UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Andrej Habazin

Sistemi za upravljanje z identitetami

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: dr. Andrej Brodnik

Ljubljana, 2016

Besedilo je oblikovano z urejevalnikom besedil LaTeX.
Slike so izdelane s pomočjo jezika PGF/TikZ.
Grafi so narisani s pomočjo programa ***gnuplot****.*

Katerakoli aplikacija, ki želi biti različna glede na uporabnika, mora slednjega nekako razlikovati. Zato, da ga razlikuje, mora obstajati način, da ga enolično označi (identificira) ter se nato glede na zapis v svoji podatkovni bazi identitet ustrezno odzove. Seveda obstajajo butične ali adhoc izvedbe baz in po drugi strani celotni sistemi, ki jih lahko samo integriramo v aplikacije. V diplomski nalogi preučite sistema MIM (Microsoft Identity Manager) in OpenIDM, ju primerjajte in ovrednotite na podlagi kakovostnih kazalnikov, ki jih določite na osnovi pregleda.

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a Andrej Habazin,

z vpisno številko 63040048,

sem avtor/-ica diplomskega dela z naslovom:

Sistemi za upravljanje z identitetami

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom prof. dr. Andreja Brodnika
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) in ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 09.09.2016 Podpis avtorja/-ice:

Zahvala

Na tem mestu bi se želel zahvaliti vsem, ki so kakorkoli pripomogli k uspešni izvedbi diplomskega dela. Še posebej bi se želel zahvaliti mentorju, prof. Brodniku za vse konstruktivne nasvete in pripombe, Mateju Gromu za pomoč pri prenosu/vzpostavitvi infrastrukture na fakultetni strojni opremi in asistentu Gašperju Feletu za vse odgovore na vprašanja, ki sem jih imel.

Never go to war. Especially with yourself.

Kazalo

Seznam uporabljenih kratic in simbolov

ACL	angl. *Access Control List*: seznam dovoljenj in pravic do nekega objekta.
AD	angl. *Active Directory*: sistem imeniške storitve podjetja Microsoft.
BAR	angl. *Business Archive*: datoteka, ki vsebuje opis poslovnega procesa v formatu BPMN.
BPM	angl. *Business Process Management*: upravljanje poslovnih procesov.
BPMN	angl. *Business Process Model and Notation*: standard za modeliranje in načrtovanje poslovnih procesov.
CA	angl. *Certification Authority*: agencija za izdajo digitalnih certifikatov.
CDDL	angl. *Common Development and Distribution License*: licenca za odprtokodne rešitve podjetja Sun Microsystems. Temelji na licenci Mozilla Public License.
CRUD	angl. *Create, Read, Update Delete*: akcije, ki se lahko izvedejo nad nekim objektom.
DLL	angl. *Dynamic Linked Library*: dinamično povezljiva knjižnica.
DSN	angl. *Data Source Name*: podatkovna struktura, ki opisuje povezavo s podatkovno bazo.

FIM angl. *Forefront Identity Manager*: sistem za upravljanje identitet podjetja Microsoft, predhodnik MIM.

IEC angl. *International Electrotechnical Commission*: mednarodna elektrotehniška komisija.

ISO angl. *International Organization for Standardization*: mednarodna organizacija za standardizacijo.

JSON angl. *JavaScript Object Notation*: format za izmenjavo podatkov.

LDAP angl. *Lightweight Directory Access Protocol*: protokol za komunikacijo z imeniškimi storitvami.

LDIF angl. *LDAP Data Interchange Format*: format za izmenjavo podatkov med LDAP imeniki.

MIM angl. *Microsoft Identity Manager*: sistem za upravljanje z identitetami podjetja Microsoft, naslednik FIM.

MPR angl. *Management Policy Rule*: upravljalsko pravilo s katerim se v MIM/FIM definira delovanje sistema

NIST angl. *National Institute of Standards and Technology*: Nacionalni inštitut za standarde in tehnologije.

ODBC angl. *Open Database Connectivity*: odprta podatkovna povezljivost - standardna metoda dostopa do podatkovne baze s SQL.

OpenIDM angl. *Open Identity Manager*: odprotokodni sistem za upravljanje z identitetami podjetja RockForge.

OSGi angl. *Open Services Gateway Initiative*: konzorcij tehnoloških inovatorjev, ki podpira razvoj odprtih specifikacij in odprtokodnih tehnologij na Java tehnologiji.

PAM angl. *Privileged Access Management*: upravljanje posebnih dostopov identitet.

RBAC angl. *Role Based Access Control*: dodelitev dovoljenj v skladu z vlogami v podjetju.

REST angl. *Representational State Transfer*: programska arhitektura za objavo virov na spletu.

SSL angl. *Secure Sockets Layer*: kriptografski protokol za zagotavljanje varne komunikacije med strežnikom in odjemalcem.

SOA angl. *Service Oriented Architecture*: storitveno naravnana arhitektura.

SoD angl. *Separation of Duties*: ločevanje zadolžitev.

SSO angl. *Single Sign On*: enotna prijava.

SQL angl. *Structured Query Language*: strukturirani jezik za poizvedbe.

TLS angl. *Transport Layer Security*: kriptografski protokol za zagotavljanje varne komunikacije med strežnikom in odjemalcem. Gre za naslednika SSL.

WAL angl. *Workflow Activity Library*: knjižnica delovnih tokov za uvedbo kompleksnih procesov v MIM.

XML angl. *Extensible Markup Language*: razširljivi označevalni jezik, format podatkov za izmenjavo strukturiranih dokumentov.

Povzetek

Sistemi za upravljanje z identitetami omogočajo večjim organizacijam upravljanje nadzor nad viri, ki jih identitete uporabljajo. Osnovna naloga teh sistemov je vključuje zagotavljanje in vzdrževanje skladnost z varnostnimi ter ostalimi politikami organizacije. Poleg tega nudi avtomatizacijo ponavljajočih se opravil skozi procese samooskrbe, ki razbremenjujejo službe za podporo uporabnikom in omogočajo prihranke ter popolno sledljivost za vsak posamezni zahtevek.

V nalogi bomo spoznali podporne varnostne protokole, ki zagotavljajo varno avtentikacijo in zaščito informacij o identitetah, principe delovanja, varnostne vidike ter najpogostejše smeri napadov. Poleg tega bomo spoznali zgradbo sistemov OpenIDM in Microsoft Identity Manager 2016, njune prednosti ter slabosti pri povezljivosti z zalednimi sistemi znotraj organizacij, namestitvi in vzpostavitvi sistema, uvedbi na primeru problemske domene ter vzdrževanju sistema.

Dodatno bomo pri uvedbi upravljanja z identitetami preverili integracijo OpenIDM in MIM z imeniško storitvijo OpenLDAP in podatkovno bazo PostgreSQL. Oba izdelka sta razvita v skladu z odprtokodno filozofijo in zato zanimiva za uporabo povsod, kjer želimo zmanjšati stroške na račun licenčnin. Podatkovno bazo PostgreSQL bomo uporabili kot vir podatkov. V imeniški storitvi OpenLDAP bomo na podlagi podatkov ustvarili uporabniške račune.

Ključne besede:

upravljanje z identitetami, OpenIDM, Microsoft Identity Manager, avtomatizacija pogostih opravil, meta imenik, varnostni protokoli

Abstract

Identity management systems allow larger organizations management and control over resources, used by identites. Primarily, these systems maintain and enforce security and other organizational policies. Secondary task is to provide a framework for automation of repetitive tasks and self service processes, which allows a reduction of workload on helpdesk services and yet provides traceability for individual request.

We'll go through some of most important supporting security protocols, which ensure identity authentication and data protection, their principles, security assessment and most common vectors of attacks. In our work, we'll present structure of two identity management systems, OpenIDM and Microsoft Identity Manager 2016, their advantages and disadvantages when it comes to connecting with backend systems within organization, installation and implementation of the system on problem domain, introduction and maintenance of the system for identity management.

Additionally, we'll evaluate integration with both OpenLDAP directory and PostgreSQL database. Both systems are developed in accordance with open source philosophy therefore they're interesting for implementation in order to reduce costs of using licensed software. PostgreSQL database will be used as an identity data source. User accounts will be created in OpenLDAP based on the data from PostgreSQL.

Key words:

Identity Management, OpenIDM, Microsoft Identity Manager, automation of common tasks, meta directory, security protocols

Poglavje 1

Uvod

S povečano gospodarsko rastjo in naraščujočo zahtevnostjo organizacij ter številom zaposlenih se poslovne združbe dandanes soočajo s problemom zagotavljanja in pravočasnega ukinjanja dostopov do informacijskih virov zaposlenim. Prav tako z razmahom oblačnih storitev marsikateri zaledni sistem ne deluje več znotraj neke organizacije.

Potrebne so torej rešitve, ki zagotavljajo: da informacije znotraj podjetja ne odtekajo nepooblaščenim osebam, ki omogočajo revizijsko sled; pregled dostopov in pravic; večnivojsko potrjevanje zahtevkov; ter razbremenitev tipičnih administrativnih opravil (oddaja zahtevka za uporabniški račun v sistemu, ponastavitev gesla, oddaja zahtevka za privilegiran dostop, administrativni račun itd.)

Organizacije, ki procesov upravljanja digitalnih identitet nimajo ustrezno rešenih in podprtih, tvegajo veliko finančno izgubo ter negativno publiciteto v primeru vdora v njihove sisteme [2]. Težave ne predstavljajo samo zunanji napadalci. Po analizi IBM Cyber Security Intelligence Index je bilo v letu 2015 kar 55% [1] vseh napadov sproženih znotraj organizacije (osebe s fizičnim ali oddaljenim dostopom do informacijskih virov organizacije; tipično redno zaposleni, pogodobeni sodelavci, svetovalci,...). Kar 31.5% [1] vseh napadov je bilo sproženih znotraj organizacije z namenom povzročiti škodo organizaciji.

Napade znotraj organizacije lahko dobro uveden proces upravljanja z identitetami omeji, saj zaposlenim dodelimo le pravice do virov, ki jih potrebujejo pri svojem delu. Enako lahko pri zamenjavi delovnega mesta poskrbimo, da po preteku prehodnega obdobja izgubi predhodne pravice in da se vsi dostopi ter uporabniški računi ukinejo s potekom zaposlitve.

1.1 Struktura diplomskega dela

V diplomskem delu bomo najprej spoznali teoretične osnove upravljanja z identitetami, potrebno terminologijo in skupne značilnosti sistemov (delegiranje dostopov in zadolžitev, eskalacija zahtevkov, revizija in potrjevanje dostopv, poročila ipd.), neodvisne od uvedbe.

V nadaljevanju se bomo dotaknili podpornih protokolov in varnostnih vidikov ter težav, s katerimi se srečujemo pri uvedbi in vzdrževanju sistemov za upravljanje z identitetami.

V tretjem delu bomo spoznali sistema za upravljanje z identitetami OpenIDM in MIM 2016, njuno arhitekturo ter način delovanja. Na primeru problemske domene iz realnega sveta bomo primerjali uvedbi obeh sistemov ter izpostavilil njune prednosti in pomanjkljivosti.

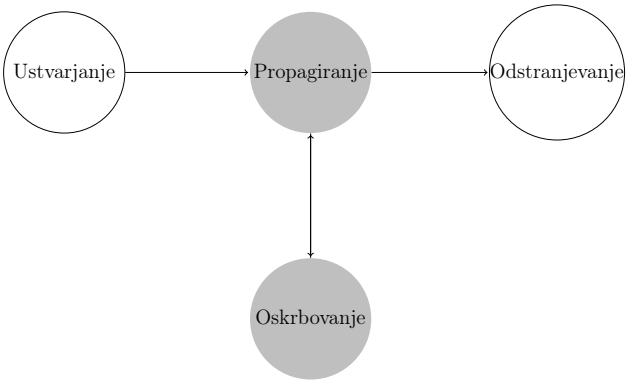
Poglavje 2

Terminologija in področje upravljanja z identitetami

2.1 Digitalna identiteta

Standard ISO/IEC 24760-1 *digitalno identiteto* definira kot „Nabor atributov, povezanih z entiteto“ [25]. V digitalnem svetu gre za zapis, ki enolično predstavlja bodisi osebo bodisi nek drug vir (npr. računalnik ali drugo napravo). Zapis je sestavljen iz različnih atributov (lastnost; v primeru osebe je to lahko ime, priimek, enotna številka zaposlenega, datum začetka delovnega razmerja, organizacijska enota ipd.). Praviloma se v sistemih za upravljanje z identitetami izogibamo osebnim podatkom kot so EMŠO, davčna številka, datum rojstva. Ti podatki se praviloma ne uporabljajo pri upravljanju identitet, saj jih je potrebno varovati v skladu z zakonodajo pred nepooblaščenim dostopom. Namesto tega uvedemo enotno številko zaposlenega, na podlagi katere povežemo identiteto s fizično osebo.

V računalniškem svetu na identiteto gledamo kot na entiteto, kateri zaupamo skozi proces avtentikacije (angl. *authentication*), s čimer zagotovimo dostop do virov in virov, do katerih ima identiteta nadzorovan dostop. Sam dostop do virov se vrši s procesom avtorizacije (angl. *authorization*). Identiteta



Slika 2.1: Življenjski cikel identitete.

predstavlja osnovni gradnik, na osnovi katerega modeliramo procese upravljanja. Ti definirajo življenjski cikel identitete, ilustriran na sliki 2.1. Vsako identiteto je potrebno ustvariti, jo razpršiti na povezane sisteme (angl. *propagate*), oskrbovati (angl. *provision*) in na koncu tudi odstraniti (angl. *deprovision*) [24].

2.1.1 Avtentikacija

Avtentikacija (angl. *authentication*) je proces, s katerim potrdimo istovetnost nekega podatka (lastnosti) ali entitete v celoti. Za razliko od procesa identifikacije, ki zgolj ugotavlja istovetnost identitete, proces avtorizacije identiteto tudi potrdi in zanjo jamči. V informacijskem svetu je tipičen primer avtentikacije kombinacija uporabniškega imena in gesla. Uporabniško ime podaja povezavo s fizičnim uporabnikom, medtem ko geslo omogoča preverjanje, ali je fizični uporabnik res tisti, za katerega se izdaja.

Metod za preverjanje avtentičnosti je več [21] (od bolj razširjenih do bolj zahtevnih za uvedbo):

- preverjanje z osebnimi številkami in gesli,
- preverjanje s simetričnimi ključi,

- preverjanje z žetoni,
- večfaktorsko preverjanje ali
- preverjanje z biometričnimi podatki.

Ne glede na metodo preverjanja avtentičnosti mora stranka, ki terja (angl. *claimant*) predstaviti dovolj informacij, na podlagi katerih ga lahko mehanizem za preverjanje ustrezno preveri. Tem informacijam rečemo tudi *dokaz o posedovanju* (angl. *Proof of Possession*).

Večfaktorsko preverjanje ali avtentikacija je mehanizem, pri katerem moramo za uspešno avtenticiranje zagotoviti dva ali več ločenih dokazov o posedovanju. Dokazi spadajo v eno od sledečih kategorij: posedovanje skrivne informacije, posedovanje predmeta za avtentikacijo in posedovanje biometrične lastnosti.

2.1.2 Avtorizacija

Avtorizacija je proces preverjanja in potrjevanja, ali ima določena identiteta pravice dostopati oz. uporabljati nek vir (angl. *resource*). V okviru avtorizacije so definirani pravila uporabe in mehanizmi, ki zagotavljajo, da sistem vsak zahtevek obravnava ter ga bodisi odobri bodisi zavrne (če gre za zlorabo ali prekoračitev pooblastil).

Sistem za preverjanje avtorizacij imajo nad viri definirane pravice (npr. branje, pisanje, izvajanje, brisanje), organizirane v sezname kontrole dostopa ali sezname ACL (angl. *Access Control List*). Slednji so definirani v širšem sklopu pristopne politike (angl. *Access Policies*).

2.2 Federirana identiteta

Dosedaj smo spoznali definicijo identitete in njeno namembnost znotraj določenega sistema ter okolja. Dandanes so zelo popularne storitve SOA (angl.

Service Oriented Architecture), ki prav tako potrebujejo digitalno identiteto, da lahko z njimi upravljamo. Federirana identiteta je način, kako identitete v različnih sistemih povežemo med seboj, da odražajo eno samo identiteto [28].

Centralizirani sistemi za upravljanje identitet so zgodovinsko delovali znotraj nekega omrežja ali domene. Z integracijo centraliziranih sistemov z zunanjimi storitvami in sistemi prihaja do izmenjave dostopov med zunanjimi ter notranjimi sistemi v obe smeri, kar vodi v decentralizacijo upravljanja z identitetami. Izzivov, ki jih federacija upravljanja z identitetami prinaša, je več, od varne izmenjave podatkov do povezljivosti med različnimi platformami in še posebej zaupanja (angl. *trust*). V ta namen se uporabljajo odprti industrijski standardi, s katerimi lahko dosežemo povezljivost, neodvisno od specifik posamezne platforme.

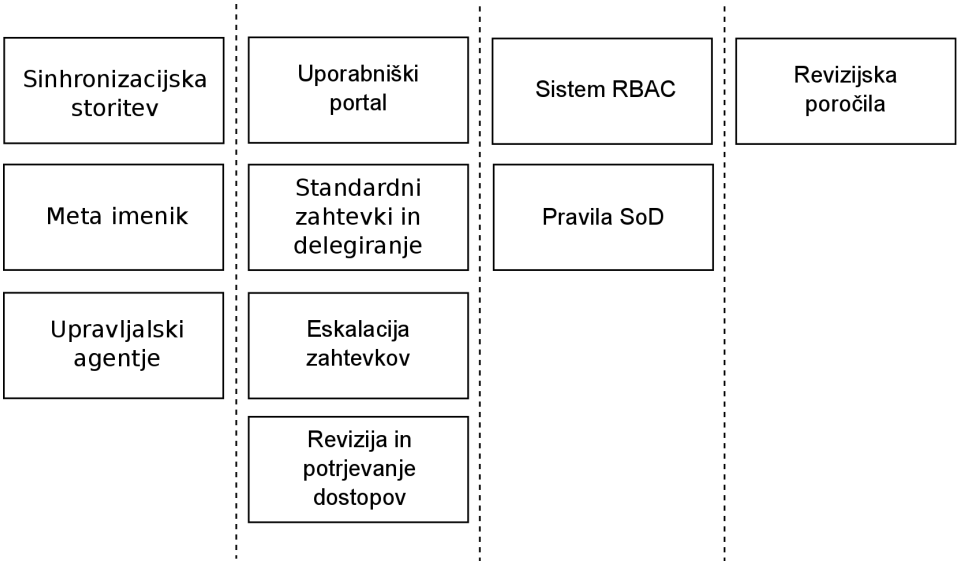
Cilj vpeljave federacije identitet je zagotoviti celovit in varen dostop do drugega domenskega področja, brez odvečne administracije ter na uporabniku neviden način. Tipični scenariji uporabe obsegajo oskrbovanje identitet in upravljanje z dostopi med dvema domenama ter spletno prijavo z enotnim geslom - SSO (angl. *Single Sign On*).

2.3 Arhitektura sistemov

Kot je razvidno na sliki 2.2, bomo v nadaljevanju spoznali arhitekturo sistema za upravljanje z identitami in njegove najpomembnejše gradnike, ki tvorijo celoto in so pomembni z vidika delovanja sistema.

2.3.1 Meta imenik

Meta imenik (angl. *metaverse*) je v svetu upravljanja z identitetami osrednji imenik (v literaturi srečamo tudi pojem repozitorij) digitalnih identitet. V njem so zbrani vsi atributi, ki definirajo identiteto. V meta imeniku so tudi modelirane povezave med različnimi podatkovnimi strukturami, ki definirajo spremljajoče objekte (npr. uporabniški račun, varnostna skupina, članstvo v



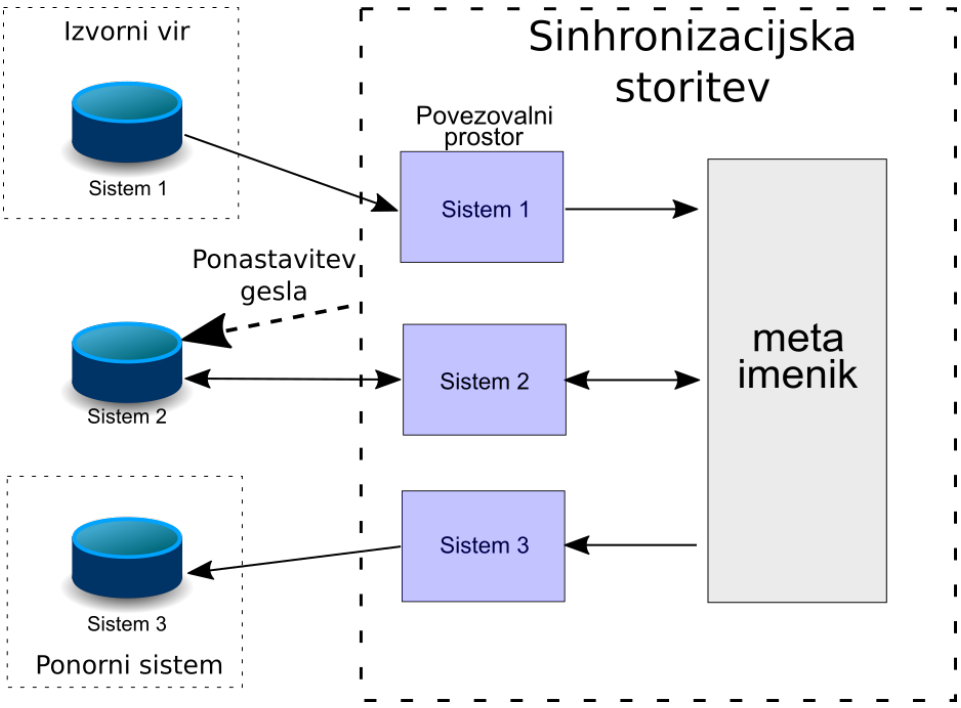
Slika 2.2: Arhitektura sistemov za upravljanje z identitetami.

skupini, računalnik, mobilna naprava itd.). Meta imenik je pogosto uveden kot objektna podatkovna baza, prednosti katere je, da lahko objekte modeliramo poljubno (dodajanje, spreminjanje in odstranjevanje atributov ter tipov objektov). Kot slabost lahko navedemo nekoliko slabše performančne zmogljivosti (število obdelanih objektov na sekundo, hitrost poizvedb) v primerjavi z relacijskimi podatkovnimi bazami. V praksi se izkaže, da prednosti objektne podatkovne baze izničijo njene slabosti.

2.3.2 Upravljalski agentje

Podatkov v meta imenik v splošnem ne moremo dodajati ročno, temveč se vnašajo preko virov podatkov (relacijske podatkovne baze, imeniške storitve, zaledni sistemi itd.). Da lahko s sistema preberemo relevantne informacije, uporabljamo upravljalske agente. *Upravljalski agent* ali *povezovalnik* je skupek kode, ki je sposobna (varne) komunikacije z nekim sistemom. Povezovalnik zna tako brati in pisati v podatkovne strukture sistema, ki predstavljajo digitalno identiteto v sistemu.

Slika 2.3 prikazuje shemo prehodov podatkov med sistemi in vlogo meta imenika. Povezovalnik vzdržuje lokalno kopijo vseh podatkov sistema, s katerim se povezuje. Ti podatki se potem na podlagi vnaprej definiranih pravil zapisujejo v meta imenik ali se iz meta imenika prenašajo. Lokalni kopiji pogosto rečemo *povezovalni prostor* (angl. *Connector Space*), saj je vmesnik med meta imenikom in ciljnim sistemom. Če sistem predstavlja izvor podatkov za nek tip objekta (npr. sistem kadrovske evidence), takemu sistemu rečemo *avtoritativni* ali *izvorni vir podatkov*. Sistem, v katerega povezovalnik podatke zgolj zapisuje, imenujemo *ponorni sistem*.



Slika 2.3: Shema sinhronizacijske storitve in prehoda podatkov med sistemi v povezovalni prostor in meta imenik.

2.3.3 Sinhronizacijska storitev

Sinhronizacijska storitev je tipično centralizirana komponenta, praviloma se izvaja kot nek proces, ki shranjuje in združuje informacije med povezanimi sistemi. Cilj sinhronizacijske storitve je zagotoviti enotno predstavitev identitete v sistemih. Osnovna naloga sinhronizacijske storitve je prenos podatkov v meta imenik in iz njega v povezovalne prostore povezovalnikov v skladu s konfiguriranimi pravili izmenjave podatkov. Sinhronizacijska storitev izmenjuje podatke z agenti v vnaprej predpisanem zaporedju, temu rečemo tudi *sinhronizacijski cikel*. Sinhronizacijska storitev lahko opravlja tudi ponastavljanje in sinhronizacijo gesel med posameznimi sistemi, kar je razvidno s slike 2.3.

2.3.4 Uporabniški portal

Uporabniški portal deluje kot vstopna točka za uporabnike, ki potrebujejo storitve upravljanja identitet. Uporabniški portal omogoča končnim uporabnikom urejanje podatkov, sprožanje in potrjevanje standardnih zahtevkov, ustvarjanje novih objektov ter atributov, vpogled v revizijsko sled in pregled poročil. Portal se uporablja tudi za upravljanje identitet in konfiguracijo sistema s strani administratorjev. Odvisno od arhitekture sistema omogoča tudi sprožanje in izvajanje delovnih tokov ter procesov in registracijo uporabnikov za procese samooskrbe (ponastavljanje gesla, zahtevki za članstva v distribucijskih skupinah itd.).

2.4 Upravljanje z digitalnimi identitetami

Med prednostne naloge upravljanja z identitetami, štejemo upravljanje z avtorizacijami, ki jih identiteta poseduje. V nadaljevanju bomo spoznali mehanizme, ki zagotavljajo, da identiteta ne pridobi ali ohrani avtorizacij, ki bi pomenile kršitev pravil ali dovoljevale možnost zlorabe. Za zagotavljanje skladnosti bomo spoznali mehanizem s katerim periodično pregledujemo do-

deljene avtorizacije, dostope in jih potrjujemo. Z vidika časovnih in finančnih prihrankov so zanimivi procesi samooskrbe preko standardnih zahtevkov, ki končnim uporabnikom omogočajo hitrejšo izvedbo ponavljajočih se opravil ali delegiranje zadolžitev.

2.4.1 Preprečevanje združljivosti nezdružljivih vlog

Pri dodeljevanju pravic identitetam se pogosto srečamo s pravilom, da določen nabor pravic in dovoljenj nikoli ne sme biti podeljen isti identiteti, saj bi s tem kršili pravila o združljivosti nezdružljivih vlog SoD (angl. *Separation of Duties* ali *Segregation of Duties*). Na področju informacijske tehnologije se tovrstna pravila uporabljajo za preprečevanje vnosa podatkov ali sprememb s strani ene same osebe, kraje podatkov ali zlorabe pooblastil in konfliktov interesov. Za to tudi obstaja ustrezna zakonska podlaga [22], ki so jo podjetja dolžna upoštevati in uvajati v svojih poslovnih procesih.

Sistemi za upravljanje z identitetami omogočajo modeliranje pravil na podlagi katerih lahko sistem zavrne dodelitev pravic, ki so v konfliktu. Kot primer iz realnega sveta lahko navedemo sledeč scenarij: uporabnik nikoli ne sme imeti pravic za vnos in potrjevanje faktur, potnih stroškov ali dopustov. V tovrstnem primeru se uporabniku vzamejo vsa dovoljenja, ki so v konfliktu. Sistem prav tako ustrezno opozori na neskladje odgovorne osebe, ki mora odpraviti nepravilnosti v predpisanem času, preden neskladje eskalira po lestvici odgovornosti.

2.4.2 Samodejno dodeljevanje vlog na podlagi atributov

Dodeljevanje vlog na podlagi atributov je rešitev, kako omejiti dostope zgolj avtoriziranim uporabnikom. Tipično se tovrstne rešitve uporabljajo v večjih okoljih, kjer je število zaposlenih dovolj veliko in organizacija okolja takšna, da omogoča vpeljavo standardiziranih pravil za nadzor dostopov (npr. uporabniki, zaposleni na lokaciji A, morajo imeti fizični dostop do stavb X in Y). V

ta namen je bil razvit nadzorni mehanizem RBAC, ki dodeljuje dovoljenja in pravice uporabnikom na podlagi pravil ter atributov (angl. *Role Based Access Control*) [23]. Osnovni gradniki modela RBAC so dovoljenja, vloge in uporabniki. Pravila RBAC modelirajo povezavo med vlogo in dovoljenji, povezavo med uporabnikom ter vlogo in povezave med vlogami. Standard NIST [19] definira štiri različne nivoje RBAC:

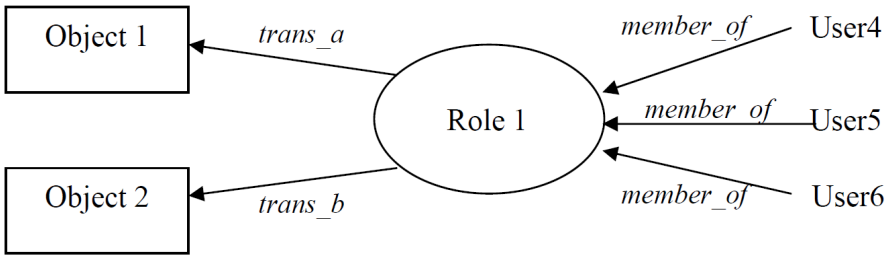
- osnovni RBAC,
- hierarhični RBAC, dodana podpora dedovanju lastnosti med vlogami,
- omejevani RBAC, dodana podpora preprečevanju združljivosti nezdružljivih vlog,
- simetrični RBAC.

Osnovni RBAC

Osnovni RBAC na sliki 2.4 v marsičem spominja na klasično upravljanje dovoljenj z varnostnimi skupinami in ga je možno povsem preprosto uvesti na sistemih, ki podpirajo koncept varnostnih skupin. Vlogo predstavlja varnostna skupina, ki združuje nabor dovoljenj. Ko uporabnika dodamo v skupino, mu s tem podelimo vlogo. V osnovnem modelu RBAC so vloge med seboj neodvisne in nepovezane, kar lahko povzroči težave, kadar imamo opravka z vlogami, pri katerih obstajajo relacije. Osnovni RBAC ne omogoča uporabe pravil o združljivosti nezdružljivih vlog SoD.

Hierarhični RBAC

Hierarhični RBAC na sliki 2.5 prinaša razširitev osnovnega modela in omogoča modeliranje povezav med posameznimi vlogami, kar omogoča uvedbo dedovanja dovoljenj med vlogami. Posledično se s tem zmanjša kompleksnost uvedbe in količine pravil v modelu RBAC, saj lahko postavimo nadredna pravila v ustrezno vlogo ter s tem poskrbimo, da vse vloge, ki jo dedujejo, dedujejo tudi



Slika 2.4: Povezava med objekti, vlogami in uporabniki. [29]

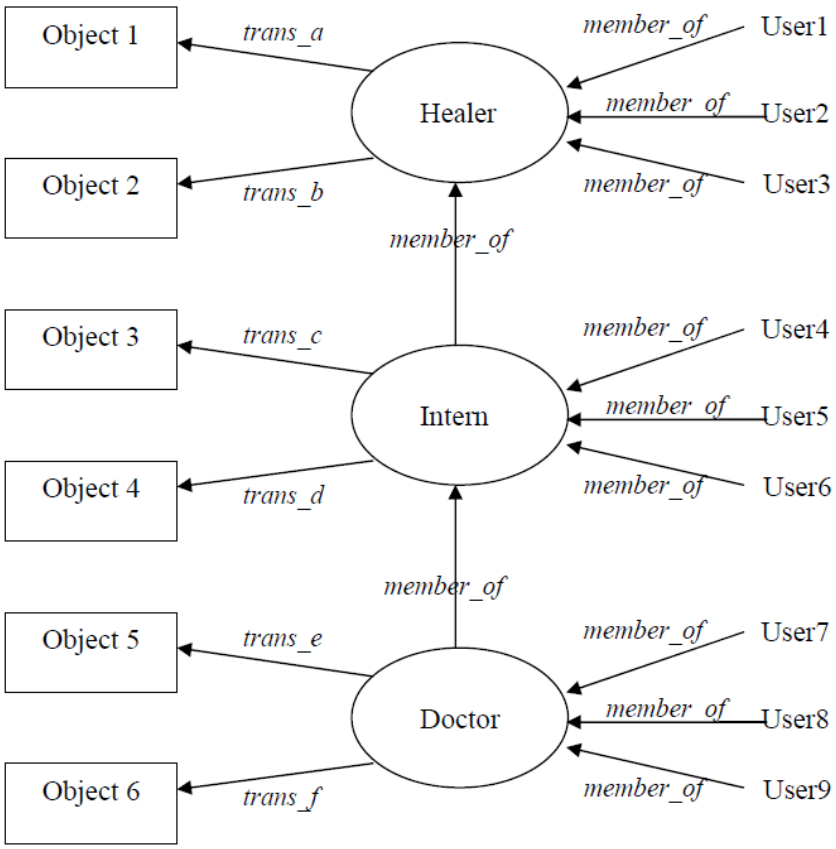
dovoljenja. Hierarhični model namreč še vedno ne dovoljuje modeliranja pravil SoD.

Omejevani RBAC

Omejevani RBAC na sliki 2.6 je razširitev hierarhičnega RBAC in prinaša možnost definiranja pravil SoD. Pravila so lahko statična ali dinamična: statična so definirana na kontekstu vloge, dinamična na kontekstu uporabnika. Primer statičnega pravila je recimo pravilo, da vloga 1 in vloga 2 ne smeta biti skupaj dodeljeni isti osebi. V primeru kršitve se obe vlogi uporabniku odstranita. Dinamična pravila delujejo drugače, v primeru kršitve zgoraj podanega primera (ko ima uporabnik že dodeljeno vlogo 1 in se mu dodeli vlogo 2) dinamični SoD odstrani vlogo 1 uporabniku in dovoli dodelitev vloge 2. Dinamični SoD je tako nekoliko bolj fleksibilen.

Simetrični RBAC

Omejevani RBAC na sliki 2.7 postavlja pravila SoD na nivoju vloge, žal to ne rešuje težav, saj lahko uporabnik še vedno dobi nabor pravic, ki niso združljive, čeprav vloge same po sebi niso problematične. Simetrični RBAC to težavo rešuje z vpeljavo pravil SoD na nivoju pravic, nad katerimi so definirane vloge. Ta model posledično omogoča vpogled v nabor pravic, dodeljenih

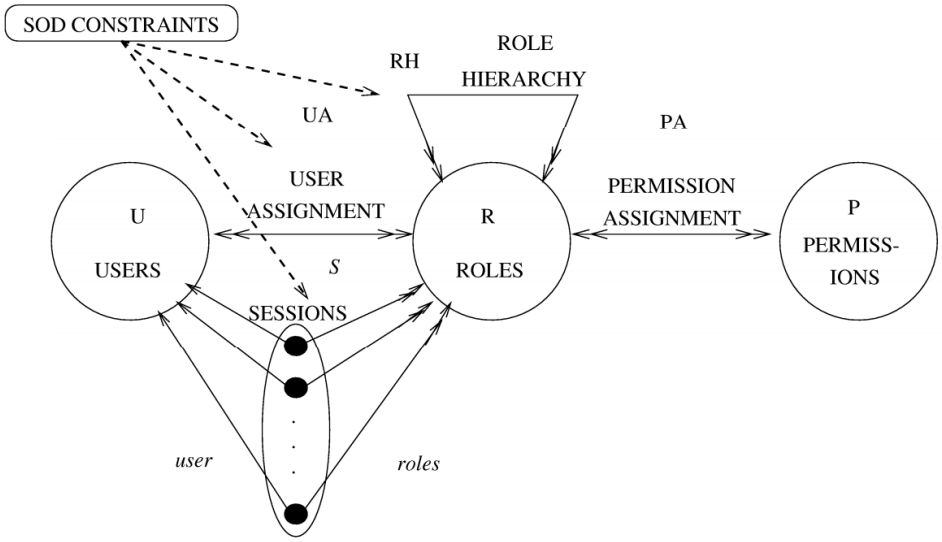


Slika 2.5: Povezave med vlogami v hierarhičnem RBAC. [29]

nekemu uporabniku ali poljubnemu naboru vlog.

2.4.3 Revizija in potrjevanje dostopov

Uporabniške dostope do sistemov je potrebno pregledovati in potrjevati sproti (ko jih uporabniki zahtevajo) in/ali periodično - tipično vsaj enkrat letno. Najpogostejši način preverjanja so *periodične kampanje* (angl. *attestation*), ki so lahko omejene na določeno organizacijsko enoto (predmet revizije so vsi uporabniki znotraj organizacijske enote in njihova dovoljenja do sistemov) ali



Slika 2.6: Shema omejevanega RBAC. [29]

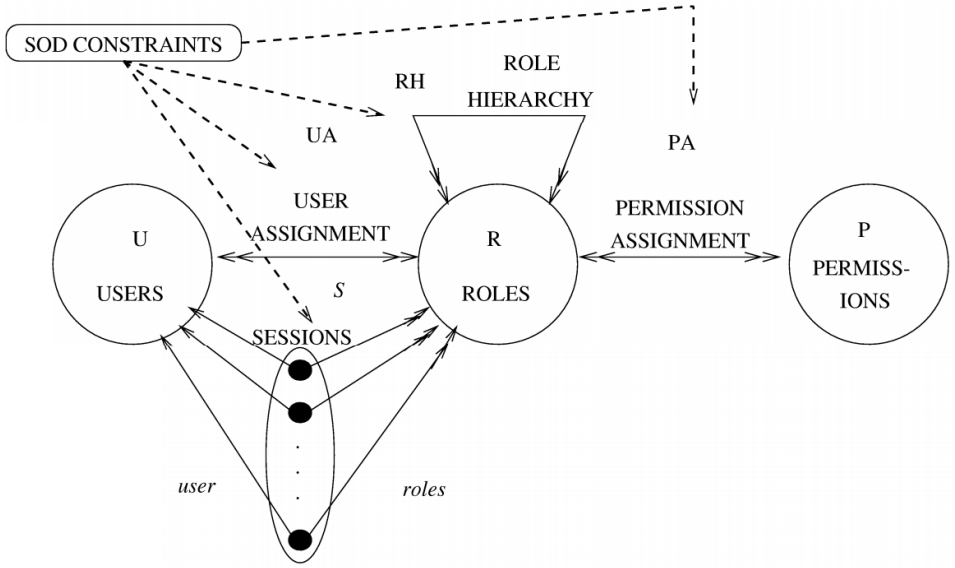
sistem (predmet revizije so vsi uporabniki sistema ter njihove vloge). Cilji revizije so sledeči:

- pregled vlog, dodeljenih uporabnikom;
- analiza tveganj, povezanih z dostopi, ki jih imajo uporabniki;
- identifikacija dostopov, ki jih uporabniki ne potrebujejo več.

Potrjevanje dostopov je lahko večnivojsko. V sklopu kampanje dostop lahko najprej potrdi uporabnik sistema ali njegov nadrejeni in nato še upravitelj sistema. Če je ugotovljeno, da uporabnik dostopa ne potrebuje več, se dostop do sistema ukine, ko je kampanja zaključena.

2.4.4 Delegiranje dostopov in zadolžitev

Delegiranje dostopov se uporablja v primerih, ko oseba s posebnimi odgovornostmi (administrativni dostop do sistema, potrjevanje dopustov podrejenim



Slika 2.7: Shema simetričnega RBAC. [29]

itd.) ni dosegljiva za daljši čas in mora tretja oseba prevzeti te odgovornosti v vnaprej dogovorjenem časovnem okviru. Delegirati je možno le dostope in pravice, za katere je vnaprej definirano, da so primerni za delegiranje. V procesu delegacije dostopov je možno delegirati le posamezno vlogo ali zadolžitev. Vsaka sprememba zadolžitev je zabeležena v revizijski sledi, če ciljni sistem podpira naprednejše scenarije, se vsaka akcija delegiranega uporabnika ustrezno beleži (npr. izvedel uporabnik B v imenu uporabnika A).

2.4.5 Samooskrba preko standardnih zahtevkov

Standardni zahtevki končnim uporabnikom omogočajo izvedbo nekega standardnega procesa, ki ga je možno avtomatizirati. Najpogostejša avtomatizirana opravila so ponastavljanja gesel v sistemih. Gartner [27] v študiji ocenjuje, da en zahtevek za ponastavitev gesla, ki ga mora izvesti služba za podporo uporabnikom, stane od 14 do 28 dolarjev. Eden večjih proizvajalcev pijač je na ta način prihranil več kot 600.000 dolarjev v prvem letu uvedbe

avtomatizacije ponastavljanja gesel.

Pri uvedbi standardnih zahtevkov se srečamo tudi s tveganji. Največje tveganje je, da brez ustreznega nadzora in preverjanja kvaliteta vnesenih podatkov s strani končnih uporabnikov upade. Standardni zahtevki lahko služijo tudi kot smer napada (npr. ponastavljanje gesla za elektronsko pošto), dodatno skrb je potrebno posvetiti temu, da jih ni možno zlorabiti (z uvedbo večfaktorske avtentikacije, vprašalnikom ali drugim načini, s katerimi uporabnik dokaže svojo istovetnost).

Prav tako je pomembno, da so avtomatizirani procesi za samooskrbo preprosti in intuitivni za uporabo (v nasprotnem primeru morajo uporabniki kontaktirati službo za podporo uporabnikom, kar izniči prednost samooskrbe). Nenazadnje je, ko se ugotavlja smiselnost vpeljave avtomatiziranih procesov, pomembno tudi ekonomsko vprašanje: ali dodatno delo z uvedbo in nadaljnjim vzdrževanjem rešitve odtehta slabosti trenutne ureditve.

2.4.6 Eskalacija zahtevkov

Eskalacija zahtevkov je mehanizem, ki zagotavlja, da se zahtevki uporabnikov po izteku vnaprej definiranega časovnega okvirja preusmerijo k naslednji odgovorni osebi (v primeru večnivojske eskalacije se zahtevek eskalira bodisi vodji bodisi posebni skupini oseb za razreševanje zamujenih zahtevkov). Na ta način poskrbimo, da se vsi zahtevki obdelajo in zaključijo v predpisanem času. Zahtevki se tako v sistemu ne izgubljajo, kvaliteto storitev in organizacije lahko ocenjujemo tudi po tem, koliko eskaliranih zahtevov se zgodi v določenem časovnem obdobju.

2.4.7 Revizijska poročila

Revizijska poročila so sestavni del sistema za upravljanje z identitetami in omogočajo pregled ter nadzor nad identitetami, njihovimi vlogami in *revizijsko sledjo* (katere vloge in dostope je identiteta imela v preteklosti, katere akcijeje

izvršil uporabnik itd.). Zaradi čedalje večje potrebe po skladnosti z internimi pravilniki organizacije in zakonskimi predpisi se tovrstna poročila uporabljajo kot dokazilo o skladnosti.

Poglavje 3

Podporni protokoli z varnostnimi vidiki

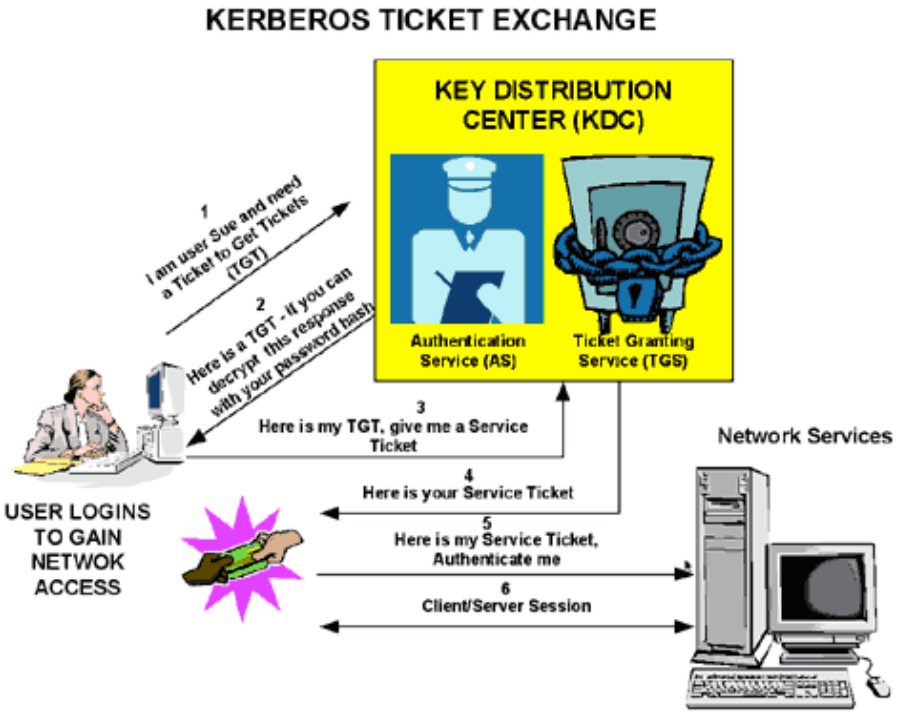
V tem poglavju bomo spoznali ključne podporne protokole, ki zagotavljajo avtentikacijo in varno komunikacijo med dvema entitetama. Protokoli, ki zagotavljajo varnost in integriteto, so z vidika upravljanja identitet nepogrešljivi ter nujno potrebni. Za varno izmenjavo podatkov moramo poskrbeti zaradi njihove občutljivosti (uporabniška imena, gesla itd.).

3.1 Kerberos

Kerberos je omrežni avtentikacijski komunikacijski protokol [10], ki za delovanje uporablja poverilnice (angl. *ticket*) in omogoča komunikacijo (ter dokazovanje istovetnosti identitete) med vozlišči preko nezavarovanih povezav. Sam protokol je bil v osnovi razvit za komunikacijo med strežniki in odjemalci ter zagotavlja medsebojno avtentikacijo - tako strežnik kot odjemalec medsebojno preverjata identiteto. Sporočila, ki se med komunikacijo izmenjujejo, so odporna na prisluškovanja na komunikacijskem kanalu, prav tako jih ni možno posneti in preprosto ponovno predvajati (angl. *record and replay*).

Avtentikacija poteka tako, da se odjemalec predstavi strežniku za avtenti-

kacijo (angl. *Authentication Server*, glej korak št. 1 na sliki 3.1), ki posreduje uporabniško ime distribucijskemu centru ključev (angl. *Key Distribution Center*). KDC nato izda poverilnico (angl. *Ticket Granting Ticket*, glej korak št. 2 na sliki 3.1), ki je podpisana s časovnim žigom. Poverilnica se nato kriptira z uporabnikovim geslom in vrne odjemalcu v kriptirani obliki. Poverilnica se tipično izda ob prvi prijavi v domeno, TGT poverilnica je časovno omejena in jo je potrebno obnoviti - praviloma je ta postopek obnove končnemu uporabniku neviden.



Slika 3.1: Shema Kerberos avtentikacije. [10]

Ko se želi odjemalec povezati z nekim vozliščem v omrežju, pošlje poverilnico storitvi TGT za potrjevanje poverilnic (angl. *Ticket Granting Service*, glej korak št. 3 na sliki 3.1), ki poverilnico overi in izda vstopnico ter sejni ključ. Storitev TGT oboje vrne odjemalcu (glej korak št. 4 na sliki 3.1), ki vstopnico pošlje storitvenemu strežniku (angl. *Service Server*, glej korak št. 5

na sliki 3.1) skupaj s storitvenim zahtevkom (angl. *Service Request*).

3.1.1 Varnostni vidiki

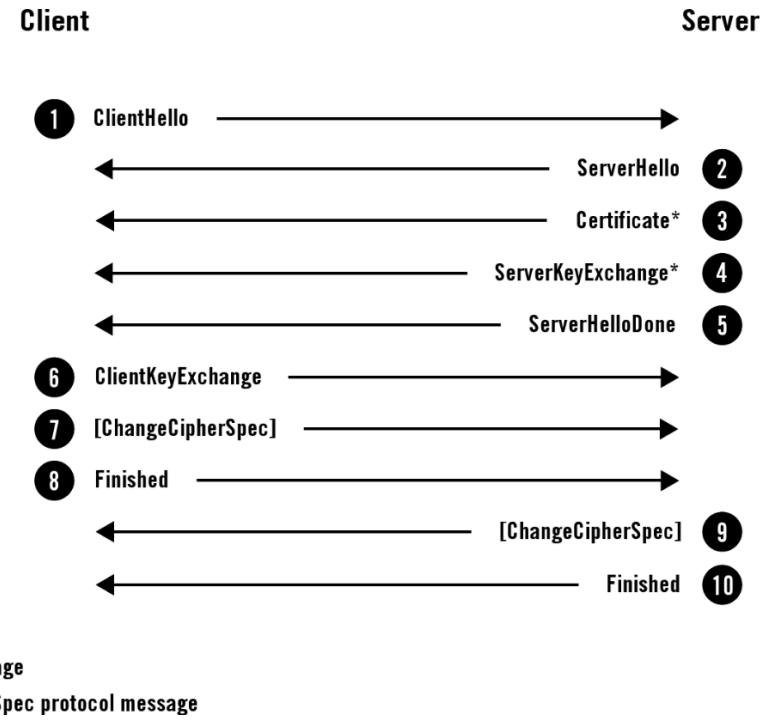
Kerberos protokol ima vrsto omejitev in pomanjkljivosti, ki omejujejo njegovo uporabnost. Protokol predvideva, da si odjemalci, uporabniški računi in storitve na strežniku med seboj zaupajo - se nahajajo v Kerberos domeni zaupanja. Na internetu so najpogostejši scenariji uporabe, ko uporabniki uporabljajo storitve z neznanih odjemalcev, ki jim ne gre zaupati. Protokol zahteva tudi časovno usklajenost med odjemalci in strežniki, poverilnice ter vstopnice so namreč časovno žigosane in imajo omejen rok trajanja. Privzeto se uri med odjemalcem in strežnikom ne smeta razlikovati za več kot 5 minut. To omejitev je sicer, odvisno od uvedbe, možno premostiti.

V primeru uporabe simetričnih kriptografskih rešitev za namene avtenticiranja, lahko vdor v distribucijski center ključev KDC ogrozi varnost vseh odjemalcev, saj se lahko napadalec lažno predstavi kot katerikoli uporabnik [17], poleg tega je tovrstni tip napada težko odkriti, saj je vsa komunikacija povsem legitimna. To pa ni edina pomanjkljivost. Ker je časovna usklajenost ključnega pomena, lahko napadalec zlorabi NTP protokol za sinhronizacijo časa [13] in ponovno predvaja pretečene vstopnice. Kerberos je prav tako ranljiv za ugibanje gesel končnih uporabnikov. Napadalec lahko posname vstopnice in jih poskuša dekriptirati, pri čemer izkorišča dejstvo, da večina uporabnikov ne uporablja dolgih kompleksnih gesel, ki bi razbijanje otežila.

3.2 Varnost na transportni plasti

TLS (angl. *Transport Layer Security*) je kriptografski protokol in je nastal iz uvedbe SSL, ki ga je leta 1996 izdalo podjetje Netscape. Na voljo je več verzij protokola, ki se uporabljajo v spletnih brskalnikih, poštnih odjemalcih, tehnologijah za takojšnje sporočanje (angl. *Instant Messaging*) in povsod, kjer je potrebna varna komunikacija med strežnikom ter odjemalci.

Pričetek komunikacije se začne z rokovanjem odjemalca in strežnika [26], s tem se vzpostavita skrita ključa na obeh straneh, s katerima se vrši nadaljnja komunikacija. Odjemalec in strežnik se morata strinjati z verzijo protokola, ki ga bosta uporabila, izbrati morata ustrezne kriptografske algoritme in izvršiti avtentikacijo ter preverjanje digitalnih certifikatov. Prvi korak v postopku rokovanja se začne, ko odjemalec pošlje strežniku sporočilo, v katerem so zbrane informacije o verziji TLS, seznam verzij protokola in seznam kriptirnih algoritmov, ki jih odjemalec podpira. Dodatno se v pozdravnem sporočilu nahajata tudi naključno generirano zaporedje osmih bitov in seznam algoritmov za stiskanje podatkov.



Slika 3.2: Shema rokovanja TLS [26].

Strežnik TLS se na pozdravno sporočilo odjemalca (glej korak št. 1 na sliki 3.2) odzove s svojim pozdravnim sporočilom, v katerem se nahajo informacije o izbranem kriptografskem algoritmu, ki ga je je odjemalec ponudil, oznaka

seje in naključno generirano zaporedje osmih bitov. Strežnik pošlje tudi svoj digitalni certifikat (glej korak št. 3 na sliki 3.2). V primeru da strežnik zahteva digitalne certifikate za komunikacijo z odjemalcem, slednjemu pošlje zahtevo za digitalni certifikat, ki vsebuje seznam podprtih tipov certifikatov in seznam CA, ki jih strežnik priznava.

Odjemalec TLS nato preveri strežnikov digitalni certifikat. Po uspešnem preverjanju odjemalec pošlje zaporedje naključno generiranih osmih bitov, ki omogoči odjemalcu in strežniku izračun skrivnega ključa, ki se uporablja za kriptiranje komunikacije (glej korak št. 6 na sliki 3.2). Zaporedje bitov je kriptirano s strežnikovim javnim ključem. V primeru da je strežnik poslal zahtevo za digitalni certifikat, mora odjemalec poslati ali svoj digitalni certifikat ali odgovor, da odjemalec ne razpolaga z digitalnim certifikatom. Strežnik se lahko na podlagi odgovora (in realizacije protokola) odloči in prekine z rokovanjem v primeru, da je avtentikacija odjemalca z digitalnim certifikatom obvezna. TLS strežnik nato preveri odjemalčev digitalni certifikat.

Odjemalec TLS pošlje sporočilo, kriptirano s skrivnim ključem, s katerim naznani, da je rokovanje s strani odjemalca zaključeno. Strežnik TLS kot odgovor pošlje zadnje sporočilo v sklopu rokovanja, kriptirano s skrivnim ključem, s katerim naznani, da je rokovanje s strani strežnika zaključeno (glej korak št. 10 na sliki 3.2). Vsa preostala komunikacija se znotraj TLS seje kriptira simetrično z vzajemnim skrivnim ključem.

3.2.1 Varnostni vidiki

Ker protokol TLS predstavlja pomembno tehnologijo za zaščito komunikacij, ostaja pod drobnogledom tako napadalcev kot tudi varnostnih analitikov. Na voljo so različni seznami najpomembnejših ranljivosti in smeri napadov [30]. V tu obravnavanem primeru se bomo dotaknil najpogostejših ranljivosti in predstavili vidike, ki vplivajo na varnost komunikacije.

Najpogostejši primer napada je uporaba tehnik socialnega inženiringa, s katerimi napadalec prepriča uporabnika, da na začetku rokovanja sprejme ne-

veljaven ali ponarejen strežniški certifikat. Sam napad se osredotoča na uporabnike. Tem vrstam napadov se v praksi poskušamo izogniti z izobraževanjem slednjih, da prepoznajo grožnjo in preprečijo komunikacijo po komunikacijskem kanalu, ki ni varen. V nadzorovanih okoljih (npr. interna omrežja podjetij), je mogoče ta problem bistveno omiliti z ustrezno varnostno politiko, ki uporabnikom preprečuje možnost sprejemanja neveljavnih strežniških certifikatov.

Zelo pogosta smer napadov je, ko se strežnik strinja z uporabo starejše verzije protokola TLS/SSL, ki vsebuje znane varnostne pomanjkljivosti, ki jih je možno praktično izkoristiti za prisluškovanje izmenjavi podatkov med odjemalcem in strežnikom po zaščitenem komunikacijskem kanalu. Primer takega napada je napad DROWN [20]. Proti tem vrstam napadov se branimo z ustreznimi sistemskimi nastavitvami na strežniku TLS, ki preprečujejo, da bi strežnik smel uporabljati verzije protokola, kriptografske algoritme in generatorjev naključnih števil, za katere je znano, da vsebujejo kritične varnostne pomanjkljivosti.

Ne glede na to, kako je nek varnostni protokol varen, se v uvedbah vedno lahko pojavi pomanjkljivost, ki jo napadalec izkoristi. Na črnem trgu je mogoče kupiti informacije o varnostnih pomanjkljivostih, za katere proizvajalec še ne ve (angl. *zero-day vulnerability*). Te je nato možno uporabiti za napad na sisteme, ki uporabljajo določeno programsko opremo. Za protokol TLS je dosedaj najbolj znana ranljivost Heartbleed [9], ki omogoča branje delovnega pomnilnika strežnika TLS in s tem razkritje skrivnih ključev. TLS dodatno omogoča tudi stiskanje podatkov, kar predstavlja dodatno smer napadov, najbolj znan med katerimi je CRIME [4]. Vrsta napadov, kjer se uporabljajo varnostne ranljivosti, je najbolj problematična, saj se proti ranljivosti za katero proizvajalec ne ve (ali pa jo je uvedel namenoma), zelo težko braniti. Podjetja, ki ponujajo zaprtokodne programske rešitve, ponujajo denarna nadomestila vsem, ki javijo ranljivosti. Pri odprtokodnih rešitvah je koda na voljo za podroben varnostni pregled in analize.

Poglavje 4

Primerjava sistemov OpenIDM in MIM

4.1 OpenIDM

OpenIDM je odportokodni sistem za upravljanje z identitetami[1]. Spisan je v programskem jeziku Java pod licenco CDDL. Ključni lastnosti sistema sta njegova prilagodljivost in razširljivost. Privzeti skriptni jezik v ogrodju je JavaScript, vsi vmesniki so objavljeni preko vmesnikov REST. Na integracijskem nivoju uporablja ogrodje OpenICF [14], ki temelji na ogrodju Identity Connectors, prvotno razvitem v podjetju Sun Microsystems. Na voljo sta plačljiva in brezplačna različica izdelka. S plačevanjem naročnine je na voljo zadnja verzija izdelka, najnovejše nadgradnje in funkcionalnosti, tehnična podpora ter izobraževanja. Z brezplačno verzijo izdelka dobimo predzadnjo verzijo, brez tehnične podpore in izobraževanj. Podjetje RockForge je predstavilo prvo verzijo izdelka leta 2010, zadnja verzija nosi oznako 4.5 in je na voljo plačljivim uporabnikom. Zadnja brezplačna verzija nosi oznako 4.0 in smo jo uporabili v tem delu.

[1] https://forgerock.org/openidm/

4.1.1 Sistemske zahteve

OpenIDM je uradno podprt tako na operacijskih sistemih Linux kot tudi na Windows Server (glej tabelo 4.1). To je pomembna prednost, saj omogoča širšo uporabo sistema in ne omejuje uporabnikov na točno določeno platformo. Podatkovne zbirke, ki jih sistem potrebuje za delo, je možno namestiti na praktično vse najpomembnejše sisteme za upravljanje s podatkovnimi bazami (glej tabelo 4.1).

Sistemske zahteve	
Zahtevana programska oprema	Java 7 ali 8
Operacijski sistemi	Red Hat Enterprise Linux 6.x/7.x CentOS Linux 6.x/7.x Ubuntu Linux 14.04 Windows Server 2008 R2 Windows Server 2012 R2
Podatkovne baze	MySQL verzije 5.5 - 5.7 [2] Microsoft SQL Server 2012 in 2014 Oracle Database 11gR2 in 12c PostgreSQL 9.3 in 9.4 IBM DB2, 10.x

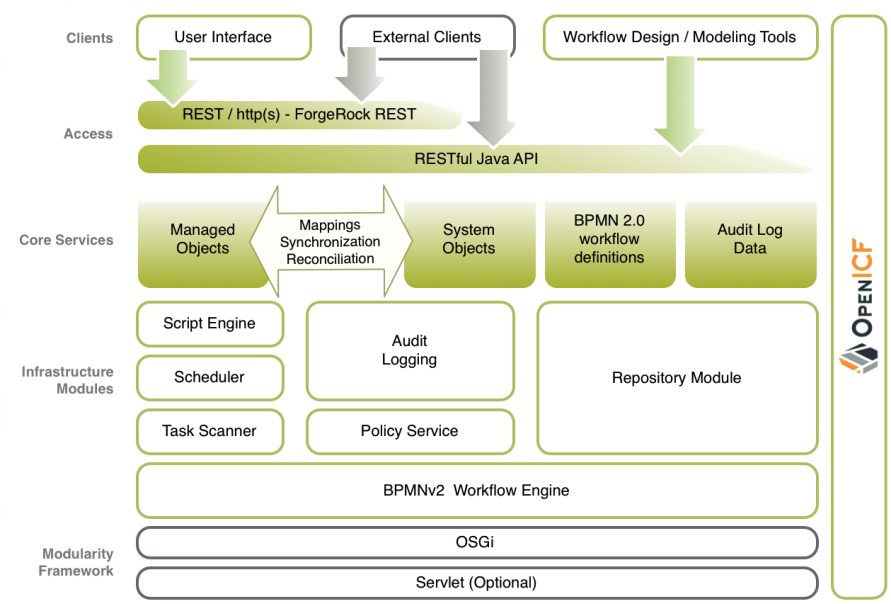
Tabela 4.1: OpenIDM: programske zahteve, podprti operacijski sistemi in podatkovne baze.

4.1.2 Arhitektura

Modularno ogrodje

Ogrodje na sliki 4.1 sestavljata dva podsistema: OSGi služi kot sistemska in storitvena platforma za programski jezik Java, ki uvaja ter podpira dinamične

[2]MySQL JDBC Driver Connector/J 5.1.18 ali novejši



Slika 4.1: Arhitektura sistema OpenIDM. [18]

komponente. OSGi je v OpenIDM realiziran s platformo Apache Felix [3, 15].

OpenIDM omogoča dostop storitev REST (spoznali jih bomo v nadaljevanju) in objektov preko protokola http(s) na nivoju strežniškega programčka (angl. *servlet*), ki se izvaja v Jetty Servlet Container. Ta nivo je opcijski in je na voljo, kadar je potrebno omogočiti dostop do dinamičnega komponentnega modela OpenIDM.

Infrastrukturni moduli

Na sliki 4.1 so prikazani infrastrukturni moduli nad modularnim ogrodjem. Moduli na tem nivoju omogočajo funkcionalnosti, ki so dostopne osnovnim storitvam. OpenIDM podpira izvajanje delovnih tokov BPMN 2.0 in časovni razporejevalnik, ki omogoča periodično izvajanje opravil (npr. sinhronizacije).

Bralnik in razporejevalnik opravil OpenIDM ponuja mehanizem branja in izvajanja ponavljajočih se opravil glede na vrednost podanega atributa. Mehanizem izvede opravilo nad vsemi objekti, ki ustrezajo kriterijem. Razporejevalnik omogoča poganjanje ponavljajočih se opravil, kot so npr. redne sinhronizacije podatkov med sistemi, predobdelave podatkov itd.

Okolje za izvajanje skript Okolje za izvajanje skript trenutno podpira programska jezika JavaScript in Groovy.

Storitev za definiranje politik Gre za modul, ki omogoča proženje dogodkov in kode ob vnaprej predpisanih situacijah ter definiranje politik za validacijo podatkov, ko se objekti in lastnosti objektov ustvarijo oz. spreminjajo.

Revizijska sled Posebna komponenta skrbi za zapisovanje vseh relevantnih sistemskih aktivnosti v za to namenjene dnevniške shrambe. Podatki obsegajo rezultate predobdelav podatkov, detajlirane korake aktivnosti, ki so se izvedle nad nekim objektom, in spremembe, ki so se zgodile bodisi znotraj bodisi izven sistema.

Repozitorij Repozitorij predstavlja osrednji meta imenik, v katerem se shranjujejo podatki in objekti iz različnih sinhronizacijskih virov v objektnem modelu JSON .

Osnovne storitve

Jedro sistema predstavljajo osnovne storitve na sliki 4.1 in se nahajajo nad infrastrukturnimi moduli. Osnovne storitve omogočajo definiranje različnih tipov objektov, preslikav in pravil za sinhronizacijo ter usklajevanje.

Objektni model Vsi objekti, s katerimi upravlja OpenIDM, so javanski objekti, ki predstavljajo objekt, kot je definiran v notaciji JSON. Zaradi te

lastnosti objektni model omogoča povezljivost in integracijo med različnimi aplikacijami, storitvami ter programskimi jeziki. Na nivoju objektnega modela lahko definiramo tudi prožilce in funkcije v podprtih skriptnih jezikih, s katerimi lahko beremo in spreminjamo podatkovne strukture v notaciji JSON. V splošnem ločimo dva tipa objektov. *Upravljani objekti* so tip objekta, ki ga definiramo, da z njim upravljamo neko entiteto (npr. uporabnike, vloge, naprave itd.) v sistemu. *Sistemski objekti* so predstavitev objektov ciljnega sistema, s katerim se OpenIDM povezuje (uporabniški račun v imeniški storitvi, vrstica v tabeli itd.)

Preslikave Preslikave definirajo pravila med izvornimi in ponornimi objekti ter njihovimi atributi med procesi sinhronizacije in usklajevanja. V preslikavah lahko definiramo prožilce za vrednotenja, filtriranje, preslikave in prilagoditve, in sicer tako izvornih kot ponornih objektov.

Sinhronizacija in usklajevanje Usklajevanje podatkov med objekti v repozitoriju in izvornimi ali ponornimi objekti je ključna lastnost sistema za upravljanje z identitetami. Sinhronizacija v OpenIDM omogoča ustvarjanje, posodabljanje ali brisanje zapisov v ciljnem sistemu, bodisi na zahtevo bodisi v skladu z definiranim urnikom izvajanja.

Vmesniki REST Vse storitve, ki jih OpenIDM ponuja navzven, so realizirane z REST (angl. *Representational State Transfer*) arhitekturo. Storitve REST predstavljajo alternativo storitvam SOAP (angl. *Simple Object Access Protocol*), če želimo uporabljati storitve SOAP, moramo na strežniški strani postaviti ustrezno storitveno infrastrukturo, prav tako moramo na odjemalčevi strani poskrbeti, da lahko pokličemo storitve po vnaprej določeni pogodobi (angl. *Service Contract*). Storitve REST so hitrejše, zanesljivejše in zmožne razširitev na več sistemov ter s tem podpirati večje število uporabnikov. Storitve REST striktno ločujejo odjemalca in strežnik, na samem strežniku se ne vzdržuje odjemalčeva seja, odgovornost za vzdrževanje stanja nosi odjemalec.

Strežnik si odgovore na zahtevke lahko zapomni, kar še dodatno izboljšuje zmogljivosti. Zelo pomembna lastnost storitev REST je tudi skrivanje več slojev preko enotnega vmesnika. Zahtevek se preko enotnega vmesnika lahko posreduje naprej v obdelavo, kar omogoča izenačevanje obremenitve (angl. *Load Balancing*) med več strežnikov. Vmesniki REST v OpenIDM omogočajo dostop do upravljanih in sistemskih objektov s pomočjo operacij CRUD, nastavitev sistema, upravljanje uporabniških računov, delovnih tokov in procesov, upravljanje z repozitorijem in nadgrajevanje sistema.

4.1.3 Povezljivost s sistemi

OpenIDM ima v brezplačni verziji na voljo nekoliko omejen nabor povezovalnikov, ki v večini primerov zadostuje. Za razliko od MIM ima OpenIDM en sam povezovalnik za podatkovne baze, znotraj konfiguracije povezovalnika izberemo tip podatkovne baze, do katere želimo dostopati. Podprte so verzije vseh večjih proizvajalcev. Z namestitvijo ustreznih gonilnikov se lahko povežemo na poljubno podatkovno bazo. Pri imeniških storitvah je na voljo povezovalnik LDAP, ki podpira Microsoft Active Directory, Sun Directory Server, Oracle Directory Server, IBM Security Directory Server in imeniške storitve, ki uporabljajo generični protokol LDAPv3. Podprte so tudi oblačne storitve (Google Apps Connector, Salesforce).

V plačljivi različici OpenIDM je privzeto na voljo več povezovalnikov. Ti so na voljo tudi v brezplačni različici, le namestiti jih moramo sami. Postopek namestitve ni pretirano zahteven.

4.1.4 Uvedba poslovnih procesov

Pri upravljanju z identitetami ne rešujemo samo problema povezljivosti z različnimi sistemi in usklajevanja podatkov med njimi. Pomembno vlogo igrajo tudi poslovni procesi, ki omogočajo upravljanje z identitetami (npr. takojšnja ukinitev uporabniškega računa, ponastavljanje gesla, delegiranje zadolžitev v

času odsotnosti ipd.). Za to potrebujemo v sistemu za upravljanje z identitetami ustrezno ogrodje, s pomočjo katerega lahko uvajamo poslovne procese in posebnosti poslovnega okolja.

OpenIDM vsebuje modul za izvajanje delovnih tokov v obliki ogrodja Activiti[3], ki podpira standard BPMN 2.0. BPMN je standard, ki omogoča grafično načrtovanje poslovnih procesov s pomočjo konstruktov (npr. človeška opravila, izvedljive skripte, podpora samodejnemu odločevanju), ki so neodvisni od platforme in predpisani s standardom. BPMN 2.0 je prinesel izboljšave predvsem pri zmožnostih izmenjave informacij med sistemi. Dodatno je definirana tudi semantika izvajanja operacij. Posledično lahko proces BPMN 2.0 izvajamo na kateremkoli stroju za izvajanje poslovnih procesov, ki podpira standard BPMN 2.0. OpenIDM ne omogoča načrtovanja poslovnih procesov v uporabniškem vmesniku, za to so na voljo primernejša orodja, ki jih lahko uporabimo. Precej večja pomanjkljivost sistema je, da ne omogoča izvajanja delovnih tokov na oddaljeni instanci strežnika BPMN. To pomeni, da obstoječih pravil in uvedbe poslovnih procesov v podjetju ne moremo uporabiti.

Ko je poslovni proces zmodeliran in pripravljen za namestitev, ustvarimo datoteko s končnico BAR, ki jo potem namestimo v *workflow* imenik. V datoteki *process-access.json* moramo v formatu JSON dodeliti pravice vlogam, ki imajo dostop in lahko sprožijo poslovni proces.

4.1.5 Procesiranje zahtevkov

Vsi zahtevki se procesirajo preko vmesnika REST, kar pomeni, da mora odjemalec z vsakim zahtevkom podati tudi prijavne podatke. Prijavni podatki se uporabijo v procesu avtentikacije, kjer OpenIDM preveri, da zahteva prihaja s strani legitimnega uporabnika. OpenIDM omogoča prijavo dveh vrst uporabniških računov, in sicer internih ter upravljanih.

OpenIDM ima privzeta dva interna uporabniška računa: anonimni in ad-

[3]https://www.activiti.org

ministratorski uporabniški račun. Oba uporabniška računa sta ločena od ostalih uporabniških računov in ju ne moremo upravljati v OpenIDM (za razliko od MIM, kjer je administratorski uporabniški račun upravljan v MIM in ga lahko ponesreči izbrišemo). Anonimni uporabniški račun omogoča dostop do sistema uporabnikom brez uporabniškega računa. Privzeto ima dodeljeno le vlogo openidm-reg, ki se uporablja za registracijo procesov za samooskrbo. Administratorski račun omogoča popoln dostop do vseh funkcionalnosti sistema in se ga lahko uporabi v primeru, ko dostop do sistema sicer ni mogoč (npr. vsi uporabniki v sistemu so onemogočeni). Upravljani uporabniški računi so vsi ostali računi, ki jih OpenIDM sinhronizira med različnimi sistemi.

Avtentikacija Ob prvem klicu vmesnika REST poskuša sistem izvesti avtentikacijo najprej kot interni uporabniški račun. Če uporabnik s tovrstnim uporabniškim računom ne obstaja, se izvede še avtentikacija upravljanega uporabniškega računa. OpenIDM podpira sejni modul JSON Web Token, ki ob uspešni avtentikaciji ustvari sejo in nastavi piškotek v odgovoru na zahtevo. Ob vsakem nadaljnjem zahtevku se ob prisotnosti piškotka za avtentikacijo preveri njegov digitalni podpis, dekriptira njegova vsebina in preveri, da piškotek ni potekel. V primeru veljavnega piškotka, se klic metode REST izvede brez avtentikacije. OpenIDM poleg avtentikacije uporabniških računov omogoča tudi avtentikacijo s podprtimi imeniškimi storitvami (bodisi LDAP bodisi Active Directory).

Avtorizacija Po uspešni avtentikaciji OpenIDM izvede avtorizacijo zahtevka. Vsi zahtevki REST preko protokola HTTP so podvrženi preverjanju, ali ima uporabnik ustrezno vlogo, ki omogoča izvedbo zahtevane operacije. Mehanizem za avtorizacijo je definiran v dveh JavaScript datotekah: *router-authz.js* in *access.js*. V datoteki *router-authz.js* je definiran mehanizem, ki zagotavlja izvajanje avtorizacijskih pravil, ki so zapisana v datoteki *access.js*. Avtorizacijska pravila predstavlja objekt z lastnostmi podanimi v tabeli 4.2

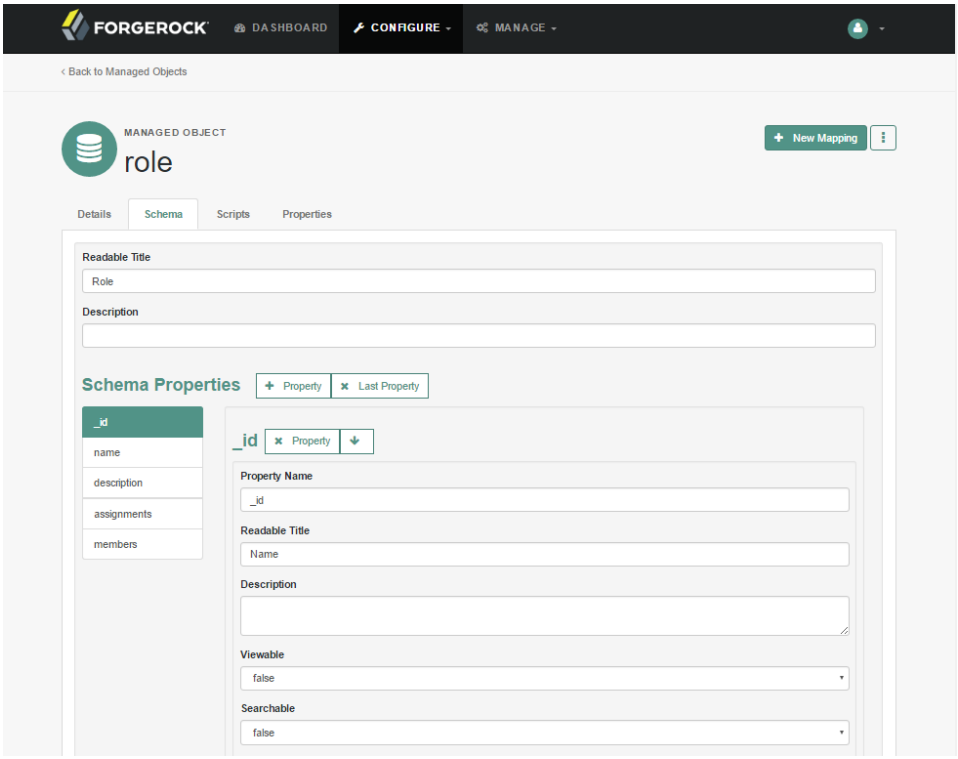
Lastnost	**Opis**
pattern	Vzorec zahtevka, za katerega velja pravilo.
roles	Seznam vlog, ločenih z vejico.
methods	Seznam dovoljenih metod, ločenih z vejico.
actions	Seznam dovoljenih operacij, ločenih z vejico.
customAuthz	Prilagojena funkcija za dodatno preverjanje avtorizacijskih pravil. Neobvezni parameter.
excludePatterns	Seznam vzorcev, za katere je pravilo izvzeto. Neobvezni parameter.

Tabela 4.2: OpenIDM: seznam lastnosti objekta za avtorizacijo.

Z vnosom definicij v datoteko *access.js* lahko definiramo svoja pravila za vloge, s katerimi podpremo dodatno razvite poslovne procese in objekte. V primeru, da programska logika za zahtevek ne najde ustreznega pravila, ki bi dovoljeval izvedbo, logika vrne HTTP napako 403 (zavrnjen dostop). Dodatno velja tudi omeniti, da se validacija zahtevkov izvaja samo za neposredne HTTP zahtevke. Za zahtevke, ki se izvršijo znotraj aplikacijskega strežnika (npr. ob proženju dogodkov), se preverjanje avtorizacijskih pravil ne izvede.

4.1.6 Shema

V administrativnem portalu OpenIDM (glej sliko 4.2) je možno dodajati in urejati definicije objektov, ki so shranjene v notaciji JSON v datoteki *managed.json*. Objektom lahko dodajamo in spreminjamo lastnosti, vsaki lastnosti lahko določimo sistemsko ime, prikazno ime, ali bo vidna na uporabniškem vmesniku, ali bo možno iskati po vrednostih te lastnosti, pravila za validiranje vsebine, podatkovni tip in ali se vsebina lastnosti kriptira (povratno ali nepovratno). Na objektu lahko definiramo metode, ki se sprožijo ob določenih dogodkih (ustvarjanje, sprememba, brisanje, shranjevanje, preverjanje in sinhronizacija objekta).



Slika 4.2: Urejevalnik objektov v OpenIDM

Tovrstni način urejanja objektov ima precej prednosti. Omogoča prožnost, uvedbo kompleksnih poslovnih pravil za vrednotenje in preverjanje ter izvajanje kode JavaScript ob različnih dogodkih. V primerjavi z MIM omogoča OpenIDM popolno kontrolo upravljanih objektov.

4.2 Microsoft Identity Manager

Microsoft Identity Manager 2016 (MIM) je zadnja različica ogrodja, ki omogoča upravljanje z identitetami. Korenine izdelka segajo v leto 2003, ko je podjetje Microsoft izdelalo Microsoft Identity Integration Server 2003 (MIIS). Kasneje se je izdelek preimenoval v Microsoft Identity Lifecycle Manager 2007 (ILM), iz katerega je nato nastal Microsoft Forefront Identity Manager 2010 in

kasneje verzija 2010 R2. Iz te verzije je leta 2016 nastal MIM. Vse tri verzije ostajajo uradno podprte[4], in sicer FIM 2010 in FIM 2010 R2 do leta 2022, MIM 2016 do leta 2026 [12].

MIM omogoča upravljanje z uporabniškimi računi, poverilnicami, varnostnimi politikami in dostopi znotraj organizacije. Poleg tega omogoča integracijo in upravljanje identitet na federiranih sistemih ter oblačnih storitvah. Dodatno podpira tudi koncept priviligiranega dostopa in upravljanja z varnostno kritičnimi vlogami ter dovoljenji - PAM. Upravljanje certifikatov za razliko od FIM podpira več instanc aktivnega imenika (angl. *multi-forest topology*). Samopostrežni procesi podpirajo odklepanje uporabniškega računa in večfaktorsko avtentikacijo za ponastavitev gesla.

4.2.1 Sistemske zahteve

Kot vidimo v tabeli 4.3, je MIM uradno podprt na zgolj dveh operacijskih sistemih proizvajalca Microsoft. Podatkovne baze, ki jih sistem uporablja za svoje delovanje, so skladne zgolj s tremi verzijami podatkovnih strežnikov.

Sistemske zahteve	
Zahtevana programska oprema	SharePoint Foundation 2010 SharePoint Foundation 2013 SP1 Visual Studio 2012 ali 2013[5]
Operacijski sistemi	Windows Server 2012 Windows Server 2012 R2
Podatkovne baze	SQL Server 2008 R2 SP3 SQL Server 2012 SP2 SQL Server 2014 SP1

Tabela 4.3: MIM 2016: programske zahteve, podprti operacijski sistemi in podatkovne baze.

[4]Podpora izdelkom se lahko spremeni brez predhodne najave.

4.2.2 Arhitektura

S stališča arhitekture MIM 2016 sestavljajo sledeče komponente naštete v tabeli 4.4 od katerih vsako lahko namestimo samostojno na ločen strežnik. Komponente prikazane na sliki 4.3 so lahko nameščene na lastne instance strežnikov ali vse na enem samem strežniku, odvisno od načrtovanih kapacitet sistema in števila sočasnih uporabnikov. V tu obravnavanem primeru je sistem MIM nameščen na enem samem strežniku.

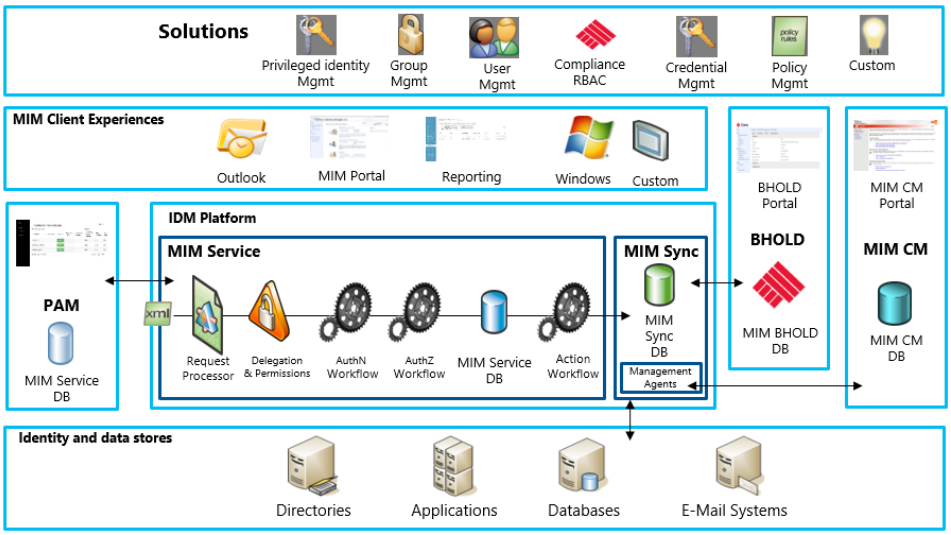
Komponenta	**Opis**
MIM Portal	Uporabniški in konfiguracijski portal.
MIM Service	Aplikacijska storitev za procesiranje zahtevkov.
MIM Synchronization Service	Sinhronizacijska storitev.
MIM Registration Portal	Portal za registracijo uporabnikov.
MIM Password Reset Portal	Portal za resetiranje gesla.
MIM Certificate Manager	Upravitelj certifikatov.
MIM BHOLD	Sistem za dodelitev pravil RBAC.
MIM PAM	Upravitelj privilegiranih dostopov.
MIM Reporting	Revizijska poročila.

Tabela 4.4: Komponente sistema MIM 2016.

4.2.3 Povezljivost s sistemi

Glavni adut MIM je povezljivost z različnimi sistemi. V ta namen ima izdelek bogato podporo imeniškim storitvam, podatkovnim bazam in izmenjavi podatkov preko spletnih storitev ter datotečnih formatov. Posebej zajeten je seznam imeniških storitev, s katerimi lahko integriramo MIM. Te storitve se lahko izvajajo tudi na platformah, ki sicer z Microsoftovim ekosistemom tipično niso nujno združljive. Oblačne storitve (Azure Active Directory, Office

[5]Zgolj opcijsko.



Slika 4.3: Arhitektura sistema MIM [11].

365) se obravnavajo enako kot katerikoli drugi sistem, ki ga integriramo.

Poleg imeniških storitev so podprte tudi podatkovne baze, s katerih lahko beremo ali pišemo informacije. V primeru da namenski povezovalnik ni na voljo, ga lahko z uporabo povezovalnika Extensible Connectivity 2.0 ustvarimo sami. V metodah vmesnika sprogramiramo komunikacijo s sistemom. Če je ta pristop preveč zahteven (potrebna so razvojna orodja), je na voljo povezovalnik PowerShell, s katerim lahko v skriptnem jeziku sprogramiramo komunikacijo s sistemom.

4.2.4 Uvedba poslovnih procesov s pravili MPR

Pravila upravljanja ali pravila MPR (angl. *Management Policy Rule*) definirajo enega ali več pogojev/dogodkov, ki se lahko zgodijo v sistemu. Pravila se uporabljajo tudi za definiranje pravic nad objekti in preslikavo delovnih tokov na dogodke. Ravno zaradi tega jih delimo na dve podvrsti:

- pravila zahtevkov MPR in
- prehodna pravila MPR.

Pravila zahtevkov MPR (angl. *Request MPR*) definirajo dovoljenja za operacije, ki se izvajajo nad objekti. Na voljo so operacije ustvarjanja, branja, spreminjanja in brisanja objekta. Tem operacijam pravimo tudi operacije CRUD (angl. *Create, Read, Update, Delete*). Pravila zahtevkov MPR se definirajo za operacije CRUD in se jih uporablja za zagotavljanje, da ima uporabnik pravice spreminjati objekt v skladu s poslovnimi pravili. Dodatno lahko tovrstna pravila MPR definirajo delovne tokove, ki se izvedejo pred ali po izvedbi operacij CRUD nad objektom. Kako točno ta proces izgleda v praksi, bomo razpravljali v nadaljevanju.

Prehodna pravila MPR imajo definirano množico objektov (angl. *set*), nad katerimi lahko definiramo neko akcijo, bodisi ko se objekt pridruži množici ali ko se ga iz množice izloči. Množice objektov so lahko ročno upravljane (uporabniki ročno dodajajo/odstranjujejo objekte iz množice), kriterijske (vsi objekti, ki ustrezajo kriterijem, so vključeni v množico) ali časovne (posebna različica kriterijskih množic - pogoji so definirani časovno in sistem jih mora periodično preverjati, če še ustrezajo pogojem). Pri tej zvrsti pravil MPR seveda ni možno definirati delovnih tokov za avtentikacijo in avtorizacijo.

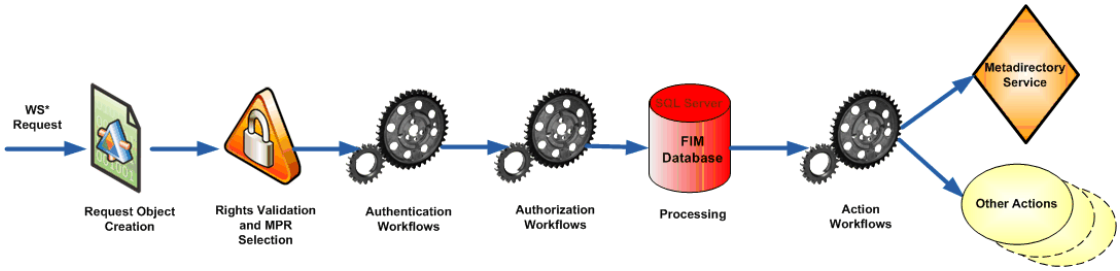
4.2.5 Procesiranje zahtevkov

V nasprotju z OpenIDM ima MIM precej bolj preprosto izdelano ogrodje za procesiranje zahtevkov. Procesiranje poljubnega zahtevka je razdeljeno v tri ključne faze:

- avtentikacija,
- avtorizacija,
- akcija.

Zaporedje procesiranja Zaporedje procesiranja zahtevka je razvidno s slike 4.4. MIM ustvari objekt tipa Request, bodisi zaradi klica spletne storitve

bodisi zaradi zahtevka, ustvarjenega v portalu MIM. Pred nadaljnjo obdelavo se preveri ali je zahtevek v skladu s pravili MPR.



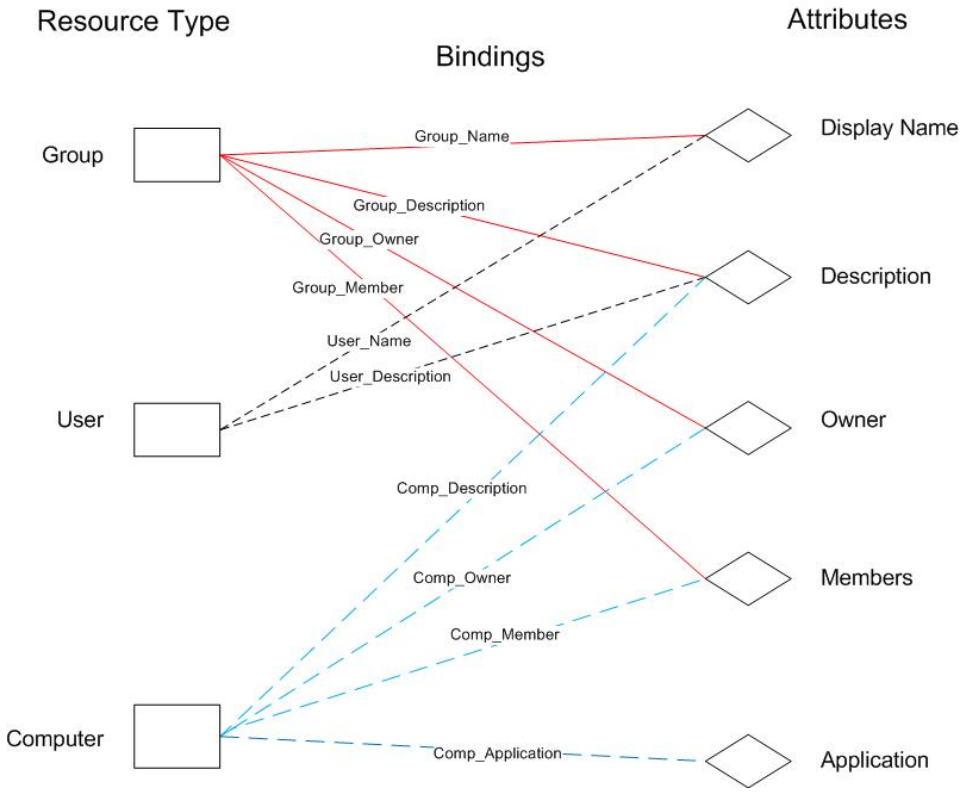
Slika 4.4: Shema procesiranja zahtevkov v MIM. [5]

V objektu Request se nahajajo informacije o zahtevniku, o tipu operacije CRUD, o objektu, nad katerim se mora izvesti akcija, in o atributih objekta. Sistem iz objekta in opisa akcij v zahtevku sestavi seznam vseh pravil MPR, ki pridejo v poštev za procesiranje tega zahtevka. Zahtevek mora ustrezati vsaj enemu pravilu MPR, ki dodeljuje pravice v skladu z zahtevkom, sicer se slednji zavrne.

Ko je zahtevek dokončno preverjen, MIM požene vse delovne tokove za avtentikacijo, ki so bili definirani v naboru ustreznih pravil MPR v nedeterminističnem zaporedju. V primeru, da katerikoli delovni tok ne uspe avtenticirati uporabnika, se procesiranje zahtevka prekine in v revizijsko sled se zapiše razlog prekinitve zahtevka. Ko se vsi delovni tokovi zaključijo, sistem zažene vse avtorizacijske delovne tokove, definirane v naboru ustreznih pravil MPR. Vsi delovni tokovi se izvedejo vzporedno in rezultati izvajanja se ne zabeležijo, dokler se vsi delovni tokovi ne zaključijo uspešno. Sledi faza procesiranja zahtevka in izvajanja operacije nad danim objektom (ustvarjanje, brisanje, spreminjanje atributov objekta ali branje). Po opravljenem procesiranju zahtevka se vzporedno izvedejo vsi akcijski delovni tokovi, definirani v naboru ustreznih pravil MPR. Operacije branja ne podpirajo izvajanja akcijskih delovnih tokov.

4.2.6 Shema MIM

V portalu MIM je možno dodajati in urejati definicije objektov, ki jih potem uporabljamo v delovnih tokovih. Definicije objektov (in pripadajočih atributov) so zapisane v shemi MIM. Shema v portalu MIM sestoji iz treh tipov komponent: objekt, atribut in povezava. Slika 4.5 podaja povezavo med njimi.



Slika 4.5: Shema relacij med atributi, objekti in povezavami [7]

Tip objekta v MIM ustvarimo z novo instanco objekta *ObjectTypeDescription*. Vsak objekt mora imeti sistemsko in prikazno ime, ki se izpisuje na uporabniškem vmesniku. Z atributi in povezavami modeliramo lastnosti objekta. Atributi lahko predstavljajo enega od sledečih podatkovnih tipov, navedenih v tabeli 4.5.

Tovrstni način urejanja objektov ima nekaj prednosti - že na začetku imamo

na voljo nabor najpogosteje uporabljanih atributov (npr. ime, priimek, uporabniško ime), ki jih lahko povezujemo na nove tipe objektov. Slabost pristopa je, da je taka shema precej rigidna, v MIM ne moremo povoziti oz. razširiti osnovne definicije atributa.

Tip	**Opis**
Binary	Predstavlja binarni podatkovni zapis.
Boolean	Predstavlja logično vrednost 1 ali 0.
DateTime	Predstavlja datumsko vrednost.
Indexed String	Predstavlja niz, ki ga je možno indeksirati.[6]
Reference	Predstavlja kazalec na poljuben objekt.
Unindexed string	Predstavlja niz poljubne dolžine, ki ga ni možno indeksirati.

Tabela 4.5: Seznam podprtih podatkovnih tipov za atribute.

[6]Največja dovoljena vrednost je 644 znakov

Poglavje 5

Oskrbovanje uporabnikov v praksi

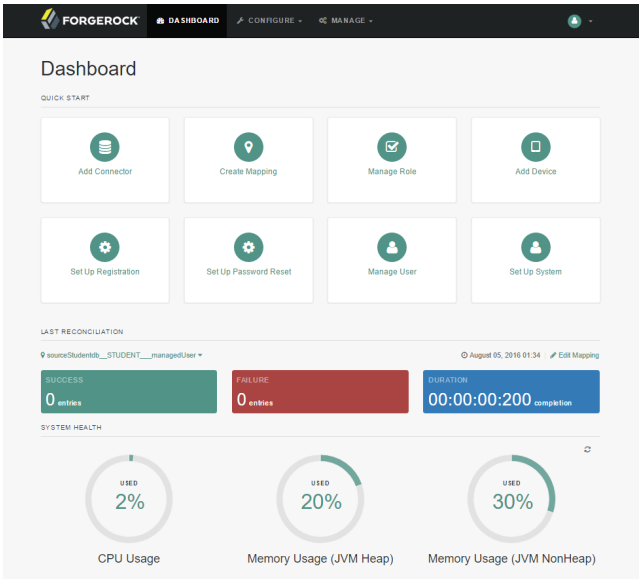
Cilj tega poglavja je uvesti proces oskrbovanja uporabnikov na obeh sistemih in primerjati sistema med sabo z vidika namestitve, razvoja rešitve, prenosa sprememb in vzdrževanja sistema. V ta namen smo pripravili dva navidezna stroja: OpenIDM se izvaja na Ubuntu Server 14.04 LTS, MIM 2016 se izvaja na Windows Server 2012 R2 Standard. Na Ubuntu Server smo uvedli delujoč OpenLDAP strežnik in podatkovno bazo PostgreSQL. Oba navidezna stroja se izvajata na platformi VMware ESXi 5.5.

5.1 Problemska domena

Za izvedbo primerjave smo si izbrali scenarij oskrbovanja redno zaposlene osebe. Proces tipično vključuje branje podatkov iz podatkovne baze kadrovske evidence, v ta namen smo na podatkovni bazi PostreSQL ustvarili podatkovno tabelo, ki simulira sistem kadrovske evidence. Ob uspešno izvedeni zaposlitvi se sklene pogodba, na podlagi katere se v pogledu podatkovne baze zaposlenih pojavi nov zaposleni. Zanj lahko že vnaprej pripravimo uporabniški račun v aktivnem imeniku, na podlagi datuma začetka zaposlitvenega razmerja mu uporabniški račun aktiviramo šele na prvi dan delovnega razmerja.

5.2 Uvedba v OpenIDM

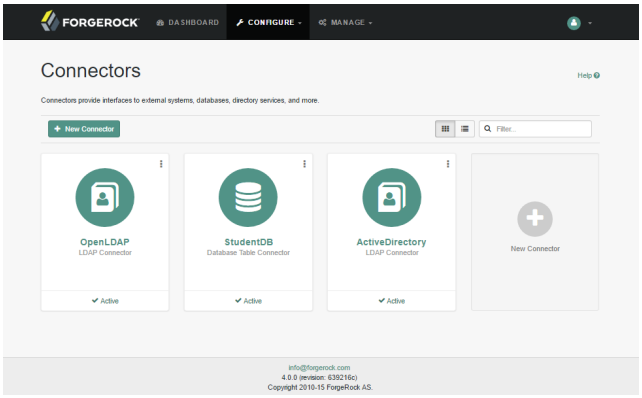
Uvedbe se v OpenIDM lotimo tako, da odpremo spletni brskalnik in odpremo administrativni del portala ter se prijavimo z uporabniškim računom, ki ima administrativne pravice. Na nadzorni plošči (glej sliko 5.1) je zbranih nekaj najpogostejših opravil in kazalnikov stanja sistema.



Slika 5.1: OpenIDM: administrativni vmesnik.

5.2.1 Dodajanje obeh povezovalnikov za podatkovno bazo

Povezovalnik za podatkovno bazo smo dodali z izbiro *Add connector* v meniju. Odprl se je vmesnik za urejanje povezovalnika, v katerem smo izbrali *Database connector* in vpisali vse potrebne informacije za povezavo s podatkovno bazo PostgreSQL ter izbrali *Save Connector Changes*. Postopek smo ponovili, tokrat smo izbrali *LDAP Connector* in znova ročno vpisali vse potrebne informacije za povezavo z imenikom LDAP. V seznamu na sliki 5.2 sta se prikazala oba povezovalnika v statusu *Active*.



Slika 5.2: OpenIDM: seznam povezovalnikov.

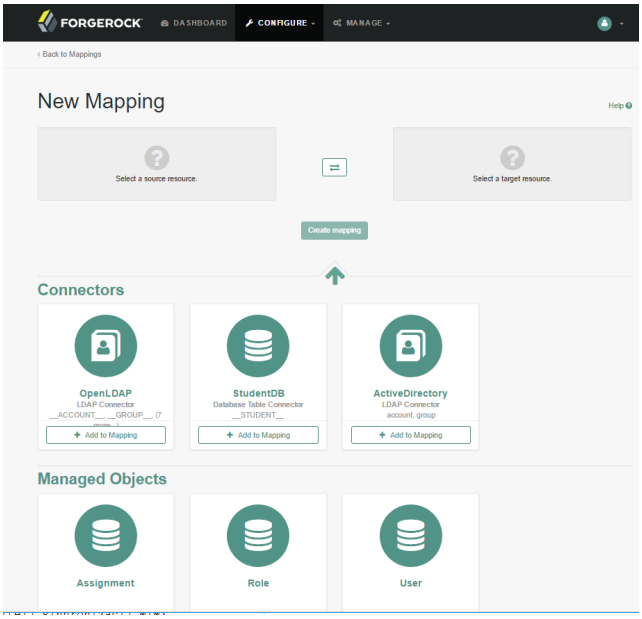
5.2.2 Mapiranje objektov iz podatkovne baze v OpenIDM

Mapiranji iz podatkovne baze v OpenIDM in iz OpenIDM v OpenLDAP smo dodali z izbiro *Create mapping* na nadzorni plošči (glej sliko 5.3). Kot izvorni vir podatkov smo uporabili podatkovno bazo zaposlenih, kot ponor smo izbrali *User* objekt v OpenIDM. V razdelku *Scheduling* smo določili pogostost izvajanja sinhronizacije med sistemoma na enkrat dnevno.

Pri drugem mapiranju smo za izvor uporabili *User* objekt, za ponor smo uporabili OpenLDAP in objekt *account*. V razdelku *Situational Event Scripts* smo za dogodek *onCreate* ustvarili JavaScript skripto, ki pošlje elektronsko pošto nadrejenemu, ko se ustvari uporabniški račun v OpenLDAP.

5.2.3 Generiranje uporabniškega imena

Za generiranje uporabniškega imena smo se odločili, da na upravljanem objektu sprožimo skripto za dogodek *postCreate*, v kateri na podlagi atributov, ki so na voljo, sestavimo unikatno uporabniško ime (tega preverimo s klici metode *openidm.query*, s katero preverimo, da ne obstaja podvojen). Ta postopek načeloma ni najbolj optimalen, saj moramo predhodno preveriti, da upravljani objekt že nima ustreznega uporabniškega računa v OpenLDAP, prav tako ne zagotavlja stoodstotne unikatnosti, saj lahko pride do primerov, ko več objek-



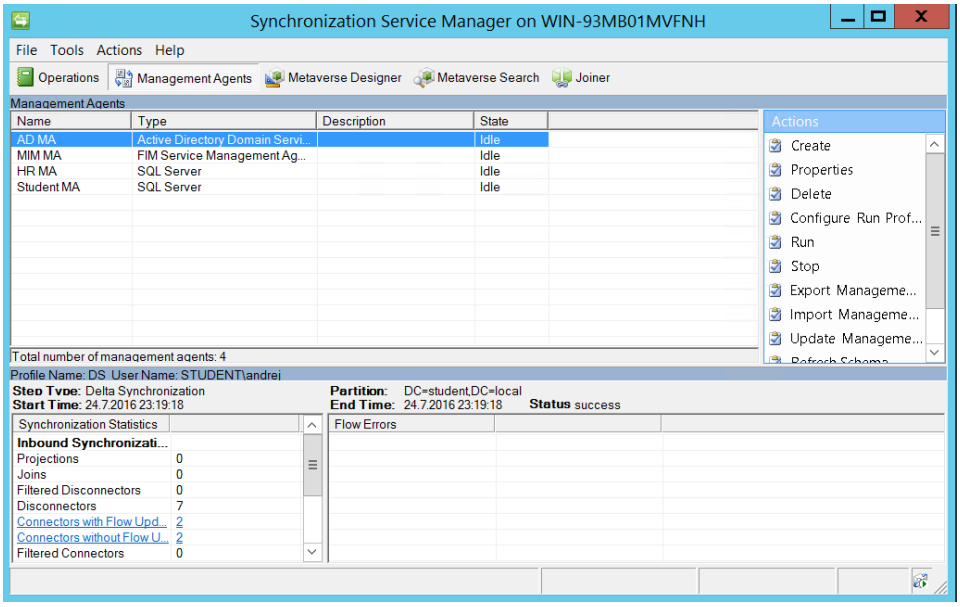
Slika 5.3: OpenIDM: ustvarjanje novega mapiranja podatkov.

tov naenkrat izvrši vpogled v repozitorij in zgenerira identično uporabniško ime.

5.3 Uvedba v MIM 2016

5.3.1 Dodajanje povezovalnika za podatkovno bazo

Ker MIM privzeto ne ponuja upravljalskega agenta za PostgreSQL (na voljo so plačljive verzije, ki jih ponujajo partnerska podjetja), smo se odločili za povezavo preko gonilnikov ODBC. V ta namen smo namestili zadnjo različico 64-bitnih gonilnikov za ODBC [16]. Za uspešno nameščenimi gonilniki ODBC smo namestili generični SQL upravljalski agent, dosegljiv na Microsoftovi podporni strani [8]. Potem smo izdelali datoteko DSN, ki je vsebovala podatke o podatkovni bazi na PostgreSQL podatkovnem strežniku. Po uspešnem ponovnem zagonu sinhronizacijske storitve MIM smo pognali upravitelja sinhronizacij MIM.



Slika 5.4: MIM: upravitelj sinhronizacij.

V upravitelju sinhronizacij na sliki 5.4 smo v razdelku *Management Agents* ustvarili novo instanco upravljalskega agenta z izbiro *Generic SQL (Microsoft)*. Na voljo je čarovnik, ki nas usmerja skozi celotno konfiguracijo upravljalskega agenta. Na koncu je potrebno še ustvariti profile izvajanja. Na voljo so sledeči osnovni profili, ki jih lahko med sabo kombiniramo v poljubnem zaporedju:

Full Import - branje vseh podatkov iz sistema,

Delta Import - branje zgolj podatkov, ki so se spremenili od zadnjega branja,

Full Synchronization - ponovna sinhronizacija vseh objektov v povezovalnem prostoru,

Delta Synchronization - sinhronizacija vseh objektov, ki so se spremenili od zadnjega branja podatkov,

Export - izvoz podatkov v ciljni sistem.

5.3.2 Dodajanje povezovalnika za OpenLDAP

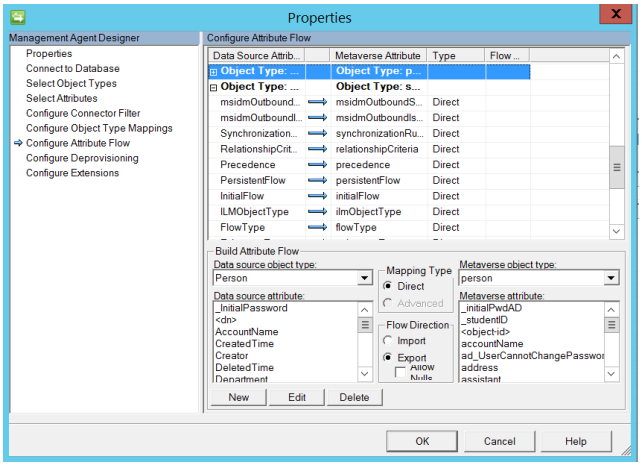
V našem delu smo želeli preveriti integracijo z MIM in imeniško storitvijo OpenLDAP. MIM privzeto ne ponuja upravljalskega agenta za OpenLDAP, z Microsoftove podporne strani smo morali prenesti namestitveni paket za generični LDAP upravljalski agent. Po opravljeni namestitvi je sledil ponovni zagon storitve in upravitelja sinhronizacij. Nato smo ustvarili novo instanco upravljalskega agenta z izbiro *Generic LDAP (Microsoft)*. Tudi v tem primeru nam je bil v pomoč čarovnik, s katerim smo ustrezno skonfigurirali upravljalski agent, na koncu smo poskrbeli tudi za profile izvajanja, s katerimi potem upravljamo z agentom.

5.3.3 Dodajanje povezovalnika za portal MIM

Povezovalnik za portal MIM potrebujemo za izmenjavo podatkov med sinhronizacijsko storitvijo in portalom, v sklopu katerega se izvajajo delovni procesi ter tokovi, ki spreminjajo podatke o identitetah. Te spremembe je nato potrebno sinhronizirati z meta imenikom sinhronizacijske storitve. Z izbiro *FIM Service Management Agent* s seznama razpoložljivih upravljalskih agentov smo ustvarili novo instanco, na kateri smo izbrali naslednje tipe objektov za sinhronizacijo: DetectedRuleEntry, ExpectedRuleEntry, Group, Person in Synchronization Rule. V razdelku *Configure Attribute Flow* (glej sliko 5.5) smo vzpostavili izmenjavo podatkov med meta imenikom in MIM portalom.

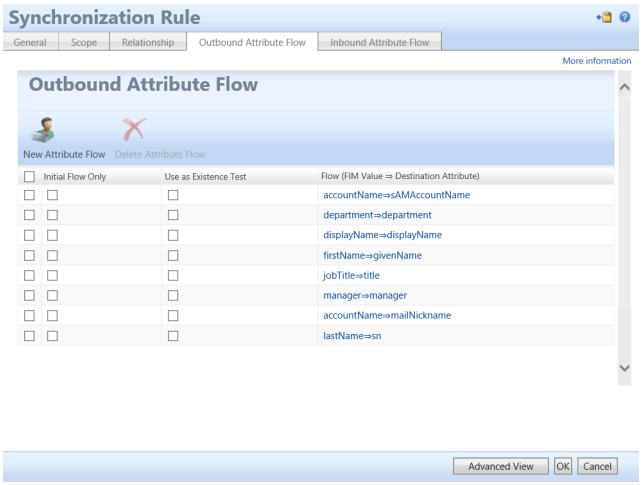
5.3.4 Dodajanje sinhronizacijskega pravila

V Portalu MIM smo pod razdelkom *Administration* izbrali opcijo *Synchronization Rules* in nato opcijo *New*. V na novo odprtem oknu za dodajanje sinhronizacijskega pravila smo morali izbrati upravljalskega agenta in smer sinhronizacije podatkov (samo vhodna, samo izhodna, obojestransko). V razdelku *Relationship* smo ustvarili povezavo med identitetami v meta imeniku MIM in sistemom, za katerega konfiguriramo sinhronizacijsko pravilo. V *In-*



Slika 5.5: MIM: definicija izmenjave podatkov med sistemoma.

bound Attribute Flow in *Outbound Attribute Flow* smo definirali vhodne ter izhodne preslikave atributov med meta imenikom in sistemom (glej sliko 5.6).

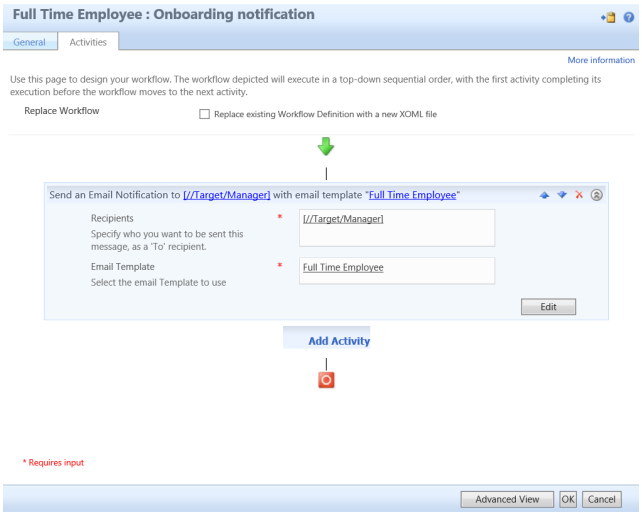


Slika 5.6: MIM: določanje preslikav atributov v sinhronizacijskem pravilu.

Za upravljalskega agenta, ki bere podatke o zaposlenih s podatkovne baze, smo uvedli zgolj vhodno pravilo, ki ustvari zapis v meta imeniku MIM. Za upravljalskega agenta LDAP smo uvedli obojestransko pravilo, saj iz LDAP preberemo atribut *objectSID*, ki omogoča prijavo uporabnikom v portal MIM.

5.3.5 Dodajanje pravila MPR

Preden smo definirali pravilo MPR smo poskrbeli za uvedbo delovnega toka, ki nadrejenemu pošlje elektronsko pošto, ko se za na novo zaposlenega ustvari uporabniški račun v LDAP. V portalu MIM smo pod razdelkom *Administration* izbrali opcijo *Workflows* in nato opcijo *New*. V na novo odprtem oknu za dodajanje delovnega toka smo v razdelku *Activities* dodali aktivnost pošiljanja elektronskega sporočila (glej sliko 5.6).



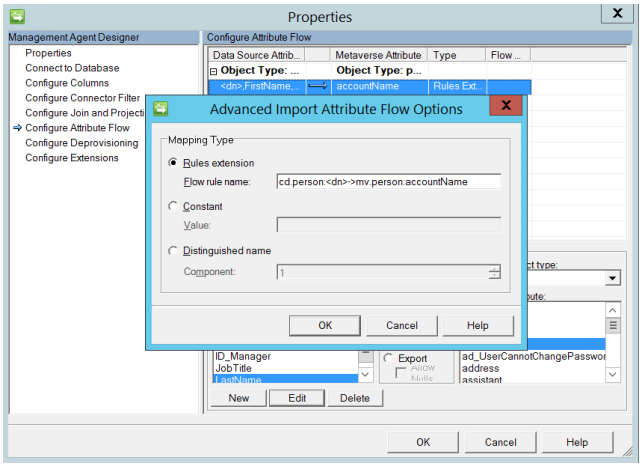
Slika 5.7: MIM: definiranje delovnega toka za obveščanje nadrejenih.

Ustvarili smo tudi ustrezno množico uporabnikov, in sicer z uporabo kriterijske množice. V navigaciji smo izbrali *Sets* in nato *New*. V na novo odprtem oknu smo v razdelku *Criteria-based Members* definirali kriterije, ki veljajo za redno zaposlene.

Šele nato smo v navigaciji izbrali *Management Policy Rules* in nato *New*. V na novo odprtem oknu smo v razdelku *General* izbrali možnost *Set Transition*, v naslednjem razdelku smo vnesli ime množice in izbrali možnost *Transition In*. V razdelku *Policy Workflows* smo izbrali delovni tok za obveščanje, ki smo ga ustvarili na začetku.

5.3.6 Dodajanje razširitve za generiranje uporabniškega imena

Ker iz portala FIM ni možen neposreden vpogled v meta imenik, prav tako ne obstaja že vgrajena aktivnost/delovni tok, ki bi omogočala generiranje unikatnega uporabniškega imena, smo se odločili za uvedbo razširitve neposredno na upravljalskem agentu. V upravitelju sinhronizacij smo izbrali agenta za podatkovno bazo in ga odprli za urejanje. V razdelku *Configure Attribute Flow* smo izbrali atribute, ki jih bomo uporabili pri generiranju uporabniškega imena (ime in priimek), nato smo v razdelku *Mapping Type* izbrali *Advanced*, s čimer smo privzeto izbrali uporabo razširitve (glej sliko 5.8).



Slika 5.8: MIM: razširitev za generiranje uporabniškega imena.

Dialog za urejanje upravljalskega agenta smo nato zaprli, iz menija *Actions* smo izbrali opcijo *Create Extension Projects* in nato *Rules Extension*. V modalnem dialogu smo določili programski jezik, verzijo orodja Visual Studio in lokacijo, kamor smo shranili kodo projekta. V projektu se je nahajal *MAExtensionObject* objekt, ki definira vmesnik *IMASynchronization*. V metodi *MapAttributesForImport* smo ustvarili metodo, ki je iz izbranih vhodnih podatkov zgenerirala unikatno uporabniško ime. Prevedeno kodo smo v obliki datoteke DLL namestil v imenik *Extensions* znotraj imenika, v katerem je

nameščena sinhronizacijska storitev MIM.

5.4 Primerjava obeh sistemov

Opažanja iz zgoraj opisanega lahko razdelimo v štiri kategorije, za vsako posebej bomo podrobno spoznali prednosti in slabosti obeh platform za upravljanje z identitetami.

5.4.1 Sistemske zahteve in povezljivost

OpenIDM podpira širšo paleto operacijskih sistemov in podatkovnih baz, na katere lahko namestimo podatkovne baze, ki jih sistem uporablja. To je zelo dobrodošlo, ko vzpostavljamo sistem v hibridnih okoljih. Na strežnike, z izjemo namestitve ustrezne verzije Jave, ni potrebno predhodno nameščati nobenega dodatnega programja.

MIM je na drugi strani podprt zgolj na dveh verzijah operacijskega sistema Windows Server. Dodatno sta zahtevani tudi namestitev in konfiguracija portala za sodelovanje SharePoint Foundation, ki se pogostokrat izkaže za precej muhastega ter sistemskim administratorjem redno povzroča sive lase.

5.4.2 Namestitev sistema

Namestitev OpenIDM je bistveno lažja, saj namestitveni paket sestavlja zip datoteka, ki se jo zgolj odpakira v poljuben imenik, v katerem se izvaja sistem. Kot pogoj za delovanje OpenIDM mora biti na sistemu nameščena Java 7 ali 8, kot minimalna različica se zahteva Java Standard Edition. Proizvajalec RockForge za delovanje priporoča verzijo Oracle Java Platform, Standard Edition.

OpenIDM privzeto uporablja noSQL podatkovno bazo OrientDB, ki zadostuje potrebam pri razvoju in začetnem testiranju, ni pa primerna za produkcijsko uporabo. Za produkcijsko uporabo je potrebno interni repozitorij

prestaviti na eno od podprtih podatkovnih baz. Proces ni zapleten, sestoji iz izvajanja skripte, ki ustvari ustrezno shemo podatkovne baze in spremembo v konfiguraciji OpenIDM, da kaže na novoustvarjeno podatkovno bazo. Za produkcijsko uporabo je potrebno tudi registrirati OpenIDM kot storitev, kar je ponovno podprto v obliki skript. Open IDM je možno namestiti na široko paleto strežniških operacijskih sistemov in podatkovnih baz, kar je velika prednost. Java je prav tako podprta na praktično vseh pomembnih platformah in ne predstavlja bistvene omejitve.

MIM na drugi strani za namestitev uporablja namestitveni program, ki preko komponente Windows Installer poskrbi za ustrezno namestitev vseh gradnikov sistema. Za slednjo obstaja več pogojev, v vsakem primeru mora obstajati podatkovna baza. Če nameščamo portal MIM, mora biti predhodno nameščen portal za sodelovanje SharePoint, na voljo mora biti več uporabniških računov, ki morajo biti ustrezno konfigurirani. Sam postopek čisto preproste namestitve v razvojnem okolju lahko traja dan ali dva, produkcijska namestitev lahko traja tudi več dni. Če ustvarjamo lastne delovne tokove, moramo namestiti tudi razvojno okolje za .NET, največkrat je to Visual Studio. MIM je možno uporabljati zgolj na dveh različnih operacijskih sistemih podjetja Microsoft, prav tako je tudi omejen na podatkovni strežnik Microsoft SQL Server in portal za sodelovanje SharePoint.

OpenIDM omogoča izjemno preprosto namestitev, ki ne spreminja nastavitev okolja, v katerega ga nameščamo. V primeru da se nam pri razvoju okolje podre (to pri učenju in spoznavanju sistema ni tako redko), ga lahko z nekaj kliki ponovno namestimo. Na istem sistemu je lahko dodatno nameščenih več okolij OpenIDM, kar olajša prototipiziranje in razvoj za več različnih strank. MIM na drugi strani, namestitvenemu programu navkljub, ne ponuja tako pozitivne uporabniške izkušnje pri nameščanju in je veliko bolj invaziven do okolja (ustvari se veliko ključev v registrih sistema, vzpostavi se navidezni imenik v spletnem strežniku IIS, v portalu za sodelovanje SharePoint se ustvarijo knjižnica in nastavitve). Ko je potrebno sistem iz kakršnegakoli razloga

ponovno namestiti je praviloma potrebno vse komponente odstraniti in ponoviti postopek namestitve, kar je časovno izjemno potratno. Poleg tega lahko v enem okolju namestimo samo eno instanco MIM, kar nekoliko zaplete razvoj sistemov za več različnih strank.

5.4.3 Razvoj in uvedba sistema

Kar zadeva razvoj in uvedbo sistema, je potrebno aktivnosti razdeliti v tri kategorije: integracija s sistemi in vzpostavitev toka podatkov iz avtoritativnih virov v ponorne sisteme, uvedba poslovnih procesov na portalu ter prenos sprememb med okolji. V nadaljevanju bomo spoznali prednosti in slabosti obeh sistemov v posameznih kategorijah.

Uvedba pravil za sinhronizacijo podatkov

MIM omogoča konfiguracijo pravil za sinhronizacijo znotraj sinhronizacijske storitve, kjer je za vsak nekoliko bolj specifičen primer potrebno uporabiti kodo, spisano v okolju .NET v obliki dinamično izvedljive knjižnice. Vsako spreminjanje programske logike pomeni programiranje, prevajanje, razhroščevanje in nameščanje nove inačice knjižnice, kar je časovno zamuden postopek. MIM dodatno omogoča še konfiguracijo sinhronizacijskih pravil v portalu MIM, ki deluje po približno istem načelu kot v OpenIDM (konfigurira se prenos atributov med meta imenikom in povezanimi agenti). Popravljanje teh pravil je lažje in hitrejše, vendar žal ne omogočajo izvajanja poljubne kode, temveč le nabor vnaprej predpisanih funkcij, kar nekoliko omeji uporabnost ter razširljivost pravil. OpenIDM je v tem primeru precej bolj napreden, konfiguracija vseh pravil se vrši v spletnem vmesniku, pravila prenosov atributov je možno razširiti s poljubno kodo, spisano v programskem jeziku JavaScript, iz katerega lahko kličemo vmesnike REST OpenIDM.

Dodatno ima MIM pri sinhroniziranju podatkov to pomanjkljivost, da ni možno preveriti delovanja pravil na vseh objektih (to lahko storimo zgolj na

posameznem objektu). OpenIDM omogoča izvajanje sinhronizacije v predogledni obliki - vse spremembe so zgolj nakazane in se ne zapišejo v repozitorij, kar je z vidika testiranja ter preverjanja pravilnosti izvajanja precej dobrodošlo.

Uvedba poslovnih procesov

Pri uvedbi poslovnih pravil lahko v MIM uporabimo pravila MPR, iz katerih kličemo delovne tokove. Na voljo je nekaj osnovnih delovnih tokov, v urejevalniku lahko definiramo svoje delovne tokove in zaporedje aktivnosti, ki se izvedejo. Dodatno je na voljo tudi knjižnica aktivnosti WAL, ki sicer ni privzeto del namestitvenega paketa MIM, a jo je povsem preprosto naknadno dodati. V kolikor z obstoječimi tokovi in aktivnostmi ni možno zadostiti potrebam poslovnih uporabnikov, je v orodju Visual Studio možno ustvariti povsem prilagojene komponente - te so ustvarjene v dinamično izvršljivi knjižnici, za katero veljajo nekatere omejitve v zvezi z naknadnim spreminjanjem ter prilagajanjem.

OpenIDM na drugi strani omogoča kreiranje in izvajanje delovnih tokov ter procesov v platformi BPM Activiti. Sama platforma je veliko bolj dozorela in omogoča višjo stopnjo prilagodljivosti pri uvedbi poslovnih procesov, na voljo je tudi opcija uvedbe delovnih tokov v programskem jeziku Java. Delovne tokove in poslovne procese za OpenIDM načrtujemo v namenskih aplikacijah ter jih nato ročno kopiramo v obliki datotek BAR. Prav tako moramo z ročnim vnosom v ustrezno konfiguracijsko datoteko poskrbeti tudi, da imajo ustrezne vloge dostop do procesov. V MIM za standardne delovne tokove ne potrebujemo dodatnih orodij, vso konfiguracijo lahko opravimo z uporabniškim vmesnikom.

Prenos sprememb med okolji

Pomemben vidik razvoja sistema je tudi mehanizem, ki omogoča prenos sprememb iz razvojnega sistema na testno in kasneje na produkcijsko okolje. Ta

mehanizem se uporablja tako za prenos začetnih sprememb kot tudi sprememb, ki jih prenašamo kasneje, ko je okolje že nekaj časa v uporabi.

MIM podpira migracijo konfiguracije sinhronizacijske storitve in konfiguracijo portala MIM (podatkovna shema, pravila MPR, nastavitve portala) v obliki datotek XML. Pri migraciji konfiguracije sinhronizacijske storitve je ob prvi migraciji potrebno ponovno konfigurirati nastavitve povezovalnih agentov, tipično so povezave na sisteme v novih okoljih različne od prejšnjih. Pohvalno je, da ob nadaljnih migracijah nastavitev sistem ohranja nastavitve, ki so odvisne od okolja. Sinhronizacijska storitev omogoča tudi izvoz/uvoz nastavitev posameznega povezovalnega agenta, kar je tudi dobrodošla funkcionalnost. Spremljajoče dinamične knjižnice, ki so bile razvite in nameščene v ustrezne imenike, je potrebno prestaviti ročno.

Konfiguracijo portala MIM se izvaža/uvaža s pomočjo PowerShell skript, najprej je potrebno posodobiti shemo, nato pravila MPR in nastavitve portala. Za razliko od nastavitev sinhronizacijske storitve je potrebno biti pozoren tudi na kakršnekoli okoljske nastavitve, saj jih med migracijo lahko povozimo. Temu se lahko izognemo tako, da pred uvozom konfiguracijske datoteke ročno popravimo vrednosti (priporočljivo) ali jih po uspešnem uvozu ponovno nastavimo na stare vrednosti v portalu MIM. Po prvem uvozu nastavitev med dvema okoljema lahko naknadne spremembe združujemo z orodjem FIM Delta [6], ki omogoča grafični pregled nastavitev, ki jih bomo uvozili v novo okolje. Prav tako kot v primeru sinhronizacijske storitve, moramo vse prilagojeno razvite dinamične knjižnice in spremljajoče konfiguracijske datoteke prenesti ročno med okolji.

Konfiguracijo sistema OpenIDM lahko izvažamo/uvažamo preko administrativnega vmesnika REST. Konfiguracija obsega več datotek v formatu JSON, vsaka datoteka vsebuje točno določeno konfiguracijo (repozitorij, urnik izvajanja, nastavitve beleženja dnevniških zapisov, konfiguracije povezovalnikov itd.). Format podatkov v zapisu JSON olajša združevanje sprememb med dvema okoljema. Zaradi organizacije nastavitev po datotekah, lahko izberemo

omejen nabor konfiguracijskih datotek, ki jih bomo uvozili (npr. urnikov izvajanja opravil ne selimo med okolji). Tako kot za MIM tudi za OpenIDM velja, da moramo vse prilagojeno razvite rešitve v Javi (v obliki datotek *.jar) prenašati ročno med okolji. V tem oziru sta sistema povsem enakovredna.

5.4.4 Vzdrževanje in upravljanje sistema

Da nam lahko sistem dobro in dolgo služi, ga je potrebno primerno vzdrževati ter nadgrajevati. OpenIDM podpira nadgradnje iz starejših verzij, sistem je potrebno ročno preklopiti v vzdrževalni način in ročno izvesti zaporedje ukazov ter korakov, s katerimi opravimo posodobitev. Sam postopek je nekoliko zapleten in je odvisen od scenarija (nadgradnja iz starejše verzije ali zgolj namestitev servisnih popravkov).

MIM nasprotno uporablja pakete Windows Installer, ki precej olajšajo nadgrajevanje sistema, saj je potrebno zgolj zagotoviti, da ima uporabniški račun dovolj pravic za izvedbo nadgradnje (dobra praksa je, da nadgradnjo izvajamo z administrativnim uporabniškim računom, s katerim smo izvedli prvotno namestitev sistema). Postopek nadgradnje je precej preprost, pomembno je predvsem, da na enako verzijo nadgradimo tako sinhronizacijsko storitev kot tudi portal, sicer praviloma pride do težav pri komunikaciji med sinhronizacijsko storitvijo in portalom MIM.

Portal MIM sicer omogoča prilagoditev uporabniškega vmesnika skozi stile CSS, da ustreza predpisani korporativni podobi. Prilagoditve uporabniškega vmesnika so možne preko definicij RCDC, vendar ne omogoča dodajanja novih uporabniških obrazcev ali t.i. čarovnika, ki bi uporabnika vodil skozi proces. Precej veliko težavo predstavlja dejstvo, da je portal zgrajen na portalu za sodelovanje SharePoint, ki ga moramo vzdrževati ločeno (popravki zanj niso na voljo v sklopu nadgradenj MIM), kar predstavlja dodaten riziko, saj lahko popravki vplivajo na stabilnost in pravilnost delovanja portala MIM. OpenIDM na drugi strani omogoča koncept predlog, s katerimi lahko določamo vizualni izgled sistema (predloge lahko nato poljubno preklapljamo). Dodatno lahko v

sistemu dodajamo svoje forme in razširjamo uporabniški vmesnik - seveda pri tem ne smemo pozabiti, da je tovrstne spremembe potrebno ročno prestavljati med okolji.

Pri upravljanju sistema so ključni urniki za sinhronizacije podatkov. Tu je OpenIDM v precejšnji prednosti, saj ima vgrajen razporejevalnik opravil, s katerim lahko načrtujemo urnik izvajanja sinhronizacij. MIM tovrstnega vmesnika ne ponuja, sinhronizacijska storitev skriptnim jezikom omogoča dostop do upravljalskih agentov, na katerih lahko poženemo ali ustavimo določen profil. Urnik sinhronizacij moramo tako ročno spisati s pomočjo skriptnega jezika in ga registrirati v izvajalniku opravil, ki je del operacijskega sistema.

5.4.5 Povzetek

V tabeli 5.1 so zbrani pomembnejši vidiki vseh kategorij, ki so bile opisane v prejšnjih odstavkih. Na podlagi zapisanega lahko vidimo, da je OpenIDM precej boljša rešitev za upravljanje z identitetami. OpenIDM je možno namestiti na več različnih operacijskih sistemov in podatkovnih baz, sam postopek namestitve je prav tako bistveno lažji. Edino področje, kjer OpenIDM nekoliko zaostaja, je posodabljanje izdelka, predvsem ker postopek zahteva več ročnega dela.

OpenIDM	**MIM 2016**
Sistemske zahteve in povezljivost	
Podprtih 5 operacijskih sistemov in 5 podatkovnih baz	Podprta 2 operacijska sistema in 3 podatkovne baze
Namestitev sistema	
Preprosta namestitev, podprtih več instanc na 1 okolju	Zahtevna namestitev, potrebno veliko spremljajočih orodij. Dovoljena samo ena instanca na okolju.
Razvoj in uvedba sistema	
Uvedba pravil za sinhronizacijo podatkov	
Zelo prilagodljiv, možnost skriptiranja, možnost predogleda izvajanja sinhronizacije	Prilagodljiv, nima možnosti skriptiranja, ni predogleda nad izvajanjem sinhronizacije.
Uvedba poslovnih procesov	
Podpora BPMN 2.0, uvedba nekoliko zahtevna	Podpora .NET delovnim tokovom, uvedba prav tako zahtevna.
Prenos sprememb med okolji	
Prenos sprememb nezahteven, možnost avtomatizacije	Prenos sprememb nezahteve, ni možnosti avtomatizacije.
Vzdrževanje in upravljanje	
Zahtevnejši postopek nadgradnje, precejšnja prilagodljivost uporabniškega vmesnika, vgrajen urnik izvajanja opravil	Preprost postopek nadgradnje, omejena prilagodljivost uporabniškega vmesnika, nima urnika izvajanja opravil.

Tabela 5.1: Povzetek primerjav med sistemoma OpenIDM in MIM.

Poglavje 6

Zaključek in ugotovitve

Področje upravljanja z identitetami je navkljub povečani standardizaciji in bolj zrelim izdelkom še vedno zelo raznoliko ter zahtevno področje informacijske tehnologije. Protokoli, ki zagotavljajo avtentikacijo in varno izmenjavo podatkov, so kljub zrelosti ter dovršenosti še vedno ranljivi na napade. Pri tem je pomembno opozoriti na dejstvo, da je večina ranljivosti bolj posledica slabega poznavanja tehnologij tako razvijalcev, sistemskih inženirjev kot tudi končnih uporabnikov. Sami kriptografski algoritmi so v teoriji precej varni, slabe in površne realizacije napadalcem ponujajo možnost napada. Ne smemo pozabiti niti na še vedno prisoten trend namenskega uvajanja varnostnih lukenj z namenom olajšati prisluškovanje vladnim službam. Odprtokodna skupnost je tu v precejšnji prednosti, saj so kritični deli kode veliko bolj pod drobnogledom razvijalcev.

Pri izdelavi diplomskega dela smo se naučili veliko o delovanju in zgradbi sistemov OpenIDM in MIM 2016. Dodatno smo spoznali tudi problematiko oskrbovanja identitet na dveh različnih platformah. Pri OpenIDM smo spoznali prilagodljivost in razširljivost odprtokodne rešitve OpenIDM, s katerimi sistemi se lahko integrira. Sama platforma je stabilna, navkljub temu da je sestavljena iz več različnih odprtokodnih tehnologij, pri čemer ima vsaka svojo filozofijo razvoja in agendo izdajanja verzij. Podporna dokumentacija izdelka

je zelo dobro spisana, na voljo je presenetljivo veliko primerov uporabe, napake je možno sporočati preko njihovega sistema za sporočanje in odpravljanje napak. Še več: javljene napake lahko spremljamo in tudi vidimo, kdaj ter v kateri verziji bodo predvidoma odpravljene. Vse to nakazuje, da podjetje RockForge zelo resno jemlje svoje stranke in namerava ostati na tržišču ter v ostri konkurenci povečati svoj tržni delež ter (p)ostati pomemben igralec na trgu.

Na drugi strani je Microsoftova rešitev, ki se sicer zelo dobro integrira tako z njihovo imeniško storitvijo in njihovimi izdelki kot tudi z oblačnimi storitvami, ki jih zadnje čase zelo agresivno oglašujejo. Žal izdelek ne omogoča tako visoke stopnje prilagodljivosti v primerjavi z OpenIDM. Opažamo, da je podjetje izgubilo tekmo z ostalimi ponudniki rešitev za upravljanje z identitetami in poskuša z ugodnim cenovnim modelom na trgu konkurirati s sicer solidnim izdelkom, ki ga je nekoliko povozil čas (gledano z vidika razširljivosti ter preprostosti razvoja na sami platformi). Trenutno podjetje tudi ni razkrilo nobenih konkretnih načrtov glede prihodnosti MIM. Glede na prevladujoči trend lahko sklepamo, da utegne ponuditi upravljanje z identitetami v oblaku, za izdelek, ki bi ga lahko uporabljali lokalno, pa se bo potrebno ozreti po konkurenci, ki je k sreči ne manjka.

Dodatek A

Povezljivost s sistemi

A.1 Seznam povezovalnikov sistema OpenIDM

Ime	**Opis**
CSV File Connector	Izmenjava podatkov preko datotek, razmejenih z dogovorjenimi znaki.
Database Table Connector	Izmenjava podatkov preko pogledov/tabel vseh pomembnejših ponudnikov podatkovnih baz.
Google Apps Connector	Upravljanje identitet v oblaku Google Apps.
LDAP Connector	Povezava do imeniških storitev preko protokola LDAP.
Salesforce Connector	Tipski povezovalnik za integracijo s sistemom Salesforce.
XML Connector	Izmenjava podatkov preko datotek XML.

Tabela A.1: OpenIDM: seznam podprtih povezovalnikov.

Ime	Opis
Groovy Connector Toolkit	Izmenjava podatkov z izvajanjem skript v skriptnem jeziku Groovy, s katerimi se povežemo s poljubnim sistemom.
Kerberos Connector	Upravljanje s Kerberos principali.
Powershell Connector Toolkit	Izmenjava podatkov z izvajanjem skript v skriptnem jeziku PowerShell, s katerimi se povežemo s poljubnim sistemom.
RACF Connector	Tipski povezovalnik za integracijo s sistemom IBM RACF.
SAP Connector	Tipski povezovalnik za integracijo s sistemom SAP HR.
Java Connector Server	Izmenjava podatkov z izvajanjem Java kode na oddaljenem sistemu.
.NET Connector Server	Izmenjava podatkov z izvajanjem .NET kode na oddaljenem sistemu.

Tabela A.2: OpenIDM: seznam povezovalnikov v plačljivi različici.

A.2 Seznam povezovalnikov sistema MIM

Ime	Opis
Active Directory Domain Services	Podprti: Active Directory 2000, 2003, 2003 R2, 2008, 2008 R2, 2012
Active Directory Lightweight Directory Services (ADLDS)	Podprt na vseh Active Directory sistemih
Active Directory Global Address List (GAL)	Podprte verzije Exchange 2000, 2003, 2007, 2010, 2013
Microsoft Azure Active Directory Connector for FIM 2010 R2	Upravljanje z imeniško storitvijo v oblaku Azure

IBM Directory Server	Podprta verzija: IBM Tivoli Directory Server 6.x
Novell eDirectory	Podprte verzije: 8.7.3, 8.8.5 in 8.8.6
Generic LDAP Connector for FIM 2010 R2	Podprta verzija LDAP v3 [1]
LDAP Data Interchange Format	Izmenjava podatkov preko datotek v formatu LDIF
Oracle Directory Servers	Podprti: Sun Directory Server 6.x, 7.x in Oracle 11
Directory Services Mark-up Language (DSML)	Podprta verzija: DSML 2.0

Tabela A.3: MIM 2016: seznam podprtih povezovalnikov za imeniške storitve.

Ime	**Opis**
Extensible Connectivity 2.0	Podpora sistemom, ki omogočajo spletne storitve ali datoteke za izmenjavo podatkov.
MIM Service	Povezava z Microsoft Identity Manager 2016
IBM DB2 Universal Database	Podprte verzije: 9.1, 9.5 or 9.7; IBM DB2 OLEDB v9.5 FP5 ali v9.7 FP1
Oracle Database	Podprti 10g or 11g [2]
Microsoft SQL Server	Podprti: SQL Server 2000, 2005, 2008, 2008 R2, 2012
Windows PowerShell Connector for FIM 2010 R2	Podprt: Windows PowerShell 2.0 in novejši

[1] Skladen z RFC 4510

[2] Odjemalec mora biti 64-bitni

Connector for Lotus Domino	Podprta: Lotus Notes v8.0.x in v8.5.x
SharePoint Services Connector for FIM 2010 R2	Podprta: SharePoint Server 2013 ali 2016[3]
Connector for Web Services	Podprta: SAP ECC 5.0 ali 6.0; Oracle PeopleSoft 9.1; Oracle eBusiness 12.1
Attribute-Value Pair text file	Izmenjava podatkov preko datotek z atributi in vrednostmi
Delimited text file	Izmenjava podatkov preko datotek, razmejenih z dogovorjenimi znaki
Fixed-Width text file	Izmenjava podatkov preko datotek določene dolžine

Tabela A.4: MIM 2016: seznam podprtih povezovalnikov za ostale sisteme.

[3]User Profile service application (UPA) mora biti omogočena

Slike

Tabele

Literatura

[1] IBM 2015 Cyber Security Intelligence Index [na spletu]. 2015. Dostopno na http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=ST&infotype=SA&htmlfid=SEJ03278USEN&attachment=SEJ03278USEN.PDF [citirano 28. jun 2016].

[2] The inside story of the biggest hack in history [na spletu]. 2015. Dostopno na http://money.cnn.com/2015/08/05/technology/aramco-hack/ [citirano 28. jul 2016].

[3] Apache Felix [na spletu]. 2016. Dostopno na http://felix.apache.org/ [citirano 28. jul 2016].

[4] CRIME: Information Leakage Attack against SSL/TLS [na spletu]. 2016. Dostopno na https://blog.qualys.com/ssllabs/2012/09/14/crime-information-leakage-attack-against-ssltls [citirano 12. avg 2016].

[5] Designing Business Policy Rules [na spletu]. 2016. Dostopno na https://technet.microsoft.com/en-us/library/ff356871(WS.10).aspx [citirano 2. avg 2016].

[6] FIM Delta [na spletu]. 2016. Dostopno na https://github.com/pieceofsummer/FIMDelta [citirano 5. avg 2016].

[7] Forefront Identity Management Schema [na spletu]. 2016. Dostopno na https://msdn.microsoft.com/en-us/library/windows/desktop/ee652458(v=vs.100).aspx [citirano 2. avg 2016].

[8] Generic SQL Connector technical reference [na spletu]. 2016. Dostopno na https://azure.microsoft.com/en-us/documentation/articles/active-directory-aadconnectsync-connector-genericsql/ [citirano 12. avg 2016].

[9] The Heartbleed Bug [na spletu]. 2016. Dostopno na http://heartbleed.com/ [citirano 12. avg 2016].

[10] Kerberos explained [na spletu]. 2016. Dostopno na https://msdn.microsoft.com/en-us/library/bb742516.aspx [citirano 1. avg 2016].

[11] Microsoft Identity Manager Documentation [na spletu]. 2016. Dostopno na https://docs.microsoft.com/sl-si/microsoft-identity-manager/ [citirano 21. jun 2016].

[12] Microsoft Support Lifecycle [na spletu]. 2016. Dostopno na https://www.support.microsoft.com/en-us/lifecycle/search?sort=PN&alpha=Microsoft%20Identity%20Manager&Filter=FilterNO [citirano 5. jul 2016].

[13] New attacks on network time protocol can defeat https and create chaos [na spletu]. 2016. Dostopno na http://arstechnica.com/security/2015/10/new-attacks-on-network-time-protocol-can-defeat-https-and-create-chaos/ [citirano 1. avg 2016].

[14] OpenICF product page [na spletu]. 2016. Dostopno na https://forgerock.org/openicf/ [citirano 27. jul 2016].

[15] OSGi aliance [na spletu]. 2016. Dostopno na https://www.osgi.org/ [citirano 28. jul 2016].

[16] PostgreSQL [na spletu]. 2016. Dostopno na `https://www.postgresql.org` [citirano 12. avg 2016].

[17] Protecting windows networks - kerberos attacks [na spletu]. 2016. Dostopno na `https://dfir-blog.com/2015/12/13/protecting-windows-networks-kerberos-attacks/` [citirano 1. avg 2016].

[18] M. Craig A. Egloff L. Hordos M. Tristl L. Frost M. Jang D. Chikhaoui A. Askåsen, P. Bryan. OpenIDM Integrator's Guide - version 4, 2015. Dostopno na `https://forgerock.org/openidm/doc/bootstrap/integrators-guide/index.html`.

[19] American National Standards Institute. Role Based Access Control, 2004. ANSI/INCITS 359-2004.

[20] Nimrod Aviram, Sebastian Schinzel, Juraj Somorovsky, Nadia Heninger, Maik Dankel, Jens Steube, Luke Valenta, David Adrian, J. Alex Halderman, Viktor Dukhovni, Emilia Käsper, Shaanan Cohney, Susanne Engels, Christof Paar, in Yuval Shavitt. DROWN: Breaking TLS with SSLv2. Objavljeno v *25th USENIX Security Symposium*, 2016.

[21] William E. Burr, Donna F. Dodson, Elaine M. Newton, Ray A. Perlner, W. Timothy Polk, Sarbari Gupta, in Emad A. Nabbus. SP 800-63-1. Electronic Authentication Guideline. Technical report, Gaithersburg, MD, United States, 2011.

[22] Sally Chan. Sarbanes-Oxley: the IT dimension: information technology can represent a key factor in auditors' assessment of financial reporting controls. *Internal Auditor*, 61(1):31–34, 2004.

[23] Waldemar Hummer, Patrick Gaubatz, Mark Strembeck, Uwe Zdun, in Schahram Dustdar. An Integrated Approach for Identity and Access Ma-

nagement in a SOA Context. Objavljeno v *16th ACM Symposium on Access Control Models and Technologies (SACMAT'11)*, 2011.

[24] Elisa Bertino in Kenji Takahashi. *Identity Management: Concepts, Technologies, and Systems*. Artech House, 2010.

[25] ISO. Information technology – Security techniques – A framework for identity management – Part 1: Terminology and concepts. ISO 24760:2011, International Organization for Standardization, Geneva, Switzerland, 2011.

[26] I. Ristic. *Bulletproof SSL and TLS: Understanding and Deploying SSL/TLS and PKI to Secure Servers and Web Applications*. Computers / Security. Feisty Duck, 2013.

[27] Kris Brittain Roberta J. Witty. Automated password reset can cut IT service desks costs. Technical report, Gartner, 2004.

[28] D. Rountree. *Federated Identity Primer*. Elsevier Science, 2012.

[29] Ravi S. Sandhu, David F. Ferraiolo, in D. Richard Kuhn. The NIST model for role-based access control: towards a unified standard. Objavljeno v *ACM Workshop on Role-Based Access Control*, strani 47–63, 2000. Dostopno na `http://doi.acm.org/10.1145/344287.344301`.

[30] R. Holz Y. Sheffer, Porticor. Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS). RFC 7457, Februar 2015. Dostopno na `https://tools.ietf.org/html/rfc7457`.