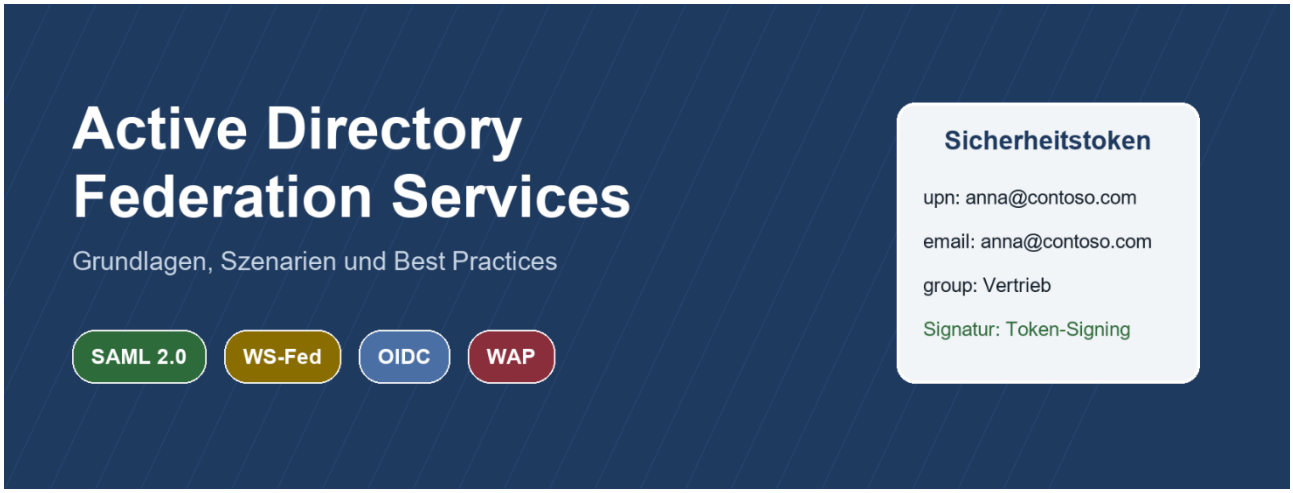
Active Directory Federation Services (ADFS) – Grundlagen, Szenarien und Best Practices



Wissen	Beratung	Schulung
Der große Überblick zu ADFS – vom Grundprinzip bis zur Migration: › Active Directory Federation Services (ADFS) — verstehen, betreiben, migrieren	Health-Check deiner Farm, Härtung, Zertifikatskonzept oder der geordnete Ausstieg Richtung Entra ID – mit jemandem, der das schon oft gemacht hat.	Workshops für dein Admin-Team: Claim-Regeln, Fehlersuche, Betrieb und Notfallübungen an der eigenen Umgebung.

Es gibt Systeme, die man liebt, und Systeme, die man betreibt. Active Directory Federation Services ADFS gehört eindeutig in die zweite Kategorie. Niemand erzählt auf der Weihnachtsfeier begeistert von seiner Föderationsfarm – aber wenn sie am Montagmorgen um 7:40 Uhr ausfällt, weil ein Zertifikat abgelaufen ist, kennt plötzlich jede Abteilungsleiterin und jeder Abteilungsleiter deinen Namen. Und deine Durchwahl.

ADFS ist Microsofts lokaler Identity Provider. Er nimmt die Identitäten aus deinem Active Directory und macht sie für Anwendungen nutzbar, die keine Ahnung von Kerberos haben und auch nie haben werden: Microsoft 365, SaaS-Dienste, Partnerportale, selbst entwickelte Webanwendungen, Cloud-Workloads. Statt jeder Anwendung ein eigenes Benutzerkonto und ein eigenes Kennwort zu spendieren, stellt ADFS ein signiertes Token aus. Die Anwendung prüft die Signatur, liest die mitgelieferten Attribute und lässt den Benutzer herein. Das ist Single Sign-On über Organisationsgrenzen hinweg – und gleichzeitig ein Stück Tier-0-Infrastruktur, das du genauso ernst nehmen musst wie deine Domänencontroller.

In diesem Beitrag bekommst du das komplette Bild: wie der Token-Fluss funktioniert, was Claims und Claim-Regeln wirklich tun, welches Protokoll du wofür nimmst, wann ADFS neben Microsoft Entra ID noch eine Daseinsberechtigung hat, wie du eine Farm samt Web Application Proxy hochverfügbar aufbaust, wie du Zertifikate im Griff behältst, wie du das Ganze gegen Passwort-Spraying und Token-Diebstahl härtest und woran du im Monitoring erkennst, dass gerade etwas schiefläuft. Wer zuerst die große Linie sehen möchte, findet sie auf der Themenseite Active Directory Federation Services (ADFS) — verstehen, betreiben, migrieren.

Fakten

ADFS auf einen Blick
• Serverrolle in Windows Server, ohne Zusatzlizenz; aktuell auf Windows Server 2016, 2019, 2022 und 2025 verfügbar.
• Funktion: Security Token Service (STS) und Identity Provider für Ressourcen außerhalb der eigenen AD-Domäne.
• Protokolle: SAML 2.0, WS-Federation, WS-Trust, OAuth 2.0 und OpenID Connect (seit ADFS in Windows Server 2016).
• Bausteine: Farm aus ADFS-Servern, Konfigurationsdatenbank (WID oder SQL Server), optional Web Application Proxy in der DMZ.
• Microsofts strategische Empfehlung lautet inzwischen ausdrücklich: Statt auf die neueste ADFS-Version zu aktualisieren, nach Microsoft Entra ID migrieren.

Architektur und Token-Fluss: was ADFS eigentlich tut

Der Security Token Service als Übersetzer

Im Kern ist ADFS ein Übersetzer. Auf der einen Seite steht dein Active Directory mit Kerberos, NTLM und LDAP – Protokolle, die im eigenen Netz hervorragend funktionieren und außerhalb davon praktisch nutzlos sind. Auf der anderen Seite stehen Anwendungen, die über HTTPS mit Browsern oder API-Clients reden und nur eines wollen: einen vertrauenswürdigen Nachweis, wer da gerade anklopft und welche Eigenschaften diese Person hat. Der Security Token Service sitzt genau dazwischen. Er authentifiziert den Benutzer gegen das AD, sammelt die benötigten Attribute ein, verarbeitet sie über Regeln und verpackt das Ergebnis in ein Token, das er mit seinem privaten Schlüssel signiert.

Der entscheidende Punkt, den viele beim ersten Kontakt übersehen: Die Anwendung bekommt nie ein Kennwort zu sehen und spricht nie mit einem Domänencontroller. Sie vertraut ausschließlich der Signatur. Wer also den Signaturschlüssel des ADFS besitzt, kann jedem beliebigen Dienst gegenüber jede beliebige Identität behaupten. Diesen Gedanken solltest du im Hinterkopf behalten – er erklärt später fast alle Härtungsmaßnahmen.

Der Weg einer Anmeldung, Schritt für Schritt

Bei den browserbasierten Protokollen SAML 2.0 und WS-Federation läuft eine Anmeldung immer nach demselben Muster ab. Der Browser ruft die Anwendung auf und hat noch keine Sitzung. Die Anwendung leitet ihn mit einer Authentifizierungsanforderung an den ADFS weiter. Der ADFS prüft, ob schon ein SSO-Cookie existiert; falls nicht, authentifiziert er den Benutzer – im internen Netz typischerweise unsichtbar per Kerberos über die integrierte Windows-Authentifizierung, von außen über die Formularanmeldung, gegebenenfalls ergänzt um einen zweiten Faktor. Anschließend holt er die Attribute aus dem AD, jagt sie durch die Claim-Pipeline und schickt das signierte Token per HTTP-POST zurück an den Browser, der es an die Anwendung weiterreicht.

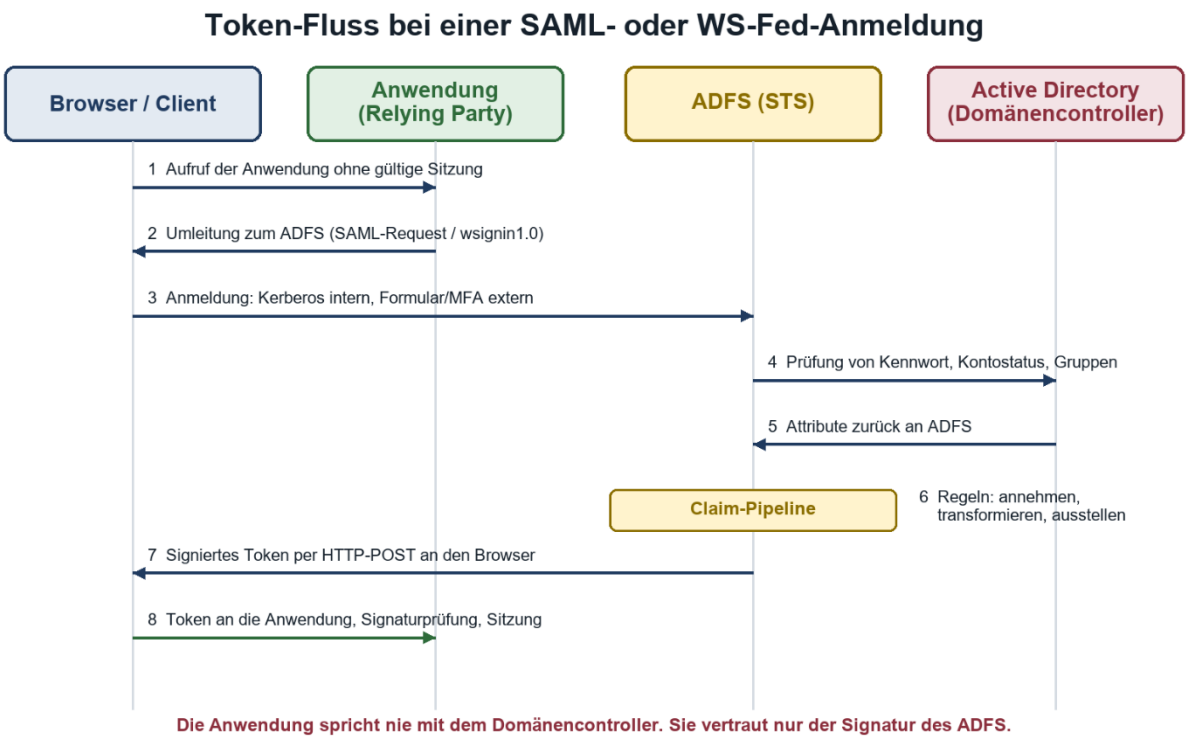


Abbildung 1: Der Token-Fluss. Die Anwendung bekommt nie ein Kennwort zu sehen – sie vertraut nur der Signatur des ADFS.

Dass das Token den Umweg über den Browser nimmt, ist kein Schönheitsfehler, sondern Absicht: Anwendung und ADFS müssen sich gegenseitig gar nicht direkt erreichen können. Eine SaaS-Anwendung im Rechenzentrum eines Anbieters braucht keinen Netzwerkpfad in dein internes Netz. Sie braucht nur das öffentliche Zertifikat, mit dem sie die Signatur prüft – und genau hier liegt der Grund, warum Zertifikatsverwaltung bei ADFS so eine große Rolle spielt.

Relying Party Trusts und Claims Provider Trusts

Vertrauen ist bei ADFS kein Gefühl, sondern ein Konfigurationsobjekt. Es gibt zwei Richtungen. Ein Relying Party Trust beschreibt eine Anwendung, für die dein ADFS Token ausstellt: ihren Bezeichner, die Endpunkte, an die das Token gehen darf, das Zertifikat zur Verschlüsselung, die Ausstellungsregeln und die Zugriffssteuerungsrichtlinie, also etwa „nur Mitglieder der Gruppe Vertrieb, von außen nur mit MFA“. Ein Claims Provider Trust beschreibt die Gegenrichtung: einen Identitätsanbieter, dessen Token dein ADFS akzeptiert. Das lokale Active Directory ist selbst so ein Claims Provider, er ist ab Werk angelegt. Weitere kommen hinzu, wenn du etwa den ADFS eines Partnerunternehmens anbindest, damit dessen Belegschaft deine Anwendungen mit den eigenen Firmenkonten nutzen kann.

Sobald mehr als ein Claims Provider existiert, muss ADFS herausfinden, wo der Benutzer eigentlich zu Hause ist. Das nennt sich Home Realm Discovery und zeigt sich dem Benutzer als Auswahlseite – oder, besser konfiguriert, gar nicht, weil ADFS die Zuordnung anhand des UPN-Suffixes oder eines Parameters aus der Anforderung selbst vornimmt.

Vertrauensstellung	Richtung	Typische Beispiele	Was du darin festlegst
Relying Party Trust	ADFS stellt Token für die Anwendung aus	Microsoft 365, Salesforce, SAP, eigene Web-Apps	Bezeichner, Endpunkte, Ausstellungsregeln, Zugriffsrichtlinie, Verschlüsselung
Claims Provider Trust	ADFS akzeptiert Token eines anderen	Active Directory (Standard), Partner-ADFS, weiterer	Akzeptanzregeln, Signaturzertifikat des

	IdP	SAML-IdP	Partners, Home Realm Discovery
Anwendungsgruppe (OAuth/OIDC)	ADFS als Autorisierungsserver	Single-Page-App, native App, Web-API	Client-ID, Redirect-URIs, Geheimnis oder Zertifikat, Berechtigungen (Scopes)

Wofür ADFS in der Praxis eingesetzt wird

In den Projekten, die ich in den letzten Jahren gesehen habe, tauchen immer wieder dieselben fünf Einsatzszenarien auf. Kaum eine Umgebung nutzt nur eines davon; meistens hat sich die Farm über die Jahre vom reinen Microsoft-365-Anmeldedienst zur zentralen Anmeldedrehscheibe für Dutzende Anwendungen entwickelt – häufig ohne dass das jemand bewusst entschieden hätte.

Szenario	Worum es geht	Heutige Einordnung
Microsoft 365 föderiert	Anmeldung an Exchange Online, SharePoint, Teams mit lokaler AD-Prüfung	Technisch weiter möglich, aber Microsoft empfiehlt Cloud-Authentifizierung (PHS, optional PTA)
Partnerzugang (B2B)	Partner melden sich mit eigenen Firmenkonten über einen Claims Provider Trust an	Für Microsoft-365-Inhalte ist Entra B2B meist einfacher; für eigene Portale bleibt ADFS ein Weg
Hybride Anwendungen	Workloads in Azure oder anderen Clouds delegieren die Anmeldung an den lokalen ADFS	Sinnvoll, solange die Identität lokal führt; mittelfristig Kandidat für Entra ID
Externer Zugriff per WAP	Interne Webanwendungen über den Web Application Proxy mit Vorauthentifizierung veröffentlichen	Funktioniert; Entra-Anwendungsproxy oder ein moderner Reverse Proxy sind Alternativen
Ältere Anwendungen per SAML	Bestandssoftware mit SAML- oder WS-Fed-Unterstützung erhält SSO	Klassischer Grund, ADFS zu behalten – oder die App als Enterprise App nach Entra umzuziehen

Tipp

Bevor du irgendetwas planst, exportiere eine Liste aller Relying Party Trusts mit Get-AdfsRelyingPartyTrust und frag bei jedem Eintrag: Wer ist verantwortlich, wird die Anwendung noch genutzt, und wann hat sie zuletzt ein Token bekommen? In einer typischen Bestandsfarm sind zwanzig bis dreißig Prozent der Trusts Leichen. Jede Leiche ist eine Angriffsfläche und ein Grund mehr, warum eine Migration länger dauert als nötig.

Claims, Regeln und Protokolle

Claims, Ausstellungsregeln und Transformationsregeln

Ein Claim ist eine Aussage über den Benutzer, die der Aussteller verbürgt: „Diese Person hat den UPN anna@contoso.com“, „sie ist Mitglied der Gruppe Vertrieb“, „sie hat sich mit zweitem Faktor angemeldet“. Jeder Claim besteht aus einem Typ – eine standardisierte URI, die auf einen Namen wie …/claims/upn endet – und einem Wert. Die Anwendung entscheidet anhand dieser Claims, wer sie ist und was sie darf.

ADFS verarbeitet Claims in einer festen Pipeline mit drei Stufen. Die Akzeptanzregeln am Claims Provider Trust legen fest, welche eingehenden Claims überhaupt in die Verarbeitung gelangen. Die Ausstellungsregeln am Relying Party Trust bestimmen, welche Claims die konkrete Anwendung

bekommt. Wer genau ist, ergänzt noch die Autorisierung über Zugriffssteuerungsrichtlinien – früher als eigene Autorisierungsregeln gepflegt, seit ADFS in Windows Server 2016 bevorzugt über Richtlinienvorlagen. Innerhalb dieser Stufen gibt es Regeln, die Attribute aus dem AD lesen und als Claim ausstellen, und Transformationsregeln, die einen vorhandenen Claim umschreiben: aus einer Gruppe wird eine Rolle, aus dem UPN wird eine NameID im Format, das die Anwendung erwartet, aus einer SID wird ein lesbarer Gruppenname.

Die Regeln werden in der Claim Rule Language formuliert. Die grafischen Vorlagen erzeugen genau diesen Text; spätestens bei der Fehlersuche wirst du ihn lesen müssen. Ein typisches Beispiel, das zuerst den UPN aus dem AD holt und ihn dann als NameID im E-Mail-Format ausgibt (die Claim-Typen sind hier verkürzt dargestellt, in der echten Regel stehen die vollständigen URIs):

```
@RuleName = "c_AppCRM_UPN_aus_AD"
c:[Type == "…/claims/windowsaccountname",
   Issuer == "AD AUTHORITY"]
 => add(store = "Active Directory",
        types = ("…/claims/upn"),
        query = ";userPrincipalName;{0}", param = c.Value);

@RuleName = "c_AppCRM_NameID_Transform"
c:[Type == "…/claims/upn"]
 => issue(Type = "…/claims/nameidentifier",
          Value = c.Value,
          Properties["…/claimproperties/format"]
            = "urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress");
```

Beachte den Unterschied zwischen add und issue: add legt den Claim nur in den internen Arbeitsspeicher der Pipeline, damit spätere Regeln ihn verwenden können. Erst issue schreibt ihn tatsächlich ins Token. Wer das verwechselt, sucht stundenlang nach einem Claim, der intern wunderbar existiert und trotzdem nie bei der Anwendung ankommt. Und weil Regeln strikt der Reihe nach abgearbeitet werden, entscheidet auch die Reihenfolge: Eine Transformation, die vor der Regel steht, die ihren Eingangs-Claim erst erzeugt, greift ins Leere.

Tipp

Gib jeder selbst geschriebenen Regel einen sprechenden @RuleName mit Präfix für die Anwendung und den Zweck, etwa c_AppCRM_Rollenmapping. Exportiere die Regelsätze aller Relying Party Trusts regelmäßig als Text in eine Versionsverwaltung. Dann kannst du nach jeder Änderung per Diff sehen, was sich geändert hat – und wer es war.

Welche Attribute du ausstellen solltest

Die häufigste Grundsatzfrage bei jedem neuen Relying Party Trust lautet: Woran erkennt die Anwendung den Benutzer eindeutig? Die Antwort hat Folgen, die oft erst Jahre später sichtbar werden – nämlich dann, wenn eine Kollegin heiratet und ihren Namen ändert, oder wenn du die Anwendung später zu Entra ID umziehen willst.

Attribut	Stärken	Schwächen	Empfehlung
UPN	Eindeutig im Forest, Standard für Microsoft-Dienste	Ändert sich bei Namensänderung oder Domänenumbau	Guter Standard, wenn UPN = E-Mail und Änderungen sauber gesteuert sind
E-Mail (mail)	Für Fachanwendungen leicht verständlich	Nicht zwingend eindeutig, kann leer sein, ändert sich	Nur als Anzeige- oder Zusatz-Claim, nicht als alleiniger Schlüssel

objectGUID / ms-DS-ConsistencyGuid	Unveränderlich über die Lebensdauer des Kontos	Für Menschen unlesbar, Anwendung muss damit umgehen	Ideal als NameID für Anwendungen mit eigener Benutzerdatenbank
Gruppen (tokenGroups)	Rollensteuerung ohne Pflege in der Anwendung	Große Token, interne Namen werden preisgegeben	Nur gefilterte, anwendungsbezogene Gruppen ausstellen, nie alle
sAMAccountName	Kompatibel mit Altanwendungen	Nur in der Domäne eindeutig, kein Domänenanteil	Nur, wenn die Anwendung nichts anderes kann

Warnung

Stelle niemals pauschal alle Gruppenmitgliedschaften in ein Token. Ein Benutzer mit 300 Gruppen erzeugt ein Token, das manche Anwendungen oder Proxys wegen der Headergröße schlicht ablehnen – und du verrätst einer externen SaaS-Anwendung nebenbei die halbe Struktur deines AD, inklusive Gruppennamen wie „Projekt Übernahme Müller GmbH – vertraulich“.

Typische Fehlerbilder bei falschen oder fehlenden Claims

Claim-Fehler sind tückisch, weil der ADFS aus seiner Sicht alles richtig macht: Er stellt erfolgreich ein Token aus und protokolliert das auch so. Die Fehlermeldung kommt von der Anwendung, und die ist oft bemerkenswert wortkarg. Mit dieser Übersicht findest du die üblichen Verdächtigen:

Symptom	Häufige Ursache	Prüfung und Lösung
Anwendung meldet „Benutzer unbekannt“	NameID im falschen Format oder mit falschem Attribut	Token per Browser-Entwicklerwerkzeug oder SAML-Tracer ansehen, NameID-Format mit der Anwendung abgleichen
Anmeldung klappt, aber keine Berechtigungen	Gruppen-Claim fehlt oder Rollenwert stimmt nicht exakt	Groß- und Kleinschreibung, Präfixe, Domänenanteil prüfen; Regelreihenfolge kontrollieren
Claim intern vorhanden, im Token nicht	add statt issue verwendet	Letzte Regel muss issue sein
Fehler nur bei einzelnen Benutzern	Attribut im AD leer (z. B. mail nicht gepflegt)	Pflichtattribute per Skript prüfen; ADFS stellt leere Werte nicht aus
Anwendung lehnt Token wegen Signatur ab	Anwendung kennt das neue Token-Signing-Zertifikat noch nicht	Metadaten in der Anwendung aktualisieren oder neues Zertifikat manuell hinterlegen
Fehler nach dem Ablauf eines Zertifikats der Anwendung	Verschlüsselungszertifikat im Relying Party Trust abgelaufen	Metadaten-Monitoring aktivieren oder neues Zertifikat im Trust hinterlegen

Protokolle im Vergleich: SAML, WS-Federation, WS-Trust, OAuth und OpenID Connect

ADFS spricht mehrere Protokolle, und die Wahl ist selten eine Geschmacksfrage. Meistens entscheidet die Anwendung, was sie kann. Trotzdem lohnt es sich zu verstehen, wofür welches Protokoll gedacht ist, denn davon hängen Sicherheitsfragen ab – vor allem bei WS-Trust.

Protokoll	Tokenformat	Typischer Client	Einsatzzweck	Hinweise
SAML 2.0	XML (SAML-Assertion)	Browser	SSO für SaaS- und Partneranwendungen	Am weitesten verbreitet; Artefakt-Bindung nur mit SQL-Datenbank
WS-Federation	XML (SAML 1.1/2.0)	Browser	Microsoft-nahe Anwendungen, SharePoint, ältere .NET-Apps	Funktional ähnlich wie SAML, Microsoft-geprägt
WS-Trust	XML (SOAP)	Rich Clients, Dienste	Token-Anforderung ohne Browser	Endpunkte wie usernamemixed nehmen Kennwörter direkt an – beliebtes Ziel für Passwort-Spraying
OAuth 2.0	JWT (Access Token)	Apps, APIs	Autorisierung für Web-APIs	Seit ADFS in Windows Server 2016; PKCE ab Windows Server 2019
OpenID Connect	JWT (ID-Token)	Moderne Web- und Mobil-Apps	Anmeldung auf Basis von OAuth 2.0	Benutzerinformationen über ein angepasstes ID-Token, nicht über den UserInfo-Endpunkt

Für Browser-SSO in klassischen Geschäftsanwendungen ist SAML 2.0 der Normalfall, WS-Federation die Microsoft-Variante davon. WS-Trust brauchst du nur, wenn ein Client ohne Browser ein Token anfordert – historisch etwa ältere Office-Clients oder Dienste mit Legacy-Authentifizierung. Für neue Web-, Mobil- und API-Szenarien führt kein Weg an OAuth 2.0 und OpenID Connect vorbei. ADFS beherrscht beides seit der Version in Windows Server 2016, mit Anwendungsgruppen für Web-Apps, native Clients und Web-APIs, und seit Windows Server 2019 auch PKCE für den Authorization Code Flow, ohne den du heute keine öffentliche Client-Anwendung mehr bauen solltest.

Wichtig

Zwei Dinge bei OpenID Connect überraschen regelmäßig: Der UserInfo-Endpunkt des ADFS liefert ausschließlich den Subject-Claim, auch wenn die Anwendung mehr anfordert. Weitere Attribute bekommt die Anwendung nur über ein angepasstes ID-Token, das du per Ausstellungsregel gestaltest. Die Scopes email und profile werden nicht unterstützt. Außerdem verlangt ADFS seit dem Update vom Oktober 2026 eine exakte Übereinstimmung der redirect_uri gemäß RFC 6749. Bisher genügte eine Präfix-Übereinstimmung; Unterpfade, abweichende Groß- und Kleinschreibung oder Platzhalter-Varianten werden jetzt abgewiesen. Der Fehler zeigt sich im Admin-Protokoll als Ereignis 364 mit MSIS9224 – inklusive der abgelehnten URI und der Client-ID. Die Korrektur erfolgt an der Client-Registrierung per Set-AdfsClient oder Set-AdfsNativeClientApplication.

ADFS oder Microsoft Entra ID: Entscheidung und Hybridmodelle

Die drei Wege der Anmeldung für Microsoft 365

Sobald Microsoft 365 im Spiel ist, ist ADFS nur eine von drei Möglichkeiten, wie lokale Identitäten in die Cloud kommen. Bei der Kennwort-Hash-Synchronisierung (PHS) überträgt Entra Connect einen nochmals gehashten Ableitungswert des AD-Kennworthashs nach Entra ID; die Anmeldung

findet komplett in der Cloud statt. Bei der Passthrough-Authentifizierung (PTA) bleibt die Kennwortprüfung lokal, schlanke Agenten nehmen die Prüfanfragen ausgehend aus der Cloud entgegen und fragen das AD. Bei der Föderation leitet Entra ID den Benutzer an den ADFS weiter, der die komplette Anmeldung übernimmt.

Kriterium	Kennwort-Hash-Sync (PHS)	Passthrough-Auth (PTA)	Föderation mit ADFS
Lokale Infrastruktur	Nur Entra Connect	Entra Connect + mindestens drei Agenten empfohlen	Farm, WAP, Load Balancer, Zertifikate, Datenbank
Abhängigkeit vom lokalen Netz bei der Anmeldung	Keine	Hoch – ohne Agenten keine Anmeldung	Hoch – ohne Farm keine Anmeldung
Conditional Access, Identity Protection	Vollständig, inkl. Erkennung kompromittierter Kennwörter	Vollständig, ohne Leaked-Credential-Abgleich, falls PHS fehlt	Möglich, aber Risikoerkennung teilweise eingeschränkt; Logik verteilt auf zwei Systeme
Lokale Anmelderichtlinien (Anmeldezeiten, Sperren)	Nur mit Verzögerung über Synchronisierung	Sofort wirksam	Sofort wirksam, beliebig erweiterbar
Betriebsaufwand	Gering	Mittel	Hoch
Angriffsfläche	Kennwort-Hash-Ableitung in der Cloud	Agenten als Tier-0-Systeme	Token-Signing-Schlüssel – Stichwort Golden SAML

Microsoft empfiehlt heute klar PHS als primären Weg, gern kombiniert mit PTA, wenn Kennwörter zwingend lokal geprüft werden müssen. Selbst dort, wo du dich für Föderation entscheidest, ist PHS als Rückfallebene sinnvoll: Fällt die Farm aus, kannst du die Domäne im Notfall auf verwaltete Authentifizierung umstellen, und deine Belegschaft arbeitet weiter.

Was für den Verbleib bei ADFS spricht – und was dagegen

Die ehrliche Antwort auf die Frage „Brauchen wir ADFS noch?“ lautet in den meisten Mittelstandsumgebungen: nein, aber ihr müsst erst aufräumen. Es gibt dennoch Konstellationen, in denen der Verbleib begründet ist. Dazu gehören Anwendungen, die WS-Trust oder SAML-Artefakt-Auflösung zwingend benötigen, sehr spezielle Claim-Logik, die Entra ID mit Claim-Mapping-Richtlinien nicht nachbilden kann, Drittanbieter-MFA ohne Entra-Integrationspfad sowie Compliance-Vorgaben, die verbieten, dass Kennwort-Hashes oder Authentifizierungsanfragen die eigene Infrastruktur verlassen – wobei Letzteres bei genauer Betrachtung oft eine Annahme und keine echte Vorgabe ist.

Dagegen spricht fast alles andere: Betriebsaufwand für Farm, WAP, Datenbank und Zertifikate, die Abhängigkeit der gesamten Cloud-Anmeldung von deinem Rechenzentrum, die zweigeteilte Richtlinienlogik, die fehlende Integration in moderne Risikoerkennung und nicht zuletzt der Umstand, dass Microsoft keine neuen ADFS-Funktionen mehr entwickelt. Mit Windows Server 2025 kam keine neue Farmverhaltensebene; die höchste bleibt die von Windows Server 2019. ADFS wird gepflegt und abgesichert, aber nicht mehr weiterentwickelt.

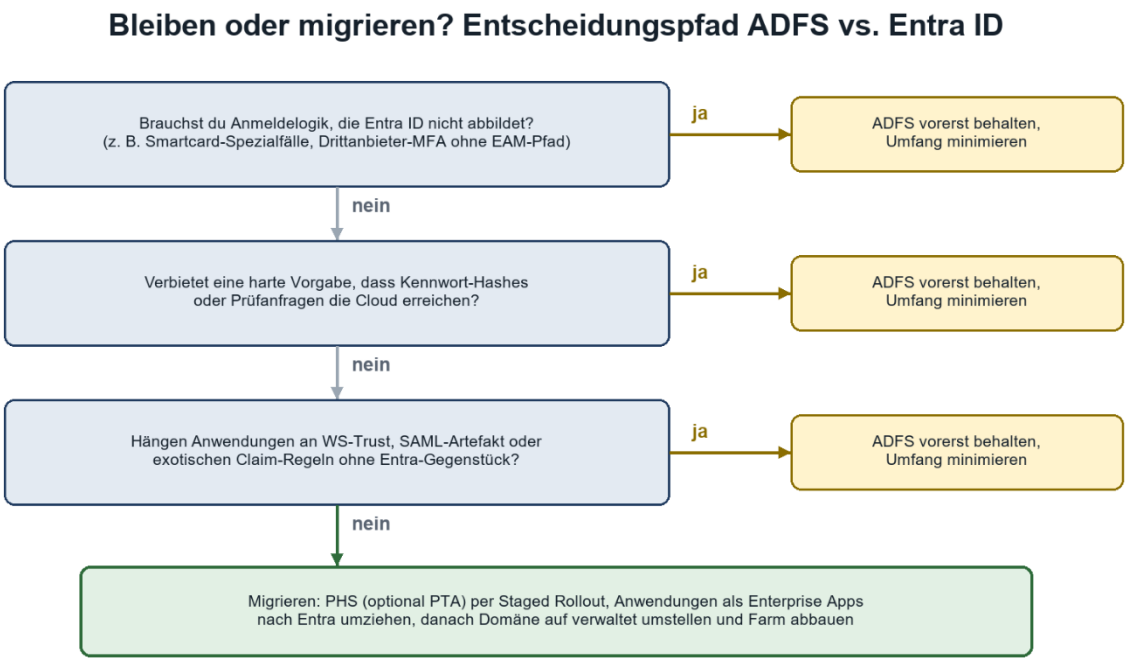


Abbildung 2: Drei Fragen entscheiden, ob ADFS bleibt. Wer alle drei mit Nein beantwortet, sollte die Migration planen.

Warnung

Föderation verlagert das Vertrauen deines gesamten Microsoft-365-Tenants auf einen lokalen Schlüssel. Bei den Angriffen rund um Solorigate haben Angreifer genau diesen Token-Signing-Schlüssel aus ADFS-Servern extrahiert und damit Token für beliebige Benutzer gefälscht – an MFA und Kennwortrichtlinien vorbei. Diese Technik heißt Golden SAML. Ein ADFS-Server ist deshalb exakt so schützenswert wie ein Domänencontroller.

Migration in Etappen statt Big Bang

Der Ausstieg aus ADFS ist kein Wochenendprojekt, aber auch kein Hexenwerk, wenn du ihn in zwei getrennten Strängen denkst. Der erste Strang betrifft die Benutzeranmeldung an Microsoft 365. Hier aktivierst du PHS (falls noch nicht geschehen) und nutzt den gestaffelten Rollout von Entra ID: Einzelne Gruppen werden auf Cloud-Authentifizierung umgestellt, während die Domäne weiterhin föderiert bleibt. Klappt alles, nimmst du die nächste Gruppe dazu; geht etwas schief, nimmst du die Gruppe einfach wieder heraus. Am Ende stellst du die Domäne von föderiert auf verwaltet um.

Der zweite Strang sind die übrigen Relying Party Trusts. Jede Anwendung wird in Entra ID als Unternehmensanwendung neu angelegt, die Claim-Regeln werden in Entra-Claim-Konfigurationen übersetzt, und die Anwendung bekommt die neuen Metadaten. Das kannst du nicht per Gruppe staffeln, das ist pro Anwendung ein Umschalttermin. Der Bericht zur ADFS-Anwendungsaktivität in Entra ID – gespeist über den Connect-Health-Agenten – zeigt dir, welche Trusts sich problemlos migrieren lassen und wo Regeln ohne Gegenstück lauern.

Tipp

Aus der Praxis: Der aufwendigste Teil einer ADFS-Migration ist fast nie die Technik, sondern die Suche nach den Verantwortlichen der Anwendungen. Fang damit an, lange bevor du den ersten Schalter umlegst. Wer für eine Anwendung keinen Verantwortlichen findet, hat einen guten Kandidaten für die Abschaltung gefunden.

Hochverfügbarkeit, Web Application Proxy und Zertifikate

Die Farm und ihre Lastverteilung

Ein einzelner ADFS-Server ist ein Single Point of Failure für jede föderierte Anmeldung. Produktiv betreibst du deshalb immer eine Farm aus mindestens zwei Knoten hinter einem Load Balancer, der den gemeinsamen Dienstnamen – etwa sts.contoso.com – auf die Knoten verteilt. Alle Knoten teilen sich dieselbe Konfiguration, dasselbe Dienstkonto und dieselben Zertifikate.

Eine Frage, die in fast jedem Projekt auftaucht: Braucht der Load Balancer Sticky Sessions? Die Antwort lautet nein. ADFS ist für die Anmeldung zustandslos, die Sitzungsinformationen stecken in Cookies beim Client, die jeder Knoten lesen kann. Microsoft rät sogar ausdrücklich von IP-Affinität ab, weil sie in bestimmten Exchange-Online-Szenarien die Last ungleich auf wenige Knoten verteilt. Viel wichtiger sind andere Punkte: Der Load Balancer darf TLS nicht aufbrechen und neu aufbauen, weil sonst Zertifikatsauthentifizierung, Proxy-Trust und Extended Protection scheitern. Er muss SNI unterstützen, die echte Client-IP erhalten oder im X-Forwarded-For-Header mitgeben – sonst arbeiten Extranet Smart Lockout und gesperrte IP-Adressen mit falschen Daten – und er sollte die Integrität der Knoten über den HTTP-Endpunkt /adfs/probe prüfen, der mit 200 OK antwortet, solange der Dienst läuft.

Windows Internal Database oder SQL Server

Die Konfiguration der Farm liegt in einer Datenbank. Du hast die Wahl zwischen der Windows Internal Database, die ADFS selbst mitbringt, und einem SQL Server. Bei WID ist der erste Knoten der primäre Server; nur er kann Konfigurationsänderungen schreiben, die übrigen ziehen sich die Daten regelmäßig per Abgleich. Fällt der primäre Knoten aus, melden sich Benutzer weiterhin an, aber du kannst nichts mehr konfigurieren, bis du einen sekundären Knoten per Set-AdfsSyncProperties zum neuen primären machst. Bei SQL Server sind alle Knoten gleichberechtigt schreibend, und die Hochverfügbarkeit verlagert sich auf die Datenbankebene – mit Failover-Cluster oder Always-On-Verfügbarkeitsgruppen.

Kriterium	Windows Internal Database (WID)	SQL Server
Maximale Farmgröße	Bis 30 Knoten bei höchstens 100 Relying Party Trusts; darüber SQL Server	Keine von ADFS festgelegte Grenze
Schreibzugriff	Nur primärer Knoten	Alle Knoten
SAML-Artefakt-Auflösung	Nicht unterstützt	Unterstützt
Token-Replay-Erkennung	Nicht unterstützt	Unterstützt
Hochverfügbarkeit der Datenbank	Implizit über mehrere Kopien	Always-On-Verfügbarkeitsgruppen oder Failover-Cluster
Zusätzliche Lizenzen und Know-how	Keine	SQL-Lizenz, DBA-Betreuung
Geeignet für	Die große Mehrheit der Umgebungen	Große Farmen, viele Trusts, Artefakt- oder Replay-Anforderungen

Fakten

Extranet Smart Lockout (ESL) speichert bekannte Anmeldeorte und Kontoaktivität im ADFSArtifactStore. Rechne mit bis zu 1 GB zusätzlichem Datenbankwachstum pro 100.000 Benutzer

und bei WID mit mindestens 5 GB freiem Platz unter C:\Windows\WID\Data, bevor du ESL aktivierst. Der Arbeitsspeicherbedarf steigt bei bis zu 500.000 Benutzern um bis zu 1 GB.

Meine Faustregel: WID ist für die allermeisten Farmen die bessere Wahl, weil sie keine zusätzliche Abhängigkeit schafft. Eine ADFS-Farm, deren Verfügbarkeit von einem SQL-Cluster abhängt, den ein anderes Team betreut und an einem Dienstagabend für Wartungsarbeiten herunterfährt, ist nicht hochverfügbar, sondern nur kompliziert. SQL Server lohnt sich, wenn du eine der Funktionen brauchst, die WID nicht kann, oder wenn du die Größengrenzen sprengst.

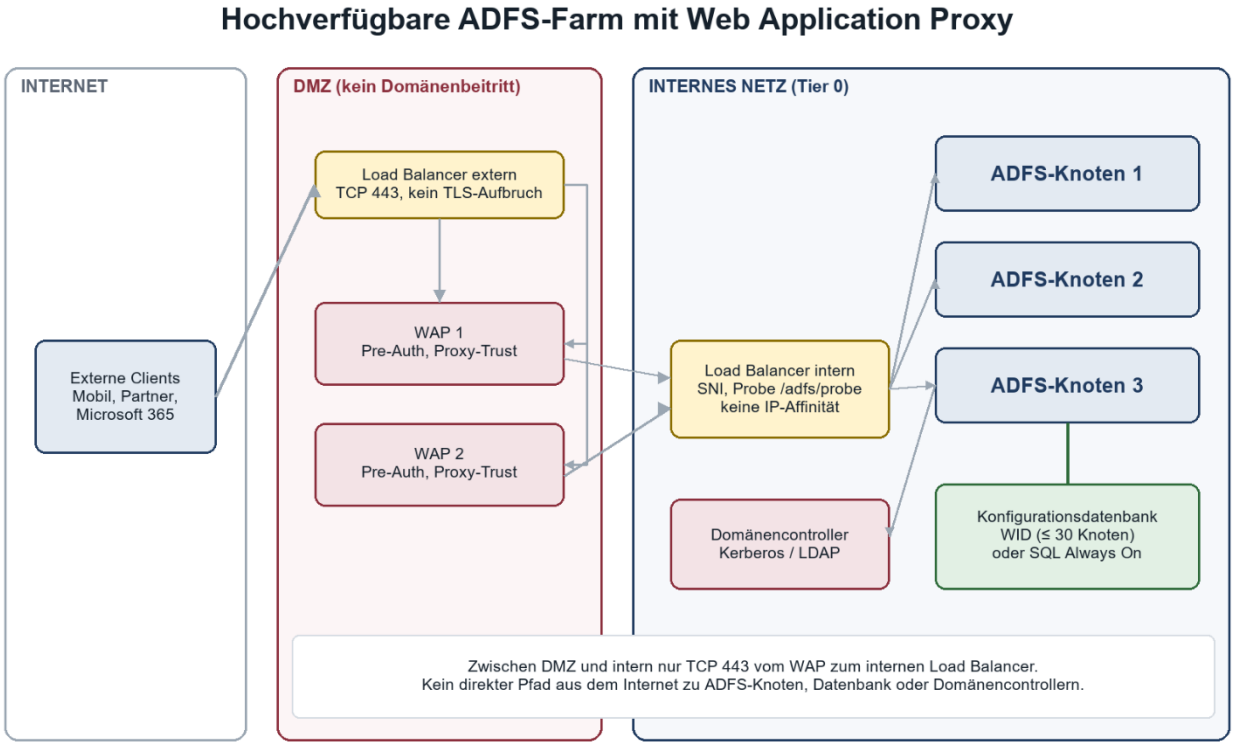


Abbildung 3: Referenzaufbau mit zwei WAP-Knoten in der DMZ, drei ADFS-Knoten intern und je einem Load Balancer ohne TLS-Terminierung.

Geo-Redundanz und Rechenzentrumsausfall

Wer sich gegen den Ausfall eines ganzen Standorts absichern will, verteilt die Farm auf zwei Rechenzentren oder Azure-Regionen. Bei einer WID-Farm gehören dann einfach weitere sekundäre Knoten an den zweiten Standort; die Konfiguration repliziert sich ohnehin. Der Haken: Liegt der primäre Knoten am ausgefallenen Standort, musst du die Rolle umziehen, bevor du wieder Änderungen vornehmen kannst. Bei SQL Server brauchst du eine standortübergreifende Verfügbarkeitsgruppe. Die Verteilung der Anfragen übernimmt ein globaler Load Balancer oder ein DNS-basierter Dienst wie Azure Traffic Manager, der über /adfs/probe erkennt, welcher Standort lebt.

Vergiss dabei nicht die Abhängigkeiten außerhalb von ADFS: Ein ADFS-Knoten ohne erreichbaren Domänencontroller ist nutzlos. Jeder Standort braucht eigene Domänencontroller, eigene WAP-Knoten und einen Internetzugang, über den der Dienstname erreichbar ist. Ausführlicher gehe ich auf die Topologievarianten und ihre Ausfallszenarien im Beitrag Active Directory Federation Services (ADFS): Architektur, Einsatzszenarien und Hochverfügbarkeit ein.

Web Application Proxy in der DMZ

ADFS-Server gehören ins interne Netz und sind Domänenmitglieder. Damit Clients aus dem Internet trotzdem an die Anmeldeseite kommen, stellst du in der DMZ den Web Application Proxy auf – eine Rollenfunktion des Remotezugriffs in Windows Server. Der WAP ist bewusst kein Domänenmitglied. Er authentifiziert sich gegenüber der Farm über ein Proxy-Trust-Zertifikat, das bei der Einrichtung ausgestellt und automatisch erneuert wird, und reicht Anmeldeanfragen nach dem Protokoll MS-ADFSPIP weiter. Drittanbieter-Proxys wie F5 Access Policy Manager oder Citrix ADC können den WAP ersetzen, sofern sie MS-ADFSPIP sauber implementieren.

Zusätzlich kann der WAP interne Webanwendungen veröffentlichen. Dabei gibt es zwei Modi. Beim Pass-Through reicht er die Anfrage unbesehen an die Anwendung durch. Bei der Vorauthentifizierung muss sich der Benutzer zuerst am ADFS anmelden; erst mit gültigem Token lässt der WAP die Anfrage zur Anwendung durch. Unangemeldeter Datenverkehr erreicht das interne System also gar nicht erst – und genau das ist der Grund, warum du für alles, was es kann, die Vorauthentifizierung wählen solltest. Für Anwendungen mit integrierter Windows-Authentifizierung übernimmt der WAP mithilfe der eingeschränkten Kerberos-Delegierung die Anmeldung an der Anwendung; dafür muss er dann doch der Domäne beitreten, was du dir genau überlegen solltest.

- Mehrere WAP-Knoten hinter einem eigenen Load Balancer betreiben, damit der Proxy nicht zum neuen Single Point of Failure wird.
- Zwischen DMZ und internem Netz ausschließlich TCP 443 vom WAP zum internen ADFS-Dienstnamen freigeben – kein RDP, kein SMB, kein LDAP.
- Der WAP muss den Dienstnamen auf die interne Farm auflösen, typischerweise über einen Eintrag in der hosts-Datei oder ein eigenes DNS.
- Bei aktivierter Extended Protection oder integrierter Windows-Authentifizierung muss das TLS-Zertifikat des WAP denselben Schlüssel wie das der ADFS-Server verwenden.
- Keine TLS-Terminierung vor dem WAP – sonst sieht ADFS nur noch die IP der Firewall, und ESL wird blind.

Zur Versionskompatibilität: Sauber ist es, wenn WAP und ADFS auf derselben Windows-Server-Version laufen. Microsoft unterstützt zwar zeitweise gemischte Kombinationen, etwa einen neueren WAP vor einer älteren Farm, aber ausdrücklich nur als Übergang während einer Migration, und neue Funktionen der jeweils höheren Version stehen dann nicht zur Verfügung. Nach einem Upgrade der Farmverhaltensebene auf Windows Server 2019 zeigt die WAP-Konfiguration übrigens weiterhin Windows Server 2016 an – das ist kein Fehler, sondern Absicht, weil der WAP keine 2019-spezifischen Funktionen hat.

Zertifikate: drei Sorten, drei Risiken

Wenn ich in einem Health-Check nur zehn Minuten Zeit hätte, würde ich sie mit den Zertifikaten verbringen. Abgelaufene Zertifikate sind mit großem Abstand die häufigste Ursache für ADFS-Ausfälle – und die peinlichste, weil man sie ein Jahr im Voraus kommen sieht.

Zertifikat	Zweck	Herkunft und Laufzeit	Erneuerung
SSL-/Dienstkommunikationszertifikat	TLS für den Dienstnamen auf ADFS und WAP	Öffentliche CA, Laufzeit nach CA-Richtlinie (heute meist deutlich unter 13 Monaten, Tendenz fallend)	Manuell: Set-AdfsSslCertificate auf jedem ADFS-Knoten, Set-WebApplicationProxySslCertificate auf jedem WAP

Token-Signing	Signiert alle ausgestellten Token	Standardmäßig selbstsigniert, 1 Jahr	Automatisch per AutoCertificateRollover; Partner müssen das neue Zertifikat kennen
Token-Decrypting	Entschlüsselt an ADFS gerichtete verschlüsselte Token	Standardmäßig selbstsigniert, 1 Jahr	Automatisch, analog zum Token-Signing
Proxy-Trust (WAP)	Authentifiziert den WAP an der Farm	Von ADFS ausgestellt	Automatisch, solange der WAP regelmäßig Kontakt zur Farm hat

Der automatische Rollover funktioniert so: 20 Tage vor Ablauf erzeugt ADFS ein neues Token-Signing- und Token-Decrypting-Zertifikat und veröffentlicht es als sekundäres Zertifikat in den Föderationsmetadaten. Fünf Tage später wird es primär, das alte bleibt bis zu seinem Ablauf sekundär. Gesteuert wird das über die Eigenschaften CertificateGenerationThreshold und CertificatePromotionThreshold. Jede Relying Party, die die Metadaten regelmäßig abruft – Entra ID tut das automatisch, sofern die Metadaten aus dem Internet erreichbar sind –, bekommt den Wechsel mit. Alle anderen nicht.

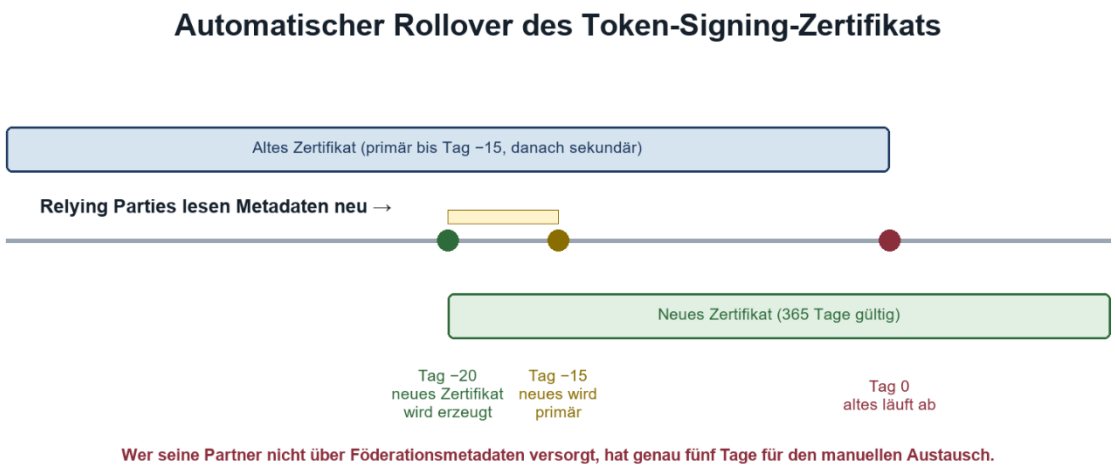


Abbildung 4: Zeitachse des automatischen Rollovers. Zwischen Erzeugung und Primärschaltung liegen standardmäßig fünf Tage.

Warnung

Klassiker aus der Praxis: Ein Partnerportal bekommt das Token-Signing-Zertifikat einmalig per E-Mail und liest keine Metadaten. Ein Jahr später rollt ADFS automatisch über, und am Tag der Primärschaltung können sich plötzlich alle Benutzer dieses Portals nicht mehr anmelden. Die Lösung ist nicht, den Rollover abzuschalten, sondern für jeden Relying Party Trust zu dokumentieren, wie er das neue Zertifikat bekommt – und die Verantwortlichen 20 Tage vorher aktiv zu informieren. ADFS warnt übrigens selbst: Ereignis 385 im Admin-Protokoll meldet bald ablaufende Zertifikate.

Für den Notfall, also wenn ein Token-Signing-Zertifikat vorzeitig ersetzt werden muss, weil du eine Kompromittierung vermutest, gibt es Update-AdfsCertificate -CertificateType Token-Signing -Urgent. Damit wird das neue Zertifikat sofort primär – mit der Folge, dass jede Relying Party, die es noch nicht kennt, sofort ausfällt. Plane deshalb vorher, wie du alle Partner innerhalb weniger Stunden versorgst. Bei Microsoft 365 aktualisierst du die Föderationseinstellungen der Domäne danach gezielt, damit Entra ID nicht auf den nächsten Metadatenabruf warten muss. Wer die

Kompromittierung ernst nimmt, rollt das Zertifikat übrigens zweimal hintereinander, damit auch das zwischenzeitlich sekundäre Zertifikat aus dem Verkehr ist.

Tipp

Statt selbstsignierter Token-Signing-Zertifikate kannst du Zertifikate einer eigenen PKI mit Schlüssel in einem Hardware-Sicherheitsmodul verwenden. Dann ist der private Schlüssel nicht mehr exportierbar, und Golden SAML wird erheblich schwieriger. Der Preis: Der automatische Rollover entfällt, und du musst den Wechsel selbst durchziehen. Für Umgebungen mit hohem Schutzbedarf ist das trotzdem die richtige Wahl.

Sicherheitshärtung, Monitoring und Betrieb

TLS und Extended Protection

Auf ADFS- und WAP-Servern gilt mindestens TLS 1.2, wo möglich TLS 1.3, mit Cipher-Suites, die Perfect Forward Secrecy bieten. Windows Server 2025 hat TLS 1.0 und 1.1 standardmäßig deaktiviert; auf älteren Versionen schaltest du sie selbst über die SCHANNEL-Einstellungen ab und prüfst vorher, ob noch ein Altgerät daran hängt. Apple-Clients verlangen über App Transport Security ohnehin TLS 1.2 mit Forward Secrecy. Achte zusätzlich darauf, dass ADFS und WAP die vollständige Zertifikatskette ausliefern – Android-Clients laden fehlende Zwischenzertifikate nicht nach und brechen die Anmeldung sonst ab.

Extended Protection for Authentication bindet die Windows-Authentifizierung an den TLS-Kanal und verhindert so, dass ein Angreifer Anmeldedaten über eine Man-in-the-Middle-Position weiterleitet. In ADFS steuert die Eigenschaft ExtendedProtectionTokenCheck dieses Verhalten. Sie steht standardmäßig auf Allow; empfehlenswert ist Require, sofern deine Clients und Proxys es vertragen. Teste das vorher mit allen Browsern und mit deinem WAP, denn Proxys, die TLS aufbrechen, sind mit Extended Protection unvereinbar – noch ein Grund, auf TLS-Terminierung zu verzichten.

```
Set-AdfsProperties -ExtendedProtectionTokenCheck Require
Set-AdfsProperties -ExtranetLockoutEnabled $true -ExtranetLockoutMode AdfsSmartLockoutEnforce `
    -ExtranetLockoutThreshold 5 -ExtranetObservationWindow (New-TimeSpan -Minutes 30)
Set-AdfsProperties -AuditLevel Verbose
```

MFA, Smart Lockout und Schutz gegen Passwort-Spraying

Die Anmeldeseite deines ADFS hängt im Internet, und das bleibt nicht unbemerkt. Passwort-Spraying – wenige häufige Kennwörter gegen sehr viele Konten – gehört zum Grundrauschen jeder öffentlichen ADFS-Instanz. Dagegen helfen mehrere Ebenen, die du kombinieren solltest:

Maßnahme	Wirkung	Hinweise
Extranet Smart Lockout	Sperrt Konten nur für Anmeldungen von unbekannten Orten, vertraute Orte bleiben nutzbar	Schwellwert unterhalb der AD-Kontosperrung setzen, damit das AD-Konto nicht gesperrt wird; erst im Log-Only-Modus beobachten
Gesperrte IP-Adressen	Bekannte Angreifer-IPs werden abgewiesen	Pflege per Set-AdfsProperties -BannedIpList; setzt korrekte Client-IP voraus
Multi-Faktor-Authentifizierung	Ein erratenes Kennwort allein reicht nicht	Über Entra-MFA-Adapter oder Drittanbieter; per Zugriffsrichtlinie mindestens extern erzwingen
Legacy-Endpunkte	Kennwortannehmende WS-Trust-	Endpunkte wie

schließen	Endpunkte sind kein Einfallstor mehr	/adfs/services/trust/2005/usernamemixed am Proxy deaktivieren, wenn kein Client sie braucht
Kennwortlose Verfahren	Kein Kennwort, nichts zu sprayen	Windows Hello for Business, Zertifikatsanmeldung; als primäre Authentifizierung konfigurierbar

Warnung

MFA schützt die Anmeldung, nicht das Token. Ein einmal gestohlenes SSO-Cookie oder Token funktioniert bis zu seinem Ablauf ohne zweiten Faktor. Halte deshalb die Lebensdauer kurz: Für nicht registrierte Geräte gelten standardmäßig acht Stunden SSO, mit „Angemeldet bleiben“ 24 Stunden; Access Token sind eine Stunde gültig. Prüfe, ob „Angemeldet bleiben“ bei dir wirklich nötig ist, und aktiviere bei SQL-Farmen die Token-Replay-Erkennung.

Dienstkonto und Tier-0-Schutz

Der ADFS-Dienst läuft heute idealerweise unter einem gruppenverwalteten Dienstkonto (gMSA). Dessen Kennwort wird automatisch rotiert und ist für keinen Menschen sichtbar – das beendet die unrühmliche Tradition, das ADFS-Dienstkonto mit einem Kennwort aus dem Jahr 2014 und der Option „Kennwort läuft nie ab“ zu betreiben. Das Dienstkonto braucht keine Adminrechte in der Domäne; gib ihm auch keine.

Die Schlüssel der selbstsignierten Zertifikate werden mit einem Schlüssel aus dem Distributed-Key-Manager-Container im AD verschlüsselt. Wer Lesezugriff auf diesen Container und Zugriff auf die ADFS-Konfigurationsdatenbank hat, kann den Token-Signing-Schlüssel rekonstruieren. Deshalb gehören ADFS-Server, ihr Dienstkonto, der DKM-Container und die Administratorenkonten in dieselbe Schutzebene wie die Domänencontroller: Tier 0, administriert nur von privilegierten Zugriffsarbeitsstationen, keine Agenten von Drittanbietern ohne Not, keine Mitnutzung für andere Dienste, keine lokalen Administratoren aus dem Helpdesk.

Ereignisprotokolle, Audits und SIEM

ADFS protokolliert in zwei Welten. Das Admin-Protokoll unter Anwendungen und Dienste › AD FS › Admin enthält Betriebs- und Fehlerereignisse. Die eigentlichen Audit-Ereignisse landen im Sicherheitsprotokoll – allerdings nur, wenn du die Überwachungsrichtlinie „Anwendung generiert“ für Erfolg und Fehler aktivierst und das Dienstkonto per Gruppenrichtlinie das Recht „Generieren von Sicherheitsüberwachungen“ besitzt. Ohne diese zwei Schritte bleibt das Sicherheitsprotokoll stumm, und du merkst es erst bei der ersten forensischen Untersuchung.

Ereignis-ID	Protokoll	Bedeutung	Wozu du es nutzt
1200	Sicherheit	Token erfolgreich für eine Anwendung ausgestellt	Nutzungsstatistik je Relying Party, Erkennung toter Trusts
1201	Sicherheit	Token-Ausstellung fehlgeschlagen	Fehlkonfigurierte Trusts, Zugriffsrichtlinie greift
1202	Sicherheit	Neue Anmeldedaten erfolgreich geprüft	Anmeldeorte und Zeiten pro Benutzer
1203	Sicherheit	Prüfung neuer Anmeldedaten fehlgeschlagen	Passwort-Spraying und Brute Force erkennen
364	Admin	Fehler bei einer Anforderung	Erste Anlaufstelle bei Anmeldefehlern,

			auch MSIS9224 bei redirect_uri
385	Admin	Zertifikat läuft bald ab	Alarm 20 Tage vor dem Ablauf
411	Sicherheit	Token-Validierung fehlgeschlagen	Hinweis auf manipulierte oder fehlerhafte Token und Kennwortfehler

Diese Ereignisse gehören ins zentrale SIEM, etwa Microsoft Sentinel, oder in ein anderes System, das Korrelationen kann. Die interessanten Muster sind selten einzelne Ereignisse, sondern Häufungen: viele 1203 für viele verschiedene Konten von wenigen IP-Adressen (Spraying), viele 1203 für ein Konto (Brute Force), 1200 für eine Relying Party, die seit Monaten keiner benutzt hat, oder Token für privilegierte Konten außerhalb der üblichen Zeiten. Für die Verfügbarkeit taugen das ADFS-Management-Pack in SCOM oder Microsoft Entra Connect Health für AD FS, der zusätzlich einen Bericht über riskante IP-Adressen liefert. Golden-SAML-Angriffe erkennst du übrigens gerade nicht im ADFS-Protokoll – gefälschte Token tauchen dort nie auf. Hier hilft nur der Abgleich: Anmeldungen in Entra ID ohne passendes 1200-Ereignis auf deiner Farm sind ein Alarmsignal.

Tipp

Das Microsoft-PowerShell-Modul ADFS Diagnostics mit Test-AdfsServerHealth prüft in wenigen Sekunden Zertifikate, Dienstzustand, Endpunkte und Konfiguration. Lass es täglich per geplanter Aufgabe laufen und schick das Ergebnis ins Monitoring. Das kostet dich eine halbe Stunde Einrichtung und erspart dir mindestens einen Montagmorgen im Jahr.

Notfallplanung, Disaster Recovery und Tests

Ein Notfallplan, den niemand je ausprobiert hat, ist kein Plan, sondern ein Gebet. Für ADFS heißt das: Sichere die Konfiguration regelmäßig mit dem AD FS Rapid Restore Tool von Microsoft, das Konfigurationsdatenbank, Zertifikate samt privaten Schlüsseln, Dienstkontoinformationen und Anpassungen der Anmeldeseite in einem verschlüsselten Paket ablegt. Bewahre das Paket getrennt von der Farm auf und schütze es wie einen Tier-0-Schlüssel – denn genau das ist es.

Test	Empfohlener Rhythmus	Was du prüfst
Knotenausfall	Vierteljährlich	Load Balancer nimmt den Knoten heraus, Anmeldungen laufen weiter
Primärwechsel bei WID	Halbjährlich	Set-AdfsSyncProperties auf sekundärem Knoten, Konfiguration wieder schreibbar
Standortausfall	Jährlich	Traffic Manager oder DNS schwenkt, Domänencontroller und WAP am Zweitstandort tragen
Wiederherstellung aus Sicherung	Jährlich in Testumgebung	Rapid Restore auf frischem Server, Trusts und Zertifikate vollständig
Notfall-Umschaltung Microsoft 365	Einmal geprobt, dokumentiert	Domäne per PHS auf verwaltet umschalten, Rückweg ebenfalls
Zertifikatswechsel unter Zeitdruck	Jährlich als Planspiel	Wer informiert welche Partner, wie schnell sind alle Relying Parties versorgt

Wichtig

Patch die Farm regelmäßig und behalte die Release Notes der monatlichen Updates im Blick. Das

Update vom Oktober 2026 zur exakten redirect_uri-Prüfung zeigt, dass auch ein reiner Sicherheitspatch Anmeldungen brechen kann. Spiel Updates erst auf einem Knoten ein, nimm ihn aus dem Load Balancer, teste die wichtigsten Anwendungen und rolle dann erst auf die übrigen Knoten aus.

Fazit

Active Directory Federation Services ADFS ist ausgereift, funktional vollständig und für bestimmte Szenarien weiterhin die richtige Antwort. Es ist aber auch ein System, das keine neuen Funktionen mehr bekommt, das eine eigene Infrastruktur mit Farm, Proxys, Datenbank und Zertifikaten verlangt und dessen Signaturschlüssel im Ernstfall das Vertrauen deines gesamten Cloud-Tenants trägt. Wer ADFS betreibt, betreibt Tier-0-Infrastruktur und muss sie auch so behandeln: hochverfügbar ohne unnötige Abhängigkeiten, gehärtet gegen Spraying und Schlüsseldiebstahl, mit überwachten Zertifikaten, vollständigen Audit-Logs und einem Notfallplan, der schon einmal geprobt wurde.

Für die meisten Organisationen ist der klügste nächste Schritt, den Umfang der Farm Schritt für Schritt zu verkleinern: Microsoft 365 per gestaffeltem Rollout auf Cloud-Authentifizierung umstellen, tote Trusts löschen, migrationsfähige Anwendungen nach Entra ID umziehen. Was danach übrig bleibt, betreibst du bewusst und mit voller Sorgfalt – oder schaltest es irgendwann ganz ab. Beides ist besser als der Zustand, den ich in Health-Checks am häufigsten sehe: eine Farm, die seit Jahren läuft, weil niemand sich traut, sie anzufassen.

Weiterlesen

› Active Directory Federation Services (ADFS): Architektur, Einsatzszenarien und Hochverfügbarkeit

FAQ: Häufige Fragen zu ADFS

Was ist der Unterschied zwischen ADFS und Microsoft Entra ID?

ADFS ist ein lokaler Identity Provider, den du selbst auf Windows Server betreibst. Er prüft Anmeldungen gegen dein Active Directory und stellt Token für Anwendungen aus. Microsoft Entra ID ist der Identitätsdienst in der Microsoft-Cloud, der dieselbe Aufgabe als Dienst erledigt – ergänzt um Conditional Access, Risikoerkennung und eine große Anwendungsgalerie. In Hybridumgebungen arbeiten beide zusammen: Entra ID leitet föderierte Anmeldungen an ADFS weiter, oder es übernimmt die Anmeldung per PHS oder PTA selbst.

Ist ADFS abgekündigt?

Nein. ADFS ist Bestandteil von Windows Server, auch von Windows Server 2025, und wird im Rahmen des Lebenszyklus der jeweiligen Serverversion mit Sicherheitsupdates versorgt. Microsoft entwickelt allerdings keine neuen Funktionen mehr und empfiehlt ausdrücklich, statt eines Upgrades auf die neueste ADFS-Version nach Microsoft Entra ID zu migrieren.

Brauche ich für ADFS zwingend einen Web Application Proxy?

Nur wenn Clients von außerhalb des internen Netzes auf die Anmeldeseite zugreifen sollen – was bei Microsoft 365, mobilen Geräten und Partnern praktisch immer der Fall ist. Ohne veröffentlichten Proxy kann außerdem Entra ID die Föderationsmetadaten nicht abrufen, und du musst neue Token-Signing-Zertifikate manuell eintragen. Statt des WAP sind Drittanbieter-Proxys möglich, sofern sie das Protokoll MS-ADFSPIP unterstützen.

Reicht die Windows Internal Database oder brauche ich SQL Server?

Für die meisten Umgebungen reicht WID: bis zu 30 Farmknoten bei höchstens 100 Relying Party Trusts. SQL Server brauchst du bei mehr Trusts oder größeren Farmen, für SAML-Artefakt-Auflösung, für Token-Replay-Erkennung oder wenn alle Knoten gleichberechtigt Konfigurationsänderungen schreiben sollen. Dann sollte der SQL Server selbst hochverfügbar sein, etwa mit Always-On-Verfügbarkeitsgruppen.

Muss der Load Balancer vor ADFS Sticky Sessions verwenden?

Nein. ADFS ist für Anmeldungen zustandslos, und Microsoft rät sogar von IP-Affinität ab. Wichtig ist, dass der Load Balancer TLS nicht terminiert, SNI unterstützt, die Client-IP durchreicht oder per X-Forwarded-For mitgibt und die Knoten über den Endpunkt /adfs/probe überwacht.

Was passiert, wenn das Token-Signing-Zertifikat abläuft?

Normalerweise nichts, weil ADFS 20 Tage vor Ablauf automatisch ein neues Zertifikat erzeugt und es fünf Tage später primär schaltet. Probleme bekommen alle Relying Parties, die das neue Zertifikat nicht über die Föderationsmetadaten beziehen: Sie lehnen die Token ab, sobald das neue Zertifikat primär ist. Dokumentiere deshalb für jeden Trust, wie er das neue Zertifikat erhält, und überwache Ereignis 385.

Welches Protokoll soll ich für eine neue Anwendung verwenden?

Für eine klassische Geschäftsanwendung mit Browser-Anmeldung ist SAML 2.0 der verbreitetste Weg. Für neue Web-, Mobil- und API-Anwendungen nimmst du OpenID Connect und OAuth 2.0 mit Authorization Code Flow und PKCE. WS-Federation nur, wenn die Anwendung es so verlangt, und WS-Trust nur für Clients ohne Browser, die nichts anderes können. Bei einer neuen Anwendung solltest du allerdings zuerst prüfen, ob du sie nicht gleich an Entra ID anbindest.

Wie schütze ich ADFS gegen Passwort-Spraying?

Aktiviere Extranet Smart Lockout mit einem Schwellwert unterhalb der AD-Kontosperrung, sorge dafür, dass ADFS die echte Client-IP sieht, pflege gesperrte IP-Adressen, erzwinge MFA für externe Anmeldungen, schließe ungenutzte WS-Trust-Endpunkte am Proxy und überwache Ereignis 1203 im SIEM auf Häufungen. Langfristig hilft am besten, Kennwörter durch kennwortlose Verfahren zu ersetzen.

Warum funktionieren OAuth-Anmeldungen nach einem Update plötzlich nicht mehr?

Seit dem Update vom Oktober 2026 prüft ADFS die redirect_uri exakt statt per Präfix. Clients, deren Registrierung nicht exakt mit der tatsächlich gesendeten URI übereinstimmt, werden mit Ereignis 364 und MSIS9224 abgewiesen. Das Ereignis nennt die abgelehnte URI und die Client-ID. Du korrigierst die Registrierung per Set-AdfsClient oder Set-AdfsNativeClientApplication; am Server selbst ist nichts zu ändern.