Web Engineering mini-project sw704e12

Kristian Kolding Foged-Ladefoged
Dan Stenholt Møller
Jens Mohr Mortensen
Mikael Midtgaard
Rasmus Hoppe Nesgaard Aaen

Exercise 1: Choice of architecture

Requirements

There are a number of requirements we have for this project. We base our choice of architecture on how well they meet these requirements.

- Must support a Multi-layered architecture
- Must be easily extendable
- Must have support for testing and test driven development

As we see it there are three options to choose from when choosing an architecture

.NET, PHP,Java EE

.NET

The Microsoft MVC architecture supported in Microsoft .NET framework

Advantage

- We are all familiar with .NET, however at varying degrees - Everyone has developed a web application at 3rd semester
- Supported by a good IDE - Visual Studio
- OOP

Disadvantage

- Microsoft Server is the Nativedb

PHP

Web development with HTML, JavaScript, and PHP

Advantage

- It is fast getting started developing a web application in PHP.

Disadvantage

- Not everyone in the group is familiar with PHP
- There are many security concerns that have to be addressed when developing in PHP, partially due to PHP being a weakly typed language.

Java EE

Java Platform (Java EE) on the application server.
Advantage
- Everyone is familiar with Java
- OOP

Disadvantages
- There is a large overhead to setting up a layered structure using Java.
- No one has previously developing a Java web application.

MVC

Advantage
- More unit testable
- No business logic in the UI
- Easier to maintain, extend, and reuse

Disadvantage
- More initial work (more code)

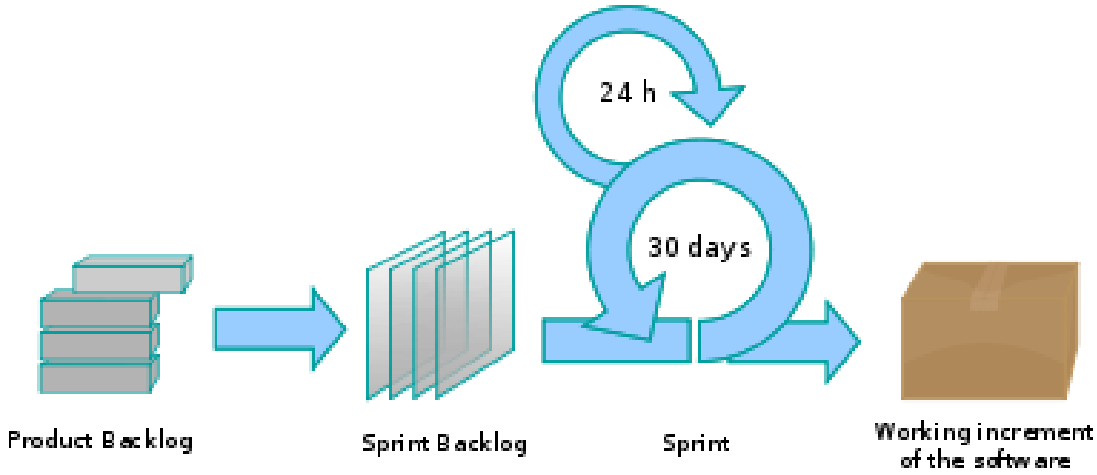
a) Write your own "hello world" application in three tier architecture of technology of your choice and discuss the experience
- Development environment: Java EE/JBoss
JBoss It is very troublesome to get started with, but when the project is up and working in Eclipse, it is just as any other project.
Working with Java is an advantage, because we all have been working with it in previous projects and are comfortable using it.
Most of us haven't had much experience with web development and therefore, xhtml is just as challenging as developing in any other web development environment.

We have had some hard times getting the environment working, mainly because the JBoss plugin for Eclipse could not detect our Java SDK's.

b) Write down your own process for your web application project and discuss why you have decided to do so

We have decided to use Scrum as our process because we have all been using this process in previous semesters.

Process:



We believe this is a good process because we can adapt to changes at any time in the project. Furthermore did we decide to try Test Driven Development, to get better quality of the product.

Exercise 2: Data I: XML etc.

Load an at least one XML file from http://www.cs.washington.edu/research/xmldatasets/ into your hello world application. Present subset of the file in a structured presentation (could be a table or structured lists). Reflect on what you have learned, on difficulties, on shortcomings and advantages of XQuery and XSLT.

XQuery

XML query language, that can query large collections of XML
You can query XML documents with XQuery to use the data in a HTML page.
Say you have a collection of paths to ebay listings categorized in listing name, if you would like to show only the listings including items you could make an XQuery.

```
<html><head/><body>
{
  for $ebay in doc("ebay.xml")
  let $listings := distinct-values($item//LISTING)
  return
    <div>
      <h1>{ string($listings/NAME) }</h1>
      <ul>
      {
        for $listing in $listings
        return <li>{ $listing }</li>
      }
      </ul>
    </div>
}
</body></html>
```

XSLT

Stylesheet and transformation language
Can be used to create XML documents into e.g. HTML or another XML document
Say you would like to show listings from a collection of eBay listings in an XML document.
You could transform the XML document into a HTML document with XSLT:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
 version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 xmlns="http://www.w3.org/1999/xhtml">

 <xsl:output method="xml" indent="yes" encoding="UTF-8"/>

 <xsl:template match="/listings">
```

```
  <html>
    <head> <title>Testing XML Example</title> </head>
    <body>
      <h1>Persons</h1>
      <ul>
        <xsl:apply-templates select="listing">
        </xsl:apply-templates>
      </ul>
    </body>
  </html>
 </xsl:template>

 <xsl:template match="listing">
  <li>
    <xsl:value-of select="information"/><xsl:text>, </xsl:text><xsl:value-of select="name"/>
  </li>
 </xsl:template>

</xsl:stylesheet>
```

Since we are working in Java EE, it makes more sense to parse the XML document directly in Java, such that we can store the XML data in an object in our model.

To load a xml file into our hello world application, we used java's DOM XML parser(Document Object Model). In Dom the entire xml files is loaded into an document object, which then can be traversed.
A snap of the code can be seen below:

```
public Document Read() {
            try {
                  URL url = new URL(urlString); // Url to the xml file
to load
                  URLConnection conn = url.openConnection();

                  DocumentBuilderFactory dbFactory =
DocumentBuilderFactory.newInstance();
                  DocumentBuilder dBuilder =
dbFactory.newDocumentBuilder();
                  doc = dBuilder.parse(conn.getInputStream());
            } catch (Exception e) {
                  e.printStackTrace();
            }

            return doc; // Document object, which can be traversed
      }
```

Exercise 3 - Data II - Semantic Web

There are two subassignments and both of them are to be part of the exam. Learning objective is to understand, be able to work with and reflect on semantic web and specifically on:
a/ RDF/RDFS/
b/ SPARQL

a/ Implement GetRDF function (through a button, http request or any other) on your data you have gathred from XML exercise. Reflect on RDF, RDFS and the technology you have used.
b/ Play with dbpedia sparql web interface and find related data to your XML exercise. Present part of the data and discuss on them advantages, disadvantages of RDF/RDFS and SPARQL also in connection to XML

RDF = Resource Description Framework
RDF can be used to describe relations or metadata on the web, in RDF you can create an object as a resource and then reference it.
SPARQL is then used to query in RDF schemas.

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:ebayurl="http://www.ebayexample.fake#">
<rdf:Description rdf:about="http://www.ebayexample.fake/index">
 <ebayurl:isaWebsiteOf rdf:resource="ebay"/>
</rdf:Description>
<rdf:Description rdf:about="ebay">
 <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
</rdf:RDF>
```

RDF validator: http://www.w3.org/RDF/Validator/
RDF can be used to model the structure of websites.

Exercise 4 - Architecture

Take one pattern from each layer (model, view and controller) and implement it in your application. Discuss how you have implemented that. Reflect on its advantages and disadvantages. Expand on the discussion of advantages and disadvantages about some selected other patterns which you have not implemented in your assignment.

In the eBay case we decided to use the Transaction Script, Transform View, and Application Controller patterns. Since the business logic of the eBay case is small, store bid, and load listings, the transaction script is in favor. The transform view has been selected because no transformation of the domain has to be made. To navigate between the listings and bidding on them, we use the application controller to assist the user.

Transaction Script:
Small scripts to handle transactions to the database, could be a script that creates a profile in the database. A transaction script can contain several other transaction scripts to make it clear where they belong.

Comments
Transaction Scripts are best used in systems with small amounts of business logic, since it is difficult to keep transactions scripts in a well-designed state, when used in a system with complicated business logic.

In the eBay-case this is favored since there is not much need for other business logic than input/ output for the listings and updates on buyers.
On the other hand is this not a good choice for our semester project since there is a need for the business logic to handle many relations between the tables. Which would mean that there will be many nested scripts, with a lot of scripts handling small interactions.

Domain Model:
You build domains that tie all subjects of the domain together, such that you can handle the data of a single domain in different models.

Comments
The domain model is favored in systems with complicated and ever changing business rules involving validation, calculations, and derivations. Since you would want to have an object that handle the models.

In the eBay case, this could be implemented, but as we only have to handle listings in association with users the domains will be very limited. Therefore making transcription scripts more attractive than the domain model.

Table Module:

In table module, you have a layer that handles all interactions with the database. Such that when you need a profile, you get it from the module and again when you want to store or alter the profile you will have to use the module. Instead of changing exactly what is needed you will have to change the entire profile.

Comments
Good to use with record sets (virtual representation of the database tables), such that you pull a part of the database into a virtual set and afterwards update the record set in the database.

In the ebay-case this is a great way to handle the system if we assume our only data source is the XML files. Since this would mean that we can handle bits of the database by loading and keeping it in the memory.
The semester project on the other hand will not benefit from table modules, because each profile have much data and no method use the methods the same way, meaning that much of the data on the profiles will be loaded into the memory without being used.

MVC
Model - Information about the domain, with non-visual data and behavior.
View - UI only about displaying the data.
Controller - Manipulates data from the model and handles data transfer to the view.

MVC structure is always a good choice, except in small systems where the model has no real behavior and transaction scripts or table models would be more valuable.

Page Controller
One controller for each logical page in the web site, such that each page controller handles a specific page or action on the web site.

Comments
Page controllers works great in sites where most of the controller logic is simple. Such as in sites where pages are easily distinguishable in terms of functionality.

In the ebay-case we can use the page controller to modify the information from the model, before presenting it to the view, because all listings are handled in the same way.
In the semester project we can also use page controllers to handle our views, because there is a clear difference between the different pages. The page controllers will make it easier to maintain afterwords.

Front Controller
One controller handles all requests to the web site. The front controller gathers information from the URL to determine what action the user is performing.

Comments

This could be efficient in security sensitive systems, where you might want some validation on the user before granting access to data. Also in systems where the difference between pages are the presentation of data instead of the data itself, the front controller can present the data differently based on previous selections.

Template View
Markers in an HTML page such that you can insert information into the HTML. Like using PHP inside an HTML page.

Comments
This is great to utilize the new features of HTML in combination with the many data manipulation tools available. Also the template view can reduce the duplication of code, because you can change presentation of data according to what data is available.

Transform View
Transforms a domain into a view.
The transformer view requests data from the domain and data source layer. The domain and data source layer returns the data that the is requested but without the format to make a proper web page.

Comments
Great in systems like the ebay case where each domain requires a view to present the data.

Application Controller
Wizard Style, where the controller can gather information about previous interactions.
The application controller have two responsibilities, deciding which domain logic and view to display the response.

Exercise 5 - Design

Design and implement a navigation and presentation for some of the data you have processed in the XML assignment. Discuss why you have chosen particular design technique and particular design and reflect on its advantages and disadvantages. Discuss and reflect on relation between design and implementation.

We chose for our eBay case

- Tag Cloud of description texts
- Navigation bars and menus
    - Min and Max price
    - Search by text
    - Search in categories

The tag cloud can be helpful to both get a quick overview of the listings and find if there is any popular words at the moment.
The navigation bars and menus can help the user to narrow down their searches, by searching in fixed price areas, on product specific texts, or in categories.

Navigation types:
Step Navigation: Next or previous item (http://imgur.com/G0SpJ)
Paging Navigation: Googles navigation bar (https://www.google.dk/#hl=da&q=navigation&oq=navigation)
Breadcrumb trail or navigation: See where you have been (e.g. AAU -> CS -> Software)
Tree Navigation: Self explained
Site Map: Get an overview of the website (http://www.b.dk/sitemap)
Directories: Folder like structure
Tag Cloud: Collection of tags, tag may vary in size according to number of hits (http://www.flickr.com/photos/tags/)
A-Z index: Sort items according to letter (http://index.wsu.edu/)
Navigation bars and menus: Search bar
Vertical Menu: Vertical menu in the left hand side (http://www.elgiganten.dk/)
Dynamic Menu: Expanding menu (http://www.amazon.co.uk/)
Dropdown Menu: Menu that drops down

Properties to think of
To pursue to follow
- What comes after an item is to be used as the next item, or is a sub-element
Visual clarity
- It should be easy to grasp
Appropriateness for type of site
- The navigation should match the type of the site
Aligning with user needs

- Dont give unneccesary options

Ease of learning

- The navigation should not be overcomplicated

Feedback to user

- Let the user know when you have registered a command

Efficiency

- Dont let the user wait, maximum 30 seconds before responding

Clear Labels'

- Make it easy for the user to see the meaning of the labels and descriptions

Exercise 6 - Requirements

Think in wider context of your web application build gradually in previous assignments. Discuss what goals it achieves or can participate in, discuss which value exchanges it can provide, which workflows it can fit in, and in which information exchange it can participate in.

Achieved Goals:

- Import eBay data through XML to the web site
- RDF to search relations between objects in the website
- Navigate between eBay listings
    - According to attributes

Value Exchange

- Referencing customers to eBay
- Bid on items listed

Workflow to fit in

- When looking for hardware auctions on eBay
- Paying for bids on eBay listings

Information Exchange

- Find eBay listings according to current highest bid
- Find eBay listings according to requirements
- Find items descriptions
- Credit card information when bidding
- Personal information when bidding

Exercise 7 - Quality I

Suggest and discuss how and why you would test the reliability and performance your application you have implemented so far in assignments. Discuss what could be bottlenecks in your application and when they can appear and why? Make a data flow graph for your application and try to analyze it from data flow testing perspective. What is possible to see and what you cannot see in this test? What are advantages of this test and disadvantages.

Performance and reliability
We would test the throughput and response time, since the database is pretty small and the biggest concern is if the web site is able to keep up with user requests. Such that the users can expect that the website will process the requests when searching for eBay listings.

Bottlenecks
Response time when requesting listings from the web site, since all requests go through the server and have to be read from the database and parsed by the server. When many users request listings in a short period of time the database will have to process each request one by one. Therefore making it a bottleneck for the users.

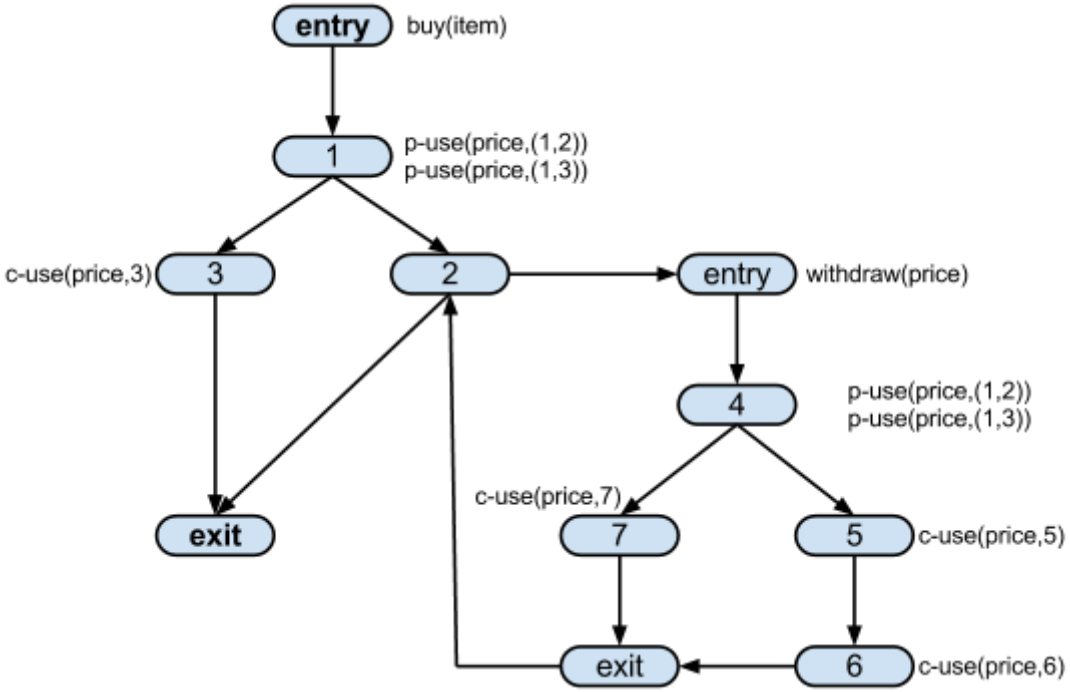
See exercise 10: scalability

Data flow graph
The ebay case does not have any code implementation, we have therefore decided to create a data flow graph based on the following pseudo code:

```
//Simple ebay buy function
//Input: item: The item you want to buy
  buy(item){
1.      if item is in stock
2.              withdraw(item price)
        else
3.              print: The item is not in stock.

//Simple ebay function to withdraw money from your account.
//Input: price: The price of the item you wish to purchase.
  function withdraw(price)
4.      if price >= money on your account
5.              account money = (account money - price)
6.              print: Item bought
        else
7.              print: you cannot afford the item
```

We can now draw a Interprocedural Control Flow Graph(ICFG) based on the two pseudo functions.



Function Cluster Level

The Function Cluster level testing is required if the c-use and p-use span over multiple functions.

Based on the ICFG we can see all the execution paths that can be testet, so called test paths.

Exercise 8 - Quality II

Based on your design, and requirements discussed in previous assignments, discuss how the design models can be utilized in testing of your application, how usage analysis can be taken into account, and how you can perform a statistical testing. Reflect on advantages and disadvantages. Answer how it would be possible to implement usage analysis extensions with log analysis library such as log4x where x is j, r, net, or php

Since we have chosen to implement a Tag Cloud we can use the most referenced tags, to implement tests on pages with popular tags. Also we can implement a test that analyses if the most used tags is also the tags the users click the most.

It is interesting to see which kind of navigation the users actually use, when navigating the web site. Experiment with the tag cloud and navigation bars by utilizing them in different ways, to get more requests through them.

Test which navigation is favored by the users, such that the less used navigation can be made more interesting for the user.

Test how often pages are requested by users, to know what kind of pages are requested most often by the users and also which kind of pages are never requested by users. Such that the web server can be utilized to favor the often requested pages. It would be interesting to know if there is any difference between how to users use the popular pages and other pages on the web site. To know if the less popular pages can be utilized to work more like the popular or the content of the pages have to be changed.

Usage Analysis
We can mine for related pages from the requests
A typical query would be:
Give me all URL's accessed in one session by users
Rank them according to number of occurrence patterns in the log
Good for the web sites with less complex and static pages.
If the patterns correlate with the links then we are fine
However, for complex applications, more fine grained log structure is necessary to assess the usage

Goals
You want to typically find out whether your designed links are followed or not whether your pages are reachable which of the pages are not used and why which data items are used are there any anomalies, i.e. invoking functions after strange navigation sequence and so on.

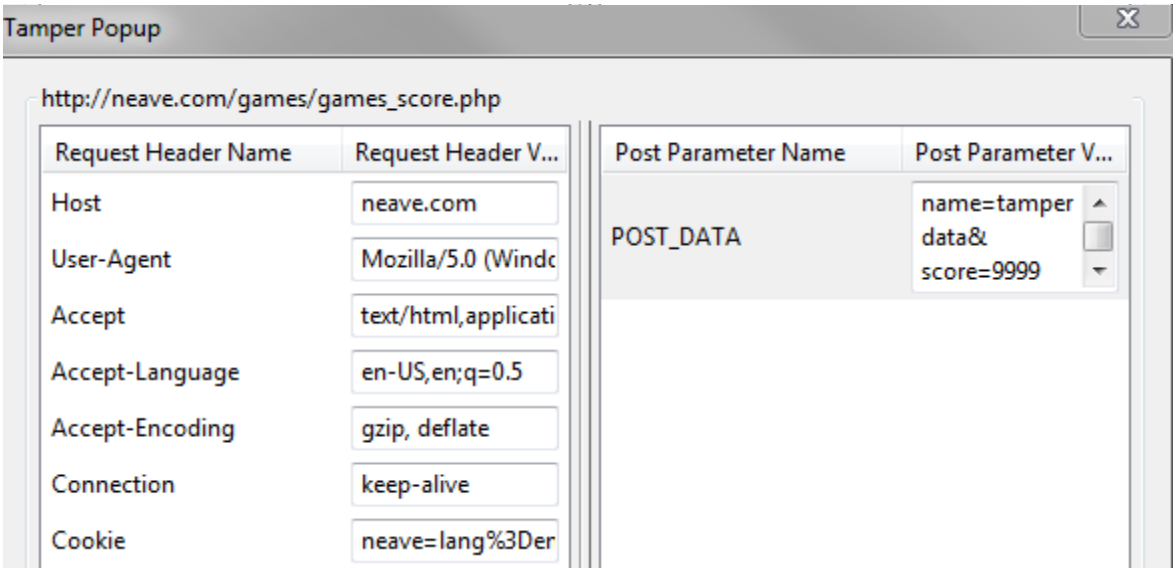
Exercise 9 - Web Application Security

Take the design of your application in the context given not only by your current implementation but also by other exercises (especially navigation design and requirements). Discuss which attacks are possible on your application and which techniques would you select for prevention and why.

Attacks

Input Validation Attacks
- Unusual input strings, can be used for SQL injection.

Form tampering
- Changing a form before it is sent to the server.



Cross-site scripting (XSS)
- When an attacker place a string or query on a website such that people using the website will load the string or query and send private information back the the attacker.

SQL Injection
- Data provided by the user is treated as an SQL statement

Command Injection Attack
- Add additional commands to a script or query, to do damage or gain information for the attacker

Defense:

Input type checking – Check e.g. for numbers
Encoding of inputs – Disallow special characters or special encoding schemes
Positive pattern matching – Recognize good input opposed to bad input
Identification of all input sources
String length – Restrict to the length of the field in the DB
String character distribution – Strings are often human readable, almost always contain printable characters

Other Attacks

Buffer overflows
- Fill the buffer, overwrite the next element

Format String Attacks
- Change a string variable to print or store information in the stack. In C “%d %s” etc can be used to get memory.

Canonicalisation Attacks
- Alternate references to the same path, used to get objects not supposed to be available e.g. C:/Program Files/ = C:/Users/Rasmus/../../Program Files/

Privilege escalations
- Gain extra access by tricking the systems to think an application have more access

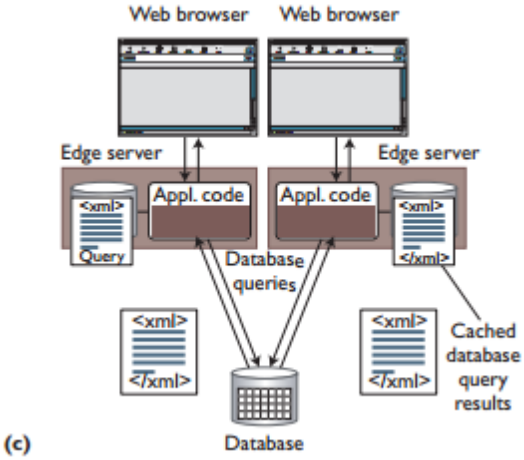
Remote malicious file inclusion
- Include a file on a website

Exercise 10 - Scalability

Take your assignment application or your student project. Discuss what scalability strategy would be the best one for your application and why. Take at least one of the presented techniques and realize it. Discuss its advantages and disadvantages and its impact.

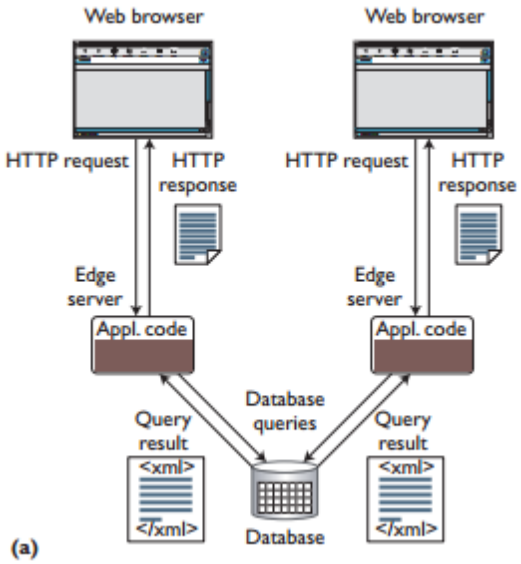
Content Blind Caching
In content blind caching we store queries and their results at the edge server. This means that when a user issues a query with an exact match of a query in the cache, we will have the result right away. We take advantage of the fact that many of the queries will be similar since the website is designed with navigation bars and menus that issues the same queries when they are executed.

With content blind caching we will not have to query any duplicate queries at the database because the queries already have the result stored with them.



Other Techniques

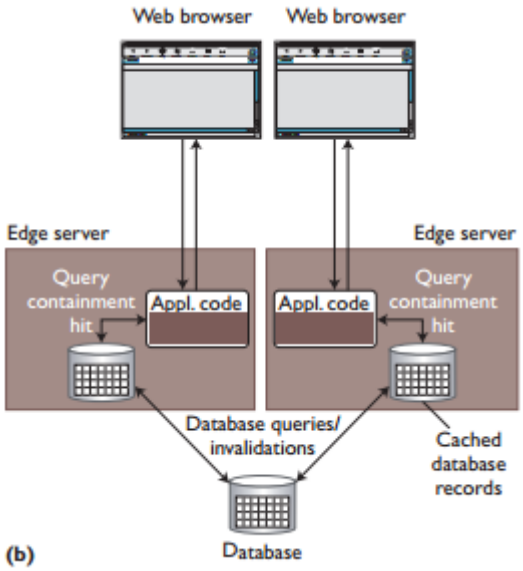
Edge Computing
The application code is replicated at several edge servers and the data is centralized. Such that the centralized server only handles the user specific content.
The centralized server will at some point become a bottleneck, because it have to handle all the dynamic content of the pages.



Edge Computing is a good solution if most content is static and therefore can be handled by the edge servers.

Content Aware Caching

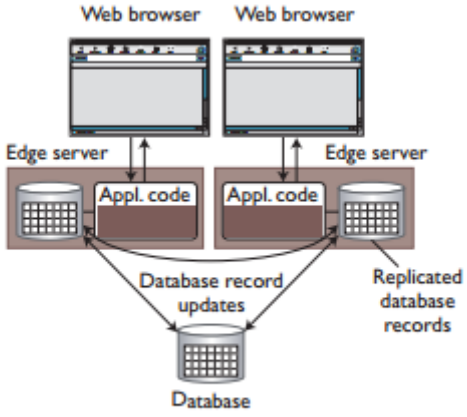
The database at the edge server only contains query results such that the database at the edge server is a partial copy of the central database. Before issuing a query to the central database the edge server check if it can evaluate the query itself by doing a *query containment check*.



The content aware caching can be used to evaluate queries executed when the user uses the search bar, that search by min/max price, text, and categories.

Data Replication

This is an extension of the Edge Computing technique, where we try to solve the problem that all user specific content requests have to be queried at a centralized server. By replicating the database records at the edge servers we can let the edge servers handle user specific requests as well. The problem with this technique, is that whenever there is an update to the database, all edge servers have to be update as well. We can reduce the update traffic by updating in a lazy fashion.



Depending on how often listings will be updated with bids, this can reduce or increase traffic. If listings are updated often with bids, then the data replication is a poor choice, because it will

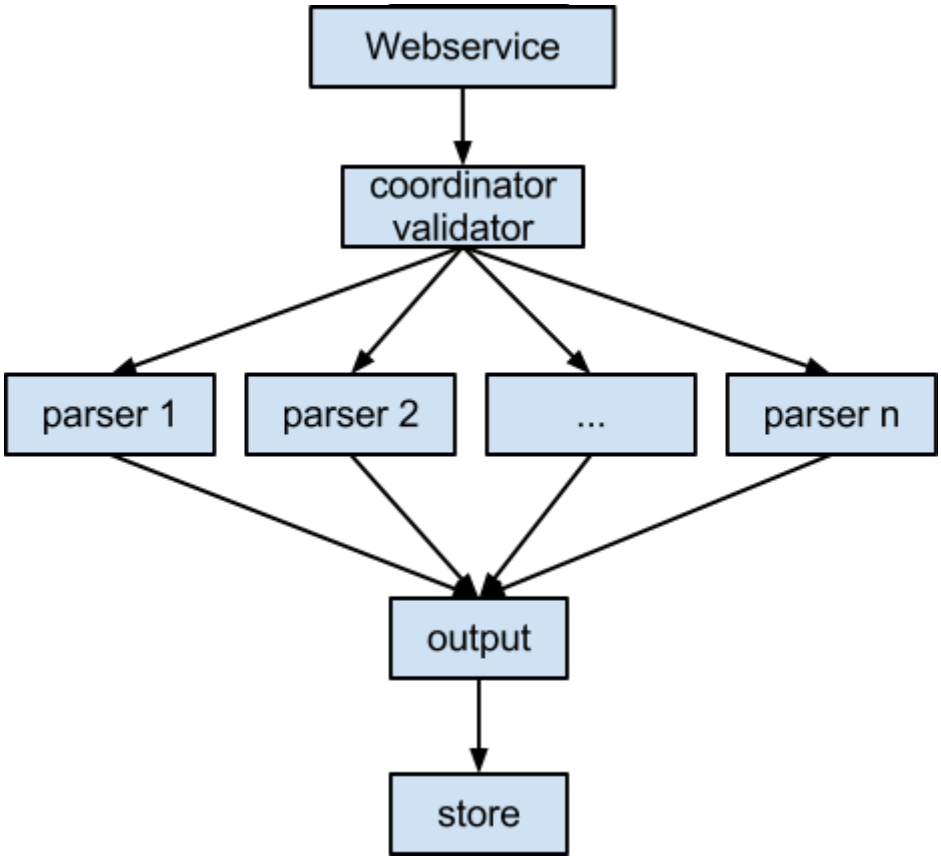
introduce much traffic. On the other hand if listings are updated infrequently the data replication can reduce much traffic at the central database.

Exercise 11: Service Based Integration

Implement web services interface for your integration with XML file load. Design an infrastructure which would allow flexible addition of more not only XML resources. Discuss advantages and disadvantages of the solutions.

Solution 1

Single input that analyses the file and distribute it to the parser able to parse it to a unified output, that is stored and shown to the user.



Advantages:

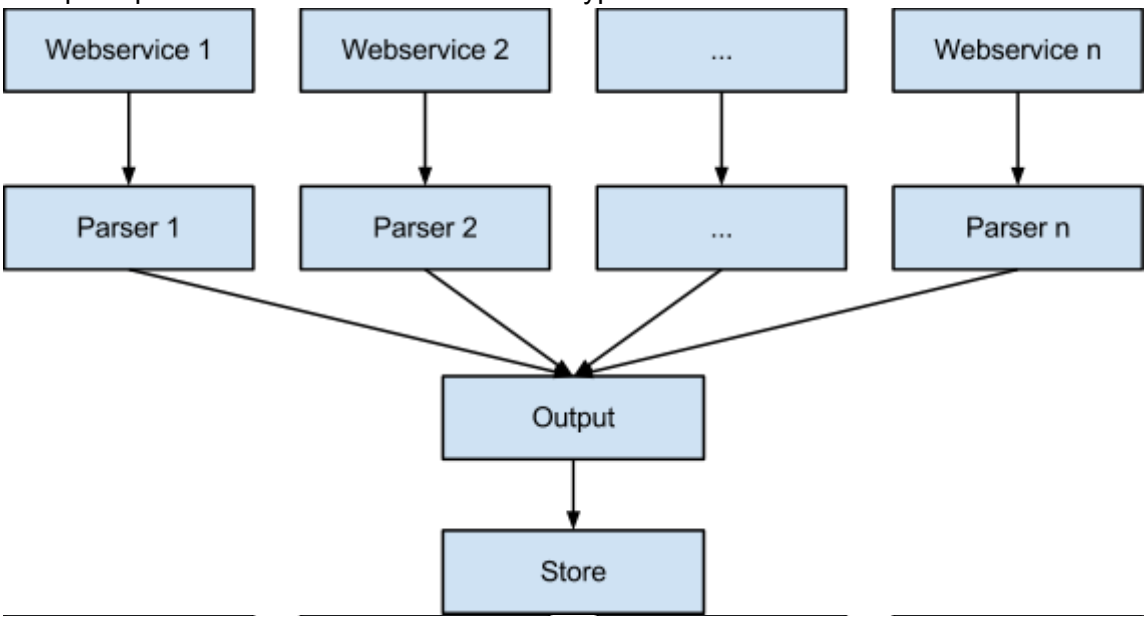
- Easy for the customer to find the entry point
- Only the coordinator have to scan for malware and SQL injection
- Only one webservice, it is therefore easy to maintain
- Code redundancy is low

Disadvantages:

- The singular webservice is a bottleneck, it limits to number of requests the service can handle
- The coordinator need to handle a lot of different types of file types.

Solution 2

Multiple input with a webservice for each filetype.



Advantage:

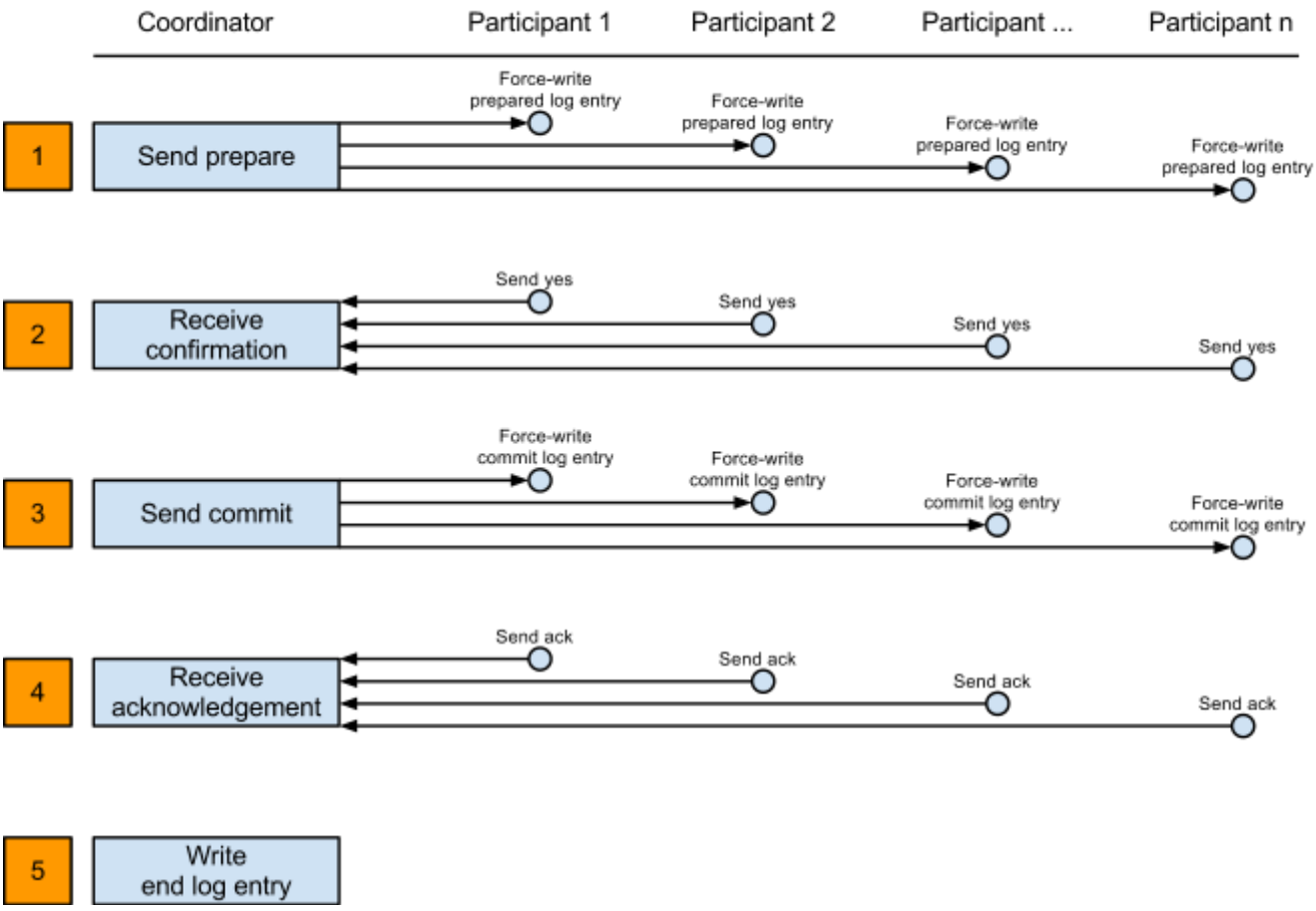
- The multiple webservices can handle alot of requests.

Disadvantage:

- Code redundancy is high
- Multiple webservices, which all need to be maintained.

Exercise 12: Fault Tolerance and Transactions

Implement 2 phase commit protocol for your application. Discuss advantages, disadvantages and reflect on problems.



Two phase commit protocol for a webservice running on distributed servers
The coordinator ensures a unanimous outcome of the transaction. Either all participants perform local commits or all perform local rollbacks.

1. Voting/Preparation phase - The coordinator ask all the participants if they are ready for a commit.
2. Coordinator receive answer from the participants. If all the participants reply “yes” the protocol continue to phase 3. If at least one of the participants either does not reply “yes” or does not reply at all, the commit will stop.
3. All the participants is ready to receive a commit. The coordinator sends the commit to all the participants.

4. If the commit has been successful the participant will send an acknowledgement package back to the coordinator. If the coordinator does not receive an acknowledgement package from at least one participants, the commit is cancelled and the participants is required to perform a local rollback.
5. The commit is successful, and the commit is now finished.

Pros

- distributed servers ACID

Cons

- Blocking protocol - Participants will never resolve their transactions if the coordinator fails permanently.
- A participant will be blocked after it has sent the coordinator an agreement message “yes”. The participant will be unblocked when a commit or rollback is received.
- Read locks
- Delay - 4 messages
- Availability - It is not certain that the webservice is always up due to blocks.

Problems

The major problem in distributed systems is that it can fail partially, so one or more servers fails while the others continue their normal operations. The servers exchange messages to agree upon whether a distributed transaction should be committed or aborted. Some of these messages might be lost due to local failures. This problem is solved by implementing a coordinator which makes sure that all the servers only commit, rollback, and abort if it is agreed upon. This solution has some advantages and disadvantages which can be read above.