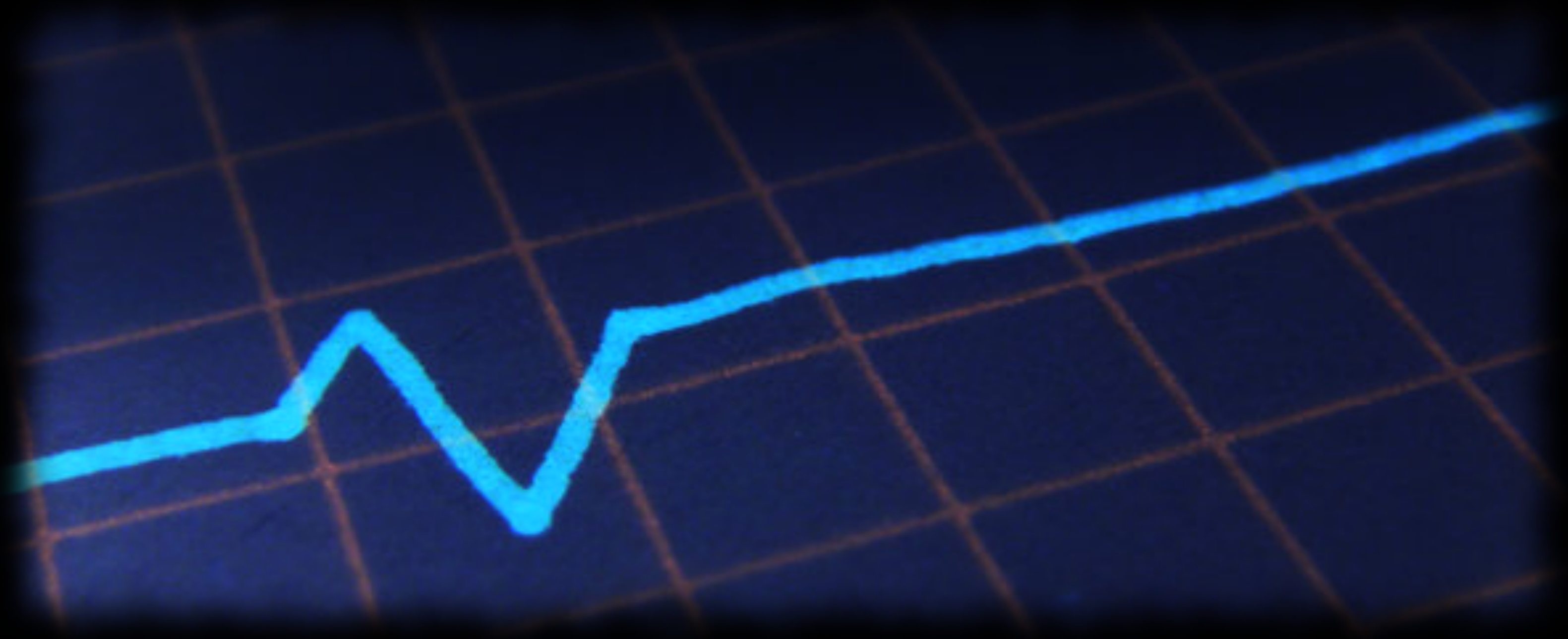
OpenTSDB

A Distributed, Scalable, Time Series Database

“Monitoring at an unprecedented level of granularity”



Benoît “tsuna” Sigoure
tsuna@stumbleupon.com

StumbleUpon

Why Yet Another Monitoring System?

- Distributed storage of monitoring data
- No Single Point of Failure
- Pulling custom graphs must be trivial & fast
- Scale to:
  - Thousands of machines
  - Many billions of data points

Existing Monitoring Systems Are Old And Rusty

Copyright 2010 - Richard Barber

Old Plow

Pinpoint The Problems

Finding needle in haystack by Bindaas Madhavi

HBase

Distributed

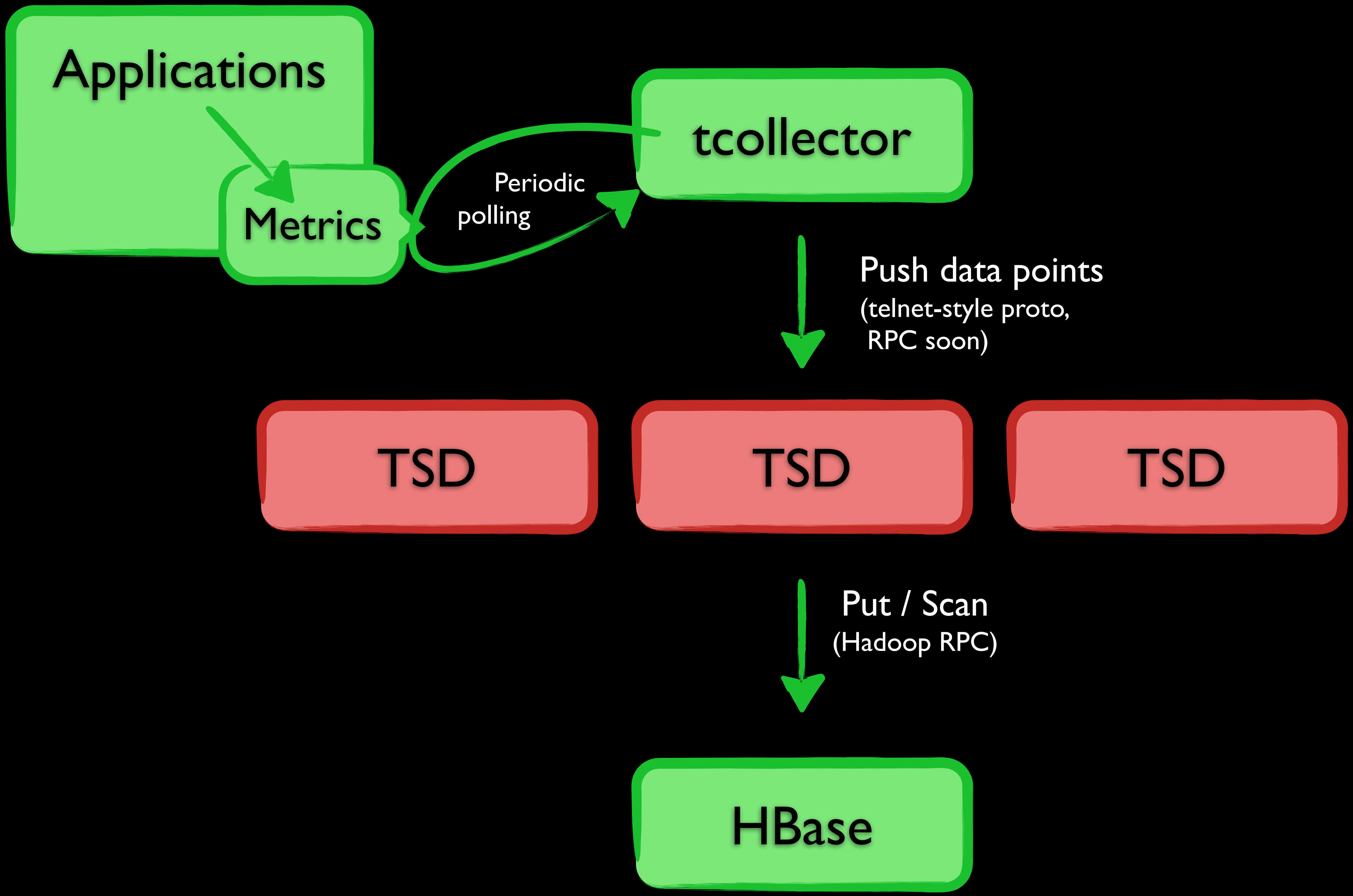
Scalable

Reliable

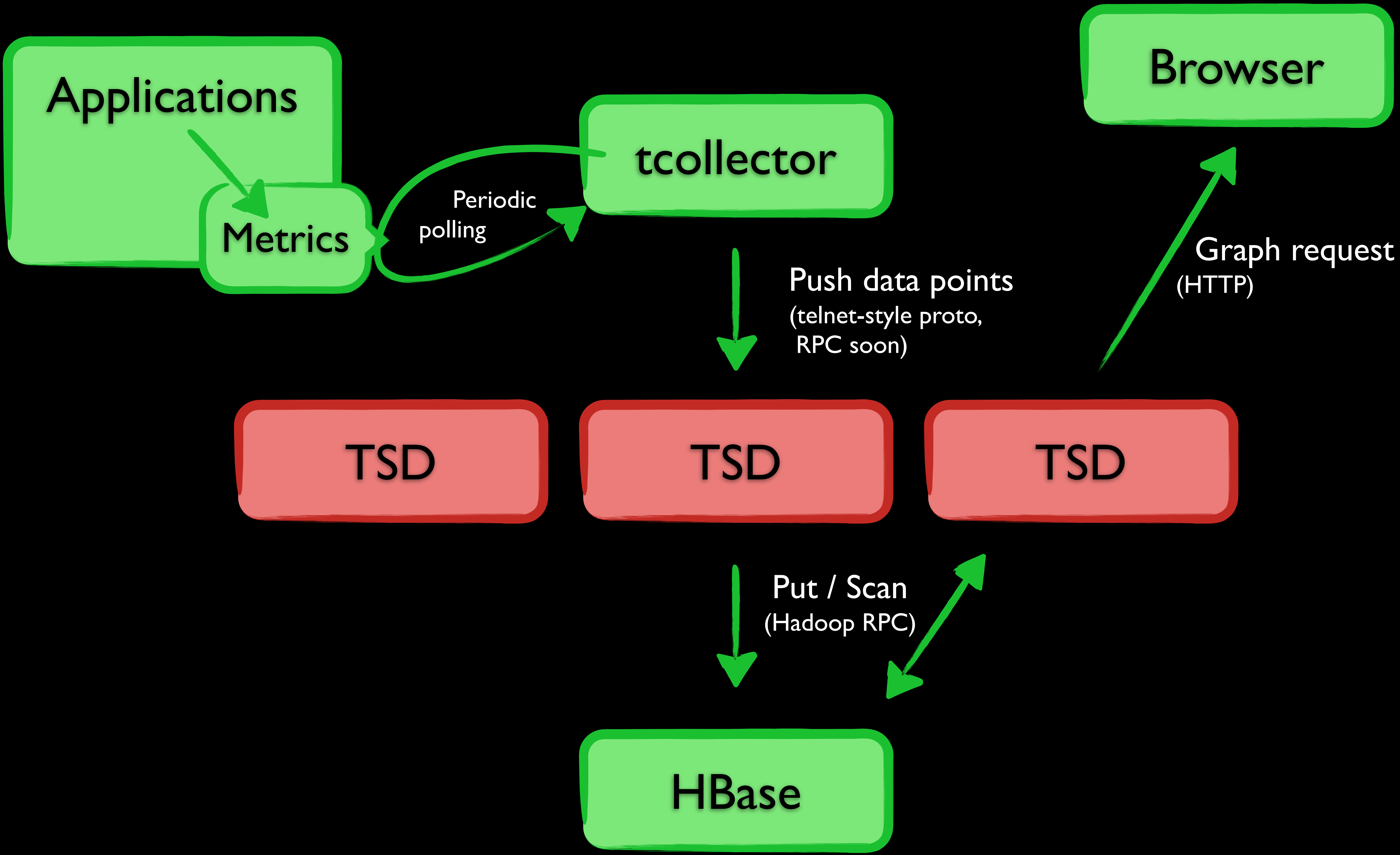
Efficient

Bright Ideas by purplemattfish

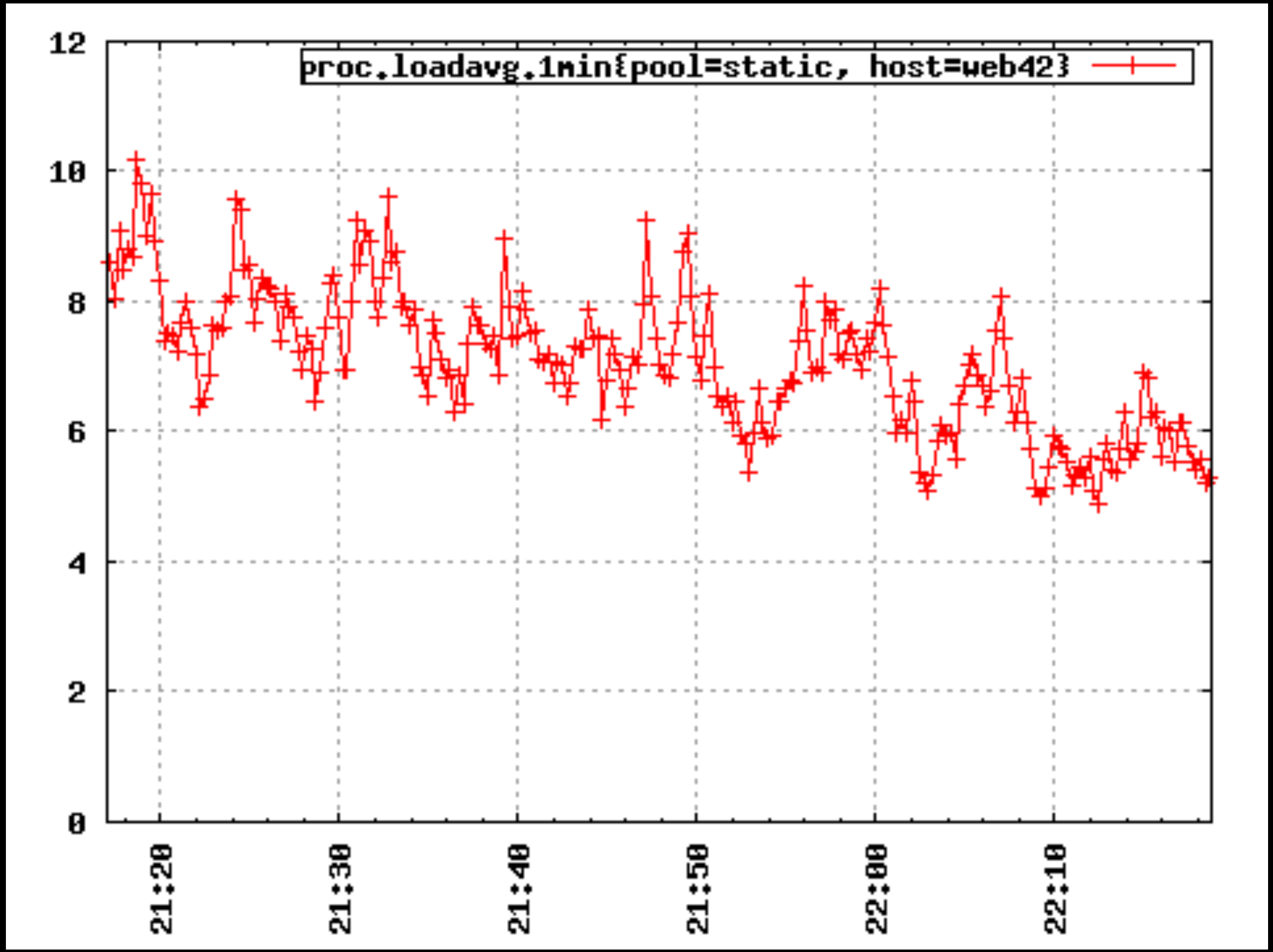
The Big Picture™



The Big Picture™



Key concepts

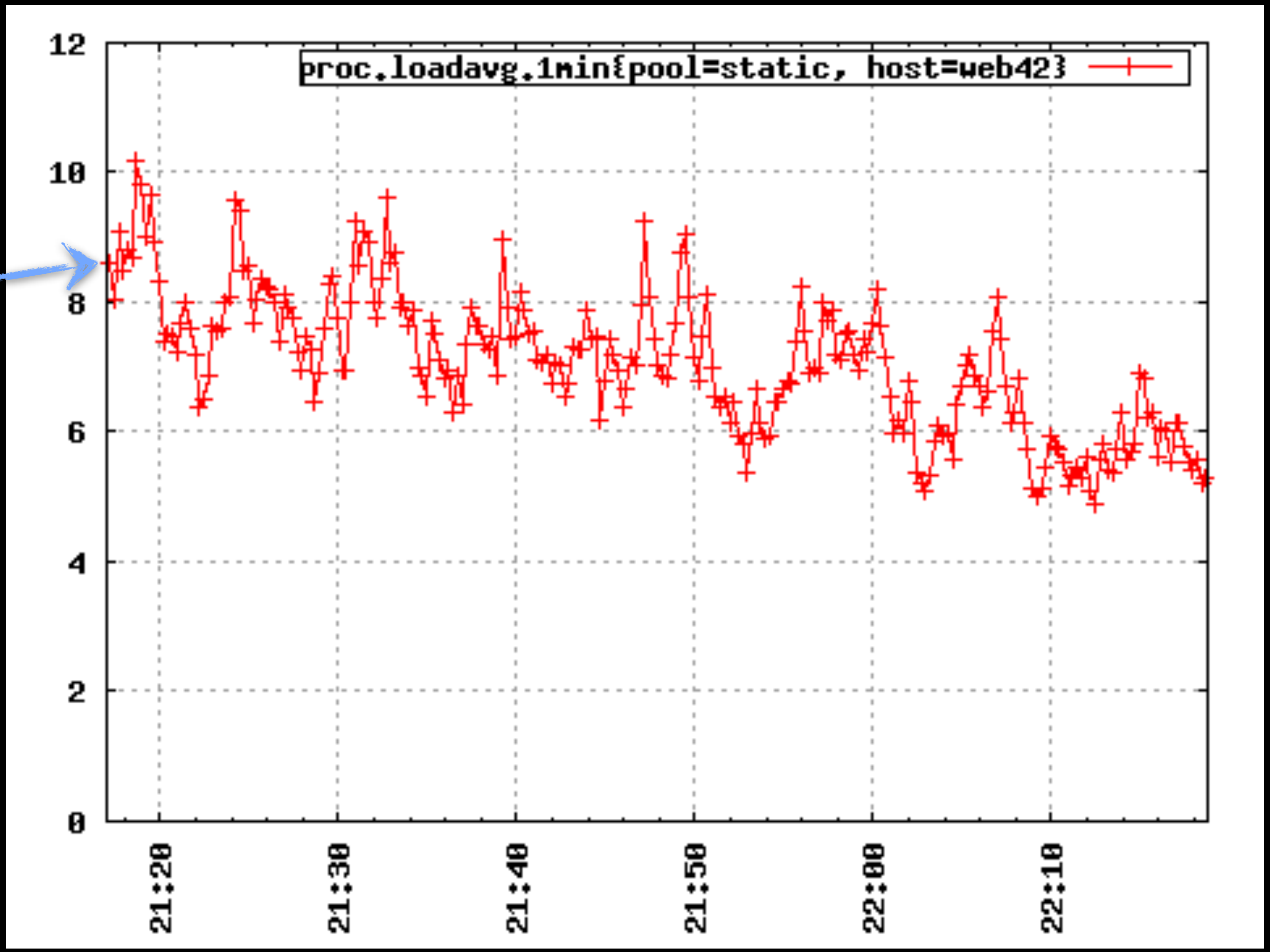


- Data Points
  (time, value)
- Metrics
  proc.loadavg.1m
- Tags
  host=web42 pool=static
- Metric + Tags = Time Series

```
put proc.loadavg.1m 1234567890 0.42 host=web42 pool=static
```

Key concepts

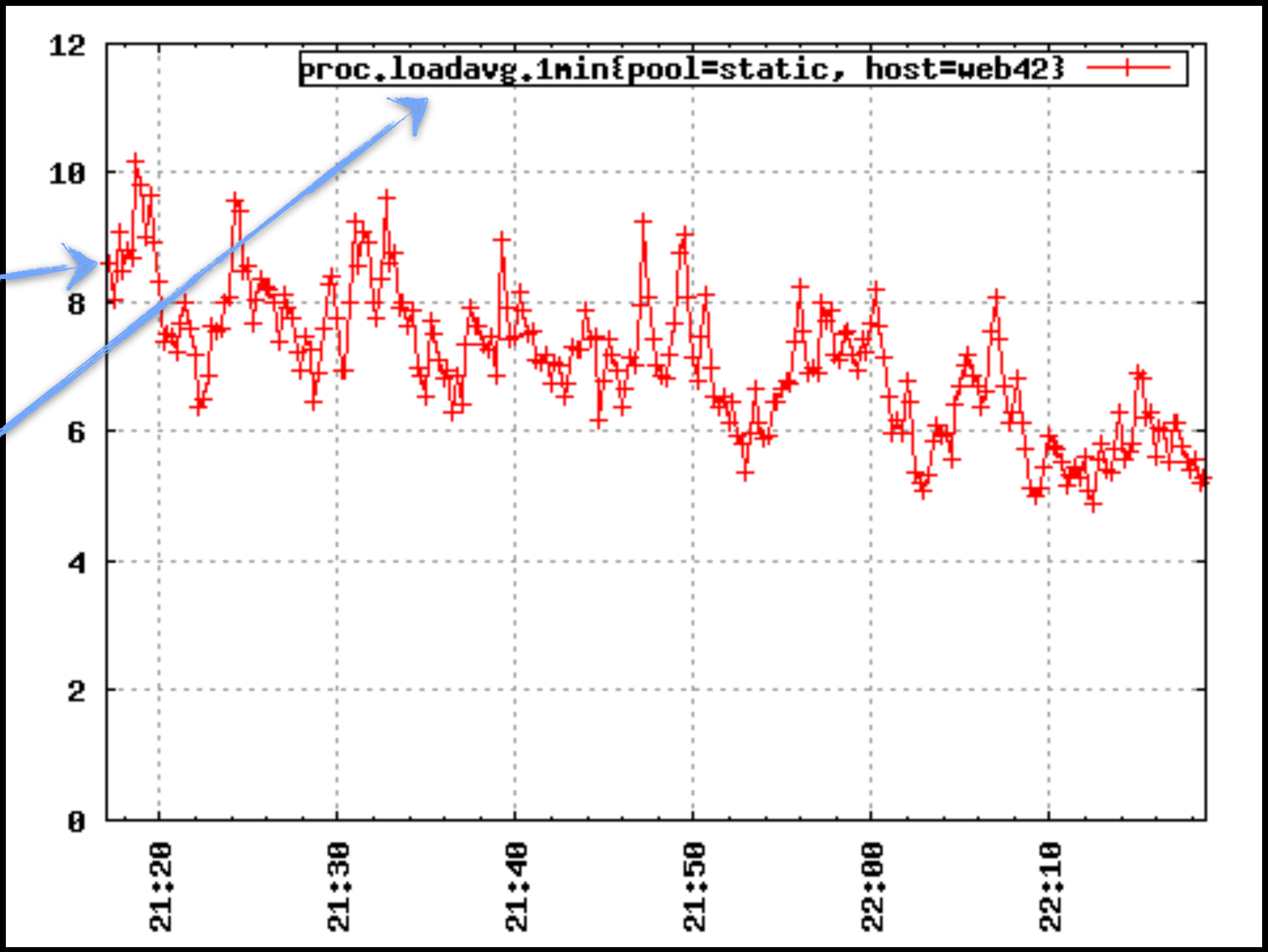
- Data Points
  `(time, value)`
- Metrics
  `proc.loadavg.1m`
- Tags
  `host=web42  pool=static`
- `Metric + Tags =` Time Series



```
put proc.loadavg.1m 1234567890 0.42 host=web42 pool=static
```

Key concepts

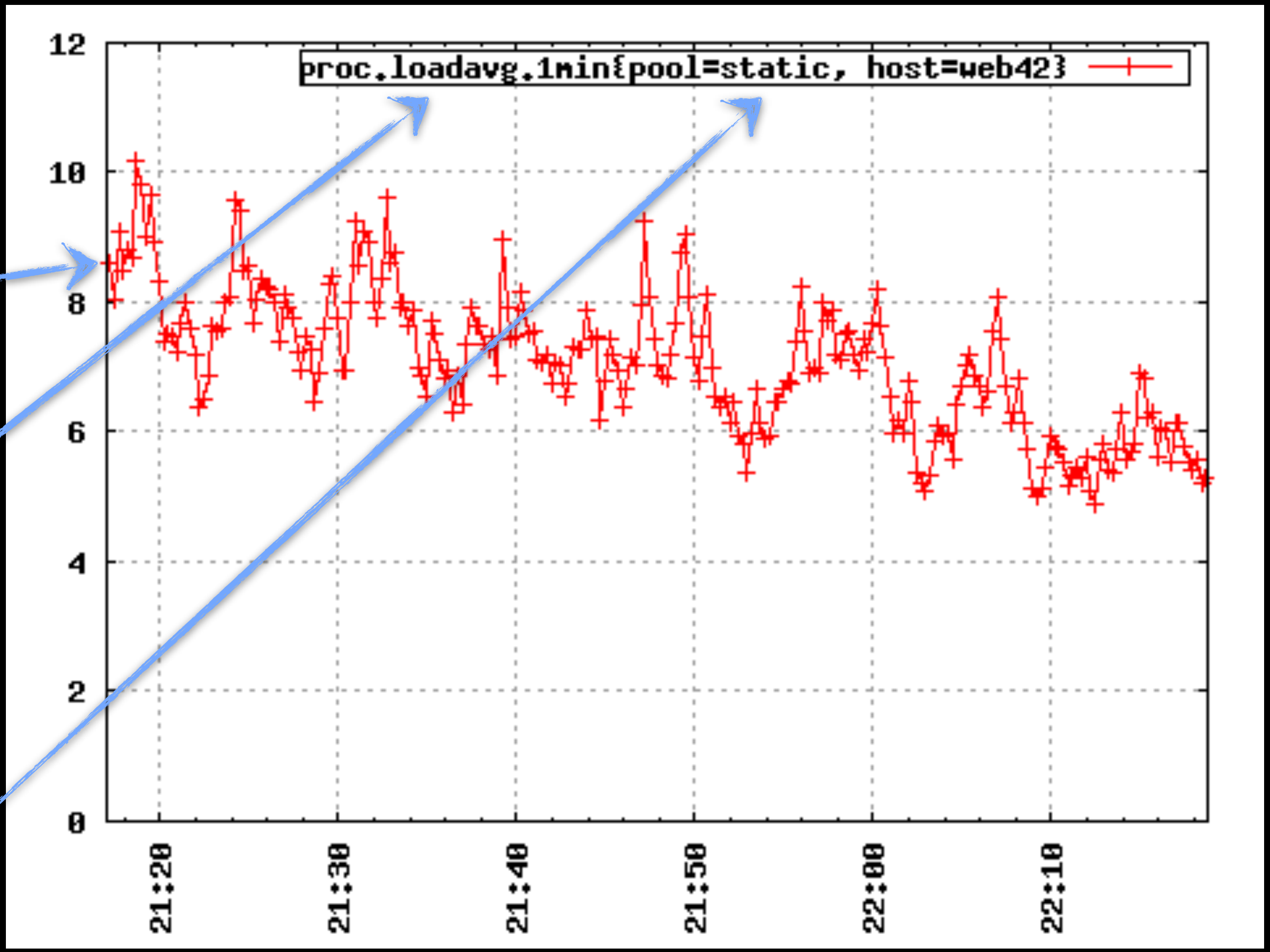
- Data Points
  (time, value)
- Metrics
  proc.loadavg.1m
- Tags
  host=web42   pool=static
- Metric + Tags = Time Series



```
put proc.loadavg.1m 1234567890 0.42 host=web42 pool=static
```

Key concepts

- Data Points
  (time, value)
- Metrics
  proc.loadavg.1m
- Tags
  host=web42   pool=static
- Metric + Tags = Time Series



```
put proc.loadavg.1m 1234567890 0.42 host=web42 pool=static
```

12 Bytes Per Datapoint

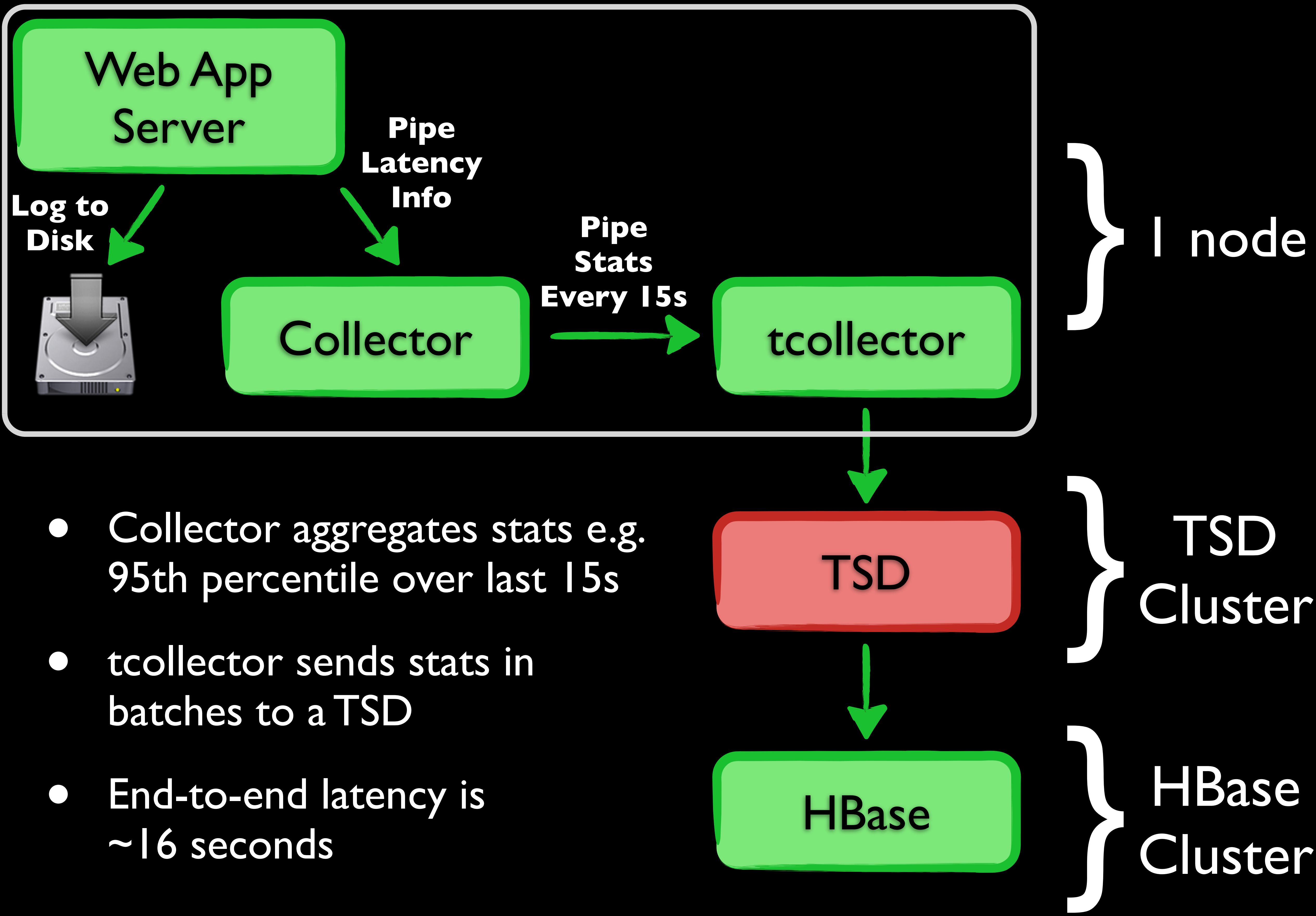
4TB per year for 1000 machines

OpenTSDB @

250 Million Datapoints/Day

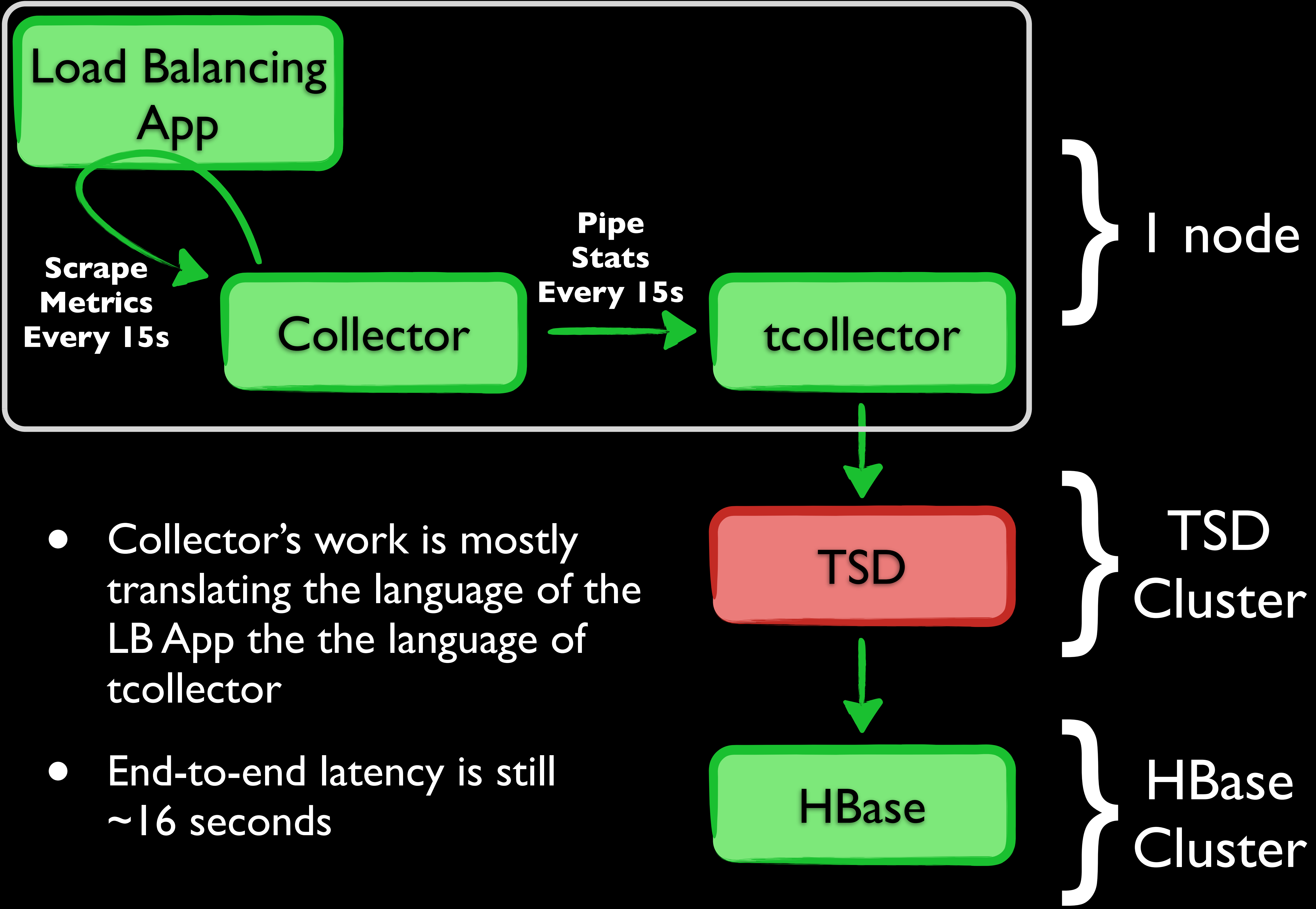
in a typical datacenter

Example: Monitoring Serving Latency



- Collector aggregates stats e.g. 95th percentile over last 15s
- tcollector sends stats in batches to a TSD
- End-to-end latency is ~16 seconds

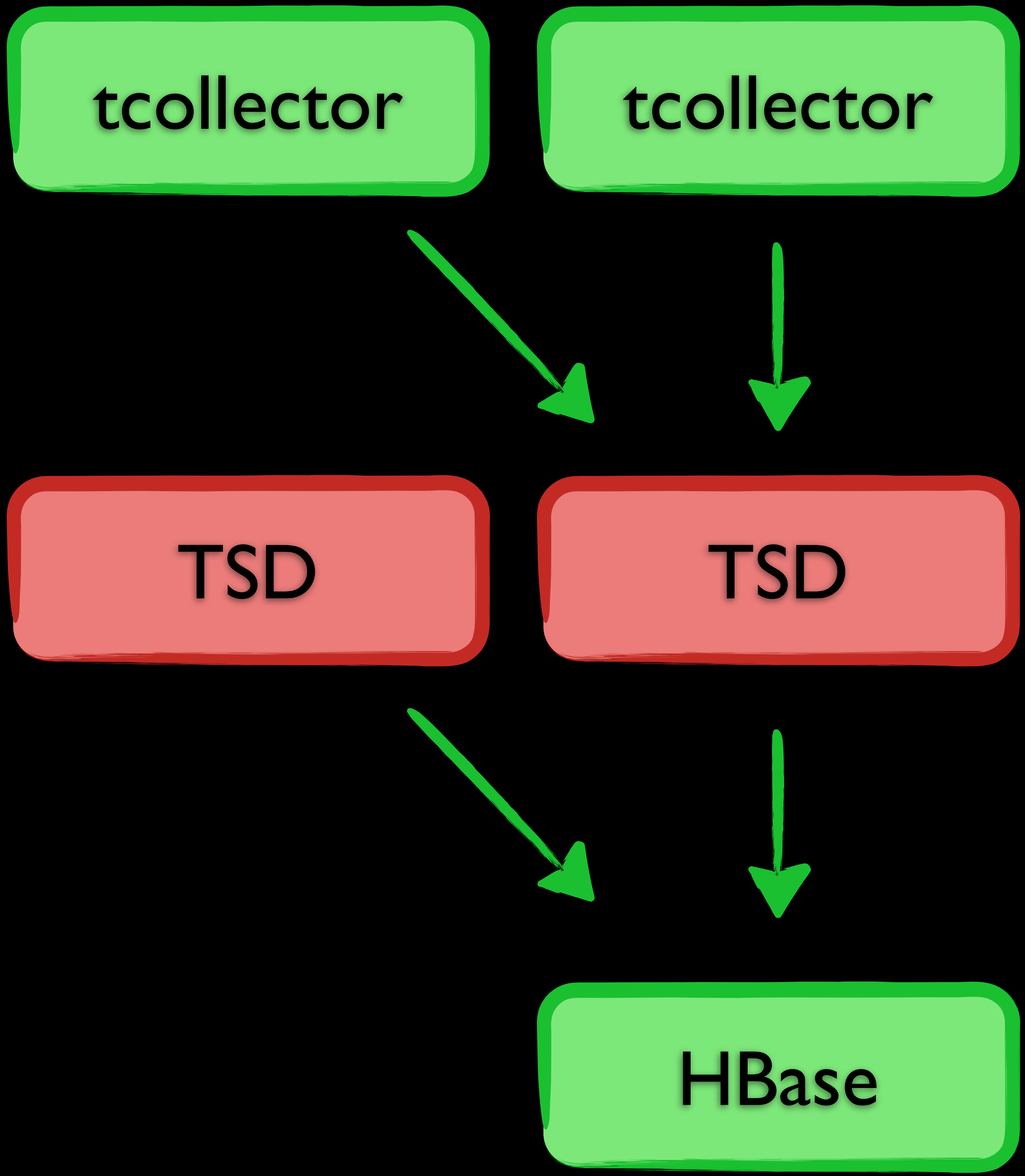
Example: Monitoring Serving Latency

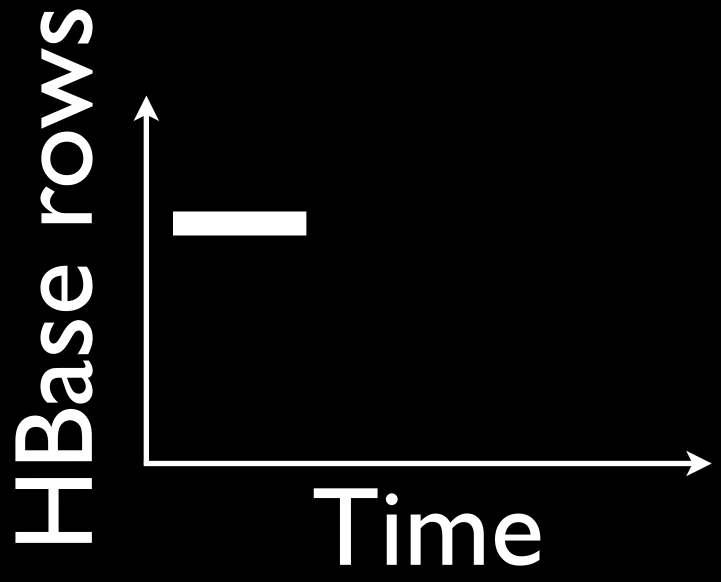


- Collector's work is mostly translating the language of the LB App the the language of tcollector
- End-to-end latency is still ~16 seconds

Problem #1

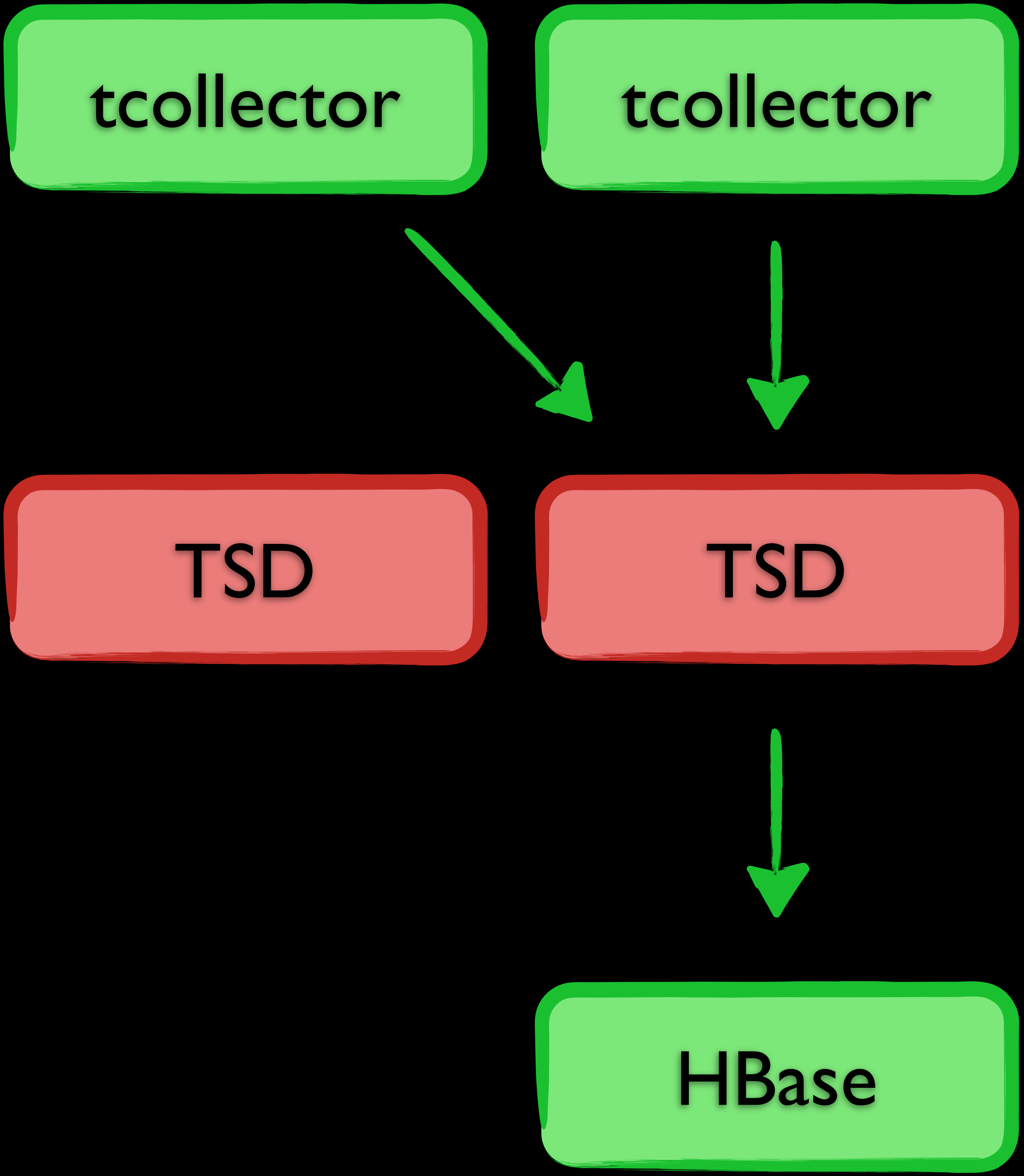
How to Synchronize Concurrent Writes (or not)

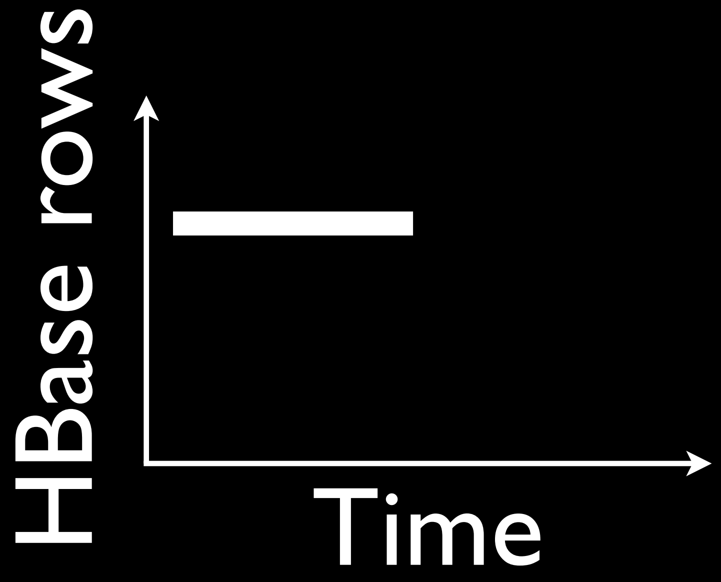




Problem #1

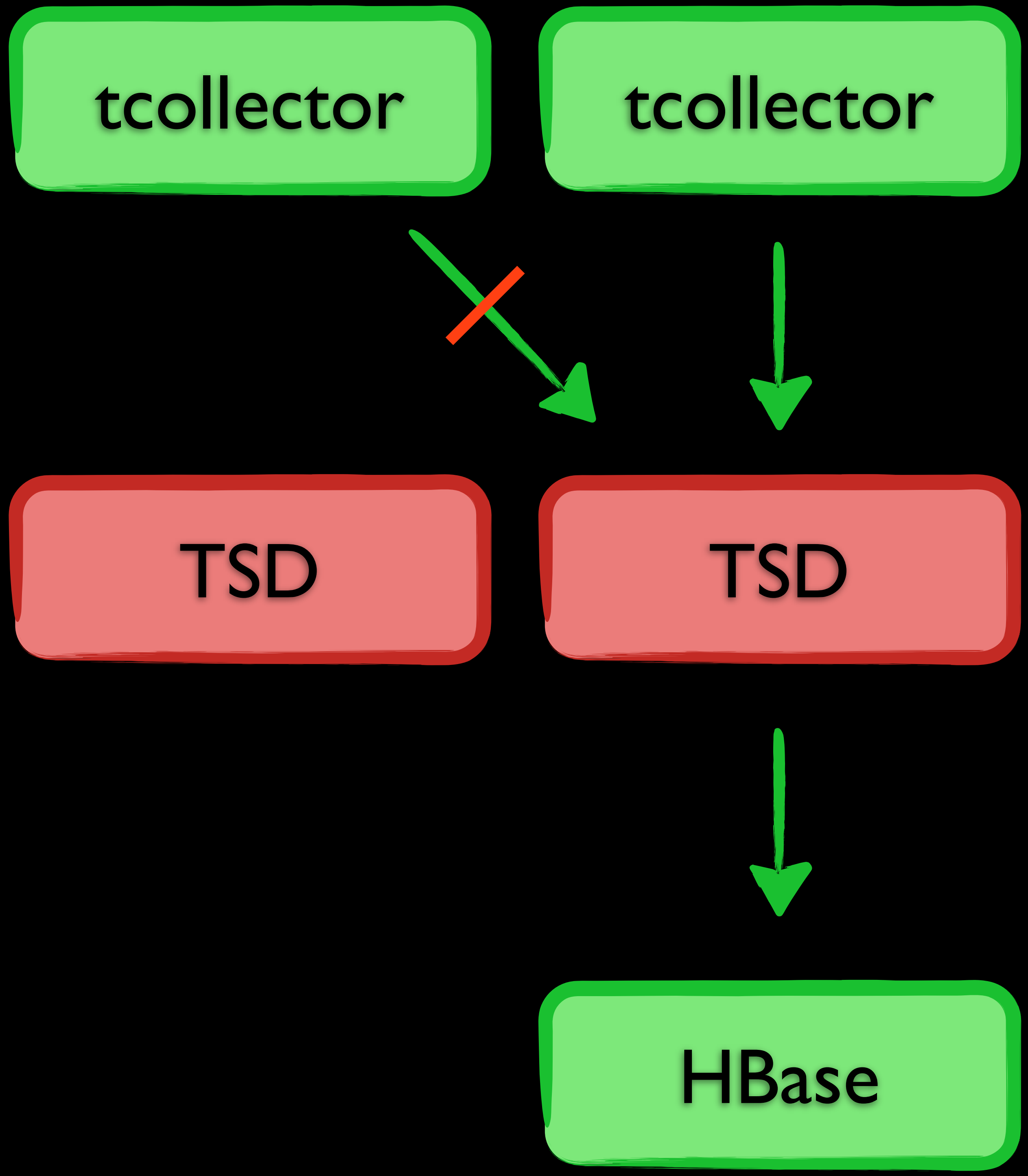
How to Synchronize Concurrent Writes (or not)





Problem #1

How to Synchronize Concurrent Writes (or not)


tcollector

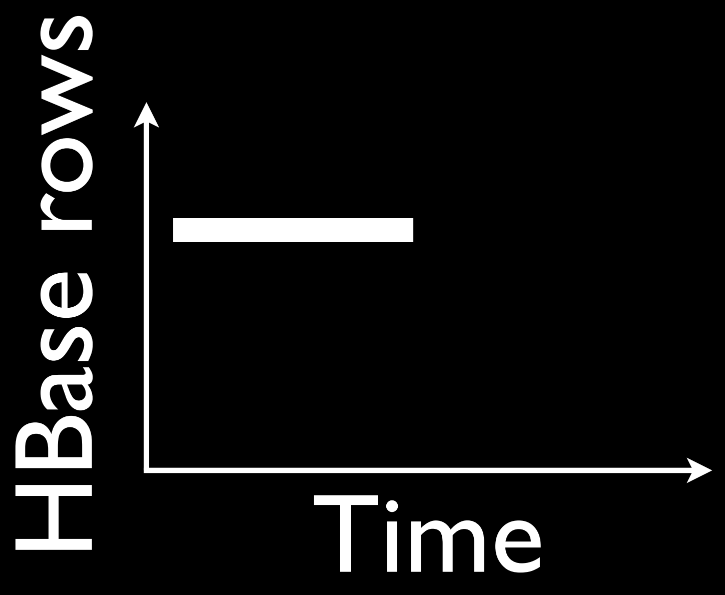
tcollector

TSD

TSD

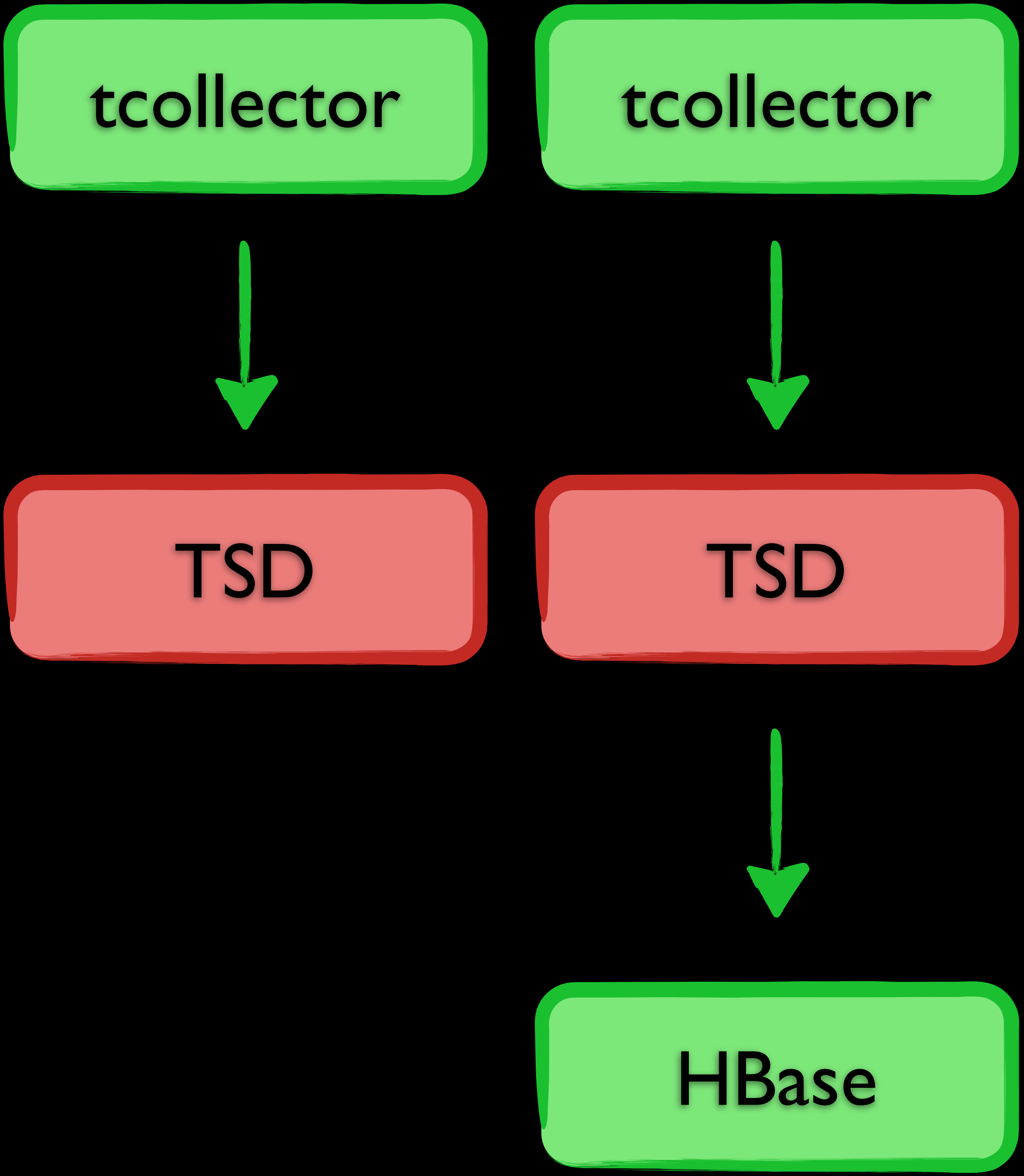
HBase

Connection Fails

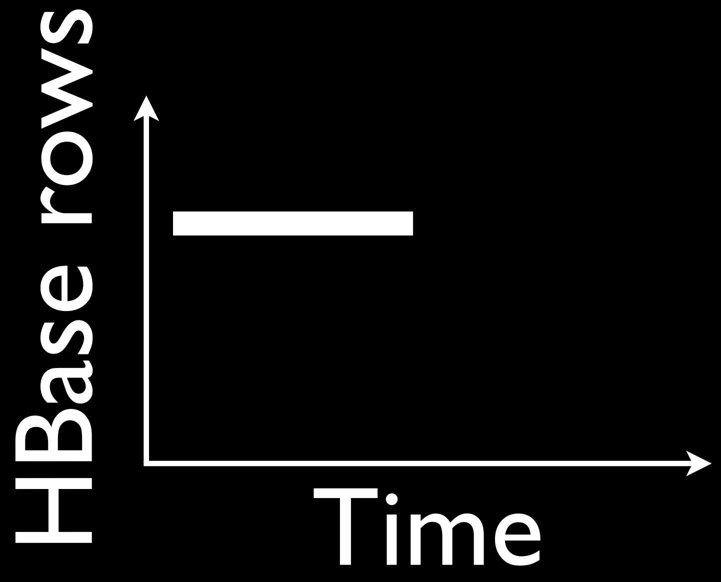


Problem #1

How to Synchronize Concurrent Writes (or not)

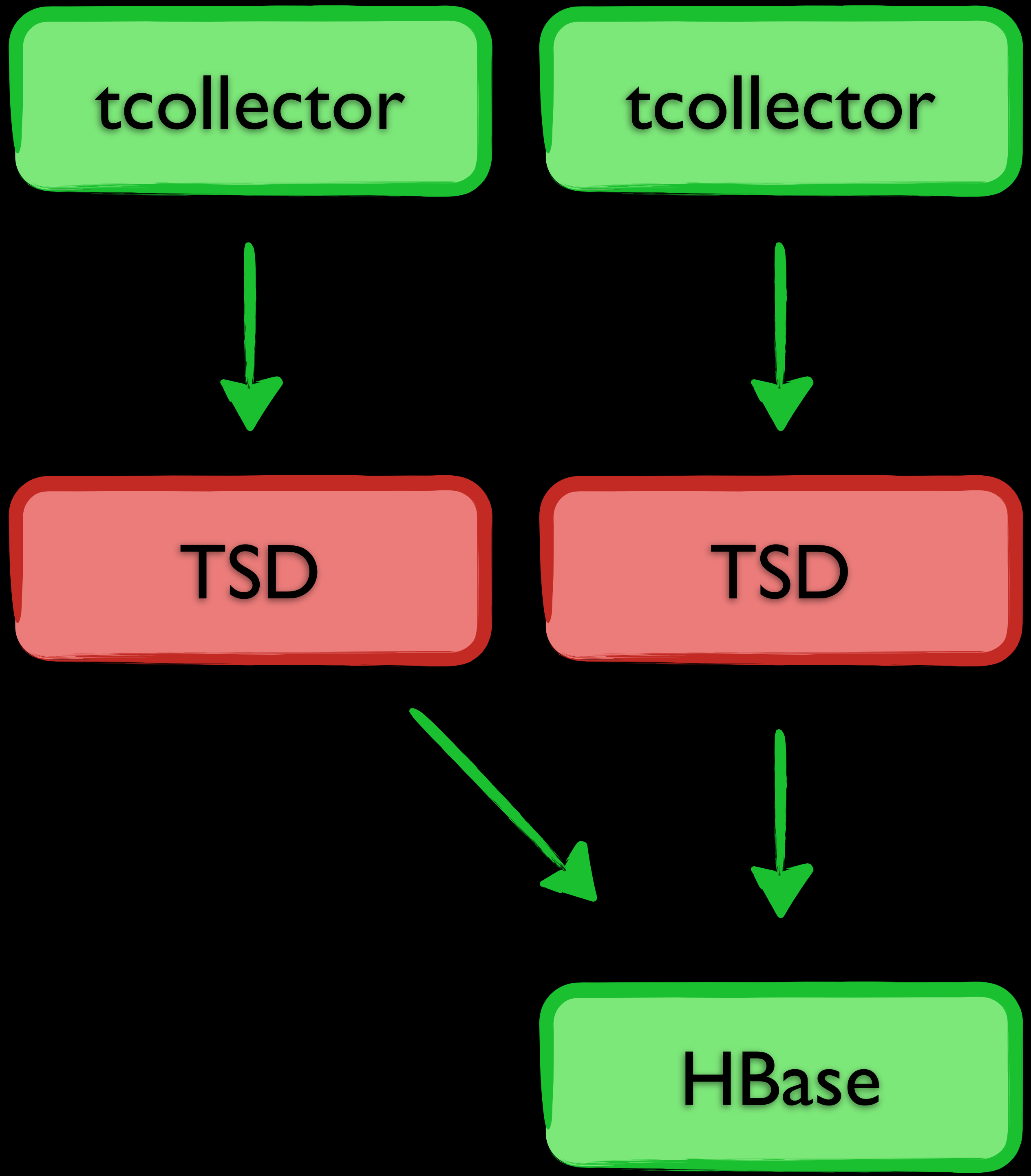


Some data gets retransmitted

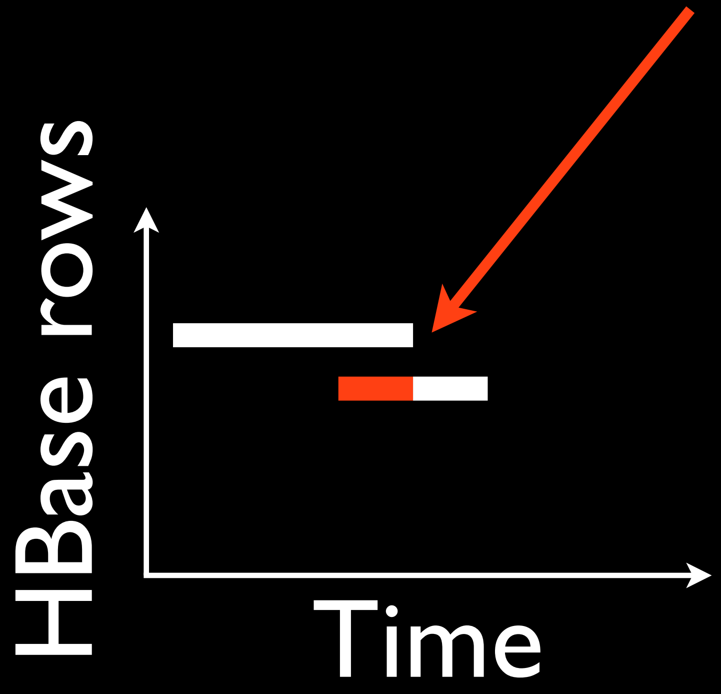


Problem #1

How to Synchronize Concurrent Writes (or not)



Now we have 2 overlapping rows containing duplicate data!



How to avoid synchronizing TSD?

- First approach: use a token system
- Clients give their token (if they have one) after connecting
- TSD verifies if it knows the token. If not, it issues a new one to the client.
- Clients with new tokens create new rows
- Clients with existing tokens append to the same rows
- ... This is too complicated!

How to avoid synchronizing TSD?

- Second approach: make writes idempotent
- Rows always start on a known, pre-set boundary (current 10 minutes)
- No need for state exchange between TSDs or between TSDs and clients
- Much simpler
- Also allows out-of-order events to be processed

Demo Time!

100% Natural, Organic Free & Open-Source

Danger in the Corn by Roger Smith

¿ Questions ?

opentsdb.net

Benoît “tsuna” Sigoure
tsuna@stumbleupon.com

StumbleUpon