

Diagram extension for Python-Markdown

using Kroki

Hideyuki KATO

Copyright © 2025 Hideyuki KATO

Table of contents

1. markdown-kroki	3
1.1 Install	3
1.2 Requirements	3
1.3 Usage	3
1.4 Process flow	4
2. Diagrams	6
2.1 ActDiag	6
2.2 BlockDiag	7
2.3 Bytefield	8
2.4 C4 with PlantUML	9
2.5 D2	10
2.6 DBML	12
2.7 Dita	13
2.8 Erd	14
2.9 Excalidraw	15
2.10 Graphviz	16
2.11 Mermaid	17
2.12 Nomnoml	19
2.13 NwDiag	20
2.14 PacketDoag	21
2.15 Pikchr	22
2.16 PlantUML	23
2.17 RackDiag	25
2.18 SeqDiag	26
2.19 Structurizr	27
2.20 Svgbob	28
2.21 Symbolator	29
2.22 Tikz	30
2.23 Vega	31
2.24 Vega-Lite	32
2.25 WaveDrom	33
2.26 WireViz	34
3. Experimental	35
3.1 MkDocs Plugin (Experimental)	35

1.

[pypi](#)
[v0.3.0](#)
[license](#)
[MIT](#)
[python](#)
[3.9](#) | [3.10](#) | [3.11](#) | [3.12](#) | [3.13](#)

Diagram extension for [Python-Markdown](#) using [Kroki server](#).

This extension converts various diagram code blocks into Base64 encoded [data: URI](#) or direct image link. This enables PDF generation with tools like [MkDocs to PDF/WeasyPrint](#) without requiring JavaScript(e.g. [Mermaid](#))

1.1 Install

```
pip install markdown-kroki
```

1.2 Requirements

1.2.1 Internet access to the public Kroki server

Default setting with no options.

1.2.2 Self-Managed Kroki server (recommended)

Here is a sample Docker Compose file.

ref. [Kroki.io](#) > [Install](#) > [Using Docker or Podman](#)

```
docker compose up -d
```

The default port used by MkDocs (`mkdocs serve`) may conflict with the default port of a Dockerized Kroki instance. Consequently, you will need to change the port configuration for one of them.

1.3 Usage

```
'''{diagram language} format=[svg|png] {img tag attribute}="value" {diagram option}="value"'''
```

- format (optional): Output image format default to svg
- img tag attribute (optional): alt, width, height, class, id, style, title
- diagram option (optional): refer to [Diagram options](#)

1.3.1 MkDocs Integration

```
# mkdocs.yml
markdown_extensions:
  - markdown_kroki:
      kroki_url: http://localhost:18000 # default: https://kroki.io
      img_src: link # default: data/data URI, link/direct link
```

1.3.2 Pelican Integration

```
# pelicanconf.py
MARKDOWN = {
  'extension_configs': {
    'markdown.extensions.codehilite': {'css_class': 'highlight'},
    'markdown.extensions.extra': {},
    'markdown_kroki': { # Add these
      'kroki_url': 'http://localhost:18000',
      'img_src': 'link'
    }
  }
```

```

    },
    'output_format': 'html5',
}

```

1.3.3 Python code

```

import markdown
from markdown_kroki import KrokiDiagramExtension

markdown_text = """
plantuml format="svg" theme="sketchy-outline" width="300"
@startuml
Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response
@enduml
"""

html_output = markdown.markdown(markdown_text, extensions=[
    KrokiDiagramExtension(kroki_url='https://kroki.io')])

print(html_output)

```

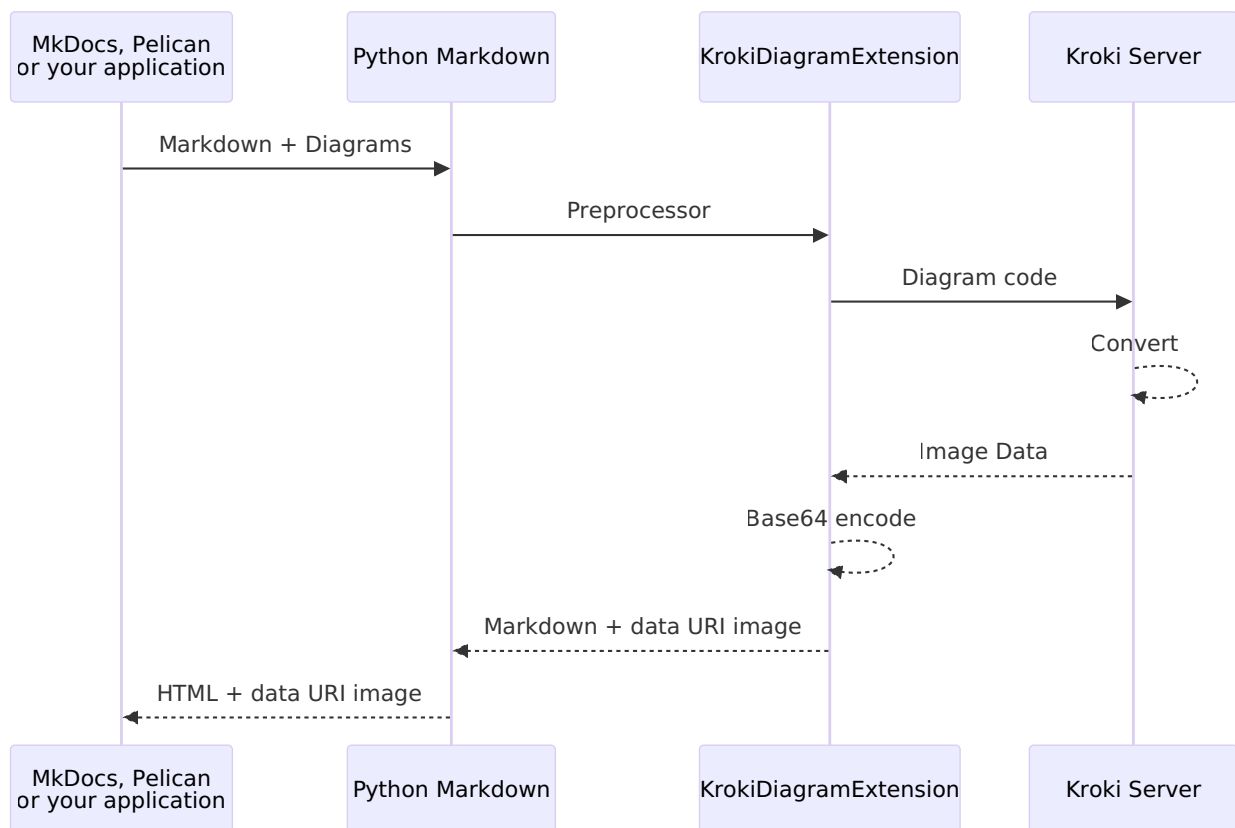
```

<p></p>

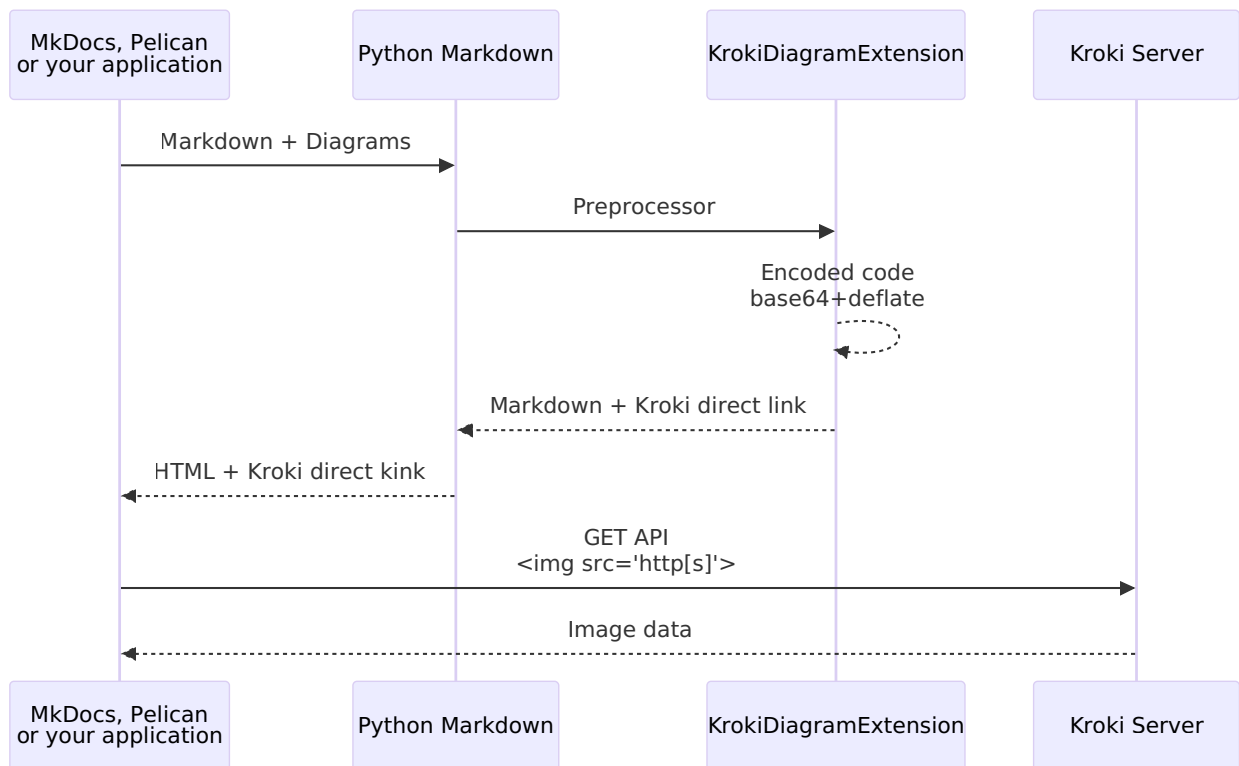
```

1.4 Process flow

1.4.1 data URI

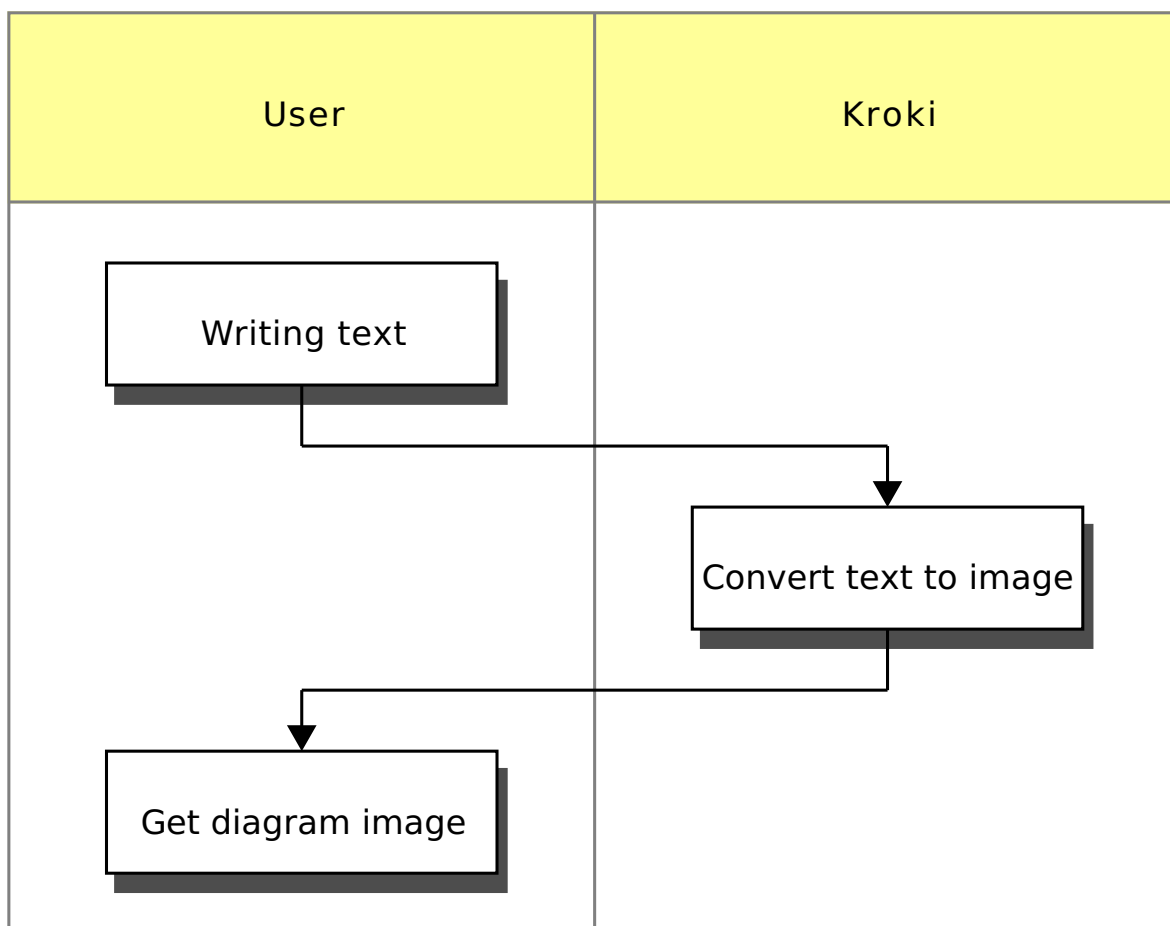


1.4.2 Direct link (GET API)

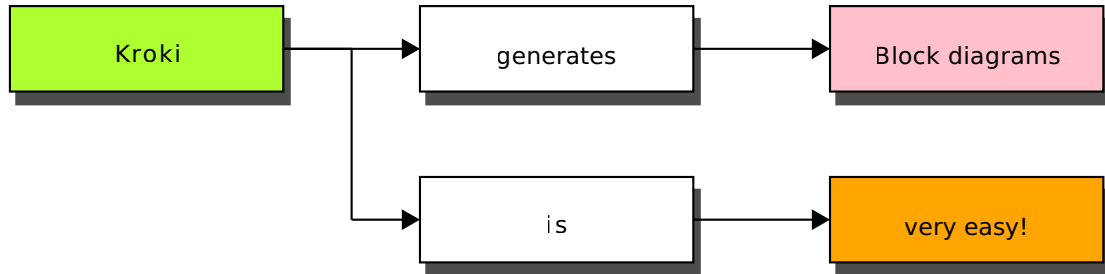


2. Diagrams

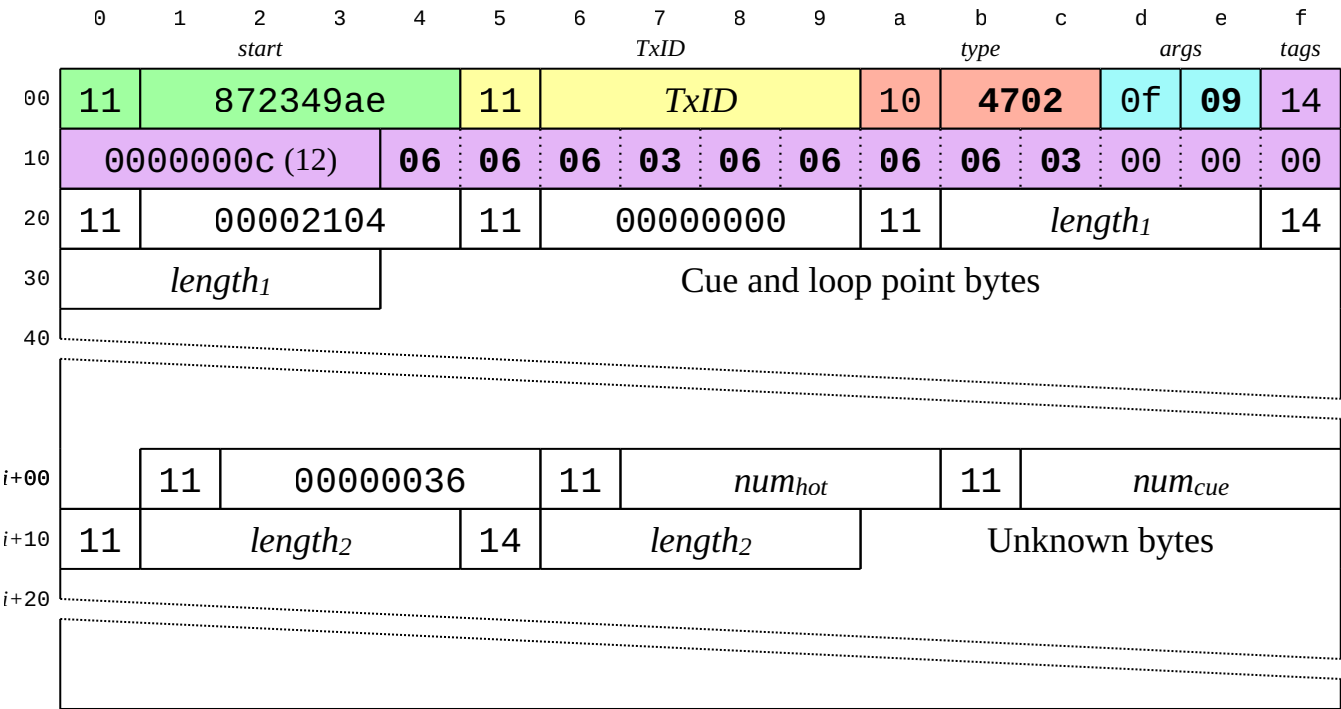
2.1 ActDiag



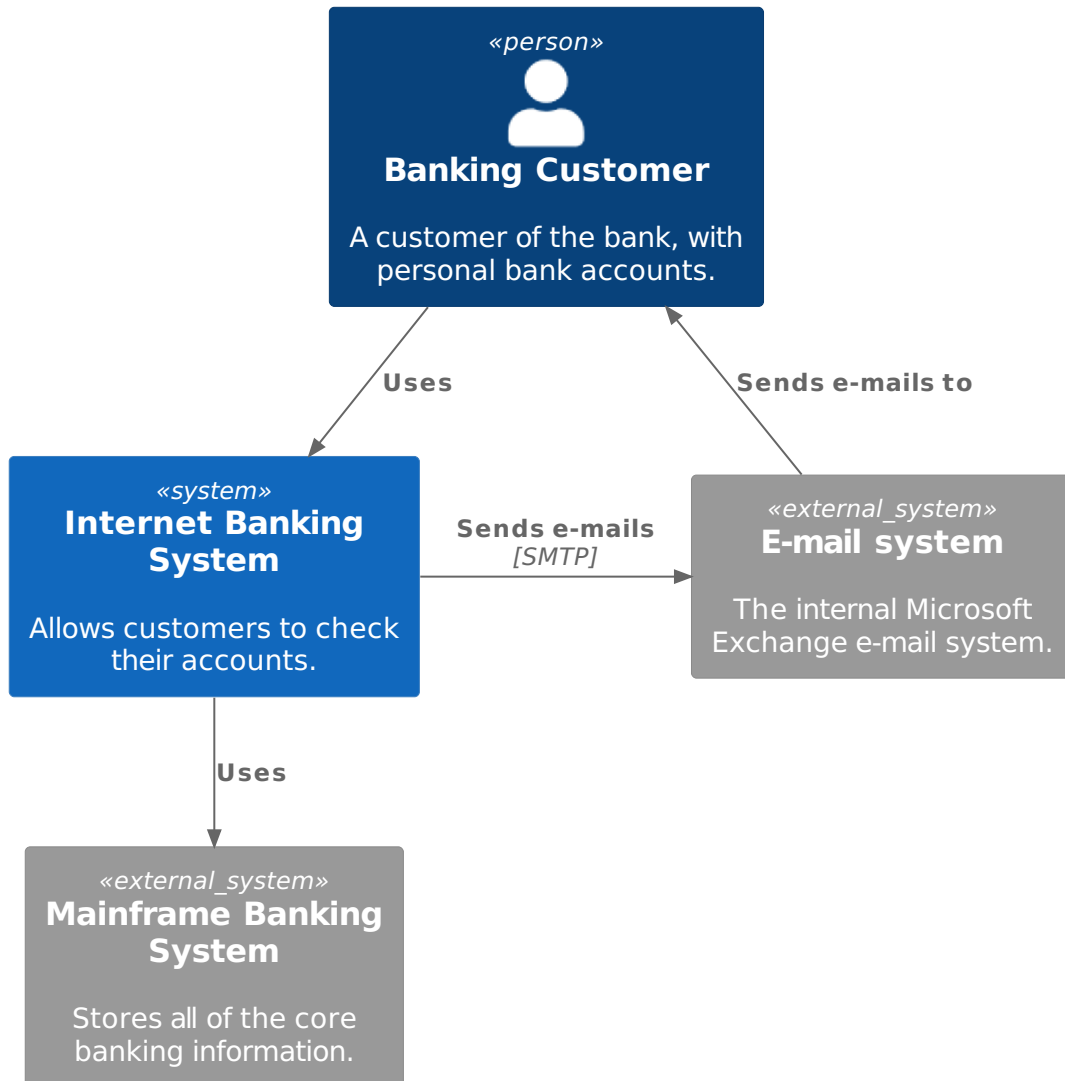
2.2 BlockDiag



2.3 Bytefield



2.4 C4 with PlantUML

System Context diagram for Internet Banking System

2.5 D2

Warning

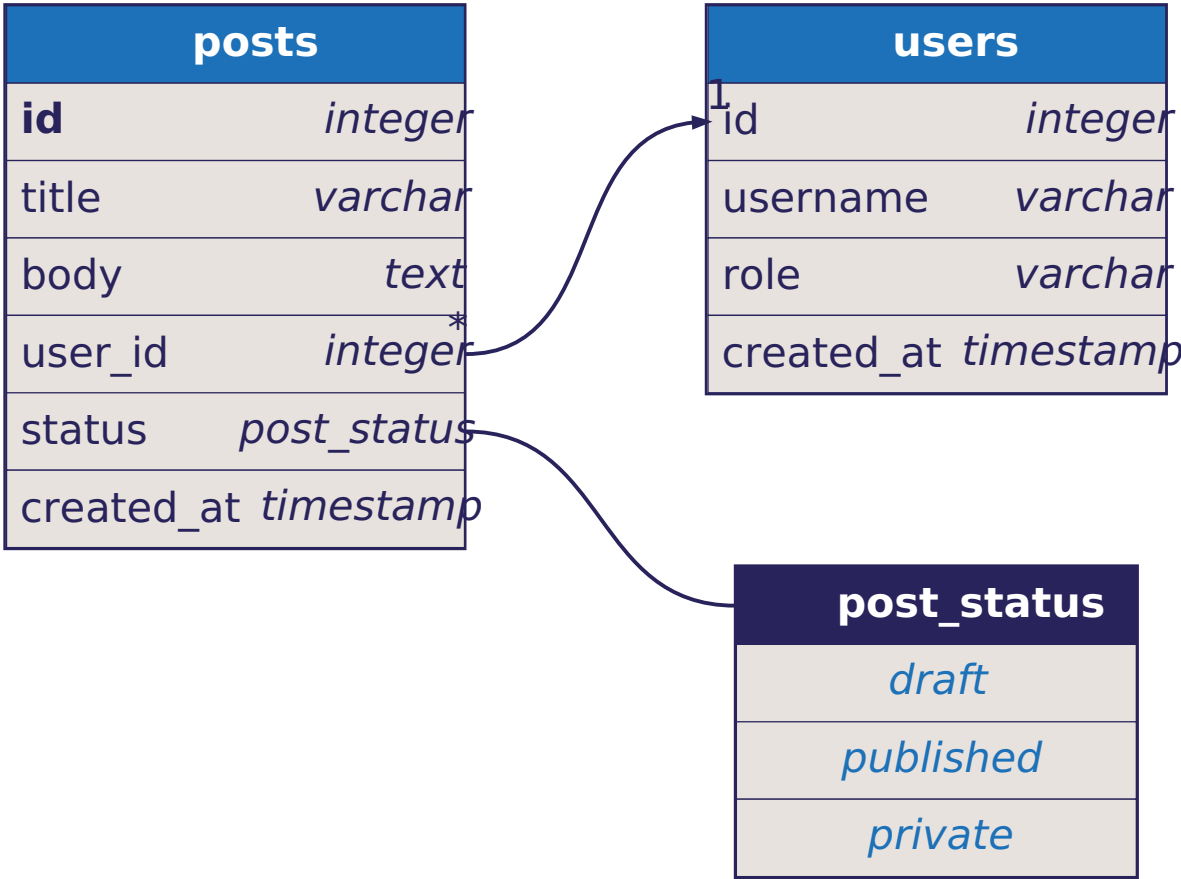
PDF rendering will fail with WeasyPrint. However, it will succeed with Playwright/Chromium.



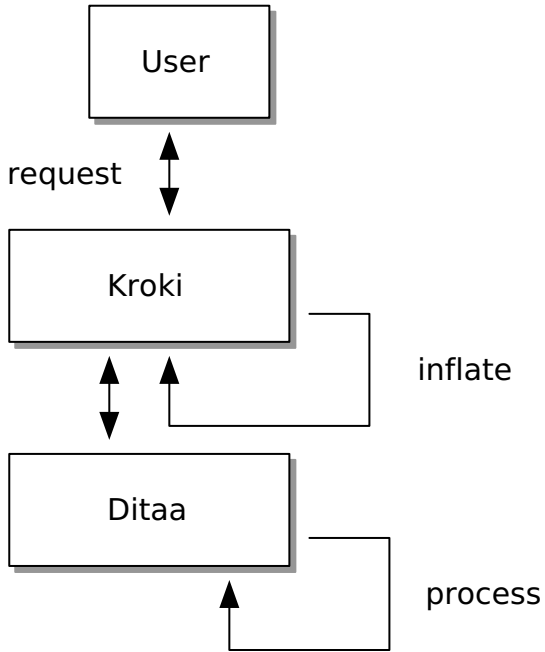
2.5.1 with Options



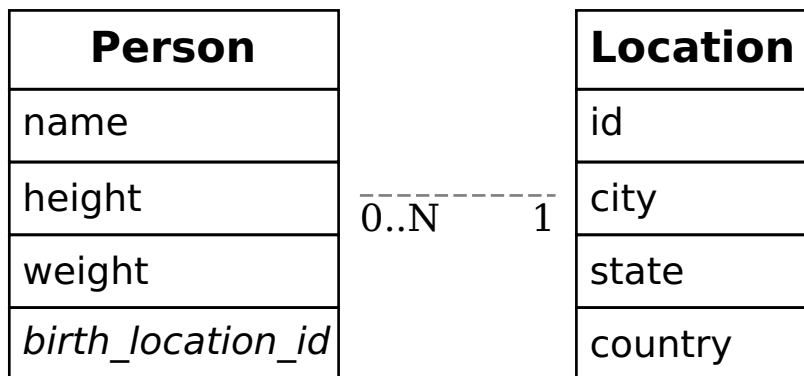
2.6 DBML



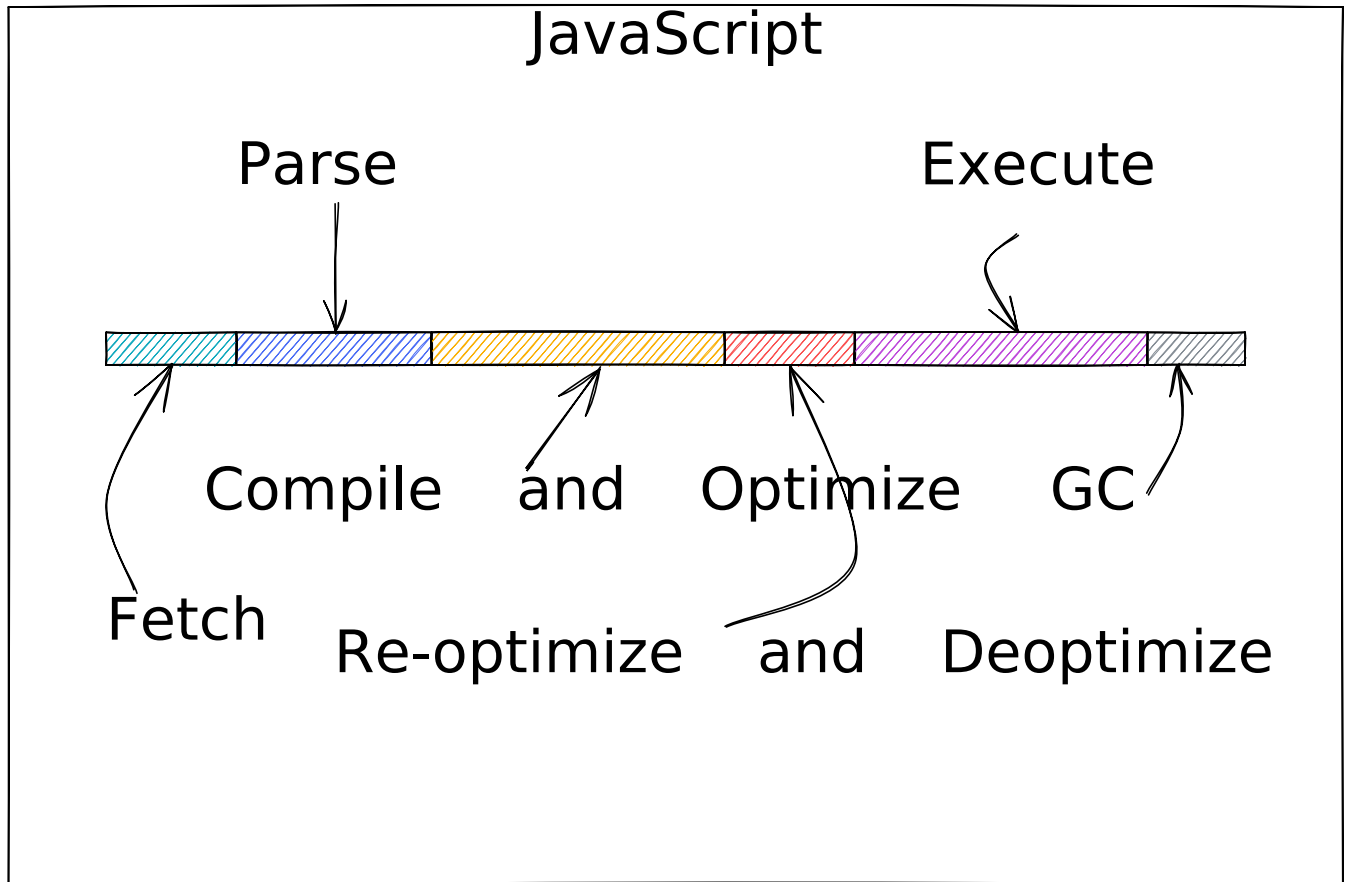
2.7 Ditaa



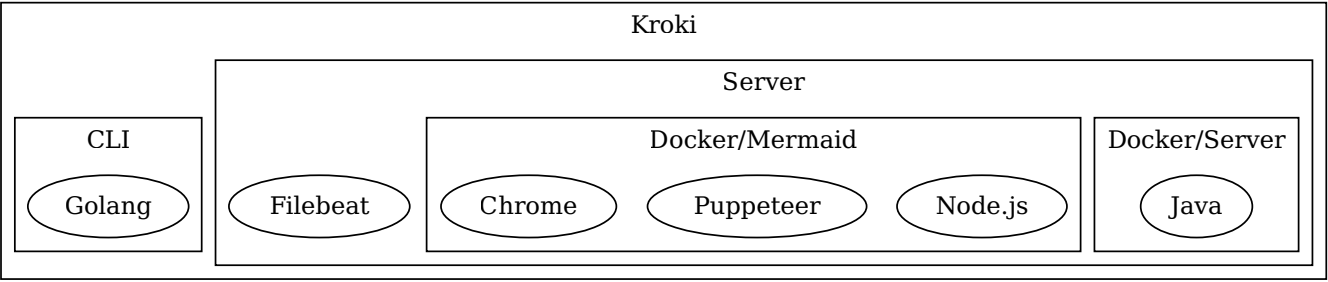
2.8 Erd



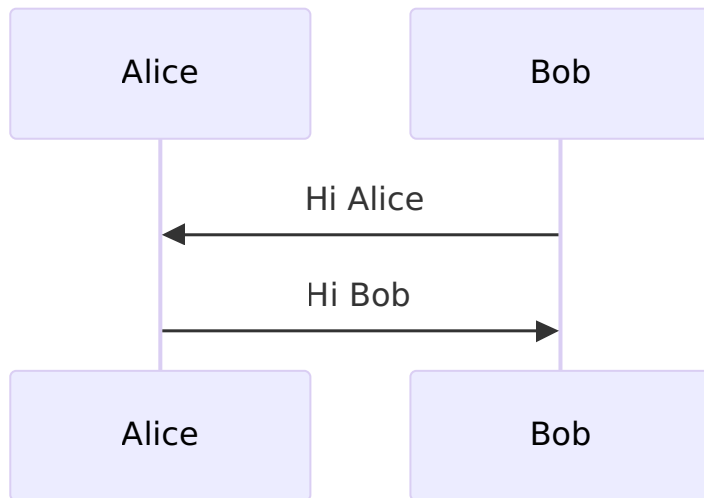
2.9 Excalidraw



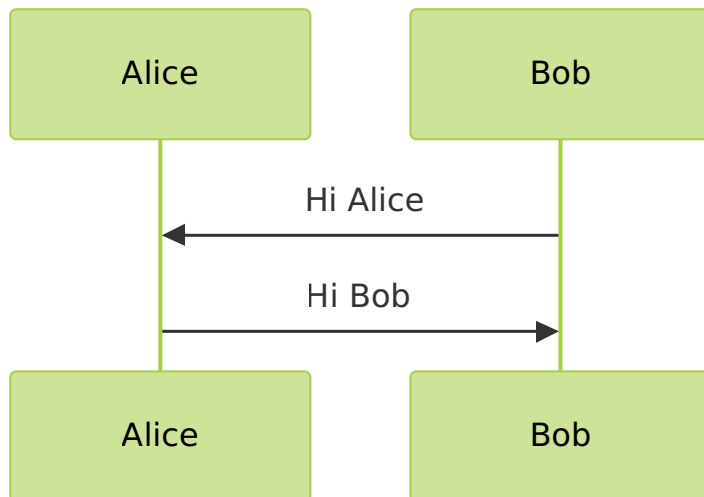
2.10 Graphviz



2.11 Mermaid



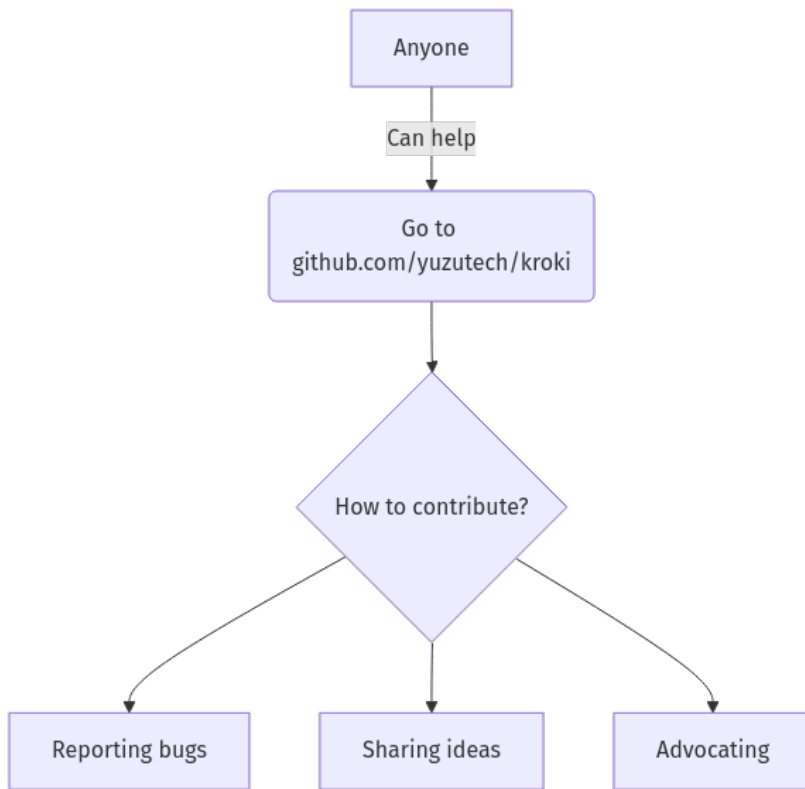
2.11.1 with options



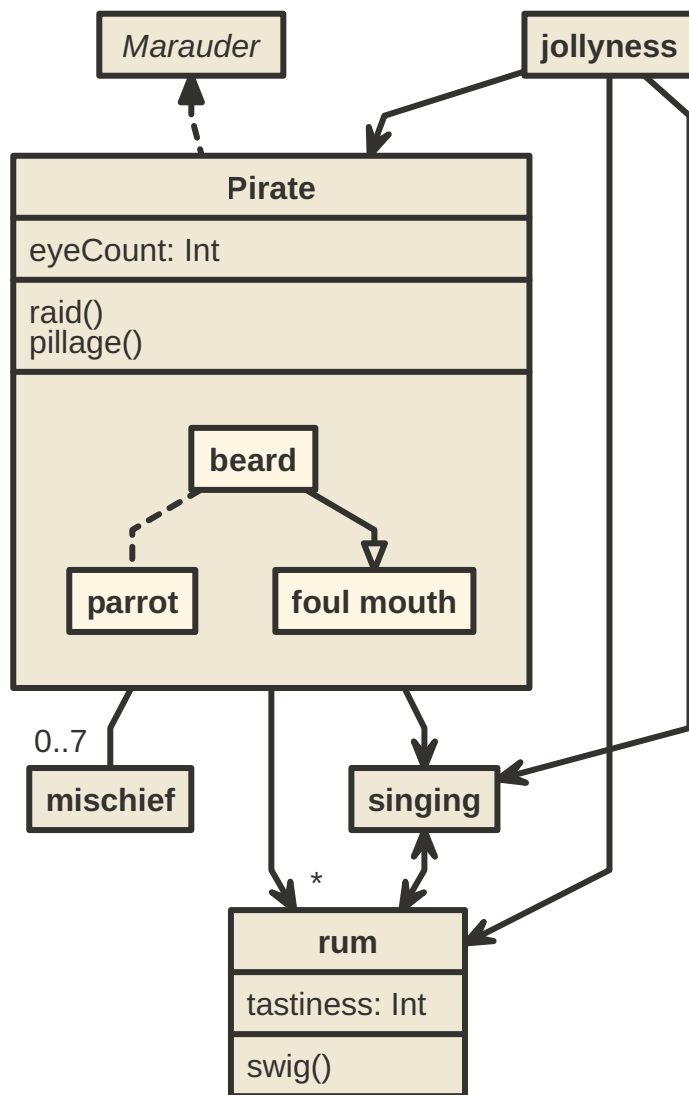
2.11.2 graph syntax

Warning

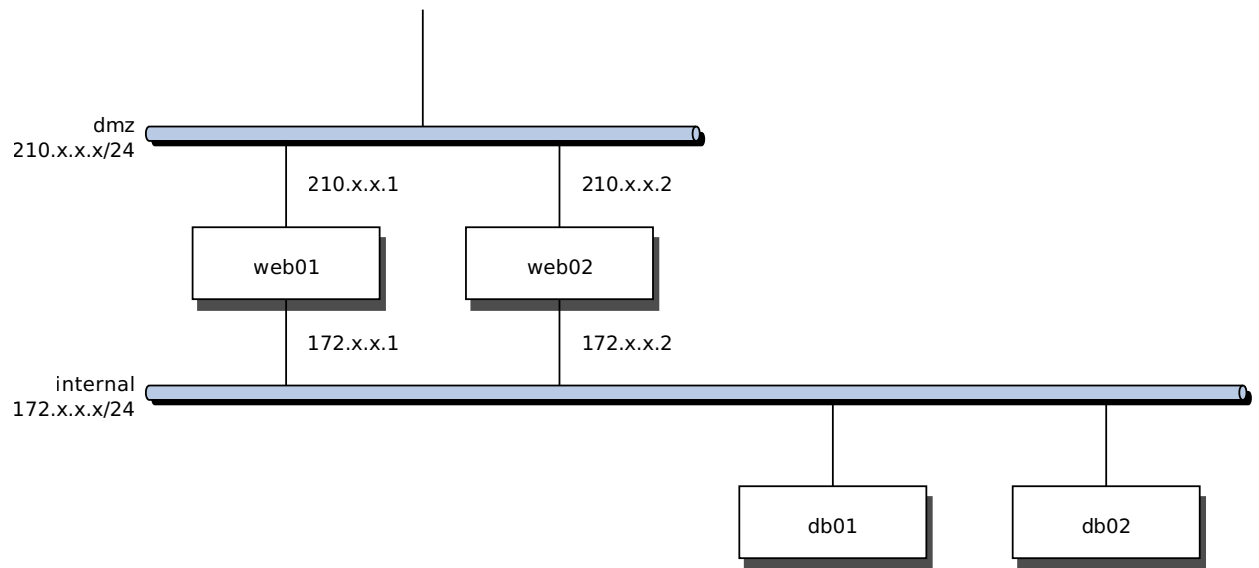
When generating PDF, text within the `graph` syntax doesn't get output in WeasyPrint. However, it does appear when using Playwright/Chromium. As a countermeasure for WeasyPrint limitations, use PNG images.



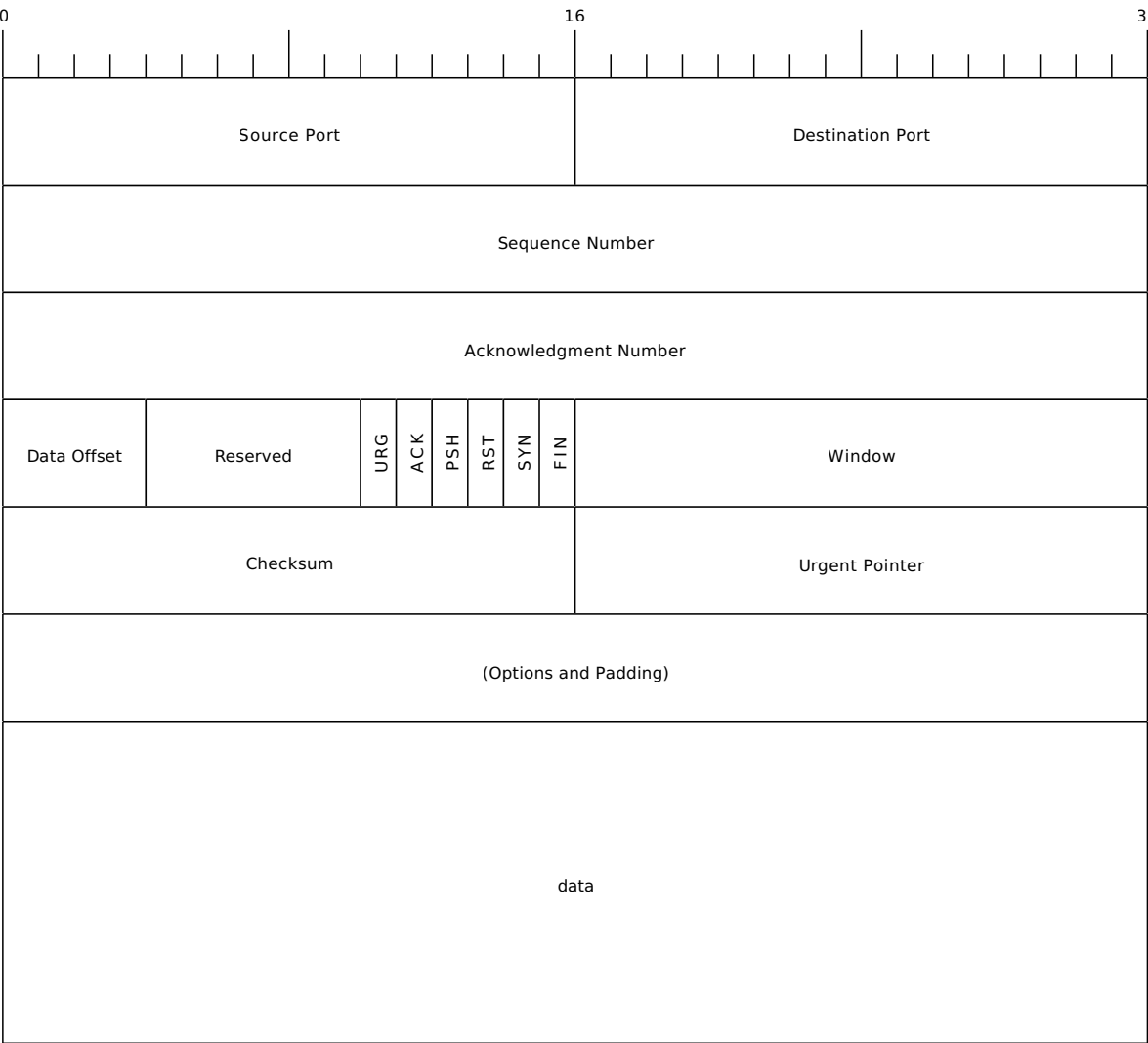
2.12 Nomnoml



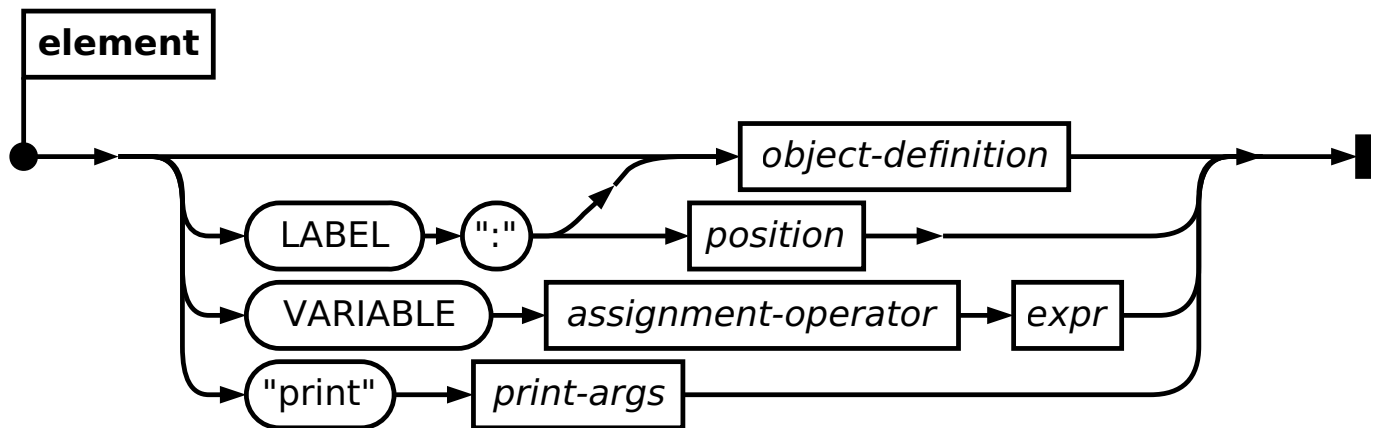
2.13 NwDiag



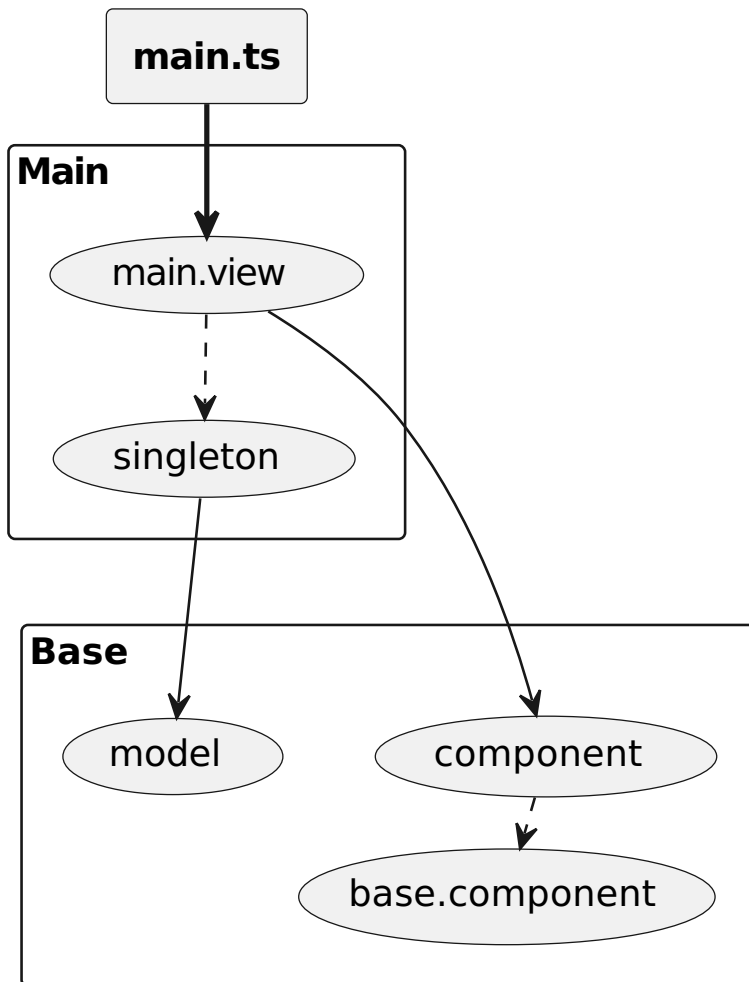
2.14 PacketDoag



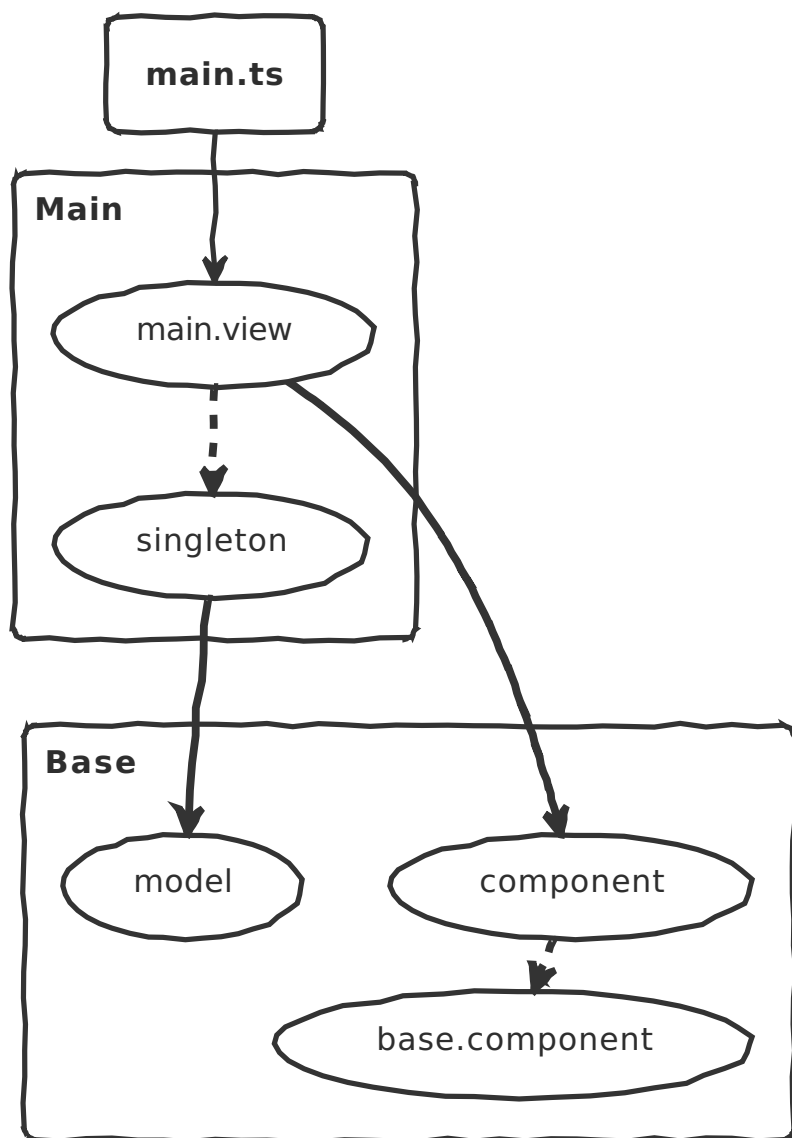
2.15 Pikchr



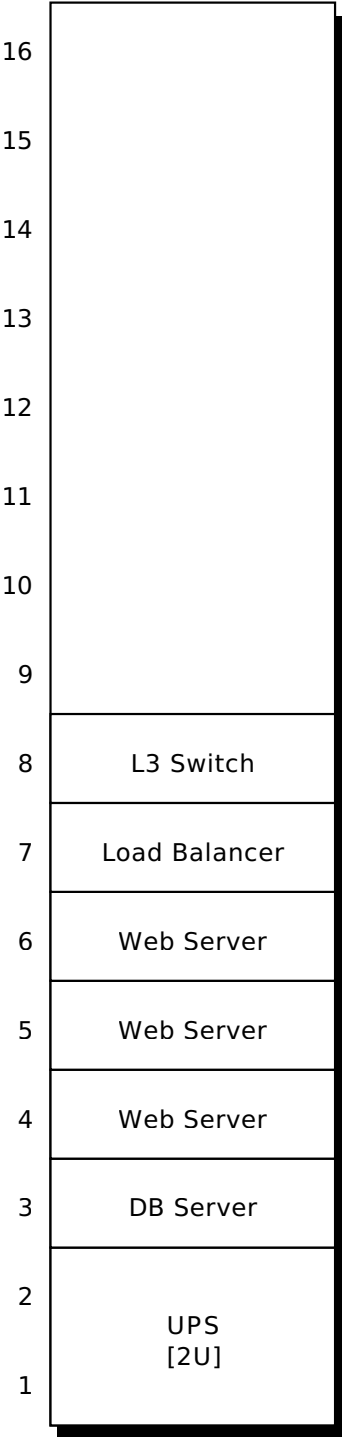
2.16 PlantUML



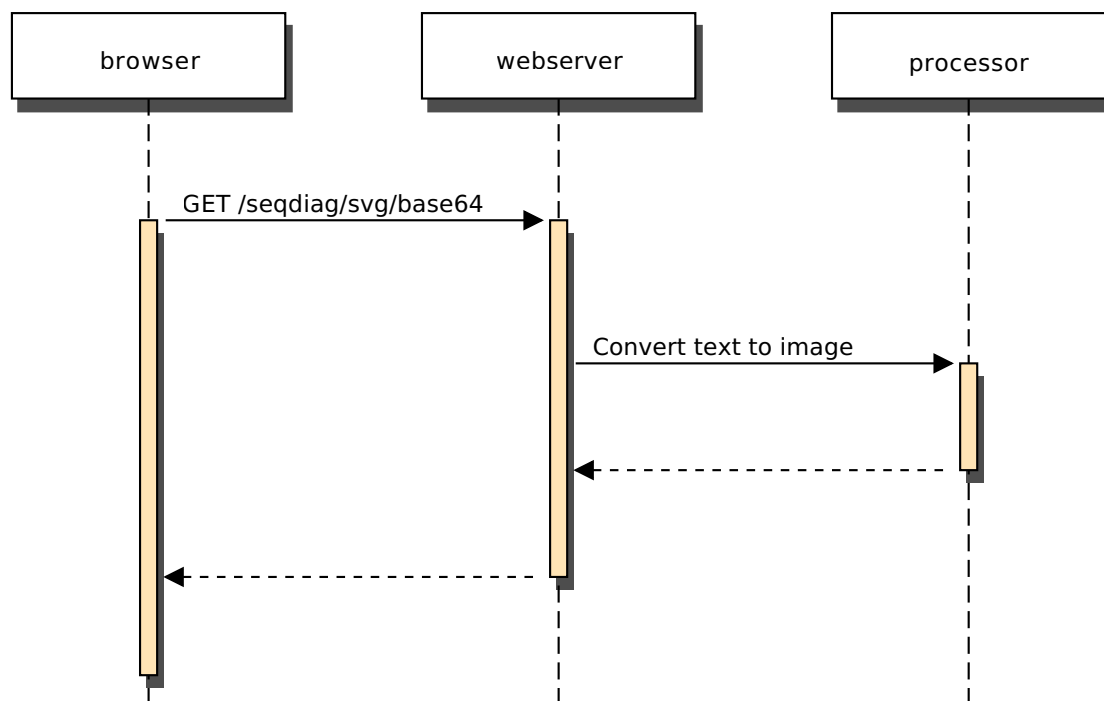
2.16.1 with options



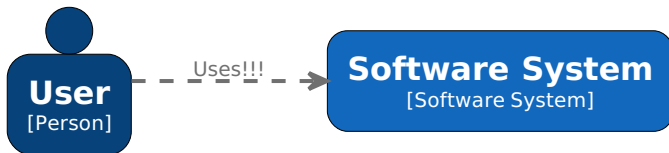
2.17 RackDiag



2.18 SeqDiag



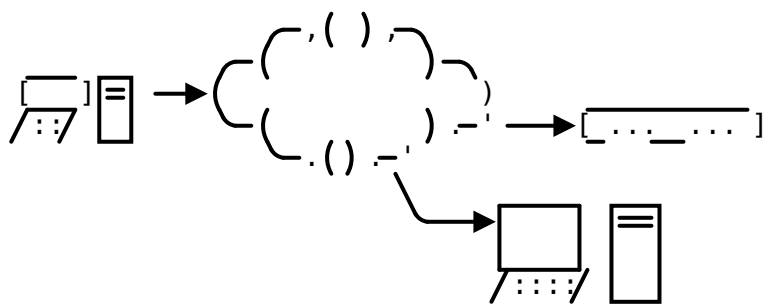
2.19 Structurizr

Software System - System Context

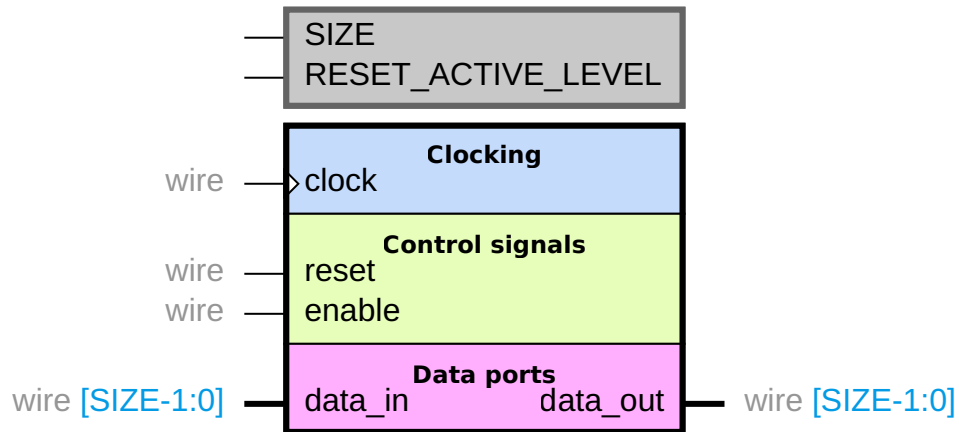
2.19.1 with options



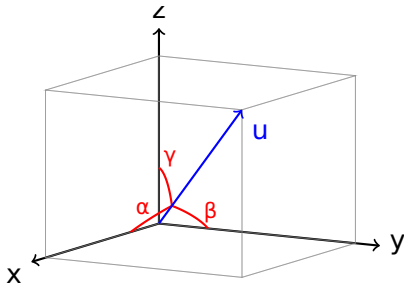
2.20 Svgbob



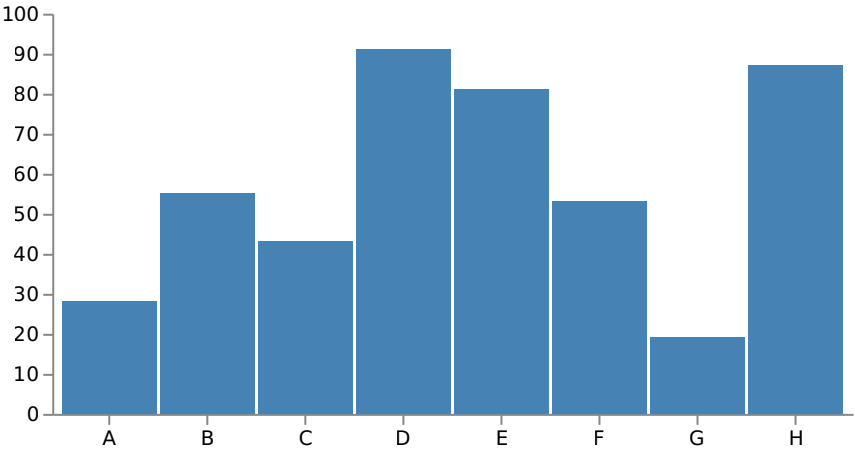
2.21 Symbolator



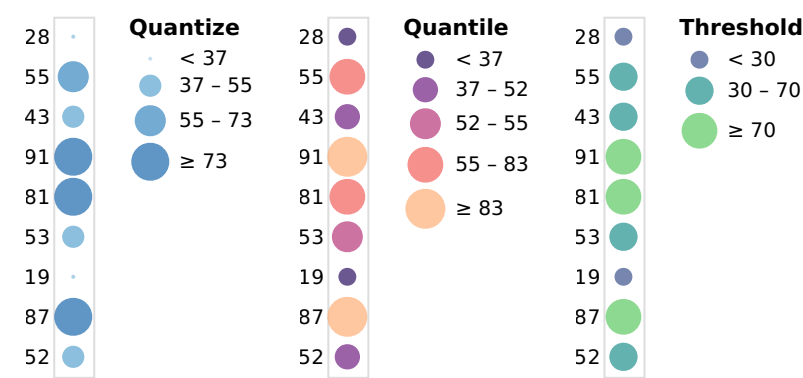
2.22 Tikz



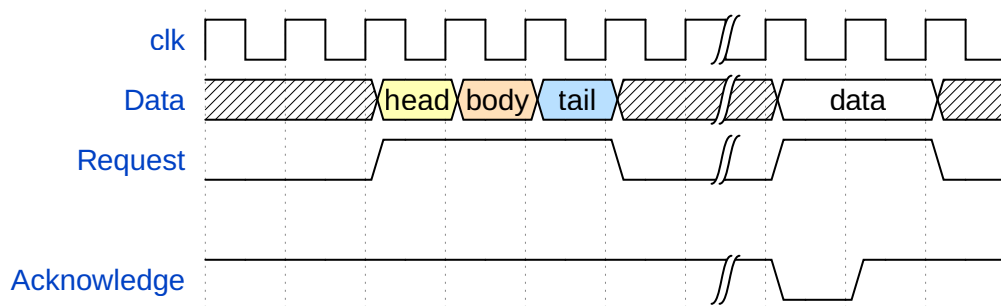
2.23 Vega



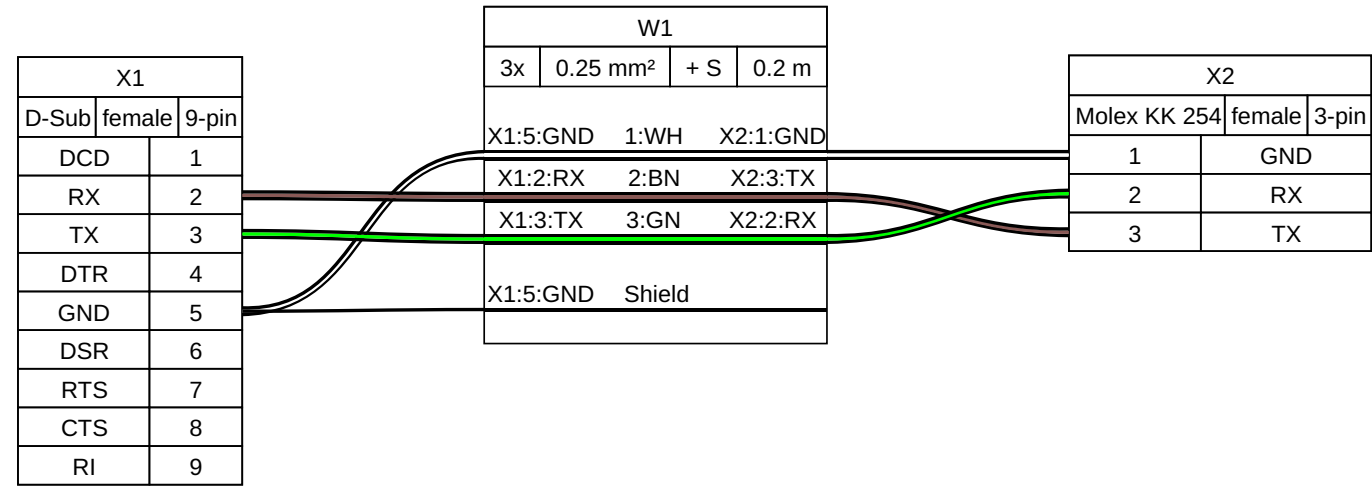
2.24 Vega-Lite



2.25 WaveDrom



2.26 WireViz



3. Experimental

3.1 MkDocs Plugin (Experimental)

Furthermore, when used as an MkDocs plugin, it offers the capability to load diagrams from external files.

3.1.1 Usage

Markdown

```
![[alt text]](diagram file)
```

mkdocs.yml

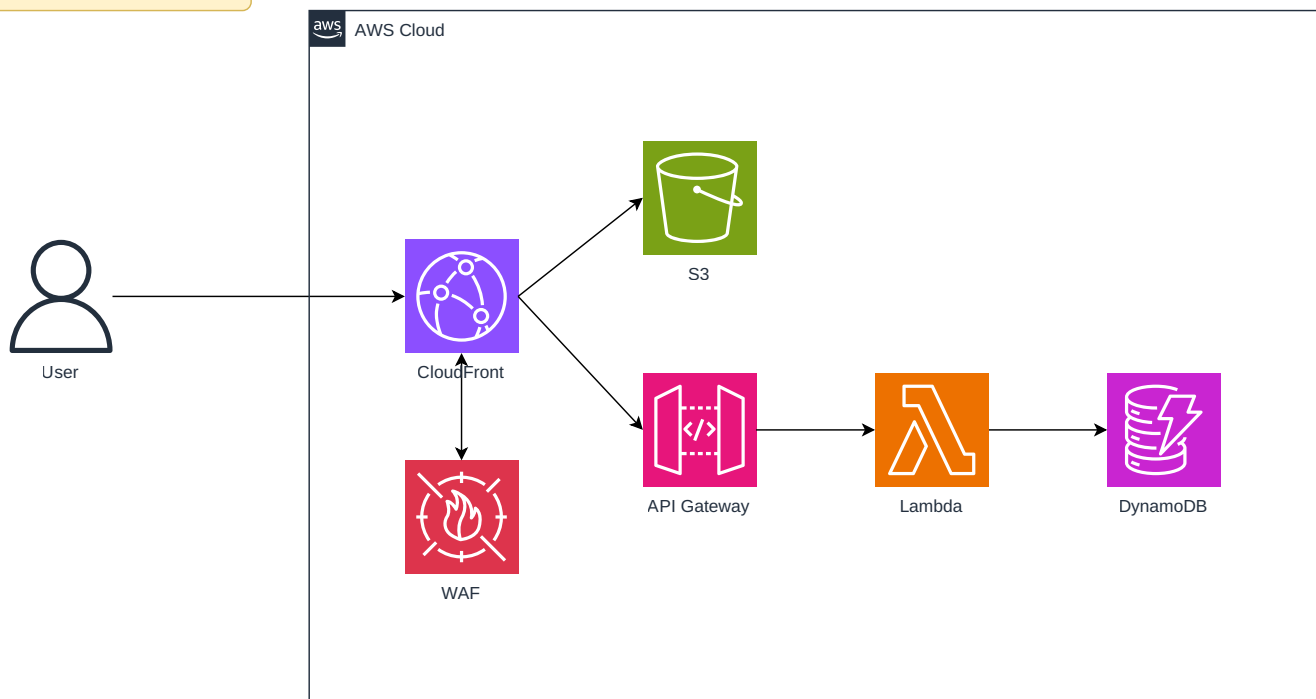
```
plugins:
- kroki_diagrams:
    kroki_url: http://localhost:18000 # default: https://kroki.io
    format: png # default: svg
    img_src: link # default: data
```

3.1.2 Samples

[Markdown code](#)

Draw.io

My Serverless Application



Excaldraw

