CADEC 2015 - REACTIVE TUTORIAL

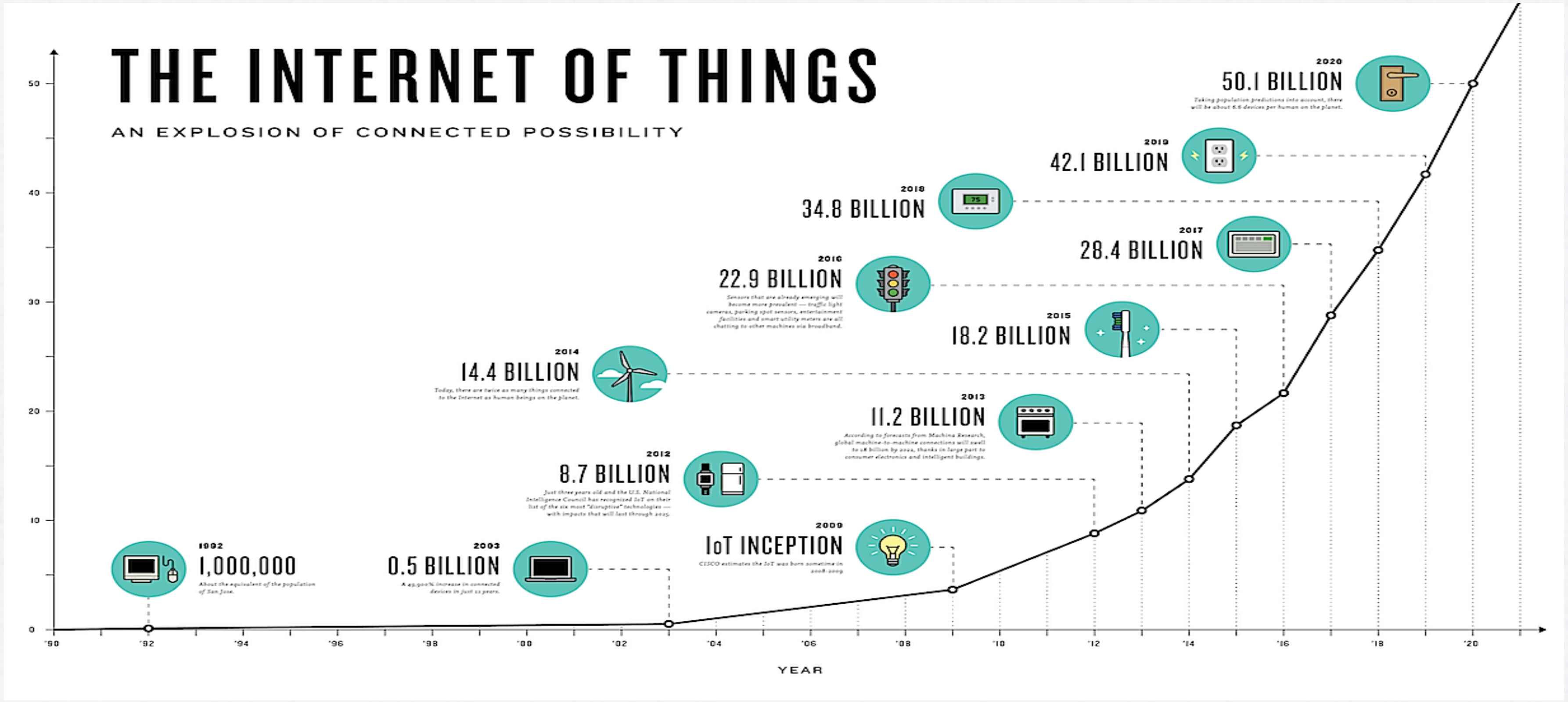
Non-blocking I/O and Reactive frameworks for scalable and resilient services

MAGNUS LARSSON, MATS EKHAMMAR

2015-01-28 | CALLISTAENTERPRISE.SE

CALLISTA
— ENTERPRISE —

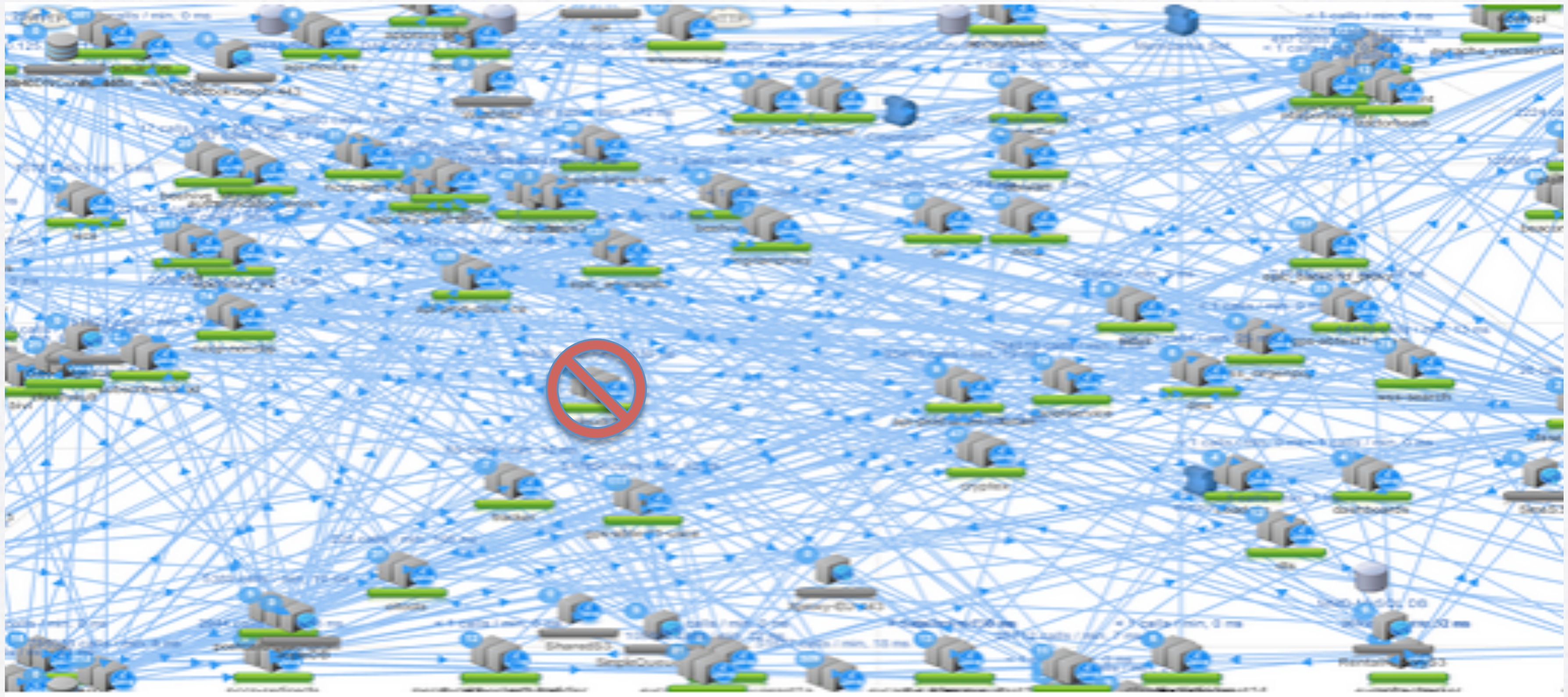
THE SCALABILITY CHALLENGE…



Source: http://www.theconnectivist.com/2014/05/infographic-the-growth-of-the-internet-of-things/

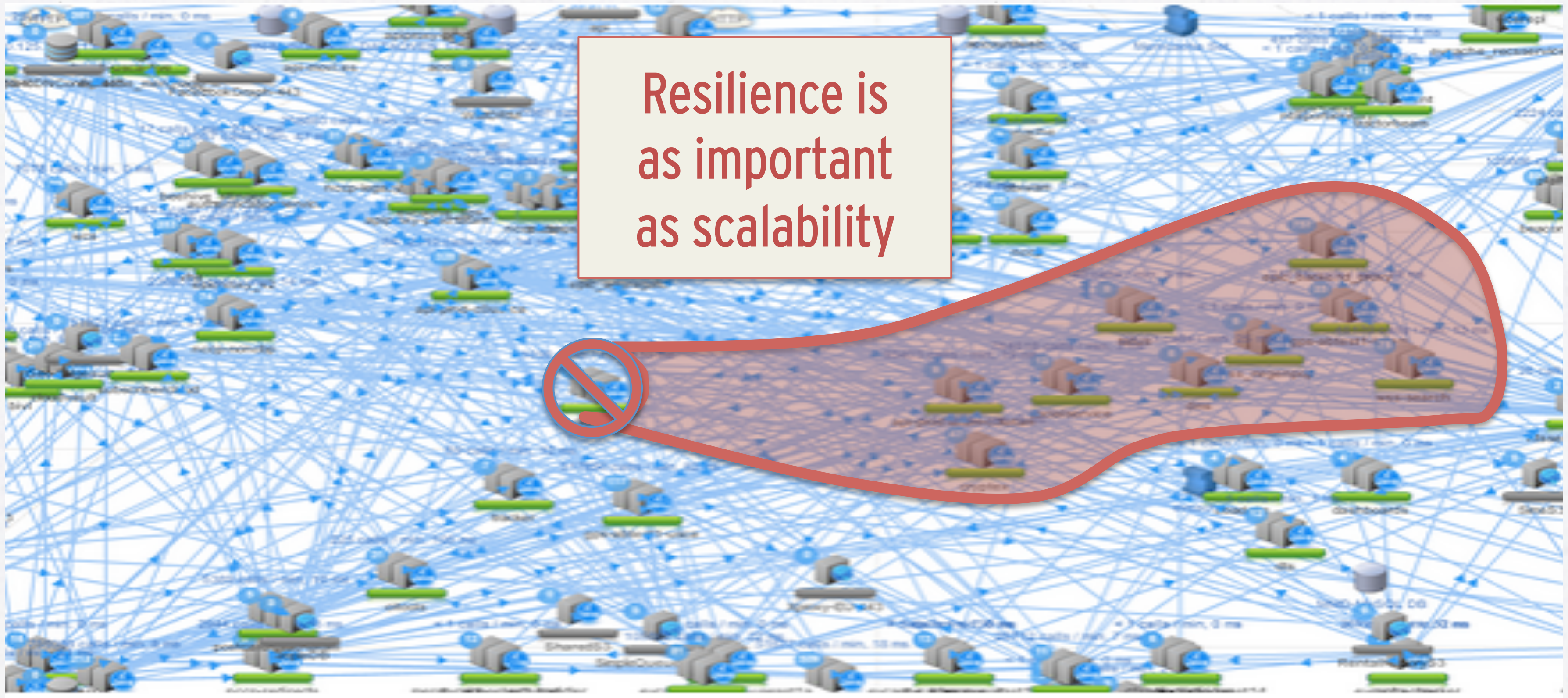
CALLISTA
— ENTERPRISE —

…SERVICES FAILS…



Source: http://techblog.netflix.com/2013/01/announcing-ribbon-tying-netflix-mid.html

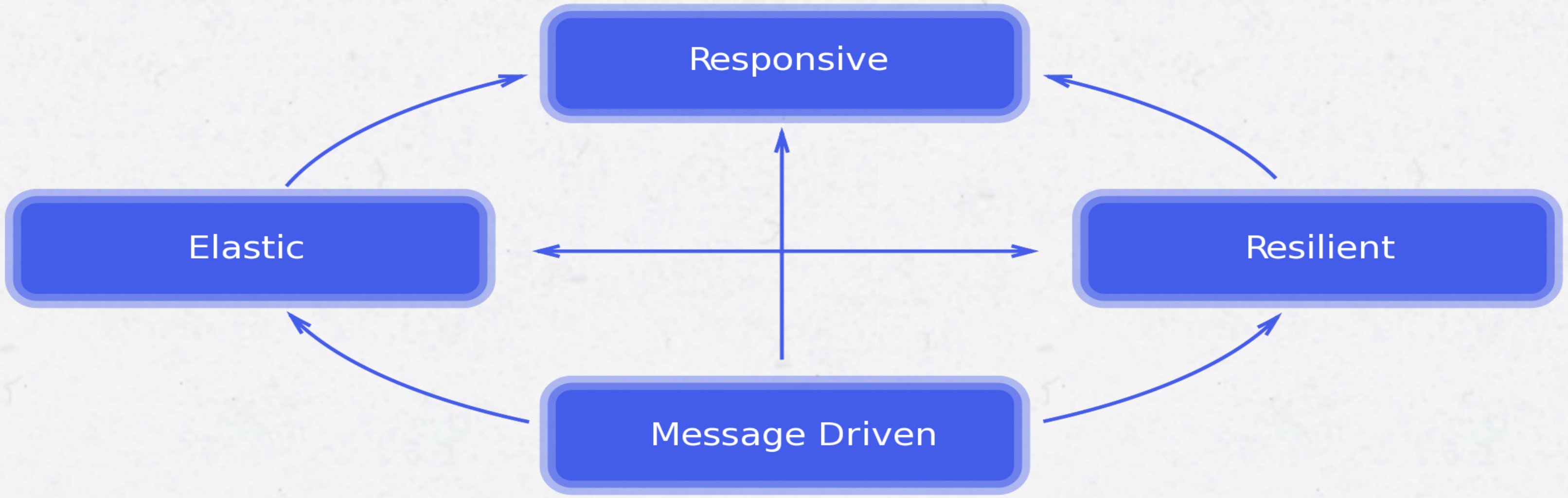
WATCH OUT FOR THE DOMINO EFFECT!



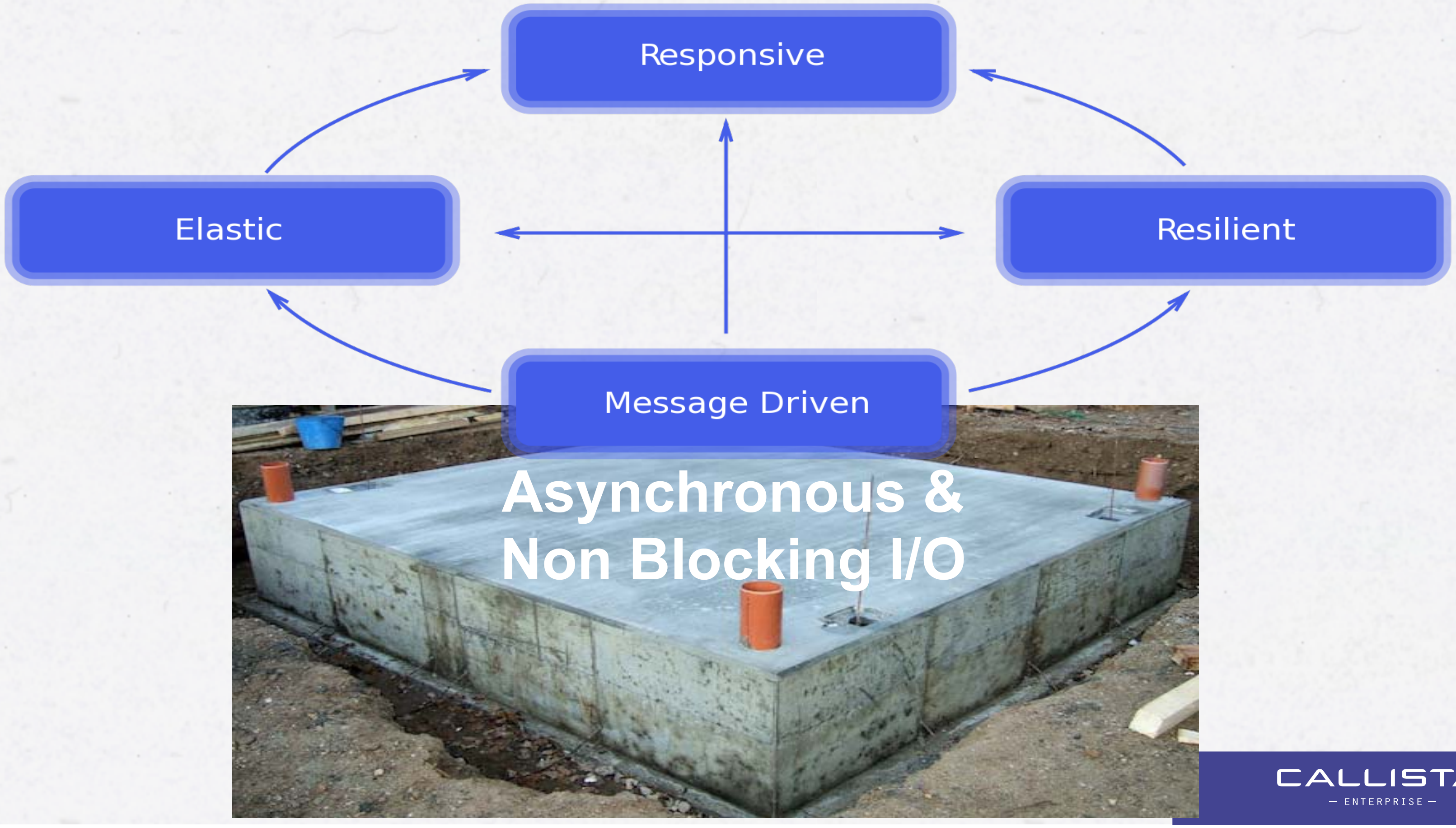
Source: http://techblog.netflix.com/2013/01/announcing-ribbon-tying-netflix-mid.html

THE REACTIVE MANIFESTO

- http://www.reactivemanifesto.org

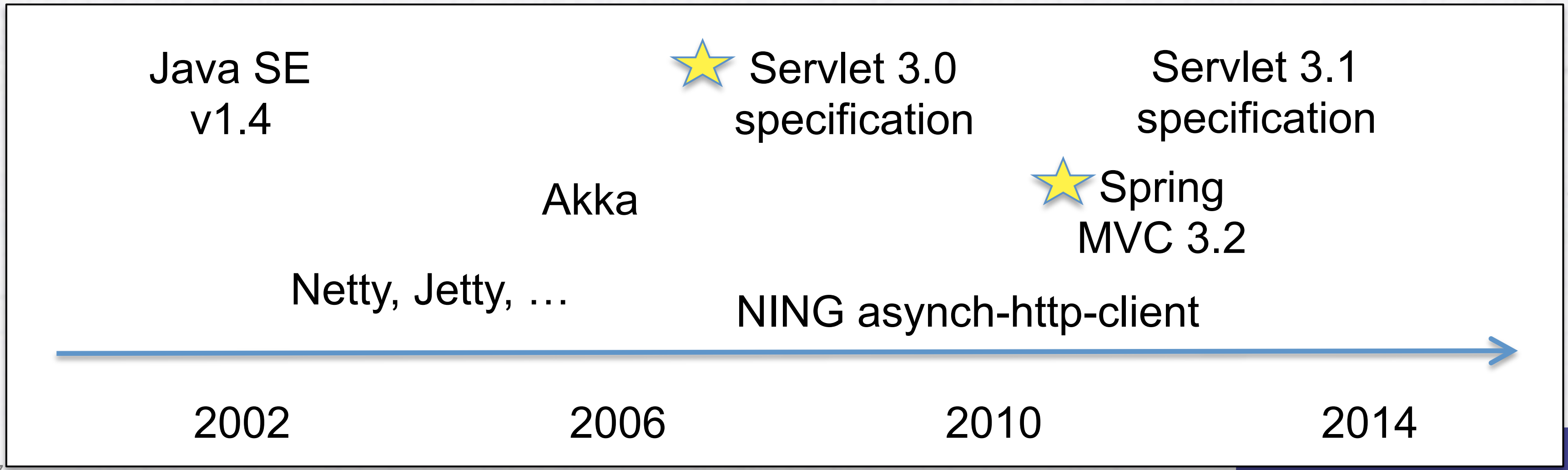


THE REACTIVE MANIFESTO

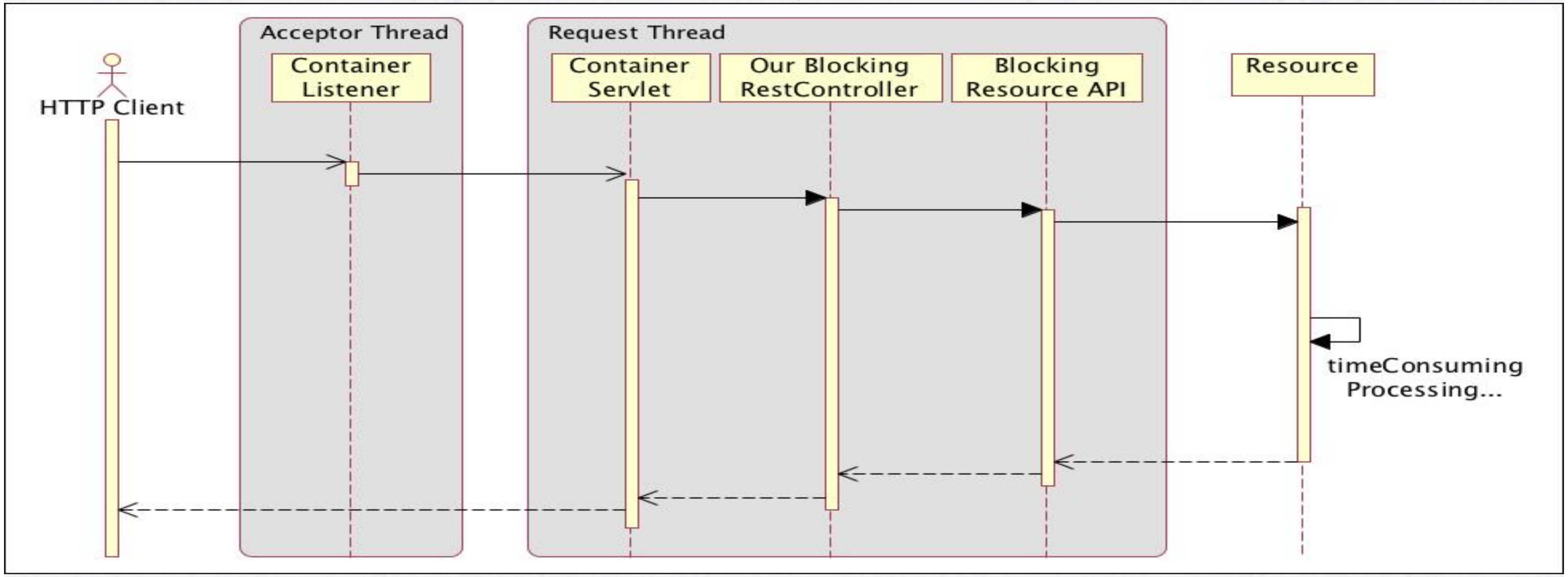


IS NON-BLOCKING I/O NEW?

- No!!!
- A short history lesson…
  - Supported in operating systems “for ever”
  - In Java SE since 2002
  - But it took some time to get mature, e.g. portable and easy to use…

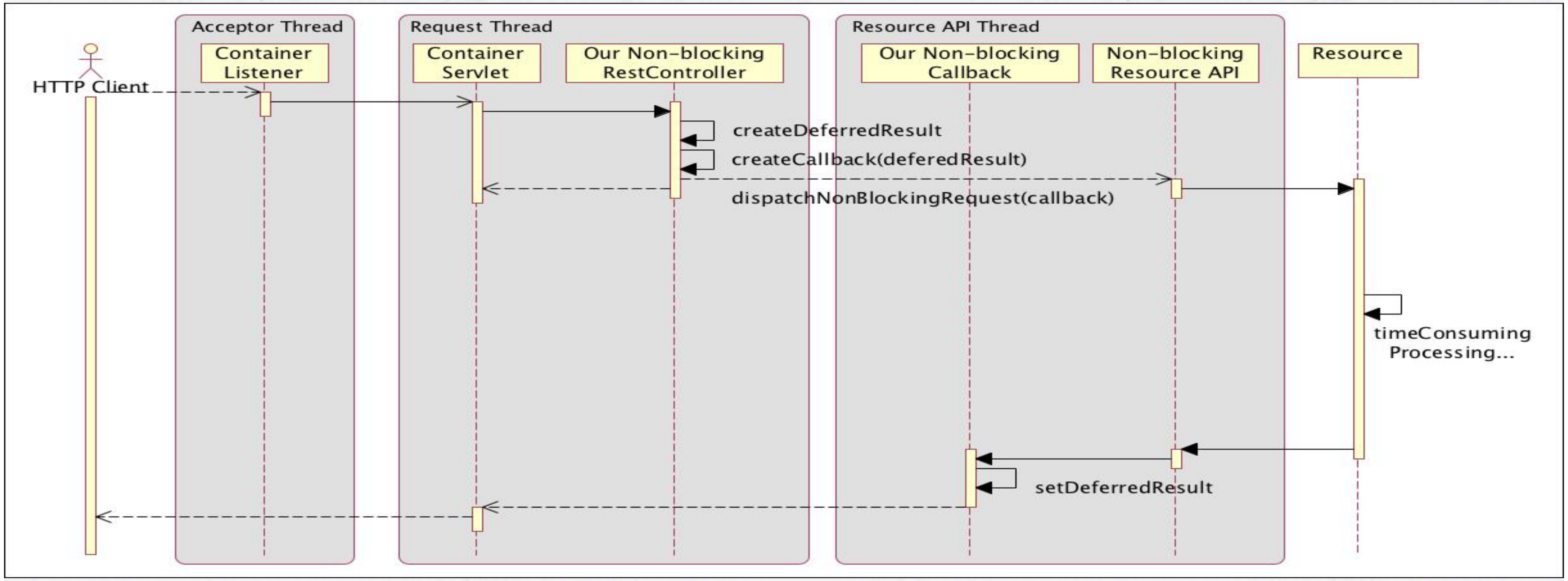


TRADITIONAL BLOCKING I/O



CALLISTA — ENTERPRISE —

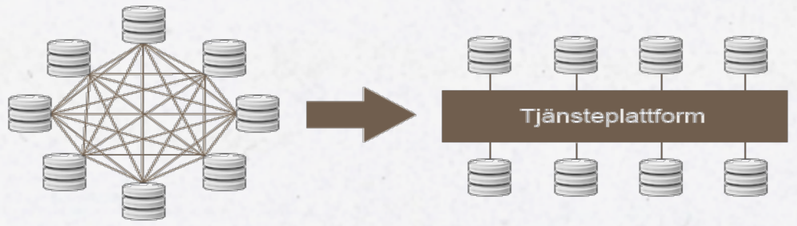
NON-BLOCKING I/O

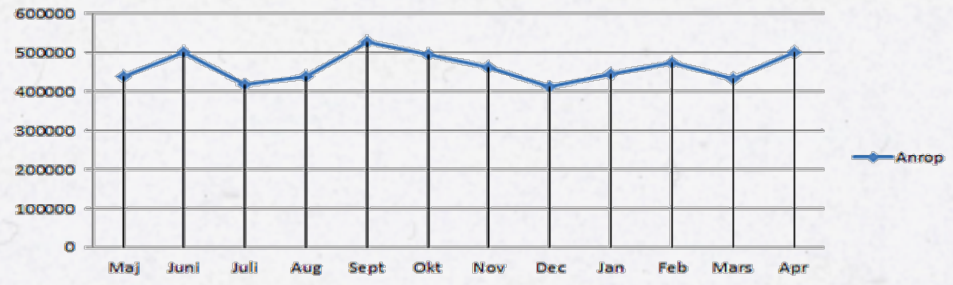

Acceptor Thread
Request Thread
Resource API Thread
HTTP Client
Container Listener
Container Servlet
Our Non-blocking RestController
Our Non-blocking Callback
Non-blocking Resource API
Resource
createDeferredResult
createCallback(deferedResult)
dispatchNonBlockingRequest(callback)
timeConsuming Processing...
setDeferredResult

Demonstration

AN EXAMPLE OF POTENTIAL PROBLEMS WITH BLOCKING I/O

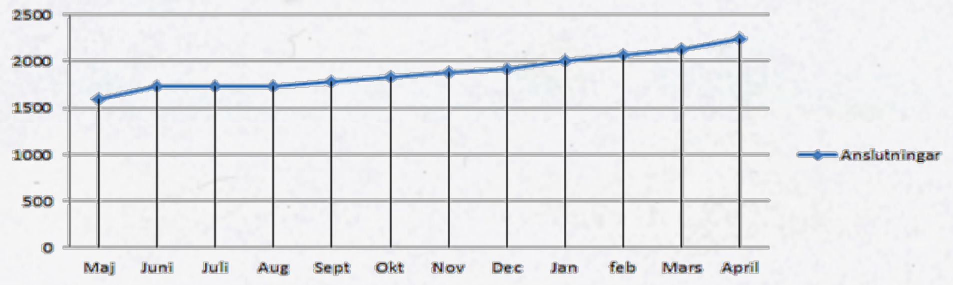
National Healthcare Service Platform





Totalt antal verksamheter anslutna till domäner

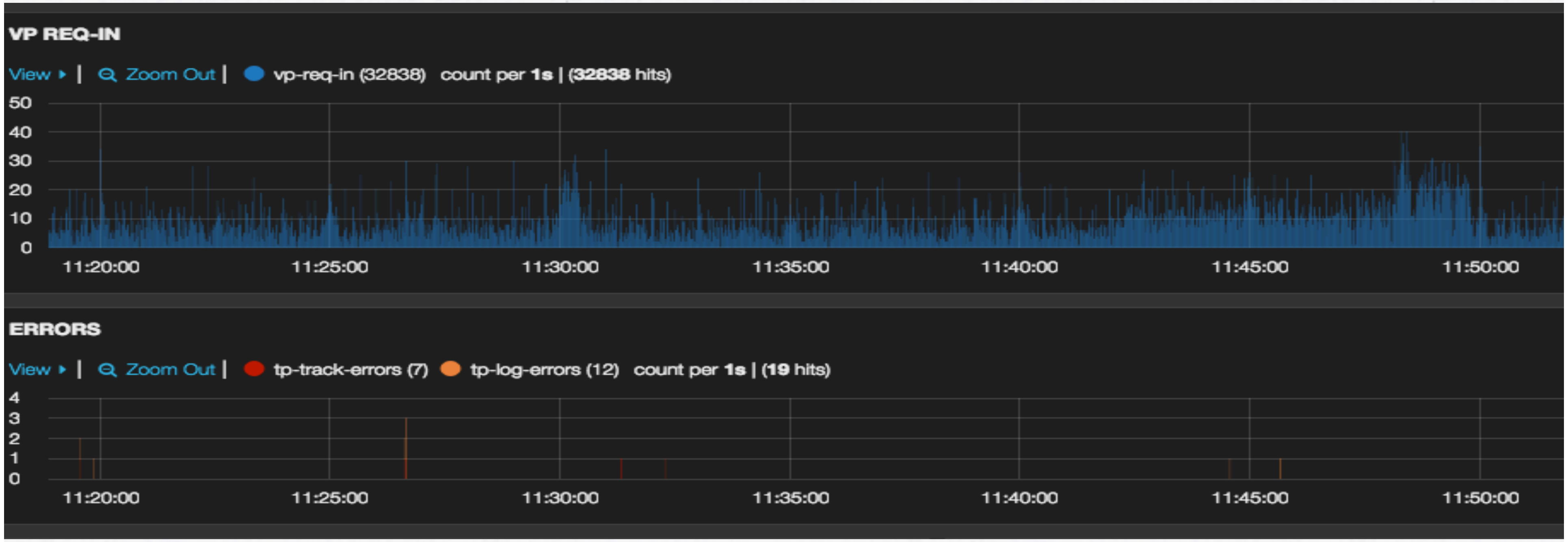
I diagrammet nedan kan du följa utvecklingen av hur många verksamheter som anslutit till Tjänsteplattformen.



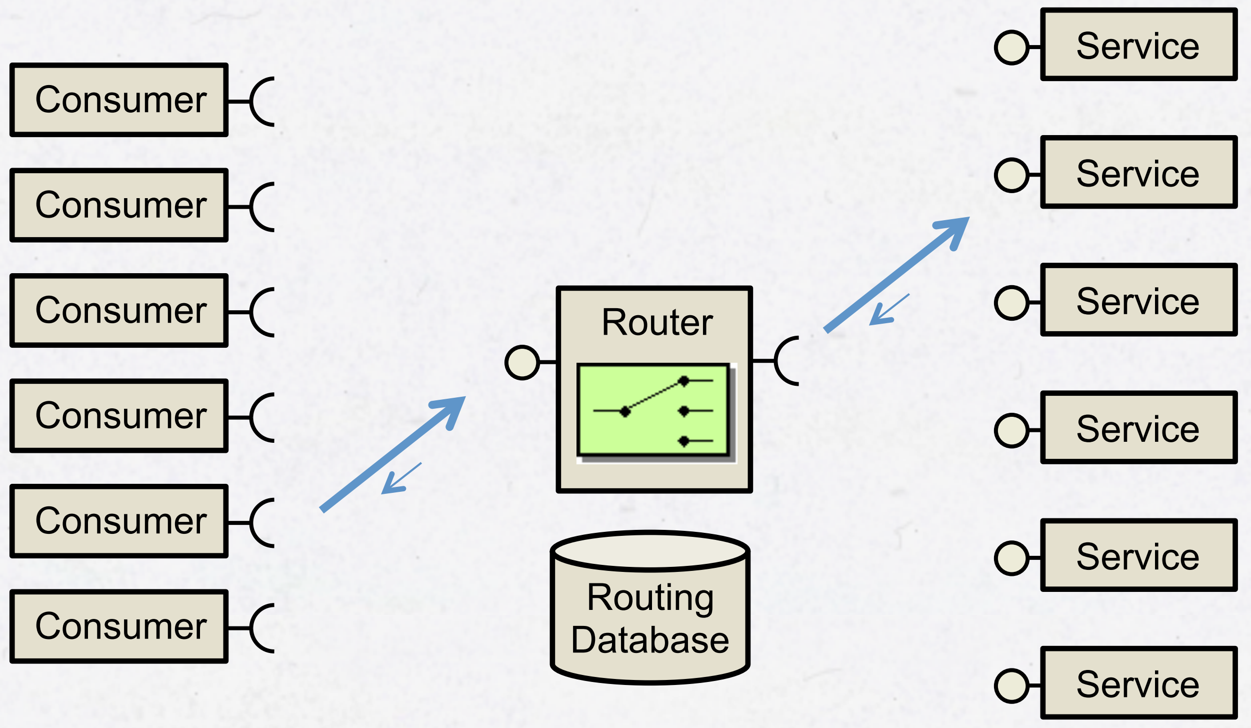
- National reference architecture
- Standardized protocols
- Standardized message formats
- Service catalog for routing
- In operation since 2010
  - > 2000 connected care units
  - > 500 000 messages/day (8h)

CALLISTA
— ENTERPRISE —

VIEW FROM THE RUNNING SYSTEM IN PRODUCTION

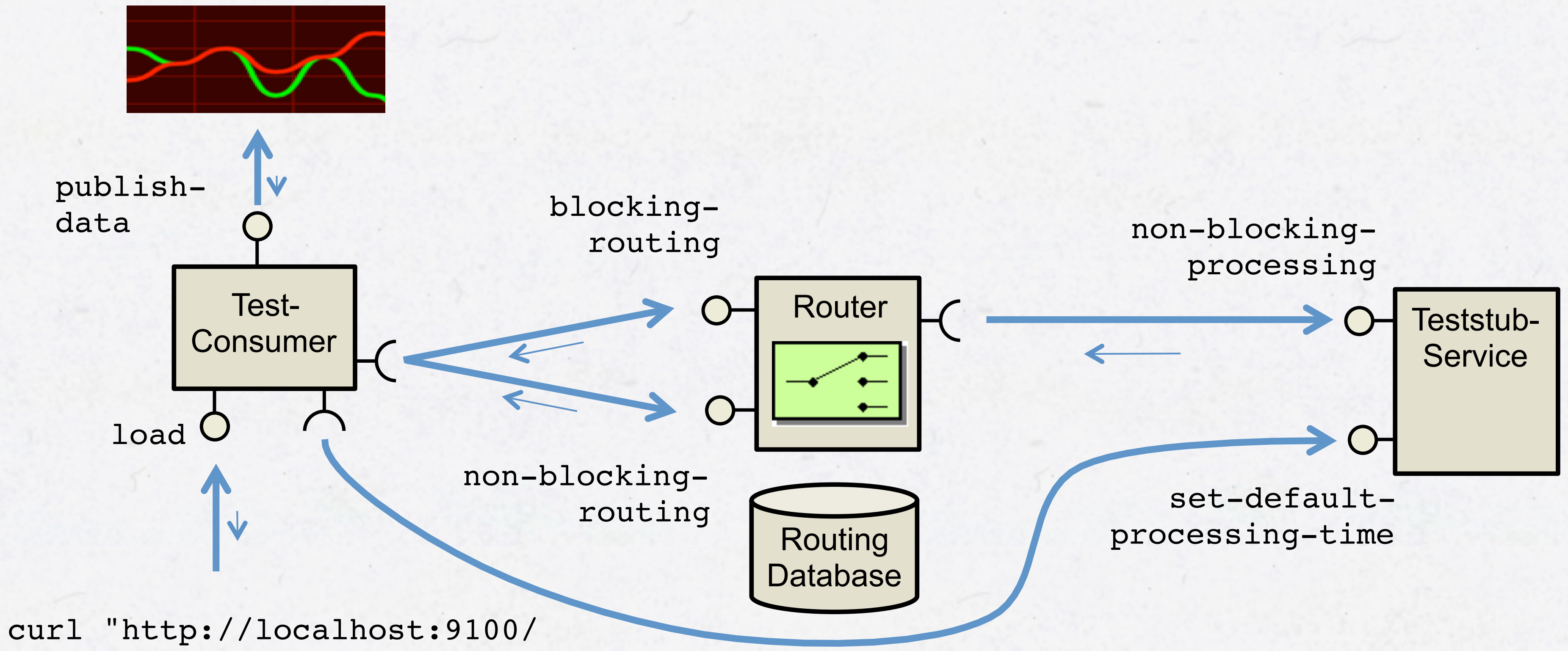


HIGH LEVEL ARCHITECTURE...



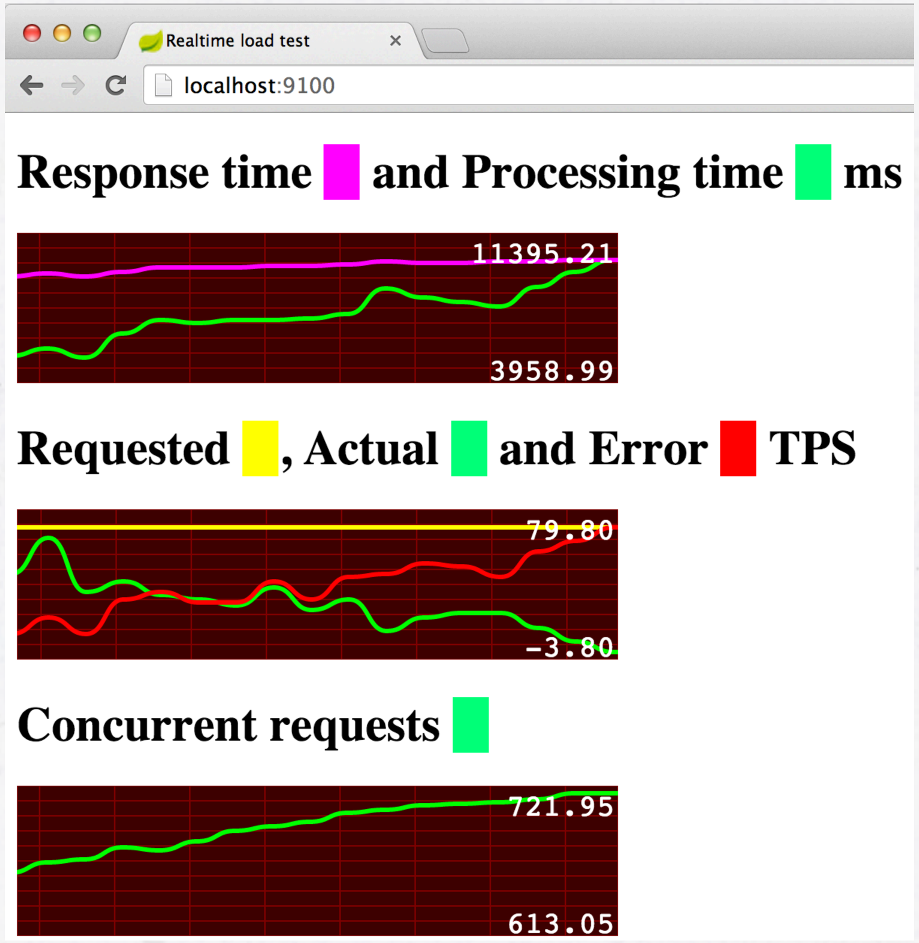
CALLISTA
— ENTERPRISE —

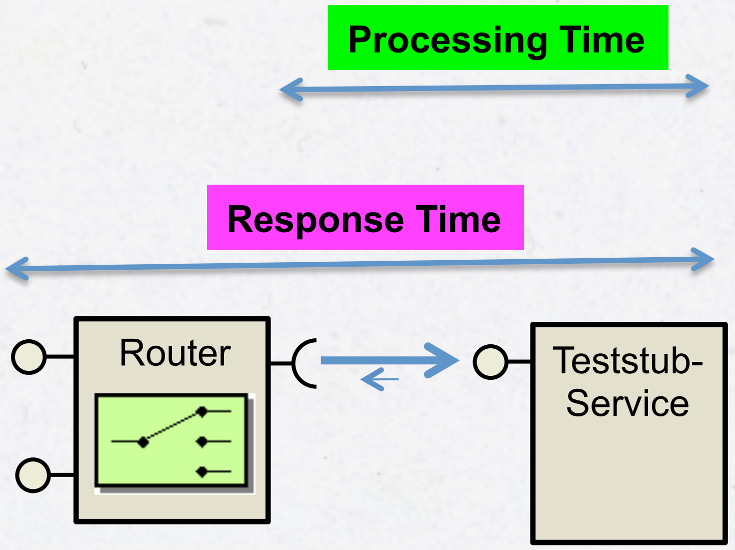
SIMULATION OF THE ENVIRONMENT



```
curl "http://localhost:9100/
   load?minMs=3000&maxMs=6000&
   test=2&tps=50"
```

SAMPLE OUTPUT FROM A LOAD TEST





DEMO

- Normal load is
  - 20 – 50 reqs/s
  - Service Provider response times: 3-6 s
  - Default request timeout: 10 s
- Start with 20 reqs/s and step up to 50 reqs/s
- If ok
  - Add a increase of load, 65 reqs/s
  - Add a minor problem, increase response times by 1s
  - What happens? Why?
- Switch to non blocking I/O and **go unleashed**!!!