



IF3110 – Web-based Application Development

Internet Application Concept

Main Concept

- › Internet Programming → Programming Internet-based Application (Internet App)
- › Application that is distributed across the place; using Internet as communication means
- › Keywords:
 - › Distributed System
 - › Communication via Internet

Characteristics

- › Concurrency
 - › Handle many tasks at the same time (e.g., send/receive data, process user requests, handle many clients)
- › Synchronize
 - › activity coordination
 - › time/timing handling
- › Exception Handling
 - › a fail in handling a user; won't cause any problems to other users

Distributed System

- › Main Classes
 - › Client Server: service requester (client) and provide services (server)
 - › Peer-to-peer: each component is in the same level in providing and requesting services
- › In a peer-to-peer system, each component plays a role as client and server

Client-Server Variant

- › Stateless Client-Server
 - › Each client/server doesn't store each other's status
 - › Each client-server interaction is independent and stateless
- › Stateful Client-Server
 - › Store info about client-server interaction
 - › Client can send message based on the existing interaction context (i.e., no need to provide such info at the message)

Other Flavour of Client-Server

- › RPC: client-server interaction seen as procedure remote invocation
- › N-tier systems: expand a server into various tiers of servers



Client-Server Trade-Off

- › Advantageous
 - › Distributing computation across several machines
 - › Client can access services remotely
 - › Client & Server can be designed and developed independently
 - › Server can process many simultaneously requests
 - › Data can be stored in either centrally in a server or distributed across clients
- › Disadvantageous
 - › communication delay
 - › need to consider sync and parallel/concurrent process in the server

Protocol Communication

- › Definition: a set of rules agreed by client-server in communication to each other
 - › Application protocol
 - › client-server can send messages to each other with following a particular format/syntax with a specific order
 - › Transport protocol
 - › a message is chunked into many packets
 - › send the packets with various network routes
 - › at the destination the packets are constructed to the original message
- › This course emphasizes at application protocol

Application Protocol using Internet

- › Web (protokol aplikasi: HTTP)
- › E-mail (IMAP, POP, SMTP)
- › Chatting
 - › open standard: IRC
 - › non standard: YM, ICQ, MSN chat, AOL, dll
- › File transfer (FTP)
- › Remote terminal (telnet)
- › Directory service (LDAP)
- › Network monitoring (SNMP)
- › Web service (SOAP)
- › Voice (SIP, XMPP, ASTERISX)
- › etc.

Internet-based vs Web-based Application

Internet-based

- › Using Internet Application Protocol or defining own protocol
- › Application at the server communicates directly to client
- › Application can be a standalone or a component for an existing application

Web-based

- › Using HTTP
- › Application communicate to Client via Web Server
- › Application commonly runs in web browser

Technology in Web-based Application Development

Web client (web browser)

Web server

URL : Uniform Resource Locator

HTTP : HyperText Transfer Protocol

HTML : HyperText Markup Language

CSS : Cascading Style Sheet

Web Programming

- CGI

- server side scripting

- client side scripting

- plug-in

Web client (web browser)

web browser

a software

runs in client/user's computer

navigate through the web

render/view the web page

Example:

Chrome (Windows)

Internet Explorer (Windows)

Mozilla Firefox (Windows & Linux)

Opera (Windows & Linux)

Safari (Mac)

lynx, berbasis teks (Linux)

Web server

web server

- a software

- runs in a server

- store web dokumen so it can be access by a user acorss the Internet

Example:

- Apache (Linux & Windows)

- MS Internet Information Server / IIS (Windows)

- Tomcat, untuk Java (Windows & Linux)

URL

(Uniform/Universal Resource Locator)

URL is locating a resource (file) in the internet

URL format is defined in RFC 1738 (<http://www.ietf.org/rfc/rfc1738.txt>)

URL has protocol type

Example

<http://www.if.itb.ac.id/>

<mailto:fulan@informatika.org>

<ftp://ftp.itb.ac.id/>

Example of URL in the web:

<http://www.itb.ac.id/campus-life/index.html>

<http://www.google.com/search?hl=en&q=URL+RFC>

<http://www.indymedia.org:8081/>

HTTP (HyperText Transfer Protocol)

HTTP is a standard web protocol communication

HTTP standard (HTTP 1.1) defined in RFC 2616

(<http://www.ietf.org/rfc/rfc2616.txt>)

Example



HTML (HyperText Markup Language)

HTML is a standar document in the web

HTML standard (HTML 4.01) accessible at
<http://www.w3.org/TR/html4/>

Example:

```
<html>
<head>
  <title>My first HTML document</title>
</head>
<body>
  <p>Hello world!<br>Welcome to my <b>first</b> HTML
page.
  </p>
</body>
</html>
```

Output:

Hello world!
Welcome to my **first** HTML page.

CSS (Cascading Style Sheet)

CSS is a mechanism to define style (font, template, colour) in a web document

CSS standard (CSS 2) is accessible at <http://www.w3.org/TR/REC-CSS2/>

Example:

```
<html>
<head>
  <title>My first HTML document</title>
</head>

<body>
  <p>Hello world!<br>Welcome to my <b>first</b> HTML page.
</p>
</body>
</html>
```

Output:

Hello world!
Welcome to my **first** HTML page.

TCP/IP (Internet) Programming

Web Programming often uses HTTP as transport protocol and HTML/XML as message format

- Text oriented
- Stateless, client server oriented
- More flexibility can be achieved by using transport or network protocol directly

Web Programming

Web Programming

CGI, executing code in server-side (perl, C)

Web server executes a program and the output is piped to HTTP response

server side scripting (PHP, ASP, JSP, Python)

Web server identifies and runs script/program and insert the outputs as a part of the web document

client side scripting (JavaScript, JScript, VBScript)

Web browser identifies and runs the script/program and insert the outputs to the web document (received from the server) or modifies the web document or does what the script orders (e.g., XHR, Open Socket)

plug-in, eksekusi program di sisi client (applet, ActiveX, Flash)

Web browser executes the program with the help of plugin and view the outputs at a defined place in the web document

Evolusi Aplikasi Web

- Web site – a collection of a document accessible through internet
- Web app – a client/server app through a web as the client, and uses Internet to communicate

Typical web app processing

Source:
Shklar, L.
Web Application Architecture:
Principles, Protocols and
Practices
Wiley Publishing, Inc., 2003

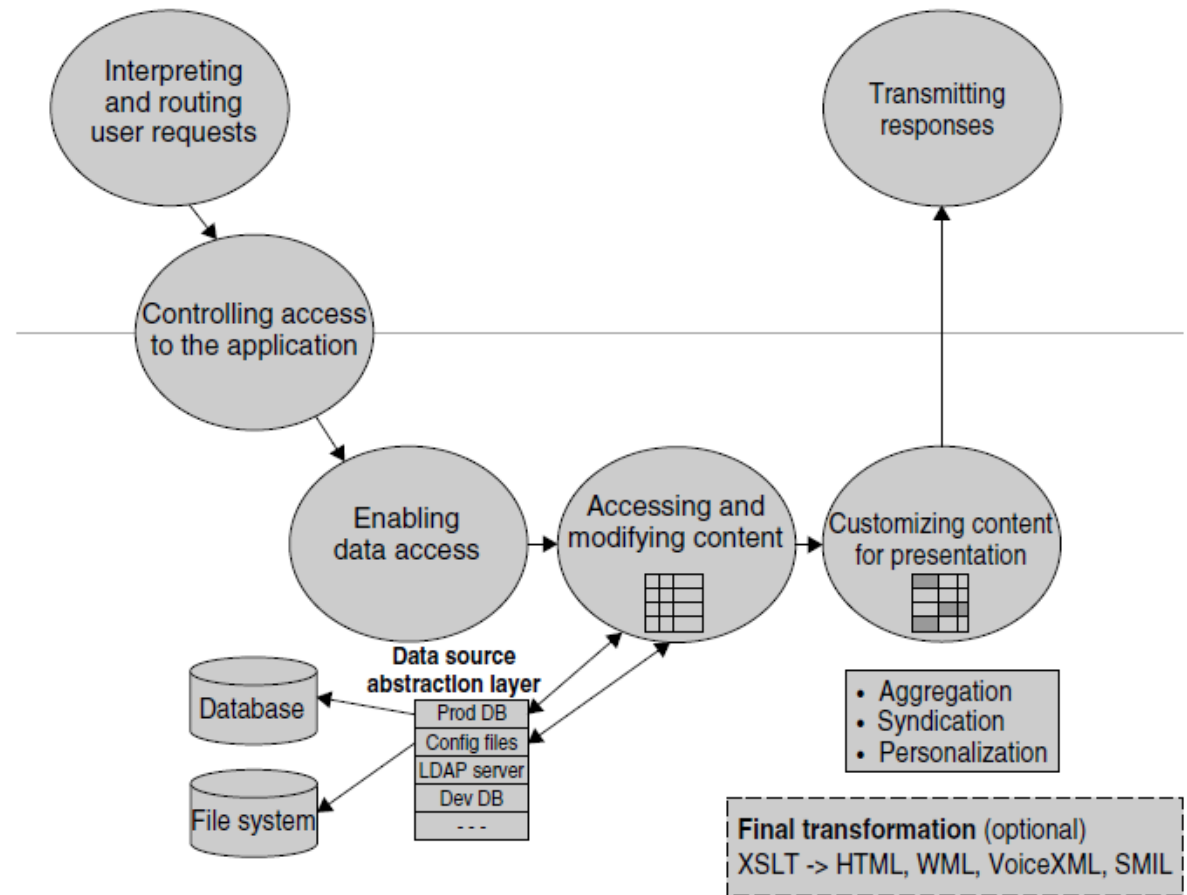


Figure 8.3 Processing flow in a typical Web application (Above the grey line—Web server; below the grey line—Web application)

Web Stack

- › A collection of software required to develop and run web applications
- › Contains all software layers
 - › Operating System
 - › Web Server
 - › Programming Language (frontend & backend)
 - › Web Framework (frontend & backend)
 - › Database Server
- › Examples:
 - › LAMP, MEAN, ...

Controlling user access

HTTP provides an authentication mechanism

- web server ask authentication info; when the request doesn't have such info

Web Application Architecture

- presentation layer: CSS
- markup / HTML/XHTML
 - templating system
 - code segregation
- presentation/page logic
- business/application logic
- persistent store

Web Application Architecture

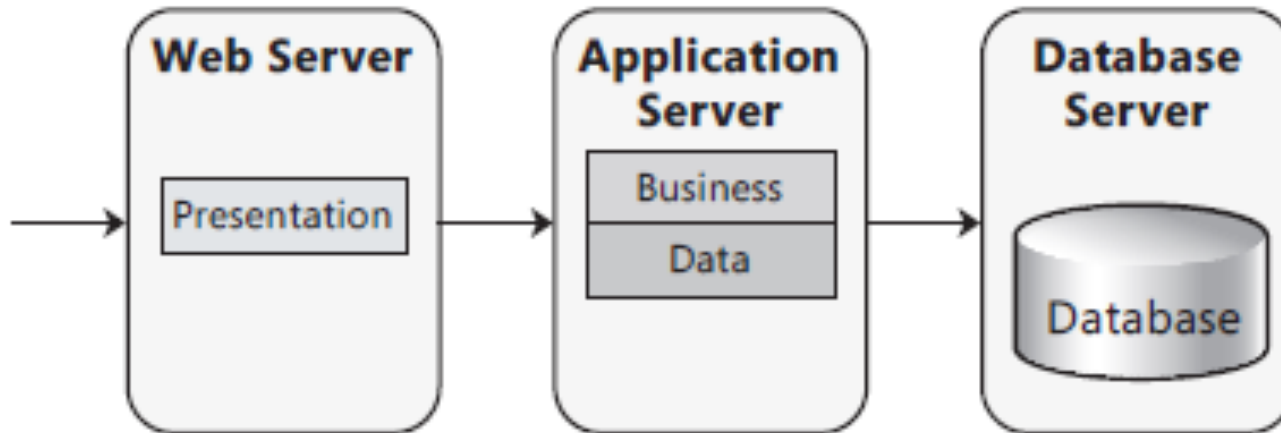


Figure 3
Distributed deployment of a Web application

Application Architecture

Source“:

Microsoft(r) Application
Architecture Guide, 2nd
Edition (Patterns &
Practices)
Microsoft Press, 2009

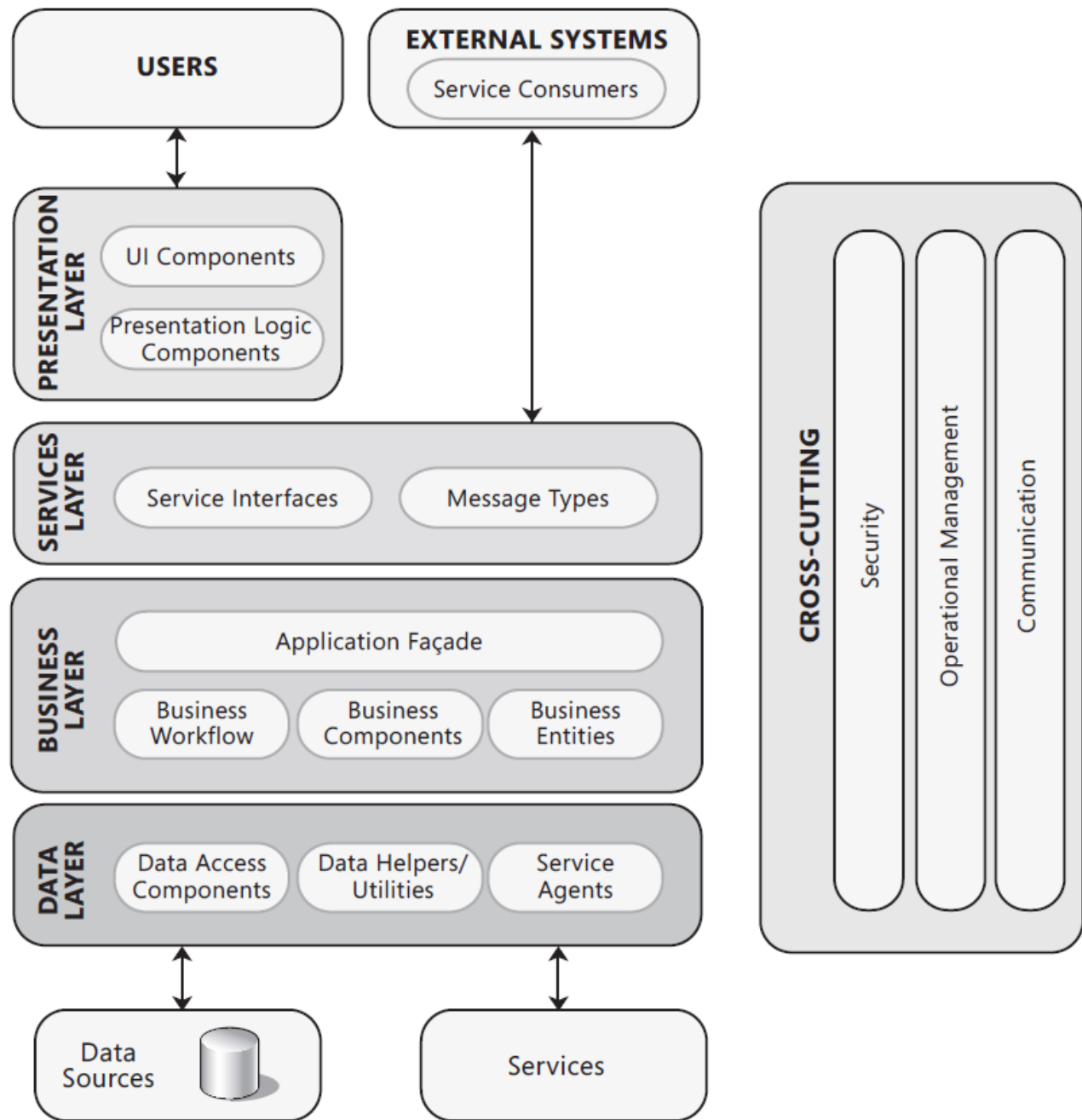


Figure 1
Common application architecture

Common Aspect in Web-App

- App request processing
- Authentication
- Authorization
- Caching
- Exception Management
- Logging & Instrumentation
- Navigation
- Page layout
- Page rendering
- Session management
- Validation
- Internationalization
- User Experience

Exercise

- › <https://developer.mozilla.org/en-US/docs/Learn>
 - › HTML
 - › CSS
- › Objective
 - › Understand the roles of those technologies
 - › Understand how those tech are processed in a webapp

Exercise Deliverables

- › Buatlah aplikasi web statis dengan HTML, CSS, dan JS sederhana yang berisi:
 1. Sebuah halaman yang berisi daftar produk barang yang menampilkan nama, harga, dan thumbnail produk
 2. Jika sebuah produk pada halaman daftar produk diklik, muncul halaman kedua yang berisi nama, harga, foto, dan deskripsi produk
- › Daftar informasi produk ada pada sebuah file json terlampir. Untuk memudahkan, Anda dapat menyalin isi file json pada sebuah JS file (hard-coded)
- › Pengumpulan:
 - › Buat sebuah project baru di gitlab.informatika.org dengan nama WBD-2021-NIM-Nama. Add user `teacher_if3110_2021` sebagai developer
 - › Buat direktori `W01-StaticWeb`, dan commit semua file yang Anda buat di direktori tersebut.
 - › Masukkan juga screenshot dari halaman-halaman yang Anda buat pada direktori tersebut.
 - › Deadline: 30 Agustus 2021, pukul 15.00