Authors' details

Gordon Mikkelsen
Hassan Mahdi
Hassan RH
Kristoffer Rath Hansen

Brief summary of the application's architecture, business and technical features

Enterprise integration patterns

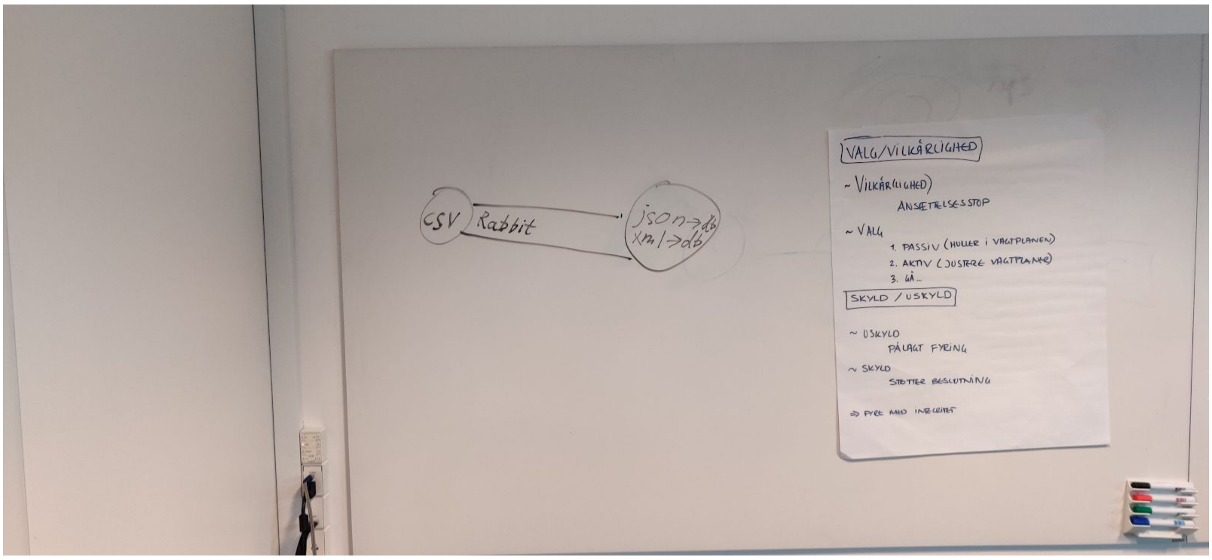
- Message Broker
  By following the message broker pattern we have implemented RabbitMq which is a middleware acting as a broker for being able to store a message and have it sent and received between two applications
- File Transfers
  The file transfers is achieved by transferring the csv file from the sender to the receiver, so the receiver can convert it to xml. This is done using RabbitMQ.
- Asynchronous Implementation with MSMQ
  We send the messages asynchronously using RabbitMQ.
- Message History
  The message history is achieved by logging all MySQL queries.
- Guaranteed Delivery
  RabbitMQ makes sure that every message is delivered successfully, else the application will not run correctly.

System that consists of a sender and a receiver where a message gets sent and a third party application namely RabbitMQ will act as a broker. When the message is sent the option that will be given in the console is either press "a" and the file convert a csv file into xml or when press "b" it will store the data in a database.

Regarding on what the system must is that it is required from the companies side that , the system must accept an existing file which is a csv file and either convert it to a xml file for future use or save the data in a database and timestamp and timestamp it.

The technical aspects of system is that as mentioned before is that it consist of two applications , sender and receiver which is written in C# .NET 4.6.1 version. A third party library has been implemented namely "RabbitMQ.Client" into the both c# projects and have a third party application up and running which is the rabbitmq-server version 3.8.1" Without the RabbitMq-server the C# project won't work as intended. For storing data we used mysql database.

Diagrams, illustrating the above



This diagram shows how we convert it from csv to xml. At the time of making this diagram, we were not sure if we wanted to do it in json or xml, but we chose xml.

Installation instructions

Expectations

We expect you have the following installed in advance:
C# .NET 4.6.1
RabbitMQ 3.8.1

Program

Download the program from Github. When you have followed the instructions below, you can run it.

Set up database

We need to set up the database. Make sure you have MySQL server and MySQL Workbench installed.

1. Open workbench and log on the running server.
2. To the left, make sure the Navigator is on Schemas and right click and make a new schema.
3. Give it a name and click apply.
4. Double click the new schema
5. Take the provided dump and load into a query.
6. Run that query.
7. You now have a database set up.

Logging of queries

To enable logging of queries we need to set up MySQL workbench to log this (it would be similar in other MySQL installs since it’s a core functionality of MySQL).

1. Go to Server->Options File.
2. Find the Logging tab and go to that.
3. Enable general log
4. Make sure the file is created in the right place.
5. Apply the changes
6. Restart the MySQL Server
7. Done. Enjoy the log!