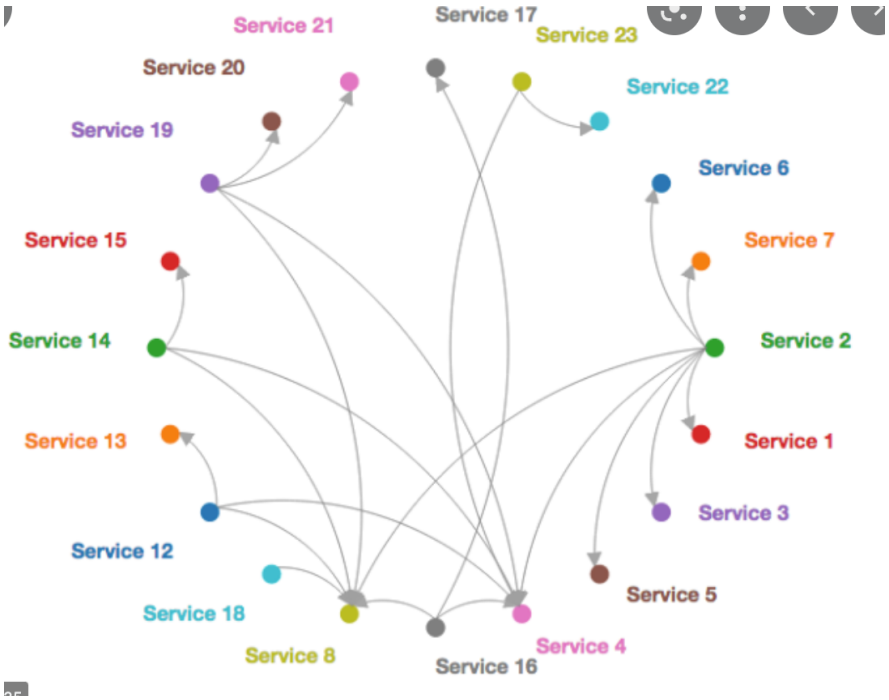
OpenAPI Developer Portal

Requirement

When working with microservices it is useful to have an inventory of all microservices, their APIs, who owns them as well as a visualisation of their dependencies. Such a visualisation allows a team to know who may be impacted if they make backward incompatible changes to their microservice APIs. This project is to create a tool that provides a database of microservices and their details and allows visualisation of their dependencies.

The functionality should be as follows:

- Provide the ability to enter details of a microservice including (add more as you see beneficial)
    - Microservice name
    - Category (e.g. Database, messaging, application)
    - Microservice lead engineer: name and email address
    - Description of microservice
    - Date created
    - Version number (maintain a history of each version for a given name)
    - Names of microservices this uses (its dependencies)
    - OpenAPI specification (this is vital) and all versions of an API
- Provide search facility:
    - Find a microservice by name
    - Find microservices in a category
    - List details of dependent microservices including lead engineer contact details
    - List all versions of a microservice
    - List all microservices with no dependencies
- Nice to have: a visualisation of the dependencies of the microservices in the database using a library like D3.js. An example view is below.



Technical Requirement

- The requirement is for the server side to be developed in Java using Spring Boot and should be based on REST microservices
- Data should be stored in a persistent store
- The application should be packaged in Docker and the application started with Docker compose
- Each OpenAPI document for a microservice should be available at a URL of the form: `http://server:portno/api/service_name/openapi.yaml`
- The OpenAPI presentation is ideally similar to Swagger editor (use this if you wish) `https://editor.swagger.io/`
- You should use SonarQube as part of your quality gate and ensure your code is of a good quality: zero security errors/warnings are mandatory
- Automated tests should be provided
    - Test coverage in sonarqube is mandatory (80% minimum)
- You should develop your code in a git repository and perform code reviews using gerrit
- The visualisation of dependencies is a nice to have. The core functionality of a searchable database of microservice is the key part

Bonus Technical Requirements

- Use Jenkins to build the software