

第109SMALLTALK勉強会

TEAMTALK

CLUSTER COMPUTING LIBRARY FOR PHARO SMALLTALK

Quentin Plessis
(カンタン プレシ)

BACKGROUND

- ▶ Multi processing is not natively possible in Pharo Smalltalk
- ▶ To perform resource intensive operations, it is necessary to split the load between several Pharo instances
- ▶ Efficiently setting up several Pharo instances is troublesome
- ▶ Establishing communication between several Pharo instances is tricky

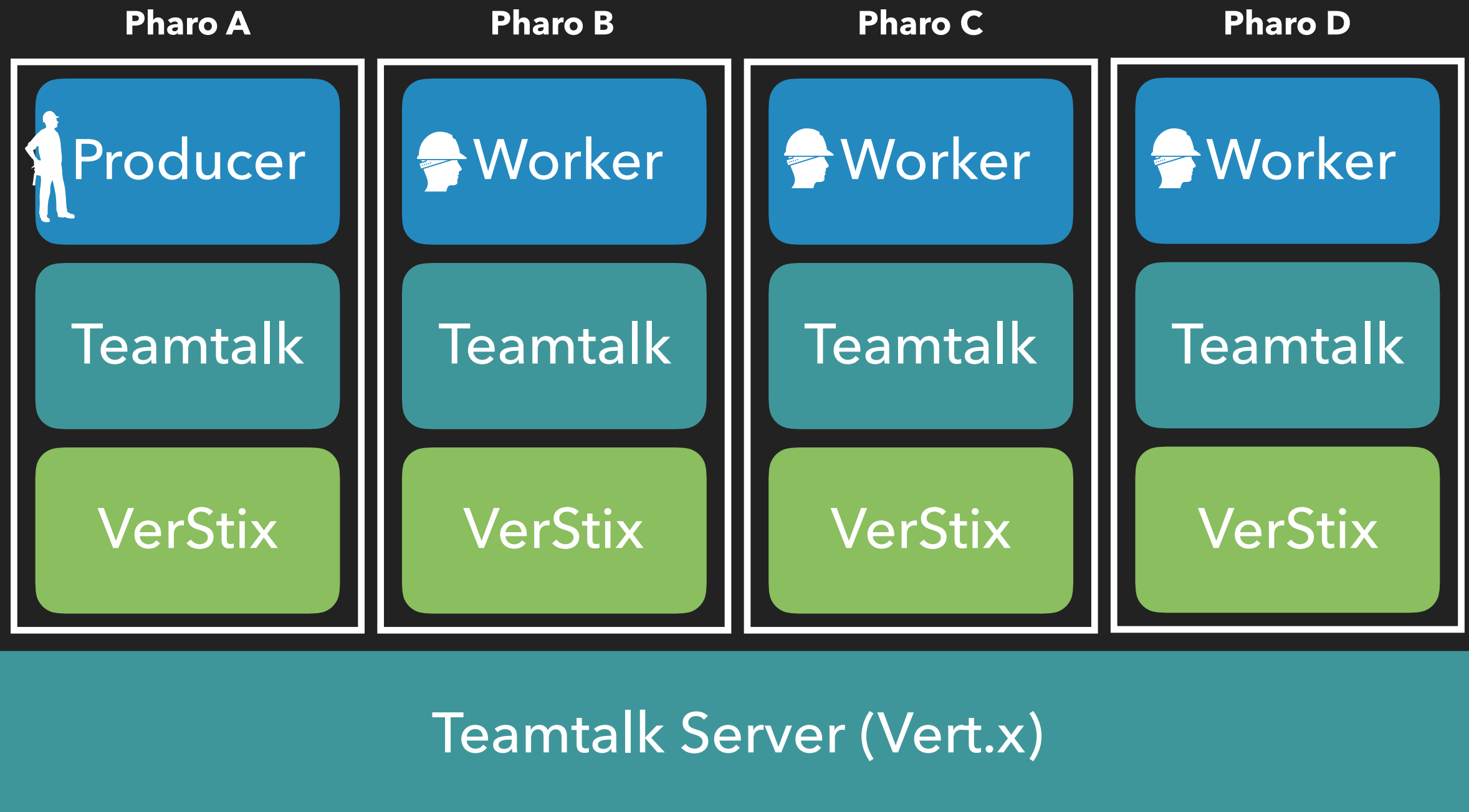
BACKGROUND

- ▶ Multi processing is not natively possible in Pharo Smalltalk
- ▶ To perform resource intensive operations, it is necessary to split the load between several Pharo instances
- ▶ ~~Efficiently setting up~~ **Docker image**
~~troublesome~~ <https://github.com/mumez/pharo-vnc-supervisor>
- ▶ ~~Establishing communication~~ **VerStix (Vert.x based communication tool)**
~~instances is risky~~ <https://github.com/mumez/VerStix>
<https://www.slideshare.net/umejava/verstix>

TEAMTALK

<https://github.com/quentinplessis/Teamtalk>

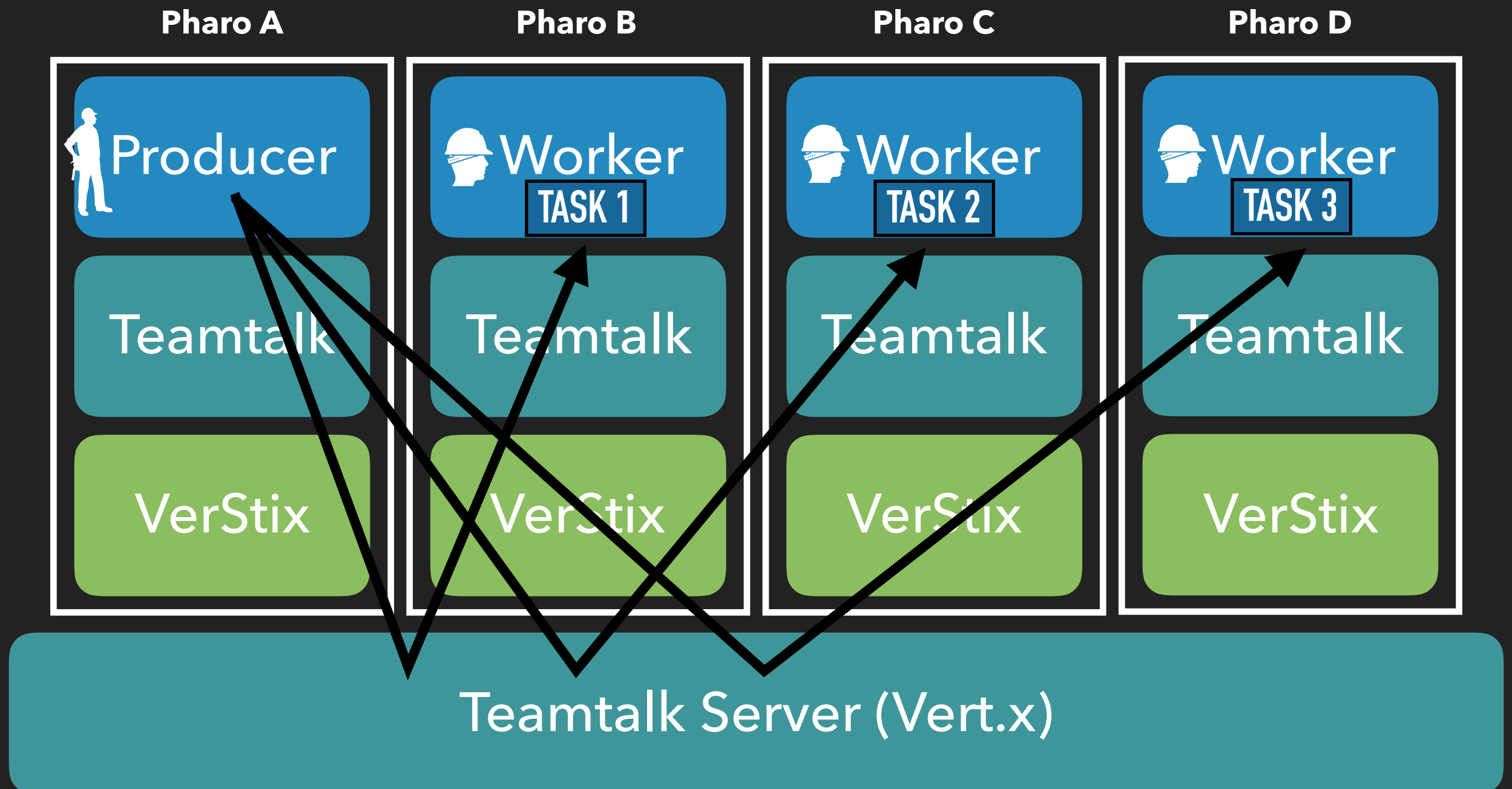
- Pharo Smalltalk library making it possible to execute tasks in a cluster of Pharos instances.



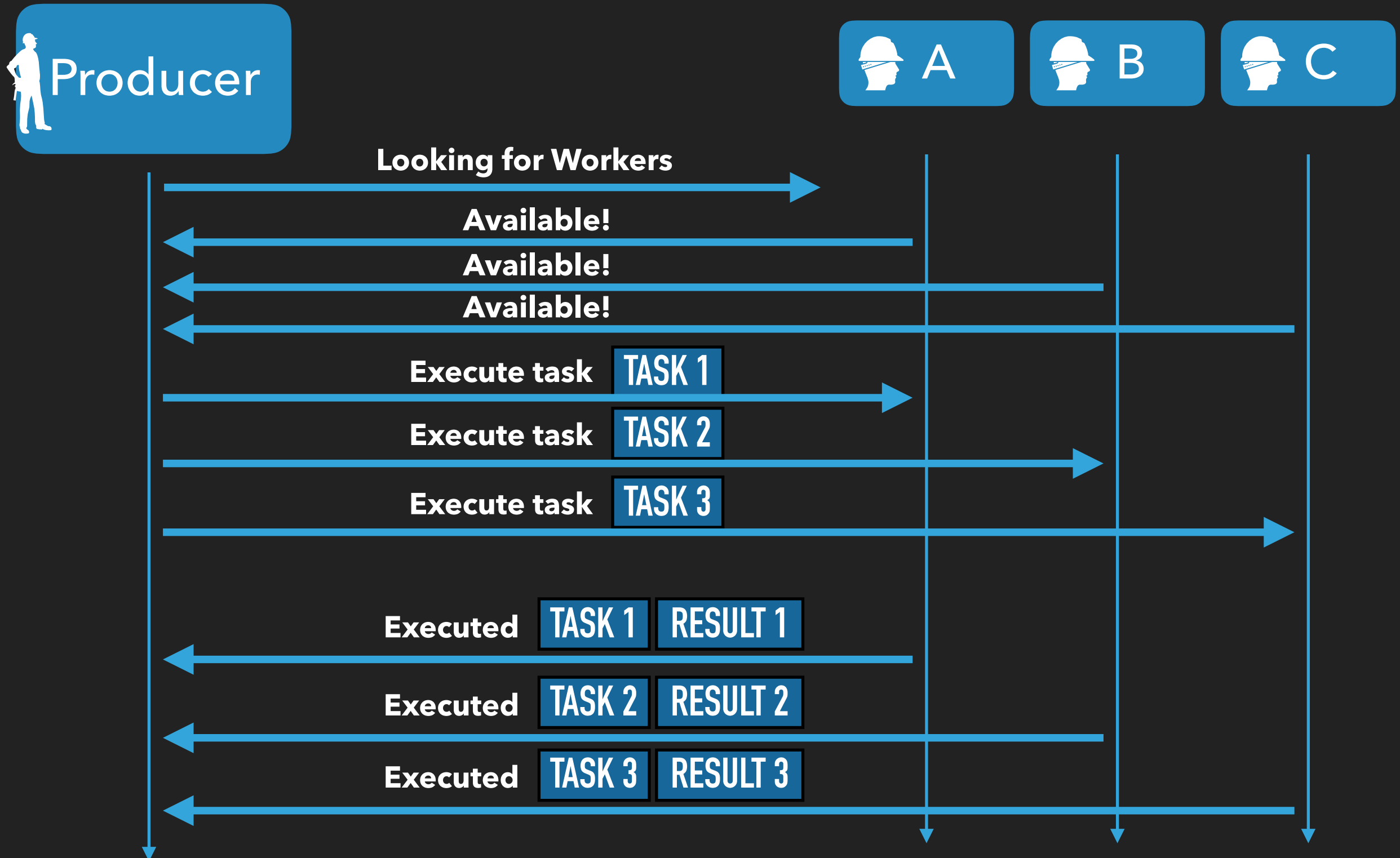
TEAMTALK

<https://github.com/quentinplessis/Teamtalk>

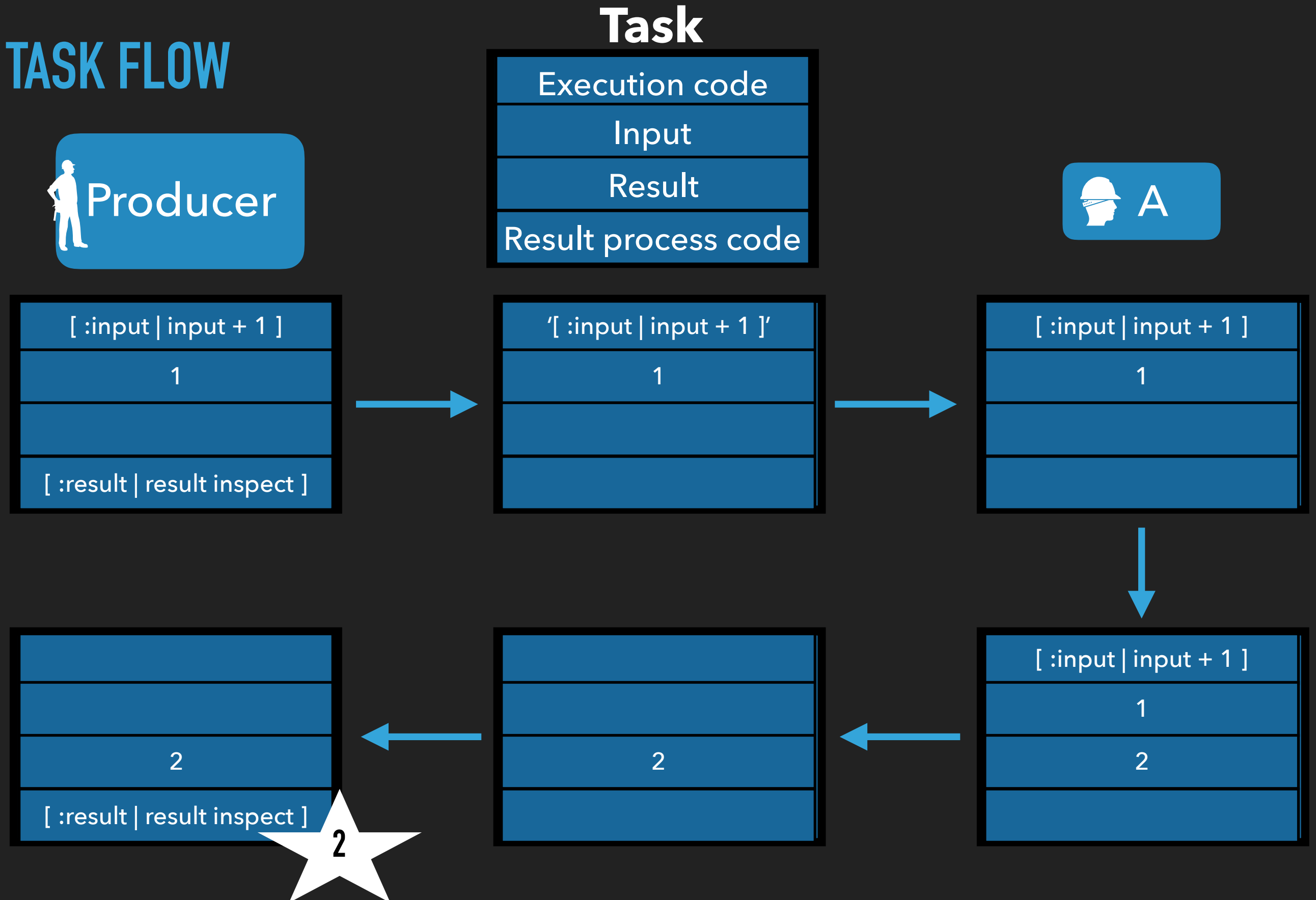
- Pharo Smalltalk library making it possible to execute tasks in a cluster of Pharos instances.



PROTOCOL



TASK FLOW



REMARKS

- ▶ Support multiple producers
- ▶ Workers can be dynamically added / removed
- ▶ Pharo instances can be on different machines and use different Pharo versions
- ▶ All communications are broadcasted throughout the cluster
- ▶ Security issues are present but currently not dealt with

GETTING STARTED

- ▶ Install Teamtalk in two Pharos images A and B

```
Metacello new
  baseline: 'Teamtalk';
  repository: 'github://quentinplessis/Teamtalk/pharo-repository';
  load.
```

- ▶ Setup Teamtalk server (docker must be installed and running)

```
docker run -p 8080:8080 plequen/teamtalk-server
```

- ▶ [Pharo Image A] Create a producer

```
producer := TTProducer host: 'localhost' port: 8080.
```

- ▶ [Pharo Image B] Create a worker

```
worker := TTWorker host: 'localhost' port: 8080.
```

- ▶ [Pharo Image A] Add a task to execute

```
task := TTTask
  executionCode: [ :input |
    input + 1
  ]
  resultProcessCode: [ :result |
    result inspect.
  ]
  input: 1.
producer addTask: task.
```

TASK EXAMPLES

<https://github.com/quentinplessis/Teamtalk>

```
producer addTask: (TTTask executionCode: [ 1 inspect. 5 seconds wait. nil ]).
producer addTask: (TTTask executionCode: [ 2 inspect. 5 seconds wait. nil ]).
producer addTask: (TTTask executionCode: [ 3 inspect. 5 seconds wait. nil ]).
producer addTask: (TTTask executionCode: [ 4 inspect. 5 seconds wait. nil ]).
```

```
start := DateAndTime now.
number := 4.
i := 0.
number timesRepeat: [
    task := TTTask
        executionCode: [
            10 seconds wait.
            'OK' inspect.
            'OK'
        ]
        resultProcessCode: [ :result |
            i := i + 1.
            i = number ifTrue: [
                (DateAndTime now asUnixTime - start asUnixTime) inspect
            ].
        ].
    producer addTask: task.
].
```

```
task := TTTask
    executionCode: [
        ZnClient new get: 'https://www.google.com/'.
    ]
    resultProcessCode: [ :result |
        result inspect.
    ].
producer addTask: task.
```

CLUSTER MANAGEMENT <https://github.com/quentinplessis/Teamtalk#cluster-setup>

requires docker and ruby

- ▶ Install Teamtalk in a Pharo image Teamtalk.image

- ▶ Setup a Teamtalk server on port 8080

```
export TEAMTALK_SERVER_PORT=8080
ruby spawner.rb --create-server $TEAMTALK_SERVER_PORT
ifconfig | grep inet
export TEAMTALK_SERVER_IP=.....
```

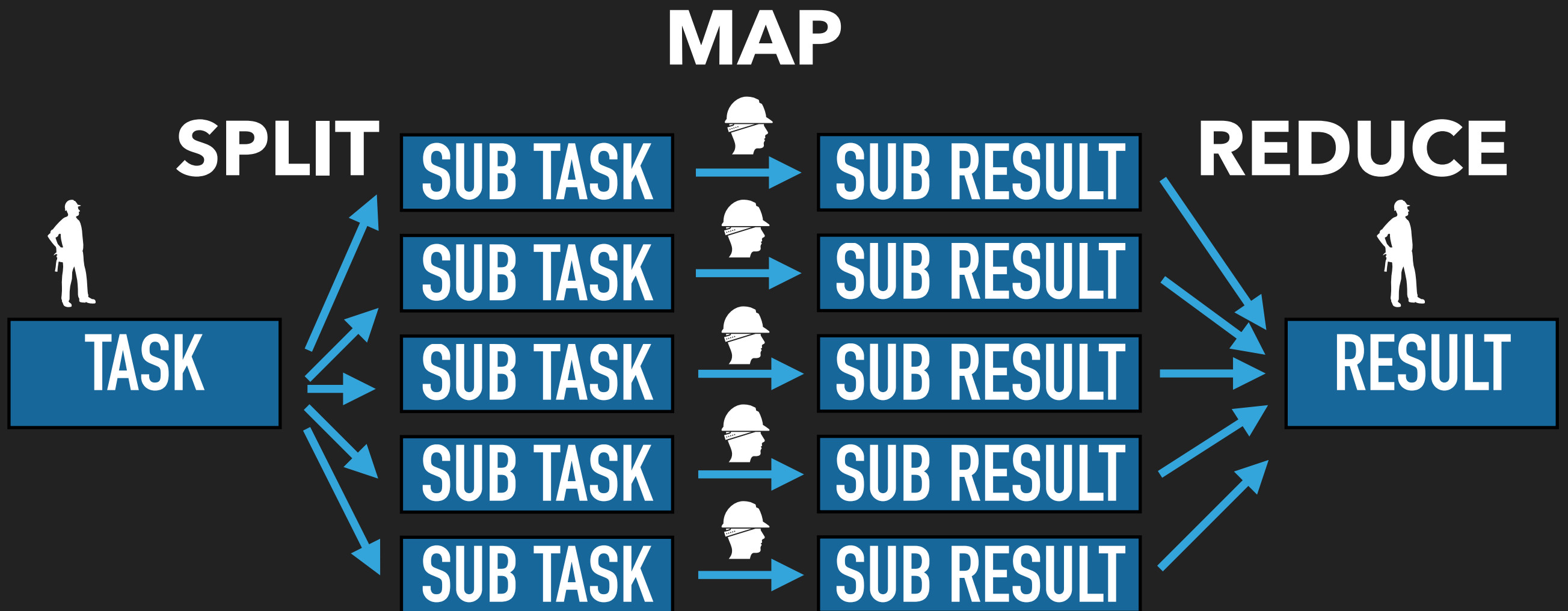
- ▶ Add consumers, potentially on different machines

```
ruby spawner.rb --create-consumer --pharo-image ./Teamtalk.image --server-host
$TEAMTALK_SERVER_IP --server-port $TEAMTALK_SERVER_PORT
ruby spawner.rb --create-consumer --pharo-image ./Teamtalk.image --server-host
$TEAMTALK_SERVER_IP --server-port $TEAMTALK_SERVER_PORT
ruby spawner.rb --create-consumer --pharo-image ./Teamtalk.image --server-host
$TEAMTALK_SERVER_IP --server-port $TEAMTALK_SERVER_PORT
```

- ▶ Add a producer

```
ruby spawner.rb --create-producer --pharo-image ./Teamtalk.image --server-host
$TEAMTALK_SERVER_IP --server-port $TEAMTALK_SERVER_PORT
```

MAP REDUCE



MAP REDUCE WITH TEAMTALK

```
mapReduce := (TTMapReduce
  splitBlockForInput: [ :input :tasksNumber |
    "split input into sub inputs here"
  ]
  mapBlockForSubInput: [ :subInput |
    "process sub inputs here"
    "map sub input to sub result"
  ]
  reduceBlockWithCallback: [ :results :callback |
    "reduce sub results here"
    callback value: results
  ])
  ttClientClass: TTProducer;
  host: 'localhost';
  port: 8080;
  yourself.

mapReduce
  input: { }
  tasksNumber: 4
  callbackDo: [ :result | result inspect ].
```

► Examples: <https://github.com/quentinplessis/Teamtalk#mapreduce>

IMPROVEMENTS

- ▶ Add labels to tasks for selective worker selection
- ▶ Propagate error handling from worker to producer
- ▶ Improve security
- ▶ Improve protocol to support additional languages
- ▶ Add a vote-based task scheduling system to remove the need for producers to manage their tasks
- ▶ Add mechanism to handle loss of worker during task execution
- ▶ ...

TEAMTALK