

# KONFERENCA JAVAONE 2008

V San Franciscu je od 6. do 9. maja 2008 potekala konferenca JavaOne, namenjena platformi java. Sodi nedvomno med največje dogodke informacijske tehnologije, saj ga obišče več tisoč strokovnjakov s tega področja. Letošnja konferenca je potekala pod geslom "Java + You", s čimer so poskušali poudariti dejstvo, da ta tehnologija na tak ali drugačen način vpliva na ekosistem potrošnika, torej na vsakega izmed nas. Med statističnimi podatki tako najdemo, da naj bi bilo v uporabi 6 milijard naprav, ki delujejo na osnovi tehnologije java (<http://www.sun.com/aboutsun/media/presskits/javaone2008/index.jsp>).

Na urniku konference je bilo preko 300 predavanj, ki so potekala v 11 vzporednih sekcijah. Kljub temu je bilo moč čutiti, da je poudarek letošnje konference na proizvodu JavaFX, ki predstavlja jedro platforme java, bogatih spletnih aplikacijah (RIA) in skriptnih jezikih. Težišče razvoja se tako spet nagiba na stran odjemalcev, pri čemer je trend gradnja bogatih spletnih odjemalcev, ki vsebujejo podobne grafične elemente kot klasične namizne aplikacije.

## COMMUNITYONE

Pred glavno konferenco JavaOne je potekala enodnevna konferenca z naslovom CommunityOne 2008. Dogodek je namenjen vsem tistim, ki jih zanima področje odprte in prostodostopne kode. Tako se tukaj zberejo razvijalci, odjemalci in ponudniki tovrstnih proizvodov.

Konferenca je bila razdeljena na več vzporedno organiziranih predavanj.

## Operacijski sistem OpenSolaris

Največji poudarek konference CommunityOne je bil prav OpenSolaris kot SUN-ov prispevek k odprti kodi. Na konferenci so delili tudi DVD-je z operacijskim sistemom.

OpenSolaris je operacijski sistem, ki je v svoji prvi verziji enak SUN-ovemu Solaris OS. To izvorno kodo je SUN predal v odprtokodni projekt (ne celotno kodo, ampak

njene sestavne dele). SUN kot razloge za prehod v odprtokodni način razvoj navaja naslednje:

- Model razvoja z odprto skupnostjo prinaša priložnosti za sodelovanje med Sunovimi inženirji in skupnostjo uporabnikov.
- Veliko uporabnikov sistema Solaris OS je zahtevalo odprto kodo.
- Omogočeno je sodelovanje v mnogih projektih, ki zahtevajo odprto kodno programsko opremo itd.

Trenutne distribucije OpenSolaris delujejo na procesorjih SPARC kot tudi na x64/x8.

## Linux OS

Predavanja so bila namenjena novostim pri različnih distribucijah (openSUSE, Ubuntu in Fedora). Najavili so openSUSE, verzijo 11.0, ki bo vsebovala: KDE 4.0 in 3.5, GNOME 2.22, NetworkManager 0.7, PulseAudio, Taskque, OpenOffice.org 2.4, Banshee 1.0. Največ izboljšav je bil deležen program za nameščanje, ki je poenostavljen in hitrejši.

## NetBeans

Na predstavitev orodja NetBeans IDE so poudarili, da je nova verzija (6.x) bistveno hitrejša, kar je bil prej eden izmed glavnih očitkov. Na samih predstavitev je bilo dejansko videti hitro odzivnost orodja. NetBeans sicer še naprej daje velik poudarek razvoju spletnih aplikacij. Tako so npr. predstavili razhroščevalnik za javascript, ki deluje neodvisno od spletnega pregledovalnika. Podpora skriptnih jezikom je res široka, tako NetBeans podpira tudi PHP na strežniku in Ruby pa tudi delo z aplikacijskimi strežniki; še posebej je dobro podprt GlassFish.

## PRIHAJAJOČE NOVOSTI PLATFORME JAVA

Glede novosti bi lahko dejali, da je java postala zrela tehnologija. Razvoj poteka v skupnosti in se osredotoča predvsem na enostavnejšo uporabo in večjo učinkovitost. Java se seveda spreminja, vendar ne hitro in korenito,

ampak skrbno, s pretehtanimi spremembami in novostmi, ki jih potrebuje informacijska industrija. Kot takšno lahko javo jemljemo kot dolgoročno platformo, ki bo igrala pomembno vlogo naslednja desetletja. JCP (Java Community Process) deluje tako, da se inovacije, ki nastajajo v skupnosti, prenesejo v formalno specifikacijo.

Javo SE 7 pričakujemo v prvi polovici leta 2009. Decembra leta 2006 je bila izdana Java SE 6, ki je uvedla številne novosti in izboljšave (API spletnih storitev, podpora skriptnim jezikom, prenovljeni API za podatkovno bazo, izboljšave Swinga, nadzor in upravljanje, vgrajene anotacije, prevajalnika API, integracija z varnostnimi storitvami, kakovost, združljivost in stabilnost). Trenutno jo uporablja 7 milijonov razvijalcev jave. Vsak mesec je približno 700.000 prenosov JDK-ja, 90,7 odstotkov osebnih računalnikov, ki so priključeni v internet, pa ima nameščeno javo. Iz tega je razvidno, da je javo prešla v fazo zrelosti. Nova verzija jave bo na voljo poleti 2009. Prva polna verzija JavaFX SDK je bila najavljena za jesen 2008. Vmes redno prihajajo posodobitve, kjer je poudarek predvsem na učinkovitosti in popravkih. Trenutno je na voljo Java SE 6u5p "Performance Release", kjer so optimizirane osnovne knjižnice in virtualni stroj. Razvoj temelji na skupnosti in rezultat tega je tudi, da ima prostodostopna distribucija Linuxa Ubuntu od verzije 8.04 naprej integrirano Java SE 6. Pri razvoju Java SE 7 se osredotočajo predvsem na podporo več jezikom. Virtualni stroj, ki je namenjen enemu jeziku, je preteklost. V zadnjem času so v porastu skriptni jeziki, ki omogočajo enostaven in predvsem hiter razvoj novih aplikacij. Projekt Da Vinci Machine se ukvarja z razširitvijo osnovne arhitekture virtualnega stroja java, tako da bo podpiral večjezičnost. Namen je, da bi drugi, predvsem dinamični programski jeziki tekli na (J)VM enako učinkovito kot java.

Modularnost pomeni boljšo ponovno uporabo, tako pri razvoju kot pri nameščanju. Pri razvoju gre za boljšo ponovno uporabo kode, pri nameščanju pa za boljšo ponovno uporabo binarnih datotek. Uvedena je nova ključna beseda "module" in novi format pakiranja (JAM datotečni format za distribucijo).

Osnovni koncepti modularnosti v programskem jeziku java:

- Deklariramo novi model z imenom `org.netbeans.core`, kamor spada tudi razred `Debugger.java`:
 

```
module org.netbeans.core;
package org.netbeans.core;
public class Debugger {
... new ErrorTracker() ...
}
```
- Razred `ErrorTracker.java` iz paketa `org.netbeans.core`. V tem

primeru je dostop do razreda `ErrorTracker` in metode `getErrorLine` omejen znotraj modula `org.netbeans.core`:

```
package org.netbeans.core.utils;
module org.netbeans.core;
module class ErrorTracker {
module int getErrorLine() { ... }
}
```

- Informacijska datoteka modula `org.netbeans/core/module-info.java` opisuje modul in odvisne knjižnice:
 

```
@Version("7.0")
@ImportModule(name="java.se.core", version="1.7+")
module org.netbeans.core;
```

V novi verziji jave bo prav tako veliko pozornosti namenjeno upravljanju in vizualni predstavitvi virtualnega stroja (VisualVM).

Spremembe na strani bogatega odjemalca bodo zajemale tako Swing kot aplete. Javanski programčki bodo doživeli spremembe v načinu namestitve (Java Web Start) in v načinu uporabe, saj jih bo možno uporabljati v obliki majhnih programčkov, ki delujejo tudi zunaj spletnega brskalnika. Na področju bogatih odjemalcev je na pohodu Java FX, ki je podrobneje opisana v nadaljevanju.

Java EE 6 bo v novi različici doživela kar nekaj sprememb, ki pa ne bodo tako obsežne kot pri prejšnji verziji. Pred dvema letoma, ko je bila izdana Java EE 5, je bilo glavno vodilo enostaven razvoj. Do sedaj je bila Java EE 5 prenesena trimilijonkrat kot samostojno ogrodje ali v okviru razvojnega ogrodja Java NetBens. Glede na pripombe se odločajo, kaj bo spremenjeno v naslednji verziji.

Pripombe glede velikosti ogrodja so pripeljale do iskanja rešitev glede pravilne velikosti, ki bi omogočala modularnost, fleksibilnost, osredotočanje na določeno področje in poenostavitve. To bodo rešili s profili (npr. spletni profil, ki bo namenjen razvijalcem internetnih aplikacij, ali profil za telekomunikacijsko industrijo ...) Takšni profili bodo obsegali komponente, ki bodo namenjene določeni skupini razvijalcev ... Vpeljali bodo tudi proces klestenja, pri čemer bodo nekatere komponente postale opsijske (npr. EJB ali JAX-RPC). V ta namen bo nastala specifikacija EJB, 3.1, v lahki obliki. Prav tako bodo delali na razširljivosti ogrodja. Omogočili bodo enostavnejšo uporabo različnih knjižnic in vpeljali skriptne jezike kot prvorazredne programske jezike.

V verziji 6 bodo nadaljevali tudi s paradigmo enostavnega razvoja, s poudarkom na spletnih aplikacijah (*Web Container*). Spletne aplikacije bodo poenostavljene z oznakami za spletne aplikacije, razvoj bo potekal brez

urejanja datotek Web.xml, omogočena bo samostojna registracija tretjih knjižnic, podprt bo razvoj spletnih storitev RESTful (JAX-RS) in razširitev s skriptnimi jeziki.

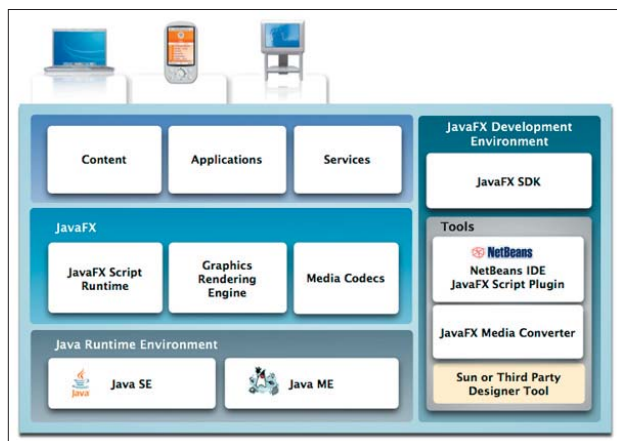
Aplikacijski strežnik GlassFish, V3, je naslednja večja izdaja aplikacijskega strežnika, ki se razvija pod okriljem skupnosti. GlassFish, V3, ima modularno, lahko in razširljivo arhitekturo. Modularna arhitektura omogoča, da je osnovno jedro aplikacijskega strežnika manjše od 100Kb in da se zažene prej kot v sekundi. Glavne značilnosti aplikacijskega strežnika GlassFish, V3:

- odprta, modularna in razširljiva platforma,
- hiter, enostaven in zanesljiv aplikacijski strežnik,
- nameščanje brez ponovnega zagona,
- podpora za različne jezike (JRuby/Rails, Jython/Django, Groovy/Grails, javascript ...),
- podpora OSGi in Java EE 6,
- integracija znotraj razvojnega ogrodja NetBeans.

## NOVA PLATFORMA ZA BOGATE ODJEMALCE JAVA FX

Java FX je družina proizvodov za izdelavo bogatih internetnih aplikacij (RIA). V času aplikacij spleta 2.0 je pomembno predvsem hitro in enostavno razvijanje grafično bogatih aplikacij. Lani so JavaFX prvič predstavili, letos smo lahko videli delujoče internetne aplikacije. Uradno naj bi zaživela z dodatkom za Java SE 6 v drugi polovici tega leta.

Java FX platforma omogoča kreiranje interaktivnih vsebin, aplikacij in storitev, od namiznih sistemov, mobilnih naprav do interaktivne televizije, pri čemer je prednost, da se enaka koda izvaja na različnih napravah.



Slika 1: Razvojno in izvajalno okolje Java FX

V juniju 2008 je napovedana predpremiere Java FX SDK za predhodno registrirane uporabnike, JavaFX Mobile pa na pomlad 2009. Trenutno je preizkušanje možno s pomočjo beta verzije vtičnika za razvojno okolje NetBeans

(JavaFX NetBeans Beta Plugin) ali prevajalnika Java FX, zadnjo verzijo pa lahko dobimo na spletnem naslovu [http://openjfx.java.sun.com/hudson/job/openjfx-compiler/lastSuccessfulBuild/artifact/\\*zip\\*/archive.zip](http://openjfx.java.sun.com/hudson/job/openjfx-compiler/lastSuccessfulBuild/artifact/*zip*/archive.zip).

Na letošnji konferenci je Sun predstavil dva proizvoda iz družine JavaFX, in sicer: Java FX Script in FX Mobile.

Java FX Script je zelo učinkoviti skriptni jezik za kreiranje bogatih odjemalcev in predstavitev multimedijских vsebin na osnovi java tehnologije. Java FX Script poenostavlja proces gradnje bogatih uporabniških vmesnikov za programerje, oblikovalce in avtorje vsebin. S pomočjo Java FX Script se zmožnosti aplikacijskih vmesnikov Java Swing, Java 2D in Java 3D varno prenašajo na različne naprave brez lokalnih nameščenj. Java razvijalci lahko trenutno uporabljajo vtičnik Java FX Script, namenjen razvojnemu okolju NetBeans. Lastnosti:

- Java FX Script uporablja deklarativno sintakso za specifikacijo komponent uporabniških vmesnikov.
- Skozi deklarativno gradnjo dosežemo inkrementalno evalvacijo.
- Java FX bo deloval v vseh glavnih razvojnih okoljih, vključno z NetBeans, ki je referenčna implementacija.
- Za razliko od številnih drugih skriptnih jezikov je Java FX Script statično tipiziran in ima podobno strukturo kode, ki omogoča ponovno uporabo in enkapsulacijo, kar je dobro za vzdrževanje in gradnjo velikih programov.
- Java FX Script je zmožen graditi uporabniške vmesnike vseh velikosti in zahtevnosti.
- Omogoča enostavno uporabo Swinga, ki je eno od najboljših razvojnih ogrodij za uporabniške vmesnike.

Dinamično strukturo jezika dobimo z modelnim pristopom, kjer se sprememba modela avtomatsko preslika na pripadajočo komponento. To dosežemo z metodo "bind", ki predstavlja dvosmerno povezavo.

Sledi primer enostavnega programčka JavaFX Script, ki odpre okno z gumbom. Sprememba modela, ki jo izvedemo s klikom na gumb, se dinamično izvede.

```
import javafx.ui.*;
class HelloWorldModel {
    attribute saying: String;
}
var model = HelloWorldModel {
    saying: "Hello World"
};
Frame {
    title: "Hello World JavaFX"
    width: 200
    content: Button {
```

```

text: bind model.saying
action: function():Void {
    model.saying = "JavaFX is cool"
}
}
visible: true
};

```

Java FX Mobile je kompletno mobilno izvajalno in aplikacijsko okolje, zgrajeno na odprtih tehnologijah java in Linux. Procesorske in omrežne zmožnosti današnjih mobilnih naprav imajo potencial za bogate internetne vsebine. JavaFX Mobile, ki je zgrajena na osnovi platforme Java ME, je tehnologija, ki omogoča izkoriščanje sposobnosti obstoječih naprav. Platformo Java ME ima danes že več kot dve milijardi mobilnih naprav. Sun pričakuje, da bo Java FX Mobile vodilna tehnologija za bogate internetne aplikacije na teh napravah. Celo več, pričakuje se, da bo to tehnologija za vse odjemalce (mobilne naprave, namizni računalniki, TV ...). Ker Sun sodeluje z različnimi partnerji na področju mobilnih naprav tudi na drugih aplikacijskih nivojih, vključno s srednjim slojem, to ponuja dodatne prednosti in gladek ter konsistenten prehod na nove tehnologije v naslednjih letih.

## IZBIRA IN PRIMERJAVA RAZLIČNIH RAZVOJNIH OGRODIJ ZA SPLETNE APLIKACIJE

Prav gotovo je bil eden od glavnih poudarkov letošnje konference JavaOne razvoj spletnih aplikacij. Ena od tem, ki se nanaša na to področje, je izbira pravega ogrodja za razvoj teh aplikacij. Na voljo je ogromno (preko sto) različnih ogrodij, zato je pomembno, da izberemo takšno, ki ustreza našim potrebam. Vsa ogrodja niso enaka, zato je pomembno, da izpostavimo zahteve in najdemo način, da zahteve izpolnimo. V pomoč so nam lahko kriteriji, ki jih Richard Pack (avtor) navaja na osnovi večletnih izkušenj, zakaj je izbira ustreznega ogrodja pomembna:

- Ogrodje je lahko zahtevno za uporabo in razumevanje za sodelujoče v projektu.
- Programerji lahko pišejo kodo, ki ni ponovno uporabna.
- Transport Ajax je neenoten.
- Neposredna trajnost povpraševanja, seje ali konteksta.
- Omejena dokumentacija.
- Neenoten dogodkovni model.
- Neuporabnost za tradicionalne oblikovalce spletnih strani.
- Z naraščanjem aplikacij je konfiguriranje ogrodja oteženo.
- Ni možnosti za uporabo razhroščevalnika.

- Možnost uporabe pogleda na način kontroler.
- Tehnologija JSP predstavlja pri velikih projektih nočno moro.
- Hitrost nameščanja.

Tradicionalni kriteriji za izbiro ogrodja:

- kako dolgo traja, da napišemo aplikacijo, ki nekaj izpiše (test "Hello World"),
- učinkovitost (Filter ali servlet),
- pogled dekoracijskega stila (ali se uporabljajo šablone ali JSP/Taglibs, kontrolni bloki java ...),
- hitrost nameščanja,
- hitrost razvoja, ki jo presojamo s tremi vprašanji (Kako hitro poteka učenje? Koliko časa traja, da dodamo enostavno lastnost? Je spreminjanje lastnosti enostavno?),
- obstoj vgrajenih mehanizmov preverjanja veljavnosti,
- internacionalizacija in večjezičnost,
- dokumentacija,
- možnost zaznamkov,
- skalabilnost,
- možnosti prenosa.

Zmeraj se moramo vprašati, kakšne so naše potrebe?

- Ali gradimo novo aplikacijo?
- Kako velika je ta aplikacija?
- Kakšni so naši scenariji uporabe?
- Ali se bo aplikacija pogosto spreminjala (gradimo za prihodnost)?
- Ali v resnici potrebujem tritirno arhitekturo?
- Kakšna je velikost in kompetenca naše skupine?

Napredni kriteriji:

- podpora Ajaxu (integrirana ali povezana z zunanji orodji),
- podpora skupnosti (aktivni neprekinjen razvoj),
- različni jeziki,
- zrelost,
- podpora tri-tirnosti (Spring, Hibernate),
- prijazni URL-ji,
- odjemalec/strežnik javascript,
- uporaba tega ogrodja tudi v drugem delu podjetja,
- izkušnje razvijalcev s tem ogrodjem.

**Primerjava nekaterih najpogosteje uporabljenih ogrodij za spletne aplikacije:**

### Grovy/Grails

Vrsta uporabe: pogled/napiši

Prednosti	Slabosti
<ul style="list-style-type: none"> <li>hitro nameščanje aplikacij</li> <li>enostavna konfiguracija</li> <li>integrirana Spring in Hibernate</li> <li>lepa izdelava šablon</li> </ul>	<ul style="list-style-type: none"> <li>počasen v primerjavi z osnovnimi ogrodji</li> </ul>

## Google Web Toolkit (GWT)

Vrsta uporabe: visoko interaktivna

Prednosti	Slabosti
<ul style="list-style-type: none"> <li>enostavnost, podobno Swing/SWT</li> <li>veliko vgrajenih gradnikov</li> <li>močna povezava med javo in javascriptom</li> <li>projektna zasnova</li> </ul>	<ul style="list-style-type: none"> <li>težavno nadzorovanje razredov css</li> <li>vgrajene samo osnovne komponente (rešeno z GWT-Ext)</li> <li>trenutno ni podpore za Javo 5/6</li> </ul>

## Struts 2

Vrsta uporabe: pogled/napiši, visoko interaktivna

Prednosti	Slabosti
<ul style="list-style-type: none"> <li>prvorazredna podpora za Ajax</li> <li>integracija z vsebnikom Spring</li> <li>kontrolerji POJO ne zahtevajo dedovanja</li> </ul>	<ul style="list-style-type: none"> <li>še zmeraj uporablja veliko konfiguracije XML</li> <li>mapiranje v XML ostaja tekoče</li> </ul>

## Tapestry

Vrsta uporabe: pogled/napiši, visoko interaktivna

Prednosti	Slabosti
<ul style="list-style-type: none"> <li>uporablja označene šablone HTML</li> <li>učinkovito nameščanje</li> <li>enostavna konfiguracija</li> <li>prvorazredna podpora Ajaxu</li> <li>lep graditelj form</li> </ul>	<ul style="list-style-type: none"> <li>dokumentacija ne zajema realnih problemov</li> <li>ni združljivosti za nazaj med glavnimi verzijami</li> </ul>

## Wicket

Vrsta uporabe: visoko interaktivna

Prednosti	Slabosti
<ul style="list-style-type: none"> <li>uporablja šablone označene HTML</li> <li>odlična dokumentacija</li> <li>komponente so razširljive</li> <li>vse narejeno v javi</li> </ul>	<ul style="list-style-type: none"> <li>vse narejeno v javi</li> </ul>

	Groovy/Grails	Google Web Toolkit	Struts 2	Tapestry	Wicket
Podpora Ajax	da	da	da	razširjena	razširjena
Jeziki	java, Groovy	java - generirana	JSP, OGNL	OGNL, Prop	OGNL
Delovanje	push	pull	push	pull	pull
Zrelost	sorazmerno mlad	mlad	mlad	zrel	zmerna
Skupnost	majhna, vendar raste	majhna	srednja	velika, aktivna	srednja, aktivna
Podpora tritirnosti	da	ne	da	da	da
Prijazni URL-ji	ne	da (z malo dela)	niso vgrajeni	da	da

Tabela 1: Primerjava razvojnih ogrodij za spletne aplikacije

## SPLETNE STORITVE RESTFUL

Na področju spletnih storitev imamo različne arhitekturne pristope. Na eni strani je to storitvena arhitektura (SOA), na drugi pa arhitektura virov (ROA), ki temelji na modelu REST (REpresentational State Transfer). Kljub temu da model REST ni nekaj novega, šele v zadnjem času pridobiva na popularnosti. V okviru procesa skupnosti java se pod oznako JSR 311 (JAX-RS: The Java API for RESTful Web Services) razvija specifikacija, ki določa aplikacijski vmesnik za spletne storitve REST. V razvoju so implementacije Restlet, JBoss RESTEasy, Apache CXF, Jersey ...

REST v petih korakih:

- Viri imajo svoj ID, ki je kar URI (npr. <http://example.com/widgets/foo>). Vir je neki objekt, ki ga želimo nasloviti. Pomembno je, da naslov resnično vsebuje podatke in da imajo le-ti predvidljivo strukturo. ID določimo z anotacijo `@Path`, relativno glede na namestitveni kontekst. Anotacije lahko uporabimo na razredih in metodah. Za spremenljive dele uporabimo parametre.

```
@Path("orders/{order_id}")
public class OrderResource {
    @Path("customer")
    CustomerResource getCustomer(...) {...}
}
```

- Povezovanje virov.

```
<order>
  <prop self="http://example.com/orders/101230">
  <customer ref="http://example.com/customers/
```



```

bar">
<product ref="http://example.com/products/21
034"/>
<amount value="1"/>
</order>

```

Informacije glede namestitvenega konteksta ponuja UriInfo. UriBuilder ponuja možnosti enostavnega kreiranja URI-jev za vire.

```

@Context UriInfo i;
OrderResource r = ...
UriBuilder b = i.getBaseUriBuilder();
URI u = b.path(OrderResource.class).build(r.id);
List<URI> ancestors = i.getAncestorResourceURIs();
URI parent = ancestors.get(ancestors.size()-1);

```

3. Uporabljamo standardne metode, ki jih zagotavlja enotni vmesnik: GET (preberi, pridobi), POST (posodobi ali kreiraj brez ključa), PUT (posodobi ali kreiraj z id-jem), DELETE (briši).

```

@Path("orders/{order_id}")
public class OrderResource {
    @GET
    Order getOrder(@PathParam("order_id") String
id) {
        ...
    }
}

```

4. Različne predstavitve podatkov. Različni formati (HTML, PGN, GIF), različne strukture (XML, JSON), podpora metapodatkom o vsebini s pomočjo glave HTTP. Za določanje predstavitve podatkov uporabljamo anotacije @ProduceMime in @ConsumeMime.
5. Komunikacije brez stanja: dolgotrajni identifikatorji, izogibanje sejam, povpraševanje se izvede neodvisno od preostalih zahtevkov, vse, kar je potrebno za proces, vsebuje povpraševanje samo.

Glavne prednosti na strani strežnika so: horizontalno skaliranje, izbira alternativnega načina delovanja, shranjevanje v predpomnilniku, zmanjšana sklopljenost, dobro delovanje z obstoječo spletno infrastrukturo. Na strani odjemalca omogoča možnost zaznamkov, enostavno eksperimentiranje na brskalniku, široko podporo programskih jezikov, izbiro podatkovnih formatov ...

Model REST upošteva lastnosti in rešitve obstoječega standarda HTTP in preizkušenih vzorcev. Če model REST primerjamo s storitveno arhitekturo (SOA), lahko ugotovimo pomanjkljivosti pri varnosti in transakcijah. SOA ima prednost pri velikih kompleksnih sistemih,

kjer z orkestracijo storitev sistem razslojimo in zagotovimo fleksibilnost. Na drugi strani lahko REST s pridom uporabljamo na različnih napravah, kjer standardi SOA še niso upoštevani. V naslednjih letih se bo dejansko pokazalo, na katerih področjih ima določena arhitektura prednost. Kot kaže sedaj, bo SOA prevladala predvsem pri komunikaciji med znanimi entitetami, npr. znotraj podjetja ali med poslovnimi organizacijami B2B. Arhitektura virov pa bo uporabljena tam, kjer spletni vmesnik predstavlja določeno konkurenčno prednost in tam, kjer dostopamo do zunanjih podatkov.

## AJAX

Ajax je pred nekaj leti prinesel možnost gradnje bogatih spletnih aplikacij. Ajax (Asynchronous JavaScript XML) temelji na javascriptovem ukazu XMLHttpRequest, ki spreminja klasični model gradnje spletne aplikacije. XMLHttpRequest omogoča asinhrono komunikacijo med odjemalcem in spletnim strežnikom. To pa pomeni, da v aplikaciji ni treba vedno znova prenašati spletne strani, ampak lahko na sami spletni strani gradimo grafične elemente, podobne tistim, ki jih poznamo v namiznih aplikacijah. Kljub temu da sta bila javascript in XMLHttpRequest uvedena že leta 1997 v Internet Explorer, pa Ajax sedaj dejansko postaja ključna tema pri gradnji spletnih aplikacij. Veliko napora je trenutno vloženo v standardizacijo (npr. OpenAjax Alliance). Obstoječi pristopi v platformi java so zaenkrat razpršeni in neenotni, obstajajo številne knjižnice (RICO, Prototype, Dojo, Zimbra), ki pa nimajo enotnih API-jev, zato so uporabniki teh knjižnic vezani na izbrano knjižnico. Pričakovati je, da se bo to področje standardiziralo preko projektov odprte kode, JCP-ja in preko iniciative OpenAJAX Alliance. Drug vidik gradnje spletnih aplikacij Ajax je manko ustreznih orodij. Na tem področju je velik napredek nova verzija NetBeans, ki vsebuje učinkovito podporo javascriptu, vključno z razhroščevalnikom, ki je neodvisen od spletnega pregledovalnika.

Marjan Vaupotič, Robert Vehovec