

# 11 - Klijent serverske aplikacije

## SADRŽAJ

**11.1** Pojam Interneta i Intraneta

**11.2** Pojam WWW

**11.3** Anatomija WWW aplikacije

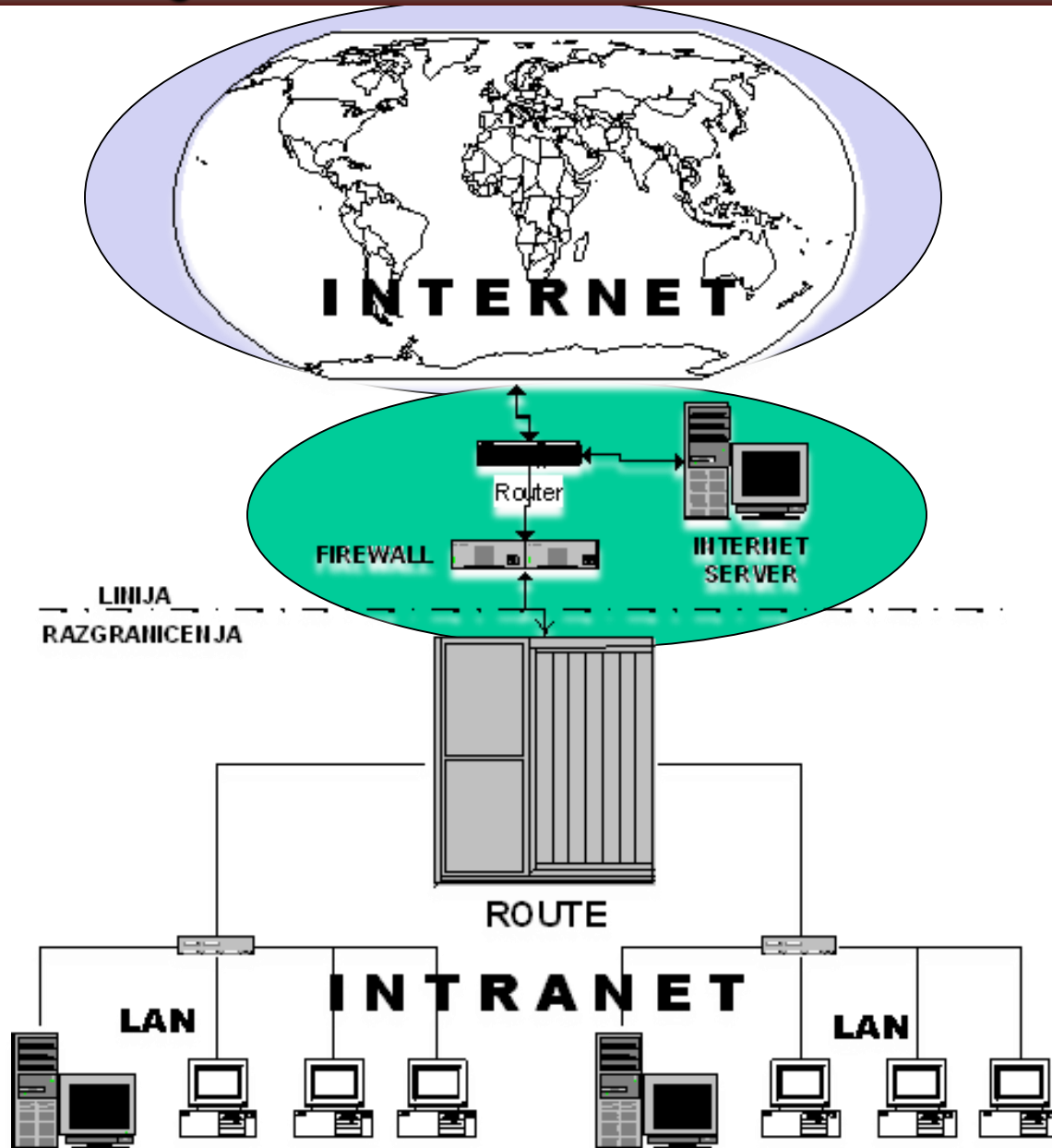
**11.4** Karakteristike WWW aplikacije

**11.5** WWW servisi

# 11.1 Pojam Interneta i Intraneta

- Internet je **kolekcija mreža**, uključujući Arpanet, NSFnet, regionalne mreže, LAN-ove, univerzitetske mreže, koje su **međusobno povezane sa skupom protokola i servisa** nazvanih "***Internet protocol suite***"
- Veoma često se ovaj skup skraćeno naziva **TCP/IP skup protokola**
- Među **najznačajnijim servisima** su *ftp, telnet, gopher, finger, remote login, e-mail, network file systems, remote printing, remote execution, name servers, terminal servers* i dr.
- Do nedavno se tehnologija koja je omogućila širenje fenomena Interneta, primenjivala **uglavnom na globalnom nivou**.
- Težište primene se sve više pomera ka LAN-ima koje koriste prednosti TCP/IP protokola **za deljenje resursa unutar izolovanog sistema**
- Takve mreže su poznate pod nazivom **Intranet** i veruje se da će njihov razvoj i primena **sledećih godina u velikoj meri zaseniti Internet**
- **Intranet i Internet** mogu da se povežu preko **firewall-a i NAT servera**
- Njegova uloga je da sa Interneta **selektivno propušta** samo određeni (bezbedni) saobraćaj kao i da **mapira privatne u javne IP adrese**

# 11.1 Pojam Interneta i Intraneta



# 11.1 Pojam Interneta i Intraneta

- Razlika između **Interneta** i **Intraneta** je između **javnosti** i **privatnosti**
- Da bi bio siguran, Intranet obično ima **veći propusni opseg**, **pouzdanije korisnike**, **veće mere bezbednosti** i **strože definisana prava pristupa**
- Preduzeće može da koristi Web kao **jednostavan i efektan način da zaposlenima omogući pristup podacima**: interni telefonski imenik, rasporedi dežurstava, cenovnik restorana i letovališta, plan godišnjih odmora, plan stručnog usavršavanja i obuke kadrova, interna akta i sl.
- Da li bi ovakve informacije trebalo **da budu javno dostupne**?
- U ovakvim slučajevima jasno je da je **prednost na strani Intraneta**.
- **Neki podaci** na Intranetu mogu da budu dostupni **samo užem krugu korisnika**; npr. poslovno rukovodstvo može da ima interaktivni pristup informacijama bitnim za donošenje poslovnih odluka.
- Sa druge strane **opšti podaci o preduzeću**, delatnost preduzeća, cenovnik usluga, adrese i telefoni predstavništava, potrebe za radnom snagom i sl. **su prirodni kandidati za Internet**.

# 11.2 Pojam **World Wide Web-a**

- Nastao je u CERN-u kao posledica potrebe naučnika za **lakšom međusobnom komunikacijom i razmenom informacija**.
- To je skup servisa koji koriste **HTML** jezik za **kreiranje** dokumenata i **HTTP** protokol za **pristup** multimedijalnim distribuiranim bazama
- Celokupna njegova arhitektura zasniva se na **klijent/server modelu**.
- Na serveru se nalazi **prezentacija**, a na klijentu **browser** (čitač).
- Zbog jednostavnosti korišćenja WEB je postigao **veliku popularnost** i postao **dominantni medijum** za razmenu informacija i usluga
- Postoji nekoliko **tehničkih** i **poslovnih** prednosti sve većeg korišćenja klijent/server aplikacija na Internetu ili Intranetu:

## 1. Tehničke prednosti

- ✓ **lakša implementacija i održavanje** - ušteda u vremenu i novcu, aplikacija se instalira na jednom mestu, nema potrebe za intezivnom obukom i tehničkom podrškom, jednostavno instaliranje novih verzija
- ✓ **brži razvoj i podrška većeg broja različitih računarskih platformi**

## 2. Poslovne prednosti - uključuju **povećano prisustvo na tržištu** i **kompetitivne prednosti**.

# 11.2 Pojam **World Wide Web-a**

- Web strana koja se u pozadini **oslanja na bazu podataka** nudi klijentima veoma veliki opseg različitih informacija na Internetu.
- Web strane kreirane pomoću nekog od HTML editora **sadrže statičke informacije**, tj. koje su bile važeće u trenutku kreiranja strane.
- Sa gledišta marketinga dobro projektovana **dinamička Web strana** može da **bude efikasnija** alatka za dopunsku ponudu roba i usluga
- **Baze podataka** koje mogu da se pretražuju i daju odgovore na najčešće postavljena pitanja mogu da budu **prvi oblik tehničke podrške**
- Da bi se informacije na strani dopunile ili ažurirale **potrebno je izmeniti samu stranu – omogućiti dinamičko menjanje informacija**.
- To može da predstavlja ozbiljan problem čije rešavanje zahteva **znatno angažovanje ljudi i sredstava**.
- Vrlo je čest slučaj da ažurni podaci već postoje u DBMS-u, a da se njihovim prenošenjem na Web **duplira i utrošeno vreme i prostor**.
- Rešenje problema leži u **konverziji iz formata DBMS u HTML oblik**.
- Postoje **dva načina** da se ovo obavi.

# 11.2 Pojam **World Wide Web-a**

1. **Prvi** i jednostavniji je da se u okviru DBMS-a formira forma **VIEW**
  - ✓ sadrži podatke koji će sa naći na strani
  - ✓ SQL Server Web Assistant služi da se ti podaci publikuju tj. prikazuju
  - ✓ HTML strana je **statička** ali njeno ažuriranje je automatizovano
2. **Drugi** način je **kreiranje u potpunosti dinamičkih strana**
  - ✓ Ova realizacija zavisi od vrste DBMS-a i Web servera koji se koriste.
  - ✓ U slučaju MS SQL server i IIS kreiraju se **.htx** i **.idc** fajlovi.
  - ✓ Prvi predstavlja **html fajl** koji je proširen sa svega šest službenih reči (**begindetail, enddetail, if, else, endif, “%z”**) koje omogućavaju vezu sa **SQL serverom** dok se u drugom fajlu nalaze **SQL naredbe**.
  - ✓ To može da bude **bilo koji niz SQL komandi** kakav bi bio napisan u **SQL script fajlu** i on može da se izvrši u jednoj ili više transakcija.
  - ✓ Da bi se omogućilo povezivanje sa bazom podataka potrebno je da se kreira i **odgovarajući ODBC ulaz**, kao i da se **definišu prava pristupa**.
  - ✓ Takođe je moguće napisati program u programskom jeziku **Java** i izvršavati ga kao **applet** koji se sadrži u HTML strani.
  - ✓ Za vezu sa bazom koristi se **JDBC**, native JDBC driver ili **ODBC**

# 11.3 Anatomija WEB aplikacije

- Pod pojmom Web aplikacije podrazumeva se širok skup programskih rešenja koje kao korisnički *interface* koriste neki **Web browser**.
- Sve Web aplikacije funkcionišu po principu **klijent – server**.
- Server predstavlja računar na kome je aplikacija smeštena i koji u zavisnosti od zahteva, korisniku isporučuje traženi rezultat tj. podatak
- Klijent može biti bilo koji **Web browser** ili neki specijalno napravljeni **klient program** koji kontaktira server da bi korisnik dobio rezultate
- **Elementarni oblik** Web aplikacije jeste **obična Internet prezentacija**.
- Web aplikacije su obično **predmet skepticizma i potcenjivanja** kada su u pitanju mogućnosti realizacije nekih kompleksnih problema ili kreiranja nekog ozbiljnijeg *software*-a.
- Kompletan *software*-ski deo Web aplikacije **smešten je na serveru**, i samim tim **veći broj ljudi može da ga koristi istovremeno**.
- Aplikacija je **sigurna od krađe, prepravljanja** i uopšte bilo kakvog **uticaja na kod** u onolikoj meri u kojoj je i sam server siguran.
- Znači, nema nikakve potrebe za komplikovanim **instalacijama, vođenja računa o kompatibilnosti i podešavanjima**.



# 11.3 Anatomija WEB aplikacije

- Nije potreban **nikakav specijalni software** za pristup aplikaciji, **nema ograničenja** da se mora koristiti isključivo sa jednog računara, a brzina komunikacije sa aplikacijom **zavisi od brzine pristupa serveru**
- Klient aplikacija je **Web čitač** (Chrome,Safari,Explorer,Mozilla,Opera) i obično je **uključena u okviru OS** instaliranog na klijentskom računaru
- Koristeći Internet kao globalnu mrežu, **on-line kupovina, elektronsko bankarstvo, informacijski sistemi u okviru firmi, turističke aplikacije** i mnogi drugi **složeni sistemi** bazirani su upravo na **Web aplikacijama**.
- Osnovne tehnologije za izradu Web aplikacija su **HTML jezik** za izradu statičkih strana i **Server-Side Scripting** jezici za generisanje HTML koda, odnosno, izradu **dinamičkih strana**.
- **Preuzimanje i prikaz statičkih Web strana** predstavlja jednu realizaciju klijent-server arhitekture,čiji su akteri **Internet pretraživač** i **Web server**
- Web server **predstavlja softver**, koji se instalira na određenom serveru koji **upravlja zahtevima za pristup određenoj Web stranici**, tako što postupa po zahtevu klijenta, isporučujući sadržaj neke Web strane.
- Web server i pretraživač **komuniciraju razmenom poruka putem HTTP**

# 11.3 Procedura rada WEB aplikacije

- Korisnik u Web pretraživaču **unos** adresu - URL ili popunjava obrazac
- Računar korisnika šalje **HTTP** ili **HTTPS** zahtev preko Interneta Web serveru gde se nalazi HTTP server, skripting jezik ili Web aplikacija
- Web aplikacija **analizira upućeni zahtev** i po potrebi:
  - ✓ povlači i/ili obrađuje **podatke**
  - ✓ povlači i/ili obrađuje **dokumenata**
  - ✓ ažurira ili pretražuje **baze podataka**
  - ✓ komunicira sa drugim **Web serverima** ili **Web servisima**
  - ✓ vrši **redirekciju** na drugi Web server (**Content Delivery Network**)
- Rezultat rada Web aplikacije može biti **Web stranica, CSS, XML, slika, video** ili **flash**, koje Web server šalje preko Interneta do računara klijenta
- Web stranice mogu biti:
  - 1. Statičke** - jednom formirane, više se ne menjaju. Obične HTML stranice, za čiji prikaz je dovoljan samo Web pretraživač.
  - 2. Dinamičke** -sadržaj se menja **u zavisnosti od interakcije** sa klijentom  
Za generisanje su potrebni **Web server** i **skripting jezik** (npr. PHP).

# 11.3 Anatomija WEB aplikacije



**Prezentacioni sloj**

**Aplikacioni sloj**

**Sloj podataka**

# 11.3 Anatomija WEB aplikacije

- Osnovni gradivni elementi dinamičke Web aplikacije su u mnogim aspektima slični sa dvoslojnom/troslojnom klijent/server arhitekturom
- Zbog velikog broja i raznolikosti proizvoda za ovu namenu, kao i OS, nije moguće obraditi sva potencijalna rešenja, pa je akcenat stavljen na Windows OS i njemu pripadajuće komponente SQL Server i II Server
- **Dvoslojna klijent/server aplikacija** tipično se sastoji od klijentskog softvera koji implementira korisnički interfejs i komunikaciju sa bazom podataka, kao što je SQL Server, radi postizanja ažurnosti podataka.
- Dakle grafički korisnički intrerfejs se obavezno nalazi na klijentskoj mašini, DBMS obavezno na serveru, dok se poslovni deo sistema (sama obrada) može nalaziti ili na klijentu ili na serveru.
- Jedan od važnih principa koji je ispoštovan u dvoslojnoj arhitekturi je princip transparentnosti, tako da korisnik ne mora da zna na kojim se platformama izvršavaju delovi softvera informacionog sistema.
- Takođe postoje realizovani DBMS kod kojih je server baze distribuiran na više platformi, ali za korisnika on predstavlja jedinstveni server baze
- Jedinstvenost pogleda na bazu je podržana na više nivoa.

# 11.3 Anatomija WEB aplikacije

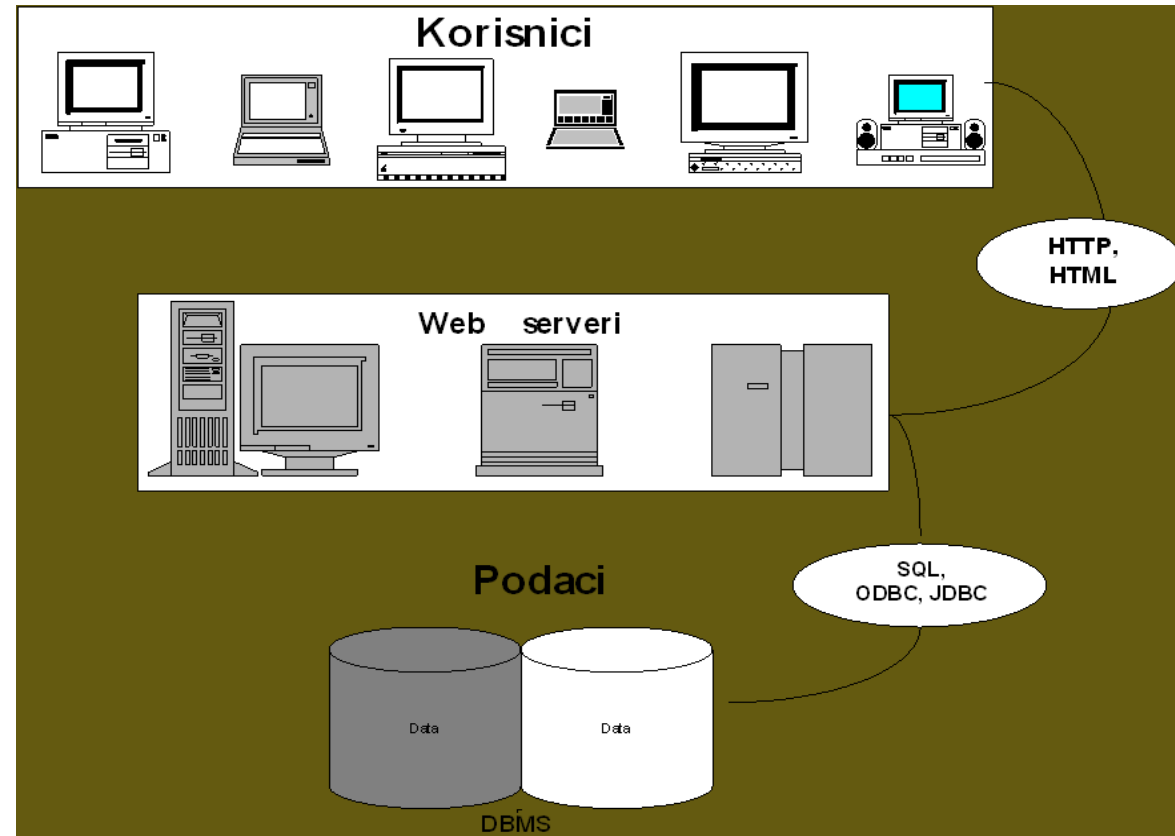
- U troslojnim sistemima javlja se nezavisni **aplikacioni server**, čime se **rasterećuje** server baze podataka ili klijentska mašina.
- Za razvoj grafičkog korisničkog interfejsa se uglavno koriste **jezici za vizualno programiranje** kao **C++**, **Delphi** ili alati kao **PowerBuilder-a**
- Programiranje aplikacionog servera je uglavnom **zasnovano na univerzalnim programskim jezicima** (C++) uz korišćenje CASE alata.
- **Server baze podataka** se uglavnom oslanja na neki od proizvoda kao što su **ORACLE**, **MS SQL Server**, **Informix**, **Sybase SQL Server**.
- **Transparentnost** je uglavnom **zasnovana na rešenjima proizvođača**, što znači da organizacija **ima svoj Intranet** na koji se DBMS oslanja.
- Pojava Web-a **dovela je do proširenja koncepta transparentnosti**.
- Pomoću Web-a postignuta je i **transparentnost u odnosu na servere**
- To omogućava **da se ostvari pristup podacima po sadržaju**, ne obazirući se na **platformu**, **OS računara**, **lokaciju računara** i sl.
- Osnova za takvu transparentnost je **jedinstven standard** za kreiranje Web dokumenata **HTML** i pojava **Web browser-a**
- **Svi ovi elementi su prisutni** i na **Web-zasnovanoj aplikaciji**.

# 11.3 Anatomija WEB aplikacije

➤ Aplikacija se implementira u nekoj kombinaciji **HTML** (korisnički interfejs) i **serverovog skript koda** (povezuje se sa bazom podataka).

➤ Komponente na server strani **služe istoj svrsi** kao i kod ručno pisanog programa i obično im se pristupa preko posebnog **skript koda**.

➤ Komponente za pristup serveru i bazi podataka su osnovna građa Web-zasnovane aplikacije.



➤ Na serverskoj strani najosnovniji elementi su servisi za rad nad **podacima**, ali pored njih postoje i neki dodatni servisi kao što su **HTTP Server (statičke)** ili **Internet Information Server (dinamičke)** strane

# 11.3 Osnovne prednosti Web aplikacija

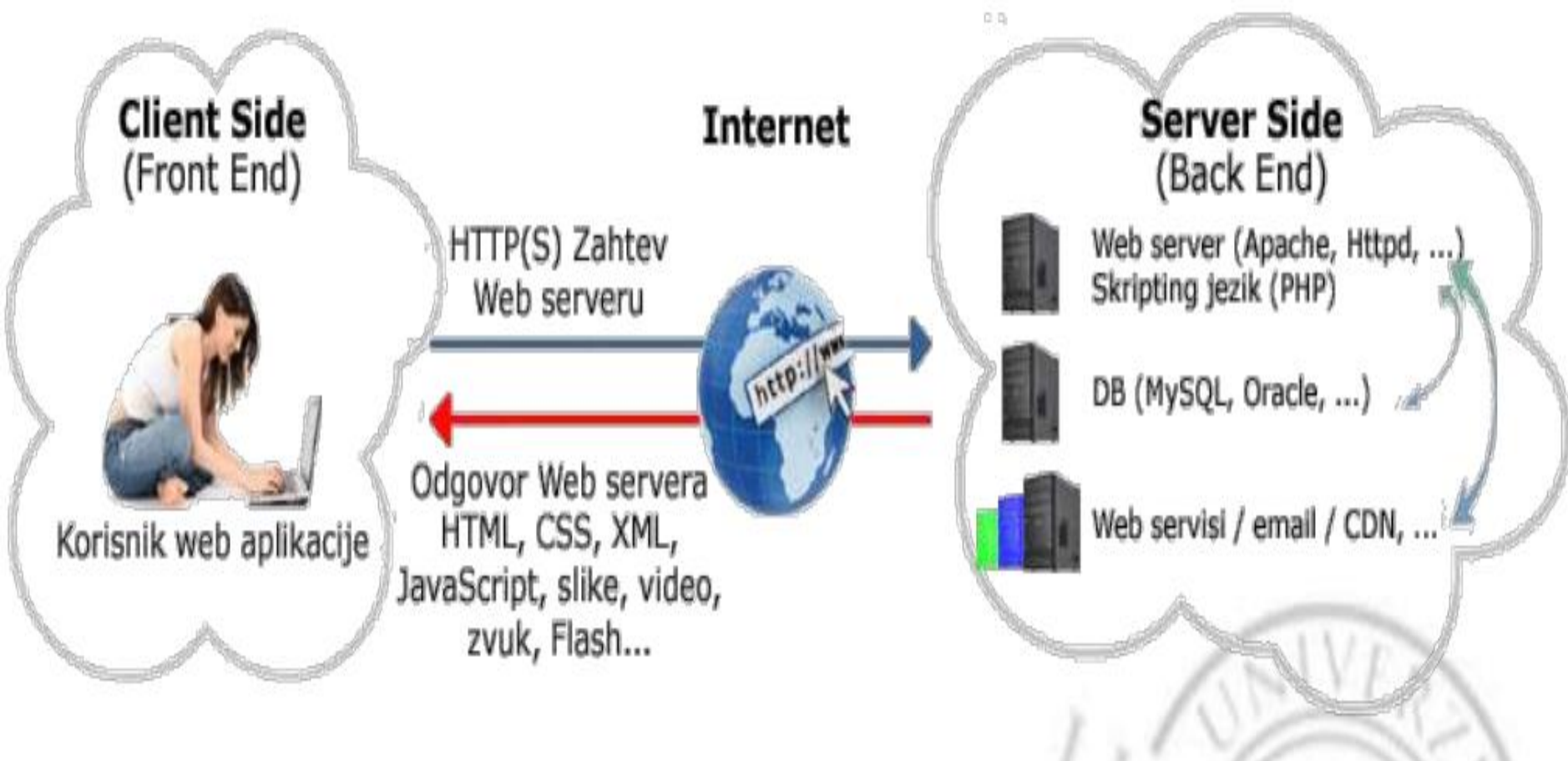
- ✓ **Kompatibilnost** jedne aplikacije sa različitim hardverskim i softverskim platformama (PC, laptop, tablet, Windows, Linux, Mac i td.
- ✓ **Ažuriranje i održavanje na serveru**, bez potrebe da se distribuira i/ili instalira na računarima krajnjih korisnika.
- ✓ **Povoljnije** za korišćenje kod organizacija sa velikim brojem korisnika
- ✓ Po kvalitetu su postale **ravnopravne** desktop aplikacijama.
- ✓ HTML5 omogućava kreiranje **bogatog interaktivnog okruženja** u okviru web pretraživača (video, audio, vizuelni efekti).
- ✓ Zauzima **malo** (ili nimalo) mesta na hard disku korisnika.
- ✓ **SaaS** (*Software as a service*)
- ✓ Neuporedivo **niža cena** u odnosu na desktop aplikacije.
- ✓ **Otežava** softversko piratstvo i/ili **reverse engineering**.
- ✓ **Laka integracija** sa drugim Web servisima.

# 11.3 Prednosti dinamičkih strana

- ✓ Omogućavaju da se program izvršava u programskim jezicima koje Web pretraživači **ne podržavaju**.
- ✓ Daju mogućnost da se programiraju dinamičke Web aplikacije **nezavisno od čitača**, bez pribegavanja programiranju **na strani klijenta**, pomoću **Java apleta**, **DHTML**-a i **ActiveX kontrola**.
- ✓ Omogućuje klijentu **podatke koji su mu inače nedostupni**.
- ✓ Često ostvaruje **brže vreme učitavanja**.
- ✓ Obezbeđuje **poboljšane mere bezbednosti**.
- Korišćenje dinamičkih Web strana **povećava opterećenje servera**, naročito ukoliko njima pristupa veći broj korisnika.
- Potrebna je **veća inicijalna investicija u hardver** Web servera, koji se koriste za generisanje dinamičkih Web strana.
- **Jezik koji se najčešće koristi** za kreiranje dinamičkih strana je **PHP**.
- **PHP** je stekao popularnost zbog svoje **jednostavnosti i sintakse** nasleđene iz programskog jezika C.
- Tokom vremena jezik se proširivao i **sticao mogućnosti za objektno orijentisano programiranje**, naročito od verzije 5.0.



# 11.3 Tok komunikacije



# 11.3 Tok komunikacije-statičke strane

## Učitavanje statičke strane

**WEB SERVER**



1. Autor piše HTML

3. Web server locira .htm datoteku

4. HTML tok (od .htm stranice) je već vraćen čitaču

2. Klijent daje zahtev za Web stranicu



5. Čitač obrađuje HTML i prikazuje stranicu

**KLIJENT**

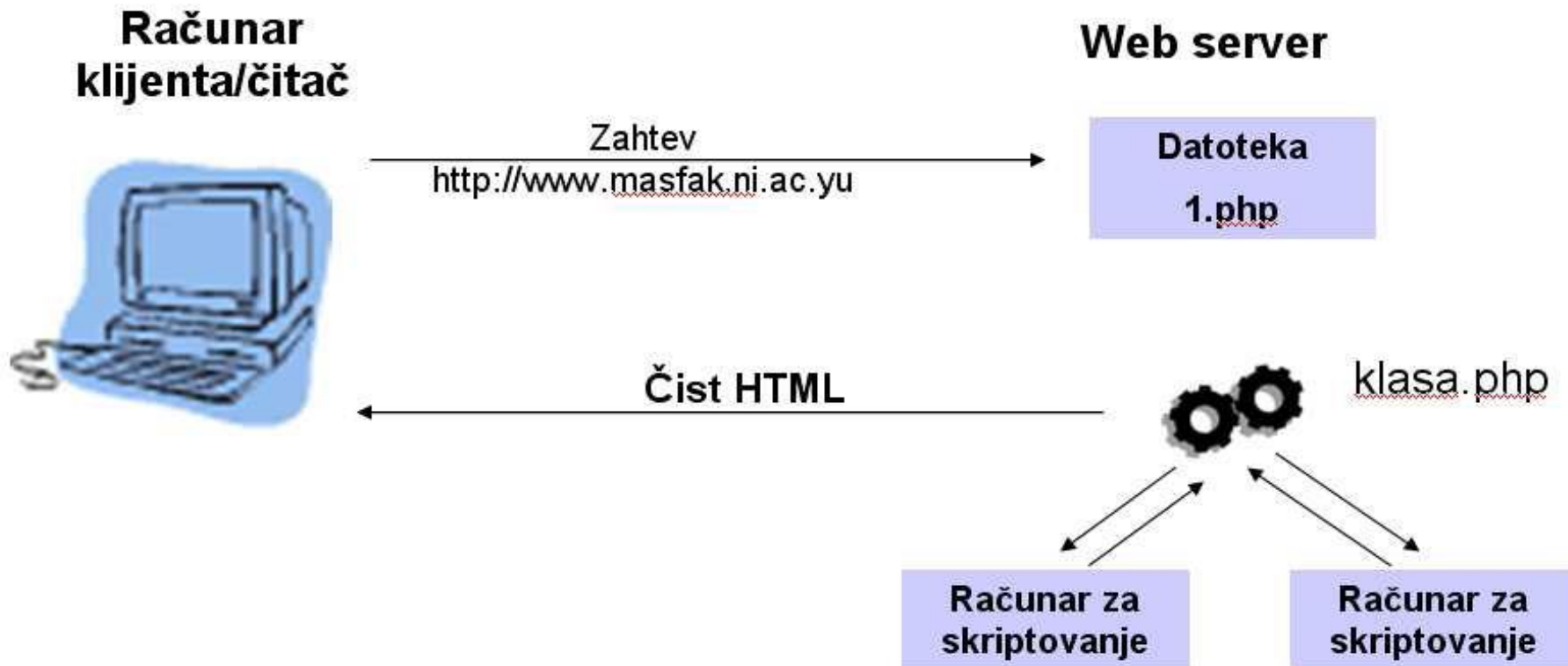
# 11.3 Tok komunikacije-dinamičke strane

## Učitavanje dinamičke strane



# 11.3 Tok komunikacije- obrada skripta

- Za kreiranje programskog koda (*script-a*) čija je uloga da **automatski generišu odgovarajući HTML kod na strani servera**, na osnovu parametara zahteva, koriste se različiti programski jezici i njihove varijante, kao što su **PHP**, **JSP** (Java), **ASP** (Visual Basic), itd.
- **Ove instrukcije nazivamo skriptovima** na strani servera (**Server - Side Scripts**), zato što se izvršavaju na Web serverima.



# 11.4 Karakteristike WWW aplikacije

- **HTML** - razlika između statičnog Web-sajta i dinamičke Web-aplikacije ogleda se u načinu kako se vrši kretanje po HTML stranama
- Statički HTML sajt može da se uporedi sa knjigom čiji se **sadržaj i indeks ne nalazi na jednom mestu**, nego su delovi raspoređeni po tekstu
- Strane na statičkom Web-u **imaju slabo definisane međusobne odnose**
- Svaka strana sadrži veze (*link*) prema ostalim relevantnim stranama, ali **sveukupno kretanje kroz sajt je uglavnom zavisno od korisnika**
- Web-aplikacija je više **okrenuta ka zadatku nego pretraživanju**.
- Veze su više kao **meni**, nego kao indeks, a moguća odredišta bilo sa koje strane **ograničena su operacijama** koje mogu da se izvrše sa te strane
- **Serverski skript kod** - jedan od osnovnih zadataka skript koda koji se sadrži u HTML strani (ili formi) je **da sakupi podatke od korisnika i prosledi ih aplikaciji na serveru**.
- Jedan od **prvih i najpopularnijih intrefejsa** za podršku skript kodu je **Common Gateway Interface (CGI)** protokol.

# 11.4 Karakteristike WWW aplikacije

- **Glavni nedostaci** ovog protokola su:
  - dinamičke strane **moraju da budu u celini konstruisane programski,**
  - **obrada grešaka** koja se ekskluzivno obavlja na server strani **može da degradira vreme odziva** kod sporih međusobnih veza
- Jedan od protokola koji su namenjeni **da unaprede interfejs** ka skript kodu je **Active Server Pages (ASP).**
- To je standardna HTML strana sa **skript kodom ugrađenim u dokument**
- Projektant može da **koristi ugrađeni kod** da bi **implementirao obradu podataka zajedno sa HTML stranom,** i da **pristupi eksternim delovima aplikacije** ili da se **poveže direktno sa bazom podataka**
- Kada browser zatraži ASP, **ugrađeni skript se izvrši od strane *scripting engine* na serveru** i rezultati se **dinamički kombinuju sa HTML pre nego što se dokument vrati browser-u.**
- Na raspolaganju su skript jezici **Visual Basic Script (VBScript), JavaScript (JScript), PHP, REXX, PERL** i mnogi drugi jezici.
- ASP nudi i druga unapređenja u odnosu na standardni CGI **oblasti generisanja strane, održavanja i unapređenja performansi.**

# 11.4 Karakteristike WWW aplikacije

- **ActiveX Server Components** - implementiranje klijent/server rešenja na Internetu predstavlja izazov sa kojim se obično ne suočavamo kod tradicionalnih klijent/server okruženja.
- Kada se instalira tradicionalna klijent/server aplikacija, zajedno sa njom se instalira i sav prateći softver neophodan za njeno izvršavanje.
- To uključuje i softver za podršku bazama podataka, bilo da je integrisan u aplikaciju ili da je isporučen kao sistemski softver-ODBC
- Ovo **nije uvek moguće** u Web zasnovanoj aplikaciji.
- Različiti *browser* podržavaju različite kontrole
- ActiveX Server Components je projektovan da rešava ove probleme.
- Mnogi programski alati su sposobni da grade ove komponente, kao što su: **MS Visual Basic, Delphi, PowerBuilder** i drugi.
- Svaka **ActiveX Server** komponenta je dostupna preko ASP skriptova.
- Osnovne komponente nude širok izbor usluga (servisa) piscu skripta, od **administracije** veza sa korisnicima do **pristupa** bazama podataka.
- Mogućnost da se lako referenciraju u okviru HTML strane predstavlja veliku uštedu u vremenu razvoja aplikacije.

# 11.4 Karakteristike WWW aplikacije

- **Servisi** - završni građevinski blok Web aplikacije je kombinacija elemenata koji radeći zajedno čine jednu celinu.
- Oni uključuju **Internet Information Server** i **SQL Server**.
- **Internet Information Server** - svi Web serveri pa i IIS su servisi koji primaju zahtev za informacijom u vidu *Universal Resource Locator*
- Kako su **dinamičke strane** za prikazivanje velikih baza podataka postajale popularnije, Web serveri su proširili svoju funkcionalnost izvan usluživanja HTTP zahteva.
- IIS, na primer, pruža podršku za administriranje OS, publikovanje, obezbeđivanje sigurnosti i komunikaciju sa bazama podataka pored standardnih HTTP, FTP i gopher servisa.
- **SQL Server** - DBMS je srce svakog informacionog sistema.
- Da bi se kreirale dinamičke Web strane potrebno je da podaci budu smešteni u odgovarajućem formatu.
- Omogućava definisanje **odgovarajućih prava pristupa podacima** za različite korisnike koji mogu da budu definisani posebno za server ili integrisani u bazu podataka korisnika OS.



# 11.4 Karakteristike WWW aplikacije

- Drugi slučaj je **pogodniji za korisnike** jer zahteva samo jedno prijavljivanje, a takođe pojednostavljuje administriranje baze.
- Značajna mogućnost je i rad sa **replikama**, kada se jedan deo podataka drži lokalno na udaljenoj lokaciji na kojoj se vrši obrada
- Kako postoji mogućnost kreiranja dinamičkih strana i bez korišćenja nekog SQL DBMS-a postavlja se pitanje zašto koristiti MSSQL Server
- Razlozi su brojni a najznačajni među njima su:
  1. mogućnost **dinamičkog pristupa** informacijama,
  2. **osiguran integritet** podataka,
  3. **definisana prava pristupa**,
  4. **podrška distribuiranim bazama podataka**,
  5. **povezanost sa operativnim sistemom**,
  6. **povezanost sa ostalim servisima**
  7. **mehanizam povezivanja različitih i međusobno nekompatibilnih softversko-hardverskih platformi u jedinstvenu celinu.**
- Neke od ovih prednosti su **zajedničke za sve relacione baze**, a one koje se odnose na **integrisanost SQL servera sa OS i servisima** su specifične

# 11.5 WEB servisi

- Web servisi predstavljaju novu primenu standardnih softverskih tehnologija koja omogućava povezivanje raznorodnih informacionih sistema na novi način, korišćenjem Internet protokola.
- Web servis je vrsta distribuirane aplikacije sa interfejsom kome se može pristupiti preko komunikacione mreže kao što je Internet.
- Klijent server je tipična arhitektura u kojoj se koriste Web servisi
- Web servis predstavlja aplikaciju koja se koristi putem Interneta, i najčešće koristi HTTP protokol
- Web servisi predstavljaju gradivne blokove savremenih inform.sistema
- Pošto je reč o distribuiranim aplikacijama na Internetu, Web servisi se adresiraju tj. pristupa im se preko URL adresa.
- Osnovna karakteristika Web servisa je **interoperabilnost**.
- Pojam interoperabilnost odnosi se na mogućnost da aplikacije pisane u različitim programskim jezicima i razvijene na različitim platformama mogu nesmetano da komuniciraju.
- Kod Web servisa poštuju se tačno definisana pravila kako bi bila omogućena komunikacija različitih tehnologija.

# 11.5 WEB servisi

- Postoji nekoliko razloga koji su usloveli pojavu i razvoj Web servisa:
- **Interoperabilnost** (*Interoperability*) je potreba za komunikacijom aplikacija razvijenih u različitim tehnologijama, u različitim programskim jezicima i na različitim platformama.
- **SOA** (*Service Oriented Architecture*) je pristup u razvoju softvera koji podrazumeva razdvajanje funkcija u odvojene servise, dostupne preko mreže. Takvi servisi olakšavaju kombinovanje i korišćenje već ugrađenih funkcionalnosti bez potrebe njihovog ponovnog razvoja
- **Skalabilnost** (*Scalability*) predstavlja potrebu da se postojeći skupovi funkcionalnosti menjaju (povećavaju ili smanjuju) uz minimalne troškove i najmanji mogući negativan uticaj na rad sistema.
- **Tipični Web servisi:** pregled kursnih lista, vremenska prognoza, kreditni biro, validacije kreditnih kartica prilikom online plaćanja i td.
- Dve najpoznatije platforme za razvoj Web servisa su: **Microsoft.NET** i **Sun J2EE** (*Java 2 Platform Enterprise Edition*).
- Bez obzira na odabranu platformu, Web servisi mogu sa lakoćom da pozivaju jedni druge.

# 11.5 Podela WEB servisa

- **SOAP Web servis** definisan je **standardima** od strane tela koja se bave standardizacijom Web-a kao što je W3C (*World Web Consortium*)
- **SOAP** (*Simple Object Access Protocol*) Web servis kome se **pristupa preko HTTP protokola** je specijalan slučaj **REST Web servisa**.
- Web servisi **igraju centralnu ulogu** u SOA pristupu razvoja softvera
- SOAP poruka je **XML dokument definisan XML šemom**, i ona **omogućava komunikaciju** klijentske aplikacije i samog Web servisa.
- Kod SOAP Web servisa, **SOAP poruke nisu predmet interesa programera koji kreira Web servis** ili piše klijentski kod za korišćenje
- **REST** (*Representational State Transfer*) predstavlja servis kod koga **nisu definisani** standardi, biblioteke, niti alati za razvoj.
- Uvođenje REST Web servisa u upotrebu predstavlja pokušaj **da se prevaziđe složenost** SOAP Web servisa.
- Klijent Web servis može **biti napisan u bilo kom programskom jeziku** koji ima odgovarajuće biblioteke za **podršku Web servisima**.
- Klijent i Web servisi **ne moraju** da budu napisani u istom jeziku
- Sprega se ostvaruje **putem XML tehnologije**

# 11.5 Osobine WEB servisa

- ❑ **Korišćenje postojeće infrastrukture** - Web servisi u osnovi **koriste HTTP i XML standarde** koji su nezavisni od proizvođača. Web servisi ne zahtevaju nove sisteme već se **oslanjaju na postojeće komunikacione mreže**, standarde za formatiranje podataka, bezbednost i infrastrukturu koja već postoji **čime se snižavaju troškovi i pospešuje interoperabilnost**.
- ❑ **Transparentnost u odnosu na programski jezik** - Web servisi i njihovi klijenti mogu da komuniciraju čak i kada su napisani u **različitim programskim jezicima** kao što su C/C++, C#, Java, PHP, Perl, Python i drugi. Svi oni obezbeđuju biblioteke i okvire (*framework*) za podršku Web servisima.
- ❑ **Modularna arhitektura** - Web servisi su sami po sebi **moduli koji se mogu koristiti pri razvoju aplikacija**. **Integracijom postojećih Web servisa** mogu se dobiti novi Web servisi (na primer, povezivanje servisa koji prati količinu robe u skladištu i servisa za online kupovinu te robe, može voditi stvaranju servisa koji automatski naručuje robu od dobavljača u zavisnosti od količine robe u skladištu).

Hvala na pažnji !!!



Pitanja

? ? ?