

University of Stuttgart
Germany



Università
della
Svizzera
italiana

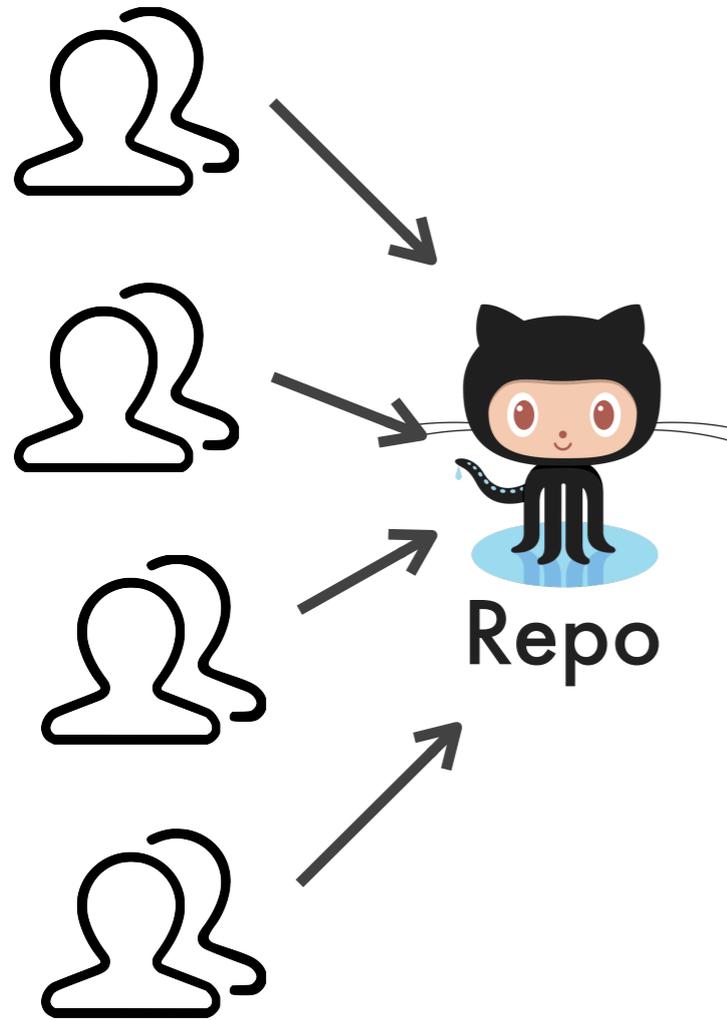


A Declarative Approach for Performance Tests Execution in Continuous Software Development Environments

**Software Institute
Faculty of Informatics
USI Lugano, Switzerland**

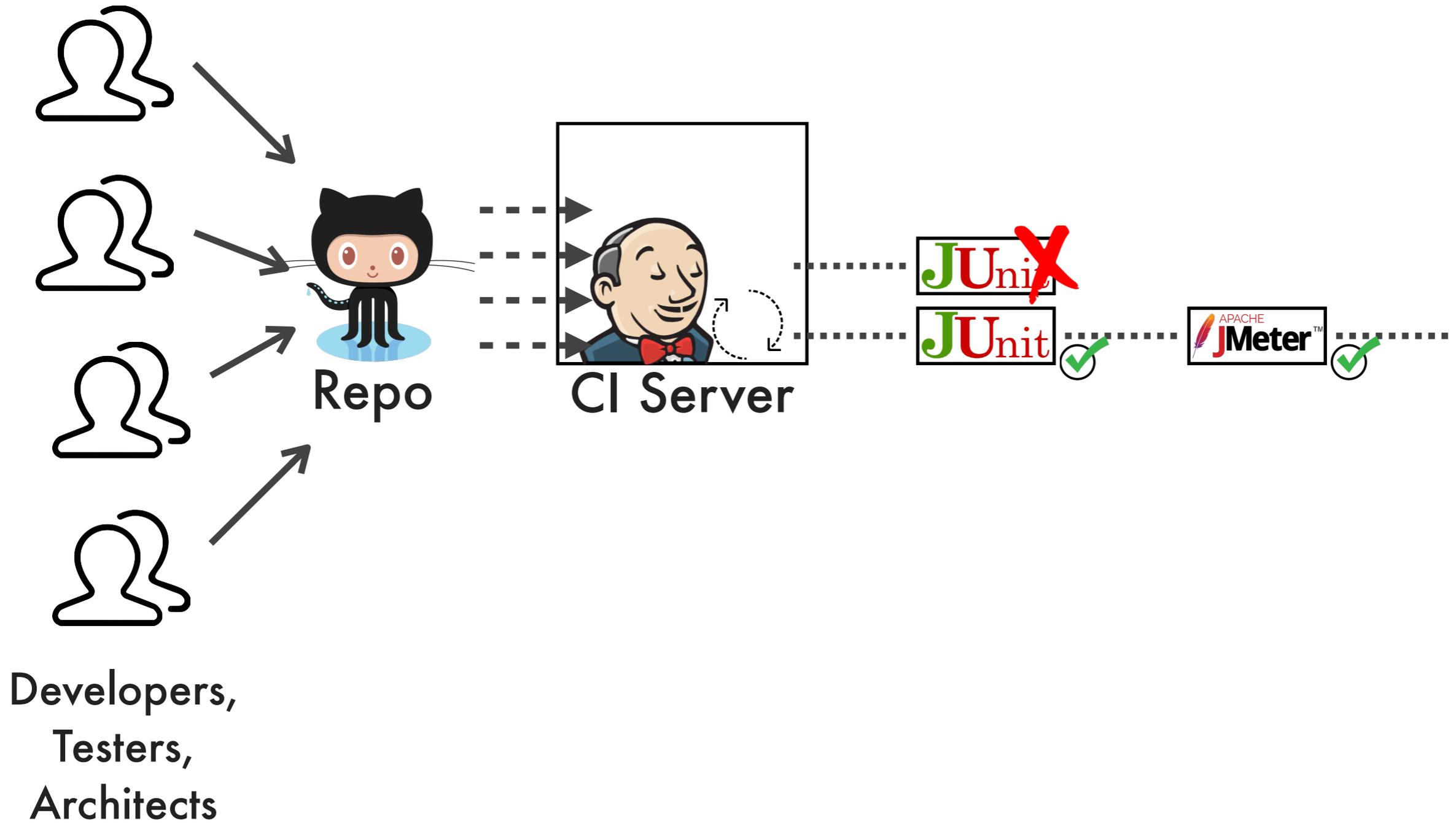
**Vincenzo Ferme
Cesare Pautasso**

Continuous Software Development Environments

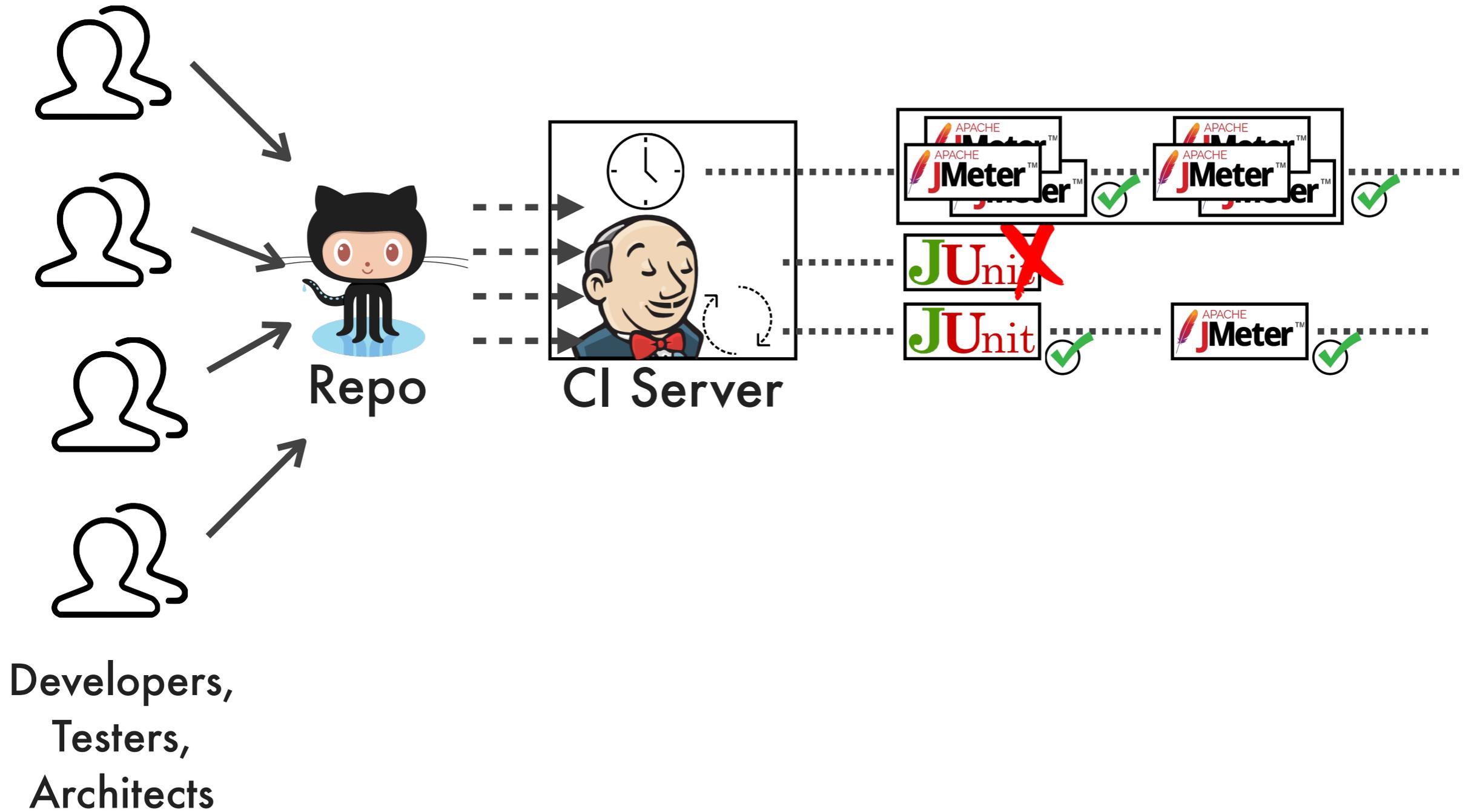


Developers,
Testers,
Architects

Continuous Software Development Environments

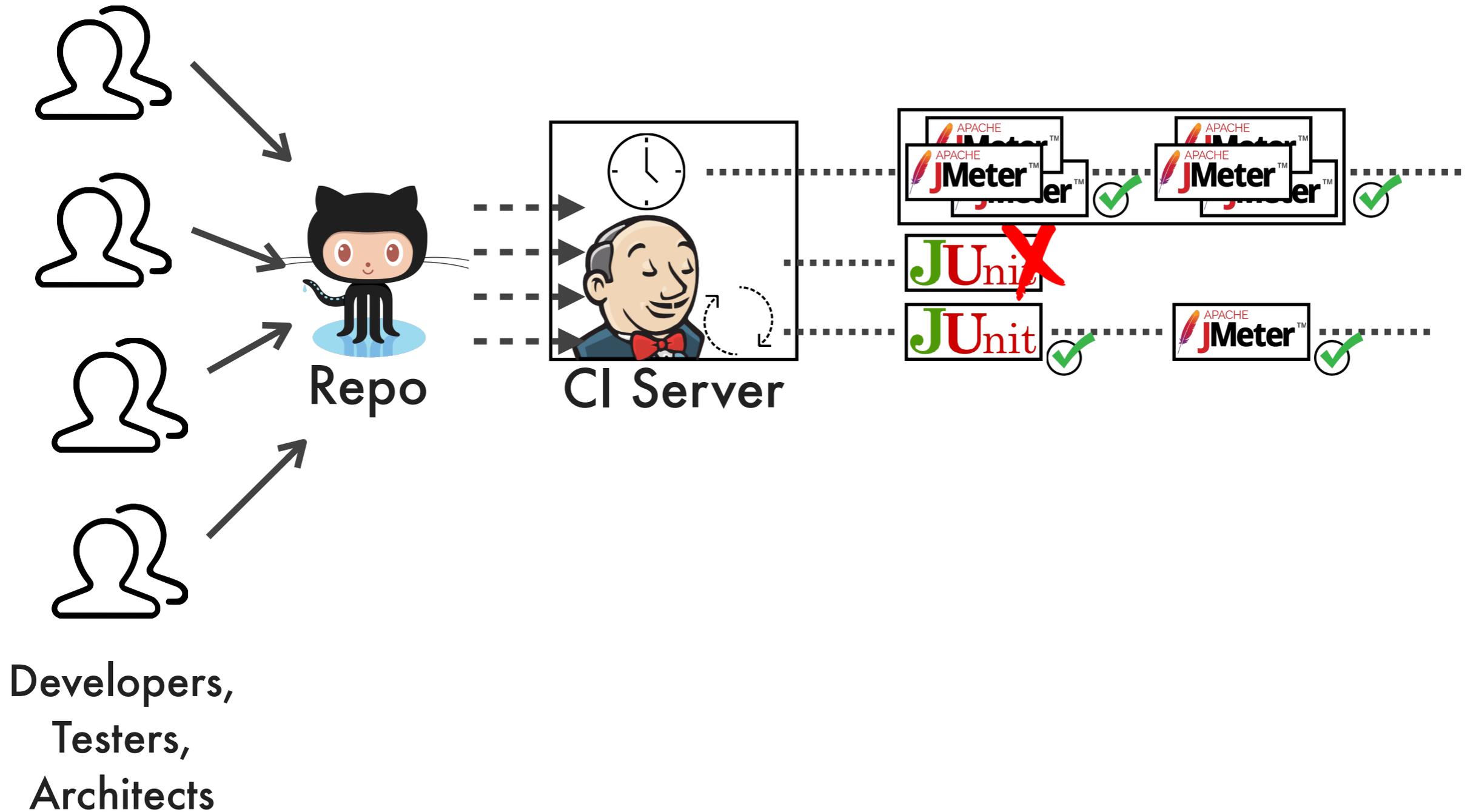


Continuous Software Development Environments



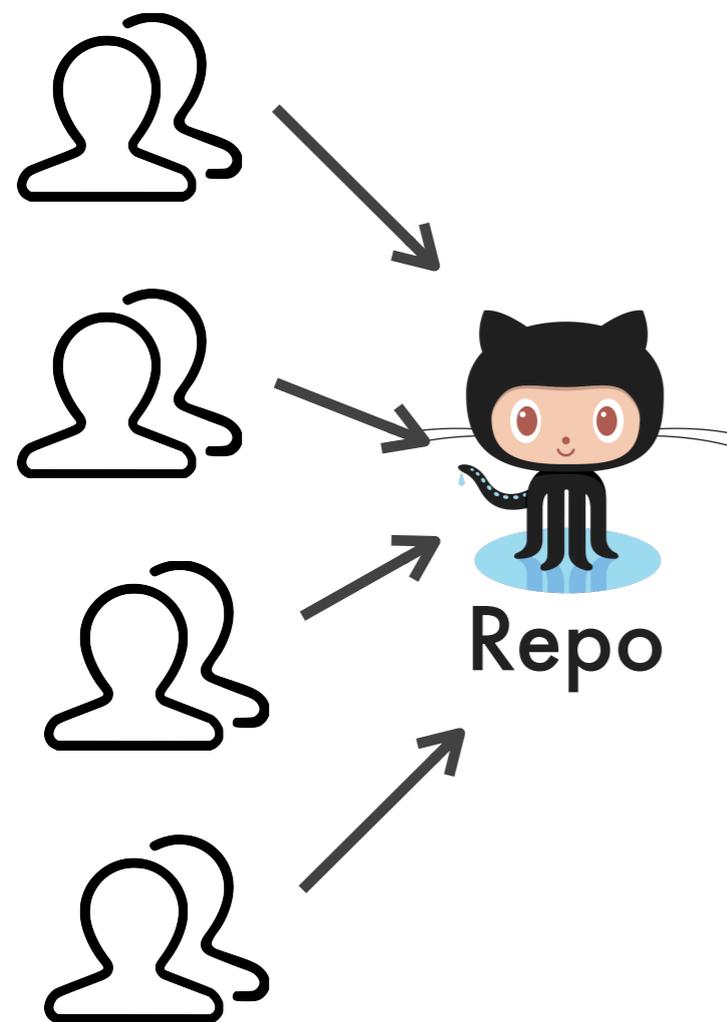
Continuous Software Development Environments

Continuous Changes

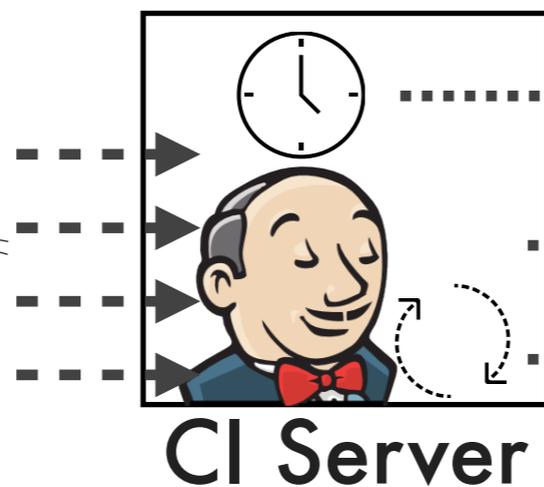


Continuous Software Development Environments

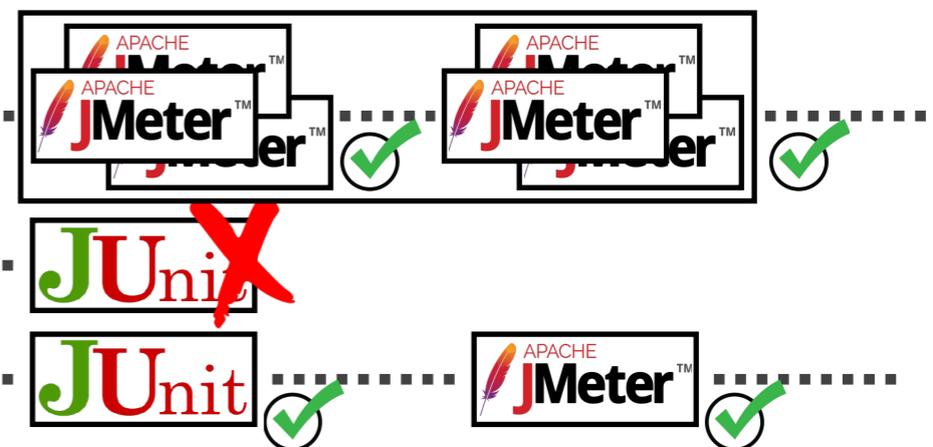
Continuous Changes



Developers,
Testers,
Architects

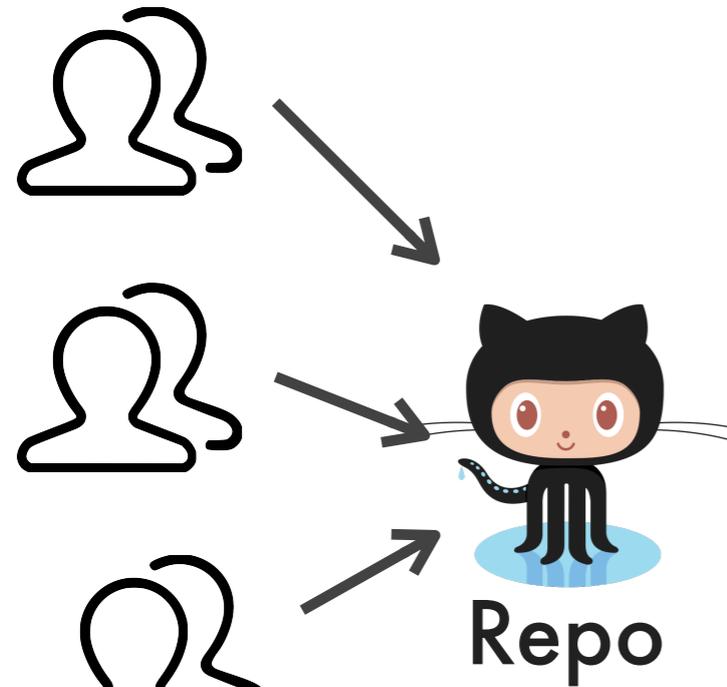


Continuous Test Execution

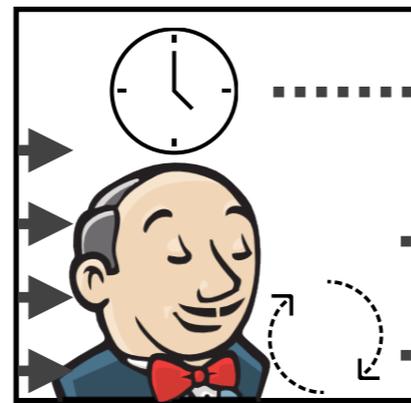


Continuous Software Development Environments

Continuous Changes



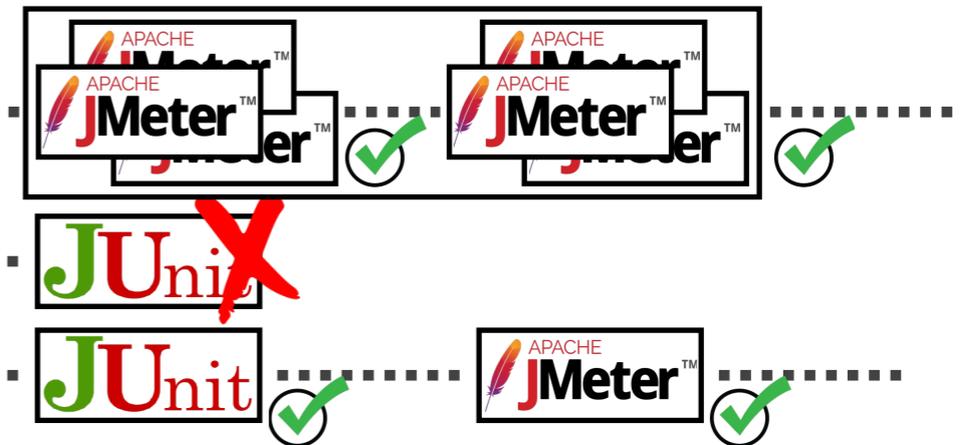
Developers,
Testers,
Architects



CI Server

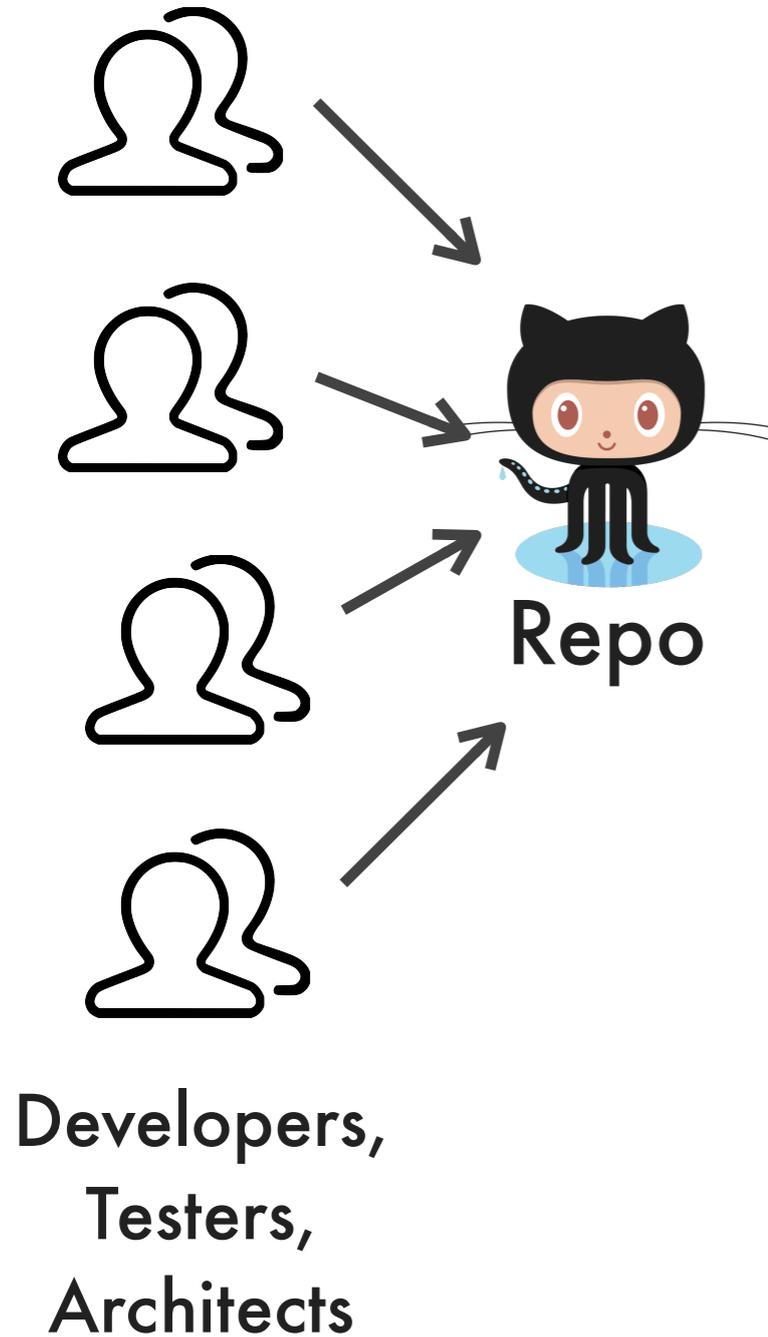
Load Test, Spike Test

Continuous Test Execution

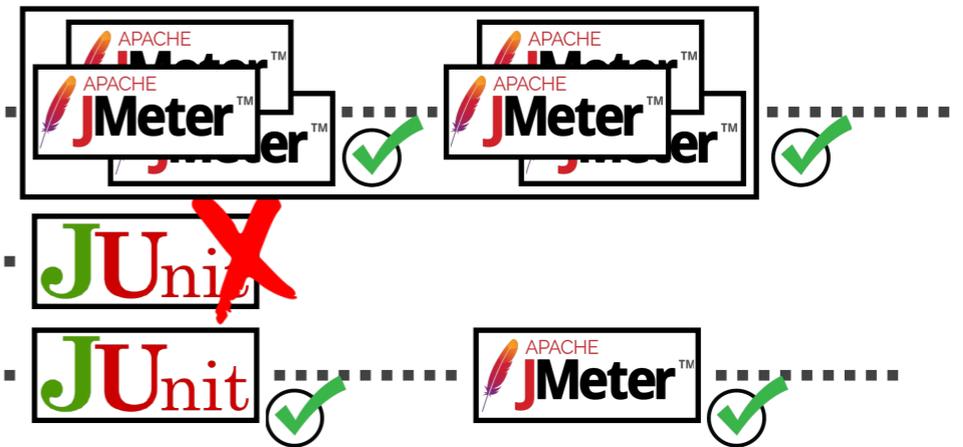


Continuous Software Development Environments

Continuous Changes



Continuous Test Execution



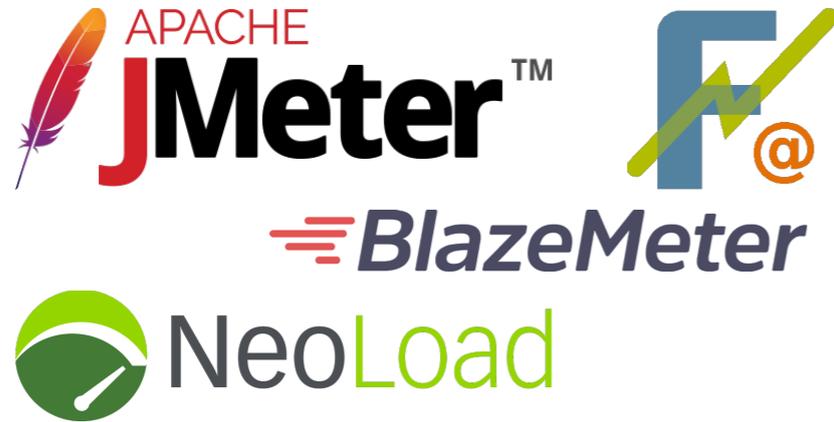
Load Test, Spike Test

Configuration Test

Capacity Test

State of Practice

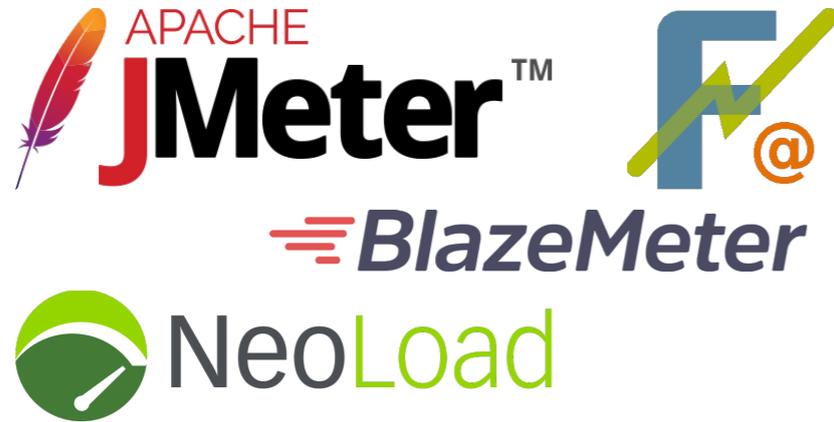
End-to-end Tests against REST APIs



Script/UI Based

State of Practice

End-to-end Tests against REST APIs



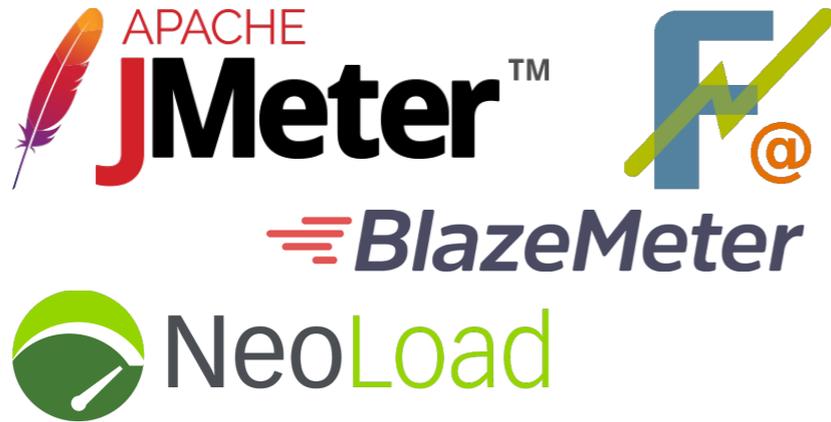
Script/UI Based



Integrated in CI Pipelines

State of Practice

End-to-end Tests against REST APIs



Script/UI Based

Integrated in CI Pipelines

CloudPerf

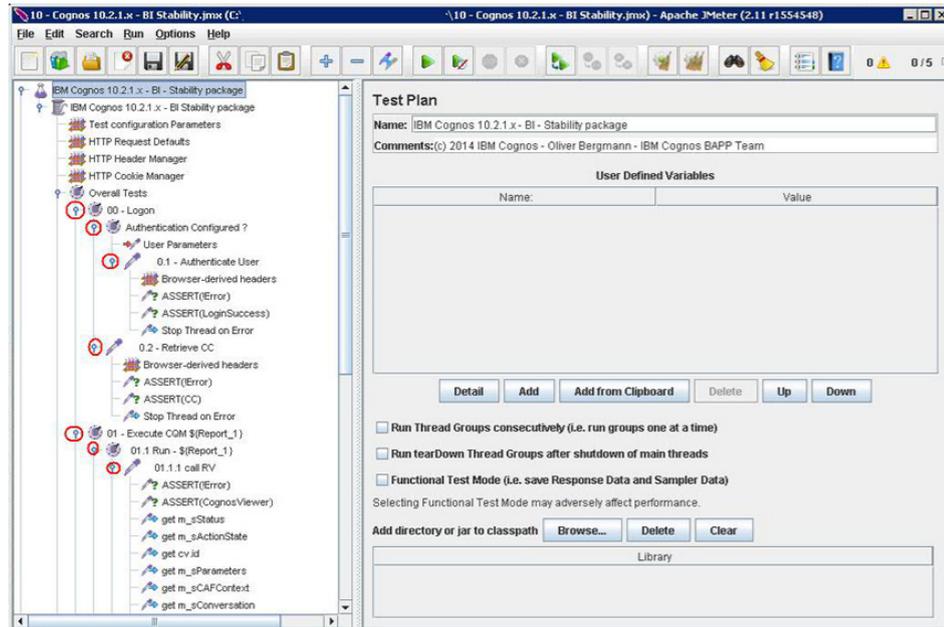
Google Vizier

DataMill

AutoPerf

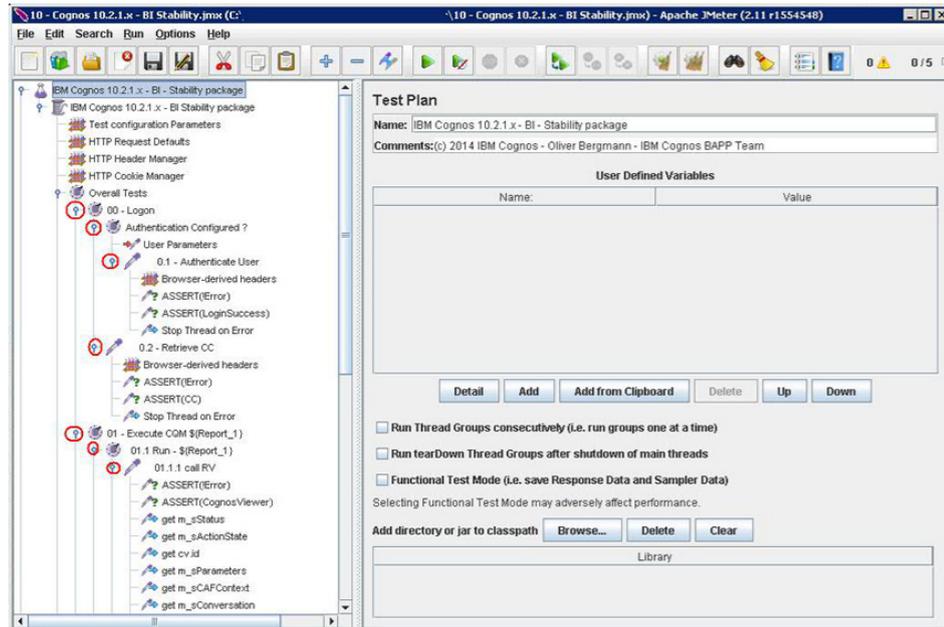
**Performance Tests
Execution Automation**

Define a Configuration Test



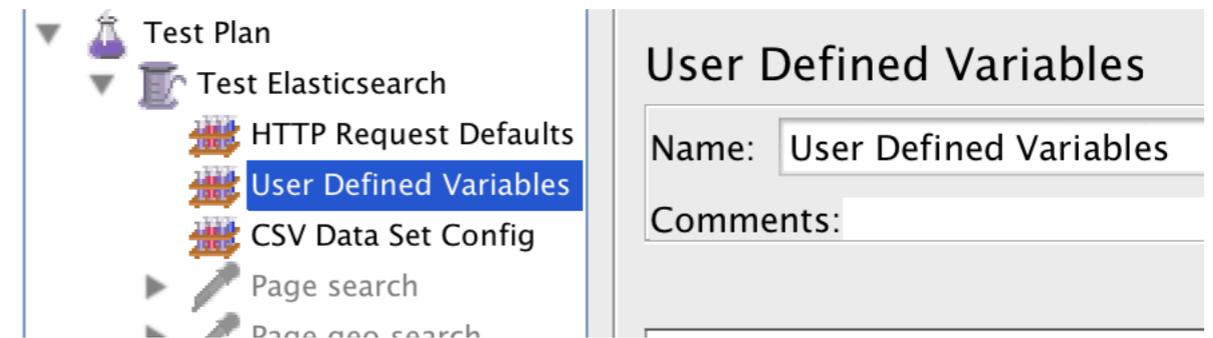
Test Definition

Define a Configuration Test



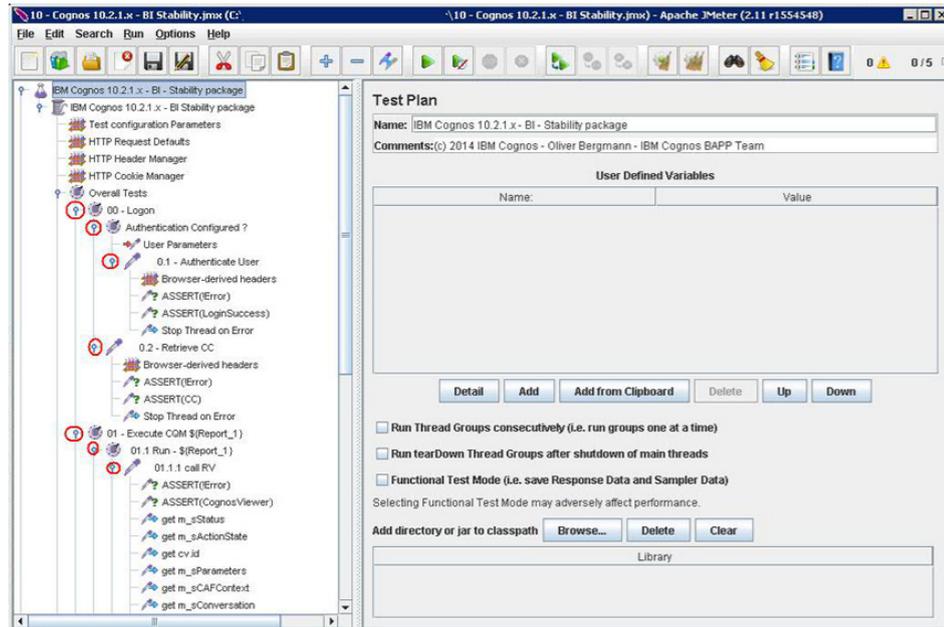
Test Definition

Simulated Users



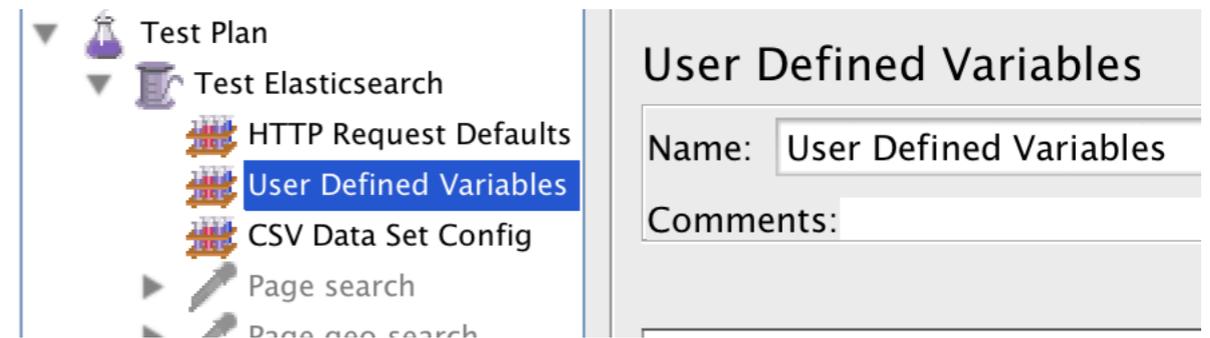
Configuration Parameters Change

Define a Configuration Test



Test Definition

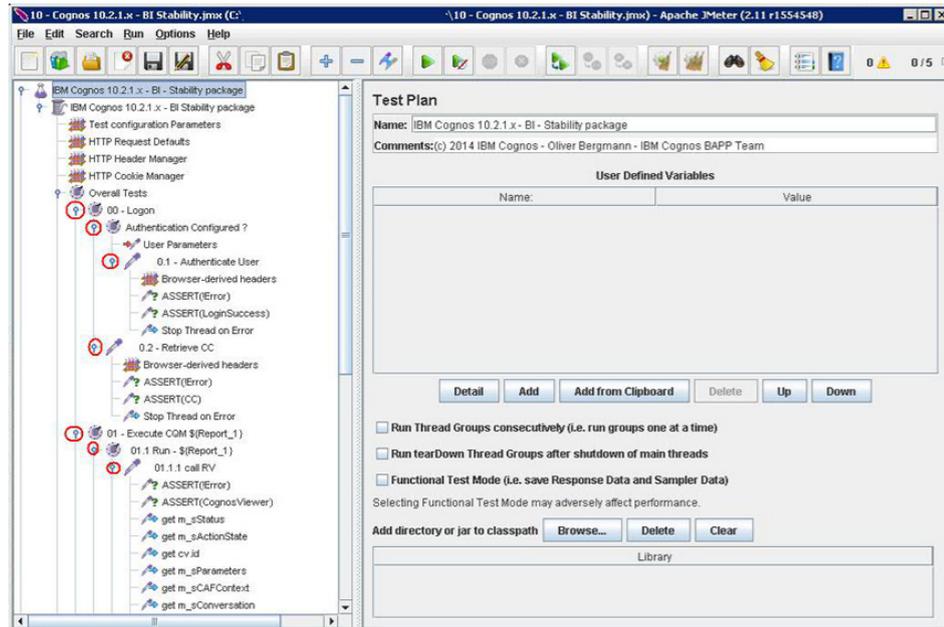
Simulated Users



System Under Test *3rd-party tools*

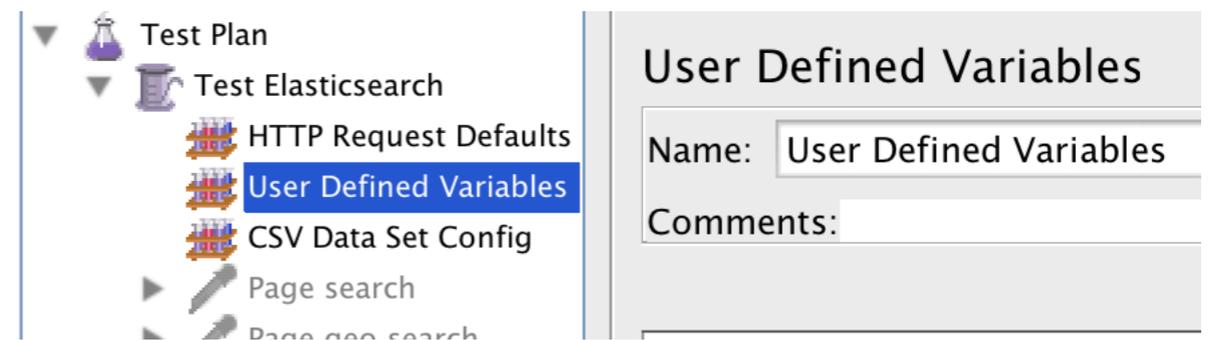
Configuration Parameters Change

Define a Configuration Test



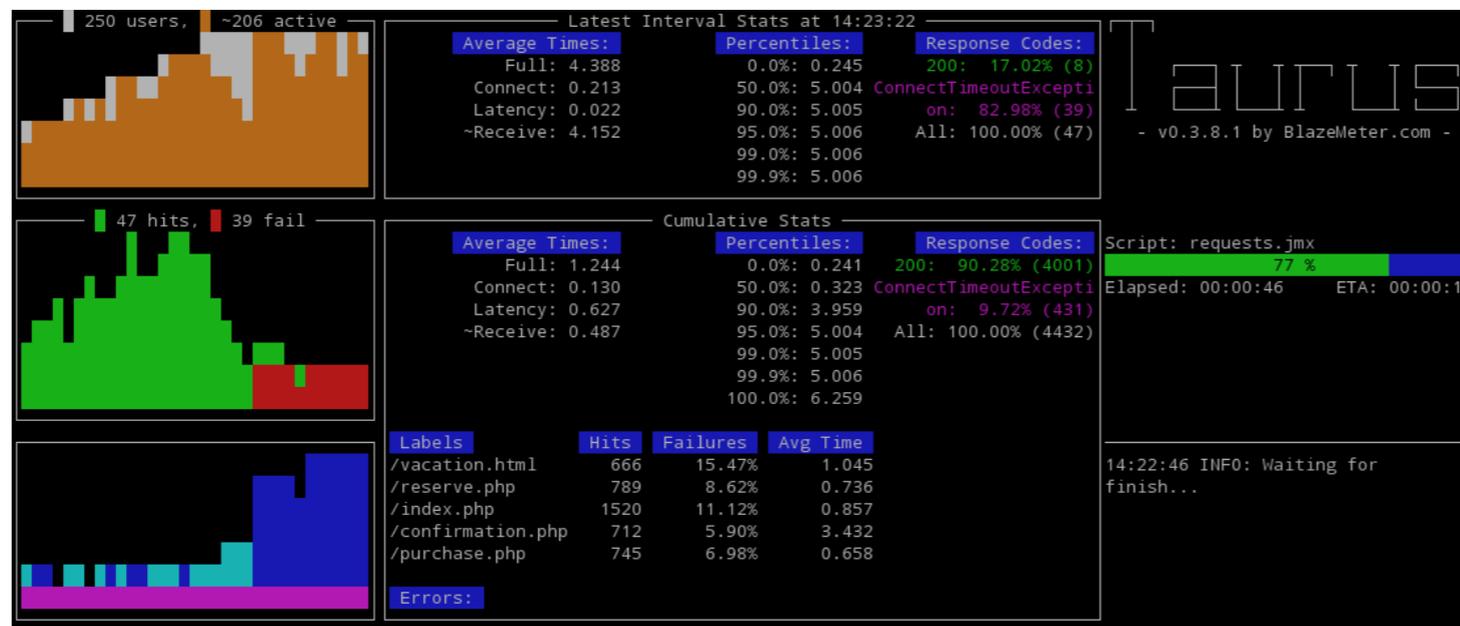
Test Definition

Simulated Users



System Under Test 3rd-party tools

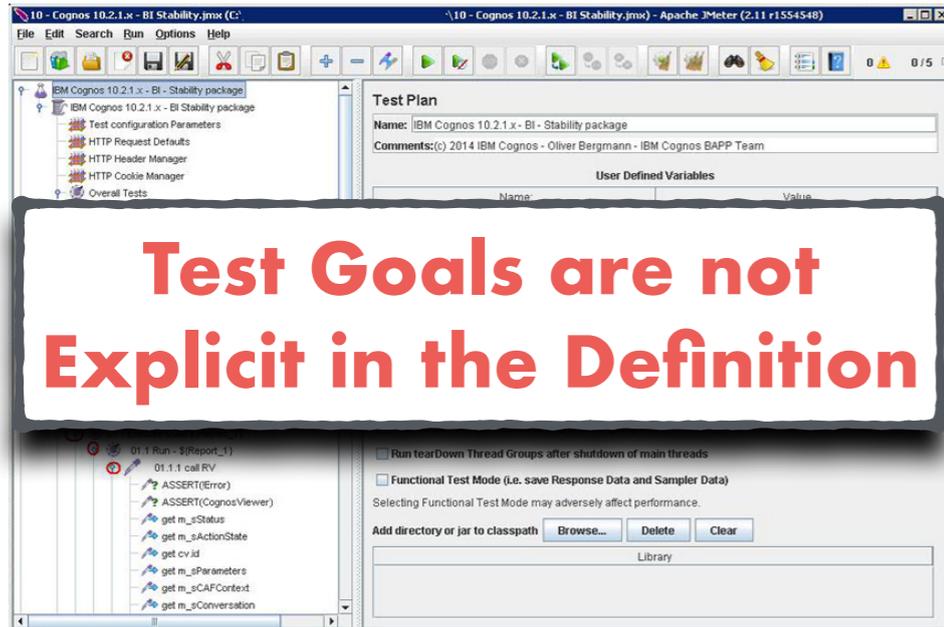
Configuration Parameters Change



Test Execution, Data Collection, Metrics Computation

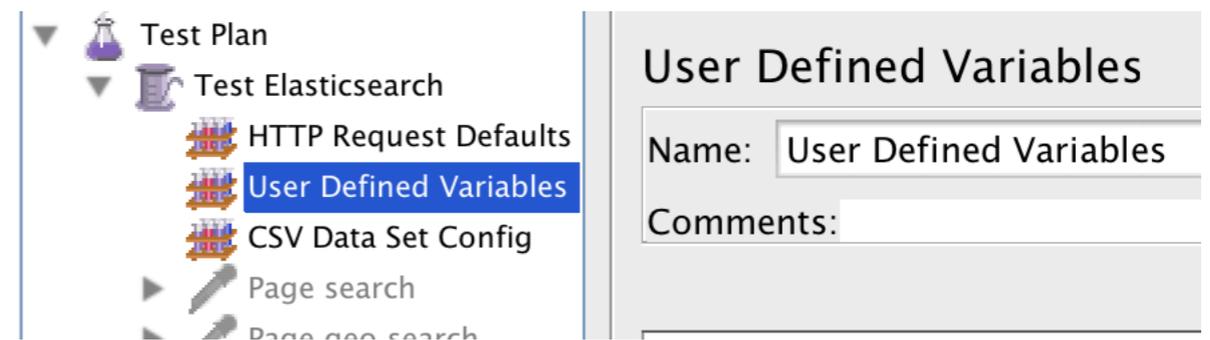


Define a Configuration Test



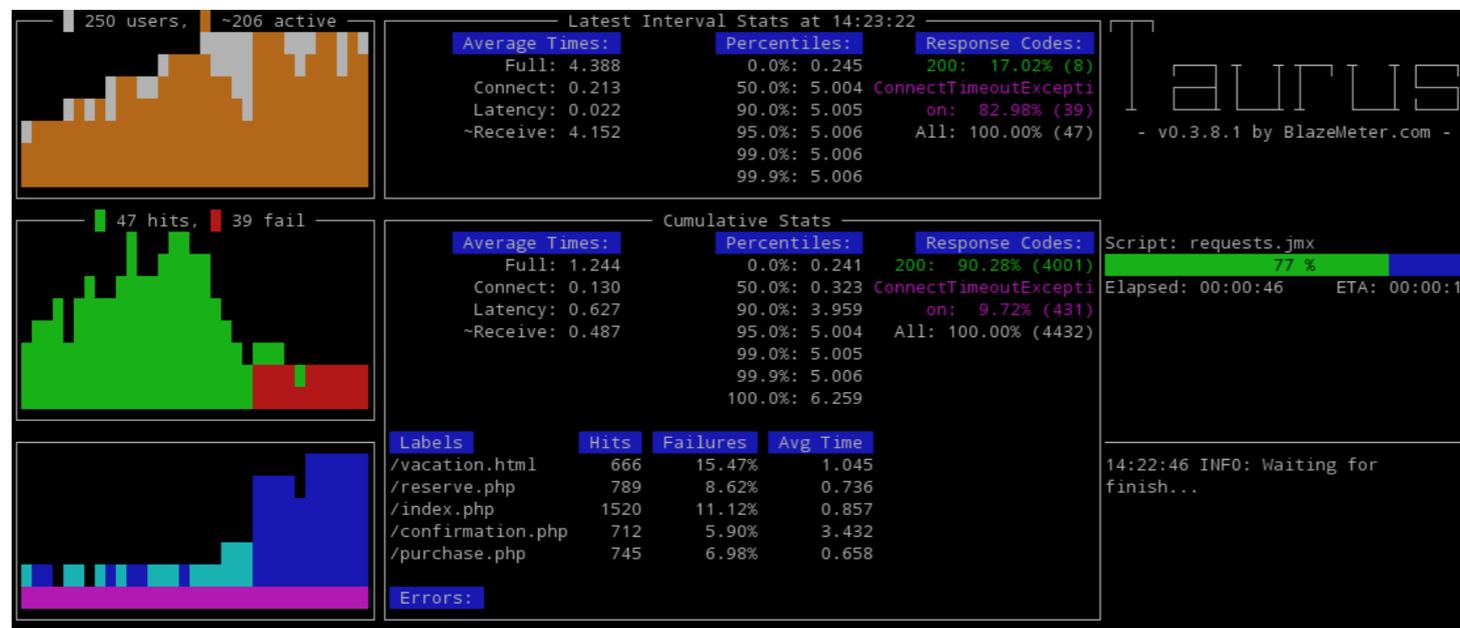
Test Definition

Simulated Users



System Under Test *3rd-party tools*

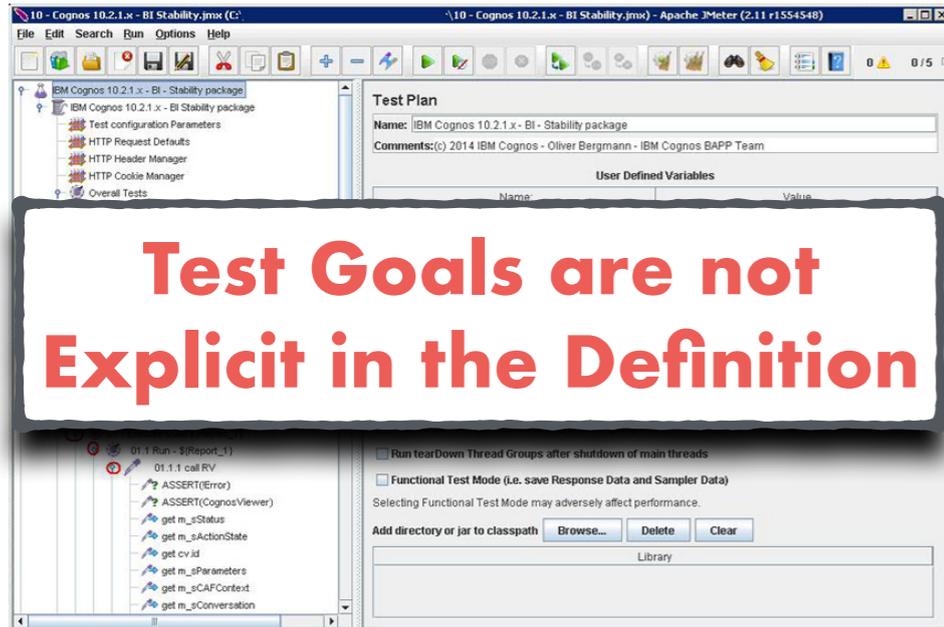
Configuration Parameters Change



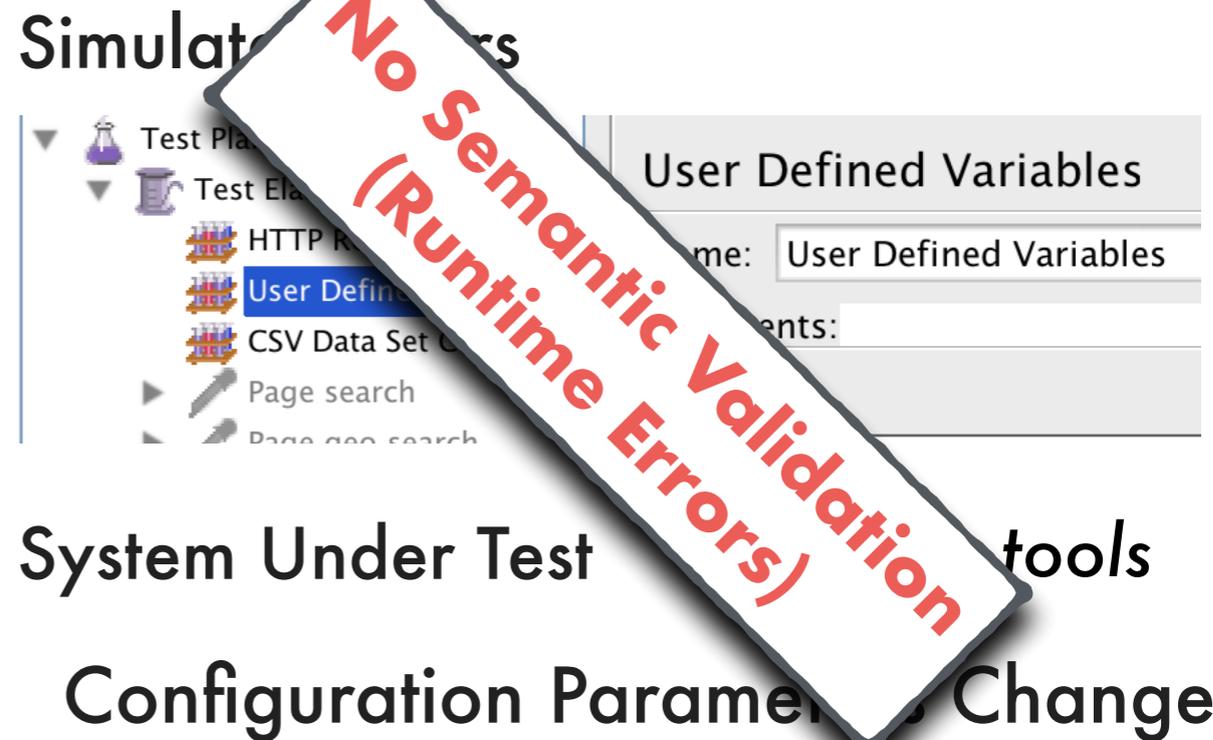
Test Execution, Data Collection, Metrics Computation



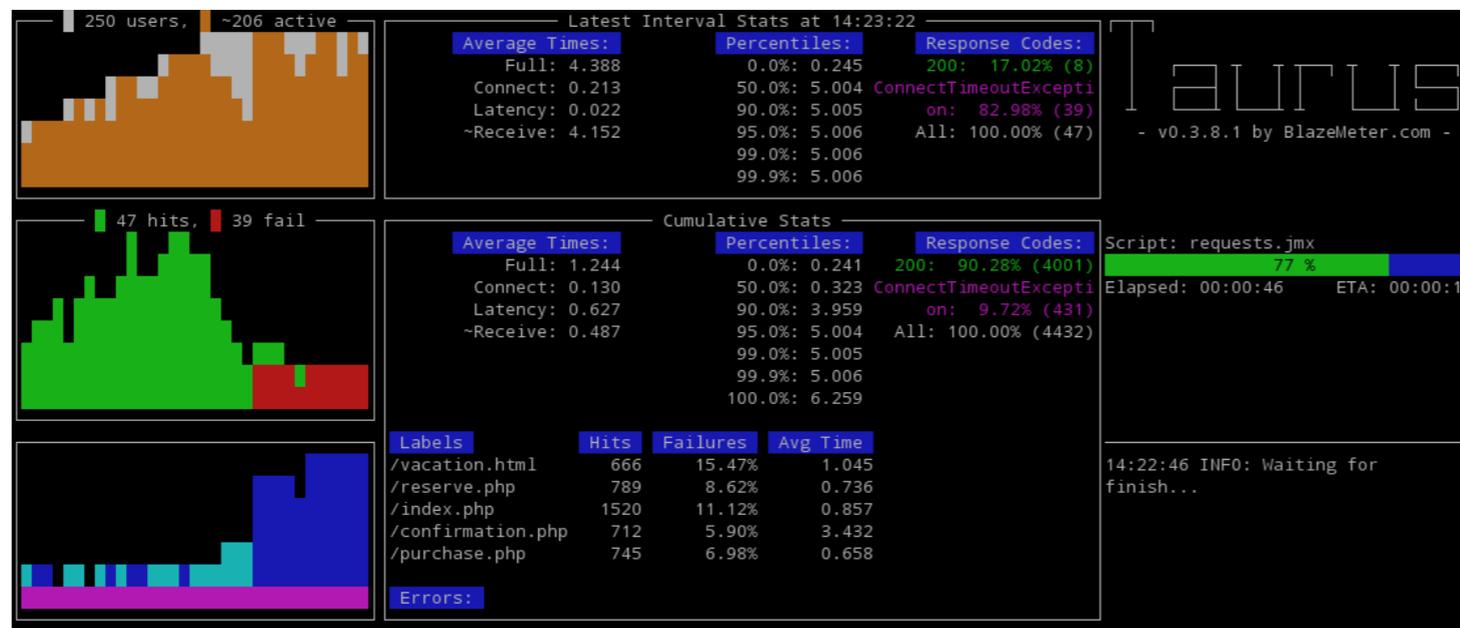
Define a Configuration Test



Test Definition



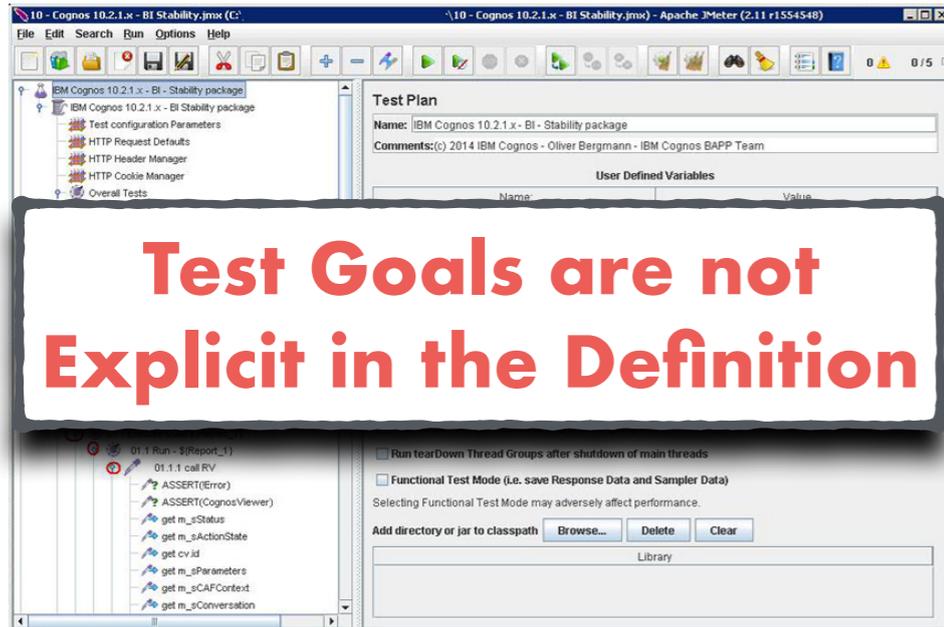
Configuration Parameters Change



Test Execution, Data Collection, Metrics Computation

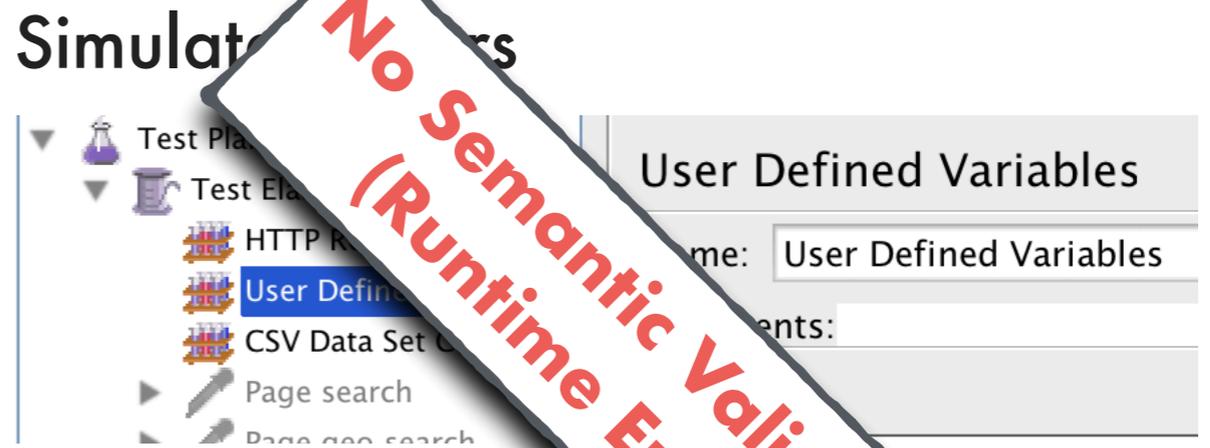


Define a Configuration Test



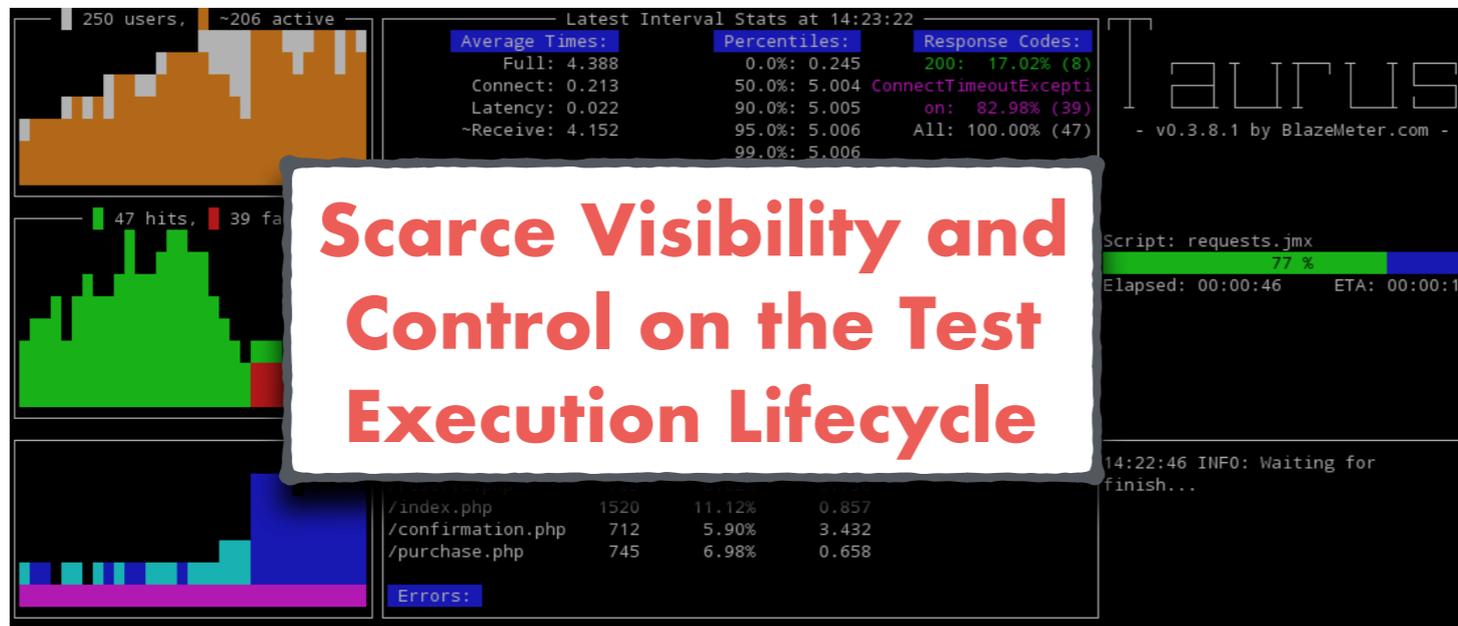
Test Goals are not Explicit in the Definition

Test Definition



System Under Test Configuration Parameters Change tools

Configuration Parameters Change



Scarce Visibility and Control on the Test Execution Lifecycle



Test Execution, Data Collection, Metrics Computation

Declarative Performance Engineering (DPE)

“Enabling the performance analyst to **declaratively specify what** performance-relevant questions need to be answered **without being concerned about how** they should be answered. ”

[Walter et al.]

[Walter et al.]

Jürgen Walter, André van Hoorn, Heiko Koziolk, Dusan Okanovic, and Samuel Kounev. Asking "What?", Automating the "How"? - The Vision of Declarative Performance Engineering. In Proc. of ICPE 2016. 91–94. 6

Declarative Performance Engineering (DPE)

Developers,
Testers,
Architects,

Performance Analyst ...

“ Enabling the ~~performance analyst~~ to **declaratively specify what** performance-relevant questions need to be answered **without being concerned about how** they should be answered. ”

[Walter et al.]

[Walter et al.]

Jürgen Walter, André van Hoorn, Heiko Koziolk, Dusan Okanovic, and Samuel Kounev. Asking "What?", Automating the "How"? - The Vision of Declarative Performance Engineering. In Proc. of ICPE 2016. 91–94. 6

DPE: Approaches for Performance Testing

“Enabling the performance analyst to **declaratively specify what** performance-relevant questions need to be answered **without being concerned about how** they should be answered.”

[Walter et al.]



[Omar et al.]



[Westermann]



[Scheuner et al.]

[Omar et al.]

Towards an automated approach to use expert systems in the performance testing of distributed systems.

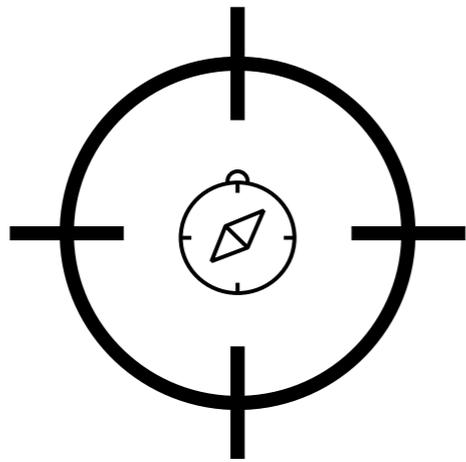
[Westermann]

Deriving Goal-oriented Performance Models by Systematic Experimentation

[Scheuner et al.]

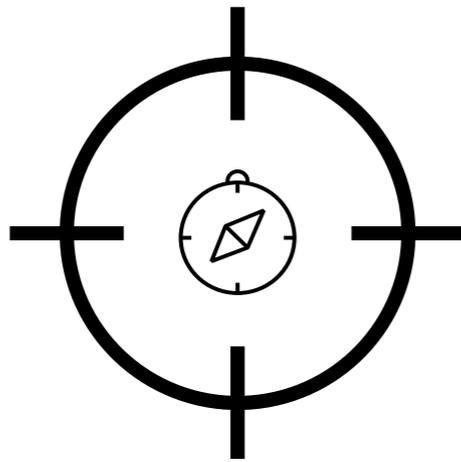
Cloud Work Bench - Infrastructure-as-Code Based Cloud Benchmarking.

Our Proposal: BenchFlow

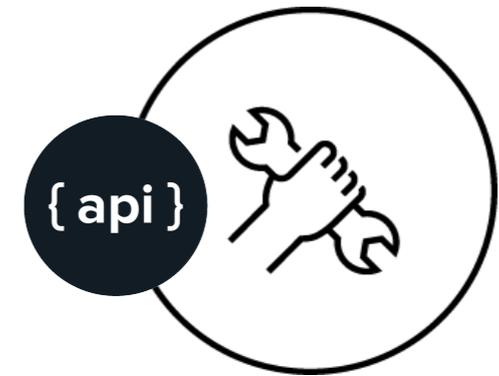


Declarative
Performance Testing DSL

Our Proposal: BenchFlow

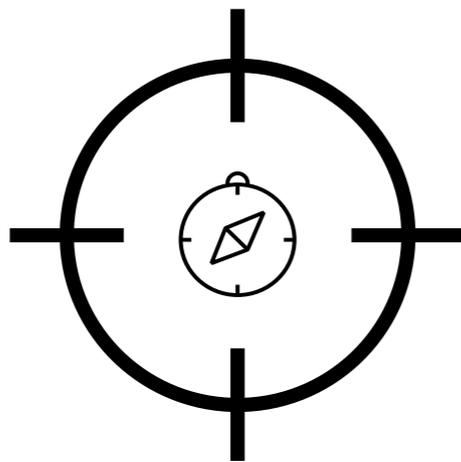


Declarative
Performance Testing DSL

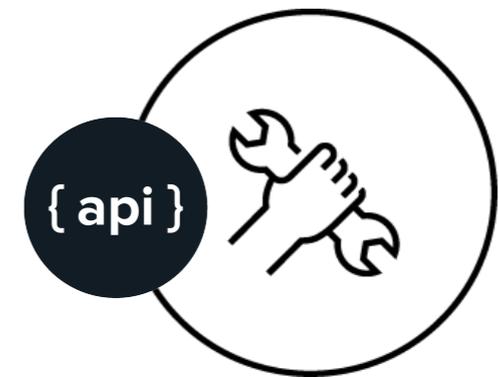


Model-driven
Framework

Our Proposal: BenchFlow



Declarative
Performance Testing DSL

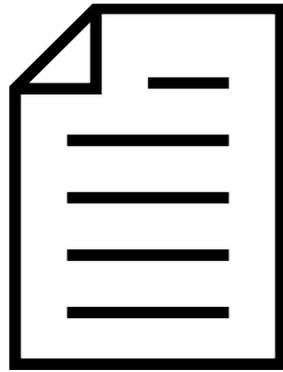


Model-driven
Framework



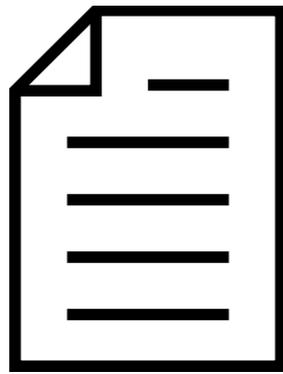
<https://github.com/benchflow>

BenchFlow: Elements of a Test



Test Definition

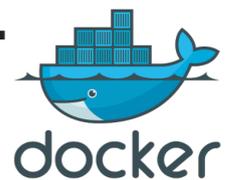
BenchFlow: Elements of a Test



Test Definition



SUT Configuration
and Deployment



Declarative Test Definition

configuration:

goal:

load_function:

quality_gates:

termination_criteria:

data_collection:

workload:

sut:

YAML

Goal Definition

(Configuration Test)

configuration:

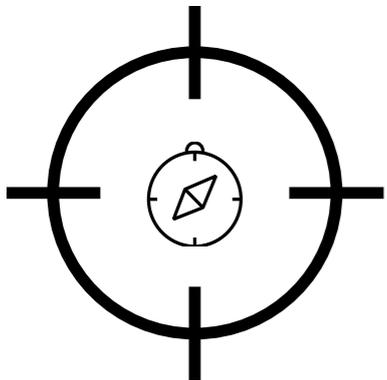
goal:

type: **configuration**

observe:

exploration:

...



Observed Metrics

...

goal:

type: **configuration**

observe:

workload:

API_1:

- **90thp_response_time**

services:

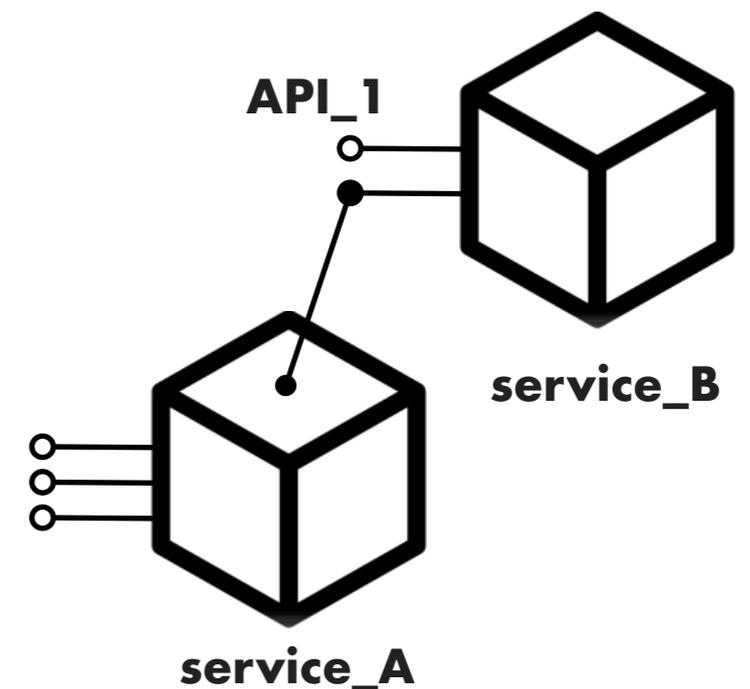
service_A:

- **cpu_avg**

service_B:

- **ram_avg**

...



Exploration Space

...

goal:

type: **configuration**

exploration:

exploration_space:

service_A:

resources:

- memory:

range: **1GB... 5GB**

step: **+1GB**

- cpus:

range: **0.1...0.5**

- replicas: **1...4**

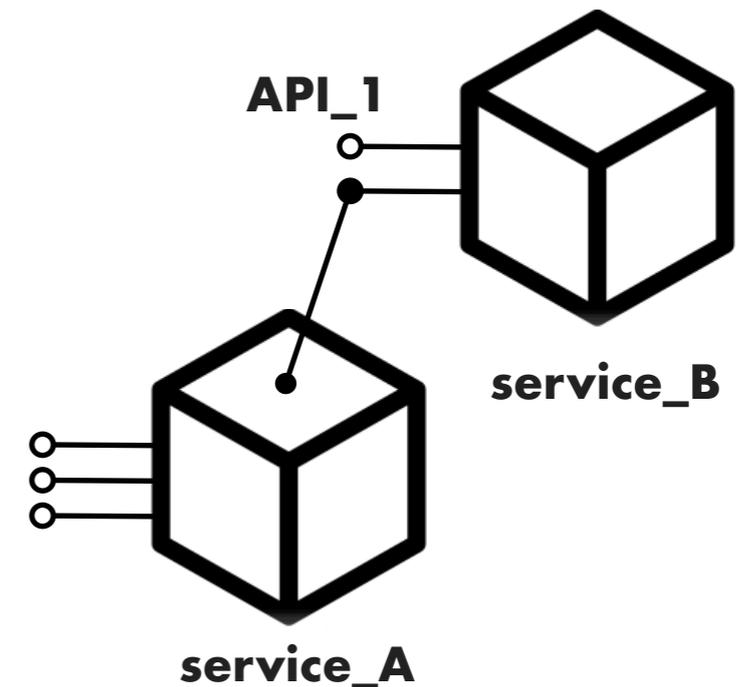
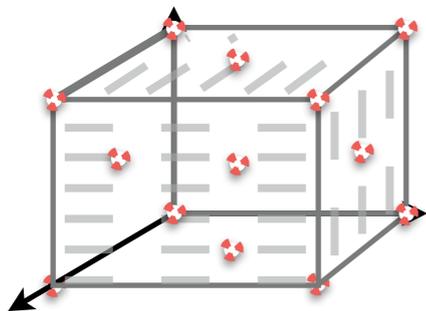
environment:

- **SIZE_OF_THREADPOOL:**

range: **5...100**

step: **+5**

...



Load Function

...

goal:

load_function:

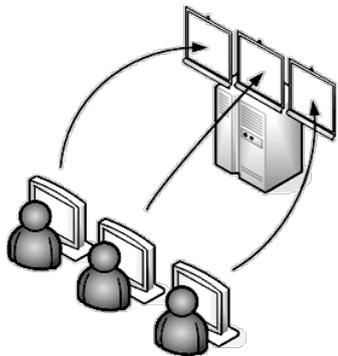
users: **1000**

ramp_up: **30s**

steady_state: **10m**

ramp_down: **30s**

...



Quality Gates

...

goal:

quality_gates:

workload:

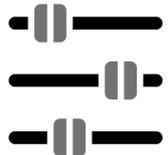
API_1:

90thp_response_time: $\leq 250\text{ms}$

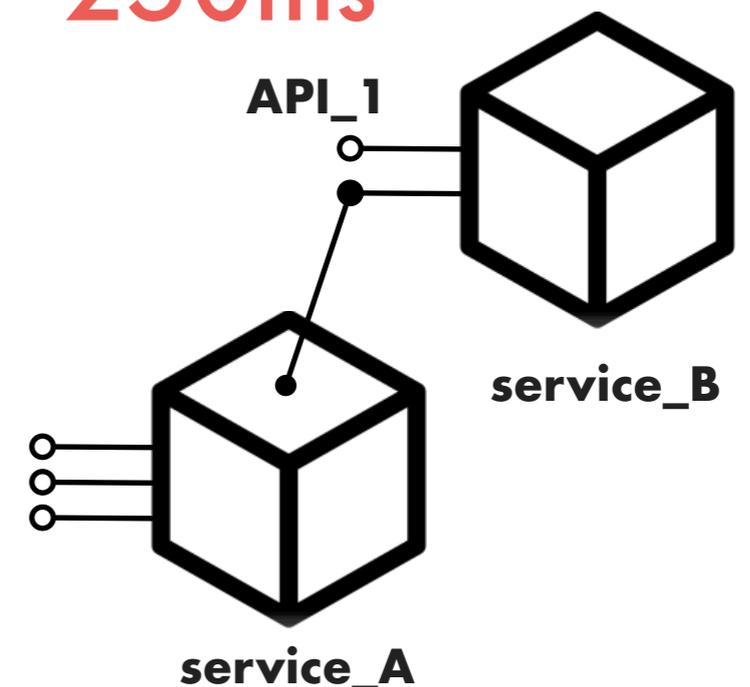
services:

service_A:

cpu_avg: $\leq 70\%$



...



Termination Criteria

...

goal:

termination_criteria:

test:

max_time: **3h**

failed_experiments: **30%**

...



Data Collection

...

data_collection:

client_side:

faban:

interval: **1s**

server_side:

service_A:

resource

mysql:

environment:

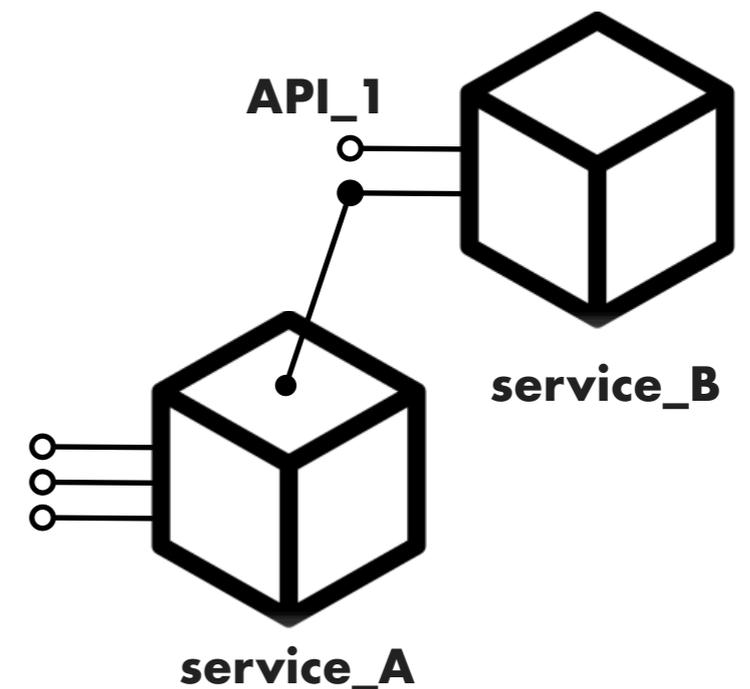
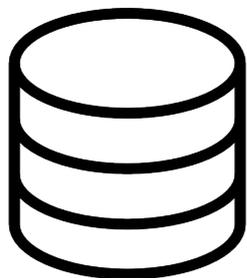
SETTING_A : **example_setting**

service_B:

- **logs**

- **resource**

...



Workload

...

workload:

<name_of_the_workload>:

driver_type: "http"

popularity: 100%

inter_operation_timings: "uniform"

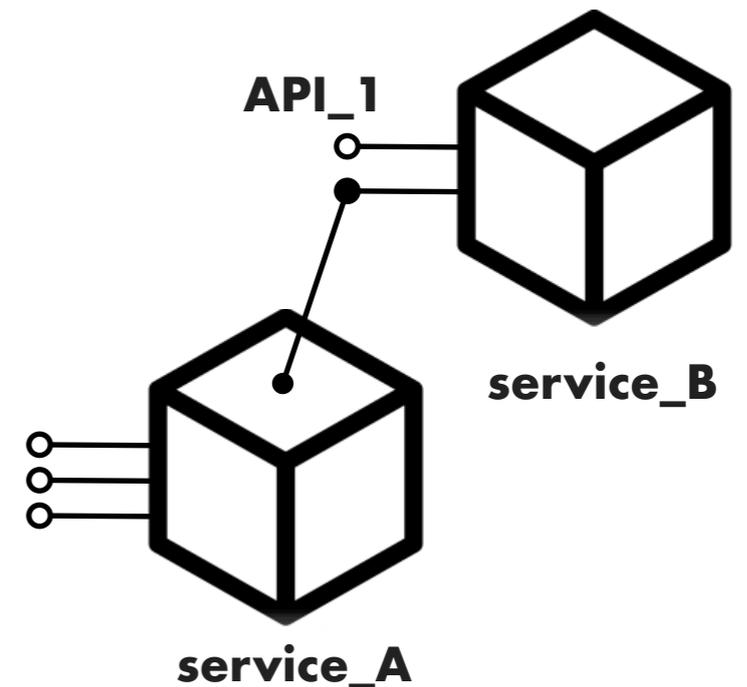
operations:

- ...

mix:

- ...

...



System Under Test Deployment

...

sut:

configuration:

target_service:

name: **service_A**

endpoint: /

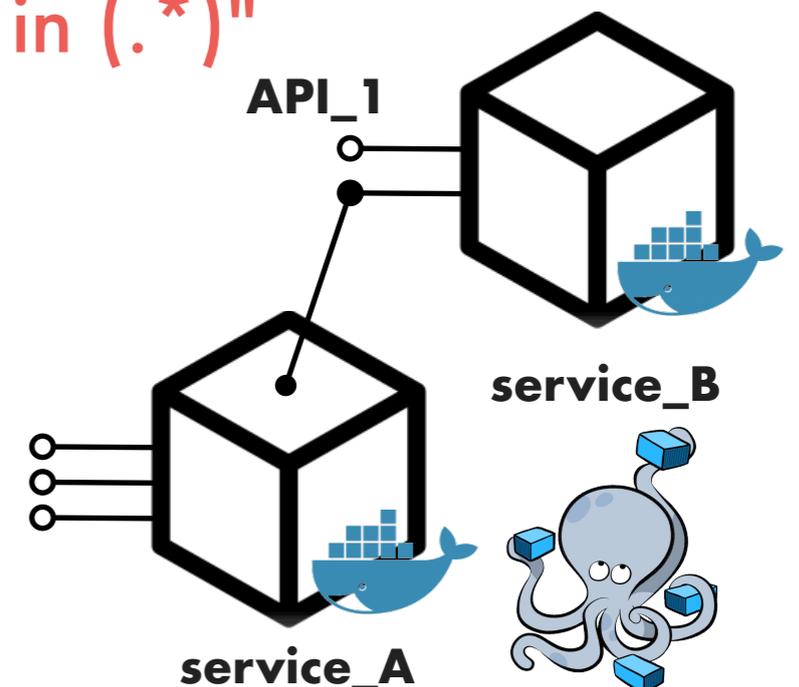
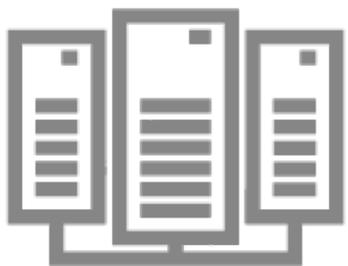
sut_ready_log_check: **"Started in (.*)"**

deployment:

service_A: **server_1**

service_B: **server_2**

...



System Under Test Deployment

...

sut:

configuration:

target_service:

name: **service_A**

endpoint: /

sut_ready_log_check: **"Started in (.*)"**

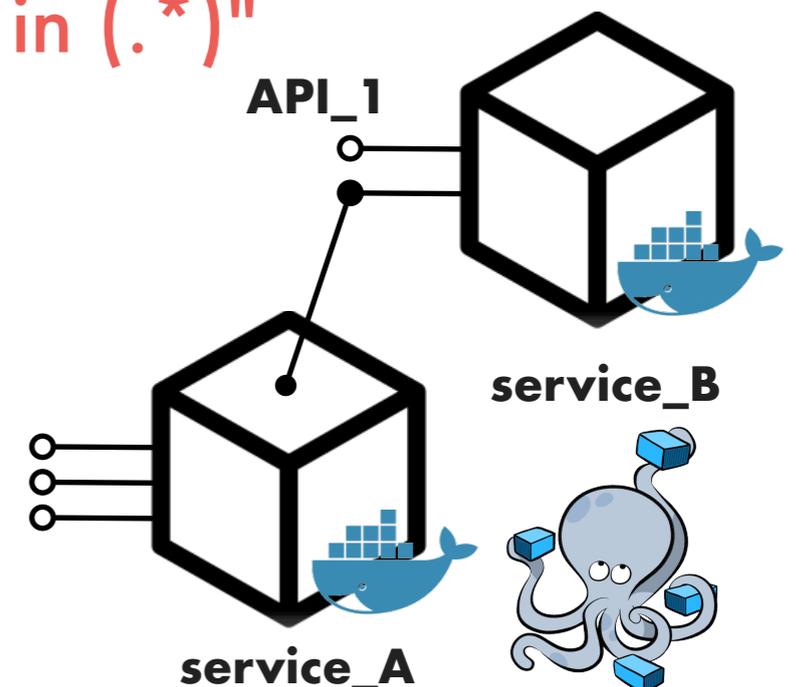
deployment:

service_A: **server_1**

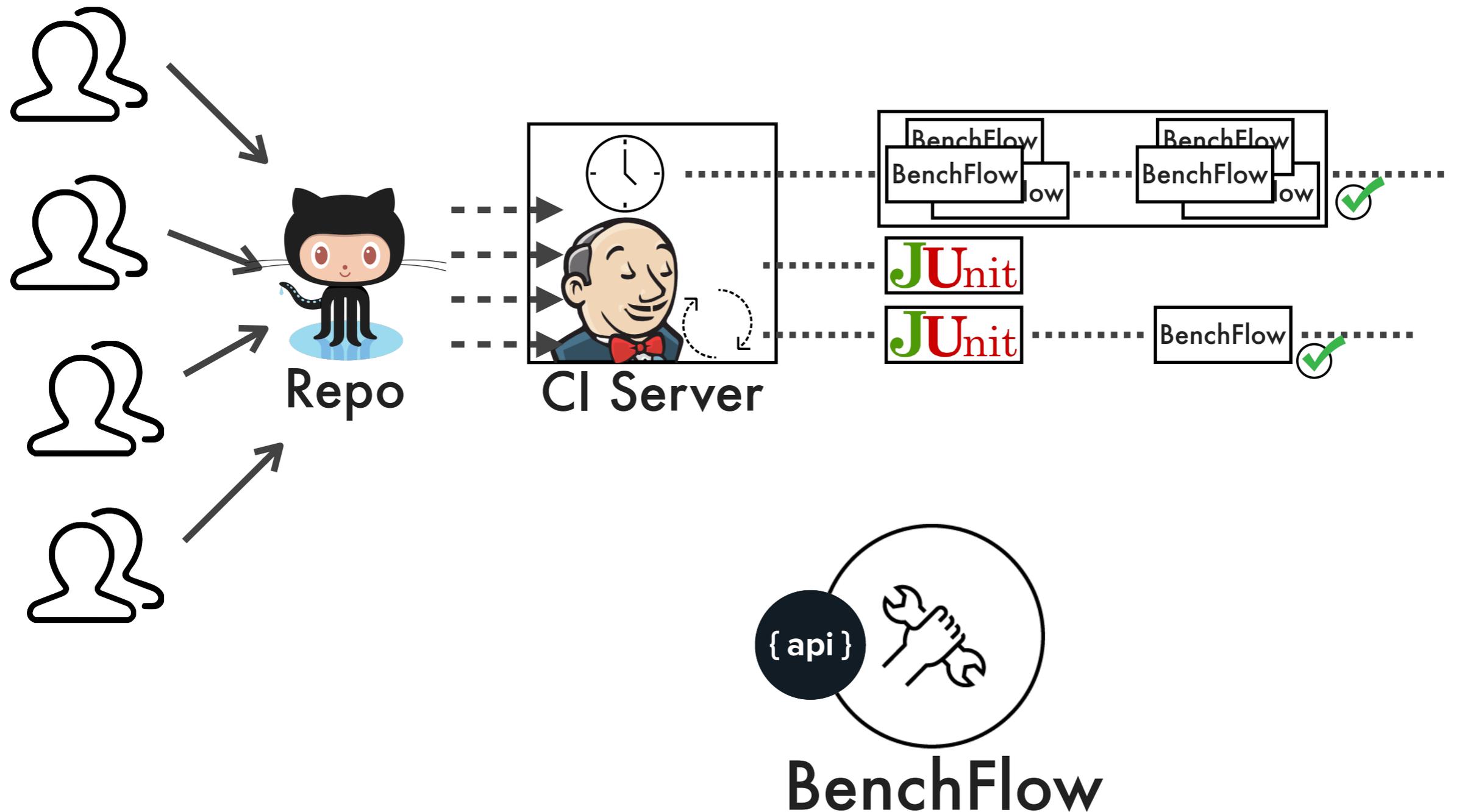
service_B: **server_2**

...

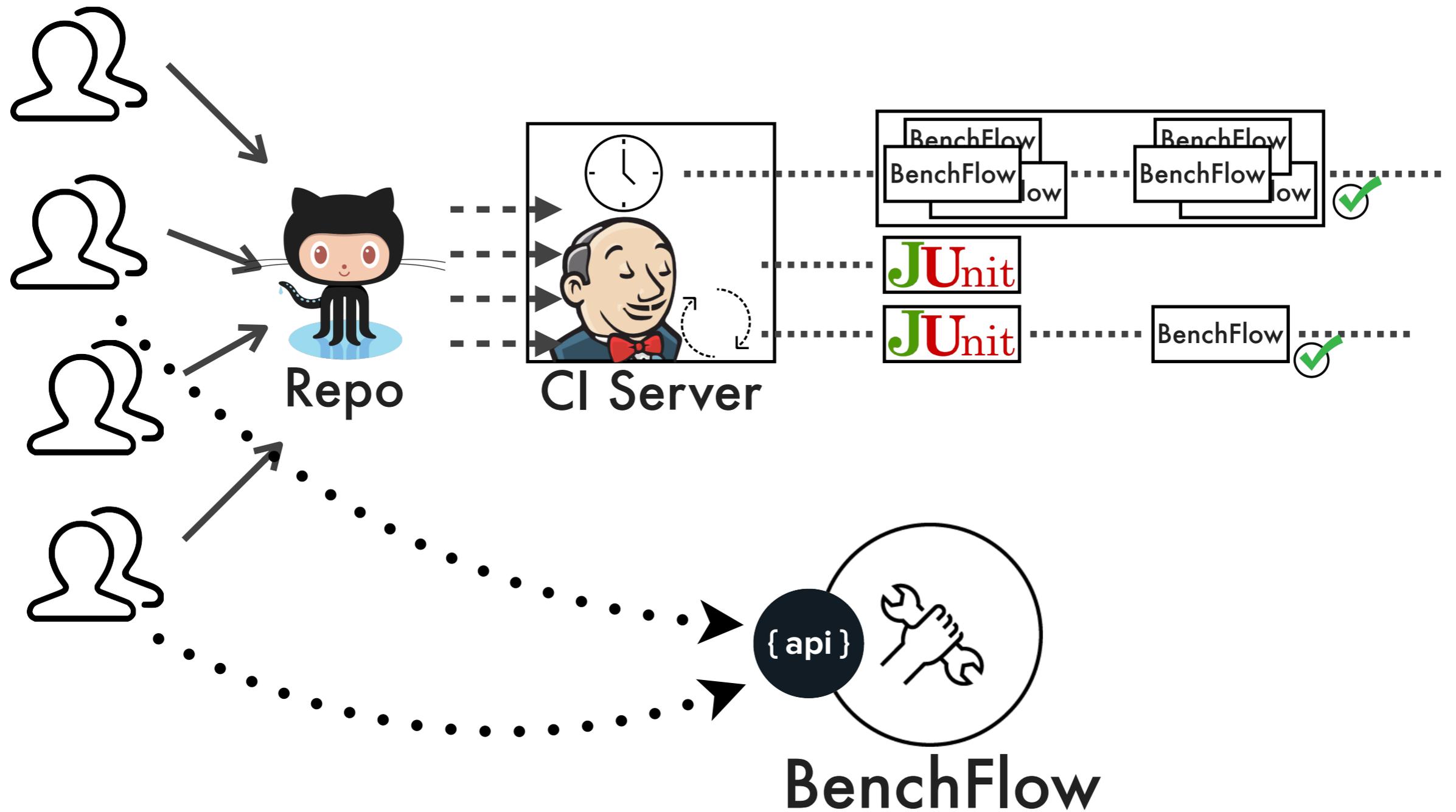
 <https://bit.ly/2EBNrDu>



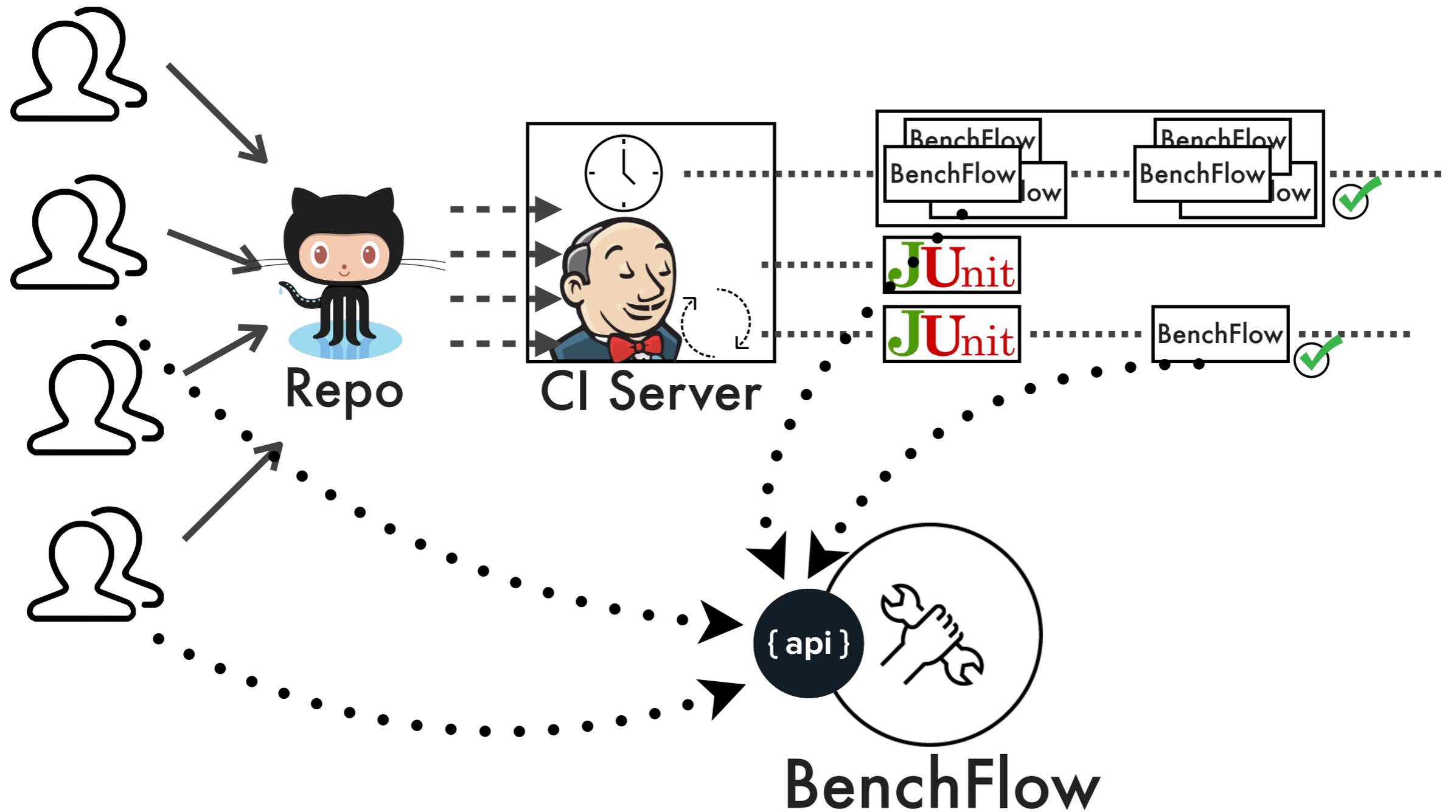
Model-driven Framework



Model-driven Framework



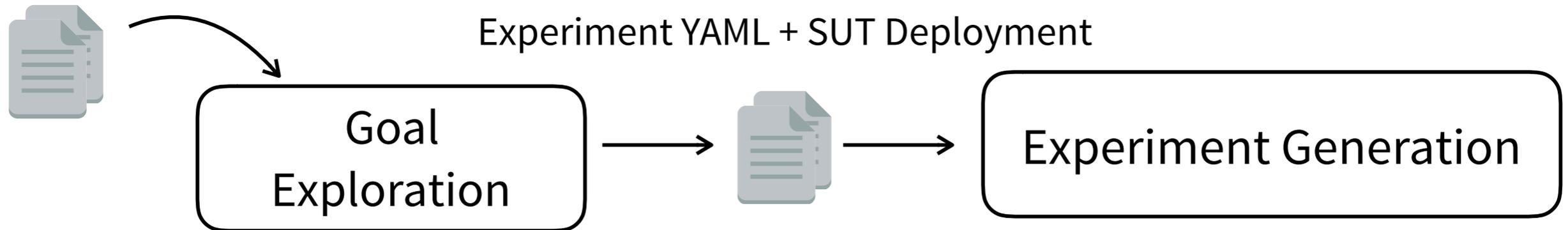
Model-driven Framework



Model-driven Framework

Test YAML + SUT Deployment

Experiment YAML + SUT Deployment

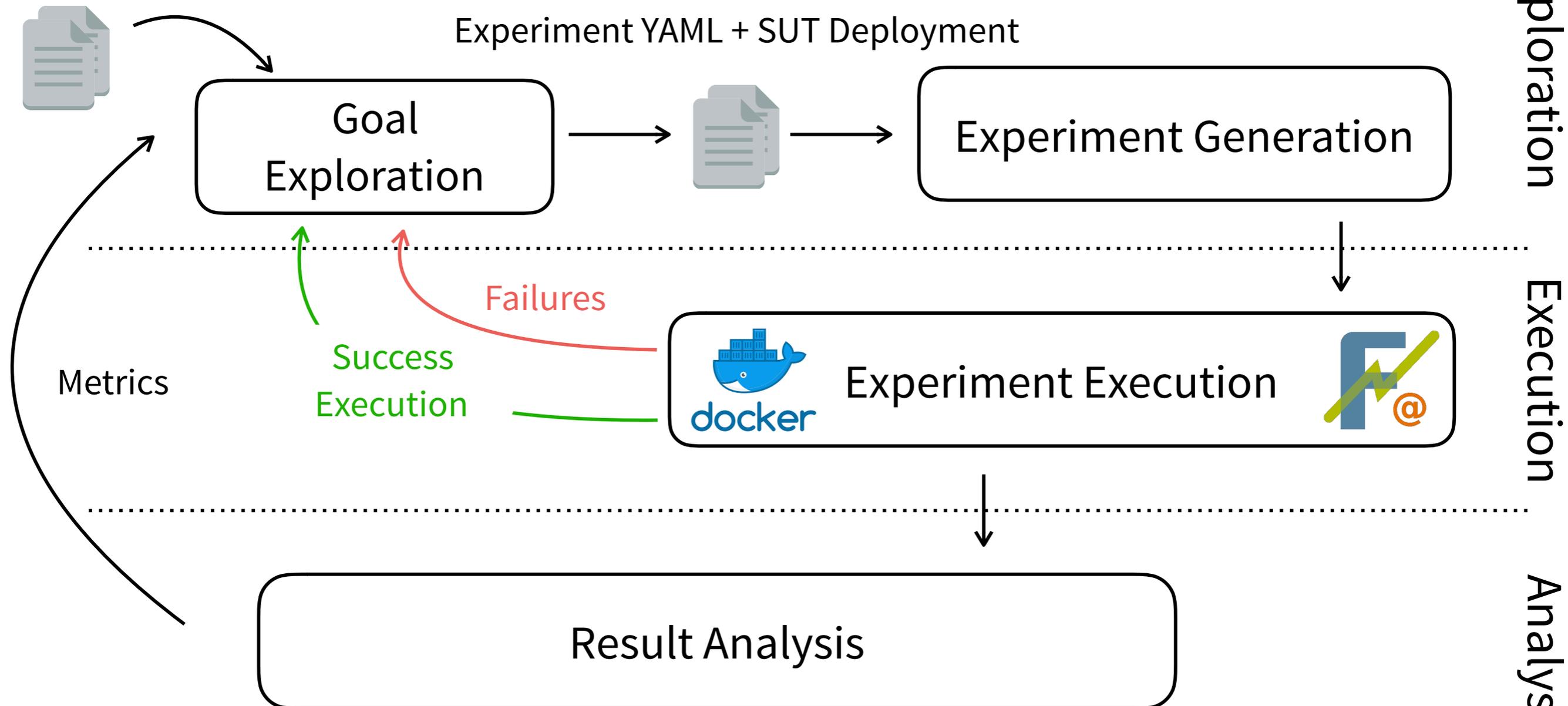


Exploration

Model-driven Framework

Test YAML + SUT Deployment

Experiment YAML + SUT Deployment



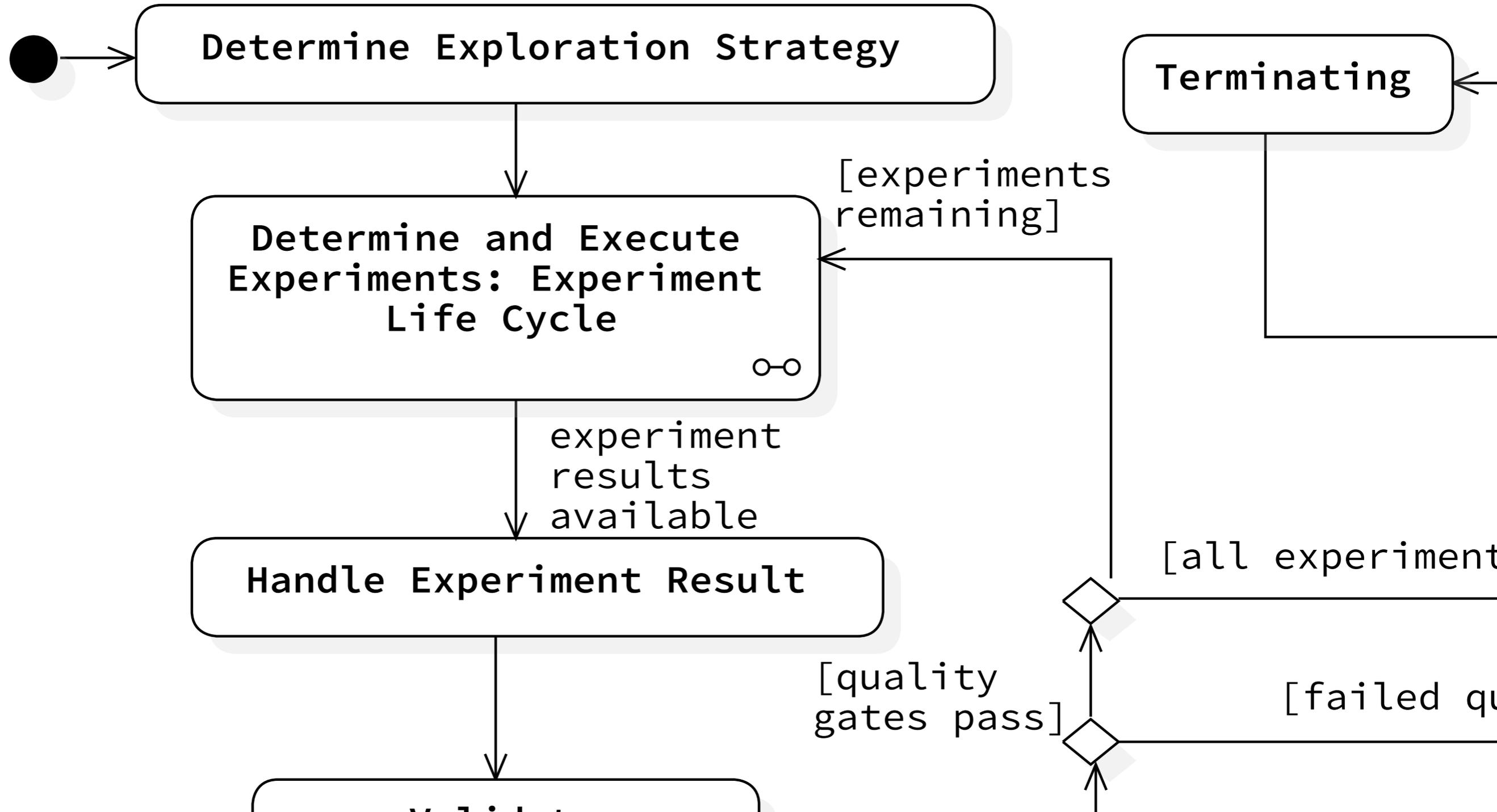
Exploration

Execution

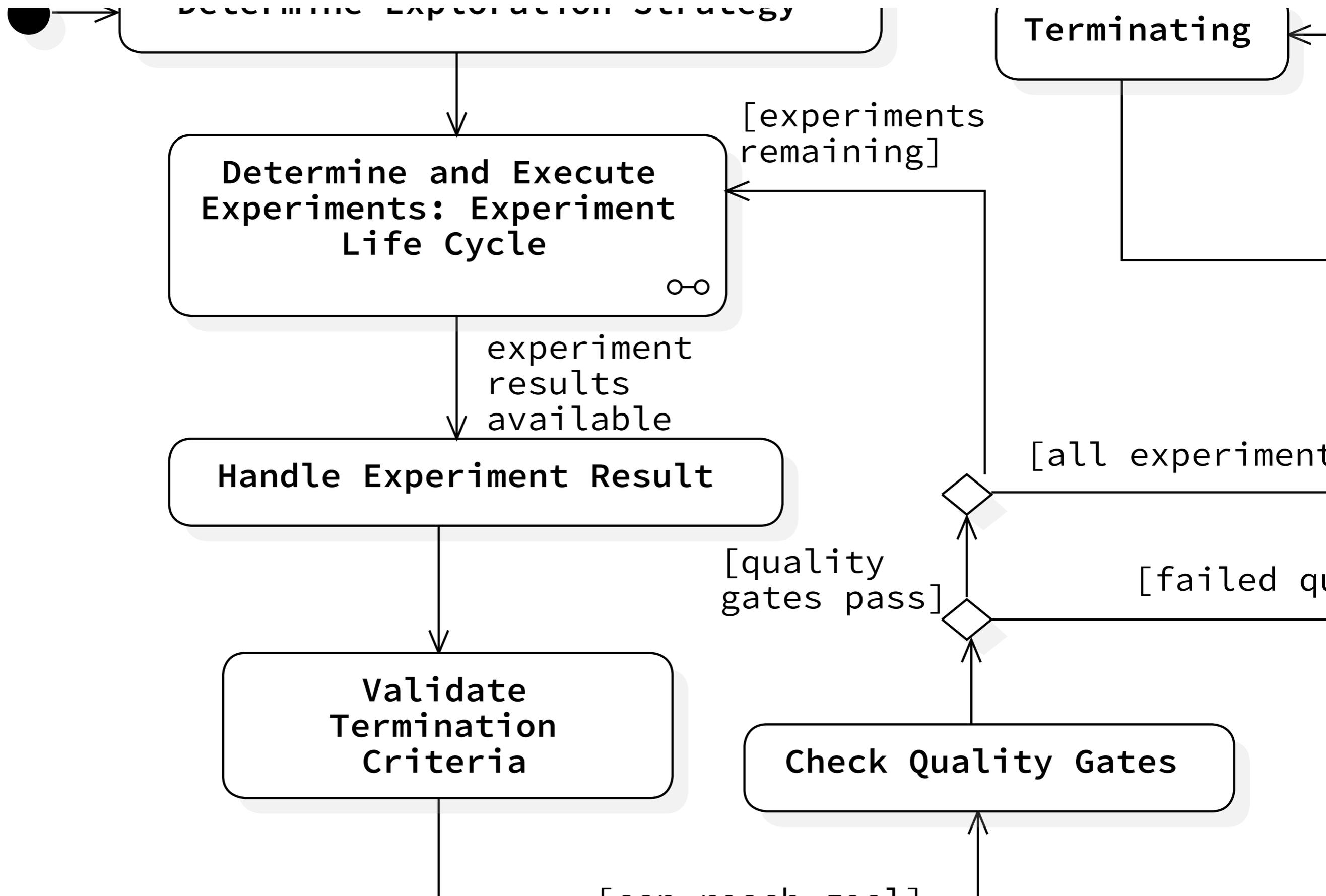
Analysis

Test Execution Lifecycle

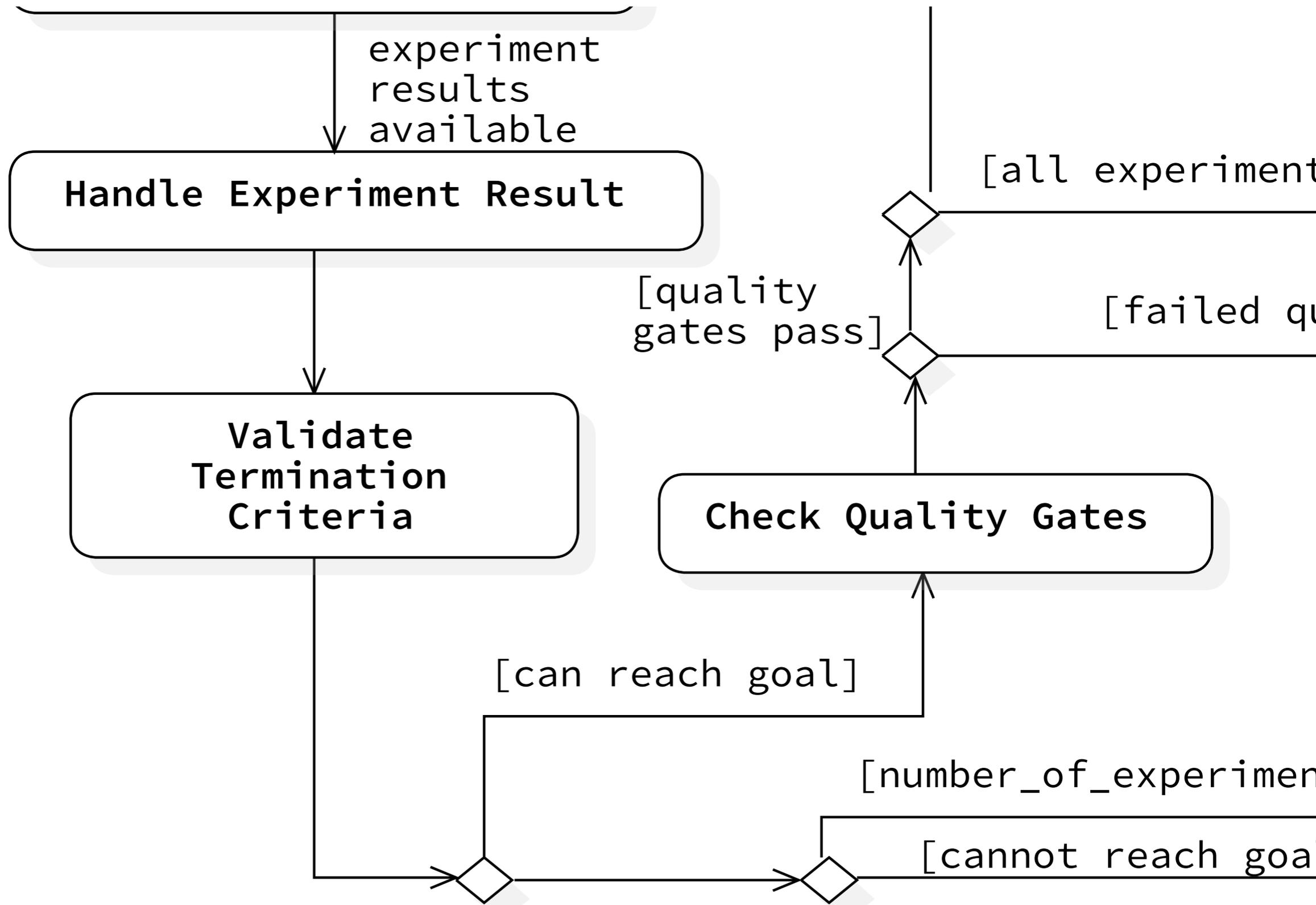
Running



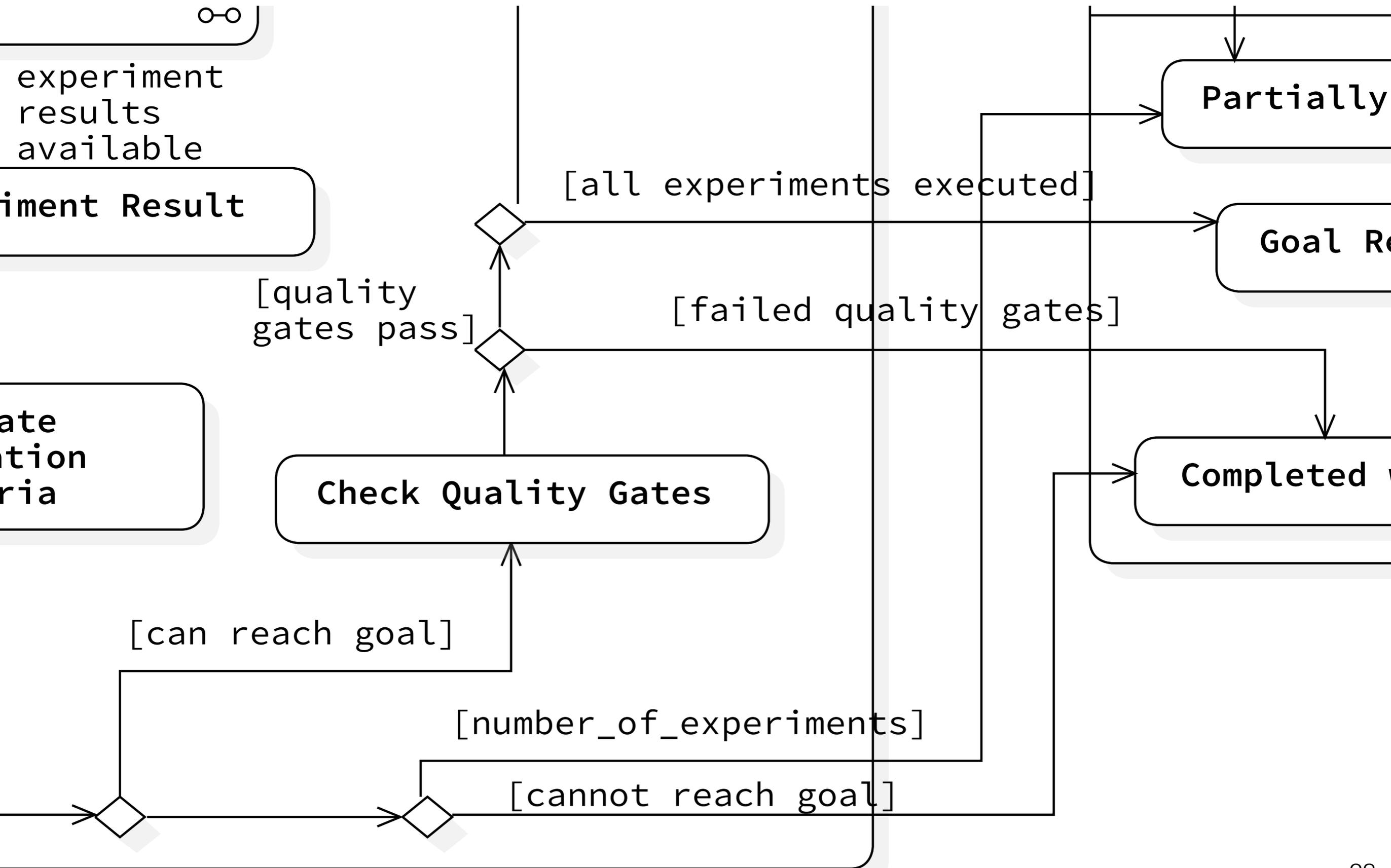
Test Execution Lifecycle



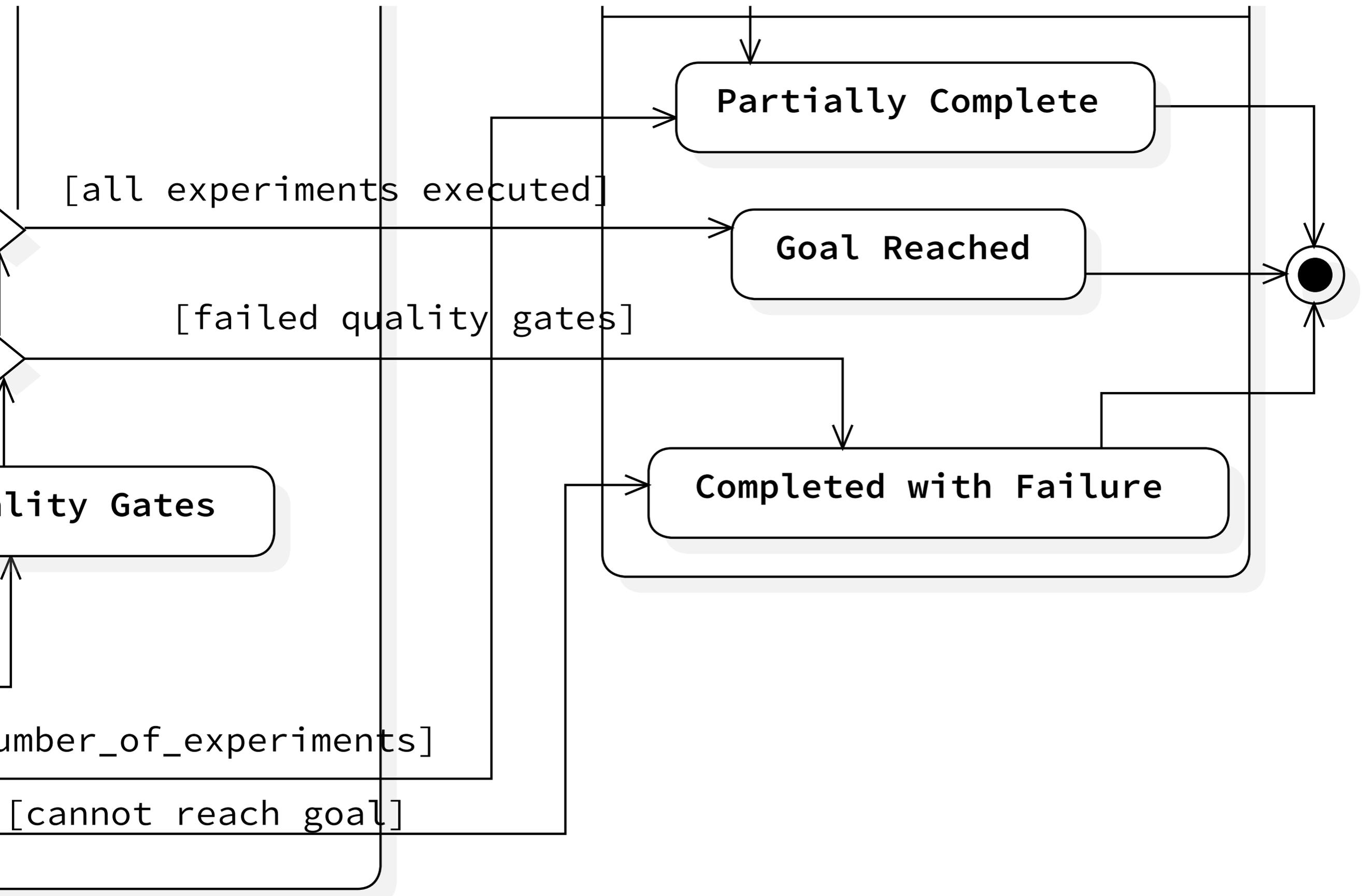
Test Execution Lifecycle



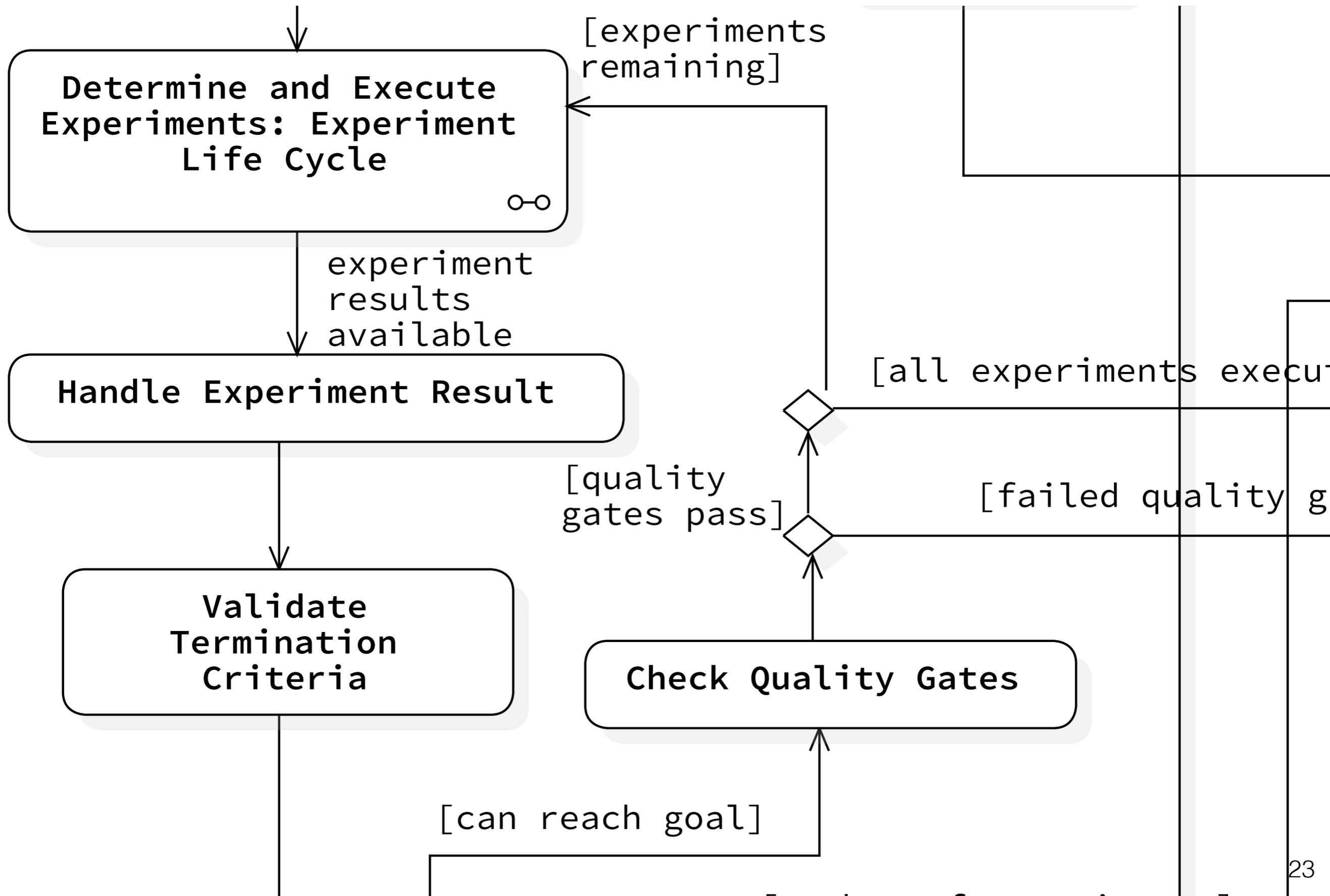
Test Execution Lifecycle



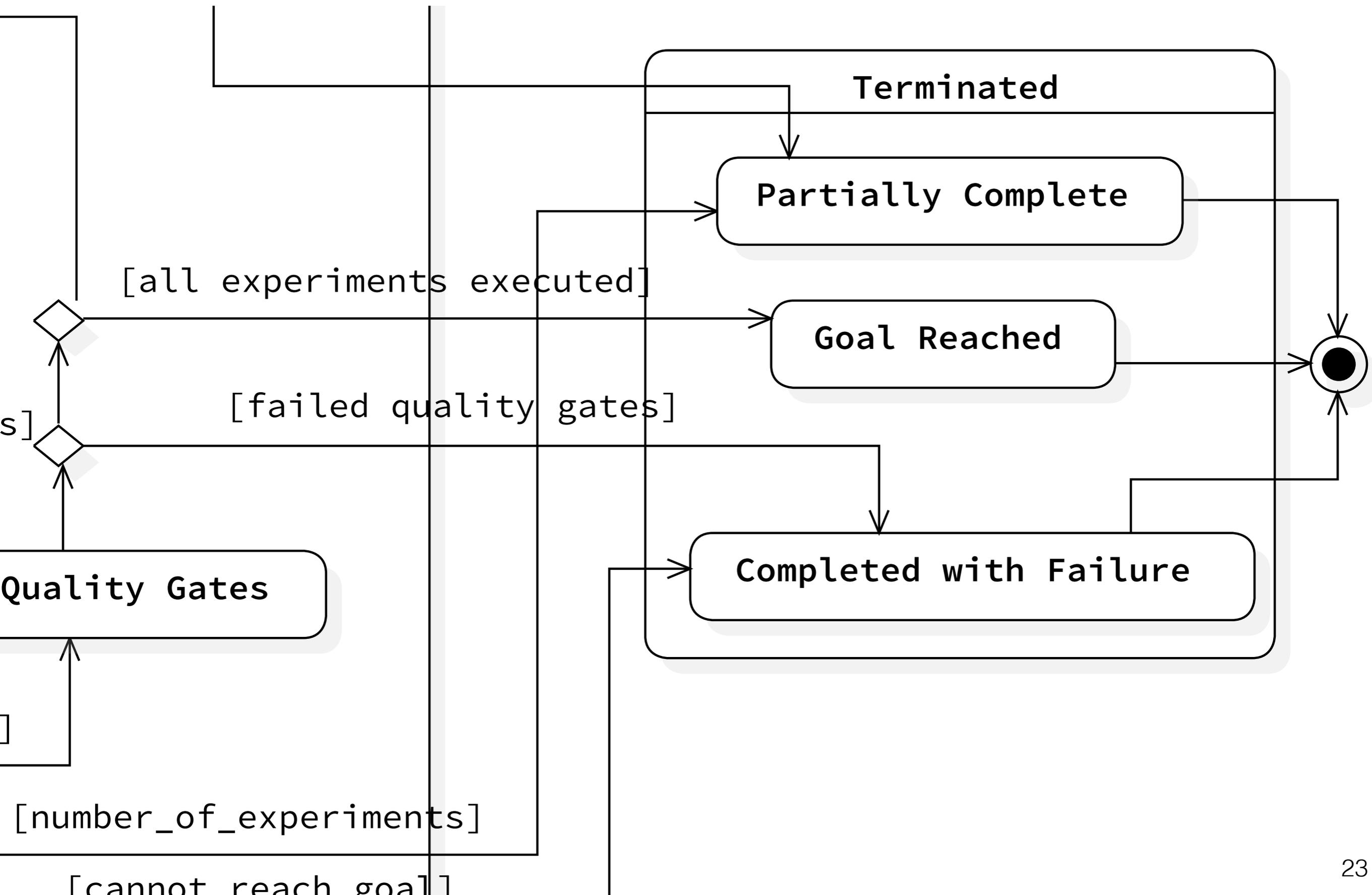
Test Execution Lifecycle



Test Execution Lifecycle



Test Execution Lifecycle



Extending the Proposed Approach

configuration:

goal:

type: **configuration**

observe:

exploration:

...

Extending the Proposed Approach

configuration:

goal:

type: **stability_boundary**

observe:

exploration:

...

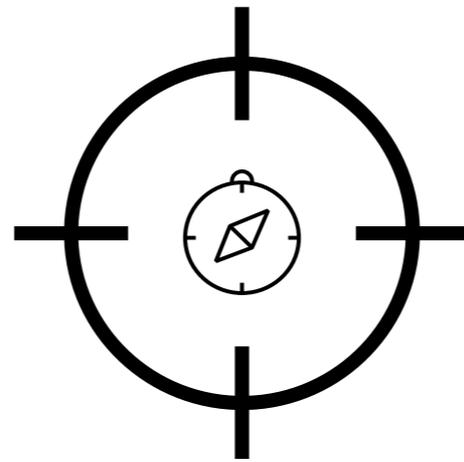
Available Goals:

LOAD

CONFIGURATION

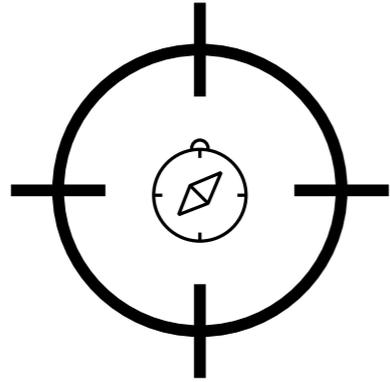
STABILITY_BOUNDARY

Extending the Proposed Approach



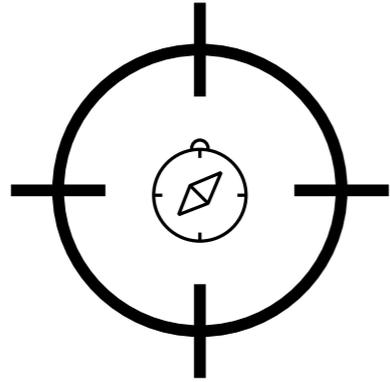
Declarative
Performance Testing DSL

Approach: Summary

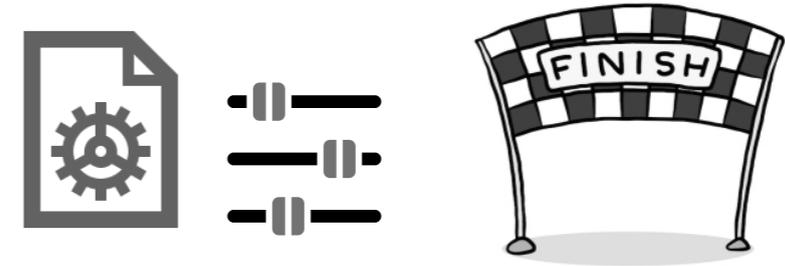


**Explicit Test Goals in
the Definition**

Approach: Summary

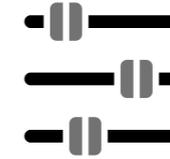
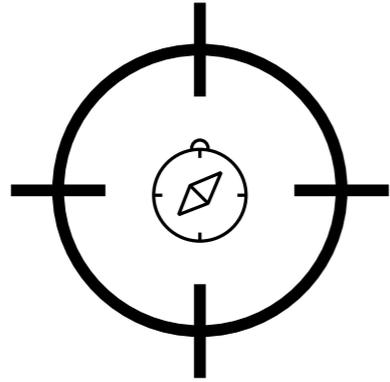


**Explicit Test Goals in
the Definition**



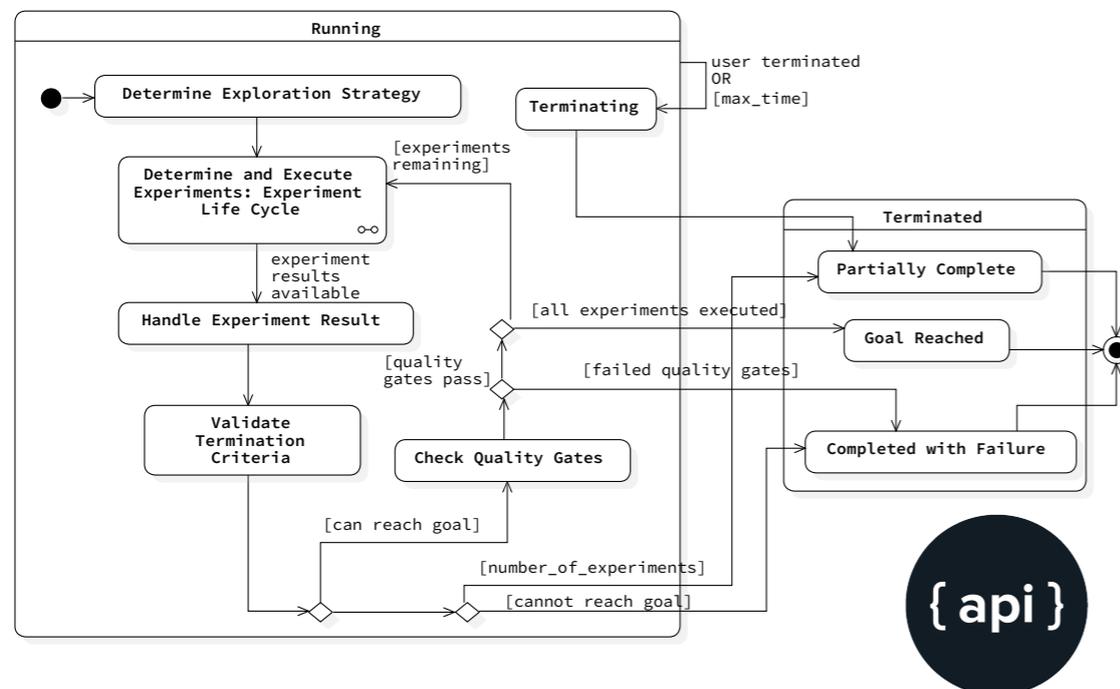
**Control on the Test
Execution Lifecycle**

Approach: Summary



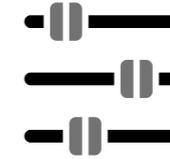
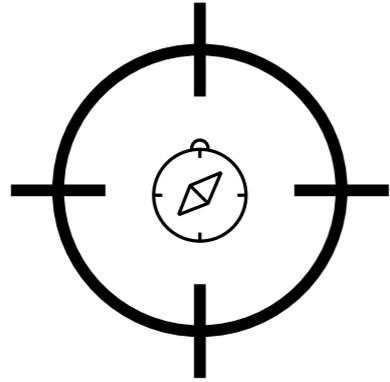
Explicit Test Goals in the Definition

Control on the Test Execution Lifecycle



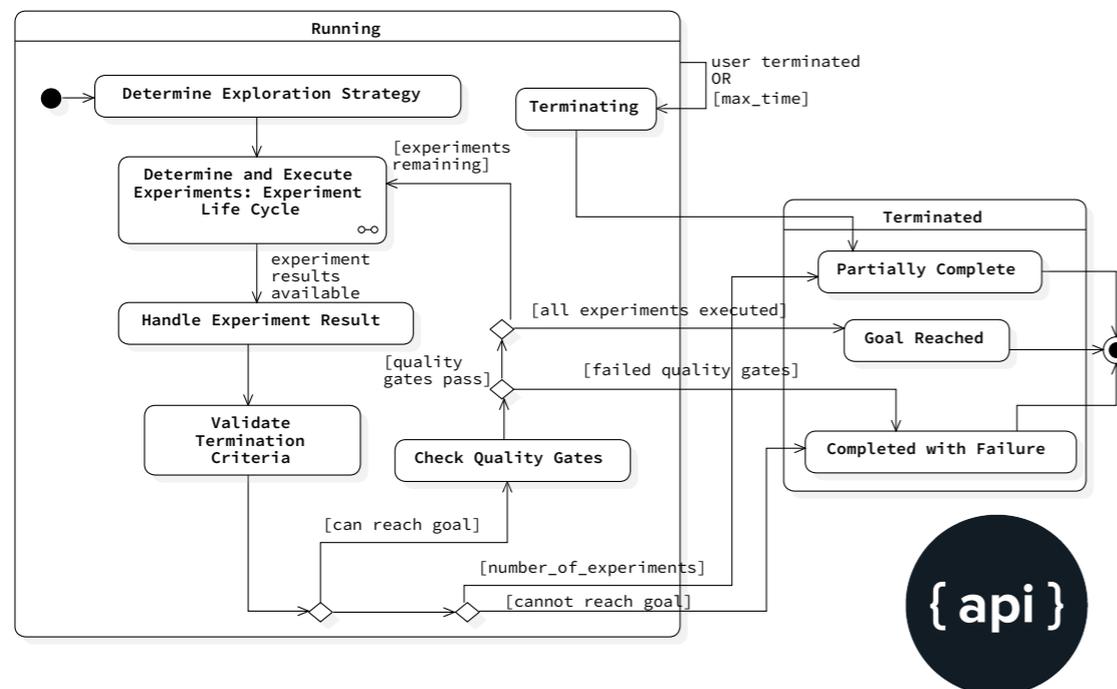
Visibility on the Test Execution Lifecycle

Approach: Summary



Explicit Test Goals in the Definition

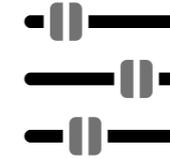
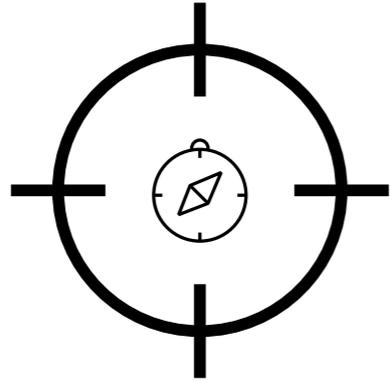
Control on the Test Execution Lifecycle



Visibility on the Test Execution Lifecycle

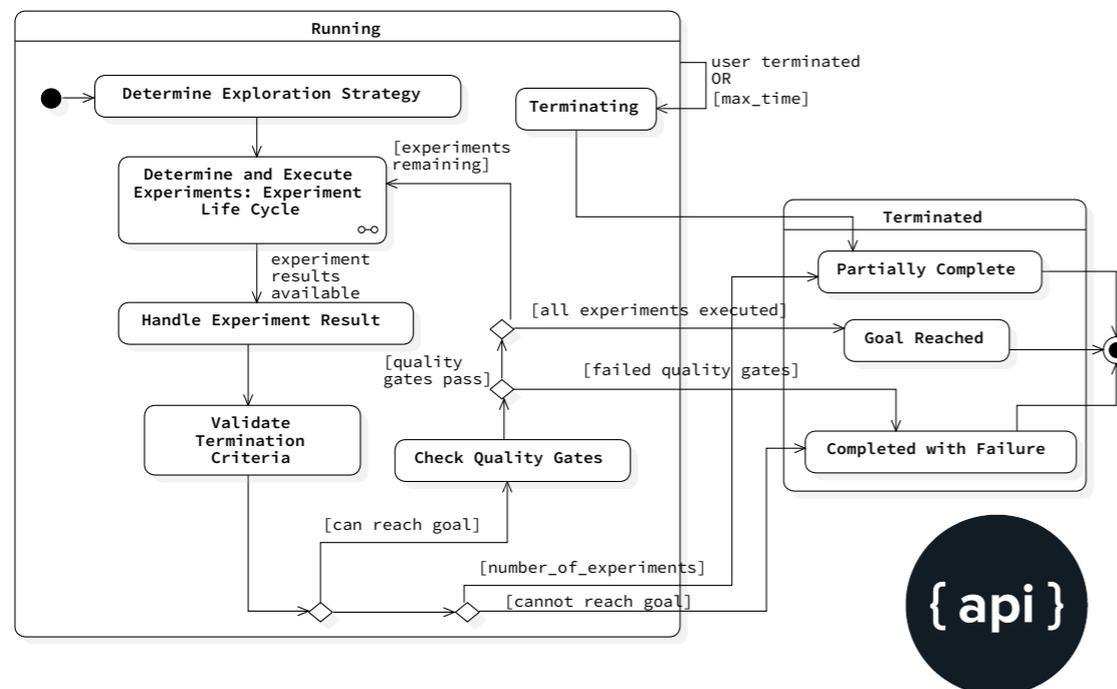
Static Syntactic and Semantic Validation

Approach: Summary



Explicit Test Goals in the Definition

Control on the Test Execution Lifecycle



Visibility on the Test Execution Lifecycle

Static Syntactic and Semantic Validation

Extensible for different Test Goals

Future Work

Apply to real-world use cases:

- Apply the approach in real-world contexts

Future Work

Apply to real-world use cases:

- Apply the approach in real-world contexts

Add more goals:

- Regression
- Assess different micro services deployment alternatives

Future Work

Apply to real-world use cases:

- Apply the approach in real-world contexts

Add more goals:

- Regression
- Assess different micro services deployment alternatives

Leverage performance test execution data:

- Prioritise performance test execution
- Skip execution of tests

Future Work

Apply to real-world use cases:

- Apply the approach in real-world contexts

Add more goals:

- Regression
- Assess different micro services deployment alternatives

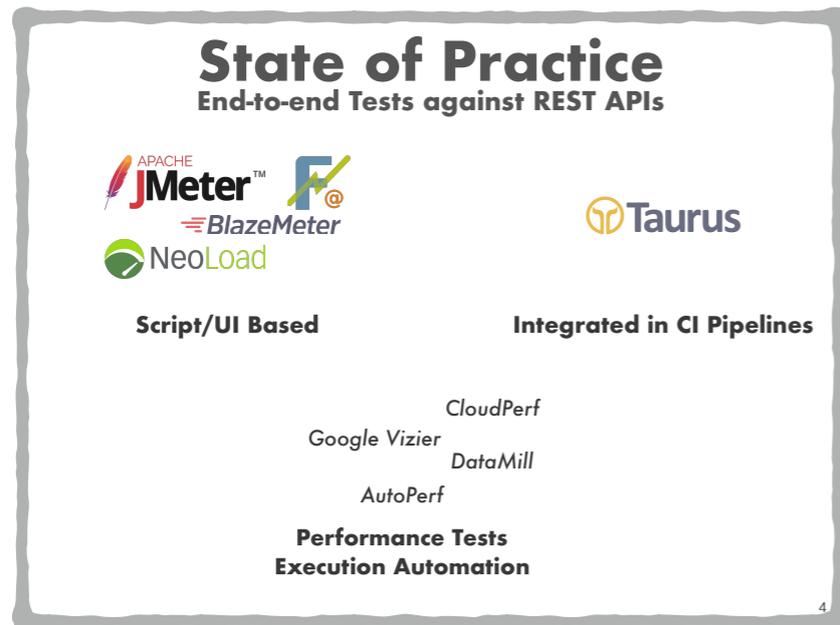
Leverage performance test execution data:

- Prioritise performance test execution
- Skip execution of tests

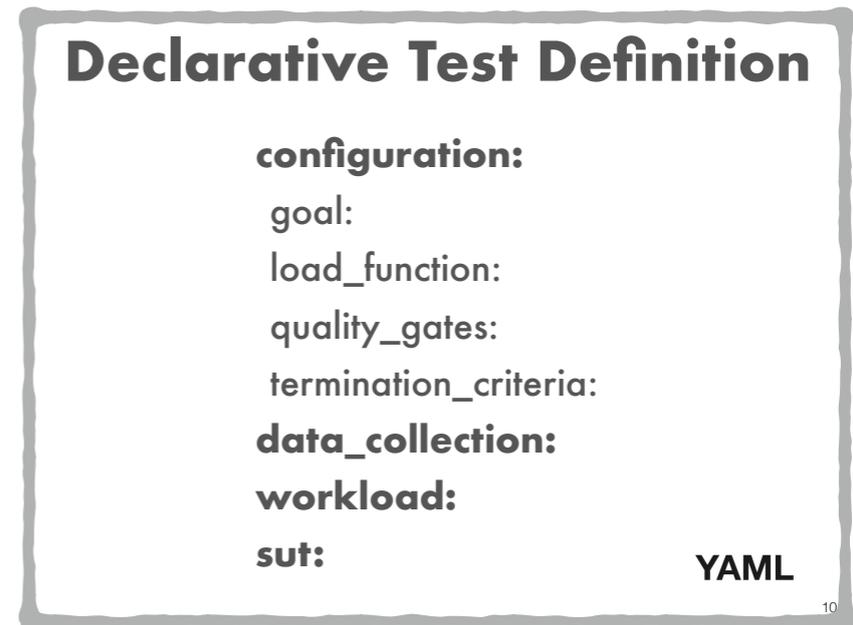
Integrate with other frameworks:

- ContinUjTy framework, to leverage production data

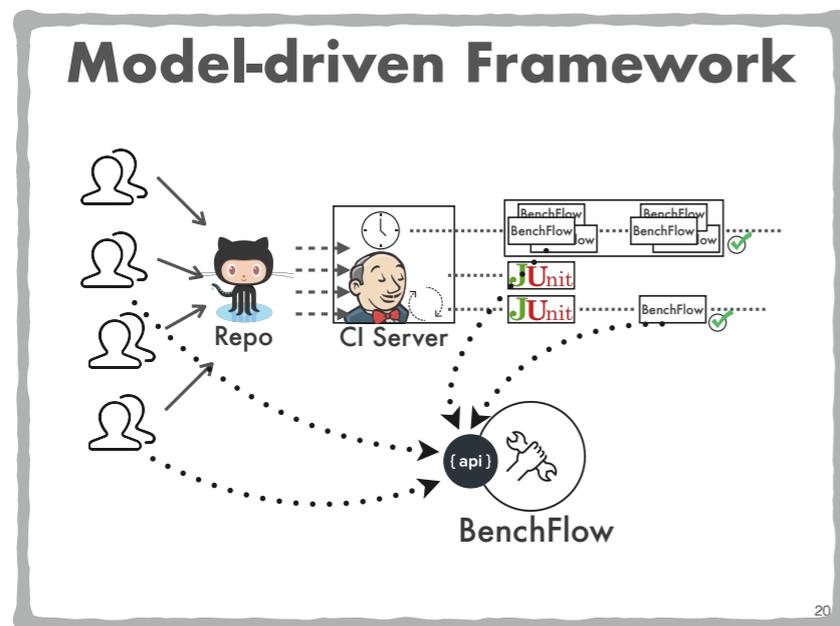
Highlights



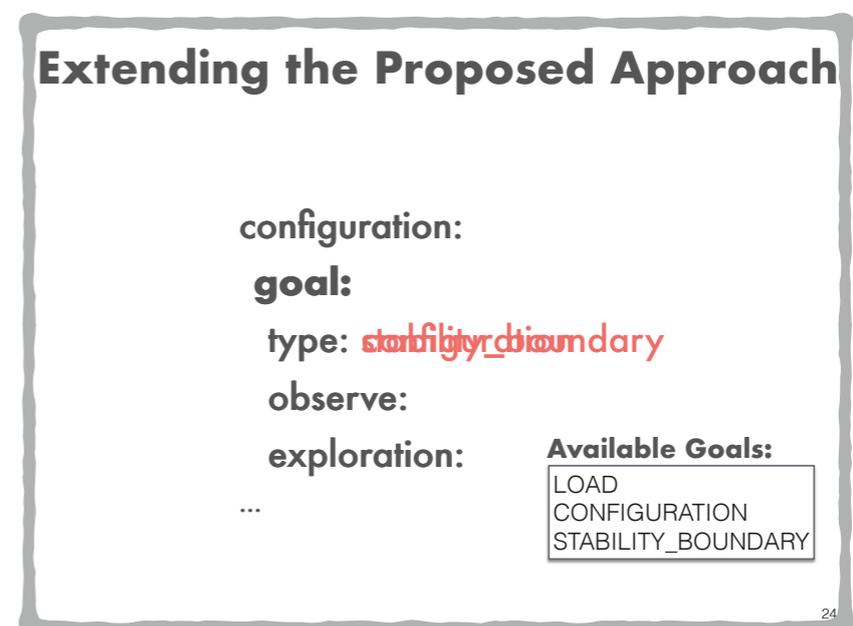
State of Practice



Declarative DSL



Model-driven Framework



Extending the Approach