Lab 13

Consider the following case:

Best Electronics is a large electronics giant that has numerous retail stores throughout the country. When customers buy computers, TV's, stereos, and other electronic equipment, they can choose to purchase a support plan. Customer-facing technology experts (the "Sysops Squad") will then come to the customers residence (or work office) to fix problems with the electronic device.

Things have not been good with the Sysops Squad lately. The current trouble ticket system is a large monolithic application that was developed many years ago. Customers are complaining that consultants are never showing up due to lost tickets, and often times the wrong consultant shows up to fix something they know nothing about. Customers and call-centre staff have been complaining that the system is not always available for web-based or call-based problem ticket entry. Change is difficult and risky in this large monolith - whenever a change is made, it takes too long and something else usually breaks. Due to reliability issues, the monolithic system frequently "freezes up" or crashes - they think it's mostly due a spike in usage and the number of customers using the system. If something isn't done soon, Best Electronics will be forced to abandon this very lucrative business line and fire all of the experts (including you, the architect).

Current process in the monolithic system:
1. Sysops squad experts are added and maintained in the system through an administrator, who enters in their locale, availability, and skills.
2. Customers who have purchased the support plan can enter a problem ticket using the sysops squad website. Customer registration for the support service is part of the system. The system bills the customer on an annual basis when their support period ends by charging their registered credit card.
3. Once a trouble ticket is entered in the system, the system then determines which sysops squad expert would be the best fit for the job based on skills, current location, service area, and availability (free or currently on a job).
4. The sysops squad expert is then notified via a text message that they have a new ticket. Once this happens an email or SMS text message is sent to the customer (based on their profile preference) that the expert is on their way.
5. The sysops squad expert then uses a custom mobile application on their phone to access the ticketing system to retrieve the ticket information and location. The sysops squad expert can also access a knowledge base through the mobile app to find out what things have been done in the past to fix the problem.
6. Once the sysops squad expert fixes the problem, they mark the ticket as "complete". The sysops squad expert can then add information about the problem and fix to the knowledge base.
7. After the system receives notification that the ticket is complete, the system send an email to the customer with a link to a survey which the customer then fills out.

We decided to write a new trouble ticket system.

a. Find the architectural characteristics of the ticket system

b. Write one or more scenario’s for each architectural characteristic
c. Define your architecture in diagram(s). Show as much architectural important aspects as possible.
d. Find the risks of your proposed architecture
e. Find options to mitigate the risks you found in part a

Part 2

Financial Risk System

Background

A global investment bank based in London, New York and Singapore trades (buys and sells) financial products with other banks (counterparties). When share prices on the stock markets move up or down, the bank either makes money or loses it. At the end of the working day, the bank needs to gain a view of how much risk they are exposed to (e.g. of losing money) by running some calculations on the data held about their trades. The bank has an existing Trade Data System (TDS) and Reference Data System (RDS) but need a new Risk System.

Trade Data System

The Trade Data System maintains a store of all trades made by the bank. It is already configured
to generate a file-based XML export of trade data at the close of business (5pm) in New York.
The export includes the following information for every trade made by the bank:

- Trade ID
- Date
- Current trade value in US dollars
- Counterparty ID

Reference Data System

The Reference Data System maintains all of the reference data needed by the bank. This includes information about counterparties; each of which represents an individual, a bank, etc. A file based XML export is also available and includes basic information about each counterparty. A new organization-wide reference data system is due for completion in the next 3 months, with the current system eventually being decommissioned.

Functional Requirements

The high-level functional requirements for the new Risk System are as follows.

1. Import trade data from the Trade Data System.
2. Import counterparty data from the Reference Data System.
3. Join the two sets of data together, enriching the trade data with information about the counterparty.
4. For each counterparty, calculate the risk that the bank is exposed to.
5. Generate a report that can be imported into Microsoft Excel containing the risk figures for all counterparties known by the bank.
6. Distribute the report to the business users before the start of the next trading day (9am) in Singapore.
7. Provide a way for a subset of the business users to configure and maintain the external parameters used by the risk calculations.

Additional Requirements

• The system must be able to cope with trade volumes for the next 5 years.
• The Trade Data System export includes approximately 5000 trades now and it is anticipated that there will be an additional 10 trades per day.
• The Reference Data System counterparty export includes approximately 20,000 counterparties and growth will be negligible.
• There are 40-50 business users around the world that need access to the report.
• Risk reports basically should be available to users 24x7, but a small amount of downtime (less than 30 minutes per day) can be tolerated.
• This system must follow bank policy that states system access is restricted to authenticated and authorized users only.
• Reports must only be distributed to authorized users.
• Only a subset of the authorized users are permitted to modify the parameters used in the risk calculations.
• Although desirable, there are no single sign-on requirements
• All access to the system and reports will be within the confines of the bank's global network.
• The following events must be recorded in the system audit logs:
– Report generation.
– Modification of risk calculation parameters.
• It must be possible to understand the input data that was used in calculating risk.
• The system should take appropriate steps to recover from an error if possible, but all errors should be logged.
• All user interfaces will be presented in English only.
• All reports will be presented in English only.
• All trading values and risk figures will be presented in US dollars only.
• A Simple Network Management Protocol (SNMP) trap should be sent to the bank's Central Monitoring Service in the following circumstances:
– When there is a fatal error with a system component.
– When reports have not been generated before 9am Singapore time.
• Input files used in the risk calculation process must be retained for 1 year.

a. Find the architectural characteristics of the risk system
b. Write one or more scenario's for each architectural characteristic
c. Define your architecture in diagram(s). Show as much architectural important aspects as possible.

What to hand in?

1. A PDF for part 1
2. A PDF for part 2