FACHKONZEPT

Identity & Access Management

März 2024

FACHKONZEPT

Identity & Access Management

März 2024

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Einleitung

Am 30.11 2016 verständigten sich die Innenminister des Bundes und der Länder im Rahmen ihrer Herbstkonferenz auf die Saarbrücker Agenda zur Informationsarchitektur der Polizeien des Bundes und der Länder als Teil der Inneren Sicherheit. Damit wurden die Weichen dafür gestellt, das Informationsmanagement grundlegend zu modernisieren und zu vereinheitlichen. Kernziele der Modernisierung sind:

- Verbesserung der **Verfügbarkeit** polizeilicher Informationen,
- Erhöhung der **Wirtschaftlichkeit**
- Stärkung des **Datenschutzes** durch Technik.

Ein zeitgemäßes, den Herausforderungen der Sicherheitslage Rechnung tragendes Informationsmanagement auf der Basis einer modernen Informationsarchitektur schafft wesentliche Voraussetzungen für eine effektive und effiziente Wahrnehmung der polizeilichen Aufgaben von Bund und Ländern zur Gewährleistung der Inneren Sicherheit durch Abwehr von Gefahren und wirksame Kriminalitätsbekämpfung.

Dieses Dokument beschreibt fachlich den Ansatz des Identity & Access Management der PG IAM für den Basisdienst IAM des Datenhausökosystems des Programms P20. Zum Zeitpunkt der Konzepterstellung wurden bereits, in Zusammenarbeit der PG IAM und des BKAs, technische Festlegungen getroffen bzw. einzelne geplante Schnittstellen umgesetzt. Mit diesem Dokument sollen die Grundlagen für ein einheitliches Verständnis festgehalten werden. Es löst alle früheren Konzepte der PG'en zum IAM im Kontext P20 ab, fasst sie zusammen und setzt sie in Kontext zu einander. Dabei soll klargestellt werden, welche Schnittstellen empfohlen sind und möglichst nah an Industriestandards bleiben. Hier nicht erwähnte im Einsatz befindliche oder geplante Lösungen sind damit explizit nicht empfohlen; nichtsdestotrotz werden solche Sonderlösungen durch den Basisdienst IAM weiterhin ermöglicht werden, wo die Priorisierung, Dringlichkeit und fehlende Flexibilität des anzuschließenden Produktes es erfordern – stets in Abstimmung mit dem Kernteam Technik (KTT). Soweit möglich, soll es sich dabei jedoch immer um Übergangslösungen auf dem Weg zu einheitlichen Lösungen handeln. Nur so kann der Basisdienst IAM effizient betrieben werden.

Tiefer gehende technische Details wie bspw. Schnittstellenbeschreibungen sind nicht Teil dieses Dokuments, sondern werden im Confluence-Bereich zum Basisdienst IAM gepflegt. Dort finden sich auch Informationen zum Beratungsangebot inkl. Möglichkeit zur Automatisierten Buchung von Sprechstunden. Wir empfehlen TN-IAM-Teams die Seite „P20 Dokumentenübersicht“ zu abonnieren, um Änderungen sofort mitzubekommen:
https://confluence.bka.extrapol.de/x/1QayBw

Auf konkrete Produkte wird sich in diesem Dokument ebenfalls nicht festgelegt. Soweit möglich und effizient betreibbar, sind im Allgemeinen Open Source Komponenten zu bevorzugen. Um eine hohe Serviceklasse zu ermöglichen, kann sich aber der Einsatz lizensierter Produkte mit entsprechendem Support-Netzwerk lohnen. Darin besteht auch einer der Gründe, möglichst nach an den Industriestandards zu bleiben bei den Schnittstellen; nur so können einzelne Komponenten bei Bedarf ohne unrealistische Kosten ausgetauscht werden. (Verhinderung eines Vendor-Lock-ins.)

Zentrales Persistieren personenbezogener Daten

Hier aktuell noch explizit ausgeklammert ist außerdem das Thema der Vermeidung einer zentralen Persistierung im Zielbild 2030+: Im Sinne der Datensparsamkeit wird angestrebt, langfristig so wenig personenbezogene Daten wie möglich im F-IAM zu persistieren. Für die Funktionen des F-IAM an sich werden folgende Informationen je Nutzer benötigt: P20-UID, TN, Berechtigungszuweisungen.

Die an das F-IAM angrenzenden Systeme haben jedoch diverse Bedarfe an weiteren Benutzerattributen. Aktuell werden immer wieder zusätzliche Attribute angefragt.

Es gibt bereits ein paar Vorschläge, wie die angrenzenden Systeme mit den jeweils spezifisch benötigten Benutzerattributen versorgt werden können, ohne dass diese zentral im F-IAM persistiert werden müssen. (Bspw. "BIMS" als Attribute Broker aus früheren Versionen von F-IAM-Konzepten. Oder die Auslagerung in Kataloge wie beim Dienststellenkatalog. Oder auch die breitere Nutzung der Möglichkeit zur Pseudonymisierung.) Aktuell ist allerdings noch nicht realistisch absehbar, welche der Lösungen für TN-IAM-Systeme und Applikationen (sowohl Eigenentwicklungen als auch Kaufsoftware) ebenfalls effizient umsetzbar sein wird. Daher wird vorerst auf die Festlegung einer konkrete Lösung verzichtet, sodass zu einem späteren Zeitpunkt diese Anforderung erneut zu betrachten ist. Anschließens ist dann eine praktikable Lösung zu erarbeiten, in einer neuen Version dieses Dokuments festzuhalten und für das F-IAM sowie die angrenzenden Systeme umzusetzen.

Inhalte aus früheren Dokumenten

Aus den folgenden Dokumenten wurden Inhalte teilweise in die genannten Kapitel eingebracht:

- Einführung einer personenbezogenen ID im Programm Polizei 2020 v2.0 vom 02.06.2022
  - Hinweise zu Aufbau und Generierung übernommen nach Kapitel 4.7.2., weitere Inhalte eingearbeitet in Confluence1.
- IAM Transformationsgrobkonzept v2.0 vom 24.11.2022
  - Prozesse übernommen in das Kapitel 4.3.
  - Die anderen Aspekte verteilt eingearbeitet in Kapitel 4. oder entfallen, da bereits umgesetzt oder im aktuellen Projektauftrag IAM festgelegt
- Anbindung von Teilnehmern an das F-IAM v1.0 vom 22.05.2023
  - Eingearbeitet in Kapitel 4., insbes. 4.6.3., 4.7.4., 4.7.5.
  - Sequenzdiagramme eingearbeitet in: 5.1. bis 5.4.
- Anbindung von Anwendungen an das F-IAM v1.5 vom 15.08.2023
  - Eingearbeitet in Kapitel 4., insbes. 4.5., 4.6.1., 4.6.2., 4.6.4., 4.7.6.
  - Sequenzdiagramme eingearbeitet in: 5.1. bis 5.4.
- Identity Management Fachkonzept v0.5 vom 15.08.2023
  - Verteilt eingearbeitet in das Kapitel 4.

Inhaltliche Änderungen (ChangeLog)

Im Folgenden wird eine Übersicht gegeben, welche konzeptuellen Inhalte geändert wurden, welche bei implizit vorhandener Regelung/langfristig ausgelegter Umsetzung klargestellt wurden und welche tatsächlich neu sind. Dabei sind die Stellen, an denen diese Inhalte zu finden sind, verlinkt.

Geänderte Inhalte

- Die P20-UID wird zunächst als zusätzliches Benutzerattribut eingeführt. Eine Umstellung als identifizierendes Merkmal ist erst möglich, wenn sie von allen TN für alle Benutzer bereitgestellt wird. Die zusätzliche Nutzung als identifizierendes Merkmal ist jedoch TN-individuell möglich: Kapitel 4.7.2.
- Zum Abfragen der AW-Funktionsrechte ist nicht zwingend die bisherige Schnittstelle (ZeRo 1) zu nutzen, sondern die Möglichkeit wird zusätzlich direkt über die TN-SCIMv2-Extended-Schnittstelle angeboten werden: Kapitel 4.7.5., S. 104
- TN-individuelle Rollen werden im F-IAM definiert und Benutzern zugewiesen werden können. Dafür muss keine eigene Schnittstelle (ZeRo 2) angesprochen werden, sondern die Möglichkeit wird direkt über die TN-SCIMv2-Extended-Schnittstelle angeboten werden: Kapitel 4.7.5., S. 104

Klarstellungen

- TN dürfen über mehrere IdPs an dieselbe Umgebung angebunden sein, sofern es hierfür gute Gründe gibt und dies mit dem F-IAM-Team abgestimmt ist: Kapitel 4.4.
- Durch einen IdP für Dritte kann das BKA Nicht-Polizei-Behörden den Zugriff auf polizeiliche Fachverfahren und zentrale Datenbanken ermöglichen: Kapitel 4.4.
- Das F-IAM ist nicht auf Privileged Account Management ausgelegt: Kapitel 4.6.2., S. 87
- Der Zugriff auf P20-Webservices durch TN-Anwendungen wird ermöglicht, indem die TN-Anwendungen einen Token-Exchange beim F-IAM vornehmen. Dabei muss ein OIDC-Access-Token durch das eigene IAM-System ausgestellt und durch die TN-Anwendung dem F-IAM übergeben werden: Kapitel 4.6.3.
- Eine LDAP-basierte IDM-Anbindung ist sowohl für TN als auch für AW nur übergangsweise angedacht. Langfristig wird SCIMv2 empfohlen: Kapitel 4.7.4., Varianten, S. 103
- Klares Staging-Konzept inklusive Serviceklassen: Kapitel 4.9.

Neue Inhalte

- Allgemeine Einleitung in das Thema IAM: Kapitel 2. und 3.
- Anwendungen, die ihre Berechtigungsinformationen über den Userprofile-Endpunkt beziehen, dürfen diese Informationen cachen: Kapitel 4.6.2., S. 87

- Die Liste der Benutzerattribute im F-IAM und auch innerhalb der Access-Tokens wurde erweitert und zugunsten der dynamischen Anforderungen der Anwendungsteams nach Confluence1 ausgelagert. Perspektivisch ist angedacht, dass Access-Tokens nur genau die Benutzerattribute enthalten, die von der Zielanwendung benötigt werden: Kapitel 4.7.1.
- Ausführungen zur Einführung eines referenzierenden P20-Dienststellenschlüssels als Benutzerattribut sowie bei Berechtigungszuweisungen mit OE-Bezug: Kapitel 4.7.3.
- Anwendungen können für bestimmte Rechte eine explizite Genehmigung durch Anwendungsbetreuer veranlassen: Kapitel 4.7.5., S. 101
- Beschreibung, wie mit Provisionierungsfehlern in Richtung Anwendung umgegangen wird: Kapitel 4.7.4., S. 98
- Beschreibung, wie mit Abweichungen beim regelmäßigen Vollabgleich zwischen F-IAM und Anwendung umgegangen wird: Kapitel 4.7.4., S. 98
- Ergänzung von Berechtigungszuweisungen mit OE-Bezug um eine optionale Vorgabe zur Vererbung an untergeordnete OEn: Kapitel 4.8., insbes. S. 110
- Folgende Sequenzdiagramme:
  - Gesamtkette der Authentifizierung per AW-OIDC, TN-SAML und Kerberos: Kapitel 5.1.6.
  - Webservice-Aufruf mit Token-Exchange: Kapitel 5.4.1.
  - Berechtigungsabfrage über Userprofile-Endpunkt: Kapitel 5.4.2.

Verständnis von Identity & Access Management

Was ist Identity & Access Management überhaupt? Die Antwort auf diese Frage ist sowohl einfach als auch sehr komplex. Der einfache Teil ist: Bei Identity & Access Management (IAM) handelt es sich um eine Reihe von Informationstechnologien, die sich mit Identitäten im Cyberspace befassen. Der komplexe Teil der Antwort nimmt den größten Teil dieses Dokuments ein.

Dieses Dokument befasst sich hauptsächlich mit *Enterprise Identity und Access Management*.

Dabei handelt es sich um IAM, das auf größere Organisationen angewendet wird. Der Schwerpunkt liegt auf der Verwaltung von Mitarbeitern, Auftragnehmern, Kunden, Partnern, Studenten und anderen Personen, die mit der Organisation zusammenarbeiten. Viele der in diesem Dokument beschriebenen Mechanismen und Prinzipien können jedoch auch auf andere Umgebungen angewendet werden.

Die Geschichte des IAM's beginnt mit der Informationssicherheit. Die Sicherheitsanforderungen erfordern eine Authentifizierung und Autorisierung von Benutzern. Bei der Authentifizierung handelt es sich um einen Mechanismus, mit dem ein Computer überprüft, ob der Benutzer tatsächlich derjenige ist, für den er sich ausgibt. Und die Autorisierung ist ein verwandter Mechanismus, mit dem ein Computer bestimmt, ob er dem Benutzer eine bestimmte Aktion erlaubt oder verweigert (Bspw. lesenden oder schreibenden Zugriff auf ein Dokument.) Fast jedes Computersystem verfügt über Mittel zur Authentifizierung und Autorisierung.

Die vielleicht am weitesten verbreitete Form der Authentifizierung ist ein Anmeldeverfahren basierend auf einem Kennwort. Der Benutzer gibt eine Kennung und ein Kennwort an. Der Computer prüft, ob das Kennwort gültig ist. Damit dieser Vorgang funktioniert, benötigt der Computer Zugriff auf eine Datenbank aller gültigen Benutzer und Kennwörter. Frühe eigenständige Informationssysteme verfügten über eigene Datenbanken, die vom Rest des Cyberspace isoliert waren. Die Daten wurden manuell gepflegt. Aber das Aufkommen der Computervernetzung erforderte weitergehende Mechanismen. Benutzer konnten auf viele Systeme zugreifen und die Systeme selbst waren miteinander verbunden. Die Pflege einer isolierten Datenbank für Benutzer in jedem System ergab keinen Sinn mehr.

Hier beginnt die wahre Geschichte der digitalen Identität.

Verzeichnisdienste und andere Benutzerdatenbanken

Der zentrale Begriff der Verwaltung von Identitäten ist ein Datensatz, der Informationen über eine Person enthält. Dieses Konzept hat viele Namen: Benutzerprofil, Persona, Benutzerdatensatz, digitale Identität und viele mehr. Der gebräuchlichste Name im Kontext des Identitätsmanagements ist *Benutzerkonto*. Konten enthalten normalerweise Informationen, die die reale Person anhand einer Reihe von Attributen wie Vor- und Nachnamen beschreiben. Der wohl wichtigste Teil sind jedoch die technischen Informationen, die sich auf den Betrieb eines Informationssystems beziehen, für welches das Konto erstellt wird. Dazu gehören operationale Parameter wie der Standort des Home-Verzeichnisses des Benutzers, eine Vielzahl von Berechtigungsinformationen wie Gruppen- und Rollenmitgliedschaft, Beschränkung des Zugriffs auf Systemressourcen usw. Benutzerkonten werden in einer Vielzahl von Formen dargestellt, die von Datensätzen relationaler Datenbank über strukturierte Datendateien bis hin zu semi-

strukturierten Textdateien reichen. Aber unabhängig von der spezifischen Methode, die zum Speichern und Verarbeiten der Datensätze verwendet wird, ist das Konto zweifellos eines der wichtigsten Konzepte im IAM-Bereich. Und das gilt auch für die Datenbanken, in denen die Konten entsprechend als Datensätze gespeichert sind.

Die Datenbanken für Benutzerkonten sind so vielfältig wie die Kontotypen. Die meisten Datenbanken für Benutzerkonten wurden in der Vergangenheit als integraler Bestandteil des monolithischen Informationssystems implementiert und nutzten dieselbe Datenbanktechnologie wie das verwendete System selbst. Dies ist eine offensichtliche Wahl und erfreut sich auch heute noch großer Beliebtheit. Daher werden viele Konten in relationalen Datenbanktabellen und ähnlichen Datenspeichern von Anwendungen gespeichert.

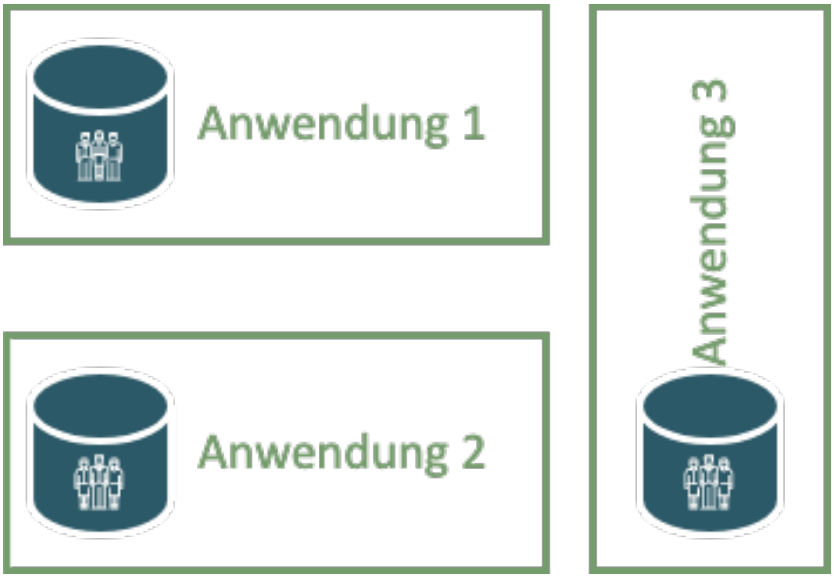

Abbildung 1: Applikationspezifischer Identitätsspeicher

Datenspeicher von Anwendungen sind normalerweise eng an die Anwendung gebunden. Daher ist es schwierig, in solchen Datenbanken gespeicherte Konten mit anderen Anwendungen zu teilen. Andererseits ist die gemeinsame Nutzung von Kontodaten im gesamten Unternehmen mehr als wünschenswert. Es macht wenig Sinn, Kontodaten in jeder Datenbank separat zu verwalten – insbesondere, wenn die meisten Konten in jeder Anwendung gleich sind. Daher besteht eine starke Motivation, Kontodatenbanken bereitzustellen, die von vielen Anwendungen gemeinsam genutzt werden können und somit sowohl eine effiziente Administration als auch einen möglichst effizienten Workflow für den Anwender ermöglichen..

Verzeichnisdienste werden in erster Linie für die Bereitstellung gemeinsamer Datenspeicherung für Anwendungen entwickelt. Während Datenbanken von Anwendungen normalerweise ihr eigenes proprietäres Protokoll verwenden, implementieren *Verzeichnisdienste* standardisierte Protokolle. Während Datenbanken für anwendungsspezifische Datenmodelle erstellt werden, erweitern Verzeichnisdienste normalerweise standardisierte Datenmodelle, was die Interoperabilität verbessert. Während Datenbanken schwergewichtig und teuer in der Skalierung sein können, sind Verzeichnisdienste darauf ausgelegt, leichtgewichtig zu sein und eine enorme Skalierbarkeit zu bieten. Das macht Verzeichnisdienste zu nahezu idealen Kandidaten für eine gemeinsame Kontodatenbank.

Der gemeinsame Identitätsspeicher erleichtert die Benutzerverwaltung. Ein Konto muss nur an einem Ort erstellt und verwaltet werden. Die Authentifizierung erfolgt weiterhin in jeder Anwendung separat. Da die Anwendungen jedoch dieselben Anmeldeinformationen aus dem gemeinsam genutzten Speicher verwenden, kann der Benutzer für alle verbundenen Anwendungen dasselbe Kennwort verwenden. Dies ist eine Verbesserung gegenüber der separaten Festlegung des Passworts für jede Anwendung.

Abbildung 2: Zentraler Verzeichnisdienst für Anwendungen

Auf gemeinsam genutzten Verzeichnisdiensten basierende Lösungen für das Identitätsmanagement sind einfach und recht kosteneffizient. Das Problem ist, dass dies meist nur bei sehr einfachen Systemen funktioniert.

Lightweight Directory Access Protokoll (LDAP)

Lightweight Directory Access Protocol (LDAP) ist ein Standardprotokoll für den Zugriff auf Verzeichnisdienste. Gemessen an den Standards des Zeitalters des Internets handelt es sich um ein altes Protokoll. Die Wurzeln von LDAP reichen bis in die 1980er Jahre zurück, bis zu einer Familie von Protokollen zur Telekommunikation namens X.500. Auch wenn LDAP alt sein mag, ist es weit verbreitet. Es handelt sich um ein sehr effizientes Binärprotokoll, das für die Unterstützung massiv verteilter gemeinsam genutzter Datenbanken entwickelt wurde. Es verfügt über einen kleinen Satz gut definierter einfacher Operationen. Die durch das Protokoll implizierten Operationen und das Metamodell für Daten ermöglichen eine sehr effiziente Replikation von Daten und horizontale Skalierbarkeit von Verzeichnisdiensten. Diese Einfachheit trägt zu geringen Latenzen und einem hohen Durchsatz bei Lesevorgängen bei. Die horizontale Skalierbarkeit und relative Autonomie von Instanzen von Verzeichnisdiensten soll die Verfügbarkeit des Verzeichnissystems erhöhen. Diese Vorteile gehen oft zu Lasten langsamer Schreibvorgänge. Da Identitätsdaten häufig gelesen, aber selten geändert werden, sind langsamere Schreibvorgänge normalerweise ein durchaus akzeptabler Kompromiss. Daher waren und sind LDAP-basierte Verzeichnisserver vielerorts noch immer die beliebtesten Datenbanken für Identitätsdaten.

LDAP ist einer der wenigen etablierten Standards im IAM-Bereich. Allerdings ist es noch lange nicht perfekt. Im ursprünglichen LDAP-Design gibt es einige Probleme, z. B. Mechanismen zur Gruppierung und einige Details von Such- und Änderungsvorgängen. LDAP würde eine umfassende Überprüfung verdienen, um die Probleme zu beheben und das Protokoll ins 21.

Jahrhundert zu bringen. Leider gab es seit Jahrzehnten kein größeres Update der LDAP-Spezifikationen.

Auch wenn LDAP seine Probleme hat, bleibt es dennoch ein nützliches Werkzeug. Die meisten Anbieter von Verzeichnisdiensten bieten proprietäre Lösungen für LDAP-Probleme. Viele Organisationen speichern Identitäten in LDAP-fähigen Datenspeichern. Es gibt viele Anwendungen, die LDAP unterstützen, hauptsächlich zur Zentralisierung der passwortbasierten Authentifizierung. LDAP ist nach wie vor ein wichtiges Protokoll im Bereich Identitäts- und Zugriffsmanagement. Um alle Anforderungen zu erfüllen, reicht LDAP allein allerdings langfristig nicht für das IAM im Kontext P20 aus.

Verzeichnisdienste sind Datenbanken

Verzeichnisdienste sind lediglich Datenbanken, die Informationen speichern, nicht mehr. Die für den Zugriff auf Verzeichnisdienst verwendeten Protokolle und APIs sind als Datenbankschnittstellen konzipiert. Das bedeutet, dass sie sich gut zum Speichern, Suchen und Abrufen von Daten eignen. Während die Daten von Benutzerkonten häufig Berechtigungsinformationen (Berechtigungen, Gruppen, Rollen usw.) enthalten, sind Identitätsspeicher für deren Auswertung nicht gut geeignet. D.h. der Verzeichnisdienst kann Informationen über die Berechtigungen eines Kontos bereitstellen, ist jedoch nicht darauf ausgelegt, eine Entscheidung darüber zu treffen, ob ein bestimmter Vorgang zugelassen oder verweigert wird. Und das ist noch nicht alles. Verzeichnisdienste enthalten keine Daten über Benutzersitzungen. Dies bedeutet, dass Verzeichnisdienste nicht wissen, ob der Benutzer derzeit angemeldet ist oder nicht. Viele Verzeichnisdienste werden für die grundlegende Authentifizierung und sogar Autorisierung verwendet. Allerdings sind die Verzeichnisdienste nicht dafür ausgelegt. Verzeichnisdienste bieten nur die grundlegendsten Funktionen. Es gibt Plug-in’s und Erweiterungen, die Teilfunktionen zur Unterstützung der Authentifizierung und Autorisierung bereitstellen. An den grundlegenden Gestaltungsprinzipien ändert sich dadurch jedoch nichts. Verzeichnisdienste sind Datenbanken, keine Authentifizierungs- oder Autorisierungsserver. Sie sollten als solche verwendet werden.

Viele Anwendungen verwenden jedoch Verzeichnisdienste, um die Authentifizierung basierend auf einem Kennwort zu zentralisieren. Tatsächlich ist dies eine gute und kosteneffiziente Möglichkeit, die passwortbasierte Authentifizierung zu zentralisieren. Dennoch sollte man sich darüber im Klaren sein, dass es sich hierbei um eine vorübergehende Lösung handelt. Es gibt viele Einschränkungen. Der richtige Weg hierfür ist die Verwendung eines Authentifizierungsservers anstelle eines Verzeichnisdienstes. Access Management (AM)-Technologien können dies ermöglichen.

Mythos eines einzelnen Verzeichnisdienstes

Ein gemeinsam genutzte Verzeichnisdienst erleichtert die Benutzerverwaltung. Dies ist jedoch keine vollständige Lösung und dieser Ansatz weist erhebliche Einschränkungen auf. Die Heterogenität der Informationssysteme macht es nahezu unmöglich, alle erforderlichen Daten in einem einzigen Verzeichnisdienst unterzubringen.

Das offensichtliche Problem ist das Fehlen einer einzigen, kohärenten Informationsquelle. Normalerweise gibt es für einen einzelnen Benutzer mehrere Informationsquellen.

Beispielsweise ist ein Personalsystem (HR-System) maßgeblich für die Existenz eines Benutzers im Unternehmen. Allerdings ist das HR-System in der Regel nicht für die Zuweisung von Mitarbeiterkennungen wie Benutzernamen verantwortlich. Es muss ein Algorithmus vorhanden sein, der die Eindeutigkeit des Benutzernamens gewährleistet, möglicherweise auch unter Einbeziehung aller aktuellen und früheren Mitarbeiter, Auftragnehmer und Partner. Darüber hinaus kann es zusätzliche Informationsquellen geben. Beispielsweise kann ein Managementinformationssystem für die Bestimmung der Benutzerrollen verantwortlich sein (z. B. in einer projektorientierten Organisationsstruktur). Das Bestandsverwaltungssystem kann für die Zuweisung der Telefonnummer an den Benutzer verantwortlich sein. Das Groupware-System kann eine maßgebliche Quelle für die E-Mail-Adresse und andere elektronische Kontaktdaten des Benutzers sein. Normalerweise gibt es 2 bis 20 Systeme, die maßgebliche Informationen für einen einzelnen Benutzer bereitstellen. Daher gibt es keine einfache Möglichkeit, die Daten in das Verzeichnissystem einzuspeisen und zu verwalten.

Im Kontext P20 kommt erschwerend hinzu, dass Benutzerdaten von min. 20 Behörden in einem gemeinsamen IAM nutzbar werden sollen, sodass es eine besonders große Vielfältigkeit der Quellsysteme gibt.

Und dann gibt es noch räumliche und technologische Barrieren. Viele komplexe Anwendungen benötigen eine lokale Benutzerdatenbank. Für einen effizienten Betrieb müssen sie die Kopien der Benutzerdatensätze in ihren eigenen Datenbanken speichern. Beispielsweise können große Abrechnungssysteme nicht effizient mit externen Daten arbeiten (z. B. weil eine relationale Datenbankverbindung erforderlich ist). Selbst wenn ein Verzeichnisdienst bereitgestellt wird, müssen diese Anwendungen daher weiterhin eine lokale Kopie der Identitätsdaten verwalten. Es scheint eine einfache Aufgabe zu sein, die Kopie mit den Verzeichnisdaten synchron zu halten. Es wird aber schnell komplex, wenn man die Details betrachtet. Darüber hinaus gibt es Altsysteme, die in der Regel überhaupt nicht auf die externen Daten zugreifen können (z. B. weil sie das LDAP-Protokoll überhaupt nicht unterstützen).

Einige Dienste müssen noch mehr Status als nur einen einfachen Datenbankeintrag behalten. Beispielsweise erstellen Dateiserver normalerweise Home-Verzeichnisse für Benutzer. Während die Kontoerstellung in der Regel nach Bedarf erfolgen kann (z. B. Benutzerverzeichnis bei der ersten Benutzeranmeldung erstellen), ist das Ändern und Löschen des Kontos wesentlich schwieriger. Der Verzeichnisserver wird das nicht tun.

Das vielleicht schmerzhafteste Problem ist die Komplexität der Richtlinien für die Zugriffskontrolle. Rollennamen und die Attributierung der Zugriffskontrolle haben möglicherweise nicht in allen Systemen die gleiche Bedeutung. Unterschiedliche Systeme verfügen in der Regel über unterschiedliche Algorithmen zur Autorisierung, die nicht miteinander kompatibel sind. Während dieses Problem mit anwendungsspezifischen Attributen für die Zugriffskontrolle gelöst werden kann, ist die Pflege dieser Attribute selten trivial. Wenn jede Anwendung über einen eigenen Satz von Attributen zur Steuerung der Richtlinien für die Zugriffskontrolle verfügt, bietet das zentralisierte Verzeichnis nur einen vernachlässigbaren Vorteil. Die Attribute können sich auch in den Anwendungen selbst befinden. Und genau so enden die meisten Einsätze. Verzeichnisdienste enthalten nur die Gruppen, Gruppen, die normalerweise ungefähr den RBAC-Rollen entsprechen. Selbst die LDAP-Standards selbst stellen in diesem Fall ein erhebliches Hindernis für die Interoperabilität dar. Es gibt mindestens drei oder vier unterschiedliche und inkompatible Spezifikationen für die Gruppendefinition in LDAP-

Verzeichnissen. Der Standard zur Verwaltung von LDAP-Gruppen ist überhaupt nicht ideal. Dies ist besonders problematisch, wenn große Gruppen verwaltet werden. Daher bieten viele Verzeichnisdienste ihre eigenen, nicht standardmäßigen Verbesserungen, was die Interoperabilität weiter erschwert. Doch selbst diese für diesen Dienst spezifischen Verbesserungen können in der Regel keine komplexen Richtlinien für die Zugriffskontrolle unterstützen. Daher werden Richtlinien für die Zugriffskontrolle und fein abgestufte Berechtigungen in der Regel nicht zentralisiert, sondern direkt in den Anwendungsdatenbanken verwaltet.

Der Ansatz eines zentralen Verzeichnisdiensten ist nur in sehr einfachen Umgebungen oder in nahezu vollständig homogenen Umgebungen möglich. In allen anderen Fällen besteht Bedarf, die Lösung durch andere Identitätsmanagement-Technologien zu ergänzen.

Dies bedeutet nicht, dass die Verzeichnisdienste oder andere gemeinsam genutzte Datenbanken nutzlos sind. Ganz im Gegenteil. Sie sind sehr nützlich, wenn sie richtig eingesetzt werden. Sie können lediglich nicht alleine in komplexen Umgebungen verwendet werden. Für den Aufbau einer Komplettlösung sind weitere Komponenten erforderlich.

Access Management

Während Verzeichnisdienste nicht für die Handhabung komplexer Authentifizierungen ausgelegt sind, sind Zugriffsverwaltungssysteme (Access-Management-Systeme, AM) genau dafür konzipiert. AM verwalten alle Arten der Authentifizierung und sogar einige Aspekte der Autorisierung. Das Prinzip aller Zutrittsverwaltungssysteme ist grundsätzlich dasselbe:

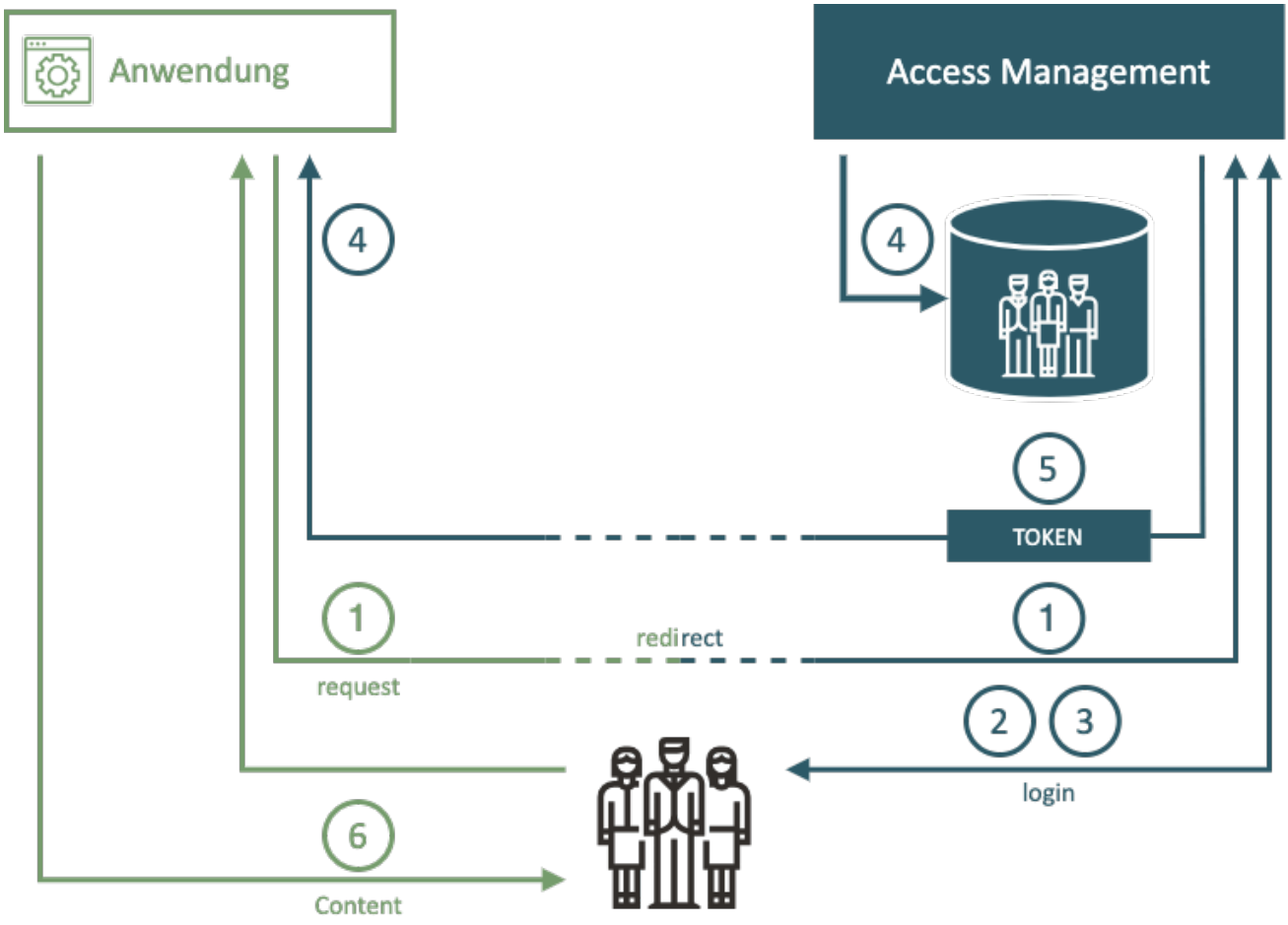

Abbildung 3: Grundprinzip Access Management

1. Das AM stellt die Schnittstelle zwischen dem Benutzer und der Anwendung dar. Dies kann durch verschiedene Mechanismen erfolgen. Die häufigste Methode besteht darin, dass die Anwendungen selbst den Benutzer zum AM umleiten, wenn keine Sitzung vorhanden ist.
2. Das AM fordert den Benutzer zur Eingabe des Benutzernamens und des Kennworts auf, interagiert mit dem Authentifizierungstoken, erstellt eine Abfrage und fordert zur Antwort auf oder leitet auf andere Weise den Authentifizierungsvorgang ein.
3. Der Benutzer gibt die Anmeldeinformationen ein.
4. Das AM prüft die Gültigkeit der Anmeldeinformationen und bewertet die Zugriffsrichtlinien.
5. Wenn der Zugriff erlaubt ist, leitet das AM den Benutzer zurück zur Anwendung. Die Umleitung enthält normalerweise ein Zugriffstoken: eine kleine Information, die der Anwendung mitteilt, dass der Benutzer authentifiziert ist.
6. Die Anwendung validiert das Token, erstellt eine lokale Sitzung und gewährt den Zugriff.

Nach diesem Vorgang arbeitet der Benutzer regulär mit der Anwendung. Lediglich der erste Zugriff erfolgt über den AM-Server. Dies ist wichtig für die Leistung und Dimensionierung des AM-Systems und wirkt sich auf die Funktionalität zur Verwaltung der Sitzungen aus.

Die Anwendungen müssen lediglich den Code bereitstellen, der in das Access-Management-System integriert werden kann. Abgesehen von diesem kleinen Integrationscode müssen Anwendungen überhaupt keinen Authentifizierungscode bereitstellen. Es ist das AM-System, das zur Eingabe des Passworts auffordert, nicht die Anwendung. Dies ist ein grundlegender Unterschied im Vergleich zu Mechanismen die auf Verzeichnisdiensten basieren. Im Falle eines Verzeichnisdienstes ist es die Anwendung, die zur Eingabe des Passworts auffordert. Im Falle eines AM erledigt der AM-Server alles. Für viele Anwendungen ist es unbedeutend, wie der Benutzer authentifiziert wurde. Sie müssen lediglich wissen, dass er authentifiziert wurde und dass die Authentifizierung stark genug war. Diese Funktion bringt eine sehr wünschenswerte Flexibilität für die gesamte Anwendungsinfrastruktur. Der Mechanismus zur Authentifizierung kann jederzeit geändert werden, ohne die Anwendungen zu beeinträchtigen.

Web Single Sign-On

Single-Sign-On-Systeme (SSO) ermöglichen es Benutzern, sich einmal zu authentifizieren und danach auf mehrere verschiedene Systeme zuzugreifen. Es gibt viele SSO-Systeme für Webanwendungen, es sieht jedoch so aus, als ob diese Systeme alle das gleiche grundlegende Funktionsprinzip verwenden. Dieser allgemeine Ablauf der Zugriffsverwaltung wird im Folgenden beschrieben:

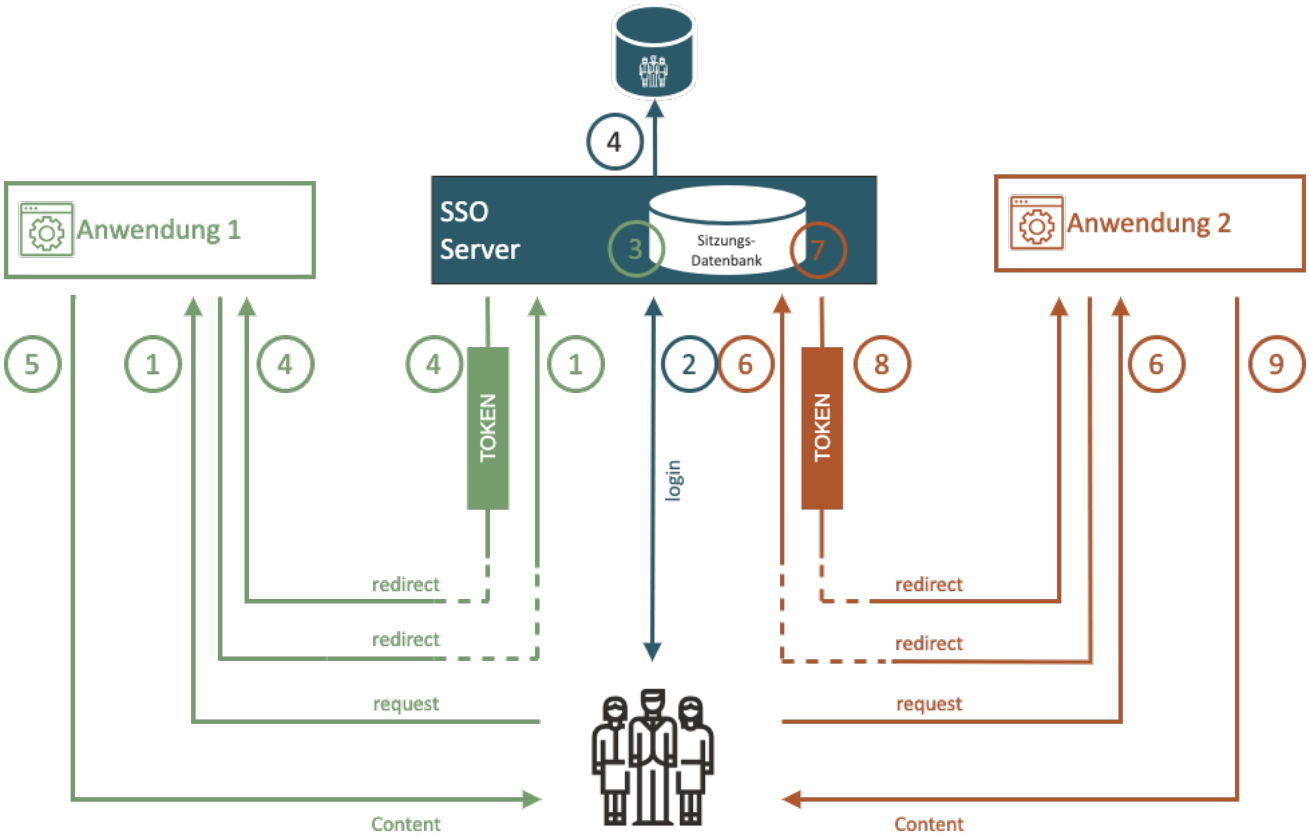

Abbildung 4: Grundprinzip Web Single-Sign-on

1. Benutzer fordert Zugriff auf Anwendung 1 an worauf diese den Benutzer zum Dienst SSO-Server der Zugriffssteuerung weiterleitet.
2. Die Zugriffssteuerung prüft, ob eine SSO-Sitzung vorhanden ist. Da noch keine Sitzung vorhanden ist, authentifiziert die Zugriffssteuerung den Benutzer.
3. Die Zugriffssteuerung richtet eine Sitzung (SSO-Sitzung) für den Webbrowser des Benutzer ein. Dies ist der entscheidende Teil des SSO-Mechanismus.
4. Der Benutzer wird zurück zur Anwendung 1 umgeleitet. Anwendung 1 baut normalerweise eine lokale Sitzung mit dem Benutzer auf.
5. Der Benutzer interagiert mit Anwendung 1.
6. Wenn der Benutzer versucht, auf Anwendung 2 zuzugreifen, leitet die Anwendung 2 den Benutzer an den Dienst (SSO-Server) der Zugriffssteuerung weiter.
7. Die Zugriffssteuerung prüft, ob eine SSO-Sitzung vorhanden ist. Da sich der Benutzer zuvor beim Dienst (SSO-Server) der Zugriffssteuerung authentifiziert hat, liegt eine gültige SSO-Sitzung vor.
8. Die Zugriffssteuerung prüft muss den Benutzer nicht erneut authentifizieren und leitet den Benutzer sofort zurück zu Anwendung 2.
9. Anwendung 2 richtet eine lokale Sitzung mit dem Benutzer ein und fährt normal fort.

Der Benutzer bemerkt in der Regel nicht, dass er beim Zugriff auf Anwendung 2 weitergeleitet wurde. Es gibt keine Interaktion zwischen den Weiterleitungen und den Umleitungen und die Verarbeitung auf dem Dienst der Zugriffssteuerung ist in der Regel sehr schnell. Es erweckt den Anschein, als ob der Benutzer die ganze Zeit bei der Anwendung 2 angemeldet war.

Autorisierung im Access Management

Die Anfrage eines Benutzers, der auf eine Anwendung zugreift, wird direkt oder indirekt über das AM weitergeleitet. Daher kann der AM-Server die Anfrage analysieren und bewerten, ob die Benutzeranfrage autorisiert ist oder nicht. Das ist in der Theorie richtig, in der Praxis ist die Situation wesentlich komplizierter.

Der AM-Server fängt normalerweise nur die erste Anforderung zum Zugriff auf die Anwendung ab, da das Abfangen aller Anforderungen Auswirkungen auf die Leistung hätte. Nach der ersten Anfrage richtete die Anwendung eine lokale Sitzung ein und fuhr mit dem Vorgang fort, ohne dass eine Kommunikation mit dem AM-Server stattfand. Daher kann der AM-Server die Autorisierung nur bei der ersten Anfrage auswerten. Dies bedeutet, dass nur sehr grobe Berechtigungsentscheidungen ausgewertet werden können. In der Praxis bedeutet dies normalerweise, dass der Access-Management-Server nur Entscheidungen zur Autorisierung nach dem Prinzip „Alles oder Nichts“ treffen kann: ob ein bestimmter Benutzer auf alle Teile einer bestimmten Anwendung zugreifen kann oder ob er überhaupt nicht auf die Anwendung zugreifen kann. Granulare Entscheidungen kann der AM-Server in der Regel alleine nicht treffen.

Einige AM-Systeme stellen Agenten bereit, die für Anwendungen bereitgestellt werden können und detailliertere Entscheidungen zur Autorisierung erzwingen. Solche Agenten verlassen sich häufig auf die HTTP-Kommunikation und treffen Entscheidungen auf der Grundlage der URLs, auf die der Benutzer zugreift. Dieser Ansatz mag in den 1990er-Jahren gut funktioniert haben, im Zeitalter von Single-Page-Webanwendungen und mobilen Anwendungen ist er jedoch nur sehr begrenzt anwendbar. In solchen Fällen wird die Autorisierung normalerweise auf *Dienste* und nicht auf *Anwendungen* angewendet.

Doch selbst die Anwendung der Autorisierung auf Service-Frontends löst das Problem nicht vollständig. Anspruchsvolle Anwendungen müssen häufig Entscheidungen zur Autorisierung auf der Grundlage eines Kontexts treffen, der in der Anfrage oder im Benutzerprofil überhaupt nicht verfügbar ist. Z.B. eine Banking-Anwendung kann eine Transaktion basierend auf der Summe früherer Transaktionen, die an diesem Tag getätigt wurden, zulassen oder ablehnen. Obwohl es möglich ist, alle Informationen zur Autorisierung im Benutzerprofil zu synchronisieren, ist dies normalerweise nicht wünschenswert. Es wäre eine großer Aufwand, diese Informationen aktuell und konsistent zu halten, ganz zu schweigen von Sicherheitsbedenken. Viele Schemata zur Autorisierung basieren auf einer bestimmten Geschäftslogik, die sich nur sehr schwer auf einem Autorisierungsserver zentralisieren lässt.

Weiterhin gibt es Implementierungsbeschränkungen. Theoretisch sollte das System für die Autorisierung nur Entscheidungen zum Zulassen/Ablehnen treffen. Dies reicht jedoch nicht aus, um eine effiziente Anwendung umzusetzen. Die Anwendung kann es sich nicht leisten, alle Objekte in der Datenbank aufzulisten, sie an den Autorisierungsserver zu übergeben und dann zu erkennen, dass der Autorisierungsserver den Zugriff auf fast alle von ihnen verweigert. Vor dem Suchvorgang muss die Autorisierung erfolgen und es müssen zusätzliche Suchfilter

angewendet werden. Das bedeutet, dass Mechanismen der Autorisierung tief in die Anwendungslogik integriert werden müssen. Dies schränkt die Anwendbarkeit zentralisierter Mechanismen für die Autorisierung erheblich ein.

AM-Systeme versprechen oft, die Autorisierung für alle Anwendungen zu vereinheitlichen und die Verwaltung unternehmensweiter Sicherheitsrichtlinien zu zentralisieren. Leider werden solch umfassende Versprechen selten eingehalten. Das AM kann theoretisch einige Anweisungen zur Autorisierung auswerten und durchsetzen. Dies kann bei Demonstrationen und sogar bei sehr einfachen Bereitstellungen gut funktionieren. Bei komplexen praktischen Einsätzen ist diese Fähigkeit jedoch äußerst begrenzt. Der überwiegende Teil der Entscheidungen zur Autorisierung wird von jeder einzelnen Anwendung getroffen und liegt völlig außerhalb der Reichweite eines AM-Systems.

SAML und OIDC

Einige Zugriffsverwaltungssysteme verwenden proprietäre Protokolle für die Kommunikation mit den Anwendungen und Agenten. Dies ist offensichtlich ein Interoperabilitätsproblem – insbesondere wenn die AM-Prinzipien in der Internetumgebung verwendet werden. Tatsächlich ist es das Internet, das die Standardisierung in diesem Bereich vorangetrieben hat.

Das erste weit verbreitete standardisierte Protokoll in diesem Bereich war Security Assertion Markup Language (SAML). Die ursprüngliche Absicht von SAML bestand darin, eine zonenübergreifende (Domain) Anmeldung und den Austausch von Identitätsdaten zwischen Organisationen im Internet zu ermöglichen. SAML ist sowohl ein Zugriffsverwaltungsprotokoll als auch ein Format zur Absicherung von Token. SAML ist recht komplex und basiert stark auf XML-Standards. Seine Spezifikationen sind lang, in mehrere Profile unterteilt, es gibt viele optionale Elemente und Funktionen und insgesamt ist SAML eine Reihe sehr umfangreicher und flexibler Mechanismen.

Der Hauptzweck von SAML ist die Übertragung von Identitätsinformationen zwischen Organisationen. Es gibt große SAML-basierte Verbände mit hunderten teilnehmenden Organisationen. Viele E-Government-Lösungen basieren auf SAML, es gibt große Partnernetzwerke, die auf SAML laufen, und insgesamt scheint SAML ein Erfolg zu sein. Dennoch war SAML ein Opfer seiner eigenen Flexibilität und Komplexität. Die neuesten Modetrends sind für SAML nicht sehr günstig. XML- und SOAP-basierte Webservice-Mechanismen geraten aus der Mode, was sich auf die Popularität von SAML auswirkt. Dies hat wahrscheinlich die Einführung anderer Zugriffsverwaltungsprotokolle motiviert.

Die neueste Mode bevorzugt RESTful-Dienste und einfachere Architekturansätze. All dies hat zur Entwicklung des OpenID Connect-Protokolls (OIDC) beigetragen. OIDC basiert auf viel einfacheren Mechanismen als SAML, verwendet jedoch dieselben Grundprinzipien. OIDC hat eine sehr bewegte Geschichte. Alles begann mit einer Reihe von Protokollen wie LID oder SXIP, die heute größtenteils vergessen sind. Es folgte die Entwicklung des OpenID-Protokolls, das noch sehr einfach war. OpenID erlangte insbesondere bei Anbietern von Internetdiensten große Aufmerksamkeit. Trotz seiner Einfachheit war OpenID nicht sehr ausgereift und stieß schnell an seine technologischen Grenzen. Es war offensichtlich, dass OpenID erheblich verbessert werden muss. Zu dieser Zeit gab es ein nahezu unabhängiges Protokoll namens OAuth, das für die Verwaltung organisationsübergreifender Autorisierungen konzipiert war. Dieses Protokoll wurde

stark weiterentwickelt, das dem ursprünglichen OAuth-Protokoll kaum noch ähnelt: OAuth2. Tatsächlich ist OAuth2 überhaupt kein Protokoll. Es handelt sich vielmehr um ein vage definiertes Framework zum Aufbau anderer Protokolle. Das OAuth2-Framework wurde verwendet, um ein organisationsübergreifendes Authentifizierungs- und Benutzerprofilprotokoll zu erstellen. Dieses neue Protokoll ist SAML viel ähnlicher als dem ursprünglichen OpenID, daher lag es nahe, es OIDC zu nennen.

Mittlerweile gibt es zwei Protokolle, die das gleiche Prinzip nutzen und fast das Gleiche tun. Das Prinzip wird im folgenden Diagramm veranschaulicht.

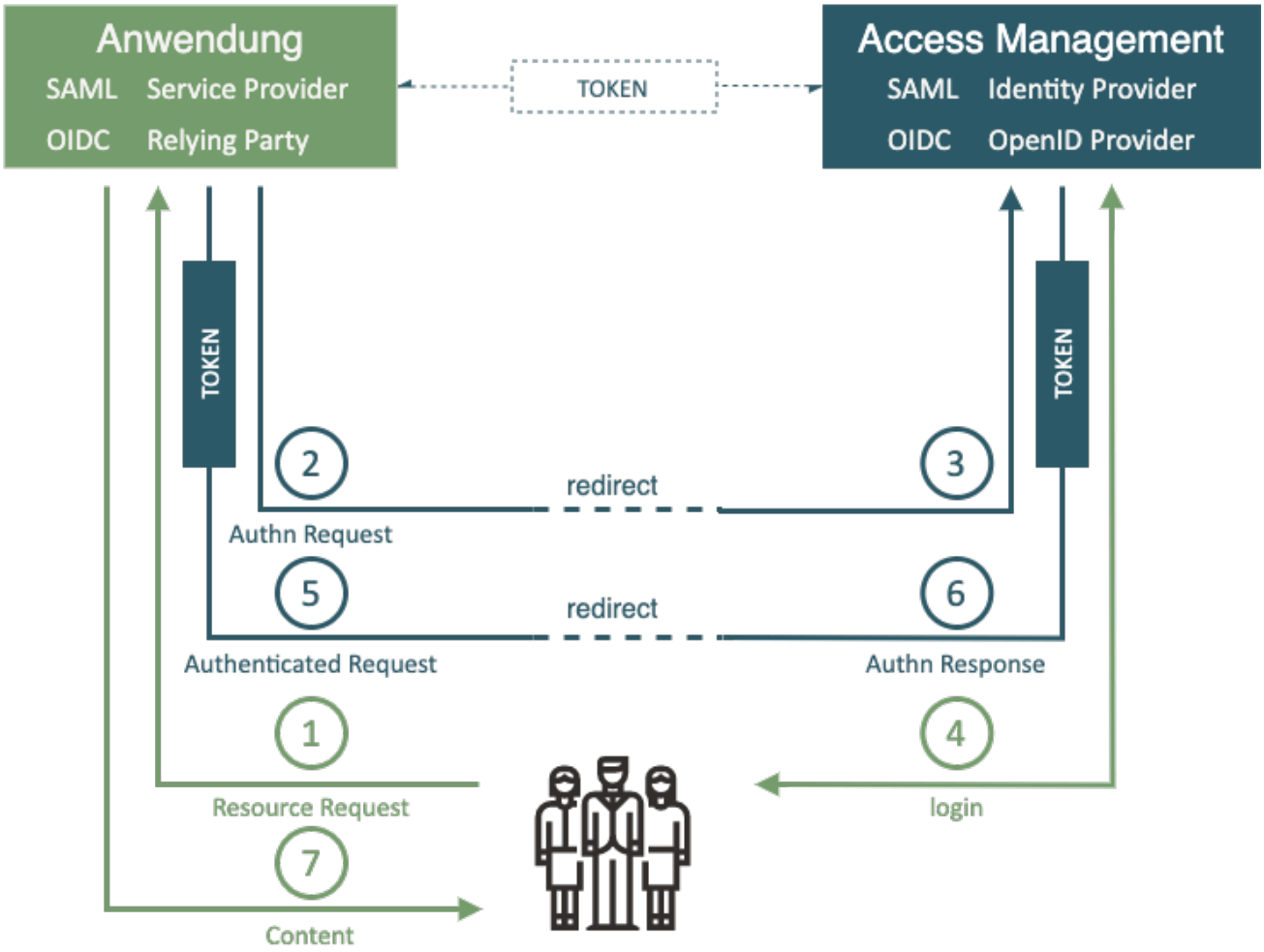

Abbildung 5: Token-basierte Authentifizierung

Die Interaktion läuft so ab:

1. Der Benutzer greift auf eine Ressource zu. Dies kann eine Webseite oder Webanwendung auf der Seite der Anwendung sein.
2. Die Seite der Anwendung verfügt nicht über eine gültige Sitzung für den Benutzer. Daher leitet es den Agenten des Benutzers (*Webbrowser*) zur Seite des AM weiter. Dieser Weiterleitung wird eine Authentifizierungsanfrage (*Authn Request*) hinzugefügt.
3. Der Agent des Benutzers folgt der Weiterleitung zur Seite des AM. Die Seite des AM erhält die Authentifizierungsanforderung und analysiert sie.

4. Wenn der Benutzer noch nicht auf der Seite des AM authentifiziert ist, erfolgt die Authentifizierung jetzt. Die Seite des Access Management fordert zur Eingabe des Benutzernamens, des Kennworts, des Zertifikats, des Einmalkennworts oder der von der Richtlinie geforderten Anmeldeinformationen auf.
5. Die Seite des AM leitet den Agenten des Benutzers zurück zur Seite der Anwendung. Die Seite des AM fügt der Umleitung eine Authentifizierungsantwort (*Authn Response*) hinzu. Der wichtigste Teil der Antwort ist ein Token. Der Token bestätigt direkt oder indirekt die Identität des Benutzers.
6. Die Seite der Anwendung analysiert die Authentifizierungsantwort und verarbeitet das Token. Das Token kann nur ein Verweis auf das echte Token sein (SAML-Artefakt) oder es kann ein Zugriffsschlüssel für einen anderen Dienst sein, der die Identität bereitstellt (OIDC UserInfo). In diesem Fall stellt die Seite der Anwendung eine weitere Anfrage (6a). Diese Anfrage erfolgt normalerweise direkt und verwendet keine Mechanismen zur Weiterleitung. Auf die eine oder andere Weise hat die Seite der Anwendung nun Ansprüche auf die Benutzeridentität.
7. Die Seite der Anwendung wertet die Identität aus, verarbeitet Berechtigungen usw. Zu diesem Zeitpunkt wird normalerweise eine lokale Sitzung mit dem Benutzer eingerichtet, um die Authentifizierungsumleitungen bei der nächsten Anfrage zu überspringen.
8. Die Seite der Anwendung stellt schließlich den Inhalt bereit.

In der folgenden Tabelle werden die in den Bereichen SAML und OIDC verwendeten Terminologie und Technologien verglichen.

	SAML	OpenID
Anwendung	Service Provider (SP)	Relying Party (RP)
Access Management	Identity Provider (IdP)	Identity Provider (IDP) oder OpenID Provider (OP)
Token	SAML Assertion (oder Artefakt)	ID Token, Access Token
Bestimmt für	Webanwendungen, Webservices Simple Object Access Protocol (SOAP)	Webanwendungen, Mobile Anwendungen, Webservices Representational State Transfer (REST)
Basiert auf	-	OAuth2
Kryptographie	XMLenc, XMLdsig	JOSE
Daten Präsentation	XML	JSON
Format	SAML	JSON Webtoken (JWT)

Tabelle 1: Access Management Terminologie

Aufmerksame Leser werden die Ähnlichkeit mit den webbasierten Mechanismen der Zugriffsverwaltung bemerken. Die Abläufe in der Darstellung wurden der Einfachheit halber auf die Abläufe für Webbrowser beschränkt. Sowohl SAML als auch OIDC haben eine breitere Anwendbarkeit als nur Webbrowser. Und die Unterschiede zwischen den Protokollen werden in diesen erweiterten Anwendungsfällen viel deutlicher. Aber der Webbrowser-Fall veranschaulicht gut die Prinzipien und Gemeinsamkeiten von SAML, OIDC und auch den einfachen Web-SSO-Systemen.

Der zentrale Unterschied zwischen SAML-, OIDC- und Web-SSO-Systemen ist der Verwendungszweck:

- SAML wurde für die Welt von Webanwendungen und SOAP-Webdiensten entwickelt. Zentralisierte (Einzel-IDP-)Szenarien werden gut gemeistert, es kann aber auch in dezentralen Verbünden funktionieren. SAML entfaltet seine Stärken beim Aufbau von großen dezentralen Föderationen.
- OIDC wurde hauptsächlich für die Verwendung mit sozialen Netzwerken und ähnlichen Internetdiensten entwickelt. Seine Philosophie ist immer noch irgendwie zentralisiert. Es funktioniert gut, wenn es einen starken Identitätsanbieter und viele vertrauende Parteien gibt. Technologisch integriert es sich besser in die RESTful-Dienste als SAML. Aktuelle Modetrends begünstigen OIDC.
- Web-SSO-Systeme sind für den Einsatz innerhalb einer einzelnen Organisation konzipiert. Dies ist ideal, um SSO zwischen mehreren kundenorientierten Anwendungen zu implementieren, sodass die Kunden keine Kenntnis darüber haben, dass sie mit vielen Anwendungen interagieren und nicht nur mit einer. Die

Web-SSO-Systeme sind nicht dafür ausgelegt, über Organisationsgrenzen hinweg zu funktionieren.

Auch wenn SAML und OIDC in erster Linie für den organisationsübergreifenden Einsatz konzipiert sind, ist es keine große Überraschung, sie innerhalb einer einzigen Organisation wieder zu finden. Die Verwendung eines offenen, standardisierten Protokolls anstelle eines proprietären Mechanismus bietet einen klaren Vorteil. Allerdings muss damit gerechnet werden, dass das SSO-System auf Basis von SAML oder OIDC einen etwas komplizierteren Aufbau haben wird als ein einfaches Web-SSO-System.

Kerberos, RADIUS und Enterprise SSO

Viele würden gerne glauben, dass heute alles auf Web-Technologien basiert und dass Nicht-Web-Mechanismen der Vergangenheit angehören. Dennoch gibt es immer noch Fälle, die nicht webbasiert sind und bei denen webbasierte SSO- und AM-Mechanismen nicht funktionieren. Innerhalb von Organisationen gibt es immer noch viele Legacy-Anwendungen. Anwendungen, die auf Rich Clients oder sogar zeichenbasierten Terminals basieren, sind immer noch zu finden. Und dann gibt es noch Netzwerkbetriebssysteme wie Windows und zahlreiche Unix-Varianten, es gibt Netzwerkzugangstechnologien wie VPN oder 802.1X etc. Es gibt immer noch viele Fälle, in denen webbasiertes Zugriffsmanagement und SSO einfach nicht funktionieren.

Das klassische Beispiel für ein Nicht-Web-SSO-System ist Kerberos. Das Protokoll entstand in den 1980er Jahren am MIT. Es handelt sich um ein Single-Sign-On-Protokoll für Betriebssysteme und Rich Clients, das auf symmetrischer Kryptographie basiert. Obwohl es sich um ein kryptografisches Protokoll handelt, ist es nicht allzu kompliziert zu verstehen und hat den Lauf der Zeit definitiv überstanden. Es wird bis heute vor allem zur Authentifizierung und SSO von Netzwerkbetriebssystemen eingesetzt. Es ist Teil der Windows-Netzwerkdomäne und oft die bevorzugte Lösung für die Authentifizierung von Unix-Servern. Die gravierendste Einschränkung von Kerberos liegt in der Verwendung der symmetrischen Kryptografie. Die Schwäche der symmetrischen Kryptographie ist die Verwaltung der Schlüssel. Die Kerberos-Schlüsselverwaltung kann recht schwierig sein, insbesondere wenn der Kerberos-Bereich sehr groß wird. Die Verwaltung der Schlüssel ist auch einer der Gründe, warum der Einsatz von Kerberos in organisationsübergreifenden Szenarien nicht sehr realistisch ist. Innerhalb einer geschlossenen Organisation ist Kerberos jedoch immer noch eine sehr nützliche Lösung.

Der größte Nachteil bei der Verwendung von Kerberos besteht darin, dass jede Anwendung und jeder Client „kerberisiert“ werden muss. Mit anderen Worten: Jeder, der an der Kerberos-Authentifizierung teilnehmen möchte, muss Kerberos-Unterstützung in seiner Software haben. Es gibt kerberisierte Versionen vieler Netzwerkdienstprogramme, sodass dies für Unix-basierte Netzwerke normalerweise kein Problem darstellt. Bei generischen Anwendungen stellt dies jedoch ein Problem dar. Es gibt eine gewisse Unterstützung für Kerberos in gängigen Webbrowsern, die oft als „SPNEGO“ bezeichnet wird. Diese Unterstützung ist jedoch normalerweise auf die Interoperabilität mit Windows-Domänen beschränkt. Daher ist Kerberos zwar immer noch nützlich für Betriebssystem-SSO, es handelt sich jedoch nicht um eine generische Lösung für alle Anwendungen.

Viele Netzwerkgeräte verwenden das RADIUS-Protokoll für das, was Netzwerktechniker „Authentifizierung, Autorisierung und Abrechnung“ (AAA) nennen. RADIUS ist jedoch ein Back-

End-Protokoll. Es kümmert sich nicht um Interaktionen mit Benutzern. Das Ziel von RADIUS besteht darin, dass das Netzwerkgerät (z. B. WLAN-Zugangspunkt, Router oder VPN-Gateway) Benutzeranmeldeinformationen validieren kann, die es als Teil eines anderen Protokolls erhalten hat. Der Client, der eine Verbindung zum VPN- oder WLAN-Netzwerk herstellt, weiß nichts über RADIUS. Daher ähnelt RADIUS dem LDAP-Protokoll und ist nicht wirklich eine Technologie des AM's.

Offensichtlich gibt es keine einfache und elegante Lösung, die SSO für alle Anwendungen bereitstellen kann. Dennoch erschien in den 1990er und frühen 2000er Jahren eine Technologie, die eine universelle SSO-Lösung für Unternehmen versprach. Es hieß „Enterprise Single Sign-On" (ESSO). Der ESSO-Ansatz bestand darin, auf jedem Client-Geräte Agenten zu installieren und zu verwenden. Der Agent erkennt, wenn ein Anmeldedialog auf dem Bildschirm erscheint, gibt den Benutzernamen und das Passwort ein und sendet den Dialog. Wenn der Agent schnell genug ist, nimmt der Benutzer den Dialog gar nicht wahr und es entsteht der Eindruck von Single Sign-On. Es gibt jedoch offensichtliche Nachteile. Den Agenten müssen alle Passwörter im Klartext bekannt sein. Es gibt ESSO-Varianten mit zufällig generierten Passwörtern oder sogar Einzelbenutzer-Passwörtern, die dieses Problem teilweise entschärfen. Der Nachteil besteht jedoch darin, dass ESSO auch in die Passwortverwaltung aller Anwendungen integriert werden muss, was nicht ganz einfach ist. Der gravierendste Nachteil von ESSO sind jedoch die Agenten. Diese funktionieren nur auf Arbeitsplätzen, die vom Unternehmen streng kontrolliert werden. Doch die Welt verändert sich, der Perimeter einer einzigen Organisation ist effektiv verschwunden und eine Organisation kann nicht wirklich alle Client-Geräte kontrollieren. Damit gehört auch ESSO mittlerweile weitgehend der Vergangenheit an.

Access Management und Daten

AM-Server und Identitätsanbieter müssen die Daten über Benutzer kennen, um ordnungsgemäß zu funktionieren. Aber das ist kompliziert. Der Zweck von Zugriffsverwaltungssystemen besteht darin, den Zugriff der Benutzer auf die Anwendungen zu verwalten. Dies bedeutet in der Regel die Verarbeitung von Authentifizierung, Autorisierung (teilweise), Überwachung des Zugriffs usw. Damit dies funktioniert, benötigt das AM-System Zugriff auf die Datenbank, in der die Benutzerdaten gespeichert sind. Es benötigt Zugriff auf Benutzernamen, Kennwörter und andere Anmeldeinformationen, Richtlinien zur Autorisierung, Attribute usw. Die AM-Systeme speichern diese Daten in der Regel nicht selbst. Sie stützen sich auf externe Datenbanken. In den meisten Fällen handelt es sich bei diesen Datenbanken um Verzeichnisdienste oder NoSQL-Datenbanken.Diese Datenbanken sind leichtgewichtig, hochverfügbar und extrem skalierbar. Das AM-System benötigt in der Regel nur einfache Attribute, daher stellen die eingeschränkten Möglichkeiten von Verzeichnissen und NoSQL-Datenbanken hier keinen limitierenden Faktor dar. Die Verbindung von Zugriffsverwaltung und leichtgewichtiger Datenbank ist eine offensichtliche und sehr intelligente Kombination.

Es gibt jedoch einen kritischen Punkt – insbesondere, wenn das AM-System auch als Single-Sign-On-Server verwendet wird. Die Daten im Verzeichnisdienst und die Daten in den Anwendungen müssen konsistent sein. Z.B. ist ein großes Problem, wenn ein Benutzer in mehreren Anwendungen unterschiedliche Benutzernamen hat. Mit welchem Benutzernamen soll er sich anmelden? Welcher Benutzername soll an die Anwendung übergeben werden? Es gibt Möglichkeiten, mit solchen Situationen umzugehen, aber das ist meist sehr umständlich und

teuer. Es ist viel einfacher, die Daten zu vereinheitlichen, bevor das AM-System bereitgestellt wird.

Auch wenn das „M“ in AM für „Management“ steht, verfügen typische AM-Systeme nur über sehr begrenzte Fähigkeiten Daten zu verwalten. Die AM-Systeme gehen in der Regel davon aus, dass die zugrunde liegende Datenbank bereits ordnungsgemäß verwaltet wird. Ein typisches Access-Management-System verfügt nur über eine sehr minimalistische Benutzeroberfläche zum Erstellen, Ändern und Löschen von Benutzerdatensätzen. Einige AM-Systeme verfügen möglicherweise über Self-Service-Funktionen (z.B. das Zurücksetzen von Passwörtern), aber selbst diese Funktionen sind normalerweise sehr eingeschränkt. Auch wenn AM auf der Tatsache beruht, dass die Daten im AM-Verzeichnisdienst und die Daten in Anwendungen konsistent sind, gibt es in der Regel keine Möglichkeit, die Daten mithilfe des AM-Systems selbst vollständig zu synchronisieren. Möglicherweise gibt es Methoden für bedarfsgesteuerte oder opportunistische Datenaktualisierungen, z.B. Erstellen eines Benutzerdatensatzes in der Datenbank, wenn sich der Benutzer zum ersten Mal anmeldet. Es gibt jedoch normalerweise keine Lösungen zum Löschen der Datensätze oder zum Aktualisieren der Datensätze inaktiver Benutzer.

Daher werden die AM-Systeme in der Regel nicht alleine eingesetzt. Der zugrunde liegende Verzeichnisdienst oder die NoSQL-Datenbank ist fast immer eine zwingende Voraussetzung für selbst die bescheidenste AM-Funktionalität. Aber damit das AM-System wirklich richtig funktioniert, braucht es etwas, um die Daten zu verwalten und zu synchronisieren. Zu diesem Zweck wird üblicherweise ein Identitätsmanagementsystem (IDM) verwendet. Tatsächlich wird in der Regel dringend empfohlen, das Verzeichnis und das IDM-System vor dem AM-System bereitzustellen. Ohne die Daten kann das AM-System nicht funktionieren. Und wenn das AM-System mit nicht ordnungsgemäß gepflegten Daten arbeitet, wird es nicht lange dauern, bis es ausfällt.

Vor- und Nachteile von Access-Management-Systemen

AM-Systeme bieten erhebliche Vorteile. Die meisten Merkmale sind durch das AM-Prinzip der zentralen Authentifizierung gegeben. Da die Authentifizierung von einem zentralen AM-Server durchgeführt wird, kann sie einfach kontrolliert und überprüft werden. Eine solche Zentralisierung kann genutzt werden, um Authentifizierungsrichtlinien konsistent anzuwenden – und sie bei Bedarf einfach zu ändern. Es ermöglicht auch eine bessere Nutzung einer Investition in Authentifizierungstechnologien. Z.B. Multifaktor- oder adaptive Authentifizierung kann recht teuer sein, wenn sie von jeder Anwendung implementiert werden muss. Wenn es jedoch im AM-Server implementiert ist, kann es von allen Anwendungen ohne weitere Investitionen wiederverwendet werden.

Allerdings gibt es auch Nachteile. Da die Zugriffsverwaltung zentralisiert ist, handelt es sich offensichtlich um einen *Single Point of Failure* (SPOF) . Wenn der AM-Server ausfällt, kann sich niemand mehr anmelden. Dies hat natürlich erhebliche Auswirkungen auf die Funktionalität aller Anwendungen. Daher müssen AM-Server hochverfügbar und skalierbar sein, was nicht immer eine leichte Aufgabe ist. Die Dimensionierung der AM-Server muss sehr sorgfältig erfolgen, da sie leicht zu Leistungsengpässen führen können. Der vielleicht gravierendste Nachteil sind jedoch die Gesamtkosten der Lösung. Die Kosten für den AM-Server selbst stellen in der Regel kein großes Problem dar. Aber der Server funktioniert nicht alleine. Der Server muss

in jede Anwendung integriert werden. Obwohl es Standardprotokolle gibt, ist die Integration alles andere als einfach. Die Unterstützung für AM-Standards und -Protokolle in den Anwendungen ist immer noch nicht universell. Insbesondere ältere Anwendungen müssen geändert werden, um ihr Authentifizierungssubsystem auf den AM-Server umzustellen. Dies ist oft so kostspielig, dass die Einführung von AM-Technologien oft nur auf eine Handvoll Anwendungen beschränkt ist. Obwohl neuere Anwendungen in der Regel AM-Protokolle in gewissem Umfang unterstützen, ist die Einrichtung immer noch keine leichte Aufgabe. Es gibt subtile Inkompatibilitäten und tückische Details, insbesondere wenn die Integration über die bloße Authentifizierung in die Verwaltung der Autorisierung- und Benutzerprofile übergeht.

Obwohl viele Organisationen den Einsatz eines AM-Systems als ersten Schritt im IAM-Projekt planen, ist dieser Ansatz selten erfolgreich. Das Projekt sieht in der Regel vor, 50–80 % der Anwendungen in die AM-Lösung zu integrieren. Die Realität ist jedoch, dass nur eine Handvoll Anwendungen problemlos in das AM-System integriert werden können. Die restlichen Anwendungen werden über ein IDM-System integriert, das nachträglich eilig zum Projekt hinzugefügt wird. Mit IDM zu beginnen und später einen Teil des Access Management hinzuzufügen, ist oft eine viel vernünftigere Strategie.

Mythos des homogenen Access Management

Es gibt mindestens zwei beliebte Zugriffsverwaltungsprotokolle für das Web. Es gibt riesige Identitätsföderationen, die auf SAML basieren. Cloud-Dienste und soziale Netzwerke nutzen in der Regel OIDC oder dessen Varianten. Es gibt Variationen und verwandte Protokolle, die für mobile Anwendungen und Dienste verwendet werden können. Dann gibt es noch andere SSO-Protokolle, die sich hauptsächlich auf die Verwendung innerhalb einer Organisation konzentrieren. Es gibt kein einziges Protokoll oder einzigenMechanismus, um alle Probleme in der Welt des AM zu lösen.

Darüber hinaus geht der Ansatz der Umleitung von AM-Systemen davon aus, dass der Benutzer über etwas verfügt, das Authentifizierungsaufforderungen anzeigen und Benutzerinteraktionen durchführen kann. Das ist normalerweise ein Webbrowser. Daher wird die ursprüngliche Variante der Zugriffsverwaltungsmechanismen vor allem bei herkömmlichen webbasierten Anwendungen eingesetzt. Variationen dieses Ansatzes sind auch auf Netzwerkdienste und Single-Page-Webanwendungen anwendbar. Dieser Ansatz ist jedoch in der Regel nicht direkt auf Anwendungen anwendbar, die Rich Clients, Betriebssystemauthentifizierung und ähnliche „traditionelle“ Anwendungen verwenden. Der Browser ist in diesen Fällen nicht die primäre Umgebung, die zur Durchführung der Authentifizierung verwendet werden kann. Es gibt einige Lösungen, die normalerweise auf eingebetteten Browsern basieren. Dies ändert jedoch nichts an der grundlegenden Tatsache, dass die AM-Technologien für diese Umgebungen nicht vollständig geeignet sind. Diese Anwendungen basieren normalerweise auf Kerberos als SSO-System oder lassen sich überhaupt nicht in ein SSO-System integrieren.

Typische IT-Umgebungen bestehen aus einer mannigfaltigen Mischung von Technologien und nicht alle davon sind vollständig webbasiert. Daher ist es ziemlich unwahrscheinlich, dass ein einziges AM-System auf alles angewendet werden kann, was in Ihrem Unternehmen eingesetzt wird. Die Authentifizierung ist sehr eng an die Benutzerinteraktion gebunden und hängt daher von der Art und Weise ab, wie der Benutzer mit der Anwendung interagiert. Da der Benutzer unterschiedliche Technologien verwendet, um mit der Webanwendung, der mobilen

Anwendung und dem Betriebssystem zu interagieren, ist es offensichtlich, dass auch die Authentifizierungs- und SSO-Methoden für diese Systeme unterschiedlich sein werden.

Daher ist damit zu rechnen, dass es in der Organisation mehrere AM- oder Single-Sign-On-Systeme geben wird, die jeweils eine eigene Technologieinsel bedienen.

> ***Wichtig***
>
> Jede Insel muss verwaltet werden.

Praktisches Access Management

Vereinheitlichendes AM-System, SSO, organisationsübergreifender Identitätsverbund, Social Login, universell einsetzbare 2-Faktor-Authentifizierung – das sind die Dinge, die gewünscht sind, wenn an IAM gedacht wird. Das sind alles absolut gültige Anforderungen. Allerdings hat alles seinen Preis. Es ist bekanntermaßen schwierig, die Kosten von AM-Lösungen abzuschätzen, da der Großteil der Kosten nicht in der AM-Software steckt. Ein großer Teil der Gesamtkosten steckt in vorhandenen Anwendungen, Diensten und Clients, die angebunden und dazu womöglich angepasst werden müssen. All dies muss bei der Planung eines Projekts berücksichtigt werden.

Obwohl AM das ist, was in der Regel beabsichtig ist, ist es in der Regel ratsam, **nicht** mit AM als ersten Schritt zu beginnen. Die Bereitstellung des AM-Systems weist viele Abhängigkeiten auf:

- einheitliche Benutzerdatenbank
- verwaltete und kontinuierlich synchronisierte Daten
- Anwendungen, die flexibel genug sind, um integriert zu werden.

Sofern eine IT-Infrastruktur nicht äußerst homogen und einfach ist, ist es sehr unwahrscheinlich, dass diese Abhängigkeiten bereits erfüllt sind. Daher ist mit ziemlicher Sicherheit davon auszugehen, dass ein Projekt, das zu Beginn versucht lediglich AM einzuführen, seine Ziele nicht erreichen wird. Wenn das AM-Projekt hingegen den richtigen Umfang und die richtige Planung hat und realistische Ziele verfolgt, sind die Erfolgsaussichten hoch.

Der vielleicht beste Weg, ein AM-Projekt zu bewerten, besteht darin, mehrere Fragen zu stellen:

- Brauche ich wirklich eine Zugriffsverwaltung für alle Anwendungen? Benötige ich eine 100-prozentige Abdeckung? Kann ich mir alle Kosten leisten? Vielleicht reicht es aus, nur ein paar Anwendungen zu integrieren, die die größten Probleme bereiten. Weiß ich, um welche Anwendungen es sich handelt? Weiß ich, was meine Benutzer während ihres Arbeitstages wirklich verwenden? Weiß ich, was sie brauchen?
- Was sind die wirklichen Sicherheitsvorteile der Bereitstellung eines AM? Werde ich die native Authentifizierung für die Anwendungen deaktivieren? Auch für Systemadministratoren? Was mache ich bei administrativen Notfällen (z. B. Systemwiederherstellung)? Wären Systemadministratoren immer noch in der Lage, das AM-System zu umgehen? Wenn ja, was ist dann der eigentliche Sicherheitsvorteil? Wenn nicht, wie sieht dann das Wiederherstellungsverfahren aus, falls das AM-System ausfällt?

- Brauche ich wirklich SSO für ältere und selten genutzte Anwendungen? Was ist hier das eigentliche Problem? Besteht das Problem darin, dass Benutzer das Passwort mehrmals am Tag eingeben? Oder liegt das eigentliche Problem darin, dass sie für verschiedene Anwendungen unterschiedliche Benutzernamen oder Passwörter eingeben müssen und die Anmeldeinformationen ständig vergessen? Vielleicht lösen einfache Datenbereinigung und Passwortverwaltung die schlimmsten Probleme und ich kann bei einem AM-Projekt eine Menge Geld sparen?

Die Technologien des AM sind der sichtbarste Teil des IAM-Systems. Aber es ist auch der teuerste Teil und der am schwierigsten einzurichtende und zu wartende Teil.

Identity Management

Identitätsmanagement (IDM) ist möglicherweise die am meisten übersehene und unterschätzte Technologie im gesamten Bereich des Identitäts- und Zugriffsmanagements (IAM). Dennoch ist IDM ein entscheidender Bestandteil fast jeder IAM-Lösung. IDM kann nahezu jeder Organisation erhebliche Vorteile bringen.

Identitätsmanagement ist genau das, was der Name sagt: Es geht um die Verwaltung von Identitäten. Dabei geht es um die Prozesse zur Erstellung von Active Directory-Konten und Postfächern für einen neuen Mitarbeiter. IDM richtet zu Beginn jedes Schuljahres Konten für Schüler ein. IDM ermöglicht es, bei einem Sicherheitsvorfall sofort jeglichen Zugriff durch einen verdächtigen Benutzer zu sperren. IDM kümmert sich während der Reorganisation um das Hinzufügen neuer Berechtigungen und das Entfernen alter Berechtigungen von Benutzern. IDM stellt sicher, dass alle Konten ordnungsgemäß deaktiviert werden, wenn der Mitarbeiter das Unternehmen verlässt. IDM richtet automatisch Berechtigungen für Schüler und Mitarbeiter ein, die für ihre Klassen bzw. Aufgaben geeignet sind. IDM zeichnet die Zugriffsrechte von Zeitarbeitern, Partnern, Supporttechnikern und allen Identitäten Dritter auf, die nicht im Personalsystem verwaltet werden. IDM automatisiert die Prozesse der Rollenanfrage und -genehmigung. IDM zeichnet jede Änderung der Benutzerrechte im Audit Trail auf. IDM regelt die jährliche Überprüfung von Rollen und Zugriffsrechten. IDM stellt sicher, dass die in den Anwendungen gespeicherten Kopien der Benutzerdaten synchronisiert und ordnungsgemäß verwaltet werden. IDM stellt sicher, dass die Daten gemäß den Datenschutzbestimmungen verwaltet werden. Und IDM erledigt viele andere Dinge, die für den effizienten und sicheren Betrieb jeder Organisation absolut unerlässlich sind.

Der größte Haken: Früher waren die IDM-Systeme teuer. Sehr teuer. Früher waren die IDM-Systeme so teuer, dass es selbst bei solch erheblichen und klaren Vorteilen sehr schwierig war, die Kosten zu rechtfertigen. Doch diese Zeit ist nun vorbei.

> ***Terminologie***
> Der Begriff Identitätsmanagement wird häufig für den gesamten Bereich des Identity & Access Managements (IAM) verwendet. Das ist etwas verwirrend, da Technologien wie Single Sign-On oder Access Management die Identitäten nicht wirklich verwalten. Solche Technologien verwalten den Zugriff auf die Anwendungen. Selbst Verzeichnisserver verwalten die Identitäten nicht selbst. Verzeichnisserver speichern die Identitäten und ermöglichen den Zugriff darauf. Tatsächlich gibt es einen ganzen Zweig von Technologien, die Identitäten verwalten. Diese Systeme sind für die Erstellung und Aufrechterhaltung von Identitäten verantwortlich. Diese werden manchmal als Identitätsbereitstellungs-, Management des Lebenszyklus von Identitäten oder Identitätsverwaltungssysteme bezeichnet. Aber angesichts des aktuellen Stands der Technik sind solche Namen tatsächlich eine Untertreibung. Diese Systeme können weit mehr als nur die Bereitstellung oder Verwaltung des Lebenszyklus von Identitäten. Nachfolgend werden diese Systeme einfach als Identitätsmanagementsysteme (IDM) bezeichnen. Wenn auf den gesamten Bereich Bezug genommen wird, der Zugriffsverwaltung, Verzeichnisdienste, Identitätsverwaltung und Governance umfasst, wird der Begriff Identity & Access Managements (IAM) verwendet.

Geschichte des Identitätsmanagements

In den 1990er Jahren gab es keine Technologie, die eindeutig als „Identitätsmanagement“ identifiziert werden konnte. Natürlich bestanden alle oben genannten Probleme schon fast seit den Anfängen der modernen Informatik. Es gab immer Lösungen für diese Probleme. In der Vergangenheit basierten die meisten dieser Lösungen auf Papier und Skripten. Das funktionierte ganz gut – bis in den 1990er- und 2000er-Jahren die große Welle der Systemintegration die Branche erfasste. Mit der Integration von Daten und Prozessen in einzelnen Anwendungen wurden die IDM-Probleme deutlich ausgeprägter. Manuelle papierbasierte Prozesse waren für das Zeitalter der Datenautobahnen einfach zu langsam. Die Skripte waren in einer Welt, in der alle paar Wochen neue Anwendungen bereitgestellt werden, zu schwierig zu verwalten. Die Bemühungen zur Integration von Identitäten begannen natürlich mit der damals modernsten Technologie zur Verwaltung von Identitäten: Verzeichnisdiensten. Wie bereits gezeigt, waren die Verzeichnisse nicht ganz die idealen Werkzeuge für diese Aufgabe. Die Verzeichnisse funktionierten nicht besonders gut in einer Umgebung, in der die Leute dachten, LDAP sei eine Art gefährliche Krankheit, in der Benutzernamen und Kennungen recht zufällig zugewiesen wurden und in der jede Anwendung darauf bestand, dass die einzigen maßgeblichen Daten diejenigen seien, die in ihrer eigenen Datenbank gespeichert seien.

Die Probleme der Integration motivierten die Einführung von IDM-Technologien Anfang der 2000er Jahre. Frühe IDM-Systeme waren lediglich Maschinen zur Datensynchronisation, die für den Betrieb mit Benutzern und Konten fest programmiert waren. Einige einfache Mechanismen zu Role-Based Access Control (RBAC) und Verwaltungsschnittstellen wurden etwas später hinzugefügt. Mitte der 2000er Jahre gab es mehrere nahezu vollständige IDM-Systeme der erste Generation. Diese Systeme waren in der Lage, Identitätsdaten zwischen Anwendungen zu synchronisieren und einige grundlegende Verwaltungsfunktionen bereitzustellen. Selbst eine so einfache Funktionalität war damals ein großer Erfolg. Die IDM-Systeme konnten die Daten ohne größere Änderungen an den Anwendungen synchronisieren und brachten daher die Kosten für die Integration auf ein angemessenes Niveau. Das Problem bestand darin, dass die Kosten für

die IDM-Systeme selbst recht hoch waren. Diese Systeme waren noch ziemlich primitiv, daher erforderte die Konfiguration und Anpassung durch eine sehr hoch spezialisierte Klasse von Ingenieuren. Diese Ingenieure waren fast ausschließlich bei IDM-Anbietern, großen Systemintegratoren und Beratungsunternehmen beschäftigt. Dies machte den Einsatz von IDM-Lösungen für viele mittlere und kleinere Organisationen unerschwinglich. Selbst große Organisationen setzten häufig IDM-Lösungen mit recht eingeschränkten Funktionen ein, um die Kosten akzeptabel zu halten.

Frühe IDM-Systeme haben sich im Laufe der Zeit weiterentwickelt und verbessert. Es gab Begleitprodukte für Identity Governance und Compliance, welche die Funktionalität erweiterten. Dennoch ist es oft nahezu unmöglich, die ursprüngliche Architektur eines Produkts zu ändern. Daher kämpfen viele IDM-Produkte der ersten Generation bis heute mit Einschränkungen des frühen Produktdesigns.

Jede IDM-Lösung muss mehr oder weniger individuell angepasst werden. Was normalerweise eher mehr als weniger bedeutet. Es muss oft sowohl das IDM-System sein, das sich anpasst, als auch die Anwendungen. Die Notwendigkeit, dass sich jede Anwendung an eine standardisierte IDM-Schnittstelle anpassen muss, bedeutet viele Änderungen an vielen verschiedenen Orten, Plattformen und Sprachen. Die Gesamtkosten aller notwendigen Umbauten belaufen sich auf eine enorme Summe. Ein solcher Ansatz wird von Zeit zu Zeit ausprobiert, scheitert jedoch fast immer. Es handelt sich nicht um einen praktischen Ansatz. Während es in der IT-Infrastruktur viele Anwendungen gibt, gibt es nur ein IDM-System. Wenn sich das IDM-System an Anwendungen und Geschäftsprozesse anpasst, sind die Änderungen in der Regel kleiner und werden alle an einem Ort und in einer einzigen Plattform implementiert.

Dabei sollte möglichst ein gesunder Mittelweg gefunden werden, um so wenig unterschiedliche Schnittstellen wie möglich einzusetzen, die sich so wenig wie möglich von Industriestandards entfernen wie möglich, um einen Vendor-Lock-in durch eine zu stark individualisierte zentrale Lösung zu verhindern.

> ***Wichtig***
>
> Das IDM-System muss anpassungsfähig sein.

Was ist dieses Identitätsmanagement überhaupt?

Identitätsmanagement ist ein einfacher Begriff, der eine sehr umfangreiche und umfassende Funktionalität bereitstellt. Es umfasst die Bereitstellung von Identitäten, einschließlich der erneuten Bereitstellung und die Aufhebung der Bereitstellung, Synchronisierung, Verwaltung der Organisationsstruktur, rollenbasierte Zugriffskontrolle, Datenkonsistenz, Genehmigungsprozesse, Audit und viele weitere Funktionen. Daher ist es ziemlich schwierig, anhand einer wörterbuchähnlichen Definition zu sagen, was Identitätsmanagement ist. Wir möchten lieber anhand einiger typischer Anwendungsszenarien beschreiben, was Identitätsmanagement ist.

Zum leichteren Verständnis ein einfaches Beispiel anhand des fiktiven Unternehmens namens ACME Laboratories Ltd.. Dieses Unternehmen hat einige tausend Mitarbeiter, ein großes Partnernetzwerk, Kunden und Lieferanten und viele anderen Dinge, die reale Unternehmen

haben. ACME Laboratories Ltd. verfügt zudem über ein IDM-System, das in seiner IT-Infrastruktur betrieben wird.

ACME Laboratories Ltd. stellt eine neue Mitarbeiterin namens Sophie ein. Sophie unterschreibt wenige Tage vor Beginn ihrer Anstellung einen Arbeitsvertrag. Der Vertrag wird von den ACME Laboratories Ltd. HR-Mitarbeitern in das HR-System eingegeben. Das IDM-System scannt regelmäßig die HR-Datensätze und entdeckt die Datensätze einer neuen Einstellung. Die IDM-Systeme ziehen den Datensatz ein und analysieren ihn. Das IDM-System übernimmt die Namen und die Mitarbeiternummer aus dem Personaldatensatz, generiert einen eindeutigen Benutzernamen und erstellt auf der Grundlage dieser Informationen einen Benutzerdatensatz im IDM-System. Das IDM-System erhält außerdem den Organisationskennung 11001 aus dem Personaldatensatz. Das IDM schaut in seinen Organisationsbaum und stellt fest, dass die Kennung 11001 zur Vertriebsabteilung gehört. Daher ordnet IDM den Benutzer automatisch der Vertriebsabteilung zu. Das IDM verarbeitet auch den Arbeitspositionscode S007 im Personaldatensatz. Die IDM-Richtlinien besagen, dass der Code S007 „Verkäufer“ bedeutet und dass jeder mit diesem Code automatisch die Rolle „Verkäufer“ erhalten sollte. Daher wird das IDM diese Rolle zuweisen. Da es sich um einen festangestellten Mitarbeiter handelt, erstellt das IDM automatisch ein Benutzerkonto im angeschlossenen Active Directory zusammen mit dem Firmenpostfach. Das Konto wird der Organisationseinheit Vertriebsabteilung zugeordnet. Die Rolle „Vertriebsagent“ berechtigt den Benutzer zu weiteren Privilegien. Daher wird das Active Directory-Konto automatisch Vertriebsgruppen und Verteilerlisten zugewiesen. Die Rolle ermöglicht auch den Zugriff auf das CRM-System, daher wird auch automatisch ein CRM-Konto erstellt und den entsprechenden Gruppen zugewiesen. All dies geschieht innerhalb weniger Sekunden, nachdem der neue HR-Datensatz erkannt wurde. Es geschieht alles automatisch.

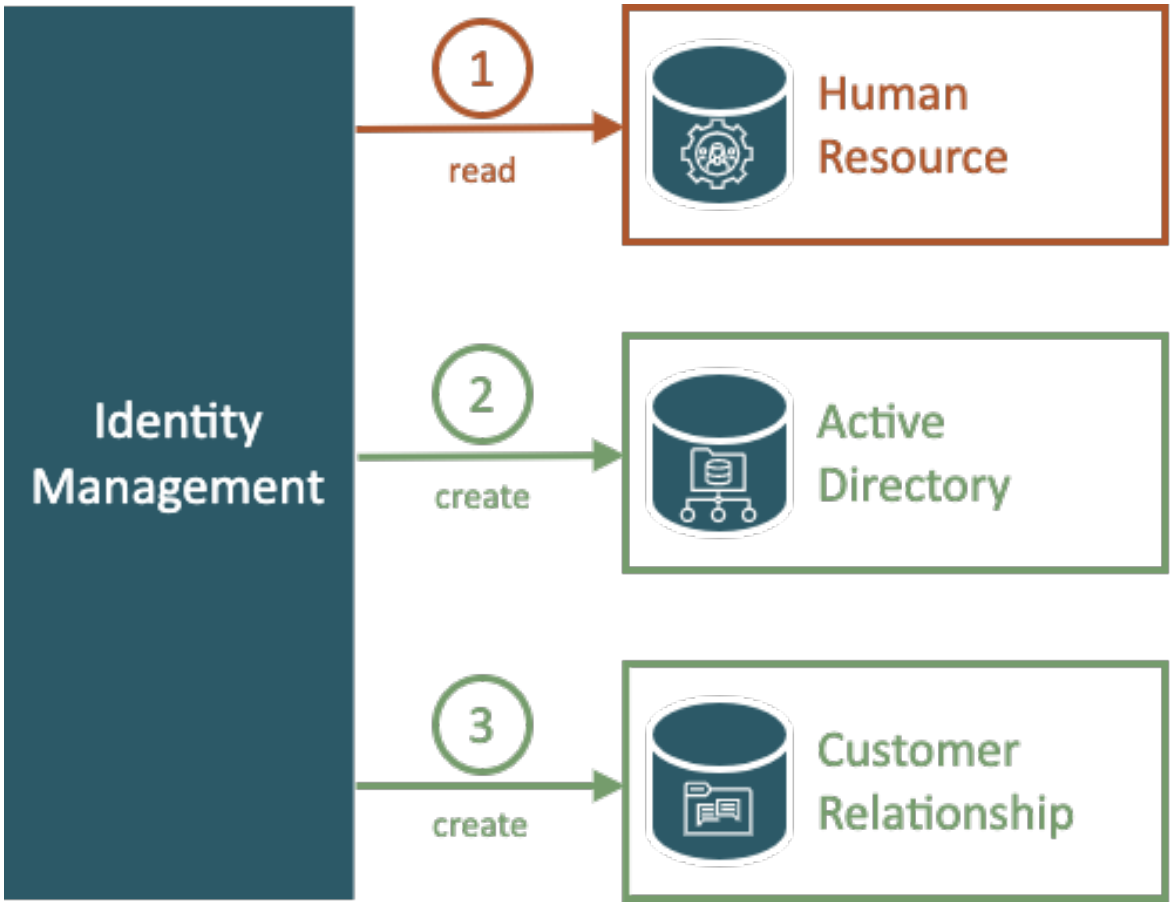

Abbildung 6: Grundprinzip Identity Management

Sophie beginnt ihre Karriere und ist eine wirklich effiziente Mitarbeiterin. Deshalb bekommt sie mehr Verantwortung. Sophie wird spezielle Marktanalysen erstellen, die auf vor Ort gesammelten empirischen Daten basieren. ACME Laboratories Ltd. ist ein wirklich flexibles Unternehmen, das ständig neue Wege erfindet, um Geschäftsabläufe effizienter zu gestalten. Deshalb haben sie diese Arbeitsposition speziell erfunden, um Sophies Fähigkeiten zu nutzen. Das bedeutet, dass es für Sophies neuen Job keinen Arbeitspositionscode gibt. Um ihre Arbeit effizient erledigen zu können, benötigt sie jedoch neue Berechtigungen im CRM-System. Glücklicherweise verfügt das ACME Laboratories Ltd. über ein flexibles IDM-System. Sophie kann sich beim IDM-System anmelden, die von ihr benötigten Berechtigungen auswählen und diese anfordern. Die Anfrage muss vom Manager von Sophie und auch vom Besitzer des CRM-Systems genehmigt werden. Sie erhalten eine Benachrichtigung über die Anfrage und können diese im IDM-System problemlos genehmigen oder ablehnen. Sobald die Anfrage genehmigt wurde, wird Sophies CRM-Konto automatisch den entsprechenden CRM-Gruppen zugewiesen. Sophie kann Minuten oder Stunden, nachdem sie die Berechtigungen angefordert hat, mit der Arbeit an ihrer Analyse beginnen.

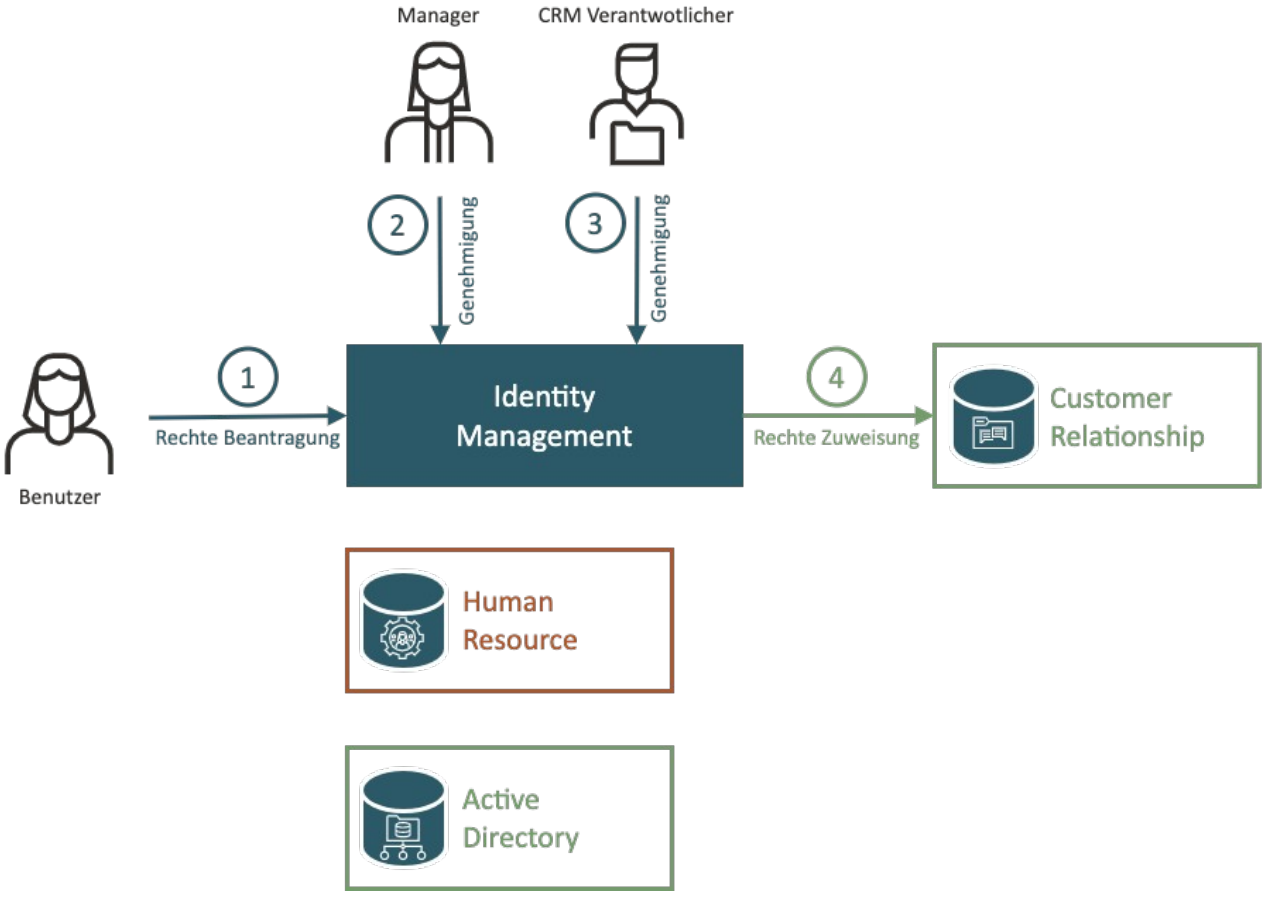

Abbildung 7: Identity Self Service

Eines Tages beschließt Sophie zu heiraten. Sie nimmt den Nachnamen ihres Ehepartners an. Sophie hat jetzt eine wirklich verantwortungsvolle Arbeitsposition, sie hat Konten in einem Dutzend Informationssystemen. Es ist keine leichte Aufgabe, ihren Namen in allen zu ändern, oder? Tatsächlich ist es sehr einfach, da ACME Laboratories Ltd. über ein IDM-System verfügt. Sophie informiert die Personalabteilung und die Personalabteilung ändert ihren Nachnamen im Personalsystem. Das IDM-System übernimmt die Änderung und gibt sie an alle betroffenen Systeme weiter. Sophie erhält automatisch eine neue E-Mail-Adresse mit ihrem neuen Nachnamen (unter Beibehaltung des alten als Alias). Sophie erhält eine Benachrichtigung, dass sie nun ihre neue E-Mail-Adresse verwenden kann.

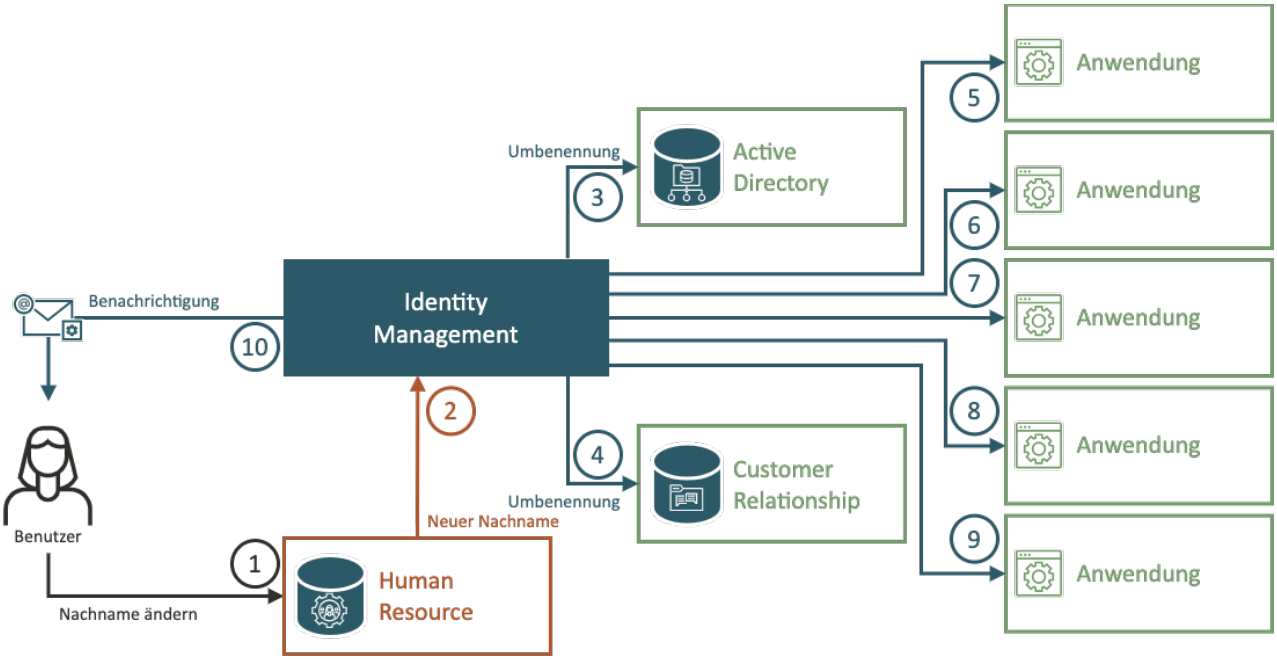

Abbildung 8: Identity Lifecycle

Später an diesem Tag erfährt Sophie, dass ihr Passwort bald abläuft. Das Ändern des Passworts in allen Anwendungen wäre eine riesige Aufgabe. Aber Sophie weiß, was zu tun ist. Sie meldet sich im IDM-System an und ändert dort ihr Passwort. Die Passwortänderung wird gemäß den vom IT-Sicherheitsbüro festgelegten Richtlinien automatisch an jedes betroffene System weitergegeben.

Im folgenden Monat passiert etwas Unerwartetes. Es gibt einen Sicherheitsvorfall. Das Sicherheitsbüro hat den Vorfall entdeckt und untersucht ihn nun. Es hat den Anschein, als wäre es ein Insider-Job gewesen. Die Sicherheitsverantwortlichen verwenden die Daten aus dem IDM-System, um ihre Ermittlungen auf Benutzer zu konzentrieren, die über Berechtigungen zum Zugriff auf betroffene Informationsbestände verfügten. Sie identifizieren Alfons als Hauptverdächtigen. Das Interessante ist, dass Alfons diese Privilegien überhaupt nicht haben sollte. Glücklicherweise führt das IDM-System auch ein Audit über jede Änderung von Berechtigungen. Dabei entdecken sie, dass es Alfons Kollege Oskar war, der Alfons diese Privilegien zuteilte. Beide Personen sollen befragt werden. Da dieser Vorfall jedoch sensible Vermögenswerte betrifft, müssen einige vorbeugende Maßnahmen ergriffen werden, bevor sich die Nachricht über den Vorfall verbreitet. Die Sicherheitsbeamten nutzen das IDM-System, um alle Konten von Alfons und Oskar sofort zu sperren. Es dauert nur wenige Sekunden, bis IDM diese Konten in allen betroffenen Anwendungen deaktiviert.

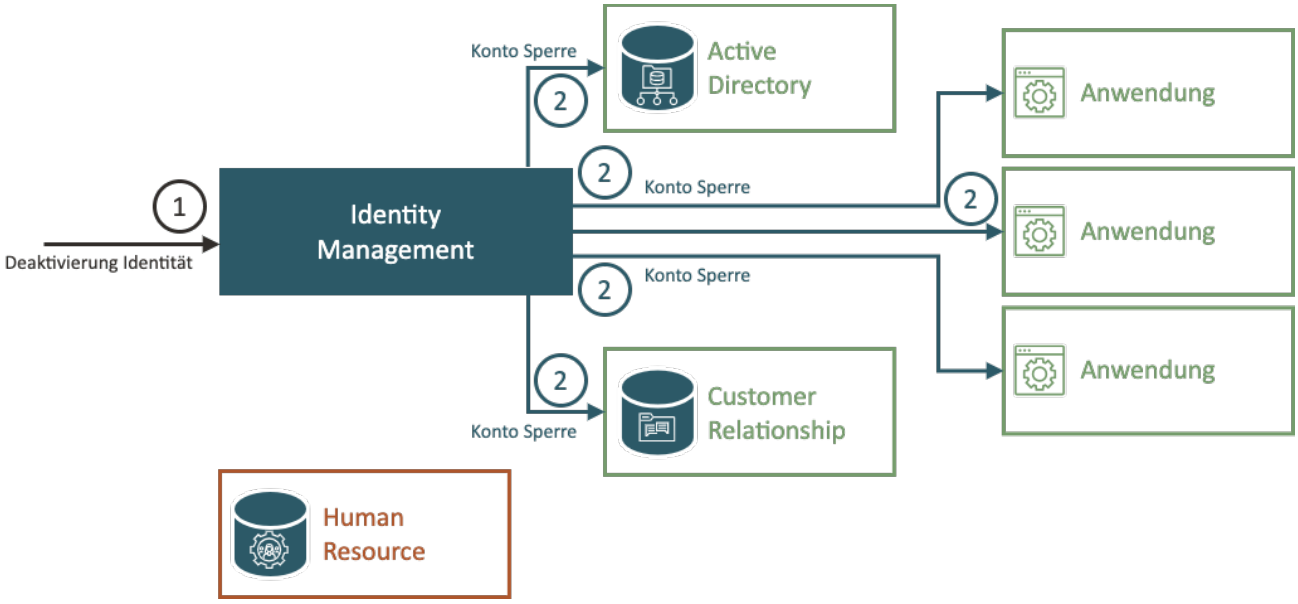

Abbildung 9: Identity Lockdown

Die Ermittlungen ergeben später, dass Oskar größtenteils unschuldig ist. Alfons missbrauchte Oskars Vertrauen und brachte ihn dazu, ihm die zusätzlichen Privilegien zu gewähren. Alfons missbrauchte die Privilegien, um an sensible Daten zu gelangen, und versuchte, diese zu verkaufen. Die Entscheidung lautet, dass Alfons das Unternehmen sofort verlassen muss, während Oskar bleiben darf. Da Oskar in diesem Fall jedoch ein schlechtes Urteilsvermögen gezeigt hat, werden seine Verantwortlichkeiten reduziert. Das IDM wird jetzt verwendet, um alle Konten von Alfons dauerhaft zu deaktivieren, Oskars Konten wieder zu aktivieren und auch sensible Privilegien zu widerrufen, die für Oskar als zu riskant gelten.

Wenige Monate später beschließt Oskar seinen Arbeitsvertrag bei ACME Laboratories Ltd. zu lösen und das Unternehmen ohne weiteren Ärger zu verlassen. Oskars Vertrag läuft Ende des Monats aus. Dieses Datum wird im HR-System erfasst und von dort in das IDM-System übernommen. Daher löscht das IDM-System um Mitternacht von Oskars letztem Arbeitstag automatisch alle Konten von Oskar.

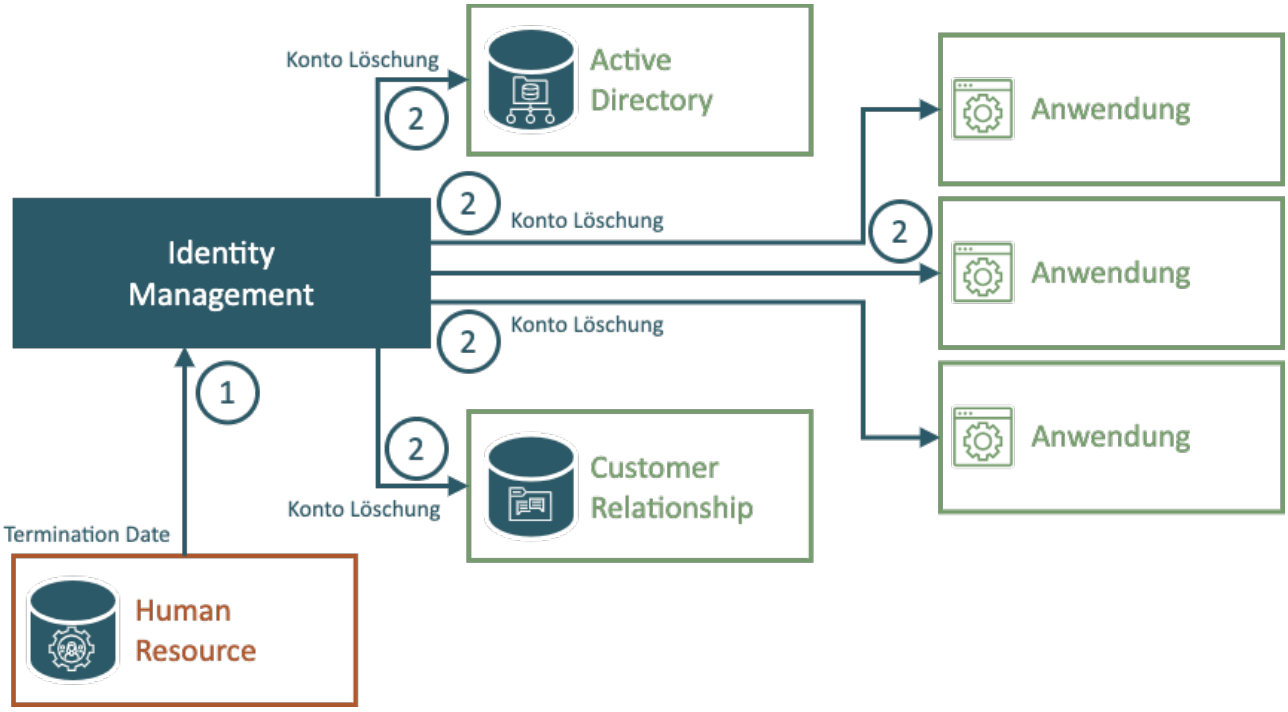

Abbildung 10: Identity Termination

Das Sicherheitsbüro hat den Sicherheitsvorfall professionell bearbeitet und das IDM-System lieferte wichtige Daten, um die Sicherheitsreaktion schnell und effizient zu gestalten. Aber das Team versucht immer, sich zu verbessern. Sie versuchen, aus dem Vorfall zu lernen und die Wahrscheinlichkeit zu verringern, dass so etwas noch einmal passiert. Das Team nutzt Daten aus dem IDM-System, um die den einzelnen Benutzern zugewiesenen Berechtigungen zu analysieren. Die übliche Aufgabe des IDM-Systems besteht darin, Konten in den Anwendungen zu erstellen und zu ändern. Das IDM-System nutzt jedoch die bidirektionale Kommunikation mit den Anwendungen. Die Analyse basiert auf realen Anwendungsdatenprozessen und wird durch das IDM-System vereinheitlicht: Was sind die echten Konten, zu welchem Benutzer gehören sie, welche Rollen haben sie, zu welchen Gruppen gehören sie usw. Das IDM-System kann Konten erkennen, die keinen eindeutigen Eigentümer haben. Das Sicherheitsteam entdeckt eine recht umfangreiche Sammlung von Testkonten, die offensichtlich beim letzten Ausfall des Rechenzentrums vor einem halben Jahr verwendet wurden. Offensichtlich haben die IT-Betriebsmitarbeiter diese Konten nach dem Ausfall vergessen. Das Sicherheitspersonal deaktiviert die Konten mithilfe der IDM-Tools und richtet einen automatisierten Prozess ein, um in Zukunft auf solche Konten zu achten.

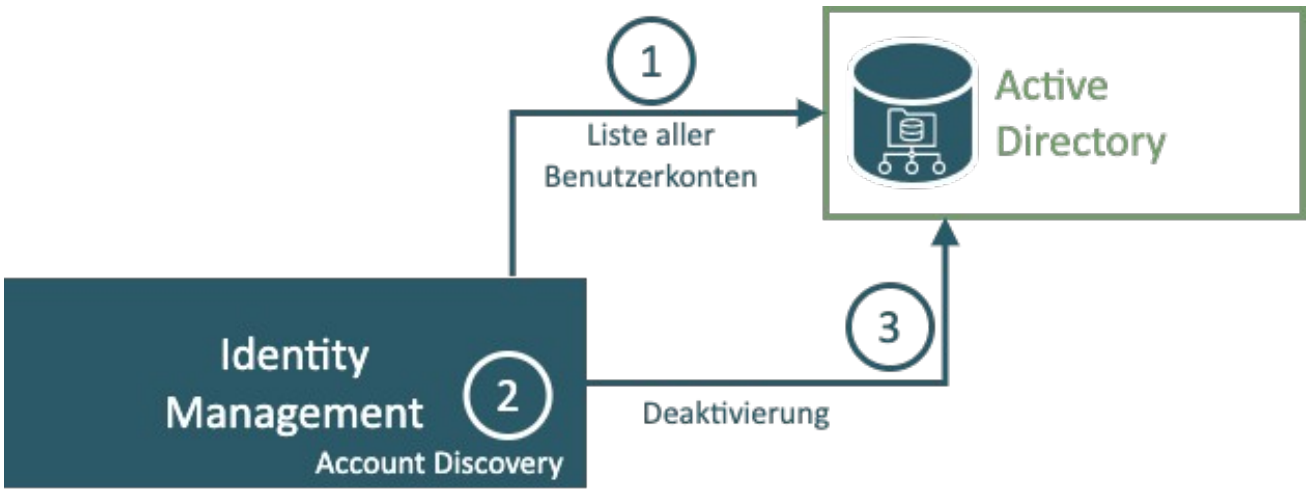

Abbildung 11: Identity Account Discovery

Wie funktioniert die Technologie?

Offensichtlich haben Identitätsmanagementsysteme viele Vorteile für Organisationen, Prozesse, Effizienz etc. Doch wie funktioniert das auf technologischer Ebene wirklich? Das Grundprinzip ist sehr einfach: Ein Identitätsmanagementsystem ist lediglich eine hochentwickelte Datensynchronisierungsmaschine.

Das Identitätsmanagementsystem übernimmt Daten aus den Quellsystemen, beispielsweise HR-Datenbanken. Es verarbeitet die Daten, ordnet die Werte zu und transformiert sie nach Bedarf. Es wird herausfinden, welche Datensätze neu sind. Die IDM-Engine führt einige (normalerweise recht komplexe) Verarbeitungen der Datensätze durch. Dazu gehören in der Regel Verarbeitungsrichtlinien wie RBAC, Organisationsrichtlinien, Passwortrichtlinien usw. Das Ergebnis dieser Verarbeitung ist die Erstellung oder Änderung von Benutzerkonten in anderen Systemen wie Active Directory, CRM-Systemen usw. Im Grunde geht es also darum, die Daten zu erhalten, sie zu ändern und zu verschieben. Das erscheint zunächst trivial, doch es kommt auf die Details an. Es ist die Art und Weise, wie das IDM-System die Daten sammelt, wie es die Daten verarbeitet und wie es die Änderungen weitergibt, die den entscheidenden Unterschied machen.

Identity Management Connectors

Ein Identitätsmanagementsystem muss eine Verbindung zu vielen verschiedenen Anwendungen, Datenbanken und Informationssystemen herstellen. Eine typische IDM-Bereitstellung umfasst Dutzende oder sogar Hunderte solcher Verbindungen. Daher ist die einfache Verbindung des IDM-Systems mit seiner Umgebung eine seiner wesentlichen Eigenschaften.

Aktuelle IDM-Systeme nutzen Konnektoren zur Kommunikation mit allen umliegenden Systemen. Diese Konnektoren basieren auf ähnlichen Prinzipien wie Datenbanktreiber. An einem Ende gibt es eine einheitliche Connector-Schnittstelle, die die Daten aller Systeme mit demselben „Protokoll“ präsentiert. Am Ende des Connectors befindet sich das native Protokoll, das die Anwendung unterstützt. Daher gibt es Konnektoren für LDAP und verschiedene LDAP-Varianten, SQL-Protokolle und -Dialekte, Konnektoren, die dateibasiert sind, Konnektoren, die Webdienste oder REST-Dienste aufrufen und so weiter. Jedes etwas fortgeschrittene IDM-System verfügt über Dutzende verschiedener Verbindungen.

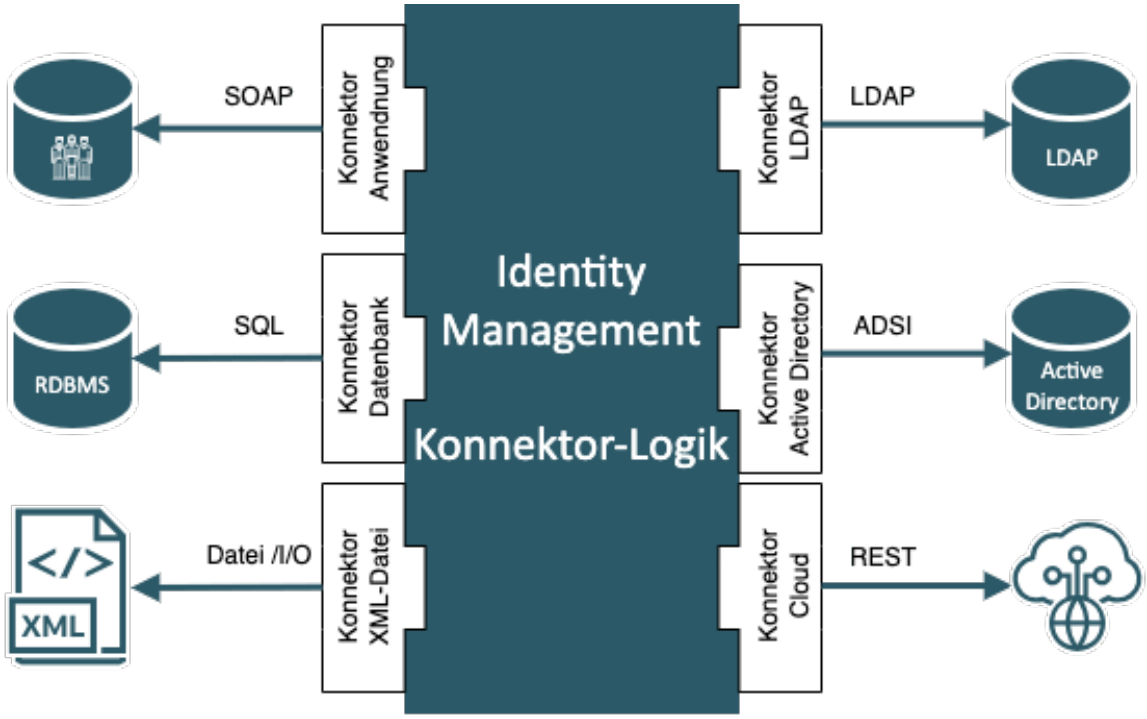

Abbildung 12: Identity Konnektor Architektur

Ein Connector ist normalerweise ein relativ einfacher Code. Die Hauptaufgabe eines Connectors besteht darin, Kommunikationsprotokolle anzupassen. Daher übersetzt der LDAP-Connector die LDAP-Protokollnachrichten in Daten, die über eine gemeinsame Connector-Schnittstelle dargestellt werden. Der SQL-Connector macht dasselbe mit SQL-basierten Protokollen. Der Connector interpretiert auch die vom IDM-System auf der gemeinsamen Connector-Schnittstelle aufgerufenen Vorgänge. Daher führt das LDAP-Protokoll den „Erstellen"-Vorgang aus, indem es eine LDAP-„Hinzufügen"-Nachricht an den LDAP-Server sendet und die Antwort analysiert. Konnektoren implementieren normalerweise den grundlegenden Satz von CRUD-Vorgängen (Create-Read-Update-Delete). Daher ist ein typischer Connector ein recht einfacher Code. Das IDM-System muss sich nicht um die Kommunikationsdetails kümmern. Der Kern des IDM-Systems kann so aufgebaut sein, dass er sich auf die generische Identitätsverwaltungslogik konzentriert, die für sich genommen normalerweise recht komplex ist. Daher ist jede Vereinfachung, die die Konnektoren bieten, mehr als willkommen.

Konnektoren greifen in der Regel auf externe Schnittstellen von Quell- und Zielsystemen zu. Es ist selbstverständlich, dass die Entwickler von Connector öffentliche, gut dokumentierte und auf offenen Standards basierende Schnittstellen wählen. Viele neuere Systeme verfügen über solche Schnittstellen. Es gibt jedoch berüchtigte Fälle, in denen die Bereitstellung einer solchen Schnittstelle verweigert wird. Trotzdem gibt es fast immer eine Möglichkeit, einen Konnektor zu bauen. Der Konnektor kann Datensätze direkt in der Anwendungsdatenbank erstellen. Oder es führt eine Datenbankroutine aus. Oder es führt möglicherweise ein Befehlszeilentool zur Kontoverwaltung aus. Es gibt fast immer eine Möglichkeit, das zu tun, was der Konnektor tun muss.

Obwohl der auf Konnektoren basierende Ansatz weit verbreitet ist, verwenden einige ältere IDM-Systeme keine Konnektoren. Einige IDM-Produkte verwenden Agenten anstelle von Konnektoren. Der Agent erledigt eine ähnliche Aufgabe wie der Connector. Der Agent ist jedoch nicht Teil der IDM-Systeminstanz. Agenten werden in jeder verbundenen Anwendung installiert und kommunizieren über ein Remote-Netzwerkprotokoll mit dem IDM-System. Das ist eine

große Belastung. Die Agenten müssen überall installiert werden. Und dann müssen sie gewartet, aufgerüstet werden, es kann zu subtilen Inkompatibilitäten kommen und so weiter. Außerdem kann die Ausführung eines Codes eines Drittanbieter in jeder Anwendung ein großes Sicherheitsrisiko darstellen. Insgesamt sind die agentenbasierten Systeme zu umständlich (und zu kostspielig) im Betrieb. Die ganze Agenten-Idee hat ihren Ursprung irgendwo in unserer digitalen Vergangenheit, als Anwendungen und Datenbanken noch keine nativen Remote-Schnittstellen unterstützten. In einer solchen Situation sind die Agenten offensichtlich besser als die Konnektoren. Selbst alte Anwendungen verfügen heute über eine Möglichkeit, Identitäten über eine Remote-Schnittstelle zu verwalten. Dabei handelt es sich in der Regel um einen Web- oder REST-Dienst, auf den über einen Connector einfach zugegriffen werden kann. Aber selbst wenn die Anwendung nur eine Befehlszeilenschnittstelle oder ein interaktives Terminal bereitstellt, gibt es Konnektoren, die dies ausreichend gut bewältigen können. Daher gelten die agentenbasierten Systeme heute allgemein als veraltet.

Identity Provisioning

Die Bereitstellung von Identitäten ist möglicherweise die am häufigsten verwendete Funktion in jedem IDM-System. Im allgemeinen Sinne bedeutet Bereitstellung von Identitäten die Pflege von Benutzerkonten in Anwendungen, Datenbanken und anderen Zielsystemen. Dazu gehören die Erstellung des Benutzerkontos, verschiedene Änderungen während dessen Lebensdauer und die dauerhafte Deaktivierung oder Löschung an dessen Ende. Das IDM-System verwendet Konnektoren, um die Benutzerkonten zu manipulieren. Und tatsächlich können gute IDM-Systeme weit mehr als nur Benutzerkonten verwalten.

Synchronisierung und Reconciliation

Die Bereitstellung von Identitäten ist möglicherweise die wichtigste Funktion eines IDM-Systems. Aber wenn ein IDM-System nur die Bereitstellung und nichts anderes übernehmen würde, wäre das schnell ein völliger Misserfolg. Es reicht nicht aus, ein Konto zu erstellen, wenn ein neuer Mitarbeiter eingestellt wird, oder dieses Konto zu löschen, wenn ein Mitarbeiter ausscheidet. Die Realität wirkt auf mysteriöse Weise und kann in kürzester Zeit leicht ein großes Durcheinander anrichten. Möglicherweise ist eine der Anwendungen ausgefallen und die Daten wurden aus einem Backup wiederhergestellt. Ein Konto, das vor einigen Stunden gelöscht wurde, wird unerwartet wiederbelebt. Es bleibt dort, lebendig, unkontrolliert und gefährlich. Möglicherweise hat ein Administrator manuell ein Konto für einen neuen Assistenten erstellt, während die Mitarbeiter der Personalabteilung noch mit der Bearbeitung der Unterlagen beschäftigt waren. Wenn der Datensatz schließlich im HR-System ankommt und verarbeitet wird, stellt das IDM-System fest, dass bereits ein in Konflikt stehendes Konto vorhanden ist, und stoppt einfach mit einem Fehler. Möglicherweise werden einige (hundert) Konten versehentlich von einem Junior-Systemadministrator gelöscht, der eine innovative Systemadministrationsroutine ausprobiert. Es gibt einfach zu viele Möglichkeiten, wie etwas schiefgehen kann. Und tatsächlich gehen sie überraschend oft schief. Für ein IDM-System reicht es nicht aus, Dinge einfach einzurichten und dann zu vergessen. Eines der wichtigsten Merkmale eines jeden IDM-Systems besteht darin, sicherzustellen, dass alles richtig ist und auch immer richtig bleibt. Beim Identitätsmanagement geht es um die kontinuierliche Pflege der Identitäten. Ohne diese Kontinuität ist das gesamte IDM-System nahezu nutzlos.

Der Trick, die Daten in Ordnung zu halten, besteht darin, zu wissen, wann die Daten nicht mehr synchron sind.

> ***Wichtig***
>
> Das IDM-System muss erkennen, wenn sich die Daten in den Anwendungsdatenbanken ändern.

Wenn ein IDM-System erkennt, dass eine Änderung stattgefunden hat, ist es nicht so schwierig, auf die Änderung zu reagieren und sie zu beheben. Die geheime Zutat ist die Fähigkeit, Veränderungen zu erkennen. Aber da gibt es ein kleines Problem, nicht wahr? Wir können nicht erwarten, dass die Anwendung bei jeder Änderung eine Benachrichtigung an das IDM-System sendet. Wir möchten die Anwendungen nicht ändern, da sonst die IDM-Bereitstellung unerschwinglich teuer wird. Die Anwendung muss passiv sein und das IDM-System muss aktiv sein. Glücklicherweise gibt es mehrere Möglichkeiten, dies zu tun.

Einige Anwendungen verfolgen die Änderungen bereits. Einige Datenbanken zeichnen für jede Zeile einen Zeitstempel der letzten Änderung auf. Einige Verzeichnisserver zeichnen zum Zweck der Datenreplikation die letzten Änderungen auf. Solche Metadaten können vom IDM-System verwendet werden. Das IDM-System kann die Zeitstempel oder Replikationsprotokolle regelmäßig auf neue Änderungen durchsuchen. Wenn das IDM eine Änderung erkennt, kann es die geänderten Objekte abrufen und basierend auf seinen Richtlinien auf die Änderung reagieren. Die auf Metadaten basierende Suche nach Änderungen ist in der Regel sehr effizient und kann daher alle paar Minuten durchgeführt werden. Dadurch kann nahezu in Echtzeit auf die Änderung reagiert werden. Diese Methode hat in verschiedenen IDM-Systemen viele Namen. Man nennt es „Live-Synchronisation“, „aktive Synchronisation“ oder einfach nur „Synchronisation“.

Auch wenn die Anwendung keine guten Metadaten verwaltet, die eine Änderungserkennung nahezu in Echtzeit ermöglichen, gibt es dennoch eine sehr einfache Möglichkeit, die für fast jedes System funktioniert. Das IDM-System ruft die Liste aller Konten in der Anwendung ab. Dann vergleicht es diese Liste mit der Liste der Konten, die dort vorhanden sein sollten. Daher vergleicht es die Realität (was vorhanden ist) mit der Politik (was vorhanden sein sollte). Das IDM-System kann auf eventuelle Unstimmigkeiten reagieren und diese beheben. Diese Methode wird als Abstimmung bezeichnet.

Das Auflisten aller Konten und deren Bearbeitung scheint eine unkomplizierte Aufgabe zu sein. Es kann jedoch extrem langsam sein, wenn die Anzahl der Konten hoch ist und die Richtlinien komplex sind. Es kann zwischen einigen Minuten und einigen Tagen dauern. Daher kann es nicht häufig ausgeführt werden. Dies einmal pro Tag auszuführen, ist nur für kleine und einfache Systeme möglich. Die Durchführung einmal pro Woche (am Wochenende) ist eine gängigere Praxis. Viele Systeme können es sich jedoch nicht leisten, es häufiger als einmal pro Monat auszuführen.

Es gibt auch andere Methoden. Am häufigsten werden jedoch Synchronisierungen und Abstimmungen verwendet. Der Nachteil der Synchronisierung besteht darin, dass sie nicht völlig zuverlässig ist. Dem IDM-System fehlen möglicherweise einige Änderungen, z.B aufgrund des Ablaufs des Änderungsprotokolls, nicht synchronisierter Systemzeiten oder verschiedener anderer Gründe. Andererseits ist die Abstimmung meist zuverlässig. Aber es ist eine sehr

anspruchsvolle Aufgabe. Daher werden diese beiden Methoden oft zusammen verwendet. Die Synchronisierung läuft ständig und verarbeitet die überwiegende Mehrheit der Änderungen. Der Abgleich wird wöchentlich oder monatlich ausgeführt und fungiert als Sicherheitsnetz, um die Änderungen abzufangen, die während der Synchronisierung möglicherweise unbemerkt geblieben sind.

Rollenbasierte Zugriffskontrolle

Die individuelle Verwaltung der Berechtigungen für jeden Benutzer ist nur dann sinnvoll, wenn die Anzahl der Benutzer sehr gering ist. Die individuelle Verwaltung von Berechtigungen wird bei nur wenigen Hundert Benutzern sehr schwierig. Bei mehr als tausend Benutzern wird eine solche Verwaltung meist zu einer unerträglichen Belastung. Die individuelle Verwaltung von Berechtigungen ist nicht nur ein riesiger Arbeitsaufwand, sondern auch eine recht fehleranfällige Routine. Dies ist seit Jahrzehnten bekannt. Daher vereinheitlichten viele Systeme gängige Kombinationen von Berechtigungen in Rollen und das Konzept der rollenbasierten Zugriffskontrolle (Role-Based Access Control, RBAC) war geboren. Die Rollen repräsentieren häufig Arbeitspositionen oder Verantwortlichkeiten, die viel näher am „Geschäft“ liegen als technische Berechtigungen. Eine Rolle kann die Konzepte eines Bankangestellten, eines Website-Administrators oder eines Vertriebsleiters widerspiegeln. Der Benutzer hat eine Rolle, die Rolle enthält Berechtigungen, Berechtigungen dienen der Autorisierung – das ist das Grundprinzip von rollenbasierter Zugriffskontrolle. Die Berechtigungen auf niedriger Ebene bleiben den Benutzern verborgen. Benutzer sind durchaus zufrieden, wenn sie sich mit den unternehmensfreundlichen Rollennamen auseinandersetzen.

> ***Terminologie***
> Der Begriff RBAC wird in der Branche häufig verwendet, die tatsächliche Bedeutung von RBAC ist jedoch nicht immer klar. Die Verwirrung wird möglicherweise durch die Tatsache verursacht, dass es eine formale RBAC-Spezifikation gibt, die als NIST-RBAC-Modell bekannt ist. Wenn von RBAC gesprochen wird, meinen einige damit das spezifische formale Modell, andere meinen alles, was diesem formalen Modell ähnlich ist, und wieder andere meinen alles, was sich mit Rollen befasst.
>
> Hier wird der Begriff RBAC in einem recht weiten Sinne verwendet. Große Identitätsmanagementsysteme implementieren normalerweise einen Mechanismus, der vom formalen NIST-RBAC-Modell inspiriert ist, der Mechanismus weicht jedoch bei Bedarf vom formalen Modell ab.
>
> Das meinen wir, wenn wir den Begriff RBAC verwenden.

Die meisten RBAC-Systeme ermöglichen die Platzierung von Rollen innerhalb anderer Rollen und schaffen so eine Rollenhierarchie. An der Spitze der Hierarchie stehen in der Regel Geschäftsrollen wie „Marketingspezialist“. Geschäftsrollen enthalten untergeordnete Rollen. Dabei handelt es sich häufig um Anwendungsrollen wie „Website-Analyse“ oder „CMS-Administrator“. Diese untergeordneten Rollen können konkrete Berechtigungen enthalten. Oder sie enthalten möglicherweise andere Rollen, die noch näher an der zugrunde liegenden Technologie liegen. Und so weiter, und so weiter, ganz unten gibt es sprichwörtliche Schildkröten. Eine Rollenhierarchie ist oft ein Muss, wenn die Anzahl der Berechtigungen und Benutzer steigt.

Kein IDM-System kann ohne den vorhandenen RBAC-Mechanismus wirklich vollständig sein. Daher unterstützt die überwiegende Mehrheit der IDM-Systeme Rollen auf die eine oder andere Weise. Die Qualität der RBAC-Unterstützung variiert jedoch erheblich. Einige IDM-Systeme unterstützen nur das absolute Minimum, das für die Inanspruchnahme der RBAC-Unterstützung erforderlich ist. Andere Systeme verfügen über hervorragende und sehr fortschrittliche dynamische und parametrische Hybrid-RBAC-Systeme. Die meisten IDM-Systeme liegen irgendwo dazwischen.

Der rollenbasierte Mechanismus ist ein sehr nützliches Verwaltungstool. Tatsächlich führt die Effizienz rollenbasierter Mechanismen häufig zu deren Überbeanspruchung. Dies ist insbesondere in größeren und irgendwie komplexen Umgebungen eine echte Gefahr. Die Personen, die Rollen in einer solchen Umgebung entwerfen, haben eine starke Motivation, die Ordnung aufrechtzuerhalten, indem sie die Rollen in kleinste wiederverwendbare Teile aufteilen und sie dann in einer Form von Anwendungs- und Geschäftsrollen neu kombinieren. Dies wird durch bewährte Sicherheitspraktiken wie das Prinzip der geringsten Privilegien noch verstärkt. Das ist eine völlig verständliche und vollkommen berechtigte Motivation. Es erfordert jedoch äußerste Sorgfalt, eine solche RBAC-Struktur wartbar zu halten. Auch wenn dies kontraintuitiv erscheinen mag, ist es durchaus üblich, dass die Anzahl der Rollen die Anzahl der Benutzer im System übersteigt. Leider verwandelt dieser Ansatz das komplexe Problem der Benutzerverwaltung in ein noch komplexeres Problem der Rollenverwaltung. Dieses Phänomen wird als Rollenexplosion bezeichnet.

Eine Rollenexplosion ist eine echte Gefahr und lässt sich definitiv nicht leicht vermeiden. Der bei den IDM-Implementierungen der ersten Generation vorherrschende Ansatz bestand darin, einfach mit den Folgen der Rollenexplosion zu leben. Bei einigen IDM-Bereitstellungen wurden sogar Tools erstellt, die Hunderttausende von Rollen automatisch generieren und (mehr oder weniger erfolgreich) verwalten konnten. Dies ist jedoch kein nachhaltiger Ansatz. Die IDM-Systeme der zweiten Generation bringen Funktionen mit, die helfen können, die Rollenexplosion von vornherein zu verhindern. Solche Mechanismen basieren meist auf der Idee, die Rollen zu dynamisieren. Die Rollen sind nicht mehr nur ein statischer Satz von Berechtigungen. Dynamische Rollen können kleine Teile algorithmischer Logik enthalten, die zum Aufbau der Berechtigungen verwendet werden. Eingaben für diese Algorithmen sind Parameter, die bei der Zuweisung der Rolle angegeben werden. Daher kann dieselbe Rolle für viele verwandte Zwecke wiederverwendet werden, ohne dass die Rollen dupliziert werden müssen. Dies kann die Anzahl der Rollen, die zur Modellierung eines komplexen Systems erforderlich sind, erheblich einschränken. Das ist die beste Waffe gegen Rollenexplosion, die wir derzeit haben.

Auch wenn das RBAC-System einige Nachteile hat, ist es für fast alle praktischen IDM-Lösungen notwendig. Es gab mehrere Versuche, das RBAC-System durch einen völlig anderen Ansatz zu ersetzen. Solche Versuche haben im Bereich der Zugriffsverwaltung und verwandten Bereichen einige Erfolge erzielt. Diese Alternativen können RBAC im Identitätsmanagement jedoch nicht ohne weiteres ersetzen. Ein beliebtes Beispiel ist die attributbasierte Zugriffskontrolle (ABAC). Die ABAC-Idee basiert darauf, die Rollen durch rein algorithmische Richtlinien zu ersetzen. Einfach ausgedrückt handelt es sich bei der ABAC-Richtlinie um eine Reihe von Algorithmen, die Benutzerattribute als Eingabe verwenden. Die Richtlinie kombiniert diese Eingabe mit den Daten zu Betrieb und Kontext. Die Ausgabe der Richtlinie ist eine Entscheidung darüber, ob ein Vorgang zugelassen oder verweigert werden soll. Dieser Ansatz ist einfach und funktioniert

möglicherweise einigermaßen gut in der Welt der Zugriffsverwaltung, wo der AM-Server viele Details über den gerade stattfindenden Vorgang kennt. Aber im IDM-Feld müssen wir das Konto einrichten, bevor sich der Benutzer zum ersten Mal anmeldet. Es liegen noch keine Daten zum Einsatz vor. Und selbst Kontextdaten sind sehr begrenzt. Dies und andere Probleme machen ABAC zu einer sehr schlechten Wahl für ein IDM-System. Daher ist RBAC der Hauptmechanismus jeder praktischen IDM-Lösung.

Role Based Access Control Projekte werden meist unter Verwendung von *Top-Down*- oder der *Bottom-Up*-Ansätzen gesteuert. Top-Down beginnt mit den bereits vorhandenen Rollen während der Bottom-Up-Ansatz mit den Daten beginnt und hieraus Rollen definiert.

Beiden Ansätzen liegt der Glaube zugrunde, dass eine gewisse Anzahl an Benutzern zu viele Rollen besitzen, die sie für die Ausübung ihrer Arbeit nicht benötigen. Dies kann schnell zu einem Sicherheitsrisiko führen und sollte ebenfalls im Risikomanagement betrachtet werden.

Top-Down RBAC

Um dieses Risiko zu vermeiden, werden Identitäten mit verschiedenen Rollen und Berechtigungen ausgestattet. RBAC bündelt nun Einzelberechtigungen und fügt diese in Rollen zusammen. Um jedoch evaluieren zu können, welche Rollen die verschiedenen Identitäten wirklich zugewiesen bekommen sollten, werden Daten benötigt. Diese werden im RBAC-Ansatz unter anderem erhoben, in dem man sehr aussagekräftige Positionen betrachtet, die tatsächlich verwendeten Autorisierungen, während einem definierten Zeitraum loggt und in dem man verantwortliche Personen, wie Manager befragt.

Diese Daten können nun mit Informationen aus standortbezogenen Richtlinien etc. ergänzt und Ausnahmen definiert werden.

Ziel ist es, die normalerweise vorherrschenden Rollen und Berechtigungen für die bestimmten Stellen, Positionen, Standorte etc. zu identifizieren sowie die Autorisierungen dahingehend zu bereinigen, dass eine konsistente Vergabe der Rollen basierend auf den Abteilungen, Stellenbeschreibungen, Positionen etc. stattfindet und inkonsistente Autorisierungen entzogen werden können. Dies beinhaltet unter anderem das Aufstellen und Umsetzen von unter anderem Least Privilege und Separation of Duties Prinzipien.

Als Ergebnis erhält man einen bereinigten Satz an Rollen, welche auf Positionen, Abteilungen und Standorte hin definiert wurden und weiterentwickelt werden kann. Diese können im Anschluss verteilt werden.

Bottom-Up RBAC

Anders als beim Top-Down-Ansatz wird beim Bottom-Up-Ansatz, wie die Bezeichnung schon vermuten lässt, mit den Daten begonnen. Diese werden unter Verwendung eines Role-Mining-Tools auf interessante Verknüpfungen und Beziehungen untersucht.

Mit Hilfe des Role-Mining-Tools können so eventuelle Rollen entdeckt werden. Jedoch befindet sich in nicht zuvor bereinigten Datensätzen sehr viel „Rauschen“ was zu unpräzisen Rollendefinitionen führen kann. „Rauschen“ beschreibt die Verunreinigung des Datensatzes mit fälschlicherweise zugewiesenen Einzelberechtigungen – also Berechtigungen, welche nicht entzogen wurden, etc.

Hybrider Ansatz

So unterschiedlich die beiden RBAC-Ansätze auch sind, sie ergänzen sich sehr gut, um das bestmögliche Resultat zu erzielen. Hierbei wird der teilweise automatisierte Bottom-Up-Ansatz, welcher datenbasierte Resultate erzielt, mit den Top-Down definierten Rollen kombiniert umso Rollen zu konstruieren und die Autorisierung für die verschiedenen Positionen, Standorte etc. zu realisieren.

Dieser hybride Ansatz kann beispielsweise nach den folgenden Schemata aufgebaut werden.

Schema A

1. Identifizierung von Rollen
2. Positionsbezogenes Aufräumen der Rollen bzw. Entfernung von „Rauschen“
3. Verwendung eines Role-Mining-Tools, welches nun die Daten analysiert und Rollen mit den jeweiligen Positionen, Abteilungen etc. verbindet.

Schema B:

1. Verwendung eines Role-Mining-Tools, um mögliche Rollen auf Basis der Daten zu identifizieren und diese Positionen, Standorten etc. zuzuweisen
2. Diese Rollen können nun durch die Verantwortlichen verifiziert und gegebenenfalls angepasst werden.

Vor- und Nachteile von RBAC

Vorteile:

- Zusammenfassung von Einzelberechtigungen zu Rollen
- Reduzierung der Komplexität der Berechtigungsvergabe
- Es ist leicht vorübergehenden Zugriff zu erteilen

Nachteile:

- Sehr aufwändig, da die Rollen definiert und auf Aktualität hin überprüft werden müssen
- Temporäre Rollenzuweisungen werden oft nicht wieder entzogen.

Identitätsmanagement und Autorisierungen

Das Grundprinzip der Autorisierung in der Informationssicherheit ist recht einfach: Nehmen Sie das Subjekt (Benutzer), das Objekt (die Dinge, auf die der Benutzer zugreifen möchte) und die Operation. Bewerten Sie, ob die Richtlinie dieses Subjekt-Objekt-Operations-Triple zulässt. Wenn die Richtlinie dies nicht zulässt, verweigern Sie den Vorgang. Das ist ganz einfach. Aber im Bereich des Identitätsmanagements müssen wir ganz anders denken. Wir müssen rückwärts arbeiten. Das IDM-System muss ein Konto für einen Benutzer einrichten, bevor der Benutzer einen Vorgang initiiert. Und wenn der Benutzer tatsächlich einen Vorgang startet, weiß das IDM-System nichts davon. Dadurch wird das Konzept der Autorisierung in der IDM-Welt irgendwie völlig auf den Kopf gestellt.

Das IDM-System ist nicht direkt an der Autorisierung beteiligt. Das IDM-System richtet Konten in Anwendungen und Datenbanken ein. Das IDM-System selbst ist jedoch nicht aktiv, wenn sich der Benutzer bei einer Anwendung anmeldet und die Vorgänge ausführt. Bedeutet das, dass das IDM-System nichts gegen Berechtigungen unternehmen kann? Definitiv nicht. Das IDM-System erzwingt keine Entscheidung zur Autorisierung. Das IDM kann jedoch die Daten verwalten, die bestimmen, wie die Berechtigung bewertet wird. Das IDM-System kann das Konto den richtigen Gruppen zuordnen, was dazu führt, dass bestimmte Vorgänge zugelassen und andere Vorgänge abgelehnt werden. Das IDM-System kann für jedes von ihm verwaltete Konto eine Zugriffskontrollliste (ACLs) einrichten. Das IDM-System wertet die Berechtigungen nicht direkt aus und setzt sie nicht durch. Es verwaltet aber indirekt die Daten, die zur Auswertung von Berechtigungen herangezogen werden. Und das ist eine äußerst wichtige Funktion.

Authentifizierung und Autorisierung sind zwei sehr wichtige Konzepte der Informationssicherheit. Und sie sind für jede Identity & Access Management-Lösung von entscheidender Bedeutung. Allerdings ist die Authentifizierung grundsätzlich recht einfach. Ja, der Benutzer kann über mehrere Anmeldeinformationstypen verfügen, die bei der adaptiven Multi-Faktor-Authentifizierung verwendet werden. Obwohl diese Beschreibung etwas beängstigend klingt, ist sie dennoch nicht so komplex. Es gibt nur ein paar Richtlinienanweisungen, die die Authentifizierung regeln. Außerdem ist die Authentifizierung in der Regel recht einheitlich: Die meisten Benutzer authentifizieren sich mit demselben Mechanismus. Die Authentifizierung ist nicht so schwer zu zentralisieren (auch wenn sie teuer sein kann). Die Authentifizierung ist daher relativ einfach zu handhaben.

Bei der Autorisierung sieht es jedoch ganz anders aus. Jede Anwendung verfügt über einen leicht unterschiedlichen Mechanismus zur Autorisierung. Und diese Mechanismen sind nicht einfach zu vereinheitlichen. Eines der größten Hindernisse besteht darin, dass jede Anwendung mit unterschiedlichen Objekten arbeitet. Die Objekte können komplexe Beziehungen zu anderen Objekten haben und alle von ihnen können auch komplexe Beziehungen zu den Subjekten haben. Die Operationen sind auch alles andere als einfach, da sie parametrisiert werden können. Und dann ist da noch der Kontext. Möglicherweise gibt es Grenzwerte pro Vorgang, tägliche Grenzwerte oder Vorgänge, die nur zu bestimmten Zeiten oder wenn sich das System in einem bestimmten Zustand befindet, zulässig sind. Und so weiter. Das lässt sich sehr schwer zentralisieren. Außerdem verfügt fast jeder Benutzer über eine leicht unterschiedliche Kombination von Berechtigungen. Das bedeutet, dass es eine große Variabilität und eine Menge zu verwaltender Richtlinien gibt. Und dann gibt es noch zwei entscheidende Aspekte, die der Komplexität eine ganz neue Dimension verleihen: Leistung und Skalierbarkeit. Entscheidungen zur Autorisierung werden laufend evaluiert. Es kommt nicht selten vor, dass eine Berechtigung für jede Anfrage mehrfach bewertet wird. Die Verarbeitung zur Autorisierung muss schnell erfolgen, wirklich schnell. Sogar ein Roundtrip über ein lokales Netzwerk kann die Leistung beeinträchtigen. Aus Komplexitäts- und Leistungsgründen sind die Mechanismen zur Autorisierung häufig eng in die Struktur jeder einzelnen Anwendung integriert. Z.B. ist es gängige Praxis, dass Richtlinien zur Autorisierung in SQL übersetzt und als zusätzliche Klauseln in SQL-Abfragen auf Anwendungsebene verwendet werden. Diese Technik nutzt die Datenbank-Engine, um schnell die Daten herauszufiltern, auf die der Benutzer nicht zugreifen darf. Diese Methode ist sehr effizient und möglicherweise die einzige praktische Option beim Umgang mit großen Datenmengen. Allerdings ist dieser Ansatz eng an das Anwendungsdatenmodell gebunden und lässt sich in der Regel kaum externalisieren.

Daher ist es nicht realistisch zu erwarten, dass die Autorisierung in absehbarer Zeit zentralisiert werden könnte. Die Richtlinien zur Autorisierung müssen in den Anwendungen verteilt werden. Die Verwaltung partieller und verteilter Richtlinien ist jedoch keine leichte Aufgabe. Jemand muss sicherstellen, dass die Anwendungsrichtlinien mit der allgemeinen Sicherheitsrichtlinie der Organisation übereinstimmen. Glücklicherweise sind die IDM-Systeme speziell für die Verwaltung und Synchronisierung von Daten in einer Vielzahl von Systemen konzipiert. Daher ist das IDM-System die offensichtliche Wahl, wenn es um die Verwaltung von Berechtigungsrichtlinien geht.

Organisationsstruktur, Rollen, Dienste und Anderes

In den 2000er Jahren drehte sich beim IDM alles um die Verwaltung von Benutzerkonten. Für eine erfolgreiche IDM-Bereitstellung reichte es aus, ein Konto zu erstellen, zu deaktivieren und zu löschen. Aber die Welt ist jetzt ein anderer Ort. Die Verwaltung der Konten reicht einfach nicht mehr aus. Ja, die automatisierte Kontoverwaltung bringt erhebliche Vorteile mit sich und ist eine notwendige Voraussetzung, um in komplexen Systemen zumindest ein Mindestmaß an Sicherheit zu erreichen. Doch oft reicht die Kontoverwaltung nicht aus, um die Kosten eines IDM-Systems zu rechtfertigen. Daher können aktuelle IDM-Systeme weit mehr als nur eine einfache Kontoverwaltung.

Es gibt viele Dinge, die ein fortschrittliches IDM-System verwalten kann:

- **Konten**

  Offensichtlich. Viele IDM-Systeme können Kontoattribute, Gruppenmitgliedschaften, Berechtigungen, Kontostatus (aktiviert/deaktiviert), Gültigkeitsdaten und alle anderen Details vollständig verwalten.

- **Berechtigungen**

  Neben der Verwaltung der Zuweisung/Entzug von Konten zu/von Berechtigungen kann das IDM-System den gesamten Lebenszyklus einer Berechtigung verwalten: eine Berechtigung erstellen, verwalten und löschen.

- Organisationsstruktur
  Das IDM-System kann die Organisationsstruktur von seiner maßgeblichen Quelle (normalerweise der Personalabteilung) übernehmen und sie mit allen Anwendungen synchronisieren, die sie benötigen. Oder das IDM selbst kann zur manuellen Pflege einer Organisationsstruktur verwendet werden.

- Server, Dienste, Geräte und „Dinge“
  Obwohl dies noch kein IDM-Mainstream ist, gibt es einige experimentelle Lösungen, die IDM-Prinzipien verwenden, um Konzepte zu verwalten, die etwas außerhalb des traditionellen IDM-Bereichs liegen. Z.B. gibt es eine IDM-basierte Lösung, die für jedes neue Projekt automatisch einen vordefinierten Satz virtueller Maschinen bereitstellen kann. Die neuen IDM-Systeme sind so flexibel, dass sie theoretisch alles verwalten können, was zumindest am Rande mit dem Konzept der Identität zu tun hat:

  - virtuelle Maschinen
  - Netzwerke

- Anwendungen
- Konfigurationen
- Geräte.

Obwohl alle diese Funktionen interessant sind, stechen einige davon deutlich hervor. Die Verwaltung von Gruppen und die Organisationsstruktur sind für fast jede neue IDM-Bereitstellung von entscheidender Bedeutung. Ihre Organisationsstruktur kann nahezu flach und projektorientiert sein oder es existieren zwölf Ebenen mit Abteilungen und Bereichen. Aber unabhängig von der Größe und Form der Organisationsstruktur muss diese anwendungsübergreifend verwaltet und synchronisiert werden, und zwar im Wesentlichen auf die gleiche Weise, wie Identitäten synchronisiert werden. Möglicherweise müssen für jede Organisationseinheit Gruppen in Active Directory erstellt und korrekt verschachtelt werden. Möglicherweise soll für jedes Mitglied eines Ad-hoc-Teams eine Verteilerliste erstellt werden. Dieser Vorgang soll so wenig Overhead wie möglich verursachen, da die Teams sonst nicht wirklich ad-hoc verwaltet werden können. Möglicherweise möchten Sie die Informationen zu Projekten mit dem System zur Problemverfolgung synchronisieren. Möglicherweise möchten Sie auch automatisch einen separaten Wiki-Bereich und ein neues Repository für den Quellcode für jedes neue Entwicklungsprojekt erstellen. Die Möglichkeiten sind nahezu endlos.

Das Organisationsstrukturmanagement ist eng mit dem Gruppenmanagement verbunden. Die Gruppen sind häufig an Arbeitsgruppen, Projekte oder Organisationseinheiten gebunden. Z.B. und das IDM-System kann automatisch mehrere Gruppen für jedes Projekt verwalten (Administrator- und Mitgliedergruppen). Diese Gruppen können zur Autorisierung verwendet werden. Ebenso kann ein IDM-System automatisch Rollen auf Anwendungsebene, Zugriffskontrolllisten (ACLs) und andere Datenstrukturen verwalten, die normalerweise für die Autorisierung verwendet werden.

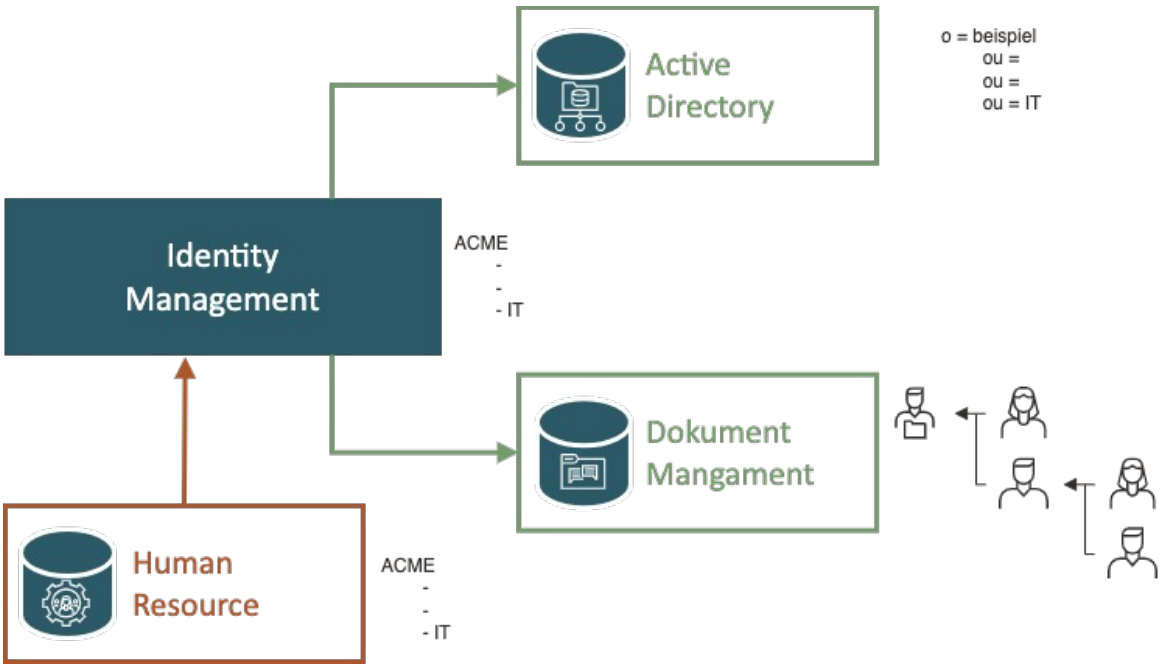

Abbildung 13: Identity Organisationsstruktur

Während diese Funktionalität in nahezu jeder Bereitstellung Vorteile bietet, ist die Verwaltung der Organisationsstruktur für Organisationen, die auf baumartigen funktionalen Organisationsstrukturen basieren, von entscheidender Bedeutung. Diese Organisationen

verlassen sich stark auf die Informationen, die aus der Organisationsstruktur abgeleitet werden. Z.B. der direkte Vorgesetzte des Dokumentautors kann das Dokument im Dokumentenmanagementsystem prüfen und genehmigen. Nur die Mitarbeiter derselben Abteilung können den Dokumententwurf sehen. Nur die Mitarbeiter einer Marketingabteilung können Marketingpläne sehen, usw. Traditionell werden solche Daten in einem unverständlichen Satz von Berechtigungsgruppen und -listen kodiert. Und das trägt dazu bei, dass Reorganisationen für IT-Administratoren ein absoluter Albtraum sind. Ein IDM-System kann die Situation jedoch deutlich verbessern. IDM kann die Gruppen automatisch erstellen. Dadurch kann sichergestellt werden, dass diesen Gruppen die richtigen Benutzer zugewiesen werden. Es kann Informationen über die Manager in allen betroffenen Anwendungen synchronisieren. Und so weiter. Und ein gutes IDM-System kann all das mit nur einer Handvoll Konfigurationsobjekten erledigen.

Das scheint fast zu schön um wahr zu sein. Und es ist fair zuzugeben, dass die Qualität der Organisationsmanagementfunktionen zwischen den IDM-Systemen erheblich variiert. Gruppenmanagement und Organisationsstrukturmanagement scheinen ein sehr problematisches Merkmal zu sein. Nur wenige IDM-Systeme unterstützen diese Konzepte auf dem Niveau, das eine praktische, sofort einsatzbereite Bereitstellung ermöglicht. Die meisten IDM-Systeme bieten hierfür eine gewisse Unterstützung, aber jede praktische Lösung erfordert umfangreiche Anpassungen.

Jeder braucht Identitätsmanagement

Ein solcher Titel mag wie eine gewaltige Übertreibung erscheinen. Aber tatsächlich kommt es der Wahrheit sehr nahe. Jedes nicht-triviale System benötigt ein Identitätsmanagement, auch wenn sich die Besitzer derartiger Systeme dessen möglicherweise nicht bewusst sind. In diesem Fall geht es hauptsächlich um die Kosten-Nutzen-Rechnung. Identitätsmanagement weist eine gewisse inhärente Komplexität auf. Auch wenn selbst sehr kleine Systeme IDM benötigen, ist der Nutzen wahrscheinlich zu gering, um die Kosten zu rechtfertigen. Das Kosten-Nutzen-Verhältnis ist für mittelständische Unternehmen deutlich besser. Ein umfassendes, automatisiertes Identitätsmanagement ist für große Systeme eine absolute Notwendigkeit. Es scheint eine Faustregel zu geben, die eine recht breite Anwendbarkeit hat:

Anzahl der Nutzer	Empfehlung
Weniger als 200	Möglicherweise wird ein automatisiertes Identitätsmanagement benötigt, aber der Nutzen ist wahrscheinlich zu gering, um die Kosten zu rechtfertigen. Die manuelle Verwaltung von Benutzerkonten ist wahrscheinlich immer noch eine praktikable Option.
200 – 2.000	Ein automatisiertes Identitätsmanagement wird benötigt und die Vorteile reichen möglicherweise gerade aus, um die Kosten zu rechtfertigen. Dennoch muss man nach einer sehr kosteneffizienten Lösung suchen. Die Automatisierung der grundlegendsten und zeitaufwändigsten Aufgaben reicht wahrscheinlich gerade aus.
2.000 – 20.000	Ein automatisiertes Identitätsmanagement ist erforderlich. Diese

	Menge kann man manuell nicht verwalten. Bei richtiger Implementierung der Identitätsmanagementlösung, werden die Vorteile viel höher sein als die Kosten.
Mehr als 20.000	Selbst erklärend.

Identity Governance

Bei *Identity Governance* handelt es sich im Grunde um ein Identitätsmanagement auf einer höheren Ebene. Das eigentliche Identitätsmanagement konzentriert sich hauptsächlich auf technische Aspekte des Lebenszyklus von Identitäten wie die automatisierte Bereitstellung, Synchronisierung, Bewertung der Rollen und Computerattribute. Andererseits abstrahiert Identity Governance von den technischen Details und konzentriert sich auf Richtlinien, Rollen, Geschäftsregeln, Prozesse und Datenanalyse. Z.B. kann ein Governance-System sich mit den Richtlinien der Aufgabentrennung (Segregation of Duties) befassen. Dies kann den Prozess der erneuten Zertifizierung von Zugriffen vorantreiben. Der Schwerpunkt kann auf der automatischen Analyse und Berichterstattung der Identitäts-, Audit- und Richtliniendaten liegen. Dadurch werden Maßnahmen zur Behebung von Verstößen gegen Richtlinien vorangetrieben. Es verwaltet die Anwendung neuer und geänderter Richtlinien, bewertet, wie das System Richtlinien und Vorschriften einhält usw. Dieser Bereich wird manchmal als Governance, Risikomanagement und Compliance (GRC) bezeichnet.

Fast alle IDM-Systeme benötigen mindestens einige Governance-Funktionen, um in der Praxis von Nutzen zu sein. Darüber hinaus sind viele Governance-Funktionen lediglich eine Weiterentwicklung von Konzepten, die vor vielen Jahren im IDM-Bereich entstanden sind. Daher ist die Grenze zwischen Identitätsmanagement und Identity Governance ziemlich fließend. Die Grenze ist so fließend, dass neue Begriffe für den einheitlichen Bereich erfunden wurden, der das eigentliche Identitätsmanagement zusammen mit der Identity Governance umfasst. Einer dieser Begriffe ist Identity Governance & Administration (IGA).

In den 2010er Jahren war es üblich, dass Identitäts-Governance-Funktionen durch spezialisierte Produkte implementiert wurden, die von den zugrunde liegenden IDM-Plattformen getrennt waren. Viele IDM- und Governance-Lösungen sind immer noch in (mindestens) zwei Produkte unterteilt. Es versteht sich vielleicht von selbst, dass vernünftige IDM-Lösungen sowohl IDM- als auch Governance-Funktionen in einem einheitlichen und gut aufeinander abgestimmten Produkt bieten sollten.

Identity Governance-Funktionen

Nachfolgend finden Sie eine Liste der Funktionen, die zur Kategorie Governance/Compliance gehören. Da die Grenzen der Governance so fließend sind, gibt es auch Funktionen, die als Governance-bezogene IDM-Funktionen betrachtet werden können.

- **Delegierte Administration**

  Grundlegende IDM-Bereitstellungen basieren normalerweise auf der Idee eines allmächtigen Systemadministrators, der fast alles tun kann. Dann gibt es Endbenutzer, die fast nichts tun können. Während dieses Konzept in kleinen und

einfachen Bereitstellungen funktionieren kann, reicht es für größere Systeme nicht aus. Große Organisationen müssen in der Regel einige Verwaltungsrechte an andere Benutzer delegieren. Dabei kann es sich um HR-Personal, Personen, die für die Verwaltung ihrer Organisationseinheiten verantwortlich sind, Administratoren, die für eine bestimmte Gruppe von Systemen verantwortlich sind, Anwendungsadministratoren usw. handeln.

- **Vertreter**

  Die delegierte Verwaltung ist sehr nützlich, aber dennoch recht statisch. Die delegierte Verwaltung wird in Richtlinien festgelegt, die nicht ganz einfach zu ändern sind. Allerdings besteht häufig Bedarf an einer Ad-hoc-Delegierung, beispielsweise einer vorübergehenden Delegation von Privilegien während des Urlaubs des Managers. Ein solcher Manager könnte einen Stellvertreter ernennen, der einige Privilegien des Managers erhalten würde. Dies geschieht alles auf Ad-hoc-Basis, ausgelöst durch eine explizite Aktion des Managers.

- **RBAC-bezogene Richtlinien**,

  wie z. B. die Segregation of Duties (SoD)-Richtlinie. Einfach ausgedrückt stellt die SoD-Richtlinie sicher, dass widersprüchliche Aufgaben nicht bei einer einzelnen Person angesammelt werden können. Dies wird normalerweise durch die Verwendung eines Mechanismus zum Ausschluss von Rollen implementiert. Es kann jedoch sein, dass es tiefer geht. Z.B. kann es erforderlich sein, dass jeder Antrag von mindestens zwei Personen genehmigt wird.

- **Richtlinien im Zusammenhang mit der Organisationsstruktur**

  Eine Organisationsstruktur mag wie ein einfacher, harmloser Baum aussehen, aber in Wirklichkeit ist sie alles andere als einfach. Theoretisch sollte die Organisationsstruktur von Geschäfts- oder Betriebsabteilungen wie der Personalabteilung verwaltet werden. Doch die Realität sieht oft ganz anders aus. Den Geschäftsabteilungen fehlen die Werkzeuge und Prozesse, um die Organisationsstruktur effizient zu verwalten. Daher übernimmt häufig ein IDM-System die Verantwortung für das Organisationsstrukturmanagement. In solchen Fällen besteht die Notwendigkeit, die Organisationsstruktur zu überwachen. Beispielsweise kann es Richtlinien geben, die einen einzelnen Manager für jede Abteilung vorschreiben. In diesem Fall muss das IDM-System möglicherweise mit Situationen umgehen, in denen es keinen oder zu viele Manager gibt.

- **Dynamische Genehmigungsschemata**

  Genehmigungsprozesse gelten in der Regel als Teil der grundlegenden Funktionalität des Identitätsmanagements, wie sie bereits in den 2000er Jahren in frühen IDM-Systemen vorhanden waren. Der Benutzer (oder der Manager) fordert eine Rollenzuweisung an. Der Vorgang durchläuft vor seiner Ausführung einen Genehmigungsprozess. Genehmigungen stellen einen sehr nützlichen Mechanismus dar, insbesondere für den Fall, dass die Rollenzuweisung nicht automatisiert werden kann, normalerweise aufgrund nicht vorhandener Richtlinien.

  Genehmigungen werden in der Regel von fast allen IDM-/Governance-Systemen durch eine Art Workflow-Engine implementiert. Dies führt jedoch häufig zu

Wartungsproblemen, insbesondere bei Bereitstellungen, bei denen der Schwerpunkt auf Identitäts-Governance-Funktionen liegt. In solchen Fällen handelt es sich bei den Genehmigungsprozessen nicht mehr um einfache quasi lineare Arbeitsabläufe. Genehmigungsprozesse sind in der Regel sehr dynamisch und ihre Natur wird fast vollständig von den Richtlinien und nicht von den Prozessabläufen bestimmt. Workflow-Engines haben es sehr schwer, mit einer solch dynamischen Situation zurechtzukommen. IDM-Systeme, die spezielle richtlinienbasierte Genehmigungs-Engines implementieren, bieten viel bessere Lösungen.

Genehmigungsmechanismen sind sehr nützlich. Allerdings haben sie auch eine dunkle Seite. Genehmigungsentscheidungen werden oft auf einer sehr subjektiven „sieht gut aus“-Basis getroffen. Dies eröffnet offensichtlich die Möglichkeit für Fehlentscheidungen und Nachlässigkeit. Eine falsche Verweigerung der Rollenzuweisung löst wahrscheinlich ein sofortiges (und gelegentlich recht emotionales) Feedback des Antragstellers aus. Die Genehmigung einer Rollenzuweisung, die nicht zugewiesen werden sollte, wird jedoch wahrscheinlich überhaupt kein Feedback auslösen. Dennoch stellt eine solche Entscheidung wahrscheinlich ein Sicherheitsrisiko dar, ein Risiko, das sehr schwer zu erkennen ist. Dies kann teilweise durch einen mehrstufigen Genehmigungsprozess gelöst werden, insbesondere für sensible Rollen. Allerdings besteht ein Trend zur „Automatisierung“ von Genehmigungsentscheidungen auf Basis von Mechanismen „künstlicher Intelligenz“. Dies scheint ein sehr nützliches und zeitsparendes Werkzeug zu sein. Allerdings ist die künstliche Intelligenz nur so gut wie die Trainingsdaten. Wenn die Maschine mit schlechten Entscheidungen trainiert wird, schlägt sie auch schlechte Entscheidungen vor. Erschwerend kommt hinzu, dass solche Entscheidungen nur sehr begrenzt sichtbar und nachvollziehbar sind. Daher müssen solche Mechanismen mit größter Vorsicht eingesetzt werden.

- **Berechtigungsverwaltung**

  ist hauptsächlich eine Sache der Identitätsverwaltung. Dabei geht es um Berechtigungen von Benutzerkonten in Zielsystemen wie Rollen- oder Gruppenmitgliedschaften. Dieser Prozess kann jedoch in beide Richtungen verlaufen. Governance-Systeme können Funktionen zur „Berechtigungserkennung“ bereitstellen, die Berechtigungen als Eingaben verwenden. Dies kann zur Bewertung von Compliance- und Richtlinienverstößen verwendet werden, kann aber auch ein wertvoller Input für die Rollenentwicklung sein.

- **Role Mining**

  IDM-Systeme werden selten auf der grünen Wiese eingesetzt. Im Normalfall sind bestehende Systeme vorhanden, es gibt Anwendungsrollen, Berechtigungen und Privilegien. Es ist keine leichte Aufgabe, IDM-Rollen zu erstellen, die dieser Umgebung zugeordnet sind. Dies ist normalerweise ein langsamer und langwieriger Prozess. Das IDM-System kann jedoch alle vorhandenen Informationen abrufen und diese zum Vorschlagen einer Rollenstruktur verwenden. Dies ist kein vollständig deterministischer Prozess, er erfordert viel Benutzerinteraktion und Optimierung und basiert häufig auf maschinellen Lernfunktionen. Es ist kein Ersatz für Rollen-

Engineering-Expertise. Maschinengestütztes Role Mining kann den Prozess jedoch deutlich beschleunigen.

- **Kampagnen zur Re-Zertifizierung**

  Die Rollenverteilung ist oft eine einfache Aufgabe. Fordern Sie eine Rolle an, die Rolle durchläuft einen Genehmigungsprozess und die Rolle wird zugewiesen. Dann vergisst es jeder. Es besteht ein erheblicher Anreiz, die Zuweisung einer neuen Rolle zu beantragen. Dennoch besteht fast kein Anreiz, die Aufhebung einer nicht mehr benötigten Rolle zu beantragen. Dies führt im Laufe der Zeit zu einer Anhäufung von Privilegien. Ein solches Horten von Privilegien kann für Mitarbeiter mit einer langen und reichen Erfahrung bei Jobwechseln ein gefährliches Ausmaß erreichen. Daher gibt es Kampagnen zur Re-Zertifizierung, die auch als „Zertifizierungs-“, „Zugangszertifizierungs“- oder „Attestierungs“-Mechanismen bezeichnet werden. Das Ziel dieser Kampagne besteht darin, zu bestätigen („zertifizieren“ oder „bezeugen“), dass der Benutzer weiterhin die zuvor zugewiesenen Berechtigungen benötigt. Kampagnen zur Re-Zertifizierung sind so konzipiert, dass sie auf sehr effiziente Weise bei einer großen Anzahl von Benutzern durchgeführt werden können. Für die Durchführung solcher Kampagnen gibt es daher spezielle Prozesse und eine ganz bestimmte Benutzeroberfläche.

  Ähnlich wie Genehmigungsprozesse bieten einige Identity-Governance-Systeme Unterstützung durch „künstliche Intelligenz“ für Prozesse der Rezertifizierung. Eine solche Unterstützung kann sehr attraktiv sein, da Prozesse der Rezertifizierung aufgrund der Vielzahl an Entscheidungen, die bei jeder Kampagne getroffen werden müssen, oft recht einschüchternd sind. Allerdings sind die Risiken einer solchen „Automatisierung“ noch ausgeprägter als im Genehmigungsfall. Eine erneute Zertifizierung ist oft der letzte Schutz gegen die gefährliche Anhäufung von Privilegien. Schlecht ausgebildete künstliche Intelligenz kann zu einer systematischen Anhäufung von Risiken im Unternehmen führen.

- **Rollenverwaltung** ist in der Regel eine recht komplexe Angelegenheit.

  Eine typische IDM-Bereitstellung umfasst wahrscheinlich eine große Anzahl von Rollen. Es ist ziemlich schwierig, diese Rollen überhaupt zu definieren. Dann ist es noch schwieriger, die Rollen aufrechtzuerhalten. Die Umgebung verändert sich ständig, daher müssen sich auch die Rollen ändern. Dies liegt in der Regel außerhalb der Befugnisse eines einzelnen Administrators. Daher werden in der Regel viele Rolleninhaber benannt, die sich um die Rollenpflege kümmern. Rollen werden häufig in Anwendungen, Kategorien, Kataloge oder Funktionsbereiche gruppiert. Das IDM-System muss sicherstellen, dass die Eigentümer über die richtigen Berechtigungen für ihre Arbeit verfügen. Das IDM-System sollte außerdem dafür sorgen, dass jede Rolle zu jedem Zeitpunkt mindestens einen Eigentümer hat, dass Rollendefinitionen regelmäßig überprüft werden und so weiter.

- **Rollen-Lifecycle-Management** ist ein dynamischer Teil der Rollenverwaltung.

  Rollenänderungen haben wahrscheinlich schwerwiegende Auswirkungen auf die Gesamtsicherheit des Systems. Daher ist es möglicherweise nicht wünschenswert, Rollenverwaltungsaufgaben einfach zu delegieren. Es kann viel sinnvoller sein, zu

verlangen, dass Rollenänderungen vor der Anwendung genehmigt werden müssen. Außerdem werden ständig neue Rollen erstellt und alte Rollen stillgelegt. Das IDM-System muss möglicherweise sicherstellen, dass keinem neuen Benutzer eine außer Dienst gestellte Rolle zugewiesen wird. Dennoch können während einer Auslaufphase weiterhin alte Rollen im System benötigt werden. Das IDM-System muss den Überblick behalten, um zu verhindern, dass veraltete Rollen für immer in den Systemen verbleiben.

- **Rollenmodellierung**

  Ein Wechsel einer einzelnen Rolle allein macht oft wenig Sinn. Die Rollen sind in der Regel so gestaltet, dass ein Satz Rollen zusammenarbeitet und ein Rollenmodell bildet. Daher kann die Genehmigung der Änderung zu jeder einzelnen Rolle zu lästig und sogar hinderlich sein. Z.B. kann es zu einer inkonsistenten Situation kommen, wenn eine Änderung genehmigt und eine andere abgelehnt wird. Daher werden Rollen und Richtlinien häufig in Modellen zusammengefasst. Die Modelle werden vollständig geprüft, versioniert und angewendet.

- **Simulation**

  IDM-Bereitstellungen sind in der Regel komplex. Es gibt viele Beziehungen, Interaktionen und Richtlinien. Es ist keine leichte Aufgabe, die Auswirkungen einer Änderung einer Rolle, Richtlinie oder Organisationsstruktur vorherzusagen. Daher bieten einige IDM-Systeme Simulationsfunktionen, die Vorhersagen und Auswirkungsanalysen geplanter Änderungen ermöglichen.

- **Compliance-Richtlinien**, Berichterstattung und Management.

  Richtlinien in der Bereich des Identitätsmanagements sind in der Regel darauf ausgelegt, strikt durchgesetzt zu werden. Dies funktioniert gut für grundlegende Richtlinien, die Teil einfacher IDM-Bereitstellungen sind. Das große Problem besteht jedoch darin, neue Richtlinien anzuwenden – insbesondere Richtlinien, die durch Vorschriften, Empfehlungen und Best Practices vorgeschrieben sind. Es ist davon auszugehen, dass ein erheblicher Teil einer Organisation diese neue Richtlinie nicht einhalten wird. Die Anwendung der Richtlinie und deren sofortige Durchsetzung kann zu erheblichen Unterbrechungen in den geschäftlichen Abläufen führen. Allerdings ist es nahezu unmöglich, sich auf neue Richtlinien vorzubereiten und deren Auswirkungen abzumildern, ohne zu wissen, welche Benutzer und Rollen betroffen sind. Daher gibt es einen zweistufigen Prozess. Die Richtlinien werden angewendet, aber noch nicht durchgesetzt. Die Richtlinien werden zur Bewertung der Auswirkungen auf die Compliance verwendet. Compliance-Berichte können verwendet werden, um Benutzer zu finden, die gegen die Richtlinie verstoßen, um Abhilfe zu schaffen. Compliance-Berichte können auch verwendet werden, um das Ausmaß und den Fortschritt der Compliance zu verfolgen.

- **Remediation**

  Gute IDM-Lösungen streben nach Automatisierung. Alle automatisierbaren Prozesse und Aktionen werden automatisiert. Z.B. wenn die Zuweisung einer Rolle aufgehoben wird und der Benutzer kein Konto mehr benötigt, wird dieses Konto automatisch gelöscht oder deaktiviert. Es gibt jedoch Handlungen, die nicht automatisiert werden können, weil sie die Entscheidung eines lebenden und

denkenden Individuums erfordern. Ein Beispiel für solche Prozesse sind Genehmigungen. Es gibt jedoch noch mehr Situationen dieser Art. Viele davon erfordern mehr Initiative als eine einfache Ja/Nein-Entscheidung. Ein Beispiel dafür ist die Verwaltung von Organisationen. Normalerweise gilt die Regel, dass jede Abteilung einen Leiter haben muss. Was sollte das IDM-System jedoch tun, wenn ein Abteilungsleiter aus der Organisation ausscheidet? Das IDM-System kann diesen Vorgang nicht stoppen, da es sicherlich gute Gründe gibt, dieser Führungskraft alle Berechtigungen zu entziehen. Die Führungskraft und alle zugehörigen Konten müssen so schnell wie möglich verschwinden. Jetzt gibt es eine Abteilung ohne Führungskraft, und das IDM-System selbst kann nichts dagegen tun. Hier kommt die Remediation zur Rettung. Der Prozess wird nach dem Vorgang gestartet, der die Führungskraft entfernt hat. Im Rahmen des Remediation-Prozesses wird eine verantwortliche Person aufgefordert, eine neue Führungskraft für die Abteilung zu ernennen. Es kann eine Vielzahl von Remediation-Prozessen geben. Bei einem einfachen Prozess werden Ja/Nein-Entscheidungen abgefragt, oder es wird die Nominierung eines Benutzers verlangt. Dann gibt es oft Möglichkeiten, generische Prozesse einzurichten, die auf völlig unerwartete Situationen anwendbar sind.

- **Automatisierung des Risikomanagements**

  Informationssicherheit ist kein Projekt, sondern ein Prozess. Es beginnt mit der Risikoanalyse, der Planung und der Ausführung, und dann geht es zurück zur Analyse, Planung und Ausführung und so weiter und so fort für immer und ewig. Die Risikoanalyse ist der Teil des Prozesses, der viel Zeit und Mühe erfordert – insbesondere wenn es um die Analyse von Bedrohungen durch Insider geht, da es in der Regel viele Insider zu analysieren gibt. Ein IDM-System kann jedoch dabei helfen, den Aufwand für die Risikoanalyse zu reduzieren. Jede einem Benutzer zugewiesene Rolle ist ein Risiko. Wenn Rollen mit relativen Risikostufen gekennzeichnet sind, kann das IDM-System die Akkumulation der Risiken für jeden Benutzer berechnen. Da jede Rolle Zugriff auf einen bestimmten Satz von Assets gewährt, kann das IDM-System Daten bereitstellen, um die Asset-Exposition für Benutzer zu bewerten.

- **Identity Analytics & Intelligence (IdA) Hierbei handelt es sich meist um Oberbegriffe.** Dabei handelt es sich in der Regel um eine Zusammenstellung mehrerer Funktionen aus dem Bereich Identity Governance, integriert in einen ganzheitlichen, risikobasierten Ansatz. Identitätsanalysen und -informationen beginnen mit einem Blick auf die Daten. Der Prozess beginnt mit der sehr realistischen Annahme, dass die Daten nicht in perfekter Ordnung sind, dass es Inkonsistenzen, Unvollkommenheiten, Risiken und alle möglichen anderen Probleme gibt. Um die Probleme zu erkennen, werden verschiedene Techniken eingesetzt. Die meisten Techniken scheinen auf der Erkennung von Anomalien und Mustern in den Daten zu basieren. Mechanismen zur Erkennung von Ausreißern suchen nach Benutzern mit Berechtigungen, die sich deutlich von den Berechtigungen ihrer Kollegen unterscheiden. Andererseits wird Role Mining verwendet, um ähnliche Berechtigungen zu erkennen, die ähnlichen Benutzern zugewiesen sind, und so neue Rollen vorzuschlagen. Viele der Identitätsanalyse- und Intelligence-Techniken basieren auf Risikomodellierung. Es gibt Mechanismen zur

Identifizierung überprivilegierter Benutzer durch Analyse der Risikobewertungen einzelner Benutzer. Ähnliche Mechanismen können verwendet werden, um Berechtigungen mit hoch eingestuften Privilegien, die einem unterprivilegiertem Benutzer zugewiesen wurden, oder ähnliche Risikoanomalien zu identifizieren.

- **Workflow-Orchestrierung**

  Einige IGA-Plattformen bieten eine Workflow-Orchestrierung. Workflow-Engines steuern Prozesse auf der Grundlage einfacher Algorithmen, die in der Regel viele manuelle Schritte umfassen, die von verschiedenen Personen in Teams ausgeführt werden müssen. IGA-Plattformen nutzen die Workflow-Automatisierung hauptsächlich zur Implementierung von Genehmigungsmechanismen. Während die Verwendung einer Workflow-Engine für Genehmigungen wie eine offensichtliche Wahl erscheint, sind Workflow-Engines möglicherweise die schlechtesten Tools, die es gibt. Genehmigungsprozesse sind in der Regel dynamische Prozesse, deren Form stark von Eingaben (Anfragen) und Richtlinieneinstellungen abhängt. Die Liste der Genehmigenden, Genehmigungsstufen und Beendigungsbedingungen hängen von der Menge der angeforderten Rollen und anderen Faktoren (z. B. der Risikostufe des Benutzers) ab, was in der Sprache der Geschäftsprozessmodellierung auf hoher Ebene nicht einfach zu handhaben ist.

  Auch wenn die Workflow-Orchestrierung für die Implementierung von Genehmigungsprozessen nahezu nutzlos ist, hat sie dennoch ihren Platz in der IGA-Plattform. Die Automatisierung von Arbeitsabläufen kann hilfreich sein, um Onboarding- (Registrierungs-) und Offboarding-Prozesse voranzutreiben. Es kann auch für einige Fälle von Remediation nützlich sein, obwohl die Remediation tendenziell eine unstrukturierte oder halbstrukturierte Aktivität ist, die besser durch fallbezogene Entscheidungen als durch die Workflow-Automatisierung gehandhabt werden kann.

  Fast alle IGA-Plattformen, die die Workflow-Automatisierung unterstützen, bringen ihre eigene (oft proprietäre) Workflow-Engine mit. Das bedeutet, dass die Administratoren lernen müssen, die Arbeitsabläufe zu konfigurieren, Benutzer sich an die neue Benutzeroberfläche anpassen müssen, Benachrichtigungen integriert werden müssen und so weiter. Es wäre viel besser, die vorhandene Workflow-Engine wiederzuverwenden, eine Engine, die von der Organisation bereits zur Steuerung aller anderen Geschäftsprozesse verwendet wird. Mit Ausnahme von Genehmigungen, die stark von der Rollenstruktur abhängen, ähneln andere IGA-Prozesse in der Regel den normalen Geschäftsprozessen in der Organisation. > Die Wiederverwendung vorhandener Plattformen zur Workflow-Automatisierung und -Orchestrierung sollte eine natürliche Wahl sein. Bis auf ein nerviges Detail. Die meisten Organisationen verfügen nicht über ein solches System. Daher kann selbst diese seltsame proprietäre Workflow-Engine, die in die IGA-Plattform eingebettet ist, immer noch recht nützlich sein.

Nicht alle IGA-Plattformen implementieren alle Funktionen. Umfang und Qualität der Umsetzung variieren stark von System zu System. Darüber hinaus verwenden einzelne IGA-Plattformen ihre eigene Terminologie, was die Situation sehr unübersichtlich macht.

Risikobasierter Ansatz zur Identitätsverwaltung

Ein risikobasierter Ansatz für Identitätsmanagement und Governance ist eine sehr gute Idee. Tatsächlich ist es eine ausgezeichnete Idee, eine der besten Ideen seit Jahrzehnten. Allerdings gibt es, wie bei vielen großartigen Ideen, Schwierigkeiten und Nachteile.

Aber Moment mal, was hat es mit dieser „risikobasierten“ Sache auf sich? Um diese Frage zu beantworten, müssen wir einen kurzen Roadtrip durch die Informationssicherheitslandschaft machen.

Der Begriff *Risiko* stammt aus der Informationssicherheitstheorie. Sicherheitsexperten haben schon vor langer Zeit erkannt, dass es nahezu unmöglich ist, ein vollkommen sicheres System zu schaffen. Wenn Sie versuchen, ein System immer sicherer zu machen, ist jeder Schritt teurer als der vorherige. Jede Gegenmaßnahme ist weniger effizient als die vorherige, aufdringlicher, weniger flexibel und schwieriger an die Geschäftsanforderungen anzupassen. Letztendlich gelangt das System in einen Zustand, in dem es für Unternehmen praktisch unbrauchbar ist, das System jedoch immer noch nicht vollständig sicher ist.

Daher entwickelten die Sicherheitsexperten ein Risikokonzept. Das *Risiko* ist ein Maß für die Gefahr, der ein bestimmter *Vermögenswert* ausgesetzt ist. Ein *Vermögenswert* wie eine Kundendatenbank kann im Hinblick auf eine bestimmte *Bedrohung* gefährdet sein, beispielsweise durch einen Hacker, der versucht, die Datenbank zu stehlen, um sie zu verkaufen. Das *Risiko* gibt Auskunft über die Wahrscheinlichkeit, dass ein *Vermögenswert* kompromittiert wird. Beispielsweise ist es offensichtlich ziemlich riskant, die Datenbank in Form einer Tabellenkalkulation auf einem jahrzehntealten Windows-Rechner zu verwalten, der mit einem offenen Internet verbunden ist. Dem Risiko kann durch *Gegenmaßnahmen* begegnet werden. Gegenmaßnahmen sind alles, was wir tun, um Systeme sicherer zu machen, angefangen von Betriebssystem-Updates über Zutrittskontrollsysteme bis hin zu bombensicheren Türen und schwer bewaffneten Wachen.

Da es nicht praktikabel ist, ein System vollständig abzusichern, müssen wir immer ein *gewisses Risiko* in Kauf nehmen. Dies wird als Restrisiko bezeichnet, ein Risiko, das uns bewusst ist, dessen Reduzierung oder Eliminierung jedoch nicht effizient ist. Auch wenn das Restrisiko nicht vollständig beseitigt werden kann, kann es *Pläne zur Risikominderung* geben. Beispielsweise können wir akzeptieren, dass das Risiko einer Sicherheitslücke im Betriebssystem besteht, und keine noch so großen automatisierten Software-Updates, Integrationen von Schwachstellendatenbanken und Wachsamkeit werden das Risiko jemals vollständig beseitigen können. Wir können das Risiko jedoch mindern, indem wir Pläne vorbereiten, die umgesetzt werden, wenn wir von einer Zero-Day-Schwachstelle betroffen sind. Der Plan kann die Sperrung des Netzwerkzugriffs auf anfällige Dienste umfassen, sobald wir von der Schwachstelle erfahren, eine Untersuchung, die nach Spuren eines Angreifers sucht, der die Schwachstelle ausnutzt, Notfallkommunikation und Notfallpläne usw. Bei der Risikominderung geht es darum, die Auswirkungen eines solchen Angriffs weniger schmerzhaft zu machen und den Schaden zu verringern.

Im Idealfall sind wir uns des Risikos vollständig bewusst, sodass wir Gegenmaßnahmen ergreifen und Pläne zur Risikominderung erstellen können. Allerdings müssen wir einiges über das Risiko

wissen, damit die Gegenmaßnahmen und Schadensbegrenzungspläne wirksam sind. Risiko ist nicht nur eine einzelne Zahl, es ist ein mehrdimensionales und oft sehr komplexes Konzept. Die Höhe des Risikos wird in einem Risikobewertungsprozess bewertet, der oft sehr langwierig und anspruchsvoll ist. Der Risikobewertungsprozess bewertet Vermögenswerte, um den Wert der Daten und Dienste zu bestimmen. Der Prozess untersucht Bedrohungen, beispielsweise die Fähigkeiten und Motivationen eines Angreifers. Die Bewertung befasst sich mit Schwachstellen, die Angreifer nutzen können, um Zugriff auf unsere Systeme zu erhalten. Dabei geht es auch um bestehende Gegenmaßnahmen, Prozesse, Richtlinien und andere Details.

Das bedeutet, dass sich der Risikobewertungsprozess mit großen und komplexen Daten befasst, die nicht von einem menschlichen Verstand allein verarbeitet werden können. Die Daten werden üblicherweise in Risikomodelle eingespeist, um das Risiko zu ermitteln. Risikomodelle sind eine Reihe komplexer mathematischer Formeln, die Daten zu Vermögenswerten, Bedrohungen, Schwachstellen und allen anderen Eingaben in eine mehrdimensionale Darstellung von Risikobereichen umwandeln. Theoretisch kann uns das Risikomodell sagen, dass wir ein hohes Risiko in der Netzwerksicherheit haben, insbesondere im Umgang mit Kundendaten – und wir sollten wirklich etwas dagegen tun!

Die Ergebnisse der Risikobewertung sollen die Umsetzung von Gegenmaßnahmen und Plänen zur Risikominderung vorantreiben. Es stehen zu viele Gegenmaßnahmen und Minderungsstrategien zur Auswahl, wir können nicht alle umsetzen. Wir wollen nur die wirklich Effizienten. Wir wollen doch nicht Zeit und Geld mit der ausgeklügelten Verschlüsselung von Daten verschwenden, bei denen es sich nur um Kopien öffentlicher Informationen handelt, oder? Die Risikobewertung soll aussagen, was wichtig ist und was nicht. Gegenmaßnahmen werden genau dort umgesetzt, wo sie benötigt werden und wo sie in der Lage sind, realen Risiken zu begegnen.

Allerdings sind Informationssysteme nicht die einfachsten Dinge, die man analysieren kann. Sie scheinen nie stillzustehen! Die Daten ändern sich, neue Integrationswege kommen hinzu, Systeme werden neu konfiguriert. Am ärgerlichsten ist jedoch, dass sich Benutzerkonten und -privilegien nahezu täglich ändern. Wenn die Risikobewertung abgeschlossen ist, sind die Ergebnisse meist bereits veraltet! Wie sollen wir das Risiko einschätzen, wenn sich alles um uns herum ständig verändert?

Die Antwort ist natürlich Automatisierung. Es gibt Teile der Risikobewertung, die nicht automatisiert werden können. Beispielsweise gibt es keine magische Methode zur automatischen Bewertung des Geschäftswerts von Datenbeständen. Einige Teile der Risikobewertung können jedoch automatisiert werden.

Hier kommen wir zurück zum Identitätsmanagement und der Governance. Fast alle Organisationen sind von Insider-Bedrohung betroffen, einer Bedrohung durch Personen, die bereits Teil der Organisation sind. Mitarbeiter, Auftragnehmer, Supporttechniker, Cloud-Dienstanbieter – sie haben bereits Zugriff auf die Daten. Diese Personen müssen nichts hacken, sie müssen keine Gegenmaßnahmen überwinden, sie sind bereits drin. Alles, was diese Personen brauchen, um die Geschäftsgeheimnisse preiszugeben, sind einfache Tastenkombinationen zum Kopieren und Einfügen. Ein Datei-Download genügt, um eine Datenbank zu verkaufen. Die Verbreitung von Cloud-Diensten macht solche „Exploits“ völlig

trivial. Es gibt keine technische Gegenmaßnahme, keinen Perimeter, der einen Insider davon abhalten könnte, ein Privileg auszunutzen, das er oder sie bereits besitzt.

Dies bedeutet, dass Identitätsdaten das Ergebnis der Risikomodellierung stark beeinflussen. Ein System, bei dem fast alle Ihre Mitarbeiter uneingeschränkten Zugriff auf eine Datenbank haben, stellt mit hoher Wahrscheinlichkeit ein viel höheres Risiko dar als alle netzwerkbezogenen Risiken zusammen. Eine einzelne Person, die über Administratorzugriff auf fast jedes System in der Organisation verfügt, ist sicherlich ein sehr attraktives Phishing-Ziel. In den Identitätsdaten fast aller Organisationen lauern hohe Risiken. Dieses Risiko lässt sich jedoch leicht reduzieren, indem die Zugriffsrechte angepasst werden. Doch wie findet man solche Risiken? Identitätsdaten sind oft komplex, systemspezifisch und in vielen Verzeichnissen, Cloud-Systemen und Anwendungsdatenbanken verteilt. Jeder Identitätsexperte kann sicherlich erkennen, wohin das führt: Natürlich zum Identitätsmanagementsystem.

Das Identitätsmanagementsystem ist ein idealer Ort für die Bewertung von Risiken im Zusammenhang mit Identitäts- und Zugriffsdaten. Wesentliche Daten sind bereits in der Datenbank des Identitätsmanagementsystems vorhanden: Benutzer, Rollen, Rollenzuweisungen, Rollenzusammensetzung, Berechtigungen, alles ist vorhanden. Berechtigungen können bewertet werden, um ihnen eine Risikobewertung zuzuweisen. Anschließend können die Daten in ein Risikomodell eingespeist werden, das bewertet, wie die Berechtigungen in Rollen zusammengefasst werden, wie die Rollen den Benutzern zugewiesen werden und Rollen und Benutzer mit hohem Risiko identifiziert werden. Das Modell wird von der Maschine schnell und effizient ausgewertet. Die Effizienz eröffnet eine ganze Reihe von Möglichkeiten, die den Kern eines risikobasierten Ansatzes zur Identity Governance ausmachen.

Da das von jedem einzelnen Benutzer ausgehende Risiko bewertet werden kann, kann leicht eine gefährliche Anhäufung von Berechtigungen in den Händen eines einzelnen Benutzers erkannt werden. Der Fokus kann dann auf der Minimierung dieses Risikos liegen, indem analysiert wird, warum sich die Privilegien angesammelt haben, ob sie alle notwendig sind, und überschüssige Privilegien zu entfernen,. Vielleicht kann über eine Änderung der Geschäftsprozesse nachgedacht werden, um die Verantwortlichkeiten auf mehrere Benutzer aufzuteilen und so das Risiko noch weiter zu senken. Das Risikomodell kann nach jedem Schritt bewertet und geprüft werden, ob bereits akzeptable Risikoniveaus erreicht wurden.

Das Risikomodell kann das Risiko jeder Rolle bewerten. Dies ermöglicht die Erkennung von Anomalien, wie z. B. der Zuweisung einer mächtigen Rolle mit hohem Risiko an einen normalen Benutzer, der angeblich ein geringes Risiko aufweist. Es ist wahrscheinlich, dass eine solche Rolle versehentlich zugewiesen wurde, oder es handelte sich möglicherweise um eine Rolle, die während eines Notfalls zugewiesen wurde und nie entfernt wurde. Die Aufhebung der Zuweisung einer solchen Rolle kann eine schnelle Möglichkeit sein, das Risiko zu verringern. Ein intelligentes System könnte mehrere Arten solcher Ausreißer vorschlagen, bei denen sich die Privilegien eines einzelnen Benutzers von der Umgebung abheben.

Sobald das Konzept des Risikos in einem System etabliert wurde, kann es in Sicherheitsrichtlinien verwendet werden. Beispielsweise ist es sinnvoll, für Hochrisikobenutzer ein stärkeres Kennwort zu verlangen oder, noch besser, automatisch eine Multi-Faktor-Authentifizierung für sie einzurichten. Es kann wünschenswert sein, die Berechtigungen von Benutzern mit hohem Risiko häufiger neu zu zertifizieren als die von Benutzern mit geringem

Risiko. Die Zuweisung einer neuen Rolle an einen Hochrisikobenutzer muss möglicherweise eine zusätzliche Genehmigungsphase durchlaufen. Die Richtlinien können das Risiko berücksichtigen. Dieser Ansatz wird oft als adaptive Sicherheit bezeichnet.

Noch mehr Vorteile ergeben sich, wenn der risikobasierte Ansatz auch auf andere Bereiche des Identitäts- und Access Managements angewendet wird. Beispielsweise kann es eine gute Idee sein, eine starke Authentifizierung für Benutzer mit hohem Risiko zu fordern und gleichzeitig eine schwächere Authentifizierung für Benutzer mit geringem Risiko zuzulassen. Dies kann auf viele Arten erreicht werden. Am einfachsten ist es vielleicht, dass das Identitätsmanagementsystem Risikobewertungen an die Benutzerprofile des Zugriffsmanagements weitergibt.

Risikobasiertes Identitätsmanagement und Governance sind in der Tat das Richtige. Allerdings steckt der Teufel im Detail, und die Realität ist viel schwieriger. Auf dieser Route gibt es versteckte Gefahren und dunkle Ecken:

- Eine Risikobewertung bedeutet nichts. Gar nichts. Die Ergebnisse sind veraltet, sobald sie vom Modell erzeugt werden. Sie müssen fortwährend und von jetzt an bis in alle Ewigkeit Beurteilungen vornehmen. Man nimmt die Ergebnisse einer Begutachtung, plant Gegenmaßnahmen, setzt sie um, nur um die ganze Arbeit noch einmal zu erledigen. Dies wird als „Sicherheitsprozess“ bezeichnet. Es hört nie auf.
- Die Risikobewertung ist fast immer subjektiv. Bei der Bewertung des Risikos einer individuellen Berechtigung werden wahrscheinlich subjektive Begriffe wie „gering“, „mittel“ und „hoch“ verwendet. Die subjektiven Begriffe verbergen sich oft hinter Scores, Zahlen, die den Eindruck erwecken, es handele sich um exakte Werte. In Wirklichkeit sind sie alles andere als genau.

  Die subjektive Risikobewertung ist so ziemlich eine Standardmethode. Manchmal werden objektive Risikomaße versucht, beispielsweise die Umrechnung des Risikos in einen Geldwert. Solche „objektiven“ Maßnahmen sind jedoch oft sehr irreführend und werden von Fachleuten für Informationssicherheit im Allgemeinen missbilligt. An einer subjektiven Risikoeinschätzung ist grundsätzlich nichts auszusetzen, solange man sich der Einschränkungen bewusst ist. Die vielleicht wichtigste Regel besteht darin, die Bewertung konsistent und verhältnismäßig zu halten. Berechtigungen, denen die Risikostufe „gering“ zugewiesen ist, sollten ungefähr das gleiche Risiko darstellen, und es sollte deutlich niedriger sein als alle Ansprüche, die als „mittleres“ Risiko gekennzeichnet sind.
- Es wird ein Risikomodell benötigt, das zur Organisation und Situation passt. Bei Risikomodellen gibt es keine Einheitslösung. Es gibt einfache Risikomodelle, die für schnelle Bewertungen in Organisationen mit geringen Sicherheitsanforderungen geeignet sind. Dann gibt es übermäßig komplexe Risikomodelle, die für Hochsicherheitsumgebungen konzipiert sind. Auch wenn ein Modell gefunden wird, das den Anforderungen entspricht, muss es noch verfeinert werden. Ein sofort einsatzbereites Risikomodell kann nicht einfach erworben werden, es würde niemals funktionieren.

- Ein Modell ist nicht Realität. Das Modell soll eine Annäherung an die Realität darstellen. Allerdings darf nicht davon ausgegangen werden, dass das Modell der Realität entspricht. Ein schlechtes Modell vermittelt ein falsches Sicherheitsgefühl und kann beispielsweise behaupten, dass alles grün ist, obwohl in Wirklichkeit große Gefährdungen vorhanden sein können.

Informationssicherheit ist ein ziemlich seltsames Gebiet. Es kann eindeutig nachgewiesen werden, dass das System unsicher ist, beispielsweise durch einen erfolgreichen Angriff auf das System. Dennoch kann nie vollständig nachgewiesen werden, dass das System sicher ist. Diese Einschränkung sorgt für große Verwirrung. Da kein Prozess für Informationssicherheit erworben werden kann, kann kein vorgefertigter risikobasierter Ansatz für die Identitätsverwaltung erworben werden, dieser muss erstellt werden. Eine fortschrittliche und intelligente Identity-Governance-Plattform ist dabei zweifellos eine große Hilfe. Allerdings ist eine solche Plattform nur ein Werkzeug. Selbst das intelligenteste und teuerste Werkzeug wird nicht die ganze Arbeit erledigen. Durch eine derartiges Werkzeug wird lediglich die Arbeit effizienter, aber es ist die Organisation die sie vorantreibt.

Ähnlich wie bei der Informationssicherheit ist „von der Stange“ auch beim Identitätsmanagement und der Governance meist nur eine Illusion. Was auch immer die kühnen Marketingaussagen sagen, man kann es nicht einfach kaufen und betreiben. Nein, nicht einmal in der Cloud. Sie können eine Identity-Governance-Plattform als Service kaufen, aber Sie können keine Identity-Governance kaufen.

Terminologie für Identitätsmanagement und Governance

Identitätsprofis, oft motiviert durch Marketingbedürfnisse, erfinden gerne neue Namen und verwenden sie, um dasselbe zu beschreiben. Daher werden viele sich überschneidende, überladene und ähnliche Begriffe verwendet.

Identity Management (IDM) wird normalerweise zur Beschreibung der Low-Level-Teile (Technologie) verwendet, während Identity Governance zur Beschreibung der High-Level-Teile (Business) verwendet wird. Dennoch ist die Grenze sehr fließend und viele IDM-Systeme bieten Governance-Funktionen, und viele Governance-Systeme stellen Funktionen auf niedriger Ebene bereit. Identity Governance and Administration (IGA) ist ein Begriff, der beide Teile zusammen beschreiben soll. Governance, Risikomanagement und Compliance (GRC) sind Begriffe, die in der Vergangenheit hauptsächlich zur Darstellung der hochrangigen Identitäts-Governance-Funktionalität verwendet wurden, später einfach als Identitäts-Governance bekannt.

Insgesamt ist die Terminologie sehr fließend. Anbieter verwenden ihre eigenen Begriffe und wählen oft eine überladene oder verwirrende Terminologie. Branchenanalysten und Berater fügen außerdem ihre eigenen Begriffe und Bedeutungen zu bestehenden Begriffen hinzu. Marketingbegriffe werden schneller erfunden, als sich die Dokumentation anpassen kann, was die Situation ziemlich unübersichtlich macht. Wir haben versucht, die Terminologie so präzise wie möglich zusammenzustellen und sie dennoch verständlich zu machen. Wir haben uns nach Möglichkeit dafür entschieden, der etablierten Branchenterminologie zu folgen, auch wenn viele Begriffe überladen und mehrdeutig sind. Wir wollten die Verwirrung jedoch nicht noch vergrößern, indem wir die Terminologie neu erfanden. Auf Unklarheiten im Text weisen wir bei

Bedarf hin. Zumindest versuchen wir, in diesem Buch eine einheitliche Terminologie zu verwenden. Im Zweifelsfall schauen Sie bitte im Glossar nach.

Komplettlösung Identitäts & Access Management

Eine umfassende Lösung für das Identitäts- und Zugriffsmanagement kann nicht mit nur einer einzigen Komponente aufgebaut werden. Es gibt kein einzelnes Produkt oder keine einzelne Lösung, die alle erforderlichen Funktionen bietet. Und da die Anforderungen so komplex und oft sogar widersprüchlich sind, ist es sehr unwahrscheinlich, dass es jemals ein einziges Produkt geben wird, das alles kann.

Um eine Komplettlösung zu erstellen, ist eine geschickte Kombination mehrerer Komponenten erforderlich. Die richtige Mischung der Zutaten für diese IAM-Gericht wird immer etwas anders sein, da keine zwei IAM-Lösungen gleich sind. Für jede praktische IAM-Bereitstellung sind jedoch drei grundlegende Komponenten erforderlich:

- Der Verzeichnisdienst oder ein ähnlicher Identitätsspeicher ist die erste Komponente. Dies ist die Datenbank, in der Benutzerkontoinformationen gespeichert werden. Die Konten werden dort in einer „sauberen“ Form gespeichert, die von anderen Anwendungen verwendet werden kann. Diese Datenbank wird dann von vielen Anwendungen gemeinsam genutzt, die eine Verbindung zu ihr herstellen können. Dieser Teil der Lösung wird normalerweise als replizierte LDAP-Servertopologie oder Active Directory-Domäne implementiert. Dies hat den Vorteil relativ geringer Kosten und hoher Verfügbarkeit. Es gibt jedoch eine große Einschränkung:

  Das Datenmodell muss einfach sein, sehr einfach. Und der Identitätsspeicher muss ordnungsgemäß verwaltet werden.

- AM ist eine zweite Hauptkomponente der Lösung. Es übernimmt die Authentifizierung und (Teil-)Autorisierung. Die Zugriffsverwaltung vereinheitlicht Authentifizierungsmechanismen. Wenn ein Authentifizierungsmechanismus im AM-Server implementiert ist, können alle integrierten Anwendungen problemlos davon profitieren. Es bietet außerdem Single Sign-On (SSO), zentralisiert Zugriffsprotokolle usw. Es ist eine sehr nützliche Komponente. Aber natürlich gibt es Einschränkungen. Das AM-System benötigt Zugriff auf Identitätsdaten. Daher benötigt es eine zuverlässige, sehr skalierbare und absolut konsistente Identitätsdatenbank als Backend. Dies wird normalerweise vom Verzeichnisdienst bereitgestellt. Leistung und Verfügbarkeit sind hier die offensichtlichen Hindernisse. Aber es gibt noch ein weiteres Hindernis, das weniger offensichtlich, aber genauso wichtig ist: die Datenqualität. Die Daten im Verzeichnisdienst müssen aktuell sein und ordnungsgemäß verwaltet werden. Aber das ist nur ein Teil des Bildes. Da die meisten Anwendungen einige Identitätsdaten lokal speichern, müssen diese Daten auch mit der Verzeichnisdatenbank synchronisiert werden. Kein AM-System kann dies gut genug leisten. Und es hat für AM überhaupt keinen Sinn, das zu tun. Das AM-System hat ganz andere architektonische Verantwortlichkeiten. Daher wird noch eine weitere Komponente benötigt.

- Das Identitätsmanagement ist die letzte, aber in vielerlei Hinsicht wichtigste Komponente. Dies ist das eigentliche Gehirn der gesamten Lösung. Das IDM-System pflegt die Daten. Es ist die Komponente, die verhindert, dass das gesamte System auseinanderfällt. Es stellt sicher, dass die Daten aktuell sind und den Richtlinien entsprechen. Es synchronisiert alle Identitätsdaten, die diese lästigen kleinen Anwendungen ständig erstellen. Es verwaltet Gruppen, Privilegien, Rollen, Organisationsstrukturen und alle anderen Dinge, die für das ordnungsgemäße Funktionieren des Verzeichnisses und der Zugriffsverwaltung erforderlich sind. Es sorgt für Ordnung im System. Und es ermöglicht lebenden und atmenden Systemadministratoren und Sicherheitsbeamten die Kontrolle über die gesamte Lösung zu behalten.

Das folgende Diagramm zeigt, wie alle diese Komponenten zusammenpassen.

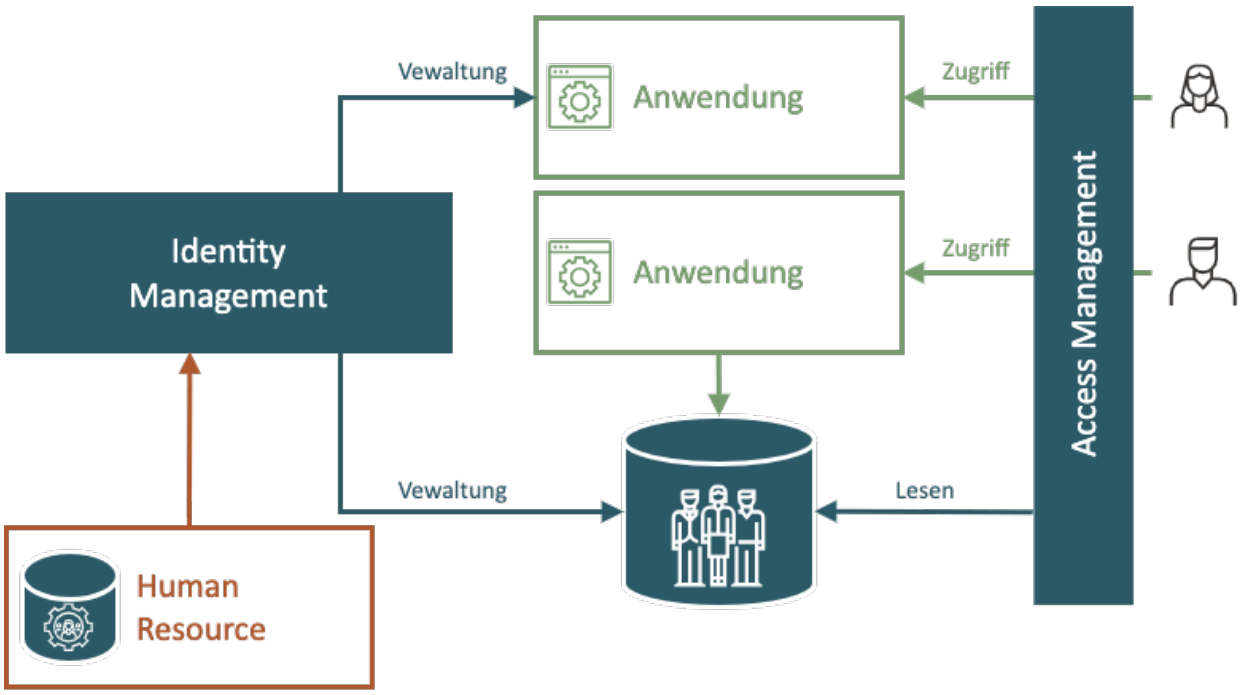

Abbildung 14: Komplettlösung Identitäts & Access Management

Es gibt mehrere Komponenten, die sehr unterschiedliche Merkmale und Eigenschaften aufweisen. Aber wenn man sie zu einer Lösung zusammenfügt, ist das Ergebnis etwas, das viel mehr ist als nur die Summe seiner Teile. Die Komponenten unterstützen sich gegenseitig. Die Lösung kann nur dann vollständig sein, wenn alle drei Komponenten vorhanden sind.

IAM und Sicherheit

Streng genommen passt Identity & Access Management (IAM) nicht vollständig in den Bereich der Informationssicherheit. Das IAM geht weit über die Informationssicherheit hinaus. IAM kann den Benutzerkomfort erhöhen, Betriebskosten senken, Prozesse beschleunigen und allgemein die Effizienz der Organisation verbessern. Darum geht es bei der Informationssicherheit nicht. Auch wenn IAM nicht unbedingt Teil der Informationssicherheit ist, gibt es dennoch große Überschneidungen. IAM befasst sich mit der Authentifizierung, Autorisierung, Prüfung, Rollenverwaltung und Governance von Objekten, die in direktem Zusammenhang mit der Informationssicherheit stehen. Daher stehen IAM und Informationssicherheit in einer engen und sehr komplizierten Beziehung zueinander.

Es ist vielleicht nicht übertrieben zu sagen, dass das IAM eine Voraussetzung für eine gute Informationssicherheit ist. Insbesondere der Teil des Identitätsmanagements (IDM) ist absolut kritisch – auch wenn dies auf den ersten Blick vielleicht nicht so offensichtlich ist. Aber die Beweise sprechen eindeutig dafür. In Sicherheitsstudien wird die Insider-Bedrohung durchgängig als eine der schwerwiegendsten Bedrohungen für eine Organisation eingestuft. Gegen die Insider-Bedrohung können die technischen Gegenmaßnahmen allerdings nicht viel ausrichten. Der Mitarbeiter, Auftragnehmer, Partner, Servicetechniker – sie alle erhalten einfach und legal Zugriffsrechte auf Ihre Systeme. Sie werden sogar die stärkste Verschlüsselung und Authentifizierung legal durchlaufen, weil sie über die Schlüssel verfügen. Firewalls und VPNs werden sie nicht aufhalten, da diese Personen sie passieren sollen, um ihre Arbeit zu erledigen.

Offensichtlich gibt es Schwachstellen. Und bei der Population von Tausenden von Benutzern ist es gut möglich, dass es auch einen Angreifer gibt. Vielleicht wurde gestern ein bestimmter Ingenieur entlassen. Er verfügt aber weiterhin über VPN-Zugriff und Administrationsrechte auf die Server. Und da er mit der Art und Weise, wie er behandelt wurde, möglicherweise nicht ganz zufrieden ist, ist die Wahrscheinlichkeit groß, dass er geneigt ist, Ihnen das Leben etwas schwerer zu machen. Vielleicht würde es helfen, einige Firmenunterlagen durchsickern zu lassen. Jetzt haben wir einen motivierten Angreifer, der sich durch keine Gegenmaßnahmen aufhalten lässt und problemlos an die Vermögenswerte gelangen kann. Jeder Sicherheitsbeauftragte kann das Ergebnis vorhersagen, ohne dass eine umfassende Risikoanalyse erforderlich ist.

Die Informationssicherheit hat keine klaren Antworten auf die Insider-Bedrohung. Und dieses Problem ist nicht einfach zu lös**en, d**a offensichtlich ein großer Sicherheitskompromiss besteht. Das Unternehmen möchte, dass Benutzer problemlos auf die Assets zugreifen können, um ihre Arbeit zu erledigen. Um die Räder einer Organisation am Laufen zu halten. Aber die Sicherheit muss die Vermögenswerte vor denselben Benutzern schützen. Und es gibt kein Allheilmittel, um dieses Problem zu lösen. Es gibt jedoch ein paar Dinge, die getan werden können, um die Situation zu verbessern:

- **Erfassen Sie, wer Zugriff auf was hat.**

  Jeder Benutzer verfügt über Konten in vielen Anwendungen im gesamten Unternehmen. Verfolgen Sie, welches Konto zu welchem Benutzer gehört. Es ist sehr schwierig, dies manuell zu tun. Aber selbst das schlechteste IDM-System kann das schaffen.

- **Zugriff schnell entfernen.**

  Bei einem Sicherheitsvorfall müssen die Zugriffsrechte in Sekundenschnelle entzogen werden. Wenn ein Mitarbeiter entlassen wird, müssen seine Konten daher innerhalb von Minuten deaktiviert werden. Für einen Systemadministrator ist es kein Problem, dies manuell zu tun. Aber wird der Administrator bei einem Sicherheitsvorfall spät in der Nacht verfügbar sein? Würden Sie Entlassungen mit der Arbeitszeit von Systemadministratoren synchronisieren? Würden Systemadministratoren nicht vergessen, alle Prozesse und Hintergrundjobs zu stoppen, die der Benutzer möglicherweise zurückgelassen hat? Das IDM-System kann das problemlos tun. Das Sicherheitspersonal kann mithilfe des IDM-Systems einfach alle Konten deaktivieren. Ein Klick genügt.

- **Richtlinien durchsetzen.**

  Behalten Sie den Überblick über die Berechtigungen, die Benutzern zugewiesen wurden. Dies bedeutet normalerweise, dass die Zuweisung von Rollen (und anderen Berechtigungen) an Benutzer verwaltet wird. Stellen Sie sicher, dass die Zuweisung vertraulicher Rollen genehmigt wird, bevor der Benutzer die Berechtigungen erhält. Vergleichen Sie die Richtlinien mit der Realität. Systemadministratoren, die Konten erstellen und Berechtigungen zuweisen, sind keine Roboter. Fehler können passieren. Stellen Sie sicher, dass die Fehler entdeckt und behoben werden. Dies ist die natürliche beste Vorgehensweise. Aber es ist fast unmöglich, dies manuell zu tun. Doch selbst ein durchschnittliches IDM-System schafft das problemlos.

- **Entzug unnötiger Rollen und Berechtigungen.**

  Rollenzuweisungen und Berechtigungen häufen sich im Laufe der Zeit an. Langjährige Mitarbeiter haben häufig Zugriff auf nahezu alle Vermögenswerte, einfach weil sie die Daten irgendwann in ihrer Karriere benötigten. Und der Zugriff auf den Vermögenswert wurde seitdem nie aufgehoben. Dies stellt ein großes Sicherheitsrisiko dar. Dies kann durch die Einführung eines papierbasierten Prozesses zur Überprüfung der Ansprüche gemildert werden. Dieser Prozess ist jedoch sehr langsam, kostspielig, fehleranfällig und muss in regelmäßigen Abständen wiederholt werden. Aber fortgeschrittene IDM-Systeme unterstützen bereits die Automatisierung dieses Prozesses mittels Re-Zertifizierung.

- **Ordnung wahren.**

  Wenn Sie sich genau an das Prinzip der minimalen Rechte halten, haben Sie wahrscheinlich erkannt, dass Sie mehr Rollen als Benutzer haben. Rollen sind abstrakte Konzepte und entwickeln sich ständig weiter. Selbst erfahrene Sicherheitsexperten können sich leicht in den Rollenhierarchien und -strukturen verlieren. Der normale Endbenutzer hat oft überhaupt keine Ahnung, welche Rollen er benötigt. Dennoch ist es nicht so schwer, die Rollen nach Kategorien zu sortieren, wenn man sie in einem guten IDM-System verwaltet. Dadurch entsteht ein Rollenkatalog, der viel einfacher zu verstehen, zu verwenden und zu pflegen ist.

- **Dranbleiben.**

  Führen Sie ein Prüfprotokoll über jede Änderung von Berechtigungen. Dies bedeutet, dass Sie den Überblick über alle neuen Konten, Kontoänderungen, Löschungen, Benutzer- und Kontoumbenennungen, Rollenzuweisungen und -aufhebungen, Genehmigungen, Änderungen der Rollendefinitionen, Richtlinienänderungen usw. behalten. Dies ist eine große Aufgabe, die manuell erledigt werden muss. Und es ist fast unmöglich, Fehler zu vermeiden. Aber eine Maschine kann das einfach und zuverlässig erledigen.

- **Suche nach Schwachstellen.**

  Fehler passieren. Systemadministratoren erstellen häufig Testkonten zur Fehlerbehebung. Und es gibt eine alte Tradition, für solche Konten triviale Passwörter festzulegen. Diese Konten werden nach Abschluss der Fehlerbehebung nicht immer bereinigt. Und es kann schlimmere Fehler geben. Systemadministratoren können einem falschen Benutzer Berechtigungen zuweisen. Der Helpdesk aktiviert möglicherweise ein Konto, das dauerhaft deaktiviert werden

sollte. Daher müssen alle Anträge permanent auf Konten überprüft werden, die nicht vorhanden sein sollten, und auf Berechtigungen, die nicht zugewiesen werden sollten. Das ist einfach zu viel Arbeit, um sie manuell zu erledigen. Dies ist nicht wirklich machbar, es sei denn, eine Maschine kann das gesamte System automatisch scannen. Dies wird als Abstimmung bezeichnet und ist eine der Grundfunktionen eines jeden anständigen IDM-Systems.

Theoretisch können alle diese Dinge manuell erledigt werden. Aber es ist in der Praxis nicht umsetzbar. Die Realität ist, dass die Informationssicherheit ernsthaft leidet – es sei denn, es gibt ein IDM-System, das Automatisierung und Transparenz bietet. Eine gute Informationssicherheit ohne IDM-System ist kaum möglich.

Zero-Trust-Ansatz

Zero-Trust ist ein Ansatz zur Gestaltung von Netzwerk- und Anwendungssystemen. Die Grundidee besteht darin, dass ein System keinem anderen System implizit vertrauen sollte, auch nicht Systemen, die sich in einem „sicheren“ Unternehmensnetzwerk befinden. Vereinfacht gesagt geht es beim Zero-Trust-Ansatz vor allem um die Entfernung von Sicherheitsgrenzen.

Viele Jahrzehnte lang wurden Unternehmensnetzwerke nach dem Ansatz „hartes Äußeres und weiches Inneres“ entworfen. Das Unternehmensnetzwerk wurde durch eine Armee spezialisierter Sicherheitssysteme und -techniken vor dem Internet geschützt, wie Firewalls, entmilitarisierte Zonen, Netzwerkverkehrsanalysatoren, Intrusion-Detection-Systeme, Netzwerk-Antivirenscanner und alles andere, was ein boomender Netzwerksicherheitsmarkt bieten kann. Während die Burgtore stark geschützt waren, war das Innere des Unternehmensnetzwerks sehr weich. Ursprünglich gab es überhaupt keine Sicherheitsmaßnahmen innerhalb von Unternehmensnetzwerken. Jeder, der hineinkam, konnte sich mit jedem System verbinden. Natürlich waren in der Regel grundlegende Mechanismen zur Authentifizierung und Autorisierung vorhanden, aber das Netzwerk war nicht segmentiert und der Datenverkehr war in der Regel nicht einmal durch grundlegende Verschlüsselung geschützt. Durch diesen Ansatz wurde ein Sicherheitsperimeter um das Unternehmensnetzwerk herum geschaffen. Wenn Sie die Sicherheit der Daten gewährleisten möchten, müssen Sie sicherstellen, dass niemand in das Netzwerk gelangt.

Natürlich macht dieser Ansatz im Internetzeitalter nicht wirklich Sinn. So etwas wie einen Netzwerkperimeter gibt es nicht mehr, jedenfalls nicht seit der Erfindung von WLAN, mobilen Daten und USB-Sticks. Es ist lächerlich einfach, den Perimeter zu durchbrechen, indem man ein WLAN-Gerät mit dem Unternehmensnetzwerk verbindet. Doch auch das war in der Regel nicht nötig, da die Daten auf USB-Sticks kopiert oder mithilfe eines VPN-Zugriffs (Virtual Private Network) problemlos außerhalb des Perimeters verschoben werden können. Unternehmenssicherheitsexperten versuchten, solchen Bedrohungen zu begegnen, indem sie die Geräte der Benutzer streng kontrollierten, z. B. USB-Anschlüsse deaktivierten, den Zugriff auf andere Netzwerke deaktivierten, während das Gerät Teil des VPN war, usw. Keine dieser Gegenmaßnahmen war jedoch wirklich effektiv und für die Benutzer meist sehr aufdringlich und unbequem. Das Aufkommen von Cloud-Diensten und mobilen Geräten war der letzte Tropfen, der traditionelle Ansatz der Unternehmensinformationssicherheit war tot. Selbst die traditionellsten Sicherheitsexperten mussten zugeben, was bereits offensichtlich war: Es gibt keinen Perimeter.

Der alte Ansatz wurde durch einen neuen ersetzt: Zero Trust. Eine Anwendung sollte keiner anderen Anwendung vertrauen, nicht einmal einer Anwendung im selben Unternehmensnetzwerk. Der Netzwerkperimeter wurde durch gegenseitige Authentifizierung und Schutz des Netzwerkverkehrs ersetzt. Jede Anwendung muss das andere Ende der Verbindung authentifizieren, unabhängig davon, ob sie mit der Partei kommuniziert, mit der sie kommunizieren soll. Der Netzwerkverkehr muss immer verschlüsselt und authentifiziert (signiert) werden, wobei immer davon auszugehen ist, dass er über ein unsicheres Netzwerk geleitet wird. Vereinfacht gesagt werden interne Netzwerke genauso wie das öffentliche Internet behandelt.

Das Konzept des Zero Trust ist überhaupt nicht neu. Es gab sie schon seit den Anfängen der Informationssicherheit. Sicherheitsexperten werden darin geschult, nichts und niemandem zu vertrauen, Richtlinien einzurichten, eine Authentifizierung zu verlangen, den Zugriff standardmäßig zu verweigern, den gesamten Netzwerkverkehr zu verschlüsseln, Privilegien zu minimieren, Risiken zu verwalten und so weiter. Daher ist der „Zero Trust“-Ansatz im Wesentlichen nur eine gründliche Anwendung angemessener Grundsätze der Informationssicherheit. Diesen Ansatz gibt es schon seit Jahrzehnten, er hatte nur unterschiedliche Namen: Tiefenverteidigung, perimeterlose Sicherheit, Netzwerkhärtung und so weiter.

Auch wenn der Zero-Trust-Ansatz nicht neu ist, hat er im Zeitalter von Cloud-Diensten und Fernzugriff dramatisch an Bedeutung gewonnen. Viele Funktionen traditioneller Anwendungen werden mittlerweile von Cloud-Diensten bereitgestellt. Solche Anwendungen benötigen Daten, um zu funktionieren, daher stellt die bloße Nutzung von „As-a-Service“-Anwendungen an sich schon eine Verletzung des Perimeters dar. Cloud-Anwendungen müssen mit On-Premise-Anwendungen zusammenarbeiten. Herkömmliche Integrationsmuster, die auf Soft Interior basieren, funktionieren in dieser schönen neuen Welt nicht mehr. All das treibt die Zero-Trust-Konzepte voran.

Natürlich ist „Zero Trust“ eher ein Wunsch als eine strikte Regel. Ein System, das auf nichts vertraut, wird überhaupt nicht funktionieren können. Selbst beim Zero-Trust-Ansatz ist immer ein gewisses Maß an Vertrauen im Spiel. Das System muss darauf vertrauen, dass die Stammschlüssel und Zertifikate authentisch sind. Es besteht das implizite Vertrauen, dass die Entwickler keine Hintertür in den Systemcode eingeführt haben. Der Zero-Trust-Ansatz sollte vielleicht als „Minimal-Trust-Ansatz“ bezeichnet werden. Allerdings ist „Zero Trust“ in Hochglanzmagazinen und Präsentationen deutlich attraktiver. Wie auch immer der Ansatz heißt, das Grundprinzip bleibt dasselbe: implizites Vertrauen in das System aggressiv minimieren.

Identität durchdringt alles, daher hat der Zero-Trust-Ansatz auch Auswirkungen auf das Identitäts- und Zugriffsmanagement. Die Auswirkungen auf Zugangsverwaltungstechnologien sind vielleicht ziemlich offensichtlich. Bei der Zugriffsverwaltung geht es hauptsächlich um die Authentifizierung. In dieser neuen Zero-Trust-Welt müssen sich die Anwendungen gegenseitig authentifizieren. Daher müssen die Zugriffsverwaltungssysteme die Authentifizierung von Nichtpersonenidentitäten wie Anwendungen und Geräten übernehmen. Viele Szenarien weichen nicht wesentlich von der üblichen Authentifizierung ab. Der einzige Unterschied besteht möglicherweise darin, dass die Authentifizierung vollständig nicht interaktiv sein muss. Es gibt jedoch auch komplexere Authentifizierungsszenarien, beispielsweise eine Anwendungsauthentifizierung im Namen eines Benutzers. Herkömmliche

Authentifizierungsmethoden (z. B. passwortbasierte Authentifizierung) sind für solche Szenarien offensichtlich schlecht vorbereitet. Daher wird der Zero-Trust-Ansatz häufig mit der Einführung neuer Authentifizierungsmechanismen kombiniert.

Die Auswirkungen des Zero-Trust-Ansatzes auf Identitätsmanagementsysteme sind viel subtiler. Der Zero-Trust-Ansatz erfordert eine gegenseitige Authentifizierung der Kommunikationsparteien. Das bedeutet, dass das Identitätsmanagementsystem nicht-personenbezogene Identitäten wie Anwendungen und Geräte verwalten muss. Diese Anforderung ist nicht neu, daher sind die meisten gut gewarteten Identitätsmanagementprodukte in dieser Hinsicht mehr als leistungsfähig. Der problematische Teil ist meist der Zusammenhang zwischen den Identitäten, die Beziehung. Wenn zwei Anwendungen kommunizieren müssen, müssen beide über die jeweils andere Bescheid wissen. API-Schlüssel, Pre-Shared Secrets, Zertifikate und anderes kryptografisches Material müssen eingerichtet werden, bevor die erste Kommunikation stattfinden kann. Die Anmeldeinformationen müssen aktualisiert werden, Schlüssel müssen regelmäßig geändert werden, Zertifikate müssen erneuert werden. Solche Anwendungszugangsdaten sind viel wichtiger als Passwörter es jemals waren, da Anwendungszugangsdaten im wahrsten Sinne des Wortes die Schlüssel zum Königreich sind. Ein Angreifer, der Zugriff auf die Anwendungsschlüssel erhält, hat Zugriff auf alle Daten in Cloud-Anwendungen, also höchstwahrscheinlich auf Ihre Gehaltsabrechnungsdaten, Kundendatenbanken, internen Dokumente und fast alles, was für Sie wichtig ist. Erfahrene Experten für Informationssicherheit wissen, dass nicht die Verschlüsselung der schwierigste Teil der Kryptographie ist. Es ist die Verwaltung der Schlüssel. Ebenso ist es nicht die Authentifizierung, die das schwierigste Problem beim Zero-Trust-Ansatz darstellt. Das Identitäts- und Anmeldeinformationsmanagement ist viel schwieriger. Heutzutage werden Anwendungsidentitäten und Anmeldeinformationen normalerweise manuell von Systemadministratoren konfiguriert. Allerdings ist ein solcher Ansatz nicht skalierbar. Die gesamte Idee von „as a Service“-Anwendungen besteht darin, Informationssysteme flexibler, dynamischer und vor allem weniger anspruchsvoll in Bezug auf die Systemadministration zu machen. Die manuelle Verwaltung von Anwendungsidentitäten steht im Widerspruch zu dieser Idee.

Was kann ein Identitätsmanagementsystem für den Zero-Trust-Ansatz tun?

Erstens kann es das Gleiche tun, was es normalerweise tut. Es kann Benutzeridentitäten verwalten. Im Zero-Trust-Modus müssen sich Benutzer überall authentifizieren, sie müssen überall, in jeder Anwendung oder jedem Dienst autorisiert werden. Während die Authentifizierung in der Regel über Zugriffsverwaltungs- oder Single-Sign-On-Systeme erfolgen kann, ist die Autorisierung deutlich schwieriger. Ein Identitätsmanagementsystem kann dies tun, es kann Berechtigungen und Privilegien verwalten, es kann unnötige Konten und Zugriffsrechte entziehen. Dieser Teil ist relativ einfach und wird von Identitätsmanagementsystemen bereits seit Jahrzehnten geleistet.

Zweitens kann das Identitätsmanagementsystem den Zugriff einer Anwendung auf eine andere Anwendung verwalten. Dies bedeutet die Verwaltung von Anwendungskonten, die Verwaltung von Anwendungsanmeldeinformationen und Berechtigungen sowie die Aufhebung der Bereitstellung der Konten, wenn die Anwendung außer Betrieb genommen wird. Das hört sich vielleicht einfach an, ist aber in Wirklichkeit alles andere als einfach. Die grundlegendste Voraussetzung ist ein Anwendungsinventar, eine verlässliche Liste der Anwendungen in einer

Organisation. Viele Organisationen verfügen jedoch überhaupt nicht darüber. Die Organisationen, die darüber verfügen, verfügen in der Regel über eine informelle Tabellenkalkulation, die nicht vollständig maschinenlesbar ist. Wie soll ein Identitätsmanagementsystem überhaupt mit der Verwaltung von Anwendungsidentitäten beginnen, wenn es keine maßgebliche Quelle gibt? Daher sollte der Aufwand mit dem Aufbau einer solchen Quelle beginnen, bei der es sich um manuell gepflegte Informationen im Identitätsmanagementsystem handeln kann. Anschließend müssen der Anwendung Anwendungskonten und Berechtigungen (Gruppen) zugeordnet werden. Viele Anwendungen haben Eigentümer, die im IDM-System verwaltet werden können. Da Anwendungskonten und Berechtigungen mit einer Anwendung verknüpft sind, können sie automatisch aufgehoben werden, wenn die Anwendung außer Betrieb genommen wird. Der vielleicht schwierigste Teil ist jedoch die Verwaltung der Anmeldeinformationen. Die Anmeldeinformationen sollten regelmäßig geändert werden. Allerdings muss diese Änderung auf beiden Seiten des Kommunikationskanals (sowohl Client- als auch Serverteil) synchronisiert werden, sonst kommt es zu einem Ausfall. Dies lässt sich selbst mit dem modernen IDM-System nur schwer automatisieren.

Insgesamt sind aktuelle Identitätsmanagement- und Governance-Plattformen nicht für die vollständige Verwaltung von Anwendungen, Anwendungskonten und Berechtigungen geeignet. In der Regel steht nur eine Teilfunktionalität zur Verfügung. Obwohl die Konzepte des Zero-Trust-Ansatzes recht alt sind, wurden sie nie systematisch und in großem Maßstab angewendet. Daher bestand kein ausreichender Bedarf, erforderliche Funktionen in Identitätsmanagementsystemen zu implementieren. Die Zukunft wird zeigen, ob die aktuelle Welle des Zero-Trust-Hypes eine solche Nachfrage mit sich bringt. Allerdings entwickeln sich nahezu alle IDM-Systeme stark weiter und entwickeln neue Funktionen. Daher ist es von entscheidender Bedeutung, eine IDM-Plattform auszuwählen, die sich weiterentwickeln lässt.

Aufbau einer Identity & Access Management Lösung

Es gibt keine einzige Lösung für das Identitäts- und Zugriffsmanagement, die für jeden geeignet ist. Jede Bereitstellung hat spezifische Anforderungen und Merkmale. Der Einsatz in einer Großbank wird sich wahrscheinlich auf Governance, Rollenmanagement und Sicherheit konzentrieren. Bei der Bereitstellung in kleinen Unternehmen steht die Kosteneffizienz im Vordergrund. Der Cloud-Anbieter wird sich auf Skalierbarkeit, Benutzererfahrung und Einfachheit konzentrieren. Einfach ausgedrückt: Eine Einheitsgröße passt nicht für alle. Fast alle IAM-Lösungen nutzen die gleichen Hauptkomponenten. Produktauswahl, Lösungstopologie und Konfiguration können jedoch erheblich variieren.

Föderiertes Identity & Access Management

In der IT läuft das föderierte Identitätsmanagement darauf hinaus, über einen gemeinsamen Satz von Richtlinien, Praktiken und Protokollen zu verfügen, um die Identität und das Vertrauen in IT-Benutzer und -Geräte in allen Organisationen zu verwalten.

Zur Unterstützung der Benutzer- und Datensicherheit wurden zentralisierte IDM-Lösungen entwickelt, bei denen sich der Benutzer und die Systeme, auf die der Benutzer zugreift, innerhalb desselben Netzwerks – oder zumindest derselben „Kontrolldomäne“ – befanden. Allerdings greifen zunehmend Benutzer auf externe Systeme zu, die grundsätzlich außerhalb ihres Einflussbereichs liegen, und externe Benutzer greifen auf interne Systeme zu. Die zunehmende Trennung des Benutzers von den Systemen, die Zugriff benötigen, ist ein unvermeidliches Nebenprodukt der Dezentralisierung, die durch die Integration des Internets in alle Aspekte des Privat- und Geschäftslebens entsteht. Die sich weiterentwickelnden Herausforderungen beim Identitätsmanagement und insbesondere die Herausforderungen im Zusammenhang mit organisationsübergreifendem Zugriff haben zu einem neuen Ansatz für das Identitätsmanagement geführt, der heute als „Föderiertes Identity & Access Management“ bekannt ist.

> ***Terminologie***
> Der Begriff Föderiertes Identity & Access Management (F-IAM) umfasst sowohl die Mechanismen zur Zugriffssteuerung (Access Management) als auch die Bereitstellung von Benutzerkonten (Identity Management).
>
> In der Literatur wird zumeist der Begriff „Föderiertes Identity Management“ verwendet ohne näher darauf einzugehen, dass unter diesem Begriff lediglich die Mechanismen der Zugangskontrolle und Zugangsteuerung verstanden werden.
>
> Es handelt sich also vereinfacht ausgedrückt um ein föderiertes Verfahren zum Single Sign On.
>
> Die in diesem Kontext notwendigen Prozesse zur Verwaltung des Lebenszyklus von Benutzerkonten, insbesondere der korrekten Terminierung der Benutzerkonten werden hingegen beiläufig oder garnicht betrachtet.

F-IAM oder „Föderation“ der Identität beschreibt die Technologien, Standards und Anwendungsfälle, die dazu dienen, die Portabilität von Identitätsinformationen über ansonsten autonome Sicherheitsdomänen hinweg zu ermöglichen. Das ultimative Ziel der Identitätsföderation besteht darin, Benutzern einer Domäne den sicheren Zugriff auf Daten oder Systeme einer anderen Domäne zu ermöglichen, nahtlos und ohne die Notwendigkeit einer vollständig redundanten Benutzerverwaltung. Die Identitätsföderation gibt es in vielen Varianten, darunter „benutzergesteuerte“ oder „benutzerzentrierte“ Szenarien sowie organisationsgesteuerte oder Business-to-Business-Szenarien.

Der Einsatz von Standards kann die Kosten senken, da die Notwendigkeit der Skalierung einmaliger oder proprietärer Lösungen entfällt. Es kann die Sicherheit erhöhen und das Risiko verringern, indem es einer Organisation ermöglicht, einen Benutzer einmal zu identifizieren und zu authentifizieren und diese Identitätsinformationen dann in mehreren Systemen, einschließlich externen Partner-Applikationen, zu verwenden. Es kann die Einhaltung der

Datenschutzbestimmungen verbessern, indem es dem Benutzer ermöglicht, zu steuern, welche Informationen weitergegeben werden, oder indem die Menge der weitergegebenen Informationen begrenzt wird. Und schließlich kann es das Endbenutzererlebnis drastisch verbessern, indem es die Notwendigkeit einer neuen Kontoregistrierung durch automatisierte Bereitstellung von Benutzerkonten im „Verbund“ oder die Notwendigkeit einer redundanten Anmeldung durch organisationsübergreifendes SSO überflüssig macht.

Der Begriff des F-IAM ist äußerst weit gefasst und entwickelt sich auch weiter. Dabei kann es sich um Benutzer-zu-Benutzer- und Benutzer-zu-Anwendungs- sowie Anwendung-zu-Anwendungs-Szenarien sowohl auf der Ebene eines Webbrowser als auch auf der Ebene der Webdienste oder der serviceorientierten Architektur (SOA) handeln. Dabei kann es sich sowohl um Szenarien mit hoher Vertrauenswürdigkeit und hoher Sicherheit als auch um Szenarien mit geringer Vertrauenswürdigkeit und geringer Sicherheit handeln. Die Ebenen der Sicherung von Identitäten, die für ein bestimmtes Szenario erforderlich sein können, werden ebenfalls durch ein gemeinsames und offenes Identity Assurance Framework standardisiert. Dabei kann es sich sowohl um benutzerzentrierte Anwendungsfälle als auch um unternehmenszentrierte Anwendungsfälle handeln. Der Begriff „Identitätsföderation“ ist von Natur aus ein allgemeiner Begriff und nicht an ein bestimmtes Protokoll, eine bestimmte Technologie, eine bestimmte Implementierung oder ein bestimmte Organisation gebunden. Identitätsföderationen können bilaterale Beziehungen oder multilaterale Beziehungen sein. Im letzteren Fall tritt die multilaterale Föderation häufig in einem vertikalen Segment auf, beispielsweise in der Strafverfolgung (wie der National Identity Exchange Federation in den USA) sowie in Forschung und Bildung (wie InCommon). Wenn die Identitätsföderation bilateral ist, können die beiden Parteien die notwendigen Metadaten austauschen, um die Beziehung umzusetzen. In einer multilateralen Föderation ist der Metadatenaustausch zwischen den Teilnehmern eine komplexere Angelegenheit.

Eine Sache, die jedoch konsistent ist, ist die Tatsache, dass der Begriff Föderation Methoden der Identitätsportabilität beschreibt, die auf offene, oft auf Standards basierende Weise erreicht werden – was bedeutet, dass jeder, der sich an die offene Spezifikation oder den offenen Standard hält, das gesamte Anwendungsspektrum nutzen kann.

Die Identitätsföderation kann auf verschiedene Arten erreicht werden. Einige davon erfordern die Verwendung formaler Internetstandards wie der OASIS Security Assertion Markup Language (SAML)-Spezifikation, andere können Open-Source-Technologien und/oder andere veröffentlichte Technologien umfassen Spezifikationen (z. B. Information Cards, OpenID, das Higgins Trust Framework oder Novells Bandit-Projekt).

Definiton

Die für die Organisationsform Föderation geltenden Leitgedanken sind auch für die technische Domäne des föderativen Identitätsmanagements von Belang. Ein hierbei entscheidender Faktor ist, dass die unterschiedlichen Föderationspartner zwar Dienste anderer Föderationspartner nutzen und hierdurch eine inhärente Abhängigkeit entsteht, allerdings ist diese geringer als bei einem vollständig zentralisierten Ansatz. Aus diesem Grund sollte eine sehr wesentliche Eigenschaft des föderativen Identitätsmanagements, der bis zu einem gewissen Grad geltende Autonomiegedanke sein. Dieser wird auch in existierenden Definitionen, beispielsweise durch das BSI, berücksichtigt:

„Das föderative Identitätsmanagement vereint mehrere administrativ unabhängige Identitätsmanagementsysteme in einem Vertrauenszirkel, in dem Identitätsinformationen ausgetauscht und gemeinsam genutzt werden können."

gemäß [BSI SOA-Security-Kompendium 2009]

Die Autonomie in Form einer administrativen Unabhängigkeit hat letztendlich wesentliche Auswirkungen auf die Verteilung identitätsbezogener Informationen.

Die Vertrauensbeziehung, welche durch Verträge und Regelwerke spezifiziert und festgehalten wird, spielt innerhalb des föderativen Identitätsmanagements ebenfalls eine entscheidende Rolle.

Demnach zählen zu einer Föderation auch alle Geschäftsabkommen, kryptografischen Grundlagen zur Sicherstellung des Vertrauens, sowie Benutzeridentifikatoren und Attribute, welche zwischen den unterschiedlichen dezentralen Sicherheits- und Richtliniendomänen ausgehandelt werden.

Demnach ist eine Föderation eine Ansammlung von „realms", in welcher Anbieter von Ressourcen auf der Basis einer Identität und assoziierter Attribute, die in einem anderen realm ausgestellt wurden, Zugriff auf eine Ressource ermöglichen. Die Föderation basiert auf einer Vertrauensbeziehung, so dass eine fundierte Zugriffskontrolle auf die durch den Föderationspartner ausgestellte Identität und Attribute erfolgen kann. Diese Definition deckt demnach sowohl das Vertrauen in die Kompetenz des Föderationspartners, den Delegationsgedanken, als auch die Dezentralität ab.

Föderation

Zusammenfassend lässt sich eine Föderation im Kontext des Programms P20 folgendermaßen definieren:

> ***Terminologie***
> Eine Föderation ist ein Zusammenschluss selbstständiger Organisationen, die auf der Basis eines Regelwerks und einem zugrunde liegenden Vertrauen Informationen mit anderen Föderationspartnern austauschen, Aufgaben meist in Form von Dienstnutzung an Föderationspartner delegieren und selbst Aufgaben übernehmen.

Föderatives Identitätsmanagement

Aufbauend auf dieser Definition lässt sich das föderative Identitätsmanagement folgendermaßen definieren:

> ***Terminologie***
> Das föderative Identitätsmanagement (FIM, F-IM, F-IdM) ist die verteilte Verwaltung von Identitätsinformationen auf der Basis einer Föderation, mit dem Ziel eine bestehende organisatorische und technologische Heterogenität zu erhalten und durch die richtige Balance zwischen Autonomie und Delegation identitätsbezogener Aufgaben auf der Basis von Dienstgütevereinbarungen und einem beidseitigen Vertrauen zwischen Dienstgeber und Dienstnehmer gemeinschaftlich identitätsbezogene Aufgaben zu lösen.

Föderiertes Access Management

Föderiertes Access Management (F-AM) ist ein hochentwickeltes Cybersicherheits- und Informationstechnologiesystem. Es ermöglicht Benutzern verschiedener Organisationen oder Domänen den Zugriff auf gemeinsame Netzwerke, Anwendungen und Ressourcen mit einem einheitlichen Satz von Anmeldeinformationen.

Anstatt dass jede Organisation separate Authentifizierungssysteme unterhält, ermöglicht F-AM die Verbindung dieser Systeme über einen zentralen Authentifizierungsmechanismus.

Jede am F-AM teilnehmende Organisation behält die Kontrolle über ihr Identitätsmanagementsystem. Diese Systeme werden dann mit einer Plattform die von Dritten angeboten wird verknüpft, die als Identity Provider (IdP) bekannt ist. Der IdP speichert und verwaltet Benutzeranmeldeinformationen sicher und erleichtert so die Authentifizierung und Autorisierung über mehrere Organisationen oder Domänen hinweg.

Der Hauptvorteil von F-AM liegt in seiner Fähigkeit, Benutzern den Zugriff auf Ressourcen in verschiedenen Domänen zu ermöglichen, ohne sich wiederholt anmelden zu müssen. Nach der Authentifizierung innerhalb ihrer Domäne können Benutzer nahtlos auf Ressourcen in anderen Verbunddomänen zugreifen.

Sicherheitsbedenken

Der Wechsel zu föderierten Identitäten als Alternative zu älteren Authentifizierungsmethoden ist nicht ohne Risiken. Die meisten Organisationen, die eine Föderation einführen, tun dies nur für eine Handvoll Anwendungen und haben Schwierigkeiten, ein Netzwerk aufzubauen, in dem über eine einzige Identität auf alle Programme zugegriffen werden kann. Dadurch sind einige Bereiche des Netzwerks allgemeinen Sicherheitsrisiken ausgesetzt, einschließlich Sicherheitsverletzungen durch die Verwendung schwacher Passwörter. Erschwerend kommt hinzu, dass in vielen Organisationen keine Pläne zur Einführung eines föderierten Identitätsmanagement vorhanden sind. Die rasante Verbreitung der Technologie hat dazu geführt, dass Unternehmen nicht in der Lage sind, das erforderliche Niveau zu dessen Verwaltung zu implementieren, um die Sicherheit auf breiter Front zu gewährleisten.

Damit föderierte Identitäten funktionieren, müssen Benutzerinformationen an den mit der Authentifizierung betrauten Dritten weitergegeben werden. Die Art dieser Informationen und die Art und Weise, wie sie weitergegeben, verarbeitet, gespeichert und geschützt werden, hat Auswirkungen auf die Sicherheit und Privatsphäre der Benutzer. Nicht alle Anbieter innerhalb einer Föderation erfüllen die gleichen Sicherheitsstandards, und die Nutzung mehrerer Anbieter

schafft zusätzliche Schwachstellen. Unternehmen müssen die von Drittanbietern verwendeten Sicherheitsprotokolle und Compliance-Maßnahmen verstehen, bevor sie Partnerschaften eingehen.

Insider-Bedrohungen und Identitätsdiebstahl, zwei häufige und besorgniserregende Sicherheitsbedenken für moderne Unternehmen, bleiben selbst bei Verwendung eines föderierten Systems problematisch. Unternehmen müssen sich der Vertrauenswürdigkeit der Benutzer im Netzwerk absolut sicher sein und über Authentifizierungsprotokolle verfügen, die sicherstellen, dass jeder Benutzer der ist, für den er sich ausgibt. Um das Risiko menschlichen Versagens zu minimieren, ist eine Schulung der Mitarbeiter erforderlich, da ein einziger kompromittierter Satz an Verbundanmeldeinformationen Hackern Zugriff auf mehrere Anwendungen gewähren und dazu führen kann, dass sich ein Verstoß schnell über ein Netzwerk ausbreitet.

Eine unsachgemäße Bereitstellung, die zu einem Privilege Creep führt, kann auch die Tür für verheerende Verstöße offen lassen. Die föderierte Identität eines Benutzers sollte nur die Zugriffsebene zulassen, die für seinen Job erforderlich ist, und jeder vorübergehende Zugriff, der für kurzfristige Projekte erforderlich ist, sollte widerrufen werden, sobald er nicht mehr benötigt wird. Automatisierte Lösungen zum Gewähren und Widerrufen des Zugriffs werden immer häufiger eingesetzt, da Unternehmen die Netzwerksicherheit verbessern und das Risiko von Datenverlust oder -diebstahl verringern möchten.

Strategie

Trotz der potenziellen Nachteile bietet die Verwendung föderierter Identitäten erhebliche Vorteile für Organisationen. Die Vereinheitlichung verschiedener Anwendungen zur Beseitigung von Engpässen und Silos sorgt für ein reibungsloseres Benutzererlebnis und ermöglicht den Mitarbeitern ein effizientes Arbeiten.

Organisationen, die auf Anwendungen angewiesen sind, mit denen föderierte Identitäten nicht verwendet werden können, sollten überlegen, ob die gleiche Funktionalität mit neueren Anwendungen erreicht werden kann oder ob die vorhandene Anwendung für die Integration in ein föderiertes System aktualisiert werden kann. Kritische Programme, denen die Funktionalität für die Föderation fehlt, erfordern zusätzliche Überlegungen, um die Sicherheit zu gewährleisten.

Mit der zunehmenden Verbreitung von Identitätsföderationen werden die daraus resultierenden Partnerschaften zwischen Anbietern und Unternehmen wahrscheinlich die Einführung strengerer Sicherheitsrichtlinien auf breiter Front vorantreiben.

Komponenten

Das föderierte Access Management umfasst ein Netzwerk miteinander verbundener Komponenten, die für die Gewährleistung eines sicheren und nahtlosen Zugriffs von Benutzern auf verschiedenen digitalen Plattformen von entscheidender Bedeutung sind.

Authentifizierung

Die Authentifizierung ist ein wichtiger Prozess, der die Identität von Benutzern prüft, die Zugriff auf Systeme oder Dienste wünschen. Dieser grundlegende Schritt dient als erste Barriere gegen unbefugtes Eindringen, indem er die Legitimität von Einzelpersonen bestätigt.

Es verwendet verschiedene Methoden wie Passwörter, biometrische Daten (wie Fingerabdrücke oder Gesichtserkennung) oder Token (wie Sicherheitsschlüssel oder Smartcards), um sicherzustellen, dass nur autorisierte Benutzer Zutritt erhalten, und schützt so die Integrität und Sicherheit des Systems.

Autorisierung

Die Autorisierung ist die entscheidende Phase nach der Benutzerauthentifizierung. Während bei der Authentifizierung die Identität eines Benutzers bestätigt wird, konzentriert sich die Autorisierung auf die Definition und Zuweisung spezifischer Zugriffsrechte innerhalb eines Systems. Sein Hauptziel besteht darin, geeignete Berechtigungen basierend auf verifizierten Identitäten, zugewiesenen Rollen oder bestimmten Attributen zu erteilen.

Auf diese Weise stellt die Autorisierung sicher, dass Einzelpersonen nur auf Ressourcen oder Funktionen innerhalb des Systems zugreifen können, die für ihre zugewiesenen Verantwortlichkeiten oder Bedürfnisse wesentlich und relevant sind.

Zugangskontrolle

Die Zugriffskontrolle ist für die Gewährleistung der Sicherheit sensibler Ressourcen eines Systems von entscheidender Bedeutung. Es funktioniert durch die Definition und Durchsetzung von Regeln, die Benutzerinteraktionen nach der Autorisierung regeln.

Durch die Festlegung umfassender Richtlinien bestimmt die Zugriffskontrolle die Zugriffsparameter und legt fest, wer wann auf bestimmte Ressourcen zugreifen kann und unter welchen Methoden oder Bedingungen der Zugriff zulässig ist.

Identitätsanbieter (Identity Provider, IdP’s)

IdP fungieren als Wächter der Benutzeridentitäten und sind für die Authentifizierung von Benutzern und die Bereitstellung wichtiger Identitätsinformationen an Dienstanbieter (SP) verantwortlich. Ihre Rolle ist von größter Bedeutung, wenn es darum geht, einen reibungslosen Benutzerzugriff auf verschiedene Dienste zu ermöglichen und die Belastung durch die Verwaltung zahlreicher Anmeldeinformationen zu vermeiden.

Das Verständnis der Feinheiten von IdPn ist entscheidend, um ihre Funktion bei der Rationalisierung der Benutzerauthentifizierung und der Vereinfachung des Zugriffs auf mehrere Plattformen oder Dienste zu schätzen.

Dienstleister (Service Provider, SP)

SP sind integrale Komponenten innerhalb digitaler Plattformen, Anwendungen oder Systeme, die für die Benutzerauthentifizierung und häufig auch die Autorisierung stark von IdPn abhängig sind.

Identity Management

In verteilten Systemen kommt es zu einer zusätzlichen Komplexität: identitätsbezogene Informationen liegen dezentral, redundant und heterogen vor. Dies hat sowohl technische als auch nicht technische Gründe. Beispielsweise stärkt die Replikation von Informationen die Fehlertoleranz eines Systems auf der technischen Ebene genauso wie die Autonomie einer Organisation auf der Geschäftsebene. Nicht zuletzt führt vor allem das fehlende Vertrauen in eine zentrale Instanz zu einer verteilten Datenhaltung. Der durch die Replikation von Daten gewonnene Mehrwert hat jedoch seinen Preis: Da identitätsbezogene Informationen sich ändern, ist es zur Vermeidung von Inkonsistenzen notwendig, Replikate miteinander zu synchronisieren.

Allgemein können Informationen zu einem bestimmten Zeitpunkt als „konsistent“ bezeichnet werden, wenn sie in diesem Moment widerspruchsfrei sind (vgl. [Brockhaus Enzy. 2006a]). In diesem Sinne bedeutet Konsistenz Widerspruchsfreiheit. In verteilten Systemen ist vor allem der Zeitpunkt von Interesse, zu welchem Änderungen an den verteilt vorliegenden Informationen vorgenommen werden, da diese Änderungen an alle Replikate verteilt werden müssen, um die Widerspruchsfreiheit aufrecht zu erhalten.

Leider kann diese „ideale Welt“ in verteilten Systemen nicht existieren, da es unmöglich ist, alle Replikate bei Änderungen „gleichzeitig“ anzupassen. Mit anderen Worten: ohne den Begriff Konsistenz zu relaxieren, wäre ein System bei jeder Änderung grundsätzlich inkonsistent. Aus diesem Grund wird in verteilten Systemen die Konsistenz in Form von Modellen der Konsistenz definiert. Demnach wird die Konsistenz auf der Basis eines Regelwerks definiert und Informationen sind dann konsistent, wenn sich die beteiligten Systeme an diese Regeln halten.

Replikation von Identitätsdaten

Eine verteilte und redundante Speicherung von Informationen ist die Ursache, welche die Betrachtung von Fragen zur Konsistenz notwendig macht. Mit anderen Worten: Wenn es sich in verteilten IDM-Systemen vermeiden ließe, dass Informationen von Identitäten repliziert werden, müsste man sich über die Konsistenz dieser Informationen keine Gedanken machen. Aus diesem Grund soll im Folgenden eine detaillierte Betrachtung der Ursachen, die zur Replikation identitätsbezogener Informationen führen, erfolgen. Darüber hinaus sollen die Konsequenzen, die durch eine Replikation entstehen, näher beleuchtet werden.

Ursachen

Wesentliche Grundgedanken des föderativen Paradigmas, die auf den ersten Blick auch konträr erscheinen mögen, sind zum einen die Dezentralität und eine damit einhergehende redundante Haltung identitätsbezogener Informationen. Dem gegenüber steht der Grundgedanke, die Verteilung identitätsbezogener Informationen zu reduzieren, durch die Auslagerung der Verwaltung von Identitätsinformationen an Identitätsprovider. Die Gewichtung dieser beiden Gedanken variiert je nach Szenario. Im Folgenden sollen die unterschiedlichen Gründe, die eine Verteilung identitätsbezogener Informationen motivieren, im Detail dargelegt werden. Natürlich ist die Gewichtung dieser Motivatoren sehr Szenario-spezifisch, da ein durch die Verteilung von Identitätsinformationen gewonnener Mehrwert immer die „Kosten“, welche durch die

Verteilung entstehen, entgegengestellt werden müssen. Letztendlich gilt es, je nach Szenario abzuwägen, ob die Vorteile der Verteilung die entstehenden Kosten rechtfertigen.

Vertrauen

Eine wesentliche Ursache für die Replikation identitätsbezogener Informationen ist ein teilweise unzureichendes Vertrauen in Drittanbieter. Da es im Identitätsmanagement vorwiegend um datenschutzrelevante und sensitive Informationen geht, ist das notwendige Vertrauen in einen Drittanbieter hoch. Hierbei ist vor allem auch die Schwierigkeit, den potentiellen Schaden bzw. das Wohlwollen des Drittanbieters richtig einzuschätzen.

Im Wesentlichen bezieht sich Vertrauen demnach auf die Kompetenz und Zuverlässigkeit eines Partners. Vertrauen spielt im Rahmen dieses Dokuments zwar an vielen Stellen eine notwendige Voraussetzung, für das Verständnis der Zusammenhänge spielt es jedoch eine untergeordnete Rolle und soll aus diesem Grund auch nicht näher betrachtet werden.

Autonomie

Die Auslagerung essentieller identitätsspezifischer Aufgaben wie die Authentifikation und das Management von Benutzern erfordert ein hohes Maß an Vertrauen. Gleichzeitig entsteht hierdurch auch ein große Abhängigkeit zwischen einem Identitätsprovider und dessen Relying Parties. Dieser Verlust an Autonomie veranlasst Dienstanbieter oftmals dazu Identitätsinformationen lokal vorzuhalten, da diese redundante Speicherung identitätsbezogener Informationen erlaubt, auch im Falle eines Ausfalls des Identitätsproviders Dienste weiterhin erbringen zu können.

Der Gedanke, die Kontrolle über das Identitätsmanagement zu behalten, nicht zuletzt auch durch unterschiedliche rechtliche und technische Anforderungen der Unternehmen begründet, führt auch wesentlich dazu, dass aktuell im Internet angebotene Dienste größtenteils immer noch isoliert betrieben werden.

Langlaufende Dienste

Neben nichttechnischen Ursachen, wie das mangelnde Vertrauen und der Erhalt der Autonomie, spielen in aktuellen verteilten Systemen immer noch technische Ursachen ein gewichtige Rolle. Eine Herausforderung stellen langlaufende Dienste dar. In langlaufenden Diensten wird ein Dienst nicht nur kurzzeitig erbracht, d.h. während der Benutzer eine aktive Session beim Dienstanbieter etabliert hat, sondern auch nachdem der Benutzer sich abgemeldet hat. Beispiele langlaufender Dienste sind Newsletter-Anwendungen, Zeitschriftenabonnements oder auch das Ausleihen eines Buches in einer Bibliothek.

Nachfolgende Abbildung zeigt den konzeptionellen Aufbau eines langlaufenden Dienstes.

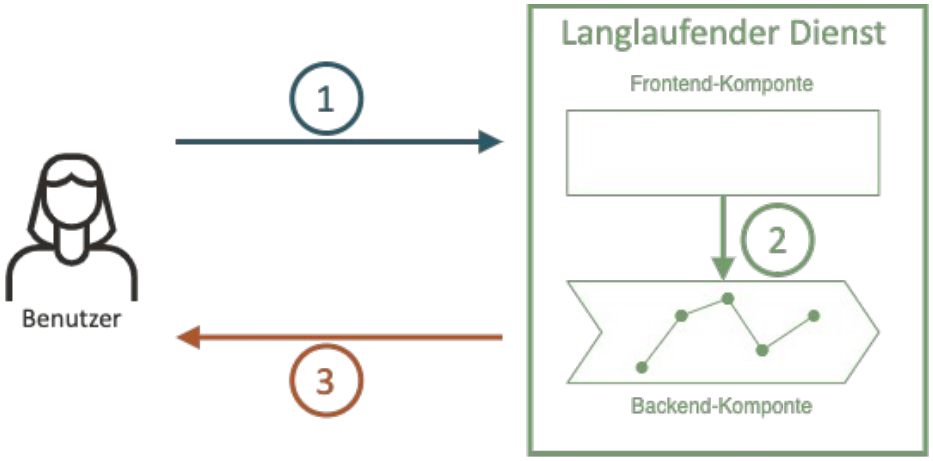

Abbildung 15: Konzeptioneller Aufbau eines langlaufenden Dienstes

Der Ablauf für die Nutzung eines langlaufenden Dienstes unterscheidet sich in Schritt (1) zunächst nicht von einem kurzlaufenden Dienst. Der Benutzer ruft über das Frontend des Dienstanbieters einen Dienst auf, welcher im Backend (2) ausgeführt wird. In Schritt (3) bekommt der Benutzer in irgendeiner Form eine Bestätigung über die Dienstausführung. Sobald sich nun der Benutzer abmeldet oder die Session nach einer Zeitüberschreitung abgelaufen ist, endet in einem kurzlaufenden Dienst die Diensterbringung und ein möglicherweise im Backend ausgeführter Dienst endet in einem kurzlaufenden Dienst mit diesem Zeitpunkt. Im Falle eines langlaufenden Dienstes wird der Dienst jedoch über diesen Zeitpunkt hinweg ausgeführt.

Legacy-Systeme

Eine weitere Ursache, welche die Replikation identitätsbezogener Informationen oftmals unumgänglich macht, ist die Verwendung von Legacy-Systemen.

Performance

Da zum einen zwar die Bandbreite heutiger Netzanschlüsse immer größer wird, dieser Bandbreitenzuwachs jedoch auch einer immer komplexeren und datenintensiveren Dienstelandschaft gegenübersteht, spielt auch heute die Performance identitätsbezogener Dienste immer noch eine Rolle. Des Weiteren ist auch die Skalierbarkeit eines Dienstes zu berücksichtigen, da ein Dienst auch unter Last eine angemessene Performance aufweisen sollte. Die Performance lässt sich dadurch verbessern, dass eine lokale Kopie identitätsbezogener Informationen in der Nähe der Systeme, die diese Daten benötigen, gespeichert werden, da hierdurch die Daten nicht von einer ausgelagerten Komponente geholt werden müssen.

Verfügbarkeit

Neben einer verbesserten Performance lässt sich durch die lokale Haltung identitätsbezogener Informationen auch eine starke zeitliche Abhängigkeit zu einer anderen Komponente vermeiden. Da lokal vorliegende Informationen jederzeit zugreifbar sind, wird die Verfügbarkeit eines Systems durch die Replikation identitätsbezogener Informationen verbessert.

Zusammenfassung

In diesem Abschnitt wurden die notwendigen Grundlagen zum Verständnis verteilter Identity Management Systeme dargelegt. Insbesondere wurde hierbei auf den Austausch identitätsbezogenerInformationen durch aktuelle Standards, -Protokolle und -Softwaresysteme

fokussiert. Des Weiteren wurden die Ursachen und Konsequenzen präsentiert, welche zu einer redundanten und dezentralen Speicherung identätsbezogener Informationen führen. Als Fazit kann festgehalten werden, dass in aktuellen verteilten IdM-Systemen die Replikation von Identitätsinformationen in den meisten Szenarien erwünscht ist. Dies resultiert auch daraus, dass der durch die Replikation gewonnene Mehrwert höher ist als die dadurch entstehenden „Kosten“, was allgemein eine Grundvoraussetzung für die verteilte Haltung von Informationen sein sollte. Im Bereich des organisationsinternen Identitätsmanagements existieren bereits Ansätze zur Sicherstellung der Konsistenz.

Anforderungen zur Sicherstellung der Konsistenz

Elementare Anforderungen an einen Ansatz zur Sicherstellung der Konsistenz in verteilten IDM-Systemen werden erheblich von den Konsequenzen der Replikation (siehe Abschnitt Ursachen) identitätsbezogener Informationen beeinflusst. Die Konsequenzen haben in Bezug auf die Sicherstellung der Konsistenz identitätsbezogener Informationen die folgenden wesentlichen Auswirkungen:

Grundlegende Strategie zur Replikation

Ein wesentlicher Aspekt der Frage zu Sicherstellung der Konsistenz ist die jeweilige Strategie zur Replikation, d.h. wie Daten repliziert und vor allem wann auftretende Änderungen an die einzelnen Replikate propagiert werden. Im Folgenden sollen die beiden wesentlichen Techniken zur Replikation – die synchrone und asynchrone Replikation – vorgestellt werden. Des Weiteren werden zwei grundsätzliche Ansätze vorgestellt, welche zur Verteilung auftretender Änderungen eingesetzt werden, der Push-Ansatz und der Pull-Ansatz.

Synchrone Replikation

Bei der Verfolgung eines datenzentrierten Modells der Konsistenz wird ein transaktionsbasierter oder synchroner Ansatz zur Propagierung auftretender Änderungen verfolgt. Die Grundidee des synchronen Ansatzes ist, dass bei einer Änderung an einem Replikat der Zugriff auf alle weiteren Replikate so lange gesperrt wird, bis alle Replikate ebenfalls geändert werden konnten. Diese Technik wird aus diesem Grund auch P*essimistic Replication* oder E*ager Replication* genannt.

Es existiert eine Vielzahl an Systemen, die Konsistenz durch synchrone Mechanismen umsetzen. Typischerweise sind synchrone Mechanismen dann von Vorteil, wenn lokale Netzwerke mit einer geringen Latenz und einer niedrigen Fehlerquote eingesetzt werden. Durch die synchrone Strategie zur Replikation kann ein System im Normalfall keinen inkonsistenten Datensatz lesen, da jegliche Operation transaktionsbasiert durchgeführt wird, d.h. eine Operation ist erst dann als erfolgreich gekennzeichnet, wenn alle Replikate aktualisiert werden konnten.

Der Nachteil der synchronen Strategie zur Replikation besteht darin, dass alle zu aktualisierende Replikate solange sowohl für einen lesenden als auch für einen schreibenden Zugriff gesperrt sind, bis die Transaktion abgeschlossen ist, d.h. entweder bis die Aktualisierung vollständig durchgeführt werden konnte, oder im Fehlerfall alle Änderungen wieder rückgängig gemacht wurden. Diese Technik der Replikation hat vor allem in verteilten Systemen einen erheblichen Einfluss auf die Performance und Verfügbarkeit der Einzelsysteme.

Asynchrone Replikation

Verschiedene Faktoren, wie eine eingeschränkte Konnektivität oder ein hoher Verteilungsgrad motivieren den Einsatz einer asynchronen Replikationsstrategie auch *Optimistic Replication* oder *Lazy Replication* genannt. Auch der Faktor Mensch trägt in verteilten Systemen häufig dazu bei, dass optimistische Verfahren eingesetzt werden müssen, vor allem dann, wenn mehrere Beteiligte kollaborativ an einer Sache arbeiten. Die Asynchronität erlaubt hierbei, dass Operationen auf lokal vorliegenden Daten ohne die Koordination mit anderen Replikaten durchgeführt werden können. Diese Strategie wird deshalb optimistisch genannt, da von der Annahme ausgegangen wird, dass eine asynchrone Ausführung in den meisten Fällen keine Probleme verursacht und falls doch Probleme aufkommen, können diese im Nachhinein ohne großen Aufwand behoben werden.

Ein wesentlicher Nachteil einer asynchronen Replikationsstrategie ist die Gefahr potentieller Inkonsistenzen. Inwieweit das Risiko von Inkonsistenzen gegeben ist und vor allem inwieweit Inkonsistenzen toleriert werden können, hängt stark von dem jeweiligen Anwendungsszenario ab.

Durch eine asynchrone Replikationsstrategie lässt sich die Verfügbarkeit und die Performance des Gesamtsystems erhöhen, da ein Schreibzugriff auch dann abgeschlossen werden kann, wenn Teilsysteme nicht erreichbar sind oder die Teilsysteme eine schlechte Performance aufweisen.

Push- und Pull-Ansatz

In asynchronen Replikationsstrategien ist die grundsätzliche Konzeptionsentscheidung, ob Aktualisierungen auf der Basis eines Push-Ansatzes oder eines Pull-Ansatzes realisiert werden.

Bei einem Push-basierten Ansatz wird die Aktualisierung der Replikate durch den Datenspeicher initiiert, auf welchem das Replikat geändert wurde. Der Vorteil eines Push-basierten Ansatzes ist, dass Änderungen sehr schnell und zeitnah propagiert werden können. In einer synchronen Replikationsstrategie ist ein Push-basierter Ansatz eine Grundvoraussetzung.

Die Herausforderung eines Push-basierten Ansatzes besteht darin, dass eine koordinierende Instanz benötigt wird, welche alle Replikate verwaltet, die angepasst werden müssen. Des Weiteren ist es notwendig, dass bei einer Nichterreichbarkeit eines Replikats mit etwaigen Sendewiederholungen gearbeitet werden muss, um sicherstellen zu können, dass Änderungen auch an alle Replikate verteilt werden. Unter Umständen verbieten Ansätze wie traditionelle Client/Server-Protokolle einen Push-basierten Ansatz aus genau diesem Grund, da der Server die Adresse der Clients nicht kennt und somit Aktualisierungen auch nicht senden kann.

Bei einem Pull-basierten Ansatz, wird eine Aktualisierung erst dann an einem Replikat vorgenommen, wenn der Datenspeicher dieses Replikats die Aktualisierung bei einem Datenspeicher eines geänderten Replikats anfragt. Typischerweise erfolgt diese Abfrage periodisch, was in der Informatik als *Polling* bezeichnet wird. Pull-basierte Ansätze gehen somit das Risiko ein, dass auftretende Änderungen nicht unmittelbar an alle Replikate propagiert werden, wodurch die Gefahr potentieller Inkonsistenzen erhöht wird. Ein Pull-Ansatz kann somit nur in Kombination mit einer asynchronen Replikationsstrategie eingesetzt werden.

Die Schwierigkeit bei einem Pull-basierten Ansatz ist die Bestimmung des Polling-Intervalls. Falls dieser zu klein gewählt wird, ist die Gefahr „unnötiger“ Anfragen gegeben, d.h. obwohl sich an den angefragten Daten nichts geändert hat, wird eine Anfrage gestellt und somit u.a. die Netzwerklast erhöht. Falls das Intervall zu gering gewählt wird, besteht die Gefahr, Änderungen zu spät zu erhalten und somit längere Zeit einen veralteten Wert vorzuhalten.

P20 Identity & Access Management

Gemäß der Begriffsdefinition Föderation und Föderatives Identitätsmanagement nimmt das Bundeskriminalamt die Rolle des Dienstleisters und die Teilnehmer die Rolle des Dienstnehmers ein.

Es gibt potentiell eine Vielzahl von Anwendungen, die an das föderierte P20 Identity & Access Management (F-IAM) des Basisdienstes IAM angebunden werden müssen..

Es gibt unterschiedliche Aspekte bei der Anbindung einer Anwendung mit jeweils mehreren Optionen, so dass keine einheitliche Vorgabe über alle Anwendungen gemacht werden kann. Dieses Dokument soll daher eine allgemeine Richtlinie darstellen, wie in Abhängigkeit verschiedener Kriterien der konkreten Anwendung die verschiedenen Optionen der Anbindung auszuwählen sind.

Anforderungen

Neben den gängigen Anforderungen an IAM-Systeme ergeben sich eine Reihe spezieller Anforderungen aus dem P20 Umfeld. Es wurde mit dem Projekt IAM im Jahr 2020 damit begonnen, diese Anforderungen aufzunehmen. Seitdem wurden sie immer wieder aus Gesprächen mit TN-IAM- und Anwendungsteams heraus aktualisiert. Besonders hervorzuhebende Anforderungen sind:

- Einhaltung von bestehenden Vorgaben zu IT-Sicherheit und Datenschutz im polizeilichen Umfeld
- nicht-funktionale Anforderungen (NFA) von P20.
- Bestandsschutz: Gewährleistung von Kompatibilität mit bestehenden Lösungen bei den TN – das Umfeld ist und bleibt dadurch besonders heterogen.
- Gewährleistung der Souveränität der TN: Jeder TN behält auch im Verbund die Hoheit über seine Daten.
- TN, die selbst noch über kein ausgereiftes IAM-System verfügen, möchten i. d. R. gerne ein breites Self Service Angebot des zentralen IAM nutzen können – TN mit ausgereiftem IAM-System hingegen möchten sämtliche Aktionen in ihrem eigenen System vornehmen können.
- Durch die heterogenen Systeme, die als IDP und SP angebunden werden sollen, gibt es besondere Anforderungen an das F-IAM als Übersetzer zwischen Protokollen, um gemeinsame Prozesse zu ermöglichen.
- Manche Teilnehmer benötigen die Möglichkeit, Berechtigungen anhand der Dienststelle des Benutzers automatisiert zu vergeben.

Sicherheitskonzepte

Die Dienste die durch die Dienstleitungen des Bundeskriminalamts auf seiner Private Cloud, der Polizei-Service-Plattform (PSP) bereitgestellt werden unterliegen erweiterten Sicherheitsrichtlinien, sogenannten Schutzzonen. Diese Schutzzonen weisen dabei jeweils spezifische Eigenschaften hinsichtlich Konnektivität und Schutzniveau auf.

Die Schutzzone 1 ist dabei die eigene interne Umgebung der jeweiligen Behörde. Die Schutzzone 2 ist die geteilte Umgebung der Polizeien. Die Schutzzone 3 ist die Umgebung zur Auswertung von Schmutzdaten. Die Schutzzone 4 ist die internet nahe Umgebung.

Der Basisdienst IAM agiert hauptsächlich in der Schutzzone 2 und bietet allerdings seine Dienste gemäß Konzeption für alle Anwendungen in den jeweiligen Schutzzonen getrennt nach Access Management und Identity Management an.

Auf Grund dieser gemäß den IT-Sicherheitsrichtlinien folgenden Kommunikationswege sind in den Schutzzonen mit erhöhtem bzw. hohen Sicherheitsbedarf eigene Mechanismen zur Zugriffssteuerung (Access Management) als auch die Bereitstellung von Benutzerkonten (Identity Management) notwendig. Diese Trennung ermöglicht die Beschränkung und Isolation der Konnektivität in der jeweiligen Schutzzone.

Daraus ergeben sich aber zwangsläufig Anforderungen an die Architektur des föderierten P20 Identity & Access Management.

Prozesse

Dieses Kapitel befasst sich mit essentiellen IAM-Prozessen und erläutert fachlich dargestellt, welche Anwendungsfälle durch die Benutzerverwaltung des TN abgedeckt werden müssen bzw. welche Methoden seitens des Basisdienstes IAM den TN-BVs/TN-IAMs zur Verfügung stehen.

Als essentielle IAM-Prozesse werden die folgenden vier Prozesse definiert:

Support-Organisation

Benutzerverwaltung

Rollen- & Berechtigungsverwaltung

Authentifizierung

Die vom Basisdienst IAM angebotenen Methoden können von den TN individuell in eigenen IAM-Prozessen verwendet werden. Bei der Planung und Umsetzung von IAM-Prozessen in Systemen mit P20-Bezug, sind die Hinweise des BSI Grundschutzbausteins ORP.4 Identitäts- und Berechtigungsmanagement1 zu beachten.

Support-Organisation

Für Benutzer, die Zugriff auf eine P20-Anwendung benötigen, wird innerhalb des TN eine entsprechende Support-Organisation benötigt, die P20-Anwendern als Ansprechpartner zur Verfügung steht. Wie das Datenhaus-Ökosystem bei TN-eigenen Betriebsprozessen berücksichtigt wird, steht den TN frei (z. B. Integration in den HelpDesk, Self-Service-Shops, etc.). Dasselbe Prinzip gilt für Betriebsprozesse zur Entstörung (z.B. Incident-Management, Problem-Management).

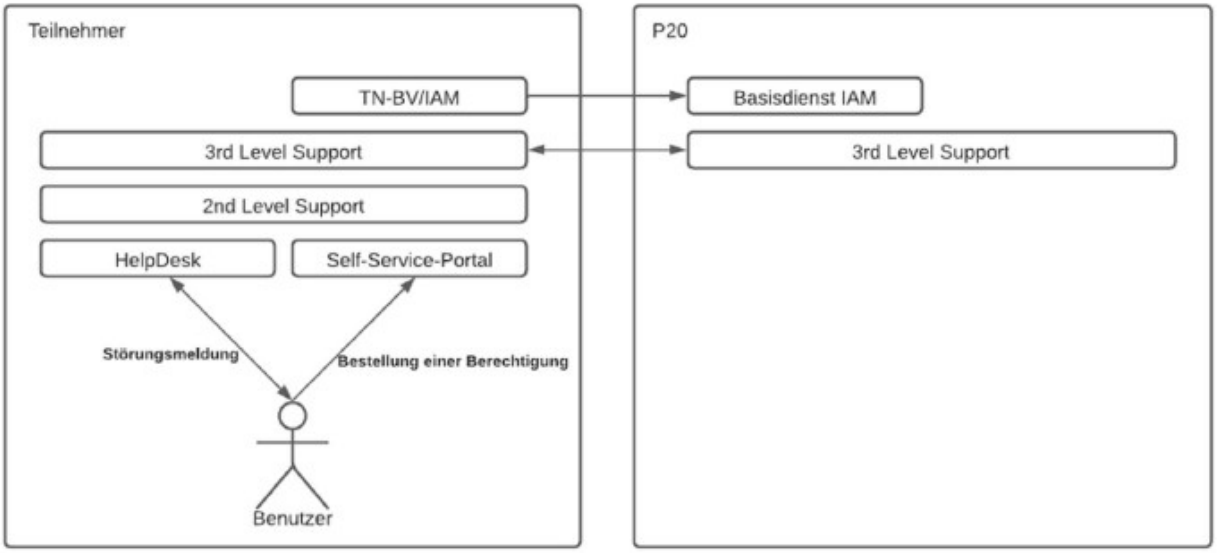

Abbildung 16: Mögliche Support-Organisation mit Schnittstelle zu zentralem 3rd-Level-Support für Basisdienst IAM

Die Abbildung zeigt beispielhaft wie eine Integration von P20, hier speziell für IAM-Anfragen, in die eigene Support-Organisation aussehen könnte. Aktuell wird durch das Programm kein zentraler Endanwender-Support bereitgestellt. Da fast sämtliche IAM-Prozesse TN-spezifisch sind, ist eine Vorqualifizierung von Anfragen und Störungen durch die Support-Organisation der Teilnehmer in jedem Fall sinnvoll.

Benutzerverwaltung

Um Benutzern Zugriff auf eine P20-Anwendung bzw. einen P20-Dienst geben zu können ist es erforderlich, zunächst ein Benutzerkonto im Basisdienst IAM anzulegen. Die manuelle Anlage neuer Benutzerkonten über eine Web-Oberfläche wird auch weiterhin möglich sein. Dieses Vorgehen eignet sich jedoch nur für geringe Benutzerzahlen und steht im Widerspruch zum Grundsatz, dass Teilnehmer die Benutzerverwaltung weiterhin in eigenen Systemen durchführen wollen. Dieses Dokument befasst sich daher vor allem mit der automatisierten Nutzung der unterstützten Schnittstellen.

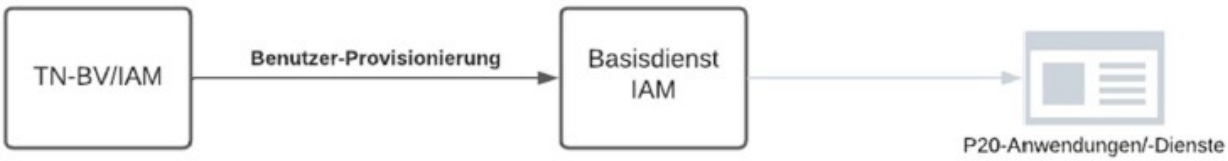

Abbildung 17: Fachliche Darstellung der Benutzer-Provisionierung in einem entfernten System

Das Quellsystem bzw. das führende System für Benutzer- und Berechtigungsinformationen ist die Benutzerverwaltung der Teilnehmer. Alle Benutzerkonten innerhalb des Basisdienstes IAM sind sog. „föderierte Benutzerkonten“ denen ein Pendant innerhalb der TN-BV gegenübersteht.

Das sog. „Identity-Lifecycle-Management“ findet vollständig innerhalb der TN-BV statt. Es sind die IAM-Prozesse der TN, die folgende Aspekte steuern:

- wann ein Benutzerkonto erstellt wird
- wie und unter welchen Umständen (z.B. Freigabeprozesse) das Konto erstellt wird
- ob und unter welchen Umständen ein Konto temporär deaktiviert wird (z.B. Elternzeit, Austritt)
- ob und unter welchen Umständen ein Konto reaktiviert wird (z.B. Wiedereintritt)
- wann und unter welchen Umständen ein Konto dauerhaft deaktiviert wird (z.B. Pension, Tod)
- den Abgleich zwischen TN-BV/IAM und Basisdienst IAM zur Sicherstellung von Synchronität der Benutzerdaten

Aus der Sicht der TN-BV handelt es sich beim Basisdienst IAM also um ein Zielsystem, welches die Berechtigung von Benutzern auf einen P20-Dienst ermöglicht.

Aus der Sicht von P20-Anwendungen ist der Basisdienst IAM das führende System für Benutzer- und Berechtigungsinformationen. Der Basisdienst IAM stellt außerdem sicher, dass basierend auf Berechtigungszuweisungen der TN über die von Ihnen verwendete IDM-Schnittstelle, die entsprechenden Anwendungen innerhalb des Datenhaus-Ökosystems die von Ihnen benötigten Benutzer- und Berechtigungsinformationen erhalten, ohne dass eine direkte IAM-Integration zwischen der TN-BV und der P20-Anwendung durchgeführt wurde.

Rollen- & Berechtigungsverwaltung

Im Bereich der Rollen- & Berechtigungsverwaltung ergibt sich durch den föderativen Ansatz des Basisdienstes IAM die Situation, dass nicht nur die Zuweisung/Änderung/Entfernung von Berechtigungen an einem Benutzerkonto durch die TN bzw. die TN-BV durchgeführt wird. Auch die Definition von Berechtigungen, unter Berücksichtigung und Verwendung der Berechtigungskonzepte der tatsächlichen Zielanwendung, obliegt den Benutzerverwaltern der Teilnehmer bzw. wird durch die TN-BV abgedeckt.

Zuweisung/Änderung/Entfernung von Berechtigungen

Das Setzen bzw. Entfernen von Berechtigungen wird, analog zur Benutzerverwaltung, durch die TN-BV gesteuert. Hierzu werden, je nach verwendeter Schnittstelle, entsprechende Methoden seitens des Basisdienstes IAM angeboten, die durch die TN-BV verwendet werden können.

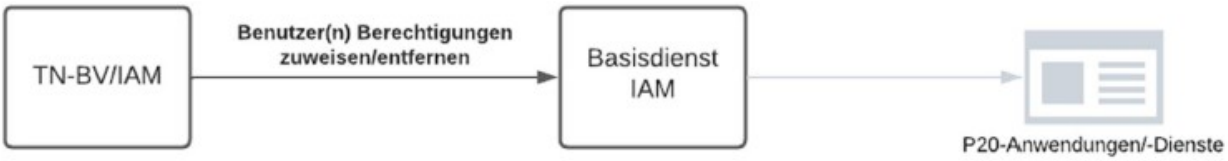

Abbildung 18: Fachliche Darstellung der Berechtigungsverwaltung in einem entfernten System

Das sog. „Identity-Lifecycle-Management“ findet vollständig innerhalb der TN-BV statt. Es sind die IAM-Prozesse der TN, die folgende Aspekte steuern:

- wie und unter welchen Umständen (z.B. Freigabeprozesse, autom. Berechtigungszuweisungen) das Konto Berechtigungen erhalten wird
- ob und wie diese Berechtigungen regelmäßig geprüft werden („Re-Zertifizierung“)
- unter welchen Umständen Berechtigungen entzogen werden (z.B. Austritt, Wechsel des Aufgabenbereichs, etc.)
- den Abgleich zwischen TN-BV/IAM und Basisdienst IAM zur Sicherstellung von Synchronität der Berechtigungsinformationen

Authentifizierung

Unabhängig vom Anwendungsfall und verwendetem Protokoll (SAML2/OIDC) handelt es sich bei den Authentifizierungsprozessen des Basisdienstes IAM immer um eine delegierte Authentifizierung. P20-Anwendungen delegieren die Authentifizierung eines zugreifenden Benutzers an den Basisdienst IAM. Der Basisdienst IAM wiederum delegiert diese Authentifizierung an den Teilnehmer des zugreifenden Benutzers. Der Teilnehmer authentifiziert den Benutzer basierend auf den bereits vorhandenen Prozessen und Werkzeugen (z.B. SSO via Kerberos, 2-Faktor-Authentifizierung mit Token-Generator oder SmartCard, etc.), föderiert diese Authentifizierung mit einem Access-Management-Werkzeug (z.B. ADFS, Forgerock AM, PingFederate/PingAccess, KeyCloak, uvm.) damit diese vom Basisdienst IAM weiterverwendet werden kann,um ein nahtloses Single-Sign-On bis in die Anwendung zu ermöglichen.

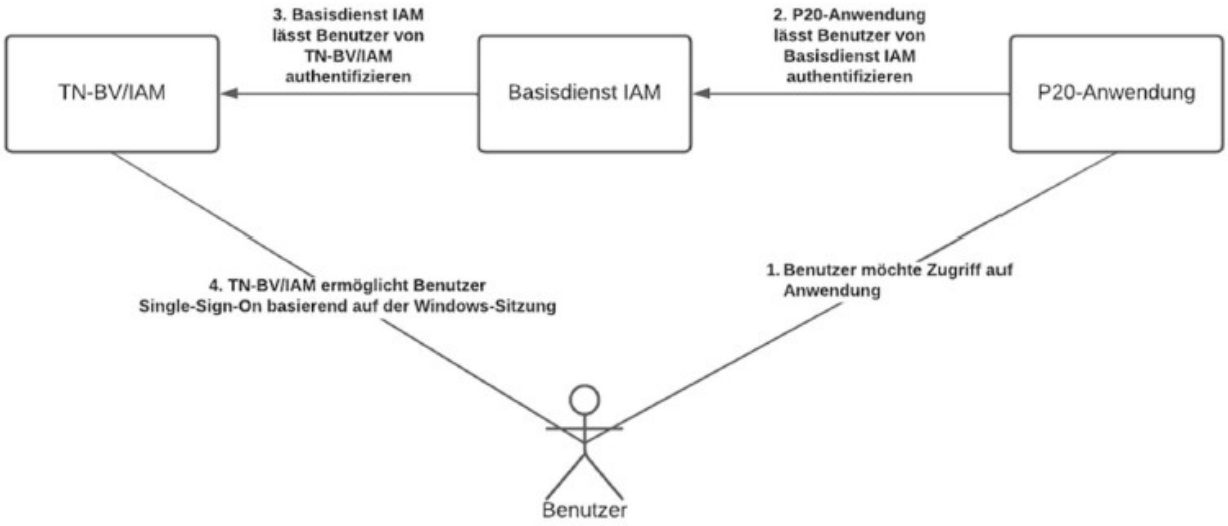

Abbildung 19: Fachliche Darstellung eines "Authentication Flows" im Kontext delegierter Authentifizierung

Der dargestellte Authentifizierungs-Ablauf ist eine fachliche Darstellung zur Verdeutlichung der generellen Funktionsweise. Technisch präzise Sequenzdiagramme, die unterschiedliche sog.

„Authentication-Flows“ unter Verwendung unterschiedlicher Protokolle (SAML2/OIDC) sowie im Kontext unterschiedlicher Anwendungsfälle zeigen, finden Sie in Anhang ???? bis ????.

Teilnehmer/IdP

Als Teilnehmer agieren im P20 Identity & Access Management sowohl die Polizeien des Bundes und der Länder, als auch Dritte die diesem Verbund nicht eindeutig zuordenbar sind allerdings Dienste, die auf der PSP bereitgestellt sind nutzen. Dazu gehören z.B. die Organisationen, die innerhalb des Schengener Informationssystems auf das nationale Auskunftssystem zugreifen.

Auch daraus ergebene sich Anforderungen an die Architektur des F-IAM, da dieses auf einer gemeinsamen Plattform betrieben wird.

Bei Bedarf kann ein P20-TN oder eine andere Organisation auch in mehrere IdPs aufgespalten werden. Dies kann sinnvoll sein, wenn es den tatsächlichen Zuständigkeiten innerhalb der Organisation entspricht, z. B. unterschiedliche Dienststellen unterschiedliche TN-IAM-Systeme nutzen. Diese Aufspaltung sollte aber nicht granularer als notwendig sein und ist mit dem F-IAM-Team abzustimmen, um Inkonsistenzen zu vermeiden.

Anwendungstypen

Bei den über das P20 Identity & Access Management abgesicherten Anwendungen werden folgende Typen unterschieden.

- **Webanwendung**

  Eine Anwendung, die ein Benutzer über einen Webbrowser aufruft. Hierbei kann es sich auch um eine Single-Page-Anwendung (SPA) handeln.

- **Rich-Client mit Backend**

  Eine auf einem Endgerät installierte Anwendung (z.B. Desktop-Anwendung oder Mobile-App) mit oder ohne dediziertem Webservice.

- **Terminal Server Umgebung**

  Prinzipiell kann es sich hierbei um jeden der vorgenannten Anwendungstypen handeln, der gemäß den im Abschnitt Sicherheitskonzepte beschriebenen Schutzzonen 3 bzw. 4 bereitgestellt ist.

  Die finale Authentisierung eines Anwenders in der Terminal Server Umgebung findet allerdings nur innerhalb der jeweiligen Schutzzone statt. Der Basisdienst IAM übernimmt in dieser Konstellation lediglich und ausschließlich die Authentisierung bis zum äußeren Perimeter der Terminal Server Umgebung. Eine weiter Interaktion mit dem Basisdienst IAM zum Bezug erweiterter Berechtigungsinformationen zum angemeldeten Benutzer ist ausgeschlossen.

  In den nachfolgenden Ausführungen ist in diesem Kontext als Anwendung immer die Terminal Server Umgebung zu verstehen.

- **Webservice**

  Eine von anderen Anwendungen über eine explizite Schnittstelle aufrufbare

Funktion, entweder Bestandteil einer Anwendung oder als anwendungsunabhängiger Dienst.

Access Management

Unabhängig vom Anwendungsfall und verwendetem Protokoll handelt es sich bei den Authentifizierungsprozessen des Basisdienstes IAM immer um eine delegierte Authentifizierung. Anwendungen delegieren die Authentifizierung eines zugreifenden Benutzers an den Basisdienst IAM. Der Basisdienst IAM wiederum delegiert diese Authentifizierung an den Teilnehmer des zugreifenden Benutzers. Der Teilnehmer authentifiziert den Benutzer basierend auf den bereits vorhandenen Prozessen und Werkzeugen, föderiert diese Authentifizierung mit seinem AM-Werkzeug damit diese vom Basisdienst IAM weiterverwendet werden kann, um ein nahtloses Single-Sign-On bis in die Anwendung zu ermöglichen.

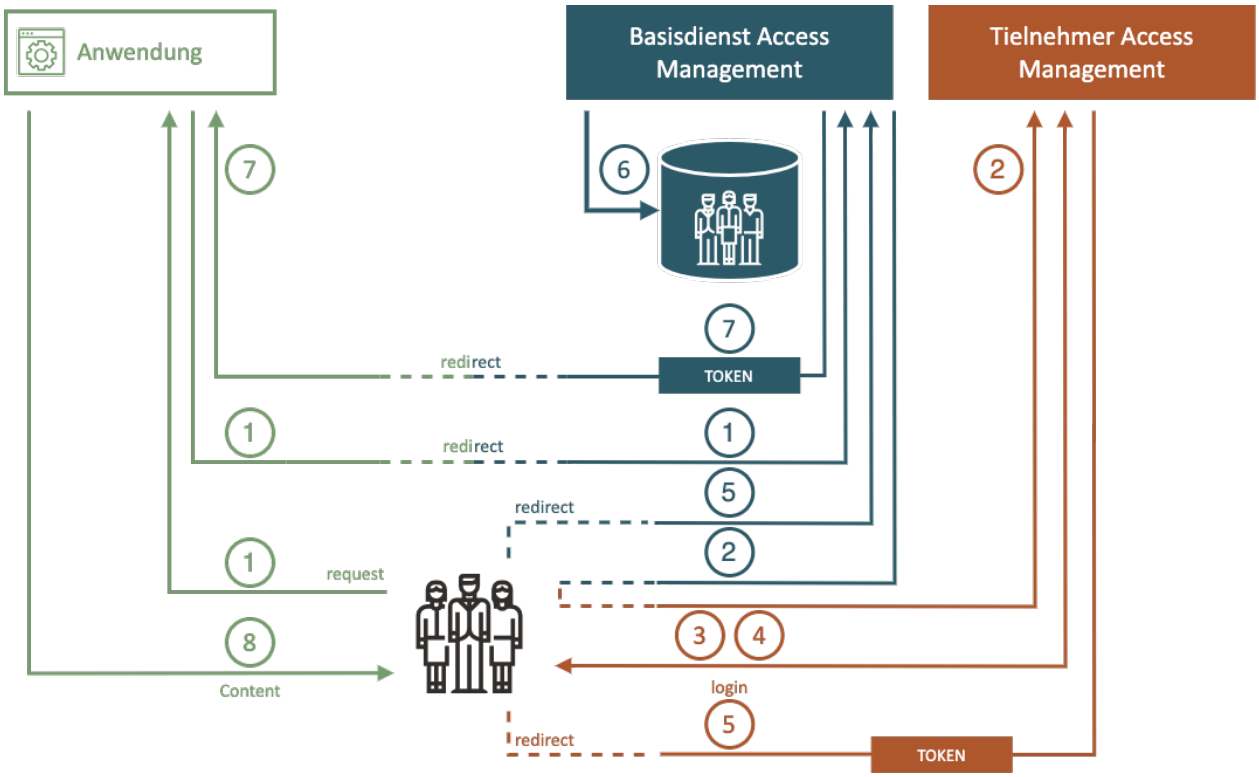

Abbildung 20: Grundprinzip Föderiertes Access Management

1. Das föderierte AM-System stellt die Schnittstelle zwischen dem Benutzer und der Anwendung dar. Dies kann durch verschiedene Mechanismen erfolgen. Die häufigste Methode besteht darin, dass die Anwendungen selbst den Benutzer zum AM-System umleiten, wenn keine Sitzung vorhanden ist.
2. Das föderierte AM-System prüft das Vorhandensein einer Sitzung und leitet den Benutzer zum teilnehmerspezifischen AM-System weiter, falls keine Sitzung vorhanden ist.
3. Der Benutzer gibt die Anmeldeinformationen gegenüber dem teilnehmerspezifischen Access-Management-System ein.
4. Das teilnehmerspezifischen AM-System prüft die Gültigkeit der Anmeldeinformationen und bewertet, falls vorhanden, die Zugriffsrichtlinien.

5. Wenn der Zugriff erlaubt ist, leitet das teilnehmerspezifischen AM-System den Benutzer zurück zum föderierte AM-System. Die Umleitung enthält normalerweise ein Zugriffstoken: eine kleine Information, die dem föderierte AM-System mitteilt, dass der Benutzer authentifiziert ist.
6. Das föderierte AM-System validiert das Token, erstellt eine lokale Sitzung und und leitet den Benutzer zur Anwendung weiter. Die Umleitung enthält ein Zugriffstoken, das vom föderierte AM-System ausgestellt wurde.
7. Die Anwendung validiert das Token, erstellt eine lokale Sitzung
8. Die Anwendung gewährt den Zugriff.

Authentisierung von Benutzern

Die nachfolgenden Inhalte betreffen alle Anwendungen, unabhängig davon, ob die Authentifizierung per OIDC oder SAML2 erfolgt. Der Einsatz der Standards bezieht sich auf die Authentifizierung gegenüber der Anwendung.

Die Authentifizierung des Benutzers gegenüber dem TN-IAM liegt in der Hoheit des jeweiligen Teilnehmers. Beim Einsatz einer Anwendung mit Verwendung eines Rich-Clients in der Topologie eines Teilnehmers betrieben wird sollte eine Abstimmung zwischen TN und Hersteller erfolgen, damit sichergestellt ist, dass die Antwort des TN-IAMs an den Rich-Client von diesem auch korrekt verarbeitet werden kann. Beispiele hierfür sind:

- Authentifizierungsanfrage per SPNEGO
- HTML-Seite zur Eingabe von Benutzername und Passwort

Autorisierung von Benutzern

Konzeptionell ist es vorgesehen, dass die Zuweisung statischer Berechtigungen (siehe hierzu Abschnitt Identity Management) vornehmlich durch die Teilnehmer erfolgt und die Daten über den Basisdienst IAM durch geeignete Schnittstellen an die Anwendung übertragen werden. Eine zusätzliche Pflege innerhalb der Anwendung zum Zuweisen von Berechtigungen sollte also vermieden werden.

Ein möglicher Grund für eine Abweichung ist, dass der Basisdienst IAM den fachlich erforderlichen Prozess zur Berechtigungszuweisung noch nicht abbilden kann, wenn also beispielsweise die Anwendung dem Zuweisen bestimmter Berechtigungen durch den Teilnehmer explizit zustimmen muss. Dieses Vorgehen erfordert jedoch immer eine Einzelfallprüfung und explizite Zustimmung durch den Betrieb des Basisdienstes IAM und muss als Ausnahme dokumentiert werden.

Anwendungsspezifischer Benutzerspeicher

- Erfordert die Bereitstellung einer API durch die Anwendung bzw. deren Plattformbetreiber, für die die Zuweisung statischer Berechtigungen.
- Im Zielbild 2030+ sollen keine personenbezogenen Daten mehr im Basisdienst IAM persistiert werden, dürfen aber ggf. weiterhin übertragen werden.

Scope im Access-Token

- Nur zum Berechtigen des grundsätzlichen Zugriffs ohne weitere Detaillierung geeignet.
- Ermöglicht Zugriff auf Berechtigungsinformationen ausschließlich zum anfragenden Benutzer.

Claim im Access-Token

- Eigenes Claim „groups“ im Token mit einer Liste der Anwendungsfunktionsrechte, die dem Benutzer für die jeweilige Anwendung zugewiesen sind.
- Beschränkt auf eine Liste von bis zu 20 Anwendungsfunktionsrechte ohne Dienststellenbezug.
- Ermöglicht Zugriff auf Berechtigungsinformationen ausschließlich zum anfragenden Benutzer.

Explizite Abfrage mit einem P20-Access-Token

- Aufruf eines userprofile-Endpunktes mit Übergabe eines P20-Access-Tokens, analog zum userinfo-Endpoint des OIDC-Standards.
- Ermöglicht Zugriff auf Berechtigungsinformationen ausschließlich zum anfragenden Benutzer.
- Erfordert ein P20-Access-Token (JWT).
- Liefert nur Informationen zu der Anwendung, deren Scope im übergebenen P20-Access-Token enthaltenen ist.
- Geeignet für sehr umfangreiche Berechtigungen ohne Dienststellenbezug.
- Funktioniert unabhängig davon, ob das P20-Access-Token per OIDC-Flow oder Token Exchange erzeugt wurde.

Autorisierung per Token Exchange

Über Token-Exchange kann eine Anwendung das Token, das sie durch die Anmeldung eines Benutzers erhalten, gegen ein Token austauschen, das von einem Webservice den die Anwendung stellvertretend für den Benutzer aufruft, akzeptiert wird.

Die Umsetzung erfolgt gemäß RFC 8693 „OAuth 2.0 Token Exchange“.

OAuth 2.0 Token Exchange ist eine Erweiterung des OAuth 2.0-Frameworks, die es einer Anwendung ermöglichen soll, einen Typ von OAuth2-Access-Token gegen einen anderen auszutauschen. Dieser Mechanismus bietet vertrauenswürdigen Diensten oder Clients die Möglichkeit, Token auf sichere Weise abzurufen oder auszutauschen.

Herkömmliches OAuth ist in erster Linie für die Delegation von Zugriffsberechtigungen konzipiert und ermöglicht der Anwendung, im Namen eines Benutzers auf Ressourcen zuzugreifen, indem sie ein Zugriffstoken erhält. Es wird häufig zur Benutzerauthentifizierung und -autorisierung in verschiedenen Szenarien verwendet, einschließlich der Anmeldung in sozialen Medien, der einmaligen Anmeldung und des Zugriffs auf Anwendungen von Drittanbietern. Token Exchange

wurde speziell für den Austausch eines Typs von Token gegen einen anderen entwickelt, typischerweise damit vertrauenswürdige Anwendungen einen anderen Typ von Token erwerben können, ohne dass eine erneute Zustimmung des Benutzers erforderlich ist.

Bei traditionellem OAuth wird der Benutzer in die Ausstellung des Tokens einbezogen. Der Benutzer erteilt der Anwendung normalerweise die Berechtigung (Consent), auf seine Daten auf einem Ressourcenserver zuzugreifen. Dies beinhaltet die Authentifizierung und Zustimmung des Benutzers. Token-Exchange wird in der Regel von vertrauenswürdigen Anwendungen oder Diensten durchgeführt, die über die erforderliche Autorisierung verfügen, und erfolgt häufig, ohne dass der Benutzer erneut seine Zustimmung erteilen muss.

> ***Wichtig***
>
> Token Exchange kann nur durch Anwendungen verwendet werden, die selbst über OAuth/OIDC authentisiert wurden.
>
> Eine Anwendung, die durch ein SAML-Token authentisiert wurde, besitzt diese Fähigkeit nicht.

Autorisierung ohne IDM-Anbindung

Grundsätzlich ist angeraten, dass alle Anwendungen, bei denen es ohne unrealistische Aufwände möglich ist, sowohl für IDM als auch AM an das F-IAM angebunden werden. Dennoch gibt es immer wieder Fälle, bei denen Anwendungsteams eine reine AM-Anwendung anstreben. Vor allem bei Kaufsoftware, die nicht geändert werden kann, kommen solche Fälle vor. Dies ist nur dann sinnvoll möglich, wenn nur Informationen zum aktuell angemeldeten Benutzer benötigt werden, nur wenige Berechtigungen und keine Berechtigungen mit Dienststellenbezug benötigt werden.

Solche Anwendungen dürfen keinen eigenen Benutzerspeicher besitzen, den sie mit Daten aus den Tokens füllen, da eine Deprovisionierung nicht möglich ist. Ausnahmen können hier nur gemacht werden, wenn den TN die Möglichkeit gegeben wird, Benutzer zu deaktivieren und zu löschen. Dies ist ggf. im Berechtigungskonzept der Anwendung zu dokumentieren.

Grundsätzlich ist es allerdings sinnvoll und für Adminkonten auch laut BSI Grundschutz vorgesehen, dass Anwendungen über einen eigenen Benutzerspeicher verfügen, um eine Fallback-Lösung zu haben, falls das F-IAM ausfallen sollte.

Es ist auch möglich, in solchen Fällen für die IDM-Anbindung einen Konnektor seitens F-IAM zu entwickeln. Die Aufwände dafür müssen allerdings ggf. vom jeweiligen Anwendungsteam kompensiert werden.

Privileged Account Management ist im F-IAM grundsätzlich nicht vorgesehen. Bisher wurden einige Ausnahmen gemacht (bspw. bei eFBS), die allerdings als technische Schulden angesehen werden.

Aus Sicht eines IAM können rein für AM angebundenen Anwendungen standardmäßig keine spezifischen Berechtigungen zugeschrieben werden. Es sind Berechtigungen im AM. Um nachvollziehen zu können, zu welcher Anwendung die Berechtigungen gehören, müssen diese Berechtigungen bisher dokumentiert werden. Hierbei handelt es sich aktuell um eine technische Schuld. Im Laufe von 2024 ist eine Anpassung des F-IAM vorgesehen, um direkt im F-IDM

abbilden zu können, zu welcher Anwendung die Berechtigungen gehören, auch ohne F-IDM-Anbindung der Anwendung.

Caching von Berechtigungen durch die Anwendung

Bei Anwendungen mit vielen Berechtigungen und ohne eigenen Benutzerspeicher können die Berechtigungen nicht über das Token an die Anwendung übermittelt werden, da die Länge des Tokens begrenzt ist. Als Workaround kann ein Caching eingerichtet werden. Bei dieser Variante ist die Gültigkeitsdauer von Tokens mit Bedacht festzulegen, da bewusst riskiert wird, mit veralteten Berechtigungen zu arbeiten. (Werden einem Benutzer Berechtigungen entzogen, greift dies in der Anwendung erst, wenn ein neues Token übermittelt wird.)

Variante 1: Caching pro Token

Die Antwort wird pro Token bis zum Ablauf seiner Gültigkeit gespeichert. Damit ist die Gültigkeit der Berechtigungsinformation identisch dazu als wären die Berechtigungen direkt im Token gespeichert.

Variante 2: Caching pro Benutzer

Die Antwort wird pro Benutzer bis zur Gültigkeit des genutzten Tokens gespeichert.

Greift derselbe Nutzer mit unterschiedlichen Tokens (z. B. über verschiedene Anwendungen) auf den Webservice zu, so würde innerhalb dieser Gültigkeitsdauer trotzdem nur einmal der Userprofile angefragt werden.

Anbindung von Teilnehmern

Dazu schaffen die Teilnehmer und der Basisdienst IAM Vertrauensstellungen zu einander, um sich auf die technischen Standards sowie auf gemeinsame organisatorische Spielregeln zu einigen. Der Endanwender, der mit entsprechenden Zugangsrechten ausgestattet ist, kann sich innerhalb dieser Vertrauensstellung bewegen, ohne sich jedes Mal neu anmelden und ausweisen zu müssen.

In der Praxis bedeutet das, dass ein Benutzer, der von einer teilnehmenden und als vertrauenswürdig geltenden Stelle identifiziert worden ist, auf Inhalte und Dienstleistungen zugreifen kann, ohne sich jedes Mal neu ausweisen zu müssen.

In diesem Kontext wird über eine reine Authentifizierung gesprochen. Weitere Fähigkeiten für eine Autorisierung sind in diesem Kontext nicht anwendbar.

Jeder Teilnehmer im P20-Umfeld stellt einen Identity Provider (IdP) bereit, der die teilnehmerspezifische Authentisierung eines Anwenders innerhalb der Föderation übernimmt. Der Basisdienst IAM übernimmt in diesem Szenario der Rolle des Service Providers (SP).

Die Optionen der Anbindung sind:

- SAML2
- OpenID Connect

Nachfolgende Tabelle gibt einen Überblick welche Anwendungstypen in der Kombination von TN-IAM und F-IAM authentisiert werden können.

Typ der Authentisierung		Typ der Anwendung		
TN-IAM	F-IAM	Webanwendung+	Rich-Client	Terminal Server
OIDC	OIDC	✓	✓	✗
OIDC	SAML	✓	✓	✓
SAML	OIDC	✓	✓	✗
SAML	SAML	✓	✓	✓

Tabelle 2: Überblick Authentisierung nach Anwendungstyp

Die eingesetzten Komponenten und Verfahren zur Authentisierung sind dem jeweiligen Teilnehmer überlassen. Für den Basisdienst IAM muss allerdings bekannt sein, für welche Option ein Teilnehmer sich entschieden hat, um die notwendigen Abläufe des dieser Entscheidung zugrundeliegenden Protokolls zu initiieren.

Anbindung von Anwendungen

In diesem Szenario übernimmt der Basisdienst IAM die Rolle des Identity Provider (IdP) und die Anwendung die Rolle des Service Providers (SP).

Authentisierung per SAML2

Für Anwendungen, die zur Benutzerauthentifizierung per SAML2 an das F-IAM angebunden sind, gibt es keine weiteren Vorgaben, da das Protokoll keinen Mechanismus vorsieht, in einer bestehenden Session die Benutzerinformationen zu aktualisieren.

Es liegt im Ermessen der Anwendung, je nach Kritikalität regelmäßig einen neuen SAML2-Flow zu initiieren, um Änderungen an den Benutzerinformationen oder auch eine zentrale vorgenommene Sperre mitzubekommen. Durch die Session-Cookies des F-IAM im Browser wird i.d.R. hierbei keine Interaktion des Benutzers notwendig sein, so dass die Benutzerfreundlichkeit nur geringfügig sinken sollte.

Authentisierung per OIDC

Die Prüfung des übergebenen Access-Tokens durch die Anwendung muss die folgenden Aspekte umfassen:

- Das Token ist noch nicht abgelaufen (gemäß Claim „exp“).
- Das Token ist signiert (Der Algorithmus ist nicht „none“).
- Die Signatur ist gemäß Public-Key des Basisdienstes IAM gültig.
  Dadurch wird implizit auch geprüft, dass das Token vom Basisdienst IAM ausgestellt wurde.
- Der eigene Scope ist im Token enthalten (gemäß Claim „scope“).

Autorisierung per Token Exchange

Die Umsetzung erfolgt gemäß RFC 8693 „OAuth 2.0 Token Exchange“. Hierbei wird konkret die *Impersonation* und nicht die *Delegation* verwendet. Die Anwendung erhält also ein für den

Benutzer ausgestelltes Access-Token und ruft damit den P20-Webservice im Namen des Benutzers auf. Für den Webservice spielt es keine Rolle, wie das mitgegebene Access-Token erzeugt wurde, solange es gültig ist.

Bei Übergabe eines gültigen Access-Tokens muss die Anfrage verarbeitet werden. Je nach Berechtigungskonzept der Anwendung kann diese Verarbeitung weitere Prüfungen beinhalten. Ein Beispiel hierfür ist das Sicherstellen, dass dem Benutzer die erforderlichen Funktionsrechte für den Webservice zugewiesen sind.

Verwendung der P20-UID

Die Identifikation des Benutzers muss zukünftig ausschließlich über die P20-UID erfolgen. Dieses Benutzerattribut wird in allen OIDC-Tokens und in perspektivisch auch in den SAML2-Assertions enthalten sein sowie in der Provisionierung übertragen.

Für die datenschutzrechtliche Protokollierung über den entsprechenden Basisdienst ist ebenfalls ausschließlich die P20-UID zu verwenden.

Weitere Details folgen in Kapitel 4.7.2.

Logout

Der Logout an der Anwendung muss keine Endpunkte am F-IAM aufrufen, es wird also kein Single-Logout gefordert oder unterstützt. Folglich wird der Benutzer einer Webanwendung durch das Session-Cookie des F-IAM im Browser beim erneuten Aufrufen unmittelbar wieder eingeloggt sein.

> ***Wichtig***
>
> Durch die delegierte Authentifizierung würde ein Logout am F-IAM nicht genügen. Durch das Session-Cookie des TN-IAM im Browser wäre der Benutzer dennoch automatisch wieder eingeloggt. Aktuell wird davon ausgegangen, dass ein Benutzer immer unter einem dedizierten Windows-Account arbeitet. Um sich tatsächlich wirksam abzumelden, muss er also die Windows-Sitzung beenden, bzw. es muss sich ein neuer Benutzer mit einem anderen Windows-Account anmelden.

Webanwendung

Für die Authentifizierung kann auf Teilnehmerseite entweder SAML2 oder OIDC eingesetzt werden. Der Basisdienst IAM stellt für jede Anwendung ebenso SAML2 oder OIDC bereit. Dabei ist OIDC aus den folgenden Gründen zu bevorzugen, insbesondere wenn die Anwendung auch andere P20-Webservices aufruft.

Die Aktualität der Benutzerattribute und Berechtigungen können sichergestellt werden (durch die Gültigkeit der Access-Tokens), ohne den Benutzerkomfort einzuschränken (durch die Verwendung von Refresh-Tokens).

Es wird nativ die Maschine-zu-Maschine-Kommunikation unterstützt, also speziell der Aufruf von Webservices.

Rich-Client mit Backend

Die Schnittstelle liegt vollständig in der Verantwortung des Herstellers.

> ***Wichtig***
>
> Gegenüber dem F-IAM muss sich ein Rich-Client genauso verhalten wie ein Browser.

Es besteht die Möglichkeit, bei der Anfrage zur Authentifizierung am F-IAM die Identifikation des Teilnehmers über einen Query-Parameter zu spezifizieren. Dies ist speziell für Rich-Clients sinnvoll, da sie auf einem Endgerät installiert sind und die Zuordnung zu einem konkreten Teilnehmer bekannt ist. Dies bietet folgende Vorteile:

- Für den Benutzer entfällt die Notwendigkeit, seinen Teilnehmer bei der Anmeldung explizit auswählen zu müssen.
- Der Rich-Client muss nicht in der Lage sein, die vom F-IAM als HTML-Antwort gesendete Auswahl-Seite anzuzeigen.

Terminal Server

Für die Authentifizierung kann auf Teilnehmerseite entweder SAML2 oder OIDC eingesetzt werden. Der Basisdienst IAM authentisiert den äußeren Perimeter (Gateway) ausschließlich über SAML2.Das Gateway verwendet seine eigenen Methoden, um einen berechtigten Benutzer Zugang zu Anwendungen zu gewähren.

Webservice

Für die Authentifizierung wird ein P20-Access-Token (JWT) verwendet. Anwendungsseitig ist OIDC zu verwenden, da für die Autorisierung des Zugriffs auf den Webservice durch die vorgelagerte Anwendung ein Token Exchange durchzuführen ist.

In diesem Kontext ist geplant, dass zwei Varianten der Ausstellung des initialen Tokens auftreten können:

- F-IAM
  Die vorgelagerte Anwendung ist in die regulären Abläufe der Authentisierung und Autorisierung des Basisdienst IAM eingebunden.
- **TN-IAM**
  Die vorgelagerte Anwendung befindet sich außerhalb der zentralen Topologie und ist in die Abläufe der Authentisierung und Autorisierung des teilnehmerspezifischen AM-System eingebunden.

Die Varianten unterscheiden sich demzufolge im Gültigkeitsbereichs des initial ausgestellten Tokens.

Der in die Anwendung einzubindende Webservice vertraut ausschließlich einem Token, das durch den Basisdienst IAM ausgestellt wurde.

Für den Fall, dass es für eine in der Topologie eines Teilnehmers bereitgestellte Anwendung unumgänglich ist, zentrale P20-Dienste einzubinden, ist folgender Ablauf zu etablieren:

1. TN-Anwendung holt sich über das TN-IAM einen TN-Access-Token (bestenfalls im Rahmen des Logins über OIDC)

2. TN-Anwendung holt sich per Token-Exchange vom F-IAM ein P20-Access-Token für den P20-Webservice
   - F-IAM muss die Tokens des TN akzeptieren → Vertrauensstellung
   - TN-Anwendung muss als OIDC-Client beim F-IAM eingerichtet werden
3. TN-Anwendung ruft einen P20-Webservice mit P20-Access-Token auf
   - Für den P20-Webservice spielt es keine Rolle, wie das Token erzeugt wurde

Identity Management

Konzeptionell ist es vorgesehen, dass die Zuweisung statischer Berechtigungen vornehmlich durch die Teilnehmer erfolgt und die Daten über den Basisdienst IAM durch geeignete Schnittstellen an die Anwendung übertragen werden.

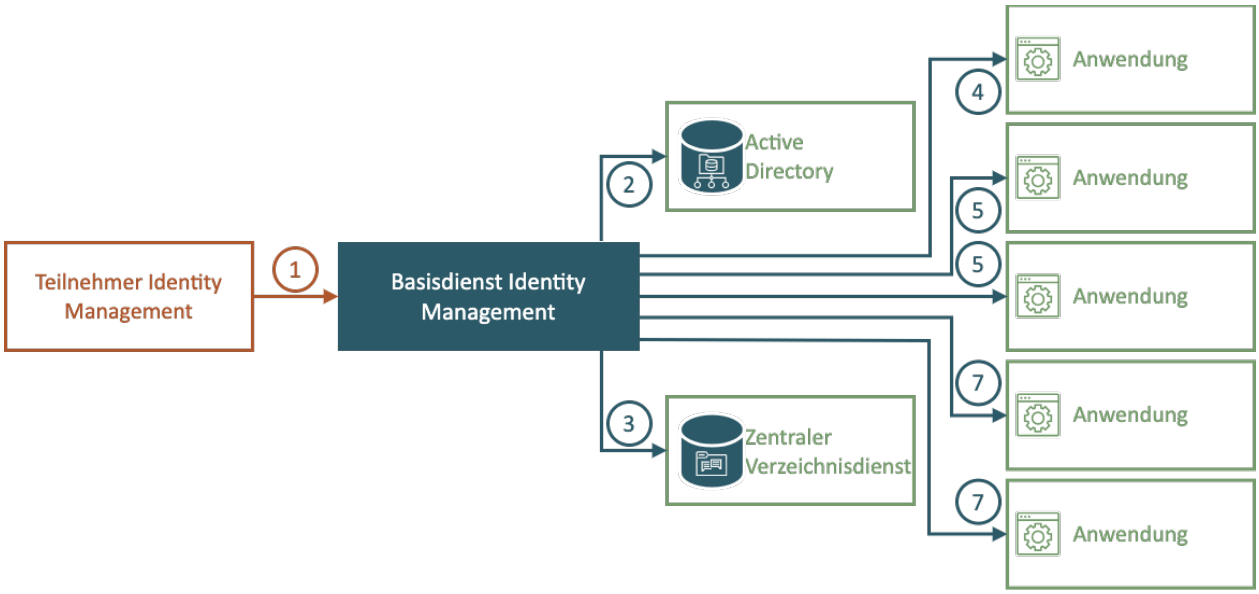

Abbildung 21: Grundprinzip Föderiertes Identity Management

Das durch einen Teilnehmer betriebene Identity Management System ist die autoritative Informationsquelle für Identitäten und deren Zuordnung von Benutzerkonten einschließlich der notwendigen Berechtigungsinformationen für die Anwendungen. Alle Identitäten innerhalb des Basisdienstes IAM sind sogenannte „föderierte Identitäten“ denen ein Pendant innerhalb des Identity Management System des Teilnehmers gegenübersteht. Die notwendige Verwaltung des Lebenszyklus von Identitäten findet demzufolge vollständig auf der Seite des Teilnehmers statt.

Aus der Sicht der Anwendungen ist der Basisdienst IAM die autoritative Informationsquelle für Benutzerkonten einschließlich der notwendigen Berechtigungsinformationen. Der Basisdienst IAM stellt außerdem sicher, dass basierend auf Berechtigungszuweisungen der TN über die von den Anwendungen vorgegebenen Schnittstellen, die entsprechenden Anwendungen innerhalb des Datenhaus-Ökosystems die benötigten Benutzerkonten und Berechtigungsinformationen erhalten, ohne dass eine direkte Integration zwischen dem Identity Management System des Teilnehmers und der Anwendung vorgenommen wurde.

Diese Aufgabentrennung führt zwangsläufig dazu, das aus Sicht eines teilnehmerspezifischen Identity Management Systems der Basisdienst IAM die autoritative Informationsquelle für Anwendungen und deren Berechtigungen darstellt.

Es liegt auf der Hand, das die Realisierung und Handhabung dieser Architektur einige Prozesse voraussetzt, um das Gesamtsystem betreibbar zu gestalten und konsistent zu halten.

Zudem ist es unabdingbar, das über alle Komponenten hinweg zu einem gegeben Zeitpunkt ein konsistenter Zustand erreicht werden muss, das als Ausgangslage definiert werden muss. Diese Ausgangslage wiederum ist nur einmal deterministisch, da sich die einzelnen Komponenten unabhängig voneinander verändern.

Attribute im P20 Basisdienst F-IAM

Eine Übersicht aller aktuell im F-IAM verwendbaren Benutzerattribute und der zugehörigen Bezeichnungen in den verschiedenen Protokollen inkl. Beispiel wird durch das F-IAM-Team in Confluence gepflegt: https://confluence.bka.extrapol.de/x/46DMD.

Grundsätzlich wird vom F-IAM selbst als Pflichtattribut ausschließlich die P20-UID benötigt. Übergangsweise werden derzeit noch weitere Attribute als Pflichtattribute umgesetzt, da dies mit den ersten angebundenen Anwendungen entsprechend vereinbart und diese technische Schuld noch nicht abgebaut wurde. Gelegentlich kommen optionale Attribute hinzu, da sie von einzelnen Anwendungen benötigt werden.

Im Zuge der Datensparsamkeit wird dabei mit den Anwendungsteams geklärt, ob das neue Attribut wirklich aus fachlichen Gründen benötigt wird. Außerdem sollen zukünftig die vom F-IAM ausgestellten Token nur noch die von der jeweiligen Applikation benötigten Attribute enthalten.

TN steht es beim Ausfüllen anderer Attribute als der P20-UID immer frei, eine eigene Pseudonymisierung vorzunehmen und außerdem wenig spezifische Informationen einzutragen wie bspw. bei der Telefonnummer die 110. Entscheidend ist, dass die Benutzer des TN am Ende mit den Anwendungen, die der TN auch tatsächlich verwendet, sinnvoll arbeiten können.

P20-UID

Das für das F-IAM zentrale Benutzerattribut ist die P20-UID (Unique Identifier).

Seitens F-IAM steht eine Schnittstelle bereit, um die P20-UID für Benutzer zu erstellen und so eine einmalige Kennung sicherzustellen. Alternativ kann der TN die UID auch im TN-IDM erstellen. Damit sind die TN dazu in der Lage, die P20-UID als Attribut zu erfassen und gemeinsam mit den anderen Attributen an das F-IAM zu übermitteln zur Verwendung in angebundenen Anwendungen.

Es wurde ein eigenes Feld für die Übertragung einer P20-UID eingeführt. Es gibt bisher allerdings keinen verbindlichen Zeitplan, bis wann die P20-UID von allen TN zu verwenden ist, bzw. dieses Feld als neuer Primärschlüssel verwendet werden kann. Solange das neue Feld nicht von allen TN befüllt wird, kann nicht auf die Verwendung des bisherigen Feldes für die Benutzerkennung verzichtet werden. Um unabhängig von anderen TN eine Datenmigration vom bisherigen identifizierenden Merkmal zu einer P20-UID durchführen zu können, haben TN jedoch die Möglichkeit, die P20-UID zusätzlich in dem bisherigen Feld der Nutzerkennung zu übermitteln. So können einzelne TN, im Kontext von P20, die aktuelle Nutzerkennung durch die P20-UID ablösen.

Aufbau der P20-UID

Die P20-UID umfasst insgesamt sechs Segmente. Die ersten vier Segmente kennzeichnen den TN/IdP, die letzten beiden die jeweilige Identität beim TN/IdP. Die Segmente werden über Feldtrenner (-) voneinander abgekoppelt. Das erhöht die zukünftige Anpassfähigkeit. Anbei die Auflistung der Segmente:

1. Kategorisierung TN oder Partner
2. Informationen zur Staatenzuordnung, des zu der Identität gehörenden Partners oder Teilnehmers
3. Informationen zur Bundeslandzuordung des Teilnehmers (Bund oder Internationale Behörden werden aufgeführt)
4. Die Partner- bzw. Teilnehmer-ID (2-11 Zeichen)
5. Informationen zum Identitätstyp (z.B. Interner Sachbearbeiter, Administrative Identität)
6. 5- bis 11-stelliger-Bereich zur freien Vergabe beim Teilnehmer, der die eindeutige Zuordnung der ID zum Benutzerkonto und damit zur Identität beim Teilnehmer sicherstellt

Eine detaillierte Auflistung der zugeordneten Stellen, der dazugehörigen Beschreibung kann der Tabelle 3: Detaillierter Aufbau der P20-UID entnommen werden. Exemplarische P20-UID-Abbildungen sind in Tabelle 4: Beispiele P20-UIDs gemäß der P20-UID Systematik enthalten.

Auch bei der Zuordnung zum Staat, des TN oder Bund-Länderangabe wird die zulässige Wertemenge über eine eigene Code-Liste vorgegeben, die von P20-IAM-Verantwortlichen gepflegt wird und sich an XPolizei-Katalogwerte anlehnt.

Die derzeit zulässigen Werte der ersten fünf Segmente sind in Confluence zu finden: https://confluence.bka.extrapol.de/x/rN7aBg.

Segment	Anzahl Zeichen	Attribut	Beschreibung	Zulässige Werte
1	1	TN/IdP	P20-Teilnehmer: T Sonstiger Partner: P	Codeliste – Vorgabe durch P20 IAM: Code-Liste: TN = T Partner = P
2	1-3	Staatenzuordnung	Staatenzuordung der Partnerbehörde / des Teilnehmers der Identität. Bei internationalen Partnern (z. B. Europol / Interpol): 0	Codeliste – Vorgabe durch P20 IAM mit Anlehnung an XPolizei-Katalogliste 208 (Staaten)
3	1-2	Bundesland	Länderzuordnung	Codeliste – Vorgabe durch P20 IAM mit Anlehnung

Segment	Anzahl Zeichen	Attribut	Beschreibung	Zulässige Werte
				an XPolizei-Katalogliste Länder: 321, Bundesbehörden:0, Internationale Behörden: 99
4	2-11	Eindeutige TN-ID/IdP-ID	Angabe der jeweiligen TN/IdP-ID (Abstimmung P20 IAM und TN/IdP)	Festlegung durch P20 IAM Betrieb TN/IdP-ID orientiert sich an Katalog 287 Teilnehmerschlüssel
5	3	Identitätstyp	z. B. TN-bezogene Anwender-Identität eines internen Anwenders, TN-bezogene Anwender-Identität eines externen Mitarbeiters, TN-bezogene Administrative Identität eines Mitarbeiters, etc.	Codeliste – Vorgabe durch P20 IAM
6	5-11	Eindeutige ID beim TN	Ein durch den Teilnehmer frei zu vergebener alphanumerischer Wert. Er muss die Eindeutigkeit dieser ID beim TN/IdP verbindlich sicherstellen.	Ein entweder durch den Teilnehmer oder P20 IAM UIDGenerator frei zu vergebener alphanumerischer Wert. Er muss die Eindeutigkeit dieser ID beim TN/IdP verbindlich sicherstellen. Wert 0-9 bzw. A-Z

Tabelle 3: Detaillierter Aufbau der P20-UID

Die folgende P20-UID stellt die exemplarisch die UID eines Mitarbeiters dar, der in der Bundespolizei für P20 sachbearbeitend tätig ist.

Abbildung 22: Exemplarische Abbildung der P20-UID eines Mitarbeiters der Bundespolizei

Die folgende P20-UID stellt die exemplarisch die UID eines Mitarbeiters dar, der bei der Polizei NRW in einem oder mehreren Fachverfahren als Administrator tätig ist.

Abbildung 23: Exemplarische Abbildung der P20-UID eines Mitarbeiters der Polizei NRW

Im Folgenden werden exemplarische Aufbauten einer P20-UID mit unterschiedlichen Varianten exemplarisch dargestellt.

Identitätsbeschreibung	UID
Anwenderkonto Interner Mitarbeiter Sachbearbeitung Bundespolizei	T-36-0-18-101-4123456
Anwenderkonto Mitarbeiter Sachbearbeitung NRW	T-36-5-05-101-NW056731
Administrationskonto für Fachanwendungen eines Mitarbeiters NRW	T-36-5-05-102-NWO56731
Anwenderkonto Mitarbeiter Sachbearbeitung einer internationalen Partnerbehörde (nur exemplarische Darstellung)	P-0-99-A17567-101-XYZ1234591

Tabelle 4: Beispiele P20-UIDs gemäß der P20-UID Systematik

Generierung der P20-UID

Die eigenständige Generierung der P20-UID kann in der Benutzerverwaltung der TN erfolgen und dem entsprechenden Benutzerobjekt im TN-IDM zugeordnet werden. Alternativ kann ein durch das P20 IAM bereitgestellter UID-Generierungsservice zur Erzeugung genutzt werden. Auch hier erfolgt die Zuordnung der UID zum Benutzerobjekt im TN-IDM-System. Grundsätzlich sollte sich ein TN dauerhaft für eine der genannten Generierungs-Varianten entscheiden. Ob und wie eine Umstellung möglich ist, ist ggf. im Einzelfall zu klären.

- Die generierte P20-UID muss folgende Eigenschaften besitzen:
- Die P20-UID wurde bis dato nicht vergeben, ist also einmalig.
- Die P20-UID ist syntaktisch richtig aufgebaut.
- Die Werte der TN- oder Partnerbezogenen Segmente stimmen mit den für diesen TN vorgesehenen Werten überein.
- Die vergebenen Werte in den einzelnen Segmenten sind zulässige Werte, also in den Codelisten für das jeweilige Segment enthalten.

P20-Dienststellenschlüssel

Ein weiteres Attribut, das besondere Bedeutung hat, ist der P20-Dienststellenschlüssel.

Hintergrund

Jeder TN hat eine organisatorische Hierarchie, wobei die einzelnen Entitäten unterschiedliche Bezeichnungen haben. Gebräuchlich sind Dienststelle oder Organisationseinheit. Teilweise

werden bei einem TN beide Bezeichnungen verwendet, um bestimmte Hierarchieebenen zu beschreiben. Es ist aber nicht einheitlich, was die gröbere und was die feinere Bezeichnung ist.

Es gibt aktuell verschiedene Stellen, an denen für mehrere TN Dienststellen gepflegt sind - aber nirgends vollständig. Es gibt keine TN-übergreifend einheitliche Konvention für den Aufbau von Dienststellenschlüsseln, der eine globale Eindeutigkeit sicherstellt.

Alle iVBSe benötigen die Hierarchie und zahlreiche Attribute zu den Dienststellen.

Zur Vereinheitlichung wurde eine AG gebildet. Der Basisdienst Kataloge soll zur Umsetzung einen P20-Dienststellenkatalog bereitstellen.

Auswirkung auf das F-IAM

Es wird ein zusätzliches Benutzerattribut für den Transport des neuen, einheitlichen P20-Dienststellenschlüssels eingeführt werden, um die bisher inkonsistent und intransparent genutzten drei Felder für Dienststelleninformationen abzulösen.

Weitere Informationen zu den Dienststellen über den P20-Dienststellenschlüssel hinaus werden zukünftig durch den P20-Dienststellenkatalog verfügbar. Nicht jedoch direkt über das F-IAM. Bei Bedarf durch Anwendungen ist der Basisdienst Kataloge zusätzlich anzubinden. Der P20-Dienststellenschlüssel ist im F-IAM vorrangig relevant, um Berechtigungen mit Bezug zur Organisationseinheit ermöglichen zu können – nicht, um Informationen über Personen und ihre Dienststellen abzurufen.

Anforderungen zur Sicherstellung der Konsistenz

Alle vorgenannten Aspekte beziehen sich auf die Kopplung der IT-Systeme mit dem Basisdienst IAM. Nämlich mit der Bereitstellung von Metadaten, die vom teilnehmerspezifischen IAM-System verwendet werden muss, um über diese Indirektion die Beziehung zwischen Benutzerkonten in Anwendungen zu spezifizieren. Das beinhaltet nicht, wie diese Metadaten auf der Teilnehmerseite zusammengefaßt werden, um die Zuweisung und den Entzug von Berechtigungen benutzerfreundlicher zu gestalten.

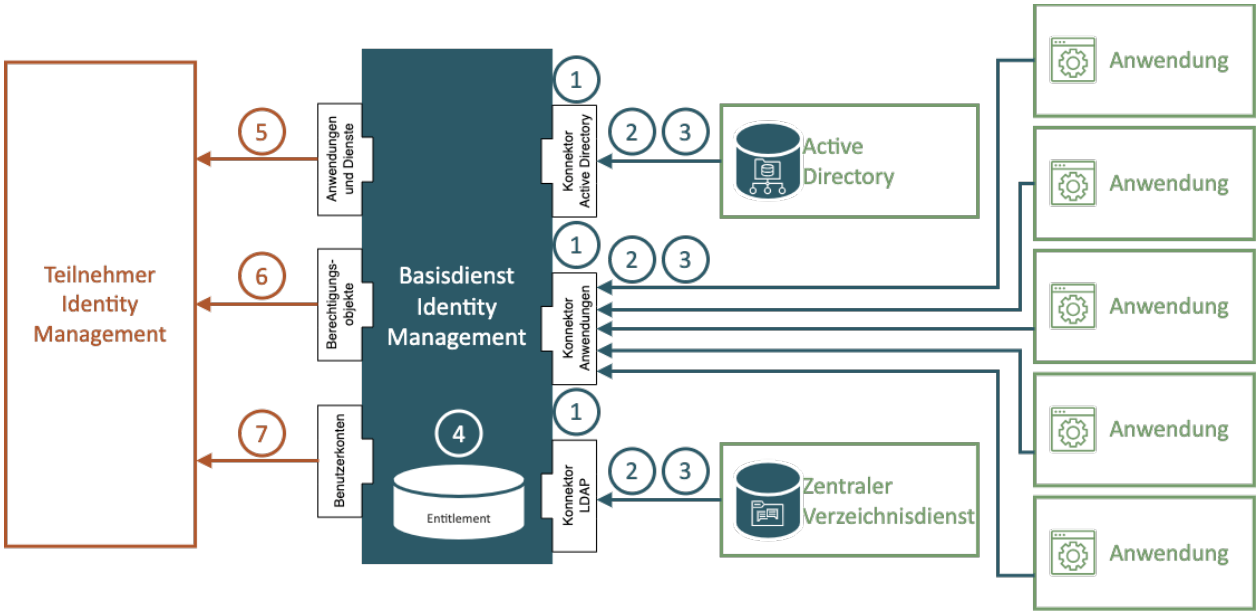

Abbildung 24: On-Boarding Föderiertes Identity Management

Über die Zeitachse gesehen wird sich der Umfang von Anwendungen, die an den Basisdienst IAM angeschlossen sind, ändern. Mit anderen Worten, es findet ein beständiges On-Boarding und Off-Boarding von Anwendungen statt. Dieser Lebenszyklus von Anwendungen ist für IAM Systeme ein regulärer Vorgang.

Neben den Veränderungen des Umfangs an Anwendungen, die durch ein IAM System gesteuert werden, ergeben sich innerhalb der Anwendungen selbst Änderungen der durch sie bereitgestellten Berechtigungsinformationen, oder sogar Berechtigungsstrukturen. Das ist demzufolge als Lebenszyklus von Berechtigung zu berücksichtigen und für IAM Systeme ein regulärer Vorgang.

Ein teilnehmerspezifisches IAM bedient sich dieser Metadaten, um seinerseits die entsprechenden Korrelationen zwischen Personen, Anwendungen, Benutzerkonten und Berechtigungen herzustellen.

> ***Wichtig***
>
> Das ist kein einmaliger Prozess der lediglich die initiale Konstellation abgleicht, sondern vielmehr ein permanenter Prozess, der die durch Änderungen in der IT-Landschaft zwangsläufig auftretenden Inkonsistenzen in den verteilten Systemen minimiert.

On-Boarding

Der reguläre Prozess der Anbindung einer Anwendung an den Basisdienst IAM gestaltet sich in folgenden abstrakten Vorgehen:

1. Bereitstellung der essentiellen Artefakte (Datenmodell, Workflow, Konnektor, usw.) um einen Anwendung anbinden zu können.
2. Auslesen der Berechtigungen.

   Das kann unterschiedliche Typen von Berechtigungen beinhalten von einfach strukturierten Rechten, wie Gruppen in einem Verzeichnisdienst bis hin zu komplexen aus mehreren Attributen bestehende Berechtigungsobjekte.

3. Initialer Abgleich der Bestandskonten einer Anwendung (Account Discovery)
   Dieser Prozess beinhaltet auch die Bereinigung der verwaisten Benutzerkonten.
4. Transformation der Berechtigungen zu eigenständigen Entitäten (Entitlement)

Off-Boarding

Der mögliche Prozess der vor der endgültigen Abschaltung einer Anwendung vorangehen sollte gestaltet sich folgendermaßen:

1. Außer Kraft setzen aller Zuweisungsrichtlinien aus den RBAC-Rollen, die die Anwendung beinhalten. Dieser Prozess kann einige Zeit in Anspruch nehmen, bis alle Richtlinien wirksam entfernt wurden. Da die RBAC-Rollen selbst Gegenstand von Genehmigungsprozessen sein können die eine entsprechende Bearbeitungszeit beanspruchen.
2. Dekommissionieren aller Berechtigungen die zu der Anwendung gehören
3. Dekommissionieren der eigentlichen Anwendung

Abgleich

Wenn beim regelmäßigen Vollabgleich zwischen F-IAM und Anwendung eine Abweichung festgestellt wird, passiert Folgendes: Für die Anwendung ist das F-IAM das führende System. Werden Abweichungen identifiziert, sendet das F-IAM daher Korrektur-Nachrichten an die Anwendung, um die Anwendung auf den Stand des F-IAM (und damit den Stand des TN) zu bringen. Hierdurch werden sowohl Änderungen übertragen als auch veraltete Stände durch das Einspielen von Backups bei der Anwendung wieder auf den aktuellen Stand gebracht.

Die Änderungen in der Anwendung können allerdings durchaus auch bewusst vorgenommen worden sein, bspw. da ein Benutzer eine bestimmte Berechtigung dringend benötigt und nicht auf das TN-IAM-Team warten kann. Daher besteht die Möglichkeit, den Abgleich durch das F-IAM temporär auszusetzen.

Der Abgleich wird ebenfalls ausgesetzt, falls im F-IAM ein Backup eingespielt werden muss, bis der Abgleich mit allen TN-IAM-Systemen erfolgt ist. Wenn beim regelmäßigen Abgleich mit einem TN-IAM-System eine Abweichung festgestellt wird, wird das F-IAM an den Stand des TN-IAM-Systems angepasst, da dieses das führende System ist.

Wenn bei einer Provisionierung in eine Anwendung ein Fehler auftritt, wird dieser Fehler in die entsprechende TN-spezifische Fehlerliste des F-IAM eingetragen. Das TN-IAM-Team kann diesen Fehler dann ggf. in Abstimmung mit dem Anwendungsteam beheben.

> ***Beispiel***
> Szenario: Ein TN überträgt eine Berechtigungszuweisung für eine der Anwendung noch unbekannte OE, die entweder dem P20-Dienststellenkatalog neu hinzugefügt oder im TN-IAM falsch eingetragen wurde.
>
> Ablauf:
> - Die Anwendung meldet beim Provisionieren durch das F-IAM einen Fehler.
> - Die Berechtigungszuweisung wird im F-IAM gelöscht.
> - Das F-IAM fügt in die TN-spezifische Fehlerliste einen Eintrag ein.
> - Die Fehlerliste ist für die TN zugänglich und muss geeignet berücksichtigt werden.
> - Spätestens beim regelmäßigen Vollabgleich des TN mit dem F-IAM wird die fehlende Berechtigungszuweisung erkannt und geeignet behandelt (z. B. erneut gesetzt, weil inzwischen auch die Anwendung die OE kennen sollte, oder auch beim TN geändert, weil es die OE tatsächlich nicht gibt.)

Schnittstellen für Teilnehmer

Der Basisdienst IAM stellt den Teilnehmern eine Schnittstelle (Strukturdaten im JSON-Format über eine REST API) bereit, an welcher der Teilnehmer für alle Anwendungen bzw. Anwendungsinstanzen die für ihn relevanten Anwendungsfunktionsrechte abrufen kann. Die Schnittstelle bietet dem Teilnehmer eine Filtermöglichkeit zur Einschränkung des Ergebnisses auf bestimmte Anwendungen oder Anwendungsinstanzen

Die für bzw. vom Basisdienst IAM bereitgestellten Schnittstellen unterscheiden grundsätzlich zwischen den TN, welche mit der Schnittstelle interagieren. Dies bedeutet für diese Schnittstellen, dass Authentifizierungsverfahren zur Identifizierung des jeweiligen Teilnehmers zum Einsatz kommen.

Der Basisdienst F-IAM stellt für den Abgleich folgenden Schnittstellen bereit:

Die für bzw. vom Basisdienst IAM bereitgestellten Schnittstellen unterscheiden grundsätzlich zwischen den Teilnehmern, welche mit der Schnittstelle interagieren. Dies bedeutet dass für diese Schnittstellen Authentifizierungsverfahren zur Identifizierung des jeweiligen Teilnehmers zum Einsatz kommen.

Der Basisdienst IAM stellt dafür folgende Schnittstellen bereit:

- Abruf der an den Basisdienst F-IAM angebundenen Anwendungen und Dienste (Abbildung Ziffer 5)
- Abruf der Berechtigungsobjekte aller an den Basisdienst F-IAM angebundenen Anwendungen und Dienste (Abbildung Ziffer 6)
- Abruf der Benutzerkonten aller an den Basisdienst F-IAM angebundenen Anwendungen und Dienste (Abbildung Ziffer 7)

Das TN-IAM verwendet die durch den Basisdienst F-IAM bereitgestellten Daten, um diese mit seinem spezifischen Datenmodel abzugleichen.

Anbindung von Teilnehmern

Die Anbindung von Teilnehmern ist über verschiedene Protokolle möglich. Hier wird eine kurze Übersicht gegeben. Technische Schnittstellendokumentationen sind in Confluence zu finden: https://confluence.bka.extrapol.de/x/qN7aBg.

Dazu zu sagen ist noch, dass es aufgrund von verschiedenen Datenmodellen trotz standardisierter Schnittstellen notwendig sein kann, zusätzlich einen Konnektor zu entwickeln – entweder seitens der Anwendung (empfohlen für unabhängige Umsetzbarkeit) oder seitens des F-IAM (klassischer Ansatz; möglich, wenn Aufwände leistbar sind und ggf. durch Budget des Anwendungsteams abgedeckt werden können).

Bei Eigenentwicklungen empfiehlt es sich daher, das Datenmodell frühzeitig mit dem F-IAM-Team abzustimmen, um spätere Aufwände der Anbindung von vornherein zu vermeiden.

Einbringung P20-UID

Pro TN/IdP ist abzustimmen, ob der UID-Generator genutzt werden oder die P20-UID durch den TN erstellt werden soll. Näheres zur P20-UID erfolgte im Kapitel P20-UID.

Anlegen eines P20-Benutzerkontos

Um einem Benutzer Zugriff auf eine P20-Anwendung gewähren zu können, ist es erforderlich, zunächst eine Identität im Basisdienst IAM (P20-Benutzerkonto) anzulegen und mit dem TN-eigenen Benutzerkonto (TN-Benutzerkonto) zu verknüpfen.

Die PG IAM empfiehlt allen TN, die Provisionierung eines P20-Benutzerkontos in den eigenen Onboarding-Prozess für neue Mitarbeiter zu integrieren. So kann z. B. im Zuge des automatischen Anlegens eines Windows-Kontos sowie eines E-Mail-Kontos auch automatisch ein P20-Benutzerkonto erstellt und ggf. mit grundlegenden Berechtigungen für bei der Organisationseinheit des Benutzers gebräuchliche P20-Anwendungen versehen werden. (Bspw. kann automatisch ein Account in Jira/Confluence angelegt werden.)

Aktualisieren eines P20-Benutzerkonto

Außer dem Anlegen und dem Löschen von Identitäten ist jede Aktion, die der Verwaltung von Benutzern dient, eine Aktualisierung. Auch das Deaktivieren eines Benutzers, nicht zu verwechseln mit der permanenten Löschung eines Kontos, erfordert es, die Daten zu einer Identität zu aktualisieren (Statusänderung).

Die PG IAM empfiehlt allen TN, das P20-Benutzerkonto im eigenen Identity Lifecycle-Management zu berücksichtigen. Konkret bedeutet dies, innerhalb der Benutzerverwaltung zu prüfen, ob sich Änderungen an einem TN-Benutzerkonto auf das P20-Benutzerkonto auswirken, und wenn ja, diese Änderung durch ein Aktualisieren des P20-Benutzerkontos dem Basisdienst IAM mitzuteilen. Dies könnte z.B. der Fall sein, wenn sich die E-Mail-Adresse eines Mitarbeiters aufgrund einer Namensänderung ändert.

Ändern des Benutzer-Status bzw. von Benutzer-Attributen

Die Attribute des Benutzerkontos (z.B. Name, E-Mail-Adresse, Status) können durch eine Aktualisierung des Benutzerkontos geändert werden. Eine Ausnahme stellt hierbei die P20-UID dar, die als primärer Identifizierer für Benutzerkonten verwendet wird. Sie kann folglich nicht mehr geändert werden.

Die PG IAM empfiehlt allen TN, im Zuge eines eigenen Identity Lifecycle-Managements Benutzer, die temporär abwesend sind (z.B. wegen Elternzeit), zu deaktivieren statt zu löschen.

Zuweisen und Entfernen von Berechtigungen

Dem Basisdienst IAM bekannte Berechtigungen (Anwendungsfunktionsrechte bzw. Rollen) können bereits existierenden Benutzerkonten zugewiesen oder entzogen werden.

Die PG IAM empfiehlt allen TN, im Rahmen eines eigenen Identity Lifecycle-Managements Berechtigungszuweisungen regelmäßig zu überprüfen und ggf. Berechtigungen zu entziehen, die nicht mehr benötigt werden. Eine solche Überprüfung könnte z.B. automatisch stattfinden, wenn ein Mitarbeiter die Abteilung wechselt bzw. andere Aufgaben erhält.

Grundsätzlich obliegt es den TN, ihren jeweiligen Benutzern Berechtigungen zuzuweisen und zu entziehen. Nach Bedarf kann zusätzlich eine Genehmigung durch Applikationsteams durch das F-IAM ermöglicht werden. Dies ist dann sinnvoll, wenn es sich um Berechtigungen in Applikationen handelt, bei denen das Applikationsteam besser beurteilen kann, ob ein User sie erhalten sollte, insbes. für administrative Berechtigungen in der Applikation.

Löschen eines P20-Benutzerkonto

Nachdem ein P20-Benutzerkonto im Basisdienst IAM gelöscht wurde, kann es nicht mehr reaktiviert werden.

Auch eine Neuanlage unter Verwendung derselben P20-UID ist nicht möglich.

Für die TN ist es außerdem erforderlich, gem. der DSGVO sowie ggf. eigenen Landespolizeigesetzen die gesetzlichen Löschfristen zu berücksichtigen. Umgekehrt ist es erforderlich, dass TN vor Ablauf dieser Fristen in der Lage sind, eine P20-UID auch auf einen ehemaligen Mitarbeiter zurückzuführen.

Zentrale Sperrung eines P20-Benutzerkonto

Beim zentralen IAM-Betrieb ist eine zentrale Sperrung eines Benutzers im Basisdienst IAM vorgesehen. Dies bedeutet, dass für einen Benutzer der Zugang zu zentralen, durch den Basisdienst IAM geschützten Anwendungen und Dienste unabhängig vom Status des Benutzers seines Heimat-IdPs (i.d.R. der TN-IdP) gesperrt werden kann. Diese Sperrung kann dann vorgenommen werden, wenn von einer Gefährdungslage auszugehen ist. Das entsprechende organisatorische Regelwerk sowie die notwendigen Anweisungen zum Vorgehen zur Sperrung und Entsperrung werden gesondert beschrieben.

Schnittstellen für Teilnehmer

LDAP-Legacy

Hierbei handelt es sich um den Industriestandard von LDAP. Die Verwaltung beschränkt sich ausschließlich auf die Entität Identity. Jegliche anderen administrative Aktivitäten werden durch autorisierte Administratoren eines Teilnehmers vollzogen, um Endanwender mit den für ihr Aufgabengebiet notwendigen Zugriffsberechtigungen auszustatten. Dafür verwenden diese Administratoren die webbasierte Oberfläche des Basisdienst F-IAM.

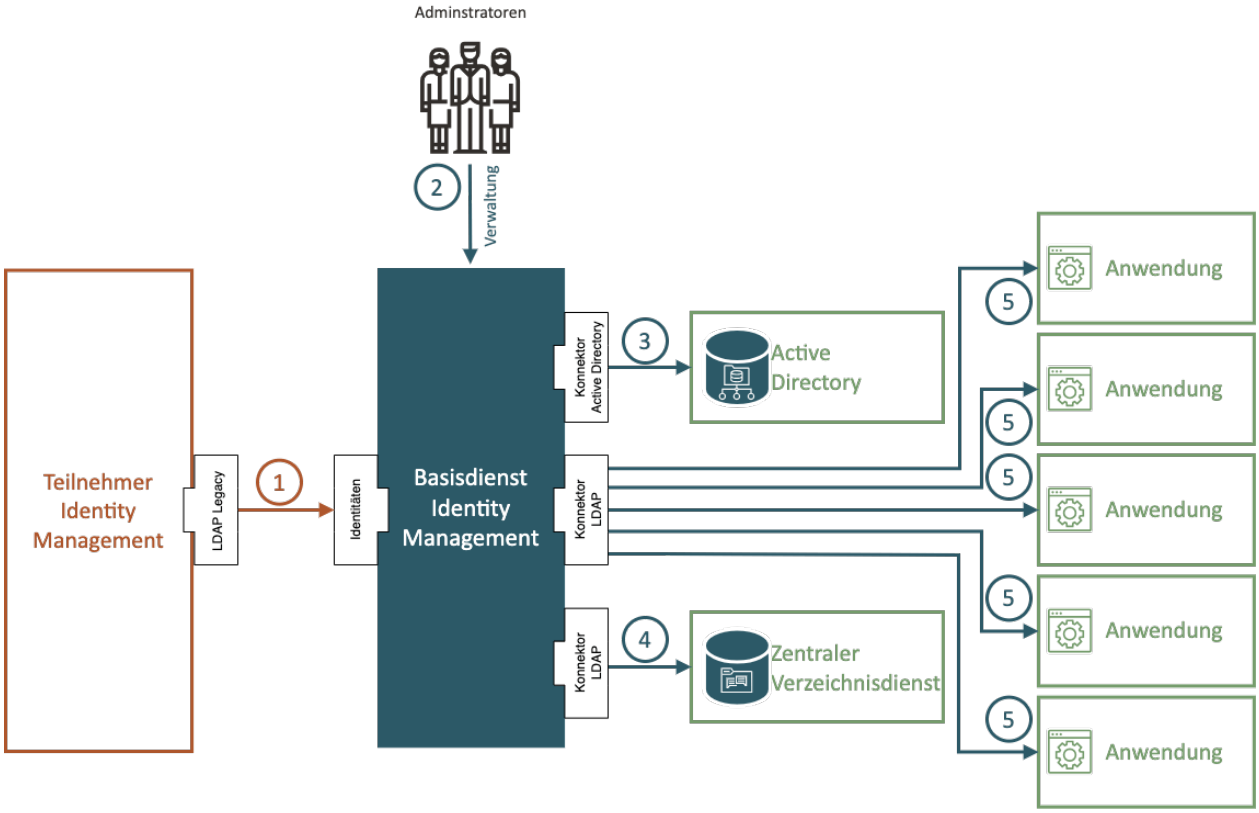

Abbildung 25: Grundprinzip LDAP-Legacy

Diese Funktionalität bleibt weiterhin erhalten. Es besteht somit keine Verbindlichkeit auf eine der anderen Varianten zur Verwaltung von Identitäten zu wechseln.

Datenquelle

Die Datenquelle auf die sich diese Variante zur Verwaltung von Identitäten abstützt ist ein durch den Teilnehmer bereitgestellter Verzeichnisdienst, der durch ein vorgelagertes Verwaltungssystem mit den verbundrelevanten Identitätsdaten versorgt.

Verfahren

Die Datenquelle führt eine Struktur (z.B. ou=P20-Benutzer), in dem Benutzerkonten angelegt werden, die vom Basisdienst IAM als „föderierte Identitäten“ interpretiert werden. Jedem Benutzerkonto in dieser Datenquelle sind die abgestimmten Attribute zugewiesen, die für die Weitergabe an die an den Basisdienst IAM angeschlossenen Anwendungen notwendig sind. Der Basisdienst IAM prüft die über alle angeschlossenen Anwendungen hinweg als verbindlich deklarierten Attribute auf Konsistenz bevor der Basisdienst IAM die zugehörige „föderierte Identität“ erzeugt.

Eine vormals erzeugte „föderierte Identität“, für die sich keine Korrelation innerhalb der Struktur mehr herstellen läßt wird durch den Basisdienst IAM als Löschung dieser „föderierten Identität“ interpretiert, was zwangsläufig zum Entzug aller Benutzerkonten und den mit diesen Konten verbundenen Berechtigungen in den angeschlossenen Anwendungen einhergeht.

Wird eine Änderung in den Attributen einer „föderierte Identitäten“ und dem aktuellen Werten dieser Attribute in der Datenquelle festgestellt, werden die Konsistenz der Werte geprüft, ob die über alle angeschlossenen Anwendungen hinweg als verbindlich deklarierten Attribute nach wie vor gewahrt ist, bevor eine Änderung an den Attributen der korrelierenden „föderierte Identitäten“ vorgenommen wird. Dadurch werden dann auch die Änderungen der Attribute in den angeschlossenen Anwendungen geändert.

Integration

Der Basisdienst IAM greift zyklisch auf die durch die Teilnehmer breitgestellte Datenquelle zu.

Das angewendete Verfahren erfolgt als *asynchron* (siehe Asynchrone Replikation) und folgt entsprechend Push- und Pull-Ansatz dem *Pull-Ansatz*.

Authentisierung

Durch den Teilnehmer werden zur Identifizierung des Basisdienst IAM die entsprechenden Zugangsberechtigen eingerichtet und die Nutzung notwendigen Zugangsdaten an den Betreiber des Basisdienst IAM übermittlet.

Transportsicherung

Die Absicherung der Transportschicht für den angebotenen Verzeichnisdienst erfolgt durch Einbringen der dafür notwendigen Zertifikate in den Basisdienst IAM.

Varianten

Mit einzelnen TN und einzelnen hoch priorisierten Anwendungen wurden Varianten von LDAP-Legacy als niedrigschwellige Übergangslösung vereinbart. (LDAP-Legacy+, LDAP-Generic)

Diese sind jedoch keinesfalls als langfristige P20 Standards zu verstehen und sollten nur dort eingesetzt werden, wo sie benötigt werden, um Zeitpläne einhalten zu können. Falls hier von weiteren Teams Bedarf besteht, so ist dieser individuell mit dem F-IAM-Team abzustimmen.

Langfristig ist auf SCIMv2-Extended umzustellen, da LDAP-Varianten in folgenden Punkten technisch unterlegen sind:

- aufwändig in Wartung/Trouble-Shooting
- große Latenz – nur zyklische Updates
- kein Datenrückfluss bei Berechtigungszuweisungen, die durch Anwendung freigeschaltet werden müssen
- Auflösen von Anwendungsrollen durch F-IAM nicht möglich, nur direkte Zuweisung von Anwendungsrechten
- keine anwendungsübergreifenden TN-Rollen möglich
- keine Berechtigungszuweisungen mit OE-Bezug möglich

SCIMv2

Die Anbindung per SCIMv2 erfolgt gemäß:

- RFC 7642 „SCIMv2: Definitions, Overview, Concepts, and Requirements“
- RFC 7643 „SCIMv2: Core Schema“
- RFC 7644 „SCIMv2: Protocol“

Die Verwaltung von Benutzerkonten in Anwendungen und Diensten und die damit einhergehende Zuweisung bzw. der Entzug von Berechtigungen ist mit diesen Standards über den Basisdienst IAM nicht möglich. Im Kontext von SCIMv2 ist der selbst Basisdienst IAM auf den die SCIMv2 Schnittstelle einwirkt. Um auf Anwendungen und Dienste einwirken zu können ist eine Schemaerweiterung erforderlich, die im Rahmen der zentralen Bereitstellung noch zu spezifizieren ist.

Aus diesem Grund wird zwischen den Schnittstellen „SCIMv2-Core“ für die Basisfunktionalität und „SCIMv2-Extended“ für die schemaerweiterte Variante unterscheiden.

SCIMv2-Core

Bei SCIMv2-Core handelt es sich um den Industriestandard SCIM 2.0. Ähnlich wie bei LDAP-Legacy können auch über SCIMv2-Core nicht alle benötigten Berechtigungsinformationen übertragen werden, sondern lediglich Identitäten angelegt werden.

SCIMv2-Extended

Bei SCIMv2-Extended handelt es sich dem Namen entsprechend um eine Erweiterung des Industriestandards, der grundsätzlich darauf ausgelegt ist, erweitert zu werden. Diese Erweiterungen dienen dazu, aus dem TN-IAM-System heraus eine Zuweisung von Berechtigungen im F-IAM zu ermöglichen.

> ***Wichtig***
> Bei SCIMv2-Extended handelt es sich um die langfristig empfohlene IDM-Schnittstelle seitens F-IAM.

Anbindung von Anwendungen

Die Provisionierung ist über verschiedene Protokolle möglich. Hier wird eine kurze Übersicht gegeben. Technische Schnittstellendokumentationen sind in Confluence zu finden: https://confluence.bka.extrapol.de/x/qN7aBg.

Dazu zu sagen ist noch, dass es aufgrund von verschiedenen Datenmodellen trotz des Bestrebens nach standardisierten Schnittstellen notwendig sein kann, zusätzlich einen Konnektor zu entwickeln – entweder seitens der Anwendung (empfohlen) oder seitens des F-IAM (möglich, wenn Aufwände leistbar sind und ggf. durch Budget der Anwendungsteam werden können).

Bei Eigenentwicklungen empfiehlt es sich daher, das Datenmodell frühzeitig mit dem F-IAM-Team abzustimmen, um spätere Aufwände der Anbindung von vornherein zu vermeiden.

Die initiale Bereitstellung erfolgt im Rahmen der Anbindung der jeweiligen Anwendungsinstanz an den Basisdienst IAM. Die Aufnahme erfolgt in Abstimmung zwischen dem Anwendungsverantwortlichen und dem Basisdienst IAM über geeignete Schnittstellen (z.B. Einlesen einer Datei, Nutzung einer Webserviceschnittstelle).

Die Anwendungsfunktionsrechte einer Anwendung ändern sich in der Regel über den Nutzungszeitraum. Bspw. kommen durch Weiterentwicklung der Anwendung neue Anwendungsfunktionsrechte hinzu, bestehende entfallen oder Beschreibungen werden angepasst. Die Initiierung von Änderungen liegt in Verantwortung des Anwendungsverantwortlichen und muss Ende-zu-Ende, also von der Anwendung bis zum Teilnehmer, koordiniert werden. Diese Koordinationsaufgabe liegt nicht in Verantwortung des Basisdienstes IAM.

Bei vereinzelten, nicht automatisierbaren Vorgängen zur Pflege der Anwendungsfunktionsrechte im Basisdienst IAM kann die Notwendigkeit der Unterstützung durch den IAM Betrieb notwendig werden. Bei Bedarf ist der IAM Betrieb über ein entsprechendes Service Request Verfahren durch den Anwendungsverantwortlichen einzubinden.

Rollenmanagement

ollen erleichtern die Zuweisung von Zugriffsberechtigungen an Benutzer und sowie die fortlaufende Überwachung dieser Zuweisungen. IAM-Lösungen bieten dafür umfassende Funktionen.

Die rollenbasierte Zugriffskontrolle gewährleistet eine höhere Sichtbarkeit und erleichtert die Zuweisung und Aufhebung der Zuweisung von Zugriffsrechten an Benutzer, setzt das Konzept der geringsten Rechte durch und ermöglicht Einblicke in die Einhaltung der Compliance.

Die rollenbasierte Zugriffskontrolle wächst und erweitert sich in der Regel, wenn neue Situationen auftreten, z. B. wenn neue Anwendungen integriert oder bestehende Anwendungen dekomissioniert werden, oder wenn sich die geschäftlichen Anforderungen ändern. Der wesentlichen Vorteile dieses Ansatzes ist die einfache Implementierung und die Überwachung der Compliance.

Terminologie

Im Kontext P20 bezeichnet eine Business-Rolle eine anwendungsübergreifende, an der Funktion eines Mitarbeiters ausgerichtete Sammlung von Berechtigungen. Eine Business-Rolle hat die primäre Aufgabe einen Stapel von Berechtigungen, für unterschiedlichste Fachanwendungen, zu repräsentieren. Dabei orientiert die Zusammensetzung der zusammengefassten Berechtigungen sich an der Funktion des Mitarbeiters bzw. den genutzten Geschäftsprozessen.

Im Zuge der Berechtigungsverwaltung von Mitarbeitern ist es ratsam, fertige Rechtekombinationen in Rollen zusammenzufassen, um diese vereinfacht Mitarbeitern zuweisen oder entziehen zu können. Dabei kann unter Anderem zwischen IT-Rollen, technischen Rollen, Anwendungsrollen und Business-Rollen unterschieden werden. In der technischen Umsetzung werden all diese Rollen i.d.R. identisch verarbeitet, d.h. es gibt keine technische Unterscheidung zwischen Anwendungs- und Business-Rollen.

Im Basisdienst IAM wird zwischen Rechten und Rollen unterschieden. Um diese Trennung deutlicher zu machen, werden die Begriffe Anwendungsfunktionsrechte und TN-Rollen (TN = Teilnehmer) verwendet.

Anwendungsfunktionsrechte sind die aus Sicht des zentralen IAM-Systems atomaren Berechtigungsentitäten, die einem Benutzer zugewiesen oder eben nicht zugewiesen sein können. In Abhängigkeit dieser Zuweisungen erlaubt die Anwendung dem Benutzer Zugriff auf bestimmte Funktionalitäten oder verwehrt diese. Anwendungsfunktionsrechte werden somit von den Anwendungen vorgegeben und im F-IAM registriert.

Innerhalb der Anwendung können diese Anwendungsfunktionsrechte technisch weiter aufgeteilt werden, weil beispielsweise verschiedene Funktionen Zugriff auf dieselbe Datenbank-Tabelle benötigen. Dies muss innerhalb der Anwendung und ohne Möglichkeiten der Einflussnahme eines Benutzers erfolgen. Der Hintergrund für diese Vorgabe ist, dass im F-IAM die Berechtigungszuweisungen revisionssicher protokolliert werden müssen. Hätte ein Benutzer die Möglichkeit, die Wirkung eines zugewiesenen Anwendungsfunktionsrechtes eigenständig zu ändern, wäre die Protokollierung im F-IAM nicht mehr aussagekräftig.

Zentrale Verarbeitung

Konzeptionell ist es vorgesehen, dass die IAM-Lösungen der Teilnehmer Rollen erstellen und im Basisdienst IAM hinterlegen.

Für die langfristige Konsolidierung von (Business-)Rollen bzw. Rollen im Allgemeinen, ist eine zentrale Speicherung und Verarbeitung von Rollen-Definitionen sinnvoll. Die zentrale Verarbeitung und automatische Identifikation von Schnittmengen ermöglicht es, langfristig aus dem IST-Zustand neue Rollen zu schöpfen mit dem Ziel, den Nutzerkreis dieser neuen, konsolidierten Rollendefinitionen zu vergrößern.

Ziel ist es, den Benutzerverwaltungen der Teilnehmer zu ermöglichen, flexibel und nach Baukasten-Prinzip, Rechte und Rollen in Business-Rollen zu orchestrieren und das Resultat als Rollendefinition abzuspeichern. In der Benutzerverwaltung des Teilnehmers erfolgt später lediglich die Zuweisung einer auf die Rollendefinition abgebildeten Entsprechung. Diese Entsprechung wird im Zuge der Autorisierung später, für den Nutzer unsichtbar, in für die Anwendung verständliche Anwendungsrechte bzw. Anwendungsrollen übersetzt.

Die Beschreibung des zentralen Konfigurators beschreibt primär den Prozess wie Rechte und Rollen von Basisdienst IAM verarbeitet werden sollen und welche Werkzeuge zur Rollenmodellierung zur Verfügung gestellt werden. Die konkrete inhaltliche Ausarbeitung von Rollen jeder Art ist jedoch nicht Bestandteil dieser Arbeit.

Rollenmodellierung

Mit dem Ziel, eine Arbeitsgruppe zu schaffen, die sich um die konkrete Ausgestaltung (Modellierung) von Business-Rollen kümmern soll, stellte die PG IAM Überlegungen dazu an, welche Personen mit welchen Profilen für eine solche Arbeitsgruppe relevant sind. Außerdem wurde versucht, eine konkrete Zielsetzung für diese neu zu gründende AG zu formulieren. Beide Vorhaben konnten von der PG IAM nicht abschließend umgesetzt werden.

Das übergeordnete Ziel Business-Rollen mit einer möglichst großen Reichweite (idealerweise bundeseinheitlich) zu definieren, würde mit hoher Wahrscheinlichkeit an den unterschiedlichen Geschäftsprozessen der Polizeibehörden scheitern. So sind der PG IAM Vorhaben bekannt, bei denen bereits die Definition von Business-Rollen innerhalb einzelner Teilnehmer auf Grund zu starker Abweichungen in den Arbeitsweisen lokaler Behörden (z.B. Kreispolizeibehörden) nicht erfolgreich war. Dieselbe Problematik verstärkt sich zunehmend bei dem Versuch, teilnehmerübergreifend Business-Rollen zu definieren.

Dennoch sind auch positiv-Beispiele bekannt bei denen es Teilnehmern möglich war, Business-Rollen zu definieren – jedoch nicht im Sinne der Definition im Programm P20. Konkret wurden hier Anwendungsrollen umgesetzt, deren Berechtigungs-Muster sich jedoch an der fachlichen Arbeit der Mitarbeiter orientiert. Demnach handelt es sich zwar um Business-Rollen, jedoch nur für jeweils eine einzige Fachanwendung (VBS, AMS, EAS, etc.). Diese Rollendefinitionen wurden von Spezialisten der Fachanwendung sowie der Fachlichkeit selbst (Polizisten/Anwender) erarbeitet. Um nach demselben Vorbild bundeseinheitliche Business-Rollen zu definieren, wäre es erforderlich mit Spezialisten aller Fachanwendungen, aller Teilnehmer sowie den betroffenen Anwendern Tätigkeitsgebiete zu definieren und zu weitestgehend zu vereinheitlichen bzw. den gemeinsamen Kern mit einer teilnehmerübergreifenden (idealerweise bundesweiten) Ausprägung zu ermitteln.

Diese erfolgt nach zwei komplementär zueinanderstehenden Konzepten:

- Rollen, die der formale RBAC-Spezifikation, die als NIST-RBAC-Modell bekannt ist, entsprechen
- Rollen, oder besser Vorlagen, bzw. Richtlinien, die im Kontext einer Organisation zugewiesen, bzw. entzogen werden

Rollen nach RBAC-Spezifikation

Role-Based Access Control (RBAC) ist eine Methode zur Verwaltung des Benutzerzugriffs auf Systeme, Netzwerke, Ressourcen oder Anwendungen auf der Basis der jeweiligen Rolle innerhalb eines Teams oder einer größeren Organisation.

Bei RBAC werden Zugriffsrechte anhand eines definierten Rollenmodells vergeben. Die festgelegten Benutzerrollen abstrahieren die Arbeitsprozesse in einer Organisation und variieren daher von Organisation zu Organisation. Mögliche Anhaltspunkte für eine zweckdienliche Aufteilung sind Abteilungen, Standorte, Kostenstellen oder Funktionen eines Mitarbeiters.

Als Alternative zur Konfiguration des spezifischen System- oder Netzwerkzugriffs für einzelne Benutzer ermöglicht es RBAC der IT-Administration, die erforderliche Zugriffsebene für sämtliche Benutzer mit einer bestimmten Arbeitsfunktion zu identifizieren und diesen Benutzern eine Rolle mit den entsprechend konfigurierten Berechtigungen zuzuweisen. So wird das Prinzip, Berechtigungen für sämtliche Benutzer einer Gruppe auf einmal hinzufügen, zu ändern und zu entfernen, oder die Zugriffsebene einzelner Benutzer schnell zu ändern, vereinfacht.

Funktionsweise

Im Kern folgen RBAC-Systeme denselben Grundprinzipien:

- Einzelnen Benutzern werden eine oder mehrere Rollen zugewiesen.

- Nutzerrollen werden Berechtigungen zugewiesen.
- Benutzern erhalten Zugang zu Berechtigungen, wenn sie aktive Mitglieder einer Rolle sind.

In vielen Fällen legen RBAC-Modelle eine Rollenhierarchie fest. Dabei ähnelt die Rollenstruktur der Hierarchie der Organisation und kann Rollen für Administratoren, Endanwender und Gäste sowie sämtliche spezialisierten Gruppen dazwischen enthalten. Bei einigen Rollenhierarchien kann es sich um Vererbungshierarchien handeln, bei denen höherrangige Nutzerrollen automatisch die ihnen untergeordneten Rollen mitsamt ihren Berechtigungen erhalten. In anderen Fällen kann die Hierarchie willkürlich sein und Benutzer, denen eine übergeordnete Rolle zugewiesen wurde, erben nicht unbedingt standardmäßig die nachgeordneten Rollen.

Abhängig vom Anwendungsfall können Organisationen, die RBAC verwenden, auch eine Aufgabentrennung durchsetzen. Dies wird dadurch erreicht, dass die Beteiligung mehrerer Benutzer mit unterschiedlichen Rollen erforderlich ist, um eine bestimmte Aufgabe oder Aktion zu initiieren. Bei dieser Praxis werden die Rollenberechtigungen regelmäßig überprüft. So wird sichergestellt, dass Benutzer nur Berechtigungen haben, die sie auch tatsächlich benötigen.

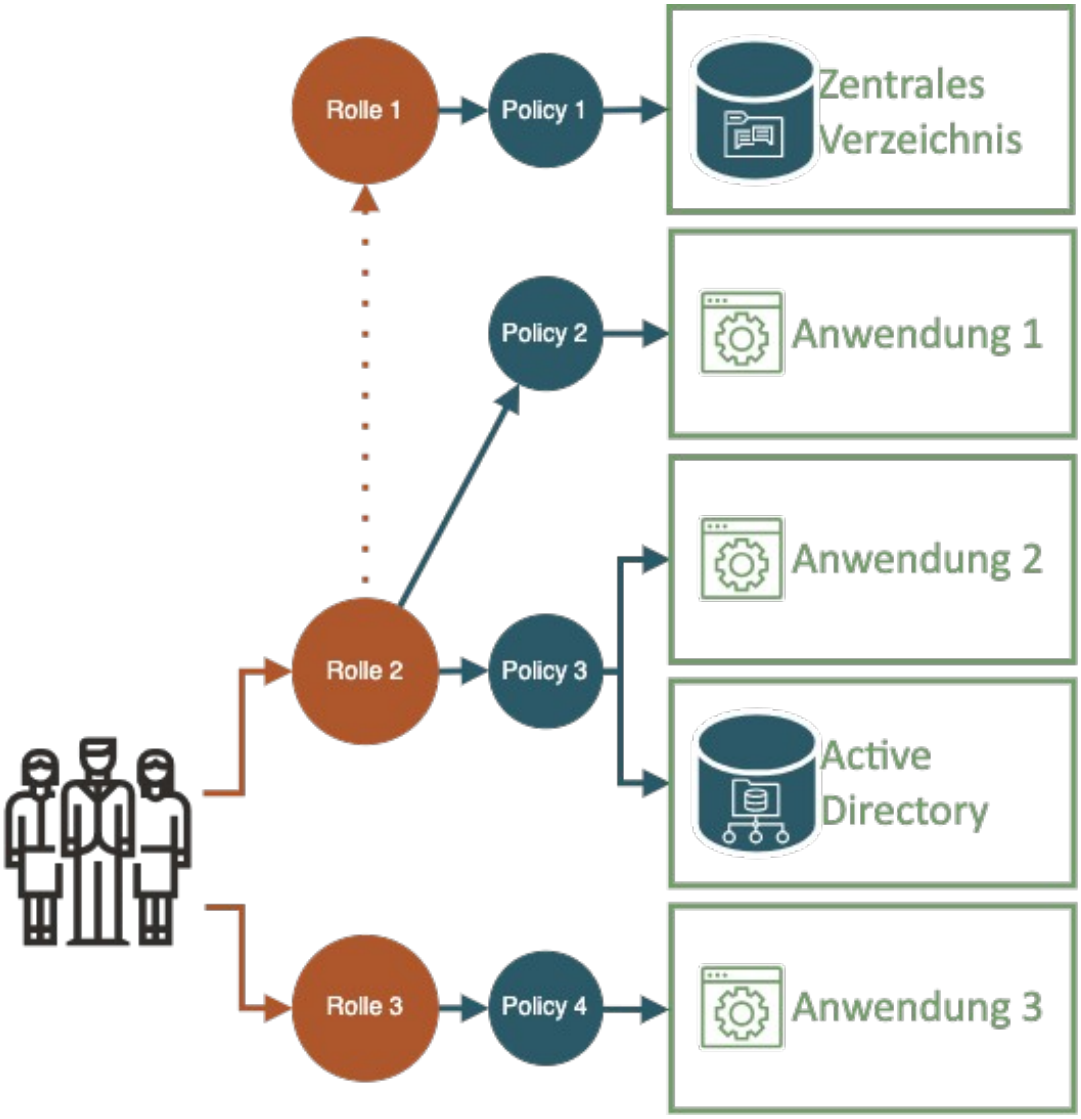

Abbildung 26: Grundprinzip Rollenmodellierung

Bei der Erstellung eines Rollenkonzeptes besteht in der PG IAM Konsens darin, eine hierarchische Rollenstruktur zu entwerfen. Konkret bedeutet das, dass Einzelrechte in einer Richtlinie (Policy) und diese in Business-Rollen zusammengefasst werden.

Einzelrechte sind nur innerhalb einer (Fach-)Anwendung gültig und bekannt. Die Zusammenfassung von Einzelrechten in Richtlinien erfolgt ebenfalls mit dem Ziel, die Vergabe und den Entzug von Berechtigungen an den Rollen die an der Funktion eines Mitarbeiters ausgerichtet sind zusammenzufassen bzw. den tatsächlichen administrativen Vorgang zu vereinfachen.

Schnittstellen für Teilnehmer

Anlegen einer neuen TN-Business-Rolle

Die Neuanlage einer TN-Business-Rolle im Basisdienst IAM erfolgt immer über die Nutzung einer der bereitgestellten Schnittstellen des Basisdienstes IAM. Prinzipiell legen Teilnehmer zunächst nur für den eigenen Geltungsbereich (Tenant) Rollen an, jedoch muss es möglich sein, eine entsprechende Berechtigung vorausgesetzt, auch für andere Teilnehmer Rollen anzulegen. Dies könnte z.B. der Fall sein, wenn ein Teilnehmer Teile seiner IT mit einem anderen Teilnehmer teilt (z.B. Zoll, BTPol, ITK). Aus diesem Grund müssen die unterschiedlichen Teilnehmer auf der Ebene der Schnittstelle unterscheidbar sein und ein wenigstens rudimentäres Berechtigungskonzept (Mandantenfähigkeit) für privilegierte Zugriffe wird benötigt.

Die Regeln und Vorgehensweisen zur Erstellung einer TN-Rolle sowie die Festlegung der berechtigten Personen zur Rollenanlage erfolgt in Hoheit und Verantwortung des jeweiligen Teilnehmers.

Eine TN-Rolle ist nur für den Teilnehmer sichtbar und zuweisbar, der sie angelegt hat. Sollten TN-Rollen zu einem späteren Zeitpunkt auch durch andere TN genutzt werden sollen, so wird fachlich eine gemeinsame Rollen-Eigentümerschaft definiert. Technisch wird jedoch jeder Teilnehmer weiterhin eine Rolle nutzen, die nur innerhalb des eigenen Mandanten Gültigkeit besitzt.

Für jede Rolle ist eine Eigentümerschaft zu definieren. Diese kann entweder bei einer natürlichen Person liegen, dem Teilnehmer allgemein oder eine Organisationseinheit des TN. Die konkrete Ausgestaltung der Rollen-Eigentümerschaft obliegt dem TN und ist abhängig von der TN-eigenen Herangehensweise bezüglich IAM (z. B. zentrale Benutzerverwaltung vs. dezentrale Benutzerverwaltung). Rollen-Eigentümer müssen Veränderungen und Löschungen der Rolle genehmigen. Eine Freigabe durch den Rollen-Eigentümer ist für die Zuweisung der Rolle an ein Benutzerkonto jedoch nicht erforderlich.

Rollen, die nach der RBAC-Spezifikation im Basisdienst IAM deklariert sind können über SCIMv2-Core und demzufolge auch über SCIMv2-Extended Benutzern zugewiesen werden.

Aktualisieren einer TN-Business-Rolle

Eine TN-Business-Rolle kann im Rahmen des Lebenszyklus der TN-Business-Rolle durch berechtigte Benutzer (z.B. Benutzerverwalter des TN) verändert werden.

Die möglichen Änderungen umfassen das Hinzufügen und Entfernen von Anwendungsfunktionsrechten sowie das Anpassen des Rollennamens und der Rollenbeschreibung.

Der Teilnehmer muss bei den Veränderungen an einer TN-Rolle die Auswirkungen auf den Benutzer im Umgang mit den einzelnen Anwendungen beachten. Das Änderungsmanagement

sollte die Auswirkungen auf die Nutzung von Anwendungen in den einzelnen unterstützten Geschäftsprozessen untersuchen. Die möglichen Auswirkungen sollten entsprechend an die Benutzer und die Benutzerunterstützungsfunktionen (Support/Helpdesk/Servicedesk beim TN) kommuniziert werden.

Den Teilnehmern muss bewusst sein, dass sich neben der eigenen Rolle, auch anwendungsseitig das Berechtigungskonzept ändern kann und einzelne Funktionsrechte, die in TN-Business-Rolle inkludiert sind, nicht mehr existieren. Es ist die Aufgabe des TN die eigene BV/IAM mit dem Basisdienst IAM synchron zu halten und Prozesse zu etablieren die in einem solchen Fall, möglichst automatisiert, anschlagen.

Löschen einer TN-Business-Rolle

Das Löschen einer TN-Rolle erfolgt durch den Eigentümer oder erfordert zumindest seine Zustimmung. Die TN-Rolle darf zum Zeitpunkt des Löschens keinem Benutzer mehr zugeordnet sein. Falls die Löschung nicht möglich ist, wird der Eigentümer über die Gründe für die fehlgeschlagene Löschung informiert.

Zuweisung einer TN-Business-Rolle

Das F-IAM protokolliert alle Änderungen an einer TN-Rolle, unabhängig von der genutzten Schnittstelle. Diese Änderungshistorie muss für berechtigte Benutzer einsehbar sein. Die Historie ist nicht veränderbar.

Rollen mit Organisationsbezug

Funktionsweise

Bevor dieses Prinzip der Vergabe und des Entzugs von „Rollen“ in einer Organisation umsetzbar ist, werden die Rechte in ähnlicher Weise wie bei der RBAC-Spezifikation spezifiziert. Der wesentliche Unterschied besteht darin, dass zum Zeitpunkt der Zuweisung die Organisationsebene als variabler Anteil spezifiziert wird und der Vergabealgorithmus diesen variablen Teil, dann auf das Berechtigungsmodell anwendet.

Schnittstellen für Teilnehmer

LDAP Generic

Zu beachten ist hierbei, dass die Rollen lediglich bezogen auf Anwendungen spezifiziert werden können.

SCIMv2-Extended

Es ist davon auszugehen, dass dieser Schnittstelle ein vollständig ausgeprägtes IAM-System vorgelagert ist. Die Rolle mit einem Bezug zu einer Organisation wird durch dieses IAM-System aufgelöst, sodass die dadurch evaluierten Berechtigungsinformationen zu den jeweiligen Benutzerkonten an den SCIM-Endpunkt des Basisdienst IAM übertragen werden. Der Basisdienst IAM übernimmt dann die Transformation der Berechtigungsinformationen in das anwendungsspezifische Berechtigungsmodell und die Übertragung dieser Daten in das Zielsystem. Hierzu kann der P20-Dienststellenschlüssel verwendet werden. Berechtigungen mit

Ebenenangaben und Vererbung sind in Abstimmung mit dem F-IAM-Team über „Scopes“ ebenfalls möglich. Technische Details finden sich in Confluence1.

Staging-Konzept

Um sowohl TN-IAM- als auch Anwendungsteams zu ermöglichen, ihre F-IAM-Anbindung in mehreren Stages zu entwickeln und zu testen, stellt das F-IAM folgende Stages bereit:

Stage	Serviceklasse	Stage der Laufzeit-umgebung	Zweck	Daten & Benutzerzugriffe
PreProd-DEV	Normal	Entwicklung (DEV)	Teilnehmern und Anwendungen einen dreistufigen Staging-Ansatz ermöglichen. (Dev → Int → Prod)	Keine Echtdaten (polizeifachliche Inhalte) erlaubt. Zugriffe grundsätzlich nur für personalisierte Accounts erlaubt. Übergangsweise Lösungen mit nicht personalisierten Accounts aber in Abstimmung mit IAM-Team möglich.
PreProd-SCHUL (EDU)	IDM: Normal AM: Hoch	Entwicklung (DEV)	Vergleichbar mit Integrationsumgebung, allerdings mit höheren Anforderungen an Verfügbarkeit. Auch als Referenz-umgebung geeignet.	Keine Echtdaten (polizeifachliche Inhalte) erlaubt. Zugriffe für personalisierte und nicht personalisierte Accounts erlaubt.
Integrations-umgebung (INT)	Normal	Produktion (PROD)	Teilnehmern und Anwendungen einen dreistufigen Staging-Ansatz ermöglichen. (Dev → Int → Prod)	Echtdaten (polizeifachliche Inhalte) erlaubt. Nur personalisierte Accounts erlaubt.
Wirkumgebung (WIRK, PROD)	IDM: Normal AM: Sehr Hoch	Produktion (PROD)	Anbindung produktiv genutzter Teilnehmer-Benutzerverwaltungen sowie produktiv genutzter Anwendungen.	Echtdaten (polizeifachliche Inhalte) erlaubt. Nur personalisierte Accounts erlaubt.

Tabelle 5: F-IAM Stages

Alle Umgebungen befinden sich bereits im Wirkbetrieb.

Im Laufe von 2024 ist geplant, in der DEV als zusätzliches Angebot die Möglichkeit bereitzustellen, als TN-IAM-Team eine angebundene Anwendung zu simulieren – und umgekehrt als Anwendungsteam einen angebundenen TN zu simulieren. So sollen die Teams bei ihrer

Entwicklung unabhängig von einer Kooperation sein und jederzeit unabhängig voneinander testen können.

Anhang

Sequenzdiagramme zur Anmeldung an Anwendungen

Webanwendung (SAML2)

Login eines Benutzers bei einer Webanwendung mittels SAML2, wobei das SAML2 HTTP-POST Binding genutzt wird.

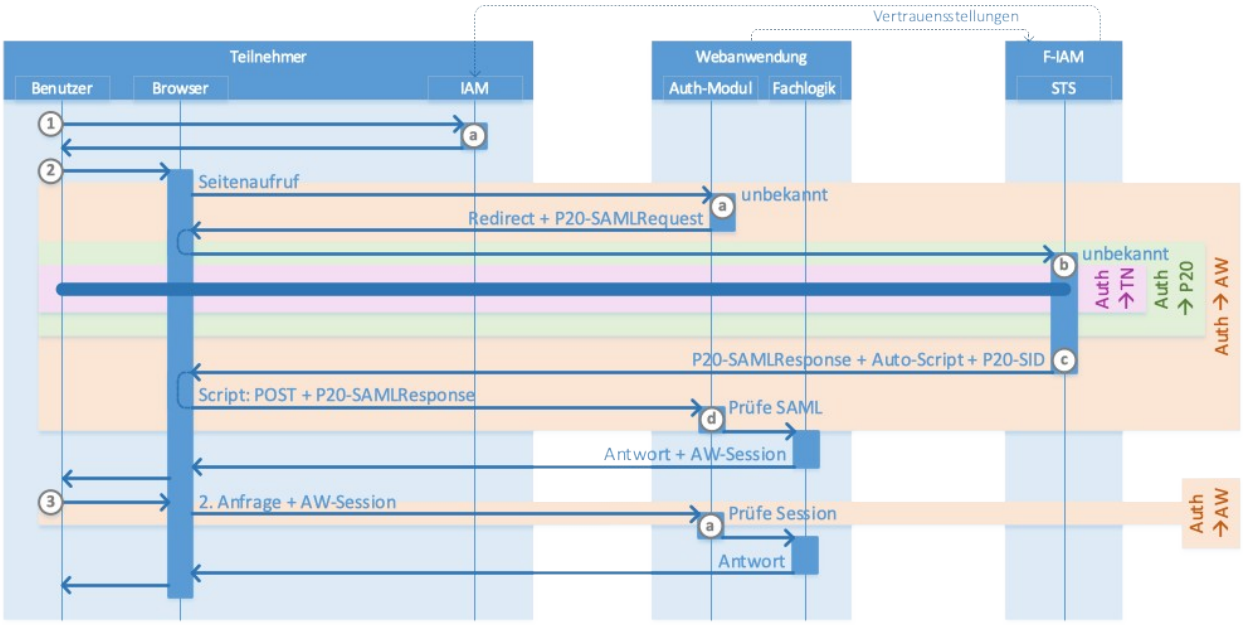

Abbildung 27: Sequenzdiagramm zur Authentifizierung per SAML2

Ablauf

1. **Login an Teilnehmer-Domäne**

   Der Benutzer meldet sich über sein Endgerät am TN-IAM an.

   a. Das TN-IAM prüft die Zugangsdaten.

2. **Seitenaufruf**

   Der Benutzer ruft mit dem Browser die Webanwendung auf.

   a. Das Auth-Modul der Webanwendung erkennt den anfragenden Benutzer nicht und startet daraufhin eine SAML2-Authentifizierung, indem es mit einem Redirect zum zentralen Security Token Service (P20-STS) antwortet und als Query-Parameter u.a. ein P20-SAMLRequest übergibt.

   b. Der aufgerufene P20-STS kennt den anfragenden Benutzer bisher nicht und veranlasst eine Authentifizierung seitens des Teilnehmer-IAMs gemäß der jeweiligen Teilnehmer-spezifischen Konfiguration.

   c. Nach Abschluss der Authentifizierung über das Teilnehmer-IAM liegt dem P20-STS die Kennung des Benutzers vor.

   Für den nun in der Webanwendung angemeldeten Benutzer werden die weiteren angefragten Attribute und Berechtigungen für die anfragende

Webanwendung ermittelt. Diese werden in der P20-SAMLResponse als Antwort gesendet. Dabei wird auch eine Session-ID als Cookie gesetzt, um beim Aufruf einer weiteren Anwendung (in diesem Sequenzdiagramm nicht enthalten) die erneute explizite Authentifizierung gegenüber dem F-IAM zu sparen.

Zusätzlich ist ein Auto-Script in der Antwort enthalten, das den Browser dazu veranlasst, automatisch ein POST mit dem P20-SAMLResponse an die Webanwendung zu senden.

d. Das Auth-Modul prüft die P20-SAMLResponse und liest die Benutzerattribute und Berechtigungen aus. Diese werden beim Weiterleiten an die Fachlogik übergeben.

Die Fachlogik sendet die Antwort. Dabei werden auch Sessioninformationen (z.B. als Session-ID-Cookie oder JWT) mitgeschickt.

3. Erneute Anfrage

Der Benutzer interagiert über den Browser mit der Webanwendung, so dass eine erneute Anfrage erzeugt wird, die auch die Session-Informationen enthält.

a. Das Auth-Modul erkennt die Session-Informationen. Es ermittelt die zugehörigen Benutzerattribute und Berechtigungen (z.B. aus der lokal gespeicherten Session oder dem JWT) und übergibt diese an die Fachlogik.

Die Fachlogik generiert die Antwort an den Browser.

Webanwendung (OIDC)

Login eines Benutzers bei einer Webanwendung mittels OIDC, wobei konkret der „Authorization Code Flow“ mit Client-Secret und PKCE (Proof Key for Code Exchange) verwendet wird.

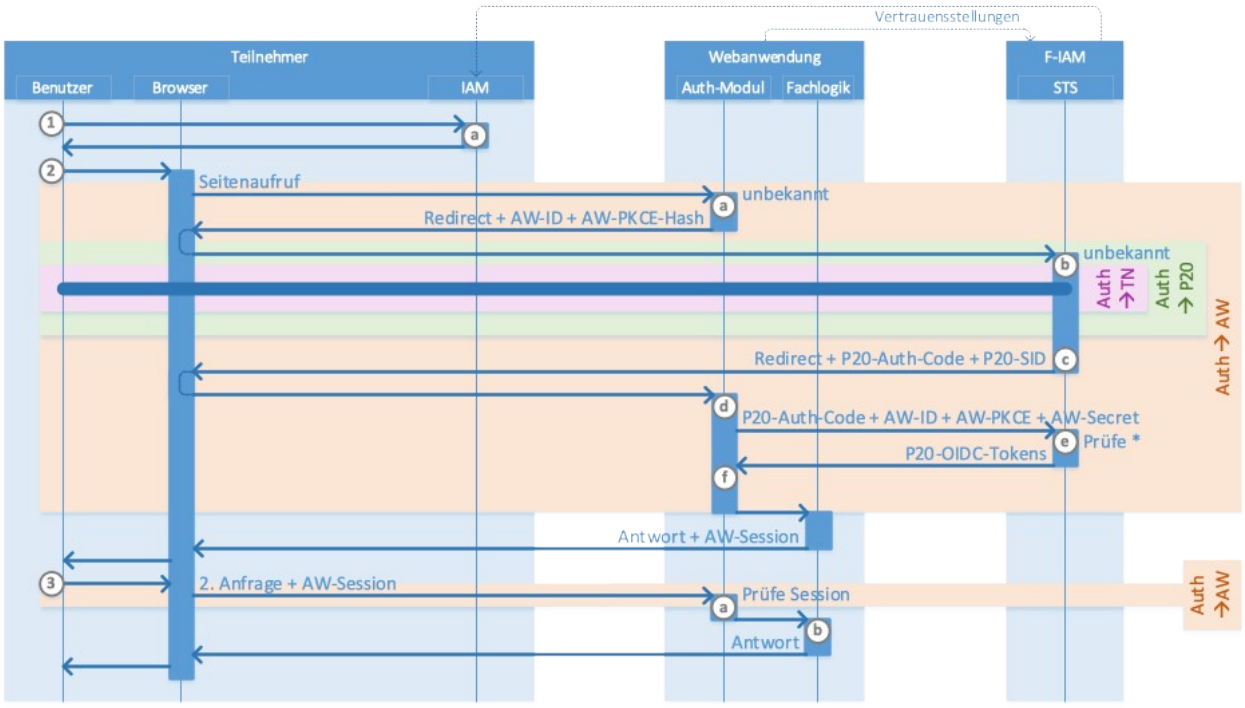

Abbildung 28: Sequenzdiagramm zur Authentifizierung per Open ID Connect

Ablauf

1. **Login an Teilnehmer-Domäne**

   Der Benutzer meldet sich über sein Endgerät am Teilnehmer-IAM an.

   a. Das TN-IAM prüft die Zugangsdaten.

2. **Seitenaufruf**

   Der Benutzer ruft im Browser die Webanwendung auf.

   a. Das Auth-Modul erkennt den anfragenden Benutzer nicht und startet daraufhin eine OIDC-Authentifizierung, indem es mit einem Redirect zum zentralen Security Token Service (P20-STS) antwortet und als Query-Parameter u.a. sowohl die eigene Anwendungs-ID (OIDC-Client-ID) als auch den Hash eines selbst generierten PKCE übergibt.

   b. Der aufgerufene P20-STS kennt den anfragenden Benutzer bisher nicht und veranlasst eine Authentifizierung seitens des Teilnehmer-IAMs gemäß der jeweiligen Teilnehmer-spezifischen Konfiguration.

   c. Nach Abschluss der Authentifizierung über das Teilnehmer-IAM liegt dem P20-STS die P20-UID des Benutzers vor.

      Als Antwort wird ein Redirect zu der Webanwendung zusammen mit einem neu generierten Autorisierungscode gesendet. Dabei wird auch eine Session-ID als Cookie gesetzt, um beim Aufruf einer weiteren Anwendung (in diesem Sequenzdiagramm nicht enthalten) die erneute explizite Authentifizierung gegenüber dem F-IAM zu sparen.

   d. Das Auth-Modul der Webanwendung verwendet den Autorisierungscode und sendet eine Tokenanfrage an den P20-STS zusammen mit der eigenen Anwendungs-ID, dem Klartext-PKCE, dessen Hashwert in 2.a übermittelt wurde, sowie dem eigenen Anwendungs-Secret.

   e. Der P20-STS prüft die übergebenen Parameter.

      Anschließend ermittelt er für die zum Autorisierungscode gehörige Kennung und die zugehörigen Benutzerattribute und Berechtigungen. Er antwortet abschließend mit den OIDC-Tokens.

   f. Das Auth-Modul der Webanwendung extrahiert die Benutzerattribute und Berechtigungen.

      Diese werden beim Weiterleiten an die Fachlogik übergeben.

      Die Fachlogik sendet die Antwort und gibt dabei auch Sessioninformationen (z.B. als Session-ID-Cookie oder JWT) mit.

3. **Erneute Anfrage**

   Der Benutzer interagiert über den Browser mit der Webanwendung, so dass eine

erneute Anfrage erzeugt wird. Dabei werden vom Browser auch die Session-Informationen übergeben.

a. Das Auth-Modul erkennt die Session-Informationen. Es ermittelt die zugehörigen Benutzerattribute und Berechtigungen (z.B. aus der lokal gespeicherten Session oder dem JWT) und übergibt diese an die Fachlogik.

b. Die Fachlogik generiert die Antwort an den Browser.

SSO zwischen P20-Webanwendungen

Ist ein Benutzer über seinen Browser bereits an einer P20-Webanwendung angemeldet, so soll er ohne erneute explizite Authentifizierung auch auf andere P20-Webanwendungen zugreifen können. Dies ist insbesondere dann für die Benutzerfreundlichkeit relevant, wenn aus einer Webanwendung durch Links auf andere Webanwendungen verwiesen wird.

Die technische Lösung liegt in der Vergabe von Session-IDs als Cookies durch den P20-STS, wobei dies unabhängig vom jeweiligen Authentifizierungsverfahren (OIDC und/oder SAML2) funktioniert.

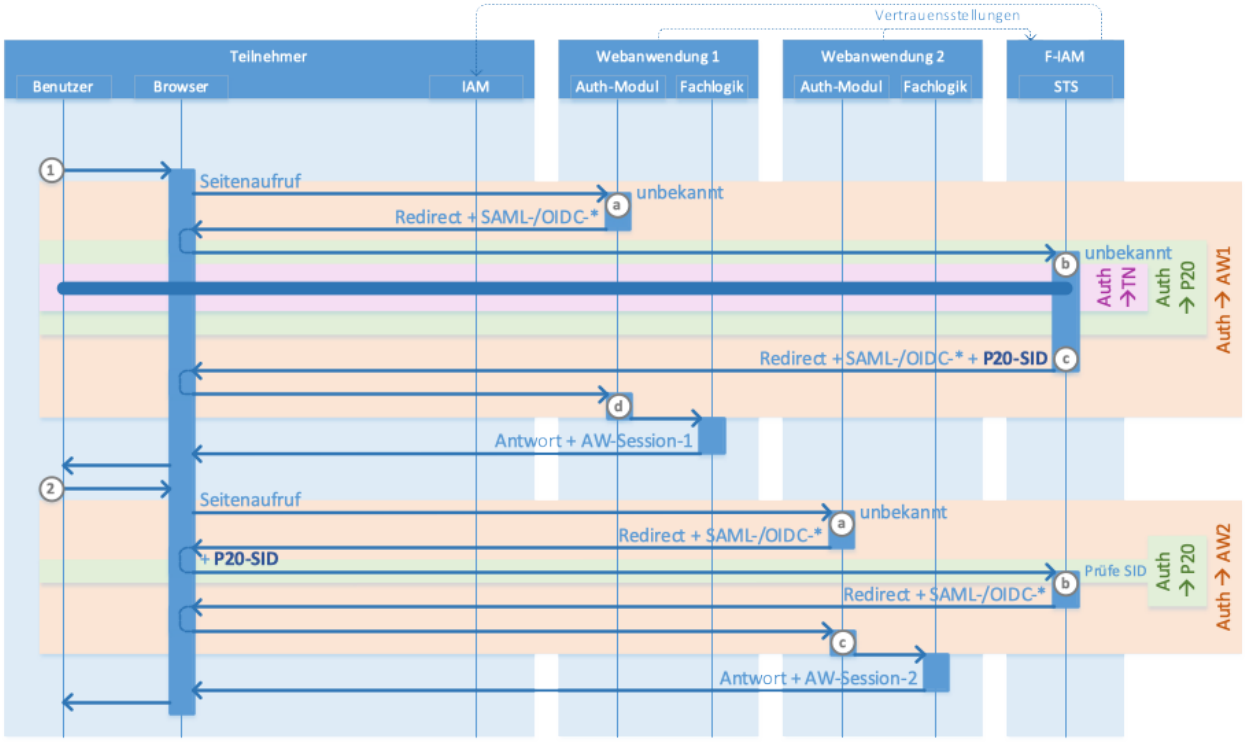

Abbildung 29: Sequenzdiagramm zum SSO zwischen P20-Webanwendungen

Ablauf:

1. **Seitenaufruf Webanwendung 1**

   Der Benutzer ruft im Browser die Webanwendung 1 auf.

   a. Das Auth-Modul erkennt den anfragenden Benutzer nicht und startet daraufhin eine SAML2- oder OIDC-Authentifizierung, die in beiden Fällen zu einem Redirect zum zentralen Security Token Service (P20-STS) als Antwort führt.

b. Der über den Redirect vom Browser aufgerufene P20-STS erkennt den anfragenden Benutzer nicht und veranlasst eine Authentifizierung seitens des Teilnehmer-IAMs gemäß der jeweiligen Teilnehmer-spezifischen Konfiguration.

c. Nach Abschluss der Authentifizierung über das Teilnehmer-IAM setzt der P20-STS die SAML2- oder OIDC-Authentifizierung fort, die in beiden Fällen zu einem Redirect zurück zur Webanwendung als Antwort führt. Hierin wird durch den P20-STS auch eine Session-ID als Cookie gesetzt.

d. Das Auth-Modul setzt die Authentifizierung fort und leitet nach Abschluss an die Fachlogik weiter, die dann die Antwort sendet.

2. **Seitenaufruf Webanwendung 2**

   Der Benutzer ruft in demselben Browser die Webanwendung 2 auf.

   e. Das Auth-Modul erkennt den anfragenden Benutzer nicht und startet daraufhin eine SAML2- oder OIDC-Authentifizierung, die in beiden Fällen zu einem Redirect zum zentralen P20-STS als Antwort führt. Beim Folgen des Redirects übergibt der Browser die im Schritt 1.c als Cookie gesetzte P20-Session-ID.

   f. Der P20-STS erkennt den anfragenden Benutzer über die Session-ID und setzt direkt die SAML2- oder OIDC-Authentifizierung gegenüber der Webanwendung fort, die in beiden Fällen zu einem Redirect zurück zur Webanwendung als Antwort führt, ohne dass es einer weiteren Interaktion mit dem Teilnehmer-IAM oder dem Benutzer bedarf.

   g. Das Auth-Modul setzt die sie Authentifizierung fort und leitet nach Abschluss an die Fachlogik weiter, die dann die Antwort sendet.

Rich-Client mit Browser-Control (OIDC)

Login eines Benutzers bei einem Rich-Client (eine auf einem Endgerät installierte Anwendung), der einen Webservice aufruft und dabei die Autorisierung des Benutzers benötigt. Hierfür wird von OIDC der „Authorization Code Flow“ mit PKCE (Proof Key for Code Exchange) über ein Browser-Control genutzt.

Ablauf:

1. **Login an Teilnehmer-Domäne**

   Benutzer meldet sich über sein Endgerät am Teilnehmer-IAM an.

   a. Das TN-IAM prüft die Zugangsdaten.

2. **Login an Anwendung**

   Der Benutzer ruft den lokalen Rich-Client auf.

   a. Der Client verwendet ein Browser-Control und übergibt u.a. die URI des zentralen Security Token Services (P20-STS) zusammen mit der eigenen Anwendungs-ID (OIDC-Client-ID) und dem Hash eines selbst generierten PKCE als Query-Parameter.

b. Der aufgerufene P20-STS kennt den anfragenden Benutzer bisher nicht und veranlasst eine Authentifizierung seitens des Teilnehmer-IAMs gemäß der jeweiligen Teilnehmer-spezifischen Konfiguration.

c. Nach Abschluss der Authentifizierung über das Teilnehmer-IAM liegt dem P20-STS die Kennung des Benutzers vor.

   Als Antwort wird ein Redirect zu der Anwendung zusammen mit einem neu generierten Autorisierungscode gesendet. Dabei wird auch eine Session-ID als Cookie gesetzt, um beim Aufruf einer weiteren Anwendung (in diesem Sequenzdiagramm nicht enthalten) die erneute explizite Authentifizierung gegenüber dem F-IAM zu sparen.

d. Statt dem Redirect zu folgen, sendet der Client eine Authentifizierungsanfrage an den P20-STS u.a. zusammen mit dem Autorisierungscode, der eigenen Anwendungs-ID (OIDC-Client-ID) sowie dem Klartext-PKCE, dessen Hashwert in 2.a übermittelt wurde.

e. Der P20-STS prüft die übergebenen Parameter.

   Anschließend ermittelt er für die zum Autorisierungscode gehörige Kennung und die zugehörigen Benutzerattribute und Berechtigungen. Er antwortet abschließend mit den OIDC-Tokens.

f. Der Client extrahiert die Benutzerattribute und Berechtigungen aus den Tokens. Ein explizites Prüfen der Tokens ist nicht nötig, da sie über eine direkte Kommunikation mit dem P20-STS empfangen wurden. Der Access- sowie das Refresh-Token werden für eine spätere Nutzung gespeichert.

   Das Access-Token wird nun von dem Client verwendet, um sich gegenüber dem eigenen Webservice zu authentifizieren.

g. Das Auth-Modul des Webservices prüft das Access-Token (gemäß Vertrauensstellung gegenüber dem F-IAM). Es extrahiert die Berechtigungszuweisungen und leitet die Anfrage an die Fachlogik weiter, die eine Antwort an den Client sendet.

h. Damit ist die Anmeldung des Benutzers über den Client am Webservice abgeschlossen. Das Access-Token wird auch bei allen weiteren Anfragen mitgesendet, so dass der Bedarf gesonderter Session-Informationen wie bei Webanwendungen entfällt.

3. **Nutzung nach Ablauf des Access-Tokens**

   a. Ist bei der weiteren Benutzung das Access-Token abgelaufen, verwendet der Client das zuvor gespeicherte Refresh-Token zusammen mit der eigenen Anwendungs-ID, um beim P20-STS jeweils ein neues ID- und Access-Token anzufragen.

b. Der P20-STS prüft das Refresh-Token und die Zugehörigkeit zur Anwendungs-ID und antwortet mit den entsprechenden neuen OIDC-Tokens. Hier sind nun auch die ggf. zwischenzeitlich geänderten Berechtigungen und Benutzerattribute enthalten.

Anmerkung: Schlägt die Prüfung des Refresh-Tokens am P20-STS fehl, so wird der Authentifizierungsvorgang neu gestartet.

c. Der Client fährt fort wie nach dem erstmaligen Empfangen der Tokens.

Mobile-App mit Webservice (OIDC)

Login eines Benutzers bei einer Mobile-App, die ihrerseits einen Webservice aufruft und dabei die Autorisierung des Benutzers benötigt. Es wird OIDC genutzt, wobei konkret der „Authorization Code Flow“ für Mobile mit PKCE (Proof Key for Code Exchange) zum Einsatz kommt. Der Ablauf entspricht im Wesentlichem dem des Rich-Clients mit Browser-Control (siehe Rich-Client mit Browser-Control (OIDC)), wobei hier allerdings der in-App-Browser und der Secure-Store für das Speichern des Refresh-Tokens zum Einsatz kommen.

Gesamtablauf AW-OIDC mit TN-SAML2 und Kerberos

Login eines Benutzers bei einer Webanwendung mittels OIDC für den Fall, dass der Teilnehmer per SAML2 am F-IAM angebunden ist und innerhalb des Teilnehmers Kerberos verwendet wird.

Dieses Szenario ergibt sich aus der Kombination der entsprechenden Sequenzdiagramme der einzelnen Authentifizierungsstufen. Da es aber den häufigsten und damit wichtigsten Fall darstellt, ist es hier nochmal explizit beschrieben.

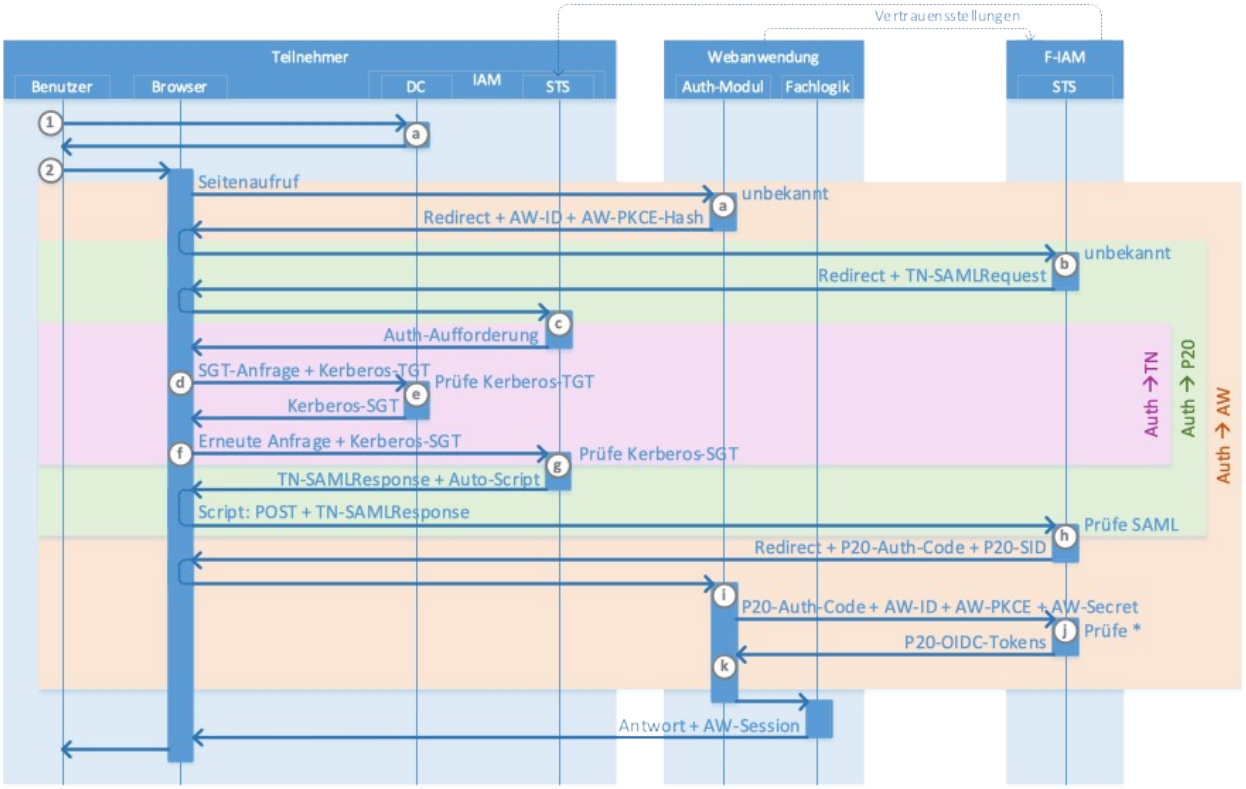

Abbildung 30: Sequenzdiagramm zum Gesamtablauf AW-OIDC mit TN-SAML2 und Kerberos

Ablauf

1. **Login an Teilnehmer-Domäne**

   Der Benutzer meldet sich über sein Endgerät am Domain Controller seines TN-IAM an.

   a. Das TN-IAM prüft die Zugangsdaten.

2. **Seitenaufruf**

   Der Benutzer ruft mit dem Browser die Webanwendung auf.

   a. Das Auth-Modul erkennt den anfragenden Benutzer nicht und startet daraufhin eine OIDC-Authentifizierung, indem es mit einem Redirect zum zentralen Security Token Service (P20-STS) antwortet und als Query-Parameter u.a. sowohl die eigene Anwendungs-ID (OIDC-Client-ID) als auch den Hash eines selbst generierten PKCE übergibt.

   b. Der P20-STS erkennt den anfragenden Benutzer nicht und startet daraufhin eine SAML2-Authentifizierung, indem es mit einem Redirect zum Teilnehmer Security Token Service (TN-STS) antwortet und als Query-Parameter u.a. ein TN-SAMLRequest übergibt.

   c. Der TN-STS erkennt den Benutzer nicht und antwortet mit einer Aufforderung zur Authentifizierung per SPNEGO.

   d. Der Browser fordert über das Betriebssystem vom Domain Controller ein Kerberos Service Granting Ticket (SGT) an, wobei das Token Granting Ticket (TGT) mit übergeben wird, das beim Anmelden des Benutzers an der Teilnehmer-Domäne 1.a empfangen wurde.

   e. Der Domain Controller prüft das TGT und Antwortet mit einem SGT für den TN-STS.

   f. Der Browser sendet seine ursprüngliche Anforderung erneut, wobei er sich nun über das SGT authentifiziert.

   g. Der TN-STS prüft den SGT und fährt mit der Authentifizierung gegenüber dem F-IAM fort, indem als Antwort ein Redirect zum P20-STS zusammen mit einem TN-SAMLResponse gesendet wird, das lediglich die Kennung des Benutzers enthält.

      Zusätzlich ist ein Auto-Script in der Antwort enthalten, das den Browser dazu veranlasst, automatisch ein POST mit dem TN-SAMLResponse an den P20-STS zu senden.

   h. Der P20-STS prüft die TN-SAMLResponse und liest die Kennung aus.

      Als Antwort wird ein Redirect zu der Webanwendung zusammen mit einem neu generierten Autorisierungscode gesendet. Dabei wird auch eine Session-ID als Cookie gesetzt, um beim Aufruf einer weiteren Anwendung (in diesem Sequenzdiagramm nicht enthalten) die erneute explizite Authentifizierung gegenüber dem F-IAM zu sparen.

i. Das Auth-Modul der Webanwendung verwendet den Autorisierungscode und sendet eine Tokenanfrage an den P20-STS zusammen mit der eigenen Anwendungs-ID, dem Klartext-PKCE, dessen Hashwert in 2.a übermittelt wurde, sowie dem eigenen Anwendungs-Secret.

j. Der P20-STS prüft die übergebenen Parameter.

   Anschließend ermittelt er für die zum Autorisierungscode gehörige Kennung und die zugehörigen Benutzerattribute und Berechtigungen. Er antwortet abschließend mit den OIDC-Tokens.

k. Das Auth-Modul der Webanwendung extrahiert die Benutzerattribute und Berechtigungen.

   Diese werden beim Weiterleiten an die Fachlogik übergeben.

   Die Fachlogik sendet die Antwort und gibt dabei auch Sessioninformationen (z.B. als Session-ID-Cookie oder JWT) mit.

Sequenzdiagramme zur delegierten Authentifizierung gegenüber dem F-IAM

Beim Login an einer P20-Anwendung wird diese die Authentifizierung an das F-IAM delegieren, was in Form eines Redirects erfolgt. Ist im Rahmen eines solchen Login-Vorgangs der Benutzer auch dem F-IAM noch nicht bekannt (Session-ID liegt noch nicht vor oder ist abgelaufen), so veranlasst es eine Authentifizierung mit dem Teilnehmer-IAM.

Das Identifizieren des zuständigen TN-IAMs durch das F-IAM kann unter Verwendung verschiedener Methoden erfolgen (In den Sequenzdiagrammen nicht dargestellt):

- Die Kennung des Teilnehmers wurde explizit als Query-Parameter übergeben. (Insbesondere bei Rich-Clients sinnvoll, die bei einem konkreten Teilnehmer installiert sind.)
- Das F-IAM antwortet zunächst mit einer HTML-Seite, auf der der Benutzer seinen Teilnehmer (Identity Provider) auswählen kann.

Den im Folgenden dargestellten Abläufen gehen jeweils immer bereits ein Redirect von einer P20-Anwendung zum P20-IAM voraus.

Delegierte Authentifizierung per SAML2

Vom F-IAM an den Teilnehmer delegierte Authentifizierung per SAML2, wobei das SAML2 http-POST Binding genutzt wird. Im TN-SAMLResponse wird im Wesentlichen die Kennung des Benutzers übertragen.

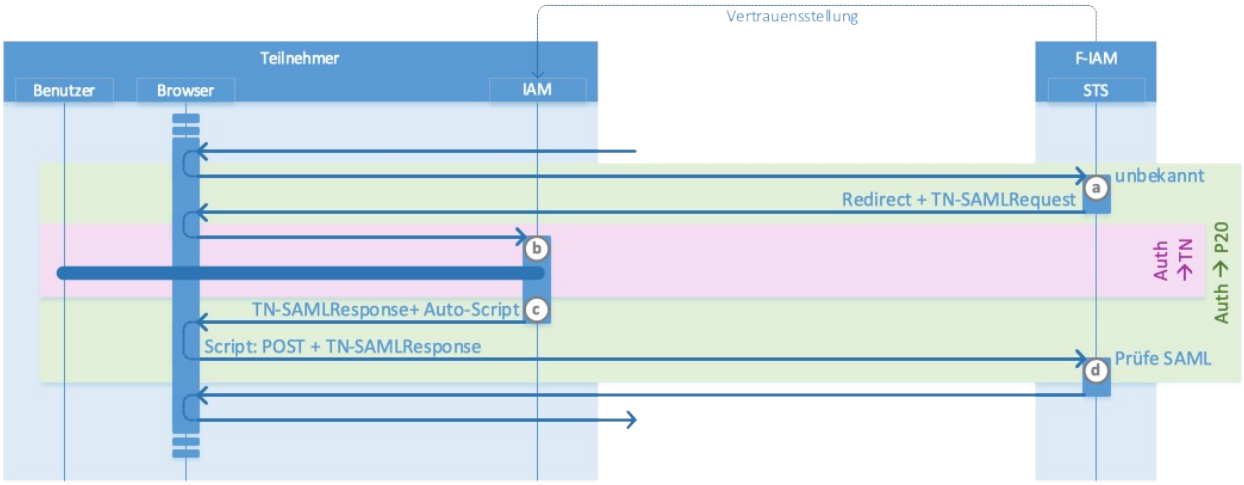

Abbildung 31: Sequenzdiagramm zur delegierte Authentifizierung per SAML2

Ablauf

Von einer Anwendung erfolgt ein Redirect an den P20-STS, weil der Benutzer dort noch nicht authentifiziert ist.

a. Der P20-STS erkennt den anfragenden Benutzer nicht und startet daraufhin eine SAML2-Authentifizierung, indem es mit einem Redirect zum Teilnehmer Security Token Service (TN-STS) antwortet und als Query-Parameter u.a. ein TN-SAMLRequest übergibt.

b. Der aufgerufene TN-STS kennt den anfragenden Benutzer bisher nicht und veranlasst die Authentifi-zierung innerhalb der Teilnehmer-Domäne (siehe 5.3. Sequenzdiagramme zur Authentifizierung gegenüber dem Teilnehmer-IAM).

c. Nach Abschluss der Authentifizierung des Benutzers wird als Antwort ein Redirect zum P20-STS zusammen mit einem TN-SAMLResponse gesendet, das lediglich die Kennung des Benutzers enthält.

d. Zusätzlich ist ein Auto-Script in der Antwort enthalten, das den Browser dazu veranlasst, automatisch ein POST mit dem TN-SAMLResponse an den P20-STS zu senden.

e. Der P20-STS prüft die TN-SAMLResponse und liest die Kennung aus. Anschließend fährt er mit der Authentifizierung gegenüber der Anwendung fort.

Delegierte Authentifizierung per OIDC

Vom F-IAM an den Teilnehmer delegierte Authentifizierung per OIDC, wobei konkret der „Authorization Code Flow“ mit Client-Secret und PKCE (Proof Key for Code Exchange) verwendet wird. Es wird hier allerdings lediglich ein ID-Token benötigt, und darin wird lediglich die Kennung des Benutzers übertragen.

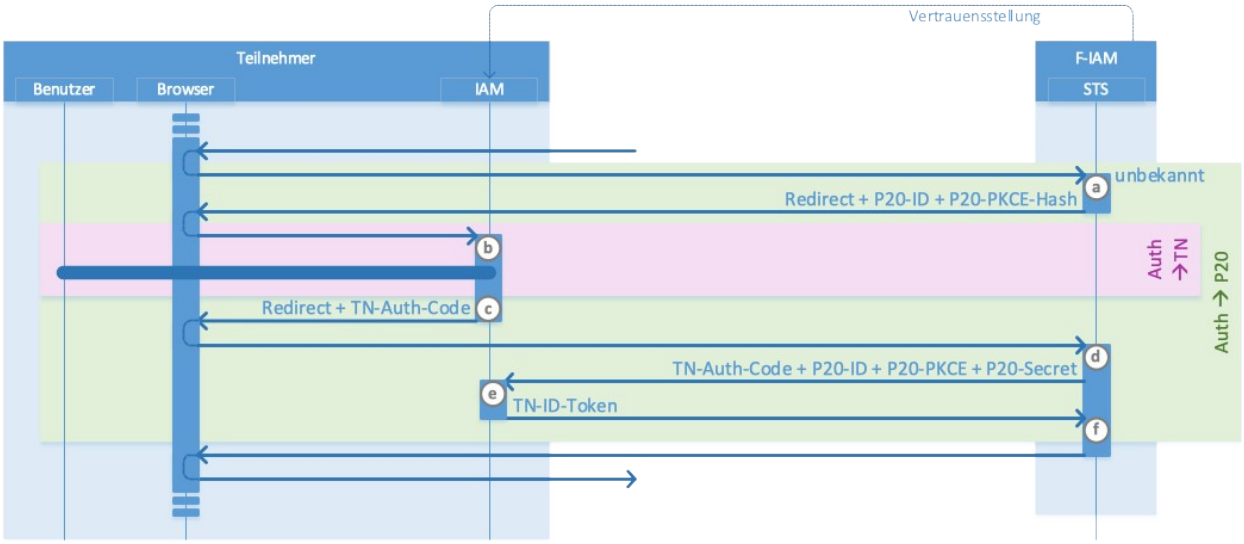

Abbildung 32: Sequenzdiagramm zur delegierte Authentifizierung per OIDC

Ablauf

Von einer Anwendung erfolgt ein Redirect an den P20-STS, weil der Benutzer dort noch nicht authentifiziert ist.

a. Der P20-STS erkennt den anfragenden Benutzer nicht und startet daraufhin eine OIDC-Authentifizie-rung, indem es mit einem Redirect zum Teilnehmer Security Token Service (TN-STS) antwortet und als Query-Parameter u.a. sowohl die eigene Anwendungs-ID (OIDC-Client-ID) als auch den Hash eines selbst generierten PKCE übergibt.

b. Der aufgerufene TN-STS kennt den anfragenden Benutzer bisher nicht und veranlasst die Authentifizierung innerhalb der Teilnehmer-Domäne (siehe Sequenzdiagramme zur Authentifizierung gegenüber dem Teilnehmer-IAM).

c. Nach Abschluss der Authentifizierung des Benutzers wird als Antwort ein Redirect zum P20-STS zusammen mit einem neu generierten Autorisierungscode gesendet.

d. Der P20-STS verwendet den Autorisierungscode und sendet eine Tokenanfrage an den TN-STS zusammen mit der eigenen Anwendungs-ID, dem Klartext-PKCE, dessen Hashwert in Schritt a übermittelt wurde, sowie dem eigenen Anwendungs-Secret.

e. Der TN-STS prüft die übergebenen Parameter.

Anschließend antwortet er mit einem ID-Token, das die Kennung des Benutzers enthält.

f. Der P20-STS fährt mit der Authentifizierung gegenüber der Anwendung fort.

Sequenzdiagramme zur Authentifizierung gegenüber dem Teilnehmer-IAM

Die Authentifizierung innerhalb des Teilnehmers liegt komplett in der Verantwortung des Teilnehmers. Die hier aufgeführten Sequenzdiagramme dienen lediglich zur Veranschaulichung, wie sich die verschiedenen Verfahren in den Gesamtprozess einfügen.

Den im Folgenden dargestellten Abläufen gehen jeweils immer bereits ein Redirect von einer P20-Anwendung zum P20-IAM und von dort ein weiterer Redirect zum TN-IAM voraus. Das TN-IAM muss als nächstes also den Benutzer innerhalb der Teilnehmer-Domäne authentifizieren.

TN-interne Authentifizierung per Kerberos

Die Authentifizierung per Kerberos innerhalb der Teilnehmer-Domäne bietet den maximalen Benutzerkomfort, da es vollständig im Hintergrund abläuft und so ein SSO mit der initialen Anmeldung des Benutzers am TN-IAM (hier nicht abgebildet) realisiert. Konkret erfolgte diese Anmeldung am Domain Controller (DC), der auch die notwendigen Kerberos-Dienste bereitstellt. Der STS ist Mitglied derselben Active Directory Domäne, so dass hier die Funktion der Integrierten Windows Authentifizierung (IWA) greift.

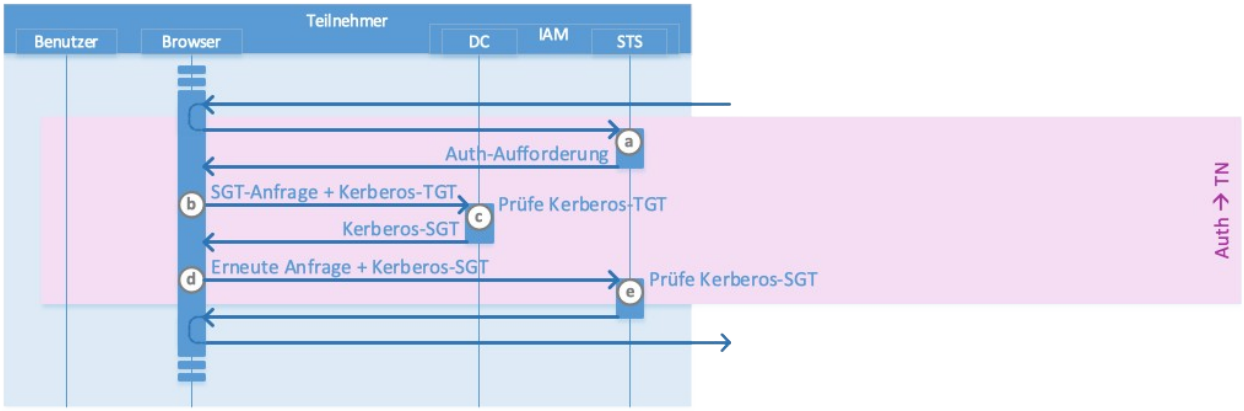

Abbildung 33: Sequenzdiagramm zur TN-interne Authentifizierung per Kerberos

Ablauf

a. Der TN-STS erkennt den Benutzer nicht und antwortet mit einer Aufforderung zur Authentifizierung per SPNEGO.

b. Der Browser fordert über das Betriebssystem vom Domain Controller ein Kerberos Service Granting Ticket (SGT) an, wobei das Token Granting Ticket (TGT) mit übergeben wird, das beim Anmelden des Benutzers an der Teilnehmer-Domäne (im Sequenzdiagramm nicht enthalten) empfangen wurde.

c. Der Domain Controller prüft das TGT und Antwortet mit einem SGT für den TN-STS.

d. Der Browser sendet seine ursprüngliche Anforderung erneut, wobei er sich nun über das SGT authentifiziert.

e. Der TN-STS prüft den SGT und fährt mit der Authentifizierung gegenüber dem F-IAM fort.

TN-interne Authentifizierung per Benutzerkennung und Passwort

Alternativ zu Kerberos kann das TN-IAM auch ein explizites Login bspw. über Benutzerkennung und Passwort am zu verwendenden STS (IdP) verlangen.

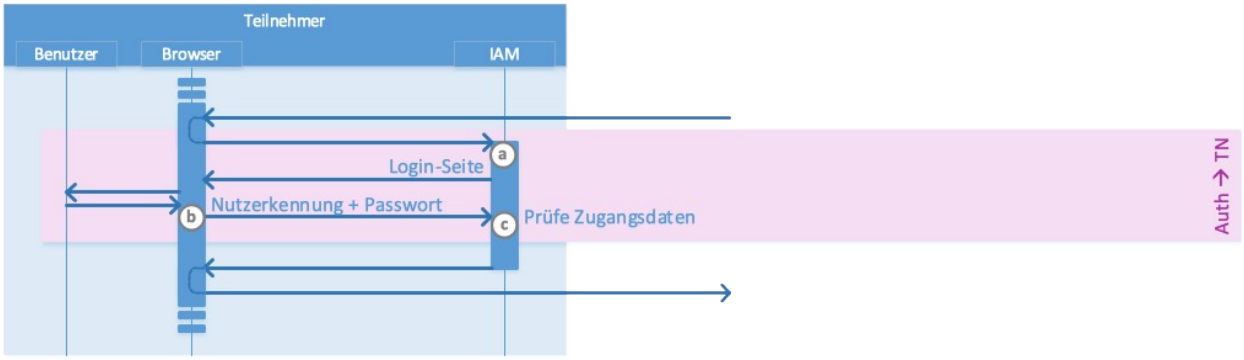

Abbildung 34: Sequenzdiagramm zur TN-interne Authentifizierung per Benutzerkennung und Passwort

Ablauf

a. Der TN-STS erkennt den Benutzer nicht und antwortet mit einem Formular zur Eingabe von Zugangsdaten.

b. Die eingegebenen Zugangsdaten werden an den TN-STS übermittelt.

c. Der TN-STS prüft die übermittelten Zugangsdaten und fährt mit der Authentifizierung gegenüber dem F-IAM fort.

Weitere Sequenzdiagramme

Den im Folgenden dargestellten Abläufen gehen jeweils immer bereits ein Redirect von einer P20-Anwendung zum P20-IAM und von dort ein weiterer Redirect zum TN-IAM voraus. Das TN-IAM muss als nächstes also den Benutzer innerhalb der Teilnehmer-Domäne authentifizieren.

Webservice-Aufruf mit Token-Exchange

Die Authentifizierung gegenüber Webservices erfolgt über dedizierte JWTs. Hierzu kann die Anwendung das Access-Token des Benutzers, mit dem es selbst aufgerufen wurde, beim F-IAM per Token-Exchange gegen ein anderes JWT für den aufzurufenden Webservice tauschen.

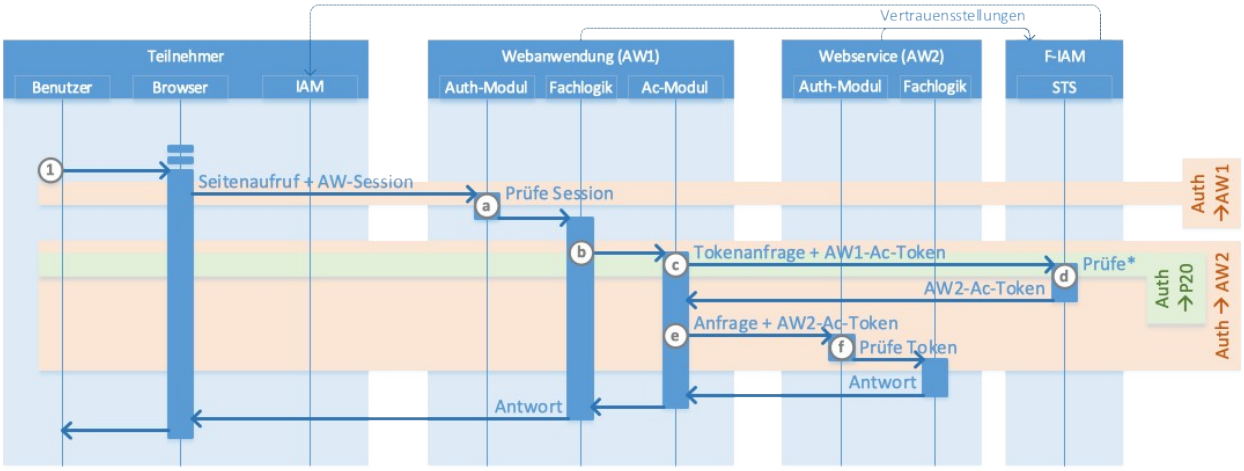

Abbildung 35: Sequenzdiagramm zum Webservice-Aufruf mit Token-Exchange

Ablauf

Der Benutzer ruft im Browser die Webanwendung auf, wobei er bereits angemeldet ist, also Sessioninformationen inkl. eines P20-Access-Tokens in der Anwendung vorliegen.

a. Das Auth-Modul der Anwendung prüft die Sessionsinformationen und leitet die Anfrage an die Fachlogik weiter.

b. Innerhalb der Fachlogik entsteht der Bedarf, einen P20-Webservice aufzurufen.

c. Die Webanwendung sendet eine Tokenanfrage („Token-Exchange“) an das F-IAM und übergibt u.a. das vorliegende Access-Token, mit dem sich der Benutzer zuvor gegenüber der Webanwendung authentifiziert hatte.

d. Das F-IAM prüft die Anfrage und antwortet mit einem Access-Token für den benötigten Webservice, das ggf. auch die Berechtigungsinformationen des anfragenden Benutzer für genau diesen Webservice enthält.

e. Die TN-Anwendung sendet die fachliche Anfrage zusammen mit dem zuvor empfangenen Access-Token an den Webservice.

f. Das Auth-Modul des Webservices prüft das übergebene Token, ermittelt die Berechtigungen und übergibt sie an die Fachlogik. Die Fachlogik erzeugt die Antwort, die dann von der Fachlogik der TN-Anwendung weiterverarbeitet wird.

Berechtigungsabfrage über Userprofile-Endpunkt

Bei manchen Anwendungen kann die Liste der Berechtigungszuweisungen potentiell sehr lang werden, so dass sie nicht in das Access-Token aufgenommen wird. In diesem Fall muss die Anwendung die Berechtigungszuweisungen über einen Userprofile-Endpunkt explizit abfragen, wobei ein Caching zum Steigern der Performance zulässig ist.

Das nachfolgende Sequenzdiagramm beschreibt den Fall, dass eine Webanwendung einen Webservice aufruft, dessen Berechtigungszuweisungen explizit abgefragt werden müssen.

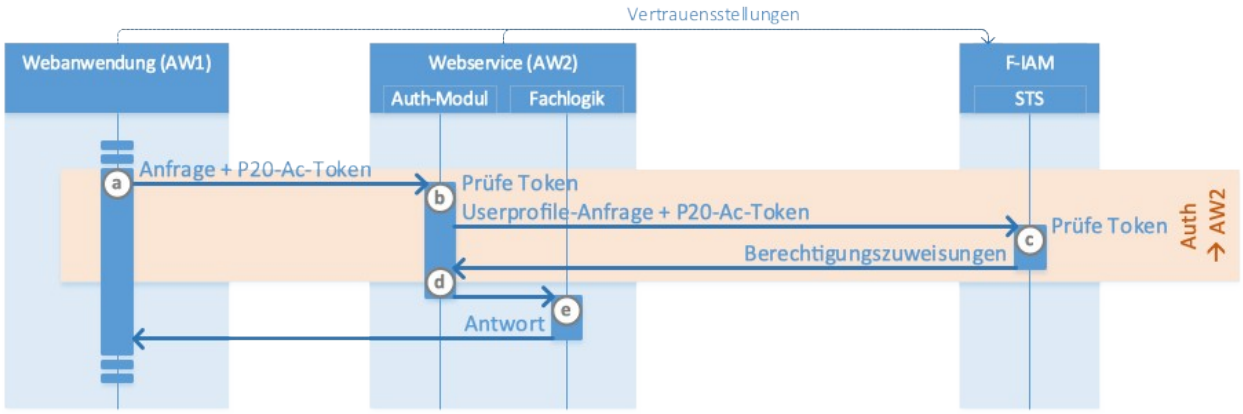

Abbildung 36: Sequenzdiagramm zur Berechtigungsabfrage über Userprofile-Endpunkt

Ablauf

Der Benutzer ist bereits bei der Webanwendung angemeldet, die gerade eine Anfrage des Benutzers bearbeitet. nnerhalb der Webanwendung entsteht der Bedarf, einen P20-Webservice aufzurufen. Hierzu wird ein dediziertes Access-Token übergeben, das zuvor bereits per Token-Echange vom F-IAM bezogen wurde.

a. Das Auth-Modul des Webservices prüft das übergebene Token.

b. Zusammen mit dem empfangenen Access-Token sendet es eine Anfrage an den Userprofile-Endpunkt des F-IAM.

c. Das F-IAM prüft das übergebene Token. Es ermittelt anschließend die Berechtigungszuweisungen des Benutzers für die Anwendung, die in dem Token angegeben ist, und sendet die Liste als Antwort zurück.

d. Dem Auth-Modul des Webservices liegen neben den Benutzerattributen aus dem Token nun auch die Berechtigungszuweisungen vor, und es leitet diese an die Fachlogik weiter.

e. Die Fachlogik verarbeitet die Anfrage und sendet die Antwort zur aufrufenden Webanwendung zurück.

Exemplarischer Login an Teilnehmer-interner Webanwendung per OIDC

TN-interne Anwendungen werden durch den TN selbst bereitgestellt und betrieben. Es greifen nur Benutzer desselben TN auf diese Anwendungen zu, eine Berechtigung/Authentifizierung über den Basisdienst IAM ist für die Nutzung nicht erforderlich. Jedoch kann es sein, dass TN-interne Anwendungen auf P20-Anwendungen zugreifen müssen und für diesen Zugriff dennoch eine Authentifizierung über den Basisdienst IAM realisieren müssen.

Der Login an Teilnehmer-internen Webanwendungen ist kein Gegenstand der Arbeiten im P20 bzw. der PG-IAM. Dieses Szenario soll aber dennoch zeigen, wie die Verfahren auch dafür übernommen werden können.

Der Unterschied zum P20-Login besteht darin, dass die Webanwendung mit dem TN-STS integriert werden muss und der Login ausschließlich mit Teilnehmer-Systemen abläuft, ohne Mitwirkung oder Kommunikation mit P20-Systemen.

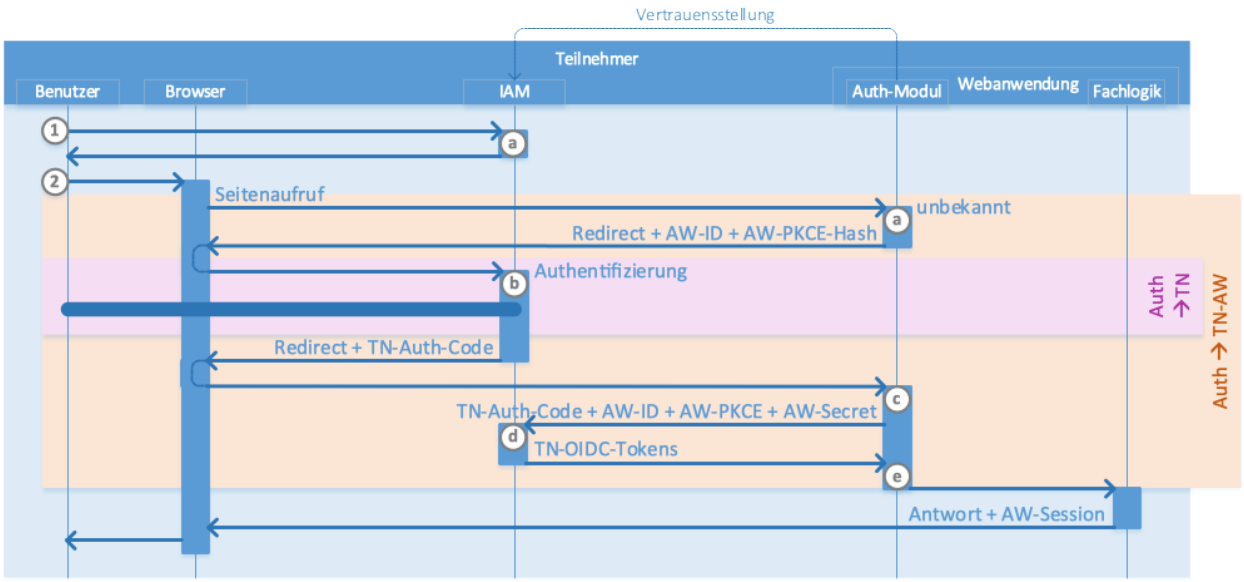

Abbildung 37: Sequenzdiagramm zum exemplarischen Login an TN-interner Webanwendung per OIDC

Ablauf

1. **Login an Teilnehmer-Domäne**

   Der Benutzer meldet sich über sein Endgerät am Teilnehmer-IAM an.

f. Das TN-IAM prüft die Zugangsdaten.

2. **Seitenaufruf**

   Der Benutzer ruft im Browser die Webanwendung auf.

   a. Das Auth-Modul erkennt den anfragenden Benutzer nicht und startet daraufhin eine OIDC-Authentifizierung, indem es mit einem Redirect zum lokalen Security Token Service (TN-STS) antwortet und als Query-Parameter u.a. sowohl die eigene Anwendungs-ID (OIDC-Client-ID) als auch den Hash eines selbst generierten PKCE übergibt.

   b. Der aufgerufene TN-STS kennt den anfragenden Benutzer bisher nicht und veranlasst eine lokale Authentifizierung (siehe 5.3. Sequenzdiagramme zur Authentifizierung gegenüber dem Teilnehmer-IAM).

      Nach Abschluss der Authentifizierung des Benutzers wird als Antwort ein Redirect zu der Webanwendung zusammen mit einem neu generierten Autorisierungscode gesendet.

   c. Das Auth-Modul der Webanwendung verwendet den Autorisierungscode und sendet eine Authentifizierungsanfrage an den TN-STS zusammen mit der eigenen Anwendungs-ID, dem Klartext-PKCE, dessen Hashwert in 2.a übermittelt wurde, sowie dem eigenen Anwendungs-Secret.

   d. Der TN-STS prüft die übergebenen Parameter.

      Anschließend ermittelt er die Benutzerattribute und Berechtigungen. Er antwortet abschließend mit den OIDC-Tokens.

   e. Das Auth-Modul der Webanwendung extrahiert die Benutzerattribute und Berechtigungen.

      Diese werden beim Weiterleiten an die Fachlogik übergeben.

      Die Fachlogik sendet die Antwort und gibt dabei auch Sessioninformationen (z.B. als Session-ID-Cookie oder JWT) mit.

Exemplarischer SSO zwischen P20- und Teilnehmer-Anwendungen

Ist ein Benutzer über seinen Browser bereits an einer Webanwendung angemeldet, so soll er ohne erneute explizite Authentifizierung auch auf andere Webanwendungen zugreifen können – unabhängig davon, ob es sich um P20- Anwendungen oder Anwendungen des eigenen Teilnehmers handelt. Dies ist insbesondere dann für die Benutzerfreundlichkeit relevant, wenn aus einer Webanwendung durch Links auf andere Webanwendungen verwiesen wird.

Die technische Lösung liegt im Setzen einer Session durch den TN-STS, wobei dies unabhängig vom jeweiligen Authentifizierungsverfahren (OIDC und/oder SAML2) funktioniert.

Voraussetzung dafür ist, dass auch für die TN-Webanwendungen eine Authentifizierung über den TN-STS erfolgt.

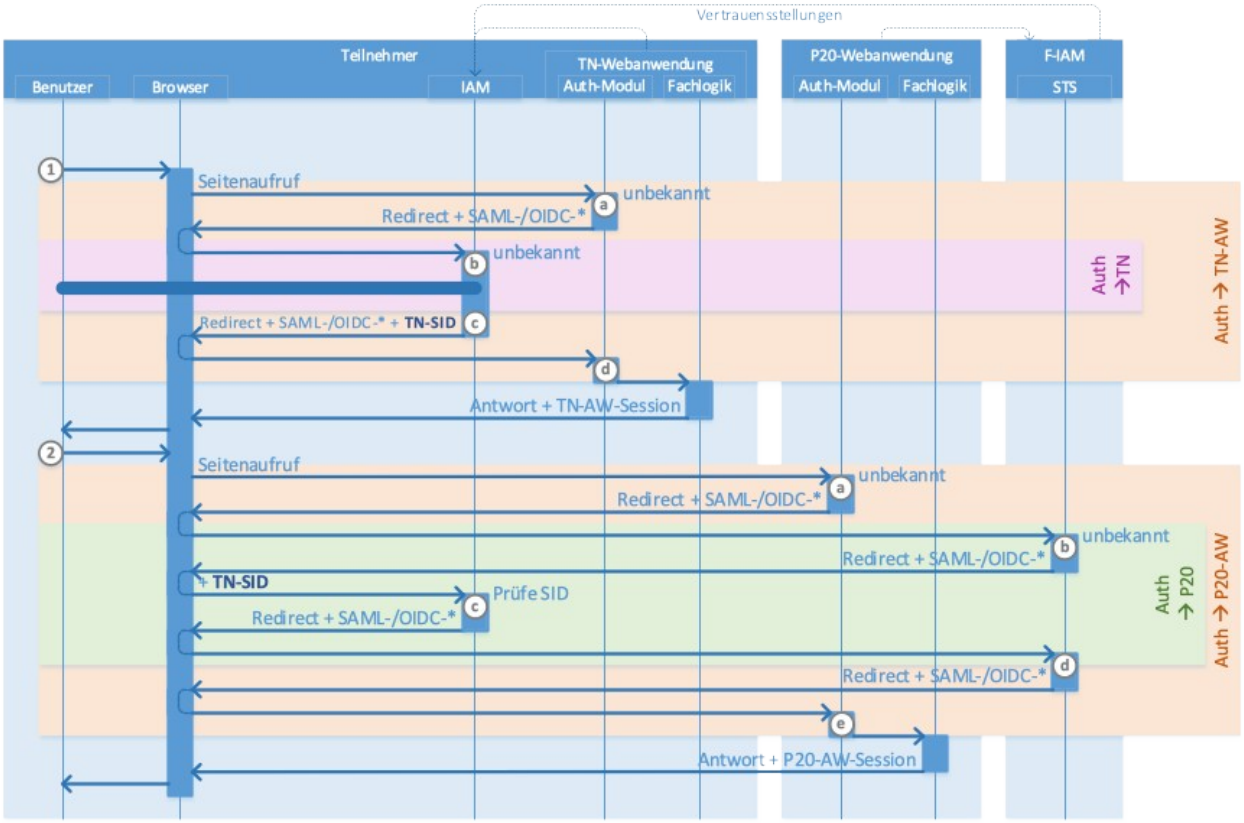

Abbildung 38: Sequenzdiagramm zum exemplarischer SSO zwischen P20- und Teilnehmer-Anwendungen

Ablauf

1. **Seitenaufruf TN-Webanwendung**

   Der Benutzer ruft im Browser eine TN-Webanwendung auf.

   a. Das Auth-Modul erkennt den anfragenden Benutzer nicht und startet daraufhin eine SAML2- oder OIDC-Authentifizierung, die in beiden Fällen zu einem Redirect zum eigenen Security Token Service (TN-STS) als Antwort führt.

   b. Der über den Redirect vom Browser aufgerufene TN-STS erkennt den anfragenden Benutzer nicht und veranlasst eine Authentifizierung gemäß der jeweiligen Teilnehmer-spezifischen Konfiguration (siehe Sequenzdiagramme zur Authentifizierung gegenüber dem Teilnehmer-IAM).

   c. Nach Abschluss der Authentifizierung setzt der TN-STS die SAML2- oder OIDC-Authentifizierung fort, die in beiden Fällen zu einem Redirect zurück zur TN-Webanwendung als Antwort führt. Hierin wird durch den TN-STS auch eine Session-ID als Cookie gesetzt.

   d. Das Auth-Modul setzt die Authentifizierung fort und leitet nach Abschluss an die Fachlogik weiter, die dann die Antwort sendet.

2. **Seitenaufruf P20-Webanwendung**

   Der Benutzer ruft in demselben Browser eine P20-Webanwendung auf.

a. Das Auth-Modul erkennt den anfragenden Benutzer nicht und startet daraufhin eine SAML2- oder OIDC-Authentifizierung, die in beiden Fällen zu einem Redirect zum P20-STS als Antwort führt.

b. Der P20-STS erkennt den anfragenden Benutzer ebenfalls nicht und startet daraufhin eine SAML- oder OIDC-Authentifizierung, die in beiden Fällen zu einem Redirect zum TN-STS als Antwort führt. Beim Folgen des Redirects übergibt der Browser die im Schritt 1.c als Cookie gesetzte TN-Session-ID.

c. Der TN-STS erkennt den anfragenden Benutzer über die Session-ID und setzt direkt die SAML- oder OIDC-Authentifizierung gegenüber der Webanwendung fort, die in beiden Fällen zu einem Redirect zurück zum P20-STS als Antwort führt, ohne dass es einer weiteren Interaktion mit dem Benutzer bedarf.

d. Der P20-STS setzt die SAML2- oder OIDC-Authentifizierung gegenüber der P20-Webanwendung fort, die in beiden Fällen zu einem Redirect zurück zum Webanwendung als Antwort führt.

e. Das Auth-Modul setzt die Authentifizierung fort und leitet nach Abschluss an die Fachlogik weiter, die dann die Antwort sendet.

Das Konzept funktioniert identisch, wenn sich der Benutzer zuerst an einer P20-Webanwendung angemeldet hat und anschließend eine TN-Webanwendung aufruft.

Aufruf P20-Webservice durch Teilnehmer-Anwendung

Wird ein P20-Webservice von der Anwendung eines Teilnehmers aufgerufen, so muss sich der Benutzer mit einem P20-Access-Token authentifizieren. Damit die TN-Anwendung ein solches Token vom P20-STS anfragen kann, muss sie dort als OIDC-Client registriert sein.

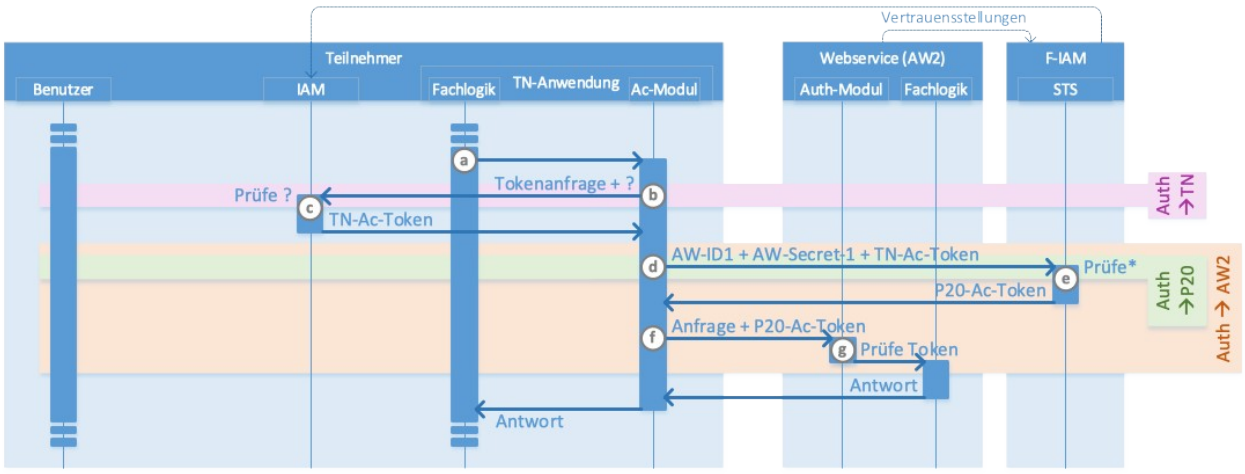

Abbildung 39: Sequenzdiagramm zum Aufruf P20-Webservice durch Teilnehmer-Anwendung

Ablauf

Der Benutzer ist bereits bei der Teilnehmer-Anwendung angemeldet, die gerade eine Anfrage des Benutzers bearbeitet.

a. Innerhalb der Teilnehmer-Anwendung entsteht der Bedarf, einen P20-Webservice aufzurufen.

b. Die TN-Anwendung sendet eine Tokenanfrage an das TN-IAM.

Die genaue Ausgestaltung dieser Anfrage hängt von dem konkreten Zusammenspiel zwischen Anwendung und IAM innerhalb des Teilnehmers ab und liegt komplett in der Hoheit des Teilnehmers. Sofern die Authentifizierung gegenüber der TN-Anwendung bereits per OIDC erfolgt ist, liegt bereits ein Access-Token vor, so dass dieser Schritt entfallen kann.

c. Das TN-IAM prüft die Anfrage und antwortet mit einem Access-Token für den anfragenden Benutzer.

d. Die TN-Anwendung sendet eine Tokenanfrage („Token-Exchange“) an das F-IAM und übergibt u.a. das zuvor empfangene Access-Token.

e. Das F-IAM prüft die Anfrage und antwortet mit einem Access-Token für den benötigten Webservice, das ggf. auch die Berechtigungsinformationen des anfragenden Benutzer für genau diesen Webservice enthält.

f. Die TN-Anwendung sendet die fachliche Anfrage zusammen mit dem zuvor empfangenen Access-Token an den Webservice.

g. Das Auth-Modul des Webservices prüft das übergebene Token, ermittelt die Berechtigungen und übergibt sie an die Fachlogik. Die Fachlogik erzeugt die Antwort, die dann von der Fachlogik der TN-Anwendung weiterverarbeitet wird.

Glossar

Dieses Glossar bezieht sich lediglich auf IAM-Begrifflichkeiten in diesem Dokument. Begrifflichkeiten aus dem Kontext P20 sind in Confluence1 zu finden. Das allgemeine P20-IAM-Glossar ist ebenfalls in Confluence zu finden.

Begriff / Abkürzung	Beschreibung
Access-Token	Kontext: OIDC Wird üblicherweise zur verwendet, um die Autorisierung gegenüber einer Anwendung zu ermöglichen. Im Kontext von P20 sind hier auch bereits die Benutzerattribute und insbesondere die P20-UID enthalten.
ACME	Automated Certificate Management Environment
ABAC	Attribute-Based Access Control Ein Mechanismus zur Verwaltung des Benutzerzugriffs auf Informationssysteme basierend auf Werten von Benutzerattributen. Die attributbasierte Zugriffskontrolle (ABAC) wertet den Zugriff dynamisch aus, indem sie einen Algorithmus verwendet, der „Attribute“ als Eingabe verwendet und eine Zugriffsentscheidung (Zulassen/Verweigern) ausgibt. Bei den Attributen handelt es sich in der Regel um Benutzerattribute aus dem Benutzerprofil, ergänzt um Kontextattribute, wie z. B. den Zeitpunkt des Zugriffs und den aktuellen Standort des Benutzers.
Assertion	Eine SAML2 Assertion ist ein XML-Nachricht, die einem SP mitteilt, dass ein Benutzer angemeldet ist. SAML-Assertion enthalten alle Informationen, die ein SP benötigt, um die Benutzeridentität zu bestätigen, einschließlich der Quelle der Assertion, des Ausstellungszeitpunkts und der Bedingungen die die Gültigkeit der Assertion ausmachen. Eine SAML2 Assertion stellt somit eine bestimmte Form eines Security Tokens dar.
AuthN Request	SAML2 AuthnRequest ist eine XML Nachricht, die ein SP an einen IdP sendet, um die Authentifizierung zu initiieren. Diese Meldung ist Base64-codiert. Zusammen mit dem Base64-codierten Request kann ein Relaystate-Token an den IdP gesendet werden
AuthN Response	Als Antwort auf einen SAML2 AuthN Request und für den Fall, dass der Benutzer durch den IdP erfolgreich authentifiziert werden konnte sendet der IdP eine XML Nachricht mit Status und Assertions an den SP zurück. Der IdP authentifiziert den Benutzer und der SP ist an diesem Prozess nicht beteiligt. Der SP erhält nur den Status der Authentifizierung
F-IAM	Der Begriff Föderiertes Identity & Access Management umfasst sowohl die Mechanismen zur Zugriffssteuerung (Access

	Management) als auch die Bereitstellung von Benutzerkonten (Identity Management). In der Literatur wird zumeist der Begriff „Föderiertes Identity Management“ verwendet ohne näher darauf einzugehen, dass unter diesem Begriff lediglich die Mechanismen der Zugangskontrolle und Zugangsteuerung verstanden werden. Es handelt sich also vereinfacht ausgedrückt um ein föderiertes Verfahren zum Single Sign On. Die in diesem Kontext notwendigen Prozesse zur Verwaltung des Lebenszyklus von Benutzerkonten, insbesondere der korrekten Terminierung der Benutzerkonten werden hingegen beiläufig oder gar nicht betrachtet.
Föderierte Identität	Digitale Identität, die in mehreren Domänen verwendet werden kann, normalerweise mithilfe einer Identitätsföderation. Informationen zur föderierten Identität werden zwischen Domänen übertragen, normalerweise in Form von Identitätsbehauptungen, die zwischen Identitätsanbietern und vertrauenden Parteien ausgetauscht werden. ISO 24760-Begriff: föderierte Identität Siehe auch: Identitätsföderation, Digitale Identität
GDPR	General Data Protection Regulation Die Datenschutz-Grundverordnung 2016/679 (DSGVO) ist eine Verordnung der Europäischen Union zum Schutz personenbezogener Daten und der Privatsphäre. Sie definiert Regeln für die Verarbeitung personenbezogener Daten in der Europäischen Union und im Europäischen Wirtschaftsraum, wobei die Bestimmungen der Verordnung auch für andere Parteien gelten.
GRC	Governance, Risk, and Compliance ist eine strukturierte Methode, um die IT mit den Unternehmenszielen in Einklang zu bringen und gleichzeitig Risiken zu verwalten und alle branchenüblichen und gesetzlichen Vorschriften einzuhalten. GRC umfasst Instrumente und Vorgänge, die die Unternehmensführung und das Risikomanagement eines Unternehmens mit der technologischen Innovation und der Übernahme von Technologien verbinden. Unternehmen benutzen GRC, um Unternehmensziele zuverlässig zu erreichen, Unsicherheiten zu beseitigen und Compliance-Anforderungen zu erfüllen.
IAM	Identity & Access Management bezieht sich auf die Prozesse, Verfahren und Technologien im Zusammenhang mit der

	Verwaltung von Identitätsdaten, einschließlich Authentifizierung, Autorisierung und Benutzerverwaltung. IAM trägt dazu bei, dass die Zugriffsrechte von Einzelpersonen je nach ihren geschäftlichen Rollen oder Beziehungen in einer Organisation entsprechend angewendet werden.
ID-Token	Kontext: OIDC Enthält üblicherweise die jeweils relevanten Attribute des Benutzers. Die Verwendung ist im Kontext von P20 nicht notwendig, da alle erforderlichen Informationen bereits über das Access-Token bezogen werden können.
IdP	Als Identity Provider oder Identitätsanbieter werden zentrale Zugangssysteme für Dienstanbieter bezeichnet. Benutzer verifizieren über IdP's ihre Identität via Passwort und/oder anderen Faktoren, um sich auf lokalen Geräten oder Internetkonten anzumelden. Über die Auslagerung der Benutzerauthentifizierung an externe Provider können Onlinedienste und Anwendungen von Single Sign-On (SSO) profitieren. Dabei erfolgt eine zentrale Anmeldung beim IdP, um die jeweiligen Anwender:innen global für alle verknüpften Anwendungen und Dienste freizuschalten.
Identity Governance	Aspekt der Verwaltung von Identitäten, einschließlich Geschäftsprozessen, Regeln, Richtlinien und Organisationsstrukturen. Jede Komplettlösung für die Verwaltung von Identitäten besteht aus zwei Hauptteilen – Identity Governance und Identity Management. Alternative Begriffe: Governance
Identity Lifecycle	Reihe von Phasen einer Identität von der Erstellung bis zu ihrer Deaktivierung oder Löschung. Es umfasst die Erstellung eines Kontos, die Zuweisung korrekter Gruppen und Berechtigungen, das Festlegen und Zurücksetzen von Passwörtern und letztendlich die Deaktivierung oder Löschung des Kontos. Siehe auch: Identitätsbereitstellung,
IDM	Identity Management ist ein Prozess zur Verwaltung digitaler Identitäten und ihres Zugriffs auf bestimmte Ressourcen im Cyberspace. Es sorgt für den richtigen Zugriff in angemessener Zeit und hilft bei der Verwaltung von Benutzerkonten sowie bei der Synchronisierung von Daten. Das Identitätsmanagement befasst sich mit dem Lebenszyklus digitaler Identitäten und verwaltet die Werte digitaler Identitätsattribute und -berechtigungen. Alternative Begriffe: Identitätsverwaltung, Benutzerverwaltung, Benutzerbereitstellung ISO 24760-Begriff: Identitätsmanagement

X.1252-Begriff: Identitätsmanagement