



SVEUČILIŠTE U ZAGREBU



Fakultet
elektrotehnike i
računarstva

Diplomski studij

Računarstvo

Znanost o mrežama

Programsko inženjerstvo i

informacijski sustavi

Računalno inženjerstvo

**Ostali (slobodni izborni
predmet)**

Raspodijeljeni sustavi

1. Raspodijeljene arhitekture programskih sustava. Centralizirana i decentralizirana rješenja.

Ak. god. 2022./2023.

Sadržaj predavanja

- **Definicija, obilježja i vrste raspodijeljenih sustava**
- Zahtjevi na raspodijeljene sustave: otvorenost, transparentnost, skalabilnost i kvaliteta usluge
- Arhitektura raspodijeljenih sustava
- Primjeri modela raspodijeljene obrade
- Studijski primjeri:
 - Raspodijeljeni sustav weba
 - Internet stvari

Definicija raspodijeljenog sustava (1)

Andrew S. Tanenbaum:

- “Skup neovisnih računala koji korisniku izgleda kao jedan cjeloviti sustav.”

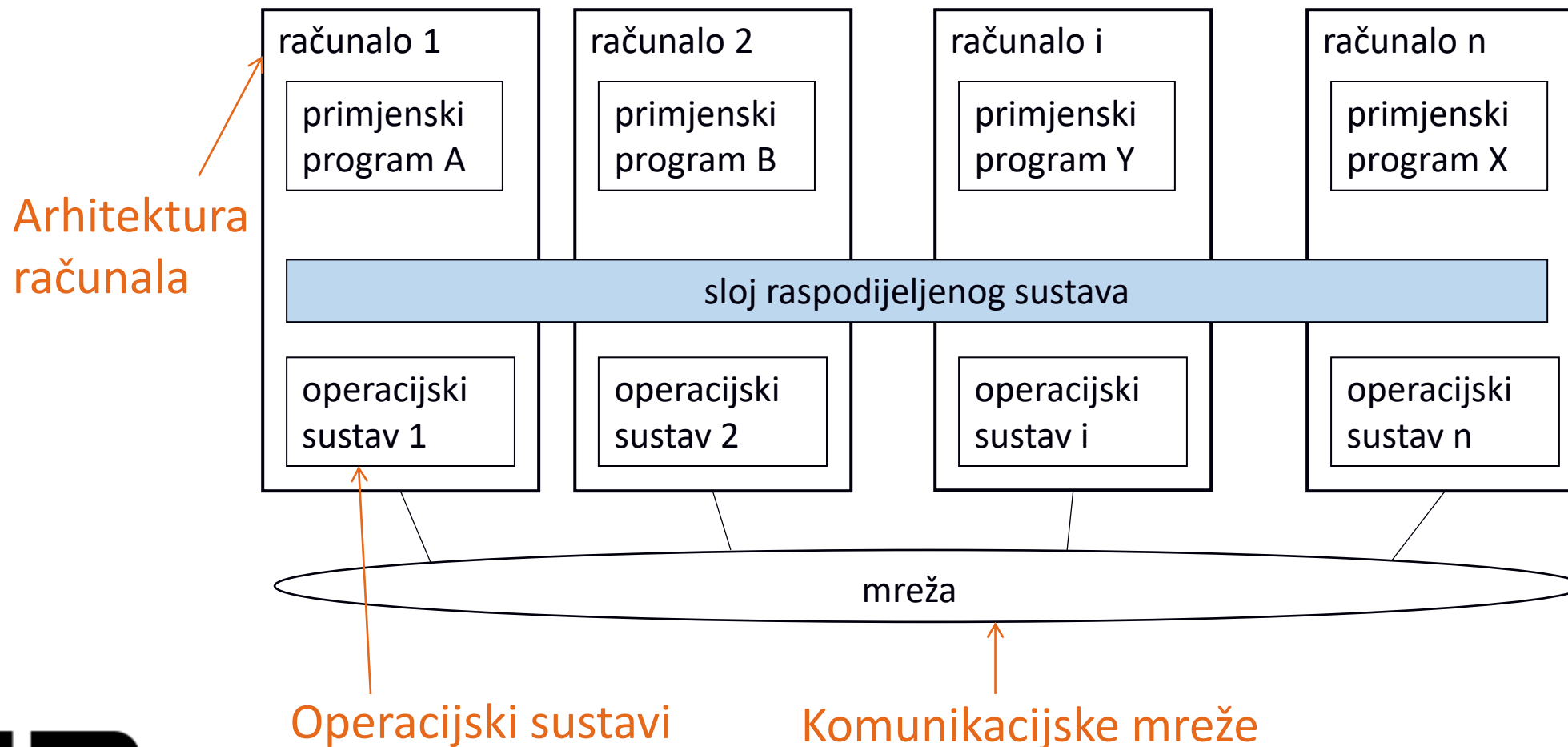
George Coulouris:

- “Sustav u kojem programske i sklopovske komponente umreženih računala komuniciraju i usklađuju svoje aktivnosti isključivo razmjenom poruka.”

Leslie Lamport:

- “Sustav u kojem kvar računala za koje uopće ne znate da postoji može učiniti vaše računalo neupotrebljivim.”

Definicija raspodijeljenog sustava (2)



Sloj raspodijeljenog sustava

Programski posrednički sloj raspodijeljenog sustava, međuoprema (engl. *middleware*)

- prikriva činjenicu da su procesi i sredstva (resursi) raspodijeljeni na više umreženih računala
- omogućuje povezivanje i suradnju aplikacija, sustava i uređaja
- omogućuje interakciju programa na aplikacijskoj razini

Razlozi za raspodijeljene sustave

Inherentna raspodijeljenost:

- korisnika, uređaja, stvari, informacija, sredstava, ...

Funkcijsko odvajanje:

- različite namjene, različite mogućnosti, različite uloge (korisnik – davatelj usluge, proizvođač – potrošač,)

Opterećenje:

- mogućnost raspodjele i uravnoteženja

Pouzdanost i raspoloživost:

- ostvarene s više komponenata na različitim mjestima

Cijena, troškovi, održavanje...

Vrste raspodijeljenih sustava (1)

Raspodijeljeni računalni sustavi:

- grozd računala (engl. *cluster*)



Računalni klaster Isabella

<http://www.srce.unizg.hr/isabella>

135 računalnih čvorova s 3100 procesorskih jezgri i 12 grafičkih procesora

- spletna računala (engl. *grid*)



CRO NGI, Hrvatska nacionalna grid infrastruktura

<https://www.srce.unizg.hr/cro-ngi>

1.868 procesorskih jezgri, 36 grafičkih procesora i 205 TB podatkovnog prostora

- računalni oblak (engl. *cloud*)



<https://www.srce.unizg.hr/hr-zoo>

Vrste raspodijeljenih sustava (2)

Sustavi za pružanje informacijskih i komunikacijskih usluga

- usluge (Skype, Facebook, MS Teams, Gmail, Twitter...)

Raspodijeljeni poslovni i transakcijski sustavi:

- poslovni i transakcijski sustavi (Amazon online shop, Bitcoin, ...)

Internet stvari (engl. *Internet of Things*, IoT)

- povezivanje stvari/uređaja na Internet – fizičkih i virtualnih objekata

Internet svega (engl. *Internet of Everything*, IoE)

- inteligentno povezivanje ljudi, procesa, podataka i stvari

Obilježja raspodijeljenog sustava

Paralelne i konkurentne aktivnosti:

- autonomne komponente sustava istodobno izvode više aktivnosti

Komunikacija razmjenom poruka:

- komponente sustava razmjenjuju podatke porukama, ne dohvaćaju ih iz zajedničke memorije

Dijeljenje sredstava:

- zajedničkim sredstvima pristupa više komponenta sustava

Nema globalnog stanja:

- niti jedan proces ne zna stanje svih ostalih procesa u svim komponentama sustava

Nema globalnog vremenskog takta:

- ograničena mogućnost vremenskog usklađivanja

Temeljni teorijski modeli

*Modeli raspodijeljenih sustava sastoje se od **procesa koji međusobno komuniciraju** i razmjenjuju poruke putem komunikacijskog kanala*

Temeljni formalizmi:

- **Komunikacijski i interakcijski model**: procesi, komunikacija, vremenska usklađenost odvijanja i komunikacije procesa
- **Model kvara**: kvarovi i njihov utjecaj na odvijanje i komunikaciju procesa
- Model sigurnosti: prijetnje odvijanju i komunikaciji procesa te mjere zaštite (ne obrađuje se u okviru predmeta Raspodijeljeni sustavi)

Sadržaj predavanja

- Definicija, obilježja i vrste raspodijeljenih sustava
- **Zahtjevi na raspodijeljene sustave: otvorenost, transparentnost, skalabilnost i kvaliteta usluge**
- Arhitektura raspodijeljenih sustava
- Primjeri modela raspodijeljene obrade
- Studijski primjeri: raspodijeljeni sustav weba, Internet stvari

Zahtjevi na raspodijeljene sustave

- **Otvorenost**

- otvoreni sustav (engl. *open system*): pruža usluge sukladno normiranim pravilima te definiranoj sintaksi i semantici

- **Transparentnost**

- prikrivanje odabranih značajki raspodijeljenog sustava

- **Skalabilnost**

- sposobnost razmjerne prilagodbe veličini (broj korisnika – količina sredstva), rasprostranjenosti (lokalno, regionalno, globalno, ...) i načinu upravljanja (jedna ili više administrativnih domena)

- **Kvaliteta usluge**

- performance (npr. vrijeme odziva), raspoloživost/pouzdanost, trošak

Otvorenost

Norma ili standard je specifikacija koja je:

- široko prihvaćena u industriji (*de facto standard*) ili zastupana od normizacijskog tijela (*de jure standard*),
- dobro definirana,
- neutralna, tj. vlasnički neovisna i
- javno dostupna.

Otvorenost je preduvjet za:

- međudjelovanje (engl. *interoperability*)
- prenosivost (engl. *portability*)
- proširljivost (engl. *extensibility*)

Transparentnost (1)

Transparentnost pristupa (engl. *access transparency*)

- prikrivanje razlika u pristupu sredstvima i predočavanju podataka (različite arhitekture računala, različiti operacijski sustavi, različite baze podataka, ...)

Lokacijska transparentnost (engl. *location transparency*)

- prikrivanje lokacije sredstva: položaj sredstva u sustavu ne treba biti i nije poznat korisniku
- primjer: poslužitelj www.fer.unizg.hr čiju lokaciju (IP-adresu) zna DNS-poslužitelj

Transparentnost (2)

Migracijska transparentnost (engl. *migration transparency*)

- prikrivanje promjene lokacije: promjena lokacije sredstva ne utječe na način pristupa sredstvu

Relokacijska transparentnost (engl. *relocation transparency*)

- prikrivanje premještanja sredstva tijekom njegove uporabe: sredstvu se može pristupiti i može se upotrebljavati tijekom njegove relokacije, tj. premještanja

Replikacijska transparentnost (engl. *replication transparency*)

- prikrivanje više istovrsnih sredstava ili više preslika nekog sredstva (sve replike nude istu funkcionalnost)

Transparentnost (3)

Konkurencijska transparentnost (engl. *concurrency transparency*)

- prikrivanje istodobne uporabe istog resursa od strane više korisnika: zajednička/dijeljena uporaba sredstva uz očuvanje konzistentnosti

Transparentnost na kvar (engl. *failure transparency*)

- prikrivanje kvara: otkrivanje kvara i obnavljanje sustava nakon kvara nije uočljivo korisnicima
- problem otkrivanja kvara: veliko opterećenje može se očitovati kao kvar (npr. nema odgovora u očekivanom vremenu)

Skalabilnost (1)

Kako bi se uz promjenu broja korisnika održale performance sustava uz prihvatljive troškove treba osigurati:

- | | |
|------------------------------|---------|
| • više (istovrsnih) dijelova | koliko? |
| • prostorno raspodijeljenih | gdje? |
| • koji komuniciraju | kako? |

Primjeri neskabilnih rješenja

- centralizirana usluga: jedan poslužitelj za sve korisnike
- centralizirani podaci: jedan poslužitelj sa svim korisničkim podacima
- centralizirani algoritam: svi podaci o sustavu poznati na glavnom procesu

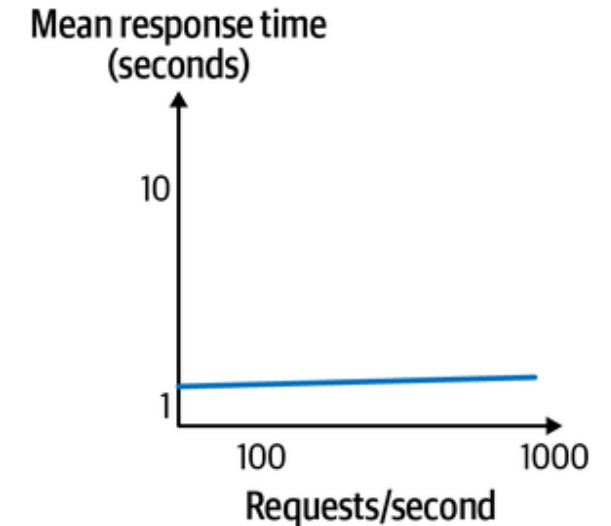
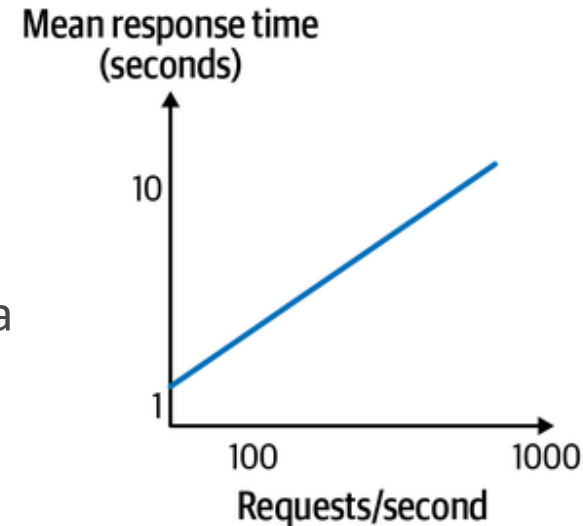
Mogu li se centralizirana rješenja nositi s povećanim brojem korisnika?

- Za motivaciju pogledati <https://www.internetlivestats.com/>

Skalabilnost (2)

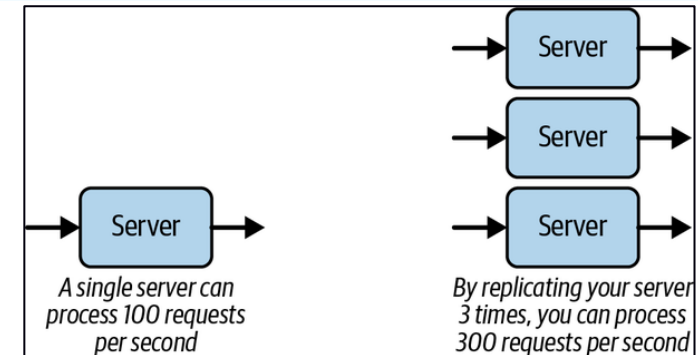
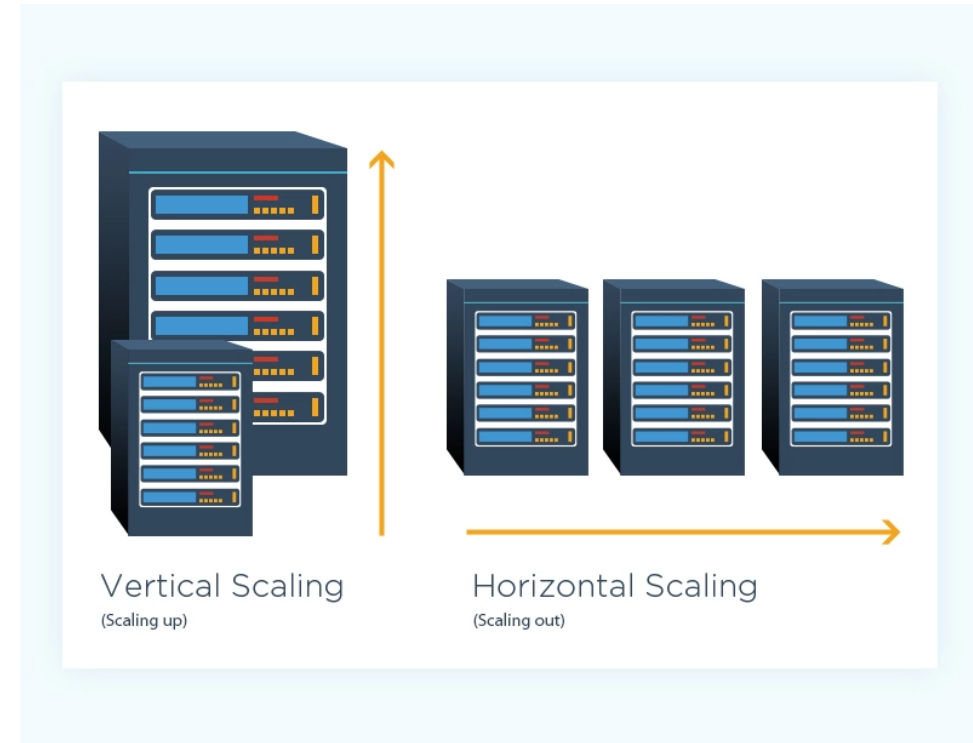
Tehnike koje omogućuju skalabilnost sustava

- Prikrivanje kašnjenja u komunikaciji
 - “radi nešto korisno dok čekaš odgovor” - asinkrona komunikacija
- Komponentno oblikovanje
 - više komponenti koji omogućuju funkcionalnost sustava (npr. raspodijeljena baza podataka, sustav imenovanja domena (DNS))
 - optimizacija implementacije pojedine komponente i cijelog sustava



Skalabilnost (3)

- Replikacija
 - replika = istovjetna kopija dijela sustava (funkcionalnosti) ili podatka
 - **horizontalno skaliranje, tzv. scale out** (radi povećanja propusnosti sustava)
 - problem: konzistentnost originala i kopije, upravljanje skupinom replika



povećanje propusnosti (broj
obrađenih zahtjeva u sekundi)

Kvaliteta usluge

Kvaliteta usluge (engl. *Quality of Service, QoS*)

- skupni naziv za nefunkcijska obilježja sustava, od kojih su posebno važna sljedeća:
 - **vrijeme odziva** (engl. *response time*): vremenski period od slanja zahtjeva do primitka odgovora
 - **propusnost** (engl. *throughput*): mjeri promet na poslužitelju ili usluzi, a izražava se brojem zahtjeva u sekundi ili bit/s.
 - **raspoloživost** (engl. *availability*): vjerojatnost da je usluga dostupna u trenutku t i da generira odgovor na korisnički zahtjev.

Iskustvena kvaliteta (engl. *Quality of Experience, QoE*)

- mjera korisnikovog zadovoljstva uslugom sustava

Oblikovanje raspodijeljenih sustava

Definiranje zahtjeva, potrebno odgovoriti na sljedeća pitanja

- Koje funkcijske zahtjeve treba ostvariti – **ŠTO** sustav treba raditi?
- Kakve nefunkcijske zahtjeve treba ostvariti – **KAKO** sustav treba raditi (kakva se kvaliteta usluge zahtijeva)?
- Temelji li se sustav na otvorenim rješenjima?
- Kakav je stupanj transparentnosti potreban i kako utječe na složenost, performance i troškove sustava?
- Kakva je skalabilnost sustava potrebna s motrišta veličine, rasprostranjenosti i upravljanja?

Sadržaj predavanja

- Definicija, obilježja i vrste raspodijeljenih sustava
- Zahtjevi na raspodijeljene sustave: otvorenost, transparentnost, skalabilnost i kvaliteta usluge
- **Arhitektura raspodijeljenih sustava**
- Primjeri modela raspodijeljene obrade
- Studijski primjeri: raspodijeljeni sustav weba, Internet stvari

Arhitektura raspodijeljenih sustava

Programska arhitektura:

- logička organizacija sustava: programske komponente sustava, njihova organizacija i interakcija

(centralizirana arhitektura ili decentralizirana arhitektura)

Sustavska arhitektura (engl. *deployment*):

- smještaj programskih komponenti na raspoložive računalne resurse

Kako predočiti raspodijeljeni sustav?

Slojevita arhitektura

- u središtu pozornosti aplikacijski sloj (sloj primjene)
- aplikacijski programi i procesi te usluge koje im pružaju niži slojevi

Arhitektura temeljena na komponentama

- npr. mikrousluga: komponenta sustava s dobro definiranim sučeljem
- mehanizam komunikacije, usklađivanja i suradnje mikroservisa

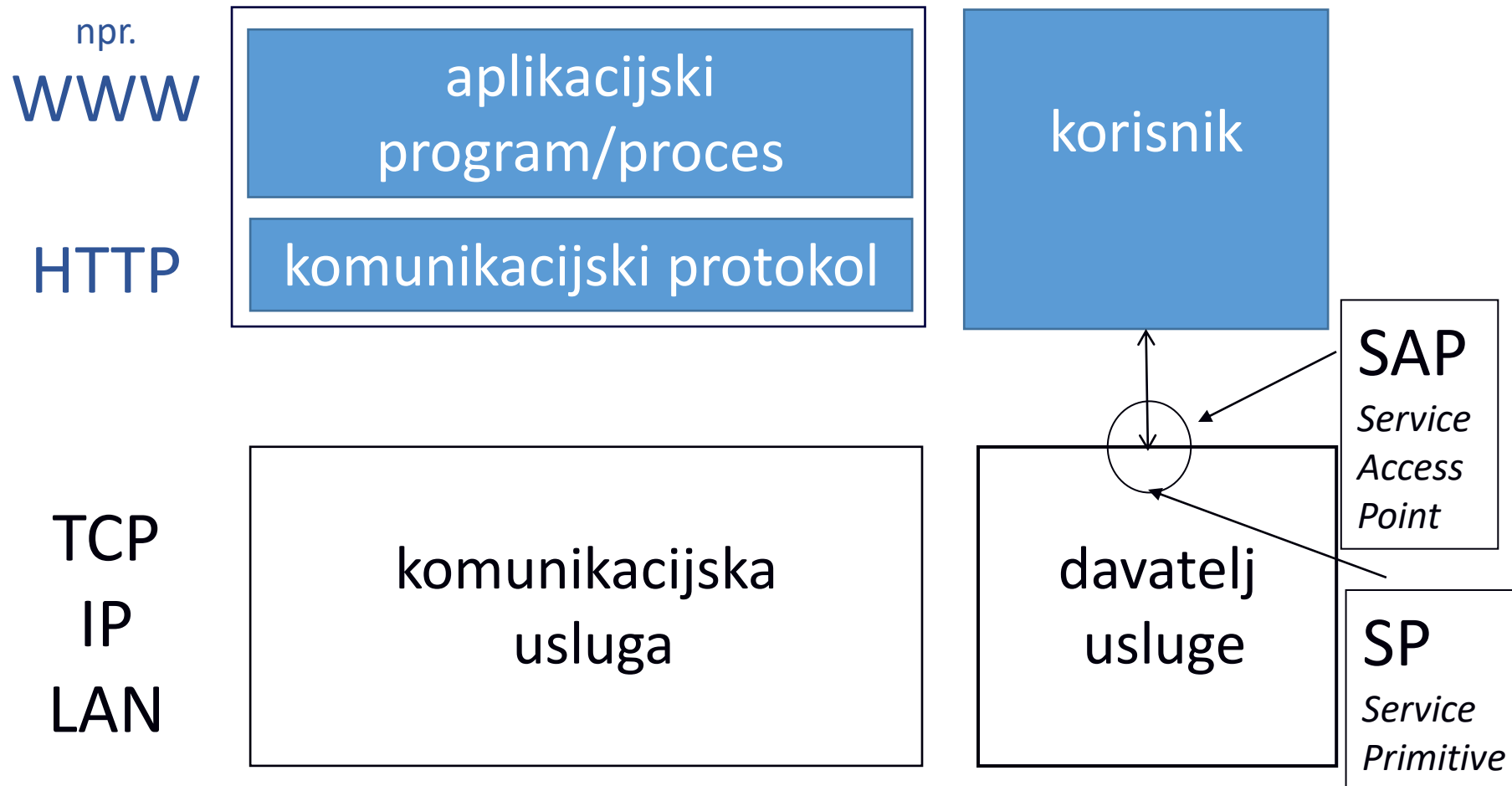
Arhitektura temeljena na podacima

- procesi komuniciraju putem zajedničkog, dijeljenog (raspodijeljenog) repozitorija

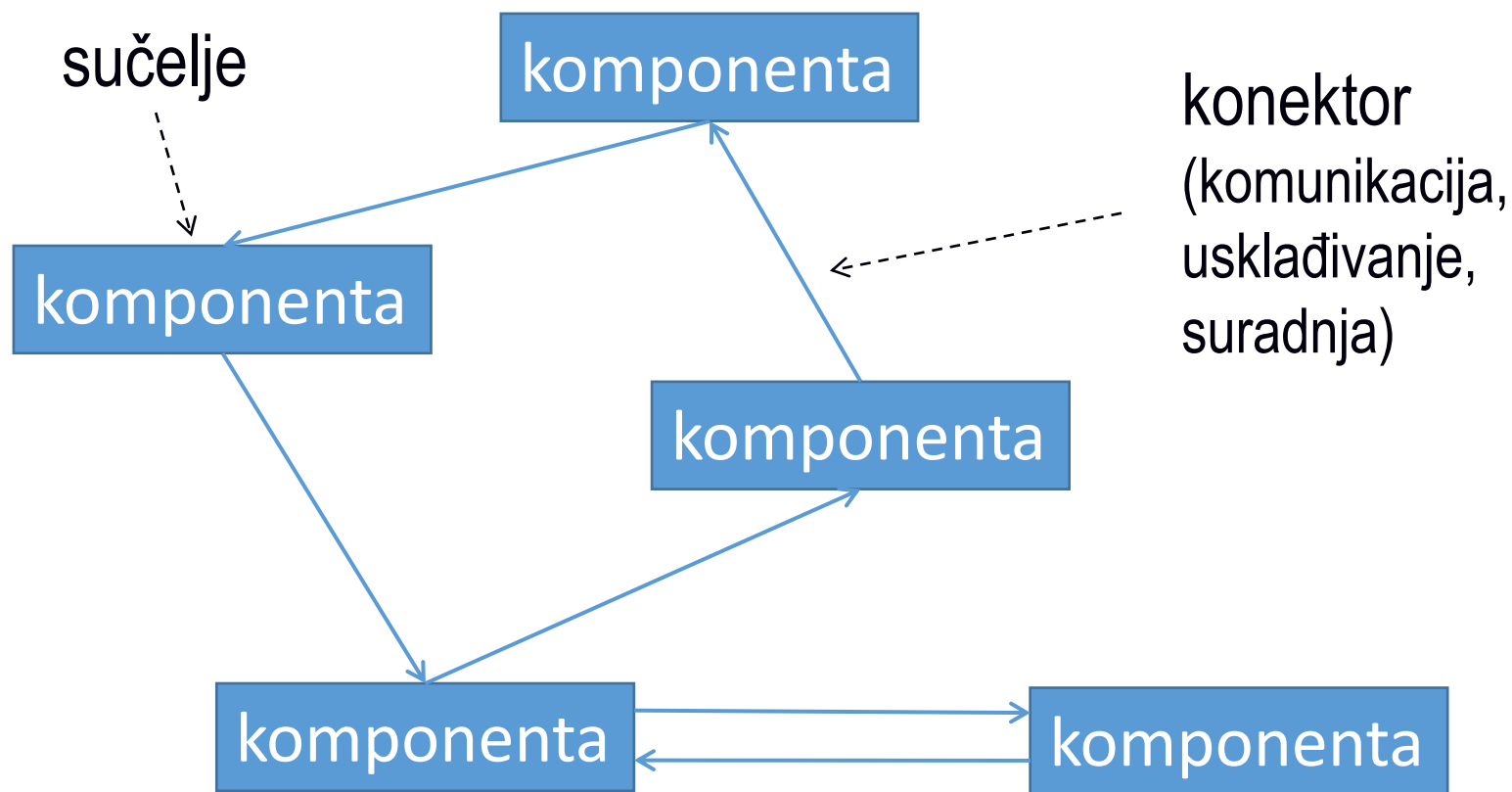
Arhitektura temeljena na događajima

- procesi komuniciraju razmjenom tzv. događaja koji prenose informacije

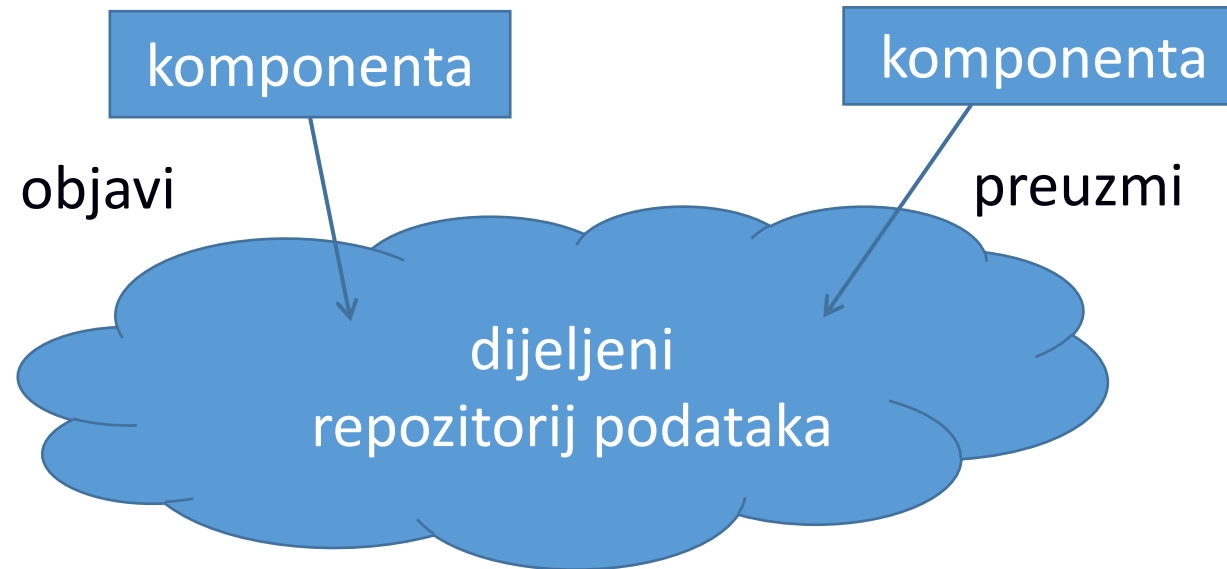
Slojevita arhitektura



Arhitektura temeljena na komponentama

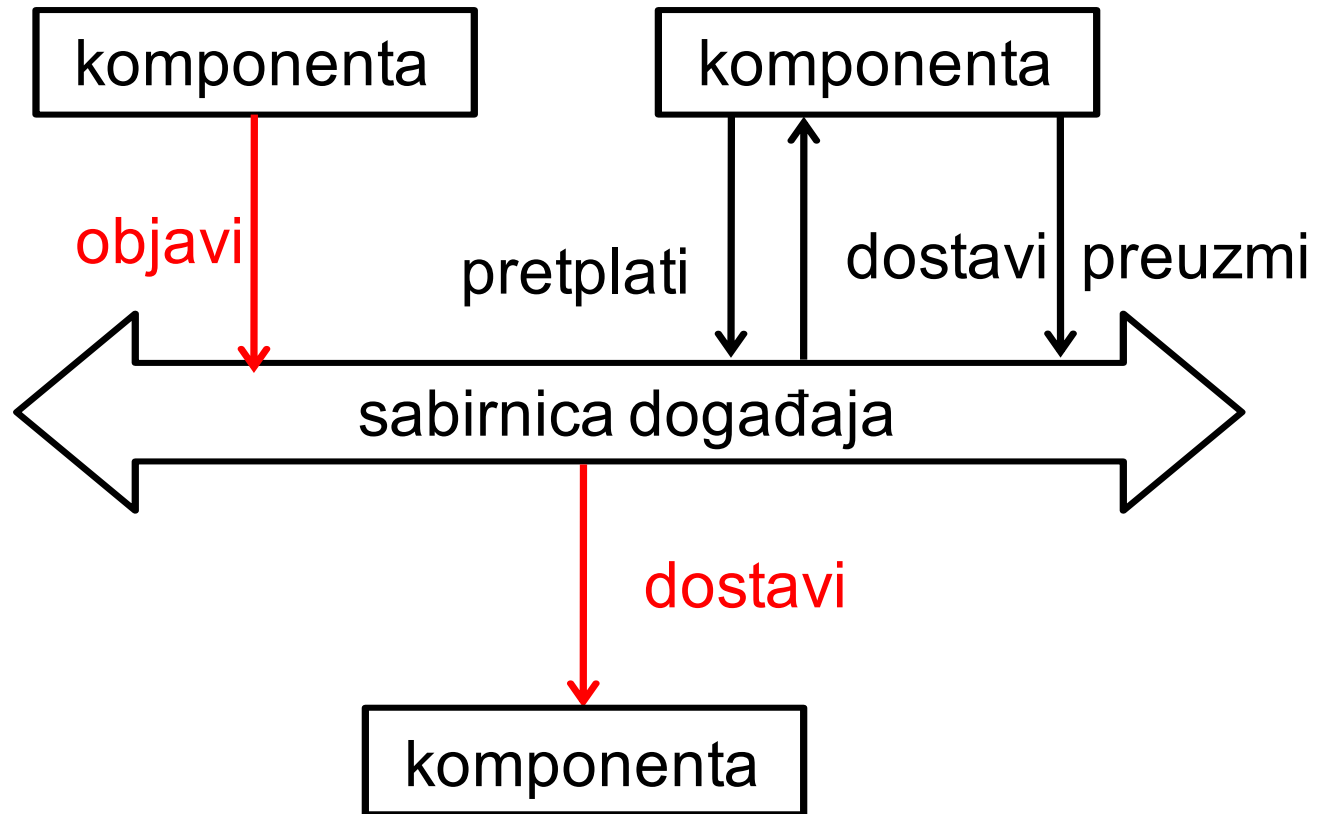


Arhitektura temeljena na podacima



Komponenta sustava upisuje (objavljuje) podatak u dijeljeni repozitorij koji omogućuje čitanje (preuzimanje) podatka drugim komponentama.

Arhitektura temeljena na događajima



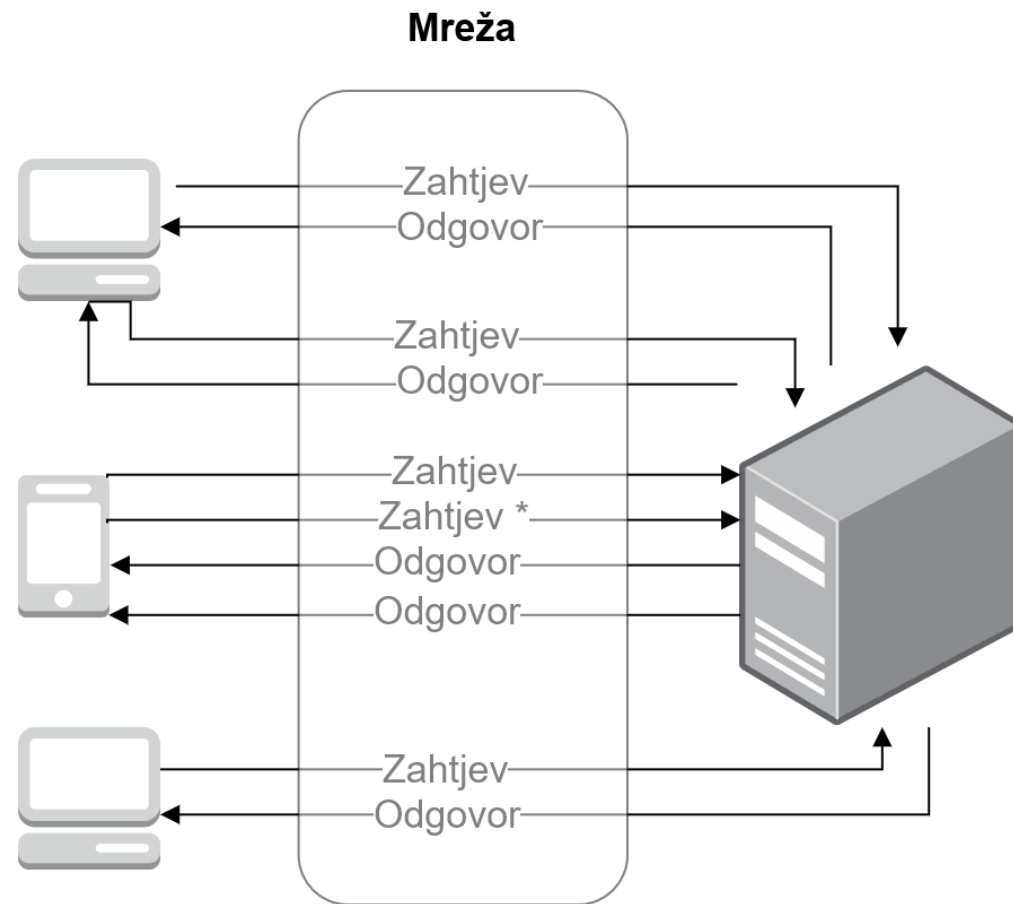
Komponenta se može pretplatiti na podatak koji joj je potreban. Kad neka komponenta objavi podatak (događaj!), to će se dostaviti (događaj!) komponenti koja je na njega pretplaćena.

Sadržaj predavanja

- Definicija, obilježja i vrste raspodijeljenih sustava
- Zahtjevi na raspodijeljene sustave: otvorenost, transparentnost, skalabilnost i kvaliteta usluge
- Arhitektura raspodijeljenih sustava
- **Primjeri modela raspodijeljene obrade**
- Studijski primjeri: raspodijeljeni sustav weba, Internet stvari

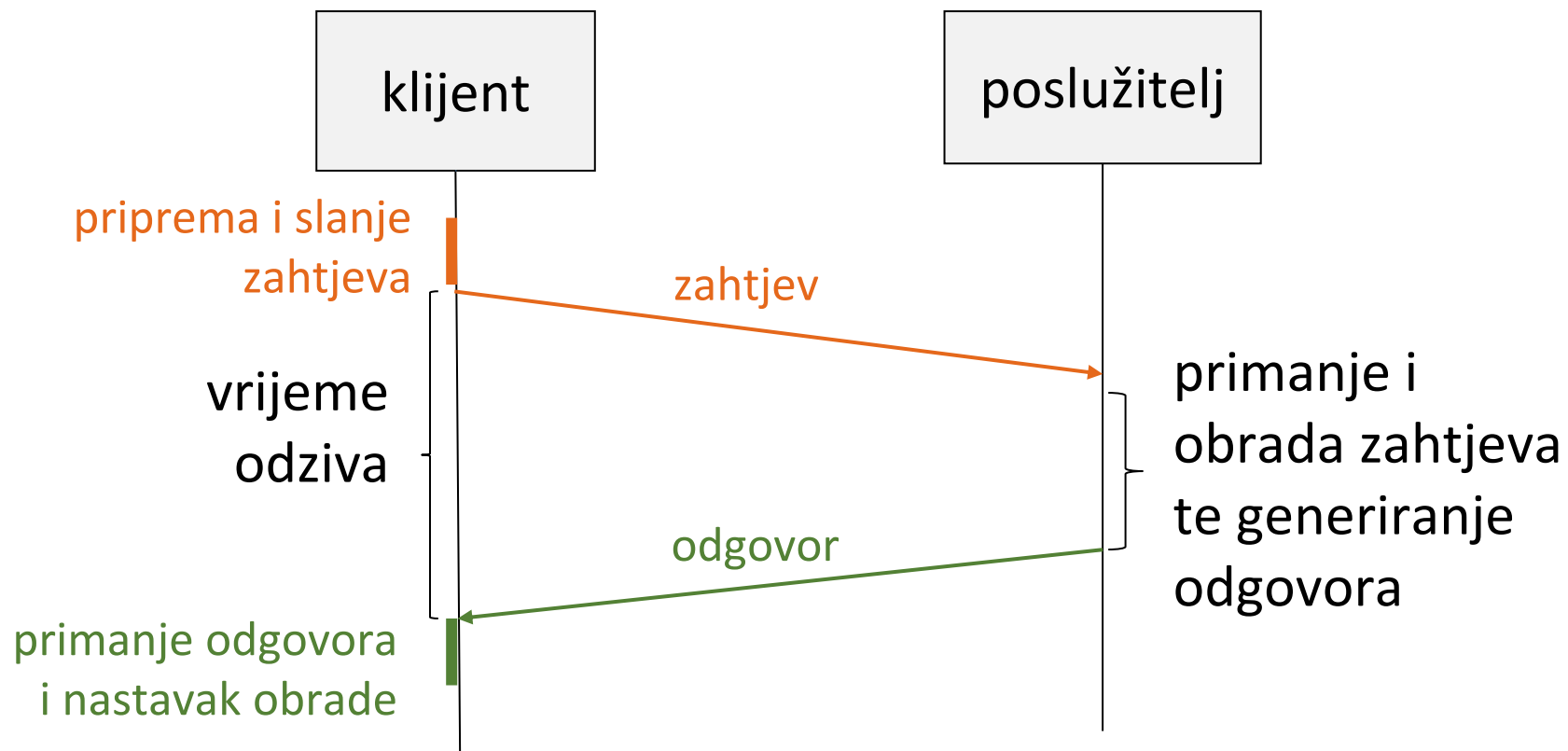
Model klijent – poslužitelj (1)

- Klijent: traži uslugu (zahtjev)
- Poslužitelj: pruža uslugu (odgovor) za više/mnogo klijenta



Model klijent – poslužitelj (2)

- Klijent šalje zahtjev i čeka odgovor
- Poslužitelj: prihvata i obrađuje zahtjev te vraća odgovor

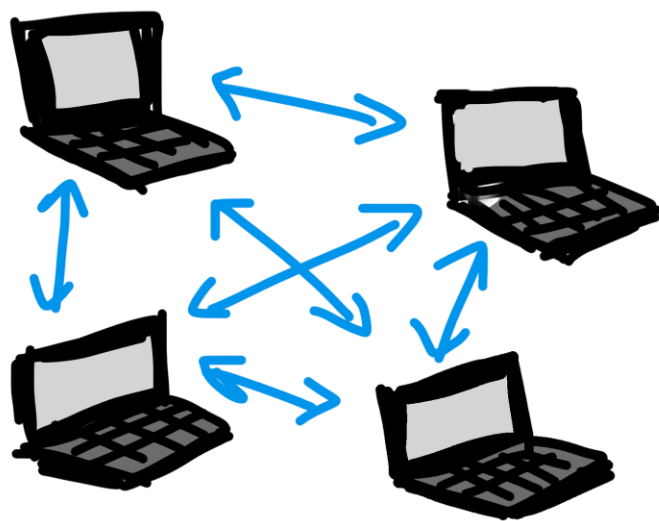


Model ravnopravnih sudionika (1)

- ravnopravni sudionik (engl. *peer*) je onaj koji može obaviti i funkciju poslužitelja i funkciju klijenta
- ravnopravni sudionici međusobno komuniciraju (engl. *Peer-to-Peer*, P2P) tako da se na aplikacijskom sloju povezuju u “prekrivajuću mrežu” (engl. *overlay network*) nad stvarnom mrežnom topologijom
- svaki čvor “plaća” sudjelovanje u mreži nudeći dio vlastitih sredstava ostalim čvorovima
- model ravnopravnih sudionika potencijalno nudi neograničena sredstva u velikim mrežama s puno čvorova

Model ravnopravnih sudionika (2)

Peer-to-peer, P2P



Decentralizirani raspodijeljeni sustav

- nema centralizirane koordinacije među *peerovima*
- ne postoji jedna točka ispada

Samoorganizirajuća mreža čvorova

- *peerovi* su međusobno neovisni, ulaze i izlaze iz sustava po volji

Programski agenti i premještanje programa

- **Programski agent** (engl. *software agent*): program koji obavlja neki posao za svog korisnika ili vlasnika, a raspolaže svojstvima kao što su **inteligencija, samostalnost, reaktivnost, proaktivnost** itd.
- **Migracija programa** (engl. *code migration*): razmjena programa između umreženih čvorova:
 - migracija procesa s jednog računala na drugo zbog (proširenja) funkcionalnosti, (uravnoteženja) opterećenja, (uvođenja) konkurentnosti,
- **Pokretni agent** (engl. *mobile agent*): programski agent koji predodčuje korisnika u mreži i za njega obavlja neki posao krećući se samostalno između čvorova u mreži;

Sadržaj predavanja

- Definicija, obilježja i vrste raspodijeljenih sustava
- Zahtjevi na raspodijeljene sustave: otvorenost, transparentnost, skalabilnost i kvaliteta usluge
- Arhitektura raspodijeljenih sustava
- Primjeri modela raspodijeljene obrade
- **Studijski primjer: raspodijeljeni sustav weba**

Osnovne postavke weba



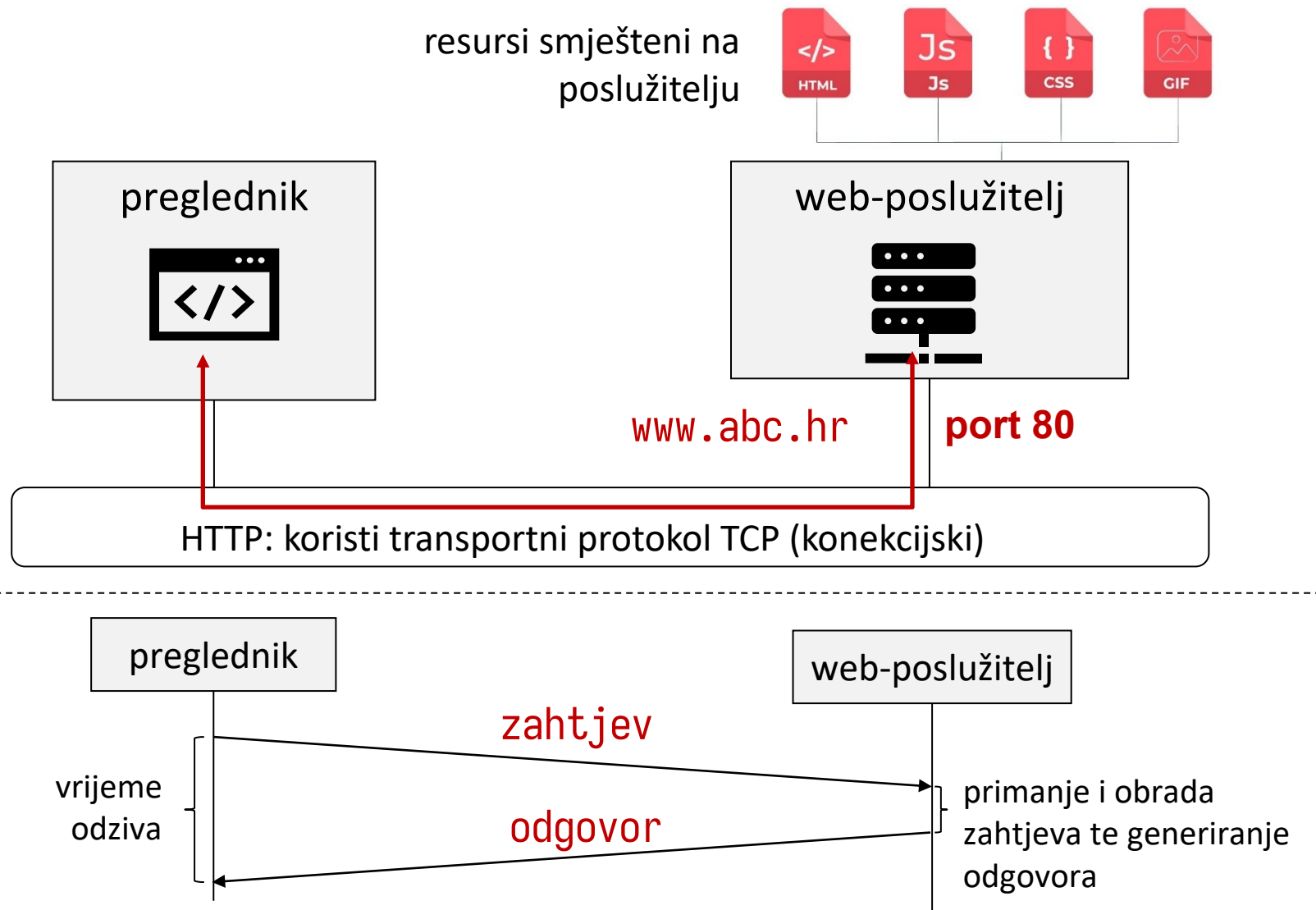
- Internetska aplikacija:
 - komunikacijski protokol aplikacijskog sloja - HTTP
 - jezik za označavanje HTML
 - standardi: *World Wide Web Consortium* (www.w3.org)
- Model klijent – poslužitelj
- Otvoreni sustav s transparentnim pristupom i konkurencijskom transparentnosti
- Transformacija weba:
 - “korisnik čita sadržaj” → “korisnik stvara sadržaj” (Web 2.0)
 - “korisnik odabire sadržaj” → “korisnik traži uslugu”:
web-usluga (engl. *Web Service*)

Hypertext Transfer Protocol (HTTP)

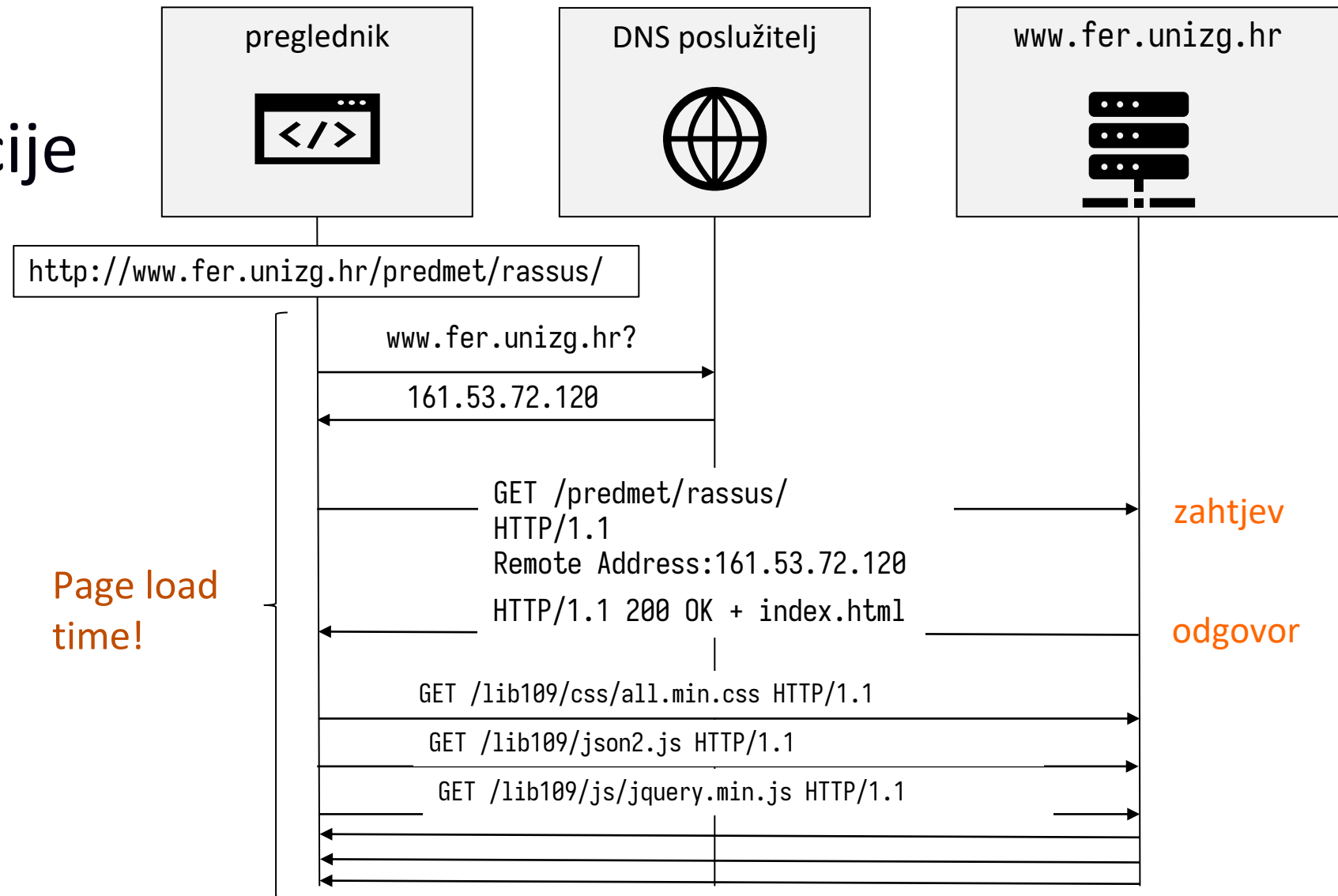
HTTP je **protokol** na aplikacijskom sloju

- protokol definira format i sadržaj poruka (zahtjeva i odgovora) te očekivano „ponašanje” poslužitelja, tj. pravila za generiranje odgovora na primljeni zahtjev
- **Zahtjev**: definira operaciju (tzv. metodu), oznaku resursa, verziju protokola, itd.
- **Odgovor**: rezultat (uspjeh, neuspjeh, pogreška, ..) opisan statusnim kôdom i sadržaj resursa (npr. datoteka HTML, CSS, JPEG...)
- Poslužitelj ne čuva stanje između dviju konverzacija s istim klijentom jer je HTTP **protokol bez očuvanja stanja (engl. *stateless protocol*)**

Podsjetimo se kako web izgleda ...



Primjer komunikacije



Transparentnost web-poslužitelja (1)

Lokacijska transparentnost:

- postiže se simboličkim imenima koja se u sustavu imenovanja domena (DNS) prevode u lokacije poslužitelja (mrežne adrese):
 - korisnik rabi simbolička imena, položaj poslužitelja (sjedišta weba) kao i bilo kojeg resursa ne treba biti i nije poznat korisniku

Migracijska transparentnost:

- ne mijenja se simboličko ime, već se samo mijenja lokacija poslužitelja (mrežna adresa) u DNS-u

Relokacijska transparentnost:

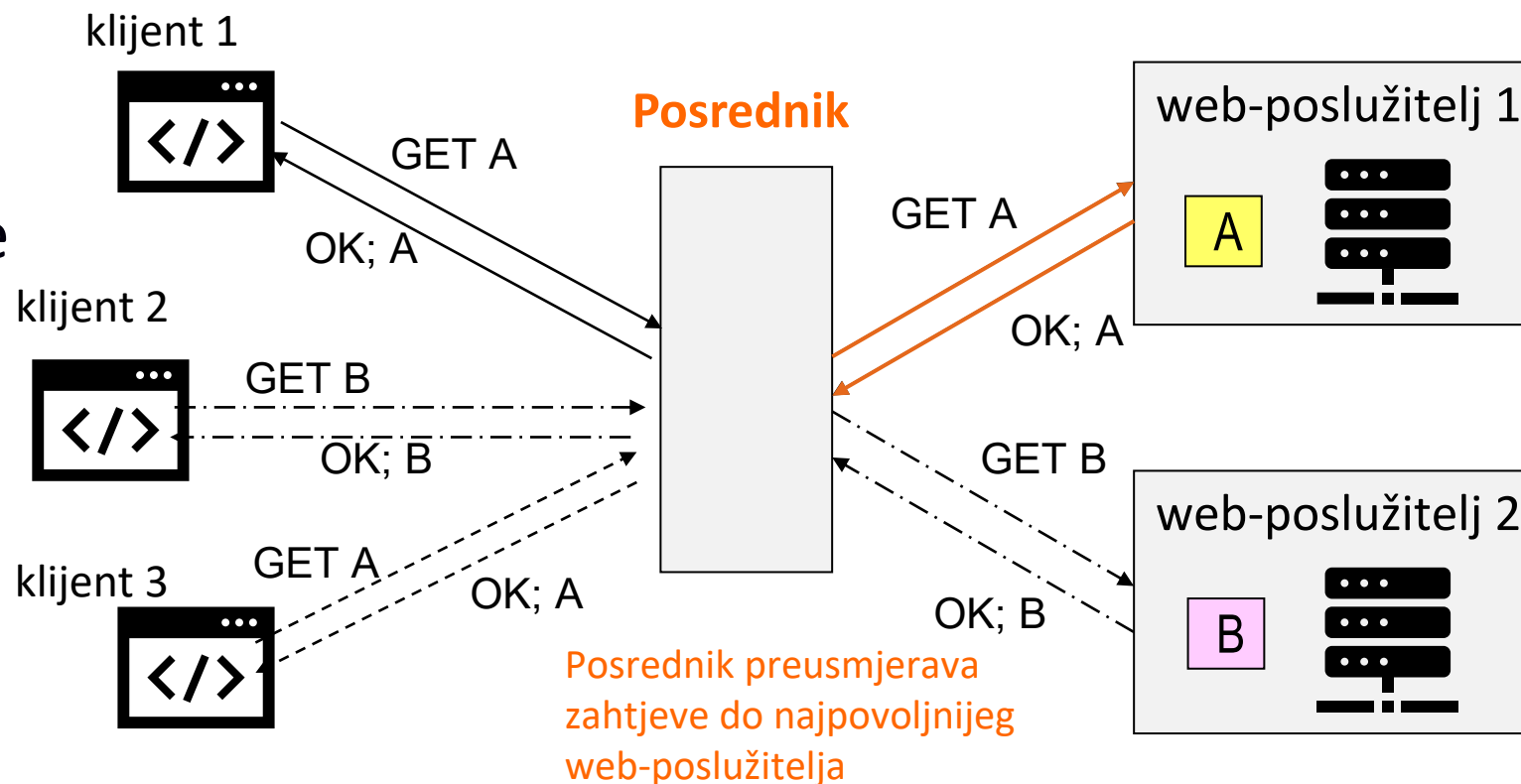
- ne zahtijeva se, poslužitelj je stacionaran i ne kreće se tijekom pružanja usluge

Transparentnost web-poslužitelja (2)

Replikacijska

transparentnost 1:

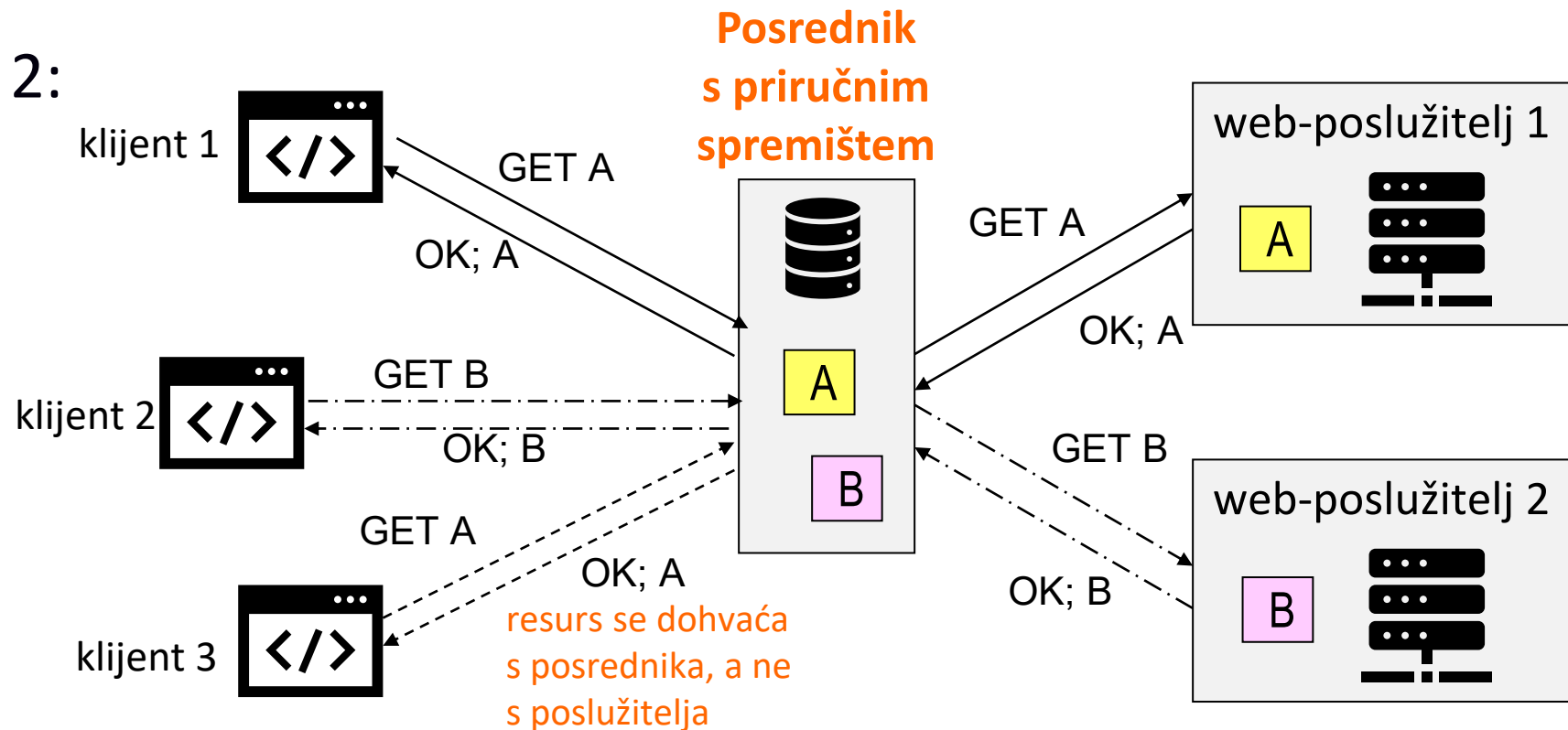
- Poslužiteljima se dostupa posredstvom posrednika (engl. *proxy*)



Transparentnost web-poslužitelja (3)

Replikacijska
transparentnost 2:

- Uvođenje
priručnog
spremišta



Sadržaj predavanja

- Definicija, obilježja i vrste raspodijeljenih sustava
- Zahtjevi na raspodijeljene sustave: otvorenost, transparentnost, skalabilnost i kvaliteta usluge
- Arhitektura raspodijeljenih sustava
- Primjeri modela raspodijeljene obrade
- **Studijski primjer: Internet stvari**

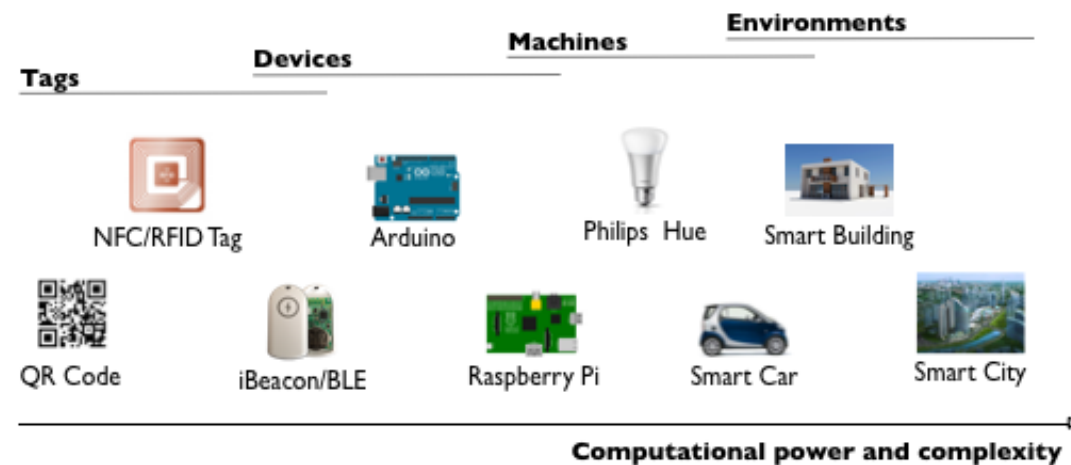
Uređaji i "stvari" postaju dio Interneta

Internet Connected
Object (ICO)



Što je "stvar"?

- Objekt iz fizičkog svijeta ili virtualnog digitalnog svijeta (virtualni objekt)
 - ima jedinstveni identifikator i povezan je na Internet (direktno ili putem posrednika), *Internet Connected Object* (ICO)
 - senzor: opažanje okoline, potencijalno kontinuirano generira podatke
 - aktuator: može izvršiti određene funkcije



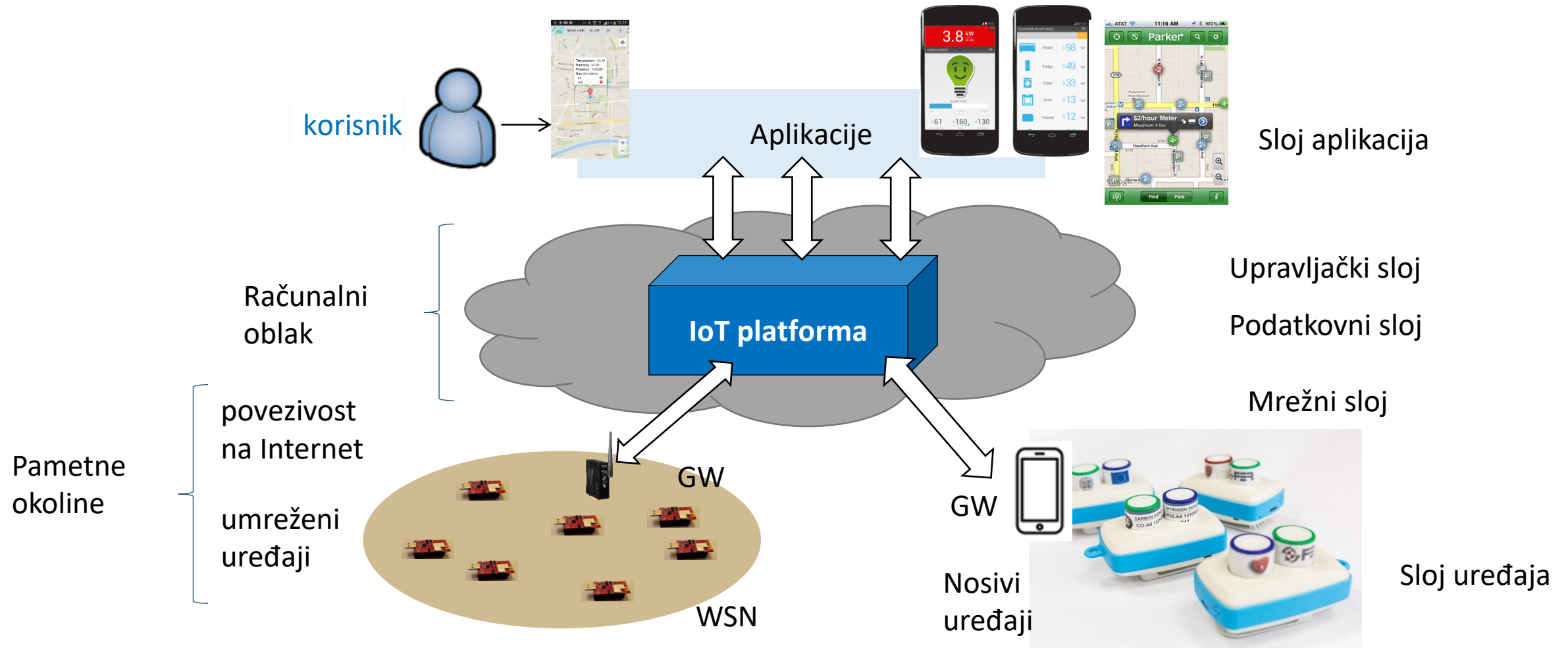
Source: Building the Web of Things: book.webofthings.io
Creative Commons Attribution 4.0

Internet of Things (IoT): definicija

ITU-T Recommendation Y.2060, 06/2012:

- *A global infrastructure for the information society, enabling **advanced services by interconnecting (physical and virtual) Things** based on, existing and evolving, interoperable information and communication technologies.*
 - *Through the exploitation of identification, data capture, processing and communication capabilities, the IoT makes **full use of things to offer services to all kinds of applications**, whilst ensuring that **security and privacy** requirements are fulfilled.*
 - *In a broad perspective, the IoT can be perceived **as a vision with technological and societal implications**.*

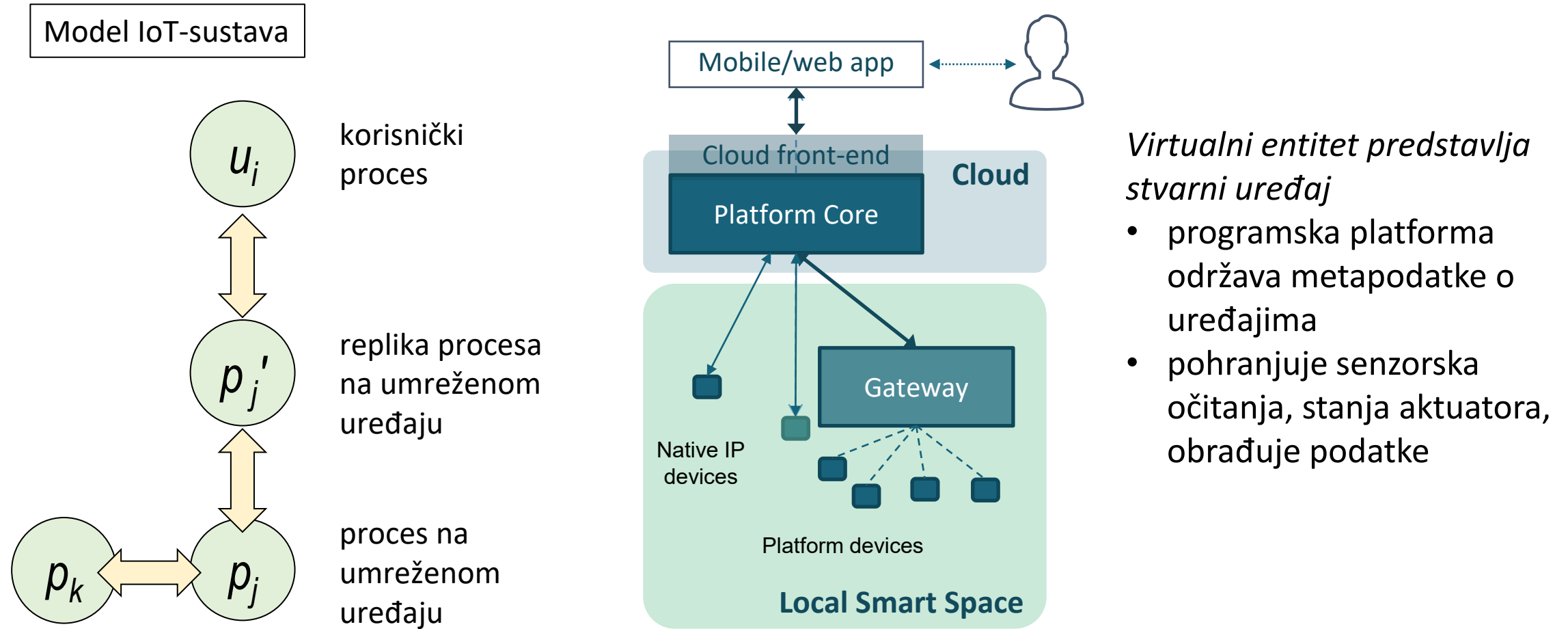
Pojednostavljena arhitektura Interneta stvari



Kako integrirati "stvari" i ponuditi aplikacije korisnicima?

- Pomoću programskih platformi (**IoT-platforma**) koje integriraju i upravljaju uređajima
 - raspodijeljeni sustav velikih razmjera
 - uređaji: često imaju vrlo ograničene resurse te su povezani na Internet putem prilaznog uređaja (engl. *gateway*)
 - potrebno je objediniti i na jedinstveni način zapisati podatke primljene iz različitih izvora
 - potreba za obradom velike količine podataka (često u stvarnom vremenu)
 - *Web of Things*: koncept koji povezuje uređaje direktno na WWW (tehnologije vezane uz protokol HTTP)

IoT-rješenje je složeni raspodijeljeni sustav



Interoperabilna IoT-platforma



<http://www.iot.fer.hr/>





Zadaci (1)

1. Čime je definirana otvorenost weba?
2. Kojim aspektima transparentnosti pridonosi sustav imenovanja domena (DNS)?
3. Kako replikacija pridonosi otpornosti na kvarove i skalabilnosti?
4. Kakvi bi se problemi pojavili kada bi se više repliciranih poslužitelja weba priključilo na mrežu izravno, a ne posredstvom zastupnika (*proxy*)?
5. Što sve utječe na vrijeme odziva poslužitelja weba?

Zadaci (2)

6. Na primjeru weba objasnite razliku između vertikalnog i horizontalnog skaliranja sustava.
7. Zašto se koriste perzistentne konekcije u HTTP-u?
8. Kako se definira uvjetni GET i kako funkcionira? Kakve prednosti donosi za performance HTTP-a?
9. Objasnite zašto današnja rješenja za IoT predstavljaju raspodijeljene sustave velikih razmjera?