate Access Mappings.

```
$sts = Get-SPTrustedIdentityTokenIssuer "Contoso ADFS"
$ap = New-SPAuthenticationProvider -TrustedIdentityTokenIssuer $sts
$wa = Get-SPWebApplication "http://intranet-crawl.contoso.local"

New-SPWebApplicationExtension -Identity $wa -Name "SharePoint Intranet ADFS" `
    -Zone Intranet -Url "https://intranet.contoso.local" -SecureSocketsLayer `
    -AuthenticationProvider $ap
```

Listing 5: Bestehende Webanwendung um eine ADFS-Zone erweitern

Danach legst du für den ersten ADFS-Benutzer eine Berechtigung an, etwa als zusätzlichen Websitesammlungsadministrator. Im People Picker tippst du dafür den exakten Wert des Identitätsclaims ein und wählst rechts den Anbieter „Contoso ADFS“. Das Wort „exakt“ ist dabei kein Stilmittel – dazu gleich mehr.

Mehrere Webanwendungen, ein Token Issuer

Ein Token Issuer kann mehrere Webanwendungen bedienen. Jede bekommt in ADFS einen eigenen Bezeichner, meist auch eine eigene Relying Party, und in SharePoint einen Eintrag in den ProviderRealms des Token Issuers. SharePoint schickt dann je nach URL den passenden Realm an ADFS.

```
$sts = Get-SPTrustedIdentityTokenIssuer "Contoso ADFS"
$sts.ProviderRealms.Add([System.Uri]"https://teams.contoso.local/", "urn:sharepoint:teams")
$sts.Update()
```

Listing 6: Zweite Webanwendung mit eigenem Realm an denselben Token Issuer hängen

Die Alternative ist eine einzige Relying Party mit dem Schalter -UseWReply am Token Issuer: SharePoint teilt ADFS dann bei jeder Anmeldung mit, wohin das Token zurück soll. Das ist bequemer, aber ADFS muss jede dieser Adressen als Endpunkt der Relying Party kennen. Für

Umgebungen mit vielen Webanwendungen und Hostnamen ist das praktisch, für die Übersicht im Inventar sind getrennte Relying Parties ehrlicher.

Token-Lebensdauer und Zertifikatswechsel

Zwei Betriebsthemen solltest du vom ersten Tag an im Blick haben. Das erste ist die Lebensdauer: SharePoint erneuert die Anmeldung, wenn das ADFS-Token in Kürze abläuft. Das Fenster dafür steuert LogonTokenCacheExpirationWindow in Set-SPSecurityTokenServiceConfig, standardmäßig zehn Minuten. Ist die Token-Lebensdauer der Relying Party in ADFS kürzer als dieses Fenster, hält SharePoint jedes frische Token für bereits abgelaufen – und schickt den Browser in eine Weiterleitungsschleife zwischen /_trust/ und ADFS. Die Lebensdauer setzt du in ADFS mit Set-AdfsRelyingPartyTrust -TokenLifetime in Minuten; sie muss deutlich über dem Fenster liegen.

Das zweite ist das Token-Signaturzertifikat. SharePoint liest keine ADFS-Metadaten nach und bemerkt einen Zertifikatswechsel nicht von selbst. Rollt ADFS automatisch auf ein neues Zertifikat, signiert es Tokens ab dem Umschaltzeitpunkt mit einem Schlüssel, den SharePoint nicht kennt, und jede Anmeldung scheitert. Deshalb gehört das neue Zertifikat in SharePoint, bevor ADFS es zum primären macht – in der Zeit, in der es in ADFS bereits als sekundäres Zertifikat bereitliegt. Wie du den Wechsel in ADFS planst, steht im Beitrag [ADFS Token-Signing-Zertifikat erneuern – mit und ohne AutoCertificateRollover](#).

```
# Neues Zertifikat (vorher als .cer vom ADFS-Server exportiert) eintragen
$newCert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate2("C:\Temp\adfs-signing-neu.cer")
New-SPTrustedRootAuthority -Name "ADFS Token-Signing sts.contoso.de neu" -Certificate $newCert
Set-SPTrustedIdentityTokenIssuer -Identity "Contoso ADFS" -ImportTrustCertificate $newCert
```

Listing 7: Token Issuer auf das neue Signaturzertifikat umstellen

WARNUNG – Das Umschaltfenster ist kurz

Der Token Issuer kennt genau ein Signaturzertifikat. Stellst du SharePoint zu früh um, scheitern die Anmeldungen bis zum Rollover in ADFS; stellst du zu spät um, scheitern sie danach. In der Praxis bewährt hat sich: Rollover in ADFS manuell zu einem festen Termin auslösen und SharePoint im selben Wartungsfenster umstellen. Bestehende Sitzungen mit gültigem FedAuth-Cookie merken davon übrigens nichts – das macht den Test trügerisch.

Symptom	Wahrscheinliche Ursache	Wo nachsehen
ADFS meldet, die vertrauende Seite sei unbekannt	Realm in SharePoint und Bezeichner in ADFS weichen ab	Get-SPTrustedIdentityTokenIssuer, Get-AdfsRelyingPartyTrust
Anmeldung an ADFS klappt, SharePoint zeigt Fehlerseite	Identitätsclaim fehlt im Token oder Zertifikat unbekannt	ULS-Protokoll, Kategorie Claims Authentication
„The root of the certificate chain is not a trusted root authority“	Kette fehlt in New-SPTrustedRootAuthority	Get-SPTrustedRootAuthority
Endlose Weiterleitung zwischen SharePoint und ADFS	Token-Lebensdauer kürzer als LogonTokenCacheExpirationWindow	Get-SPSecurityTokenServiceConfig, Get-AdfsRelyingPartyTrust
Anmeldung klappt, aber „Zugriff verweigert“	Berechtigung mit falschem Wert vergeben (People Picker)	Websiteberechtigungen, Benutzerinformationsliste
Seit gestern scheitern alle neuen Anmeldungen	Token-Signaturzertifikat in ADFS gewechselt	Get-AdfsCertificate

Tabelle 3: Die häufigsten Fehlerbilder nach der Einrichtung

Für die ADFS-Seite der Fehlersuche gibt es zwei passende Beiträge: [ADFS-Anmeldung schlägt fehl – die häufigsten Ursachen und ihre Lösung](#) für die typischen Ursachen und [ADFS-Ereignisprotokolle lesen – Admin-Log, Debug-Tracing und Auditing](#) für das Lesen von Admin-Log und Debug-Tracing. Auf SharePoint-Seite ist das ULS-Protokoll dein Freund – auch wenn es sich selten so anfühlt.

Der People Picker und das Problem der ungeprüften Eingaben

Jetzt kommt der Teil, über den Benutzer sich beschweren und den Administratoren gern übersehen. Bei Windows-Authentifizierung fragt der People Picker das Active Directory: Gibt es diesen Benutzer, wie heißt er, welche Gruppen passen zur Eingabe? Beim vertrauenswürdigen Identitätsanbieter kann SharePoint das nicht. Der Standard-Anbieter weiß nichts über den Verzeichnisdienst hinter ADFS. Microsoft beschreibt das Verhalten offen: Der People Picker kann Benutzer und Gruppen nicht suchen, auflösen oder prüfen; stattdessen löst er jeden eingegebenen Wert auf, auch wenn es kein gültiger Claim ist.

Skizze 4: Der People Picker – ohne und mit Claims Provider

Links glaubt SharePoint alles. Rechts fragt jemand nach.

Standard: SPTrustedClaimProvider

anna.bergr@contoso.de

Aufgelöst: anna.bergr@contoso.de

Jeder Text wird ungeprüft zum Claim – Tippfehler inklusive.

Folgen:

- Berechtigung für eine Person, die es nicht gibt
- „Zugriff verweigert“ für die Person, die es gibt
- Gruppen nur per exaktem Rollenwert
- Helpdesk-Tickets mit Schuldzuweisung an ADFS

Mit Claims Provider (LDAP oder UPA)

anna.bergr@contoso.de

Kein Treffer – meintest du Anna Berger?

Suche im AD per LDAP oder im User Profile Service.

Optionen:

- UPA-gestützter Anbieter (Subscription Edition)
- Community-Lösung wie LDAPCP
- eigener Claims Provider aus der Entwicklung
- Wert muss zum Identitätsclaim passen

Der Claims Provider ändert nichts an der Anmeldung. Er sorgt nur dafür, dass bei der Vergabe von Berechtigungen exakt der Wert eingetragen wird, den ADFS später im Token liefert.

Skizze 4: Ohne Claims Provider wird jeder Tippfehler zur Berechtigung – mit Claims Provider fragt jemand nach.

In der Praxis heißt das: Wer „anna.bergr@contoso.de“ eintippt, bekommt einen grünen Haken und eine Berechtigung für eine Person, die es nicht gibt. Anna Berger selbst sieht „Zugriff verweigert“, ruft beim Helpdesk an, und das Helpdesk sucht das Problem bei ADFS, weil ADFS ja „neu“ ist. Bei Gruppen ist es noch tückischer: Der Wert muss exakt dem Rollen-Claim entsprechen, also CONTOSO\Projekt-Leser und nicht Projekt-Leser. Auch hier gibt es keinen Widerspruch, nur später keinen Zugriff.

FAKTEN — Warum das kein Bug ist

Der Standard-Anbieter für vertrauenswürdige Identitätsanbieter hat schlicht keine Datenquelle. SharePoint kennt von ADFS nur, was im Token steht, und ein Token gibt es erst bei der Anmeldung. Bei der Vergabe von Berechtigungen ist der betreffende Benutzer aber gar nicht angemeldet – oft war er noch nie da. Deshalb braucht es einen Claims Provider, der eine echte Quelle befragt.

Lösung 1: UPA-gestützter Anbieter in der Subscription Edition

Mit SharePoint Server Subscription Edition hat Microsoft einen eingebauten Ausweg geschaffen: Der People Picker kann Benutzer und Gruppen über den Benutzerprofildienst auflösen. Voraussetzung ist, dass der Profildienst die Benutzer synchronisiert – per AD-Import mit dem Authentifizierungsanbieter für vertrauenswürdige Anbieter oder per Microsoft Identity Manager. Das Profilattribut SPS-ClaimID (in der Oberfläche „Claim User Identifier“) muss dabei aus demselben Attribut kommen, das ADFS als Identitätsclaim liefert. Gruppen löst der Anbieter nur über ihre SID auf; dafür brauchst du ein Mapping für den groupsid-Claim und eine passende Regel in ADFS.

```
$sts = Get-SPTrustedIdentityTokenIssuer "Contoso ADFS"
$cp = New-SPClaimProvider -AssemblyName "Microsoft.SharePoint, Version=16.0.0.0,
Culture=neutral, publicKeyToken=71e9bce11e9429c" `
-DisplayName "Contoso ADFS (Profile)" -Description "UPA-gestützter People Picker" `
-Type "Microsoft.SharePoint.Administration.Claims.SPTrustedBackedByUPAClaimProvider" `
-TrustedTokenIssuer $sts
Set-SPTrustedIdentityTokenIssuer $sts -ClaimProvider $cp
```

Listing 8: UPA-gestützten Claims Provider an einen bestehenden Token Issuer hängen (Subscription Edition)

Beim Neuanlegen eines SAML-Token-Issuers geht es kürzer: New-SPTrustedIdentityTokenIssuer kennt in der Subscription Edition den Schalter -UseUPABackedClaimProvider. Der Haken an der Lösung ist die Abhängigkeit: Der People Picker findet nur, wer im Profildienst steht. Läuft die Synchronisierung nicht, findet er auch niemanden – wenigstens ehrlich.

Lösung 2: LDAP- oder eigener Claims Provider

Für SharePoint 2016 und 2019 – und überall dort, wo der Profildienst nicht die richtige Quelle ist – bleibt ein eigener Claims Provider. Er wird als Farmlösung bereitgestellt, an den Token Issuer gehängt und fragt bei jeder Eingabe im People Picker eine echte Quelle ab, typischerweise das Active Directory per LDAP. Im Feld hat sich dafür die Community-Lösung LDAPCP etabliert, für Entra ID als Quelle gibt es ein Schwesterprojekt. Beide sind keine Microsoft-Produkte: Du holst dir Code in die Farm, den du bei jedem kumulativen Update mittesten und dessen Pflege du im Blick behalten musst. Alternativ entwickelt euer Team einen eigenen Anbieter nach Microsofts Vorlage für Claims Provider – das ist Entwicklungsarbeit, keine Konfiguration.

Ansatz	Versionen	Quelle	Aufwand und Risiko
Standard ohne Claims Provider	alle	keine	kein Aufwand, dafür ungeprüfte Eingaben und Tippfehler-Berechtigungen
UPA-gestützter Anbieter	Subscription Edition	Benutzerprofildienst	eingebaut; Profilsynchronisierung muss sauber laufen, Gruppen nur per SID
LDAPCP o. ä.	2016, 2019, Subscription Edition	Active Directory per LDAP	schnell eingerichtet; Fremdcode in der Farm, Updates mittesten
Eigener Claims Provider	alle	frei wählbar	volle Kontrolle; Entwicklung, Tests und Pflege im eigenen Haus

Tabelle 4: Wege, den People Picker zum Nachfragen zu bringen

TIPP – Claims Provider vor der ersten Berechtigung

Richte den Claims Provider ein, bevor die Fachabteilungen anfangen, Berechtigungen zu vergeben. Jede Berechtigung, die vorher mit Tippfehler entstanden ist, bleibt sonst als Karteileiche in den Websites liegen – und Karteileichen haben die unangenehme Eigenschaft, bei jedem Berechtigungsaudit wieder aufzuerstehen.

Die Migrationsfalle: den Identitätsclaim wechseln

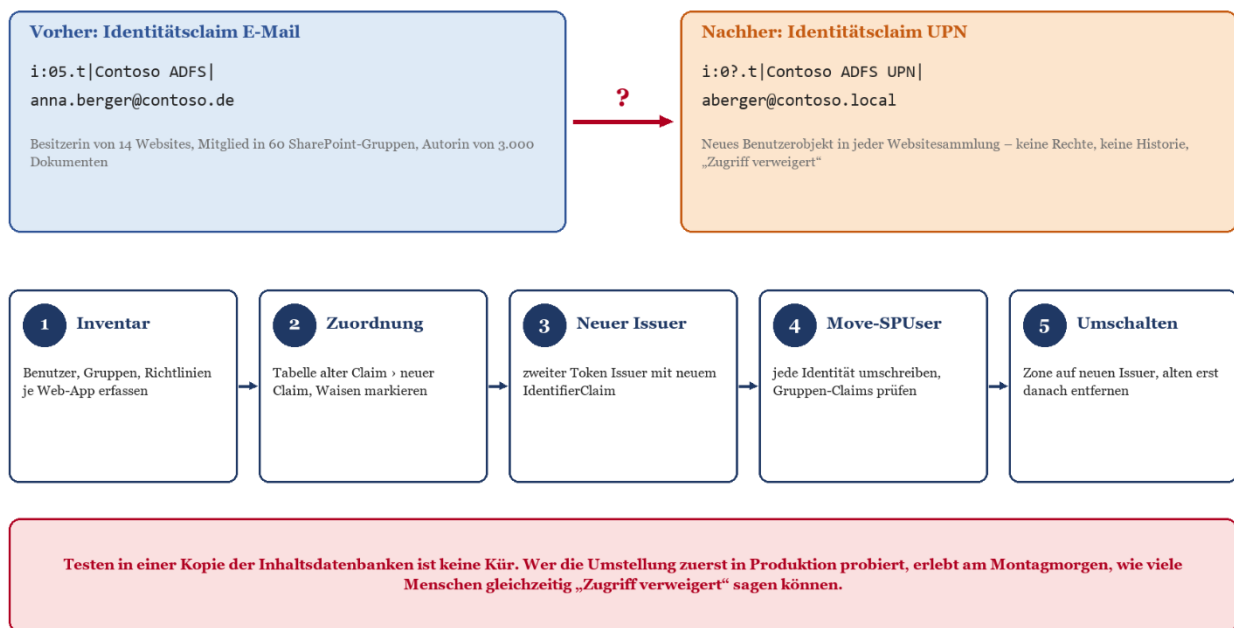
Irgendwann kommt der Tag. Die Firma heißt anders, die E-Mail-Domäne wechselt, der UPN soll endlich zur Mail-Adresse passen, oder SharePoint soll künftig über einen anderen Identitätsanbieter anmelden. Und jemand sagt: „Wir stellen den Identitätsclaim einfach auf UPN um.“ An dieser Stelle lohnt es sich, kurz innezuhalten.

Warum es kein „einfach“ gibt

SharePoint speichert Benutzer nicht als AD-Objekte, sondern als codierte Claim-Zeichenkette. Darin stecken der Typ des Identitätsclaims, die Art des Anbieters, der Name des Token Issuers und der Wert: aus `i:05.t|Contoso ADFS|anna.berger@contoso.de` wird beim Wechsel auf UPN eine ganz andere Zeichenkette. Für SharePoint ist das ein neuer Benutzer. Er hat keine Berechtigungen, keine Gruppenmitgliedschaften und keine Verbindung zu den Dokumenten, die die alte Identität erstellt hat. Und weil Set-SPTrustedIdentityTokenIssuer keinen Parameter für den Identitätsclaim hat, kommst du um einen neuen Token Issuer ohnehin nicht herum.

Skizze 5: Die Migrationsfalle beim Wechsel des Identitätsclaims

Für SharePoint ist ein neuer Claim ein neuer Mensch – mit null Berechtigungen



Skizze 5: Neuer Identitätsclaim, neue Identität – ohne Migration startet jeder Benutzer bei null.

Dasselbe gilt übrigens, wenn sich nur der Wert ändert und nicht der Typ. Heiratet Anna und heißt danach `anna.schmidt@contoso.de`, ist sie für SharePoint ebenfalls ein neuer Mensch. Das ist der kleine Bruder der Migrationsfalle, und er kommt deutlich häufiger vor. Dafür brauchst du einen geregelten Prozess mit Move-SPUser – egal welchen Identitätsclaim du gewählt hast.

So läuft eine Migration ohne Rechteverlust

- **Inventar:** Für jede Webanwendung erfassen, welche ADFS-Identitäten in Websitesammlungen, SharePoint-Gruppen und Benutzerrichtlinien der Webanwendung stehen – Benutzer und Rollen-Claims getrennt.
- **Zuordnungstabelle:** Zu jeder alten Identität den neuen Claim-Wert ermitteln, meist per Abgleich mit dem Active Directory. Konten ohne Gegenstück markieren statt raten.

- Neuer Token Issuer: Einen zweiten Issuer mit dem neuen -IdentifierClaim anlegen, auf ADFS-Seite eine passende Relying Party mit eigenem Bezeichner.
- Identitäten umschreiben: Mit Move-SPUser jede Identität auf den neuen codierten Wert umstellen. Auch Rollen-Claims enthalten den Namen des Token Issuers und müssen mit, wenn sich der Name ändert.
- Umschalten: Die Zone auf den neuen Token Issuer umstellen, testen, den alten Issuer erst entfernen, wenn wirklich nichts mehr auf ihn zeigt.

```
$neu = Get-SPTrustedIdentityTokenIssuer "Contoso ADFS UPN"
$map = Import-Csv "C:\Migration\zuordnung.csv" -Delimiter ";" # Spalten: Alt;NeuWert

foreach ($z in $map) {
    $user = Get-SPUser -Web "https://intranet.contoso.local" -Identity $z.Alt
    $newClaim = New-SPClaimsPrincipal -Identity $z.NeuWert -TrustedIdentityTokenIssuer $neu
    Move-SPUser -Identity $user -NewAlias $newClaim.ToEncodedString() -IgnoreSID -
    Confirm:$false
}
```

Listing 9: Identitäten anhand einer Zuordnungstabelle umschreiben – erst in einer Testfarm mit Datenbankkopie

WARNUNG — Move-SPUser ist kein Probelauf

Move-SPUser schreibt die Identität in den Inhaltsdatenbanken um. Es gibt keinen Rückgängig-Knopf, nur die Datensicherung. Teste das Skript mit einer Kopie der Inhaltsdatenbanken in einer Testfarm, prüfe Stichproben – Besitzer, Gruppenmitglieder, „Erstellt von“ – und plane das produktive Umschalten in ein Wartungsfenster. Microsoft schreibt für Claims-Identitäten den Schalter -IgnoreSID vor; ohne ihn versucht das Cmdlet, Windows-SIDs abzugleichen, die es hier nicht gibt.

Für den Wechsel von Windows-Authentifizierung zu ADFS gibt es übrigens ein eigenes Werkzeug: Convert-SPWebApplication kennt die Umstellung von Claims-Windows auf Claims-Trusted-Default mit Angabe des Token Issuers und einer Zuordnungsdatei. Das ist der Weg, wenn eine bestehende Windows-Webanwendung auf ADFS umzieht – und auch dort gilt: Testfarm zuerst.

Und wenn ADFS selbst einmal geht?

ADFS läuft in vielen Unternehmen seit über zehn Jahren zuverlässig, und für eine SharePoint-Farm im eigenen Rechenzentrum gibt es keinen Grund zur Hektik. Trotzdem lohnt es sich, den Ausstieg mitzudenken, denn jeder Wechsel des Identitätsanbieters ist für SharePoint ein Wechsel des Token Issuers – und damit genau die Migrationsfalle von oben. Die Subscription Edition kann sich per OpenID Connect direkt an Entra ID anmelden; wer heute einen Identitätsclaim wählt, der dort ebenfalls zur Verfügung steht, spart sich später viel Arbeit. Den Überblick über die Zielpattform findest du auf der Seite [Microsoft Entra ID](#), das Vorgehen für SAML-Anwendungen im Beitrag [SAML-Anwendungen von ADFS nach Entra ID migrieren](#). Und SharePoint gehört bei der Bestandsaufnahme selbstverständlich ins Inventar – siehe [Relying Parties inventarisieren – die Bestandsaufnahme vor der Entra-Migration](#).

FAKTEN — OpenID Connect in der Subscription Edition

Die Subscription Edition unterstützt OpenID Connect als zweites Protokoll für vertrauenswürdige Identitätsanbieter, sowohl mit ADFS als auch mit Entra ID. Das Prinzip mit Token Issuer, Claim-Mappings und Identitätsclaim bleibt gleich, ebenso die Folgen eines Identitätsclaim-Wechsels. Wie ADFS OpenID Connect grundsätzlich abbildet, beschreibt der Beitrag [OAuth 2.0 und OpenID Connect mit ADFS Application Groups](#).

Fazit: Eine Stunde Einrichtung, eine Entscheidung für Jahre

Die SAML-Anbindung von SharePoint Server an ADFS ist kein Hexenwerk: Relying Party mit Bezeichner und /_trust/-Endpunkt, eine LDAP-Claim-Regel, das Token-Signaturzertifikat als vertrauenswürdige Stammzertifizierungsstelle, ein Token Issuer mit Claim-Mappings, eine Zone mit dem neuen Anbieter. Die Default-Zone behält Windows für den Crawler, und die Token-Lebensdauer liegt über dem Erneuerungsfenster von SharePoint.

Die eigentliche Arbeit steckt in zwei Entscheidungen, die man gern unterschätzt. Der Identitätsclaim legt fest, unter welcher Zeichenkette jeder Benutzer für immer in deinen Inhaltsdatenbanken steht – wähle ihn so, dass er Umfirmierungen, Namenswechsel und einen späteren Anbieterwechsel möglichst gut übersteht. Und der People Picker braucht einen Claims Provider, bevor die erste Fachabteilung Berechtigungen vergibt. Wenn du eine bestehende Anbindung prüfen oder einen Identitätsclaim-Wechsel planen willst, unterstützen wir dich im [Consulting zu ADFS \(Active Directory Federation Services\)](#). Wer das Ganze lieber einmal im Labor zerlegt, ist in den [ADFS-Schulungen](#) richtig. Relying Parties, Claim Rules und die SharePoint-Anbindung vertieft außerdem das Buch – mehr dazu unter [ADFS in der Praxis – das Buch im Detail](#).

WEITER — Passend zum Thema

- › [Relying Party Trust anlegen – per Metadaten und von Hand](#) – die ADFS-Seite der Anbindung im Detail
- › [ADFS Claim Rules Leitfaden](#) – die Regelsprache hinter E-Mail, UPN und Rollen
- › [ADFS Token-Signing-Zertifikat erneuern – mit und ohne AutoCertificateRollover](#) – damit SharePoint den Zertifikatswechsel übersteht
- › [Access Control Policies in ADFS – Zugriffsregeln ohne Claim-Rule-Akrobatik](#) – wer überhaupt an SharePoint darf
- › [SPN, DNS und Kerberos für ADFS – warum die interne Anmeldung auf Formular zurückfällt](#) – damit die interne Anmeldung an ADFS per Kerberos klappt
- › [SharePoint](#) – alles rund um SharePoint Server

FAQ: SharePoint Server mit ADFS und SAML

Unterstützt SharePoint Server SAML 2.0 mit ADFS?

Der Token Issuer in SharePoint arbeitet mit passivem WS-Federation und SAML-1.1-Token. Das SAML-2.0-Protokoll SAML-P spricht er nicht. ADFS beherrscht beides, für SharePoint richtest du eine WS-Federation-Relying-Party ein.

Welche URL trage ich in ADFS als Endpunkt ein?

Die URL der Zone, die über ADFS angemeldet wird, mit /_trust/ am Ende – im Beispiel https://intranet.contoso.local/_trust/ – und die Adresse muss per HTTPS erreichbar sein.

Was ist der Realm im Trusted Identity Token Issuer?

Der Bezeichner der Relying Party in ADFS. Beide Werte müssen zeichengenau übereinstimmen, sonst kennt ADFS die vertrauende Seite nicht.

Soll ich E-Mail oder UPN als Identitätsclaim nehmen?

Wenn UPN und E-Mail gleich sind, ist der UPN meist die robustere Wahl, weil er immer gefüllt und im Forest eindeutig ist. Weichen sie ab, ist die E-Mail für Benutzer lesbarer. Wichtig ist, dass der Wert selten wechselt.

Kann ich den Identitätsclaim nachträglich ändern?

Nicht am bestehenden Token Issuer. Du legst einen neuen an und schreibst alle Identitäten mit Move-SPUser um, sonst verlieren die Benutzer ihre Berechtigungen.

Warum findet der People Picker jeden Benutzer, auch erfundene?

Weil der Standard-Anbieter für vertrauenswürdige Identitätsanbieter keine Datenquelle hat und jede Eingabe ungeprüft auflöst. Abhilfe schafft ein Claims Provider: UPA-gestützt in der Subscription Edition oder eine LDAP-basierte Lösung.

Muss die Default-Zone Windows-Authentifizierung behalten?

Ja, Microsoft verlangt das für den Suchcrawler. ADFS richtest du zusätzlich in der Default-Zone oder in einer erweiterten Zone ein.

Was muss ich beim Wechsel des ADFS-Signaturzertifikats in SharePoint tun?

Das neue Zertifikat mit New-SPTrustedRootAuthority eintragen und mit Set-SPTrustedIdentityTokenIssuer -ImportTrustCertificate am Token Issuer hinterlegen – abgestimmt auf den Zeitpunkt, an dem ADFS umschaltet.

Warum hängt die Anmeldung in einer Weiterleitungsschleife?

Häufigste Ursache ist eine Token-Lebensdauer in ADFS, die kürzer ist als LogonTokenCacheExpirationWindow in SharePoint. Verlängere die Lebensdauer der Relying Party oder verkürze das Fenster.

Kann SharePoint auch ohne ADFS direkt an Entra ID?

Die Subscription Edition unterstützt OpenID Connect mit Entra ID. Für SharePoint ist das ein neuer Token Issuer, also gelten dieselben Regeln für den Identitätsclaim wie bei einer Migration.