**Arhitekture raspodijeljenih sustava**

Oblikovanje arhitekture programske potpore

◼ Donošenje ispravnih odluka u oblikovanju arhitekture

◼ Stilovi arhitekture programske potpore

◼ Modeli i pogledi na arhitekturu

◼ Procjena arhitekture (metoda kompromisa te analiza troškova i dobiti)

◼ Vrednovanje i sukladnost arhitekture s principima oblikovanja (12 + 5)

◼ Dokumentiranje arhitekture

**Arhitekturni stil** - familija sustava definirana sličnim oblicima strukturne organizacije i opisana jasno definiranim rječnikom komponenti i konektora te pripadajućim topološkim ograničenjima

**Raspodijeljeni sustav -** sustav sastoji se od skupa nezavisnih računala, povezanih mrežom i posredničkim programima, koja omogućuje koordinaciju rada i dijeljenje resursa, tako da korisnici doživljavaju sustav kao jedinstveni računalni sustav

Izazovi

◼ Složenost

◼ Sigurnost

◼ Upravljanje (engl.

Manageability)

◼ Nepredvidljivost (engl.

Unpredictability)

Svojstva

◼ Dijeljenje resursa

◼ Otvorenost (heterogenost)

◼ Konkurentnost (engl.

Concurrency)

◼ Skalabilnost

◼ Neosjetljivost na greške

◼ Transparentnost

Primjeri >>> Klijent – poslužitelj - najčešća

◼ Sustavi ravnopravnih sudionika (engl. Peer-to-Peer) P2P

◼ Srodne socijalne mreže (engl. affinity communities)

◼ Kolaborativno izračunavanje (engl. collaborative computing)

◼ Slanje poruka u stvarnom vremenu (engl. Instant Messaging)

◼ Upravljanje složenim sustavima (automobil, tramvaj, …)

Problemi oblikovanja raspodijeljenih sustava

◼Kako se alociraju i pokreću funkcije poslužitelja?

◼ Kako se definiraju i šalju parametri između klijenta i poslužitelja?

◼ Kako se rukuje neuspjesima (pogreškama) u komunikaciji?

◼ Kako se postavlja i rukuje sa sigurnošću?

◼ Kako klijent pronalazi poslužitelja?

◼ Koje strukture podataka koristiti i kako rukovati s njima?

◼ Koja su ograničenja u istovremenom radu dijelova raspodijeljenog

sustava?

◼ Kako se uopće skupina komponenata usuglašava oko zajedničkih

pitanja?

**ARHITEKTURA KLIJENT - POSLUŽITELJ**Poslužitelj (*engl. server*)

◼ Program koji dostavlja uslugu drugim programima koji su spojeni na njega preko komunikacijskog kanala.

Klijent (*engl. client*):

◼ Program koji pristupa poslužitelju (ili više njih) tražeći uslugu

◼ Poslužitelju mogu pristupiti mnogi klijenti istovremeno

Primjeri : The World Wide Web, E-mail, Network File System -NFS, Transaction Processing System, Remote Display System, Communication System, Database System

Sekvenca aktivnosti:

1. Poslužitelj započinje s radom. Spajanje još nije dozvoljeno.

2. Poslužitelj dozvoljava spajanje („sluša”) i čeka na dolazak klijentskog zahtjeva.

3. Klijenti započinju s radom i obavljaju razne operacije

4. Kada klijent pokuša spajanje na poslužitelja, poslužitelj mu to omogući (ako želi).

5. Poslužitelj čeka na poruke koje dolaze od spojenih klijenata.

6. Kada pristigne poruka nekog klijenta poslužitelj poduzima akcije kao odziv na tu poruku.

7. Klijenti i poslužitelj nastavljaju s navedenim aktivnostima sve do odspajanja ili prestanka rada.

Funkcionalnost poslužitelja

1. Inicijalizacija poslužitelja
2. Započinje slušati klijentska spajanja
3. Rukuje sljedećim tipovim događaja koje potiču klijenti:
   1. Prihvaća spajanje
   2. Odgovara na poruke
   3. Rukuje odspajanjem klijenta
4. Može prestati slušati i mora čisto završiti rad