

Skogul

Kristian Lyngstøl, Fornebu, Juni 2019

I dag

- 10 millioner målepunkter
- Polles hvert 5 minutt
- (33k/s)

I "morgen"

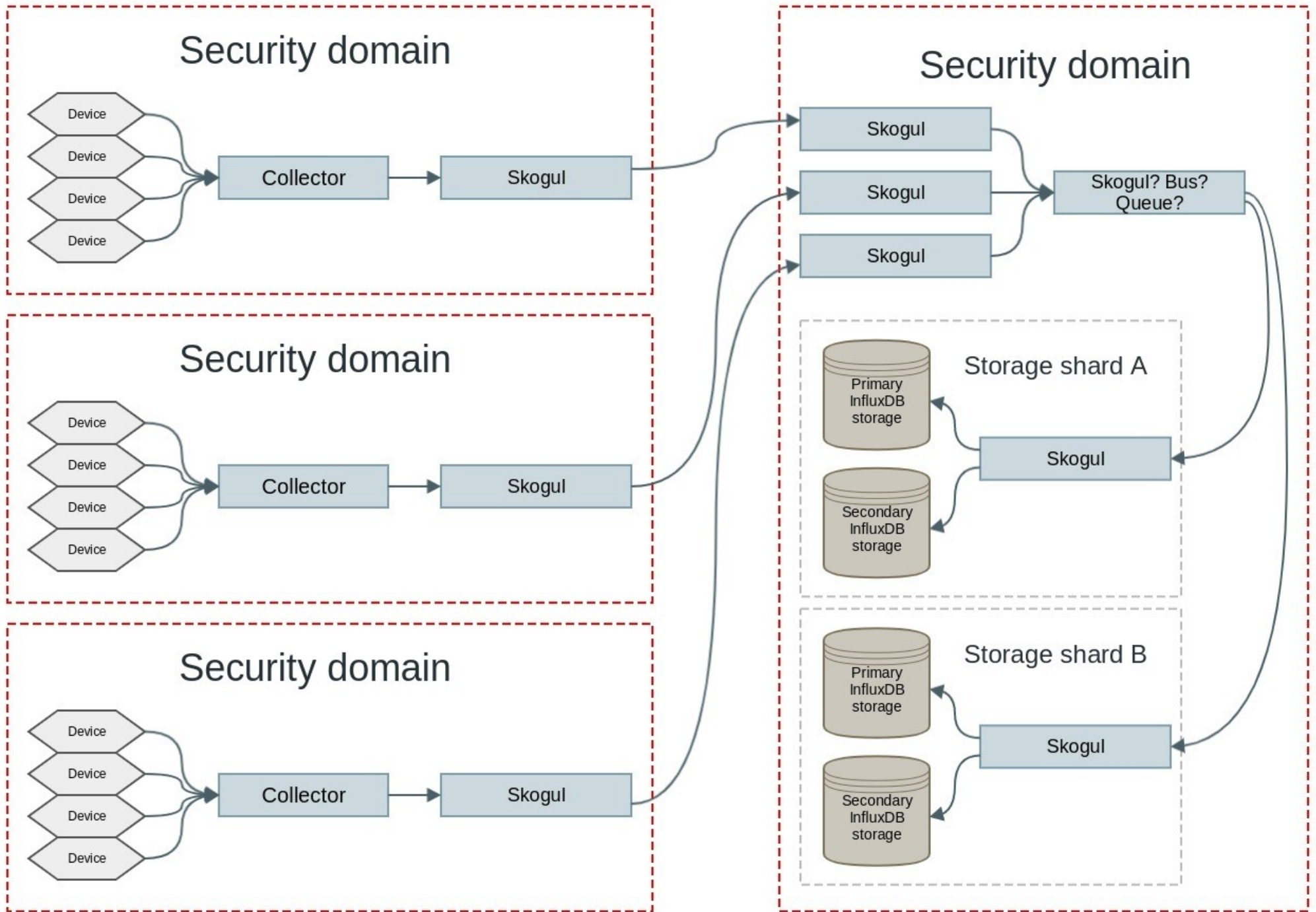
- 1 milliard målepunkter
- Oppdateres hvert 10. sekund
- (100M/s)

Og...

- Må støtte både ny og gammel plattform samtidig
- Flere forskjellige kilder
 - Streaming telemetri fra JunOS
 - BGP Monitoring Protocol
 - DNS-metrics?
 - Masse wifi-data fra massemarked
- Sikkerhetssoner
- Eksportering til eksterne

Mange ukjente

- Hva slags input-format er best egnet
- Hva slags oppdateringsfrekvens
- Storage backend? Influx? MnR? MySQL?
- Hvor skal kompleksitet knytta til robusthet ligge?
- Osv



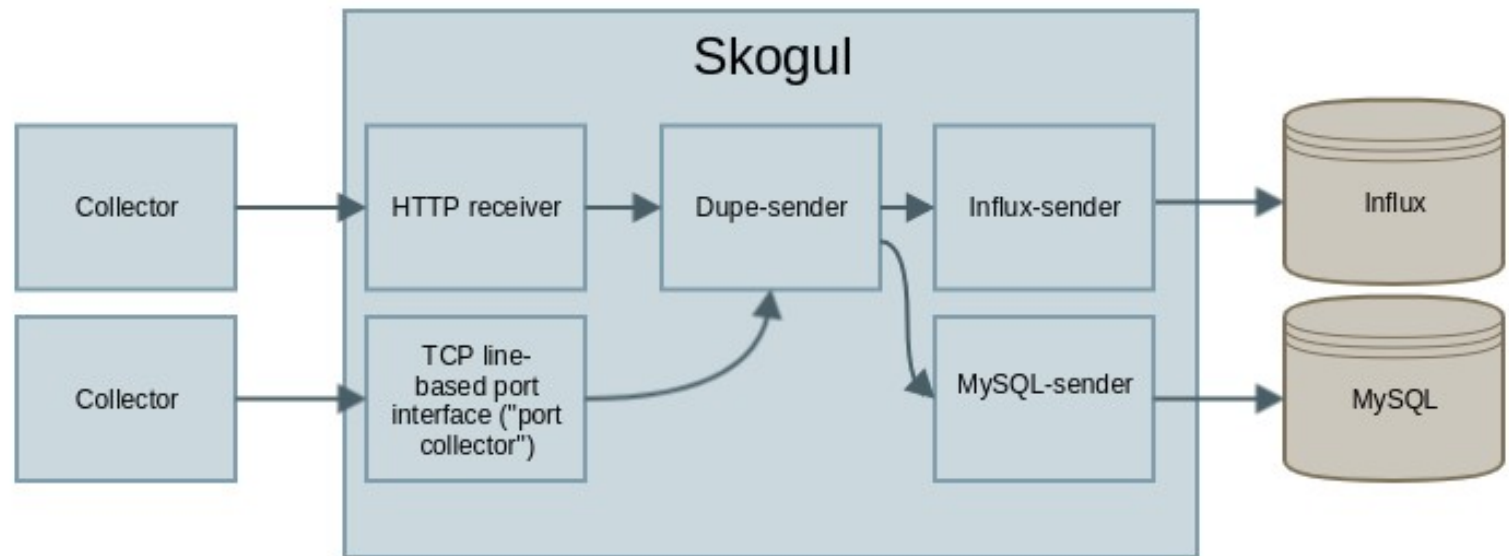
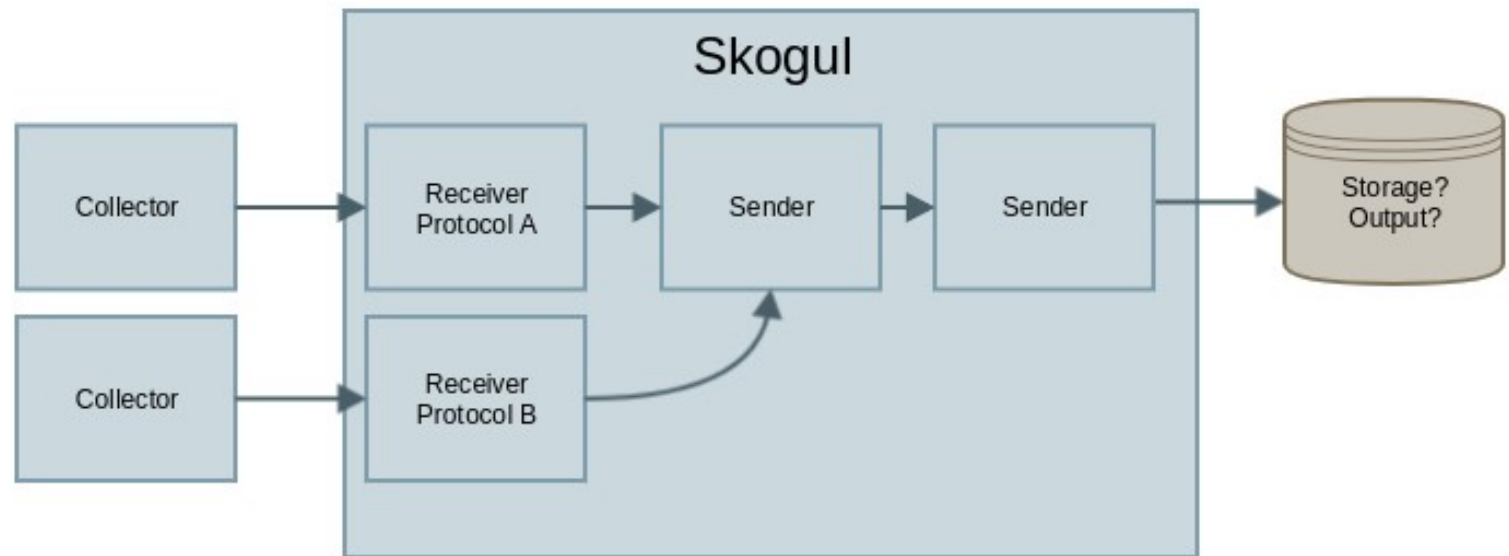
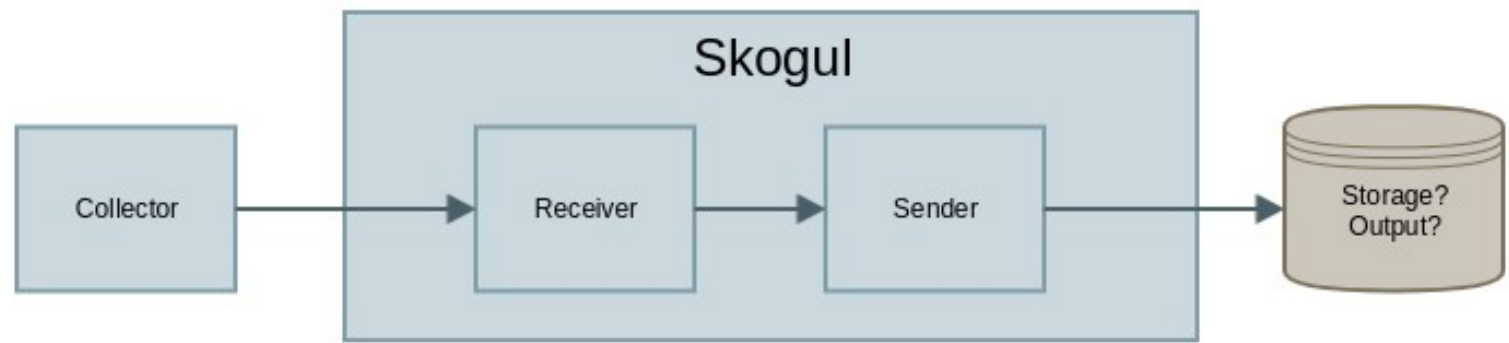
Skogul...

- Leverer piler!
- Løsriver hvordan data samles inn fra hvordan det transporteres og lagres

eller....

- Har et fleksibelt internt format
- Kan ta i mot og skrive til forskjellige Ting™ samtidig
- Er ENKEL
- Er RASK
- Et rammeverk og en cli-kommando for enkel “fra x til y”-scenarier

Design



Design

- Receiver – hvordan data “oppstår”/tas i mot.
 - HTTP, MQTT, linje-orientert TCP/fifo, stdin...
- Sender – hva man gjør videre
 - Intern: Batch, dupe, failover, retry, mm
 - Ekstern: Influxdb, HTTP, MySQL, Postgresql, MQTT, mm

Go

- Raskt
- Ganske “rent”
- Mye positivt over hele fjøla
- Litt umodent på dependencies - men ble dramatisk bedre med “go moduler”
- Testing, dokumentasjon, testbare eksempler, osv

Go - samtidighet

- Ting er samtidig “by default”.
- Ikke det samme som multithreading – men nesten
 - Hver “go routine” kjører hver for seg, men ikke sikkert at det er en egen tråd. Noe man ikke (skal) tenke på.
- Alle biblioteker som er verdt noe kan brukes fra flere go routines samtidig.
- Go routines? Go routiner? Gårutiner?
 - Trenger norsk språkråd her....

skogul-x2y

- Binær som støtter de fleste skogul receivere og sendere
- Bør dekke 90% av use-casene

```
$ skogul-x2y -sender influx://db4:8086/foo/bar
```

```
$ skogul-x2y -sender influx://db4:8086/foo/bar -batch 1000
```

```
$ skogul-x2y -receiver tcp://[::1]:1234 -sender mnr://mor:2101
```

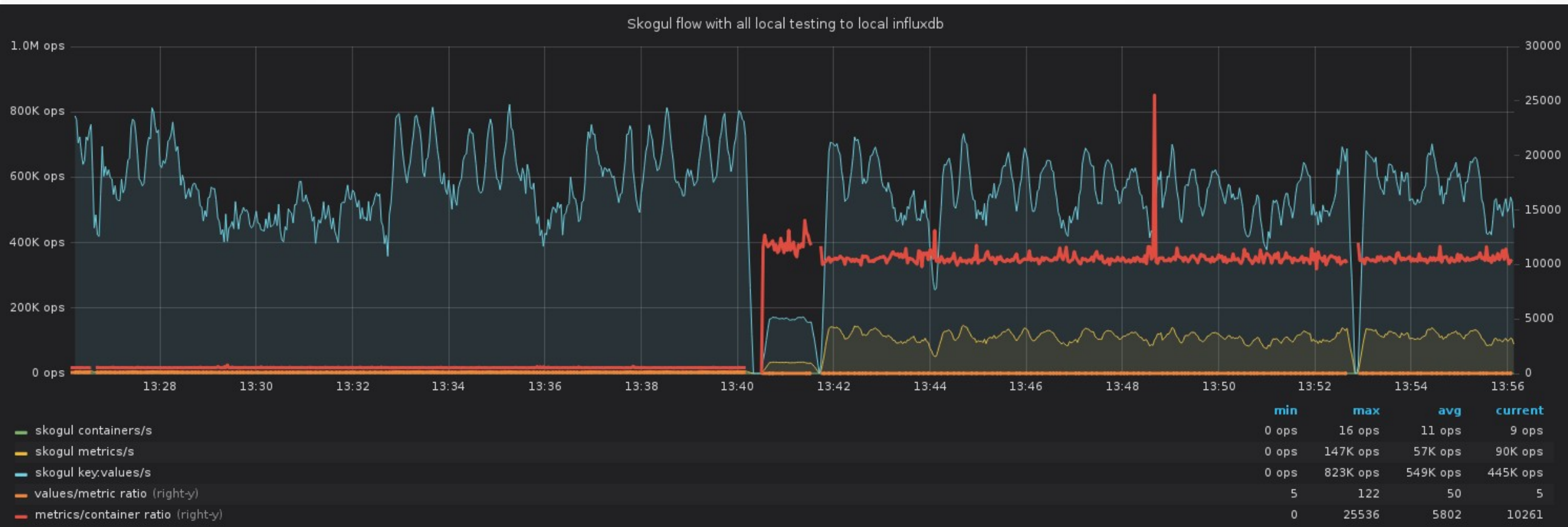
Mer kompliserte kjeder...

- Skriv ny binær som bruker skogul-rammeverket
- Foreløpig nødvendig for mer avanserte kjeder, f.eks. skriving til flere databaser samtidig
- Blir muligens mindre viktig over tid - vi får se
- Skogul er lagd for at dette skal være stuerent

Ytelse?

- Cirka 1M målepunkter per sekund på en laptop
- Første forsøk ga cirka 3-4mill/s til disk på server

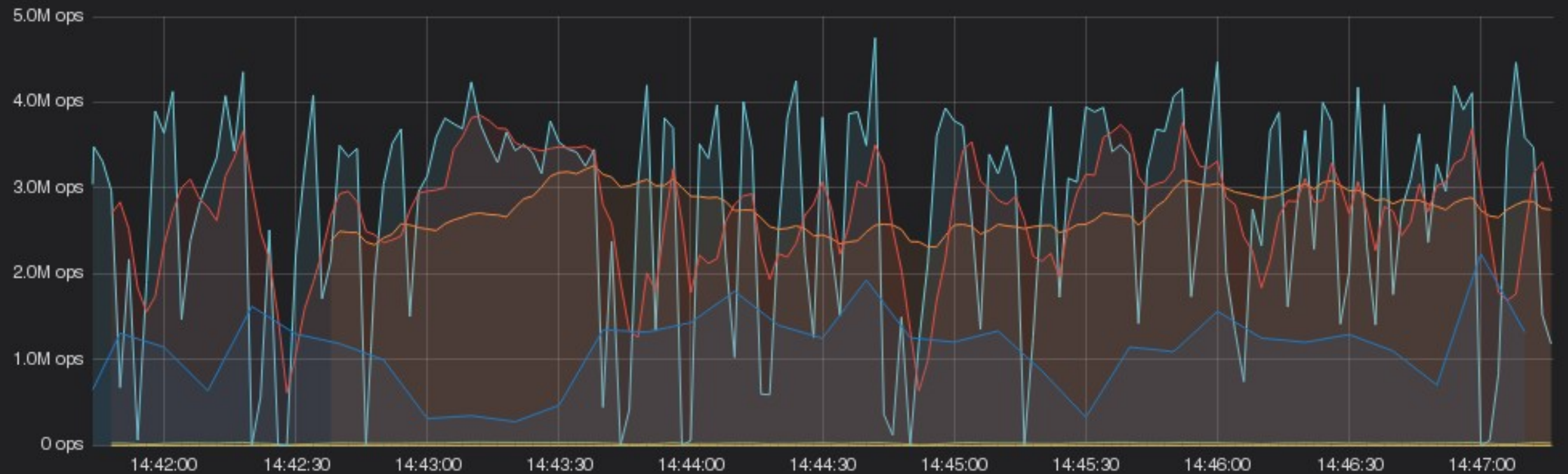
Veldig tidlig test (laptop)



Plaging av influx...

Skogul operating rates

timeshift -5s



	min	max	avg	current
metric rate	6.16K ops	38.49K ops	27.27K ops	28.48K ops
container rate	12.10 ops	76.30 ops	54.00 ops	56.30 ops
value rate raw	0 ops	4.75M ops	2.71M ops	1.19M ops
value rate 30	2.31M ops	3.26M ops	2.75M ops	2.75M ops
value rate 5	615.84K ops	3.85M ops	2.73M ops	2.85M ops
value std deviation by 10s	259.46K ops	2.24M ops	1.13M ops	1.32M ops

Testing

```
kly@r2d2:~/src/skogul$ go test -test.short ./... -cover
ok      github.com/KristianLyng/skogul      (cached)          coverage: 59.6% of statements
?      github.com/KristianLyng/skogul/cmd/skogul-x2y  [no test files]
?      github.com/KristianLyng/skogul/internal/mqtt   [no test files]
ok      github.com/KristianLyng/skogul/parser  (cached)          coverage: 100.0% of statements
ok      github.com/KristianLyng/skogul/receiver (cached)          coverage: 59.4% of statements
ok      github.com/KristianLyng/skogul/sender  (cached)          coverage: 53.0% of statements
ok      github.com/KristianLyng/skogul/transformer (cached)          coverage: 100.0% of statements
kly@r2d2:~/src/skogul$
```

Test – ytelse og mer

```
kly@r2d2:~$ skogul-x2y -receiver test:// -sender null:// -count debug://
2019/06/12 09:33:31 No Period set for Counter-sender. Using 1 second intervals.
2019/06/12 09:33:32 Debug:
{
  "metrics": [
    {
      "timestamp": "2019-06-12T09:33:32.760050821+02:00",
      "metadata": {
        "skogul": "counter"
      },
      "data": {
        "rate_containers": 9366,
        "rate_metrics": 93666,
        "rate_values": 4683318,
        "total_containers": 9367,
        "total_metrics": 93670,
        "total_values": 4683500
      }
    }
  ]
}
^C
kly@r2d2:~$
```

Testing

- Unit tester – grei coverage, enkelt å skrive
- Test-receiver for generering av dummy-data...
- Sendere for testing
 - null, delay, debug, count (stats), mm
- Bruk en skogul-x2y til å lage data, send til en annen over \$PROTO, som skriver til disk.

Veien videre

- Kapacitor, influx og aggregater
- Industrialisere deployments
- Stor sentral DB? Flere? Osv.
 - Mange ideer. Må testes. Skogul er kun ett verktøy.
- Flere senders og receivers etter behov – og parsere. (BMP, f.eks)