Heatington

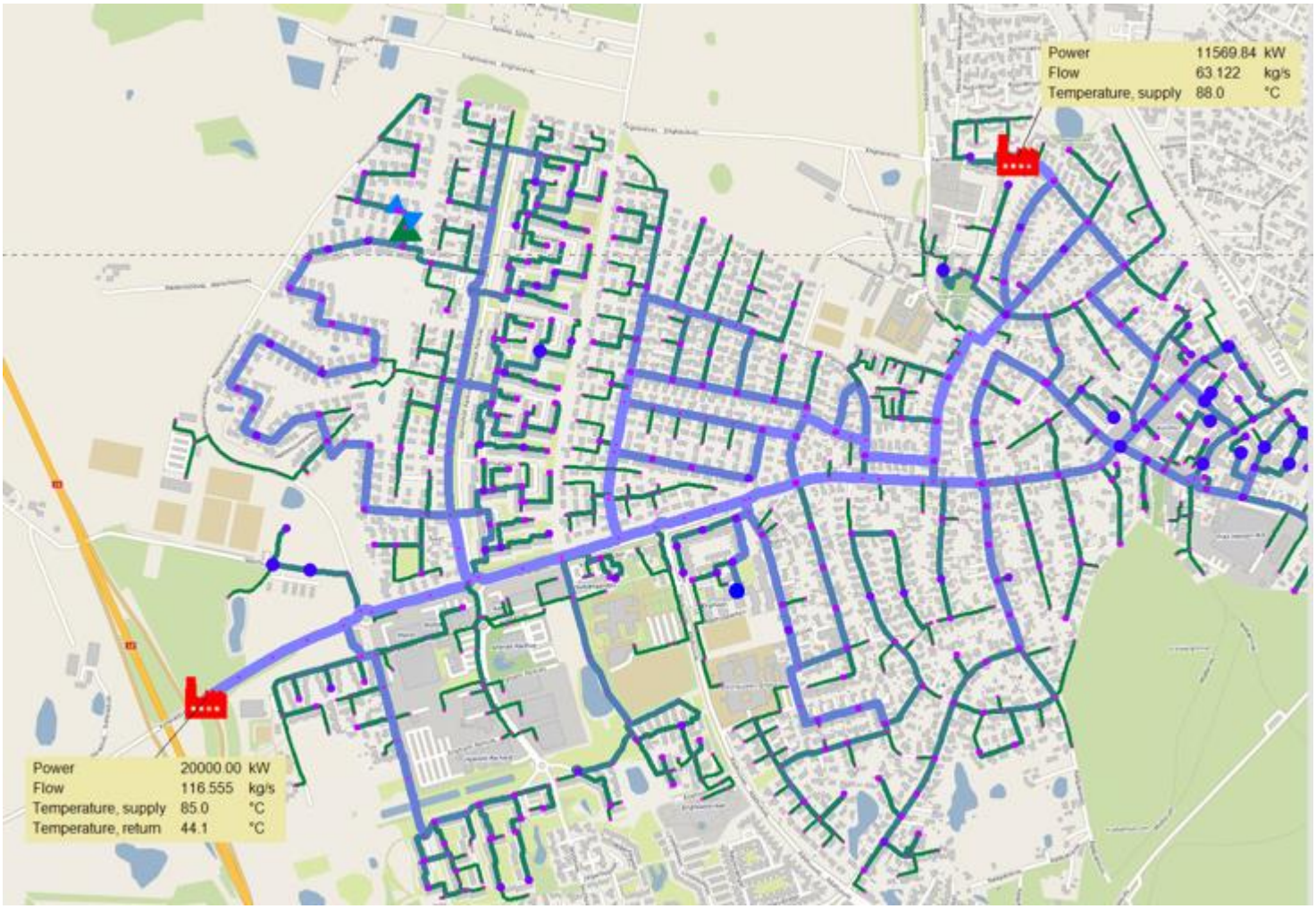
> Open Coding Style Guidelines

> How to install Docfx

What are we developing

- Heat production optimization for a district heating utility
  - Secure heat availability for all buildings in the district heating network
  - Produce heat for the lowest costs
  - Utilize the electricity market for the highest profit or lowest costs
- Five Components
  - Asset Manager (AM)
  - Source Data Manager (SDM)
  - Result Data Manager (RDM)
  - Optimizer (OPT)
  - Data Visualization (DV)
- Two Scenarios
  - Single heating area, one gas boiler, one oil boiler
  - Single heating area, one gas boiler, one oil boiler, one gas motor, one electric boiler
- Two Periods
  - Winter period
  - Summer Period

Example of heating net



Component Diagram

component-diagram.png

Class Diagram

class-diagram.png

Introduction

This is Group's number 16, Second Semester Project Documentation

Project Name: Heatington

Installation

following the official documentation

> Documentation Link

To install using .NET SDK (it has to be installed previously)

```
dotnet tool update -g docfx
```

How does it work?

DocFx automatically creates documentation based on the contents of a file. This Documentation is available in the `API` tab on server.

If you want to add your own, more detailed documentation. Simply add a Markdown (`.md`) file to `/docs` directory in the right folder coresponding to the real location of a class in a project.

It's available in the `Docs` tab.

How to run it?

Run in the same directory as `docfx.json`

```
docfx docfx.json --serve
```

Now you can preview the website on http://localhost:8080.

If you want to rebuild your documentation run this command in a *new* terminal window

```
docfx docfx.json
```

You may encounter some probles with running `docfx` command

Sometimes `docfx` is not automatically added to your `PATH`. In order to fix that find the location of your dotnet tools. For UNIX based systems it should be `~/.dotnet/tools/docfx`. Then either add

your `dotnet/tools` directory to global `PATH` or just run `docfx` from explicitly specified path.

e.g

```
~/.dotnet/tools/docfx docfx.json --serve
```

How to add Markdown files?

in `toc.yml` file inside your directory add a desired markdown file, then add this to the YAML file

```
- name: <Name of the component>
  href: <NameOfTheFile>.md
```

How to add new directories?

in `toc.yml` file inside your directory add a new directory then add this to the YAML file

```
- name: <Name of the directory>
  href: <PathToTheDirectory>/toc.yml
```

REMEMBER to add the `toc.yml` file later inside of the new directory, pointing onto the `.md` files

Useful Resources

DocFx runs their own (of course) version of Markdown

> Brief Introduction to a DocFx (15min Video)

> [Documentation about docfx markdown]

Namespace Heatington.AssetManager

Classes

AssetManager

HeatingGrid

ProductionUnit

Enums

ProductionUnitsEnum