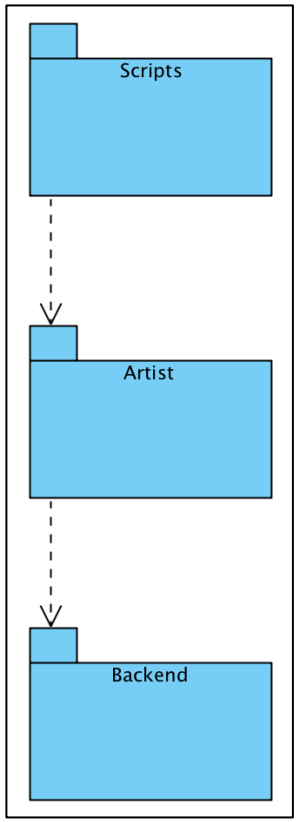
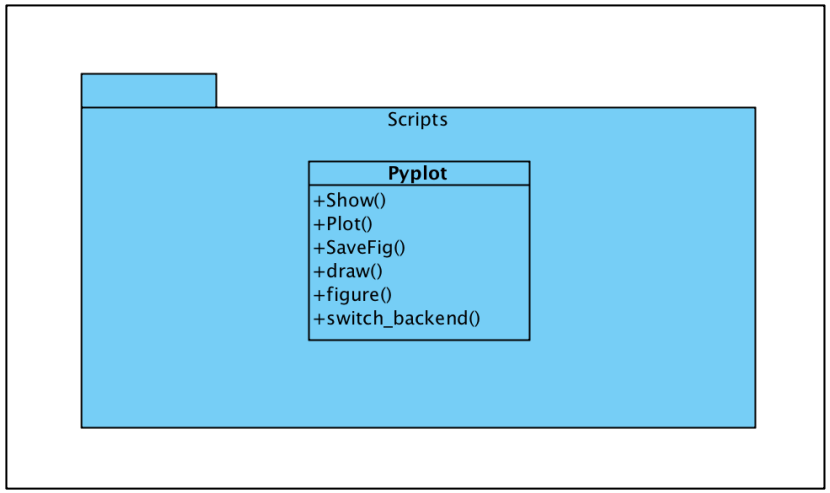
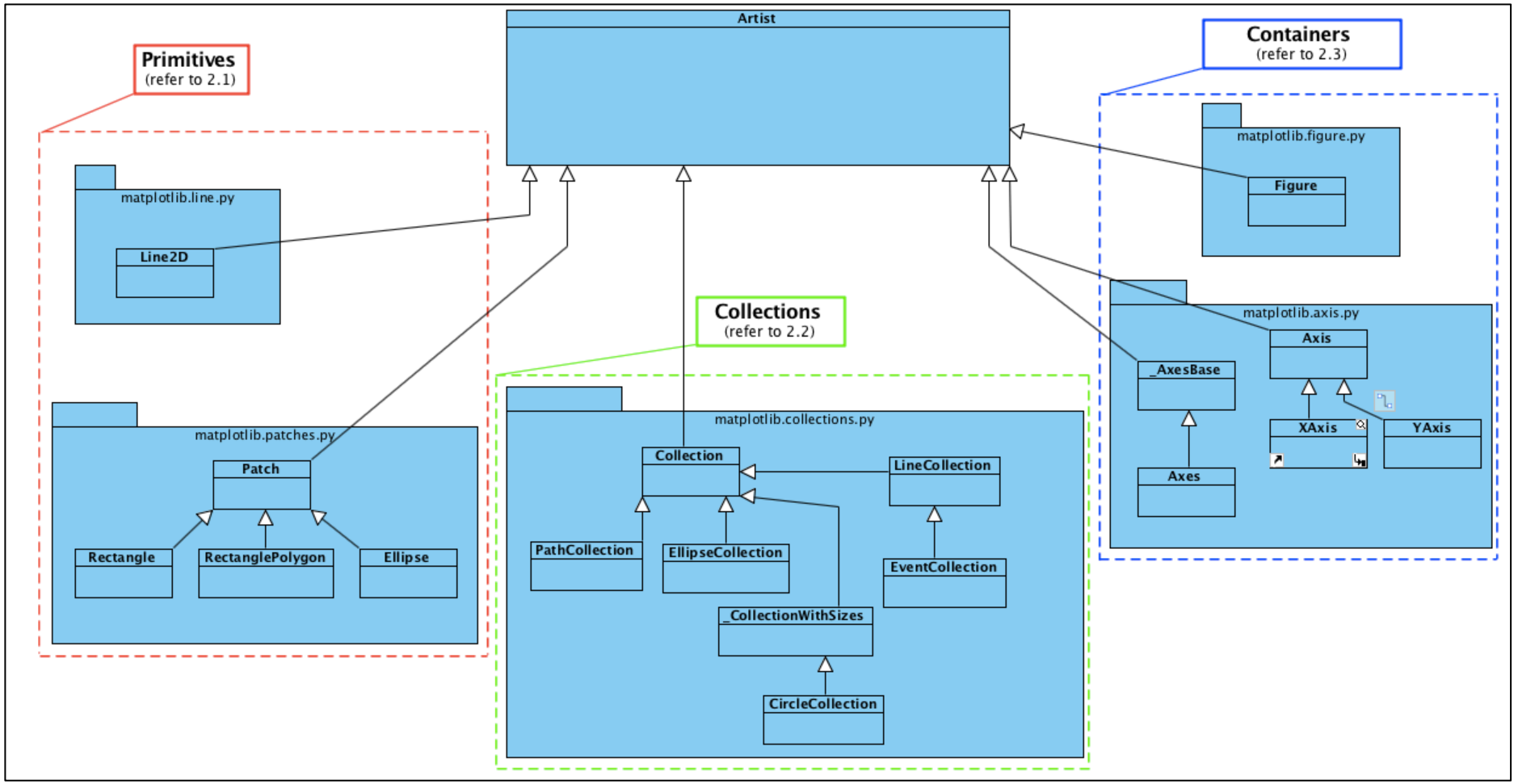
High-level Architecture Diagram of Matplotlib



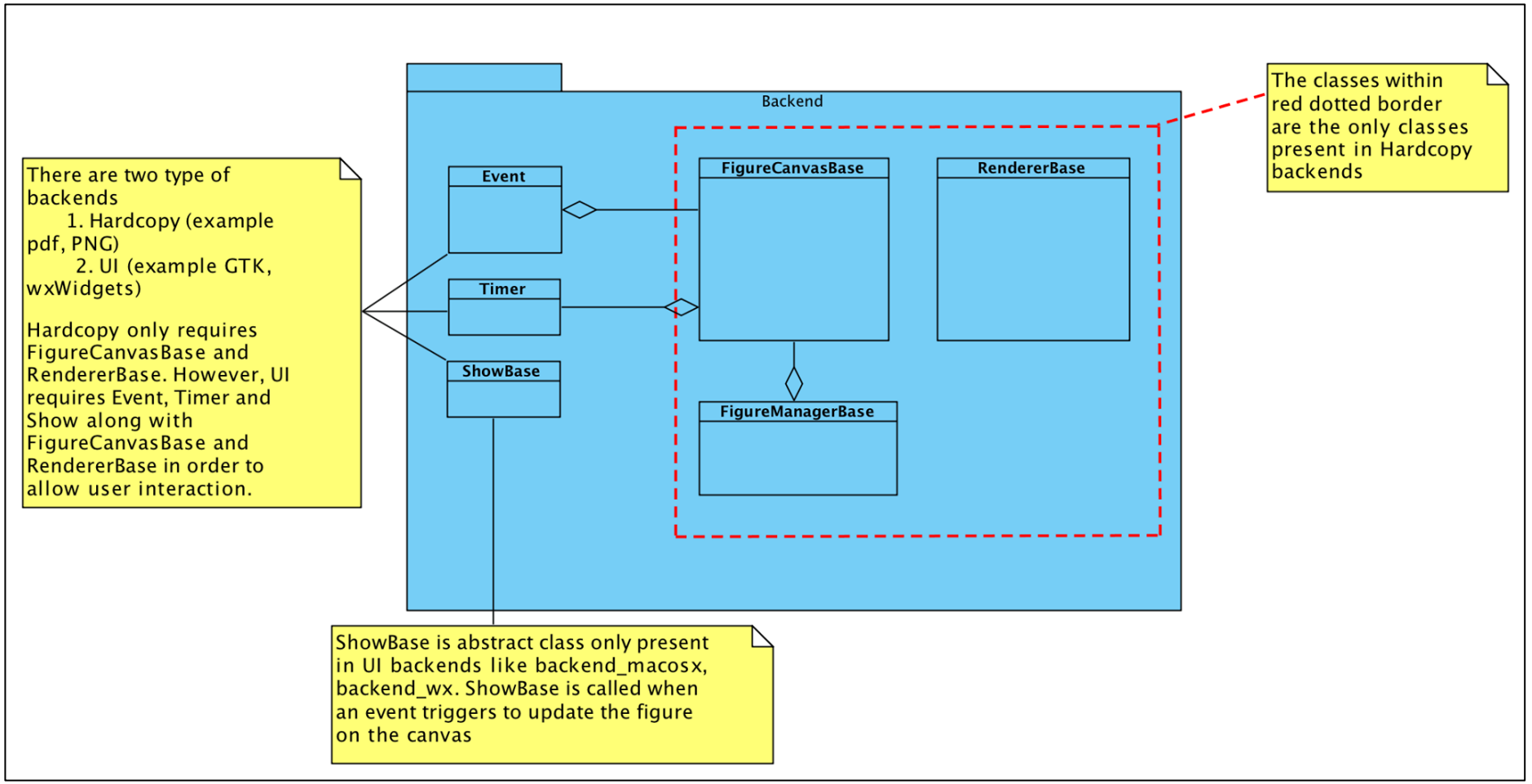
Script Layer



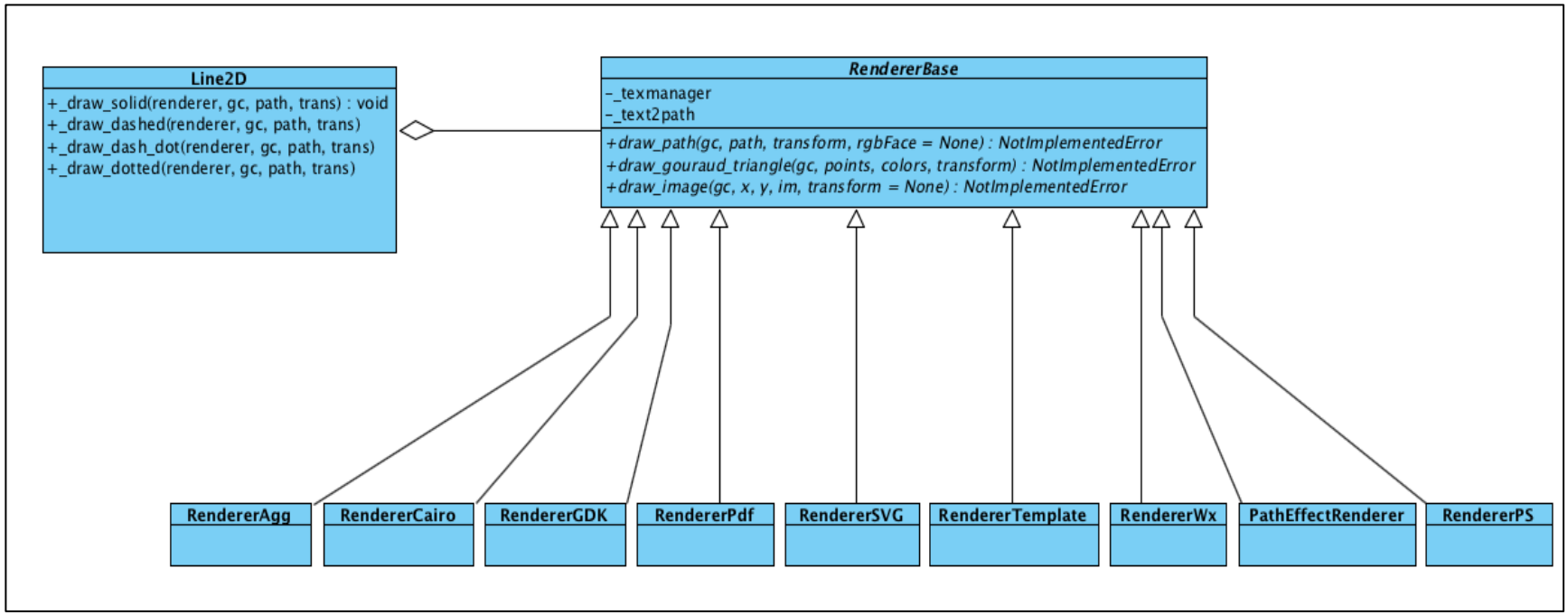
Artist Layer

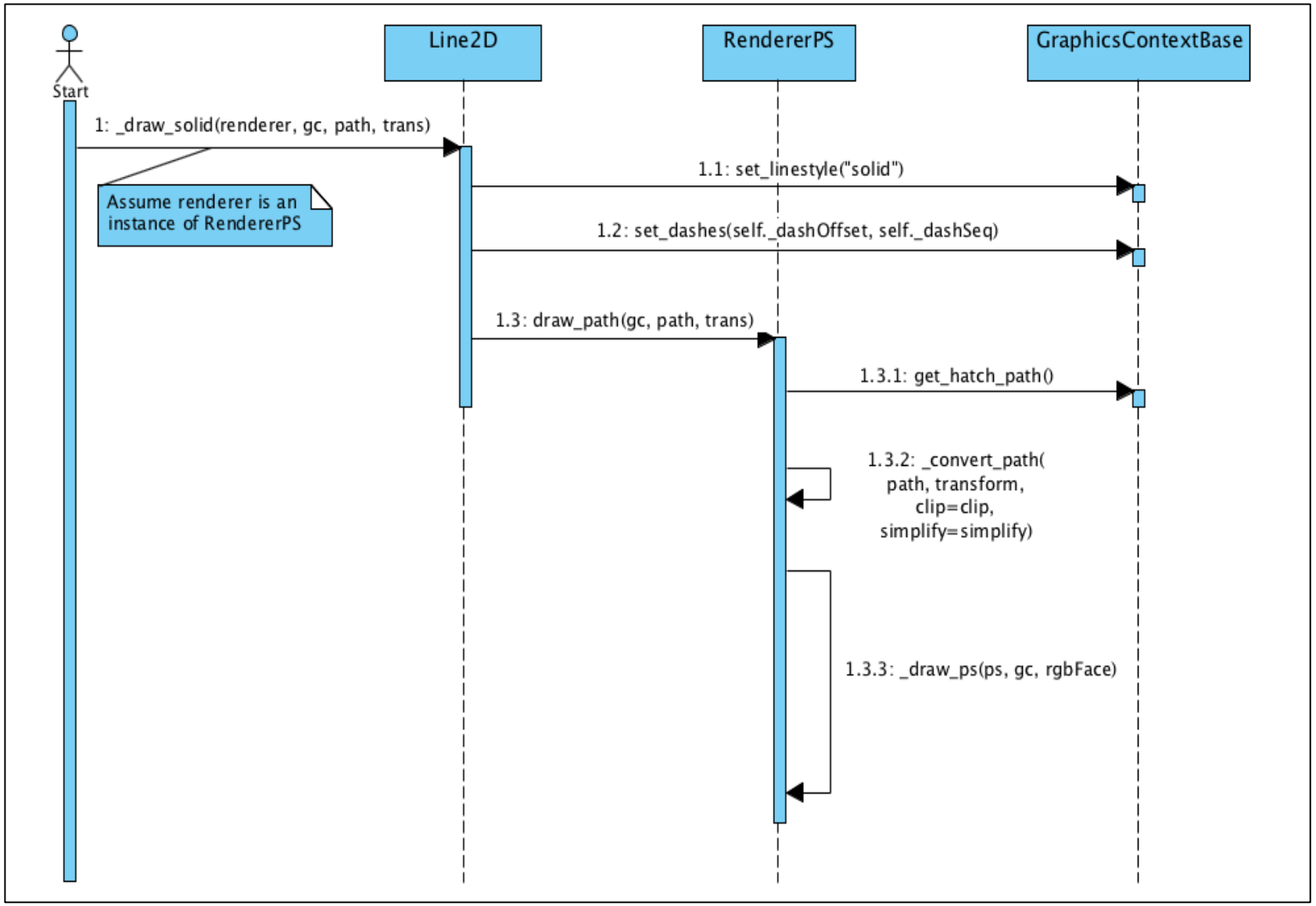


Backend Layer

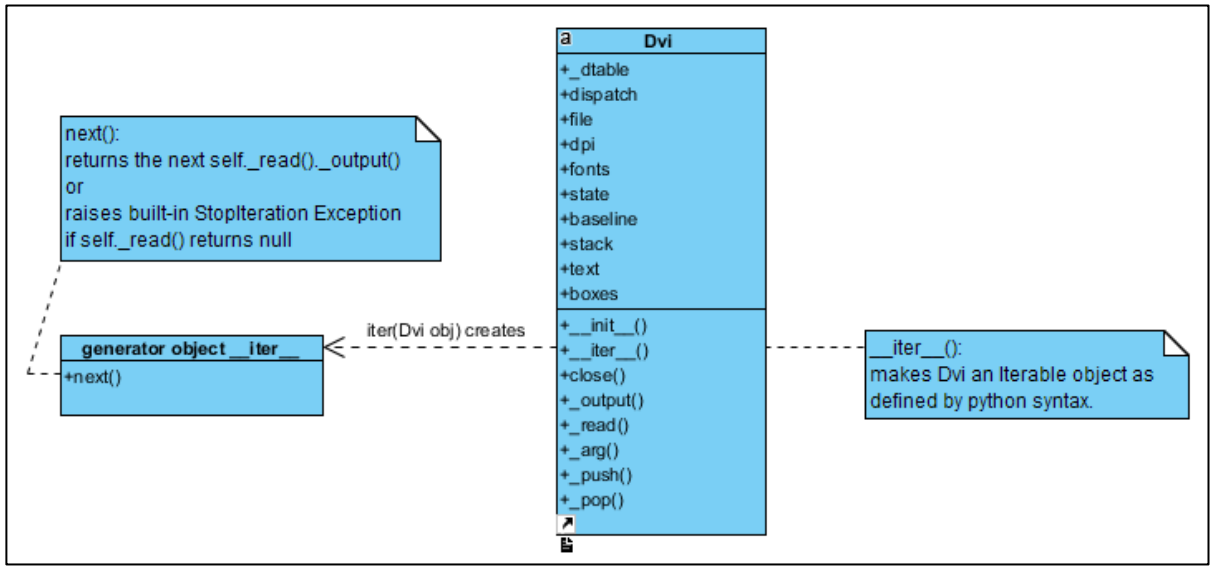


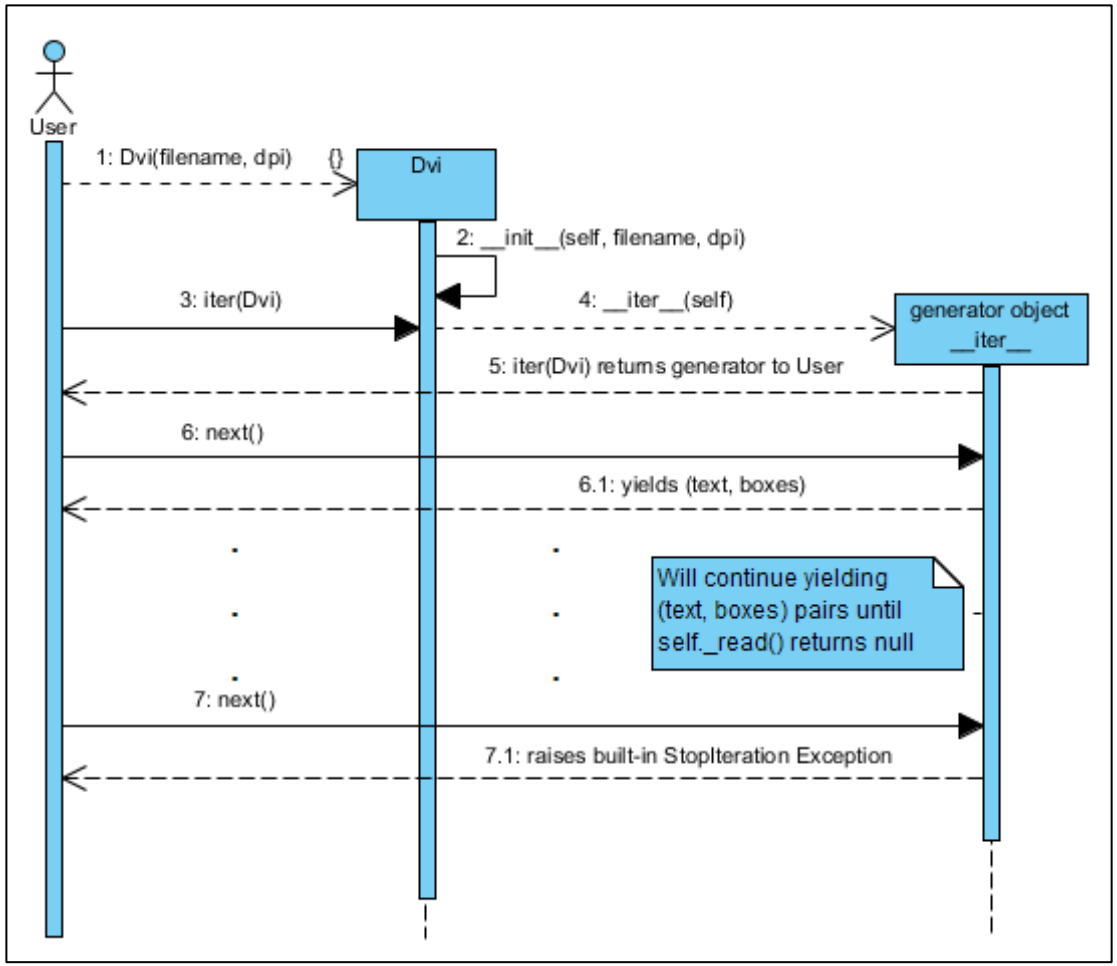
Design Pattern 1 – Strategy Design Pattern (UML and Sequence Diagram)



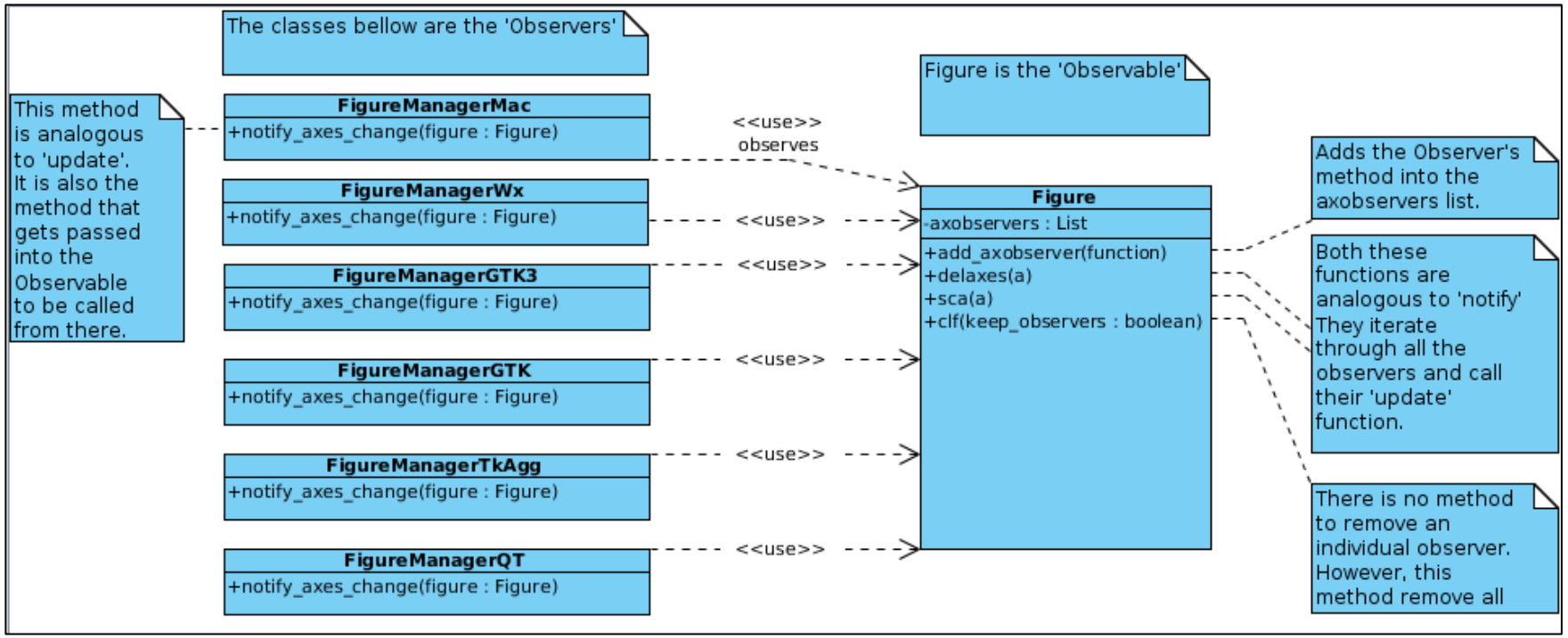

Start
Line2D
RendererPS
GraphicsContextBase
1: _draw_solid(renderer, gc, path, trans)
Assume renderer is an instance of RendererPS
1.1: set_linestyle("solid")
1.2: set_dashes(self._dashOffset, self._dashSeq)
1.3: draw_path(gc, path, trans)
1.3.1: get_hatch_path()
1.3.2: _convert_path(path, transform, clip=clip, simplify=simplify)
1.3.3: _draw_ps(ps, gc, rgbFace)

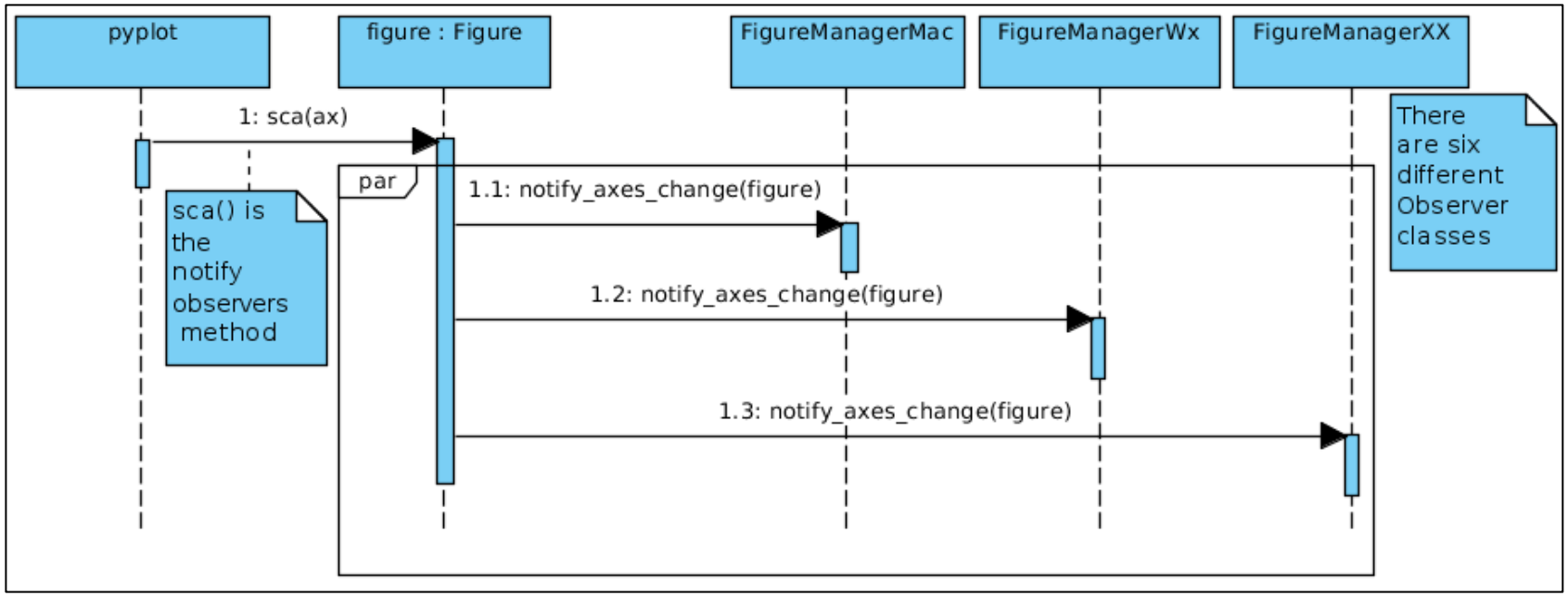
Design Pattern 2 – Iterator Design Pattern (UML and Sequence Diagram)




User
1: Dvi(filename, dpi)
{}
Dvi
2: __init__(self, filename, dpi)
3: iter(Dvi)
4: __iter__(self)
generator object
__iter__
5: iter(Dvi) returns generator to User
6: next()
6.1: yields (text, boxes)
Will continue yielding
(text, boxes) pairs until
self._read() returns null
7: next()
7.1: raises built-in StopIteration Exception

Design Pattern 3 – Observable Design Pattern (UML and Sequence Diagram)




pyplot
figure : Figure
FigureManagerMac
FigureManagerWx
FigureManagerXX
1: sca(ax)
sca() is the notify observers method
par
1.1: notify_axes_change(figure)
1.2: notify_axes_change(figure)
1.3: notify_axes_change(figure)
There are six different Observer classes