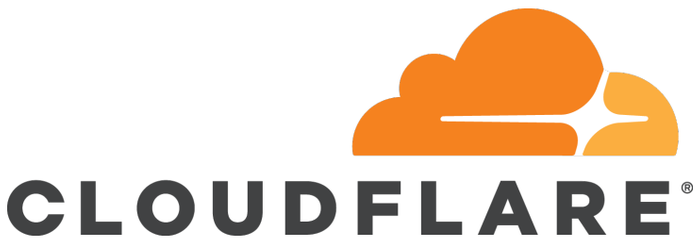
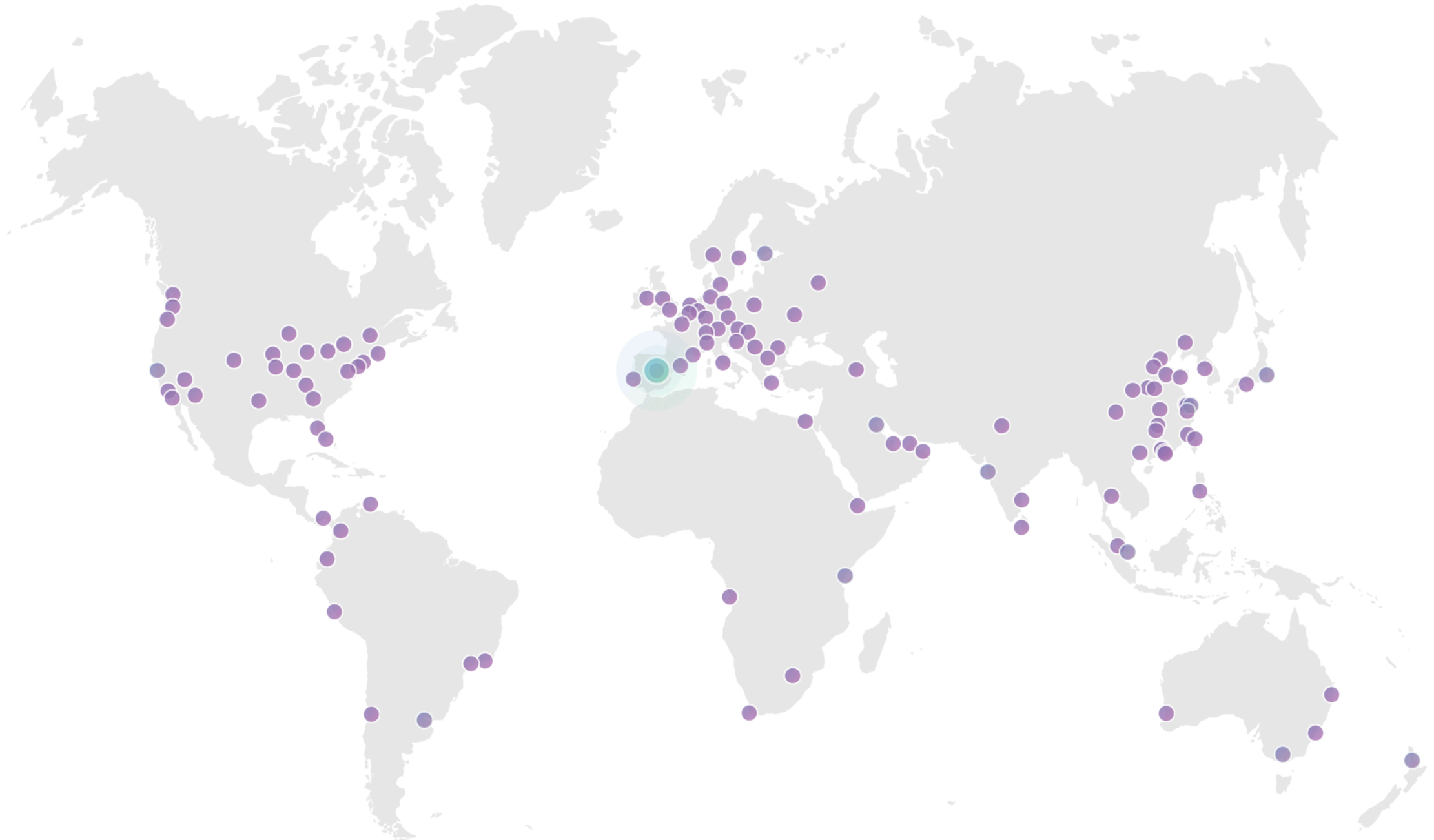



# Gatelogic FRP framework

## ...200 lines of Python I still regret...

Marek Majkowski @majek04

# Who we are?

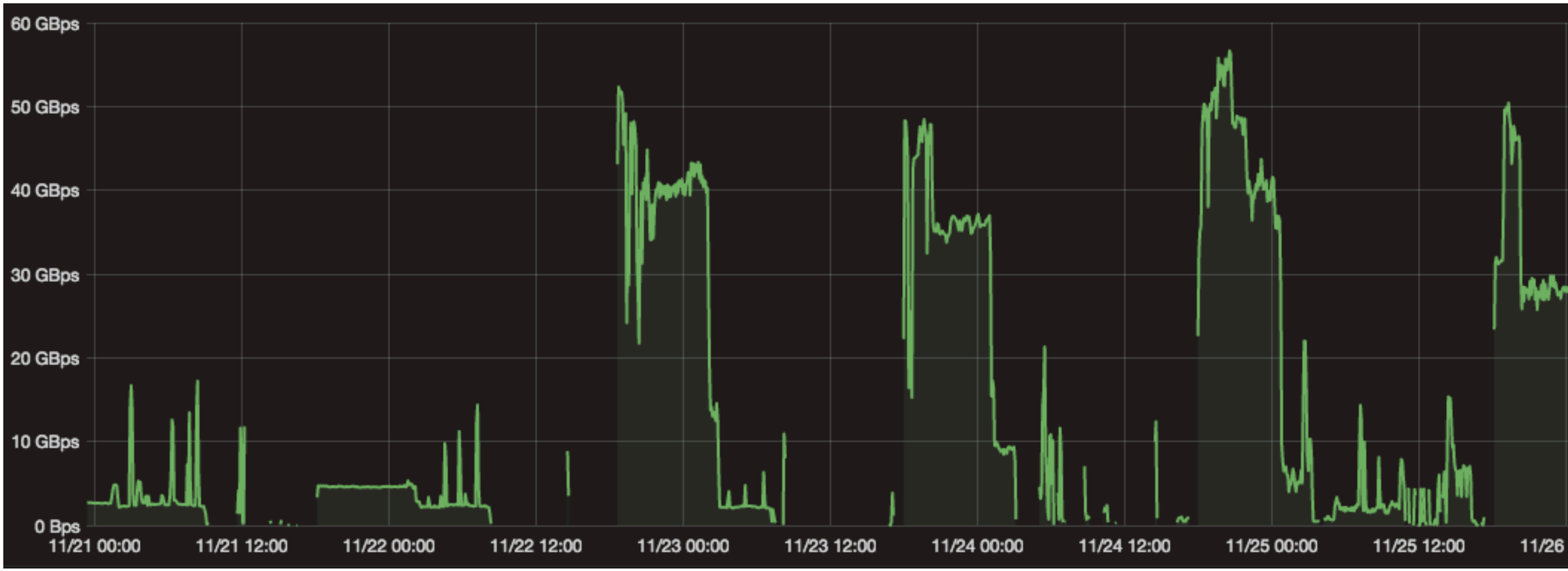


# Reverse proxy


- Optimizations
- Caching
- DDoS protection
- Security

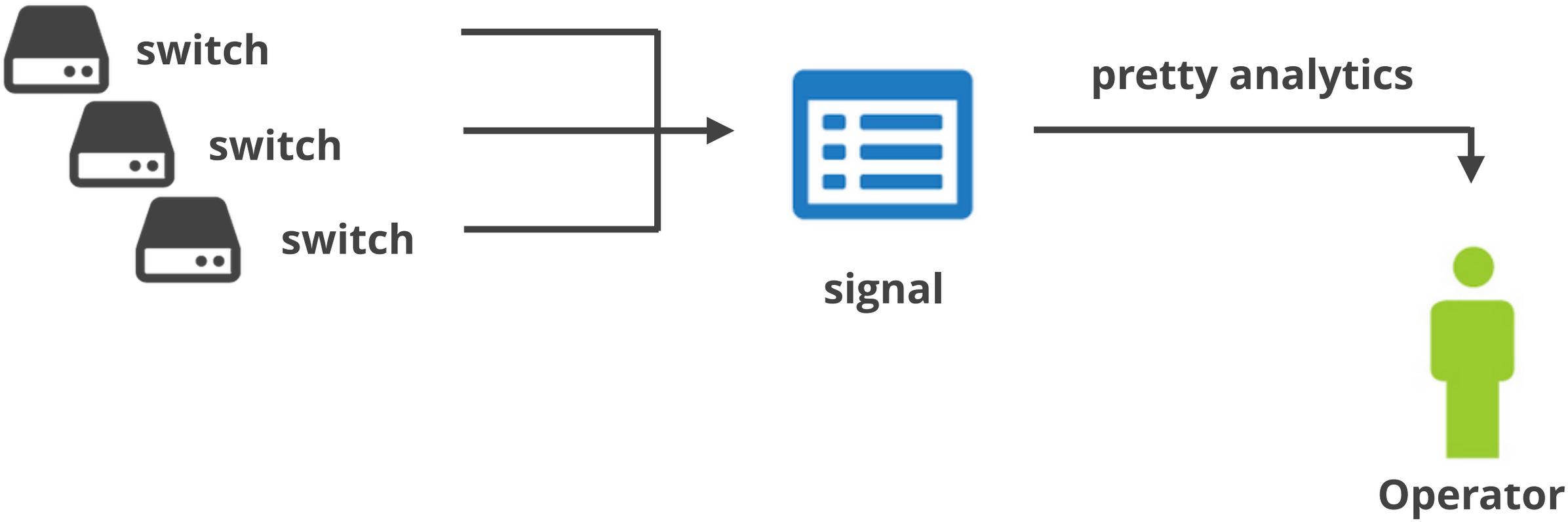
CLOUDFLARE

# Attacked




CLOUDFLARE

# Signal

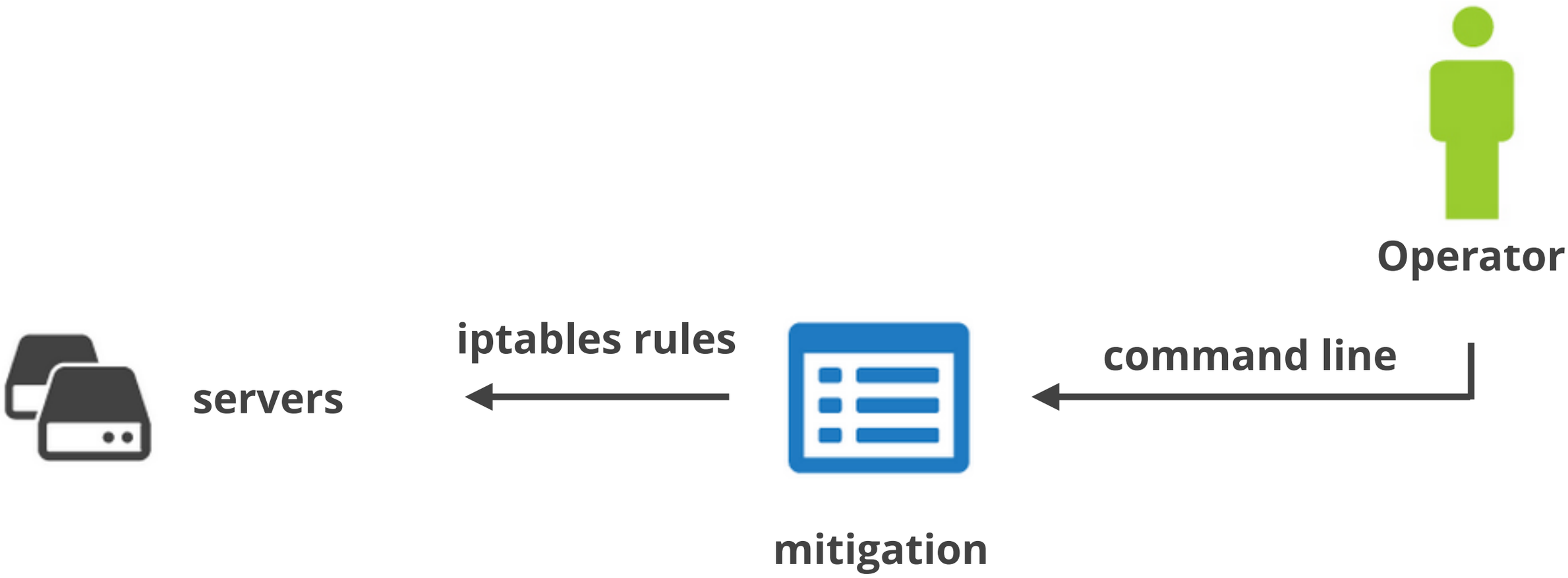


# Signal

```
INCOMING/UDP  samples=11836 pps={total= 65.176M, accounted= 65.174M}
   Mpps   rule hint
  1.160     dnsbpf -- --ip=173.245.59.                www.          com
  3.953     dnsbpf -- --ip=173.245.59.                www.       .com
  3.892     dnsbpf -- --ip=173.245.58.                        .com
  4.169     dnsbpf -- --ip=173.245.58.                        .com
  3.750     dnsbpf -- --ip=173.245.58.                www.         .com
  4.098     dnsbpf -- --ip=173.245.59.                      .com
  0.965     dnsbpf -- --ip=173.245.59.                      .com
  3.910     dnsbpf -- --ip=173.245.59.                       com
  0.973     dnsbpf -- --ip=173.245.59.                www.       .com
  3.811     dnsbpf -- --ip=173.245.58.                www.     .com
  3.770     dnsbpf -- --ip=173.245.59.                       .com
  3.879     dnsbpf -- --ip=173.245.58.                       .com
  3.857     dnsbpf -- --ip=173.245.58.                      .com
  3.975     dnsbpf -- --ip=173.245.58.                www.       .com
  0.545     dnsbpf -- --ip=                  .
  3.406     dnsbpf -- --ip=                  .
  0.315     dnsbpf -- --ip=173.245.58.
  0.415     dnsbpf -- --ip=173.245.59.                *.       .com
```

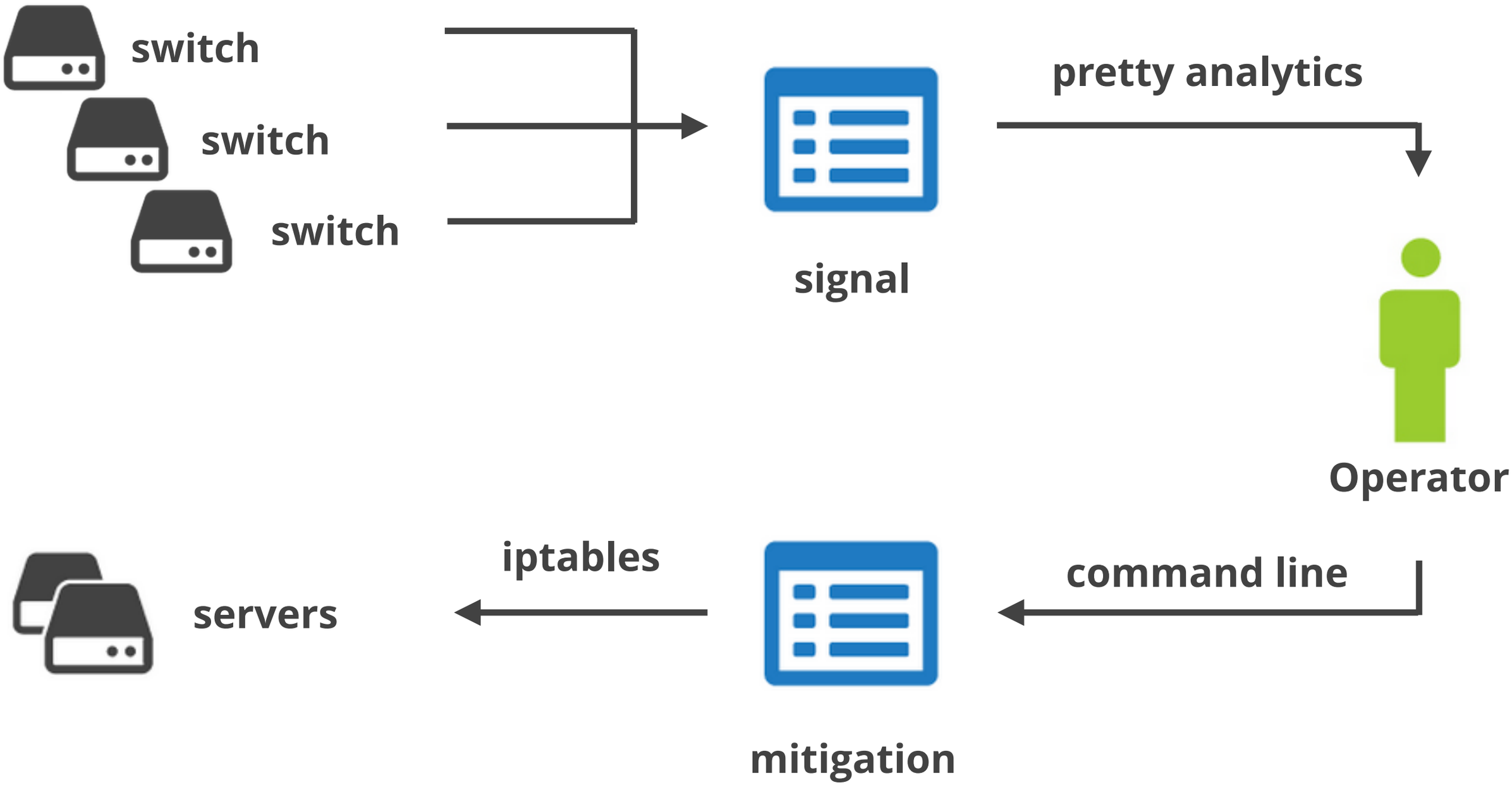

WHAT?

# Mitigation



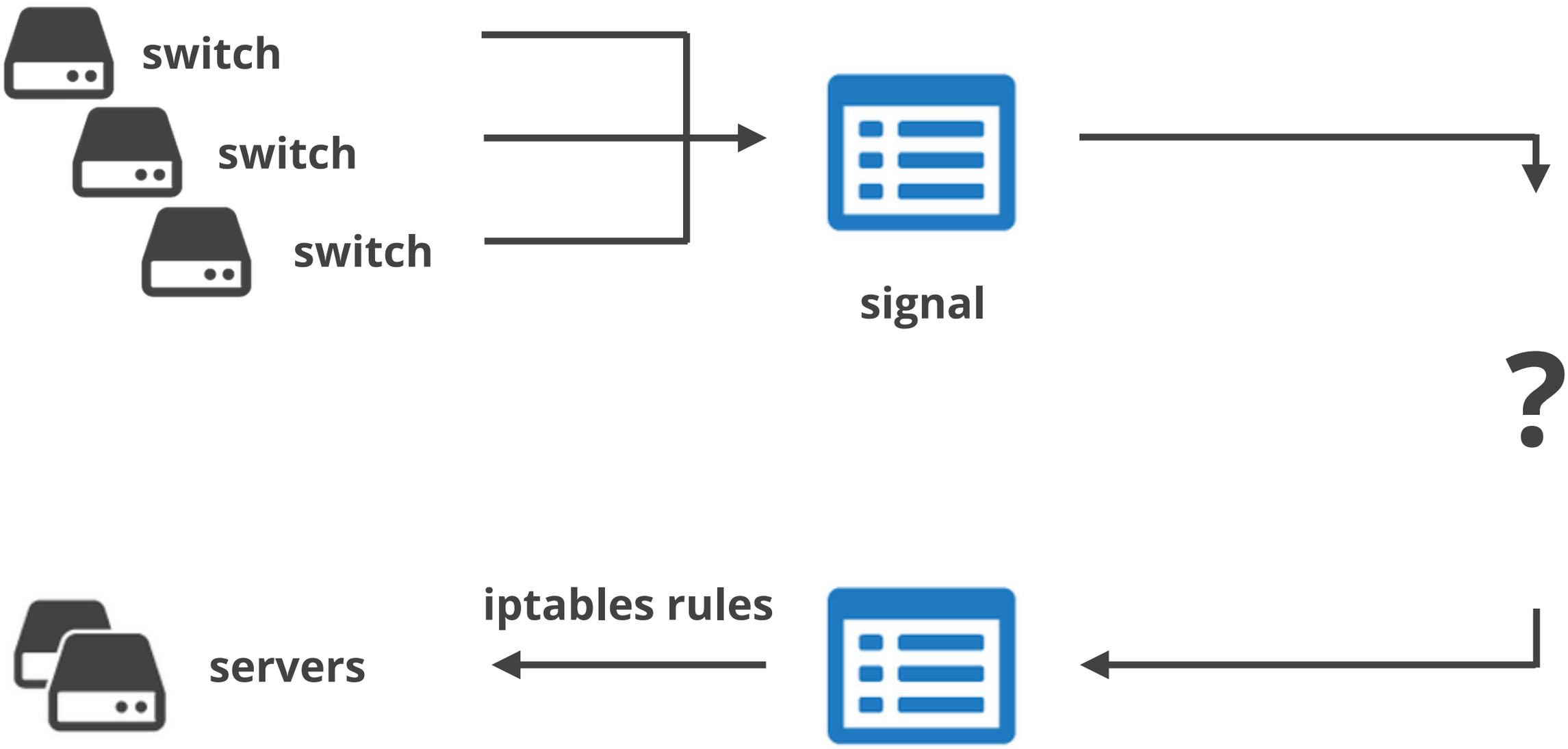
# Mitigation

```
$ gatekeeper dnsbpf list
[ ] Imported: 0 removed: 0 trusted: 25
--expiry=365d -- --ip=173.245.[redacted] [redacted] www.[redacted].com
--expiry=7d -- --ip=[redacted] [redacted] m.[redacted].com
--expiry=7d -- --ip=[redacted] [redacted] m.[redacted].com
--expiry=7d -- --ip=[redacted] [redacted] www.[redacted].com
--expiry=7d -- --ip=[redacted] [redacted] www.[redacted].com
--ip=173.245.[redacted] [redacted] www.[redacted].com
--ip=173.245.[redacted] *.www.[redacted].com
--ip=173.245.[redacted] [redacted] www.[redacted].com
--ip=[redacted] [redacted] [redacted].com
--ip=[redacted] [redacted] [redacted].com
--ip=[redacted] *.[redacted].com *.[redacted].com *.www.[redacted].com
--ip=[redacted] *.www.[redacted].com *.[redacted].com
--ip=[redacted] *.www.[redacted].com *.www.[redacted].com
--ip=[redacted] [redacted] [redacted], [redacted], [redacted]
--ip=[redacted] *.[redacted].com *.[redacted].com *.www.[redacted].com
--ip=[redacted] *.[redacted].com *.[redacted].com
```


switch
switch
switch
signal
pretty analytics
Operator
command line
iptables
servers
mitigation

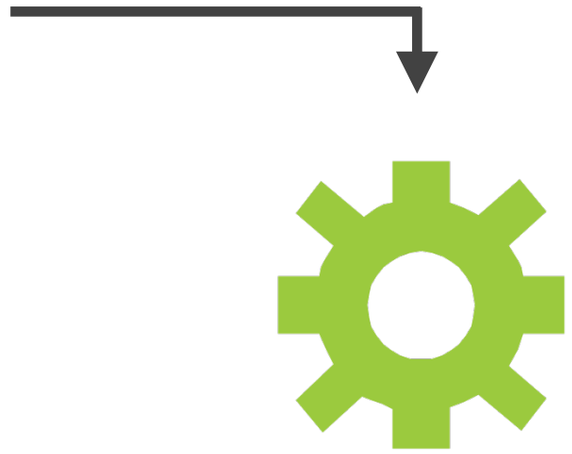
# Copy-pasta

# "Business logic"



# 12 months later…

```
--ip=1.2.3.4 example.com
```



Business logic

```
--ip=1.2.3.4 example.com --qps=100
```

```
example.com = FREE | PAID
```

```
--ip=1.2.3.4 example.com
```

Business logic

```
--ip=1.2.3.4 example.com --qps=500
```

```
example.com subdomains:
(www, ns1, ns2)
```

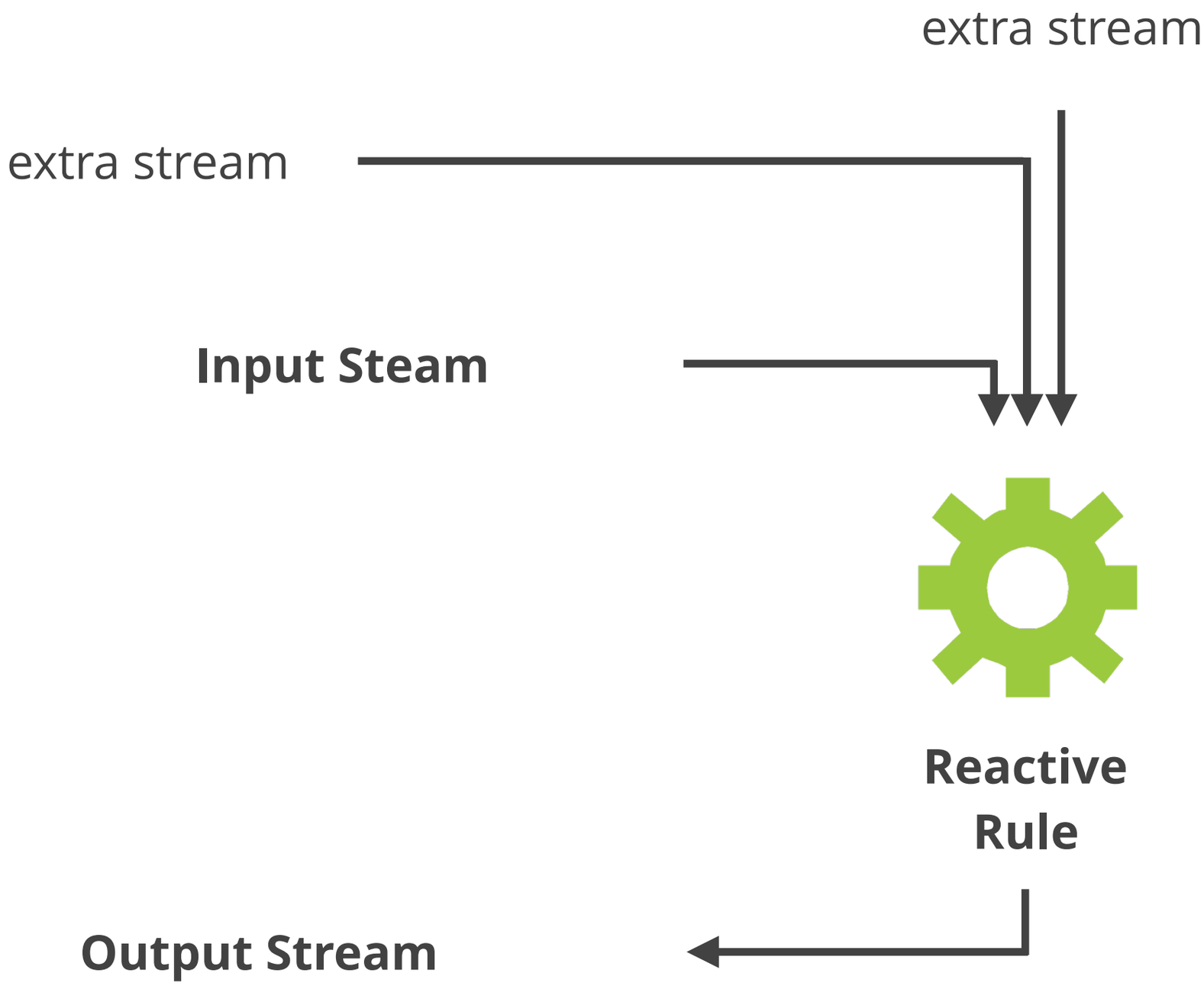
```
example.com = FREE | PAID
```

```
--ip=1.2.3.4 example.com
```

Business logic

```
--ip=1.2.3.4 example.com --except www,n1,ns2 --qps=500
```

CLOUDFLARE


extra stream
extra stream
Input Steam
Reactive
Rule
Output Stream

# Reactive rule

```
def dns_mitigation(attack, plan, subdomains, toggles):
    domain = attack['domain']

    if toggles['all_mitigations_disabled']: return

    qps = 100
    if plan[domain] == 'business':
        qps = 500

    mitigation =
        attack['description'] + \
        ' --qps=%s' % qps + \
        ' --except=%s'.join(subdomains[domain])

    return mitigation
```

# Subscriptions

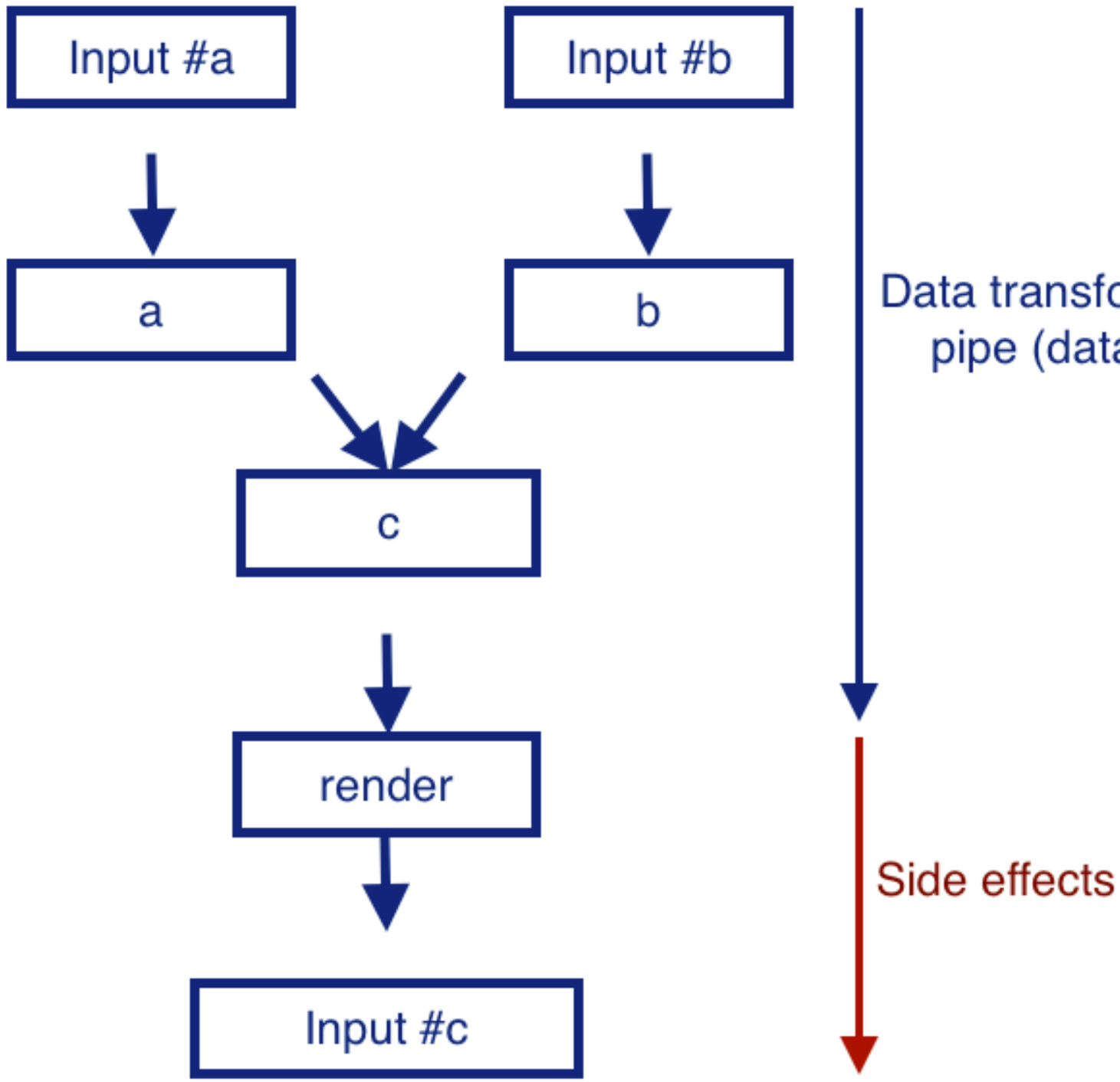
```
def dns_mitigation(attack, plan, subdomains, toggles):
    domain = attack['domain']

    if toggles['all_mitigations_disabled']: return

    qps = 100
    if plan[domain] == 'business':
        qps = 500

    mitigation =
        attack['description'] + \
        ' --qps=%s' % qps + \
        ' --except=%s'.join(subdomains[domain])

    return mitigation
```

# Business logic

- Hard problem!
- Multiple DB lookups
- Wait for operator confirmation
- Critical path

CLOUDFLARE

# Functional reactive programming




interaction

reaction

# models - Excel

PowerBI_Test_Data.xlsx - Excel — Mark Kaelin

File Home Insert Page Layout Formulas Data Review View Tell me what you want to do Share

L14

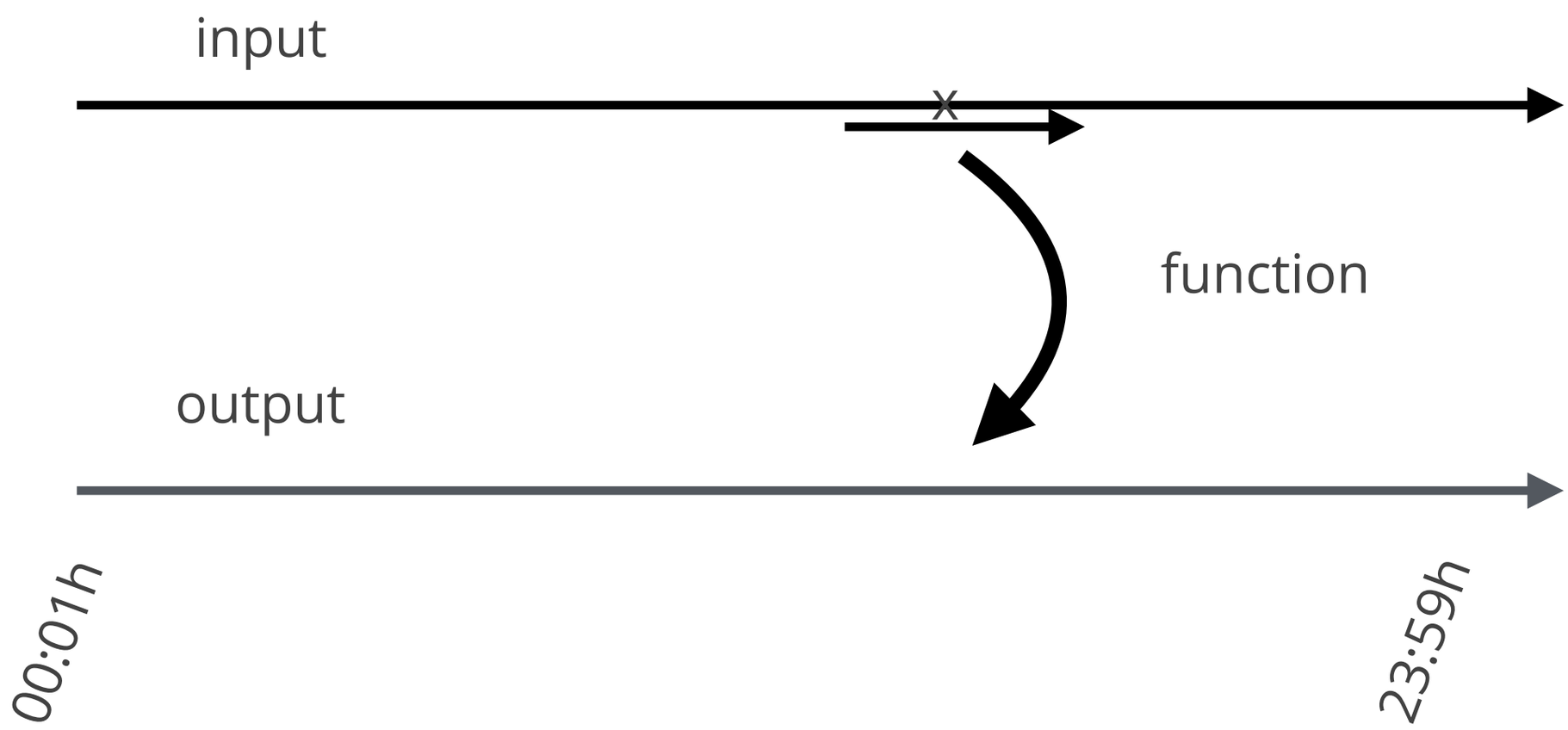
| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Stock Name | Symbol | Shares | Purchase Price | Cost Basis | Current Price | Market Value | Gain/Loss | Dividend/share | Annual Yield |
| 2 | Apple | AAPL | 100 | $90.00 | $9,000.00 | $144.13 | $14,413.27 | $14,269.14 | $2.28 | 1.58% |
| 3 | Microsoft | MSFT | 200 | $32.00 | $6,400.00 | $65.57 | $13,114.14 | $13,048.57 | $1.56 | 2.38% |
| 4 | Salesforce | CRM | 150 | $25.00 | $3,750.00 | $82.57 | $12,385.50 | $12,302.93 | $0.00 | 0.00% |
| 5 | Oracle | ORCL | 250 | $50.00 | $12,500.00 | $44.56 | $11,138.75 | $11,094.20 | $0.64 | 1.44% |
| 6 | Hewlett Packard Enterprise | HPE | 500 | $18.00 | $9,000.00 | $17.69 | $8,842.50 | $8,824.82 | $0.26 | 1.47% |
| 7 | Alphabet | GOOG | 100 | $225.00 | $22,500.00 | $833.36 | $83,336.00 | $82,502.64 | $0.00 | 0.00% |
| 8 | Intel | INTC | 200 | $22.00 | $4,400.00 | $36.07 | $7,213.00 | $7,176.94 | $1.09 | 3.02% |
| 9 | Cisco | CSCO | 225 | $18.00 | $4,050.00 | $33.24 | $7,478.78 | $7,445.54 | $1.16 | 3.49% |
| 10 | Qualcomm | QCOM | 185 | $65.00 | $12,025.00 | $56.48 | $10,447.88 | $10,391.40 | $2.12 | 3.75% |
| 11 | Amazon | AMZN | 50 | $800.00 | $40,000.00 | $897.64 | $44,882.00 | $43,984.36 | $0.00 | 0.00% |
| 12 | Redhat | RHT | 100 | $95.00 | $9,500.00 | $86.26 | $8,626.00 | $8,539.74 | $0.00 | 0.00% |
| 13 | Facebook | FB | 1000 | $17.00 | $17,000.00 | $141.64 | $141,640.00 | $141,498.36 | $0.00 | 0.00% |
| 14 | Twitter | TWTR | 500 | $45.00 | $22,500.00 | $14.61 | $7,302.55 | $7,287.94 | $0.00 | 0.00% |
| 15 | | | | | | | | | | |

Sheet1

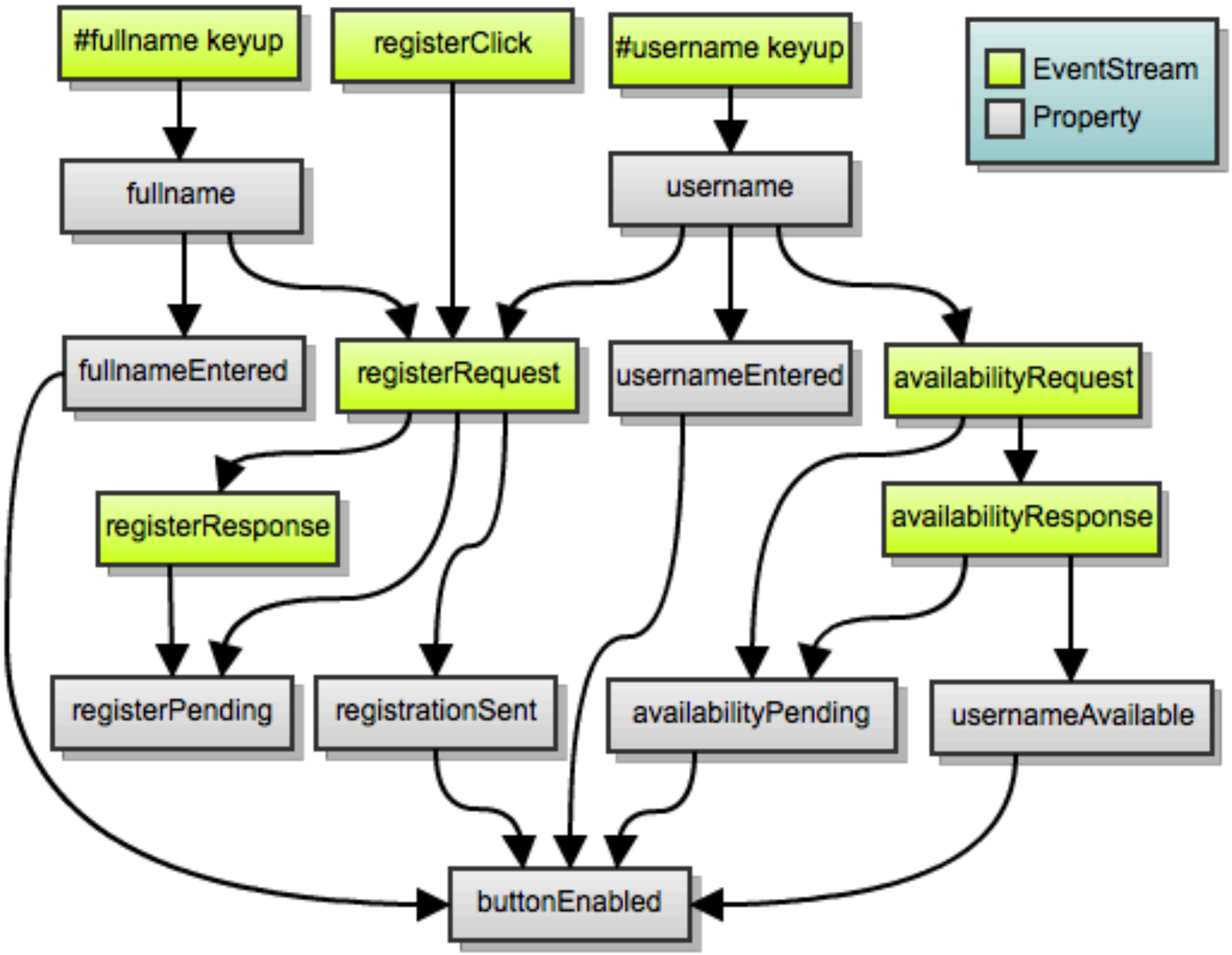
Ready 100%

CLOUDFLARE

# models - Materialized data



# models - Signals



# Pure FRP is useless

- Weird language - (ELM  anyone?)
- Fixed signal flow
- Strictly no side-effects

# Dirty FRP is awesome

- Weird language
  - **Python**
- Fixed signal flow
  - **Attacks come and go, but patterns fixed**
- Strictly no side-effects
  - **Dynamic "subscriptions", but idempotent**

# Prior art - Trellis

PEAK

PYTHON ENTERPRISE APPLICATION KIT

**Trellis**

UserPreferences

The PEAK Developers' Center | FrontPage | RecentChanges | TitleIndex | WordIndex | SiteNavigation | HelpContents

## Event-Driven Programming The Easy Way, with `peak.events.trellis`

(NOTE: As of 0.7a1, many new features have been added to the Trellis API, and some old ones have been deprecated. If you are upgrading from an older version, please see the porting guide for details.)

Whether it's an application server or a desktop application, any sufficiently complex system is event-driven -- and that usually means callbacks.

Unfortunately, explicit callback management is to event-driven programming what explicit memory management is to most other kinds of programming: a tedious hassle and a significant source of unnecessary bugs.

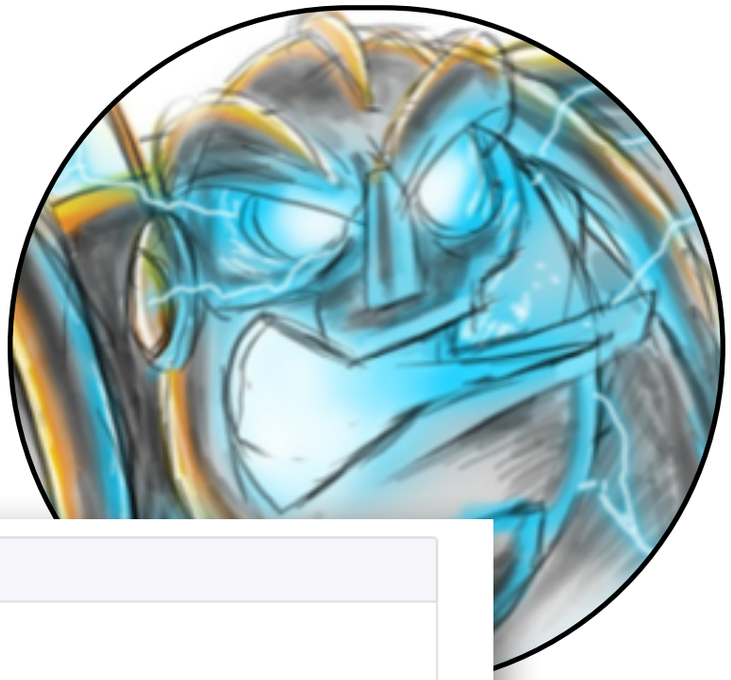
# Software Transactional Memory (STM) And Observers

The Trellis is built on a simple Python STM (Software Transactional Memory) and "Observer Pattern" implementation. This document specifies how that implementation works, and tests it.

You should read this document if you plan to implement your own Trellis cell types or other custom data structures, or if you just want to know how things work "under the hood".

**Table of Contents**

# Gatelogic!



README.md

## Gatelogic - somewhat reactive programming framework

Gatelogic is a functional reactive programming / software transactional memory inspired framework.

In classic FRP programming model, the signal graph is defined on the start and doesn't change during the lifetime of a program. For our usage we needed something more dynamic. We also wanted to do express the computable code in an easy to understand Python, as opposed to a classic composition of pure functions.

The idea is to write pretty simple, side-effect free functions that describe some business logic, while allowing them to dynamically "subscribe" arbitrary signal sources.

For example, a basic "action" code could look like:

```
def action(row):
    if confirm_hub.get('all_mitigations_enabled').value != 'True':
        return None
```

This will actually "subscribe" to `confirm_hub/all_mitigations_enabled` input signal. Note, that this subscription is *declared* in the code, as opposed to predefined as input. Change of that input signal value will trigger recomputation of all dependent action functions.

**https://github.com/cloudflare/gatelogic**

CLOUDFLARE

# Gatelogic

- Input - ReadableHub
  - update(full_data)
- Processing - ComputableHub
  - maintain(key, function)
  - unmaintain
- Subscriptions - QueryHub
  - update(full_data)

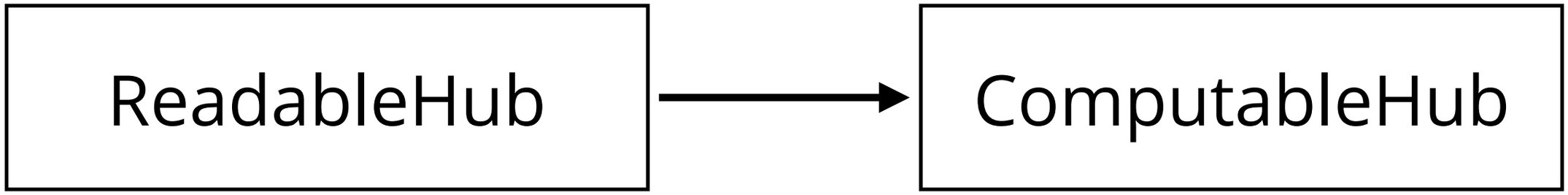
# Input data - a dict

```
{
    '00001': {ip:'1.2.3.4', port: 80, domain: 'bar.com',
    '00002': {ip:'1.2.3.5', port: 80, domain: 'foo.com',
    ...
}
```

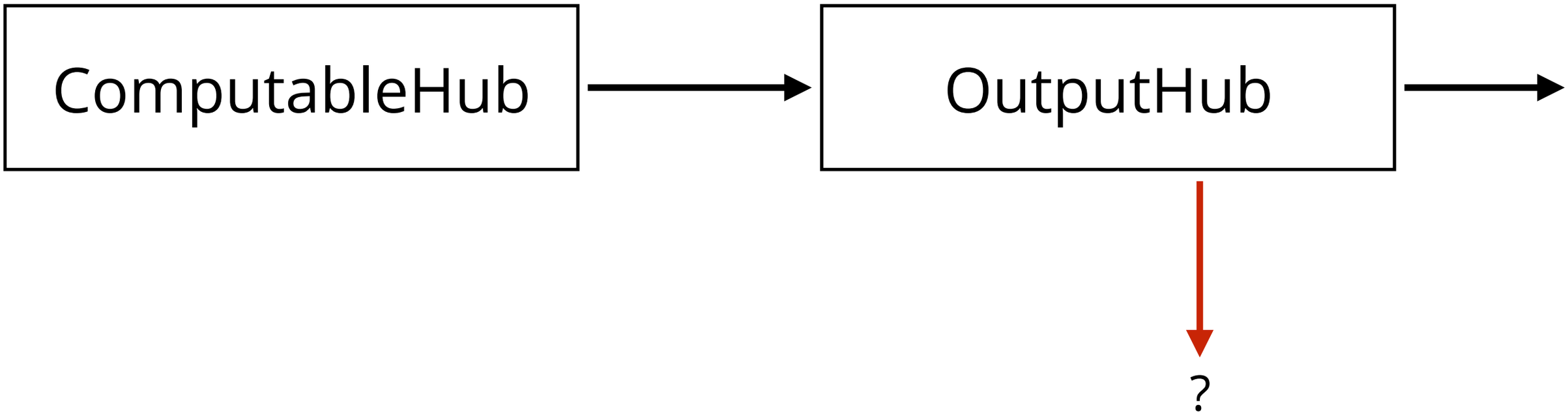
update()

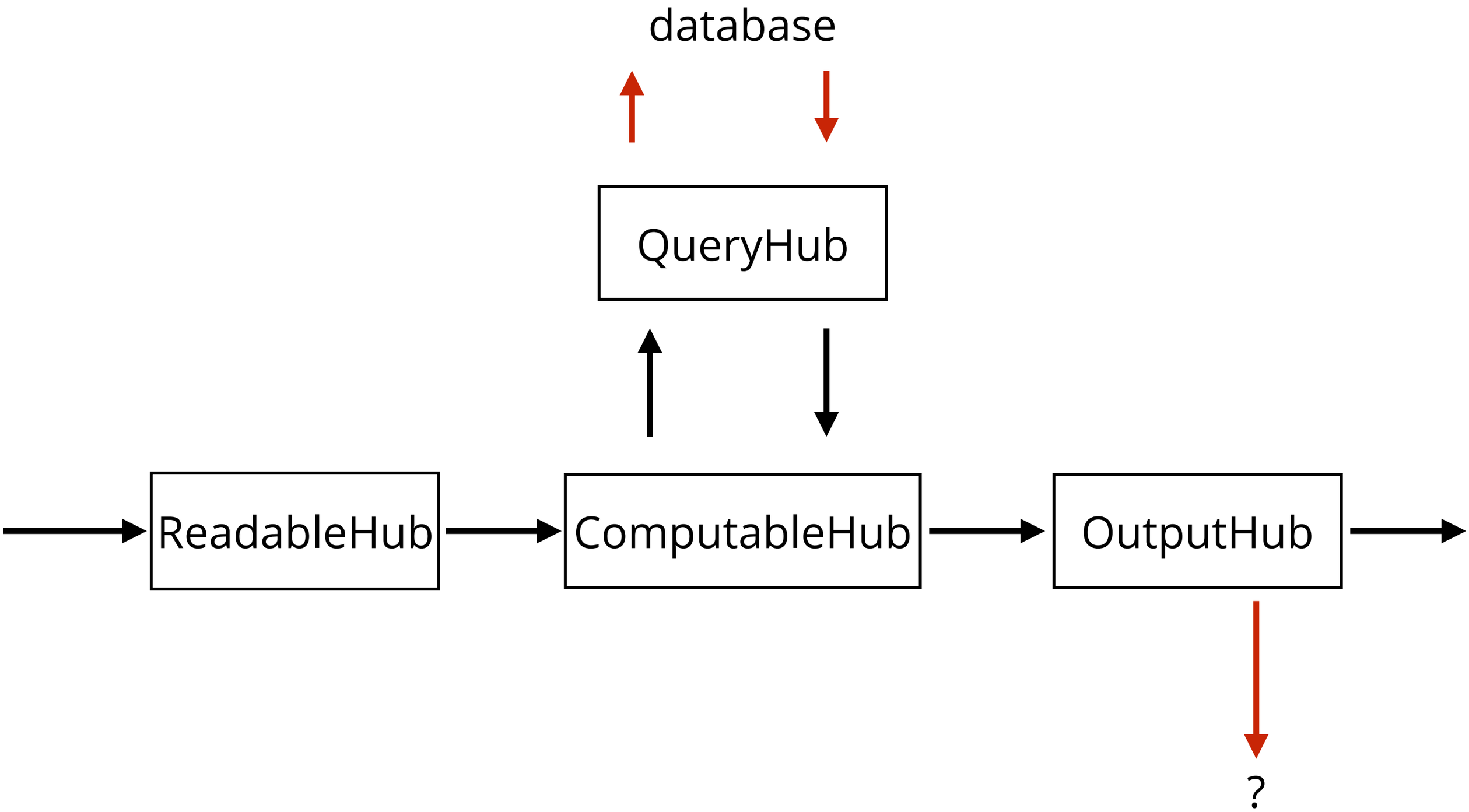
ReadableHub

```
update

{
    'attack1': 'example.com',
...
}
```

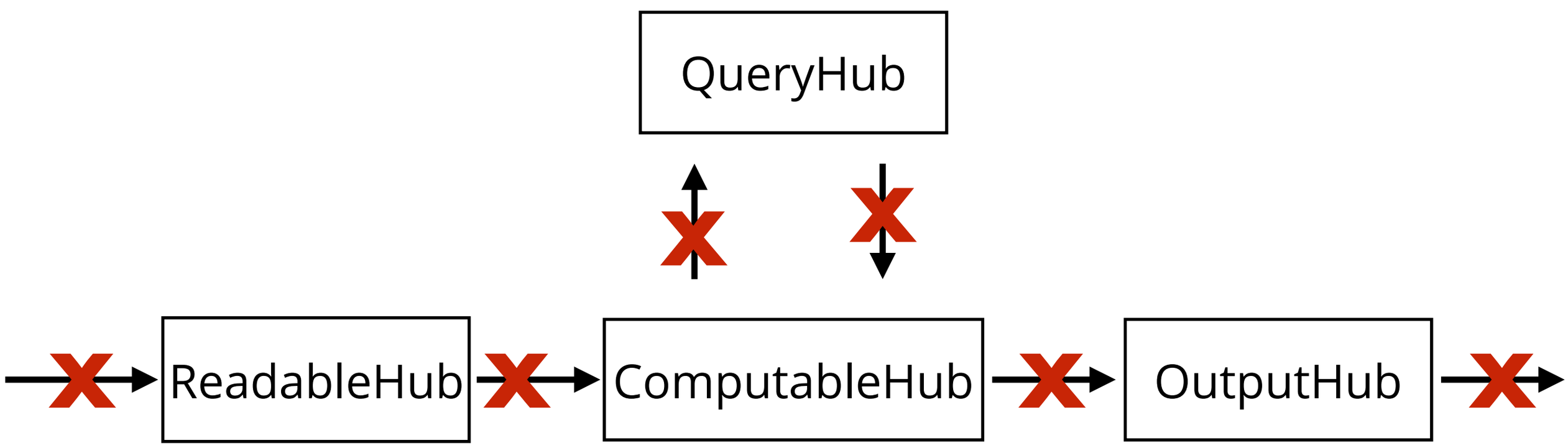



```
def on_hook(_, kind, k, row):
    if kind == 'add':
        mitigations.maintain(k, action, row)

    if kind == 'delete':
        mitigations.unmaintain(k)

subscribe(readable, on_hook)

def action(row):
   return None
```


ComputableHub
OutputHub
?


database
QueryHub
ReadableHub
ComputableHub
OutputHub
?

# Materialized



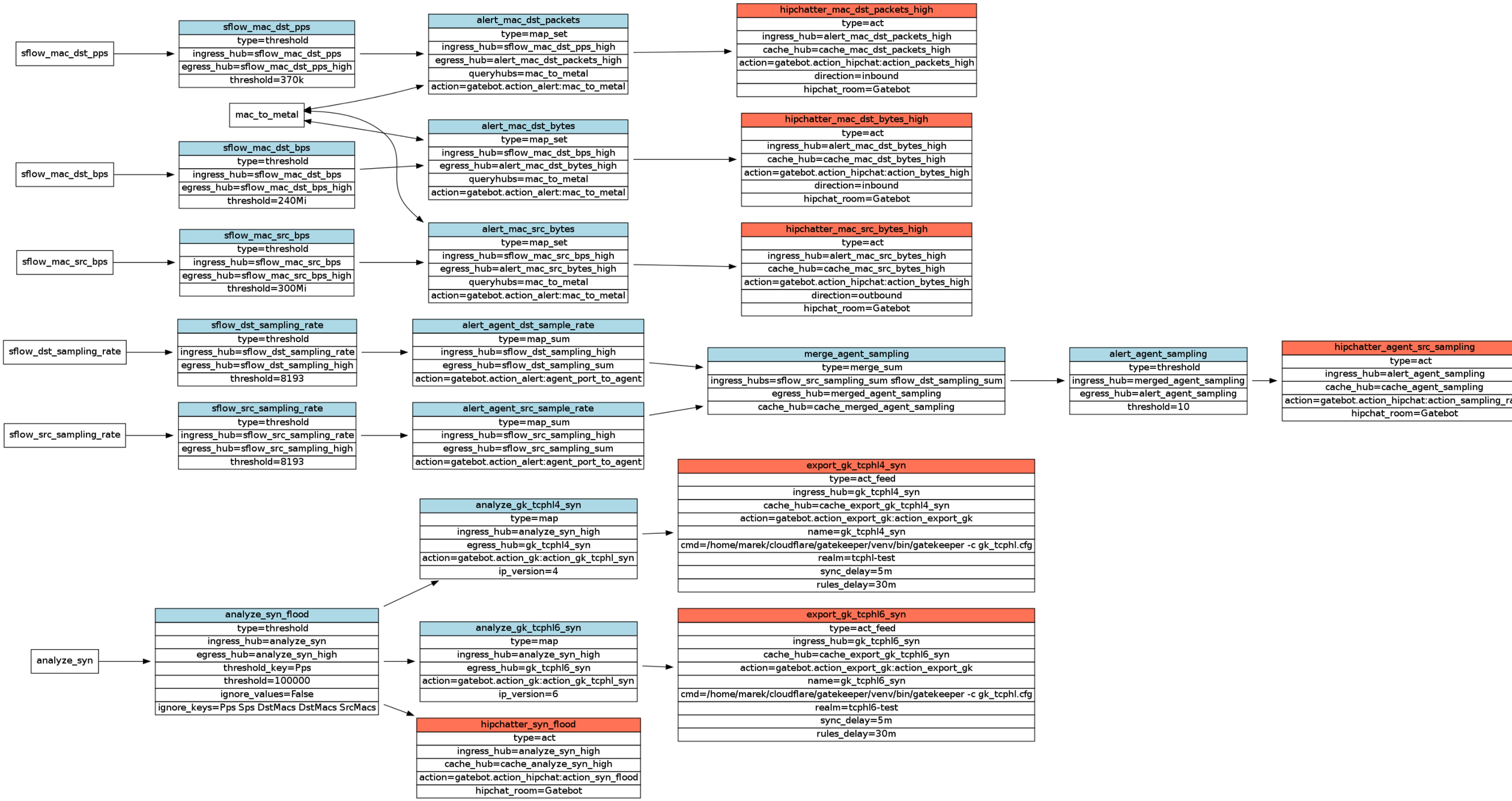
```
def action(row, plan_hub, subdomain_hub, toggle_hub):
    domain = row.value
    if not domain:
        return None

    if toggle_hub.get('all_mitigations_disabled').value != 'True':
        return None

    qps = 100
    if plan_hub.get(domain).value in ('business', 'b'):
        qps = 500

    sd = (subdomain_hub.get(domain).value or '').split(' ')

    mitigation = \
                domain + \
                ' --qps=%s ' % qps + \
                ' '.join('--except=%s' % s for s in sd)

    return mitigation
```

CLOUDFLARE

# It works!

- Solid foundation!
- Composable!
- Scalable
- Maintainable
- But:
  - no event loop
  - lacks higher-order abstractions

CLOUDFLARE


sflow_mac_dst_pps
sflow_mac_dst_pps
type=threshold
ingress_hub=sflow_mac_dst_pps
egress_hub=sflow_mac_dst_pps_high
threshold=370k
alert_mac_dst_packets
type=map_set
ingress_hub=sflow_mac_dst_pps_high
egress_hub=alert_mac_dst_packets_high
queryhubs=mac_to_metal
action=gatebot.action_alert:mac_to_metal
hipchatter_mac_dst_packets_high
type=act
ingress_hub=alert_mac_dst_packets_high
cache_hub=cache_mac_dst_packets_high
action=gatebot.action_hipchat:action_packets_high
direction=inbound
hipchat_room=Gatebot
mac_to_metal
sflow_mac_dst_bps
sflow_mac_dst_bps
type=threshold
ingress_hub=sflow_mac_dst_bps
egress_hub=sflow_mac_dst_bps_high
threshold=240Mi
alert_mac_dst_bytes
type=map_set
ingress_hub=sflow_mac_dst_bps_high
egress_hub=alert_mac_dst_bytes_high
queryhubs=mac_to_metal
action=gatebot.action_alert:mac_to_metal
hipchatter_mac_dst_bytes_high
type=act
ingress_hub=alert_mac_dst_bytes_high
cache_hub=cache_mac_dst_bytes_high
action=gatebot.action_hipchat:action_bytes_high
direction=inbound
hipchat_room=Gatebot
sflow_mac_src_bps
sflow_mac_src_bps
type=threshold
ingress_hub=sflow_mac_src_bps
egress_hub=sflow_mac_src_bps_high
threshold=300Mi
alert_mac_src_bytes
type=map_set
ingress_hub=sflow_mac_src_bps_high
egress_hub=alert_mac_src_bytes_high
queryhubs=mac_to_metal
action=gatebot.action_alert:mac_to_metal
hipchatter_mac_src_bytes_high
type=act
ingress_hub=alert_mac_src_bytes_high
cache_hub=cache_mac_src_bytes_high
action=gatebot.action_hipchat:action_bytes_high
direction=outbound
hipchat_room=Gatebot
sflow_dst_sampling_rate
sflow_dst_sampling_rate
type=threshold
ingress_hub=sflow_dst_sampling_rate
egress_hub=sflow_dst_sampling_high
threshold=8193
alert_agent_dst_sample_rate
type=map_sum
ingress_hub=sflow_dst_sampling_high
egress_hub=sflow_dst_sampling_sum
action=gatebot.action_alert:agent_port_to_agent
merge_agent_sampling
type=merge_sum
ingress_hubs=sflow_src_sampling_sum sflow_dst_sampling_sum
egress_hub=merged_agent_sampling
cache_hub=cache_merged_agent_sampling
alert_agent_sampling
type=threshold
ingress_hub=merged_agent_sampling
egress_hub=alert_agent_sampling
threshold=10
hipchatter_agent_src_sampling
type=act
ingress_hub=alert_agent_sampling
cache_hub=cache_agent_sampling
action=gatebot.action_hipchat:action_sampling_ra
hipchat_room=Gatebot
sflow_src_sampling_rate
sflow_src_sampling_rate
type=threshold
ingress_hub=sflow_src_sampling_rate
egress_hub=sflow_src_sampling_high
threshold=8193
alert_agent_src_sample_rate
type=map_sum
ingress_hub=sflow_src_sampling_high
egress_hub=sflow_src_sampling_sum
action=gatebot.action_alert:agent_port_to_agent
analyze_gk_tcphl4_syn
type=map
ingress_hub=analyze_syn_high
egress_hub=gk_tcphl4_syn
action=gatebot.action_gk:action_gk_tcphl_syn
ip_version=4
export_gk_tcphl4_syn
type=act_feed
ingress_hub=gk_tcphl4_syn
cache_hub=cache_export_gk_tcphl4_syn
action=gatebot.action_export_gk:action_export_gk
name=gk_tcphl4_syn
cmd=/home/marek/cloudflare/gatekeeper/venv/bin/gatekeeper -c gk_tcphl.cfg
realm=tcphl-test
sync_delay=5m
rules_delay=30m
analyze_syn
analyze_syn_flood
type=threshold
ingress_hub=analyze_syn
egress_hub=analyze_syn_high
threshold_key=Pps
threshold=100000
ignore_values=False
ignore_keys=Pps Sps DstMacs DstMacs SrcMacs
analyze_gk_tcphl6_syn
type=map
ingress_hub=analyze_syn_high
egress_hub=gk_tcphl6_syn
action=gatebot.action_gk:action_gk_tcphl_syn
ip_version=6
export_gk_tcphl6_syn
type=act_feed
ingress_hub=gk_tcphl6_syn
cache_hub=cache_export_gk_tcphl6_syn
action=gatebot.action_export_gk:action_export_gk
name=gk_tcphl6_syn
cmd=/home/marek/cloudflare/gatekeeper/venv/bin/gatekeeper -c gk_tcphl.cfg
realm=tcphl6-test
sync_delay=5m
rules_delay=30m
hipchatter_syn_flood
type=act
ingress_hub=analyze_syn_high
cache_hub=cache_analyze_syn_high
action=gatebot.action_hipchat:action_syn_flood
hipchat_room=Gatebot

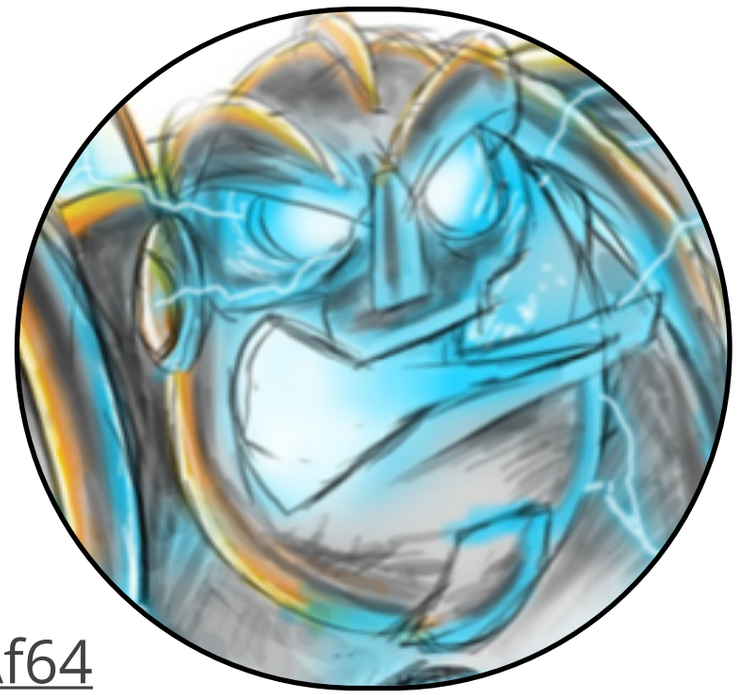
# 204 loc

```
marek@ubuntu-saucy:~/cloudflare/gatelogic/gatelogic$ cloc *py
       3 text files.
       3 unique files.
       0 files ignored.

http://cloc.sourceforge.net v 1.60  T=0.01 s (240.8 files/s, 23197.8 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
Python                           3             61             24            204
-------------------------------------------------------------------------------
SUM:                             3             61             24            204
-------------------------------------------------------------------------------
```

CLOUDFLARE

# Thanks!

- FRP is grea
  - http://www.flapjax-lang.org/
  - https://www.youtube.com/watch?v=mEvo6TVAf64
  - https://www.youtube.com/watch?v=Agu6jipKfYw



**https://github.com/cloudflare/gatelogic**

marek@cloudflare.com

@majek04

CLOUDFLARE