

Osnove virtualnih okruženja

Umrežena virtualna okruženja

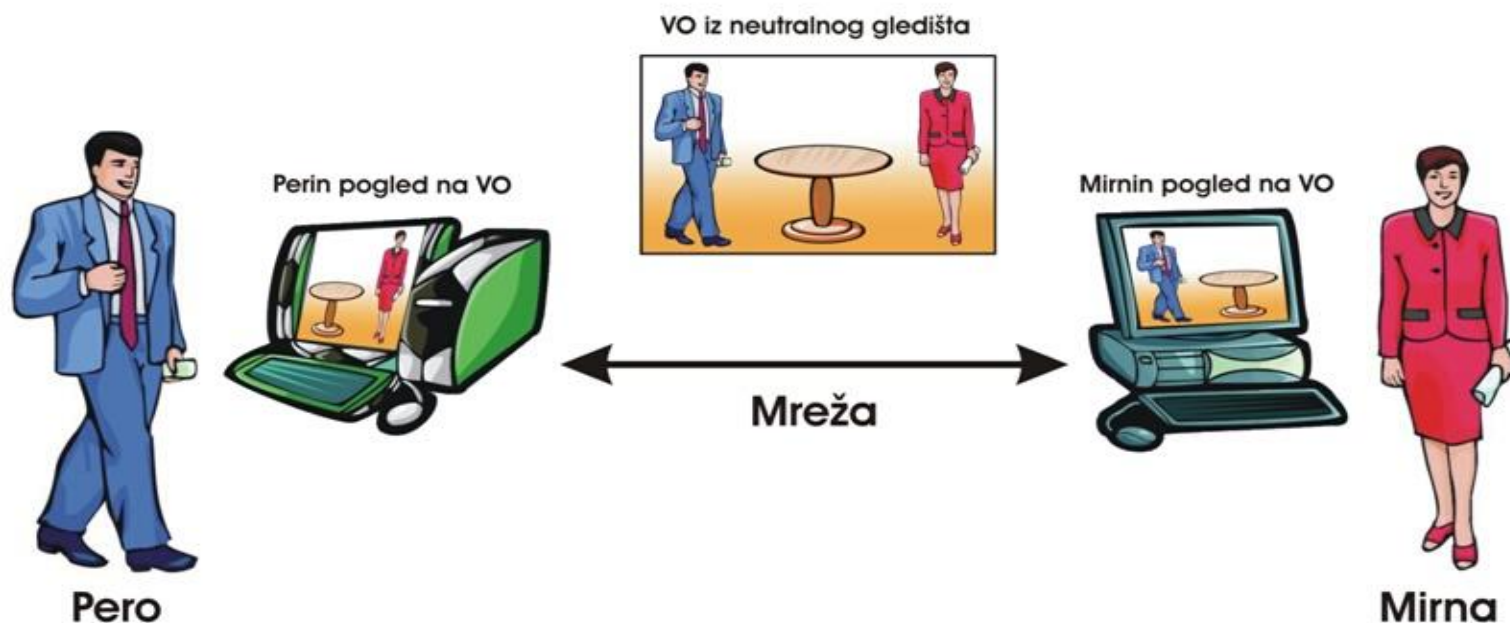
Prof. dr. sc. Igor S. Pandžić

Prof. dr. sc. Maja Matijašević

dr. sc. Mirko Sužnjević



- ◆ Uvod
 - Definicija
 - Tehnički izazovi
 - Osnovni model UVO
- ◆ Upravljanje zajedničkim dinamičkim stanjem
- ◆ Oblikovanje programskog rješenja UVO
- ◆ Virtualni svjetovi
 - Mrežni zahtjevi
 - Poslovni modeli
- ◆ Umrežene igre

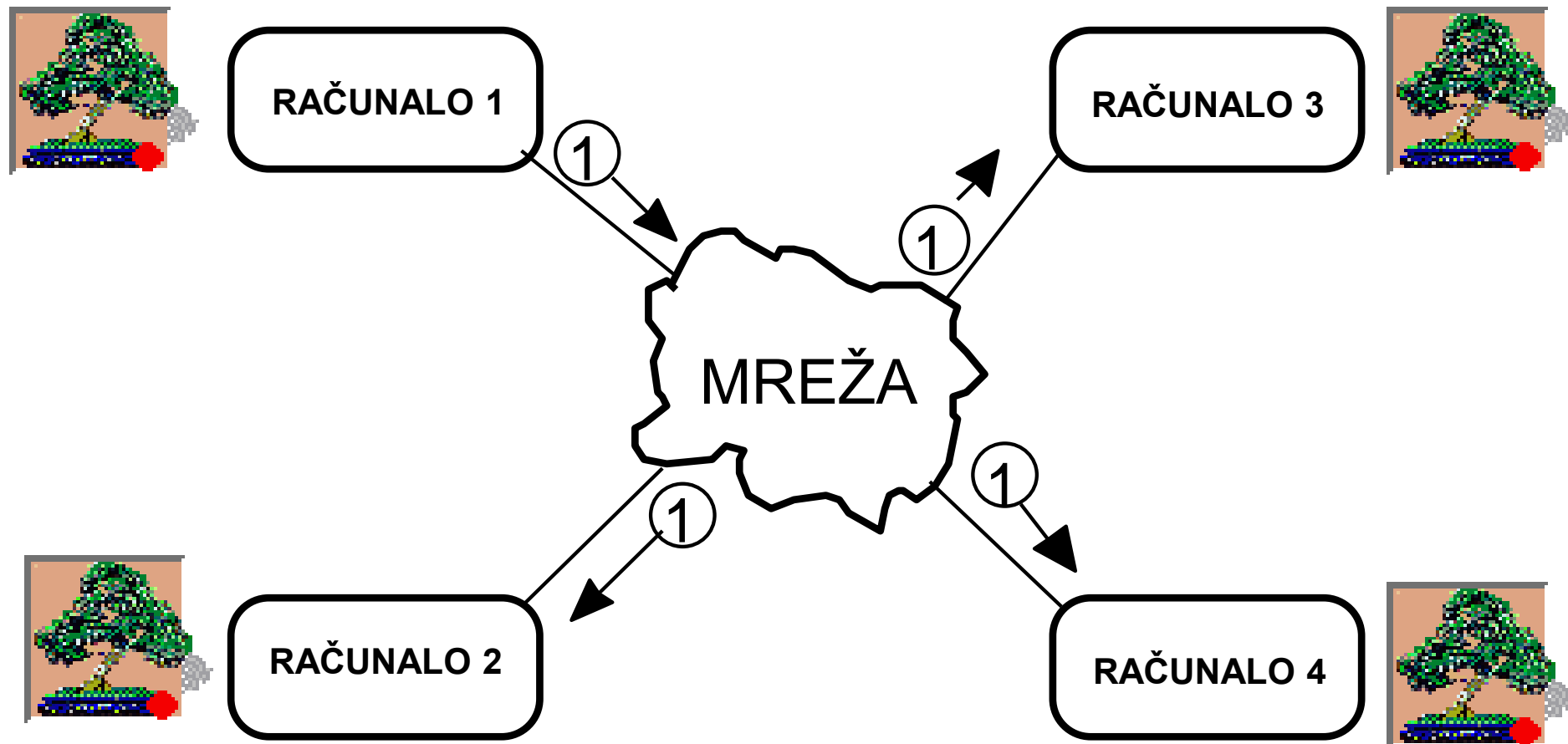


- ♦ Fizički udaljeni korisnici sudjeluju u zajedničkom virtualnom okruženju
- ♦ Svako računalo ima lokalnu kopiju okruženja
- ♦ Svaki korisnik upravlja svojim 3D prikazom i okruženjem
- ♦ Sve kopije okruženja se međusobno sinkroniziraju putem mreže
- ♦ Korisnici vide jedni druge jer su grafički prikazani u okruženju

- ◆ Doživljaj zajedničkog prostora
 - Korisnici osjećaju da su u istom (stvarnom ili zamišljenom) prostoru
- ◆ Doživljaj zajedničkog prisustva
 - Svaki od korisnika ima svoju reprezentaciju unutar virtualnog svijeta
- ◆ Doživljaj zajedničkog vremena
 - Događaji (izgledaju kao da) se događaju u isto vrijeme
- ◆ Mogućnost komunikacije
- ◆ Mogućnost interakcije
 - S virtualnim svijetom i drugim korisnicima



Kako radi umreženo virtualno okruženje



- ♦ Virtualni svjetovi – popularni u 2000-tima, danas ih ponovno popularizira virtualna stvarnost
 - Druženje i društvene mreže (Facebook Horizon - <https://www.oculus.com/facebookhorizon/>)
 - Virtualna telekonferencija/zajednički rad (Terf - <https://www.3dicc.com/>)
 - Učenje/obuka na daljinu (Second Life - <https://secondlife.com/>)
 - Virtualni svjetovi za djecu (Club Penguin - ugašen)
 - Minecraft – igra ili virtualni svijet?
 - ...
- ♦ Umrežene višekorisničke igre – primarna svrha je zabava



UMREŽENA VIRTUALNA OKRUŽENJA

- ◆ Prikaz korisnika
- ◆ Interakcija korisnika u virtualnom okruženju
- ◆ Podrška za prirodnu komunikaciju
- ◆ Karakteristike tehnologije i načina umrežavanja
- ◆ Prilagodljivost veličini (engl. *scalability*)



- ◆ Lik korisnika unutar virtualnog okruženja se obično naziva avatar
- ◆ Od riječi avatāra iz sanskrita koja znači inkarnaciju ili manifestaciju nekog od bogova u tijelu čovjeka ili životinje
- ◆ Prvi put avatar se koristi u ovom značenju 1985 u igri *Ultima IV: Quest of the Avatar*
- ◆ Oblici avatara variraju od jednostavnih geometrijskih do složenih humanoidnih avatara visokih detalja
- ◆ Karakteristike avatara često definiraju i mogućnosti interakcije kao i posredno na količinu podataka koje se prenose mrežom
- ◆ Današnji avatari najčešće imaju niz gesti i izraza lica kroz koje korisnici mogu komunicirati s drugima (primjerice ples)



- ◆ Načini interakcije zadani karakterom aplikacije
- ◆ Učestali načini interakcije
 - Kretanje
 - Geste
 - Stvaranje zvukova
 - Interakcije s predmetima
 - Interakcije s drugim korisnicima
- ◆ Mehanizmi za upravljanje pravom pristupa – ako više korisnika pokušava udariti virtualnu loptu kako odrediti tko je uspio?
- ◆ Vremenski okvir interakcije?

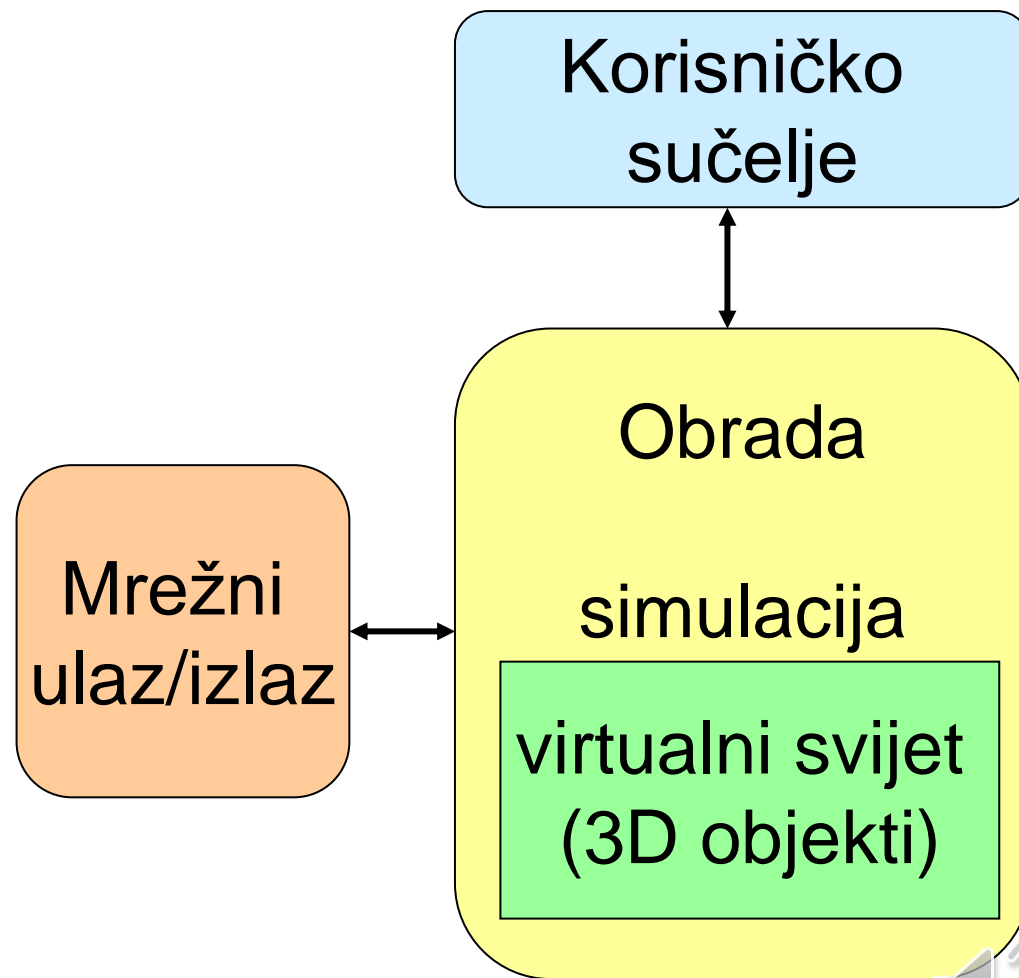


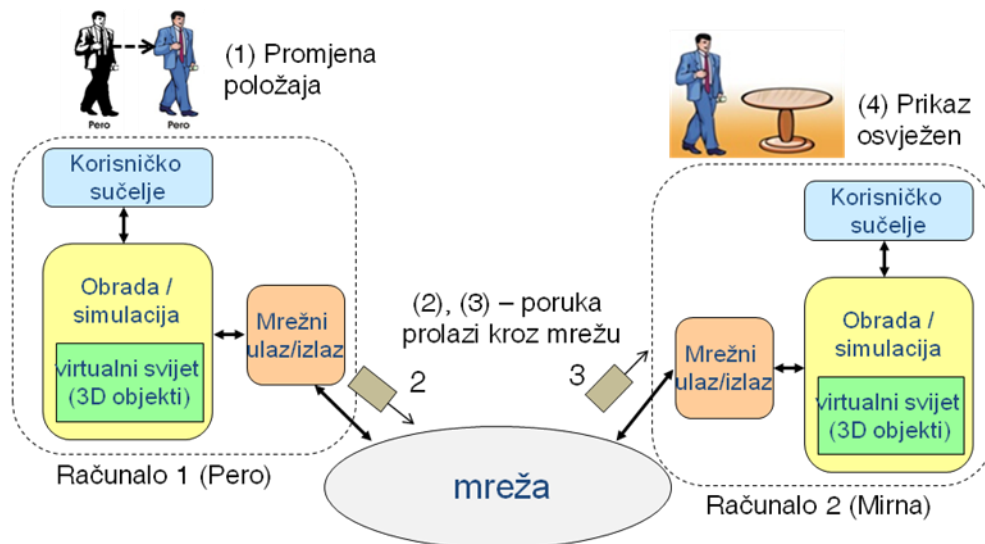
- ◆ Većina UVO-a podržava pismenu komunikaciju
- ◆ Komunikacija putem gesti je također vrlo česta (komande avataru)
 - Mahanje
 - Ples
 - Potvrda i negacija
 - Virtualna stvarnost – geste iz stvarnog svijeta (Facebook Horizon)
- ◆ U posljednje vrijeme UVO-a također donose ugrađenu podršku za glasovnu komunikaciju, a ista može biti i ograničena pozicijom avatara unutar UVO (ukoliko su avatari dva korisnika predaleko neće se čuti)

Karakteristike tehnologije i načina umrežavanja

- ◆ Korisnici istog UVO-a mogu koristiti pristupne uređaje i mreže heterogenih karakteristika
- ◆ Pristupni uređaji imaju definirane minimalne uvjete (minimalnu konfiguraciju) koja podržava pristup određenom UVO
- ◆ Karakteristike mrežnih parametara variraju ovisno o tipu UVO te o njegovim karakteristikama
- ◆ „Skrivanje” mreže i njenih utjecaja je najveći tehnički izazov UVO-a
- ◆ Najvažniji mrežni parametri koji utječu na UVO:
 - Propusnost
 - Kašnjenje
 - Kolebanje kašnjenja
 - Gubitak paketa

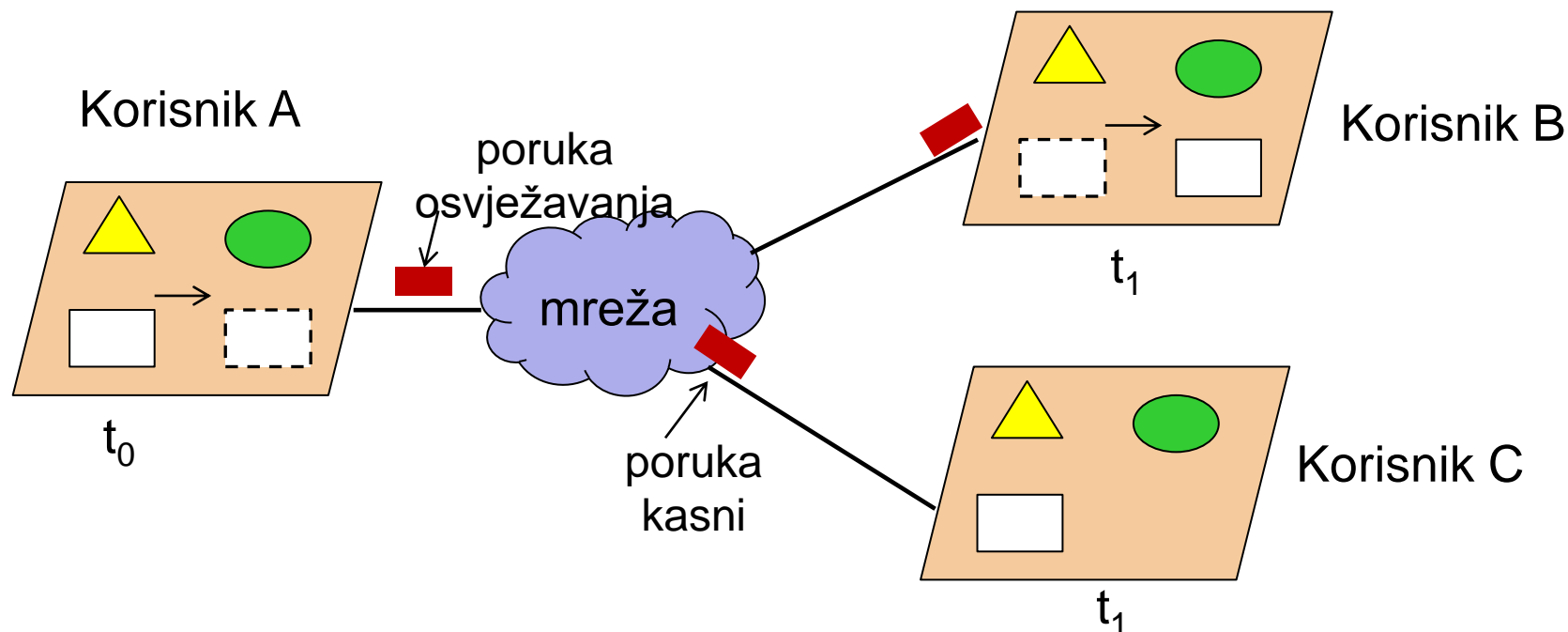
- ♦ Korisničko sučelje, ulazni i izlazni uređaji
- ♦ Skup 3D geometrijskih objekata (oblik, boja i sl.), i drugih komponenata (npr. svjetlo, zvuk) u računalu
- ♦ Obrada – sadrži simulaciju vezana uz primjenu (početne postavke, simulacijska petlja), obrada 3D grafike i zvuka
- ♦ Mrežna povezanost





- ◆ Svaki korisnik ima pokrenutu instancu VO
- ◆ Ne moramo slati elemente koji se ne mijenjaju (npr. teksture, gotovi 3D objekti)
- ◆ Problemi
 - Promet raste s brojem korisnika – skalabilnost
 - Kašnjenje nije isto za sve korisnike - nekonzistentnost



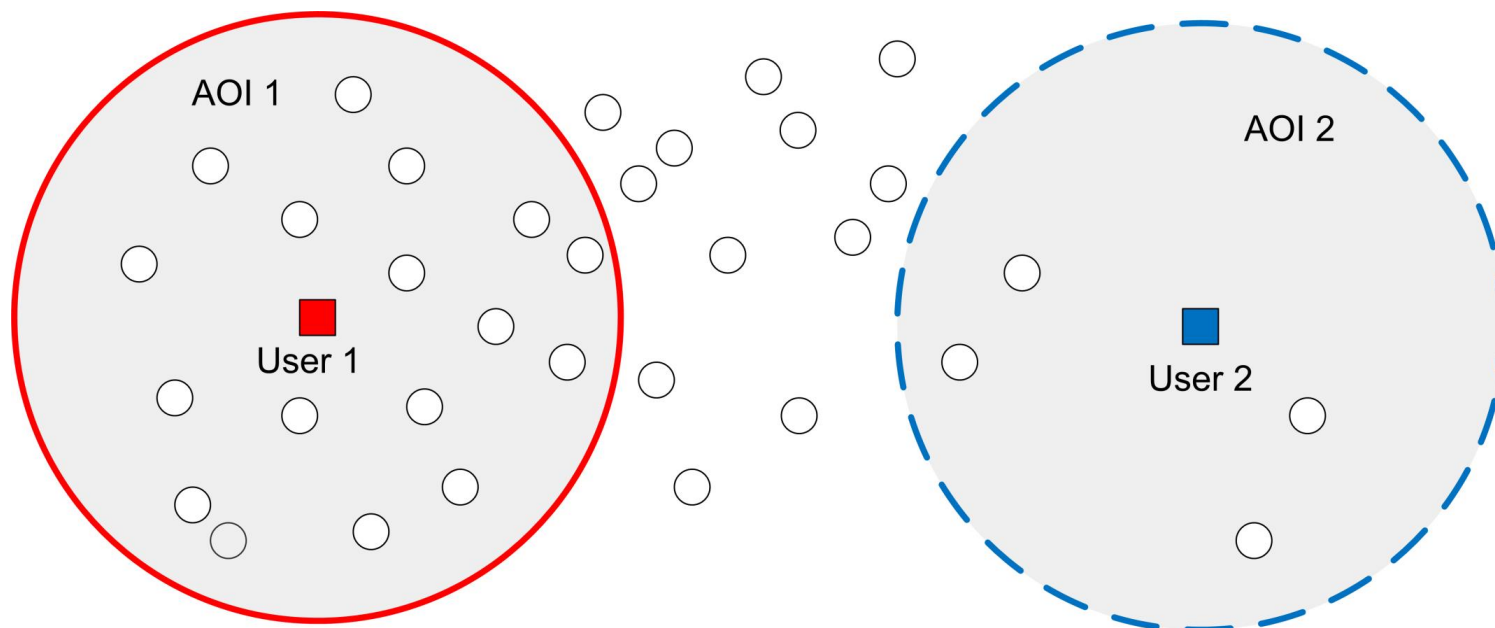


- ♦ Ključni izazov: održavanje raspodijeljenog zajedničkog stanja konzistentnim, ili približno konzistentnim, uz promjene koje nastaju (asinkrono, dinamički i raspodijeljeno)
- ♦ Zahtjeve na konzistentnost treba uskladiti s namjenom aplikacije
- ♦ Češće slanje poruka osvježavanja skraćuje potencijalni vremenski interval u kojem može doći do odstupanja, ali više opterećuje mrežu i proces za slanje i primanje poruka

- ♦ Skalabilnost – koliko neki sustav može rasti, a da se pri tom ne naruši njegova funkcija
- ♦ UVO-a mogu biti vrlo popularna – mehanizmi za osiguranje skalabilnost su vrlo važni
 - Second Life – na vrhuncu popularnosti imao 1 milijun korisnika
 - World of Warcraft – na vrhuncu popularnosti 12 milijuna korisnika
- ♦ Parametri koji utječu na skalabilnost:
 - Složenost grafičkog prikaza svijeta i avatara
 - Količina podataka koju generira svijet
 - Obrada ulaza i izlaza
 - Odražavanje dinamičkog zajedničkog stanja

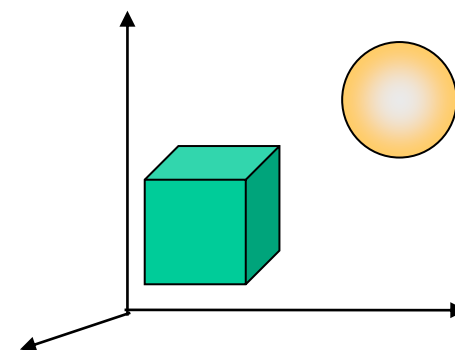


- ♦ engl. Area Of Interest Management, AOIM
- ♦ Prosljeđuju se samo relevantne poruke
- ♦ Umjesto eksponencijalnog rasta prometa s brojem korisnika, linearan rast: ključan preduvjet za postojanje modernih UVO s desecima tisuća istovremenih korisnika



Upravljanje dinamičkim zajedničkim stanjem

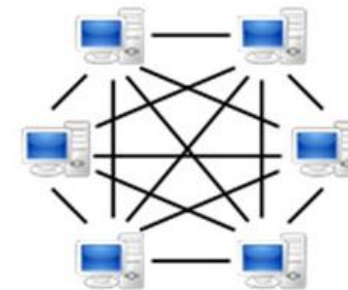
- ♦ Zahtjev: održati konzistentno stanje (isto stanje na svim replikama virtualnog svijeta)
- ♦ Zajedničko stanje = skup varijabli stanja svih pojedinačnih objekata u UVO
 - varijable za jedan objekt: položaj, orijentacija, brzina, izgled, itd.
- ♦ Osnovni problemi:
 - odnos konzistentnost – propusnost*
 - odnos konzistentnost – mrežno kašnjenje*



Rješenja problema upravljanja zajedničkim stanjem

♦ Na razini arhitekture

- Centralizirani repozitorij informacija
- Raspodijeljeni (distribuirani) repozitorij informacija



♦ Na razini protokola

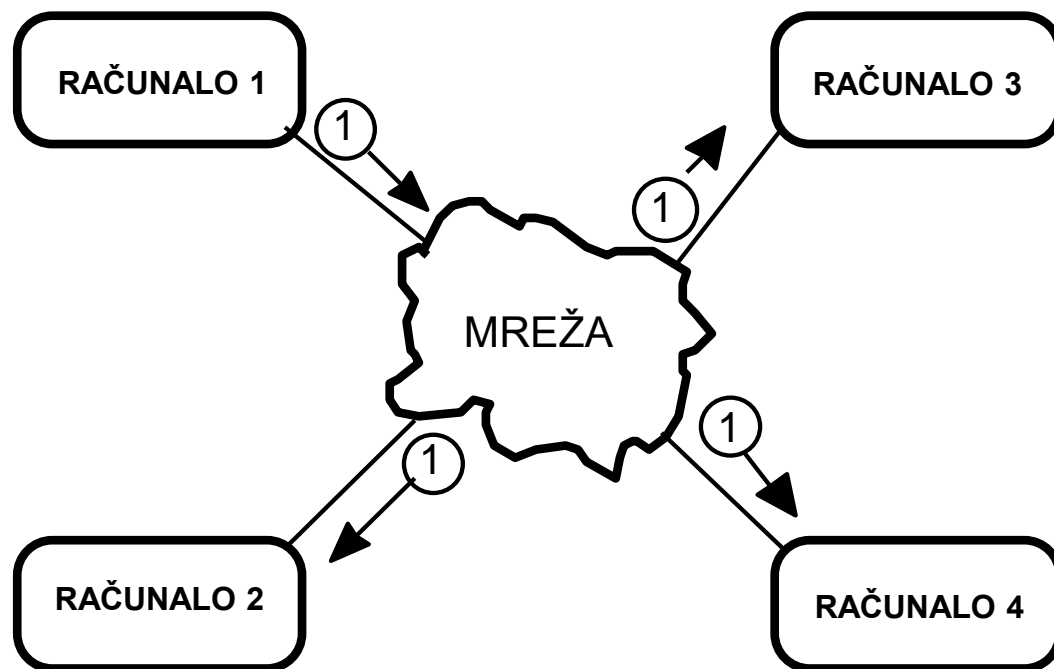
- Jaka konzistentnost - svakom trenutku stanje svake replike UVO-a je istovjetno
 - Koristi se protokol koji osigurava konzistentnost u distribuiranom okruženju
 - Protokol konzistentnosti je na aplikacijskoj razini
- Slaba konzistentnost – stanje svake replike UVO-a može biti različito u istom trenutku
 - Razlika se pokušava minimizirati kako bi njen utjecaj bio neprimjetan
 - Koristi se učestala regeneracija (slanje) stanja
 - Koriste se tehnike predviđanja i *dead reckoning*

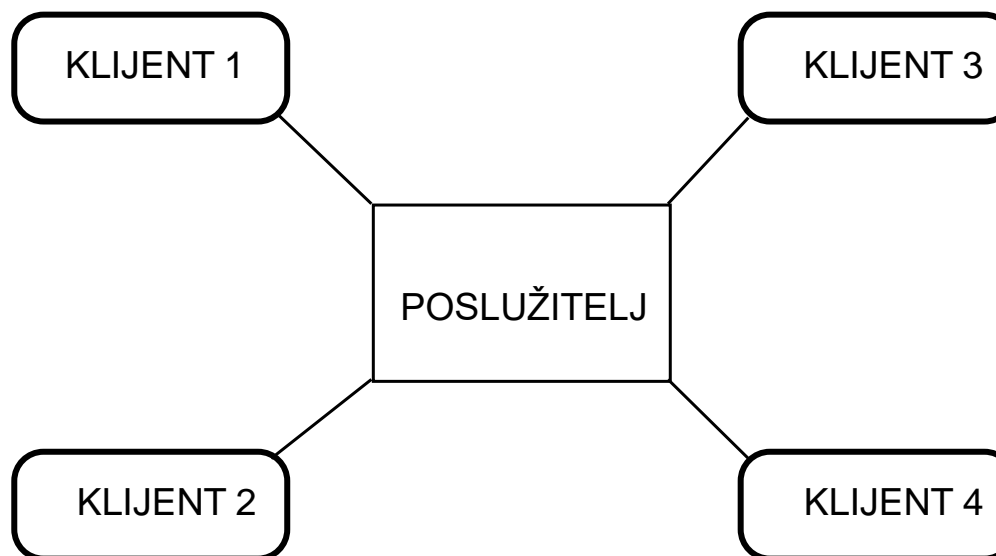
♦ Zahtjevi:

- Učinkovit prijenos poruka
- Podrška za AOIM
- Upravljanje sjednicom
- Trajni zapis stanja
- Kontrola pristupa, naplata

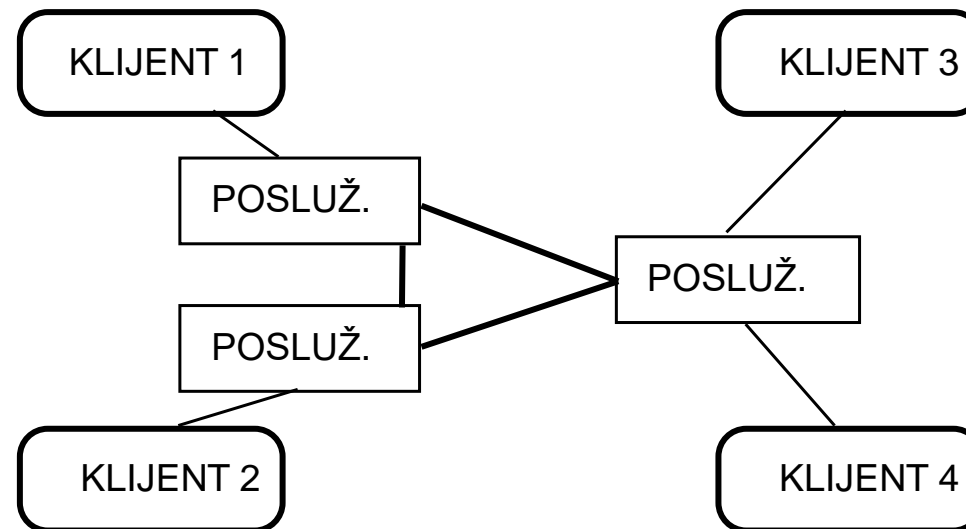
♦ Rješenja:

- Klijent/poslužitelj
- Više poslužitelja
- Ravnopravni procesi (engl. peer-to-peer)

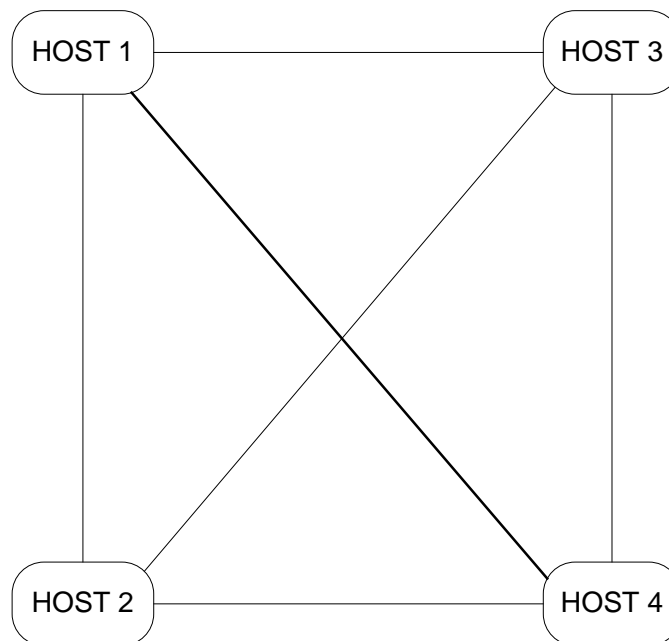




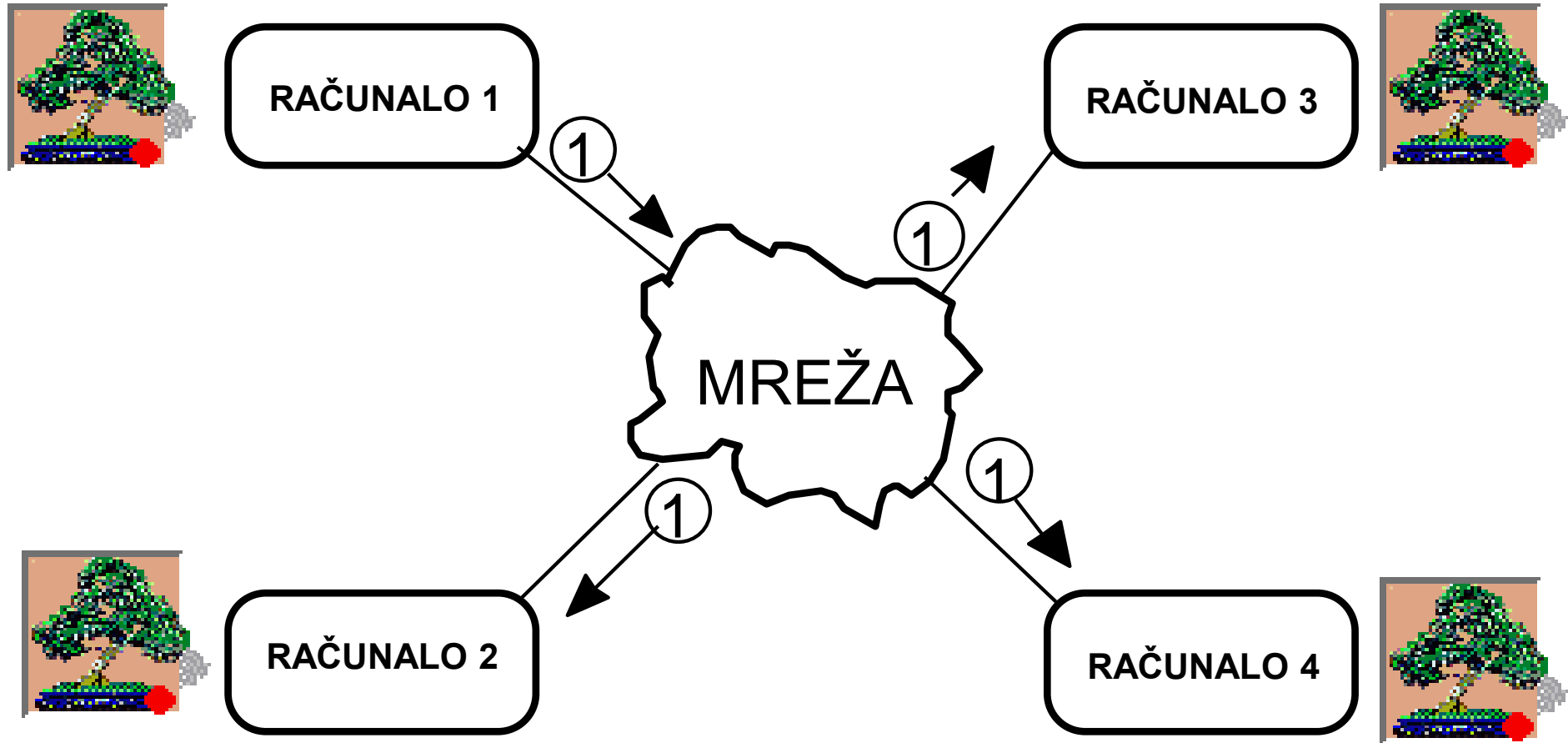
- ♦ Vrlo učinkovita arhitektura
- ♦ Filtriranje prometa, kontrola sjednice, trajni zapis, kontrola pristupa i naplata izvedeni na poslužitelju
- ♦ Poslužitelj usko grlo: ograničena prilagodljivost veličini

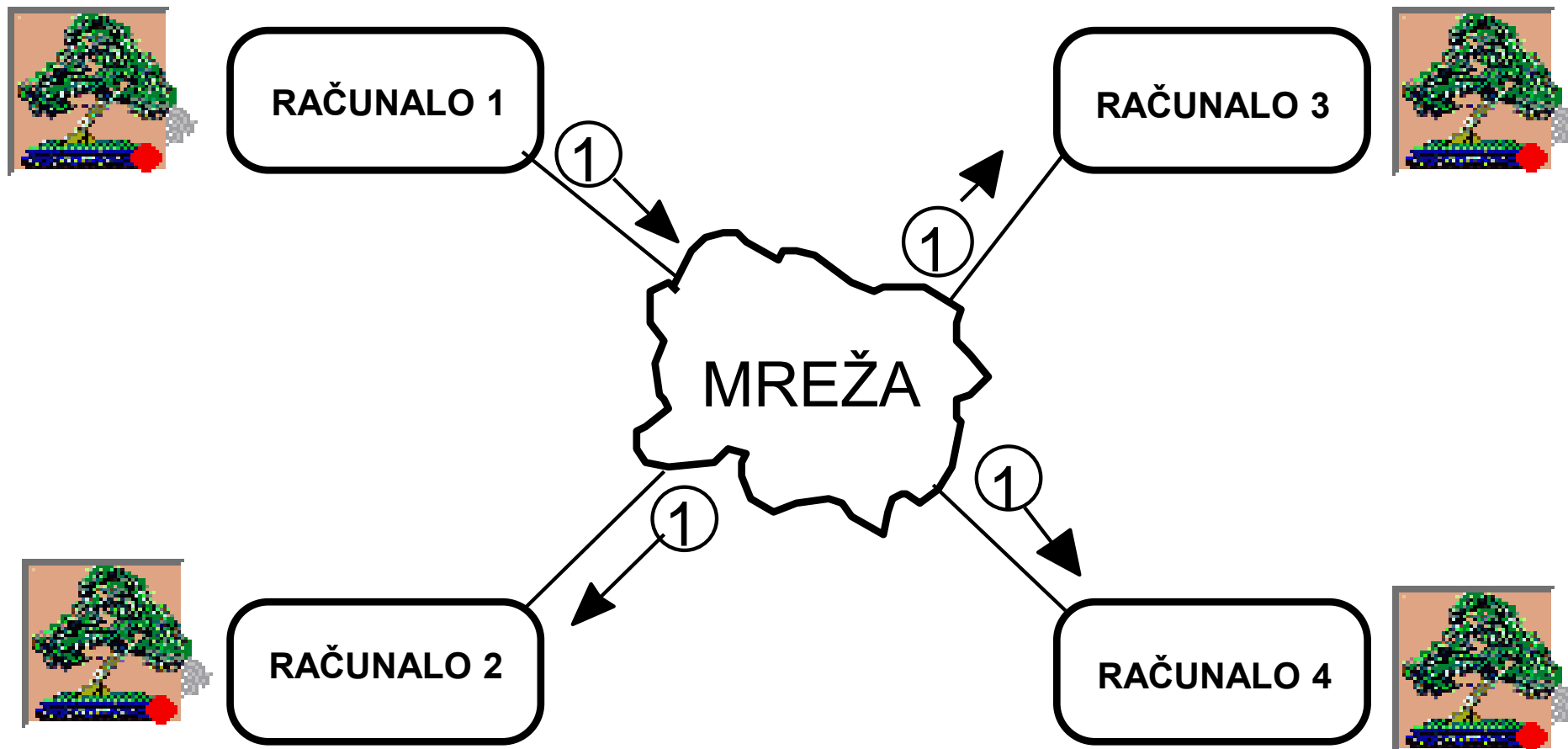


- ♦ Većina dobrih strana arhitekture klijent/poslužitelj
- ♦ Povećava se kašnjenje
 - potreba za vrlo brzom vezom između poslužitelja
- ♦ Izvedba relativno složena



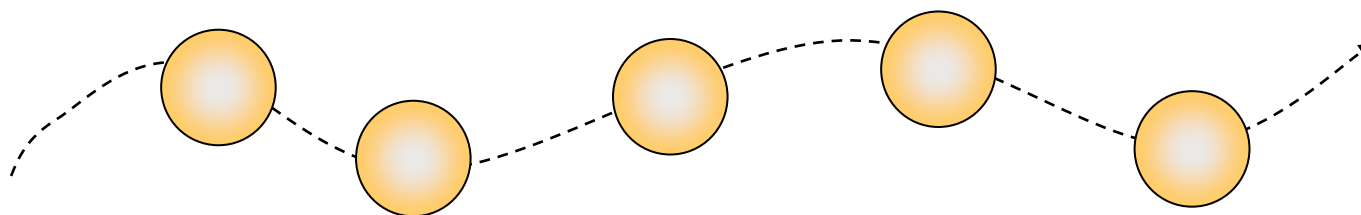
- ◆ Direktna komunikacija između svih čvorova
- ◆ Prvi sustavi UVO (npr. Doom) koristili su ovo rješenje
- ◆ Problemi: trajni zapis, kontrola pristupa, AOIM, upravljanje sjednicom, kontrola varanja, migriranje informacija u slučaju odspajanja jednog korisnika





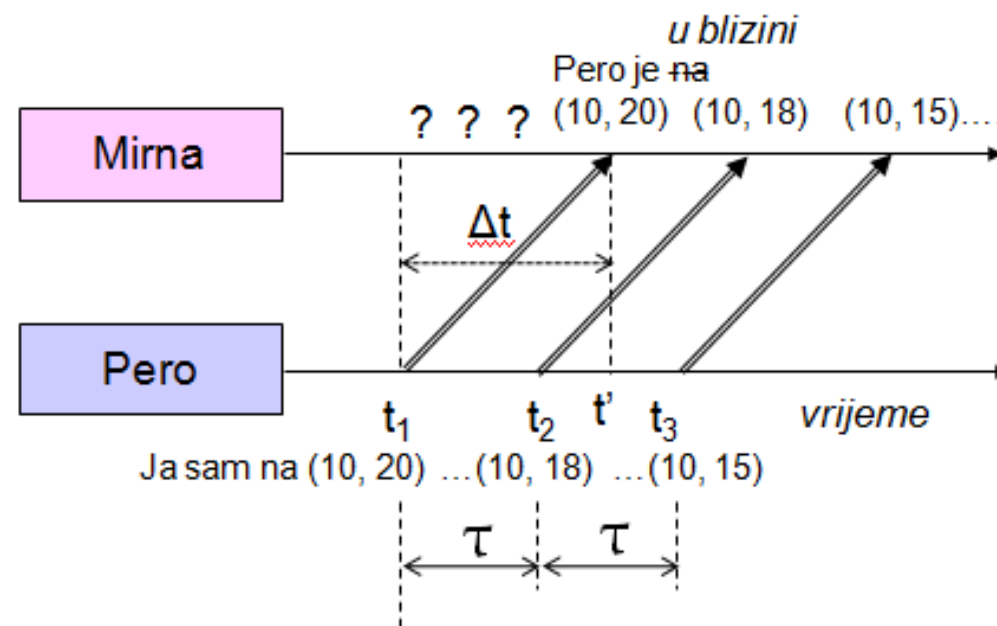
- ◆ Jaka konzistentnost (zajednički podaci)
 - Ograničena kašnjenjem najsporijeg klijenta – informacija o promjeni mora doći i do tog klijenta
 - Ograničena nivoom interaktivnosti samog UVO-a
 - Primjerice Time To Kill nekih oružja u Call of Duty franšizi je 200 ms - nemoguće omogućiti takvu interaktivnost u sustavu s jakom konzistentnosti
- ◆ Slaba konzistentnost (replicirani podaci)
 - Koristi učestala osvježanja – ograničena propusnošću
 - Ograničena arhitekturom (skalabilnošću) – kako primiti i poslati osvježanja za veliki broj korisnika u slučaju P2P arhitekture?

- ♦ Vlasnik entiteta razašilje poruke osvježavanja u pravilnim vremenskim razmacima, koristeći za dostavu nepouzdan protokol



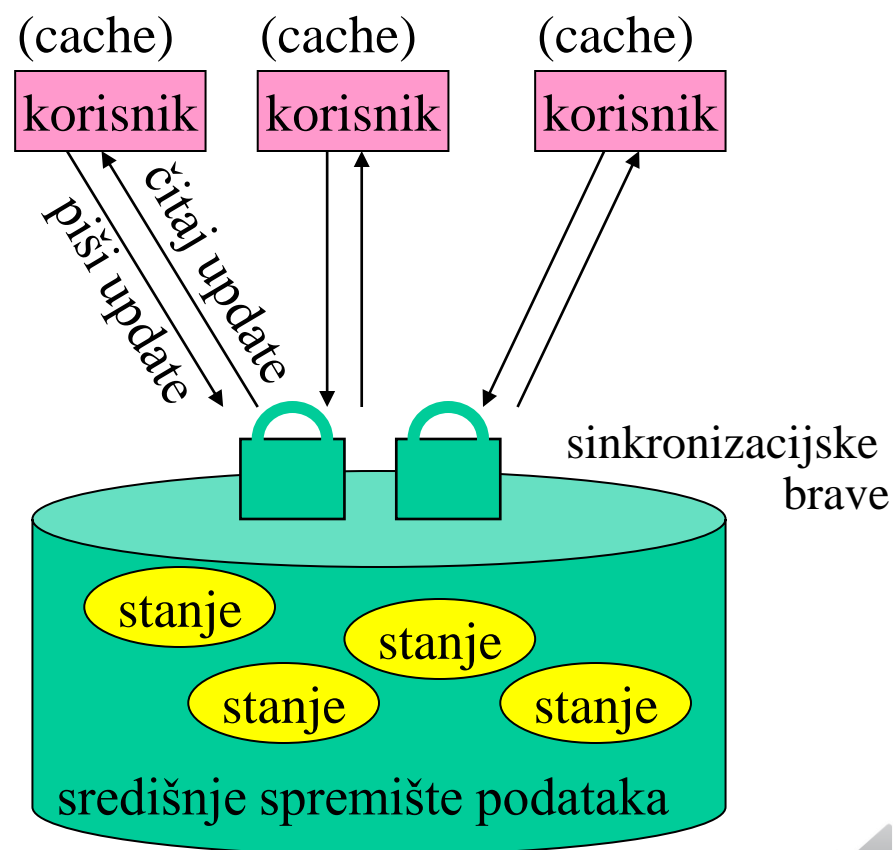
- ♦ Svaka poruka osvježavanja sadrži puni opis stanja entiteta (npr. položaj, orijentaciju, izgled itd.)
- ♦ Svaki primatelj ima lokalni spremnik
- ♦ Primatelji ne potvrđuju primitak poruka, a izgubljene poruke osvježavanja se nadomještaju novijima [primjer 1]
- ♦ Nema očuvanja globalnog redoslijeda poruka osvježavanja

- ◆ Primjer pokazuje nekonzistentnost uzrokovanu kašnjenjem
- ◆ Za maksimalnu konzistentnost, Pero bi trebao čekati da Mirna potvrdi primitak, pa tek onda slati iduću poruku: gubitak propusnosti i interaktivnosti
- ◆ Nemoguće je postići visoku dinamičnost u okruženju i visoku konzistenciju



Središnji repozitorij informacija s zajedničkim podacima

- ♦ Klijent – poslužitelj uz jaku konzistenost
- ♦ Središnji repozitorij sadrži sve podatke o zajedničkom stanju UVO
- ♦ Svi čvorovi u svakom trenutku imaju identični pogled na zajedničko stanje
- ♦ Središnji repozitorij nadzire čitanje i pisanje stanja, kao i poredak pri osvježavanju stanja



Prednosti i nedostaci središnjeg repozitorija s zajedničkim podacima

♦ Prednosti

- Jednostavan programski model
- Garantirana konzistentnost
- Nema vlasništva nad podacima, središnji poslužitelj se brine o osvježavanju stanja i redoslijedu

♦ Nedostaci:

- Nepredvidivost što se tiče trajanja i čekanja na pristup podacima i osvježavanje
- Zamjetna količina dodatne obrade (*overhead*) – pouzdani komunikacijski protokol i/ili učestalo slanje stanja od strane poslužitelja

Raspodijeljeni repozitorij s zajedničkim podacima

- ◆ Peer to peer uz jaku konzistentnost
- ◆ Očuvanje konzistentnosti se sada prebacuje na protokol za očuvanje
- ◆ Sve replike su uvijek u istom stanju te imaju kompletnu informaciju o cijelom virtualnom svijetu
- ◆ Primjer korištenja: strategije u stvarnom vremenu (engl. Real Time Strategies) poput Starcraft 2
- ◆ Prednosti: bolja otpornost sustava, nema kašnjenja uzrokovanih komunikacijom sa središnjim sustavom, omogućuje dobru skalabilnost s obzirom na mrežni promet (deterministička simulacija)
- ◆ Mane: najsporiji klijent usporava sustav, ne može se koristiti za sustave s visokom razinom interakcije

- ◆ Klijent – poslužitelj sa slabom konzistentnošću
- ◆ Svaki klijent posjeduje repliku stanja UVO-a (najčešće samo lokalnu repliku određenu područjem interesa)
- ◆ Poslužitelj posjeduje centralnu kopiju koja ima „stvarno” stanje prema kojem se osvježavaju replike na klijentima
- ◆ Poslužitelj sljedeće dvije funkcije:
 - Osigurava jednoznačni poredak poruka osvježavanja stanja
 - Razašilje novo stanje svim zainteresiranim klijentima
- ◆ Najčešća varijanta u današnjim sustavima

Raspodijeljeni repozitorij s raspodijeljenim podacima

- ◆ Peer to peer s slabom konzistentnošću
- ◆ Ne postoji kompletna „točna” replika
- ◆ Moguća i parcijalna replikacija virtualnog svijeta – kombinirani podaci sa svih replika daju ukupnu sliku svijeta
- ◆ Za neke primjene dodatna obrada kao kod središnjeg repozitorija može biti neprihvatljiva, ili pak ne trebaju apsolutnu konzistentnost
 - primjer: simulator leta – “glatki” pomak, odnosno prikaz promjene položaja bez trzaja ili zamrzavanja slike, je važniji od apsolutne točnosti položaja u svakom trenutku

Primjer 1: učestalost osvježavanja

Pretpostavimo da izvor šalje 25 poruka osvježavanja svake sekunde. Na temelju toga, prikaz se može osvježiti svakih:

$$1/25 = 0.04 \text{ s} = 40 \text{ ms}$$

Jedna izgubljena poruka utječe na prikaz u trajanju od 40 ms, što za najveći broj primjena, ljudima nije uočljivo.

Dizajner aplikacije može pretpostaviti da će visoka učestalost poruka osvježavanja učiniti male nekonzistencije nevidljivima za korisnike.

Primjeri aplikacija koje koriste učestalo osvježavanje:

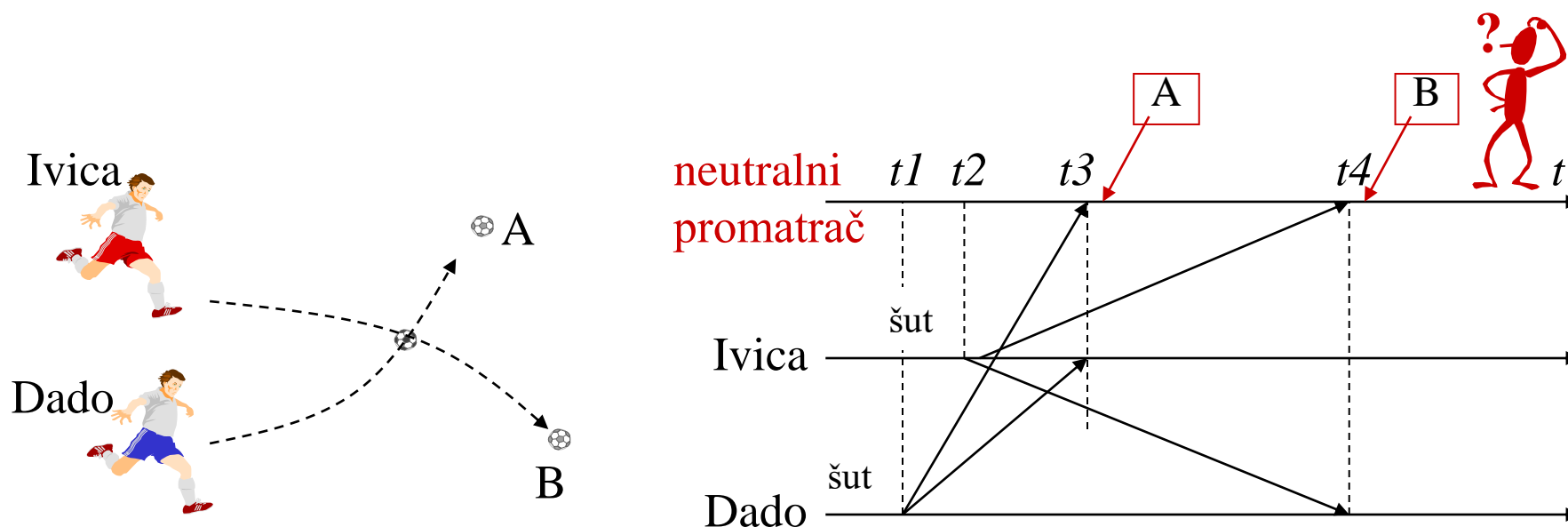
- Silicon Graphics Dogfight simulator leta
- Višekorisničke igre na PC-u: Overwatch, Battlefield, World of Warcraft



- ◆ Kada nema potpune konzistentnosti dolazi do problema vlasništva nad objektima
- ◆ Treba spriječiti da više igrača istovremeno mijenja stanje entiteta [primjer 2, primjer 3]
 - Na primjer, promjena položaja je pisanje varijable stanja položaja entiteta (x, y, z)
- ◆ Uvodi se eksplicitno vlasništvo nad entitetom
 - Tipičan primjer je korisnikov *avatar*, čiji je vlasnik korisnik koji njime upravlja
 - Drugim objektima upravlja poslužitelj *lock manager*
- ◆ *Lock manager* osigurava da svaki entitet u zadanom trenutku ima samo jednog vlasnika

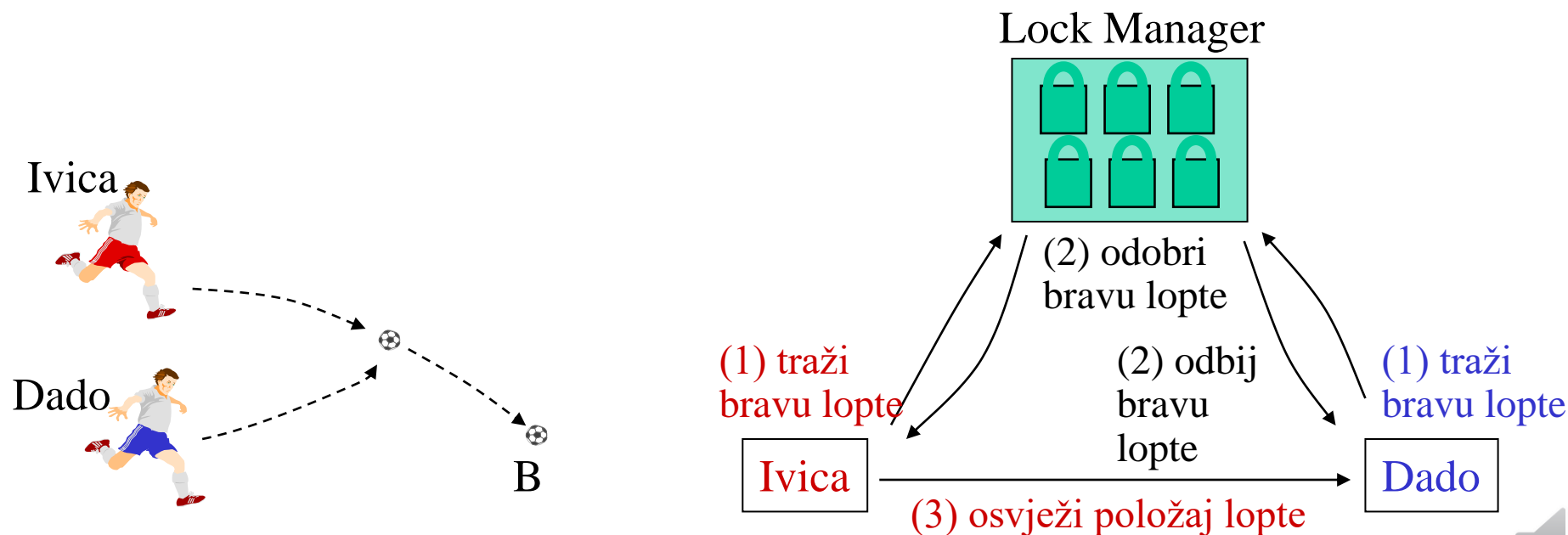
Primjer 2: nema vlasništva

Ivica i Dado igraju nogomet i pokušavaju istovremeni šutnuti loptu. Svaki host šalje poruku osvježavanja stanja lopte (položaj). Neki promatrači mogu prvo primiti Ivičinu poruku, a drugi Dadinu. Osim ako se Ivica i Dado slože oko toga tko je zapravo šutnuo loptu, svaki će nastavljati osvježavati položaj neovisno o drugome. Promatrači će vidjeti titranje lopte između novih položaja ovisno o pristizanju poruka osvježavanja od Ivice i Dade.



Primjer 3: eksplicitno vlasništvo

Ivica i Dado izdaju zahtjev za vlasništvom poslužitelju *lock manager* prije osvježavanja stanja lopte. Ovisno o politici upravljanja bravama (npr. FIFO, round-robin, prioriteti, itd.), *lock manager* daje bravu jednom od korisnika, u ovom primjeru, Ivici. Ivica onda može mijenjati položaj lopte dok god je u vlasništvu brave.



Oblikovanje programskog rješenja UVO

- ◆ Strukturiranje virtualnog prostora
- ◆ Arhitektura raspodijeljene aplikacije UVO
- ◆ Vrste mrežnog prometa i komunikacijski protokoli
- ◆ Način distribucije poruka
- ◆ Izvedba simulacijske petlje UVO
- ◆ Primjeri sustava za razvoj UVO

- ◆ Podijeliti okruženje na jedinice prihvatljive po:
 - broju korisnika u svakoj jedinici
 - složenosti geometrije svake jedinice
 - preciznosti koordinata
- ◆ Primjer: World of Warcraft
 - Korisnici podijeljeni na veći broj poslužitelja, po nekoliko tisuća korisnika po poslužitelju (engl. žargon *shard*)
 - Nema interakcije s korisnicima na drugom poslužitelju
- ◆ Primjer: Second Life
 - Virtualni svijet podijeljen u ćelije 256 x 256 m
 - Za svaku ćeliju odgovoran zaseban poslužitelj

Učitavanje
Poruke sustava
Događaji
Osvježavanje stanja
Tekst
Zvuk
Video

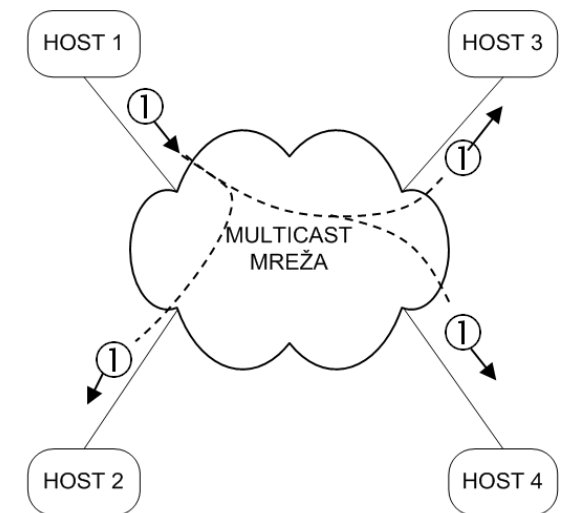
{ 3D predmeti
Teksture (slike)
Modeli ponašanja
Modeli prikaza korisnika

{ Predmeti { Animacija
Deformacija
Korisnici { Položaj tijela
Izraz lica
Kretanje

Mrežni zahtjevi i protokoli za pojedine tipove podataka

Vrsta prometa	Količina podataka	Potrebna pouzdanost	Protokol
Učitavanje			TCP
Poruke sustava			TCP
Događaji			TCP
Osvježavanje stanja			UDP
Tekst			TCP
Zvuk			UDP (RTP)
Video			UDP (RTP)

- ◆ Jednoodredišna komunikacija
 - Najčešći način povezivanja
 - Neučinkovitost se u praksi kompenzira izvedbom AOIM na poslužitelju
- ◆ Višeodredišno razašiljanje
 - Učinkovitije, ali složenija izvedba

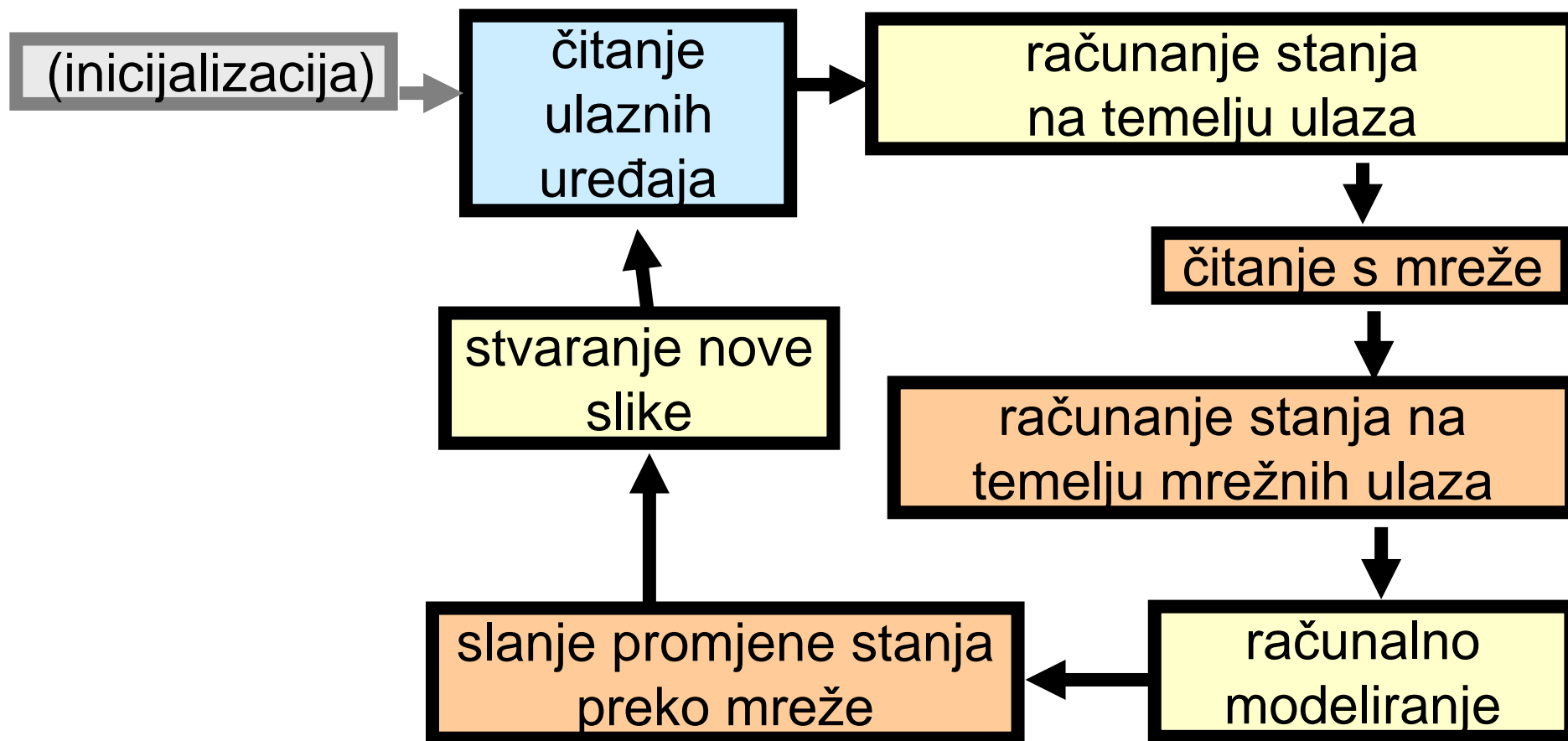


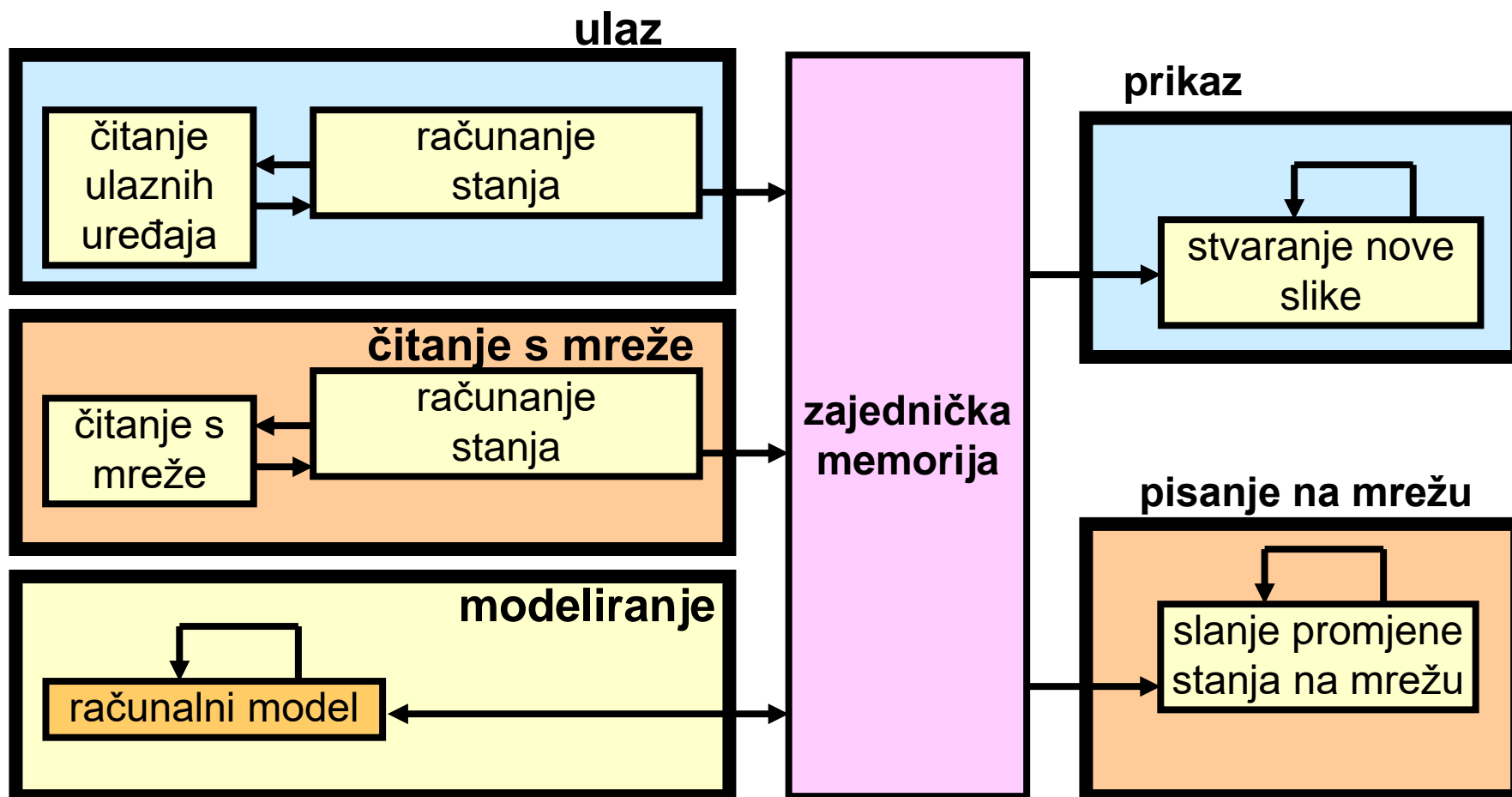
♦ Zahtjev:

- Omogućiti nesmetano i efikasno izvođenje većeg broja modula

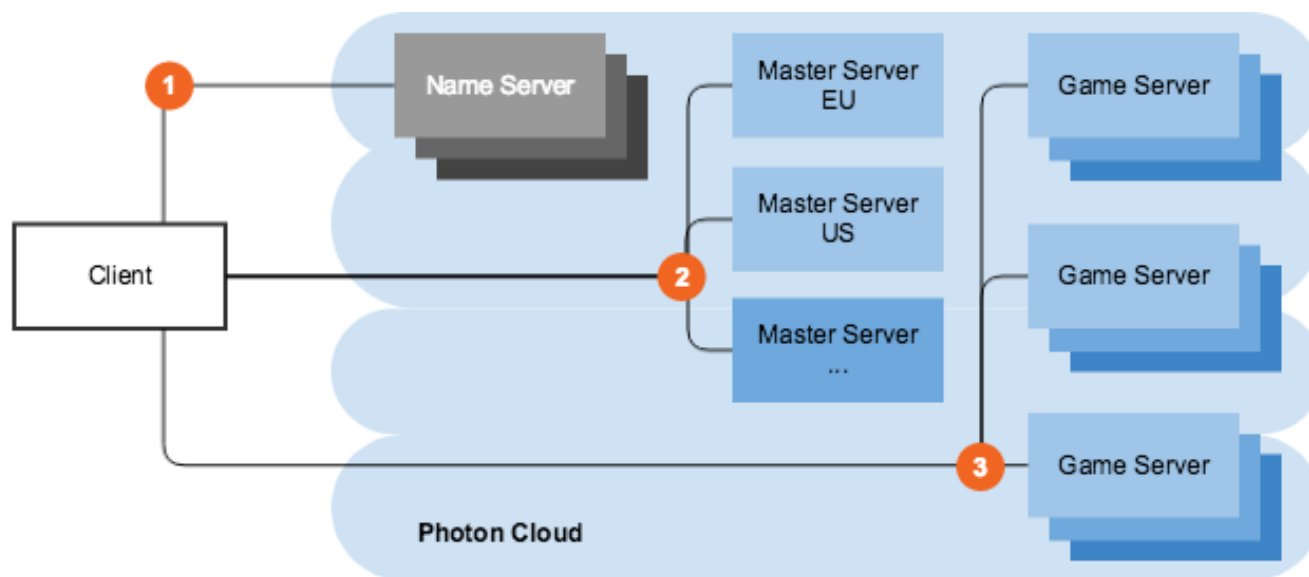
♦ Rješenja:

- ~~Jednonitno programsko rješenje~~
- Višenitno programsko rješenje





- ♦ Svi današnji moderni sustavi za izradu igara podržavaju kreiranje umreženih i višekorisničkih inačica
- ♦ Unity
 - Sustav za UVO unutar Unitya nije dobar te se najčešće koristi biblioteka Photon Unity Networking
 - Unity trenutno radi na reviziji mrežnog koda
- ♦ Unreal
- ♦ Godot



Virtualni svjetovi

- ◆ engl. Virtual Worlds



Virtualni svjetovi - primjeri



Second Life



- ◆ Virtualnim novcem kupuju se dobra i usluge
- ◆ Primjeri usluga: kampiranje, rad u trgovini, zabava (često “za odrasle”), izrada sadržaja
- ◆ Primjeri dobara: zgrade, vozila, nakit, odjeća, animacije, biljke, životinje, umjetnine
- ◆ Virtualni novac se kupuje i prodaje
 - Primjer: SL LindeX: 12.12.2018. tečaj 253,90 L\$ za 1\$
 - Pomoću dane valute trgovci unutar Second Lifea mogu zarađivati kreirajući sadržaje (odjeću, obuću, kuće itd.)

- ◆ Pored ranije opisanih osnovnih svojstava, karakteristike specifične za virtualne svjetove:
- ◆ Zajednički, perzistentni svijet za sve korisnike
- ◆ Korisnici stvaraju svijet!
- ◆ Goleme količine sadržaja koji se stalno mijenja
 - Korisnici kupuju “zemlju”, grade, trguju, programiraju
- ◆ Kada se korisnik kreće kroz svijet, mora stalno učitavati nove ili osvježene sadržaje

- ◆ Osvježavanje stanja korisnika: slično kao u igrama
- ◆ Daleko veći promet generira učitavanje modela svijeta/predmeta i njihovo osvježavanje prilikom promjena
- ◆ Načini na koje se smanjuje promet:
 - Lokalno spremanje (engl. *cache*) smanjuje promet pri drugoj i idućim posjetama istom prostoru
 - UDP za sve osim kritičnih poruka, npr. autentikacija
 - Kompresija (JPEG, zlib)
 - Primjer: Second Life – koristi sve navedene tehnike



- ◆ Mjerenja prilikom posjete Veleposlanstvu Švedske
 - Prva posjeta: blokiranje, dugo čekanje na učitavanje predmeta
 - Druga posjeta: normalan rad, prosjek prometa na klijentu 640 kbit/s

Smjer poruka	Tip prometa	Promet (kb), 1. posjeta	Promet (kb), 2. posjeta
Poslužitelj - klijent	Učitavanje predmeta, tekstura...	11563	2792
	Osvježavanje stanja, događaji	91	50
	Poruke sustava i ostalo	276	151
Klijent -poslužitelj	Učitavanje (zahtjevi)	71	14
	Osvježavanje stanja, događaji	101	87
	Poruke sustava i ostalo	112	69

- ◆ Zbog mrežnih i računalnih zahtjeva jedan poslužitelj služi oko 40 klijenata
 - Usporedba s igrama: stotine ili tisuće klijenata po poslužitelju

- ◆ Virtualni svjetovi su vrhunac popularnosti imali u 2000-tima
 - Club Penguin – čak 200 milijuna registriranih računa – ugašen 2017., a nasljednik Club Penguin Island nije uspio te je najavljeno gašenje
 - Second Life – milijun korisnika u 2007 godini – još uvijek radi, ali s vrlo malim brojem korisnika (oko 45 000)
 - Današnja popularnost puno manja, ali još uvijek postoji veliki broj virtualnih svjetova
- ◆ Minecraft – igra ili virtualni svijet?
 - Prosinac 2017 – 74 milijuna aktivnih korisnika
- ◆ Virtualna i proširena stvarnost donose nove oblike virtualnih svjetova
 - Facebook Horizons
 - Holoportation

Lista virtualnih svjetova (nepotpuna)

- ◆ Active Worlds
- ◆ AltspaceVR
- ◆ Anyland
- ◆ Avakin Life
- ◆ Bigscreen
- ◆ Ceek
- ◆ Cisco Spark in VR
- ◆ Community Garden
- ◆ Cryptovoxels
- ◆ DiveReal
- ◆ Edorable
- ◆ EmbodyMe
- ◆ Endless Riff
- ◆ Engage
- ◆ Eventual VR
- ◆ Ever, Jane
- ◆ Facebook Spaces
- ◆ Geekzonia
- ◆ Guru Gedara
- ◆ High Fidelity
- ◆ Hyperfair VR
- ◆ IMVU
- ◆ Inlight Spark
- ◆ JanusVR
- ◆ Kiss or Kill
- ◆ Kitely
- ◆ LivCloser
- ◆ LiveLike
- ◆ Mark Space
- ◆ Mozilla Hubs
- ◆ Myst Online: Uru Live
- ◆ NeosVR
- ◆ Neutrans
- ◆ Occupy White Walls
- ◆ Oculus Rooms
- ◆ Pluto VR
- ◆ Rec Room
- ◆ Rumii
- ◆ Sansar
- ◆ Second Life
- ◆ Sinespace
- ◆ Somnium Space
- ◆ STYLY
- ◆ SurrealVR
- ◆ Topik
- ◆ There.com
- ◆ TheWaveVR
- ◆ Twinity
- ◆ Center
- ◆ VirBELA
- ◆ Virtual Paradise
- ◆ Virtual Universe
- ◆ VRChat
- ◆ vTime
- ◆ Worldopoly
- ◆ Worlds Adrift



- ♦ Većina virtualnih svjetova koristi kombinacije triju modela
 - Oglašavanje, kanal za prodaju
 - Pretplate
 - Često besplatno osnovno članstvo, naplata za “premium” članstvo
 - Prodaja virtualnih dobara
 - Virtualni novac koji se može trošiti unutar svijeta
 - Dodaci za avatare
- ♦ Ograničenja poslovnog modela s oglašavanjem
 - Svjetovi za djecu – mnogi roditelji nisu skloni izlaganju djece reklamama
 - Tvrtke nisu sklone oglašavanju u nekontroliranom okruženju
 - Troškovi: stalna prisutnost npr. u SL zahtijeva zaposlenika

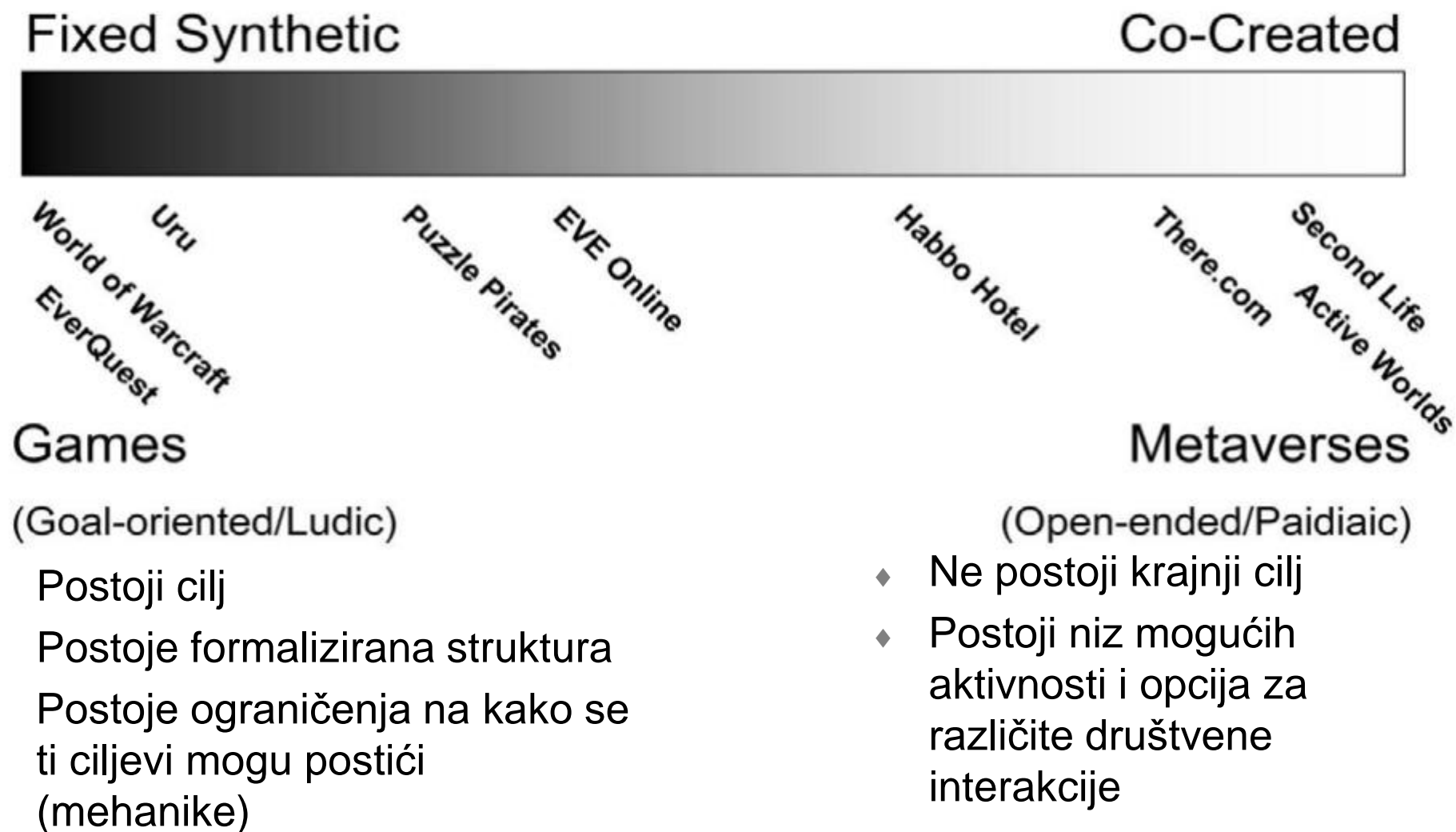
Primjeri poslovnih modela

Svijet	Vlasnik	Ciljani korisnici	Područje	Poslovni model	Korisnici (2008)	Rast 07-08
BarbieGirls	Mattel	Djevojčice	SAD, svijet	Prodaja igračaka, pretplate	3,3M	26%
ClubPenguin	Disney	Djeca	SAD, svijet	Pretplate	10,8M	23%
CyWorld	SK Telecom	Tinejdžeri, 20te	Koreja	Virtualna dobra	13,7M	-1%
Habbo Hotel	Sulake, FI	Tinejdžeri	Europa, svijet	Oglašavanje, virtualni novac	6,3M	52%
NeoPets	Viacom	Djeca	Svijet	Premium članstvo, oglašavanje, virtualna dobra	6,1M	-16%
ActiveWorlds		Odrasli	SAD, svijet	Pretplate, hosting	0,1M	181%
Lively	Google	Odrasli	Svijet	Oglašavanje	0,6M	Ugašen
IMVU		Odrasli	SAD, svijet	Premium članstvo, oglašavanje	4,8M	22%
Second Life	Linden Labs	Odrasli	Svijet	Pretplate, virtualni novac i dobra, oglašavanje	1,6M	-45%



Umrežene igre

Konceptualne razlike između igara i virtualnih svjetova



Pearce, C. (2011). *Communities of play: Emergent cultures in multiplayer games and virtual worlds*. MIT Press.

- ◆ Perzistentnost
 - Virtualni svjetovi su uglavnom perzistentni – postoje i mijenjaju se i dok korisnik nije u njima
 - Kod igara je puno češće iznova stvaranje virtualnog svijeta iz pohranjenih podataka (primjerice svaka bitka u Call of Duty se odvija na istoj mapi koja se iznova kreira)
- ◆ Mijenjanje 3D svijeta
 - Virtualni svjetovi dopuštaju dodavanje novih slika, tekstura, zvukova, videa i 3D objekata (primjerice Second Life)
 - Igre ne dopuštaju dodavanje novih informacija u svoj klijent (iako mogu dopuštati mijenjanje svijeta primjerice Fortnite)
- ◆ Karakteristike mrežnog prometa – jako ovisne o „promjenjivosti” 3D svijeta jer ako korisnici mogu dodati nove informacije, drugi korisnici moraju te iste informacije preuzeti

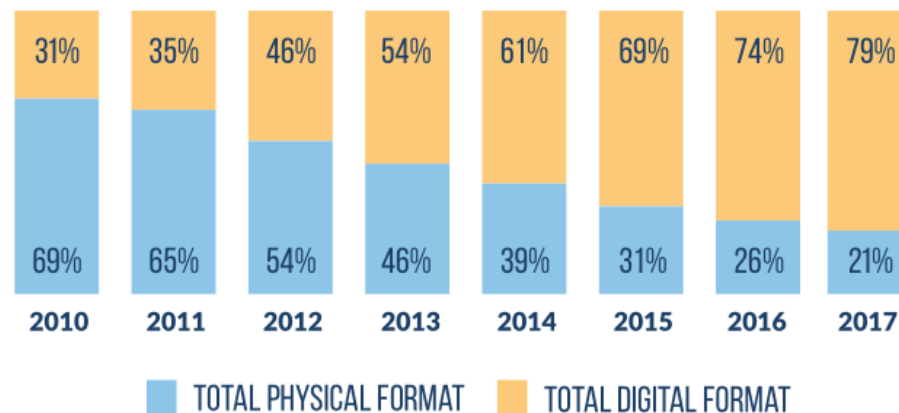


- ♦ Više igrača
- ♦ Zaštita autorskih prava (Digital rights management – DRM)
- ♦ Društvene igre
- ♦ Igre na pokretnim uređajima
- ♦ Distribucija sadržaja
- ♦ Igre na konzolama (Xbox one – 300 000 poslužitelja, prethodna generacija Xbox live samo 30 000)



RECENT DIGITAL* AND PHYSICAL SALES INFORMATION

*Digital format sales include subscriptions, digital full games, digital add-on content, mobile apps, and social network games.



Peak concurrent players

Fortnite – 8,3 million
League of Legends – 7,5 million
Dota 2 – 1,2 million
World of Tanks – 1,1 million
Steam – 16 (5 u igrama) million

- ◆ Peer-to-Peer (P2P)
 - Dobra skalabilnost
 - Nedostaci: loša kontrola varanja, distribucija stanja virtualnog svijeta, problem konzistencije
 - Danas se rijetko koristi “samo P2P” (*Demigod*)
- ◆ Klijent – poslužitelj
 - Klasičan pristup
 - Dobra kontrola
 - Usko grlo – sve ide kroz poslužitelj
 - Više poslužitelja na strani proizvođača (World of Warcraft)
 - Farme poslužitelja na strani proizvođača (EvE online)
 - Klijent postaje poslužitelj (Warcraft III)
 - Namjenski (engl. dedicated) poslužitelji koje mogu održavati igrači (Call of Duty)



Demigod – zašto je P2P loš u praksi

- ♦ Prva igra nastala na konceptu DOTA-e
- ♦ Potpuna P2P arhitektura
- ♦ Dosta dobre ocjene kritičara
- ♦ Inicijalno privučeno puno igrača iako je igra naplaćivana
- ♦ Tehnički problemi
 - Problemi s Network Address Translator (NAT) poslužiteljima – poslužitelji za probijanje NAT-a nisu mogli obraditi veliki broj zahtjeva (veliko kašnjenje u obradi zahtjeva)
 - Veliki broj korisnika koji su bili ilegalni (18 000 regularnih i 100 000 ilegalnih)
 - Veliki broj korisnika su bili od spojeni od svojih mečeva
 - Vrlo loša iskustvena kvaliteta – katastrofalno lansiranje igre što je dugoročno upropastilo



Raspored funkcija unutar igre na klijenta i poslužitelja

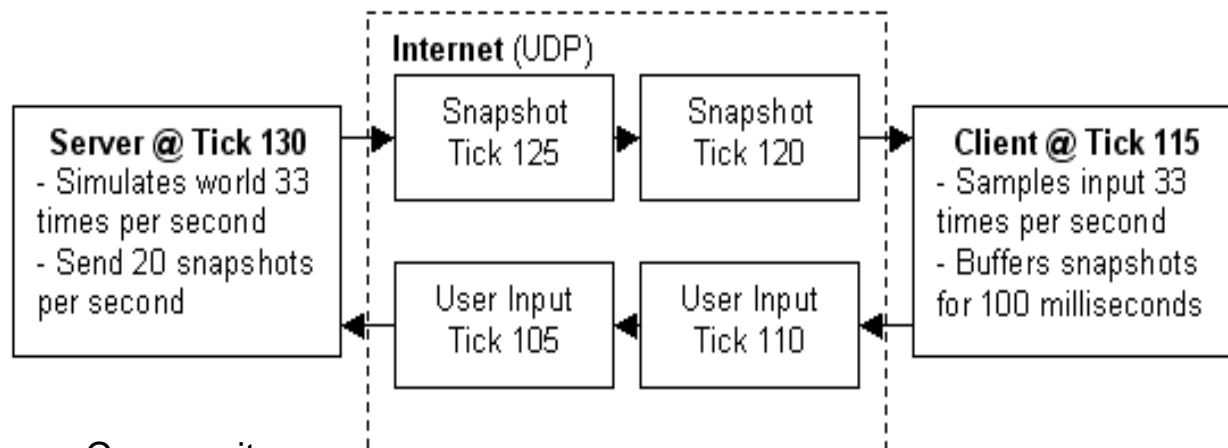
- ◆ Osnovne funkcije
 - Unos komandi (uvijek na klijentu)
 - Izračun stanja virtualnog svijeta
 - Iscrtavanje stanja virtualnog svijeta (renderiranje virtualne scene)
- ◆ Sve funkcije na klijentu (nema poslužitelja)
- ◆ Iscrtavanje stanja virtualnog svijeta na klijentu, logika izračuna stanja virtualnog svijeta na poslužitelju
 - Tradicionalni način rasporeda funkcija
 - Poslužitelj šalje osvježanja stanja
- ◆ Logika izračuna stanja virtualnog svijeta te iscrtavanje virtualnog svijeta na poslužitelju – igre u oblaku (engl. cloud gaming)
 - Poslužitelj šalje video strujanje (dodatna funkcija kodiranja videa na poslužitelju)
 - Visoki zahtjevi na propusnost mreže
- ◆ Hibridni – dio na klijentu, a dio funkcija na poslužitelju



- ♦ Svaki period t server izračunava novo stanje svijeta na temelju primljenih unosa klijenata
- ♦ Taj period definira broj otkucaja (engl. tickrate) – koliko puta u sekundi se treba izračunati stanje svijeta – mjeri se u Hz
- ♦ Brzina izračuna stanja na poslužitelju ne mora biti ista kao i broj poslanih osvježenja – primjerice ako se stanje nije promijenilo ne mora se slati osvježenje
- ♦ Igre interpoliraju pozicije i stanje virtualnog svijeta između dva osvježenja stanja radi fluidnog prikaza, ali na razini poslužitelja postoje samo stanja izračunata svakog period t
- ♦ Brzina prikaza igre na računalu je izražena u broju sličica u sekundi (engl. framerate)

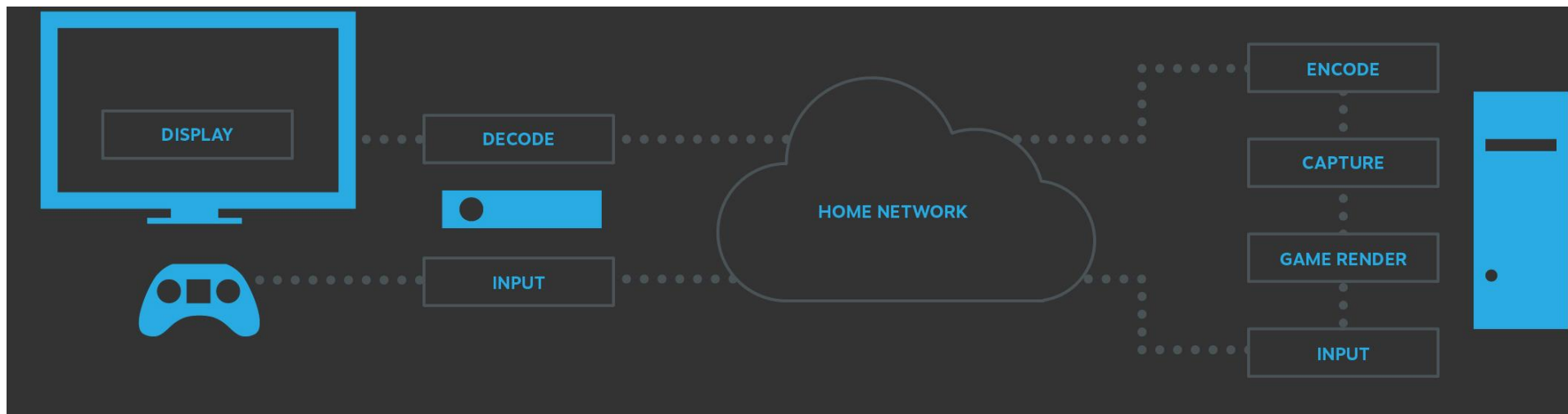
Mrežno funkcioniranje Source pokretačkog sustava (tradicionalni koncept)

- ♦ Source pokretački sustav (engl. game engine) pokreće igre poput Counter Strike Source (CSS), Team Fortress 2 (TF2), Left 4 Dead (L4D) itd.
- ♦ Simulacija se izvršava na poslužitelju.
 - Tickrate 66 za CSS, a 30 za TF2 i L4D – ovisno i interaktivnosti igre
 - Poslužitelj šalje razlike između pojedinih snimaka, a ne cijelo stanje
- ♦ Klijent određuje mrežne parametre
 - Dojavljuje mrežnu propusnost – dizajnirano za propusnost od 40-60 Kb/s
 - Dojavljuje koliki broj osvježanja želi po sekundi (20 je predefiniрани broj)
 - Klijent ne šalje zasebno svaku komandu već ih šalje određenom brzinom (30 puta u sekundi je predefiniрани broj) te je obično više od jedne naredbe u jednom paketu
 - Klijent može zahtijevati kompletnu snimku stanja



Izvor: Valve Developer Community

- ◆ Primjer na platformi Steam In-Home Streaming
- ◆ “Oblak” je snažno kućno računalo
- ◆ Osnovne funkcije:
 - Unos komandi
 - Računanje stanja igre
 - Renderiranje igre
 - Snimanje i kodiranje videa
 - Dekodiranje videa i prikaz

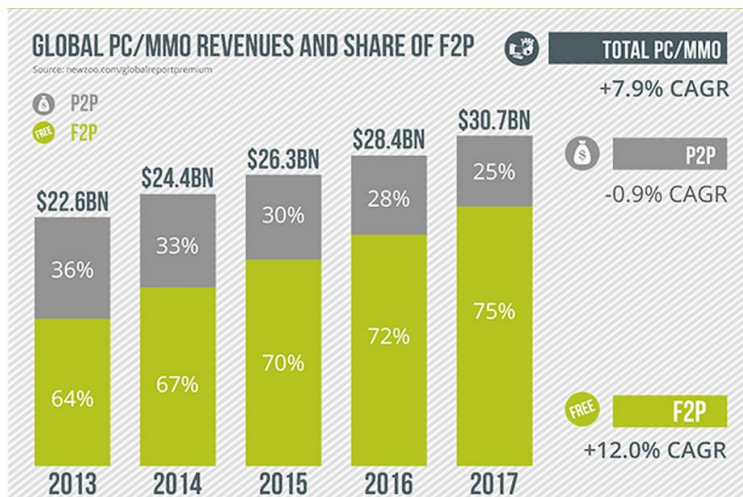


Izvor: https://support.steampowered.com/kb_article.php?ref=3629-RIAV-1617

- ♦ *Pay-to-Play (P2P)*
 - Prodaja igre (Guild Wars)
 - Pretplata (World of Warcraft)
 - Prodaja dodataka na igru
 - Manji – (Downloadable Content - DLC) (Mass Effect 3)
 - Veći – ekspanzije (Age of Conan: Rise of the Godslayer)
- ♦ *Free-to-Play (F2P)*
 - Besplatni dio igre ograničen (Star Wars: The Old Republic)
 - Prodaja virtualnih dobara koji utječu na performanse (Travian)
 - Prodaja kozmetičkih dodataka (Dota 2)
 - “Porez” na trgovinu između igrača (Diablo 3)
- ♦ *Hibridni modeli*
 - Igra se plaća, ali postoje i dodatne mogućnosti unutar igre

Tržište – udio pojedinog modela

- ♦ F2P model preuzima tržište
- ♦ Mnoge „premium” igre imaju F2P komponente (primjerice World of Warcraft)
 - Prvi plaćeni jahači konj unutar WoW igre je imao cijenu od 25 USD (dvije mjesečne pretplate)
 - Nakon izdavanja se stvorio rep čekanja od 140 000 ljudi koji su htjeli ga kupiti (čekalo se i do 7 sati) – 3,5 milijuna USD



TOP PC GAMES BY REVENUE IN 2017

FREE-TO-PLAY		PREMIUM	
TITLE	REVENUE	TITLE	REVENUE
LEAGUE OF LEGENDS	\$2.1B	PUBG	\$714M
DUNGEON FIGHTER ONLINE	\$1.6B	OVERWATCH	\$382M
CROSSFIRE	\$1.4B	CS:GO	\$341M
WORLD OF TANKS	\$471M	DESTINY 2	\$218M
DOTA 2	\$406M	GRAND THEFT AUTO V	\$118M
ROBLOX	\$310M	BATTLEFIELD 1	\$113M
MAPLESTORY	\$279M	MINECRAFT	\$92M
HEARTHSTONE	\$217M	GUILD WARS 2	\$87M
BLADE & SOUL	\$178M	DIVINITY: ORIGINAL SIN 2	\$85M
FIFA ONLINE 3	\$163M	RAINBOW SIX: SIEGE	\$67M
TOTAL:	\$7.1B	TOTAL:	\$2.2B



Izvori: SuperData research <https://www.superdataresearch.com/market-data/market-brief-year-in-review/>
<https://newzoo.com/globalgamesreport>

- ♦ S. Singhal & M. Zyda, *Networked Virtual Environments: Design and Implementation*, Addison-Wesley, 1999 (poglavlja 4, 5)
- ♦ Abdelkhalek, A.; Bilas, A. and Moshovos, A. (2001), Behavior and Performance of Interactive Multiplayer Game Servers, Proc. Int. IEEE Symposium on the Performance Analysis of Systems and Software.
- ♦ Gao Huang, Meng Ye, Long Cheng, “Modeling System Performance in MMORPG”, in IEEE CSG Workshops, 2004, pp. 512-518.
- ♦ Seay, A. F., Jerome, W. J., Lee, K. S., and Kraut, R. E. 2004. Project massive: a study of online gaming communities. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM, New York, NY, 1421-1424.
- ♦ Kumar, Chhugani, Kim, Kim, Nguyen, Dubey: “Second Life and the New Generation of Virtual Worlds”, IEEE Computer, 2008
- ♦ D-Lib Magazine, December 2005, Volume 11, Nr. 12 (John Kirriemuir)
- ♦ ESA: Entertainment Software Association, <http://www.theesa.com/>
- ♦ DFC Intelligence, <http://www.dfcint.com>
- ♦ “Telepresence report”, Howard S. Lichtman, Human Productivity Lab
- ♦ Virtual Worlds Management, <http://www.virtualworldsmanagement.com/>
- ♦ JP Morgan Global Equity Research, 2009