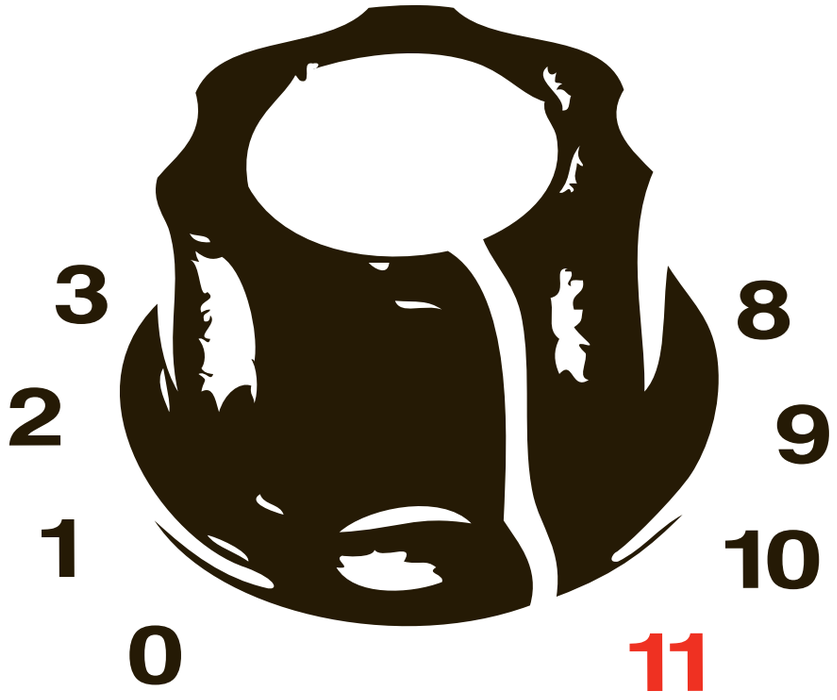
TorqueBox

Moc Javy – piękno Rubiego



Marek.


JBoss®
by Red Hat

BoxGrinder

JBoss AS 7

+

fedora

=

project:odd


©2011 Google - Imagery ©2011 TerraMetrics

No, do rzeczy panie!

Dziś w menu:

50 / 40 / 10

Czym jest TorqueBox?

Ruby i Java? Huh?

DO RZECZY!


3
8
2
9
1
10
0
11

Cel.

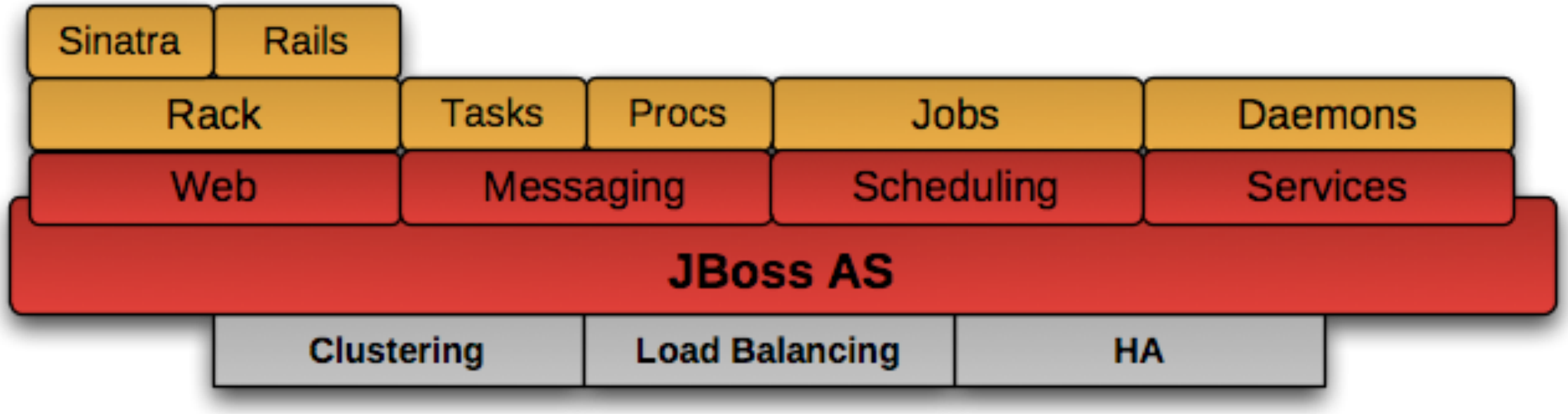
Web: Rails, Sinatra, Rack.

Ponadto: wiadomości, zadania, usługi, [...].

Brzmi znajomo?

Java już to ma,
wykorzystajmy to!

Randkowanie JBoss AS oraz Ruby.


Sinatra
Rails
Rack
Tasks
Procs
Jobs
Daemons
Web
Messaging
Scheduling
Services
JBoss AS
Clustering
Load Balancing
HA

Ale dlaczego Ruby?

Bob lubi Rubiego.

Dynamiczny.

Brak kompilacji.

Liberalny.

Zwięzły.

```
Set<Person> people = new HashSet<Person>();

for (Team each : teams) {
  people.addAll(each.getMembers());
}

for (Person each : people) {
  each.promote();
}
```

com/foo/Anything.java

```
teams.
   collect(&:members).
      flatten.uniq.each &:promote!
```

anything.rb

Wporzo, ale co z wydajnością?

JRuby FTW!

Bardzo szybkie środowisko.

Prawdziwe wątki.

Biblioteki Javy, za free.


3
8
2
9
1
10
0
11

TB 1.x – JBoss AS 6

TB 2.x – JBoss AS 7

Instalacja.

```
$ wget http://torquebox.org/builds/LATEST/torquebox-dist-bin.zip
$ unzip -q torquebox-dist-bin.zip

$ export TORQUEBOX_HOME=$PWD/torquebox-1*
$ export JBOSS_HOME=$TORQUEBOX_HOME/jboss
$ export JRUBY_HOME=$TORQUEBOX_HOME/jruby
$ export PATH=$JRUBY_HOME/bin:$PATH
```

Zbyt skomplikowane?

```
$ gem install torquebox-server --pre \
  --source http://torquebox.org/2x/builds/
LATEST/gem-repo/
```

```
$ torquebox run
```

Tak, na Windowsie również.

Zainstalowane, co dalej?

Aplikacja webowa, powiedzmy Rails.

```
$ rails new app -m \
  $TORQUEBOX_HOME/share/rails/template.rb
$ cd app
$ rake torquebox:deploy['/app']
```

http://localhost:8080/app/

To może jakiś kod co działa?

```
$ rails g scaffold Coffee \
  name:string description:text
$ rake db:migrate
```

http://localhost:8080/app/coffees/

Demo!

Nie takie rzeczy widziałem!

Integracja z Javą.

Ruby

```
class SomeController

  def index
    session[:password] = 'sw0rdfish'
  end

end
```

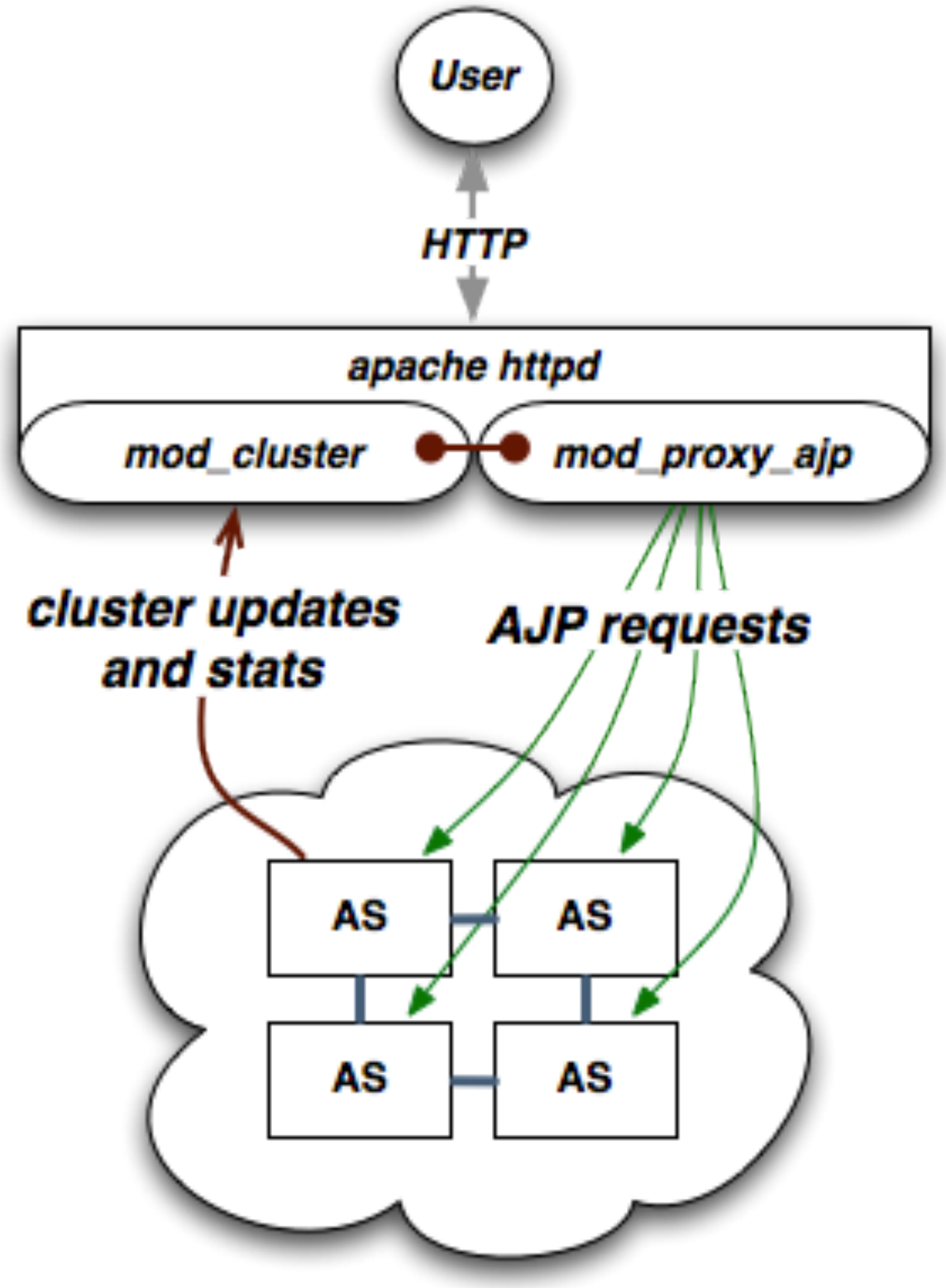
Java

```
public class SomeServlet {
  public void doGet(HttpServletRequest req,
                    HttpServletResponse resp) {
    request.getSession().getValue("password");
  }
}
```

Klastrowanie?

Sklastruj JBoss'a!

mod_cluster


User
HTTP
apache httpd
mod_cluster
mod_proxy_ajp
cluster updates
and stats
AJP requests
AS
AS
AS
AS

Dynamiczna konfiguracja!

Koniec o aplikacjach webowych.

Zaplanowane zadania.

```
class NewsletterSender
    def run
      subscriptions = Subscription.find(:all)
      subscriptions.each do |e|
        send_newsletter(e)
      end
    end
end
```

app/jobs/newsletter_sender.rb

```
jobs:
  monthly_newsletter:
description: first of month
job: NewsletterSender
cron: ‘0 0 0 1 * ?’
```

config/torquebox.yml

Messaging.

Zadania w tle.

```
class Something
  def foo
  end

  def bar
  end
end
```

```
something = Something.new
something.foo
something.bar
```

```
class Something
  include TorqueBox::Messaging::Backgroundable

  def foo
  end

  def bar
  end
end
```

```
something = Something.new
something.background.foo
something.bar
```

```
class Something
  include TorqueBox::Messaging::Backgroundable
  always_background :foo

  def foo
  end

  def bar
  end
end
```

```
something = Something.new
something.foo
something.bar
```

Kolejki.

```
queues:
  /queues/questions
  /queues/answers
topics:
  /topics/new_accounts
  /topics/notifications
```

config/torquebox.yml

```
messaging:
  /topics/print:
    PrintHandler:
      color: ‘red’
/queues/popular:
  AdultObserver:
    filter: "age >= 18"
    concurrency:  5
```

config/torquebox.yml

```
include TorqueBox

queue = Messaging::Queue.new '/queues/foo'
queue.create
...
queue.destroy
```

```
include TorqueBox::Messaging

class PrintHandler < MessageProcessor
  def initialize(opts)
    @color = opts['color']
  end
  def on_message(body)
    puts “#{body} in #{@color}”
  end
end
```

config/torquebox.yml

Usługi.

Nie-webowe, długie demony.

```
class MyService
  def initialize(opts={})
    @queue = Messaging::Queue.new(opts[:queue])
  end
  def start
    Thread.new { run }
  end
  def stop
    @done = true
  end
  def run
    until @done
      @queue.publish(Time.now)
      sleep(1)
    end
  end
end
```

app/processors/my_service.rb

Cacheowanie.

Wstrzykiwanie.

```
package com.mycorp;

@ApplicationScoped
class Something {
  @Inject
  private Else else;

  public void print(String what) {
    # blah
  }
}

@ApplicationScoped
class Else {

}
```

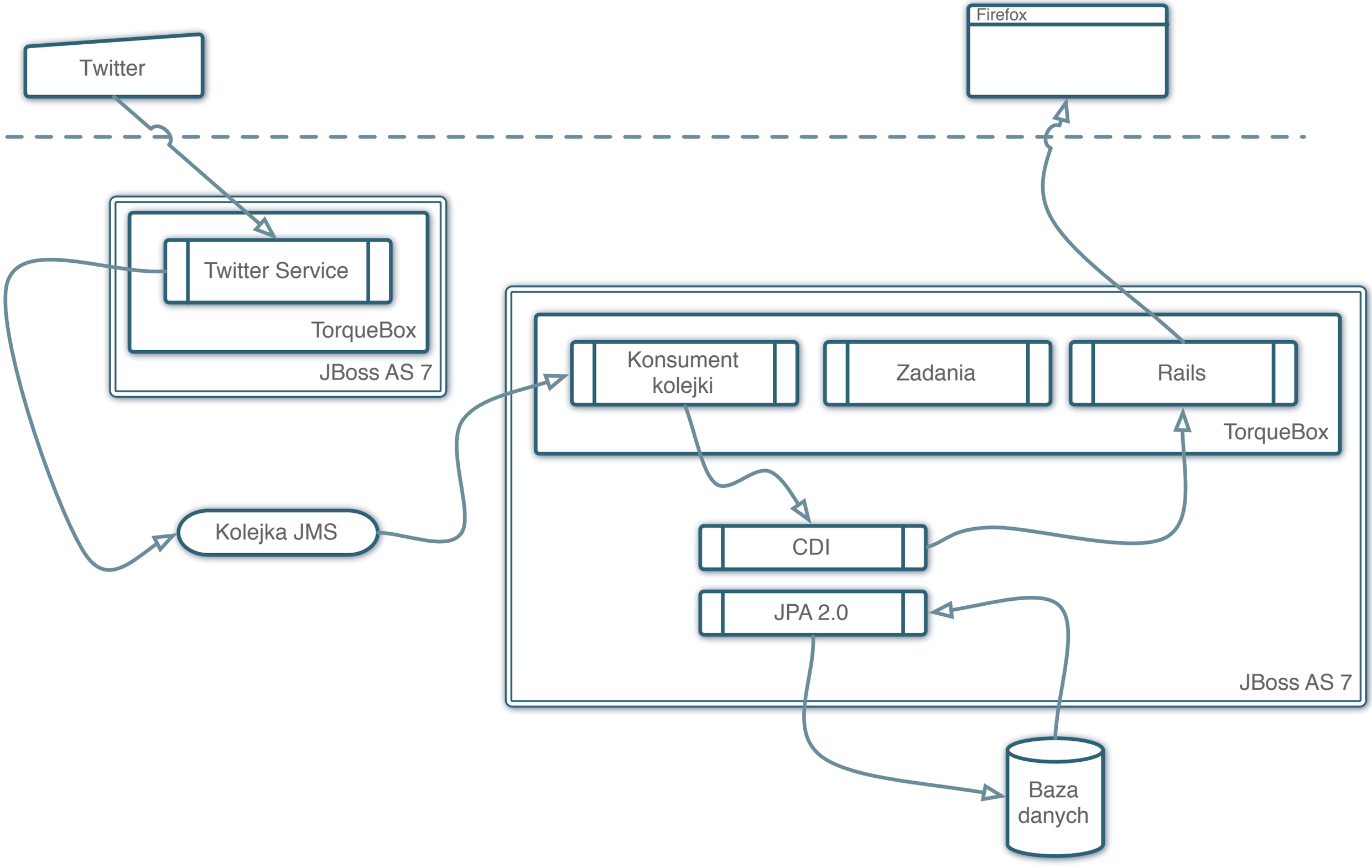
```
class MyService
  include TorqueBox::Injectors

  def initialize(opts={})
    @thing = inject(Java::com.mycorp.Something)
    @thing.print('Marek')
  end
end
```

CDI to tylko przykład: kolejki, tematy, heroina, inne rzeczy.

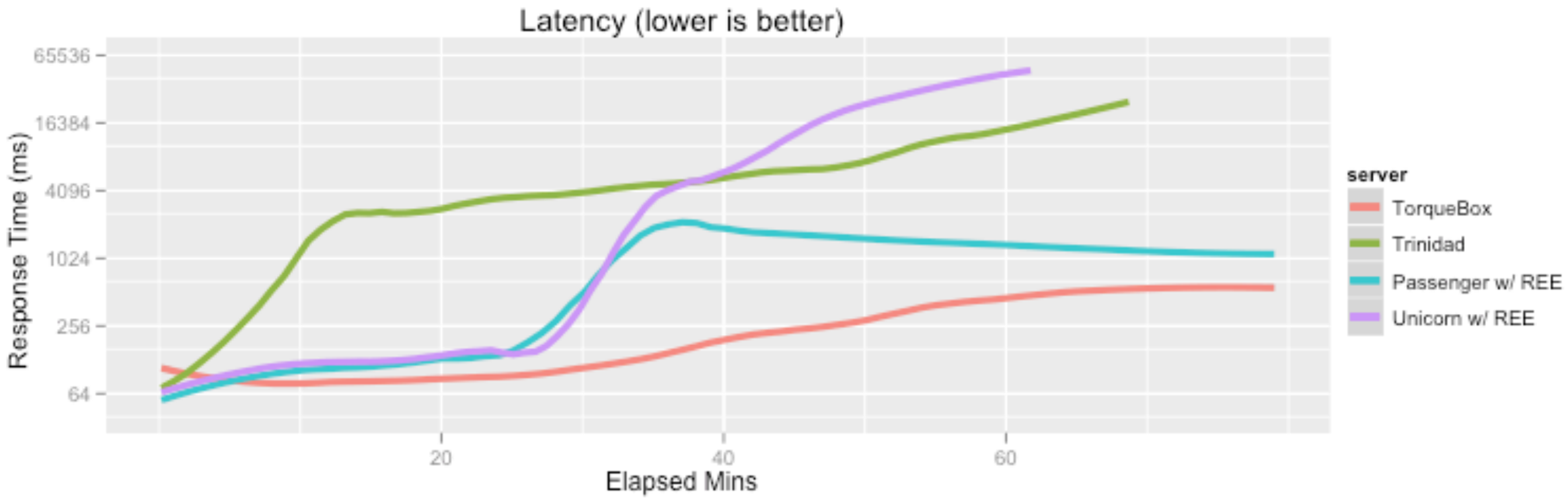
Drugs are bad, m’kay?

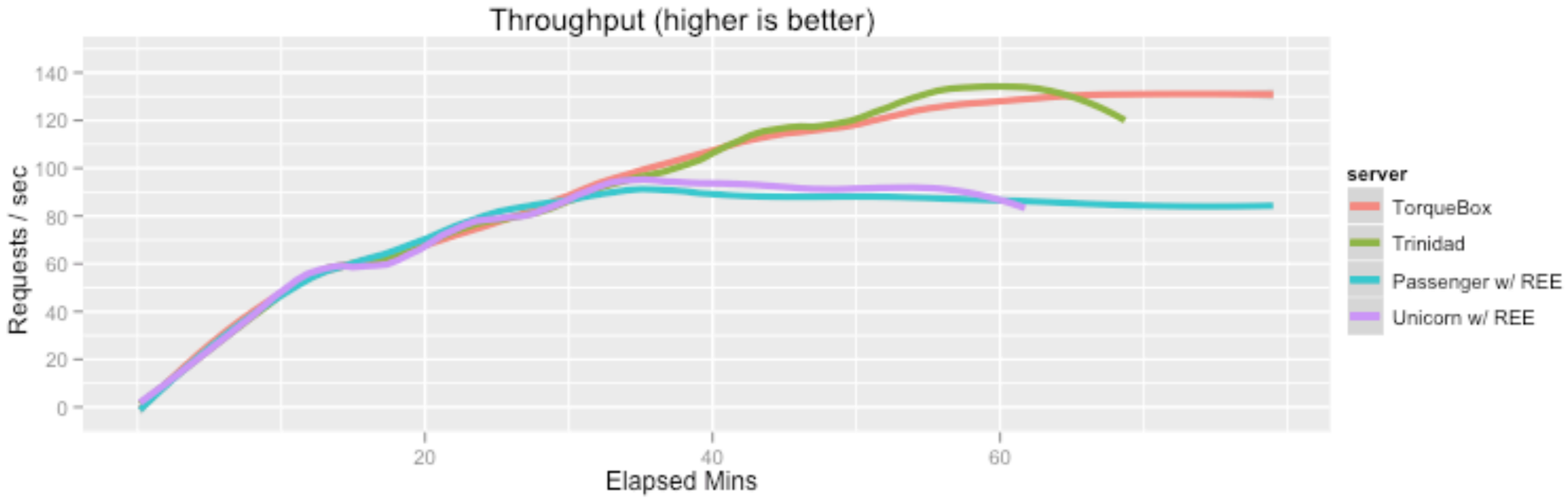
```
inject_topic( "/topics/questions" )
inject_queue( "/queues/answers" )
inject_namig( "java:comp/env/jdbc/myDS )
```

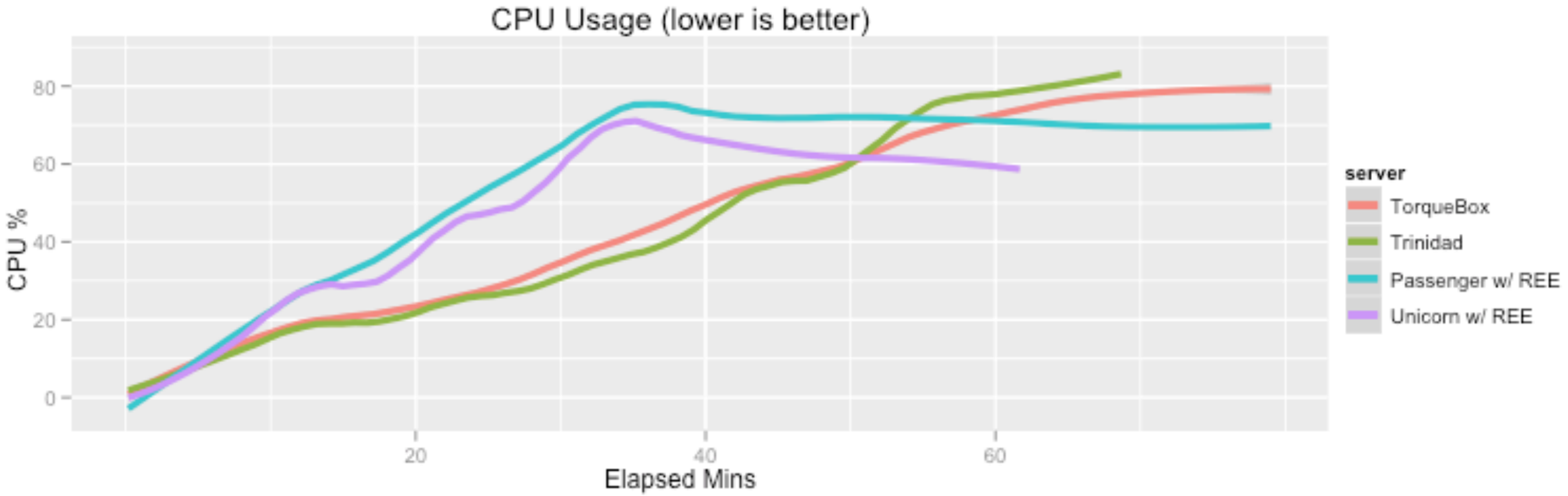
Demo!

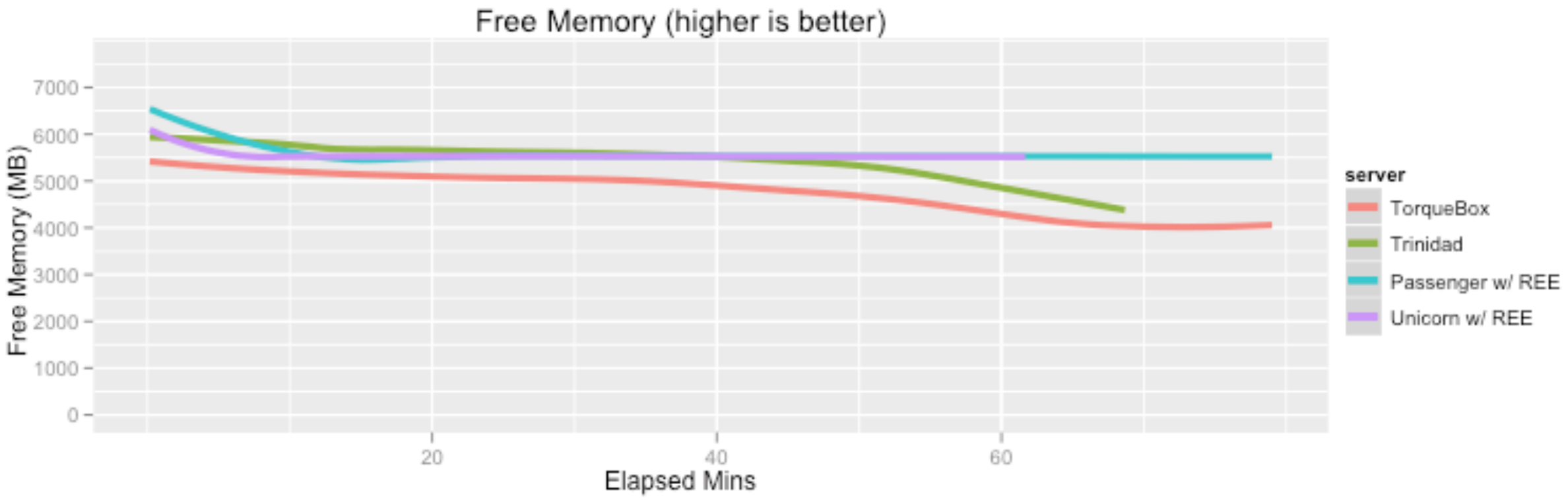

Twitter
Firefox
Twitter Service
TorqueBox
JBoss AS 7
Konsument kolejki
Zadania
Rails
TorqueBox
Kolejka JMS
CDI
JPA 2.0
JBoss AS 7
Baza danych

OK, fajnie, ale jak to jest wydajne?


Latency (lower is better)
Response Time (ms)
65536
16384
4096
1024
256
64
20
40
60
Elapsed Mins
server
TorqueBox
Trinidad
Passenger w/ REE
Unicorn w/ REE


Throughput (higher is better)
Requests / sec
140
120
100
80
60
40
20
0
20
40
60
Elapsed Mins
server
TorqueBox
Trinidad
Passenger w/ REE
Unicorn w/ REE


CPU Usage (lower is better)
CPU %
0
20
40
60
80
20
40
60
Elapsed Mins
server
TorqueBox
Trinidad
Passenger w/ REE
Unicorn w/ REE

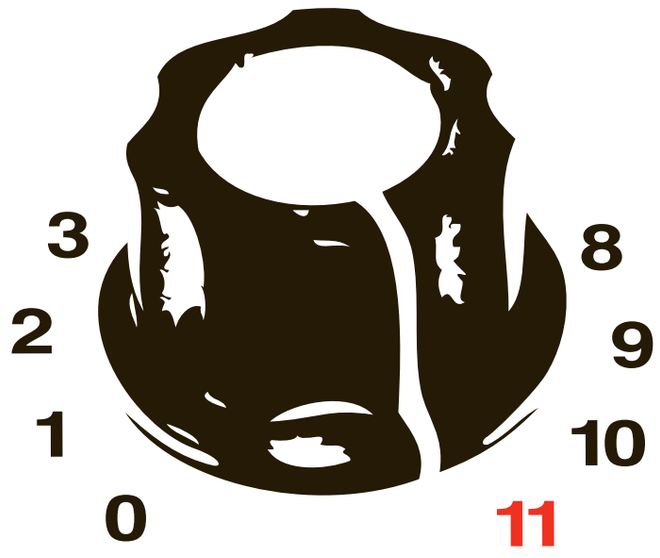

Free Memory (higher is better)
Free Memory (MB)
0
1000
2000
3000
4000
5000
6000
7000
20
40
60
Elapsed Mins
server
TorqueBox
Trinidad
Passenger w/ REE
Unicorn w/ REE

http://torquebox.org/
http://github.com/torquebox

http://github.com/goldmann/confitura-2011-torquebox-demo

IRC: #torquebox on FreeNode

@torquebox



Dzięki.

Nie zapomnij o naklejkach!