Distribution and Integration Technologies (TDIN)

Assignment # 2

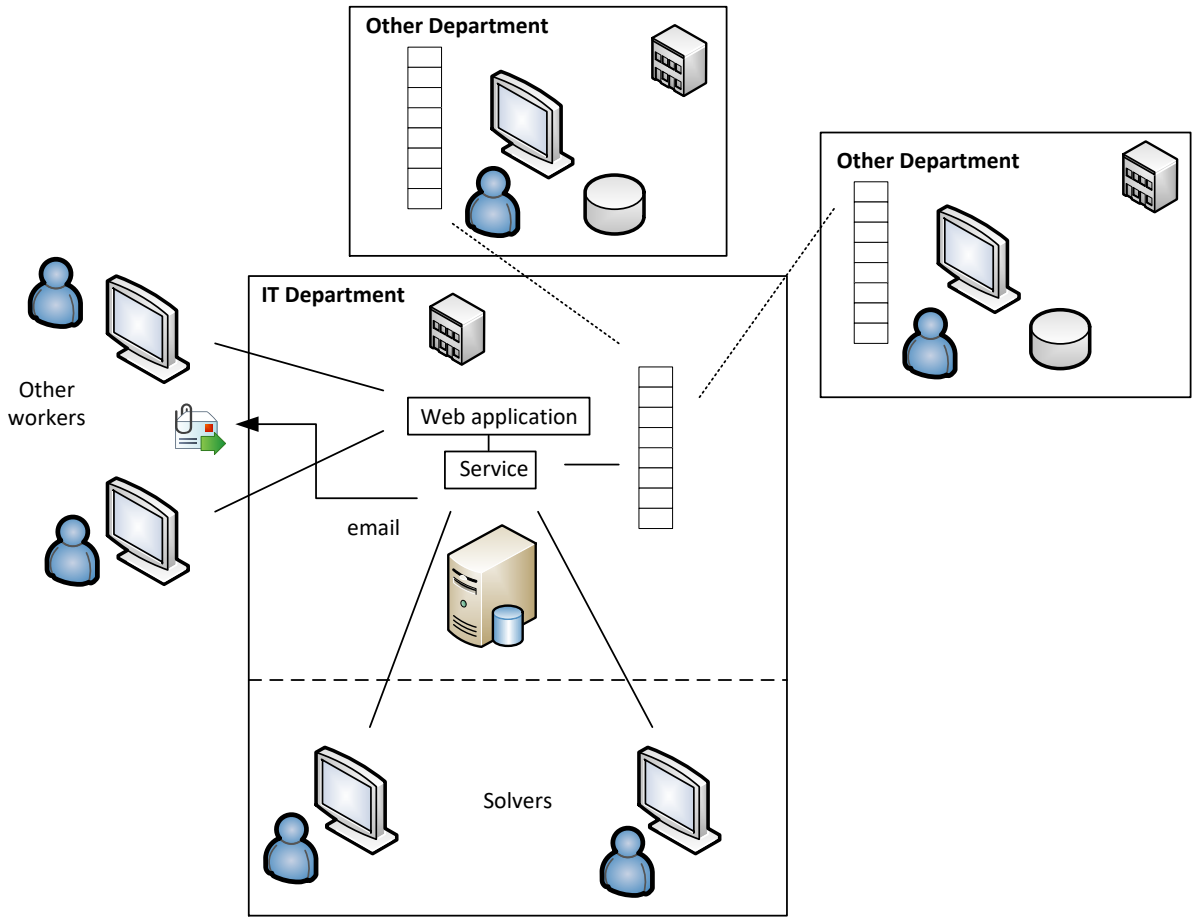
An enterprise distributed system

2017-18

Scenario

In a large organization there is an internal service to solve problems posed by the enterprise workers through 'trouble tickets'. A trouble ticket identifies its author and contains the description of a problem. The trouble ticket is then picked by a solver in the IT department. That solver may have a need to ask secondary questions to other departments. In that case he creates those secondary tickets and sends them to a specialized solver in another department. After those answers, he can finally respond to the original user via email.

In the IT department there is a permanent server hosting the web application through which the enterprise workers create their trouble ticket and follow their state. This server can also run any necessary services and is connected to the all enterprise intranet.



Requirements

The main requirements and operation of the all system are as follows:

- The IT web application accepts new 'trouble tickets' from the identified enterprise workers. Each trouble ticket should specify the author (name and email address), a title and a description. The application also assigns it a unique id and the date and time of creation.

- The trouble tickets evolve in the system through a series of states: 'unassigned' immediately after creation, 'assigned to (name of solver)' when it is picked up by a solver, 'waiting for answers', when (and if) any secondary question is emitted and not yet answered, 'solved' when the final email is sent.
- The enterprise workers can follow their 'trouble tickets' through a list of their own trouble tickets, displayed by the web application.
- The solvers run a GUI application. After identification, they can see their own assigned tickets not yet 'solved', and all 'unsigned' tickets. They can pick an 'unassigned' one. After that, that ticket will only show in his application. The solver can also emit secondary questions dispatched to other departments using asynchronous queues (the other departments applications are not always running). When all secondary questions (if any) have their answers and the solver is capable of solving the trouble ticket, he should write the final answer, which is sent to the original worker using email. After finishing a trouble ticket it disappears from the solver list.
- In the other departments there is only one solver, running a GUI application, allowing him to answer new questions (each one related to some original trouble ticket). The answer is sent directly to the IT server (always on) and should appear immediately (if the application is running), without user action, in the IT solver application that has created the question.
- All the trouble tickets, their state, secondary questions and answers are recorded in the IT server in a persistent way (file, relational or no relational BD).
- The other department applications should also persist the non-answered secondary questions.

Technologies

Implement this distributed application following the Service Oriented Architecture (SOA) principles. You can choose any appropriate technologies but following the previously stated requirements.

The service at the IT server should support several operations like:

- add a new trouble ticket
- change the state of a trouble ticket
- assign a solver
- associate a secondary question
- record the answer of a secondary question
- get the trouble tickets associated to a solver
- get the trouble tickets created by a worker
- …
- any other convenient operations

Realization

The web application, services and the GUI applications for the IT department and other departments should be developed. For testing and demonstration, the server, services and applications can run on the same computer. For simplification you can assume that only one other department exists.

Report

You should write a small report containing a detailed architecture specification (functionality, the modules and their interaction), the testing done, and a graphical representation (screen captures) of the main flows of use.

You should state also all the conditions and instructions to build, install and run your applications.