

COMEDY

акторы для гибкого
масштабирования 

ВИКТОР ИСАЕВ

Team Lead проекта SAYMON

14-летний опыт разработки приложений на Java, C++ и JavaScript.

NodeJS занимается последние 3,5 года в рамках разработки бэкэнда платформы SAYMON.



NodeJS однопоточный



Варианты масштабирования

> **CLUSTER**

> **PM2**

> **COMEDY**



CLUSTER

```
var cluster = require('cluster');
var http = require('http');
var numCPUs = 4;

if (cluster.isMaster) {
    for (var i = 0; i < numCPUs; i++) {
        cluster.fork();
    }
} else {
    http.createServer(function(req, res) {
        res.writeHead(200);
        res.end('process ' + process.pid);
    }).listen(8000);
}
```



CLUSTER

```
var cluster = require('cluster');
var http = require('http');
var numCPUs = 4;

if (cluster.isMaster) {
    for (var i = 0; i < numCPUs; i++) {
        cluster.fork();
    }
} else {
    http.createServer(function(req, res) {
        res.writeHead(200);
        res.end('process ' + process.pid);
    }).listen(8000);
}
```



PM2

```
[joni] ~/keymetrics/PM2 $ pm2 scale app +3
```

```
[PM2] Scaling up application
```

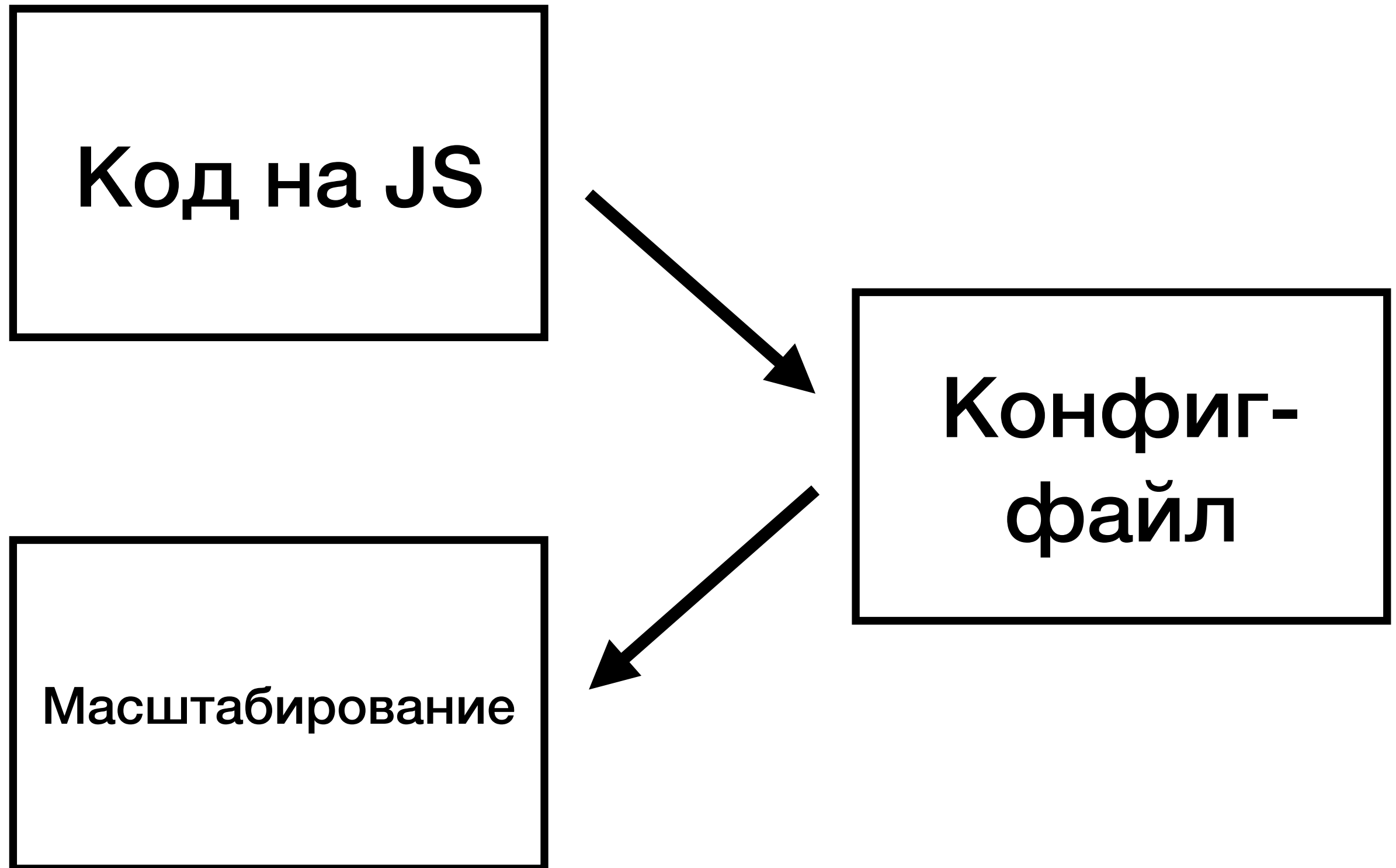
```
[PM2] Scaling up application
```

```
[PM2] Scaling up application
```

App name	id	mode	pid	status	restart	uptime	memory
app	0	cluster	16660	online	0	12s	18.379 MB
app	1	cluster	16669	online	0	12s	20.359 MB
app	2	cluster	16692	online	0	12s	18.488 MB
app	3	cluster	16715	online	0	12s	18.383 MB
app	4	cluster	16779	online	0	0s	20.125 MB
app	5	cluster	16786	online	0	0s	18.082 MB
app	6	cluster	16809	online	0	0s	20.289 MB

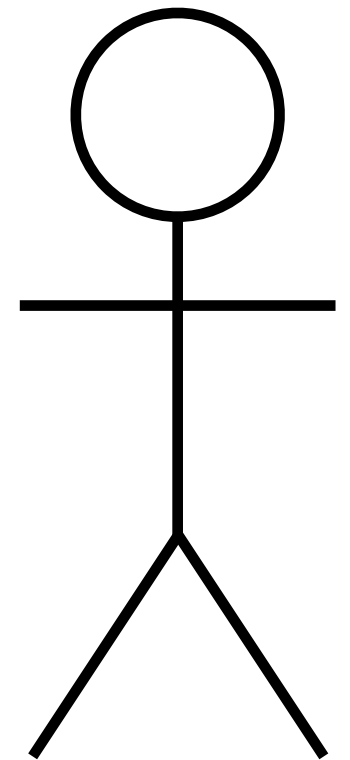


COMEDY



ЧТО ТАКОЕ АКТОР?

- > принимает сообщения
- > отправляет сообщения
- > порождает дочерние акторы



COMEDY

```
var actors = require('comedy');

// Actor definition class.
class MyActor {
  sayHello(to) {
    console.log(`Hello, ${to}!`);
  }
}

actors()
  // Get a root actor reference.
  .rootActor()
  // Create a class-defined child actor.
  .then(rootActor => rootActor.createChild(MyActor))
  .then(myActor => {
    // Our actor is ready, we can send messages to it.
    myActor.send('sayHello', 'world');
  });
```



COMEDY

```
var actors = require('comedy');

// Actor definition class.
class MyActor {
  sayHello(to) {
    console.log(`Hello, ${to}!`);
  }
}

actors()
  // Get a root actor reference.
  .rootActor()
  // Create a class-defined child actor.
  .then(rootActor => rootActor.createChild(MyActor))
  .then(myActor => {
    // Our actor is ready, we can send messages to it.
    myActor.send('sayHello', 'world');
  });
```



COMEDY

```
var actors = require('comedy');

// Actor definition class.
class MyActor {
  sayHello(to) {
    console.log(`Hello, ${to}!`);
  }
}

actors()
  // Get a root actor reference.
  .rootActor()
  // Create a class-defined child actor.
  .then(rootActor => rootActor.createChild(MyActor))
  .then(myActor => {
    // Our actor is ready, we can send messages to it.
    myActor.send('sayHello', 'world');
  });
```



COMEDY

```
var actors = require('comedy');

// Actor definition class.
class MyActor {
  sayHello(to) {
    console.log(`Hello, ${to}!`);
  }
}

actors()
  // Get a root actor reference.
  .rootActor()
  // Create a class-defined child actor.
  .then(rootActor => rootActor.createChild(MyActor))
  .then(myActor => {
    // Our actor is ready, we can send messages to it.
    myActor.send('sayHello', 'world');
  });
```



COMEDY

```
{  
  "MyActor": {  
    "mode": "forked",  
    "clusterSize": 3  
  }  
}
```



COMEDY

Результат

```
$ ps ax | grep node
11031 ?          S1      0:00 node /tmp/simple-class.js
11041 ?          S1      0:00 node /tmp/forked-actor-worker.js MyActor
11046 ?          S1      0:00 node /tmp/forked-actor-worker.js MyActor
11048 ?          S1      0:00 node /tmp/forked-actor-worker.js MyActor
```

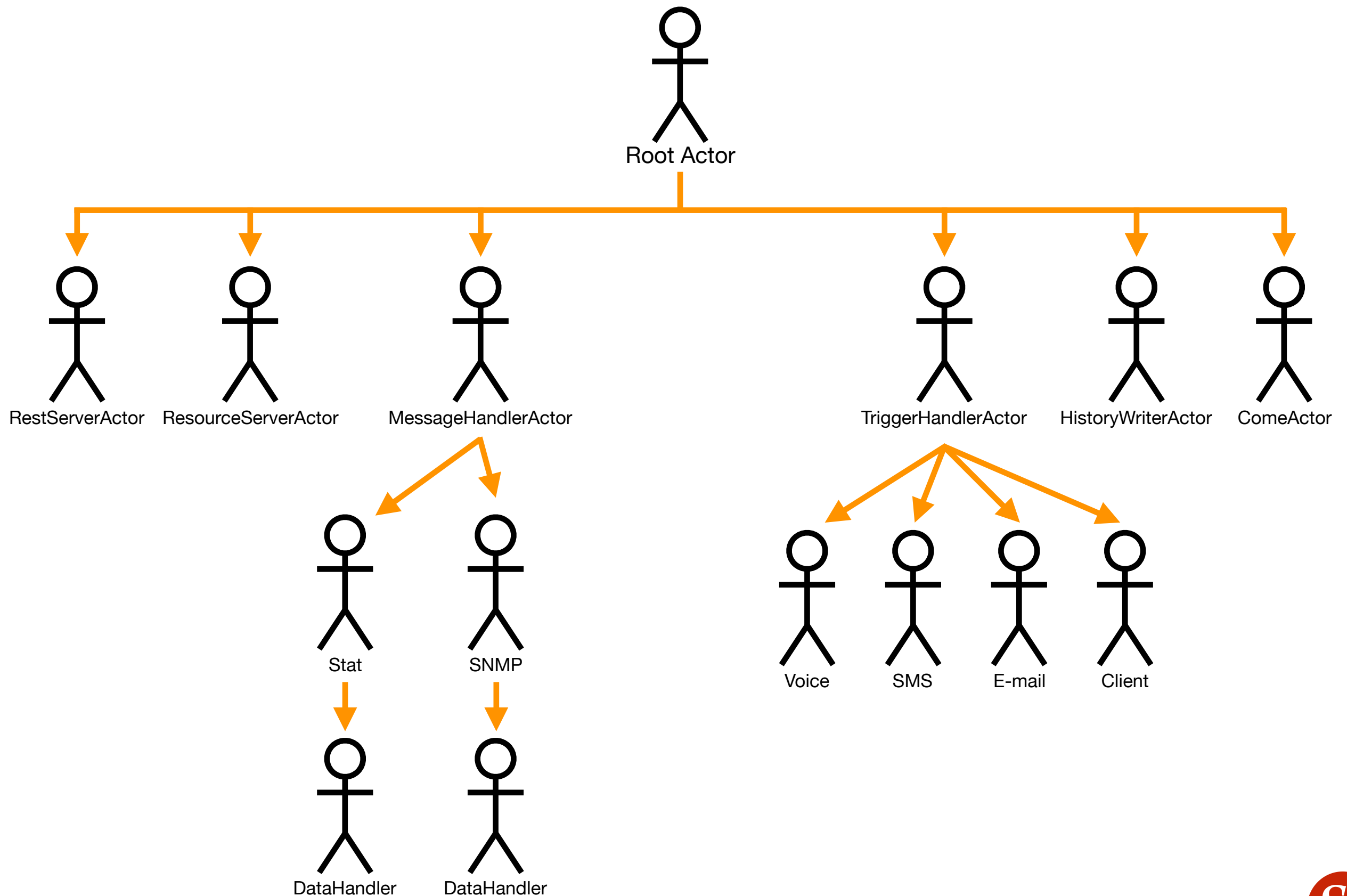


МАСШТАБИРОВАНИЕ ЧЕРЕЗ АКТОРЫ

- > Описать приложение в терминах акторов (иерархия)
- > Настроить акторы через конфиг-файл



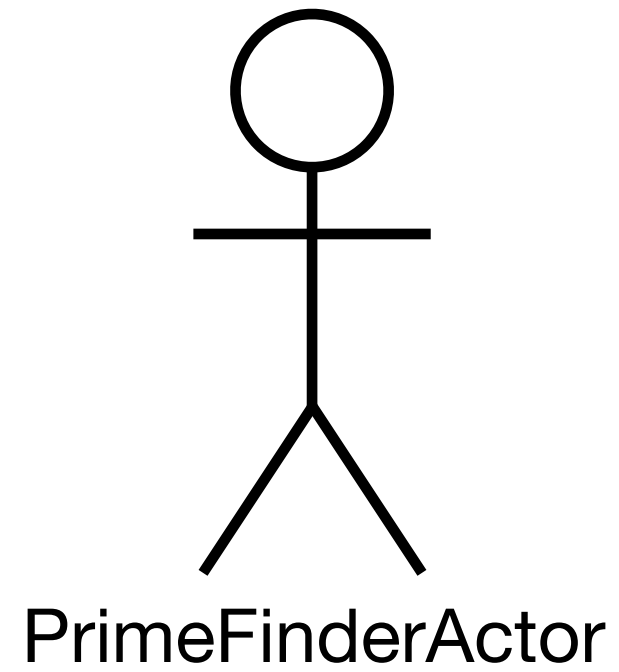
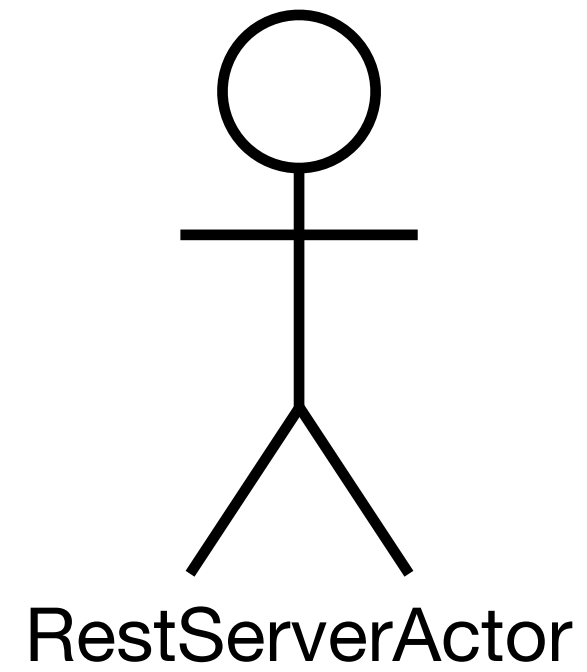
ПРИМЕР ИЕРАРХИИ АКТОРОВ

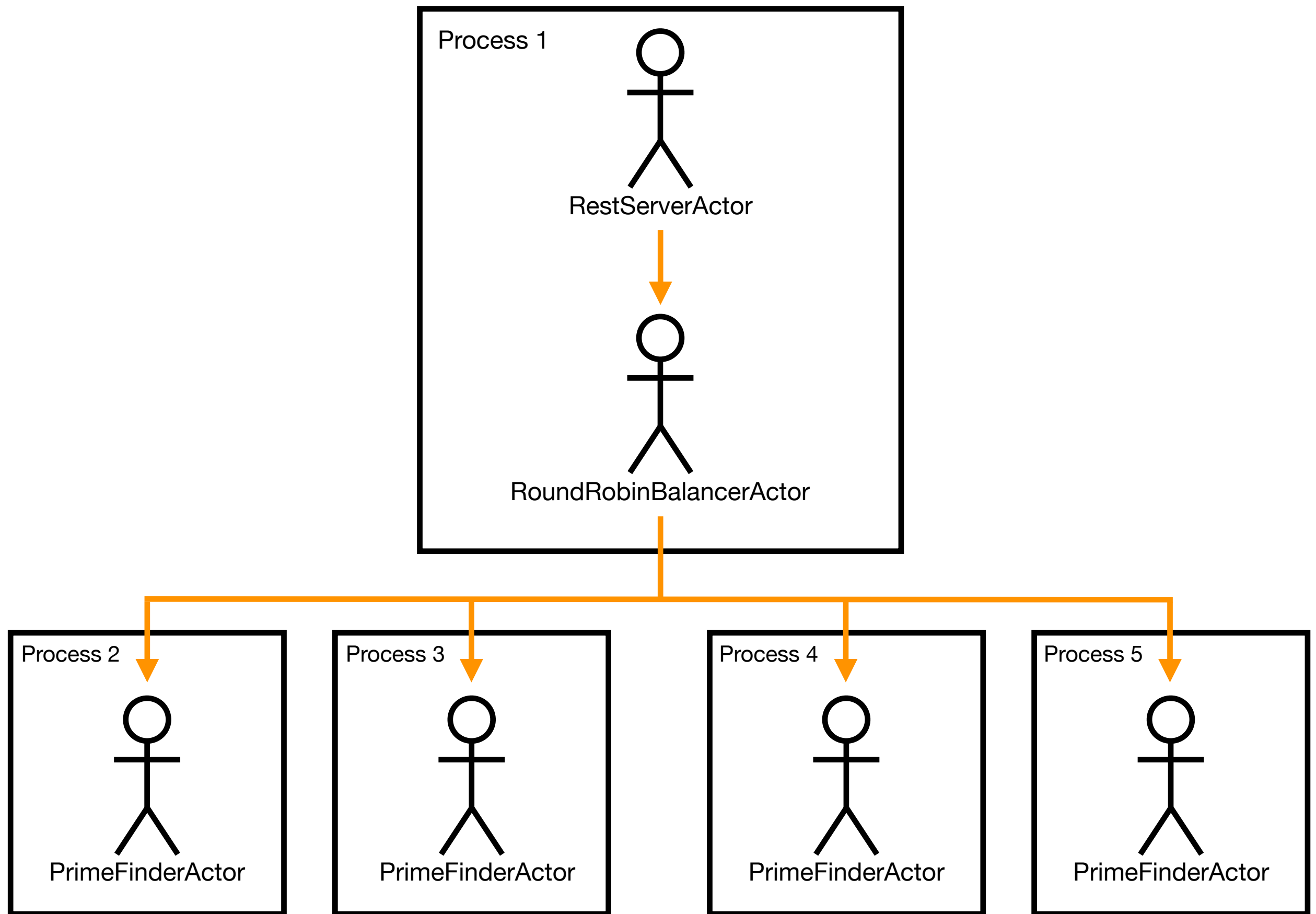


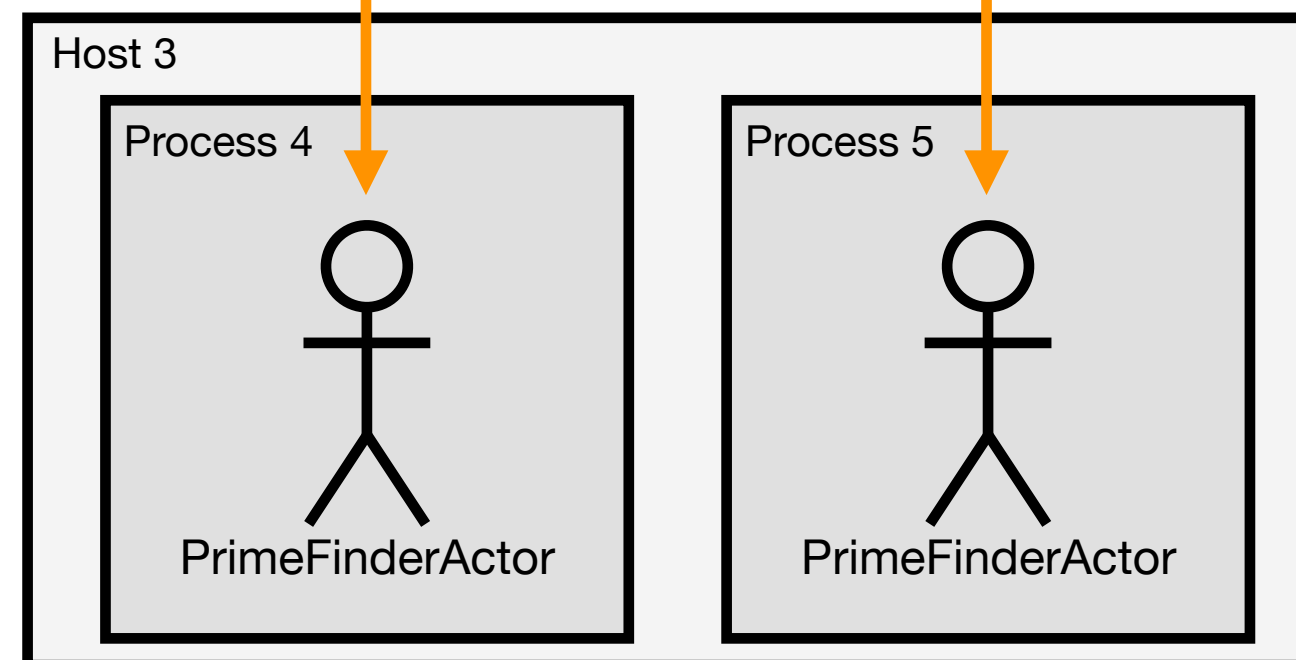
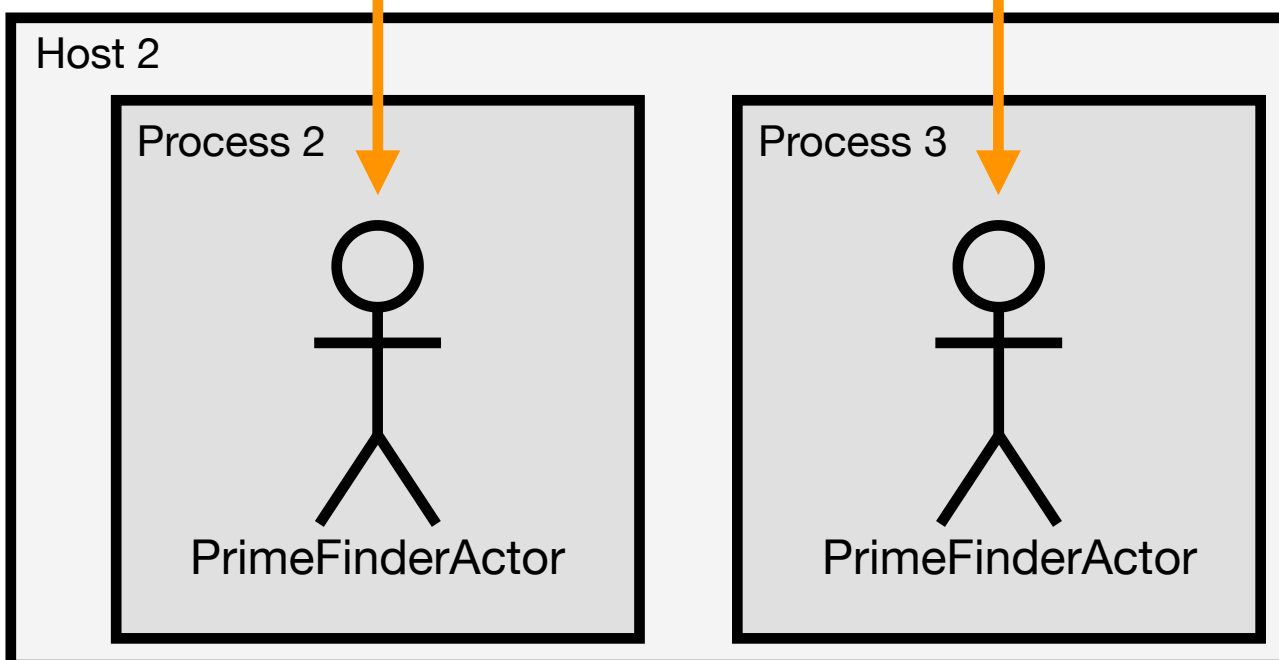
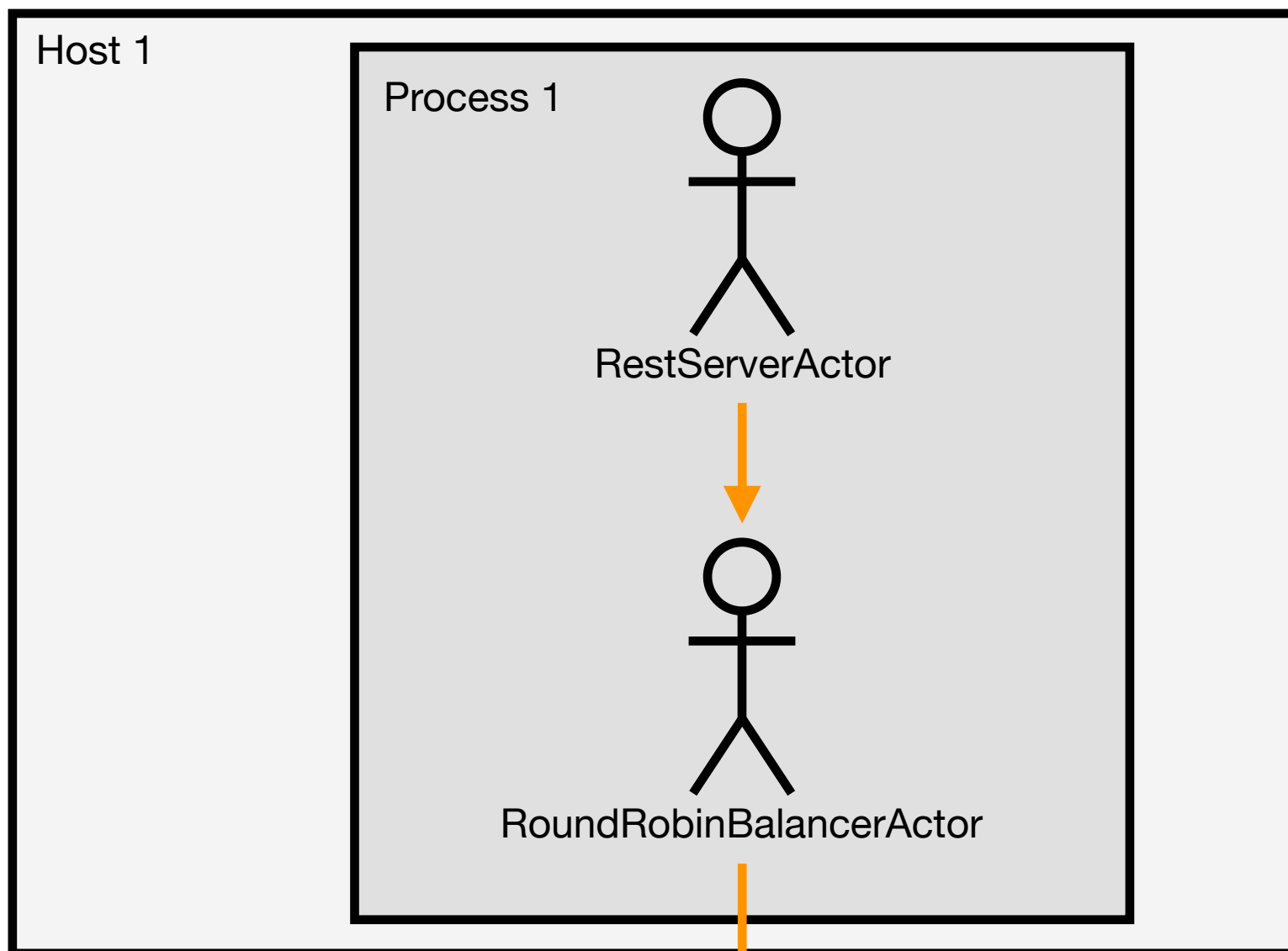
DEMO

github.com/untu/comedy-presentation









ВИКТОР ИСАЕВ

weekens@gmail.com

github.com/weekens

