A RESTy-JSON API with spray.io and maven

Ignasi Marimon-Clos i Sunyol
@ignasi35
#scalabcn

cc BY NC SA

sponsored by

special appearance

thanks!

special thank you's

JPradel (@agile_jordi): review and organization

JCasadellà (@jcasadella): review

JRudolph for simplifying step 4

https://github.com/jrudolph/scala-meetup-spray/commit/c4991938ec7b26b183d701aac73949e7822cf45e

about me

n. /iŋ'nazi/

1) problem solver, Garbage Collector, mostly java, learning scala, some agile

2) kayaker

3) under construction

4) wears glasses

what we saw

- BDD (cucumber)
- TDD (scalatest)
- sbt (inifinite run)
- intelliJ
- JAX-RS
- Jetty

https://github.com/funcio/bdd-jaxrs-hands-on

what we'll see

- BDD (cucumber)
- TDD (scalatest)
- sbt (inifinite run)
- intelliJ
- JAX-RS
- Jetty

- BDD (jbehave?)
- TDD (scalatest)
- mvn (infinite run)
- eclipse (worksheet)
- spray-can/https/...
- spray-json

https://github.com/ignasi35/scala-meetup-spray

why? (1/2)

- I'm comfy with eclipse
- I'm comfy with maven (I know that's weird)
- *(no more) Typesafe delivers a prepacked env*
  - *still there s/where*
- eclipse-maven integration is solid
- I needed infinite-test in maven at work
- I like trolling Jordi and Jose

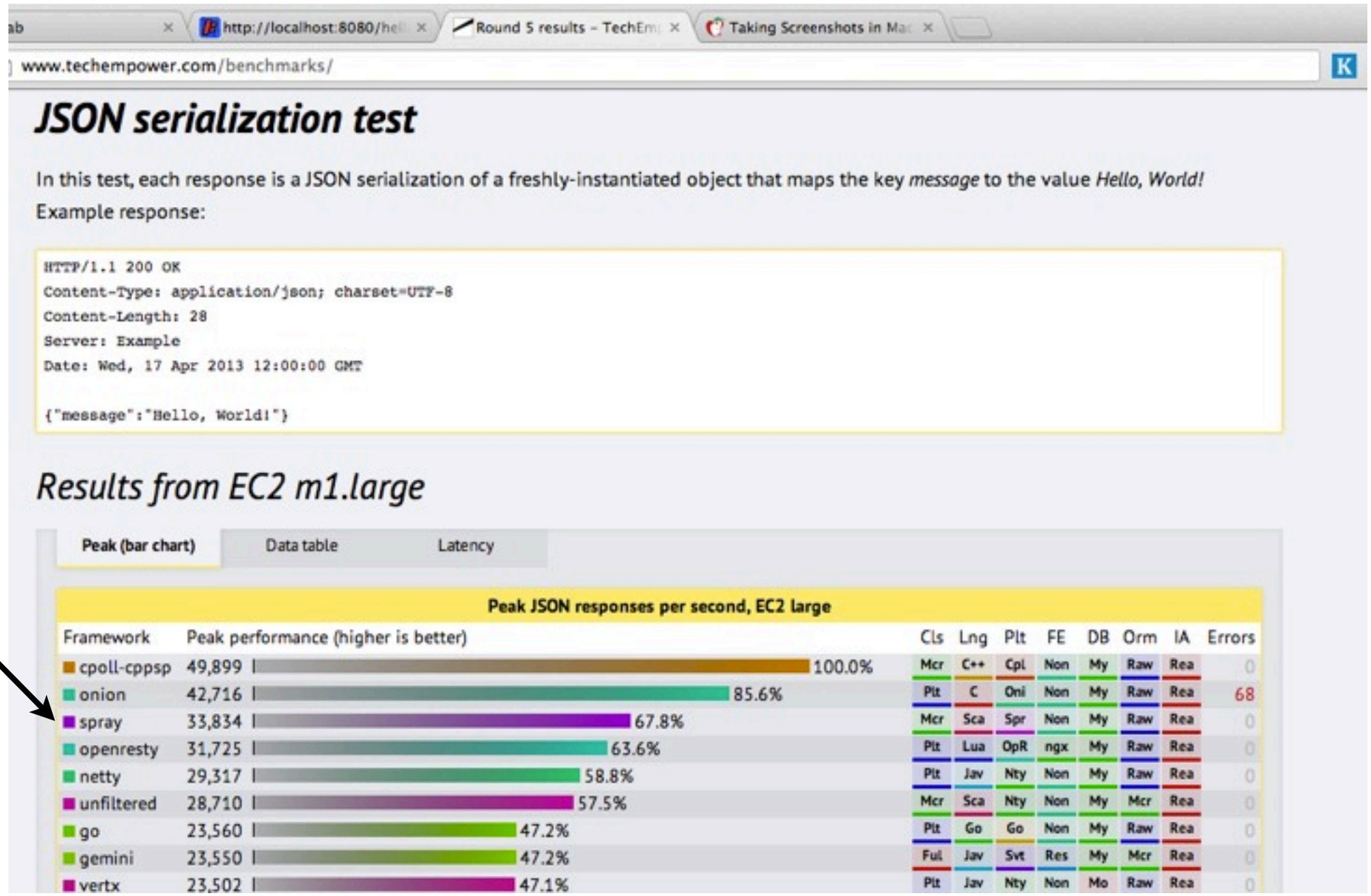
why? (2/2)

- spray.io has akka
  - spray 1.x has akka 2.x
- spray.io is modularized (that's useful)
- I don't need java API (@see play)
- spray-json is f*&[ing simple (via shapeless)

http://spray.io/project-info/current-versions/

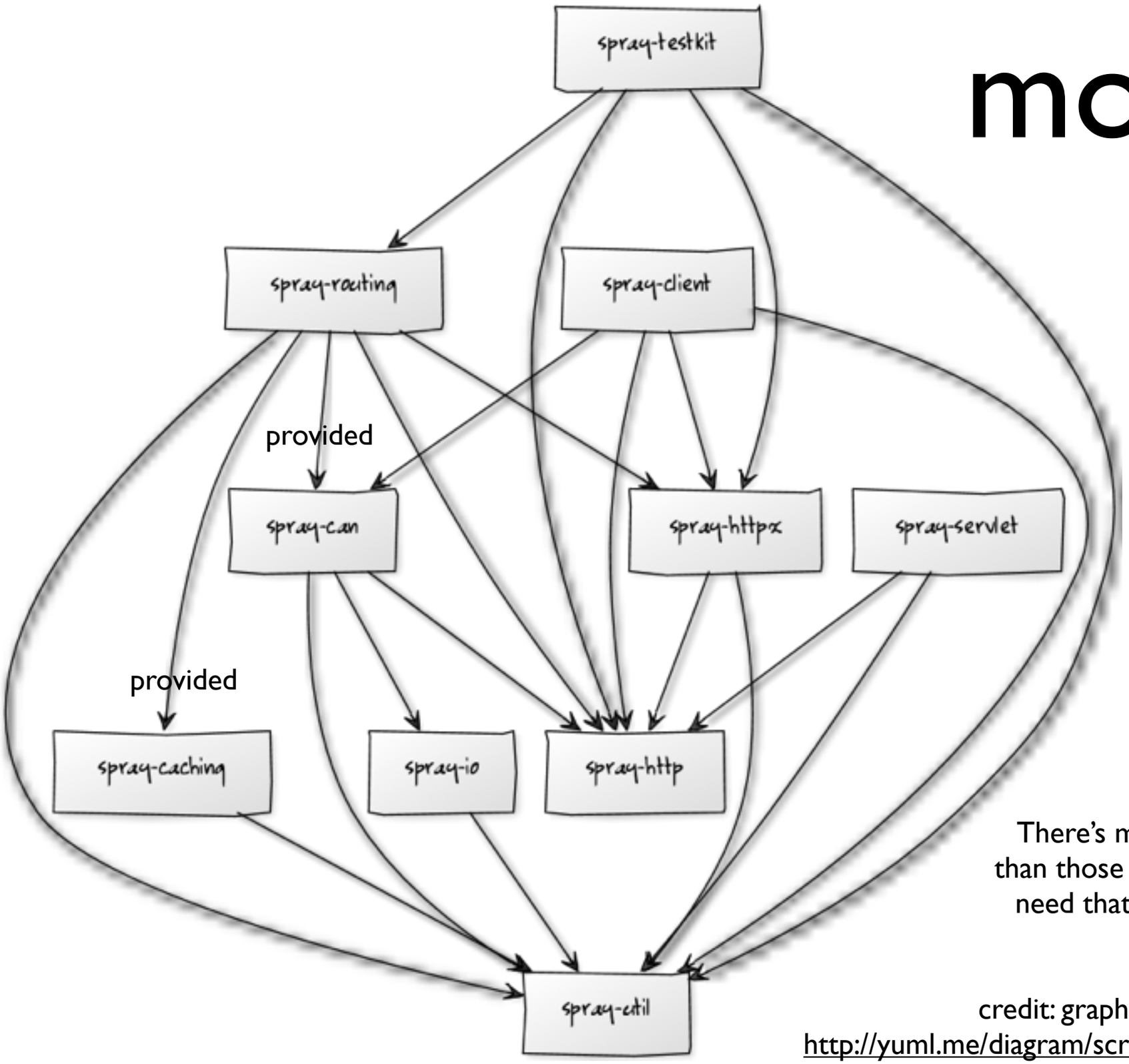
why? (bonus)

- spray.io beats all other JVM stacks/fw (*)



http://www.techempower.com/benchmarks/

spray modules



There's more provided dep's
than those labelled but we don't
need that level of detail now.

credit: graphic made using yuml.me
http://yuml.me/diagram/scruffy/class/[A]->[B],[A]->[C],[B]->[C]

spray

- One thing done right
- spray-routing => simple entry point.

- @sirthias (Mathias Doenitz) in Akka 2.2
- magnet pattern
  - alternate to method overload
- IOBridge, directives, Marshallers, ...???

SHOWTIME®

step 1

- Given an HTTP server
- When I GET /hello
- Then I obtain “hello”

step 1 (tips)

- Edit a web.xml
- Create a mapping
- Provide a BeanConnectionBeanFactory
- Add a servers.xml
- Invoke the Olds Gods and the New

specially dedicated to @ch4lit

step 2

- Given an HTTP server
- When I GET /hello.json
- Then I obtain “{msg:‘world’}”

directives + complete

- path/pathPrefix/pathTest
- authenticate
- get/post/delete/...
- decodeRequest/encodeResponse/...
- cache/alwaysCache/...

- complete

http://spray.io/documentation/spray-routing/predefined-directives-alphabetically/

step 3

- Given an HTTP server
- When I GET /books.json
- Then I obtain the complete books stock

step 4

- Given an HTTP server
- And I POST a book.txt in /books
- When I GET /books.json
- Then I obtain the updated books stock

MEGA-KUDOS to
https://github.com/jrudolph

step 5

- Given an HTTP server
- And a programmatic HTTP client
- When I GET /hello.json
- Then I obtain “{msg:‘world’}”

conclusions

- scala in maven is tough
  - mvn plugin lacks infini-run, infini-install,...
- scala worksheet rocks
- spray allows a quick entry point
- docs and examples still to be improved
  - because I'm very dumb
- ...

thanks!