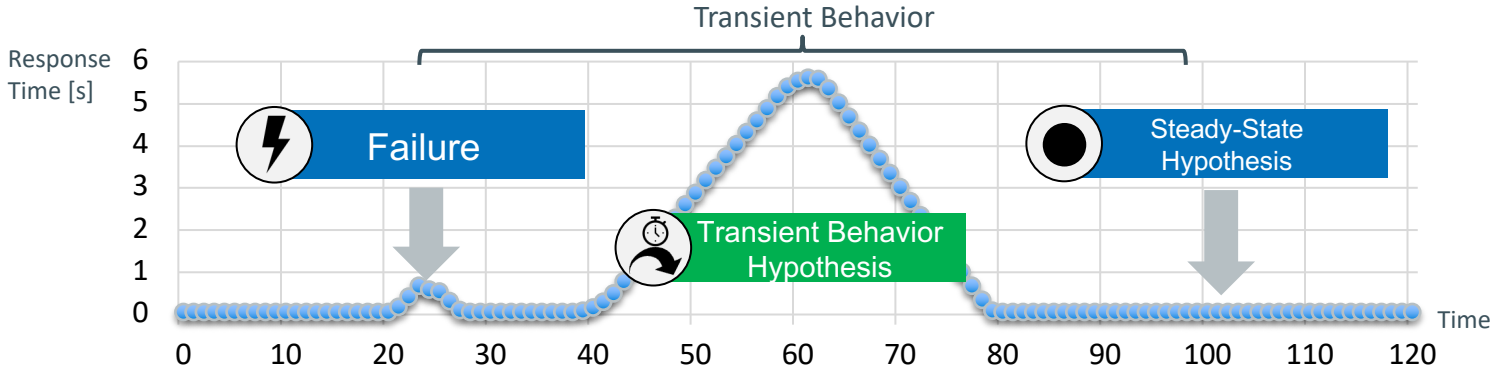
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG
University of Stuttgart
software campus
Bundesministerium für Bildung und Forschung
dqualizer
SPEC Research



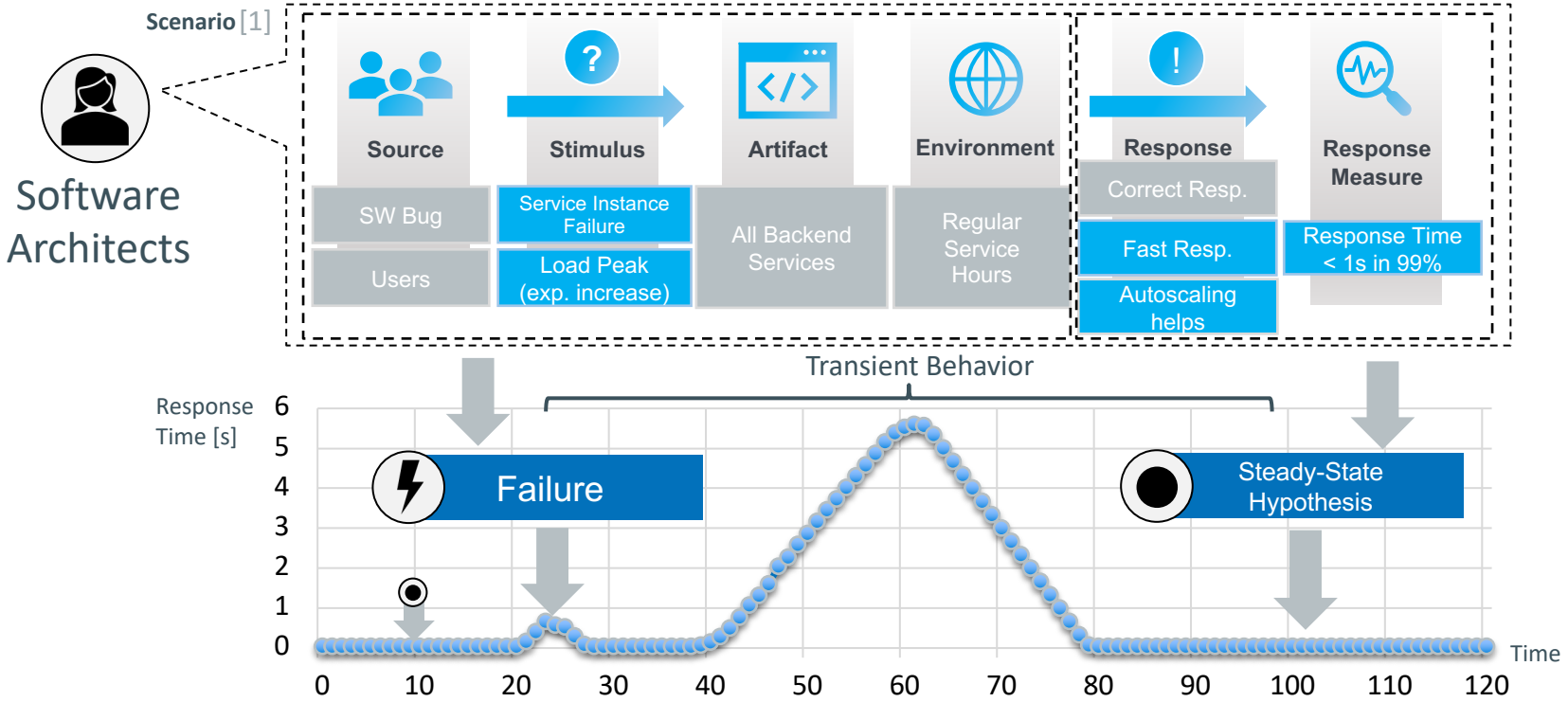
Sebastian Frank, Alireza Hakamian, Denis Zahariev, André van Hoorn

Verifying Transient Behavior Specifications in Chaos Engineering Using Metric Temporal Logic and Property Specification Patterns

[1] Frank, S., Hakamian, A., Wagner, L., Kesim, D., Zorn, C., von Kistowski, J., & van Hoorn, A. (2022). Interactive Elicitation of Resilience Scenarios Based on Hazard Analysis Techniques. In *European Conference on Software Architecture* (pp. 229-253). Springer, Cham.

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

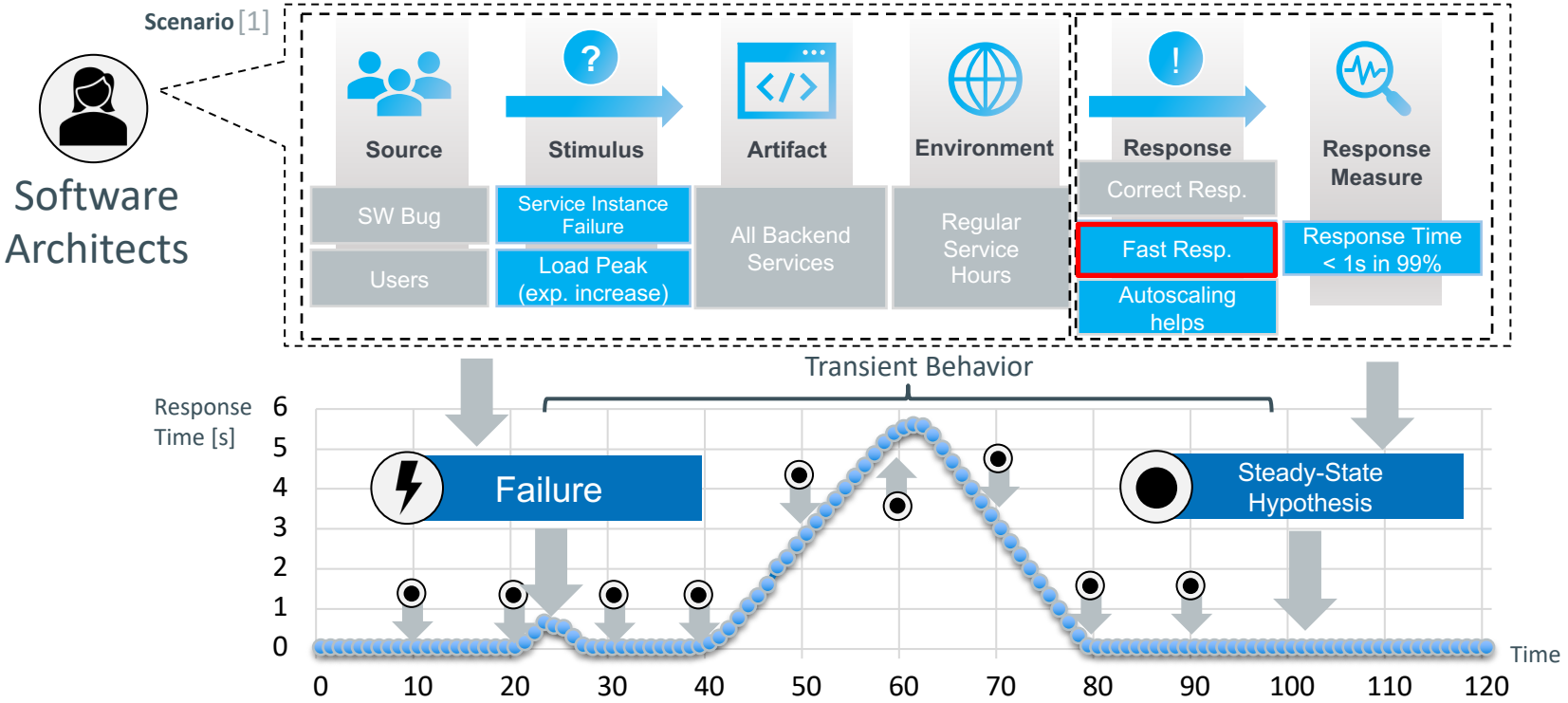
Transient Behavior in Chaos Engineering?



[1] Frank, S., Hakamian, A., Wagner, L., Kesim, D., Zorn, C., von Kistowski, J., & van Hoorn, A. (2022). Interactive Elicitation of Resilience Scenarios Based on Hazard Analysis Techniques. In *European Conference on Software Architecture* (pp. 229-253). Springer, Cham.

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

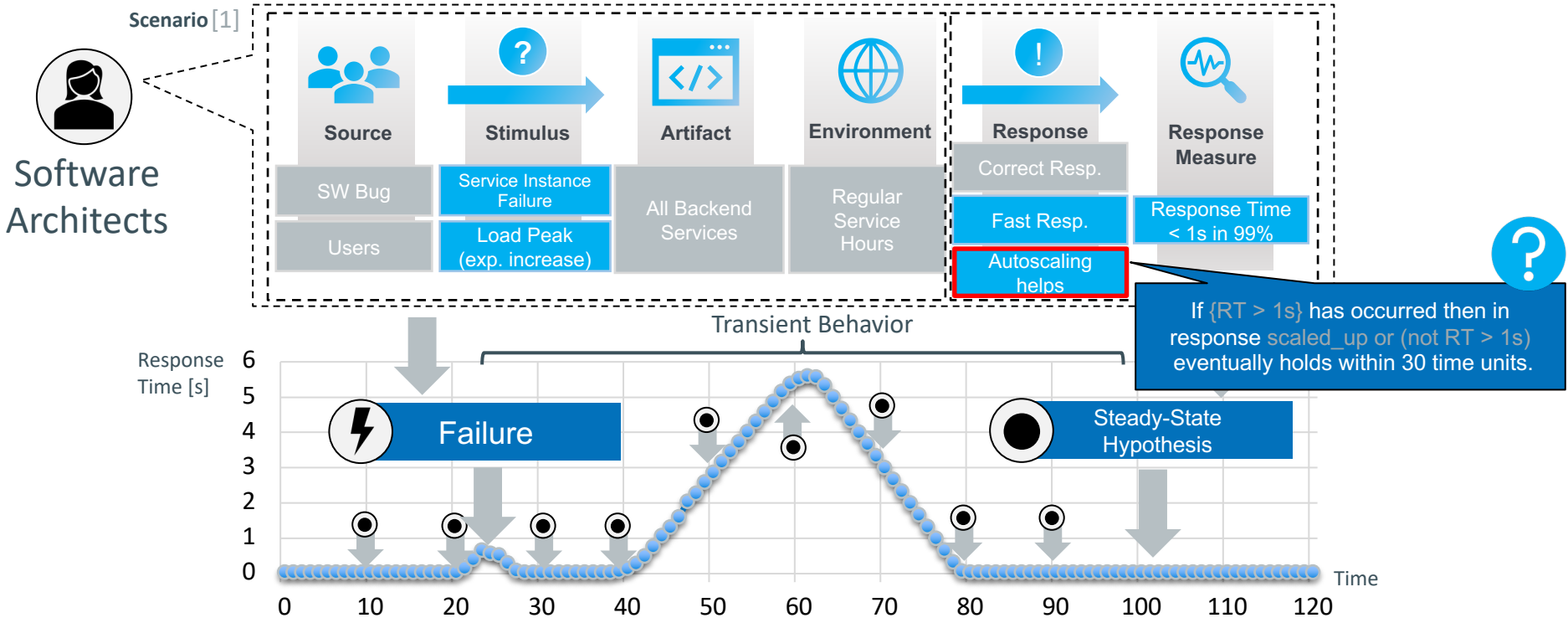
Transient Behavior in Chaos Engineering?



[1] Frank, S., Hakamian, A., Wagner, L., Kesim, D., Zorn, C., von Kistowski, J., & van Hoorn, A. (2022). Interactive Elicitation of Resilience Scenarios Based on Hazard Analysis Techniques. In *European Conference on Software Architecture* (pp. 229-253). Springer, Cham.

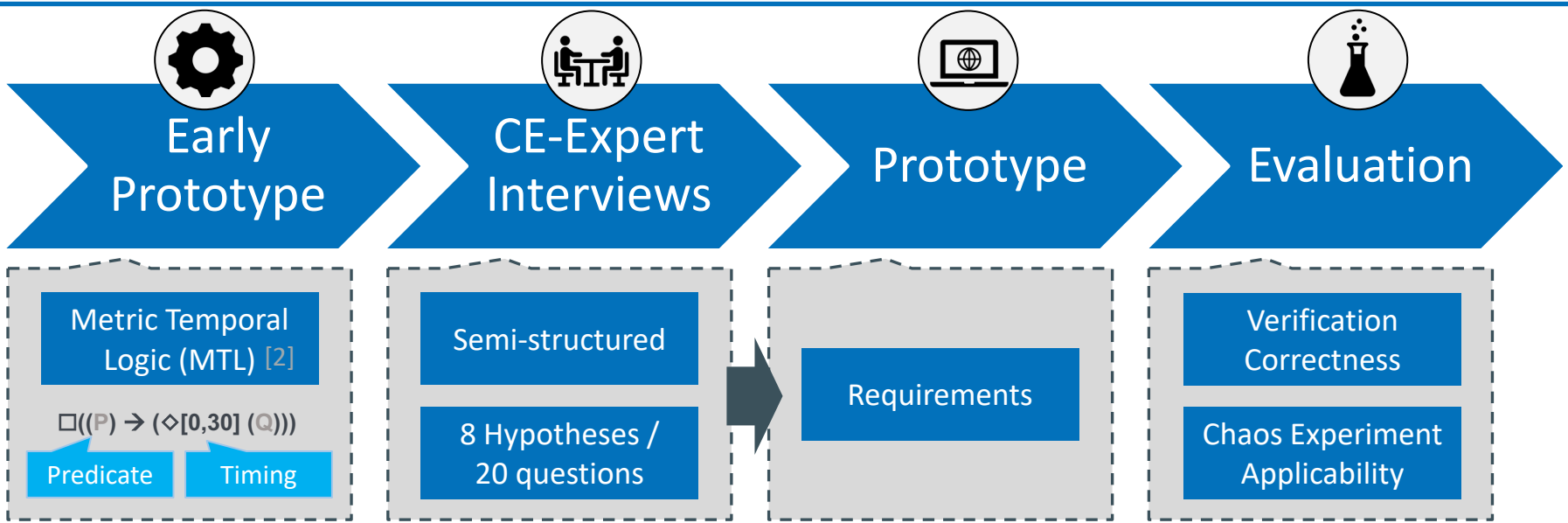
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Transient Behavior in Chaos Engineering?



[2] Koymans, Ron. "Specifying real-time properties with metric temporal logic." *Real-time systems* 2.4 (1990): 255-299.

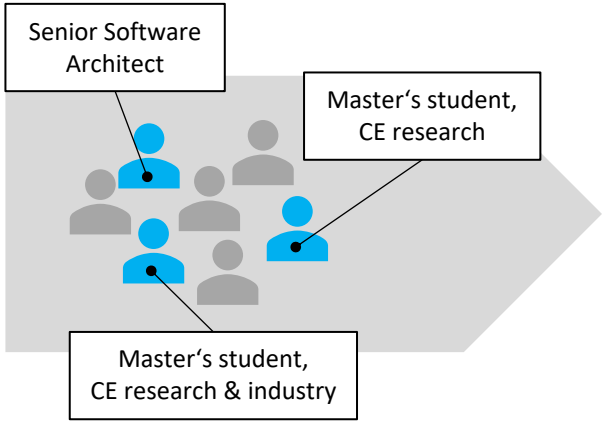
Method



[3] Autili, Marco, et al. "Aligning qualitative, real-time, and probabilistic property specification patterns using a structured english grammar." *IEEE Transactions on Software Engineering* 41.7 (2015): 620-638.

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CE-Expert Interviews: Impressions



Partially achieved
Lot of manual work
Successfully
Non-steady-state behavior verification
Outside of CE
Use Cases
Assistance helpful
MTL expressiveness & PSP readability
MTL Specification
Visualizations
More information than satisfaction
Verification
Extensibility
Useability!
Quality

Response Pattern

Globally, if P has occurred then in response Q eventually holds within 30 time units.

□((P) → (◇[0,30] (Q)))

Requirements

[3]

Property Specification Pattern (PSP) Support

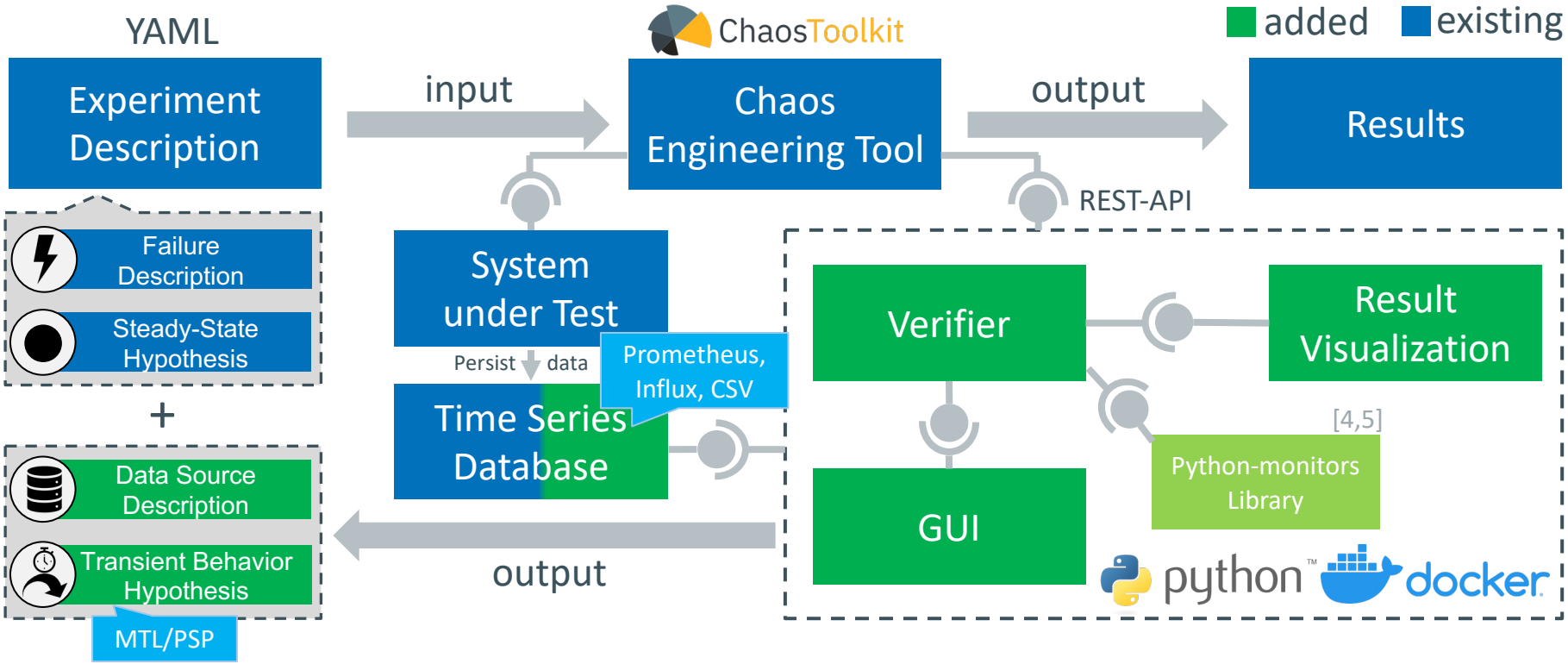
Visualization / GUI

Stand-alone tool

[4] Dogan Ulus. 2019. Online Monitoring of Metric Temporal Logic using Sequential Networks. *CoRR* (2019). arXiv:1901.00175 http://arxiv.org/abs/1901.00175

[5] Dogan Ulus. 2019. https://github.com/doganulus/python-monitors

Transient Behavior Verifier (Tool)



Using the Transient Behavior Verifier

```
"specification": "always((response_time_high(rt))
 -> (once[0,30] (scaled_up(scaling1) or scaled_up(scaling2)
  or (not response_time_high(rt))))",
"specification_type": "mtl",
"future-mtl": "true",
```

If {RT > 1s} has occurred then in response scaled_up or (not RT > 1s) eventually holds within 30 time units.

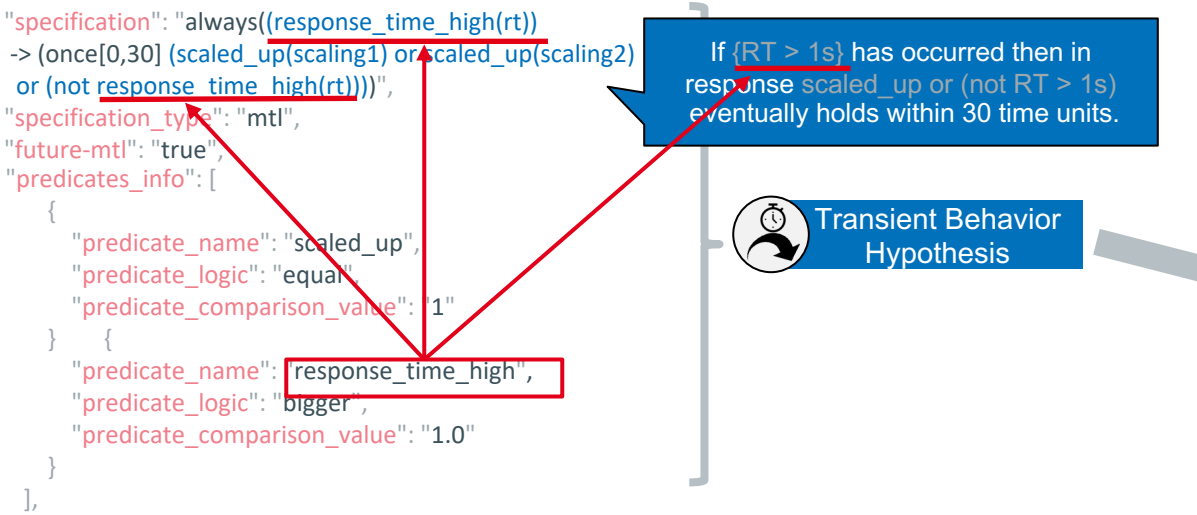
Transient Behavior Hypothesis

Metrics

Verifier

Result Visualization

Using the Transient Behavior Verifier



If {RT > 1s} has occurred then in response scaled_up or (not RT > 1s) eventually holds within 30 time units.

Transient Behavior Hypothesis

Metrics

Verifier

Result Visualization

Using the Transient Behavior Verifier

```
"specification": "always((response_time_high(rt))
 -> (once[0,30] (scaled_up(scaling1) or scaled_up(scaling2)
  or (not response_time_high(rt)))))",
"specification_type": "mtl",
"future-mtl": "true",
"predicates_info": [
    {
      "predicate_name": "scaled_up",
      "predicate_logic": "equal",
      "predicate_comparison_value": "1"
    }    {
      "predicate_name": "response_time_high",
      "predicate_logic": "bigger",
      "predicate_comparison_value": "1.0"
    }
  ],
"measurement_source": "influx",
  "measurement_points": [
    {
      "measurement_name": "rt",
      "measurement_column": "SELECT \"AvgResponseTime\"
FROM \"TimeBatchRuns\".\"autogen\".\"Batch_Time\""
    },
    [same for 'scaling_ex1' and 'scaling_ex2']
```

If {RT > 1s} has occurred then in response scaled_up or (not RT > 1s) eventually holds within 30 time units.

Transient Behavior Hypothesis

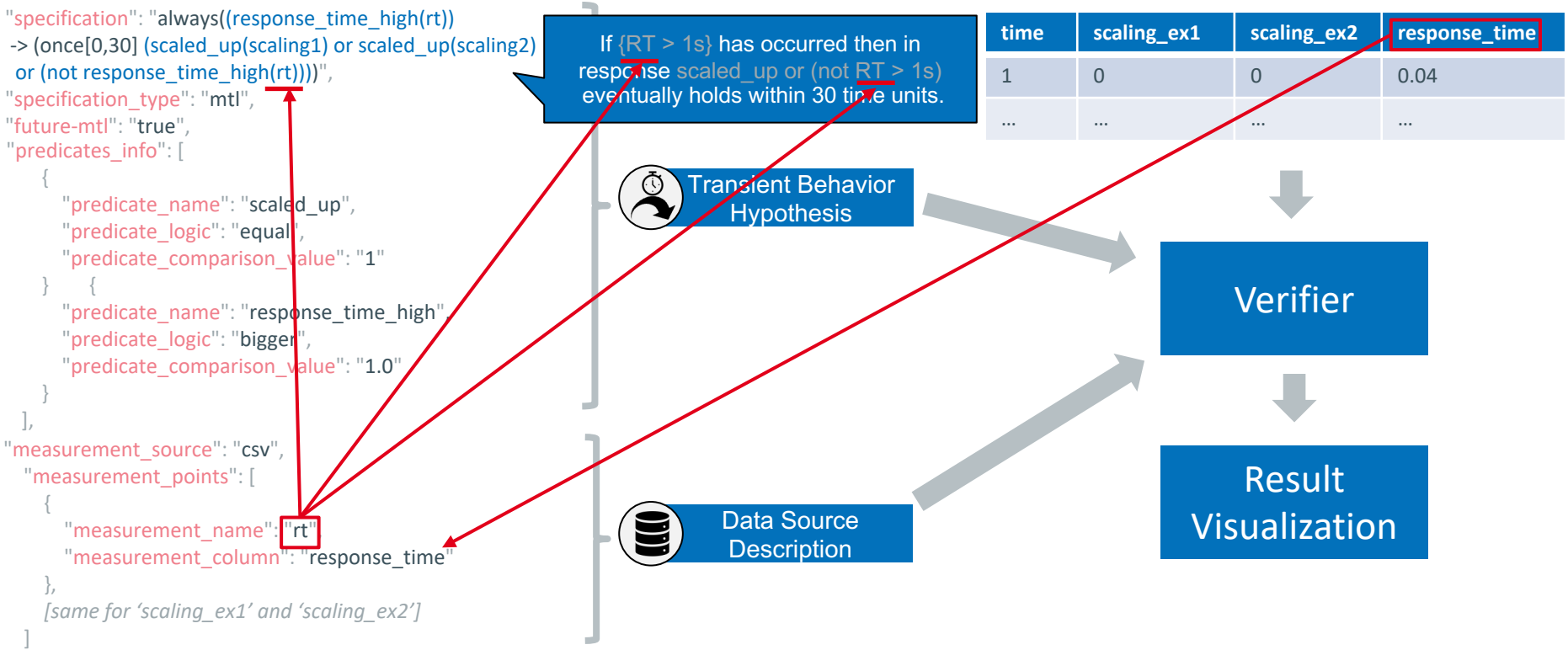
Data Source Description

Metrics

Verifier

Result Visualization

Using the Transient Behavior Verifier



Using the Transient Behavior Verifier


```
"specification": "always((response_time_high(rt))
 -> (once[0,30] (scaled_up(scaling1) or scaled_up(scaling2)
  or (not response_time_high(rt)))))",
"specification_type": "mtl",
"future-mtl": "true",
"predicates_info": [
    {
      "predicate_name": "scaled_up",
      "predicate_logic": "equal",
      "predicate_comparison_value": "1"
    }    {
      "predicate_name": "response_time_high",
      "predicate_logic": "bigger",
      "predicate_comparison_value": "1.0"
    }
  ],
"measurement_source": "csv",
  "measurement_points": [
    {
      "measurement_name": "rt",
      "measurement_column": "response_time"
    },
    [same for 'scaling_ex1' and 'scaling_ex2']
  ]
```

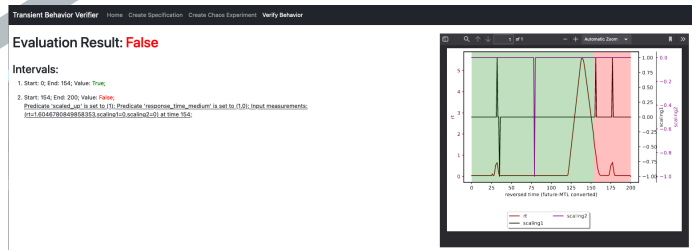
If {RT > 1s} has occurred then in response scaled_up or (not RT > 1s) eventually holds within 30 time units.

time	scaling_ex1	scaling_ex2	response_time
1	0	0	0.04
...	...	...	...

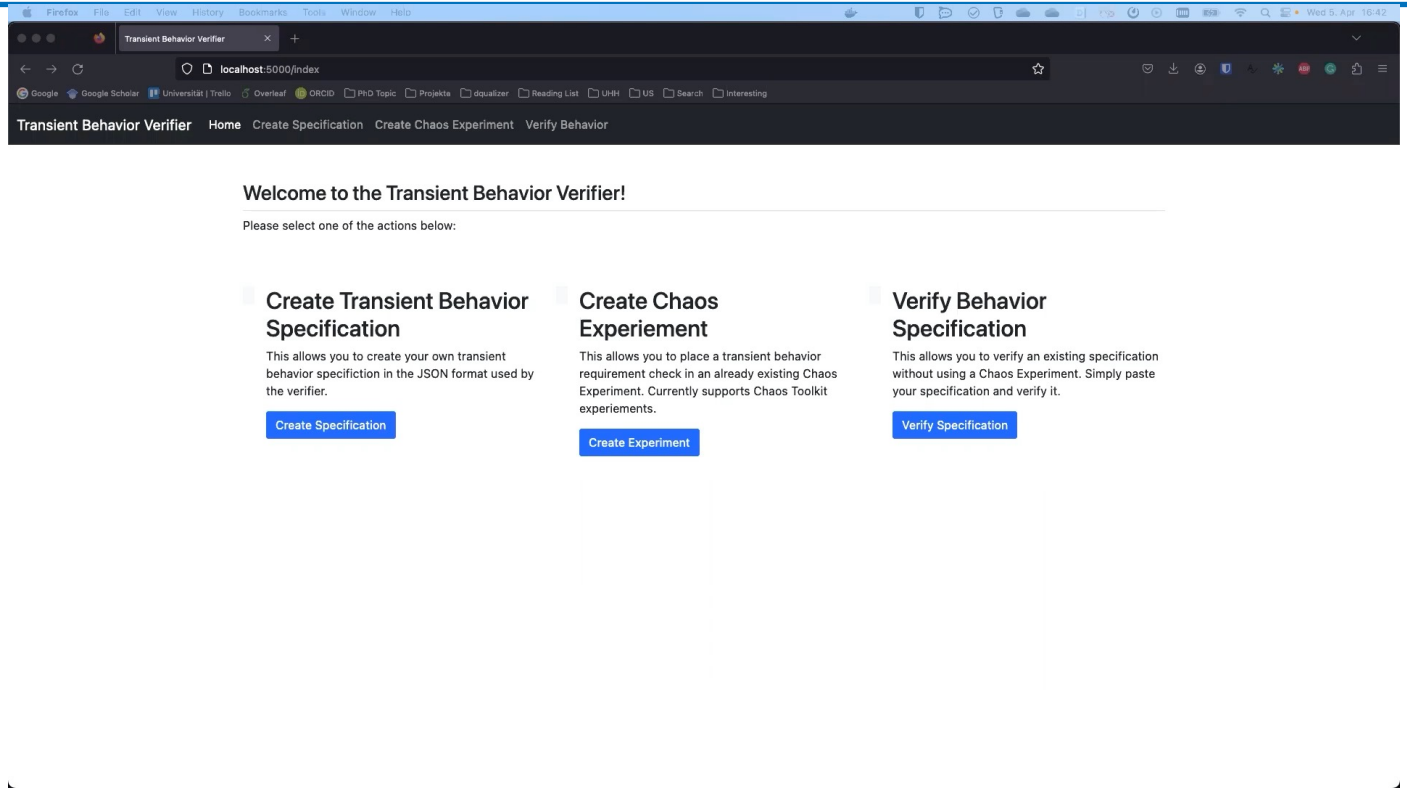
Transient Behavior Hypothesis

Data Source Description

Verifier



Tool Demo


Transient Behavior Verifier
localhost:5000/index
Transient Behavior Verifier
Home
Create Specification
Create Chaos Experiment
Verify Behavior
Welcome to the Transient Behavior Verifier!
Please select one of the actions below:
Create Transient Behavior Specification
This allows you to create your own transient behavior specifiction in the JSON format used by the verifier.
Create Specification
Create Chaos Experiement
This allows you to place a transient behavior requirement check in an already existing Chaos Experiment. Currently supports Chaos Toolkit experiements.
Create Experiment
Verify Behavior Specification
This allows you to verify an existing specification without using a Chaos Experiment. Simply paste your specification and verify it.
Verify Specification

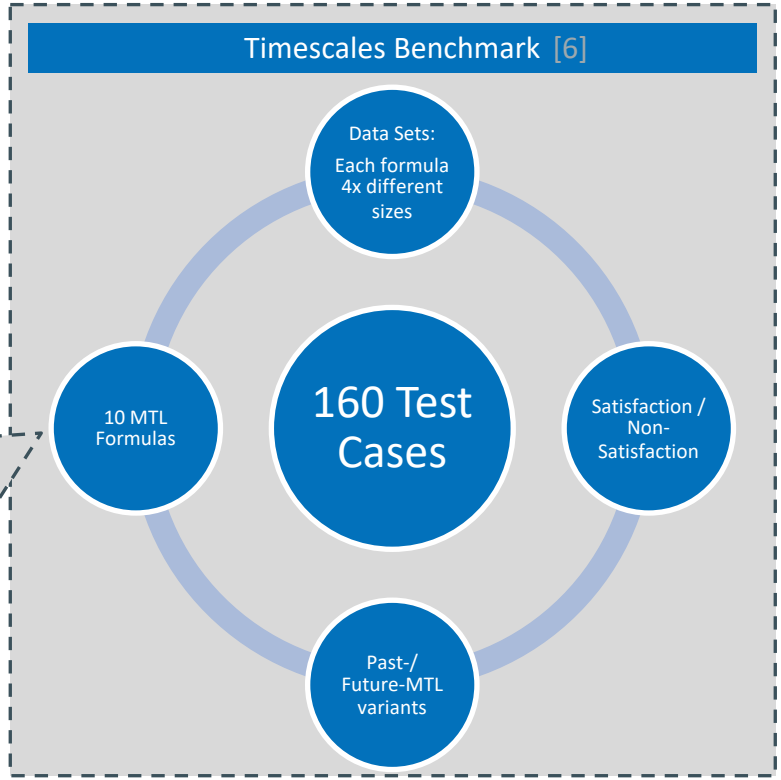
[6] Ulus, D. (2019, October). Timescales: A benchmark generator for MTL monitoring tools. In *International Conference on Runtime Verification* (pp. 402-412). Springer, Cham.

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Evaluation: Verification Correctness

Only 4 PSP types

Pattern	Scopes
Absence	3
Universality	3
Recurrence	2
Response	2



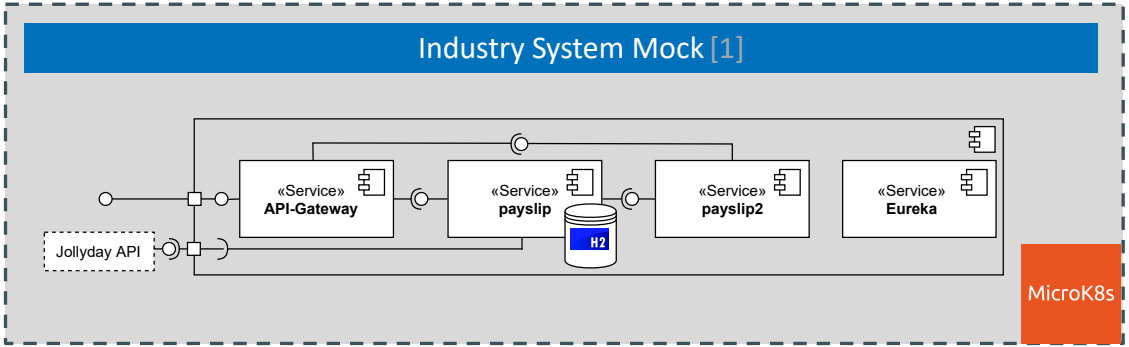
Transient Behavior Verifier

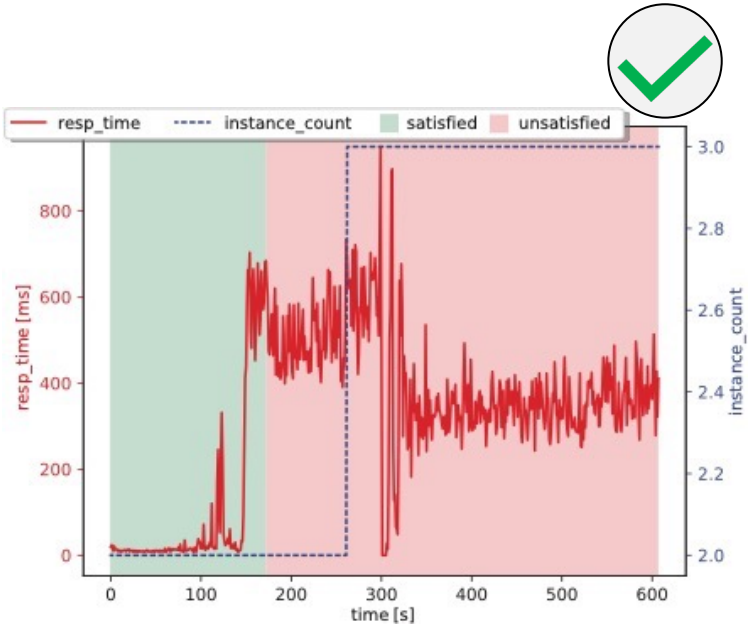
[1] Frank, S., Hakamian, A., Wagner, L., Kesim, D., Zorn, C., von Kistowski, J., & van Hoorn, A. (2022). Interactive Elicitation of Resilience Scenarios Based on Hazard Analysis Techniques. In *European Conference on Software Architecture* (pp. 229-253). Springer, Cham.

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Evaluation: Applicability in Chaos Experiments

	Chaos Experiment #1	Chaos Experiment #2	Chaos Experiment #3
Fault [1]	Instance Crash	Workload Increase	Workload Increase
Steady-State Hyp.	Depl. fully available	Depl. fully available	Depl. fully available
Transient-State Hyp.	New Instance in <30s; RT < 100ms	When RT > 150ms: new instance in 30-60s	Pod CPU usage <80%
Specification Type	Past-MTL	PSP	Past-MTL
Monitoring	InfluxDB	InfluxDB	Prometheus

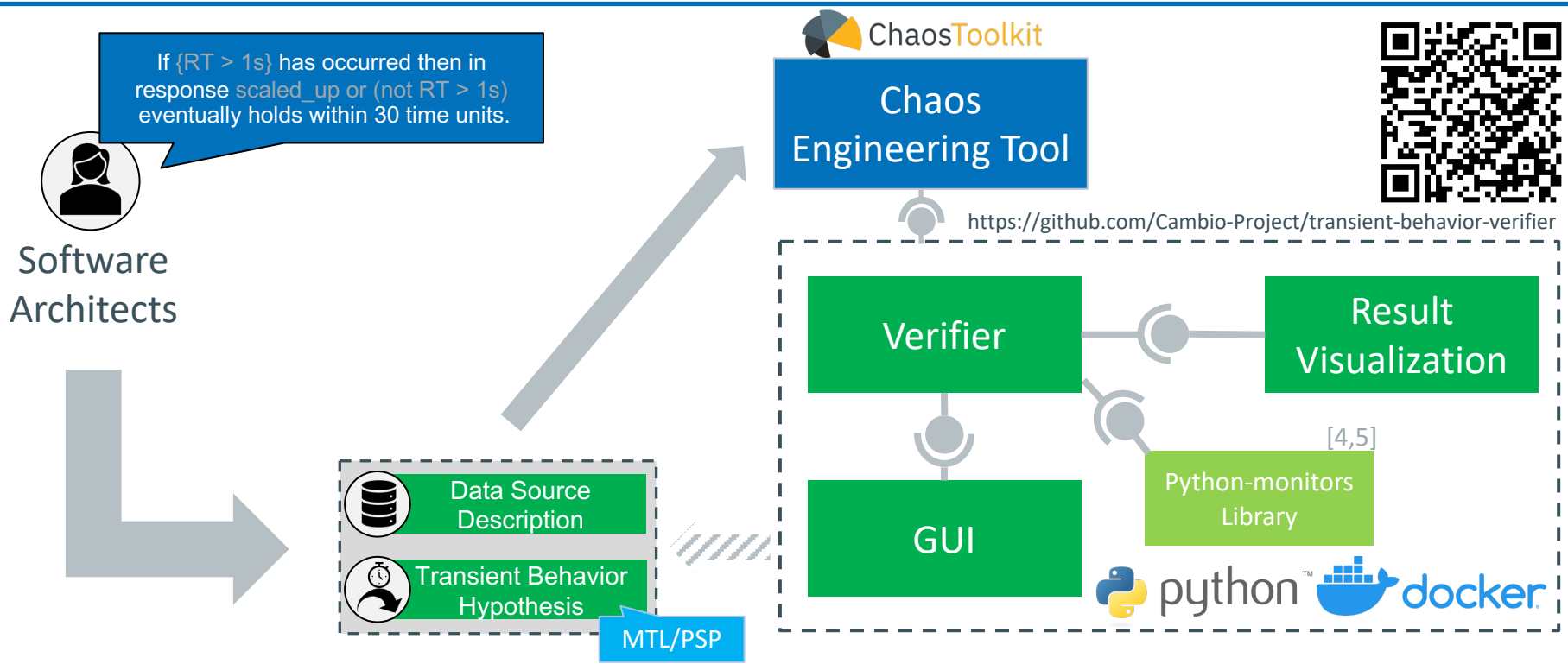




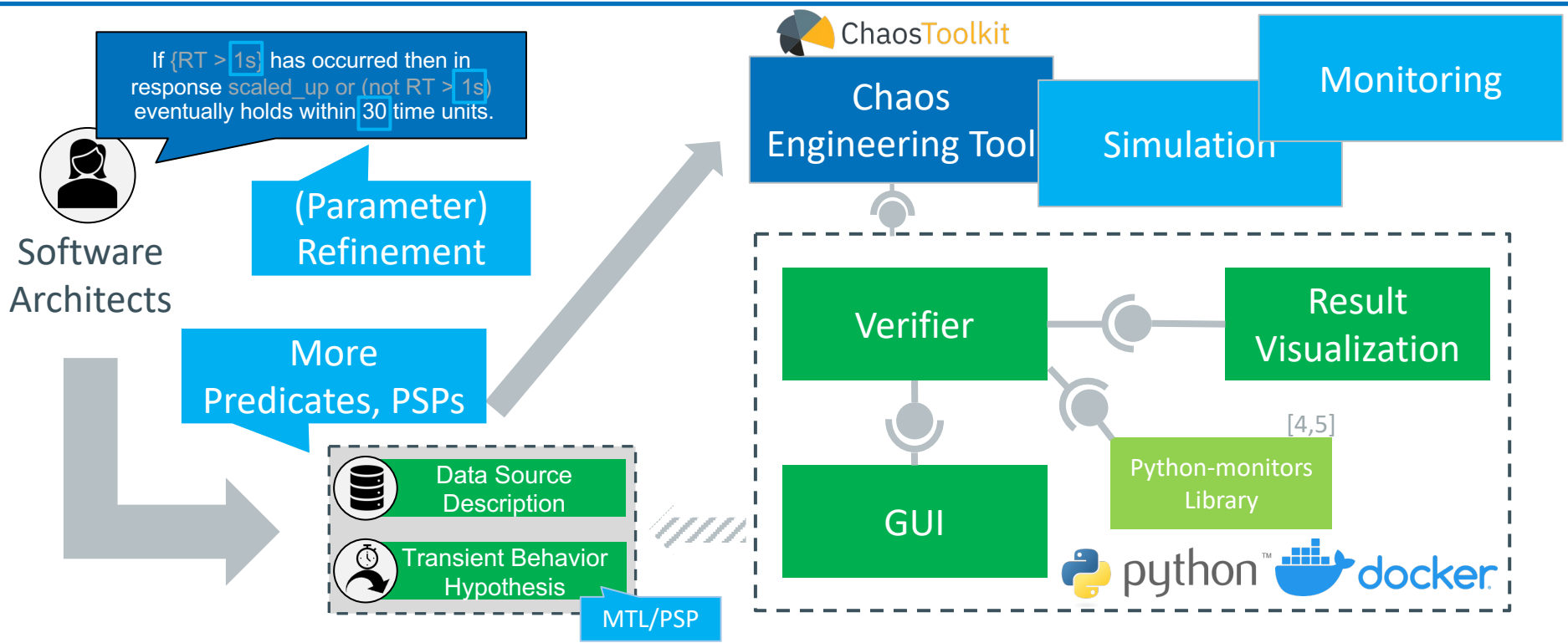
[4] Dogan Ulus. 2019. Online Monitoring of Metric Temporal Logic using Sequential Networks. *CoRR* (2019). arXiv:1901.00175 http://arxiv.org/abs/1901.00175

[5] Dogan Ulus. 2019. https://github.com/doganulus/python-monitors

Summary



[4] Dogan Ulus. 2019. Online Monitoring of Metric Temporal Logic using Sequential Networks. *CoRR* (2019). arXiv:1901.00175 http://arxiv.org/abs/1901.00175

[5] Dogan Ulus. 2019. https://github.com/doganulus/python-monitors

Future Work