

11<sup>th</sup> Central and Eastern European Software Engineering  
Conference in Russia - CEE-SECR 2015

October 22 - 24, Moscow



# Запрягай носорога

## Java Scripting API

Александр Хрущев



WINFOX INTERACTIVE

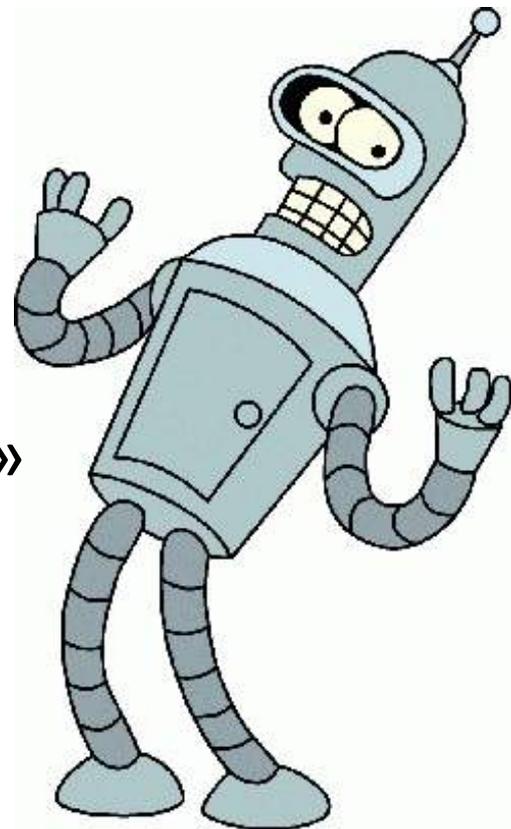
# Дано:

- Робот для online-игры
  - REST-сервис с JSON
  - Авторизация OAuth
  - Работа с массивами данных
- Проблемы:
  - Нет времени постигать игровой процесс
  - Игра постоянно меняется



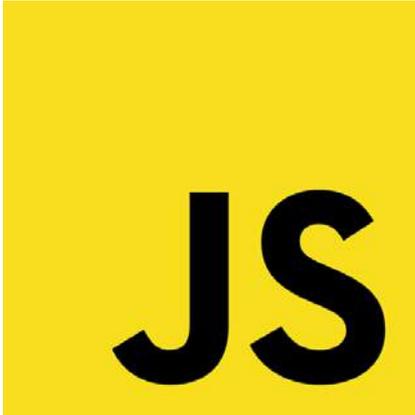
# ХОТИМ

- Много аккаунтов
- Интерактивность
- Nivemind
- Менять логику «на лету»
- Менять обращения к серверу «на лету»



# Скриптовый движок

- Производительность
- Быстрый старт
- Программист не нужен для разработки логики

The logo for JavaScript, consisting of the letters 'JS' in a bold, black, sans-serif font, centered on a bright yellow square background.

**JS**

# Минутка бюрократии

- CommonJS
- ECMA-262
- JSR-292



# CommonJS

- Modules – модульная структура, `require()`
- Packages – поддержка пакетов
- Package mappings – поддержка зависимостей
- JSJI – middleware для Web-сервера
- Promises – асинхронная цепочка вызовов (`when/then`)

# ESMA-262

- Стандарт скриптового языка ECMAScript. Цели стандарта
  - Не сломать web
  - Улучшить удобство использования языка
  - Улучшить защищенность
  - Как можно меньше нового синтаксиса
  - *Защитить разработчиков от самих себя*

# JSR-292

- Supporting Dynamically Typed Languages on the Java™ Platform
- Реализация: Da Vinci Machine
- Цель:
  - Не использовать Permanent Generation
  - Предоставить invokedynamic

# Rhino

- Open-source движок для JavaScript
- Поддерживается Mozilla Foundation
- Соответствует ECMA-262 (v 3)
- MPL/GPL лицензия
- LiveConnect
- Поддерживает разные режимы выполнения
- Предоставляет отладчик



# LiveConnect

---

```
// Import the stuff we'll need.
importClass(Packages.javax.swing.JFrame);
importClass(Packages.javax.swing.JButton);
importClass(java.awt.event.ActionListener);

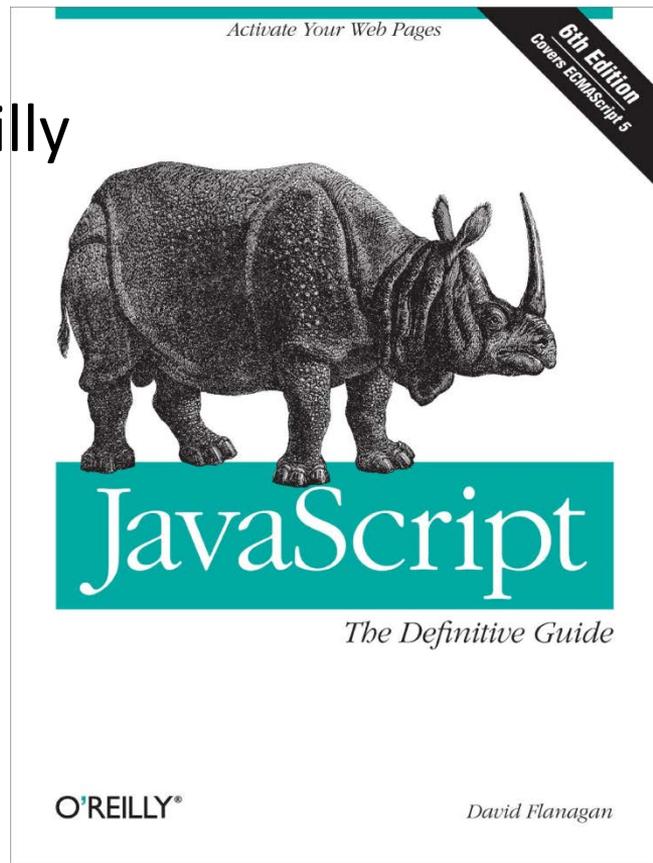
// Create some Java objects.
var frame = new JFrame("Hello World");
var button = new JButton("Hello World");

// Implement the ActionListener interface.
var listener = new ActionListener({
    actionPerformed: function(e) { print("Hello!"); }
});

var p = new java.awt.Polygon();
p.xpoints = java.lang.reflect.Array.newInstance(java.lang.Integer.TYPE,5);
p.ypoints = java.lang.reflect.Array.newInstance(java.lang.Integer.TYPE,5);
for(var i = 0; i < p.xpoints.length; i++) {
    p.xpoints[i] = i;
    p.ypoints[i] = i * i;
}
```

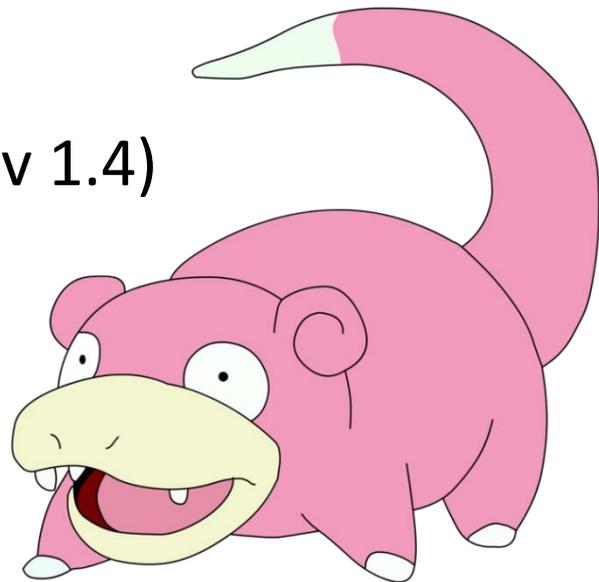
# Детство.

- Вырос из Netscape Javigator
- Назван в честь зверя на книжке O'Reilly
- Ahead-of-Time Compilation (AOT)
- Детские болезни
  - AOT - долго
  - GC за носорогом не убирает
  - Нет отладки



# Отрочество.

- Добавили интерпретируемый режим (v 1.4)
  - + нет проблем с GC
  - - тормозит
- Оставили режим компиляции
  - + не тормозит
  - - проблема 64k
- Добавили Rhino JavaScript Debugger (v 1.5)
  - + это удобно



# ~~Юность.~~ Старость

- Последняя версия 1.7 R5 (2015 год!) для Java 7
- Rhino JavaScript Debugger сильно прокачали
- Для Java 8 возможен backport Rhino

# JavaScript в Rhino

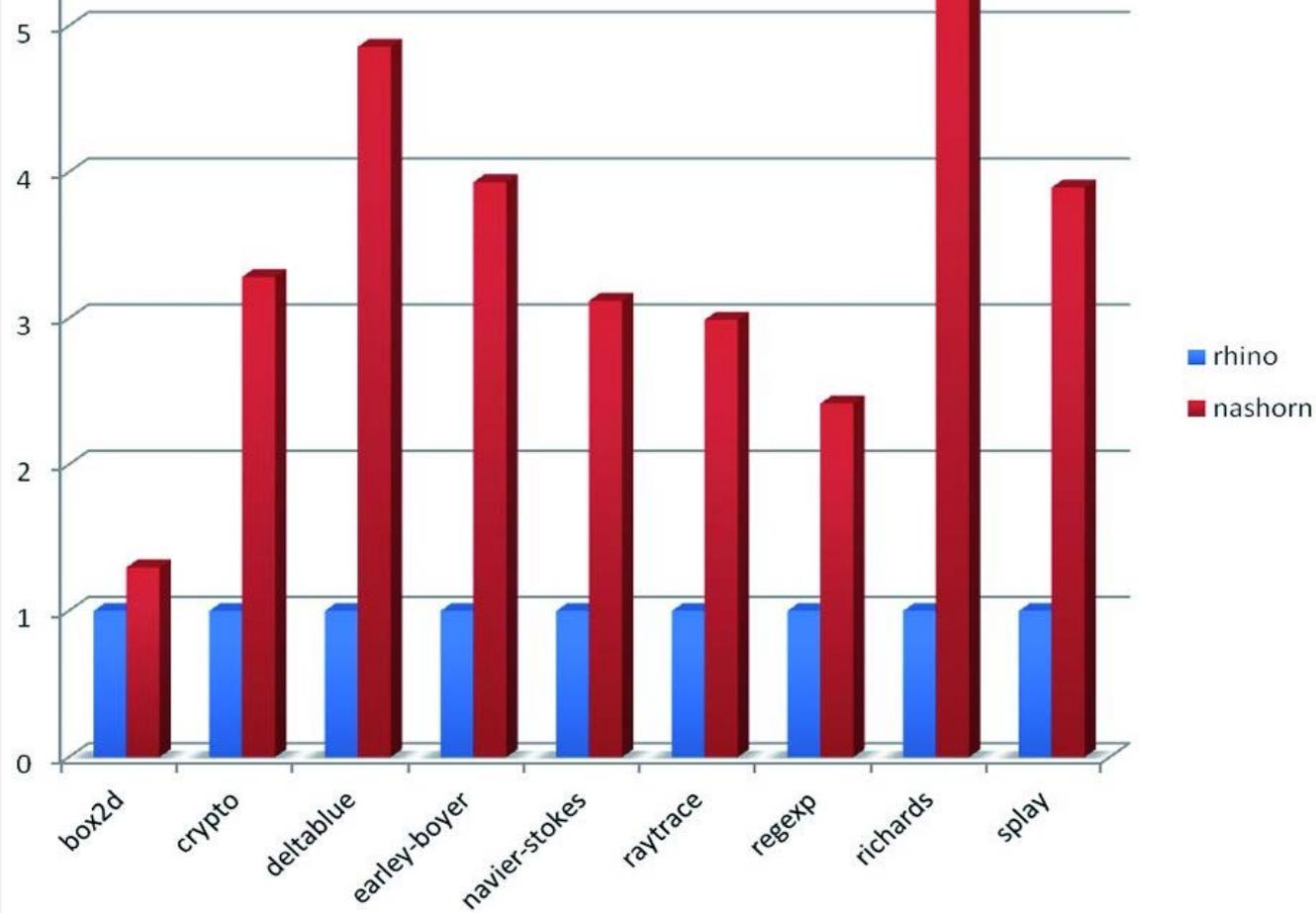
- Медленный интерпретируемый режим  
или
- generated bytecode for method exceeds 64K limit
- Баги реализации от Sun

# Nashorn

- Open-source движок для JavaScript
- Разработан Oracle
- Соответствует ECMA-262 5.1
- Распространяется по GPL лицензии
- Поставляется с Java 8 (и 9)
- Основан на Da Vinci Machine
- Just-in-Time (JIT) Compilation
- Отладчик есть в Netbeans 8



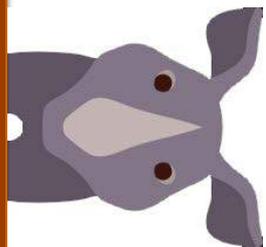
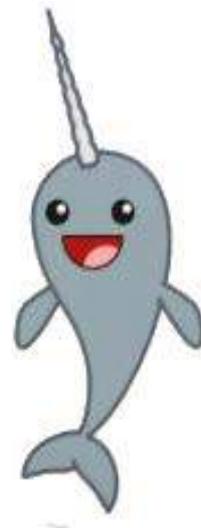
# Nashorn VS Rhino



# Где работают носороги

- Narwhal

- Реализация CommonJS на Rhino
- Свой менеджер пакетов Tusk



- Ringo (многопоточный CommonJS на Rhino)

- Scripturian (CommonJS + куча всего)

- Поддержка многих скриптовых языков и языков шаблонов Velocity и Succinct

# Где работают носороги

- Avatar.JS
  - Node.JS для Java
  - Совместим со многими Node.JS пакетами

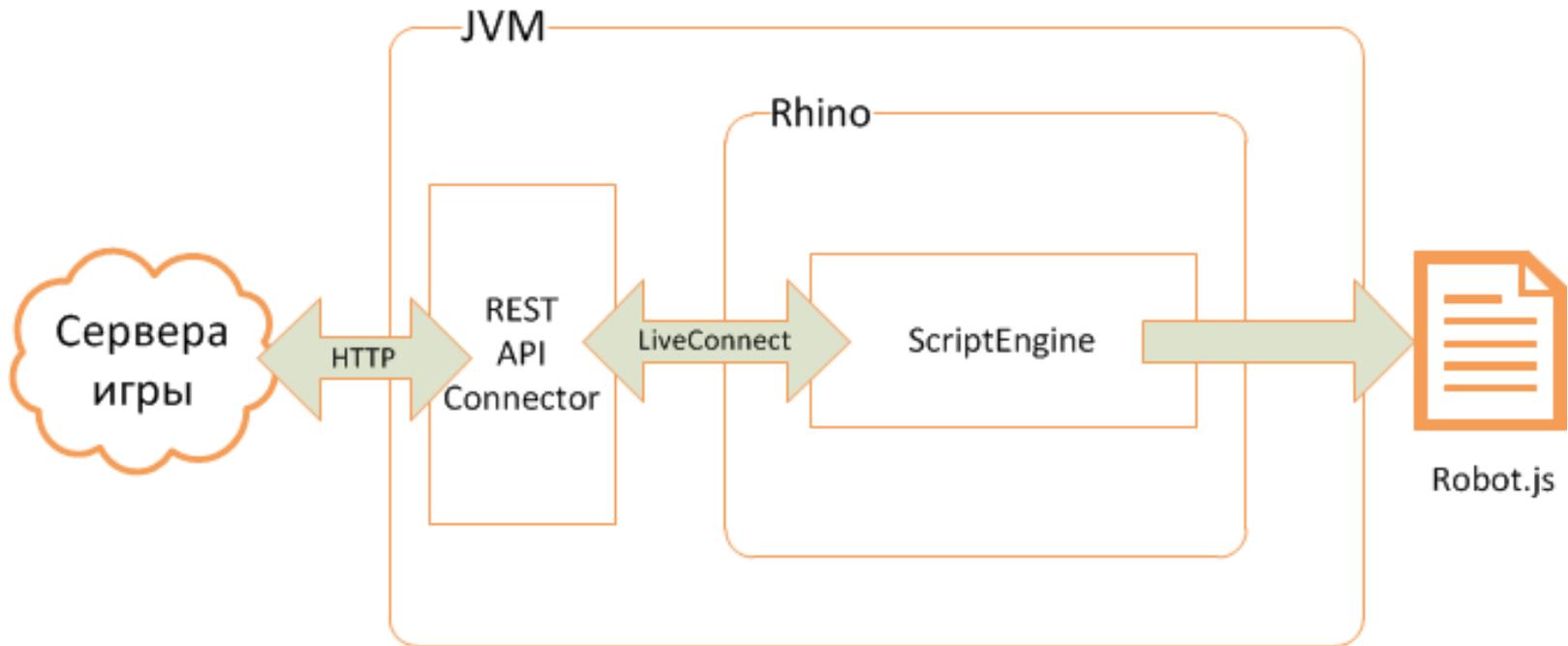
Avatar.js

# Мобильные платформы?

- Android
  - Rhino работает в интерпретируемом режиме
  - Nashorn не работает, т.к. не реализован JSR 292
- iOS
  - Не работает, но... с iOS 7 есть JavaScriptCore

# 2014. Meanwhile in Russia...

- Первый прототип робота ScriptFighter
- Эксперимент удался



# Нам не нужно

- CommonJS

```
1  var assert = require("assert");
2
3  var foxtrot = function(f){
4    return ("The earth is " + f);
5  }
6
7
8  describe("foxtrot()", function(){
9    it("should return 'The earth is round', if passed 'round'", function(){
10     var retval = foxtrot("flat");
11
12     assert.equal(retval, "The earth is round");
13   });
14 });
```

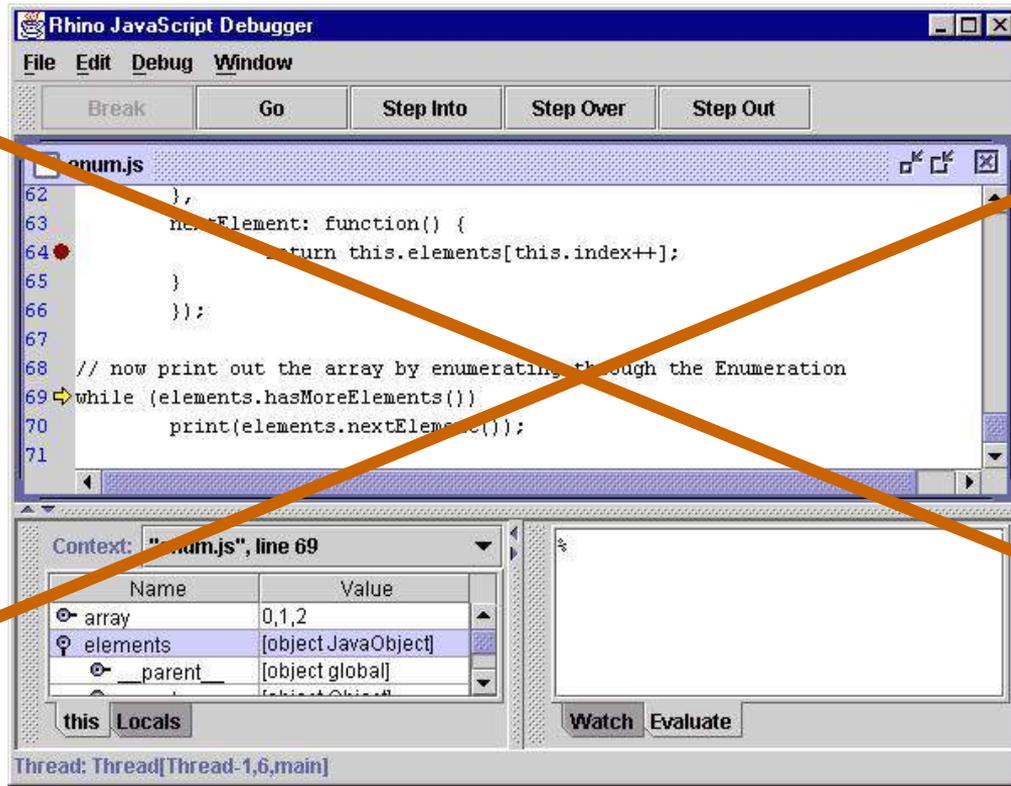
# Нам не нужно

- LiveConnect

```
1  importPackage(org.eclipse.jetty.server);
2  importPackage(org.eclipse.jetty.server.handler);
3  importPackage(org.eclipse.jetty.servlet);
4  importPackage(javax.servlet.http);
5
6  var contextHandler = new ServletContextHandler(ServletContextHandler.SESSIONS);
7  contextHandler.setContextPath('/');
8  java.lang.System.exit(0);
9  contextHandler.addServlet(
10     new ServletHolder(new HttpServlet({
11         doGet: function(request, response) {
12             response.setContentType('text/plain');
13             response.getWriter().println('Yeah!');
14         }
15     })),
16     '/test'
17 );
```

# Нам не нужно

- Rhino JavaScript Debugger



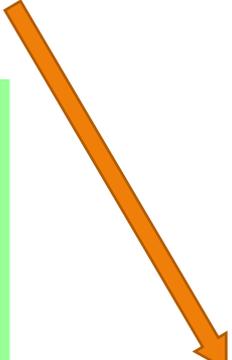
# Нам нужно

- Многопоточность
- IDE
- События
- СУБД
- Отладка
- Модули (пакеты)

# Цели

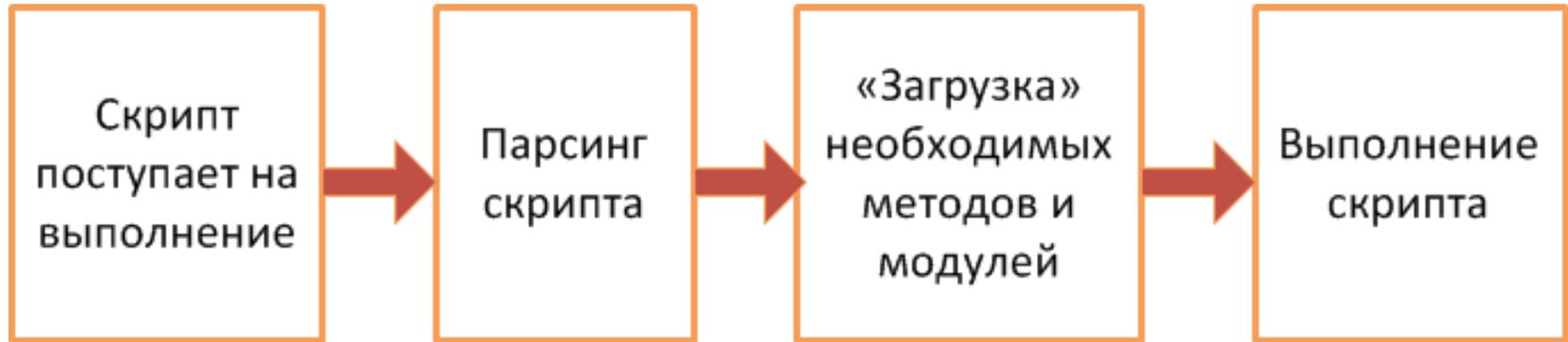
- Простой код
- Спрятать методы Java
- Не подключать модули в коде
- Встроенная пополняемая библиотека методов
- Обмен данными между потоками
- Хранение и обработка массивов данных
- Отладка и профилирование скриптов
- Расширяемая модульная структура

# «Глушим» LiveConnect



```
125
126     Context context = Context.enter();
127
128     context.setClassShutter(new ClassShutter() {
129
130     @Override
131     public boolean visibleToScripts(String fullClassName) {
132         for(String allowedClass: allowedClasses) {
133             if(fullClassName.equalsIgnoreCase(allowedClass)) {
134                 return true;
135             }
136         }
137         return false;
138     }
139     });
140
```

# Избавляемся от require()



# Классы модулей

- BasicModule (abstract)
  - registerMethods
  - registerEvents
- Connector – подключение REST, SOAP, SMPP, и т.д.
- Listener – проху, входящий REST, SOAP
- DatabaseConnector – подключение к DB
- Library – набор методов

# Структура модуля

Обертки доступных методов на Java Script

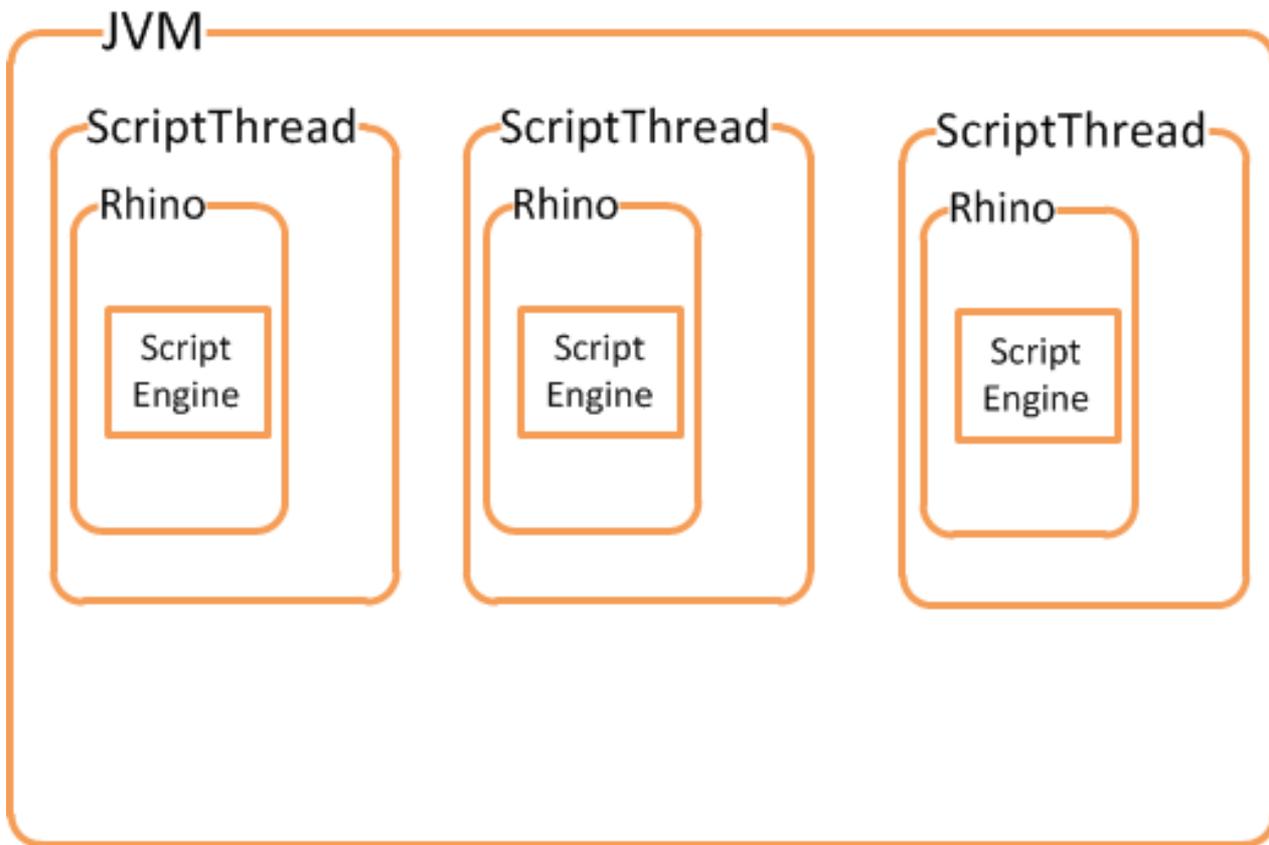
Доступные методы

Внутренние методы класса

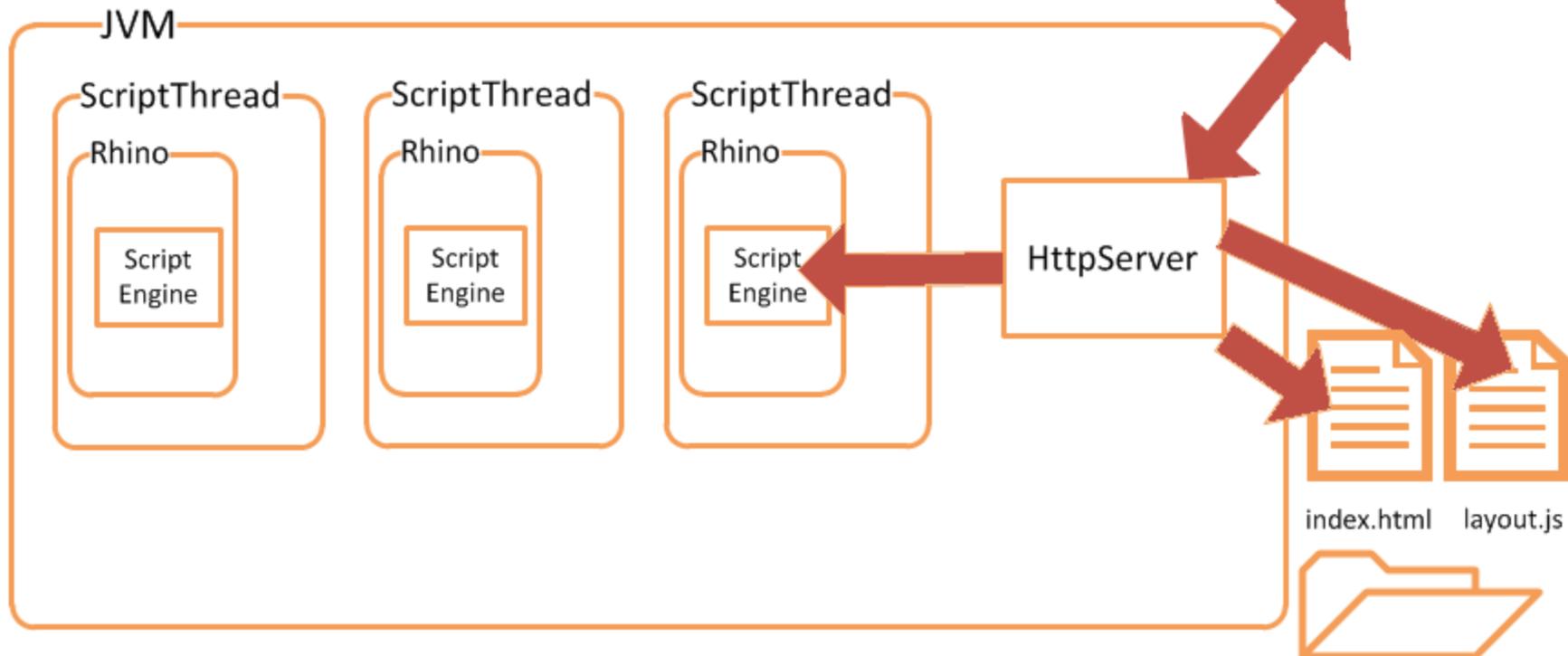
Главный класс, extends класс модуля

Вспомогательные классы

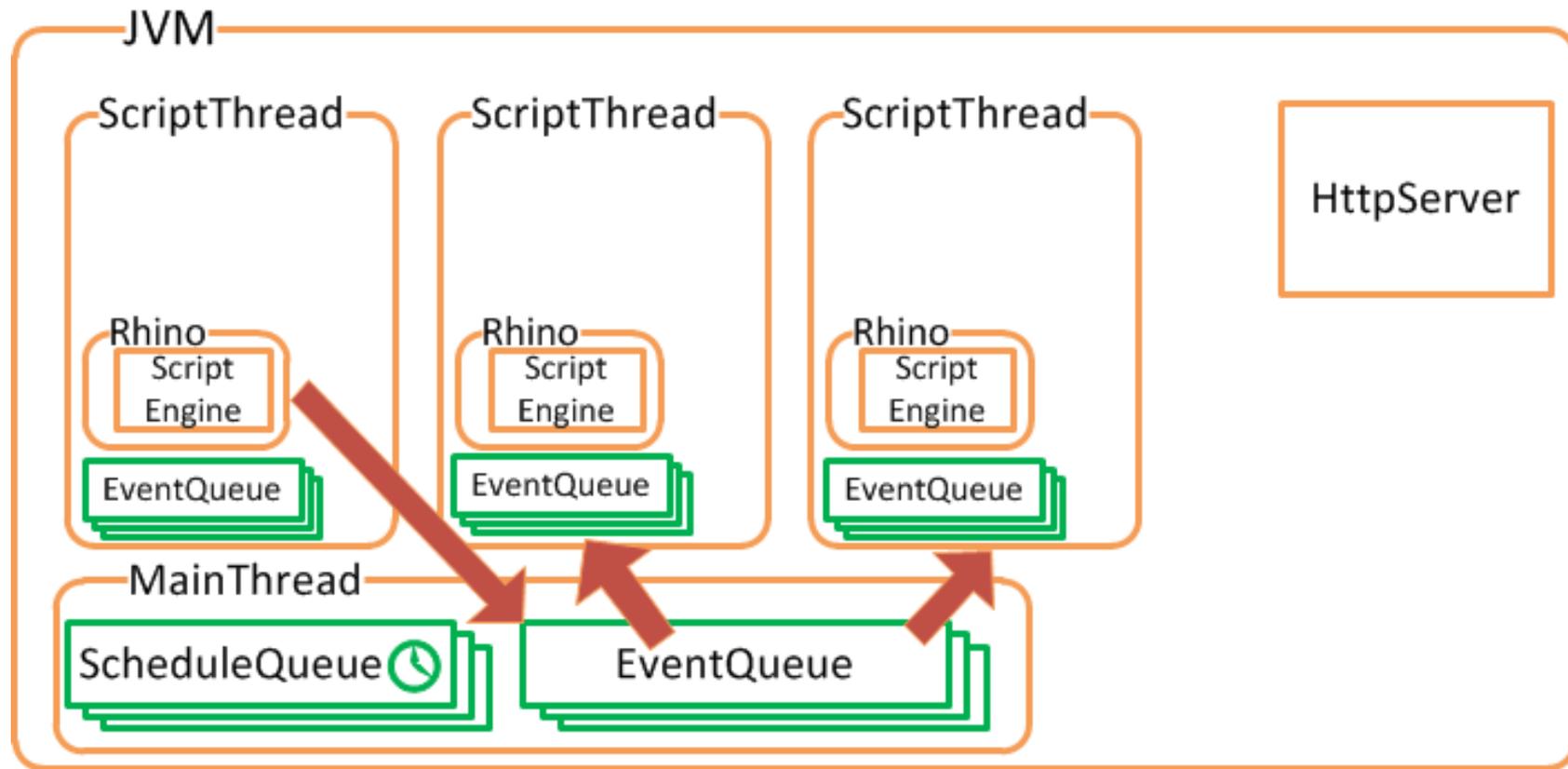
# МНОГОПОТОЧНОСТЬ



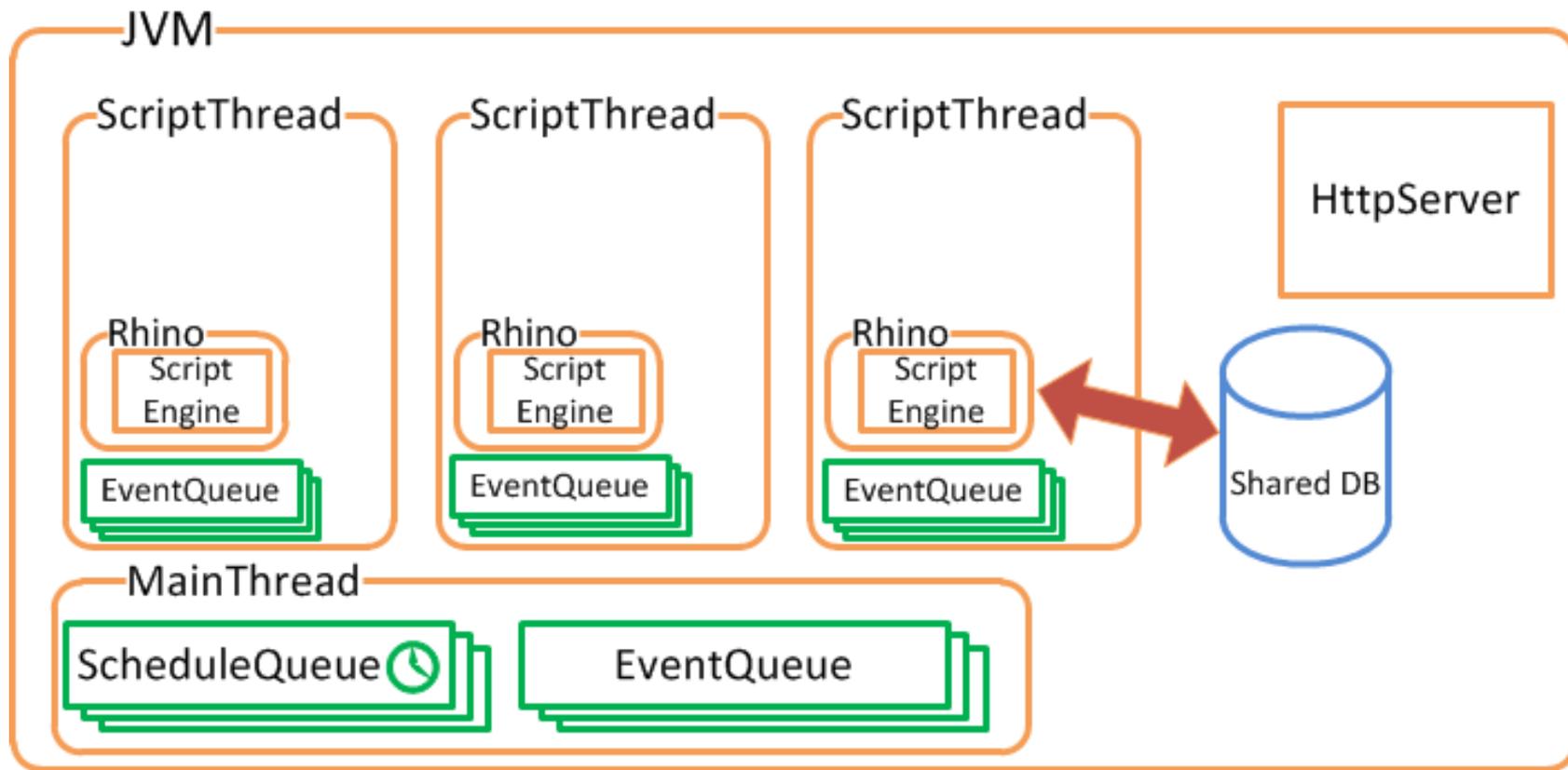
# IDE



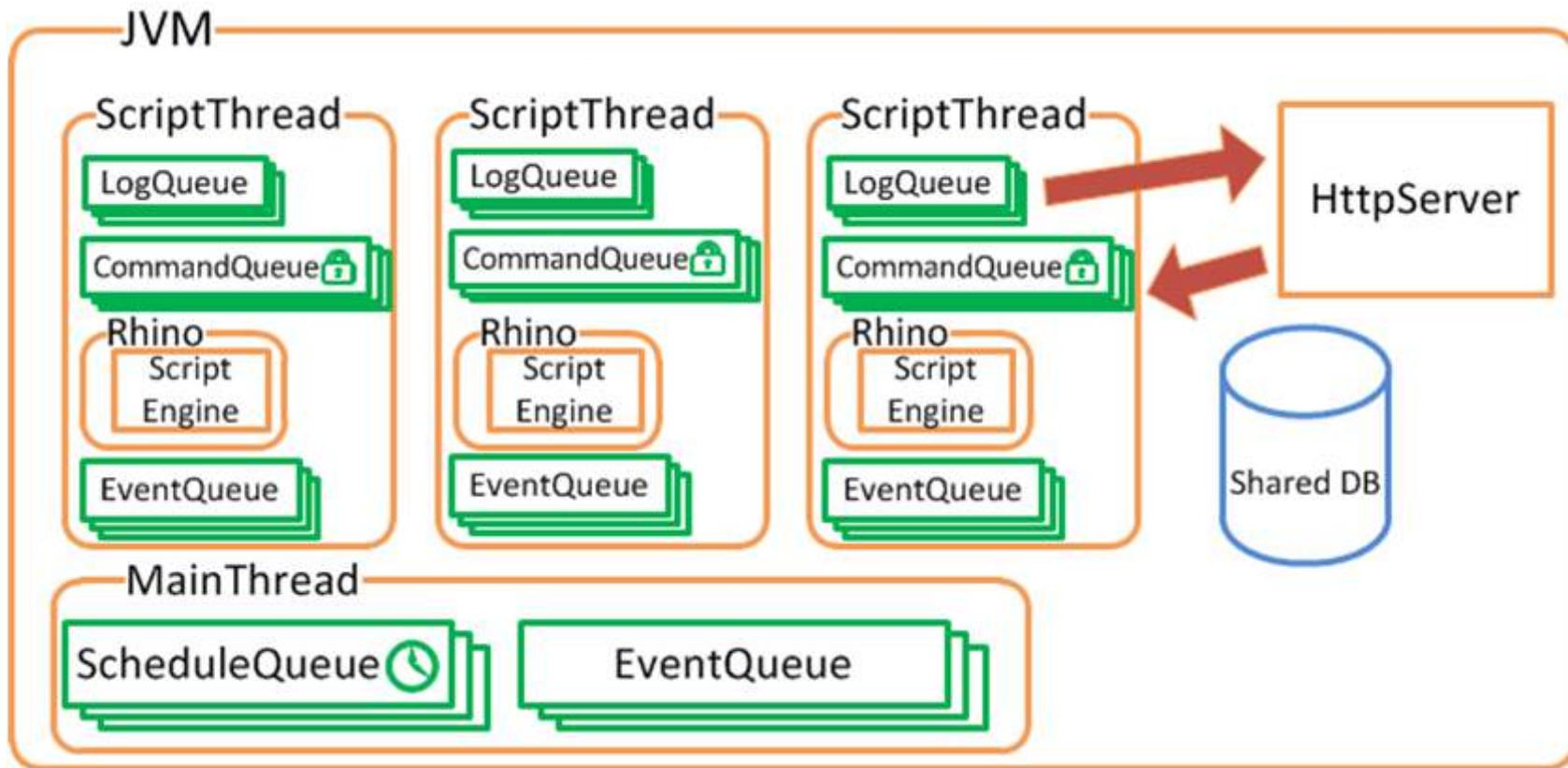
# События



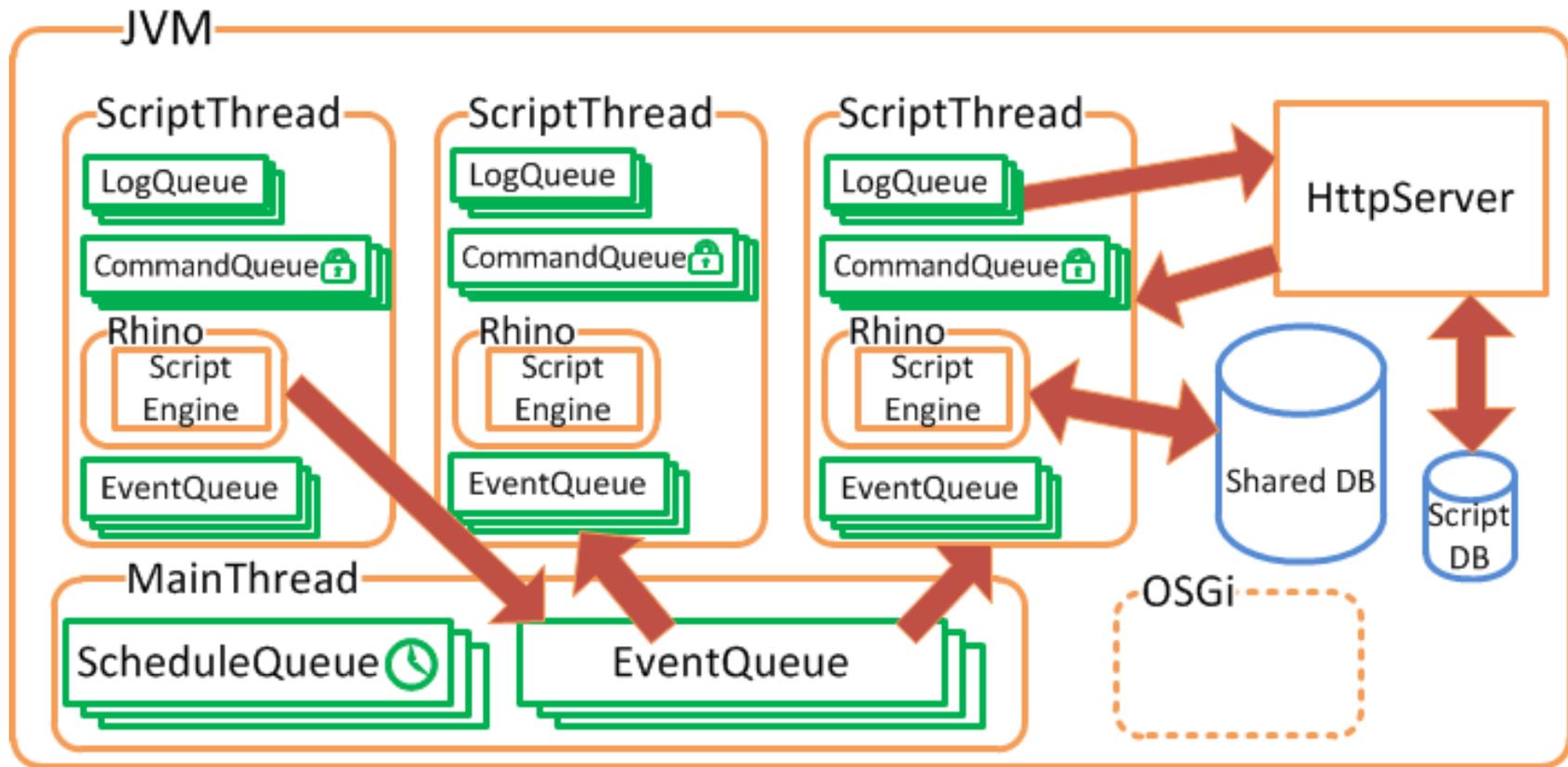
# СУБД



# Отладка



# Модули (пакеты)



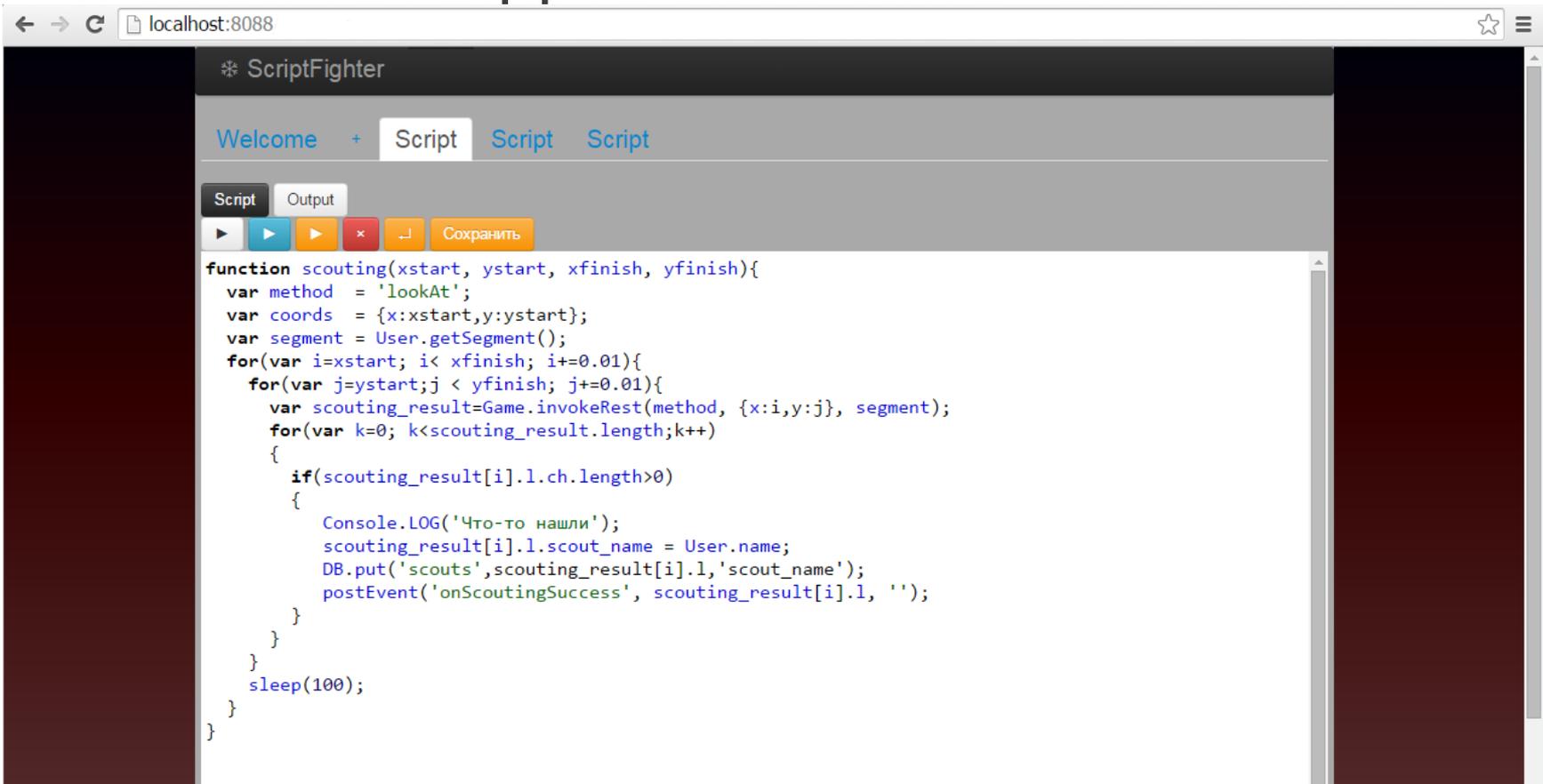
# Успехи

- Стабильная версия (uptime несколько недель)
- Боты появляются в топах игры
- Немного продается
- Работает в составе проектов
  - Рассылка push-уведомлений
  - Почтовая рассылка
  - Синхронизация удаленных баз SQLite
  - Обработка видео (в процессе)

# Планы

- Закончить перевод на Nashorn
- Реализовать пакеты
- Сделать lite-версию для iOS

# Как это выглядит



The image shows a web browser window with the address bar set to localhost:8088. The page title is "ScriptFighter". The browser interface includes a tab labeled "Script" and a toolbar with buttons for "Script", "Output", and "Сохранить" (Save). The main content area displays a JavaScript function named "scouting" with the following code:

```
function scouting(xstart, ystart, xfinish, yfinish){
  var method = 'lookAt';
  var coords = {x:xstart,y:ystart};
  var segment = User.getSegment();
  for(var i=xstart; i < xfinish; i+=0.01){
    for(var j=ystart; j < yfinish; j+=0.01){
      var scouting_result=Game.invokeRest(method, {x:i,y:j}, segment);
      for(var k=0; k<scouting_result.length;k++)
      {
        if(scouting_result[i].l.ch.length>0)
        {
          Console.LOG('Что-то нашли');
          scouting_result[i].l.scout_name = User.name;
          DB.put('scouts',scouting_result[i].l,'scout_name');
          postEvent('onScoutingSuccess', scouting_result[i].l, '');
        }
      }
    }
  }
  sleep(100);
}
```

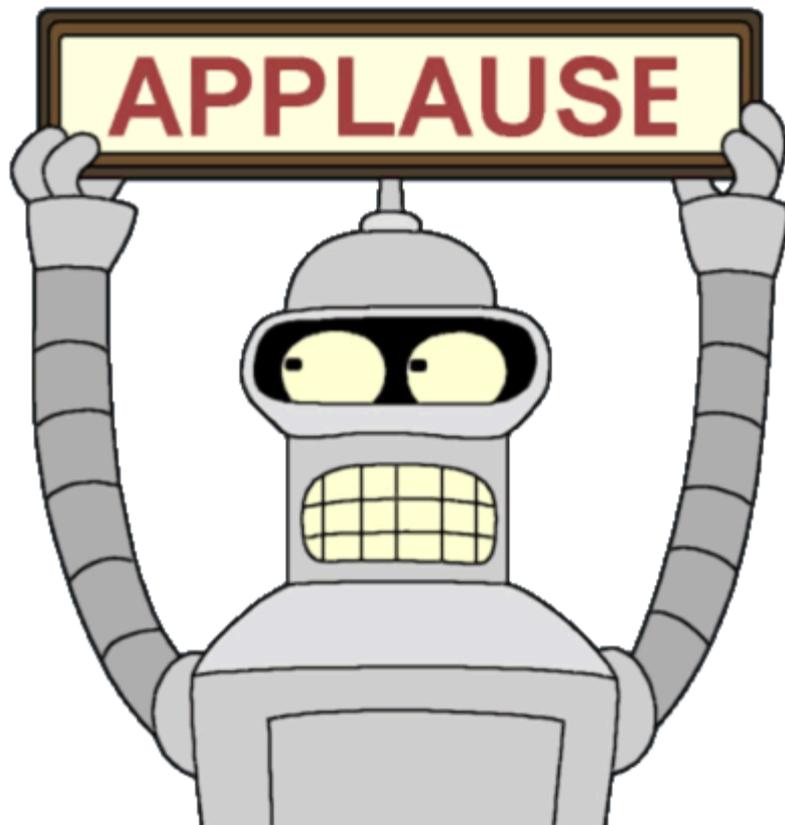
# Пример кода

```
function scouting(xstart, ystart, xfinish, yfinish){
  var method = 'lookAt';
  var coords = {x:xstart,y:ystart};
  var segment = User.getSegment();
  for(var i=xstart; i < xfinish; i+=0.01){
    for(var j=ystart; j < yfinish; j+=0.01){
      var scouting_result=Game.invokeRest(method, {x:i,y:j}, segment);
      for(var k=0; k<scouting_result.length;k++)
      {
        if(scouting_result[i].l.ch.length>0)
        {
          Console.LOG('Что-то нашли');
          scouting_result[i].l.scout_name = User.name;
          DB.put('scouts',scouting_result[i].l,'scout_name');
          postEvent('onScoutingSuccess', scouting_result[i].l, '');
        }
      }
    }
  }
  sleep(100);
}
```



Нет проблем при переходе на Nashorn!

На этом все



# references

- <https://jaxenter.de/java-8-nashorn-der-weg-zur-polyglotten-vm-1389>
- <http://habrahabr.ru/post/229247/>
- <https://en.wikipedia.org/wiki/ECMAScript>
- [http://www-  
archive.mozilla.org/js/liveconnect/lc3\\_method\\_overloading.html](http://www-archive.mozilla.org/js/liveconnect/lc3_method_overloading.html)
- [https://en.wikipedia.org/wiki/Da\\_Vinci\\_Machine](https://en.wikipedia.org/wiki/Da_Vinci_Machine)
- <https://www.jcp.org/en/jsr/detail?id=292>