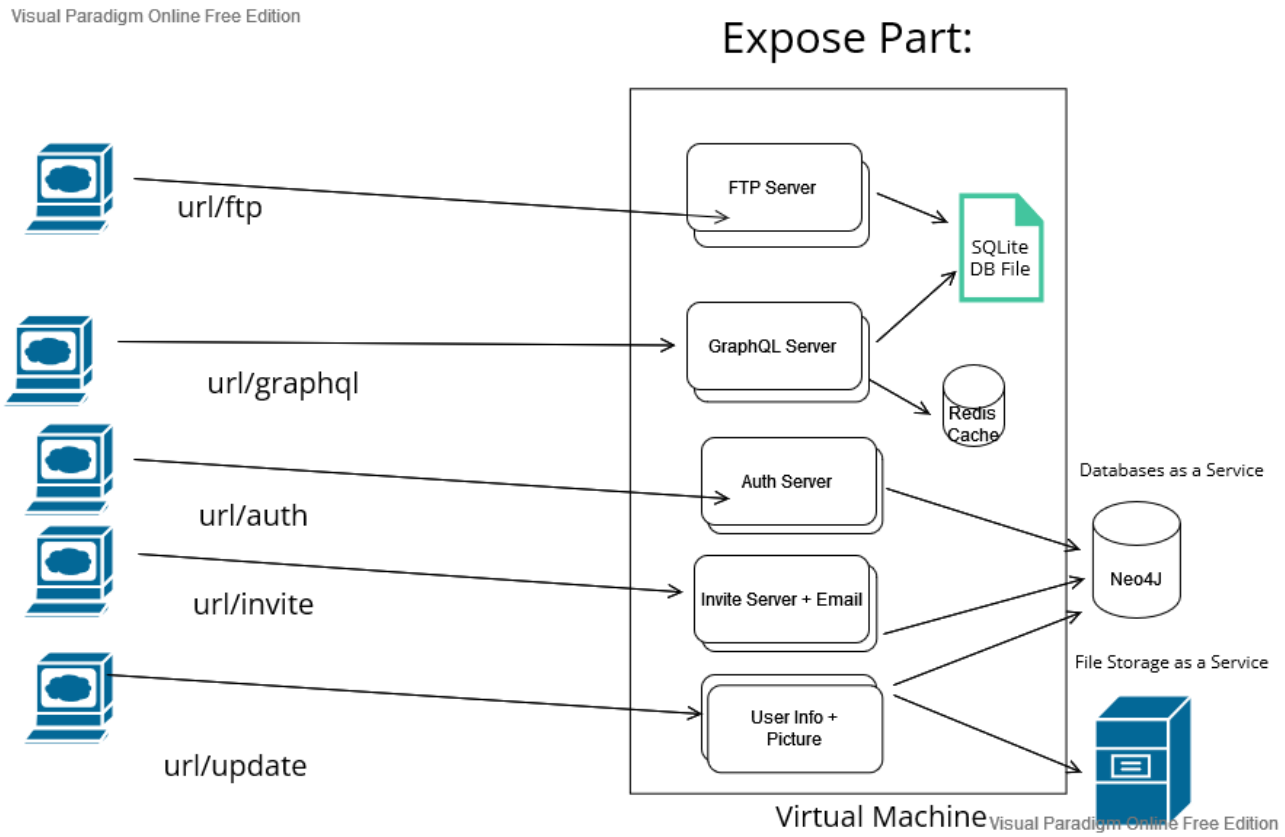
The Overview of the system

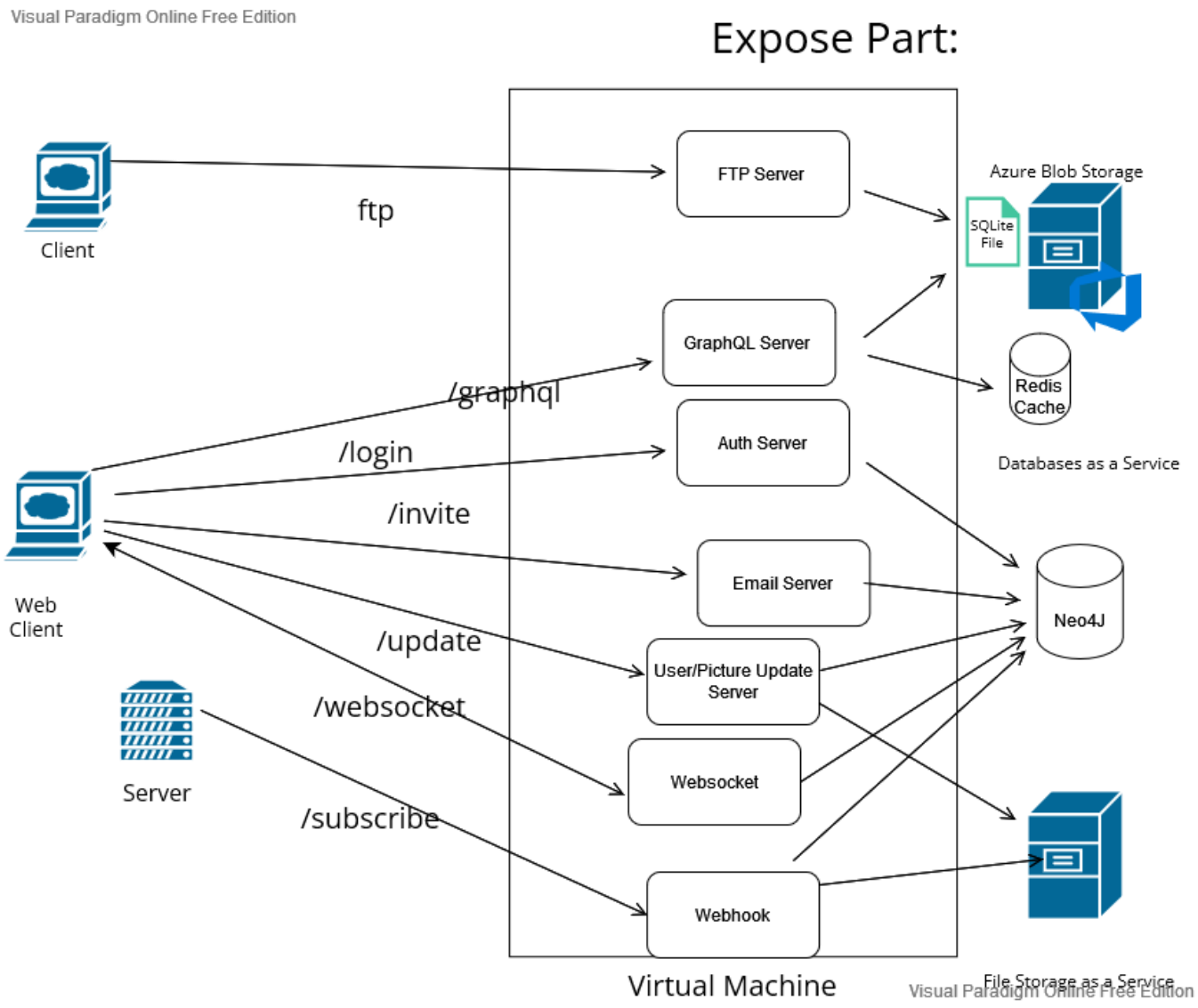
Initial Diagram made after reading the requirements.

We read the initial requirements. We choose some initial servers to create and what sort of storage those servers would use as well as which database we would use in the project. With most of the services being thought of as REST API servers. And only the GraphQL server and FTP server would be different. The GraphQL and FTP would use the same products.db file, so it would be stored in public storage. GraphQL would also use Redis as cache layer.



Second diagram created

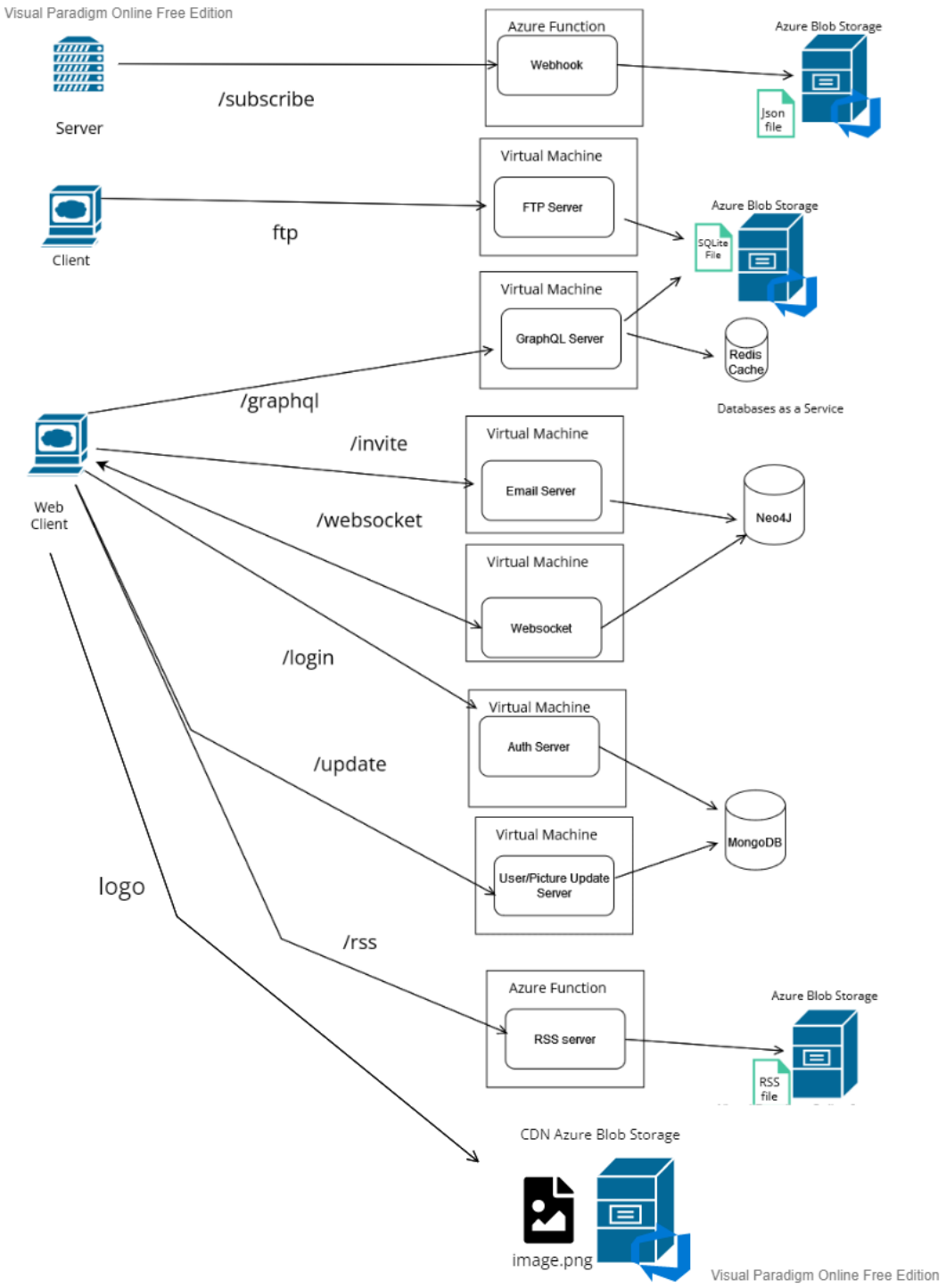
Added 2 other servers – Webhook and Websockets. Currently All services are running on the same virtual machine.



Final version:

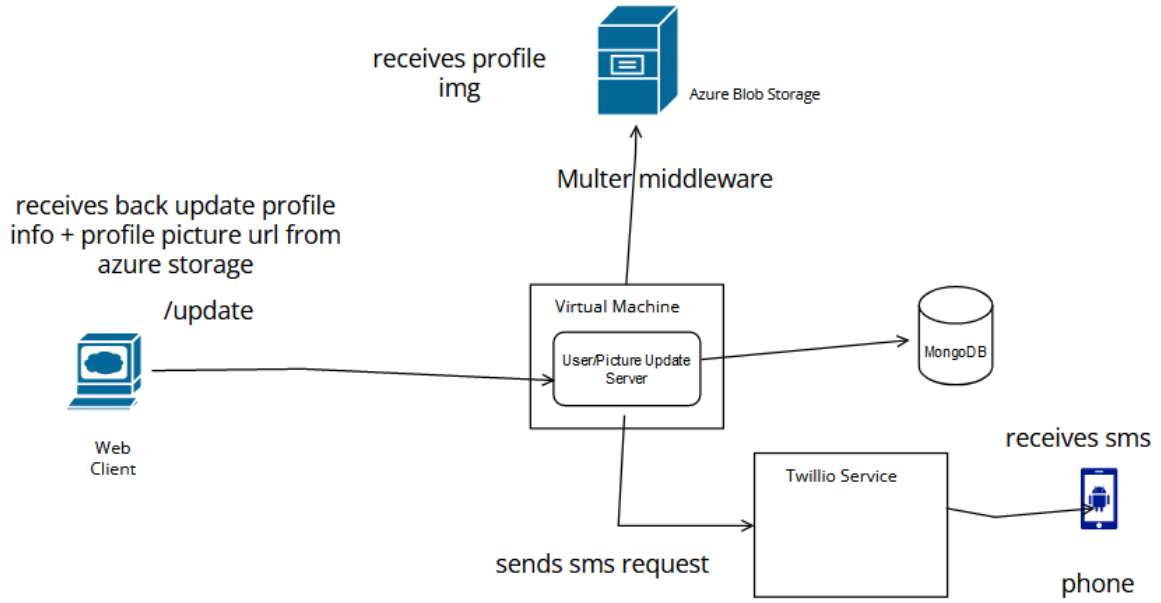
In the Final Version, we split up all services in separate virtual machines. RSS server added. Webhook + RSS server were made as Azure Functions. Different services access different databases with only minimal information shared. Save Webhook and RSS server files in Azure webstorage. Lastly, we implemented the logo CDN trough Azure Blob Storage.

Expose Part



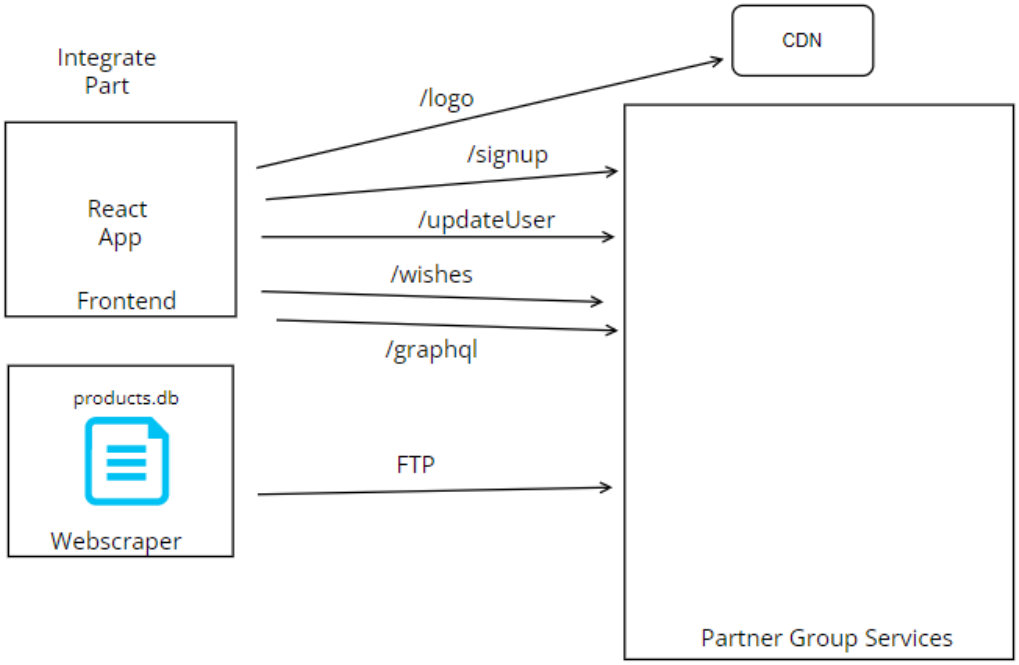
Update profile info diagram

The client sends a request with the user data, chosen image and content type as “multipart/form-data”. The request is coming to the “User Update” Server, it gets handled by the multer middleware that extracts the file and uploads it to Azure Blob Storage as well as the sms middleware that sends a message to user’s phone through Twilio Service. We then get data provided in request and update corresponding user fields in MongoDB. Lastly we send back a response with the updated info and picture url from Azure storage.



Integrate Part

For the integrate part we created a website that would display content to the user and integrate with the Partner Group. On the website we display our logo from partner’s CDN as well as communicate with their services to do authentication, query the products, wish items and update user data. We also implemented a webscraper that scrapes data from the chosen website, saves data in products.db and saves it over FTP to partner group’s server.



The creators of different services:

Subscribe webhook – Alexandru Sandrovschii
FTP Server – Radu Cazacu
GraphQL Server (aka the products path) – Alexandru Sandrovschii
Email Server – Aleksandar Minchev
Websocket (aka friends path) –Radu Cazacu
Authentification Server – Alexandru Sandrovschii
UserPicture Update Server – Aleksandar Minchev
RSS Server – Aleksandar Minchev
Frontend integration - Katerina
Webscraper- Katerina

Links to the expose services:

http://68.219.117.56:8001/ - Email Server,
http://13.74.136.176:8000/ - Auth Server ,
http://20.123.30.112:8004/ - Update Info Server,
https://rss-function-system-integration.azurewebsites.net/ - RSS,
https://system-integration-webhook-functions.azurewebsites.net/api/subscribe - Subscribe Webhook ,
http://68.219.99.197:8002/ - Graphql Server (aka the products path),
http://40.115.107.130:8080/ - FTP server (username:radu,password:admin)
http://68.219.97.37:8003/ - Websocket server (aka friends path)

Links to the integrate services:

https://silogogoat.blob.core.windows.net/logo/silogo.png - CDN
React App (sgoatfrontend.azurewebsites.net) - Products frontend

Links to documentation:

https://storagesystemintegration.z13.web.core.windows.net/ - Documentation for GraphQL
https://rss-function-system-integration-apim.portal.azure-api.net/ - All other services Documentation