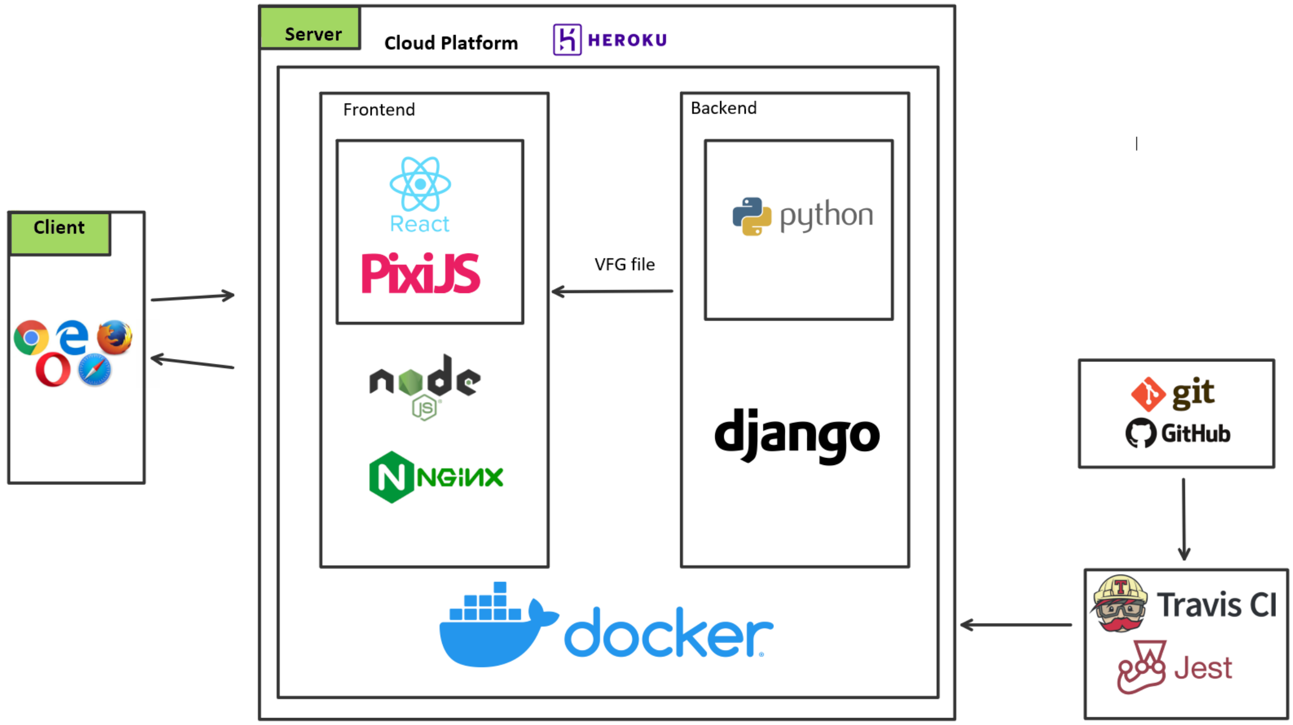
# Technologies and Learning Resources

The system consists of the frontend and the backend. Note that in our project, we only do the frontend development.

As the frontend framework is changed from Unity to JavaScript, the system architecture deployment related to the frontend has also been adjusted accordingly. In order to improve the frontend rendering performance on browsers, our frontend project plans to use React framework. The web server has also been changed from Apache to lightweight Nginx.

The planned system architecture is shown below. Since this project does not involve any changes to the backend and database, this page will only introduce the tools that will be used in this frontend project.



The planned tools of the project are listed below.

- **Technologies and Learning Resources#GitHub**
- **Technologies and Learning Resources#Visual Studio Code**
- **Technologies and Learning Resources#Travis CI+Jest**
- **Technologies and Learning Resources#Nginx**
- **Technologies and Learning Resources#Docker**
- **Technologies and Learning Resources#Visual Studio Code API**
- **Technologies and Learning Resources#PixiJS**
- **Technologies and Learning Resources#React**
- **Technical Environment Plan#MaterialUI**

## GitHub

The project uses GitHub as a tool for project iteration, collaboration within team and version control. To increase the efficiency of developing, it’s recommended to install Git on the development environment.

## Visual Studio Code

Visual Studio Code is a great code editor for many programming languages. It has Git commands built-in and many extensions to improve developer ‘s productivity.

- ## Travis CI+Jest

Travis CI is a useful tool to test and can easily synchronize the repository on GitHub. It was also adopted by the original project Planimation. Here, we continue to use its service to continuous integration.

Jest is a test framework designed for JavaScript.

- ## Nginx

Nginx is a fast and dynamic web server. Compared to Apache, 4 times more concurrent connections are handled Technologies and Learning Resources#[1]. We use Nginx to replace Apache Web Server when deploying.

- ## Docker

The original project uses Docker to run the application on the cloud platform. Therefore, we continue to use it in this project, although several modifications need to be done during deployment.

- ## Visual Studio Code API

Use Visual Studio Code API to build a plugin(extension) for VS Code.

- ## PixiJS

PixiJS is a HTML5 Creation Engine which is suggested to be used in the project. To gain better performance, our project plans to use React+PixiJS.

- ## React

An open-source declarative frontend framework based on JavaScript. React encourages the developer to break the UI interface into components which makes code is easier to understand and maintain. React offers a better performance on rendering UI in the browsers, so it’s very efficient for a frontend project requiring lots of interactions and animation

- ## MateriaUI

React components for faster and easier web development. Build your own design system, or start with Material Design. which provides a robust, customizable, and accessible library of foundational and advanced components, enabling you to build your own design system and develop React applications faster.

## References:

[1] https://blog.coolicehost.com/ten-great-advantages-of-nginx/

## Resources:

- Travis CI: https://docs.travis-ci.com/
- Jest: https://jestjs.io/docs/getting-started
- Docker: https://docs.docker.com/
- Visual Studio Code API: https://code.visualstudio.com/api
- Pixi JS: https://www.pixijs.com/
- Pixi JS Tutorial : https://github.com/kittykatattack/learningPixi
- Pixi JS Demos : https://pixijs.io/examples/#/demos-basic/container.js
- React: https://reactjs.org/docs/getting-started.html
- ReactPIXI: https://reactpixi.org/
- MaterialUI: https://mui.com/