Software Architecture 2014
Group Assignment

Designing a Service-Oriented Architecture (SOA) of a system for supporting sustainable consumption and generation of electricity

Problem context and description

IT University has decided to support an initiative for contributing to sustainable society by minimizing carbon footprint of electricity production. The initiative is aimed at encouraging households and industry consumers to adapt to renewable energy and to rely less on national electricity grid. To accomplish this task, it has been proposed that electricity consumers will be given tax deductions if their electricity consumption from national grid remains below a certain threshold. Moreover, if some of them have renewable energy units (e.g., solar panels and wind turbines) installed at their premises, they can sell their extra electricity through the national electricity grid.

IT University has been assigned the responsibility to design software for the proposed energy grid system. The software should support monitoring of the energy consumption by households and industry consumers. It should be able to determine how much electricity is being consumed (by green energy production units) and how much is being contributed to the main electricity grid. Selling price of the electricity being consumed or contributed to the main grid may vary during different time of the day (e.g., it is more expensive during peak hours when electricity demand is high and less expensive during other hours). Other than having the embedded components for monitoring energy utilization and contribution, the system should also have services for collecting different kinds of data and computing metrics for making special offers for low cost energy when the supply is more than demand. The system should expose its features through service APIs and web-based interfaces that can be used by different stakeholders of the system. Following are some of the expected functionalities for using the system.

Household and industry consumers should be able to perform following activities.

- They can view their energy consumption from national grid on hourly, daily, weekly, monthly and yearly basis.
- They can view their energy consumption from privately installed renewable energy sources on hourly, daily, weekly, monthly and yearly basis.
- They can define an upper limit on the energy consumption either by national grid or local production units.
- They can specify selling price of the energy they want to contribute.
- They can view their electricity consumption charges and amount they gain by selling electricity on hourly, daily, weekly, monthly and yearly basis.
- They can block their contribution to the national electricity grid.

Employees of national electricity grids should be able to perform following activities.

- They should be able to monitor energy consumption patterns by a particular consumer or region on hourly, daily, weekly, monthly and yearly basis.
- They should be able to generate reports based on consumption patterns.

- They can view the prices that an individual user wants for adding its electricity in the national grid. They can accept this price, reject it or enter into negotiation.
- They can block contribution by the privately operated renewable energy sources into national grid.

Employees at taxation should be able to perform following activities.

- They should be able to generate reports based on consumption patterns.
- They should be able to compute taxation reliefs (in %age) based on electricity consumption patterns.

Your team has been assigned to analyse the requirements and design Service-Oriented Architecture (SOA) of the electricity consuption monitoring and contributing system based on your knowledge and understanding of the above-described requirements. You are required to generate and specify the quality requirements using concrete scenarios (e.g., Performance, Usability, Security, and Maintainability) of the system called GreenEnergyGrid. Only top ten or twelves scenarios should be taken into account for design. Specify the required quality attributes in terms of the scenarios to characterise quality requirements, e.g., modifiability, security, performance, and usability. You are required to use the architectural and/or design patterns known to support Service-Oriented architecture design decision, SoaML, for modeling those decisions, and the template (used in the previous phase) for documenting the design decisions and their rationale.

P.S: No need to breakdown the scenarios into 6 elements.

Project objective:

- Gaining skills and experience in specifying quality attributes and making architectural design decisions for a medium size software system.
- Gaining skills and experience in identifying services to be incorporated in a service-oriented architecture, making architectural design decisions for a distributed software system using SOA patterns, and modelling SOA using SoaML.
- Getting familiarity with the role of Unit and Integration Testing and Automated Testing tools in DevOps methods with respect to software architecture.

Main Tasks:

Task 1 – Architecture Design

Based on your understanding and knowledge of the given set of requirements design a suitable SOA for the abovementioned system. The architecture design activity should be driven by the Architecturally Significant Requirements (ASRs) of the system (i.e., you may choose to use Personas and templates associated with it and 10-12 scenarios). The architecture design must incorporate the architectural and design patterns (i.e., for reference and learning about patterns, you can consult the GoF book or SOA patterns website). You are expected to document key architectural design decisions and possible alternative considered in a given template along with the rationale for the design decision (i.e., 5-6 main design decision should be documented). The architecture should be

described using any three (i.e., preferably logical, process, and physical views) of the 4+1 views and SoaML modelling language.
You are required to use the architectural and/or design patterns known to support Service-Oriented architecture design decision, SoaML, for modeling those decisions, and the template (used in the previous phase) for documenting the design decisions and their rationale.

Task 2 – Generating Test Cases for Automated Testing as part of DevOps
Your team is expected to generate and report at least 5 Unit Tests and 5 Integration Tests that can be used with Automated Testing facilites of Integrted Development Environments or Automated Testing tools. For generating and reporting test cases (i.e., unit and integration), you are expected to use the requirements specifications and scenarios generated for designing software archtiecture. You can also identify the automated tools that you can used for exeucting the designed test cases.

Breakdown of the main tasks:

1. Generate concrete scenarios for the given set of quality attributes (i.e., 10 scenarios, 2-3 quality attributes). Reason to choose a particular quality scenario should be explained.
2. Identify the potential services required to expose different features of the system. You are encouraged to follow the process (i.e., steps) described in [1] and [2] for identifying the potential services. However, whichever process you follow for this task, please report the steps followed clearly but concisely.
3. Document the different decisions taken in order to identify the required services.
4. Make design decisions using suitable architectural styles and patterns. Document design decisions along with the rationale for the choice made (in a given template) for capturing design decisions – only 6 key design decisions captured using templates. For the others, just describe the design decisions, patterns used, and justification for the use of design decisions with respect to quality attributes.
5. Model the Service Oriented Architecture (SOA) of the system by systematically selecting and applying appropriate elements of SoaML and show the use of patterns (e.g., described at the site [5]) (Refer to the class tutorial on SoaML or other relevant material).
6. For modelling the architecture, you can use any tool which supports modeling using SoaML [3]. It is strongly suggested that you try to use a tool which is either open source or offers a free trial period, for example, Objecteering [4] is one such tool. The designed SOA should at least contain the Service (including composed services) and Component layers.
7. Generate and report at least 5 Unit Tests and 5 Integration Tests that can be used with Automated Testing facilites of Integrted Development Environments or Automated Testing tools.

Submission: A report consisting of the artefacts produced in the abovementioned tasks. You are expected to provide a brief description and explanation of each of the task performed with respect to learning objectives of the course for this topic and any other reflections or observations. Please include all the source files (e.g., Latex source files, Word files, and diagrams and their sources which can be opened in the relevant tools) in a compressed folder for submission. Submit your assignment to an assignment box provided in Moodle.

Due Date: 10 October 2014, 5pm.

Some relevant sources:

1. James Densmore, Tim Bohn, An engineering paradigm for Service Oriented Architecture http://www.ibm.com/developerworks/rational/library/may07/densmore-bohn/index.html#notes
2. Samudra Gupta, Service Oriented architecture –Part1 and Part2: http://www.blogjava.net/enetor/archive/2006/05/22/47502.html
3. Service oriented architecture Modeling Language: http://www.omg.org/spec/SoaML/
4. Relevant chapters from DevOps: A Software Architect's Perspective by Bass, L. et al.
5. Objecteering: http://www.objecteering.com/ (A tool for modeling architecture using SoaML)
6. SOA patterns – a well known site is – http://www.soapatterns.org/

Following book is relevant for architectural styles and design patterns.

7. Buschmann, F., et al., 1996, Pattern-Oriented Software Architecture: A system of Patterns, John-Wiley.

Following are some of the reference material you can look at to get yourself familiar with domain.

8. Wattdepot, https://code.google.com/p/wattdepot/
9. Google PowerMeter, http://www.google.com/powermeter/about/