UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

David Jelenc

Obvladovanje zaupanja v globalnih porazdeljenih sistemih

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Denis Trček

Ljubljana, 2009

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01578/2009
Datum: 01.09.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **DAVID JELENC**

Naslov: **OBVLADOVANJE ZAUPANJA V GLOBALNIH PORAZDELJENIH SISTEMIH**

TRUST MANAGEMENT IN GLOBAL DISTRIBUTED SYSTEMS

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

V diplomskem delu obdelajte problematiko obvladovanja zaupanja v sodobnih storitveno usmerjenih arhitekturah. Izhajajte iz pregleda teoretičnih osnov modeliranja zaupanja, nato izvajanja nadgradite s problematiko ugleda. V osrednjem delu se osredotočite na arhitekture in tehnologije, ki omogočajo obvladovanje zaupanja in ugleda. Poudarek dajte na povezovanju trenutno centralizirane lastne rešitve za obvladovanje zaupanja, imenovane trustGuard tako, da bomo dobili porazdeljeno rešitev, ki bo uporabna v globalnem okolju. Pri tem analizirajte najprimernejše arhitekturne in tehnološke pristope, to je arhitekture strežnik-odjemalec in vsak z vsakim, podajte prednosti in slabosti posameznih pristopov ter vse skupaj realizirajte v okolju spletnih storitev.

Mentor:

prof. dr. Denis Trček

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Franc Solina

Namesto te strani **vstavite** original izdane teme diplomskega dela s podpisom mentorja in dekana ter ˇžigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani David Jelenc,

z vpisno številko 6040057,

sem avtor diplomskega dela z naslovom:

Obvladovanje zaupanja v globalnih porazdeljenih sistemih

S svojim podpisom zagotavljam, da:

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

V Ljubljani, dne 9. 9. 2009 Podpis avtorja:

Zahvala

To delo zagotovo ne bi nastalo, če ne bi bil deležen velike vzpodbude in podpore s strani staršev. Pa ne le v času izdelave diplomske naloge, temveč tudi tekom celotnega študija na Fakulteti za računalništvo in informatiko. Ovir na poti do cilja ni bilo malo, niti niso bile majhne. Z vajino pomočjo mi je uspelo. Iskrena hvala.

Ne smem pozabiti na sošolce in ostale prijatelje, ki ste me spremljali med študentskimi leti. Pa naj si gre za to, da ste me motili, bodrili, spodbujali ali pustili pri miru, ko se tako želel. Brez vas tega ne bi pisal.

Posebno zahvalo moram nameniti mentorju, prof. dr. Denisu Trčku, ki ne samo, da me je s svojimi strokovnimi komentarji in skrbnimi popravki vodil pri izdelavi diplomskega dela, ampak me je še mnogo pred njegovim nastankom navdušil za področje obvladovanja zaupanja in tako poskrbel, da motivacije za delo ni manjkalo.

Zahvala gre tudi ostalim kolegom iz Laboratorija za e-medije za vse konstruktivne pripombe in nasvete; dr. Damjanu Kovaču, ki me je odlično zalagal s strokovnimi članki in mi s svojo disertacijo olajšal pregled nad obstoječimi pristopi obvladovanja zaupanja, asistentu Iztoku Starcu, ki mi je s postavitvijo ustrezne računalniške infrastrukture izredno olajšal izdelavo praktičnega kot tudi teoretičnega dela diplome, ter kolegici Evi Zupančič za zanimive in uporabne napotke pri izbiri strokovnih člankov ter uporabne nasvete za delo v okolju LaTeX.

Kazalo

Seznam uporabljenih kratic in simbolov

API	Application Programming Interface (programski vmesnik)
CA	Certificate Authority (overitelj digitalnih potrdil)
CIA	Confidelity, Integrity, Availability (zauponost, celovitost, razpoložljivost)
CORBA	Common Object Request Broker Architecture
CRL	Certificate Revocation List (seznam preklicanih certifikatov)
DCOM	Distributed Component Object Model
DTG	Distributed trustGuard
GPG	GNU Privacy Guard
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
OZ	Upravljanje zaupanja
OU	Upravljanje ugleda
P2P	peer-to-peer (vsak-z-vsakim)
PGP	Pretty Good Privacy (razmeroma dobra zasebnost)
PKI	Public Key Infrastructure (infrastruktura javnih ključev)
RMI	Remote Method Invocation
SOAP	Simple Object Access Protocol
SPOF	Single Point of Failure (kritična točka izpada)
SSKJ	Slovar slovenskega knjižnega jezika
SUPB	Sistem za upravljanje s podatkovno bazo
UDDI	Universal Description Discovery and Integration
W3C	World Wide Web Consortium
WSDL	Web Service Description Language
XML	eXtensible Markup Language
ZVOP	Zakon o varstvu osebnih podatkov

Povzetek

Naloga obravnava področje obvladovanja zaupanja v globalnih porazdeljenih sistemih. V uvodnih poglavjih podamo definicijo zaupanja in ugleda, njunega razlikovanja ter opredelimo njuno razmerje do pojma varnosti. Opredelimo pojem obvladovanja zaupanja ter ugleda, si ogledamo osnovne arhitekturne pristope pri izgradnji takšnih sistemov in opravimo pregled prevladujočih kvantitativnih in kvalitativnih modelov za upravljanje zaupanja. Nadaljujemo z analizo arhitekture odjemalec-strežnik in arhitekture vsak-z-vsakim (P2P), njunih prednosti in omejitev ter orišemo koncept spletnih storitev.

Osrednji del zajema diskusijo o načinih integracije globalno porazdeljenih sistemov. Opredelimo centraliziran sistem kot izpeljanko iz modela odjemalec-strežnik, ter sistem vsak-z-vsakim, ki izhaja iz arhitekture P2P. Izpostavimo njune omejitve in prednosti ter navedemo okoliščine, kdaj je kateri model primerno uporabiti. Posebna pozornost je namenjena konceptu predpomnenja kot načina pohitritve delovanja posameznega sistema, izboru identifikatorja, ki enolično določa entiteto tudi v povezanem sistemu, ter možnim načinom izmenjave ocen med sistemi.

Ključne besede:

zaupanje, ugled, varnost, globalno porazdeljeni sistemi, odjemalec-strežnik, vsak-z-vsakim, spletne storitve, predpomnenje, enolični identifikator, izmenjava ocen, integracija.

Abstract

This work tackles the barriers of managing trust in globally distributed systems. In the beginning we define terms of trust and reputation, the difference between them, and describe how they relate to security. Later on we present the definitions that are needed for trust and reputation management, analyse the basic architectural approaches for constructing trust managing systems, and give an overview of quantitative and qualitative trust models. We continue with analysing client-server and peer-to-peer (P2P) architectures, stating their advantages and weak points and describing web services in brief.

The core of this work is an analysis of the problem, which is the best approach to integrate globally distributed trust management systems. We construct a centralised system, that derivates from client-server model, and a P2P system, which originates in P2P model. For each of them, we provide their strong points and weaknesses, and recommend the type of environment they should be used in. Caching, as means of improving performance, is also discussed, alongside selecting the appropriate entity identification, which should be unique even in integrated systems. At the end, we present two distinct strategies of exchanging assessments between systems.

Key words:

trust, reputation, security, globally distributed systems, client-server, peer-to-peer, web service, caching, unique entity identification, exchanging assessments, integration.

Poglavje 1

Uvod

1.1 Motivacija in cilji

Sistemi za upravljanje zaupanja so sistemi, ki hranijo podatke o ocenjeni kakovosti medsebojnih interakcij med entitetami, ki jih le-te podajajo druga o drugi. Te ocene predstavljajo vhod za izračun stopenj zaupanja in ugleda, ki predstavljajo sodilo za odločitve ostalih entitet ali vstopiti v interakcijo z izbrano entiteto tj. ali je ta entiteta zaupanja vredna. Tipičen primer takega sistema je spletno mesto eBay[1] za elektronsko licitiranje in trgovanje (angl. bidding and e-commerce), ki ima zelo učinkovit model za obvladovanje zaupanja. V njem kupec po koncu vsake transakcije oceni prodajalca (in obratno), ali je kupljeno blago v skladu z obljubljenim oz. ali je bilo plačilo korektno izvedeno. Ta ocena je nato osnova za odločanje ostalih uporabnikov ali s tem prodajalcem (in kupcem) poslovati.

V želji, da bi izračunana ocena stopnje zaupanja čim točneje izražala dejansko stanje, ti sistemi potrebujejo čim več vhodnih podatkov (tj. podanih ocen). Svojo natančnost tako utemeljujejo s kvantiteto, kar se sklada s splošnim dojemanjem načina ugotavljanja zaupanja; več kot imamo izkušenj, podatkov, mnenj o subjektu, lažje sprejmemo odločitev. Zapisano bi lahko povzeli tudi z besedami slovenskega pregovora: *"Več glav več ve."*

Sistemi za upravljanje zaupanja zbirajo ocene o interakcijah med entitetami znotraj svojega območja delovanja, medtem ko delovanje entitet ni tako omejeno. Te iste entitete lahko nastopajo tudi v drugih (podobnih) sistemih. Ni si težko zamisliti uporabnika, ki nastopa v vlogi prodajalca hkrati na spletišču eBay in Amazon[2]. Pričakovali bi, da bo izračunana stopnja

[1] http://www.ebay.com
[2] http://www.amazon.com

zaupanja v uporabnika v obeh sistemih enaka, vendar se lahko – ravno zaradi prej omenjenih mej delovanja sistemov – oceni zaupanja iste entitete celo diametralno razlikujeta. Glede na zapisano iz prejšnjega odstavka lahko trdimo, da bi bila izračunana stopnja zaupanja, ki bi upoštevala podane ocene o uporabniku iz obeh sistemov, bolj natančna.

Tako smo prišli do srža problematike, ki ga to diplomsko delo obravnava; kako učinkovito povezati različne globalno porazdeljene sisteme za upravljanje zaupanja, da bo izračunana stopnja zaupanja posamezne entitete znotraj izbranega sistema ne le odraz ocen podanih znotraj meja tega sistema, temveč tudi ocen, ki jih je ta entiteta pridobila v drugih sistemih. Tako izračunana stopnja zaupanja po našem mnenju natančneje odraža dejanskost in je lahko boljše merilo za odločanje.

V sklopu naloge je bila izdelana tudi spletna aplikacija *Distributed trustGuard*, ki realizira eno od v nadaljevanju podanih rešitev.

1.2 Pregled vsebine

Prvo poglavje ima namen bralca seznaniti s širšim področjem dela in z motivi za delo.

V drugem poglavju je narejen pregled konceptov zaupanja, ugleda, upravljanja zaupanja in ugleda, gradnje sistemov za upravljanje zaupanja in ugleda ter opravljen pregled tradicionalnih in modernih modelov za obvladovanje zaupanja.

V tretjem poglavju je predstavljena analiza porazdeljenih računalniških sistemov, in sicer modela odjemalec-strežnik ter modela vsak z vsakim. Nato nadaljujemo s konceptom spletnih storitev.

V četrtem poglavju sta predstavljena centraliziran sistem integracije globalno porazdeljenih sistemov in sistem vsak-z-vsakim. Izdatna pozornost je namenjena obravnavi predpomnenja, določitvi enoličnega identifikatorja ter osvetlitvi problema izmenjevanja ocen. Prav tako je med obravnavani vprašanji govora o realizirani aplikaciji *Distributed trustGuard*.

Peto poglavje predstavlja zaključek dela, kjer so strnjene ugotovitve in podane smernice nadaljnjega dela.

Dodatek je namenjen podrobnejši predstavitvi realizirane aplikacije.

Poglavje 2

Zaupanje in ugled

2.1 Splošno o zaupanju in ugledu

Zaupanje (angl. trust) je ključna komponenta vsake transakcije pri e-poslovaju [1]. Velikokrat mora entiteta A zaupati entiteti B, če želi z njo stopiti v interakcijo – kupec npr. mora zaupati prodajalcu pri elektronskem nakupu. Splošno velja, da bolj kot so udeležene strani v transakciji časovno in prostorsko ločene, večje je tveganje. Tako se transakcija ne izvrši, vse dokler stran, ki ima moč začeti transakcijo, ne poseduje določene stopnje zaupanja o nasprotni strani, da bo ta resnično izpolnila svoje obveze.

Zaupanje je tesno povezano s pojmom ugleda (angl. reputation), ki se navezuje na zanesljivost in posredno določa stopnjo zaupanja na osnovi priporočil in ocen članov skupnosti. Takšna ocena pa ima lahko konkretne ekonomske učinke – ponudniki e-storitev lahko s pomočjo sistemov za upravljanje ugleda izboljšajo kakovost svoje ponudbe. Tipična primera sta v uvodu omenjeni spletišči eBay ter Amazon.

2.2 Zaupanje in varnost

Zaupanje je v tesni povezavi s pojmom varnosti (angl. security). Vendar se je potrebno zavedati, da varnost sama po sebi še ne zagotavlja zaupanja; varnost je praviloma potreben, ne pa zadosten pogoj.

Varnost je v standardih definirana kot mehanizem za zagotavljanje *zaupnosti, celovitosti* in *razpoložljivosti* informacij (angl. Confidelity, Integrity, Availability, CIA [10]). Glavna naloga informacijske varnosti je zagotavljati lastnosti CIA za informacijske vire znotraj določene domene. CIA se zagotavlja

z uporabo različnih varnostnih storitev, temelječih na varnostnih mehanizmih. To lahko označimo kot tradicionalne varnostne mehanizme, katerih namen je varovanje informacij informacijskih virov pred zlonamernimi uporabniki preko mehanizmov omejevanja dostopa le avtoriziranim uporabnikom. Ponudnik informacijskega vira lahko s pomočjo varnostne politike določi množico uporabnikov za izvajanje določenih akcij na ponujenem informacijske viru.

Mnogokrat pa se pojavi obratna situacija, kjer se morajo uporabniki varovati pred ponudniki virov, informacij in storitev, saj lahko ti delujejo zlonamerno in ponujajo napačne in zavajajoče informacije. V zgornjem odstavku opisani varnostni mehanizmi pri taki vrsti groženj odpovejo, medtem ko sistemi za upravljanje zaupanja in ugleda lahko varujejo tudi pred takimi grožnjami. Na tem primeru se pokaže bistvena razlika med varnostnimi mehanizmi in sistemi za upravljanje zaupanje in ugleda. Razliko med tema dvema pristopoma k pojmu varnosti nekateri označujejo s pojmoma *trdna* varnost (angl. hard security) za tradicionalne sisteme in *mehka* varnost (angl. soft security) za socialne mehanizme kot so sistemi za upravljanje zaupanja in ugleda. Vendar poglavitni razlog, ki ločuje zaupanje od varnosti, tiči v dejstvu, da je zaupanje močno subjektiven pojem in vključuje tudi sociološki in psihološki vidik.

2.3 Definicija zaupanja in ugleda

Slovar slovenskega knjižnega jezika (SSKJ [12]) pravi, da je zaupanje *prepričanje, da je kdo sposoben, voljen narediti, kar se pričakuje.* Angleški slovar Oxford Reference Dictionary opredeljuje zaupanje kot *trdno prepričanje v zanesljivost, iskrenost, sposobnost osebe ali stvari.* Ta definicija je le ena v vrsti mnogih, ki obstajajo v literaturi. Domeni e-poslovanja pa se najbolj prilega definicija iz [4], ki pravi, da je *"zaupanje entitete A v entiteto B za storitev X merljivo prepričanje, ki ga ima A v B za zanesljivo delovanje znotraj določenega časovnega okvira v določenem kontekstu v relaciji do storitve X."*

Formalno predstavimo zaupanje kot usmerjeno relacijo med dvema entitetama – izvorno in ciljno [2]; izvorno entiteto poimenujmo *upnik* (angl. trustor) ponorno pa *zaupnik* (angl. trustee). Upnik poseduje neko stopnjo zaupanja v ciljno entiteto, zaupnika, da lahko izvede ustrezno nalogo v določenem kontekstu. Predpostavka je, da je izvorna entiteta razmišljujoča ali razumna (angl. congnitive) – ima zmožnost ocenjevanja in odločanja o ciljni entiteti na podlagi prejetih informacij in preteklih izkušenj. Ciljna entiteta je lahko abstrakten pojem – oseba, organizacija, računalnik, itd – ali razumna entiteta.

Kontekst zaupanja pomeni specifičen namen ali področje, kot je npr. poštenost pri prodajanju blaga preko spleta. O vzajemnem zaupanju (angl. mutual trust) lahko govorimo takrat, ko dve entiteti zaupata druga drugi znotraj istega konteksta. Seveda pa morata obe biti razumni entiteti.

Ugled (angl. reputation) je tesno povezan s stopnjo zaupanja v ciljno entiteto. Slovar Concise Oxford Dictionary opredeljuje ugled kot *"tisto, kar se običajno govori ali verjame o značilnosti osebe ali stvari"*. Ta definicija se sklada s trditvijo, da je ugled pridobljena lastnost socialne mreže in je javno vidna vsem njenim članom [7]. Socialna mreža (skupnost) se označuje kot socialna struktura vozlišč, ki so običajno posamezniki ali organizacije. Med vozlišči obstajajo povezave, ki predstavljajo različne tipe relacij.

Povezavo med zaupanjem in ugledom je težko natančno definirati, predvsem zato, ker temelji zaupanje na osebnem in subjektivnem odnosi do ciljne entitete. To je razvidno iz sledečih izjav:

(a) *"Zaupam ti, ker imaš dober ugled."*

(b) *"Zaupam ti, čeprav imaš slab ugled."*

Prva pravi, da je zaupanje pogojeno z ugledom ciljne entitete, druga pa da obstaja zaupanje slabemu ugledu ciljne entitete navkljub – torej mora obstajati subjektivno znanje izvorne entitete o ciljni entiteti. To je lahko posledica osebne izkušnje med entitetama in ima večjo težo kot njen ugled. Če osebne izkušnje ni, se zaupanje lahko oblikuje na podlagi ugleda ali priporočil drugih.

2.4 Obvladovanje zaupanja in ugleda

Entitete, ki nastopajo v okoljih e-poslovanja so razpršene v prostoru in največkrat nimajo neposrednega medsebojnega stika in izkušenj. Obstajati mora ustrezen mehanizem – sistem za vzpostavitev razmerij (relacij) zaupanja. Lahko govorimo o sistemih za obvladovanje zaupanja (OZ) in sistemih za obvladovanje ugleda (OU), ki so primer t. i. "mehkega" varnostnega mehanizma oz. "mehke" varnosti.

Obvladovanje zaupanja pomeni zbiranje potrebnih informacij za vzpostavitev relacij zaupanja ter dinamičen nadzor in prilagajanje obstoječih relacij zaupanja med entitetami [1]. Taki sistemi morajo imeti naslednje značilnosti:

(1) Entitete morajo biti v sistemu prisotne daljši čas.

(2) Ocene o trenutnih interakcija se shranijo na eno ali več lokacij.

(3) Ocene preteklih interakcij so vodilo za odločitve o trenutnih interakcijah.

Značilnost (1) pomeni, da entiteta ne more spremeniti svoje identitete, saj bi tako prekinila vez s preteklim obnašanjem. Za lastnost (2) je potrebna pripravljenost entitet, da podajo svoje ocene preko ustreznega protokola za ocenjevanje. Lastnost (3) pa določa, za kaj se bo tak sistem v različnih okoljih uporabil.

2.5 Razlike med sistemi za upravljanje zaupanja in upravljanje ugleda

Glavna razlika med tema dvema vrstama sistemov je v tem, da pri zaupanju sistem vrne rezultat, ki je odraz subjektivnega pogleda na stopnjo zaupanja izbrane entitete – govorimo o *privatni informaciji* kot odraz neposredne izkušnje – pri ugledu pa sistem vrne rezultat, ki je odraz celotne mreže in je javno dostopen (*javna informacija*, pridobljena od tretjih entitet).

Potrebno je še poudariti, da obstajajo sistemi, ki vsebujejo elemente obeh, tako da ločnica ni vedno jasno vidna. Skupna točka obeh vrst sistemov je podpora odločanju.

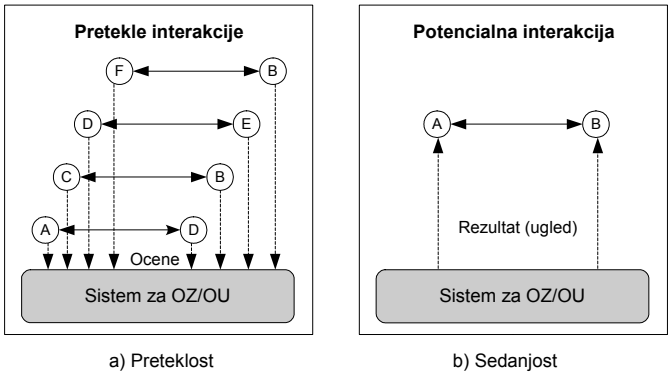
2.6 Arhitektura sistemov za upravljanje zaupanja in ugleda

Arhitektura določa način širjenja ocen in rezultatov med člani mreže. Obstajata dva glavna tipa arhitekture; *centraliziran* in *porazdeljen*. V obeh primerih sta pomembna dva vidika:

- *Komunikacijski protokol*, ki določa način podajanja ocen med člani skupnosti kot tudi način pridobivanja rezultatov o posameznih entitetah.
- *Algoritem za izračun* ugleda oz. stopnje zaupanja, temelji na dobljenih ocenah in lahko tudi na drugih informacijah.

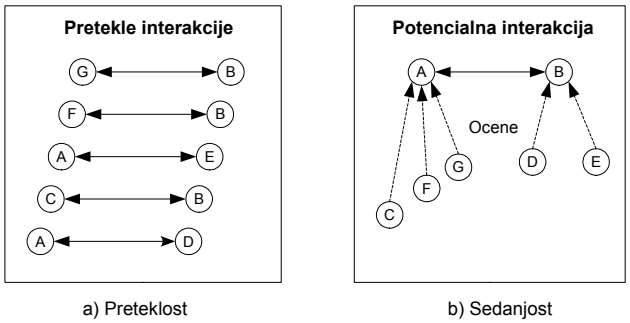
Centraliziran sistem (slika 2.1, vir [1]) shranjuje vse ocene na eno centralno lokacijo ter vrača javno dostopen rezultat za vsako entiteto. Po vsaki interakciji sistem sprejme ocene vseh udeleženih entitet in ažurira rezultat – ugled kot funkcijo sprejetih ocen. Ta je ves čas javno dostopen vsem članom skupnosti. Centralizirani sistemi so najbolj pogosto komercialno uporabljeni. Taka sta

tudi naša primera eBay ter Amazon. Slabost takih sistemov je majhna skalabilnost in lahko odpovejo v primeru prevelikega števila interakcij.



Slika 2.1: Centraliziran sistem.

V porazdeljenem okolju (slika 2.2, vir [1]) ni neke centralne lokacije, ki bi shranjevala ocene in omogočala pridobivanje rezultata ugleda za posamezno entiteto. Namesto tega se uporabljajo porazdeljene lokacije za shranjevanje ocen; te so lahko tudi na samih vozliščih omrežja tj. entitetah samih. Tako entitete lokalno, *pri sebi* shranujejo informacije o zgodivini interakcij z drugimi entitetami. Entiteta, ki želi v interakcijo z določeno ciljno entiteto, mora nati ustrezne porazdeljene lokacije oz. pridobiti ocene od čimveč članov skupnosti, ki so imeli neposredno izkušnjo s ciljno entiteto. Porazdeljen sistem je bistveno bolj kompleksen od centraliziranega. Zelo pomemben je učinkovit komunikacijski protokol za porazdeljene poizvedbe in ažuriranje ocen. Prednost takega sistema je v skalabilnosti in robustnosti pri velikem številu entitet oz. agentov.



Slika 2.2: Porazdeljen sistem.

Porazdeljeni pristop se uporablja pri sistemih P2P. Entitete hkrati nastopajo kot odjemalci in strežniki. Cilj sistema OU v omrežju P2P je ugotoviti, katera entiteta je najbolj zanesljiva pri ponujanju kakovostnih storitev in virov ter katera entiteta ponuja najbolj zanesljive informacije. V porazdeljenih okoljih tipa P2P je vsaka entiteta odgovorna za zbiranje in agregacijo ocen drugih entitet. Vendar je zaradi narave porazdeljenega omrežja ponavadi nemogoče ali časovno prezahtevno pridobiti ocene vseh interakcij z izbrano entiteto. Zaradi tega se ugled izračuna samo na podmnožici ocen, ponavadi na lokalni soseščini izvorne entitete.

2.7 Tradicionalni sistemi obvladovanja zaupanja

V tem razdelku si bomo ogledali nekatere tipične obstoječe sisteme za obvaldovanje zaupanja, ki jih lahko označimo kot trdne varnostne mehanizme, saj ne vključujejo sociološke in psihološke komponente zaupanja. Osredotočajo se na izvajanje varnostnih storitev v specifičnem kontekstu; nekateri uporabljajo ustrezne jezike za opis politik zaupanja, ki določajo zahteve in kriterije za ciljne entitete, da so zaupanja vredne. Prvi tradicionalni sistemi so bili sistemi za nadzor dostopa.

Vsi ti sistemi ne upoštevajo časovne dinamike in psihološko-socialne dimenzije zaupanja, ampak le tehnološko, zato jih lahko označimo kot *trde sisteme*. Velja omeniti nekatere tradicionalne sisteme OZ, katerih večina temelji na infrastrukturi javnih ključev (angl. Public Key Infrastructure, PKI). PKI temelji na principu digitalnih potrdil, ki so digitalno podpisana s strani pooblaščenih overoviteljev (angl. Certificate Authority). Digitalno potrdilo preko overovitelja povezuje identiteto uporabnika in njegov javni ključ in tako predstavlja elektronsko identifikacijo. Pri PKI gre za tehnično in organizacijsko strukturo procesa, ki zajema izdajanje in preklicevanje digitalnih potrdil. Z vzpostavljeno infrastrukturo PKI je možno digitalno podpisovanje, enkripcija sporočil, overjanje in vzpostavitev mreže zaupanja med lastniki digitalnih potrdil.

Omenimo še TRUSTe, ki je izjema, saj upošteva samo sociološko dimenzijo zaupanja in ima glavni namen pospeševati ter povečati e-poslovanje. TRUSTe ni model zaupanja ampak zbirka uporabnih smernic, priporočil in prakse na področju varnosti spletnih aplikacij in mobilnih naprav. Njen glavni cilj je zmanjšati ranljivost in tveganja znotraj področja varnosti ter posledično povečati vrednost zaupanja med sodelujočimi entitetami znotraj e-poslovanja.

2.7.1 X.509

X.509 je hierarhičen model za podporo zaupanja preko overjanja. Vsaka entiteta mora imeti digitalno potrdilo, ki je podpisano s strani vrhovnega overitelja (angl. CA root) ali druge entitete, ki je overjena s strani vrhovne CA posredno ali neposredno. Ta model podpira zaupanje v obliki hierarhičnega drevesa; zaupanje med dvema entitetama obstaja, če med njima obstaja pot skozi graf preko vrhovne CA. Govorimo o t. i. *verigi overjanja*. Obstajajo različni algoritmi za iskanje poti in računaje stopnje zaupanja skozi verigo overjanja. X.509 je standard za PKI, poleg tega določa tudi format in strukturo digitalnih potrdil in seznam preklicanih potrdil (angl. Certificate Revocation List – CRL).

2.7.2 PGP

PGP (angl. Pretty Good Privacy) je model za podporo zaupanja preko overjanja, a se od X.509 razlikuje po tem, da nima hierarhične strukture. Največkrat se uporablja pri e-pošti med uporabniki interneta. Model predvideva, da vsaka entiteta lahko digitalno podpiše javne ključe in tako nastopa tudi v vlogi CA. Model tako implicira zaupanje entitete v lastni javni ključ. Zaupanje nastopa v treh vidikih, in sicer kot *zaupanje v lastnika* (stopnja zaupanja v lastnika javnega ključa), *zaupanje v podpis* (stopnja zaupanja v prejeto digitalno potrdilo) in zaupanje v *veljavnost ključa* (stopnja zaupanja v javni ključ). Slabost tega modela je v tem, da nima dobro definiranih mehanizmov za ustvarjanje in pridobivanje ter distribucijo potrdil, zato ni primeren za okolja e-poslovanja, ampak za osebno komunikacijo. Omenimo še, da je PGP komercialen produkt, njegovo odprtokodno različico pa predstavlja GPG[1].

2.7.3 PolicyMaker

PolicyMaker je sistem, osnovan na modelu X.509, ki določa dovoljene akcije v povezavi z javnim ključem. Identiteta uporabnika je povezana z javnim ključem preko digitalnega potrdila. Sistem preveri ali je zahtevana akcija v skladu z lokalno politiko. Lokalna politika je množica trditev za zaupanje. PolicyMaker uporablja pojem poverilnic, ki so digitalno podpisane trditve entitet. Trditev je lahko politika ali poverilnica. Vhod v sistem so lokalna politika, prejete poverilnice in imena akcij, ki jih javni ključ lahko izvede.

[1]GNU Privacy Guard – http://www.gnupg.org

Rezultat, ki ga sistem vrne, je logična vrednost ali seznam omejitev, da se akcija lahko izvrši.

2.7.4 KeyNote

KeyNote je naslednik PolicyMaker-ja in je bil razvit z namenom, da izboljša njegove pomanjkljivosti. Ima enak princip trditev in poizvedb z dvema prednostnima – standardizacija in lažja integracija. V primerjavi s sistemom PolicyMaker ima sistem KeyNote večino funkcij izvedenih v samem jedru. Uporablja samo eno različico jezika za opis storitev (PolicyMaker jih lahko uporablja več), kar omogoča lažjo integracijo s strojem za preverjanje skladnosti.

2.8 Modeli zaupanja

Naša osnovna klasifikacija loči *kvalitativn*e in *kvantitativne* modele za izračun stopnje zaupanja. V splošnem se stopnja lahko izračuna na podlagi lastnih izkušenj (zasebna informacija), priporočil drugih (javna informacija) ali kot kombinacija obojega. V nadaljevanju sledi prikaz nekaterih modelov z osnovnimi značilnostmi. Nekateri od njih so zgolj akademski, drugi pa so implementacijo doživeli v komercialnih aplikacijah. Tako poleg omenjenih *eBay* ter *Amazon* poznamo še *Epinions, BizRate, Slashdot ...* V implementiranih modelih prevladuje centralizirana arhitektura.

2.8.1 Kvantitativni modeli

Pri teh modelih je stopnja zaupanja podana preko kvantitativne vrednosti. Sledijo različne razvrstitve glede na uporabljene algoritme za izračun.

Modeli z enostavnimi vsotami in povprečji

Ti modeli predstavljajo najenostavnejše pristope pri izračunu stopnje zaupanja in ugleda. V osnovi gre za seštevanje pozitivnih in odštevanje negativnih ocen. Obstajajo različne variacije tega modela. Tako nekateri izračunavajo ponderirana povprečja glede na ugled ocenjevalca, starost podanih ocen, razlike med oceno in trenutnim rezultatom in podobno.

Te modele uporabljajo zlasti spletna mesta za elektronske licitacije in trgovanje (*eBay, Amazon*). *eBay* sestavlja množica kupcev in prodajalcev. Pri vsaki izvedeni transakciji kupec oceni prodajalca in obratno z oceno 1, 0 in

-1. Ugled uporabnika se izračuna kot skupek vseh ocen, dobljenih v zadnjih 180 dneh. Novi uporabniki dobijo ugled enak 0. Na spletišču *Amazon* uporabniki ocenjujejo drug drugega z ocenami med 1 in 5. Ugled se izračuna kot povprečje vseh dobljenih ocen v celotnem času delovanja sistema.

Verjetnostni modeli

Verjetnostni modeli predvidevajo obnašanje entitete v prihodnosti na podlagi informacij o preteklem obnašanju. Temeljijo na verjetnostnih in statističnih metodah. Predvidevanje določenega obnašanja entitete je izraženo kot verjetnost. Verjetnost pomeni kakšna je stopnja zaupanja oz. ugleda izbrane entitete. Nova – *aposteriorna* (angl. posterior) vrednost se izračuna na podlagi prejšnjih – *apriornih* (angl. prior) vrednosti in novo podane ocene. Ločimo *binomske* (t. i. dvovrednostne) modele za možnost ocenjevanja preko dveh vrednosti (npr. dobro, slabo obnašanje) ter *multinomske* (t. i. večvrednostne) modele za možnost ocenjevanja z več vrednostmi oz. stopnjami (npr. slabo, povprečno, dobro, odlično). Prednost verjetnostnih modelov je, da izračun stopnje zaupanja in ugleda temelji na matematični teoriji, slabost pa, da so prekompleksni za razumevanje s strani običajnega uporabnika. Primer verjetnostnih model predstavljajo *Bayesovi modeli* [1].

Modeli prepričanja

Modeli prepričanja temeljijo na teoriji verjetnosti. Vsota verjetnosti čez vse možne izide ni nujno enaka 1, ostanek verjetnosti se označuje kot *negotovost* (angl. uncertainty). Primer je Dempster-Shaferjeva teorija evidence.

Dempster-Shaferjeva teorija služi kot osnova za Jøsangovo *subjektivno logiko* [5]. *Mnenje* (angl. opinion) je definirano kot $\omega_p^A = (b, d, u)$ in določa prepričanje entitete A v resničnost trditve p; b pomeni stopnjo *zaupanja* (angl. belief), d stopnjo *nezaupanja* (angl. disbelief) in u stopnjo *nedoločenosti* (angl. uncertainty), kjer velja $b + d + u = 1$.

Subjektivna logika vsebuje okoli 10 različnih operatorjev[5], tu predstavljamo le najpomembnejše; *konjunkcijo*, *konsenz* in *priporočilo*:

(a) **Konjunkcija.** Naj bosta $\omega_p^A = (b_p^A, d_p^A, u_p^A)$ in $\omega_q^A = (b_q^A, d_q^A, u_q^A)$ mnenji entitete A o dveh različnih trditvah p in q. Potem konjunkcija predstavlja

skupno mnenje A o trditvah p in q in je definirana kot:

$$\omega^A_{p\wedge q} = \omega^A_p \wedge \omega^A_q = \{b^A_{p\wedge q}, d^A_{p\wedge q}, u^A_{p\wedge q}\} \text{ kjer } \begin{cases} b^A_{p\wedge q} = b^A_p b^A_q \\ d^A_{p\wedge q} = d^A_p + d^A_q - d^A_p d^A_q \\ u^A_{p\wedge q} = b^A_p u^A_q + u^A_p b^A_q + u^A_p u^A_q \end{cases} \tag{2.1}$$

Konjunkcija je komutativna in asociativna in zahteva statistično neodvisne argumente.

(b) **Konsenz.** Naj bosta $\omega^A_p = (b^A_p, d^A_p, u^A_p)$ in $\omega^B_p = (b^B_p, d^B_p, u^B_p)$ mnenji entitet A in B o isti trditvi p. Potem konsenz predstavlja mnenje kompozitne entitete $[A, B]$ o trditvi p:

$$\omega^{A,B}_p = \omega^A_p \oplus \omega^B_p = \{b^{A,B}_p, d^{A,B}_p, u^{A,B}_p\} \text{ kjer } \begin{cases} b^{A,B}_p = \frac{b^A_p u^B_p + b^B_p u^A_p}{u^A_p + u^B_p - u^A_p u^B_p} \\ d^{A,B}_p = \frac{d^A_p u^B_p + d^B_p u^A_p}{u^A_p + u^B_p - u^A_p u^B_p} \\ u^{A,B}_p = \frac{u^A_p u^B_p}{u^A_p + u^B_p - u^A_p u^B_p} \end{cases} \tag{2.2}$$

Konsenz je komutativen in asociativen in zahteva statistično neodvisne argumente. Namen operatorja je zmanjšati negotovost. Dodajmo še, da dogmatičnih mnenej – mnenj, kjer je $u = 1$ – ni mogoče združiti s konsenzom. To pa zaradi tega, ker je konsenz definiran ko *dovzetnost za sprejemanje različnih mnenj* (angl. room for influence). Nemogoče se zdi združiti dve nasprotujoči si in dogmatski trditvi.

(c) **Priporočilo.** Naj bosta A in B dve entiteti, kjer je $\omega^A_B = \{b^A_B, d^A_B, u^A_B\}$ mnenje entitete A o priporočilih entitete B in naj bo p trditev, kjer je $\omega^B_p = \{b^B_p, d^B_p, u^B_p\}$ mnenje entitete B o trditvi p. Potem je mnenje entitete A o trditvi p rezultat priporočila entitete B in je definirano kot:

$$\omega^{AB}_p = \omega^A_p \otimes \omega^B_p = \{b^{AB}_p, d^{AB}_p, u^{AB}_p\} \text{ kjer } \begin{cases} b^{AB}_p = b^A_B b^B_p \\ d^{AB}_p = b^A_B d^B_p \\ u^{AB}_p = d^A_B + u^A_B + b^A_B u^B_p \end{cases} \tag{2.3}$$

Operator za priporočilo je asociativen, ni pa komutativen. Zato se operator lahko uporabi samo na začetku ali na koncu verige mnenj. Pomemben je vrstni red kombiniranja mnenj. V verigi, kjer obstaja več kot ena entiteta s priporočilom, je pomembno, da so mnenja neodvisna, kar pomeni, da se ista entiteta v verigi mnenj ne sme pojaviti več kot enkrat. Pomembno je omeniti, da je uporaba operatorja smiselna samo ob predpostavki, da je priporočilo tranzitivno, kar pomeni, da entitete v verigi ne spreminjajo vrednosti oz. obnašanja.

Prednost subjektivne logike je močan formalizem za računanje sestavljenih mnenj. Operator za priporočilo se npr. uporablja za izračun stopnje zaupanja v verigah in grafih za overjanje (npr. model X.509). Slabost subjektivne logike je v tem, da predvideva racionalne entitete, ki znajo ustrezno določati vrednosti (b, d, u). Poleg tega je razumevanje operatorjev za povprečnega uporabnika pretežko, saj je pogojeno z razumevanjem verjetnosti in logike.

2.8.2 Kvalitativni model

Ljudje raje ocenjujemo stvari z diskretnimi kot pa z zveznimi merami. Raje povemo, da subjektu *zaupamo*, kot da mu *70% zaupamo, 10% ne zaupamo, v preostalih 20% pa smo negotovi*. Pri kvalitativnih modelih je stopnja zaupanja (tudi ugled) do ciljne entitete določena s kvalitativno opisno vrednostjo iz vnaprej določene diskretne množice. Ponavadi obstaja tudi ustrezna preslikava v kvantitativne (numerične) vrednosti za ustrezne računske algoritme; ti lahko vračajo kvalitativni ali kvantitativni rezultat. V večini primerov pa se za izračun uporablja hevristika. [1]

Kvalitativni model [2] izhaja iz psihološko-sociološke dimenzije zaupanja. Relacija zaupanja med izvorno entiteto A in ciljno entiteto B je predstavljena kot relacija $\omega_{A,B}$, stopnja zaupanja pa je predstavljena kot ocena (mnenje), ki ima lahko sledeče kvalitativne vrednosti s pripadajočimi numeričnimi vrednostmi:

- *zaupanja vreden* (1),
- *neodločen* (0) in
- *zaupanja nevreden* (-1).

Za relacijo zaupanja predpostavimo, da:

(1) ni refleksivna ($\omega_{A,A}$ ne obstaja vedno),

(2) je asimetrična (v splošnem $\omega_{A,B} \neq \omega_{B,A}$) in

(3) ni tranzitivna (če obstaja $\omega_{A,B}$ in $\omega_{B,C}$, ni nujno da obstaja $\omega_{A,C}$).

Zaupanje v socialni mreži (skupnosti), ki vsebuje n entitet, je predstavljeno z *matriko zaupanja* M velikosti $n \times n$. Posamezen element $\omega_{i,j}$ določa stopnjo

zaupanja entitete i v entiteto j. Znak "–" pomeni, da relacija zaupanja ne obstaja ali pa ni poznana.

$$M = \begin{bmatrix} \omega_{1,1} & \omega_{1,2} & \cdots & \omega_{1,n} \\ \omega_{2,1} & \omega_{2,2} & \cdots & \omega_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{n,1} & \omega_{n,2} & \cdots & \omega_{n,n} \end{bmatrix} \tag{2.4}$$

Naj poudarimo, da nad matriko ne veljajo enake operacije kot v linearni algebri. Vrstice predstavljajo zaupanje izbrane entitete do ostalih, stolpci pa predstavljajo zaupanje (t. i. *vektor zaupanja*) celotne skupnosti do izbrane entitete. Nad elementi matrike obstajajo operacije v obliki $\omega^+_{i,k} = op_i(\omega^-_{i,k}, \omega^-_{j,k})$, kjer je op_i prefiksni operator za izračun stopnje zaupanja entitete i in j do entitete k. Oznaka "$-$" označuje vrednost pred operacijo, oznaka "$+$" pa označuje rezultat operacije.

Omenimo tri osnovne prefiksne operatorje: *optimistični* ($\uparrow$), *pesimistični* ($\downarrow$) in *uravnotežen* ($\leftrightarrow$). Rezultati operacij $\omega^+_{i,k}$ so določeni v tabeli (2.1, vir [6]), v splošnem pa veljajo sledeče zakonitosti:

$$\uparrow_i (1, \omega_{j,k}) = 1,\ \downarrow_i (-1, \omega_{j,k}) = -1,\ \leftrightarrow_i (0, \omega_{j,k}) = 0,\ op_i(-, \omega_{j,k}) = - \tag{2.5}$$

Rezultat operacij na vektorjih zaupanja dajejo stopnjo ugleda posamezne entitete in so lahko osnova za odločanje o interakcijah.

Tabela 2.1: Definicijska tabela kvalitativnih operatorjev.

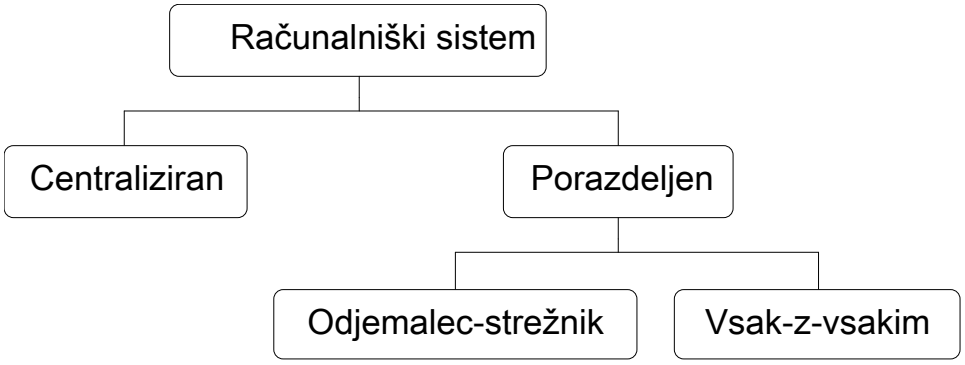
$\omega^-_{i,k}$	$\omega^-_{j,k}$	$\omega^+_{i,k}, \uparrow_i$	$\omega^+_{i,k}, \downarrow_i$	$\omega^+_{i,k}, \leftrightarrow_i$
1	1	1	1	1
1	0	1	0	0
1	-1	1	-1	0
1	-	1	1	1
0	1	1	0	0
0	0	0	0	0
0	-1	0	-1	0
0	-	0	0	0
-1	1	1	-1	0
-1	0	0	-1	0
-1	-1	-1	-1	-1
-1	-	-1	-1	-1
-	*	-	-	-

Poglavje 3

Porazdeljeni sistemi

3.1 Splošno

Računalniške sisteme lahko opredelimo bodisi kot centralizirane (angl. centralised) ali kot porazdeljene (angl. distributed). Porazdeljene sisteme lahko še naprej razdelimo v model odjemalec-strežnik (angl. client-server) ali model vsak-z-vsakim (angl. peeer-to-peer, P2P), slika 3.1. Nadaljnjo delitev modela P2P bomo spoznali v nadaljevanju.



Slika 3.1: Klasifikacija računalniških sistemov.

Spomnimo, da je osnovni namen pričujočega dela predstaviti model, kako več globalno porazdeljenih sistemov za upravljanje ugleda oz. zaupanja učinkovito in transparentno (tj. za končnega uporabnika nevidno) povezati na tak način, da bo podan rezultat stopnje zaupanja odraz ne le lokalnih ocen, temveč tudi ocen, ki jih je sistem pridobil od drugih sistemov. Očitno je, da imamo opravka s porazdeljenim sistemom in da se nam obetata dve

ločeni paradigmi izgradnje takega sistema, model odjemalec-strežnik ter model vsak-z-vsakim. V nadaljevanju sledita opis in primerjava obeh arhitekturnih modelov. Poseben razdelek je namenjen tudi predstavitvi koncepta spletnih storitev.

3.2 Model odjemalec-strežnik

3.2.1 Splošno

Bistvo takega modela je, da obstaja nekdo, ki določeno storitev želi – to je odjemalec (angl. client) – in nekdo drug, ki je zahtevani posel sposoben izvesti ter odjemalcu posredovati odgovor – to je strežnik (angl. server). [8]

Odjemalci in strežniki komunicirajo preko računalniškega omrežja. Slednji so največkrat visoko zmogljivi računalniki, sposobni hitreje opraviti računsko zahtevnejša opravila kot odjemalci, ki del bremena tako prelagajo na strežnike. Odjemalci ne delijo podatkov, procesne moči ali drugih računalniških virov s strežnikom ali drugimi odjemalci, temveč le pošljejo zahtevo strežniku in čakajo na odgovor, kar je, kot bomo videli v nadaljevanju, poglavitna razlika med modelom odjemalec-strežnik ter modelom vsak-z-vsakim.

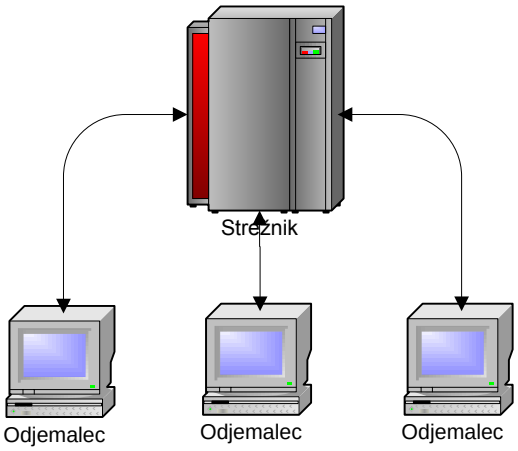
Standardne omrežne funkcije, kot so elektronska pošta, dostop do spletnih strani, dostop do podatkovne baze itd., temeljijo na tej arhitekturi. Npr. spletni brskalnik na uporabnikovem računalniku nastopa v vlogi odjemalca, ki lahko dostopa do informacij na poljubnem spletnem strežniku. Ko želimo preveriti stanje našega bančnega računa s pomočjo aplikacije za spletno bančništvo, naš brskalnik posreduje zahtevo spletnemu strežniku na banki. Ta strežnik – sedaj v vlogi odjemalca za dostop do podatkovne baze – posreduje zahtevo v podatkovno bazo, kjer se prebere stanje na računu. Rezultat se vrne spletnemu strežniku, ta ga posreduje brskalniku in slednji ga na koncu prikaže uporabniku.

V opisanem primeru se model odjemalec-strežnik pojavi dvakrat. Najprej med brskalnikom ter spletnim strežnikom, nato pa še v komunikaciji med spletnim strežnikom in podatkovno bazo.

3.2.2 Prednosti

Prednosti modela odjemalec-strežnik lahko povzamemo v naslednjih točkah:

Ločevanje vlog. Ta arhitektura uvaja vloge in odgovornosti za posamezne komponente porazdeljenega računalniškega sistema. Odjemalci vedo le



Slika 3.2: Arhitekturni model odjemalec-strežnik.

za storitev, ki jo strežnik ponuja in na katerem naslovu je dosegljiva, podrobnosti implementacije pa so odjemalcem skrite, kar pripelje do dodatne prednosti, in sicer lažjega vzdževanja (angl. ease of maintenance).

Lažje vzdrževanje. Mogoče je zamenjati, popraviti, nadgraditi ali celo preseliti strežnik, brez da bi odjemalci za to sploh vedeli.

Lažje zagotavljanje varnosti. Vsi podatki so shranjeni na strežniku, ki ima praviloma boljše varnostne mehanizme kot odjemalci sami. Strežnik lahko tako bolje nadzoruje dostop do podatkov in razporeja zasedenost računalniških virov.

Lažje obvladovanje sprememb podatkov. Ker so podatki centralizirani, je lažje obvladovati njihove spremembe.

3.2.3 Slabosti

Model odjemalec-strežnik prinaša tudi nekatere pomanjkljivosti, in sicer:

Ozko grlo. Ko se število sočasnih dostopov do strežnika poveča, lahko ta kaj kmalu postane preobremenjen. Zahteve odjemalcev se nabirajo v vrsti, medtem ko so ti blokirani[1]. Gre za osnovno pomanjkljivost, ki ta model

[1]Na tem mestu velja omeniti, da poznamo dve paradigmi pri klicu oddaljenih procedur, in sicer sinhrono (blokirajočo) in asinhrono (neblokirajočo). Pri prvi odjemalec stoji, medtem ko strežnik obdeluje zahtevo, pri drugem pa odjemalec nadaljuje z delom in ga strežnik z

tare že od spočetka. Na sliki 3.2 lahko opazimo, da je strežnik v sredini stična točka vseh zahtev in tako predstavlja ozko grlo sistema.

Pomanjkanje robustnosti. Če v tem modelu pride do kritične okvare strežnika, celotno omrežje stoji. Imamo tako imenovano (kritično) točko izpada (angl. single point of failure, SPOF), ki v našem primeru predstavlja izpad strežnika.

Navedene pomanjkljivosti so bile tiste, ki so pripeljale do razvoja arhitekturnega modela vsak-z-vsakim.

3.3 Model vsak-z-vsakim

3.3.1 Splošno

Model vsak-z-vsakim (angl. peer-to-peer, P2P) sestavljajo vrstniki (angl. peers), od katerih vsak prispeva delež svojih kapacitet (procesna moč, podatki, pasovna širina, ...) neposredno ostalim vrstnikom, brez kakršnegakoli vmesnega posrednika. Vrstniki hkrati nastopajo v vlogi porabnikov (odjemalcev) in ponudnikov (strežnikov) storitev, kar je poglavitna razlika med modelom odjemalec-strežnik, kjer odjemalci zgolj porabljajo in strežniki zgolj ponujajo storitve.

Popularnost so temu arhitekturnemu modelu prinesli sistemi za deljenje datotek, kot so Napster[2], KaZaA[3], protokol BitTorrent in drugi.

3.3.2 Klasifikacija modelov

V *"čistih"* omrežjih P2P so vrstniki popolnoma enaki. Vsak združuje vlogi odjemalca in strežnika. V takih omrežjih ni centralne entitete, ki bi nadzorovala omrežje, niti ni centralnega usmrejevalnika. Primera takih omrežij sta Gnutella (različice pred v0.4) ter Freenet[4].

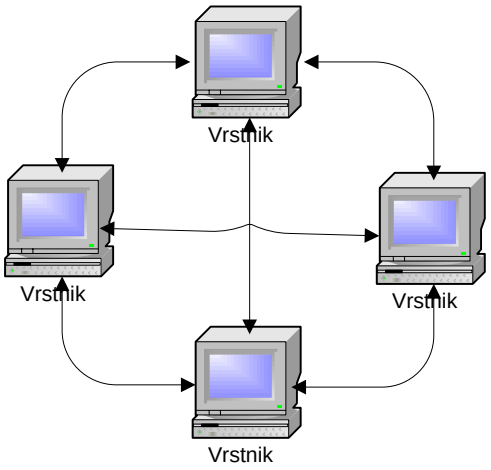
Poleg *čistih* obstaja množica *hibridnih* modelov, ki grupirajo vrstnike v dve skupini; v odjemalce (angl. client node) in nadzornike (angl. overlay node). Vsak vrstnik lahko nastopi v katerikoli vlogi, odvisno od trenutnih potreb omrežja. Diferenciacija na *normalne* in *boljše* vrstnike se izvaja

dodatnim klicem obvesti, ko je njegova zahteva obdelana. Za naš primer asinhroni klic ne pride v upoštev, saj odjemalci ne morejo nadaljevati z delom brez obdelane zahteve.

[2]http://free.napster.com

[3]http://www.kazaa.com

[4]http://freenetproject.org



Slika 3.3: Arhitekturni model vsak-z-vsakim.

zato, da se lahko omrežje učinkovito spopade s problemi skalabilnosti v čistih P2P omrežjih. Primer takega hibridnega omrežja je Gnutella (od različice v0.4 dalje). Posebno podvrsto *hibridnih* modelov predstavljajo omrežja, ki imajo centralni strežnik kot zagonski mehanizem (angl. bootstrap mechanism) na eni in čisto P2P arhitekturo za prenos podatkov na drugi strani. Takim omrežjem pravimo *centralizirana omrežja*, saj ne morejo delovati brez centralnih strežnikov. Primer takega omrežja je omrežje eDonkey (eD2k)[5].

3.3.3 Odkrivanje vrstnikov

Starejša omrežja P2P so podvajala vire skozi vsa vozlišča omrežja. To je omogočalo lokalno iskanje, ampak je hkrati zelo povečalo promet med vozlišči.

Moderna omrežja uporabljajo centralne strežnike in usmerjeno iskanje. Centralni strežniki se tipično uporabljajo za pridobivanje potencialnih soležnikov (Tor), koordinacijo njihovih aktivnosti (folding@home) in iskanje (Napster, eMule). Decentralizirano iskanje se je najprej opravljalo tako, da se je iskalna poizvedba poslala čez vsa vozlišča. Danes se uporabljajo strategije usmerjenega iskanja (angl. directed search strategy) s pomočjo super-vrstnikov (angl. supernode) in porazdeljenih zgoščevalnih funkcij (angl. distributed hash function).

[5]http://www.edonkey.com

3.3.4 Prednosti

Vsebina tega razdelka je na nek način zrcalna vsebini poglavja o slabostih arhitekture odjemalec-strežnik. Arhitektura P2P v svojem bistvu ponuja odgovor največjima očitkoma modela odjemalec strežnik, in sicer:

Odprava ozkega grla. V omrežjih vsak-z-vsakim vsa vozlišča predstavljajo odjemalce in strežnike, zato s številom vrstnikov (tj. porabo) narašča tudi zmogljivost omrežja.

Robustnost sistema. V sistemih P2P – to posebno velja za *čiste* sisteme – lahko vrstniki najdejo informacije in sodelujejo med seboj, ne da bi se morali zanašati na centralno entiteto. Izpad posameznega vrstnika ima zanemarljiv vpliv na delovanje celotnega omrežja. S tem je odpravljena (kritična) točka izpada.

3.3.5 Slabosti

Literatura kot slabost tega arhitekturnega modela navaja predvsem različne možnosti zlorab zoper posameznega uporabnika kot tudi zoper celotno omrežje, ki jih povzroča ne-varno napisana koda. V preteklosti se je to zgodilo omrežju FastTrack, ko so nekatera podjetja uspela vriniti pokvarjene delčke datotek v omrežje na takšen način, da so datoteke, ki so jih uporabniki prenašali preko omrežja, postale neuporabne. Tarča večine takih napadov so bile glasbene datoteke formata mp3. Vendar so sčasoma odjemalci omrežij P2P postali boljši in danes vključujejo cel kup varnostnih mehanizmov za preverjanje in enkripcijo datotek.

Zavoljo doslednosti iz prejšnjega razdelka pa si sedaj poglejmo nekatere pomanjkljivosti modela P2P v primerjavi s prednostnimi modela odjemalec-strežnik:

Združevanje vlog. Za razliko od modela odjemalec-strežnik model P2P zopet združi vlogo odjemalca in strežnika v eni entiteti. To pomeni, da je zgradba vozlišč kompleksnejša, saj vsako vozlišče nastopa v vlogi odjemalca in strežnika. Tako se odpirajo dodatne možnosti napak in zlorab.

Težje vzdrževanje. Ne samo, da imajo vrstniki kompleksnejšo zgradbo, temveč vsaka strukturna sprememba na njih pomeni, da je potrebno spremeniti vsako vozlišče v omrežju.

Težje zagotavljanje varnosti. Podatki v modelu P2P niso shranjeni na centralni lokaciji, zato je zagotavljanje varnosti težje. Vsako vozlišče je zadolženo za zagotavljanje varnosti podatkov, ki jih ima v svoji domeni. Lahko bi rekli, da so podatki v omrežju P2P varni tako, kot je varen najmanj varen vrstnik v omrežju.

Težje obvladovanje sprememb podatkov. Sprememba podatka na enem vozlišču zahteva sinhronizacijo s podatki v ostalih vozliščih, ki vsebujejo kopijo tega podatka. Tako se odpre še velik izziv obvladovanja konsistence porazdeljenih podatkov, ki ga v modelu odjemalec-strežnik nismo imeli.

3.4 Spletne storitve

3.4.1 Definicija spletnih storitev

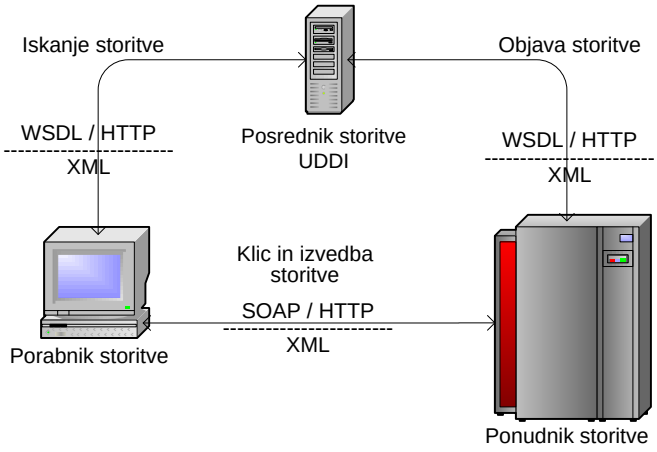
World Wide Web Consortium (W3C)[6] opredeljuje spletne storitve kot "programsko opremo za podporo interoperabilni komunikaciji med računalniki." Osnovne tri komponente spletnih storitev so sporočila SOAP, opisnik WSDL in imenik UDDI.

Sporočila SOAP (Simple Object Access Protocol) so sporočila zapisana v formatu eXtensible Markup Language (XML)[7], ki si jih računalniški sistemi pri uporabi spletnih storitev izmenjujejo. Najpogosteje uporabljeni komunikacijski protokol je Hypertext Transfer Protocol (HTTP). Opisnik spletne storitve je podan v formatu Web Service Description Language (WSDL), ki je prav tako realiziran v jeziku XML. Opisnik vsebuje informacije o sami storitvi (angl. service), pristopni točki storitve (angl. service endpoint), vezavi storitve (angl. binding), vmesniku (angl. interface) in operacijah (angl. operation). Odkrivanje spletnih storitev je realizirano s pomočjo imenika UDDI. V njem lahko ponudniki spletnih storitev storitve objavijo, iskalci pa poiščejo.

Spletne storitve so realizirane preko programskega vmesnika (angl. application programming interface, API), do katerega lahko dostopamo preko omrežja kot je internet. Drugi pristopi, ki ponujano podobne funkcije kot spletne storitve, so Common Object Request Broker Architecture (CORBA)[8], Microsoftov Distributed Component Object Model (DCOM)[9] in Sunov Remote

[6] http://www.w3.org
[7] Označevalni jezik, ki je podoben jeziku HTML.
[8] http://www.omg.org/gettingstarted/corbafaq.htm
[9] http://msdn.microsoft.com/sl-si/library/cc201989.aspx

Method Invocation (RMI)[10]. Tipično pa z izrazom spletne storitve zajamemo odjemalce in strežnike, ki komunicirajo preko protokola HTTP, uporabljajo pa standarde SOAP, UDDI in WSDL.



Slika 3.4: Arhitektura spletnih storitev.

3.4.2 Odkrivanje spletnih storitev

Izvedeli smo, kako spletne storitve delujejo, sedaj pa si poglejmo še, kako iskalci spletnih storitev te storitve najdejo.

Da bi lahko porabniki storitev (angl. consumers, tudi service requesters) storitve našli, jih ponudniki (angl. service providers) opremijo z opisom, dokumentom WSDL. Ta opis lahko nato objavijo v imeniku UDDI[11], ki tako nastopa v vlogi posrednika storitve (angl. service broker).

Razmerje med porabnikom, ponudnikom ter posrednikom storitve podaja slika 3.4. Porabnik s pomočjo imenika UDDI poišče želeno spletno storitev oz. pridobi njen opisnik. Nadaljnja komunikacija se odvija le še med porabnikom ter ponudnikom storitve.

Omenili smo že, da obstaja precej alternativ spletnim storitvam, vendar ježiček na tehtnici naše izbire predstavlja odlična podpora interoperabilnosti in to na nivoju poslovnih rešitev. Z izmenjavo sporočil v jeziku XML je odstranjena odvisnost od posamezne implementacijske tehnologije. Različni sistemi za upravljanje zaupanja oz. ugleda so lahko realizirani v poljubnih

[10] http://java.sun.com/j2se/1.5.0/docs/guide/rmi

[11] Universal Description Discovery and Integration je platformno neodvisen in na jeziku XML osnovan register poslovnih subjektov.

programskih jezikih, njihova medsebojna komunikacija pa bo tako s pomočjo spletnih storitev vedno potekala v jeziku XML.

3.4.3 Primer sporočila SOAP

Za bolj nazorno predstavo si oglejmo še, kako v praksi izgleda primer klica spletne storitve in odgovora, ki ga spletna storitev vrne.

Denimo, da smo soočeni s problemom izbire najugodnejšega kredita. Za lažje odločanje si bomo pomagali z izračunom na oddaljenem računalniku. Ta izračun bomo opravili s pomočjo spletne storitve. Njej na vhodu podamo znesek kredita, želeno število obrokov, ter obrestno mero. Spletna storitev pa nam kot odgovor posreduje sporočilo z višino mesečnega obroka.

XML 3.1: Klic spletne storitve.

```
<SOAP-ENV:Envelope
SOAP-ENV:encodingStyle=
   'http://schemas.xmlsoap.org/soap/encoding/'
xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>
 <SOAP-ENV:Body>
  <calculate>
   <principal xsi:type='xsd:double'>10000.0</principal>
   <months xsi:type='xsd:int'>12</months>
   <rate xsi:type='xsd:float'>0.08</rate>
  </calculate>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

XML 3.2: Odgovor spletne storitve

```
<SOAP-ENV:Envelope
SOAP-ENV:encodingStyle=
   'http://schemas.xmlsoap.org/soap/encoding/'
xmlns:SOAP-ENV='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>
 <SOAP-ENV:Body>
  <calculateResponse>
   <calculateResult xsi:type='xsd:double'>
    1326.95
```

```
    </calculateResult>
   </calculateResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.5 Zaključek

V tem poglavju smo si ogledali temeljni paradigmi za povezovanje porazdeljenih sistemov, kot je model odjemalec-strežnik in model P2P. Prav tako smo spoznali pojem spletnih storitev in utemeljili njihovo uporabo v naši problemski domeni. Pridobljeno znanje nam bo koristilo v naslednjem poglavju, v katerem bomo skušali zgraditi model povezovanja različnih centraliziranih sistemov za upravljanje zaupanja in ugleda.

Poglavje 4

Integracija globalno porazdeljenih sistemov

4.1 Uvod

V prejšnjem poglavju smo ugotovili, da obstajata dva temeljna pristopa pri izdelavi porazdeljenega računalniškega sistema, in sicer izvedenka iz modela odjemalec-strežnik ter izvedenka iz modela P2P. V tem poglavju bomo analizirali, kakšen pristop bi bil bolj primeren za povezovanje različnih centraliziranih[1] sistemov za upravljanje zaupanja, ki so globalno porazdeljeni, kar pomeni, da se nahajajo na različnih prostorskih lokacijah.

V okviru te naloge je bila izdelana tudi aplikacija *Distributed trustGuard (krajše DTG)*, ki ni toliko sistem za upravljanje zaupanja ali ugleda, kot porazdeljen register podanih ocen o entitetah. Podatki, ki jih hrani DTG, so lahko vhodni podatki za druge aplikacije, ki iz podanih ocen izračunajo stopnjo zaupanja za posamezno entiteto. DTG se lahko povezuje z različnimi sistemi za upravljanje zaupanja. V sklopu tega poglavja je pri vsakem obravnavanem vprašanju tudi opisan način razrešitve v aplikaciji *Distributed trustGuard*.

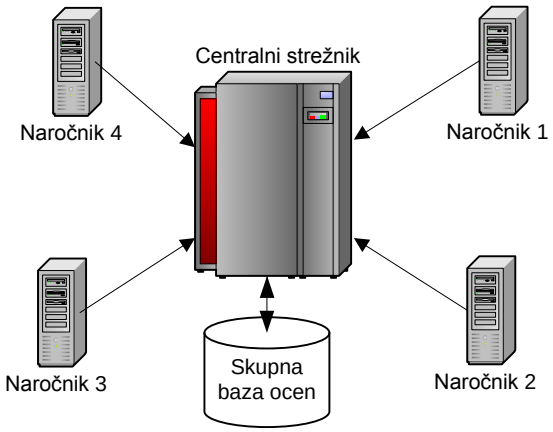
[1]Spomnimo, da smo v razdelku 2.6 spoznali dva pristopa izgradnje sistema za upravljanje zaupanja, in sicer centraliziranega ter porazdeljenega. Gradnja in delovanje slednjih sta izredno kompleksni, zato je ta model zaenkrat zgolj v akademski domeni, v praksi (eBay, Amazon) pa se uporabljajo centralizirani sistemi.

4.2 Centraliziran sistem

4.2.1 Opis

Centraliziran sistem bomo imenovali izpeljanko iz modela odjemalec-strežnik. Predpostavlja obstoj centralne lokacije, osrednjega strežnika, na katerega so priključeni vsi ostali sistemi za upravljanje zaupanja. Ta centralni strežnik je odgovoren za shranjevanje ocen iz vseh sodelujočih sistemov. Sistemom, ki so z njim povezani – imenujmo jih *naročniki* – omogoča dve osnovni storitvi, in sicer:

- poizvedbo po ocenah ali ugledu posamezne entitete ter
- shranjevanje (vnos) ocen entitet.



Slika 4.1: Arhitektura centraliziranega sistema.

Ko želi naročnik izračunati stopnjo ugleda določene entitete, pošlje poizvedbo na centralni strežnik. Slednji nato vrne vse ocene o tej entiteti ali že izračunan rezultat. In podobno, ko v sistemu nekega naročnika entiteta poda oceno o drugi entiteti, se ocena ne shrani lokalno v sistem naročnika, temveč na centralni strežnik in postane nemudoma dostopna vsem ostalim naročnikom.

Vidimo, da ta sistem prenese velik del bremena, ki ga sicer opravlja sistem za upravljanje zaupanja, na centralni strežnik. Lahko bi celo zapisali, da naročniki upravljanje zaupanja *oddajo v zunanje izvajanje* (angl. outsourcing) centralnemu strežniku. Nepovezani so morali sami v celoti skrbeti za upravljanje zaupanja, sedaj, povezani, pa postanejo le vmesnik za dostop do

centralnega strežnika. To je še posebej hvaležna lastnost, saj je sistem za upravljanje zaupanja navadno podporna komponenta drugih sistemov, npr. sistema za elektronsko trgovanje in licitiranje, in si želimo, da glavnemu sistemu odvzame čim manj računalniških virov. V prejšnjih poglavjih omenjena eBay ter Amazon tako ne bi več shranjevala ocen lokalno, temveč bi jih pošiljala na centralni strežnik, ki bi nato postregel z odgovorom vsakemu naročniku. Tako izračunana stopnja zaupanja neke entitete ne bi bila zgolj odraz ocen posameznega naročnika (npr. le sistema Amazon), temveč tudi ostalih sistemov kot so eBay, slovenska Bolha[2] in ostalih naročnikov, ki bi bili povezani s centralnim strežnikom.

4.2.2 Prednosti

Centraliziran sistem ima mnoge prednosti. Večino jih lahko izpeljemo iz prednosti modela odjemalec-strežnik (ločene vloge, lažje vzdrževanje, varnost, lažje obvladovanje sprememb podatkov), posebej pa velja izpostaviti naslednje:

(a) **Enostavnost.** Očitno je, da pade celotno breme hranjenja podatkov (lahko tudi računanja stopenj ugleda in zaupanja) ter skrb za izdelavo varnostnih kopij na ramena centralnega strežnika. Naročnikom je potrebno realizirati le zanesljiv in učinkovit komunikacijski kanal, preko katerega komunicirajo s strežnikom. Taka rešitev se zdi še posebej privlačna za sisteme, ki nimajo realiziranega sistema za upravljanje zaupanja, si pa ga želijo.

(b) **Konsistentni izračuni.** Zagotovljen je konsistenten izračun stopnje zaupanja za posamezno entiteto na kateremkoli naročniku, saj so vhodni podatki za izračun na vseh naročnikih enaki. Za primer poglejmo sistem na sliki 4.1. V njem bo vsak naročnik izračunal enako stopnjo zaupanja v določeno entiteto, saj bo uporabljal iste podatke kot ostali trije. To je pomembna razlika med centraliziranim modelom in modelom P2P.

4.2.3 Omejitve

Ker pa ima vsaka palica dva konca, tudi ta model pestijo mnoge slabosti oz. omejitve. Nekatere so povsem tehnične, druge bolj poslovne narave:

(a) **Ozko grlo.** Če si zamislimo centralni sistem, na katerega se povezuje večje število sistemov, je očitno, da potrebujemo sistem ogromnih kapacitet.

[2] http://www.bolha.com

Prostorske kapacitete centralnega strežnika bi morale biti praktično neomejene, prav tako tudi pasovna širina med naročniki in sistemom. Medtem ko se zdi prvi pomislek ob današnji tehnologiji podatkovne shrambe še rešljiv, vzbuja drugi večje dvome. Delno rešitev bi lahko poiskali v lokalnem predpomnilniku vsakega naročnika. Vendar bi tako, ob uporabi zastarelih ocen, izgubili konsistenten izračun ter povečali kompleksnost posameznega odjemalca, ki predstavljata glavni prednosti centraliziranega sistema.

(b) **Manjša robustnost.** V primeru izpada centralnega strežnika, ostanejo vsi naročniki brez podatkov in sistema. Zato velja poudariti, da mora vsak naročnik posedovati zadostno stopnjo zaupanja v centralni strežnik, da mu prepusti podatke v shrambo. Če pogledamo še z nasprotnega zornega kota; podatki na centralnem strežniku so zelo dragoceni in potrebna je dodatna previdnost ter skrb pri ravnanju z njimi, saj brez tega naročniki ne bodo pripravljeni zaupati svojih ocen.

(c) **Nepripravljenost deljenja ocen.** Če je imel naročnik prej pomisleke glede varnosti in zanesljivosti centralnega strežnika, jih ima v tem primeru iz povsem ekonomskih razlogov. Omenimo, da trenutno prevladuje mnenje, da je lastnik sistema, v katerem ocena nastane, lastnik lete in kot tak jo ima vso pravico ne razkriti oz. obdržati zase. To potrjujejo tudi zgledi iz prakse, saj lahko v literaturi [9] zasledimo, kako je eBay protestiral, ko je Amazon ob splavitvi sistema za obvladovanje zaupanja omogočil uporabnikom, da uvozijo svoj ugled iz sistema eBay. Glavni očitek lastnikov eBaya je bil prav ta, da so ocene njihova last in ne last uporabnikov. Zato predpostavimo, da z uporabo centralnega strežnika vsak naročnik soglaša z deljenjem lastnih ocen z vsemi ostalimi naročniki. Seveda si lahko zamislimo tako realiziran centralni strežnik, v katerem je mogoč dodaten nadzor nad tem, kateri naročniki lahko pridobijo posamezne ocene, vendar je z vidika celotnega sistema najboljše, da vsi naročniki delijo ocene z vsemi ostalimi. Le v tem primeru lahko zagotovimo najbolj točen in konsistenten izračun stopenj zaupanja. V primeru selektivnega razkrivanja ocen se lahko upravičeno vprašamo, koliko je tako početje smotrno z vidika sistema kot celote, saj posamezni naročniki zasegajo strežnikove kapacitete, k natančnosti in konsistentnim rezultatom pa ne prispevajo. Iz napisanega lahko sklepamo, da je potrebno pri dodajanju novega naročnika centralnemu strežniku pridobiti soglasje vseh obstoječih naročnikov, kar utegne postati kamen spotike in zaviralni moment pri širitvi takega sistema.

4.2.4 Povzetek prednosti in slabosti

Opažamo, da ima centraliziran sistem – svoji preprostosti navkljub – precej omejitev. Te niso le v domeni tehnologije, temveč tudi poslovne politike lastnikov obstoječih sistemov za upravljanje zaupanja. Netehnične omejitve utegnejo po našem mnenju celo bolj vplivati na izbiro modela povezovanja sistemov. V naslednjem razdelku si bomo ogledali, kako bi se z obravnavanimi problemi spopadel model vsak-z-vsakim.

Bralcu, ki je v tem poglavju neuspešno iskal "razrešitve vprašanj v aplikaciji Distributed trustGuard" naj povemo, da aplikacija implementira model P2P, zato bo večkrat omenjena v nadaljevanju.

4.3 Sistem vsak-z-vsakim

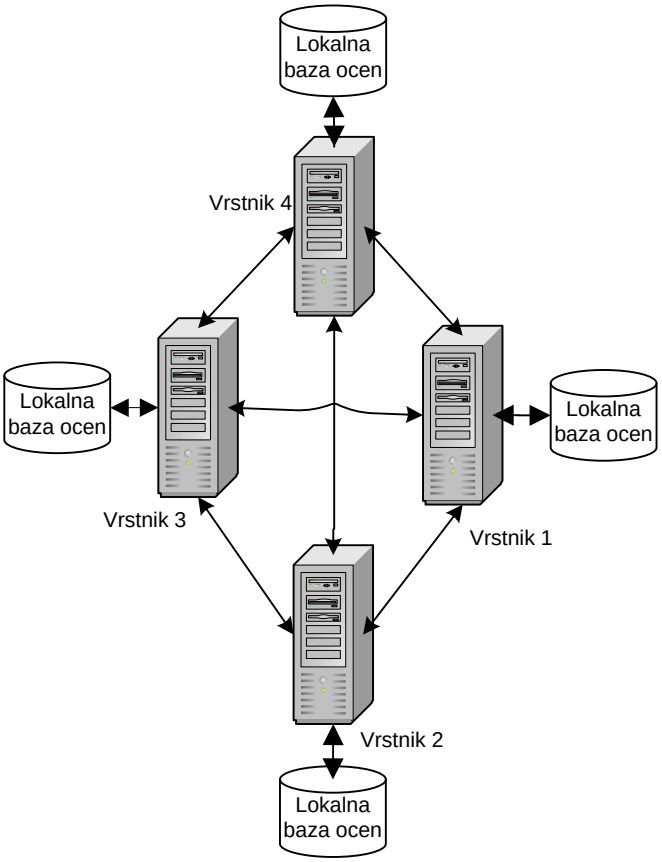
4.3.1 Opis

Sistem vsak-z-vsakim bomo imenovali arhitekturo vezave sistemov za upravljanje zaupanja, ki je izpeljana iz modela P2P. Sistem vsak-z-vsakim za delovanje ne potrebuje centralnega strežnika. Namesto tega se zanaša za učinkovito sodelovanje vrstnikov. Vsak sistem lokalno shranjuje ocene, ki jih podajo entitete, in jih nato deli z ostalimi vrstniki v mreži P2P. Arhitektura modela P2P je prikazana na sliki 4.2.

Vidimo, da je ta sistem v primerjavi s centraliziranim zelo drugačen. Če smo pri slednjem govorili o *zunanjem izvajanju* upravljanja zaupanja, pri tem tega ne moremo. Vsak vrstnik v mreži P2P je v celoti zadolžen za upravljanje zaupanja v svojem sistemu. Sam lokalno shranjuje ocene in izračunava stopnje zaupanja, ostalim vrstnikom pa preko programskega vmesnika (spletnih storitev) nudi dostop do lastne (lokalne) baze ocen. Odločitev, s katerimi vrstniki se bo povezoval, je v celoti prepuščena posameznemu vrstniku in ni odvisna od soglasja ostalih vrstnikov, kot je to potrebno pri centraliziranemu sistemu.

4.3.2 Odkrivanje

V vsakem modelu P2P je potrebno imeti mehanizem, ki določa, kako vozlišča izvejo za lokacijo drugih vozlišč. V razdelku 3.3.2 smo zapisali, da obstajajo *centralizirana omrežja*, ki potrebujejo centralne strežnike kot zagonske mehanizme. Ker sistemi za upravljanje zaupanja komunicirajo s pomočjo spletnih storitev, njihovo odkrivanje pa je standardizirano z imeniki



Slika 4.2: Arhitektura sistema vsak-z-vsakim.

UDDI, predlagamo, da sistemi objavijo svoje storitve v imeniku UDDI. Sistem, ki objavlja storitev, mora poleg ostalih podatkov, ki jih imenik UDDI zahteva, nujno navesti še v kakšen kontekst spadajo njegove ocene; npr. spletna prodaja in nakup, pisanje člankov, deljenje nasvetov itd.

Na podlagi objavljenih podatkov v registru UDDI se lahko ostali vrstniki odločijo, s katerimi sistemi se bodo povezali oz. odkrijejo njihovo lokacijo. V priloženi implementaciji je imenik UDDI realiziran v Javanskem okolju s pomočjo odprtokodne aplikacije Apache jUDDI[3].

[3]http://ws.apache.org/juddi

4.3.3 Omejitve

Začnimo tokrat z omejitvami takega sistema. Za lažji začetek lahko izhajamo iz slabosti arhitekturnega modela vsak-z-vsakim:

(a) **Kompleksnejša zgradba.** Pri centraliziranem modelu smo zapisali, da naloge sistema za upravljanje zaupanja padejo na centralni strežnik, njegovi naročniki pa zgolj realizirajo komunikacijski kanal. Pri tem modelu pa mora vsak vrstnik realizirati lasten sistem za upravljanje zaupanja, prav tako mora imeti metodo, ki pove, kako od drugih sistemov pridobljene ocene vključiti v izračun stopenj zaupanja. Hkrati mora ponuditi vmesnik preko katerega partnerji dostopajo do njegovih ocen. Za sisteme, ki še nimajo realiziranega sistema za upravljanje zaupanja, predstavlja ta sistem večji zalogaj kot centraliziran.

(b) **Večje varnostno tveganje.** V sistemu centralnega strežnika so vsi podatki shranjeni na eni lokaciji. Imamo le eno točko dostopa in če jo zadostno zavarujemo, je sistem relativno varen. V sistemu P2P pa je potrebno zagotoviti varnost pri vsakem vrstniku v mreži. V kolikor je eden kompromitiran, lahko nepooblaščeni subjekti dostopajo do podatkov vseh vrstnikov, ki so z njim povezani.

(c) **Nekonsistentni izračuni.** V centraliziranemu sistemu je – ob pogoju, da vsi delijo ocene z vsemi – izračun stopnje zaupanja za določeno entiteto vedno enak. V sistemu P2P pa tega v splošnem ne moremo trditi. Konsistenten izračun lahko pričakujemo le v primeru, da imamo množico sistemov, ki so med seboj popolnoma povezani in hkrati noben od njih ni povezan kakšnim drugim sistemom. Tako je tudi v primeru na sliki 4.2.

4.3.4 Prednosti

V kolikor smo pripravljeni sprejeti zgoraj naštete omejitve sistema P2P, nam ta ponudi tudi mnoge prednosti:

(a) **Odprava ozkih grl.** Z odpravo centralnega strežnika se znebimo osrednje točke in tako tudi ozkega grla. Vsa komunikacija sedaj poteka le med tistimi vrstniki, ki želijo biti povezani in ne med vsemi.

(b) **Večja robustnost.** Izpad posameznega sistema vpliva le na tiste vrstnike, ki so (bili) z njim povezani.

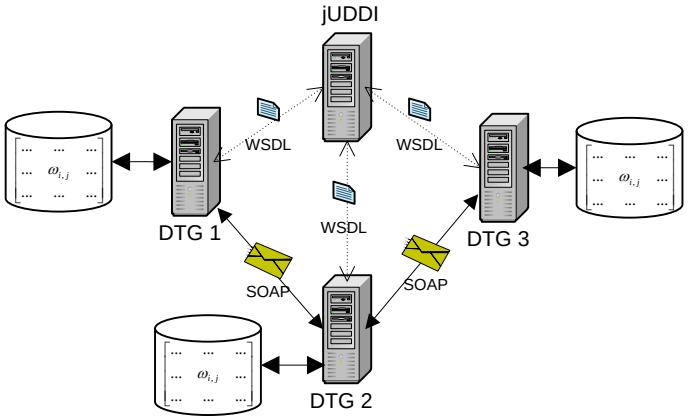
(c) **Večja svoboda sodelovanja.** Centraliziran model predpostavlja, da vsi delijo ocene z vsemi. Izrazili smo pomisleke glede takega modela, tako tehnične kot poslovne. Slednji so lahko po našem mnenju zaviralni element, saj se tako "prisilno" sodelovanje v praksi le redko uresniči. Model P2P omogoča sistemom, da se povezujejo le s tistimi, s katerimi si želijo. Ko se želi nek vrstnik povezati z novim vrstnikom, potrebuje le njegovo soglasje in ne več soglasja vseh ostalih vrstnikov.

4.3.5 Povzetek prednosti in slabosti

Na prvi pogled se zdi, da je cena tega modela previsoka – izguba konsistence izračunov stopenj zaupanja, kompleksnejša zgradba sistemov in dodatna varnostna tveganja so vsekakor neugodni očitki. Vendar model vsak-z-vsakim ponuja alternativo "prisilnemu" sodelovanju, ki ga implicira centraliziran model. Lastniki posameznega sistema se lahko odločijo, s katerimi ostalimi sistemi se bodo povezali ne da bi vplivali na ostale vrstnike v mreži. Ta svoboda je po mnenju avtorja nujen pogoj, če želimo, da tak model ugleda luč v praksi.

4.3.6 Izbrana rešitev za Distributed trustGuard

Distributed trustGuard implementira model vsak-z-vsakim, torej so vsi vrstniki povezani v mrežo P2P. Aplikacija uporablja kvalitativno algebro za izračun stopenj zaupanja. Ocene so shranjene v matriki zaupanja M.



Slika 4.3: Primer povezave sistemov *Distributed trustGuard*.

Na sliki 4.3 so prikazani trije vrstniki DTG, ki so povezani v mrežo P2P. DTG 2 je povezan z DTG 1 in 3, DTG 1 in 3 pa le z DTG 2. Vsi se "zavedajo obstoja" imenika UDDI, iz katerega so pridobili informacijo o lokaciji ostalih vrstnikov. Iz slike lahko sklepamo, da bo lahko najbolj natančne izračune stopnje zaupanja entitet dosegel DTG 2, saj bo poleg lastnih ocen uporabil še ocene iz sistema DTG 1 in DTG 3. Sistemi DTG komunicirajo z izmenjavo sporočil SOAP. Format, v katerem se ocene izmenjujejo, je XML in je podrobno opisan v dodatku A, kot tudi celotna aplikacija *Distributed trustGuard*.

4.4 Predpomnenje

Pri povezovanju računalniških sistemov imamo vedno opravka z zakasnitvami. Te zakasnitve lahko sistem pripeljejo do roba uporabnosti ali celo preko. Zamislimo si spletno trgovino, pri kateri moramo na vsaki strani čakati po nekaj sekund, da se nam izpiše stopnja ugleda prodajalcev artiklov. Ob današnjem hitrem tempu življenja in pojavu vsesplošnega pomanjkanja potrpežljivosti lahko hitro zaključimo, da takšno spletišče ne bi imelo veliko uporabnikov.

Predpomnenje (angl. caching)[4] je koncept, ki takšne zakasnitve skrajša. Gre za tehniko, pri kateri podatke, ki jih pogosto potrebujemo, shranimo na hitro-dostopno lokacijo, kajti dostop do izvorne lokacije traja predolgo. Predpomnenje zato vsebuje element predvidevanja, ki pa je lahko napačno. Motivacija za uporabo predpomnilnika v našem primeru izhaja iz ugotovitve, da bi nenehno izvajanje poizvedb na sistemih, s katerimi smo povezani, preveč upočasnilo delovanje obeh sistemov ter preobremenilo komunikacijski kanal.

Pri predpomnenju imamo v našem primeru opravka z dvema nasprotujočima silama, in sicer z obremenitvijo sistema pri osveževanju predpomnilnika na eni in možnostjo napačnega izračuna stopnje zaupanja na drugi strani. Na tem mestu povejmo, da je možnost napačnega izračuna odvisna od občutljivosti postopka za izračun. Denimo, da iz množice 100 podanih ocen izračunana stopnja zaupanja neke entitete ostane nespremenjena še za naslednjih 10 oddanih ocen. V takem primeru odsotnost najnovejših 10 ocen ne vpliva na točnost izračuna stopnje zaupanja.[5]

[4]Cahe – varno mesto za skrivanje ali shrambo stvari (Webster's New World Dictionary of the American Language, Third College Edition, 1988).

[5]Povejmo, da sta številki 100 in 10 navedeni zgolj kot primera. Kdaj lahko nek postopek izračuna stopnje ugleda označimo kot občutljiv oz. neobčutljiv, je odvisno od okoliščin, v katerih se sistem nahaja.

V praksi se je potrebno odločiti za konkreten pristop. Izbira je odvisna od kapacitet, ki jih imamo na voljo. Če imamo nasičen komunikacijski kanal, doživljamo konkretne upočasnitve pri sinhronizaciji predpomnilnika in uporabljamo postopek za izračun, ki je neobčutljiv, potem izberemo daljšo periodo osveževanja predpomnilnika. Primer lahko analogno obrnemo in izberemo krajšo periodo osveževanja, če imamo občutljiv postopek izračuna, neizkoriščen komunikacijski kanal ter sistem, ki ga sinhronizacija občutno ne upočasnjuje. V realnosti izbira ne bo tako enostavna, zato bo potrebno poiskati pravo mero s pomočjo izračunov, simulacij in testiranj.

Implementiran model vsebuje predpomnilnik z ocenami iz ostalih sistemov in se osvežuje na določeno časovno periodo, ki jo je moč spremeniti tekom izvajanja. Če se med periodo osveževanja ocena na oddaljenem strežniku spremeni, postane le-ta v predpomnilniku netočna. Netočnost ocene je odvisna od občutljivosti postopka za izračun stopnje zaupanja. Distributed trustGuard trenutno še ne implementira postopkov za izračun stopenj ugleda, temveč le hrani ocene.

Namen tega podpoglavja ni podati dokončne rešitve dileme predpomnenja, ampak le opozoriti na dvome, ki se pri njegovi uporabi pojavijo. Konfiguracija sistema je odvisna od od zgoraj opisanih dejavnikov. Najti njihovo idealno kombinacijo pa je vse prej kot enostavna naloga in bo predmet nadaljnjih raziskav.

4.5 Identifikacija entitet

4.5.1 Enolični identifikator

Do sedaj smo povsod predpostavljali, da obstaja nekaj, po čemer lahko entitete enoznačno identificiramo tudi potem, ko združimo imenske prostore različnih sistemov, nikjer pa nismo navedli, kaj to je. Potrebujemo torej identifikator, ki je enoličen, kar pomeni, da lahko ta identifikator uporablja le ena entiteta. Kot odgovor na dlani se nam, sicer z določenimi omejitvami, ponuja elektronski naslov (angl. e-mail). Ima ga praktično vsak, ki uporablja internet, in je enoličen – nek naslov ima lahko le en posameznik[6].

[6]Omenimo, da je mogoče, da si več ljudi deli isti elektronski naslov, vendar so vsi o tem seznanjeni. Tu se domneva, da ko nekdo registrira naslov, postane njegov lastnik in nobena druga oseba ne more tega naslova pridobiti brez vedenja lastnika.

4.5.2 Varnost e-mail naslovov

Izmenjava ocene pomeni, da si izmenjamo podatek, ki ga lahko predstavimo kot trojico (i, p, o), kjer i predstavlja e-mail naslov izvorne entitete (ocenjevalca), p e-mail naslov ponorne (ocenjene) entitete ter o oceno iz domene vrednosti ocen. Če entiteta z naslovom *ana@email.com* oceni entiteto z naslovom *beno@email.com* z oceno *zaupanja vreden (1)* dobimo zapis *(ana@email.com, beno@email.com, 1)*.

V luči boja proti neželeni elektronski pošti (angl. spam) se lahko upravičeno vprašamo, ali je dobro, da si sistemi izmenjujejo elektronske naslove oz. kako verjetno je, da pride do nedovoljenega razkrivanja elektronskih naslovov. Očitno imamo opravka še z enim varnostnim vprašanjem. Ob zagotovitvi trenutnih standardnih varnostnih ukrepov (kriptiran komunikacijski kanal, dobro načrtovani in zgrajeni sistemi) ostanejo edini šibki člen upravljavci sistema, te pa k pravilnemu vedenju zavezujejo veljavni predpisi. Potrebno se je zavedati, da izmenjava ocen poteka na strojnem nivoju in da je sistem tako zasnovan, da z izjemo upravljavcev, ostali uporabniki nimajo dostopa do elektronskih naslovov.

Težava drugačne vrste nastopi takrat, ko nek uporabnik ne dovoli, da sistem, v katerega je prijavljen, komurkoli razkrije njegov elektronski naslov. Zakon o varstvu osebnih podatkov (ZVOP-1, [11]) v 6. členu pravi, da je osebni podatek "katerikoli podatek, ki se nanaša na posameznika, neglede na obliko, v kateri je izražen". Iz definicije izhaja, da je tudi elektronski naslov osebni podatek in zato je vsak uporabnik upravičen do takšne zahteve. Rešitev iz zagate bi lahko našli v tem, da bi uporabnike "prisilili", da sprejmejo pogoje uporabe, v katere bi zapisali, da soglašajo z razkrivanjem e-mail naslovov sistemom, s katerimi se obravnavani sistem povezuje. Kljub temu, da bi omenjen način za večino uporabnikov deloval, saj večina nad branjem pogojev uporabe ni posebej navdušena, verjamemo, da imamo dovolj inženirskega znanja, da poiščemo elegantnejšo pot iz zagate.

Možna rešitev izhaja iz uporabe enosmernih zgoščevalnih funkcij (angl. cryptographic hash function). Enosmerne zgoščevalne funkcije so kriptografske preslikave, ki preslikujejo poljubno velike vhodne nize podatkov v nize točno določene dolžine ti. izvlečke (angl. hash value). Pri tem morajo veljati naslednje lastnosti [3]:

(1) Učinkovit izračun (angl effective computation); za poljuben vhod x je izračun izvlečka $H(x)$ enostaven in hiter.

(2) Primarna rezistenca (angl. first preimage resistance, non-invertability); iz

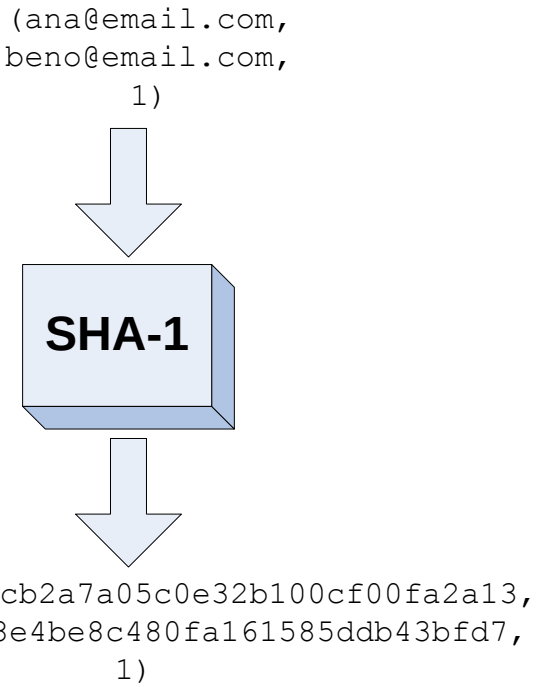
danega izvlečka $H(x)$ je računsko zahteven problem (angl. computationally infeasible) najti vhod x.

(3) Sekundarna rezistenca (angl. second preimage resistance); za dani x_1 je računsko zahteven problem najti tak x_2, $x_1 \neq x_2$, da velja $H(x_1) = H(x_2)$.

(4) Kolizijska rezistenca (angl. collision resistance); za poljuben x_1 je računsko zahteven problem najti tak x_2, da velja $H(x_1) = H(x_2)$.

Med pomembnejšimi predstavniki enosmernih zgoščevalnih funkcij najdemo družino funkcij SHA, RIPEMD ter MD, med katerimi je najbolj znana MD5.

Sedaj kot identifikator ne uporabimo elektronskega naslova, temveč njegov izvleček. Za izračun uporabimo SHA-1. Naša trojica *(ana@email.com, beno@email.com, 1)* tako postane *(3cde5e6c7a58acb2a7a05c0e32b100cf00fa2a13, 4cd53fc7d7d0f8e4be8c480fa161585ddb43bfd7, 1)*. Grafično je omenjen postopek prikazan na sliki 4.4.



Slika 4.4: Uporaba zgoščevalne funkcije SHA-1.

Prejemnik lahko ugotovi, komu pripada podana ocena le v primeru, da pozna izvorna naslova (ana@email.com, beno@email.com) in iz njiju

izračuna izvleček. Do izmenjave dejanskih naslovov ne pride, kar pomeni, da spoštujemo zahteve uporabnikov ter hkrati obdržimo vse lastnosti enoličnega identifikatorja.

4.6 Izmenjava ocen

4.6.1 Problem

Zamislimo si sistema za upravljanje zaupanja A in B. Sistema sta povezana in pripravljena izmenjevati ocene, ki jih entitete na vsakem izmed njiju podajajo druga o drugi. Pojavi se vprašanje, katere ocene si naj izmenjata.

Denimo, da je v sistemu A množica entitet M, v sistemu B pa množica entitet N. Naj bo $P = M \cap N$. P tako predstavlja množico entitet, ki se nahajajo v obeh sistemih. Entitete v množici P so torej tiste, katerih ocene bodo predmet izmenjave. Če je presek P prazen, izmenjevanje ocen ni smiselno. V razdelku 4.5.2 smo zapisali, da je ocena predstavljena kot trojica (i, p, o). Glede na to, ali je v preseku P le ponorna (ocenjena) entiteta p ali tudi izvorna (ocenjevalec) i, ločimo dve vrsti izmenjave: širšo in ožjo.

Da bo razlaga jasnejša, si v pomoč pripravimo zgled in konkretizirajmo sistema A in B. Denimo, da imamo v registru entitet sistema A štiri entitete[7]: α, β, γ in δ. Register sistema B ima tudi le štiri entitete: β, γ, δ in ϵ. V preseku registrov entitet P se tako nahajajo β, γ, δ. Formalno lahko zapišemo $A = \{\alpha, \beta, \gamma, \delta\}$, $B = \{\beta, \gamma, \delta, \epsilon\}$ in $P = \{\beta, \gamma, \delta\}$. Uporabimo kvalitativno algebro in definirajmo matriki M_A in M_B, ki pripadata sistemoma A in B.

$$M_A = \begin{array}{c} \\ \alpha \\ \beta \\ \gamma \\ \delta \end{array} \begin{array}{c} \begin{array}{cccc} \alpha & \beta & \gamma & \delta \end{array} \\ \begin{bmatrix} - & -1 & 0 & -1 \\ 1 & - & 0 & -1 \\ 0 & 0 & - & -1 \\ 1 & -1 & 0 & - \end{bmatrix} \end{array} \qquad M_B = \begin{array}{c} \\ \beta \\ \gamma \\ \delta \\ \epsilon \end{array} \begin{array}{c} \begin{array}{cccc} \beta & \gamma & \delta & \epsilon \end{array} \\ \begin{bmatrix} - & 1 & -1 & 1 \\ 0 & - & 1 & -1 \\ 0 & 1 & - & -1 \\ 0 & 1 & 0 & - \end{bmatrix} \end{array}$$

4.6.2 Širša izmenjava ocen

Širša izmenjava ocen pomeni, da sistem A od sistema B pridobi tiste ocene, ki so jih v sistemu B pridobile entitete iz preseka P. Torej vse tiste ocene, ki so jih v sistemu B pridobile entitete, ki "domujejo" v obeh sistemih. Domicil izvornih

[7]Entitete bi morali označevati z elektronskim naslovom, a tukaj zavoljo jedrnatega zapisa raje uporabimo črke grške abecede.

entitet pri tej vrsti izmenjave ni pomemben, kar pomeni, da izmenjamo tudi ocene, ki so jih podale entitete, ki jih v sistemu A ni. Formalno lahko zapišemo, da se izmenjajo vse ocene (i, p, o) za katere velja $p \in P$. V našem primeru bi sistem A poslal sistemu B celotni 2., 3., in 4. stolpec matrike M_A, sistem B pa sistemu A celotni 1., 2., in 3. stolpec matrike M_B. Na sliki so omenjeni elementi matrik M_A in M_B odebeljeni.

$$M_A = \begin{array}{c} \\ \alpha \\ \beta \\ \gamma \\ \delta \end{array} \begin{array}{c} \begin{array}{cccc} \alpha & \beta & \gamma & \delta \end{array} \\ \begin{bmatrix} - & \mathbf{-1} & \mathbf{0} & \mathbf{-1} \\ 1 & - & \mathbf{0} & \mathbf{-1} \\ 0 & \mathbf{0} & - & \mathbf{-1} \\ 1 & \mathbf{-1} & \mathbf{0} & - \end{bmatrix} \end{array} \qquad M_B = \begin{array}{c} \\ \beta \\ \gamma \\ \delta \\ \epsilon \end{array} \begin{array}{c} \begin{array}{cccc} \beta & \gamma & \delta & \epsilon \end{array} \\ \begin{bmatrix} - & \mathbf{1} & \mathbf{-1} & 1 \\ \mathbf{0} & - & \mathbf{1} & -1 \\ \mathbf{0} & \mathbf{1} & - & -1 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & - \end{bmatrix} \end{array}$$

Takšna izmenjava ocen zagotavlja, da se pri izračunu stopnje ugleda neke entitete upoštevajo vse ocene, ki jih je ta pridobila v kateremkoli od povezanih sistemov. Tako izračunana stopnja ugleda nam da realnejšo sliko in popolnejši rezultat o ugledu entitete kot ožja izmenjava. Mikavna je za tiste sisteme, ki imajo malo število entitet in se za izračun stopnje zaupanja zanašajo na pridobljene podatke iz drugih sistemov.

Ponovimo, da sistem A od sistema B pridobi ocene, ki so jih podale entitete, ki jih v sistemu A ni. To pomeni, da sistem B razkrije sistemu A tudi take elektronske naslove, ki ji sistem A nima v svojem registru entitet, kar je lahko, kot smo že zapisali, v neskladju z željami in dovoljenji lastnikov teh entitet. Zato je pri tej vrsti izmenjave nujno, da uporabljamo mehanizem skrivanja elektronskih naslovov, kot je npr. uporaba enosmernih zgoščevalnih funkcij iz razdelka 4.5.2. Hkrati pa je potrebno znotraj sistemov omogočiti shranjevanje ocen, pri katerih izvorne entitete ni v registru entitet.

4.6.3 Ožja izmenjava ocen

Ta način izmenjave se od širše razlikuje po tem, da sistem A od sistema B pridobi le tiste ocene, ki so jih podale in prejele entitete, ki jih imata oba sistema v registru entitet. Formalno to zapišemo, da za vsako izmenjano oceno (i, p, o) velja $i \in P \wedge p \in P$. V našem primeru bi sistem A poslal sistemu B vrednosti na presečiščih 2., 3., in 4. stolpca z 2., 3. in 4. vrstico matrike M_A. Sistem B pa bi sistemu A poslal ocene na stičiščih 1., 2., in 3. vrstice s 1., 2., in 3. stolpcem matrike M_B. Omenjeni elementi matrik M_A in M_B so na sliki odebeljeni.

$$M_A = \begin{array}{c} \\ \alpha \\ \beta \\ \gamma \\ \delta \end{array} \begin{array}{c} \begin{array}{cccc} \alpha & \beta & \gamma & \delta \end{array} \\ \begin{bmatrix} - & -1 & 0 & -1 \\ 1 & - & \mathbf{0} & \mathbf{-1} \\ 0 & \mathbf{0} & - & \mathbf{-1} \\ 1 & \mathbf{-1} & \mathbf{0} & - \end{bmatrix} \end{array} \qquad M_B = \begin{array}{c} \\ \beta \\ \gamma \\ \delta \\ \epsilon \end{array} \begin{array}{c} \begin{array}{cccc} \beta & \gamma & \delta & \epsilon \end{array} \\ \begin{bmatrix} - & \mathbf{1} & \mathbf{-1} & 1 \\ \mathbf{0} & - & \mathbf{1} & -1 \\ \mathbf{0} & \mathbf{1} & - & -1 \\ 0 & 1 & 0 & - \end{bmatrix} \end{array}$$

Na mestu je vprašanje, kaj tako pridobljene ocene pomenijo za izračun stopenj zaupanja. Pri širši izmenjavi smo govorili o popolnejšem rezultatu. Kaj pa pri ožji? Zamislimo si, da opazujemo zaprto skupino entitet in njihove medsebojne interakcije, ki potekajo tudi izven našega sistema; npr. v sistemih, s katerimi smo povezani. Z ožjo izmenjavo od povezanih sistemov pridobimo informacije o interakcijah te množice entitet. Dodatnih težav – kot so razkrivanje elektronskih naslovov in podpora shranjevanju ocen, ki so jih podale entitete, ki jih ni v našem registru – ni, saj že imamo vse podatke o entitetah, o katerih zbiramo ocene. V aplikaciji *Distributed trustGuard* je podprt način ožje izmenjave.

4.7 Zaključek

Pogledali smo si osnovna modela za povezavo sistemov za upravljanje zaupanja, opisali prednosti in omejitve obeh modelov ter izpostavili ključne dileme, na katere naletimo pri njihovi izgradnji, kot so predpomnenje, učinkovit identifikacijski mehanizem in strategije izmenjevanja ocen. V vsakem razdelku smo zapisali, kako so zgoraj omenjene dileme razrešene v implementirani aplikaciji.

Poglavje 5

Zaključek

Pomen zaupanja pri elektronskem poslovanju narašča. V uporabi je mnogo različnih modelov, ki imajo en skupen cilj – zmanjšati tveganje oziroma povečati verjetnost, da se izbere pravilna odločitev. Vsi modeli se pri izračunu zanašajo na čim večje število podanih ocen, saj lahko tako izračunajo zanesljivejše stopnje zaupanja.

V diplomskem delu so predstavljeni koncepti zaupanja, ugleda in njunega upravljanja, predstavljeno je razlikovanje med tradicionalnim in modernim pojmovanjem zaupanja ter nekateri trenutno najpogosteje uporabljeni modeli za upravljanje zaupanja. Posebna pozornost je namenjena proučitvi arhitekturnega modela odjemalec-strežnik ter modela vsak-z-vsakim.

Lahko bi zapisali, da smo v drugem in tretjem poglavju analizirali trenutno stanje in zbirali znanje drugih, da smo lahko v četrtem predlagali dva lastna tipa rešitev integracije globalno porazdeljenih sistemov za upravljaje zaupanja, in sicer centraliziran sistem ter sistem vsak-z-vsakim. Ugotovili smo, da ima vsak pristop svoje prednosti in omejitve. Opažamo, da so centralizirani sistemi primerni za ustaljeno poslovno sodelovanje, saj zahtevajo precejšnjo stopnjo zaupanja med sodelujočimi sistemi ter zaupanja v centralni strežnik in pripravljenost za popolno deljenje podatkov o entitetah in ocenah vseh vpletenih. Zadnji pomislek vzbuja dvome v množičnost uporabe centraliziranega modela, saj v praksi opažamo, da pripravljenost deljenja ocen z drugimi ni vedno prisotna. Dodatni omejitvi centraliziranega modela predstavljata še pomanjkanje skalabilnosti in robustnosti. Model vsak-z-vsakim na drugi strani zahteva kompleksnejšo zgradbo povezanih sistemov, saj zahteva od vsakega vrstnika, da realizira svoj del sistema za upravljanje zaupanja. Vendar s tem modelom pridobimo večjo robustnost, skalabilnost ter svobodo sodelovanja. V nasprotju s centraliziranim sistemom, za povezavo

z novim vrstnikom v modelu P2P soglasje vseh povezanih vrstnikov ni več potrebno. O tem se morata strinjati le vrstnika, ki se povezujeta. Pri taki shemi povezave sistemov lahko prihaja do odstopanj pri izračunanih stopnjah zaupanja na posameznih vozliščih, saj je jasno, da za izračun ne uporabljajo vsi istih podatkov. Vendar bi izračun pri gosto povezani mreži sistemov moral konvergirati k isti vrednosti pri vseh vrstnikih. Kombiniranja obeh modelov v delu nismo obravnavali, lahko pa trdimo, da je mogoče in je lahko iztočnica kakšnega drugega podobnega dela.

Prav tako smo predlagali možne načine uporabe predpomnilnika z namenom pohitritve delovanja sistema. Narejena diskusija jasno pokaže, da moramo pri njegovi izdelavi in uporabi upoštevati obremenitve sistema pri sinhronizaciji predpomnilnika na eni ter možnost napačnega izračuna stopnje zaupanja, kot posledice uporabe zastarelih podatkov in občutljivosti postopka izračuna stopnje zaupanja, na drugi strani.

Kot enolični identifikator entitet smo predlagali elektronski naslov. Njegovo nadgradnjo predstavljajo izvlečki, ki jih dobimo z uporabo enosmernih zgoščevalnih funkcij. S tem zagotovimo večjo stopnjo zasebnosti ter zmanjšamo možnost zlorabe. Na koncu pa smo definirali dve možnosti izmenjave ocen: širšo in ožjo; pri prvi gre za to, da sistem od povezanega pridobi vse ocene, ki so jih pridobile entitete domujoče v obeh sistemih, domicil izvornih entitet (ocenjevalcev) pa ni pomemben, pri drugi pa izmenjamo le tiste ocene, ki so jih podale entitete domujoče v obeh sistemih, za entitete, ki prav tako domujejo v obeh sistemih.

Nenazadnje smo v Javanskem okolju s pomočjo obilice odprtokodnih orodij realizirali sistem za upravljanje zaupanja, ga namestili na več strežnikov in jih uspešno povezali z uporabo modela vsak-z-vsakim.

Področja, ki niso v tem delu obravnavana, vsekakor pa sodijo v to problemsko domeno, so možnosti kombiniranja centraliziranega modela in modela vsak-z-vsakim, možnost dodeljevanje uteži posameznim sistemom, ker lahko ocenam iz nekega sistema zaupamo bolj kot drugim, možnost podpore različnih modelov za upravljanje zaupanja ter iskanje metod pretvarjanja ocen iz enega modela v drugega, saj bi lahko tako povezali sisteme, ki podpirajo različne vrste modelov za upravljanje zaupanja.

Napisanemu navkljub pa se je potrebno zavedati, da tovrstni modeli povezovanja ne bodo zaživeli, v kolikor ne bo volje in pripravljenosti deljenja ocen. Potreben bo miselni preskok v glavah lastnikov sistemov. Če dva različno velika sistema združita podatke, povečata natančnost izračunanih stopenj zaupanja. Večji sistem prispeva več podatkov kot manjši in v manjši meri izboljša natančnost izračuna v primerjavi z manjšim. Vendar se natančnost

poveča in večja kot bi bila v stanju nepovezanosti. Večji sistem prispeva več ocen in pridobi manj natančnosti. A vseeno pridobi! Lastnikom sistemov je prepuščena presoja, ali je takšna pridobitev vredna deljenja ocen. Za dobrobit uporabnikov in z vidika izboljšave natančnosti lastnega sistema, je avtor tega besedila v to prepričan.

Dodatek A

Distributed trustGuard

A.1 Opis in funkcije

V okviru diplomske naloge je bila izdelana spletna aplikacija Distributed trustGuard. Aplikacija je v funkcionalnem smislu nadaljevanje dela na aplikaciji trustGuard, ki je bila izdelana v okviru doktorske disertacije dr. Damjana Kovača [1], in predstavlja sistem za upravljanje ugleda ter implementira kvalitativni model zaupanja. Osnovne funkcije, ki jih realizira so:

(a) oddaja ocene o posamezni entiteti,

(b) poizvedba ocen posamezne entitete,

(c) poizvedba ugleda posamezne entitete,

(d) ročna in avtomatizirana sinhronizacija ocen s povezanim sistemom za upravljanje zaupanja in

(e) administracija aplikacije in njenih podatkov.

Aplikacija kot tudi celotna izvorna koda aplikacije se nahaja na priloženem CD-ju.

A.2 Uporabljene tehnologije

Aplikacija je realizirana v programskem jeziku Java Enterprise Edition[1] in za izvajanje potrebuje aplikacijski strežnik. V fazi razvoja je bil vseskozi

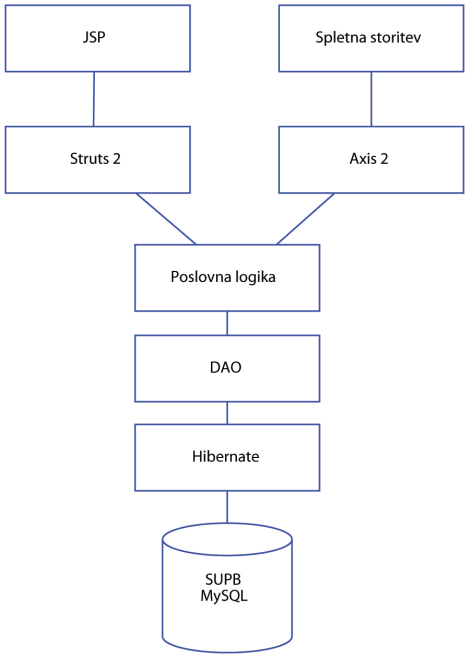
[1] http://java.sun.com/javaee

uporabljen odprtokodni aplikacijski strežnik Tomcat 6[2], vendar je mogoče aplikacijo pognati na kateremukoli javanskemu aplikacijskemu strežniku. Persistenca podatkov se zagotavlja s pomočjo odprtokodnega sistema za upravljanje s podatkovno bazo MySQL[3] ter prav tako odprtokodne tehnologije Hibernate[4] za preslikavo med objektnim in relacijskim modelom.

Spletni vmesnik je zgrajen v tehnologiji Servlet[5] ter odprtokodnega ogrodja Struts 2[6], spletne storitve pa so realizirane v ogrodju Axis 2[7], ki je prav tako odprtokodno.

A.3 Arhitektura

Visokonivojska arhitektura oz. način kako so glavne komponente programa sklopljene med seboj, si lahko ogledamo na sliki A.1.



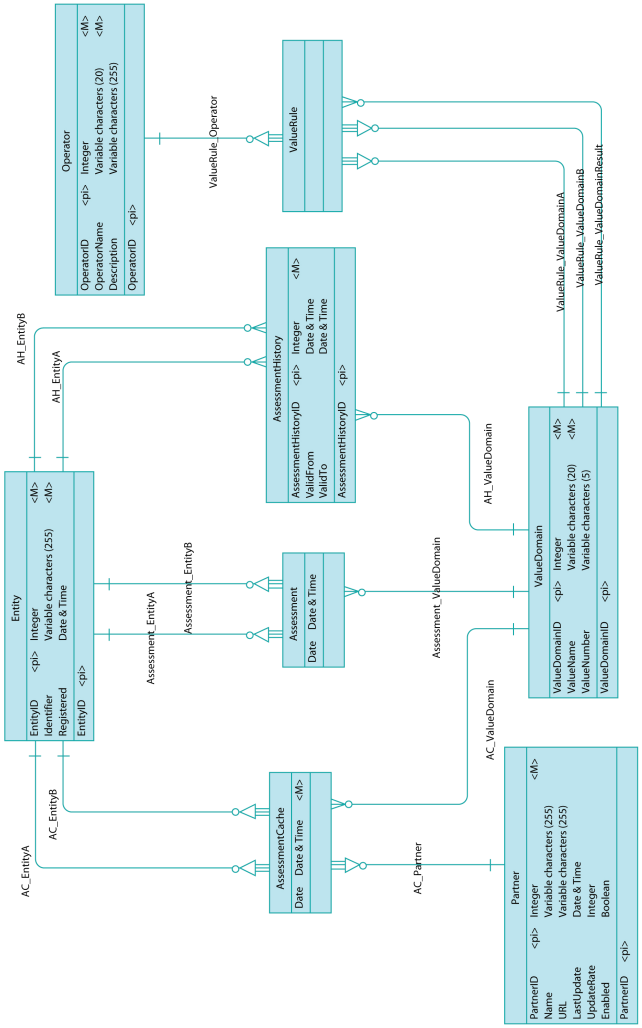
Slika A.1: Arhitektura spletne aplikacije.

[2]http://tomcat.apache.org
[3]http://www.mysql.com
[4]http://www.hibernate.org
[5]http://java.sun.com/products/servlet
[6]http://struts.apache.org
[7]http://ws.apache.org/axis2

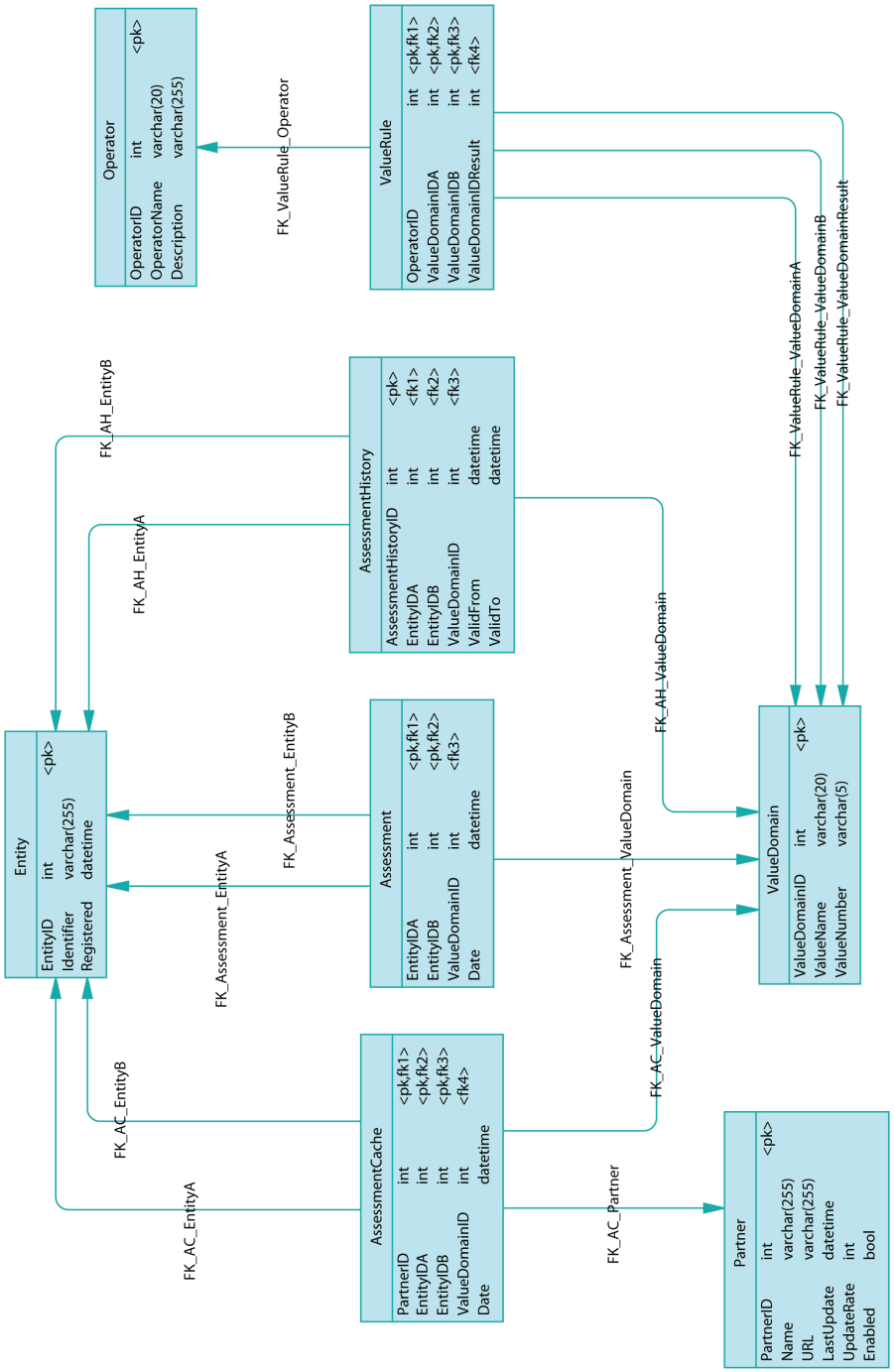
A.4 Podatkovni model

Omenili smo že, da je podatkovna persistenca zagotovljena s pomočjo sistema za upravljanje s podatkovno bazo MySQL. Konceptualni ter logični model sta podana na slikah A.2 in A.3. Podatkovni model in poizvedbe so zgrajeni v skladu s SQL standardom, kar pomeni, da jih je (teoretično) mogoče realizirati v poljubnem SUPB-ju.

Še enkrat lahko omenimo, da preslikavo med objektnim in relacijskim modelom ter visoko stopnjo prenosljivosti (angl. portability) zagotavlja ogrodje Hibernate.



Slika A.2: Konceptualni podatkovni model.



Slika A.3: Logični podatkovni model za SUPB MySQL.

A.5 Vmesnik spletne storitve

Distributed trustGuard se lahko povezuje z drugimi sistemi za upravljanje zaupanja. Vse kar mora storiti sistem, ki želi sinhronizirati ocene z DTG, je poklicati metodo spletne storitve povezanega sistema DTG. Na vhodu mu poda seznam nizov, ki predstavljajo identifikatorje entitet (e-mail naslovi), kot rezultat pa prejme seznam trojic (i, p, o)[8]. Sledi ogled SOAP zahteve in odgovora nanjo.

XML A.1: Klic metode getAssessments.

```
<ns1:getAssessments xmlns:ns1='http://ws.dtg'>
  <ns1:identifiers>djelenc@gmail.com</ns1:identifiers>
  <ns1:identifiers>info@siol.net</ns1:identifiers>
  <ns1:identifiers>info@fri.uni-lj.si</ns1:identifiers>
</ns1:getAssessments>
```

XML A.2: Odgovor na zahtevo spletne storitve.

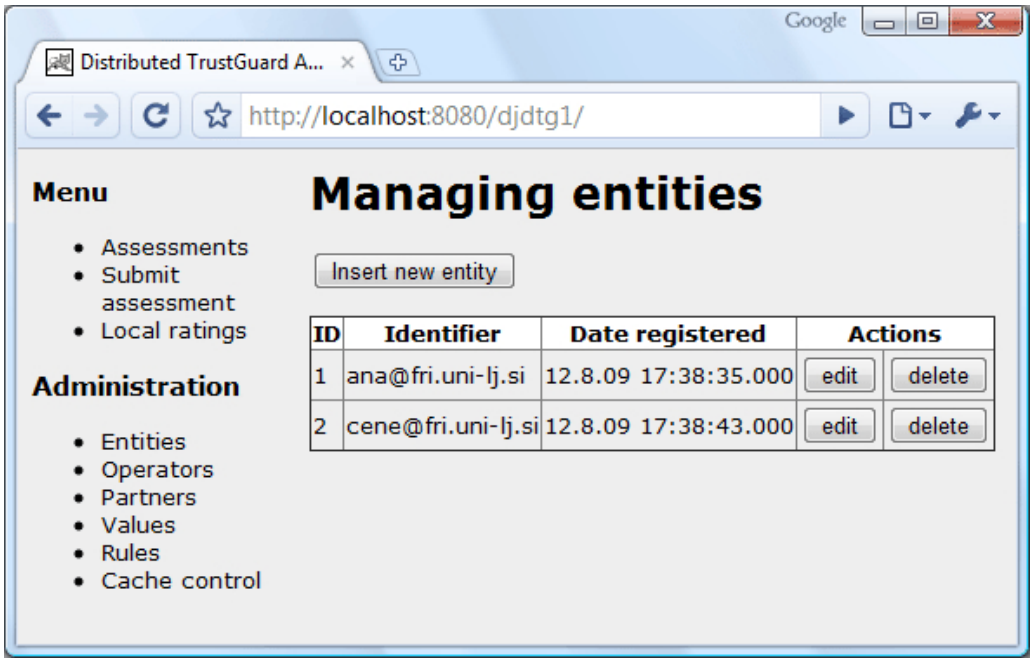
```
<ns:getAssessmentsResponse xmlns:ns='http://ws.dtg'
  xmlns:ax21='http://dto.dtg/xsd'>
  <ns:return type='dtg.dto.AssessmentWS'>
    <ax21:date>2009-07-21T14:10:15.000Z</ax21:date>
    <ax21:source>info@fri.uni-lj.si</ax21:source>
    <ax21:target>info@siol.net</ax21:target>
    <ax21:value>0.5</ax21:value>
  </ns:return>
  <ns:return type='dtg.dto.AssessmentWS'>
    <ax21:date>2009-07-21T16:35:01.000Z</ax21:date>
    <ax21:source>info@siol.net</ax21:source>
    <ax21:target>djelenc@gmail.com</ax21:target>
    <ax21:value>1</ax21:value>
  </ns:return>
  <ns:return type='dtg.dto.AssessmentWS'>
    <ax21:date>2009-08-07T12:31:19.000Z</ax21:date>
    <ax21:source>djelenc@gmail.com</ax21:source>
    <ax21:target>info@siol.net</ax21:target>
    <ax21:value>-0.5</ax21:value>
  </ns:return>
</ns:getAssessmentsResponse>
```

[8]Trojica je v implementaciji dejansko četverček, saj vsaki trojici dodamo še datum nastanka ocene, kajti za vsako prenešeno oceno moramo vedeti, kdaj je nastala.

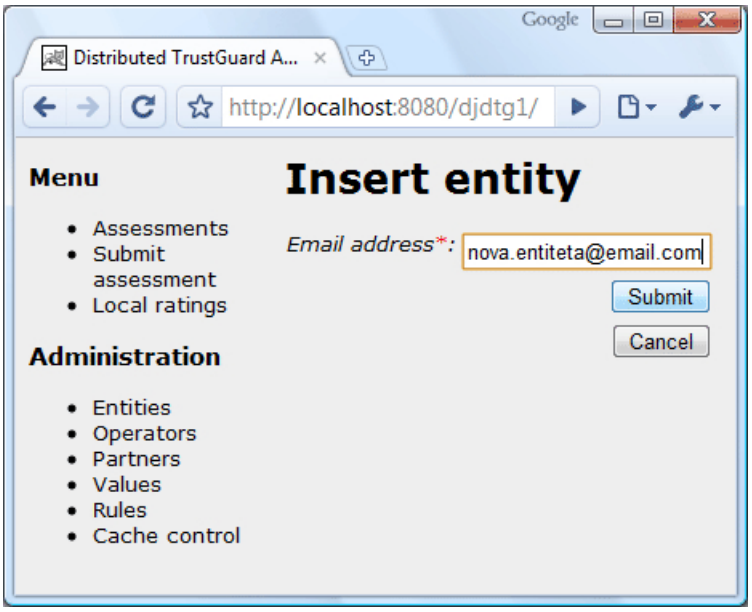
A.6 Spletni vmesnik

Spletni vmesnik služi za administracijo aplikacije. V tej različici se z njegovo pomočjo tudi oddajajo ocene in sestavljajo poizvedbe po ocenah in ugledu entitet. V praksi pa bi tak sistem bilo smotrno integrirati z namensko aplikacijo (kot je aplikacija za spletno prodajo in licitiranje, spletni forum, spletna svetovalnica ali kakšen drug sistem, ki potrebuje sistem za upravljanje zaupanja), v kateri bi *Distributed trustGuard* postal zgolj del celote takega sistema.

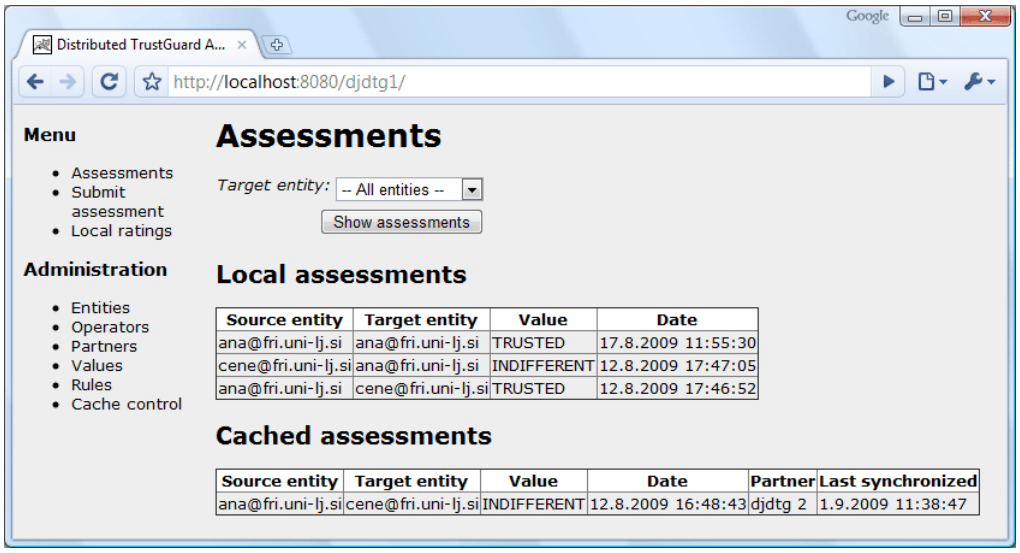
V administraciji urejamo register (šifrant) entitet, upravljamo operatorje in vrednosti kvalitativne algebre ter pravila za izračun stopenj zaupanja, upravljamo seznam partnerskih sistemov, s katerimi se naš sistem povezuje, ter ročno osvežimo vsebino predpomnilnika ali nastavimo periodo avtomatičnega osveževanja in ga poženemo.



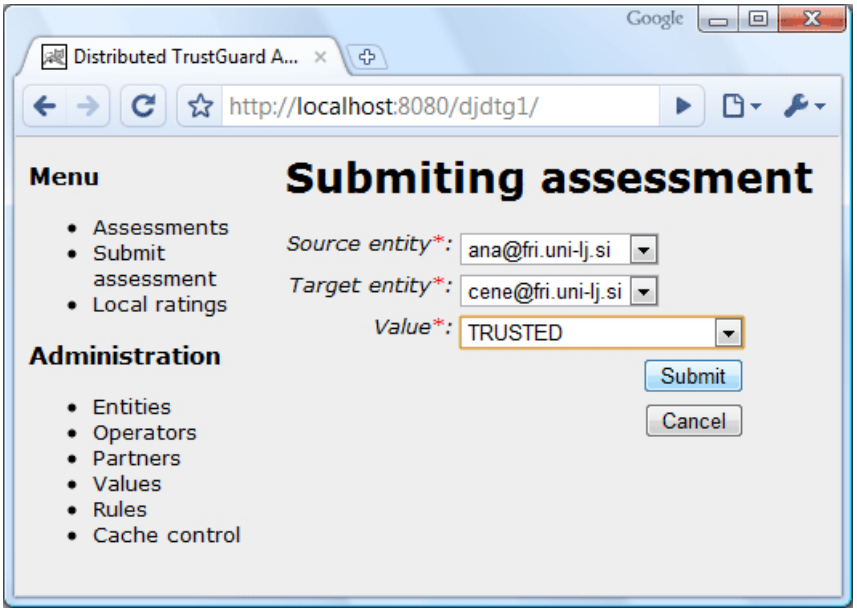
Slika A.4: Pregled nad entitetami.



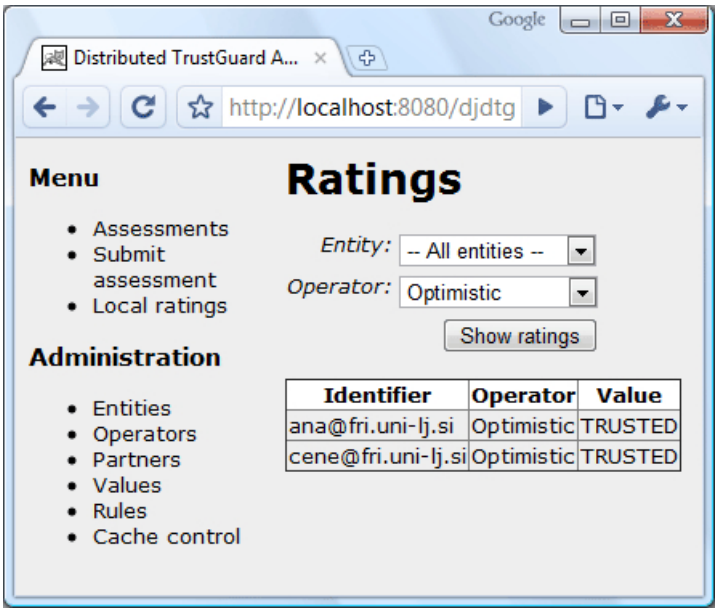
Slika A.5: Vnašanje novih entitet.



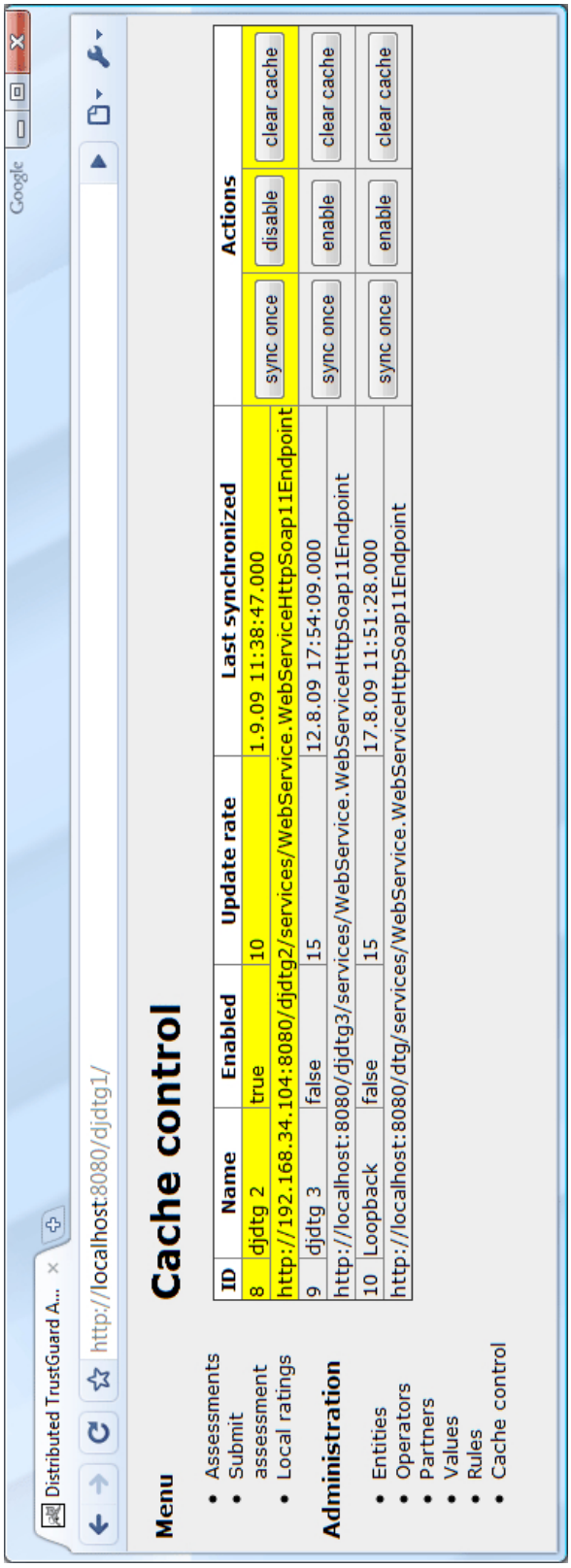
Slika A.6: Prikaz vseh ocen.



Slika A.7: Vnašanje ocene.



Slika A.8: Izračunane lokalne stopnje ugleda.



Slika A.9: Upravljanje predpomnilnika.

Slike

Tabele

Literatura

[1] Kovač, D.: Obvladovanje zaupanja v storitveno usmerjenih arhitekturah. Doktorska disertacija. Fakulteta za računalništvo in informatiko, Univerza v Ljubljani. Ljubljana, 2009.

[2] Trček, D.: Towards trust management standardization. Computer Standards & Interfaces, 26 (6): 543–548, 2004.

[3] Trček D.: Managing Information Systems Security and Privacy. Springer Verlag. Heidelberg, 2006.

[4] Dimitrakos, Bicarregui: Towards A Framework for Managing Trust in e–Services. Proceedings of the Fourth International Conference on Electronic Commerce Research, ATSMA, IFIP, INFORMS 2, strani 360–381, Dallas, Texas, ZDA, 2001.

[5] Jøsang, A.: A logic for uncertain probabilities. International Journal of Uncertainty, Fuzziness and Knowledge–Based Systems, 9(3): 279–311, 2001.

[6] Zupančič, E.: Obvladovanje zaupanja v informacijskih sistemih s pomocjo simulacij. Diplomska naloga. Fakulteta za računalništvo in informatiko, Univerza v Ljubljani. Ljubljana, 2009.

[7] Jøsang, Ismail, Boyd: A survey of trust and reputation systems for online service provision. Decision Support Systems, 43(2): 618–644, 2007.

[8] Vidmar, T.: Informacijsko-komunikacijski sistem. Ljubljana: Pasadena, 2002.

[9] Resnick, Zeckhauser, Friedman, Kuwabara: Reputation Systems: Facilitating Trust in Internet Interactions. Communications of the ACM, 43 (12): 45-48. 2006.

[10] International Standards Organization: IT - Security techniques: Information security management systems: Requirements. ISO/IEC 27001. Geneva, 2005

[11] Zakon o varstvu osebnih podatkov (ZVOP-1). 2004. Uradni list Republike Slovenije, 86 (5. 8. 2004).

[12] ZRC SAZU: Slovar slovenskega knjižnega jezika. Spletna izdaja. Slovenska akademija znanosti in umetnosti, Znanstvenoraziskovalni center Slovenske akademije znanosti in umetnosti, Inštitut za slovenski jezik Frana Ramovša ZRC SAZU, 2000. Dostopno na naslovu http://bos.zrc-sazu.si/sskj.html.