

ŠIAULIŲ UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Remigijus Sabestinas
Informatikos specialybės II kurso neakivaizdinio skyriaus studentas

**PASAULINIO TINKLO PASLAUGOS (WEB-SERVISAI)
GRID TINKLUOSE**

WEB-SERVICES AND GRID'S

MAGISTRO DARBAS

Darbo vadovas:
Lekt. V.Giedrimas

Recenzentas:
Lekt. V. Sirius

Šiauliai, 2009

Turinys

I. Įvadas	3
I.1 Praktinė reikšmė	3
I.2 Tikslas	3
II. Teorinė dalis	4
II.1 Temos analizė	4
II.1.1 Pasaulinio tinklo paslaugos Grid tinkluose	4
II.2 Darbo srities analizė	5
II.2.1 Į paslaugas orientuota architektūra	5
II.2.2 Pasaulinio tinklo paslaugos	8
II.2.3 Pasaulinio tinklo paslaugos ir virtualizacija	9
II.2.4 Grid tinklai	10
II.3 Darbo srities modelis	13
III. Projektinė dalis	14
III.1 Įrankių ir priemonių pasirinkimo analizė	14
III.1.1 Agentų bendravimo kalba (ACL)	14
III.1.2 Grid paslaugų architektūra	15
III.1.3 Duomenų tipai	16
III.1.4 XML programavimo kalba	17
III.1.5 Globus Toolkit	17
IV. Darbo eigos aprašymas	19
IV.1 Darbų eigos grafas	19
IV.2 Problemos ir jų sprendimas	20
IV.2.1 Transporto lygmuo	20
IV.2.2 Paslaugų pateikimo lygmuo	20
IV.3 Galutinio projekto stovio aprašymas	22
IV.4 Darbo rezultatai	26
IV.5 Patarimai, pastebėjimai, rekomendacijos	26
V. Išvados	27
VI. Literatūra	28
VII. Anotacija	29
VII.1 Annotation	29
VIII. Priedas A	30

I. Įvadas

Paskirstytų sistemų naudojimas įmonėse ir akademinėse organizacijose pastaraisiais metais padidėjo proporcingai didėjant prieigų prie interneto ir pasaulinių tinklų dėl HTTP protokolo subrendimo ir saugios komunikacijos technologijų pagerinimo. Paskirstymo stiliai aprėpia grupes nuo vietinių vienasrūšių sistemų, veikiančių kaip klasteris globalinio masto Grid tinkluose iki gausaus kiekio agentų, sudarytų iš didelių sistemų skaičiaus ir palaikančių operacinių aplinkų asortimentą.

Anksčiau, paskirstytos sistemos, bendradarbiaudavo naudodamos tam priklausančius protokolus ir sistemų administratoriai naudojo specialius metodus, kad valdytų sistemas, kurios galėjo būti kitame mieste ar netgi kitame pasaulio krašte. Daug standartų buvo išvystytų per keletą metų, kad palengvintų sėkmingą šių paskirstytų sistemų aptarnavimą ir palaikymą. Šiandien, pagrindinės technologijos paskirstytose sistemose yra į paslaugą orientuotos architektūros (SOA), pasaulinio tinklo paslaugos ir lygiagrečiųjų ir paskirstytų skaičiavimų tinklai, iš kurių visi pastebi reikšmingas investicijas į visų tipų ir dydžių organizacijų standartizaciją naudojant šias naujas technologijas.

I.1 Praktinė reikšmė

Šis projektas nukreips pradedančiuosius teisinga linkme pasaulinio tinklo paslaugų Grid tinkle kūrime, padės nepasimesti technologijų gausoje, kurios reikalingos paslaugoms.

I.2 Tikslas

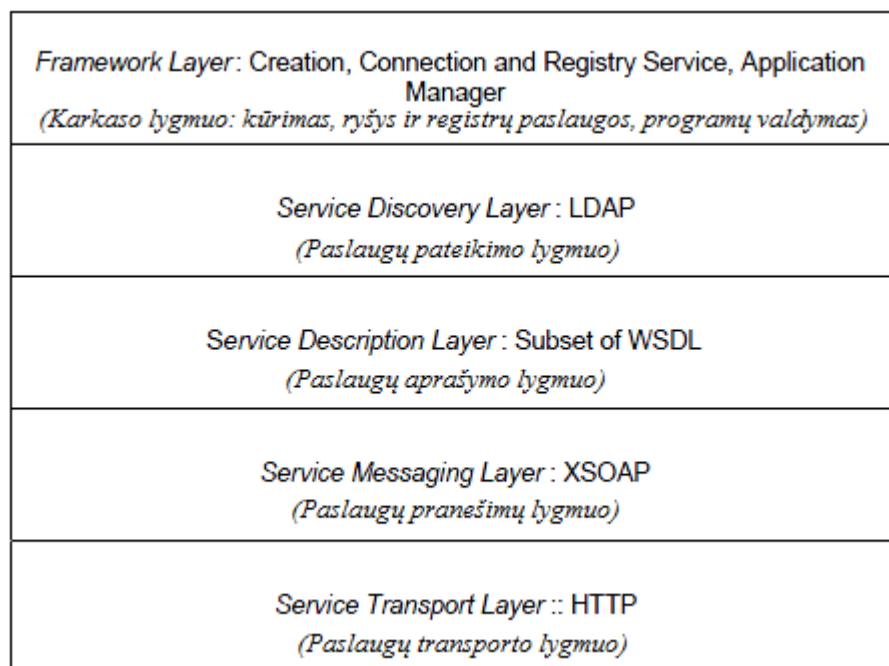
Šio projekto pagrindinis tikslas yra: išanalizuoti pasaulinio tinklo paslaugų ir Grid tinklų veikimą nuo žemiausių lygių ir pabandyti sukurti modelį, kuriuo remiantis būtų galima kurti pasaulinio tinklo paslaugas Grid tinkle, taip pat sukurti demonstracinę Grid tinklo paslaugą.

II. Teorinė dalis

II.1 Temos analizė

II.1.1 Pasaulinio tinklo paslaugos Grid tinkluose

Svarbiausias dabartinis vystymasis Grid sistemose yra pasaulinio tinklo paslaugų modelio adaptavimas ir naudojimas Grid sistemoje, kaip pagrindinė architektūra Grid paslaugoms. Kiekviena Grid paslauga yra apibrėžta pasaulinio tinklo paslaugų apibūdinimo kalba (WSDL) ir prieinama per protokolą, kaip jungtis paslaugai. Dabartinė tendencija Grid tarpinėje įrangoje yra įsisavinti sudarytą pasaulinio tinklo paslaugų modelį [1]:



1 pav. (Pasaulinio tinklo paslaugų modelis)

Norint suprasti pasaulinio tinklo paslaugų ir Grid tinklų veikimą, jų integraciją ir realizaciją, reikia išnagrinėti tokius terminus kaip į servisą orientuota architektūra, į servisą orientuoti Grid tinklai, virtualizacija, abstrakčios komunikacijos architektūra, agentų bendravimo kalba ir susieti su pasaulinio tinklo paslaugų modeliu. Iš šio modelio mums esminės skiltys bus:

- 1 Transporto Lygmuo (angl. Transport Layer) – kaip veikia Grid agentai ir pasaulinio tinklo paslaugos šiame lygmenyje.
- 2 Paslaugų Pateikimo Lygmuo (angl. Service Discovery Layer) – registru mechanizmai pasaulinio tinklo paslaugose ir Grid tinkluose.

II.2 Darbo srities analizė

II.2.1 Į paslaugas orientuota architektūra

Į paslaugas orientuota architektūra (angl. Service-Oriented Architecture) susideda iš tokių terminų [5]:

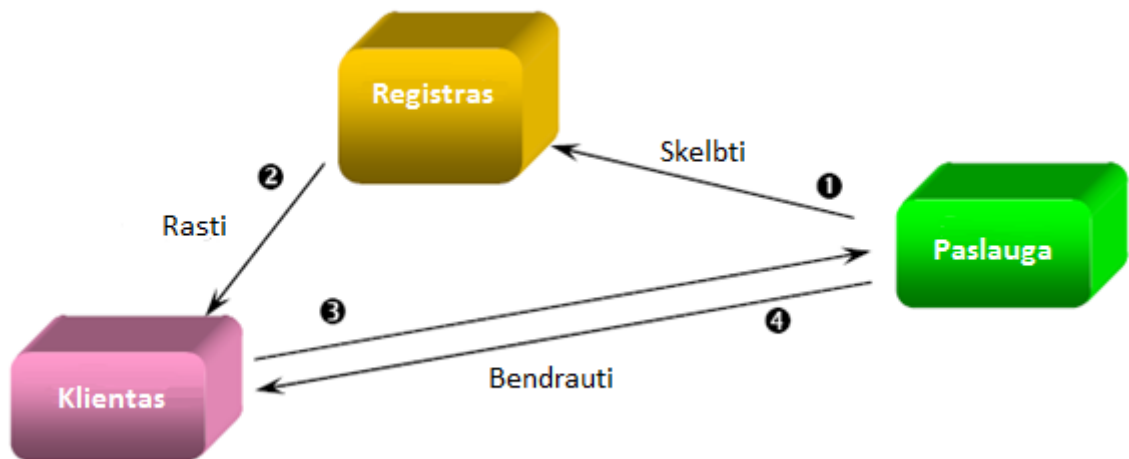
- Architektūra – oficialus sistemos apibūdinimas, apibrėžia jos tikslą, funkcijas, išoriškai matomas ypatybes ir sąsajas. Tai taip pat apima apibūdinimą sistemos vidaus komponentų ir jų santykių.
- Paslauga – programinės įrangos komponentas, prie kurio gali gauti prieigą per tinklą, kad aprūpintų funkcionalumu paslaugos klausėją (angl. requester).
- Į paslaugą orientuotos architektūros siejasi su stiliumi statyti patikimas ir paskirstytas sistemas, kurios tiekia funkcionalumą kaip paslaugas.

Įvertiname SOA, kaip architektūrinį stilių, kuris pabrėžia komponentų realizacijas, kaip modulinės paslaugas, kurios gali būti atrastos ir panaudotos klientų. Paslaugos apskritai turi šias savybes [5, 6]:

- Paslaugos gali būti individualiai naudingos, arba jos gali būti integruotos, sukomponuotos, kad teiktų aukštesnio lygmens paslaugas. Kita privilegija yra ta, kad tai skatina egzistuojančio funkcionalumo pakartotinį naudojimą.
- Paslaugos susisiečia su jų klientais, keisdamos žinutes: jie yra apibrėžti žinučių, kurias jie gali priimti ir atsakymai, kuriuos jie gali duoti.
- Paslaugos gali dalyvauti darbų eigoje, kur užsakymas, kuriame žinutės nusiunčiamos ir gaunamos, paveikia įvykdytų paslaugų operacijų rezultatą. Šis supratimas yra apibrėžtas kaip „paslaugų choreografija“.
- Paslaugos gali būti visiškai save valdančios arba jos gali priklausyti nuo tinkamumo kitų paslaugų, arba nuo egzistavimo išteklių tokių kaip duomenų bazė. Paprasčiausiu atveju paslauga galėtų įvykdyti skaičiavimą, tokį kaip duoto skaičiaus kubo šaknies skaičiavimas, negalėdama sietis su bet kokiais išoriniais ištekliais. Priešingai, paslaugai, kuri įvykdo valiutos pakeitimą reikėtų skubiai atliekamos prieigos prie valiutos kurso informacijos, kad duotų teisingas vertes.

- Paslaugos reklamuoja detales tokias kaip jų gebėjimas, sąajos, politika ir palaikomi komunikacijos protokolai. Interpretavimo detalės tokios kaip programavimo kalba ir išteklių (angl. hosting) priėmimo platforma nekelia jokių rūpesčių klientams.

Iliustracija (2 pav.) iliustruoja paprastą paslaugos sąveikos ciklą, kuris prasideda nuo paslaugos, reklamuojančios save savarankiškai per žinomą registrą paslaugą (1). Potencialus klientas, kuris gali būti ir kita paslauga, klausinėja registrą (2), kad surastų paslaugą, kuri tenkina jo norus. Registras grąžina (galbūt tuščią) sąrašą tinkamų paslaugų ir klientas išsirenka vieną, kuriai perduoda prašymo žinutę tam, naudodamas bet kokį bendrai pripažintą (3) protokolą.



2 pav. (Paslaugos sąveikos ciklas)

Paslaugos sąveika į paslaugą orientuotoje aplinkoje, kurią iliustracija rodo paprasčiausiam atvejui, bet realiajame pasaulyje, tokia kaip komercinė paraiška procesas gali būti žymiai sudėtingesnė. Pavyzdžiui, duota paslauga gali palaikyti tikrai HTTPS2 protokolą, būti apriboti vartotojų lygiai. Reikalingas autentiškumo nustatymas, skirtingi atlikimo lygmenys skirtingiems vartotojams. Paslaugos gali pateikti daug informacijos apie save ir klientas gali panaudoti šią informaciją, kad padarytų jos pasirinkimą. Tam tikriems požymiams, tokiems kaip užmokesčio terminai ir garantuoti paslaugos lygmenys, gali būti, kad reikia nustatyti derybų procesą anksčiau, negu klientas galės panaudoti paslaugą, kurią išsirinko. Ir, tuo metu, kai ši iliustracija rodo paprastą sinchroninę, dvikryptę žinutės keitimo struktūrą, struktūros įvairovė yra įmanoma - pavyzdžiui, sąveika gali būti vienu keliu, arba atsakymas gali atvykti ne iš paslaugos, į kurią klientas nusiuntė prašymą, bet nuo kažkokios kitos paslaugos, kuri užbaigė sandorį.

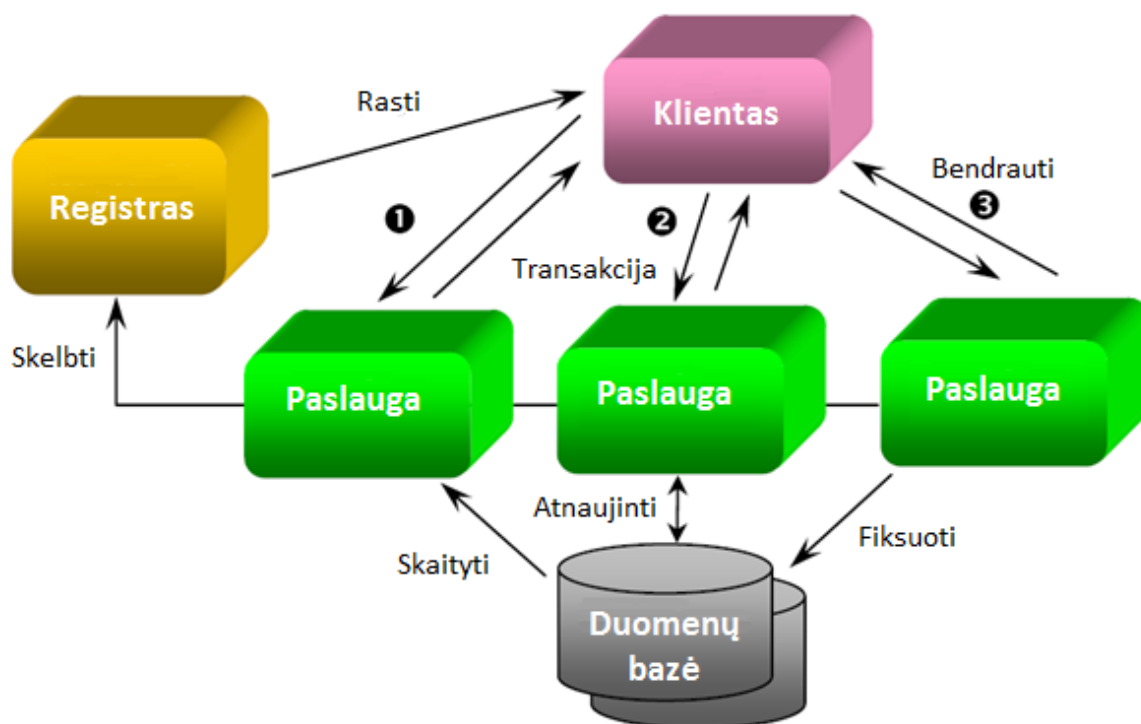
SOA privalumai [5]:

- Lankstumas: paslauga gali būti ant bet kokio serverio ir perkelta kai būtina. Iki tol, kol ji prižiūri savo registrų įėjimą, numatomi klientai sugebės surasti ją.
- Dydžio keičiamumas: paslaugos gali būti pridėtos ir pašalintos pagal keičiamus reikalavimus.
- Pakeičiamumas: jei originalios sąsajos yra išsaugotos, nauja ar atnaujinta paslaugos realizacija gali būti įvesta be žalingos įtakos vartotojams.
- Klaidos tolerancija: jei serveris, programinės įrangos komponentas ar tinklo segmentas sugenda, ar paslauga tampa neprieinama dėl bet kokios kitos priežasties, klientai gali klausinėti registrą kitų paslaugų, kurios siūlo reikalingą funkcionalumą ir tęsti veiklą be pertraukimo.

Daugiažingsnės transakcijos [5]:

- Kiekvieno tarpinio žingsnio pabaigoje paslauga turi grąžinti klientui pakankamą būsenos informaciją, kad įgalintų bet kokią kompetentingą paslaugą identifikuoti ir tęsti sandorį.
- Klientas turi pasiūsti būsenos informaciją, bet kokiam paslaugų servisui, kad būtų apdirbta kita transakcija.
- Išrinkta paslauga turi sugebėti priimti ir reguliuoti būsenos informaciją, kurią pateikė klientas, nepriklausomai nuo to ar apdorojo ją savarankiškai ankstyvesniuose žingsniuose.

Iliustracija (3 pav.) rodo klientą, užsiėmusį trijų žingsnių sandoriu su keliomis paslaugomis, iš kurių kiekviena sugebėtų elgtis su bet kokia dalimi ar visu sandoriu lygiagrečiai su kitomis. Paslauga, kuri eina su (1) žingsniu kaupia detales sandorio progresų duomenų bazėje ir grąžina informaciją, kurios prašoma, į klientą su sandorio vardu. Klientas galėtų prašyti patvirtinimo iš vartotojo prieš sandorio vardo perdavimą į kitą paslaugą, kuri naudoja šią informaciją, kad gautų būsenos informaciją iš duomenų bazės, tada galima pradėti (2) žingsnį. Ši paslauga tada atnaujina duomenų bazę ir grąžina papildomą informaciją į klientą. Pagaliau, klientas perduoda atgal sandorio vardą į trečią paslaugą su prašymu užbaigti sandorį.



3 pav. (Kliento bendravimas su paslaugomis)

Pavyzdyje (3 pav.) paskyros (angl. account) duomenys ir transakcijos detalės gali būti laikomos kaip būsenos informacija, bet paskyros duomenys yra nuolatiniai, tuo metu, kai transakcijos detalės turi egzistuoti tiksliai tuo metu, kai transakcija yra progrose. Kad sumažinti būsenų kiekį, kurį turi perduoti tarp klientų ir paslaugų, kritiškai pranešimo duomenys ir sandorio detalės yra laikomi duomenų bazėje.

II.2.2 Pasaulinio tinklo paslaugos

Dauguma žmonių yra pažįstami su prieiga prie tinklo per naršyklę, kuri siūlo į vartotoją orientuotą sąsają informacijai ir į vartotoją orientuotoms paslaugoms, tokioms kaip internetiniai aukcionai ir mažmeninės parduotuvės. Kai vartotojas prašo tinklalapio, prašymas yra siunčiamas į nutolusį tinklo serverį, kuris grąžina informaciją hiperteksto kalba (HTML), tai forma, kuri leidžia naršyklei pristatyti informaciją naudojant pasirinktą šriftą, spalvą ir paveikslėlius, visi faktoriai, kurie yra priimtinau vartotojui.

Pasaulinio tinklo paslaugos yra paskirstyti programinės įrangos komponentai, kurie aprūpina informacija programinę įrangą, o ne vartotojus, per programinės įrangos orientuotą sąsają. Informacija yra suformuota naudojant XML kalbą, kad lengviau ir greičiau galėtų būti nagrinėjama

ir apdirbama, o tik po to paruošiama parodymui. Tinklu pagrįstoje mažmeninėje operacijoje, pavyzdžiui, pasaulinio tinklo paslaugos, kurios gali veikti ant plačiai atskirtų serverių galėtų aprūpinti sąskaitybą, inventoriaus kontrolę, apsipirkimą ir kreditinės kortelės leidimo paslaugas, iš kurių visoms gali būti taikomi sinchronizuoti laikai vieno pirkimo metu. Pasaulinio tinklo paslaugos publikuoja savo funkcijų ir sąsajų detales, bet jos laiko savo realizacijos detales privačiai. Tokiu būdu klientas ir paslauga, kurie palaiko bendrus komunikacijos protokolus gali kartu veikti nepriklausomai nuo platformų, ant kurių jos eina, ar programavimo kalbos, kuriose jie yra parašyti. Todėl pasaulinio tinklo paslaugos yra ypač taikomos paskirstytai heterogeninei aplinkai [8].

Pagrindinė naudojama pasaulinio tinklo paslaugų specifikacija yra [9]:

- XML (angl. eXtensible Markup Language) - programavimo kalba žinučių formavimui ir suformuotų duomenų pasikeitimui.
- SOAP (angl. Simple Object Access Protocol) - XML pagrįstas protokolas tam, kad apibrėžtų pranešimų informaciją, jų turinius ir apdirbtą informaciją žinutei.
- WSDL (angl. Web Services Description Language) - XML pagrįsta kalba naudojama apibūdinti požymius, sąsajas ir kitas pasaulinio tinklo paslaugų ypatybes. WSDL dokumentas gali būti perskaitytas potencialaus kliento, kad sužinotų apie paslaugą.

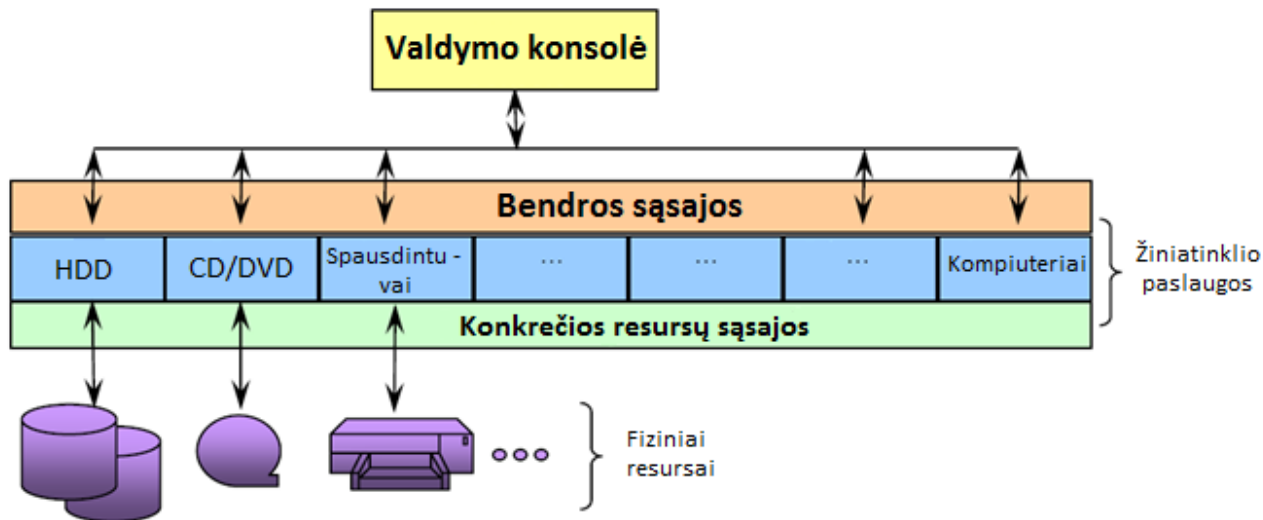
Nors pasaulinio tinklo paslaugos gali palaikyti, bet kokį komunikacijos protokolą, ir gali pasiūlyti jo klientams pasirinkimą, tačiau bendras yra SOAP per HTTP ar HTTPS. Tai prisideda prie pasaulinio tinklo paslaugų populiarumo, kadangi HTTP ir HTTPS yra plačiai naudojami ir tipiškai nekelia problemų su ugniasienėmis organizacijose, kurios leidžia dvikryptį HTTP srautą.

II.2.3 Pasaulinio tinklo paslaugos ir virtualizacija

Lengvumas, su kuriuo pasaulinio tinklo paslaugos gali būti įgyvendintos ir gebėjimas gauti prieigą prie jų nuo bet kokios platformos, vietinės ar nutolusios, privedė prie jų greito priėmimo prie sistemos valdymo sąsajų, tokių kaip virtualizacijos agentai, kurie suteikia bendras valdymo sąsajas skirtingiems ištekliams. Pavyzdžiui, pasaulinio tinklo paslaugos galėtų būti suprojektuotos, kad atstovautų tam tikram prietaisui, priimdamos kontrolę ir kontroliuodamos prašymus per standartizuotas sąsajas, susisiekdamos su ištekliais per jo gimtąją sąsają ir grąžindamos rezultatą užklausėjui standartiniu formatu.

Iliustracija (4 pav.) iliustruoja, kaip žinantis standartus vadybos pultas (administratorius) gali valdyti komplektą nepanašių išteklių per pasaulinio tinklo paslaugų virtualizacijos sluoksnį, iš kurių

kiekviena naudoja standartizuotas sąsajas, kad susisiektų su jungtiniais ištekliams per ištekliams priklausančias sąsajas. [5]

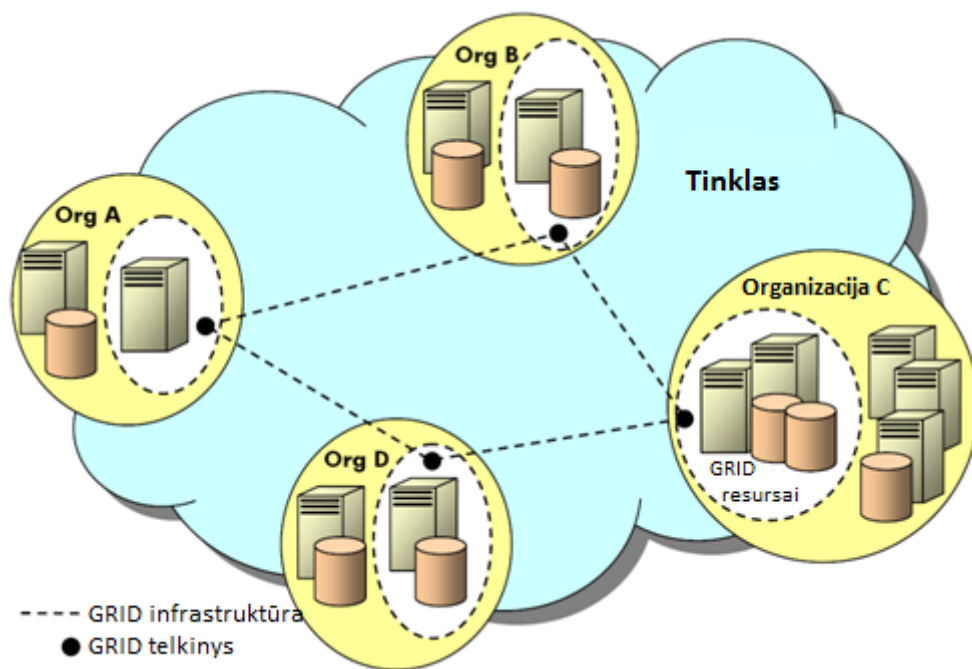


4 pav. (Virtualizacija)

II.2.4 Grid tinklai

Grid skaičiavimų tinklai yra paskirstyto skaičiavimo forma, kurie naudoja skirtingus išteklius tokius kaip skaičiavimo mazgus, atmintį, taikomas programas ir duomenis, dažnai sklinda per skirtingas fizines vietas ir administracines sritis, yra optimizuoti per virtualizaciją ir kolektyvinį valdymą. Grid yra dažnai klasifikuojami kaip skaičiavimo tinklai, kurie pasižymi naudojamų skaičiavimo išteklių dalinimusi arba tinklai, kurie palaiko duomenų išteklių gavybą. Šitie skirtingumai daugiausia diktuoja aparatinės įrangos infrastruktūros tipą - pavyzdžiui mazgams ant duomenų tinklų, galbūt, reikia suteikti plačiąjuosčio tinklo prieigą, tuo metu, kai Grid, kurių pirminis naudojimas yra ilgai paraleliai veikiančios aplikacijos yra daugiau reikalingi aukšto atlikimo skaičiavimo mazgai. Praktiškai, Grid yra linkęs atskleisti įvairių funkcinių klasifikacijų savybes ir jų mastas linkęs ilgai didėti.

Iliustracija (5 pav.) iliustruoja Grid, prie kurio prisidėjo keturios susijusios organizacijos savo skaičiavimo ir laikymo ištekliams. Kiekviena organizacija turi naudoti saugumo procedūras, kad sutrukdytų nesankcionuotai prieigai prie jos privačių Grid vartotojų išteklių. [5]



5 pav. (Grid tinklas)

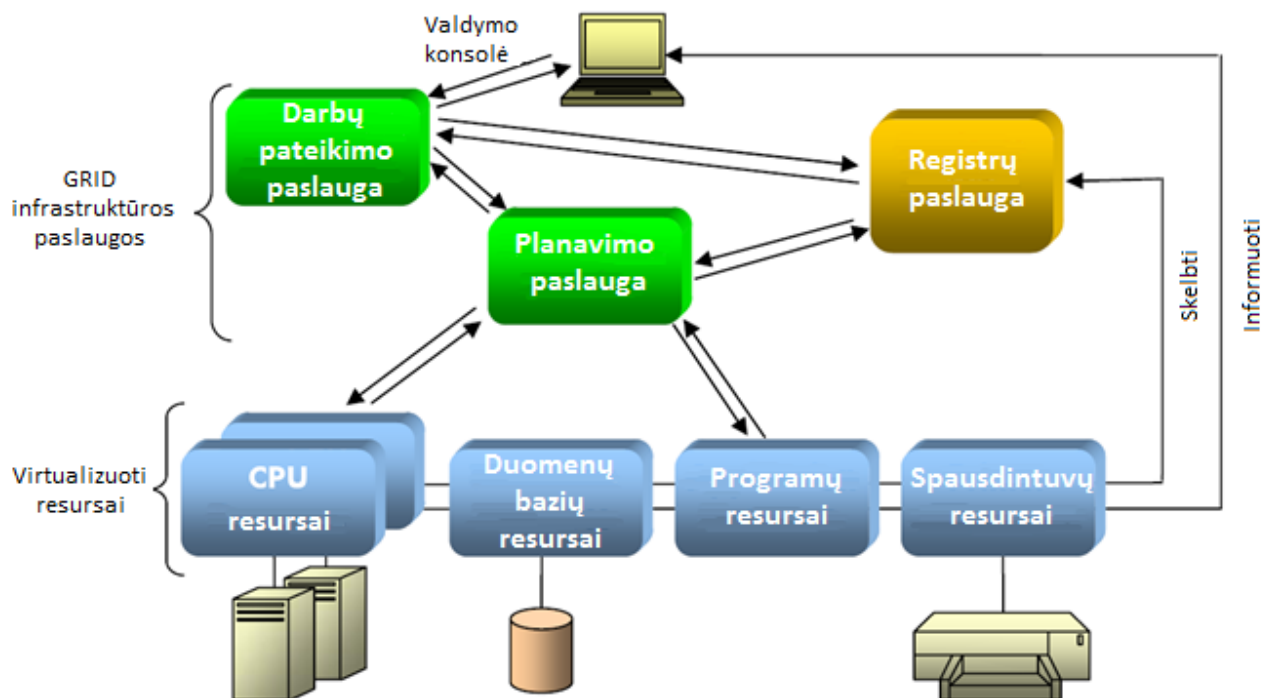
Nors Grid skaičiavimas buvo sugalvotas tyrinėjimo organizacijose, kad palaikytų skaičiavimus intensyvioms mokslinėms aplikacijoms ir turėtų dažnai masinę tyrinėjimo duomenų bazę, visų tipų ir dydžių įmonės pradeda pripažinti technologiją kaip fondą lanksčiai vadybai ir naudojimui jų vidaus IT ištekliams, įgalindamos juos geriau spręsti verslo tikslus, tokius kaip kainos mažinimas, judrumo didinimas ir bendradarbiavimas.

Grid skaičiavimas yra tokiu būdu fondas bendram aukšto atlikimo skaičiavimui ir duomenų pasidalinimui, adaptyviai įmonei, ir vizijai naudoti skaičiavimo, kuriame skaičiavimo ištekliai tampa mokesčio per naudojimą prekėmis. Tuo metu, kai tikslus apibrėžimas yra svarstytas, bendros Grid savybės yra, kad jie yra linkę būti stambūs ir plačiai paskirstyti, reikalingas decentralizuotos vadybos, apimti gausius heterogeninius išteklius, ir turėti trumpalaikius vartotojų gyventojus. Vadinasi Grid iliustruoja reikalingumą labai keičiamo dydžio, patikimos, nepriklausomos nuo platformos architektūros, kuri palaiko saugią operaciją ir standartizuotas sąsajas bendroms funkcijoms.

II.2.5 Į paslaugas orientuoti Grid tinklai

Grid tinklų naudojimą palengvina įvairi tarpinė įranga: programinės įrangos komponentai ir protokolai, kurie suteikia reikalingą kontroliuojamą prieigą prie išteklių. Ligi šiol, Grid tinklai buvo sudaryti naudojant daugiausia specialius viešus komponentus ar tam priklausančias technologijas. Tuo metu, kai įvairiems viešiams ir komerciniams sprendimams sekėsi jų dalykinėse srityse, jie apribojo potencialą, kaip pagrindą ateinančios kartos Grid tinklams, kurie turės būti labai lankstūs dydžio atžvilgiu ir glaudžiai sąveikaujantys, kad patenkintų globalinių įmonių norus. Pastaraisiais metais tapo aišku, kad yra didelis susiejimas tarp Grid skaičiavimo tinklų tikslų ir naudos, kurią suteikia SOA pagrįstos pasaulinio tinklo paslaugos. Greitas kilimas pasaulinio tinklo paslaugų technologijų ir standartų, pateikia evoliucinį kelią nuo paprastos architektūros į standartizuotą, į paslaugą orientuotą, ateities įmonių klasifikuotus Grid tinklus. [7]

Kaip į paslaugą orientuoti Grid tinklai galėtų būti konstruojami parodo (6 pav.) iliustracija. Čia yra paprastas Grid tinklas, kur paslaugos yra panaudotos virtualizuotiems ištekliams ir aprūpinti kitas Grid funkcijas. [5]

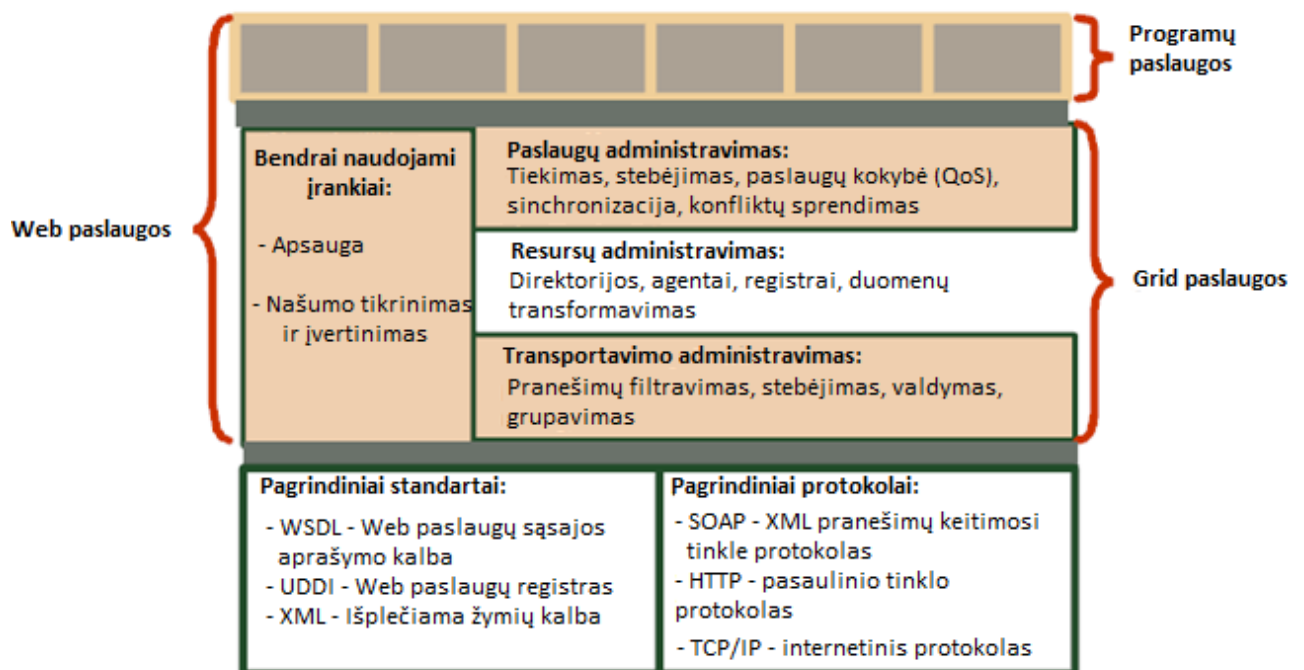


6 pav. (Pasaulinio tinklo paslaugų modelis)

Paveikslėlyje (6 pav.) valdymo konsolė naudojama nusiųsti darbas ir valdyti Grid resursus. Konsolės valdymo programinė įranga susisiekiama su registru paslauga, kad surastų Grid resursus, o paskui susisiekiama su resursų valdymo paslauga, kad periodiškai gautų informaciją apie jų našumo ir

būsenos duomenis, taip žinoma, kurie resursai yra perkrauti ir neprieinami. Vartotojas perduoda darbus darbų pateikimo paslaugai, kuri suranda Grid planavimo paslaugą ir perduoda jai prašymus. Planavimo paslauga suranda ir susisiečia su paslauga, kuri galėtų įvygdyti mūsų darbą ir paprašo informacijos apie resursų būseną, kurie bus reikalingi darbui. Taip planavimo paslauga klausinėja registru paslaugos, kad surastų visus tinkamus resursus ir susisiečia su jais individualiai per jų jungtines paslaugas, kad nustatytų tinkamumą. Jei pakankami išteklių yra pasiekiami, planavimo paslauga renka geriausią pasiekiamą komplektą ir perduoda jų detales programų paslaugai su prašymu pradėti darbus, kitu atveju, ji atideda darbo vygdymą iki tol, kol atsilaisvins reikalingi resursai. Kai tik darbas yra užbaigiamas, programų paslauga praneša rezultatą planavimo paslaugai, kuri informuoja darbų pateikimo paslaugą. Tada darbų pateikimo paslauga informuoja vartotoją. [5]

II.3 Darbo srities modelis



III. Projektinė dalis

III.1 Įrankių ir priemonių pasirinkimo analizė

III.1.1 Agentų bendravimo kalba (ACL)

Agentų bendravimo kalba (angl. Agent Communication Language) iškilo tada, kai buvo sukurta KQML (angl. Knowledge Query and Manipulation Language – Žinių užklauskos ir valdymo kalba), o vėliau ją pakeitė FIPA (angl. Foundation for Intelligent Physical Agents – Protingų fizinių agentų atsiradimo pagrindas). ACL įkūnija abstrakčiąją bendravimo architektūrą, kuri gali būti atvaizduojama konkrečiuose komunikaciniuose protokoluose, taip sukuriant atvirą aplinką. ACL ir virtualios sąsajos leidžia sukurti bendrą API (angl. Application Programming Interface), kuris palengvina kompleksiskai paskirstytų taikomųjų programų tobulinimą. ACL sudaro trys pagrindiniai komponentai: [13]

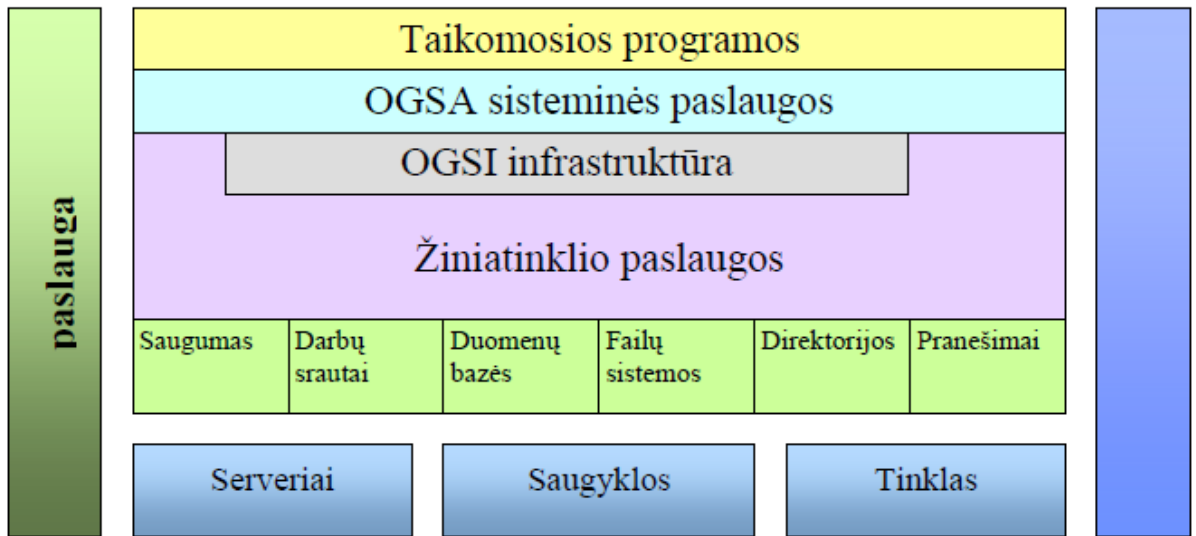
- **Pranešimų klasifikacija.** Agento bendravimo procesai yra susisteminti ir klasifikuojami pagal iš anksto apibrėžtą pranešimų klasifikavimo grupę, dažnai vadinamą – performatyvais. Įvairius performatyvų skaičius skiriasi skirtingose agentų bendravimo kalbose. Pats paprasčiausias turi mažiau nei penkis performatyvus, o sudėtingesni, tokie kaip KQML ir FIPA, turi daugiau nei dvidešimt. Performatyvai parenkami tokie, kurie leistų agentams sąveikauti tokiais sudėtingais būdais, kokie jie būtų, jeigu būtų naudojama sudėtingesnė agentų bendravimo kalba. Ypatingai FIPA ir KQML turi specializuotus performatyvus tokioms užduotims kaip pranešimų persiuntimas (peradresavimas) arba kvietimas teikti pasiūlymus, kuriuos mes atitinkamai traktuojame kaip komunikacinio lygmens funkcijas, arba kaip terminus, kuriuos reikia apibrėžti taikomųjų programų ontologijoje. Kita priešingybė yra tai, kad Agentas-0 galėtų atlikti sudėtingesnius pranešimus, jis priklausytų nuo pagrindinių procesų, kuriuos FIPA ir KQML traktuoja kaip primityvius. Tikslus performatyvų taikymas teikia papildomą naudą, nes išvengiama tolimesnio interpretavimo lygmens, kurio reikalautų neegzistuojantis arba neapibrėžtas pranešimų klasifikavimas. Be to, jie leidžia atlikti subendrintą API programavimą.
- **Pranešimo turinio schema.** Pranešimai, kuriais apsieičia agentai, yra naudojami keistis informacija apie savo aplinką arba kai kurias problemines sritis, taigi pranešimų turinys turi atspindėti jų būklę. ACL reikalauja, kad agentai priimtų sutartą ir bendrai suprantamą kurios nors srities programavimo kalbą, kurią jie galėtų naudoti bendraujant. Tokia kalba paprastai

vadinama ontologija, o šiame kontekste gali būti suprantamas kaip pranešimo turinio schema. Komunikaciniame lygmenyje pagrindas yra tik pranešimuose naudojamos kalbos apibrėžimas. Toks kalbos terminams paprastai turėtų būti nurodomas per URI (angl. Uniform Resource Identifier), kurį savo ruožtu viduje galėtų naudoti agentai, mąstydami apie kalbos terminus. Praktiškai kalbos terminus galėtų apibrėžti CORBA IDL arba XML schema. Dinamiška ir besivystanti Grid aplinkos prigimtis reikalauja daugybinių ontologijų paramos, kurias turėtų sukurti, prižiūrėti ir publikuoti nepriklausomi projektuotojai. ACL vienas iš pagrindinių reikalavimų yra jas aprėpti.

- **Meta informacija.** Agentai turi gebėti atskirti pranešimus pagal jų vidinę būseną nuo pranešimo konteksto. Šis komunikacijos kontekstas apima informaciją apie patį komunikacijos procesą: siuntėją, gavėją, siuntimo laiką, pranešimo identifikatorių ir komunikacijos giją. Agentas gali tuo pasinaudoti, atmesdamas pranešimą arba atskirdamas vieną nuo kito siuntėjus. Sudėtinguose sąveikų protokoluose agentai turi gebėti nustatyti, kokiam pokalbiui priklauso žinutė, o pokalbis gali būti laikomas P2P sesijos n agentams apibendrinimui. Tokia informacija paprastai nėra prieinama objektinėse sistemose, tačiau agentų sistemose ji privalo būti prieinama.

III.1.2 Grid paslaugų architektūra

OGSA yra į paslaugas orientuota architektūra, pritaikyta Grid tinklui (žr. 7 pav.). OGSA palaiko virtualios organizacijos resursų ir paslaugų sukūrimą, valdymą ir jų dinaminį komponavimą. Pagrindinis OGSA komponentas yra OGSF. Tai Grid programinės įrangos infrastruktūros standartas pagrįstas pasaulinio tinklo paslaugų standartais. Visi resursai, tiek fiziniai, tiek loginiai OGSA modelyje yra laikomi paslaugomis. Šios paslaugos yra projektuojamos naudojant pasaulinio tinklo paslaugų architektūrą. Praktiškai Grid paslauga – tai pasaulinio tinklo paslauga. Pasaulinio tinklo paslaugos viena iš technologijų, kuri leidžia kurti laisvai susietas paslaugas ir klientus. Todėl tai tampa labiausiai tinkama technologija naujos kartos Grid paslaugų įgyvendinimui. Tokiu būdu Grid paslaugos gali naudotis pranešimų perdavimo (angl. messaging), paslaugų sąsajos apibrėžimo (angl. description) ir suradimo (angl. discovery) modeliais. Pagrindinis tikslas yra pernešimas pranešimų lygyje, tai pasiekama naudojant XML formatą pranešimų keitimuisi. [14]



7 pav. (OGSA architektūra)

III.1.3 Duomenų tipai

Kadangi pasaulinio tinklo paslaugos ir agentai aprašyti JAVA programavimo kalba, apsirašo skirtingai, tai duomenų tipai irgi bus skirtingi. GT4 remiasi Apache Axis [2] realizacija, todėl nuo to priklauso kokius duomenų tipus galima persiųsti tinklu. Standartiniai atitikmenys tarp WSDL ir JAVA yra tokie [14]:

WSDL	JAVA
xsd:base64Binary	byte[]
xsd:boolean	boolean
xsd:byte	byte
xsd:dateTime	java.util.Calendar
xsd:decimal	java.math.BigDecimal
xsd:double	double
xsd:float	float
xsd:hexBinary	byte[]
xsd:int	int
xsd:integer	java.math.BigInteger
xsd:long	long
xsd:QName	javax.xml.namespace.QName
xsd:short	short
xsd:string	java.lang.String

III.1.4 XML programavimo kalba

XML (angl. *Extensible Markup Language*) buvo sukurta 1996 metais. Ją sukūrė XML darbo grupė, padedama W3C (angl. World Wide Web Consortium) konsorciūmo. XML aprašo duomenų objektų klasę, vadinamą XML dokumentu ir iš dalies aprašo šiuos dokumentus apdorojančių kompiuterinių programų elgesį. Savo struktūra XML dokumentai atitinka SGML dokumentus. XML dokumentai yra sudaryti iš elementų, vadinamų esybėmis, kuriose yra išnagrinėti (angl. *parsed*) arba neišnagrinėti duomenys. Išnagrinėti duomenys yra sudaryti iš simbolių. Vieni simbolių junginiai sudaro duomenis, kiti junginiai sudaro žymes. Žymėmis yra koduojama dokumento sandara ir loginė struktūra. XML pateikia sandaros ir loginės struktūros apribojimų pritaikymo mechanizmą. [15]

XML yra naudojama kaip žymių kalbų sudarymo šablonas [15]:

- XML naudoja neribojamą žymių skaičių, t.y. skirtingai nei HTML nėra jokių apribojimų jų kiekiui – vartotojas pats gali susikurti žymes atsižvelgdamas į informacijos pobūdį.
- Kiekviena XML kalba yra skirta savo taikomajai sričiai, tačiau kalbos pasižymi bendromis savybėmis
- Dokumentams apdoroti yra skirti bendri įrankiai.

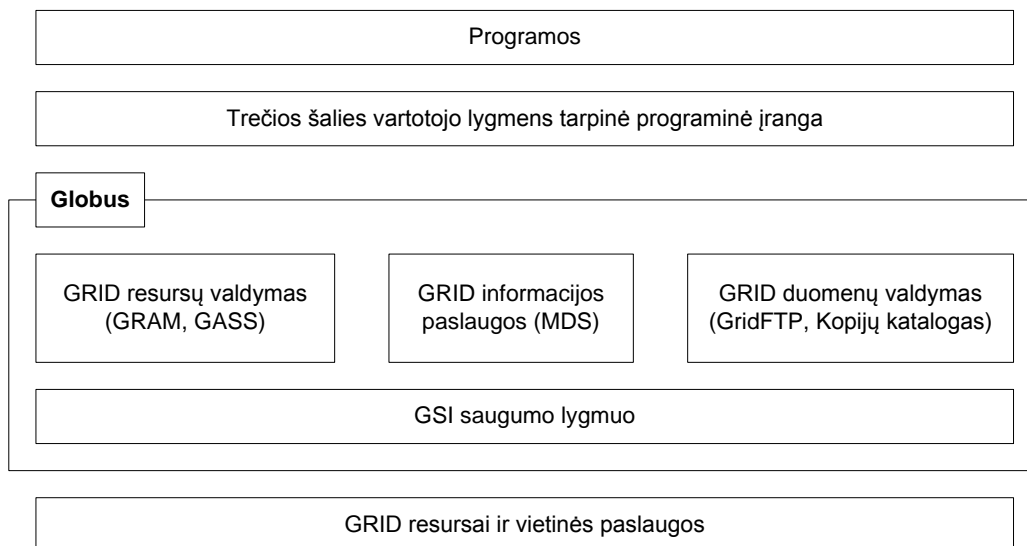
Galima būtų išskirti šiuos pagrindinius XML kalbos paskirties bruožus [15]:

- Atskirti sintaksę nuo semantikos. Tai yra reikalinga todėl, kad yra kuriamas bendras informacijos struktūrizavimo mechanizmas (naršyklės vaizduojamos semantikos yra apibrėžiamos stilių lentelėmis).
- Leisti kurti žymes bet kokiai įsivaizduojamai taikomųjų programų sričiai.
- Palaikyti internacionalizaciją ir nepriklausomybę nuo platformų.
- Ateityje vaizduoti struktūrizuotą informaciją, kuri apimtų ir duomenų bases.

III.1.5 Globus Toolkit

Globus projektas pateikia atviro kodo programinių priemonių komplektą, kuris gali būti naudojamas kuriant Grid skaičiavimo terpes ir Grid pagrindu veikiančias programas. Jis leidžia saugiai dalintis skaičiavimų pajėgumais, duomenų bazėmis ir kitais resursais korporacijos, įstaigos ir geografinėse ribose neaukojant vietinio savarankiškumo. Globus priemonių komplekto

pagrindinės paslaugos, sąsajos ir protokolai leidžia vartotojams vientisai prieiti prie nuotolinių resursų, kai tuo pat metu išlaikoma vietinė kontrolė nusakanti, kas ir kada gali naudotis resursais. [16]



8 pav. (Globus architektūra)

Globus architektūra, pavaizduota 8 paveiksle, turi tris pagrindines paslaugų grupes, kurios prieinamos per saugumo lygmenį. Šios grupės yra:

- resursų valdymas
- duomenų valdymas
- informacijos paslaugos

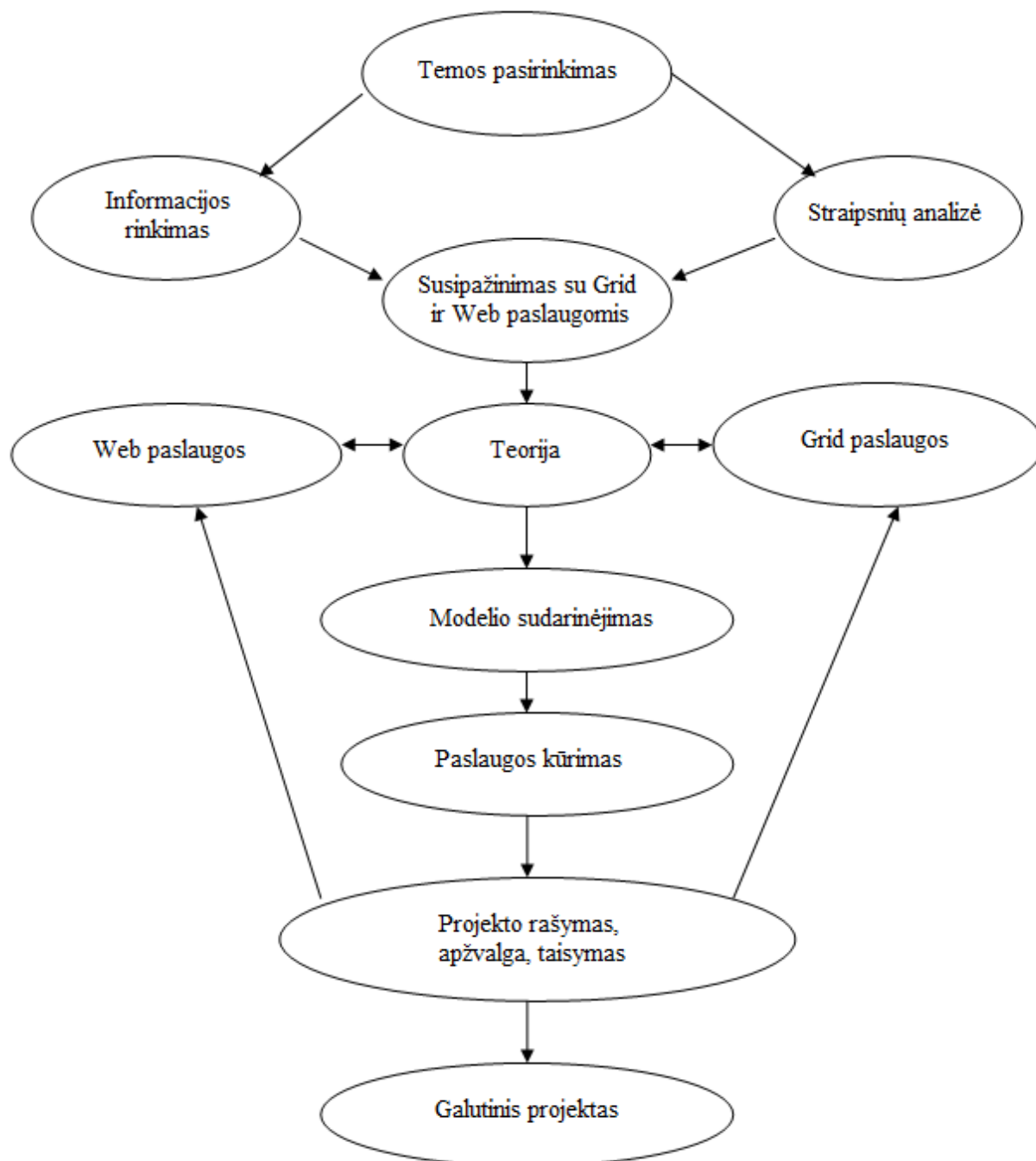
Vietinių paslaugų lygmuo apima operacinės sistemos paslaugas, tinklo paslaugas, tokias kaip TCP/IP, klasterio planavimo paslaugas, kurias teikia apkrovos paskirstytojas (angl. Load Leveller), uždavinių pateikimą, eilių užklausas ir kt. Aukštesni Globus modelio lygmenys suteikia sudėtinių arba heterogeninių klasterių integracijos galimybę. Pagrindinių paslaugų lygmuo turi Globus priemonių komplekto kūrimo blokus, skirtus saugumui, uždavinių pateikimui, duomenų ir resursų informacijos valdymui. Aukšto lygio paslaugų ir priemonių lygmuo turi įrankius, kurie integruoja žemesnio lygio paslaugas arba realizuoja trūkstamą funkcionalumą.

Globus Toolkit teikia įvairius komponentus ir galimybes:

- Realizuotų paslaugų rinkinys infrastruktūros valdymui.
- Įrankiai naujų paslaugų kūrimui Java, C ir Python kalbose.
- Galinga realizuojanti standartus saugumo infrastruktūra
- Programavimo sąsaja - API (skirtingom kalbom) ir komandinės eilutės įrankiai, priėjimui prie skirtingų paslaugų ir galimybių.

IV. Darbo eigos aprašymas

IV.1 Darbų eigos grafas



IV.2 Problemos ir jų sprendimas

IV.2.1 Transporto lygmuo

Transporto lygmuo (angl. Transport Layer) yra pirmasis aspektas mano darbe, kurį man teko nagrinėti, norint sukurti Grid pasaulinio tinklo paslaugų modelį. Kaip žinia, norint kad kas nors veiktų mums reikia turėti tam tikrą terpę, kurioje paslaugos sėkmingai komunikuotų. Transporto lygmens pagrindas yra XML pranešimų perdavimas (XML protokolas, kuris labiau žinomas kaip SOAP). Grid agentai naudoja JAVA programavimo kalbą, tai norint kad jie komunikuotų, mums reikia agento žinutę parašytą JAVA kalba transformuoti į XML kalbą.

JAVA	XML (SOAP)
<pre>public class Marriage { Person husband; Person wife; }</pre>	<pre><Marriage xmlns='uriForMarriage'> <husband> <name>Don Box</name> <age>37</age> </husband> <wife> <name>Barbara Box</name> <age>27</age> </wife> </Marriage></pre>

Kaip matome JAVA kalboje yra klasės, o XML tekstinė informacija. Suprogramuoti paprastą programėlę, kad JAVA kodą konvertuotų į XML formatą būtų įmanoma, tačiau perprogramuoti visą JAVA kodą būtų labai sudėtinga, o pritaikymas tokių programų pasaulinio tinklo paslaugoms būtų nenaudingas, nes atsiranda papildomas konvertavimo komponentas, kurį reiktų įdiegti kiekvienai paslaugai. Kad viso šito nereiktų, naudosime GT4 (angl. Globus Toolkit 4). [3]

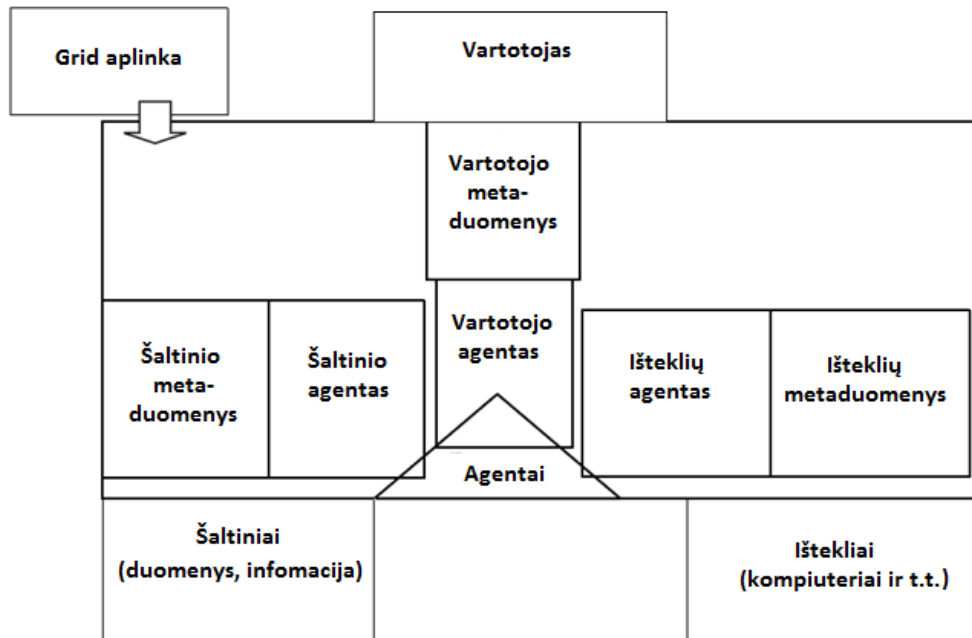
IV.2.2 Paslaugų pateikimo lygmuo

Paslaugų pateikimo lygmuo (angl. Service Discovery Layer) taip pat svarbus šiame projekte. Pagrindinis jo aspektas yra UDDI (angl. Universal Description, Discovery, Integration) projektas. Šis projektas suteikia standartizuotą metodą skelbti ir informuoti apie esančias pasaulinio tinklo paslaugas. Tai lyg „geltonųjų puslapių“ knyga, kurią sudaro 3 dalys:

- Balti puslapiai (pagrindinė susisiekimo informacija)
- Geltonieji puslapiai (pasaulinio tinklo paslaugų klasifikavimas)

- Žali puslapiai (palaikomų funkcijų aprašymas)

Kadangi UDDI yra standartizuotas, tai Grid agentai turi perimti pasaulinio tinklo paslaugų standartinius aprašus. Kad būtų lengviau įsivaizduoti pateiktu agento, kaip komponento Grid tinkle struktūrą: [4]



9 pav. (Agento struktūra Grid tinkle)

Agento sąsaja atrodo taip:

```

Agento sąsaja {
    Boolean užklausa(kintamasis x, kintamasis y);
    Void informuoti();
    Void deinformuoti();
    Void registruoti();
    Void išregistruoti();
    Void gauti();
}

```

Kadangi UDDI apsiraso WSDL kalba, tai agentą reikia transformuoti į ją:

Eilutę „Boolean užklausa(kintamasis x, kintamasis y);“ transformuojame taip:

```

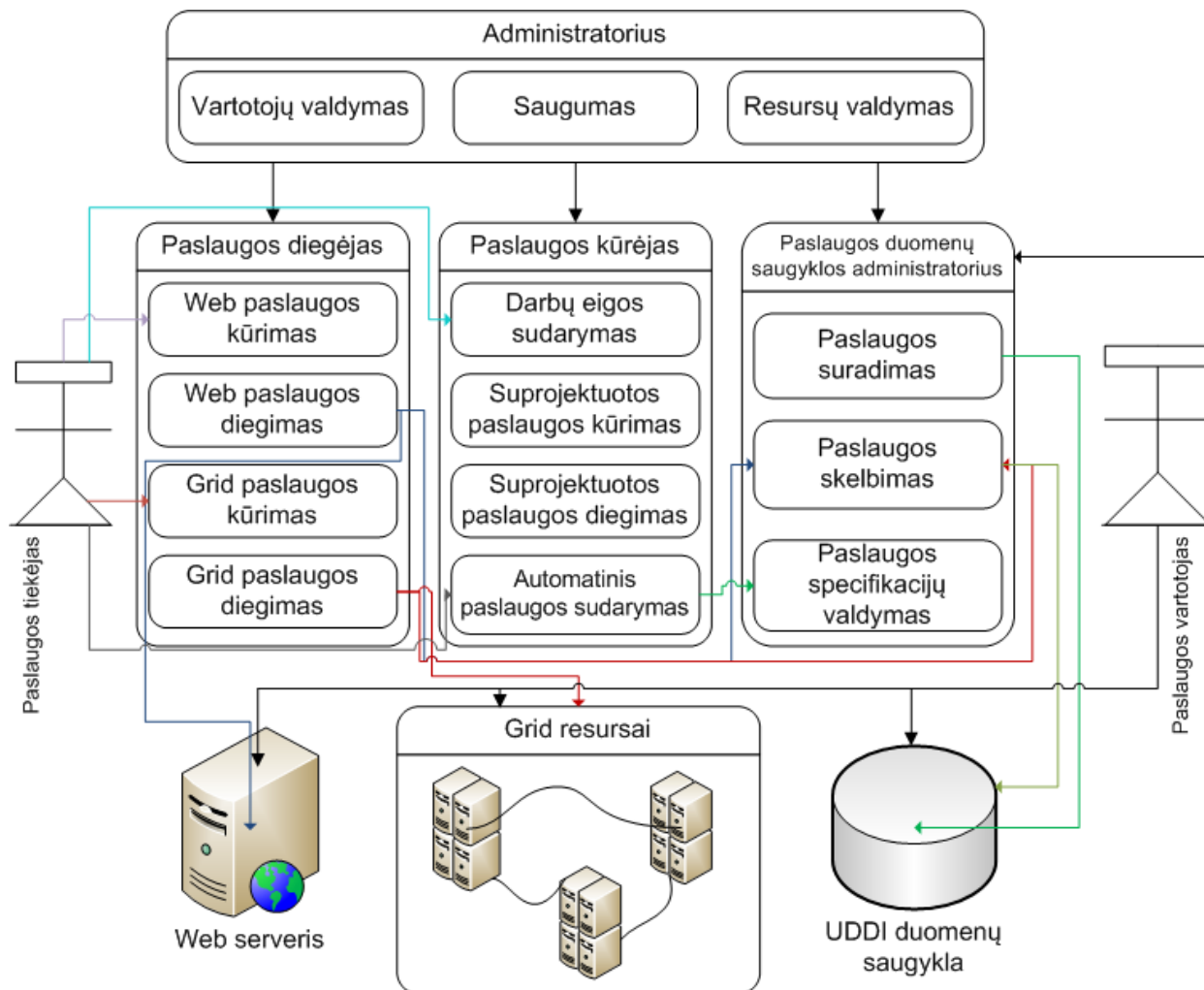
<message name="užklausa">
  <part name="kintamasis x" type="kintamasis x"/>
  <part name="kintamasis y" type="kintamasis y"/>
</message>

```

Atlikę tokią transformaciją galime pabandyti sukurti Grid tinklo paslaugą.

IV.3 Galutinio projekto stovio aprašymas

Išanalizavus pasaulinio tinklo paslaugas ir Grid agentus susidariau bendrą vaizdą, kaip pasaulinio tinklo paslaugas integruoti į Grid tinklą. Pateikiu šio modelio bendrą struktūrą:



10 pav. (Modelio struktūra)

Šį modelį sudaro trys pagrindinės dalys: paslaugos diegėjas, paslaugos kūrėjas ir paslaugos duomenų saugyklos administratorius.

Paslaugos diegėjas. Tiekėjai pateikia savo tinklo paslaugas diegimui.

- **Web paslaugos kūrimas ir diegimas.** Paslaugos diegėjas automatizuoja ir supaprastina diegimo procesą, kad paprasti vartotojai galėtų sukurti ir publikuoti pasaulinio tinklo paslaugas. Šioje dalyje turėtų būti pateikiamas pasaulinio tinklo paslaugų kodas JAVA, JAR, WAR kalbomis, taip pat automatiškai sukuriamas WSDL aprašas. Po sėkmingo

sukūrimo, aprašyta paslauga būtų talpinama APACHE-AXIS serveryje, taip pat paskelbta UDDI duomenų saugykloje.

- **Grid paslaugos kūrimas ir diegimas.** Kad įdiegtų Grid tinklo paslaugą, paslaugos tiekėjai turi paruošti paslaugos kodus ir tinkamai sukonfigūruoti juos diegimui. Paslaugos diegėjas supaprastina tinklo paslaugų diegimo procesą sukurdamas įrankius, kurie reikalingi tinklo paslaugai, o vėliau jas publikuoja per paslaugos skelbimo komponentą, taip pasiekiant UDDI katalogą. Tiekėjas pateikia JAVA kodo realizaciją, tada vyksta kodo patikrinimas ir jeigu viskas gerai, sukuriamas Grid tinklo paketinis archyvas GAR. Jame taip pat turėtų būti pateikti WSDL, JNDI, WSDD.

Paslaugos kūrėjas. Paslaugos kūrėjas padeda vartotojams susikurti paslaugas, naudojant pasaulinio tinklo, Grid tinklo ir kitas sukauptas paslaugas, kurios jau yra publikuojamos paslaugos duomenų saugyklos administratoriaus. Paslaugoms kurti naudojama WS-BPEL (angl. Web Services Business Process Execution Language) [11] kalba ir ActiveBPEL (angl. ActiveBPEL Community Edition Engine) [10], kad sukurtų WS-BPEL procesus. Darbų eigos sudaryme naudojamas diagramų įrankis, kad vartotojai galėtų modeliuoti darbų eigą naudojant WS-BPEL pagrindines ir suformuotas funkcijas.

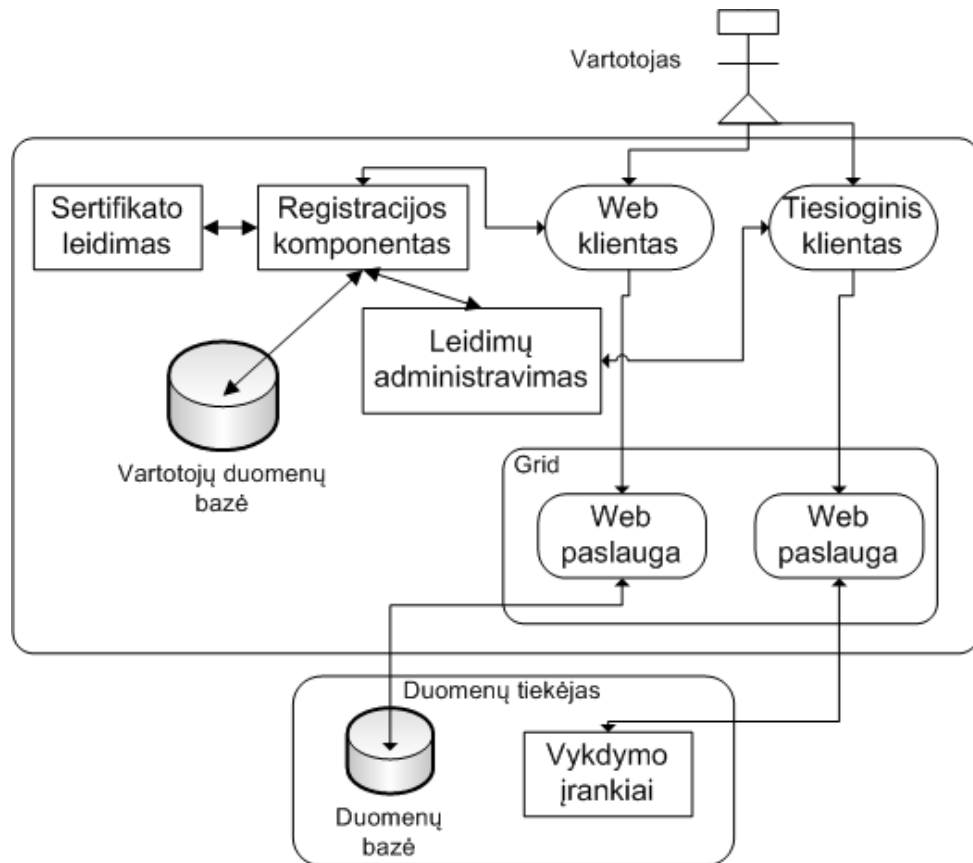
- **Darbų eigos sudarymas ir paslaugos kūrimas.** Leidžia vartotojams sukurti darbų eigą susidedančią iš tokių pagrindinių reikalavimų kaip iškvietimas, gavimas, atsakymas. Darbų eigos sudarymas automatizuoja ir supaprastina kūrimo procesą, kuris padeda paprastiems vartotojams, nežinantiems darbų eigos kūrimo sintaksės WS-BPEL. Čia paslaugos kūrimas turėtų būti suprantamas, kaip naujos paslaugos kūrimas senosios pagrindu arba šabloninis kūrimas.
- **Suprojektuotos paslaugos diegimas.** Šiame komponente darbų eigos kūrimo sintaksė WS-BPEL turėtų būti verčiama į UDDI suprantamus OWL-S (angl. Semantic Markup for Web Services) procesus. [17]
- **Automatinis paslaugos sudarymas.** Kadangi UDDI saugykloje prie kiekvienos paslaugos įvesties ir išvesties informacijos galima lengvai prieiti, Grid tinklo vartotojai gali apibrėžti įvesties ir išvesties reikalavimus. Taip paslauga būtų automatiškai kuriama pagal reikalavimus. Šiame komponente turėtų būti įgyvendinamas kodų konvertavimo mechanizmas, nes paslaugos pateikiamos JAVA kalba, o turime kodą interpretuoti WSDL kalba (žr. skyrių „IV.2 Problemos ir jų sprendimas“).

Paslaugos duomenų saugyklos administratorius. Šis sistemos komponentas palaiko funkcionalumą, kuris yra reikalingas publikuoti, atrasti Grid tinklus ir sukurtas paslaugas, naudojant semantiškai padidintą UDDI saugyklą. Apache jUDDI [12] ir PostgreSQL duomenų saugojimui pilnai įgyvendina UDDI duomenų saugyklą.

- **Paslaugos suradimas.** Pasaulinio tinklo paslaugų vartotojai privalo sugebėti surasti paslaugas, kurios įvykdytų jų reikalavimus. Tai mechanizmas leidžiantis paslaugos duomenų saugyklos administratoriui naudojant UDDI standartinius paieškos kriterijus, atrasti URL adresus mums reikalingų paslaugų.
- **Paslaugos skelbimas.** Paslaugos, kurios yra įdiegtos paslaugos diegėjo, yra paskelbtos UDDI saugykloje. Paslaugos duomenų saugyklos administratorius gauna informaciją iš paslaugos tiekėjo per paslaugos diegėją. Ši informacija yra transformuota į OWL-S paslaugos apibrėžimą. OWL-S apibrėžimas tokiu atveju yra išverstas į atitinkamą UDDI formą ir užregistruotas UDDI duomenų saugykloje.
- **Paslaugos specifikacijų valdymas.** Gebėjimas atrasti tinkamas paslaugas yra žymiai padidinamas naudojant paslaugos specifikacijas. Standartiniai pasaulinio tinklo paslaugų apibūdinimai, kaip OWL-S ir WSMO, yra rekomenduojami specifikacijų valdyme, kurie apibrėžia paslaugos parametrus ir funkcionalumo tipus.

Kaip žinome, Grid tinklas ir pasaulinis tinklas struktūriškai skiriasi, galima sakyti pasaulinis tinklas apima ir Grid tinklus. Dabar norint prisijungti prie Grid tinklo reikia naudoti sertifikatais pagrįstus tarpinius serverius, tačiau jei mes pasaulinio tinklo paslaugas perkeliame į Grid tinklą, teoriškai prisijungimo turėtų nebelikti. Bet Grid tinklo komunikavimas su vartotoju tampa neįmanomas, vadinasi turime palikti tarpinį serverį ir įtraukti tiesioginį prisijungimą (žr. 11 pav.). UDDI registrų paslauga turėtų būti ne Grid tinkle, nes dabartiniai pasaulinio tinklo paslaugų tiekėjai negalės prieiti prie jos. Yra du variantai šiai problemai išspręsti:

1. Sukuriame UDDI registrų bazę Grid tinkle ir ją sinchronizuojame per tam tikras sąsajas su pasaulinio tinklo UDDI registrų baze.
2. Naudojame pasaulinio tinklo paslaugų UDDI registrų bazę ir sukuriame Grid tinklo paslaugoms atitinkamas sąsajas, kad būtų įmanoma prieiti prie šios informacijos.



11 pav. (Prisijungimo schema)

Grid tinklo paslauga.

Paslaugos kūrimą galima suskaidyti į keturias pagrindines dalis:

1. Reikia sukurti paslaugos sąsają.
2. Reikia sugeneruoti pagrindinį ir palaikymo pirminį kodą.
3. Reikia realizuoti paslaugą.
4. Reikia įdiegti paslaugą.

Paslaugos sąsaja parašyta JAVA kalba, GT4 pagalba paverčiama į WSDL kalbą, arba galima iškart rašyti WSDL kalba. Po to vyksta pirminių kodų kūrimas su GT4 XML įrankiais ir susiejimas su SOAP protokolu. Realizacija vyksta per GT4 „GridServiceImpl“ klasę. Kad paslaugą būtų sėkmingai įmanoma įdiegti, jei reikia parašyti jos deskriptorių, kuris aprašys ir konfigūruos paslaugą. Galiausiai, visas parašytas rinkmenas reikia suarchyvuoti į GAR archyvą.

Pilną paslaugos demonstracinį kodą galite pamatyti „Priedas A“. Šios paslaugos kodas parodo kaip realizuojama paprasčiausia paslauga. Paslaugos funkcija – nustatyti ar skaičius yra pirminis.

IV.4 Darbo rezultatai

Agentais paremtos sistemos yra nauja technologija, kuri suteikia tiesioginį pasaulinio tinklo paslaugų pakaitalą Grid tinkle, tačiau nėra labai populiaros ir mažai kur naudojamos dėl integracijos ir realizacijos problemų (žr. skyrių „IV.2 Problemos ir jų sprendimas“). Buvo sukurtas modelis, kuris realizuoja pasaulinio tinklo paslaugas Grid tinkle, tačiau dėl tų pačių integracijos ir realizacijos problemų, „paslaugos kūrėjo“ komponente turėtų būti įgyvendinta kodų transformavimo modelis, šis modelis turėtų atitikti GT4 specifikacijas. Taip pat, iškyla UDDI saugyklos suderinamumo problema, reikėtų projekto, kaip ją realizuoti Grid tinkle arba kaip padaryti, kad tiek iš Grid, tiek iš pasaulinio tinklo būtų įmanoma prieiti prie jos. Agentų sistemos reikalauja nemažų žinių apie jų struktūras, pagrindinė problema neteisingas sąsajų naudojimas ir metodų nežinojimas, tačiau pasižymi itin dideliu universalumu, tiek programavimo kalbų pasirinkime, tiek procedūrų realizavimu. Plačiai naudojamas GT4 yra šiuo metu nepakeičiamas įrankis Grid tinkle paslaugoms kurti. Grid tinklo pasaulinio tinklo paslauga buvo realizuota per GT4 projektą. Paslauga buvo sukurta su JAVA programavimo kalba ir jos pagrindinė funkcija – nustatyti ar skaičius yra pirminis.

IV.5 Patarimai, pastebėjimai, rekomendacijos

Prieš pradėdami dirbti su Grid paslaugomis, išsiaiškinkite „Apache Axis“ ir “Apache Tomcat” technologijas, tai Jums žymiai palengvins bendrą pasaulinio tinklo paslaugų realizacijos sampratą. Ypatingą dėmesį reikėtų atkreipti į transporto lygmenį ir kaip komunikuoja paslauga su paslauga. Tik gerai įgudęs programuotojas sugebės tinkamai dirbti su Grid paslaugomis, todėl darbas su jomis reikalauja didelių pastangų ir reikia pagalvoti ar verta pradėti kurti eksperimentines paslaugas.

V. Išvados

1. Išanalizuotas pasaulinio tinklo paslaugų veikimas ir Grid tinklas.
2. Siekiant ištirti paslaugų specifikacijas, sukurti šie projektai:
 - a. Eksperimentinis paslaugų modelis.
 - b. Demonstracinė Grid tinklo paslauga (JAVA-WSDL)
3. Tolesnis domėjimasis šiuo projektu gali privesti prie to, kad galėsite rinkai pasiūlyti naujas ir reikalingas paslaugas Grid tinkle.

VI. Literatūra

- [1] XCAT 2.0: A Component-Based Programming Model for Grid Web Services. Prieiga per internetą: <http://www.extreme.indiana.edu/xcat/publications/tr-xcat.pdf>
- [2] Developing Web Services in a Computational Grid Environment. Prieiga per internetą: <http://www.escience.cam.ac.uk/papers/ICSC2004.pdf>
- [3] Globus Toolkit Version 4 Grid Security Infrastructure. Prieiga per internetą: <http://www.globus.org/toolkit/docs/4.0/security/GT4-GSI-Overview.pdf>
- [4] Systems Development in a GRIDs Environment. Prieiga per internetą: <http://epubs.cclrc.ac.uk/bitstream/2121/Systems%20Development%20in%20a%20GRIDs%20Environment.pdf>
- [5] An Overview of Service-oriented Architecture, Web Services and Grid Computing. Prieiga per internetą: <http://h71028.www7.hp.com/ERC/downloads/SOA-Grid-HP-WhitePaper.pdf>
- [6] Web Service Grids: An Evolutionary Approach. Prieiga per internetą: http://www.omii.ac.uk/dissemination/paper_web_service_grids.pdf
- [7] Web and Grid Services. Prieiga per internetą: <http://www.neresc.ac.uk/events/presentations-talks/WebGridServices.pdf>
- [8] Web Services Architektūra. Prieiga per internetą: <http://www.webservicesarchitect.com/content/articles/webservicesarchitectures.pdf>
- [9] Grid Market Directory: A Web Services based Grid Service Publication Directory. Prieiga per internetą: <http://www.gridbus.org/papers/gmd-report.pdf>
- [10] ActiveBPEL Community Edition Engine. Prieiga per internetą: <http://www.activevos.com/community-open-source.php>
- [11] Web Services Business Process Execution Language (WSBPEL). Prieiga per internetą: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
- [12] Java implementation of the Universal Description, Discovery, and Integration (UDDI). Prieiga per internetą: <http://ws.apache.org/juddi/>
- [13] Semantics for an Agent Communication Language. Prieiga per internetą: http://reference.kfupm.edu.sa/content/k/q/kqml_as_an_agent_communication_language_7677.pdf
- [14] Interaktyvių e – mokymo sistemų pritaikymas paslaugoms orientuotai Grid architektūrai. Prieiga per internetą: http://vddb.library.lt/fedora/get/LT-eLABa0001:E.02~2007~D_20070816_142543-58777/DS.005.0.02.ETD
- [15] Tinklinių duomenų bazių sandara tinklo paslaugų suradimui. Prieiga per internetą: http://vddb.library.lt/fedora/get/LT-eLABa-0001:E.02~2004~D_20040924_105246-65674/DS.005.0.02.ETD
- [16] Globus Toolkit. Prieiga per internetą: http://kopustas.elen.ktu.lt/studentai/media/grid_martusevicius_v_v0.8.doc?id=doktorantai&cache=cache
- [17] Semantic Markup for Web Services. Prieiga per internetą: <http://www.w3.org/Submission/OWL-S/>

VII. Anotacija

Šis darbas aprašo pagrindinius Grid paslaugų kūrimo metodus, kurie bando pasiūlyti operacinius suderinamumus, stabilumą ir produktyvumą, nepaisant to, kad pasaulinio tinklo paslaugos yra jauna technologija su keliais standartais. Tikimės, kad Grid paslaugos sukurtos iš pasaulinio tinklo paslaugų parodys vykdymo problemas, bent jau pradžioje, ir kad eksperimentavimas ir alternatyvų kūrimas, pareikalaus efektyvesnių pasaulinio tinklo paslaugų realizacijų. Pasiūlytas modelis sukurtas Grid architektūros pagrindu, taip išvengiant rizikos perkurti visą infrastruktūrą.

VII.1 Annotation

This work has described an approach to building Grid Services that attempts to promote interoperability, stability and productivity, despite the fact that Web Services are a young technology with few standards. We anticipate that Grid services built from Web Services will show performance problems, at least initially, and that experimentation and development of alternate, more efficient implementations of some of these Web Services will be necessary. Proposed model is based on existing Grid architecture, without the risk of rebuilding the whole infrastructure.

VIII. Priedas A

JAVA sąsaja (Pirminis.java)

```
package org.globus.ogsa.guide.impl;
public interface pirminis {
    public boolean pirminis(int skaicius);
}
```

JAVA sąsajos konvertavimas į WSDL (build.xml)

```
<target name="generateWSDL" description="Provide the service interface" unless="wsdl.present"
depends="setenv">
<!-- This command will generate a WSDL file and populate it with a binding supporting the
required GridService PortType.-->
    <ant antfile="${build.services}" target="generateWSDL">
        <property name="interface.package" value="org.globus.ogsa.guide.impl"/>
        <property name="interface.name" value="Pirminis"/>
        <property name="generated.dir" value="${package.dir}"/>
    </ant>
</target>
```

Generuojamas pirminis ir palaikymo kodas (build.xml)

```
<target name="stubs" unless="stubs.present" description="Generate service stubs"
depends="generateWSDL">
<ant antfile="${build.services}" target="generateStubs">
<property name="schema.file.dir" value="${package.dir}/Pirminis"/>
<property name="schema.file" value="PirminisService.wsdl"/>
</ant>
</target>
```

Paslaugos realizacija (PirminisImpl.java)

```
package org.globus.ogsa.guide.impl;
import java.rmi.RemoteException;
import org.globus.ogsa.guide.PirminisPortType;
import org.globus.ogsa.impl.core.service.ServiceSkeleton;
```

```

public class PirminisImpl extends ServiceSkeleton implements PirminisPortType {
    public PirminisImpl() {
        super("Pirminis");
    }
    public boolean pirminis(int skaicius) throws RemoteException {
        boolean PirminisSk = true;
        int x = 2;
        while ((PirminisSk) && (x <= skaicius/2)) {
            if (skaicius % x == 0) {
                PirminisSk = false;
            }
            else {
                x = x + 1;
            }
        }
        return (PirminisSk);
    }
}

```

Paslaugos deskriptorius (config.wsdd)

```

<?xml version="1.0" encoding="UTF-8"?>
<deployment name="defaultServerConfig" xmlns="http://xml.apache.org/axis/wsdd/"
xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
<service name="guide/pirminis/PirminisFactoryService" provider="Handler" style="wrapped">
<parameter name="className" value="org.globus.ogsa.impl.core.factory.DynamicFactoryImpl"/>
<parameter name="persistent" value="true"/>
<parameter name="schemaPath" value="schema/core/factory/factory_service.wsdl"/>
<parameter name="instanceSchemaPath" value="schema/guide/Pirminis/PirminisService.wsdl"/>
<parameter name="handlerClass" value="org.globus.ogsa.handlers.RPCURIProvider"/>
<parameter name="instanceClass" value="org.globus.ogsa.guide.impl.PirminisImpl"/>
</service>
</deployment>

```

Visi papildomi pirminiai kodai (build.xml) pateikti kompaktiniame diske.