Monitoring Complex Systems

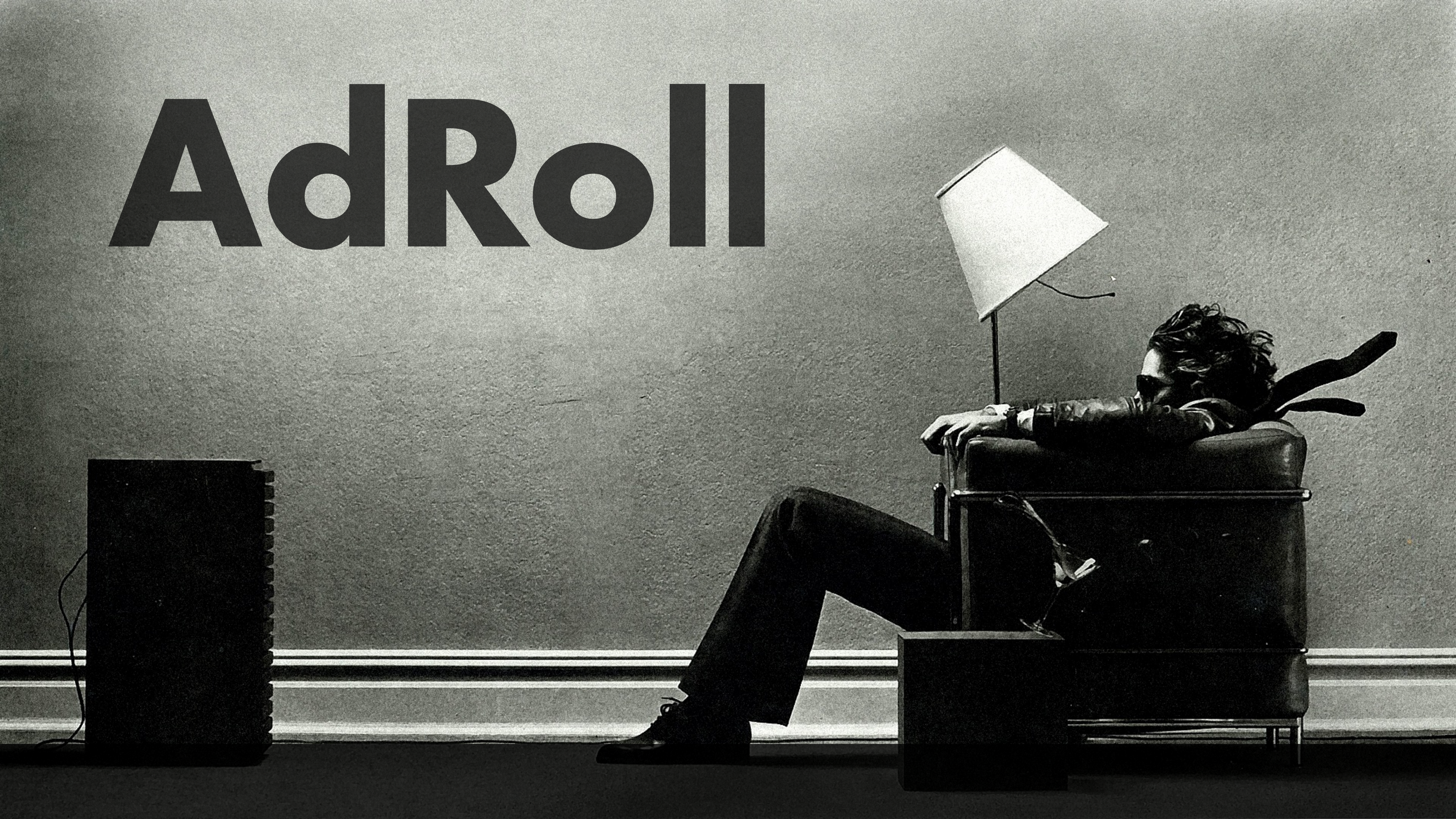
KEEPING YOUR HEAD ON STRAIGHT IN A HARD WORLD

I do things to/with computers.


I build real-time
systems.


I build fault-
tolerant systems.


I build critical
systems.


AdRoll

Less this.


More this.

Engineering + Mathematics = ads

Engineering + Mathematics = ads

YOU'RE WELCOME


REAL-TIME
BIDDING

The Problem Domain

- Low latency (< 100 ms per transaction)

The Problem Domain

- Low latency (< 100 ms per transaction)
- Firm real-time system

The Problem Domain

- Low latency (< 100 ms per transaction)
- Firm real-time system
- Highly concurrent (~2 million transactions per second, peak)

The Problem Domain

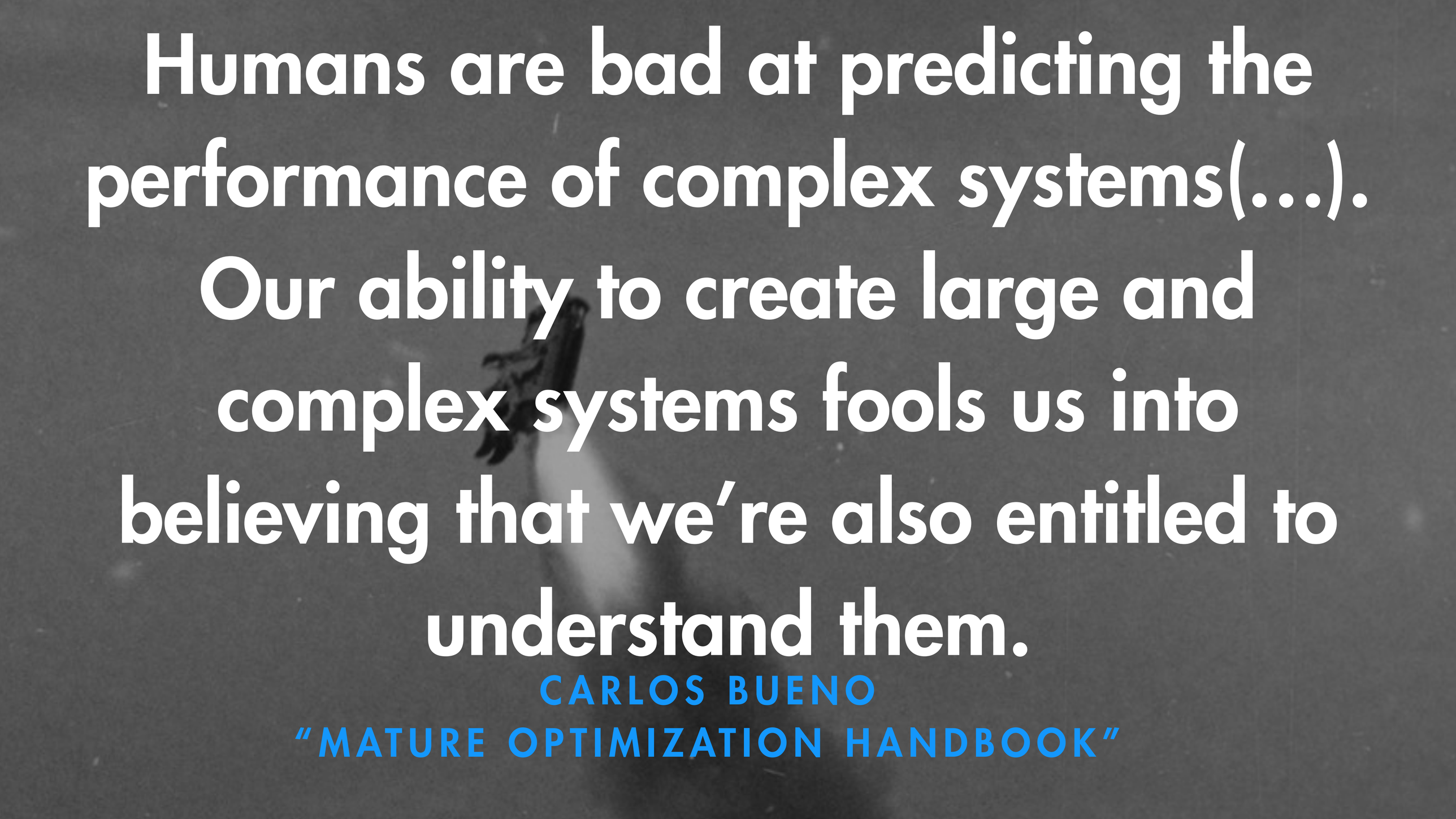
- Low latency (< 100 ms per transaction)
- Firm real-time system
- Highly concurrent (~2 million transactions per second, peak)
- Global, 24/7 operation

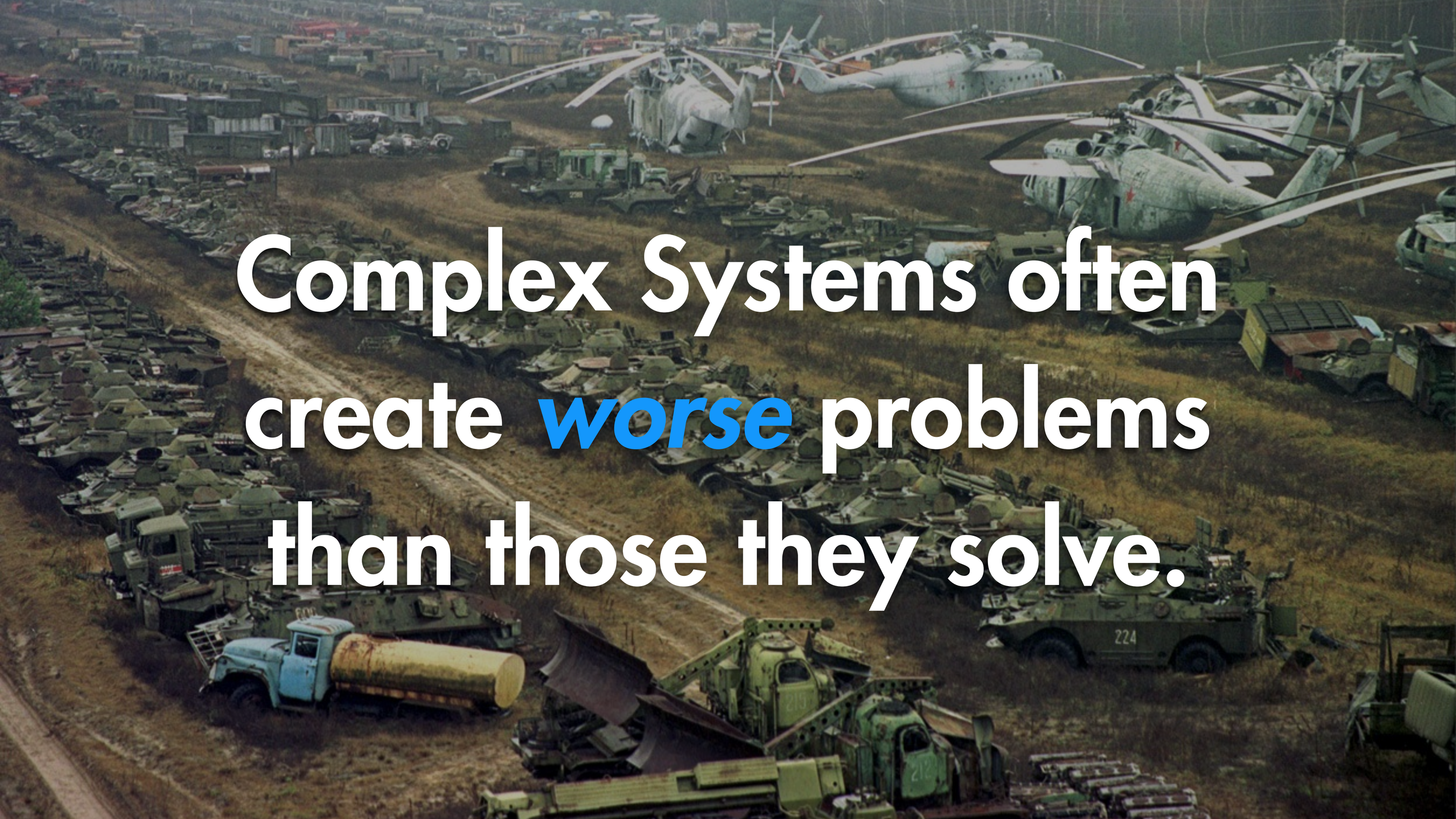

I build
Complex Systems

Complex Systems

- Non-linear feedback
- Tightly coupled to external systems
- Difficult to model, understand


Bad things happen when
Complex Systems fail.


Humans are bad at predicting the performance of complex systems(...). Our ability to create large and complex systems fools us into believing that we're also entitled to understand them.
CARLOS BUENO
"MATURE OPTIMIZATION HANDBOOK"


Complex Systems often
create *worse* problems
than those they solve.

The key challenge to sustaining a complex system is maintaining our understanding of it.


What can be done?
HASTA MANANA
DAGELIJKS OPEN VA 17U30
Tapas
Vinos
Sangria
Cocktails
Aperitivos
Cafés
Cervezas
Zumos
Refrescos
¡BIENVENIDO!

Compile-time guarantees are not sufficient.

Compile-time guarantees are not sufficient.

DON'T SCRIMP ON THEM, THOUGH

Ahead of time verification is not sufficient.

Ahead of time verification is not sufficient.

DON'T SCRIMP ON THESE, EITHER


We need insight into the
running system.

What are we looking for?

- VM killers

What are we looking for?

- VM killers
- Application performance regressions

What are we looking for?

- VM killers
- Application performance regressions
- Abnormal application behavior

What are we looking for?

- VM killers
- Application performance regressions
- Abnormal application behavior
- Surprises

INSTRUMENTATION

BEAM is ready to play.

erlang:memory/1

erlang:memory/1

- ets
- binary
- atom
- total
- processes
- system

erlang:statistics/1

erlang:statistics/1

- run_queue
- garbage_collection
- io

erlang:system_info/1

erlang:system_info/1

- port_count
- process_count
- *_limit


What about
our own work?


Exometer

Important Terms

metric	a measurement
entry	a receiver and aggregator of metrics
reporter	that which samples entries periodically and ships them to another system
subscription	the definition of the regular interval on which reporters sample entries

exometer

- Responsive upstream (Ulf Wiger never sleeps?)
- Metric collection, aggregation and reporting decoupled.
- Static and dynamic configuration.
- Very low, predictable runtime overhead.

Defining Entries

```
{predefined, [
  {[erlang,memory],
   {function, erlang, memory,
    ['$dp'], value,[ets,binary]},
   []
  },

  {[erlang, statistics],
   {function, erlang, statistics,
    ['$dp'], value, [run_queue]},
   []
  },

  {[erlang, gc],
   {function, erlang, statistics,
    [garbage_collection], match,
    {total_coll, rec_wrd, '_'}},
   []
  },

  {[boodah, freq_cap, not_found], spiral},
  {[boodah, freq_cap, ok], spiral},
  {[boodah, freq_cap, timeout], spiral}
]},
```

Defining Entries

```
{predefined, [
  {[erlang,memory],
   {function, erlang, memory,
    ['$dp'], value,[ets,binary]},
   []
  },

  {[erlang, statistics],
   {function, erlang, statistics,
    ['$dp'], value, [run_queue]},
   []
  },

  {[erlang, gc],
    {function, erlang, statistics,
     [garbage_collection], match,
     {total_coll, rec_wrd, '_'}},
    []
   },

   {[boodah, freq_cap, not_found], spiral},
   {[boodah, freq_cap, ok], spiral},
   {[boodah, freq_cap, timeout], spiral}
]},
```

Defining Entries

```
{predefined, [
  {[erlang,memory],
   {function, erlang, memory,
    ['$dp'], value,[ets,binary]},
   []
  },

  {[erlang, statistics],
   {function, erlang, statistics,
    ['$dp'], value, [run_queue]},
   []
  },

  {[erlang, gc],
   {function, erlang, statistics,
    [garbage_collection], match,
    {total_coll, rec_wrd, '_'}},
   []
  },

  {[boodah, freq_cap, not_found], spiral},
  {[boodah, freq_cap, ok], spiral},
  {[boodah, freq_cap, timeout], spiral}
]},
```

Defining Entries

```
{predefined, [
  {[erlang,memory],
   {function, erlang, memory,
    ['$dp'], value,[ets,binary]},
   []
  },

  {[erlang, statistics],
   {function, erlang, statistics,
    ['$dp'], value, [run_queue]},
   []
  },

  {[erlang, gc],
   {function, erlang, statistics,
    [garbage_collection], match,
    {total_coll, rec_wrd, '_'}},
   []
  },

  {[boodah, freq_cap, not_found], spiral},
  {[boodah, freq_cap, ok], spiral},
  {[boodah, freq_cap, timeout], spiral}
]},
```

Defining Entries

```
{predefined, [
  {[erlang,memory],
   {function, erlang, memory,
    ['$dp'], value,[ets,binary]},
   []
  },

  {[erlang, statistics],
   {function, erlang, statistics,
    ['$dp'], value, [run_queue]},
   []
  },

  {[erlang, gc],
   {function, erlang, statistics,
    [garbage_collection], match,
    {total_coll, rec_wrd, '_'}},
   []
  },

  {[boodah, freq_cap, not_found], spiral},
  {[boodah, freq_cap, ok], spiral},
  {[boodah, freq_cap, timeout], spiral}
]},
```

Defining Reporters

```
{ reporters,
  [
   { exometer_report_statsd,
     [
      {hostname, "localhost"},
      {port, 8125},
      {type_map,
       [
        {[erlang,statistics,run_queue],
         histogram},
        {[erlang, gc, tot_coll], histogram},
        {[erlang, gc, rec_wrd], histogram},
        {[erlang,memory,ets], gauge},
        {[erlang,memory,binary],gauge},

        {[boodah,freq_cap,not_found],gauge},
        {[boodah,freq_cap,ok],gauge},
        {[boodah,freq_cap,timeout],gauge}
      ]},
```

Defining Reporters

```
{ reporters,
  [
   { exometer_report_statsd,
     [
      {hostname, "localhost"},
      {port, 8125},
      {type_map,
       [
        {[erlang,statistics,run_queue],
         histogram},
        {[erlang, gc, tot_coll], histogram},
        {[erlang, gc, rec_wrd], histogram},
        {[erlang,memory,ets], gauge},
        {[erlang,memory,binary],gauge},

        {[boodah,freq_cap,not_found],gauge},
        {[boodah,freq_cap,ok],gauge},
        {[boodah,freq_cap,timeout],gauge}
      ]},
```

Defining Reporters

```
{ reporters,
  [
   { exometer_report_statsd,
     [
      {hostname, "localhost"},
      {port, 8125},
      {type_map,
       [
        {[erlang,statistics,run_queue],
         histogram},
        {[erlang, gc, tot_coll], histogram},
        {[erlang, gc, rec_wrd], histogram},
        {[erlang,memory,ets], gauge},
        {[erlang,memory,binary],gauge},

        {[boodah,freq_cap,not_found],gauge},
        {[boodah,freq_cap,ok],gauge},
        {[boodah,freq_cap,timeout],gauge}
      ]},
```

Defining Subscriptions

```
{ report,
   [
    { subscribers,
      [
       {exometer_report_statsd, [erlang, statistics],
        run_queue,  1000},

       {exometer_report_statsd, [erlang, gc],
        tot_coll, 1000},
       {exometer_report_statsd, [erlang, gc],
        rec_wrd, 1000},

       {exometer_report_statsd, [erlang, memory],
        ets, 10000},
       {exometer_report_statsd, [erlang, memory],
        binary, 10000},
       {exometer_report_statsd,
        [boodah, freq_cap, not_found],  one,
1000},
       {exometer_report_statsd,
        [boodah, freq_cap, ok], one, 1000},
       {exometer_report_statsd,
        [boodah, freq_cap, timeout], one, 1000}
      ]}
]}
```

Defining Subscriptions

```
{ report,
   [
    { subscribers,
      [
       {exometer_report_statsd, [erlang, statistics],
        run_queue,  1000},

       {exometer_report_statsd, [erlang, gc],
        tot_coll, 1000},
       {exometer_report_statsd, [erlang, gc],
        rec_wrd, 1000},

       {exometer_report_statsd, [erlang, memory],
        ets, 10000},
       {exometer_report_statsd, [erlang, memory],
        binary, 10000},
       {exometer_report_statsd,
        [boodah, freq_cap, not_found],  one,
1000},
       {exometer_report_statsd,
        [boodah, freq_cap, ok], one, 1000},
       {exometer_report_statsd,
        [boodah, freq_cap, timeout], one, 1000}
      ]}
]}
```

Defining Subscriptions

```
{ report,
   [
    { subscribers,
      [
       {exometer_report_statsd, [erlang, statistics],
        run_queue,  1000},

       {exometer_report_statsd, [erlang, gc],
        tot_coll, 1000},
       {exometer_report_statsd, [erlang, gc],
        rec_wrd, 1000},

       {exometer_report_statsd, [erlang, memory],
        ets, 10000},
       {exometer_report_statsd, [erlang, memory],
        binary, 10000},
       {exometer_report_statsd,
        [boodah, freq_cap, not_found],  one,
1000},
       {exometer_report_statsd,
        [boodah, freq_cap, ok], one, 1000},
       {exometer_report_statsd,
        [boodah, freq_cap, timeout], one, 1000}
      ]}
]}
```

Doing it dynamically.

```
1> exometer:new([a, histogram], histogram).
ok

2> exometer:get_value([a, histogram]).
{ok,[{n,0},
     {mean,0},
     {min,0},
     {max,0},
     {median,0},
     {50,0},
     {75,0},
     {90,0},
     {95,0},
     {99,0},
     {999,0}]}
```

```
3> exometer_report:add_reporter(
       exometer_report_tty, []).
ok

4> exometer_report:subscribe(
       exometer_report_tty,
       [a, histogram], mean, 1000, []).
ok

exometer_report_tty: a_histogram_mean
1393627070:0
exometer_report_tty: a_histogram_mean
1393627071:0
exometer_report_tty: a_histogram_mean
1393627072:0
```

These are all loosely coupled at runtime.

Configuration is static, but you can adapt it on the fly.

Creating Reporters

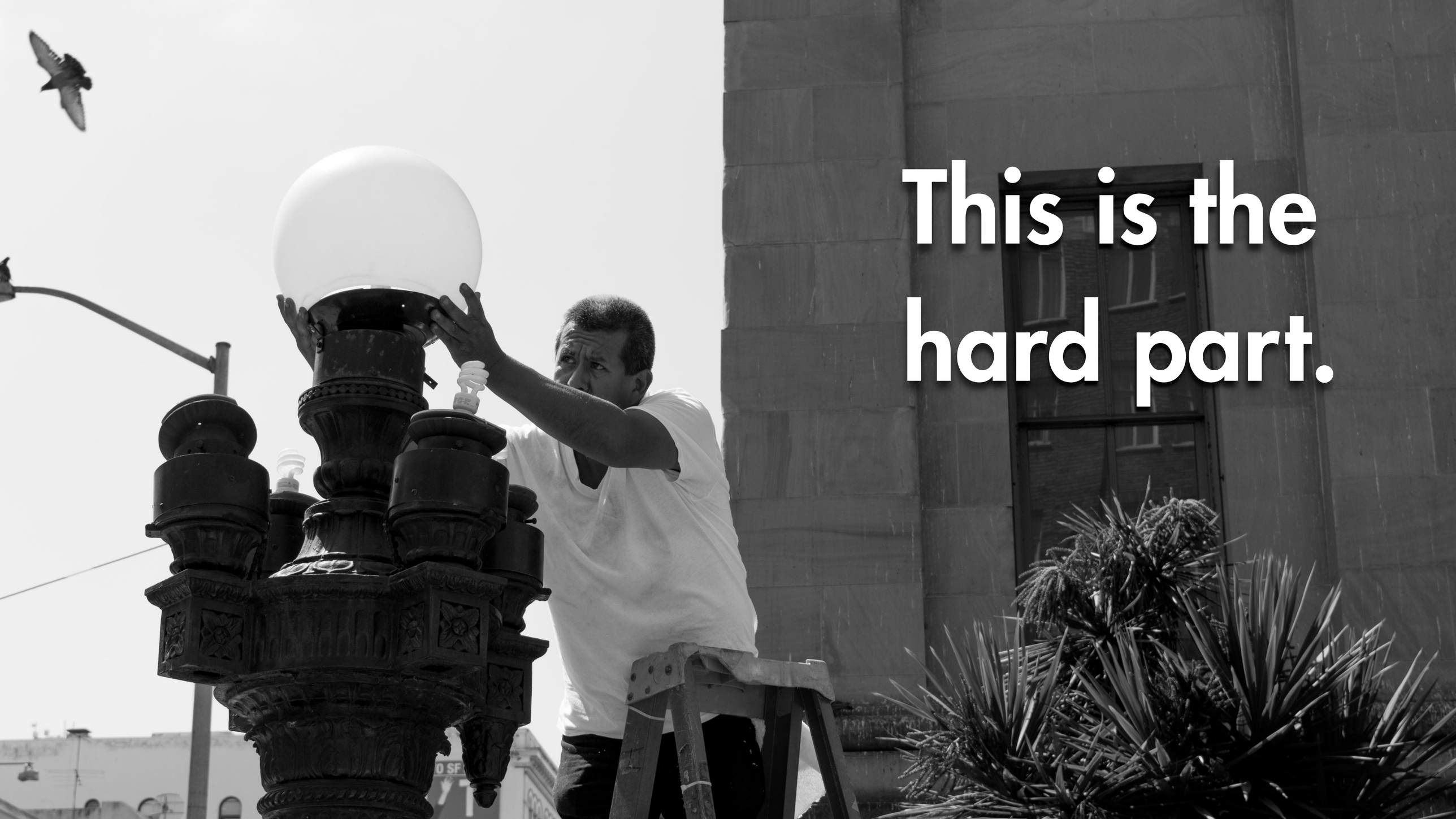
- Very easy to add your own reporters and entries.
- Reporters and entries can be proprietary. Just have to be loaded at runtime.
- Authors are responsive to issues.


Why not…
…folsom?
…statman?
…vmstat?

Okay, great. We have instrumentation.


Now what?

MONITORING

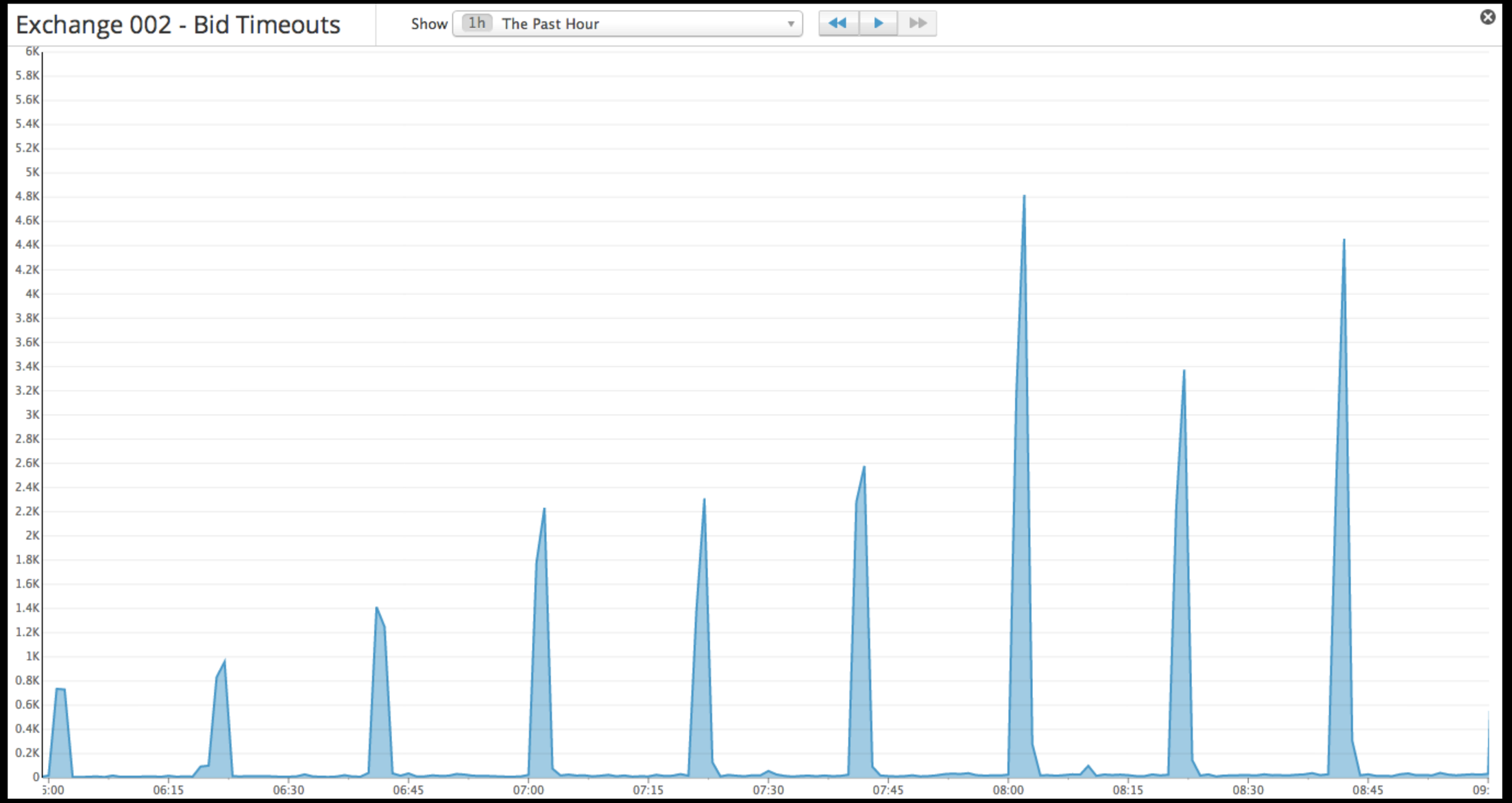

This is the
hard part.

Visualization

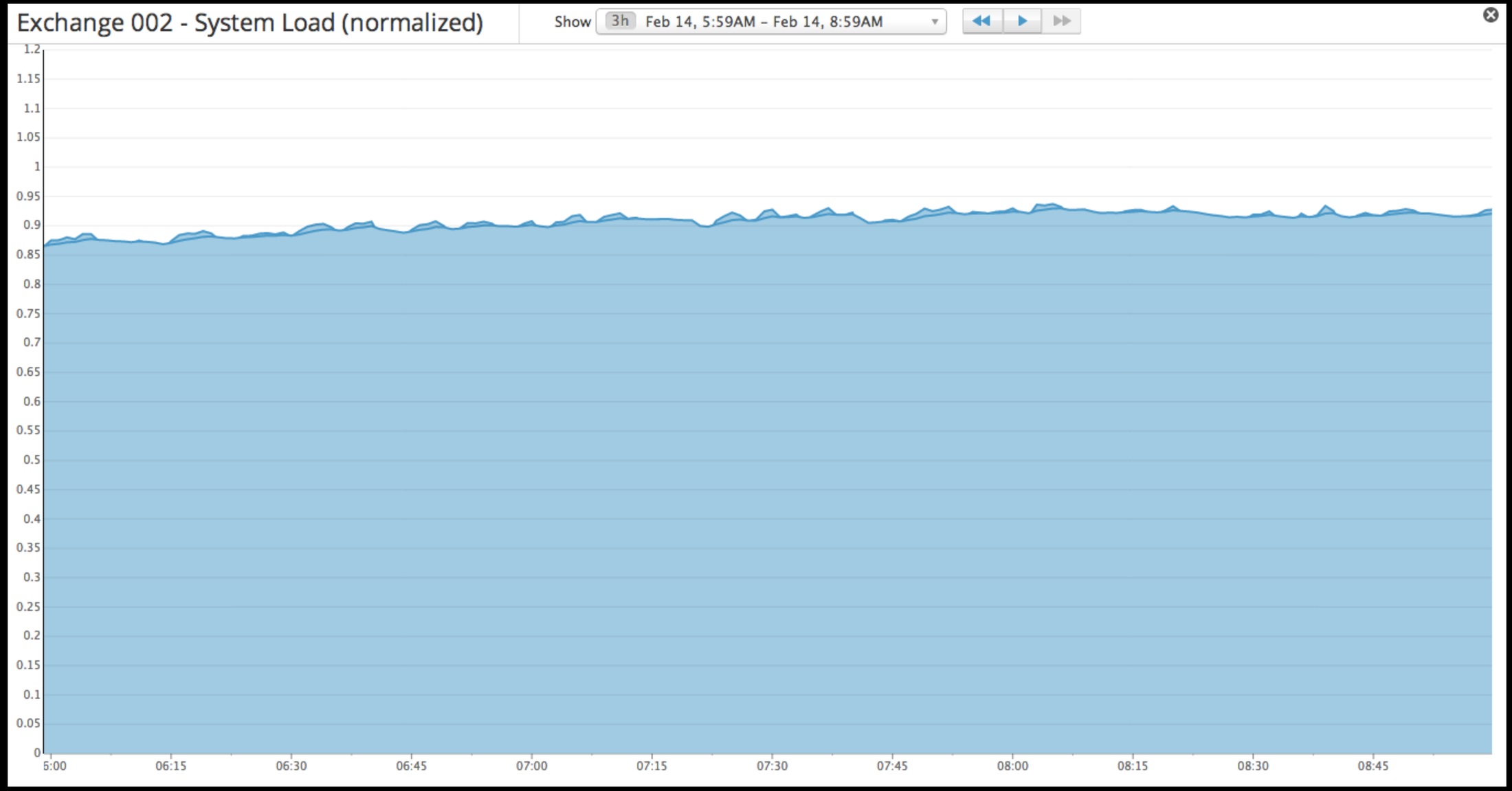
Alerting

Analysis

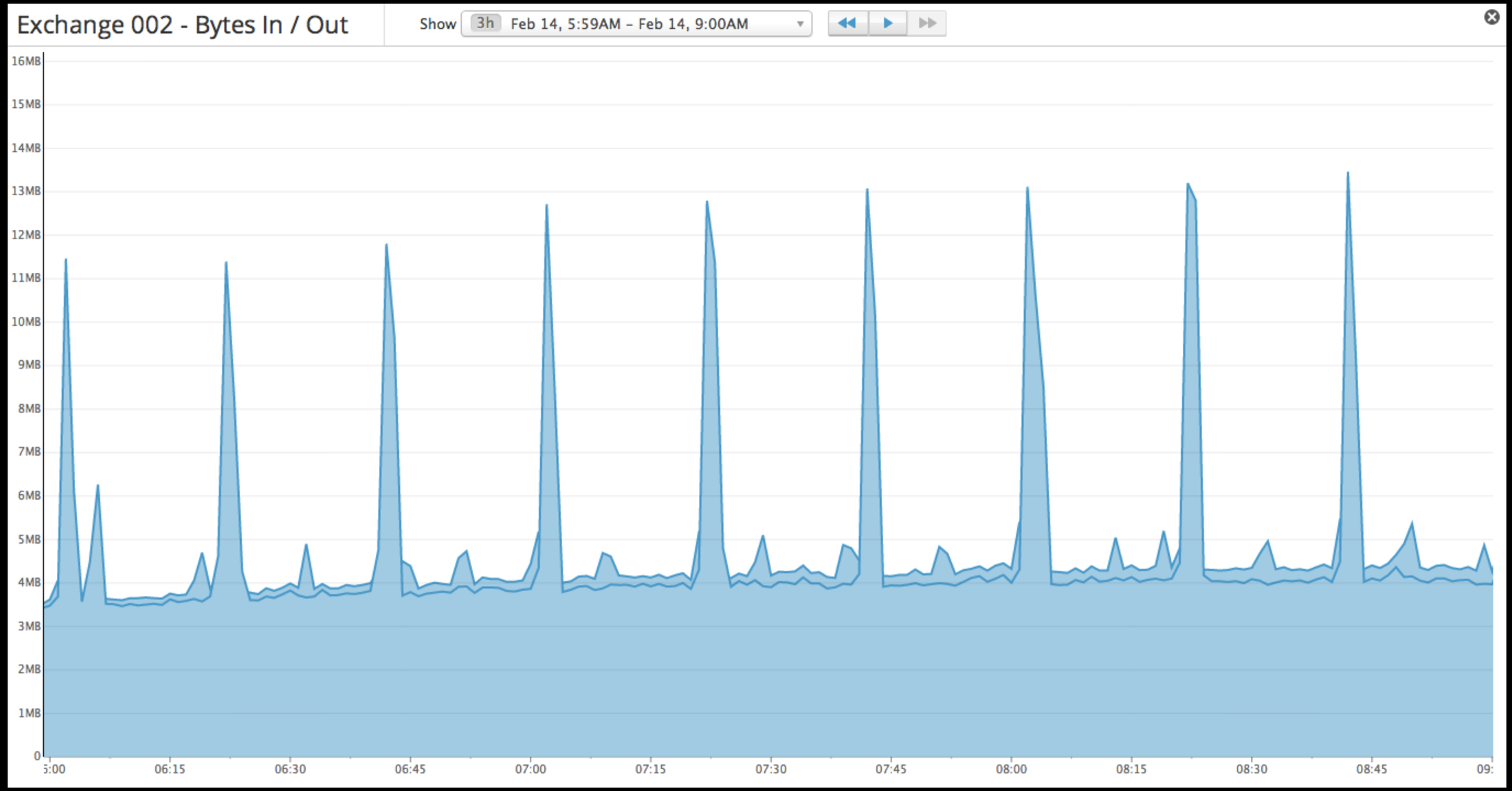
Visualization tells you how things look but not why.



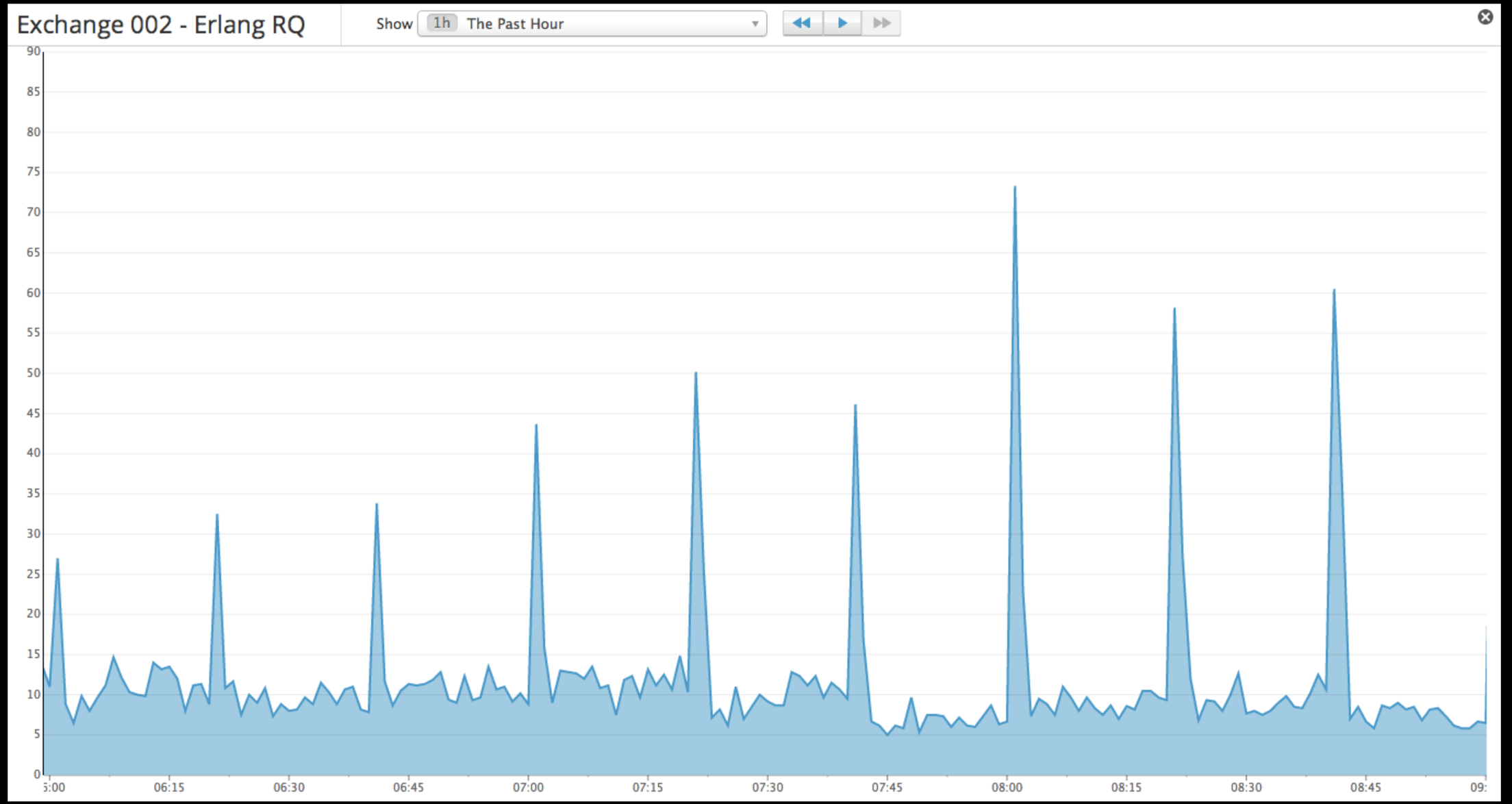
Periodic Bid Timeouts



Consistent System Load



Correlated Network Traffic Spikes



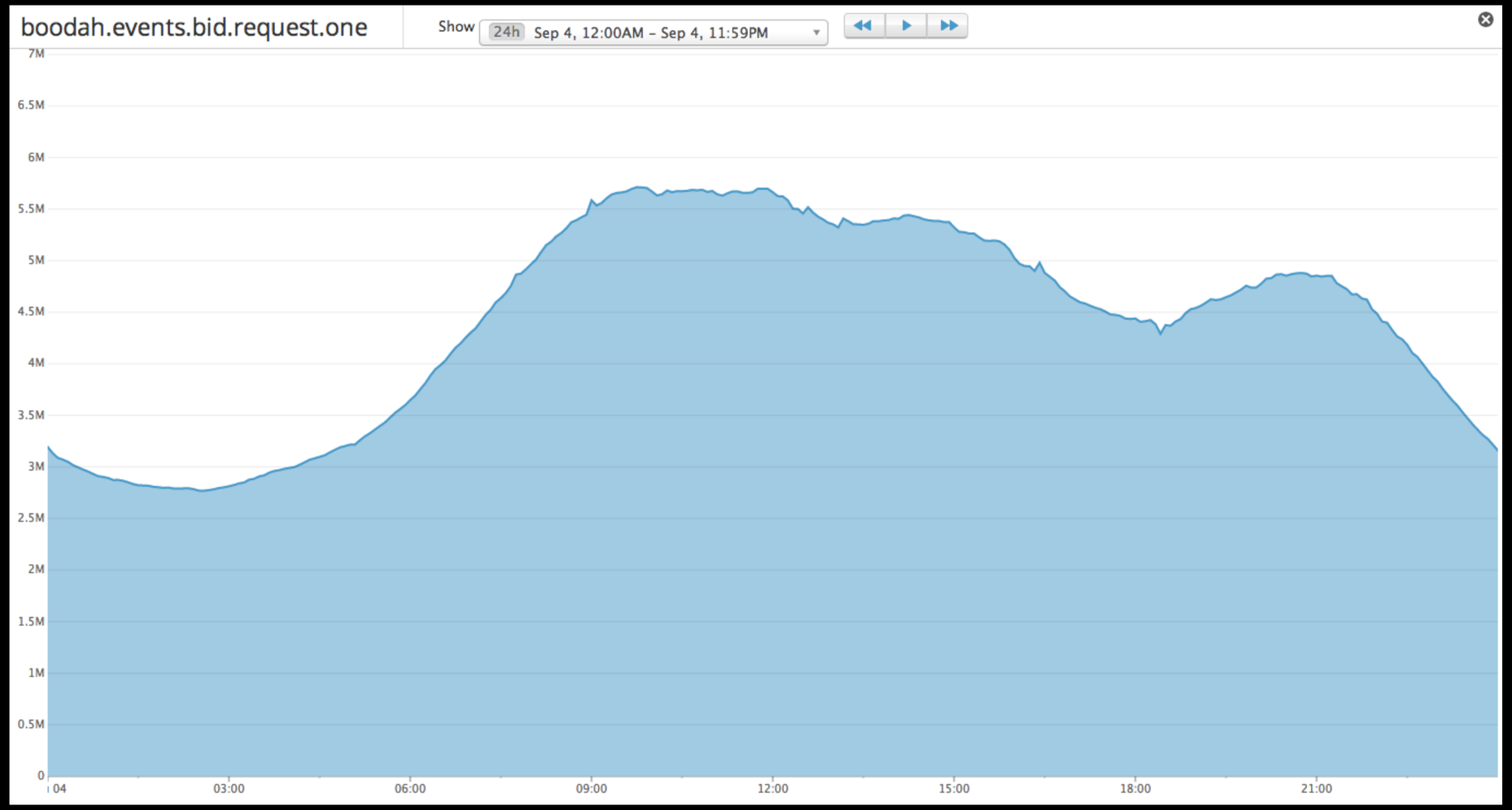
Correlated Run Queue Spikes

What happened?

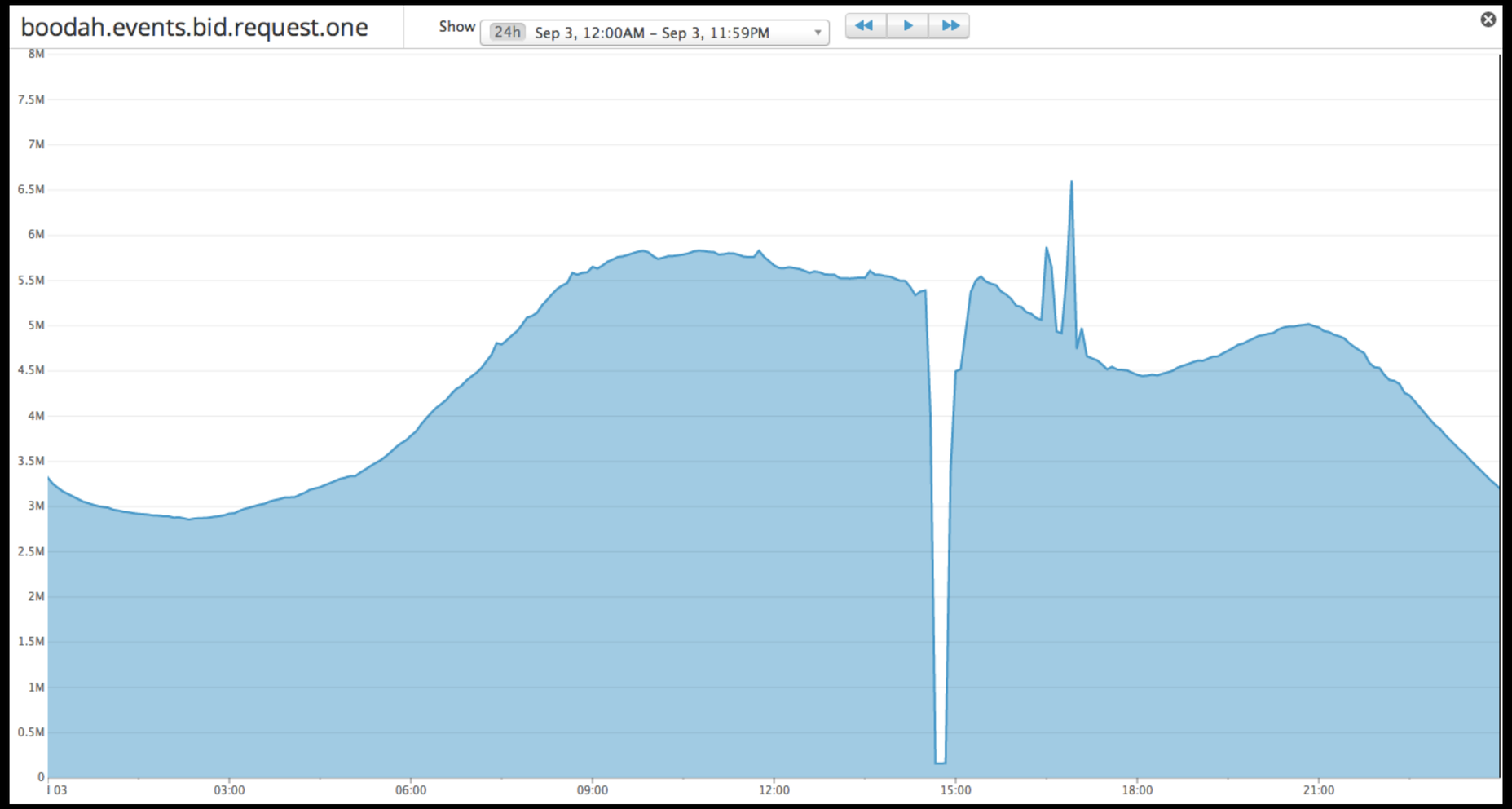
- Scheduler threads were locked to CPUs
- Background process comes on every 20 minutes, consumes a lot of CPU time
- No cpu-shield was set up on our production systems
- OS bumped a scheduler thread off its CPU, backing up its run-queue

Alerting tells you that something happened, but not why.

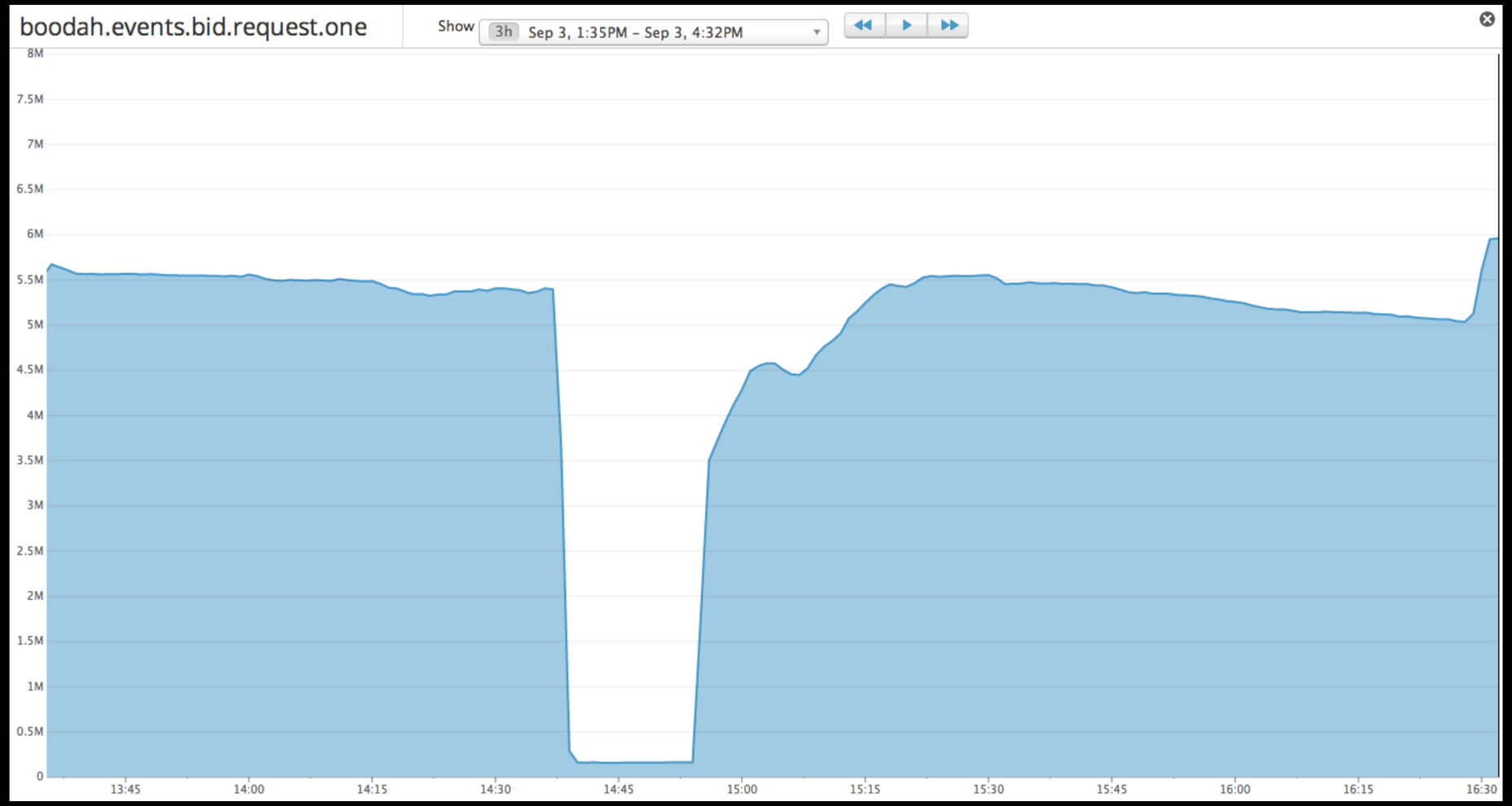
A normal day.

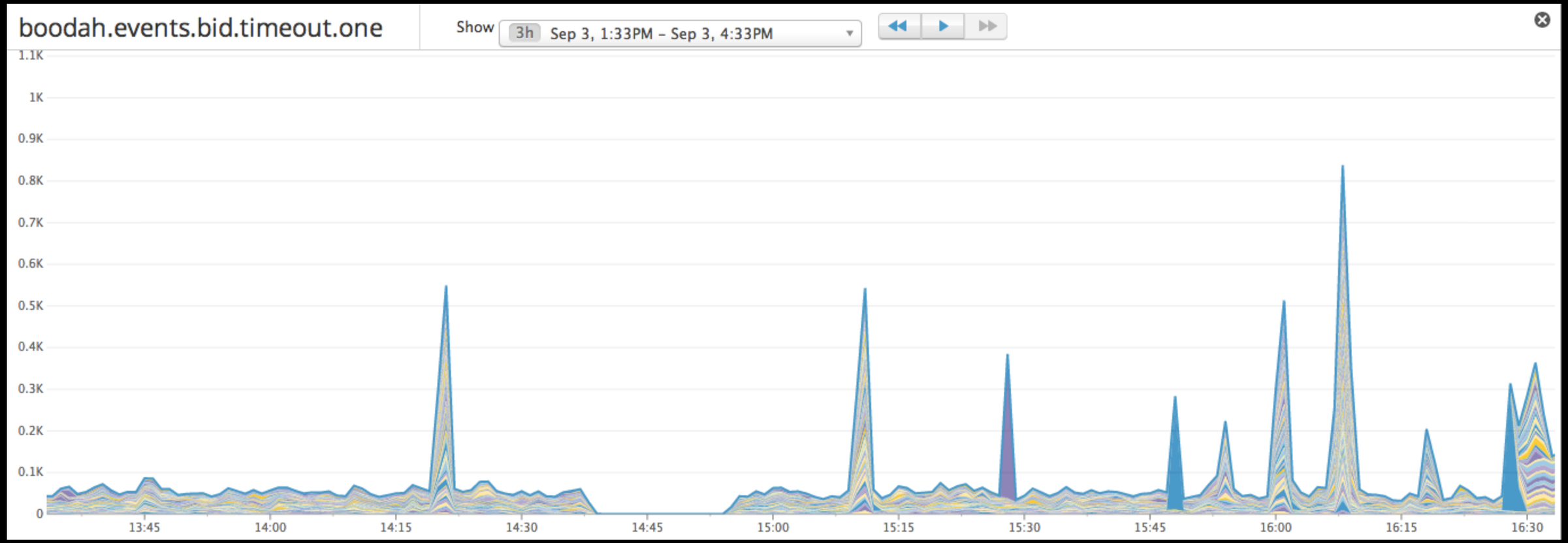

boodah.events.bid.request.one
Show
24h
Sep 4, 12:00AM – Sep 4, 11:59PM
7M
6.5M
6M
5.5M
5M
4.5M
4M
3.5M
3M
2.5M
2M
1.5M
1M
0.5M
0
04
03:00
06:00
09:00
12:00
15:00
18:00
21:00

Wat?


boodah.events.bid.request.one
Show
24h
Sep 3, 12:00AM – Sep 3, 11:59PM
8M
7.5M
7M
6.5M
6M
5.5M
5M
4.5M
4M
3.5M
3M
2.5M
2M
1.5M
1M
0.5M
0
03
03:00
06:00
09:00
12:00
15:00
18:00
21:00

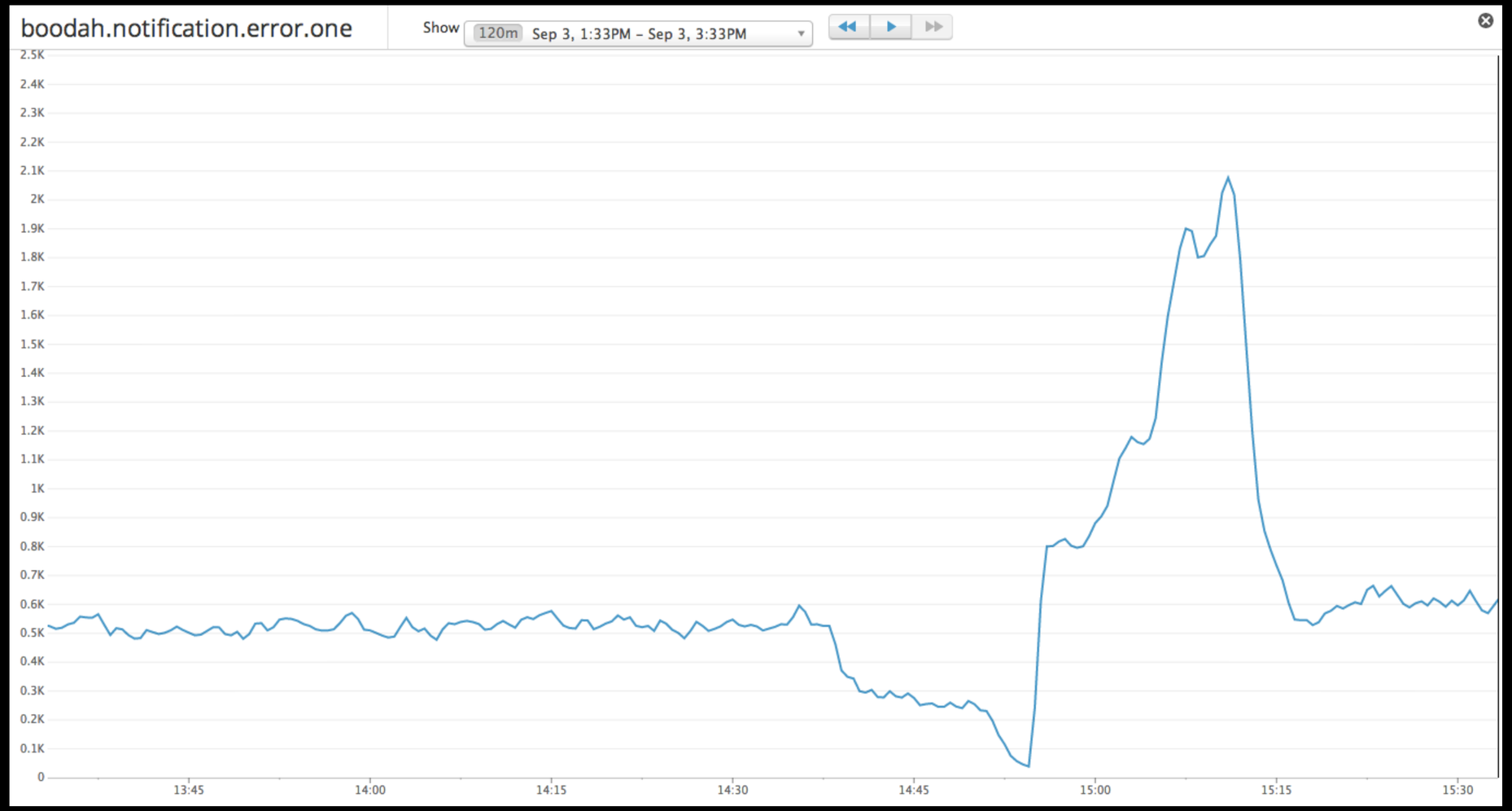
That's some cliff.





Timeouts look good.

Errors prior are okay.

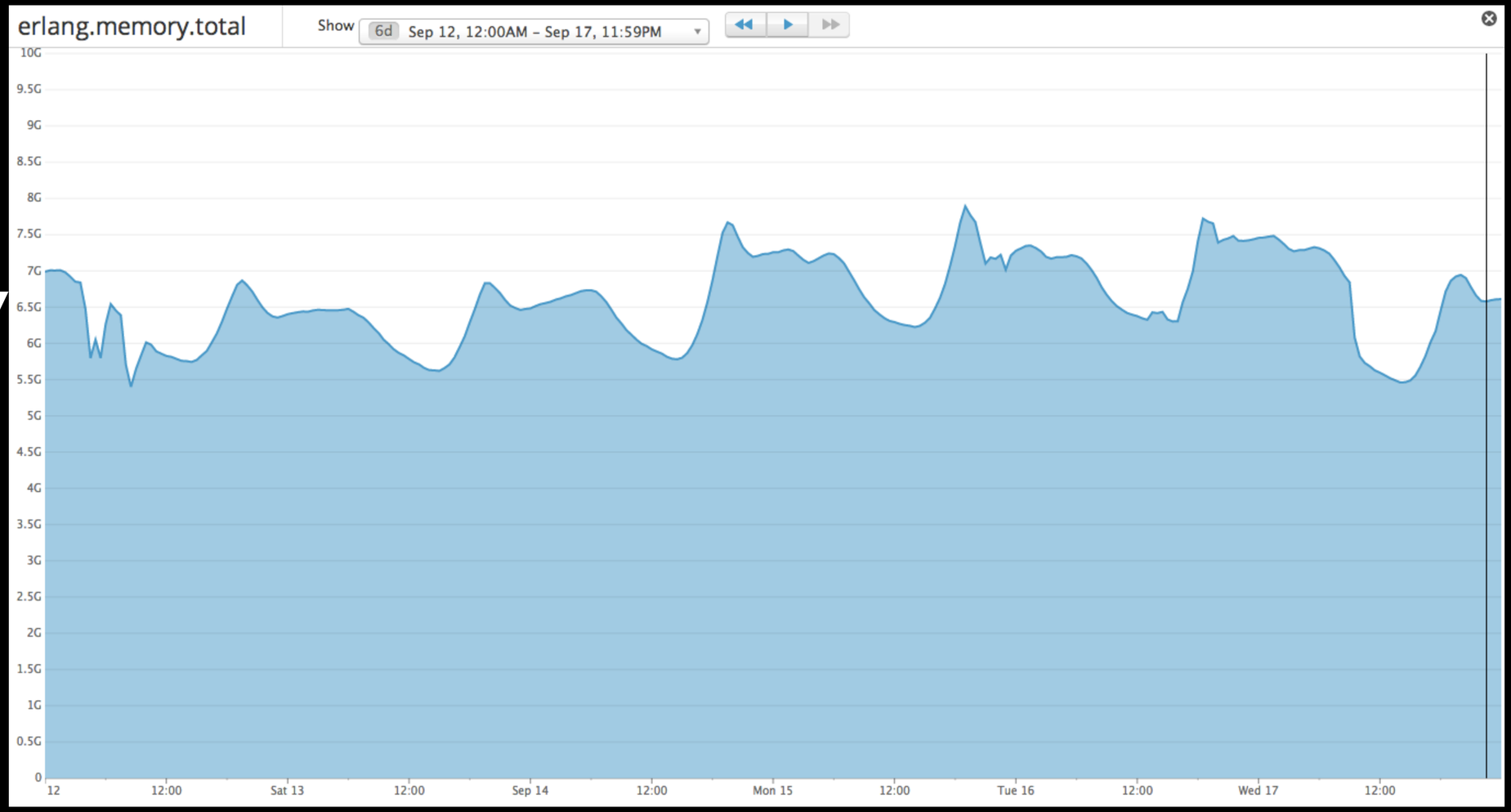


What happened?

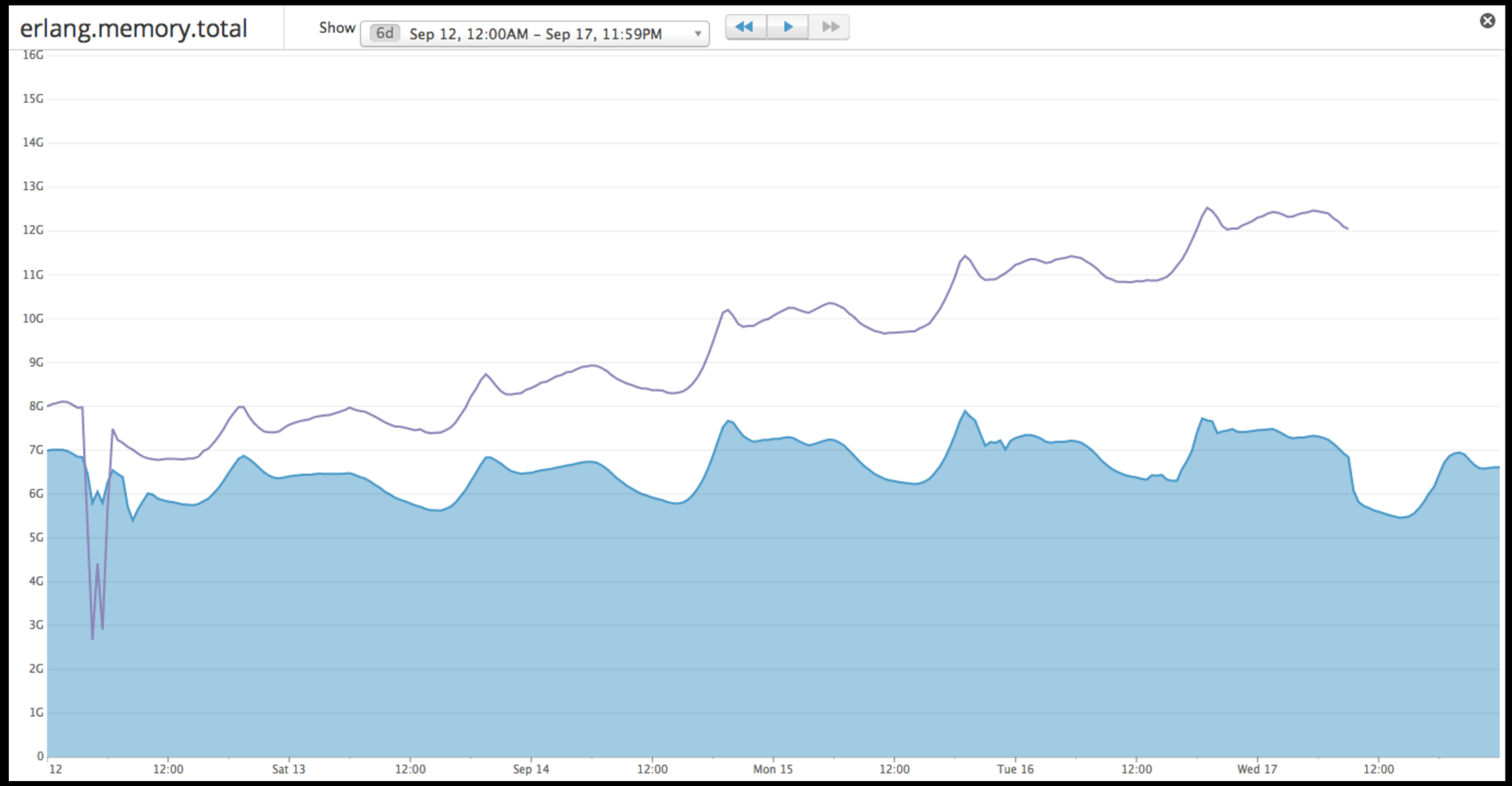
“Uh, hey guys, you know Facebook is down, right?”

Analysis gives you why
but only if you know
how to ask for what.

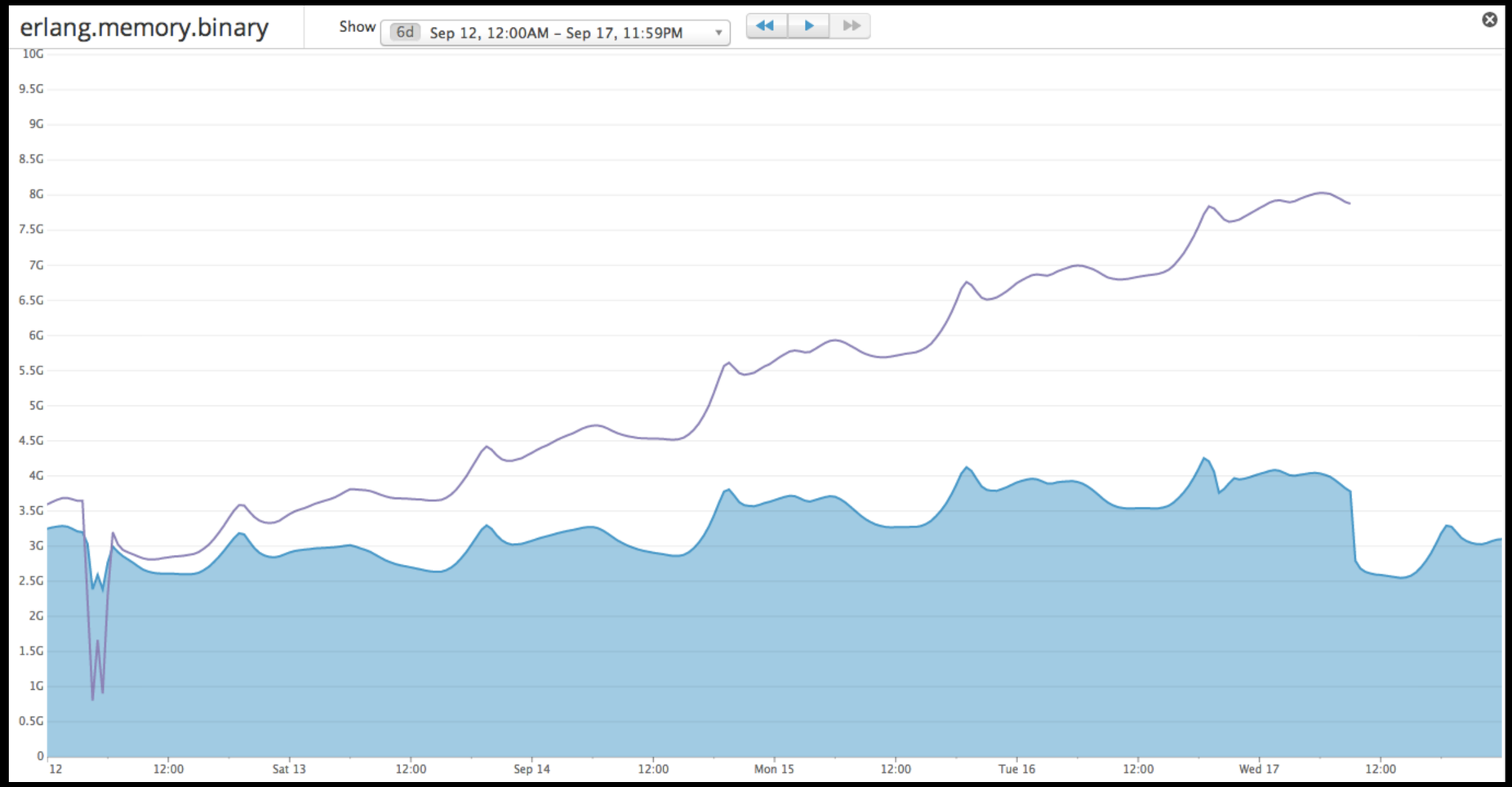
The memory use of a bidder.

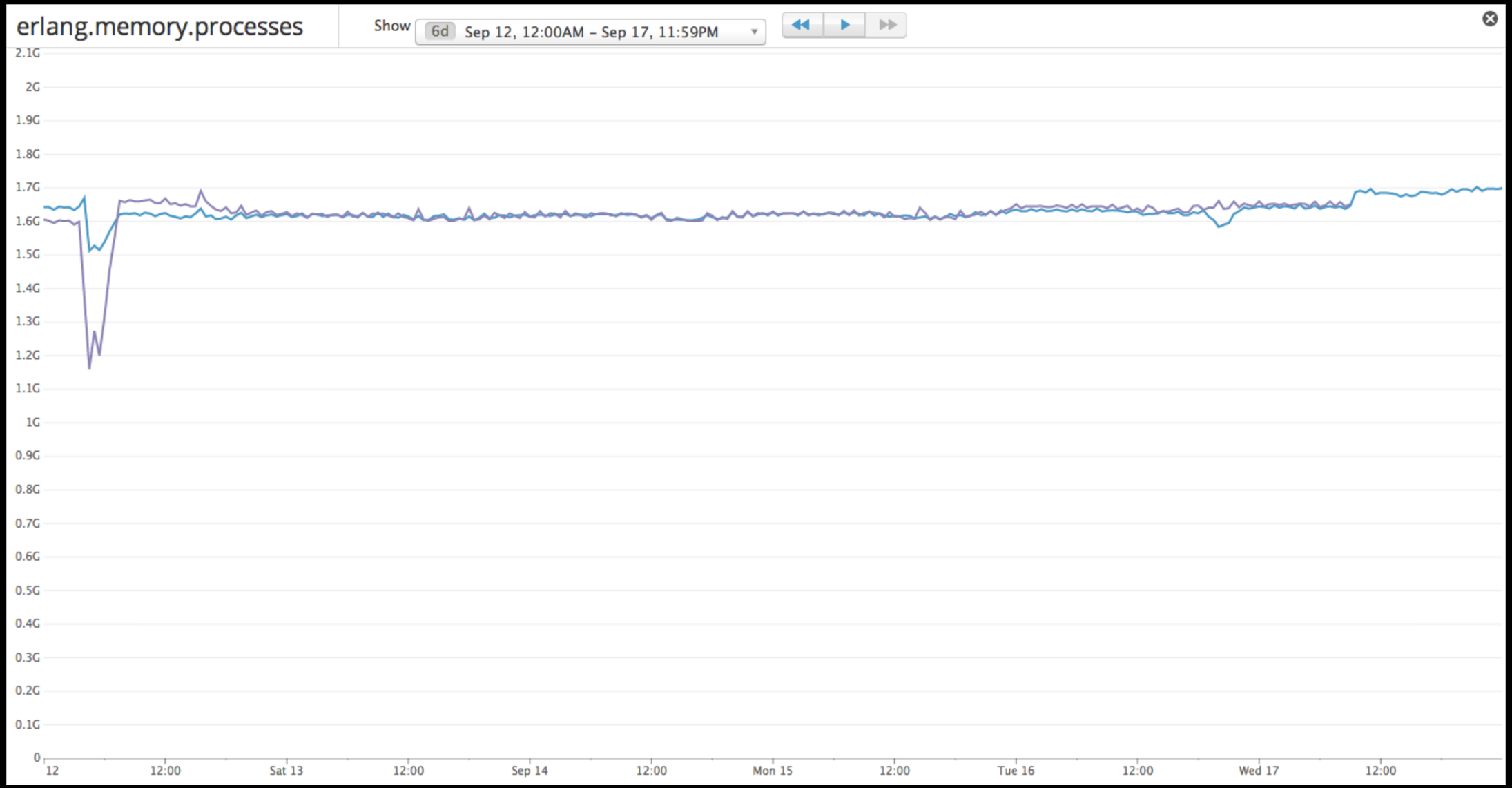


ಠ_ಠ


erlang.memory.total
Show
6d
Sep 12, 12:00AM – Sep 17, 11:59PM
16G
15G
14G
13G
12G
11G
10G
9G
8G
7G
6G
5G
4G
3G
2G
1G
0
12
12:00
Sat 13
12:00
Sep 14
12:00
Mon 15
12:00
Tue 16
12:00
Wed 17
12:00

It's all binaries.





Not in processes.


erlang.memory.ets
Show
6d
Sep 12, 12:00AM – Sep 17, 11:59PM
4G
3.8G
3.6G
3.4G
3.2G
3G
2.8G
2.6G
2.4G
2.2G
2G
1.8G
1.6G
1.4G
1.2G
1G
0.8G
0.6G
0.4G
0.2G
0
12
12:00
Sat 13
12:00
Sep 14
12:00
Mon 15
12:00
Tue 16
12:00
Wed 17
12:00

Not in ETS.


Come on now.

What happened?

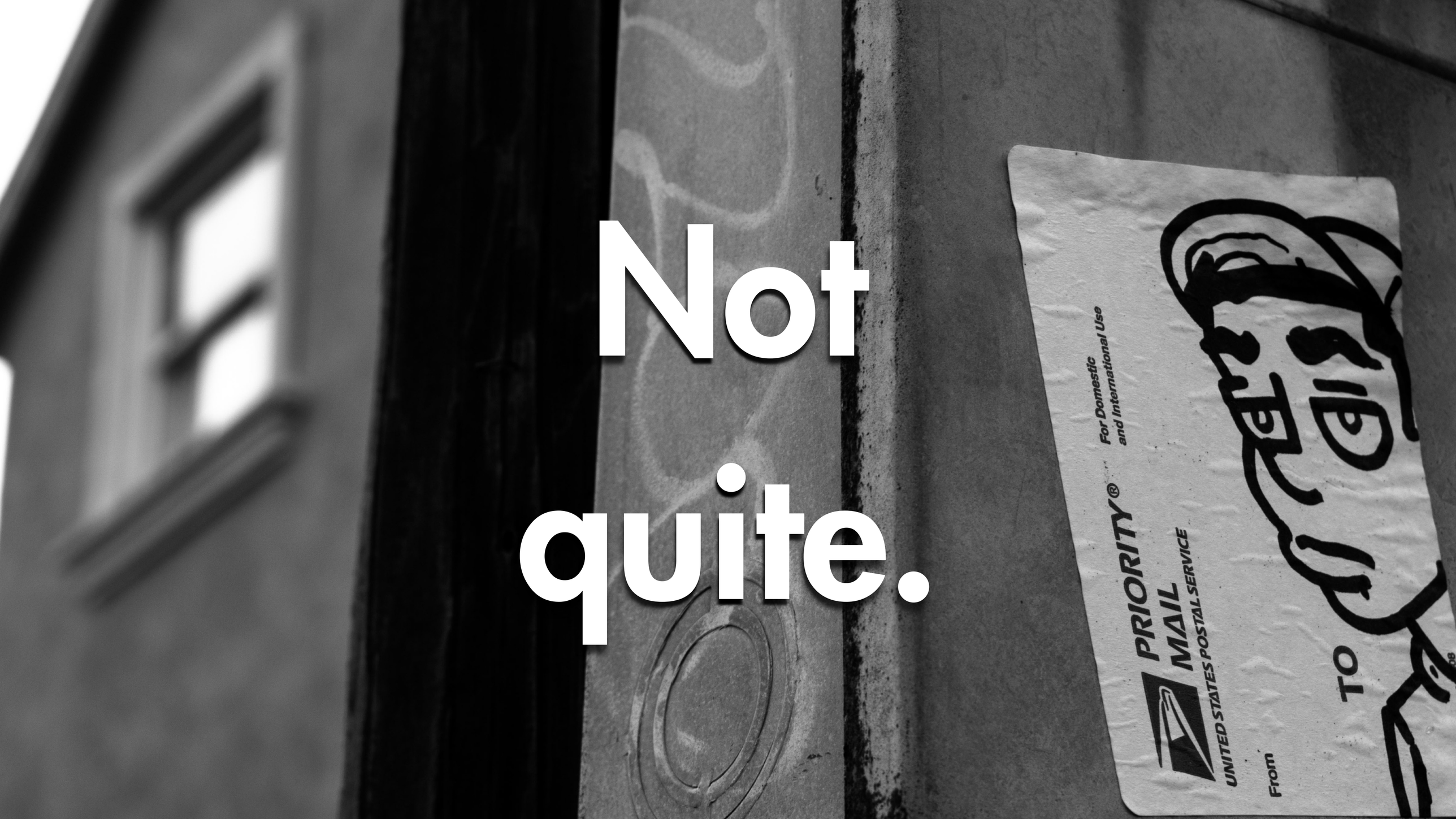
A jiffy bug.

A jiffy bug.

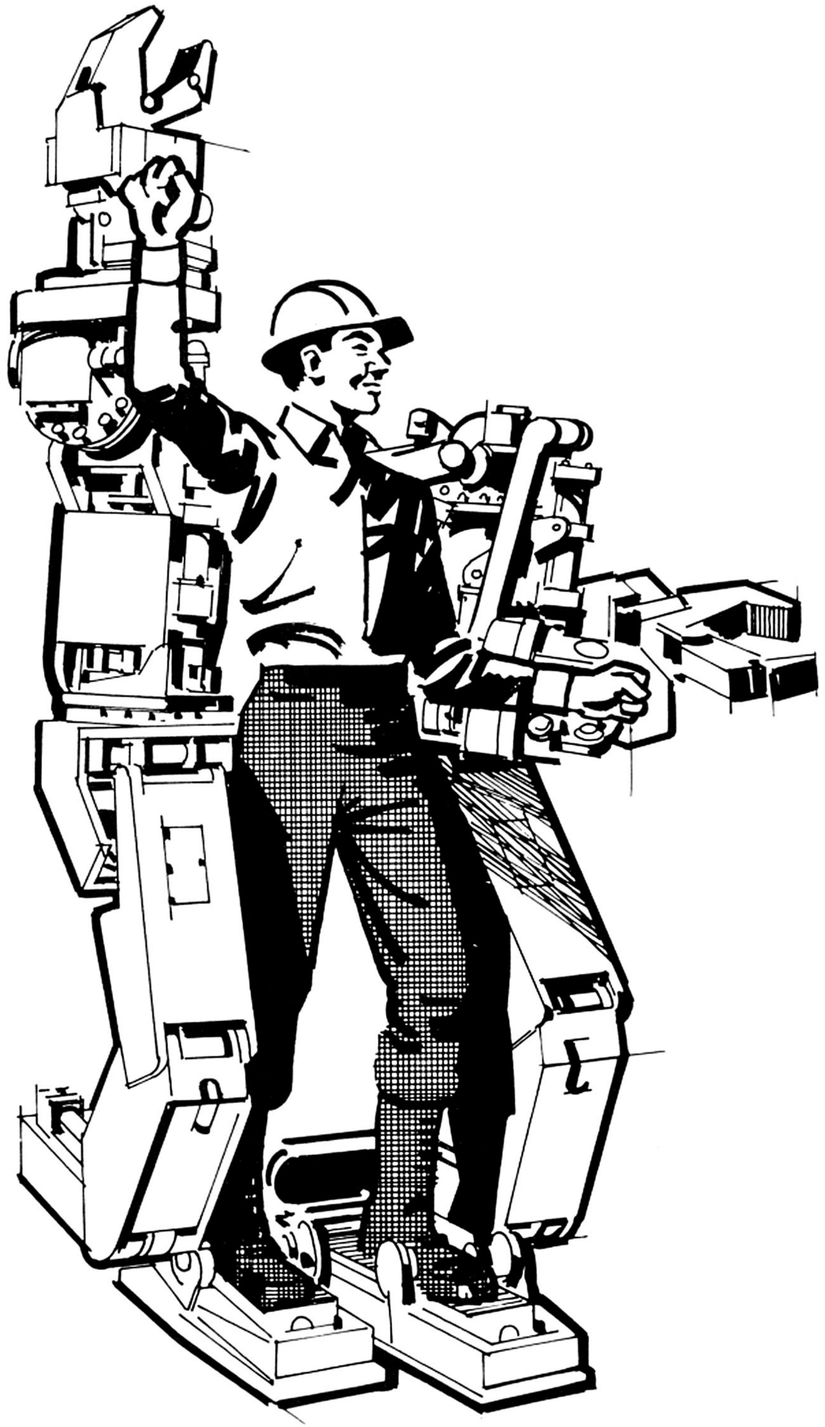
A BYTE HERE, A BYTE THERE EVENTUALLY IT TURNS INTO REAL MEMORY.

Okay, great. We have monitoring *and* instrumentation.

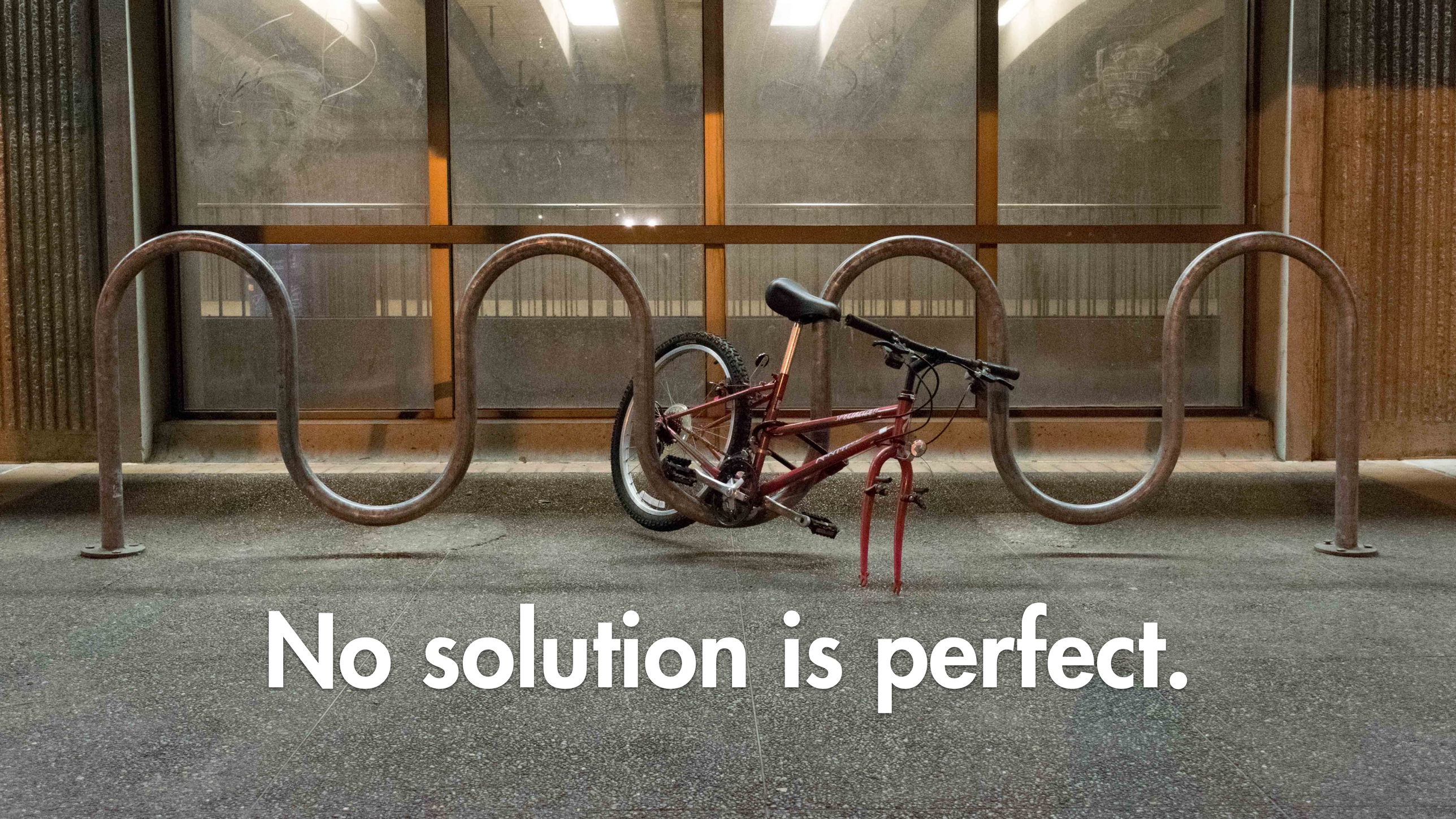
Now all our problems are solved, right?

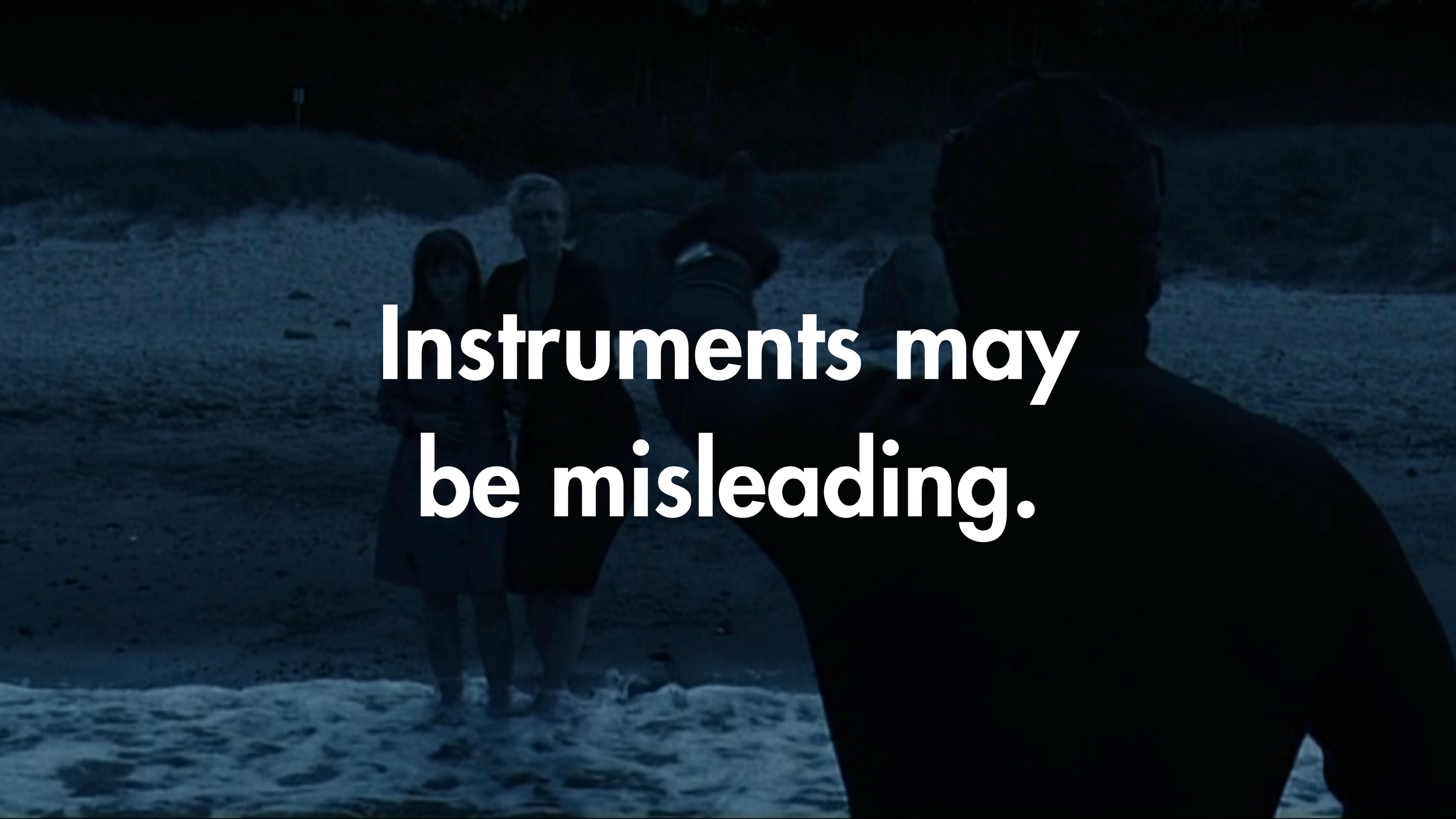

Not
quite.
PRIORITY®
MAIL
UNITED STATES POSTAL SERVICE
For Domestic
and International Use
From
TO


Instruments make up
for our lack of
insight.
WARNING
BE RECORDED
ON THIS VEHICLE
Free Credit
Monitoring
No Credit Card
Needed. No Catch.
App Store



Monitoring makes up for our frailty.


No solution is perfect.


Instruments may
be misleading.

Instruments may be overwhelming.

Instruments may be inaccurate.


Instruments may
be ignored.


What can
be done?


A little
paranoia never hurt
anyone.

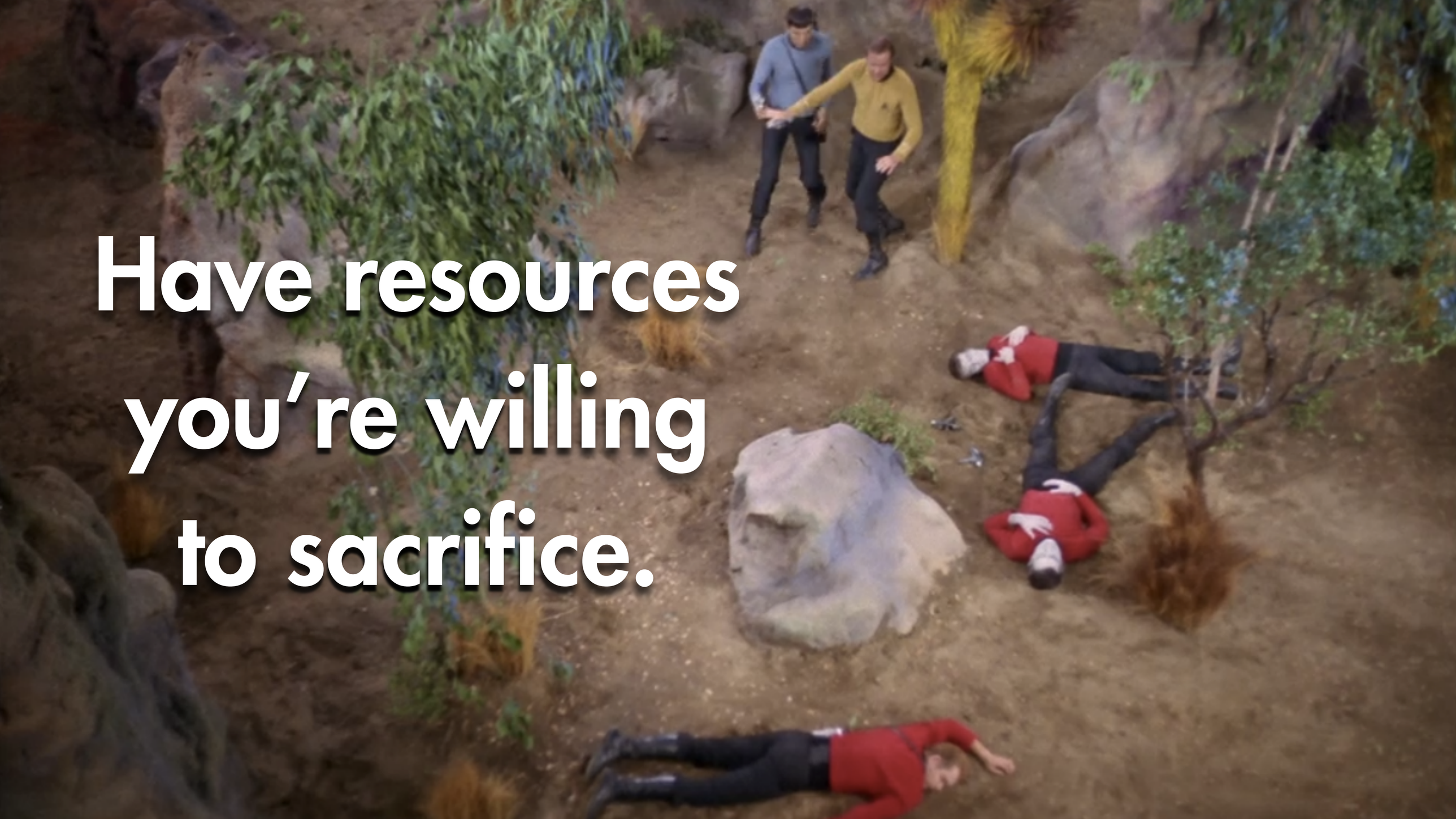

Use glass displays.

Train.


Trains to city
Forest Park
Trains to city
Forest Park
No smoking


Keep sight of
the main goal.


Recognize that
even simple
systems go
wrong
sometimes.
OTEL


Have resources
you're willing
to sacrifice.

Thanks, folks!

<3

@bltroutwine