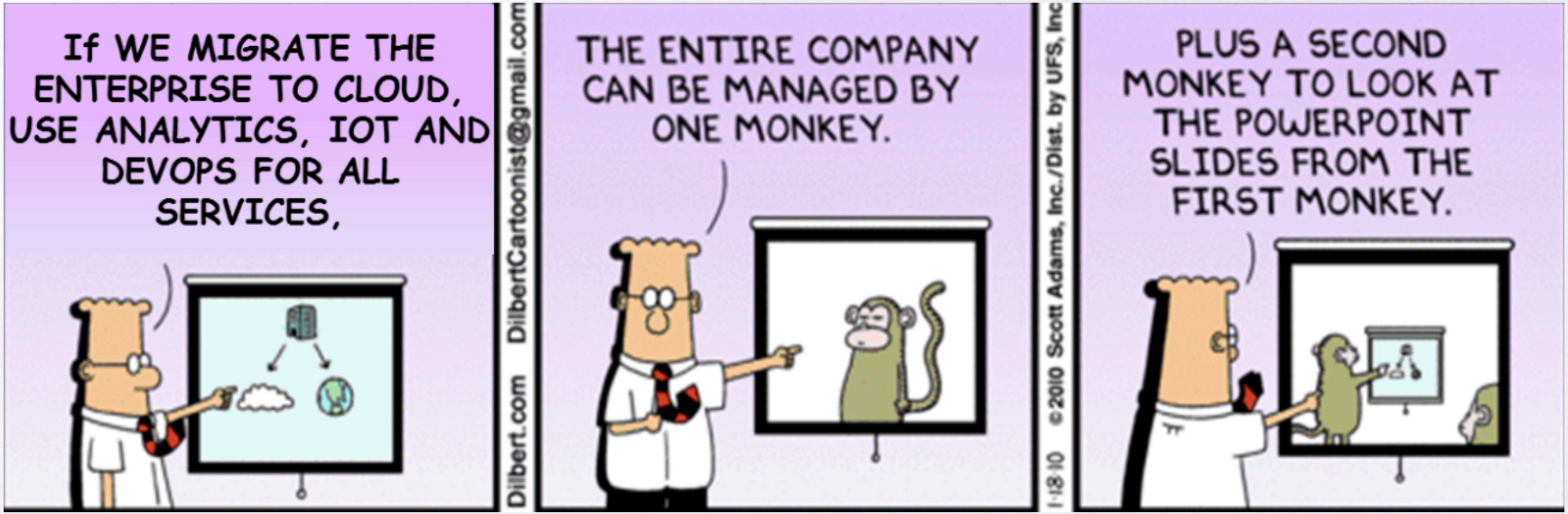
Погружение в Интернет Вещей с Java 9

WWW.BELL-SW.COM

2018


If WE MIGRATE THE ENTERPRISE TO CLOUD, USE ANALYTICS, IOT AND DEVOPS FOR ALL SERVICES,
THE ENTIRE COMPANY CAN BE MANAGED BY ONE MONKEY.
PLUS A SECOND MONKEY TO LOOK AT THE POWERPOINT SLIDES FROM THE FIRST MONKEY.
Dilbert.com DilbertCartoonist@gmail.com
1-18-10 ©2010 Scott Adams, Inc./Dist. by UFS, Inc

Кто здесь?

Александр Белокрылов

@gigabel

BELLSOFT

Liberica – Java 10 для Raspberry Pi
http://bell-sw.com

Бывшие работодатели:

ORACLE

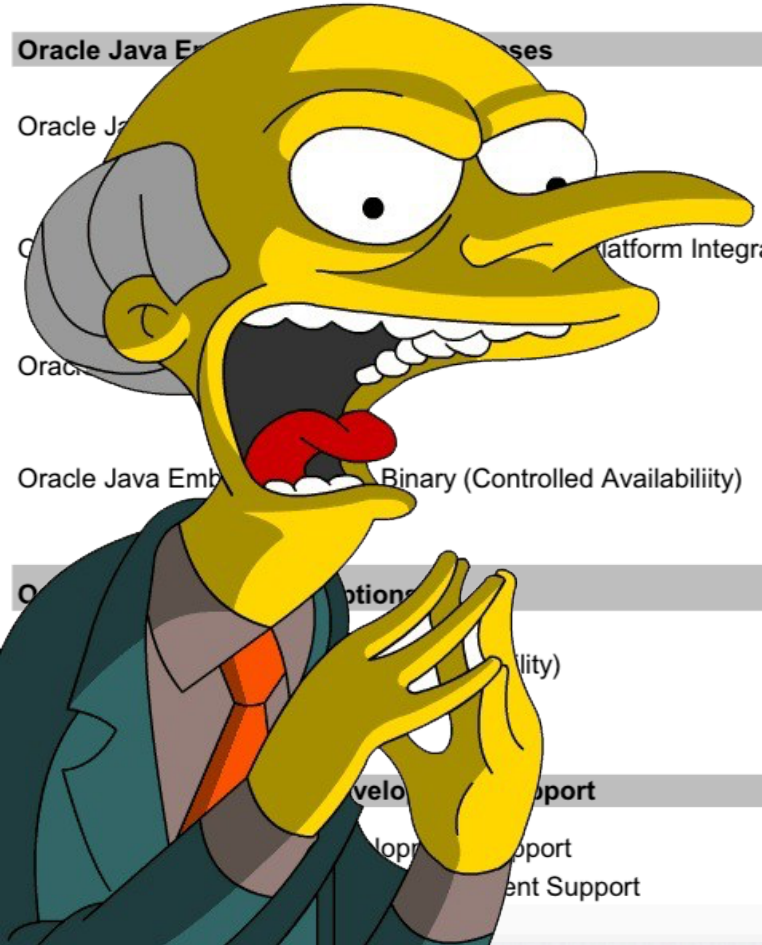
Sun microsystems

2012 год

Java™

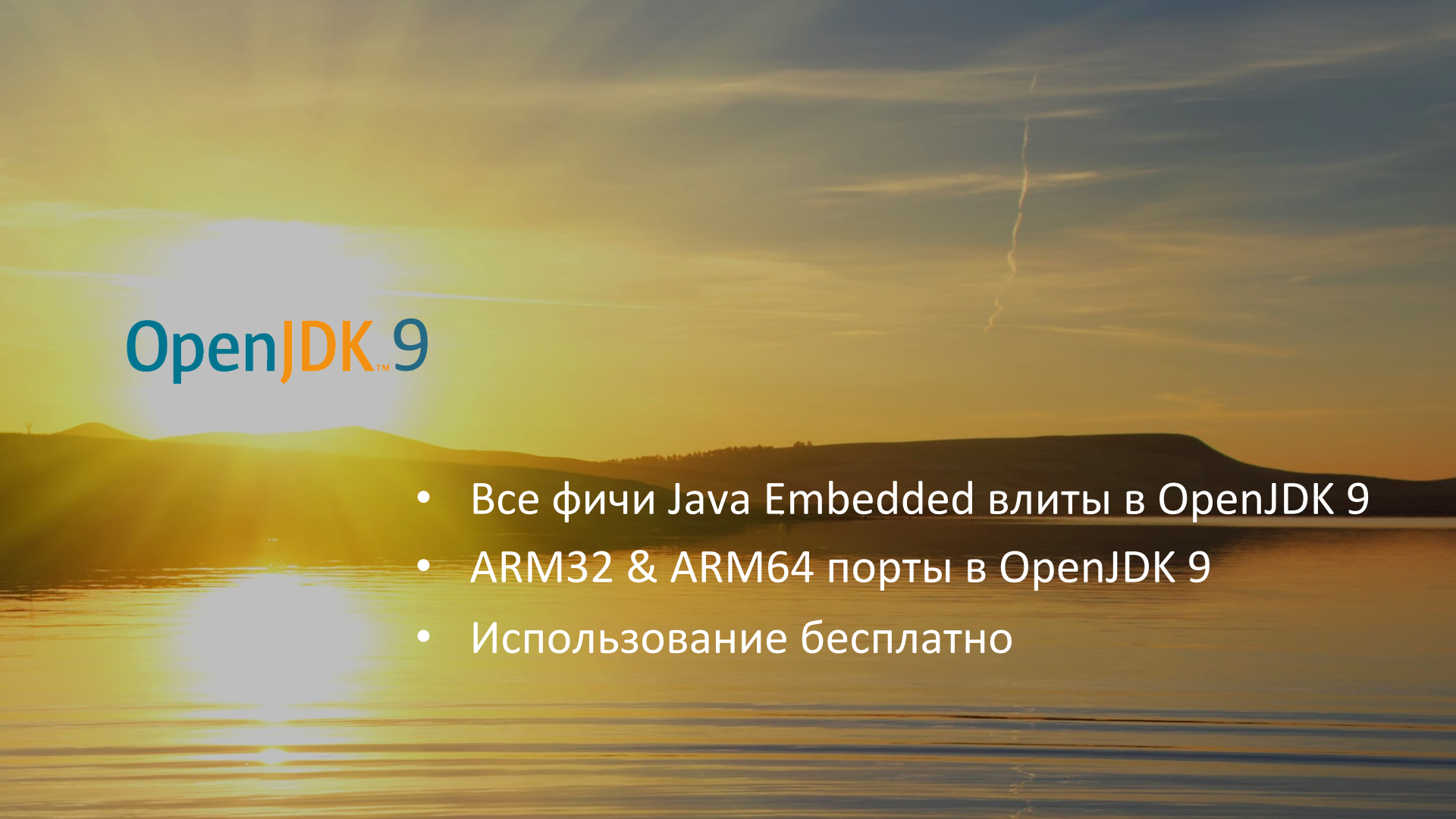
Java SE Embedded 7

No, it is not for free



	License Price	Software Update License & Support	Metric	Notes
Oracle Java E ses				
Oracle J	300.00	57.00	Java Embedded Processor	1, 3
C latform Integrator	300.00	57.00	Java Embedded Processor	1, 5
Oracl	300.00	57.00	Java Embedded Processor	1, 4
Oracle Java Emb Binary (Controlled Availabiliity)	450.00	85.50	Java Embedded Processor	1, 7
O tions				
lity)	1,000.00	190.00	Java Embedded Processor	1, 7
	Annual Fee			
velo pport				
lopr pport	50,000.00		Standard Binary	2
ent Support	75,000.00		Standard Binary	2, 7

Java SE Embedded 8 – последний релиз



- Все фичи Java Embedded влиты в OpenJDK 9
- ARM32 & ARM64 порты в OpenJDK 9
- Использование бесплатно

Интернет Вещей

IoT Architecture

IoT Gateway

IoT Cloud Service

Security
Device management
Stream processing
Data enrichment
Message store
Integration

Enterprise

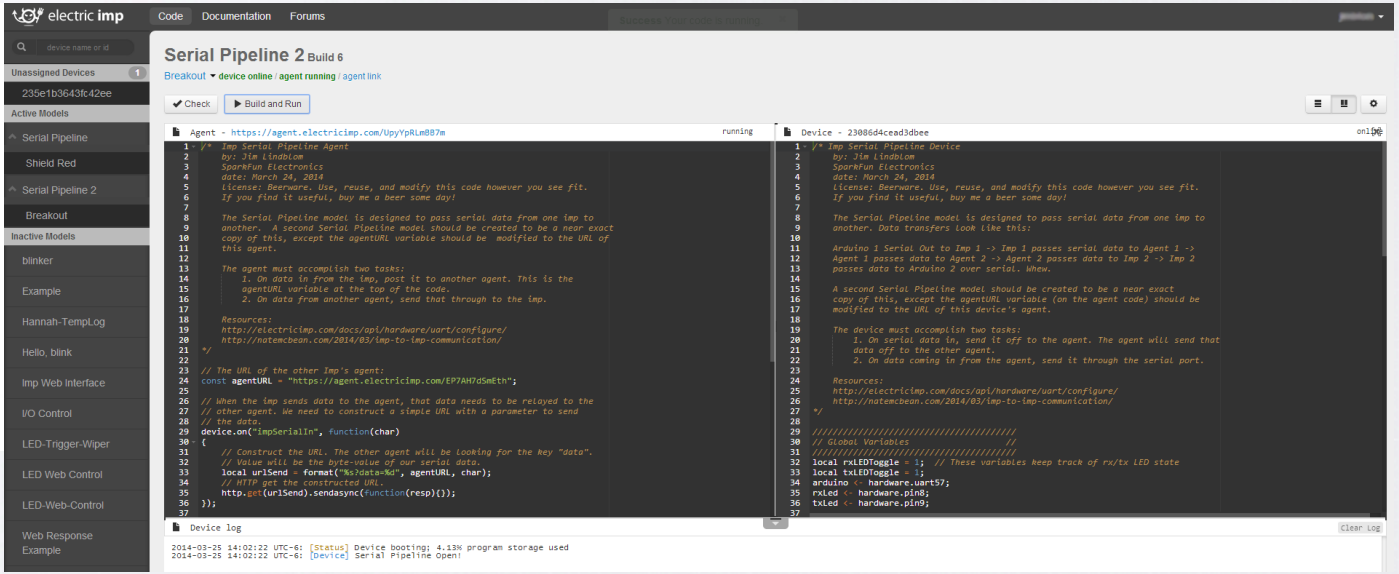
Consumer services

IoT устройства: Маленькие

- ARM Cortex M0 – M7
- MIPS32
  - RAM 4 – 512KB
  - ROM 4 – 2MB

Electric Imp www.electricimp.com

Electric Imp



IoT устройства: Большие

- ARM A5 – A9
- Intel Atom
  - RAM 256 MB+
  - ROM 256 MB+

IoT Gateway

- Большие устройства

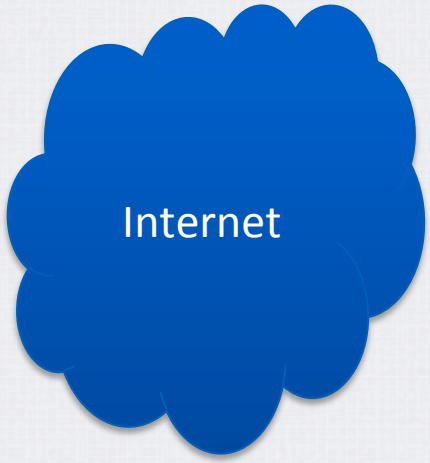
SuperMicro
Intel Core I7
RAM 64GB

Dell
Intel Atom
RAM 4GB

Eurotech
ARM Cortex A8
RAM 512 MB

Advantech
ARM Cortex A9
RAM 1GB

IoT gateway: Connectivity for Edge Devices



IoT gateway: Modular Software

Термометр

Газовая турбина

Connectivity

Internet

IoT gateway: Stream processing

Термометр

Газовая турбина

Стрим-аналитика

Connectivity

Internet

IoT gateway: Дополнительные функции

Термометр

Газовая турбина

Стрим-аналитика

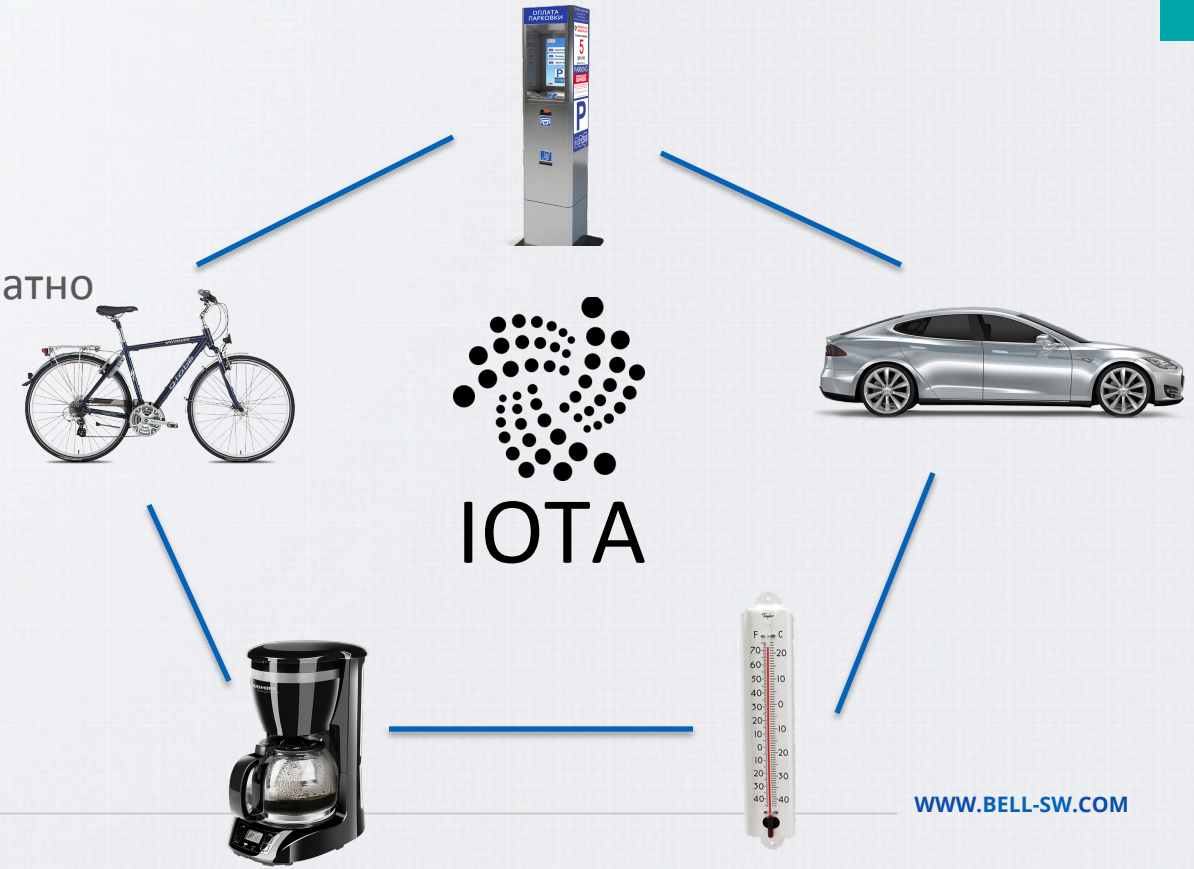
Платежная система

Connectivity

Internet

IOTA – криптовалюта IoT

- Расширяемая
- Децентрализованная
- Цена транзакции – бесплатно
- Клиент на Java



Java в IoT

- Eclipse Kura
  - OSS
  - Используется несколькими производителями Eurotech, Sierra Wireless

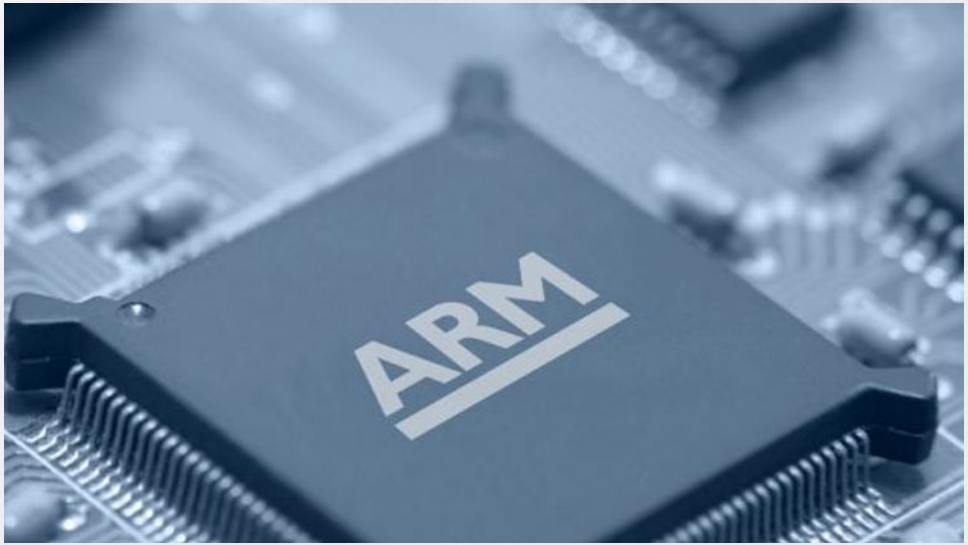
- Bosch IoT Gateway Software
- Oracle IoT Gateway
- GE Predix Edge

Java 9 дистрибутивы для IoT Gateway

- Oracle Java SE Embedded 8
  - No Java SE Embedded 9
  - No Binary for ARM in JDK 9
- Azul Systems, Zulu Embedded 9
- BellSoft, Liberica 9

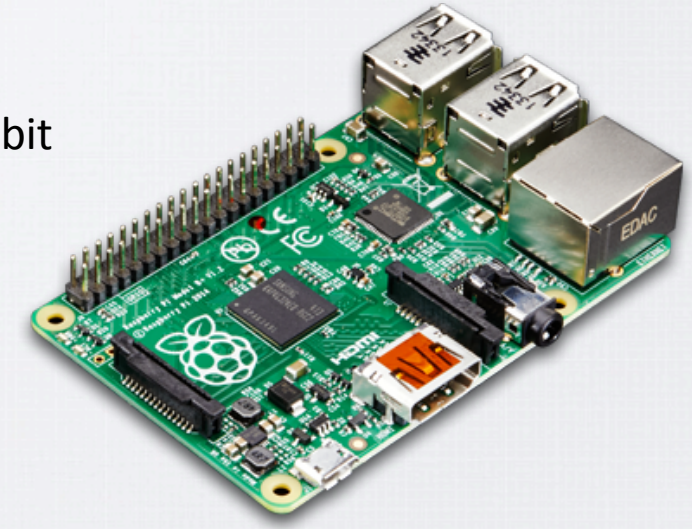
Open JDK ARM ports

- ARM
  - 32 bit / “64 bit”
    - ARM v6
    - ARM v7
    - ARM v8
- AARCH64
  - 64 bit only



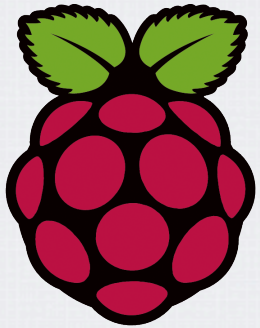
Raspberry Pi 3

SoC Broadcom 2837
4× ARM Cortex-A53 64-bit



Linux on Raspberry Pi 3

- Raspbian **32 Bit**
  - All Raspberry Pi versions supported
- Альтернативы:
  - openSUSE Linux 64 Bit



Java 9/10 фичи для IoT

- **Модуляризация**
- **AppCDS**
- **http/2**
- **Process API**

Модуляризация

Построим граф зависимостей

```
jdeps -R -s ThermoApp/
      ThermoApp -> java.base
      ThermoApp -> java.logging
      ThermoApp -> jdk.dio
```

Сделаем под себя


```
jlink \
    -p /usr/lib/jvm/jdk-9.0.4-bellsoft-arm32-vfp-hflt/jmods/ \
    --add-modules java.base,java.logging,jdk.dio \
    --output customjdk
```

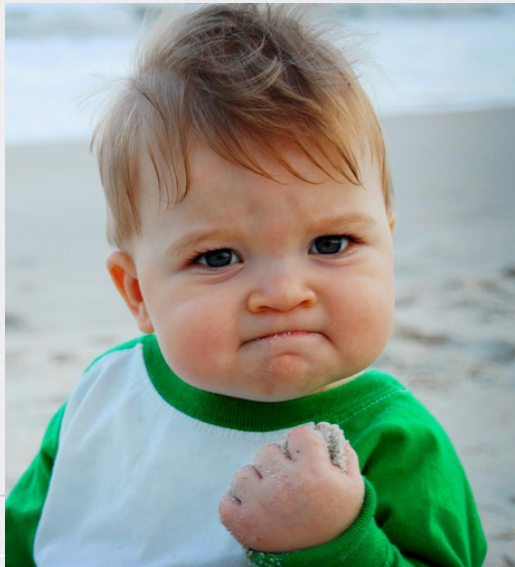
Избавимся от лишнего


```
cp customjdk minjdk

rm -rf minjdk/lib/server/ minjdk/lib/client/
```

Впечатляемся результатом

```
du -sh /usr/lib/jvm/jdk-9.0.4-bellsoft-arm32-vfp-hflt customjdk/ minjdk/
      379M    /usr/lib/jvm/jdk-9.0.4-bellsoft-arm32-vfp-hflt
      49M     customjdk
      28M     minjdk Java9 Platform Module System
```

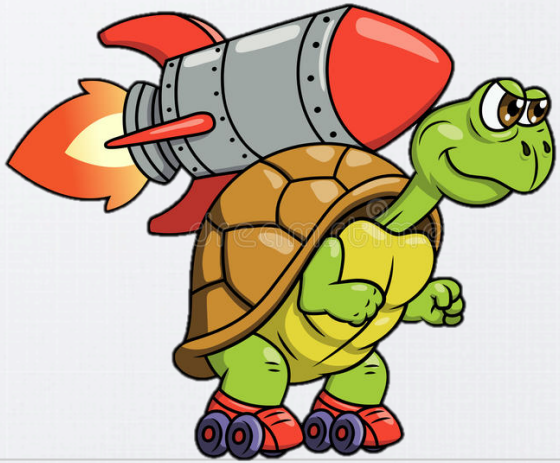


Class Data Sharing

Class Data Sharing

AppCDS

- Цели
  - Сократить время старта
  - Уменьшить использование RAM за счет разделения мета-данных классов
- Oracle WebLogic стартует на 30% быстрее с AppCDS



AppCDS on Raspberry Pi

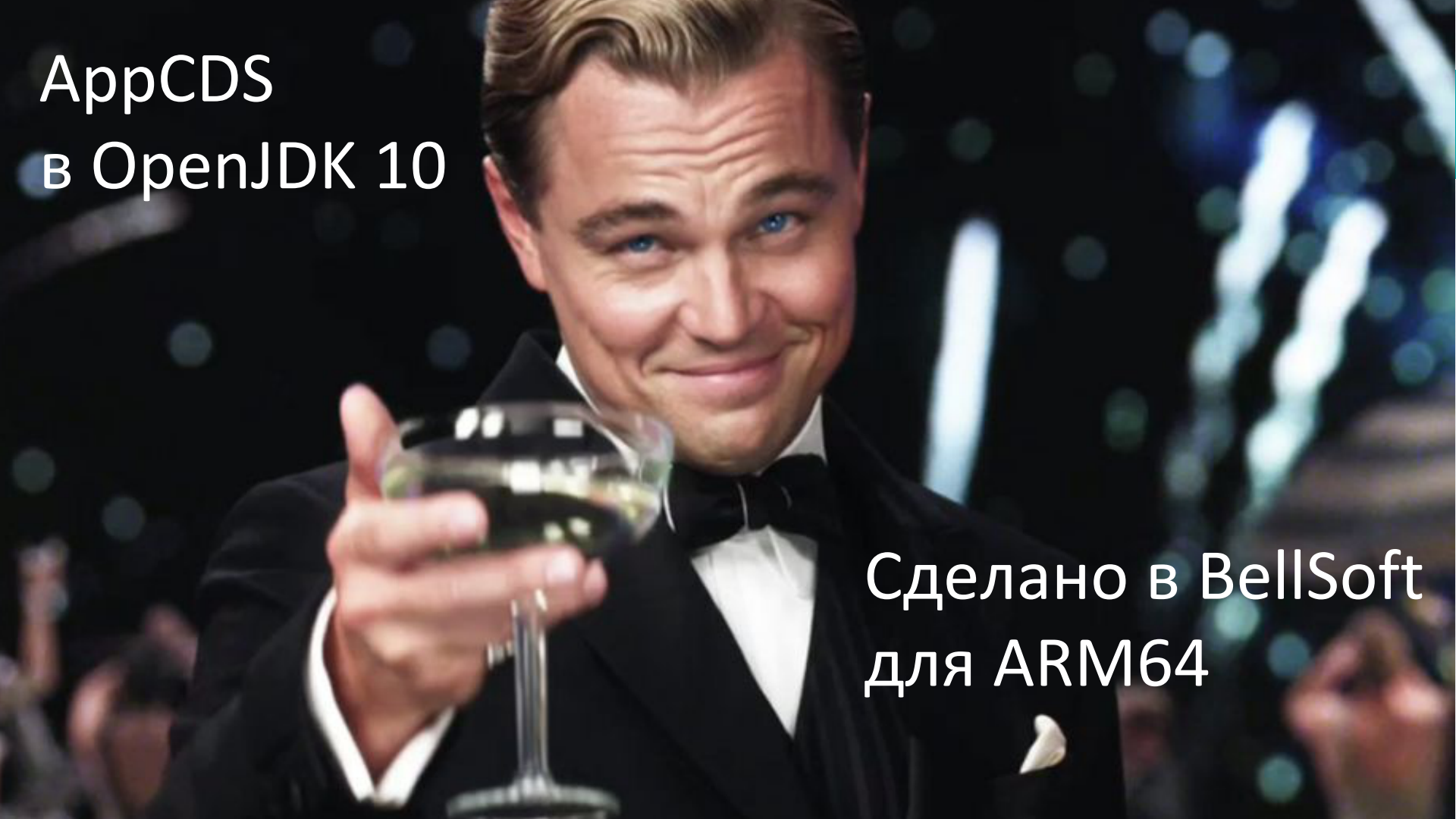
Apache Flink 1.4

На Raspberry Pi 3 стартует на 15% быстрее

AppCDS: доступность

- Oracle Java 8.40+
- Oracle Java 9.x

```
-XX:+UnlockCommercialFeatures
```


AppCDS
в OpenJDK 10
Сделано в BellSoft
для ARM64

Запустить приложение и создать список классов


```
-XX:+UseAppCDS -Xshare:off  \
    -XX:DumpLoadedClassList=classlist.clj
```

Создать кеш


```
-XX:+UseAppCDS -Xshare:dump  \
   -XX:SharedClassListFile=sample.classlist \
   -XX:SharedArchiveFile=sample.jsa
```

Получаем удовольствие

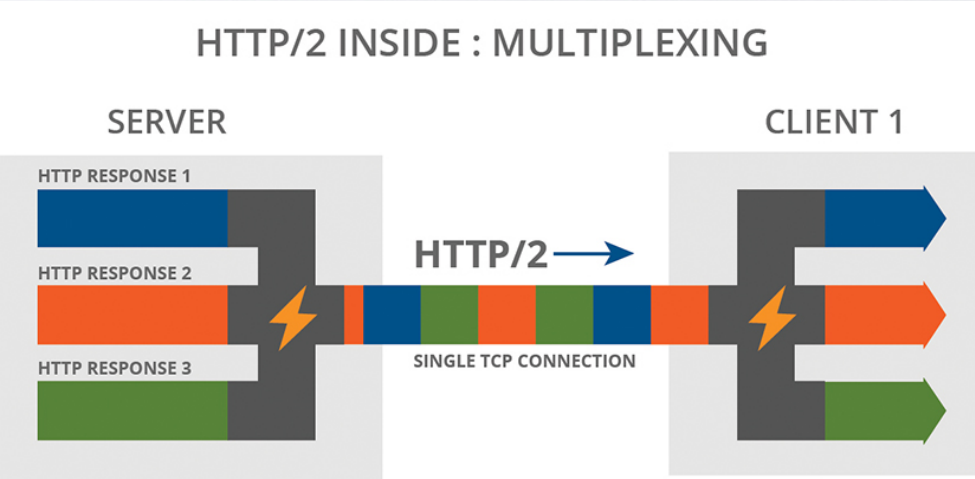
```
-XX:+UseAppCDS -Xshare:on  \
   -XX:SharedArchiveFile=sample.jsa
```

http/2


• Multiplexing
• Сжатие заголовков HPAC
• Server Push
• Бинарный протокол

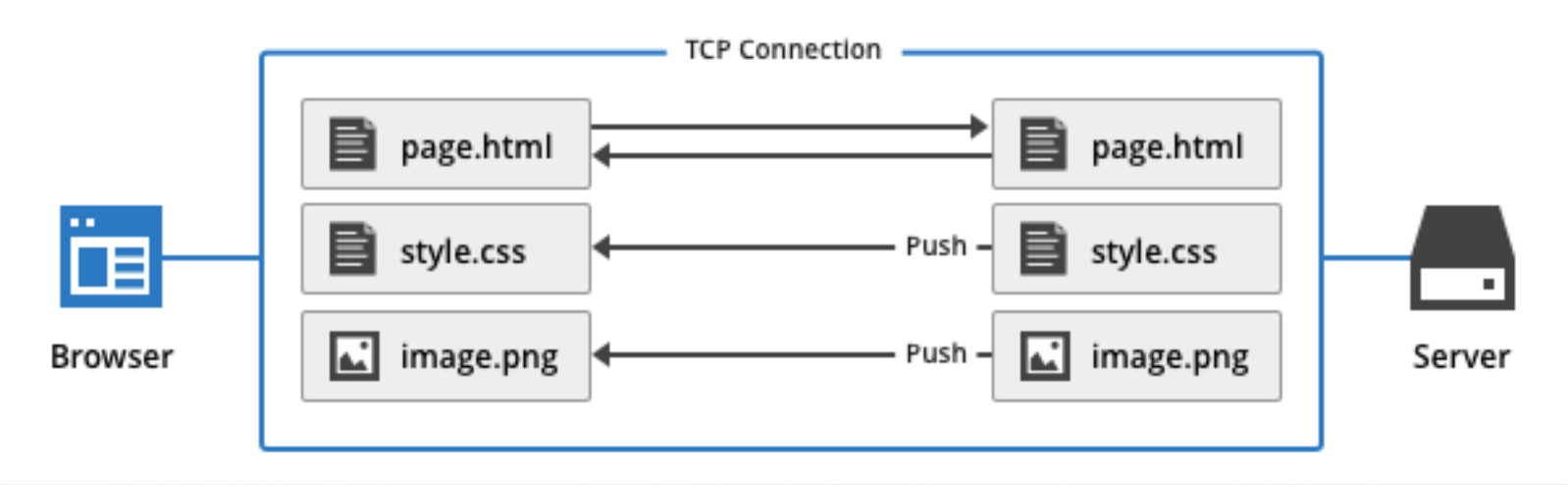
Multiplexing

- Коннекция может иметь несколько стримов
- Стрим разделен на фреймы
- Порядок Request/Response может быть произвольным



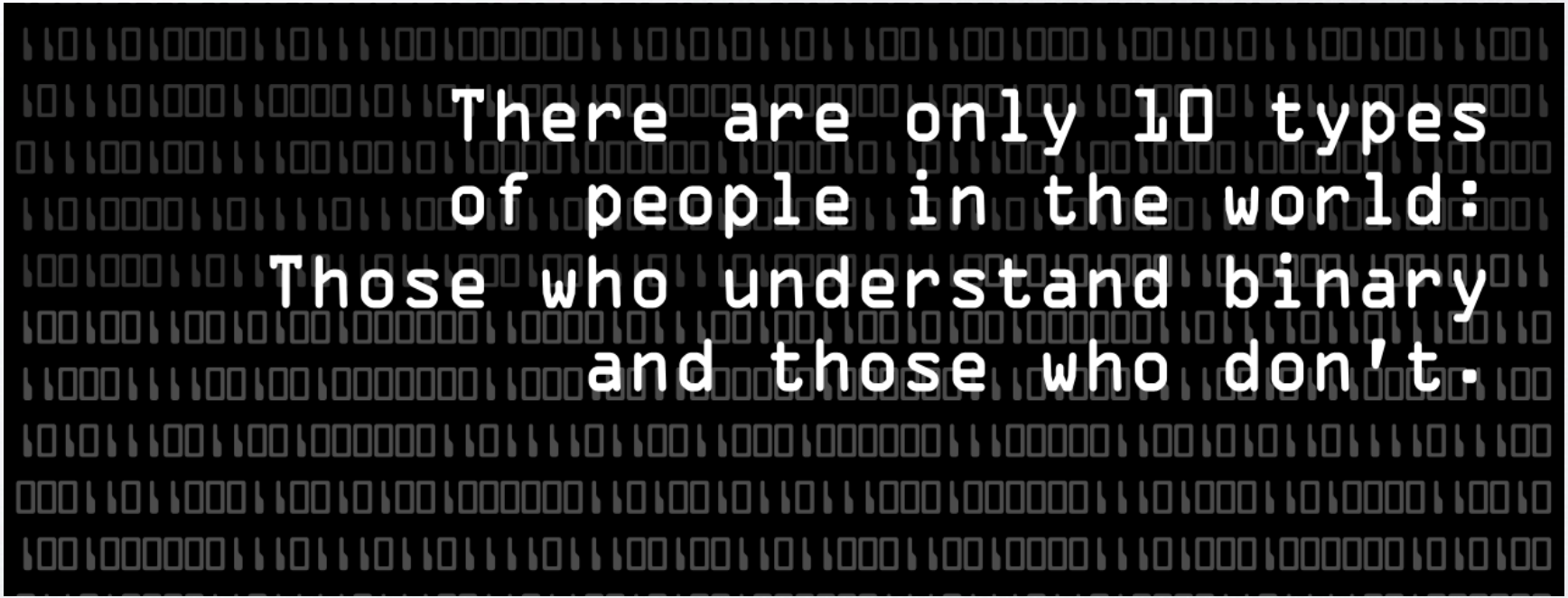
Server push

- Сервер может посылать данные до запроса
- Клиент хранит дополнительные данные в кэш
- Клиент может отказаться от получения данных



Сжатие заголовков HPAC

Бинарный протокол



Publisher/Subscriber или Request/Response?

http/2 в Java

- JEP 110
  - Java 9/10
  - jdk.incubator.httpclient
- JEP 321
  - Java 11
  - java.net.httpclient

HTTP/2, Java 9

Синхронный запрос

```
HttpClient client = HttpClient
        .newBuilder()
        .build();

HttpRequest request = HttpRequest
        .newBuilder(uriIp)
        .GET()
        .build();

HttpResponse<String> response = client
        .send(request, HttpResponse.BodyHandler.asString());
```

HTTP/2, Java 9

Асинхронный запрос

```
HttpClient client = HttpClient
        .newBuilder()
        .build();

HttpRequest asyncRequest = HttpRequest
        .newBuilder(uriIp)
        .GET()
        .build();

CompletableFuture<HttpResponse<String>> asyncResponse = client
        .sendAsync(asyncRequest, HttpResponse.BodyHandler.asString());
```

Process API


MAX IAS 140
CARB HEAT
PUSH COLD
FLAP WARNING:
DO NOT LOWER FLAPS
ABOVE 98 M.P.H.

OS Process access

JEP 102: Process API Updates

Расширение возможностей управления процессами операционной системы

Java 9: Process API

- **JEP 102: Process API Updates**
  - Расширен класс `java.lang.Process`
  - Новый интерфейс `ProcessHandle`

Java 9: Process API

```
static Stream<ProcessHandle> allProcesses()
Stream<ProcessHandle> children()
int compareTo(ProcessHandle other)
static ProcessHandle current()
Stream<ProcessHandle> descendants()
boolean destroy()
boolean destroyForcibly()
Long pid()
ProcessHandle.Info info()
boolean isAlive()
static Optional<ProcessHandle> of(long pid)
CompletableFuture<ProcessHandle> onExit()
Optional<ProcessHandle> parent()
boolean supportsNormalTermination()
```

Precess API: allProcesses()


```
ProcessHandle.allProcesses().filter(ProcessHandle::isAlive)
    .filter(process -> process.info().command().isPresent())
    .forEach(process -> printProcessInfo(process));
}
```

```
PROCESS INFORMATION
==================
Process id: 27557
Command: /Library/Java/JavaVirtualMachines/jdk-9.0.4.jdk/Contents/Home/bin/java
Arguments: ProcessAPISample
Command line: /Library/Java/JavaVirtualMachines/jdk-9.0.4.jdk/Contents/Home/bin/ja
Start time: 2018-03-04T21:17:21.963Z
Run time duration: 410ms
Owner: alexanderbelokrylov
```

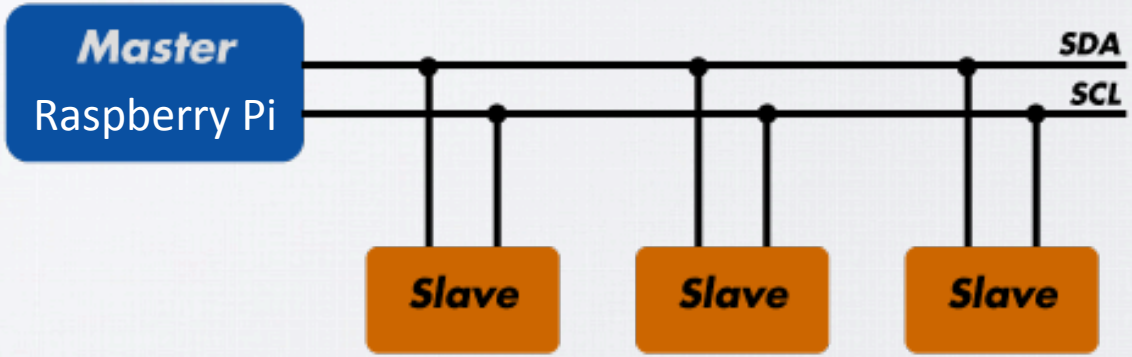
Precess API: Триггер на завершение процесса


```
Process p = new ProcessBuilder("/usr/bin/java","-version").start();
ProcessHandle process = p.toHandle();
CompletableFuture<ProcessHandle> onExit = process.onExit();

onExit.thenAccept(ph -> System.out.printf("PID %d terminated%n", ph.pid()));
```

Process API для IoT

I2C интерфейс



Датчик
Температуры
DS1621

Опрос I2C шины

- I2c-tools

```
pi@raspberrypi:~/$ i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: 60 -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

DEMO

CoolBeer Notification Demo

CoolBeer Notification Demo

CoolBeer Notification Demo

I2C

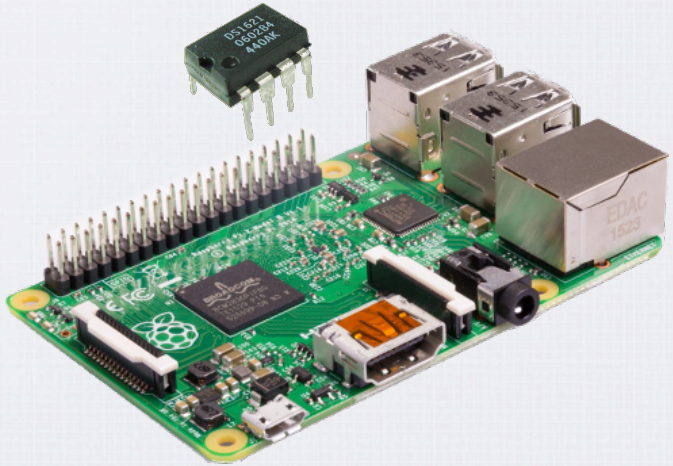
Raspberry Pi 3

Telegram

Java 9
DIO
Apache Flink
Telegram Bot

DIO (Device-IO API) example

- Connect DS1621 thermo sensor to RPi pins:
  - ground, 3.3v, I2C SDA/SCL
- Java coding:
  - open I2CDevice
  - write sensor config
  - get current temperature



```
class DS1621 extends I2CSensor {
   private final int I2C_DEVICE_ADDRESS  = 0x4f; // Sensor address
   // values from DS1621 Datasheet
   private final byte DS1621_SET_ACCESS_CONFIG = (byte)0xAC;
   private final byte DS1621_START_CONVERT     = (byte)0xEE;
   private final int  DS1621_READ_TEMPERATURE  = 0xAA;

   DS1621() throws IOException {
       I2CDeviceConfig config = new I2CDeviceConfig.Builder()
         .setAddress(I2C_DEVICE_ADDRESS, I2CDeviceConfig.ADDR_SIZE_7)
         .build();
  i2cDevice = (I2CDevice)DeviceManager.open(config);
      write(DS1621_SET_ACCESS_CONFIG, (byte)0);
      write(DS1621_START_CONVERT);
   }

   byte getTemperature() throws IOException {
      byte[] data = read(DS1621_READ_TEMPERATURE, 1);
      return data[0];
   }
}
```

Temperature stream processing

Sensor module → Движок потоковой аналитики ↔ Telegram module

Движки потоковой аналитики

Temperature stream processing

Sensor module

Apache Flink

Flink application

Telegram module

Выводы

- Java 9/10 позволяет использовать проверенные решения Stream Processing
- Функционал Java 9 отвечает требованиям IoT устройств
- OpenJDK получил код Java SE Embedded

Write once, run anywhere!