REST NO MORE
USING ACTORS FOR THE INTERNET OF (LEGO) TRAINS & RASPBERRY PIS

Johan Janssen, Info Support @johanjanssen42

Disclaimer:
No Lego was harmed beyond repair during the project.

CONTENT

- Why?
- Getting started
- Architecture
- Actors
- Remote actors
- Shared protocol
- HTTP vs Actors
- Conclusion
- Challenges
- Questions

WHY?


Why?


The Internet of THINGS

GETTING STARTED

MINIMAL INGREDIENTS FOR 1 TRAIN ABOUT € 50

- Raspberry Pi A+ / Raspberry Pi Zero
- Wifi dongle
  - EDUP Ultra-Mini Nano USB 2.0 802.11n
- USB battery pack
  - Anker® 2. Gen Astro Mini 3200mAh
- Infrared transmitter
  - Keyes 38KHz IR Infrared Transmitter Module for Arduino

COMPARISON

	Idle (mA)	Memory (MB)	CPU (Mhz)	Size (mm)
RPi A+	100	256	700	65 *56
RPi Zero	100	512	1000	65 * 30
RPi B+	200	512	700	85 *56
RPi 2 B	230	1024	4*900	85 *56
Particle Photon	80-100	128KB	120	38 * 21

ARCHITECTURE


Architecture

LTCC
(Angular)

LTCC
(Java)

RPi-Cam-Web-Interface
(C)

DeviceControl
(Java)

SwitchControl
(Java)

Infrared
(C and LI

Servo
(Python)

RFID
(C)

LTCC
(Angular)

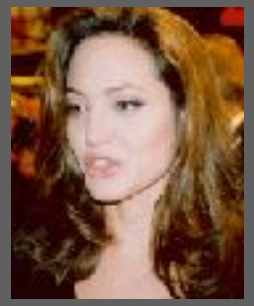
LTCC
(Scala/Akka)

RPi-Cam-Web-Interface
(C)

DeviceCon
(Scala/Akka)

ntrol
(Scala/Akka)

Leds with Photon
(C)



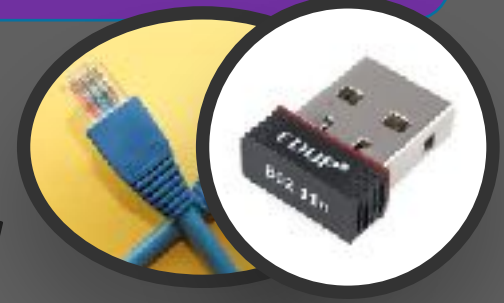
Infrared
(C and

RFID
(C)

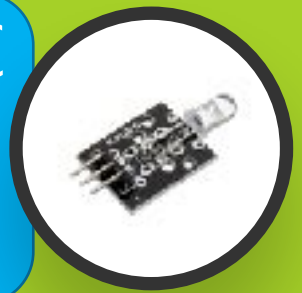
SwitchControl (Pi)

LTCC (Laptop / Pi)

Camera (Pi)



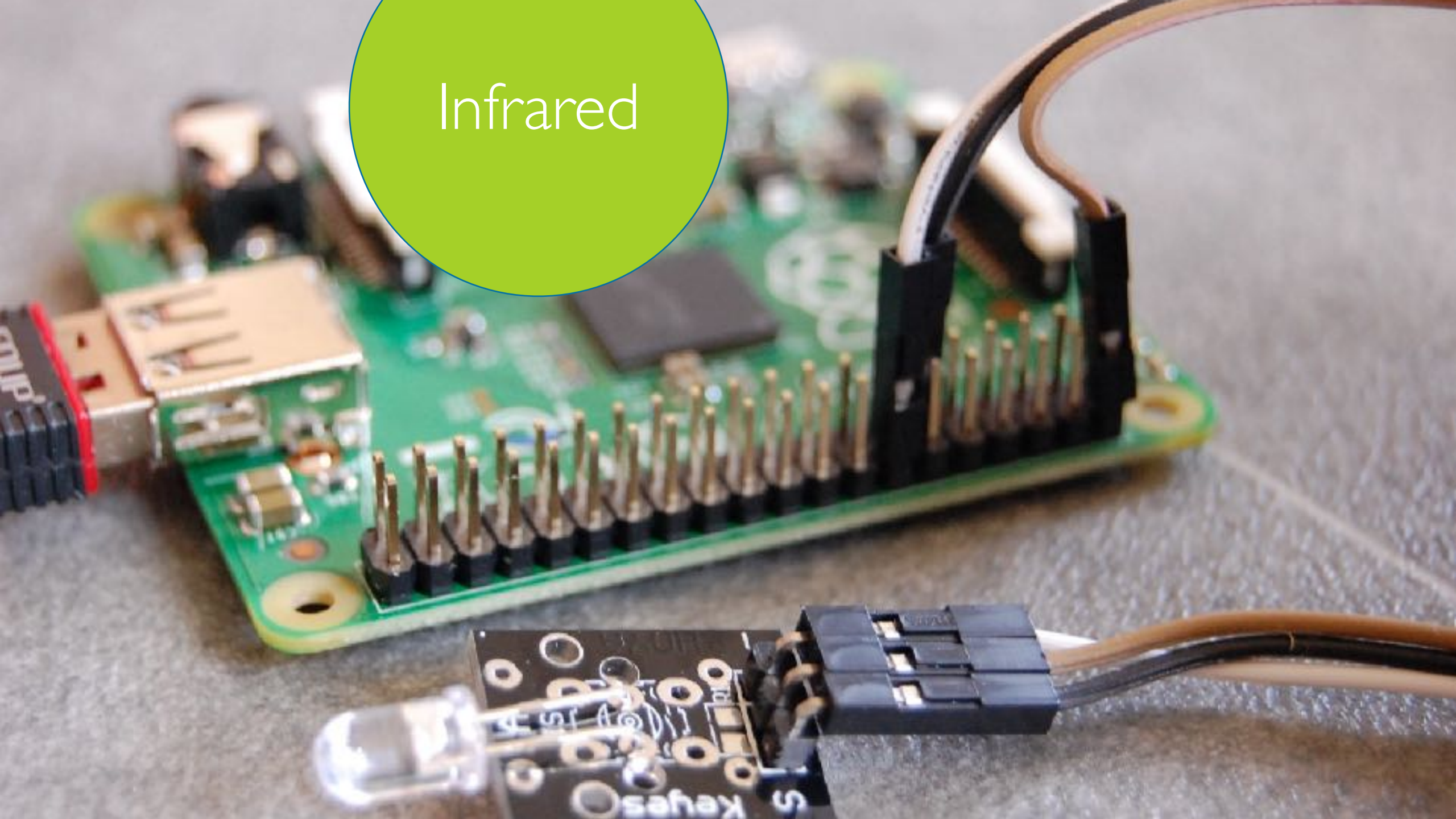
DeviceControl (Pi)



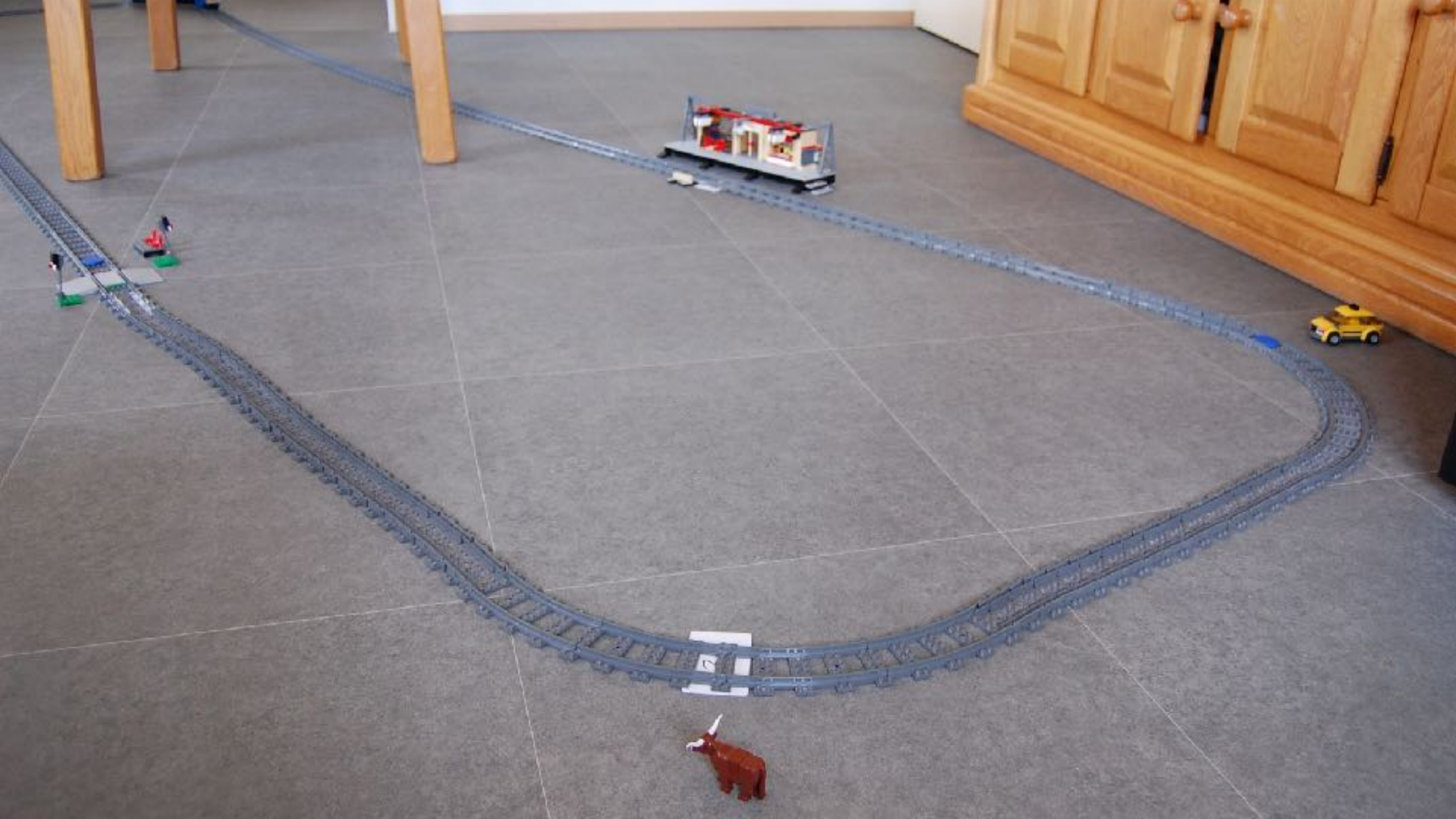
Lego Train


Original
controls

Infrared




Sound




Camera

RPi Cam Control v4.3.0: mycam@raspberrypi

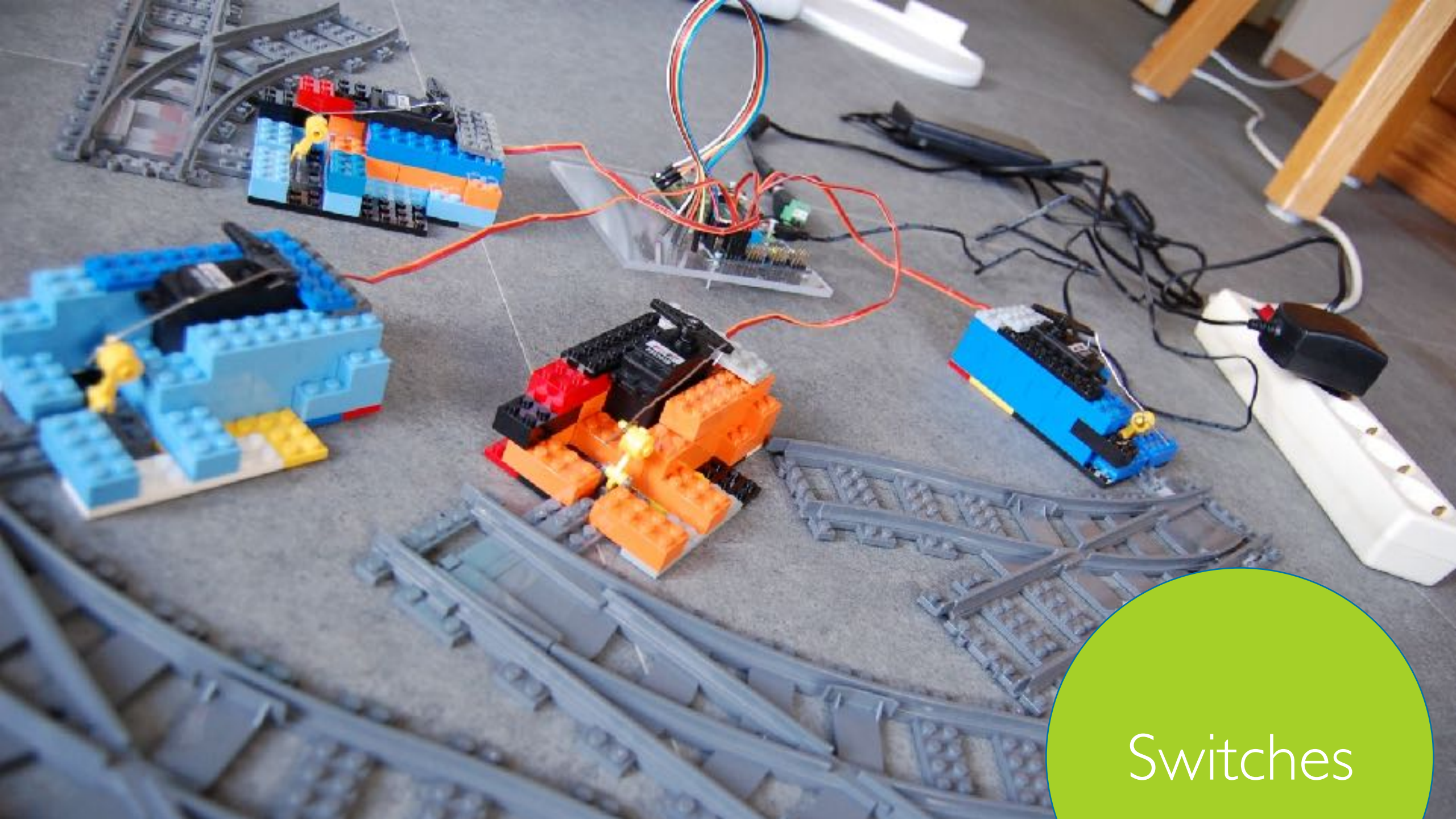
record video start | record image | timelapse start

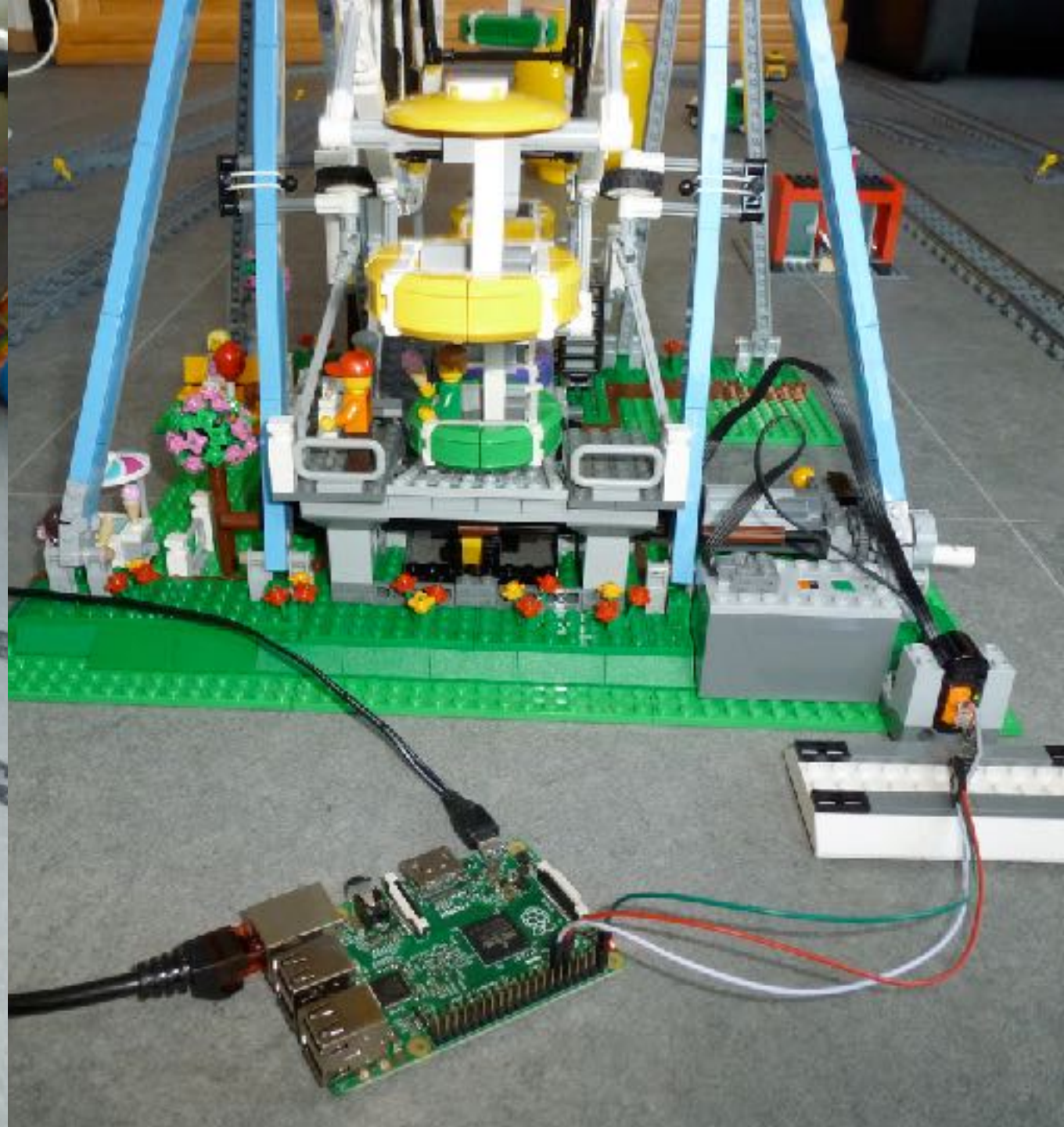
motion detection start | stop camera

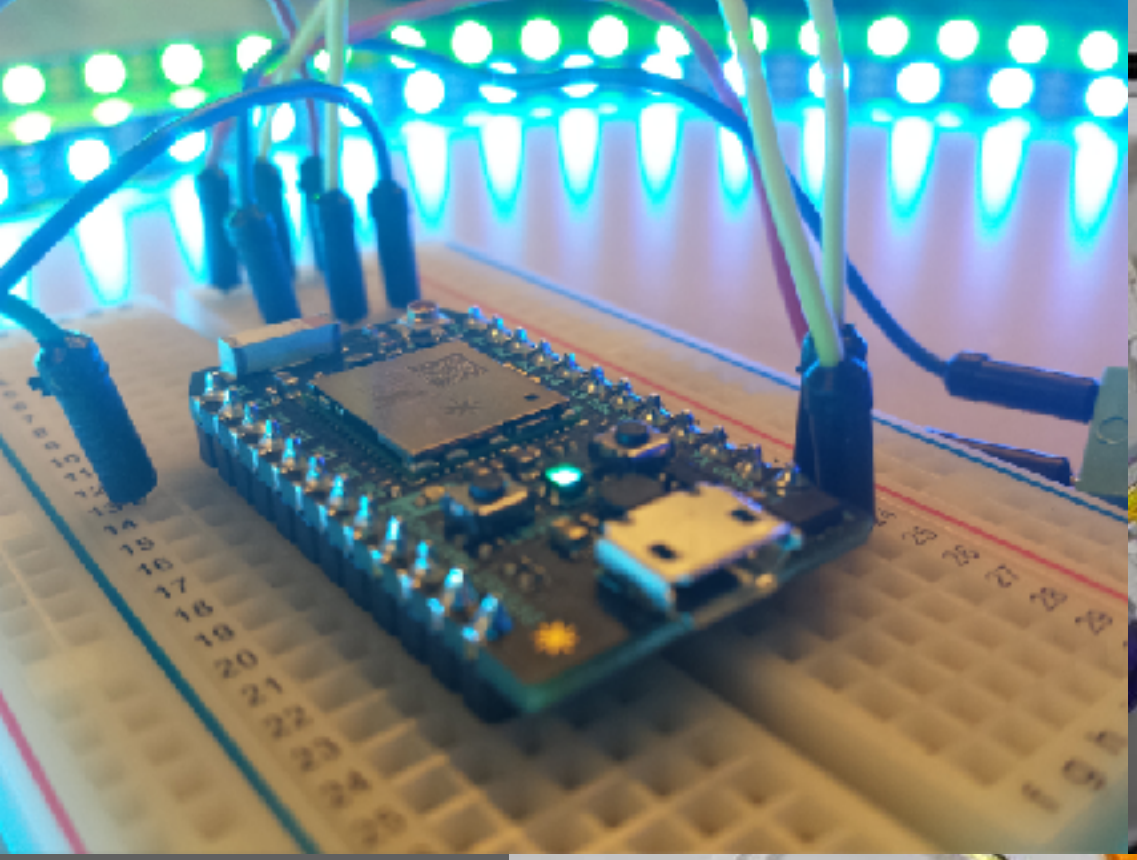
Download Videos and Images

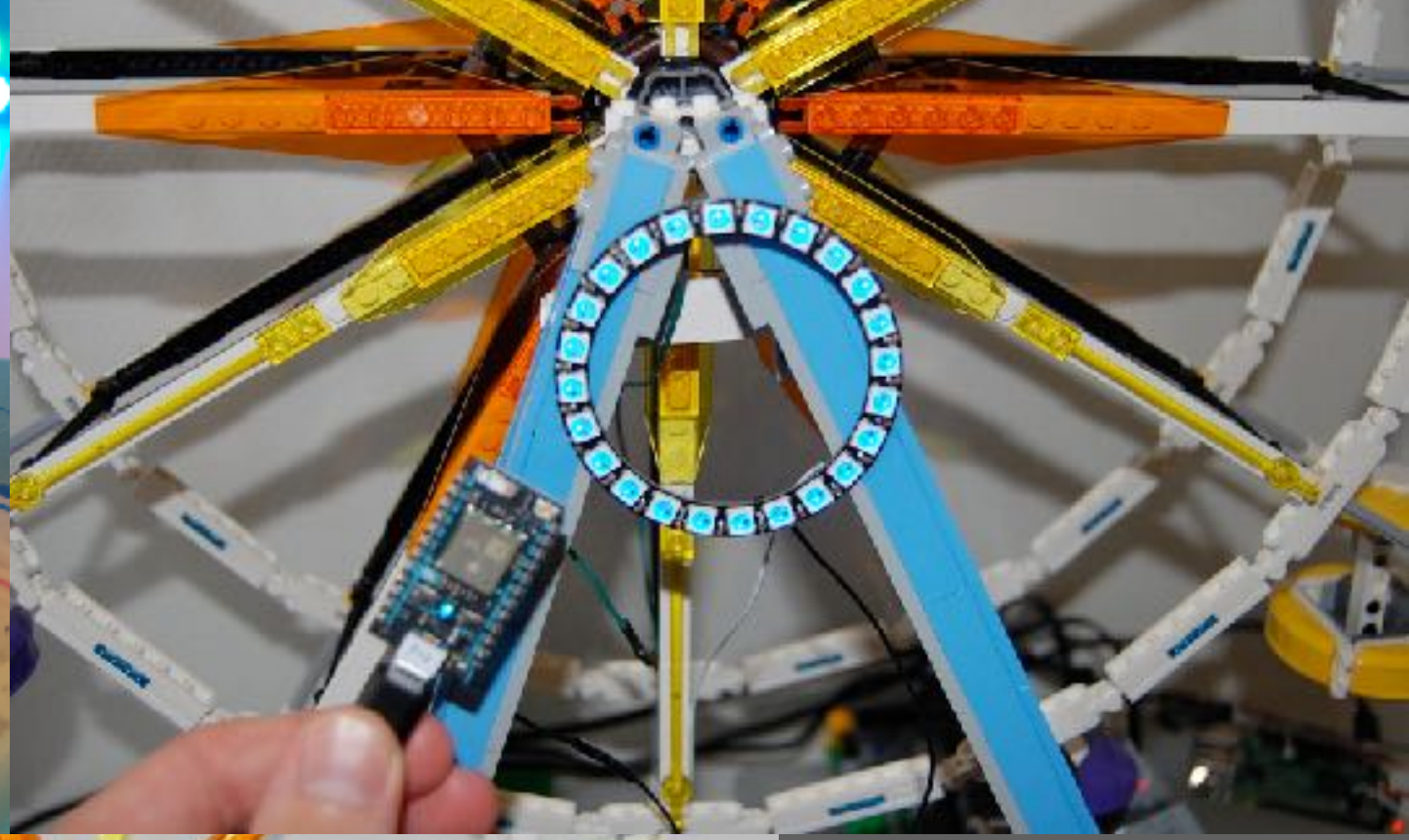
Settings

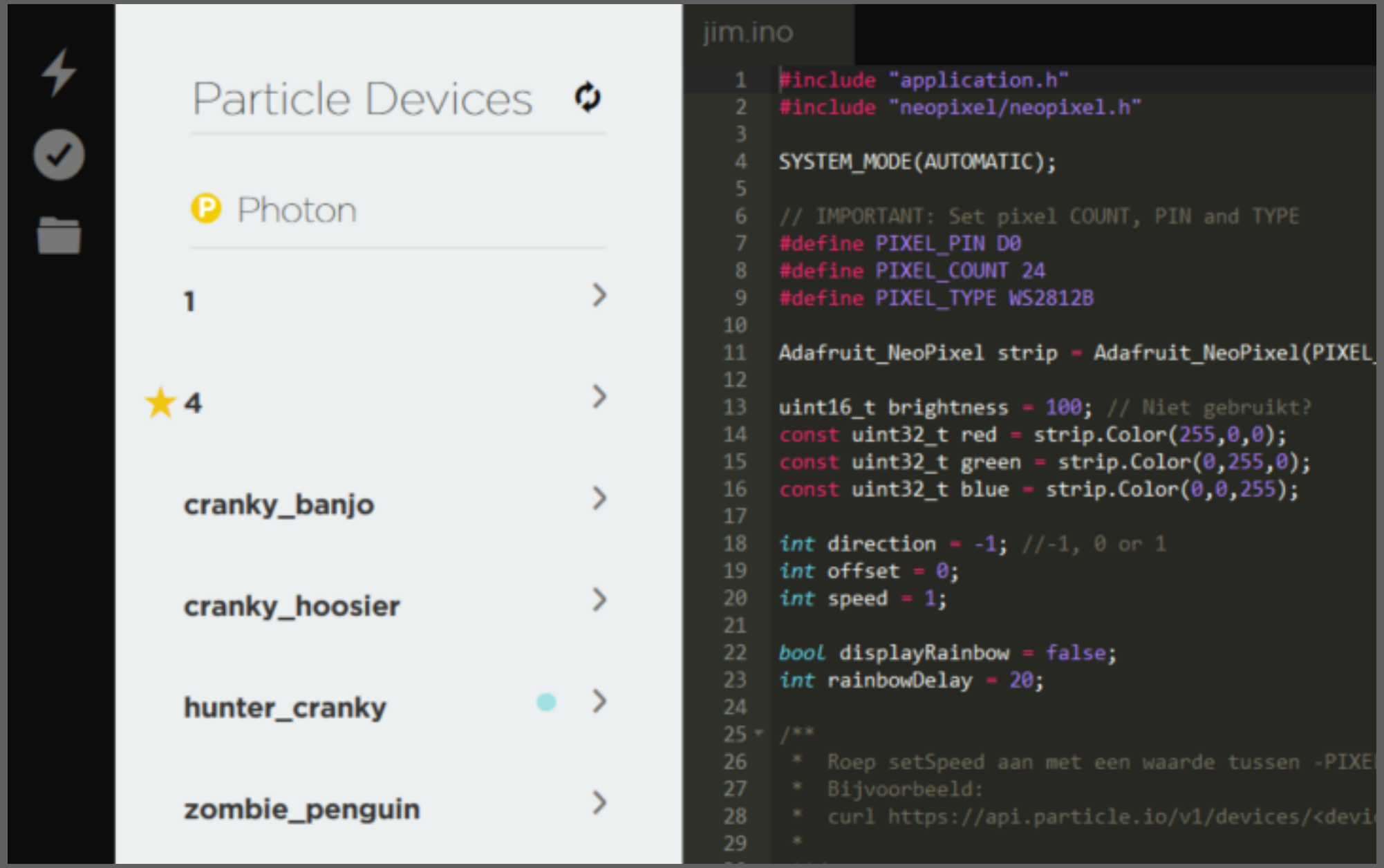
System


Switches

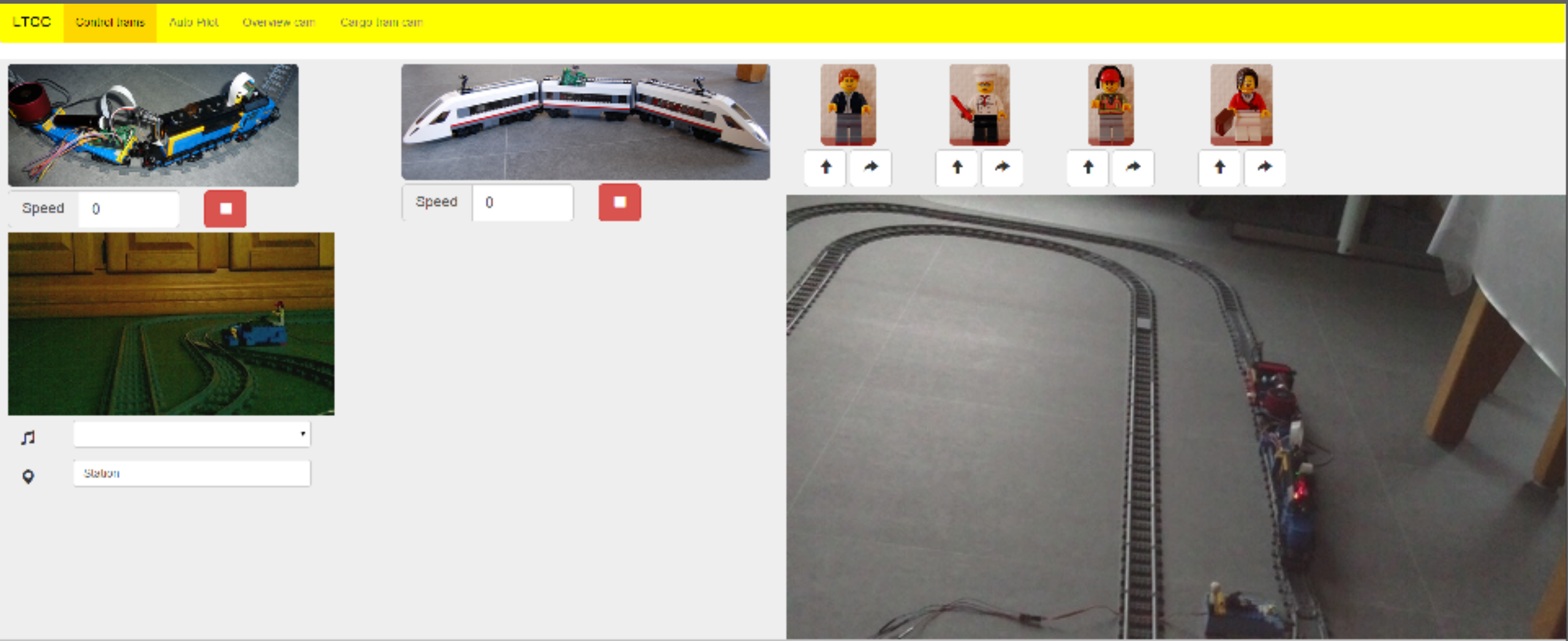







Particle Devices
Photon
1
4
cranky_banjo
cranky_hoosier
hunter_cranky
zombie_penguin
jim.ino
#include "application.h"
#include "neopixel/neopixel.h"
SYSTEM_MODE(AUTOMATIC);
// IMPORTANT: Set pixel COUNT, PIN and TYPE
#define PIXEL_PIN D0
#define PIXEL_COUNT 24
#define PIXEL_TYPE WS2812B
Adafruit_NeoPixel strip = Adafruit_NeoPixel(PIXEL
uint16_t brightness = 100; // Niet gebruikt?
const uint32_t red = strip.Color(255,0,0);
const uint32_t green = strip.Color(0,255,0);
const uint32_t blue = strip.Color(0,0,255);
int direction = -1; //-1, 0 or 1
int offset = 0;
int speed = 1;
bool displayRainbow = false;
int rainbowDelay = 20;
/**
* Roep setSpeed aan met een waarde tussen -PIXE
* Bijvoorbeeld:
* curl https://api.particle.io/v1/devices/<devi
*

LTCC APPLICATION



ACTORS

AKKA ACTORS

```
class Worker extends Actor {
  def receive = {
    case x =>
      println(x)
  }
}

val system = ActorSystem("ExampleActorSystem")

val workerActorRef = system.actorOf(Props[Worker])
workerActorRef ! "Hello conference"
```

REMOTE ACTORS

AKKA REMOTE ACTOR CALL

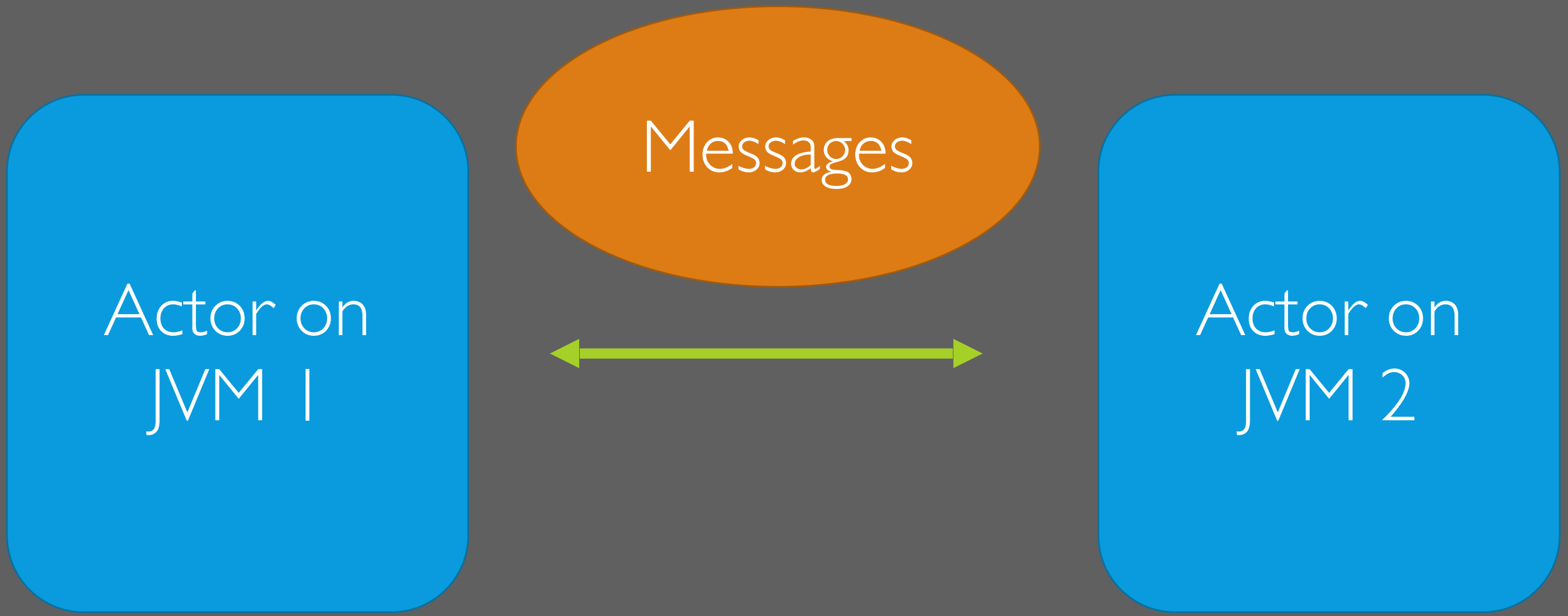
```
val workerActorRef =
system.actorOf(Props[Worker])
```

```
val workerActorRef =
system.actorSelection("akka.tcp://
ExampleActorSystem@127.0.0.1:9005
/user/workerActor")
```

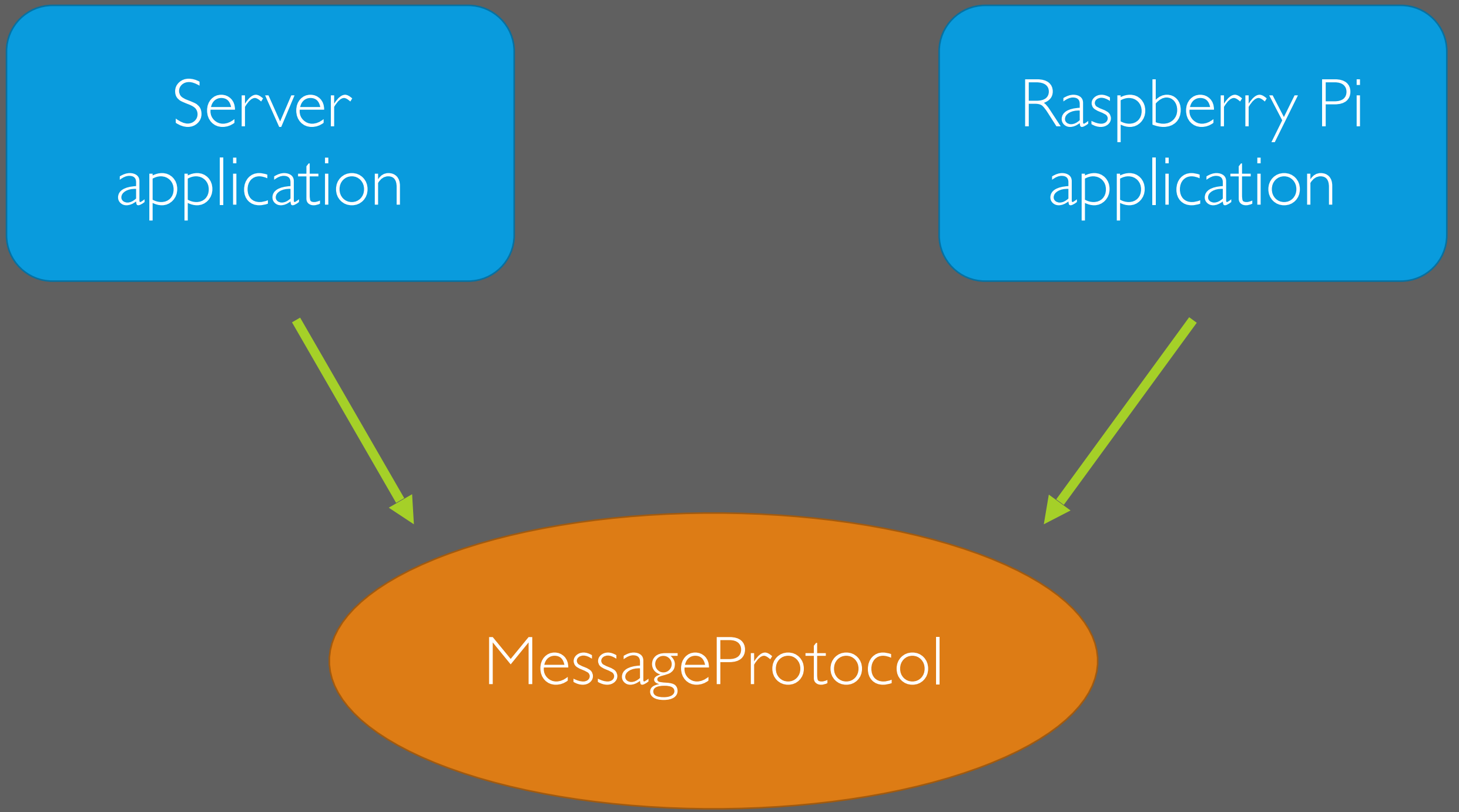
AKKA REMOTE ACTOR CONFIGURATION

```
akka {
  actor {
    provider = "akka.remote.RemoteActorRefProvider"
  }
  remote {
    enabled-transports = ["akka.remote.netty.tcp"]
    netty.tcp {
      hostname = "127.0.0.1"
      port = 9002
    }
  }
}
```

SHARED PROTOCOL


Messages
Actor on JVM 1
Actor on JVM 2

CONCRETE EXAMPLE


Server application
Raspberry Pi application
MessageProtocol

EXAMPLE MESSAGE

```
object MusicServiceMessage {
  case class Play(filename: String)
  case class MusicList(filenames: List[Song])
}
```

MESSAGE USED BY APPLICATION

```
val actorRef = context.actorSelection(
    "akka.tcp://[Actorsystem]@
    [IP]:[port]/user/musicservice")

actorRef !
[packagename].MusicServiceMessage.Play(filena
me)
```

HTTP VS REMOTE ACTOR

ADVANTAGES **REMOTE ACTORS**

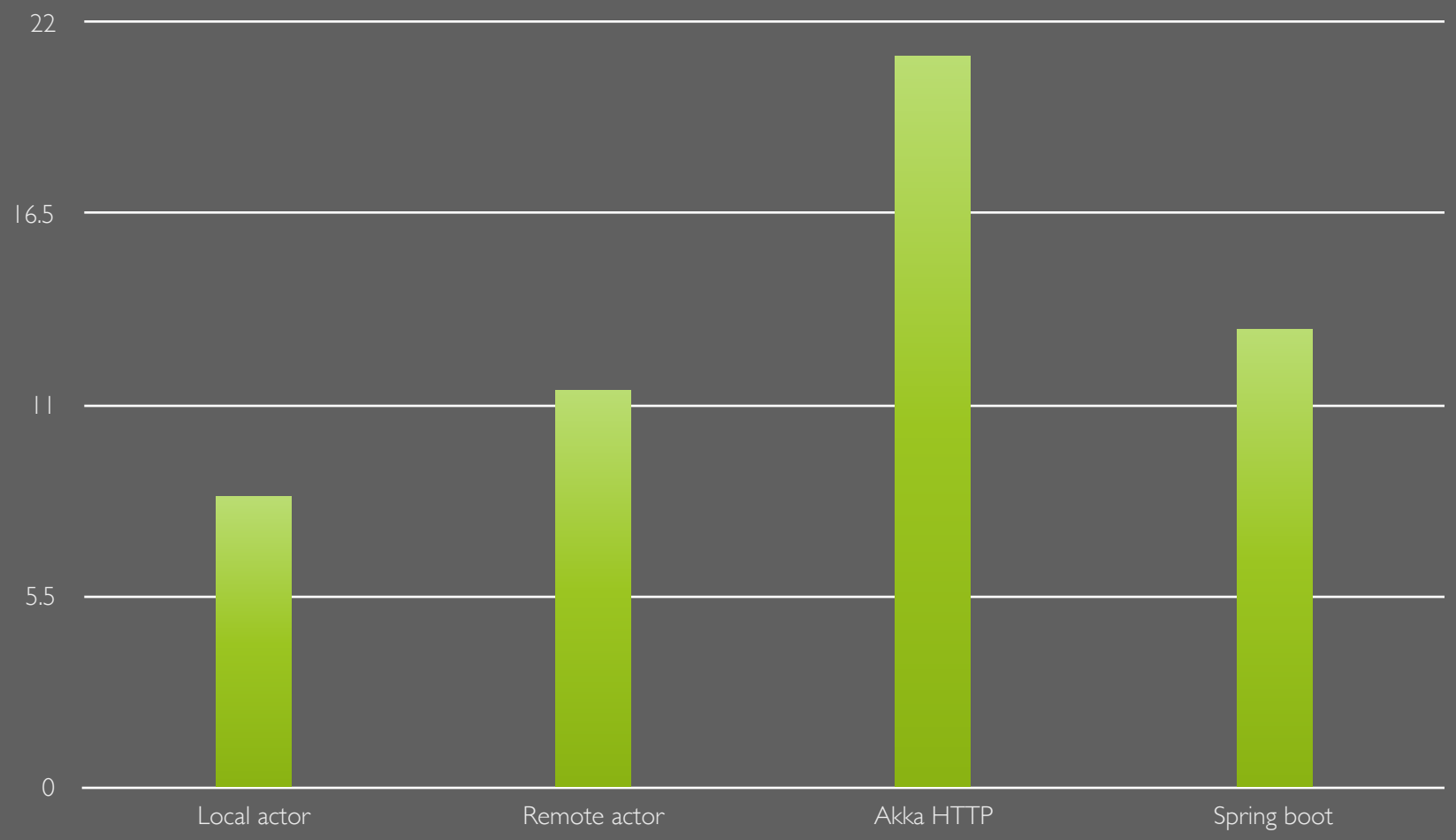
- No converting to JSON/SOAP
- More natural programming

- Concurrent on default
- Built-in load balancer
- Built-in circuit breaker

ADVANTAGES **HTTP**

- Indepedent of technology
- Loosely coupled

FAT JAR (SBT ASSEMBLY) IN MB



GATLING

```
class ExampleSimulation extends Simulation {
  val scn = scenario("My scenario").repeat(100) {
    exec(
      http("Ping")
        .get("http://localhost:8080/ping")
        .check(status.is(200))
    ).pause(100 millisecond)
  }

  setUp(scn.inject(
     rampUsers(1000) over (10 seconds) // Changing
  ))
}
```

PERFORMANCE TEST SETUP

HTTP: /ping

pong

JVM 1
Akka HTTP

HTTP: /pong

pong

JVM 2
Akka HTTP

HTTP: /ping

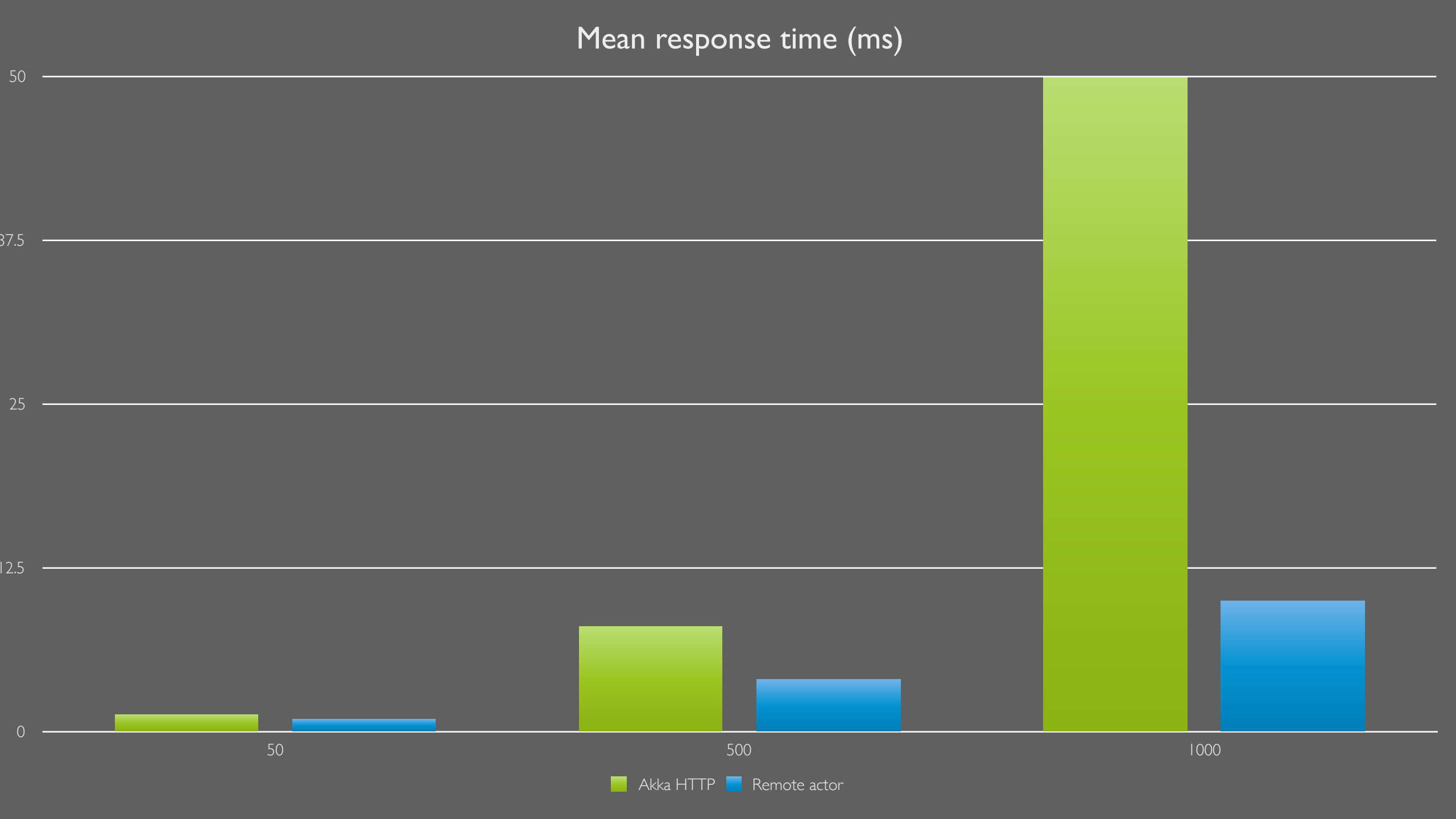
pong

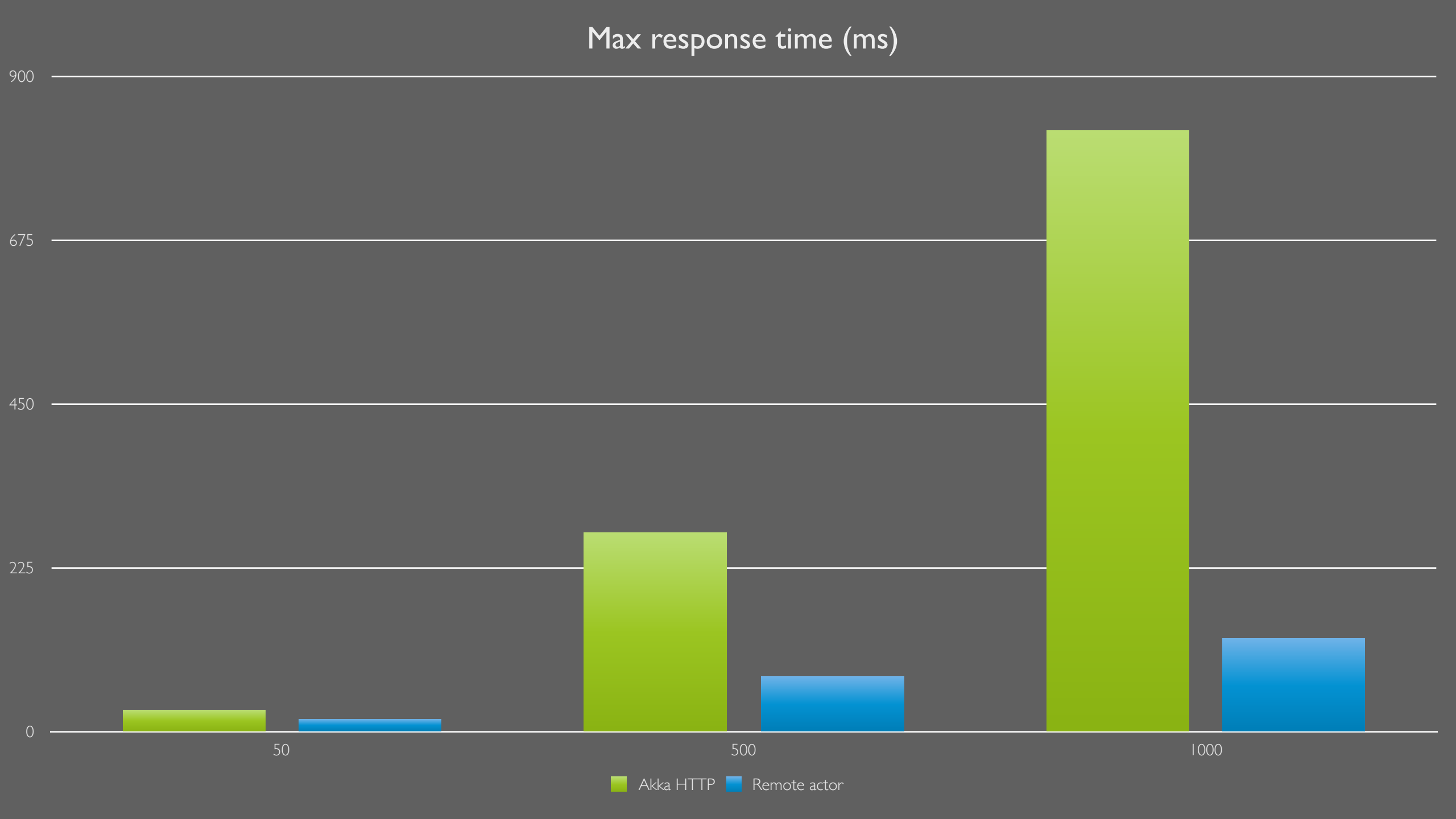
JVM 1
Akka HTTP

Akka over TCP: pong

pong

JVM 2
Akka remote actor


Mean response time (ms)
50
37.5
25
12.5
0
50
500
1000
Akka HTTP
Remote actor

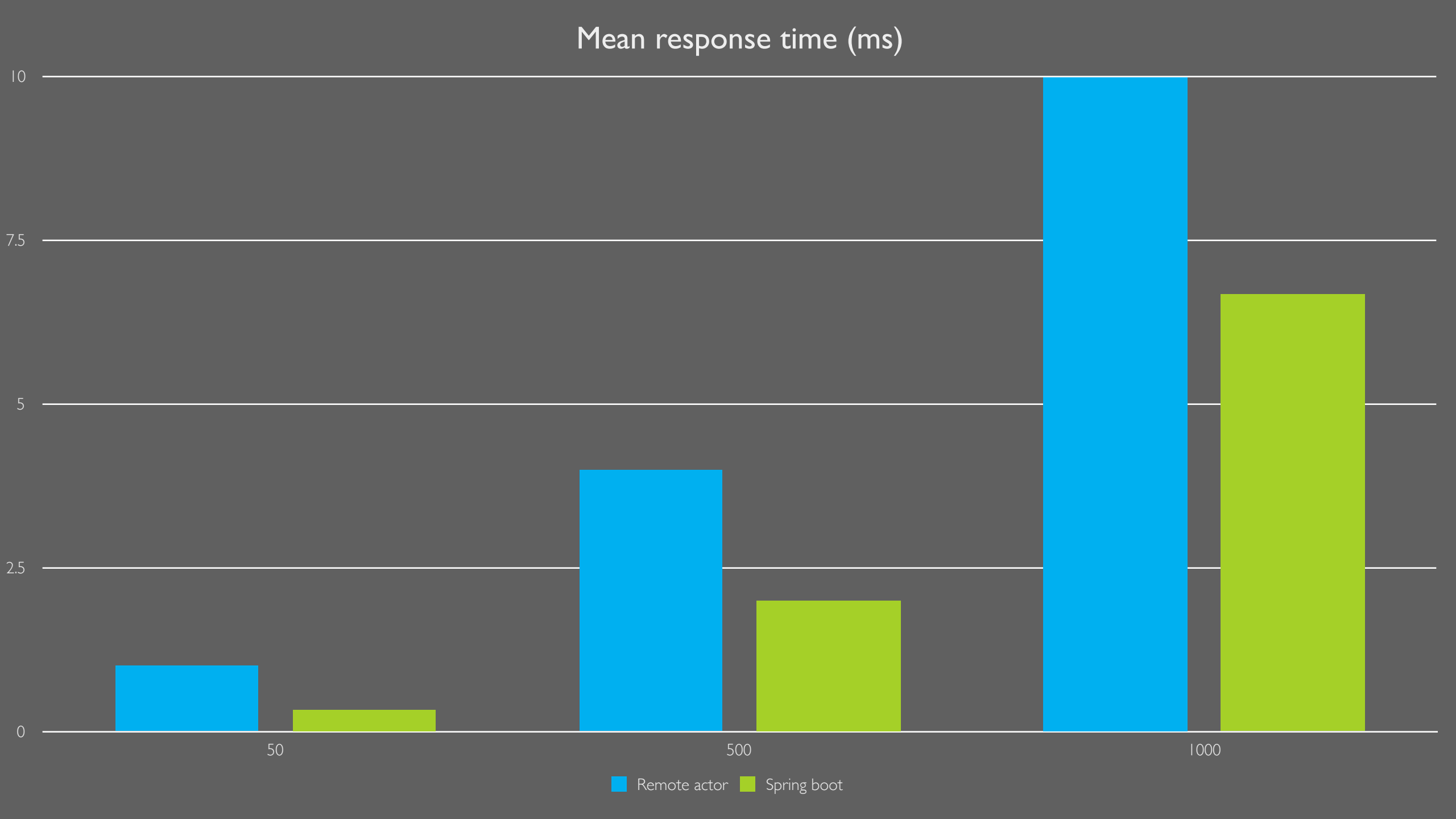

Max response time (ms)
900
675
450
225
0
50
500
1000
Akka HTTP
Remote actor

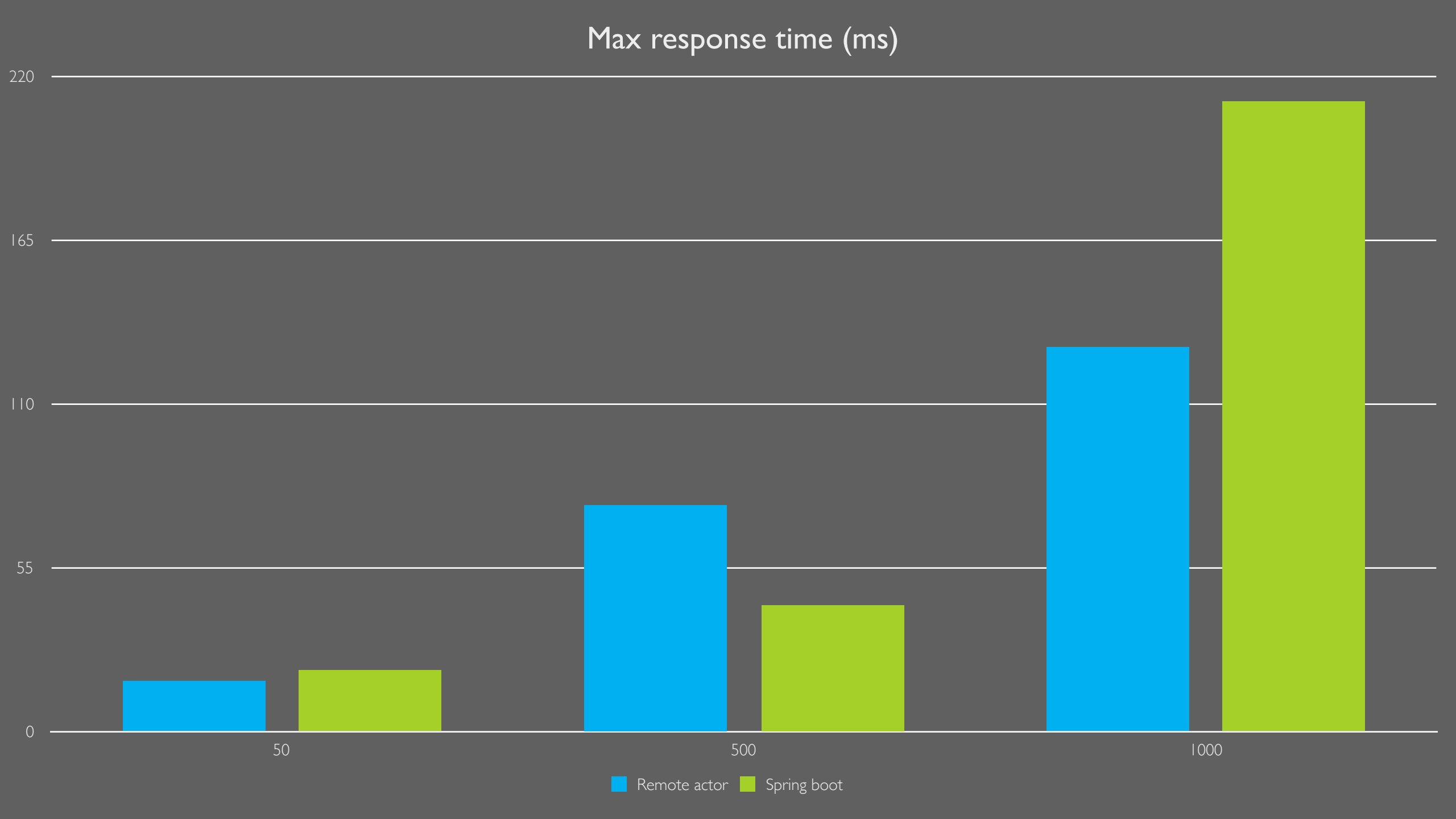
GRADUATION STUDENT

- REST could handle around 600 users
- Remote actors probably around 3300 users

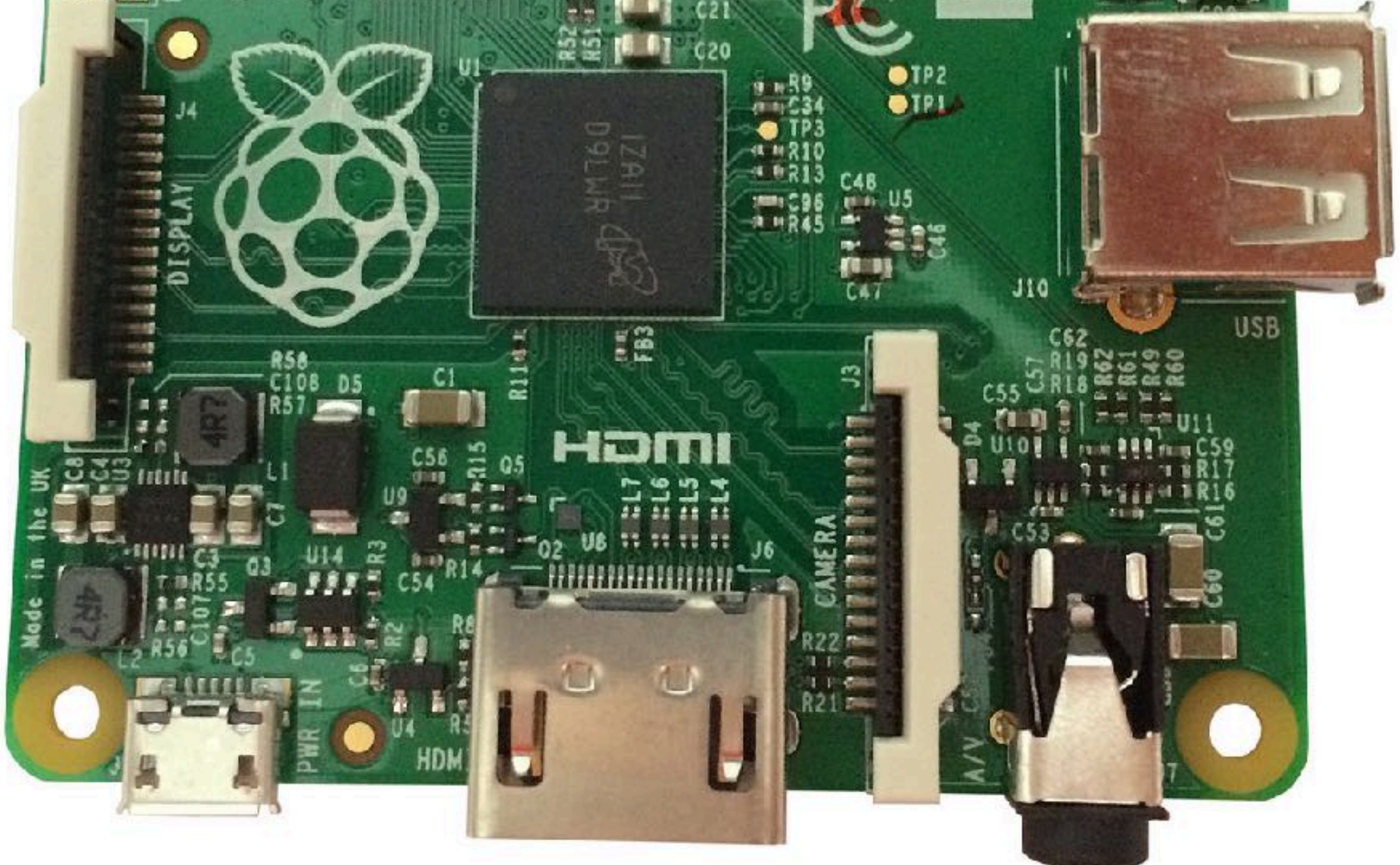
REST is dead,
long live remote actors!

- Johan Janssen


Mean response time (ms)
10
7.5
5
2.5
0
50
500
1000
Remote actor
Spring boot


Max response time (ms)
220
165
110
55
0
50
500
1000
Remote actor
Spring boot


Challenges


DISPLAY
HDMI
CAMERA
USB
PWR IN
A/V
Made in the UK
U1
J4
J3
J6
J10
TP1
TP2
TP3
U5
U9
U8
U10
U11
U14
Q2
Q3
Q5
D4
D5
FB3
L1
L2
L4
L5
L6
L7
4R7


MAR 2024
ACELL
MAR 2019


ing
is for


BARTEL


1806 km
KUTAISI
GEORGIA
8900 km
DAEGU
KOREA
1921 km
7800 km
GYUMRI
ARMENIA
403 km
LUOYANG
CHINA
КОШИЦЕ
СЛОВАКИЯ
1096 км
БЪРНО
ЧЕХИЯ
ПОЗНАН
ПОЛША
САНКТ-ПЕТЕРБУРГ
РУСИЯ
2435 км
МАКЕДОНИЯ
324 км


ekonomie
CVAL RYTÍRSKYCH KONÍ
D. HAMMETT VRAŽEDNÉ POVÍCK
FRANCOUZŠTINA
ERICH FROM

DO IT YOURSELF

- https://github.com/johanjanssen/
  - LCC
  - LCCInstallScript

CONCLUSION

Java

akka

The best part!!

QUESTIONS?

Johan Janssen, Info Support
@johanjanssen42
Johan.Janssen@infosupport.com