VERT.X

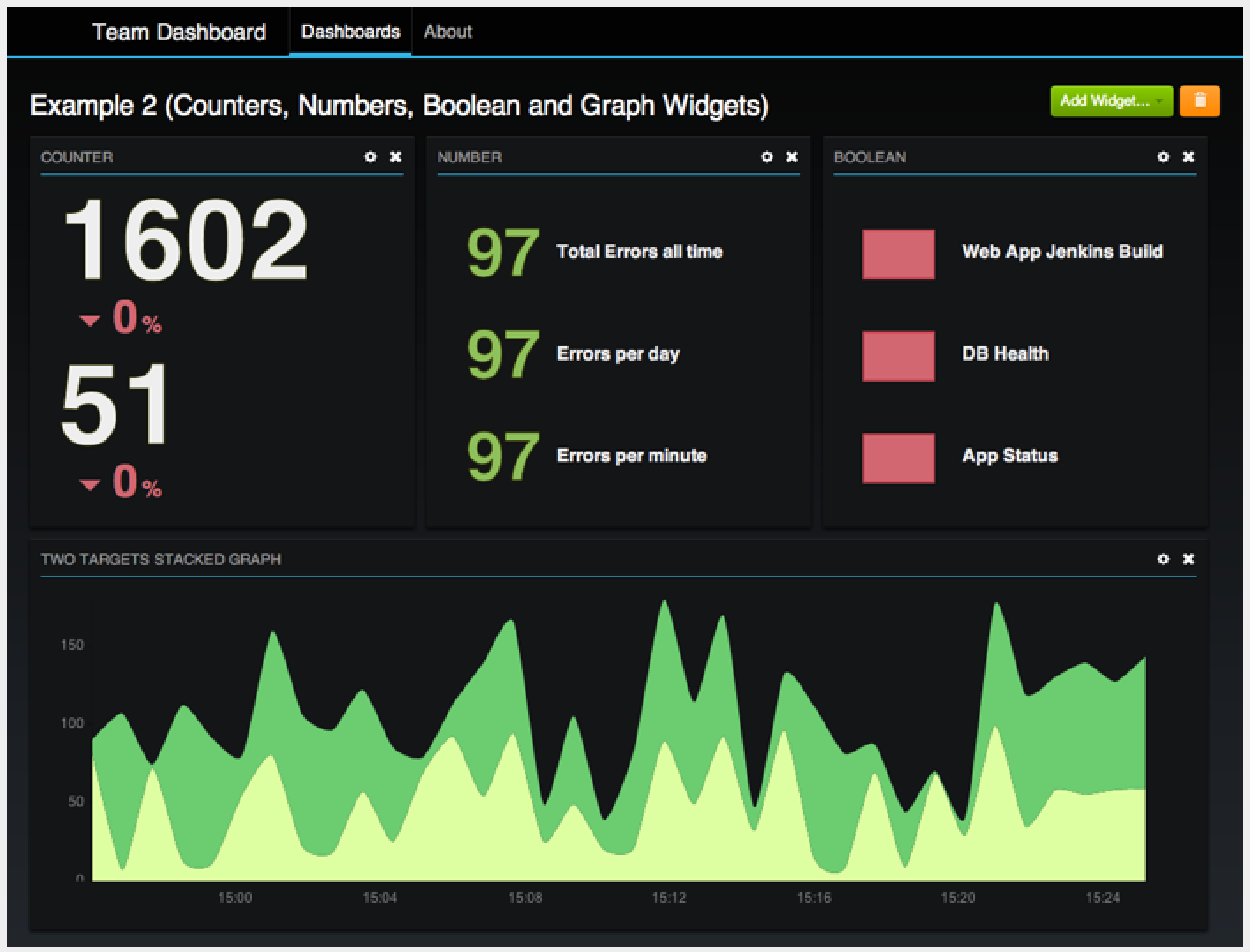
MICHAEL KUTY

MICHAEL KUTY

- Web application developer
- TCP Cloud - Django, SaltStack, ..
- Robotice - Python, Celery, ..
- https://github.com/michaelkuty

NODE.JS FOR JVM

WHY ?



Example 2 (Counters, Numbers, Boolean and Graph Widgets)

COUNTER

1602
▼ 0%

51
▼ 0%

NUMBER

97 Total Errors all time

97 Errors per day

97 Errors per minute

BOOLEAN

🟥 Web App Jenkins Build

🟥 DB Health

🟥 App Status

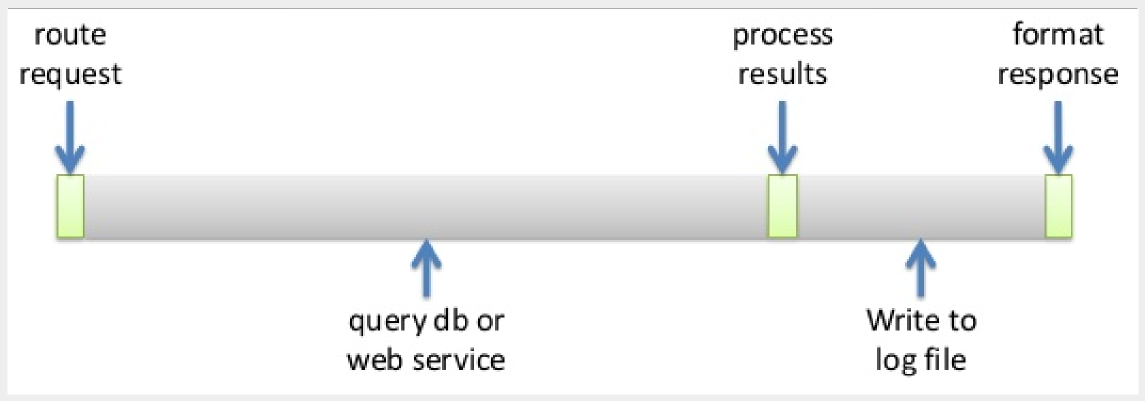
TWO TARGETS STACKED GRAPH

ADDRESSING THE PROBLEM

- C10K problem
- Server push
- Many long lived connections

2008

REQUEST WAITING ... RESPONSE



Thread per Request/Response

REACTOR PATTERN

- Event loop
- (Douglas C. Schmidt, 1995)

1995

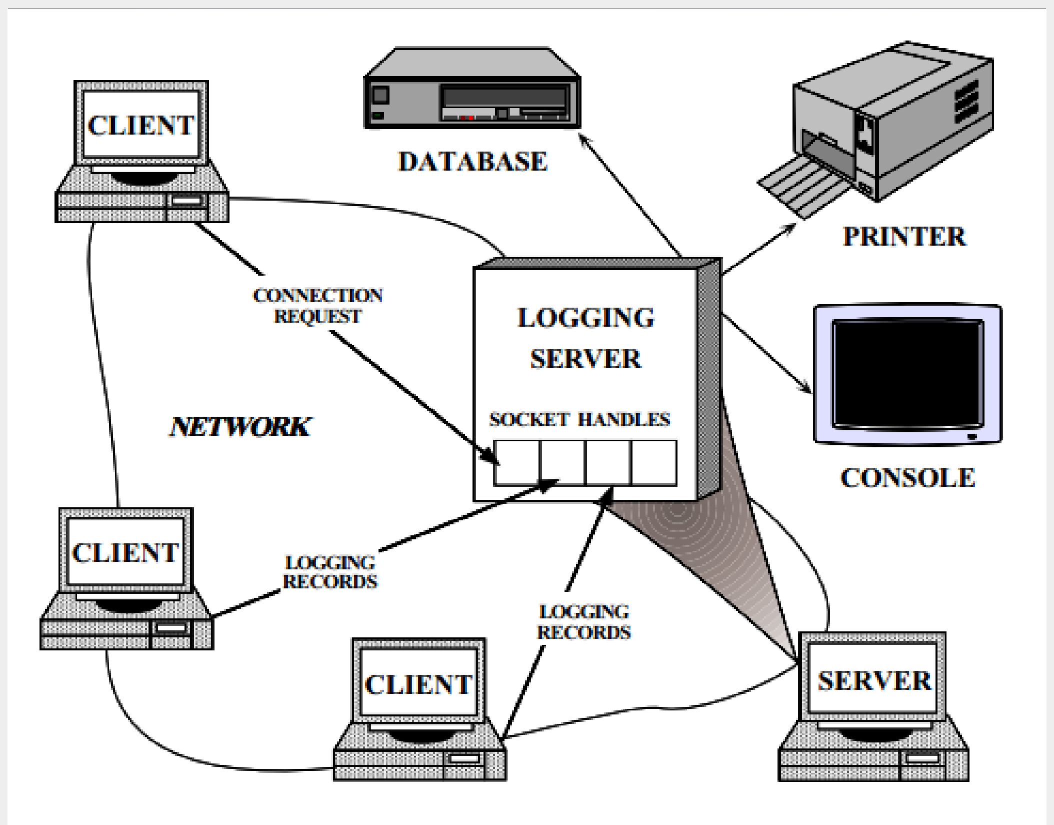
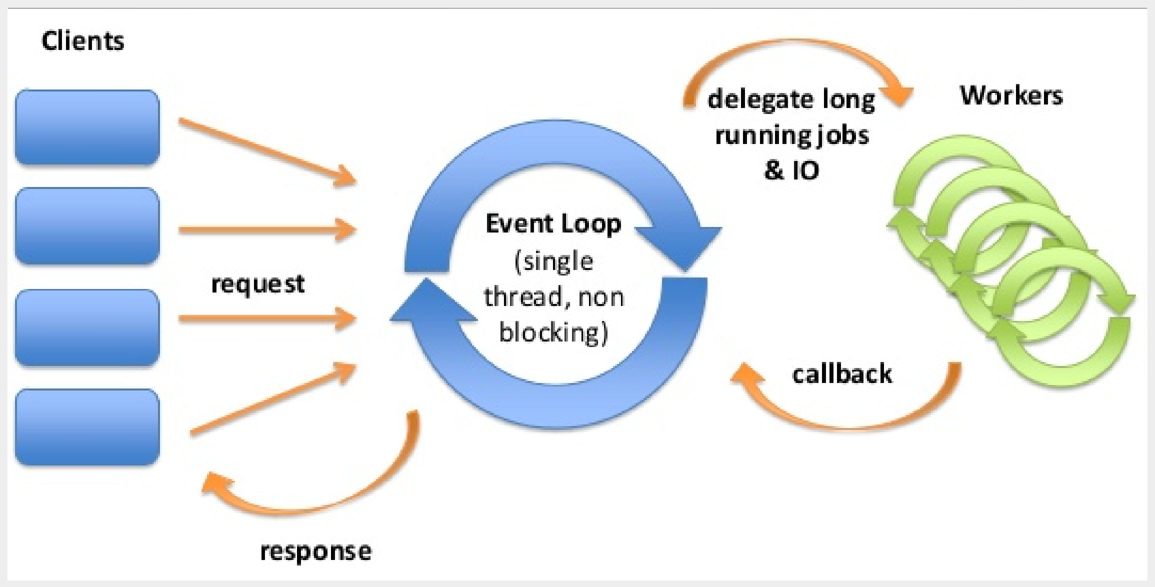
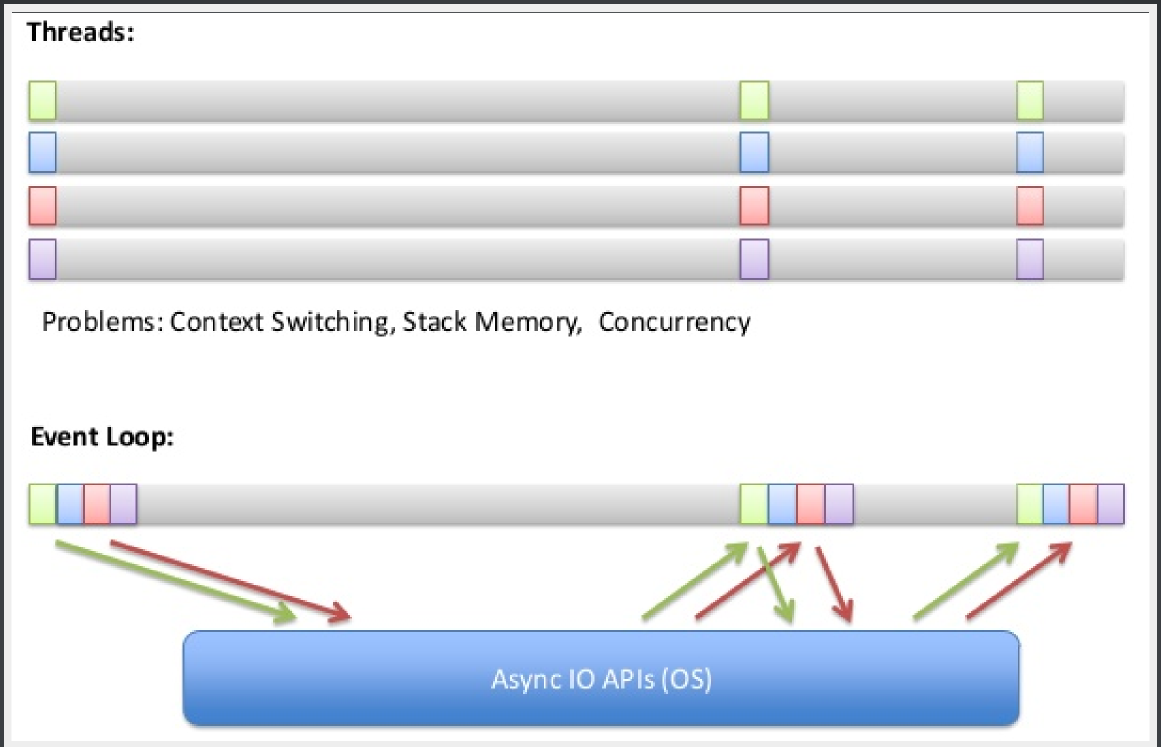


Figure 1: Distributed Logging Service

... LOOP


Clients
request
response
Event Loop
(single
thread, non
blocking)
delegate long
running jobs
& IO
Workers
callback



Threads:

Problems: Context Switching, Stack Memory, Concurrency

Event Loop:

Async IO APIs (OS)

SOLUTIONS

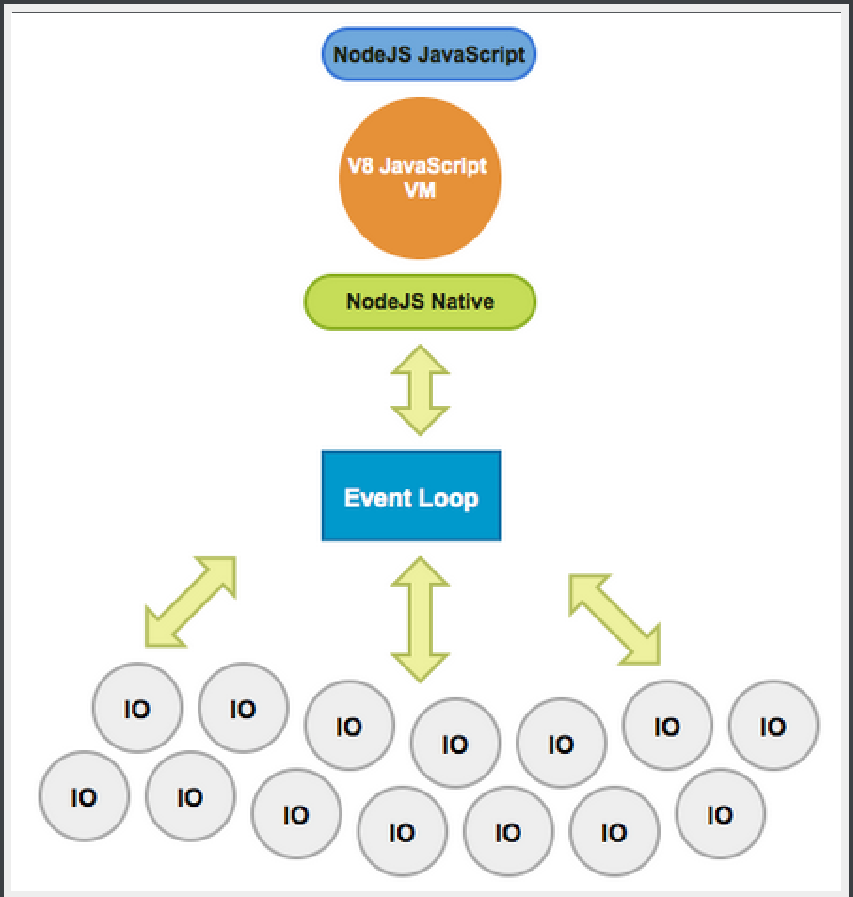
SERVERS

- Nginx
- Jetty

PLATFORMS

- Node.js (JavaScript, JXCore)
- Tornado, Eventlet (Python)
- EventMachine (Ruby)

2009 - NODE.JS


NodeJS JavaScript
V8 JavaScript VM
NodeJS Native
Event Loop
IO
IO
IO
IO
IO
IO
IO
IO
IO
IO
IO
IO
IO
IO
IO

2011 - VERT.X

GENERAL

- Asynchronous by nature
- Simillarities with Node.js but not a clone !
- High-Performing
- Slim and leightweight core
- Polyglot
- Inspired also from Erlang, RabbitMQ

CORE API

- TCP / SSL
- HTTP(s) and compression
- WebSockets / SockJS
- File System
- Event Bus
- Timers, Buffers
- Streaming and Pumps
- UDP
- DNS
- ...

INTERNALS

- Build on top of Netty 4
- Hazelcast for group management of cluster members
- Java7+
- Jackson for JSON

POLYGLOT

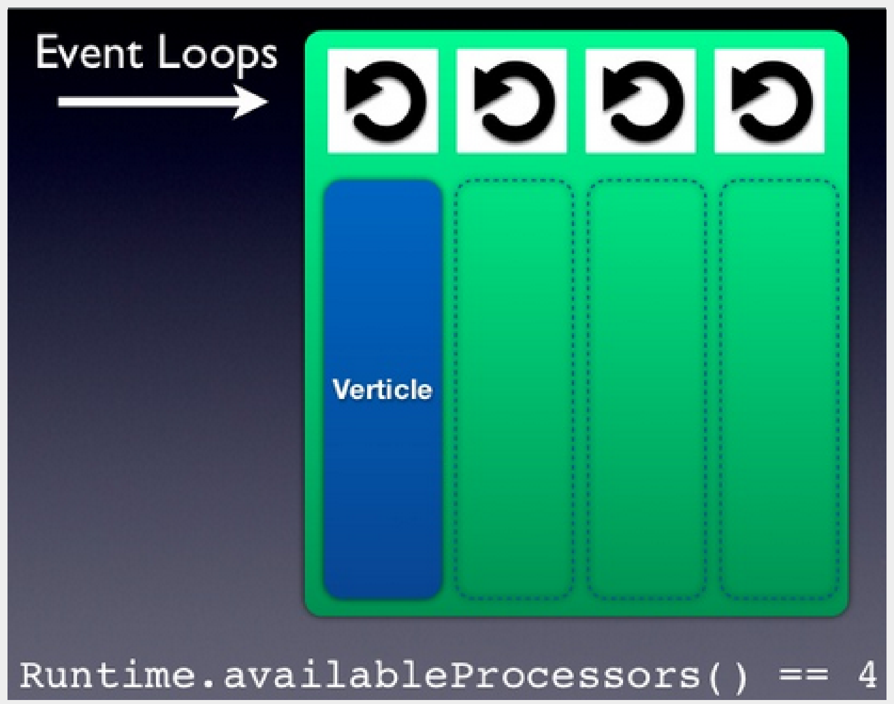
- Java
- JS (Rhino, Nashorn)
- Groovy
- Ruby
- Python
- Scala
- Clojure
- PHP

VERTICLE

- Execution unit of Vert.x
- Single threaded
- Verticles communicate via message passing
- Sounds like the Actor Model

DEMO

VERT.X INSTANCE



EVENT LOOPS

- EventLoops (Usually one per Core)
- Multi-reactor pattern
- No concurrency issues (synchronization, locking, ..)

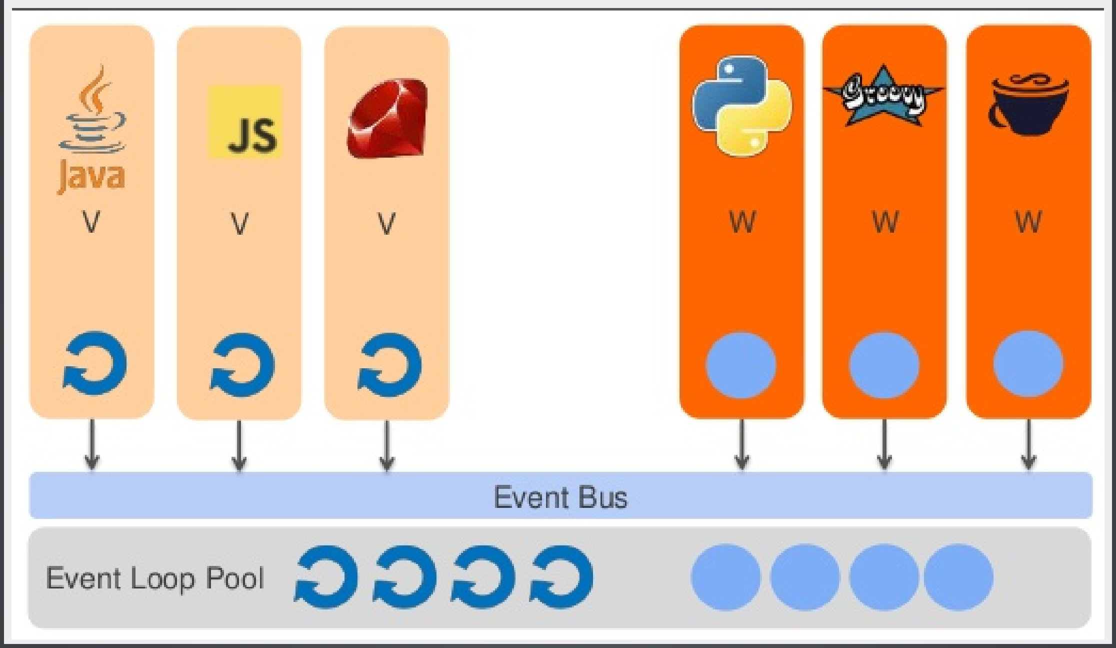
Never block the EventLoop!

EVENT BUS

- Pub/Sub/P2P
- Bridges - (AMQP, 0MQ, Kafka, ..)
- JSON, BSON, Byte[], ...

Nervous system

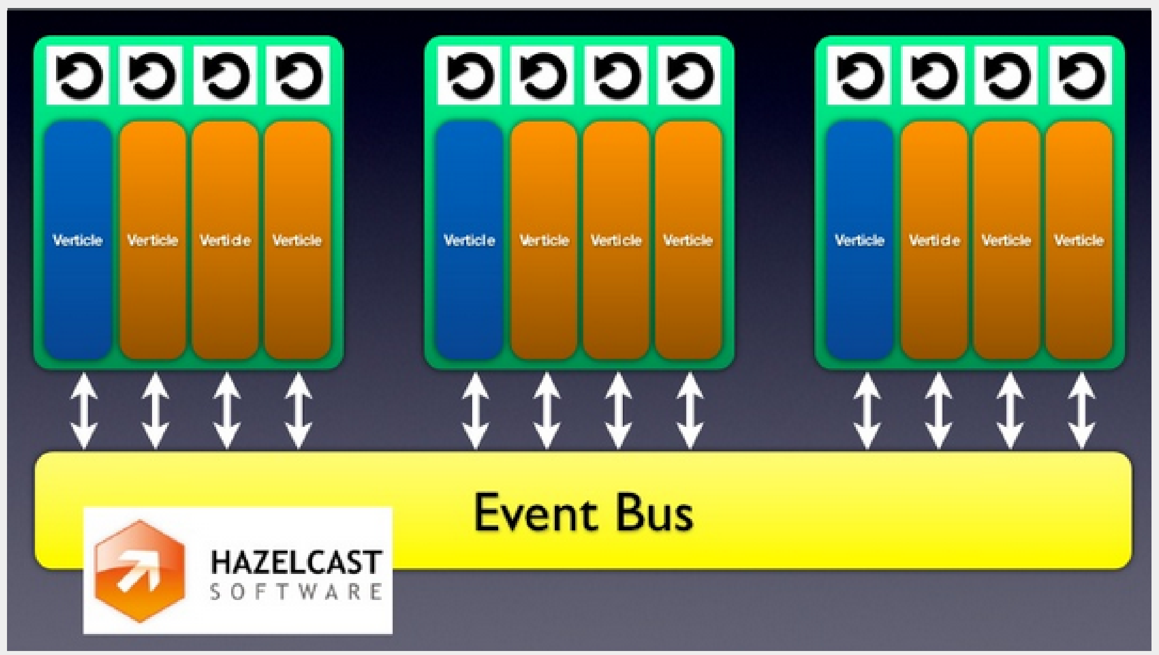
EVENT BUS



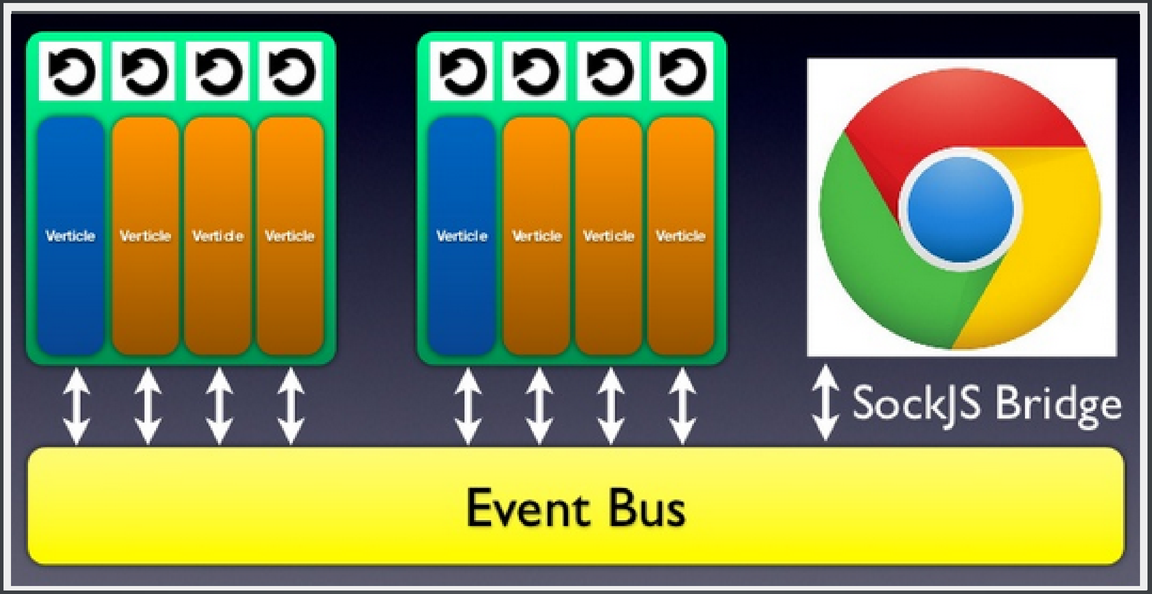
CLUSTERED EVENT BUS

- Connects multiple Vert.x instances(JVM)
- Hazelcast as cluster manager(pluggable)
- Cluster manager used for group management not transport !!

CLUSTERED EVENT BUS



EVENT BUS & BROWSER



DEMO

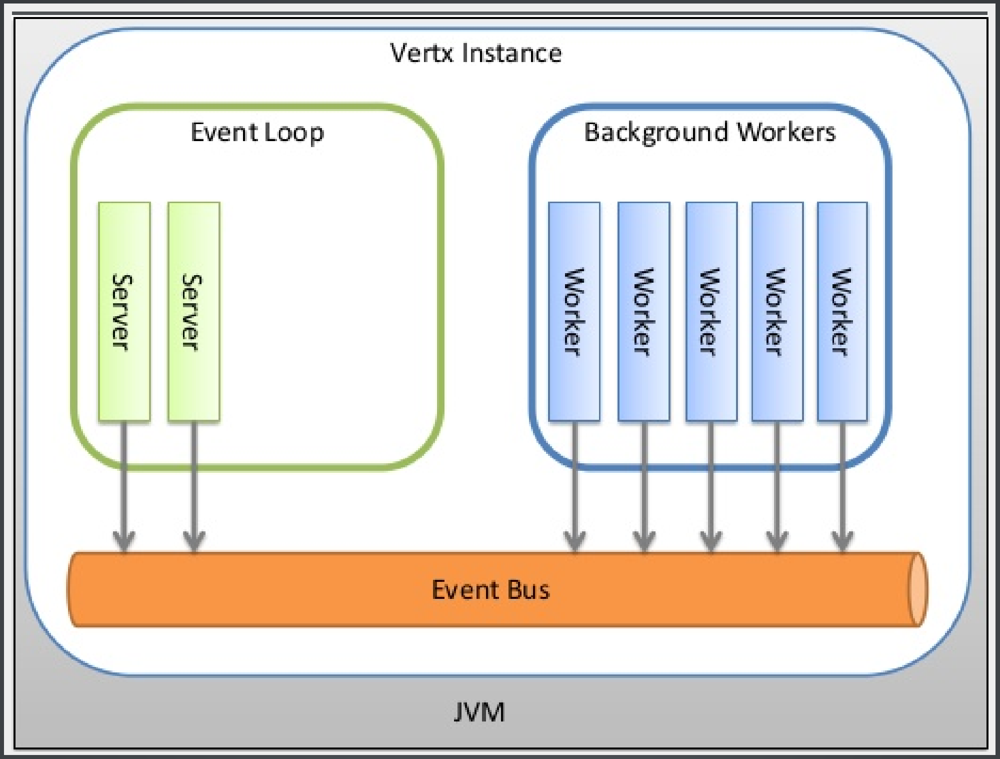
BLOCKING TASKS

THE BACKGROUND POOL

- Hybrid model
- Worker Verticles
- blocking calls (jdbc, ...)

vertx.pool.worker.size

WORKERS


Vertx Instance
Event Loop
Server
Server
Background Workers
Worker
Worker
Worker
Worker
Worker
Event Bus
JVM

EVENT DRIVEN

CALLBACKS HELL

```
().registerHandler("mymod.file.save", new Handler<Message<JsonObject>>() {

d handle(Message<JsonObject> msg

d("vertx.auth.authorise", msg.body().getObject("sessionID"), new Handler<Messag
blic void handle(Message<

  if (reply.body().getObj                             & reply.body().getObject("
      //query = ...reply.
      eb.send("vertx.mong                             er<Message<JsonObject>>()
          public void han                             lt) {

              if (reply.b                             uals("ok") && reply.body()

                  vertx.f                             dat", new AsyncResultHand
                      pub                             AsyncFile> ar) {

                                                      r.result();
                                                      uffer 5 times into a file
                                                      r("foo");
                              for (int i = 0; i < 5; i++) {
                                  asyncFile.write(buff, buff.length() * i, new
                                      public void handle(AsyncResult ar) {
                                          if (ar.succeeded()) {
                                              log.info("Written ok!");
                                              // etc
                                          } else {
                                              log.error("Failed to write", ar.c
                                          }
```

CALLBACKS HELP

- Rx (Reactive Extensions)
- https://github.com/vert-x/mod-rxvertx

MODULES

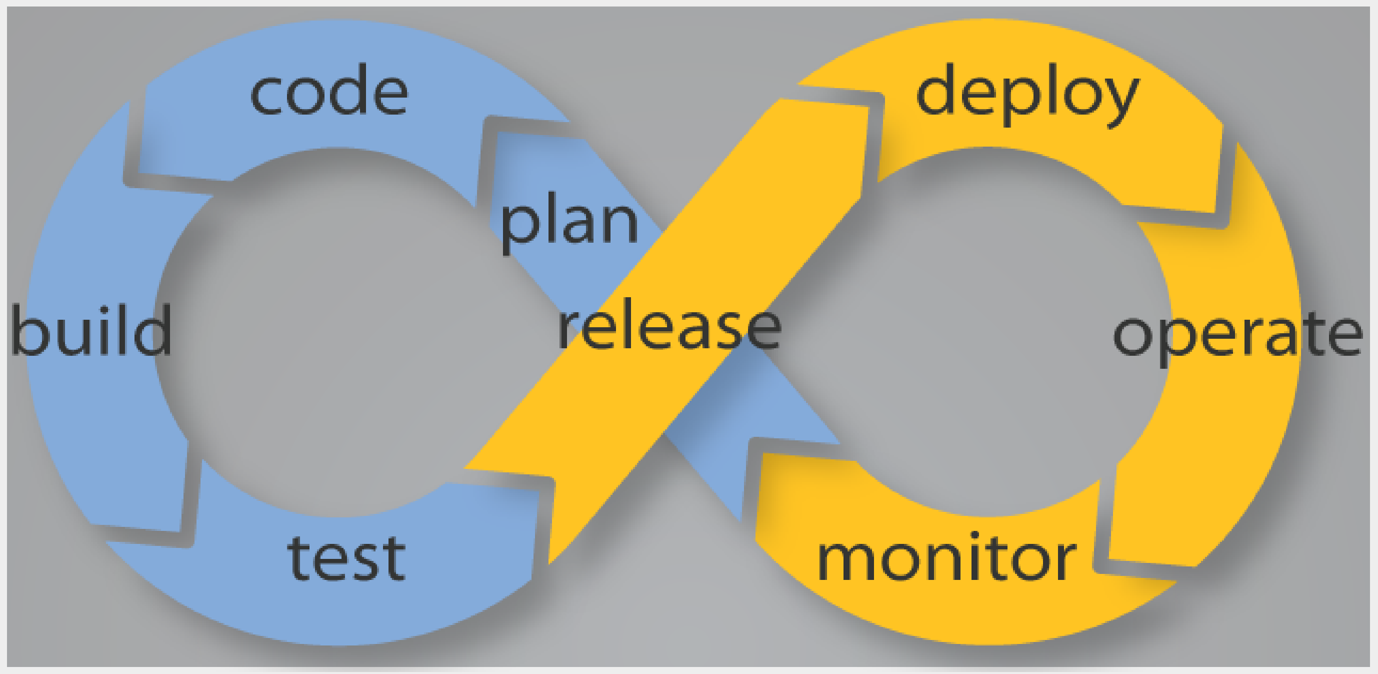
- Even new language support is done via modules!
- No changes to core needed
- Runnable modules vs. includable modules
- Repositories (Maven, Bintray, ..)

HIGH AVAILABILITY

- Automatic failover
- HA groups
- Network partitions - Quora

CLUSTERS

DEMO


code
plan
build
release
test
deploy
operate
monitor

DEVELOPING WITH VERT.X

- IDEs..
- Text editor
- Maven, Gradle, ..
- Auto-redeploy

DEPLOYING VERT.X

- Fat Jar, ZIP, verticles
- Modules, versions, configs
- Vert.x and Java version ?

PROJECT INFO

- Tim Fox - father of Vert.x
- VMware - incubator
- Eclipse Foundation
- Employed by Red Hat

SECURITY

ENTERPRISE READY ?

3.0 +

- Remove Vert.x platform
- Event bus encryption
- Code generation tool
- Simplify cluster configuration
- Wildcards on eventbus
- Server management - GUI
- ...

coming soon

SUMMARY

- Horizontal & vertical scaling
- High availability
- Polyglot
- Easily extendable
- Growing ecosystem

QUESTIONS ?

THANKS

http://spkr8.com/t/42891

IMAGE CREDITS

- jbandi - http://www.slideshare.net/jbandi/vertx-asynchronous-eventdriven-web-applications-on-the-jvm
- Matt Stine - http://www.slideshare.net/mstine/vertx-14673790
- Sascha Möllering - http://www.slideshare.net/saschamoellering/vertx-in-productionfinal