Event driven architecture with Apache Spark and Spring Reactor

Vedran Krtalić i Zvonko Žibrat, APIS IT d.o.o

Sadržaj

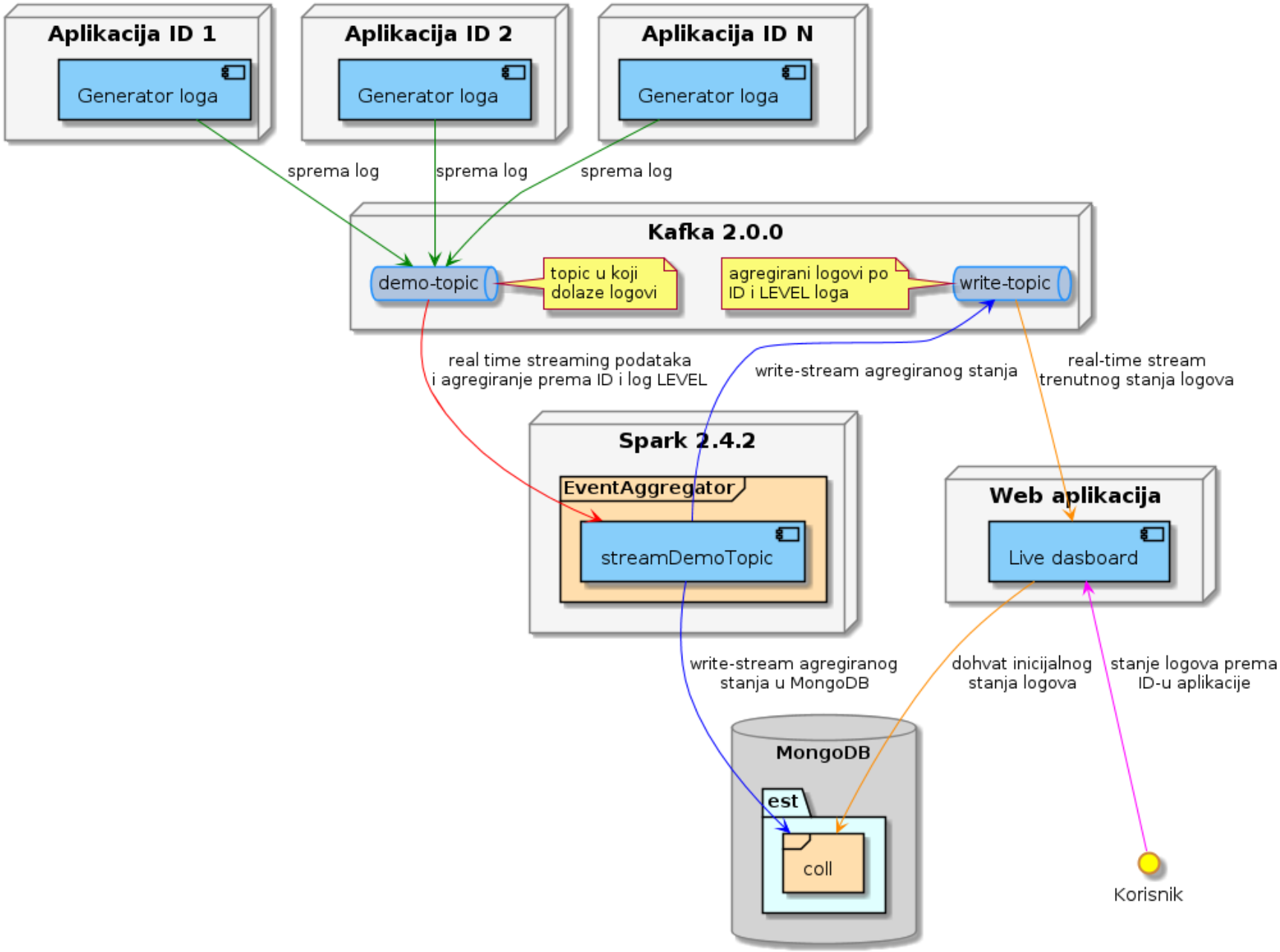
- O APIS IT d.o.o
- Poslovni use case
- Apache Kafka
- Apache Spark
- Spring Reactor

APIS IT d.o.o

- pruža strateške, stručne i provedbene usluge javnom sektoru RH u planiranju, razvoju, podršci i održavanju IT sustava
- više od 50 godina iskustva u IT poslovima s oko 450 zaposlenika
- izdvojeni projekti:
  - OIB
  - Fiskalizacija
  - Osobni korisnički pretinac
  - Izbori i referendumi
- jedan od najvećih data centara u RH

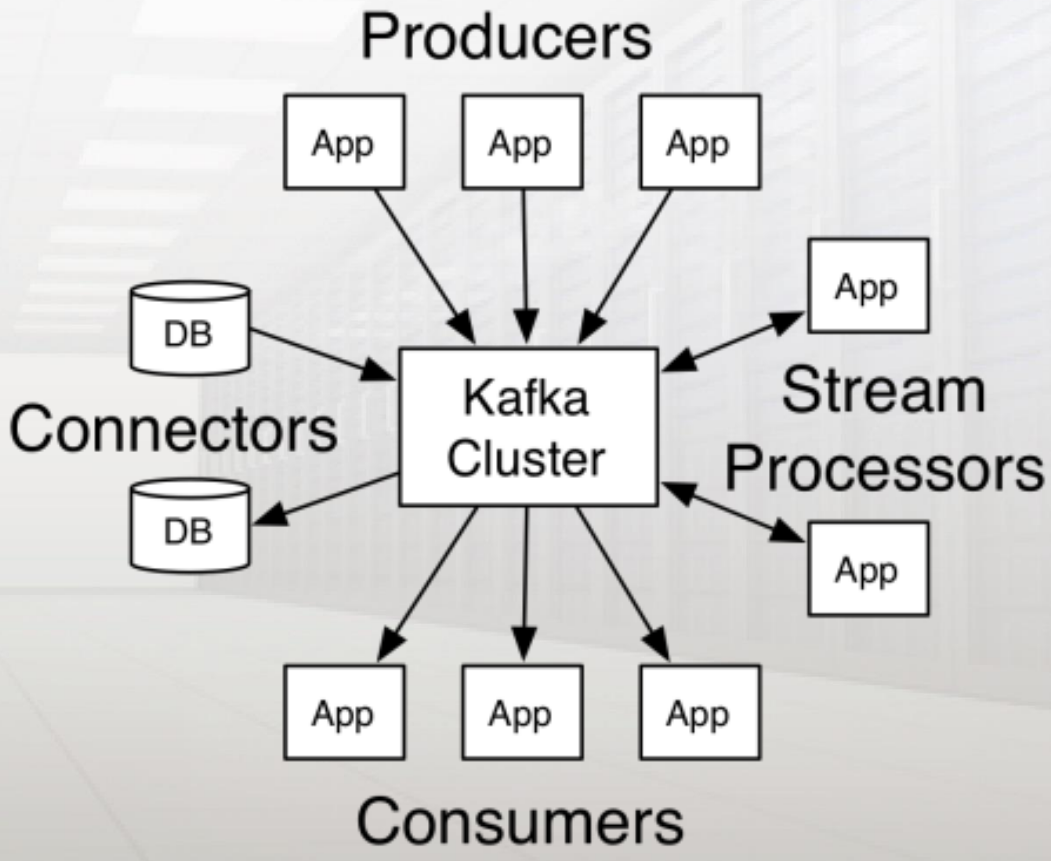
Poslovni use-case

- Live dashboard analitika:
  - prikaz raspodjele razine logova prema ID-u aplikacije
  - prikaz podataka u stvarnom vremenu uz kašnjenje od nekoliko sekundi
  - promjene se trebaju prikazati bez interakcije korisnika
  - mobile-friendly web aplikacija


Aplikacija ID 1
Generator loga
Aplikacija ID 2
Generator loga
Aplikacija ID N
Generator loga
sprema log
sprema log
sprema log
Kafka 2.0.0
demo-topic
topic u koji dolaze logovi
agregirani logovi po ID i LEVEL loga
write-topic
real time streaming podataka i agregiranje prema ID i log LEVEL
write-stream agregiranog stanja
real-time stream trenutnog stanja logova
Spark 2.4.2
EventAggregator
streamDemoTopic
Web aplikacija
Live dasboard
write-stream agregiranog stanja u MongoDB
dohvat inicijalnog stanja logova
stanje logova prema ID-u aplikacije
MongoDB
est
coll
Korisnik

Apache Kafka

- distribuirana platforma za razmjenu podataka
- podaci se spremaju u „topic”



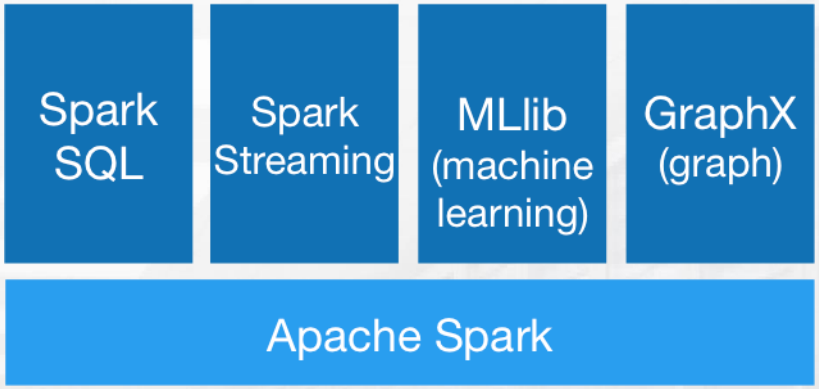
Apache Spark

- framework za obradu i analizu podataka
- podržani programski jezici – Scala, Java, Python, R
- temelji se na RDD API - **R**esilient **D**istributed **D**ataset – nepromjenjiva kolekcija
- kasnije su se razvili DataFrame API (podaci organizirani u kolone – strukturiranje podataka) i DataSet API (DataFrame + podrška za razne objekte)

Apache Spark - aplikacije

- 2 tipa operacija
  - transformacije – map, distinct, filter...
  - akcije – reduce, collect, count, take...
    - akcije -> (1..n) Job -> (1..n) Stage -> (1..n) Task
- lazy evaluation – izvođenje se pokreće samo na akcije, a transformacije proizvode nove RDDove
- za izvođenje treba Cluster Manager – YARN ili Spark standalone

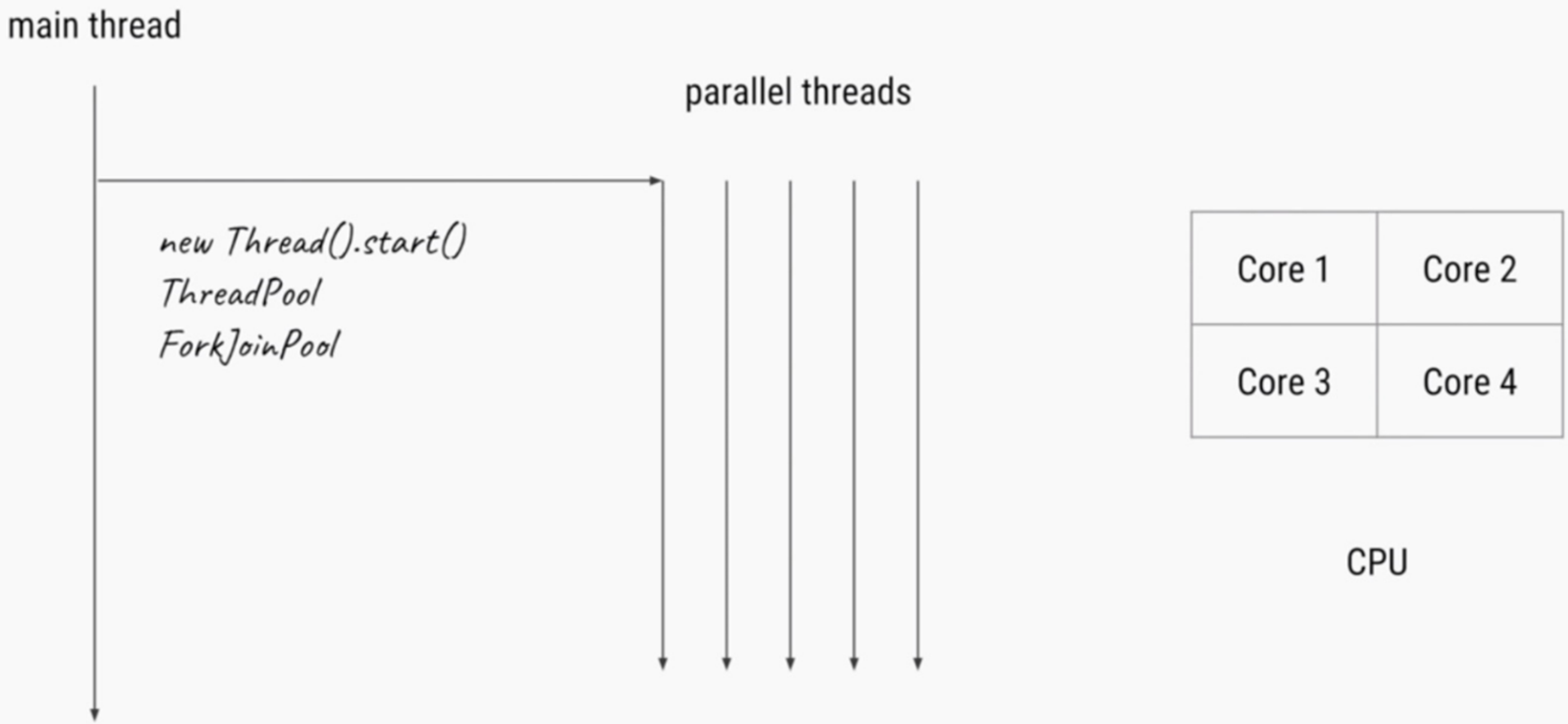
Apache Spark - biblioteke



- SQL i DataFrames – naredbe pisane SQL-om
- Spark Streaming
- MLlib – za strojno učenje
- GraphX – za rad s grafovima

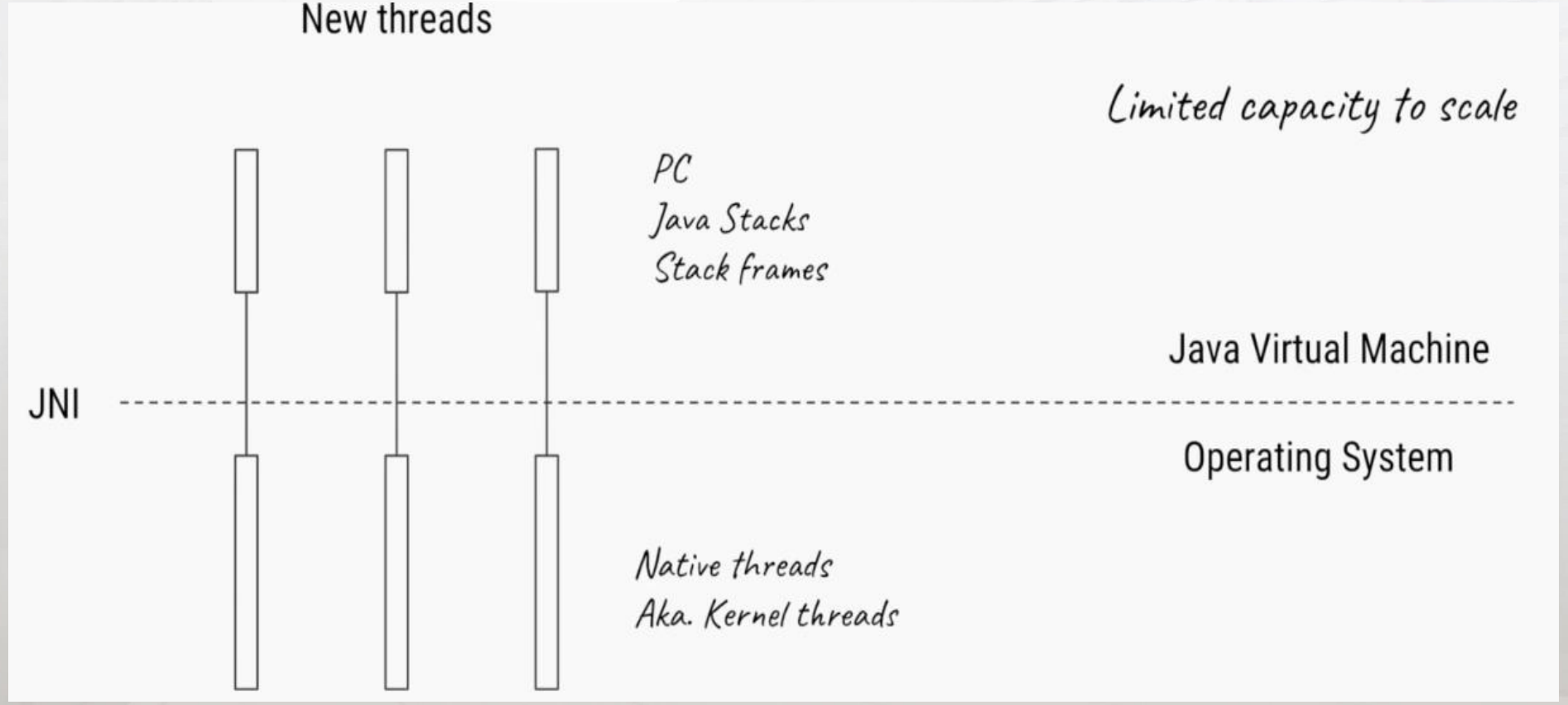
Spring Reactor – zašto?

- Uobičajeni način distribucije rada po dretvama



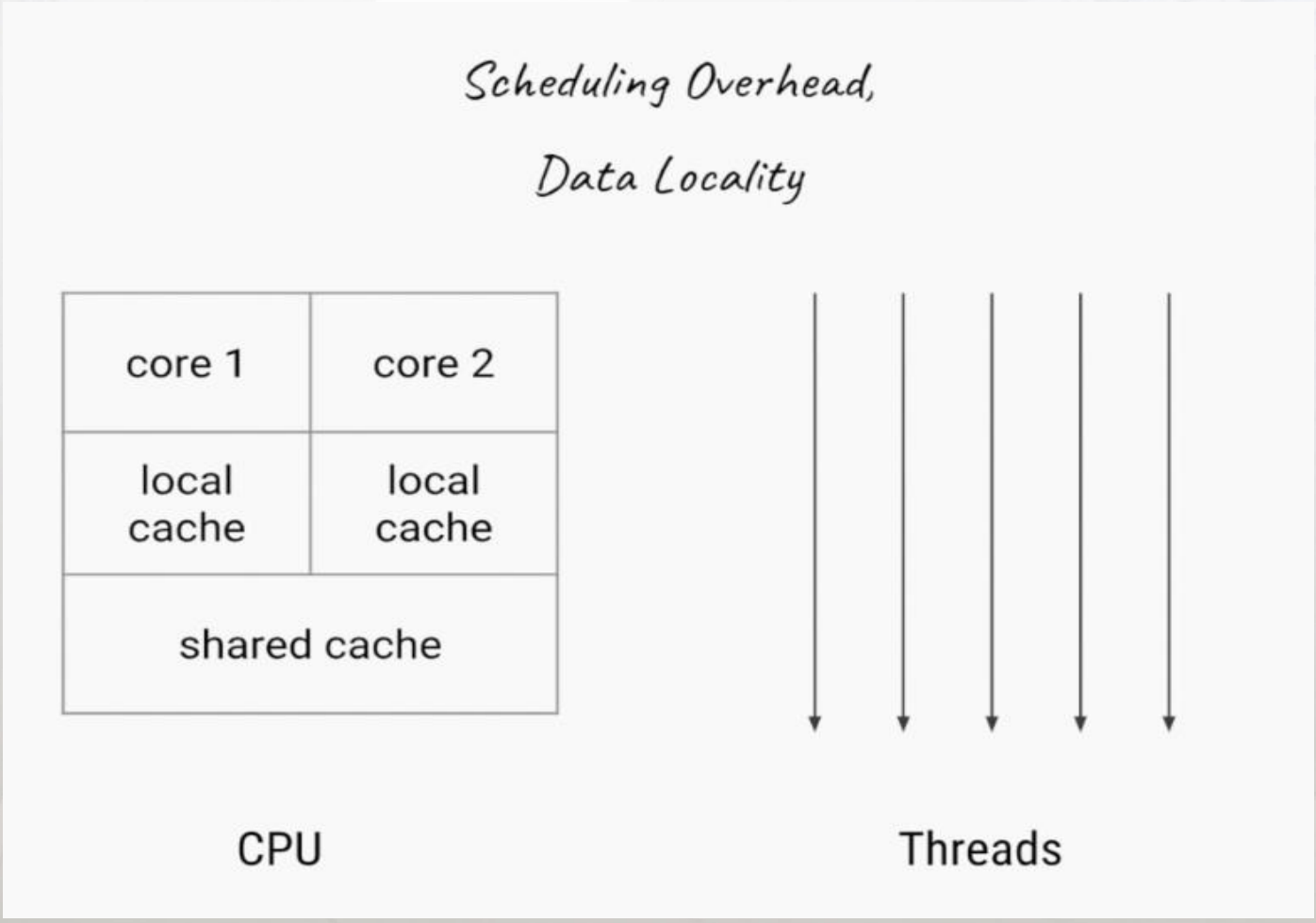
Spring Reactor – zašto?

- Prevelik broj thread-ova - problemi



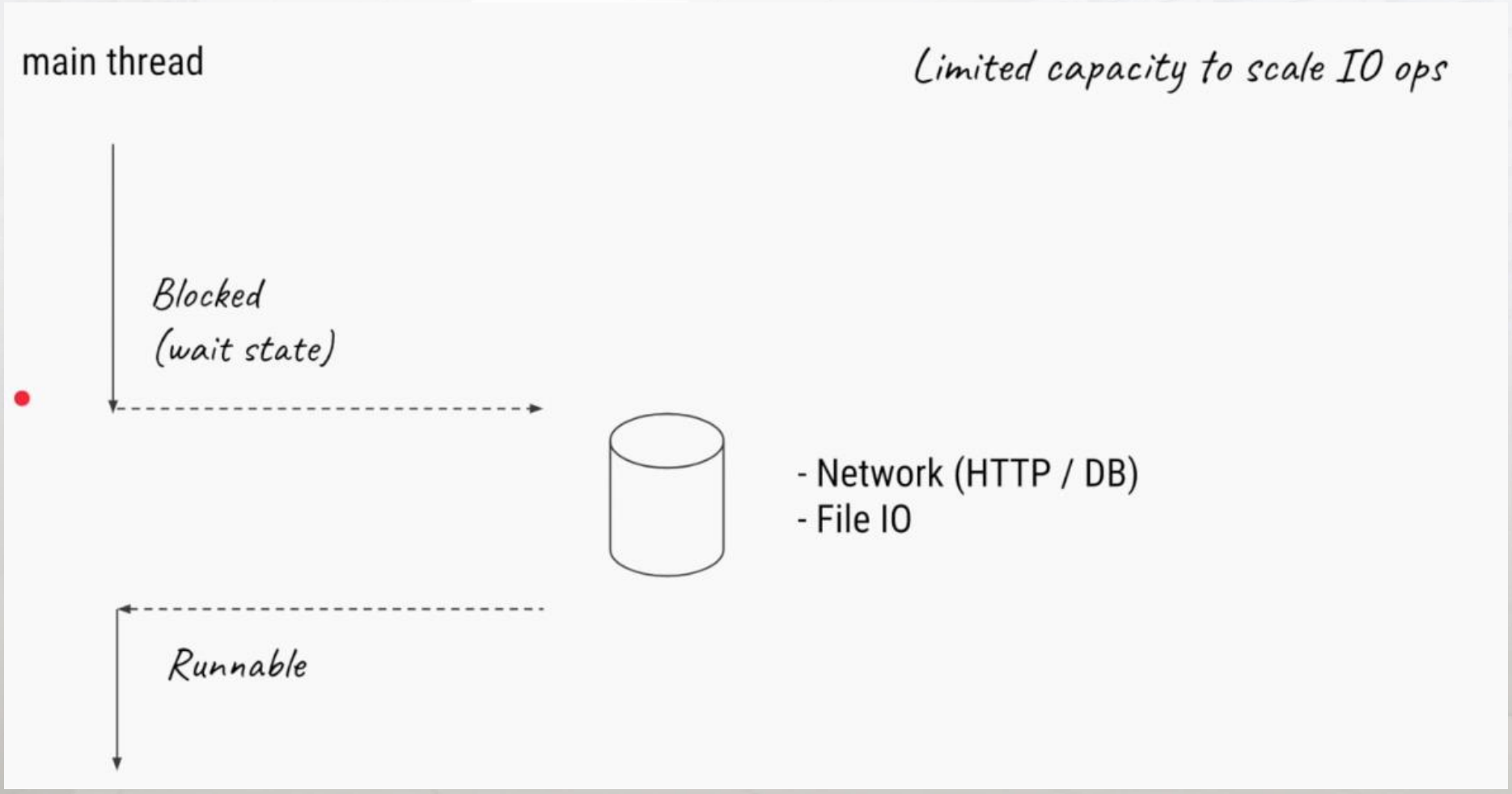
Spring Reactor – zašto?

- Prevelik broj thread-ova - problemi



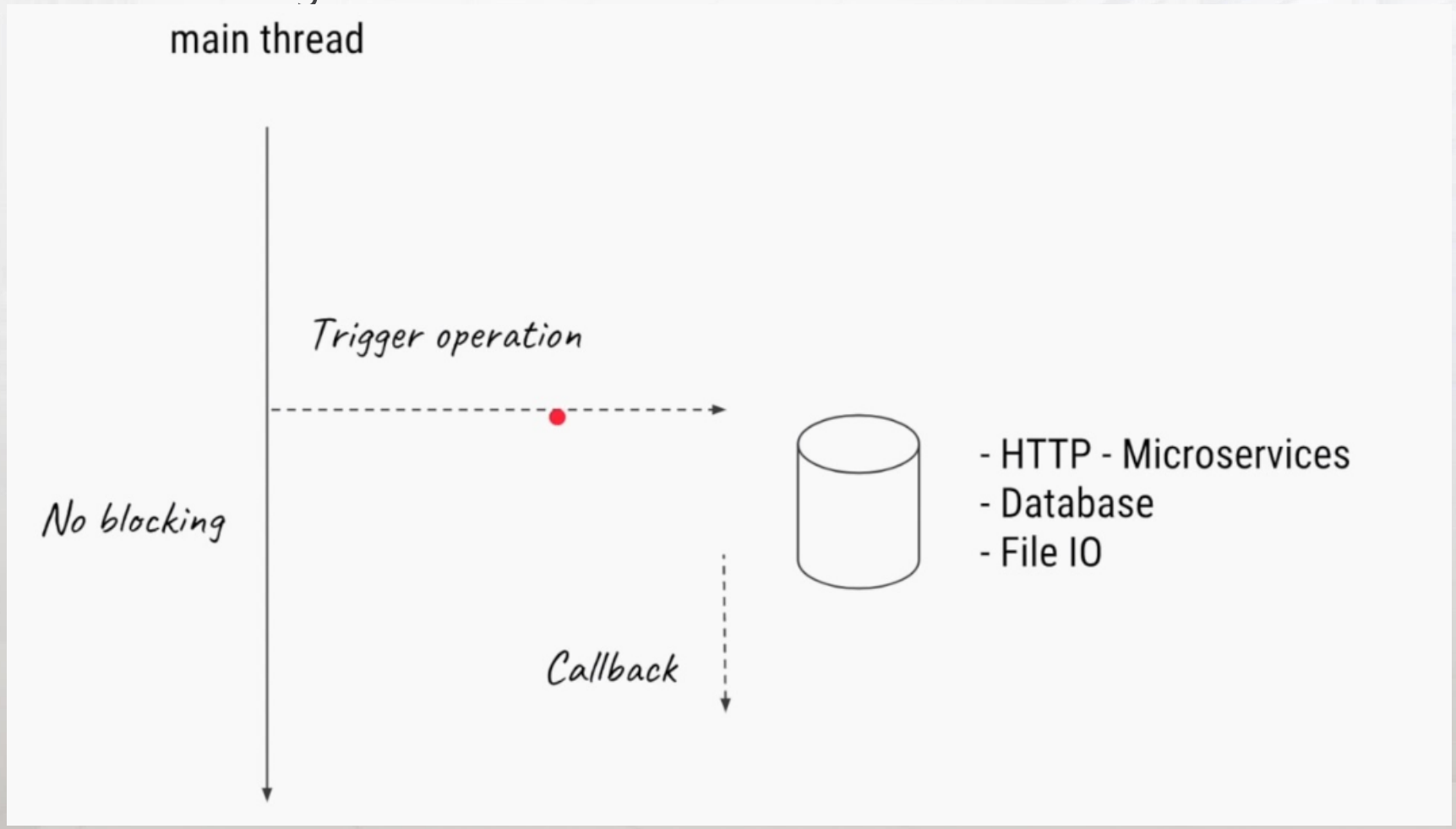
Spring Reactor – zašto?

- Blocking IO



Spring Reactor – zašto?

- Non blocking IO



Spring Reactor – zašto?

- Java standard async CompletableFuture, Java NIO
- Servlets 3.0, 3.1

Spring Reactor

- Reactive vs. async

Hvala na pažnji

www.apis-it.hr