UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Boštjan Lasnik

STREŽNIŠKA ARHITEKTURA ZA ZAGOTAVLJANJE UPORABNIŠKE IZKUŠNJE V MASOVNIH VEČIGRALSKIH SPLETNIH IGRAH

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE

RAČUNALNIŠTVO IN INFORMATIKA

Mentor: viš. pred. dr. Alenka Kavčič

Ljubljana, 2014

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00388 / 2013
Datum: 4.4.2013

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

Kandidat: **BOŠTJAN LASNIK**

Naslov: **STREŽNIŠKA ARHITEKTURA ZA ZAGOTAVLJANJE UPORABNIŠKE IZKUŠNJE V MASOVNIH VEČIGRALSKIH SPLETNIH IGRAH**
SERVER ARCHITECTURE FOR SECURING USER EXPERIENCE IN MASSIVELY MULTIPLAYER ONLINE GAMES

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Masovne večigralske spletne igre postajajo vedno bolj popularne in privabljajo veliko število igralcev, ki se srečujejo v navideznih svetovih. Različne igre ponujajo različne elemente igralnosti, vse pa stremijo k dobri uporabniški izkušnji, za katero je zelo pomembna tudi arhitekturna zasnova same igre. V okviru diplomske naloge predstavite pristope k izdelavi masovnih večigralskih spletnih iger, predvsem z vidika njihove arhitekturne zasnove. Proučite tudi, kako izbrana arhitektura vpliva na uporabniško izkušnjo v smislu zagotavljanja razpoložljivosti, zanesljivosti in razširljivosti navideznih svetov. Na podlagi ugotovitev razvijte koncept strežniške arhitekture za masovne večigralske igre, ki temelji na arhitekturi s porazdeljenimi strežniškimi storitvami. Razvit koncept arhitekture primerjajte z uporabljenimi arhitekturami v priljubljenih obstoječih igrah.

Mentor:
Alenka Kavčič
viš. pred. dr. Alenka Kavčič

Dekan:
prof. dr. Nikolaj Zimic

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani **Boštjan Lasnik**, z vpisno številko **63090252**, sem avtor diplomskega dela z naslovom:

Strežniška arhitektura za zagotavljanje uporabniške izkušnje v masovnih večigralskih spletnih igrah

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom **viš. pred. dr. Alenke Kavčič**,
- 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 _____ Podpis avtorja: _____

ZAHVALA

Zahvaljujem se mentorici viš. pred. dr. Alenki Kavčič za usmerjanje in pomoč pri izdelavi diplomskega dela.

Posebno zahvalo namenjam staršem in prijateljem za spodbujanje in podporo v času študija.

Kazalo

Kazalo slik

Slovar pojmov

MMO (Massively Multiplayer Online Game): Masovne večigralske spletne igre. Zvrst večigralskih iger.

MMORPG (Massively Multiplayer Online Role Playing Game): Masovne večigralske spletne igre z igranjem domišljijskih vlog. Specifična podzvrst masovnih večigralskih iger, pri kateri igralec z lastnim likom vrši naloge, ki omogočajo njegovo napredovanje.

Crowding: Gnetenje. Fenomen v masovnih večigralskih igrah do katerega pride, ko povečanje števila igralcev in interakcij znotraj nekega območja preseže zmogljivosti strojne opreme, zadolžene za to območje navideznega sveta.

Real-time: Aspekt realnega časa pri zagotavljanju zanesljivosti strežniške arhitekture. Realni čas določa dovoljeno oz. neopazno trajanje zakasnitve interakcij igralcev.

Loose coupling: Pristop k načrtovanju programske opreme, pri katerem so različni programski moduli šibko povezani in funkcionalno neodvisni.

Povzetek

V diplomskem delu je predstavljena problematika zagotavljanja dobre uporabniške izkušnje igralca masovne večigralske spletne igre kot posledice načrtovanja strežniške arhitekture. Predstavljeni so različni pristopi k načrtovanju in implementaciji strežniške arhitekture za zagotavljanje razširljivosti, zanesljivosti in razpoložljivosti masovne večigralske spletne igre. Poudarjen je problem razširljivosti navideznega sveta poljubnemu številu igralcev brez izgube zanesljivosti izvajanja njihovih interakcij z navideznim svetom. V tem kontekstu so prikazane obstoječe rešitve omenjene problematike z uporabo učinkovitega upravljanja z interesnimi polji igralcev. Na tej podlagi je izdelan koncept celovite strežniške arhitekture, ki omenjene težave odpravlja z uporabo funkcionalne delitve igralnih storitev. Podrobneje so opisani različni tipi storitev, kar omogoča prikaz simulacije navideznega sveta na konkretnem primeru. Koncept je nato primerjan z rešitvami komercialnih masovnih večigralskih spletnih iger.

Ključne besede: Masovne večigralske spletne igre, razširljivost, zanesljivost, razpoložljivost, navidezni svet, interesno polje, storitev.

Abstract

The following thesis focuses on problems of ensuring satisfying user experience of massively multiplayer online game players as a result of a server architecture design. Different approaches to ensuring scalability, reliability, and availability of massively multiplayer online games are presented. The main focus is on ensuring scalability of the virtual world with massive amount of players without the loss of reliable processing of player interactions. Existing approaches, based on effective management of players' interests fields are presented. A server architecture concept, that focuses on alleviating the mentioned problems by functionally splitting game services is designed and presented. Different types of services are implemented, which allows for a full example of virtual world simulation to be shown. The concept serves as a basis for comparison with current solutions, deployed by commercial massively multiplayer online games.

Keywords: Massively multiplayer online games, scalability, reliability, availability, virtual world, interest field, service.

Poglavje 1

1 Uvod

Razvijalci masovnih večigralskih spletnih iger (angl. Massively Multiplayer Online Game) so zaradi velike konkurence na trgu in vse večjih zahtev igralcev prisiljeni v neprestano raziskovanje in izpopolnjevanje svojih navideznih svetov, kar se odraža v kompleksnih programskih in strojnih rešitvah. Različne masovne večigralske spletne igre imajo kot glavno prodajno točko izjemno različne elemente igralnosti, vsem pa je značilno, da se v razvojnem ciklu ukvarjajo s problemom zagotavljanja dobre uporabniške izkušnje igralcev. Dobra uporabniška izkušnja ni rezultat samo ustreznih in unikatnih elementov igralnosti, temveč tudi učinkovite strežniške arhitekture in zalednih sistemov, ki te elemente podpirajo. Pri zagotavljanju učinkovite strežniške arhitekture so pri vseh igrah prisotni izzivi za zagotavljanje razpoložljivosti, zanesljivosti in razširljivosti navideznih svetov.

V diplomskem delu je najprej predstavljena osnovna struktura vsake masovne večigralske spletne igre in implikacije omenjenega žanra na njihov razvoj in implementacijo, v poglavju 4 pa so obravnavani trije glavni izzivi pri načrtovanju strežniške arhitekture s podrobnejšo predstavitvijo izziva načrtovanja razširljivosti navideznega sveta. Poudarek je na razpošiljanju zahtev in interakcij igralcev med različne strežniške grozde in delitvijo navideznega sveta. V poglavju 5 je razvit koncept strežniške arhitekture za masovne večigralske spletne igre, ki temelji na arhitekturi s porazdeljenimi strežniškimi storitvami. Ker koncept temelji na znani problematiki načrtovanja strežniške arhitekture lahko služi kot primerna osnova za primerjavo z rešitvami komercialnih masovnih večigralskih spletnih iger, katera je prikazana na koncu 5. poglavja.

Poglavje 2

2 Masovne večigralske spletne igre

V tem poglavju je predstavljena zvrst masovnih večigralskih spletnih iger, njene značilnosti in njen vpliv na izzive v razvojnem ciklu. Prikazani so osnovni elementi masovnih večigralskih spletnih iger in koncepti načrtovanja vsebine igre. V nadaljevanju je opisana kompleksnost razvoja iger te zvrsti in problematika, s katero se soočajo razvijalci.

2.1 Masovne večigralske spletne igre kot zvrst

Masovne večigralske spletne igre so unikaten pristop k načrtovanju iger v večigralskem načinu. Osnovni element zvrsti je prisotnost enega ali večih navideznih svetov, ki so neodvisni od igralcev. To pomeni, da se navidezni svet spreminja in »živi« ne glede na to, ali nek igralec v danem trenutku igra ali ne. Te igre navadno podpirajo na tisoče simultanih igralcev is no neodvisne od geografske lege igralcev in njihove igralne platforme. Igralci lahko v navideznem svetu vršijo interakcije med seboj in med svetom, na čemer temeljijo elementi igralnosti te zvrsti.

Najpopularnejša podzvrst so masovne veičgralske spletne igre z igranjem domišljijskih vlog. Podzvrst določa, da ima vsak igralec nadzor nad lastnim likom v navideznem svetu in z njim vrši vse interakcije. Tako se lahko definira proces napredovanja, kar pomeni, da liki v navideznem svetu pridobivajo nivoje, moč, ali pa boljše zmogljivosti svojih interakcij s svetom in drugimi igralci. Ker vsak igralec v navideznem svetu vidi predstavo navideznega sveta in igralce v svojem vidnem polju, se lahko v igro vključi dodatne elemente igralnosti, ki temeljijo na združevanju igralcev in razvoj kulture, ekonomije, in medigralskih odnosov v navideznem svetu. Tako lahko razvijalci ustvarijo elemente igralnosti, ki za uspešno premagovanje nalog in ovir zahtevajo združevanje igralcev v skupine. Ker sta socialna interakcija med igralci in igranje igre kot akt zabave združena v isti produkt, je zvrst masovnih večigralskih spletnih iger izjemno popularna. Izidi iger kot so Ultima Online, Everquest in Eve Online od poznih 90. let naprej so močno povečali popularnost omenjene zvrsti. Prve igre te zvrsti so podpirale le okoli 50 simultanih igralcev, sodobne pa jih zaradi razvoja tehnologije podpirajo na tisoče. V letu 2012 je bilo število igralcev masovnih večigralskih spletnih iger ocenjeno na več kot 500 milijonov, ti pa letno proizvedejo več kot 13 milijard dolarjev[10], kar redno privablja nove razvijalce masovnih

večigralskih iger. Na sliki 1 je prikazan navidezni svet v masovni večigralski spletni igri Guild Wars 2.

Slika 1: Prikaz osnovnih elementov igralnosti ter interakcij v obliki boja in pogovora v masovni večigralski spletni igri Guild Wars 2.

2.2 Izzivi pri načrtovanju masovnih večigralskih spletnih iger

Komercialni uspeh igre je neposredno odvisen od kvalitete igralnosti neke igre. To se nanaša tako na same elemente igralnosti igre, kot tudi na zaledne sisteme, ki te elemente podpirajo. Ker se masovne večigralske spletne igre zaradi implementacije procesa napredovanja navadno ne obnašajo kot produkt ampak kot storitev, sta kvaliteta storitev in dobra uporabniška izkušnja izjemnega pomena. Zaradi kompleksnosti razvoja in zahtev iger te zvrsti je velikokrat v razvojni cikel vključeno veliko število različnih razvijalcev in načrtovalcev igre. Navadno se razvoj deli na razvijalce, ki so zadolženi za dejansko implementacijo sistemov za podporo večigralske igre in razvoj različnih pogonov, npr. za umetno inteligenco, fiziko, premike v svetu ipd., in na načrtovalce igre, ki omenjene sisteme izrabijo za načrtovanje vsebine igre in principov napredovanja. Zaradi dolgotrajnosti in visoke cene razvoja iger te

zvrsti je nevarnost neuspeha na tržišču zelo velika. Kot primer lahko navedemo masovno večigralsko spletno igro Star Wars: The Old Republic, katere razvojni proces je trajal več kot 5 let, stroške razvoja pa ocenjujejo na okoli 200 milijonov dolarjev[7].

Za zagotavljanje ustrezne uporabniške izkušnje in posledično večje verjetnosti za uspeh igre morajo biti zanimivi in unikatni elementi igralnosti igre podprti z učinkovitimi zalednimi sistemi. Ti sistemi so z izjemo programske opreme na odjemalcu razviti za maksimalno učinkovitost in čim manjšo porabo sistemskih virov. V tem diplomskem delu je poudarek predvsem na problematiki razvoja sistemov za zagotavljanja uporabniške izkušnje preko učinkovitosti strežniške arhitekture. Ta se nanaša na dostopnost igre neodvisno od števila igralcev, njihove geografke lege in platforme. Navidezni svet »živi« na strežniku, programska oprema odjemalca pa služi le kot odjemalec za prikaz sveta in zajem interakcij igralca. Dodaten izziv za zagotavljanje dobre uporabniške izkušnje je zanesljivost izvajanja interakcij igralcev. Mnoge masovne večigralske spletne igre vsebujejo elemente igralnosti, ki podpirajo izjemno hitro in dinamično izmenjavo interakcij med igralci. Ker se interakcije pošiljajo preko interneta je za dobro uporabniško izkušnjo potrebna implementacija sistemov za prikrivanje in kompenzacijo zamika pošiljanja paketov. Eden izmed glavnih problemov na katerega se nanaša diplomsko delo je razširljivost strežniške arhitekture na poljubno število igralcev. Ker so sistemski viri strežniške arhitekture omejeni in obstaja potreba po izvajanju interakcij v realnem času, je potrebna implementacija mehanizmov za učinkovito deljenje virtualnega sveta.

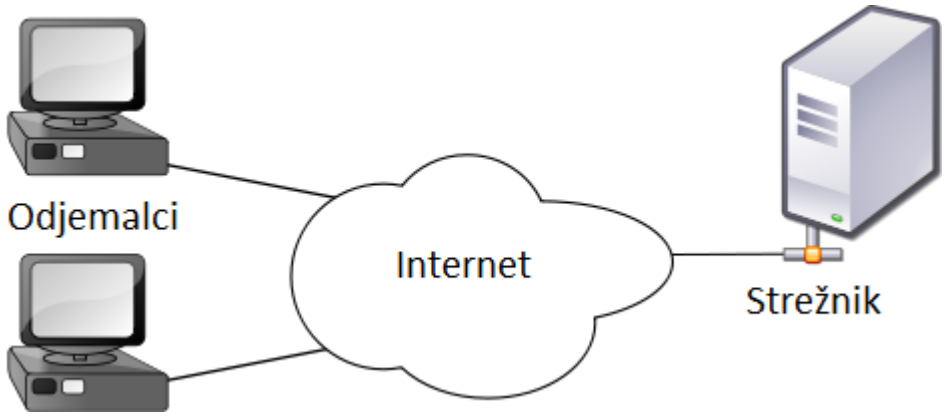
Poglavje 3

3 Struktura masovne večigralske spletne igre

V tem poglavju so na kratko predstavljeni in opisani osnovni deli v arhitekturi masovnih večigralskih spletnih iger. Opisani so različni pristopi k načrtovanju mrežne arhitekture za podporo navideznim svetovom v omenjenih igrah, kar je ena izmed najpomembnejših odločitev v razvojnem ciklu in ima posledice na ves nadaljnji razvoj igre.

3.1 Arhitektura masovne večigralske spletne igre

Večina masovnih večigralskih spletnih iger je v osnovi deljena na odjemalce in strežnike. Posledica tega je uporaba interneta in ustrezno definiranih protokolov za zagotavljanje komunikacije preko interneta kot glavnega komunikacijskega kanala. Poenostavljen prikaz strukture masovne večigralske spletne igre je prikazan na sliki 2.



Slika 2: Prikaz osnovnih gradnikov arhitekture masovnih večigralskih spletnih iger.

Igralci imajo neposreden dostop do igre samo preko programske opreme na njihovih računalnikih. Naloga odjemalca v omenjeni arhitekturi je le izris navideznega sveta, igralčevega lika in zajema interakcij igralca z navideznim svetom in ostalimi igralci. V tem delu se sklicujemo na igre, katerih navidezni svetovi so predstavljeni v treh dimenzijah. Igralci se v navideznem svetu gibljejo prosto (z upoštevanjem pravil in omejitev igre) in izvajajo interakcije s svetom in ostalimi igralci.

Zaradi uporabe interneta kot komunikacijskega kanala ločimo več vrst strežniških arhitektur. Vse možne arhitekture so na kratko predstavljene v naslednjih poglavjih.

3.1.1 Odjemalec-strežnik

Arhitektura odjemalec-strežnik je najosnovnejša arhitektura, kjer sta osnovna dela odjemalec (eden ali več) in en strežnik, kot komunikacijski kanal pa se uporablja internet. Omenjena arhitektura ni primerna za masovne večigralske spletne igre, saj podpira majhno število igralcev, to pa je omejeno z zmogljivostmi strojne opreme strežnika. Število podprtih igralcev je neposredno odvisno od procesorske moči in količine pomnilnika na strežniku. Stanje igre in potrjevanje akcij igralcev poteka na strežniku.

3.1.2 Zrcaljena arhitektura odjemalec-strežnik

Ta arhitektura je strukturo podobna arhitekturi odjemalec-strežnik a se razlikuje v tem, da imamo več strežnikov, kjer vsak izmed njih skrbi za svojo kopijo stanja igre. Odjemalci so porazdeljeni na različne strežnike kar poveča število podprtih igralcev v navideznem svetu. S tem pristopom pride do potrebe po sinhronizaciji stanja igre in igralcev kar je glavni aspekt principa zanesljivosti. Izzivi pri sinhronizaciji stanja igre v izjemno dinamičnih navideznih svetovih z velikim številom interakcij in igralcev so podrobneje opisani v poglavju 4.

3.1.3 Vsak z vsakim

Arhitektura vsak z vsakim (angl. Peer to Peer) je s stališča razširljivosti arhitekture najbolj primerna, vendar se v sodobnih masovnih večigralskih spletnih igrah ne uporablja. V omenjeni arhitekturi je vsak od odjemalcev obenem tudi strežnik, kar pomeni, da izrablja procesorske sposobnosti igralčeve strojne opreme in ima skupno kopijo stanja igre. Neposredne avtoritete v tej arhitekturi ni, kar omogoča goljufanje in potrjevanje neveljavnih interakcij igralcev. Programska oprema na odjemalcu je ranljiva z različnimi napadi (npr. z obratnim inženiringom), kar napadalcu omogoči vpogled v stanje igre in igralcev in njihovo manipulacijo.

3.1.4 Porazdeljene strežniške storitve

Pri tem pristopu k načrtovanju strežniške arhitekture masovnih večigralskih spletnih iger je prisotnih več odjemalcev in več strežnikov. Avtoriteta pri vodenju stanja igre je še vedno strežnik, s katerim igralec komunicira. Ta pristop omogoča fizično ločitev strežnikov in računalniških procesov glede na vrste interakcij in specifične elemente igralnosti. Omenjena arhitektura je poljubno razširljiva, a se ponovno pojavljajo izzivi pri sinhronizaciji stanja igralcev in igre. V poglavju 5 je prikazan konkreten primer koncepta strežniške arhitekture, ki temelji na tem pristopu.

Poglavje 4

4 Zagotavljanje dobre uporabniške izkušnje preko strežniške arhitekture

Uporabniška izkušnja igralcev v masovnih večigralskih spletnih igrah se deli na izkušnjo kot posledico elementov igralnosti ter na funkcionalnost in odzivnost igre. Funkcionalnost in odzivnost sta neposredno povezani z izbiro strežniške arhitekture iz poglavja 3.

Ustrezna uporabniška izkušnja kot posledica funkcionalnosti in odzivnosti igre se deli na tri glavne stebre razvoja strežniške arhitekture v masovnih večigralskih spletnih igrah:

- razpoložljivost,
- zanesljivost,
- razširljivost.

Vsak steber razvoja je podrobneje opisan v naslednjih poglavjih.

4.1 Razpoložljivost

Prvi izziv pri zagotavljanju dobre uporabniške izkušnje je zagotavljanje razpoložljivosti storitev igre. Izraz storitev igre se v tem primeru uporablja kot sama osnovna funkcionalnost igre, torej povezljivost odjemalca s strežnikom. Razpoložljivost se neposredno nanaša na količino časa, v katerem se nek proces, strežnik, ali funkcionalna storitev izvajajo nemoteno. Storitev, ki je vedno na voljo in se izvaja brez omejitev, ima odlično razpoložljivost. Različni procesi in funkcionalne storitve imajo različne prioritete razpoložljivosti in posledično različno vplivajo na uporabniško izkušnjo igralcev. Kot primer lahko podamo primerjavo storitve za prijavo uporabnikov s storitvijo, ki upravlja s pogovori med igralci. V primeru da storitev za prijavo nenapovedano preneha z delovanjem, se igralci praviloma ne morejo prijaviti v igro in posledično ne morejo igrati, medtem ko odpoved storitve za pogovor igralcev ni tako kritična in omogoča nadaljnje igranje v navideznem svetu.

Problem pri razpoložljivosti se pojavi kot delna ali popolna odpoved izvajanja neke funkcionalne storitve. Odpoved dela storitve za 10 minut je slaba, odpoved cele storitve za 1 minuto je slabša. Izguba podatkov o igralcih in stanju igre kot posledica odpovedi strežnika ali storitve je nepopravljiva. Glavno pravilo pri načrtovanju visoko razpoložljivih sistemov je izogibanje implementaciji sistemov, ki temeljijo na eni sami komponenti. To pravilo se lahko

razlaga s pogleda strojne ali programske opreme. Kršenje omenjenega pravila je npr. sistem za shranjevanje podatkov o stanju igralcev in navideznega sveta brez kloniranja podatkov na trdih diskih. Programska oprema z eno komponento je lahko sistem, ki je zadolžen za pisanje in branje podatkov iz pomnilnika.

Glavni pokazatelj razpoložljivosti je statistika. Če je prisoten sistem, ki temelji na eni sami komponenti, in ta komponenta odpove za 1 uro vsakih 9 ur, ima ta sistem 90% razpoložljivost. V primeru da sistem temelji na 2 komponentah, je verjetnost da prva komponenta odpove med odpovedjo druge komponente 10%, kar omogoča 99% razpoložljivost sistema. Sama razpoložljivost individualne komponente je lahko večja. Če ima vsaka komponenta 99% razpoložljivost, in je odpoved celotnega sistema možna samo v primeru odpovedi obeh komponent, je razpoložljivost celotnega sistema 99.99%. Taka razpoložljivost pomeni izpad sistema za eno uro vsakih 13 mesecev, kar je v večini primerov sprejemljivo[11].

4.2 Zanesljivost

Problem pri zagotavljanju zanesljivosti v masovnih večigralskih spletnih igrah neposredno izhaja iz dejstva, da komunikacija med odjemalcem in strežnikom poteka preko internetnega medija. Hitrost prenosa podatkov med strežnikom in odjemalcem je različna za vsakega igralca posebej, česar razvijalci ne morejo zaobiti. Zanesljivost je neposredno odvisna od razpoložljivosti in razširljivosti. Uporabniška izkušnja igralcev s stališča zanesljivosti izvira iz časa izvajanja in pravilnosti vrstnega reda interakcij igralcev.

V ta namen definiramo pojem realnega časa. Realni čas izvajanja neke interakcije pomeni trajanje interakcije od pričetka interakcije (npr. zamaha meča) do pravilnega rezultata interakcije (npr. zadetek drugega igralca). Interakcija se izvede v realnem času, če je ta manjši od igralčevega dojemanja obstoja zakasnitve kot posledice prenosa podatkov med strežnikom in odjemalcem. Ker mora biti vsaka interakcija potrjena na strežniku je minimalna zakasnitev izvajanja interakcije čas od pošiljanja internetnega paketa do prejema potrditve akcije s strani strežnika. Dejanska zakasnitev je še nekoliko večja, saj je potrebno upoštevati procesiranje interakcije na odjemalcu in strežniku, zakasnitev prikaza na ekranu in zakasnitev naprave, ki je zajela interakcijo (npr. tipkovnica). Toleranca zakasnitve paketov je neposredno odvisna od tipa igre. Če je igra dinamična in ima elemente igralnosti, ki omogočajo hitre interakcije med igralci (npr. boj med igralci), je relativno majhen zamik močno opazen. To pomeni da npr. igralci iz Evrope v igri s strežniki na Kitajskem ne morejo izvajati interakcij v realnem času.

Rešitev za kompenzacijo ali za prikrivanje zakasnitve paketov je več, vse pa imajo pozitivne in negativne posledice na funkcionalnost elementov igralnosti:

- večje število podatkovnih centrov,
- odjemalec sam nadzoruje stanje igre,
- kompenzacija zakasnitve na odjemalcu (ekstrapolacija in interpolacija),
- kompenzacija zakasnitve na strežniku (ustrezno načrtovanje elementov igralnosti in previjanje časa).

4.2.1 Več podatkovnih centrov

Najenostavnejša rešitev je postavitev več podatkovnih centrov na različnih geografskih lokacijah po svetu. Ta rešitev je najbolj uspešna, a hkrati najdražja za razvijalce igre. S tem pristopom se uporabnike po uspešni prijavi lahko preusmeri na strežnike, ki so geografsko najbližji njihovi lokaciji (v kolikor razvijalci posedujejo informacije o lokaciji igralcev). Uspešen primer uporabe omenjene metode je masovna večigralska spletna igra World of Warcraft. Igra razpolaga z desetimi podatkovnimi centri po vsem svetu[8].

4.2.2 Odjemalec sam nadzoruje stanje igre

Omenjena rešitev je v teoriji najboljša, saj skoraj popolnoma izniči zakasnitev interakcij. Ta rešitev se najpogosteje uporablja v kombinaciji s strežniško arhitekturo vsak z vsakim. Problem nastane, ker se s tem pristopom izgubi zaupanja vredna avtoriteta, kar odpre možnosti za goljufanje in potrjevanje neustreznih interakcij igralcev.

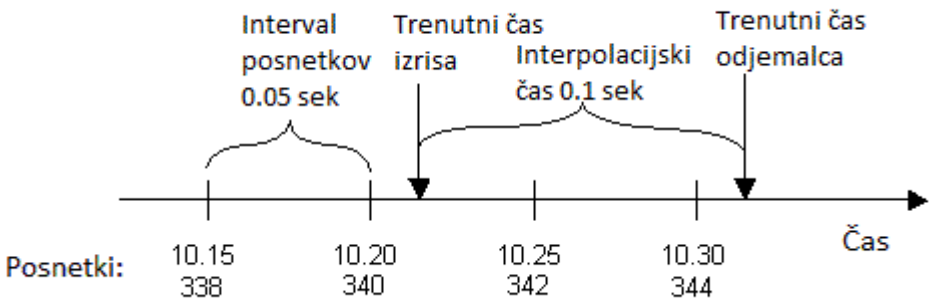
4.2.3 Kompenzacija zakasnitve paketov na odjemalcu

Odjemalec v večini primerov nima nadzora nad upravljanjem stanja igre in igralcev in ima kot nalogo le izris stanja navideznega sveta in zajem interakcij igralca. Pravilno stanje igre in igralcev v navideznem svetu je nadzorovano in upravljano s strani strežnika. V primeru, ko razvijalci ne implementirajo sistemov za kompenzacijo zakasnitve paketov, bi igralci pri premikanju »poskakovali« po navideznem svetu. To je posledica nelinearnosti prejemanja paketov od strežnika kot posledice izgube in zakasnitve paketov na poti do odjemalca. Za

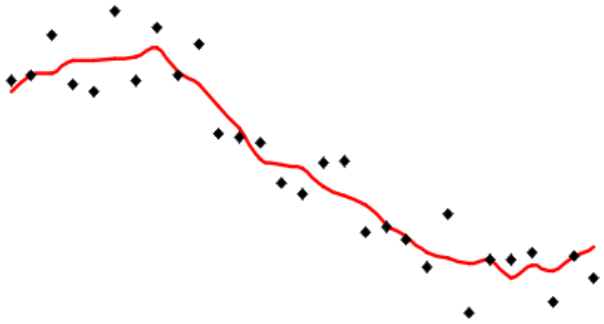
kompenzacijo zakasnitve paketov na odjemalcu se uporablja princip napovedovanja pozicije igralcev. Odjemalec napove naslednjo pozicijo premikajočega se igralca, paket s strežnika pa to napoved samo potrdi ali izniči. Najpogostejša pristopa sta ekstrapolacija in interpolacija.

Ekstrapolacija je poizkus napovedi pozicije igralca v prihodnosti. Ko paket s strežnika prispe do odjemalca, se pozicija igralca popravi na ustrezno vrednost, določeno s strani strežniške avtoritete. Nova pozicija se vedno računa med čakanjem na naslednji paket, upošteva pa se igralčeva hitrost in trenutna pozicija. Za ta pristop je potrebno na odjemalcu popolnoma razviti enak sistem za premikanje kot na strežniku, kar v razvojnem smislu pomeni kloniranje kode. Večji problem pri tem pristopu izvira iz dejstva, da je točna napoved pozicije mogoča samo v primeru, ko je hitrost igralca enakomerna. Pri neenakomerni hitrosti se pozicija ne mora izračunati natančno, kar oteži izračune pri potrjevanju zadetkov igralcev.

Interpolacija je zgladitev poti igralca v premikanju. Stare pozicije igralcev se shranjujejo in njihove nove pozicije niso takoj izrisane v navideznem svetu. Namesto takojšnega izrisa nove pozicije so vse pozicije vseh igralcev zamaknjene na nek konstanten čas. Pot med zadnjima dvema pozicijama se interpolira in ustvari gladek premik. Prikaz interpolacije v odvisnosti od igralnega časa odjemalca je viden na sliki 3. Interpolacijski interval bi v idealnih razmerah moral biti enak zakasnitvi paketov. Pristop deluje v primeru, ko ni izgube paketov in ko je zakasnitev konstantna. Slabost tega pristopa je, da poleg same zakasnitve paketov zaradi prenosa preko interneta pride še do dodatne zakasnitve zaradi interpolacije, kar pomeni, da vsi igralci dejansko vidijo stanje navideznega sveta v preteklosti. Ta dejavnik mora biti nujno odpravljen s kompenzacijo zakasnitve na strežniški strani. Na sliki 4 je prikazana interpolirana pot v odvisnosti od dejanskih pozicij, ki jo izriše programska oprema odjemalca.



Slika 3: Prikaz interpolacije v odvisnosti od igralnega časa.



Slika 4: Prikaz interpolirane poti (rdeča črta) med pozicijami, potrjenimi na strežniku (črne pike).

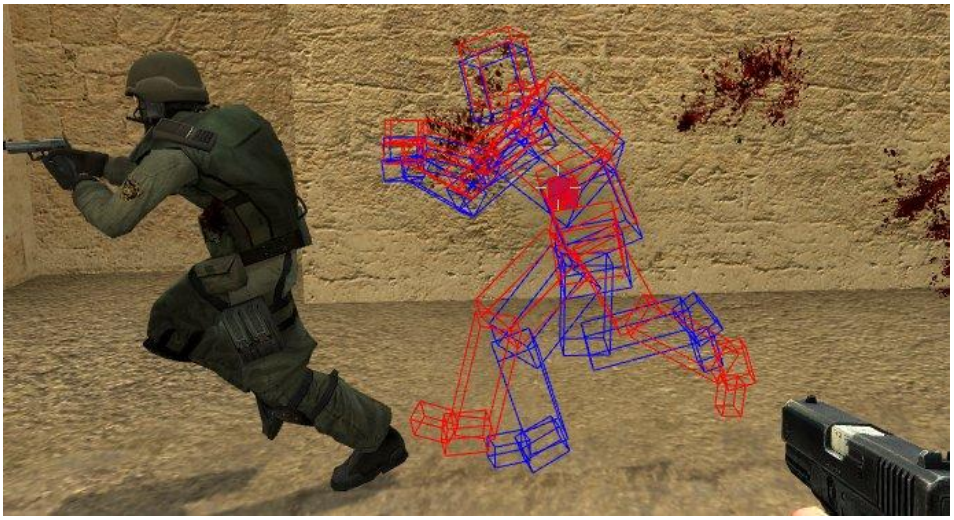
4.2.4 Kompenzacija zakasnitve paketov na strežniku

Strežnik nima potrebe po napovedi stanja igre, saj ima praviloma prioriteto pri vodenju stanja igre. Naloga kompenzacije zakasnitve na strežniku je potrjevanje akcij igralcev. To je potrebno, saj se med časom, ko igralec prične interakcijo in pošlje paket strežniku in ko ga strežnik dejansko prejme, igralni čas premakne naprej in stanje igre ni več enako kot na odjemalcu ob pričetku interakcije. To je posebej problematično pri potrjevanju paketov, ki nosijo informacijo o interakcijah med igralci, npr. pri zamahu meča proti drugemu igralcu. V času, ko je igralec pritisnil tipko za zamah meča, je igralec na svojemu izrisu navideznega sveta nasprotnega igralca videl neposredno pred seboj, a zaradi zakasnitve strežnik ob prejemu paketa nasprotnega igralca ne vidi na isti poziciji kot odjemalec. Možnih rešitev za zagotavljanje zanesljivosti je ponovno več.

Ena od rešitev je, da strežnik ignorira omenjeni problem. Ta rešitev se opazi v igrah, kjer je igralni element igre dejansko sledenje igralcu in napovedovanje pozicije nasprotnih igralcev v obliki logičnega razmisleka igralca samega. Ta pristop je enak kot v resničnem svetu (npr. vojak mora upoštevati moč in smer vetra pri strelih iz daljave). Problem v masovnih večigralskih spletnih igrah nastane pri različno dolgih zakasnitvah paketov nasprotnih igralcev. Na samo napoved igralca o poziciji nasprotnih igralcev zato vpliva tudi dejavnik, na katerega igralec nima vpliva. Za različne igralce je torej potrebno izračunati različne napovedi, kar ni ustrezen element igralnosti igre.

Druga rešitev je previjanje časa, kar pomeni, da strežnik shranjuje določeno število preteklih stanj igre. Ko na strežnik prispe nov paket, se primerjajo časovne značke na paketu in poustvari se preteklo stanje igre. Ta princip zahteva sinhronizirane igralne ure na odjemalcu in strežniku. V najslabšem primeru strežnik ne bo mogel uporabiti te tehnike, ker je zakasnitev paketa večja od dejanske najstarejše shranjene verzije stanja igre. Problem pri

tem pristopu izhaja it tega, da so zakasnitve igralcev lahko različne. Poleg tega lahko pride do anomalij, kot je npr. potrditev izvedbe udarca igralca A nad igralcem B, a ko potrditveni paket prispe do igralca B, le-ta že stoji za zidom oz. nekim drugim zakloniščem. Kljub temu da igralec B vidi svoj lik kot skrit, bo še vedno prejel potrditev uspešnega zadetka s strani igralca A. V tej implementaciji so vse interakcije med igralci neposredno odvisne od časa zakasnitve teh igralcev, tako da igralci z večjo zakasnitvijo močno vplivajo na igralce z manjšo. Strežniške rešitve na tem principu morajo uporabiti omejitev največje dovoljene zakasnitve, zmanjšati količino preteklih stanj za igralca z velikim zamikom ali uporabiti drugo rešitev. Primer previjanja časa je viden na sliki 5. Strežnik primerja interakcijo s preteklo pozicijo igralca, kar omogoči potrditev zadetka[12].



Slika 5: Primer previjanja časa na strežniku kot oblika kompenzacije zakasnitve.

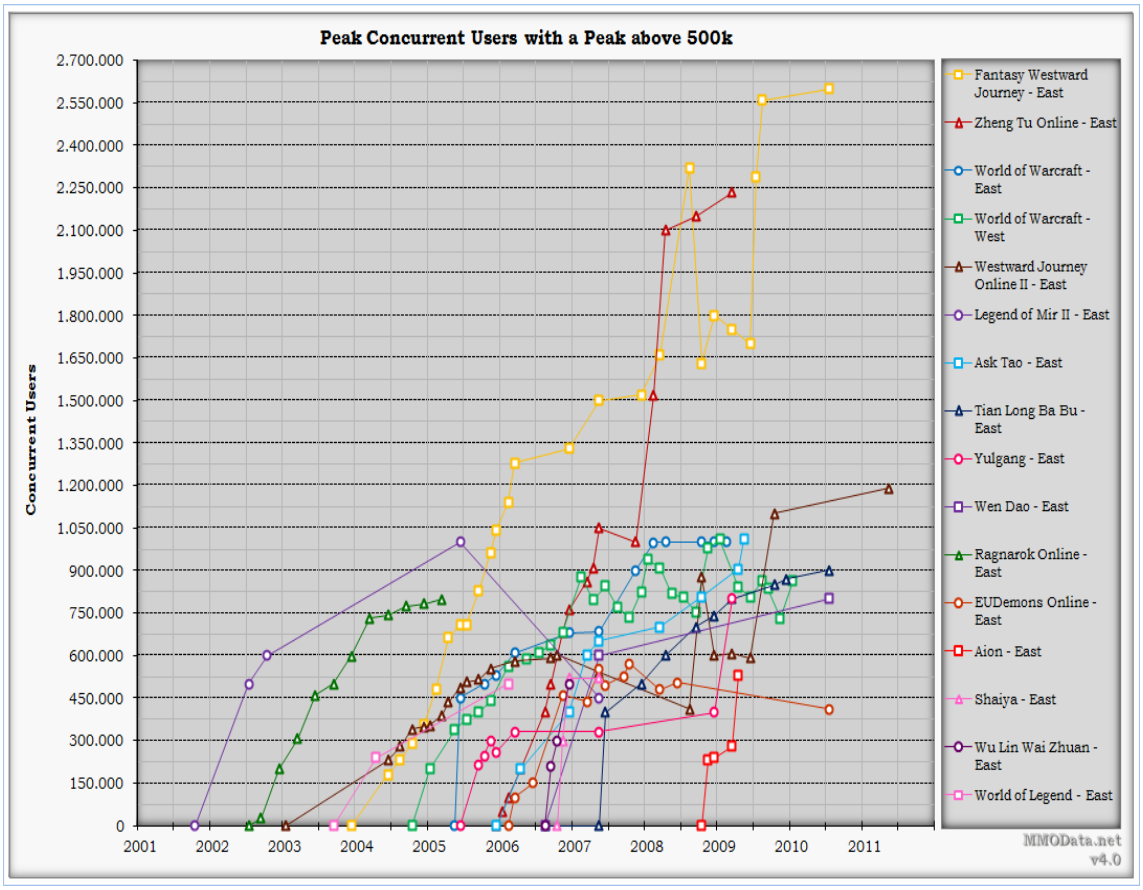
Še ena izmed rešitev, ki se velikokrat uporablja v kombinaciji z drugimi rešitvami je skrivanje zakasnitve skozi elemente igralnosti igre. Primer tega je, da ob vhodu akcije poleg pošiljanja paketa odjemalec nemudoma prične z animacijo in grafičnim ali zvočnim učinkom. Do samega prikaza rezultata interakcije pride šele ob koncu izvajanja animacije.

4.3 Razširljivost

Storitev masovne večigralske igre je razširljiva, ko ne omejuje števila simultanih igralcev v igri. Potrebno je poudariti, da popolnoma poljubno razširljiva storitev ne obstaja zaradi tehničnih omejitev kot so npr. procesorska moč, količina delovnega spomina na strežnikih in

pasovna širina povezave do podatkovnega centra. Razširljivost je marsikdaj enačena z razpoložljivostjo in definira razliko med večigralskimi spletnimi igrami in masovnimi večigralskimi spletnimi igrami. Če nek navidezni svet v igri ne zadošča razvojnemu stebru razširljivosti so novi igralci postavljeni v čakalno vrsto za vstop v ta svet. To ne zadošča pogoju razpoložljivosti, saj igralcem zaradi preobremenjenosti strežnika ni razpoložljivo igralno mesto v nekem navideznem svetu.

Na sliki 6 so prikazana največja števila simultanih igralcev v nekaj masovnih večigralskih spletnih igrah.



Slika 6: Največja števila simultanih igralcev v popularnih masovnih večigralskih spletnih igrah skozi leta[9].

Vrste igralcev za vstop v navidezni svet so lahko izjemno dolge ob izidu težko pričakovane igre. V primeru, da navidezni svet nima eksplicitno določene zgornje meje števila igralcev, se le-ta lahko povzpne nad mejo zmogljivosti strojne opreme. Zaradi prekoračenja procesorske moči in pasovne širine pride do zakasnitve interakcij uporabnikov, ki bi morale biti izvedene v realnem času. Interakcije igralcev so postavljene v čakalno vrsto ali pa se sploh ne izvedejo.

To ni sprejemljiva uporabniška izkušnja s stališča zagotavljanja zanesljivosti. Zakasnitev tipa interakcij je neposredno odvisna od funkcionalnosti strežnika. Če je proces na strežniku funkcionalno odgovoren za več vrst interakcij (npr. premiki, boj, pogovori …), so zakasnjene vse omenjene interakcije, medtem ko funkcionalno ločeni procesi še vedno omogočajo sprejemljivo uporabniško izkušnjo. Kot primer lahko podamo zakasnjen proces za pogovore med igralci. Premikanje in boj v omenjenem navideznem svetu potekata nemoteno, medtem ko so le pogovori med igralci zakasnjeni.

Zmotno je misliti, da preprosto dodajanje dodatne strojne opreme in strežnikov lahko poveča število igralcev na strežniku kot v primeru velikih spletnih storitev. Glavna razlika se nahaja v razpošiljanju sporočil med strežniškimi grozdi. Vsaka spletna zahteva v spletnih storitvah je obravnavana kot ločena celota in je neodvisna od zahtev ostalih uporabnikov. S tem je omogočeno neodvisno razpošiljanje zahtev med strežnike in strežniške grozde. Pri procesih masovnih večigralskih spletnih iger pa je glavni izziv pri razpošiljanju med seboj soodvisnih interakcij. Vsak uporabnik poleg svojih interakcij ob izrisu navideznega sveta vidi tudi interakcije drugih igralcev v njegovem vidnem polju in globalne interakcije celotnega navideznega sveta. Dodajanje strežnikov v strežniški grozd ne zadosti tem potrebam, saj je potrebno stanja igre in igralcev na različnih strežnikih in v različnih procesih sinhronizirati v realnem času brez izgube aspekta zanesljivosti.

4.3.1 Pristopi k razpošiljanju sporočil

Zahtevnost razširljivih storitev v masovnih večigralskih spletnih igrah se neposredno odraža v problemu sinhronizacije stanja igre za vse igralce v sprejemljivih časovnih okvirih. Pri velikem številu simultanih igralcev znotraj posameznega navideznega sveta postane problematična uspešna sinhronizacija vseh igralcev in njihovih interesnih polj.

Interesno polje igralca je območje okoli igralca, v katerem igralec izrazi zanimanje za interakcije okolice in drugih igralcev. To pomeni, da interakcij igralcev in objektov igre, ki so izven igralčevega interesnega polja oz. so preveč oddaljeni, da bi bili pomembni za igralca, ni potrebno pošiljati igralcu. Poleg upravljanja z interesnim poljem je še vedno potrebna delitev navideznega sveta na manjša območja za potrebe zmanjšanja procesorske obremenjenosti. S tem se pridobi podlago za nadaljnje zrcaljenje navideznih svetov in območij. Z zagotavljanjem upravljanja interesnega polja igralca in ustreznim deljenjem navideznega sveta na manjša območja se pridobi podlago za izbiro ustrezno razširljivih arhitektur. Tu se potem razvijalci odločijo za zrcaljen pristop, pristop s porazdeljenimi strežniškimi storitvami ali pristop z uporabo obeh načinov. Za potrebe procesiranja interakcij velikega števila igralcev

se za del rešitve uporablja strežniške grozde, ki vsebujejo veliko število združenih strežnikov, ti pa posledično zagotavljajo večjo procesorsko moč. V idealnem primeru se sporočila razpošilja enakomerno med več fizičnih strežnikov v grozdu.

V storitvah masovnih večigralskih spletnih iger se učinkovito razpošiljanje sporočil omogoči z zagotavljanjem štirih pogojev:

- z ustrezno regionalizacijo navideznega sveta,
- z upravljanjem strežniških grozdov glede na regionalizacijo,
- razpošiljanjem interakcij glede na njihov tip,
- z dinamičnim dodajanjem procesorske moči prezasedenim procesom.

4.3.2 Regionalizacija

Prvi pogoj za razširljive storitve v masovnih večigralskih spletnih igrah je zagotavljanje regionalizacije in delitve navideznega sveta na območja. To je soodvisno od samega načrtovanja navideznega sveta igre in mora biti vzpostavljeno zgodaj v razvojnem ciklu igre. Za zagotavljanje učinkovitega drobljenja navideznega sveta se le-ta lahko razdeli na 2 različna načina:

- geografsko,
- vedenjsko.

Geografska delitev je delitev navideznega sveta na posamezna območja ob inicializaciji navideznega sveta. Meja med območji je zasnovana vnaprej. Ta pristop je večinoma značilen za igre, v katerih obstaja fizična oz. neposredna meja med območji. Primer je igra, v kateri zidovi hiše fizično ločijo sobe. Interesna polja igralcev so v tem primeru vedno vsebovana znotraj posameznih sob, prehodi med stenami pa niso mogoči. Mogoča je tudi delitev navideznega sveta brez neprehodnih mej med območji.

Vedenjska delitev uporablja različne tipe interakcij za zagotavljanje delitve na območja. Primer tega je npr. delitev sveta na zemeljski del in zračni del v igri, kjer so premiki dovoljeni po tleh in po zraku. Ker igralci v zraku zaradi večje hitrosti potrebujejo veliko večjo interesno polje kot na zemlji, je smiselna delitev na območja.

Uporaba regionalizacije za povečanje razširljivosti storitev masovnih večigralskih spletnih iger je strokovno poimenovana upravljanje z interesnimi polji igralcev[6]. Na sliki 7 je prikazana konkretna delitev navideznega sveta na območja v masovni večigralski spletni igri Guild Wars 2.

Slika 7: Delitev navideznega sveta na območja v igri Guild Wars 2.

4.3.3 Združevanje strežnikov v grozde

Vse razširljive rešitve uporabljajo principe strežniških grozdov za povečanje procesorske zmogljivosti. Strežniške grozde se lahko uporabi skupaj z regionalizacijo za učinkovito razpošiljanje sporočil.

Igralci pri tem pristopu izberejo oz. jim je vnaprej izbrana pripadnost nekemu vnaprej določenemu igralnemu svetu. Z uporabo zrcaljene arhitekture odjemalec-strežnik in geografske regionalizacije se pripravi ustrezno razširljivo arhitekturo. S tem omogočimo prisotnost enakega navideznega sveta na več strežniških grozdih. Vsak grozd lahko skrbi za lasten igralni svet, ki je dejansko kopija istega navideznega sveta. Igralni svetovi so lahko poljubno poimenovani in poljubno mnogi. Primer zrcaljenja navideznega sveta je npr. delitev na jezikovne igralne svetove. Z geografsko regionalizacijo lahko za vsako območje

navideznega sveta skrbi ločen strežniški proces, vsak strežniški grozd pa ima nadzor samo nad lastnimi procesi. Vsako območje in vsak navidezni svet torej obstajata v ločenih procesih tolikokrat, kolikor igralnih svetov ima igra. S tem v igri pridobimo konsistentnost na tri različne načine:

- igralci nimajo interakcije med različnimi svetovi,
- igralci nimajo interakcije med različnimi območji,
- igralci znotraj regije imajo lahko neposredne interakcije s katerimkoli drugim igralcem v tem območju.

Z zagotavljanjem geografske regionalizacije in posledično z upravljanem strežniških grozdov se pridobi zadovoljiv približek razširljive strežniške arhitekture, ki je v mnogih primerih še vedno v uporabi v sodobnih masovnih večigralskih spletnih igrah. Ta pristop ima tudi negativne posledice, saj strežniki in strežniški procesi za območja z majhnim številom igralcev porabljajo majhno količino procesorske moči, medtem ko je strežnikom z velikim številom igralcev primanjkuje. Primer delitve na igralne svetove v igri Guild Wars 2 je prikazan na sliki 8.

Slika 8: Prikaz igralnih svetov v Ameriškem podatkovnem centru v igri Guild Wars 2.

4.3.4 Razpošiljanje interakcij glede na njihov tip

Z upoštevanjem upravljanja strežniških grozdov se razporejanje obremenitve lahko doseže na dva načina:

- razporejanje obremenitve na nivoju igralcev. Igralci so razporejeni na različne strežnike ob prijavi v igro,
- razporejanje glede na interakcije – strežniki upravljajo z lastnimi procesorskimi zmogljivostmi glede na vzorce interakcij igralcev.

Razporejanje obremenitve na nivoju igralcev je standardna izbira v masovnih večigralskih spletnih igrah in se sklicuje na standardne rešitve razpošiljanja sporočil. Igralci so razporejeni na različne strežnike v grozdih, za pravilno preusmerjanje in posledično razporejanje pa skrbijo ustrezni usmerjevalni protokoli na usmerjevalnikih oz. namenska programska oprema na posebnih strežniških procesih. V kompleksnih rešitvah je prisoten visok nivo medprocesne komunikacije, saj samo razporejanje na usmerjevalnikih ni dovolj za sinhronizacijo stanja igre.

4.3.5 Dinamično dodajanje procesorske moči

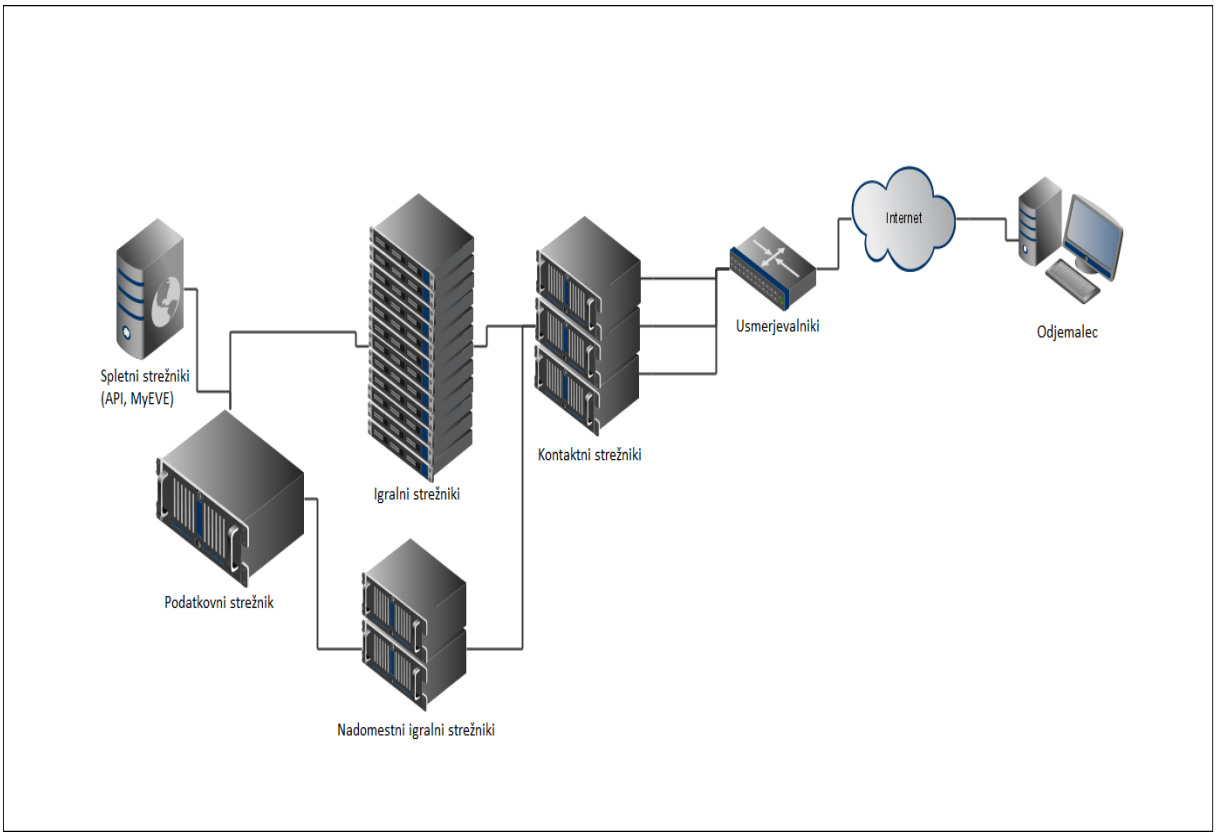
Kljub uspešnemu drobljenju navideznega sveta na manjša območja, ter uporabi strežniških grozdov in igralnih svetov, ostaja problem prezasedenosti samega strežniškega procesa. V masovnih večigralskih spletnih igrah lahko iz različnih razlogov pride do nenadnega povečanja števila igralcev v nekem območju. Pri tem lahko pride do gnetenja (angl. Crowding), kar pomeni presežek zmogljivosti strežnika in pripadajočega procesa za omenjeno območje. Gnetenje je fenomen v masovnih večigralskih spletnih igrah, ko število igralcev v nekem območju preseže meje zmogljivosti strojne opreme, zadolžene za to geografsko regijo. To v končni obliki pomeni nezmožnost izvajanja interakcij igralcev v realnem času.

Ob prezasedenosti procesov se ta problem v prisotnosti ustreznega upravljanja strežniških grozdov rešuje z dinamičnim dodajanjem dodatne procesorske moči. Rešitev je več:

- območja so lahko pomanjšana ali naprej dinamično deljena, novo nastali programski procesi lahko prevzamejo nadzor nad novo ustvarjenimi območji,

- nekatere rešitve ob povečanju števila igralcev nove interakcije v nekem območju preusmerijo na bolj zmogljive strežnike, ki so vnaprej pripravljeni na visoko število igralcev.

Omenjena problematika je prisotna tudi v masovni večigralski spletni igri Eve Online. V tej igri imajo igralci nadzor nad lastno vesoljsko ladjo, s katero potujejo po vesoljih, ki tvorijo navidezni svet. Strežniška arhitektura v Eve Online ni deljena na različne grozde, ampak obstaja samo en velik strežniški grozd za vsa vesolja. To pomeni, da za vsako vesolje obstaja unikaten programski proces, ki skrbi za vse igralce v tem vesolju. Strežniki v tem grozdu zadostujejo izzivu razširljivosti, saj imajo za zagon novih procesov na voljo veliko procesorske zmogljivosti. Individualni procesi in posledično individualna vesolja pa niso poljubno razširljiva. Podrobneje je strežniška arhitektura v igri Eve Online prikazana na sliki 9. Na sliki je prikazana arhitektura s samo enim strežniškim grozdom[3].



Slika 9: Strežniška arhitektura igre Eve Online.

Eve Online rešuje problem gnetenja z uporabočasovne upočasnitve. Časovna upočasnitev ima dve fazi:

- ko pride do velike bitke v nekem vesolju oz. do velikega povečanja števila igralcev znotraj nekega vesolja, se najprej vsi manj pomembni procesi na strežniku, zadolženemu za omenjeno vesolje, ustavijo. To pomeni, da igralci na drugih, manj zasedenih vesoljih izgubijo povezavo in se morajo ponovno prijaviti v igro. Strežniška arhitektura po ponovni prijavi na novo zažene zaustavljene procese na drugih, manj zasedenih strežnikih. To je posledica dejstva, da imajo prioriteto procesi, v katerih pride do nenadnega povečanja števila igralcev. Prisotnost sistema s prioritetami omogoča, da se igralci v tem vesolju bitke lahko udeležijo takoj in brez ponovne prijave v igro. Sistem s prioritetami tako skrbi za uspešno upravljanje s procesi v nenapovedanih bitkah, medtem ko so vnaprej napovedane bitke in veliki dogodki vnaprej preusmerjeni na bolj zmogljive namenske strežnike,
- druga faza nastopi, ko je strežnik na robu porabe svojih procesorskih zmogljivosti. Tu se prične faza upočasnitve igralnega časa, ki ga lahko upočasni tudi na 10 odstotkov običajnega časa. V praksi to pomeni, da se igralci v tej bitki bojujejo v počasnem posnetku, a še vedno lahko izvajajo interakcije v realnem času. Količino upočasnitve časa se dinamično, v odvisnosti od lastnih zmogljivosti, določi na strežniku. Brez časovne upočasnitve bi prišlo do izpada strežnika oz. do prezasedenosti strežnika, kar bi onemogočilo uspešno izvajanje interakcij v realnem času in posledično slabo uporabniško izkušnjo igralcev.

Zaradi uporabe upočasnitve časa je bil 28. 7. 2013 v Eve Online dosežen rekord v številu igralcev v eni bitki. V bitki, ki je trajala približno 6 ur, je bilo prisotnih nekaj več kot 4000 igralcev[2], vse interakcije pa so bile simulirane na enem igralnem strežniku. Izsek iz bitke je prikazan na sliki 10.

Slika 10: Bitka v Eve Online, kjer je sodelovalo več kot 4000 igralcev, celotna bitka pa je potekala znotraj enega procesa na enem igralnem strežniku.

4.3.6 Upravljanje z interesnimi polji

Regionalizacija in upravljanje s strežniškimi grozdi sta neposredno odvisna od načina upravljanja z interesnimi polji igralcev. Eden od načinov upravljanja je napovedno upravljanje interesnih polj. Vsakemu igralcu se doda vnaprej določeno velikost osebnega interesnega polja. Interakcija med igralci je možna samo v primeru prekrivanja interesnih polj. Zaradi preprostosti preverjanja prekrivanja interesnih polj se v odvisnosti od hitrosti igralca doda še napovedno interesno polje. S tem omogočimo več vrst interakcij med igralci:

- pri prekrivanju dveh ali več interesnih polj igralcev nastane potreba po visoki frekvenci izmenjave pozicijskih sporočil vsem igralcem v preseku,
- prekrivanje napovednih interesnih polj brez prekrivanja osebnih interesnih polj določa obstoj možnosti, da bosta igralca v prihodnosti izjavila potrebo po visoki frekvenci izmenjave pozicijskih sporočil,
- ko se nobeno izmed polj ne prikriva je izdana zahteva za nizko frekvenco izmenjave pozicijskih sporočil.

Natančna definicija interesnih polj in njihovo upravljanje je kritičnega pomena za uspešno sinhronizacijo stanj igre med ločeni strežniški procesi. Konkretna uporaba upravljanja z interesnimi polji je podana v naslednjem poglavju.

4.4 Razširljivost z uporabo arhitekture porazdeljenih strežniških storitev

V poglavju 4.3 je bila razlaga razširljivosti strežniške arhitekture večinoma predstavljena z uporabo in nadgradnjo arhitekture zrcaljenih igralnih strežnikov, v tem poglavju pa je opisan pristop z arhitekturo porazdeljenih strežniških storitev. Ta pristop z implementacijo učinkovitih mehanizmov sinhronizacije stanja igre omogoči boljšo razširljivost in predvsem boljše možnosti nastavitve in prilagodljivosti strežniške arhitekture. Z uspešnimi mehanizmi sinhronizacije stanja igre in medprocesne komunikacije se ob presežku igralcev deljena območja navideznega sveta dinamično delijo na več manjših podobmočij. Za vsak manjši del skrbi nov ločen proces ali storitev. S tem se omogoči odlična uporabniška izkušnja, saj je prehod med navideznimi mejami območij za igralce transparenten, medtem ko so igralci v zrcaljeni arhitekturi ob prehodu med območji prisiljeni v novo vzpostavitev povezave na drug strežnik.

4.4.1 Storitev

Za uspešen prikaz delovanja in uporabe omenjene arhitekture je potrebno ponovno definirati pojem procesa. Proces je v arhitekturi porazdeljenih strežniških storitev samostojna funkcionalna enota, ki skrbi za majhen, a specifičen del funkcionalnosti navideznega sveta. Navidezni svet ali območje navideznega sveta tako postane skupek storitev, ki med seboj tvorijo zaokroženo enoto funkcionalnosti navideznega sveta. Razporejanje interakcij je s tem avtomatizirano.

Upravljanje s storitvami in pravilno razpošiljanje sporočil igralcev je predpogoj za uspešno delovanje storitev, zahtevan pa je dobro definiran protokol medprocesne in medstrežniške komunikacije. Storitve je mogoče funkcionalno ločiti na različne načine. Ločene storitve so lahko odvisne od samega tipa interakcij igralcev v navideznem svetu. To pomeni, da se neodvisno od ostalih storitev lahko loči npr. storitve za shranjevanje stanja igre, storitve za pogovore med igralci in storitve za vodenje umetne inteligence. Vsak strežnik je lahko zadolžen za več storitev ali pa eno samo. Tako lahko npr. strežnik, ki skrbi za neko območje, delo razdeli na storitve fizike, umetne inteligence, vodenja premikov in boja.

4.4.2 Dogodek

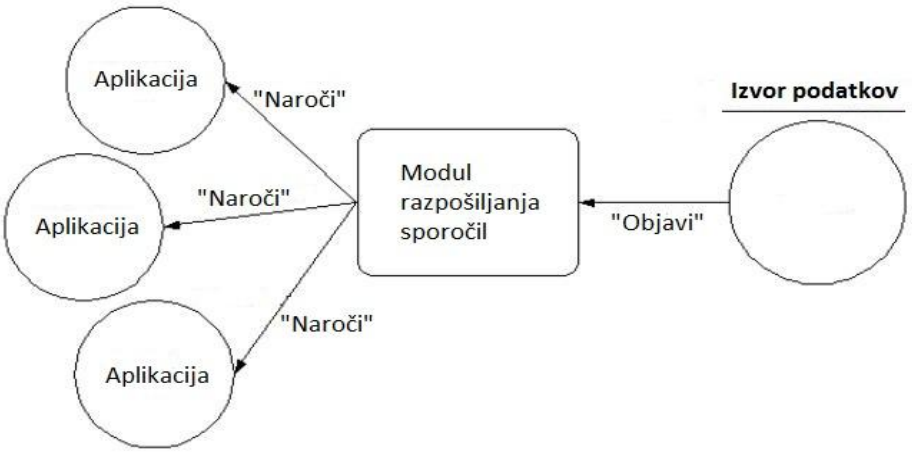
Ker se z uporabo storitev ustvari skupna platforma za izvajanje interakcij je potrebno ponovno definirati sama sporočila odjemalcev. Ker v omenjeni arhitekturi vse storitve delujejo na enakem principu je potrebno poenotiti vse medstrežniške in medprocesne interakcije skupaj z interakcijami uporabnikov. Vsa sporočila z interakcijami so poenotena pod enak vmesnik dogodka, ti pa morajo vsebovati zadostno število potrebnih informacij o stanju igre, ki ga spreminjajo. S tem se loči funkcionalnost storitev in sposobnost izvajanja dogodkov s šibko odvisnostjo od drugih procesov in storitev (angl. loose coupling)[1].

4.4.3 Izzivi pri načrtovanju interakcije med igralci na različnih procesih

S stališča igralca naj prehodi med sosednjimi območji ne bi vplivali na same elemente igralnosti igre. Izzivi za konsistentnost elementov igralnosti so torej[6]:

- igralci naj ne bi prejemali informacij o interakcijah, ki so izven njihovega interesnega polja,
- potrebno je določiti pravila igre in ustrezno sinhronizacijo stanja igre igralcev, ki so na meji med območjema in posledično nimajo popolnoma jasne pripadnosti fizičnemu strežniku,
- upravljanje z interesnimi polji igralcev mora biti nemoteno kljub temu, da interesno polje presega mejo med območjema. To s konkretnega vidika igralca pomeni, da mora imeti dostop do podatkov o interakcijah igralcev na drugih strežnikih, v kolikor se ti nahajajo v njegovem vidnem polju.

Dejanski problem upravljanja z interesnimi polji je torej pripadnost strežniku in posledični dostop do lokalne kopije stanja igre med različnimi strežniki. V strežniški arhitekturi se za rešitev zahtev po podatkih o interakcijah uporabi arhitekturni stil objavi-naroči (angl. publish-subscribe pattern), ki ga prikazuje slika 11.



Slika 11: Prikaz delovanja arhitekturnega stila objavi-naroči.

4.4.4 Medprocesna komunikacija

Glavni izziv pri načrtovanju arhitekture s porazdeljenimi strežniškimi storitvami je zagotavljanje ustrezne medstrežniške komunikacije ter sinhronizacije stanja igre in igralcev na fizično ločenih strežnikih v realnem času. Zaradi poenotenja sporočil in interakcij pod enak vmesnik dogodka je rešitev problema trivialna. Različne storitve na različnih fizično ločenih strežnikih imajo tako poenoten način prejemanja in obdelovanja dogodkov. Tudi pri tej rešitvi se lahko uporabi arhitekturni stil objavi-naroči. Konkreten primer problematične situacije je prikazan na sliki 12. Na sliki so s črnimi pikami prikazani igralci in njihova interesna polja v odvisnosti od navideznih mej med strežniki. Najbolj problematičen primer je v tretjem kvadratku, kjer se igralec in njegovo interesno polje nahajata na meji med ločenima območjema, kar v praksi pomeni na meji med ločenima fizičnima strežnikoma.

Slika 12: Primer igralcev, njihovih pozicij in interesnih polj v odvisnosti od mej med fizičnimi strežniki[6].

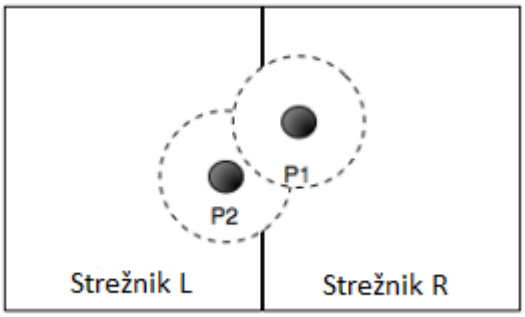
Pri zagotavljanju medprocesne komunikacije je poudarek na glavnem izzivu medprocesne komunikacije v masovnih večigralskih spletnih igrah, kar so interakcije med igralci na storitvah, ki skrbijo za območja virtualnega sveta. Te interakcije imajo prioriteto, saj morajo biti izvedene v realnem času, prenos stanja navideznega sveta in igralcev med strežniki pa mora biti transparenten in omogočati nemoteno funkcionalnost elementov igralnosti.

Ker se na sliki 12 levi igralec in njegovo interesno polje v celoti nahajata znotraj območja strežnika 1 je situacija neproblematična. Prav tako je s stališča medprocesne komunikacije nezanimiva situacija na strežniku 2, saj sta oba igralca še vedno znotraj območja. Bolj zanimiva je situacija, kjer pride do prekrivanja interesnih polj igralcev na navidezni meji med strežnikoma. V tem primeru lahko pride do štirih različnih scenarijev:

- igralec na eni strani meje med strežnikoma mora videti dogodke igralcev, ki so na drugi strani meje, v kolikor so igralci v njegovem interesnem ali vidnem polju,
- igralec v premikanju lahko v kateremkoli primeru naglo spremeni smer in prečka mejo med strežnikoma,
- nek dogodek, ki ima izvor na enem območju lahko konča na drugem. Primer tega je preko meje izstreljen izstrelek,
- rezultat nekega dogodka vpliva na stanje sveta na obeh straneh meje. Primer tega je eksplozija objekta ravno na meji med strežnikoma.

Pri razlagi scenarijev je potrebno predpostaviti, da so mehanizmi za komunikacijo med strežnikoma in med igralci implementirani. To konkretno pomeni implementacijo arhitekturnega stila objavi-naroči med strežnikoma in med strežnikom in igralcem.

Rešitev omenjenih scenarijev se sklicuje na sliko 13. Na sliki je prikazana situacija igralca P1 in P2 in njunih interesnih polj v odvisnosti od meje med strežnikoma L in R.



Slika 13: Primer scenarija igralcev P1 in P2 glede na mejo med strežnikoma L in R[6].

Prvi scenarij določa, da so dogodki igralca P2 na neki strani meje med strežnikoma vidni igralcu P1 na drugi strani meje, v kolikor interesno ali vidno polje igralca P1 vključuje igralca P2. Problem je trivialen, saj imata oba sosednja strežnika po predpostavki odprt komunikacijski kanal. Ko se igralec P2 na strežniku L približa meji med strežnikoma, strežnik L pošlje poseben dogodek strežniku R. Dogodek vsebuje ukaz za naročanje igralca P2 na strežnik R preko arhitekturnega stila objavi-naroči. S tem je stanje obeh igralcev vidno strežniku R in posledično igralcu P1. Ker je prišlo do prekrivanja interesnih ali vidnih polj je postopek istočasno tudi obraten. Postopek je za oba igralca transparenten.

Drugi scenarij rešuje problem pri nagli spremembi smeri igralca v smeri proti meji med strežnikoma. Pri prehodu čez navidezno mejo med območjema se mora stanje igralca prenesti iz enega strežnika na drugega. V primeru igralca P1 to pomeni, da strežnik R pošlje dogodek strežniku L, dogodek pa predstavlja ukaz za spremembo pripadnosti igralca P1 strežniku L. Poleg spremembe pripadnosti se pošlje tudi sam objekt igralca in njegovo stanje. Vsi dogodki igralca P1, ki med prenosom stanja prispejo na strežnik R, se preusmerijo na strežnik L in počakajo, da se prenos v celoti zaključi. Ker prehod med strežniki poteka v obliki enega samega sporočila z objektom, se prenos zaključi hitro in ga igralec ne opazi. Pravila prenosa igralca med strežnikoma so lahko določena na več načinov. Eden on načinov je takojšen prenos v primeru, ko igralec prečka mejo, spet drugi je omejitev tega prenosa glede na frekvenco prehodov oz. igralčevo relativno pozicijo.

Tretji scenarij opisuje rešitev problema, ko nek izstrelek iz prvega območja pristane na drugem območju. Za prikaz se ponovno vzame strežnika L in R in predpostavko, da je izvor izstrelka na strežniku L, cilj pa na strežniku R. V tem primeru mora strežnik R prejeti dva dogodka. Prvi dogodek prejme, ko je izstrelek izstreljen, pošiljanje pa je avtomatsko kot posledica povezanosti obeh strežnikov. Dogodek se pošlje le v primeru, ko je izstrelek izstreljen znotraj interesnega polja strežnika R. Strežnik R s tem izve hitrost izstrelka in lahko izračuna cilj izstrelka. Drugo obvestilo prejme v trenutku, ko izstrelek prestopi navidezno mejo med območjema. S tem lahko še enkrat preračuna cilj izstrelka in upošteva vse vmesne dogodke, ki so se zgodili med prehodom izstrelka. Igralci, razen v primeru, ko pride do vmesnega dogodka, ne opazijo zakasnitve. Vmesni dogodek je lahko npr. zadetek nekega drugega igralca znotraj izračunane poti. Zaradi dveh sporočil lahko strežnik R uspešno ugotovi novo stanje izstrelka. Zakasnitev se poveča za čas izračuna novega dogodka, kar v ustrezni strežniški infrastrukturi in pasovni širini ni opazno.

Četrti scenarij najbolje odraža celotno uporabnost in funkcionalnost tega pristopa k načrtovanju strežniške arhitekture. Sama sinhronizacija stanja je izmed vseh scenarijev najbolj zahtevna in zakasnjena. V tem scenariju je nujna ločitev dogodkov na lokalne in skupne. Lokalni dogodki so dogodki, ki se v celoti izvedejo znotraj meja nekega območja kljub temu, da jih igralci drugih območij, v kolikor so dogodki znotraj njihovega interesnega polja (prvi

scenarij), lahko vidijo. Skupni dogodki so dogodki, ki se hkratno zgodijo na več strežnikih. Kot rešitev se implementira mehanizem, ki med vsemi sosednjimi strežniki določa primarni strežnik. Ta določitev je lahko poljubna, primer je določitev primarnega strežnika z uporabo najmanjše unikatne številke strežnika. Vsak strežnik skrbi za svoje lokalne dogodke, primarni pa skrbi še za skupne dogodke. Ko se skupni dogodek zgodi na primarnem strežniku, ga le-ta izvede in obvesti vse sosednje strežnike. Če pa skupni dogodek izvira na sekundarnem strežniku, ga le-ta najprej pošlje primarnemu in počaka na potrditev.

Tehnika je razširljiva na poljubno število sosednjih strežnikov. Sekundarni strežnik obvesti naslednji strežnik s prvo manjšo unikatno številko, proces pa se nadaljuje dokler se dogodek ne pošlje ustreznemu primarnemu strežniku. Tako za mejo s štirimi strežniki v najslabšem primeru potrebujemo tri ločena pošiljanja dogodka. Z ustreznimi mehanizmi za pošiljanje dogodkov in pasovno širino se v večini primerov ta zakasnitev izvede brez izgube aspekta zanesljivosti.

4.4.5 Toleranca napak in okrevanje po sesutju

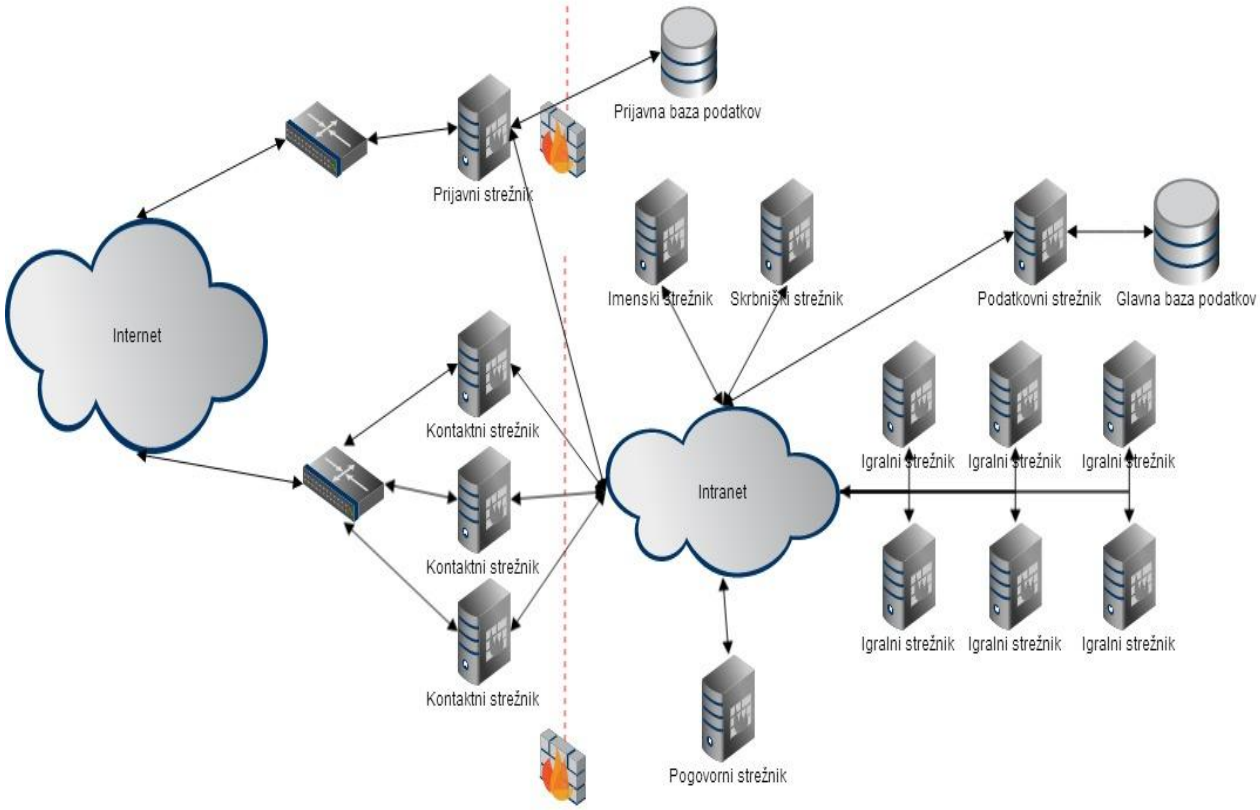
Masovne večigralske spletne igre so zaradi velikega števila igralcev in velike računske zahtevnosti zelo dojemljive na nenapovedana sesutja storitev in procesov. V večini iger se zato uporablja posebne storitve, ki so po funkcionalnosti namenjene shranjevanju podatkov o stanju igre in igralcev. Zaradi ogromne količine podatkov je nujno potrebna delitev na tista stanja igre, ki morajo biti vedno ustrezno skladiščena, in tista brez take potrebe. Primer tega bi bila primerjava koordinat igralca in njegova trenutna količina življenja. Sistem bi brez rednega skladiščenja igralčeve pozicije ob naslednji prijavi igralca postavil v navidezni svet na napačne koordinate, medtem ko je preprosta ponastavitev zdravja igralca na maksimalno dovolj dobra rešitev. Poleg stanja igralcev je potrebno tudi skladiščenje stanja navideznega sveta. V nekaterih igrah je zadovoljiva rešitev ponastavitev sveta na prvotno stanje, v drugih igrah pa je potrebno celotno stanje obnoviti. Storitve skladiščenja podatkov in storitve igralnih območij morajo nujno biti fizično ločene.

V primeru ko je celovita obnova stanja navideznega sveta po sesutju izjemnega pomena, se v arhitekturi porazdeljenih strežniških storitev lahko izrabi uporabljen koncept dogodkov. Ker so vsi dogodki in interakcije poenotene pod enak vmesnik z natančno definiranimi omejitvami, strukturo in ukazi, ima sistem zmožnost enostavnega beleženja dogodkov. Beleženje dogodkov lahko poteka na poljubni lokaciji in od poljubne časovne enote dalje. Igralna storitev po ponovni vzpostavitvi prebere beležko dogodkov in poustvari enako stanje navideznega sveta kot pred sesutjem.

Poglavje 5

5 Koncept arhitekture s porazdeljenimi strežniškimi storitvami

V tem poglavju je predstavljen koncept strežniške rešitve za podporo masovnim večigralskim spletnim igram, ki temelji na arhitekturi s porazdeljenimi strežniškimi storitvami. Koncept je prikazan na sliki 14 in predstavlja funkcionalno delitev strežnikov in pripadajočih storitev. Slika prikazuje funkcionalno delitev strežnikov in pripadajočih storitev in način komunikacije odjemalca preko kontaktnih strežnikov. Ti z uporabo večih mrežnih kartic fizično ločujejo javno omrežje od strežniške arhitekture. Ta delitev določa način komunikacije in smer pretoka podatkov in interakcij med strežniškimi storitvami in odjemalcem. Koncept z delitvijo storitev glede na funkcionalnost ustreza pogoju razširljivosti.



Slika 14: Koncept strežniške arhitekture s porazdeljenimi strežniškimi storitvami za potrebe masovnih večigralskih spletnih iger.

V nadaljevanju poglavja so podrobneje predstavljene minimalne storitve, ki so potrebne za samo upravljanje strežniške arhitekture in določajo pravila za medprocesno komunikacijo. Te storitve neposredno ne podpirajo elementov igralnosti igre, a omogočajo razširljivost arhitekture in upravljanje s storitvami za zagotavljanje konsistentnosti izvajanja interakcij. Kasneje je na realnem primeru prijave igralca podan prikaz upravljanja strežniške arhitekture z uporabo minimalnih storitev.

V nadaljevanju so predstavljene še specifične storitve, ki so po funkcionalnosti bližje zagotavljanju podpore elementom igralnosti masovnih večigralskih spletnih iger. Prikazni so tudi njihovi skupni osnovni programski gradniki, ki definirajo pravila za pretok interakcij in zagotavljajo uspešno simulacijo navideznega sveta in interakcij igralcev.

5.1 Minimalne storitve

Zaradi izjemno spremenljivega števila igralcev v masovni večigralski spletni igri se pojavi potreba po dinamičnem vklopu in izklopu instanc igralnih storitev. Instanca storitve dejansko pomeni le ločen programski proces, v katerem se omenjena storitev izvaja. Minimalne storitve tako zadovoljujejo potrebo po centralnem sistemu za nadzor nad trenutno zasedenostjo strežnikov in razporeditvijo igralnih storitev. Minimalne storitve so storitve, ki zadostujejo osnovnemu namenu upravljanja s celotno strežniško arhitekturo igre. Ta se neposredno ne nanaša na samo funkcionalnost igre, ampak skrbi za zagon in administracijo igralnih storitev. V kontekstu minimalnih storitev so prikazane skrbniška, izvajalna, imenska in kontaktna storitev, vendar se v koncept po potrebi lahko vključi dodatne in bolj specifične storitve.

5.1.1 Skrbniška storitev

Skrbniška storitev je storitev, ki nadzoruje izvajanje vseh ostalih storitev v strežniški arhitekturi. V tem konceptu je prisotna le ena instanca skrbniške storitve s centralnim nadzorom. Vanjo so lahko vgrajeni mehanizmi za nadzor, zagon in ustavitev posameznih storitev preko uporabniškega vhoda. Vhod in nadzor nad storitvami sta lahko upravljana ročno prek neposrednega vhoda skrbnikov igre ali preko avtomatizacije procesa. Upravljanje temelji na uspešni vzpostavitvi komunikacijskega kanala med imensko in skrbniško storitvijo in na uspešno definiranih pravilih in omejitvah igralnih storitev.

5.1.2 Izvajalna storitev

Izvajalna storitev je storitev, ki se izvaja na vsakem fizičnem strežniku. Posebnost izvajalne storitve je popolna samostojnost in avtomatski zagon ob zagonu strežnika. Izvajalna storitev skrbi za zagon in ustavitev storitev na istem fizičnem strežniku in služi kot vmesnik med storitvami strežnika in skrbniško storitvijo. Storitve, zagnane s strani izvajalne storitve, morajo z njo vzpostaviti ustrezen komunikacijski kanal. To omogoča nemoteno obveščanje o stanju posamezne storitve in detekcijo sesutja, preobremenitve ali ustavitve. Storitve z izvajalno storitvijo vzpostavijo način komunikacije in časovni interval za sporočanje njihovega stanja. Izvajalna storitev v tem konceptu v primeru, ko prejme zahtevo s strani skrbniške storitve, neposredno zaganja in ustavlja storitve na tem strežniku.

5.1.3 Imenska storitev

Imenska storitev služi kot imenik vseh storitev v strežniški arhitekturi. Vse izvajalne storitve morajo v omenjenem konceptu v določenem časovnem intervalu obveščati imensko storitev o stanju storitev, za katere so zadolžene. Imenska storitev ima za vsako storitev informacijo o njeni verziji, verziji navideznega sveta, unikatni številki, internetnem naslovu fizičnega strežnika ipd. Imenska storitev od izvajalnih storitev redno prejema informacije o zasedenosti strežnika in stanju pripadajočih storitev. Te informacije primerja z vnaprej definiranimi pravili in omejitvami storitev ter uspešno ugotavlja preobremenjenost fizičnih strežnikov. Vnaprej določena pravila in omejitve so lahko poljubne, primer sta omejitev maksimalne porabe procesorja ali pa maksimalno število igralcev v nekem območju navideznega sveta.

5.1.4 Kontaktna storitev

Kontaktna storitev je vmesna storitev med odjemalcem in strežniško arhitekturo. Ker je strežniška arhitektura v tem konceptu zaradi zagotavljanja varnosti z uporabo večih mrežnih kartic na kontaktnih strežnikih fizično ločena od javno dostopnega omrežja, postane kontaktna storitev edina dostopna točka v strežniško arhitekturo. Kontaktnih storitev je lahko poljubno mnogo, prijavna storitev pa razpolaga s podatki o zasedenosti vsake kontaktne

storitve (npr. preko imenske storitve). Ker so kontaktne storitve edina dostopna točka za vse igralce, lahko ob njihovem premajhnem številu postanejo ozko grlo komunikacije.

V tem konceptu je prijavni strežnik ob uspešni prijavi odjemalca zadolžen za uspešno preusmeritev odjemalca na ustrezno kontaktno storitev. Kontaktna storitev je torej tretji nivo razpošiljanja sporočil. Optimalno razširljivost je tako moč doseči z ločitvijo na podatkovne centre, ustreznimi nastavitvami razporejanja sporočil na usmerjevalnikih in ustrezno implementiranimi mehanizmi preusmerjanja sporočil odjemalcev na kontaktnih storitvah. Kontaktna storitev poleg osnovne razširljivosti podpira tudi razpošiljanje sporočil in interakcij glede na njihov tip. Tako npr. interakcije med igralci v obliki pogovora ne obremenjujejo igralnega strežnika, temveč so avtomatsko preusmerjene na ustrezne strežnike s pogovornimi storitvami.

Dodatna dobra lastnost kontaktnih storitev je dejstvo, da programska oprema igralca nikoli neposredno ne vzpostavi povezave z igralnimi strežniki. V primeru da se kateri od procesov sesuje ali pa pride do zagona nove storitve, je postopek za uporabnika transparenten. Igralec »čaka« na kontaktni storitvi do vzpostavitve ali ponovnega zagona ustrezne instance igralnega procesa in ne izgubi povezave z igro. Posledično se mu ni potrebno znova prijavljati v sistem, kar omogoča boljšo uporabniško izkušnjo.

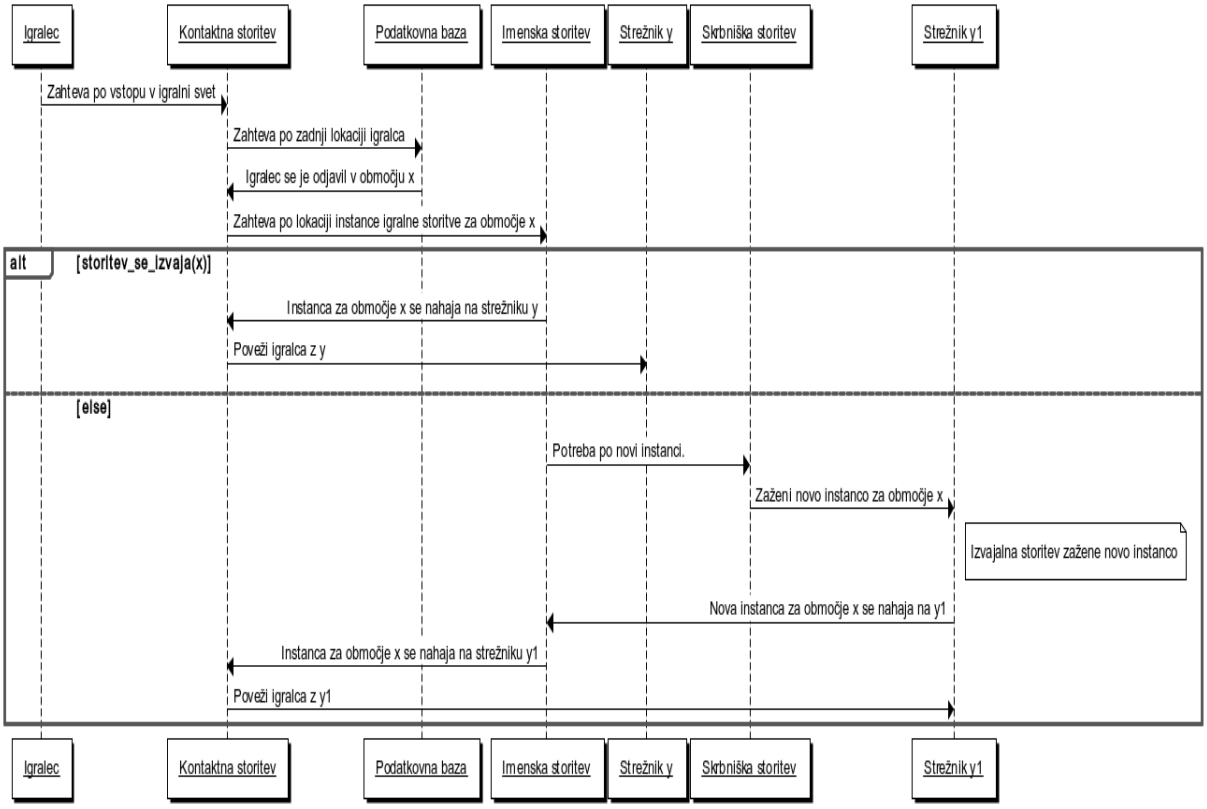
5.1.5 Prikaz postopka na konkretnem primeru

Konkreten prikaz postopka interakcij med minimalnimi storitvami je mogoče prikazati na realnem primeru. Predpostavimo, da je strežniška arhitektura zagnana, kar pomeni, da se minimalne storitve izvajajo in so vnesene v imenik imenske storitve.

Primer temelji na primeru prijave novega igralca v navidezni svet. To konkretno pomeni, da je odjemalec vzpostavil povezavo z dodeljeno kontaktno storitvijo. Kontaktna storitev preko komunikacije s podatkovno bazo prejme informacijo o zadnji lokaciji igralca v navideznem svetu (eden od elementov igralnosti je vstop v navidezni svet na isti lokaciji, kjer je bil opravljen zadnji izhod). Kontaktna storitev nato preko odprtega komunikacijskega kanala z imensko storitvijo poizve o stanju storitve za to območje navideznega sveta. V primeru da imenska storitev v svojem imeniku ne najde aktivne instance igralne storitve za omenjeno območje igralnega sveta, le-ta še ne obstaja. Potrebna je dodatna komunikacija s skrbniško storitvijo, ki odredi zagon nove instance igralne storitve za omenjeno območje. Izvajana storitev po prejeti zahtevi uspešno zažene igralno storitev in se uskladi z imensko storitvijo o internetnem naslovu strežnika igralne storitve in načinu komunikacije.

Po uspešnem zagonu storitve se ta vnese v imenik imenskega strežnika, ta pa komunikacijski storitvi sporoči informacijo o igralni storitvi. Komunikacijska storitev tako pridobi potrebno informacijo o lokaciji igralne storitve, na katero preusmeri čakajočega odjemalca. S tem je bil preko komunikacijske storitve uspešno vzpostavljen končni komunikacijski kanal med igralno storitvijo in odjemalcem. Sam postopek je z izjemo čakanja na zagon igralne storitve za odjemalca transparenten.

Komunikacijska storitev ob prijavi novih igralcev v isto območje preko imenske storitve ponovno pridobi podatke o igralni storitvi za to območje in v primeru obstoječe ustrezne instance igralne storitve nove uporabnike nemudoma preusmeri nanjo. Igralna storitev je v nenehnem kontaktu s pripadajočo izvajalno storitvijo, kateri poroča stanje o številu igralcev, izvajalna storitev pa te informacije skupaj z informacijami o zasedenosti fizičnega strežnika usklajuje z imensko storitvijo. Tako lahko imenska storitev v primeru prezasedenosti procesorja ali pomnilnika odredi premik ostalih storitev s prezasedenega strežnika na ostale, manj zasedene strežnike ali pa v primeru presežka maksimalnega števila igralcev odredi zagon nove instance igralne storitve za enako območje igralnega sveta. Sekvenčni diagram na sliki 15 prikazuje način komunikacije med minimalnimi storitvami za omenjen primer.



Slika 15: Sekvenčni diagram medprocesne komunikacije minimalnih storitev za primer prijave novega uporabnika v navidezni svet.

5.2 Specifične storitve

Minimalne storitve same po sebi ne zagotavljajo podpore funkcionalnostim elementov igralnosti in simulaciji navideznega sveta, ampak so namenjene predvsem upravljanju s storitvami in razpošiljanju sporočil igralcev. Dodatno razpošiljanje se lahko vrši tudi z razporejanjem različnih sporočil odjemalcev. Za uspešno razporejanje sporočil na ta način mora biti, podobno kot med storitvami v strežniški arhitekturi, tudi med kontaktno storitvijo in odjemalcem vzpostavljen protokol za izmenjavo sporočil. Preprosta različica protokola je lahko 8 bitna maska sporočila, ki določa unikaten tip sporočila. Tako imajo npr. sporočilo za premik igralca, sporočilo za interakcijo z drugim igralcem in sporočilo za pogovor med igralci drugačno unikatno masko.

Protokol za izmenjavo sporočil je neposredno vezan na elemente igralnosti igre, kar skupaj z arhitekturo porazdeljenih strežniških storitev omogoča poljubno nastavljivost in razširljivost strežniške arhitekture. Kot posledico lahko v omenjenem konceptu tako opazimo dodatno funkcionalno delitev storitev na igralno, pogovorno, prijavno in podatkovno storitev. Sama delitev po funkcionalnosti sicer ni zahtevana, a omogoči zmanjšanje procesne obremenjenosti strežnikov. Omenjeni koncept omogoča zagon vseh specifičnih storitev na enem fizičnem strežniku oz. na večih, fizično ločenih strežnikih. Praviloma je izjema prijavna storitev, ki skrbi za prijavo uporabnikov in preusmeritev uporabnikov na optimalne komunikacijske storitve. Prijavna storitev mora biti zaradi zahteve po neposrednem kontaktu dostopna preko javnega omrežja. Omenjen koncept omogoča dodatno zagotavljanje varnosti z uporabo dodatnega požarnega zidu na ločeni prijavni podatkovni bazi.

5.2.1 Prijavna storitev

Prijavna storitev je navadno prva storitev, s katero odjemalec vzpostavi komunikacijo. Storitev sama po sebi ni obvezna, a je v večini iger prisotna. Pri prijavi se ustrezno preveri in potrdi stanje računa ter preveri poslane prijavne podatke. Prijavna storitev v primeru uspešne prijave izda zahtevo sejni storitvi za vzpostavitev igralne seje z odjemalcem, kar je predpogoj za uspešno povezavo na igralne storitve. Odjemalcu se v okviru medprocesne komunikacije na strežnikih dodeli unikatna oznaka, ki ga v času veljavne seje določa. V večjih masovnih večigralskih spletnih igrah je vzpostavljenih več podatkovnih centrov z namenom zmanjševanja zakasnitve paketov preko interneta. Prijavna storitev je lahko osamljena storitev na enem izmed podatkovnih centrov in poleg prijave lahko skrbi za preusmeritev igralcev

glede na njihovo geografsko lokacijo. To je seveda mogoče le v primeru, ko podatkovna baza igralcev vsebuje geografske podatke igralcev. Prijavna storitev glede na geografske podatke igralca preusmeri na ustrezne podatkovne centre, kar omogoči boljo zanesljivost igre.

Prijavna storitev je avtoriteta pri zagotavljanju dostopa do igre, saj v zaledni bazi podatkov ugotavlja informacije o računu igralca. To je izrednega pomena, saj je igra lahko plačljiva. Zaradi velike potrebe po varnosti in omejevanju dostopa do prijavne storitve in prijavne baze podatkov je varnost glavno vodilo pri načrtovanju te storitve. Prijavna baza podatkov je skoraj vedno ločena od prijavne storitve in skrbno nadzorovana s strani razvijalcev igre. V omenjenem konceptu je prijavna storitev razširjena s funkcionalnostjo razpošiljanja uporabnikov na optimalne kontaktne storitve. Tako lahko s poizvedbami na imensko storitev ugotovi najmanj zasedeno komunikacijsko storitev in uporabniku pošlje zahtevo za vzpostavitev povezave, to pa prepreči ozko grlo kontaktne storitve.

5.2.2 Pogovorna storitev

Pogovorna storitev je v omenjenem konceptu podana le kot konkreten primer funkcionalne delitve storitev, ki so bolj specifične elementom igralnosti igre. Predpostavljeno je, da igra kot element igralnosti omogoča pogovor med igralci. Ta pogovor je lahko neposreden med igralci oz. posreden med območji navideznega sveta. Koncept ponuja možnost ločitve pogovorne storitve od igralne storitve, saj je za uspešno ločitev potrebna le ločitev programske kode za pogovor in definiranje nove unikatne maske sporočil za pogovorna sporočila. Taka ločitev ima dobre in slabe lastnosti. Slabe so povečanje kompleksnosti celotnega sistema in dodatno razpošiljanje sporočil med podsistemi, medtem ko se glavna dobra lastnost pokaže pri odpovedi pogovorne storitve. Če pogovorna storitev odpove, se igralna storitev še vedno izvaja. Igralci kljub nezmožnosti uporabe pogovora do ponovnega zagona pogovorne storitve še vedno lahko igrajo v navideznem svetu, kar je večinoma sprejemljiva odločitev.

5.2.3 Podatkovna storitev

Ena najpomembnejših storitev za uspešno izvajanje interakcij v masovni večigralski spletni igri je podatkovna storitev. Ta storitev v omenjenem konceptu skrbi za hrambo podatkov o igralcih in stanju navideznega sveta in pripadajočih objektov.

Masovne večigralske spletne igre lahko vsebujejo zelo veliko število igralcev in objektov ter velik in konstantno spreminjajoči se navidezni svet. Posledica tega je ogromna

količina podatkov, ki jih igra proizvede. Natančno in pravočasno načrtovanje principov podatkovnega skladiščenja je ena najpomembnejših odločitev v razvoju masovnih večigralskih spletnih iger.

Za shranjevanje podatkov se v omenjenem konceptu uporablja zaledno bazo podatkov poljubnega tipa in vmesno podatkovno storitev. Skladno z ogromno količino podatkov, ki jih je potrebno hraniti, in frekvenco posodabljanja in poizvedovanja se lahko uporabi standardne rešitve na nivoju podatkovne baze, ki temeljijo predvsem na indeksiranju in normalizaciji vnosov ter na izogibanju uporabe preveč kompleksnih poizvedb. Pristopi k načrtovanju podatkov in izbiri podatkov za shranjevanje se močno razlikujejo med različnimi masovnimi večigralskimi spletnimi igrami. To je posledica velike odvisnosti od elementov igralnosti, količine podatkov in želene frekvence poizvedb in posodobitev. Izbira načina hranjenja podatkov je torej neposredno odvisna od dejavnikov igre in ne more biti vnaprej definirana.

Podatkovna storitev je v omenjenem konceptu zasnovana tako, da ustreza osnovnim zahtevam obdelave podatkov v masovnih večigralskih spletnih igrah. Pomembna sta odzivnost in zanesljivost posodobitve podatkov in izvajanja poizvedb. En izmed načinov, kako doseči želeno funkcionalnost in odzivnost, je prisotnost ločene vmesne storitve, ki skrbi za izmenjavo podatkov med igralnimi storitvami in podatkovno bazo. Ker se v podatkovne storitve lahko integrira dodatne in nestandardne mehanizme, specifične za vsako igro posebej, se s tem lahko znatno pohitri odzivnost celotnega podatkovnega sistema. Ostale storitve v arhitekturi tako nikoli neposredno ne dostopajo do podatkovne baze. Podatkovna storitev sama nadzoruje dostop do podatkovne baze preko nadzorovanih komunikacijskih kanalov, omogočena pa je tudi uporaba predpomnenja podatkov. Ker se podatki nahajajo v pomnilniku podatkovne storitve, so poizvedbe in spremembe podatkov izjemno hitre. Ta pristop omogoča razširljivost standardnih pristopov za poizvedbe. Razvijalci lahko razvijejo nestandardne oblike poizvedb in kompleksnejše strukture podatkov, brez da bi bili neposredno odvisni od tipa izbrane podatkovne baze. Podatkovna storitev mora v zaledju v odvisnosti od tipa izbrane podatkovne baze imeti izdelane mehanizme interakcije s samo podatkovno bazo, vendar se s tem pristopom pridobi nadzor nad frekvenco posodobitev in tipi poizvedb.

Kjer je smiselno, se podatkovne storitve za izračune in posodobitve v realnem času ne uporablja neposredno. Podatki o igralcih in stanju sveta se tako večinoma nahajajo v pomnilniku igralnega strežnika in se le občasno sinhronizirajo s podatkovno storitvijo. To zagotovi sprejemljivo zanesljivost sinhronizacije podatkov med igralnimi in podatkovnimi storitvami in zelo učinkovito odzivnost. Časovni interval sinhronizacije med igralnimi in podatkovnimi storitvami ter med podatkovnimi storitvami in podatkovno bazo mora biti ustrezno izbran v odvisnosti od tipa in pomembnosti podatkov. Tako morajo biti pomembni podatku v primeru sesutja igralnega strežnika sinhronizirani in shranjeni, medtem ko so manj pomembni lahko izgubljeni.

Ustreznost sistema hranjenja podatkov v omenjenem konceptu se prav tako odraža v smiselnem filtriranju poizvedb. V primeru, da neka interakcija igralca v navideznem svetu neposredno pomeni kreiranje poizvedbe na podatkovno storitev, lahko igralec z veliko kliki ustvari veliko poizvedb, kar lahko zmanjša odzivnost podatkovnega sistema.

5.3 Igralne storitve

Omenjen koncept funkcionalne delitve strežniške arhitekture na storitve omogoča optimalno razširljivost in preprostost razvoja dodatnih storitev. Z ustrezno definiranimi protokoli in pravili za izmenjavo interakcij med storitvami je arhitektura poljubno razširljiva in učinkovita, omogoča pa tudi enostavno upravljanje celotne strežniške arhitekture. Kljub zadovoljivi funkcionalni delitvi storitev na minimalne in specifične so igralne storitve avtoriteta pri simulaciji navideznega sveta. Te so zadolžene za dejansko simulacijo in upravljanje navideznega sveta in so posledično najbolj obremenjene. Ker so podrobnosti igralnih storitev neposredno popolnoma odvisne od elementov igralnosti masovne večigralske spletne igre, so si konkretne implementacije pogonov za fiziko, umetno inteligenco, upravljanje s premiki ipd. med seboj zelo različne. Omenjen koncept sicer omogoča delitev omenjenih pogonov na ločene storitve, vendar se v tem segmentu predpostavi, da vsi tečejo znotraj iste igralne storitve.

Ne glede na izbiro načina ločitve pogonov igralne storitve je potrebno poenotenje načina simulacije igralcev in sveta. To vključuje poenotenje sporočil in akcij, ki jih igralec pošilja igralnemu strežniku na isti vmesnik, ter enoten in razširljiv mehanizem za upravljanje z objekti igre. To omogoča, da se objektom igre dinamično dodaja komponente, zadolžene za spreminjanje in simulacijo interakcij v navideznem svetu. V tem poglavju je predstavljen enoten sistem za sprejemanje, obdelavo in razpošiljanje interakcij uporabnikov za potrebe simulacije navideznega sveta. Ta sistem se lahko uporabi ločeno na katerikoli igralni storitvi in omogoča popolno razširljivost komponent in pogonov simulacije.

5.3.1 Osnovni zahteve

Za celovito učinkovitost sistema so v omenjenem konceptu zahtevane štiri osnovne zahteve. Prva izmed zahtev je poenotenje interakcij in sporočil, ki si jih strežnik in odjemalec izmenjujeta. Vsa sporočila, interakcije in rezultati interakcij simulacije navideznega sveta so poenotena preko koncepta dogodka, vsak dogodek pa ima poleg dejanske informacije o

načinu simulacije navideznega sveta tudi informacijo o lastnem tipu. Implementacije mehanizmov za razpošiljanje dogodkov in upravljanje z njimi preko enotnega vmesnika berejo in razpošiljajo dogodke, novi tipi sporočil pa se lahko enostavno kreirajo v obliki novih unikatnih dogodkov.

Druga zahteva je poenotenje stanja objektov igre in njihovega prehoda. Glede na elemente igralnosti igre se definira več specifičnih objektov stanj, ki definirajo metode za njihov prehod in potrditev, omogočajo pa tudi zakasnjeno izvajanje. Tako je mogoče poenotenje prehoda stanja med objekti igre za pogone zajema uporabniškega vnosa, umetne inteligence, upravljanja s premiki igralca ipd. Ker imajo dogodki v odvisnosti od stanja definirano tudi trajanje stanja oz. čas prehoda in želen začetek, se dogodke lahko razvršča glede na pričetek simulacije. Tako se lahko ob prehodu stanja objektov igre brez čakanja na dejansko kreiranje dogodkov nemudoma kreirajo novi dogodki, ki se bodo izvedli v prihodnosti. Objekti za upravljanje stanja so sami zadolženi za razvrščanje in razporejanje dogodkov v vrste in vsebujejo lastno definirane metode za izvedbo prehoda med stanji. Konkretni koncepti različnih objektov stanj za objekte igre so prikazani na primeru v naslednjem poglavju.

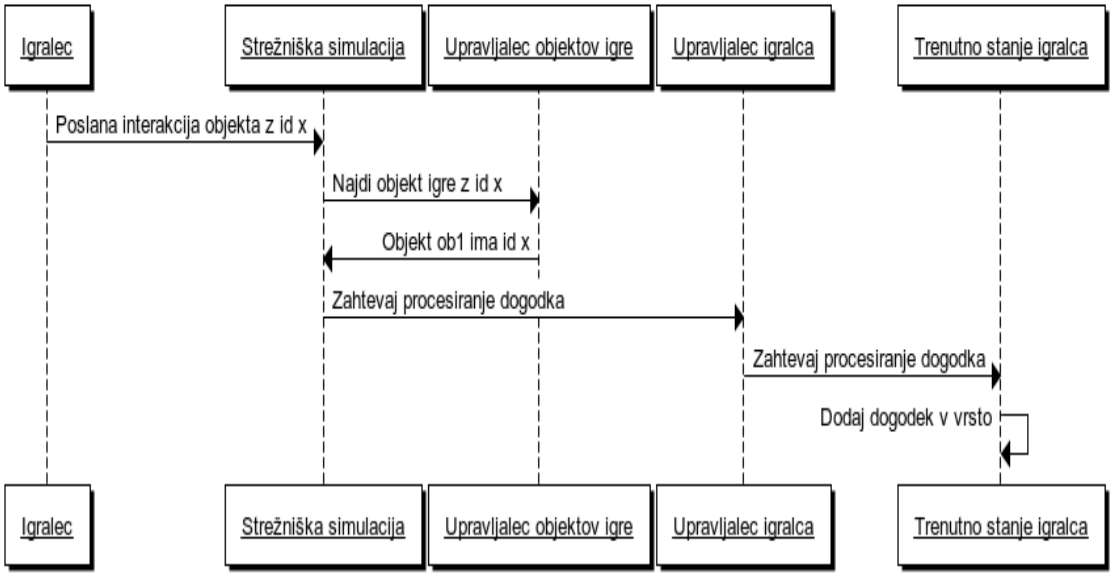
Tretja zahteva je poenotenje objektov igre. Objekti igre pod enak vmesnik združijo vse elemente igre, ki so del simulacije. To vključuje celotno območje igralnega sveta, računalniško vodene igralce, odjemalce igre, predmete v igri ipd. Vsak objekt igre ima definirano številko objekta igre, kateremu pripada, kar omogoča hierarhično vsebovanost in upravljanje z objekti igre. Ker simulacija poteka simultano na strežniku in na odjemalcu, morajo imeti vsi objekti igre identično kopijo na strežniku in odjemalcu, ta pa mora biti ustrezno sinhronizirana. Strežniška simulacija je sicer prioriteta pri odločanju o rezultatih interakcij, a je kopija objektov igre na odjemalcu potrebna za izris navideznega sveta in napovedovanje stanja igre v mehanizmih za kompenzacijo zakasnitve paketov. Kompleksnejši primer objekta igre je vmesnik akter, ki določa objekt igre nekega igralca. Akter mora poleg standardnih lastnosti objekta igre imeti definirano tudi lastnost lastnega trenutnega stanja, ki ga spreminja glede na uporabniški vhod igralca.

Četrta zahtevaje poenotenje upravljavcev objektov igre. Ker se v igri neprestano ustvarjajo novi objekti igre, je zahtevana prisotnost ustreznih upravljavcev za kreiranje objektov igre, kateri ustrezno vršijo izdelavo unikatnih objektov. Upravljavci objektov igre lahko neposredno vplivajo na prehod med stanji objektov igre z implementacijo lastnih pravil igre in upravljanjem prejetih dogodkov. Ker se lahko upravljavci dinamično dodajo ali odstranijo iz objekta igre, to omogoča visoko prilagodljivo ločitev objektov igre. Tako se objekt igre npr. s preprostim dodajanjem upravljavca pogona umetne inteligence in pripadajočega objekta stanja prične odzivati na dogodke umetne inteligence, z dodajanjem upravljavca nadzornega pogona pa moderator v igri enostavno pridobi nadzor in pogled

nekega igralca, ki ima npr. težave z razumevanjem igralnih elementov. Dodajanje upravljavca statistike omogoči enostavno beleženje spremembe nekega objekta igre, kar je izjemnega pomena za oblikovalce igre in za merjenje podatkov o navideznem svetu.

5.3.2 Primer celovite simulacije navideznega sveta

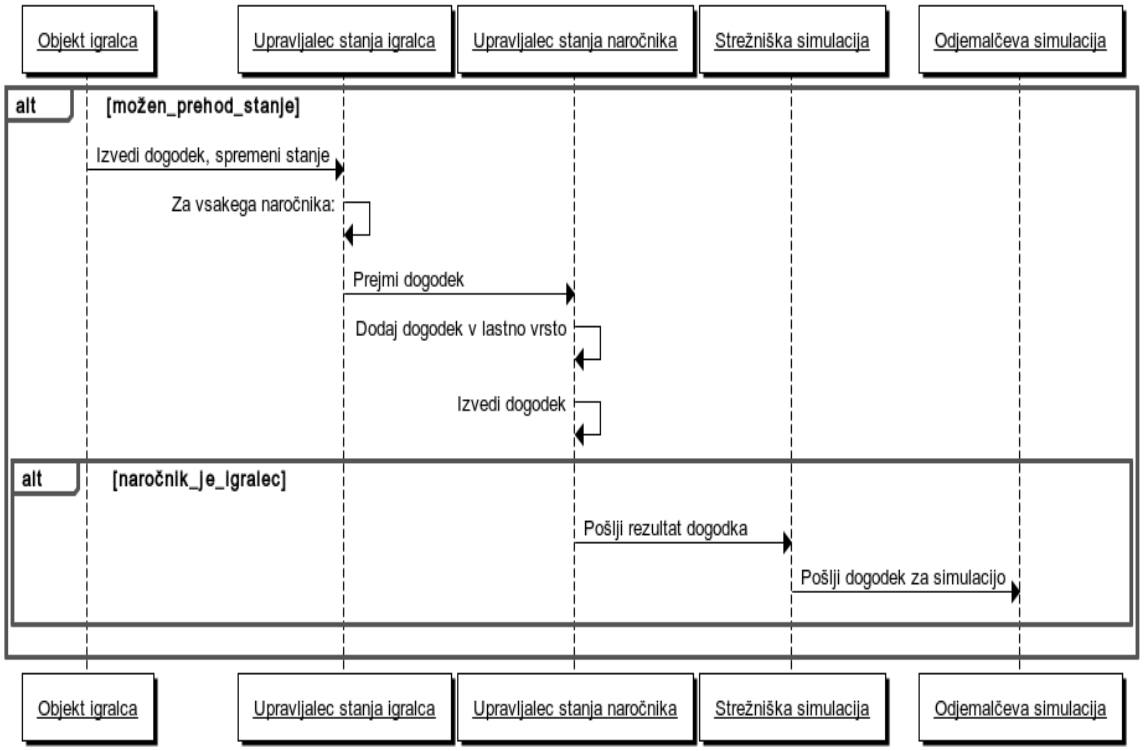
Z ustrezno delitvijo na osnovne gradnike igralne storitve je mogoča ponazoritev primera celovite simulacije sveta v odvisnosti od poslanega dogodka. Sekvenčni diagram na sliki 16 prikazuje začetek simulacije s prejemom poslanega dogodka, določitev igralca kot objekta igre na območju igralnega sveta in dodajanje dogodka v vrsto dogodkov. Dogodek se doda v vrsto le v primeru, ko trenutno stanje igralca dovoljuje prejemanje novih dogodkov. Dogodek se v vrsto doda glede na željen čas simulacije dogodka, saj omenjen koncept omogoča simulacijo dogodka v prihodnosti.



Slika 16: Sekvenčni diagram prejema dogodka igralca.

Ker je proces strežniške simulacije samostojen, ima definirano glavno zanko simulacije. Ta zanka v nekem časovnem intervalu kliče metodo za simulacijo vseh dogodkov v navideznem svetu, ta metoda pa preko upravljavca za napovedovanje dogodkov pošilja zahteve vsem objektom igralcev za procesiranje dogodkov iz lastnih vrst. Tako je na strežniku vsak objekt igralca oz. akter zadolžen za svojo vrsto dogodkov, te pa lahko vrši glede na lastno definirane metode za prehode med stanji. Ker je možna implementacija sistema za medsebojno odvisnost

objektov igre preko arhitekturnega stila objavi-naroči, je mogoče spremembo stanja sporočiti drugim objektom igre. To je ena izmed glavnih prednosti omenjenega pristopa k načrtovanju strežniške arhitekture, saj se s tem učinkovito upravlja interesno polje igralca. Rezultata vsake interakcije torej ni potrebno razpošiljati vsem igralcem, ampak samo tistim znotraj interesnega polja. Vsi igralci so tako preko upravljavca interesnega polja objektov igre naročeni na dogodke o spremembi stanja drugih objektov igre znotraj njihovih interesnih polj. To znatno pohitri procesiranje dogodkov in zmanjša porabo pasovne širine na strežniški farmi. Na sekvenčnem diagramu na sliki 17 je prikazan potek obdelave interakcije in razpošiljanja dogodkov naročenim objektov.



Slika 17: Prikaz izvedbe dogodka in pošiljanje novega dogodka kot rezultata vsem naročenim objektom igre.

Po prejemu dogodka se ta pošlje vsem potrebnim odjemalcem, njihova simulacija igre pa spremembo prikaže na zaslonu. Upravljavci na odjemalčevi strani s tem pridobijo zadostno število informacij o spremembi navideznega sveta. Upravljavci, zadolženi za kreiranje objektov igre, tako kreirajo posebne objekte igre, prisotne le na strani odjemalca. Ti objekti igre torej niso akterji in nimajo identične kopije na strežniku. Primer je npr. kreiranje vizualnih efektov in zvočnih učinkov. Ta pristop omogoča učinkovito ločitev simulacije in prikaza navideznega sveta, kar razvijalcem omogoči podporo različnim platformam brez

spreminjanja sistemov za simulacijo navideznega sveta. Ker je omenjen koncept le podlaga za specifična pravila masovne večigralske spletne igre, je neodvisen od specifičnih elementov igralnosti igre. Vsa pravila igre in konkretne implementacije elementov igralnosti igre se nahajajo v obliki ločenih upravljavcev objektov igre, definiranih stanj in pravil za prehod med stanji. Vključitev teh objektov v ta sistem je dinamična in preprosta, saj so vse interakcije in prehodi poenoteni pod enak vmesnik, za ustrezno izvedbo pa skrbi zaledni sistem igralnega procesa.

5.4 Primerjava koncepta z obstoječimi rešitvami

V tem poglavju je predstavljena primerjava koncepta, ki smo ga razvili v okviru diplomskega dela, z arhitekturnimi rešitvami v nekaterih sodobnih masovnih večigralskih spletnih igrah. Primerjava je prikazana na podlagi dostopnih informacij sodobnih masovnih večigralskih spletnih iger Guild Wars 2 in Eve Online, kar omogoča prikaz konkretnih prednosti, slabosti, ter posledic na igralne elemente. Ker sodobnejša masovna večigralska spletna igra Guild Wars 2 sledi in učinkovito izrablja tradicionalni pristop k načrtovanju strežniške arhitekture, na katerem sloni tudi naš koncept, omogoča prikaz primerjave na konkretnih primerih igralnih elementov igre. Za razliko od tradicionalnega pristopa pa nekoliko starejša masovna večigralska spletna igra Eve Online temelji na povsem drugačnem pristopu k načrtovanju arhitekture, ta pa je primeren za primerjavo razlik v omenjenem konceptu.

Primerjava našega koncepta in uporabljenih rešitev se v največji meri lahko prikaže na posrednih posledicah izbire arhitekture, kar so npr. razširljivost območij navideznega sveta in reševanje problemov v primeru skokovitega povečanja števila igralcev. Primerjavi sta opravljeni glede na vsako igro posebej, prikazane pa so tako razlike in podobnosti kot tudi konkretni primeri elementov igralnosti.

5.4.1 Primerjava z Guild Wars 2

Guild Wars 2 je masovna večigralska spletna igra, ki je izšla leta 2012. Z več kot tremi milijoni prodanih kopij sodi v vrh omenjenega žanra. Strežniška arhitektura sloni na tradicionalnem pristopu, saj navidezni svet deli na območja. Ta izbira pri načrtovanju strežniške arhitekture se nanaša na izhodišča, na katera se sklicuje tudi v diplomskemu delu razvit koncept. Prehodi med območji tako igralca za trajanje prehoda postavijo v stanje čakanja, kar se na programski opremi igralca prikaže kot statičen prikaz nalaganja novega

območja. Poleg tega je navidezni svet kloniran na več enakih instanc, vsak igralec pa ima pripadnost neki instanci navideznega sveta oz. enemu izmed 51 igralnih svetov. Vsak igralni svet razpolaga z lastnim strežniškim grozdom, za vsako območje navideznega sveta pa skrbi ločen programski proces. Z izjemo pogovora in združevanja igralcev v skupine je onemogočena neposredna interakcija med igralci različnih območjih navideznega sveta.

Poleg delitve območij ločenim igralnim procesom se masovna večigralska spletna igra Guild Wars 2 poslužuje tudi funkcionalne delitve na ločene storitve. Tako sta, kot je običajno pri arhitekturi masovne večigralske spletne igre, prisotna ločena prijavna in kontaktna storitev. Igralci so tako neposredno povezani samo z omenjenima storitvama, za medprocesno komunikacijo pa skrbi zaledni sistem. Funkcionalna delitev na storitve je vidna tudi v ločeni storitvi za trgovanje in pogovor med igralci. Pogovorna storitev v ozadju skrbi za ustrezno združevanje igralcev v skupine in posledično omogoča pogovor med igralci različnih igralnih svetov. Storitev za nadzor trgovanja lahko vodi enotno instanco trgovine med igralci vseh igralnih svetov tako evropskega kot ameriškega podatkovnega centra.

Ločitev območij na posamezne procese je s pridom izkoriščena v primeru skokovitega povečanja števila igralcev v nekem območju. V primeru, ko število igralcev preseže zmogljivosti strojne opreme oz. vnaprej določeno maksimalno število igralcev, sistem avtomatsko zažene t.i. prelivno instanco območja. Prelivna instanca območja nima pripadnosti igralnemu svetu in posledično združuje igralce različnih igralnih svetov. Namen prelivnih instanc je izogibanje čakalnim vrstam za različna območja navideznega sveta, kar omogoča boljšo uporabniško izkušnjo. Igralci v prelivnih instancah so avtomatsko vključeni v čakalno vrsto za ustrezno instanco igralnega sveta, kateremu pripadajo, a to ne preprečuje nadaljnjega igranja na prelivni instanci.

Ločitev območij na ločene procese se izkaže pri zagotavljanju razpoložljivosti strežniške arhitekture v primeru posodobitev igralnega sveta. Ko razvijalci pripravijo novo posodobitev igralnega sveta se vsi procesi območij igralnega sveta in vse zahtevane storitve avtomatsko klonirajo in zaženejo. Razlika med identičnimi kopijami je le v verziji območja igralnega sveta. Obstoječim igralcem v igri se prikaže obvestilo o novi posodobitvi in količini časa, ki ga lahko uporabijo v stari verziji navideznega sveta pred avtomatsko odjavo iz sistema. Igralci, ki nemudoma posodobijo svojo programsko opremo, in novo prijavljeni igralci se tako avtomatsko povežejo na instanco območja igralnega sveta z novejšo verzijo. To prikaže iluzijo popolne razpoložljivosti sistema, saj igralci ne čakajo na ponovni zagon strežniške arhitekture.

Masovna večigralska spletna igra Guild Wars 2 nudi podporo velikim bitkam med igralci v obliki tekmovanja treh igralnih strežnikov. Tako je večkrat prisotna bitka več kot 300 igralcev v sorazmerno majhnem območju, kar je identično problemu gnetenja. Ker strežniška

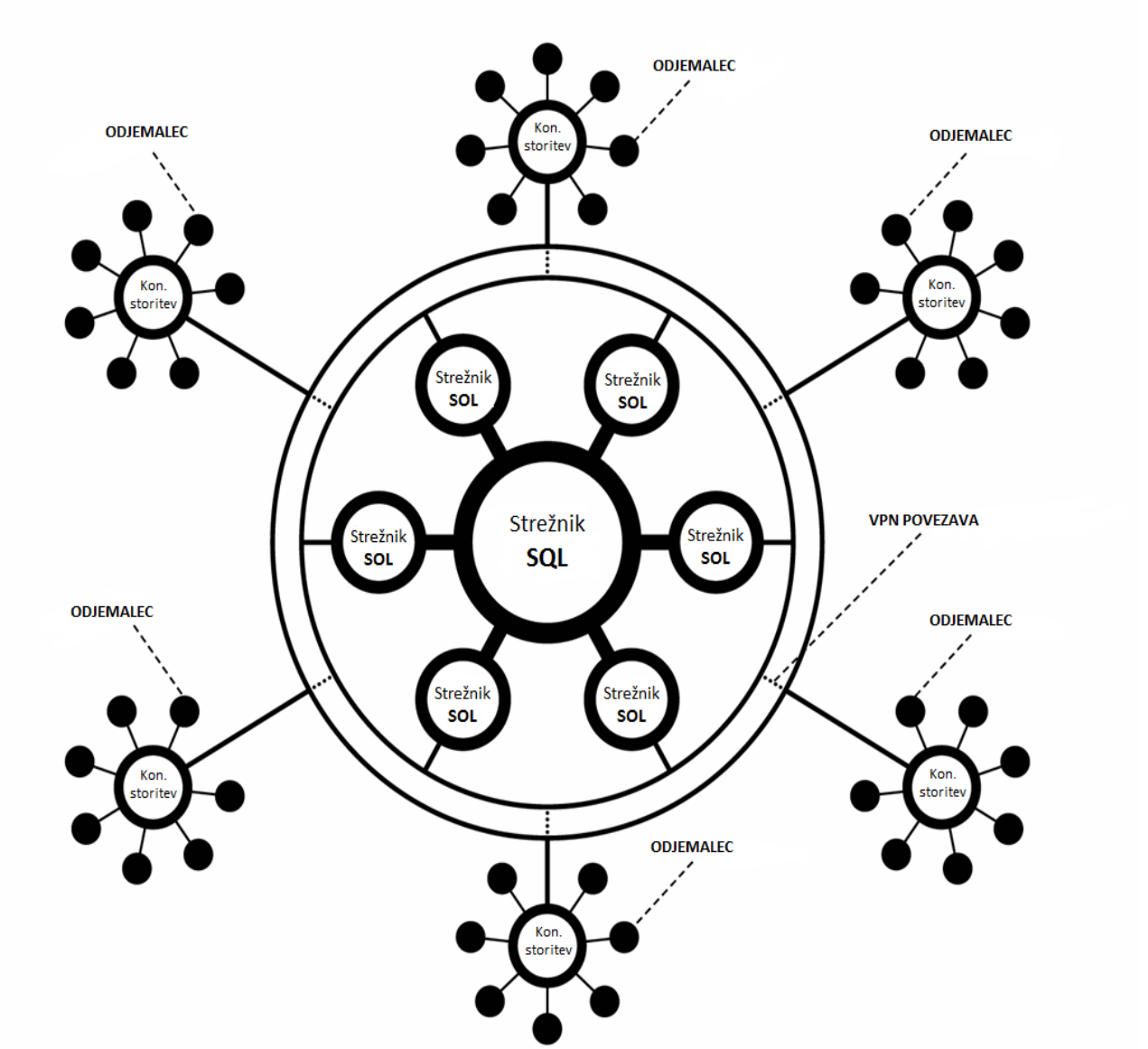
arhitektura ne podpira dinamičnega dodajanja procesorske moči prezasedenim procesom, so razvijalci primorani omejiti maksimalno število igralcev v teh tekmovanjih. Kljub omejitvi pa prihaja do odstopanj pri zagotavljanju zanesljivosti v velikih bitkah in posledično do zakasnitve pri izvajanju interakcij med igralci. Tu je ponovno možen prikaz podobnosti z našim konceptom preko prisotnosti funkcionalne delitve na storitve znotraj obsega elementov igralnosti igre. Naš enači akcije premika in akcijo izvajanja interakcije boja pod enak vmesnik dogodka, v primeru igre Guild Wars 2 pa je prikazana nadaljnja delitev teh dogodkov. Kljub prezasedenosti strežnika za izvajanje interakcij igralcev v obliki boja in posledično nekaj sekundni zamik pri njihovem izvajanju, poteka izvajanje interakcij za premike, skoke in izmike brez zakasnitve. To naznanja funkcionalno ločitev storitev za premikanje igralcev in za izvajanje boja na ločena fizična strežnika, podobno kot to podpira tudi v diplomskem delu razvit koncept.

Zaradi ločitve igralcev na ločene igralne svetove je, podobno kot pri našem konceptu, mogoča in podprta ločitev storitev za skladiščenje podatkov o stanju igralcev. Tako ima igra Guild Wars 2 centralizirano glavno podatkovno bazo in pripadajoče storitve za upravljanje s podatkovno bazo, prisotni pa so tudi ločeni procesi in podatkovne baze za lokalno shranjevanje podatkov o igralcih glede na njihov igralni svet.

Strežniška arhitektura v masovni večigralski spletni igri Guid Wars 2 je zaradi podobnih pristopov k reševanju problemov razširljivosti in implementacijo funkcionalno ločenih storitev dobro primerljiva s konceptom, razvitim v diplomskem delu.

5.4.2 Primerjava z Eve Online

Eve Online je masovna večigralska spletna igra, ki je izšla leta 2003, njena značilnost pa je popolnoma drugačen pristop k načrtovanju strežniške arhitekture od tradicionalnega modela. Na sliki 18 je prikazan primer strežniške arhitekture in povezave s podatkovno bazo v masovni večigralski spletni igri Eve Online.



Slika 18: Prikaz ciklične enotne strežniške arhitekture v Eve Online. Za vse solarne strežnike (strežnik SOL) obstaja skupna podatkovna baza SQL[4].

Edina podobnost z našim konceptom je prisotnost kontaktnih storitev. Namen kontaktnih storitev je podoben, saj poleg zagotavljanja varnosti dostopa preko fizične ločitve igralnih strežnikov skrbijo za učinkovito preusmerjanje uporabnikov na ustrezne igralne strežnike glede na njihovo zadnjo prijavo. Prvotni namen kontaktnih storitev je bila kompenzacija zakasnitve paketov, kar je leta 2003 predstavljalo dokaj velik problem. Zaradi znatnega napredka pri pretočnosti interneta sedaj ni več potrebe po geografsko ločeni prisotnosti kontaktnih storitev, tako da so vse kontaktne storitve locirane skupaj z igralnimi strežniki v istem podatkovnem centru v Londonu. Glavna razlika med našim konceptom in rešitvijo v

Eve Online je v upravljanju z igralnimi storitvami. Medtem ko v diplomskem delu razvit koncept dovoljuje in pričakuje funkcionalno delitev elementov igralnosti na ločene procese in kreiranje novih instanc območij navideznega sveta pri povečanju števila igralcev, strežniška arhitektura v Eve Online temelji na enem samem igralnem strežniku in na eni sami instanci navideznega sveta. Ta igralni strežnik je dejansko super-računalnik z ogromno procesorsko močjo, ima pa celo napredno strojno opremo podobno opremi, ki jo uporablja vojska[5].

Ker je navidezni svet v Eve Online prikazan kot sklop večih galaksij z manjšimi vesolji in ozvezdji, se vsako ozvezdje (ekvivalentno območju navideznega sveta) zažene kot ločen proces na enemu izmed strežnikih rezil super-računalnika. Ta proces skrbi za vse interakcije v danem ozvezdju in se upravlja kot ločen solarni strežnik (strežnik SOL). Ozvezdij je več kot 5000, manj zasedena oz. tista, ki zahtevajo manj procesorske moči, so lahko združena na istem strežniku SOL, tista, ki zahtevajo večjo procesorsko moč, pa imajo dodeljen lasten strežnik SOL. Za prehode med ozvezdji so, podobno kot v našem konceptu, zadolžene kontaktne storitve. Te so zadolžene za upravljanje z medprocesno komunikacijo strežnikov SOL. Poleg preusmerjanja igralcev so kontaktne storitve zadolžene tudi za predpomnenje statičnih oz. manj spremenljivih podatkov. Tako se znatno zmanjša število poizvedb na centralno podatkovno bazo.

Dodatna razlika med našim konceptom in arhitekturo v Eve Online je prisotnost ene same, centralizirane podatkovne baze. Ta je v neposrednem kontaktu z vsemi strežniki SOL in skrbi za shranjevanje stanja navideznega sveta in pripadajočih objektov. Ker je podatkovna baza osamljena, prihaja do izjemno visoke frekvence poizvedb in posodobitev, najvišja zabeležena frekvenca poizvedb pa znaša okoli 2000 poizvedb na sekundo. Podatkovna baza je locirana na ločenem strežniku in skrbno nadzorovana, optimizirana in načrtovana za maksimalno učinkovitost[4].

Drugačen pristop k upravljanju z interakcijami igralcev in dogodkov na posameznih strežnikih SOL. Glavna procesna moč za izvajanje dogodkov v arhitekturi Eve Online izvira iz uporabe posebne vrste programskega jezika Python, imenovane Stackless Python. Stackless Python preko koncepta Taskleta omogoča izjemno hiter sistem za izvajanje interakcij v ločenih programskih nitih. Tasklet je ovojnica za učinkovito upravljanje s programskimi nitmi brez večje izgube učinkovitosti sistema pri njihovem visokem številu. Ker je sistem za upravljanje s programskimi nitmi izjemo enostaven, se te lahko izvajajo izjemno hitro. Glavna prednost uporabe Stackless Pythona je možnost dinamične vključitve in izključitve programov, napisanih v drugih programskih jezikih, kot sta npr. C in C++. Vključitev je izjemno hitra, saj poteka ne glede na trenutno vsebino programskega sklada. V diplomskem delu razvit koncept temelji na nadzorovani simulaciji z omejenim številom procesov, arhitektura v Eve Online pa podpira neomejeno število sočasnih niti v obliki Taskletov.

Izbira strežniške arhitekture z osamljenim igralnim procesom omogoča izjemno povečanje maksimalnega števila sočasnih igralcev v navideznem svetu. Eve Online podpira do 65000 sočasnih igralcev na enem samem igralnem strežniku. Ker je igralni strežnik osamljen, imajo ti igralci možnost neposredne interakcije s katerimkoli drugim igralcem, kar v tradicionalnem pristopu zaradi reševanja problema gnetenja z uporabo ločenih igralnih svetov ni mogoče. Izbira takega pristopa pripomore k razvoju in implementaciji posebnih elementov igralnosti navideznega sveta. Tako je prisoten koncept teritorialne zasedenosti, kar v masovnih večigralskih spletnih igrah s tradicionalnim pristopom k načrtovanju strežniške arhitekture ni mogoče. Eve Online zaradi enotnosti navideznega sveta mogoča pripadnost nekega ozvezdja nekemu zavezništvu, saj obstaja le ena instanca tega ozvezdja. Boj med igralci je eden glavnih elementov igralnosti, saj izbrani strežniški model zaradi povečanja maksimalnega števila igralcev v bitki ponuja veliko večjo fleksibilnost od tradicionalnega sistema. Ker so vsi igralci del istega navideznega sveta, je teoretično mogoča bitka med vsemi igralci, do katere lahko pride brez namenskih strežnikov. Ker sistem ne dovoljuje pripadnosti igralca igralnim svetovom, se pripadnost lahko določi glede na elemente igralnosti (združevanje igralcev v zavezništva), kar omogoči boljšo uporabniško izkušnjo.

Slabost takšnega pristopa se pokaže v primeru skokovitega povečanja števila igralcev v posameznem ozvezdju navideznega sveta. Arhitektura kot celota je zadovoljivo razširljiva in omogoča veliko večje število sočasnih igralcev kot pri tradicionalnem pristopu, a ne rešuje problema pri skokovitem povečanju igralcev znotraj posameznega ozvezdja. Eve Online rešuje ta problem na dva načina. Prvi del rešitve je implementacija tehnologije za izjemno učinkovito medprocesno komunikacijo med različnimi strežniki SOL. Ta je trenutno še v razvoju, omogočala pa bi dinamično dodajanje procesorske moči prezasedenim strežnikom SOL s prenosom podatkov o stanju igre in igralcev z enega strežnika na drugega brez posledic na njihovo učinkovitost. Drugi del rešitve pa temelji na posebnem elementu igralnosti upočasnitve igralnega časa. Ta omogoča, da se igralni čas upočasni, s tem pa strežnik SOL prejme manj simulacijskih dogodkov, ki jih uspe simulirati v zadovoljivem realnem času. Rešitev je kljub učinku počasnega posnetka bitke dobra, saj igralcem omogoča izvajanje interakcij v zadovoljivem času brez upada zanesljivosti.

Strežniška arhitektura v Eve Online se od koncepta, razvitega v diplomskem delu, močno razlikuje, saj uporablja pristop enega igralnega strežnika. Medtem ko so nekatere minimalne storitve funkcionalno še vedno ločene, poteka simulacija dogodkov posameznega ozvezdja v celoti na istem strežniku SOL. Prav tako ne prihaja do kloniranja igralnega sveta na različne igralne strežnike in posledične segregacije igralcev. To omogoča poljubno razširljivo strežniško arhitekturo kot celoto (npr. preko dodajanja novih ozvezdij) in implementacijo posebnih elementov igralnosti, ki jih pri tradicionalnem pristopu ni mogoče doseči brez namenskih strežniških rešitev.

Poglavje 6

6 Sklepne ugotovitve

V diplomskem delu je bila predstavljena kompleksnost načrtovanja in implementacije strežniške arhitekture za potrebe masovnih večigralskih spletnih iger. Prikazani so bili konkretni izzivi pri zagotavljanju zanesljivosti, razpoložljivosti in razširljivosti iger. Na podlagi teh izzivov je bil izdelan koncept strežniške arhitekture, ki te izzive rešuje z uporabo pristopa porazdeljenih strežnih storitev. Koncept temelji na funkcionalni delitvi storitev na minimalne in specifične, ki skupaj tvorijo ustrezno celoto za zagotavljanje razvoja elementov igralnosti igre in njihovo simulacijo. Koncept je zaradi vzporednic pri standardnem načrtovanju strežniške arhitekture primerljiv z nekaterimi komercialnimi masovnimi spletnimi igrami.

V diplomskem delu razvit koncept je zaradi osnovne delitve storitev in poenotenja načina razpošiljanja interakcij poljubno razširljiv. Služi kot uspešna podlaga za razvoj dodatnih specifičnih storitev in prilagojenih namenskih elementov igralnosti. Prav tako se neposredno ne nanaša na način simulacije navideznega sveta in načrtovanje objektov igre, kar je bolj specifično vsaki igri posebej. Zaradi fizične ločitve strežnikov in storitev je možna poljubna delitev igralnih storitev na ločene pogone, kar zagotavlja ustrezno razpoložljivost in razširljivost.

Nadgradnje koncepta vključujejo vključitev učinkovitih pristopov k pošiljanju podatkov o interakcijah in učinkovit sistem za izmenjavo objektov igre med strežnikom in odjemalci. Rešitev je enostavno razširljiva za poljuben tip večigralskih iger, posebno učinkovita pa je v primerih, kjer število igralcev presega zmogljivosti enega strežnika.

Masovne večigralske spletne igre ostajajo pomemben žanr na področju računalniških iger, zaradi vse večje popularnosti in zahtev igralcev pa bo v prihodnje zanimivo spremljanje pristopov k reševanju novih izzivov načrtovanja strežniške arhitekture.

Literatura

[1] M. Assiotis in V. Tzanov. (2005, May) A Distributed Architecture for Massive Multiplayer Online. Dostopno na: http://pdos.csail.mit.edu/6.824-2005/reports/assiotis.pdf

[2] BBC. (2013, Jul.) Eve players stage giant online space battle. Dostopno na:. http://www.bbc.co.uk/news/technology-23489293

[3] J. Harrison. (2010, Jan.) EVE Scalability Explained. Dostopno na: http://www.talkunafraid.co.uk/2010/01/eve-scalability-explained/

[4] K. V. Jónsson, The Server Technology Of Eve Online: How To Cope With 300,000 Players In One World. Shanghai, China: Game Developers Conference China, 2010.

[5] C. Khaw. (2013, Jun.) Tranquility, the military-grade 2,500GHZ monster that powers Eve Online. Dostopno na: http://www.pcgamer.com/2013/06/15/eve-online/

[6] F. Lu, Load Balancing for Massively Multiplayer Online Games . Newcastle, UK, 2006

[7] (2014) Masovna večigralska spletna igra Star Wars: The Old Republic. Dostopno na: http://en.wikipedia.org/wiki/Star_Wars:_The_Old_Republic

[8] R. Miller. (2009) WoW's Back End: 10 Data Centers, 75,000 Cores. Dostopno na: http://www.datacenterknowledge.com/archives/2009/11/25/wows-back-end-10-data-centers-75000-cores/

[9] (2014) MMOData. Dostopno na: http://users.telenet.be/mmodata/Charts/PCU-1.png

[10] O. Petitte. (202) MMO infographic. Dostopno na: http://www.pcgamer.com/2012/12/15/mmo-infographic/

[11] A. Thor, Massively Multiplayer Game Development 2. Boston, USA, 2005

[12] (2014) Valve Source Multiplayer Networking. Dostopno na: https://developer.valvesoftware.com/wiki/Source_Multiplayer_Networking