MTAT.03.229 – Enterprise System Integration

Regular Exam – 9 January 2012

(40 points)

Notes:

- The exam is open-book and open-laptop. Web browsing is allowed, but you are not allowed to use e-mail clients nor instant messaging clients.
- If you find that there is not enough information in the text below to answer a given question, and you need to make additional assumptions, please write down your assumptions together with your answer.

PROBLEM 1 [15 POINTS]

1. We consider the billing system of an Internet Service Provider (ISP). The ISP classifies customer accounts into two categories:

1. CC: payment deducted from credit card
2. DD: payment deducted by direct debit from the customer's bank account

The ISP sends invoices by email to each customer on the first working day of each month (Day 1). On Day 7, DD customers have the full outstanding amount automatically debited from their bank account, while CC Customers have the full outstanding amount automatically debited from their credit card. If an automatic transaction fails for any reason, the customer is notified by email and a business process is started to resolve the non-payment issue. For payment processing, the ISP relies on an external service that can process debit requests both from bank accounts and credit cards.

Task 1.1 (12 points): Identify 3-5 services that the ISP needs to implement and to maintain. For each service, give a very brief description of its purpose and indicate its type (e.g. intermediary, process-centric).

2. The ISP now decides to introduce a third type of customer account:

3. Manual account: Customer account that has no credit card and no direct debit authority

Customers with manual accounts must ensure payment of the full outstanding amount is made by Day 7. They can pay by accessing their existing account on the ISP's web site and entering their credit card details and amount to be paid.

If a customer with a manual account has not paid the full outstanding amount by Day 7, the customer is notified by email and the business process for resolving non-payment incidents is started.

Task 1.2 (3 points): Identify one or multiple additional service(s) or application front-end(s) that the ISP needs to introduce into its SOA as a result of this change? Provide a very short description of each additional service and indicate its type. Do any of these additional services or application front-ends need to interact with any of the services listed in your previous answer?

PROBLEM 2 [25 POINTS]

Engineers at automobile design company CaaS routinely need to run simulations of new automobile designs or automobile component designs, in order to study their mechanical properties. Engineers design the automobile or automobile components using a Computer-Aided Design (CAD) tool. The design is exported as a CAD file and uploaded to the main server of the company's computer grid via an FTP client. Engineers then login to this main server and schedule their simulation on the computer cluster by using a command-line tool in the server (the command-line tool is a Python script). Simulations may take between one hour and 12 hours depending on the complexity of the design (average of 6 hours). However, since the capacity of the computing cluster is limited, it usually takes between 2 and 6 hours between the moment a simulation is scheduled by an engineer and the moment it starts. To monitor the status of a simulation, engineers need to login to the server and use another command-line tool. Engineers can also cancel an ongoing simulation using yet another command-line tool. Finally, engineers get an e-mail when their simulation is completed and they need to login to the main server to retrieve the file containing the simulation outputs. They then load this file into their CAD tool in order to analyze the results. Around 20 simulations per day are executed on CaaS's computer cluster. CaaS has 12 engineers who routinely run simulations and another 10 engineers who run simulations occasionally.

Several problems have been identified with the current way of running simulations. First, the process is cumbersome for engineers, and it is difficult to teach it to new engineers who join the company. As CaaS is foreseeing a significant expansion in the next year, and plans to double its number of engineers, it will become crucial to make this process simpler. Also, given this planned expansion of the company, it is clear that the capacity of the computer cluster will quickly reach its limits and scaling up the cluster would be expensive. To scale up, CaaS is planning to rent servers from cloud computing providers on a per-hour basis, to run some of the less critical simulations (simulations of the most critical vehicle models would still be done in the local computer cluster). The fact that external cloud-based servers will be used will make the procedure for starting and monitoring simulations more complex and engineers are against this additional complexity, given that it takes away time from them that could be used for more productive purposes. Finally, a third issue with the current simulation process is that no history of past simulations is kept. Sometimes one engineer runs a simulation, fetches the results into his/her laptop, and when another engineer wants to inspect the results of this simulation, it is difficult for them to retrieve these results. It has become evident that engineers need a searchable and browsable archive of all previous simulations.

Task 2: To address the above problems, you are asked to design a RESTful architecture for a simulation environment to be deployed in the main server of the computer cluster. Engineers will be able to start, monitor and retrieve the output of their simulations directly from their CAD tool by means of a special CAD tool plugin that will be developed for them. This plugin will be developed in Java Standard Edition (Java SE), which is the plugin development technology supported by the CAD tool. The plugin will interact with the server-side simulation environment via HTTP operations.

The output of your design should be a specification of an interface for the simulation environment. The interface should provide all operations that the CAD tool plugin

would need to invoke to start and manage simulations on the server-side simulation environment. This includes the ability to retrieve the full history of past simulations and the ability to retrieve the history of past simulations filtered by one or both of the following criteria:

- The time window when the simulations were run (captured by a start date and an end date of the time window)
- A keyword, which may be the simulation's unique identifier, part of the simulation's name or part of the simulation's description given by the engineer who started the simulation.

For each operation provided by the simulation environment you need to specify:

- The name of the operation
- A brief explanation of what the operation does
- A list of input and output parameters of the operation. For each parameter, give a name, a brief explanation of the meaning of the parameter, whether the parameter is "required" or "optional", and a description of how the parameter will be encoded, i.e. whether the parameter will be transferred using URL tunnelling (form-encoded) or in the body of the request/response, and in the latter case, what Internet media type (content-type) will be used.
- The relative URL where the operation will be available.
- Whether the operation uses the GET, POST, PUT or DELETE method.
- A list of HTTP status codes that the operation may return and for each status code, one or two sentences explaining under what conditions will this status code be returned.

For reference, you may find examples of RESTAPI documentations at:

- http://www.peej.co.uk/articles/restfully-delicious.html
- http://text-processing.com/docs/