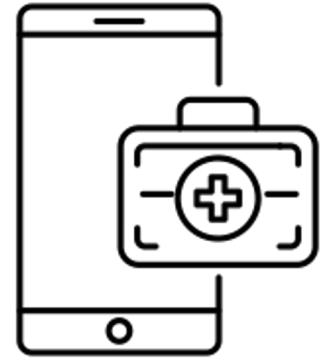
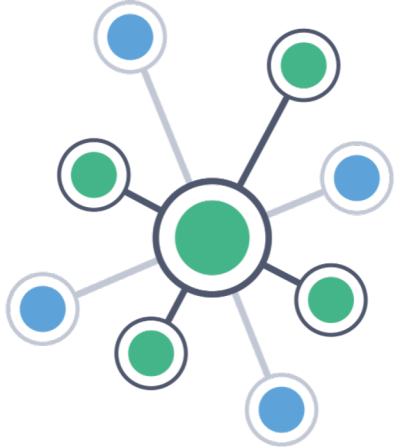




# The EPI Framework:

A dynamic infrastructure to support healthcare use cases

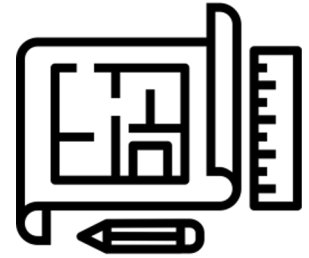
Jamila Alsayed Kassem



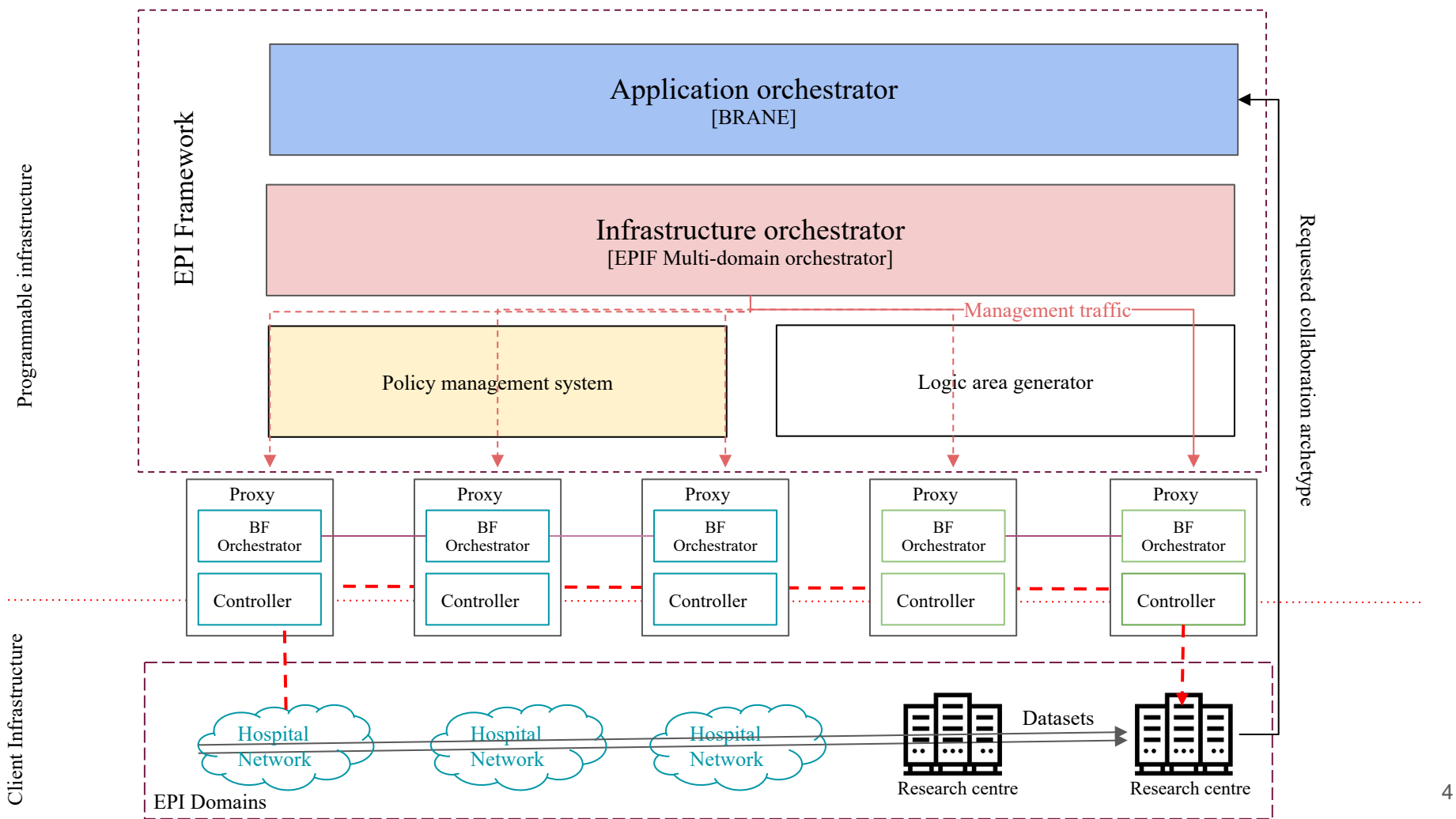
## Since last time:

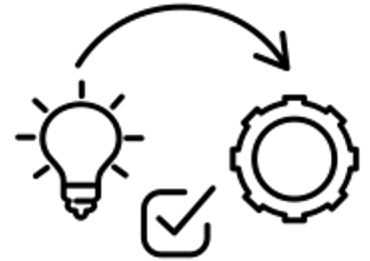
---

- eScience2021 conference accepted paper
  - <https://www.escience2021.org/>
  - Presentation: Sept. 20-23
- CompSys2021 contribution
- Teaching duties:
  - 2 students successful supervision
  - 3 students ongoing
- BF implementing
- Chaining functionality
- Ongoing experimentation
- Security by design discussion



# EPIF: The Architecture





# Proxy Implementations

# The proxy within the architecture

---

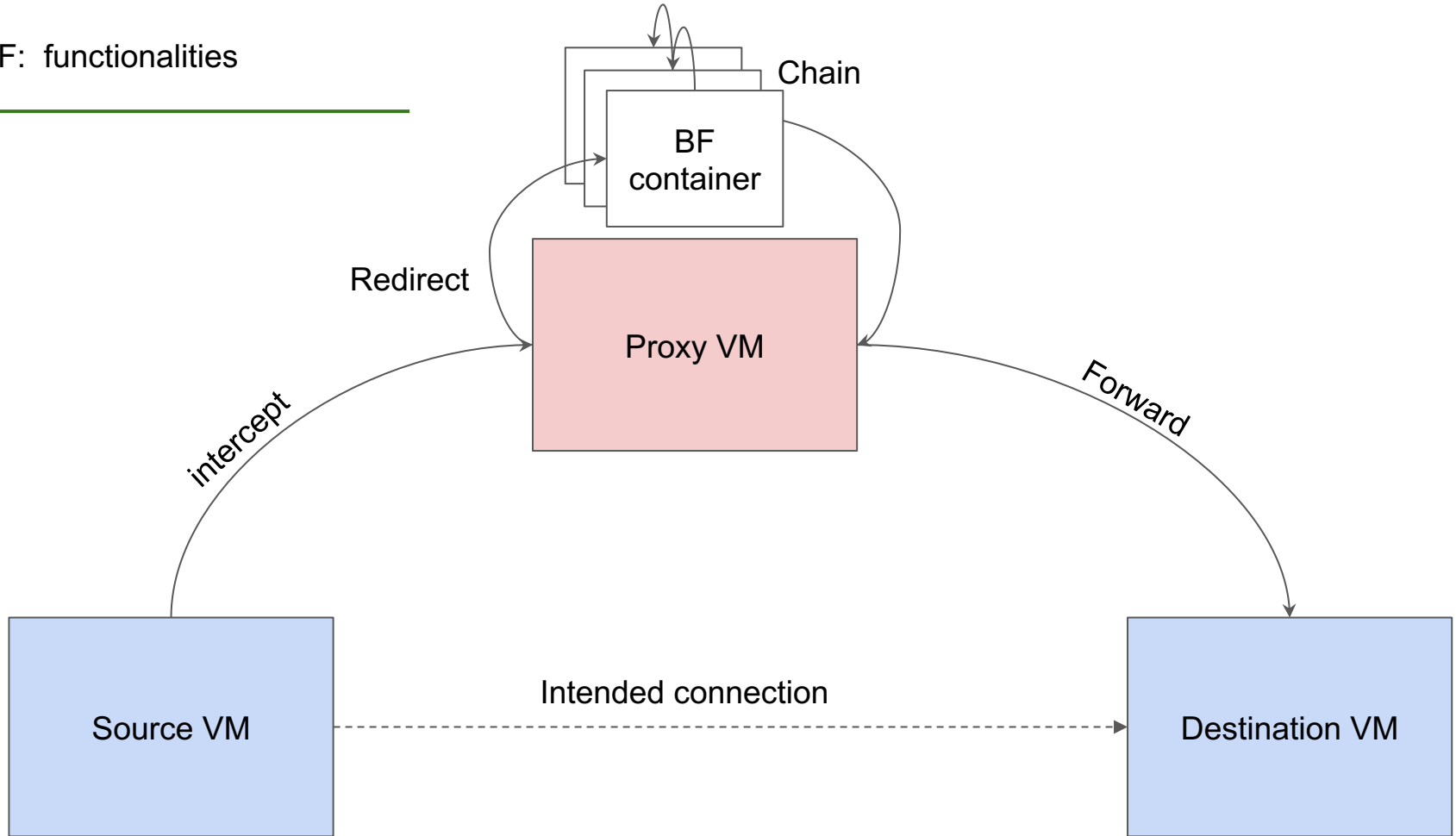
1. The proxy serves as the EPIF actors
2. Functionalities within the architecture:
  - a. Manipulate traffic by intercepting and redirecting packets
  - b. hosting bridging functions
  - c. chaining the BFs once instantiated
3. Proxy servers:
  - a. NGINX
  - b. SOCKS5
  - c. SOCKS6

***Which proxy is the most suitable choice in terms of time overhead and throughput?***

***Design decision: depends on the application requirements and the relevance of performance parameters (eScience paper)***

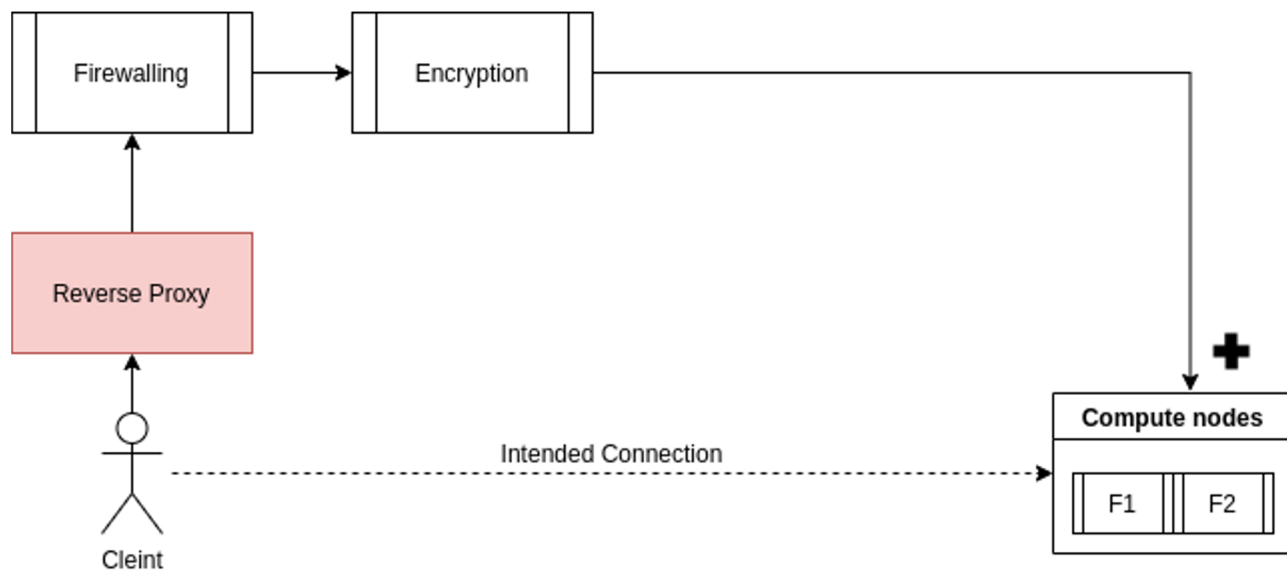
# EPIF: functionalities

---



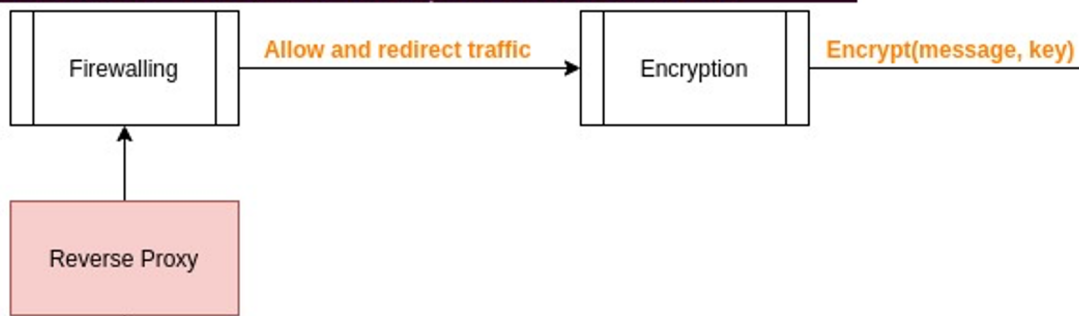
## Chaining scenario

---





```
vagrant@vagrant:~$ sudo docker-compose up
Starting firewall ... done
Recreating encryptionserver ... done
Recreating nginx ... done
Attaching to firewall, encryptionserver, nginx
```



```
vagrant@vagrant:~/client$ sudo docker-compose up
Starting s-client ... done
Attaching to s-client
s-client | b'I can confirm I got your message!'
s-client | 0.0017595291137695312
s-client exited with code 0
```

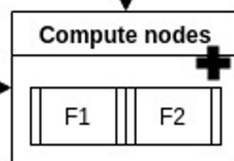
"Hello to the server via the proxy"

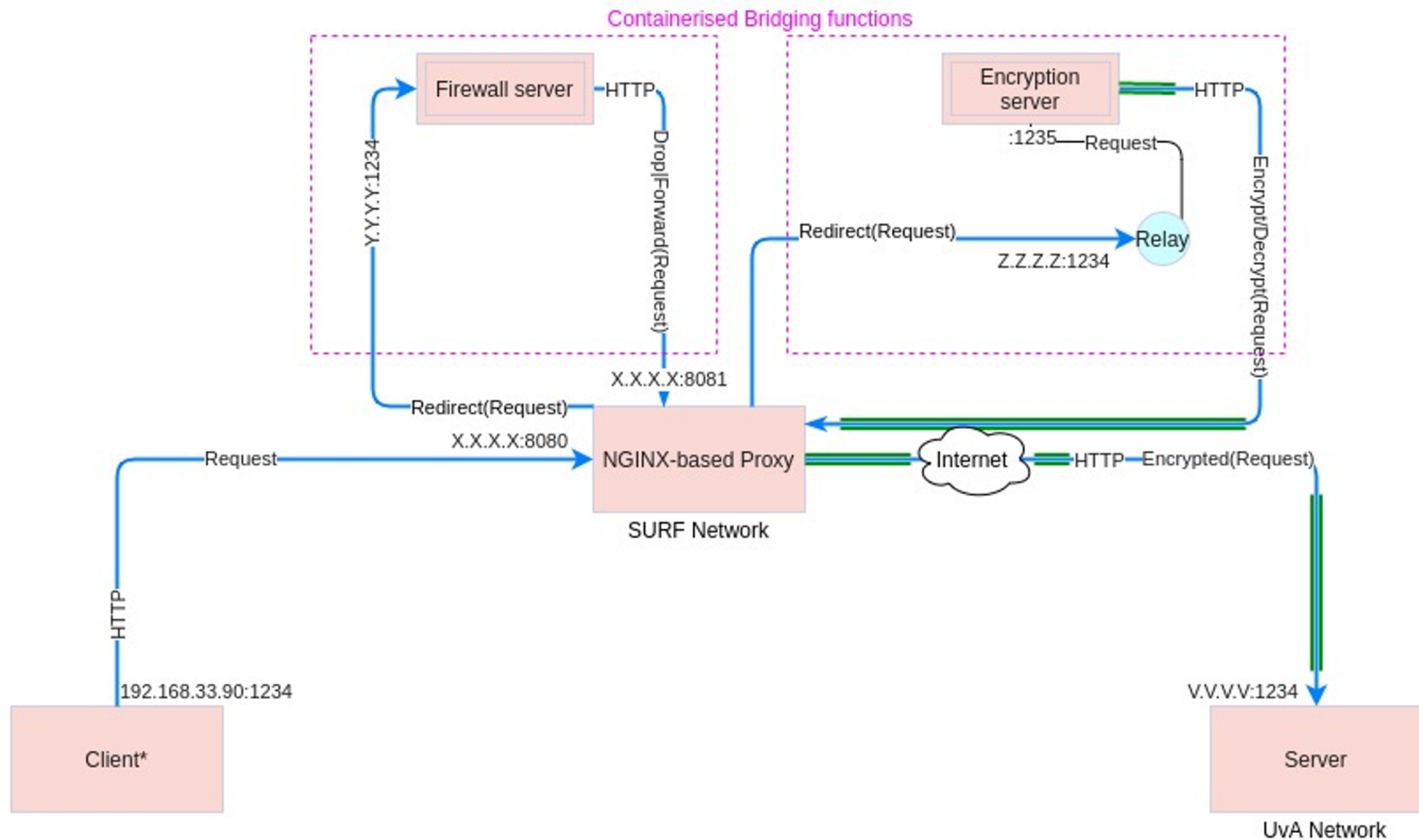


Intended Connection

```
vagrant@vagrant:~/server$ sudo docker-compose up
Starting d-server ... done
Attaching to d-server
```

"I confirm I got your message"





\*The specific port for this application request is shared with the client apriori



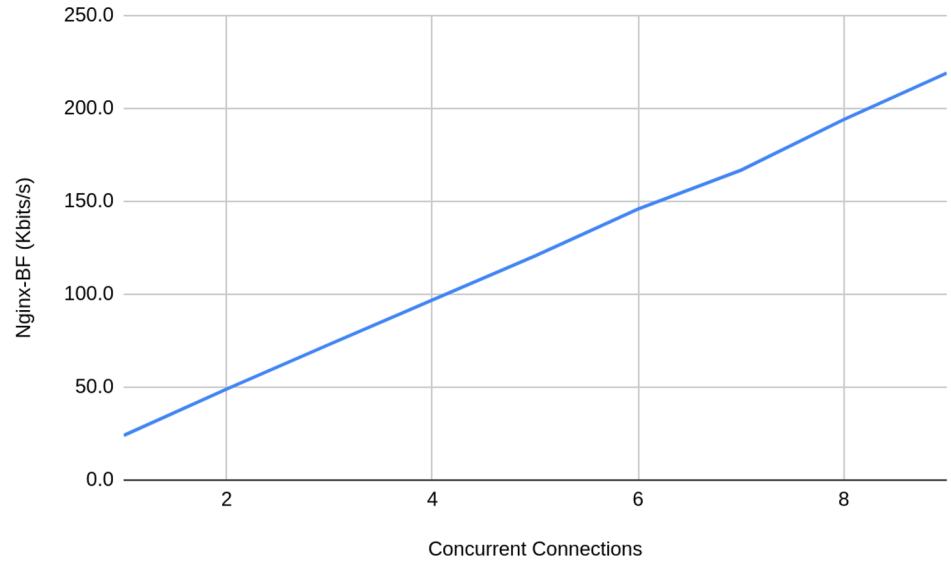
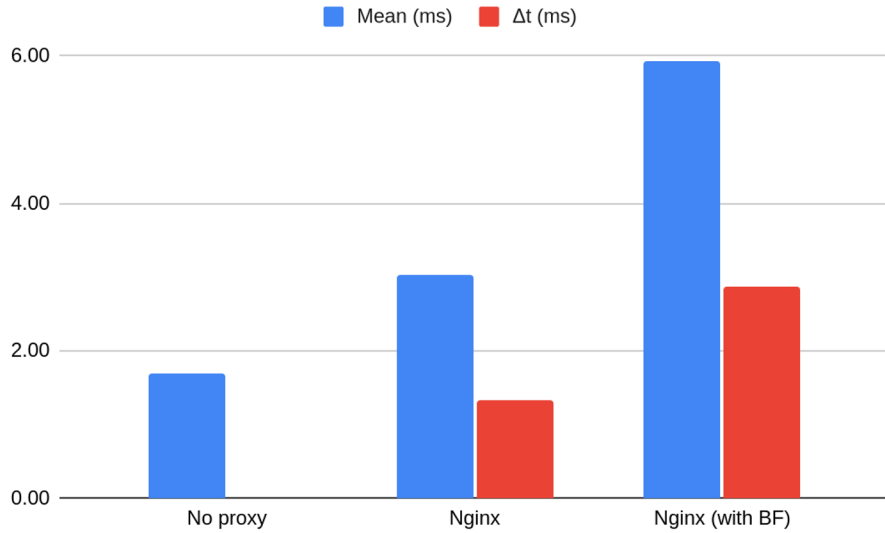
# Experiments

# Interesting parameters

---

- Network and performance:
  - Time overhead: Baseline to the previous experiment setups
  - Memory overhead
  - Throughput
  - Proxy resilience
  - Scalability
- Proxy security and the added security value
- Success rate
- Infrastructure flexibility

# Preliminary Results



# Future work

---

## Implementation wise:

- Real network environment
- Implementing more python functions
  - Packet inspection
  - UDP hole punching, etc.
- Dynamic BF chaining
- Uniform interfaces of bridging functions
- BRANE integration
- DSL extension to express BFs
- Real test-beds with introducing real health data

## Research and open questions:

- Security by design
- Auditing distributed proxies across domains.
- ***How would the performance scale with increasingly complex chains?***
- ***Other performance parameters overlooked?***
- ***How do we scale the proxy? What is factoring into this decision?***
- Policy integration

# July

# September

# October

# November

▲ EPIF/BRANE: Components implementation and experimentation

Education tasks

BRANE integration and DSL extension

Security discussions and implementation

EPIF/BRANE/Other PhDs

Initial integration of policy