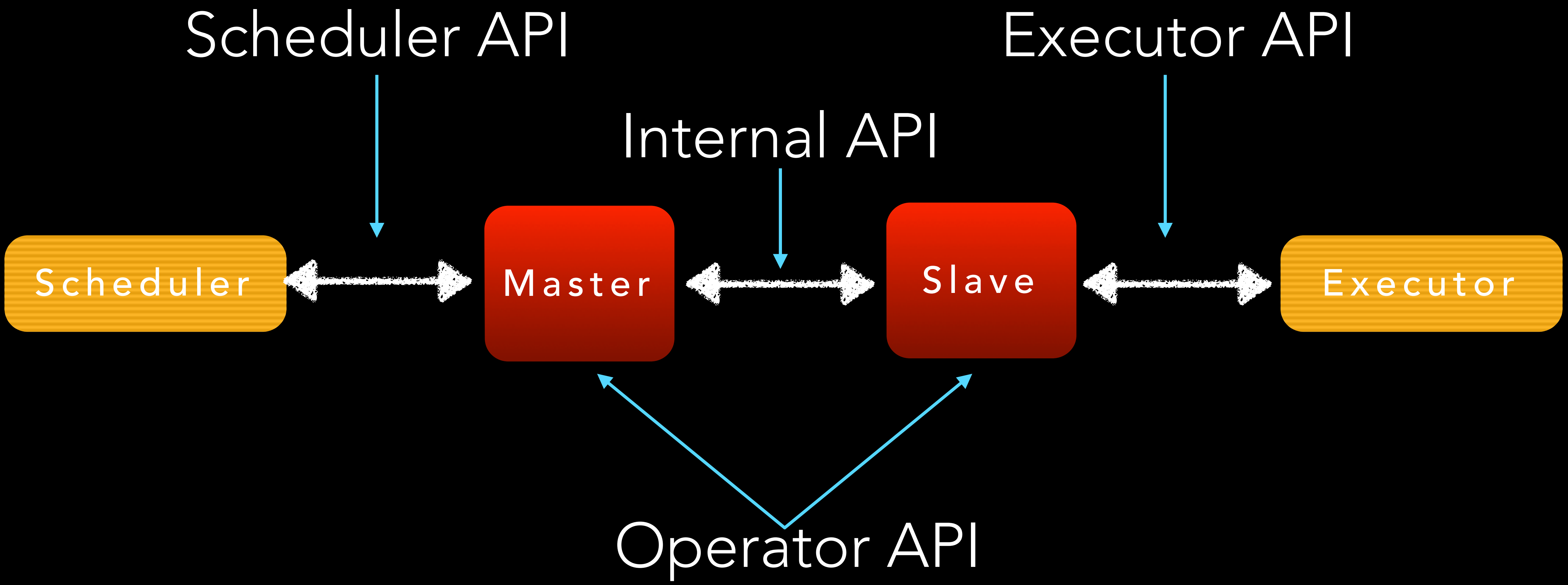
NEW

MESOS HTTP API

@vinodkone
@ijimene

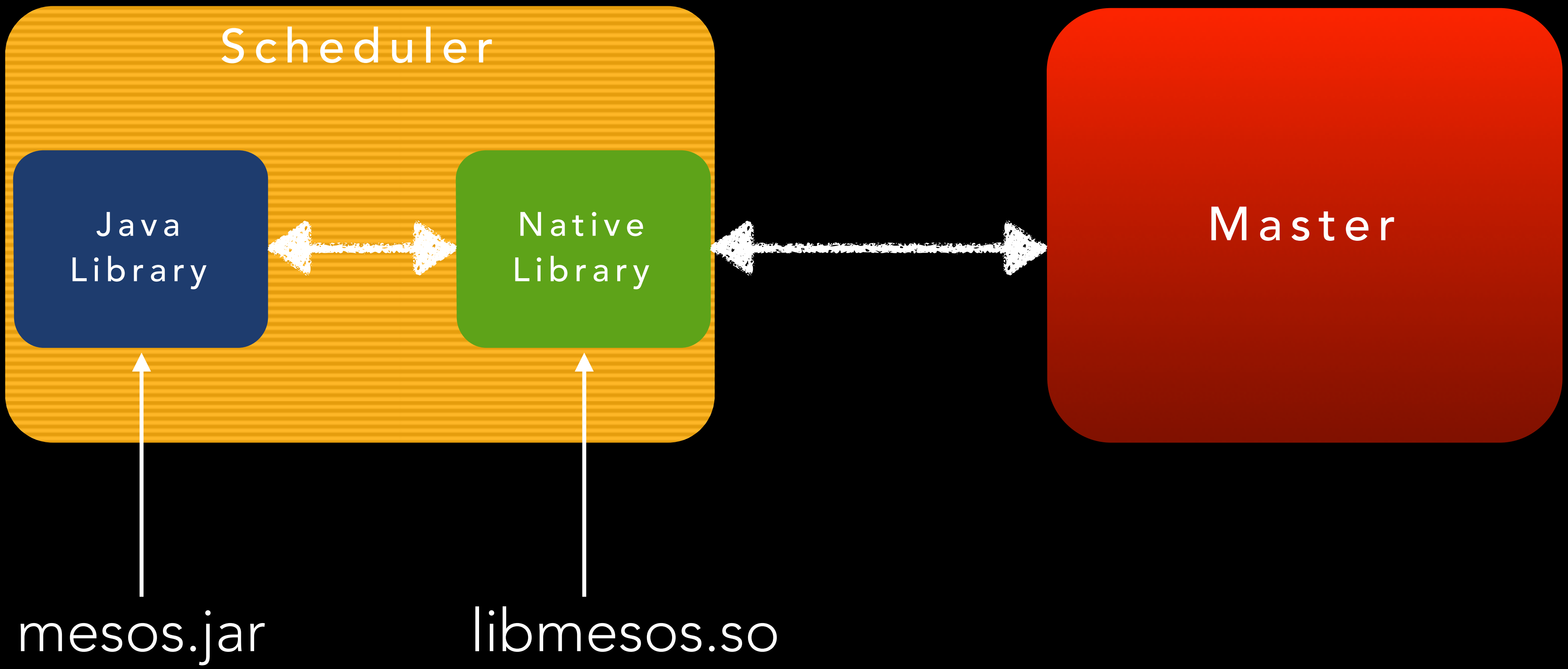
Mesos 1.0 IS COMING

MESOS APIS


Scheduler API
Executor API
Internal API
Scheduler
Master
Slave
Executor
Operator API

Once you've accepted
your flaws,
no one can use them
against you.

DEPENDENCE ON NATIVE LIBRARY



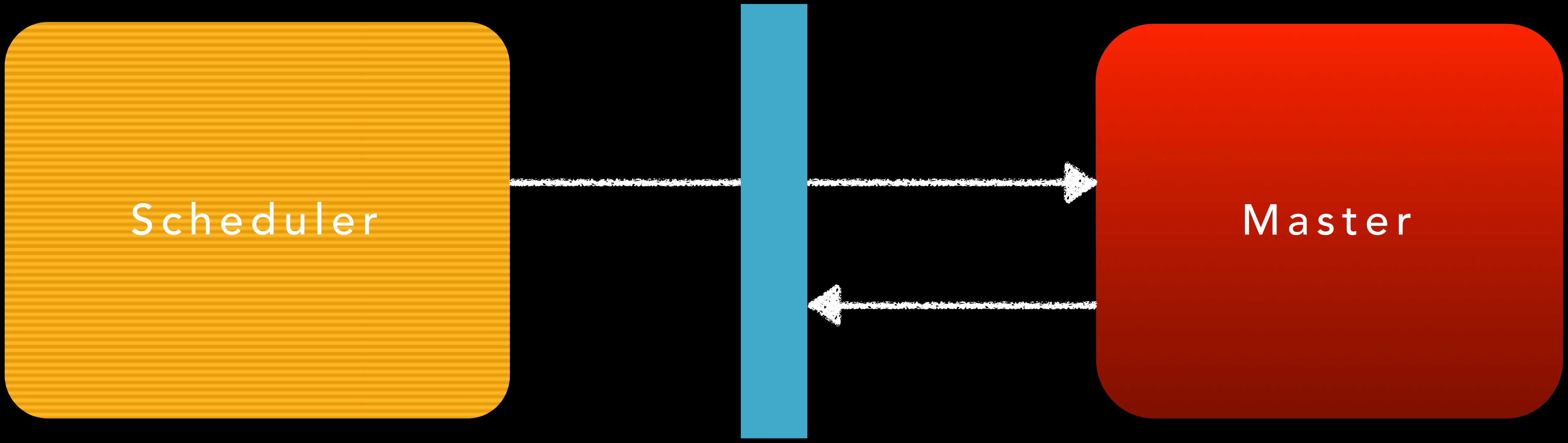
Hard to debug

Not portable

NON-STANDARD FRAMEWORK API

```
POST /master/mesos.internal.LaunchTasksMessage HTTP/1.1

User-Agent: libprocess/scheduler-1234-23-23342@127.0.0.1:8081

Libprocess-From: scheduler-1234-23-23342@127.0.0.1:8081

Connection: Keep-Alive

Host:
Transfer-Encoding: chunked
```

NETWORKING CONSTRAINTS



Firewall

Containers :(

Network partitions :(

LACK OF API VERSIONING

`GET /metrics/snapshot` $\longrightarrow$ JSON

Version ❓

`GET /state.json` $\longrightarrow$ “version” : 0.23.0

PITA FOR MESOS DEVS

Lot of boiler plate to add new calls/events

Forced upgrade dependencies

"You either die a hero…
or live long enough to see yourself ~~become the villain~~"
replaced by a better API

–HARVEY DENT

1.0 GOALS

Consistent APIs

Versioning

NEW HTTP API

- Standard HTTP 1.1
- Versioned !
- Well documented

NEW MESOS APIS

Endpoint	API	Hosted by
/api/v1/scheduler	Scheduler API	Master
/api/v1/executor	Executor API	Slave
/api/v1/internal	Internal API	Master
/api/v1/admin	Operator API	Master / Slave

SCHEDULER HTTP API

- Based on Calls and Events
- Scheduler opens connections to the master
  - A persistent connection to receive events
  - One (or more) connection(s) to send *calls*

RATIONALE

Simplicity

Off-the-shelf HTTP client libraries

No native dependencies

RATIONALE

Upgradability

Familiar to existing APIs

killTask() $\xrightarrow{\quad}$ Call.Kill

RATIONALE

Extensibility

Easy to add support for new features

CALLS	Old API
SUBSCRIBE	start()
TEARDOWN	stop()
ACCEPT	acceptOffers()
DECLINE	declineOffer()
REVIVE	reviveOffers()
KILL	killTask()
SHUTDOWN	* Shutdown executor *
ACKNOWLEDGE	acknowledgeStatusUpdate()
RECONCILE	reconcileTasks()
MESSAGE	sendFrameworkMessage()
REQUEST	requestResources()

EVENTS	Old API
SUBSCRIBED	registered() / reregistered()
OFFERS	resourceOffers()
RESCIND	offerRescinded()
UPDATE	statusUpdate()
MESSAGE	frameworkMessage()
FAILURE	slaveLost() / executorLost()
ERROR	error()
HEARTBEAT	* Periodic heartbeats *

PROTOCOL

- Every call is a HTTP POST request
  - application/json or application/x-protobuf
- SUBSCRIBE call results in a "200 OK" ***streaming*** response
  - Record-IO formatted events
  - chunked encoding
- All successful non-SUBSCRIBE calls result in "202 Accepted"

SUBSCRIPTION REQUEST

```
POST /api/v1/scheduler  HTTP/1.1

Host: masterhost:5050
Content-Type: application/json
Accept: application/json
Connection: close

{
   “type”                 : “SUBSCRIBE”,

   “subscribe” : {
      “framework_info”   : {
           “user” :  “foo”,
            “name” :  “Example HTTP Framework”
      },

      “force” : true
  }
}
```

SUBSCRIPTION RESPONSE

```
HTTP/1.1 200 OK

Content-Type: application/json
Transfer-Encoding: chunked

<Event Length>
{
 "type"            : "SUBSCRIBED",
 "subscribed"       : {
    "framework_id"                        :  {"value":"12220-3440-12532-2345"},
    "heartbeat_interval_seconds" : 15
  }
}
<more events>
```

KILL

Response: HTTP/1.1 202 Accepted

```
POST /api/v1/scheduler  HTTP/1.1

Host: masterhost:5050
Content-Type: application/son
Accept: application/json
Connection: close

{
  “framework_id” : {“value” : “12220-3440-12532-2345”},

  “type”                  : “KILL”,

  “kill”              : {
    “task_id”                   :  {“value” : “12220-3440-12532-my-task”},
    “slave_id”                  :  {“value” : “12220-3440-12532-S123345”}
  }
}
```

DISCONNECTIONS & PARTITIONS

- Master tracks the persistent subscription connection
    - Reconnect within failover timeout
- Subscribe.force : Failover
- Periodic HEARTBEATs sent by master

VERSIONING

Explicit

Simple

/api/v1/scheduler

/api/v1/executor

/api/v1/admin

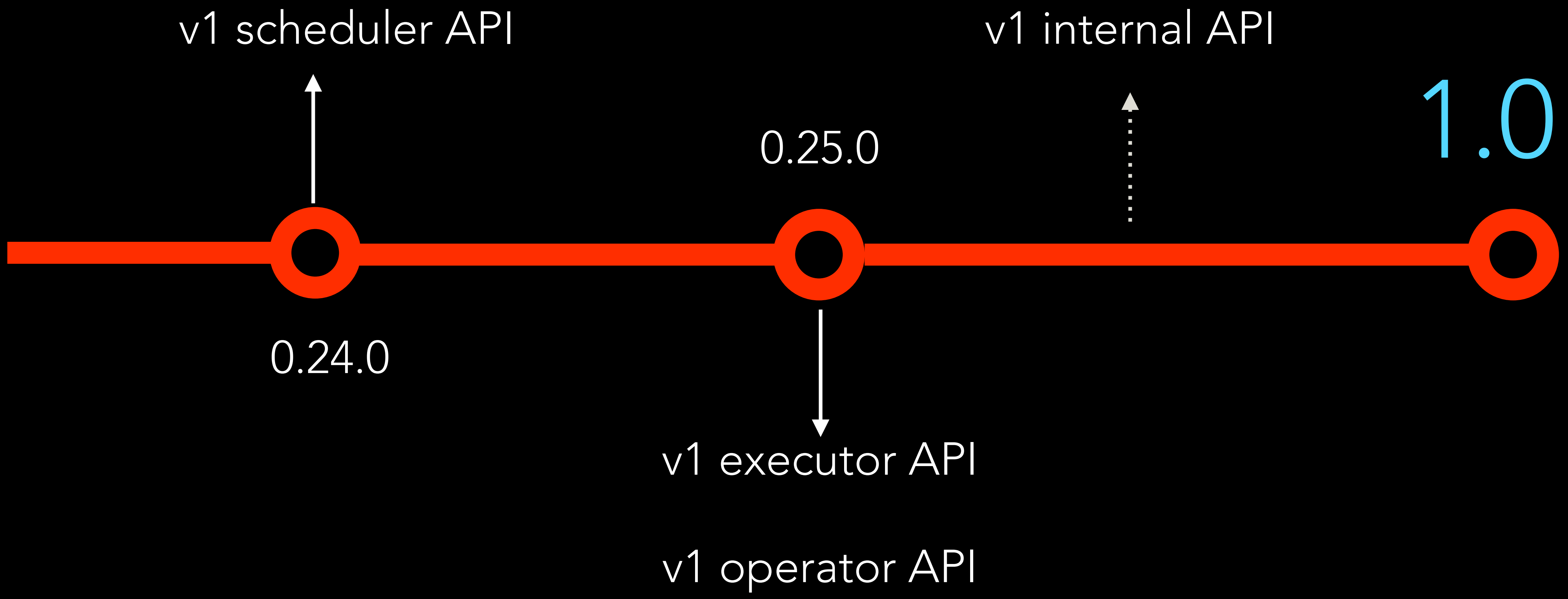
/api/v1/internal

Avoid version explosion

API VERSION VS RELEASE VERSION

- API version == Major release version
  - v1 API supported by 1.0.0, 1.4.0, 1.20.0
- vN API released in N-1 release version
  - vN API considered stable in the last N-1 release
- Version bumping
  - Major/API version bumped for backwards incompatible changes (> yearly)
  - Minor version bumped regularly (4-8 weeks)

ROAD TO 1.0


v1 scheduler API
v1 internal API
1.0
0.25.0
0.24.0
v1 executor API
v1 operator API

DEMO

IT'S ALIVE!
quickmeme.com

RATIONALE

Simplicity

RATIONALE

Simplicity

Upgradability

RATIONALE

Simplicity

Upgradability

Extensibility

ACKNOWLEDGEMENTS

THANK YOU

MESOS-2288