

# ARHITEKTURA INFORMACIONOG SISTEMA

## Definicija informacionog sistema (IS)

"Informacioni sistem je skup podataka o nekom sistemu iz okruženja i skup postupaka za njihovo održavanje i korišćenje".

"Postupci" se javljaju u dva vida:

Održavanje: Izmena sadržaja podataka u cilju njihovog usklađivanja sa stanjem sistema iz okruženja.

Korišćenje: Uvid u sadržaj podataka za potrebe svakodnevnog rada ili za izveštavanje.

## Komponente IS

IS čine četiri osnovne komponente - jedna u vezi podataka i tri u vezi postupaka:

- D** Podaci, sa perzistencijom (potencijal trajanja sve do eksplicitnog uklanjanja).
- A** Pristup podacima, u oba smera (očitavanje i izmena).
- T** Transformacija, obrada podataka bilo kog vida.
- P** Prezentacija, korisnički interfejs u oba smera (ulaz i izlaz).

Navedene komponente su osnova za razmatranje arhitekture IS.

## Arhitekture IS

Prva i danas zastarela varijanta arhitekture IS je potpuno centralizovana i zasnovana na običnom terminalu ("dumb terminal") i serveru:

**T ( ) <---> S (P, T, A, D)**

Sve je na serveru, uključujući tu i kompletnu prezentaciju. Terminal, tekstuelni ili grafički, hardverski intepretira ulaz i izlaz podataka.

Prednost ove arhitekture jeste njena maksimalna sigurnost. Svi postupci su locirani na jednom mestu - na serveru - i sa strane udaljenih klijenata nisu moguće nikakve zloupotrebe pošto se na njima ne nalaze nikakve aktivne softverske komponente. Druga pogodnost je u tome što je održavanje IS, odnosno odgovarajućeg softvera, centralizovano.

Nedostatak ove arhitekture, koji je čini neprimenjivom u današnjim uslovima distribuiranog poslovanja, jeste to što je svo opterećenje locirano na serveru.

Danas su prisutne tri savremene varijante arhitekture IS:

- ✓ arhitektura "Klijent - Server";
- ✓ arhitektura "Klijent - Server aplikacije - Server podataka";
- ✓ arhitektura "Klijent - Server prezentacije - Server aplikacije - Server podataka".

U oblasti elektronskog poslovanja i elektronske uprave danas su dominantne poslednje dve arhitekture.

Pre razmatranja ovih arhitektura neophodno je objasniti mogućnost raspodele pojedinih komponenti IS.

Komponenta pristupa **A** sadrži dva sloja:

- Ab** bazični sloj pristupa koji se nalazi uvek tamo gde su i podaci, na odgovarajućem serveru; kod baze podataka to je deo softvera zadužen za izvršavanje SQL naredbi;
- Ae** prošireni sloj pristupa koji može biti tamo gde je i **Ab** ili odvojeno od njega; kod IS sa bazom podataka to su ili upapred pripremljene SQL naredbe ili deo softvera koji te naredbe generiše dinamički, na osnovu dijaloga sa korisnikom.

Komponenta prezentacije **P** u posebnim okolnostima može da sadrži dva sloja:

- Pb** bazični sloj prezentacije koji se nalazi uvek tamo gde je korisnik, na klijentu; to je softver koji izvršava neki skript prilikom dijaloga sa korisnikom u oba smera (tipični primer za to je browser koji prikazuje HTML stranice);
- Pe** prošireni sloj prezentacije koji se nalazi odvojeno od **Pb**, na odgovarajućem serveru, i koji dinamički generiše skript za **Pb** (tipični primer za to je tehnologija aktivnih stranica, ASP ili JSP).

Pregled varijanti i podvarijanti arhitektura IS koji sledi obuhvata i tri kriterijuma valjanosti arhitekture IS. To su:

Cen	centralizacija održavanja IS:	mogućnost instalacije novih verzija softvera sa jedne lokacije;
Zas	zaštita i sigurnost IS:	mogućnost obezbeđenja od neovlašćenog pristupa, zloupotreba rada i oštećenja podataka;
Ras	raspodele opterećenja u IS:	mogućnost raspoređivanja opterećenja pri radu na razne računare i/ili lokacije.

U pregledu koji sledi oznake za nivoe valjanosti su u rasponu 1-loše do 5-idealno, a uz pojedine podvarijante dati su i prigodni komentari.

## Arhitektura "Klijent-Server"

Podvarijanta----- Cen Zas Ras

**K(P) <---> S(T, Ae, Ab, D)** 5 5 1

Centralizacija održavanja i sigurnost i zaštita su idealni, ali je zato mogućnost raspodele opterećenja loša i svodi se samo na sloj prezentacije.

**K(P, T, Ae) <---> S(Ab, D)** 1 1 5

Loša centralizacija je zbog toga što se instalacija i održavanje moraju vršiti posebno na svakom klijentu. Sigurnost i zaštita su loši pošto je celokupna transformacija na klijentu odakle su podaci iz baze podataka neposredno dostupni i u režimu izmene. Raspodela opterećenja je idealna, na nivou teoretskog maksimuma.

*Kod naredne dve podvarijante T2 na serveru podrazumeva korišćenje SQL procedura na nivou baze podataka ("stored procedure").*

**K(P, T1) <---> S(T2, Ae, Ab, D)** 3 3 3

Centralizacija održavanja je poboljšana zato što se deo transformacije nalazi na serveru, uključujući tu i svo obraćanje podacima. S obzirom da se sa klijenta više "ne vide" podaci nego samo funkcionalnosti, bolji su i sigurnost i zaštita.

**K(P, T1, Ae) <---> S(T2, Ae, Ab, D)** 2 2 4

U odnosu na prethodnu varijantu, na klijentu se nalazi i deo pristupa podacima koji ne uključuje režim izmene nego samo režim očitavanja. To pogoršava centralizaciju održavanja i zaštitu i sigurnost, dok se raspodela opterećenja poboljšava.

## Arhitektura "Klijent - Server aplikacije - Server baze podataka"

Podvarijanta----- Cen Zas Ras

*Za sve podvarijante koje slede karakteristika je postojanje dva specijalizovana servera - servera aplikacije Sa ("Application server") i servera podataka ("Database server"). Rezultat toga je da su sva tri kriterijuma valjanosti u rasponu "dobro-idealno".*

*T na nivou servera aplikacije podrazumeva servlete, a na nivou servera podataka SQL procedure.*

*Za tri podvarijante koje slede na klijentu se nalazi samo sloj prezentacije.*

K (P) <---> Sa (T, Ae) <---> Sd (Ab, D) 4 4 4

Ovde je narušavanje zaštite i sigurnosti moguće samo sa servera aplikacije (teoretska mogućnost, pošto server aplikacije nije na lokaciji korisnika). Dobra raspodela opterećenja se može postići uvođenjem više servera aplikacija koji opslužuju server podataka.

K (P) <---> Sa (T1) <---> Sd (T2, Ae, Ab, D) 4 5 3

Ovde je poboljšana sigurnost zato što server aplikacije više "ne vidi" neposredno podatke

K (P) <---> Sa (T1, Ae) <---> Sd (T2, Ae, Ab, D) 4 4 3

*Za tri podvarijante koje slede na klijentu se uz sloj prezentacije nalazi i deo transformacije koji je po pravilu zadužen za dobijanje izvedenih podataka za prezentaciju. To pogoršava centralizaciju održavanja ali zato poboljšava raspodelu opterećenja.*

K (P, T1) <---> Sa (T2, Ae) <---> Sd (Ab, D) 3 4 5

K (P, T1) <---> Sa (T2) <---> Sd (T3, Ae, Ab, D) 3 5 3

K (P, T1) <---> Sa (T2, Ae) <---> Sd (T3, Ae, Ab, D) 3 4 4

## Arhitektura "Klijent - Server prezentacije - Server aplikacije - Server podataka"

Podvarijanta----- Cen Zas Ras

*Za sve podvarijante koje slede karakteristika je postojanje tri specijalizovana servera - servera prezentacije ("Web server"), servera aplikacije Sa ("Application server") i servera podataka ("Database server"). Prezentacija je raspodeljena: na klijentu se nalazi tzv. "univerzalni klijent" (browser), a na serveru prezentacije generator HTML (ili XML) stranica.*

K (Pb) <---> Sp (Pe) <---> Sa (T, Ae) <---> Sd (Ab, D) 5 4 4

Kod ovog tradicionalnog pristupa sva poslovna logika i svi upiti nalaze se na serveru aplikacije u vidu servleta.

K (Pb) <---> Sp (Pe) <---> Sa (T1) <---> Sd (T2, Ae, Ab, D) 5 5 3

Ovde je sloj transformacije raspodeljen između servera aplikacije (servleti) i servera podataka (SQL procedure). Server aplikacije nema neposredni pristup podacima.

K (Pb) <---> Sp (Pe) <---> Sa (T1, Ae) <---> Sd (T2, Ae, Ab, D) 5 4 3

Razlika u odnosu na prethodnu varijantu je u tome što server aplikacije "vidi" podatke, ali samo u režimu očitavanja.

## Preporučena arhitektura IS

Za potrebe informacionog sistema elektronske uprave ("e-government") poželjna je arhitektura koja će, s obzirom na veliki broj dislociranih korisnika, prvenstveno obezbediti visok nivo zaštite i sigurnosti i povoljnu mogućnost raspodele opterećenja. Pri tome, degradacija centralizacije instalacije treba da bude do prihvatljivog nivoa.

Arhitektura koja zadovoljava prethodne zahteve je kombinacija arhitektura sa dve (Sa,Sd) i tri vrste servera (Sp,Sa,Sd) koja će optimalno opsluživati dve vrste korisnika:

- ✓ interni korisnici: korisnici unutar državne uprave koji pristupaju sistemu preko intraneta - specijalne interne mreže (koriste Sa i Sd);
- ✓ eksterni korisnici: korisnici izvan državne uprave (ustanove i građani) koji pristupaju sistemu preko Interneta - javne univerzalne mreže (koriste Sp, Sa i Sd).

Podrazumeva se da bi pristup IS u režimu izmene podataka bio isključivo preko intraneta, odnosno da bi se preko Interneta mogli raditi samo uvid u podatke i izveštavanje.

Razlog zašto za obe vrste korisnika nije prihvaćen objedinjeni pristup sa tri vrste servera leži u još uvek nedovoljnoj sigurnosti univerzalnog klijentskog softvera, odnosno browsera. U uslovima režima izmene podataka IS bi bio podložan raznim vrstama zloupotreba i napada od kojih bi najblaži bio u vidu zagušenja (denial of service). U prilog tome govori okolnost da se u praksi browser "Internet explorer" pokazao kao najnestabilnija i najranjivija komponenta u okviru Microsoft palete softvera.

Preporučena arhitektura se može prikazati kako sledi:

```

                                     Sa (T2 ,Ae) <---> Sd (T3 ,Ae ,Ab ,D)
Ke (Pb)    <---> Sp (Pe) <--->
Ki (P ,T1) <----->
```

Ovde je sa Ke označen eksterni a sa Ki interni klijent. Komponenta transformacije T1 je isključivo u funkciji izvođenja podataka za potrebe prezentacije. Celokupna poslovna logika sadržana je u komponenti transformacije T2, a komponenta transformacije T3 realizuje određene kritične operacije koje su po prirodi stvari nepromenjive. Mogući režimi rada na relaciji Sa - Sd su:

```

                                     Sa (T2 ,Ae) <---> Sd (T3 ,Ae ,Ab ,D)

direktna izmena                    T2di ,Ae -----> Ab ,D
direktno ocitavanje                T2do ,Ae <----- Ab ,D
indirektna izmena                  T2ii -----> T3i ,Ae .Ab ,D
indirektno ocitavanje              T2io <----- T3o ,Ae ,Ab ,D
```

## Zaključne napomene

Preporučena ili eventualno nešto drugačija arhitektura IS koja bi proizašla iz detaljne analize svakako treba da zadovolje sledeće zahteve:

- ✓ Na klijentu u bilo kojoj varijanti (intranet, Internet) ne sme da postoji komponenta pristupa Ae, odnosno ni jedna SQL naredba u bilo kom obliku. U suprotnom, u posebnim okolnostima bilo bi moguće narušavanje zaštite i sigurnosti (tzv "SQL injection" efekat).
- ✓ Konekciju sa bazom podataka odnosno Sd uspostavlja isključivo Sa, a klijent uspostavlja konekciju sa nekim od servera iznad toga (Sa za intranet, Sp za Internet).
- ✓ Način komunikacije sa Sa treba da bude isti i za intranet i za Internet. U suprotnom, bile bi neophodne dve varijante Sa.

Prethodni pregledi arhitekture IS su bili konceptualni, u smislu da su naznačeni po jedan klijent K i po jedan od svake vrste servera Sx. U stvarnosti će važiti sledeće:

- ✓ Postojeće više servera podataka Sd razmeštenih na pogodnim lokacijama uz primenu tehnika distribucije i replikacije (lokalne i udaljene).
- ✓ Na svakoj lokaciji sa serverom podataka Sd postojeće bar jedan server aplikacije Sa, a radi raspodele opterećenja može biti i više njih u paraleli.
- ✓ Na svakoj lokaciji dostupnoj eksternim korisnicima (preko Interneta) postojeće uz prethodno i jedan ili više servera prezentacije Sp.
- ✓ Postojeće veliki broj klijenata (i intranet i Internet) raspoređenih po velikom broju lokacija širom države.
- ✓ Uz sve navedene vrste servera postojeće i niz pomoćnih ("firewall", "proxy", server elektronske pošte itd.).