FRAMEWORK

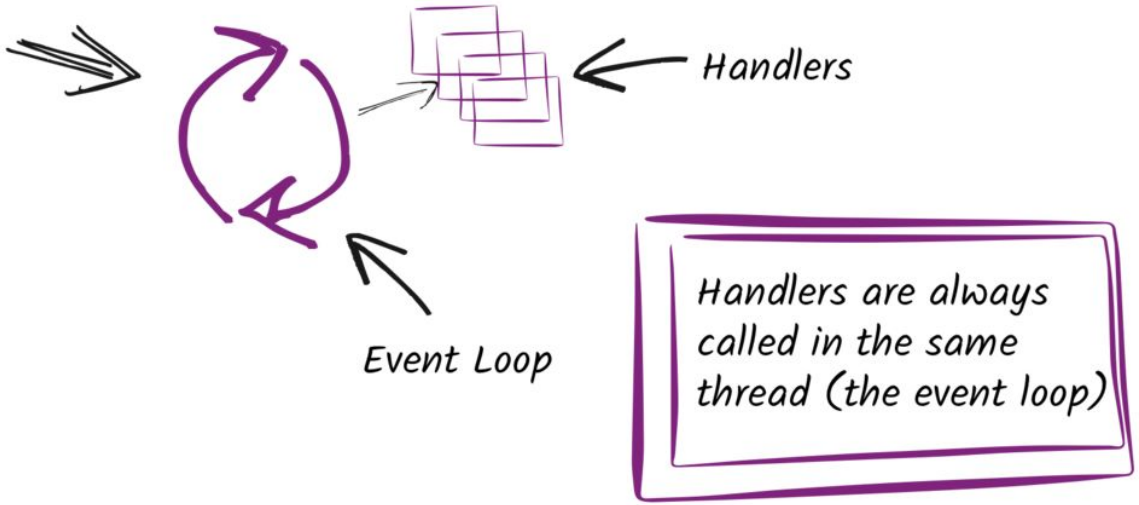
UNOPINIATED

REACTIVE

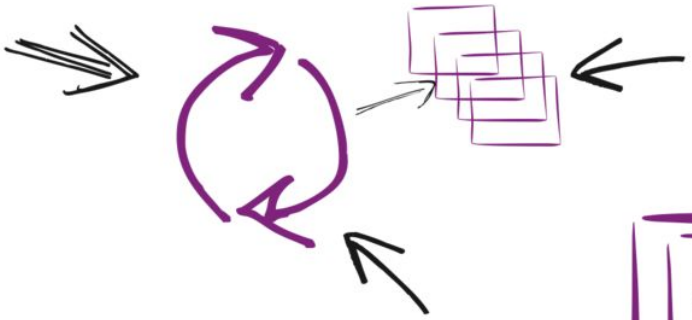
POLYGLOT

DISTRIBUTED

REACTOR-PATTERN

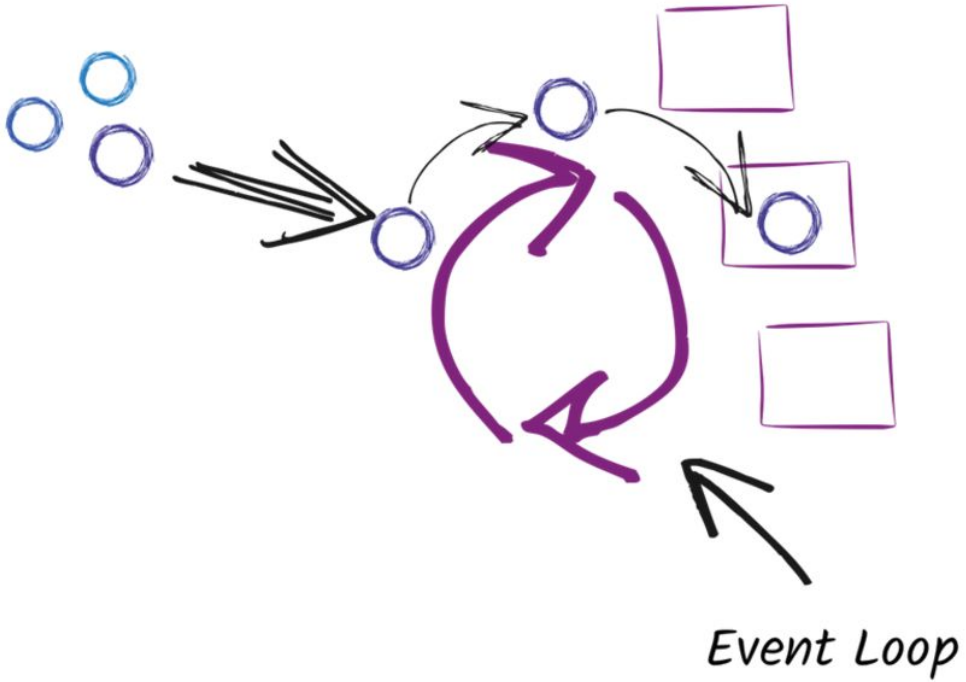


REACTOR-PATTERN

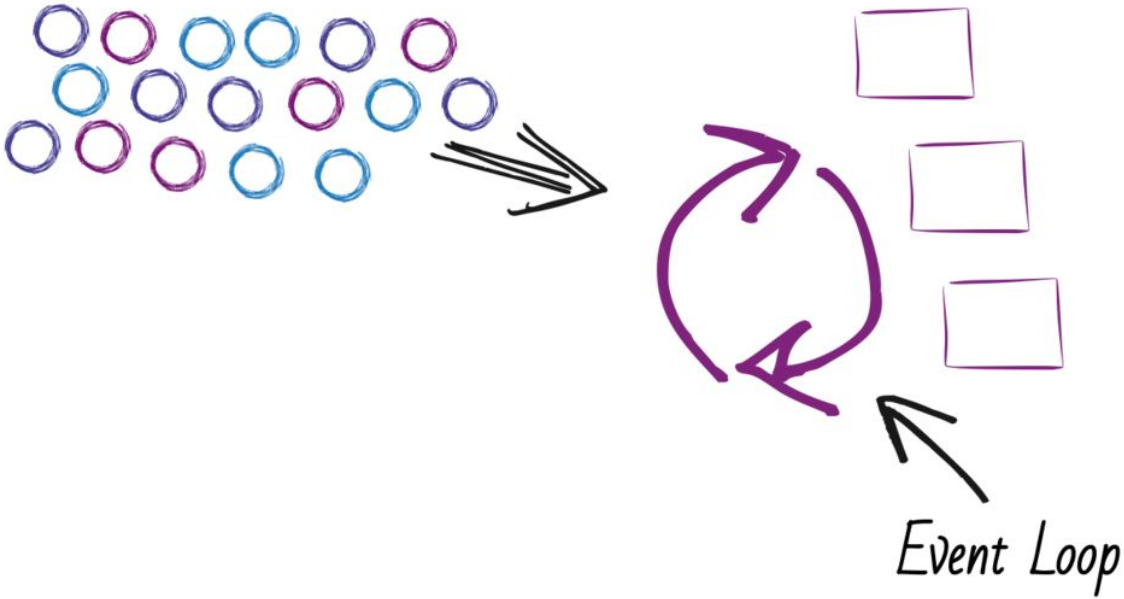


Handlers must never block the thread calling them.

REACTOR-PATTERN



REACTOR-PATTERN



WHAT IS  ?

A MESSAGE

A NOTIFICATION

A HTTP REQUEST

A COMMAND

A FILE

A RESULT, AN ERROR

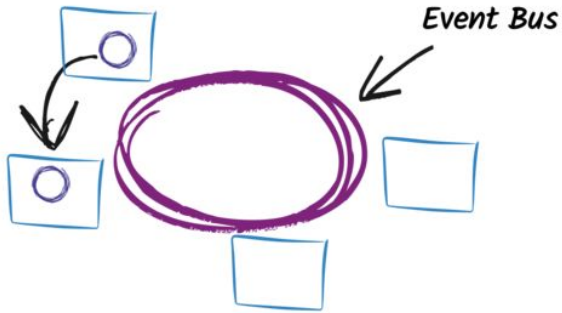
ASYNCHRONOUS DEVELOPMENT MODEL

```
void operation(param1, param2, Handler<T> handler) {
  // ...
  T t = ...
  handler.handle(t)
  // ...
}
```

```
void handle(T t) {
  // do something with t
}
```

```
operation(1, 2, t -> System.out.println(t));
```

THE EVENT BUS

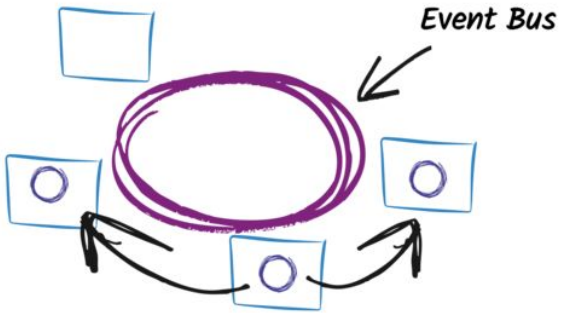


Point to Point

Publish / Subscribe

Request / Response

THE EVENT BUS

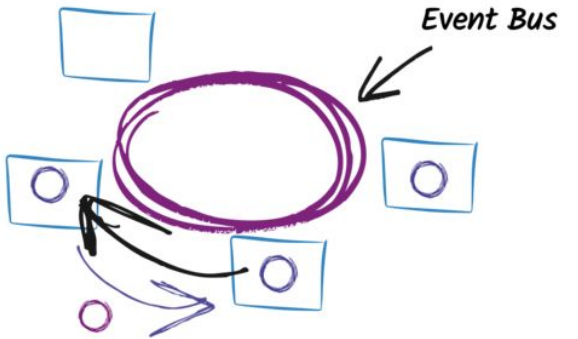


Point to Point

Publish / Subscribe

Request / Response

THE EVENT BUS

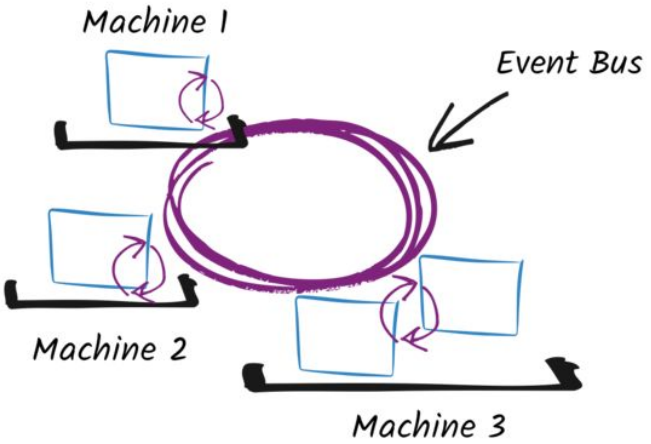


Point to Point

Publish / Subscribe

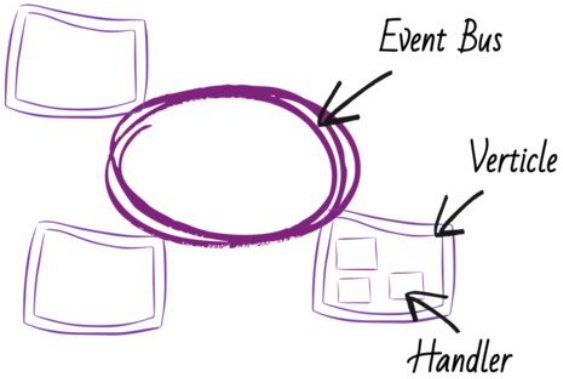
Request / Response

THE EVENT BUS



The event bus allows distributed communication.

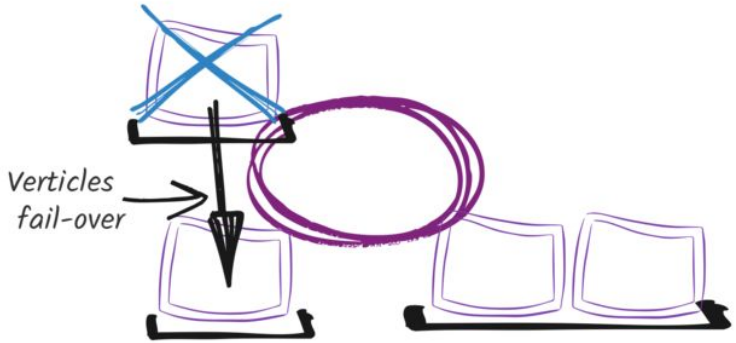
VERTICLES
AN AGENT-LIKE MODEL



Verticles are chunks of code that get deployed and run by Vert.x.

```
vertx.deployVerticle("my.verticle");
```

VERTICLES
AN AGENT-LIKE MODEL



Dead verticles are restarted on a running node.

```
vertx.deployVerticle("my.verticle"),
                new DeploymentOptions().setHA(true));
```