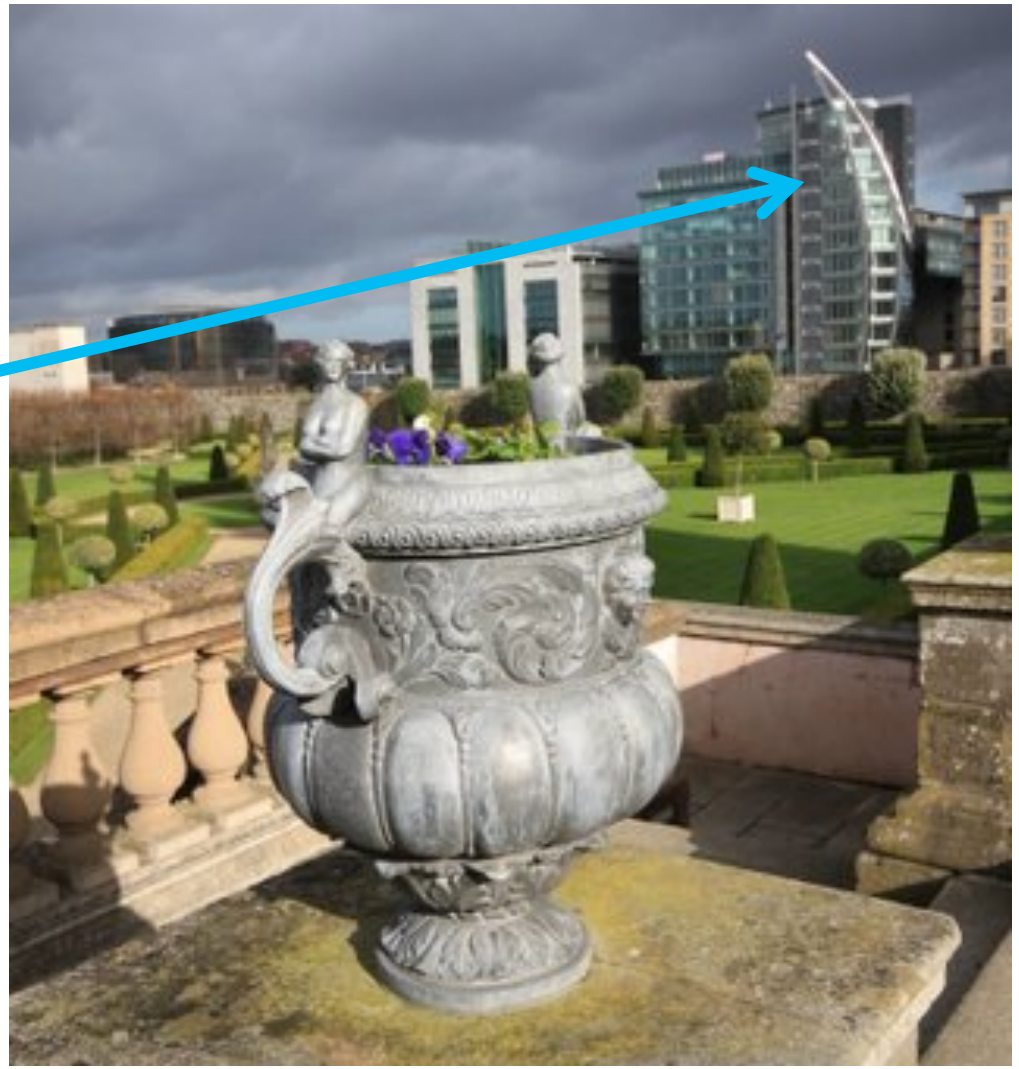
Real Time Bidding with Erlang

Aol PLATFORMS.

About Me

2012 Erlang at AOL

Erlang Here



Agenda

What is RTB ?

RTB Exchange Architecture

Bottlenecks

Tracing and Debugging

What is RTB ?

Online Advertising

Placement with Advertiser's Banner

Free Content on WWW

Paid for by Advertising

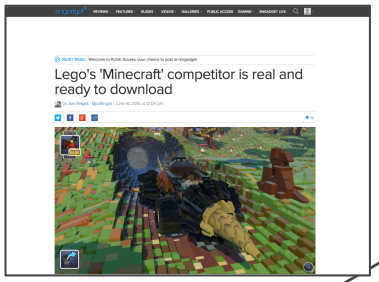
Upfront agreements between Publishers and Advertisers

What is Real Time Bidding

- The buying a selling of online advertisements in real time while a page is loading

Exoclick, www.exoclick.com (2014)

Adserving Workflow



Web Server

SSP

AdServer

RTB

DSP 1

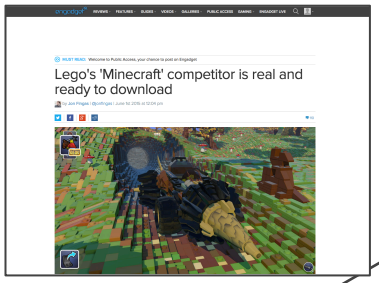
DSP 2

DSP 3

DSP N

Advertiser CDN

Adserving Workflow



Web Server

SSP

AdServer

RTB

DSP 1

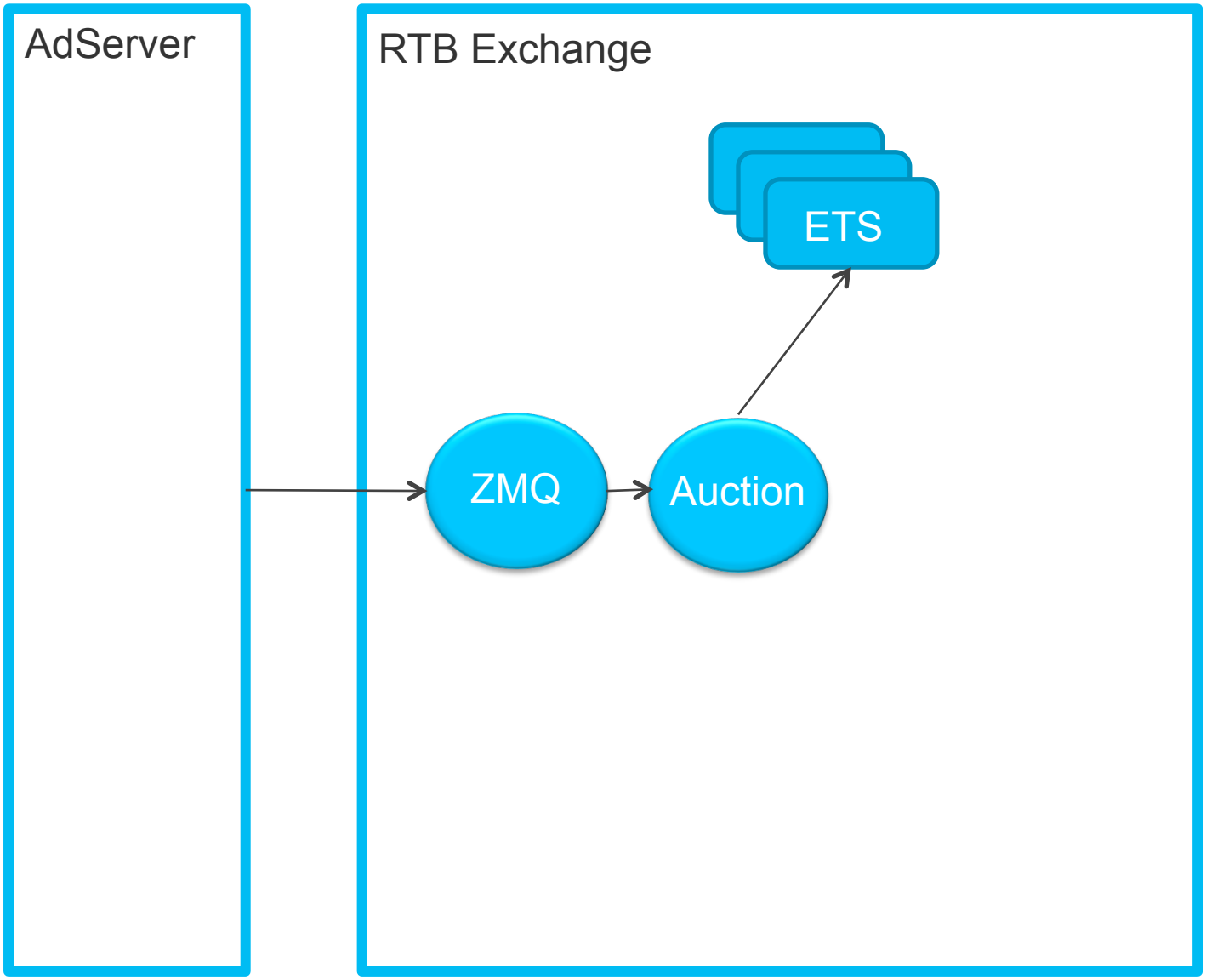
DSP 2

DSP 3

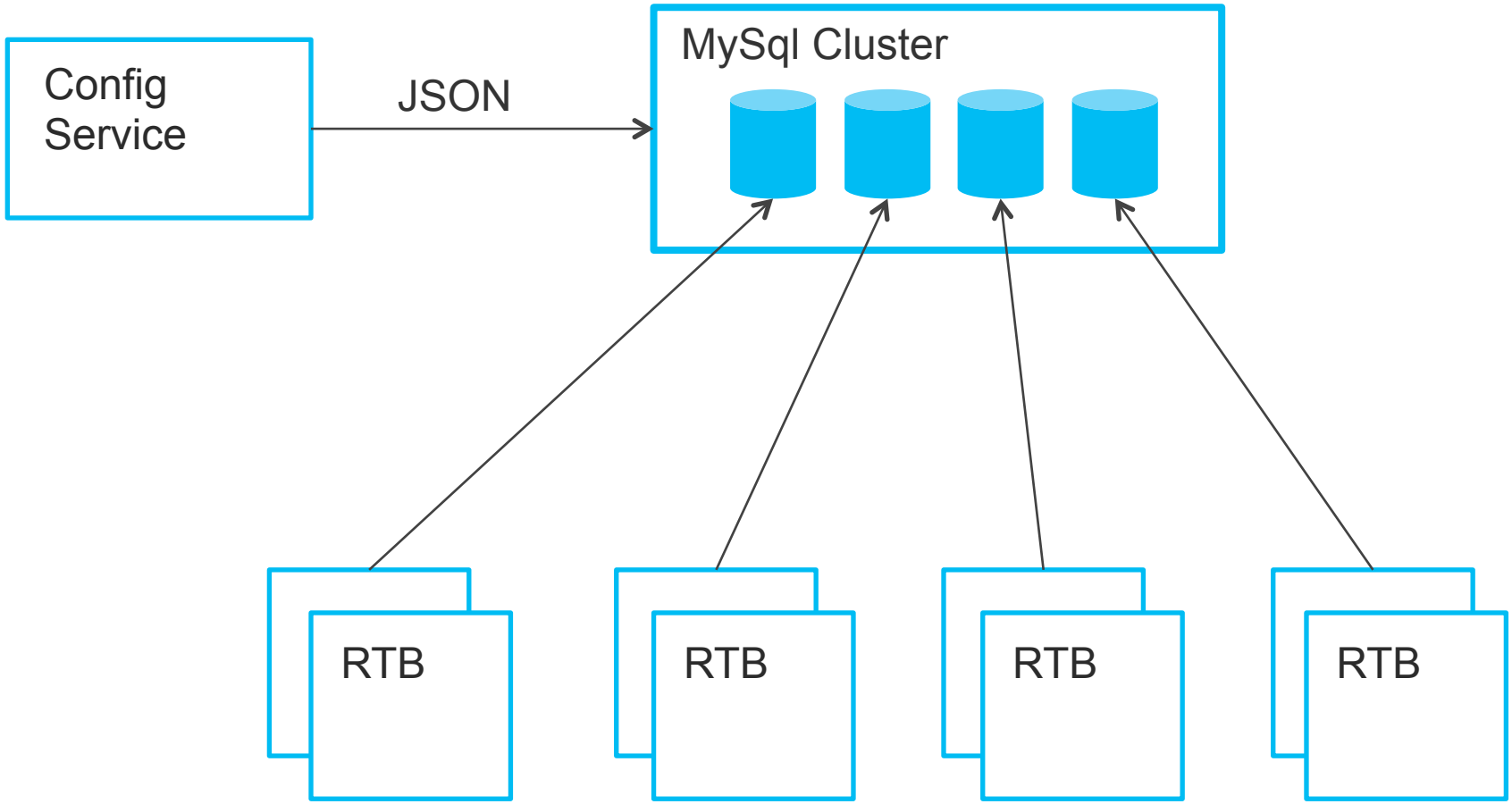
DSP N

Advertiser CDN

RTB Exchange Architecture

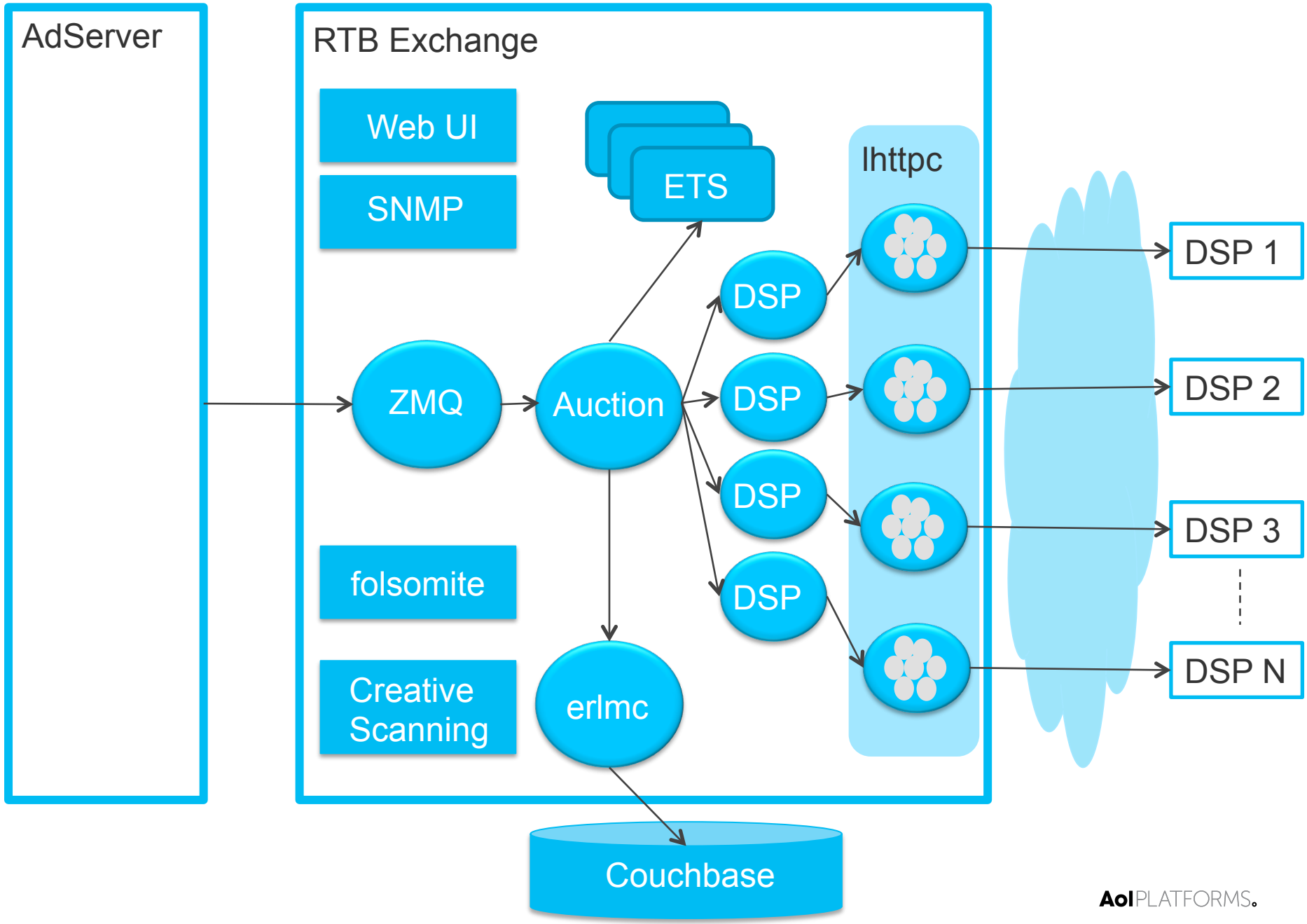

AdServer
RTB Exchange
ETS
ZMQ
Auction

Retrieving Campaign Data



Hashing Node Name to MySql Server

```
1 hash_name(Node, MySqlHosts) ->

2     Sum = lists:sum(atom_to_list(Node)),

3     Index = Sum rem length(MySqlHosts),

4     lists:nth(Index, MySqlHosts).
```


AdServer
RTB Exchange
Web UI
SNMP
ETS
lhttpc
DSP
DSP
DSP
DSP
ZMQ
Auction
folsomite
Creative Scanning
erlmc
Couchbase
DSP 1
DSP 2
DSP 3
DSP N

How Much does it Scale ?

8,000

The number of bid requests build and sent to DSPs every second per host at peak

1 Million

The number of placements

50 Billion

The number of bid requests build and sent
to DSPs every day

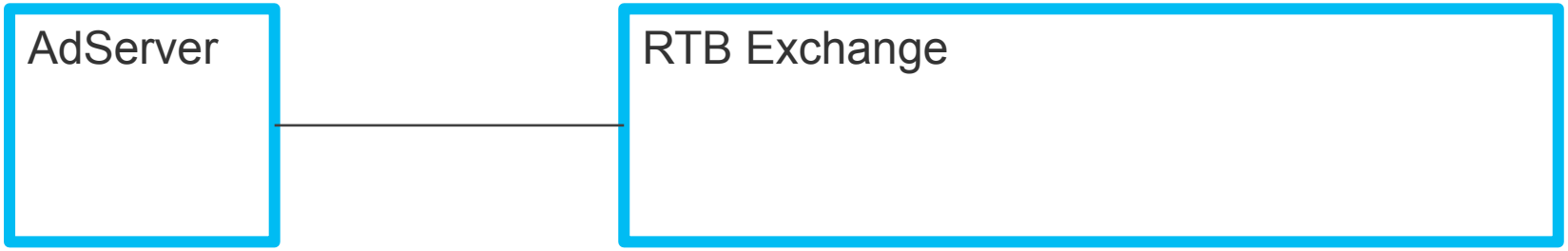
A Highly Concurrent System

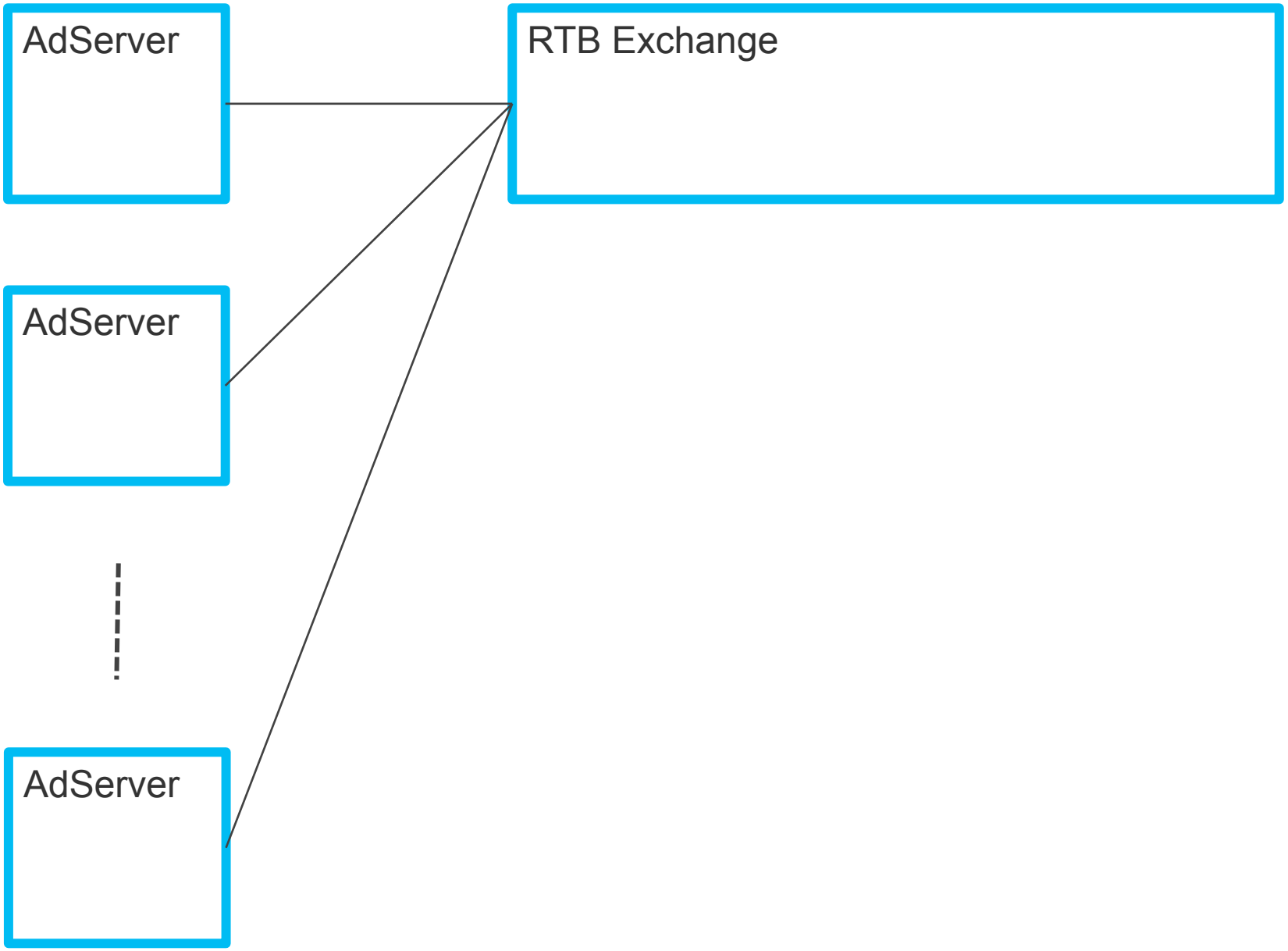
Independent Requests

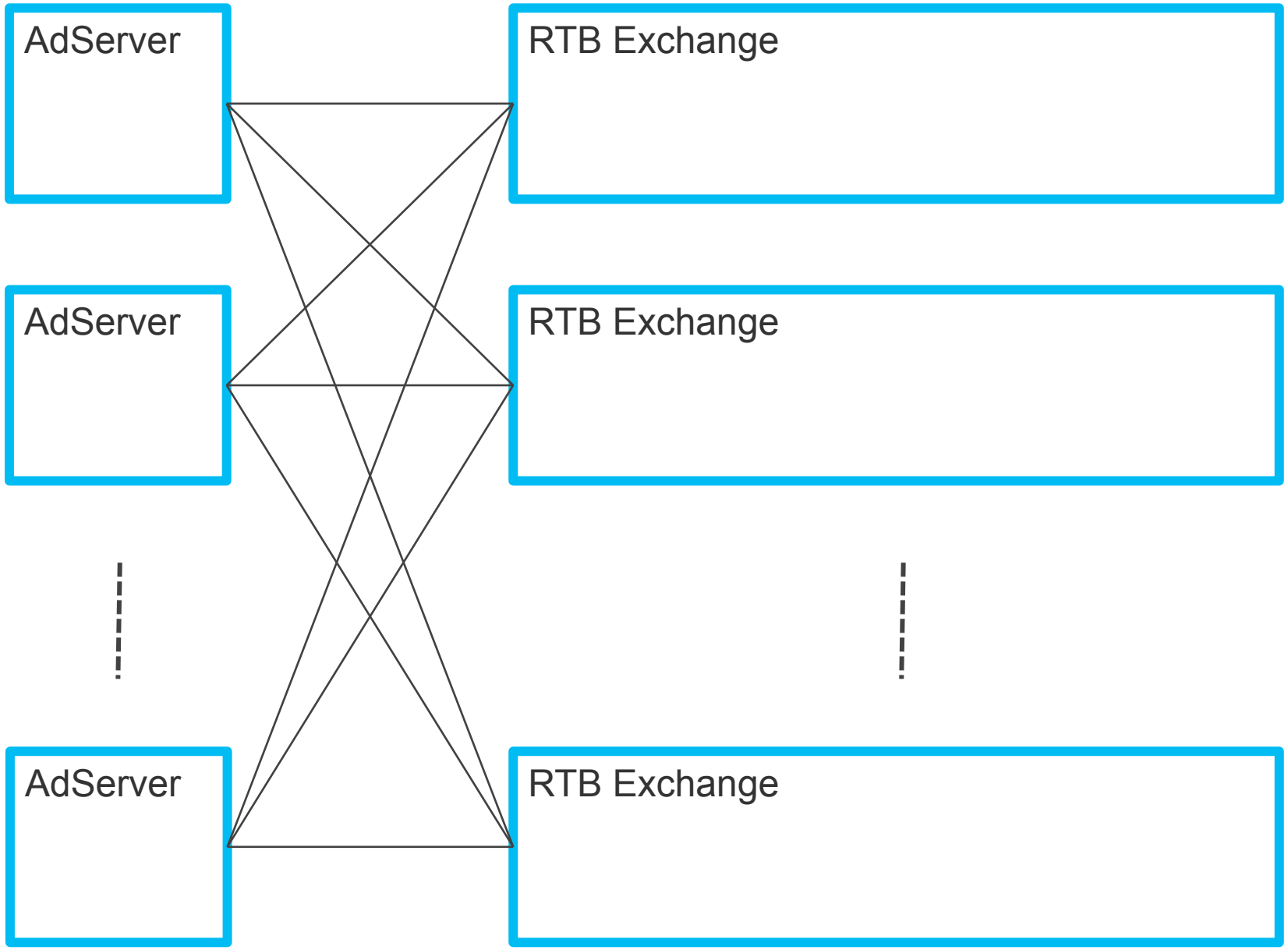
Independent Auctions

Independent DSP Processes

ZMQ Interface

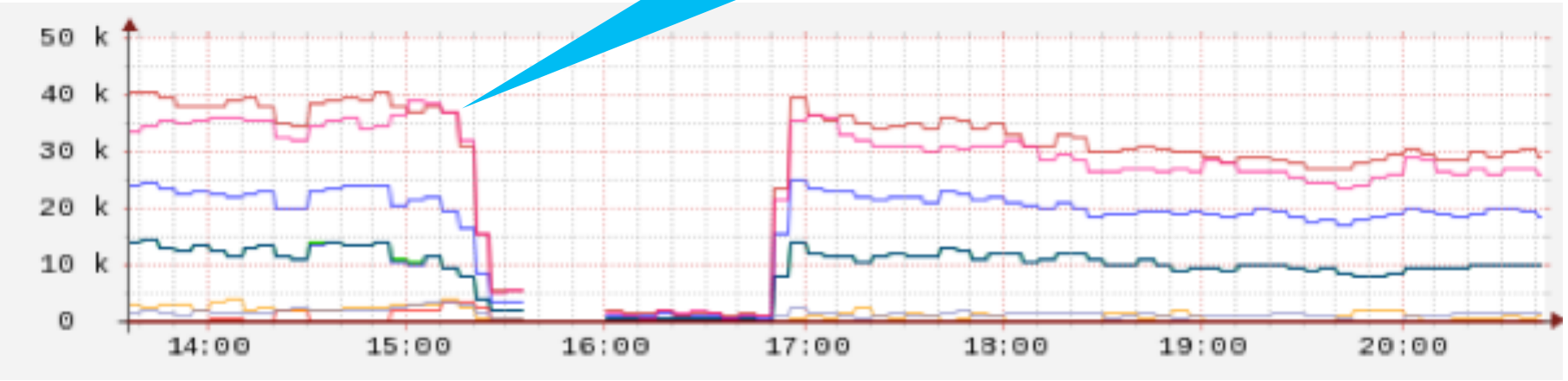

AdServer
RTB Exchange


AdServer
RTB Exchange
AdServer
AdServer


AdServer
AdServer
AdServer
RTB Exchange
RTB Exchange
RTB Exchange

The Symptom

Requests handled by
RTB Exchange
Drop off for 1.5 hours



The Problem

- 10,000s messages in ZMQ Mailbox
- Single Process
- Active Socket
- Mailbox full of messages being sent and received
- Selective receive when sending to adserver
- Other applications running on same host

How we Fixed It

escript for montoring

- RPC call to check message queue length
- erlang:halt/1 to shut down the node
- heart to Restart node

High Priority ZMQ Process

- process_flag(priority, high)
- Convert every request into a process ASAP

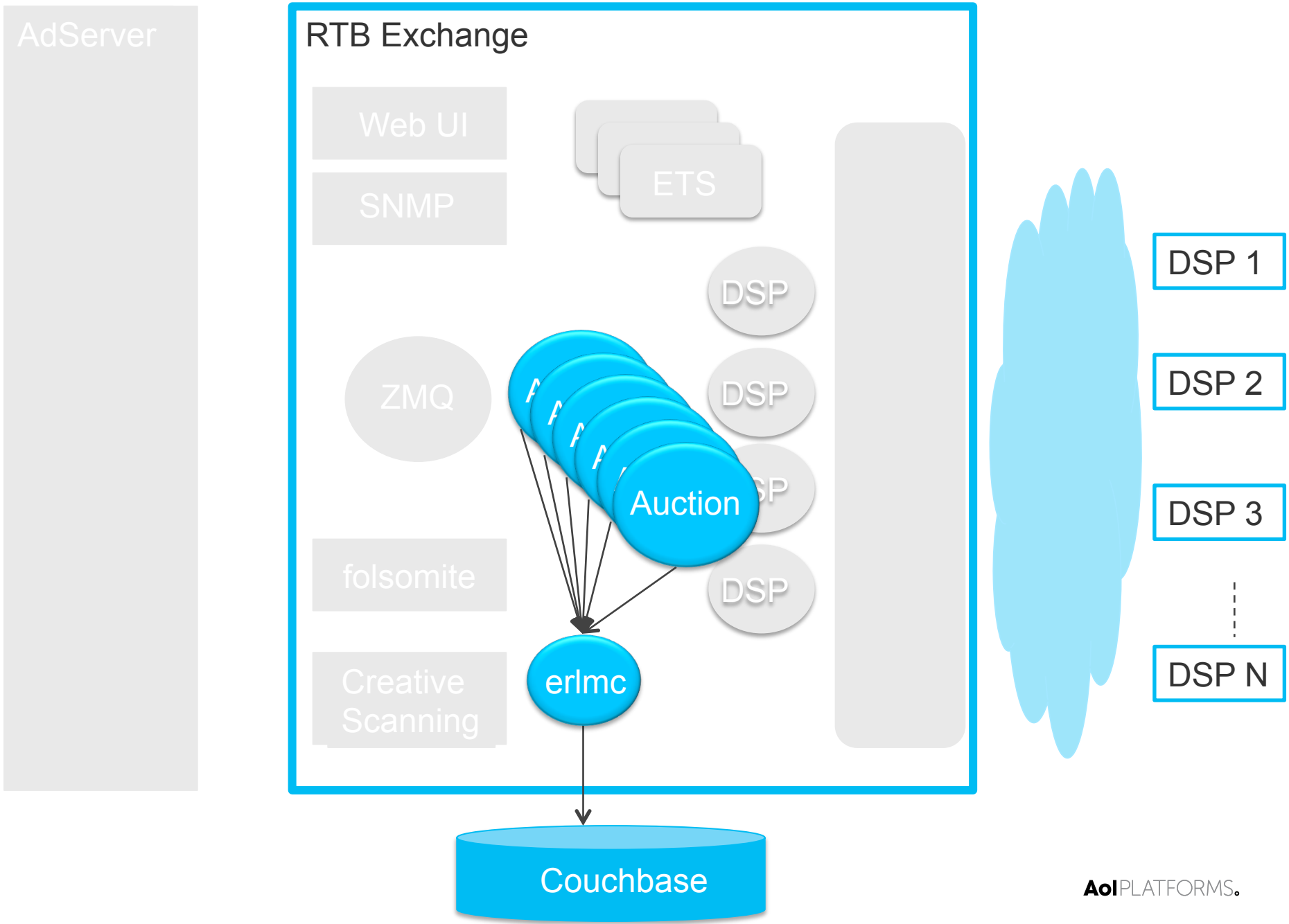
Passive Socket

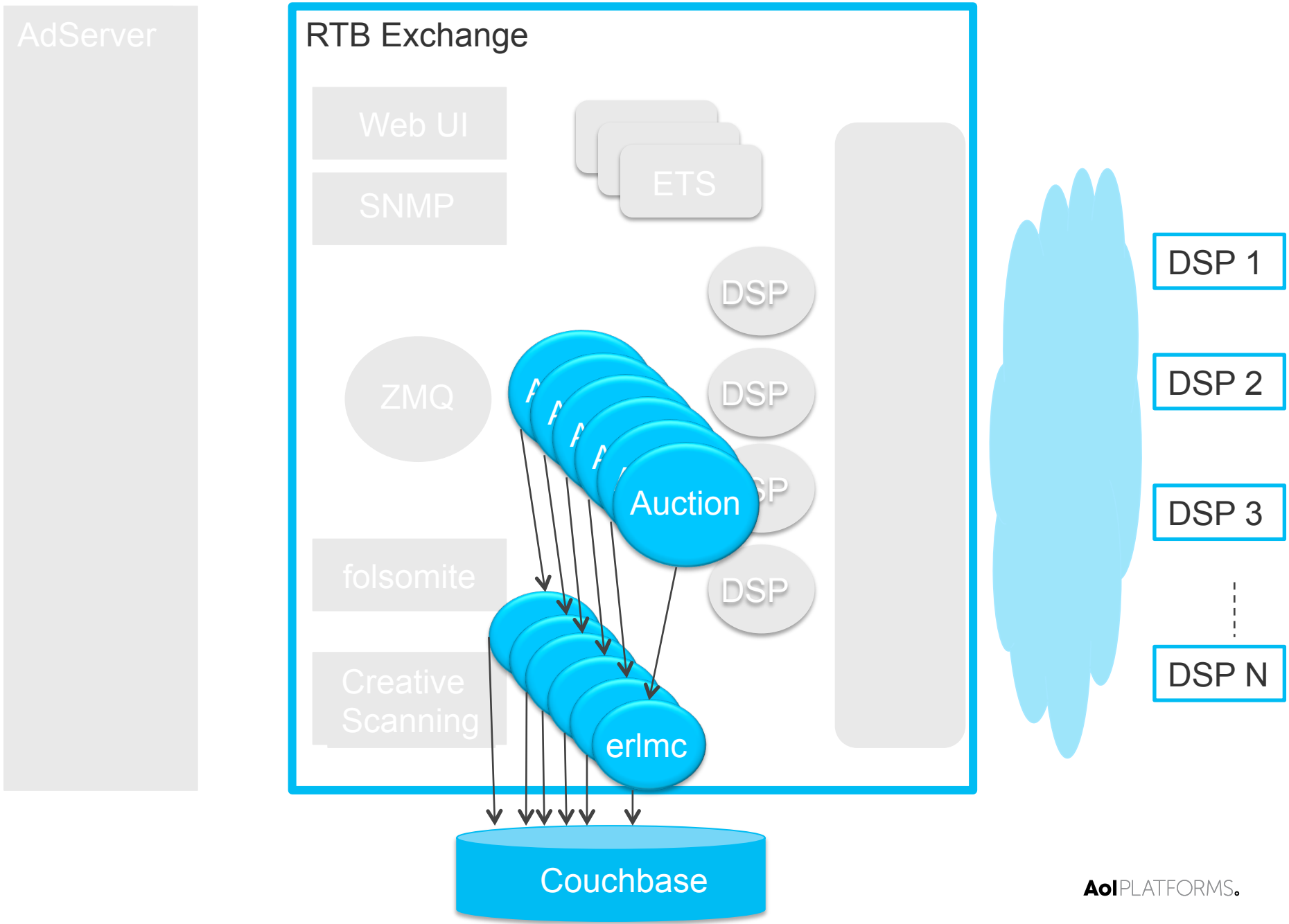
- Polling for requests
- ZMQ High Water Mark

Multiple ZMQ Endpoints

- Gives us >1 ZMQ process
- Enabled by configuration

Pooling Resources

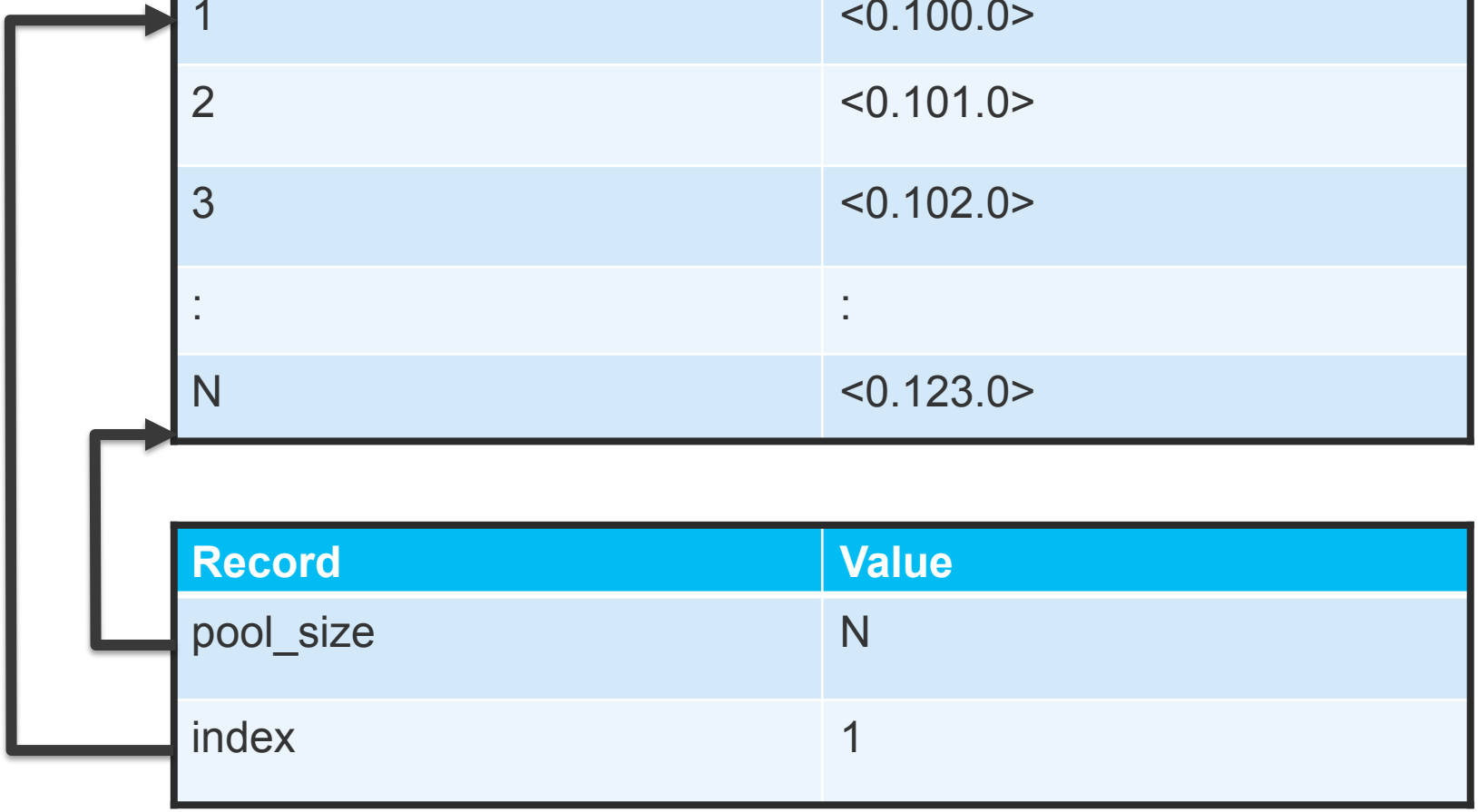

AdServer
RTB Exchange
Web UI
SNMP
ETS
DSP
DSP
ZMQ
Auction
folsomite
DSP
Creative Scanning
erlmc
Couchbase
DSP 1
DSP 2
DSP 3
DSP N


AdServer
RTB Exchange
Web UI
SNMP
ETS
DSP
DSP
DSP
ZMQ
Auction
folsomite
Creative Scanning
erlmc
Couchbase
DSP 1
DSP 2
DSP 3
DSP N

Storing Resources in ETS

Index	erlmc process
1	<0.100.0>
2	<0.101.0>
3	<0.102.0>
:	:
N	<0.123.0>

Record	Value
pool_size	N
index	1



Round Robin Lookup in ETS

```
1 get_worker() ->
2   PoolSize = ets:lookup_element(?TABLE, pool_size, 2),
3   Index = ets:update_counter(?TABLE, index, {2,1,PoolSize,1}),
4   ets:lookup_element(?TABLE, Index, 2).
```

Random ETS Lookup

```
1 get_worker() ->
2   PoolSize = ets:lookup_element(?TABLE, pool_size, 2),
3   Index = erlang:phash2(self(), PoolSize) + 1,
4   ets:lookup_element(?TABLE, Index, 2).
```

phash2 vs update_counter

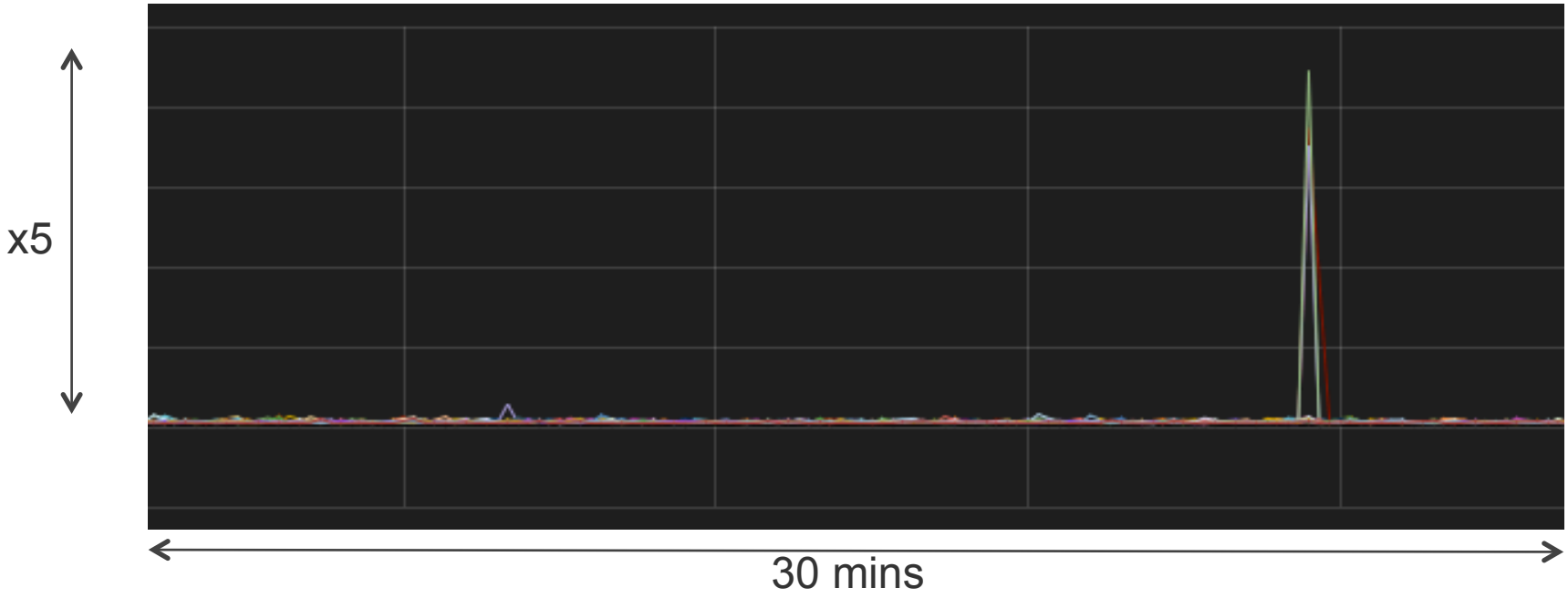
update_counter useful for round_robin

phash2 useful for random selection

phash2 faster than update_counter (x1.5 - x2)

phash2 does no writes to ETS hence no locking

Tracing and Debugging



Bid Collection Time

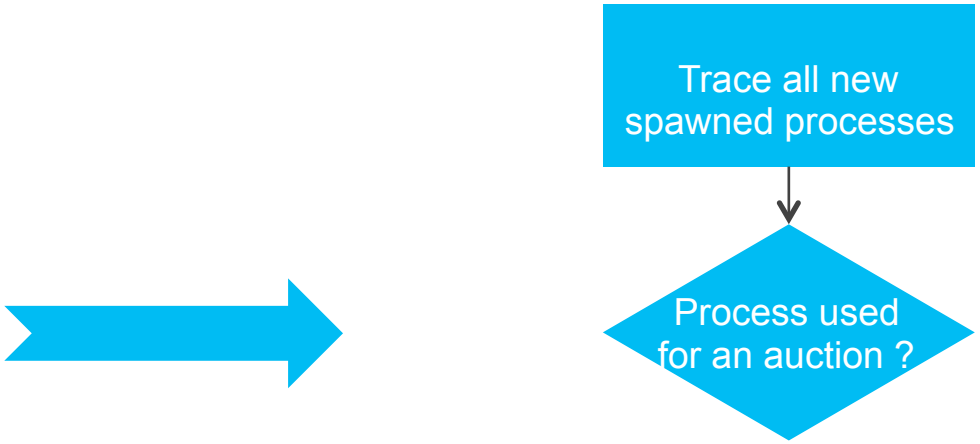
- Spike in Timer every 20-30 mins
- X5 magnitude
- System not under load

Tracing the Spike

Trace all new spawned processes

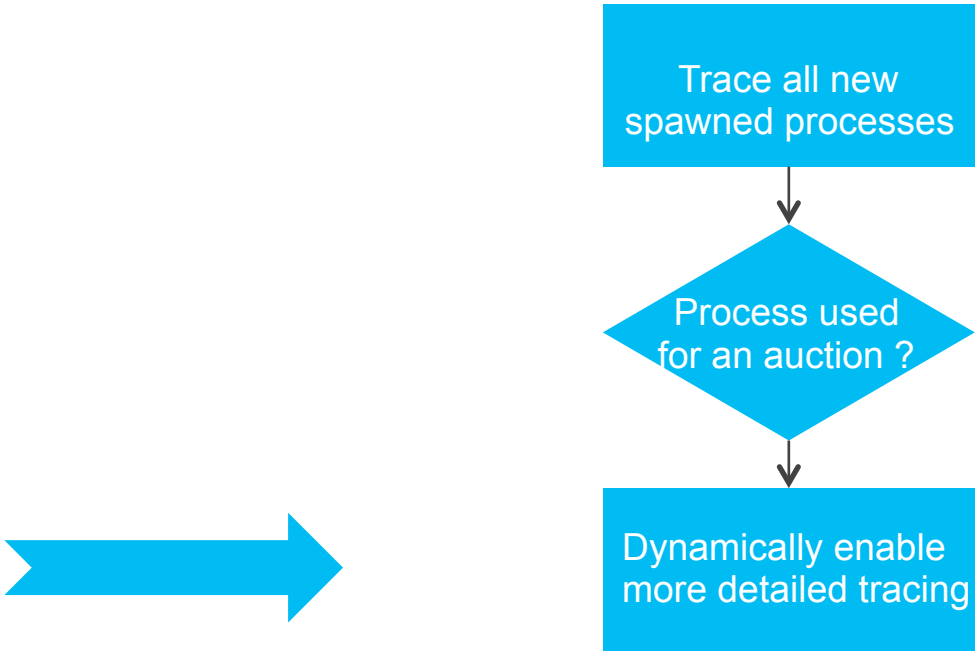
```
1     dbg:tracer(process,
                 {fun trace_handler/2,
                  {IO,           %% file to log to
                   dict:new(),   %% dict for recording info in
                   Threshold,    %% timer threshold
                   Count}}),

2     {ok, Tracer} = dbg:get_tracer(),

3     erlang:trace(new, true, [call, timestamp, {tracer, Tracer}]).
```

Tracing the Spike



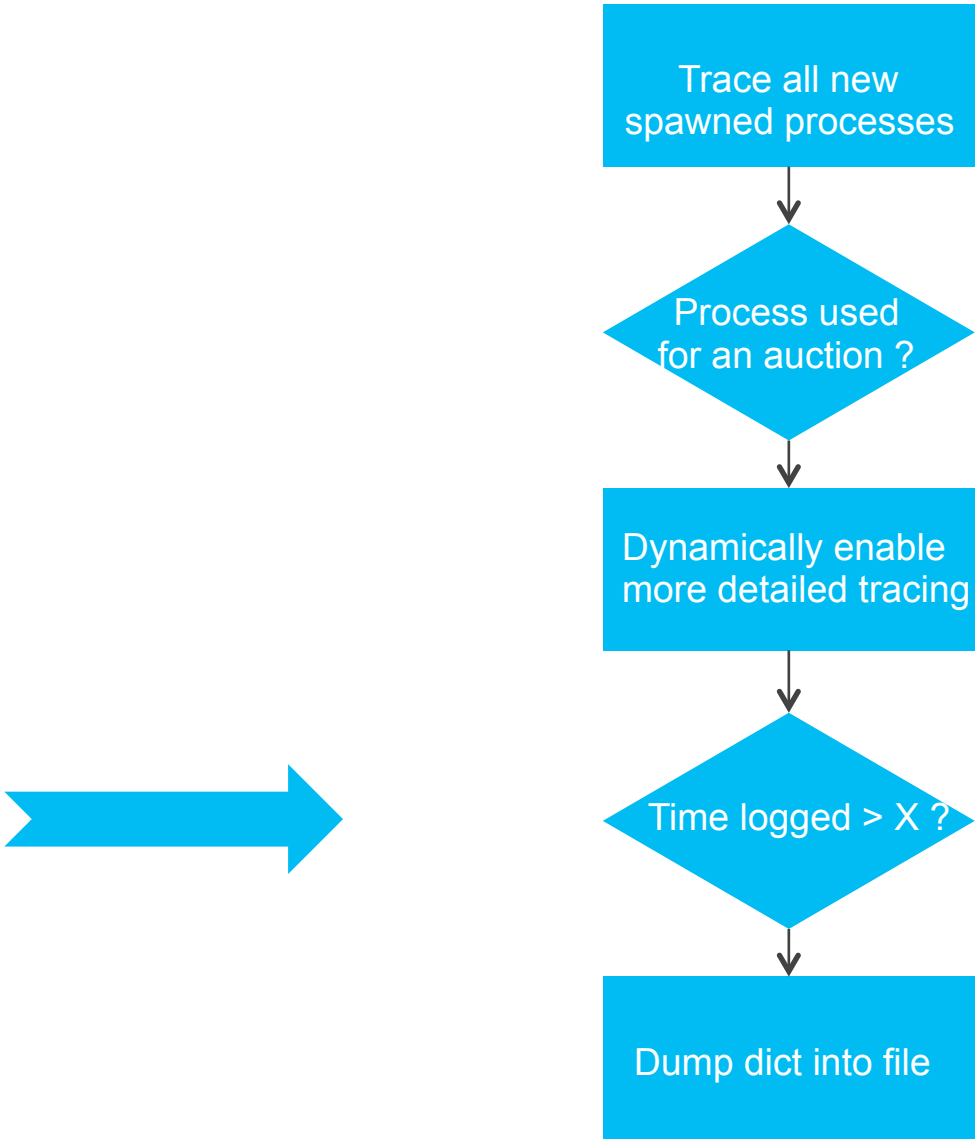
```
1    dbg:tp(bid_processor, process_request, x),
```

Tracing the Spike



```
1 trace_handler({trace_ts, Pid, call,
                 {bid_processor, process_request, _},
                 TS},
                 {IO, Dict, Threshold, Count}) ->

2     erlang:trace(Pid, true, [timestamp,  %% trace with timestamps
                               'receive',  %% messages received
                                call,      %% funcion calls
                                running,   %% when scheduled in/out
                                {tracer, self()}]),

3     Dict1 = dict:append(Pid, {TS, process_request}, Dict),

4     {IO, Dict1, Threshold, Count};
```

Tracing the Spike



Conclusions

Highly Concurrent System

Single Processes are bottlenecks

- High Priority
- Process Pools

Erlang VM a Great Platform for Debugging Production Systems

The Future

Scale for more Publishers and Demand Partners

Continue to Build a Great Erlang Team

Support Growth Erlang Community

- Internships
- University Presentations

Erlang Factory - Dublin Sept 2015