University of Stuttgart
Germany

HPI Future SOC Lab Day

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Continuous Performance Testing for Microservices

Reliable Software Systems
Institute for Software Technologies
University of Stuttgart, Germany

Vincenzo Ferme
info@vincenzoferme.it

Outline

Outline

1. **Continuous Software Development Environments**

Outline

1. Continuous Software Development Environments
2. Continuous Performance Testing of Microservices:
   1. Challenges
   2. Our Approach
   3. Preliminary Results
   4. BenchFlow Automation Framework

Outline

1. Continuous Software Development Environments

2. Continuous Performance Testing of Microservices:
   a. Challenges
   b. Our Approach
   c. Preliminary Results
   d. BenchFlow Automation Framework

3. Approach Extensions:
   a. ContinuITy Framework

Outline

1. **Continuous Software Development Environments**

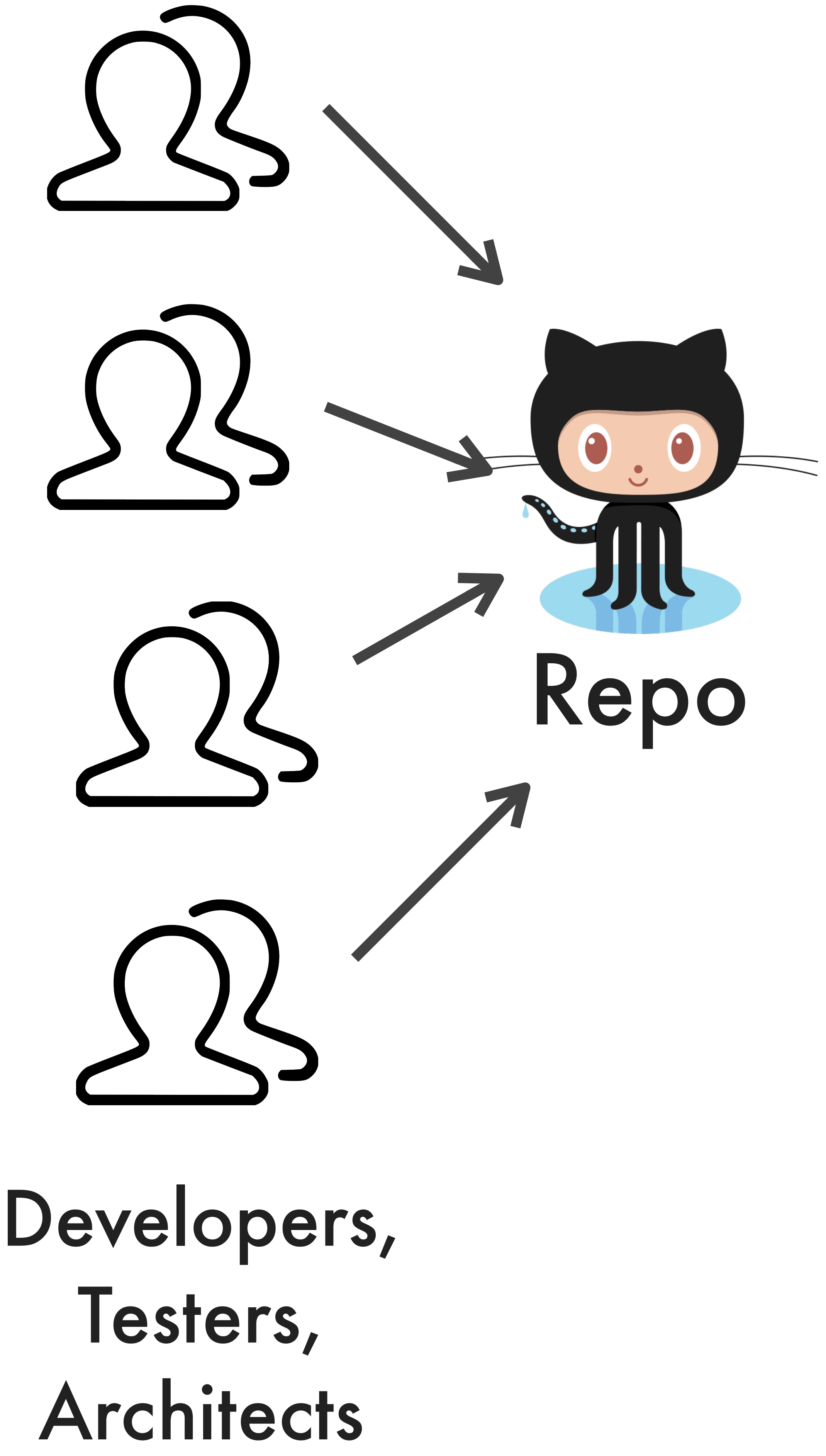
2. **Continuous Performance Testing of Microservices:**
   a. **Challenges**
   b. **Our Approach**
   c. **Preliminary Results**
   d. **BenchFlow Automation Framework**

3. **Approach Extensions:**
   a. **ContinuITy Framework**

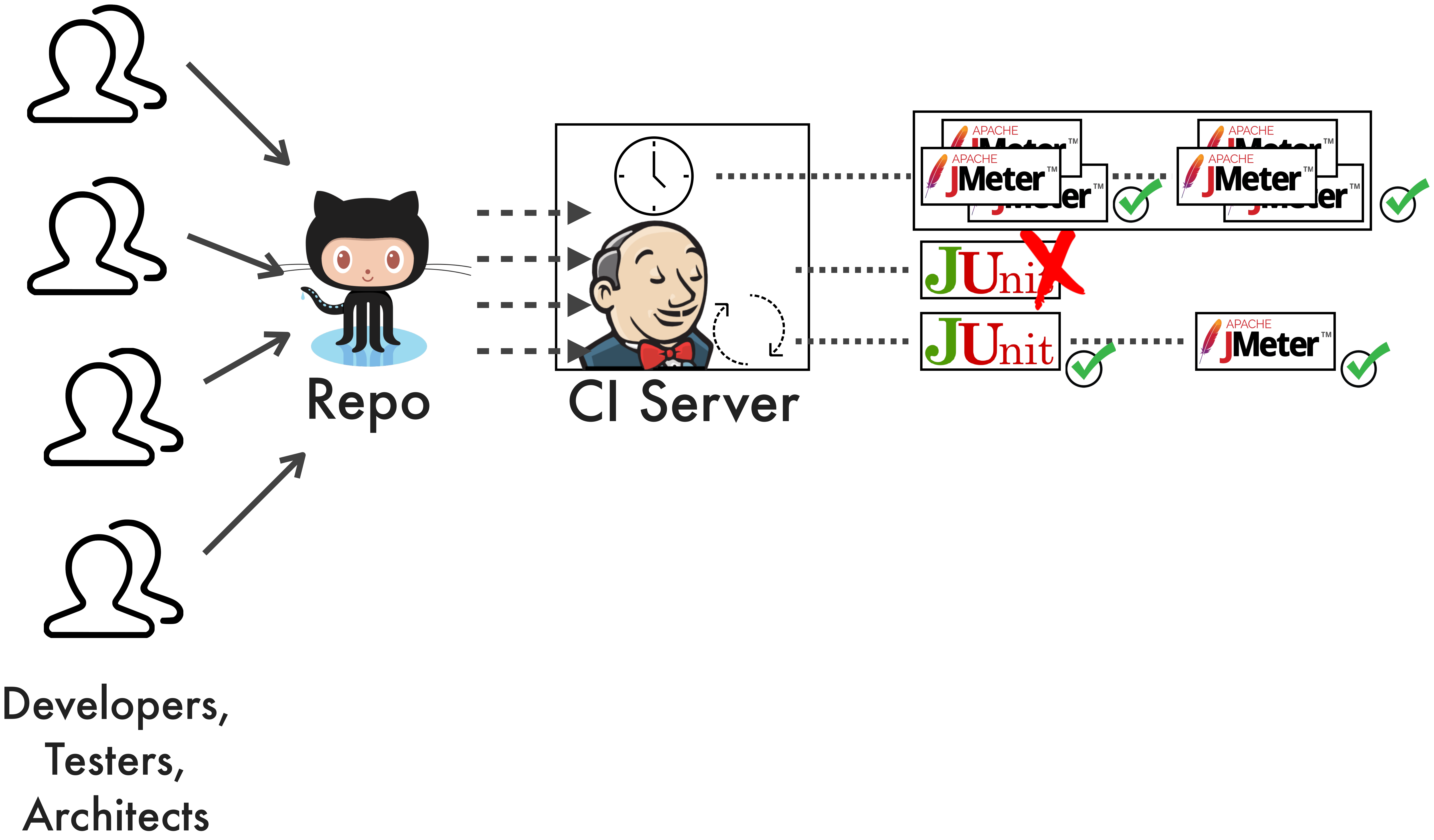
4. **Future Work**

Continuous Software Development Environments

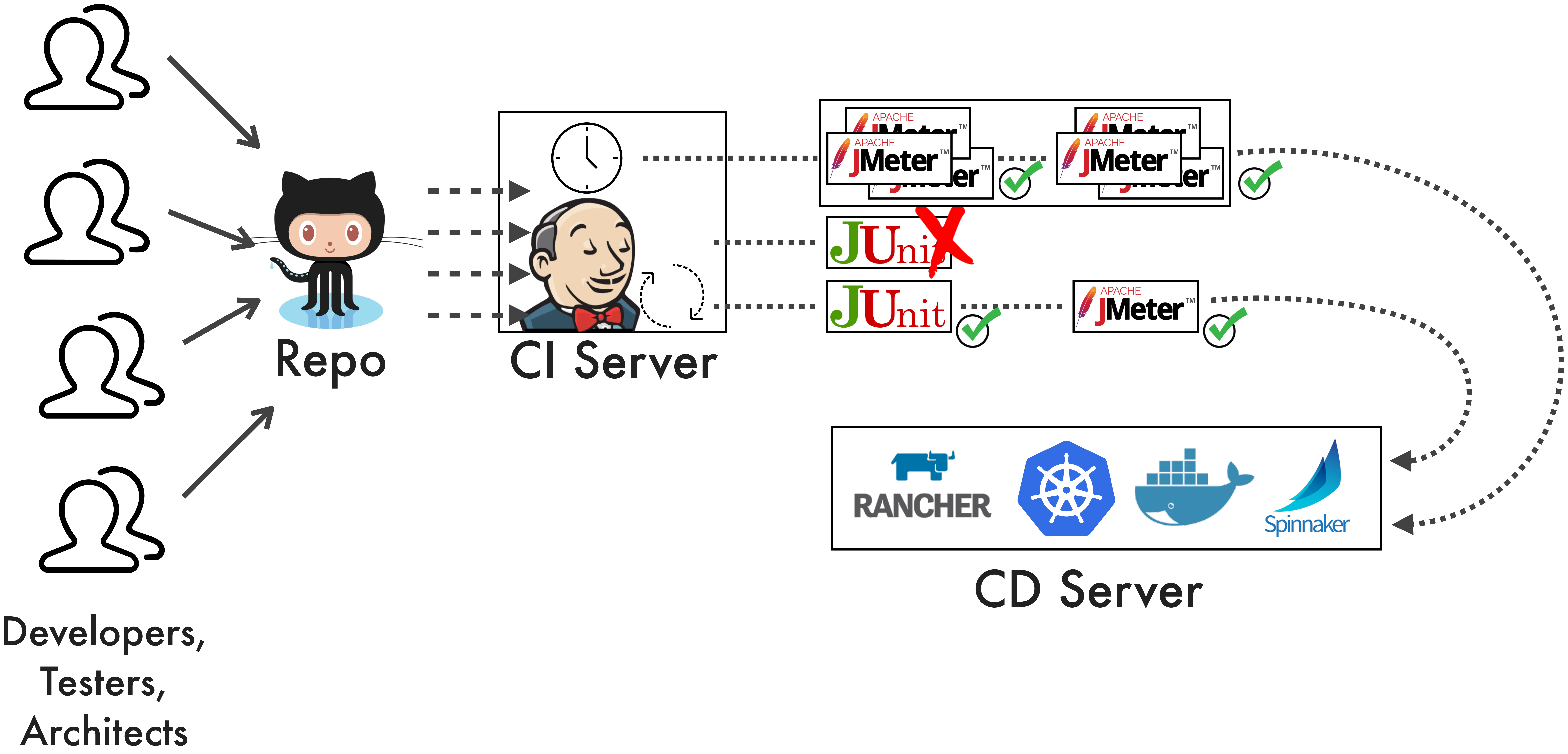
Continuous Software Development Environments



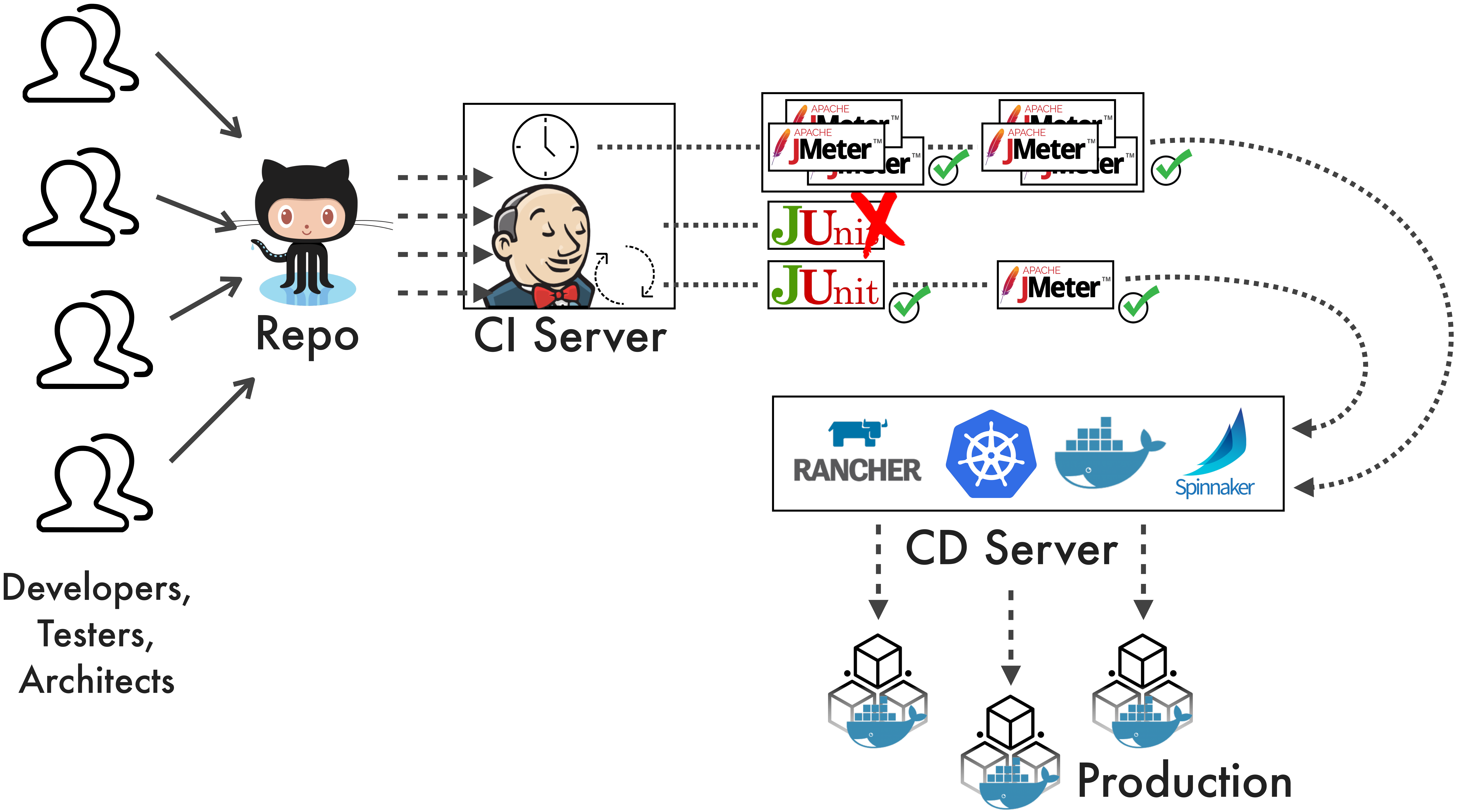
Continuous Software Development Environments



Continuous Software Development Environments



Continuous Software Development Environments



Challenges

Challenges

The system changes frequently:
- Need to keep the test definition updated (manual is not an option)

Challenges

The system changes frequently:
- Need to keep the test definition updated (manual is not an option)

Too many tests to be executed:
- Need to reduce the number of executed tests

Challenges

The system changes frequently:

- Need to keep the test definition updated (manual is not an option)

Too many tests to be executed:

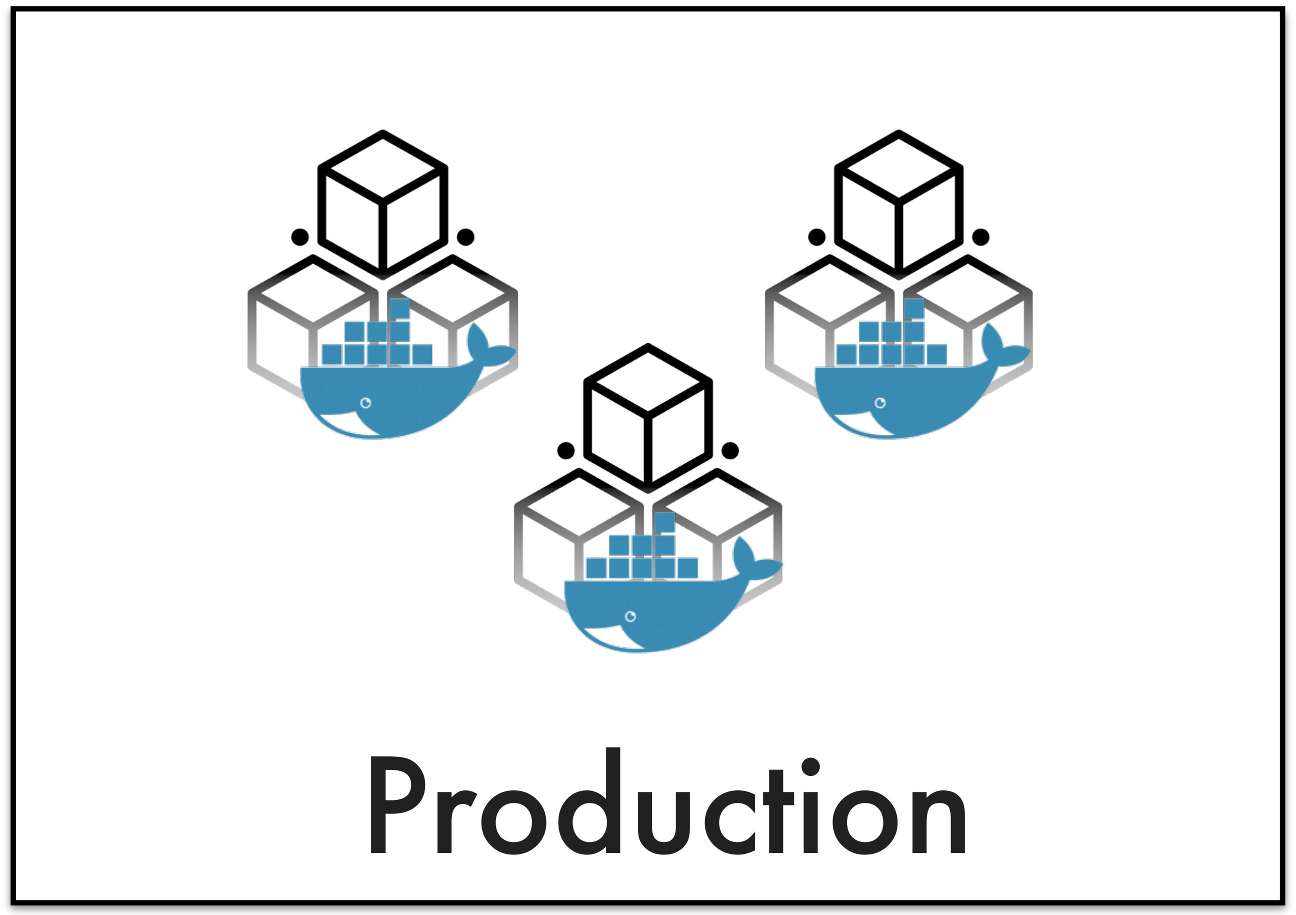
- Need to reduce the number of executed tests

Need to integrate tests in delivery pipelines:

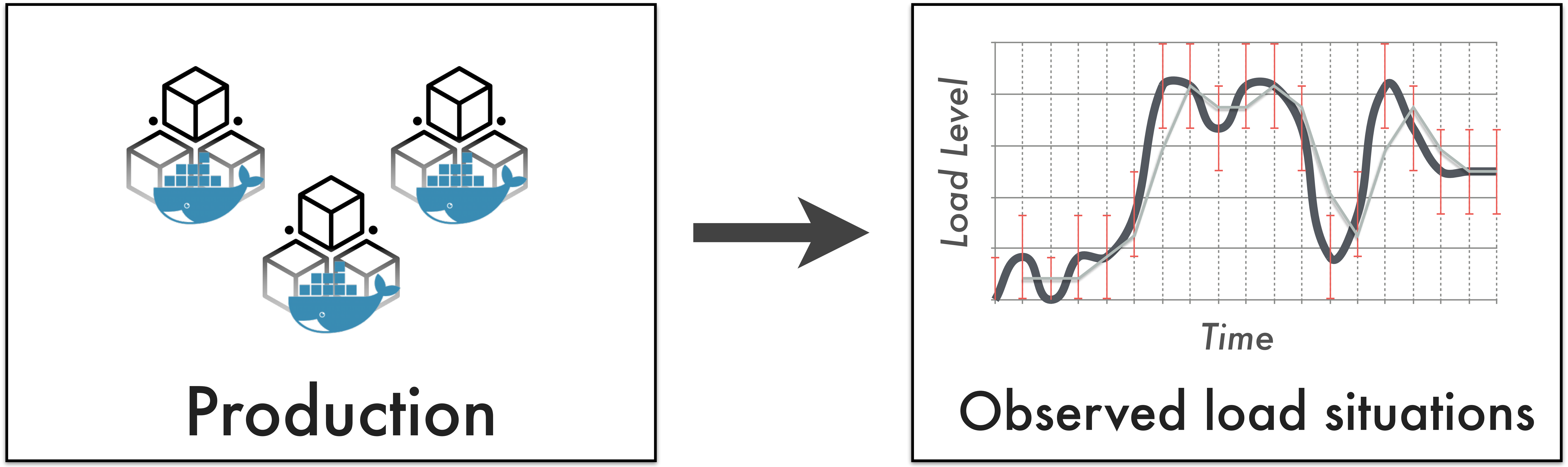
- Need to have complete automation for test execution
- Need to have clear KPIs to be evaluated in the pipeline

Our Approach

Our Approach



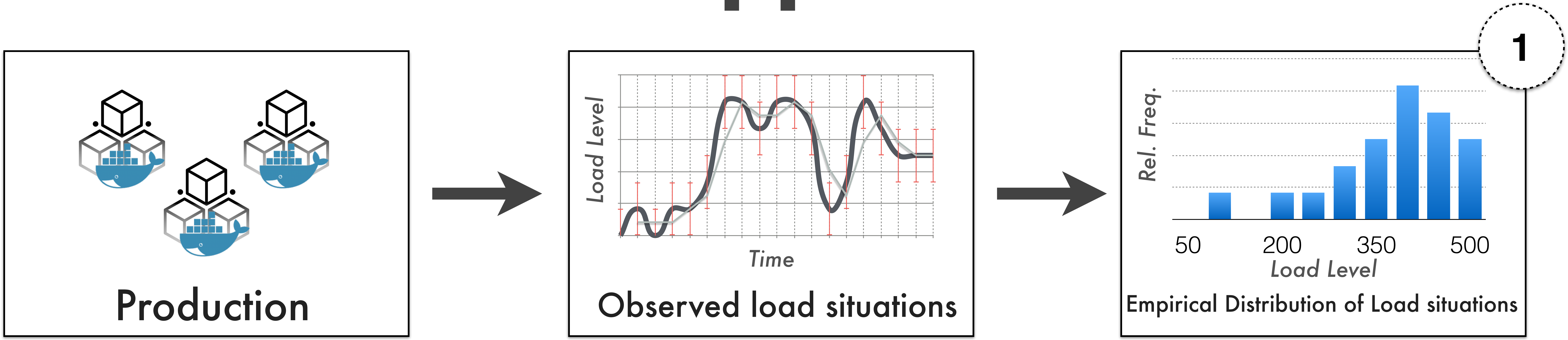
Our Approach



Production

Observed load situations

Our Approach

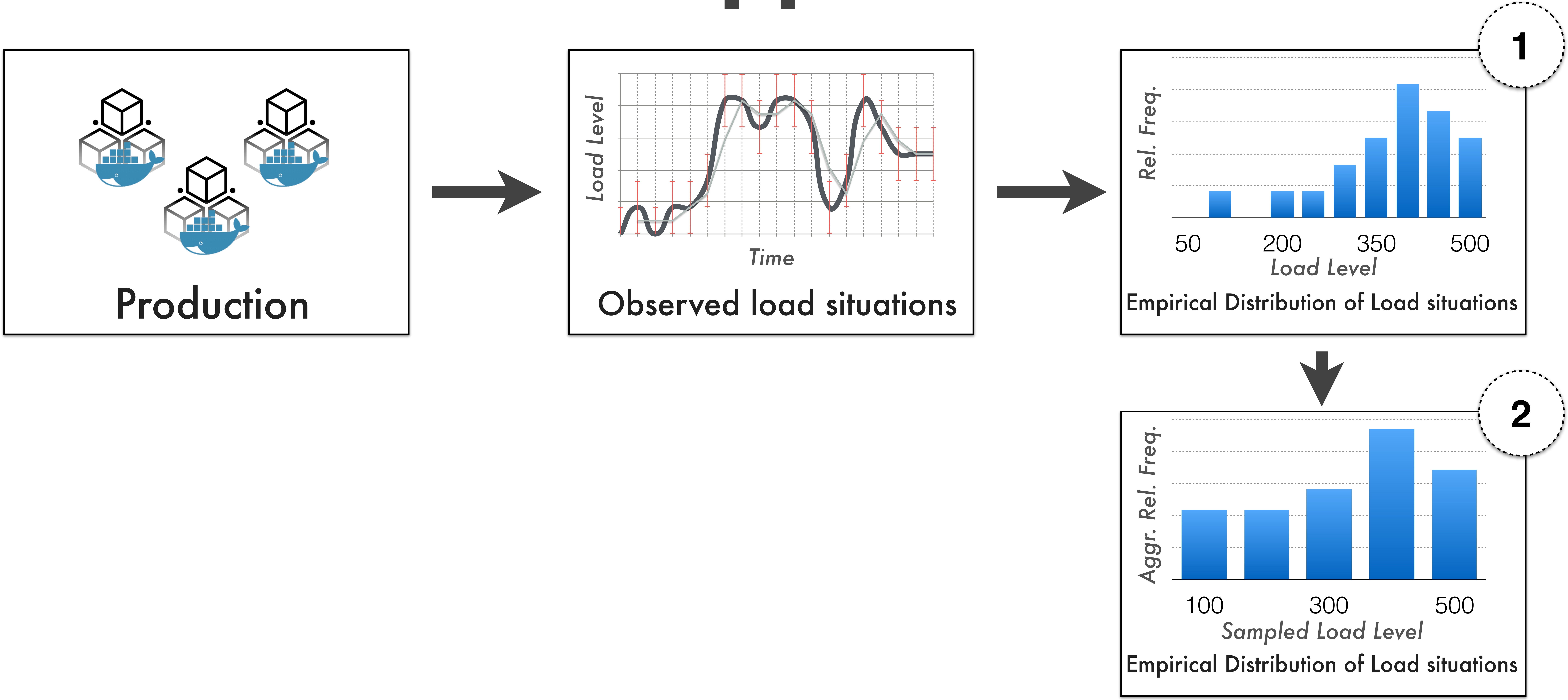


Production

Observed load situations

Empirical Distribution of Load situations

Our Approach



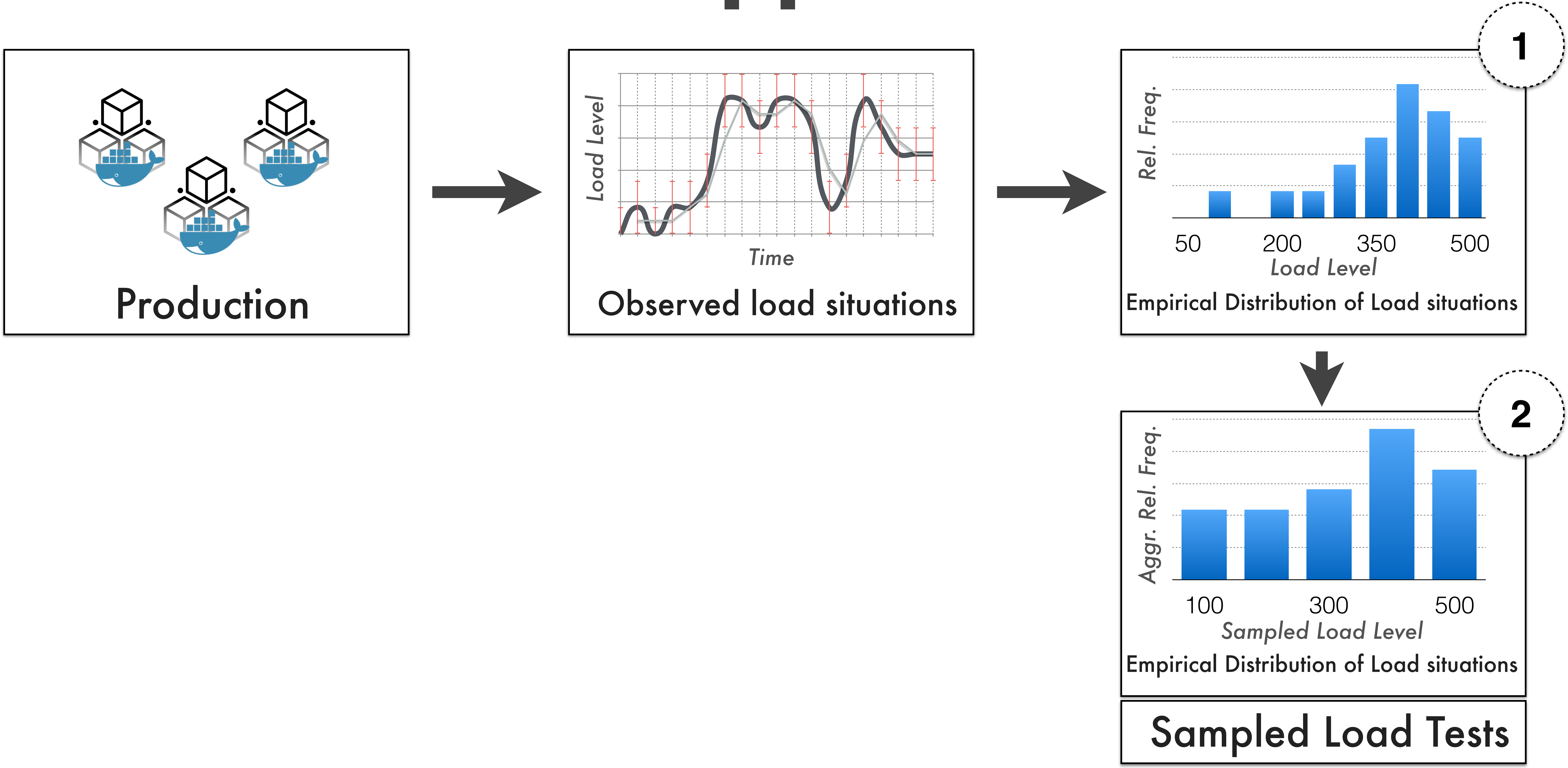
Production

Observed load situations

Empirical Distribution of Load situations

Empirical Distribution of Load situations

Our Approach



Production

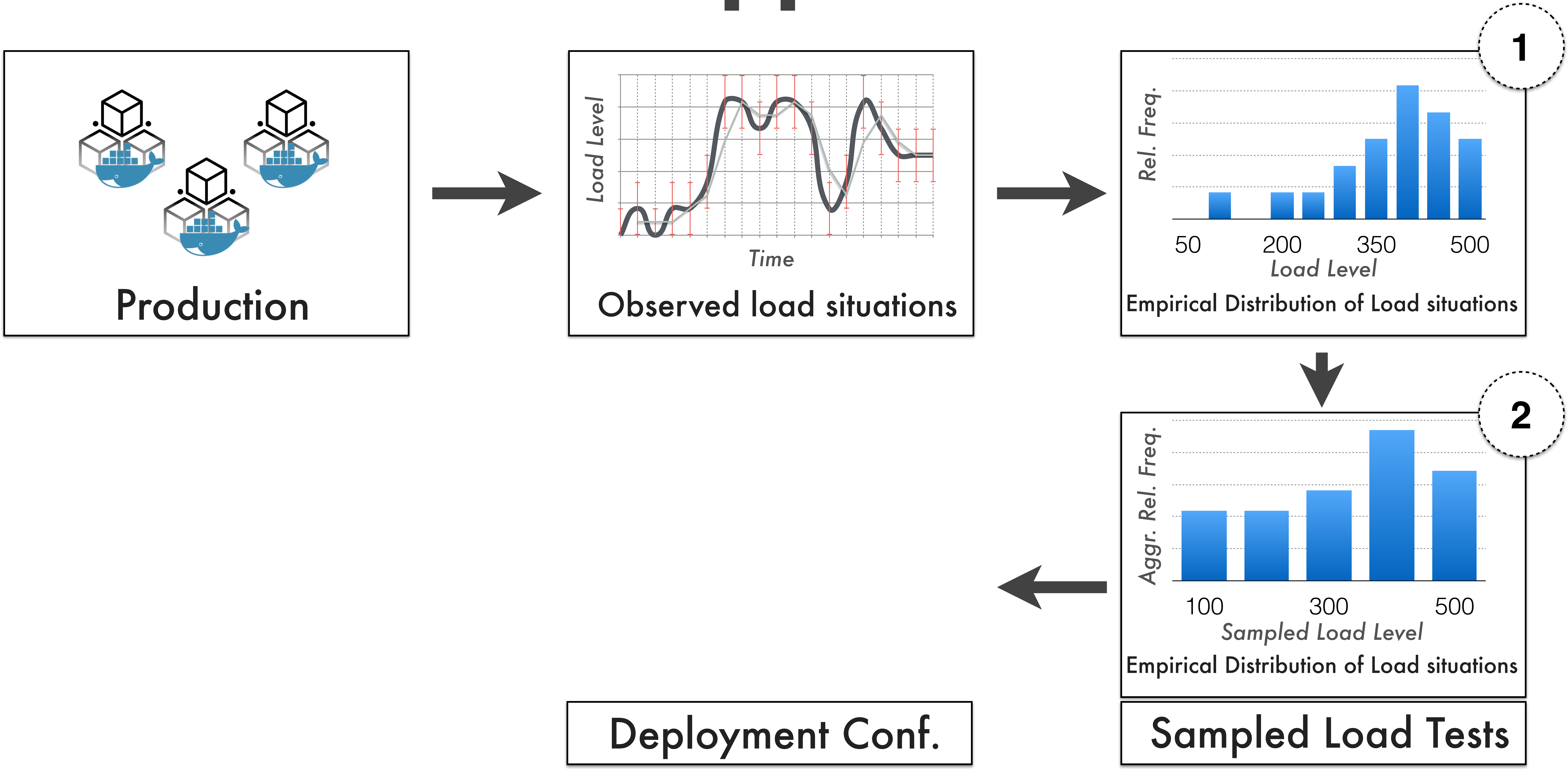
Observed load situations

Empirical Distribution of Load situations

Empirical Distribution of Load situations

Sampled Load Tests

Our Approach



Production

Observed load situations

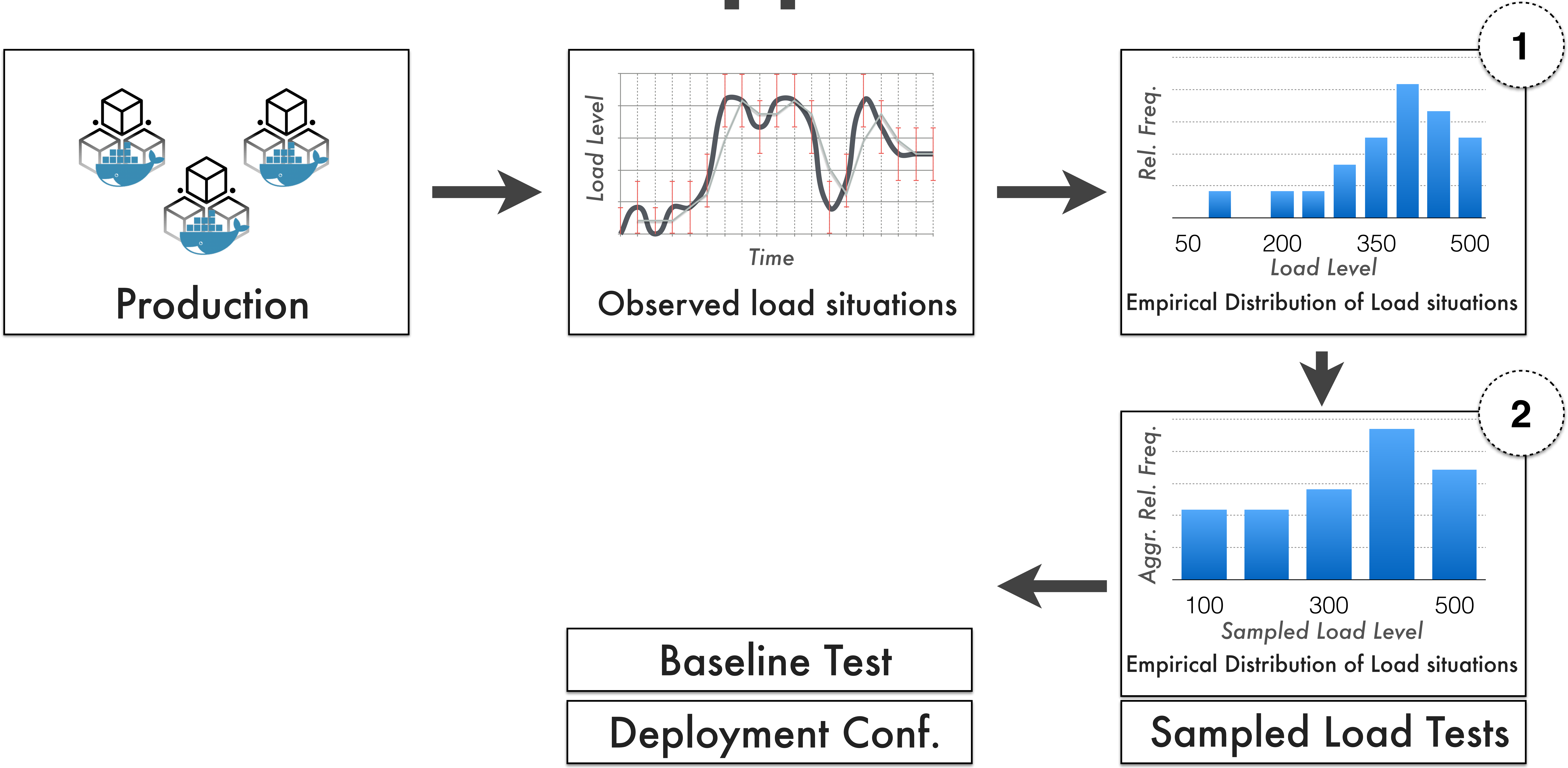
Empirical Distribution of Load situations

Empirical Distribution of Load situations

Sampled Load Tests

Deployment Conf.

Our Approach



Production

Observed load situations

Empirical Distribution of Load situations

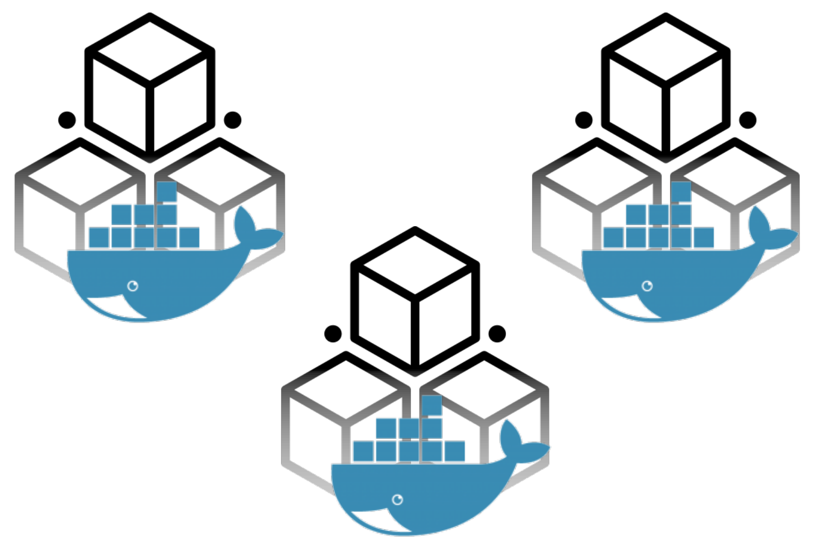
Empirical Distribution of Load situations

Sampled Load Tests

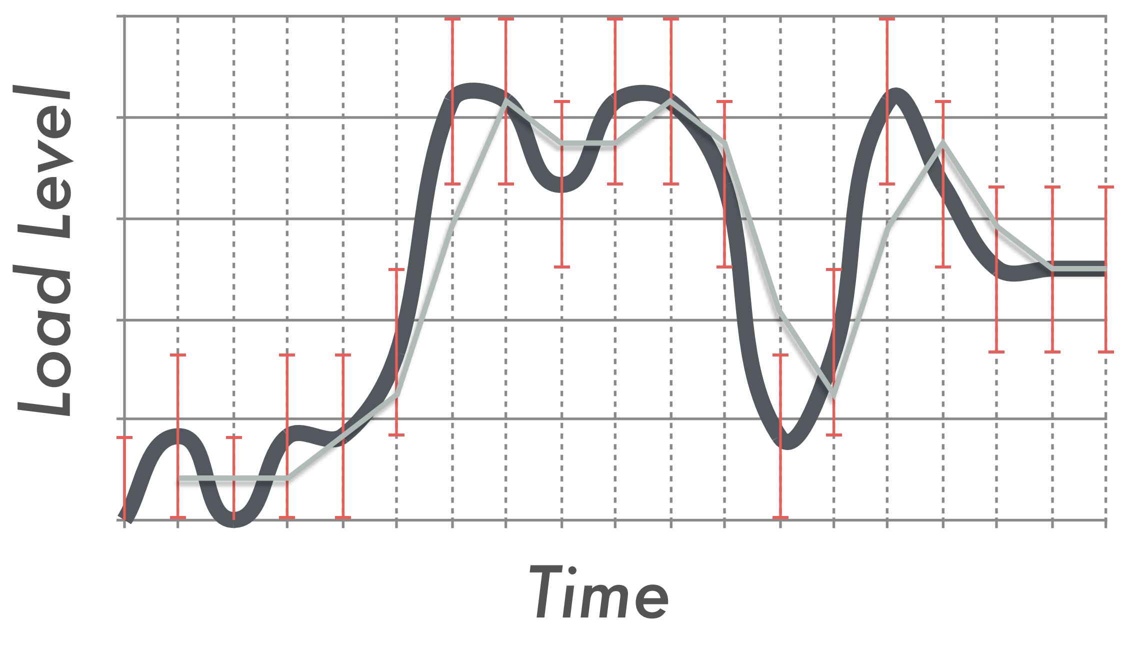
Baseline Test

Deployment Conf.

Our Approach

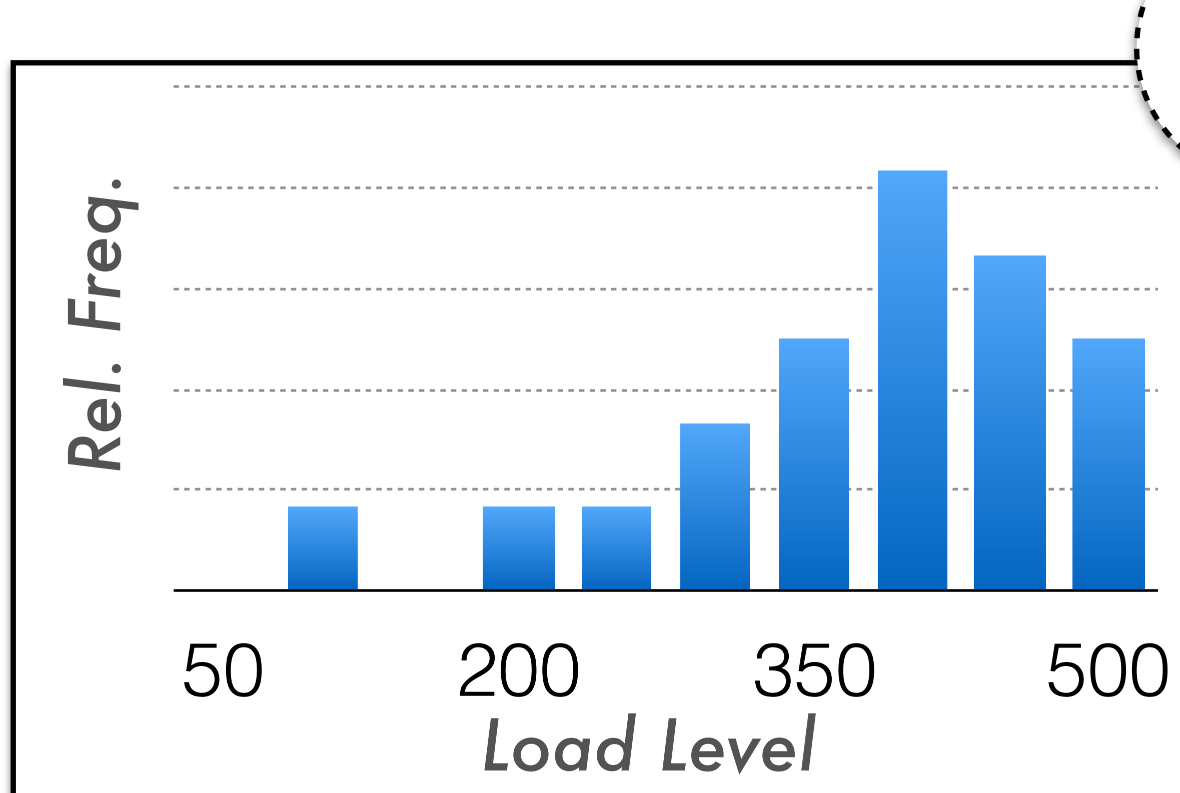


Production



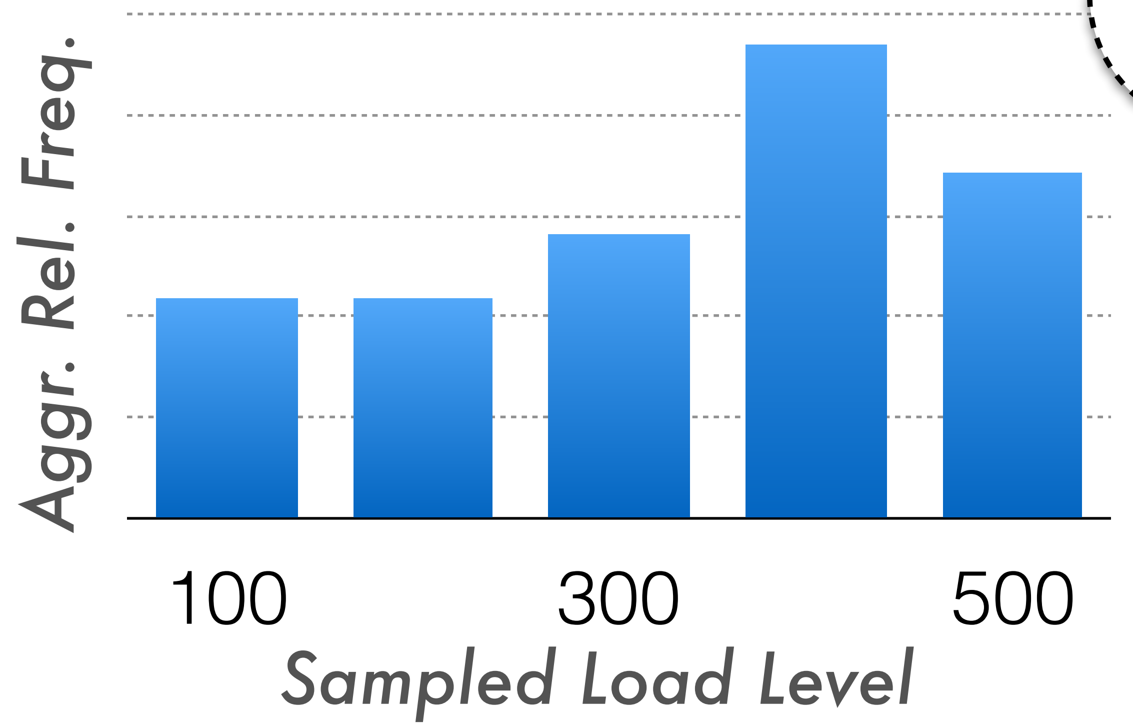
Observed load situations

1



Empirical Distribution of Load situations

2



Empirical Distribution of Load situations

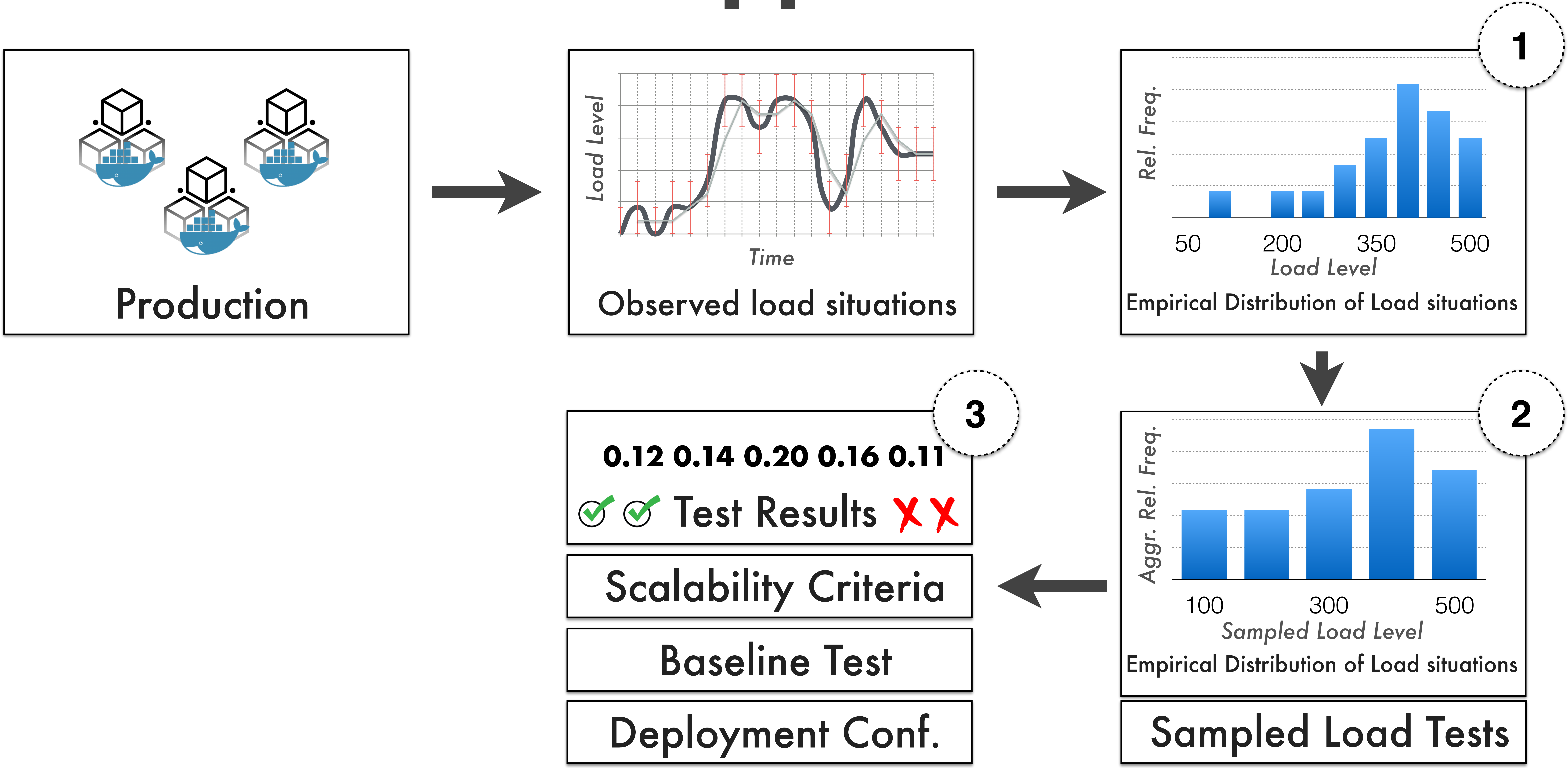
Sampled Load Tests

Scalability Criteria

Baseline Test

Deployment Conf.

Our Approach



Production

Observed load situations

Empirical Distribution of Load situations

Empirical Distribution of Load situations

Sampled Load Tests

0.12 0.14 0.20 0.16 0.11

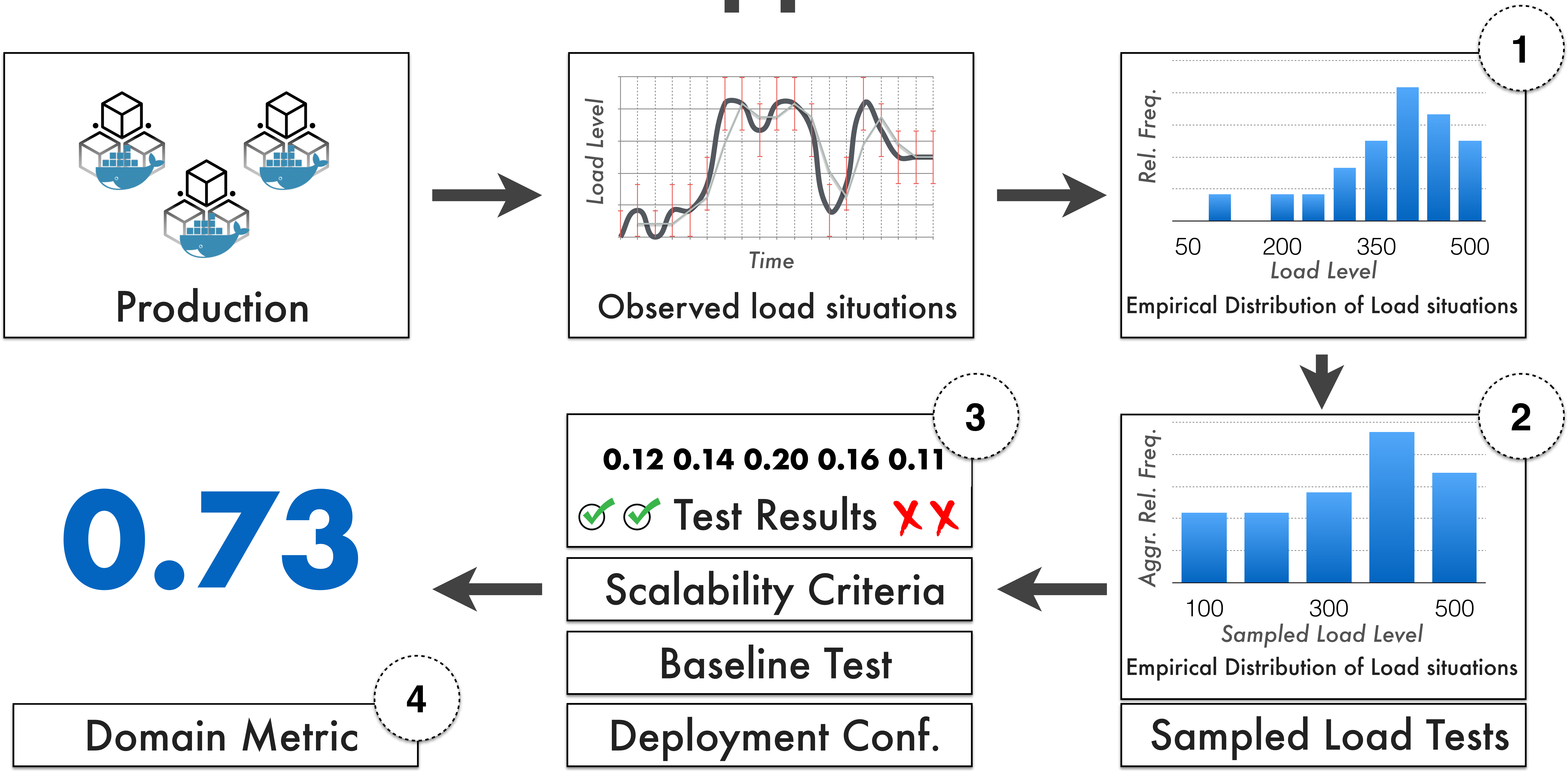
Test Results

Scalability Criteria

Baseline Test

Deployment Conf.

Our Approach



Production

Observed load situations

1

Empirical Distribution of Load situations

2

Empirical Distribution of Load situations

Sampled Load Tests

3

0.12 0.14 0.20 0.16 0.11

✓ ✓ Test Results ✗ ✗

Scalability Criteria

Baseline Test

Deployment Conf.

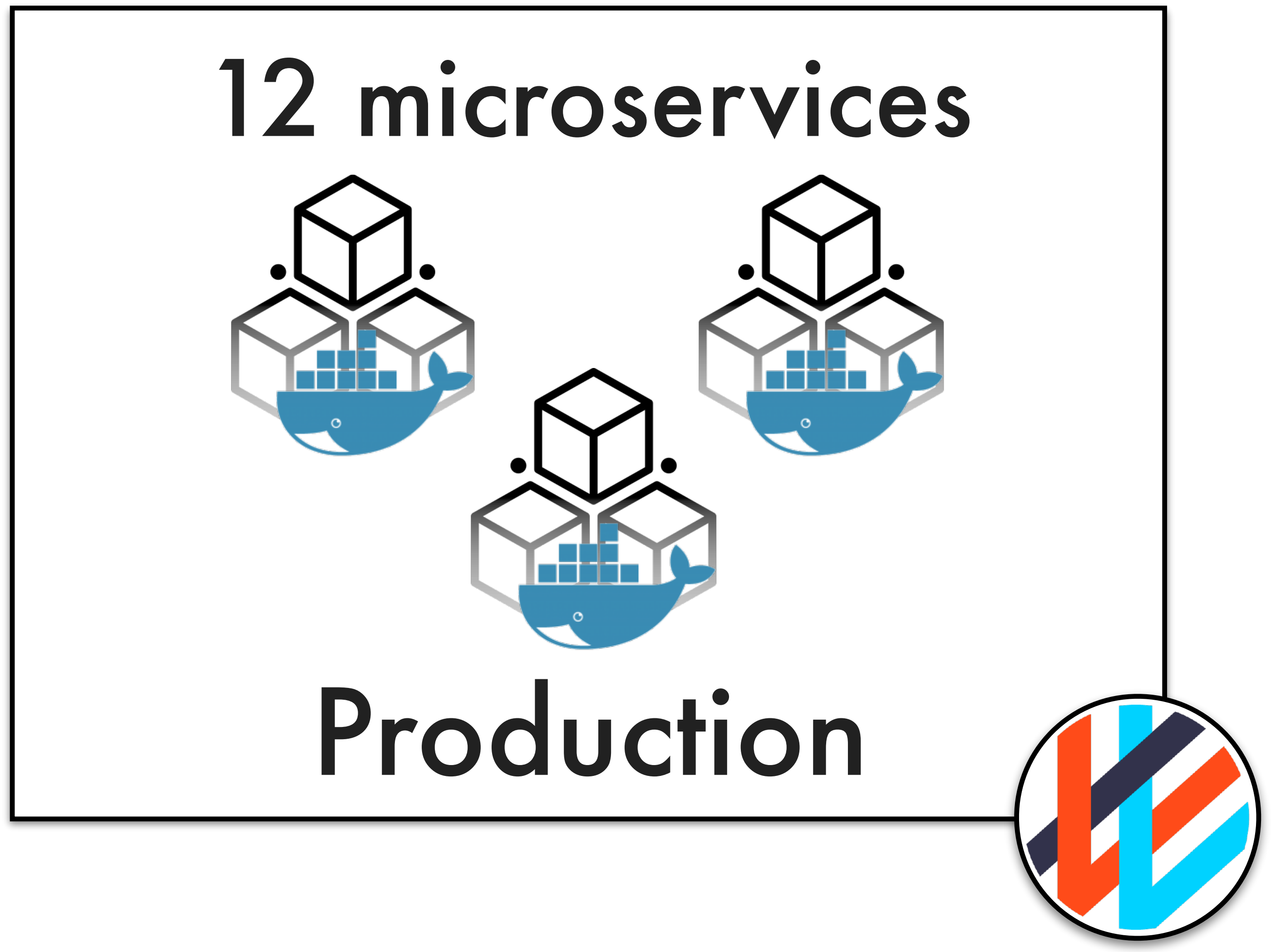
0.73

4

Domain Metric

Preliminary Experiments

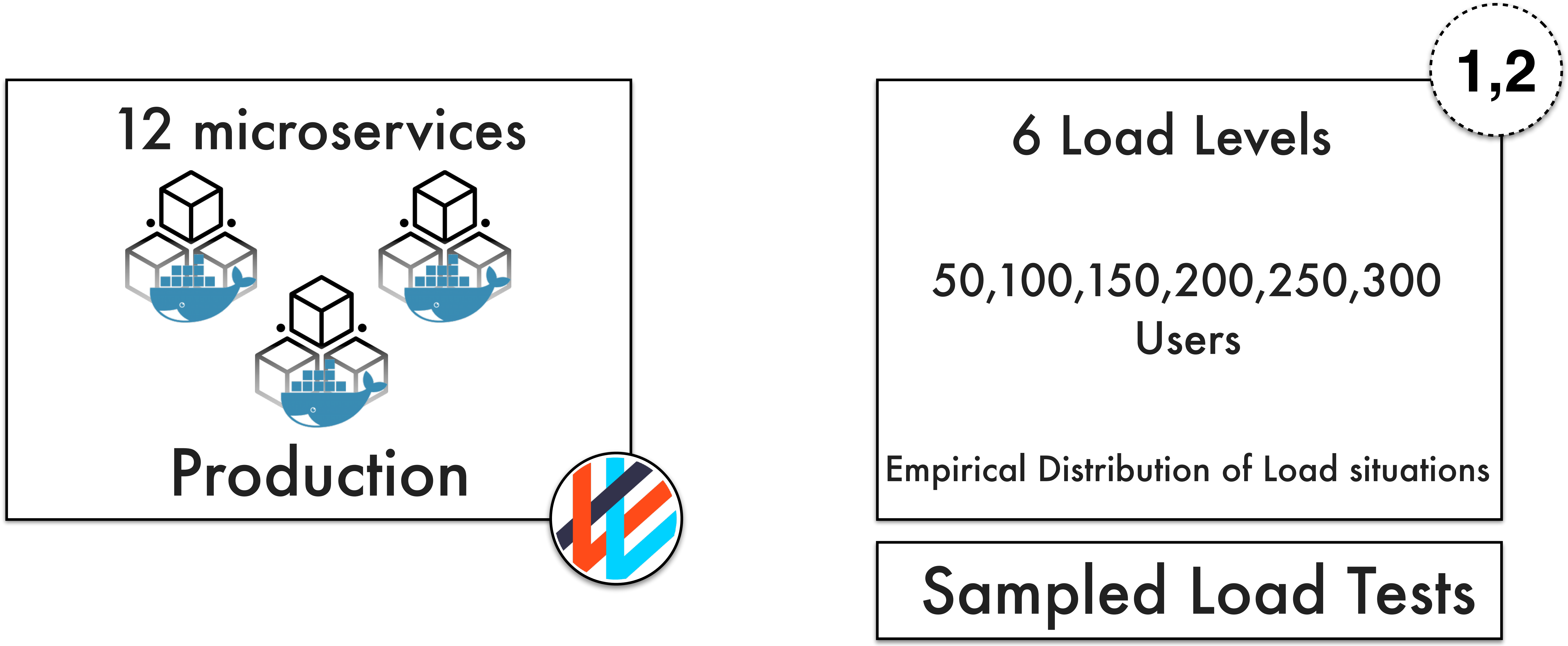
Preliminary Experiments



12 microservices

Production

Preliminary Experiments



12 microservices

Production

1,2

6 Load Levels

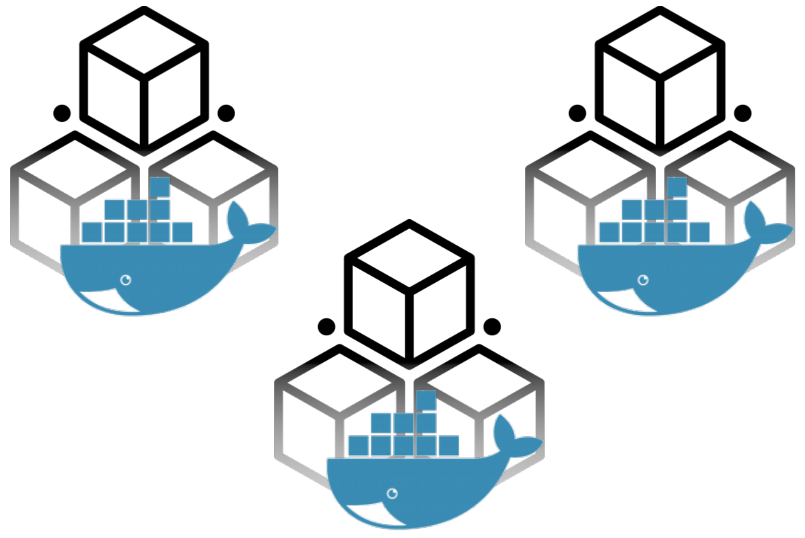
50,100,150,200,250,300 Users

Empirical Distribution of Load situations

Sampled Load Tests

Preliminary Experiments

12 microservices



Production

1,2

6 Load Levels

50,100,150,200,250,300
Users

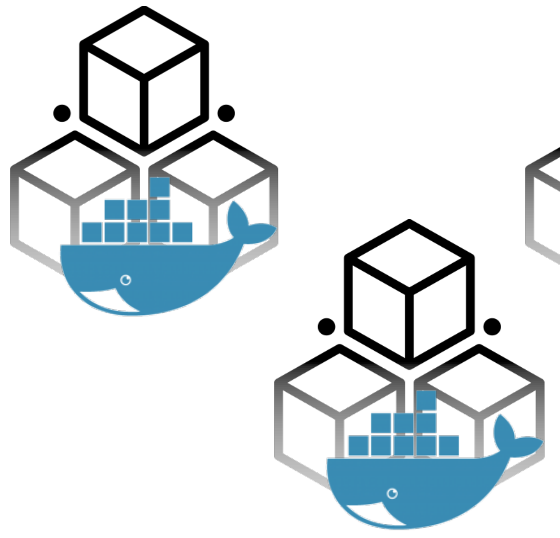
Empirical Distribution of Load situations

Sampled Load Tests

Custom Op. Mix

Preliminary Experiments

12 microservices



Production

1,2

6 Load Levels

50,100,150,200,250,300
Users

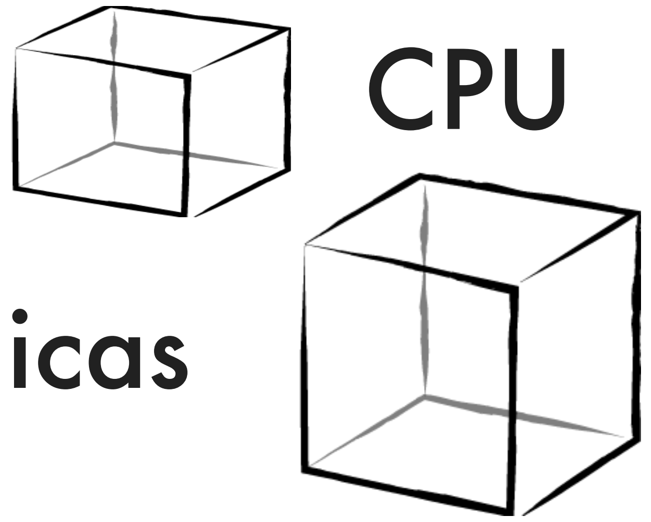
Empirical Distribution of Load situations

Sampled Load Tests

Custom Op. Mix

3

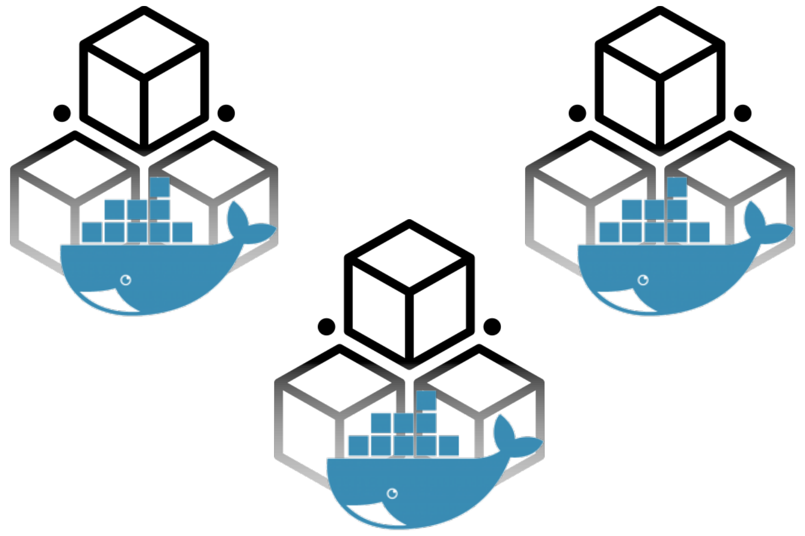
10 configurations



Deployment Config.

Preliminary Experiments

12 microservices



Production

1,2

6 Load Levels

50,100,150,200,250,300
Users

Empirical Distribution of Load situations

Sampled Load Tests

Custom Op. Mix

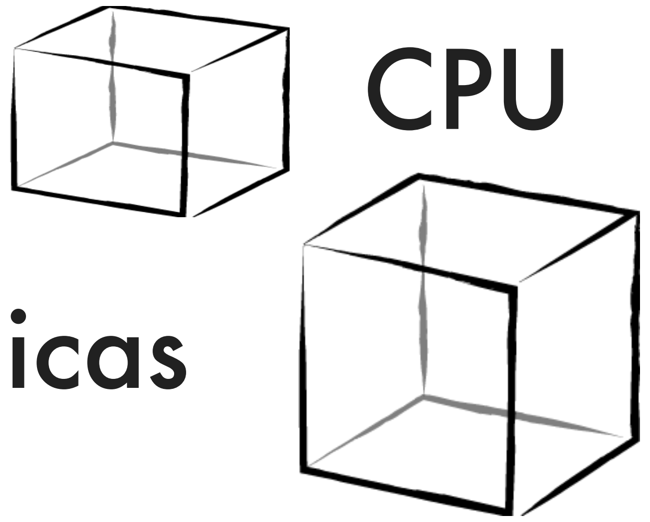
3

10 configurations

RAM

CPU

Replicas

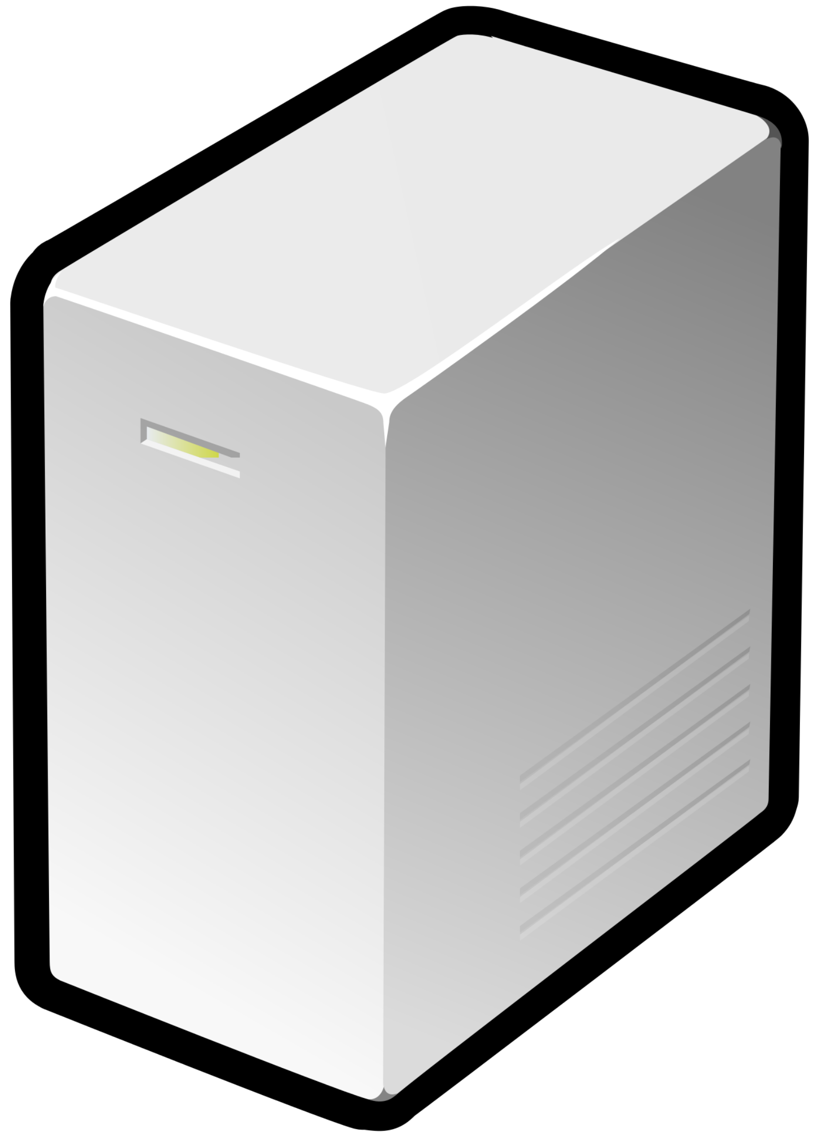


Deployment Config.

Scal = avg + 3σ

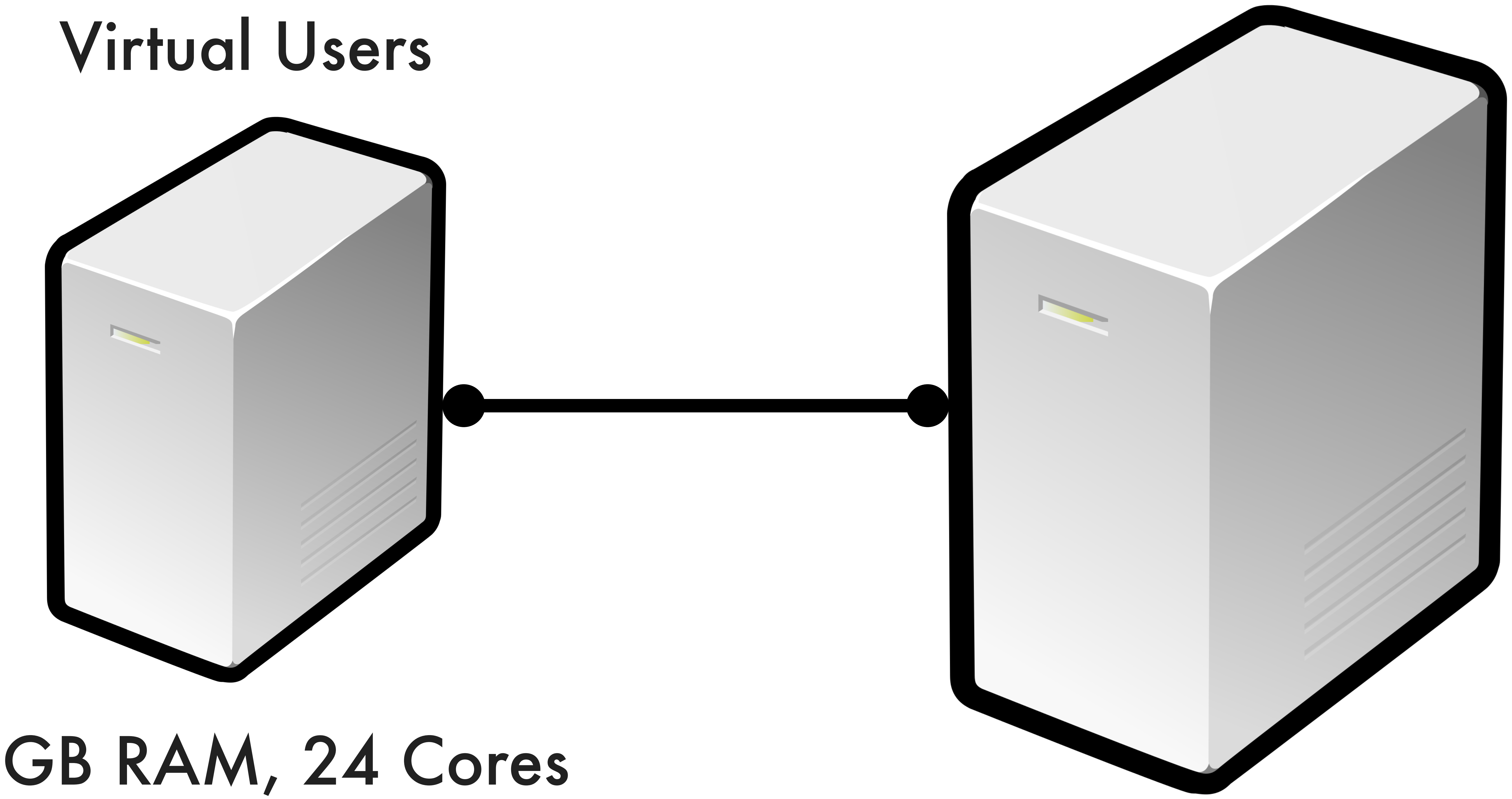
Experiments Infrastructure

Virtual Users



32GB RAM, 24 Cores

Experiments Infrastructure



Virtual Users

32GB RAM, 24 Cores

System Under Test

896GB RAM, 80 Cores

Preliminary Experiments Results

Deployment Configuration: 1 GB RAM, 0.25 CPU, 1 Replica

Preliminary Experiments Results

API	Scalability Criteria (PASS/FAIL)
GET /	**PASS**
GET /cart	**PASS**
POST /item	**FAIL**

Custom Op. Mix

Deployment Configuration: 1 GB RAM, 0.25 CPU, 1 Replica

Preliminary Experiments Results

API	Scalability Criteria (PASS/FAIL)
GET /	**PASS**
GET /cart	**PASS**
POST /item	**FAIL**

Custom Op. Mix

Users	Aggr. Rel. Freq.
50	0.10582
100	0.18519
150	0.22222
200	0.22222
250	0.20370
300	0.06085

Aggr. Rel. Freq.

Deployment Configuration: 1 GB RAM, 0.25 CPU, 1 Replica

Preliminary Experiments Results

API	Scalability Criteria (PASS/FAIL)
GET /	**PASS**
GET /cart	**PASS**
POST /item	**FAIL**

Users	Aggr. Rel. Freq.
50	0.10582
100	0.18519
150	0.22222
200	0.22222
250	0.20370
300	0.06085

Custom Op. Mix

Aggr. Rel. Freq.

Contrib. to Domain Metric

Deployment Configuration: 1 GB RAM, 0.25 CPU, 1 Replica

Preliminary Experiments Results

API	Scalability Criteria (PASS/FAIL)
GET /	**PASS**
GET /cart	**PASS**
POST /item	**FAIL**

Users	Aggr. Rel. Freq.
50	0.10582
100	0.18519
150	0.22222
200	0.22222
250	0.20370
300	0.06085

Max: 0.20370

Custom Op. Mix

Aggr. Rel. Freq.

Contrib. to Domain Metric

Deployment Configuration: 1 GB RAM, 0.25 CPU, 1 Replica

Preliminary Experiments Results

API	Scalability Criteria (PASS/FAIL)
GET /	**PASS**
GET /cart	**PASS**
POST /item	**FAIL**

Custom Op. Mix

Users	Aggr. Rel. Freq.
50	0.10582
100	0.18519
150	0.22222
200	0.22222
250	0.20370
300	0.06085

Aggr. Rel. Freq.

Max: 0.20370

Actual: 0.13580

Contrib. to Domain Metric

Deployment Configuration: 1 GB RAM, 0.25 CPU, 1 Replica

Preliminary Experiments Results

Deployment Configuration: 1 GB RAM, 0.25 CPU, 1 Replica

Preliminary Experiments Results

Users	Contribution
50	0.10582
100	0.18519
150	0.22222
200	0.07999
250	0.13580
300	0.04729

Contrib. to Domain Metric

Deployment Configuration: 1 GB RAM, 0.25 CPU, 1 Replica

Preliminary Experiments Results

Users	Contribution
50	0.10582
100	0.18519
150	0.22222
200	0.07999
250	0.13580
300	0.04729

Max: 1

Contrib. to Domain Metric

Domain Metric

4

Deployment Configuration: 1 GB RAM, 0.25 CPU, 1 Replica

Preliminary Experiments Results

Users	Contribution
50	0.10582
100	0.18519
150	0.22222
200	0.07999
250	0.13580
300	0.04729

Contrib. to Domain Metric

Max: 1

Actual:

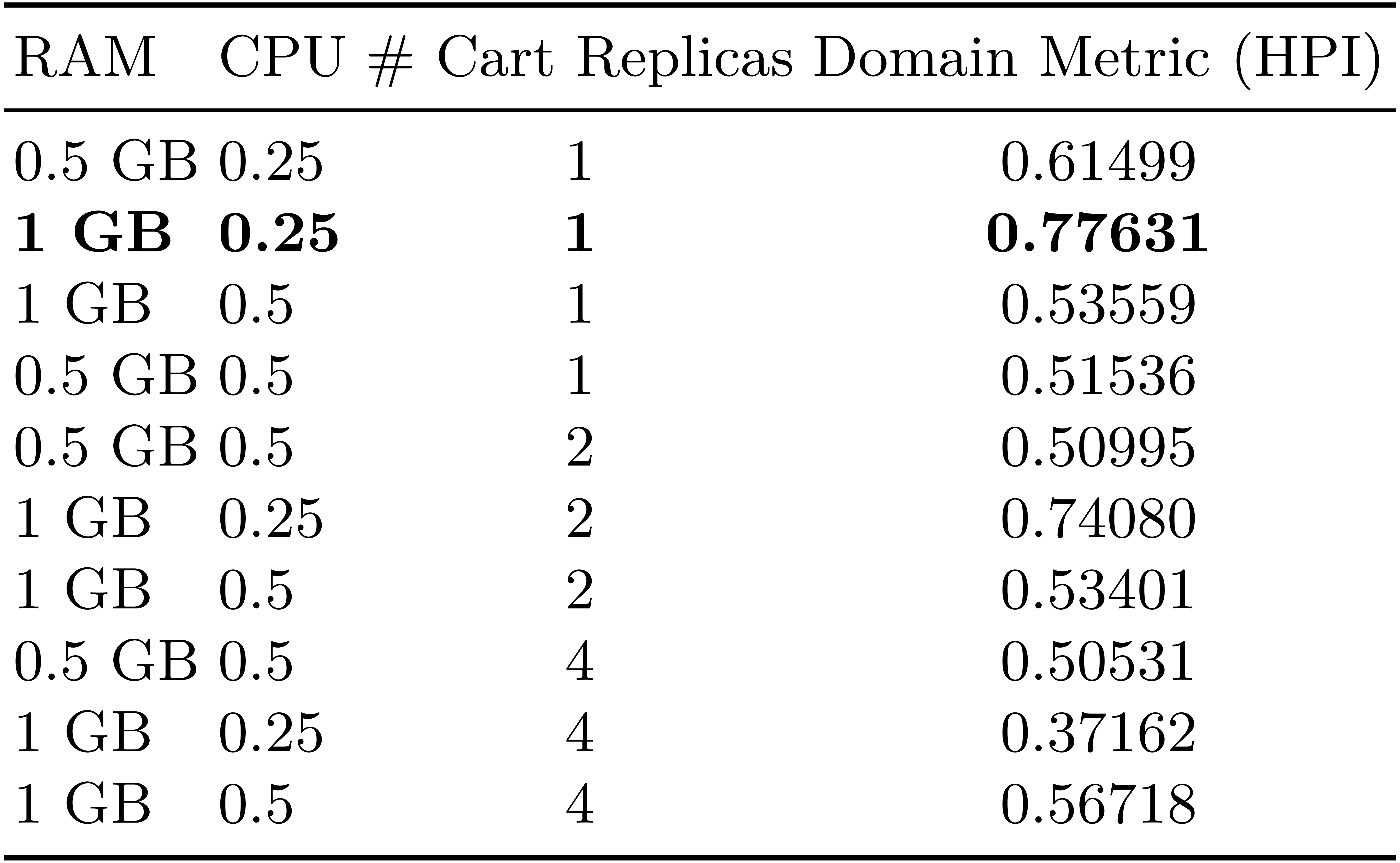
0.77631

4

Domain Metric

Deployment Configuration: 1 GB RAM, 0.25 CPU, 1 Replica

Preliminary Experiments Results

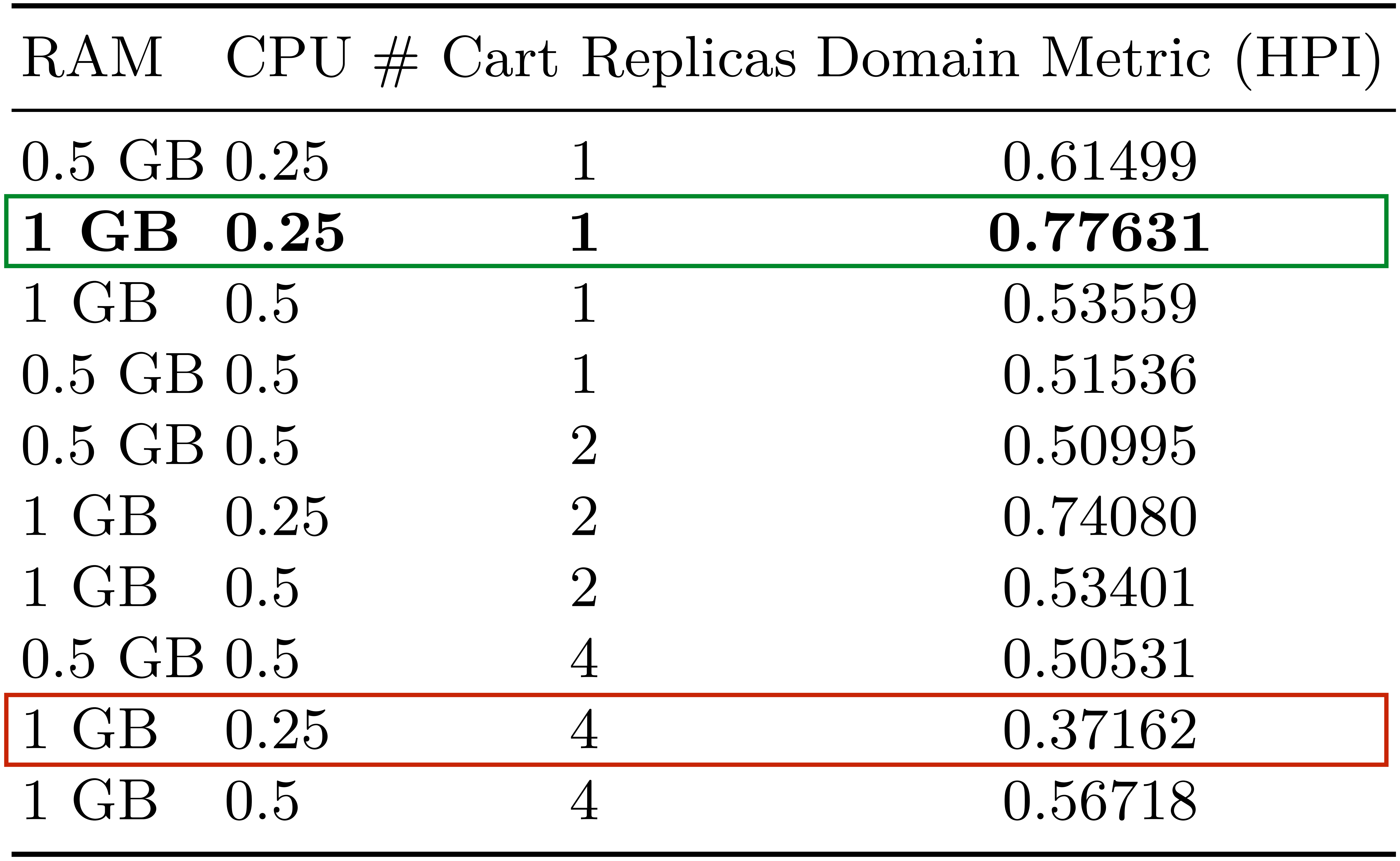


RAM	CPU	# Cart Replicas	Domain Metric (HPI)
0.5 GB	0.25	1	0.61499
1 GB	**0.25**	**1**	**0.77631**
1 GB	0.5	1	0.53559
0.5 GB	0.5	1	0.51536
0.5 GB	0.5	2	0.50995
1 GB	0.25	2	0.74080
1 GB	0.5	2	0.53401
0.5 GB	0.5	4	0.50531
1 GB	0.25	4	0.37162
1 GB	0.5	4	0.56718

Preliminary Experiments Results

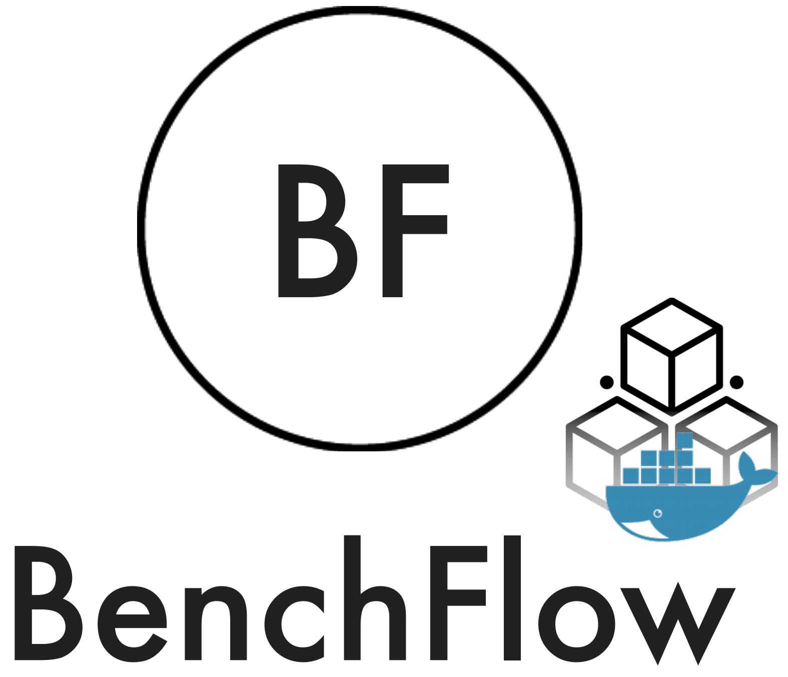
RAM	CPU	# Cart Replicas	Domain Metric (HPI)
0.5 GB	0.25	1	0.61499
1 GB	**0.25**	**1**	**0.77631**
1 GB	0.5	1	0.53559
0.5 GB	0.5	1	0.51536
0.5 GB	0.5	2	0.50995
1 GB	0.25	2	0.74080
1 GB	0.5	2	0.53401
0.5 GB	0.5	4	0.50531
1 GB	0.25	4	0.37162
1 GB	0.5	4	0.56718

Preliminary Experiments Results



RAM	CPU	# Cart Replicas	Domain Metric (HPI)
0.5 GB	0.25	1	0.61499
1 GB	**0.25**	**1**	**0.77631**
1 GB	0.5	1	0.53559
0.5 GB	0.5	1	0.51536
0.5 GB	0.5	2	0.50995
1 GB	0.25	2	0.74080
1 GB	0.5	2	0.53401
0.5 GB	0.5	4	0.50531
1 GB	0.25	4	0.37162
1 GB	0.5	4	0.56718

BenchFlow Automation Framework



BenchFlow Automation Framework

DSL

BF

BenchFlow

BenchFlow Automation Framework

DSL

- Workload Modeling

BF

BenchFlow

BenchFlow Automation Framework

DSL

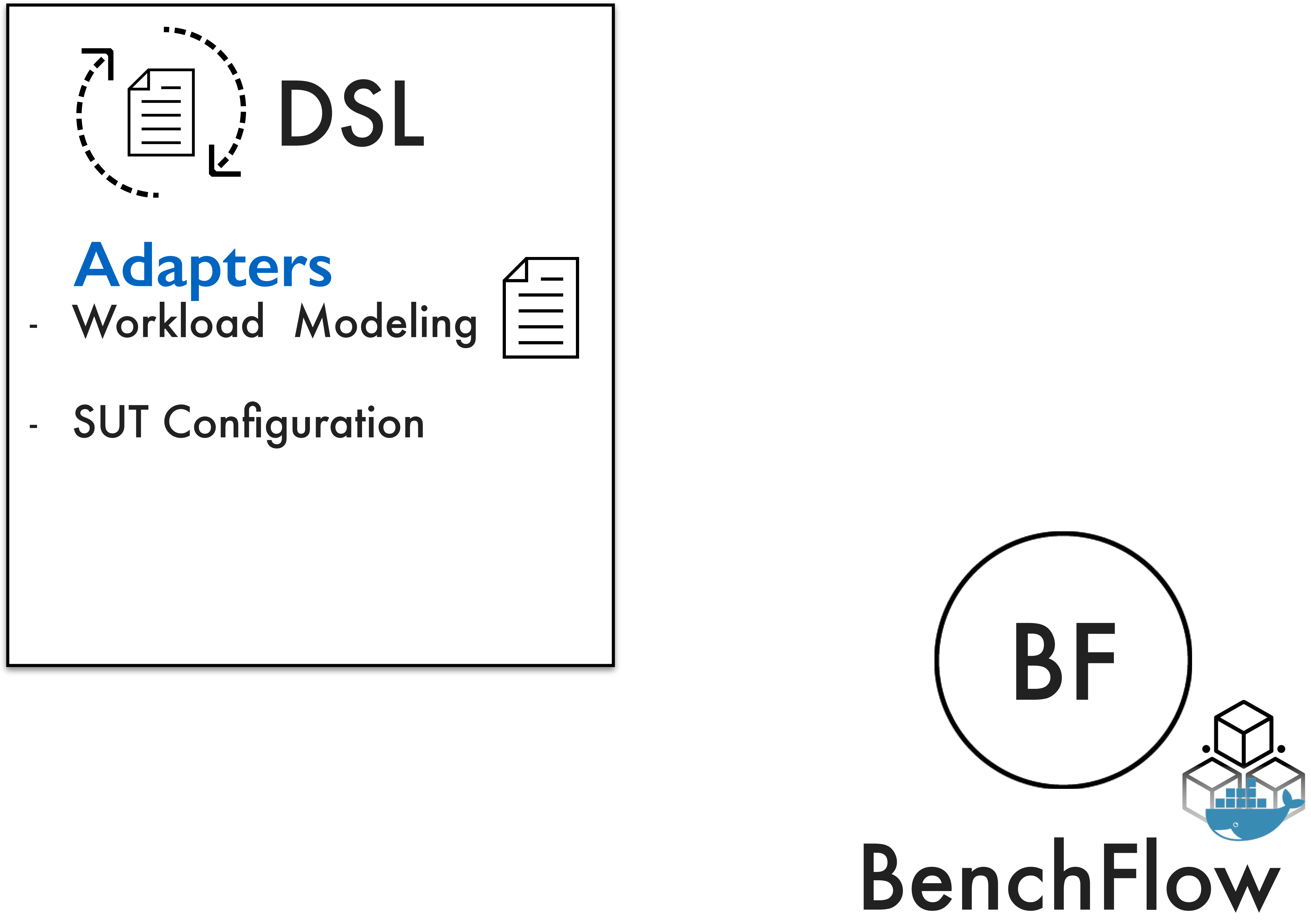
Adapters

- Workload Modeling

BF

BenchFlow

BenchFlow Automation Framework



DSL

Adapters

- Workload Modeling
- SUT Configuration

BenchFlow Automation Framework

DSL

Adapters

- Workload Modeling
- SUT Configuration
- SUT Deployment

BF

BenchFlow

BenchFlow Automation Framework

DSL

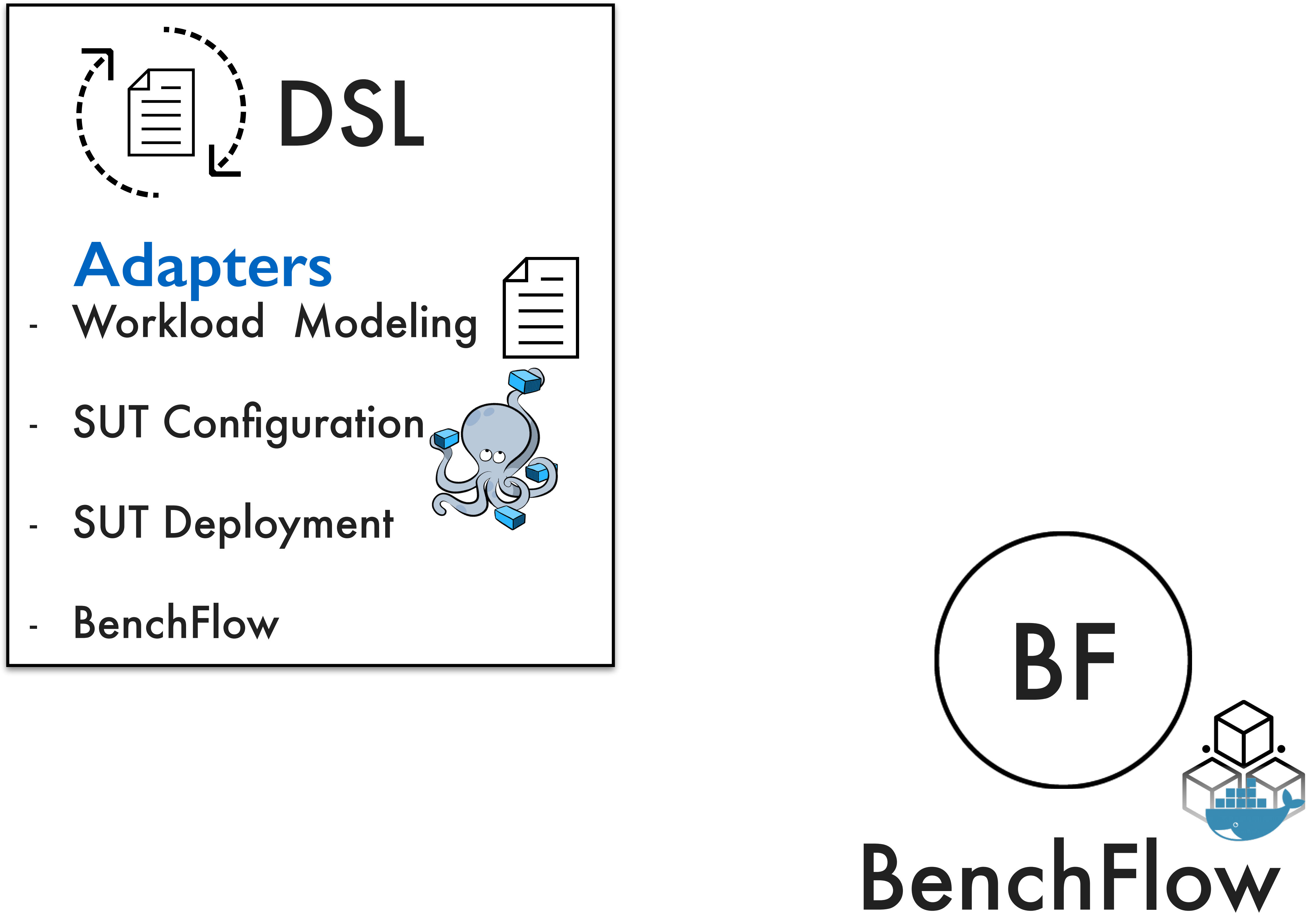
Adapters

- Workload Modeling
- SUT Configuration
- SUT Deployment

BF

BenchFlow

BenchFlow Automation Framework



DSL

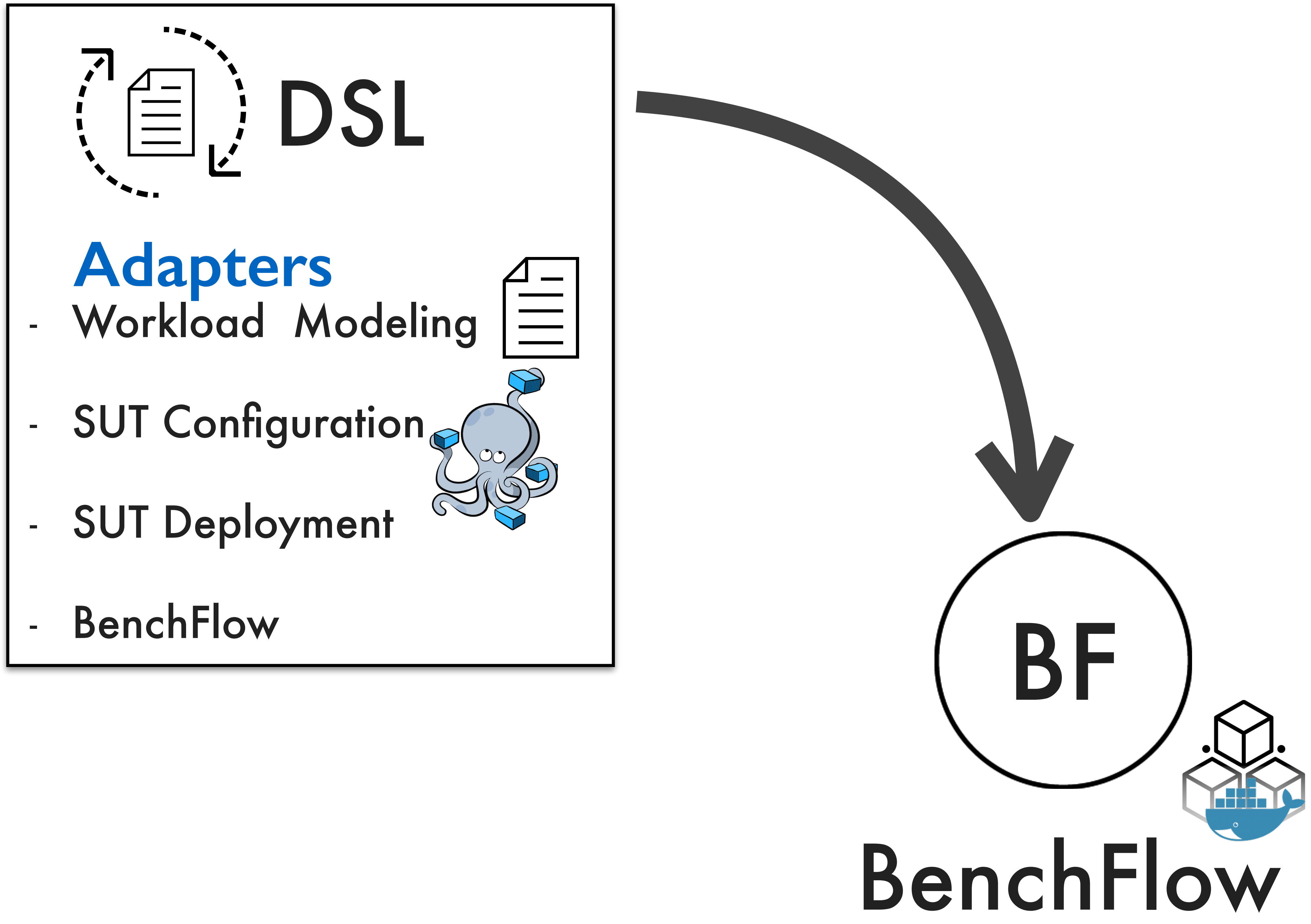
Adapters

- Workload Modeling
- SUT Configuration
- SUT Deployment
- BenchFlow

BF

BenchFlow

BenchFlow Automation Framework



DSL

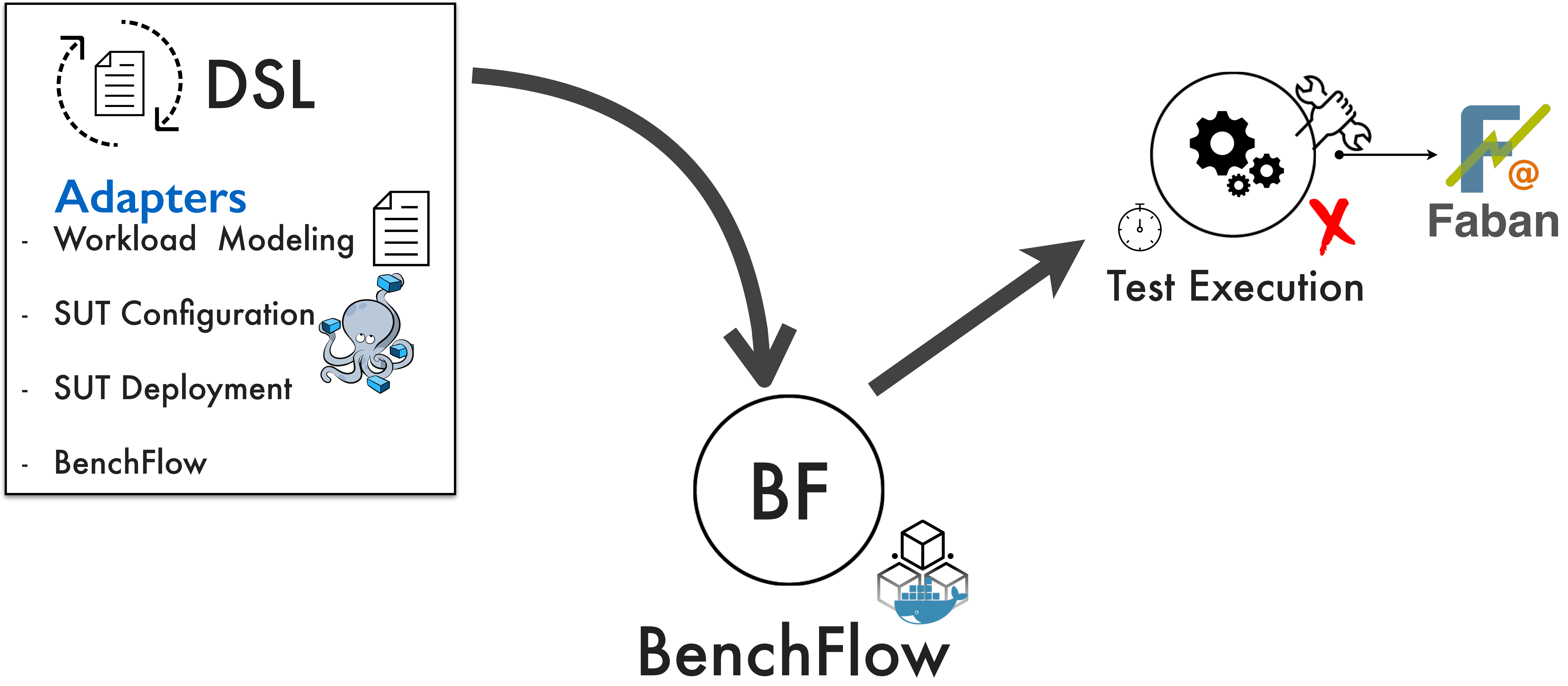
Adapters

- Workload Modeling
- SUT Configuration
- SUT Deployment
- BenchFlow

BF

BenchFlow

BenchFlow Automation Framework

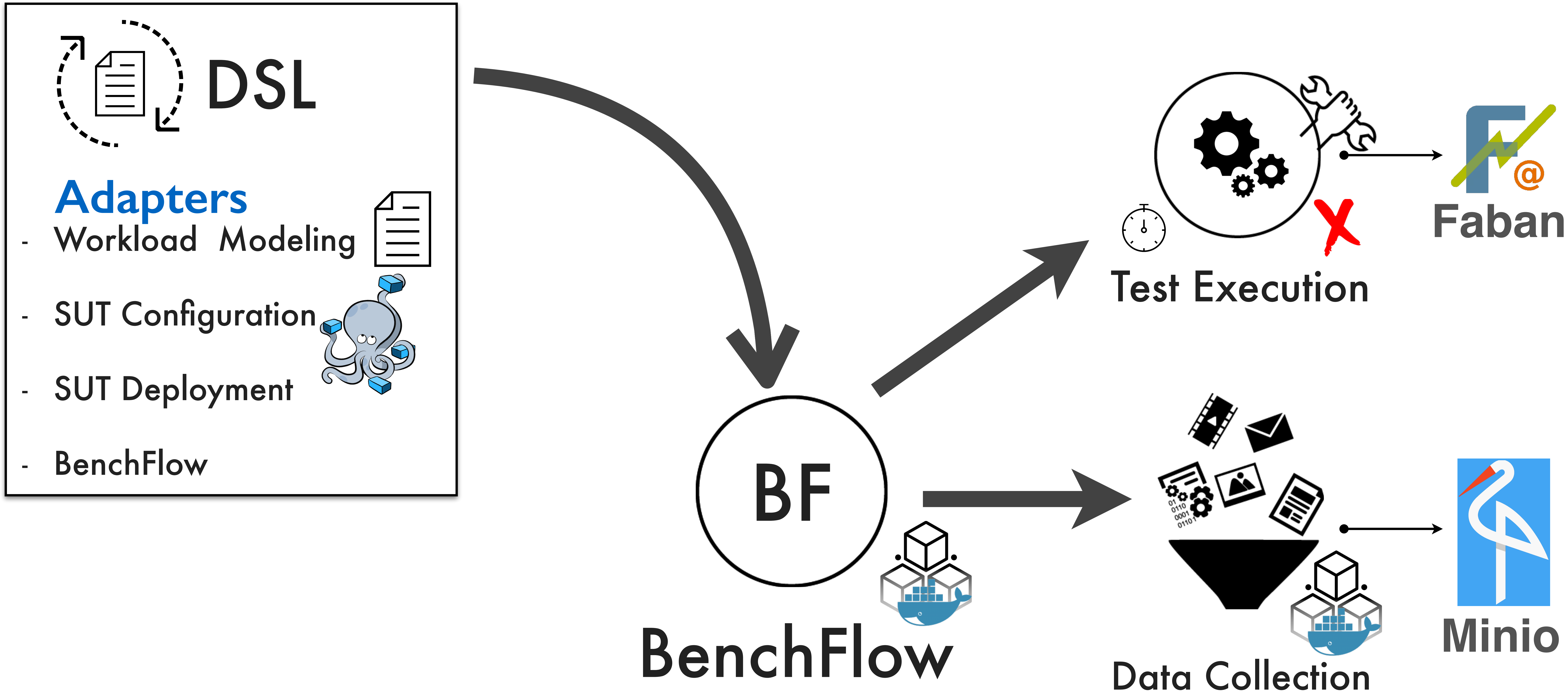


DSL

Adapters

- Workload Modeling
- SUT Configuration
- SUT Deployment
- BenchFlow

BenchFlow Automation Framework



DSL

Adapters

- Workload Modeling
- SUT Configuration
- SUT Deployment
- BenchFlow

BF

BenchFlow

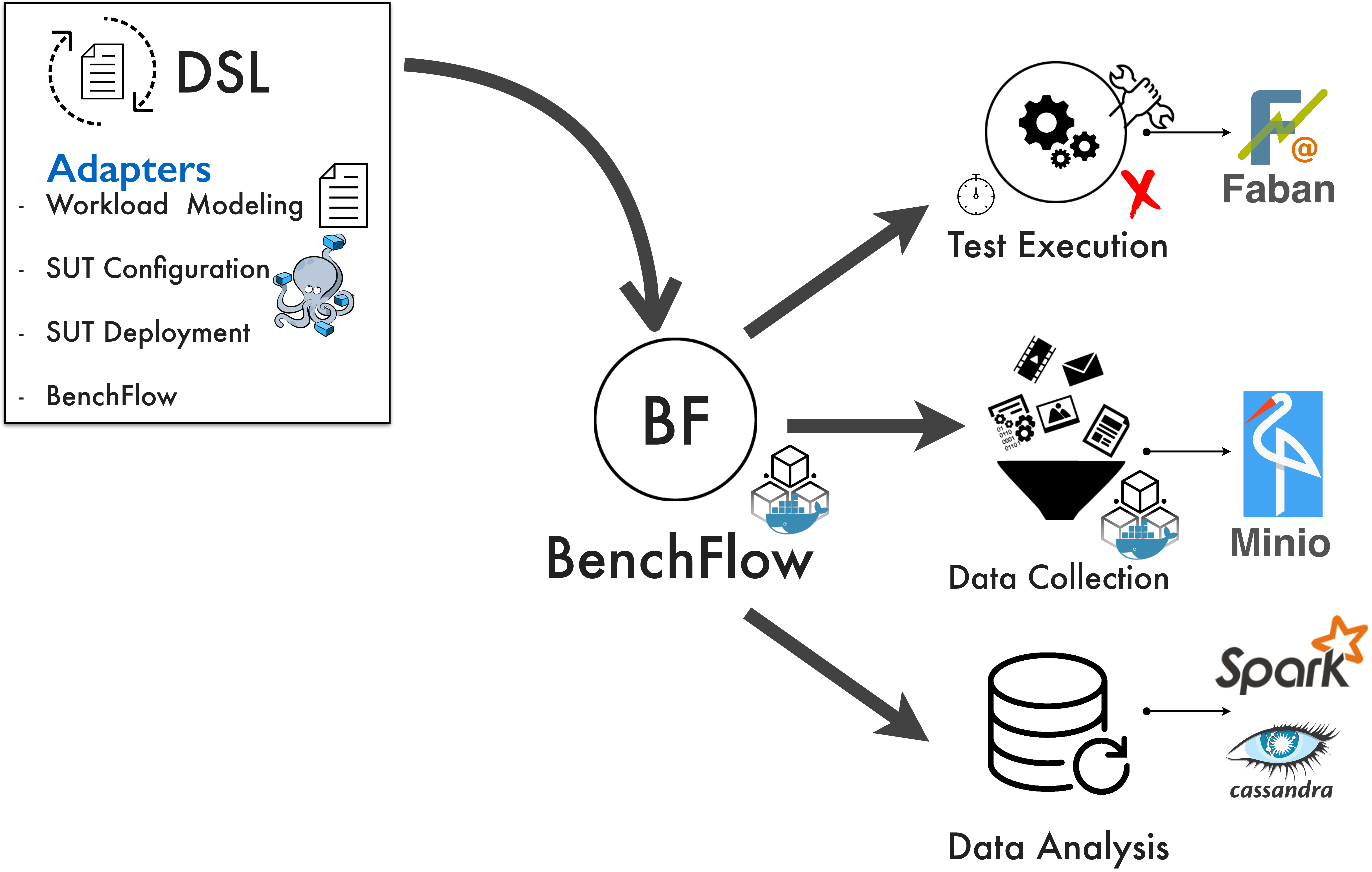
Test Execution

Faban

Data Collection

Minio

BenchFlow Automation Framework



DSL

Adapters

- Workload Modeling
- SUT Configuration
- SUT Deployment
- BenchFlow

BF

BenchFlow

Test Execution

Faban

Data Collection

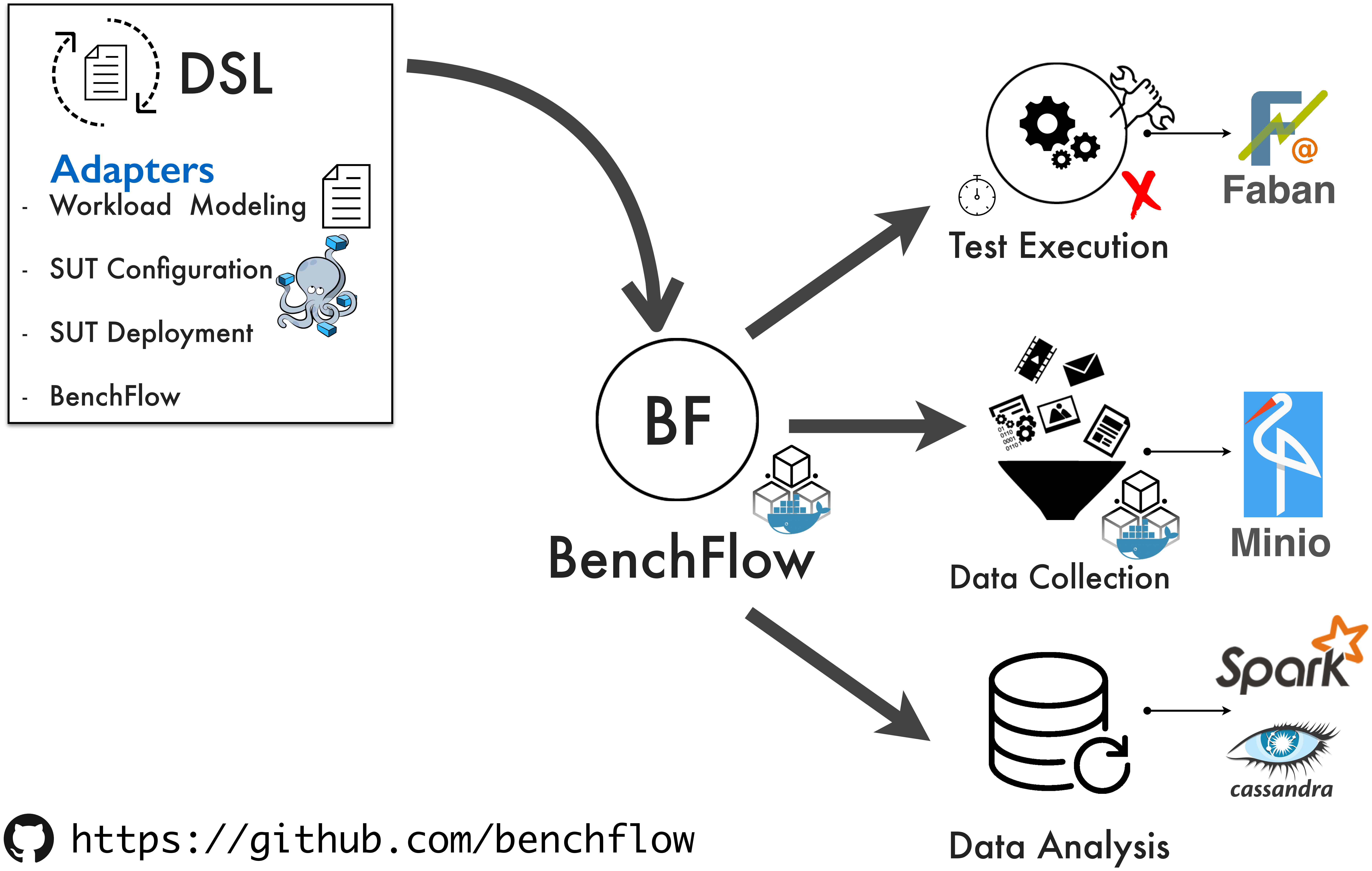
Minio

Data Analysis

Spark

cassandra

BenchFlow Automation Framework



DSL

Adapters

- Workload Modeling
- SUT Configuration
- SUT Deployment
- BenchFlow

BF

BenchFlow

Test Execution

Faban

Data Collection

Minio

Data Analysis

Spark

cassandra

https://github.com/benchflow

BenchFlow Automation Framework

Generation

[Ferme et al.]
Vincenzo Ferme and Cesare Pautasso. A Declarative Approach for Performance Tests Execution in Continuous Software Development Environments. In Proc. of ICPE 2018. 261-272.

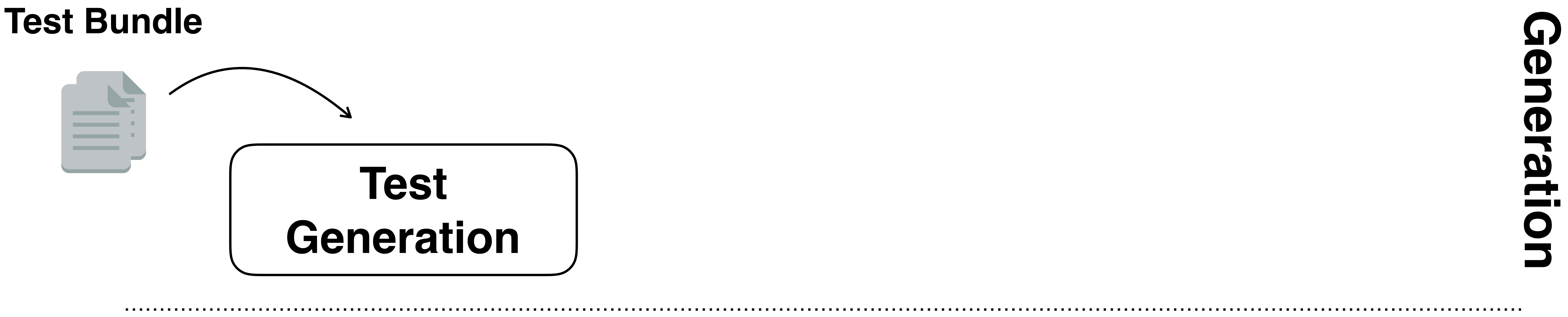
BenchFlow Automation Framework

Test Generation

Generation

[Ferme et al.]
Vincenzo Ferme and Cesare Pautasso. A Declarative Approach for Performance Tests Execution in Continuous Software Development Environments. In Proc. of ICPE 2018. 261-272.

BenchFlow Automation Framework



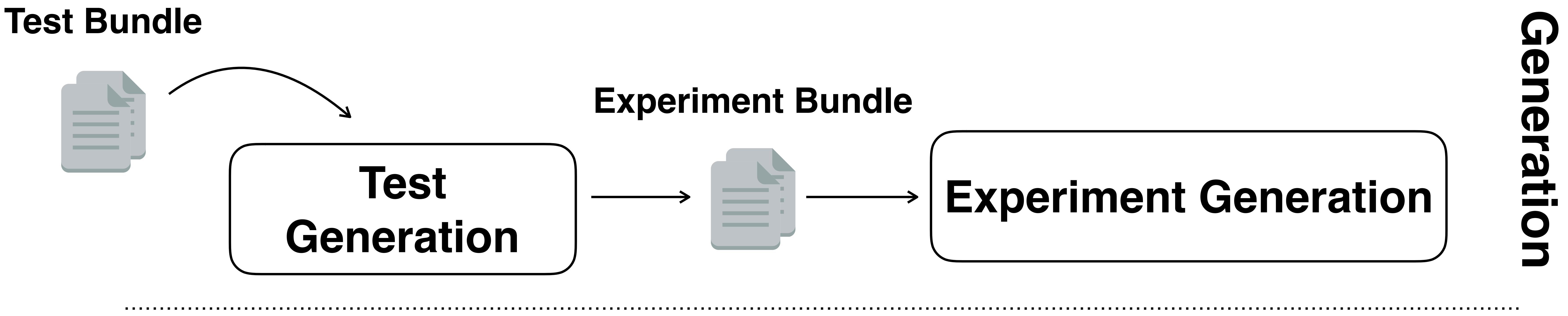
Test Bundle

Test Generation

Generation

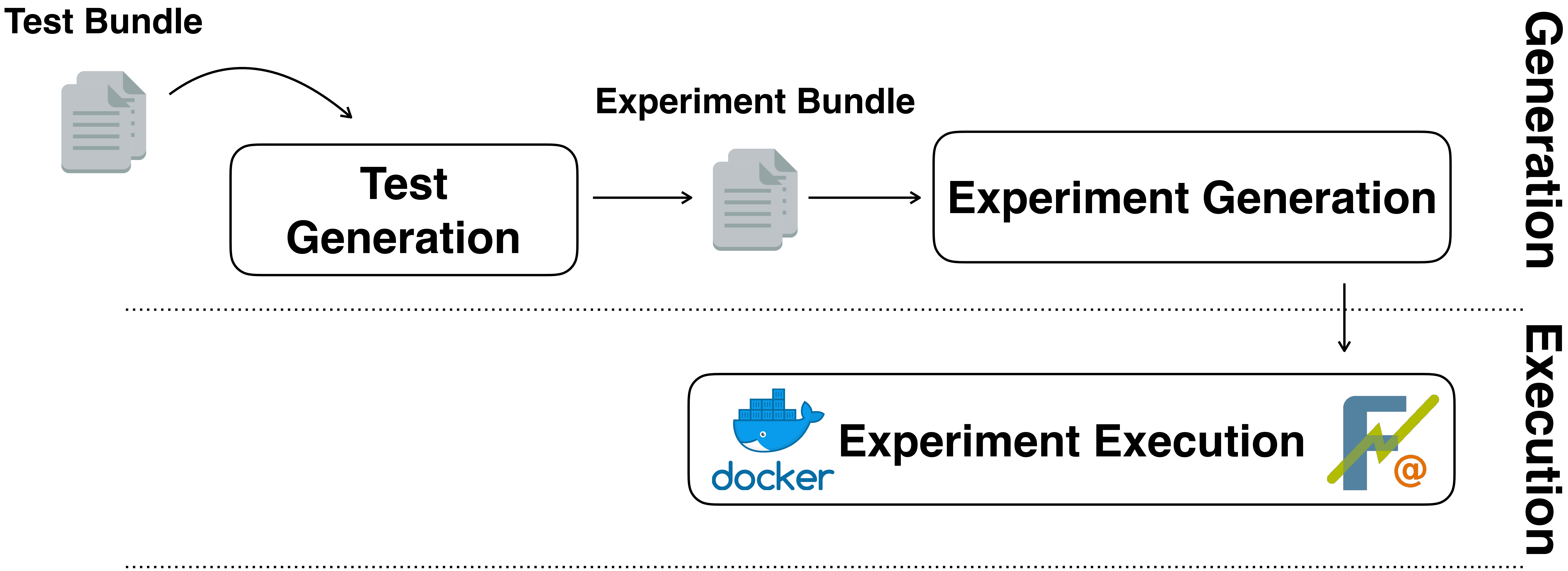
[Ferme et al.]
Vincenzo Ferme and Cesare Pautasso. A Declarative Approach for Performance Tests Execution in Continuous Software Development Environments. In Proc. of ICPE 2018. 261-272.

BenchFlow Automation Framework



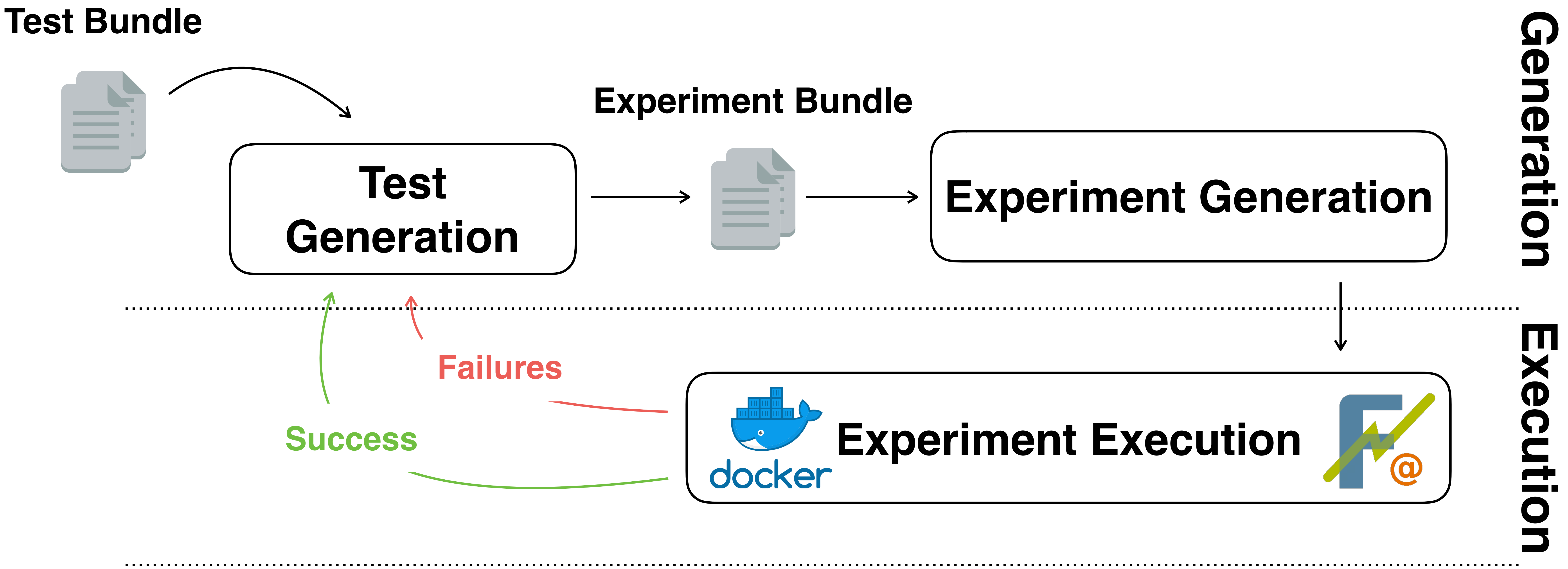
[Ferme et al.]
Vincenzo Ferme and Cesare Pautasso. A Declarative Approach for Performance Tests Execution in Continuous Software Development Environments. In Proc. of ICPE 2018. 261-272.

BenchFlow Automation Framework



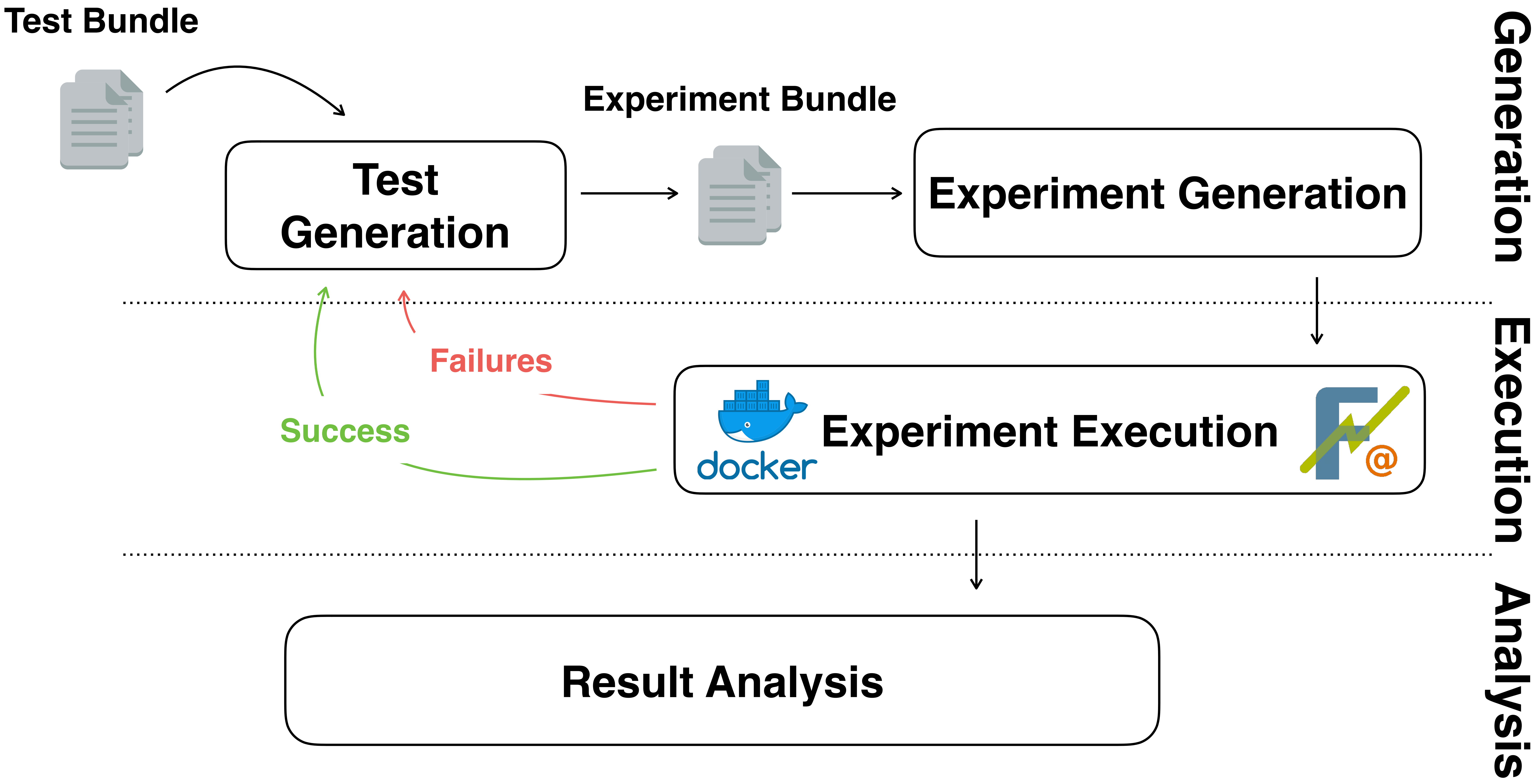
[Ferme et al.]
Vincenzo Ferme and Cesare Pautasso. A Declarative Approach for Performance Tests Execution in Continuous Software Development Environments. In Proc. of ICPE 2018. 261-272.

BenchFlow Automation Framework



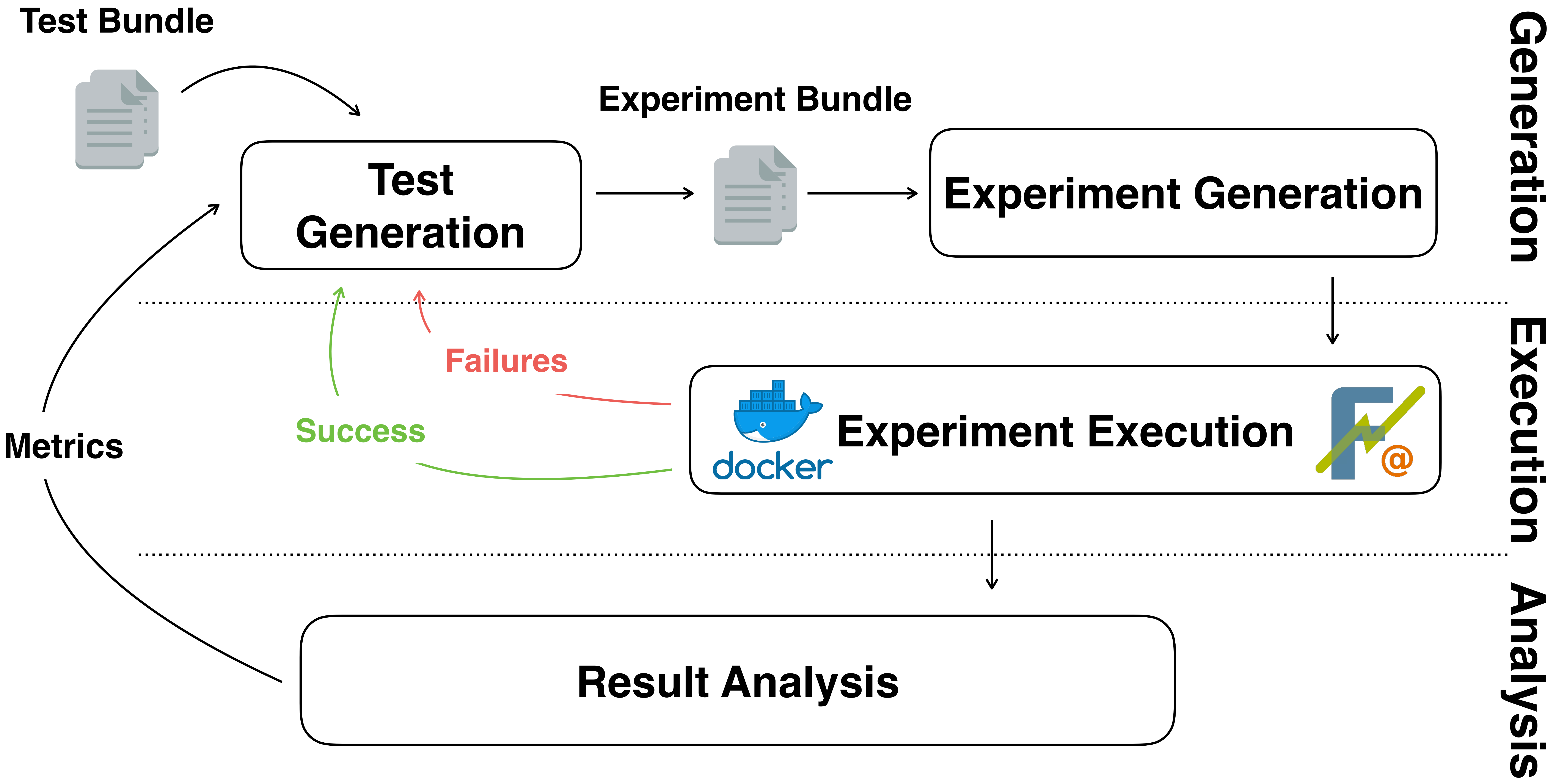
[Ferme et al.]
Vincenzo Ferme and Cesare Pautasso. A Declarative Approach for Performance Tests Execution in Continuous Software Development Environments. In Proc. of ICPE 2018. 261-272.

BenchFlow Automation Framework



[Ferme et al.]
Vincenzo Ferme and Cesare Pautasso. A Declarative Approach for Performance Tests Execution in Continuous Software Development Environments. In Proc. of ICPE 2018. 261-272.

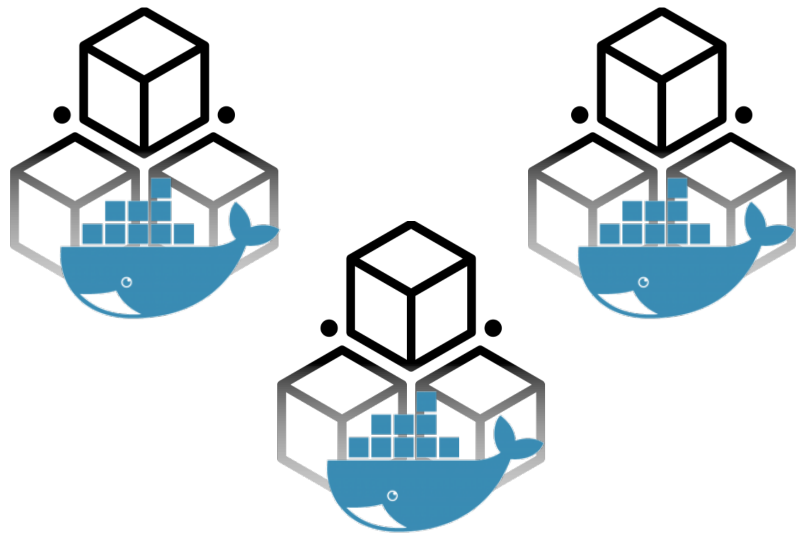
BenchFlow Automation Framework



[Ferme et al.]
Vincenzo Ferme and Cesare Pautasso. A Declarative Approach for Performance Tests Execution in Continuous Software Development Environments. In Proc. of ICPE 2018. 261-272.

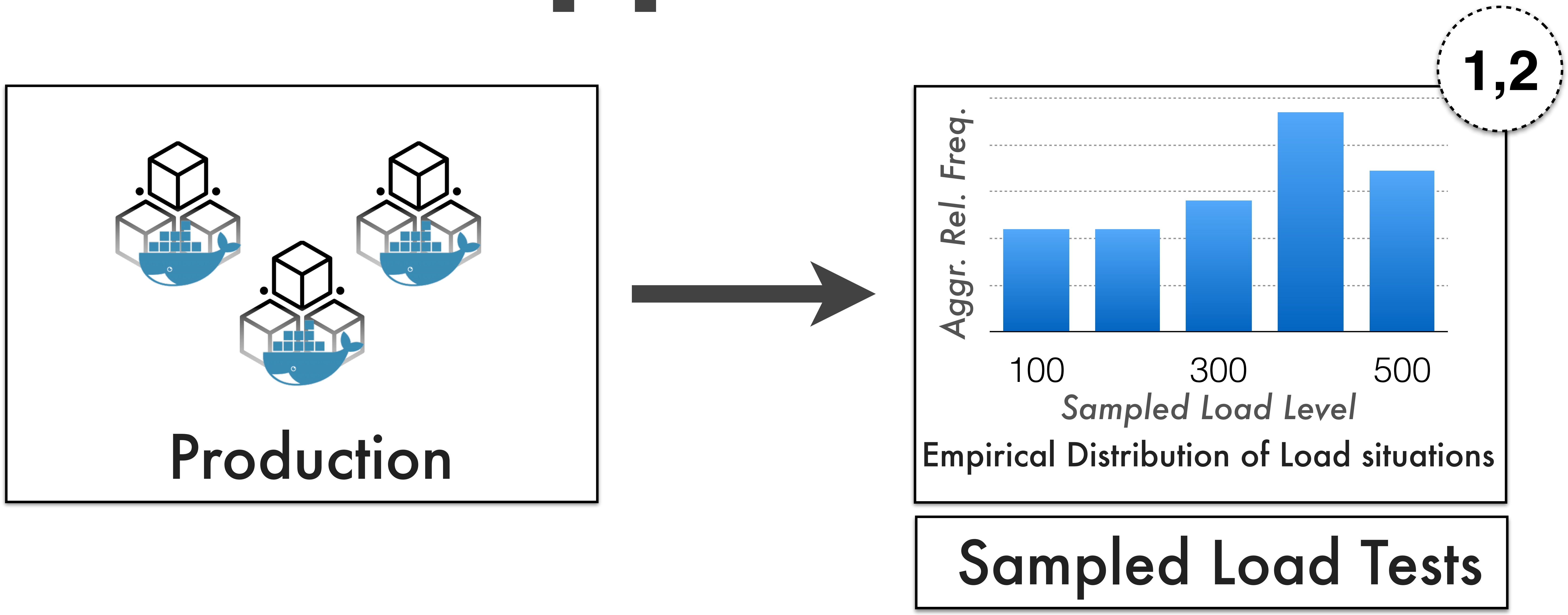
Approach: Summary

Approach: Summary



Production

Approach: Summary

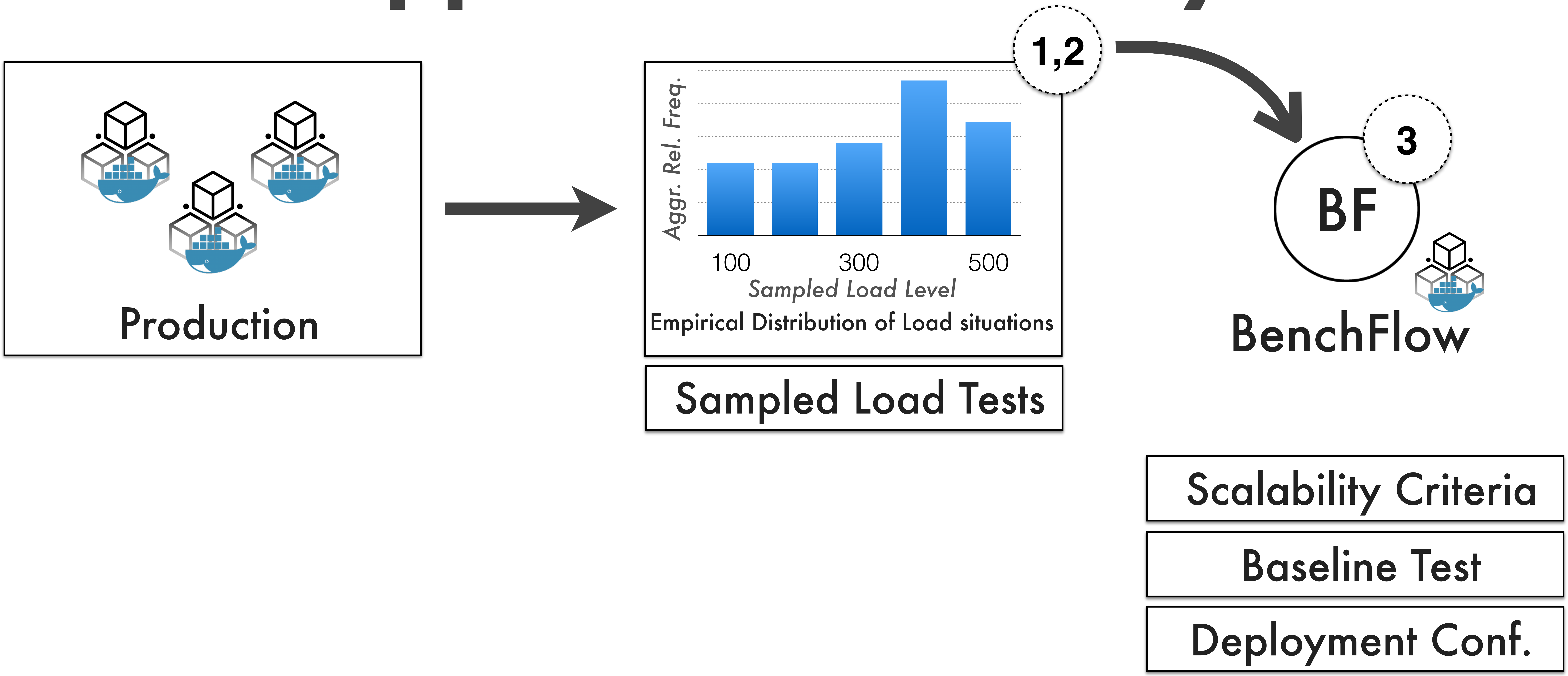


Production

Empirical Distribution of Load situations

Sampled Load Tests

Approach: Summary



Production

Empirical Distribution of Load situations

Sampled Load Tests

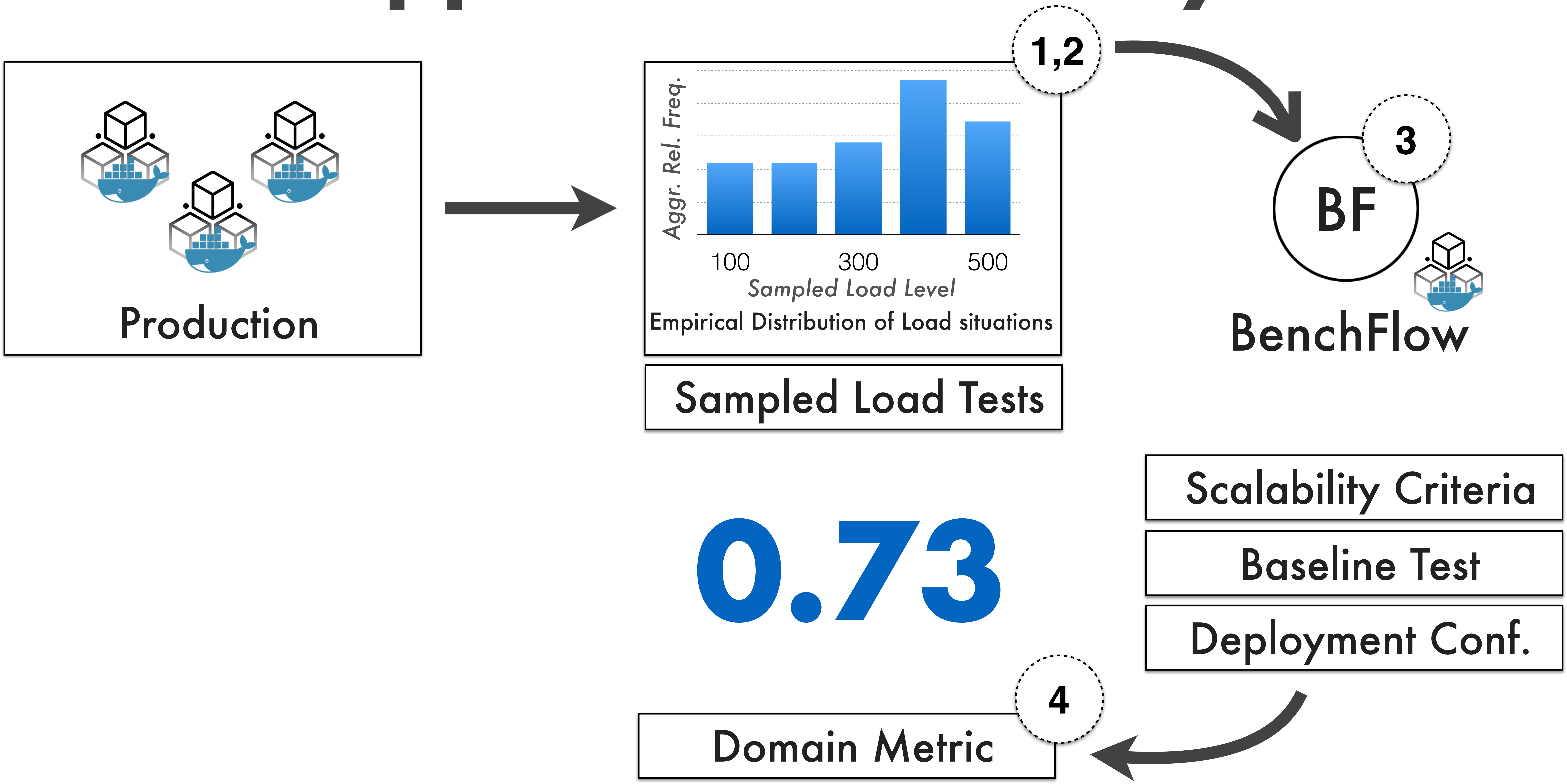
BenchFlow

Scalability Criteria

Baseline Test

Deployment Conf.

Approach: Summary



Production

Empirical Distribution of Load situations

Sampled Load Tests

BenchFlow

Scalability Criteria

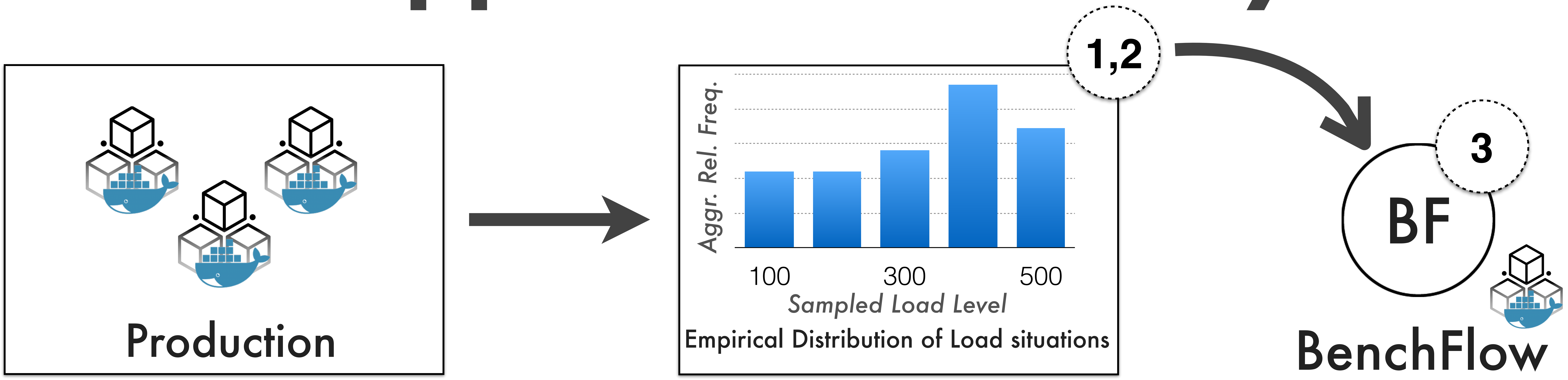
Baseline Test

Deployment Conf.

0.73

Domain Metric

Approach: Summary



Sampled Load Tests

Scalability Criteria

Baseline Test

Deployment Conf.

4

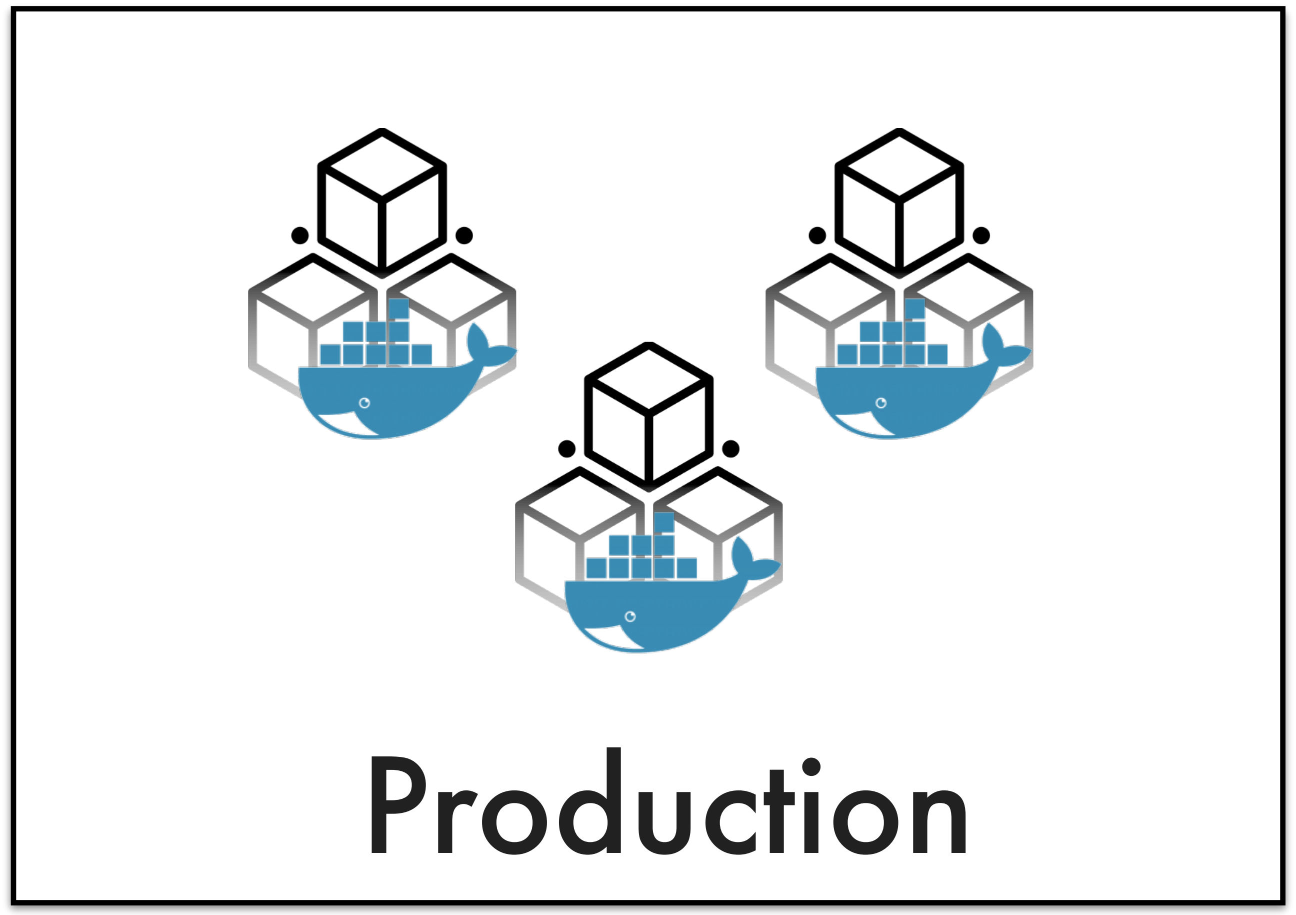
Domain Metric

Examples of Use of Domain Metric KPI:

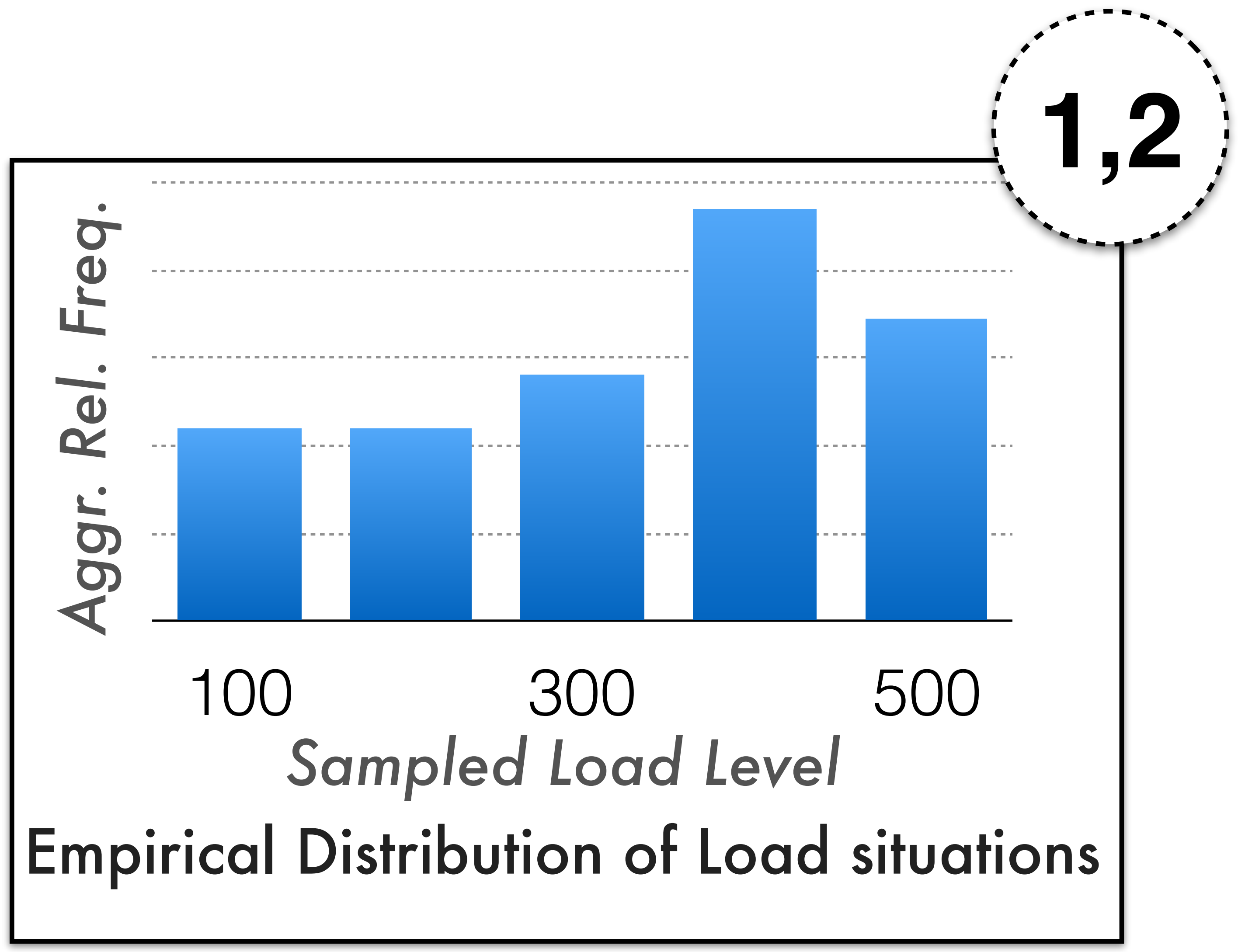
1. Decide if the system is ready for deployment

0.73

Approach: Summary

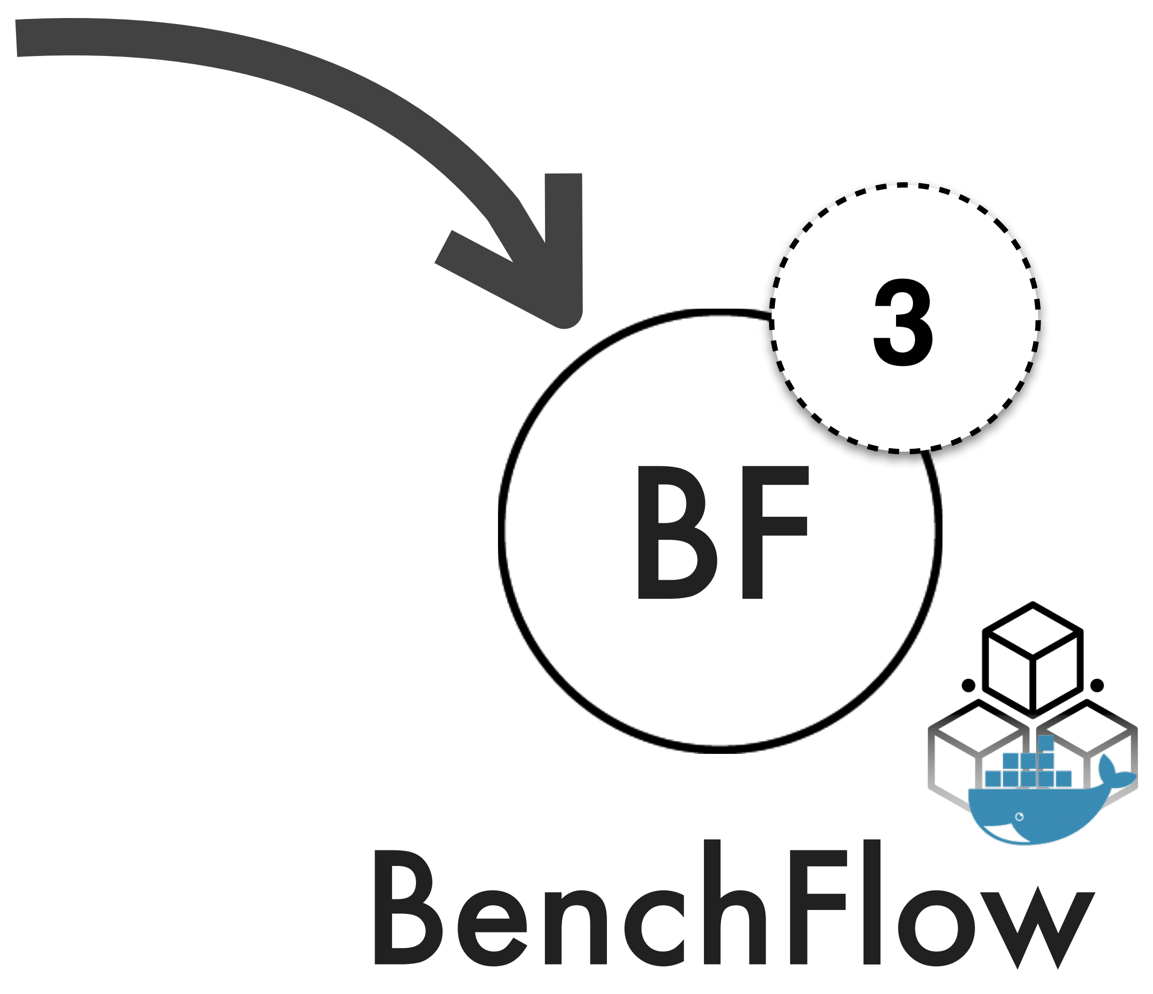


Production



Empirical Distribution of Load situations

Sampled Load Tests



BenchFlow

Examples of Use of Domain Metric KPI:

1. Decide if the system is ready for deployment
2. Assess different deployment configurations

Scalability Criteria

Baseline Test

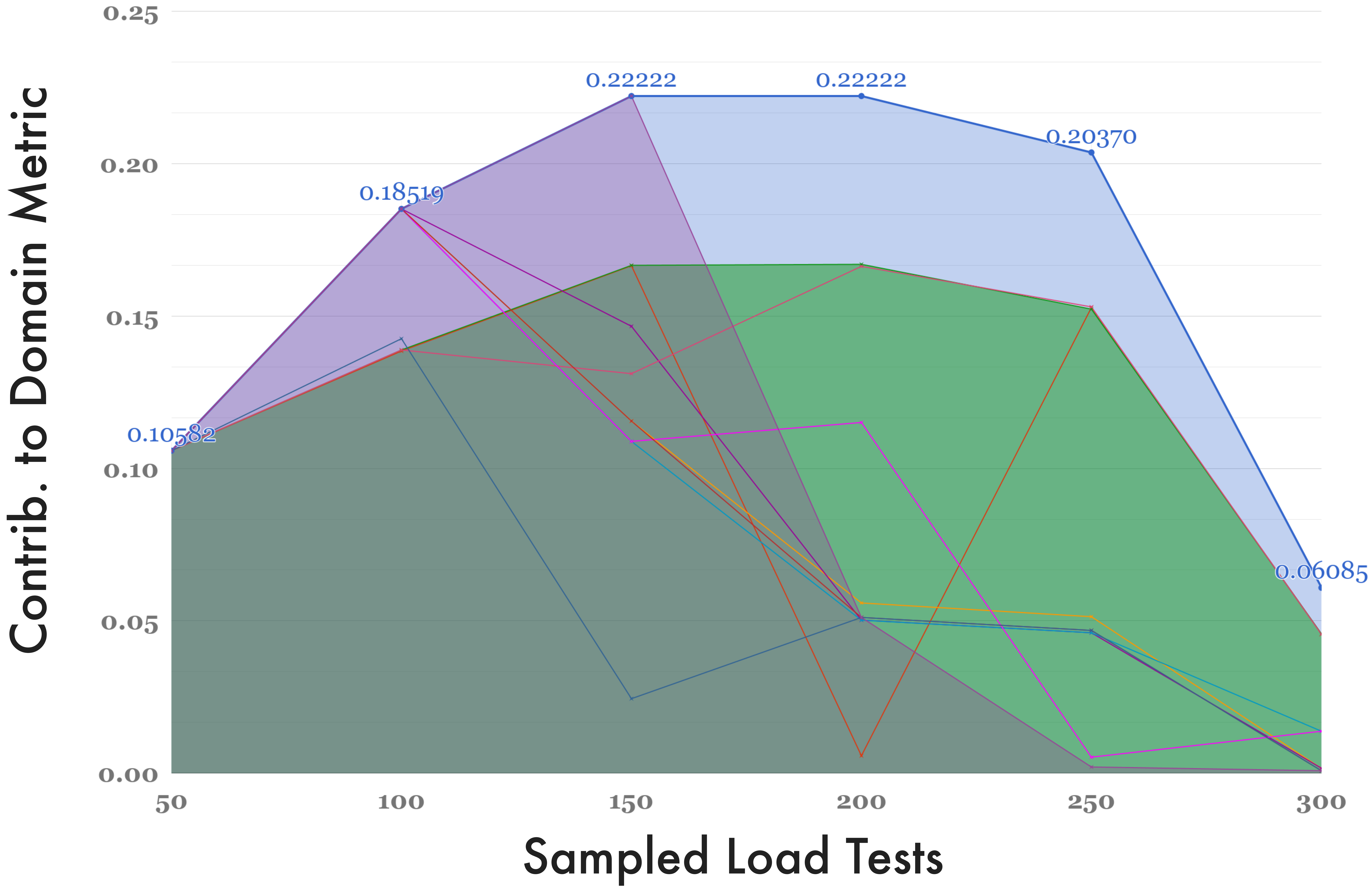
Deployment Conf.

0.73

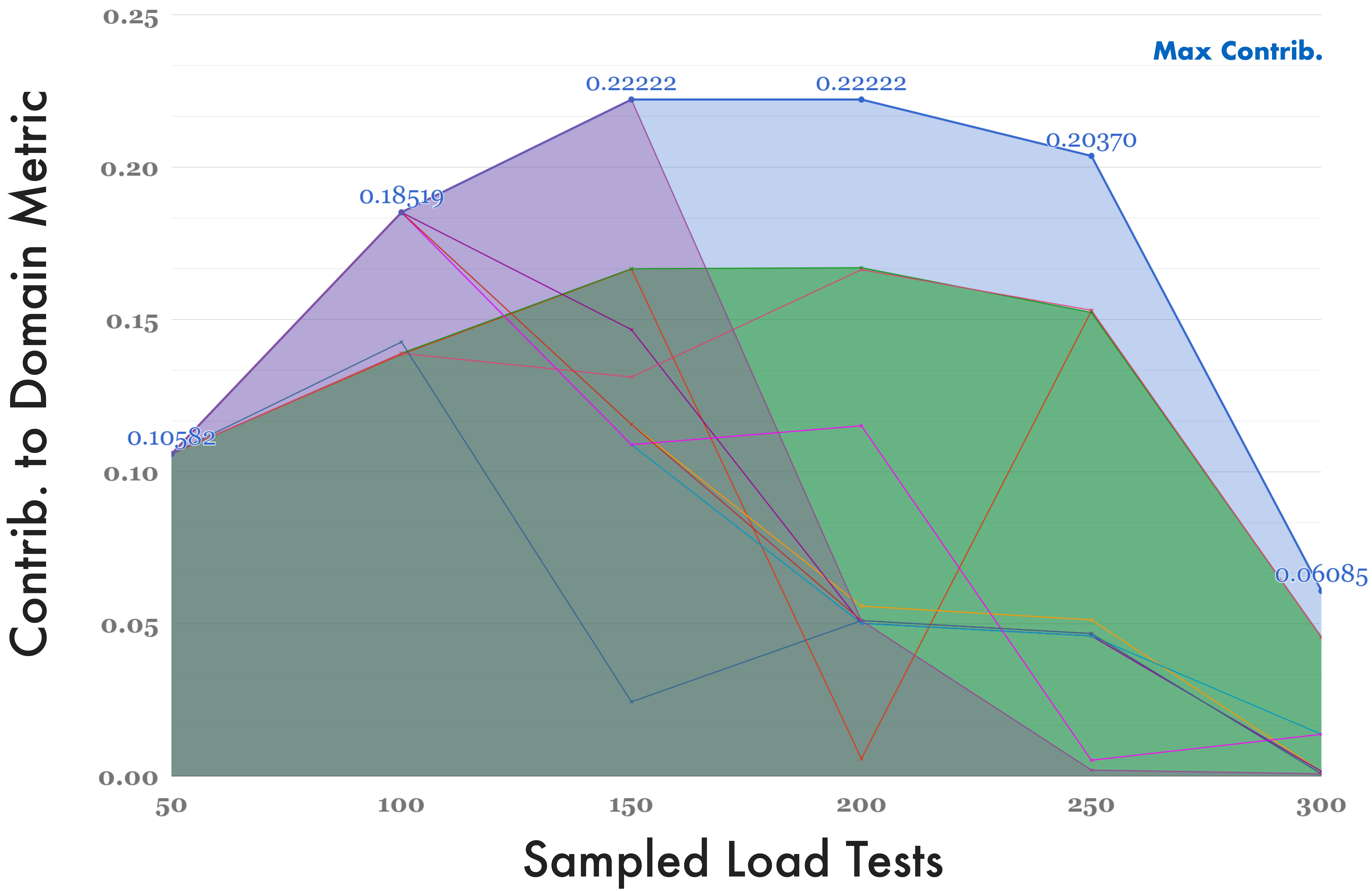
4

Domain Metric

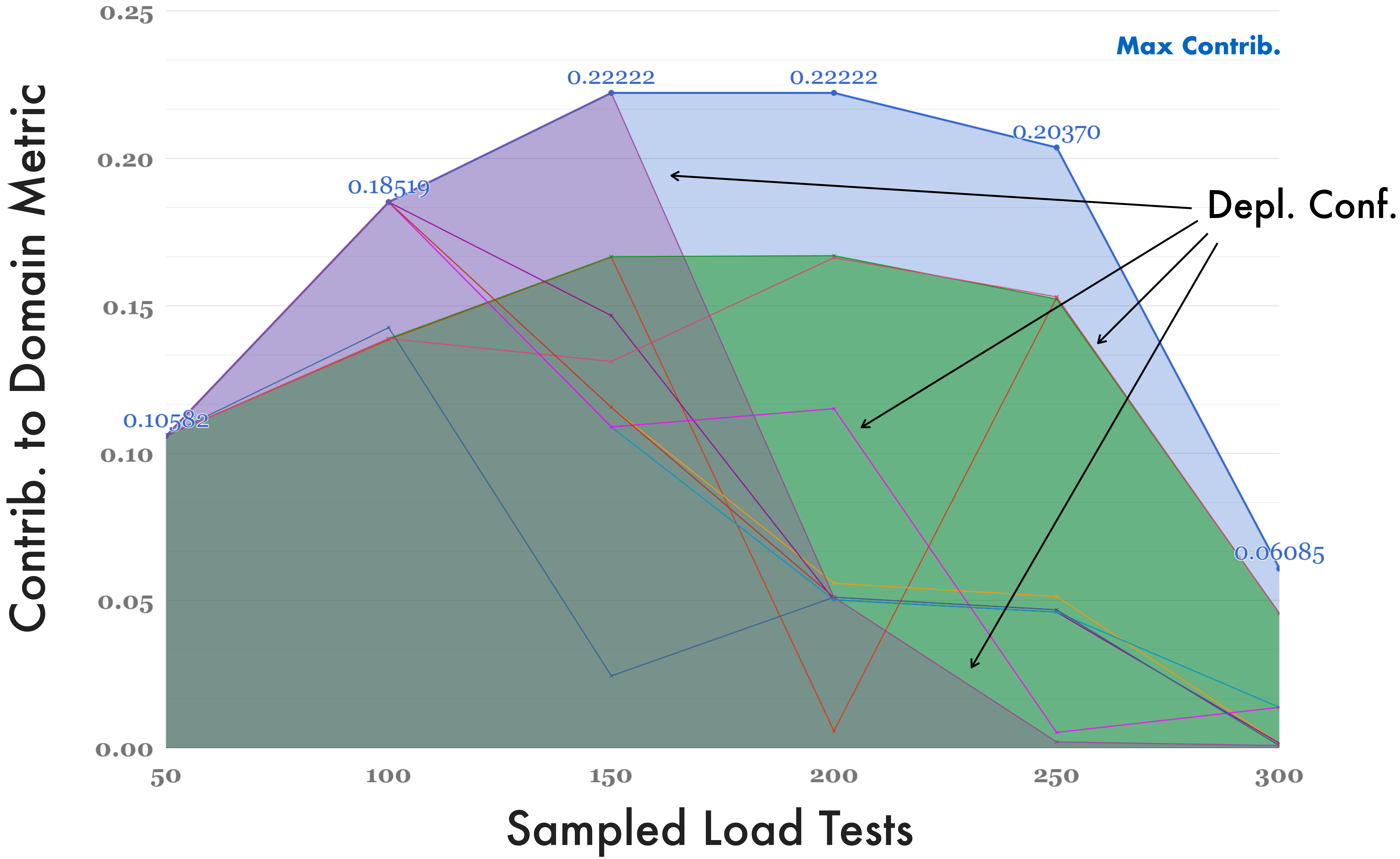
Approach: Summary



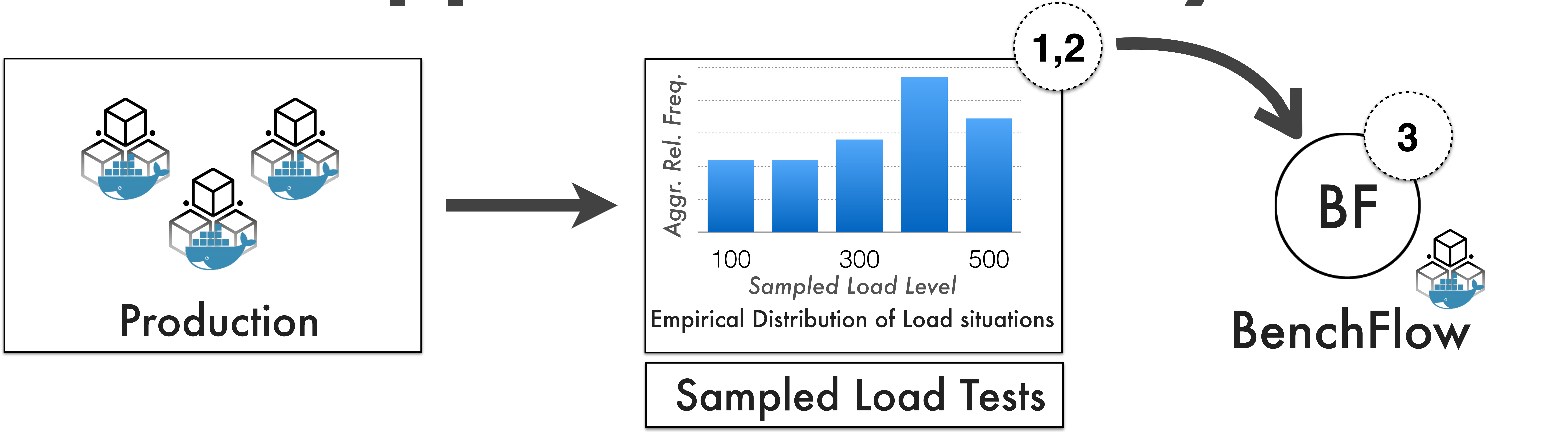
Approach: Summary



Approach: Summary



Approach: Summary



Production

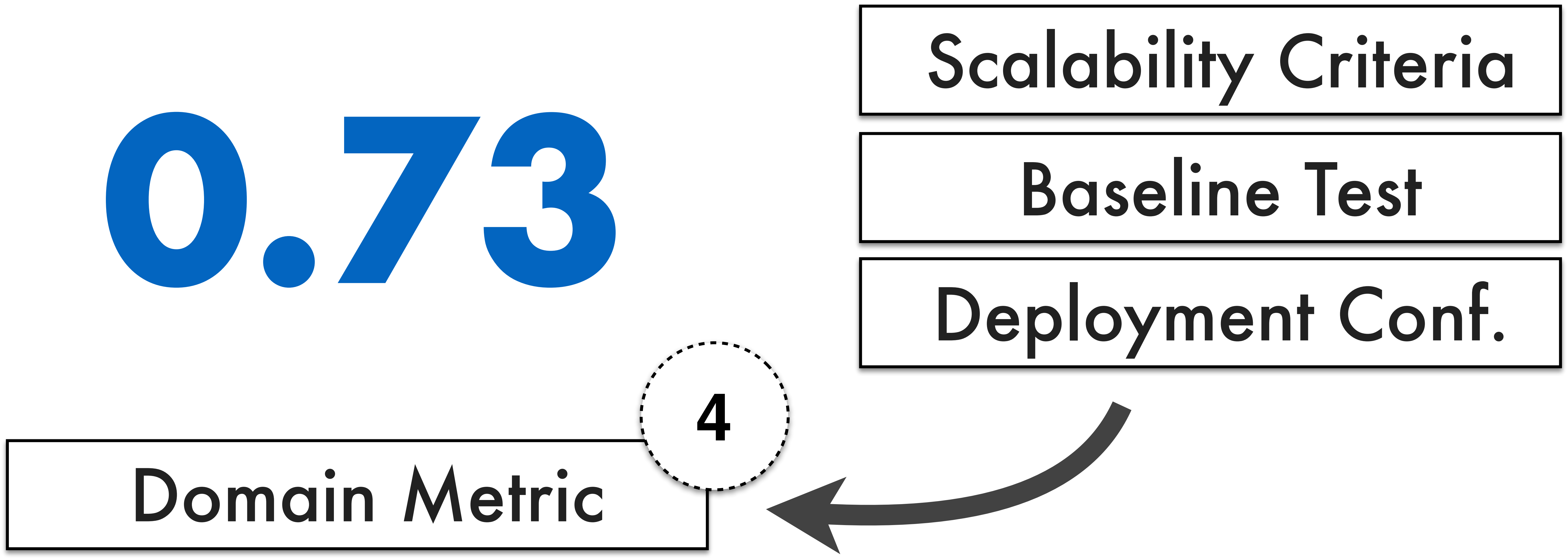
Empirical Distribution of Load situations

Sampled Load Tests

BenchFlow

Examples of Use of Domain Metric KPI:

1. Decide if the system is ready for deployment
2. Assess different deployment configurations



0.73

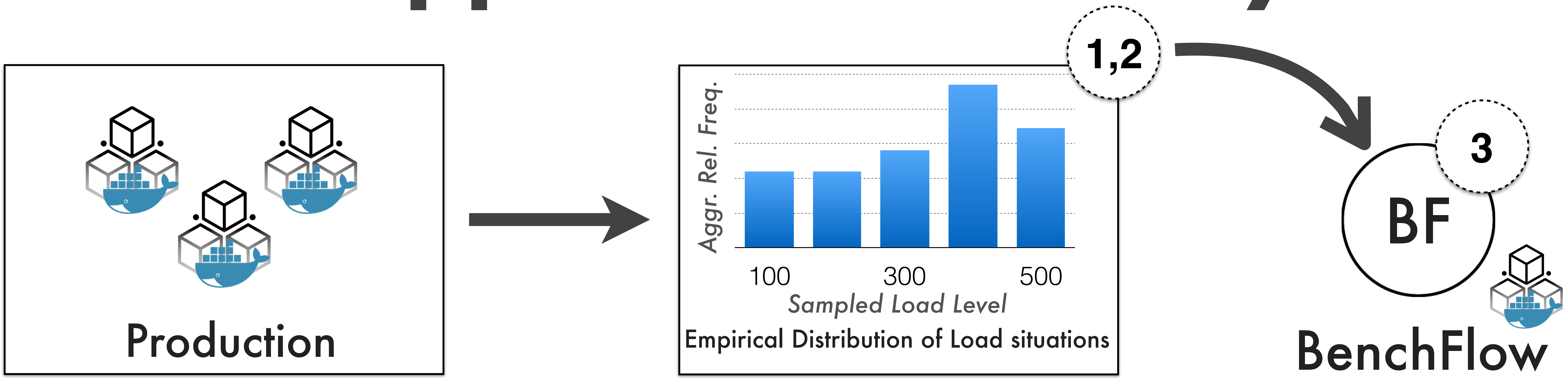
Scalability Criteria

Baseline Test

Deployment Conf.

Domain Metric

Approach: Summary



Sampled Load Tests

Examples of Use of Domain Metric KPI:

1. Decide if the system is ready for deployment
2. Assess different deployment configurations
3. Configure scaling plans and autoscalers

0.73

Scalability Criteria

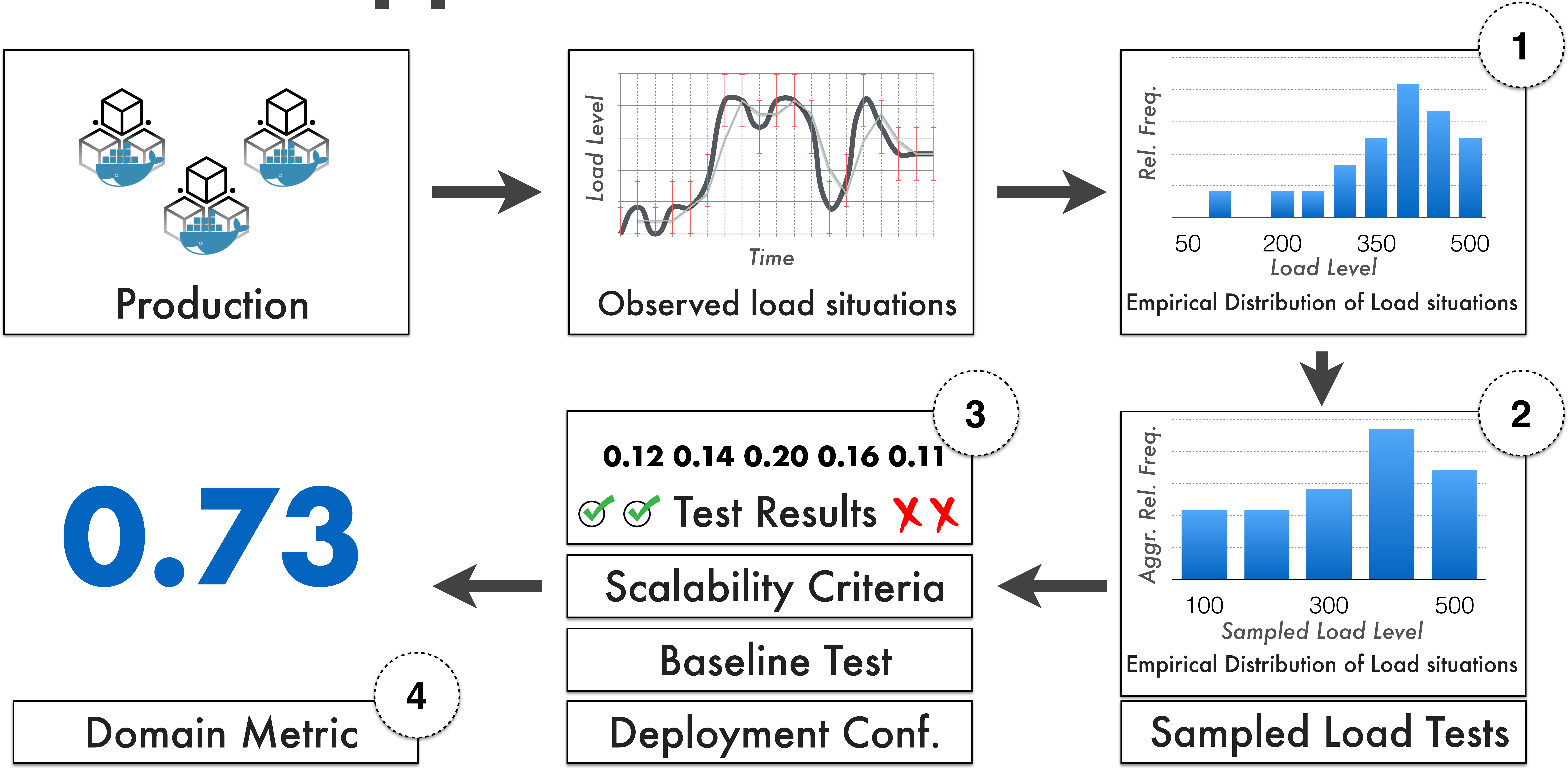
Baseline Test

Deployment Conf.

4

Domain Metric

Approach: Extensions



Production

Observed load situations

Empirical Distribution of Load situations

Empirical Distribution of Load situations

Sampled Load Tests

0.12 0.14 0.20 0.16 0.11

✓ ✓ Test Results ✗ ✗

Scalability Criteria

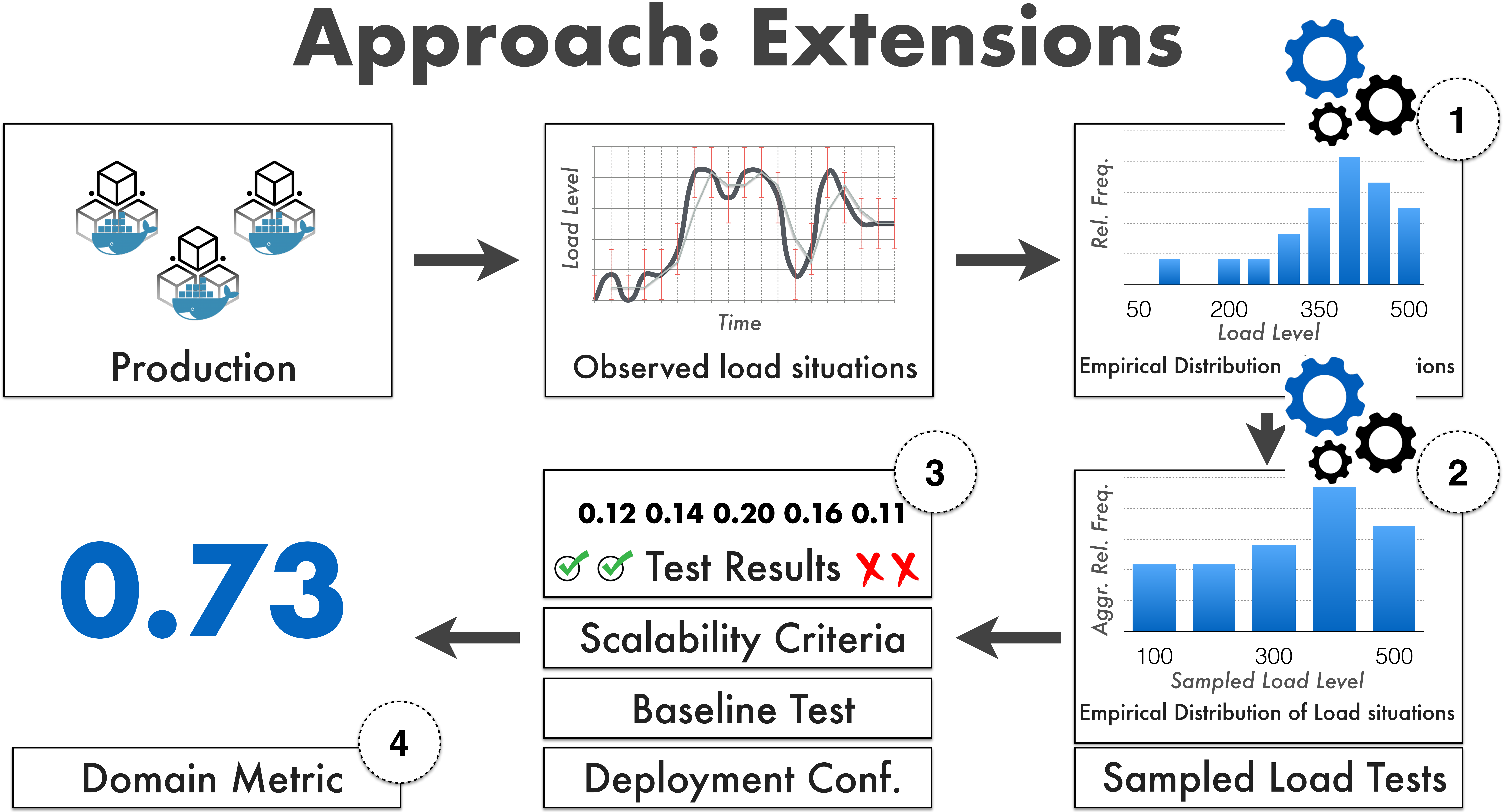
Baseline Test

Deployment Conf.

0.73

Domain Metric

Approach: Extensions



Production

Observed load situations

Empirical Distribution ions

1

2

Empirical Distribution of Load situations

Sampled Load Tests

3

0.12 0.14 0.20 0.16 0.11

✓ ✓ Test Results ✗ ✗

Scalability Criteria

Baseline Test

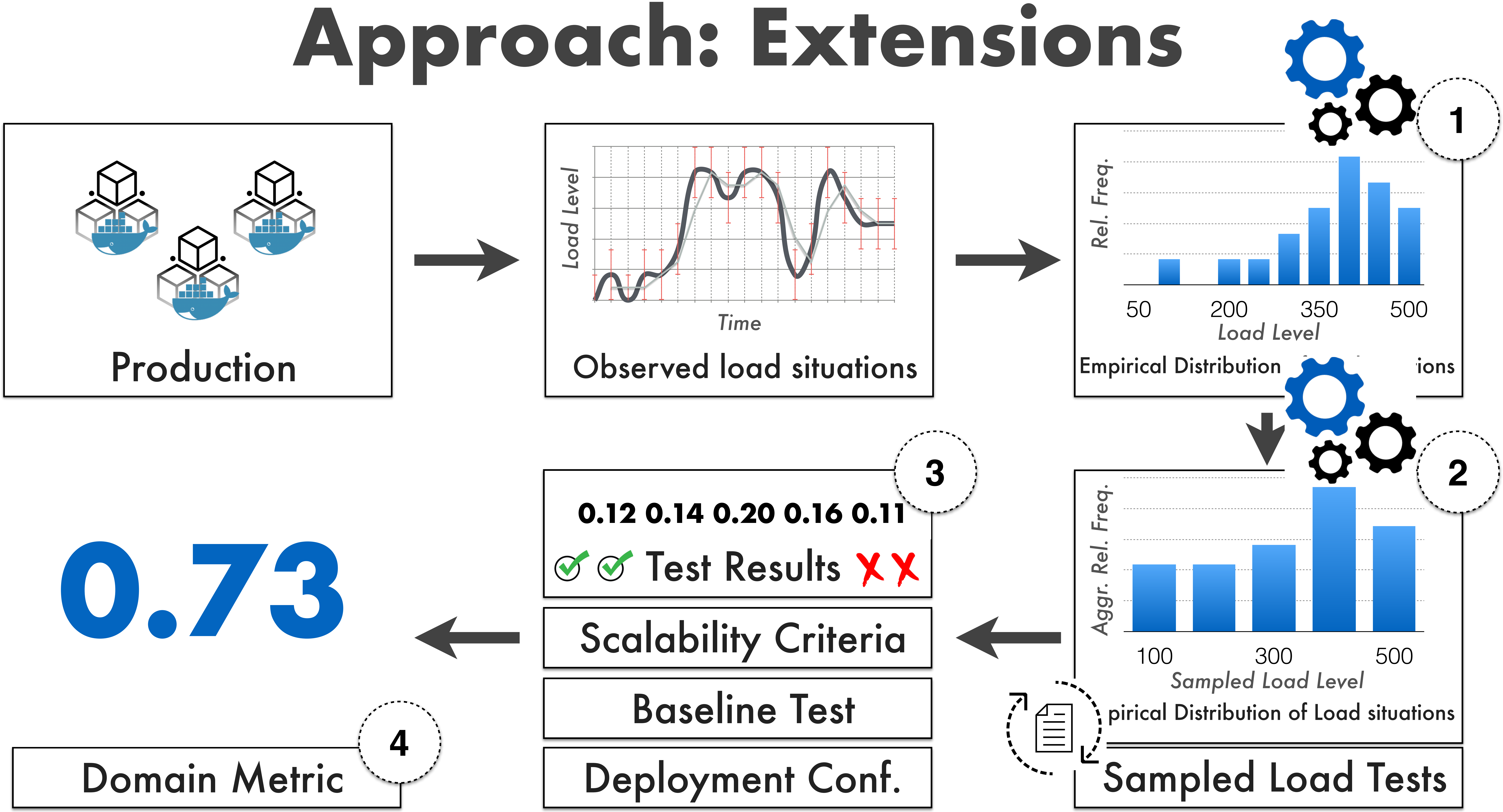
Deployment Conf.

0.73

4

Domain Metric

Approach: Extensions



Production

Observed load situations

Empirical Distribution ions

pirical Distribution of Load situations

Sampled Load Tests

0.12 0.14 0.20 0.16 0.11

✓ ✓ Test Results ✗ ✗

Scalability Criteria

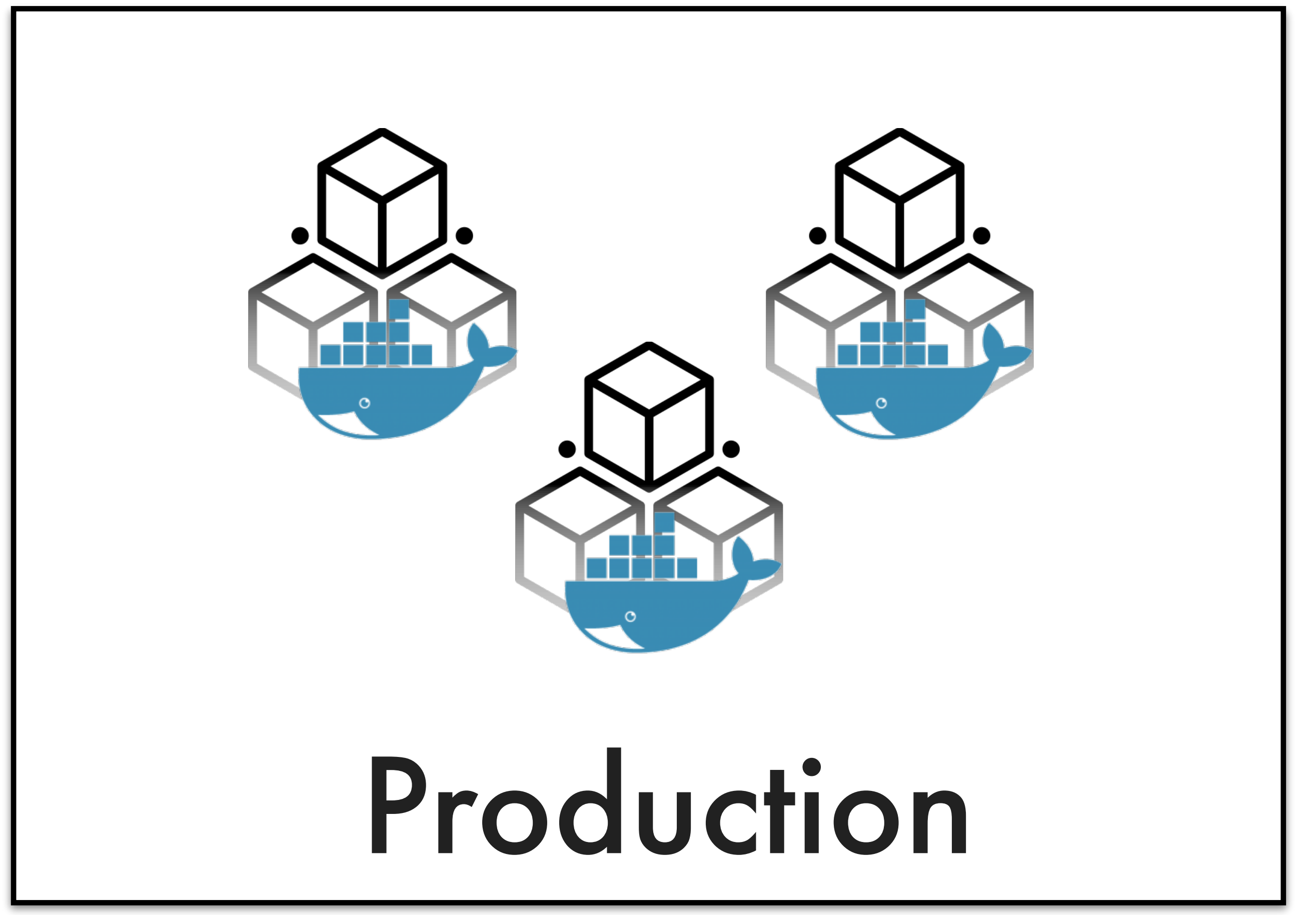
Baseline Test

Deployment Conf.

0.73

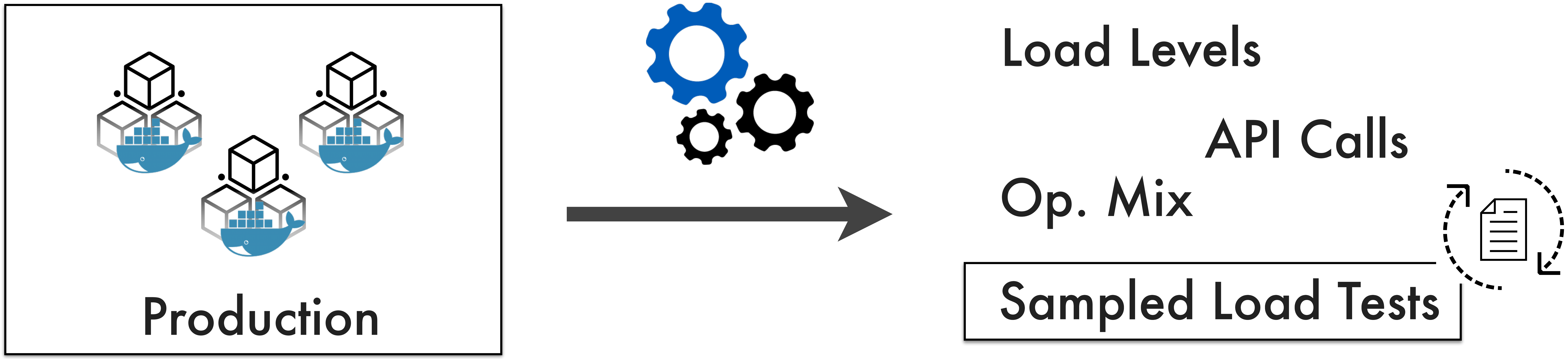
Domain Metric

ContinuITy Framework



[Schulz et al.]
Henning Schulz, Tobias Angerstein, and André van Hoorn. Towards Automating Representative Load Testing in Continuous Software Engineering. In Proc. of ICPE 2018 Companion. 123-126.

ContinuITy Framework



Production

Load Levels

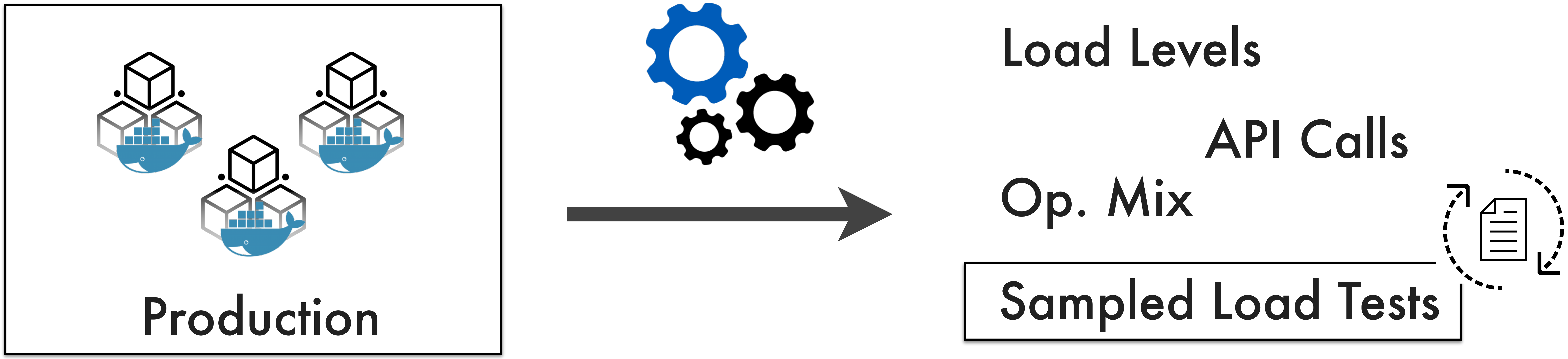
API Calls

Op. Mix

Sampled Load Tests

[Schulz et al.]
Henning Schulz, Tobias Angerstein, and André van Hoorn. Towards Automating Representative Load Testing in Continuous Software Engineering. In Proc. of ICPE 2018 Companion. 123-126.

ContinuITy Framework



Production

Load Levels

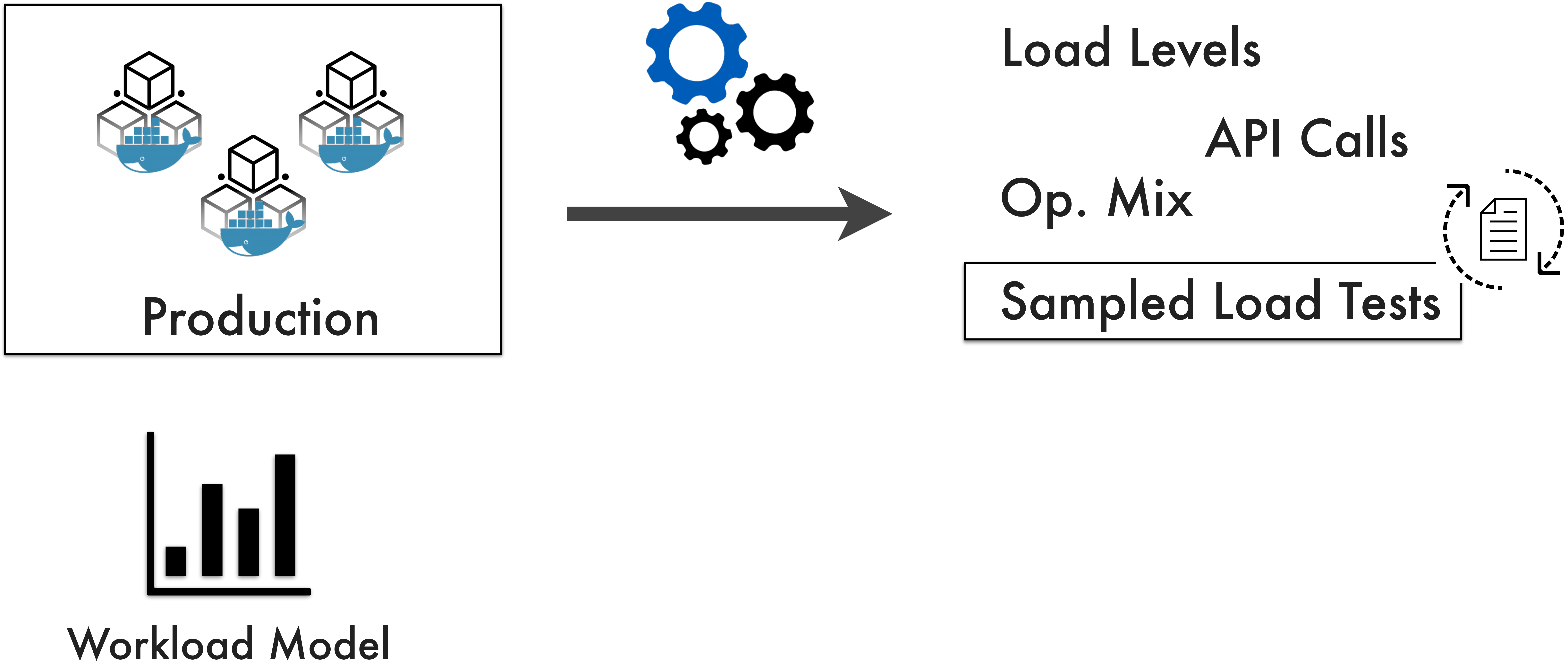
API Calls

Op. Mix

Sampled Load Tests

[Schulz et al.]
Henning Schulz, Tobias Angerstein, and André van Hoorn. Towards Automating Representative Load Testing in Continuous Software Engineering. In Proc. of ICPE 2018 Companion. 123-126.

ContinuITy Framework



Production

Load Levels

API Calls

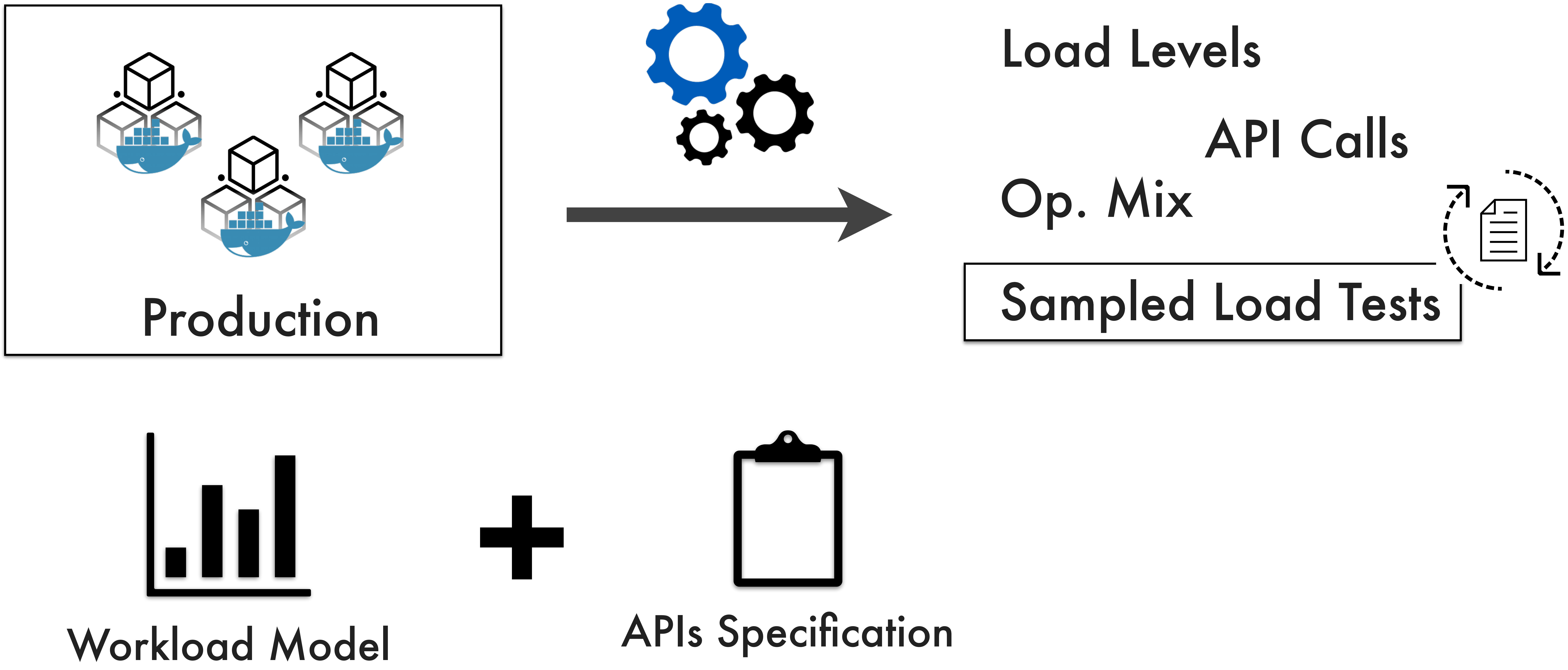
Op. Mix

Sampled Load Tests

Workload Model

[Schulz et al.]
Henning Schulz, Tobias Angerstein, and André van Hoorn. Towards Automating Representative Load Testing in Continuous Software Engineering. In Proc. of ICPE 2018 Companion. 123-126.

ContinuITy Framework



Production

Load Levels

API Calls

Op. Mix

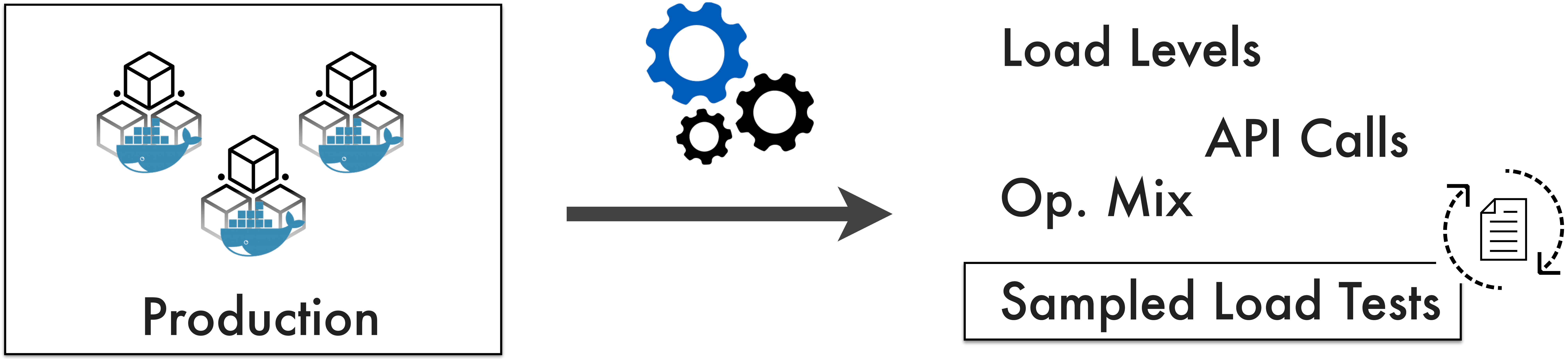
Sampled Load Tests

Workload Model

APIs Specification

[Schulz et al.]
Henning Schulz, Tobias Angerstein, and André van Hoorn. Towards Automating Representative Load Testing in Continuous Software Engineering. In Proc. of ICPE 2018 Companion. 123-126.

ContinuITy Framework



Production

Load Levels

API Calls

Op. Mix

Sampled Load Tests

Workload Model + APIs Specification + Annotations Model

[Schulz et al.]
Henning Schulz, Tobias Angerstein, and André van Hoorn. Towards Automating Representative Load Testing in Continuous Software Engineering. In Proc. of ICPE 2018 Companion. 123-126.

Future Work

Future Work

Automated update of tests definition:

- Automatically update load levels, API calls, operations mix

Future Work

Automated update of tests definition:

- Automatically update load levels, API calls, operations mix

Declarative Definitions for Load Testing:

- Make performance engineering activities more accessible

Future Work

Automated update of tests definition:

- Automatically update load levels, API calls, operations mix

Declarative Definitions for Load Testing:

- Make performance engineering activities more accessible

Declarative Performance Test Regression:

- Extend the proposed approach to continuously assess regressions in performance and PASS/FAIL CI/CD pipelines

Future Work

Automated update of tests definition:

- Automatically update load levels, API calls, operations mix

Declarative Definitions for Load Testing:

- Make performance engineering activities more accessible

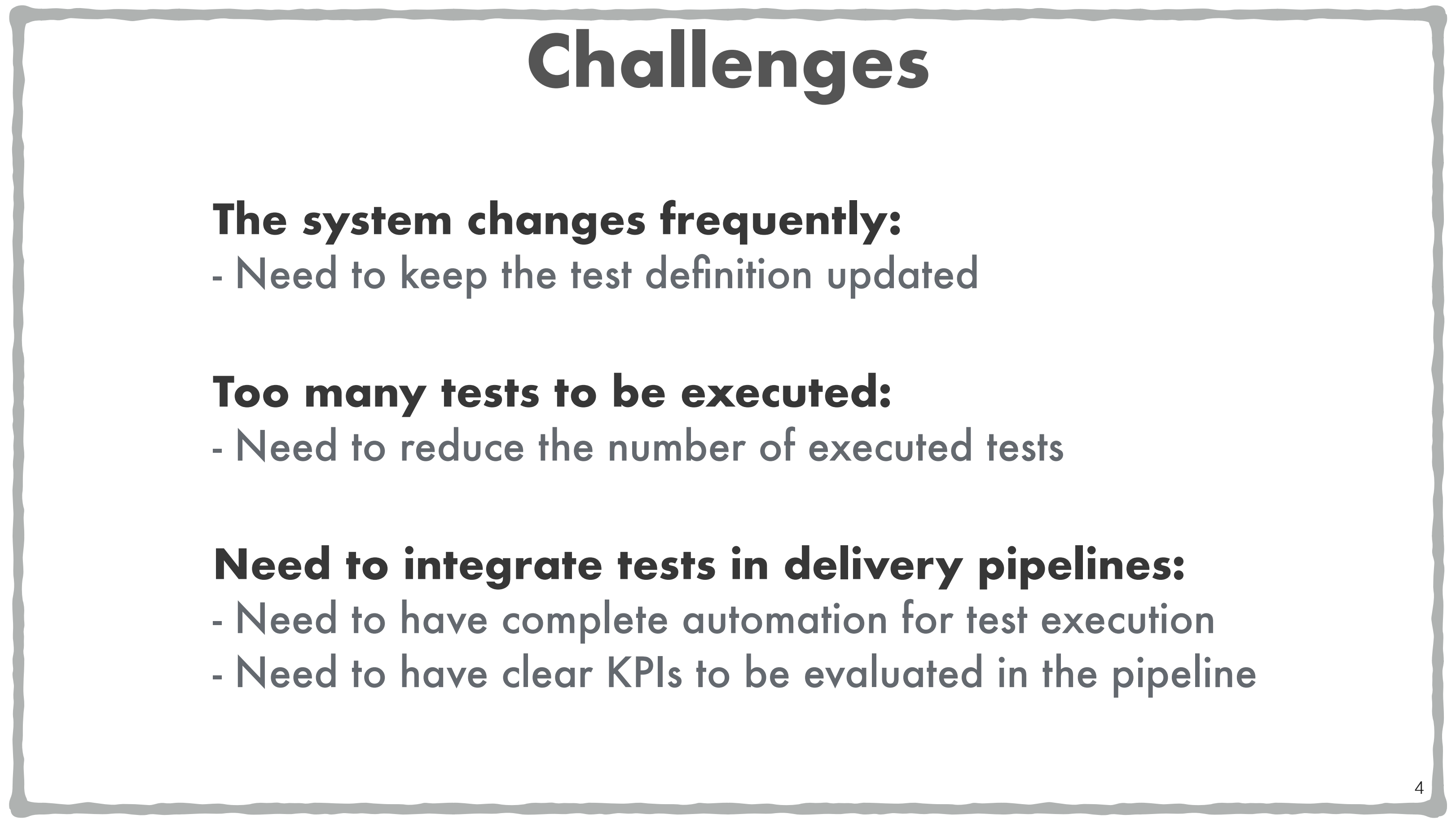
Declarative Performance Test Regression:

- Extend the proposed approach to continuously assess regressions in performance and PASS/FAIL CI/CD pipelines

Prioritisation and Selection of Load Tests:

- Exploit performance models to reduce even more the # of tests

Highlights



Challenges

The system changes frequently:
- Need to keep the test definition updated

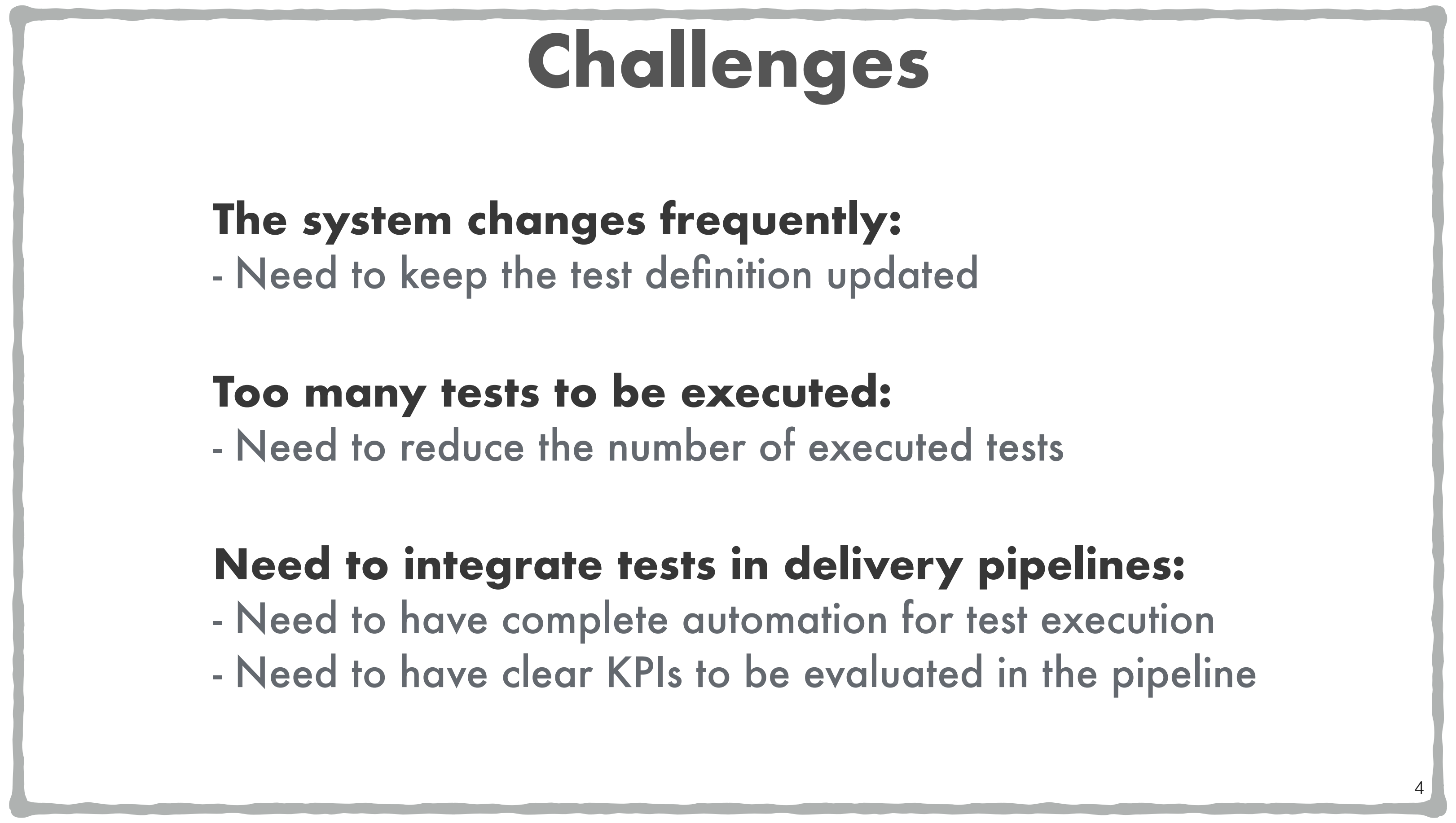
Too many tests to be executed:
- Need to reduce the number of executed tests

Need to integrate tests in delivery pipelines:
- Need to have complete automation for test execution
- Need to have clear KPIs to be evaluated in the pipeline

4

Challenges

Highlights



Challenges

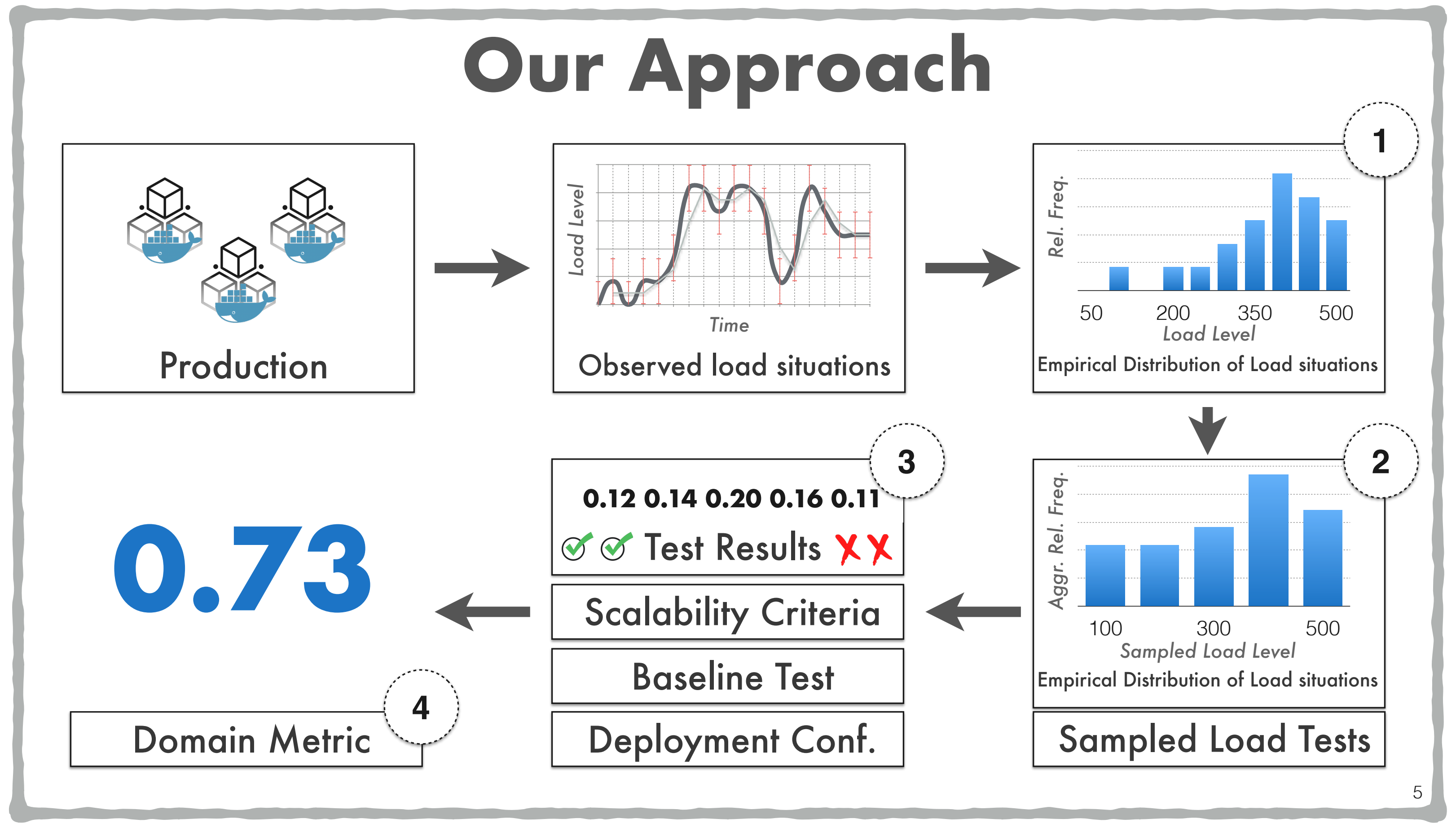
The system changes frequently:
- Need to keep the test definition updated

Too many tests to be executed:
- Need to reduce the number of executed tests

Need to integrate tests in delivery pipelines:
- Need to have complete automation for test execution
- Need to have clear KPIs to be evaluated in the pipeline

4

Challenges

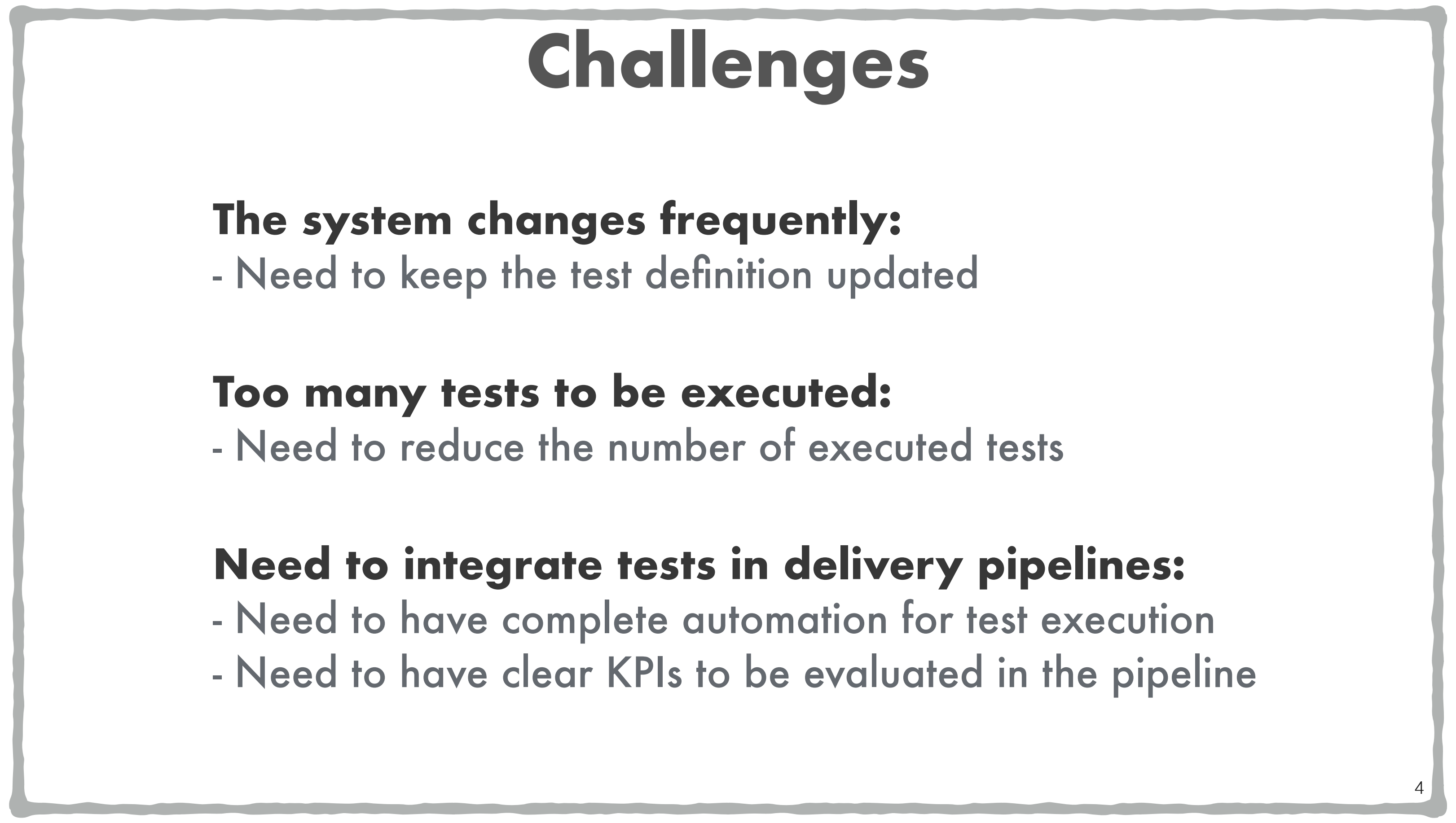


Our Approach

Production → Observed load situations → Empirical Distribution of Load situations (1) → Empirical Distribution of Load situations (2) / Sampled Load Tests → Test Results / Scalability Criteria / Baseline Test / Deployment Conf. (3) → 0.73 / Domain Metric (4)

Our Approach

Highlights



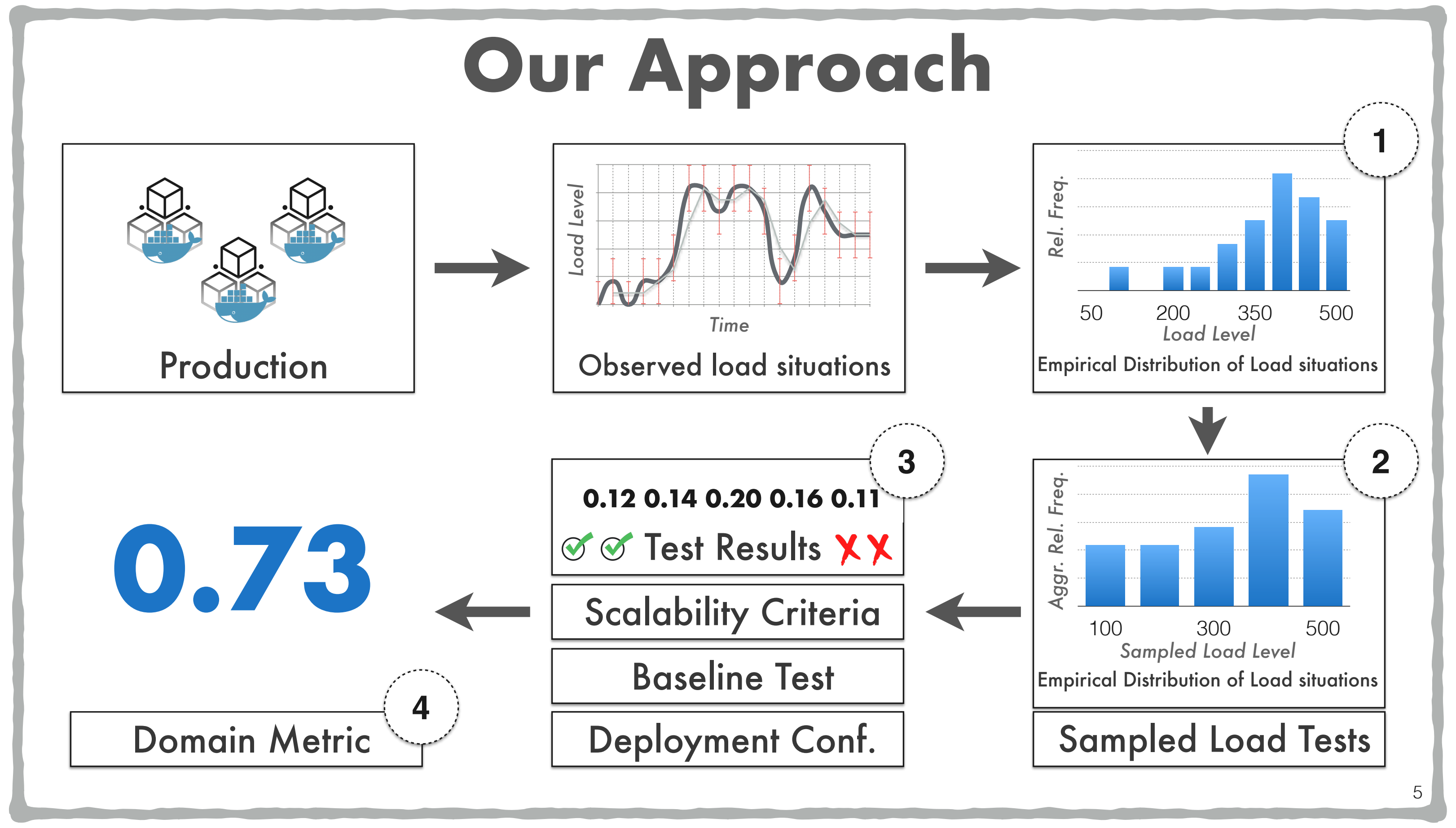
Challenges

The system changes frequently:
- Need to keep the test definition updated

Too many tests to be executed:
- Need to reduce the number of executed tests

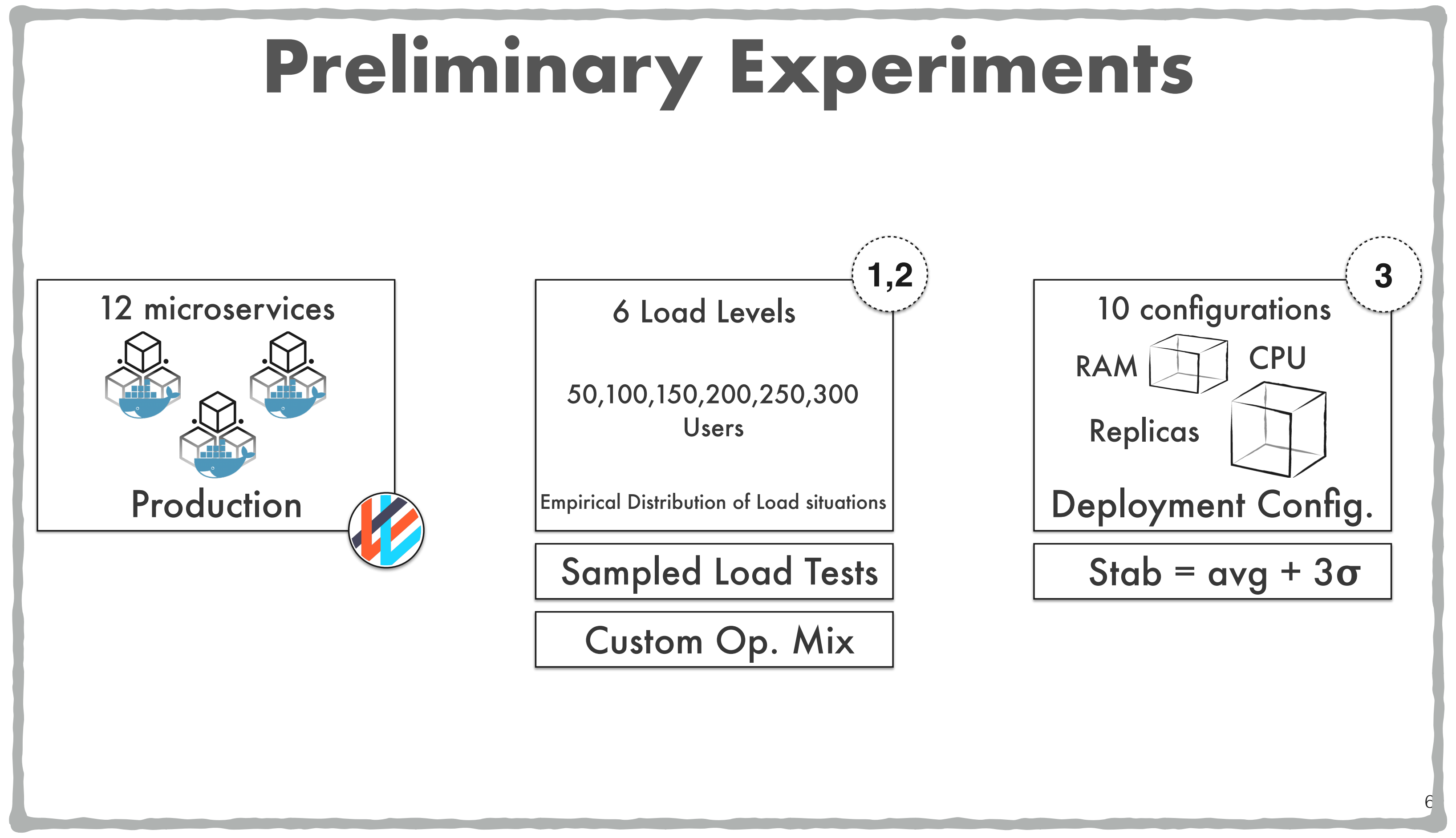
Need to integrate tests in delivery pipelines:
- Need to have complete automation for test execution
- Need to have clear KPIs to be evaluated in the pipeline

Challenges



Our Approach

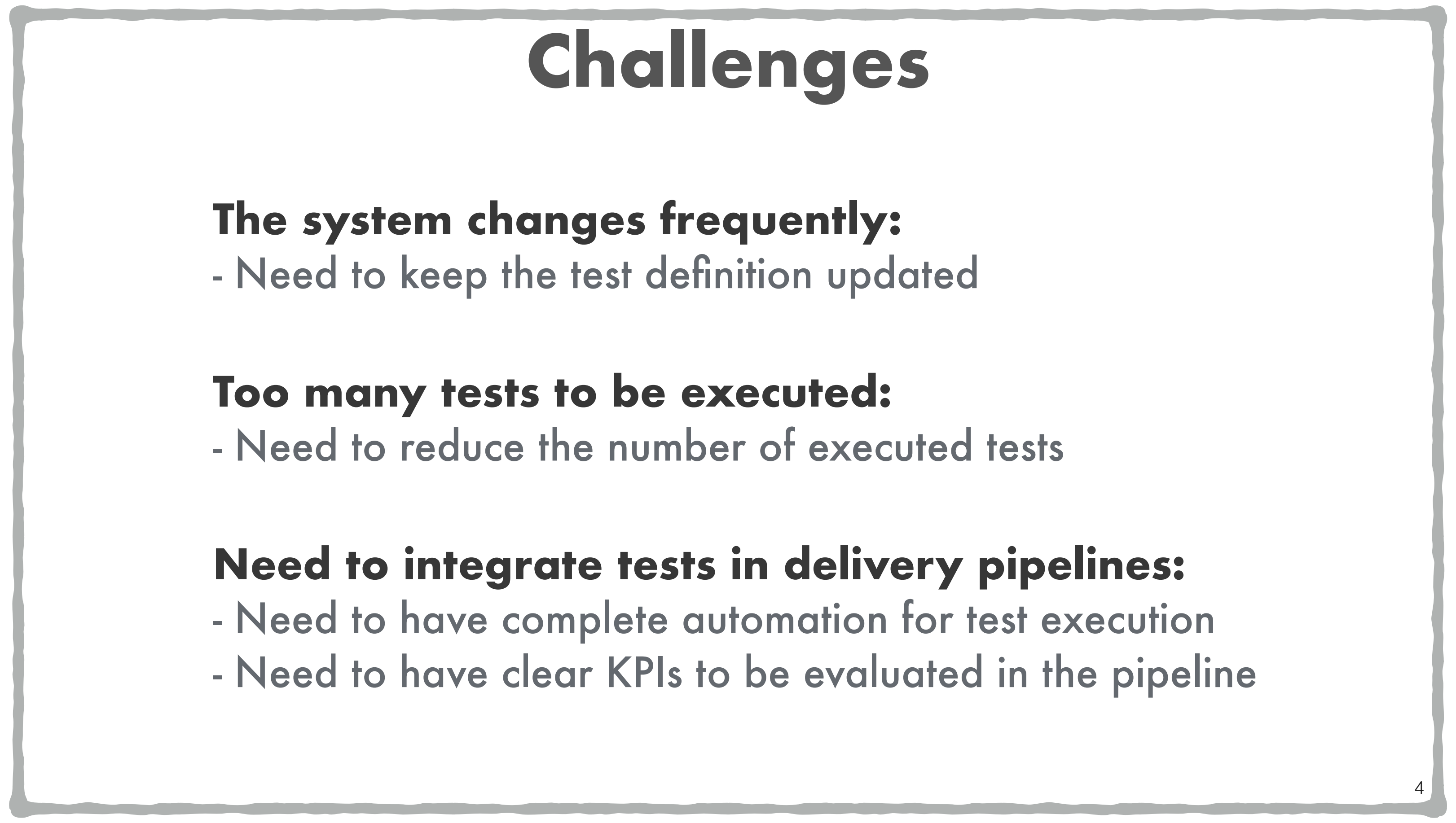
Our Approach



Preliminary Experiments

Preliminary Experiments

Highlights



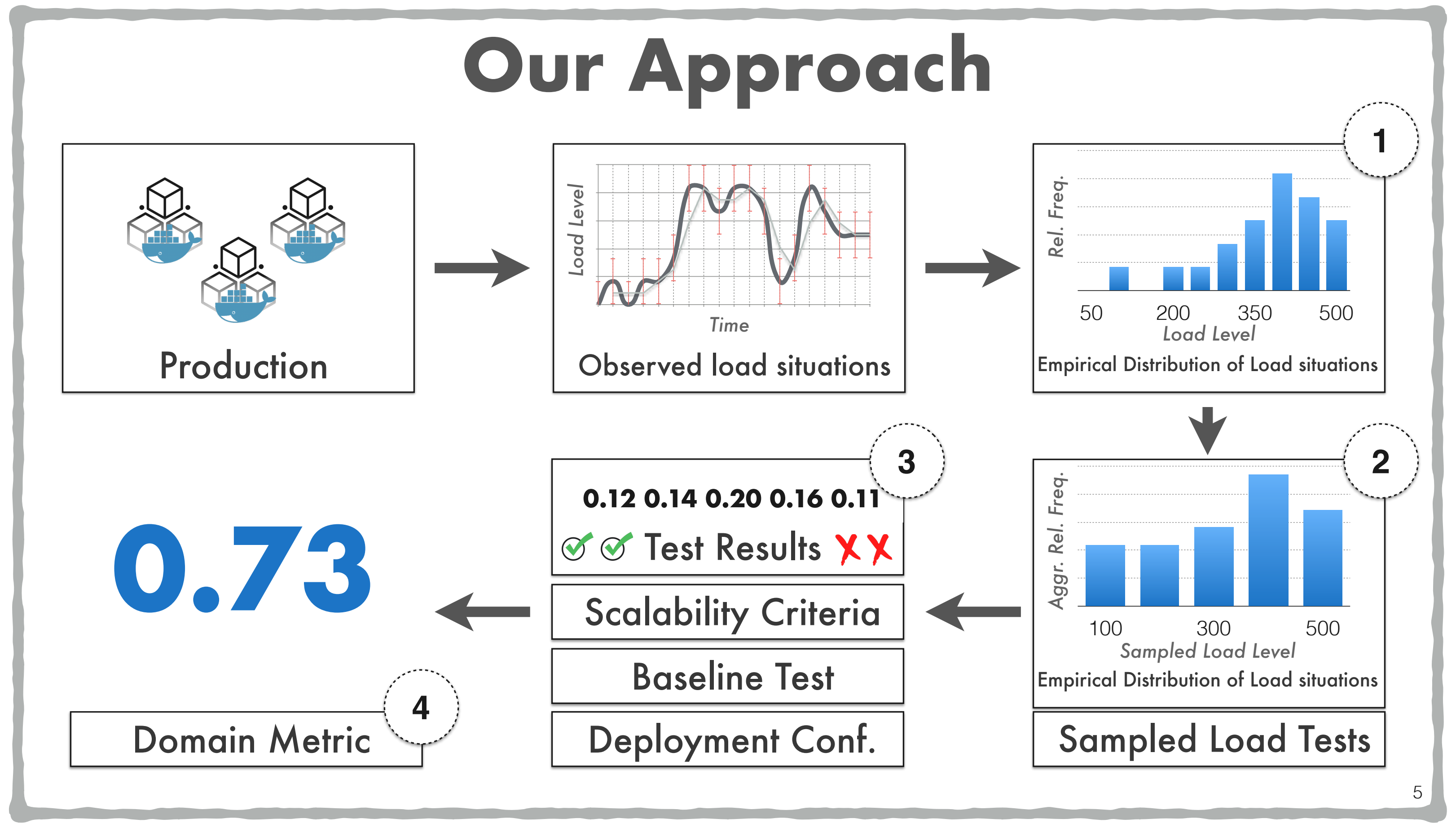
Challenges

The system changes frequently:
- Need to keep the test definition updated

Too many tests to be executed:
- Need to reduce the number of executed tests

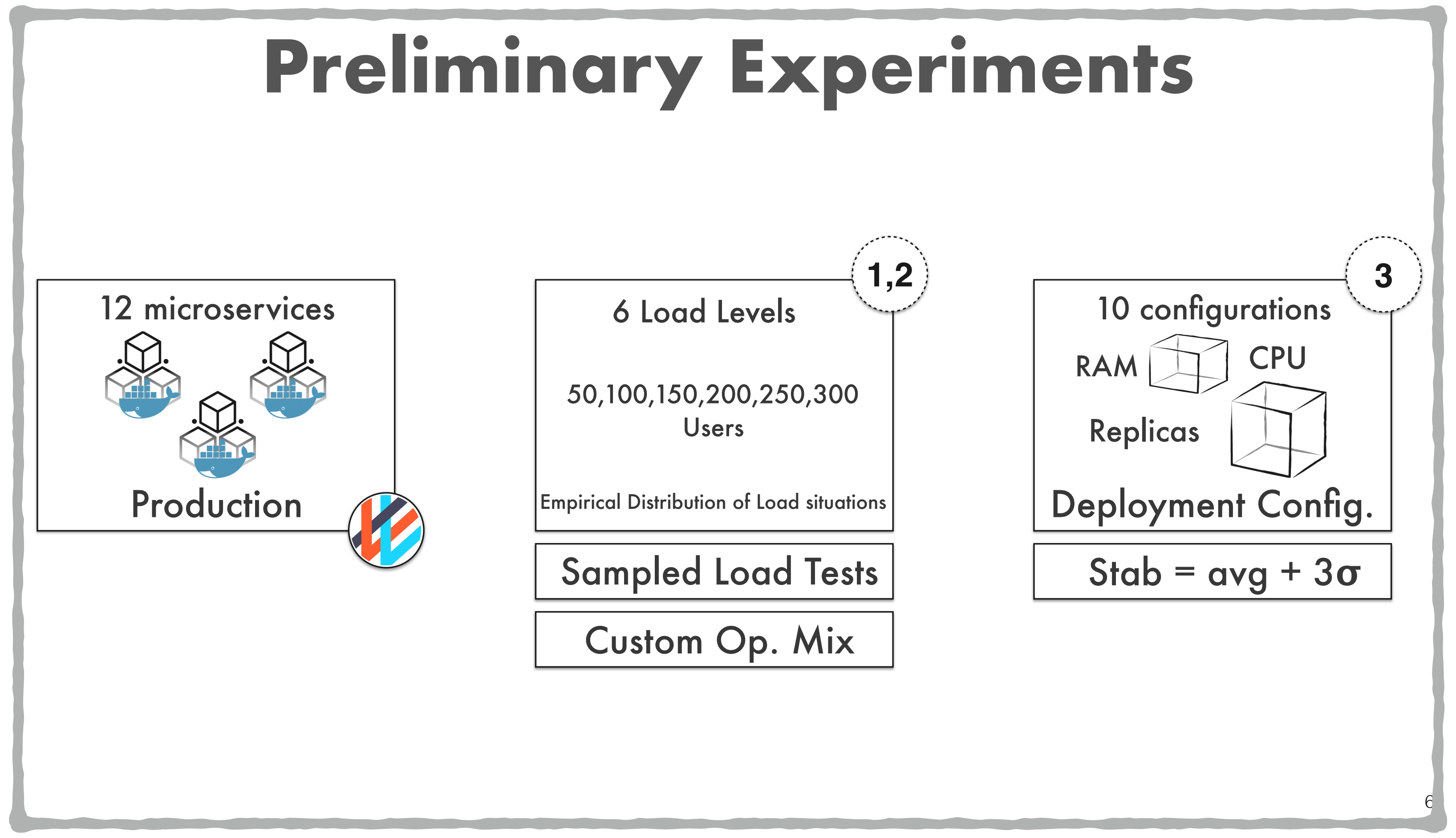
Need to integrate tests in delivery pipelines:
- Need to have complete automation for test execution
- Need to have clear KPIs to be evaluated in the pipeline

Challenges



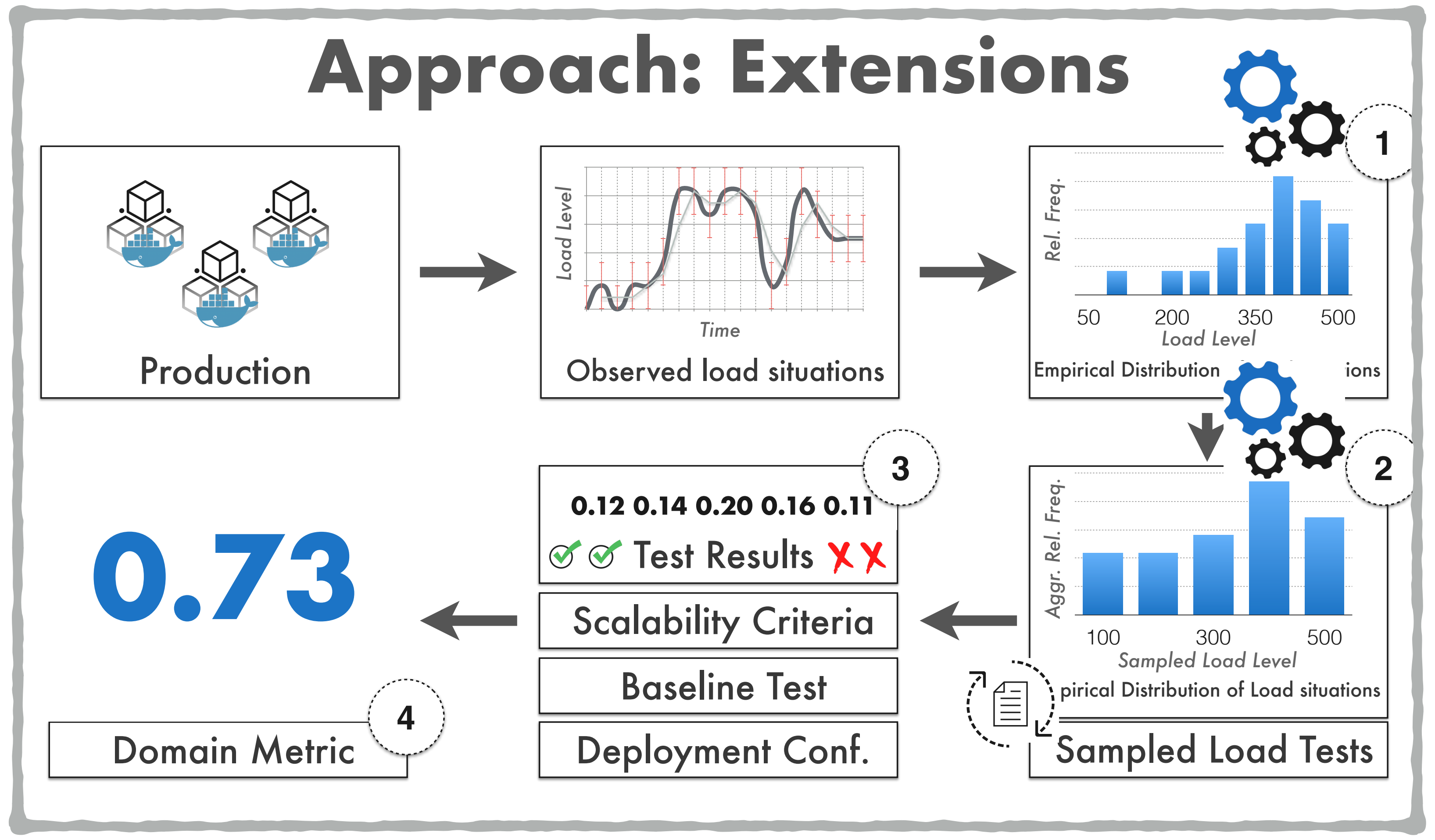
Our Approach

Our Approach



Preliminary Experiments

Preliminary Experiments



Approach: Extensions

Extensions and Future Work