

Distribuirani sistemi

Pojam distribuiranog sistema

- **Distribuirani sistem je skup nezavisnih računara koji svojim korisnicima izgleda kao jedinstven koherentan sistem**
 - Softver obezbeđuje da računari koji komuniciraju preko komunikacione mreže rade kao jedan sistem
 - Distribuirani sistem integriše razne aplikacije koje se izvršavaju na različitim računarima u jedan sistem
- **Karakteristična svojstva:**
 - Autonomni procesni elementi, poznati i kao **čvorovi** (engl. nodes), mogu biti hardverski uređaji ili softverski procesi
 - Jedinstven koherentan sistem: korisnici ili aplikacije imaju **doživljaj jedinstvenog sistema** ⇒ neophodna **saradnja čvorova**

“The Internet was done so well that most people think of it as a natural resource like the Pacific Ocean, rather than something that was man-made. When was the last time a technology with a scale like that was so error-free? The Web, in comparison, is a joke. The Web was done by amateurs.”

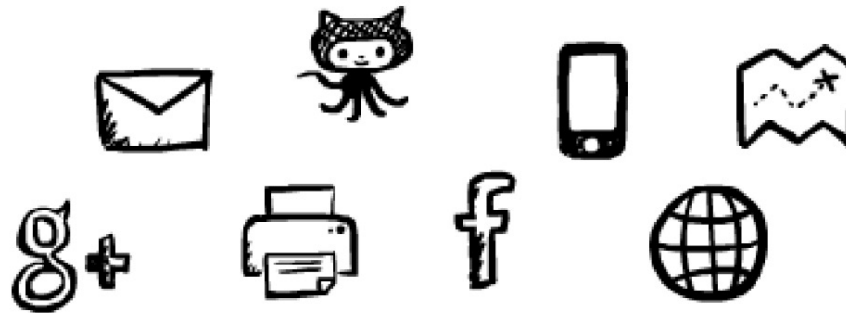


Alan Kay (1940-)

Izvor: <http://www.drdobbs.com/architecture-and-design/interview-with-alan-kay/240003442>

Primeri distribuiranih sistema

- Internet – razmena informacija
- Blokčejn – razmena vrednosti
- Računarstvo u oblaku (engl. *cloud computing*)
- Distribuirani fajl sistemi i skladišta podataka (Dropbox, Gdrive, IPFS)
- Takođe: mejl serveri, GPS, ATM, telefonija, senzorske mreže, git...



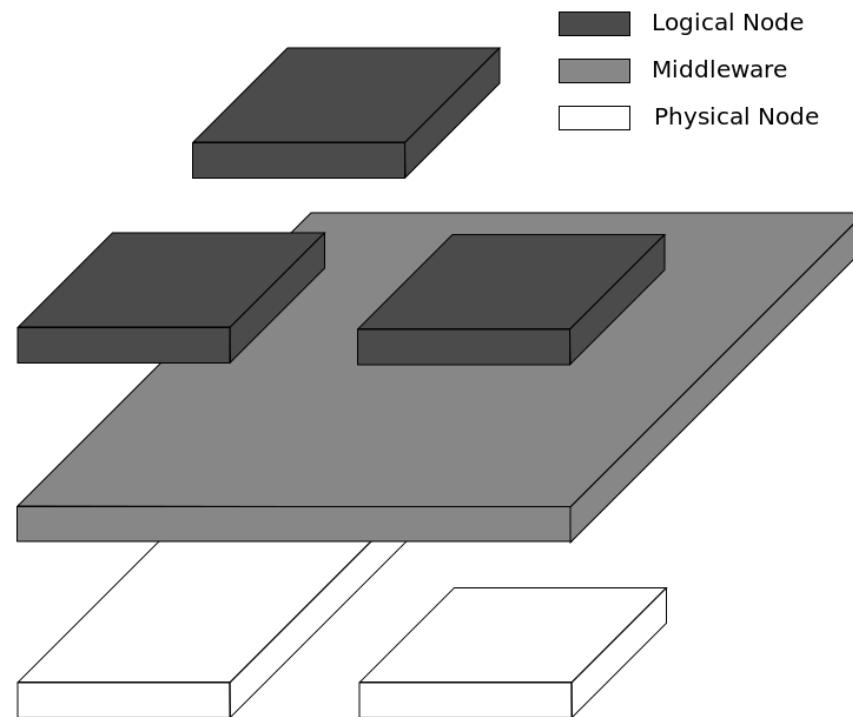
Izvor: http://www.csnedelja.mg.edu.rs/static/resources/v3.0/petl_distribuirani_nj.pdf

Osobine distribuiranih sistema

- Tri glavne karakteristike distribuiranih sistema:
 - **konkurentnost komponenti** (engl. *concurrency*)
 - U distribuiranom sistemu je dozvoljeno da više klijenata istovremeno pristupi istom resursu
 - **nepostojanje globalnog sata** (engl. *global clock*)
 - Komunikacija se zasniva samo na slanju poruka putem mreže
 - **nezavisan otkaz komponenti** (engl. *independent failures*)
 - Otkaz pojedinačnih komponenti neće uticati na rad sistema

Midlver (engl. *middleware*)

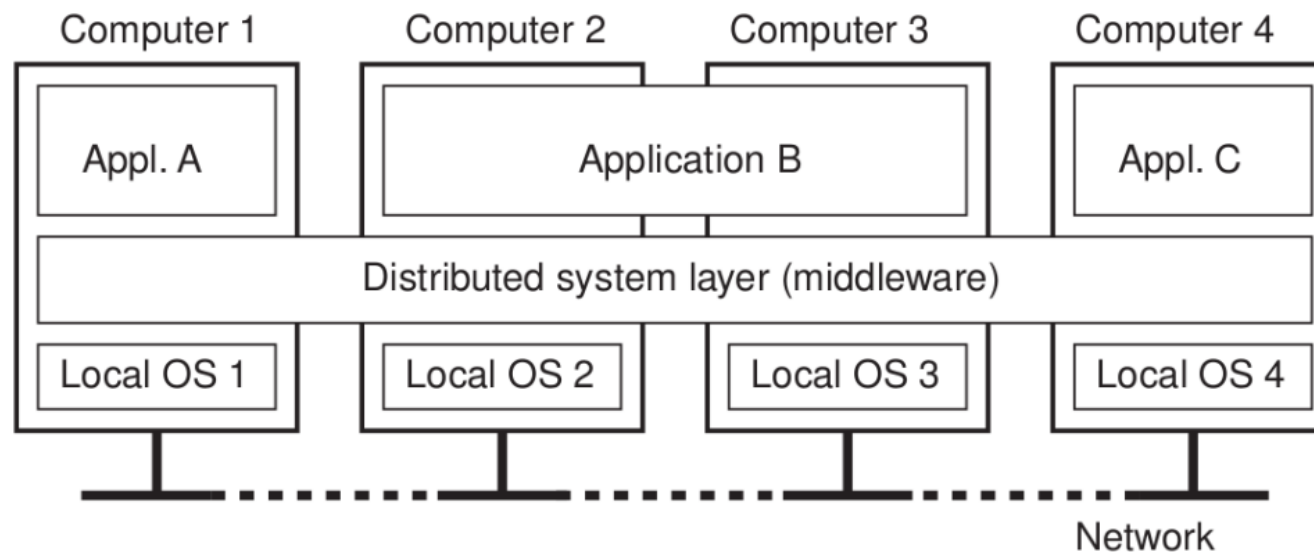
- Midlver je operativni sistem distribuiranih sistema
- Sadrži protokole, tj. najčešće korišćene komponente i funkcije i pruža svim aplikacijama jedinstven interfejs – softverski lepak



Izvor: <https://en.wikipedia.org/wiki/Middleware>

Prednosti distribuiranih sistema

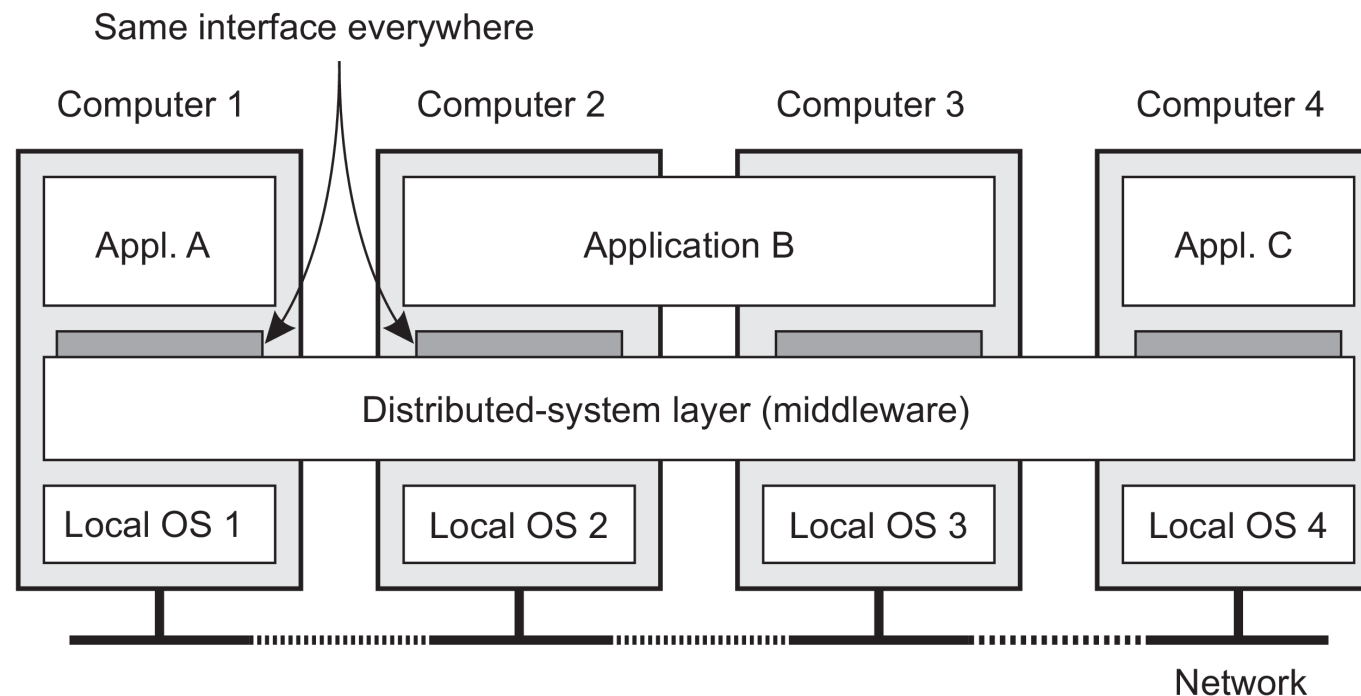
- **Razlike među računarima i način komunikacije sakriveni** su od korisnika
- Korisnici i aplikacije **interaguju** sa distribuiranim sistemom **na konzistentan i jednobrazan način** bez obzira na mesto i vreme interakcije



Izvor: <https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/>

Prednosti distribuiranih sistema

- Lako se proširuju
- Podržavaju heterogene računare i mreže
 - Midlver sloj se prostire na više mašina i svim aplikacijama nudi isti interfejs, posreduje u povezivanju aplikacija – softverska magistrala



Izvor: <https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/>

Nedostaci distribuiranih sistema

- U odnosu na centralizovane sisteme:
 - Softver je veoma složen
 - Umanjene su performanse zadataka koji se mogu obaviti na jednom računaru (zbog trajanja komunikacije)
 - Može biti smanjena je bezbednost sistema



Ciljevi razvoja distribuiranih sistema

1. Podržati **deljenje resursa**

- Omogućiti korisnicima da lako pristupaju i dele udaljene resurse
- Resursi mogu biti periferije, memorijski medijumi, podaci, fajlovi, servisi, mreže, itd.
- Olakšavaju saradnju i razmenu informacija – npr. Internet

2. Učiniti **distribuiranost neprimetnom**

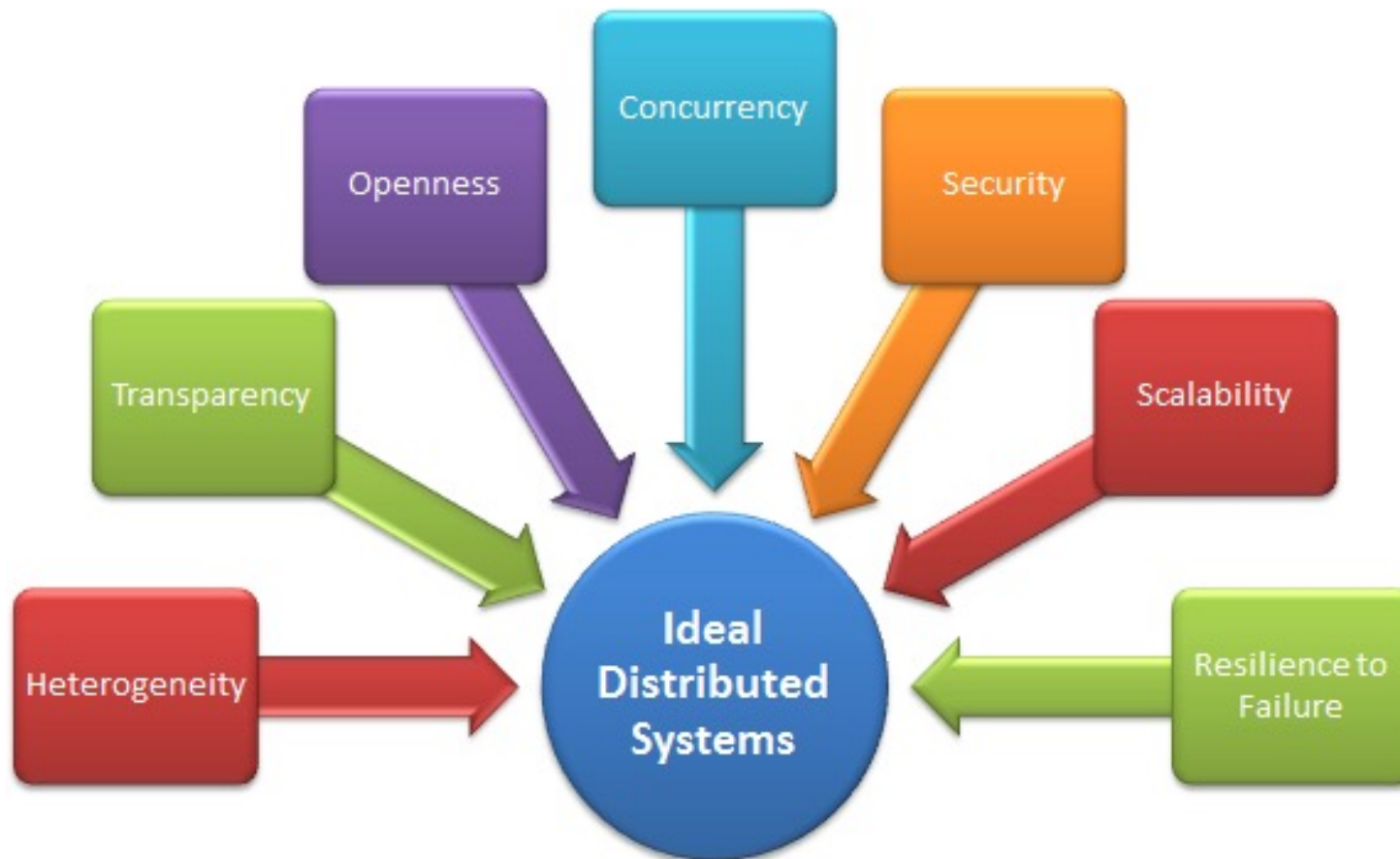
- Cilj je učiniti distribuiranost procesa i resursa neprimetnim, tj. nevidljivim za krajnje korisnike i aplikacije

3. Učiniti sistem **otvorenim**

- Sistem nudi komponente koja se lako koriste od i integrišu u druge sisteme, interoperabilnost, proširivost, odvajanje pravila od mehanizama

4. Omogućiti **skalabilnost** sistema

Osobine idealnog distribuiranog sistema



Izvor: <http://www.ejbtutorial.com/distributed-systems/challenges-for-a-distributed-system>

Transparentnost

- **Sakriti distribuiranost procesa i resursa**
 - Pristup – sakriti razlike u reprezentaciji podataka i kako se pristupa resursu
 - Lokacija – sakriti gde se resurs nalazi
 - Migracija – sakriti da se resurs može prebaciti na drugu lokaciju
 - Relokacija – sakriti da se resurs može prebaciti na drugu lokaciju dok je u upotrebi
 - Replikacija – sakriti da se resurs može kopirati na više mesta
 - Konkurentnost – sakriti da se resurs može takmičiti više korisnika
 - Greške – sakriti otkaz i oporavak resursa
 - Perzistentnost – sakriti da je resurs u memoriji ili na disku

Otvorenost

- Omogućiti interakciju sa servisima iz drugih otvorenih sistema, nezavisno od okruženja:
 - Sistemi treba da su **usaglašeni sa dobro definisanim interfejsima**
 - Sistemi treba da **lako međusobno sarađuju**
 - Sistemi treba da podrže **prenosivost aplikacija**
 - Sistemi treba da budu **lako proširivi**
- Implementacija otvorenosti – pravila:
 - Koji nivo konzistentnosti se traži za podatke koji se keširaju na klijentu?
 - Koje operacije dozvoljavamo da izvršava preuzeti programski kod?
 - Koje zahteve po pitanju kvaliteta usluge podešavamo kada se suočavamo sa promenljivim propusnim opsegom?
 - Koje nivo tajnosti zahteva komunikacija?
- Implementacija otvorenosti – mehanizmi:
 - Dozvoliti (dinamičko) podešavanje pravila keširanja
 - Podržati različite nivoe poverenja za programski kod
 - Pružiti podesive parametre kvaliteta usluge (engl. *Quality of service* – QoS) na nivou pojedinačnih tokova podataka (engl. *data stream*)
 - Ponuditi različite enkripcione algoritme

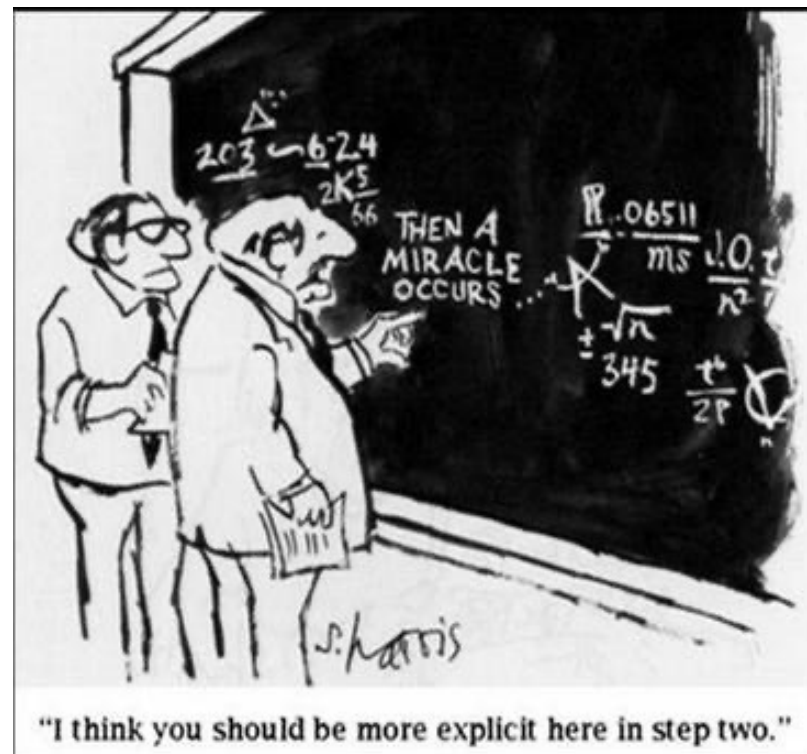
Skalabilnost

- Za mnoge moderne distribuirane sisteme olako se koristi pridev “skalabilni” bez jasnog objašnjenja zašto se sistem smatra skalabilnim
- Skalabilnost se može meriti po najmanje tri osnova:
 - **Skalabilnost veličine** (engl. *size scalability*) : broj korisnika i/ili procesa
 - **Geografska skalabilnost** (engl. *geographical scalability*): maksimalna udaljenost između čvorova
 - **Administrativna skalabilnost** (engl. *administrative scalability*): broj administrativnih domena
- Većina sistema se, u određenoj meri, nose sa skalabilnošću veličine.
 - Tipično rešenje: više moćnih servera koji rade nezavisno i paralelno. Danas je glavni izazov u omogućavanju geografske i administrativne skalabilnosti

Pogrešne pretpostavke (engl. *pitfalls*)

Mnogi distribuirani sistemi su nepotrebno kompleksni usled grešaka koje zahtevaju naknadne ispravke. Prema Deutsch-u, u česte pogrešne pretpostavke spadaju:

- Mreža je pouzdana
- Mreža je sigurna
- Topologija se ne menja
- Ne postoji kašnjenje
- Protok je beskonačan
- Cena transporta je nula
- Postoji jedan administrator
- Mreža je homogena



Izvor: <http://www.stufffundieslike.com/2011/05/the-false-premise/>

scenes from distributed systems

a "linearizable" system

(like etcd)

☺ I can has answer?

NO. can't you see we're having a leader election?

☹ ☹ ☹ ☹

an eventually consistent system

(like DNS)

☺ What's the IP for julia.com

right

1.1.1.1 ☺

2.2.2.2 ☺

... 2 hours later ...

Um, 1.1.1.1

(you can always get an answer though!)

replication is hard

☺ just copying data from my primary
secondary
hate to tell you but uh the primary changed 5 minutes ago
OMG ☹

the network is fine BUT

☺ hi
actually
I'm gonna garbage collect for 2 minutes no reply for you!
☹

clocks lie

☺ it's 5pm ☺ it's 5pm
... 2 minutes later ...
5:02 5:17

just got paged
☹ why?

with 1000s of machines... it gets weird

normal { ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺
☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺
☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺ ☺

☺ you would not BELIEVE what I'm doing right now

Izvor: <https://drawings.jvns.ca/distributed-systems/>

Klase distribuiranih sistema

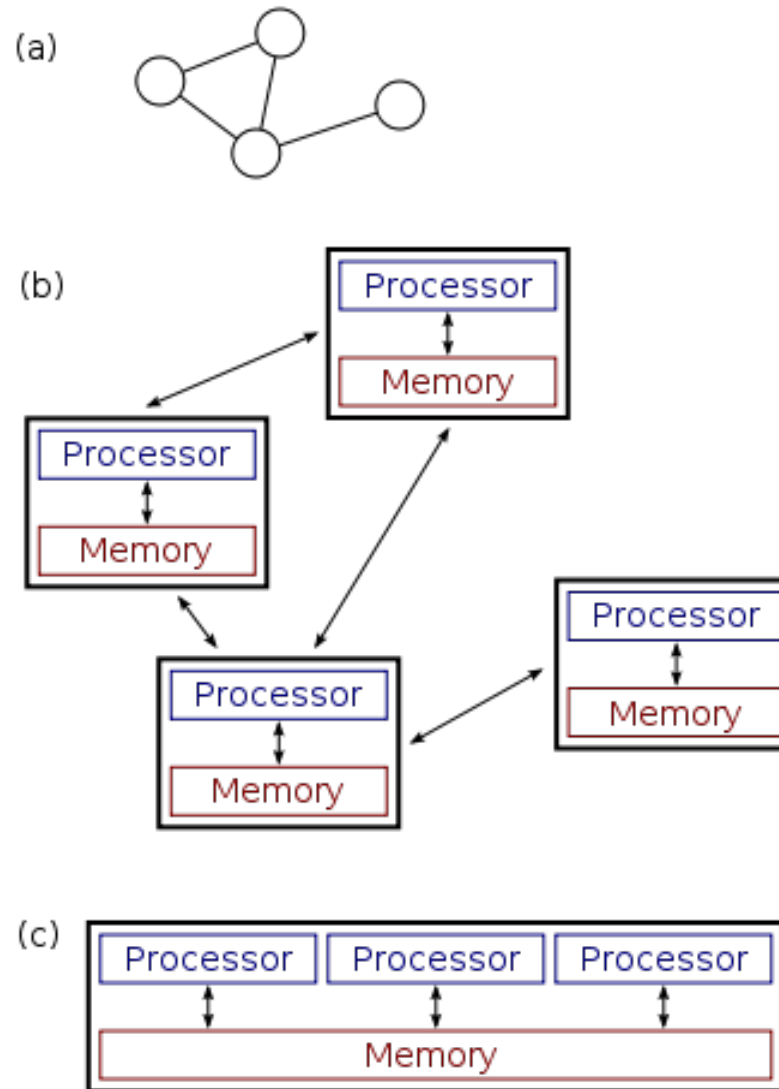
Tipovi paralelnih i distribuiranih sistema

- **Sistemi sa deljenom memorijom** (engl. *shared memory*)
 - Teže se implementiraju u hardveru
 - Lakše se programiraju
 - Neki autori ih posmatraju **samo kao paralelne sisteme**
- **Sistemi sa slanjem poruka** (engl. *message passing*)
 - Lakše se implementiraju u hardveru
 - Teže se programiraju



Tipovi arhitektura sistema

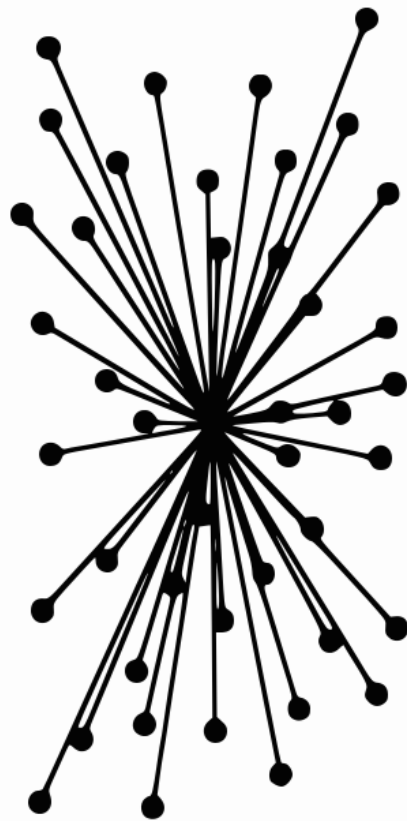
Slanje poruka



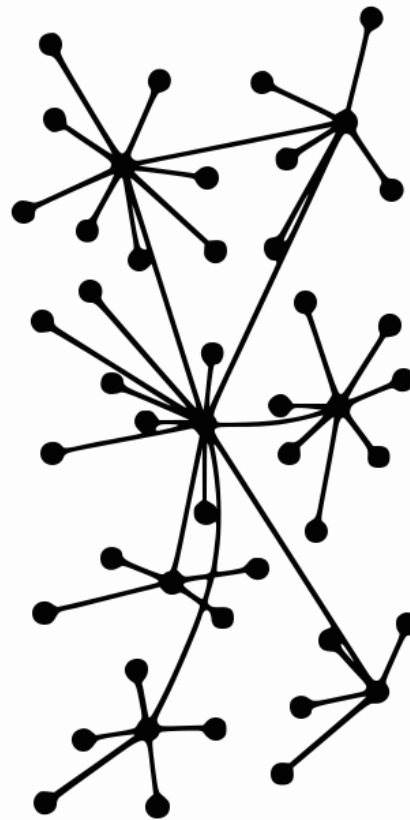
Deljena memorija

Izvor: http://cs312.osuosl.org/slides/22_distributed_systems.html#1

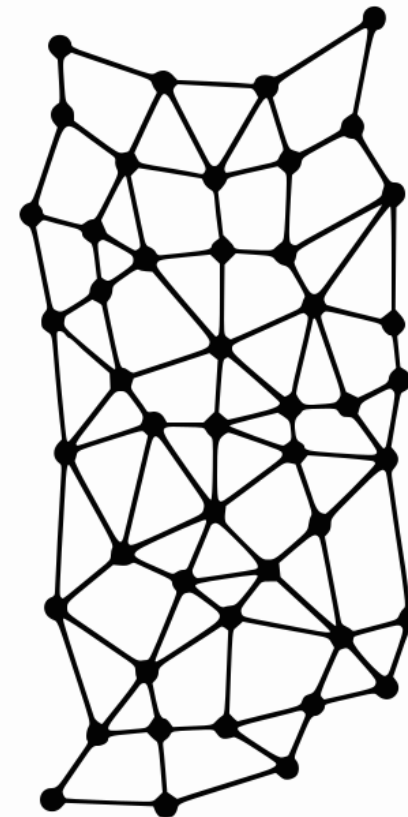
Organizacija sistema



Centralized



Decentralized



Distributed

Izvor: Centralized, decentralized and distributed network models, Paul Baran (1964)

Formalni opis konkurentnih sistema

- CSP (engl. *Communicating Sequential Processes*) – Tony Hoare 1978.
 - **Formalni jezik za opis obrazaca interakcije u konkurentnim sistemima**
 - Vrsta procesne algebra (računa) bazirana na slanju poruka putem kanala
 - Primenjen za specifikaciju i verifikaciju konkurentnih aspekata bezbedonosno-kritičnih sistema, implementiran u jeziku Go
 - Opis dozvoljenih kombinacija procesa i događaja u BNF notaciji

Formalni opis konkurentnih sistema

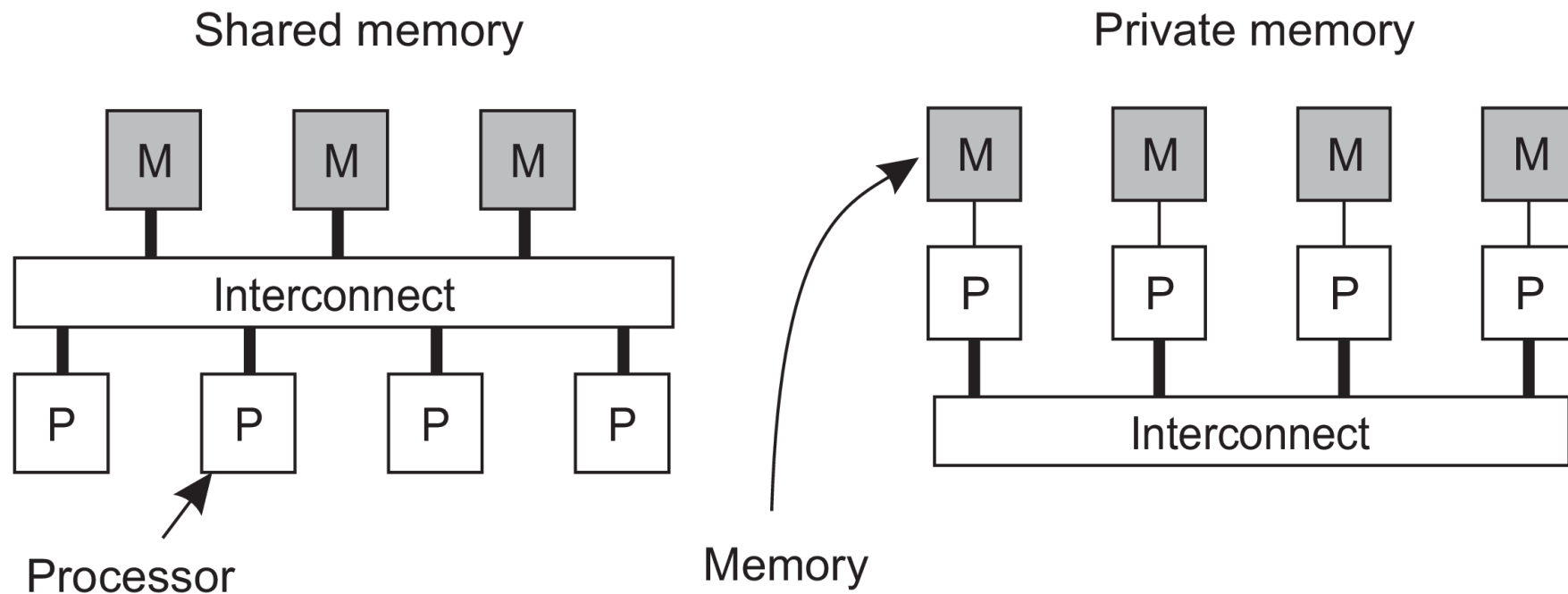
- CCS (*Calculus of Communicating Systems*) – Robin Milner 1980.
 - Formalni jezik uključuje primitive za paralelno slaganje, izbor između akcija i ograničavanje opsega
- Pi-calculus (račun) – Robin Milner, Turing kompletan model
 - Model izračunavanja za konkurentne sisteme
 - Dozvoljava da kanali međusobno šalju svoja imena, na ovaj način omogućava opis konkurentnih izračunavanja u kojima konfiguracija mreže može da se menja tokom izvršavanja
 - Sintaksa pi računa dozvoljava: predstavljanje procesa, sinhronu komunikaciju između procesa preko kanala, stvaranje novih kanala i replikaciju procesa

Klase distribuiranih sistema

- **Distribuirani računarski sistemi (visokih performansi)**
(engl. *high performance distributed computing systems*)
 - Klasteri
 - Grid računarstvo
 - Računarstvo u oblaku (engl. *cloud computing*)
- **Distribuirani informacioni sistemi** (engl. *distributed information systems*)
- **(Distribuirani sistemi za) prožimajuće računarstvo**
(engl. *distributed systems for pervasive computing*)

Paralelno računarstvo

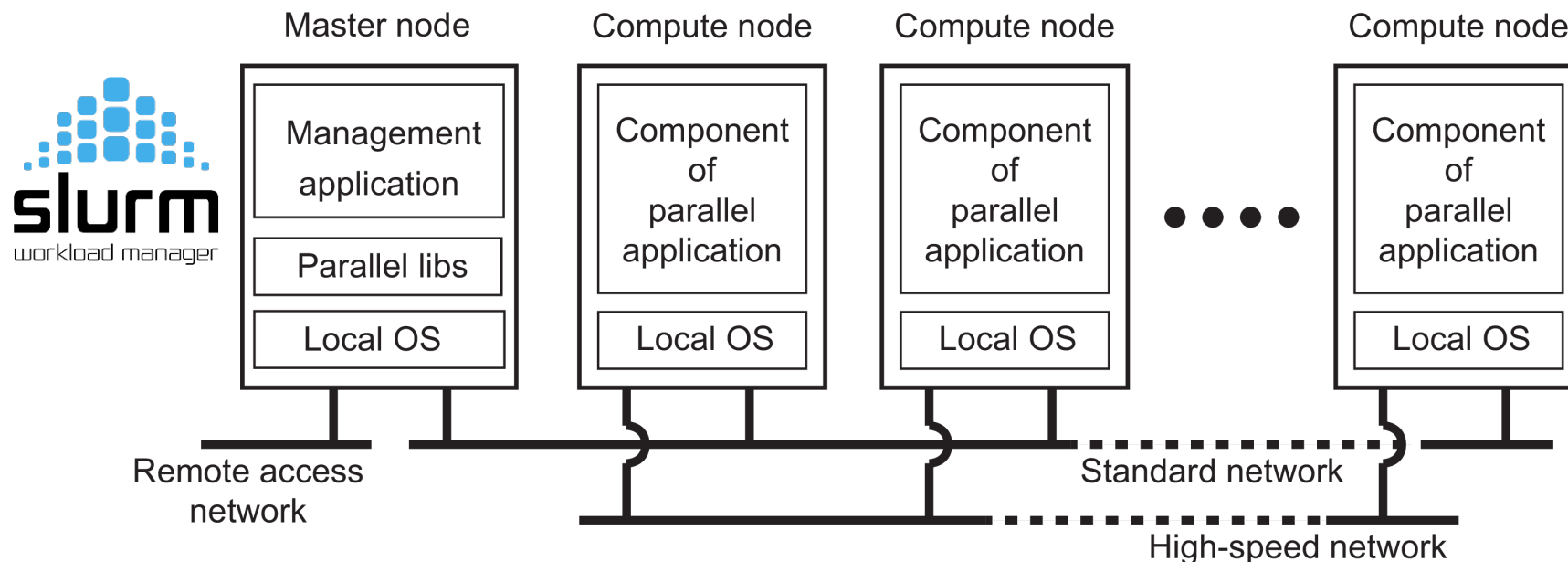
- Računarski sistemi visokih performansi nastali su kao paralelni multiprocesorski računari
- Multiprocesor/višejezgarni procesor naspram multiračunara



Izvor: <https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/>

Klasteri

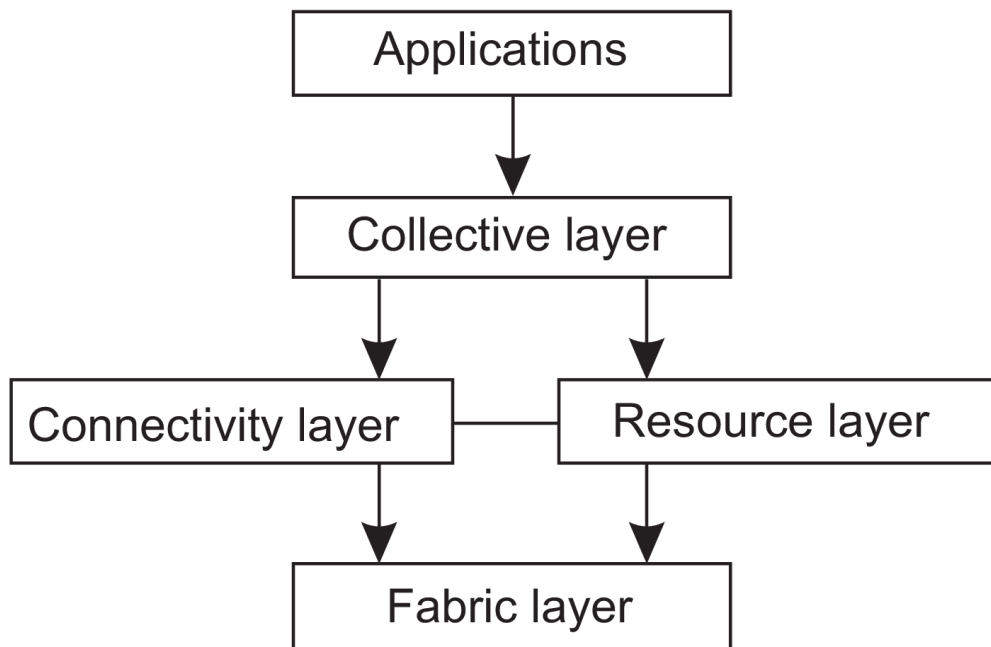
- U suštini skup “klasičnih” računara visokih performansi povezanih putem LAN (engl. *Local Area Network*)
 - Homogen: isti operativni sistem, skoro identičan hardver
 - Jedan upravljački čvor



Izvor: <https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/>

Grid računarstvo

- Viši stepen distribuiranosti – puno čvorova na raznim lokacijama
 - Heterogeni, razmešteni u okviru više organizacija, povezani WAN
 - Koriste virtuelne organizacije (grupisanje korisnika) kako bi upravljali autorizacijom dodele resursa



Slojevi u grid računarstvu

Fabric: pruža interfejs ka lokalnim resursima (radi upita stanja i mogućnosti, zaključavanja resursa, itd.)

Connectivity: komunikaciono/transakcioni protokoli, npr. za prenos podataka između resursa. Uključuje različite autentifikacione protokole

Resource: upravlja jednim resursom, kao što je kreiranje procesa ili čitanje podataka

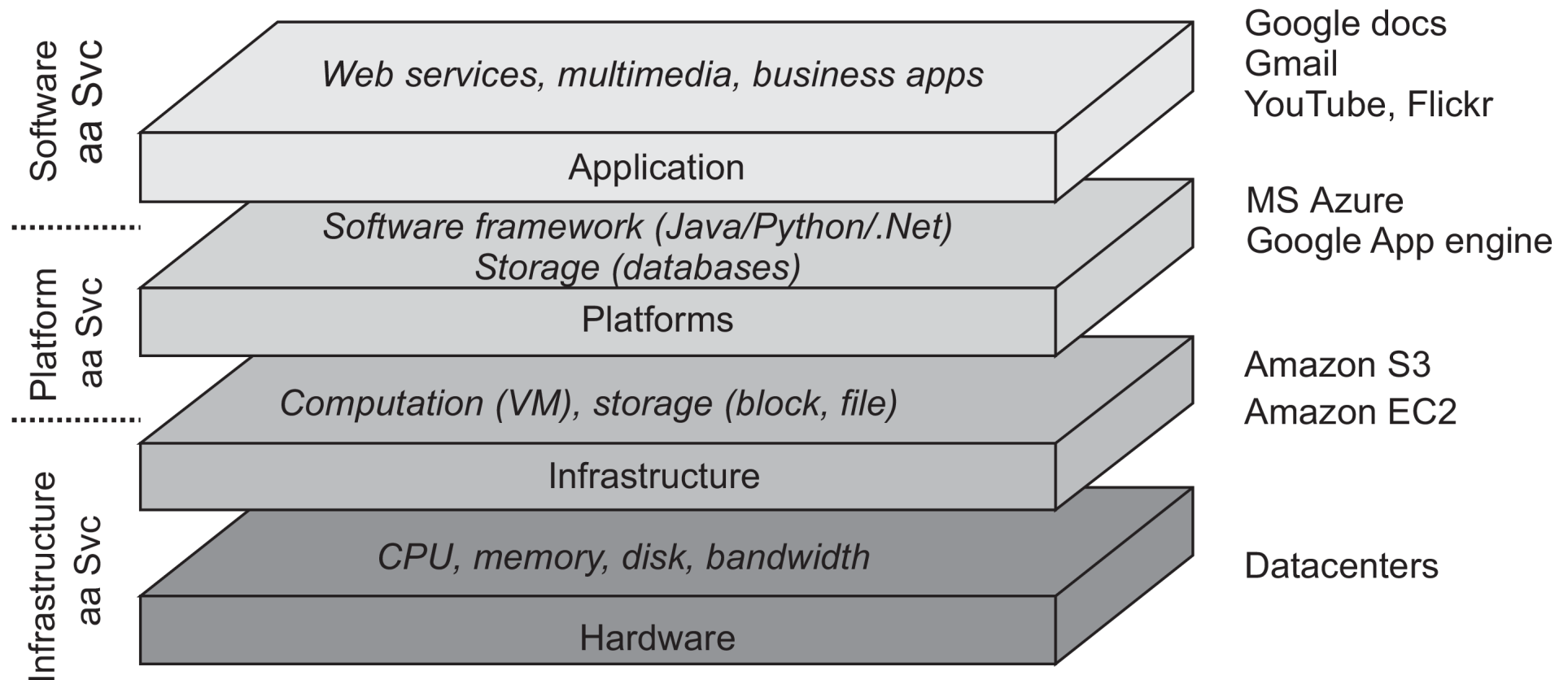
Collective: upravlja pristupom ka više resursa: otkrivanje, planiranje, replikacija

Application: sadrži same grid aplikacije u jednoj organizaciji.

Izvor: <https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/>

Računarstvo u oblaku

- Utility computing – klijent šalje posao u data centar, plaća prema resursima
- Pruža lako upotrebljiv i dostupan skup virtuelizovanih resursa



Izvor: <https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/>

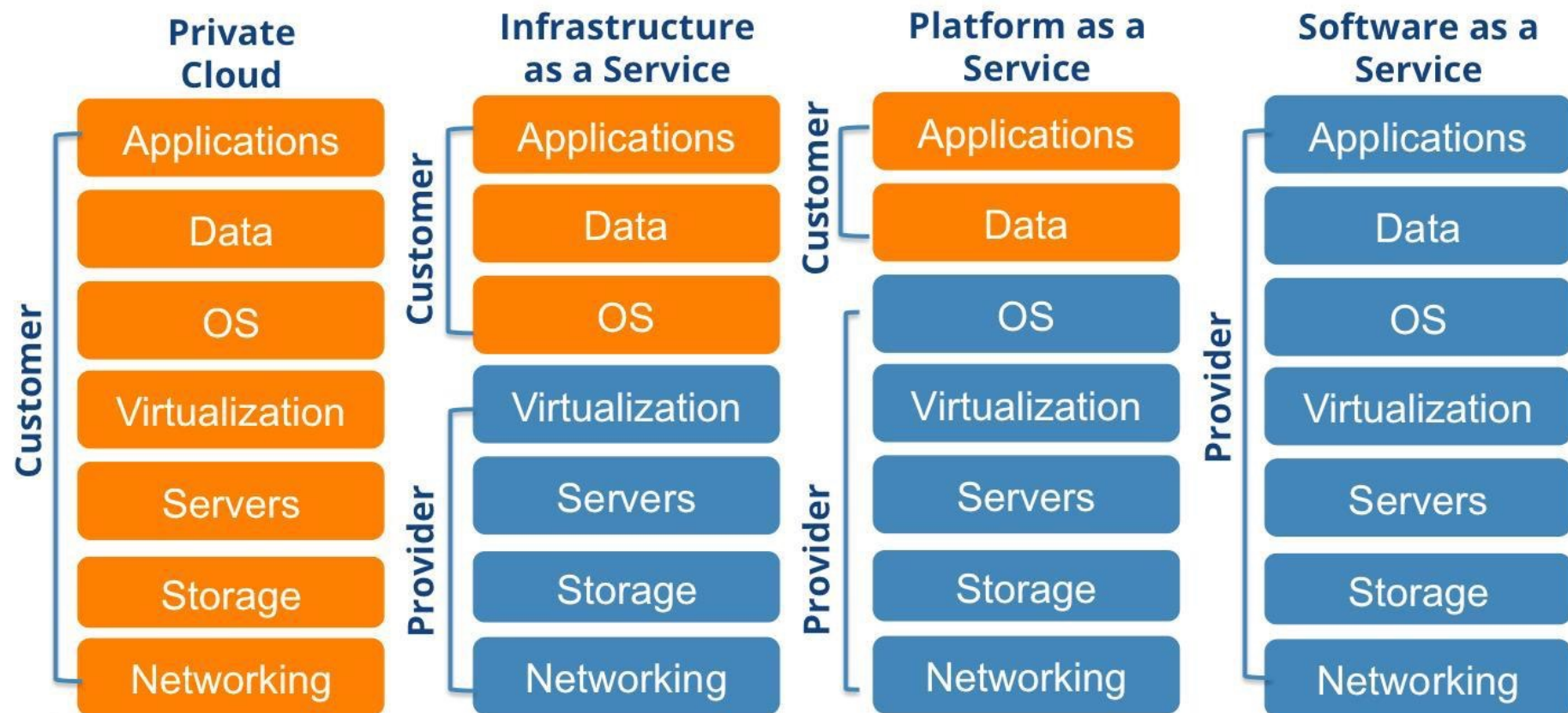
Računarstvo u oblaku

- U praksi se računarski oblaci organizuju u četiri sloja:
 - **Hardverski:** procesori, memorija, ruteri, napajanje i sistemi za hlađenje. Klijenti nemaju pristup ovom delu sistema.
 - **Infrastrukturni:** koristi virtuelizacione tehnike. Razvija se oko alokacije i upravljanja virtuelnim uređajima za čuvanje podataka i virtuelnim serverima.
 - **Platformski:** pruža apstrakcije višeg nivoa za čuvanje podataka i slične primene. Primer: Amazon S3 sistem za čuvanje podataka pruža API pomoću koga se (lokalno kreirani) fajlovi mogu organizovati i čuvati u tzv. koficama (engl. *buckets*).
 - **Aplikacioni:** same aplikacije, kao što su kancelarijski alati (tekst procesori, tabelarna izračunavanja, izrada prezentacija). Uporediv sa skupom aplikacija koje dolaze sa operativnim sistemima.

Izvor: <https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/>

Računarstvo u oblaku (engl. *cloud computing*)

- Tri različita tipa servisa: IaaS, PaaS, SaaS



Izvor: <http://www.smartcloudcomputing.net/2017/09/09/the-cloud-models-demystified/>

Distribuirani informacijski sistemi

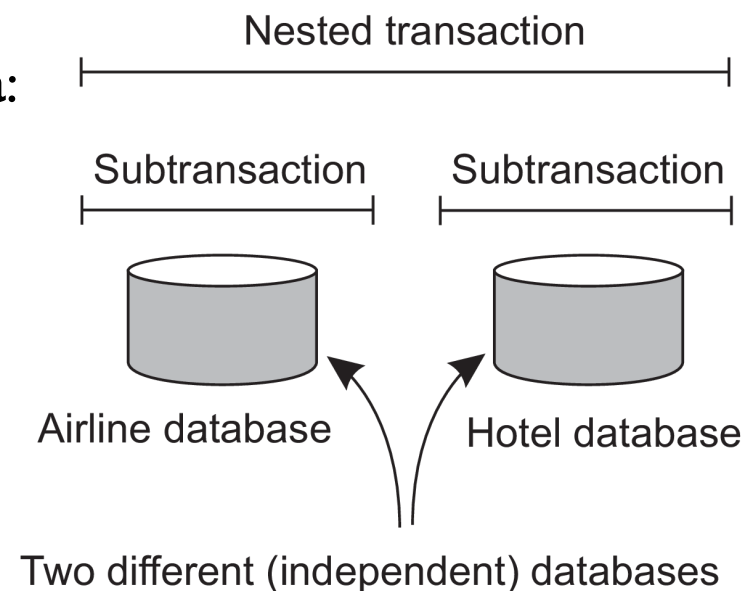
- Situacija:
 - Organizacije suočene sa tim da u radu koriste mnogo mrežnih aplikacija, s toga je postizanje interoperabilnosti teško
- Osnovni pristup:
 - Mrežna aplikacija se izvršava na serveru čime su njene usluge (servisi) dostupni udaljenim **klijentima**. Jednostavna integracija: klijenti kombinuju zahteve za (različite) aplikacije u jedan zahtev, šalju ga, zahtev se izvršava kao **distribuirana transakcija** i potom se predstavlja koherentan rezultat klijentu – glavna ideja: **ili se izvrše svi ili nijedan zahtev**
- Sledeći korak:
 - Omogućiti direktnu komunikaciju između aplikacija što dovodi do pojma **integracije poslovnih aplikacija** (engl. *Enterprise Application Integration* – EAI)

Primer EAI: ugnježdene transakcije

- Operacije sa bazom u vidu **transakcija**
 - Zahtevaju specijalne primitive: BEGIN_TRANSACTION, END_TRANSACTION, ABORT_TRANSACTION, READ, WRITE...
 - Glavno svojstvo transakcije: **ili se izvrše sve operacije ili nijedna**
 - **Ugnježdene transakcije** omogućavaju distribuiranje transakcije na više računara

- Transakcije podležu **ACID** pravilima:

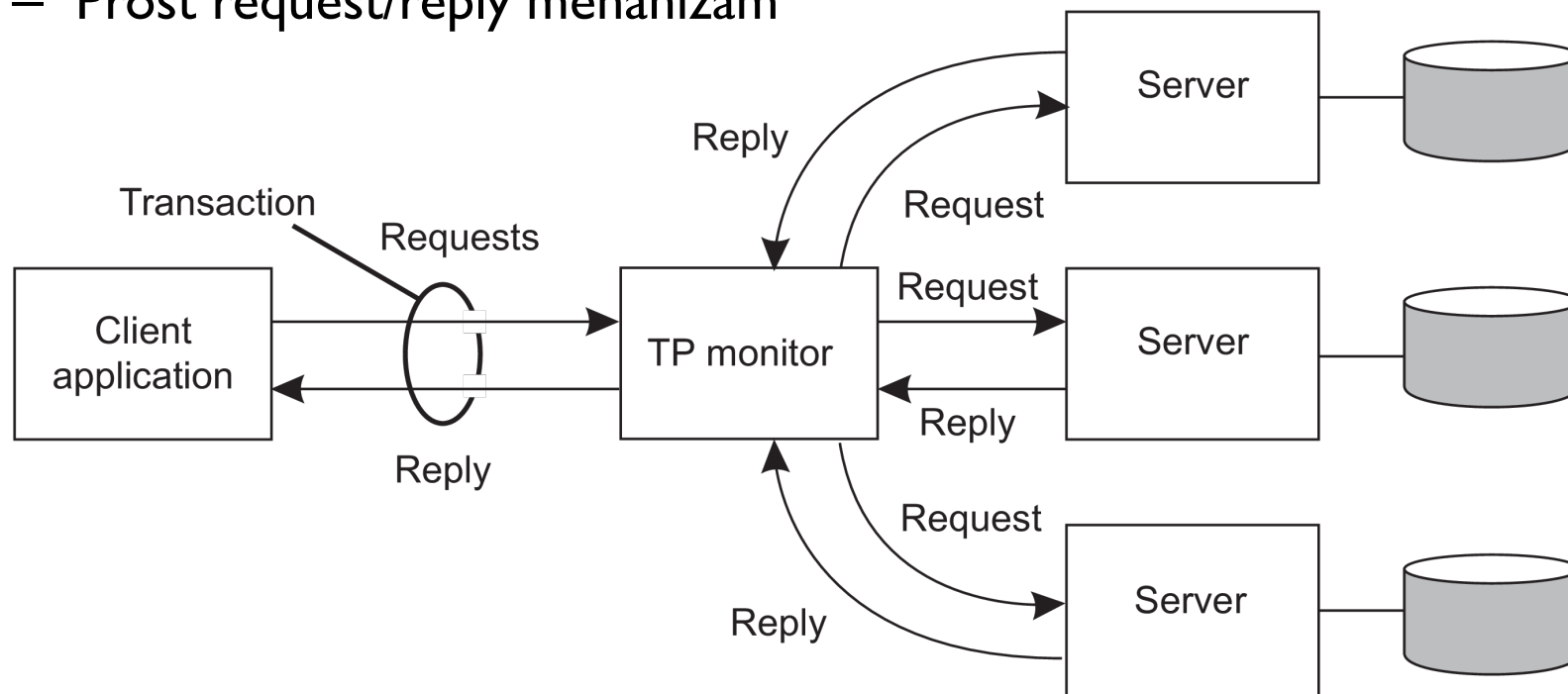
- **Atomic**: spoljni svet ima utisak da se dešavaju nedeljivo
- **Consistent**: transakcija ne krši sistemske invarijante
- **Isolated**: konkurentne transakcije ne smetaju jedna drugoj
- **Durable**: komit transakcije označava da su promene trajne



Izvor: <https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/>

TPM: Transaction Processing Monitor

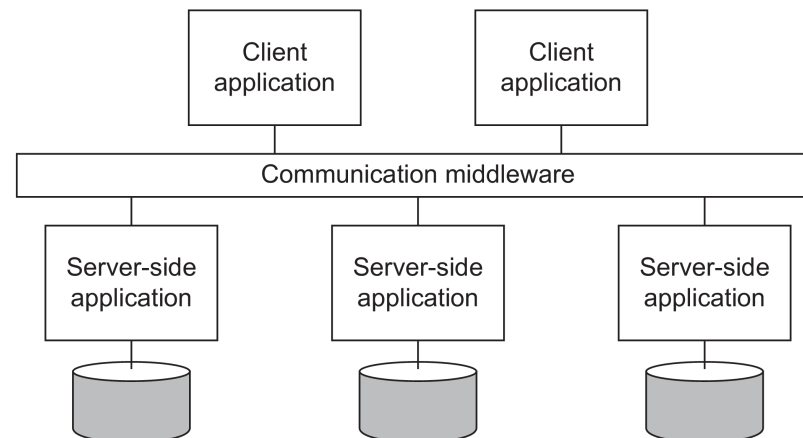
- TPM je prvo rešenje za koordinaciju izvršavanja ugnježdenih transakcija u distribuiranim sistemima
 - Protokol distribuirani komit
 - Prost request/reply mehanizam



Izvor: <https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/>

Midlver i EAI

- Savremeni midlver DS pruža komunikaciona sredstva za integraciju:
 - **Remote Procedure Call (RPC)**: zahtevi (engl. *requests*) se šalju pozivom lokalne procedure, pakuju se u poruke, obrađuju, odgovor se šalje kao poruka i rezultat se dobija kao povratna vrednost poziva
 - RMI (Remote Method Invocation) – isto kao RPC, samo sa objektima umesto funkcija
 - Mana: caller i callee moraju da rade u isto vreme i da znaju tačno kako da se obraćaju
 - **Message-Oriented Middleware (MOM)**: poruke se šalju logičkoj tački kontakta, tj. objavljuju se i potom prosleđuju pretplaćenim (engl. *subscribed*) aplikacijama, objavi/pretplati se (engl. *publish/subscribe*) sistemi



Izvor: <https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/>

Mehanizmi integracije aplikacija

- **Prenos fajla** (engl. *file transfer*): tehnički prost, nije fleksibilan:
 - Utvrditi format i strukturu fajla (XML, JSON, ...)
 - Rešiti upravljanje fajlovima
 - Propagacija ažuriranja i notifikacije
- **Deljena baza** (engl. *shared database*): mnogo fleksibilniji, ali zahteva zajedničku shemu podataka i predstavlja potencijalno usko grlo (engl. *bottleneck*)
- **RPC**: efikasan kada je neophodno izvršavanje serije akcija
 - Omogućava aplikaciji A da koristi informacije dostupne samo aplikaciji B, bez da A dobije direktan pristup tim informacijama
- **Slanje poruka** (engl. *messaging*): RPC traži da se pozivajuća (caller) i pozvana (callee) procedura izvršavaju istovremeno. Slanje poruka omogućava razdvajanje u vremenu i prostoru

Distribuirani prožimajući računarski sistemi

- Rastuća sledeća generacija distribuiranih sistema kod kojih su **čvorovi mali, mobilni i vrlo često ugrađeni u veće sisteme**, karakteriše ih činjenica da se **računarski sistem prirodno uklapa u korisničko okruženje**
- Tri (preklapajuće) klase:
 - **Svepristuni** (engl. *ubiquitous*) **računarski sistemi**: prožimajući i kontinualno prisutni, tj. postoji kontinualna interakcija između sistema i korisnika. Računarstvo se pojavljuje bilo kada i bilo gde
 - **Mobilni računarski sistemi**: prožimajući, ali sa naglaskom na činjenicu da su uređaju inherentno prenosivi
 - **Senzorske (i aktuatorske) mreže**: prožimajući, ali sa naglaskom na stvarno (kolaborativno) osećanje (engl. *sensing*) i dejstvo (engl. *actuation*) na okruženje

Sveprisutni računarski sistemi

- Glavne karakteristike **sveprisutnih računarskih sistema**:
 1. **Distribuiranost** (engl. *distribution*) – uređaji su umreženi, distribuirani i pristupačni na transparentan način
 2. **Interakcija** (engl. *interaction*) – interakcija između korisnika i uređaja je krajnje neprimetna
 3. **Razumevanje konteksta** (engl. *context awareness*) – sistem je „svestan“ korisničkog konteksta kako bi optimizovao interakciju
 4. **Autonomija** (engl. *autonomy*) – uređaji rade autonomno bez ljudske intervencije te su s toga visoko samo-upravljivi
 5. **Inteligencija** (engl. *intelligence*) – sistem kao celina može da prihvati širok spektar dinamičkih akcija i interakcija

Mobilni računarski sistemi

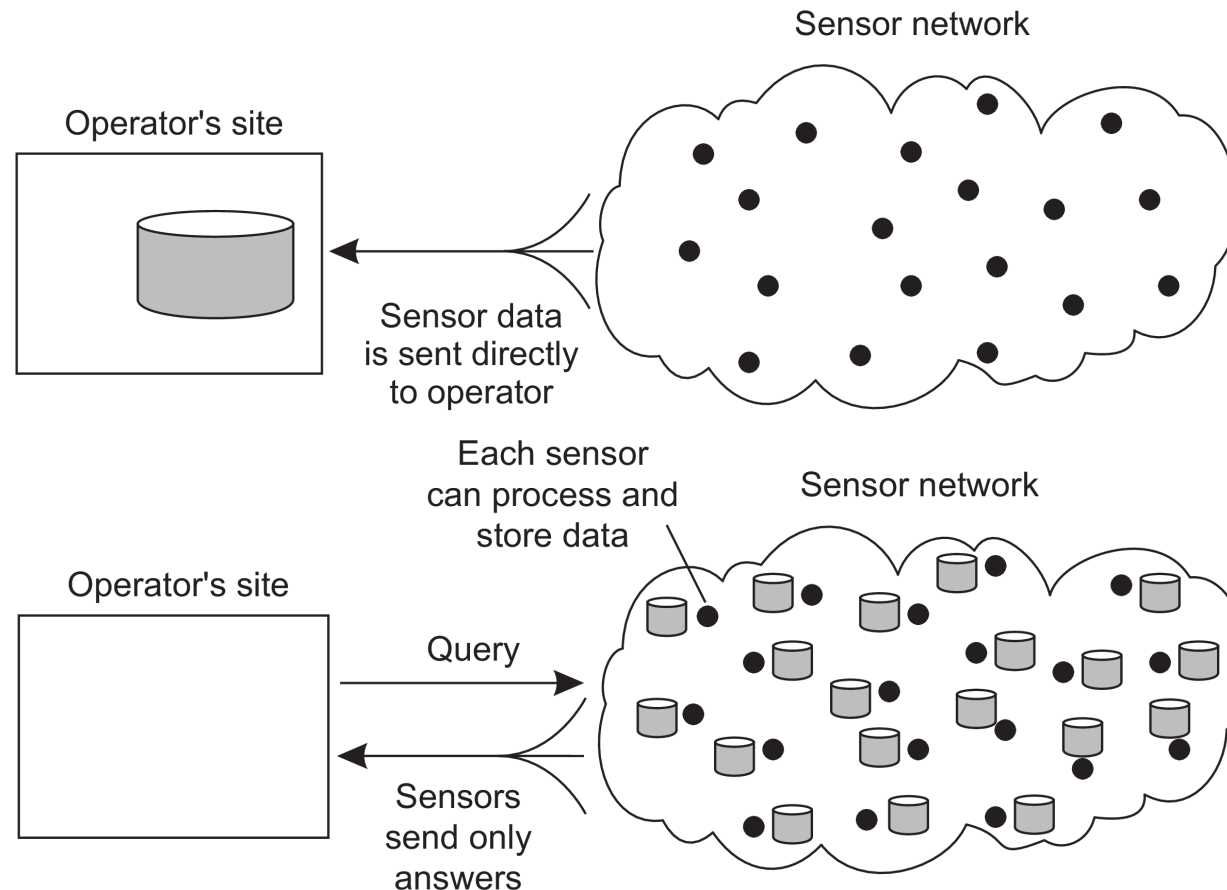
- Glavne karakteristike **mobilnih računarskih sistema**:
 1. **Širok spektar** različitih mobilnih uređaja (pametni telefoni, tableti, GPS uređaji, daljinski upravljači, ...), tipično bežično povezanih
 2. **Mobilnost** povlači da se **lokacija uređaja menja tokom vremena** ⇒ promena lokanih servisa, dostupnosti, itd. Ključan servis: otkrivanje (engl. *discovery*)
 3. **Komunikacija može postati teška**: nema stabilnih ruta, ali takođe ni garantovane povezanosti (engl. *connectivity*) ⇒ umrežavanje otporno na prekide (engl. *disruption-tolerant networking*) – tehnike slanja poruka zasnovane na plavljenju (engl. *flooding*)

Senzorske mreže

- Glavne karakteristike **senzorskih mreža**:
 1. Sastoji se od **čvorova** za koje je **priključen jedan ili više senzora**
 2. **Čvorovi** mogu da se ponašaju i kao **aktuatori**, npr. automatska aktivacija prskalice kada se detektuje požar
 3. Čvorovi su:
 - **Mnogobrojni** (od desetina pa do hiljada)
 - **Jednostavni** (mali kapacitet memorije/izračunavanja/komunikacije)
 - Najčešće **napajani pomoću baterija** (ili čak bez baterija)

Senzorske mreže

- Senzorske mreže kao distribuirane baze podataka, potreba za obradom podataka unutar mreže (npr. stablo u mreži i agregacija), dva ekstremna rešenja:



Izvor: <https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/>