

Claim Rules nach Conditional Access übersetzen

ADFS · Conditional Access · Entra ID
**Aus Regelsprache werden
Schalter – und ein paar Regeln
bleiben auf der Strecke**
*Netzwerkstandort, Gruppen, Anwendung, Client-Typ; was sich 1:1
übersetzen lässt, was nur mit Umweg geht und was du besser
beendigst.*
boddenberg.de

```

c:[Type ==
  ".../groupsid",
  Value == "S-1-5-..."]
=> issue(Type =
  ".../permit");

exists([Type ==
  ".../x-ms-proxy"])
=> issue(...deny);

```

Richtlinie
Benutzer ☒
Zielressource ☒
Netzwerk ☒
Client-Apps ☐
MFA ☒

WISSEN Grundlagen, Architektur und alle Praxisbeiträge rund um ADFS an einem Ort. > Active Directory Federation Services	BERATUNG Dreißig Relying Parties, jede mit eigener Autorisierungslogik? Wir übersetzen sie mit dir in ein schlankes Richtlinienset. > Consulting zu ADFS	SCHULUNG Claim Rules lesen, Conditional Access bauen, im Nur-Bericht-Modus prüfen – einmal im Labor, bevor es ernst wird. > ADFS-Schulungen
---	---	--

Irgendwo in deiner ADFS-Farm liegt eine Regel, die so aussieht, als hätte sie jemand nachts um drei in einem Anfall von Regex-Begeisterung geschrieben. Sie erlaubt der Gruppe Personal den Zugriff auf das Bewerberportal, aber nur aus dem Firmennetz, außer für drei Leute aus dem Betriebsrat, die von zu Hause dürfen, wenn sie MFA machen, und ActiveSync ist sowieso verboten. Funktioniert seit 2016. Niemand traut sich, sie anzufassen. Und jetzt soll genau diese Anwendung nach Entra ID – mitsamt ihrer Logik. Willkommen bei der Frage, wie du ADFS Claim Rules nach Conditional Access übersetzt.

Die gute Nachricht: Der größte Teil dessen, was in ADFS-Autorisierungs- und MFA-Regeln steht, lässt sich in Conditional Access ziemlich direkt abbilden – Gruppen, Netzwerkstandorte, Anwendungen, Client-Typen. Die weniger gute: Conditional Access ist keine Programmiersprache. Was ADFS mit einem beliebigen Claim und einem regulären Ausdruck erledigt hat, braucht in Entra ID entweder einen Umweg über Daten oder eine ehrliche Entscheidung, die Regel nicht mitzunehmen. Dieser Beitrag liefert dir die Übersetzungstabelle, zeigt die Lücken und wie du sie schließt. Was ADFS grundsätzlich ist und warum es in vielen Unternehmen seit über zehn Jahren zuverlässig läuft, steht auf der Übersichtsseite [Active Directory Federation Services](#). Wie die Claim-Rule-Sprache selbst funktioniert, erklärt der [ADFS Claim Rules Leitfaden](#) – hier setzen wir voraus, dass du eine Regel lesen kannst. Die Entra-Seite mit Lizenzen, Rollen und dem großen Bild von Conditional Access findest du unter [Microsoft Entra ID](#).

FAKTEN — Claim Rules und Conditional Access in Kürze In ADFS steuern drei Regelsätze je Relying Party den Zugriff: Issuance Authorization Rules (oder ab Windows Server 2016 eine Access Control Policy), Additional Authentication Rules für MFA und die Issuance Transform Rules für den Inhalt des Tokens. Nach Conditional Access wandern nur die ersten beiden. Conditional Access wertet nach der primären Anmeldung alle zutreffenden Richtlinien gemeinsam aus. Es gibt keine Reihenfolge; eine blockierende Richtlinie gewinnt immer,

Gewährungsanforderungen addieren sich.

Für „Von einem bestimmten Netzwerk“, „Von bestimmten Gruppen“ und „Von Geräten mit bestimmter Vertrauensstufe“ nennt Microsoft direkte Gegenstücke. Für „Mit bestimmten Claims in der Anfrage“ heißt es in der Migrationsdokumentation wörtlich: lässt sich nicht migrieren.

IP-basierte Named Locations: höchstens 195 Standorte mit je bis zu 2000 IP-Bereichen, nur CIDR-Masken größer als /8.

Conditional Access erfordert eine Entra-ID-P1-Lizenz (in vielen Microsoft-365-Paketen enthalten); risikobasierte Bedingungen benötigen P2.

Zwei Denkmodelle: Regelkette gegen Richtlinienstapel

Bevor du die erste Regel übersetzt, lohnt sich ein Blick auf den grundsätzlichen Unterschied, denn er erklärt die meisten Überraschungen. ADFS arbeitet pro Relying Party eine Kette ab. Eine Anfrage kommt mit Kontext-Claims herein – Client-IP, ob sie über den Web Application Proxy kam, welcher Endpunkt angesprochen wurde, welche Client-Anwendung sich meldet. Dann wird authentifiziert, dann prüfen die Additional Authentication Rules, ob MFA nötig ist, dann entscheiden die Authorization Rules mit permit- und deny-Claims, ob überhaupt ein Token ausgestellt wird. Jede Regel kann auf jeden Claim schauen, Werte per Regex vergleichen und Zwischenergebnisse für spätere Regeln hinterlegen.

Conditional Access denkt anders. Es gibt keine Kette pro Anwendung, sondern einen Stapel von Richtlinien für den ganzen Tenant. Jede Richtlinie beschreibt eine Situation („Wer greift von wo womit auf was zu?“) und eine Folge („blockieren“ oder „gewähren, wenn ...“). Bei jeder Anmeldung sammelt Entra ID alle Richtlinien ein, deren Bedingungen zutreffen, und verlangt die Summe ihrer Anforderungen. Reihenfolge? Gibt es nicht. Ein „else“-Zweig? Auch nicht. Das klingt nach weniger Macht und ist es auch – aber es ist dafür auf einen Blick audittierbar, was man von einer gewachsenen Claim-Rule-Sammlung nicht behaupten kann.

Skizze 1: Die alte Pipeline und die neue Entscheidung

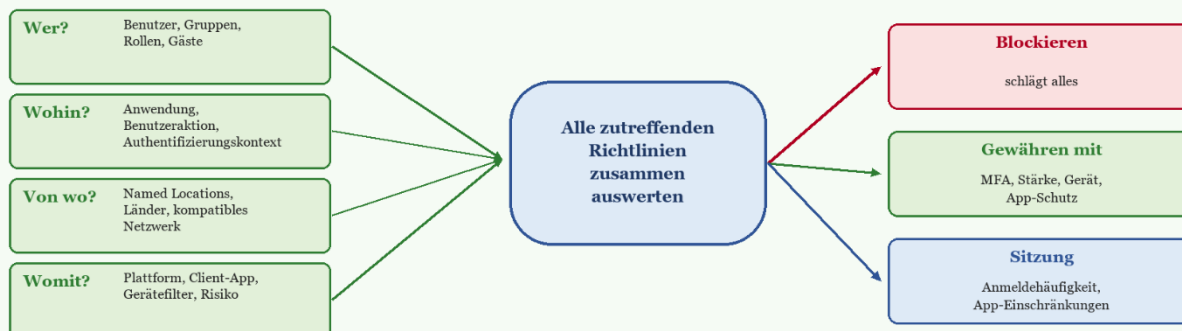
ADFS arbeitet Regeln der Reihe nach ab. Conditional Access sammelt alle passenden Richtlinien und verlangt die Summe.

ADFS: Regelketten pro Relying Party



Reihenfolge zählt · Regex erlaubt · jede Bedingung aus jedem Claim · Fehlerseite der Farm

Entra ID: Conditional Access nach der Anmeldung



Skizze 1: Oben die sequenzielle ADFS-Pipeline je Relying Party, unten die gemeinsame Auswertung aller zutreffenden Conditional-Access-Richtlinien.

Was das für deine Übersetzung bedeutet

Daraus folgen drei Regeln, die dir viel Ärger ersparen. Erstens: Übersetze nicht Regel für Regel, sondern Absicht für Absicht. Schreib zu jeder Relying Party einen Satz wie „Personal darf das Bewerberportal von überall nutzen, aber außerhalb der Standorte nur mit MFA“. Dieser Satz ist dein Bauplan, nicht der Regeltext. Zweitens: Trenne Zuweisung und Bedingung. Wer eine Anwendung überhaupt benutzen darf, regelst du in Entra ID an der Enterprise App über „Zuweisung erforderlich“ und die zugewiesenen Gruppen. Conditional Access kümmert sich um das Wie und Woher. Drittens: Bündle. Zehn Relying Parties mit derselben MFA-außerhalb-des-Netzes-Regel werden zu einer Richtlinie mit zehn Zielanwendungen, nicht zu zehn Richtlinien.

Aspekt	ADFS Claim Rules	Conditional Access
Geltungsbereich	je Relying Party, plus globale MFA-Regeln	tenantweit, Zielressourcen je Richtlinie
Auswertung	sequenziell, Zwischenergebnisse per add()	alle zutreffenden Richtlinien gleichzeitig
Konflikte	deny schlägt permit	Blockieren schlägt Gewähren
Bedingungen	jeder Claim, Regex, NOT EXISTS	festgelegte Signale: Benutzer, Ressource, Netzwerk, Plattform, Client-App, Gerät, Risiko
Zeitpunkt	vor der Ausstellung des Tokens an die RP	nach der primären Authentifizierung in Entra ID
Test	Testfarm oder Mut	Nur-Bericht-Modus und What-If-Werkzeug
Fehlermeldung	Fehlerseite der Farm, anpassbar	Standardmeldung von Entra ID, Branding nur global

Tabelle 1: Die beiden Modelle im direkten Vergleich

TIPP — Erst exportieren, dann denken

Bevor du irgendetwas übersetzt, ziehst du die Autorisierungs- und MFA-Regeln aller Relying Parties als Text heraus (Listing 1). Ein Screenshot der Konsole zeigt dir weder die Reihenfolge noch die Parameter.

Wenn du die Farm noch nicht vollständig kennst, hilft dir der Beitrag [Relying Parties inventarisieren – die Bestandsaufnahme vor der Entra-Migration](#) bei der vollständigen Bestandsaufnahme.

```
# Auf einem ADFS-Server, PowerShell als Administrator
Get-AdfsRelyingPartyTrust |
    Select-Object Name, Enabled, AccessControlPolicyName,
        IssuanceAuthorizationRules, AdditionalAuthenticationRules |
    Export-Clixml -Path C:\Export\rp-zugriffsregeln.xml

# Globale MFA-Regeln der Farm nicht vergessen
Get-AdfsAdditionalAuthenticationRule
Get-AdfsGlobalAuthenticationPolicy | Select-Object PrimaryExtranetAuthenticationProvider,
    PrimaryIntranetAuthenticationProvider, AdditionalAuthenticationProvider

# Welche Access Control Policies gibt es, und welche Parameter haben sie?
Get-AdfsAccessControlPolicy | Select-Object Name, IsBuiltIn, PolicyMetadata
```

Listing 1: Autorisierungs- und MFA-Logik der Farm sichern

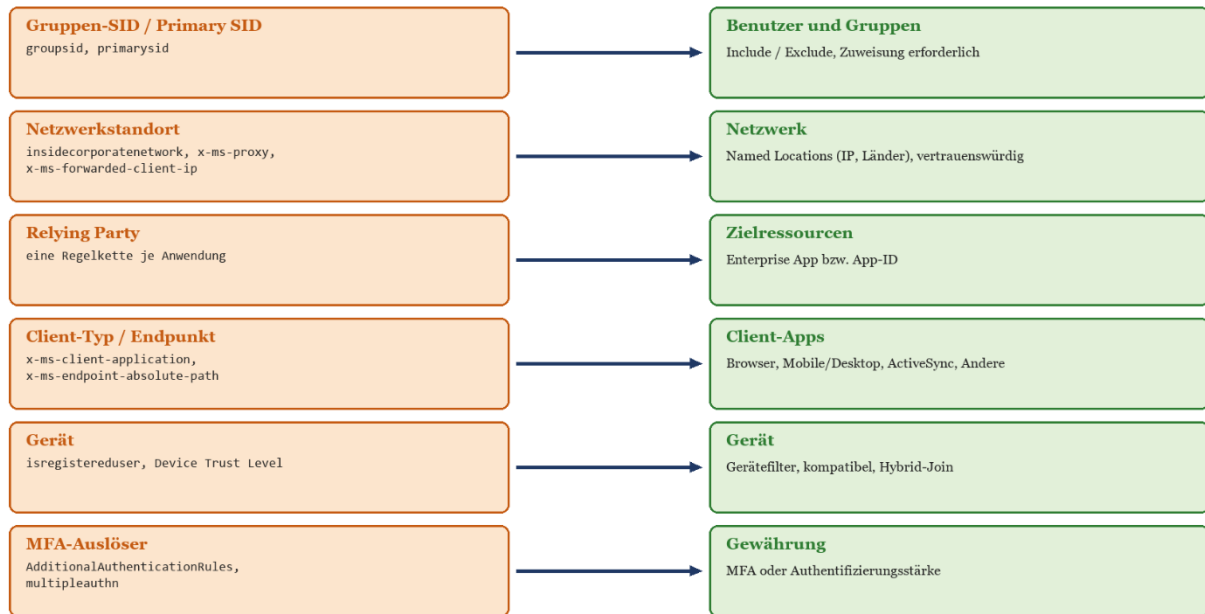
Ein Hinweis zum Export: Relying Parties mit Access Control Policy haben meist leere IssuanceAuthorizationRules – die Logik steckt dann in der Policy und ihren Parametern (ResourceAccessControlPolicyParameters). Wie diese Policies aufgebaut sind, beschreibt der Beitrag [Access Control Policies in ADFS – Zugriffsregeln ohne Claim-Rule-Akrobatik](#). Für die Übersetzung sind sie ein Geschenk, weil sie bereits in derselben Begriffswelt denken wie Conditional Access: Gruppe, Netzwerk, Gerät, MFA.

Die Übersetzungstabelle: Standort, Gruppen, Anwendung, Client-Typ

Jetzt zum Kern. Die folgende Landkarte zeigt, welcher ADFS-Baustein in welchem Conditional-Access-Baustein landet. Danach gehen wir die vier großen Gruppen durch, jeweils mit typischer Claim Rule und ihrem Gegenstück.

Skizze 2: Welcher ADFS-Baustein wird zu welchem CA-Baustein?

Links die Claim-Typen und Optionen, rechts das Ziel in Entra ID



Skizze 2: Links die Claim-Typen und Optionen aus ADFS, rechts ihr Ziel in Entra ID.

ADFS-Regel (typisch)	Conditional Access / Entra ID	Übersetzbar?
Permit für alle Benutzer	„Zuweisung erforderlich“ = Nein, oder Zuweisung an eine Alle-Benutzer-Gruppe	1:1
Permit für Gruppe (groupsid)	Gruppe der Enterprise App zuweisen, „Zuweisung erforderlich“ = Ja	1:1
Permit für einzelnen Benutzer (primarysid)	Benutzer direkt zuweisen (besser: Gruppe)	1:1
Deny für Gruppe	Richtlinie „Zugriff blockieren“ mit dieser Gruppe als Include	1:1
Deny wenn x-ms-proxy existiert (Extranet)	Blockieren für alle Netzwerke, vertrauenswürdige Standorte ausgeschlossen	1:1, Netz muss bekannt sein
MFA wenn insidecorporatenetwork = false	MFA gewähren, Netzwerk: alle, ausgeschlossen: vertrauenswürdig	1:1
MFA für Gruppe	MFA bzw. Authentifizierungsstärke, Include: Gruppe	1:1
MFA für nicht registrierte Geräte	Konformes oder Hybrid-Entra-Gerät oder MFA verlangen	1:1
IP-Bereich per Regex auf x-ms-forwarded-client-ip	IP-basierte Named Location mit CIDR-Bereichen	mit Umweg
Bedingung auf Attributwert (z. B. department)	dynamische Gruppe, dann Gruppe als Bedingung	mit Umweg
Blockieren von ActiveSync / Legacy-Endpunkten	Client-Apps: Exchange ActiveSync und Andere Clients blockieren	1:1

Regel nur für einen Endpunkt (x-ms-endpoint-absolute-path)	Client-App-Bedingung oder Authentifizierungsflüsse	teilweise
Bedingung auf User-Agent (x-ms-client-user-agent)	Geräteplattform; freies Muster nicht möglich	teilweise
Beliebiger Claim in der Anfrage	kein Gegenstück	nein
Werte aus SQL-/LDAP-Attributspeicher als Bedingung	kein Gegenstück; Daten vorher nach Entra bringen	nein

Tabelle 2: Die Übersetzungstabelle für typische Autorisierungs- und MFA-Regeln

Netzwerkstandort: insidecorporatenetwork, x-ms-proxy und die IP

Die meisten Standortregeln in ADFS stützen sich auf eines von drei Signalen: den Claim `insidecorporatenetwork` (false, wenn die Anfrage über den Web Application Proxy kam), die bloße Existenz von `x-ms-proxy` oder einen Regex auf `x-ms-forwarded-client-ip`. Die ersten beiden sagen eigentlich nur „kam über den Proxy“ – ein Signal, das es in Entra ID nicht gibt, weil dort jede Anmeldung aus dem Internet kommt. Die Übersetzung lautet deshalb immer: Wie sehen deine öffentlichen Ausgangs-IP-Adressen aus? Die trägst du als IP-basierte Named Location ein und markierst sie als vertrauenswürdig.

```
# Typische ADFS-Regel: MFA für alle Anfragen von außerhalb
c:[Type == "http://schemas.microsoft.com/ws/2012/01/insidecorporatenetwork",
  Value == "false"]
=> issue(Type =
"http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmethod",
  Value = "http://schemas.microsoft.com/claims/multipleauthn");

# Typische ADFS-Regel: bestimmte Netze per Regex zulassen
c:[Type == "http://schemas.microsoft.com/2012/01/requestcontext/claims/x-ms-forwarded-client-ip",
  Value =~ "^(203\.0\.113\.0|198\.51\.100\.0)$"]
=> issue(Type = "http://schemas.microsoft.com/authorization/claims/permit", Value = "true");
```

Listing 2: Zwei klassische Standortregeln aus der Claim-Rule-Sprache

Der Regex in Listing 2 beschreibt in Wahrheit zwei /24-Netze. Solche Ausdrücke rechnest du in CIDR-Notation um – und an dieser Stelle findest du regelmäßig Leichen: Netze von Standorten, die vor Jahren geschlossen wurden, oder ein Regex, der wegen eines vergessenen Punkts deutlich mehr Adressen erlaubt als gedacht. Die Übersetzung ist also gleichzeitig eine kleine Sicherheitsprüfung.

WARNUNG — Das Firmennetz ist nicht mehr das, was es war

In ADFS hieß „intern“ oft: kam nicht über den WAP. Mit Split-Tunnel-VPN, Homeoffice und Cloud-Proxys geht Datenverkehr deiner Benutzerinnen und Benutzer aber über ganz andere Ausgangsadressen ins Internet. Wenn du einfach alle IP-Adressen des Rechenzentrums als vertrauenswürdig einträgst, nimmst du womöglich Adressen auf, hinter denen auch Gäste-WLAN oder Partner sitzen. Lieber eine Named Location zu wenig als eine zu viel.

Eine Besonderheit gilt, solange deine Domäne in Entra ID noch gefördert ist: Ist in den Einstellungen für vertrauenswürdige IPs der MFA-Option die Option aktiviert, Anforderungen von förderierten Benutzern aus dem Intranet zu überspringen, behandelt Conditional Access jede Anmeldung mit dem `insidecorporatenetwork`-Claim von ADFS als Anmeldung von einem vertrauenswürdigen Standort. Das ist bequem für die Übergangszeit, aber auch eine Abhängigkeit, die du bei der Umstellung auf verwaltete Authentifizierung kennen musst – dazu mehr in Kapitel 4.

Gruppen und Benutzer

Gruppenregeln sind der einfachste Teil, mit einer Voraussetzung: Die Gruppe muss in Entra ID existieren. In ADFS prüfst du eine SID aus dem Token, in Entra ID eine Gruppe aus dem Verzeichnis. Für synchronisierte AD-Gruppen bedeutet das, dass Entra Connect sie mitnimmt. Die SID aus der Regel ordnest du über onPremisesSecurityIdentifier der Cloud-Gruppe zu, das geht mit Microsoft Graph PowerShell in einer Zeile:

```
Connect-MgGraph -Scopes "Group.Read.All"
Get-MgGroup -Filter "onPremisesSecurityIdentifier eq 'S-1-5-21-1111111111-2222222222-3333333333-4711'" |
Select-Object Id, DisplayName, OnPremisesSyncEnabled
```

Listing 3: Zur Gruppen-SID aus der Claim Rule die Entra-Gruppe finden

Erlaubende Gruppenregeln wandern in die Zuweisung der Enterprise App, nicht in Conditional Access. Das ist wichtig, weil Conditional Access keine Richtlinie „nur Gruppe X darf rein“ kennt – du könntest höchstens „alle außer Gruppe X blockieren“ bauen, und das ist schwerer zu lesen und fehleranfälliger. Ablehnende Gruppenregeln („deny für Praktikanten“) werden dagegen zu einer Blockieren-Richtlinie oder, eleganter, zu einem Ausschluss aus der zugewiesenen Gruppe.

Anwendung: aus der Relying Party wird die Zielressource

In ADFS hängt jede Regel an genau einer Relying Party – die Anwendung ist implizit. In Conditional Access wählst du sie unter den Zielressourcen aus, und zwar als Enterprise App. Das funktioniert erst, wenn die Anwendung tatsächlich an Entra ID angebunden ist; vorher gibt es dort schlicht nichts, worauf sich eine Richtlinie beziehen könnte. Wie du eine SAML-Anwendung umziehst, steht im Beitrag [SAML-Anwendungen von ADFS nach Entra ID migrieren](#). Für Microsoft 365 selbst ist die Zielressource „Office 365“ der richtige Griff, statt einzelne Dienste zu wählen, die untereinander abhängig sind.

Client-Typ: von x-ms-client-application zu Client-Apps

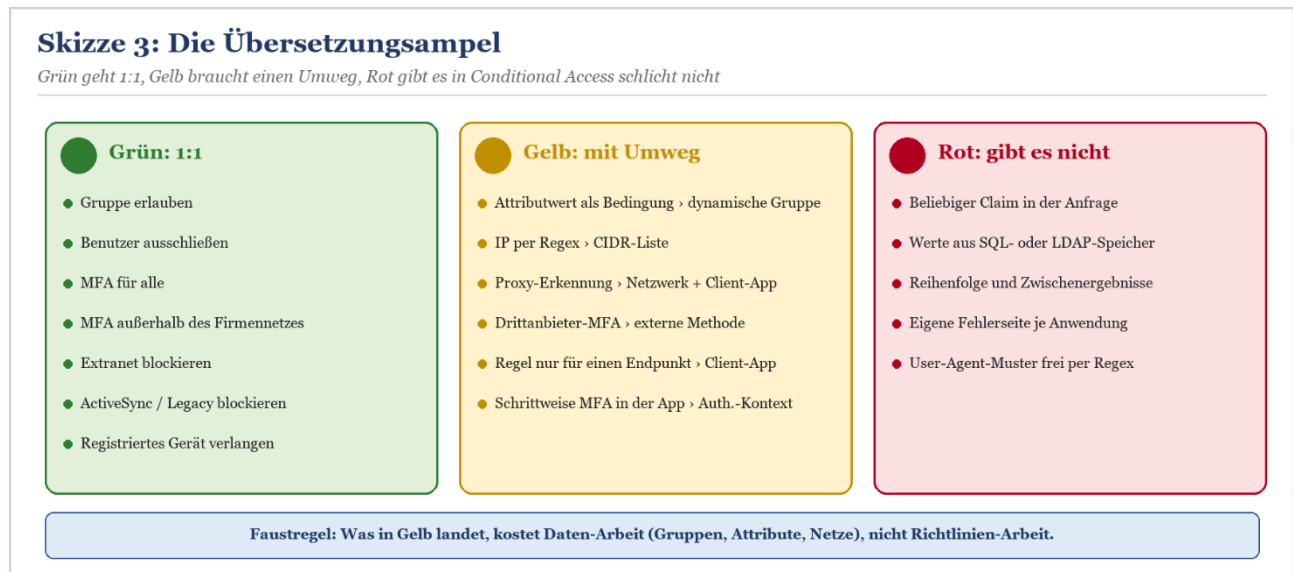
So stand es früher in jeder Anleitung: die fünf Regeln der Office 365 Client Access Policy, mit denen ADFS anhand von x-ms-client-application und x-ms-endpoint-absolute-path entschied, ob Exchange ActiveSync, Outlook über Basic Auth oder nur Browser von außen durften. Diese Regeln waren nötig, weil ADFS sonst nicht unterscheiden konnte, ob Exchange Online im Namen eines Clients anfragt. In Conditional Access ersetzt du das Ganze durch die Bedingung Client-Apps mit den Kategorien Browser, Mobile Apps und Desktopclients, Exchange ActiveSync-Clients und Andere Clients. Die Standardaufgabe – Legacy-Authentifizierung blockieren – ist eine einzige Richtlinie.

Claim in ADFS	Bedeutung	Gegenstück in Conditional Access
x-ms-client-application	Name der Client-Anwendung, z. B. Microsoft.Exchange.ActiveSync	Client-Apps: Exchange ActiveSync-Clients, Andere Clients
x-ms-endpoint-absolute-path	angesprochener ADFS-Endpunkt (passiv, WS-Trust)	Client-Apps; Authentifizierungsflüsse für Gerätecodefluss
x-ms-client-user-agent	User-Agent des Geräts	Geräteplattform (Android, iOS, Windows, macOS, Linux)
x-ms-proxy	Anfrage kam über den WAP	Netzwerk: alle außer vertrauenswürdige Standorte
x-ms-forwarded-client-ip	Client-IP laut Proxy	IP-basierte Named Location

Tabelle 3: Kontext-Claims des Anfrage-Kontexts und ihre Nachfolger

Was es nicht 1:1 gibt – und wie du die Lücke schließt

Jetzt der unbequeme Teil. Conditional Access kennt eine feste Menge an Signalen. Alles, was in ADFS auf einem anderen Claim beruhte, landet in einer von zwei Schubladen: Umweg über Daten oder Abschied. Die Ampel zeigt, wo deine Regeln wahrscheinlich landen.



Skizze 3: Die Übersetzungsampel – die meisten Regeln landen in Grün, die interessanten in Gelb, die Altlasten in Rot.

Gelb: Attribute, Regex und Drittanbieter-MFA

Der häufigste gelbe Fall ist eine Bedingung auf einem Benutzerattribut: „Nur wer department = Vertrieb hat“ oder „Externe mit extensionAttribute5 = Partner blockieren“. Conditional Access filtert Benutzer nicht nach Attributen. Der Umweg ist eine dynamische Gruppe in Entra ID mit genau dieser Regel; die Gruppe verwendest du dann in Zuweisung oder Richtlinie. Voraussetzung ist, dass das Attribut in Entra ID ankommt – bei Erweiterungsattributen also, dass Entra Connect sie synchronisiert. Ein schöner Nebeneffekt: Die Logik ist danach für alle sichtbar, nicht nur für die eine Person, die den Regex versteht.

Zweiter gelber Fall: ein MFA-Adapter eines Drittanbieters in ADFS. Conditional Access kann externe Authentifizierungsmethoden einbinden, die die früheren benutzerdefinierten Steuerelemente (Custom Controls) ablösen. Ob dein Anbieter das unterstützt, klärst du mit ihm – und ob du die zusätzliche Abhängigkeit behalten willst, wenn Entra-MFA mit Authenticator, FIDO2 oder Windows Hello for Business ohnehin zur Verfügung steht. Wie die Kopplung von ADFS mit Entra-MFA in der Übergangszeit aussieht, beschreibt [ADFS mit Entra-MFA koppeln – Adapter, Zertifikat und Fallstricke](#).

Dritter gelber Fall: Schritt-für-Schritt-MFA innerhalb einer Anwendung. Manche Anwendungen haben in ADFS per RequestedAuthenticationContext oder eigener Relying Party für den Adminbereich eine stärkere Anmeldung angefordert. In Entra ID gibt es dafür den Authentifizierungskontext: Die Anwendung fordert ihn an, eine Richtlinie hängt die Anforderung daran. Das funktioniert nur, wenn die Anwendung ihn auch anfordert – für viele ältere SAML-Anwendungen ist das nicht der Fall. Dann bleibt die zweite Enterprise App für den Adminbereich, wie früher die zweite Relying Party.

Rot: beliebige Claims, fremde Attributspeicher, Reihenfolge

Microsoft sagt es in der Migrationsdokumentation ohne Umschweife: Die Option „Mit bestimmten Claims in der Anfrage“ lässt sich nicht migrieren. Dasselbe gilt für Regeln, die per add() Werte aus

einem SQL- oder LDAP-Attributspeicher holen und darauf entscheiden, für Regeln, die auf Zwischenergebnisse anderer Regeln aufbauen, und für individuelle Fehlerseiten je Anwendung. Die drei Auswege sind ehrlich gesagt immer dieselben:

- Die Information zur Datenquelle machen: Was als Bedingung gebraucht wird, landet als synchronisiertes Attribut oder als Gruppenmitgliedschaft in Entra ID. Das ist der häufigste und beste Weg.
- Die Entscheidung in die Anwendung verlagern: Viele Anwendungen können Rollen selbst prüfen, wenn sie einen Gruppen- oder Rollen-Claim bekommen. Dann autorisiert die Anwendung, nicht der Identity Provider.
- Die Anwendung bewusst auf ADFS lassen, bis sie abgelöst wird. ADFS ist nicht tot, und eine einzelne Relying Party mit Sonderlogik ist kein Grund für Panik – nur ein Grund, die Farm sauber weiterzubetreiben.

WICHTIG — Nicht alles, was rot ist, war je sinnvoll

Ein guter Teil der roten Regeln existiert, weil sie irgendwann jemand für einen Sonderfall gebaut hat, der nicht mehr existiert. Frag bei jeder roten Regel zuerst die Fachabteilung, ob die Bedingung noch gebraucht wird. Erfahrungsgemäß löst sich so ein Drittel der Lücken in Luft auf – ein seltenes Beispiel für Technik, die durch Nachfragen verschwindet statt durch Arbeit.

Lücke	Ursache	Wie du sie schließt
Bedingung auf beliebigem Claim	CA kennt nur feste Signale	Attribut synchronisieren, dynamische Gruppe
Attributspeicher SQL/LDAP	Entra ID liest keine fremden Quellen	Daten per Entra Connect oder Provisioning nach Entra ID, sonst Anwendung auf ADFS lassen
Regex auf IP	Named Locations nur CIDR	Regex in CIDR-Bereiche umrechnen, Altnetze streichen
Proxy-Erkennung	in der Cloud gibt es keinen WAP	vertrauenswürdige Standorte; mit Global Secure Access auch das kompatible Netzwerk
Regelreihenfolge, else-Zweig	CA wertet alles gemeinsam aus	Include/Exclude sauber schneiden, Blockieren-Richtlinien sparsam
Eigene Fehlerseite je RP	einheitliche Entra-Meldung	Unternehmensbranding, Hinweis im Intranet, Servicedesk vorbereiten
Drittanbieter-MFA-Adapter	kein ADFS-Adapter in Entra ID	externe Authentifizierungsmethode oder Entra-MFA

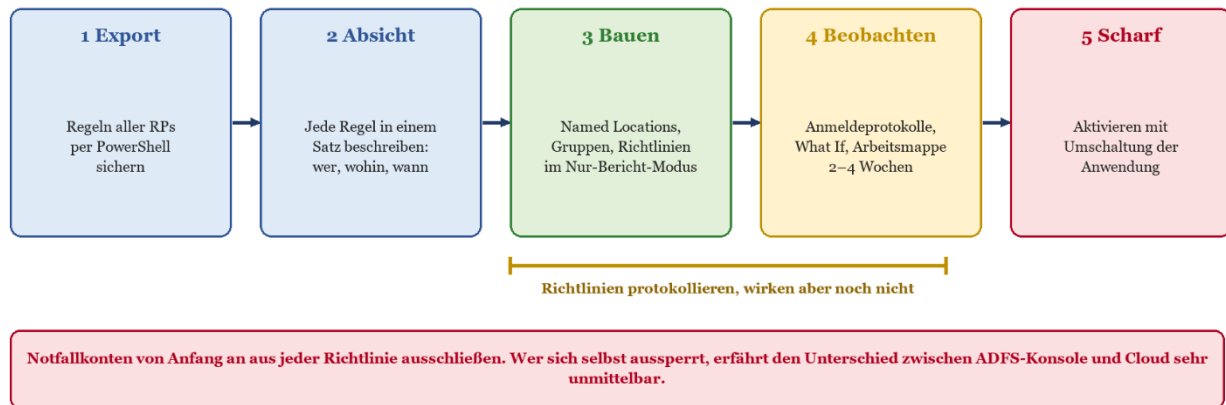
Tabelle 4: Typische Lücken und ihre Lösung

Richtlinien bauen, testen und die Nahtstelle zur Föderation

Die Übersetzung auf Papier ist das eine, die Richtlinien im Tenant das andere. Damit du die Kontrolle behältst, gehst du in festen Etappen vor – und die wichtigste davon ist die, in der scheinbar nichts passiert.

Skizze 4: Vom Regelexport zur scharfen Richtlinie

Fünf Etappen – die längste davon ist das Zuschauen im Nur-Bericht-Modus



Skizze 4: Fünf Etappen vom Regelexport bis zur aktiven Richtlinie.

Named Location und Richtlinie per Microsoft Graph PowerShell

Im Entra Admin Center klickst du das in wenigen Minuten zusammen. Wenn du aber zwanzig Relying Parties übersetzt, willst du die Richtlinien reproduzierbar anlegen. Listing 4 baut die Standortregel aus Listing 2 nach: eine vertrauenswürdige Named Location und eine Richtlinie, die außerhalb davon MFA verlangt – zunächst im Nur-Bericht-Modus. Die Platzhalter für App-ID und Gruppen-IDs ersetzt du durch deine Werte.

```

Connect-MgGraph -Scopes "Policy.ReadWrite.ConditionalAccess", "Policy.Read.All",
"Application.Read.All"

# 1. Vertrauenswürdige Named Location aus den umgerechneten Netzen
$loc = New-MgIdentityConditionalAccessNamedLocation -BodyParameter @{
    "@odata.type" = "#microsoft.graph.ipNamedLocation"
    displayName   = "Standorte contoso"
    isTrusted     = $true
    ipRanges      = @(
        @{ "@odata.type" = "#microsoft.graph.ipV4CidrRange"; cidrAddress = "203.0.113.0/24" },
        @{ "@odata.type" = "#microsoft.graph.ipV4CidrRange"; cidrAddress = "198.51.100.0/24" }
    )
}

# 2. Richtlinie: MFA außerhalb der Standorte, erst nur protokollieren
$policy = @{
    displayName = "CA-Bewerberportal: MFA außerhalb der Standorte"
    state       = "enabledForReportingButNotEnforced"
    conditions  = @{
        applications = @{ includeApplications = @("<App-ID der Enterprise App>") }
        users        = @{ includeGroups = @("<ID Gruppe Personal>")
                          excludeGroups = @("<ID Gruppe Notfallkonten>") }
        locations    = @{ includeLocations = @("All"); excludeLocations = @($loc.Id) }
        clientAppTypes = @("all")
    }
    grantControls = @{ operator = "OR"; builtInControls = @("mfa") }
}
New-MgIdentityConditionalAccessPolicy -BodyParameter $policy

```

Listing 4: Standortregel als Named Location und Conditional-Access-Richtlinie

Für die Legacy-Regeln der alten Client Access Policy reicht eine zweite Richtlinie mit clientAppTypes exchangeActiveSync und other und dem Gewährungssteuerelement block. Wenn die Anwendung zugewiesen ist und die Gruppe Personal in der Zuweisung steht, ist die Bewerberportal-Regel vom Anfang damit komplett übersetzt – bis auf die drei Leute vom

Betriebsrat, die jetzt einfach Mitglied der zugewiesenen Gruppe sind und außerhalb der Standorte MFA machen wie alle anderen. Ausnahmen, die keiner mehr erklären kann, sind die eigentliche technische Schuld.

TIPP — Nur-Bericht-Modus ernst nehmen

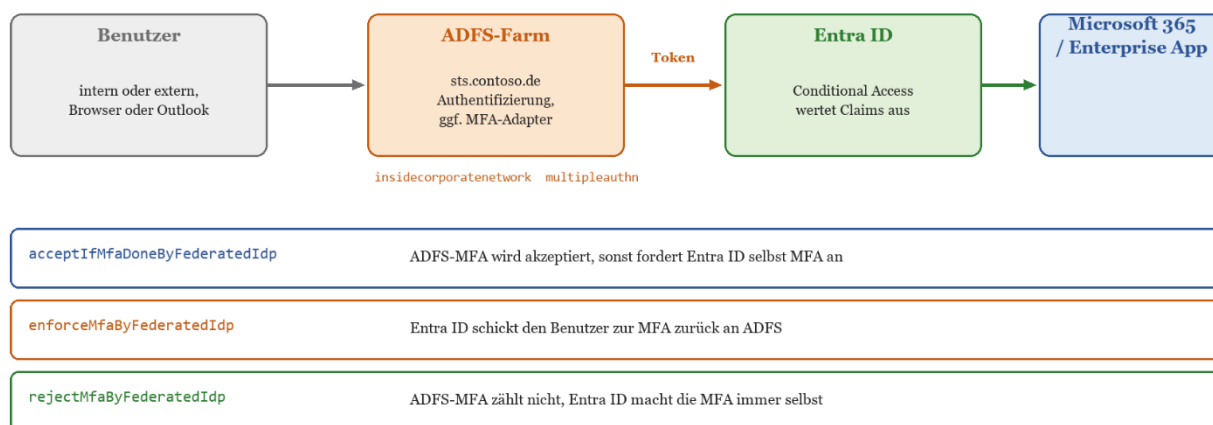
Lass neue Richtlinien mindestens zwei bis vier Wochen im Nur-Bericht-Modus laufen und schau in die Anmeldeprotokolle: Dort steht je Anmeldung, welche Richtlinie zugetroffen hätte und mit welchem Ergebnis. Das What-If-Werkzeug im Entra Admin Center beantwortet die Frage „Was passiert, wenn Benutzer X von Ort Y mit Client Z kommt?“, bevor jemand anruft. Beides gab es in ADFS nicht, und es ist der beste Grund, sich auf Conditional Access zu freuen.

Die Nahtstelle: Wer macht die MFA, solange ADFS noch föderiert?

Solange deine Domäne in Entra ID föderiert ist, melden sich Benutzer bei ADFS an, und ADFS schickt einen Token mit Claims an Entra ID. Conditional Access kann dann MFA verlangen – und die Frage ist, wer sie erbringt. Das regelt die Einstellung `federatedIdpMfaBehavior` der Domänenföderation. Mit `acceptIfMfaDoneByFederatedIdp` akzeptiert Entra ID eine MFA, die ADFS bereits gemacht und per `multipleauthn`-Claim mitgeteilt hat; fehlt sie, fordert Entra ID selbst MFA an. Mit `enforceMfaByFederatedIdp` schickt Entra ID den Benutzer für die MFA zurück an ADFS. Mit `rejectMfaByFederatedIdp` zählt die ADFS-MFA gar nicht, Entra ID macht die MFA immer selbst.

Skizze 5: Die Nahtstelle, solange die Domäne föderiert ist

ADFS liefert Claims, Conditional Access entscheidet – `federatedIdpMfaBehavior` regelt, wer die MFA macht



Skizze 5: In der Übergangszeit liefert ADFS die Claims, Conditional Access trifft die Entscheidung.

```

Connect-MgGraph -Scopes "Domain.ReadWrite.All"
$fed = Get-MgDomainFederationConfiguration -DomainId "contoso.de"
$fed | Select-Object Id, IssuerUri, FederatedIdpMfaBehavior

# Entra ID übernimmt die MFA, ADFS-MFA wird nicht mehr anerkannt
Update-MgDomainFederationConfiguration -DomainId "contoso.de" `
    -InternalDomainFederationId $fed.Id `
    -BodyParameter @{ federatedIdpMfaBehavior = "rejectMfaByFederatedIdp" }

# So stand es früher in jeder Anleitung (MSOnline, ausgemustert):
# Set-MsolDomainFederationSettings -DomainName contoso.de -SupportsMfa $true

```

Listing 5: MFA-Verhalten der föderierten Domäne prüfen und umstellen

Für die Migration ist das ein praktischer Hebel: Du kannst Conditional Access mit MFA bereits aktiv betreiben, während ADFS noch die primäre Authentifizierung macht. Stellst du die MFA auf Entra ID um, registrieren sich die Benutzer frühzeitig für Entra-MFA, und der spätere Wechsel von föderiert auf verwaltet verliert seinen Schrecken. Wie dieser Wechsel mit Staged Rollout aussieht, steht in [Microsoft 365 von Federated auf Managed umstellen – mit Staged Rollout](#). Vergiss dabei nicht die Einstellung zum insidecorporatenetwork-Claim aus Kapitel 2: Nach der Umstellung kommt dieser Claim nicht mehr, und alles, was sich darauf verlassen hat, braucht Named Locations.

WARNUNG — MFA auf beiden Seiten ist kein Sicherheitsgewinn, sondern Ärger

Wenn ADFS per Additional Authentication Rule MFA verlangt und Conditional Access mit rejectMfaByFederatedIdp ebenfalls, machen Benutzer zweimal MFA – und rufen beim zweiten Mal den Servicedesk an. Entscheide dich je Phase für eine Seite und schalte die ADFS-Regel ab, sobald Entra ID übernimmt. Was ADFS in der Zwischenzeit trotzdem gegen Passwort-Spray schützen muss, bleibt davon unberührt.

Härtung für die Übergangszeit: [Extranet Smart Lockout – ADFS gegen Passwort-Spray schützen](#).

WEITER — Passende Beiträge für die nächsten Schritte

- › [ADFS Claim Rules Leitfaden](#)
- › [Access Control Policies in ADFS – Zugriffsregeln ohne Claim-Rule-Akrobatik](#)
- › [Relying Parties inventarisieren – die Bestandsaufnahme vor der Entra-Migration](#)
- › [SAML-Anwendungen von ADFS nach Entra ID migrieren](#)
- › [ADFS mit Entra-MFA koppeln – Adapter, Zertifikat und Fallstricke](#)
- › [Microsoft 365 von Federated auf Managed umstellen – mit Staged Rollout](#)
- › [Microsoft Entra ID](#)

Fazit: Absicht übersetzen, nicht Syntax

ADFS Claim Rules nach Conditional Access zu übersetzen, ist zu zwei Dritteln Fleißarbeit und zu einem Drittel Archäologie. Gruppen werden zu Zuweisungen, Standortregeln zu Named Locations, MFA-Regeln zu Gewährungssteuerelementen, die alte Client Access Policy zu einer einzigen Richtlinie gegen Legacy-Authentifizierung. Was übrig bleibt – beliebige Claims, fremde Attributspeicher, Regelketten –, schließt du über Daten: Attribute synchronisieren, dynamische Gruppen bauen, Rollen in die Anwendung verlagern. Und was sich so nicht schließen lässt, darf auf ADFS bleiben, bis die Anwendung selbst in Rente geht. Die Farm läuft dafür genauso zuverlässig weiter wie bisher.

Wenn du vor einem Berg von Relying Parties mit gewachsener Autorisierungslogik stehst und ein schlankes Richtlinienset daraus machen willst, unterstützen wir dich im [Consulting zu ADFS \(Active Directory Federation Services\)](#). Die Hintergründe zu Claim Rules, Access Control Policies und Migrationspfaden vertieft [ADFS in der Praxis – das Buch im Detail](#), und wer das Übersetzen einmal in Ruhe im Labor üben möchte, findet das passende Format in den [ADFS-Schulungen](#).

FAQ: ADFS Claim Rules und Conditional Access

Kann ich ADFS Claim Rules direkt in Conditional Access importieren?

Nein. Es gibt keinen Import und keine gemeinsame Regelsprache. Du übersetzt die Absicht jeder Regel in Zuweisungen und Richtlinien. Der Export per Get-AdfsRelyingPartyTrust ist dein Bauplan.

Wohin gehören Issuance Authorization Rules in Entra ID?

Erlaubende Regeln in die Benutzerzuweisung der Enterprise App mit „Zuweisung erforderlich“, ablehnende Regeln und Bedingungen wie Netzwerk oder Gerät in Conditional Access.

Was ist das Gegenstück zu insidecorporatenetwork?

Eine IP-basierte, als vertrauenswürdig markierte Named Location. Solange die Domäne föderiert ist, kann Conditional Access den Claim von ADFS bei entsprechender Einstellung zusätzlich als vertrauenswürdigen Standort werten.

Wie übersetze ich einen Regex auf x-ms-forwarded-client-ip?

Rechne die erfassten Adressen in CIDR-Bereiche um und trage sie in eine Named Location ein. Pro Named Location sind bis zu 2000 Bereiche erlaubt, Masken kleiner oder gleich /8 nicht.

Wie ersetze ich die Office 365 Client Access Policy aus ADFS?

Mit einer Conditional-Access-Richtlinie, die für die Client-Apps Exchange ActiveSync und Andere Clients den Zugriff blockiert, ergänzt um MFA-Richtlinien für Browser und moderne Clients.

Kann Conditional Access nach Benutzerattributen wie department filtern?

Nicht direkt. Baue eine dynamische Gruppe mit der Attributregel und verwende diese Gruppe in Zuweisung oder Richtlinie. Das Attribut muss dafür in Entra ID vorhanden sein.

Was mache ich mit Regeln, die „bestimmte Claims in der Anfrage“ prüfen?

Laut Microsoft lassen sie sich nicht migrieren. Prüfe, ob die Bedingung noch gebraucht wird, bringe die Information als Attribut oder Gruppe nach Entra ID oder lass die Anwendung vorerst auf ADFS.

Wer macht die MFA, solange Microsoft 365 noch über ADFS föderiert ist?

Das regelt federatedIdpMfaBehavior an der Domänenföderation: ADFS-MFA akzeptieren, MFA bei ADFS erzwingen oder ADFS-MFA ablehnen und immer Entra-MFA verwenden.

Brauche ich für Conditional Access eine besondere Lizenz?

Ja, Microsoft Entra ID P1, die in vielen Microsoft-365-Paketen enthalten ist. Risikobasierte Bedingungen benötigen P2.

Wie teste ich übersetzte Richtlinien ohne Risiko?

Im Nur-Bericht-Modus. Die Anmeldeprotokolle zeigen, was die Richtlinie bewirkt hätte, und das What-If-Werkzeug simuliert einzelne Anmeldungen. Notfallkonten schließt du von Anfang an aus.

Muss ich die ADFS-MFA-Regeln nach der Übersetzung löschen?

Deaktivieren solltest du sie, sobald Entra ID die MFA übernimmt, sonst machen Benutzer doppelt MFA. Die exportierten Regeln bewahrst du als Dokumentation auf.