

Web dizajn i programiranje

Prof. dr.sc. Dragutin Kermek
Sveučilište u Zagrebu
Fakultet organizacije i informatike
Pavlinska 2, Varaždin 42000
dkermek@foi.hr

20. dio

Arhitektura velikih Web/Internet sustava

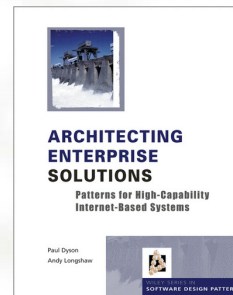
- Postavljanje arhitekture velikih web/internet sustava može se relizirati kroz uzorke (Pattern)
- «Uzorak je rješenje za problem koji se pojavljuje u specifičnom kontekstu» Buschmann, F., Henney, K., Schmidt, D.C. *Pattern-Oriented Software Architecture: On Patterns and Pattern Languages*, Wiley & Sons, 2007.



ARCHITECTING ENTERPRISE SOLUTIONS

- ❑ Dyson, P., Longshaw, A.
*Architecting Enterprise Solutions:
Patterns for High-Capability
Internet-Based Systems*

- ❑ Uzorci za Internet bazirane
sustave velike snage
 - Temeljni uzorci
 - Uzorci performansi sustava
 - Uzorci kontrole sustava
 - Uzorci evolucije sustava



Uvodna razmišljanja o Internet aplikacijama

- ❑ Nefunkcijske karakteristike:
 - Raspoloživost (Availability) – MTBF, MTTF, koliko je sustav korisnicima na raspolaganju
 - Performanse (Performance) – vrijeme odgovora na zahtjev korisnika
 - Skalabilnost (Scalability) – prilagođavanje rada sustava kod povećanja broja korisnika
 - Sigurnost (Security) – stavljanje dovoljne zaštite tako da sustav normalno djeluje uz prihvatljivu razinu privatnosti prema svojim informacijama i kontrolirani pristup do njihovih funkcionalnosti
 - Upravljivost (Manageability) – mogućnost nadgledanja i promjene ponašanja sustava u radu
 - Održavanost (Maintability) – kako se jednostavno uklanja problemi postojećeg sustava ili se neovisno nadograđuju dijelovi sustava
 - Fleksibilnost (Flexibility) – mogućnost stvaranja nove verzije sustava unutar razvojne okoline
 - Portabilnost (Portability) – kako jednostavno sustav može migrirati na nove okoline

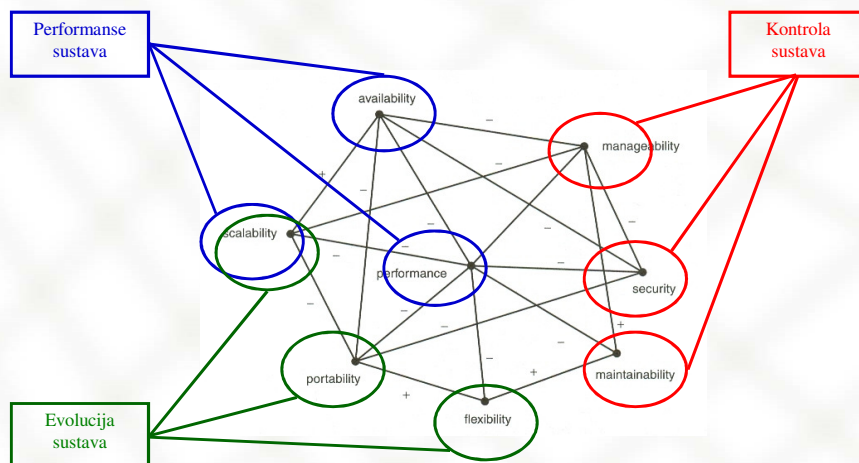


Grupiranje nefunkcijskih karakteristika

- Performanse sustava – sustav treba uvijek biti na raspolaganju i treba uvijek brzo odgovoriti
 - Performanse
 - Raspoloživost
 - Kratkoročna skalabilnost
- Kontrola sustava – sustav treba uvijek biti pod kontrolom
 - Sigurnost
 - Upravljivost
 - Održavanost
- Evolucija sustava – treba omogućiti da se sustav s vremenom jednostavniji promjeni ili mu se doda funkcionalnost ili migrira između tehnologija i platformi
 - Fleksibilnost
 - Portabilnost
 - Dugoročna skalabilnost



Balansiranje nefunkcijskim karakteristikama



Klasifikacija uzoraka za Internet bazirane sustave velike snage

- ❑ Temeljni uzorci
- ❑ Uzorci performansi sustava
- ❑ Uzorci kontrole sustava
- ❑ Uzorci evolucije sustava



Klasifikacija uzoraka za Internet bazirane sustave velike snage

- ❑ Temeljni uzorci
 - APPLICATION SERVER ARCHITECTURE
 - PERIPHERAL SPECIALIST ELEMENTS



Klasifikacija uzoraka za Internet bazirane sustave velike snage

- ❑ Uzorci performansi sustava
 - ACTIVE-REDUNDANT ELEMENTS
 - LOAD-BALANCED ELEMENTS
 - SESSION FAILOVER
 - DEDICATED WEB AND APPLICATION SERVERS
 - COMMON PERSISTENT STORE
 - DATA REPLICATION
 - CONNECTION LIMITATION
 - RESOURCE POOLING
 - LOCAL CACHE
 - OFFLINE REPORTING



Klasifikacija uzoraka za Internet bazirane sustave velike snage

- ❑ Uzorci kontrole sustava
 - CONTINUAL STATUS REPORTING
 - OPERATIONAL MONITORING AND ALERTING
 - 3-CATEGORY LOGGING
 - SYSTEM OVERVIEW
 - DYNAMICALLY-ADJUSTABLE CONFIGURATION
 - DEMILITARIZED ZONE
 - INFORMATION OBSCURITY
 - SECURE CHANNELS
 - KNOWN PARTNERS



Klasifikacija uzoraka za Internet bazirane sustave velike snage

- ❑ Uzorci evolucije sustava
 - DYNAMICALLY-DISCOVERABLE ELEMENTS
 - EXPANDABLE HARDWARE
 - VIRTUAL PLATFORM
 - SWAPPABLE STAGING ENVIRONMENT
 - SEPARATE SYSTEM-MANAGED DATA



Arhitektonski pogledi

- Prema *ISO/IEC 42010 IEEE Std 1471-2000 Systems and software engineering - Recommended practice for architectural description of software-intensive systems* postoji više arhitektonskih pogleda:
 - Logički pogled – opisuje komponente i softverske elemente sustava. On sastavlja softverske elemente sustava u logičke jedinice i opisuje odnose između tih jedinica.
 - Infrastrukturni pogled – pokazuje sklopovske uređaje (servere, rutere, i sl) koje koristi sustav i kako su oni povezani.
 - Instalacijski pogled – pokazuje kako su logičke komponente pridružene različitim sklopovskim elementima sustava.



Gradivi blokovi Internet sustava

- ❑ Web server – HTML, WML, XML, RSS (Apache, MS IIS i dr.)
- ❑ Aplikacijski server
 - J2EE - Sun SJAS, IBM WebSphere, BEA WebLogic, JBoss, Apache Tomcat i Geronimo, Oracle AS i dr.)
 - .NET framework
- ❑ Baza podataka
 - MySQL, PostgreSQL,
 - MS SQL, Oracle, DB2
- ❑ Imenički servisi (LDAP)
- ❑ Usmjernik - ruter (Router)
- ❑ Sigurnosna stijena – vatrozid (Firewall)
- ❑ Raspoređivač opterećenja (Load Balancer)
- ❑ Email server
 - SMTP
 - POP, IMAP
- ❑ Konzola za upravljanje (Management Console)
 - Tivoli, OpenView

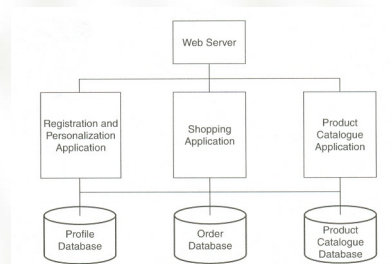


Zamišljena tvrtka GlobalTech

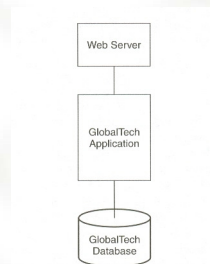
- ❑ Tvrtka proizvodi potrošačka dobra širokog raspona, od jeftinih do vrlo skupih proizvoda
- ❑ Tvrtka ima razne distributere svojih proizvoda, od malih neovisnih distributera za skupe proizvode do multinacionalnih trgovina na malo
- ❑ Služi za prikaz stanja sustava
- ❑ Moguće promjene, poboljšanja



APPLICATION SERVER ARCHITECTURE



Logička arhitektura ranih
Internet sustava

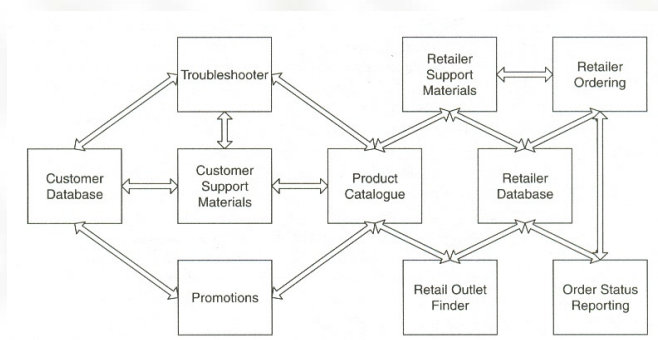


Ciljana logička arhitektura

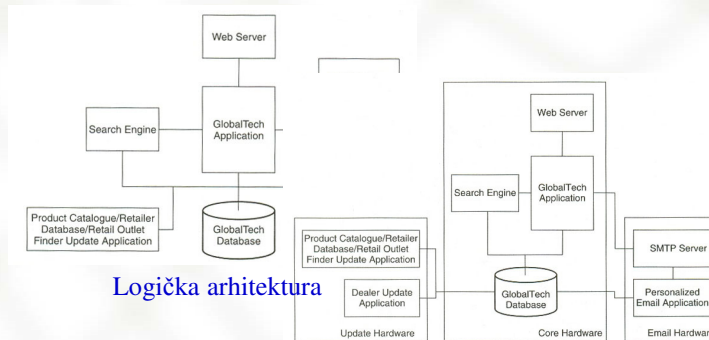


APPLICATION SERVER ARCHITECTURE

- Logička podjela aplikacije u odvojene interesne dijelove



PERIPHERAL SPECIALIST ELEMENTS



Logička arhitektura

Instalacijska arhitektura

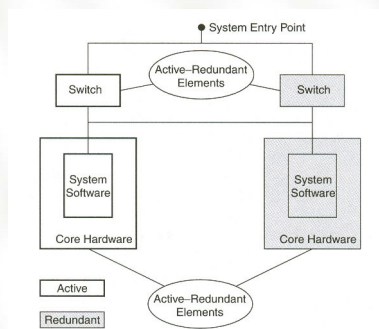
17

Web dizajn i programiranje - FOI
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin



ACTIVE-REDUNDANT ELEMENTS

Dodaju se rezervni (redundantni) elementi koji preuzimaju ulogu u slučaju ispada aktivnih elemenata.



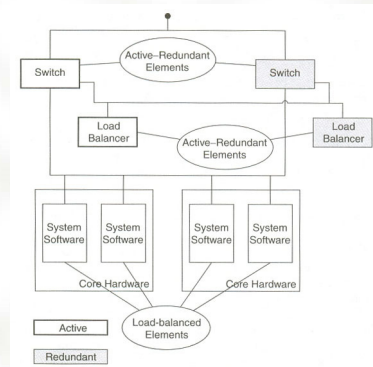
18

Web dizajn i programiranje - FOI
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin



LOAD-BALANCED ELEMENTS

Dodaju se elementi na koje može upravljač opterećenja (Load Balancer) prenosi zahtjevi kako bi se ostvarilo ravnomjerno opterećenje.



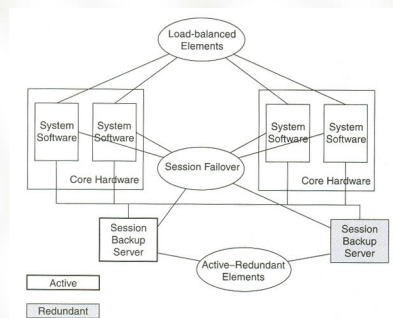
19

Web dizajn i programiranje - FOI
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin



SESSION FAILOVER

Dodaju se elementi koji mogu preuzeti upravljanje sesijama između više elemenata i u slučaju ispada jednog elementa preuzeti/prenijeti sesiju na drugi.



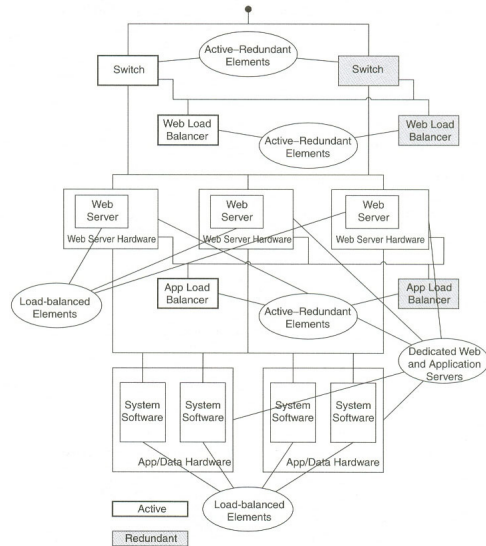
20

Web dizajn i programiranje - FOI
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin



DEDICATED WEB AND APPLICATION SERVERS

Postoje specijalizirani web poslužitelji i specijalizirani aplikacijski poslužitelji kako bi se postavila njihova optimalna konfiguracija s obzirom na specifične poslove koji svako od njih obavlja.



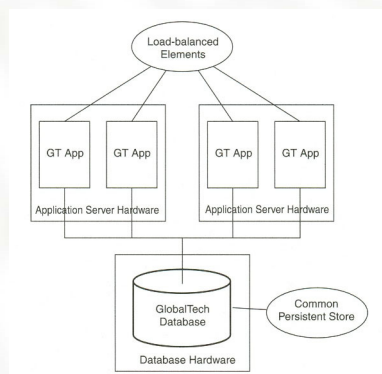
21

Web dizajn i programiranje - FOI
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin



COMMON PERSISTENT STORE

Dodaje se zajedničko premissite podataka za sve aplikacije u sustavu.



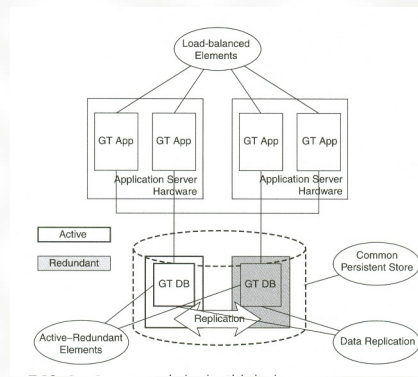
22

Web dizajn i programiranje - FOI
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin

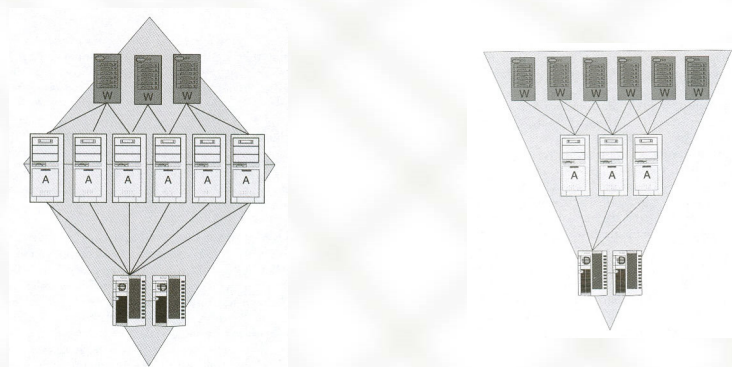


DATA REPLICATION

Zajedničkom spremištu podataka dodaje se aktivno redundarni element između kojih se obavlja replikacija podataka kako bi redundantni element bio uvijek spreman preuzeti ulogu u slučaju ispada aktivnog elementa.



Struktura romba/dijamanta ili obrnuti trokut/piramida



Troškovi?



CONNECTION LIMITATION

- ❑ Može se desiti situacija značajnog interesa za Web mjesto (ili napad)
- ❑ Preveliki broj korisnika degradira usluge za sve korisnike
- ❑ Treba ograničiti broja korisnika koji se mogu spojiti na sustav kako bi se izbjeglo zagušenje resursa i pad performansi



RESOURCE POOLING

- ❑ Kako maksimizirati korištenje ograničenih i/ili skupih resursa?
- ❑ Klasičan primjer: baza podataka
- ❑ Odnos broj korisnika web mjesta i broj korisnika baza podataka (pitanje broja licenci!)



LOCAL CACHE i OFFLINE REPORTING

- ❑ Pravilnim postavljanjem Cache sustva može se značajno utjecati na performanse sustava
- ❑ Razni izvještaji, analize i sl. mogu se obavljati nad podacima koji su pripremljeni i prenešeni na zamjenska računala tako da se mogu preuzeti na računala korisnika.



CONTINUAL STATUS REPORTING

- ❑ Za efikasnu kontrolu sustava potrebne su informacije koje su prikupljane u pravilnim intervalima
- ❑ Pitanja:
 - Koji elementi mogu imati najveći utjecaj na raspoloživost i performanse ako ispadnu?
 - Koji elementi mogu biti usko grlo za performanse i skalabilnost ako se približe maksimalnom kapacitetu?
 - Postoje li bilo koji nepouzdati element?
 - Gdje će elementi zapisivati svoj status?
 - Koji će biti format informacije statusa i kako će ih elementi proslijediti?
 - Koje opterećenje (procesor, memorija, mrežni promet i sl.) će generirati predložena razina nadgledanja?
- SNMP – Simple Network Management Protocol



OPERATIONAL MONITORING AND ALERTING

- ❑ Prikupljene informacije promatraju se za indikatore sustava kojem prijeti pad i upozorava se tim za podršku sustava kako bi mogli poduzeti preventivne akcije ako je potrebno



3-CATEGORY LOGGING

- ❑ Ako je sustav pao ili degradiraju performanse potrebno je znati što je sustav radio u to vrijeme
- ❑ Kategorije podataka:
 - debug – obično sadrži informacije o izvršavanju kao koje su metode bile pozivane na softverskoj komponenti
 - information – jednostavna upozorenja o uvjetima sustava kao što su vremenski prekidi (timeout), nedostajući podaci ili neobičajeni tijek izvođenja
 - error – stvari koje su vrlo nepovoljne kao što je problem kod spajanja s bazom podataka ili gubitak veze između web servera i raspoređivača opterećenja



SYSTEM OVERVIEW

- ❑ Mehanizam za agregiranje nadziranih i dnevničkih podataka koji su se generirali prethodnim načinima
- ❑ Složenost implementacije ovisi o:
 - granulaciji elemenata sustava koji se nadziru
 - količini informacija koja će se agregirati od svakog elementa
 - pogledu na informacije.
- ❑ Za Internetski sustav može biti po jedan predstavnik za:
 - sve web servere
 - sve aplikacijske servere
 - servere u klasteru baza podataka
 - podržavajuće strukture (usmjerivači, preklopnici i sl.).



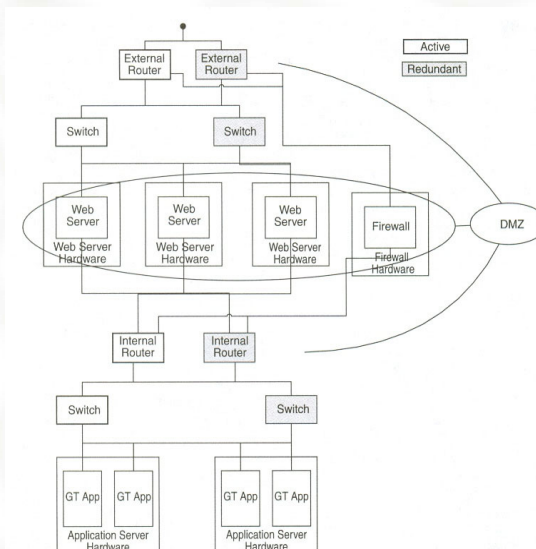
DYNAMICALLY-ADJUSTABLE CONFIGURATION

- ❑ Ako se sustav približava granici nefunkcionalne karakteristike kako se može reagirati na tu promjenu u korištenje bez da se prekine sustav u izvršavanju?
- ❑ Element sustava bit će fleksibilniji ako čita konfiguracijske podatke iz vanjskog izvora
 - kod podizanja – treba restart ili reboot
 - dinamičko – teže implementirati zbog rada sa resursima.
- ❑ Potrebni podaci o:
 - broju simultanih zahtjeva koji se mogu postaviti web serveru
 - broj simultanih sesija koje može održavati aplikacijski server
 - algoritam raspodjeljivača opterećenja
 - broj simultanih veza prema serveru baze podataka
 - veličina cache-a



DEMILITARIZED ZONE

- ❑ Pruža područje sustava koje je odvojeno od vanjskih korisnika i od unutarnjih podataka i funkcionalnosti
- ❑ Uključivanje sigurnosne stijene



INFORMATION OBSCURITY

- ❑ Kako osigurati da prikupljeni osjetljivi podaci koji su pohranjeni u sustavu budu zaštićeni od neautoriziranih pristupa?
- ❑ Treba obaviti gradaciju informacija koje su pohranjene u sustavu u odnosu na osjetljivost.
- ❑ Važnije stavke podataka učiniti nerazumljivim korištenjem tehnika enkripcije i zatamljenja

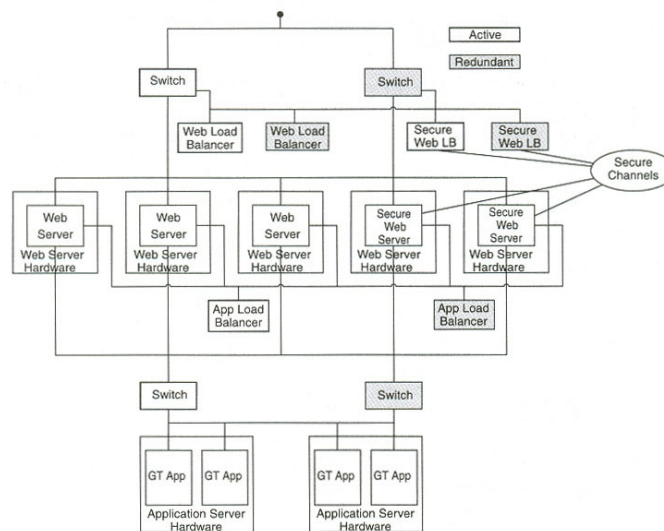


SECURE CHANNELS

- ❑ Kako osigurati da podaci koji se šalju preko javnog ili polujavnog prostora budu zaštićeni u prijenosu?
- ❑ Sigurni kanali se kreiraju tako da učine nerazumnim osjetljive podatke u prijenosu korištenjem enkripcije i klijent-server autentikacije.
- ❑ Virtual Private Network - VPN
- ❑ Secure Socker Layer - SSL



SECURE CHANNELS

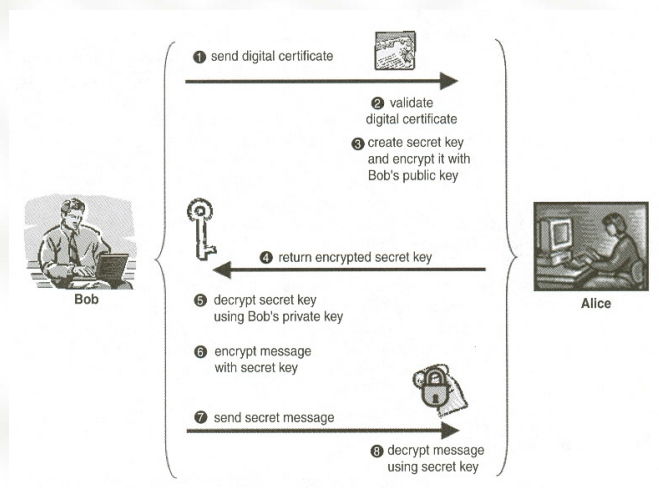


37

Web dizajn i programiranje - FOI
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin



SECURE CHANNELS i KNOWN PARTNERS



38

Web dizajn i programiranje - FOI
Prof.dr.sc. D.Kermek, Fakultet organizacije i informatike, Varaždin

