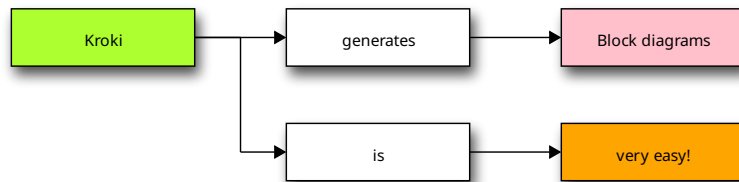
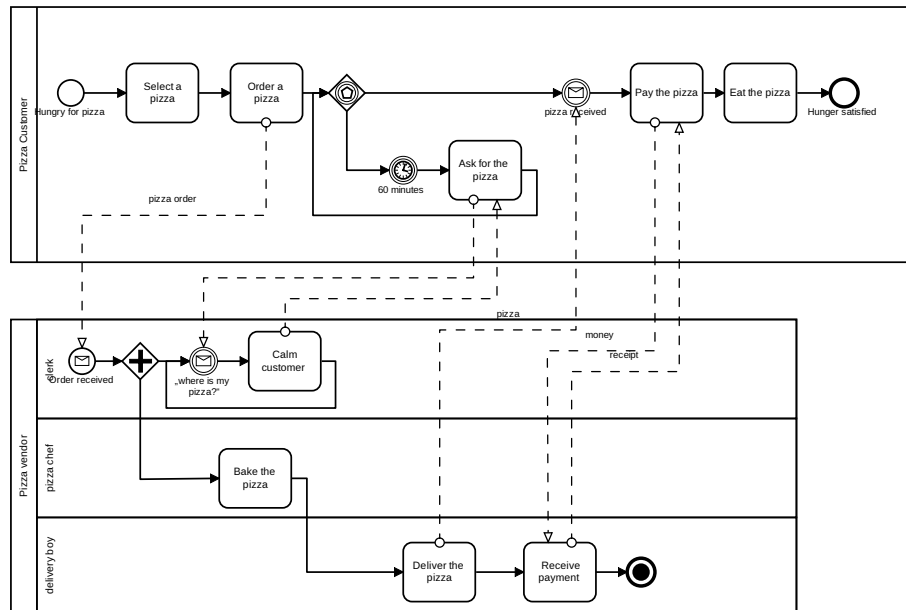


Kroki Support!

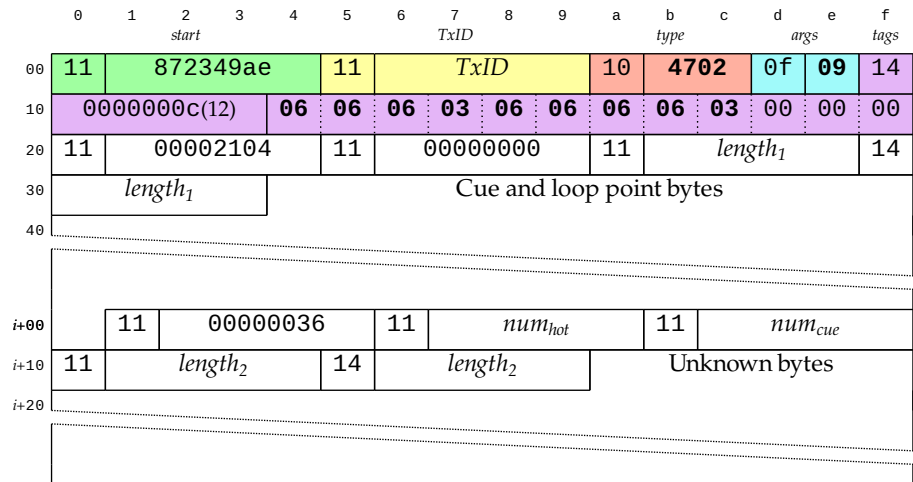
BlockDiag



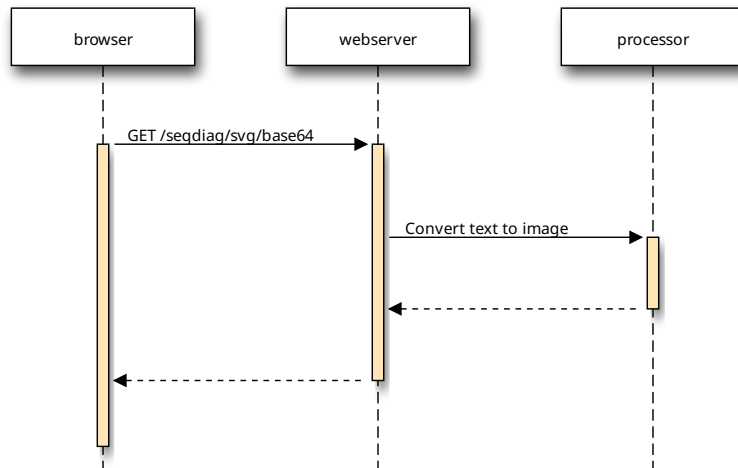
BPMN



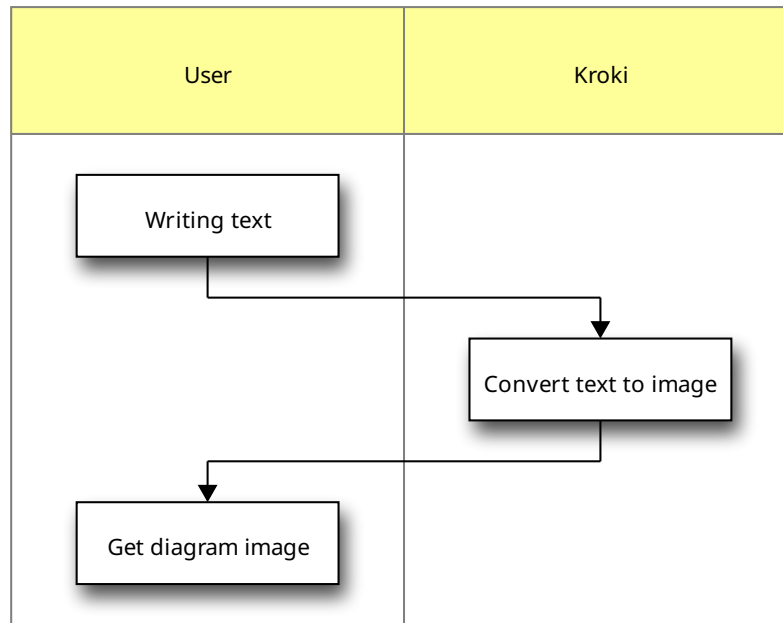
Bytefield



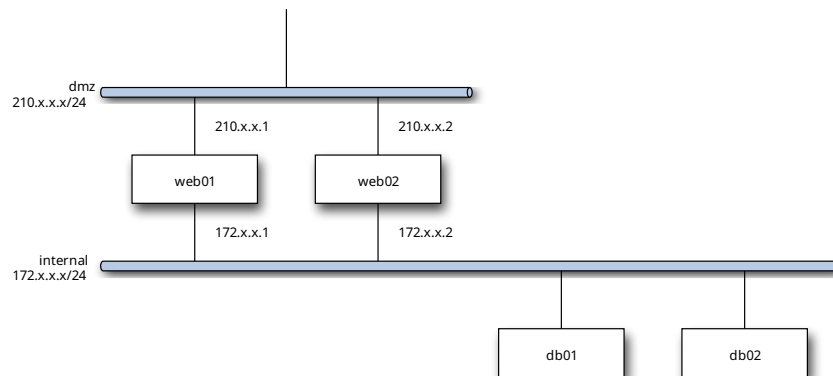
SeqDiag



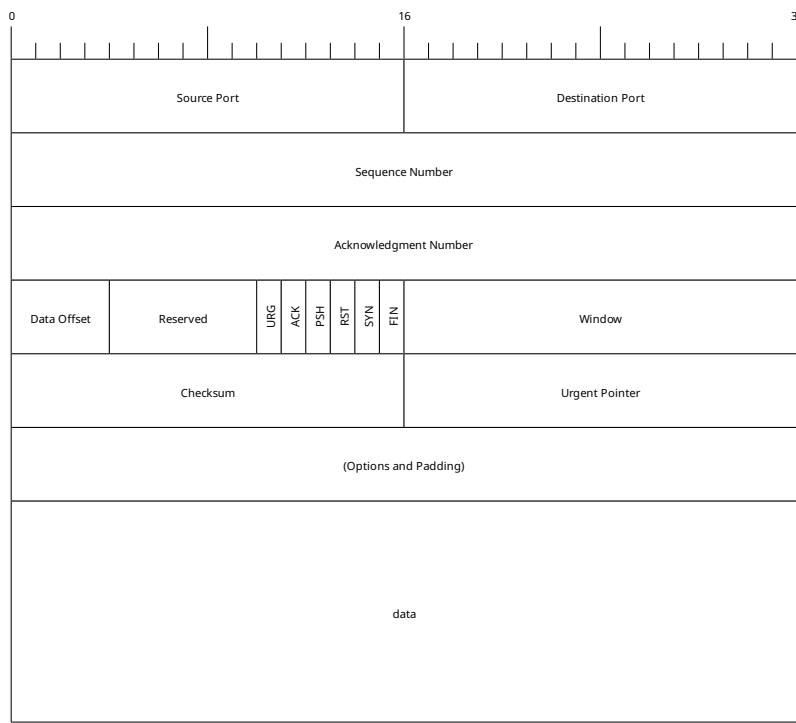
ActDiag



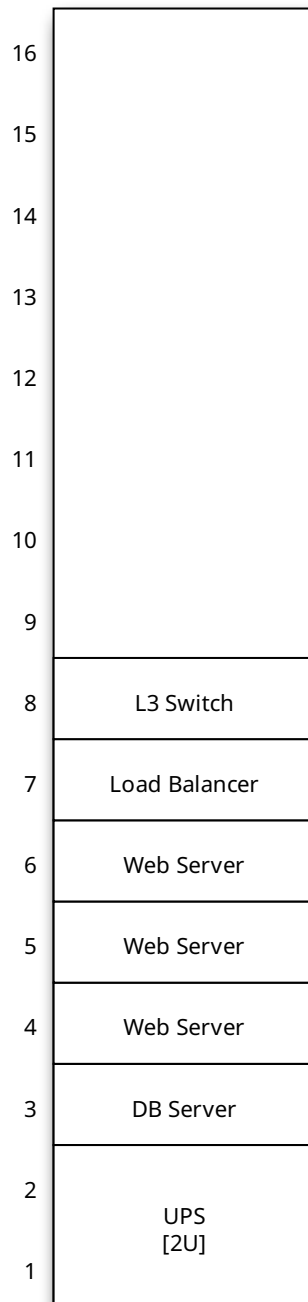
NwDiag



PacketDiag



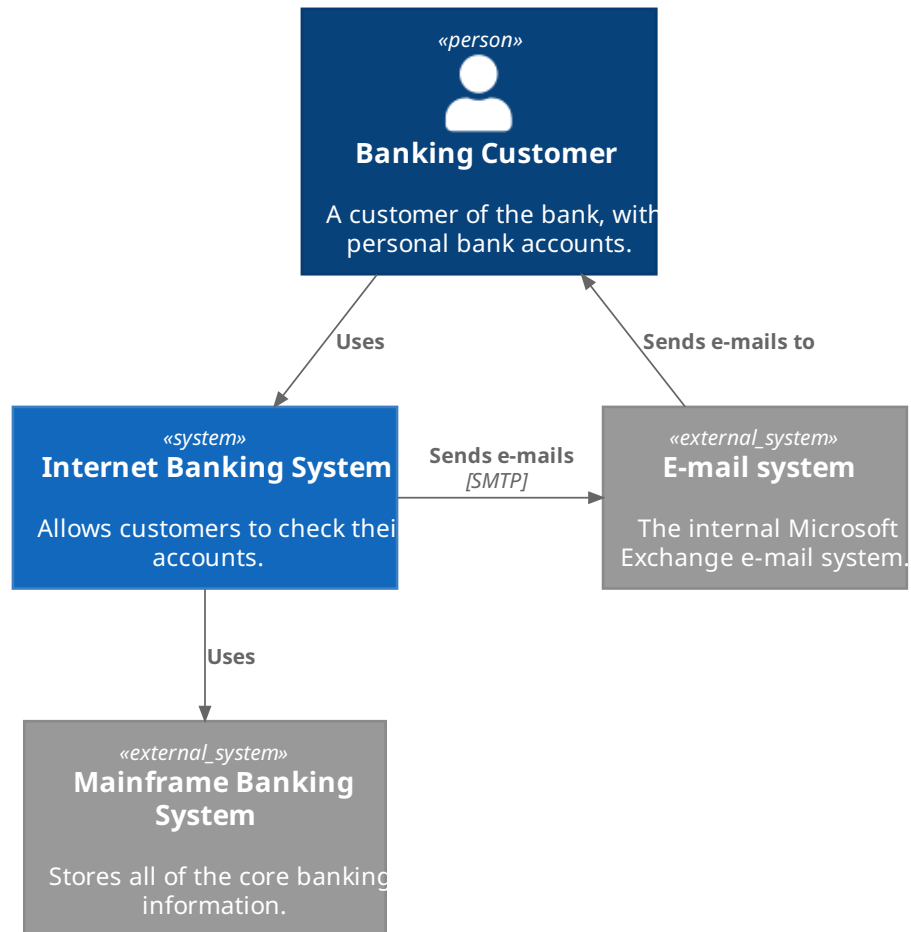
RackDiag



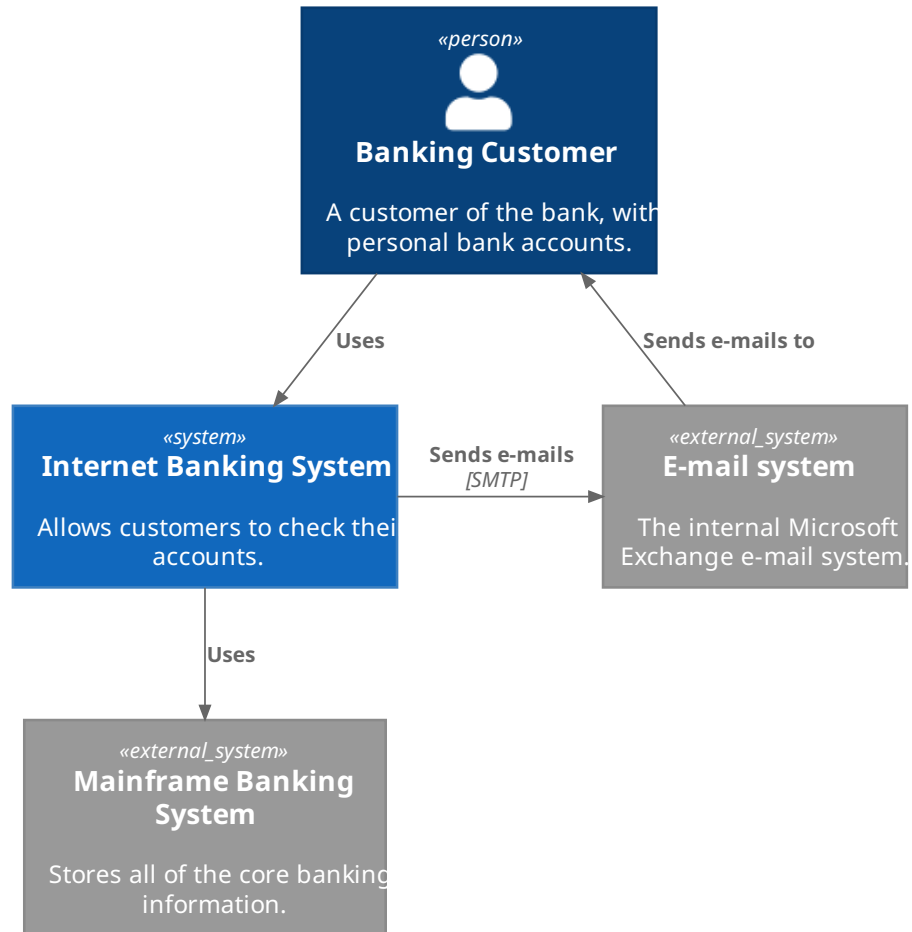
C4 with PlantUML

You can use the proper kroki type `c4plantuml`, or the alias `c4`:

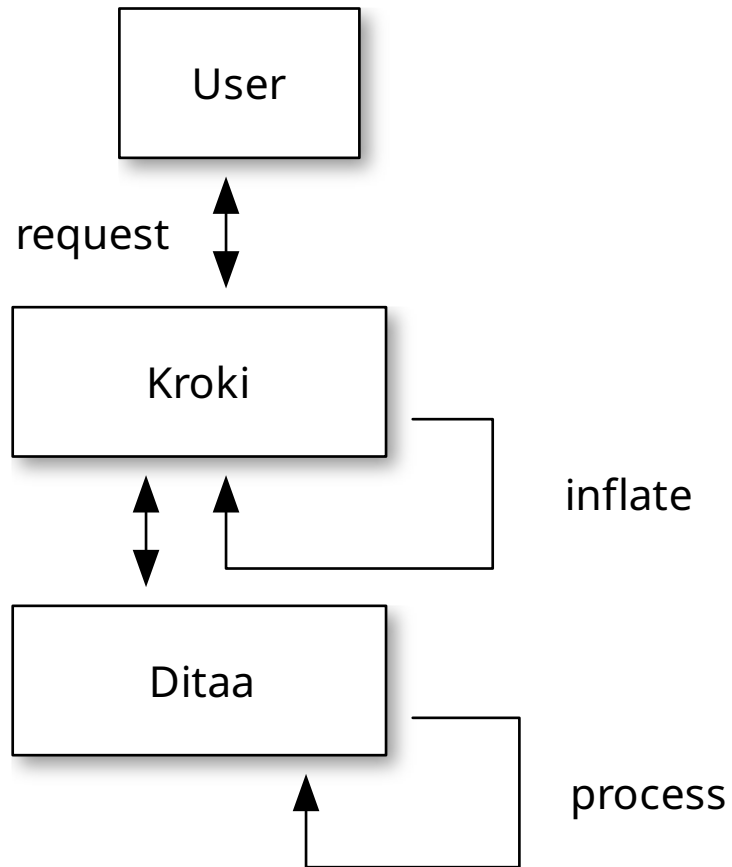
System Context diagram for Internet Banking System



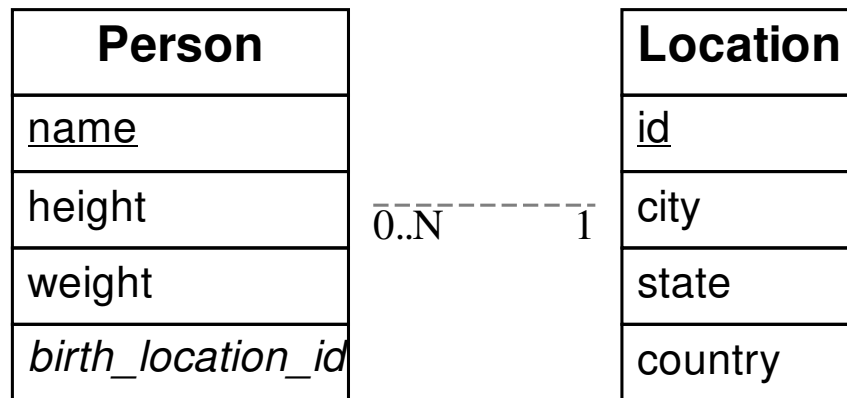
System Context diagram for Internet Banking System



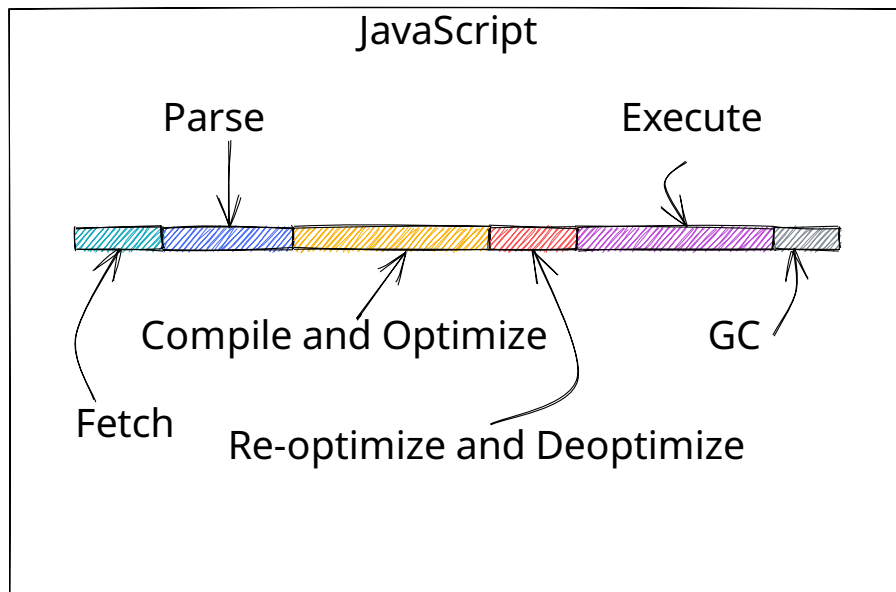
Ditaa



ERD

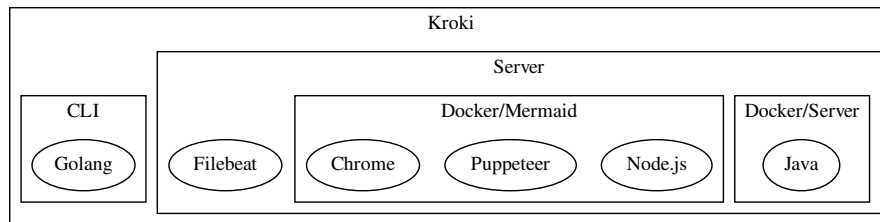
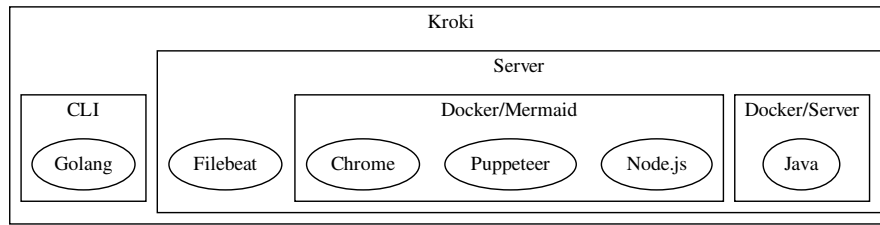


Excalidraw



GraphViz

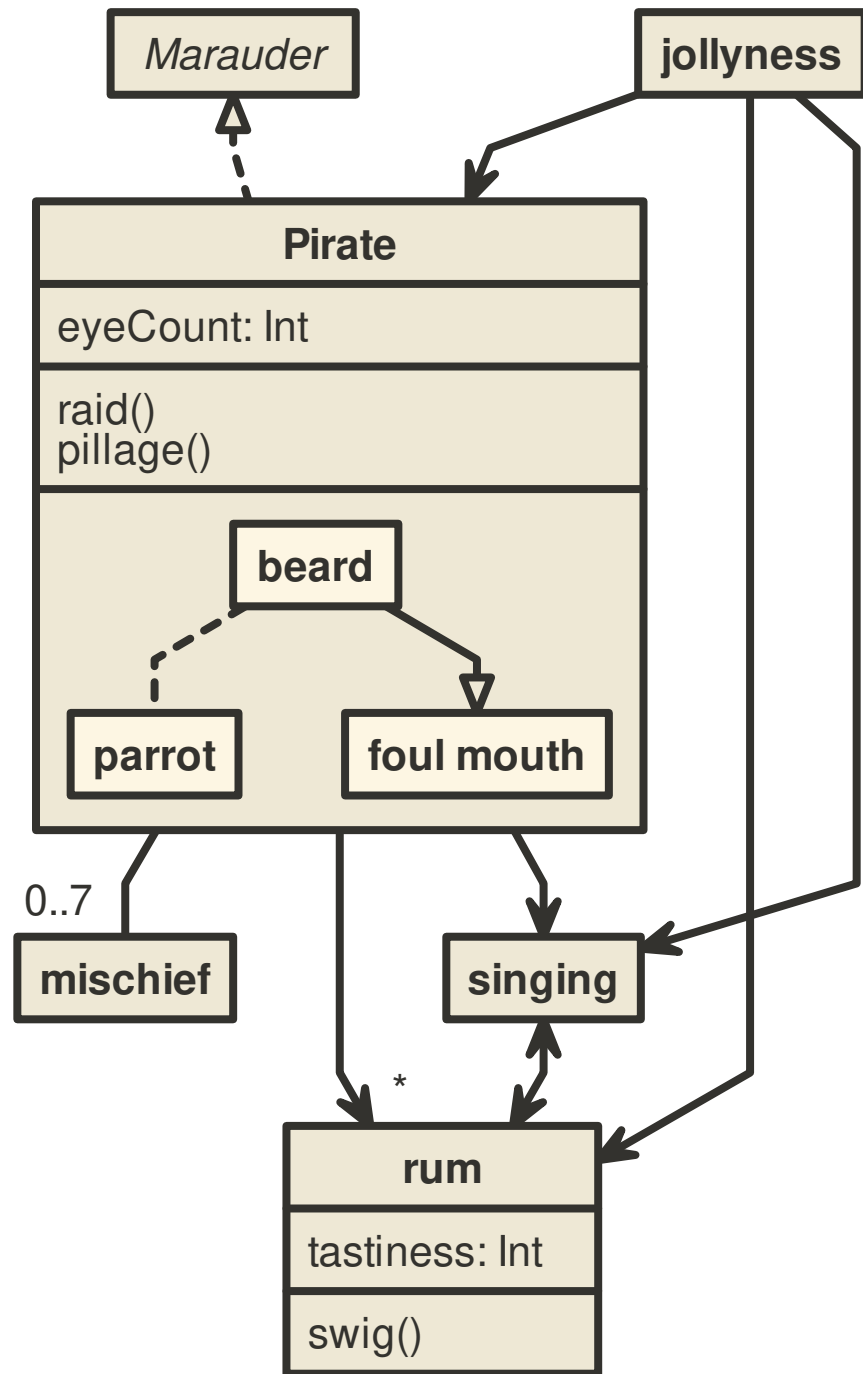
You can use the proper kroki type `graphviz`, or the alias `dot`:



Mermaid

```
graph TD
  A[ Anyone ] -->|Can help | B( Go to github.com/yuzutech/kroki )
  B --> C{ How to contribute? }
  C --> D[ Reporting bugs ]
  C --> E[ Sharing ideas ]
  C --> F[ Advocating ]
```

Nomnoml



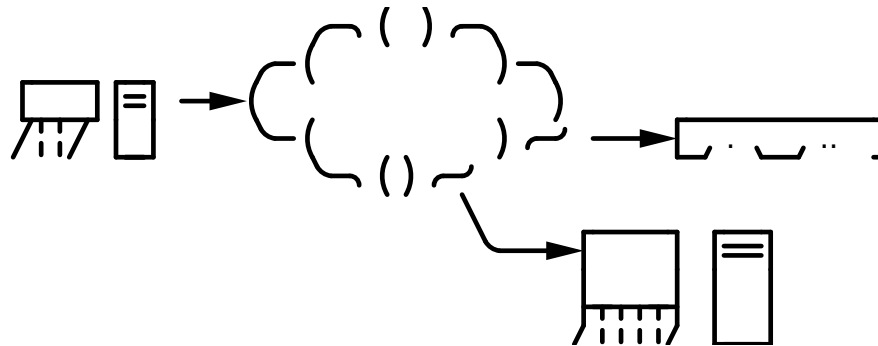
PlantUML

```
skinparam monochrome true
skinparam ranksep 20
skinparam dpi 150
skinparam arrowThickness 0.7
skinparam packageTitleAlignment left
skinparam usecaseBorderThickness 0.4
skinparam defaultFontSize 12
skinparam rectangleBorderThickness 1

rectangle "Main" {
    (main.view)
    (singleton)
}
rectangle "Base" {
    (base.component)
    (component)
    (model)
}
rectangle "<b>main.ts</b>" as main_ts

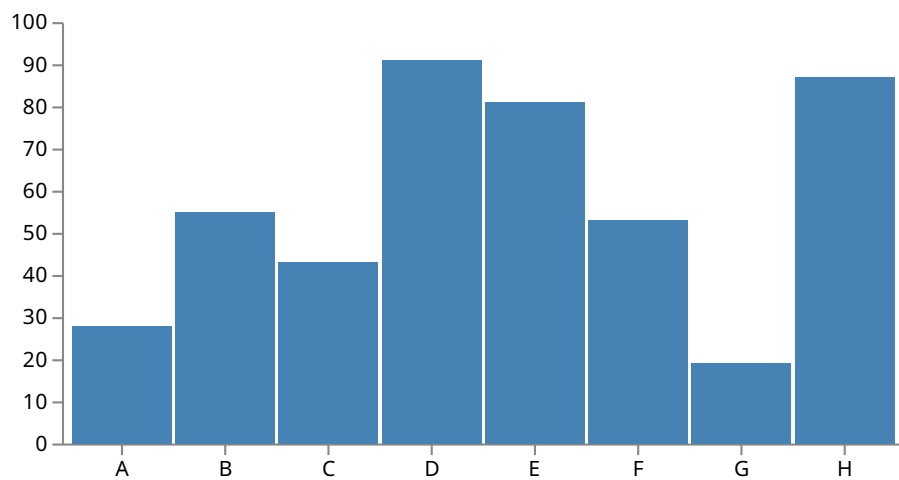
(component) ..> (base.component)
main_ts ==> (main.view)
(main.view) --> (component)
(main.view) ...> (singleton)
(singleton) ---> (model)
```

Svgbob

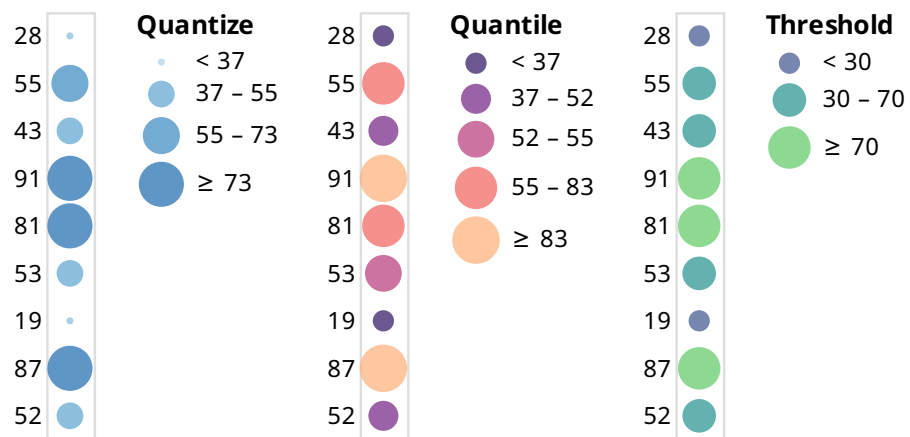


UMlet

Vega



Vega-Lite



WaveDrom

