

SAML-Anwendungen von ADFS nach Entra ID migrieren

ADFS · Entra ID · SAML

SAML-Anwendungen umziehen, ohne dass jemand das Licht ausmacht

Enterprise App anlegen, Claim Rules übersetzen, Zertifikat und Metadaten beim Dienstleister tauschen – und den Umschaltmoment so legen, dass er langweilig bleibt.

ADFS
sts.contoso.de

Entra ID
Enterprise App

SaaS

boddenberg.de

WISSEN

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

› [Active Directory Federation Services](#)

BERATUNG

Vierzig Relying Parties, drei davon mit Claim Rules, die niemand mehr versteht? Wir übersetzen mit dir – Regel für Regel.

› [Consulting zu ADFS](#)

SCHULUNG

Enterprise Apps anlegen, Claims nachbauen, Token verglichen – einmal im Labor, bevor es in der Produktion ernst wird.

› [ADFS-Schulungen](#)

Jede ADFS-Farm hat sie: die eine SAML-Anwendung, die vor acht Jahren ein Kollege angebunden hat, der heute Ziegen in der Uckermark züchtet. Die Claim Rules dazu sehen aus wie ein Gedicht in einer toten Sprache, und der Dienstleister hat seitdem dreimal den Namen gewechselt. Genau diese Anwendung soll jetzt nach Entra ID. Das Gute vorweg: Es ist weniger dramatisch, als es klingt. Eine SAML-Anwendung von ADFS nach Entra ID zu migrieren, ist im Kern ein Umzug von vier Dingen – Identifier, Antwort-URL, Claims und Zertifikat. Das weniger Gute: Wer beim Claims-Teil schlampt, merkt es erst, wenn sich jemand anmeldet und in einem frischen, leeren Konto landet. Oder in dem eines Kollegen. Beides ist schlecht, das Zweite ist schlechter.

Dieser Beitrag zeigt dir den Weg für genau eine Anwendung, so wie du ihn in der Praxis gehst: Bestandsaufnahme der Relying Party, Enterprise Application in Entra ID anlegen, Claim Rules übersetzen (inklusive der Transformationen, die in Entra ID anders heißen), Zertifikat und Metadaten beim Dienstleister tauschen, Parallelbetrieb und Umschaltzeitpunkt planen und testen. Was ADFS grundsätzlich ist und warum es in vielen Unternehmen seit über zehn Jahren zuverlässig läuft, steht auf der Übersichtsseite [Active Directory Federation Services](#). Die Entra-Seite des Themas – Lizenzen, Rollen, Conditional Access und alles, was nach dem Umzug kommt – findest du auf der Seite [Microsoft Entra ID](#). Hier gehen wir davon aus, dass Entra Connect deine Benutzer und Gruppen bereits synchronisiert. Ohne das zieht keine Anwendung um, denn Entra ID kann nur Claims ausstellen für Benutzer, die es kennt.

FAKTEN – SAML-Migration von ADFS nach Entra ID in Kürze

SAML-2.0-Anwendungen bindest du in Entra ID als Enterprise Application an – entweder aus der App-Galerie oder als eigene Anwendung außerhalb der Galerie (non-gallery).

Aus Sicht des Dienstleisters ändern sich Aussteller (Microsoft Entra Identifier mit deiner Tenant-ID), Anmelde-URL (der SAML-Endpunkt deines Tenants) und das Signaturzertifikat. Identifier und Antwort-URL der Anwendung bleiben in der Regel gleich.

Entra ID erzeugt beim Einrichten von SAML ein eigenes, selbstsigniertes Signaturzertifikat je

Anwendung, gültig für drei Jahre, und warnt 60, 30 und 7 Tage vor Ablauf per E-Mail.

Pro Claim erlaubt Entra ID höchstens zwei verkettete Transformationen. `RegexReplace()` gibt es, aber keine frei programmierbare Claim-Rule-Sprache und keine fremden Attributspeicher wie SQL oder LDAP.

Was ADFS mit Access Control Policies und Autorisierungsregeln erledigt, übernehmen in Entra ID die Benutzerzuweisung („Zuweisung erforderlich“) und Conditional Access.

Bevor du klickst: Die Relying Party auseinandernehmen

Der häufigste Fehler bei dieser Migration passiert, bevor überhaupt jemand das Entra Admin Center öffnet: Man schaut sich die Relying Party in der ADFS-Konsole kurz an, sieht „E-Mail-Adresse als NameID“ und denkt, das sei es gewesen. Ist es selten. Eine gewachsene Relying Party hat Endpunkte, die niemand mehr kennt, eine Signaturoption, die vom Standard abweicht, eine Verschlüsselung, die irgendwann jemand eingeschaltet hat, und Claim Rules, die sich gegenseitig Zwischenergebnisse zuschieben. Wenn du die gesamte Farm vor dir hast, hilft dir der Beitrag [Relying Parties inventarisieren – die Bestandsaufnahme vor der Entra-Migration](#) bei der Priorisierung. Hier geht es um die eine Anwendung, die jetzt dran ist.

Export statt Screenshot

Sichere zuerst die komplette Konfiguration der Relying Party. Das ist dein Bauplan und gleichzeitig dein Rückweg. Ein Screenshot der Konsole reicht nicht, weil er die Hälfte nicht zeigt – etwa die Reihenfolge der Regeln oder Endpunkte mit Index. Das ADFS-Modul liefert alles, was du brauchst:

```
# Auf einem ADFS-Server der Farm, PowerShell als Administrator
$rp = Get-AdfsRelyingPartyTrust -Name "SaaS-Personalportal"

# 1. Vollständiger Export als Rückweg und Bauplan
$rp | Export-Clixml -Path "C:\Migration\rp-personalportal.xml"

# 2. Die Claim Rules im Klartext – das ist die Übersetzungsvorlage
$rp.IssuanceTransformRules | Out-File "C:\Migration\rp-personalportal-claims.txt" -Encoding utf8

# 3. Die Eckdaten, die du in Entra ID nachbauen musst
$rp | Select-Object Name, Identifier, Enabled, AccessControlPolicyName,
    SignatureAlgorithm, SamlResponseSignature, EncryptClaims, TokenLifetime
$rp.SamlEndpoints | Select-Object Protocol, Binding, Location, Index, IsDefault
```

Listing 1: Relying Party exportieren und die migrationsrelevanten Eigenschaften auslesen

Achte bei der Ausgabe besonders auf vier Dinge. Erstens: Hat die Anwendung mehrere Assertion-Consumer-Endpunkte mit unterschiedlichem Index? Dann brauchst du in Entra ID mehrere Antwort-URLs. Zweitens: Steht `SamlResponseSignature` auf etwas anderem als dem ADFS-Standard „AssertionOnly“? Dann musst du die Signaturoption in Entra ID passend einstellen. Drittens: Ist `EncryptClaims` aktiv und ein Verschlüsselungszertifikat hinterlegt? Viertens: Welche Access Control Policy hängt dran – „Permit everyone“ oder eine Gruppe?

Den echten Token mitschneiden

Die Claim Rules zeigen dir, was ADFS ausstellen soll. Der echte Token zeigt dir, was tatsächlich ankommt – und das ist nicht immer dasselbe. Melde dich deshalb vor der Migration mit einem normalen Testkonto an der Anwendung an und schneide die SAML-Antwort mit, zum Beispiel mit den Entwicklerwerkzeugen des Browsers oder einer SAML-Erweiterung. Dekodiere die Base64-Antwort und lege das XML neben den Export. Später vergleichst du den Token aus Entra ID Zeile

für Zeile damit. Diese zehn Minuten sparen dir im Zweifel zwei Tage Fehlersuche mit einem Hersteller-Support, der „bei uns ist alles korrekt konfiguriert“ in vier Sprachen kann.

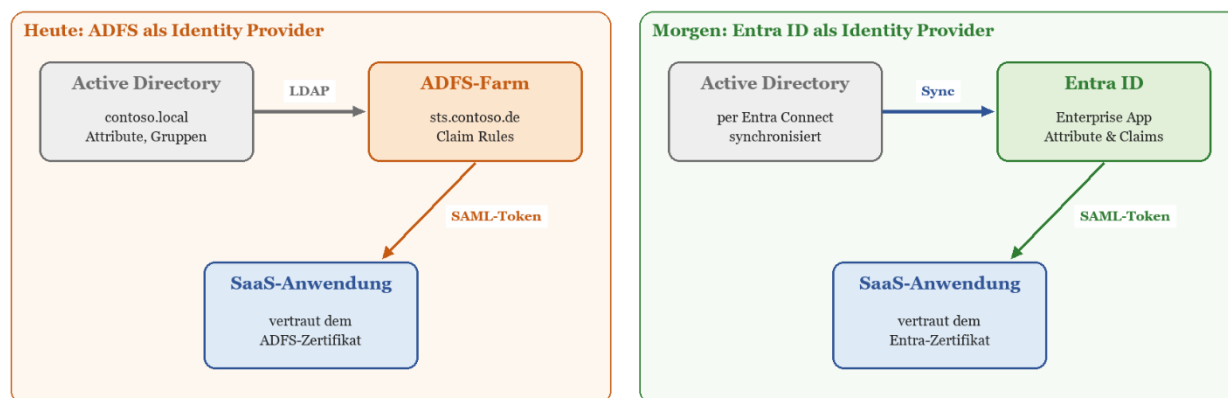
TIPP – Den Migrationsassistenten in Entra ID nutzen – aber nicht blind

Wenn auf deinen ADFS-Servern die Agents von Entra Connect Health laufen, zeigt Entra ID unter Enterprise apps > Usage & insights > AD FS application migration alle Relying Parties mit Anmeldungen der letzten 30 Tage. Jede bekommt einen Status: Ready to migrate, Needs review oder Additional steps required.

Für SAML-Anwendungen gibt es dort sogar eine assistierte Migration, die Identifier, Antwort-URL, kompatible Claims und Gruppenzuweisungen übernimmt. Das Signaturzertifikat wird nicht übernommen, Conditional Access auch nicht. Voraussetzung ist eine Lizenz für Entra ID P1 oder P2. Der Assistent ist ein guter Startpunkt und eine noch bessere Plausibilitätsprüfung. Das Ergebnis prüfst du trotzdem gegen deinen Export und den mitgeschnittenen Token – Microsoft-eigene Relying Parties wie Microsoft 365 tauchen dort übrigens gar nicht erst auf.

Skizze 1: Vorher, nachher – was der Dienstleister davon mitbekommt

Die Anwendung vertraut künftig einem anderen Aussteller. Alles andere soll für sie gleich aussehen.



Was beim Dienstleister ausgetauscht wird

Aussteller (Issuer)	<code>http://sts.contoso.de/adfs/services/trust</code>	Microsoft-Aussteller mit deiner Tenant-ID
Anmelde-URL	<code>https://sts.contoso.de/adfs/ls/</code>	SAML-Endpunkt deines Tenants
Signaturzertifikat	Token-Signing der Farm	eigenes Zertifikat je Enterprise App
NameID muss passen	z. B. <code>sAMAccountName</code> aus AD	<code>user.onpremisesamaccountname</code>

Skizze 1: Für die Anwendung ändert sich der Aussteller samt Zertifikat. Der NameID-Wert muss identisch bleiben, sonst findet sie ihre Benutzer nicht wieder.

Enterprise Application anlegen und Basis-SAML konfigurieren

In Entra ID heißt das Gegenstück zur Relying Party „Enterprise Application“. Technisch besteht sie aus einer App-Registrierung und einem Dienstprinzipal, aber das darf dir für SAML weitgehend egal sein: Du arbeitest im Entra Admin Center unter Entra ID > Enterprise apps > All applications > New application. Gibt es die Anwendung in der Galerie, nimm die Galerievorlage – sie bringt oft passende Claims und eine Schritt-für-Schritt-Anleitung des Herstellers mit. Gibt es sie nicht, wählst du „Create your own application“ und „Integrate any other application you don't find in the gallery (Non-gallery)“. Danach unter Single sign-on die Methode SAML auswählen.

Was du aus der Relying Party übernimmst

Einstellung in der ADFS-Relying-Party	Entsprechung in der Enterprise App	Hinweis aus der Praxis
Bezeichner (Identifiers)	Basic SAML Configuration › Identifier (Entity ID)	Muss im Tenant eindeutig sein. Neben der ADFS-RP darf er parallel existieren, denn das ist ein anderer Aussteller.
Endpunkte › SAML Assertion Consumer	Basic SAML Configuration › Reply URL (Assertion Consumer Service URL)	Mehrere Endpunkte mit Index sind möglich. Den Standard-Endpunkt markieren.
Endpunkte › SAML Logout	Basic SAML Configuration › Logout URL	Wird oft vergessen. Ohne sie endet die Abmeldung in der Anwendung im Nirgendwo.
–	Basic SAML Configuration › Sign on URL	Nur für SP-initiierte Anmeldung aus My Apps. Bei ADFS gibt es keine direkte Entsprechung.
Erweitert › Sicherer Hashalgorithmus	SAML Certificates › Signing Algorithm	SHA-256 ist Standard. SHA-1 nur, wenn die Anwendung wirklich nichts anderes kann.
SamlResponseSignature (nur PowerShell)	SAML Certificates › Signing Option	AssertionOnly entspricht „Sign SAML assertion“, MessageAndAssertion entspricht „Sign SAML response and assertion“.
Verschlüsselung › Zertifikat	Token encryption	Funktion von Entra ID P1/P2. Öffentlichen Schlüssel der Anwendung hochladen und aktivieren.
Signatur › Zertifikat für signierte Anforderungen	Verification certificates (signierte Anforderungen erzwingen)	Nur aktivieren, wenn die Anwendung ihre AuthnRequests tatsächlich signiert.
Überwachung › Metadaten-URL der Anwendung	nicht vorhanden	Entra ID liest keine Metadaten der Anwendung automatisch nach. Änderungen beim Dienstleister pflegst du von Hand.
Access Control Policy	Properties › Assignment required + Users and groups	Gruppen zuweisen statt Regeln schreiben. Bedingungen wie MFA oder Netzwerk gehören in Conditional Access.
TokenLifetime	Token-Lebensdauer-Richtlinie (nur per Graph)	Selten nötig. Wenn die ADFS-RP einen abweichenden Wert hatte, frag nach dem Grund, bevor du ihn nachbaust.

Tabelle 1: Die Eckdaten einer ADFS-Relying-Party und wo du sie in Entra ID wiederfindest

Dasselbe per Microsoft Graph PowerShell

Bei einer Anwendung klickst du. Bei zwanzig nicht. Für wiederholbare Migrationen legst du die Enterprise App mit Microsoft Graph PowerShell an. Die Vorlagen-ID 8adf8e6e-67b2-4cf2-a259-e3dc5476c621 steht für die Nicht-Galerie-Anwendung und ist in der Microsoft-Dokumentation genau dafür beschrieben. Früher stand an dieser Stelle in jeder Anleitung New-AzureADApplication aus dem AzureAD-Modul – das Modul ist ausgemustert, der Weg führt heute über Graph.

```
Connect-MgGraph -Scopes "Application.ReadWrite.All", "AppRoleAssignment.ReadWrite.All",
"Group.Read.All"

# Nicht-Galerie-Anwendung aus der Vorlage erzeugen (App-Registrierung + Dienstprinzipal)
$neu = Invoke-MgInstantiateApplicationTemplate `
    -ApplicationTemplateId "8adf8e6e-67b2-4cf2-a259-e3dc5476c621" `
    -DisplayName "Personalportal (SAML)"
$appId = $neu.Application.Id
$spId = $neu.ServicePrincipal.Id
Start-Sleep -Seconds 30 # Replikation abwarten, sonst laufen die nächsten Aufrufe ins Leere

# SAML als SSO-Modus, Zuweisung erzwingen
```

```
Update-MgServicePrincipal -ServicePrincipalId $spId -PreferredSingleSignOnMode "saml" `
-AppRoleAssignmentRequired:$true

# Identifier und Antwort-URL aus dem ADFS-Export übernehmen (Werte sind Platzhalter)
Update-MgApplication -ApplicationId $appId `
-IdentifierUri @"(https://portal.contoso.com/saml)" `
-Web @{ RedirectUri = @"(https://portal.contoso.com/saml/acs)" }

# Signaturzertifikat erzeugen und als aktives Zertifikat setzen
$zert = Add-MgServicePrincipalTokenSigningCertificate -ServicePrincipalId $spId `
-DisplayName "CN=Personalportal SAML" -EndDateTime (Get-Date).AddYears(2)
Update-MgServicePrincipal -ServicePrincipalId $spId -PreferredTokenSigningKeyThumbprint
$zert.Thumbprint

# Die Gruppe aus der Access Control Policy zuweisen
$gruppe = Get-MgGroup -Filter "displayName eq 'APP-Personalportal-Benutzer'"
$rolle = (Get-MgServicePrincipal -ServicePrincipalId $spId).AppRoles |
Where-Object { $_.DisplayName -eq "User" }
New-MgServicePrincipalAppRoleAssignedTo -ServicePrincipalId $spId `
-PrincipalId $gruppe.Id -ResourceId $spId -AppRoleId $rolle.Id
```

Listing 2: Enterprise App für SAML per Microsoft Graph PowerShell anlegen, Zertifikat erzeugen, Gruppe zuweisen

WARNUNG – „Zuweisung erforderlich“ ist kein Detail

Steht Assignment required auf „No“, darf sich jeder Benutzer deines Tenants an der Anwendung anmelden – auch Gäste, je nach Einstellung. In ADFS war das Gegenstück eine Access Control Policy, die vielleicht nur eine einzige Gruppe durchgelassen hat. Wer diese Einstellung beim Umzug übersieht, hat aus einer Anwendung für zwölf Personen aus dem Personalwesen eine Anwendung für alle gemacht. Das fällt meistens jemandem auf, der es nicht hätte sehen sollen.

Bedingungen, die über „wer darf“ hinausgehen – MFA nur von außen, Zugriff nur von verwalteten Geräten, Sperre für bestimmte Länder – baust du nicht in der Enterprise App, sondern als Conditional-Access-Richtlinie mit dieser Anwendung als Ziel. Wie du dabei aus Autorisierungsregeln und Access Control Policies saubere Richtlinien machst, steht im Beitrag [Claim Rules nach Conditional Access übersetzen](#). Wer in ADFS bereits mit [Access Control Policies in ADFS – Zugriffsregeln ohne Claim-Rule-Akrobatik](#) gearbeitet hat, hat es hier deutlich leichter als jemand, der dieselbe Logik in zwanzig handgeschriebenen Autorisierungsregeln versteckt hat.

Claims-Mapping übertragen: Wenn Transformationen anders heißen

Jetzt kommt der Teil, an dem sich entscheidet, ob die Migration ein ruhiger Dienstag wird. ADFS hat eine eigene Claim-Rule-Sprache mit Bedingungen, Attributspeichern, Zwischenergebnissen und regulären Ausdrücken – im Prinzip eine kleine Programmiersprache, die man lieben oder fürchten kann. Wer tiefer einsteigen will, findet die Grundlagen im [ADFS Claim Rules Leitaden](#). Entra ID dagegen arbeitet deklarativ: Du wählst unter Single sign-on > Attributes & Claims für jeden Claim ein Quellattribut, optional eine Bedingung und höchstens zwei Transformationen. Das klingt nach weniger, und es ist auch weniger. Für die allermeisten SaaS-Anwendungen reicht es trotzdem.

Skizze 2: Zwei Pipelines, ein Ziel – derselbe Token

ADFS rechnet mit Regeln und Attributspeichern, Entra ID mit Quellattributen und höchstens zwei Transformationen

ADFS: Claim-Rule-Sprache



Entra ID: Attribute & Claims



Der Engpass ist fast nie die Transformation, sondern die Datenquelle: Was Entra Connect nicht synchronisiert, kann Entra ID auch nicht in einen Token schreiben.

Skizze 2: ADFS rechnet mit Regeln, Entra ID mit Quellattributen. Fehlt das Attribut in Entra ID, hilft keine Transformation der Welt.

Erst die Standard-Claims aufräumen

Eine frisch angelegte Enterprise App stellt ungefragt Claims aus: NameID mit dem UPN, dazu Vorname, Nachname, E-Mail-Adresse und Name. ADFS dagegen schickt nur, was eine Regel ausdrücklich erzeugt. Manche Anwendungen stören sich an zusätzlichen Attributen nicht, andere überschreiben damit fröhlich Profildaten. Gleiche die Liste deshalb mit dem mitgeschnittenen ADFS-Token ab und entferne alles, was dort nicht vorkam. Achte auch auf den Namen des Claims: Entra ID trennt Name und Namespace. Der ADFS-Claim für die E-Mail-Adresse aus dem bekannten Namespace .../ws/2005/05/identity/claims wird also zu Name „emailaddress“ mit eben diesem Namespace. Hat die Anwendung kurze Attributnamen wie „email“ erwartet, lässt du den Namespace leer.

Die NameID ist heilig

Wenn du nur einen Satz aus diesem Beitrag mitnimmst, dann diesen: Die NameID muss nach der Migration exakt denselben Wert haben wie vorher. Die Anwendung verknüpft ihre Benutzerkonten über diesen Wert. Hat ADFS den sAMAccountName geschickt und Entra ID schickt jetzt den UPN, legt die Anwendung im besten Fall neue, leere Konten an. Im schlechtesten Fall passt ein UPN zufällig auf ein altes Konto, das jemand anderem gehört hat.

In Entra ID stehen für die NameID unter anderem user.userprincipalname, user.mail, user.onpremisesamaccountname, user.employeeid, user.objectid, die Erweiterungsattribute 1 bis 15 und Verzeichniserweiterungen aus Entra Connect zur Auswahl. Das Format wählst du separat: Default, Persistent, Email address, Unspecified oder Windows domain qualified name. Schickt die Anwendung in ihrer Anfrage eine NameIDPolicy mit Format mit, gewinnt übrigens diese – auch das ist ein Grund, den alten Token mitzuschneiden.

WICHTIG – Der Sonderfall objectGUID und ms-DS-ConsistencyGuid

Manche Anwendungen wurden mit einer persistenten NameID auf Basis der objectGUID oder eines anderen AD-Attributs angebunden, das Entra ID nicht direkt als Quelle anbietet. Dann gibt es zwei

saubere Wege: Das Attribut per Entra Connect in ein Erweiterungsattribut oder eine Verzeichniserweiterung synchronisieren und von dort ausstellen – oder mit dem Dienstleister eine einmalige Umschlüsselung der Konten auf einen neuen Identifier vereinbaren. Der dritte Weg, „wird schon passen“, ist keiner.

Typische Claim-Rule-Konstrukte und ihre Entra-Entsprechung

Die folgende Tabelle ist das Herzstück jeder Übersetzung. Sie listet die Konstrukte, die in gewachsenen Relying Parties immer wieder auftauchen, und was du in Entra ID daraus machst. Die Namen der Funktionen sind die aus dem Entra Admin Center – also englisch, mit Klammern und ohne Gnade.

Konstrukt in ADFS	Entsprechung in Entra ID	Stolperstein
Vorlage „LDAP-Attribute als Claims senden“ (mail, givenName, sn, department ...)	Claim mit Quellattribut user.mail, user.givenname, user.surname, user.department	Nur synchronisierte Attribute sind verfügbar. Exotische AD-Attribute erst per Entra Connect in ein Erweiterungsattribut bringen.
Vorlage „Eingehenden Claim umwandeln“ (z. B. E-Mail zu NameID mit Format)	Name identifier value mit Quellattribut und Name identifier format	Formatwahl prüfen: Default ist nicht dasselbe wie Unspecified.
Vorlage „Gruppenmitgliedschaft als Claim senden“ (fester Wert, wenn Mitglied)	Claim conditions: Benutzertyp plus Gruppe, Quelle ist ein konstanter Wert	Bis zu 50 Gruppen je Anwendung über alle Claims. Für Rollen oft eleganter: App-Rollen, die Entra ID als role-Claim ausstellt.
Alle Gruppen über tokenGroups ausstellen	Gruppen-Claim (Add a group claim), z. B. „Groups assigned to the application“, Ausgabe als sAMAccountName	Ab 150 Gruppen schickt Entra ID im SAML-Token keine Liste mehr, sondern einen Verweis auf Graph. Damit kann kaum eine SaaS-Anwendung etwas anfangen – Gruppen filtern.
Konstanter Wert: => issue(Type = "...", Value = "contoso")	Claim mit konstantem Wert als Source attribute (ohne Anführungszeichen eingeben)	Keiner – das ist der einfachste Fall der ganzen Migration.
Verkettung: Value = c1.Value + "@" + c2.Value	Join() mit Trennzeichen	Bei der NameID entfernt Join() einen vorhandenen Domänenteil vor dem Verketteten.
Präfix abschneiden mit RegExReplace(c.Value, "^CONTOSO\\", "")	Extract() – After matching, ExtractMailPrefix() oder RegExReplace()	RegexReplace() kann mehr, ist aber schwerer zu warten. Die einfachste Funktion gewinnt.
Groß- und Kleinschreibung erzwingen	ToLowercase() bzw. ToUppercase()	Wird bei sAMAccountName-basierten IDs gern übersehen – manche Anwendungen unterscheiden Schreibweisen.
NOT EXISTS([Type == "..."]) => issue(...) als Rückfallwert	IfEmpty() bzw. IfNotEmpty()	Funktioniert nur innerhalb eines Claims, nicht über mehrere Claims hinweg.
Bedingung mit regulärem Ausdruck: c:[Type == "...", Value =~ "^DE"]	StartWith(), EndWith(), Contains() mit Ausgabe bei Treffer und Ausgabe bei Nichttreffer	Der Migrationsassistent meldet dieses Muster als UNSUPPORTED_CONDITION_PARAMETER – lösbar, aber Handarbeit.
Mehrstufig: erst add(...) als Zwischenergebnis, dann issue(...)	Zwei verkettete Transformationen am selben Claim	Mehr als zwei Stufen gehen nicht. Dann Logik vereinfachen oder das Ergebnis vorab in ein Attribut schreiben.
Attributsspeicher SQL, LDAP oder eigene DLL	Kein direktes Gegenstück. Daten nach Entra ID synchronisieren oder Custom Claims Provider prüfen	Der echte Blocker. Hier entscheidet sich, ob die Anwendung überhaupt umziehen kann.

Autorisierungsregeln (Issuance Authorization Rules) oder Access Control Policy	Assignment required, Benutzer- und Gruppenzuweisung, Conditional Access	Gehört nicht in die Claims, sondern in Zuweisung und Richtlinien.
Claims wie insidecorporatenetwork oder x-ms-forwarded-client-ip	Benannte Standorte in Conditional Access	Kein Claim-Ersatz. Wenn die Anwendung selbst nach dem Netz entscheidet, mit dem Hersteller reden.

Tabelle 2: Typische ADFS-Claim-Rule-Konstrukte und ihre Entsprechung in Entra ID

Ein Beispiel, einmal komplett übersetzt

Nehmen wir eine typische Relying Party für ein Personalportal. Sie stellt E-Mail, Vor- und Nachname aus dem AD aus, baut eine Portal-ID aus Präfix und Personalnummer und vergibt eine Admin-Rolle über eine Gruppenmitgliedschaft. In ADFS sieht das so aus:

```
@RuleName = "LDAP-Attribute"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname",
  Issuer == "AD AUTHORITY"]
=> issue(store = "Active Directory",
  types = ("http://contoso.de/claims/email",
    "http://contoso.de/claims/vorname",
    "http://contoso.de/claims/nachname",
    "http://contoso.de/claims/personalnummer"),
  query = ";mail,givenName,sn,employeeID;{0}", param = c.Value);

@RuleName = "Portal-ID aus Präfix und Personalnummer"
c:[Type == "http://contoso.de/claims/personalnummer"]
=> issue(Type = "http://contoso.de/claims/portalid", Value = "CT-" + c.Value);

@RuleName = "Admin-Rolle über Gruppe"
c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid",
  Value == "S-1-5-21-1111111111-2222222222-3333333333-4711"]
=> issue(Type = "http://contoso.de/claims/portalrolle", Value = "PortalAdmin");
```

Listing 3: Drei Claim Rules einer typischen SaaS-Relying-Party (Werte anonymisiert)

In Entra ID wird daraus keine einzige Zeile Code, sondern eine Handvoll Einträge unter Attributes & Claims:

Claim (Namespace + Name)	Quelle in Entra ID	Transformation / Bedingung
http://contoso.de/claims/email	user.mail	–
http://contoso.de/claims/vorname	user.givenname	–
http://contoso.de/claims/nachname	user.surname	–
http://contoso.de/claims/portalid	Transformation	Join() mit Parameter 1 = Konstante „CT“, Trennzeichen „-“, Parameter 2 = user.employeeid
http://contoso.de/claims/portalrolle	Konstante „PortalAdmin“	Claim condition: Benutzertyp Members, Gruppe APP-Personalportal-Admins

Tabelle 3: Die drei Regeln aus Listing 3 als Einträge unter Attributes & Claims

Zwei Dinge fallen auf. Erstens verschwindet die Zwischenstufe mit dem Personalnummer-Claim komplett, weil Entra ID direkt auf user.employeeid zugreift – vorausgesetzt, Entra Connect synchronisiert employeeID, was in der Standardkonfiguration der Fall ist. Zweitens wird aus dem Vergleich auf eine Gruppen-SID eine Bedingung auf die synchronisierte Gruppe. Wer die Gruppe vorher nicht in den Sync-Bereich aufgenommen hat, sucht sie an dieser Stelle vergeblich und wundert sich.

TIPP — Transformationen testen, bevor ein Benutzer es tut

Für RegexReplace() bietet Entra ID direkt im Dialog „Test transformation“ an – mit Testwerten, nicht mit echten Benutzerdaten. Für alle anderen Funktionen gilt: Testbenutzer mit bewusst unbequemen

Werten anlegen. Leere Personalnummer, Umlaute im Nachnamen, ein Bindestrich im Vornamen. Genau die Benutzer, die im echten Leben zuerst anrufen.

Zertifikat und Metadaten beim Dienstleister tauschen

Bis hierhin hast du nur in deinem eigenen Tenant gearbeitet, und nichts ist passiert. Die Anwendung vertraut weiter ADFS, die neue Enterprise App liegt still daneben. Der eigentliche Umzug findet auf der anderen Seite statt: beim Dienstleister. Dort ersetzt du die IdP-Konfiguration – und das ist der Moment, in dem du die Hoheit über den Zeitplan teilweise abgibst.

Was der Dienstleister von dir bekommt

Auf der Seite Single sign-on der Enterprise App findest du unter „Set up ...“ und „SAML Certificates“ alles, was die Anwendung braucht. Die meisten Anwendungen akzeptieren eine dieser drei Varianten:

- Die App Federation Metadata URL der Enterprise App. Sie ist anwendungsspezifisch, weil sie die Anwendungs-ID als Parameter appid enthält, und liefert Aussteller, Anmelde-URL und Zertifikat in einem Rutsch. Kann die Anwendung Metadaten regelmäßig nachladen, ist das der Königsweg – spätere Zertifikatswechsel erledigen sich dann fast von selbst.
- Die Federation Metadata XML als Datei zum Hochladen. Gleicher Inhalt, aber eingefroren.
- Die Einzelwerte: Login URL, Microsoft Entra Identifier, Logout URL und das Zertifikat als Base64-, Raw- oder PEM-Datei. Das ist der klassische Weg für Anwendungen mit einem Formular und drei Textfeldern.

WARNUNG — Das Zertifikat ist anwendungsspezifisch – nicht das von ADFS

Bei ADFS signiert das eine Token-Signing-Zertifikat der Farm alle Relying Parties. Entra ID hat für jede Enterprise App ein eigenes Signaturzertifikat. Lade beim Dienstleister also genau das Zertifikat dieser Enterprise App hoch, nicht ein Zertifikat aus einer anderen Anwendung und schon gar nicht das der Farm. Klingt banal. Ist trotzdem die zweithäufigste Ursache für „Signature validation failed“ am Umschalttag.

Ablaufdatum und Benachrichtigung gleich mit erledigen

Das von Entra ID erzeugte Zertifikat ist drei Jahre gültig. Das Ablaufdatum kannst du nach dem Speichern nicht mehr ändern; wenn du eine kürzere Laufzeit willst, legst du über „New Certificate“ ein neues an, wählst das Datum und machst es aktiv. Entra ID schickt 60, 30 und 7 Tage vor Ablauf eine E-Mail an bis zu fünf Adressen. Standardmäßig ist das nur die Adresse der Person, die die Anwendung angelegt hat – und die arbeitet in drei Jahren mit einiger Wahrscheinlichkeit woanders. Trag deshalb sofort eine Verteilerliste ein. Wer das Thema Zertifikatswechsel aus der ADFS-Welt kennt, findet die Parallelen im Beitrag [ADFS Token-Signing-Zertifikat erneuern – mit und ohne AutoCertificateRollover](#) – mit dem Unterschied, dass es nach der Migration nicht mehr ein Zertifikat für alle gibt, sondern eines pro Anwendung.

Aspekt	ADFS	Entra ID
Zertifikat	ein Token-Signing-Zertifikat für die ganze Farm	eigenes Signaturzertifikat je Enterprise App
Erneuerung	AutoCertificateRollover oder manuell	neues Zertifikat anlegen, beim Dienstleister hinterlegen, aktiv setzen
Laufzeit	abhängig von Konfiguration und Rollover	Standard drei Jahre, kürzer wählbar

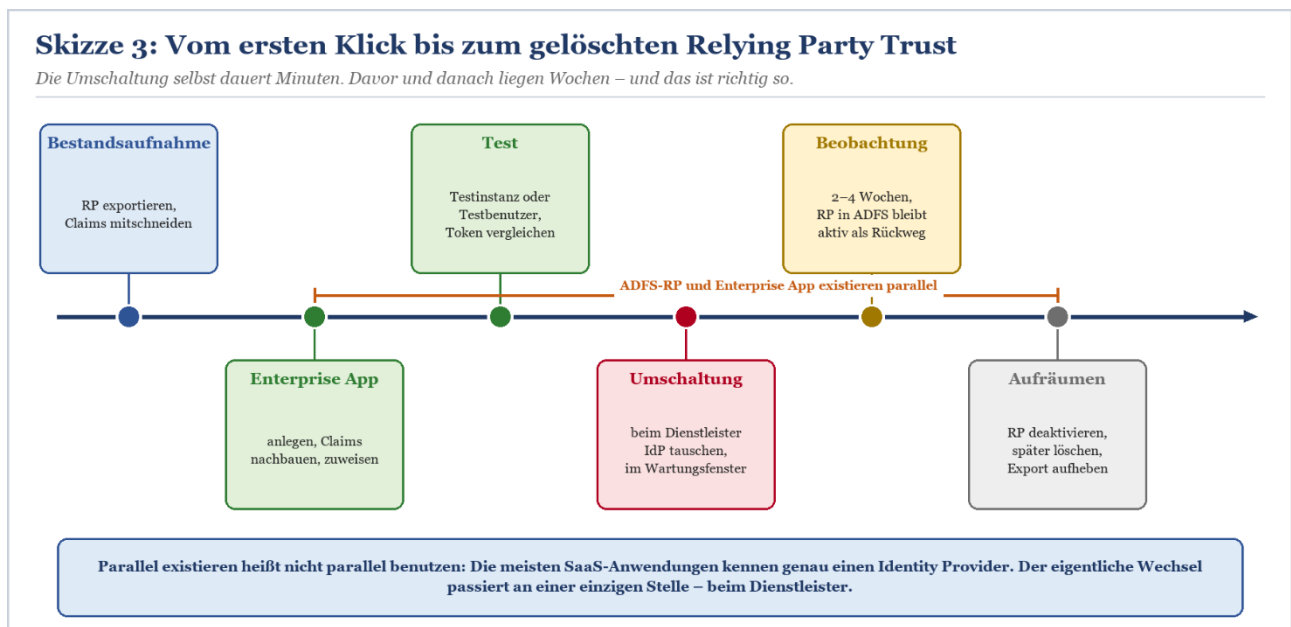
Warnung vor Ablauf	eigene Überwachung nötig	E-Mail 60, 30 und 7 Tage vorher an bis zu fünf Adressen
Schutz des privaten Schlüssels	deine Verantwortung auf der Farm	liegt bei Microsoft, sofern du kein eigenes Zertifikat hochlädst

Tabelle 4: Signaturzertifikate in ADFS und Entra ID im Vergleich

Der letzte Punkt der Tabelle ist kein Nebensatz. Jede Anwendung, die du von der Farm nimmst, ist eine Anwendung weniger, deren Tokens ein gestohlener Token-Signing-Schlüssel fälschen könnte. Was Golden SAML bedeutet und warum der Schlüssel der Farm so wertvoll ist, erklärt der Beitrag [ADFS-Sicherheit: Angriffe, Härtung, Golden SAML und MFA](#).

Parallelbetrieb, Umschaltzeitpunkt und Test

„Parallelbetrieb“ ist bei SAML-Anwendungen ein großes Wort für eine meist kleine Sache. Auf deiner Seite existieren Relying Party und Enterprise App problemlos nebeneinander – sie haben denselben Identifier, aber unterschiedliche Aussteller, und keiner der beiden weiß vom anderen. Auf der Seite der Anwendung sieht es anders aus: Die meisten SaaS-Anwendungen kennen genau einen Identity Provider. Wie weich dein Umzug wird, entscheidet also nicht Microsoft, sondern der Hersteller.



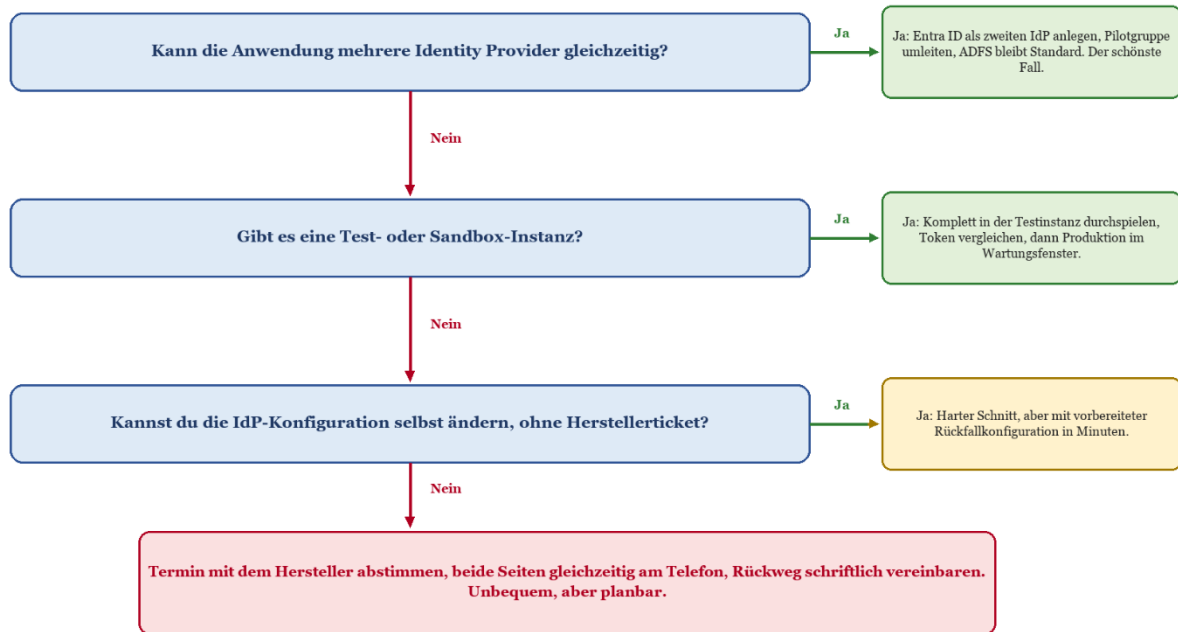
Skizze 3: Die Umschaltung dauert Minuten. Die Wochen davor und danach entscheiden, ob es dabei bleibt.

Drei Varianten, die du vorfinden wirst

Im besten Fall kann die Anwendung mehrere Identity Provider gleichzeitig, oft gesteuert über die E-Mail-Domäne oder eine Gruppenzuordnung. Dann legst du Entra ID als zweiten IdP an, schickst eine Pilotgruppe darüber und lässt ADFS für alle anderen Standard. Im zweitbesten Fall gibt es eine Test- oder Sandbox-Instanz, die du komplett auf Entra ID umstellst und durchtestest, bevor die Produktion an der Reihe ist – genau das empfiehlt auch Microsoft in seinem Phasenmodell. Im ungemütlichsten Fall gibt es nur die Produktion und genau einen IdP-Eintrag. Dann ist die Umschaltung ein harter Schnitt, und deine Vorbereitung entscheidet, ob er fünf Minuten oder einen Nachmittag dauert.

Skizze 4: Wie viel Parallelbetrieb erlaubt dein Dienstleister?

Drei Fragen entscheiden, ob du sanft umziehst oder einen harten Schnitt planst



Skizze 4: Drei Fragen an den Dienstleister bestimmen, wie viel Parallelbetrieb möglich ist

Den Umschaltzeitpunkt wählen

Ein guter Umschaltzeitpunkt hat drei Eigenschaften: wenig Betrieb in der Anwendung, beide Seiten erreichbar und genug Zeit bis zum nächsten kritischen Termin. Ein Personalportal stellst du nicht am Tag vor der Gehaltsabrechnung um, ein Reisekostentool nicht am Monatsende. Bestehende Sitzungen in der Anwendung laufen nach der Umstellung meist bis zu ihrem Ablauf weiter, deshalb merken viele Benutzer den Wechsel erst bei der nächsten Anmeldung. Das ist bequem, kann aber Fehler verschleiern. Plane die Prüfung deshalb mit frischen Sitzungen, also im privaten Browserfenster.

WICHTIG – Lesezeichen auf die ADFS-Anmeldeseite

IdP-initiierte Anmeldungen laufen bei ADFS oft über die Seite `idpinitiatedsignon` oder über Links mit `loginToRp`-Parameter auf `sts.contoso.de`. Diese Links funktionieren nach der Umstellung entweder nicht mehr oder – schlimmer – führen weiter zur alten Relying Party, solange die noch aktiv ist. Ersetze sie im Intranet, in Portalen und Dokumentationen durch den Link aus My Apps oder die Anmelde-URL der Anwendung. Am zuverlässigsten findest du solche Links im Audit-Log der Farm, nicht in der Erinnerung der Kollegen.

Testen wie die Buchhaltung, nicht wie der Admin

Entra ID bietet auf der Seite Single sign-on die Funktion „Test single sign-on“. Sie zeigt dir bei Fehlern die konkrete Fehlermeldung samt Lösungshinweis und ist ein guter erster Schritt. Der eigentliche Test ist aber der Vergleich: Melde dich mit einem Testbenutzer an, schneide die SAML-Antwort mit und lege sie neben den ADFS-Token aus Kapitel 1. Prüfe NameID, Format, jeden einzelnen Claim und die Gruppen. Danach meldest du dich mit einem Konto an, das der Anwendung nicht zugewiesen ist – das muss scheitern. Klappt es trotzdem, steht Assignment required noch auf „No“.

Skizze 5: Die Testmatrix vor dem Go

Grün heißt: Token enthält dasselbe wie bei ADFS, und der Benutzer landet im richtigen Konto

	SP-initiiert	IdP-initiiert	Neuer Benutzer	Gast / extern	Abmelden
NameID identisch	Pflicht	Pflicht	Pflicht	wenn genutzt	Pflicht
Attribute vollständig	Pflicht	Pflicht	Pflicht	wenn genutzt	–
Gruppen / Rollen	Pflicht	wenn genutzt	Pflicht	–	–
Zugriff verweigert, wenn nicht zugewiesen	Pflicht	Pflicht	Pflicht	Pflicht	–

Der häufigste Fehler im Test: Der Admin testet mit dem eigenen Konto, das zufällig überall Mitglied ist. Nimm ein Konto, das so langweilig ist wie das der Buchhaltung.

Skizze 5: Nicht jede Zelle ist für jede Anwendung relevant. Die Zeile „NameID identisch“ ist es immer.

Im Fehlerfall sind die Anmeldeprotokolle von Entra ID dein bester Freund: Dort siehst du je Anmeldung die Anwendung, den Fehlercode und ob eine Conditional-Access-Richtlinie gegriffen hat. Typische Fehlerbilder in der ersten Stunde sind eine nicht passende Antwort-URL (die Anwendung schickt eine andere ACS-Adresse als hinterlegt), ein unbekannter Identifier (Tippfehler oder abschließender Schrägstrich) und ein Benutzer ohne Zuweisung. Auf der Seite der Anwendung sind es fast immer das falsche Zertifikat oder eine NameID, die zu keinem Konto passt. Für den Vergleich mit dem alten Verhalten hilft dir der Beitrag [ADFS-Ereignisprotokolle lesen – Admin-Log, Debug-Tracing und Auditing](#) – solange die Relying Party noch aktiv ist, kannst du dort jederzeit nachsehen, was ADFS ausgestellt hat.

Der Rückweg und das Aufräumen

Der Rückweg ist bei SAML-Anwendungen erfreulich simpel, solange du ihn vorbereitet hast: Die alte IdP-Konfiguration beim Dienstleister (ADFS-Metadaten oder die drei Einzelwerte plus Zertifikat) liegt griffbereit, die Relying Party in ADFS bleibt aktiv. Im Notfall trägst du die alten Werte wieder ein, und die Anwendung spricht wieder mit der Farm. Deshalb deaktivierst du die Relying Party erst nach einer Beobachtungsphase von zwei bis vier Wochen – mindestens einen Monatsabschluss sollte die Anwendung ohne Rückfallwunsch überstanden haben.

```
# Nach der Beobachtungsphase: Relying Party deaktivieren, aber noch nicht löschen
Disable-AdfsRelyingPartyTrust -TargetName "SaaS-Personalportal"

# Rückweg in der Beobachtungsphase oder kurz danach
Enable-AdfsRelyingPartyTrust -TargetName "SaaS-Personalportal"

# Erst wenn sicher ist, dass niemand mehr zurückwill
Remove-AdfsRelyingPartyTrust -TargetName "SaaS-Personalportal"
```

Listing 4: Relying Party nach der Migration stufenweise außer Betrieb nehmen

Den Export aus Listing 1 hebst du auf, auch wenn die Relying Party längst gelöscht ist. Er ist die einzige Dokumentation, die garantiert stimmt. Wenn auf diese Weise eine Anwendung nach der anderen umgezogen ist und auch Microsoft 365 nicht mehr über die Farm läuft – der Weg dorthin steht im Beitrag [Microsoft 365 von Federated auf Managed umstellen – mit Staged Rollout](#) –, bleibt am Ende eine Farm ohne Aufgabe. Was dann zu tun ist, beschreibt [ADFS abschalten – die Farm sauber außer Betrieb nehmen](#). Bis dahin läuft ADFS für alle noch nicht migrierten

Anwendungen einfach weiter, und zwar genauso zuverlässig wie vorher. Niemand zwingt dich, alles in einem Quartal zu erledigen.

WEITER — Passende Beiträge für die nächsten Schritte

- › [Relying Parties inventarisieren – die Bestandsaufnahme vor der Entra-Migration](#)
- › [ADFS Claim Rules Leitfaden](#)
- › [Claim Rules nach Conditional Access übersetzen](#)
- › [SAML-Anwendung an ADFS anbinden – Praxisbeispiel SaaS](#)
- › [Microsoft 365 von Federated auf Managed umstellen – mit Staged Rollout](#)
- › [ADFS abschalten – die Farm sauber außer Betrieb nehmen](#)
- › [Microsoft Entra ID](#)

Fazit: Vier Dinge umziehen, eines davon gründlich

Eine SAML-Anwendung von ADFS nach Entra ID zu migrieren, ist kein Hexenwerk. Identifier und Antwort-URL kopierst du, das Zertifikat lädst du beim Dienstleister hoch, die Zuweisung ersetzt die Access Control Policy. Die eigentliche Arbeit steckt in den Claims – und dort fast nie in der Frage, wie eine Transformation heißt, sondern in der Frage, ob das benötigte Attribut in Entra ID überhaupt ankommt und ob die NameID exakt denselben Wert behält. Wer den alten Token mitschneidet, die Claims Zeile für Zeile vergleicht, mit einem langweiligen Testkonto prüft und die Relying Party ein paar Wochen als Rückweg stehen lässt, erlebt einen Umschalttag, an dem schlicht nichts passiert. Das ist bei Identitätsprojekten das höchste Lob.

Wenn du eine ganze Reihe von Anwendungen vor dir hast und dabei auf Claim Rules stößt, die sich nicht in zwei Transformationen pressen lassen, lohnt sich ein zweiter Blick von außen – dafür gibt es das [Consulting zu ADFS \(Active Directory Federation Services\)](#). Die Hintergründe zu Claim Rules, Relying Parties und Migrationspfaden vertieft [ADFS in der Praxis – das Buch im Detail](#), und wer die Übersetzung von Claim Rules einmal in Ruhe im Labor üben möchte, findet das passende Format in den [ADFS-Schulungen](#).

FAQ: SAML-App von ADFS nach Entra ID migrieren

Wie migriere ich eine SAML-App von ADFS nach Entra ID?

Relying Party exportieren und den aktuellen Token mitschneiden, in Entra ID eine Enterprise Application (Galerie oder Non-gallery) mit SAML anlegen, Identifier, Antwort-URL und Claims übertragen, Benutzer oder Gruppen zuweisen, beim Dienstleister Metadaten bzw. Anmelde-URL, Aussteller und Zertifikat tauschen und anschließend testen. Die Relying Party bleibt bis zum Ende der Beobachtungsphase als Rückweg aktiv.

Kann Entra ID die ADFS-Relying-Party automatisch übernehmen?

Teilweise. Mit Entra Connect Health auf den ADFS-Servern zeigt der Bereich „AD FS application migration“ alle aktiven Relying Parties samt Migrationsstatus und bietet für SAML-Anwendungen eine assistierte Migration an. Signaturzertifikat und Conditional Access werden nicht übernommen, und komplexe Claim Rules musst du selbst nachbauen.

Kann ich die ADFS-Claim-Rule-Sprache in Entra ID weiterverwenden?

Nein. Entra ID kennt keine Claim-Rule-Sprache. Du konfigurierst Claims über Quellattribute, Bedingungen und höchstens zwei Transformationen je Claim, etwa Join(), Extract(), IfEmpty() oder RegexReplace().

Was ist das Gegenstück zu RegExReplace aus ADFS?

Die Transformation RegexReplace(). Oft reichen aber einfachere Funktionen wie Extract(), ExtractMailPrefix(), StartWith() oder ToLowercase(), die leichter zu warten sind.

Warum landen Benutzer nach der Migration in leeren Konten?

Weil sich die NameID geändert hat. Die Anwendung verknüpft Konten über diesen Wert. Prüfe Quellattribut und Format der NameID in Entra ID und vergleiche sie mit dem mitgeschnittenen ADFS-Token.

Kann ich das ADFS-Token-Signing-Zertifikat in Entra ID weiterverwenden?

Das ist nicht vorgesehen und auch nicht sinnvoll. Entra ID erzeugt für jede Enterprise App ein eigenes Signaturzertifikat, das du beim Dienstleister hinterlegst. Der Migrationsassistent übernimmt das ADFS-Zertifikat ausdrücklich nicht.

Können ADFS-Relying-Party und Enterprise App gleichzeitig existieren?

Ja. Sie verwenden denselben Identifier, aber unterschiedliche Aussteller. Ob die Anwendung beide gleichzeitig nutzen kann, hängt vom Dienstleister ab – viele SaaS-Anwendungen erlauben nur einen Identity Provider.

Wie ersetze ich Access Control Policies aus ADFS?

Mit „Assignment required“ in den Eigenschaften der Enterprise App und der Zuweisung von Benutzern oder Gruppen. Bedingungen wie MFA, Netzwerk oder Gerätestatus setzt du mit Conditional Access um.

Was mache ich mit Claims aus SQL- oder LDAP-Attributspeichern?

Entra ID liest keine fremden Attributspeicher. Bringe die Daten per Entra Connect in Erweiterungsattribute oder Verzeichniserweiterungen oder prüfe einen Custom Claims Provider. Gelingt beides nicht, bleibt die Anwendung vorerst auf ADFS.

Wie viele Gruppen kann Entra ID im SAML-Token ausstellen?

Bis zu 150 Gruppen. Darüber hinaus schickt Entra ID einen Verweis auf Microsoft Graph statt der Liste. Filtere deshalb, etwa mit „Groups assigned to the application“.

Wann lösche ich die Relying Party in ADFS?

Erst nach einer Beobachtungsphase von einigen Wochen ohne Rückfallwunsch. Zunächst deaktivieren mit Disable-AdfsRelyingPartyTrust, später löschen mit Remove-AdfsRelyingPartyTrust. Den Export bewahrst du auf.

Muss ich ADFS abschalten, wenn die erste Anwendung umgezogen ist?

Nein. Die Farm betreut weiter alle anderen Relying Parties. Du migrierst Anwendung für Anwendung im eigenen Tempo und nimmst die Farm erst außer Betrieb, wenn sie keine Aufgabe mehr hat.