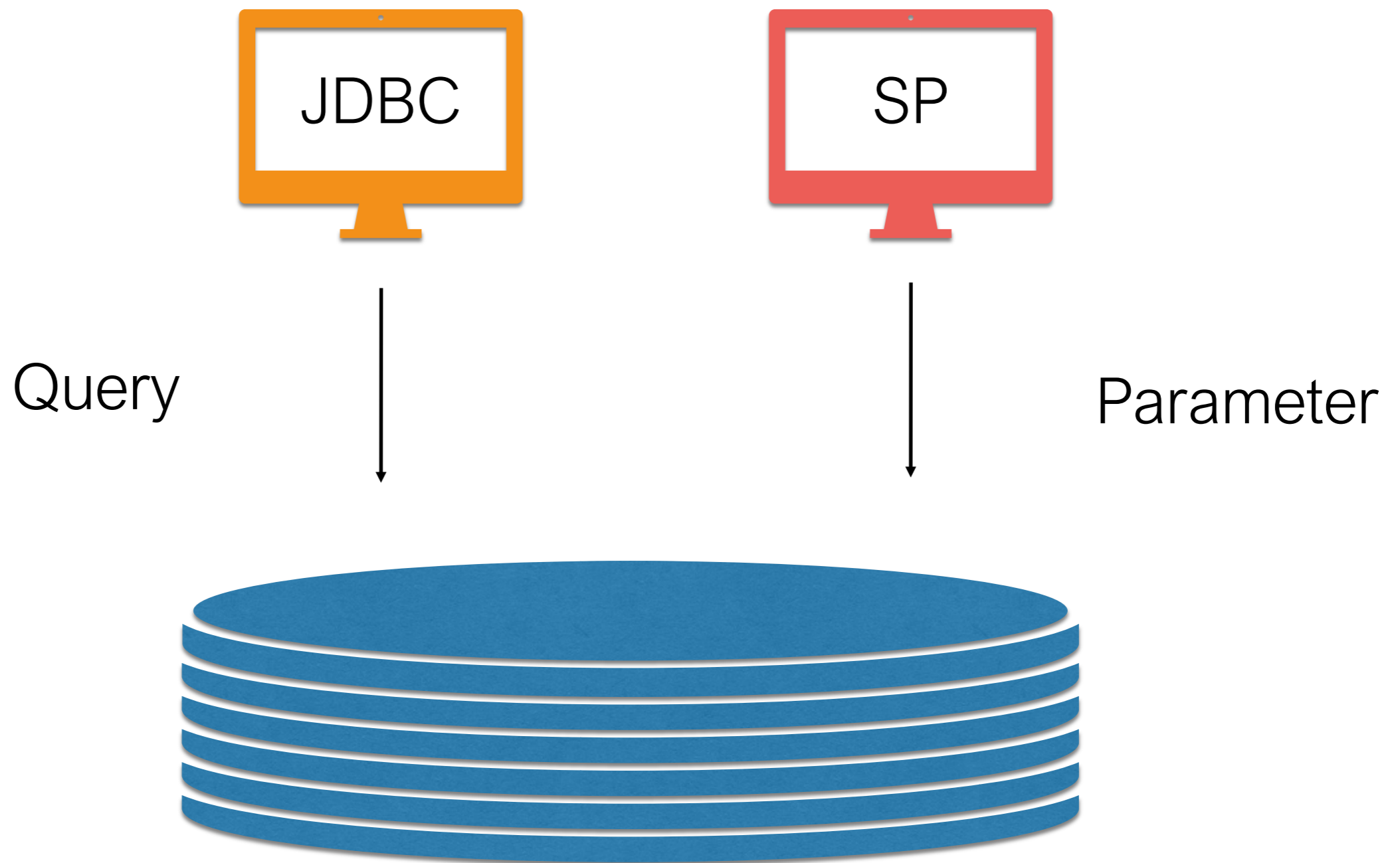


Introduction to Benchmark

Database Systems
DataLab, CS, NTHU
Spring, 2020

JDBC / SP ?



Outline

- VanillaBench Project
 - Introduction to VanillaBench
 - Starting Up Server for Benchmarking
 - Setting Benchmark Configurations
 - Running Benchmark Client
- VanillaBench

Outline

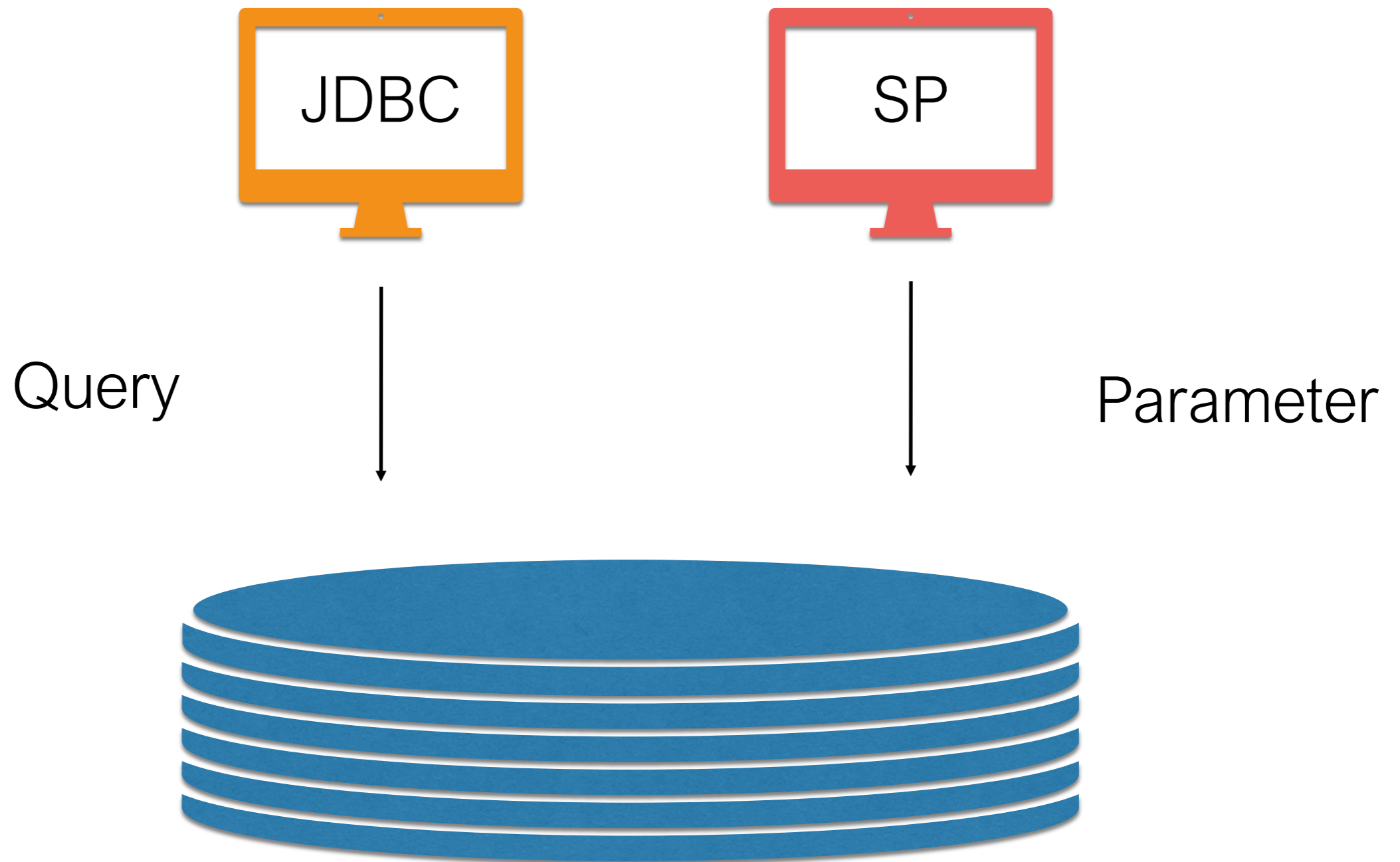
- **VanillaBench Project**
 - **Introduction to VanillaBench**
 - Starting Up Server for Benchmarking
 - Setting Benchmark Configurations
 - Running Benchmark Client
 - VanillaBench

VanillaBench

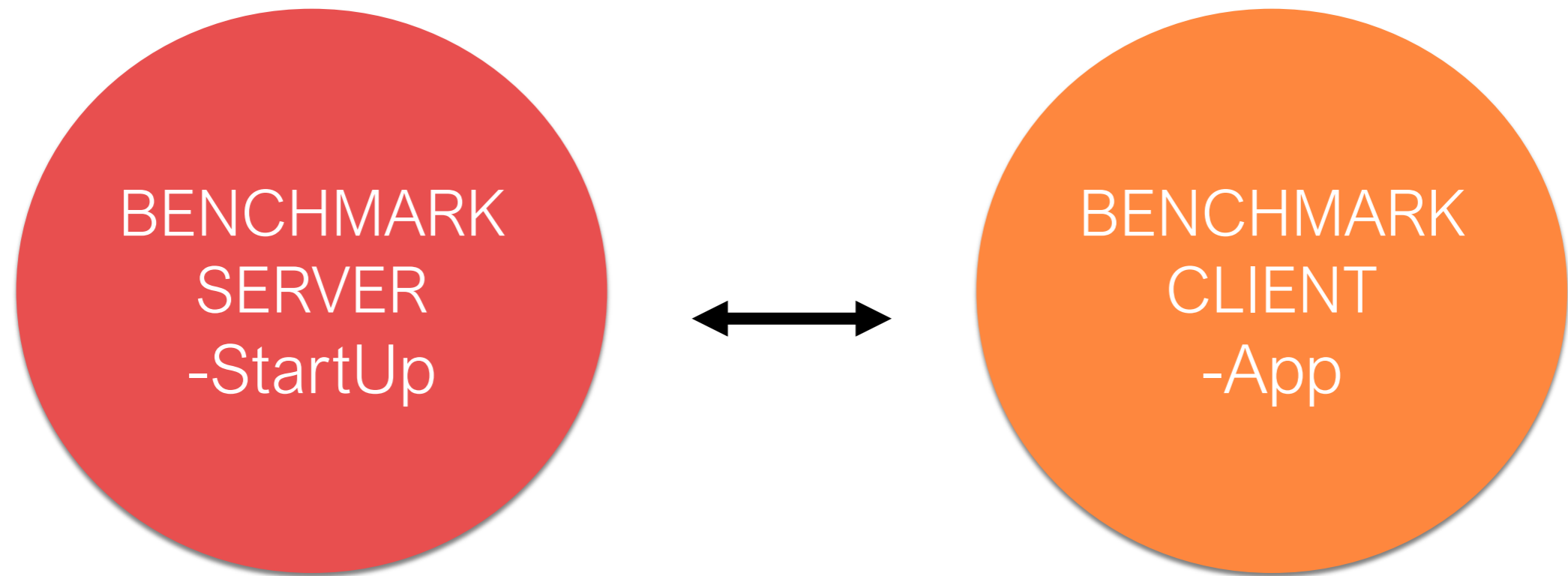


- VanillaBench is a project designed for automatically benchmarking VanillaCore
- It contains several benchmark procedures
- It also has a lot of adjustable testing parameters

JDBC / SP ?



Two Main Methods



Outline

- VanillaBench Project
 - Introduction to VanillaBench
 - Starting Up Server for Benchmarking
 - Setting Benchmark Configurations
 - Running Benchmark Client
- VanillaBench

Clone the Project First

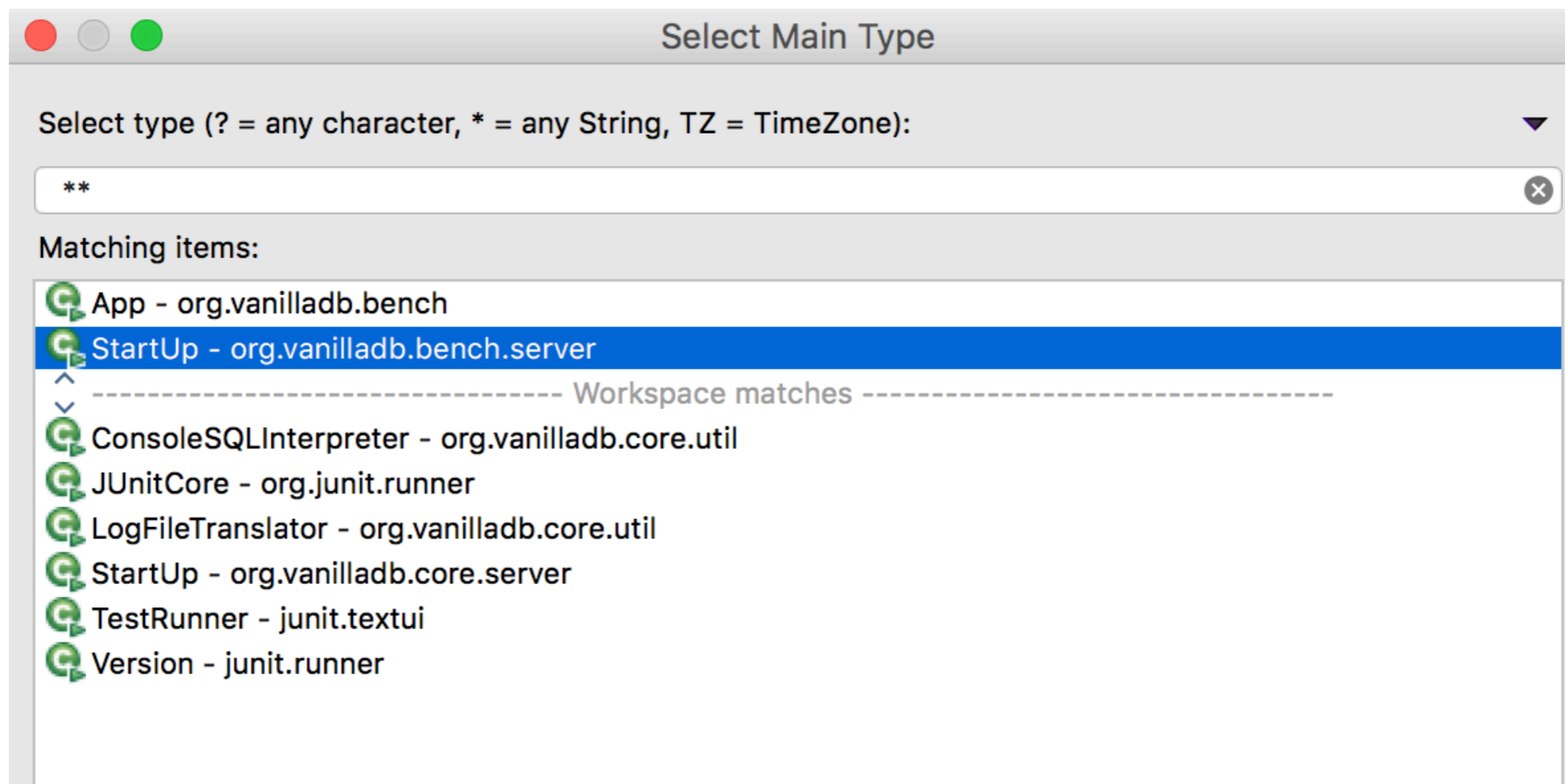
- The code of VanillaBench has been pushed to vanilladb repository
- All you need is to clone from the remote repository

```
> git clone
```

- You can clone from here:
 - <https://shwu10.cs.nthu.edu.tw/courses/databases/2020-spring/db20-assignment-2>

Starting Up Server (1/2)

- To benchmark a VanillaDB server, you need to start up the server in another entry point



Starting Up Server (2/2)

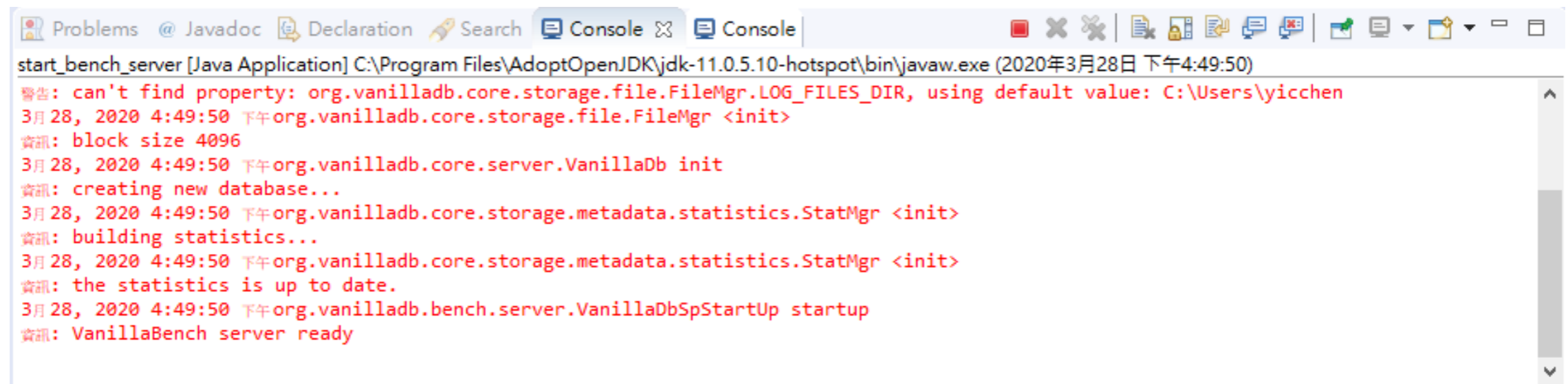
- You also need to add one more VM argument for benchmarking.
- Don't forget to add the Database Directory Name

```
VM arguments:  
-Djava.util.logging.config.file=target/classes/java/util/logging/logging.properties  
-Dorg.vanilladb.bench.config.file=target/classes/org/vanilladb/bench/vanillabench.properties  
-Dorg.vanilladb.core.config.file=target/classes/org/vanilladb/core/vanilladb.properties
```

Variables...

You can copy those arguments from [here](#)

Server Messages



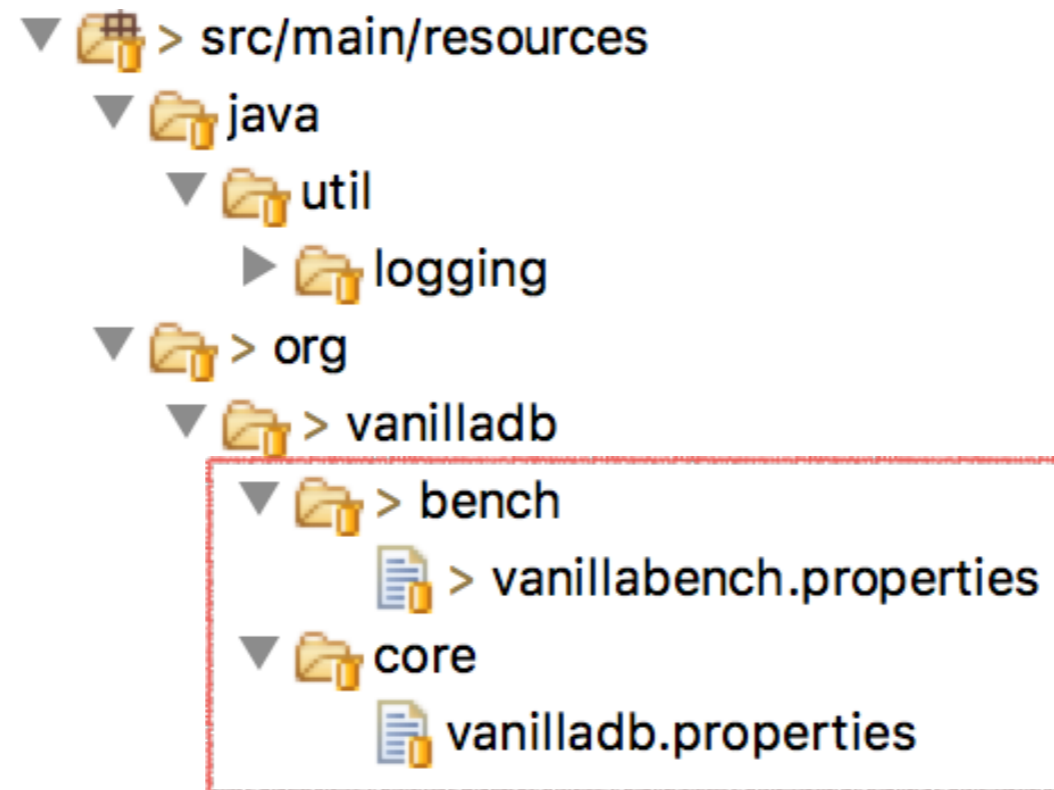
The screenshot shows an IDE console window with the following text:

```
start_bench_server [Java Application] C:\Program Files\AdoptOpenJDK\jdk-11.0.5.10-hotspot\bin\javaw.exe (2020年3月28日 下午4:49:50)
警告: can't find property: org.vanilladb.core.storage.file.FileMgr.LOG_FILES_DIR, using default value: C:\Users\yicchen
3月 28, 2020 4:49:50 下午 org.vanilladb.core.storage.file.FileMgr <init>
资讯: block size 4096
3月 28, 2020 4:49:50 下午 org.vanilladb.core.server.VanillaDb init
资讯: creating new database...
3月 28, 2020 4:49:50 下午 org.vanilladb.core.storage.metadata.statistics.StatMgr <init>
资讯: building statistics...
3月 28, 2020 4:49:50 下午 org.vanilladb.core.storage.metadata.statistics.StatMgr <init>
资讯: the statistics is up to date.
3月 28, 2020 4:49:50 下午 org.vanilladb.bench.server.VanillaDbSpStartup startup
资讯: VanillaBench server ready
```

You should see similar messages if there is nothing wrong.

Setting Benchmark

- Benchmark project also has its own set of properties files



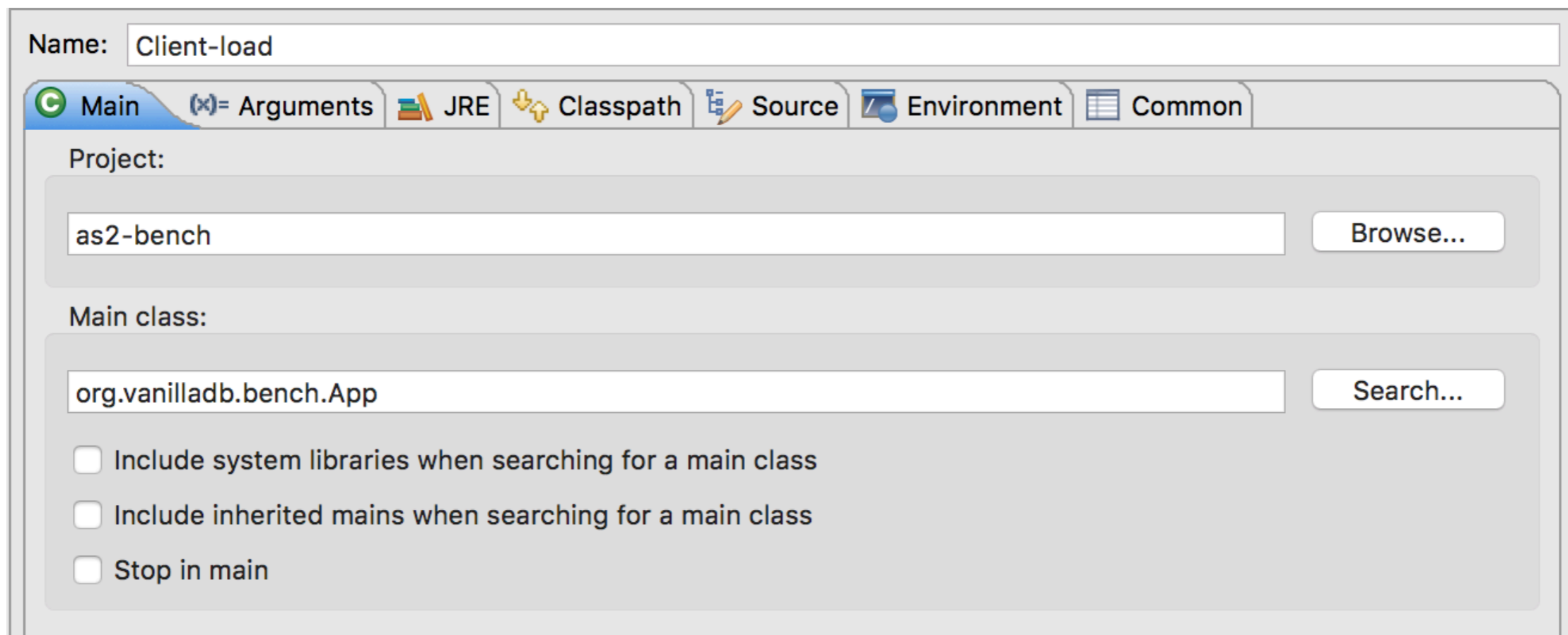
```
1
2 #
3 # Basic Parameters
4 #
5
6 # The running time for warming up before benchmarking
7 org.vanilladb.bench.BenchmarkParameters.WARM_UP_INTERVAL=3000
8 # The running time for benchmarking
9 org.vanilladb.bench.BenchmarkParameters.BENCHMARK_INTERVAL=6000
10 # The number of remote terminal executors for benchmarking
11 org.vanilladb.bench.BenchmarkParameters.NUM_RTES=10
12 # The IP of the target database server
13 org.vanilladb.bench.BenchmarkParameters.SERVER_IP=127.0.0.1
14 # 1 = JDBC, 2 = Stored Procedures
15 org.vanilladb.bench.BenchmarkParameters.CONNECTION_MODE=2
16 # The path to the generated reports
17 org.vanilladb.bench.StatisticMgr.OUTPUT_DIR=
18 # Whether the RTEs display the results of each transaction
19 org.vanilladb.bench.rte.TransactionExecutor.DISPLAY_RESULT=false
20 # The number of items in the testing data set
21 org.vanilladb.bench.as2.As2Constants.NUM_ITEMS=100000
```

Important Parameters

- CONNECTION_MODE
 - It means whether it should use JDBC or stored procedures

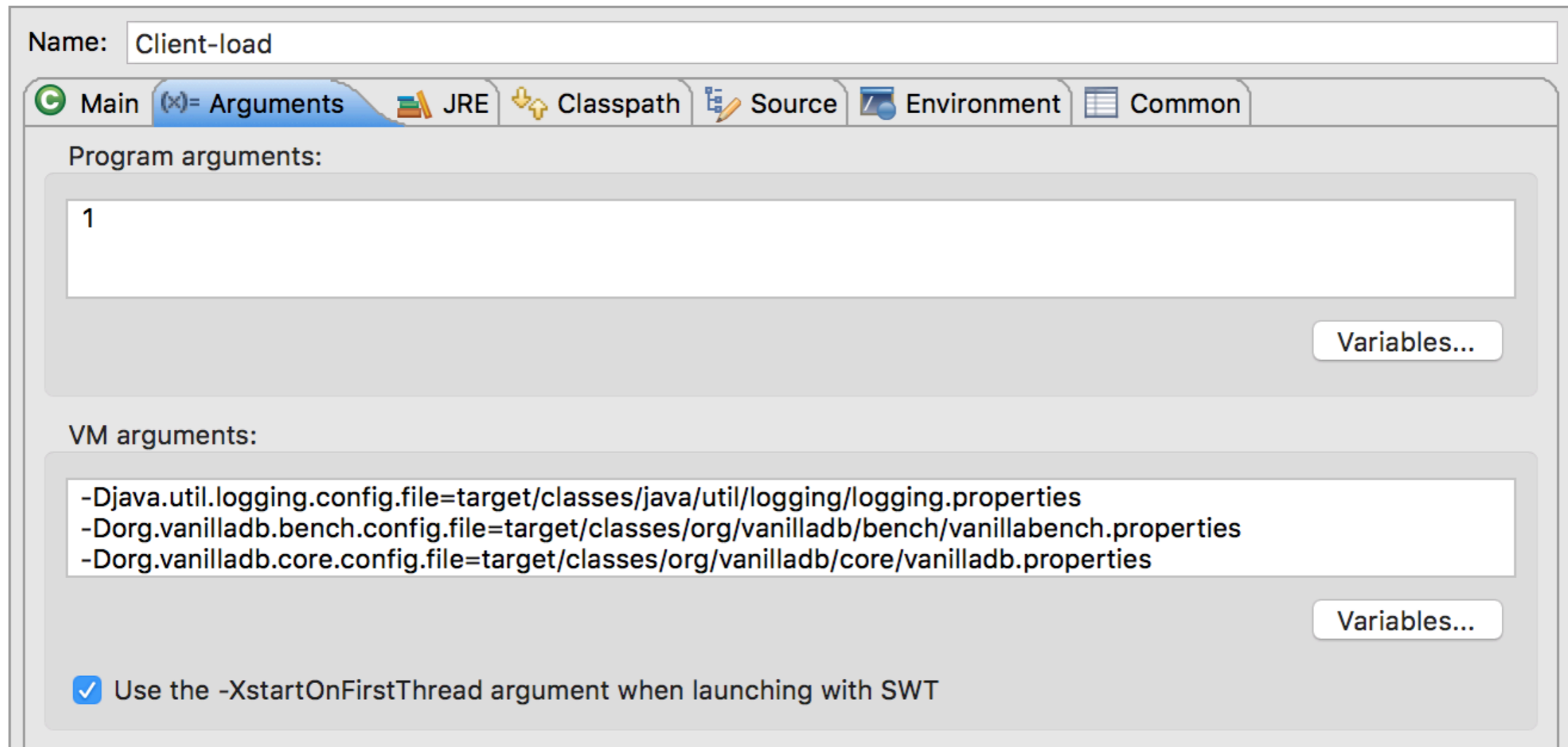
Running Client

- To run clients, create a run configuration for it



Arguments

- We also need to set some arguments
- Program Arguments
 - [Action]
 - 1 : Load Test-Bed
 - 2 : Lunch Benchmark
- VM Arguments
 - Use the same as the server

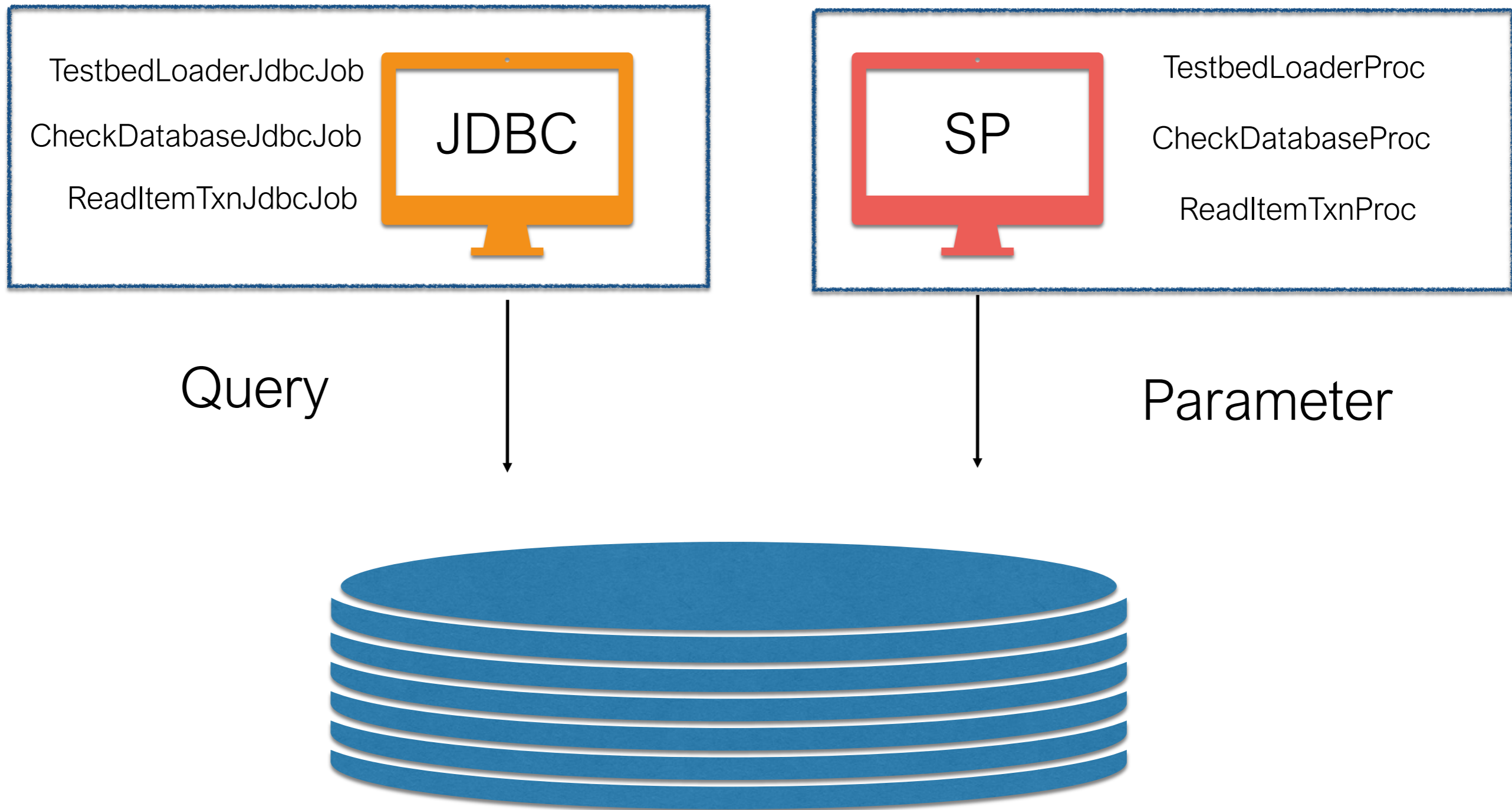


You can copy those arguments from [here](#), then click 'Apply' and 'Run'

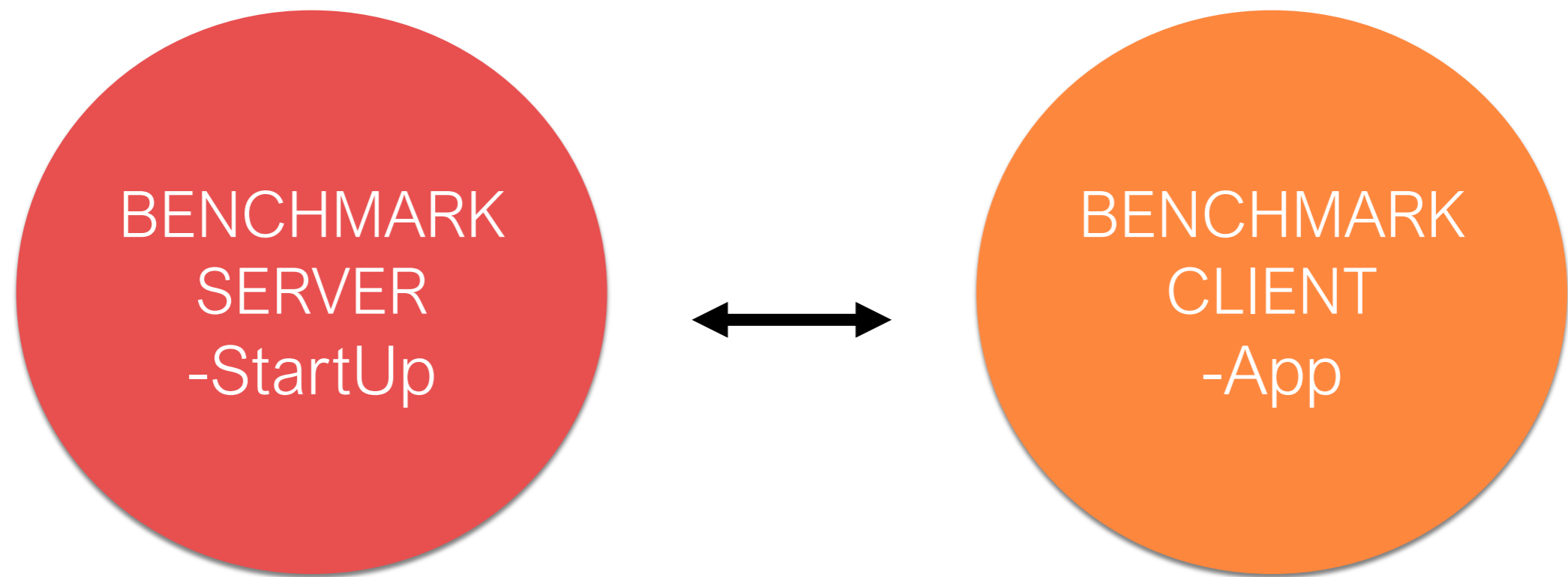
Notes

- Before running testing, please **load testbed** first
- Check if the output directory path that show in properties file exists
- If not, create one or change the path

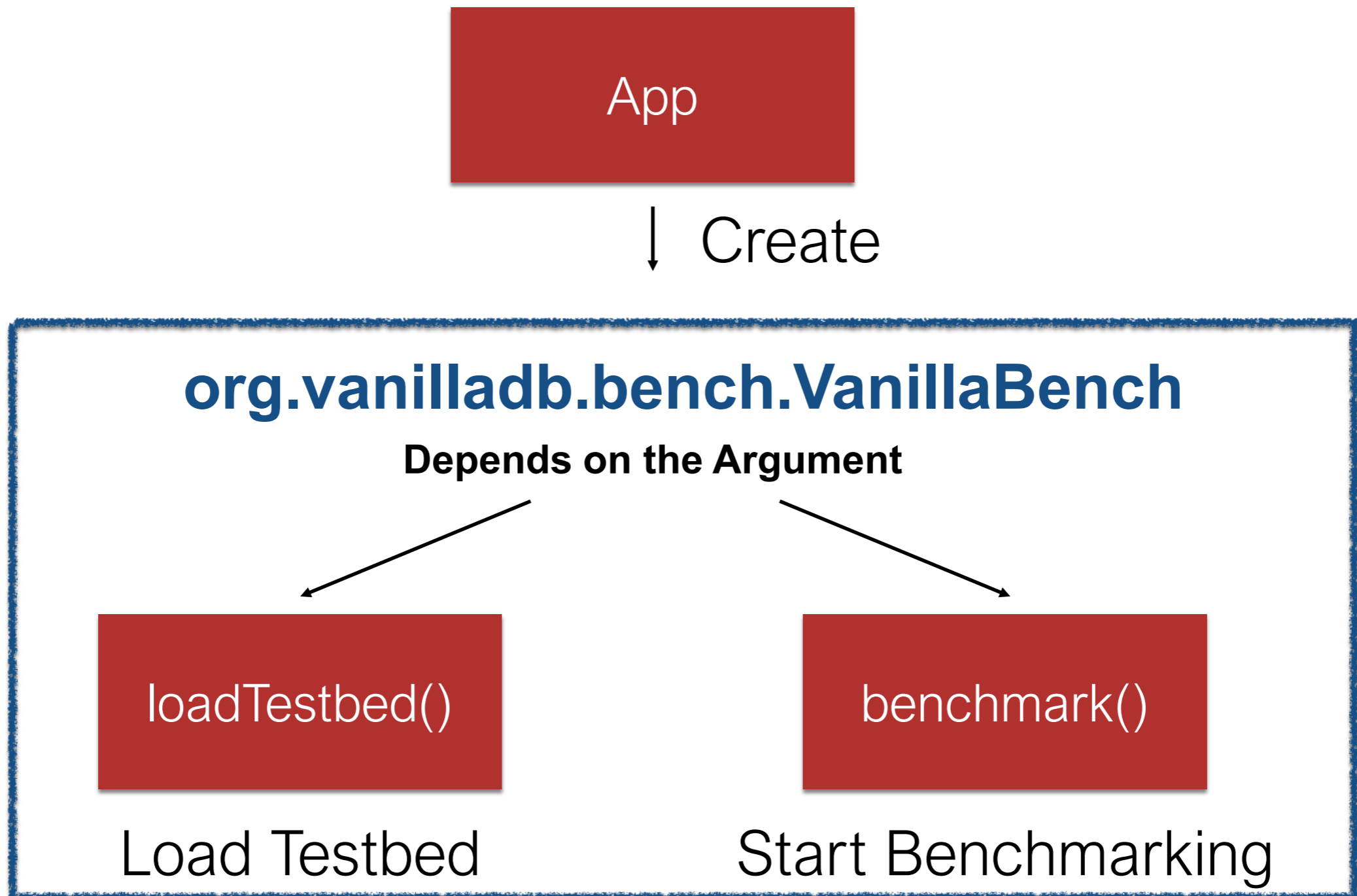
JDBC / SP ?



Two Main Methods



The Workflow of A Client



Loading Testbed



loadTestbed()

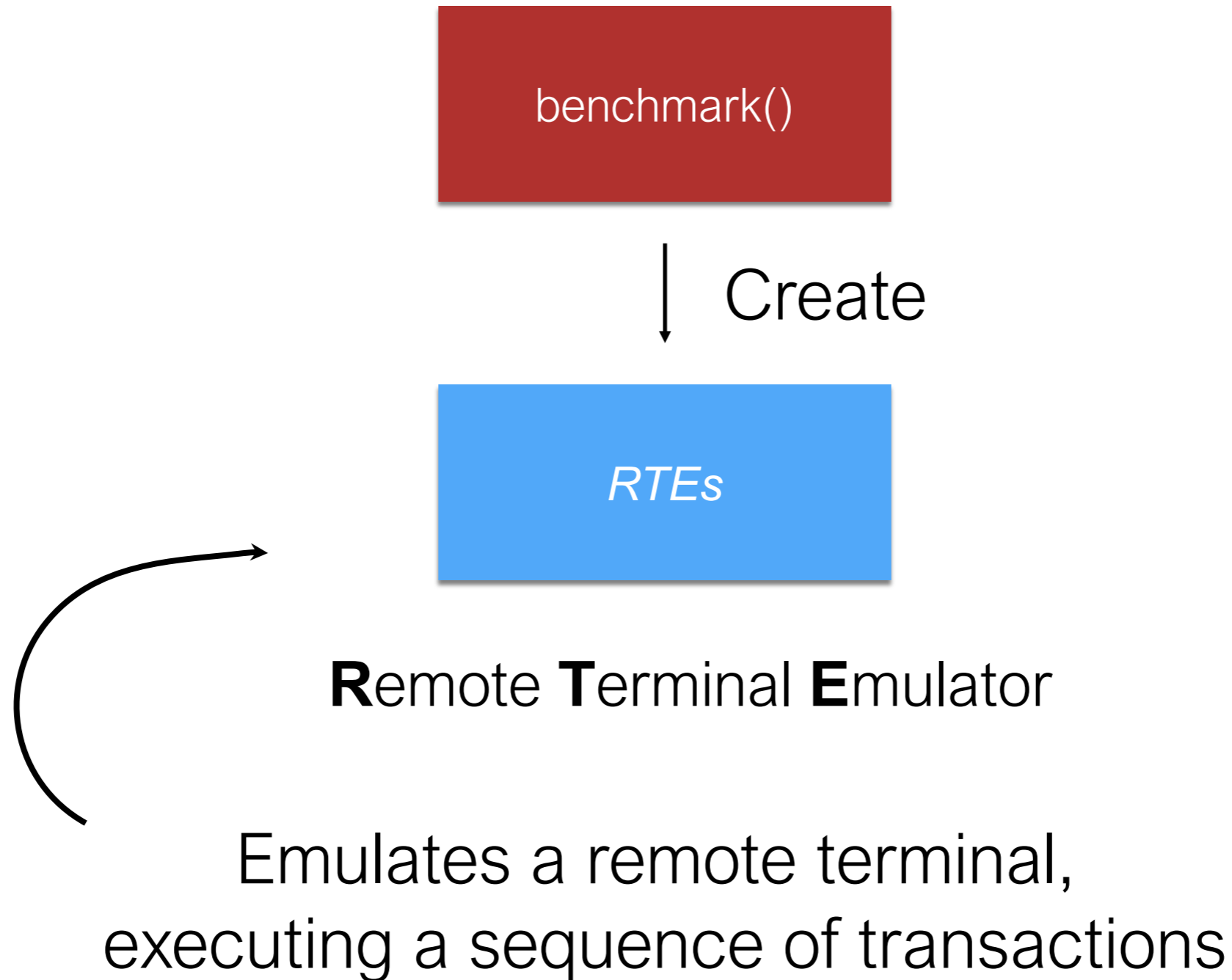


Connect to server and execute:

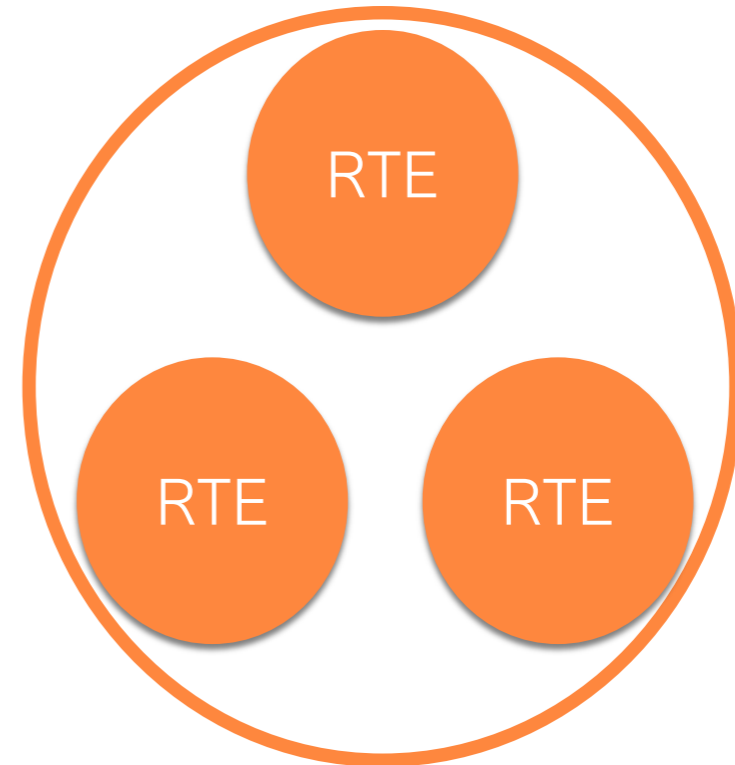
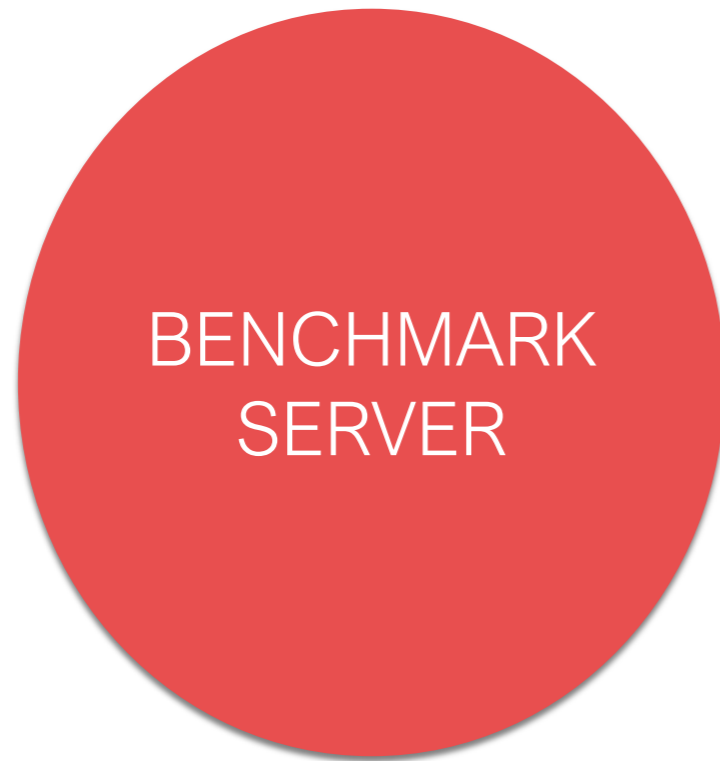


TestbedLoader

Starting Benchmark

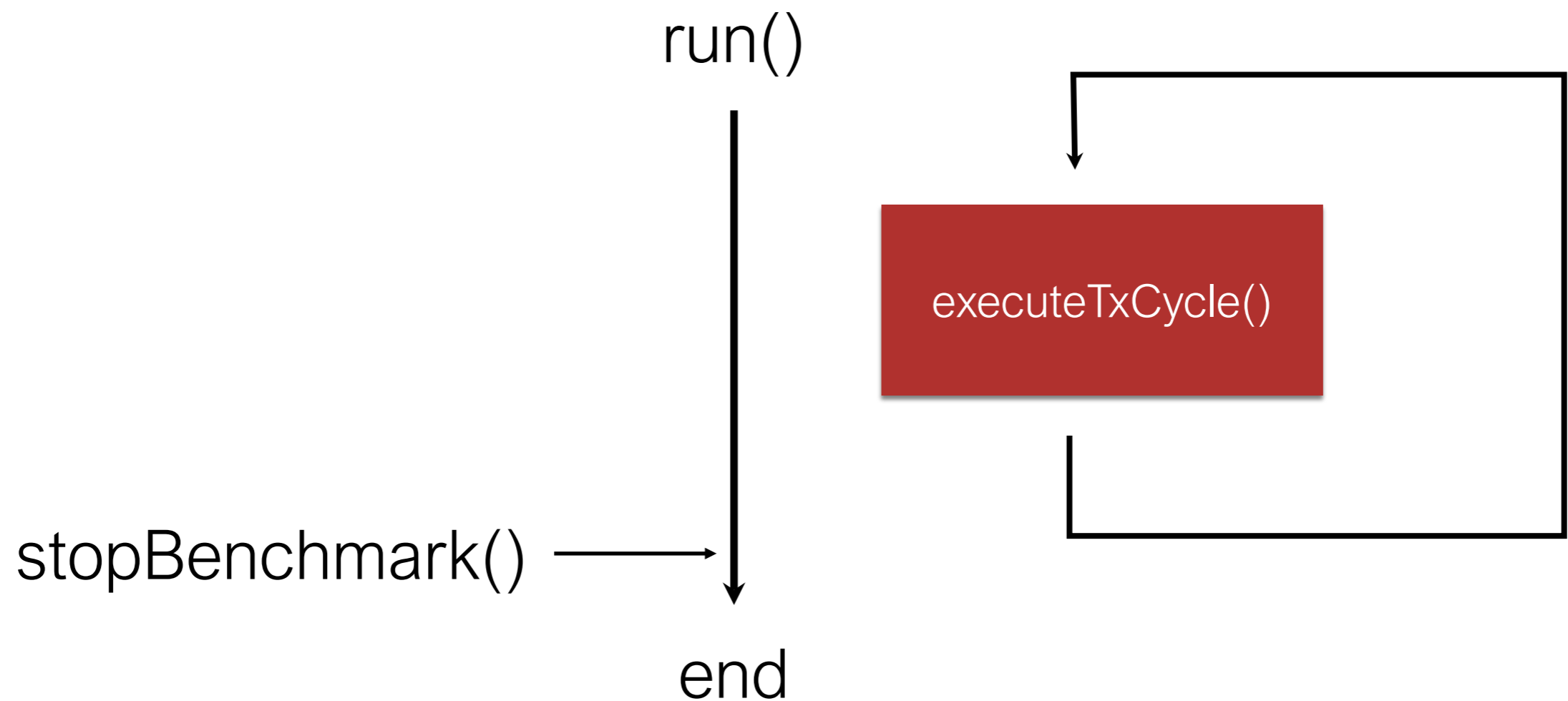


Server & Client

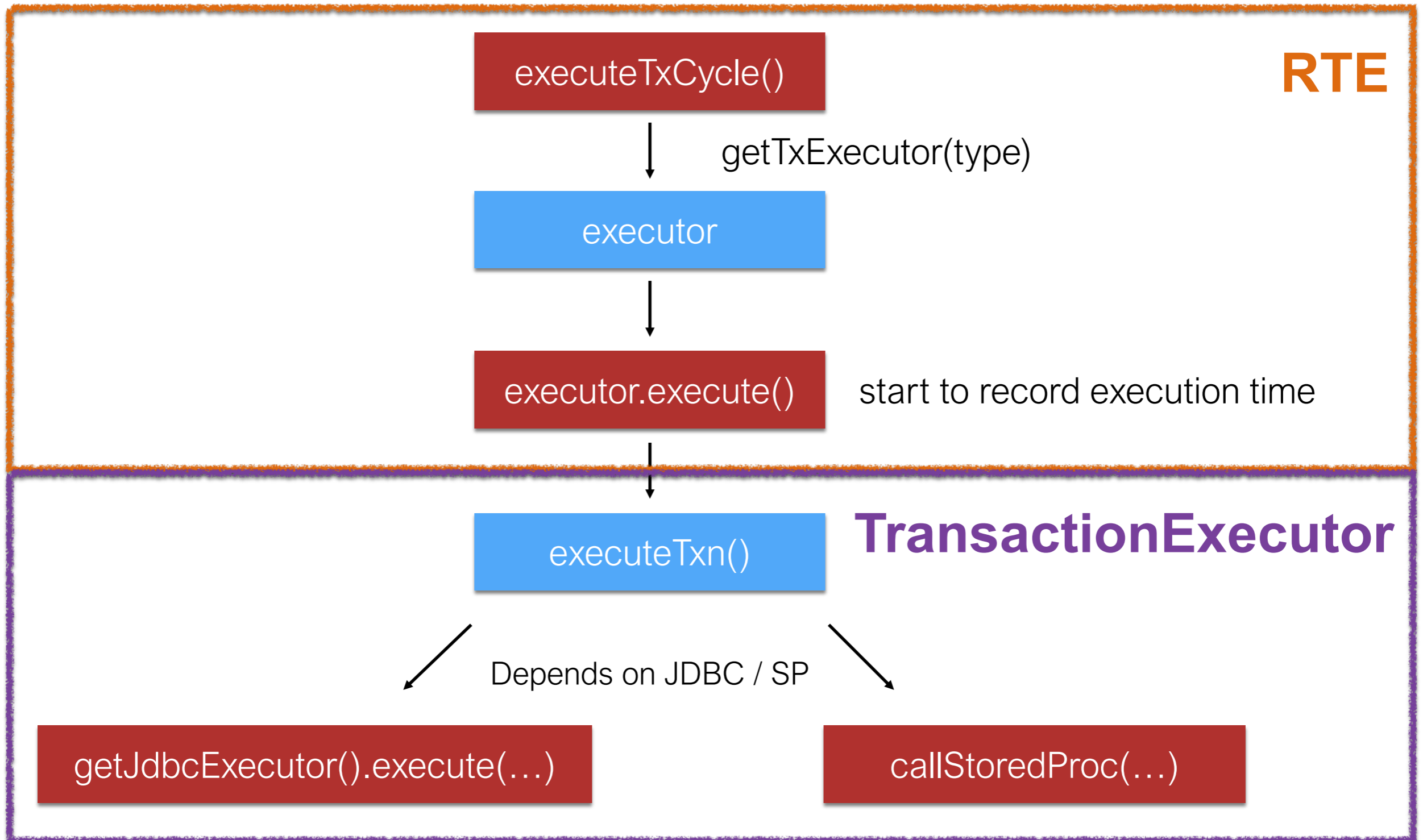


RTE's Life Cycle

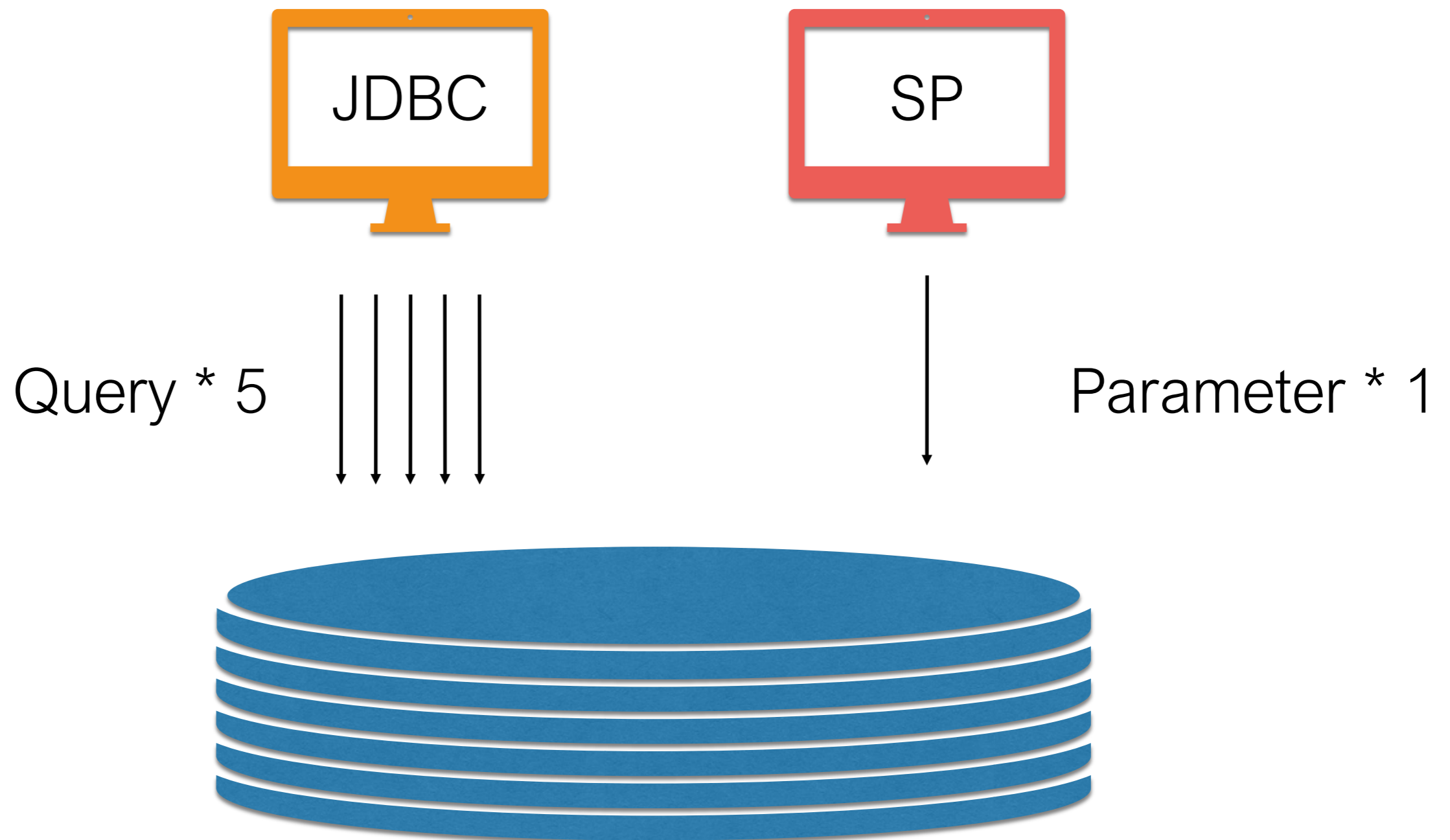
org.vanilladb.bench.rte.RemoteTerminalEmulator



Executing a Tx



JDBC / SP ?



Workflow of Executing a Tx

- General steps
 - Generate parameters from TxParamGenerator
- Store Procedure
 - **callStoredProc(...)** to execute a stored procedure on the **remote server**, with the generated parameters
 - Remote server will return a **SutResultSet** when the procedure is finished
- JDBC
 - **getJdbcExecutor().execute(...)** to execute a JDBC Job in the **local**, with the generated parameters
 - Job will be executing each sql via JDBC connector

How Server Process a StoredProc call ?

- When the server receive a remote procedure call, it will ask StoredProcFactory to generate the appropriate StoredProcedure
- The server will then call the StoredProcedure methods:
 - `prepare(Obj...)`
 - preparing the parameters
 - `execute()`
 - executing the transaction
 - this method will return the final result to the client

Q&A

- If you got any problem, you could check here first
- <https://shwu10.cs.nthu.edu.tw/courses/databases/2020-spring/faq>
- If your problem was very unique, just send us an email