

2. Уеб дизайн



доц. д-р Тодорка Глушкова
гл. ас. д-р Мая Стоева

Тема 2

- Семантичен WEB
- Архитектура
- Smart Clients
- Комуникация със сървъра- протоколи, портове
- Адресиране в Интернет
- HTTP
- Потребителски сесии и състояния

УЕБ 3.0. Семантичен Уеб

- Терминът Semantic Web е въведен от Тим Бърнърс-Лий, създателя на WWW и директор на WWW Consortium (W3C), който ръководи развитието на технологии и стандарти, свързани със семантичния Web.
- Той определя Semantic Web като **"мрежа от данни, които могат да се обработват директно и индиректно от машини."**
- Тим Бърнърс-Лий изразява първоначално визията за Semantic Web, като «Имам една мечта за Web , при която компютрите стават способни за анализ на всички данни в Web съдържанието, връзките и сделките между хората и компютрите»
- Пълната реализация все още предстои да се случи.

Семантичен Уеб

"Семантичният Web осигурява обща рамка, която да позволява данните да се споделят и използват повторно".

- Семантичният Web се разработва като информационно пространство, полезно не само за комуникация човек с човек, но и като пространство в което могат ефективно да си сътрудничат и компютрите.
- Едно от главните препятствия за реализиране на тази идея се състои в това, че голяма част от информацията в Web е предназначена за разбиране от човек и че подобна структура е неразбираема за компютрите.
- За преодоляване на това препятствие се разработват езици за описание на информацията във форма пригодна за машинна обработка.

УЕБ 3.0. Семантичен Уеб

- Семантичният уеб е разширение на сегашния (синтактичен) уеб, при което информацията е представена с добре дефинирано значение, което позволява по-добра кооперативна работа между хора и компютри.
- Докато синтактичният УЕБ работи с база данни, семантичният УЕБ използва база знания
- Основните езици за представяне на информацията в семантичния УЕБ– RDF, RDFS и OWL.

Web браузърът- нашият персонален асистент

- Браузърът следи и запомня от какво се интересува потребителя;
- Браузърът се консултира със запомнената информация за предишните желания;
- Браузърът локализира потребителя и предлага най-добрите възможности според местоположението

Web 3.0 и Web-дизайна

- Опростяване на дизайна;
- насочване вниманието на потребителя към най-важното.



Цел на семантичния Уеб

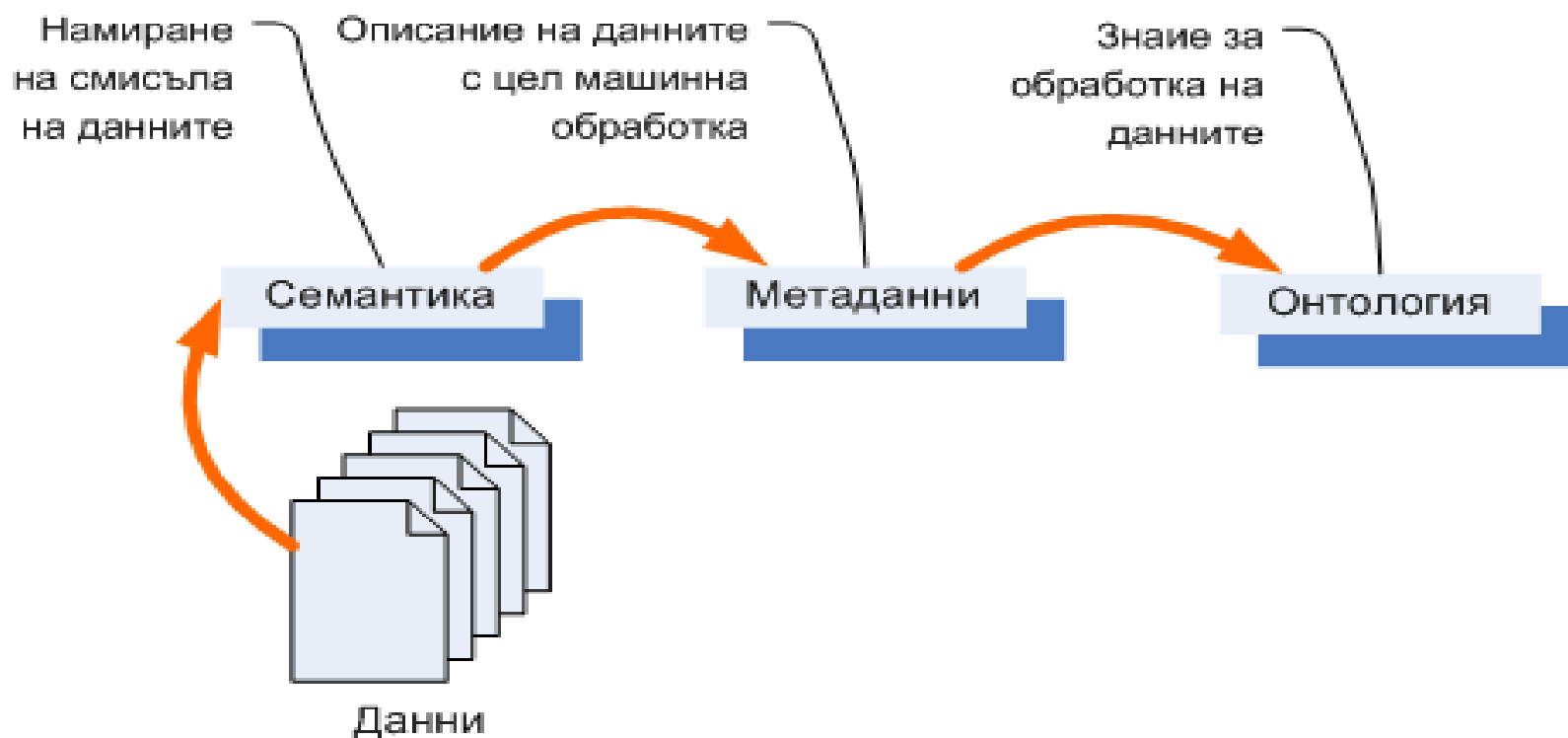
- Основната цел е да се превърне в база за еволюцията на Web, като дава възможност на потребителите да намерят, споделят и комбинират информация по-лесно.
- За да се реализира тази цел е необходимо компютрите да «разбират» семантичната (смислова) същност на информация, предадена им от хора или други компютри. Подобно "разбиране" изисква съответните източници на информация да бъдат семантично структурирани.
- Съвременната тенденция в развитието на Интернет се заключава в преход от документи „читаеми от компютрите“ (machine readable) към документи „разбираеми от компютрите“ (machine understandable).
- Следващият етап в развитието на Интернет – Semantic Web – представлява преход към ново ниво на представяне на данните – ниво на знания и автоматизирана обработка.

Същност на семантичния Уеб

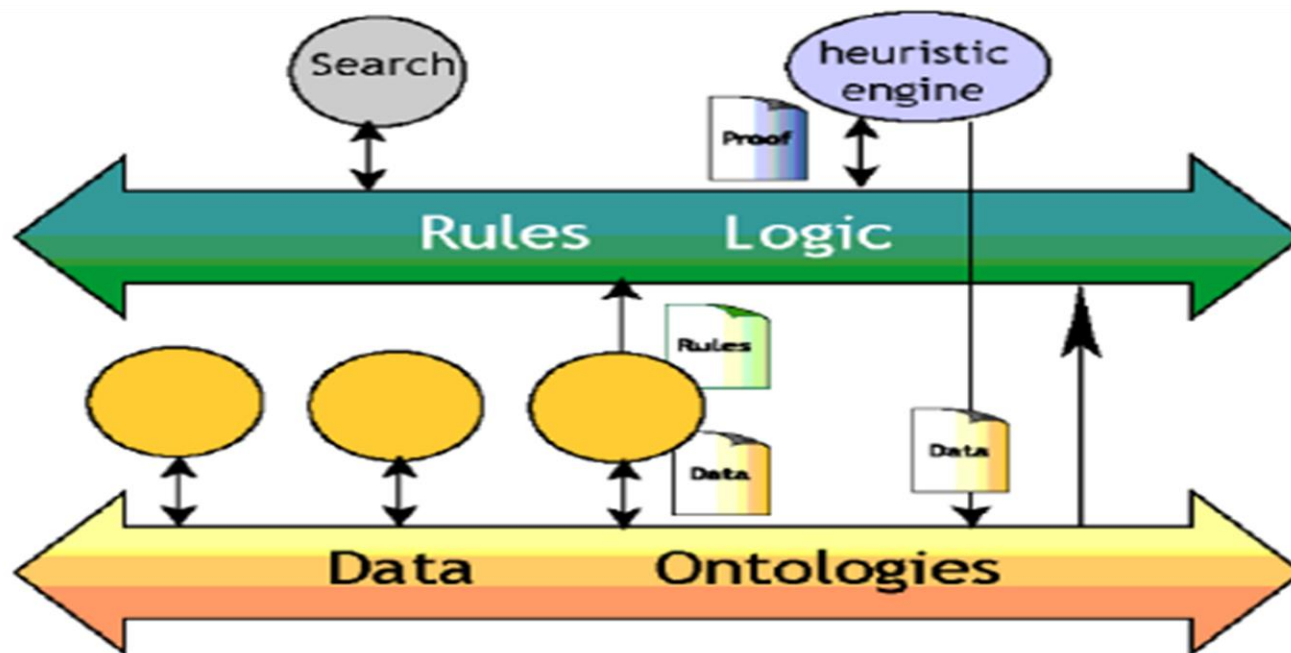
- Semantic Web представлява мрежа от информационни възли, които са свързани помежду си по такъв начин, че съществуващата информация може лесно да се обработва семантично от компютрите.
- Semantic Web може да се разглежда като ефективен начин за представяне на данни в Интернет, или като глобално свързана база данни.
- Използват се стандарти, които са разработени не само от консорциума W3C, но и други организации, като задачата е да се осигури взаимодействие на тези стандарти между себе си в децентрализирана система, без участие на човека.
- Според професор Джон Сова, – Semantic Web е много - дисциплинарна тема, която обединява теориите и методите в три области: логика, онтология и теория на моделите.

Основни термини в семантичния Web

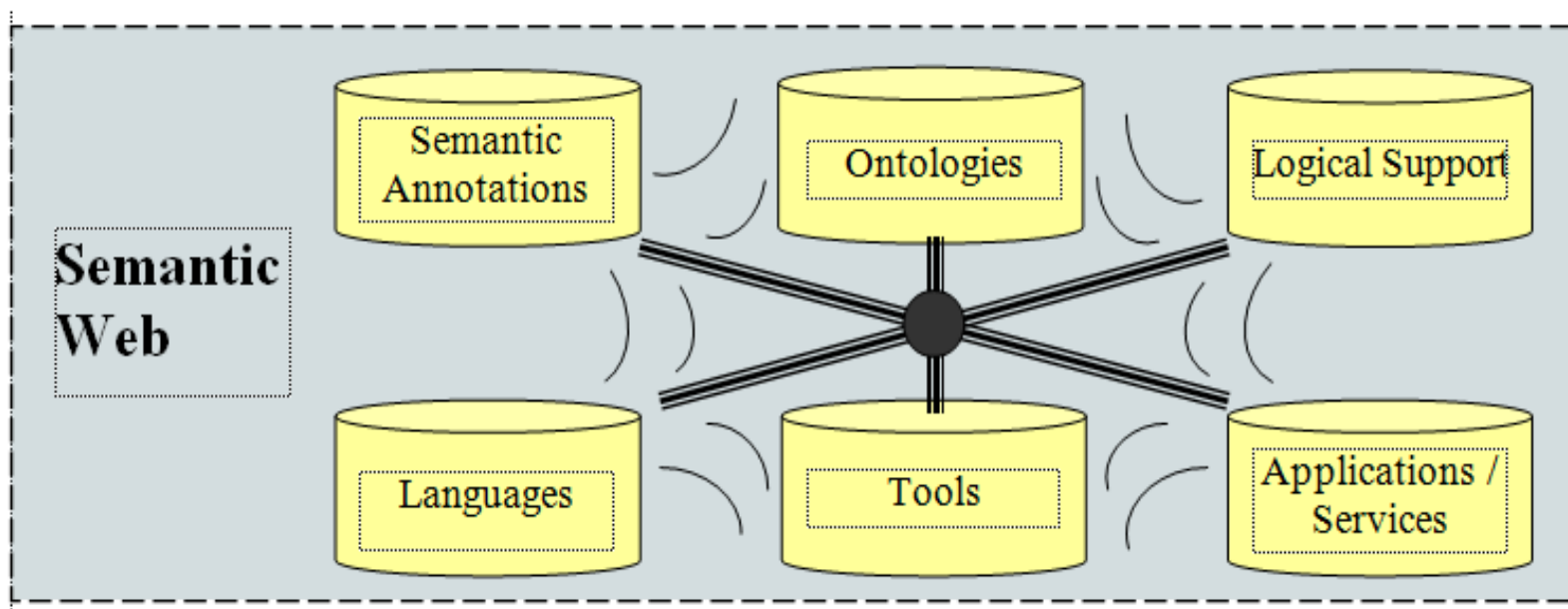
Термините "семантика", "метаданни", "онтологии" стоят в основата на семантичния Web.



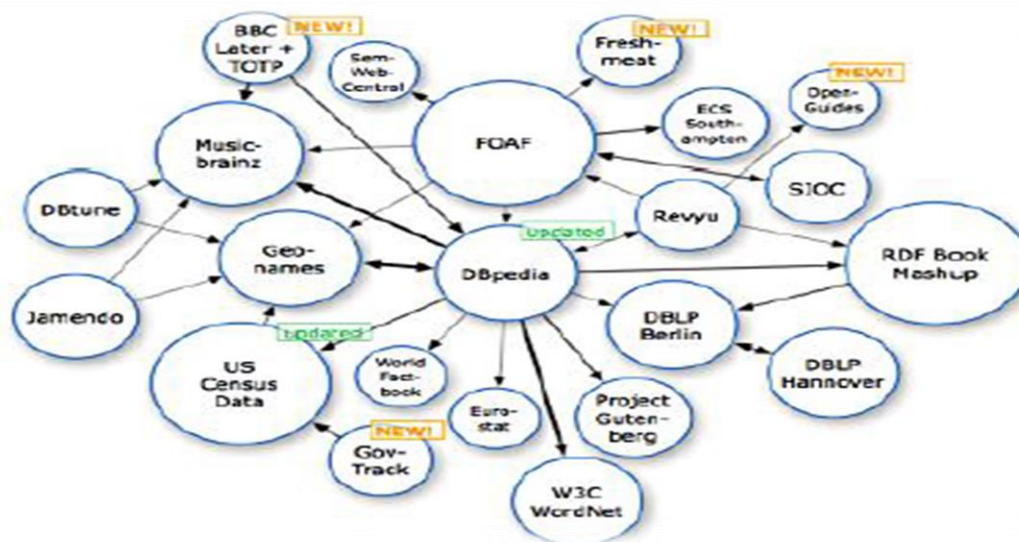
Семантичен УЕБ- управляван от правила



Средства за изграждане на Уеб 3.0



Онтология



Формална спецификация, дефинираща отношения между група от термини

Формално описание чрез: XML (eXtensible Markup Language); RDF (Resource Description Framework)

RDF-представяне



Два обекта и отношение между тях

RDF използва URI (Uniform Resources Identifier)

Езици и речници

- Описанието на отношенията изисква документи, чрез които да се представят в УЕБ думи и логика, чрез която да се описват връзките между тези думи.
- **Онтологията(ontology)** е речник, описващ обектите и отношенията между тях
- **Схемата(schema)** е метод за организиране на информацията

Някои средства за работа със СХЕМИ И ОНТОЛОГИИ

- **RDF Vocabulary Description Language schema (RDFS)** -създава базовата езикова рамка (framework) посредством добавяне на класове, подкласове и свойства към ресурсите;
- **Web Ontology Language (OWL)**–формализира онтоологиите, описва отношения между класовете и използва логика за извършване на дедукции. Може да формира нови класове на базата на съществуващата информация.

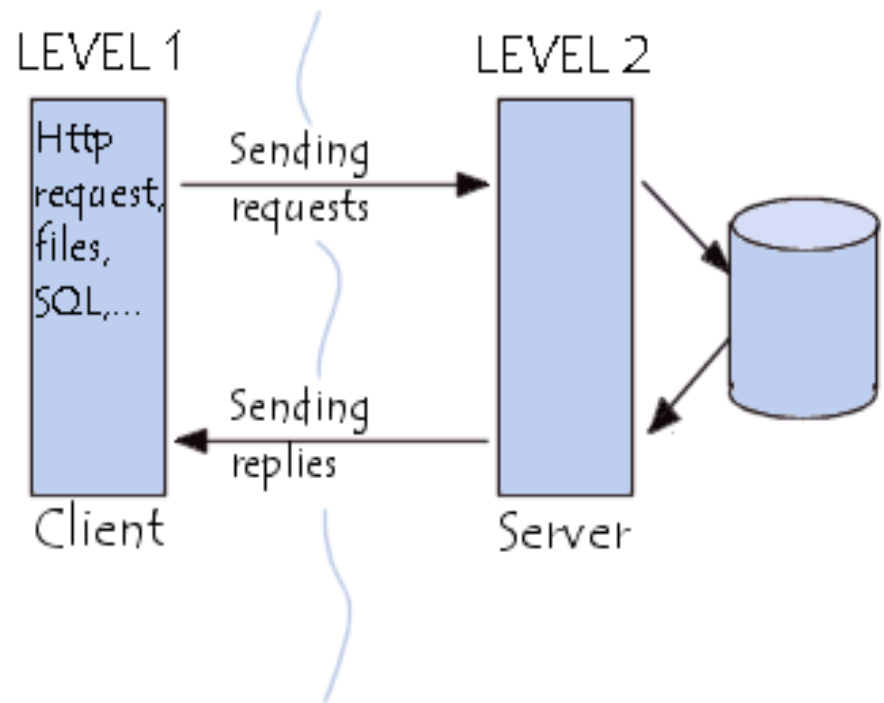
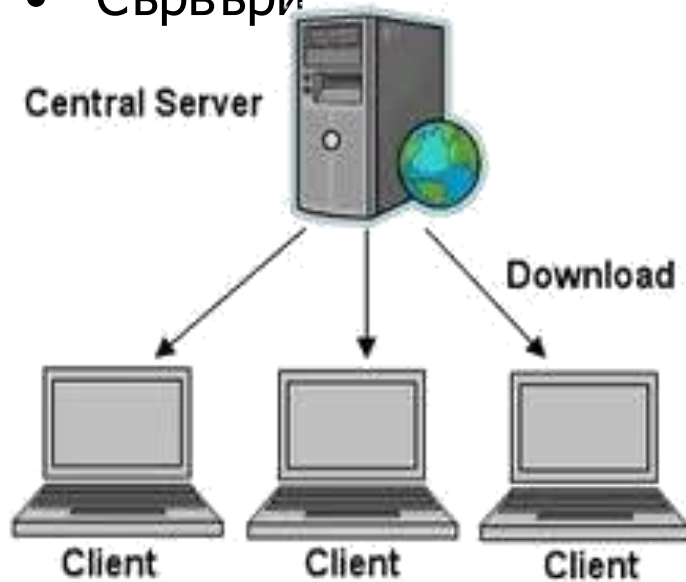
Проблеми при работа с ОНТОЛОГИИ

- Трудни за създаване, реализиране и поддържане;
- В зависимост от областта могат да включват огромни на брой дефиниции и отношения.

Препоръчва се да се използват логика и правила съвместно с онтологите.

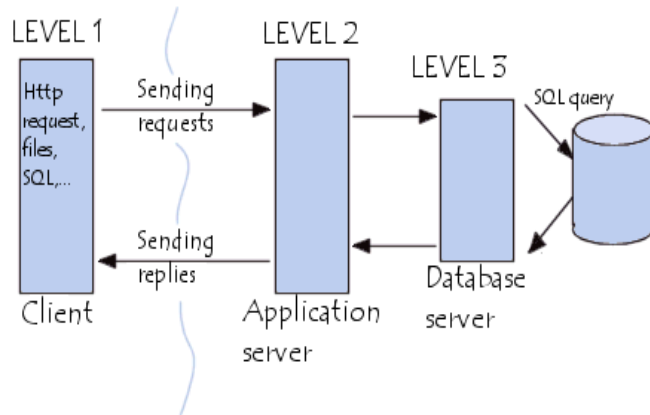
Архитектура на Уеб: клиент-сървър модел

- Двуслоен модел:
- Клиенти
- Сървъри



Архитектура на Уеб: трислоен модел

- ❑ Презентационен слой
- ❑ Бизнес (логически) слой
- ❑ Слой данни



Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.



Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.



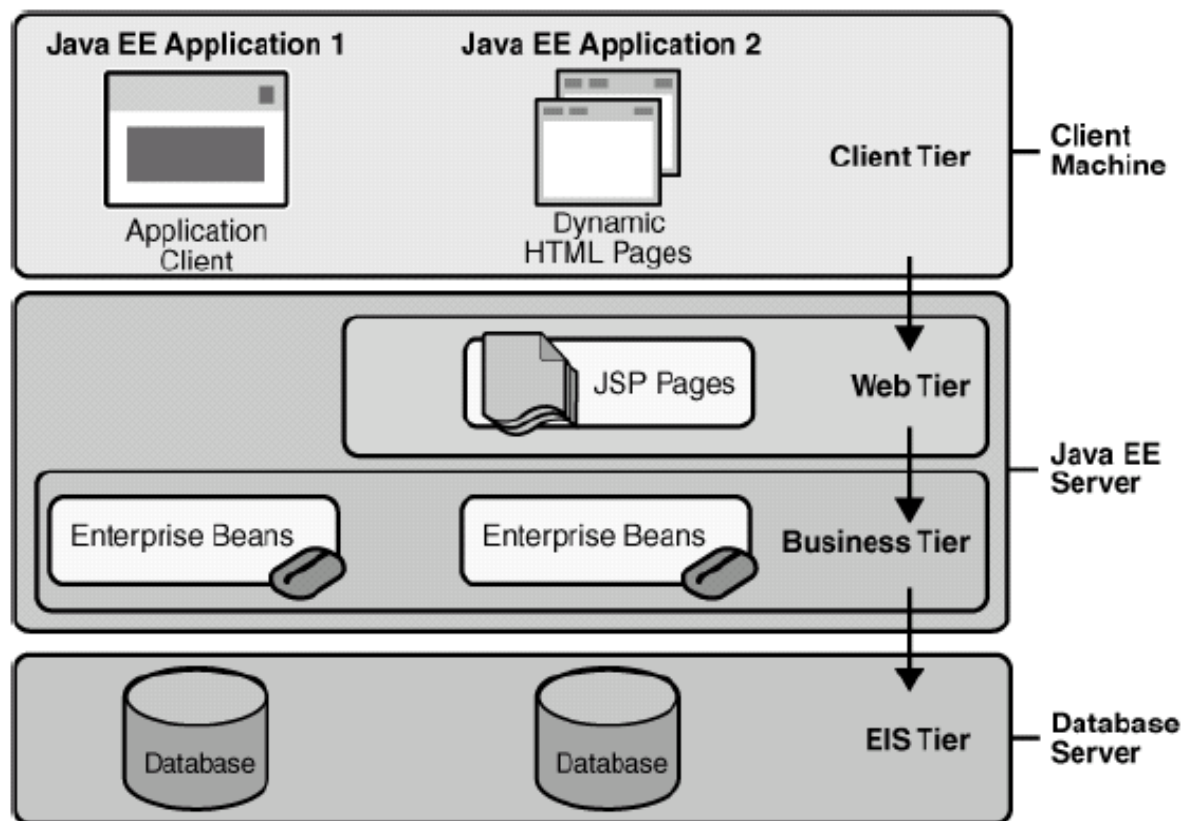
Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

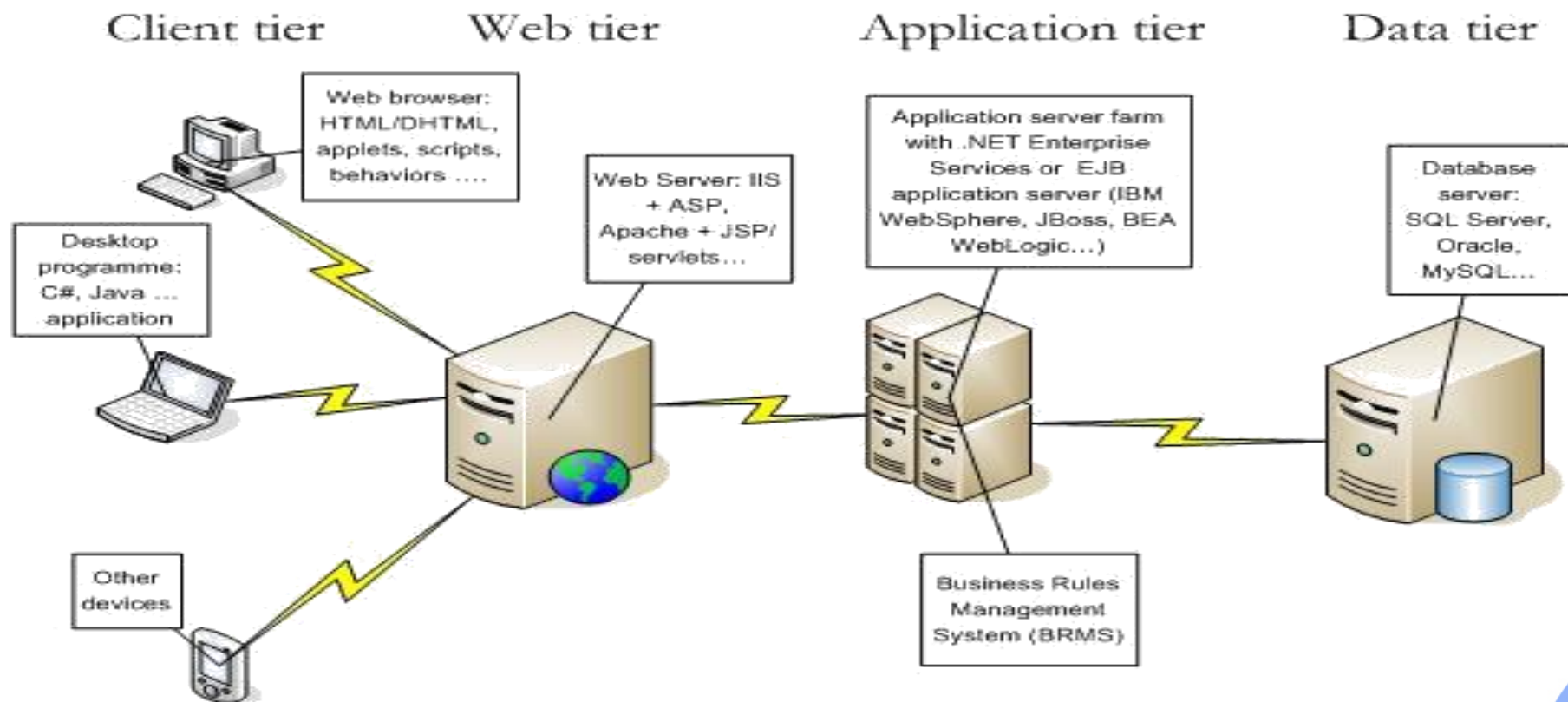


Архитектура на Уеб: N-слоен модел

- Презента-
ционен
(клиентски)
слой
- Уеб слой
- Бизнес слой
- Слой данни



Технологии за изграждане на слоевете



Клиентски слой

- Курсът по Уеб дизайн разглежда изграждането на приложния (клиентски) слой на Уеб приложенията
- Клиентският слой на Уеб приложенията може да бъде два вида:
 - **в браузър** – Уеб браузер зарежда Уеб страници с хипертекст и маркирано съдържание (HTML, XML, WML, *ML) и евентуален клиентски скриптов код (JavaScript, Flex или Java аplet)
 - **извън браузър** – на клиентската машина се изпълнява самостоятелно софтуерно приложение, без използването на Уеб браузер

Фокус на курса: клиентски слой в браузър

Курсът разглежда презентационен
слой в Уеб браузър.

Видове Уеб клиенти

- **«Тънък» клиент (англ. *thin client*)** - клиентска програма с клиент-сървърна архитектура, която пренася всички задачи по обработка на информация към сървъра. По този начин, за работата на 'тънкия' клиент е необходим сравнително мощен сървър. Типичен пример за тънък клиент е компютър с браузер, използван за представяне на презентационен слой с маркирано съдържание (HTML).
- **«Дебел» клиент (англ. *thick client*)** - клиентската програма обработва информация независимо от сървъра, като използва сървъра само за съхранение на данни.

Предимства и недостатъци на тънкия клиент

- **Предимства:**

- Лесна дистрибуция и управление на версиите – на нулева цена (zero-install, auto-update)
- Ниски изисквания към клиентската машина
- Прост клиент
- Липса на изисквания за сигурност към клиента

- **Недостатъци:**

- Липса на разширяемост
- По-беден потребителски интерфейс
- Ниска продуктивност и мощност

Тенденция: умен клиент (Smart Client)



- Источник: <http://khason.net/blog/action-required-smart-client-users-group/>

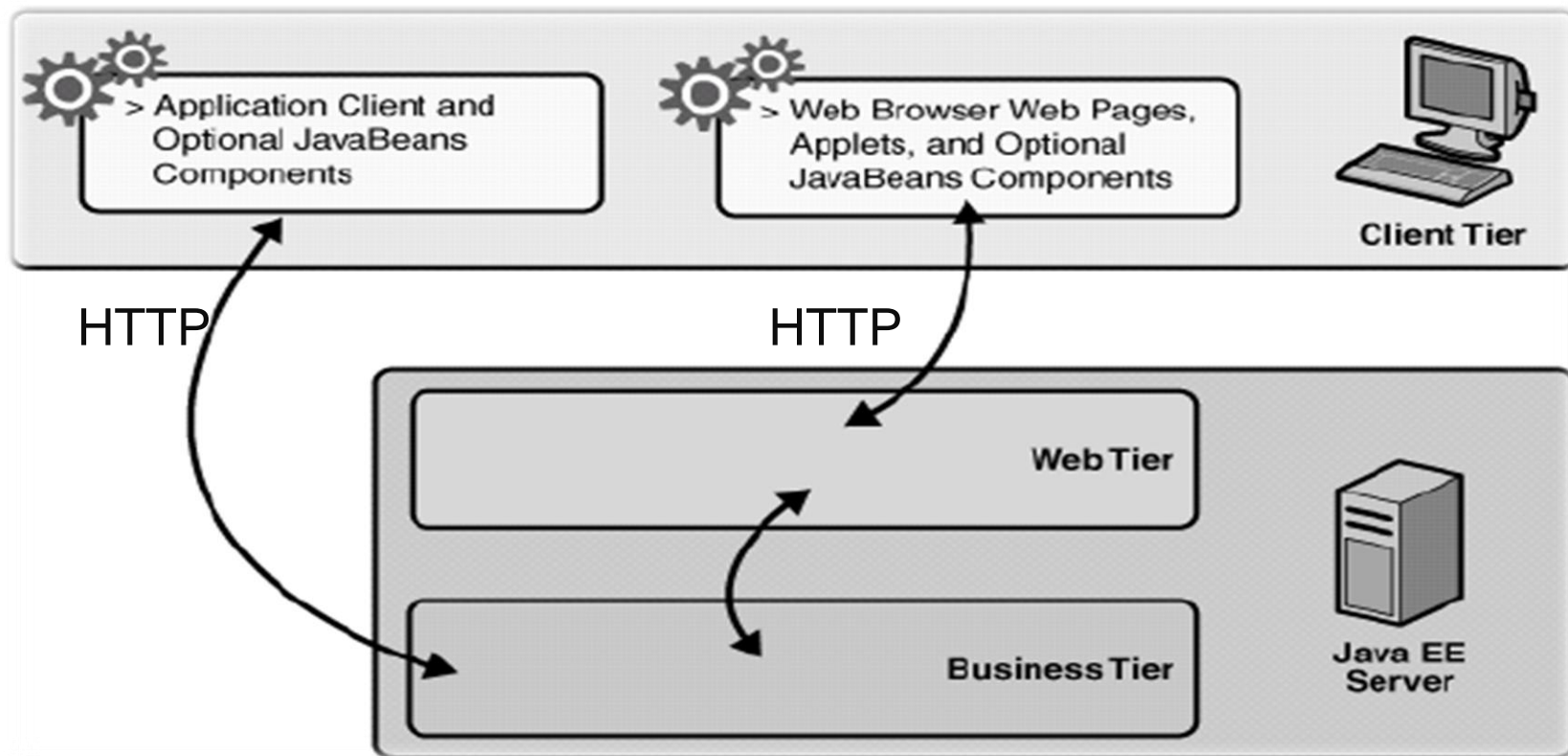
Изграждане на умни клиенти

- **Технологии за умни клиенти:**

- Flex на Adobe.
- JavaFX на SUN/Oracle
- Silverlight на Microsoft
- DataSnap/WebServices на CodeGear

Демо: <http://www.smartclient.com/#Welcome>

Комуникация със сървъра



Source: The Java™ EE 5 Tutorial

Интернет протоколен стек



ISO/OSI

TCP/IP

Приложно ниво

Представително
ниво

Сесийно ниво

Транспортно
ниво

Мрежово ниво

Канално ниво

Физическо ниво

Приложно ниво

HTTP, FTP, SMTP,
POP, DNS, RIP, SNMP

TCP

Транспортно ниво

UDP

IP - Internet слой

ARP <-> Мрежов контролер <-> RARP

Ниво за достъп до мрежата

TCP (Transmission Control Protocol) и UDP (User Datagram Protocol)

TCP

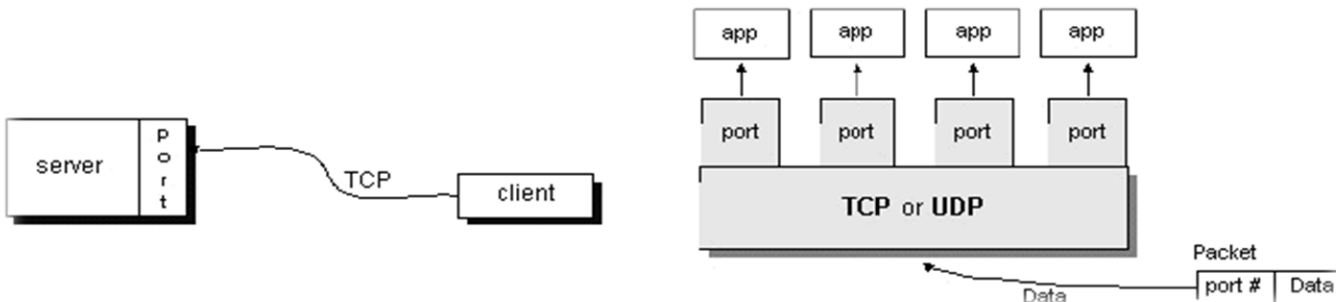
- Потвърждава получаването на IP пакети и изисква повторно предаване на повредени или изгубени пакети.
- Запазва реда на изпращане на пакетите.
- Високи разходи за управление.

UDP

- Ненадежден – не гарантира получаването на пакетите.
- Позволява получателят да детектира повредени пакети но не гарантира реда на предаване на пакетите.
- По-бърз, ниски разходи за управление.

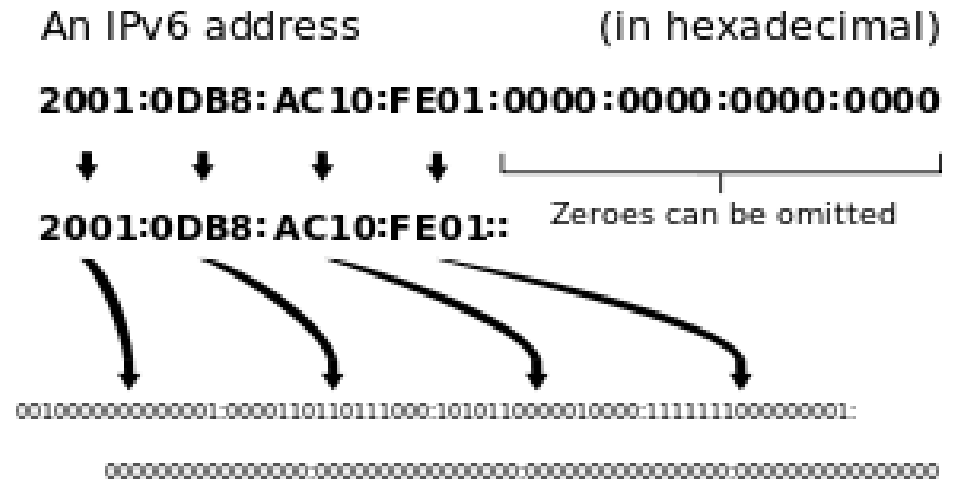
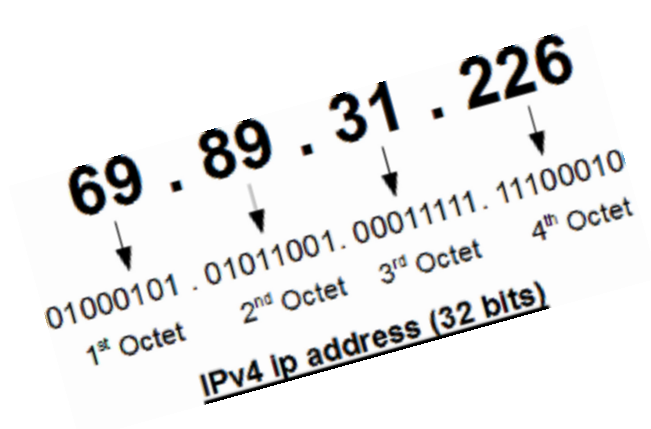
Портове

- Портовете са абстракция за програми (услуги) на сървъра.
- Номерирани са от 0 до 65 535 (16 битови числа)
- Портове от 0 до 1023 са заети от услуги като ехо, FTP, HTTP (port 80), ...



Интернет адресиране

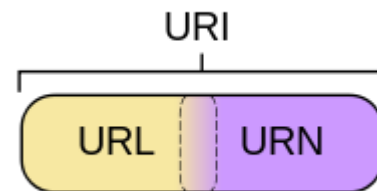
- Интернет адресът е уникален и се състои от 4 (IPv4) или 6 байта (IPv6)
- Задава адрес на устройство, включено в Интернет





Uniform resource identifier (URI)

- Единният ресурсен идентификатор (URI) идентифицира име или ресурс в Уеб.



- Съществуват два вида URI:
- Uniform Resource Name (URN) - идентифицира име, напр. `urn:www.agxml.org:schemas:all:2:0`
- Uniform Resource Locator (URL) - адресира ресурс в Интернет, напр. `https://example.com/`



Uniform Resource Locator (URL)

- URLs са съставени от пет части:
 - 1. Схема, известна като протокол
 - 2. Ауторити (authority)
 - 3. Път (path)
 - 4. Заявка (query string)
 - 5. Фрагментен идентификатор, известен още като секция или референция (ref)
- `<scheme>://<authority><path>?<query>#<fragment>`

Пример за URL

<http://www.ibiblio.org/javafaq/javabooks/myApp?isbn=123#toc>

- 1. scheme : http
- 2. authority : www.ibiblio.org
- 3. path : /javafaq/books/javabooks/myApp
- 4. query string : isbn=123
- 5. fragment identifier : toc

Authority на свой ред може да се раздели на потребит. информация (user info), хост (host) и порт (port).

Например за <http://admin@www.blackstar.com:8080/>

user info : admin; host : www.blackstar.com; 3. port : 8080

HyperText Transfer Protocol (HTTP)

- **Hypertext Transfer Protocol (HTTP)** – приложен протокол за Уеб.
- Ориентиран е към комуникация от типа заявка (от клиента) и отговор (от сървъра) - request-response protocol
- Уеб браузерът (клиент) се реферира като user agent (UA)
- HTTP **прокси сървъри** (proxy servers) улесняват комуникацията между клиенти и сървъри и се позиционират в частни мрежи.

Структура на HTTP транзакция

HTTP използва client-server модел:

- о HTTP клиентът отваря връзка (connection) и изпраща заявка (request message) към даден HTTP сървър;
- о сървърът връща отговор, евентуално с искания ресурс;
- о след връщане на отговора сървърът затваря връзката -> HTTP е stateless protocol, и не поддържа връзка между транзакциите).
- Форматът на заявката се състои от:
 - Заглавен ред,
 - Нула или повече заглавни редове (header lines),
 - Празен ред (CR/LF), и
 - Опционално – тяло на съобщението (message body) – файл или генерирани данни от сървърното приложение

HTTP формат на отговора

1. Отговорът съдържа статус код и текст.
2. Статус кодът е трицифрено число, като първата цифра е индикатор за тип на отговора, както следва:

1xx - informational message only

2xx - success of some kind

3xx - redirects the client to another URL

4xx - an error on the client's part

5xx - an error on the server's part

Често срещани статус кодове

200 OK

Успешно изпълнение; ресурсът или резултата от изпълнението на сървърния код се намира в тялото на съобщението.

301 Moved Permanently

302 Moved Temporarily

303 See Other Location: <http://example.org/other>

404 Not Found-Заявеният ресурс не съществува.

500 Server Error

URL кодиране (encoding)

- URL Encoding – процес на конвертиране на URL до валиден URL format.
- URL encoding се прилага обикновено за конвертиране на данни от html формите, понеже такива данни могат да съдържат специални символи като '/', '.', '#', '`' и др., които могат:
 - а) Да имат специално значение – напр. '#'
 - б) Да не са валидни символи за URL – напр. празна позиция
 - с) Да бъдат променени по време на трансфера
- Пример: "\$ & < > ? ; # : = , " ' ~ + %" ще се кодира като "%24+%26+%3C+%3E+%3F+%3B+%23+%3A+%3D+%2C+%22+%27+%7E+%2B+%25"

HTTP POST метод 1/2

POST заявката изпраща данни към сървъра.

POST се различава **GET** заявка по следното:

- В тялото на съобщението има блок от данни, които се изпращат от клиента
- В тялото на съобщението има допълнителни хедъри като Content-Type: и Content-Length:
- Заявеният URI не указва ресурс, който трябва да бъде върнат, а адресира сървърен компонент
- HTTP отговорът е резултат от изпълнението на този сървърен код, а не е статичен файл
- Обичайна употреба на POST заявка – за изпращане на данните на HTML форма към сървърно приложение.

HTTP POST метод 2/2

- POST заявката може да се използва за изпращане на каквито и да е данни, не само от форми
- GET заявката също може да се ползва за изпращане на данни от форма, като самите данни са URL-encoded и се добавят към заявения URI.

HEAD метод

HEAD заявката прилича на GET заявка, като обаче сървърът връща само хедърите, но без искания ресурс тоест без message body).

Използва се за проверка на характеристиките на ресурса.

Трансфер на параметри от HTML форма

car Rent - Microsoft Internet Exp...
File Edit View Favorites Tools >>
User - Log on
User: testuser
Password: *****
Log on Reset
Done Local intranet

HTML Code

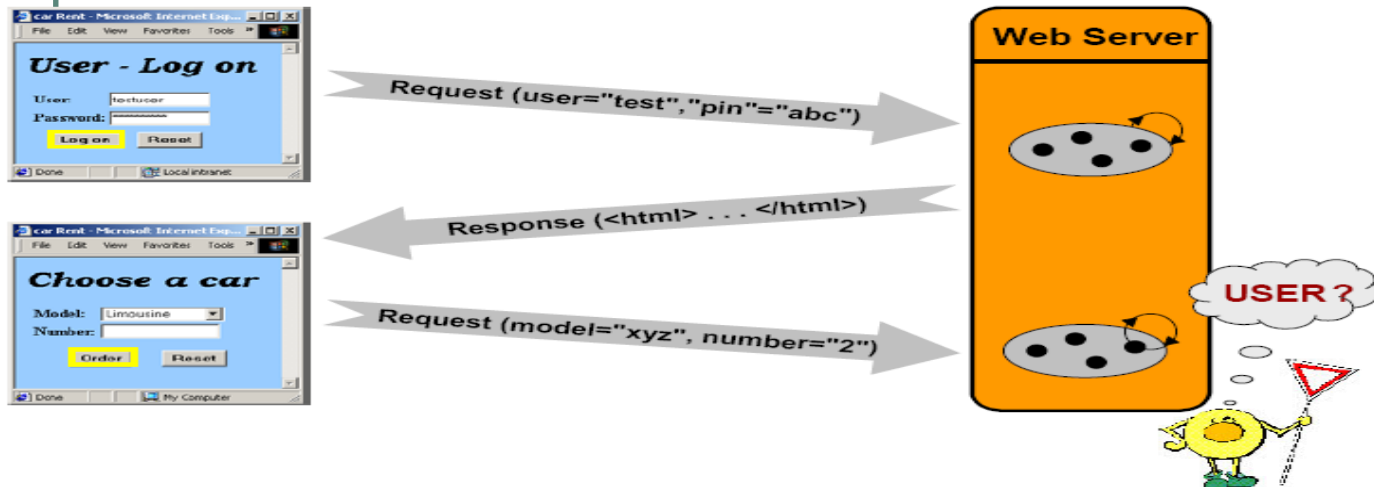
```
1 <form method="POST" action=" URI-Path ">  
2 <input type="text" name="uname">  
3 <input type="password" name="user-pin">
```



LogOnServlet	X
<pre>1 public void doPost (HttpServletRequest req, HttpServletResponse res) { 2 3 String uname; 4 String password; 5 6 uname = req.getParameter("uname"); 7 password = req.getParameter("user-pin"); 8 9 // Processing of the transferred parameters 10 11 }</pre>	

HTTP не поддържа състояние

- Всяка заявка се обработва отделно от Уеб сървъра
- Доставят се само данните от последната заявка
- **НО:** последователността от поредица заявки и отговори в рамките на едно приложение може да е логически обвързана -> **SESSION**



Как се поддържа сесия?

- **Session Tracking** - поддържане на информация за контекста на използване на приложението между извикването на няколко страници от него
- **Реализации**
 1. Чрез скрити полета във формата (Hidden Form Fields)
 2. Пренаписване на URL (URL rewriting)
 3. Бисквитки (Cookies)
 4. Сесийни обекти на сървъра (напр. HttpSession)

1. Скрити полета

- Insert hidden fields in HTML forms
- Transferred like usual form fields
- No display by the browser

LogOnServlet X

```
1 public class HiddenFieldsServlet extends HttpServlet {
2
3 public void doGet ( HttpServletRequest req, HttpServletResponse res )
4     throws IOException, ServletException {
5     ...
6     out.println("<input type='text' name='colour'> ");
7     out.println("<input type='hidden' name='selModel' value='cabrio'>");
8     ...
9 }
10 }
```

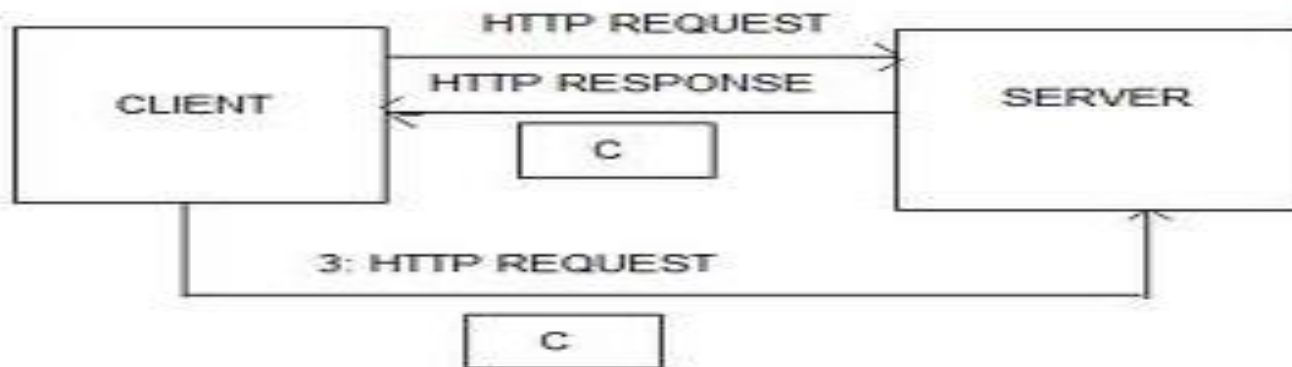


2. Пренаписване на URL

- При деактивирани в браузъра cookies, трасирането на сесията е възможно с т.нар. URL rewriting. Сървърното приложение добавя session ID към всички URL, които се съдържат в генерирания от него отговор.
- При активиране на дадена хипервръзка, session ID се прехвърля заедно с нейния URL и се прочита от сървъра.

3. Бисквитки (Cookies)

Бисквитки (Cookies) – механизъм за поддръжка на състоянието чрез задържане у клиента на кратки текстове с информация за контекста на клиентската сесия.

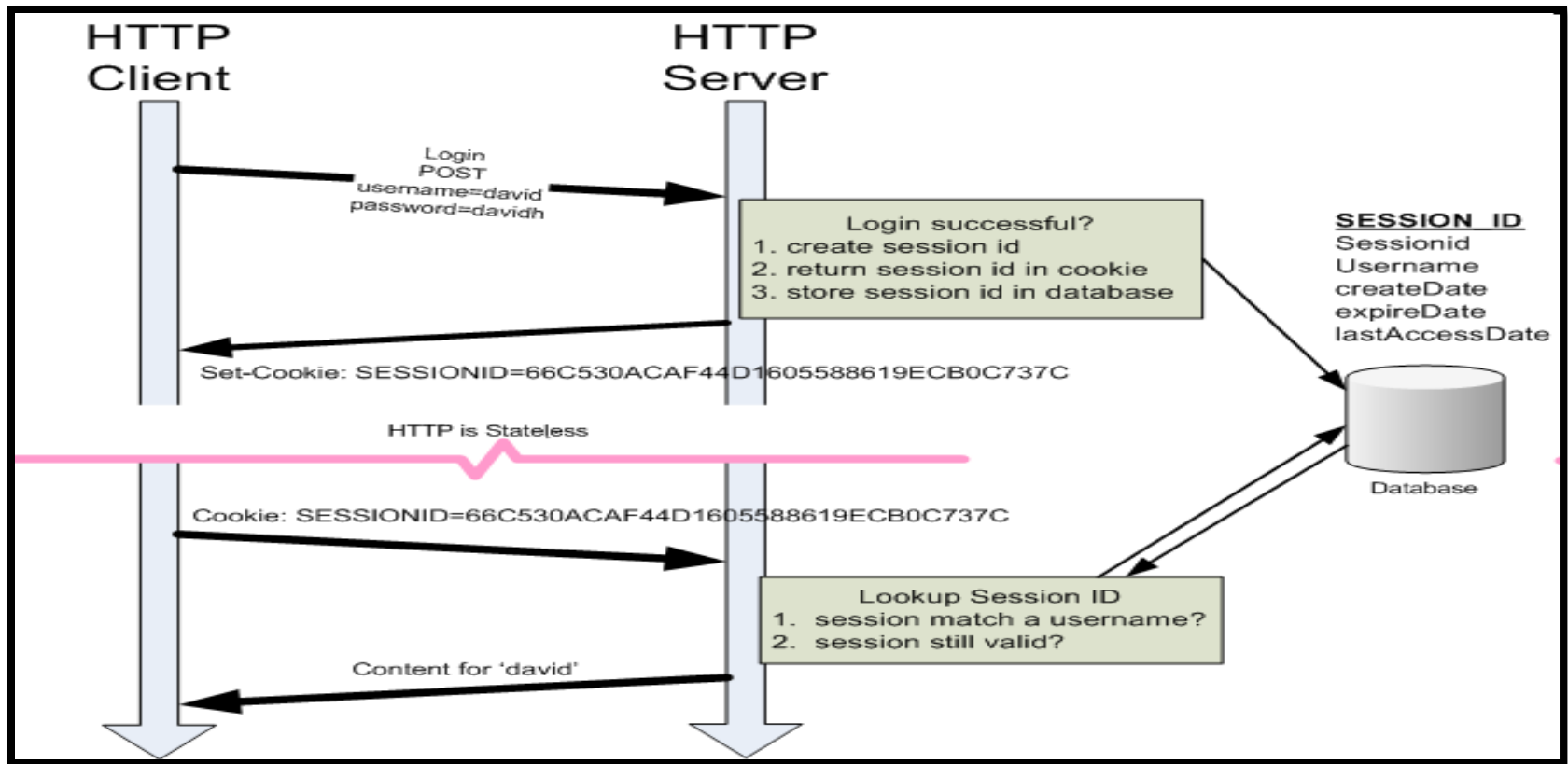


Виж <http://www.cookiecentral.com/faq/>

Същност на Cookies

- Бисквитката (cookie) е малък текст с информация, изпратен от сървъра, съхранен (евентуално!) от брауъра и по-късно изпратен към същия сървър.
- Стойността на бисквитката идентифицира клиента по уникален начин => бисквитките могат да се използват за поддръжка на сесията.
- Бисквитката има име (**name**), единична стойност (**single value**), и опционално – атрибути (**optional attributes**) като коментар, path и domain, maximum age, и version number.

Изполване на Cookies



4. Трасиране на сесия (Session Tracking)

Трасирането на сесията е механизъм за поддръжка на сесиен обект от страна на сервърния компонент за даден браузър за определен период от време, посредством размяна на биквитка с идентификатора на сесията или пренаписване на URL с този идентификатор.

- Server side storage of session data
- Identification of a session using the session ID
- Cookies or URL rewriting for ID transfer



A screenshot of a Microsoft Internet Explorer window titled "carRent - Microsoft Internet...". The browser's menu bar shows "File", "Edit", "View", and "Favorites". The page content includes a "User:" label with a text input field containing "testuser", a "Password:" label with a password input field, and two buttons labeled "Log on" and "Reset".

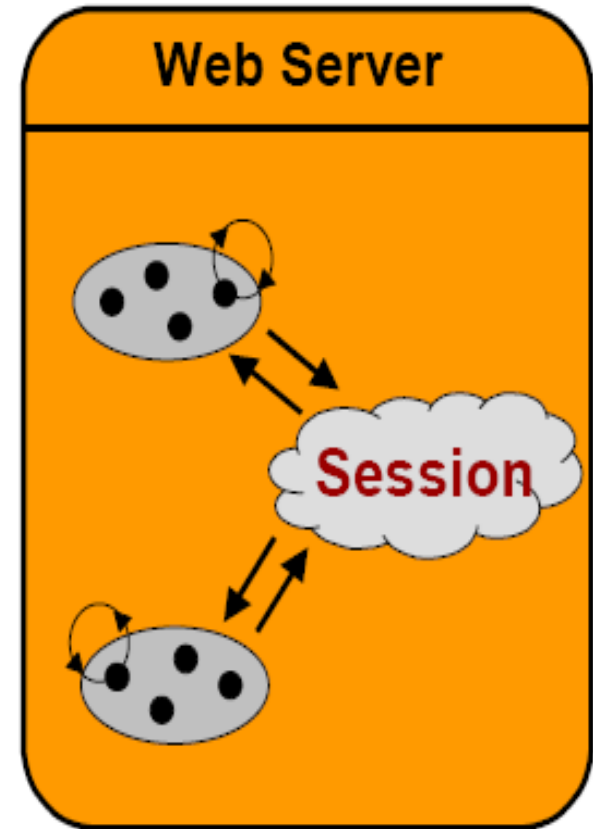
Request (Parameter_1)



A screenshot of a Microsoft Internet Explorer window titled "carRent - Microsoft Internet...". The browser's menu bar shows "File", "Edit", "View", and "Favorites". The page content includes a "Model:" label with a dropdown menu showing "Linousine", a "Number:" label with a text input field containing "25", and two buttons labeled "Order" and "Reset".

Response (SessionID)

Request (Parameter_2)



Литература

- ✓ Patrick Carey, New Perspectives on HTML, CSS, and Dynamic HTML, 5th Edition, Course Technology, Boston, USA, ISBN-13: 978-1-111-52643-6 ISBN-10: 1-111-52643-5, 2017
- ✓ Склар, Д., Принципи на уебдизайна. Четвърто издание, Дуо Дизайн, 2010, ISBN: 9789548396343.
- ✓ Crowder, Ph., D. Crowder. Creating Web Sites. Bible, Third Edition, Wiley Publishing, Inc., 2008, ISBN: 978-0-4702-2363-5.
- ✓ Frain, B., Responsive Web Design with HTML5 and CSS3, Packt Publishing Ltd., 2012, ISBN: 978-1-84969-318-9.
- ✓ Duckett, J. HTML and CSS: Design and Build Websites, John Wiley & Sons, Inc., 2011, ISBN-10: 1118008189, ISBN-13: 978-1118008188.

Литература

- ✓ Freeman, A. The Definitive Guide for HTML5. Apress, 2011, ISBN-10: 1430239603.
- ✓ McIntire, P. Visual Design for the Modern Web, New Riders, 2011, ISBN: 978-0-321-51538-4.
- ✓ Osborn, J., Adobe Dreamweaver CS6 Digital Classroom, John Wiley & Sons, Inc., 2012, ISBN-10:111812409X, ISBN-13:978-1118124093.
- ✓ Robbins, J., Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics, Fourth Edition, 2012, O'Reilly, ISBN-10: 1449319270, ISBN-13: 978-1449319274.
- ✓ The Definitive Guide for HTML5, by Adam Freeman, Apress, 2011
- ✓ JavaScript Essentials, by Sven Lennartz and Vitaly Friedman, Smashing Media GmbH, Freiburg, Germany, Dec. 2011

Литература

- ✓ Pro jQuery, by Adam Freeman, Apress, Feb. 2012
- ✓ Patrick Carey, New Perspectives on HTML5 and CSS3, 7th Edition, Comprehensive, ISBN: 978-1-305-50393-9, Cengage Learning, Boston, MA 02210 USA, 2018
- ✓ Programming the Semantic Web, by Toby Segaran; Colin Evans; Jamie Taylor, O'Reilly Media, Inc., July 14, 2009
- ✓ Mobile Design for iPhone and iPad, Sven Lennartz, Vitaly Friedman, Adobe Flash Professional CS5, Katherine Ulrich, 2010
- ✓ Professional Workflow for Web designers, Luke Reimer, 2011
- ✓ Flexible Web Design: Creating Liquid and Elastic Layouts with CSS, Zoe Mickley Gillenwater, 2009
- ✓ <http://www.wowebook.com/>
- ✓ <http://www.w3schools.com/>