

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Miroslav Popović

**NADZIRANJE PRISTUPA RAČUNALNIM
SUSTAVIMA ZASNOVANIM NA
USLUGAMA**

MAGISTARSKI RAD

Zagreb, 2006.

Magistarski rad izrađen je na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave Fakulteta elektrotehnike i računarstva

Mentor: prof. dr. sc. Siniša Srbljić

Magistarski rad ima 133 stranice.

Rad br.: _____

Povjerenstvo za ocjenu u sastavu:

1. Akademik prof. dr. sc. Leo Budin - predsjednik
2. Prof. dr. sc. Siniša Srbljić - mentor
3. Doc. dr. sc. Darko Huljenić - Ericsson Nikola Tesla Zagreb

Povjerenstvo za obranu u sastavu:

1. Akademik prof. dr. sc. Leo Budin - predsjednik
2. Prof. dr. sc. Siniša Srbljić - mentor
3. Doc. dr. sc. Darko Huljenić - Ericsson Nikola Tesla Zagreb

Datum obrane: 26. svibnja 2006

ZAHVALA

Hvala mentoru prof.dr.sc. Siniši Srbljiću na velikom, nesobičnom trudu i znanju koje je prenio. Od Vas sam naučio uistinu mnogo.

Hvala Andri Milanoviću, Ivanu Skuliberu, Ivanu Bencu, Dejanu Škvorcu, Danielu Skrobi, Matiji Podravcu i Ivanu Gavranu na korisnim i vrijednim raspravama, komentarima i savjetima te poticajima koje su pružili. Imao sam veliko zadovoljstvo raditi s tako vršnjim ljudima i u tako dobroj okolini.

Hvala Ivanu Žužaku na implementaciji i iznimno dobroj suradnji.

Hvala svim suradnicima i zaposlenicima na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave.

Renee, hvala što si uvijek spremam pomoći. Stjepane, hvala na jakoj kavi. Svim prijateljima, hvala što ste mi i dalje ostali prijatelji, iako se nismo često družili. Hvala svima koje nisam u mogućnosti spomenuti, a na bilo koji način su mi pomogli.

Na kraju, najviše zahvaljujem svojoj obitelji. Uvijek ste mi u životu bili uzor, potpora, motivacija i najveća snaga. Ovaj rad je napravljen za ponos vama i meni. Posebna zahvala upućena je bratu za podršku, brigu, strukovne savjete i probleme koje je dijelio sa mnom tijekom izrade magistarskog rada.

"Znanstvenik se pita mnoga pitanja...od toga je samo mali dio njih znanstvene prirode!"

SADRŽAJ:

1. UVOD	1
2. RAČUNARSTVO ZASNOVANO NA USLUGAMA	3
2.1. MOTIVACIJA KORIŠTENJA USLUGA	4
2.2. ARHITEKTURA ZASNOVANA NA USLUGAMA	5
2.2.1. <i>Osnovna arhitektura zasnovana na uslugama</i>	6
2.2.2. <i>Proširenje arhitekture zasnovane na uslugama</i>	8
2.3. TEHNOLOGIJE WEB USLUGA	10
2.3.1. <i>Tehnologije Web Services usluga</i>	11
2.3.2. <i>Proširenja Web Services tehnologije</i>	12
3. SIGURNOST U RASPODIJELJENIM RAČUNALnim SUSTAVIMA	14
3.1. NAČINI UGROŽAVANJA SIGURNOSTI.....	14
3.2. POSTUPCI ZA USPOSTAVU SIGURNOSTI	18
3.2.1. <i>Kriptografski postupci</i>	19
3.2.2. <i>Autentikacijski postupci</i>	24
3.2.3. <i>Nadziranje pristupa</i>	27
3.2.4. <i>Postupak autorizacije</i>	32
3.2.5. <i>Praćenje korištenja</i>	33
3.3. MODELI NADZORA PRISTUPA	35
3.3.1. <i>Model razlikovnog nadzora pristupa</i>	35
3.3.2. <i>Model nadzora zasnovan na ulogama</i>	36
3.3.3. <i>Model dosljednog nadzora pristupa</i>	37
3.4. NAČELA OBLIKOVANJA NADZORA PRISTUPA	39
3.4.1. <i>Razdvajanje funkcionalnosti</i>	39
3.4.2. <i>Uspostava zaštićene domene</i>	41
4. XML RAZINA SIGURNOSTI USLUGA.....	45
4.1. PRIMJER POTREBE XML SIGURNOSTI.....	45
4.2. XML TEHNOLOGIJE SIGURNOSTI	46
4.2.1. <i>WS-Security</i>	47
4.2.2. <i>Digitalno XML potpisivanje</i>	49
4.2.3. <i>XML kriptiranje</i>	51
4.2.4. <i>SAML autentikacija</i>	54
4.2.5. <i>XACML nadzor pristupa</i>	58
5. SIGURNOST RAČUNALNIH SPLETOVA	60
5.1. SIGURNOST GLOBUS SUSTAVA	60
5.2. CAS SUSTAV	63
5.3. VOMS SUSTAV	64
5.4. PRIMA	66
6. NADZIRANJE PRISTUPA RAČUNALnim SUSTAVIMA ZASNOVANIM NA USLUGAMA	69
6.1. ZAHTJEVI SIGURNOSTI	70
6.2. UPRAVLJAČKI PODACI NADZORA PRISTUPA	71
6.2.1. <i>Podaci o korisniku</i>	72
6.2.2. <i>Podaci o usluzi</i>	73
6.2.3. <i>Odredbe usluga</i>	73
6.2.4. <i>Podaci o pravima pristupa</i>	74
6.3. OSNOVNE FUNKCIONALNOSTI SUSTAVA NADZORNIK	75
6.4. PROTOKOLI RAZMJENE PODATAKA	76
6.4.1. <i>Protokol registracije</i>	77
6.4.2. <i>Pripremanje upravljačkih podataka</i>	78
6.4.3. <i>Protokol autentikacije</i>	81
6.4.4. <i>Protokol ispitivanja upravljačkih podataka</i>	83

6.5. ARHITEKTURA SUSTAVA NADZORNIK	84
<i>6.5.1. UP spremnik.....</i>	<i>86</i>
<i>6.5.2. Registracija</i>	<i>87</i>
<i>6.5.3. Autentikacija.....</i>	<i>88</i>
<i>6.5.4. PP spremnik</i>	<i>90</i>
<i>6.5.5. Pozivatelj.....</i>	<i>91</i>
<i>6.5.6. Zastupnik nadzora pristupa.....</i>	<i>93</i>
<i>6.5.7. Bilježnik korištenja.....</i>	<i>95</i>
7. PROGRAMSKO OSTVARENJE SUSTAVA NADZORNIK.....	97
7.1. SUSTAV MICROSOFT .NET FRAMEWORK.....	97
<i>7.1.1. Posluživanje Web Forms stranica ASP.NET podsustavom</i>	<i>99</i>
7.2. REGISTRACIJA.....	101
<i>7.2.1. Usluga WSRegistration</i>	<i>101</i>
<i>7.2.2. Primjenski sustav MediatorWebRegistration</i>	<i>102</i>
7.3. AUTENTIKACIJA	108
7.4. PP PODSUSTAV.....	111
<i>7.4.1. Ostvarenje ACModule razreda.....</i>	<i>113</i>
7.5. POSTAVLJANJE SUSTAVA NADZORNIK	116
8. ZAKLJUČAK	119
9. LITERATURA.....	121
10. ŽIVOTOPIS	127
11. SAŽETAK	128
12. SUMMARY	129
13. KLJUČNE RIJEČI.....	130
14. DODATAK A	131

1. Uvod

Tržište programske potpore, vođeno zahtjevima poslovanja i korisnika mreže Internet, postavlja velike zahtjeve za brzim razvojem i dinamičkim dostavljanjem programskih funkcionalnosti (engl. *software development and delivery*). Tradicionalni način pružanja programskih funkcionalnosti u obliku programskih proizvoda suočava se s problemom produljenja vremena potrebnog za razvoj, izrgradnju i isporuku novih programskih funkcionalnosti. Model računarstva zasnovan na programskim proizvodima nameće potrebu kupovanja programskog proizvoda (engl. *purchasing*), postavljanja programskog proizvoda (engl. *deployment*) i podešavanja postavljenog programskog proizvoda (engl. *configuration*). Opisani proces unosi znatnu tromost koja prethodi korištenju kupljene programske funkcionalnosti. Dodatno, ovaj pristup otežava održavanje i izmjenu programskih proizvoda bez utjecaja na korisnika.

Model računarstva zasnovanog na uslugama uvodi novi način razvoja i dostavljanja programskih funkcionalnosti koje se pruža i povezuje primjenom otvorenih Web tehnologija. U modelu računarstva zasnovanog na uslugama, usluga je nositelj programske funkcionalnosti. Pružatelji usluga iznajmljuju programske funkcionalnosti korisnicima te se korištenje programskih funkcionalnosti razdvaja od njihova posjedovanja (engl. *possession*) i vlasništva (engl. *ownership*). Programske funkcionalnosti se izgrađuju kao skup raspodijeljenih usluga koje je moguće povezati za vrijeme pružanja usluge. Time usluge u odnosu na programske proizvode skraćuju postupak koji prethodi korištenju usluga te olakšavaju uporabu, postavljanje i izmjenu programskih funkcionalnosti.

Primjena otvorenih Web tehnologija za pružanje usluga omogućuje povezivanje i suradnju usluga u poslovnim procesima na Internetu. Kritični zahtjev povezivanja i suradnje poslovnih organizacija jest sigurnost. Poslovne organizacije nisu spremne povezivati se i surađivati na tržištu usluga, ako je time ugrožena sigurnost njihova poslovanja. Stoga poslovne organizacije štite svoje usluge sigurnosnim sustavom koji zadovoljava odredbe njihovih sigurnosnih zahtjeva. Odredbe sigurnosti poslovnih organizacija postavljaju najveće zahtjeve na autentikaciju, autorizaciju i nadzor pristupa korisnika. Nadzor pristupa uslugama posebno je značajan, jer neovlašteno korištenje usluga poslovnog sustava može prouzročiti štetu za poslovnu organizaciju u obliku novčanih gubitaka i otkrivanja povjerljjivih podataka.

Nadzor pristupa je mehanizam koji tumači korisnička prava pristupa, donosi odluke o dozvoli pristupa te sukladno odlukama omogućuje ovlašteni pristup sredstvima i sprječava neovlašteni pristup sredstvima u sustavu. U raspodijeljenim računalnim sustavima

functionalnosti nadzora pristupa uopćene su u okvir koji razlikuje zakonodavnu, upravnu i izvršnu funkcionalnost nadzora pristupa. Zakonodavna funkcionalnost definira prava pristupa korisnika u sustavu. Zadaća upravnih funkcionalnosti je tumačenje prava pristupa i donošenje odluka o dozvoli pristupa. Izvršne funkcionalnosti nadzora pristupa provode odluke o dozvoli pristupa koje donose upravne funkcionalnosti. Ostvarenjem navedenih funkcionalnosti nadzora pristupa, u računalnom sustavu oblikuje se zaštićena domena s nadziranim pristupom.

U magistarskom radu oblikovana je, ostvarena i opisana organizacija i arhitektura sustava *Nadzornik* za nadziranje pristupa uslugama raspodijeljenog sustava. Sustav *Nadzornik* razvijen je u suradnji tvrtke Ericsson Nikola Tesla d.d. i Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu. Istraživanje je potpomognuto nacionalnim tehnološkim projektom *CroGRID* pod pokroviteljstvom Ministarstva znanosti, obrazovanja i športa. *Nadzornik* stvara zaštićenu domenu u kojoj nadzire korištenje usluga. *Nadzornik* dodatno pruža sigurnosne usluge registracije i autentikacije korisnicima i uslugama te bilježi korištenje usluga. Primjenom *Nadzornika*, pružateljima usluga omogućuje se izlaganje funkcionalnosti usluge uz nadzor pristupa koji ostvaruje *Nadzornik*. Korisnici imaju mogućnost na siguran način koristiti ponuđene usluge iz *Nadzornikove* domene.

Magistarski rad je organiziran u osam poglavlja. U drugom poglavlju opisana je ideja računarstva zasnovanog na uslugama. Istaknuta su svojstva usluga i motivacija korištenja usluga. Prikazan je osnovni model korištenja, ponude i potražnje usluga te proširenja osnovnog modela. Opisana je najčešće primjenjivana i najprihvaćenija Web Services tehnologija za ostvarenje usluga. Treće poglavlje sadrži opis osnovnih pojmove o sigurnosti u raspodijeljenim računalnim sustavima. Opisani su postupci uspostave sigurnosti, a posebice se u opisu naglašava postupak nadzora pristupa. Četvrto poglavlje daje pregled XML razine sigurnosti Web usluga. Izneseni su osnovni problemi svojstveni XML razini sigurnosti te su opisani najvažniji standardi za uspostavu XML sigurnosti. Peto poglavlje opisuje pregled sigurnosnih sustava računalnih spletova.

U šestom poglavlju opisana je arhitektura i organizacija sustava *Nadzornik*. Predstavljeni su zahtjevi i svojstva sustava, opisani upravljački podaci i podjela funkcionalnosti te razrađeni komunikacijski protokoli u sustavu. U sedmom poglavlju detaljno je opisano programsko ostvarenje sustava *Nadzornik*. Opisana je programska okolina i upotrijebljeni alati za izgradnju pojedinih dijelova sustava. Prikazan je način rada i uporabe sustava *Nadzornik*. U osmom poglavlju iznesen je zaključak rada kao i smjernice budućega rada.

2. Računarstvo zasnovano na uslugama

Težnja globalizacije, jeftinoga povezivanja putem Interneta i oglašavanja informacija primjenom Web tehnologija, vodi prema stalnom porastu broja korisnika Web zajednice. Korisnici ostvaruju različite interese kroz aktivnosti koje im putem Interneta nude primjenski sustavi Web-a. Međutim, Web pruža ograničeni skup aktivnosti korisnicima. Većina aktivnosti svodi se na usluge: *pretraživanja* – Internet se koristi kao izvor informacija, *komunikaciju* – Internet se koristi kao komunikacijska infrastruktura za razmjenu elektroničke pošte, glasovnih ili drugih oblika poruka, *zabavu* – Internet se koristi kao sredstvo za zabavu, što uključuje mrežne igre, razmjenu audio i video sadržaja, i *nabavu* – Internet se koristi za kupovanje različite robe.

Sve nabrojene usluge jednostavnih funkcionalnosti pružaju korisnicima rezultate usluge u obliku koji zahtijeva vizualnu ili zvučnu obradu od strane čovjeka. Uvođenje otvorenih Internet standarda, koji predstavljaju podatke u računalu čitljivom obliku, daje početni poticaj razvoju naprednih usluga. Primjenom otvorenih Internet standarda napredne usluge pružaju složene funkcionalnosti koje se ostvaruju obradom rezultata postojećih raspodijeljenih usluga. Time se usluge povezuju i konačna funkcionalnost se pruža kroz niz međudjelovanja usluga. Usluge postaju osnovni gradivni elementi raspodijeljenih primjenskih sustava, što vodi do preokreta u načinu na koji se obavlja elektroničko poslovanje. Tipičan primjer povezivanja usluga pronalazi se u B2B računarstvu (engl. *business-to-business*), gdje jedna poslovna transakcija pokreće više drugih transakcija.

Napredne Web usluge zahtijevaju potporu autonomnosti usluga, slabo povezivanje (engl. *loose-coupling*) te jednostavno razvijanje usluga sastavljanjem (engl. *composition*) i ponovnim korištenjem usluga (engl. *reusability*). Stoga je razvijen novi model arhitekture za stvaranje, ponudu, potražnju, i pružanje usluga. Primjenski sustavi koji se zasnivaju na tom arhitekturnom modelu oblikuju posebnu granu računarstva – *računarstvo zasnovano na uslugama* (engl. *Service-Oriented Computing, SOC*) [1].

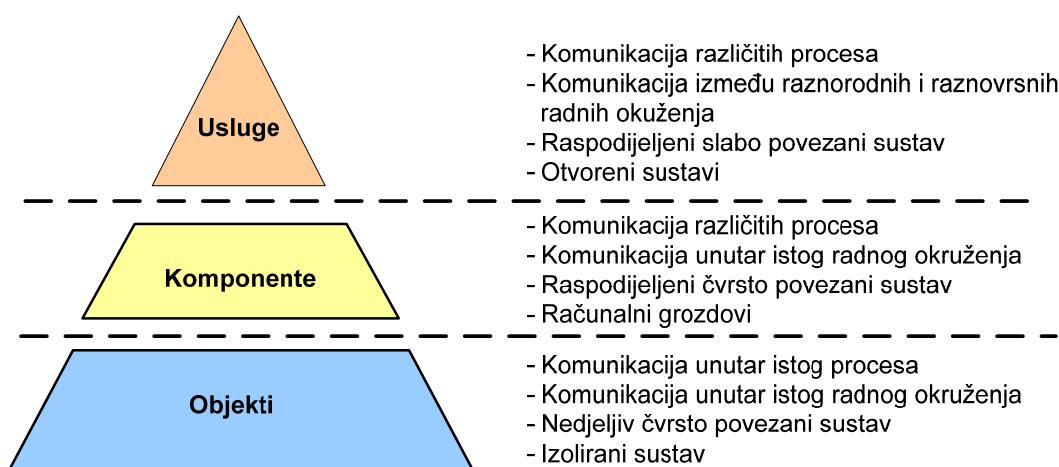
U ovom poglavlju, u odjelu 2.1 ističe se motivacija korištenja usluga u izgradnji otvorenih, raspodijeljenih računalnih sustava. Osim motivacije, navode se glavna značajna svojstva usluga. U odjelu 2.2 prikazan je model računalne arhitekture zasnovane na uslugama (engl. *Service-Oriented Architecture*) u kojem se ističe osnovni model pružanja i korištenja usluga te proširenja funkcionalnosti osnovnog modela. Tehnologije Web usluga opisane su u odjelu 2.3.

2.1. Motivacija korištenja usluga

Informacijski sustavi sve se više grade s težnjom povezivanja i suradnje s drugim programskim sustavima. Radi otvaranja za suradnju s drugim sustavima, takvi informacijski sustavi nazivaju se *otvoreni sustavi* (engl. *open-ended systems*). Razvoj računarstva zasnovanog na uslugama uvelike je motiviran zahtjevima izgradnje otvorenih sustava.

Izgradnja otvorenih sustava zahtjeva nadvladavanje problema povezivanja i suradnje raznorodnih programskih sustava. Problemi povezivanja raznorodnih programskih sustava javljaju se zbog raznorodnih tehnologija korištenih za izgradnju sustava, raznorodnih računalnih okolina i različitih operacijskih sustava na kojima se ti sustavi izvode, različitih formata za predstavljanje podataka te različitih komunikacijskih protokola. Zahtjevana obilježja otvorenih sustava su jednostavna proširivost sustava, prijenos podataka standardiziranim protokolima, predstavljanje podataka standardiziranim formatima zapisa, povezivanje primjenom standardnih Interneta protokola i slično. Stoga programski elementi za izgradnju otvorenih sustava moraju ostvariti sučelje neovisno o posebnostima okoline u kojoj su izgrađeni, omogućiti upravljivost i prilagodljivost prema dinamičkim promjenama sustava te slabu povezanost sustava.

Na slici 2-1 prikazane su tri različite vrste programske elemenata i načina oblikovanja koji se primjenjuju u izgradnji informacijskih sustava [2]. *Objekti* su programski elementi koji komuniciraju u istoj radnoj okolini, unutar istoga procesa. Sustavi izgrađeni od objekata su nedjeljivi i čvrsto povezani unaprijed zadanim međuzavisnostima objekata. Takvi informacijski sustavi su izolirani na jednom računalu, na kojem korisnik ostvaruje međudjelovanje sa sustavom te nisu prikladni za postizanje otvorenosti sustava.



Slika 2-1: Hijerarhija i usporedba programskih elemenata

Komponente su programski elementi koji komuniciraju između različitih procesa unutar kojih se izvode, ali ne postižu neovisnost od radne okoline u kojoj su ostvareni. Komponente svoju funkcionalnost grade od objekata te ju izlažu primjenom vlasničkih (engl. *proprietary*), najčešće nestandardiziranih protokola. Sustavi izgrađeni od komponenti raspodijeljeni su na logičke cjeline koje su čvrsto povezane vlastitim komunikacijskim protokolima. Takvi informacijski sustavi podržavaju izvođenje na više istovrsnih računala koja se nazivaju grozdovi (engl. *cluster*) te ne ispunjavaju u potpunosti zahtjeve povezivanja i suradnje otvorenih sustava.

Usluge su samoopisujući (engl. *self-describing*) programski elementi s dobro definiranim sučeljima, koji svoju funkcionalnost izlažu primjenom prihvaćenih, otvorenih Internet standarda. Funkcionalnosti usluge izgrađuju se od komponenti i objekata te slabim povezivanjem s drugim uslugama koje nude gotove funkcionalnosti. Usluge prikrivaju tehnološka svojstva radne okoline u kojoj se izvode. Usluge se izvode kao autonomne funkcionalne jedinice koje omogućuju povezanost procesa iz raznovrsnih okolina i izgradnju otvorenih sustava.

Vizija suvremenog programskog inženjerstva zasniva se na uslugama [3, 4]. Koncept usluga usmjerava programsко inženjerstvo prema dinamičkom dostavljanju funkcionalnosti (engl. *software delivery*) na tržište programske potpore vođeno zahtjevima poslovanja i korisnika usluga. Usluge unaprjeđuju dostavljanje i razvoj (engl. *software development*) programskih funkcionalnosti razdvajanjem posjedovanja programskog vlasništva od uporabe programskih funkcionalnosti. Programske funkcionalnosti se poslužuju kao skup raspodijeljenih usluga koje je moguće povezati za vrijeme pružanja usluge. Na taj način pomiču se granice ponovne uporabe (engl. *reuse*), postavljanja (engl. *deployment*) i evolucije (engl. *evolution*) programskih funkcionalnosti.

2.2. Arhitektura zasnovana na uslugama

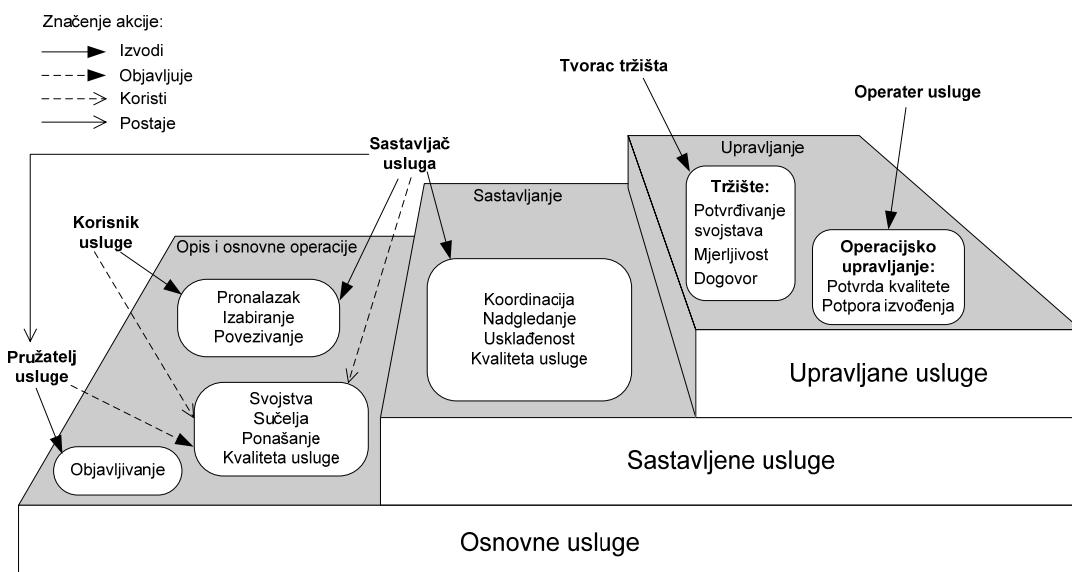
Arhitektura zasnovana na uslugama definira model pružanja i korištenja programskih funkcionalnosti u obliku usluga. Cilj arhitekture zasnovane na uslugama je uspostavljanje jednostavnog modela za stvaranje inovativnih poslovnih procesa, lanaca dobiti funkcionalnosti (engl. *value chain*) te upravljanja tržištem usluga.

Na slici 2-2 prikazana je slojevitost arhitekture računarstva zasnovanog na uslugama na način kako je predstavljena u radu [5]. Autor u navedenom radu opisuje viziju funkcionalnosti potrebnih u arhitekturi računarstva zasnovanog na uslugama. Stoga je

arhitektura, opisana u nastavku, skup načela koja obuhvaćaju postojeće elemente arhitekture zasnovane na uslugama i opisuju viziju proširenja arhitekture zasnovane na uslugama.

Razlikuju se tri sloja arhitekture: osnovni sloj, sloj sastavljanja i upravljački sloj. Osnovni sloj pruža potporu za izgradnju jednostavnih primjenskih sustava zasnovanih na uslugama. Funkcionalnosti potrebne za sastavljanje nove usluge od postojećih usluga (engl. *composition*) pripadaju sloju sastavljanja. Na najvišoj razini apstraktno su definirani načini upravljanja uslugama na tržištu usluga, koji su smješteni u sloj upravljanja. Svaki sloj ostvaruje vlastite funkcionalnosti i pruža potporu koju koristi sljedeći sloj.

U iduća dva odjeljka opisuju se slojevi arhitekture zasnovani na uslugama, što je prikazano na slici 2-2. U odjeljku 2.2.1 opisuje se osnovni sloj arhitekture zasnovane na uslugama, a u odjeljku 2.2.2 slojevi sastavljanja i upravljanja kojima se predlažu koncepti proširivanja funkcionalnosti osnovnog sloja.



Slika 2-2: Slojevita arhitektura zasnovana na uslugama

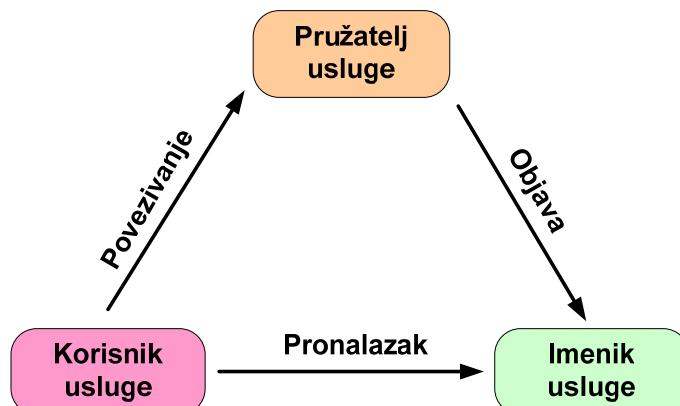
2.2.1. Osnovna arhitektura zasnovana na uslugama

Jednostavna izgradnja primjenskih sustava zasnovanih na uslugama ostvaruje se na osnovnom sloju arhitekture zasnovane na uslugama (slika 2-2). Glavne funkcionalnosti i zadaci tog sloja su opis usluga i potpora osnovnih operacija korištenja, ponude i potražnje usluga. Sudionici koji izvode operacije i ostvaruju funkcionalnosti *osnovnog sloja* jesu korisnik i pružatelj usluga.

Opis usluge sadrži opise svojstava (engl. *service capabilities*), sučelja, ponašanja i kvalitete usluge. Opis svojstva usluge izražava namjenu i rezultate dobivene uslugom. Opis

sučelja usluge izražava operacije usluge, ulazne i izlazne parametre operacija, strukture poruka te poruke za izražavanje pogreške. U opisu ponašanja usluge zadaje se očekivano ponašanje usluge tijekom izvođenja, na primjer radni tijek procesa (engl. *process workflow*). Opis kvalitete usluge izražava važna funkcionalna i nefunkcionalna svojstva kvalitete usluge poput cijene, metrike izvođenja (na primjer, vrijeme odziva), svojstva sigurnosti, nepovredivosti transakcije, pouzdanosti, raspoloživosti i slično. Standardiziran je način opisivanja sučelja usluge, dok su opisi ostalih svojstava usluge predmet daljnjih istraživanja.

Osnovne operacije ponude, potražnje i korištenja usluga obuhvaćaju objavu (engl. *publish*) opisa usluge, pronalazak (engl. *discovery*) usluge, izabiranje usluge (engl. *selection*) i povezivanje s uslugom (engl. *binding*). Objavom opisa usluge javno se oglašava ponuda usluge. Od javno ponuđenih usluga operacijom pronalaska izabire se skup usluga s traženim svojstvima. Na osnovi opisa kvalitete usluge, operacijom izbora se iz skupa pronađenih usluga s traženim svojstvima izabire najprikladnija usluga. Izabrana usluga koristi se operacijom povezivanja s uslugom. Postoje razvijeni standardi za objavu usluga, otkrivanje usluga i povezivanje s uslugama, dok su standardi za izbor usluga predmet daljnjih istraživanja.



Slika 2-3: Komunikacijski model arhitekture zasnovane na uslugama

Tipičan scenarij ponude, potražnje i korištenja usluge ostvaruje se komunikacijskim modelom koji je proširenje modela korisnik-poslužitelj (engl. *client-server*). U proširenom komunikacijskom modelu prikazanom na slici 2-3 sudjeluju pružatelj, imenik i korisnik usluge. Pruzatelj usluge ostvaruje i održava uslugu dostupnu putem mreže, nudi uslugu te nadzire pristup usluzi. Imenik usluge održava popis javno ponuđenih usluga koje nude poslužitelji. Korisnik usluge poziva usluge i koristi ponuđene funkcionalnosti usluga od pružatelja usluge.

Međusobna povezanost triju sudionika ostvaruje se izvođenjem triju osnovnih operacija: objave, pronalaska i povezivanja. Imenik usluga posreduje u *povezivanju*

korisnika usluge s poslužiteljem usluge i uloga mu je omogućiti pružateljima *objavljanje* usluga te korisnicima olakšati *pronalaženje* usluga. Pružatelj usluge objavljuje u imeniku opis usluge koju nudi. Na osnovi objavljenih opisa u imeniku, korisnik pretražuje, pronalazi i izabire uslugu koju želi koristiti. Na osnovi poznatog opisa izabrane usluge, korisnik izvodi povezivanje s pružateljem usluge.

2.2.2. Proširenje arhitekture zasnovane na uslugama

Osnovna arhitektura zasnovana na uslugama ne obuhvaća probleme sastavljanja usluga, upravljanja transakcijama usluga, sigurnost pružanja usluga, upravljanja uslugama na tržištu i slično. Navedeni problemi su predmeti istraživanja arhitekture zasnovane na uslugama. Svrstani su u sloj sastavljanja usluga i upravljački sloj arhitekture zasnovane na uslugama. Ta dva sloja proširuju koncepte osnovnog sloja arhitekture zasnovane na uslugama. Sloj sastavljanja usluga i upravljački sloj prikazani su slikom 2-2.

Sloj sastavljanja usluga (engl. *composition layer*) obuhvaća uloge i funkcionalnosti nužne za izgradnju nove usluge od više postojećih usluga. Postojeće usluge od kojih se grade nove usluge nazivaju se *osnovne usluge* (engl. *component service*), dok se dobivene usluge nazivaju *sastavljene usluge* (engl. *composite service*). Sastavljene usluge služe sastavljaču usluga (engl. *service aggregator*) kao osnovne usluge za daljnje sastavljanje novih usluga ili ih korisnici izravno koriste. Sastavljači usluga postaju pružatelji usluga objavljanjem opisa sastavljenih usluga. Funkcionalnosti koje izvode sastavljači usluga obuhvaćaju koordinaciju usluga, nadgledanje usluga, usklađivanje usluga, i kvalitetu sastavljanja usluga.

Koordinacija usluga [6] (engl. *coordination*) podrazumijeva kontrolu izvođenja osnovnih usluga i upravljanje tijekom podataka između osnovnih usluga. Taj postupak se ostvaruje definiranjem tijeka izvođenja procesa (engl. *workflow definition*) i korištenjem izvoditelja radnog tijeka (engl. *workflow engine*). Izvoditelj radnog tijeka upravlja izvođenjem usluga na osnovi definiranog procesa tijeka izvođenja.

Nadgledanje usluga (engl. *monitoring*) podrazumijeva pretplatu na obavijest o informacijama koje stvaraju osnovne usluge te objavljanje obavijesti višim slojevima sastavljenih usluga. Funkcionalnosti nadgledanja se ostvaruju filtriranjem (engl. *filtering*), sakupljanjem (engl. *summarizing*) i povezivanjem (engl. *correlating*) prikupljenih obavijesti.

Usklađivanje usluga (engl. *conformance*) provjerava podudarnost razmjene poruka sastavljenih usluga s osnovnim uslugama. Provjerava se da li su poruke sadrže podudarne vrste podataka i da li su iste strukture. Dodatno se od osnovnih usluga zahtijeva ispunjavanje

određenih pravila sastavljanja te se izvodi udruživanje podataka (engl. *data fusion activities*). Tipično se udruživanjem podataka na osnovi određenog skupa podataka, te njihovom obradom, dobiva jedinstvena željena informacija.

Kvaliteta sastavljanja usluga (engl. *quality of service composition*) podrazumijeva prikupljanje i primjenu podataka o kvaliteti osnovnih usluga s ciljem zaključivanja o kvaliteti sastavljenih usluga. Parametri kvalitete usluge su ukupna cijena sastavljene usluge, svojstva izvođenja, način provedbe autentikacije korisnika, privatnost, nepovredivost, pouzdanost i raspoloživost.

Upravljački sloj je najviši sloj arhitekture zasnovane na uslugama koji omogućuje izgradnju upravljanih usluga (engl. *managed service*). Upravljane usluge pružaju upravljivost značajnih, kritičnih usluga i tržišta usluga. Upravljivost značajnih, kritičnih usluga ostvaruje se operacijskim upravljanjem (engl. *operations management*). Upravljivost tržišta usluga postiže se uspostavom otvorenih tržišta usluga (engl. *service marketplaces*).

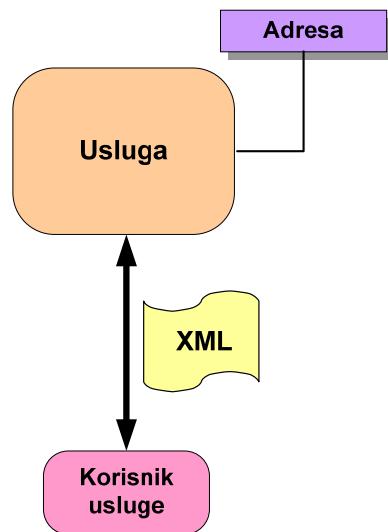
Operacijsko upravljanje podrazumijeva upravljanje okolinom za izvođenje usluga, nadgledanje rada sastavljenih usluga i ostvarivanje kritičnih, nefunkcionalnih radnih svojstava primjenskih sustava sastavljenih od usluga. Operacijsko upravljanje pruža potporu izvođenju kritičnih primjenskih sustava koji zahtijevaju upravljanje radnom okolinom, postavljanje (engl. *deployment*) usluga i primjenskih sustava u radnoj okolini i slično. Funkcionalnosti operacijskog upravljanja obuhvaćaju pružanje detaljne statistike o izvođenju primjenskih sustava, davanje procjene učinkovitosti primjenskog sustava, davanje uvida u cjelokupno poslovanje i dostavljanje dojave o izvršenju pojedinih aktivnosti ili postizanju određenog uvjeta. Za izvođenje takvih funkcionalnosti upravljanja odgovoran je *operator usluga* (engl. *service operator*). Operater usluga je korisnik ili sastavljač usluga.

Otvorena tržišta usluga pružaju funkcionalnosti opskrbnih lanaca (engl. *supply chain*) i industrijskog trgovanja (engl. *industry trade*). Opskrbni lanac uspostavlja jednoznačnosti proizvoda i usluga, standardne poslovne terminologije i opisa poslovnih procesa. Industrijsko trgovanje se odnosi na poslovno pregovaranje (engl. *negotiation*), potvrđivanje svojstava usluge (engl. *service certification*) i jamstvo kvalitete usluga (engl. *quality assurance*), vrednovanje usluga (engl. *rating services*), mjerljivost usluga (engl. *service metrics*) te provedba dogovora na razini usluga (engl. *service level agreement, SLA*). Otvorena tržišta stvaraju i održavaju tvorci tržišta (engl. *market-maker*). Tvorci tržišta su združene organizacije koje upoznavaju proizvođače i prodavače.

2.3. Tehnologije Web usluga

Web tehnologija je *de facto* standard pružanja usluga. Naziv Web usluga [5] pripisuje se svakoj usluzi koja sadrži dvije glavne značajke prikazane modelom na slici 2-4. Svaka Web usluga sadrži adresu koja jedinstveno određuje mjesto na Web-u gdje je moguće komunicirati s uslugom. Adresa usluge na Web-u predstavljena je URL adresom [9]. Svakoj Web usluzi svojstvena je komunikacija porukama zasnovana na otvorenom XML standardu. Podaci koji se razmjenjuju s uslugom predstavljeni su u XML obliku.

XML (*eXtensible Markup Language*) [10] je općeprihvaćeni standard za predstavljanje podataka neovisnih o računalnoj platformi. XML opisuje podatke otvorenim i dobro definiranim tekstualnim formatom te jednoznačnim zapisom koji podržava standardizirani način obrade podataka. Obrada podataka izvodi se na standardiziran način. Podaci se zapisuju u imenovane XML cjeline. Te su cjeline samostalni, samoopisujući blokovi podataka. Blok podataka XML cjeline može sadržavati nove XML cjeline, čime se podržava hijerarhijska organizacija podataka.



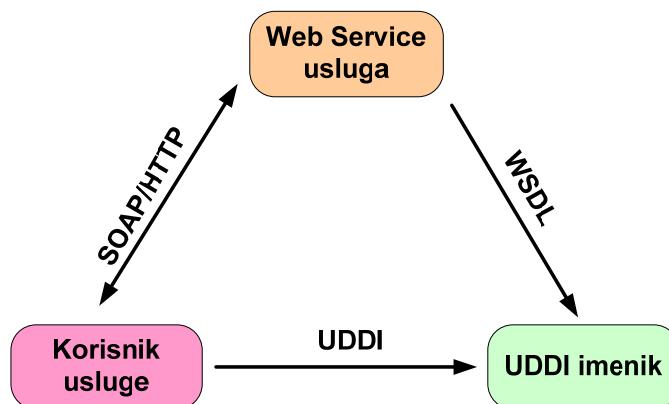
Slika 2-4: Minimalistički model Web usluge

Pod utjecajima različitih smjerova iz područja računarstva razvijene su tri glavne skupine tehnologija i protokola računarstva zasnovanog na Web uslugama [3]. Jedan dio razvijenih tehnologija usmjeren je prema semantici usluga i dolazi od inicijative zajednice semantičkog Web-a (*Semantic Web*) i organizacije W3C. U drugu skupina protokola, koju razvija poslovna zajednica uz potporu organizacije OASIS i United Nations CEFAC, ubrajaju se tehnologije zasnovane na standardu ebXML (*Electronic Business XML*). Međutim, najšire je prihvaćena i *de facto* standardizirana Web Services tehnologija pružanja usluga, koju podupiru najveći glavni proizvođači programske potpore kao što su Microsoft,

IBM, Sun i drugi. Sva razmatranja u nastavku ovog rada odnose se na Web Services tehnologiju pružanja usluga, koju se detaljnije opisuje u sljedećem odjeljku.

2.3.1. Tehnologije Web Services usluga

Web Services usluge [7] definiraju standardni skup tehnologija za ostvarenje arhitekture zasnovane na uslugama (slika 2-5). *WSDL* je jezik za opisivanje osnovnih svojstava Web Services usluge. U *UDDI imeniku* objavljuje se WSDL opis Web Services usluge. Korisnik usluge primjenom UDDI standardnog protokola pretražuje WSDL opise u UDDI imeniku i izabire Web Services uslugu s odgovarajućim svojstvima. Povezivanje i komunikacija između korisnika i izabrane Web Services usluge ostvaruje se razmjenom SOAP poruka najčešće putem HTTP protokola.



Slika 2-5: Tehnologije Web Services usluga

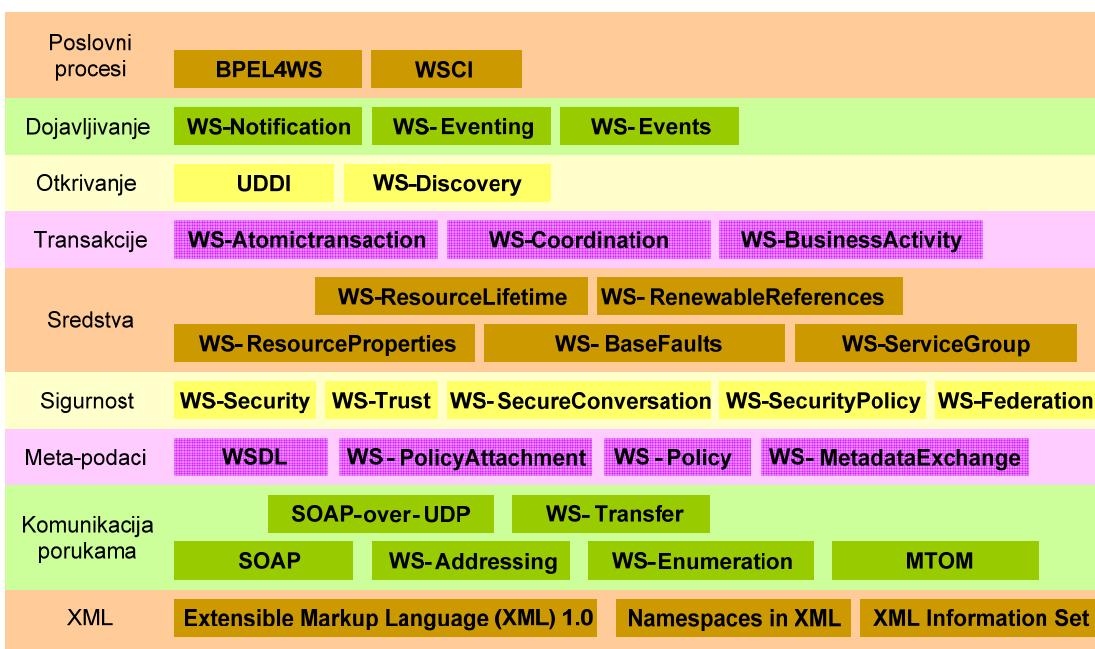
WSDL (Web Services Description Language) [8] je jezik za opis svojstava sučelja Web Services usluga zasnovan na XML jeziku. WSDL opis sučelja definira ponuđene operacije koje usluga izvodi, strukturu poruka zahtjeva i odgovora pojedinih operacija usluge te ulazne i izlazne parametre. Dodatno, WSDL opis opisuje pristupnu točku usluge i prijenosni komunikacijski protokol za razmjenu poruka s uslugom. WSDL jezik ne obuhvaća opis svojstava sigurnosti usluge, semantiku usluge te svojstva ostvarenja i izvođenja usluge.

UDDI [11] (engl. *Universal Description Discovery and Integration*) definira protokole usluge hijerarhijske baze podataka koja sadrži WSDL opise Web Services usluga. UDDI protokol standardizira pretraživanje i objavljivanje podataka o uslugama primjenom načela bijelih, žutih i zelenih stranica telefonskog imenika. *Bijele stranice* (engl. *white pages*) sadrže podatke o pružateljima usluge. *Žute stranice* (engl. *yellow pages*) sadrže podatke o kategoriji usluga. *Zelene stranice* (engl. *green pages*) sadrže tehničke podatke o uslugama.

SOAP [12] je jednostavan protokol zasnovan na XML jeziku namijenjen razmjeni poruka u raspodijeljenoj raznorodnoj okolini. Za prijenos SOAP poruka primjenjuju se različiti prijenosni protokoli. Najčešće korišteni prijenosni protokoli su HTTP (*Hyper Text Transfer Protocol*) [13], FTP (*File Transfer Protocol*) [14], SMTP (*Simple Mail Transfer Protocol*) [15]. SOAP poruke najčešće služe za poziv udaljenih procedura (engl. *remote procedure call*). Budući je pritom uobičajena komunikacija oblika zahtjev-odgovor, te je HTTP komunikacijski protokol zahtjev-odgovor komunikacijskog obrasca, HTTP je pogodan za prijenos SOAP poruka. Pritom su poruke SOAP zahtjeva ugniježđene u HTTP zahtjeve, a poruke SOAP odgovora u HTTP odgovore.

2.3.2. Proširenja Web Services tehnologije

Osim osnovnih tehnologija koje koriste Web Services usluge, razvijen je niz tehnologija koje nadopunjuju osnovne Web Services tehnologije. Najviše proširenja dolazi iz zajednice poslovnog računarstva i iz znanstvene zajednice računarstva spletova (engl. *grid computing*). Područje razvoja Web Services tehnologija vrlo je aktivno, čemu u prilog govori stog Web Services protokola prikazan slikom 2-6. Na lijevom kraju slike prikazana su područja problematike prema kojima su protokoli razvrstani.



Slika 2-6: Stog Web Services protokola

XML predstavlja najniži sloj protokola koji obuhvaća osnovni XML standard i njegova proširenja. Iznad njega je sloj komunikacije porukama u kojem se kao osnovni standard nalazi SOAP i WS-Addressing te standardi za povezivanje SOAP poruka s

prijenosnim protokolima. Sloj meta-podataka sadrži WSDL standard za opis osnovnih svojstava usluge i dodatke kojima se ostvaruje prošireni opis usluga. Sloj sigurnosti obuhvaća standarde za sigurnost Web Services usluga, čiju osnovicu čini standard WS-Security.

U sloju sredstava nabrojeno je nekoliko specifikacija WSRF (*Web Services Resources Framework*) radnog okvira. WSRF radni okvir [16] primjer je važnog proširenja Web Services usluga, nastao zajedničkim naporom dviju istraživačkih zajednica. Znanstvena zajednica je započela s razvojem Grid usluga [17, 18] koje su pružale očuvanje svojstvenih vrijednosti i stanja nakon izvođenja Web Services usluga. Zatim se koncept Grid usluga uskladio s postojećim skupom WS standarda i formirao se skup WSRF specifikacija. WSRF skup specifikacija definira standardizirana sučelja i format poruka za pristup vrijednostima svojstava usluge, te za upravljanje životnim ciklusom usluge, što je osnova za ostvarivanje Web usluga s čuvanjem stanja (engl. *stateful services*).

Transakcijski sloj definira protokole potrebne za primjenu Web Services usluga u poslovnim okruženjima. Sloj otkrivanja predstavljen je UDDI protokolom i služi za oglašavanje i pronalaženje opisa usluga. Sloj dojavljivanja obuhvaća standarde koji podržavaju dojavu događaja i automatizaciju pretplate i objave događaja, od kojih je glavni standard WS-Notification [19]. Na vrhu stoga su protokoli za organiziranje poslovnih procesa gdje je najznačajniji protokol BPEL4WS (*Business Process Execution Language for Web Services*) [20].

3. Sigurnost u raspodijeljenim računalnim sustavima

Računalna sigurnost je grana računarstva koja se razvija od početka računarskih znanosti. Razvijeni su mnogi načini sigurnosne zaštite na poljima kriptografije, uspostavljeni su mnogi protokoli sigurnosti za očuvanje tajnosti, vjerodostojnosti, mehanizmi uspostave sigurne komunikacije, modeli nadzora pristupa i slično. Međutim, unatoč razvijenim postupcima uspostave sigurnosti, sigurnost velikog broja računalnih sustava je nezadovoljavajuća. U većini sustava odlučan i sposoban napadač može provaliti i otuđiti ili oštetići podatke.

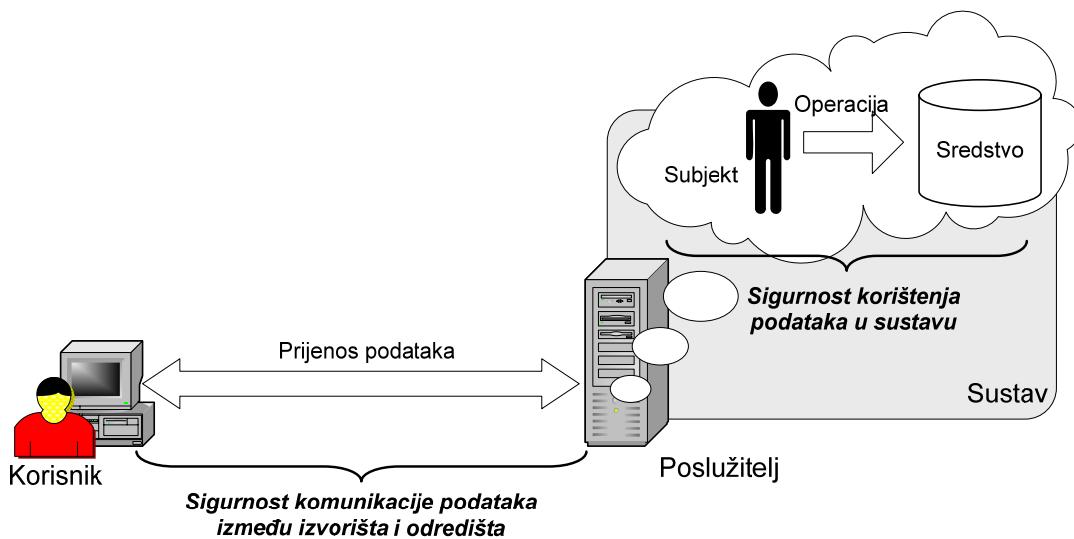
Internet je globalnim povezivanjem računala dodatno otežao problem računalne sigurnosti. Prije povezivanja lokalnih mreža u Internet, računalni sustavi su imali mali broj korisnika, od kojih su svi bili pripadnici iste organizacije. Danas se milijuni korisnika širom svijeta spajaju na Internet. Svaki korisnik je potencijalni napadač. Ako se *provali* u sustav (engl *compromise*), sustav automatski može poslužiti napadačima kao sredstvo za provaljivanje drugih sustava na Internetu i tako širiti "*zarazu*". Sustavi i korisnici često se suočavaju sa zlonamjernom programskom potporom. Cilj sigurnosnih sustava je onemogućiti napade napadača, na vrijeme ih prepoznati i suzbiti.

U ovom poglavlju opisani su načini na koje napadači mogu narušiti sigurnost računalne komunikacije i sigurnost podataka spremljenih u računalnom sustavu. Razmatraju se svojstva sigurnosti podataka koje se želi zaštititi i očuvati primjenom sigurnosnih mehanizama. Daje se pregled postupaka pomoću kojih se uspostavlja sigurnost u komunikacijskom kanalu i sigurnost pristupa podacima sustava. Posebna pažnja posvećena je nadzoru pristupa i autorizaciji, te su prikazani različiti modeli nadzora pristupa i načela oblikovanja nadzora pristupa u raspodijeljenim sustavima.

3.1. Načini ugrožavanja sigurnosti

S obzirom na mjesto ugrožavanja sigurnosti, računalna sigurnost raspodijeljenih sustava dijeli se na rješavanje dviju vrsta problema (slika 3-1). Jedna vrsta problema je održavanje sigurnosti podataka u komunikaciji dviju strana. Na primjer, uljezu se ne smije dopustiti mijenjanje ili čitanje podataka koji se razmjenjuju između dviju strana koje komuniciraju. Drugi problem je održavanje sigurnosti uporabe podataka i sredstava u sustavu. Na primjer, ne smije se dopustiti neovlašteno mijenjanje i korištenje tudihih podataka. Ako se u raspodijeljenom računalnom sustavu želi korisniku jamčiti sigurnost obavljanja

operacije nad podatkom u sustavu, onda se mora jamčiti sigurnost podataka u komunikaciji i sigurnost podataka u sustavu.



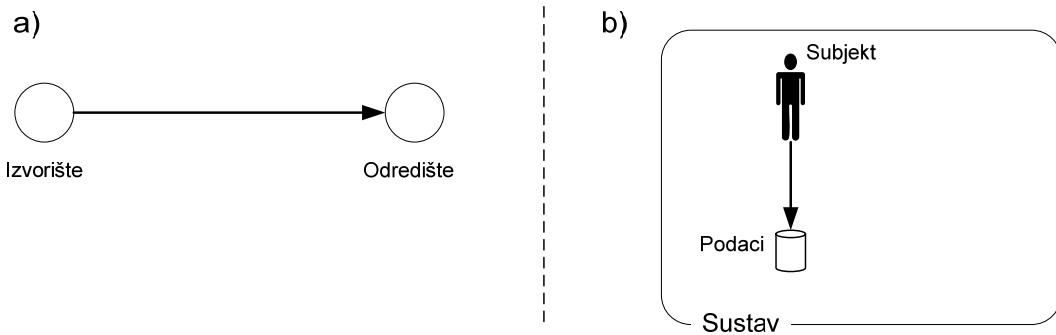
Slika 3-1: Dva problema zaštite sigurnosti

Neovlaštene korisnike obično se naziva napadačima ili uljezima. Uljezi često pokušavaju narušiti sigurnost podataka koji se razmjenjuju komunikacijom ili se nalaze spremljeni u računalnom sustavu. Interesi napadača najčešće su usmjereni na ugrožavanje sigurnosti vojnih informacija, bankovnih informacija, državnih i bolničkih informacija. Sigurnost podataka uključuje tri osnovna svojstva sigurnosti: tajnost, nepovredivost i raspoloživost podataka [21, 22]. Zadaća sigurnosnih postupaka je zaštитiti navedena svojstva sigurnosti podataka.

Tajnost ili *povjerljivost* (engl. *secrecy*, *confidentiality*) je svojstvo sigurnosti koje zahtijeva zaštitu od neovlaštenog čitanja informacija razmijenjenih komunikacijom, odnosno zaštitu čitanja informacija koje se nalaze spremljene u sustavu. Svojstvo *nepovredivosti* ili *vjerodostojnosti* podataka (engl. *integrity*) zahtijeva zaštitu od neovlaštene izmjene podataka i unošenja neispravnih podataka u komunikaciju ili među podatke spremljene u sustavu. *Raspoloživost* podataka (engl. *availability*) odnosi se na osiguravanje dostupnosti podataka koji se razmjenjuju komunikacijom te podataka koji su spremljeni u sustavu.

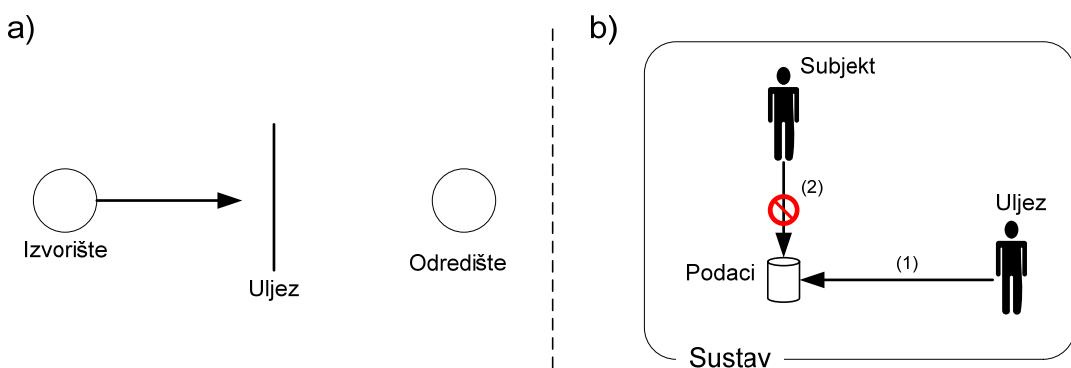
Navedenim svojstvima sigurnosti dodaju se još zahtjevi autentičnosti i pribilježenosti [24]. *Autentičnost* (engl. *authenticity*) zahtijeva utvrđivanje i provjeru autentičnosti identiteta sudionika s kojim se komunicira kao i subjekta koji u sustavu izvodi određene operacije. *Pribilježenost* (engl. *accountability*) odnosi se na praćenje i zapisivanje svih operacija ili akcija koje korisnik izvodi u sustavu.

Na slici 3-2 prikazan je slučaj u kojem su osigurani svi elementi sigurnost podataka u komunikaciji i u sustavu. Slika 3-2a prikazuje komunikaciju dvaju sudionika, gdje se razmjena informacija opisuje protokom informacija od izvorišta do odredišta. Na slici 3-2b prikazani su podaci spremljeni u sustavu i subjekt koji je ovlašten čitati i mijenjati podatke prema svojim potrebama.



Slika 3-2: Slučaj u kojem nije narušena sigurnost podataka

Napadači mogu na više načina ugroziti jedno ili više svojstava sigurnosti podataka. Stoga razlikujemo različite vrste napada: *prekid*, *prisluškivanje*, *izmjena* i *izmišljanje*. Svaki od napada analizira se usporedno u kontekstu sigurnosti podataka koji se razmjenjuju komunikacijom i u kontekstu korištenja podataka koji se nalaze spremljeni u sustavu.

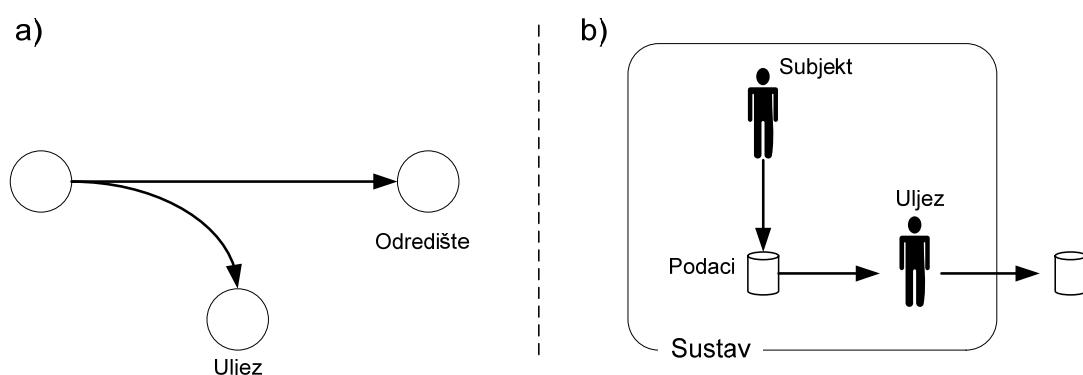


Slika 3-3: Ugrožavanje sigurnosti napadom prekida

Na slici 3-3 prikazano je narušavanje sigurnosti napadom prekida. Na slici 3-3a prikazan je slučaj u kojem uljez sprječava normalni protok informacija od izvorišta do odredišta. Uljez zaustavlja protok informacija, te odredište ne prima informacije koje mu se šalju iz izvorišta. Na slici 3-3b prikazan je slučaj u kojem podatak subjekta postaje nedostupan djelovanjem uljeza. Nedostupnost podatka može biti posljedica brisanja ili neispravnosti podatka. Slično, podatak može postati nedostupan ako poslužitelj podataka

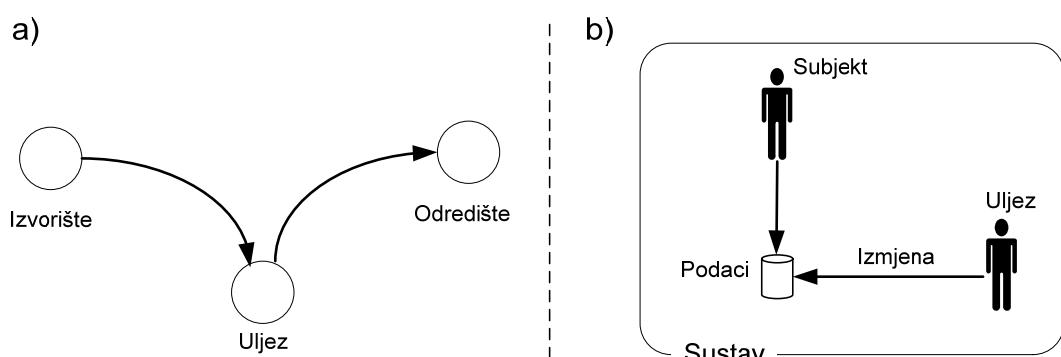
zbog velikog opterećenja koje stvaraju zahtjevi uljeza ne može poslužiti podatke korisnicima. Napadom prekida ugrožava se raspoloživost informacija.

Na slici 3-4 prikazano je narušavanje sigurnosti prisluškivanjem. Na slici 3-4a informacije koje se šalju s izvořišta dolaze do odredišta, međutim uljez također prima informaciju poslanu odredištu. Vrsta prisluškivanja je i neovlašteno preslikavanje podataka prikazano na slici 3-4b. Prisluškivanjem je ugrožena tajnost informacija, jer uljez znajući povjerljive informacije ima mogućnost iskoristiti ih na štetu strane koja se koristi informacijama.



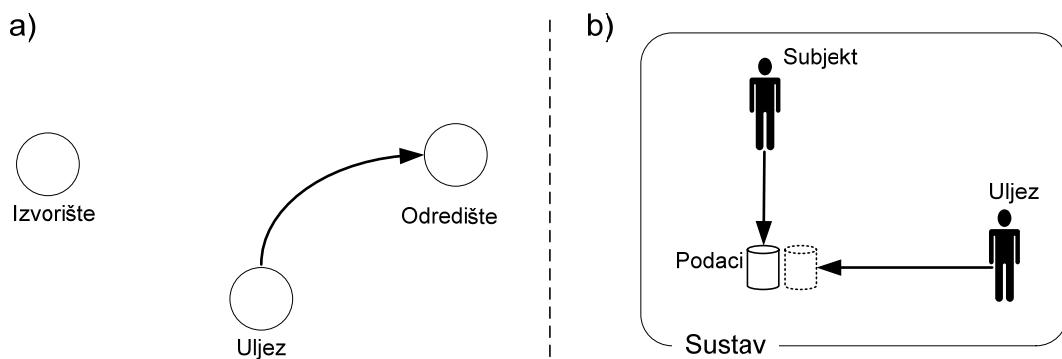
Slika 3-4: Ugrožavanje sigurnosti prisluškivanjem

Na slici 3-5 prikazano je narušavanje sigurnosti izmjenom podataka. Na slici 3-5a protok informacija narušen je izmjenom protoka. Uljez presreće informacije poslane s izvořišta, mijenja ih i šalje odredištu, te odredište prima izmijenjene informacije. Na slici 3-5b uljez mijenja postojeći podatak u sustavu, npr. zapis u bazi podataka ili program, i time omogućuje svoje daljnje djelovanje. Na primjer, izmjenom programa uljez podesi program da tajno zapisuje korisničke radnje. Napad izmijene podataka ugrožava nepovredivost i autentičnost podataka.



Slika 3-5: Ugrožavanje sigurnosti izmjenom podataka

Na slici 3-6 prikazano je narušavanje sigurnosti izmišljanjem informacija. Na slici 3-6a prikazan je slučaj u kojem uljez izmišlja protok informacija. Uljez šalje podatke i predstavlja se odredištu u ulozi izvorišta. U izmišljanje se ubraja slučaj prikazan na slici 3-6b, gdje subjekt dodaje podatke koji inače u sustavu ne postoje. Tako primjerice uljez dodaje zapis o dospijeću novaca na svoj bankovni račun u bazu podataka ili dodaje zapis o pravima pristupa na tuđi račun. Prilikom izmišljanja informacija uljez ugrožava autentičnost i nepovredivost informacija.



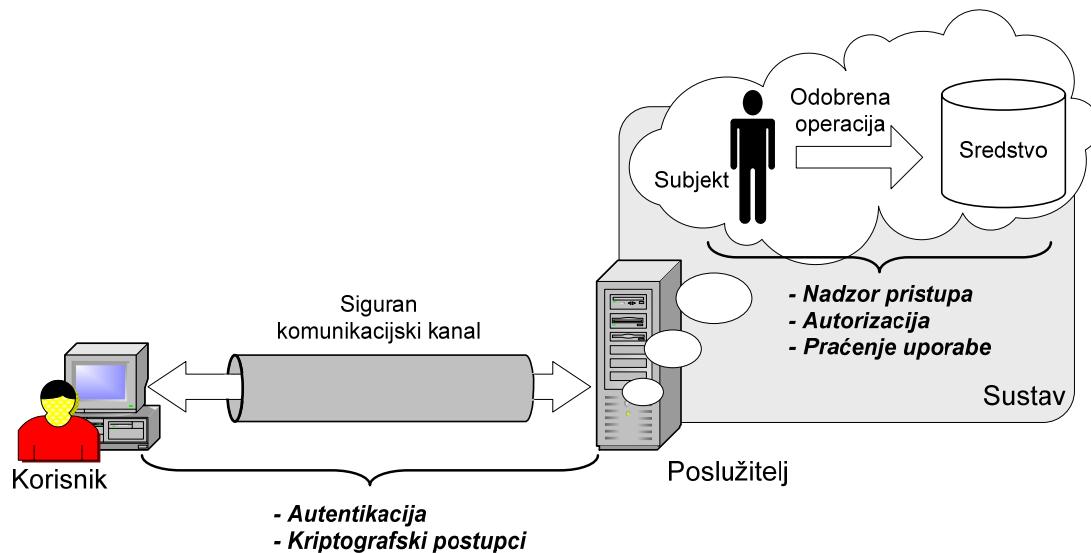
Slika 3-6: Ugrožavanje sigurnosti izmišljanjem podataka

3.2. Postupci za uspostavu sigurnosti

Sigurnosne odredbe (engl. *security policy*) definiraju ciljeve koji su značajni za uspostavu sigurnosti u računalnim sustavima. Ciljevi definirani sigurnosnim odredbama posebni su u svakom sustavu i ovise o okolini u kojoj je sustav postavljen, o ulozi i funkciji koju sustav ostvaruje i slično. Na primjer, u sustavima koji su postavljeni na izolirana računala ne treba primjenjivati postupke za sigurnost međuračunalne komunikacije. Međutim, u sustavima koji su raspodijeljeni i povezani putem javno dostupne mreže treba pažljivo osmislitи sustav sigurnosti i njime obuhvatiti zaštitu računalne komunikacije.

Postupci za uspostavu sigurnosti obuhvaćaju kriptografske postupke, postupke autentikacije, autorizacije, nadzor pristupa i praćenje uporabe. Navedeni postupci služe kao osnova za izgradnju sustava sigurnosti. Uloga sustava sigurnosti je sprječavanje djelovanja uljeza. Na slici 3-7 prikazana su dva glavna problema uspostave sigurnosti u modelu korisnik-poslužitelj. Prvi problem je uspostava sigurne komunikacije između korisnika i poslužitelja. Sigurna komunikacija ostvaruje se uspostavom *sigurnog kanala* između sudionika. Za uspostavu sigurnog kanala sudionici koji komuniciraju moraju se predstaviti i uvjeriti u identitet svoga sugovornika. Uspostava sigurnog kanala postiže se postupkom obostrane autentikacije sudionika. Nepovredivosti i tajnosti podataka osigurava se

kriptografskim postupcima zaštite. Budući da je autentikacijski postupak usko povezan s kriptografskim postupcima, u nastavku se najprije objašnjavaju kriptografski postupci a zatim autentikacijski postupak.



Slika 3-7: Rješavanje dvaju problema uspostave sigurnosti

Drugi problem javlja se nakon uspostave sigurnog kanala i vezan je uz nadzor operacija koje korisnik izvodi na poslužitelju. Pod tom prepostavkom, korisnikovi zahtjevi stižu sigurnim kanalom do poslužitelja. Međutim, poslužitelj ne poslužuje sve primljene zahtjeve. Poslužitelj odlučuje koji će primljeni korisnikov zahtjev poslužiti a koji odbaciti. Ako bi poslužitelj posluživao sve primljene zahtjeve, zahtjevi jednog korisnika mogli bi ugroziti sigurnost podataka drugoga korisnika. Stoga poslužitelj vodi računa o *ovlastima korisnika* koje korisnički zahtjevi ne smiju prekoračiti. Neovlaštene radnje moraju se suzbiti, a poslužiti se mora sve ispravne zahtjeve. U slučaju propusta u kojem je narušena sigurnost, analizom propusta mора se omogućiti oporavak i spriječiti ponavljanje istog ili sličnog propusta. Postupcima nadzora pristupa, autorizacije i praćenja uporabe ispunjavaju se navedeni zahtjevi i uspostavlja sigurnost posluživanja sredstava u sustavu. Postupak nadzora pristupa u svrhu uspostave sigurnosti u sustavu razmjenjuje upravljačke informacije s druga dva navedena postupka, te dodatno s postupkom autentikacije.

3.2.1. Kriptografski postupci

Znanost koja se bavi razvojem i primjenom postupaka kriptiranja i dekriptiranja zove se kriptografija [25, 26, 33]. Pojam kriptografija je preuzet iz grčkog jezika (grč. *kryptos* = tajno, *graphein* = pisati), a ima značenje "tajnog pisanja". Kriptografija se prije

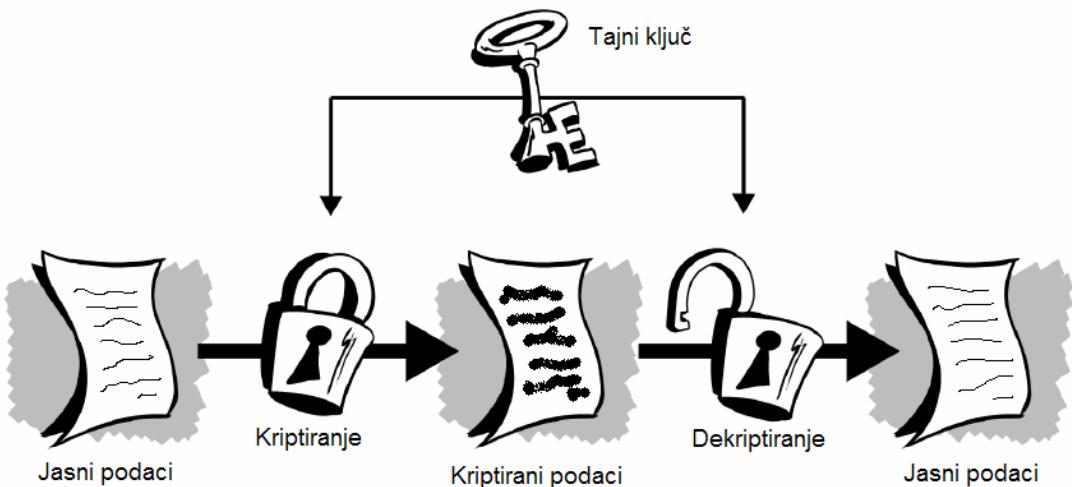
svega razvijala u vojne svrhe te se primjenjivala za potrebe očuvanja tajnosti vojnih i drugih povjerljivih informacija.

Tajnost informacija se osigurava primjenom općeg modela kriptiranja prikazanog na slici 3-8. Osnovno načelo zaštite tajnosti podataka jest pretvorba podataka. Izvorni, čitljivi podaci čiju se tajnost želi zaštititi zovu se *jasni podaci* (engl. *plaintext*). Tajnost jasnih podataka štiti se postupkom *kriptiranja* (engl. *encryption*). Postupak kriptiranja je pretvorba jasnih podataka u nečitljive i nerazumljive podatke. Rezultat postupka kriptiranja su *kriptirani podaci* (engl. *ciphertext*). Stvaranjem kriptiranih podataka skriva se informacija zapisana u jasnim podacima. *Dekriptiranje* (engl. *decryption*) je postupak obrnut kriptiranju, a njegovom primjenom nad kriptiranim podacima dobiva se izvorni sadržaj jasnih podataka.



Slika 3-8: Pretvorba podataka postupcima kriptiranja i dekriptiranja

S obzirom na odnos između postupka kriptiranja i postupka dekriptiranja, razlikuju se dvije vrste kriptografskih sustava: simetrični i asimetrični kriptografski sustavi. *Simetrični kriptografski sustavi* zasnivaju se na uporabi tajnoga ključa (engl. *secret key*). *Ključ* je vrijednost koja u algoritmu kriptiranja ili dekriptiranja služi kao parametar za pretvorbu podataka. U simetričnim kriptografskim sustavima isti tajni ključ služi za kriptiranje i za dekriptiranje. Tajni ključ se još naziva i simetrični ključ. Na slici 3-9 prikazan je simetrični kriptografski sustav.



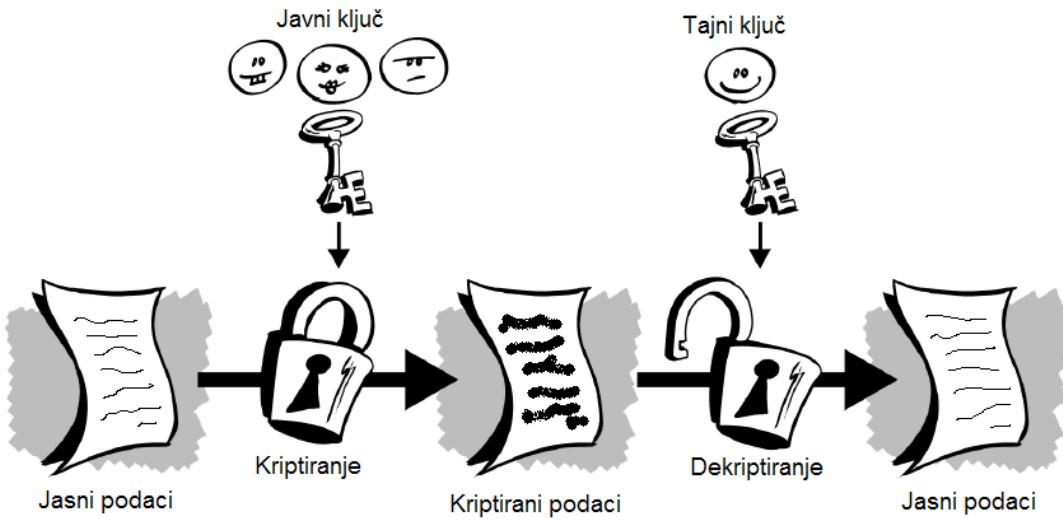
Slika 3-9: Simetrični postupak kriptiranja i dekriptiranja

Primjer jednostavnog simetričnog sustava je Cezarova šifra. Rimski car Julije Cezar na jednostavan je način kriptirao poruke koje je slao svojim generalima: svako slovo u poruci zamijenio je slovom koje je tri mesta dalje u abecedi. Na primjer, slovo A u poruci zamijenilo se slovom D, slovo B u poruci zamijenilo se slovom E i tako redom. Generali koji su znali tajni ključ, tj. pravilo pomaka za tri, mogli su pročitati izvorne podatke. Algoritmi zasnovani na simetričnom ili tajnom ključu često se primjenjuju u računarstvu, jer omogućuju učinkovitu zaštitu tajnosti podataka. Primjer često primjenjivanog algoritma simetričnog kriptiranja u računalnim sustavima je DES algoritam [27]. NIST organizacija za sigurnost računalnih sustava preporuča uporabu AES algoritma [28].

Za uspostavljanje simetričnog kriptografskog sustava koji predstavlja komunikacijski kanal između dviju strana potrebno dogovoriti tajni ključ. Tajni ključ se dogovara prije uspostave komunikacijskog kanala i poznat je samo stranama koje uspostavljaju komunikaciju. Znatan nedostatak simetrične kriptografije je nerješen problem tajnosti postupka kojim se dogovara tajni ključ. Taj nedostatak rješava se upotrebom asimetrične kriptografije.

Asimetrični kriptografski sustavi za kriptiranje i dekriptiranje primjenjuju dva različita ključa, koji zajedno čine jedinstveni par ključeva. Par ključeva zove se javno-tajni par i sastoji se od javnog ključa (engl. *public key*) i tajnog ključa (engl. *private key, secret key*). Javni ključ je javno dostupan svima, dok je tajni ključ poznat samo njegovu vlasniku. Javni ključ se primjenjuje za kriptiranje podataka, a podaci kriptirani javnim ključem mogu se dekriptirati samo uporabom tajnog ključa. Moguće je i obrnuti smjer kriptiranja, gdje se kriptira tajnim ključem, a dekriptira javnim ključem.

Slika 3-10 prikazuje postupak kriptiranja i dekriptiranja u asimetričnom kriptografskom sustavu. Za uspostavu komunikacijskog kanala potrebno je prethodno javno obznaniti javni ključ kako bi druga strana došla u posjed javnog ključa. Pomoću javnog ključa bilo tko može poslati povjerljive podatke vlasniku tajnog ključa kriptirajući ih njegovim javnim ključem. Zajamčeno je da nitko osim vlasnika tajnog ključa ne može dekriptirati i pročitati podatke kriptirane javnim ključem.



Slika 3-10: Asimetrični postupak kriptiranja i dekriptiranja

Primjer često primjenjivanog asimetričnog kriptografskog sustava je RSA [29]. Asimetrični postupci kriptiranja i dekriptiranja su računalno zahtjevni, te je komunikacija kriptirana asimetričnim postupkom kriptiranja računalno zahtjevna. Zbog tog nedostatka, asimetričnom se kriptografijom koristi u svrhu razmjene tajnog ključa između sugovornika. Nakon što sugovornici na siguran i povjerljiv način izmijene tajni ključ, dalje se u komunikaciji koristi tajni ključ za simetrično kriptiranje i dekriptiranje.

Osim očuvanja tajnosti podataka, asimetrična kriptografija nudi rješenje i za očuvanje nepovredivosti i izvornosti podataka. *Digitalno potpisivanje* je kriptografski postupak kojim se štiti nepovredivost i izvornost podataka. Digitalno potpisivanje zasniva se na asimetričnoj kriptografiji te na postupku sažimanja podataka.

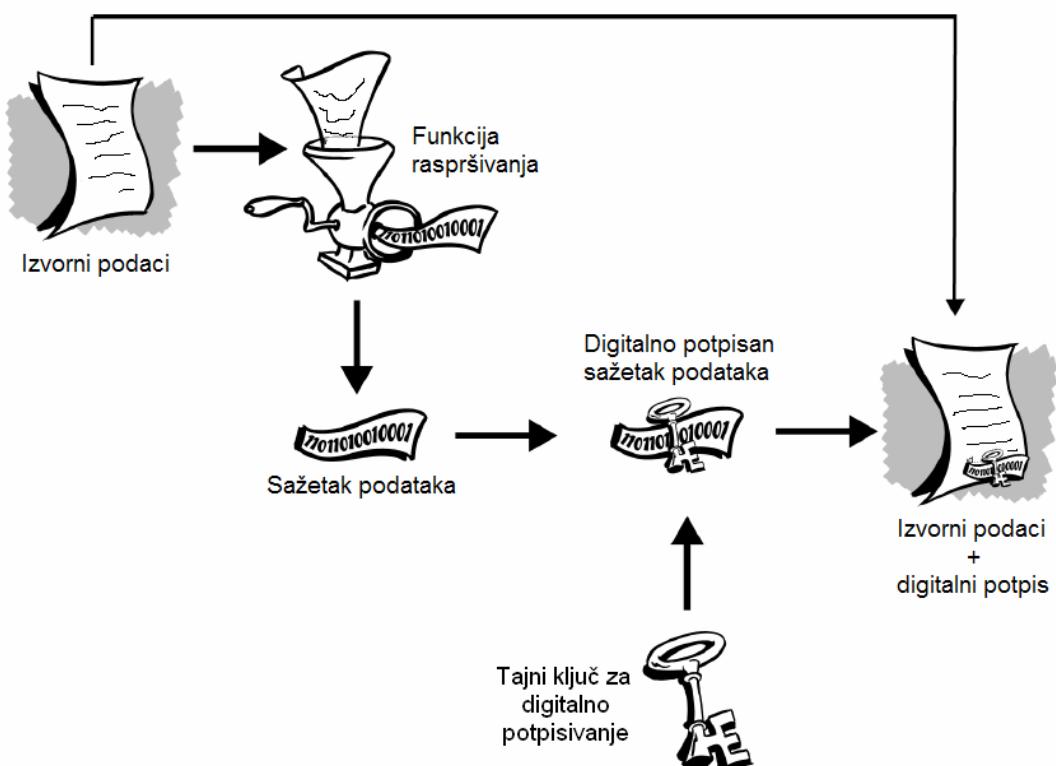
Sažimanje podataka postiže se primjenom funkcije *raspršivanja* (engl. *hash function*). Funkcija raspršivanja H prima kao ulaz poruku m proizvoljne duljine i kao rezultat stvara sažetak h u obliku niza nula i jedinica:

$$h = H(m)$$

Važno svojstvo postupka sažimanja je jednosmjernost, tj. nepostojanje obrnutog postupka. Jednostavno je pomoću funkcije raspršivanja iz m izračunati h , ali je računski

neizvedivo iz h naći m . Drugo važno svojstvo je osjetljivost funkcije na promjenu ulaznog podatka. Bilo kakva promjena ulaznog podatka znatno se odražava na rezultat, tj. rezultira posve drugčijim podacima na izlazu.

Zaštita nepovredivosti podataka postiže se digitalnim potpisivanjem podataka. Podaci koji se nastoje digitalno potpisati sažimaju se uporabom funkcije raspršivanja. Sažetak poruke kriptira se tajnim ključem pošiljatelja poruke i zajedno s izvornim podacima šalje se primatelju. Sažetak poruke kriptiran tajnim ključem i izvorna poruka čine digitalno potpisanoj poruci. Postupak stvaranja digitalno potpisane poruke prikazan je na slici 3-11.



Slika 3-11: Postupak digitalnog potpisivanja

Po primitku digitalno potpisane poruke, primatelj dekriptira primljeni sažetak poruke javnim ključem pošiljatelja. Na osnovi dekriptiranja sažetka primatelj može biti siguran da je izračunani sažetak kriptirao i poslao pošiljatelj. Dodatno, primatelj mora provjeriti je li izvorni podaci koje je primio uz sažetak doista odgovaraju sažetku koji je izračunao i kriptirao pošiljatelj. Provjeru izvodi izračunom sažetka izvornih podataka koje je primio, primjenjujući isti algoritam sažimanja koji je upotrijebio pošiljatelj. Usporedbom primljenog i izračunatog sažetka primatelj provjerava nepovredivost poruke. Ako se sažeci podudaraju, nepovredivost prenesene poruke je očuvana. Dodatno, osim nepovredivosti, primatelj je potvrđio vjerodostojnost poruke, odnosno autentičnost pošiljatelja koji ju je poslao. Ako je pri prijenosu poruke mrežom došlo do promjene u samo jednom bitu, dva

sažetka bitno će se razlikovati, te će postupak provjere ispravnosti digitalnog potpisa dati negativan ishod.

Uvođenjem asimetrične kriptografije javnih ključeva javlja se problem objave javnih ključeva, tj. povjerenja u javno objavljene ključeve. Problem se rješava uvođenjem *vjerodajnica* (engl. *credential*) koje izdaje povjerljiva treća strana [29, 31]. Vjerodajnice obuhvaćaju skup podatka o vlasniku čiju vjerodostojnost jamči izdavatelj. Javni ključevi obično se objavljaju vjerodajnicama u obliku *iskaznice* (engl. *certificates*). U najjednostavnijem obliku iskaznica je podatkovna struktura koja sadrži dvije vrste informacija: identitet vlasnika iskaznice te javnog ključa vlasnika iskaznice.

Ovlašteni izdavatelj iskaznica (engl. *certificate authority*) izdaje iskaznice osobama i organizacijama, te im uz iskaznicu izdaje i privatni tajni ključ. Posjedovanjem privatnog tajnog ključa koji odgovara javnom ključu u iskaznici, vlasnik dokazuje identitet naveden u iskaznici. Izdavatelj iskaznica digitalno potpisuje izdane iskaznice, te se ispravnost izdanih iskaznica provjerava provjerom potpisa. Najšire primjenjivani oblik iskaznica na Internetu je X.509 [32]. X.509 je standard koji određuje način izdavanja i sadržaj iskaznica.

3.2.2. Autentikacijski postupci

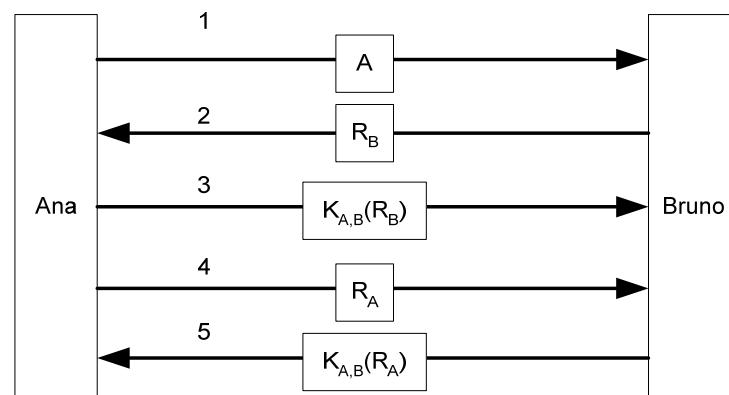
Autentikacija [26, 33] je postupak kojim se provjerava računalni identitet subjekta. Postupkom autentikacije provjerava se da li identitet kojim se subjekt predstavlja u računalu doista odgovara identitetu subjekta. Postupak autentikacije je sastavni dio postupka uspostave sigurnog komunikacijskog kanala, gdje se provjerava da li identitet sudionika s kojim se komunicira zaista odgovara identitetu kojim se sudionik predstavlja i želi nastupati u komunikaciji. Provjera računalnog identiteta subjekta je zahtjevan zadatak koji obuhvaća autentikacijske protokole i protokole zasnovane na kriptografiji.

Jednostavni i često upotrebljavani oblik autentikacije zasniva se na uporabi korisničkog imena i zaporce. Na primjer, prilikom uporabe osobnog računala, pomoću autentikacije korisničkim imenom i zaporkom započinje se rad na računalu. Poznavanjem zaporce koja je pripisana uz korisničko ime, korisnik potvrđuje svoj računalni identitet predstavljen korisničkim imenom.

Složeniji autentikacijski protokoli primjenjuju kriptografiju za provjeru identiteta sudionika. Autentikacijske protokole svrstava se u dvije skupine s obzirom na kriptografske metode koje primjenjuju: autentikacijske protokole zasnovane na simetričnoj kriptografiji i autentikacijske protokole zasnovane na asimetričnoj kriptografiji.

Pri razmatranju autentikacijskog protokola zasnovanog na simetričnoj kriptografiji, radi jednostavnosti se polazi od prepostavke da sudionici već dijele tajni simetrični ključ. Postupak kojim sudionici dolaze u posjed tajnog simetričnog ključa detaljno je opisan u [26]. U opisu autentikacijskog protokola skraćenim oznakama A i B označavaju se sudionici Ana i Bruno, a njihov zajednički tajni ključ označen je sa $K_{A,B}$. Protokol se zasniva na načelu kojim jedna strana izaziva drugu stranu na odgovor, pri čemu je odgovor točan samo ako druga strana poznaje tajni ključ.

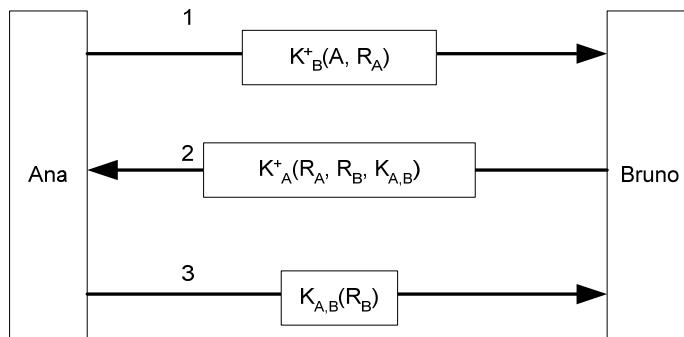
Na slici 3-12 prikazana je dvostrana autentikacija zasnovana na simetričnoj kriptografiji. Prvo Ana šalje svoj identitet Bruni (1) i time mu daje naznaku da želi s njim uspostaviti komunikacijski kanal. Bruno zatim šalje izazov R_B Ani (2). Izazov je najčešće oblika slučajnog broja. Na izazov Ana uzvraća odgovorom u kojem je primljeni izazov kriptiran tajnim ključem $K_{A,B}$ (3). Kada Bruno primi odgovor $K_{A,B}(R_B)$, dekriptira poruku uporabom tajnog ključa i provjerava sadrži li poruka izazov R_B . Ako potvrdi ispravan odgovor, onda Bruno zna da je s druge strane komunikacijskog kanala doista Ana, jer nitko drugi osim Ane ne može kriptirati izazov R_B ključem $K_{A,B}$. Međutim, Ana još uvijek nije potvrdila da je s druge strane komunikacijskog kanala doista Bruno. Stoga, Ana šalje izazov R_A (4) na koji Bruno mora uzvratiti odgovorom $K_{A,B}(R_A)$ (5). Nakon što Ana dekriptira odgovor tajnim ključem $K_{A,B}$ i provjeri R_A , sigurna je da komunicira s Brunom.



Slika 3-12: Dvostrana autentikacija zasnovana na simetričnoj kriptografiji

Pri razmatranju autentikacijskog protokola zasnovanog na asimetričnoj kriptografiji prepostavlja se da sudionici nisu lažno objavili javni ključ. Rješenje problema lažnog objavljivanja javnih ključeva zasniva se na uvođenju autentikacijskog poslužitelja [26]. Ponovo u komunikaciji sudjeluju Ana i Bruno označeni oznakama A i B. Anin javni ključ označava se oznakom K_A^+ , a Brunin javni ključ oznakom K_B^+ . Anin tajni ključ označava se oznakom K_A^- , a Brunin tajni ključ oznakom K_B^- . Simetrični tajni ključ koji dijele Ana i Bruno označava se sa $K_{A,B}$.

Na slici 3-13 prikazana je dvostrana autentikacija zasnovana na asimetričnoj kriptografiji. Smatra se da Ana posjeduje Brunin javni ključ, i obrnuto, da Bruno posjeduje Anin javni ključ. Ana započinje uspostavu komunikacijskog kanala slanjem prve poruke Bruni (1). U poruci koju Ana šalje Bruni nalazi se njezin identitet A i izazov R_A . Anina poruka je kriptirana Bruninim javnim ključem K_B^+ . Bruno jedini može dekriptirati poruku zakriptiranu ključem K_B^+ koristeći se svojim tajnim ključem K_B^- . Na primljeni izazov Bruno mora uzvratiti odgovorom koji, osim izazova R_A , sadrži izazov autentikacije Ani R_B te stvoreni tajni ključ $K_{A,B}$ koji Bruno želi dijeliti s Anom. Sadržaj odgovora na Anin izazov Bruno kriptira Aninim javnim ključem K_A^+ i šalje Ani (2). Koristeći se svojim tajnim ključem K_A^- , Ana dekriptira Brunin odgovor te užvraća odgovorom na Brunin izazov (3). Njezin odgovor sadrži primljeni izazov R_B kriptiran dijeljenim tajnim ključem $K_{A,B}$ koji joj je poslao Bruno. Time Ana dokazuje da je samo ona mogla dekriptirati izazov koji je kriptirao Bruno, te je uspostavljen kanal u kojem se komunikacija odvija na osnovi simetričnog kriptografskog ključa $K_{A,B}$.



Slika 3-13: Dvostrana autentikacija zasnovana na asimetričnoj kriptografiji

U komunikaciji se nastoji što rjeđe primjenjivati ključeve asimetričnog kriptosustava i time smanjiti vjerojatnost njihova otkrivanja. Stoga se obično nakon završetka autentikacije uspostavlja sjednica za tajnost komunikacije putem uspostavljenog komunikacijskog kanala. Prilikom uspostave sjednice stvara se dijeljeni ključ sjednice, ili sjednički ključ (engl. *session key*), koji osigurava tajnost komunikacije simetričnim postupkom kriptiranja poruka. Na slici 3-13 ključ $K_{A,B}$ predstavlja sjednički ključ uspostavljen nakon dvostrane autentikacije sudionika Ane i Brune.

Razlog stvaranja sjednice i sjedničkih ključeva je manja mogućnost napada ponavljanjem (engl. *replay attack*). Sjednički ključevi odbacuju se nakon što uspostavljeni komunikacijski kanal više nije potreban za komunikaciju, te se pri novoj potrebi komunikacije stvara novi kanal i novi sjednički ključevi. Korištenjem novog ključa sjednice prilikom svakog uspostavljanja kanala, sudionici su zaštićeni od ponovljenog odigravanja

uspostave sjednice. Ako uljez otkrije ključ jedne sjednice, ugrožena je sigurnost samo tijekom trajanja uspostavljene sjednice. S novom sjednicom stvara se novi ključ sjednice i sigurnost je ponovo uspostavljena.

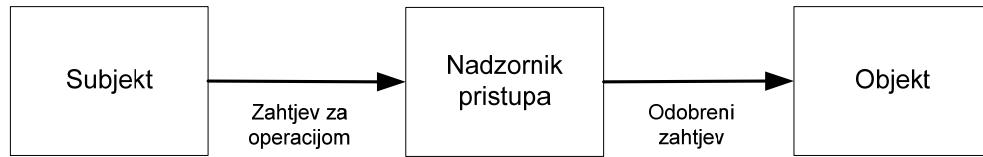
U raspodijeljenim računalnim sustavima čest je problem što korisnik mora ponavljati postupak autentikacije prilikom pristupa na različite poslužitelje. Stoga se teži ostvarivanju jedinstvene uspostave sjednice, ili SSO (engl. *Single-Sign-On*), što za sobom povlači razmjerenjivanje autentikacijskih podataka korisnika između različitih poslužitelja. Zahtjev za jedinstvenom sjednicom naročito je izražen u električnom poslovanju, gdje se primjenom tehnologije Web Services povezuju različita poduzeća (engl. *enterprise*). Stoga je nastao pokušaj standardiziranja jedinstvene sjednice pomoću XML podatkovnog formata koji je rezultirao SAML standardom. SAML standard objašnjen je u odjeljku 4.2.4.

3.2.3. Nadziranje pristupa

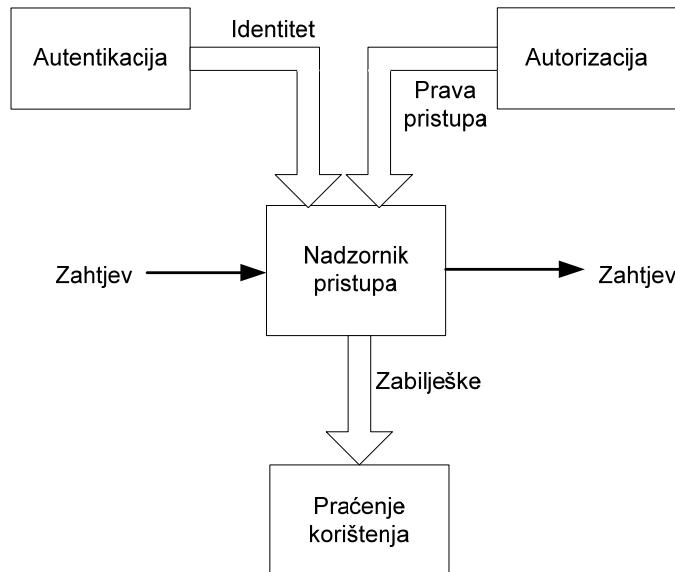
U komunikacijskom modelu korisnik-poslužitelj, nakon što se uspostavi siguran kanal između korisnika i poslužitelja, korisniku je omogućeno da poslužitelju sigurnim kanalom šalje svoje zahtjeve. Zahtjevi uključuju izvođenje operacija nad sredstvima koja nadzire poslužitelj. Poslužitelj nadzire pristupe sredstvima i izvodi samo one zahtjeve za koje korisnik ima odgovarajuća prava pristupa (engl. *access rights*).

Nadzor pristupa je mehanizam koji provjerava prava pristupa korisnika, donosi odluke o pravima pristupa, te sukladno odlukama omogućuje ovlašteni pristup sredstvima ili sprječava neovlašteni pristup sredstvima. Postupci autorizacije su usko povezani s postupkom nadzora pristupa, te se pojmovi nadzora pristupa i autorizacije često isprepleću u literaturi.

Opći model za nadzor pristupa prikazan je na slici 3-14. Glavni sudionici u procesu nadzora pristupa su subjekt, objekt ili sredstvo te nadzorni sustav. Subjekte su računalni procesi koji se izvode u korist korisnika. Subjekti postavljaju zahtjeve za pristup objektima. Izraz objekt ili sredstvo ravnopravno se primjenjuje u značenju podataka, procesa, programskih objekata i usluga. Pomoću njih se obilježava stanje u sustavu ili se ostvaruju operacije koje se koriste preko sučelja. Pristup objektima štiti *nadzorni sustav* (engl. *reference monitor*). Svaki put kada subjekt zahtijeva izvođenje operacije nad objektom, poziv provjerava nadzorni sustav. Ako subjekt ima pravo izvođenja operacije koju je zatražio, nadzorni sustav odobrava zatraženi zahtjev.

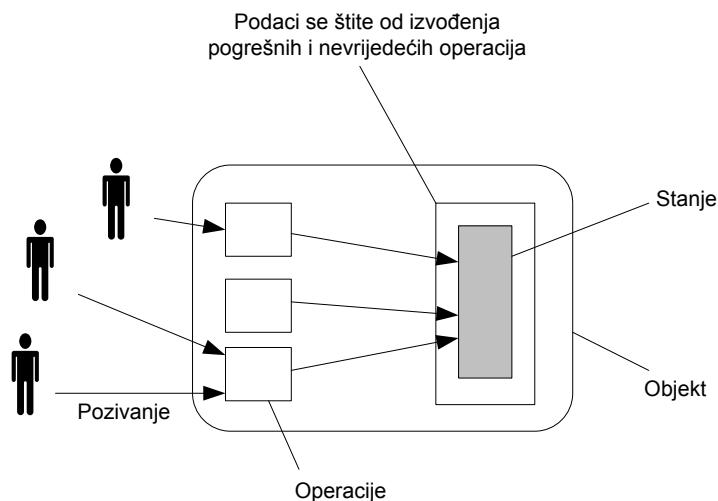


Slika 3-14: Opći model nadzora pristupa



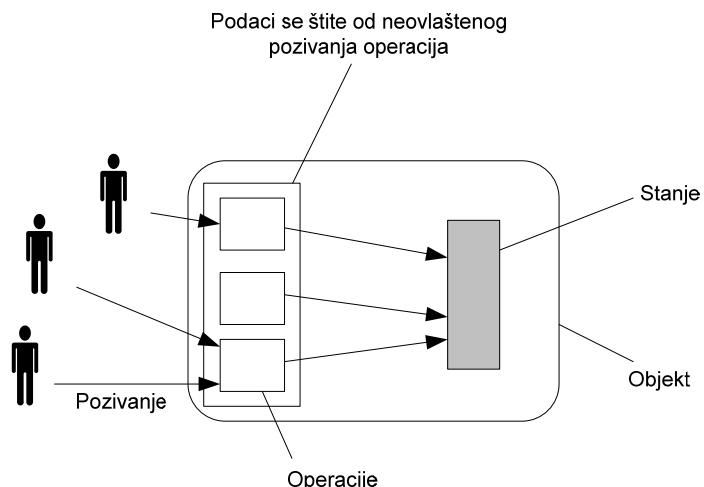
Slika 3-15: Tijek upravljačkih informacija uključenih u nadzor pristupa

Tijekom nadzora pristupa koriste se upravljačke informacije koje su povezane s drugim postupcima uspostave sigurnosti. Slika 3-15 prikazuje tijek upravljačkih informacija tijekom nadzora pristupa. Od sustava za autentikaciju nadzornik pristupa dobiva subjekto identitet. Osim subjektova identiteta, nadzornik primjenjuje prava pristupa dodijeljena subjektu tijekom postupka autorizacije. Prikupljene informacije koristi pri donošenju odluke o propuštanju ili odbijanju subjektovog zahtjeva. Uz provođenje donesenih odluka, postupkom praćenja uporabe nadzornik pristupa bilježi informacije o subjektovu zahtjevu.



Slika 3-16: Zaštita ispravnosti podataka pri izvođenju operacija

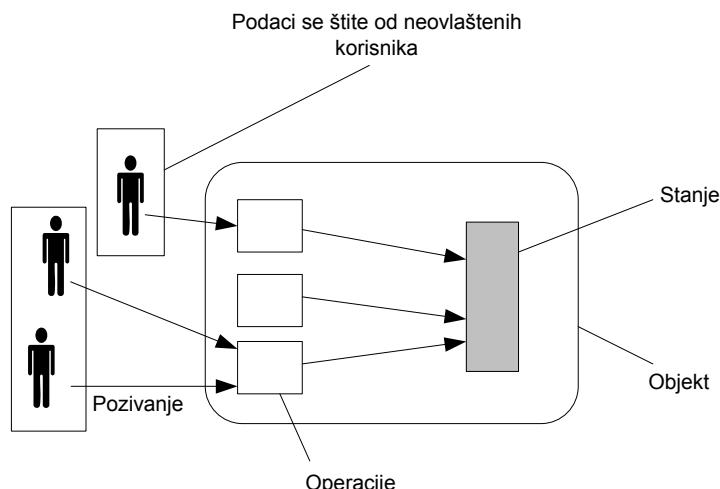
Postoje tri načina nadzora pristupa objektima. Prvi način nadzora pristupa prikazan na slici 3-16 usmjeren je izravno zaštiti podataka obuhvaćenih u objektima sustava. Bez obzira na različite operacije koje se izvode nad podacima, glavni zadatak nadzora pristupa tijekom izvođenja operacija je osigurati i uspostaviti ispravnost podataka u objektu. Takva vrsta zaštite javlja se uglavnom u sustavima za upravljanje bazom podataka. U bazama podataka postavljaju se različita pravila i uvjeti koji moraju vrijediti i koje se provjerava tijekom promjene podatka.



Slika 3-17: Zaštita od neovlaštenog pozivanja operacija

Drugi način nadzora pristupa prikazan na slici 3-17 usmjeren je zaštiti na osnovi operacija koje se smiju pozvati. U tom se slučaju navodi tko smije pozvati pojedine operacije da bi pristupio podacima. Na primjer, u objektno-usmjerenom sustavu se za svaku operaciju

izloženu uporabi definira kojim se korisnicima dopušta pozvati tu operaciju. Dodatno je moguće zaštititi cijelokupno sučelje ili cijelokupni objekt, te se time određuje različita *razlučivost nadzora pristupa* (engl. *access control granularity*).



Slika 3-18: Zaštita pristupa od strane neovlaštenih korisnika

Treći način nadzora pristupa prikazan na slici 3-18 usmjerava nadzor pristupa izravno prema korisnicima. Samo se određenim korisnicima dopušta pristup bez obzira na operacije koje žele izvesti. Pristup se zabranjuje svim drugim korisnicima. Na primjer, pristup bazi podataka u banci zabranjuje se svakome, osim rukovoditeljima. Primjer su i Web stranice mnogih fakulteta gdje je pristup određenim podacima i primjenskim sustavima dopušten samo osoblju fakulteta, a studentima je pristup zabranjen.

Razvijeni su različiti modeli nadzora pristupa koji odgovaraju zahtjevima različitih sigurnosnih odredbi. Razlike među glavnim modelima nadzora pristupa opisane su u odjeljku 3.3. Bez obzira na razlike u načinima zapisa prava pristupa u različitim modelima nadzora pristupa, najčešće se nadzor pristupa tumači na modelu matrice pristupa.

Matrica pristupa (engl. *access control matrix*) uobičajeni je i osnovni način opisivanja prava pristupa. Subjekti se predstavljaju recima matrice pristupa, a objekti se predstavljaju stupcima. Elementi matrice pristupa sadrže dozvole prava pristupa subjekta iz odgovarajućeg reda objektu iz odgovarajućeg stupca. Drugim riječima, ako nešto nije dopušteno upisom u matricu, onda je zabranjeno. Elementi od kojih se sastoji matrica pristupa prikazani su u tablici 3-1.

Problem primjene matrice pristupa jest njezin rast s brojem subjekata (korisnika) i objekata. Mnogi elementi matrice ostaju nepotpunjeni, jer subjekt u općem slučaju nema pristup velikom broju objekata, te je matrica pristupa nepraktična za zapisivanje prava

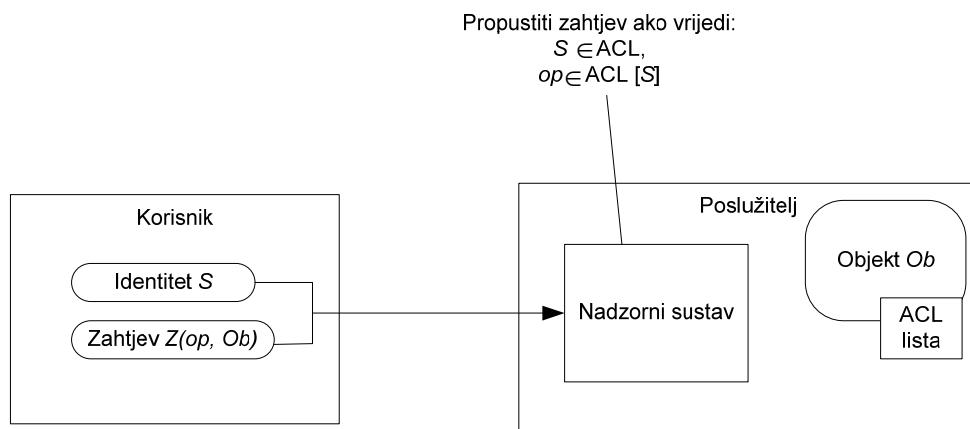
pristupa. Zbog toga se matrica najčešće prikazuje u obliku lista prava pristupa objektu i propusnica za pristup objektu.

Tablica 3-1: Primjer zadavanja prava pristupa matricom pristupa

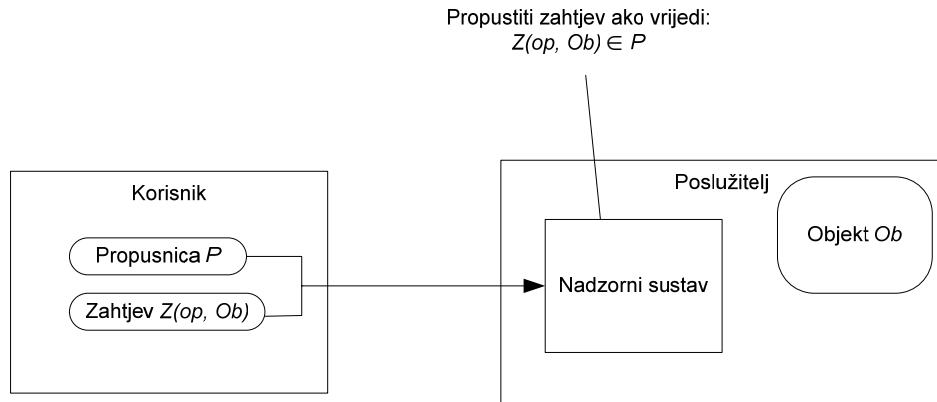
Subjekti	Objekt 1 (Datoteka)	Objekt 2 (Pisač)	Objekt 3 (Funkcija)	...	Objekt M (Operacija sustava)
Korisnik 1	Čitanje Pisanje	Poništavanje ispisa	Pozivanje	...	Mijenjanje sata sustava
Korisnik 2	-	Ispisivanje	-	...	Podizanje sustava
.
.
.
Korisnik N	Izvođenje	-	Pozivanje	...	Gašenje sustava

Lista prava pristupa objektu ili *ACL lista* (engl. *access control list, ACL*) predstavlja stupce matrice pristupa i sadrži popis korisnika koji imaju određena prava pristupa objektu. Svaki objekt ima pripadajuću ACL listu. Time se matrica pristupa raspodjeljuje na osnovi stupaca po svim objektima. *Propusnice* (engl. *capability tickets*) predstavljaju retke matrice pristupa. Svakom subjektu je dodijeljena propusnica za pristup objektima. Propusnica sadrži popis objekata za koje subjekt ima pravo pristupa.

Razlika u načinu primjene ACL liste i propusnice za nadzor pristupa objektima prikazana je na slikama 3-19 i 3-20. Kada primjenjujući ACL listu subjekt S pokuša pristupiti objektu Ob operacijom op , nadzorni sustav na poslužitelju provjerava ACL listu prava pristupa objektu Ob . Ako se subjekt koji je pokušao pristupiti objektu nalazi u listi prava pristupa objektu i ima traženo pravo pristupa, pristup mu se odobrava. U protivnom, subjektu se ne odobrava pristup objektu.



Slika 3-19: Nadzor pristupa pomoću ACL liste



Slika 3-20: Nadzor pristupa pomoću propusnice

U slučaju uporabe propusnica, subjekt za pristup objektu predočuje nadzornom sustavu propusnicu P . Sadrži li subjektova propusnica zatraženo pravo pristupa op objektu Ob , subjektu se odobrava pristup. Ako subjektova propusnica ne sadrži pravo pristupa objektu, pristup objektu se ne odobrava. Poslužitelj u ovom slučaju čak ne mora poznavati subjekta, nego samo provjerava predočenu propusnicu uz uvjet povjerenja onome tko je izdao propusnicu.

3.2.4. Postupak autorizacije

Pojam autorizacije se koristi u dvojakom značenju. Autorizacija u smislu dozvole (engl. *permission*) ima značenje prava pristupa za izvođenje određenih operacija. Na primjer, u tom smislu "*Ana ima pravo pristupa bazi podataka*" označava Aninu dozvolu pristupa bazi. "*Ana ima zabranu pristupa bazi*" primjer je negativne autorizacije (engl *negative authorizaton*). Autorizacija u smislu postupka uspostave sigurnosti odnosi se na proces kojim se u sustavu dodjeljuje, oduzima i upravlja pravima pristupa subjekata.

U raspodijeljenim računalnim sustavima proces dodjele i upravljanja pravima pristupa je složen i zamoran administratorski posao kojeg je moguće automatizirati i pojednostaviti autorizacijskim postupcima. Nepraktičan način administriranja prava pristupa jest postavljati prava pristupa za svakog korisnika na svakom računalu u sustavu. Iako vremenski dugotrajan i podložan pogreškama, takav se način administriranja prava pristupa primjenjuje u *mrežnim operacijskim sustavima* [34] (engl. *network operating systems*). Prikladniji način administriranja prava pristupa jest raspodijeliti prava pristupa upotrebom propusnica. Propusnice raspodjeljuju prava pristupa subjektima. Primjer uporabe propusnice za kraći zapis prava pristupa matrice pristupa opisan je u odjeljku 3.2.3. Propusnica ima ulogu ulaznice koju subjekt predočava nadzorniku pristupa kada želi pristupiti određenom objektu. Propusnice je prilikom izdavanja nužno digitalno potpisati

kako bi se onemogućila njihova izmjena od strane neovlaštenih subjekata. Prilikom uporabe propusnice, poslužitelj sredstva ne mora nužno poznavati subjekta. Dovoljno je na poslužitelju sredstva provjeriti dobivenu propusnicu, uvjeriti se da propusnica nije izmijenjena od trenutka izdavanja te da propusnica odobrava zatraženi pristup.

Poopćenje propusnice koje se primjenjuje u novije vrijeme jest *iskaznica svojstava* (engl. *attribute certificate*). Za razliku od iskaznice javnih ključeva opisane u odjeljku 3.2.1, iskaznice svojstava uz identitet vlasnika iskaznice popisuju svojstva vlasnika iskaznice. U posebnom slučaju, iskaznice svojstava koriste se za popisivanje prava pristupa vlasnika iskaznice nad određenim sredstvima. Poput drugih iskaznica, iskaznice svojstava izdaju posebni ovlašteni izdavatelji – *izdavatelji iskaznica svojstava* (engl. *attribute certification authorities*). Izdavatelji iskaznica svojstava potpisuju izdane iskaznice svojstava i time sprječavaju njihovo neovlašteno mijenjanje od strane vlasnika kojima se iskaznica izdaje ili zlonamjernih napadača koji nisu vlasnici iskaznice.

Prenošenje prava pristupa (engl. *delegation*) je način dodjele i upravljanja pravima pristupa u raspodijeljenim računalnim sustavima, gdje se podskup prava pristupa prenosi s jednog procesa na drugi. Time se omogućuje jednostavnije upravljanje i raspodjeljivanje radnih procesa bez posljedica na sigurnost. Na primjer ako korisnik želi rezervirati tiskanje velikog dokumenta u vrijeme kada to nikome neće smetati, onda mora pisaču privremeno prenijeti pravo čitanja dokumenta koji je ostavio za tiskanje.

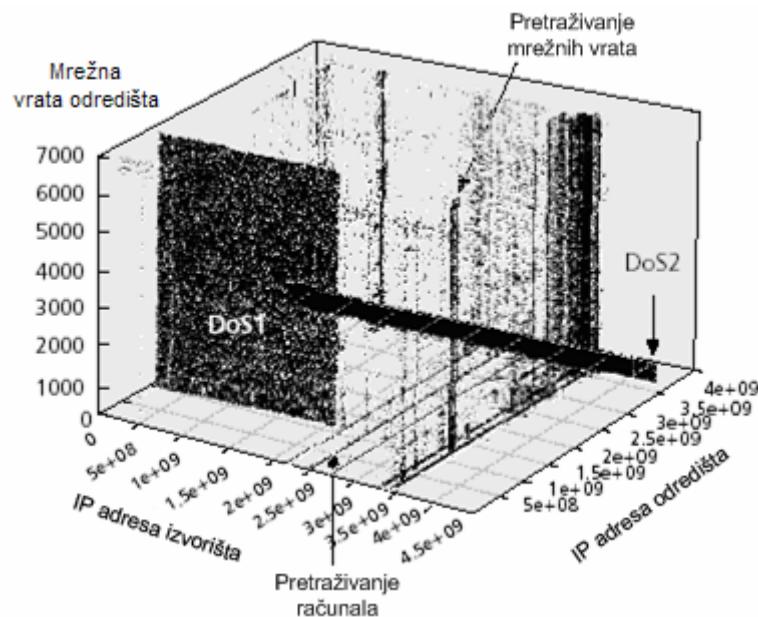
Prenošenje prava pristupa ostvaruje se na nekoliko načina. Način opisan u [35] primjenjuje stvaranje *zastupnika prava* (engl. *proxy*). Zastupnik prava je sredstvo u obliku značke kojim subjekt dodjeljuje prava pristupa drugom subjektu. Subjekt, odnosno proces, stvara zastupnika prava i dodjeljuje mu podskup vlastitih prava pristupa. Slično, zastupnik prava stvara novog zastupnika prava koji ima ista ili manja prava u odnosu na izvornog zastupnika prava.

3.2.5. Praćenje korištenja

Praćenje korištenja je postupak tijekom kojeg se bilježe radnje i akcije koje korisnici izvode i zahtijevaju od sustava. U općem slučaju praćenje se izvodi u svrhu *naplate korištenja* (engl. *billing*) i izrade *korisničkog poosobljavanja* (engl. *user profile*). Za potrebe sigurnosti sustava, praćenje korištenja omogućuje analizu napada na sustav, pronalaženje i uklanjanje sigurnosnih propusta te otkrivanje uljeza. Međutim, u sigurnosnim sustavima često se zanemaruje praćenje korištenja, što predstavlja neodgovoran pristup organiziranja sigurnosti sustava.

Praćenjem se nastoje zabilježiti neuspješni i uspješni napadi na sustav, jer na osnovi tih napada stvara se slika o sigurnosti i sigurnosnim propustima u sustavu. Nažalost, nije uvijek moguće zabilježiti napade koji su bez uspjeha pokušali ugroziti sigurnost sustava. Međutim, mnogo je ozbiljniji problem kada se ne uspije zabilježiti napad koji je ugrozio sigurnost sustava.

Zabilježeni neuspješni napadi mogu uputiti na smjer suzbijanja česte napadačke aktivnosti i poboljšavanja rada sustava. Nakon zabilježenog uspješnog napada, moguće je analizirati scenarij na koji je sustav ranjiv. Jedna mogućnost je da analizu izvode sigurnosni stručnjaci, dakle čovjek. Druga je mogućnost da analizu radi računalo te se onda primjenjuju različite računalne tehnike kao što su *dubinska analiza podataka* (engl. *data mining*) i raspoznavanje uzorka (engl. *pattern recognition*). Primjer opisan u [36] vizualizacijom olakšava analizu napada zagušenja poslužitelja (engl. *denial of service, DoS*) (slika 3-21). Podaci se prikupljaju iz *glavne komunikacijske mreže* (engl. *backbone*) te se napade vizualizira u trodimenzionalnom prostoru na osnovi tri parametra dobivena iz komunikacijskih paketa. Parametri su *IP adresa izvorišta* (engl. *souce IP address*), *IP adresa odredišta* (engl. *destination IP address*) i *mrežna vrata odredišta* (engl. *destination port*). U trodimenzionalnom prostoru na slici se uočavaju napadačke aktivnosti *pretraživanja mrežnih vrata* (engl. *portscan*), *pretraživanja računala* (engl. *hostscan*) te napadi prekida dostupnosti poslužitelja zagušivanjem poslužitelja (DoS napadi).



Slika 3-21: Vizualizacija napadačkih aktivnosti u 3D prostoru

Postupkom praćenja korištenja povećava se mogućnost otkrivanja napadača, što utječe na odvažnost napadača. Napadač mora osmisliti način prikrivanja tragova napada da

ga se na osnovi njih ne bi otkrilo. Napad mora zbog toga biti pažljivo isplaniran prilikom čega napadač često gubi odvažnost i odustaje od napada. Time se smanjuje vjerojatnost i broj napada na sustav. Napadi se često događaju iznutra od korisnika koji pripada sustavu. Na primjer, u banci se događa da službenici prekorače svoja dopuštenja ili iskoriste svoj položaj, te zanemarujući poslovnu etiku ponašanja izvedu novčane transakcije bez znanja stranke. Sve transakcije se u sustavu bilježe pod imenom službenika koji izvodi transakciju. Jednostavnim pregledavanjem obavljenih transakcija moguće je provjeriti tko je izveo koju transakciju i kada. Davanjem do znanja da se sve transakcije i korištenje bankovnog sustava prati, smanjuje se vjerojatnost mogućih napada iznutra.

3.3. Modeli nadzora pristupa

U primjeni su razvijeni različiti modeli nadzora pristupa koji su prilagodeni potrebama različitih sigurnosnih odredbi. Na primjer, vojni sustavi koji prvenstveno nastoje zaštитiti tajnost informacija zahtijevaju drugačiji model nadzora pristupa nego poslovni sustavi koji pružaju širok skup različitih funkcionalnosti korisnicima. Razlike među modelima nadzora pristupa uočljive su u načinima zadavanja pravila pristupa i različitim mehanizmima provedbe nadzora. Zapisani skup pravila pristupa na osnovi kojih se provodi mehanizam nadzora predstavlja *odredbe nadzora pristupa* (engl. *access control policy*). Na osnovi podataka zapisanih u odredbama nadzora pristupa mehanizmi provedbe nadzora izvode nadzor pristupa. Tri glavna modela nadzora pristupa su: model *razlikovnog nadzora pristupa*, model nadzora *zasnovan na ulogama* i model *dosljednog nadzora pristupa*.

3.3.1. Model razlikovnog nadzora pristupa

Odredbe modela *razlikovnog nadzora pristupa* (engl. *Discretionary Access Control, DAC*) [22] zasnivaju se na razlikovanju identiteta subjekta te razlikovanju prava pristupa za svakog od subjkata. Osnova DAC modela je matrica pristupa koja služi kao okvir za definiranje prava pristupa i donošenje odluka o dozvoli pristupa. DAC model predstavlja stanje sustava kao uređenu trojku (S, O, A), gdje je S skup subjkata, O skup objekta, i A je matrica pristupa opisana u odjeljku 3.2.3. Element matrice $A[s, o]$ sadrži prava koja subjekt s ima nad objektom o . Iako je matrica pristupa glavni mehanizam za donošenje odluka u DAC modelu, postoji nekoliko proširenja tog mehanizma (kao npr. negativne autorizacije, prenošenje prava pristupa i slično).

Negativne autorizacije (engl. *negative authorization*) uvode se radi izražavanja zabrane prava pristupa. Time se rješava nedostatak osnovnog modela matrice pristupa u kojem se zabrana pristupa podrazumijeva. Ako se zabrana pristupa podrazumijeva,

mehanizam zaključivanja donosi odluku o zabrani pristupa nakon bezuspješnog pretraživanja svih dozvola pristupa. Uvođenje negativnih autorizacija ubrzava nadzor pristupa, ali dovodi do pojave nejednoznačnosti prava pristupa sukobljavanjem s dozvolama pristupa. Sukob između negativnih autorizacija i dozvola prava pristupa rješava se davanjem prednosti negativnim autorizacijama. Kad god subjekt ima negativnu autorizaciju i dozvolu prava pristupa, subjektu se zabranjuje pristup.

Osnova DAC modela je mehanizam kojim subjekt dodjeljuje prava pristupa drugim subjektima i na njih prenosi prava pristupa. *Prenošenje prava pristupa* (engl. *delegation*) ostvaruje se upotrebom propusnica. Prenošenje prava pristupa ključan je mehanizam kojim DAC model podupire ostvarivanje jednostavne autorizacije i administracije sustava, zbog čega mnogi sustavi usvajaju DAC. Prenošenjem prava administracijskog pristupa posebnim subjektima se povjerava proces administracije upisivanja ili uklanjanja prava pristupa i upravljanja mehanizmom nadzora pristupa. Postupci administracije prava pristupa dijele se na centraliziranu administraciju i vlasničku administraciju. *Centralizirana administracija* (engl. *centralized administration*) dopušta samo *administratorima*, odnosno povlaštenim subjektima, prava dodjeljivanja i oduzimanja autorizacija. U vlasničkoj administraciji (engl. *ownership administration*) tvorac ili vlasnik objekta dodjeljuje i oduzima prava pristupa objektu. Vlasnička administracija često se javlja u svojstvu *prenošenja prava administriranja* (engl. *administration delegation*). Time je omogućeno da vlasnik objekta subjektima dopusti pravo dodjeljivanja i oduzimanja autorizacije za pristup tom objektu. Na taj se način omogućuje raspodjeljivanje administriranja autorizacija.

3.3.2. Model nadzora zasnovan na ulogama

Model nadzora pristupa *zasnovan na ulogama* (engl *Role-Based Access Control, RBAC*) [22, 37] oblikuje nadzor pristupa sustavu prema načinu podjele odgovornosti u poslovnim organizacijama. Osnovni smisao RBAC modela je pojednostavljenje autorizacije i administracije prava pristupa definiranjem *uloga* (engl. *role*). Uloga označava određenu funkciju subjekta unutar organizacije te definira skup akcija i odgovornosti vezanih uz danu funkciju. Na primjer, uloga subjekta unutar organizacije može biti tajnik, zaposlenik, finansijski voditelj, itd. U RBAC modelu ulogama se izravno pridružuju prava pristupa te se njima oblikuju odredbe nadzora pristupa. Prava pristupa potrebna za izvođenje operacija ne pridružuju se korisnicima izravno, već ih se dodjeljuje preko uloge koja obuhvaća dopuštenje izvođenja operacije. Postupkom dodjele uloge prava pristupa korisnika definirana su ulogom koja mu je dodijeljena. Uloge posreduju u pristupu korisnika objektima. Svaki korisnik je u službi svoje uloge i na osnovi nje pristupa objektima.

Objedinjenje prava pristupa ulogama pojednostavljuje administriranje prava pristupa. Ako korisnik treba izvesti određenu operaciju, korisniku je potrebno dozvoliti obnašanje odgovarajuće uloge. Dodjela uloge jednostavniji je postupan od izravnog pridruživanja pojedinačnih prava pristupa korisniku. Također, kada korisnik promijeni svoju ulogu unutar organizacije, potrebno je samo oduzeti korisniku dozvolu obnašanja uloge.

Osim osnovnog RBAC modela, s vremenom su razvijena različita proširenja. Sukladno razlikama u skupu podržanih mogućnosti postoji osnovni, hijerarhijski, ograničavajući i simetrični RBAC model. Simetrični RBAC je spoj hijerarhijskog i ograničavajućeg RBAC modela. Hijerarhija uloga dopušta predstavljanje relacije uloga-poduloga. To je prirodan način organiziranja uloga koji odražava način na koji su ustrojene obveze i prava u organizaciji. Hijerarhijom se omogućuje nasljeđivanje među ulogama i višestruka ponovna uporaba već definiranih prava pristupa.

U RBAC je uvedeno i postavljanje ograničenja (engl. *constraints*) brojnosti uloga dodijeljenih korisniku i brojnosti ulogom obuhvaćenih prava pristupa. Ograničenja pojednostavljaju ostvarenje zahtjeva i načela dobro definiranog nadzora pristupa kao što su načelo najmanje privilegije i načelo razdvajanja dužnosti. *Načelo najmanje privilegije* (engl. *least privilege*) rješava sukob pozitivnih i negativnih autorizacija davanjem prednosti negativnim autorizacijama. *Razdvajanje dužnosti* (engl. *separation of duty, SoD*) ograničava subjektu mogućnost skupljanja i centraliziranja velikog broja prava pristupa. Centralizirana prava pristupa omogućuju potencijalnom uljezu prisvajanje velikog broja prava pristupa ako se uspije predstaviti u ime subjekta s velikim brojem prava pristupa.

3.3.3. Model dosljednog nadzora pristupa

Model *dosljednog nadzora pristupa* (engl. *Mandatory Access Control, MAC*) [22, 23] potječe iz vojnih krugova. Nastao je u okruženju gdje postoji središnji zakonodavac ovlasti (engl. *central authority*) i strogo utvrđena pravila pristupa koja se dosljedno provode. MAC model ostvaruje nadzor pristupa usmjeren prema problemu povjerljivosti podataka i ograničenju tijeka informacija u računalnim sustavima. Odredbama MAC modela nadzire se pristup objektima na osnovi unaprijed definirane razredbe subjekata i objekata u sustavu. Objekti su pasivni sudionici i sadrže informacije koje se štiti, a subjekti su aktivni sudionici koji pristupaju objektima. Osnovu modela kojim se odlučuje ima li subjekt pravo pristupiti objektu osmisili su Bell i LaPadula [38].

Tablica 3-2: Popis operacija koje su definirane u MAC modelu

Operacija	Pregled podataka	Izmjena podataka
-----------	------------------	------------------

	operacijom	operacijom
Izvođenje	-	-
Čitanje	+	-
Dodavanje	-	+
Pisanje	+	+

Objekti se označuju skupom O , a subjekti skupom S , te vrijedi relacija S je podskup od O . U tablici 3-2 definirane su operacije koje subjekt može izvesti nad objektom. Operacija izvođenje obavlja se bez uvida u podatke i bez mijenjanja podataka. Operacija čitanje izvodi se s uvidom u podatke ali bez mijenjanja podataka. Operacija dodavanje izvodi se bez uvida u podatke uz mogućnost mijenjanja podataka. Operacija pisanje izvodi se s uvidom u podatke i s mogućnošću mijenjanja podataka.

U modelu se definiraju dvije komponente sigurnosti: *vrsta povjerljivosti* i *skup kategorija*. Vrsta povjerljivosti P potpuno je uređeni skup sastavljen od elemenata *stroga tajna* (VT), *tajna* (T), *povjerljiva informacija* (PI) i *nekritična informacija* (NI). Među njima vrijedi relacija veće povjerljivosti $VT > T > PI > NI$. Skup kategorija je neuređeni skup (npr. NATO, nuklearno, vojno), a označuje se sa K . Uvodi se pojam *razine* oznake R , koja se sastoji od definiranih dviju komponenti sigurnosti. Razina R zapisuje se kao $R = (p, k)$, gdje je $p \in P$ i k je podskup od K . Među razinama se definira uređena relacija \propto na sljedeći način:

$$R = (p, k) \propto R' = (p', k') \Rightarrow p \leq p', k \subset k'$$

Svakom objektu o u sustavu pripisana je jedna razina $R(o)$, a svakom subjektu s pripisuju se dvije razine: $R(s)$ i $maxR(s)$. Veličina $maxR(s)$ zove se ovlast (engl. *clearance*). Ovlast $maxR(s)$ maksimalna je razina sigurnosti za subjekt s i ona je statički definirana, tj. ne mijenja se. $R(s)$ je trenutna sigurnosna razina subjekta s . Valja istaknuti da relacija $R(s) \propto maxR(s)$ vrijedi uvijek. Trenutna razina sigurnosti se mijenja u granicama maksimalnih ovlasti subjekta. Prava pristupa subjekta određuju se na osnovi četiriju definiranih pravila:

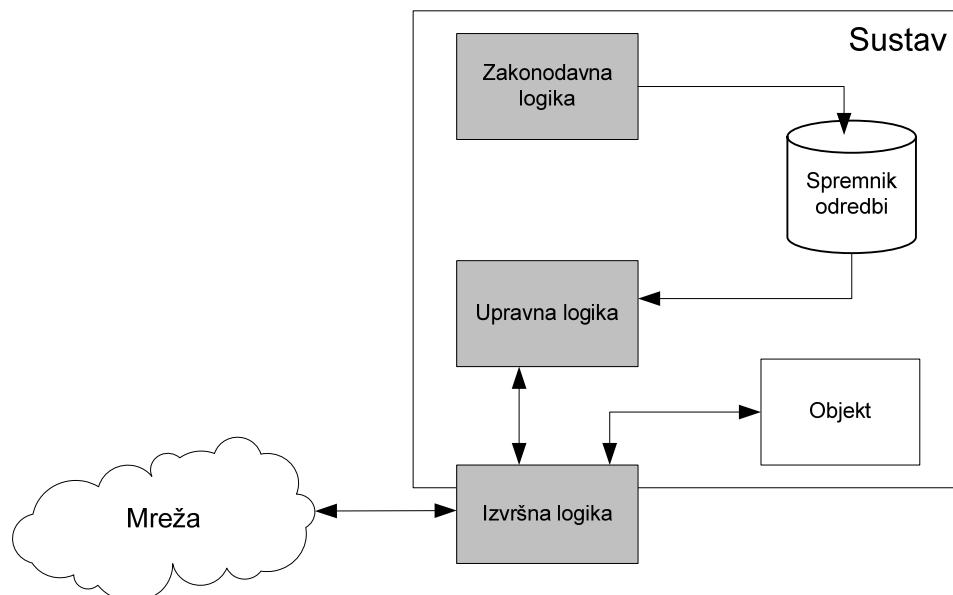
1. Subjekt dobiva pravo čitanja samo onih objekata koji su ispod njegove maksimalne razine ovlasti ili su na njegovoj razini ovlasti. Drugim riječima, da bi subjekt s mogao dobiti pravo uvida u podatke objekta o , mora vrijediti $R(o) \propto maxR(s)$.
2. Da bi subjekt s pročitao objekt o , mora biti ispunjen uvjet $R(o) \propto R(s)$.
3. Da bi subjekt s pisao u objekt o , mora biti ispunjen uvjet $R(o) = R(s)$.
4. Da bi subjekt s dodao nešto u objekt o , mora biti ispunjen uvjet $R(s) \propto R(o)$.

3.4. Načela oblikovanja nadzora pristupa

Najveći doprinosi razvoju nadzora pristupa dolaze od strane komercijalnih proizvođača mrežne opreme, kao što su CISCO, HP, Motorola i drugi, te od strane poslovne zajednice gdje se ističu BEA, SAP, Microsoft, Sun, IBM i drugi. Na osnovi njihova rada i istraživanja oblikovana su načela i smjernice za poboljšanje sigurnosti i jednostavniju izgradnju sustava nadzora pristupa. Istim se dva načela u skladu kojih treba postupati tijekom oblikovanja sustava nadzora pristupa. Prvo je razdvajanje funkcionalnosti nadzora pristupa, a drugo je uspostava pristupnih točaka.

3.4.1. Razdvajanje funkcionalnosti

IETF radni okvir nadzora pristupa (engl. *IETF access control framework*) [39, 49] služi kao osnova za oblikovanje arhitekture sustava nadzora pristupa. Na slici 3-22 prikazana je arhitektura sustava nadzora pristupa koju definira IETF radni okvir. Razlikuju se tri zasebne funkcionalnosti: zakonodavna logika, upravna logika i izvršna logika. *Zakonodavna logika* (engl. *policy management tool*) pruža sučelje koje omogućuje mijenjanje aktivnih odredbi u sustavu te zadavanje i spremanje novih odredbi u sustav. Odredbe se spremaju u spremnik odredbi. Spremnik odredbi obično se ostvaruje kao hijerarhijska baza podataka [42, 44] kojoj se pristupa LDAP protokolom (*Lightweight Directory Access Protocol*) [40, 41]. Druga mogućnost ostvarivanja spremnika odredbi zasniva se na jeziku i protokolu XACML (*eXtensible Access Control Language*) [1, 54].



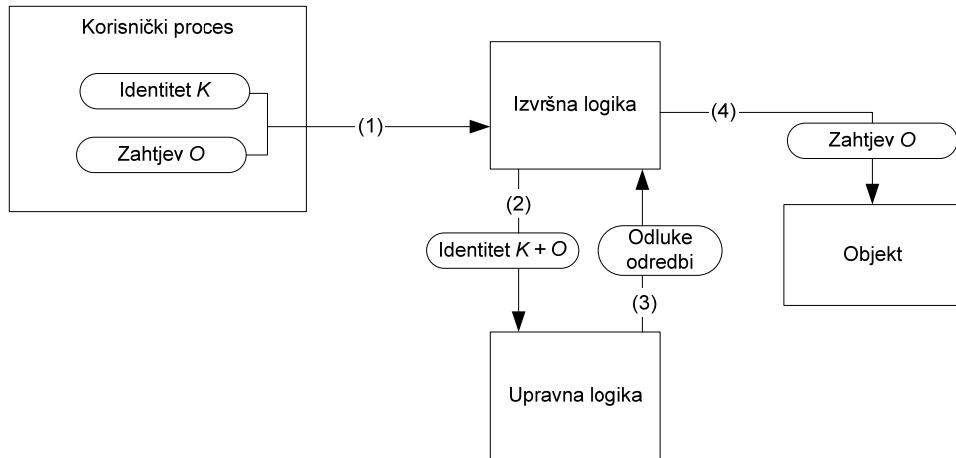
Slika 3-22: IETF arhitektura sustava nadzora pristupa

Upravna logika (engl. *policy decision point, PDP; access decision function, ADF*) odredbi još se naziva potrošačem odredbi [39, 49]. Uloga upravne logike je dohvaćanje odredbi iz spremnika odredbi, tumačenje i dostavljanje odredbi izvršnoj logici. Upravna logika u određenim slučajevima mora prevesti skup pravila koje dohvati iz spremnika te ih prilagoditi obliku koji razumije izvršna logika. U slučaju kada ima više primjeraka izvršne logike, upravna logika izvodi odabir primjeraka izvršne logike kojima treba poslati odluke donesene na osnovi odredbi. Prilikom osvježavanja vrijednosti odredbi ili na osnovi vanjskog utjecaja, upravna logika asinkronim načinom komunikacije dojavljuje odluke novih odredbi izvršnoj logici.

Izvršna logika (engl. *policy enforcement point, PEP; access enforcement function, AEF*) [39, 49] primjenjuje akcije prema pravilima odluke u odredbi primljenoj od upravne logike i prema uvjetima u radnoj okolini sustava (trenutni mrežni promet, doba dana, itd). Izvršna logika izvodi akcije odobravanja ili odbijanja zahtjeva te stavljanje zahtjeva u red posebnih akcija. Posebnim akcijama izvršna logika bilježi primljene zahtjeve, potražuje dodatne autentifikacijske vjerodajnice i slično.

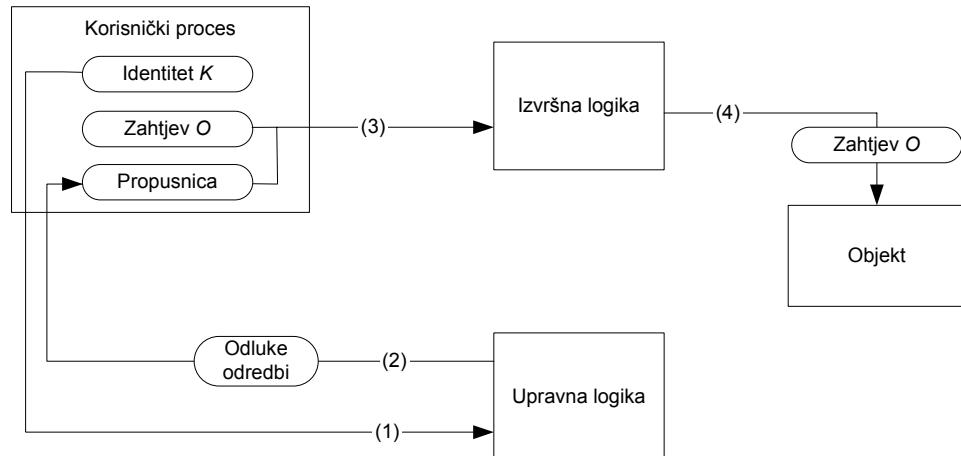
Upravnu i izvršnu logiku moguće je sjediniti u jedinstveno ostvarenu cjelinu sustava ili ih ostvariti razdvojene i raspodijeljene na različitim računalima. U mnogim je sustavima upravna logika razdvojena od objekata, usluga i primjenskih sustava, dok je izvršna logika tjesno vezana uz njih. Suradnja između razdvojene upravne i izvršne logike izvodi se posebnim protokolima. Protokoli suradnje između upravne i izvršne logike opisuju se primjenom dva klasična modela suradnje: *modelom potraživanja* (engl. *pull model*) i *modelom ponude* (engl. *push model*) [47, 48]. Potraživanje odnosno ponuda ističe način kojim izvršna logika dolazi do odluka odredbi o nadzoru pristupa objektu sustava.

Model potraživanja prikazan je na slici 3-23. Korisnički proces stvara zahtjev za korištenje objekta O , povezuje zahtjev objekta s identitetom korisnika K i šalje ih do izvršne logike (1). Na osnovi identiteta korisnika K i identiteta objekta O , izvršna logika zahtijeva od upravne logike odluku odredbi o nadzoru pristupa (2,3). Odluka definira pravo pristupa za postavljeni zahtjev na osnovi koje izvršna logika odobrava ili zabranjuje pristup objektu (4).



Slika 3-23: Model potraživanja odluka donesenih odredbama

Model ponude prikazan je na slici 3-24. Prije nego korisnički proces postavi zahtjev za pristup objektu sustava, korisnik se predstavlja upravnoj logici i zatražuje propusnicu (1). Upravna logika stvara propusnicu i upisuje u nju odluke odredbi o pravima pristupa korisnika koji se predstavio (2). Korisnički proces prilaže dobivenu propusnicu prilikom svakog zahtjeva objekta O postavljenog izvršnoj logici (3). Izvršna logika provjerava odluke odredbi upisane u propusnici te odobrava ili odbacuje zahtjev u skladu s pravom pristupa definiranim u odluci (4). Identitet K ne šalje se u izvršnu logiku jer nije nužan za rad izvršne logike. U odlukama odredbi upisanima u propusnicu izvršna logika nalazi potrebne informacije o nadzoru pristupa potvrđene od strane upravne logike kojoj vjeruje.



Slika 3-24: Model ponude odluka donesenih odredbama

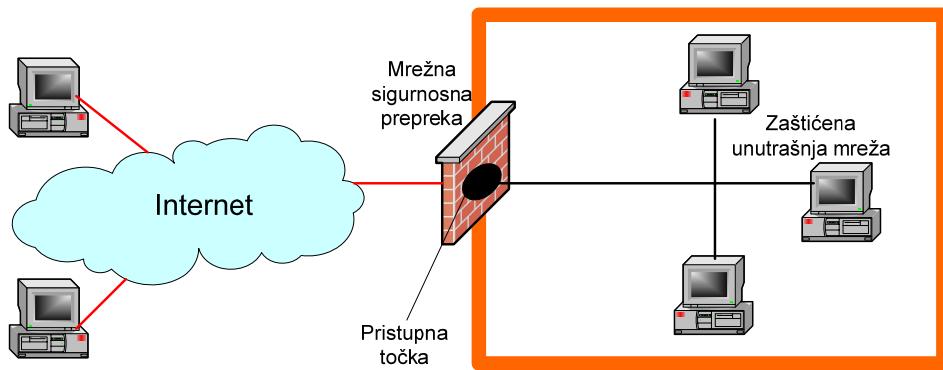
3.4.2. Uspostava zaštićene domene

Povjerljivi podaci poslovnih organizacija (npr. povjerljivost poslovanja, planovi razvoja proizvoda, promidžbeni planovi, finansijske analize) dostupni su u svakom trenutku zaposlenicima organizacije putem unutrašnje lokalne mreže organizacije. Dodatno,

zaposlenicima iz unutrašnje mreže potreban je pristup globalnoj mreži Internet. Povezivanje unutrašnje lokalne mreže na mrežu Internet zahtijeva nadzor pristupa unutrašnjoj mreži. Nije poželjno dopustiti konkurentske organizacijama ili zlonamjernim korisnicima otkrivanje povjerljivih podataka organizacije. Osim opasnosti od otkrivanja povjerljivih podataka, putem globalne mreže prijeti opasnost ulaska neželjenih podataka i programa u unutrašnju mrežu sustava poslovne organizacije. To se posebice odnosi na programe virusa, crva i ostalih programskih nametnika koji narušavaju sigurnost sustava, uništavaju vrijedne podatke, prave štetu koju administratori moraju popraviti i slično.

Nadzor nad unošenjem i iznošenjem podataka, programa i dokumenata pojednostavljuje se izgradnjom zaštićene domene sustava, te ako se sav ulazni ili izlazni promet sustava ostvaruje isključivo putem prolaza namijenjenih za tu svrhu. Takvi se prolazi zovu *pristupne točke* (engl. *access point*). Pristupna točka zasniva se na načelu zaštite prema kojem su se u srednjem vijeku osiguravali dvorci i utvrde. Na primjer, dvorce se štitilo mehanizmom pokretnog mosta kojim se prelazilo preko jarka iskopanog oko dvorca. Time se postiglo da sve osobe koje ulaze i izlaze iz dvorca prelaze preko jednog mosta, gdje ih se preispitivalo i provjeravalo zašto dolaze ili odlaze, te provodilo ulazno-izlazne odredbe s dvora.

Uspostavom pristupnih točaka, čija je uloga slična pokretnom mostu srednjovjekovnog dvorca, postiže se zaštita raspodijeljenih računalnih sustava poslovnih organizacija. Računala poslovne organizacije povezuju se u unutrašnju lokalnu mrežu i sav promet iz poslovne organizacije ili u poslovnu organizaciju prolazi putem pristupne točke. Pristupna točka prema potrebi i odredbama sustava propušta ili zaustavlja mrežni promet između unutrašnje lokalne mreže i vanjske globalne mreže. Pristup izvana do bilo kojeg dijela raspodijeljenoga sustava štiti nadzorni sustav. Nadzorni sustav provjerava je li pristup u skladu s odredbama pristupa sustava. Primjer pristupne točke u sustav prikazan je na slici 3-25. U sustavu je moguće definirati jednu ili više pristupnih točaka.

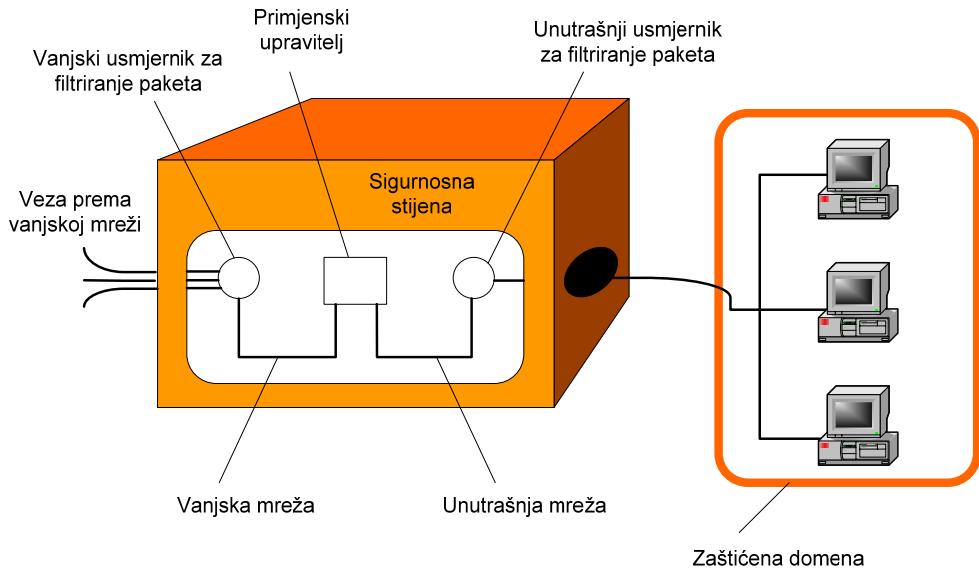


Slika 3-25: Primjer pristupne točke i zaštite unutrašnje mreže sustava

Sigurnosna prepreka

Sigurnosna prepreka prikladan je način za uspostavljanje zaštićene domene i ostvarivanje nadzora pristupa sredstvima sustava i unutrašnje mreže sustava. Sigurnosna prepreka ima različite uloge u svrhu zaštite sustava. Sigurnosne prepreke najčešće se primjenjuju za praćenje dolaznog prometa, provjeru identiteta vanjskog korisnika koji želi pristupiti sredstvima sustava te u autentikaciji korisnika na osnovi sigurnosnih znački (engl. *token*), adresa i sličnog. Dodatno se sigurnosne prepreke primjenjuju pri provjeri sigurnosnih i poslovnih odredbi za propuštanje zahtjeva te pri provjeri je li vanjski korisnik ovlašten pristupiti do određenih sredstava u sustavu. Često ih se koristi i za kriptiranje poruka radi zaštite povjerljivosti poslovnih informacija koje se šalju preko nesigurne javne mreže Interneta.

Postoje dvije osnovne izvedbe sigurnosne prepreke: usmjernik za filtriranje paketa (engl. *packet-filtering gateway*) i upravitelj primjenske razine (engl. *application-level gateway*). Te su izvedbe često spojene u jedinstvenu cjelinu kako je prikazano na slici 3-26. Usmjernik za filtriranje paketa je podvrsta sigurnosne prepreke koja u mrežnom komunikacijskom sloju provjerava zaglavljiva mrežnih paketa i na osnovi adresa izvořišta i odredišta odlučuje koji od mrežnih paketa treba propustiti, a koji odbaciti. Usmjernik za filtriranje paketa na vanjskoj mreži štiti sustav od vanjskih paketa, dok unutrašnji usmjernik za filtriranje paketa filtrira odlazne pakete. Na primjer, unutrašnji Web poslužitelj sustava moguće je zaštititi od zahtjeva korisnika koji ne pripadaju unutarnjoj mreži pomoću usmjernika za filtriranje paketa koji će odbaciti sve dolazne pakete koji stižu iz vanjske mreže prema Web poslužitelju.



Slika 3-26: Složena izvedba sigurnosne prepreke

Upravitelj primjenske razine je vrsta sigurnosne prepreke koja provjerava sadržaj dolaznih ili odlaznih poruka. Tipičan primjer je upravitelj elektroničke pošte (engl. *mail gateway*) koji odbacuje dolaznu i odlaznu poštu koja prekoračuje određenu veličinu. Neki napredniji upravitelji za poštu su sposobni razlikovati i filtrirati *nametljivu poštu* (engl. *spam*). Primjer upravitelja primjenske razine je i upravitelj koji dopušta pristup do poslužitelja digitalne knjižnice, ali samo do sažetaka dokumenata. Ako korisnik želi pogledati cijeli dokument, onda započinje protokol elektroničkog plaćanja.

Posebna vrsta upravitelja primjenske razine je zastupnički upravitelj (engl. *proxy gateway*). Ta vrsta upravitelja radi kao prednja strana (engl. *front end*) specifičnog primjenskog sustava i osigurava da se do primjenskog sustava propuste samo one poruke koje zadovoljavaju određene kriterije. Na primjer, mnoge Web stranice sadrže jednostavne skriptne programe (engl. *scripts, applets*) koji se izvršavaju na korisničkoj strani u pregledniku Web stranica te potencijalno sadržavaju štetan kod. U tom slučaju želi se spriječiti da takav kôd prodre do korisnika u unutrašnjoj mreži. Zastupnički upravitelj uređuje i zabranjuje prolazak određenih zahtjeva i stranica, ili pri prolasku mijenja stranice koje sadrže izvodivi kôd.

4. XML razina sigurnosti usluga

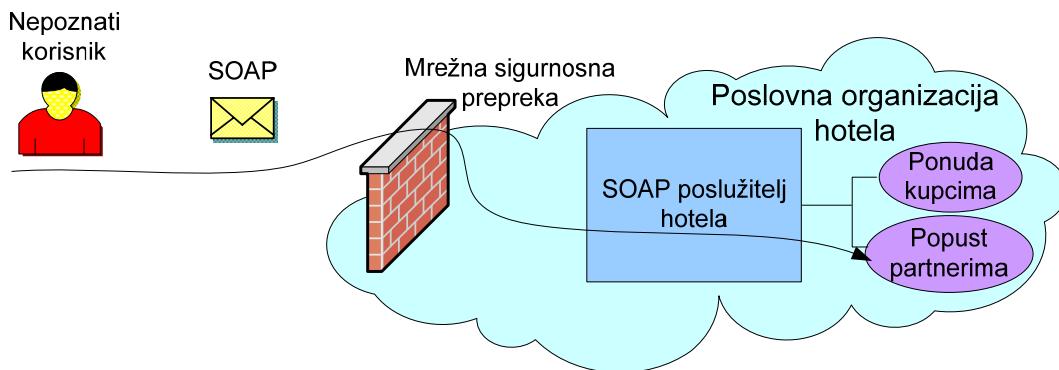
Komunikacija između Web Services usluga ostvaruje se razmjenom XML poruka oblikovanih prema SOAP protokolu. SOAP protokolom se propisuje oblik XML poruka koje interpretira i obrađuje SOAP poslužitelj Web Services usluga. SOAP poruke prenose se između Web Services usluga nekima od standardnih prijenosnih protokola, najčešće HTTP protokolom. Sigurnost SOAP poruka ne oslanja se na mehanizme sigurnosti iz sloja prijenosnih protokola. SOAP protokol zahtijeva posebne mehanizme zaštite podataka. Tim mehanizmima se postiže nezavisnost SOAP poruka od prijenosnog protokola, pa se SOAP poruke mogu na siguran način razmjenjivati bilo kojim prijenosnim protokolom. Sigurnost koja se ostvaruje za komunikaciju s Web Services uslugama postiže se na XML razini zaštitom pojedinačnih poruka (engl. *message-level security*).

U ovom poglavlju se primjerom pokazuje potreba uspostave mehanizama XML sigurnosti, primjenom kojih je moguće zaštititi komunikaciju porukama. XML tehnologije sigurnosti koje podržavaju zaštitu poruka prikazane su u drugom dijelu poglavlja.

4.1. Primjer potrebe XML sigurnosti

Na slici 4-1 prikazan je primjer SOAP poslužitelja koji ne štiti Web Services usluge računalnog sustava mehanizmima XML sigurnosti. Hotel primjenom navedenog računalnog sustava električkim putem nudi uslugu rezervacije smještaja kao Web Services uslugu i koristi mrežnu sigurnosnu prepreku za zaštitu svog poslovnog sustava. Hotel nudi posebnu ponudu popusta za svoje poslovne partnere i razlikuje tu ponudu od uobičajene ponude ostalim korisnicima. Ponude hotela svrstane su u dvije različite operacije za rezervaciju smještaja: operaciju *ponude partnerima* i operaciju *ponude kupcima*. Prema zahtjevima hotela samo se poslovnim partnerima hotela želi operacijom *ponude partnerima* omogućiti pristup posebnoj ponudi rezervacije smještaja. Nepoznatim korisnicima nudi se pristup operaciji *ponude kupcima*, ali za njih nije namijenjena operacija *ponude partnerima*.

SOP poslužitelj hotela raspolaže informacijama o uslugama koje poslužuje (ime usluge i ime operacija svake usluge), te ima mogućnosti obrade i posluživanja dolaznih SOAP zahtjeva. Međutim, SOAP poslužitelj ne ostvaruje provjeru šalje li dolazni SOAP zahtjev poslovni partner ili nepoznati korisnik. Dodatno, SOAP poslužitelj ne razlikuje sigurnosne odredbe mrežnih usluga i ne provodi autorizaciju i nadzor pristupa na razini SOAP poruka.



Slika 4-1: Primjer sigurnosnog propusta

Mrežna sigurnosna prepreka ostvaruje sigurnost u mrežnom sloju prijenosnih protokola. Njezina je uloga provjeriti zaglavje mrežnih paketa i propuštati mrežne pakete na osnovi adrese izvorišta i odredišta paketa. Mrežna sigurnosna prepreka nije sposobna razlikovati SOAP poruke čija obrada slijedi poslije prijenosa mrežnih paketa. Stoga se mrežnom sigurnosnom preprekom do SOAP poslužitelja prosljeđuju SOAP poruke korisnika koji ima pravo pristupa barem jednoj operaciji usluge poslužitelja. Bez ugrađene provjere sigurnosti poruka pristiglih SOAP poslužitelju, SOAP zahtjev propušten poslužitelju poziva operaciju bilo koje Web Services usluge koja se izvodi na poslužitelju. U tom slučaju je rizično na istom poslužitelju pružati usluge različitog stupnja osjetljivosti, koje imaju različite sigurnosne odredbe.

U primjeru na slici 4-1 nepoznati korisnik može uputiti SOAP zahtjev do SOAP poslužitelja putem mrežne sigurnosne prepreke. Mrežna sigurnosna prepreka prosljeđuje zahtjev nepoznatog korisnika do SOAP poslužitelja, jer nepoznati korisnik ima pristup operaciji ponude kupcima. Mrežna sigurnosna prepreka ne prepoznaje operaciju koju nepoznati korisnik poziva. SOAP poslužitelj bez ugrađenih postupaka provjere dolaznih poruka ne razlikuje je li zahtjev došao od nepoznatog korisnika ili poslovnog partnera. SOAP poslužitelj stoga izvodi zahtjev, te je nepoznati korisnik iskoristio nezaštićeni poslužitelj i pozvao operaciju namijenjenu isključivo za poslovne partnere.

4.2. XML tehnikije sigurnosti

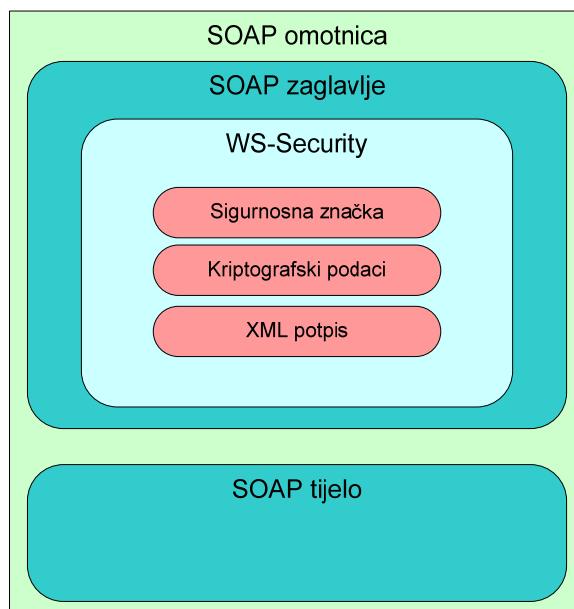
Standardi sigurnosnih protokola XML razine omogućili su uspostavu sigurnosti komuniciranja SOAP porukama. Različitim specifikacijama XML sigurnosti definira se način zaštite sadržaja i strukture SOAP poruka te provedba mehanizama za provjeru sigurnosti SOAP poruka (autentikacija, kriptiranje, provjera prava pristupa i sl.). Način kojim se uspostavlja sigurnost na razini SOAP poruka oslanja se na primjenu SOAP sigurnosne prepreke. Korisnici prema pravilima specifikacija XML sigurnosti prilažu

sigurnosne informacije unutar SOAP poruka. Sigurnosne informacije javljaju se u obliku potpisa, značke, kriptiranih podataka i sličnog. SOAP sigurnosna prepreka na osnovi tih informacija provodi mehanizame provjere sigurnosti te otkriva i zaustavlja uljeze prije nego dobiju mogućnost pristupa mrežnoj usluzi.

Organizacije W3C i OASIS razvile su nekoliko sigurnosnih protokola za provedbu sigurnosti na XML razini. WS-Security standard razvijen je kao okvir sigurnosti Web Services usluga, koji obuhvaća postupke digitalnog XML potpisivanja i XML kriptiranja. SAML autentikacije i XACML odredbe dodatno su razvijeni protokoli za podržavanje jedinstvene sjednice korisnika i nadziranje pristupa uslugama.

4.2.1. WS-Security

Specifikaciju WS-Security [50] definirala je organizacija OASIS i njome je opisan standardiziran način sigurne razmjene SOAP poruka između mrežnih usluga. WS-Security definira mehanizme za nepovredivost, povjerljivost i autentičnost SOAP poruka. U slučaju dospijeća SOAP poruke na odredište, utvrđuje se nepovredivost sadržaja koji je posao izvorni pošiljatelj i utvrđuje se autentičnost pošiljatelja poruke. U situacijama razmjene povjerljivih informacija, dodatno je potrebno osigurati povjerljivost informacija kriptiranjem poruka.



Slika 4-2: Struktura sigurnosnih informacija u SOAP poruci prema WS-Security pravilima

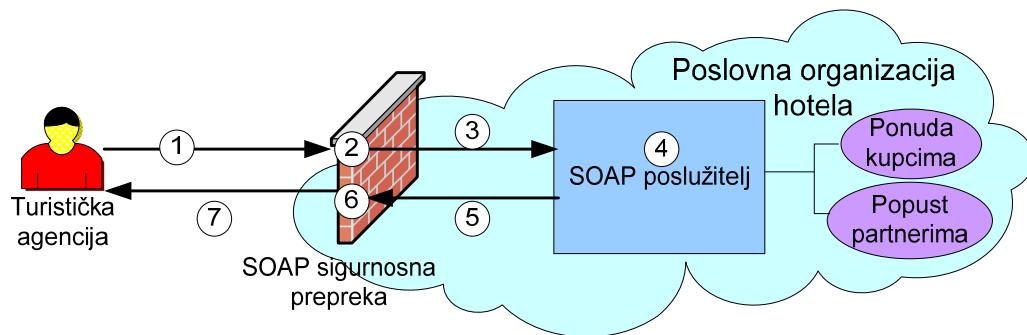
WS-Security definira sintaksna pravila umetanja sigurnosnih podataka u SOAP poruku. Umetnutim sigurnosnim podacima štiti se sigurnost podataka u SOAP poruci. SOAP poruka sa sigurnosnim podacima oblikovanim prema WS-Security standardu prikazana je na

slici 4-2. Sigurnosni podaci umeću se u zaglavlje SOAP poruke. Sigurnosne informacije u SOAP porukama uključuju sigurnosnu značku, XML potpis i kriptografske podatke. Kriptografski podaci i XML potpis ne moraju nužno biti poredani kako je prikazano na slici. Mogu biti poredani i obrnutim redoslijedom, ovisno o tome provodi li se prvo postupak XML kriptiranja a tek onda postupak potpisivanja poruke. Postupak XML kriptiranja osigurava povjerljivost XML podataka u SOAP poruci i opisan je u odjeljku 4.2.3. Potpisivanje poruke XML potpisom osigurava nepovredivost i autentičnost podataka SOAP poruke i opisano je u odjeljku 4.2.2.

Sigurnosna značka (engl. *security token*) je oznaka koja se umeće u SOAP poruku, a služi kao sigurnosna propusnica odnosno iskaznica identiteta koju treba pokazati ako se želi pristupiti u domenu zaštićenog sustava. Postoji nekoliko vrsta sigurnosnih znački. Najčešće se primjenjuje sigurnosna značka *ime-zaporka*, poput na primjer korisničkog imena i zaporke koji se navode pri provjeravanju elektroničke pošte. Ime-zaporka je čitljiva sigurnosna značka. Međutim, postoje i sigurnosne značke koje su zapisane u binarnom obliku i stoga nisu čitljive čovjeku. Takve se sigurnosne značke zovu binarne sigurnosne značke. Na primjer, X.509 vjerodajnica je vrsta binarne sigurnosne značke. Vrsta sigurnosne značke koja omogućuje *jedinstvenu prijavu* sustavu (engl. *single-sign-on*, SSO) jest SAML značka. SAML značka je opisana u odjeljku 4.2.4.

U primjeru komunikacije turističke agencije s uslugom hotela (slika 4-3) objašnjava se uporaba pojedinih sigurnosnih informacija koje se WS-Security standardom uključuju u SOAP poruke. SOAP poslužitelj i SOAP sigurnosna prepreka prikazani su kao dvije zasebne cjeline, međutim, u stvarnosti mogu biti ostvarene kao jedna cjelina. U primjeru se turističku agenciju smatra poslovnim partnerom hotela. Da bi se pomoću XML sigurnosne prepreke zaštitilo ugrožavanje sigurne komunikacije s hotelom, poruka koju šalje agencija uključuje sigurnosne podatke koji jamče vjerodostojnost poruke i autentičnost pošiljatelja poruke. To se svojstvo ostvaruje stvaranjem potpisa sadržaja poruke i uključivanjem tog potpisa u SOAP poruku (1). SOAP sigurnosna prepreka na osnovi sigurnosne značke i potpisa priloženih u zaglavlju primljene SOAP poruke provjerava je li strana koja je zatražila uslugu zaista poslovni partner hotela i je li poruka izmijenjena pri prijenosu (2). SOAP sigurnosna prepreka propušta do SOAP poslužitelja samo one zahtjeve koji zadovoljavaju oba uvjeta. U ovom slučaju turistička agencija je poslovni partner hotela te se zahtjev prosljeđuje SOAP poslužitelju (3). SOAP poslužitelj će izvesti operaciju koju pošiljatelj zahtijeva i oblikovati odgovor (4). U ovom slučaju turistička agencija je pozvala operaciju rezerviranja hotela uz poseban popust. Uz pretpostavku da je informacija o popustu koji ostvaruju poslovni partneri hotela povjerljiva informacija nedostupna svim korisnicima, odgovor mora biti kriptiran, te

SOAP sigurnosna prepreka kriptira tajni dio poruke i uključuje kriptografske podatke u zaglavlje SOAP poruke (6). SOAP odgovor se šalje pozivatelju odnosno turističkoj agenciji (7).



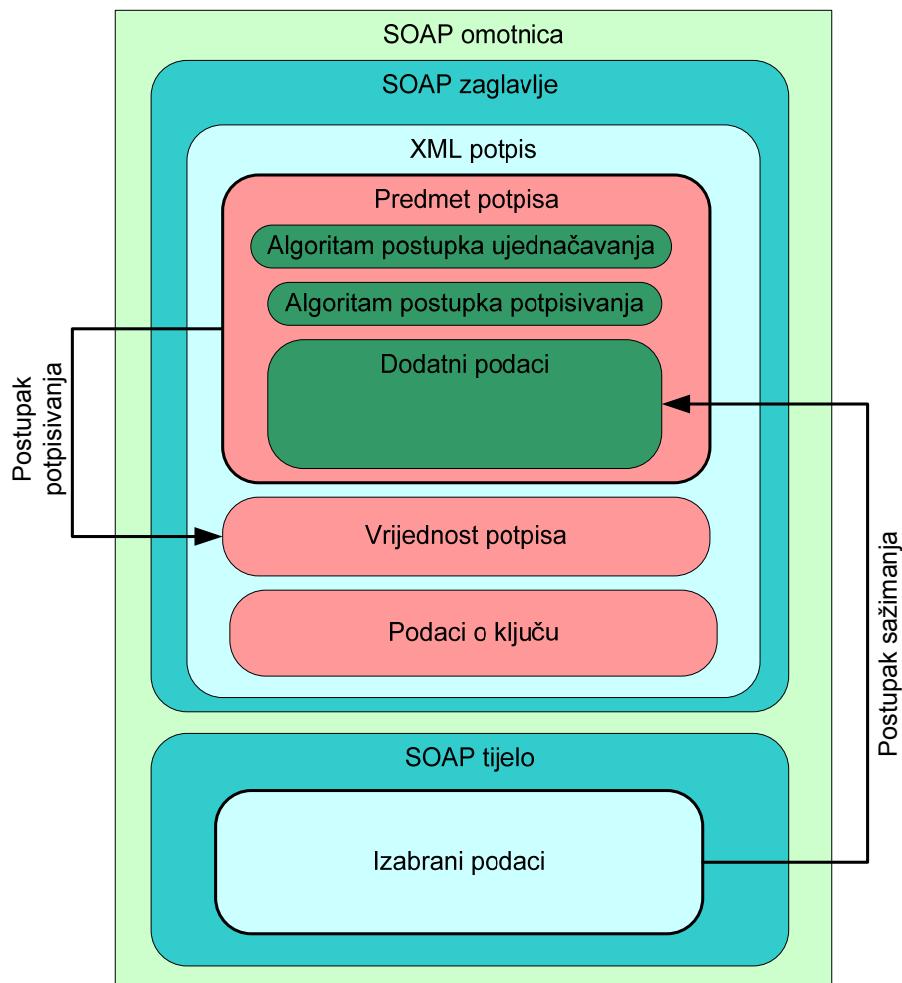
Slika 4-3: Scenarij upotrebe WS-Security standarda

4.2.2. Digitalno XML potpisivanje

Digitalni XML potpis služi kao čvrst dokaz nepovredivosti i autentičnosti XML podataka te kao dokaz neporicljivosti o tome tko je stvorio dokument. Digitalno XML potpisivanje koristi standard XML Signature [51] koji opisuje XML sintaksu za predstavljanje i povezivanje kriptografskih potpisa i podataka u XML dokumentima. XML potpis razlikuje se od drugih protokola za potpisivanje poruka, kao što je recimo PGP, po tome što osim potpisivanja cijelog dokumenta podupire i potpisivanje pojedinačnih dijelova XML dokumenta. XML potpis definira i mehanizme za provjeru potpisa (engl. *countersigning*) te postupke ujednačavanja (engl. *canonicalization*).

Postupci ujednačavanja su potrebni za XML potpisivanje zato što postupci sažimanja (engl. *digest*) rade nad XML podacima kao nad nizom okteta. Problem se javlja zbog mogućnosti da dva različita niza okteta predstavljaju isti XML dokument. Na primjer, ako se promijeni redoslijed svojstava koja predstavljaju jednu cjelinu XML dokumenta, dobiveni XML dokument bit će logički istovjetan izvornoj verziji XML dokumenta. Međutim, takva dva logički istovjetna XML dokumenta neće imati isti niz okteta, te se sažimanjem neće proizvesti isti sažetak. Postupci ujednačavanja primjenjuju se u svrhu stvaranja identičnog niza okteta za logički istovjetne XML dokumente.

Standard digitalnog XML potpisivanja moguće je primijeniti na proizvoljan XML dokument, a na slici 4-4 prikazan je primjer strukture koja se primjenjuje za digitalno XML potpisivanje SOAP poruke. XML potpis se umeće u zaglavlje SOAP poruke prema WS-Security standardu. Osobitost je kako se u tom slučaju ostvaruje potpisivanje *izabranih podataka* iz SOAP tijela.



Slika 4-4: Struktura XML potpisa u SOAP poruci

U stvaranju XML potpisa izabranih podataka iz SOAP tijela primjenjuju se dva kriptografska postupka: postupak sažimanja i postupak potpisivanja. XML potpis sastoji se od tri glavna dijela: *predmet potpisa*, *vrijednosti potpisa* i *podataka o ključu*. Izabrani podaci iz SOAP tijela najprije se sažimaju postupkom sažimanja. Nakon sažimanja, sažetak izabranih podataka se ugnježduje u *predmet potpisa*. Zatim se primjenom postupka potpisivanja potpisuje cjelina *predmet potpisa*. *Predmet potpisa*, osim sažetka izabranih podataka, sadrži podatke koji opisuju svojstva postupka potpisivanja primijenjenog na *predmet potpisa*. Potpisivanje *predmeta potpisa* ujedno znači potpisivanje sažete vrijednosti, odnosno izabranih podataka SOAP tijela iz kojih je dobivena sažeta vrijednost. Potpisani podaci iz *predmeta potpisa* upisuju se u cjelinu *vrijednost potpisa*. *Podaci o ključu* su proizvoljni dio XML potpisa koji se po potrebi uključuje u potpis.

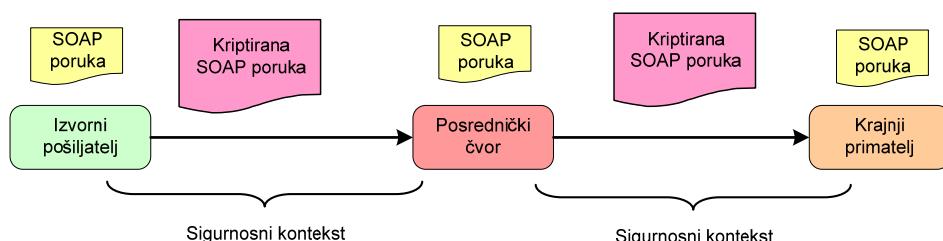
Predmet potpisa sadrži podatak o postupku ujednačavanja, vrsti korištenog algoritma za potpisivanje te dodatne podatke koji opisuju XML potpis. *Algoritam postupka ujednačavanja* jednoznačno definira postupak pomoću kojeg se podaci ujednačavaju prije

nego ih se digitalno potpisuje. Ujednačavanje se primjenjuje na cjelinu *predmet potpisa*. Potpisani podaci zapisuju se u cjelinu *vrijednost potpisa*. Algoritam postupka potpisivanja je dio koji sadrži točne podatke o vrsti postupka kojim se stvara kriptografski potpis cjeline *predmet potpisa*. U *predmetu potpisa* uvijek se nalazi bar jedan dio s dodatnim podacima. U *dodatnim podacima* nalazi se sažetak izabranih podataka iz tijela SOAP poruke, ime algoritma korištenog za sažimanje te opis postupaka pretvorbe podataka provedenih prije sažimanja. Primjer postupaka pretvorbe podataka je već spomenuto ujednačavanje podataka ili pretvaranje iz tabličnog u jednostavni netablični tekst. Ako je primijenjeno više postupaka pretvorbe podataka, njihov je redoslijed bitan.

Cjelina *podaci o ključu* namijenjena je za podatke pomoću kojih se prepoznaže ključ kojim se provjerava potpis na strani primatelja SOAP poruke. XML potpisom nije propisan način na koji se povezuju vrijednosti ključa s identifikatorom ključa navedenim u cjelini *podaci o ključu*. *Podaci o ključu* su proizvoljni dio XML potpisa, jer primjenski sustav koji stvara XML potpis po potrebi uključuje *podatke o ključu* u XML potpis. Dodatno, cjelina *podaci o ključu* koristi se u XML kriptiranju.

4.2.3. XML kriptiranje

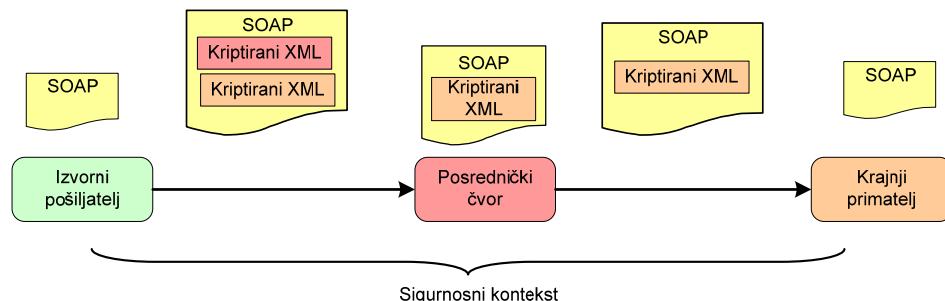
S ciljem uspostave standardnog načina zaštite povjerljivosti podataka u XML dokumentima, organizacija W3C je standardom XML Encryption [52] definirala mehanizme XML kriptiranja. XML kriptiranje razlikuje se od standardnog kriptiranja dokumenata i poruka. Osim kriptiranja cijelog dokumenta, XML kriptiranje podupire i kriptiranje pojedinačnih dijelova XML dokumenta. To se svojstvo primjenjuje za očuvanje povjerljivost podataka tijekom prijenosa SOAP poruka preko posrednih komunikacijskih čvorova.



Slika 4-5: Uspostava sigurnosti od točke do točke

Na slici 4-5 prikazan je primjer SOAP poruke koja se na putu do odredišne Web Services usluge djelomično obraduje na posredničkom čvoru. U tom se slučaju primjenom sigurnog transportnog protokola osigurava samo prijenos poruke od jednog do drugog čvora. Za potrebe obrade poruke posrednički čvor mora dekriptirati cijelu poruku. Time posrednički

čvor dobiva i uvid u podatke koji su namijenjeni isključivo odredišnom čvoru. Na posredničkom čvoru postoji prekid sigurnosti kojom se štiti povjerljivost podataka. Primjena sigurnosti na razini prijenosnih protokola predstavlja zadovoljavajuće rješenje samo u slučajevima kada svi sudionici komunikacije vjeruju jedni drugima.



Slika 4-6: Uspostava sigurnosti na razini komunikacije primjenskih sustava

Ako nije moguće vjerovati posredničkom čvoru, odnosno postoji sumnja da će se na posredničkom čvoru zloupotrijebiti podatke namijenjene odredišnoj Web usluzi, onda je potrebno primijeniti XML kriptiranje. Razmatra se prethodno navedeni scenarij u kojem korisnik šalje poruku odredišnoj Web usluzi preko posredničkog čvora. U scenaru prikazanom na slici 4-6 korisnik uz pomoć XML kriptiranja posebno kriptira dio namijenjen posredničkom čvoru, a posebno kriptira dio namijenjen odredišnoj usluzi. Kad posrednički čvor primi poruku, on zna dekriptirati samo svoj dio poruke. Posrednički čvor nije u mogućnosti dekriptirati dio poruke namijenjen odredišnom čvoru. Posrednički čvor obrađuje svoj dio poruke i poruku šalje odredišnoj Web Services usluzi. Odredišni čvor dekriptira dio poruke koji je namijenjen isključivo njemu.

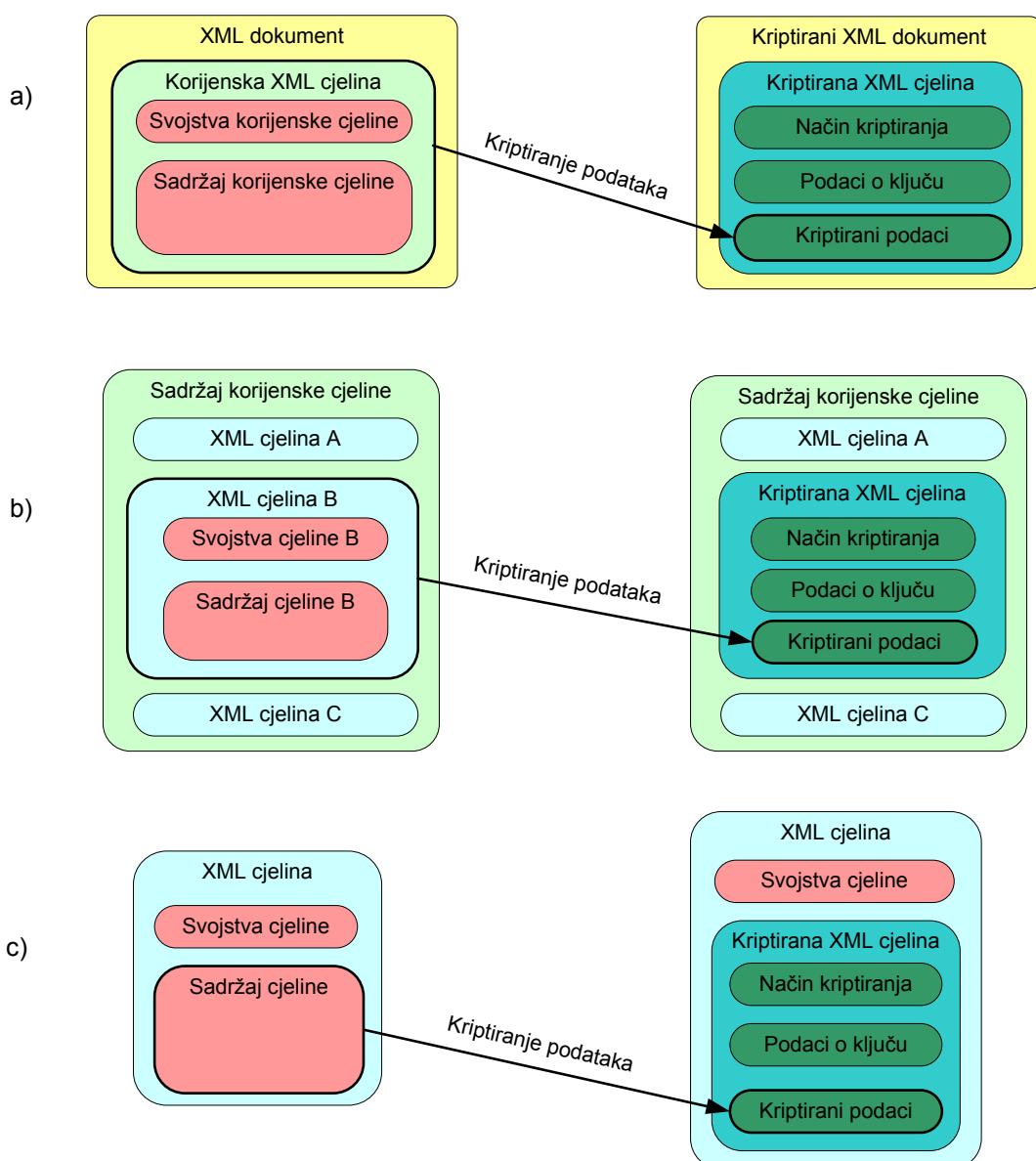
XML kriptiranje nudi nekoliko različitih mogućnosti kriptiranja. Moguće je kriptirati cijeli XML dokument, samo jednu samoopisujuću XML cjelinu, sadržaj podataka unutar samoopisujuće XML cjeline dokumenta, podatke koji nisu XML podaci (npr. JPEG slike) i kriptirani dio XML dokumenta (tzv. nadkriptiranje).



Slika 4-7: Osnovna struktura XML kriptiranja

Osnovni oblik koji se dobije kao rezultat XML kriptiranja jest *kriptirana XML cjelina* prikazana na slici 4-7. *Način kriptiranja* određuje algoritam kriptiranja koji se

primjenjuje za kriptiranje. Dodatno, način kriptiranja određuje kriptira li se cijeli dokument, jedna XML cjelina ili samo sadržaj podataka unutar XML cjeline. U slučaju kriptiranja cijelog XML dokumenta korijensku XML cjelinu XML kriptiranjem zamjenjuje *kriptirana XML cjelina* (slika 4-8a). Ako se radi o kriptiranju jedne XML cjeline XML dokumenta, rezultat kriptiranja je *kriptirana XML cjelina* na mjestu XML cjeline koju se kriptiralo u izvornom XML dokumentu (slika 4-8b). Kriptira li se sadržaj podataka unutar XML cjeline, dobije se *kriptirana XML cjelina* koja zamjenjuje sadržaj podataka XML cjeline iz izvornog XML dokumenta (slika 4-8c).

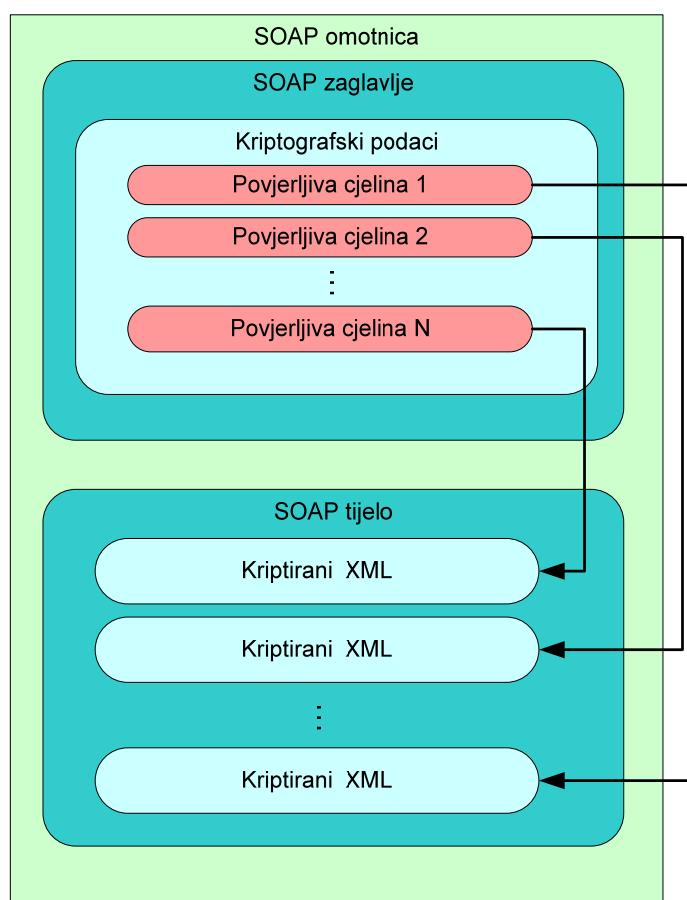


Slika 4-8: Mogućnosti kriptiranja različitih dijelova XML dokumenta

Podaci o ključu predstavljaju istu cjelinu koja pri XML potpisu služi za rapoznavanje ključa kojim se digitalno potpisalo podatke. U ovom slučaju ta se cjelinu

primjenjuje za raspoznavanje ključa kojim je moguće pročitati kriptirane podatke. Ključ koji se koristi za kriptiranje ne navodi se u toj cjelini. U toj cjelini navodi se samo podatak koji omogućuje primatelju poruke određivanje ključa kojim je moguće dekriptirati kriptirane podatke. Često identifikator upućuje na ključ koji je sadržan u sigurnosnoj znački. *Zakriptirani podaci* predstavljaju kriptirani oblik podataka čija se povjerljivost štiti kriptiranjem.

Kako je SOAP poruka također XML dokument, za kriptiranje podataka u SOAP poruci vrijede ista pravila koja vrijede za kriptiranje XML podataka. Razlika je što se dodatno u zaglavlju SOAP poruke navodi lista svih povjerljivih podataka u poruci koji su kriptirani na strani pošiljatelja poruke. Na slici 4-9 prikazan je način umetanja liste povjerljivih podataka u zaglavlje SOAP poruke prema pravilima WS-Security standarda.



Slika 4-9: XML kriptiranje podataka u SOAP poruci

4.2.4. SAML autentikacija

Specifikacija SAML (*Security Assertion Markup Language*) [53] definira standardan način zapisivanja i razmjene sigurnosnih obilježja (engl. *attributes*) pripisanih identitetu korisnika. Organizacija OASIS definirala je SAML standard pomoću SAML izjava (engl.

SAML assertion) i SAML protokola. Sigurnosna obilježja i identitet korisnika zapisuju se *SAML izjavama*, a SAML protokol definira način razmjene SAML izjava između povjerljivih organizacija. Razmjena SAML izjava prepostavlja povjerenje između organizacija, ali se ne pruža način uspostave povjerenja. SAML standard pruža neutralan i standardan način razmjene sigurnosnih informacija u odnosu na vlasnička (engl. *proprietary*) sigurnosna rješenja. Primjena SAML standarda omogućuje uspostavu jedinstvenog identiteta (engl. *federated identity*) u raspodijeljenim sustavima.

Razmjena sigurnosnih svojstava i ostvarivanje jedinstvenog identiteta nužni su za povezivanje primjenskih sustava poslovne organizacije (engl. *Enterprise Application Integration*, EAI) i povezivanje poslovnih organizacija. Ostvarivanje jedinstvenog identiteta postiže se razmjenom autentikacijskih obilježja korisnika. Razmjenjivanjem autentikacijskih obilježja primjenom SAML izjava omogućuje se standardan način povezivanja sigurnosnih sustava koji ne zahtjeva održavanje i sinkronizaciju autentikacijskih podataka između imenika.

Glavna je uloga SAML izjave, koja sadrži autentikacijska svojstva korisnika, pružanje potpore za ostvarivanje jedinstvene sjednice. Ostvarivanjem jedinstvene sjednice, krajnjim korisnicima nudi se poboljšanje kvalitete uporabljivosti (engl. *quality of experience*) raspodijeljenog sustava. Jedinstvena sjednica omogućuje korisnicima autenticiranje na jednom mjestu u raspodijeljenom sustavu, a zatim im omogućuje pristup proizvoljnim uslugama i sredstvima bez dodatne autentikacije. SAML izjave, osim izricanja autentičnosti korisnikova identiteta, izriču i druga korisnička obilježja. Na sličan se način primjenom različitih korisničkih obilježja moguće je poboljšati krajnje iskustvo korisnika i kvalitetu upotrebljivosti sustava.

Prije usvajanja SAML standarda, uobičajeni nači ostvarenja jedinstvene sjednice bila je uporaba *električnih tragova* (engl. *cookie*) koji se zadržavaju u Internet preglednicima. Pojavom SAML standarda, istu informaciju koju se spremalo u električke tragove, moguće je na standardizirani način spremiti u XML strukturu SAML izjava. Osim toga, električkim tragovima je komunikacija bila ograničena na komunikaciju preglednika i primjenskog sustava. Primjenom SAML standarda podupire se i komunikacija primjenskog sustava s drugim primjenskim sustavom.

Primjer ostvarivanja jedinstvene sjednice

Ostvarivanje jedinstvene sjednice na osnovi dijeljenja autentikacijskih informacija između povjerljivih sudionika zorno se opisuje na sljedećem primjeru. Neka čuvar čuva pristup glavnom ulazu zgrade u kojoj su smješteni poslovni uredi. Posjetitelji su na glavnom

ulazu zgrade dužni pokazati svoje iskaznice i autenticirati se čuvaru. Čuvar pregledava ispravnost identifikacijskih kartica i provjerava popis posjetitelja te odobrava njihov ulazak u zgradu. Pretpostavlja se da posjetitelj želi obići nekoliko ureda u zgradi. Svaki ured ima vlastitog čuvara koji čuva ulaz u ured. Posjetitelj se stoga treba autenticirati na ulazu u svaki od ureda. Čuvar svakog ureda iznova izvodi zaseban autentikacijski postupak provjere identiteta svakog posjetitelja. Za pristup u različite uredske posjetitelj mora imati posebne identifikacijske značke.

Pojedinačni uredi u zgradi mogu dogovoriti i uspostaviti povjerenje u autentikaciju koju izvodi čuvar na glavnem ulazu u zgradu. Zgrada u tom slučaju postaje povjerljiva domena sastavljena od ureda u zgradi koji dijele zajedničku autentikaciju korisnika. Jedno rješenje takvog dijeljenja autentikacijskih informacija jest izdavanje privremene identifikacijske značke svakom posjetitelju nakon uspješne autentikacije na ulazu u zgradu. Identifikacijska značka s ulaza zgrade ima ograničeno vrijeme trajanja. Ako je znački isteklo vrijeme trajanja, značka je neispravna. Posjetitelj pokazuje identifikacijsku značku dobivenu na glavnem ulazu pri ulasku u bilo koji ured. Čuvar ureda provjerava ispravnost jedinstvene značke s glavnog ulaza prije nego pusti posjetitelja u ured. Posjetitelj ne mora za svaki ured imati posebne identifikacijske značke, već samo značku kojom se autenticira na ulazu u zgradu.

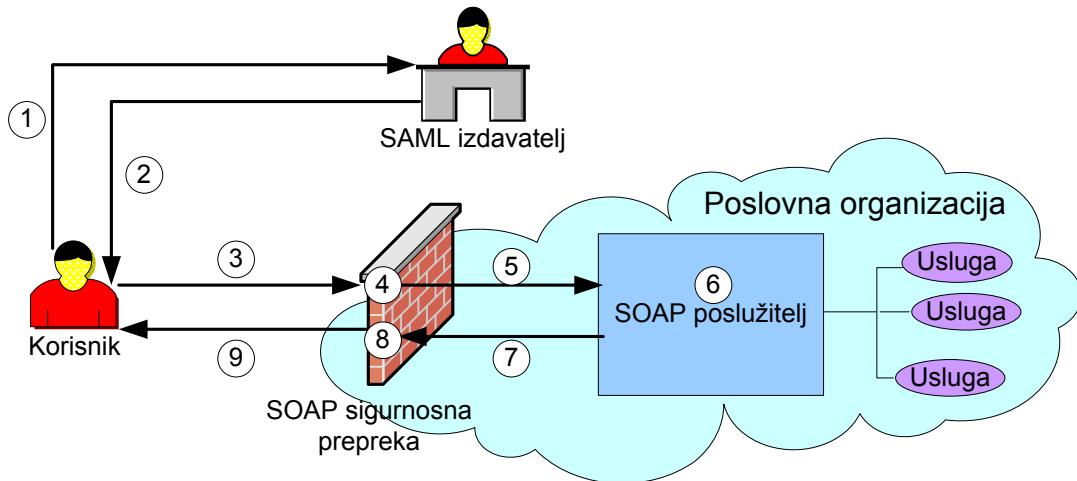
Primjer SAML izjava

Koncept izdavanja SAML izjava sličan je principu izdavanja privremenih znački. SAML izjave su tvrdnje koje vrijede određeno vrijeme. Imaju značenje privremenih znački s obilježjima koja nose određenu tvrdnju o autentikaciji korisnika ili o nekim njegovim važnim svojstvima. Primjer SAML izjave o autentikaciji korisnika izrečene razgovornim jezikom je:

"Subjekt po imenu *Marko* vlasnik je javnog ključa s oznakom *MarkovKljuč*. Izdavatelj koji je izdao ovu izjavu autenticirao je *Marka* XML potpisom. Izjava vrijedi od trenutka X do trenutka Y."

Slično se može SAML izjavom tvrditi neko drugo korisnikovo obilježje, na primjer:

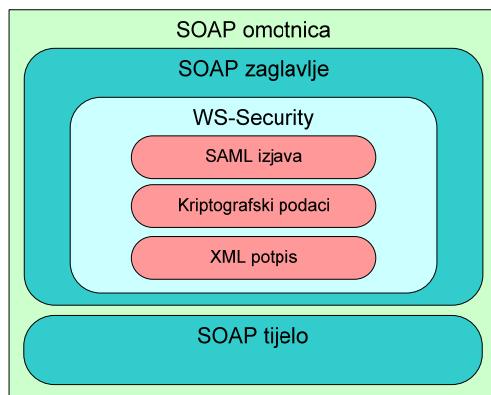
"Subjekt po imenu *Marko* vlasnik je javnog ključa s oznakom *MarkovKljuč*. Dodatno, *Marko* je student Fakulteta elektrotehnike i računarstva u Zagrebu (FER). Izjavu je izdao i potpisao FER. Izjava vrijedi od trenutka X do trenutka Y."



Slika 4-10: Koncept ostvarivanja jedinstvene sjednice pomoću *SAML izjava*

Na slici 4-10 prikazan je koncept prema kojem korisnik pomoću *SAML izjava* ostvaruje jedinstvenu sjednicu u sustavu poslovne organizacije. *SAML izdavatelj* je povjerljiva treća strana koja izdaje *SAML izjave*. Pristup do usluga na poslužitelju štiti SOAP sigurnosna prepreka. SOAP sigurnosna prepreka ima povjerenje u autentikacijske izjave koje izdaje *SAML izdavatelj*. *SAML izdavatelj* izdaje određenu izjavu na *SAML upit* korisnika koji zahtijeva izjavu (1). Tijekom slanja zahtjeva korisnik se autenticira *SAML izdavatelju*. *SAML izdavatelj* provjerava autentičnost korisnika i kao rezultat vraća *SAML odgovor* (2). U odgovoru se nalazi SAML izjava koju korisnik uključuje u SOAP zahtjev, kojeg zatim šalje prema SOAP poslužitelju (3). SOAP sigurnosna prepreka presreće i provjerava valjanost autentikacijske SAML izjave (4). Na osnovi valjane SAML izjave, zahtjev se prosljeđuje do SOAP poslužitelja (5) te se obrađuje na poslužitelju (6). Pripremljeni odgovor šalje se natrag do SOAP sigurnosne prepreke (7), koja prema potrebi izvodi sigurnosne operacije kojima štiti povjerljivost i vjerodostojnost podataka (8). Korisniku se vraća odgovor na zahtjev (9).

Primjenom SAML sigurnosnog protokola *SAML izdavatelj* postavlja *SAML upit* i dobiva SAML izjave u *SAML odgovoru*. SAML protokol nadovezuje se na komunikacijske



Slika 4-11: Mjesto umetanja SAML izjave u SOAP poruci

protokole. Najčešće i najznačajnije je nadovezivanje SAML protokola na SOAP protokol. Na slici 4-11 prikazan je način umetanja SAML izjava u SOAP poruke. Usporedbom slike 4-11 sa slikom 4-2 vidi se da SAML izjava po sintaksi WS-Security standarda odgovara sigurnosnoj znački, jer se umeće prema WS-Security standardu upravo na mjesto sigurnosne značke u SOAP zaglavljtu.

4.2.5. XACML nadzor pristupa

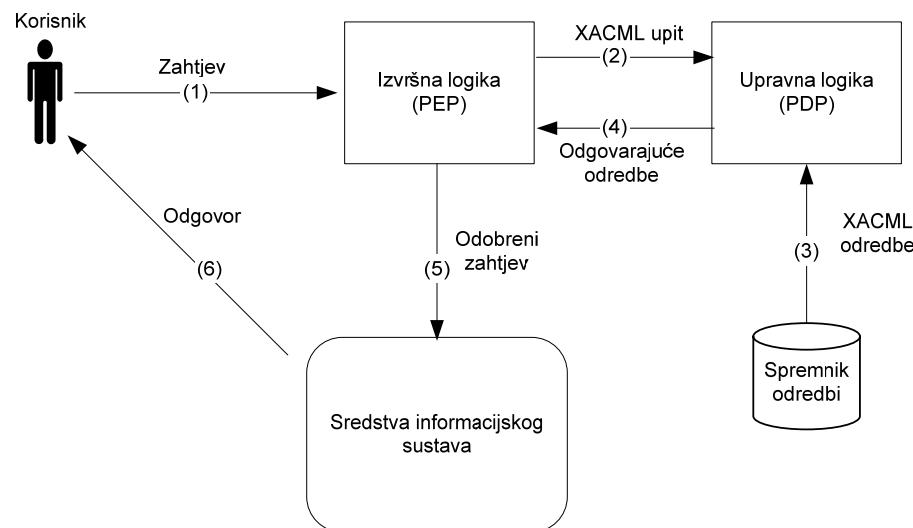
Specifikaciju XACML (*eXtensible Access Control Markup Language*) [54] za izražavanje odredbi o nadzoru pristupa u XML obliku definirala je organizacija OASIS. XACML jezikom i protokolom izražavaju se i razmjenjuju pravila i odredbe nadzora pristupa. Mehanizmi za nadzor pristupa primjenjuju odredbe nadzora pristupa i na osnovi njih donose odluku o pristupu do objekta.

XACML standard omogućuje zamjenu postojećih, posebnih vlasničkih (engl. *proprietary*) mehanizama za nadzor pristupa. Prije nego je usvojen XACML, svaki je sustav sigurnosti ostvarivao svoje vlastito rješenje za nadzor pristupa. XACML ima niz prednosti u odnosu na druge načine ostvarivanja nadzora pristupa. Administratori definiraju odredbe za nadzor pristupa jednom i ne moraju ih prepisivati i prevoditi u druge vlasničke odredbe pristupa. Graditelji sigurnosnih sustava ne moraju definirati vlastite jezike za opis odredbi i pisati programe koji će ih koristiti. Umjesto toga, koristi se postojeći standardizirani jezik. Dodatno, XACML odredbe mogu se povezivati i oslanjati na definirane odredbe te graditi nove odredbe korištenjem već izgrađenih.

XACML standard definira dvije funkcijeske cjeline: *PEP* (engl. *Policy Enforcement Point*) i *PDP* (engl. *Policy Decision Point*). Funkcijeske cjeline odgovaraju funkcionalnostima izvršne i upravne logike, što je opisano u odjeljku 3.4.1. PDP donosi odluku treba li zahtjev propustiti ili odbaciti, a PEP propušta ili odbacuje pojedini zahtjev. PEP i PDP komuniciraju XACML jezikom za postavljanje upita. Razmjena autorizacijskih odluka odredbi između PEP i PDP moguće je ostvariti SAML protokolom. Arhitektura sustava za nadzor pristupa zasnovanog na XACML standardu prikazana je na slici 4-12.

U tipičnom XACML scenariju, subjekt (npr. korisnik, radna stanica) želi izvesti određenu operaciju nad sredstvom sustava koji se štiti. Subjekt šalje svoj zahtjev do sustava koji štiti sredstvo (npr. datotečnog sustava ili Web poslužitelja) (1). Taj se sustav u kontekstu nadzora pristupa zove *PEP*. Koristeći se jezikom XACML, PEP stvara *XACML upit* sastavljen na osnovi svojstava subjekta, operacije, sredstva i drugih potrebnih informacija. XACML upit se šalje u *PDP* (2). PDP analizira upit, dohvata odredbe napisane u XACML

jeziku koje se odnose na zahtjev (3) te na osnovi njih donosi odluku treba li zahtjev propustiti ili odbaciti. Odgovor se zapisuje u XACML jeziku i vraća natrag do PEP (4), koji zatim provodi odluku o dopuštanju ili zabrani subjekta zahtjeva (5, 6).



Slika 4-12: Arhitektura nadzora pristupa zasnovana na XACML standardu

XACML definira istodobno jezik za odredbe o nadzoru pristupa i jezik za upite i odgovore. Jezik za odredbe omogućuje izražavanje odredbi o tome "tko smije što napraviti i kada". Jezik za upite i odgovore omogućuje postavljanje upita je li određeni pristup u skladu s definiranim odredbama te opisuje odgovore na takve upite. Primjenom XACML specifikacije definira se način zapisa pravila u odredbama te algoritam za primjenu jednog pravila u slučaju više primjenjivih pravila u odredbama.

U XACML zapisu odredbe o nadzoru pristupa su liste koje se sastoje od četiri elementa: subjekta, ciljnog objekta, akcije i uvjetnog događaja. *Subjekt* je korisnik ili grupa. *Ciljni objekt* predstavlja XML dokument, uređaj ili datoteku. Akcije su operacije čitanja, pisanja, stvaranja i brisanja. *Uvjetni događaj* (engl. *provision*) predstavlja radnju koja se mora izvršiti tijekom upotrebe XACML pravila. Primjer uvjetnih događaja su slanje upozorenja, traženje dodatnih vjerodajnica (engl. *credential*), započinjanje postupka prijave sustavu i slično.

5. Sigurnost računalnih spletova

U ovom poglavlju opisano je nekoliko sustava sigurnosti koji se primjenjuju u zaštiti računalnih spletova. Opisuje se sigurnost Globus sustava koji je jedan od najraširenijih sustava računalnih spletova. Posebice se ističu CAS i VOMS kao sustavi za autorizaciju te sustav PRIMA za nadzor pristupa u raspodijeljenom sustavu.

5.1. Sigurnost Globus sustava

Uloga sigurnosti u računalnim spletovima pokazuje se na primjeru Globus sustava [55, 57]. Globus je sustav velikog razmjera (engl. *large scale*) u kojem se računski zadaci raspodjeljuju na veliki broj računala te uključuju velik broj datoteka i drugih računalnih sredstava. Sredstva u računalnim spletovima (engl. *grid*) [56] često su smještena u različitim administrativnim domenama i na različitim zemljopisnim položajima.

Sigurnosne odredbe Globus sustava

Zbog velikog broja široko raspodijeljenih korisnika i sredstava, uloga sigurnosti je ključna u Globus sustavu. U odredbama sigurnosti za Globus sustav nabrojeni su zahtjevi koje moraju ostvariti mehanizmi sigurnosti. Odredbe sigurnosti obuhvaćaju sljedećih osam zahtjeva objašnjenih u nastavku.

Okolina se sastoji od više administrativnih domena. Radna okolina Globus sustava sastoji se od više administrativnih domena, gdje svaka domena ima svoje lokalne sigurnosne odredbe. Lokalne odredbe domene nije moguće mijenjati i prilagođavati radi sudjelovanja u Globus sustavu, niti odredbe Globus sustava mogu upravljati lokalnim sigurnosnim odlukama. Stoga se odredbe Globus sustava odnose na operacije između administrativnih domena.

Lokalne operacije (tj. operacije koje se izvode samo unutar jedne domene) prepuštene su nadležnosti lokalnih sigurnosnih odredbi. Vezano uz prethodni zahtjev, Globus prepostavlja podložnost lokalnih operacija jedino lokalnim sigurnosnim odredbama. Ako je operacija započeta i izvršena unutar jedne domene, onda se sva pitanja sigurnosti rješavaju prema lokalnim odredbama te domene. Globus sustav ne nadzire sigurnost lokalnih operacija pojedinih domena.

Globalne operacije (tj. operacije koje uključuju nekoliko domena) zahtijevaju poznavanje pokretača operacije u svakoj od domena u kojima se operacija izvodi. Pokretač, bez obzira je li to korisnik ili proces koji se izvodi u korist korisnika, mora biti lokalno

poznat u svim domenama u kojima se operacija izvodi. Na primjer, globalno ime korisnika preslikava se u zasebno lokalno ime unutar pojedine domene. Pravila preslikavanja definirana su zasebno u pojedinim domenama.

Operacije između različitih domena zahtijevaju uzajamnu autentikaciju. Na primjer, ako se korisnik jedne domene koristi uslugom iz druge domene, onda se identitet korisnika provjerava u drugoj domeni. Jednako je važno sa strane korisnika uvjeriti se u identitet usluge koju se želi koristiti.

Globalna autentikacija zamjenjuje lokalne autentikacije. Prethodna dva zahtjeva objedinjuju se na sljedeći način: ako je identitet korisnika provjerjen na globalnoj razini i korisnik je pod tim identitetom poznat i u lokalnoj domeni, onda nema potrebe korisnikov identitet ponovno provjeravati u lokalnoj domeni. Autentikacija na razini Globus sustava smatra se vjerodostojnom i jamči za identitet korisnika koji je autenticiran na globalnoj razini.

Nadzor pristupa je u nadležnosti lokalne sigurnosti. Nakon što je korisnik potvrđio svoj identitet, još uvijek je potrebno provjeriti njegova prava pristupa prema sredstvu kojim se koristi. Proces nadzora pristupa sredstvima Globus sustav prepušta lokalnoj domeni u kojoj se sredstvo nalazi.

Korisnicima je omogućeno prenošenje svojih prava pristupa na procese. Globus prepostavlja scenarij u kojem tijek izvođenja procesa određuje slijed operacija koje se izvršavaju u različitim domenama. Takva vrsta procesa autonomno zastupa interes korisnika i zahtijeva prenošenje korisnikovih prava pristupa. Prenošenjem prava pristupa na procese, Globus omogućuje izvođenje autonomnih procesa koji se često nazivaju *agenti*.

Skupu procesa u istoj domeni omogućuje se dijeljenje vjerodajnica. Cilj je postići učinkovitost izvođenja skupine procesa iz iste domene ako se izvode u korist istog korisnika. Skupini procesa pruža se mogućnost međusobnog dijeljenja istog skupa vjerodajnica. Ne zahtijeva se od svakog procesa zasebni skup vjerodajnica. Opisanim pristupom želi se osigurati svojstvo razmjernog rasta autentikacije s povećanjem broja procesa.

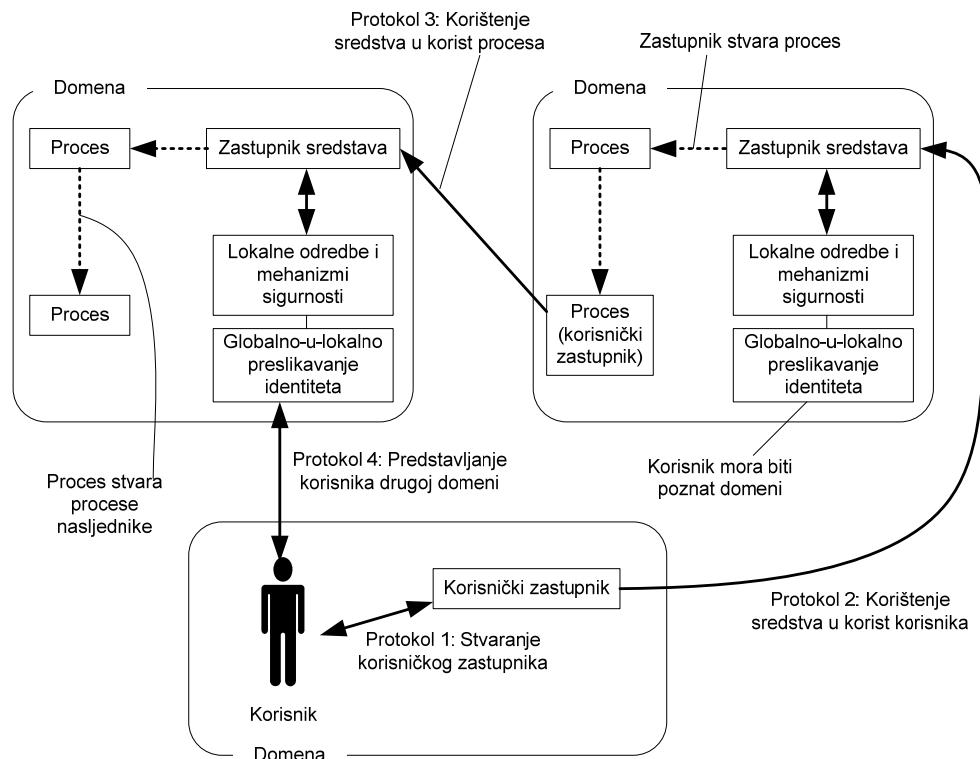
Arhitektura sigurnosti Globus sustava

Pod prepostavkom da pojedine domene ostvaruju vlastite sigurnosne odredbe, sigurnost Globus sustava usmjerenja je sigurnosnim problemima koji uključuju više domena. Oblikovana arhitektura sigurnosti usredotočena je na predstavljanje identiteta korisnika u udaljenoj domeni i korištenje sredstava iz udaljene domene. Zbog toga su u sustavu Globus

ostvareni mehanizmi za uzajamnu autentikaciju između različitih domena i prepoznavanje korisnika iz udaljene domene.

U Globus sustavu nalaze se dvije vrste zastupnika: *korisnički zastupnik* i *zastupnik sredstava*. *Korisnički zastupnik* je proces koji se izvodi u ograničenom razdoblju u korist korisnika. *Zastupnik sredstava* je proces koji se izvodi u određenoj domeni i koristi za prevodenje globalnih operacija nad sredstvima domene. Primjenom zastupnika sredstava operacije nad globalnim sredstvima prevode se u lokalne operacije koje se izvode prema pravilima lokalne domene. Tijekom pristupa sredstvima, *korisnički zastupnik* komunicira sa *zastupnikom sredstava*.

Arhitektura sigurnosti Globus sustava sastoји se od korisnika, *korisničkih zastupnika*, *zastupnika sredstava* i procesa. U arhitekturi sigurnosti definiraju se četiri protokola njihove međusobne suradnje, što je prikazano na slici 5-1.



Slika 5-1: Arhitektura Globus sustava

Prvi protokol opisuje postupak kojim korisnik stvara i prenosi svoja prava na korisničkog zastupnika. S ciljem prenošenja prava pristupa korisničkom zastupniku, korisnik predaje zastupniku odgovarajući skup vjerodajnica.

Drugi protokol definira postupak korištenje sredstva u drugoj domeni. Postupak se izvodi uporabom stvorenog korisničkog zastupnika. Nakon autentikacije korisničkog

zastupnika, u udaljenoj domeni se primjenom zastupnika sredstava stvara proces koji predstavlja korisnika. Stvoreni proces ima sličnu ulogu kao i korisnički zastupnik, samo što tu ulogu izvodi u udaljenoj domeni. Stvorenom procesu odobrava se pristup sredstvu na osnovi pravila nadzora pristupa lokalne domene tog procesa.

Ako proces stvoren u udaljenoj domeni pokreće dodatna izračunavanja u drugim domenama, onda se koristi treći protokol pomoću kojega proces zahtijeva korištenje sredstva u drugoj domeni. U tom slučaju proces stvara svog korisničkog zastupnika, slično kao što to radi korisnik. Preko svog korisničkog zastupnika proces zahtijeva korištenje sredstva u udaljenoj domeni.

Posljednjim protokolom definira se način predstavljanja korisnika u drugoj domeni. Prepostavlja se da korisnik u toj domeni ima definiran svoj korisnički račun. U tom slučaju potrebno je ostvariti pretvorbu vjerodajnica koje se dodjeljuju korisničkom zastupniku u vjerodajnice prilagođene udaljenoj domeni. Opisanu pretvorbu vjerodajnica moguće je obaviti preko posebnih poslužitelja kao što su CAS, VOMS i sl. Navedeni sustavi opisani su u nastavku poglavlja. Pojedinosti svih protokola sigurnosti koji se primjenjuju u Globus sustavu opisani su u [57].

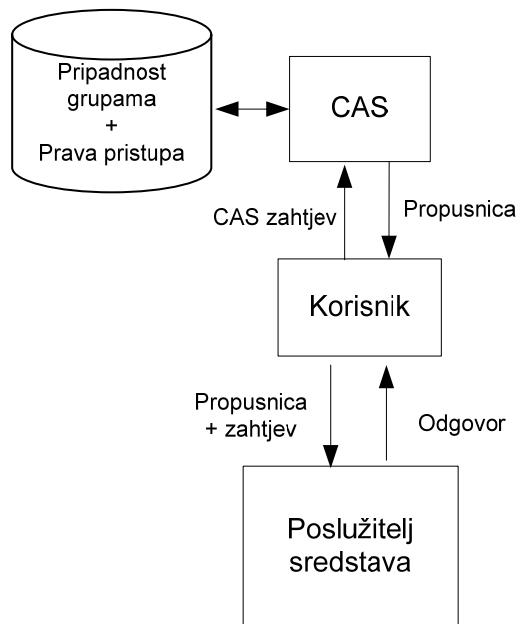
5.2. CAS sustav

Osnovni problem koji se rješava u sustavima s CAS poslužiteljem (*Community Authorization Server*) [58] jest predstavljanje odredbi nadzora pristupa vjerodajnicama u raspodijeljenim *prividnim zajednicama* (engl. *virtual organization*). Prividne zajednice dinamičke su tvorevine stvorene od više administrativnih domena koje su povezane suradnjom na zajedničkom zadatku. U takvom okruženju javlja se problem izražavanja različitih sigurnosnih odredbi i prilagođavanja sigurnosnih odredbi između pojedinačnih administrativnih domena, te organiziranja sigurnosnih odredbi u hijerarhijske strukture.

CAS poslužitelj odgovoran je za upravljanje odredbama koje rukovode procesom nadzora pristupa sredstvima u prividnoj zajednici. *CAS poslužitelj* sadrži zapise o izdavateljima vjerodajnica, korisnicima, poslužiteljima i sredstvima koja čine zajednicu te o grupama u koje su oni organizirani. Osim toga, sadrži pravila koja definiraju *tko* (koja grupa) ima pravo pristupa, koju *vrstu* pristupa ima i nad *kojim* sredstvima. *CAS poslužitelj* koristi DAC model nadzora pristupa te omogućuje prilagodljivost i visoku razlučivost nadzora pristupa nad operacijama koje je moguće izvoditi.

Na slici 5-2 prikazan je tijek akcija i uloga *CAS poslužitelja* u autentikaciji i autorizaciji korisnika za ostvarenje prava pristupa sredstvu iz druge domene. U CAS sustavu

sudjeluju korisnik, *CAS poslužitelj* i *poslužitelj sredstava*. Korisnik želi koristiti sredstvo poslužitelja iz druge domene, ali nema odgovarajuća prava pristupa ili ima neprikladan oblik vjerodajnice s pravima pristupa. Korisnik se stoga u *CAS zahtjevu* autenticira *CAS poslužitelju* i traži propusnicu s pravima pristupa za izvođenje skupa akcija nad sredstvima druge domene. *CAS poslužitelj* vraća propusnicu u odgovoru. Korisnik šalje zahtjev za korištenjem sredstva *poslužitelju sredstva* u drugoj domeni i prikazuje mu dobivenu propusnicu. Pristup sredstvu ovisi o tome dopušta li *poslužitelj sredstva* zajednici pravo raspolaganja njegovim sredstvima. Ako zajednica ima odobrenje pristupa lokalno na poslužitelju, ispituju se prava pristupa sadržana u propusnici. Ako propusnica u sebi sadrži zatraženo pravo pristupa, onda *poslužitelj sredstva* odobrava korisniku pristup do traženog sredstva i vraća mu odgovor.



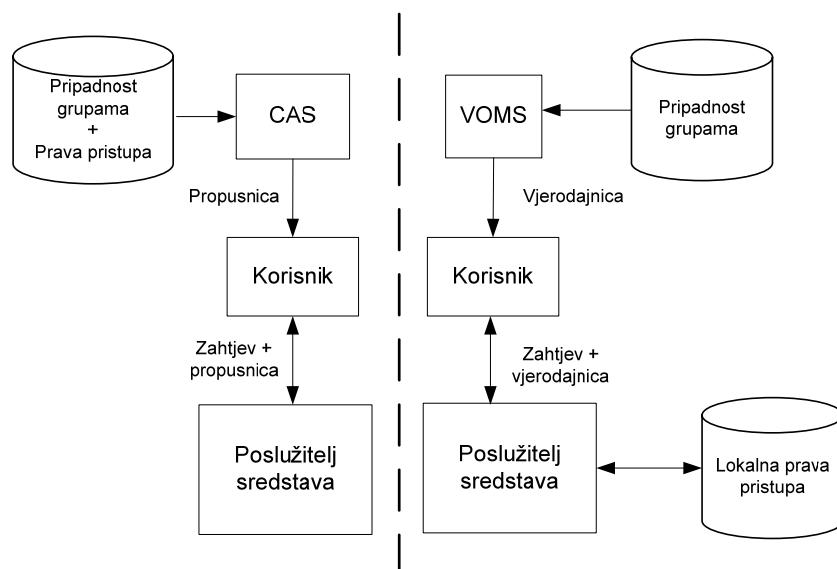
Slika 5-2: Način primjene CAS sustava

5.3. VOMS sustav

Sustav VOMS (*Virtual Organization Membership System*) [58, 60] zasniva se na *VOMS poslužitelju* koji služi za potporu upravljanja nadzorom pristupa u prividnim zajednicama. VOMS je sustav znatno sličan sustavu CAS. Oba sustava rješavaju problem predstavljanja subjekta u različitim administrativnim domenama raspodijeljene okoline opisane zahtjevima sustava Globus. Način komunikacije korisnika s VOMS poslužiteljem i poslužiteljem sredstava sličan je kao u sustavu CAS. Za izvođenje operacije u udaljenoj domeni prividne zajednice korisnik traži kratkoročnu vjerodajnicu od VOMS poslužitelja zajednice. VOMS poslužitelj stvara korisnikovu vjerodajnicu i šalje ju korisniku. S

dobivenom vjerodajnicom, korisnik zahtjeva sredstvo od poslužitelja sredstva iz druge domene. Poslužitelj sredstva odobrava ili odbija korisnikov zahtjev.

CAS i VOMS razlikuju se u izražajnosti koju pružaju. Izdavanjem vjerodajnice, VOMS poslužitelj postiže veću izražajnost nego što to CAS postiže izdavanjem propusnice. Vjerodajnica koju izdaje VOMS, osim propusnice, sadržava podatke o članstvu korisnika u grupama te korisnikovim ulogama u prividnoj zajednici. Uporabom propusnice u vjerodajnici, VOMS poslužitelj omogućuje isto što i CAS poslužitelj. Razlika između sustava VOMS i CAS prikazana je na slici 5-3.



Slika 5-3: Razlika između sustava VOMS i CAS

CAS izravno izdaje propusnice korisniku. Izravnim izdavanjem propusnice, poslužitelju sredstava oduzima se nadležnost nadzora pristupa nad vlastitim sredstvima i nadležnost se predaje CAS administratoru. Lokalni administrator odobrava ili zabranjuje zajednici pristup, ali ne može utjecati na ono što zajednica ima pravo koristiti u njegovoj lokalnoj domeni. Time se narušava osnovno načelo računalnih spletova, gdje svatko upravlja svojim sredstvima i odlučuje koja sredstva želi ponuditi zajednici.

VOMS dodatno omogućuje više kontrole u nadležnosti lokalnog poslužitelja sredstava. Korisnikova vjerodajnica sadrži podatke o grupi ili ulozi korisnika u zajednici, ali ne sadrži propusnicu s pravima pristupa. Nepostojanjem propusnice, lokalnom se poslužitelju sredstava ne nameću prava pristupa zajednice koju predstavlja VOMS poslužitelj. Time je poslužitelju sredstva omogućeno da na osnovi grupe i uloga nadzire pristup zajednici svojim sredstvima.

5.4. PRIMA

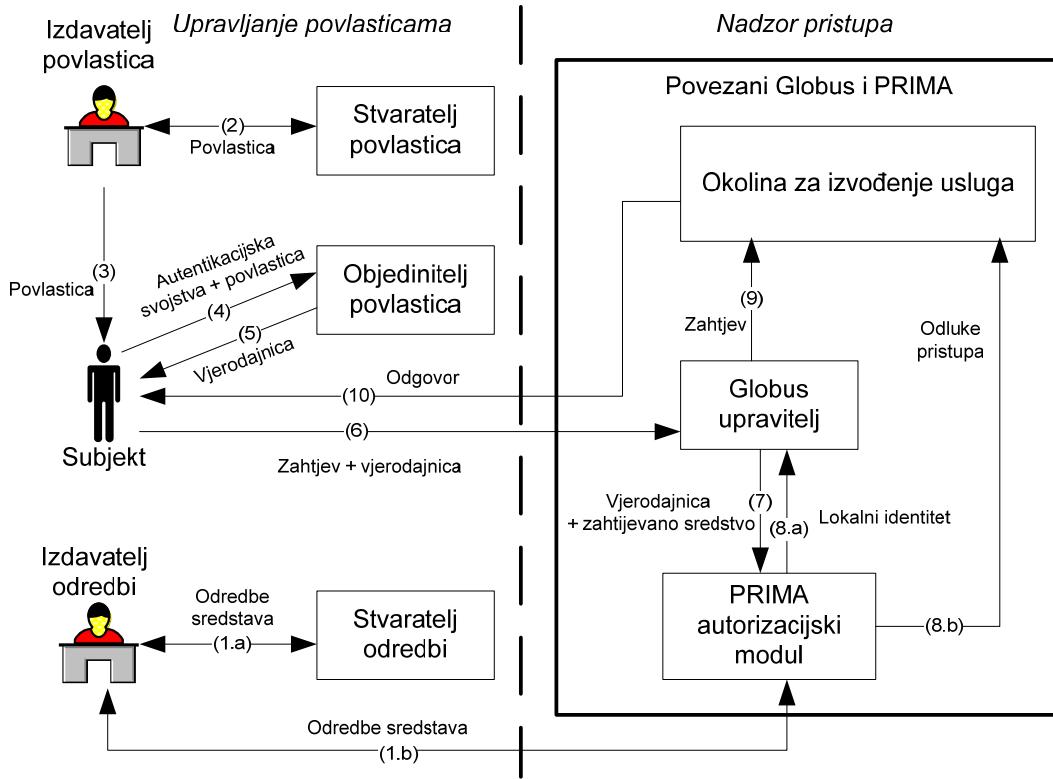
PRIMA [60] (*System for Priviledge Management*) je sustav za nadziranje pristupa sredstvima na osnovi *povlastica* (engl. *privilege*). Povlastica je skup različitih svojstava subjekta koji subjektu omogućuju pristup sredstvima. PRIMA omogućuje uspostavu pojedinačnog povjerenja između dva subjekta iz različitih administrativnih domena i pri tome ne zahtijeva formalno povjerenje između samih domena. PRIMA sustav oblikovan je za potpuno raspodijeljenu okolinu i omogućuje potporu za stvaranje malih spontanih prijelaznih zajednica. PRIMA ne nameće zahtjeve za ostvarivanje zajedničke infrastrukture, kao što su primjerice poslužitelji zajednice. Međutim, ako su takvi poslužitelji postavljeni (npr. VOMS poslužitelj), PRIMA ih može koristiti za napredno upravljanje većim zajednicama.

PRIMA omogućuje uporabu najmanje povlastice za pristup sredstvima. Korisnik prema vlastitom izboru grupira svoje dozvole za pristup u povlastice i ima mogućnost upućivanja zahtjeva s najmanjim skupom povlastica dovoljnim za pristup. Time se smanjuje rizik od zlouporabe korisnikovih povlastica u slučaju narušavanja sigurnosti povlastice. Dodatno, korisnicima je dopušteno prenositi povlastice nad sredstvima za koja su nadležni. PRIMA podupire dinamičko stvaranje korisničkih računa s povlasticama koje izdaju ovlaštene strane (administratori sredstava, vlasnici sredstava, voditelji grupa i slično).

PRIMA primjenjuje povlastice u obliku X.509 iskaznice svojstava (engl. *attribute certificate, AC*) [62] i time na siguran način pripisuje povlastice korisnicima. Osim povlastica, PRIMA dodatno primjenjuje odredbe sredstava zapisane u obliku X.509 iskaznice svojstava i time na siguran način povezuje odredbe sa sredstvima. Jedan dio sustava PRIMA posvećen je administriranju povlastica i odredbi, dok je drugi dio upravna i izvršna logika. Upravna logika u obliku autorizacijskog modula Globus sustava donosi odluke o pristupu na osnovi kombinacije korisničkih povlastica i odredbi sredstava. Izvršna logika provodi nadzor pristupa koristeći se *podacima* upravne logike i ACL listama jedne vrste datotečnog sustava. Podržan je i XML oblik ACL listi korišten u projektu Slashgrid [62].

Arhitektura PRIMA sustava prikazana je na slici 5-4. Ovlašteni *Izdavatelj odredbi* (engl. *policy authority*) pomoću *Stvaratelja odredbi* (engl. *policy creator*) stvara odredbe pristupa sredstvima (1.a) te ih šalje PRIMA autorizacijskom modulu (1.b). *Izdavatelji povlastica* (engl. *attribute authority*) koriste *Stvaratelja povlastica* (engl. *privilege creator*) za stvaranje povlastica pojedinih subjekata (2). Subjekti izabiru povlastice koje žele koristiti uz određenu skupinu zahtjeva (3). Dodatno, subjekti grupiraju povlastice sa svojim

autentikacijskim svojstvima koristeći *Objedinitelj povlastica* (engl. *priviledge combinator*) (4). Rezultat je vjerodajnica koju korisnik upotrebljava pri postavljanju zahtjeva (5).



Slika 5-4: Arhitektura PRIMA sustava

Pristup subjekta sredstvima nadzire se zajedničkim međudjelovanjem sustava PRIMA s dijelovima sustava *Globus*. *Globus upravitelj* (engl. *Globus gateway*) prima zahtjeve i vjerodajnice subjekata (6). Iz zahtjeva zatim izdvaja podatak o zahtijevanom sredstvu, te šalje *PRIMA autorizacijskom modulu* vjerodajnicu i podatak o zahtijevanom sredstvu (7).

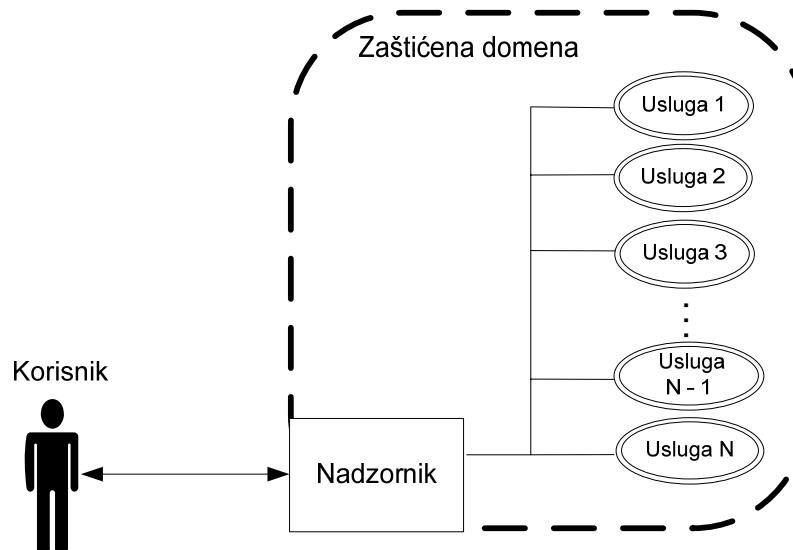
PRIMA autorizacijski modul predstavlja glavni dio nadzora pristupa. On ostvaruje upravnu logiku nadzora pristupa preslikavanjem identiteta subjekta i donošenjem odluka o nadzoru pristupa. Preslikavanje identiteta subjekta izvodi se na osnovi vjerodajnice. Identitet subjekta sadržan u vjerodajnici preslikava se u korisnički identitet lokalne domene. Preslikani lokalni identitet vraća se *Globus upravitelj* (8.a). Odluke o nadzoru pristupa donose se na osnovi lokalnog identiteta i odredbi sredstava pohranjenih u *PRIMA autorizacijskom modulu*. Primjenom lokalnih korisničkih identiteta, odredbe sredstava definiraju prava pristupa datotekama, ograničenje mrežnog pristupa i slično. Odluke pristupa šalju se *Okolini za izvođenje usluga* koja služi kao izvršna logika (8.b). Izvršna logika

primjenjuje odluke pristupa za posluživanje subjektova zahtjeva što ga *Globus upravitelj* prosljeđuje *Okolini za izvođenje usluga* (9, 10).

6. Nadziranje pristupa računalnim sustavima zasnovanim na uslugama

Nadziranje pristupa uslugama jedna je od osnovnih funkcionalnosti sustava zasnovanih na uslugama. Mehanizme za nadziranje pristupa uslugama u sustavima zasnovanim na uslugama poželjno je ostvariti u okviru jedinstvenog sigurnosnog podsustava. Primjena jedinstvenog sigurnosnog podsustava ubrzava razvoj i olakšava upravljanje uslugama u sustavima zasnovanim na uslugama. Nadalje, tijekom razvoja novih usluga potrebno je samo izgraditi poslovnu logiku usluge, dok se sigurnost usluga ostvaruje primjenom jedinstvenog sigurnosnog podsustava.

U praktičnom dijelu magistarskog rada oblikovan je i razvijen sustav *Nadzornik* [65]. Sustav *Nadzornik* omogućava nadzor pristupa uslugama računalnih sustava zasnovanih na uslugama. Sustav je razvijen u suradnji tvrtke Ericsson Nikola Tesla d.d. i Fakulteta elektrotehnike i računarstva. Suradnja je potpomognuta nacionalnim tehnološkim projektom *CroGRID* pod pokroviteljstvom Ministarstva znanosti, obrazovanja i športa. Sustav *Nadzornik* je dio računalne okoline za izgradnju raspodijeljenih primjenskih sustava zasnovanih na uslugama [66 – 73].



Slika 6-1: Okolina i primjena sustava *Nadzornik*

Slika 6-1 prikazuje okolinu i primjenu sustava *Nadzornik*. Sustav *Nadzornik* primjenjuje se za izgradnju zaštićene domene nad skupom usluga. Dodatno, sustav u zaštićenoj domeni nadzire pristup korisnika do usluga. Sustav *Nadzornik* ostvaruje funkcionalnost sigurnosnog posrednika u komunikaciji korisnika i usluga. Samo ovlašteni

korisnici imaju pravo pristupa uslugama u zaštićenoj domeni pri čemu se štiti sigurnost korisnika i usluga.

Zahtjevi sigurnosti koji određuju svojstva sustava *Nadzornik* opisani su u odjeljku 6.1. Upravljački podaci na kojima se zasniva nadzor pristupa definirani su u odjeljku 6.2. Osnovne funkcijeske cjeline sustava *Nadzornik* opisane su u poglavljju 6.3. Protokoli komunikacije i razmjene upravljačkih podataka opisani su u odjeljku 6.4. Arhitektura sustava *Nadzornik* prikazana je u odjeljku 6.5.

6.1. Zahtjevi sigurnosti

Arhitektura sustava *Nadzornik* oblikovana je na osnovi sljedeća četiri zahtjeva: sustav *Nadzornik* ostvaruje zaštićenu domenu za nadzor pristupa uslugama, omogućuje prijavu novih usluga i korisnika u zaštićenu domenu, prilagođava nadzor pristupa zahtjevima usluga i zasniva nadzor pristupa na povjerenju u administratora sustava.

Usluge su osnovne programske jedinke za koje sustav *Nadzornik* ostvaruje nadzor pristupa. Nadzor pristupa skupu usluga ostvaruje se izgradnjom zaštićene domene. Zaštićena domena izgrađuje se ostvarenjem upravitelja primjenske razine. Upravitelj primjenske razine upravlja prosljeđivanjem poruka zahtjeva uslugama u zaštićenoj domeni. Upravljanje prosljeđivanjem definira se upravljačkim podacima nadzora pristupa.

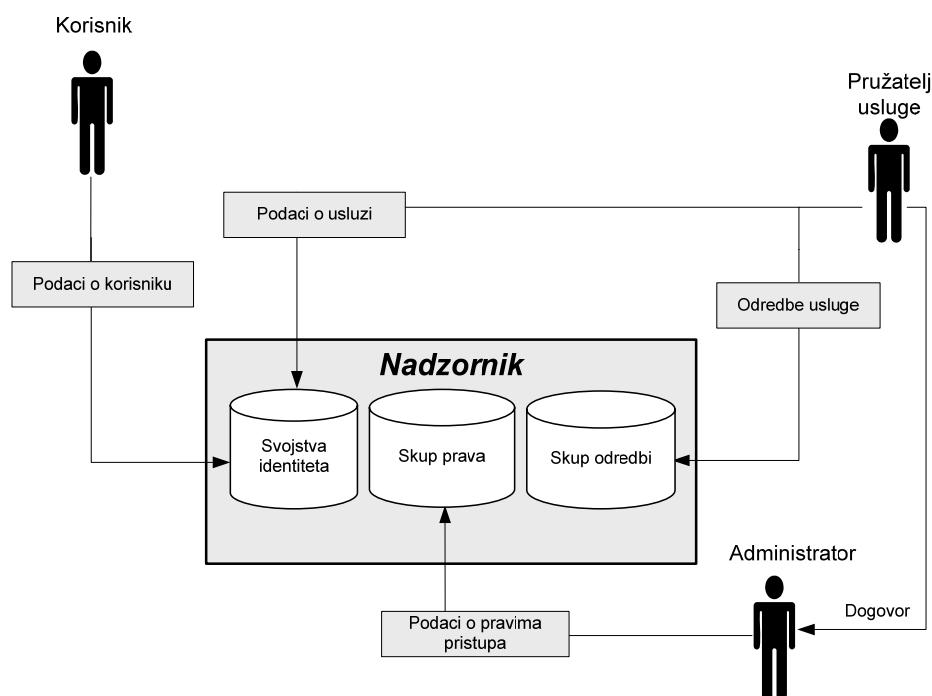
Skup usluga zaštićene domene sustava *Nadzornik* moguće je proširiti dodavanjem novih usluga. Dodavanje novih usluga proširuje domenu novim funkcionalnostima te omogućava rast i razvoj domene. Dodavanje usluga i korisnika ostvaruje se javno dostupnim postupcima registracije i autentikacije.

Pružatelji usluga definiraju sigurnosne zahtjeve usluga prilikom registriranja usluga u zaštićenu domenu. Sigurnosni zahtjevi pružatelja usluga definiraju se u odredbama usluga. Na osnovi definiranih odredbi, mehanizam nadzora pristupa odlučuje koje postupke sigurnosti treba izvoditi tijekom nadzora pristupa usluzi. Postupci sigurnosti obuhvaćaju provjeru identiteta korisnika usluge, provjeru autorizacije korisnika usluge i bilježenje korištenja usluge.

Prijava usluga u zaštićenu domenu sustava *Nadzornik* zasniva se na povjerenju pružatelja usluge u administratora sustava *Nadzornik*. Administrator sustava ovlašten je postavljati prava za pristup uslugama u zaštićenoj domeni. Administrator definira prava pristupa usluzi prema dogovoru s pružateljem usluge. Na osnovi definiranih prava pristupa izvodi se sigurnosni postupak provjere autorizacije za korištenje usluge.

6.2. Upravljački podaci nadzora pristupa

Sustav *Nadzornik* ostvaruje nadzor pristupa uslugama uporabom upravljačkih podataka. Upravljački podaci grupirani su u sljedeće tri skupine podataka: svojstva identiteta, skup prava i skup odredbi. Slika 6-2 prikazuje razredbu upravljačkih podataka i definira koje od navedenih podataka u sustav unose korisnici, pružatelji usluga i administratori.



Slika 6-2: Upravljački podaci za nadzor pristupa u sustavu *Nadzornik*

Upravljački podaci *svojstva identiteta* sadrže informacije koje se primjenjuju tijekom autentikacije usluga i korisnika registriranih u sustavu *Nadzornik*. Osim ovih informacija, *svojstva identiteta* sadrže opise posebnih svojstava korisnika i usluga. Podaci *svojstva identiteta* izgrađeni su od skupa podataka o korisnicima i skupa podataka o uslugama. Podaci o korisniku sadrže informacije koje opisuju korisnika, a definira ih korisnik tijekom registriranja u sustav *Nadzornik*. Podaci o usluzi sadrže informacije koje opisuju uslugu, a definira ih pružatelj usluge tijekom registriranja usluge u sustav *Nadzornik*.

Upravljački podaci *skup odredbi* definiraju uvjete korištenja usluga registriranih u sustavu *Nadzornik*. Sustav *Nadzornik* na osnovi ovih podataka odabire odgovarajuće mehanizme za nadziranje pristupa pojedinoj usluzi u zaštićenoj domeni. Podaci *skup odredbi* izgrađeni su od skupa odredbi usluga koje definiraju pružatelji usluga. U odredbama usluga definiraju se operacije koje usluga nudi za korištenje. Osim toga, definira se je li tijekom

pristupa usluzi potrebno provjeriti identitet i prava pristupa korisnika, te pratiti korištenje usluge.

Upravljački podaci *skup prava* primjenjuju se za provjeru prava pristupa pojedinim operacijama usluge. Navedene podatke definira administrator sustava u dogovoru s pružateljem usluge. Administrator za svaku uslugu definira grupe korisnika ili pojedinačne korisnike s pravima pristupa pojedinim operacijama usluge.

6.2.1. Podaci o korisniku

Podaci o korisniku primjenjuju se za utvrđivanje identiteta i svojstava korisnika koji se prijavljuje u sustav postupkom autentikacije. Podaci o korisniku sadrže autentikacijske podatke korisnika i svojstva pridružena korisniku. Tablica 6-1 opisuje strukturu podataka o korisniku.

Tablica 6-1: Struktura podataka o korisniku

Podaci o korisniku	
Autentikacijski podaci	Korisničko ime
	Zaporka
Pridružena svojstva	Puno ime i prezime
	Jedinstveni matični broj građana (JMBG)
	Telefonski podaci
	Podaci adrese elektroničkog poštanskog pretinca
	Dodatni podaci

Autentikacijski podaci omogućavaju provjeru identiteta korisnika koji se prijavljuju u sustav *Nadzornik*. Ovi podaci sastoje se od korisničkog imena i zaporce. Tijekom prijave korisnika u sustav, korisnik unosi vlastito korisničko ime i tajnu zaporku. Sustav *Nadzornik* omogućava prijavu korisnika samo ako zadanim korisničkom imenu odgovara unesena zaporka.

Pridružena svojstva sadrže informacije koje dodatno opisuju identitet korisnika. Pridružena svojstva sadrže ime i prezime korisnika, jedinstveni matični broj građana (JMBG), telefonske podatke, adresu elektroničke pošte i skup dodatnih podataka. Dodatni podaci omogućavaju unošenje dodatnih proizvoljnih svojstava identiteta korisnika.

6.2.2. Podaci o usluzi

Podaci o usluzi primjenjuju se za utvrđivanje identiteta i svojstava usluge koju pružatelj usluge prijavljuje u sustav. Podaci o usluzi sadrže autentikacijske podatke i svojstva pridružena usluzi. Tablica 6-2 opisuje strukturu podataka o usluzi.

Tablica 6-2: Struktura podataka o usluzi

Podaci o usluzi	
Autentikacijski podaci	Ime usluge
	Zaporka
Pridružena svojstva	Puno ime usluge
	Podaci pripadnosti
	Dodatni podaci

Autentikacijski podaci omogućavaju provjeru identiteta usluge koju se prijavljuje u sustav *Nadzornik*. Ovi podaci sastoje se od korisničkog imena i zaporce. Navedeni podaci koriste se tijekom prijave usluge u sustav. Sustav *Nadzornik* omogućava prijavu usluge u sustav samo ako se tijekom prijave uz ime usluge navede odgovarajuća zaporka.

Pridružena svojstva sadrže informacije koje dodatno opisuju identitet usluge. Ovi podaci sadrže puno ime usluge, podatke pripadnosti i skup dodatnih podataka. Podaci pripadnosti određuju primjenski sustav u korist kojeg se pruža uslugu. Dodatni podaci omogućavaju unošenje dodatnih proizvoljnih svojstava identiteta usluge.

6.2.3. Odredbe usluga

Upravljački podaci *odredbe usluge* (engl. *service access policy*) definiraju način na koji je moguće korisiti uslugu. Ovi podaci sadrže postavke ponude, postavke provjere pristupa i postavke praćenja korištenja. Tablica 6-3 prikazuje strukturu podataka o odredbama usluge.

Podaci o *postavkama ponude* sadrže adresu na kojoj je dostupna usluga, adresu na kojoj je dostupan opis pristupnog sučelja usluge i popis operacija usluge koje je moguće koristiti. *Postavke provjere pristupa* definiraju treba li tijekom nadzora pristupa usluzi provjeravati identiteta korisnika i provjeru autorizacije korisnika. Zastavica identifikacije određuje treba li provjeravati identitet korisnika za svaki pristup usluzi. Zastavica autorizacije određuje treba li provjeravati korisnikova prava pristupa za svaki pristup usluzi. *Postavke praćenja korištenja* upravljaju postupkom praćenja korištenja usluge. Ovi podaci

sadrže zastavicu praćenja korištenja i predložak praćenja korištenja. Ako je zastavica praćenja korištenja postavljena, pri korištenju usluge treba stvarati bilješke korištenja. Predložak praćenja korištenja definira podatke koji se postupkom praćenja korištenja zapisuju u bilješkama korištenja [74].

Tablica 6-3: Struktura odredbi usluge

Odredbe usluge	
Postavke ponude	Adresa usluge
	Adresa opisnika usluge
	Popis ponudenih operacija
Postavke provjere pristupa	Zastavica identifikacije
	Zastavica autorizacije
Postavke praćenja korištenja	Zastavica praćenja korištenja
	Predložak praćenja korištenja

6.2.4. Podaci o pravima pristupa

Podaci o pravima pristupa primjenjuju se za dozvoljavanje korištenja usluga. Podaci o pravima pristupa definiraju se uporabom grupa i dozvola pristupa. *Grupa* je skupina korisnika koje administrator udružuje radi jednostavnijega postavljanja dozvola pristupa. U tablici 6-4 kao primjer su prikazane grupa A i grupa B. Grupu A čine korisnici Marko i Ana, te svi korisnici koji su članovi grupe B. Nadalje, grupu B čine korisnici Ankica i Branko.

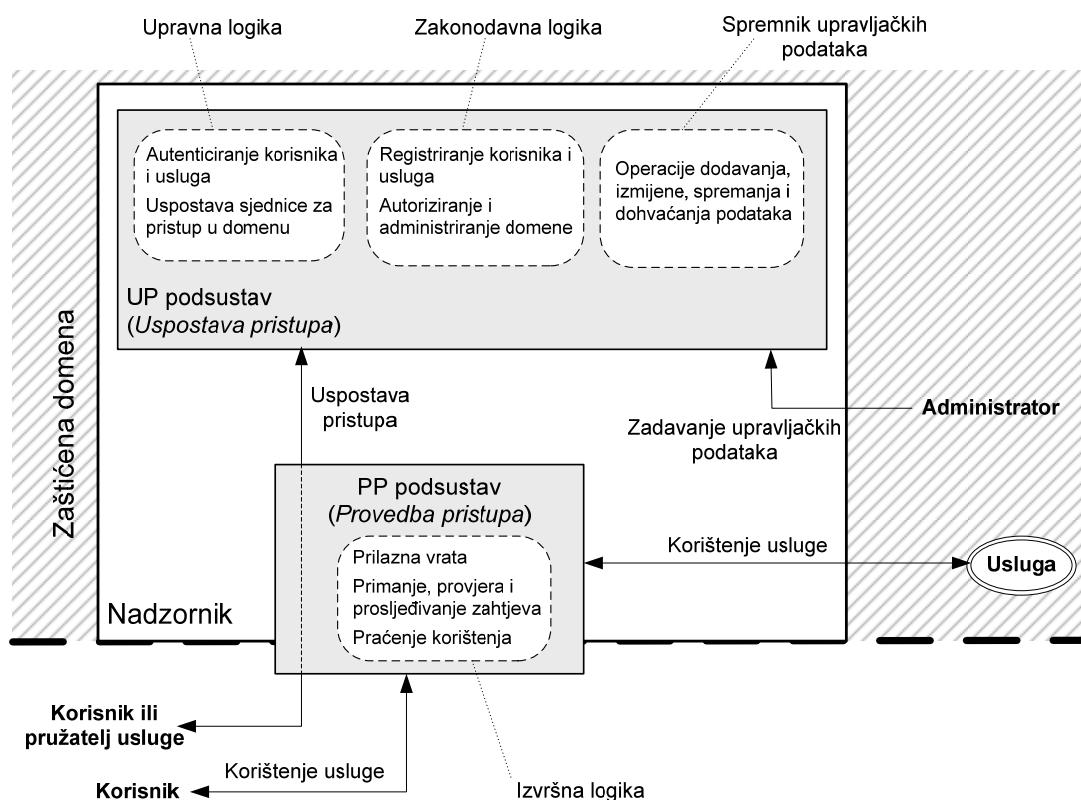
Tablica 6-4: Struktura podataka o pravima pristupa prikazana primjerom

Podaci o pravima pristupa			
Grupe	Ime grupe		Ime korisnika
	Grupa A		Marko
			Ana
			Grupa B
	Grupa B		Ankica
			Branko
Dozvole pristupa	Korisničko ime ili ime grupe	Ime usluge	Ime operacije
	Ana	Vijesti	Kultura
			Svijet
			Šport
		Izletnik	Pregled odredišta
			Pregled hotela
			Konačna cijena
	Grupa B	Vijesti	Sport
			Zabava

Dozvole pristupa (engl. *permission*) sadrže informacije o akcijama koje određeni korisnik ili grupa ima pravo izvesti. U sustavu *Nadzornik* akcija je poziv operacije usluge. Akcije su definirane imenom usluge i imenom operacije. U tablici 6-4 kao primjer prikazane su dozvole pristupa za korisnika Ana i grupu B. Korisnik Ana ima dozvolu pozivanja operacija "Kultura", "Svijet" i "Šport" koje pruža usluga Vijesti, te operacija "Pregled odredišta", "Pregled hotela" i "Konačna cijena" koje pruža usluga Izletnik. Korisnici u grupi B imaju dozvolu pozivanja operacija "Sport" i "Zabava" koje pruža usluga Vijesti.

6.3. Osnovne funkcionalnosti sustava Nadzornik

Podjela funkcionalnosti u sustavu *Nadzornik* prikazana je slikom 6-3. Dvije osnovne funkcionalnosti, *Uspostava pristupa* i *Provedba pristupa*, ostvarene su primjenom *UP* i *PP* podsustava. *UP* podsustav izvodi postupke za stvaranje upravljačkih podataka za nadzor korištenja usluga u domeni. *PP* podsustav primjenjuje postojeće upravljačke podatke i na osnovi njih nadzire korištenje usluga.



Slika 6-3: Osnovne funkcione celine sustava *Nadzornik*

UP podsustav dostupan je korisnicima i pružateljima usluga za uspostavljanje pristupa posredovanjem *PP* podsustava. *UP* podsustav prima zahteve uspostave pristupa na osnovi kojih stvara i spremi upravljačke podatke korisnika i usluga. *PP* podsustav na osnovi

raspoloživih upravljačkih podataka ne prihvaca ili prihvaca zahtjeve korištenja usluga. Ako upravljački podaci korisnika ili usluge nisu spremjeni u *UP* podsustavu, korisnik i usluga nemaju pravo pristupa domeni.

Funkcionalnosti *UP* i *PP* podsustava razrađene su i grupirane prema IETF radnom okviru nadzora pristupa koji je opisan u odjeljku 3.4.1. *UP* podsustav ostvaruje funkcionalnosti spremnika upravljačkih podataka, zakonodavne logike i upravne logike. *PP* podsustav ostvaruje funkcionalnost izvršne logike.

Spremnik upravljačkih podataka pruža osnovne operacije spremanja, dohvaćanja, dodavanja i izmjene upravljačkih podataka. *Zakonodavna logika* omogućuje korisnicima, pružateljima usluga i administratorima zadavanje upravljačkih podataka. Funkcionalnosti zakonodavne logike obuhvaćaju postupke registriranja korisnika i usluga, autoriziranje korisnika i administriranje domene. Korisnici i usluge primjenjuju zakonodavnu logiku kako bi registrirali svoje upravljačke podatke. Korisnici registriraju podatke o korisniku, dok pružatelji usluga registriraju podatke o usluzi i odredbama usluge. Administrator autorizira korisnike i administrica upravljačke podatke. Funkcionalnosti *upravne logike* obuhvaćaju autenticiranje korisnika i usluge te uspostavljanje sjednice za pristup uslugama u domeni. Autentikacijom se provjerava identitet korisnika i usluga te uspostavlja sjednica. Postupak uspostave sjednice obuhvaća pripremanje upravljačkih podataka potrebnih izvršnoj logici za nadzor pristupa.

Funkcionalnosti *izvršne logike* obuhvaćaju primanje i prosljeđivanje korisničkih zahtjeva, ispitivanje upravljačkih podataka i praćenje korištenja. Izvršna logika prima korisničke zahtjeve, a prosljeđivanje izvodi tek ako ispitivanjem upravljačkih podataka utvrdi da je postavljeni zahtjev odobren. Ako upravljački podaci zahtijevaju praćenje korištenja, izvršna logika bilježi podatke o postavljenim zahtjevima.

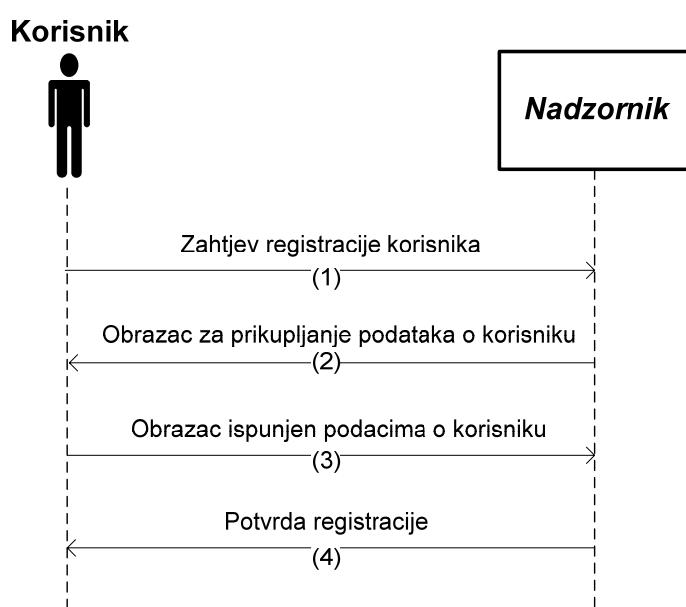
6.4. Protokoli razmjene podataka

Protokoli razmjene podataka u sustavu *Nadzornik* dijele se na protokole dogovora pristupa i protokole provedbe dogovora. U protokole dogovora pristupa ubrajaju se protokoli registracije i autentikacije. Navedenim protokolima definira se komunikaciju korisnika i usluga sa sustavom *Nadzornik*. Protokolom registracije u sustav se unose upravljački podaci o korisnicima i uslugama. Protokolom autentikacije uspostavlja se i obustavlja sjednica između korisnika ili usluge i sustava. Dodatno, postupkom uspostave sjednice razmjenjuju se i pripremaju upravljački podaci potrebni za protokol provedbe dogovora. Protokol provedbe dogovora je protokol ispitivanja upravljačkih podataka za nadzor pristupa. Protokolom

ispitivanja upravljačkih podataka provjeravaju se prava korištenja usluga te se donosi odluka o dozvoli pristupa.

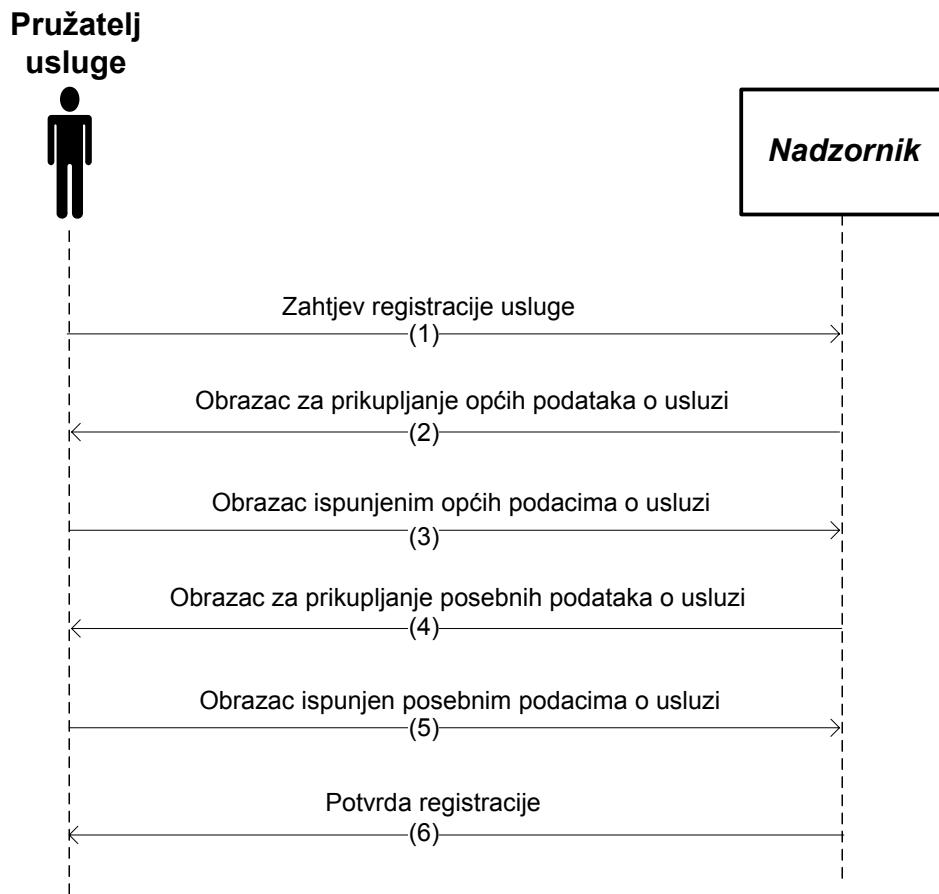
6.4.1. Protokol registracije

Razlikuju se dvije vrste protokola registracije ovisno o tome registrira li se korisnik ili usluga. Slikom 6-4 prikazan je protokol registracije korisnika. Korisnik započinje protokol registracije slanjem zahtjeva registracije sustavu *Nadzornik* (1). Sustav *Nadzornik* odgovara na zahtjev slanjem obrasca za prikupljanje podataka o korisniku (2). Korisnik popunjava tražene podatke u obrascu i šalje ispunjeni obrazac do sustava *Nadzornik* (3). *Nadzornik* čita podatke iz obrasca, zapisuje ih u spremnik i vraća potvrdu registracije korisniku (4).



Slika 6-4: Protokol registracije korisnika u sustav *Nadzornik*

Slika 6-5 prikazuje protokol registracije usluge. Pružatelj usluge započinje protokol slanjem zahtjeva registracije sustavu *Nadzornik* (1). *Nadzornik* odgovara slanjem obrasca za prikupljanje *općih podataka* o usluzi (2). Opći podaci o usluzi su podaci koji sadrže informacije o usluzi neovisne o vrsti usluge. Na primjer, adresa opisnika usluge je opći podatak o usluzi. Pružatelj usluge popunjava obrazac i šalje ispunjeni obrazac do sustava *Nadzornik* (3). *Nadzornik* tumači podatke iz obrasca, na osnovi njih dinamički stvara obrazac prilagođen za prikupljanje *posebnih podataka* o usluzi i šalje ga pružatelju usluge (4). Posebni podaci o usluzi su podaci koji ovise o vrsti usluge. Na primjer, posebni podaci o usluzi su popis operacija koje usluga nudi. Pružatelj usluge popunjava obrazac posebnim podacima i šalje ga sustavu *Nadzornik* (5). *Nadzornik* zapisuje podatke prikupljene iz oba obrasca u spremnik podataka, te vraća potvrdu registracije pružatelju usluge (6).



Slika 6-5: Protokol registracije usluge u sustav *Nadzornik*

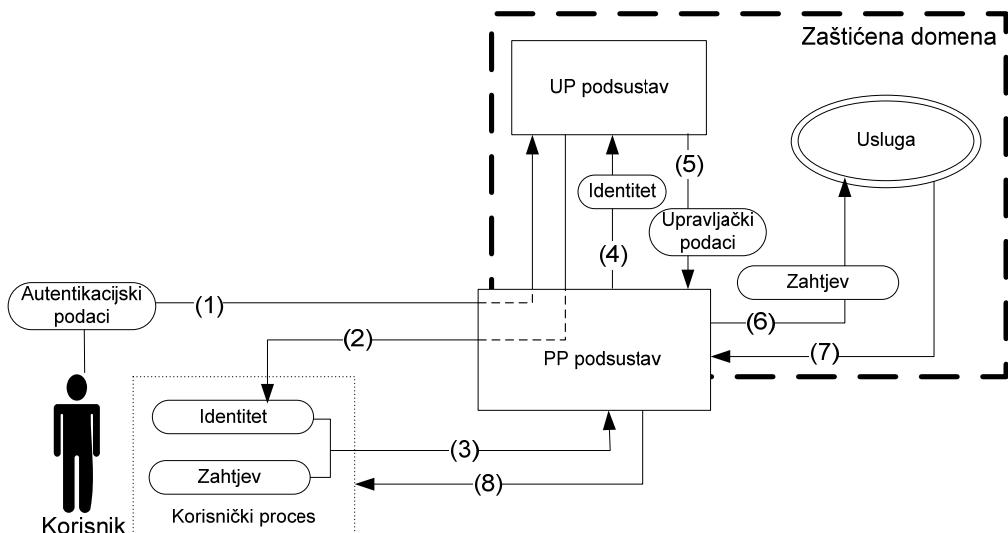
6.4.2. Pripremanje upravljačkih podataka

Nakon što su upravljački podaci korisnika spremjeni u sustavu, postupkom autentikacije ti se podaci pripremaju za nadzor korištenja usluga u domeni sustava *Nadzornik*. Priprema upravljačkih podataka obuhvaća postupke stvaranja dodatnih podataka sjednice, obrade i pretvorbe podataka te razmjenu podataka između *UP* podsustava, *PP* podsustava i korisnika koji se autenticira.

Postoje dva klasična modela pripreme upravljačkih podataka. Oba modela zasnivaju se na modelima suradnje izvršne i upravne logike: modelu *potraživanja* i modelu *ponude*. Navedeni modeli suradnje izvršne i upravne logike opisani su u odjeljku 3.4.1. U ostvarenju sustava *Nadzornik*, *UP* podsustav i *PP* podsustav raspodijeljeni su na različita računala. Nadzor pristupa uporabom upravljačkih podataka pripremljenih primjenom klasičnih modela nije prilagođen raspodijeljenoj okolini u kojoj komuniciraju *UP* i *PP* podsustavi sustava *Nadzornik*. Stoga je osmišljen model pripremanja upravljačkih podataka prilagođen okolini u kojoj komuniciraju udaljeni *UP* i *PP* podsustavi.

Primjena modela potraživanja

Slika 6-6 prikazuje *model potraživanja* upravljačkih podataka za nadzor pristupa na primjeru sustava *Nadzornik*. Korisnik se autenticira *UP* logici i pokreće zahtjev za stvaranje korisničke sjednice (1). *UP* podsustav stvara sjednicu korisnika, potpisuje identifikacijski podatak sjednice i vraća ga korisniku (2). Identifikacijski podatak sjednice je korisnikov novi identitet u sustavu. U svaki sljedeći zahtjev za korištenje usluge u zaštićenoj domeni, korisnik umeće identitet ostvarene sjednice (3). Primivši zahtjev, *PP* podsustav na osnovi dobivenog identiteta potražuje i dohvaca iz *UP* podsustav upravljačke podatke za nadzor pristupa (4, 5). U upravljačkim podacima je sadržana odluka provedbe pristupa. Na osnovi odluke o dozvoli pristupa iz upravljačkih podataka, *PP* podsustav proslijeđuje zahtjev korisnika usluzi (6). Usluga šalje odgovor *PP* podsustavu (7), a *PP* podsustav proslijeđuje odgovor korisniku (8).



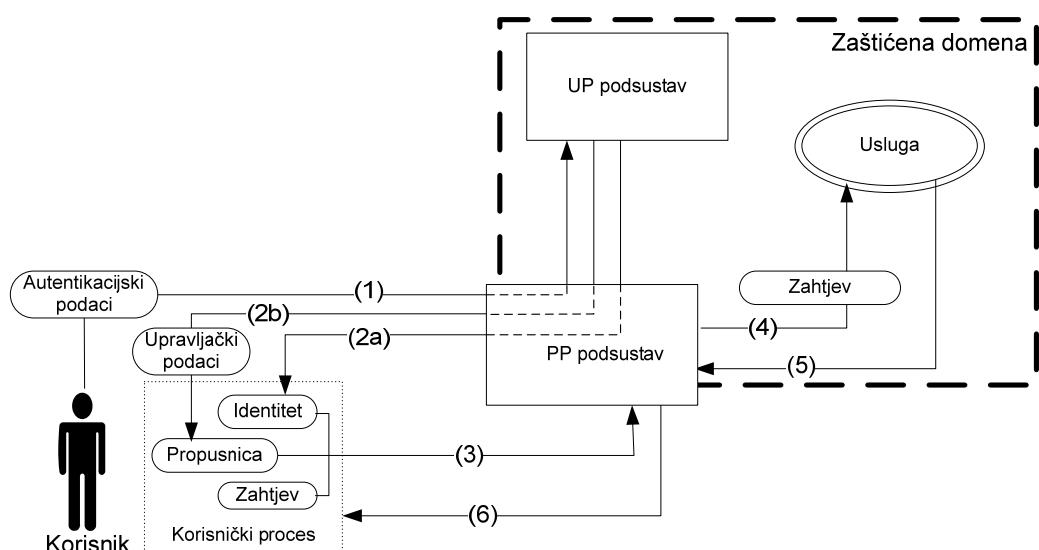
Slika 6-6: Model potraživanja upravljačkih podataka

Model potraživanja uzrokuje učestalu komunikaciju između *UP* i *PP* podsustava. Kako su *UP* i *PP* podsustavi raspodijeljeni na različita računala, učestala komunikacija između njih stvara znatan komunikacijski teret na mreži i narušava radna svojstva i sigurnost sustava *Nadzornik*.

Primjena modela ponude

Na slici 6-7 prikazan je *model ponude* upravljačkih podataka za nadzor pristupa na primjeru sustava *Nadzornik*. Korisnikov početni korak je autentikacija *UP* podsustavu (1). Tijekom autentikacije, *UP* podsustav stvara sjednicu korisnika, potpisuje identifikacijski podatak sjednice i izdaje ga korisniku kao njegov novi identitet u domeni (2a). Osim toga,

UP podsustav u istom koraku stvara korisnikovu propusnicu s upravljačkom podacima za dozvolu pristupa u domeni. Upravljački podaci moraju biti potpisani, zbog toga što njihovo prenošenje izvan zaštićene domene predstavlja sigurnosni rizik. U svaki sljedeći zahtjev za korištenje usluge, upućen putem *PP* podsustava, korisnik umeće svoj identitet u domeni te propusnicu s potpisanim upravljačkim podacima (3). Primivši zahtjev, *PP* podsustav provjerava potpis upravljačkih podataka u propusnici. Na osnovi dobivenog identiteta i dozvola pristupa iz propusnice, *PP* podsustav prosljeđuje korisnikov zahtjev usluzi (4). Usluga šalje odgovor *PP* podsustavu (5), a *PP* podsustav prosljeđuje odgovor korisniku (6).



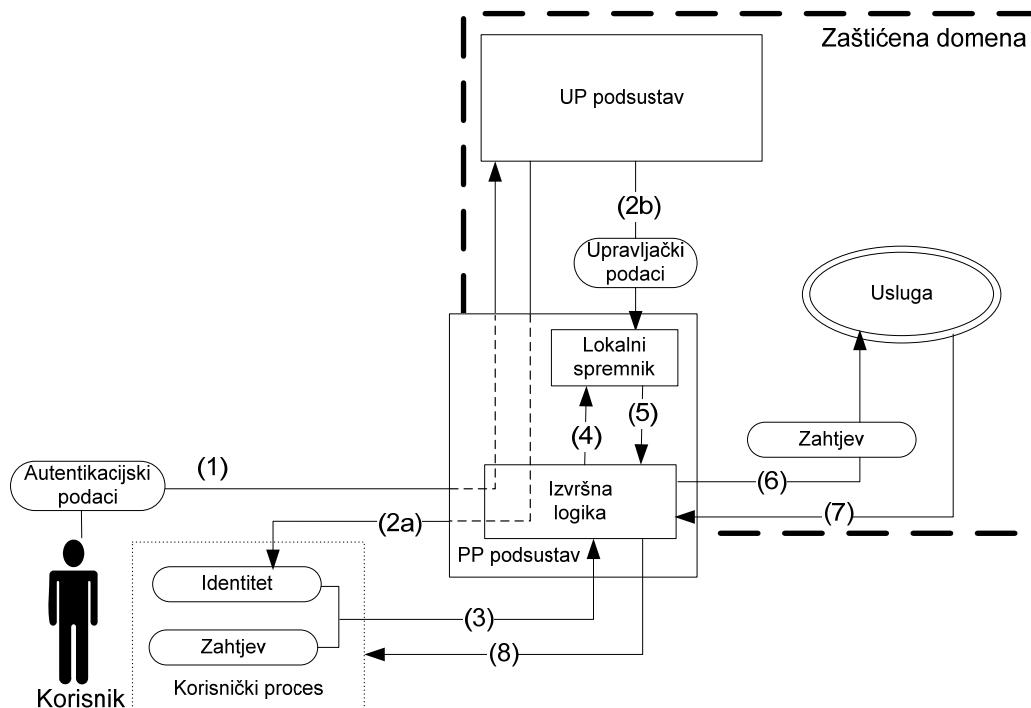
Slika 6-7: Model ponude upravljačkih podataka

Provjera potpisa upravljačkih podataka računalno je zahtjevna operacija koja se u ovom modelu izvodi za svaki primljeni zahtjev. Učestale provjere potpisa veće količine upravljačkih podataka kod *modela ponude* negativno utječu na radna svojstva nadzora pristupa sustava *Nadzornik*.

Primjena mješovitog modela ponude i potraživanja

U okviru magistarskog rada oblikovan je novi *mješoviti model ponude i potraživanja* (*PP* model) prikazan na slici 6-8. Prema mješovitom *PP* modelu pripremanja upravljačkih podataka korisniku se tijekom autentikacije (1) izdaje novi identitet (2a) te se upravljački podaci s pravima pristupa upisuju u lokalni spremnik *PP* podsustava (2b). Kao kod *modela potraživanja*, u zahtjev za korištenje usluge upućen putem *PP* podsustava korisnik umeće svoj identitet u domeni (3). Nakon primjera zahtjeva, izvršna logika *PP* podsustava primjenjuje dobiveni identitet i potražuje upravljačke podatke za nadzor pristupa iz lokalnog spremnika (4, 5). U upravljačkim podacima je sadržana odluka provedbe pristupa, na osnovi

koje se odbija ili odobrava zahtjev korisnika. Odobreni zahtjev korisnika prosljeđuje se usluzi (6). Usluga šalje odgovor *PP* podsustavu (7), a *PP* podsustav prosljeđuje odgovor korisniku (8).



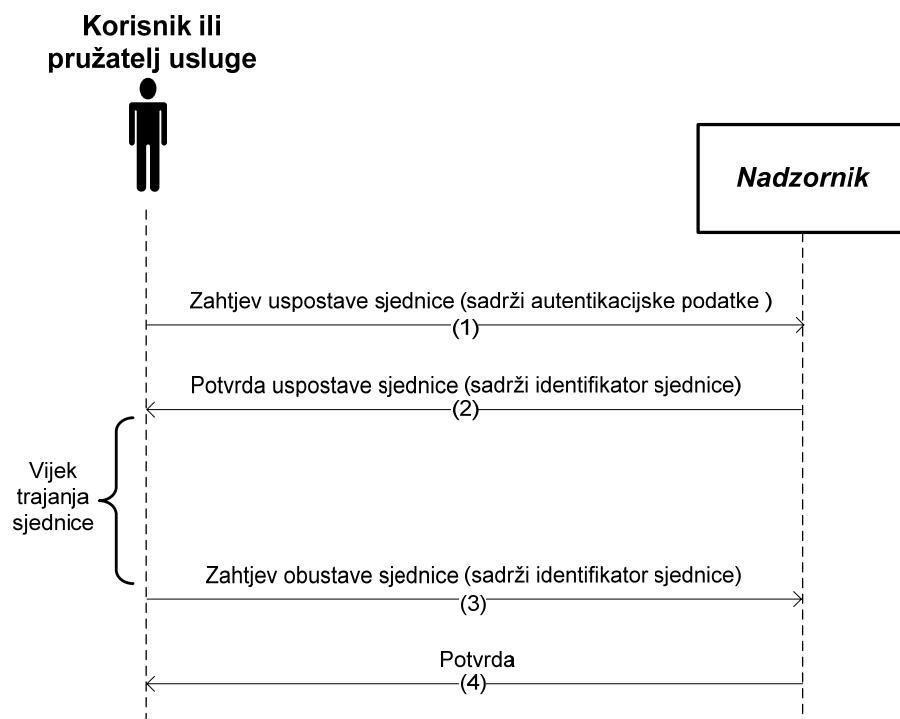
Slika 6-8: Mješoviti model utiskivanja i potraživanja upravljačkih podataka

U mješovitom *PP* modelu odluke o provedbi pristupa dohvaćaju se iz lokalnog spremnika, te stoga ne moraju biti potpisane. Time se izbjegava izvođenje računski zahtjevnog postupka provjere potpisa koji se primjenjuje u *modelu ponude*. Nakon uspostave sjednice korisnika, izvršna logika prima upravljačke podatke lokalno. Za razliku od *modela potraživanja*, izbjegava se dohvaćanje upravljačkih podataka iz *UP* podsustava tijekom obrade pojedinih zahtjeva korisnika. Na taj način *mješoviti PP model* smanjuje mrežno opterećenje *modela potraživanja*, te ostvaruje bolja svojstva nadzora pristupa od oba klasična modela.

6.4.3. Protokol autentikacije

Protokol autentikacije prikazan je slikom 6-9. Sustav *Nadzornik* primjenom protokola autentikacije provjerava identitet korisnika i pružatelja usluga i stvara sjednicu koja omogućava nastavak rada u sustavu.

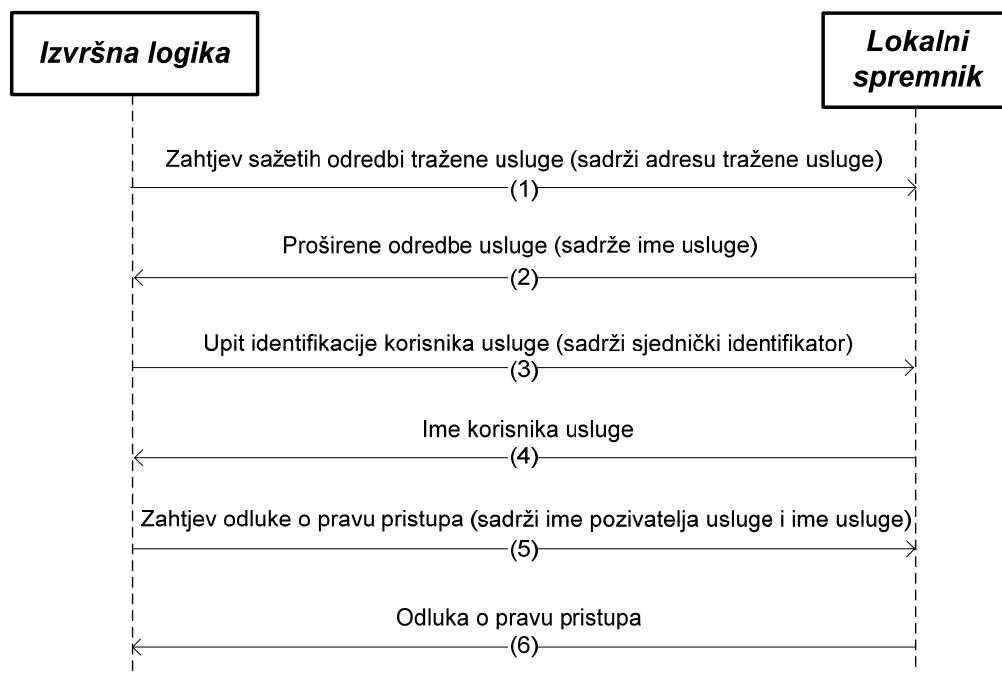
Protokol autentikacije započinje dolaskom subjektova zahtjeva za uspostavom sjednice sa sustavom *Nadzornik* (1). Subjekt protokola autentikacije je korisnik ili pružatelj usluge. Subjektov zahtjev sadrži njegove autentikacijske podatke sastavljene od podatkovnog para ime-zaporka. *Nadzornik* provjerava ispravnost primljenih subjektovih autentikacijskih podataka uspoređujući ih s njegovim registriranim podacima. Subjektu se vraća potvrda uspostave sjednice (2). U slučaju neispravnih autentikacijskih podataka, vraća se negativna potvrda uspostave sjednice. U slučaju ispravnih autentikacijskih podataka, *Nadzornik* priprema subjektove upravljačke podatke za nadzor pristupa te vraća potvrdu uspješno uspostavljene sjednice. Potvrda uspješno uspostavljene sjednice sadrži identifikator uspostavljene sjednice. Uspostavljena sjednica subjekta traje sve dok subjekt ne zatraži obustavu sjednice. Tijekom vijeka trajanja sjednice, subjekt koristi usluge u zaštićenoj domeni. Subjekt traži obustavu sjednice slanjem zahtjeva obustave sjednice u kojem je naveden identifikator sjednice koju treba obustaviti (3). Nadzornik obustavlja sjednicu uništavanjem pripremljenih upravljačkih podataka za nadzor pristupa te vraća potvrdu o uspješnom ili neuspješnom obustavljanju sjednice.



Slika 6-9: Protokol autentikacije u sustavu *Nadzornik*

6.4.4. Protokol ispitivanja upravljačkih podataka

Protokol ispitivanja upravljačkih podataka za nadzor pristupa u sustavu *Nadzornik* prikazan je na slici 6-10. U protokolu sudjeluju izvršna logika i lokalni spremnik *PP* podsustava. Izvršna logika putem ovog protokola ispituje upravljačke podatke i primjenjuje ih za nadzor pristupa korisnika uslugama. Upravljački podaci pripremljeni su u lokalnom spremniku primjenom mješovitog *PP* modela pripremanja upravljačkih podataka.



Slika 6-10: Protokol ispitivanja upravljačkih podataka u sustavu Nadzornik

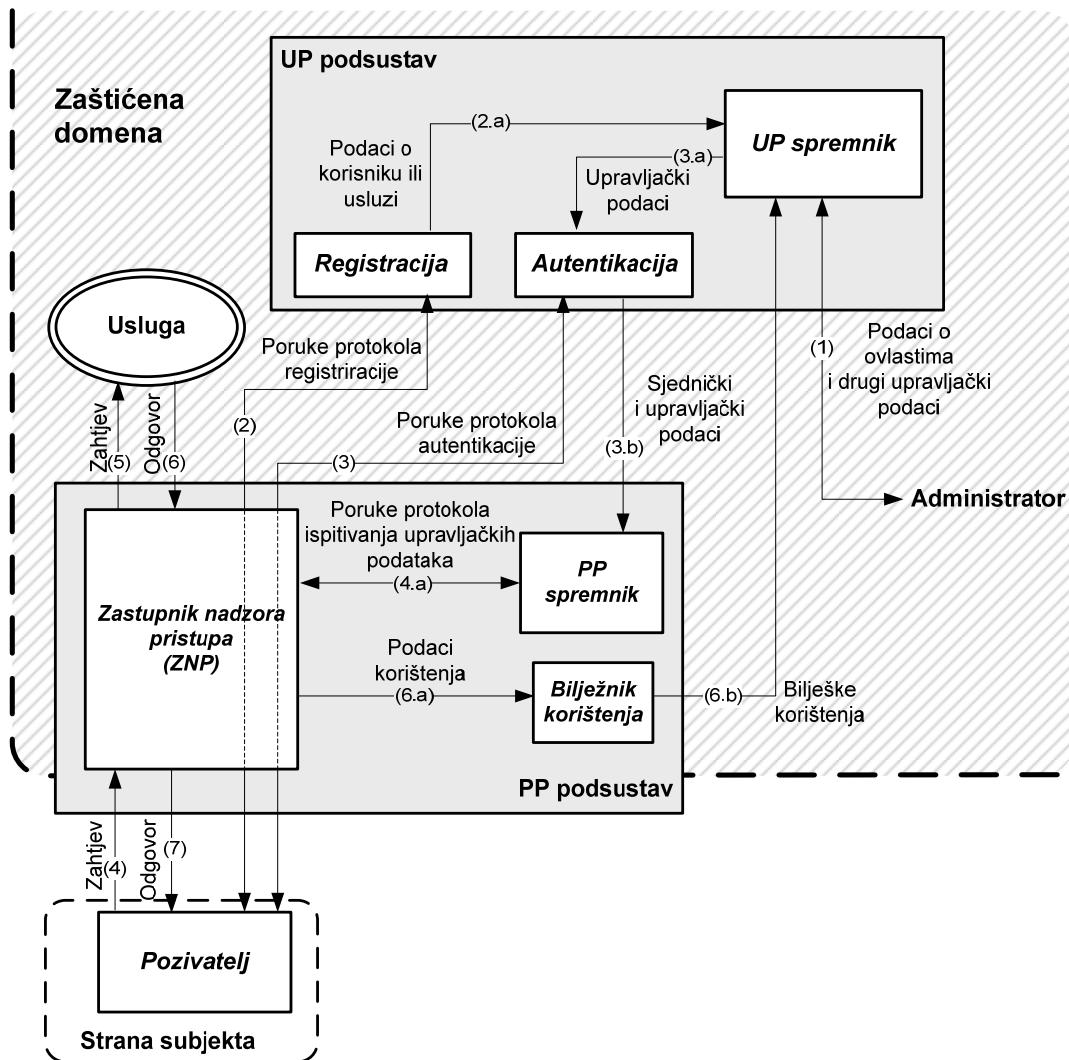
Protokol se izvodi za svaki zahtjev korištenja usluge kojeg subjekt pošalje izvršnoj logici. Izvršna logika koristi se podacima sadržanim u subjektovom zahtjevu korištenja usluge, te od lokalnog spremnika *PP* podsustava zahtijeva proširene odredbe usluge koju traži korisnik (1). U zahtjevu potraživanja proširenih odredbi lokalnom se spremniku šalje adresa usluge koju je zatražio korisnik. Lokalni spremnik na osnovi adrese tražene usluge odgovara slanjem sažetih odredbi tražene usluge (2). Podaci sažetih odredbi usluge sadrže zastavice za provjeru identiteta i autorizacije subjekta, zastavicu za praćenje korištenja usluge, te ime i adresu usluge. Postupak praćenja korištenja usluge nije obuhvaćen ovim protokolom i zasebno se provodi u izvršnoj logici. Dodatno, provjeru autorizacije korisnika nije moguće provesti bez prethodno provjerenog identiteta korisnika. Protokol ispitivanja upravljačkih podataka nastavlja se prema zahtjevima sigurnosti koji su definirani navedenim zastavicama u odredbama usluge.

Ako se u odredbama usluge postavljenom zastavicom za provjeru identiteta zahtjeva *provjera identiteta* subjekta, izvršna logika šalje upit identifikacije subjekta u lokalni spremnik i ispituje subjektov identitet (3). Upit sadrži identifikator sjednice kojim se subjekt služi u sustavu. Lokalni spremnik odgovara podatkom o imenu subjekta koji se u sustavu služi zadanom sjednicom (4). Ako se postavljenom zastavicom za provjeru autorizacije zahtjeva *provjera autorizacije* subjekta, izvršna logika šalje zahtjev dohvata odluke o pravu pristupa subjekta traženoj usluzi (5). Izvršna logika umeće u zahtjev ime subjekta dobiveno postupkom identifikacije te ime usluge dobiveno u sažetim odredbama usluge. Lokalni spremnik vraća odluku o pravu pristupa koja ili odobrava ili spriječava pristup subjekta traženoj usluzi (6).

6.5. Arhitektura sustava Nadzornik

Organizacija arhitekture sustava *Nadzornik* izvedena je na osnovi podjele funkcionalnosti iz odjeljka 6.3, te na osnovi *mješovitog PP modela* opisanog u odjeljku 6.4.2. Globalna arhitektura sustava *Nadzornik* prikazana je slikom 6-11. Arhitektura sustava organizirana je u tri glavne cjeline: *UP* podsustav, *PP* podsustav i subjekt. *UP* podsustav namijenjen je za dogovaranje pristupa u zaštićenu domenu. *PP* podsustav namijenjen je za provedbu nadzora pristupa uslugama zaštićene domene. Subjekt označava korisnika ili uslugu koji primjenom odgovarajuće potpore šalju zahtjeve korištenja usluga sustavu *Nadzornik*. *UP* i *PP* podsustavi sastoje se od modula čije se arhitekture zasebno opisuju u nastavku ovog odjeljka. U odjeljku se opisuje i modul *Pozivatelj* smješten lokalno na strani subjekta. Komunikacija između pojedinih modula sustava ostvaruje se protokolima autentikacije, registracije i ispitivanja upravljačkih podataka. Dodatno, podsustavi *PP* i *UP* ostvaruju nadzor pristupa mješovitim *PP* modelom suradnje.

UP podsustav sastoji se od modula *UP spremnika*, *Registracije* i *Autentikacije*. *UP spremnik* je osnovni spremnik upravljačkih podataka sustava *Nadzornik*. U njega se spremaju svi upravljački podaci potrebni za nadzor pristupa. Unutar spremnika dodatno se spremaju podaci bilješki korištenja usluga. Upravljačke podatke u *UP spremnik* unose administrator, korisnici i pružatelji usluga. Administrator izravno upisuje podatke o ovlastima i druge upravljačke podatke u *UP spremnik* (1). Korisnici i pružatelji usluga registriraju upravljačke podatke primjenom modula *Registracije* (2). *Registracija* je modul zakonodavne logike koji je javno dostupan posredovanjem *PP* podsustava. Registriranje podataka o korisniku i usluzi ostvaruje se protokolom registracije. Primjenom protokola registracije, modul *Registracije* prikuplja podatke o korisniku i usluzi, te sprema prikupljene podatke u *UP spremnik* (2.a).



Slika 6-11: Globalna arhitektura sustava Nadzornik

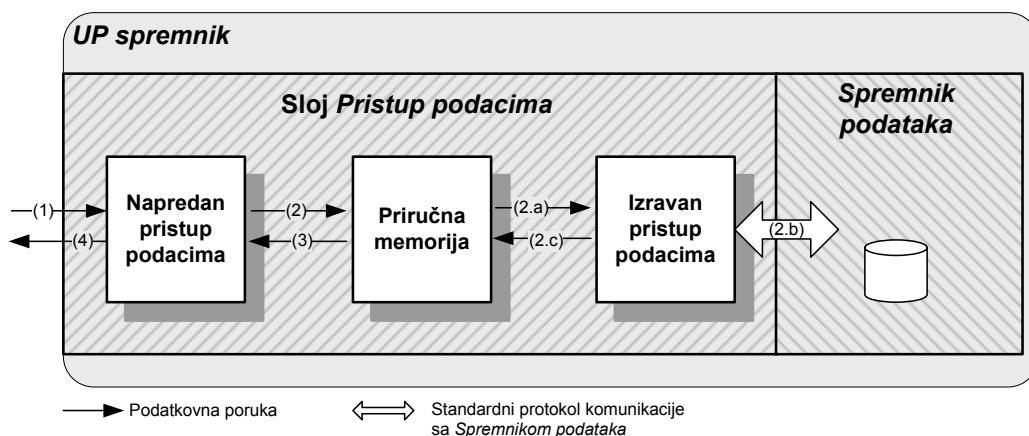
Autentikacija je modul upravne logike *UP* podsustava, javno dostupan korisnicima posredovanjem *PP* podsustava. Uloga modula *Autentikacije* je razmjena upravljačkih podataka koji su registrirani u *UP* podsustavu, te pripremanje podataka koje *PP* podsustav primjenjuje za nadzor pristupa. Primjenom modula *Autentikacije* subjekt uspostavlja ili obustavlja sjednicu sa sustavom *Nadzornik* (3). Uspostava i obustava sjednice izvode se protokolom autentikacije. *Autentikacija* koristi upravljačke podatke iz *UP spremnika* (3.a), te u *PP spremnik* zapisuje sjedničke i upravljačke podatke o uspostavljenim sjednicama. Zapisivanje upravljačkih i sjedničkih podataka u *PP spremnik* predstavlja prvi dio pripremanja upravljačkih podataka nadzora pristupa mješovitim *PP* modelom. Razmjena identifikatora uspostavljene sjednice s *Pozivateljem* predstavlja drugi dio pripremanja upravljačkih podataka. *Pozivatelj* lokalno, na strani subjekta, sprema primljeni identifikator uspostavljene sjednice i umeće ga u sve zahtjeve koje subjekt šalje tijekom trajanja sjednice.

PP podsustav sastoji se od modula *Bilježnik korištenja*, *PP spremnik* i *Zastupnik nadzora pristupa* (ZNP). *Bilježnik korištenja* je modul koji na osnovi podataka korištenja usluga stvara bilješke korištenja i zapisuje ih u *UP spremnik*. *PP spremnik* je lokalni spremnik sjedničkih i upravljačkih podataka *PP* podsustava. Zbog razmjene upravljačkih podataka s *UP* podsustavom primjenom mješovitog *PP* modela, *PP spremnik* je nužni dio *PP* podsustava. U *PP spremnik* spremaju se upravljački podaci o uslugama i korisnicima koji su uspostavili sjednicu. ZPN modul ostvaruje izvršnu logiku *PP* podsustava. On prima zahtjeve od *Pozivatelja* (4), provjerava ispravnost zahtjeva protokolom ispitivanja upravljačkih podataka (4.a) te prosljeđuje zahtjev usluzi ako je zahtjev ispravan (5). ZNP modul zatim prima odgovor od usluge (6), bilježi korištenje usluge (6.a, 6.b) i vraća odgovor usluge *Pozivatelju* (7).

6.5.1. UP spremnik

UP spremnik sprema upravljačke podatke o svim uslugama i korisnicima sustava *Nadzornik*. *UP spremnik* omogućava modulu *Registracije* upisivanje upravljačkih podataka o korisnicima i uslugama. Modul *Autentikacije* dohvata upravljačke podatke iz *UP spremnika*. Administrator sustava *Nadzornik* ima izravan pristup upravljačkim podacima u *UP spremniku*. Dodatno, *Bilježnik korištenja* sprema bilješke korištenja u *UP spremnik*.

Arhitektura *UP spremnika* prikazana je na slici 6-12. *UP spremnik* sastoji se od sloja *Spremnik podataka* i *Pristup podacima*. *Spremnik podataka* ostvaren je bazom podataka. U sloju *Pristup podacima* nalaze se podmoduli za pristupanje podacima u *Spremniku podataka*. Postoje tri podmodula za pristup *Spremniku podataka*: *Napredan pristup podacima*, *Priručna memorija* i *Izravan pristup podacima*. Podmodulom *Izravan pristup podacima* ostvaruju se operacije brisanja, izmjene i upisa nad podacima u *Spremniku podataka*.



Slika 6-12: Arhitektura modula *UP spremnik*

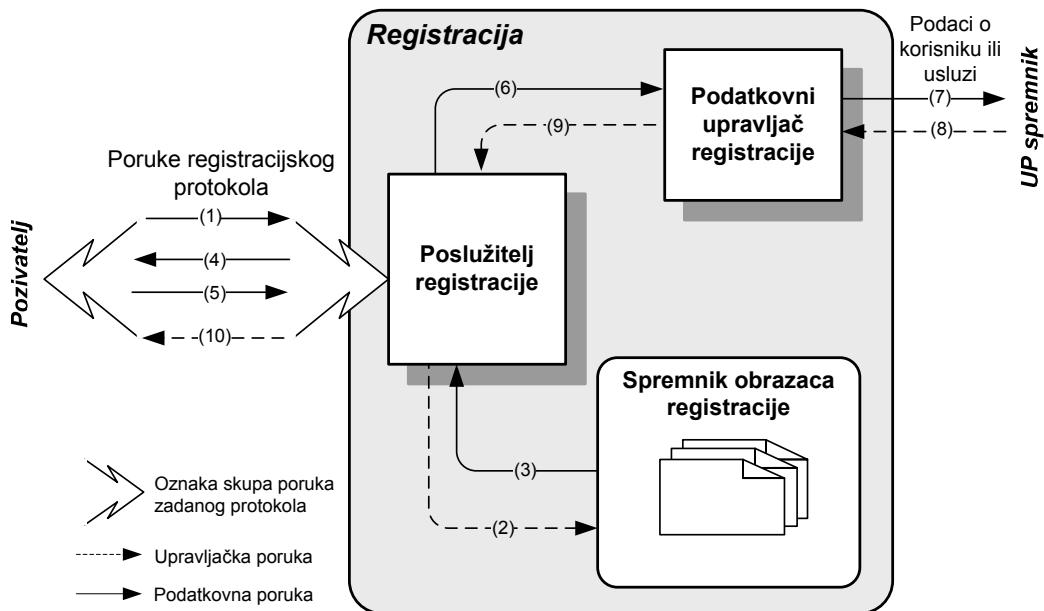
Podmodul *Priručna memorija* omogućuje brz pristup do podataka privremenim spremanjem već korištenih podataka iz *Spremnika podataka*. Podmodul *Napredni pristup podacima* pojednostavljuje uporabu *Priručne memorije*. Detalji arhitekture *UP spremnika* opisani su u [43].

Uporaba *UP spremnika* započinje zahtjevom operacije upisa, dohvata, izmjene ili brisanja podataka (1). *Napredan pristup podacima* prima zahtjev i prosljeđuje ga *Priručnoj memoriji* (2). Ako *Priručna memorija* sadrži odgovor na zahtjev, onda se posredstvom *Naprednog pristupa podacima* odgovara na zahtjev (3, 4). U protivnom, primjenjuje se *Izravan pristup podacima* (2.a). *Izravan pristup podacima* komunicira sa *Spremnikom podataka* protokolom koji je ovisan o ostvarenju *Spremnika podataka* (2.b). *Izravan pristup podacima* izvodi operacije nad podacima u *Spremniku podataka* i stvara odgovor koji šalje *Priručnoj memoriji* (2.c). Priručna memorija privremeno sprema podatak o odgovoru na postavljeni zahtjev i vraća odgovor (3, 4).

6.5.2. Registracija

Registracija je modul koji protokolom registracije prikuplja podatke o korisniku ili usluzi, te prikupljene podatke zapisuje u *UP spremnik*. Modul *Registracija* sastoji se od tri podmodula: *Poslužitelj registracije*, *Spremnik obrazaca registracije* i *Podatkovni upravljač registracije*. *Poslužitelj registracije* prima, obrađuje i poslužuje poruke protokola registracije. On tijekom obrade koristi ostala dva podmodula *Registracije*. *Spremnik obrazaca registracije* sadrži obrasce koji se tijekom protokola registracije primjenjuju za prikupljanje podataka o korisniku ili usluzi. *Podatkovni upravljač registracije* omogućuje jednostavni upis prikupljenih podataka o korisniku ili usluzi u *UP spremnik*. Arhitektura modula *Registracija* prikazana je na slici 6-13.

Protokoli registracije korisnika i usluge, objašnjeni u odjeljku 6.4.1, razlikuju se po broju obrazaca koje se popunjavanja tijekom registracije. Na slici 6-13 prikazan je protokol registracije korisnika, koji se izvodi popunjavanjem jednog obrasca. Zahtjev registracije je početna poruka protokola registracije korisnika (1). Zahtjev obrađuje *Poslužitelj registracije*. *Poslužitelj registracije* dohvaća potrebni obrazac registracije korisnika iz *Spremnika obrazaca registracije* (2, 3). Potom šalje obrazac korisniku na popunjavanje (4), te od korisnika dočekuje ispunjeni obrazac (5). Nakon primitka ispunjenog obrasca, *Poslužitelj registracije* izdvaja podatke popunjene u obrascu.



Slika 6-13: Arhitektura modula *Registracija*

Protokol registracije usluge izvodi se popunjavanjem dvaju obrazaca. Popunjavanje prvog obrasca za registraciju usluge provodi se slično koracima (1 – 5). Popunjavanje drugog obrasca izvodi se ponavljanjem koraka (2 – 5), ali se iz *Spremnika obrazaca registracije* dohvaća drugi obrazac za registraciju usluge. Dohvaćeni obrazac šalje se na popunjavanje pružatelju usluge, koji popunjava taj obrazac i šalje ga *Poslužitelju registracije*. *Poslužitelj registracije* izdvaja podatke popunjene u oba obrasca, te započinje obradu prikupljenih podataka.

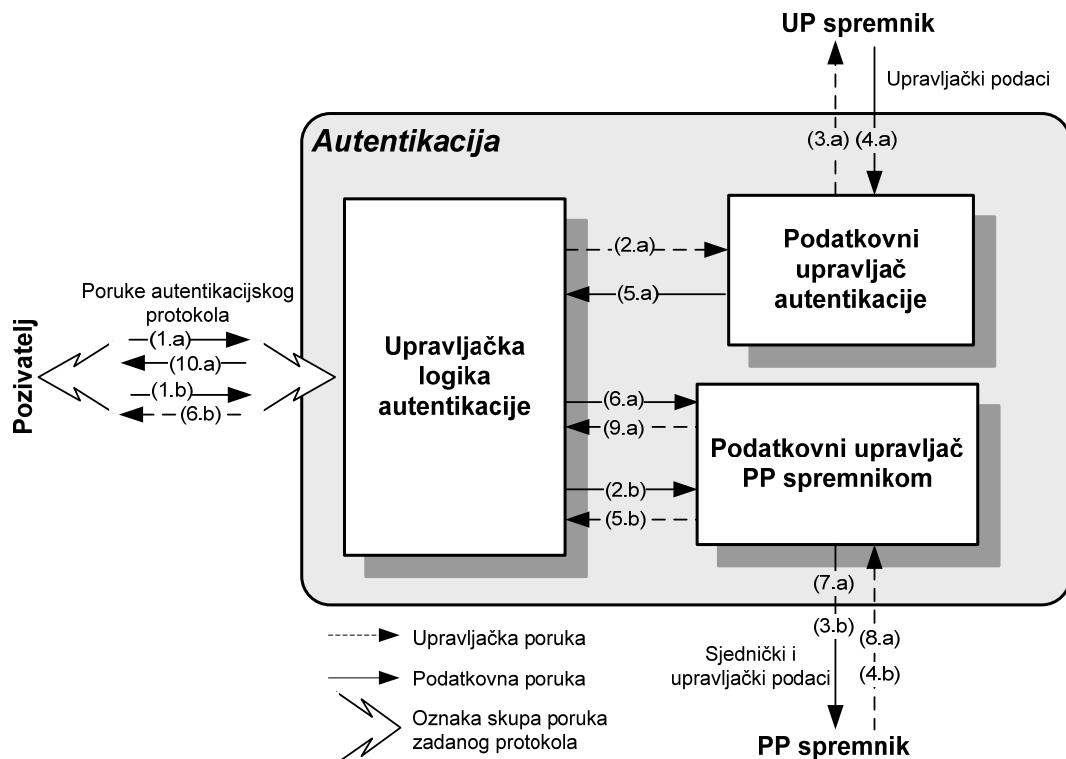
Nakon što je završeno popunjavanje registracijskih obrazaca, *Poslužitelj registracije* šalje sve prikupljene podatke u *Podatkovni upravljač registracije* (6). *Podatkovni upravljač* zapisuje podatke u *UP spremnik* (7, 8) te vraća u *Poslužitelj registracije* potvrdu o uspješnosti zapisivanja podataka (9). *Poslužitelj registracije* potom vraća *Pozivatelju* potvrdu uspješnosti cijelokupnog procesa registracije (10).

6.5.3. Autentikacija

Autentikacija je modul koji omogućava uspostavljanje i obustavljanje sjednica sa sustavom *Nadzornik*. Dodatno, modul *Autentikacije* priprema upravljačke podatke za nadzor pristupa subjekta tijekom trajanja sjednice. Protokol autentikacije opisan je u odjeljku 6.4.3, a pripremanje upravljačkih podataka mješovitim *PP* modelom opisano je u odjeljku 6.4.2. Arhitektura modula *Autentikacije* prikazana je na slici 6-14.

Modul *Autentikacije* sastoji se od podmodula *Upravljačka logika autentikacije*, *Podatkovni upravljač autentikacije* i *Podatkovni upravljač PP spremnikom*. *Upravljačka*

logika autentikacije ostvaruje komunikaciju s modulom *Pozivatelj* na strani subjekta, stvara sjednicu i pomoću ostala dva podmodula obavlja podatkovne operacije. *Podatkovni upravljač registracije* omogućuje dohvata podataka o korisniku ili usluzi iz *UP spremnika*. *Podatkovni upravljač PP spremnikom* omogućuje izmjene sjedničkih i upravljačkih podataka u *PP spremnik*.



Slika 6-14: Arhitektura modula *Autentikacija*

Na slici 6-14 razlikuju se scenariji uspostave sjednice i obustave sjednice. Slovom "a" uz redni broj poruke na strelicama označuju se poruke u scenariju uspostave sjednice. Na sličan način, slovom "b" uz redni broj poruke na strelicama označuju se poruke u scenariju obustave sjednice.

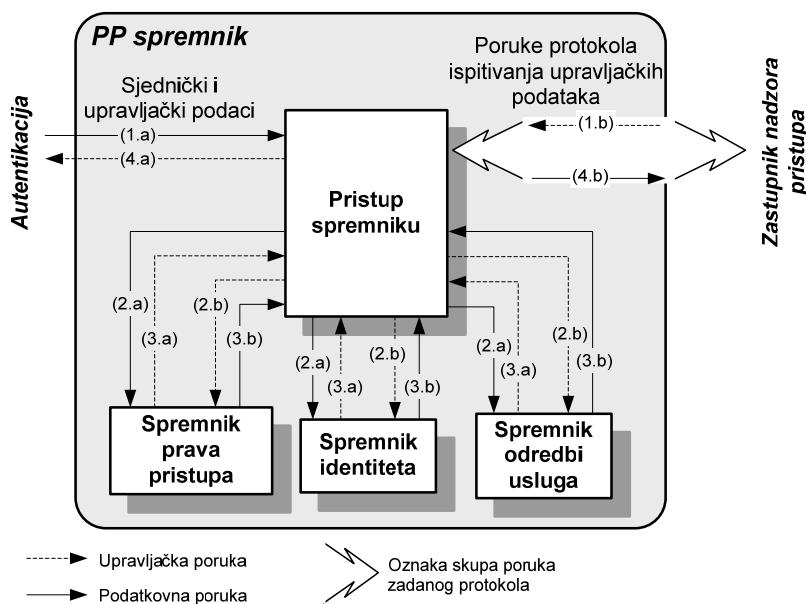
Pozivatelj započinje scenarij uspostave sjednice slanjem zahtjeva za uspostavu sjednice (1.a). Zahtjev sadrži autentikacijske podatke subjekta. *Upravljačka logika autentikacije* prima zahtjev i traži od *Podatkovnog upravljača autentikacije* upravljačke podatke subjekta koji je postavio zahtjev (2.a). *Podatkovni upravljač autentikacije* dohvaća tražene upravljačke podatke iz *UP spremnika* (3.a, 4.a) te ih vraća *Upravljačkoj logici autentikacije* (5.a). *Upravljačka logika autentikacije* provjerava autentičnost subjekta usporedbom dohvaćenih upravljačkih podataka i autentikacijskih podataka iz zahtjeva za uspostavu sjednice. Ako je subjekt uspješno autenticiran, *Upravljačka logika autentikacije*

stvara identifikator uspostavljene sjednice, te priprema upravljačke podatke za nadzor pristupa subjekta. Pripremanje upravljačkih podataka izvodi se upisivanjem identifikatora sjednice i upravljačkih podataka u *PP spremnik* primjenom *Podatkovnog upravljača PP spremnikom* (6.a, 7.a, 8.a, 9.a). Postupak uspostave sjednice završava slanjem potvrde o uspješnoj ili neuspješnoj uspostavi sjednice (10.a). Potvrda uspješne sjednice sadrži identifikator uspostavljene sjednice.

U scenariju obustave sjednice, *Pozivatelj* šalje zahtjev za obustavom sjednice (1.b). Zahtjev obustave sjednice sadrži identifikator sjednice koju treba obustaviti. *Upravljačka logika autentikacije* prima zahtjev i pomoću *Podatkovnog upravljača PP spremnikom* uklanja iz *PP spremnika* sjedničke i upravljačke podatke pripremljene tijekom postupka uspostave sjednice (2.b, 3.b, 4.b, 5.b). Obustava sjednice završava slanjem potvrde o uspješno ili neuspješno obustavljenoj sjednici (6.b).

6.5.4. *PP spremnik*

U *PP spremnik* spremaju se upravljački podaci o uslugama i korisnicima koji su uspostavili sjednicu te njihovi pripadni sjednički podaci. *PP spremnik* omogućuje modulu *Autentikacije* upisivanje i brisanje sjedničkih i upravljačkih podataka nastalih tijekom uspostave sjednice. Nadalje, *Zastupniku nadzora pristupa* (ZNP modul) omogućuje se ispitivanje upravljačkih podataka pripremljenih u *PP spremniku*.



Slika 6-15: Arhitektura modula *PP spremnik*

Arhitektura *PP spremnika* prikazana je na slici 6-15. *PP spremnik* sastoji se od podmodula *Pristup spremniku* i tri spremnika: *Spremnika prava pristupa*, *Spremnika*

identiteta i Spremnika odredbi usluga. Pristup spremniku prima i obrađuje zahtjeve pristigle od modula *Autentikacije* i *ZNP* modula te upravlja operacijama nad spremnicima. Svaki od spremnika sprema jedan podskup upravljačkih podataka sustava *Nadzornik* koji su detaljno opisani u odjeljku 6.2. *Spremnik odredbi usluga* sprema sažete odredbe usluga. Podaci sažetih odredbi usluge sadrže zastavice za provjeru pristupa i praćenje korištenja, te ime i adresu usluge. *Spremnik identiteta* sprema svojstva identiteta korisnika i usluga proširena identifikatorom njihovih uspostavljenih sjednica. *Spremnik prava pristupa* sprema podatke o pravima pristupa.

Na slici 6-15 razlikuju se scenariji unošenja izmjena u *PP spremnik* i ispitivanja *PP spremnika*. Slovom "a" uz redni broj poruke na strelicama označuju se poruke scenarija unošenja izmjena u *PP spremnik*. Na sličan način, slovom "b" uz redni broj poruke na strelicama označuju se poruke scenarija ispitivanja *PP spremnika*.

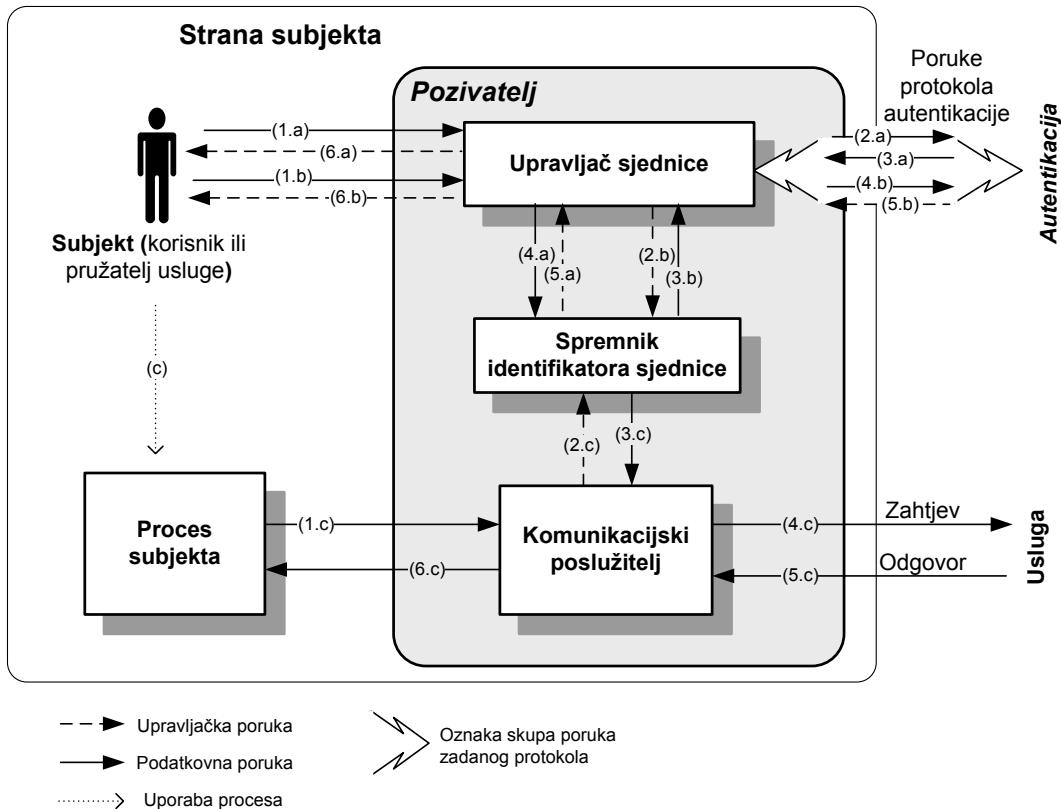
U scenariju unošenja izmjena *Autentikacija* šalje *PP spremniku* zahtjev za upisivanjem ili brisanjem sjedničkih i upravljačkih podataka. Podmodul *Pristup spremniku* prima zahtjev, te upisuje ili briše podatke u odgovarajućim spremnicima *PP spremnika* (2.a, 3.a). Scenarij unošenja izmjena završava vraćanjem potvrde o uspješnosti zatražene izmjene (4.a).

U scenariju ispitivanja *ZNP* modul šalje upit *PP spremniku* (1.b). Upiti *ZNP* modula definirani su protokolom ispitivanja upravljačkih podataka. Podmodul *Pristup spremniku* odgovara na upit dohvaćanjem podataka iz odgovarajućeg spremnika (2.b, 3.b, 4.b).

6.5.5. Pozivatelj

Pozivatelj je modul smješten lokalno na strani subjekta koji ostvaruje komunikaciju subjekta sa sustavom *Nadzornik*. *Pozivatelj* uspostavlja i obustavlja sjednicu subjekta sa sustavom *Nadzornik*. Dodatno, *Pozivatelj* tijekom trajanja uspostavljenje sjednice priprema subjektove zahtjeve u odgovarajući oblik koji se šalje sustavu *Nadzornik*. Arhitektura modula *Pozivatelj* prikazana je na slici 6-16.

Pozivatelj se sastoji od tri podmodula: *Upravljač sjednice*, *Spremnik identifikatora sjednice* i *Komunikacijski poslužitelj*. *Upravljač sjednice* komunicira s modulom *Autentikacije* primjenom protokola autentikacije, te uspostavlja ili obustavlja sjednicu. *Spremnik identifikatora sjednice* služi *Upravljaču sjednice* za spremanje identifikatora uspostavljenje sjednice. *Komunikacijski poslužitelj* presreće zahtjeve procesa subjekta, te u njih umeće identifikator sjednice iz *Spremnika identifikatora sjednice*.



Slika 6-16: Arhitektura modula *Pozivatelj*

Pozivatelj sudjeluje u komunikacijskom scenariju uspostave sjednice, korištenja usluge i obustave sjednice. Komunikacija podmodula u scenariju uspostave sjednice označuje se rednim brojem poruke i slovom "a". Na sličan se način komunikacija podmodula u scenariju obustave sjednice označuje rednim brojem poruke i slovom "b". Poruke scenarija korištenja usluge označene su slovom "c".

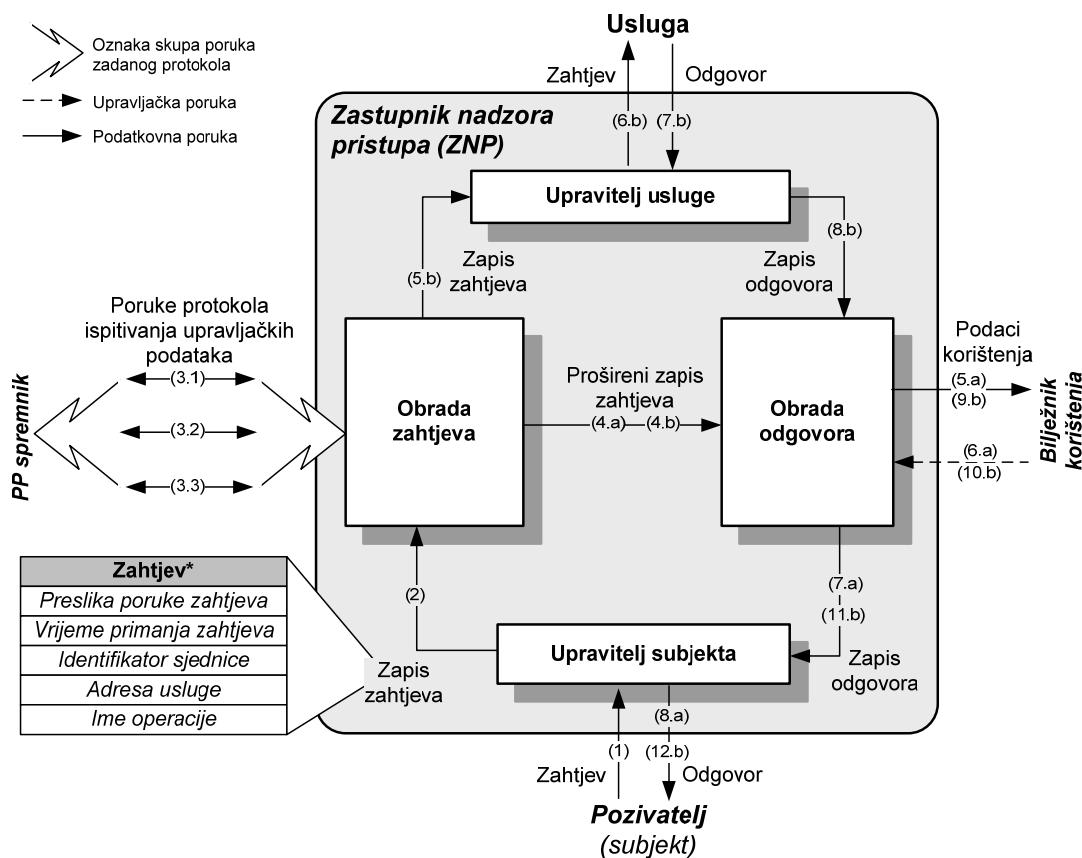
Tijekom uspostave sjednice, subjekt šalje ime korisnika ili usluge i zaporku u *Upravljač sjednice* (1.a). *Upravljač sjednice* šalje zahtjev uspostave sjednice prema modulu *Autentikacije* (2.a) i prima potvrdu o uspostavi sjednice (3.a). Iz potvrde o uspješno uspostavljenoj sjednici izdvaja identifikator sjednice i zapisuje ga u *Spremnik identifikatora sjednice* (4.a, 5.a). Zatim obavještava subjekta o uspostavljenoj sjednici (6.a).

U scenariju obustave sjednice, subjekt obustavlja uspostavljenu sjednicu slanjem zahtjeva za obustavom sjednice u *Upravljač sjednice* (1.b). *Upravljač sjednice* dohvaca identifikator uspostavljene sjednice iz *Spremnika identifikatora sjednice* (2.b, 3.b) i umeće ga u zahtjev obustave sjednice koji šalje modulu *Autentikacije* (4.b). Prima potvrdu o uspješnosti obustavljene sjednice (5.b) te obavještava korisnika ili pružatelja usluge o obustavljenoj sjednici (6.b).

Tijekom trajanja uspostavljene sjednice, subjekt se koristi radnim procesom (c) koji šalje zahtjev korištenja usluge iz zaštićene domene sustava Nadzornik (1.c). Komunikacijski poslužitelj presreće stvoreni zahtjev i u poruku zahtjeva umeće identifikator uspostavljene sjednice dohvaćen iz *Spremnika identifikatora sjednice* (2.c, 3.c). Tako izmijenjena poruka zahtjeva šalje se *Nadzorniku* (4.c). Prima se odgovor na poslani zahtjev (5.c) te se odgovor prosljeđuje do radnog procesa subjekta (6.c).

6.5.6. Zastupnik nadzora pristupa

Zastupnik nadzora pristupa (ZNP) je modul koji ostvaruje mehanizme nadzora pristupa za zaštitu domene sustava *Nadzornik*. ZNP modul izvodi nadzor pristupa odbacivanjem neovlaštenih zahtjeva korištenja usluga, proslijedivanjem ovlaštenih zahtjeva korištenja usluga i bilježenjem podataka o korištenju usluga. Izvršavanje nadzora pristupa koordinira se pomoću upravljačkih podataka pripremljenih u *PP spremniku*, te identifikatora sjednice subjekta koji šalje zahtjeve korištenja usluge. Bilježenje podataka o korištenju



Slika 6-17: Arhitektura modula *Zastupnik nadzora pristupa*

usluga izvodi se primjenom modula *Bilježnik korištenja*.

Arhitektura *ZNP* modula prikazana je na slici 6-17. *ZNP* modul sastoji se od podmodula *Upravitelj subjekta*, *Upravitelj usluge*, *Obrada zahtjeva* i *Obrada odgovora*. *Upravitelj subjekta* ostvaruje komunikacijski kanal za razmjenu poruka sa subjektom. U njemu se poruke zahtjeva parsiraju, tumače i pretvaraju u strukturu *Zapis zahtjeva*. Poruke odgovora stvaraju se na osnovi strukture *Zapis odgovora*. Strukture *Zapis zahtjeva* i *Zapis odgovora* primjenjuju se tijekom obrade zahtjeva i odgovora u *ZNP* modulu. *Obrada zahtjeva* je podmodul koji pomoću strukture *Zapis zahtjeva* i protokola ispitivanja upravljačkih podataka provjerava primljeni zahtjev te odlučuje o proslijedivanju zahtjeva do usluge. Komunikaciju s uslugom ostvaruje *Upravitelj usluge*. Ovaj podmodul uspostavlja komunikacijski kanal s uslugom putem kojeg usluzi šalje obrađeni zahtjev i od nje prima odgovor. Parsiranjem odgovora usluge, *Upravitelj usluge* stvara strukturu *Zapis odgovora*. Podmodul *Obrada odgovora* objedinjuje podatke iz strukture *Zapis zahtjeva* i *Zapis odgovora* u podatke korištenja. Nadalje, podmodul *Obrada odgovora* bilježi podatke korištenja usluge primjenom *Bilježnika korištenja*, te omogućuje *Upravitelju subjekta* proslijedivanje odgovora usluge prema subjektu.

Subjekt upućuje zahtjev *Upravitelju subjekta* te započinje izvođenje nadzora pristupa u *ZNP* modulu (1). *Upravitelj subjekta* parsira subjektov zahtjev s ciljem stvaranja strukturiranih podataka o zahtjevu. Strukturirani podaci *Zapis zahtjeva* obuhvaćaju presliku poruke zahtjeva, vrijeme kada je poruka zahtjeva primljena, *identifikator sjednice* pod kojom je *Pozivatelj* stvorio zahtjev, *adresu usluge* i *ime operacije* usluge koju se poziva. Struktura *Zapis zahtjeva* predaje se podmodulu *Obrada zahtjeva* (2). Na osnovi te strukture podmodul *Obrada zahtjeva* započinje postupak provjere zahtjeva protokolom ispitivanja upravljačkih podataka u *PP spremniku*.

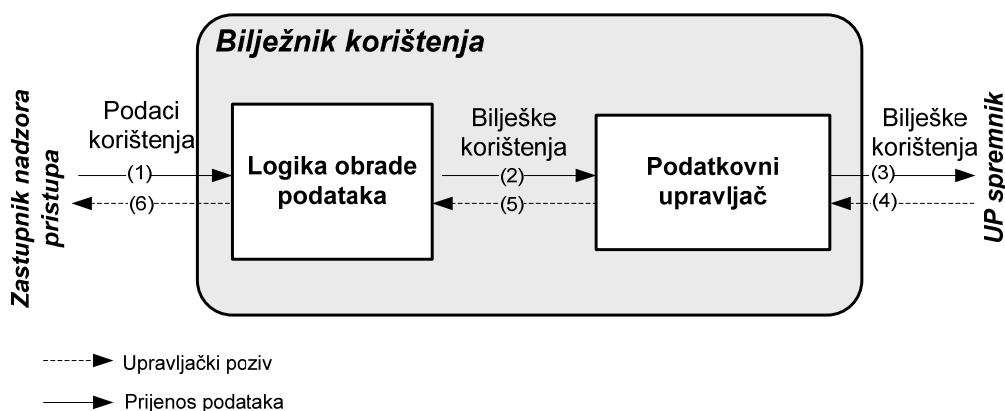
Postupak provjere zahtjeva izvodi se protokolom ispitivanja upravljačkih podataka u tri koraka označena oznakama (3.1 – 3.3). Protokol ispitivanja upravljačkih podataka opisan je u odjeljku 6.4.4. Tijekom **prvog koraka** provjere zahtjeva (3.1), podmodul *Obrada zahtjeva* doznaće *ime usluge* i *odredbe usluge* definirane za *adresu usluge* zadane u *Zahtjevu**. *Odredbe usluge* definiraju potrebu za provjerom identiteta i autorizacije korisnika. Ako je potrebno provjeriti subjektov identitet, podmodul *Obrada zahtjeva* izvodi drugi korak provjere. U **drugom koraku** provjere dohvaća se *ime subjekta* na osnovi *identiteta sjednice* upisanog u *Zapis zahtjeva* (3.2). Ako se odredbama usluge dodatno zahtijeva autorizacija identificiranog subjekta, podmodul *Obrada zahtjeva* izvodi treći korak provjere. **Treći korak** provjere izvodi se na osnovi *imena subjekta* dobivenog u drugom

koraku provjere, *imena usluge* dohvaćenog u prvom koraku provjere i *imena operacije usluge* (3.3). Rezultat ispitivanja je odluka o pravu pristupa koja odobrava ili ne odobrava pristup subjekta usluzi. U ovisnosti o donesenoj odluci, u nastavku nadzora pristupa razlikuje se scenarij neodobrenog zahtjeva i scenarij odobrenog zahtjeva. Slovom "a" uz redni broj poruke na strelicama označuje se scenarij neodobrenog zahtjeva. Slovom "b" uz redni broj poruke na strelicama označuje se scenarij odobrenog zahtjeva.

U scenariju ***neodobrenog zahtjeva***, podmodul *Obrada zahtjeva* šalje u podmodul *Obrada odgovora* strukturu *Zapis zahtjeva* proširenu podatkom o neodobravanju pristupa. Podmodul *Obrada odgovora* primjenom *Bilježnika korištenja* bilježi podatke o neodobrenom zahtjevu korištenja usluge (5.a, 6.a). Dodatno, *Obrada odgovora* stvara u strukturi *Zapis odgovora* poruku odgovora na neodobreni zahtjev korištenja usluge te pomoću *Upravitelja subjekta* šalje odgovor subjektu (7.a, 8.a). U scenariju ***odobrenog zahtjeva***, podmodul *Obrada zahtjeva* šalje u *Obradu odgovora* strukturu *Zapis zahtjeva* proširenu podatkom o odobravanju zahtjeva i vremenskom trenutku prosljedivanja zahtjeva usluzi (4.b). Potom se struktura *Zapis zahtjeva* prosljeđuje *Upravitelju usluge* (5.b) te se zahtjev šalje usluzi (6.b). Usluga odgovara na zahtjev (7.b), odgovor se parsira i pretvara u strukturu *Zapis odgovora* te šalje podmodulu *Obrada odgovora* (8.b). Struktura *Zapis odgovora* slična je strukturi *Zapis zahtjeva*, s razlikom što sadrži presliku odgovora i vrijeme primanja odgovora od usluge. U podmodulu *Obrada odgovora* objedinjuju se *Zapis odgovora* i *Zapis zahtjeva* u podatke korištenja usluge, te ih bilježi *Bilježnik korištenja* (9.b, 10.b). Zatim se putem *Upravitelja subjekta* prosljeđuje odgovor subjektu (11.b, 12.b).

6.5.7. Bilježnik korištenja

Bilježnik korištenja ostvaruje praćenje korištenja usluga stvaranjem bilješki



Slika 6-18: Arhitektura modula *Bilježnik korištenja*

korištenja. Bilješke korištenja stvara na osnovi podataka korištenja koje šalje *ZNP* modul. Bilješke korištenja spremaju se u *UP spremnik*.

Arhitektura *Bilježnika korištenja* prikazana je na slici 6-18. *Bilježnik korištenja* sastoji se od *Logike obrade podataka* i *Podatkovnog upravljača*. *Logika obrade podataka* obrađuje podatke o korištenju usluga i na osnovi njih stvara bilješke korištenja. *Podatkovni upravljač* pruža jednostavan način upisa bilješki korištenja u *UP spremnik*. Bilježenje korištenja započinje primanjem podataka korištenja koje šalje *ZNP* modul (1). *Logika obrade podataka* prima podatke korištenja, stvara bilješke korištenja te ih šalje u *Podatkovni upravljač* (2). *Podatkovni upravljač* upisuje bilješke korištenja u podatkovni spremnik (3, 4) i vraća potvrdu (5). Logika obrade podataka vraća *ZPN* modulu konačnu potvrdu o uspješnosti cjelokupnog postupka (6).

7. Programsко ostvarenje sustava Nadzornik

Sustav *Nadzornik*, njegovi podsustavi i moduli ostvareni su primjenom *Microsoft .NET Framework* [75] sustava potpore izgradnji i izvođenju primjenskih sustava. Moduli sustava *Nadzornik* povezani su labavo primjenom načela arhitekture zasnovane na uslugama. Pojedini moduli izgrađeni su kao razredi u jeziku C# i ostvareni primjenom metodologije razvoja programske potpore zasnovane na objektima.

Moduli *UP spremnik* i *Pozivatelj* ostvareni su primjenom prethodno izgrađenih cjelina sustava *MidArc* [42], dok su moduli *Registracija*, *Autentikacija* i *PP* podsustav u cijelosti razvijeni za potrebe sustava *Nadzornik*. Funkcionalnosti *UP spremnika* u potpunosti su ostvarene primjenskim sustavom *Gospodarenje podacima* koji je dio sustava *MidArc*. Primjenski sustav *Gospodarenje podacima* koristi bazu podataka *Microsoft SQL Server 2000* [78] i skup razreda za izvođenje operacija nad podacima oblikovanim u bazi podataka. Programsko ostvarenje primjenskog sustava *Gospodarenje podacima* opisano je u [43, 44]. Funkcionalnosti modula *Pozivatelja* u potpunosti su ostvarene primjenskim sustavom *Tentacle* koji je dio sustava *MidArc*. *Tentacle* je ostvaren kao *Windows Forms* primjenski sustav koji upravlja radom lokalnog komunikacijskog zastupnika odlaznog HTTP prometa, a opis njegovog programskog ostvarenja nalazi se u [45].

U odjeljku 7.1 opisuje se sustav potpore izgradnji i izvođenju primjenskih sustava *Microsoft .NET Framework* koji je korišten u izgradnji modula *Registracije*, *Autentikacije* i *PP* podsustava. Ostvarenje modula *Registracije* opisano je u odjeljku 7.2, a ostvarenje modula *Autentikacije* u odjeljku 7.3. Ostvarenje modula *PP* podsustava prikazano je u odjeljku 7.4. Konačno, u odjeljku 7.5 opisuje se postavljanje i prilagođavanje ostvarenog sustava *Nadzornik*.

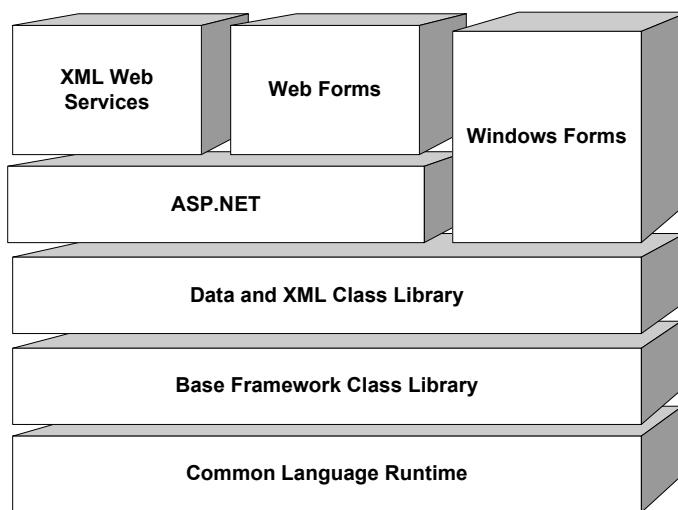
7.1. Sustav Microsoft .NET Framework

Sustav Microsoft .NET Framework namijenjen je potpori razvoju i izvođenju raspoloženih primjenskih sustava povezanih Internetom. Sustav sadrži skup programskih jezika, razvojnih okruženja, programskih knjižnica i drugih razvojnih alata namijenjenih brzom razvoju raspoloženih primjenskih sustava. Nadalje, sustav uključuje i prividni stroj koji omogućuje strojno neovisno izvođenje razvijenih primjenskih sustava.

Osnovni dijelovi sustava *Microsoft .NET Framework* prikazani su na slici 7-1. Najniži sloj sustava .NET Framework čini podsustav potpore izvođenju *Common Language Runtime (CLR)*. CLR podsustav sastoji se od prividnog stroja (engl. *virtual machine*) koji

pomoću *JIT* prevoditelja (engl. *just-in-time compiler*) [76] izvodi strojno nezavisni kôd *CIL* (*Common Intermediate Language*). Programi prevedeni u *CIL* kôd mogu se izvoditi na bilo kojoj sklopoškoj platformi i operacijskom sustavu na kojem je postavljen *CLR* podsustav.

Sljedeći sloj sustava *.NET Framework* čini skup programskih knjižnica *Base Framework Class Library*. Navedene programske knjižnice podupiru rad s podatkovnim tipovima, ulazno-izlazne operacije i ostale temeljne funkcionalnosti potrebne za ostvarenje programske logike. Iznad sloja *Base Framework Class Library* nalazi se skup programskih knjižnica nazvanih *Data and XML Class Library*. Navedene knjižnice pružaju potporu naprednom radu s podacima u bazama podataka te potporu obradi *XML* dokumenata.



Slika 7-1: *.NET Framework* sustav potpore izgradnji i izvođenju primjenskih sustava

Viši slojevi sustava *.NET Framework* omogućuju izgradnju i izvođenje triju osnovnih vrsta primjenskih sustava: primjenski sustavi *Windows Forms*, *XML Web Services*, i *Web Forms*. Primjenski sustavi *Windows Forms* zasnovani su na grafičkom sučelju i pristupa im se lokalno na računalima s operacijskim sustavom *Windows*. Primjenski sustavi *XML Web Services* i *Web Forms* su primjenski sustavi kojima se pristupa s udaljenih računala primjenom globalne mreže Internet. Primjenski sustavi *XML Web Services* omogućuju udaljenim korisnicima izvođenje primjenske logike primjenom *Web Services* skupa tehnologija. Primjenski sustavi *Web Forms* pružaju udaljenim korisnicima funkcionalnosti primjenske logike putem grafičkog sučelja izgrađenog primjenom dinamičkih Web stranica. Izvođenje primjenskih sustava *XML Web Services* i *Web Forms* ostvaruje *ASP.NET* podsustav [77]. Komunikacija udaljenih računala s *ASP.NET* podsustavom ostvaruje se putem Microsoftovog poslužitelja *Internet Information Services (IIS)*.

7.1.1. Posluživanje Web Forms stranica ASP.NET podsustavom

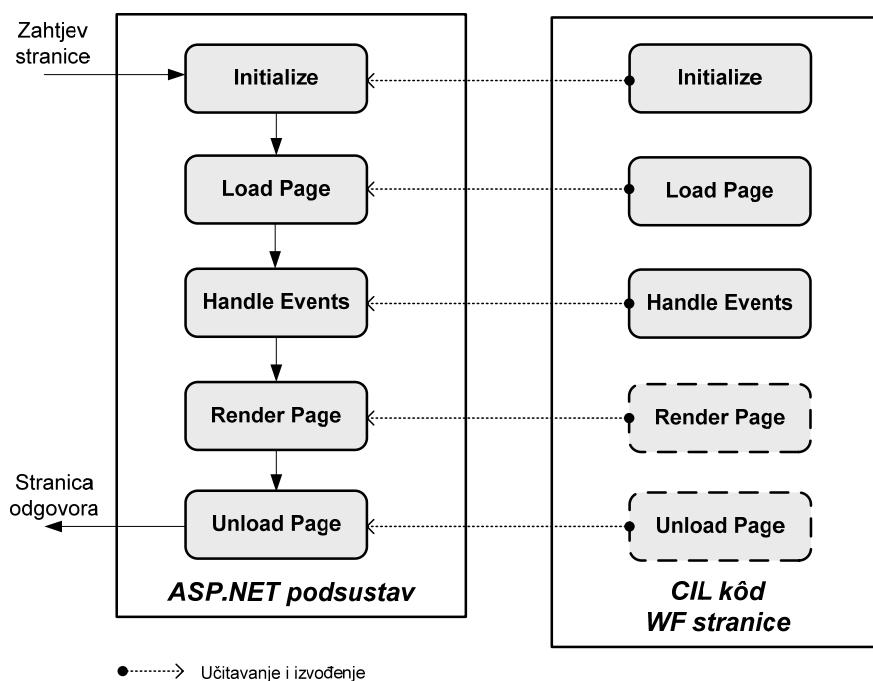
U osnovnom modelu posluživanja dinamičkih Web stranica, poslužitelj stvara novu stranicu svaki put kada je korisnik zatraži. Korisnik traži dohvati stranicu od poslužitelja slanjem HTTP zahtjeva, a poslužitelj vraća HTTP odgovor koji sadrži novostvorenu stranicu. Nakon što poslužitelj završi obradu korisničkog zahtjeva i pošalje stranicu korisniku, svi podaci o stranici brišu se iz lokalne memorije poslužitelja. Time se oslobođaju sredstva potrebna za posluživanje što većeg broja korisničkih zahtjeva, no varijable i stanje stranice ne pamte se na poslužitelju. Nakon primanja korisnikovog zahtjeva za postavljanje stranice (engl. *post request*), kojim se nastavlja korisnikova interakcija s dinamičkom stranicom, poslužitelj iznova provodi postupak stvaranja i posluživanja stranice odgovora. Pritom su poslužitelju dostupne samo informacije koje je korisnik dodao na stranicu, jer su one uključene u zahtjev za postavljanje stranice.

Opisani osnovni model posluživanja dinamičkih Web stranica primjenjen je u ostvarenju registracije korisnika i usluga sustava *Nadzornik*. Pružatelj usluge zahtjeva više dinamičkih stranica koje interaktivno nadopunjuje podacima o usluzi tijekom sjednice posluživanja stranica registracije. Stoga je tijekom sjednice posluživanja stranica registracije potrebno pamtitи zajedničke vrijednosti obrade registracijskih stranica na poslužitelju. Za pamćenje vrijednosti obrade na poslužiteljskoj strani primjenjuju se napredne mogućnosti podsustava *ASP.NET*.

ASP.NET podsustav omogućuje jednostavnu uporabu podataka za postavljanje stranice i pamćenje varijabli na poslužiteljskoj strani. Podaci zahtjeva za postavljanje stranica sadrže informaciju o promjeni stanja stranice na strani korisnika. Podsustav *ASP.NET* tumači podatke zapisane u zahtjevu za postavljanje stranica i sprema ih u objekt *ViewState*. Podaci objekta *ViewState* dostupni su tijekom obrade zahtjeva za postavljanje stranice i olakšavaju programsko ostvarenje logike za posluživanje stranice. Nadalje, *ASP.NET* omogućuje pamćenje podataka koji nisu uzrokovani akcijama korisnika, već služe za pamćenje varijabli na poslužitelju. Na primjer, objekt *SessionState* u *ASP.NET* podsustavu pamti podatke tijekom posluživanja niza uzastopnih zahtjeva jedne sjednice.

ASP.NET podsustav ostvaruje posluživanje dinamičkih *WF (Web Forms)* stranica primjenom modela poticanog događajima (engl. *event-driven model*). Postupak posluživanja *WF* stranica zasnovan na modelu poticanom događajima prikazan je na slici 7-2. Postupak se izvodi u pet koraka: *Initialize*, *Load Page*, *Handle Events*, *Render Page* i *Unload Page*. Tijekom prva tri koraka izvode se temeljne funkcionalnosti dinamičke izgradnje stranice i obrade korisnikovih informacija. Navedenim koracima ostvaruje se glavnina primjenskih

funkcionalnosti *WF* stranice i stoga se oni najčešće programski ostvaruju. Posljednja dva koraka omogućuju detaljnije upravljanje procesom posluživanja dinamičkih stranica, no najčešće se ne ostvaruju posebno već se koriste postojeće funkcionalnosti razreda *ASP.NET* podsustava.

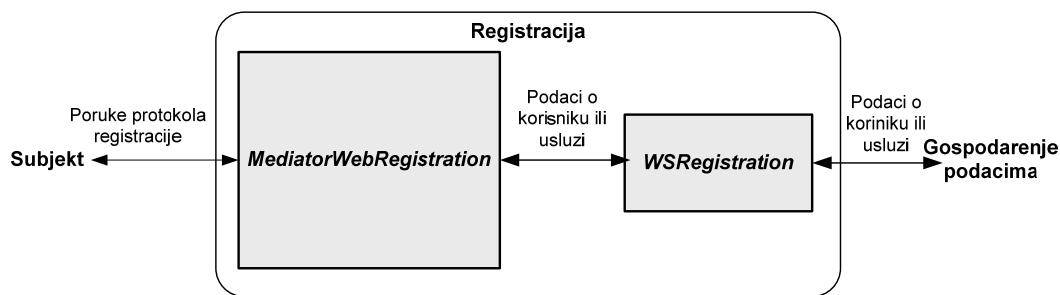


Slika 7-2: Postupak posluživanja *Web Forms* stranice *ASP.NET* podsustava

Prvi korak postupka posluživanja stranice je *Initialization*. U tom koraku se gradi objekt *ViewState* na osnovi informacija koje je korisnik unio na stranicu. Drugi korak postupka je *Load Page*, koji služi za dinamičku izgradnju *obličja* (engl. *controls*) na stranici koju se poslužuje korisniku. U navedenom koraku moguće je i očitavati promjene stanja dinamičkih obličja ako je korisnik poslao zahtjev za postavljanje stranice. Treći korak je *Handle Events* kojim se obrađuju događaji potaknuti korisnikovim akcijama na dinamičkoj stranici. Četvrti korak je *Render Page*, koji započinje proces izgradnje stranice oblikovane na poslužitelju tijekom prva tri koraka obrade stranice. Izgradnju stranice automatizirano izvodi *ASP.NET* podsustav, osim ako je korišteno posebno oblikovanje stranice. Zadnji korak postupka je *Unload Page*, u kojem se izgrađena stranica šalje korisniku. Na kraju zadnjeg koraka uništavaju se sredstva zauzeta tijekom obrade stranice.

7.2. Registracija

Modul *Registracija* izgrađen je od dva međusobno povezana sustava *MediatorWebRegistration* i *WSRegistration* kao što je prikazano na slici 7-3. Primjenjski sustav *MediatorWebRegistration* namijenjen je prikupljanju registracijskih podataka o korisniku ili usluzi putem Web stranica. Sustav *WSRegistration* ostvaren je kao *Web Services* usluga koja primjenskom sustavu *MediatorWebRegistration* pruža funkcionalnosti jednostavnog upisivanja prikupljenih podataka u primjenjski sustav *Gospodarenje podacima*. Usluga *WSRegistration* izlaže skup operacija za upis podataka u primjenjski sustav *Gospodarenje podacima* putem standardnih *Web Services* pristupnih sučelja. Primjenjski sustav *MediatorWebRegistration* na osnovi *WSDL* opisnika usluge *WSRegistration* nalazi informacije potrebne za pozivanje operacija usluge *WSRegistration* i poziva odgovarajuće operacije.



Slika 7-3: Programsко ostvarenje modula Registracija

7.2.1. Usluga *WSRegistration*

Operacije usluge *WSRegistration* prikazane su u dodatku A. Operacijom *RegisterUser* upisuju se podaci o korisniku u primjenjski sustav *Gospodarenje podacima*. Operacijom *RegisterService* u primjenjski sustav *Gospodarenje podacima* upisuju se *opći* podaci o primjenskoj usluzi. *Opći* podaci o primjenskoj usluzi su podaci koji ne ovise o posebnostima primjenske logike usluge. Primjer *općih* podataka su adresa i opis usluge koji su pridruženi uslugama neovisno o njihovim funkcionalnostima. Operacijom *RegisterFunctions* spremaju se podaci o usluzi koji su *posebni* i ovise o posebnostima primjenske logike usluge. Primjer posebnih podataka su podaci o operacijama koje primjenska usluga izlaže, jer brojnost i vrsta operacija ovise o primjenskoj logici usluge.

Usluga *WSRegistration* ostvarena je primjenom *XML Web Services* potpore *ASP.NET* podsustava i izložena je pomoću *IIS* mrežnog poslužitelja računala domaćina. Za pristup usluzi iz primjenskih sustava potrebno je izgraditi posrednički razred *RegistrationProxy*. Razred *RegistrationProxy* gradi se automatski na osnovi *WSDL* opisa

usluge *WSRegistration* primjenom *wsdl.exe* alata. Slika 7-4 prikazuje postupak izgradnje razreda *RegistrationProxy*. Alat *wsdl.exe* kao ulaz prima *WSDL* opis sučelja usluge *WSRegistration*, a za izlaz daje datoteku s programskim ostvarenjem lokalnog zastupnika u jeziku C#.

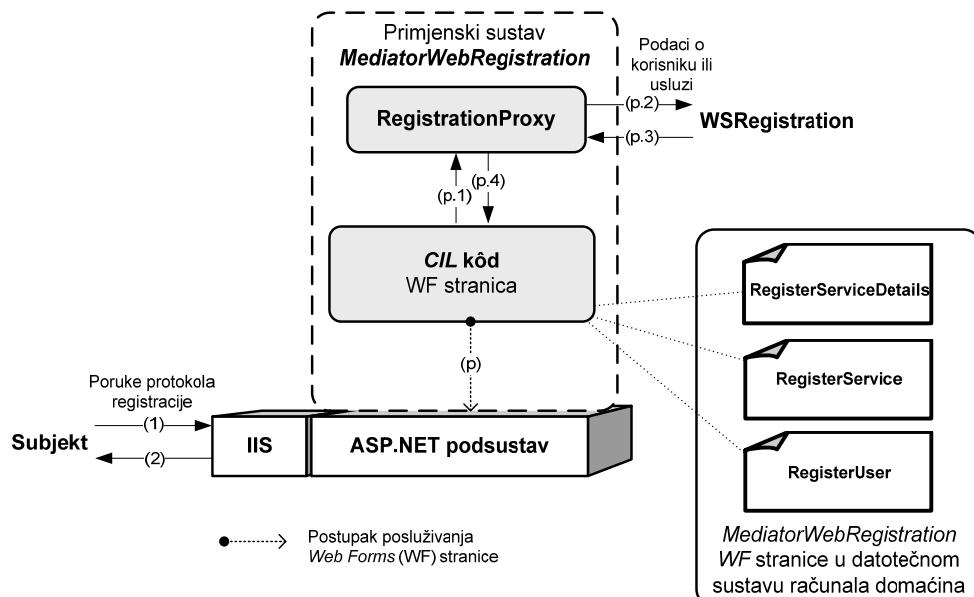


Slika 7-4: Postupak izgradnje razreda *RegistrationProxy*

7.2.2. Primjenski sustav *MediatorWebRegistration*

Osnova primjenskog sustava *MediatorWebRegistration* je skup dinamičkih Web stranica kojima sustav prikuplja podatke o korisnicima i uslugama. Podaci o korisnicima i uslugama prikupljaju se iz poruka protokola registracije koje šalju subjekti. Primjenski sustav *MediatorWebRegistration* prima poruke protokola registracije, obrađuje ih, izdvaja podatke o korisnicima ili uslugama, te tijekom rada koristi uslugu *WSRegistration* za upisivanje podataka u sustav za gospodarenje podacima.

Na slici 7-5 prikazano je programsko ostvarenje primjenskog sustava *MediatorWebRegistration* u sustavu *Nadzornik* te njegove veze prema subjektima i usluzi *WSRegistration*. Primjenski sustav *MediatorWebRegistration* primjenjuje *Web Forms (WF)* stranice *ASP.NET* podsustava za ostvarenje funkcionalnosti prikupljanja podataka pomoću



Slika 7-5: Programsko ostvarenje primjenskog sustava *MediatorWebRegistration*

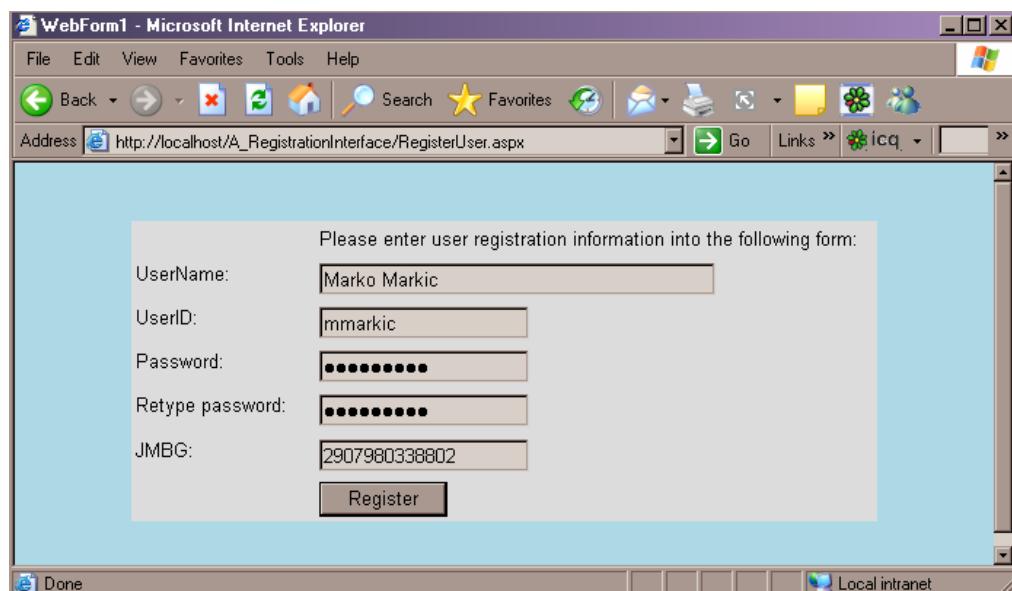
Web obrazaca. Upravljačka logika *MediatorWebRegistration* WF stranica razvijena je u jeziku C# i prevedena u CIL kôd kojeg izvodi ASP.NET podsustav.

Postupak registracije započinje zahtjevom za registraciju koji subjekt šalje IIS poslužitelju (1). IIS poslužitelj potom dostavlja udaljenom subjektu WF stranice registracije primjenom potpore ASP.NET podsustava (2). Tijekom postupka posluživanja stranica, ASP.NET podsustav izvodi primjensku logiku stranica (p). Primjenska logika stranica prikuplja podatke koje je subjekt unio na stranice. Dodatno, primjenska logika stranica upisuje prikupljene podatke o uslugama ili korisnicima u sustav za gospodarenje podacima. Upis podataka u sustav za gospodarenje podacima izvodi se posredovanjem razreda *RegistrationProxy* (p.1, p.4). Razred *RegistrationProxy* zaprima podatke od primjenske logike stranica i poziva operacije usluge *WSRegistration* (p.2, p.3) primjenom standardnih *Web Services* protokola.

WF stranice registracije sastoje se od dvije grupe stranica namijenjenih registraciji korisnika i registraciji usluga. Stranice za registraciju usluga uključuju stranicu za unos općih podataka o usluzi i stranicu za unos posebnih podataka o usluzi.

Ostvarenje stranice za registraciju korisnika

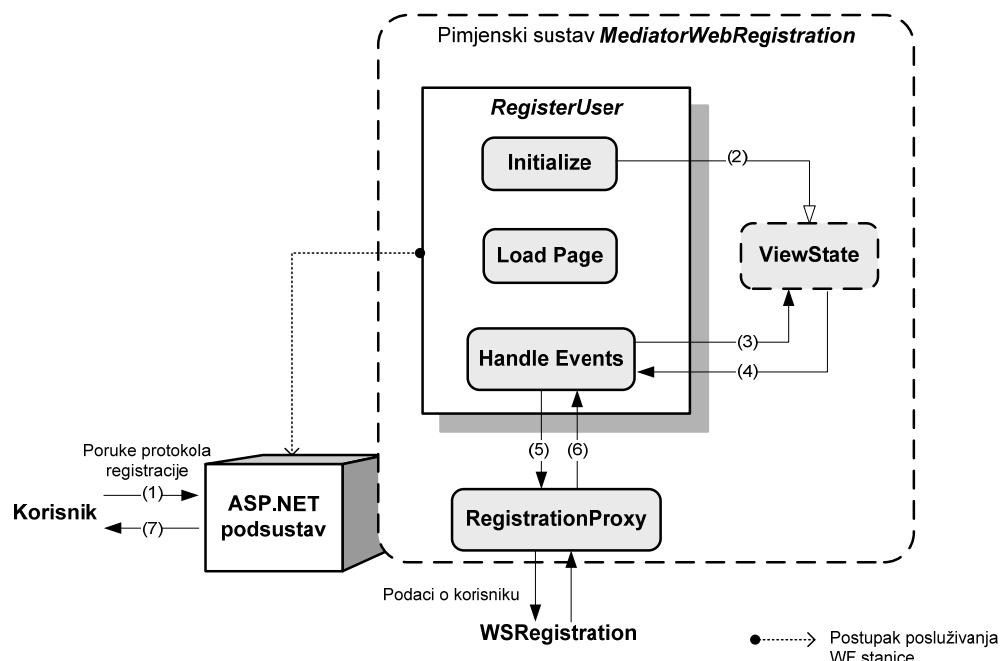
Postupak registracije korisnika izvodi se prema protokolu registracije korisnika koji je opisan u odjeljku 6.4.1 i 6.5.2. Nakon što korisnik zatraži registraciju od IIS mrežnog poslužitelja *MediatorWebRegistration* stranica, poslužitelj korisniku šalje *RegisterUser* stranicu prikazanu na slici 7-6. Stranica sadrži polja za unos imena i prezimena korisnika,



Slika 7-6: Registracijska stranica za prikupljanje podataka o korisniku

imena korisnika u sustavu, zaporce i JMBG-a. Nakon unosa potrebnih podataka, pritiskom tipke "Register" korisnik šalje poslužitelju zahtjev za postavljanje stranice koja sadrži sve unesene informacije.

Na slici 7-7 prikazana je obrada zahtjeva za postavljanje stranice *RegisterUser*. Nakon što *ASP.NET* podsustav primi zahtjev za postavljanje stranice (1), izvodi se postupak prikupljanja podataka o korisniku. U *Initialize* koraku obrade stranice *RegisterUser* stvara se objekt *ViewState* i u njega se zapisuju vrijednosti koje je korisnik unio na stranicu (2). *Load Page* korakom obrade potom se oblikuje stranica odgovora na zahtjev postavljanja stranice. U *Handle Events* koraku obrade stranice, iz objekta *ViewState* čitaju se podaci koje je korisnik unio na stranicu (3, 4) i potom se pozivom operacije *RegisterUser* usluge *WSRegistration* upisuju podaci o korisniku u sustav za gospodarenje podacima (5, 6). Konačno, na oblikovanu stranicu odgovora upisuje se potvrda uspješnosti postupka registracije korisnika. Podsustav *ASP.NET* izgrađuje HTML stranicu odgovora i šalje ju korisniku HTTP odgovorom (7).

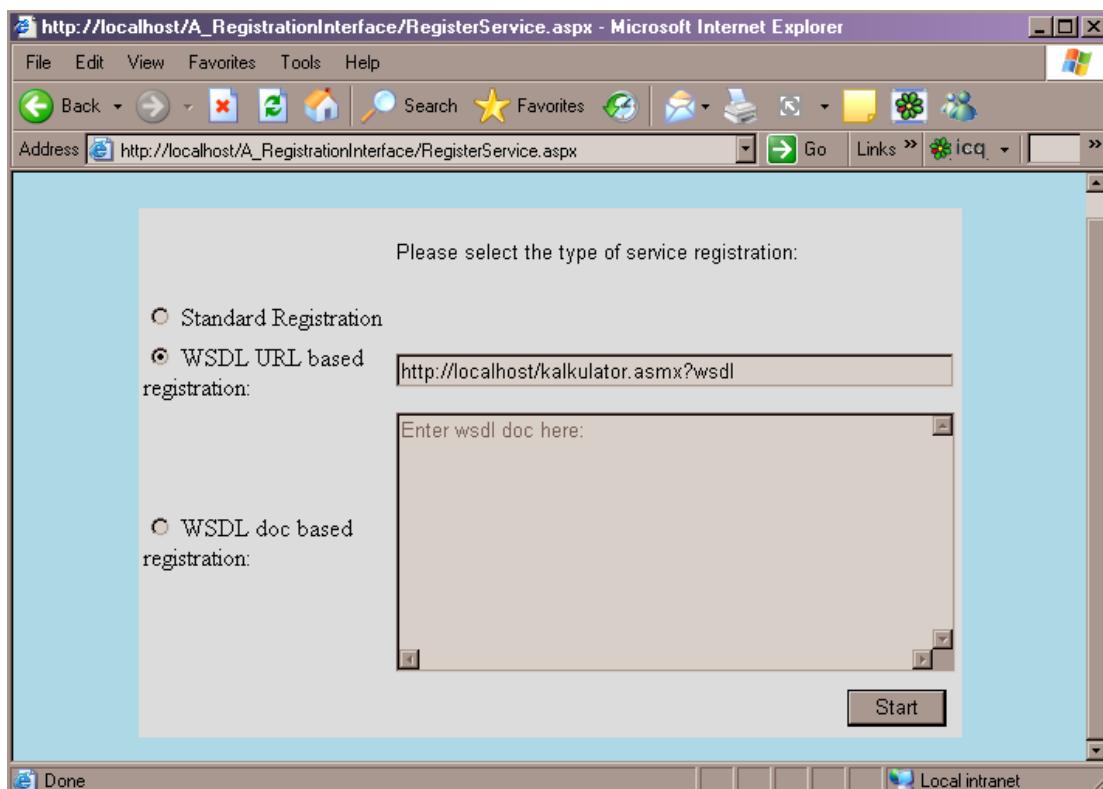


Slika 7-7: Programsko ostvarenje obrade zahtjeva za postavljanje stranice *RegisterUser*

Ostvarenje stranice za registraciju općih informacija o usluzi

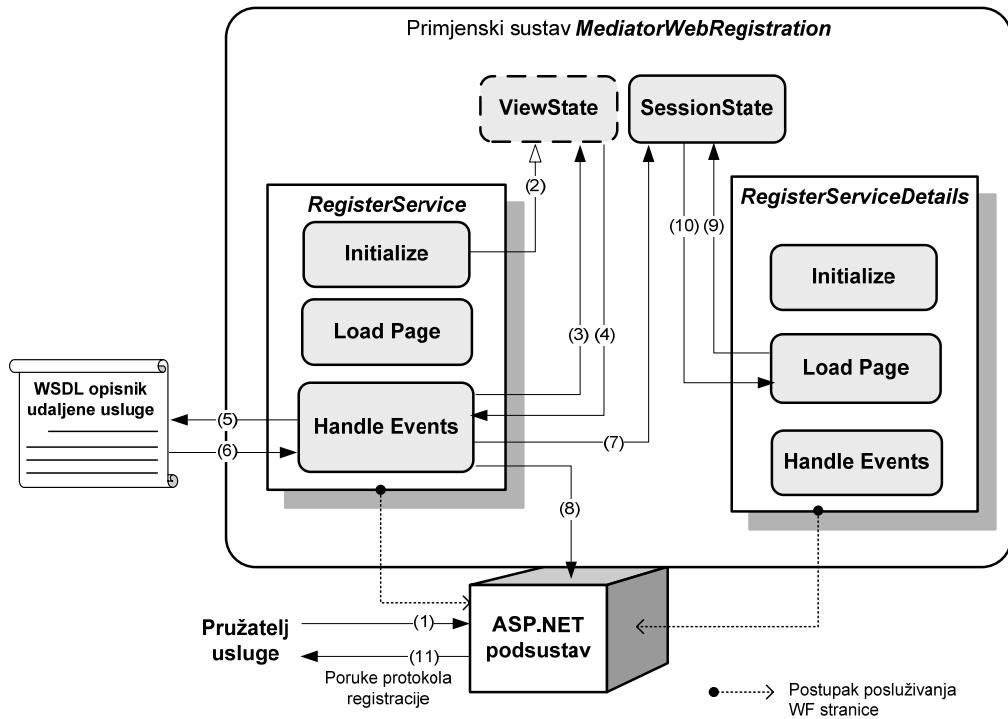
Postupak registracije usluge izvodi se prema protokolu registracije usluge koji je opisan u pododjeljku 6.4.1 i 6.5.2. Stranica *RegisterService* početna je stranica registracije usluge. Nakon što pružatelj usluge zatraži registraciju nove usluge od *IIS* mrežnog poslužitelja *MediatorWebRegistration* stranica, poslužitelj odgovara *RegisterService*

stranicom prikazanom na slici 7-8. Stranica omogućava izravni unos *WSDL* opisnika ili unos URL adrese *WSDL* opisnika usluge koja se registrira. Nakon popunjavanja podataka o *WSDL* opisniku usluge, pritiskom tipke "Start" pružatelj usluge šalje poslužitelju zahtjev za postavljanje stranice *RegisterService* koji sadrži unesene informacije.



Slika 7-8: Registracijska stranica za prikupljanje općih podataka o usluzi

Na slici 7-9 prikazan je postupak obrade zahtjeva za postavljanje stranice *RegisterService*. Nakon primanja zahtjeva (1) podsustav *ASP.NET* pokreće obradu zahtjeva izvođenjem koraka *Initialize*. U ovom koraku obrade stvara se objekt *ViewState* i u njega se zapisuju vrijednosti koje je pružatelj usluge unio na stranici registracije. U *Load Page* koraku obrade oblikuje se stranica odgovora na zahtjev za postavljanje stranice. U *Handle Events* koraku obrade dohvata se *WSDL* opisnik usluge. Ako je korisnik na stranici registracije unio URL adresu *WSDL* opisnika, onda se u ovom koraku iz objekta *ViewState* čita unesena adresa (3, 4) te se s nje dohvata *WSDL* opisnik usluge (5, 6). Ako je korisnik na stranici registracije unio čitav *WSDL* opisnik usluge, onda se opisnik izravno očitava iz objekta *ViewState* (3, 4). Dohvaćeni *WSDL* opisnik usluge potom se zapisuje u objekt *SessionState* (7). *SessionState* je dijeljeni objekt koji sprema zajedničke informacije jedne sjednice posluživanja *MediatorWebRegistration* stranica. Na kraju ovog koraka obrade, poziva se *ASP.NET* podsustav i preusmjerava se obrada zahtjeva na stranicu *RegisterServiceDetails* (8).



Slika 7-9: Programsko ostvarenje obrade zahtjeva za postavljanje stranice *RegisterService*

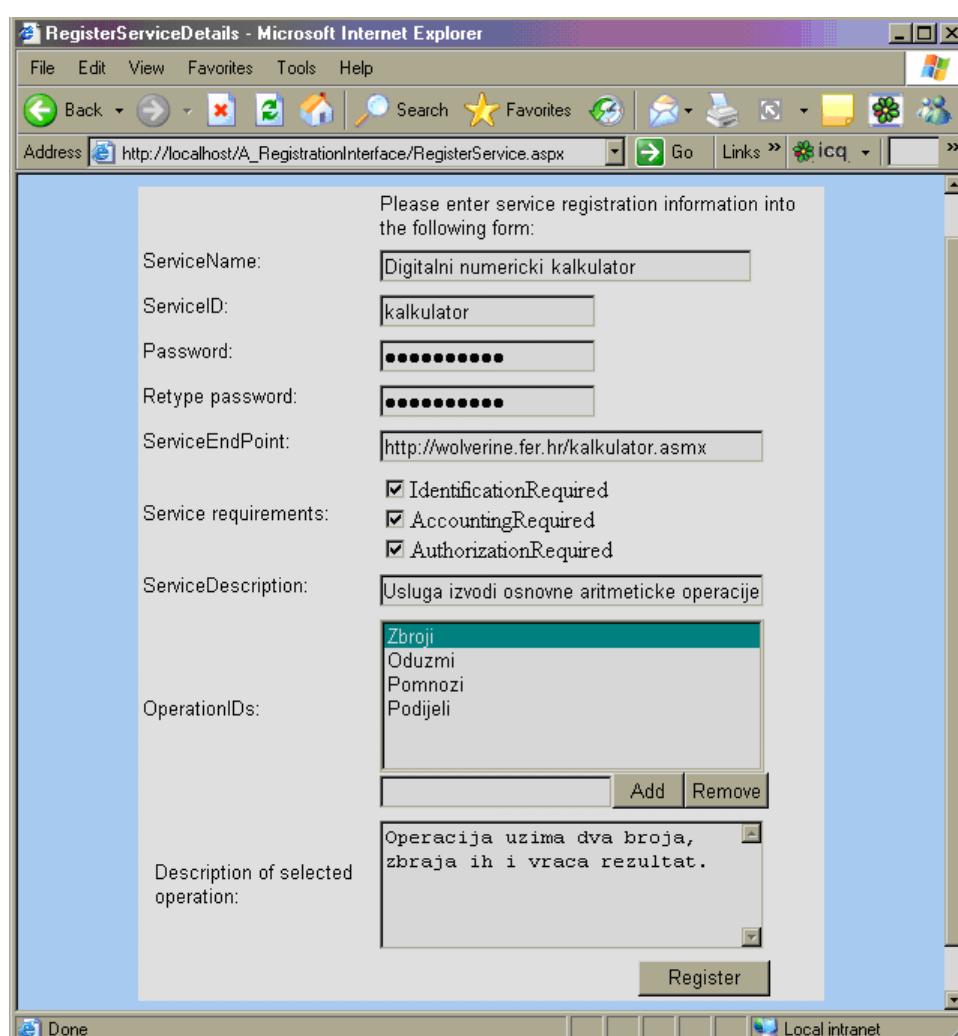
Tijekom izvođenja *Load Page* logike *RegisterServiceDetails* stranice, dohvaća se *WSDL* opisnik iz dijeljenog objekta *SessionState* (9, 10). Nadalje, u *Load Page* koraku obrade stranice *RegisterServiceDetails* parsira se *WSDL* opis udaljene usluge iz dohvaćenog *WSDL* opisnika te se dinamički oblikuje stranica odgovora prilagođena registraciji udaljene usluge. Podsustav *ASP.NET* izgrađuje HTML stranicu odgovora i šalje ju pružatelju usluge HTTP odgovorom (11).

Ostvarenje stranice za registraciju posebnih informacija o usluzi

Na slici 7-10 prikazan je primjer stranice *RegisterServiceDetails* koja je prilagođena za prikupljanje registracijskih podataka o usluzi "kalkulator". Prvi dio stranice sadrži opće informacije o usluzi i sastoji se od polja za unos imena, identifikatora usluge u sustavu, zaporce, URL adrese usluge i korisniku razumljivog opisa usluge te od polja kojima se određuje da li je tijekom nadzora pristupa usluzi potrebno izvoditi postupke provjere korisnikovog identiteta, autorizacije i praćenja korištenja usluge. Drugi dio stranice sadrži popis operacija usluge koji se automatski popunjava analizom *WSDL* opisnika usluge. Popis operacija moguće je izmijeniti dodavanjem novih operacija i brisanjem postojećih operacija primjenom tipki "Add" i "Remove". Opis svake operacije iz popisa moguće je pregledati i izmijeniti u zasebnom polju koje se nalazi ispod popisa. Pružatelj usluge popunjava podatke o usluzi na stranici *RegiserServiceDetails* i pritiskom na tipku "Register" šalje poslužitelju

zahtjev za postavljanje stranice *RegiserServiceDetails*. U zahtjevu se prenose vrijednosti koje je pružatelj usluge unio na stranici.

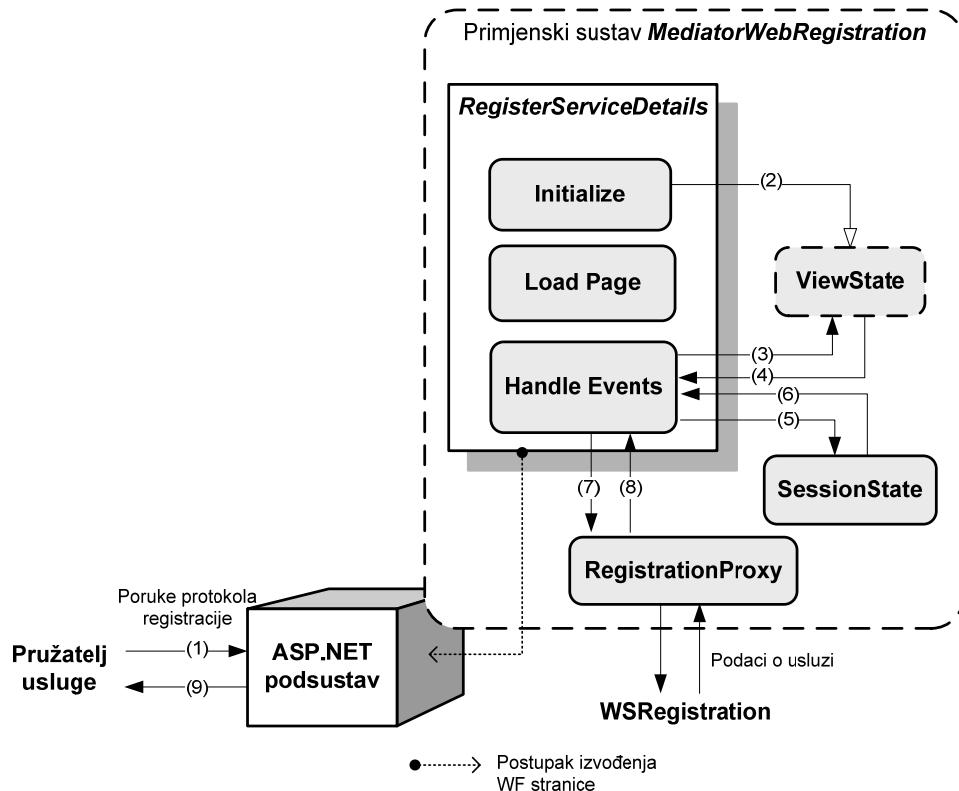
Obrada zahtjeva za postavljanje stranice *RegisterServiceDetails* prikazana je na slici 7-11. Nakon primanja zahtjeva od pružatelja usluge (1), podsustav *ASP.NET* izvodi *Initialize* korak obrade zahtjeva. U ovom koraku obrade stvara se objekt *ViewState* i u njega se zapisuju vrijednosti koje je pružatelj usluge unio na *RegiserServiceDetails* stranici (2). U *Load Page* koraku obrade oblikuje se stranica odgovora na zahtjev za postavljanje. U *Handle*



Slika 7-10: Registracijska stranica za prikupljanje posebnih podataka o usluzi

Events koraku obrade iz objekta *ViewState* čitaju se podaci o usluzi koje je pružatelj usluge unio na *RegiserServiceDetails* stranicu (3, 4). Nadalje, iz objekta *SessionState* čitaju se dijeljeni podaci o *WSDL* opisniku (5, 6), koje je u objekt *SessionState* spremila početna stranica registracije usluge. Potom se pozivom operacije *RegisterService* usluge *WSRegistration* upisuju podaci o usluzi u sustav za gospodarenje podacima (7, 8). Na kraju

ovog koraka obrade, na oblikovanu stranicu odgovora upisuje se potvrda o uspješnosti postupka registracije usluge. Podsustav *ASP.NET* potom izgrađuje HTML stranicu odgovora i šalje ju pružatelju usluge HTTP odgovorom (9).



Slika 7-11: Programsko ostvarenje obrade nastavka zahtjeva registracije stranice *RegisterServiceDetails*

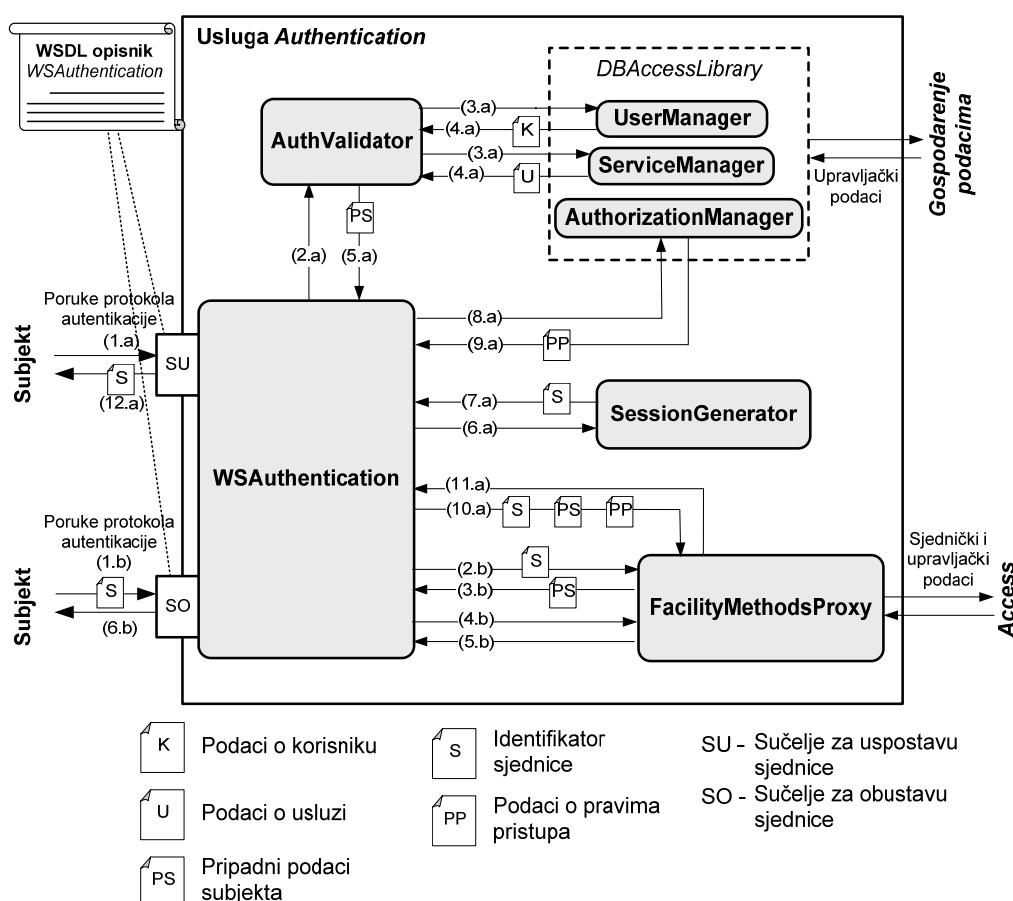
7.3. Autentikacija

Modul *Autentikacija* ostvaren je kao usluga *Authentication* koja je izgrađena primjenom jezika C# i *XML Web Services* potpore *ASP.NET* podsustava. Usluga *Authentication* izvodi se u *CLR* podsustavu sustava *.NET Framework*, a ostvarene funkcionalnosti primjenske logike pružaju se korisnicima putem standardnih *Web Services* sučelja. Primjenom poslužitelja *IIS* funkcionalnosti usluge *Authentication* ponuđene su javno svim korisnicima globalne ili lokalne mreže.

Pristupno sučelje usluge *Authentication* definirano je *WSDL* opisnikom usluge i sastoji se od operacija prikazanih u dodatku A. Usluga *Authentication* pruža udaljenim korisnicima operacije *LogOn* i *LogOff* namijenjene uspostavljanju i obustavljanju sjednica sa sustavom *Nadzornik*. Pozivom operacije *LogOn* subjekt uspostavlja sjednicu sa sustavom *Nadzornik*. Subjekt u pozivu operacije *LogOn* šalje autentikacijske podatke koje je registrirao tijekom registracije u sustav *Nadzornik*. Kao rezultat uspješne uspostave sjednice,

korisnik dobiva od usluge *Authentication* identifikator uspostavljene sjednice. Operacijom *LogOff* subjekt obustavlja sjednicu koju je uspostavio sa sustavom *Nadzornik*. Subjekt u pozivu operacije *LogOff* šalje identifikator uspostavljene sjednice.

Programsko ostvarenje usluge *Authentication* prikazano je na slici 7-12. Usluga *Authentication* ostvarena je razredima *WSAuthentication*, *AuthValidator*, *SessionGenerator*, *UserManager*, *ServiceManager* i *AuthorizationManager*. Funkcionalnosti primjenske logike usluge izlažu se sučeljima koja ostvaruje razred *WSAuthentication*. Sučelja razreda *WSAuthentication* ostvarena su primjenom *Web Services* tehnologije te su opisana *WSDL* opisnikom. Razred *WSAuthentication* sadrži sučelja za uspostavu sjednice (*SU*) i sučelja za obustavu sjednice (*SO*). Primjenom navedenih sučelja, razred *WSAuthentication* prima zahtjeve i ostvaruje upravljačku logiku obrade primljenih zahtjeva. Tijekom obrade zahtjeva, razred *WSAuthentication* koristi se razredima *AuthValidator*, *SessionGenerator* i *FacilityMethodsProxy*. Razred *AuthValidator* provjerava autentičnost subjekta koji želi uspostaviti sjednicu. Razred *SessionGenerator* stvara identifikator novo uspostavljene

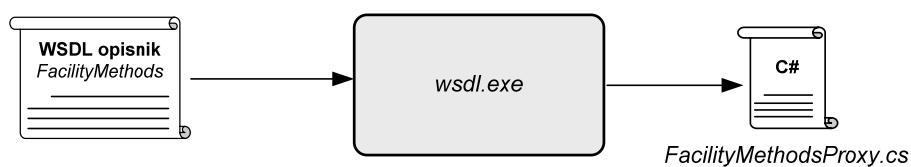


Slika 7-12: Programsко ostvarenje modula Autentikacija uslugom Authentication

sjednice. Razred *FacilityMethodsProxy* izvodi operacije brisanja i upisivanja podataka u *PP spremnik* ostvaren primjenskim sustavom Access. Programska knjižnica *DBAccessLibrary* sadrži razrede za pristup podacima u sustavu *Gospodarenje podacima*. Navedenu programsku knjižnicu koriste razredi *WSAuthentication* i *AuthValidator*. Razredi *UserManager*, *ServiceManager* i *AuthorizationManager* dijelovi su programske knjižnice *DBAccessLibrary* koji dohvaćaju podatke o korisnicima, uslugama i pravima pristupa.

Usluga *Authentication* primjenjuje se za uspostavu i obustavljanje sjednice sa sustavom *Nadzornik*. U scenariju uspostavljanja sjednice, razred *WSAuthentication* putem sučelja *SU* prima zahtjev uspostave sjednice subjekta (1.a). Pozivanjem razreda *AuthValidator*, razred *WSAuthentication* provjerava autentičnost subjekta koji je postavio zahtjev uspostave sjednice (2.a). Pomoću razreda *UserManager* i *ServiceManager*, *AuthValidator* dohvaća iz sustava *Gospodarenje podacima* autentikacijske podatke o korisniku ili usluzi koje je subjekt registrirao tijekom registracije u sustav *Nadzornik* (3.a, 4.a). *AuthValidator* potom izvodi usporedbu autentikacijskih podataka sadržanih u zahtjevu subjekta i dohvaćenih podataka. Ako se podaci podudaraju, subjekt je uspješno autenticiran, te se podaci o subjektu šalju *WSAuthentication* razredu (5.a). Razred *WSAuthentication* stvara identifikator sjednice koristeći *SessionGenerator* (6.a, 7.a), te primjenom razreda *AuthorizationManager* dohvaća podatke o pravima pristupa subjekta (8.a, 9.a). Uporabom razreda *FacilityMethodsProxy*, razred *WSAuthentication* šalje prikupljene podatke u lokalni spremnik primjenskog sustava *Access* (10.a, 11.a). Scenarij uspostavljanja sjednice završava nakon što razred *WSAuthentication* vratí subjektu potvrdu o uspostavljenoj sjednici s priloženim identifikatorom uspostavljenе sjednice (12.a).

U scenariju obustave sjednice, *WSAuthentication* razred putem sučelja *SO* prima zahtjev obustave sjednice (1.b). Zahtjev obustave sjednice sadrži identifikator sjednice koju se obustavlja. Primjenom razreda *FacilityMethodsProxy*, razred *WSAuthentication* zahtijeva brisanje sjedničkih podataka (2.b) iz lokalnog spremnika primjenskog sustava *Access*. U odgovoru sustava *Access* sadržano je ime subjekta koji u sustavu *Nadzornik* obustavlja sjednicu (3.b). U sljedećem koraku brišu se podaci o subjektu (4.b, 5.b) te se vraća odgovor o potvrdi obustavljenе sjednice (6.b).



Slika 7-13: Postupak izgradnje razreda FacilityMethodsProxy

Razred *FacilityMethodProxy* izgrađen je automatski primjenom alata *wsdl.exe*. Slika 7-13 prikazuje postupak izgradnje razreda *FacilityMethodProxy* primjenom alata *wsdl.exe*. Alat *wsdl.exe* kao ulaz prima *WSDL* opis sučelja primjenskog sustava *Access*. Izlaz je datoteka napisana u jeziku *C#* koja sadrži programsko ostvarenje lokalnog zastupnika putem kojeg se komunicira s udaljenim spremnikom primjenskog sustava *Access*.

7.4. PP podsustav

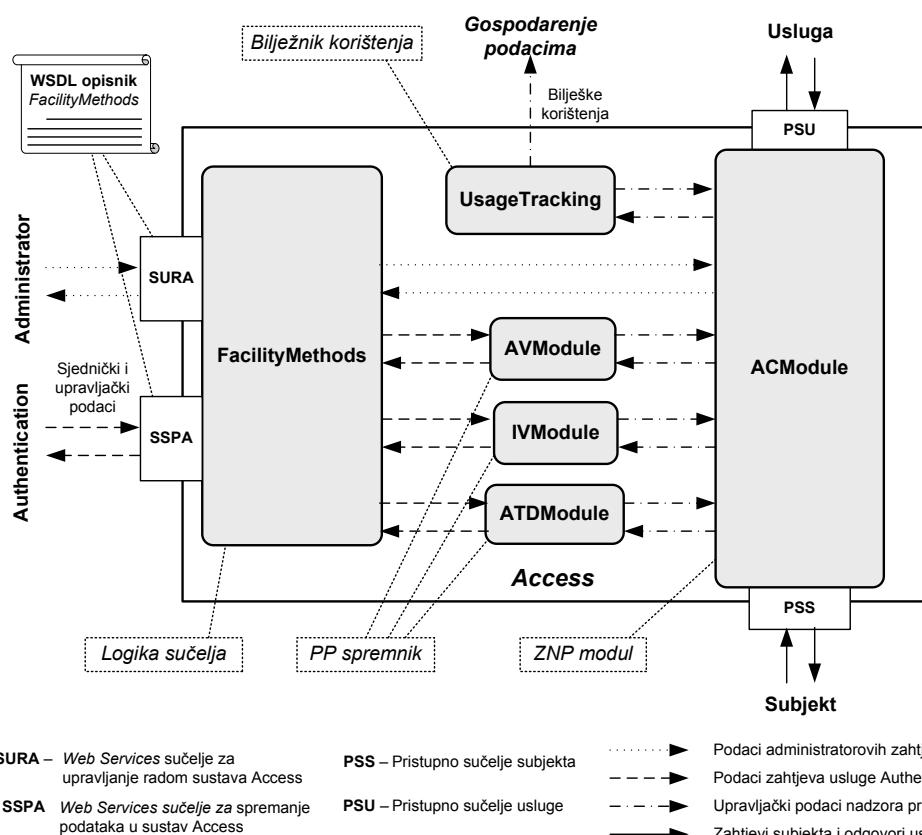
PP podsustav ostvaren je primjenskim sustavom *Access* u okviru kojeg su ostvareni svi moduli *PP* podsustava. Primjenski sustav *Access* ostvaruje module *ZNP*, *PP spremnik* i *Bilježnik korištenja*. Nadalje, primjenski sustav *Access* ostvaruje sučelja putem kojih omogućava drugim podsustavima sustava *Nadzornik* uporabu ostvarenih funkcionalnosti. Primjenski sustav *Access* ostvaruje komunikaciju s primjenskim sustavima *Gospodarenje podacima* i *Authentication*, administratorom sustava *Nadzornik* te subjektima i uslugama iz zaštićene domene sustava *Nadzornik*. Primjenski sustav *Access* sprema u sustav *Gospodarenje podacima* podatke o bilješkama korištenja usluga. Usluga *Authentication* pohranjuje u sustav *Access* upravljačke podatke za nadzor pristupa subjekata uslugama u domeni sustava *Nadzornik*. Administrator sustava *Nadzornik* upravlja radom primjenskog sustava *Access*. Subjekti koji žele koristiti primjenske usluge iz domene sustava *Nadzornik* ostvaruju pristup putem primjenskog sustava *Access*, a sustav *Access* preuzima komunikaciju s uslugama unutar domene. Programsko ostvarenje *PP* podsustava primjenskim sustavom *Access* prikazano je slikom 7-14.

Moduli *ZNP*, *PP spremnik* i *Bilježnik korištenja* ostvareni su čvrsto povezanim razredima primjenskog sustava *Access* i izvode se u jednom radnom procesu sustava *.NET Framework*. Moduli učestalo komuniciraju tijekom obrade zahtjeva subjekata koji žele koristiti primjenske usluge, pa su čvrstim povezivanjem modula ostvarena znatno bolja radna svojstva u odnosu na labavo povezani sustav. Modul *PP spremnik* ostvaren je razredima *AVModule*, *IVModule* i *ATDModule*. *ZNP* modul ostvaren je razredom *ACModule*. Modul *Bilježnik korištenja* ostvaren je razredom *UsageTracking*. *WebServices* sučelja koja izlažu funkcionalnosti primjenskog sustava *Access* ostvarena su razredom *FacilityMethods*.

Razredi *ATDModule*, *IVModule* i *AVModule* ostvaruju funkcionalnosti spremnika različitih vrsta upravljačkih podataka koji se spremaju u primjenski sustav *Access*. *ATDModule* sprema podatke o sažetim odredbama usluge koji sadrže zastavice za provjeru pristupa i praćenje korištenja, te ime i adresu usluge. *IVModule* sprema podatke o identifikatorima sjednice i subjektima koji su uspostavili sjednicu sa sustavom *Nadzornik*.

AVModule sprema podatke o pravima pristupa subjekata. Razredi *ATDModule*, *IVModule* i *AVModule* spremaju podatke u XML datoteke *atd.xml*, *iv.xml* i *av.xml*. Podatke spremljene u navedenim datotekama moguće je mijenjati uporabom razreda *FacilityMethods* s kojim su razredi *ATDModule*, *IVModule* i *AVModule* čvrsto povezani.

Razred *ACModule* ostvaruje funkcionalnosti nadzora pristupa primjenom dva radna stanja: *ONEMOGUĆEN* i *OMOGUĆEN*. U stanju *ONEMOGUĆEN*, razred *ACModule* ne omogućuje subjektima korištenje usluga u domeni sustava *Nadzornik*. U stanju *OMOGUĆEN*, razred *ACModule* omogućuje subjektima korištenje usluga u domeni sustava *Nadzornik* i štiti korištenje usluga provođenjem nadzora pristupa. *PSS* i *PSU* su pristupna sučelja za komunikaciju subjekta i usluge s razredom *ACModule*. Ako je razred *ACModule* u stanju *OMOGUĆEN*, putem tih sučelja ostvaruje se komunikacija razreda sa subjektima i uslugama. Nadalje, razred *ACModule* čvrsto je povezan s razredom *UsageTracker* kojemu tijekom nadzora pristupa šalje podatke korištenja usluga. Stanje razreda *ACModule* moguće je promijeniti uporabom razreda *FacilityMethods* s kojim je razred *ACModule* čvrsto povezan.



Slika 7-14: Programsko ostvarenje PP podsustava primjenskim sustavom Access

Razred *UsageTracker* ostvaruje funkcionalnosti za upisivanje podataka o bilješkama korištenja usluga u primjenski sustav *Gospodarenje podacima*. Razred *UsageTracker* ostvaruje komunikaciju s primjenskim sustavom *Gospodarenje podacima* primjenom skupa programskih knjižnica sustava *.NET Framework* za pristup bazi podataka *Microsoft SQL Server 2000*. Podaci o korištenju usluga koje razred *UsageTracker* primi od *ACModula* u neizmijenjenom obliku zapisuju se u primjenski sustav *Gospodarenje podacima*.

Razredom *FacilityMethods* ostvarena je logika *Web Services* pristupnog sučelja primjenskog sustava *Access*. Razred *FacilityMethods* ostvaruje dvije vrste sučelja, SSPA (sučelje za spremanje podataka u sustav *Access*) i SURA (sučelje za upravljanje radom sustava *Access*). SSPA sučelja izlažu funkcionalnosti za spremanje podataka u primjenski sustav *Access* primjenom razreda *ATDModule*, *IVModule* i *AVModule*. Navedena sučelja koristi primjenski sustav *Authentication*. SURA sučelja izlažu funkcionalnosti za upravljanje radom nadzora pristupa promjenom radnog stanja razreda *ACModule*. Administrator primjenom navedenih sučelja ostvaruje pristup do razreda *ACModule*.

U dodatku A prikazane su operacije izložene SSPA sučeljem za spremanje podataka u primjenski sustav *Access*. Operacije omogućuju usluzi *Authentication* upisivanje i brisanje upravljačkih podataka korisnika ili primjenske usluge u primjenskom sustavu *Access*. Operacija *SetAccessPolicy* upisuje podatke sažetih odredbi primjenske usluge, dok ih operacija *DelAccessPolicy* briše iz primjenskog sustava *Access*. Operacija *SetIdentityInfo* upisuje podatke o identitetu sjednice i subjektu koji koristi zadanu sjednicu. Operacija *DelIdentityInfo* briše navedene podatke. Operacija *SetIdentityAuth* upisuje podatke o pravima pristupa subjekta za pristup uslugama u domeni sustava *Nadzornik*, dok ih operacija *DelIdentityAuth* briše.

U dodatku A prikazane su operacije izložene SURA sučeljem za upravljanje radom primjenskog sustava *Access*. Operacija *IsActive* provjerava radno stanje razreda *ACModule*. *ActivateModule* operacija pokreće rad razreda *ACModule* i neovisno o prethodnom stanju mijenja stanje razreda *ACModule* u *OMOGUĆEN*. Operacija *DeactivateModule* zaustavlja rad razreda *ACModule* i neovisno o prethodnom stanju mijenja stanje razreda *ACModule* u *ONEMOGUĆEN*.

7.4.1. Ostvarenje *ACModule* razreda

ACModule ostvaruje glavni mehanizam provedbe nadzora pristupa korisnika uslugama. Mehanizam provedbe pristupa izvodi se koristeći pripremljene upravljačke podatke u razredima *ATDModule*, *IVModule* i *AVModule*, te strukture podataka za provedbu

nadzora pristupa. Mehanizam provedbe pristupa opisan je u odjeljku 6.5.6, a pripremanje upravljačkih podatka opisano je u odjeljku 6.4.2. Podatkovne strukture za provedbu nadzora pristupa izgrađuju se parsiranjem HTTP zahtjeva korisnika te HTTP odgovora usluga.

HTTP zahtjevi korisnika dijele se na HTTP zahtjeve korištenja *Web Services* usluga i HTTP zahtjeve pregleda Web stranica. Zahtjevi dohvata Web stranica razlikuju se od zahtjeva korištenja *Web Services* usluga po tome što ne sadrže SOAP poruku. Dok SOAP poruka navodi operaciju pozvane *Web Services* usluge, zahtjev dohvata stranice podrazumijeva operaciju pregleda stranice. Zahtjevi dohvata Web stranica su jednostavniji oblik zahtjeva *Web Services* usluge i u nastavku opisa smatraju se specijalnim slučajem korištenja *Web Services* usluga.

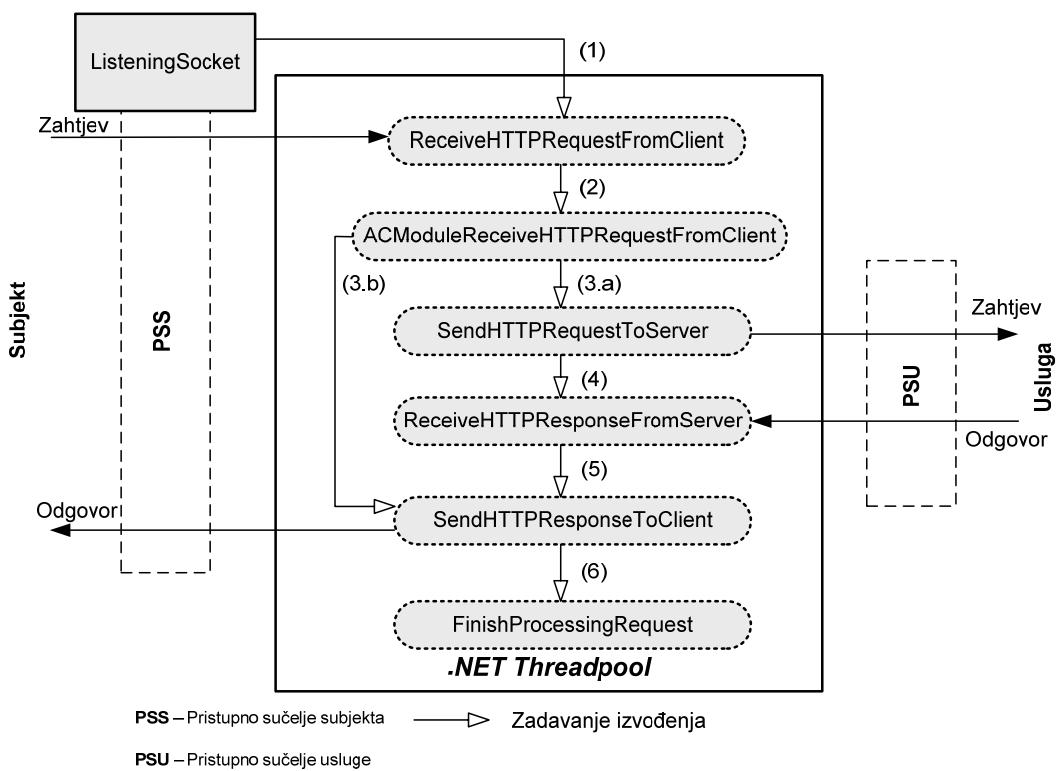


Slika 7-15: Primjer prepoznatih podataka prilikom parsiranja SOAP poruke zahtjeva koja se prenosi HTTP zahtjevom

Primjer SOAP poruke zahtjeva koja se prenosi HTTP protokolom prikazan je na slici 7-16. *ACModule* parsiranjem dobivenog HTTP zahtjeva prepoznaće adresu, identitet sjednice i operaciju Web Services usluge koju subjekt poziva. Razred *ACModule* prepoznaće adresu usluge iz naslovnog reda HTTP zahtjeva. Identifikator sjednice prepoznaće se iz *Sessionid* zaglavlja HTTP zahtjeva. Konačno, naziv pozvane operacije usluge prepoznaće se

iz tijela SOAP poruke koja se prenosi HTTP zahtjevom. Pročitani podaci zapisuju se u *ACModuleMessage* strukturu. *ACModuleMessage* struktura izgrađena je od pročitanih podataka, razreda i podrazreda HTTP zahtjeva i HTTP odgovora te dodatnih podataka o zahtjevu i odgovoru, a detaljnije je opisana u [81].

ACModuleMessage osnovna je dijeljena struktura podataka u procesu obrade zahtjeva u razredu *ACModule*. Proces obrade zahtjeva sastoji se od šest koraka: primanja zahtjeva, donošenja odluke, proslijedivanja zahtjeva i primanja odgovora od usluge, vraćanja odgovora korisniku i uništavanja stvorenih struktura i zauzetih sredstava. Na slici 7-17 prikazani su koraci u procesu obrade zahtjeva na razini programskog ostvarenja funkcijskih cjelina razreda *ACModule*.



Slika 7-16: Funkcijske cjeline procesa obrade zahtjeva u *ACModule* razredu

Proces obrade zahtjeva izvodi skup dretvi programske knjižnice *.NET Threadpool*. Svaki korak kružnog procesa jedna je funkcijска cjelina koja se šalje na izvođenje skupu dretvi. Ako u skupu dretvi nema slobodne dretve, izvođene funkcijске cjeline stavljaju se u red izvođenja. Nakon što jedna od dretvi završi s radom i postane slobodna, ona preuzima izvođenje prve funkcijске cjeline iz reda izvođenja.

Proces obrade zahtjeva započinje na *ListeningSocket* priključnici (engl. *socket*) povezanoj s pristupnim sučeljem *PSS*. Navedena priključnica zaprima dolazne HTTP

zahtjeve koji se mrežom prenose kao niz mrežnih TCP paketa [80]. Primjenom *ListeningSocket* priključnice pokreće se izvođenje funkcijске cjeline *ReceiveHTTPRequestFromClient* i primanje HTTP zahtjeva u obliku slijeda TCP paketa (1). *ReceiveHTTPRequestFromClient* funkcijска cjelina prima slijed TCP paketa, izgrađuje HTTP zahtjev, parsira HTTP zahtjev i popunjava strukturu *ACModuleMessage* pročitanim podacima iz HTTP zahtjeva. Po završetku izvođenja, funkcijска cjelina *ReceiveHTTPRequestFromClient* započinje izvođenje funkcijске cjeline *ACModuleReceiveHTTPRequestFromClient* (2). Navedena funkcijска cjelina provjerava prava pristupa na osnovi pročitanih podataka u *ACModuleMessage* strukturi. Provjerom prava pristupa donosi se odluka treba li zahtjev proslijediti do odredišne usluge. Ako se zahtjev proslijedi na odredište, izvođenje se nastavlja funkcijskom cjelinom *SendHTTPRequestToServer* (3.a). Ako se zahtjev ne proslijedi na odredište, izvođenje se nastavlja funkcijskom cjelinom *SendHTTPRequestToClient* (3.b).

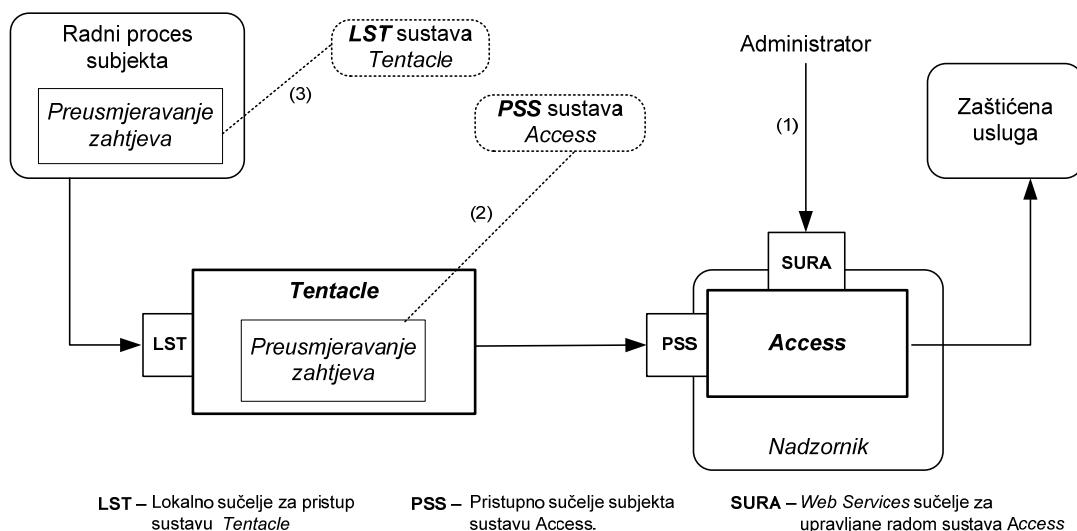
Na osnovi podataka o zahtjevu u *ACModuleMessage* strukturi, funkcijска cjelina *SendHTTPRequestToServer* stvara pristupno sučelje *PSU* putem koje proslijedi HTTP zahtjev na odredište poziva usluge. Zahtjev se proslijedi u obliku slijeda TCP paketa. Funkcijска cjelina *SendHTTPRequestToServer* pokreće izvođenje funkcijске cjeline *ReceiveHTTPResponseFromServer* (4) koja prima HTTP odgovor od usluge. Nakon primanja odgovora, navedena funkcijска cjelina pokreće izvođenje funkcijске cjeline *SendHTTPResponseToClient* (5) koja vraća odgovor subjektu. Na kraju, funkcijска cjelina *SendHTTPResponseToClient* započinje izvođenje funkcijске cjeline *FinishProcessingRequest* (6), kojom se uništavaju stvorene strukture i zauzeta sredstva tijekom obrade zahtjeva.

7.5. Postavljanje sustava *Nadzornik*

Sustav *Nadzornik* potrebno je prije uporabe postaviti (engl. *installation*) i prilagoditi njegove parametre rada (engl. *configuration*). S obzirom da su dijelovi sustava labavo povezani, sustav je moguće raspodijeliti postavljanjem primjenskih sustava *Authentication*, *Registration* i *Access* na različita računala. Računala na koja se raspodjeljuju primjenski sustavi *Authentication*, *Registration* i *Access* moraju imati postavljen .NET Framework sustav i IIS mrežni poslužitelj. Nadalje, primjenski sustav *Gospodarenje podacima* ostvaruje se na računalu koje ima postavljenu bazu podataka Microsoft SQL Server 2000, dok primjenski sustav *Tentacle* mora biti postavljen lokalno na računalima svih subjekata koji se koriste sustavom *Nadzornik*.

Nakon što su primjenski sustavi *Authentication* i *Registration* postavljeni, potrebno je subjektima omogućiti javni pristup navedenim sustavima. Dodatno, potrebno je primjenskim sustavima *Authentication* i *Registration* omogućiti spajanje s bazom podataka primjenskog sustava *Gospodarenje podacima*.

Tijekom uporabe sustava *Nadzornik* potrebno je osigurati ispravno preusmjeravanje zahtjeva subjekta prema sustavu *Nadzornik*. Slika 7-18 prikazuje postavljanje preusmjeravanja zahtjeva subjekta prema sustavu *Nadzornik*. Administrator pokreće rad sustava *Access* pozivom operacije *ActivateModule* iz *SURA* sučelja (1). Zatim se na lokalnom računalu subjekta pokreće primjenski sustav *Tentacle* i njegovoj logici za preusmjeravanje zahtjeva zadaje se adresa *PSS* sučelja sustava *Access* (2). Primjenski sustav *Tentacle* nudi grafičko sučelje za zadavanje adrese *PSS* sučelja na koju se preusmjeravaju zahtjevi subjekta.



Slika 7-17: Preusmjeravanje zahtjeva prema sustavu *Nadzornik*

Primjenski sustav *Tentacle* preusmjerava primljene zahtjeve subjekta na *PSS* sučelje, a prima zahtjeve subjekta putem *LST* sučelja. Stoga je potrebno u radnom procesu subjekta postaviti preusmjeravanje zahtjeva na lokalnu adresu *LST* sučelja primjenskog sustava *Tentacle* (3). Način postavljanja preusmjeravanja zahtjeva ovisi o primjenskom sustavu koji subjekt koristi za slanje zahtjeva. Na primjer, zahtjeve dohvata Web stranica stvara Internet preglednik stoga je primjenom ugrađene potpore preglednika potrebno postaviti preusmjeravanje zahtjeva pregleda Web stranica prema sustavu *Nadzornik*.

Zahtjevi korištenja *Web Services* usluga stvaraju se programski u *.NET Framework* razvojnoj okolini i drugim programskim okolinama. U navedenim okolinama potrebno je programski ostvariti preusmjeravanje zahtjeva korištenja *Web Services* usluge. Postoje

gotovi alati koji omogućavaju stvaranje i preusmjeravanje zahtjeva korištenja *Web Services* usluga. Stvaranje i preusmjeravanje zahtjeva korištenja usluga za potrebe ispitivanja rada sustava *Nadzornik* ostvareno je primjenom postojećeg primjenskog sustava *WebServicesStudio*. Primjenski sustav *WebServicesStudio* nudi jednostavno grafičko sučelje za pozivanje usluga s mogućnostima preusmjeravanja zahtjeva korištenja usluga.

Nakon što se na opisani način postave i prilagode prarametri rada svih dijelova sustava *Nadzornik*, sustav *Nadzornik* je spremjan za uporabu. Subjekti potom šalju zahtjeve korištenja usluga sustavu *Nadzornik*. Sustav *Nadzornik* prima zahtjeve korištenja usluga i nadzire korištenja usluga u domeni koju štiti.

8. Zaključak

Računarstvo zasnovano na uslugama omogućuje novi način razvoja i ponude programskih funkcionalnosti. Programske funkcionalnosti nude se u obliku usluga. Usluge se oglašavaju, pronalaze, koriste i sastavljaju te omogućuju jednostavno stvaranje novih usluga i poslovnih procesa. Poslovni procesi, međutim, postavljaju velike zahtjeve na sigurnost usluga. Stoga sustavi koji pružaju usluge moraju izgraditi odgovarajuću potporu sigurnosti. Jedan od važnijih elemenata sigurnosti u takvim sustavima je nadzor pristupa uslugama.

U okviru magistarskog rada definirana je arhitektura, te je oblikovan i programski ostvaren sustav *Nadzornik*. Razvijeni sustav stvara zaštićenu domenu unutar koje nadzire izlaganje i korištenje usluga. Pružatelji usluga prijavljuju usluge u domenu *Nadzornika* i izlažu funkcionalnosti svojih usluga. Korisnici se prijavljuju i koriste usluge ponuđene unutar zaštićene domene sustava *Nadzornik*.

Funkcionalnosti sustava *Nadzornik* grupirane su u dva glavna podsustava: podsustav *Uspostava pristupa* i podsustav *Provedba pristupa*. Podsustav *Uspostava pristupa* ostvaruje spremanje i zadavanje upravljačkih podataka za nadzor pristupa. Podsustav *Provedba pristupa* primjenjuje upravljačke podatke i na osnovi njih nadzire korištenje usluga. Ovakva podjela funkcionalnosti omogućuje raspodjeljivanje tih dvaju podsustava na više računala. Podsustav *Uspostava pristupa* ostvaruje postupke registracije i autentikacije. Postupkom registracije prikupljaju se i spremaju upravljački podaci o korisnicima i uslugama u spremnik podsustava *Uspostava pristupa*. Na osnovi tih podataka korisnicima i uslugama omogućena je uspostava sjednice za rad u domeni sustava. Postupak autentikacije provjerava identitet korisnika i usluga te uspostavlja sjednicu. Dodatno, postupak autentikacije zapisuje upravljačke podatke u podustav *Provedba pristupa*. Podsustav *Provedba pristupa* ostvaruje pristup uslugama u zaštićenoj domeni sustava *Nadzornik* i izvodi nadzor pristupa primjenom upravljačkih podataka koje prima od podsustava *Uspostava pristupa*. Rukovođen upravljačkim podacima, podsustav izvodi proslijđivanje ovlaštenih zahtjeva korištenja usluga, odbacivanje neovlaštenih zahtjeva korištenja usluga i bilježi podataka o korištenju usluga.

Podsustavi *Uspostava pristupa* i *Provedba pristupa* surađuju izmjenjujući upravljačke podatke o dogovorenom pristupu unutar domene. Razmjena upravljačkih podataka nužna je za ispravan nadzor pristupa u domeni. Protokol *ponude* (engl. *push*) i protokol *potraživanja* (engl. *pull*) dva su najznačajnija protokola razmjene upravljačkih

podataka u sustavima nadzora pristupa. Protokoli su analizirani i vrednovani za primjenu u sustavu *Nadzornik*. Vrednovanje je provedeno s obzirom na raspolijeljenost podsustava *Uspostava pristupa* i *Provedba pristupa* na različita računala. Vrednovanjem je utvrđena potreba za oblikovanjem novog protokola prilagođenog razmjeni upravljačkih podataka u danoj raspolijeljenoj okolini. Oblikovan je i ostvaren protokol na temelju postojeća dva protokola razmjene podataka.

U sustavu *Nadzornik* uočeno je nekoliko mogućih smjernica za proširenja i poboljšanja tijekom budućeg rada. Predlaže se proširenje sustava uvođenjem potpore za suradnju sa sigurnosnim uslugama drugih sigurnosnih sustava. Proširenje je poželjno ostvariti primjenom dinamičkog i kasnog povezivanja sigurnosnih usluga. Radi povezivanja i suradnje s drugim sigurnosnim rješenjima, potrebna je primjena standardnih XML sigurnosnih protokola za autentikaciju i autorizaciju.

9. Literatura

- [1] M.P. Singh and M.N. Huhns, "**Service-Oriented Computing: Semantics, Processes, Agents,**" John Wiley & Sons, Ltd., 2005.
- [2] R. Sessions, "**Fuzzy Boundaries: Objects, Components, and Web Services,**" *ACM Queue*, Vol. 2, No. 9, December 2004, pp. 40-47.
- [3] M. Turner, D. Budgen, P. Brereton, "**Turning Software into a Service,**" *Computer*, Vol. 36, No. 10, October 2003, pp. 38-44.
- [4] J.-Y. Chung, K.-J. Lin, and R.G. Mathieu (guest editors), "**Web Services Computing: Advancing Software Interoperability,**" *Computer*, Vol. 36, No. 10, October 2003, pp. 35-37.
- [5] M.P. Papazoglou and D. Georgakopoulos, "**Service-Oriented Computing**", *Communications of the ACM*, Vol. 46, No. 10, October 2003.
- [6] C. Peltz, "**Web Services Orchestration and Choreography**", *Computer*, vol 36. No. 10, 2003. pp. 46-52.
- [7] D. Booth et al., "**Web Services Architecture,**" W3C Working Group Note, 2004, <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>.
- [8] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "**Web Service Description Language (WSDL) 1.1,**" W3C Note, March 2001; <http://www.w3.org/TR/wsdl>.
- [9] T. Berners-Lee, L. Masinter, and M. McCahill, "**Uniform Resource Locators (URL),**" RFC 1738, December 1994, <http://www.ietf.org/rfc/rfc1738.txt>.
- [10] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau, "**Extensible Markup Language (XML) 1.0**", third edition, W3C Recommendation, February 2004, <http://www.w3.org/TR/REC-xml/>.
- [11] T. Bellwood, et al: "**UDDI Version 3.0**", UDDI Spec TC, July 2002, <http://uddi.org/pubs/uddi-v3.00-published-20020719.htm>.
- [12] M. Gudgin, M. Hadley, N. Mendelsohn, J.J. Moreau, and H.F. Nielsen "**SOAP Version 1.2 Part 1: Messaging Framework**", W3C Recommendation, June 2003,,<http://www.w3.org/TR/soap12-part1/>.
- [13] J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "**Hypertext Transfer Protocol - HTTP/1.1,**" RFC 2616, June 1999, <http://www.ietf.org/rfc/rfc2616.txt>.
- [14] J. Postel and J. Reynolds, "**File Transfer Protocol (FTP)**", RFC 959, October 1985, <http://www.ietf.org/rfc/rfc959.txt>.

- [15] J. B. Postel, "**Simple Mail Transfer Protocol (SMTP)**", RFC 821, August 1982, <http://www.ietf.org/rfc/rfc0821.txt>.
- [16] K. Czajkowski et al., "**WS-Resource Framework**", Globus Alliance, IBM, Computer Associates International, Fujitsu Laboratories of Europe, and Hewlett-Packard, version 1.0, May 2004; <http://www.globus.org/wsrf>.
- [17] D. Talia: "**The Open Grid Services Architecture: Where the Grid Meets the Web**", IEEE Internet Computing, Vol. 6, No. 6, November 2002, pp. 67-71.
- [18] S. Tuecke, et al: "**Open Grid Services Infrastructure (OGSI) Version 1.0**", June 2003; http://www-unix.globus.org/toolkit/draft-ggf-ogsi-gridservice-33_2003-06-27.pdf.
- [19] OASIS Web Services Notification TC, **WS-Notification**, set of related specifications; http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn
- [20] T. Andrews et al., "**Business Process Execution Language for Web Services (BPEL4WS)**", Microsoft, IBM, Siebel Systems, BEA, and SAP, version 1.1, May 2003; <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>.
- [21] B. W. Lampson, "**Computer Security in the real World**," *Computer*, Vol. 37, No. 6, June 2004, pp. 37-46.
- [22] E. Bertino and R. Sandhu "**Database Security: Concepts, Approaches, and Challenges**," *IEEE Transactions on Dependable and Secure Computing*, vol.2, no.1, January 2005, pp. 2-19.
- [23] Ravi S. Sandhu. "**Lattice-Based Access Control Models**," *Computer*, Vol. 26, No. 11, November 1993, pp. 9-19.
- [24] M. Andrews and J. A. Whittaker (editor), "**Computer Security**", *IEEE Security & Privacy*, Vol. 2, No. 5, September 2004, pp. 68-71.
- [25] "**An Introduction to Cryptography**", PGP 6.5.1 documentation, Network Associates Inc. and its Affiliated Companies, November 1998, <http://www.pgpi.org/doc/pgpintro/>.
- [26] D. Škvorc, "**Sigurni prijenos podataka u mrežama s posredničkim sustavima**", *diplomski rad*, Fakultet elektrotehnike i računarstva, Zagreb, 2003.
- [27] Withdrawn FIPS Publication 46-3, "**Data Encryption Standard (DES)**", NIST, published in October 1999, withdrawn in May 2005, <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>.
- [28] FIPS Publication 197, "**Advanced Encryption Standard (AES)**", NIST, November 2001, <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.

- [29] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman, "**A Method for Obtaining Digital Signatures and Public-Key Cryptosystems**," *Communications of the ACM*, Vol. 21, No. 2, February 1978, 120-126.
- [30] "**Introduction to Public-key Cryptography**", *PGP 6.5.1 documentation*, chapter 1, Network Associates, Inc. and its Affiliated Companies, <ftp://ftp.pgpi.org/pub/pgp/6.5/docs/english/IntroToCrypto.pdf>.
- [31] D. Richard Kuhn, Vincent C. Hu, W. Timothy Polk, and Shu-Jen Chang, "**Introduction to Public Key Technology and the Federal PKI Infrastructure**", NIST SP 800-32, February 2001, <http://www.csrc.nist.gov/publications/nistpubs/800-32/sp800-32.pdf>.
- [32] ITU-T, **Recommendation X.509**, *Information technology - Open Systems Interconnection - The Directory*: Public-key and attribute certificate frameworks, pre-published August 2005, <http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=T-REC-X.509>.
- [33] A.S. Tanenbaum, "**Computer Networks**", 4th Edition, Prentice Hall, 2003
- [34] A.S. Tanenbaum, "**Distributed Operating Systems**", Prentice Hall, 1995
- [35] B. Neuman, "**Proxy-Based Authorization and Accounting for Distributed Systems**", *Proceedings of the 13th International Conference on Distributed Computing Systems*, IEEE, 1993, pp. 283-291.
- [36] H. Kim and S. Bahk, "**Real-Time Visualization of Network Attacks on High-Speed Links**," *IEEE Network*, Vol. 18, No. 5, September 2004, pp. 30-39.
- [37] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman, "**Role-Based Access Control Models**", *Computer*, Vol. 29, No. 2, February 1996, pp 38-47.
- [38] D.E. Bell and L.J. LaPadula, "**Secure Computer Systems: Unified Exposition and Multics Interpretation**," Technical Report MTR 2997, The Mitre Corporation, Bedford, Mass., 1976.
- [39] R. Yavatkar, D. Pendarakis, and R. Guerin, "**A Framework for Policy-based Admission Control**", RFC 2753, January 2000; <http://www.faqs.org/rfcs/rfc2753.html>.
- [40] Clayton Donley, "**LDAP Programming, Management, and Integration**", Manning Publications, 1st edition, November 2002.
- [41] M. Wahl, T. Howes, and S. Kille, "**Lightweight Directory Access Protocol (v3)**", RFC 2251, December 1997.
- [42] S. Srbljić, I. Skuliber, I. Benc, M. Štefanec, A. Milanović, B. Dellas, S. Dešić, L. Budin, N. Bogunović, D. Huljenić, and A. Carić, "**Service Development and Application Integration with Public Information System Mediator**", *Proceedings of*

the 12th IEEE Mediterranean Electrotechnical Conference (MELECON 2004), Dubrovnik, Croatia, May 2004, Vol. 2, pp. 713-718.

- [43] I. Benc, "Gospodarenje podacima u posredniku javnog informacijskog sustava", *magistarski rad*, Fakultet elektrotehnike i računarstva, Zagreb, 2003.
- [44] M. Popović, "Priručna memorija posredničkog sustava", *diplomski rad*, Fakultet elektrotehnike i računarstva, Zagreb, 2003.
- [45] I. Grudenić, "Lokalni zastupnici u middleware sustavima", *diplomski rad*, Fakultet elektrotehnike i računarstva, Zagreb, 2002.
- [46] F. Plavec, "Direktorij middleware sustava", *diplomski rad*, Fakultet elektrotehnike i računarstva, Zagreb, 2002.
- [47] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence, "AAA Authorization Framework", *RFC 2904*, August 2000; <http://www.faqs.org/rfcs/rfc2904.html>.
- [48] S. Farrell, J. Vollbrecht, P. Calhoun, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence, "AAA Authorization Requirements", *RFC 2906*, August 2000; <http://www.faqs.org/rfcs/rfc2906.html>.
- [49] M. Lorch, B. Cowles, R. Baker, L. Gommans, P. Madsen, A. McNab, L. Ramakrishnan, K. Sankar, D. Skow and M. Thompson, "Conceptual Grid Authorization Framework and Classification", Global Grid Forum GFD-1.38, November 2004; <http://www.ggf.org/documents/GFD.38.pdf>.
- [50] A. Nadalin, C. Kaler, P. Hallam-Baker, and R. Monzillo "Web Services Security: SOAP Message Security 1.0", OASIS Standard 200401, March 2004.
- [51] D. Eastlake, J. Reagle, D. Solo, M. Bartel, J. Boyer, B. Fox, B. LaMacchia, and E. Simon, "XML-Signature Syntax and Processing", W3C Recommendation, 12 February 2002, <http://www.w3.org/TR/xmldsig-core/>.
- [52] T. Imamura, B. Dillaway, and E. Simon, "XML Encryption Syntax and Processing," W3C Recommendation, December 2002, <http://www.w3.org/TR/xmlenc-core/>.
- [53] "Security Assertion Markup Language", version 2.0, OASIS Committee Specification Set, March 2005, <http://www.oasis-open.org/committees/security/#documents>.
- [54] "Extensible access control markup language (XACML)", version 2.0, OASIS Committee Specification, February 2005, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.
- [55] Globus Project, <http://www.globus.org>

- [56] I. Foster and C. Kesselman, "**Computational Grids: The Future of High Performance Distributed Computing**," San Mateo, CA: Morgan Kaufman, 1998.
- [57] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "**A Security Architecture for Computational Grids**," *Proceedings of the 5th ACM conference on Computer and communications security*, San Francisco, California, USA, 1998, pp. 83-92.
- [58] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke, "**A Community Authorization Service for Group Collaboration**," *Third IEEE International Workshop on Policies for Distributed Systems and Networks*, Monterey, California, USA, June 2002, pp. 50-59.
- [59] D. W. O'Callaghan and B. A. Coghlan, "**Bridging Secure WebCom and European DataGrid Security for Multiple VOs over Multiple Grids**", *Third International Symposium on Parallel and Distributed Computing/Third International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks* (ISPDC/HeteroPar'04), July 2004. pp. 225-231.
- [60] M. Niinimaki, J. White, W. Som de Cerff, J. Hakkala, T. Niemi, M. Pitkanen, "**Using Virtual Organizations Membership System with EDG's Grid Security and Database Access**," *Database and Expert Systems Applications, 15th International Workshop on* (DEXA'04), August 2004, pp. 517-522.
- [61] M. Lorch, D. B. Adams, D. Kafura, M. S. R. Koneni, A. Rathi, and S. Shah, "**The PRIMA System for Privilege Management, Authorization and Enforcement in Grid Environments**", *Fourth International Workshop on Grid Computing*, Phoenix, AR, USA, November 2003, pp. 109-118.
- [62] S. Farrell and R. Housley, "**An Internet Attribute Certificate Profile for Authorization**", RFC 3281, April 2002; <http://www.ietf.org/rfc/rfc3281.txt>.
- [63] D.W. Chadwick, A. Otenko, and E. Ball, "**Role-Based Access Control With X.509 Attribute Certificates**", *IEEE Internet Computing*, Vol. 7, No. 2, March 2003, pp. 62-69.
- [64] **SlashGrid**, <http://www.gridpp.ac.uk/authz/slashgrid/>, posjećeno 2005-08-29.
- [65] M.Popovic, I. Benc, and S. Srbljić, "**Access Control in Hierarchies of Protected Cells**", *Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics*, Orlando, Florida, USA, July 2005, Vol. 3, pp. 356-361.
- [66] D. Skrobo, A. Milanovic, and S. Srbljic, "**Distributed Program Interpretation in Service-Oriented Architectures**", *Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics*, Orlando, Florida, USA, July 2005, Vol. 4, pp. 193-197.

- [67] A. Milanovic, S. Srblic, D. Skrobo, D. Capalija, and S. Reskovic: "**Coopetition Mechanisms for Service-Oriented Distributed Systems**", *Proceedings of the 3rd International Conference on Computing, Communications and Control Technologies*, Austin, Texas, USA, July 2005, pp. 118-123.
- [68] I. Gavran, "**Korisnički jezik programskog modela zasnovanog na uslugama**", *magistarski rad u izradi*, Fakultet elektrotehnike i računarstva, Zagreb, 2006.
- [69] D. Skrobo, "**Raspodijeljeno usporedno interpretiranje programa u arhitekturama zasnovanim na uslugama**", *magistarski rad*, Fakultet elektrotehnike i računarstva, Zagreb, 2006.
- [70] D. Škvorc, "**Prividna mreža računalnih sustava zasnovanih na uslugama**", *magistarski rad*, Fakultet elektrotehnike i računarstva, Zagreb, 2006.
- [71] M. Podravec, "**Otkrivanje i postavljanje usluga u sustavima zasnovanim na uslugama**", *magistarski rad*, Fakultet elektrotehnike i računarstva, Zagreb, 2006.
- [72] A. Milanović, "**Programski model zasnovan na uslugama**", *doktorska disertacija*, Fakultet elektrotehnike i računarstva, Zagreb, 2005.
- [73] **Programmable Internet Environment**, <http://www.pie.fer.hr/>, posjećeno 2005-08-29.
- [74] I. Sučić, "**Praćenje korištenja sredstava usluga putem posrednika**", *diplomski rad*, Fakultet elektrotehnike i računarstva, Zagreb, 2005.
- [75] J. Richter, "**Applied Microsoft .NET Framework Programming**", 1st edition, Microsoft Press, January 2002.
- [76] S. Srblić, "**JEZIČNI PROCESORI 2: Analiza izvornog i sinteza ciljnog programa**", Element, Zagreb, 2003.
- [77] J. Liberty, D. Hurwitz, "**Programming ASP.NET**", 2nd edition, O'Reilly, September 2003.
- [78] K. Delaney, "**Inside Microsoft SQL Server 2000**", Microsoft Press, November 2000.
- [79] D. Raggett, A.L. Hors, and I. Jacobs "**HTML 4.01 Specification**", W3C Recommendation, December 1999, <http://www.w3.org/TR/html4/>.
- [80] J. Postel, "Transmission Control Protocol", IETF Informational Standard, RFC 793, <http://www.apps.ietf.org/rfc/rfc793.html>.
- [81] I. Zuzak, "**Implementation of the Access subsystem of the distributed Computing Environment**", *Proceedings of the Fourth Summer Camp 2004*, Ericsson Nikola Tesla, Zagreb, 2004.

10. Životopis

Rođen sam 29. srpnja 1980. u Požegi. Maturirao sam 1998. godine u Gimnaziji u Požegi te upisao prvu godinu Fakulteta elektrotehnike i računarstva u Zagrebu. Za postignuti uspjeh na prvoj godini studija dodijeljena mi je nagrada "Josip Lončar". Od druge godine studija primao sam stipendiju Ministarstva znanosti i tehnologije. Znanstveno-nastavno vijeće Fakulteta elektrotehnike i računarstva odobrilo mi je upis programa za završetak studija s naglaskom na znanstvenoistraživačkom radu. Diplomirao sam 2003. godine na studiju računarstva diplomskim radom pod naslovom "Priručna memorija posredničkog sustava".

Tijekom dviju završnih godina studija bio sam demonstrator na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave na laboratorijskim vježbama iz dva kolegija dodiplomske nastave: "Automati, formalni jezici i jezični procesori I" te "Automati, formalni jezici i jezični procesori II". Aktivno sam sudjelovao u projektu "Middleware Architecture in New Generation Networks" u suradnji tvrtke Ericsson Nikola Tesla d.d. i Fakulteta elektrotehnike i računarstva. Tijekom ljeta 2002. godine bio sam sudionik ljetne radionice u tvrtki Ericsson Nikola Tesla d.d. u organizaciji Zavoda za elektroniku, mikroelektroniku, računalne i intelligentne sustave i istoimene tvrtke.

Akademске godine 2003./2004. upisao sam poslijediplomski studij na Fakultetu elektrotehnike i računarstva, smjer Jezgra računarstva. Zaposlen sam na projektu 0036051 "Raspodijeljeni ugrađeni računalni sustavi" čiji je voditelj akademik prof.dr.sc. Leo Budin. Obavljam poslove znanstvenog novaka u suradničkom zvanju asistenta na Zavodu za elektroniku, mikroelektroniku, računalne i intelligentne sustave na Fakultetu elektrotehnike i računarstva. Suradnik sam na tehnološkom projektu TP-01/036-29 "Okrilje posrednika javnog informacijskog sustava" pod vodstvom mentora prof.dr.sc Siniše Srbljića.

Sudjelovao sam na međunarodnoj konferenciji s člankom iz područja sigurnosti računalnih sustava zasnovanih na uslugama.

11. Sažetak

Nadzor pristupa uslugama predstavlja neizostavnu funkcionalnost računalnih sustava zasnovanih na uslugama. Usluge računalnih sustava udružuju se u zaštićene domene primjenom specijaliziranih sigurnosnih sustava za nadzor pristupa. Magistarski rad opisuje arhitekturu i programsko ostvarenje sustava *Nadzornik* kojim se uspostavlja zaštićena domena i nadzire pristup uslugama ostvarenim primjenom Web Services tehnologija. Sustav omogućuje učlanjivanje korisnika i usluga u zaštićenu domenu sustava *Nadzornik* primjenom postupaka registracije i autentikacije. Dodatno, sustav ostvaruje mehanizme putem kojih nadzire pristup uslugama u zaštićenoj domeni. Sustav se sastoji od podsustava *Provedba pristupa* i podsustava *Uspostava pristupa*. Podsustav *Uspostava pristupa* ostvaruje funkcionalnosti registracije i autentikacije putem otvorenih i standardiziranih sučelja. Podsustav *Provedba pristupa* ostvaruje zaštićenu domenu i izvodi nadzor pristupa uslugama na gruboj i visokoj razini razlučivosti. Tijekom nadzora pristupa uslugama, podsustavi *Provedba pristupa* i *Uspostava pristupa* razmjenjuju upravljačke podatke. U okviru rada definiran je novi mješoviti model razmjene upravljačkih podataka između podsustava *Provedba pristupa* i *Uspostava pristupa* metodom *ponude-potražnje*. Definirani model prilagođen je razmjeni upravljačkih podataka u raspodijeljenoj okolini u kojoj se podsustavi *Provedba pristupa* i *Uspostava pristupa* nalaze na udaljenim računalima.

12. Summary

Access control is a critical mechanism for enforcing security in service-oriented information systems. Services of information systems are grouped into protected domains established by specialized access control security systems. The master thesis presents architecture and implementation of access control system named *Access*. *Access* system is developed to establish protected domain and control access to services implemented as Web Services. Registration and authentication of *Access* system allow new users and services to join to protected domain. Moreover, *Access* system establishes access control mechanisms and ensures secure and controlled access to services. The system consists of *Access Decision Function* and *Access Enforcement Function* subsystems. *Access Decision Function* subsystem provides registration and authorization through well-defined, open, and standardized interfaces. *Access Enforcement Function* subsystem implements protected domain and enforces coarse-grained and fine-grained access control. *Access Decision Function* and *Access Enforcement Function* subsystems exchange access control data. The thesis defines new hybrid push-pull model for exchanging access control data. Hybrid push-pull data exchange model is applicable for exchanging access control data when *Access Enforcement Function* and *Access Decision Function* subsystems are located on remote machines of distributed environment.

13. Ključne riječi

Raspodijeljeni sustavi, otvoreni sustavi, Internet, računarstvo zasnovano na uslugama, sigurnost, sigurnosna prepreka, nadzor pristupa, odredbe nadzora pristupa, zastupnik nadzora pristupa.

Distributed systems, open-ended systems, Internet, Service-Oriented Computing (SOC), security, firewall, access control, access control policy, access control proxy.

14. Dodatak A

Tablica 14-1: Operacije usluge *WSRegistration*

Ime operacije	Opis operacije	
RegisterUser	<p>Operacija je namijenjena upisu podataka o korisniku u primjenski sustav <i>Gospodarenje podacima</i>. Ulaznim parametrima operacije navodi se skup podataka o korisniku, a povratna vrijednost operacije definira uspješnost registracije podataka te sadrži upravljačku poruku u slučaju pojave pogreške.</p>	
	Ulagni parametri	<ul style="list-style-type: none"> • string <i>NameSurname</i> - puno ime i prezime korisnika • string <i>UserID</i> - ime korisnika u sustavu <i>Nadzornik</i> • string <i>Password</i> - zaporka • string <i>JMBG</i> - jedinstveni matični broj građana
	Izlagni parametri	<ul style="list-style-type: none"> • <i>RegistrationStatus</i> <ul style="list-style-type: none"> ○ boolean <i>Code</i> – vrijednost TRUE ako je registracija uspješno provedena ○ string <i>Message</i> – upravljačka poruka kojom se opisuje pogreška u registraciji
RegisterService	<p>Operacija je namijenjena upisu općih podataka o usluzi u primjenski sustav <i>Gospodarenje podacima</i>. Ulaznim parametrima operacije navodi se skup općih podataka o usluzi, a povratna vrijednost operacije definira uspješnost registracije podataka te sadrži upravljačku poruku u slučaju pojave pogreške.</p>	
	Ulagni parametri	<ul style="list-style-type: none"> • string <i>ServiceName</i> – puno ime usluge • string <i>ServiceID</i> – ime usluge u sustavu <i>Nadzornik</i> • string <i>Password</i> – zaporka • string <i>ServiceEndPoint</i> – adresa usluge • boolean <i>IdentificationRequired</i> – zastavica identifikacije • boolean <i>AuthorizationRequired</i> – zastavica autentikacije • boolean <i>AccountingRequired</i> – zastavica praćenja korištenja • string <i>ServiceDescription</i> – opisnik usluge • string <i>WSDLLocation</i> – adresa opisnika usluge
	Izlagni parametri	<ul style="list-style-type: none"> • <i>RegistrationStatus</i> <ul style="list-style-type: none"> ○ boolean <i>Code</i> – vrijednost TRUE ako je registracija uspješno provedena ○ string <i>Message</i> – upravljačka poruka kojom se opisuje pogreška u registraciji
RegisterFunction	<p>Operacija je namijenjena upisu posebnih podataka o operaciji usluge u primjenski sustav <i>Gospodarenje podacima</i>. Ulaznim parametrima operacije navodi se skup posebnih podataka o operaciji usluge, a povratna vrijednost operacije definira uspješnost registracije podataka te sadrži upravljačku poruku u slučaju pojave pogreške.</p>	
	Ulagni	<ul style="list-style-type: none"> • string <i>ServiceID</i> – ime usluge u sustavu

	parametri	<p><i>Nadzornik</i></p> <ul style="list-style-type: none"> • string <i>FunctionID</i> – ime operacije usluge • string <i>FunctionDescription</i> – opis operacije usluge
	Izlazni parametri	<ul style="list-style-type: none"> • <i>RegistrationStatus</i> <ul style="list-style-type: none"> ◦ boolean <i>Code</i> – vrijednost TRUE ako je registracija uspješno provedena ◦ string <i>Message</i> – upravljačka poruka kojom se opisuje pogreška u registraciji

Tablica 14-2: Operacije usluge *Authentication*

Ime operacije	Opis operacije	
<i>LogOn</i>		Operacija je namijenjena za uspostavljanje sjednice. Ulaznim parametrima operacije navode se autentikacijski podaci subjekta, a povratna vrijednost definira sjednicu subjekta te sadrži upravljačku poruku u slučaju pojave pogreške.
	<i>Ulazni parametri</i>	<ul style="list-style-type: none"> • string <i>SubjectID</i> – ime korisnika ili usluge u sustavu • string <i>SuppliedPassword</i> – zaporka
	<i>Izlazni parametri</i>	<ul style="list-style-type: none"> • <i>AuthenticationStatus</i> <ul style="list-style-type: none"> ◦ bool <i>Code</i> – vrijednost TRUE ako je operacija uspješno izvedena ◦ string <i>Message</i> – upravljačka poruka kojom se opisuje pogreška kod uspostave sjednice ◦ string <i>SessionID</i> – identifikator sjedničkog ključa
<i>LogOff</i>	Operacija je namijenjena za obustavljanje sjednice. Ulaznim parametrima navodi se identifikator sjednice koju se želi obustaviti. Povratna vrijednost definira uspješnost obustavljanja sjednice, te sadrži upravljačku poruku u slučaju pogreške.	
	<i>Ulazni parametri</i>	<ul style="list-style-type: none"> • string <i>SessionID</i> – identifikator sjednice
	<i>Izlazni parametri</i>	<ul style="list-style-type: none"> • <i>AuthenticationStatus</i> <ul style="list-style-type: none"> ◦ bool <i>Code</i> – vrijednost TRUE ako je operacija uspješno izvedena ◦ string <i>Message</i> – upravljačka poruka kojom se opisuje pogreška kod obustavljanja sjednice ◦ string <i>SessionID</i> – identifikator sjednice

Tablica 14-3: Operacije za spremanje podataka u primjenski sustav *Access* izložene SSPA sučeljem

Ime operacije	Opis operacije	
<i>SetAccessPolicy</i>	Operacija je namijenjena upisivanju podataka sažetih odredbi usluge u primjenski sustav <i>Access</i> . Ulaznim parametrima navode se podaci sažetih odredbi usluge. Povratna vrijednost definira uspješnost upisivanja podataka sažetih odredbi usluge.	
	Ulagni parametri	<ul style="list-style-type: none"> • <i>ServiceInfo</i> <ul style="list-style-type: none"> ○ string <i>ServiceID</i> - ime usluge u sustavu <i>Nadzornik</i> ○ string <i>ServiceEndPoint</i> – adresa usluge • <i>ServiceSecurityProfile</i> <ul style="list-style-type: none"> ○ boolean <i>IdentificationRequired</i> – zastavica identifikacije ○ boolean <i>AuthorizationRequired</i> – zastavica autorizacije ○ boolean <i>AccountingRequired</i> – zastavica praćenja korištenja
	Izlagni parametri	<ul style="list-style-type: none"> • boolean <i>Success</i> – vrijednost TRUE ako je operacija uspješno provedena
<i>DelAccessPolicy</i>	Operacija je namijenjena brisanju podataka sažetih odredbi usluge iz primjenskog sustava <i>Access</i> . Ulaznim parametrom navodi se ime usluge, a povratna vrijednost definira uspješnost brisanja podataka sažeti odredbi usluge.	
	Ulagni parametri	<ul style="list-style-type: none"> • string <i>ServiceID</i> – ime usluge u sustavu <i>Nadzornik</i>
	Izlagni parametri	<ul style="list-style-type: none"> • boolean <i>Success</i> – vrijednost TRUE ako je operacija uspješno provedena
<i>SetIdentityInfo</i>	Operacija za upisivanje podataka o identitetu sjednice u primjenski sustav <i>Access</i> . Ulaznim parametrima navode se podaci o sjednici subjekta, a povratna vrijednost definira uspješnost upisivanja podataka.	
	Ulagni parametri	<ul style="list-style-type: none"> • string <i>SubjectID</i> – ime korisnika ili usluge u sustavu <i>Nadzornik</i> • string <i>SessionID</i> – identifikator sjednice
	Izlagni parametri	<ul style="list-style-type: none"> • boolean <i>Success</i> – vrijednost TRUE ako je operacija uspješno provedena
<i>DelIdentityInfo</i>	Operacija je namijenjena brisanju podataka o identitetu sjednice iz primjenskog sustava <i>Access</i> . Ulaznim parametrom navodi se identifikator sjednice koju treba izbrisati, a povratna vrijednost potvrđuje uspješnost brisanja subjektove sjednice.	
	Ulagni parametri	<ul style="list-style-type: none"> • string <i>SessionID</i> – identifikator sjednice
	Izlagni parametri	<ul style="list-style-type: none"> • string <i>SubjectID</i> – ime korisnika ili usluge u sustavu <i>Nadzornik</i>
<i>SetIdentityAuth</i>	Operacija je namijenjena upisivanju podataka o pravima pristupa subjekta u primjenski sustav <i>Access</i> . Ulaznim parametrima navode se operacije usluga koje subjekt smije pozivati. Povratna vrijednost definira uspješnost upisivanja podataka.	
	Ulagni parametri	<ul style="list-style-type: none"> • string <i>SubjectID</i> – ime korisnika ili usluge u

		<ul style="list-style-type: none"> • sustavu <i>Nadzornik</i>
	Izlazni parametri	<ul style="list-style-type: none"> • Authorizations [] <i>ListOfAuthorizations</i> – lista dozvola pristupa subjekta • boolean <i>Success</i> – vrijednost TRUE ako je operacija uspješno provedena
<i>DeleteIdentityAuth</i>	Operacija je namijenjena brisanju podataka o pravima pristupa subjekta iz primjenskog sustava <i>Access</i> . Ulagnim parametrom navodi se ime subjekta, a povratna vrijednost definira uspješnost brisanja podataka.	
	Ulagni parametri	<ul style="list-style-type: none"> • string <i>SubjectID</i> – ime korisnika ili usluge u sustavu <i>Nadzornik</i>
	Izlazni parametri	<ul style="list-style-type: none"> • boolean <i>Success</i> – vrijednost TRUE ako je operacija uspješno provedena

Tablica 14-4: Operacije za upravljanje radom primjenskog sustava *Access* izložene SURA sučeljem

Ime operacije	Opis operacije	
<i>IsActive</i>	Operacija je namijenjena provjeri stanja <i>ZNP</i> modula. Ne sadrži ulazne parametre, a povratna vrijednost sadrži poruku o stanju modula.	
	Ulagni parametri	-
	Izlazni parametri	<ul style="list-style-type: none"> • string <i>Message</i> – upravljačka poruka kojom se opisuje stanje ZNP modula
<i>ActivateModule</i>	Operacija za pokretanje <i>ZNP</i> modula. Ne sadrži ulazne parametre, a povratna vrijednost definira poruku o uspješnosti pokretanja.	
	Ulagni parametri	-
	Izlazni parametri	<ul style="list-style-type: none"> • string <i>Message</i> – upravljačka poruka kojom se opisuje uspješnost pokretanja ZNP modula
<i>DeactivateModule</i>	Operacija je namijenjena zaustavljanju <i>ZNP</i> modula. Operacija ne sadrži ulazne parametre, a povratna vrijednost definira poruku o uspješnosti zaustavljanja modula.	
	Ulagni parametri	-
	Izlazni parametri	<ul style="list-style-type: none"> • string <i>Message</i> – upravljačka poruka kojom se opisuje uspješnost zaustavljanja ZNP modula