

plantuml-babel.org

Derek Feichtinger

October 18, 2019

Contents

1	Links	1
2	Information on the local installation	2
2.1	Help text	3
3	simple test	3
4	Diagram type examples	3
4.1	sequence diagrams	3
4.2	old style activity diagrams	5
4.3	new style activity diagrams	7
4.3.1	swimlanes	7
4.4	Class diagrams	7
4.5	Component diagrams	8
4.6	Mindmaps	10
5	skinparam	14
5.1	Gradients	14
6	Scaling	15
7	TODO using SVG graphics	16

1 Links

- Homepage: <http://plantuml.com/>
- Downloads: <http://plantuml.sourceforge.net/>

- Source code: <https://github.com/plantuml/plantuml>
- Language Reference: http://plantuml.com/PlantUML_Language_Reference_Guide.pdf

2 Information on the local installation

Emacs version: GNU Emacs 26.2 (build 2, x86_64-pc-linux-gnu, GTK+ Version 3.22.30)
 of 2019-04-14
 org version: 9.2.5

Emacs variable org-plantuml-jar-path:/home/dfeich/.emacs.d/javalib/plantuml.jar

PlantUML version 1.2019.08 (Sat Jul 13 21:25:14 CEST 2019)
 (GPL source distribution)
 Java Runtime: OpenJDK Runtime Environment
 JVM: OpenJDK 64-Bit Server VM
 Java Version: 1.8.0_222-8u222-b10-1ubuntu1~18.04.1-b10
 Operating System: Linux
 OS Version: 4.15.0-65-generic
 Default Encoding: UTF-8
 Language: en
 Country: US
 Machine: dflt2w
 PLANTUML_LIMIT_SIZE: 4096
 Processors: 4
 Max Memory: 4,649,385,984
 Total Memory: 313,524,224
 Free Memory: 306,981,080
 Used Memory: 6,543,144
 Thread Active Count: 1

The environment variable GRAPHVIZ_DOT has not been set
 Dot executable is /opt/anaconda/python3.6/bin/dot
 Dot version: dot - graphviz version 2.40.1 (20161225.0304)
 Installation seems OK. File generation OK

 PlantUML version 1.2019.08 (Sat Jul 13 21:25:14 CEST 2019)

Last available version for download : 1201911

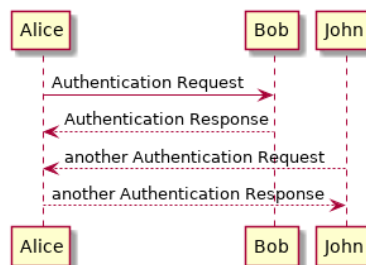
A newer version is available for download.

2.1 Help text

```
java -jar "$jpath" -help
```

3 simple test

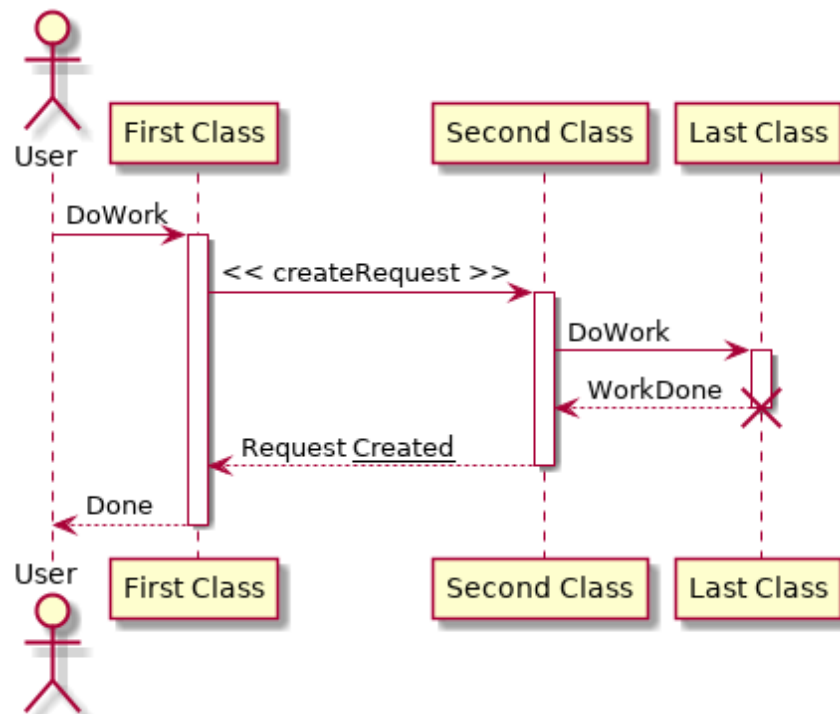
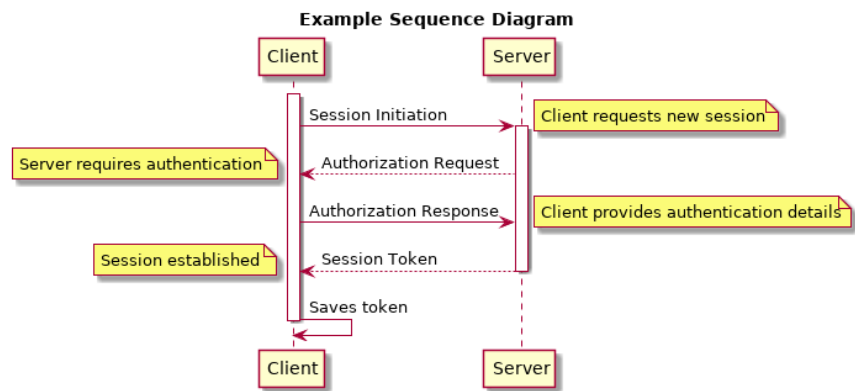
```
@startuml
' this is a comment
Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response
John --> Alice: another Authentication Request
Alice --> John: another Authentication Response
@enduml
```

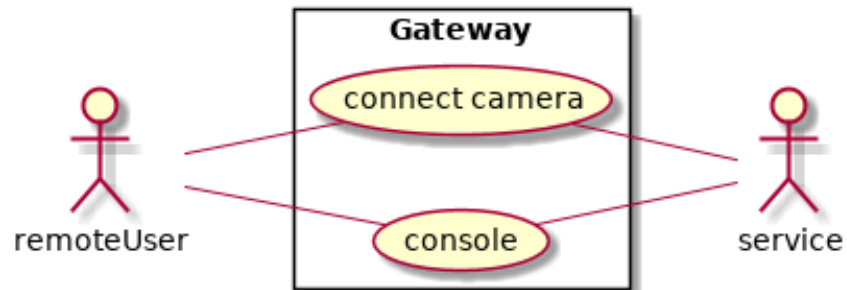


4 Diagram type examples

4.1 sequence diagrams

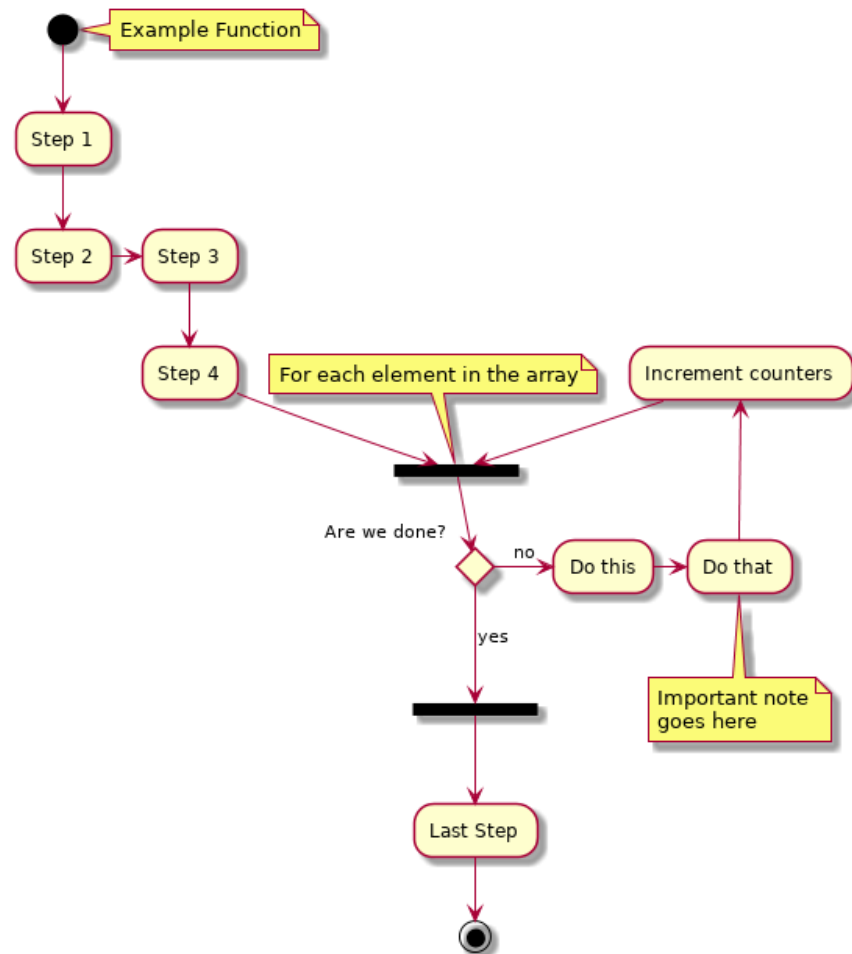
Note: The `skin` parameter I used in the earlier versions of this document is no longer supported by plantuml.



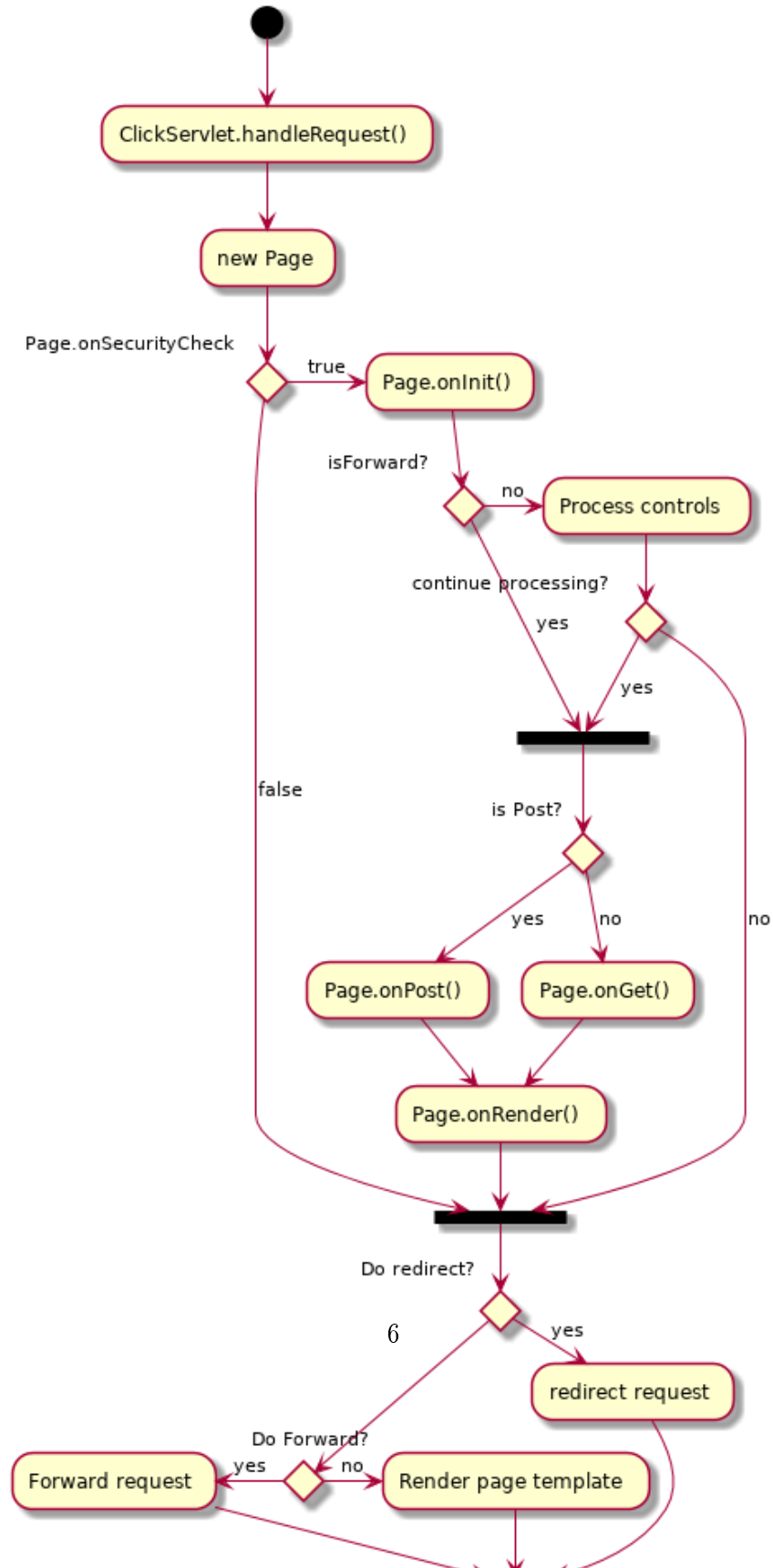


4.2 old style activity diagrams

Example Activity Diagram



Servlet Container

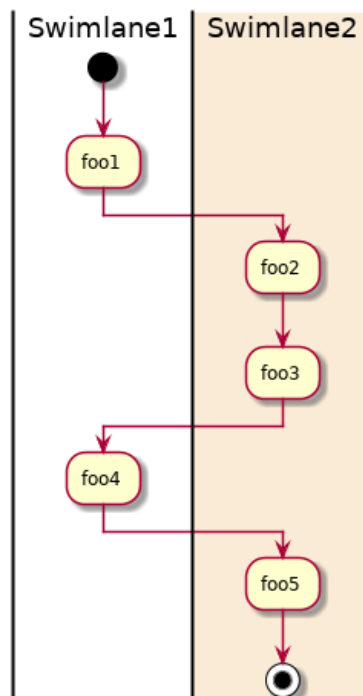


4.3 new style activity diagrams

- <http://plantuml.sourceforge.net/activity2.html>

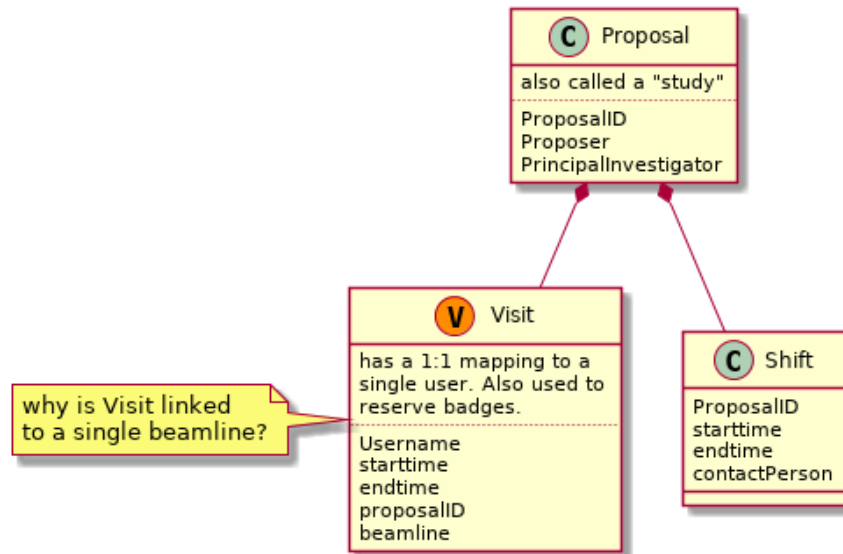
4.3.1 swimlanes

Swimlanes actually are activity diagrams using the new syntax.



4.4 Class diagrams

<http://plantuml.sourceforge.net/classes.html>



4.5 Component diagrams

@startuml

```

package "Some Group" {
    HTTP - [First Component]
    [Another Component]
}
  
```

```

node "Other Groups" {
    FTP - [Second Component]
    [First Component] --> FTP
}
  
```

```

cloud {
    [Example 1]
}
  
```

```

database "MySQL" {
    folder "This is my folder" {
        [Folder 3]
    }
}
  
```



```

frame "Foo" {
  [Frame 4]
}
}

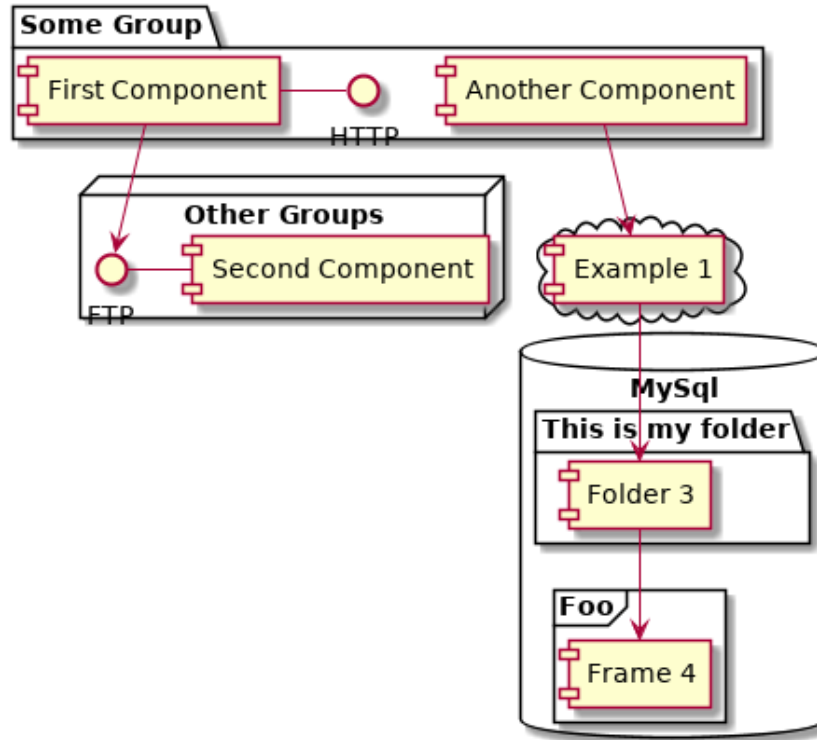
```

```

[Another Component] --> [Example 1]
[Example 1] --> [Folder 3]
[Folder 3] --> [Frame 4]

```

@enduml



Example by Cecil Westerhof on emacs-orgmode.gnu.org mailing list 2019-10-18 Fri

@startuml

```

component [Producer 1\nProducer 2\nProducer ...\nProducer n] as Producers

```

```

cloud {
    [Internet] as Internet1
}

node RabbitMQ #LightSteelBlue {
    [Exchange]
    [Queue 1\nQueue 2\nQueue ... \nQueue n] as Queues
}

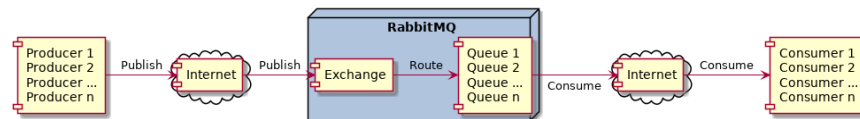
cloud {
    [Internet] as Internet2
}

[Consumer 1\nConsumer 2\nConsumer ... \nConsumer n] as Consumers

[Producers] -> [Internet1] : Publish
[Internet1] -> [Exchange] : Publish
[Exchange] -> [Queues] : Route
[Queues] -> [Internet2] : Consume
[Internet2] -> [Consumers] : Consume

@enduml

```



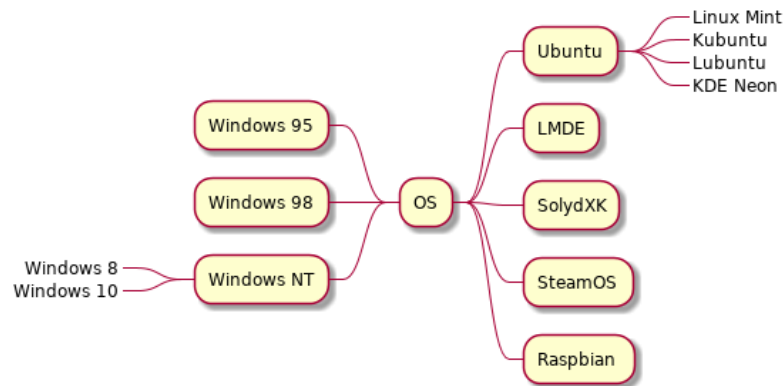
4.6 Mindmaps

- 2019-07-21 Sun Needs plantuml-1.2019.08 or newer. Still in testing and features may change
- <http://plantuml.com/mindmap-diagram>
- Nice Link about mindmaps in PlantUML: <http://hangaroundtheweb.com/2019/07/mind-maps-in-spacemacs/>

The examples are taken from the official plantuml page.
This syntax looks like the most versatile and useful to me

- Leading "+/-" specify hierarchy level and whether the node is on the right or left of the central node.
- Underscores directly following the leading position characters prevent the creation of a box around an item.

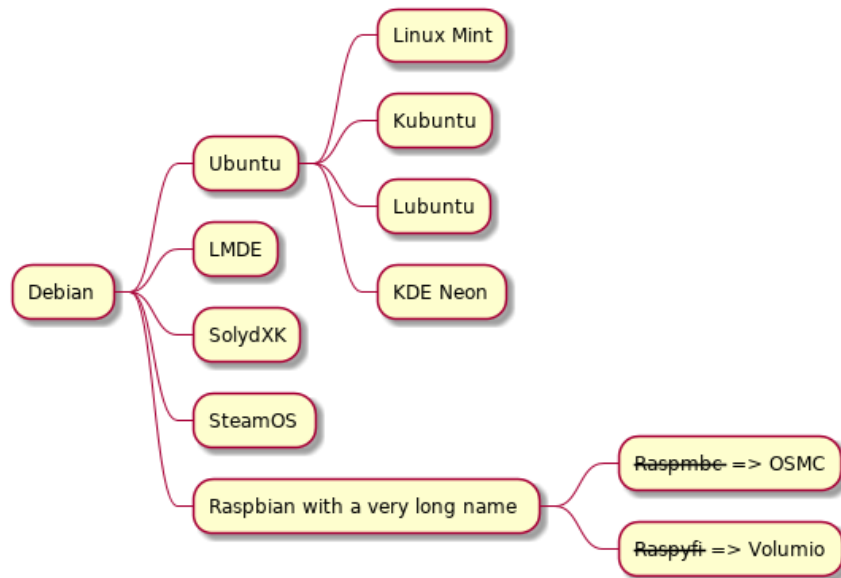
```
@startmindmap
+ OS
++ Ubuntu
+++_ Linux Mint
+++_ Kubuntu
+++_ Lubuntu
+++_ KDE Neon
++ LMDE
++ SolydXK
++ SteamOS
++ Raspbian
-- Windows 95
-- Windows 98
-- Windows NT
---_ Windows 8
---_ Windows 10
@endmindmap
```



A mindmap based on org mode syntax. Note that the org headline asterisks need to be escaped by "\", inside of a source block. It's nice that they allow for an org mode syntax, but I think this is less convenient to

write and work with. The org headlines do not allow for text following them (syntax error).

```
@startmindmap
* Debian
** Ubuntu
*** Linux Mint
*** Kubuntu
*** Lubuntu
*** KDE Neon
** LMDE
** SolydXX
** SteamOS
** Raspbian with a very long name
*** <s>Raspmbc</s> => OSMC
*** <s>Raspyfi</s> => Volumio
@endmindmap
```



```
@startmindmap
caption figure 1
title My super title
```

```

* <&flag>Debian
** <&globe>Ubuntu
*** Linux Mint
*** Kubuntu
*** Lubuntu
*** KDE Neon
** <&graph>LMDE
** <&pulse>SolydXX
** <&people>SteamOS
** <&star>Raspbian with a very long name
*** <s>Raspmbc</s> => OSMC
*** <s>Raspyfi</s> => Volumio

```

```

header
My super header
endheader

```

```

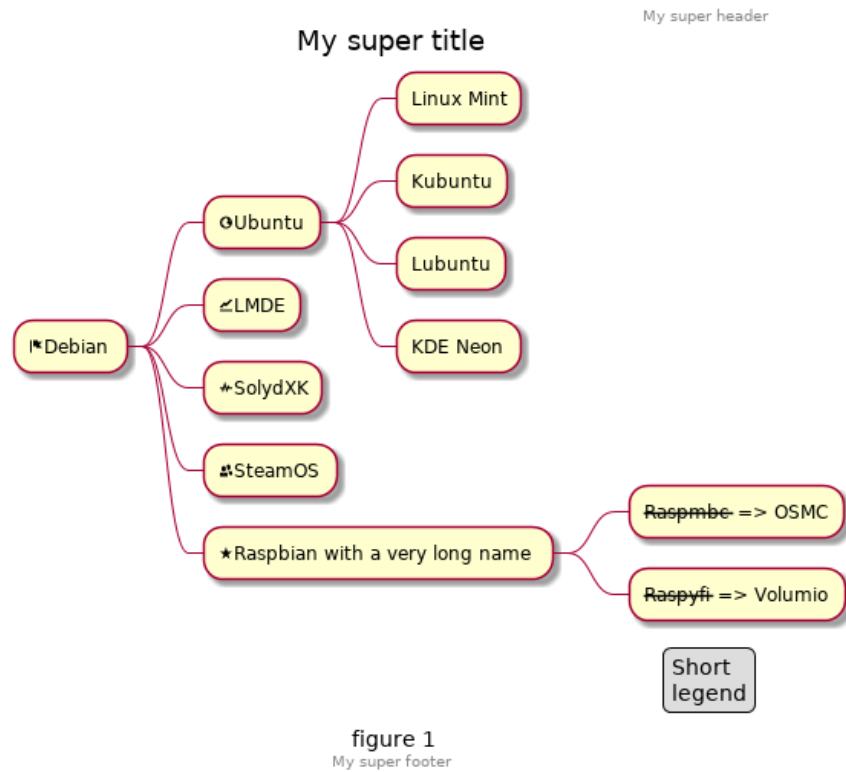
center footer My super footer

```

```

legend right
  Short
  legend
endlegend
@endmindmap

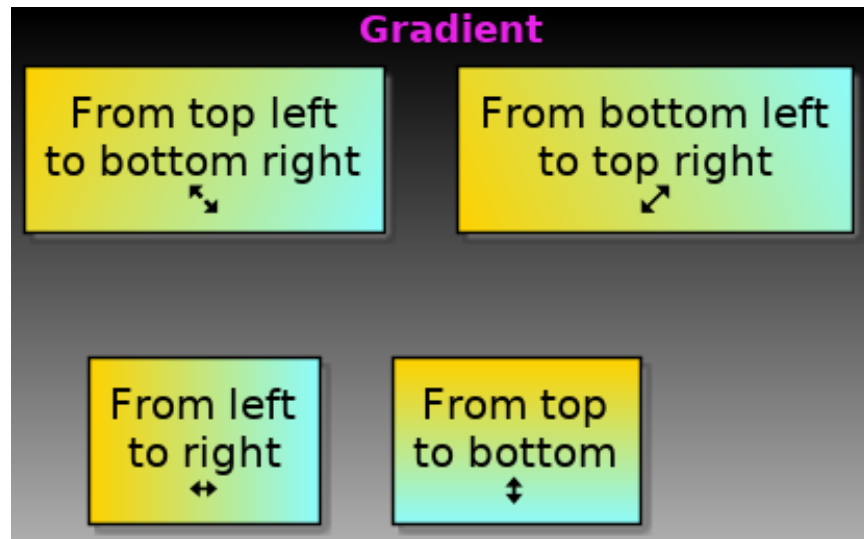
```



5 skinparam

5.1 Gradients

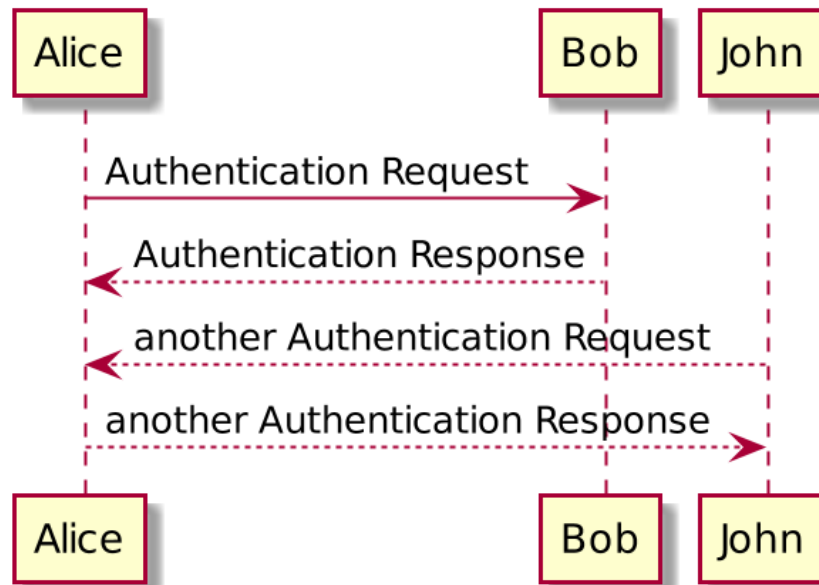
Minimally adapted from <https://blog.jdriven.com/2017/10/plantuml-pleasantness-use-gradients>



6 Scaling

```
@startuml
scale 2
```

```
Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response
John --> Alice: another Authentication Request
Alice --> John: another Authentication Response
@enduml
```



7 TODO using SVG graphics

The *svg* package uses inkscape to separate the text and graphical elements of the SVG into a Tex file (*.pdf_{tex}) and a PDF file containig the graph elements. E.g. `svg-sequence1.svg` into `svg-sequence1.pdftex` and `svg-sequence1.pdf`.

Currently, SVG pictures can only be rendered correctly, **if the picture is in the same directory** as the tex source file (and therefore also the org source file).

Note: with the current org version 9.1.14 and Emacs 26.1 the SVG is not correctly displayed in the org buffer, but the SVG renders fine in the exported Latex PDF.

```

@startuml
' this is a comment
Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response
John --> Alice: another Authentication Request
Alice --> John: another Authentication Response
@enduml
  
```