

Projektiranje informacijskih sustava

SDLC faza dizajna – dizajn arhitekture

Ak. god. 2011/2012

Dizajn arhitekture

- Dizajn arhitekture opisuje cjelokupni softver (i onaj koji se razvija i onaj koji je potreban za funkcioniranje (npr. OS, baza podataka,...)), hardver i mrežnu okolinu potrebnu za funkcioniranje aplikacije.
- Dizajn arhitekture se najvećim dijelom izvodi iz nefunkcionalnih zahtjeva koji definiraju sigurnosne zahtjeve, zahtjeve prema performansama sustava, lokalizacijske zahtjeve i sl.

Dizajn arhitekture

- Dokumenti (*deliverables*) koje se trebaju generirati u ovom koraku dizajna arhitekture su:
 1. Specifikacija dizajna arhitekture
 2. Specifikacija hardvera i softvera (detaljno opisuje potrebne hardverske (uključivo “pomoćni hardver” npr. printeri, mrežni diskovi za backup,) i softverske komponente za rad nove aplikacije)

Svrha dizajna arhitekture

- Dizajn arhitekture za velike informacijske sustave je izuzetno složen.
- Jedna od glavnih namjena dizajna arhitekture je točno odrediti koji dijelovi aplikacije (softvera) će biti pridruženi kojoj komponenti hardvera (npr. baza podataka će biti na serveru, a aplikacija na klijentu) (UML deployment diagram).

Dizajn arhitekture

- Dizajna arhitekture softverskog sustava uključuju identifikaciju softverskih elemenata (komponenti).
- Softverske komponente zahtjevaju hardverske komponente za svoje izvođenje.
- Stoga dizajn arhitekture informacijskog sustava uključuje i softverske komponente i hardverske komponente.

Softverske komponente

- Vrlo česta podjela softverskih elemenata s obzirom na funkciju je na 4 osnovna dijela:
 1. Pohrana podataka (*data storage*) – većina informacijski sustava zahtijeva pohranu podataka bilo u datoteci bilo u složenoj bazi podataka. Ti podaci su definirani u podatkovnom modelu (ERD).
 2. Pristup podacima (*data access logic*) – dio aplikacije kojim se podaci upisuju, dohvaćaju..., najčešće danas iz relacijskih baza podataka SQL upitima.

Softverske komponente

3. Aplikacijska logika (*application logic*) – dio aplikacije koji izvodi funkcionalnosti definirane u procesnom modelu (DFD), slučajevima korištenja i funkcionalnim zahtjevima.
4. Prezentacijska logika (*presentation logic*) – dio aplikacije koji pruža podatke korisniku i od korisnika prihvata ulazne informacije i naredbe.

Dizajn arhitekture distribuiranih aplikacija

- Također je uobičajno da su veliki informacijski sustavi u pravilu distribuirane aplikacije u smislu da u sustavu postoji više različitih zasebnih računala (različitih performansi, moguće i tako jednostavnih kao što su “dumb” terminal) koje “učeštvuju” u aplikaciji pa ćemo zbog toga ovdje obraditi česte predloške (stilove) arhitekture takvih aplikacija.

Dizajn arhitekture

- Kada promatramo distribuirane aplikacije hardverske komponente možemo podijeliti na:
 - Klijenti
 - Ulazno/Izlazni uređaji korišteni od strane korisnika
 - PC, laptop, ugradbena računala, mobiteli
 - Serveri
 - Velika računala na kojima se nalazi softver
 - Dostupni mnogim korisnicima
 - Mreža
 - Povezuje računala različitim brzinama (modem, ADSL, HDSPA, Ethernet)

Dizajn arhitekture distribuirane aplikacije

- Distribuirane aplikacije mogu se podijeliti na:
 - arhitekture zasnovane na serveru (*Server-Based Architecture*)
 - arhitekture zasnovane na klijentu (*Client-Based Architecture*)
 - arhitekture sa ravnopravnim učešćem (*Client-Server Architecture*)

Serveri

- Serveri su namijenjeni pristupu velikog broja klijenata.
- Najčešće su “snažna” (procesorski) računala.
- Hardverski, servere možemo podijeliti na tri osnovna tipa:
 1. “Mainframe” računala
 - Velika i izrazito snažna računala
 - Visoka cijena (nekoliko milijuna \$)
 - IBM zSeries

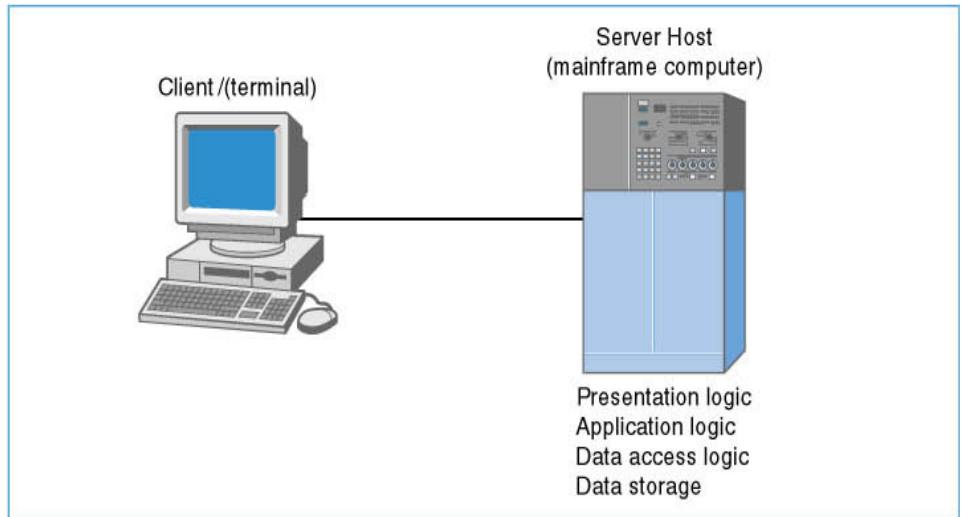
Serveri

grid?, kluster?, cloud?

2. “Minicomputer” računala
 - Velika računala
 - Cijena reda veličine stotine tisuća dolara
 - IBM, HP, Sun
 - RISC arhitektura
3. “Microcomputer” (osobna računala)
 - Mala desktop računala
 - Cijena reda veličine desetke tisuća dolara

Arhitektura zasnovana na serveru

- Server je mainframe računalo.
- Sve komponente aplikacije se izvršavaju na serveru od pohrane podataka, pristupa podacima, aplikacijske logike i prezentacijske logike. (Novija verzija je ipak barem dio prezentacijske logike na klijentu).



Arhitektura zasnovana na serveru

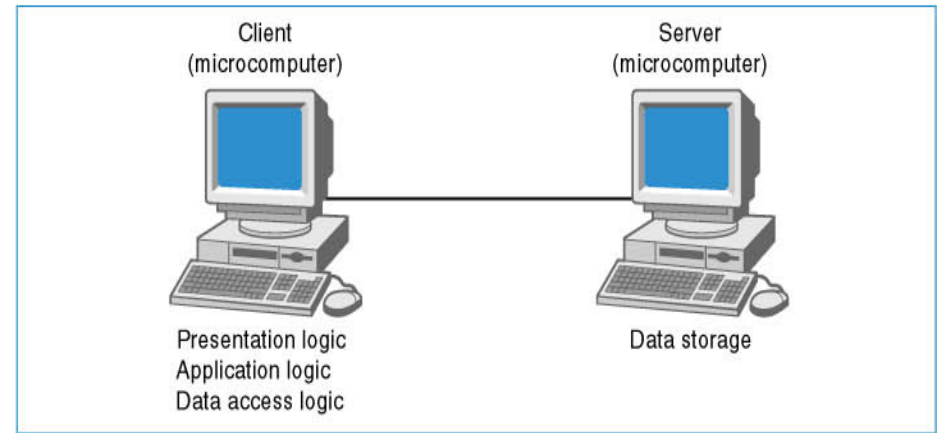
- Klijent je obično samo terminal.
- Omogućava korisniku samo da šalje i prima poruke bez obrade podataka.
- To su intranet aplikacije koje mogu koristiti svoje protokole za komunikaciju.
- Ovakve arhitekture se danas sve rjeđe koriste.

Arhitektura zasnovana na serveru

- Prednosti:
 - Jednostavnost implementacije jer je cijela aplikacija na jednom mjestu.
 - Ne zahtjeva komplicirano upravljanje aplikacijom jer je cijela aplikacija na jednom serveru (npr. kod *upgradea*).
- Nedostatci:
 - S većim brojem korisnika i usluga server postaje opterećen jer je cjelokupna funkcionalnost na serveru. Zato serveri i trebaju biti “moćni”.
 - Skupa nadogradnja centralnog servera jer su mainframe računala inače skupa.

Arhitektura zasnovana na klijentu

- Klijenti su osobna računala. Ne mogu biti terminali jer se logika aplikacije, prezentacija, i pristupa podacima prebacuje na stranu klijenta. Npr. .NET aplikacija na klijentovom računalu koja samo pristupa bazi podataka na nekom serveru.



Arhitektura zasnovana na klijentu

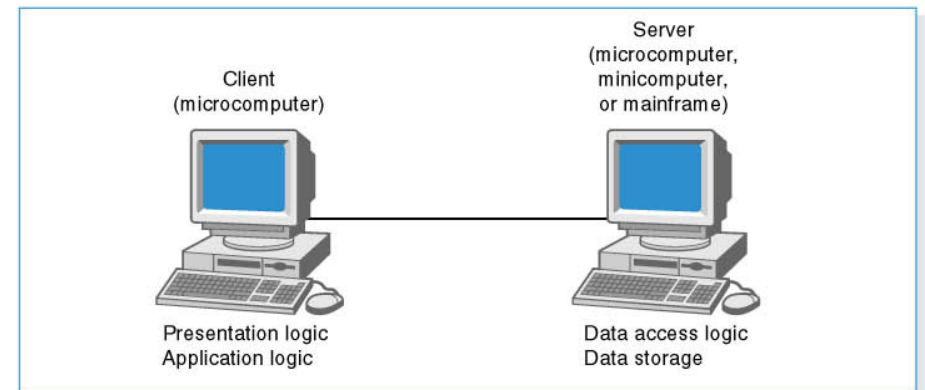
- Server služi samo za pohranu podataka.
- To su obično intranet aplikacije koje mogu, ali i ne moraju biti zasnovane na Internet protokolima.
- Inače kada govorimo o jednom informacijskom sustavu cilj je da podaci budu „centralizirani”. Ne na način da se ne može koristiti distribuirana baza podataka, već na način da svi podaci budu dostupni svim korisnicima sustava.
- Možete imati distribuiranu bazu tako da imate dio podataka na jednom serveru dio na drugom, ali korisniku to izgleda kao da su podaci centralizirani.

Arhitektura zasnovana na klijentu

- Prednosti:
 - Rasterećenje servera jer je aplikacija prebačena na klijenta, a na serveru se nalaze samo podaci.
- Nedostatci:
 - Radi slabog kapaciteta obrade i klijenti postaju opterećeni.
 - Problem kod upgradea jer trebate raditi upgrade na svim klijentskim računalima.

Arhitektura sa ravnopravnim učešćem

- Arhitektura sa ravnopravnim učešćem balansira između klijenta i servera.
- Prezentacijska logika se prebacuje na stranu klijenta.
- Aplikacijska logika se prebacuje dijelom ili cjelokupna na stranu klijenta.



Arhitektura sa ravnopravnim učešćem

- Pohrana podataka i logika pristupa podacima je na strani servera. To su obično Internet aplikacije uobičajeno temeljene na Internet protokolima. Ovo je tipična arhitektura web aplikacija.
- Smještaj aplikacijske logike određuje da li se radi o “thin” ili “fat” klijent arhitekturi.
- Ako je samo prezentacijska logika na strani klijenta onda je to tzv. “thin” klijent. Npr. HTML prikaz web stranice. Ako je već i dio aplikacijske logike prebačen na stranu klijenta onda je tzv. “fat” klijent. Npr. Java Script koji obrađuje podatke sa forme prije no što se podaci upišu u bazu podataka.

Arhitektura sa ravnopravnim učešćem

- Prednosti:
 - Obično su podržani različiti tipovi servera i klijenata pa niste vezani uz jednog proizvođača hardvera kao što je slučaj sa serverski orijentiranom arhitekturom.
 - Ako se radi o web aplikacijama kod kojih je prezentacija praktički standardizirana (kroz HTML) lakše je odvajanja prezentacijske logike od aplikacijske logike.
 - Obično se aplikacija distribuira na više servera pa je otpornost na kvarove veća jer ako padne jedan server drugi može preuzeti njegov posao (ali ovo možete i kod arhitekture na strani klijenta).

Arhitektura sa ravnopravnim učešćem

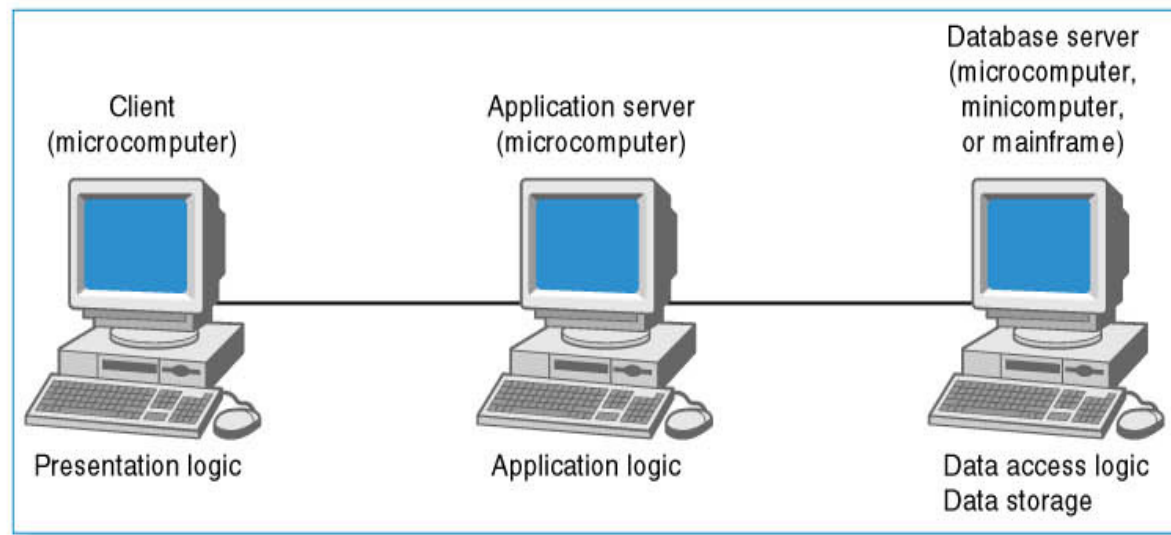
- Nedostatci:
 - Kompleksnost jer je dio aplikacije na strani klijenta, a dio na strani servera pa ih trebate koordinirati. Puno je lakše razvijati aplikaciju koja sve funkcionalnosti ima na jednom mjestu.
 - Novi programski jezici i tehnike se stalno razvijaju jer se ovakve aplikacije šire.
 - Kompleksniji upgrade (osim za web aplikacije) jer se moraju upgradati svi serveri i svi klijenti.

Softverske komponente

- Najjednostavniji način raspodjele logike aplikacije je da se na serveru nalazi pohrana podataka i pristup podacima, dok se cjelokupna aplikacijska i prezentacijska logika nalaze na klijentu.
- Ovakva podjela se naziva dvoslojna (*two-tiered*) arhitektura jer samo dva računala učestvuju u raspodjeli sustava.

Više-slojna arhitektura

- Za razliku od dvoslojne arhitekture višeslojna arhitektura koristi više računala.
- Jedan od pristupa je da se prezentacijska logika stavi na stranu klijenta, aplikacijska logika stavi na posebni aplikacijski serveru, dok se pohrana podataka i logika pristupa smjesti na posebnom podatkovnom serveru.
- Ovo je tzv. troslojna arhitektura (*three-tiered*).



Više-slojna arhitektura

- Podjela aplikacije može ići i na više od tri računala. Takva arhitektura se naziva više-slojnom (*n-tiered*) arhitekturom.
- Neki autori ne odvajaju posebno hardver na kojem se smještaju dijelovi aplikacije, već ukoliko je sam softver odvojen na posebne funkcionalnosti takvu arhitekturu nazivaju slojnom (*layerd ili tiered*) arhitekturom.

Više-slojna arhitektura

- Prednosti:
 - Raspodjeljuje obradu podataka na razne servere i time izjednačava opterećenje među pojedinim serverima
 - Još veća skalabilnost
- Nedostatci:
 - Veće opterećenje mreže
 - Teže za razvoj i testiranje softvera

Izbor dizajna arhitekture

- Svaka arhitektura softverskog sustava ima svoje prednosti i nedostatke.
- Proces odabira arhitekture kreće od nefunkcionalnih zahtjeva.
- Prvi korak je prerada nefunkcionalnih zahtjeva iz faze analize u detaljniji opis nefunkcionalnih zahtjeva iz kojih se onda može odabrati arhitektura.
- Nakon toga se iz detaljnih nefunkcionalnih zahtjeva i dizajna arhitekture radi specifikacija hardvera i softvera.

Nefunkcionalni zahtjevi

- Nefunkcionalni zahtjevi se dijele na četiri osnovna tipa:
 1. Operativni (*operational*) zahtjevi
 2. Zahtjevi za performanse (*performance requirements*)
 3. Sigurnosni (*security*) zahtjevi
 4. Kulturološki/politički (lokalizacijski) (*cultural/political*) zahtjevi

Operativni zahtjevi

- Operativni zahtjevi specificiraju okolinu u kojoj sustav treba raditi i kako se ta okolina može mijenati tijekom vremena. Okolina se odnosi na operacijske sustave, sistemski softver (npr. baza podataka) i druge informacijske sustave s kojima je naš sustav u interakciji. Operativni zahtjevi se dijele na četiri glavna područja:
 1. Tehnički okolinski zahtjevi (*technical environment requirements*)
 2. Sistemski integracijski zahtjevi (*system integration requirements*)
 3. Prijenosni zahtjevi (*portability requirements*)
 4. Zahtjevi prema održavanju (*maintainability requirements*)

Operativni zahtjevi

- Tehnički okolinski zahtjevi specificiraju hardver i softver na koji se naš sustav oslanja tj. na njemu treba raditi. Obično su fokusirani na operacijski sustav (razlika je da li radite aplikaciju za Linux ili Windows OS), softver za baze podataka (razlika je da li radite aplikaciju za Oracle ili za MySQL bazu podataka) i drugi sistemski softver (npr. web aplikacija je prilagođena za Internet Explorer).
- Ponekad to mogu biti zahtjevi prema hardveru, npr. aplikacija se treba izvoditi i na mobitelu -> dimenzije *displeja* itd.

Operativni zahtjevi

- Sistemski integracijski zahtjevi su zahtjevi koji se odnose na interoperabilnost našeg sustava sa nekim drugim informacijskim sustavom.
- Najčešće se interoperabilnost ostvaruje modulom, komponentom ... aplikacije koja omogućava eksport/import podataka iz baze druge aplikacije.

Operativni zahtjevi

- Prijenosni zahtjevi definiraju kako se tehnički okolinski zahtjevi mogu mijenjati tijekom vremena.
- Često se dešava da se poslovne potrebe mijenjaju (npr. obavijest korisniku o poslanoj pošiljci treba biti unutar 1 dana -> ispis iz baze koji se slao poštom treba poslati e-mailom ili možda kao SMS poruku), da se mijenjaju raspoložive tehnologije (danas svi imaju mobitel, prije 10 godina nisu imali), a u skladu s time se treba mijenjati i naš informacijski sustav.

Operativni zahtjevi

- Zahtjevi prema održavanju specificiraju promjene poslovnih zahtjeva koje se mogu predviditi. Npr. recimo da predviđamo da će se otvoriti još jedan dislocirani studij na FESB-u. Kako napraviti bazu studenata ako će postojati još jedna dislocirana lokacija na kojoj će se studenti upisivati?
- Održavanje se također odnosi i na *upgrade* sustava pa se tako definiraju i vremenski razmaci u kojima će se isporučivati nova verzija sustava (*update cycle*).

Operativni zahtjevi

Technical Environment Requirements	Special hardware, software, and network requirements imposed by business requirements	<ul style="list-style-type: none">• The system will work over the Web environment with Internet Explorer.• All office locations will have an always-on network connection to enable real-time database updates.• A version of the system will be provided for customers connecting over the Internet via a small screen PDA.
System Integration Requirements	The extent to which the system will operate with other systems	<ul style="list-style-type: none">• The system must be able to import and export Excel spreadsheets.• The system will read and write to the main inventory database in the inventory system.
Portability Requirements	The extent to which the system will need to operate in other environments	<ul style="list-style-type: none">• The system must be able to work with different operating systems (i.e., Linux; Windows XP)• The system may need to operate with handheld devices such as a Palm.
Maintainability Requirements	Expected business changes to which the system should be able to adapt	<ul style="list-style-type: none">• The system will be able to support more than one manufacturing plant with 6 months advance notice.• New versions of the system will be released every six months.

Zahtjevi za performanse

- Zahtjevi za performanse se odnose na performanse sustava kao što su pouzdanost sustava (*reliability*), kapacitet, vrijeme odziva (*response time*) i sl.
- Svaki zahtjev za performansama treba biti moguće testirati na razvijenom sustavu pa se zajedno sa zahtjevima definiraju i testovi kojima će se provjeriti da li sustav ispunjava dane zahtjeve. Time se sprječava postavljanje loše definiranih zahtjeva za performansama sustava. Npr. ubacivanje podataka sa forme XY u bazu treba biti brzo.

Zahtjevi za performanse

- Zahtjevi za performanse se dijele na tri osnovna tipa:
 1. Zahtjevi za brzinom (*speed requirements*)
 2. Zahtjevi za kapacitetom (*capacity requirements*)
 3. Zahtjevi za dostupnošću i pouzdanošću (*availibilty and reliability requirements*)

Zahtjevi za performanse

- Zahtjevi za brzinom specificiraju kako brzo sustav treba raditi.
- Najvažnije je tzv. vrijeme odgovora (*response time*) sustava koje se definira kao vrijeme koje je potrebno da sustav odgovori na korisnički zahtjev.
- Naravno da je poželjno da vrijeme odgovara na korisnički zahtjev bude što manje, ali čak su i korisnici svjesni da određeni zahtjevi traju. Npr. kada radite neke aktivnosti lokalno npr. tipkate, podrazumijeva se da je odziv sustava trenutni.

Zahtjevi za performanse

- Kada radite neke aktivnosti koje su distribuirane (ne moraju biti samo mrežno, nego i lokalno) npr. dohvaćate podatke iz baze to može trajati. Općenito se smatra prihvatljivim vrijeme odgovara ispod 7 sekundi za aktivnosti koje uključuju mrežni promet.
- Drugi aspekt brzine sustava se odnosi na to koliko se brzo promjene u sustavu propagiraju kroz cijeli sustav. Npr. koliko brzo je narudžba koju je korisnik unio preko web forme promijenila stanje skladište?

Zahtjevi za performanse

- Zahtjevi za kapacitetom se svode na predviđanje broja korisnika koji će koristiti naš sustav i to:
 1. na sveukupni broj korisnika sustava
 2. te na maksimalni broj korisnika koji istovremeno može koristiti sustav.

Zahtjevi za performanse

- Zahtjevi za kapacitetom su bitni za određivanje veličine baza podataka, ali i za određivanje jačine procesora ili određivanje pojasne propusnosti mreže.
- Za procesnu snagu ili propusnost mreže su bitnija tzv. vršna korisnička opterećenja (*peek*) tj. maksimalni broj korisnika u jednom trenutku.
- Naravno da je lakše predviditi broj korisnika u nekom “zatvorenom” sustava (npr. IS banke), nego u nekoj otvorenoj web aplikaciji.

Zahtjevi za performanse

- Zahtjevi za dostupnošću i pouzdanošću se odnose na to koliko dugo (vremenski) je sustav na raspolaganju korisniku.
- Danas nije neuobičajno osobito kod web aplikacija da su aplikacije dostupne korisniku 24x7.
- Kada raditi *backup* baza? Da li organizirati korisničku službu 24 sata?
- [AAS \(Advanced Automation System\)](#) projekt koji je pokrenula FAA (*Federal Aviation Administration*) 1981, a prvi sustav je instaliran 1992 u Seattlu je zahtijevao 99,999999% pouzdanost sustava tj. da sustav ne bude više od 3 sekunde godišnje “spušten”. Za izradu je izabran 1988 IBM kao najbolji ponuđač sa cijenom od 3,6 milijardi. Projekt nije uspio jer je cijena ustvari bila preniska.

Zahtjevi za performanse

Type of Requirement	Definition	Examples
Speed Requirements	The time within which the system must perform its functions	<ul style="list-style-type: none">• Response time must be less than 7 seconds for any transaction over the network• The inventory database must be updated in real time.• Orders will be transmitted to the factory floor every 30 minutes.
Capacity Requirements	The total and peak number of users and the volume of data expected	<ul style="list-style-type: none">• There will be a maximum of 200 simultaneous users at peak use times.• A typical transaction will require the transmission of 10K of data.• The system will store data on approximately 5,000 customers for a total of about 2 MB of data.
Availability and Reliability Requirements	The extent to which the system will be available to the users and the permissible failure rate due to errors.	<ul style="list-style-type: none">• The system should be available 24 x 7, with the exception of scheduled maintenance.• Scheduled maintenance shall not exceed one 6-hour period each month.• The system will have 99% uptime performance.

Sigurnosni zahtjevi

- Sigurnost informacijskog sustava je stupanj zaštite podataka od oštećenja ili gubitka uzrokovanog namjernim postupkom (hakeri i sl.) ili slučajnim događajem (npr. izgori zgrada).
- Sigurnost se postiže standardnim postupcima u održavanju informacijskog sustava kao što su instalacija i održavanje vatrozaštite (*firewall*), sustava za detekciju upada (*intrusion detection*), *backup* podataka.

Sigurnosni zahtjevi

- Međutim sigurnost sustava se ne postiže samo prilikom održavanja sustava već i prilikom dizajna i implementacije sustava pomoću autorizacije i autentikacije korisnika, enkripcije podataka i sl., ali ovi mehanizmi zaštite trebaju biti ugrađeni već tijekom dizajna aplikacije.

Sigurnosni zahtjevi

- Sigurnost postaje sve veći problem zbog razvoja Interneta i prelaska velikog broja aplikacija na Internet. Od početka 80-tih kada je FBI prvi put počeo pratiti računalni kriminal najveća opasnost je dolazila unutar same kompanije pa je 80% slučajeva krađe, neautoriziranog pristupa, sabotáže i sl. dolazilo od samih djelatnika, a samo 20% izvana, dok je danas taj omjer obrnut.

Sigurnosni zahtjevi

- Ispunjavanje sigurnosnih kriterija košta (kao što košta i ispunjava drugih kriterija).
- Koliko potrošiti na povećanje sigurnosti informacijskog sustava?
- Obično se radi tako da se procjeni koliki trošak bi izazvao sigurnosni problem te da se pomnoži sa vjerojatnošću (sve su ovo procjene) problema te se tako dobije koliko se treba potrošiti na sigurnost.
- Npr. ako je potencijalni gubitak 100 milijuna dolara, a vjerojatnost 10% onda se isplati potrošiti 10 milijuna dolara na sigurnost.

Sigurnosni zahtjevi

- Sigurnosni zahtjevi se dijele na četiri osnovne grupe:
 1. Procjena vrijednosti sustava (*system values estimation*)
 2. Zahtjevi za kontrolom pristupa (*access control requirements*)
 3. Zahtjevi za enkripcijom i autentikacijom (*encryption and authentication requirements*)
 4. Kontrola virusa (*virus control requirements*)

Sigurnosni zahtjevi

- Procjena vrijednosti sustava se odnosi na procjenu poslovne vrijednosti sustava i procjenu kako tu poslovnu vrijednost zaštititi.
- Vrlo često je vrijednost podataka puno veća od same infrastrukture i puno teže nadoknadiva.
- Često je sam informacijski sustav vrijednost za sebe. Npr. kompanije koje posluju isključivo preko weba kada padne informacijski sustav uopće ne mogu poslovati. Procjenjuje se da Amazon.com zarađuje oko 1.8 milijuna \$ na sat. Znači ako bi njima “pao” web site 1 sat to bi bio gubitak kompanije od oko 1.8 milijuna \$.

Sigurnosni zahtjevi

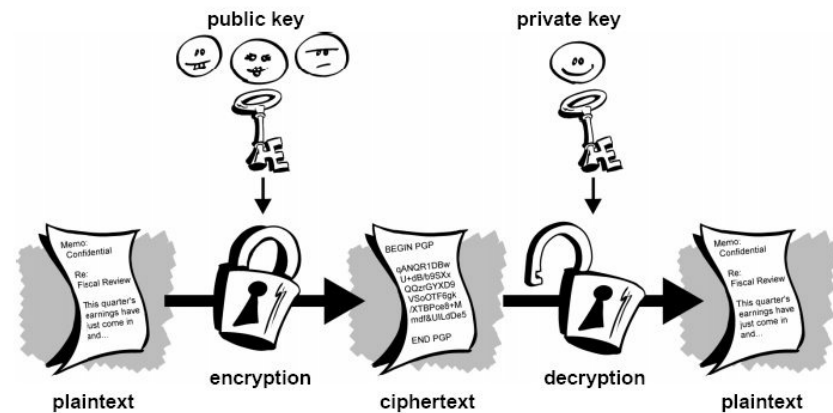
- Zahtjevi za kontrolom pristupa se odnose na podatke koji su tajni i ne trebaju biti dostupni svima.
- Zahtjevi za kontrolom pristupa određuju tko može pristupiti kojim podacima i što sa tim podacima može raditi (brisati ili samo čitati).
- Kontrola pristupa zahtjeva autorizaciju korisnika koja može biti implementirana unutar same aplikacije ili čak može biti napravljena prilikom održavanja sustava (www.fesb.hr/~kiki/korisnicka).

Sigurnosni zahtjevi

- Najbolji način spriječavanja neautoriziranog pristupa podacima je enkripcija podataka.
- Može se koristiti na razini podataka u bazi, može se koristiti na razini prijenosa podataka preko mreže,...
- Dva su osnovna načina enkripcije podataka – simetrična i asimetrična. Kod simetrične enkripcije koristi se isti ključ i za enkripciju i dekripciju, dok se kod asimetrične koriste različiti. Danas se u informacijskim sustavima najčešće koristi asimetrična enkripcija sa parom ključeva privatni/javni.

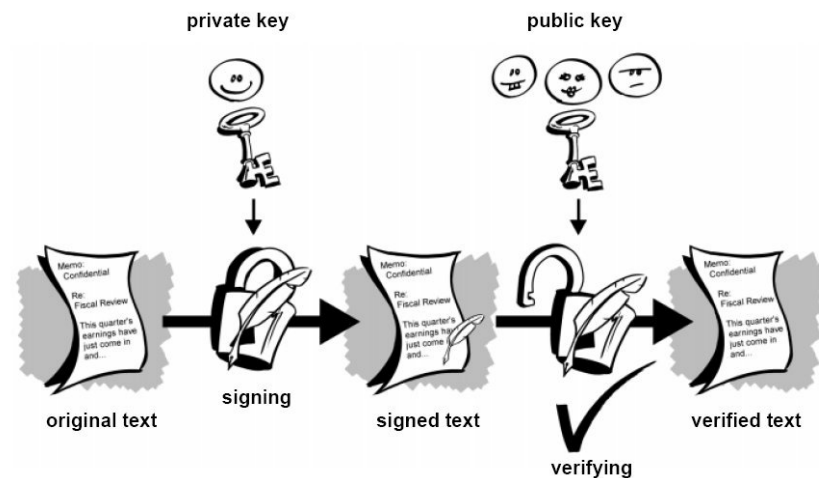
Sigurnosni zahtjevi

- Kao što i samo ime kaže jedan ključ je javni (svima poznat) dok je tajni ključ poznat samo vlasniku ključa.
- Svi mogu kriptirati sadržaj, a samo vlasnik ključa ga može dekriptirati.



Sigurnosni zahtjevi

- Ovaj mehanizam se koristi i kod digitalnog potpisa.
- Digitalni potpis ima istu funkciju kao i običan potpis tj. omogućava primatelju informacije da provjeri da li je pošiljatelj informacije autentičan.



Sigurnosni zahtjevi

- Jedan od najčešćih sigurnosnih problema danas je infekcija sustava sa računalnim virusom.
- Posljedice zaraze mogu biti banalne (popup poruke i sl.) ili mogu biti izuzetno teške (uništavanje podataka).
- Istraživanja pokazuju da se 50% kompanija zarazi bar jednom godišnje.

Sigurnosni zahtjevi

Type of Requirement	Definition	Examples
System Value Estimates	Estimated business value of the system and its data	<ul style="list-style-type: none"> • The system is not mission critical but a system outage is estimated to cost \$50,000 per hour in lost revenue. • A complete loss of all system data is estimated to cost \$20 million.
Access Control Requirements	Limitations on who can access what data	<ul style="list-style-type: none"> • Only department managers will be able to change inventory items within their own department. • Telephone operators will be able to read and create items in the customer file, but cannot change or delete items.
Encryption and Authentication Requirements	Defines what data will be encrypted where and whether authentication will be needed for user access	<ul style="list-style-type: none"> • Data will be encrypted from the user's computer to the Web site to provide secure ordering. • Users logging in from outside the office will be required to authenticate.
Virus Control	Requirements to control the spread of viruses	<ul style="list-style-type: none"> • All uploaded files will be checked for viruses before

Kulturološki/politički zahtjevi

- Kulturološki/politički zahtjevi su zahtjevi specifični za pojedine države ili pojedina područja (zemljopisna, organizacijska) u kojima se sustav implementira. Dijelev se na:
 1. Zahtjeve za višejezičnost (*multilingual requirements*)
 2. Zahtjeve za prilagodbom (*customization requirements*)
 3. Eksplicitno postavljanje “podrazumijevanih” normi (*making unstated norms explicit*)
 4. Pravne zahtjeve (*legal requirements*)

Kulturološki/politički zahtjevi

- Zahtjevi za višejezičnost se rješavaju na dva načina.
- Aplikacija istovremeno može raditi na više jezika (*concurrent multilingual system*).
- Aplikacija ima više verzija za različite jezike i u jednom trenutku samo jedna verzija može biti instalirana i aktivna (*discrete multilingual system*).

Kulturološki/politički zahtjevi

- Zahtjevi za prilagodbom dovode do razlike u aplikaciji u pojedinim dijelovima organizacije. Npr. različiti format adrese od države do države (više kolona u bazi). To naravno komplicira održavanje aplikacije.
- Eksplicitno postavljanje “podrazumijevanih” normi se odnosi na korištenje definiranih normi koje se razlikuju od države do države. Npr. format datuma koji se koristi u aplikaciji (npr. američki datum 04/05/11 – evropski datum 04/05/11), valuta i sl.

Kulturološki/politički zahtjevi

- Pravni zahtjevi se odnose na zakone koji se mogu odnositi na aplikaciju, a razlikuju se među državama.
- Npr. Georgia Institute of Technology je 1997 kažnjena jer je web stranica njihovog kampa za studente u Francuskoj bila na engleskom, a francuski zakon kaže da najveći dio web site u francuskoj treba biti na francuskom.

Kulturološki/politički zahtjevi

Multilingual Requirements	The language in which the system will need to operate	<ul style="list-style-type: none">• The system will operate in English, French, and Spanish.
Customization Requirements	Specification of what aspects of the system can be changed by local users	<ul style="list-style-type: none">• Country managers will be able to define new fields in the product database to capture country-specific information.• Country managers will be able to change the format of the telephone number field in the customer database.
Making Unstated Norms Explicit	Explicitly stating assumptions that differ from country to country	<ul style="list-style-type: none">• All date fields will be explicitly identified as using the month-day-year format.• All weight fields will be explicitly identified as being stated in kilograms.
Legal Requirements	The laws and regulations that impose requirements on the system	<ul style="list-style-type: none">• Personal information about customers cannot be transferred out of European Union countries into the United States.• It is against U.S. federal law to divulge information on who rented what videotape, so access to a customer's rental history is permitted only to regional managers.

Izbor dizajna arhitekture

- Izbor dizajna arhitekture je često vezan uz postojeću hardversku infrastrukturu u kompaniji, npr. ako već postoji mainframe server onda će se vjerojatno i nova aplikacija raditi za takav hardver.
- Ako je potpuno slobodan, izbor arhitekture radi se s obzirom na sve postavljene zahtjeve.

Izbor dizajna arhitekture

- Pored postavljenih zahtjeva uspoređuju se i:
 - Cijena infrastrukture (klijenata, servera i mreže)
 - Cijena razvoja (razvoj klijent-server aplikacija je skuplji nego serverskih aplikacija)
 - Skalabilnost aplikacija (mogućnost povećanja kapaciteta aplikacije bez zahvata u samoj aplikaciji) (Google, Facebook,...)

Izbor dizajna arhitekture

Requirements	Server-Based	Client-Based	Thin Client-Server	Thick Client-Server
Operational Requirements				
System Integration Requirements	✓		✓	✓
Portability Requirements			✓	
Maintainability Requirements	✓		✓	
Performance Requirements				
Speed Requirements			✓	✓
Capacity Requirements			✓	✓
Availability/Reliability Requirements	✓		✓	✓
Security Requirements				
High System Value	✓		✓	
Access Control Requirements	✓			
Encryption/Authentication Requirements			✓	✓
Virus Control Requirements	✓			
Cultural/Political Requirements				
Multilingual Requirements			✓	
Customization Requirements			✓	
Making Unstated Norms Explicit			✓	
Legal Requirements	✓		✓	✓

Specifikacija hardvera i softvera

- Specifikacija hardvera i softvera detaljno opisuje potrebne hardverske i softverske komponente.
- Ponekad je moguće aplikaciju izvoditi na postojećem hardveru i softveru, ali rijetko i obično je potrebno kupiti novi hardver ili softver za funkcioniranje novog sustava tj. napisati specifikaciju.

Specifikacija hardvera i softvera

- Određuje softverske potrebe
 - OS, posebne softverske potrebe
 - Obuka, garancija, održavanje, kupnja licenci
- Određuje hardverske potrebe
 - Server(i), klijenti, periferni uređaji, uređaji za pohranu
 - Minimalna konfiguracija

Čimbenici pri odabiru hardvera i softvera

1. **Functions and Features** What specific functions and features are needed (e.g., size of monitor, software features)
2. **Performance** How fast the hardware and software operates (e.g., processor, number of database writes per second)
3. **Legacy Databases and Systems** How well the hardware and software interacts with legacy systems (e.g., can it write to this database)
4. **Hardware and OS Strategy** What are the future migration plans (e.g., the goal is to have all of one vendor's equipment)
5. **Cost of Ownership** What are the costs beyond purchase (e.g., incremental license costs, annual maintenance, training costs, salary costs)
6. **Political Preferences** People are creatures of habit and are resistant to change, so changes should be minimized
7. **Vendor Performance** Some vendors have reputations or future prospects that are different from those of a specific hardware or software system they currently sell

Primjer specifikacije

Standardni
klijent

Standardni
web server

Standardni
aplikacijski server

Standardni
server baze
podataka

Operating System	<ul style="list-style-type: none"> • Windows • Mozilla 	<ul style="list-style-type: none"> • Linux 	<ul style="list-style-type: none"> • Linux 	<ul style="list-style-type: none"> • Linux
Special Software	<ul style="list-style-type: none"> • Real Audio • Adobe Acrobat Reader 	<ul style="list-style-type: none"> • Apache 	<ul style="list-style-type: none"> • Java 	<ul style="list-style-type: none"> • Oracle
Hardware	<ul style="list-style-type: none"> • 40-gig disk drive • Pentium • 17-inch Monitor 	<ul style="list-style-type: none"> • 80-gig disk drive • Pentium 	<ul style="list-style-type: none"> • 80-gig disk drive • Pentium 	<ul style="list-style-type: none"> • 200-gig disk drive • RAID • Quad Pentium
Network	<ul style="list-style-type: none"> • Always-on Broadband preferred • Dial-up at 56Kbps possible with some performance loss 	<ul style="list-style-type: none"> • Dual 100 Mbps Ethernet 	<ul style="list-style-type: none"> • Dual 100 Mbps Ethernet 	<ul style="list-style-type: none"> • Dual 100 Mbps Ethernet