AISecurity-Research-Template

Guangtao Zhang

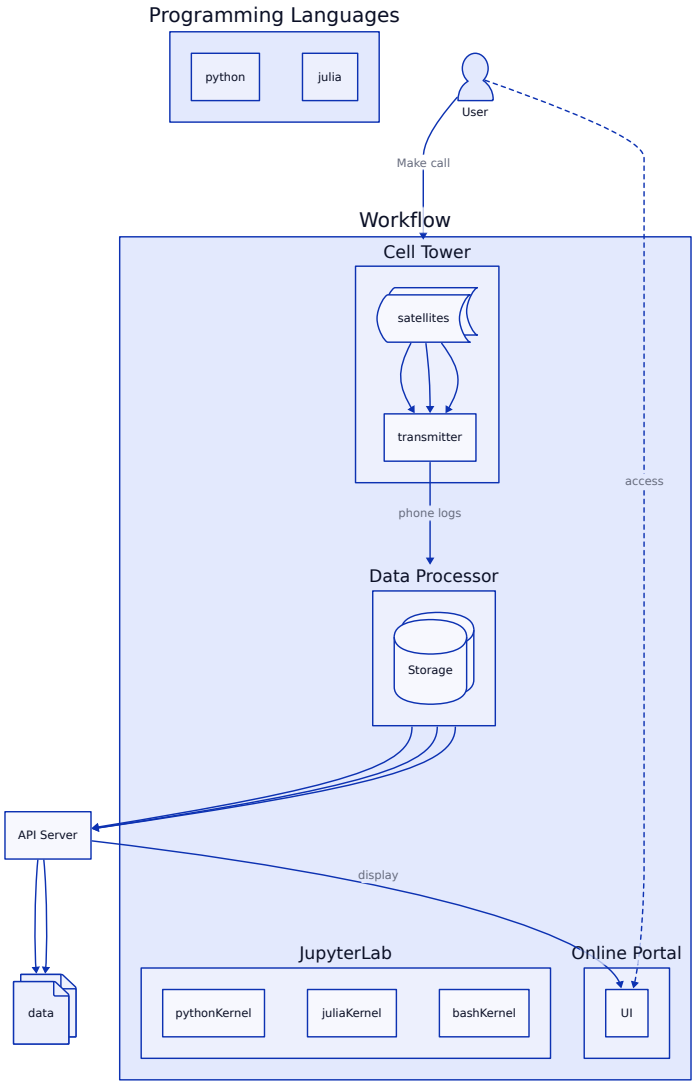
2023-12-27

Table of contents

1. Overview

2. Workflow

```
from IPython.core.display import SVG
import os

os.environ["PATH"] += os.pathsep + "$PATH"
SVG(filename='flow.svg')
```


Programming Languages
python
julia
User
Make call
Workflow
Cell Tower
satellites
transmitter
access
phone logs
Data Processor
Storage
API Server
display
JupyterLab
Online Portal
data
pythonKernel
juliaKernel
bashKernel
UI

3. Acknowledgments

- Guidelines for secure AI system development

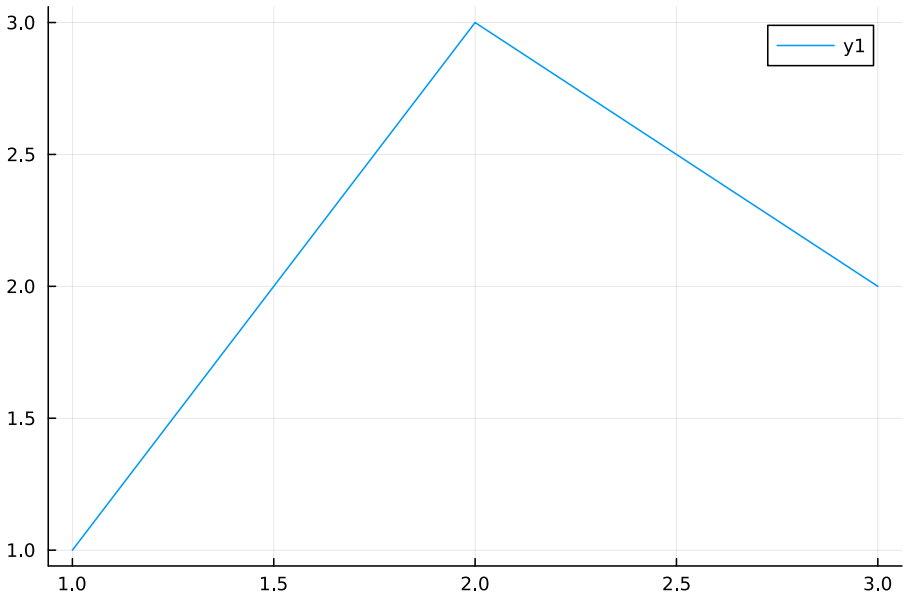
4. Julia

Part I.

Julia Plots Test

```julia
using Plots
```

```julia
display(plot([1,3,2]))
```



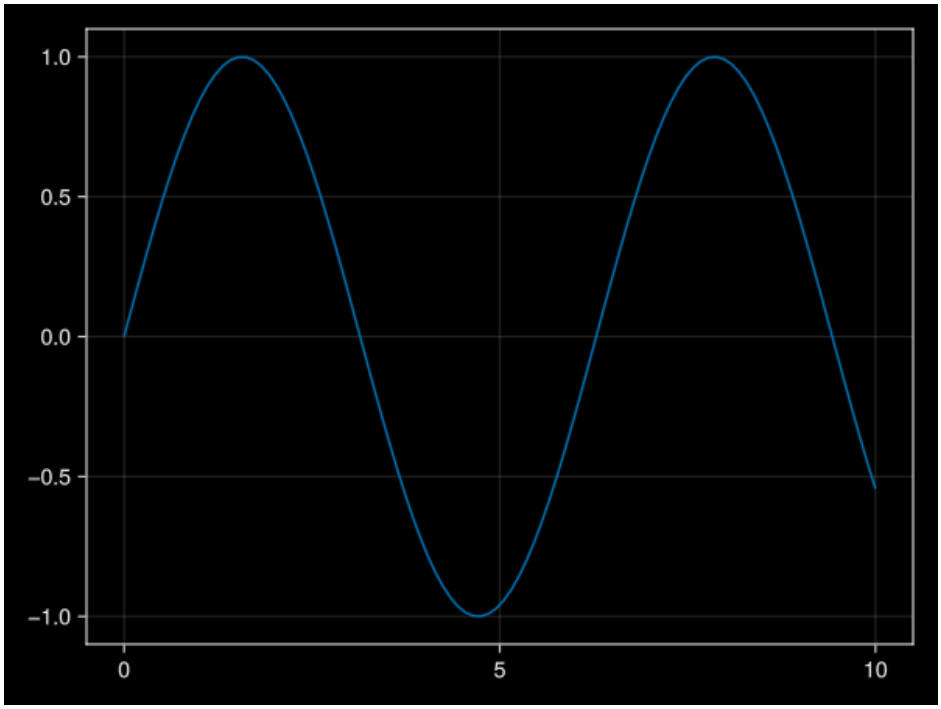

```julia
using GLMakie # All functionality is defined in Makie and every backend re-exports

Base.@kwdef mutable struct Lorenz
    dt::Float64 = 0.01
     ::Float64 = 10
     ::Float64 = 28
     ::Float64 = 8/3
    x::Float64 = 1
    y::Float64 = 1
```

```
    z::Float64 = 1
end

function step!(l::Lorenz)
    dx = l.  * (l.y - l.x)
    dy = l.x * (l.  - l.z) - l.y
    dz = l.x * l.y - l.  * l.z
    l.x += l.dt * dx
    l.y += l.dt * dy
    l.z += l.dt * dz
    Point3f(l.x, l.y, l.z)
end

attractor = Lorenz()

points = Observable(Point3f[]) # Signal that can be used to update plots efficient
colors = Observable(Int[])

set_theme!(theme_black())

fig, ax, l = lines(points, color = colors,
    colormap = :inferno, transparency = true,
    axis = (; type = Axis3, protrusions = (0, 0, 0, 0),
              viewmode = :fit, limits = (-30, 30, -30, 30, 0, 50)))

record(fig, "lorenz.mp4", 1:120) do frame
    for i in 1:50
        # update arrays inplace
        push!(points[], step!(attractor))
        push!(colors[], frame)
    end
    ax.azimuth[] = 1.7pi + 0.3 * sin(2pi * frame / 120) # set the view angle of th
```

```
    notify(points); notify(colors) # tell points and colors that their value has b
    l.colorrange = (0, frame) # update plot attribute directly
end
```

```
"lorenz.mp4"
```

```
f = Figure()
ax = Axis(f[1, 1])
x = range(0, 10, length=100)
y = sin.(x)
lines!(ax, x, y)
f
```




```
using Base64

function display_mp4(filename)
    display("text/html", string("""<video autoplay controls><source src="data:vide
    Base64.base64encode(open(read,filename)),""" type="video/mp4"></video>"""))
end
```

```
display_mp4 (generic function with 1 method)
```

```
display_mp4("lorenz.mp4")
```

```
Unable to display output for mime type(s): text/html
```

5. Python

Part II.

OpenAIPy Test

```
from scipy.special import comb
```

```
com = comb(5, 2, exact = False, repetition=True)
```

```
com
```

```
15.0
```

```
import os
from openai import OpenAI

client = OpenAI(api_key = os.getenv("OPENAI_API_KEY"))
```

```
completion = client.chat.completions.create(
  model="gpt-4",
  messages=[
    {"role": "system", "content": "You are a Security Enginner"},
  ]
)
```

```
completion.choices[0].message
```

```
ChatCompletionMessage(content='As a Security Engineer, my primary responsibility is
```