Project Loom:

A Summary For Busy Developers

Michael Easter

@codetojoy

Project Loom

~~Fibers~~ and Continuations

OpenJDK

Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki ·IRC

Bylaws · Census
Legal

JEP Process

Source code
Mercurial
GitHub

Tools
Mercurial
Git
jtreg harness

Groups
(overview)

JDK 19

This release will be the Reference Implementation of version 19 of the Java SE Platform.

Status

The main-line code repository is open for bug fixes, small enhancements, and JEPs as proposed and tracked via the JEP Process.

Schedule

2022/06/09	Rampdown Phase One (fork from main line)
2022/07/21	Rampdown Phase Two
2022/08/11	Initial Release Candidate
2022/08/25	Final Release Candidate
2022/09/20	General Availability

OpenJDK

Installing
Contributing
Sponsoring
Developers' Guide
Vulnerabilities
JDK GA/EA Builds

Mailing lists
Wiki · IRC

Bylaws · Census
Legal

JEP Process

Source code
Mercurial
GitHub

Tools
Mercurial
Git
jtreg harness

Groups
(overview)

JDK 19

This release will be the Reference Implementation of version 19 of the Java SE Platform.

Status

The main-line code repository is open for bug fixes, small enhancements, and JEPs as proposed and tracked via the JEP Process.

Schedule

2022/06/09	Rampdown Phase One (fork from main line)
2022/07/21	Rampdown Phase Two
2022/08/11	Initial Release Candidate
2022/08/25	Final Release Candidate
2022/09/20	General Availability

JEP 425: Virtual Threads (Preview)

JEP 428: Structured Concurrency (Incubator)

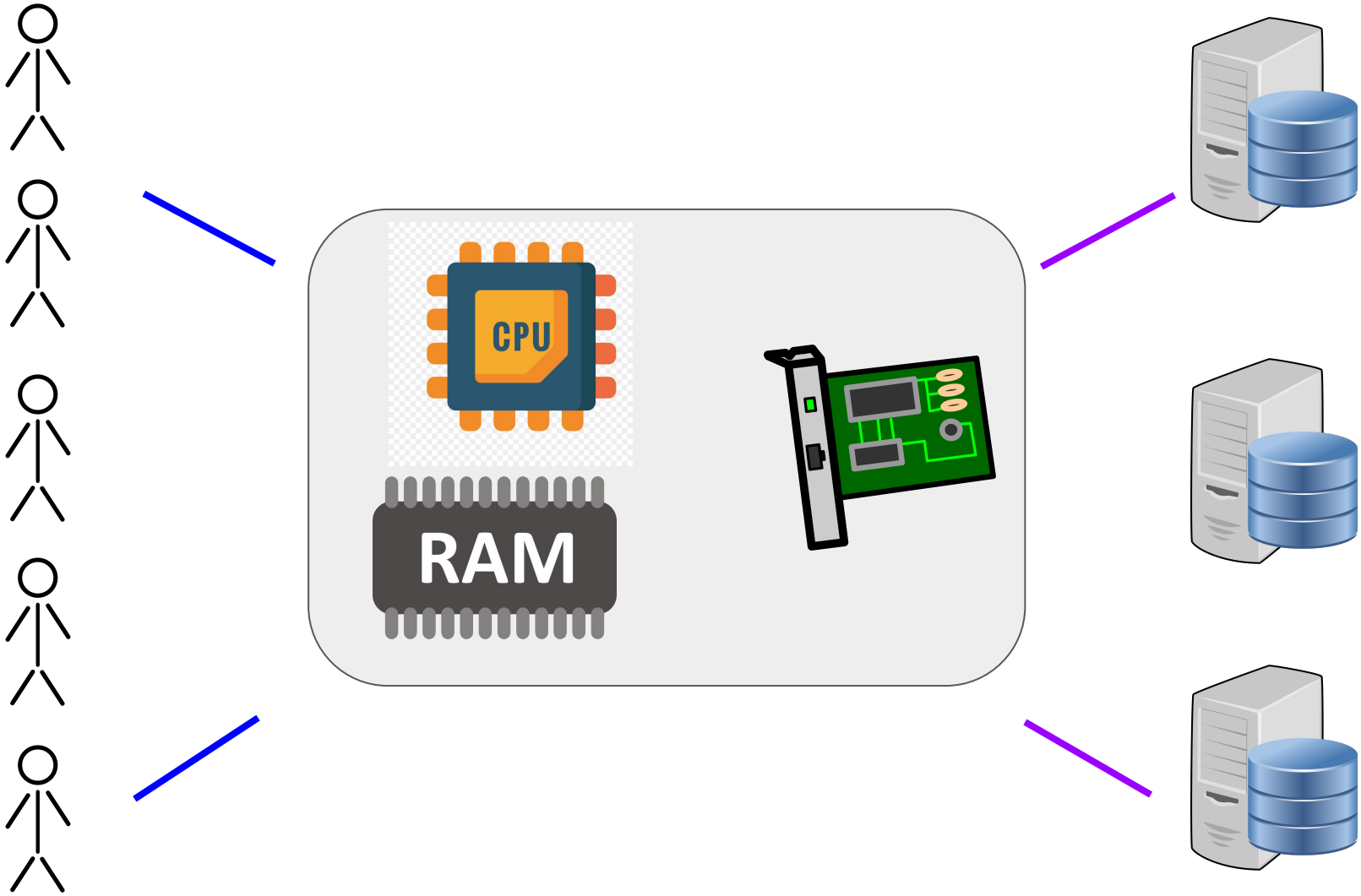
Virtual Threads

Platform threads

```
14  void createManyThreads(int numThreads) throws Exception {
15      Thread thread = null;
16
17      for (int i = 0; i < numThreads; i++) {
18          thread = new Thread(Runner::doWork);
19          thread.start();
20      }
21
22      thread.join();
23  }
```

Virtual threads

```
14  void createManyThreads(int numThreads) throws Exception {
15      Thread thread = null;
16
17      for (int i = 0; i < numThreads; i++) {
18          thread = Thread.startVirtualThread(Runner::doWork);
19      }
20
21      thread.join();
22  }
```


CPU
RAM

OS (Kernel)

JVM

OS (Kernel)

OS (Kernel)

JVM

Scheduler

OS (Kernel)

JVM

Scheduler

OS (Kernel)

Heap

JVM

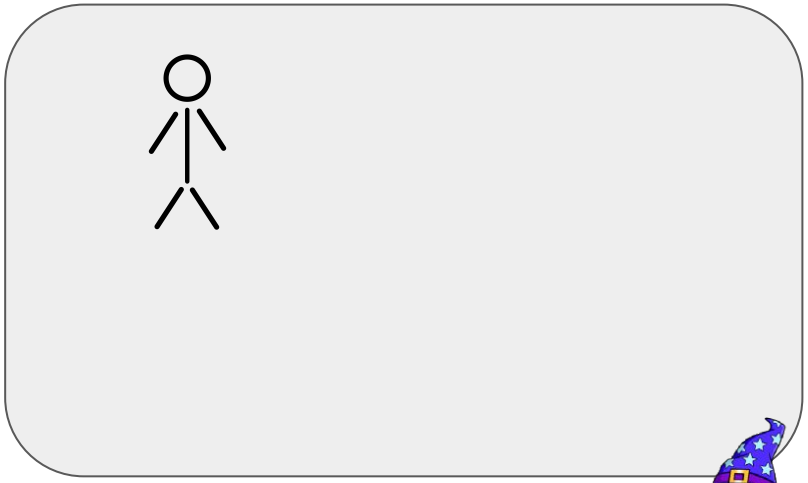
Scheduler

OS (Kernel)

Heap

Continuation

JVM



Project Loom

~~Fibers~~ and Continuations

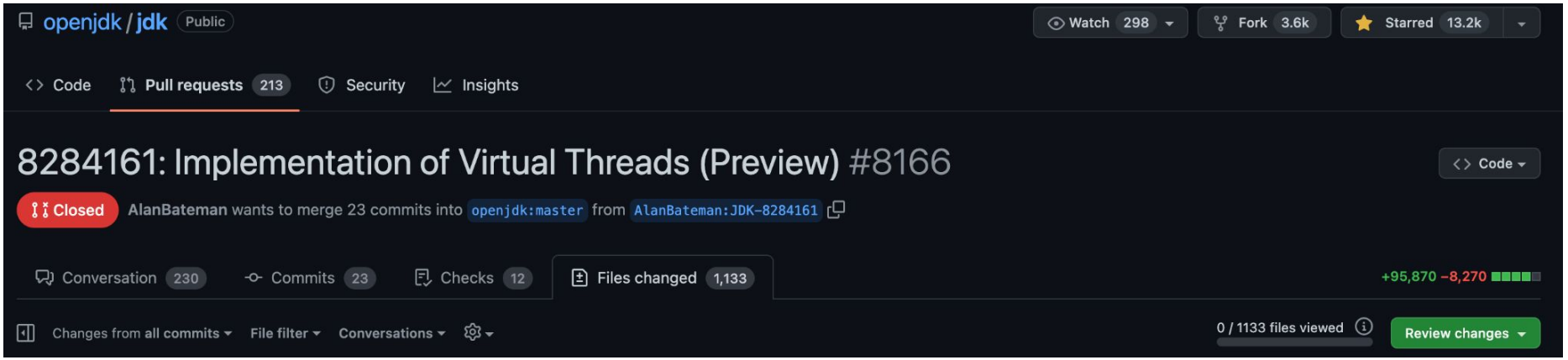
Project Loom

~~Fibers~~ and Continuations

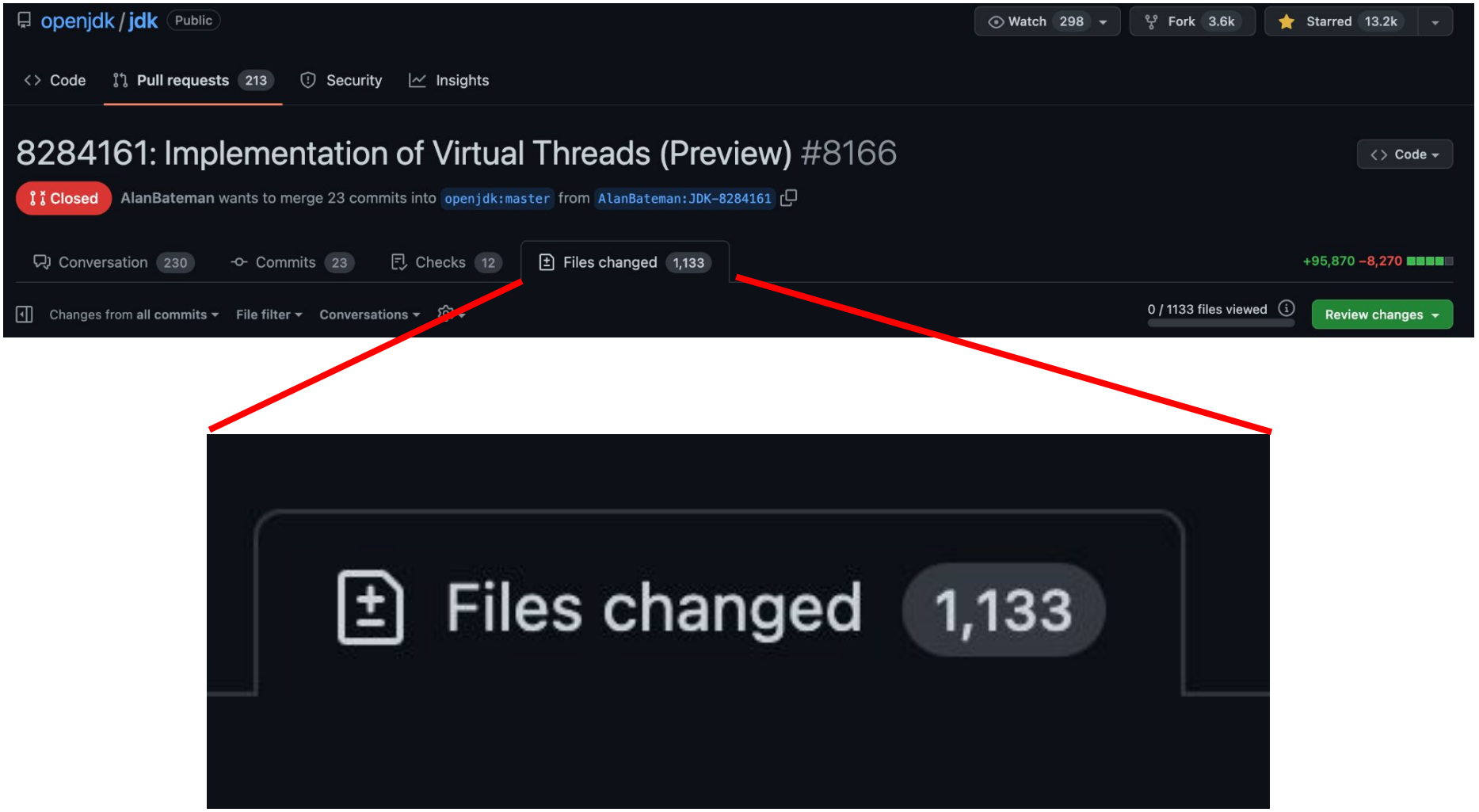
Project Loom

~~Fibers~~ and Continuations

Pull Request



Pull Request



Pull Request

openjdk / jdk Public

Code | Pull requests 213 | Security | Insights

8284161: Implementation of Virtual Threads (Preview) #8166

Closed AlanBateman wants to merge 23 commits into `openjdk:master` from `AlanBateman:JDK-8284161`

Conversation 230 | Commits 23 | Checks 12 | Files changed 1,133 +95,870 −8,270

Changes from all commits | File filter | Conversations | 0 / 1133 files viewed | Review changes

Pull Request

openjdk / jdk Public

Code | Pull requests 213 | Security | Insights

8284161: Implementation of Virtual Threads (Preview) #8166

Closed AlanBateman wants to merge 23 commits into `openjdk:master` from `AlanBateman:JDK-8284161`

Conversation 230 | Commits 23 | Checks 12 | Files changed 1,133 +95,870 −8,270

Changes from all commits | File filter | Conversations | 0 / 1133 files viewed | Review changes

Pull Request

Codes like sync, scales like async

```
49  void run() throws Exception {
50      try (var executor = Executors.newVirtualThreadPerTaskExecutor()) {
51          int numTasks = 10;
52          for (int i = 0; i < numTasks; i++) {
53              executor.submit(new MyTask(i, database));
54          }
55      }
56  }
```

Codes like sync, scales like async

```
30  class MyTask implements Runnable {
31      @Override
32      public void run() {
33          var user = database.findUser(id);
34          // ...
35      }
```

Pros

JVM

Pros

sync

JVM

Pros

sync

scale

JVM

Pros

sync

scale

JVM

simple

Pros

sync

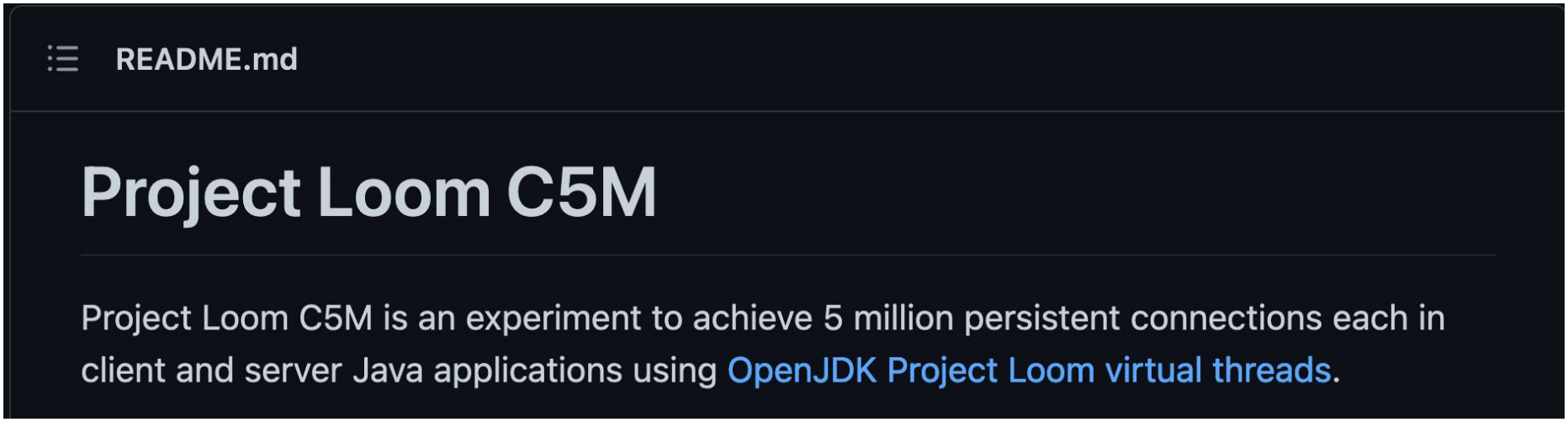
scale

JVM

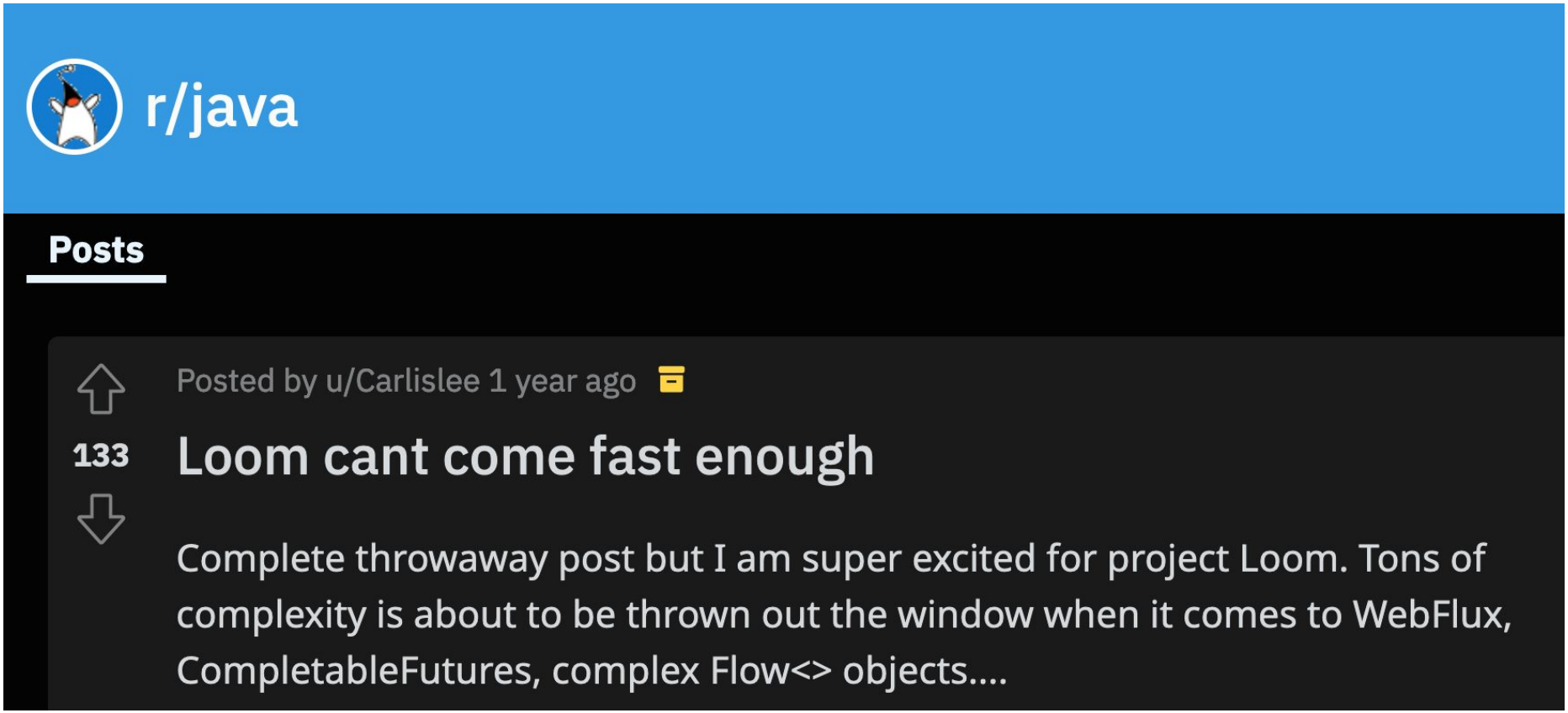
simple

debug,
tooling,
monitoring

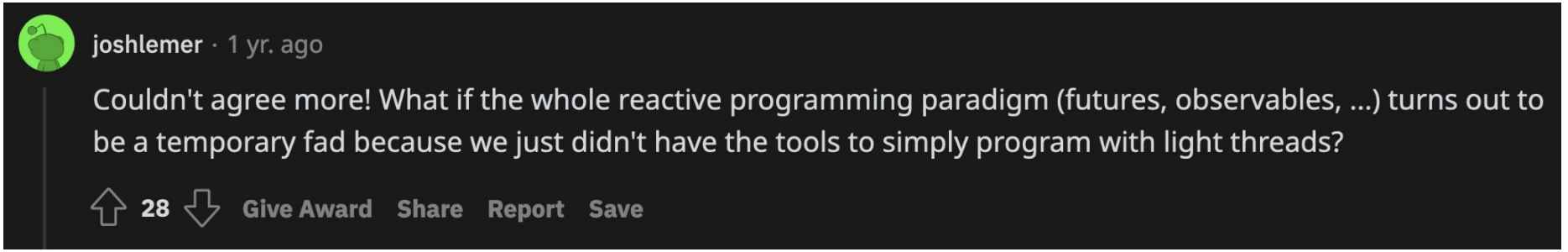
Loom C5M


README.md
Project Loom C5M
Project Loom C5M is an experiment to achieve 5 million persistent connections each in client and server Java applications using OpenJDK Project Loom virtual threads.

Hype


r/java
Posts
Posted by u/Carlislee 1 year ago
133
Loom cant come fast enough
Complete throwaway post but I am super excited for project Loom. Tons of complexity is about to be thrown out the window when it comes to WebFlux, CompletableFutures, complex Flow<> objects....

Hype

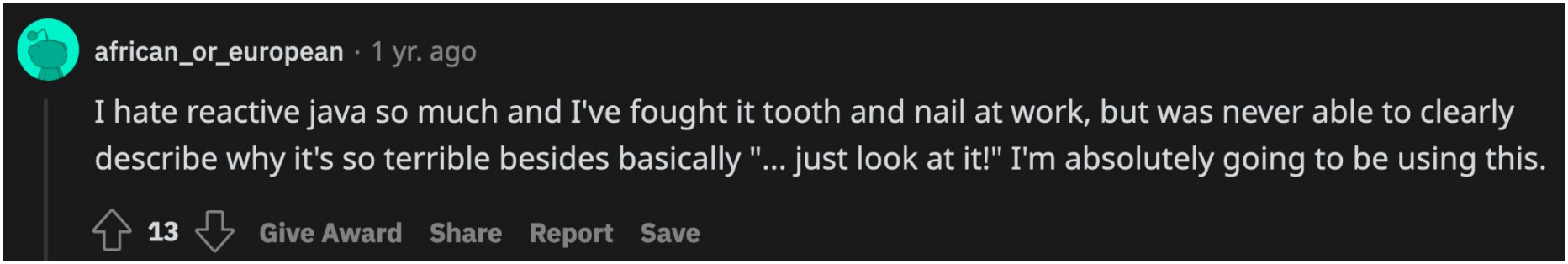


joshlemer · 1 yr. ago

Couldn't agree more! What if the whole reactive programming paradigm (futures, observables, ...) turns out to be a temporary fad because we just didn't have the tools to simply program with light threads?

28 Give Award Share Report Save

Hype



african_or_european · 1 yr. ago

I hate reactive java so much and I've fought it tooth and nail at work, but was never able to clearly describe why it's so terrible besides basically "... just look at it!" I'm absolutely going to be using this.

13 Give Award Share Report Save

“Loom will kill Reactive programming”

Reactive


Oleh Dokuka

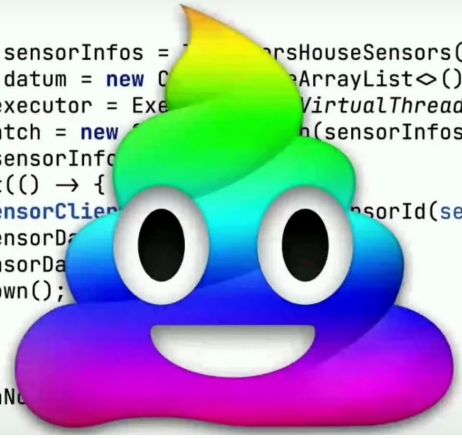
:-|

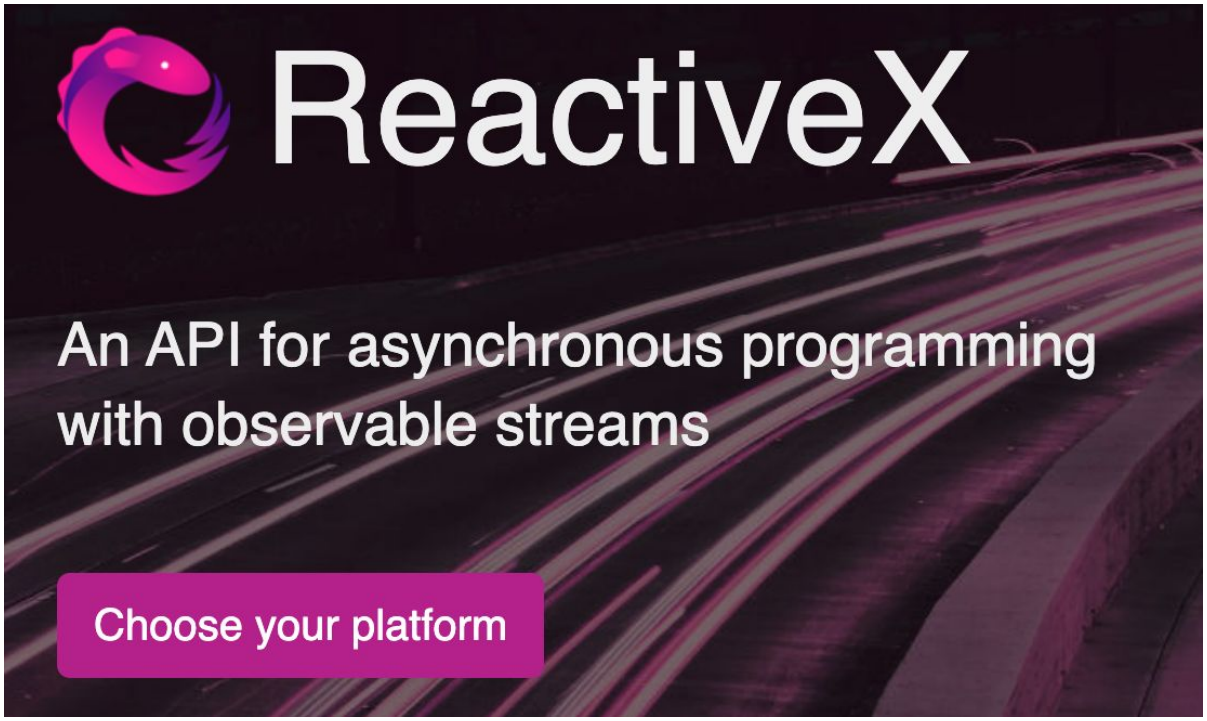

Project Loom

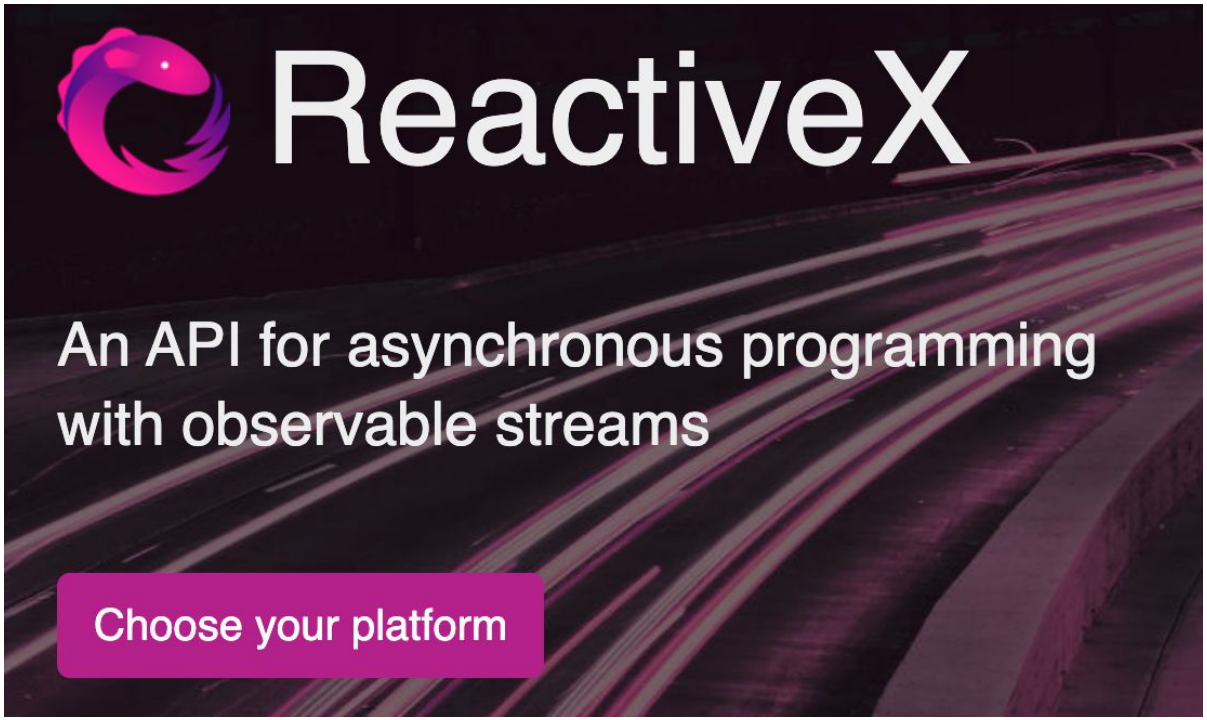

Oleh Dokuka

:-|


```
List<SensorInfo> sensorInfos = rsHouseSensors(userId, houseId);
List<SensorData> datum = new ArrayList<>();
ExecutorService executor = VirtualThreadPerTaskExecutor();
CountDownLatch latch = new (sensorInfos.size());
for (SensorInfo sensorInfo
  executor.submit(() -> {
    double t = sensorClier nsorId(sensorInfo.id());
    SensorData sensorDa
    datum.add(sensorDa
    latch.countDown();
  });
}
latch.await(1000,
executor.shutdownN
```




ReactiveX
An API for asynchronous programming
with observable streams
Choose your platform

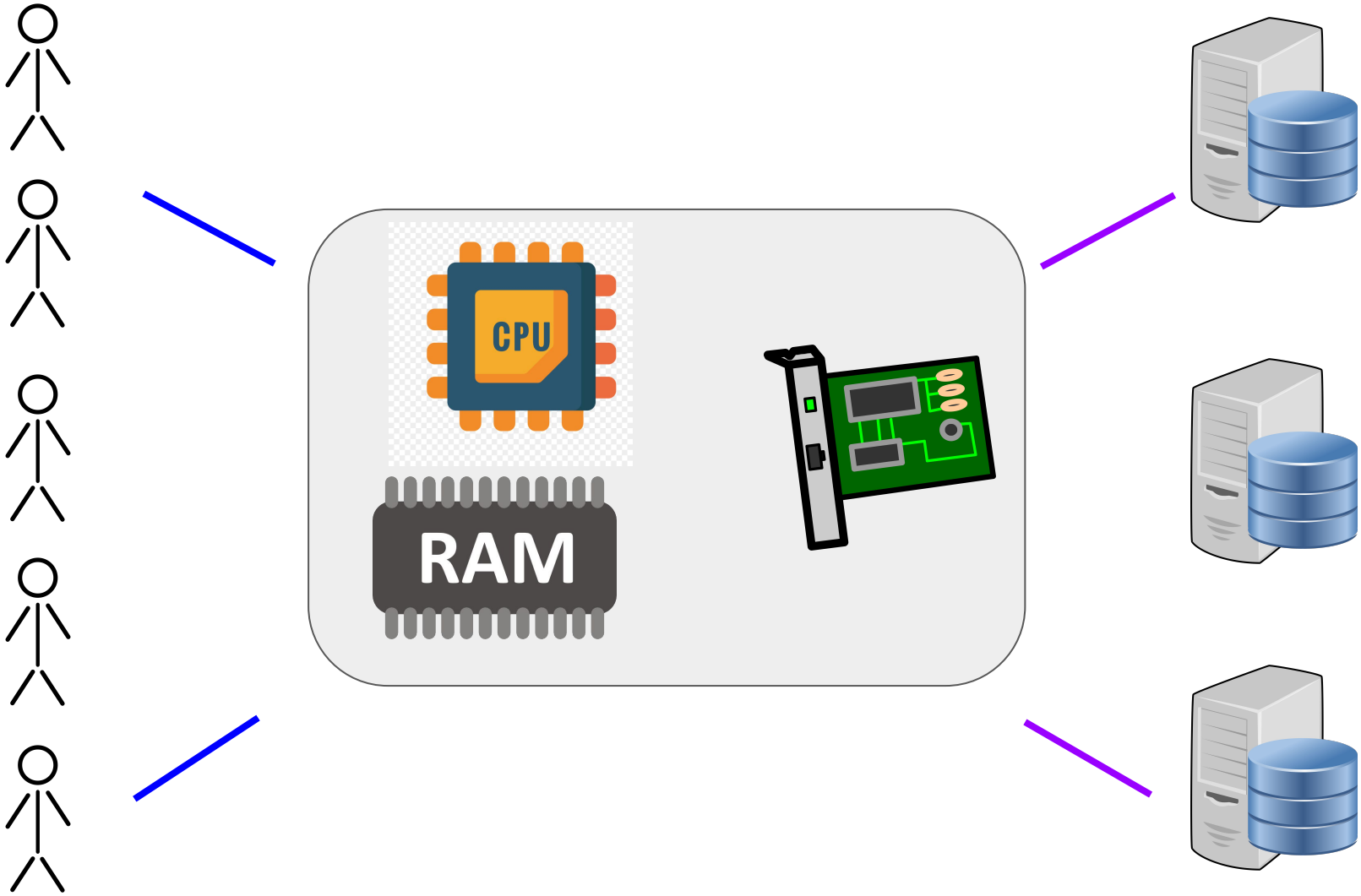


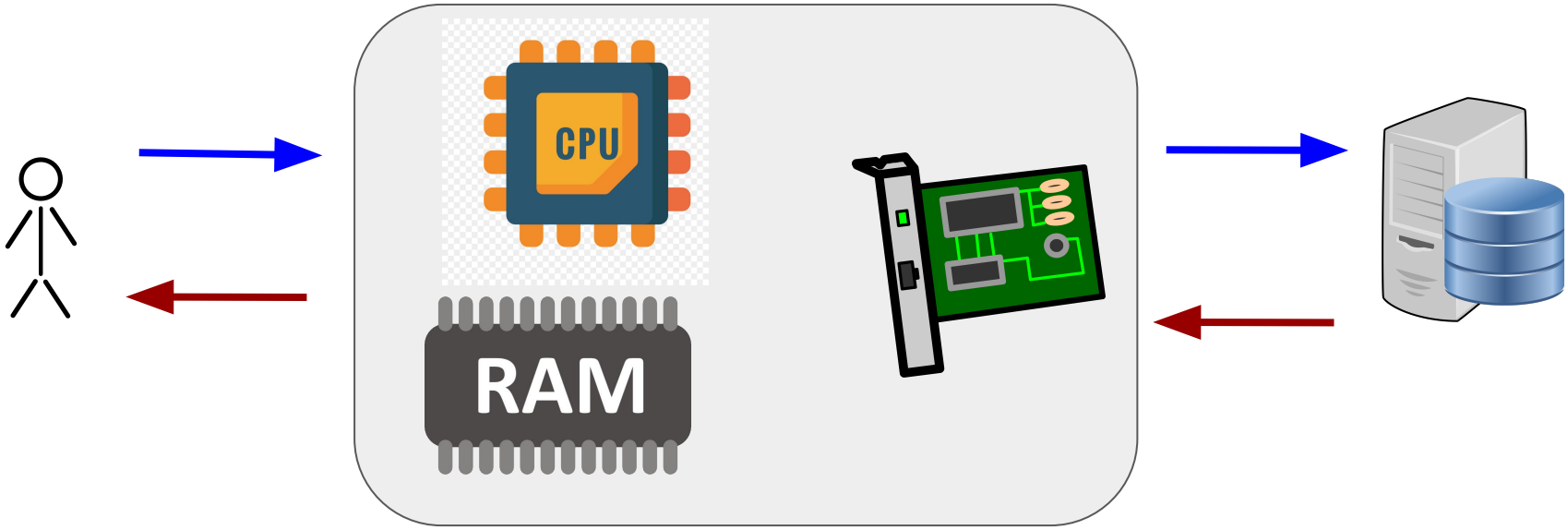
Languages

- Java: RxJava
- JavaScript: RxJS
- C#: Rx.NET
- C#(Unity): UniRx
- Scala: RxScala
- Clojure: RxClojure
- C++: RxCpp
- Lua: RxLua
- Ruby: Rx.rb
- Python: RxPY
- Go: RxGo
- Groovy: RxGroovy
- JRuby: RxJRuby
- Kotlin: RxKotlin
- Swift: RxSwift
- PHP: RxPHP
- Elixir: reaxive
- Dart: RxDart

The Reactive Manifesto

Published on September 16 2014. (v2.0)

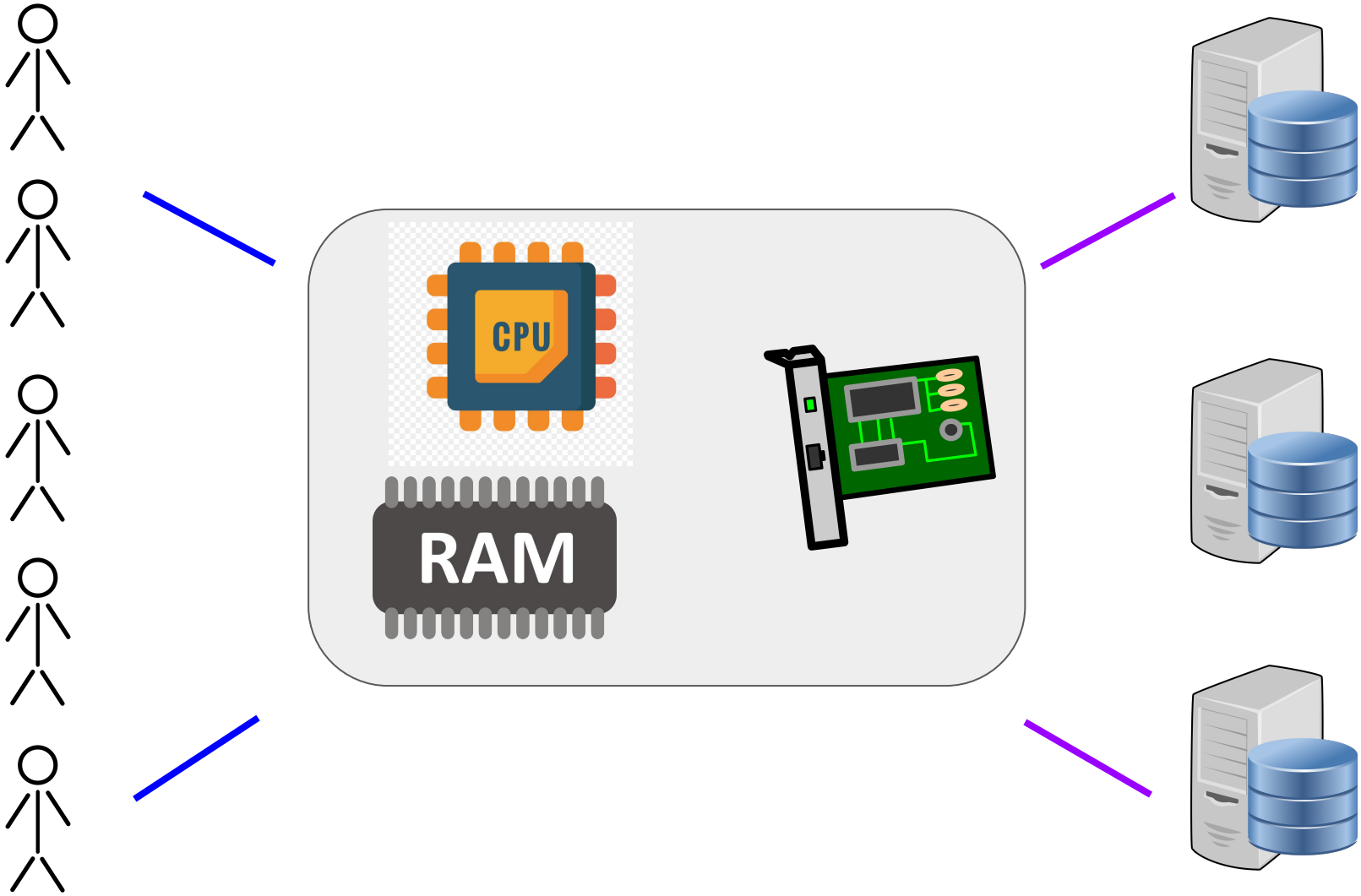

CPU
RAM

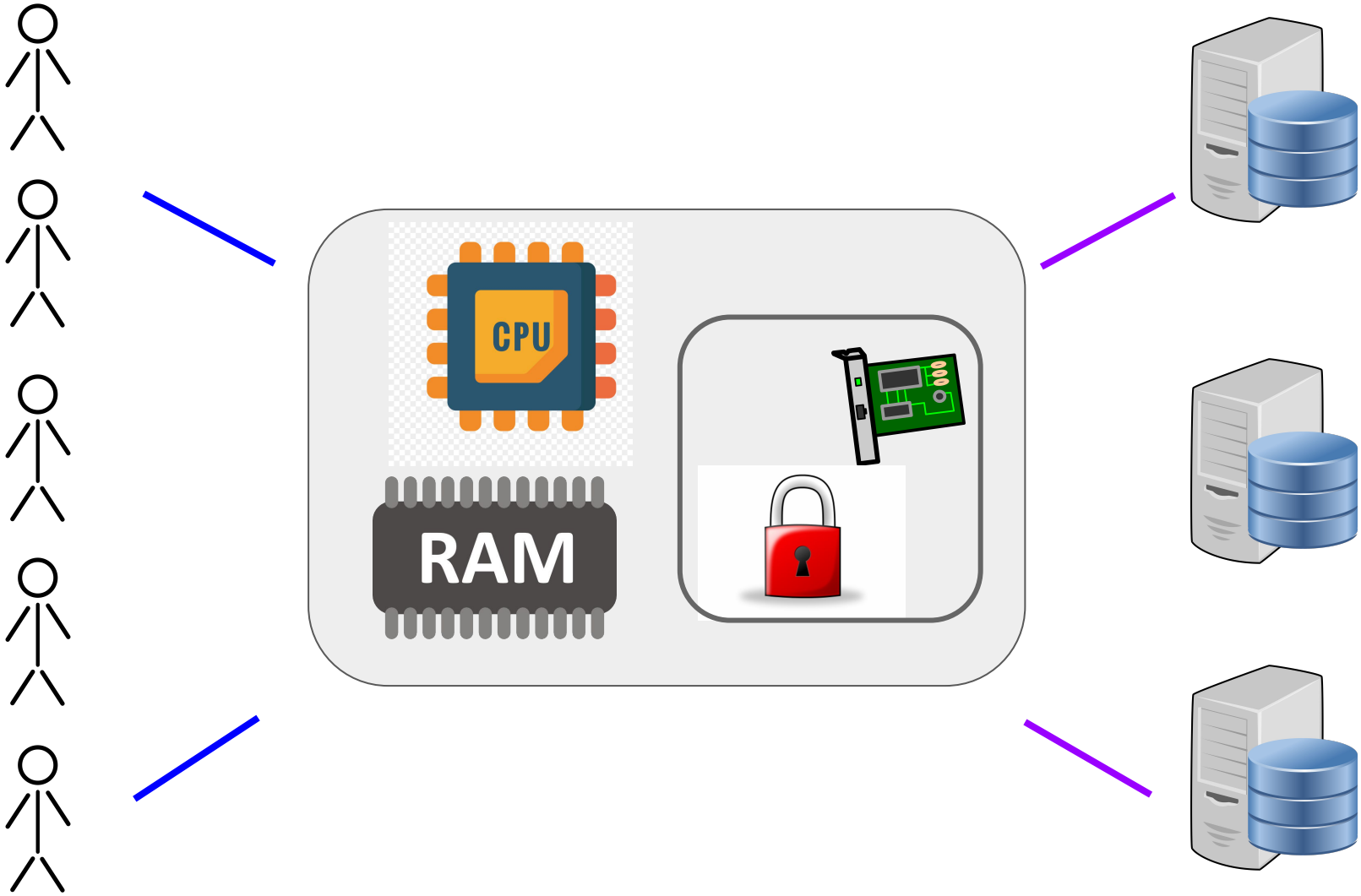

CPU
RAM

Cons

scale

JVM


CPU
RAM


CPU
RAM

Locks

```
45  class MyTask implements Runnable {
46      @Override
47      public void run() {
48          try {
49              var database = pool.acquireDatabase();
50              var user = database.findUser(id);
51              // ...
52          } catch (Exception ex) {
53              // #yolo
54          } finally {
55              pool.releaseDatabase();
56          }
57      }
```

Locks

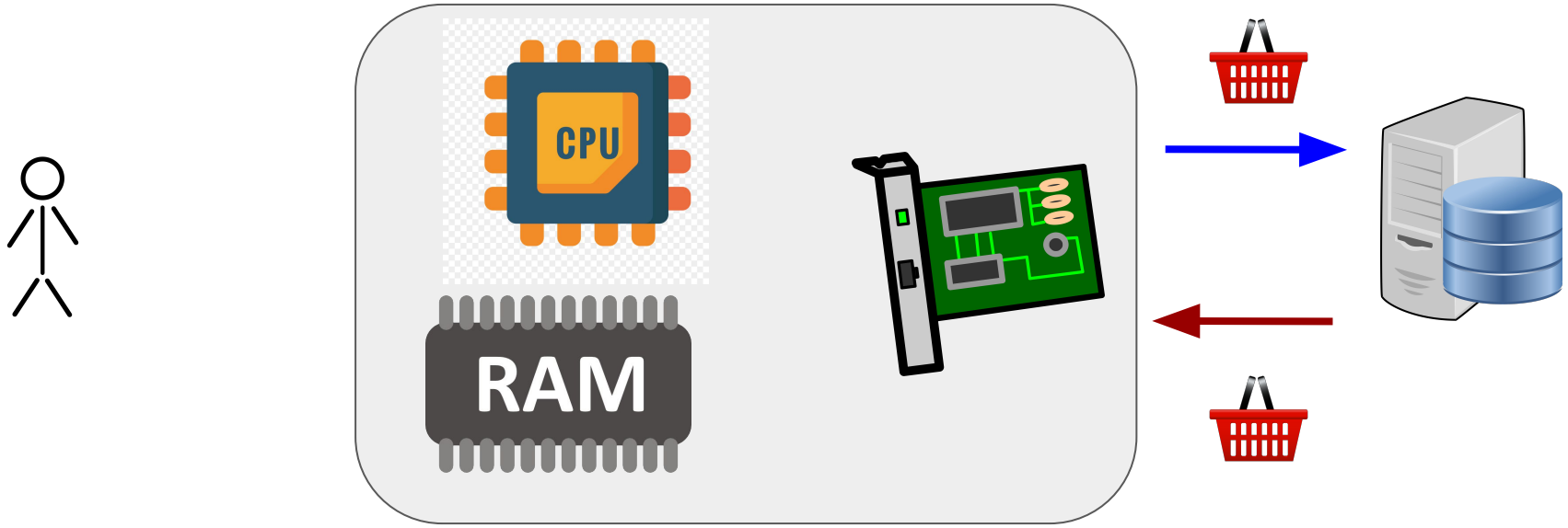
```
30  class ConnectionPool {
31      private static final int MAX_AVAIL = 2;
32      private final Semaphore available = new Semaphore(MAX_AVAIL
33      private final Database database = new Database();
34
35      Database acquireDatabase() throws InterruptedException {
36          available.acquire();
37          return database;
38      }
39
40      void releaseDatabase() {
41          available.release();
42      }
43  }
```

Cons

scale

back pressure

JVM

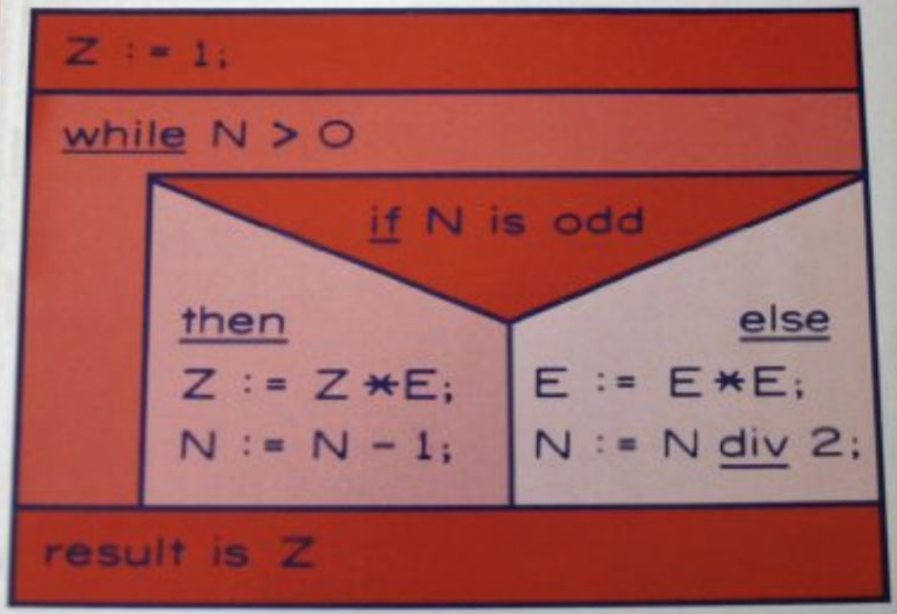

CPU
RAM

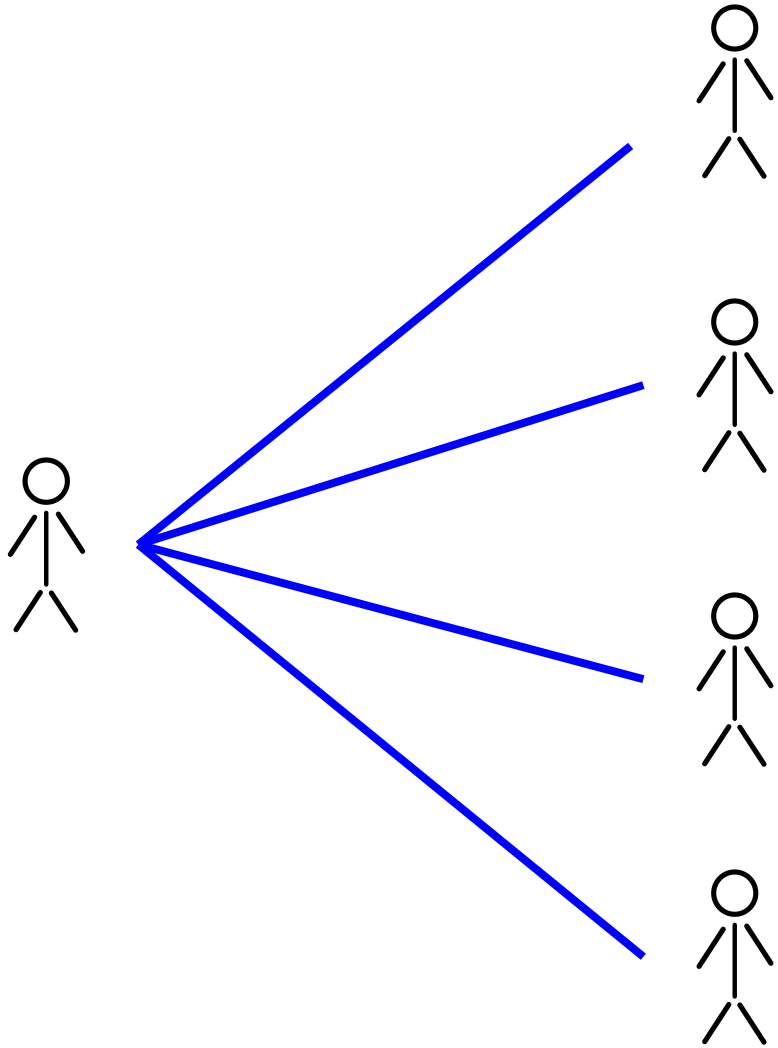
Structured Concurrency

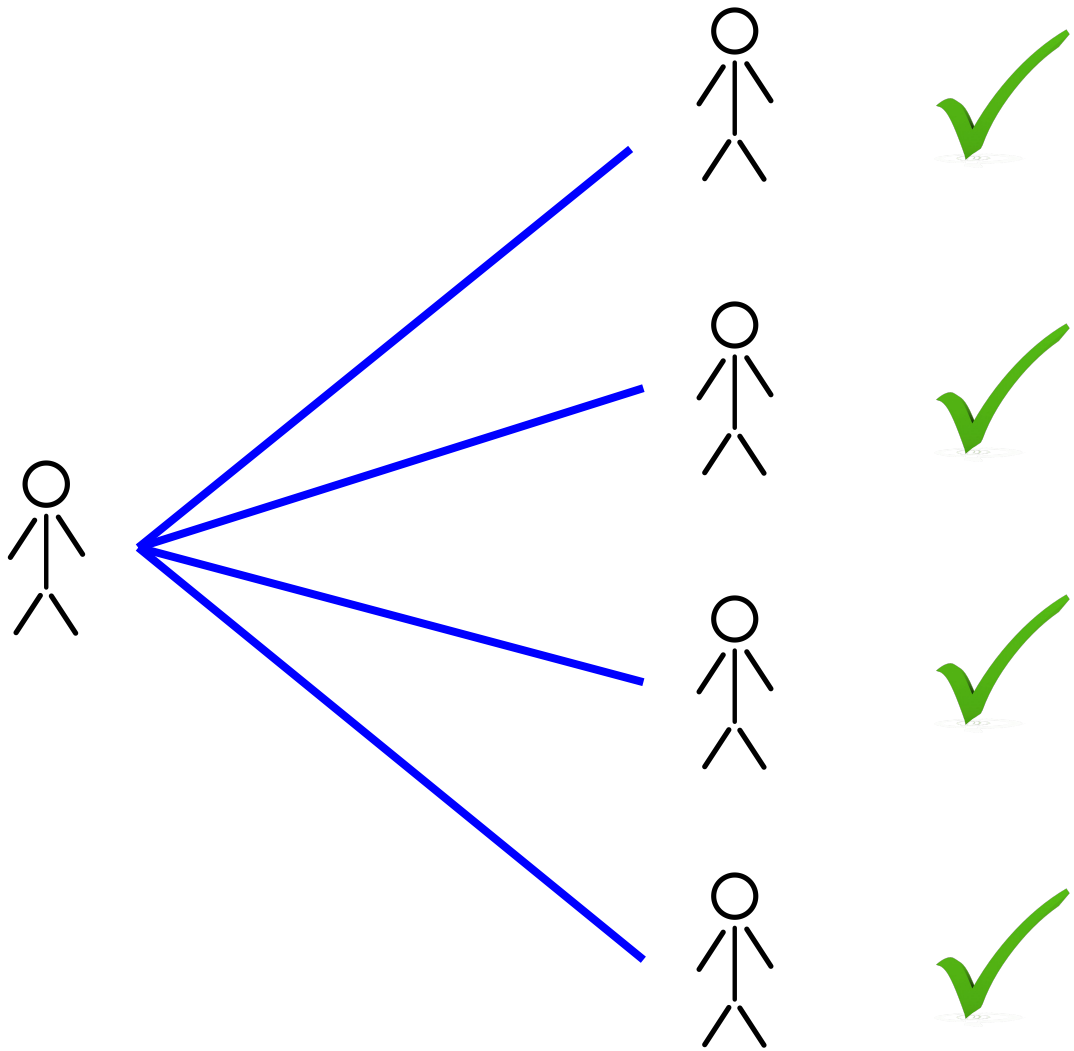
```
792415C0    55          push ebp
792415C1    89E5        mov ebp, esp
792415C3    8B45 08     mov eax, [ebp+0x08]
792415C6    DB28        fld tword [eax]
792415C8    8B4D 0C     mov ecx, [ebp+0x0C]
792415CB    DB29        fld tword [ecx]
792415CD    DEC1        faddp
792415CF    8B55 10     mov edx, [ebp+0x10]
792415D2    DB3A        fstp tword [edx]
792415D4    DB68 0A     fld tword [eax+0x0A]
792415D7    DB69 0A     fld tword [ecx+0x0A]
792415DA    DEC1        faddp
792415DC    DB7A 0A     fstp tword [edx+0x0A]
792415DF    5D          pop ebp
792415E0    C2 0C00     ret 0x000C
```

STRUCTURED PROGRAMMING AND PROBLEM-SOLVING WITH PASCAL

Richard B. Kieburtz

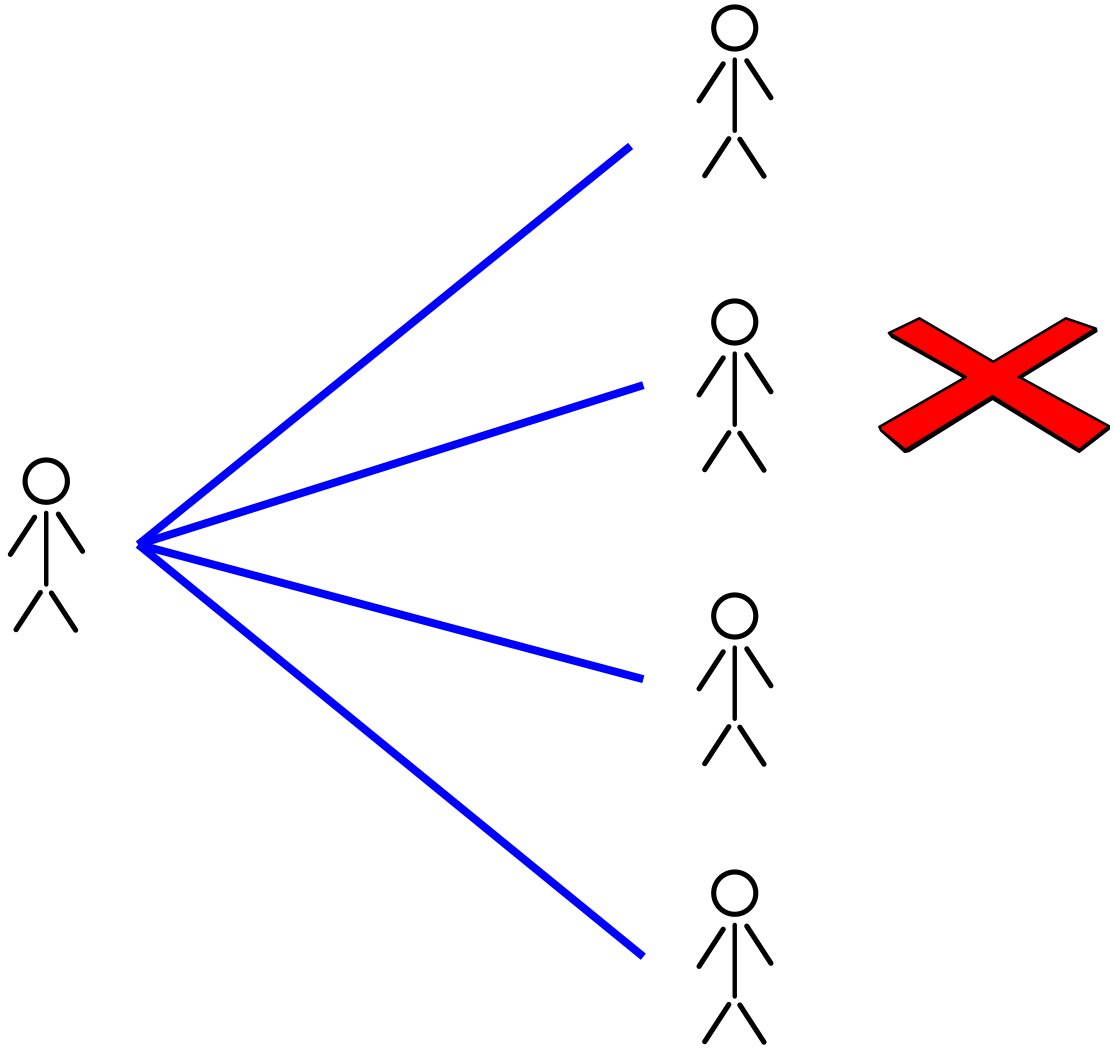


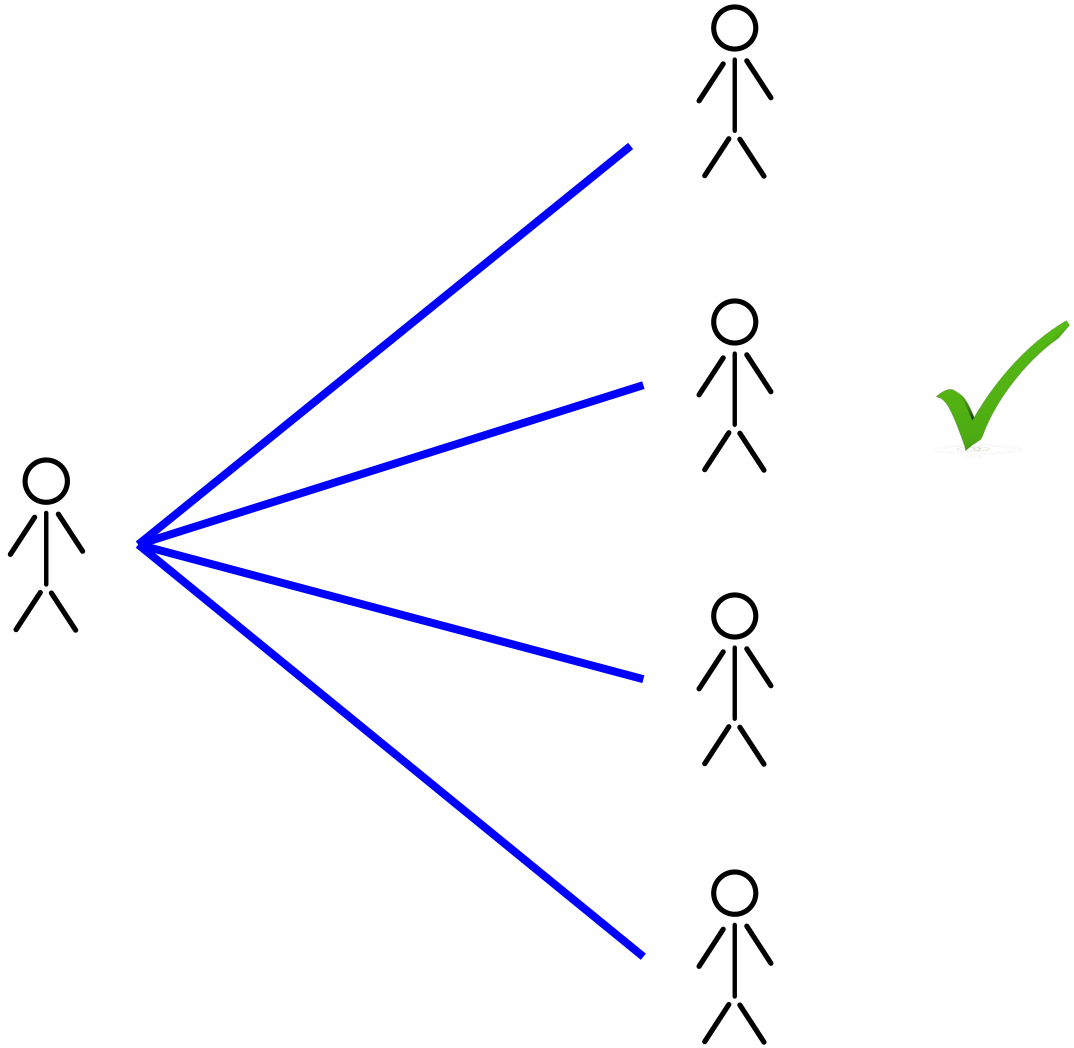




Invoke All

```
35  String run() throws Exception {
36      try (var scope = new StructuredTaskScope.ShutdownOnFailure()) {
37          Future<String> foo = scope.fork(() -> taskFoo());
38          Future<String> bar = scope.fork(() -> taskBar());
39
40          scope.join();
41          scope.throwIfFailed();
42
43          return foo.resultNow() + " " + bar.resultNow();
44      }
45  }
```





Invoke Any

```
35  String run() throws Exception {
36      try (var scope =
37              new StructuredTaskScope.ShutdownOnSuccess<String>()) {
38          Future<String> foo = scope.fork(() -> taskFoo());
39          Future<String> bar = scope.fork(() -> taskBar());
40
41          scope.join();
42
43          return scope.result();
44      }
45  }
```

thank you
danke
謝謝
ngiyabonga
شكراً جزيلاً
рахмат
teşekkür ederim
Баярлалаа
спасибо
faafetai lava
mersi
kia ora
barka
welalin
tack
spas
dank je
misaotra
matondo
paldies
grazzi
mahalo
tapadh leat
vinaka
спасибі
blagodaram
gracias
хвала
kiitos
dankie
nanni
nandri
dhanyavad
asante
manana
obrigada
tenki
bedankt
hvala
maururu
köszönöm
bayarlalaa
gràçie
akun
dankon
aciú
djiere dieuf
tau
mochchakkeram
murakoze
enkosi
дякую
mamnun
dziękuję
chnorakaloutioun
gratias ago
gràcies
sulpáy
go raibh maith agat
chokrane
sobodi
dėkuji
taiku
obrigado
sukriya
kop khun krap
mèsi
sagolun
arigatô
takk
dakujem
trugarez
ありがとう
tanemirt
grazie
didi madloba
najis tuke
terima kasih
rahmet
shukriya
mercé
мерси
kam sah hamnida
rahmat
diolch
dhanyavadagalu
ευχαριστώ
xièxie
তোমাকে ধন্যবাদ
감사합니다
merci

Code, Credits, Links

Available here on GitHub

POST-SCRIPT
TODO: DELETE FROM HERE