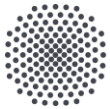


University of Stuttgart
Germany
SQA

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MiSim — A Lightweight and Extensible Simulator for a Scenario-based Resilience Evaluation of Microservice Architectures @SSP2021

André van Hoorn
Alireza Hakamian
Sebastian Frank
Lion Wagner

Orcas

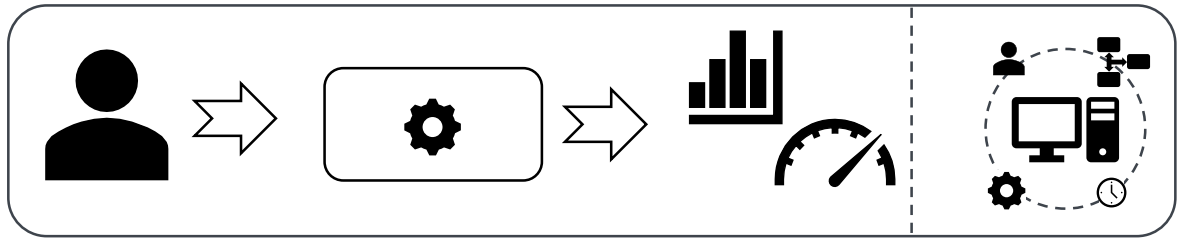
Cambio

Baden-Württemberg Stiftung

Bundesministerium für Bildung und Forschung

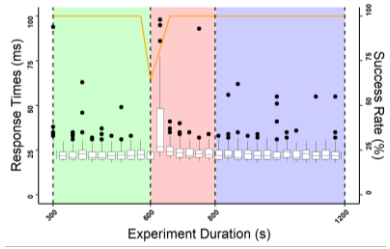
Scientific Background and Motivation

Scientific Background and Motivation



12 Resilience Scenario[1]

ChaosToolkit (CTK) Experiments



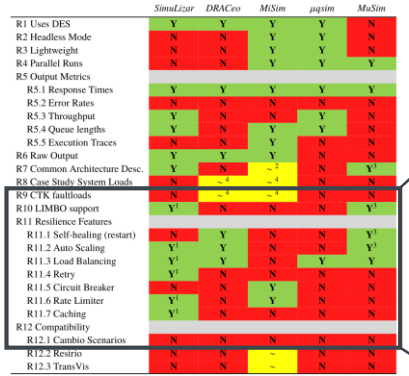
predict/ prioritize

Simulation

+ Lower execution costs
+ Allows Parallel Running
+ Can Run on dedicated hardware

24 Requirements for the Simulator

Requirements Evaluation



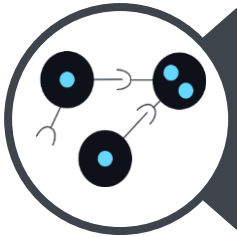
	SimuLizar	*DRACeo*	*MiSim*	*µqsim*	*MuSim*
R1 Uses DES	Y	Y	Y	Y	N
R2 Headless Mode	N	N	Y	Y	N
R3 Lightweight	N	N	Y	Y	N
R4 Parallel Runs	N	N	Y	Y	Y
R5 Output Metrics					
R5.1 Response Times	Y	Y	Y	Y	Y
R5.2 Error Rates	N	N	N	N	N
R5.3 Throughput	Y	N	N	Y	N
R5.4 Queue lengths	Y	N	Y	Y	N
R5.5 Execution Traces	N	N	Y	N	N
R6 Raw Output	Y	Y	Y	N	N
R7 Common Architecture Desc.	Y	N	~ [4]	N	Y[3]
R8 Case Study System Loads	N	~ [4]	~ [4]	N	N
R9 CTK faultloads	N	~ [4]	~ [4]	N	N
R10 LIMBO support	Y[1]	N	N	N	Y[3]
R11 Resilience Features					
R11.1 Self-healing (restart)	N	Y	N	N	Y[3]
R11.2 Auto Scaling	Y[1]	Y	N	N	Y[3]
R11.3 Load Balancing	Y[1]	Y	N	Y	Y
R11.4 Retry	Y[1]	N	N	N	N
R11.5 Circuit Breaker	N	N	Y	N	N
R11.6 Rate Limiter	Y[1]	N	Y	N	N
R11.7 Caching	Y[1]	N	N	N	N
R12 Compatibility					
R12.1 Cambio Scenarios	N	N	N	N	N
R12.2 Resirio	N	N	~	N	N
R12.3 TransVis	N	N	~	N	N

	SimuLizar	*DRACeo*	*MiSim*	*µqsim*	*MuSim*
⋮					
R9 CTK faultloads	N	~ [4]	~ [4]	N	N
R10 LIMBO support	Y[1]	N	N	N	Y[3]
R11 Resilience Features					
R11.1 Self-healing (restart)	N	Y	N	N	Y[3]
R11.2 Auto Scaling	Y[1]	Y	N	N	Y[3]
R11.3 Load Balancing	Y[1]	Y	N	Y	Y
R11.4 Retry	Y[1]	N	N	N	N
R11.5 Circuit Breaker	N	N	Y	N	N
R11.6 Rate Limiter	Y[1]	N	Y	N	N
R11.7 Caching	Y[1]	N	N	N	N
R12 Compatibility					
R12.1 Cambio Scenarios	N	N	N	N	N
⋮					

[2]

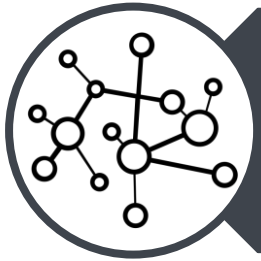
What does MiSim do?

What does MiSim do?



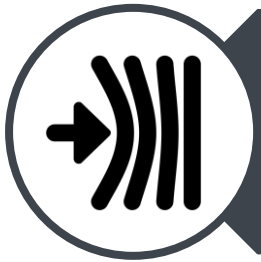
Architecture Description

Experiment Description



Service Oriented Architectures

- Message Traveling (Failures/Timeouts/Dependencies)
- Separation of Service Instances



Resilience Pattern Behavior

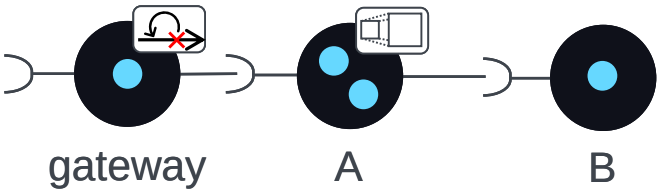
- Circuit Breaker
- Retry
- …

Chaos Injections

- Chaos Monkey (Instance/Service Failure)
- Network Delay

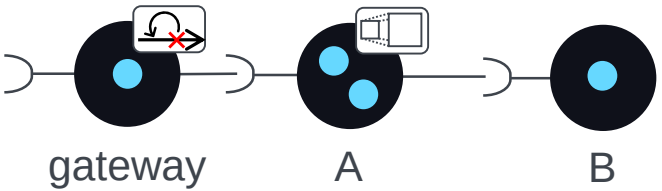
How does MiSim work?

How does MiSim work?
Architecture Description



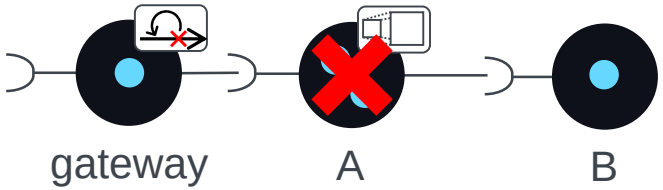

```
{
  "network_latency": ".02+.002-0.001",
  "microservices": [
    {
      "name": "gateway",
      "instances": 1,
      "capacity": 10000,
      "loadbalancer_strategy": "even",
      "operations": [
        {
          "name": "API_Endpoint",
          "demand": 1,
```

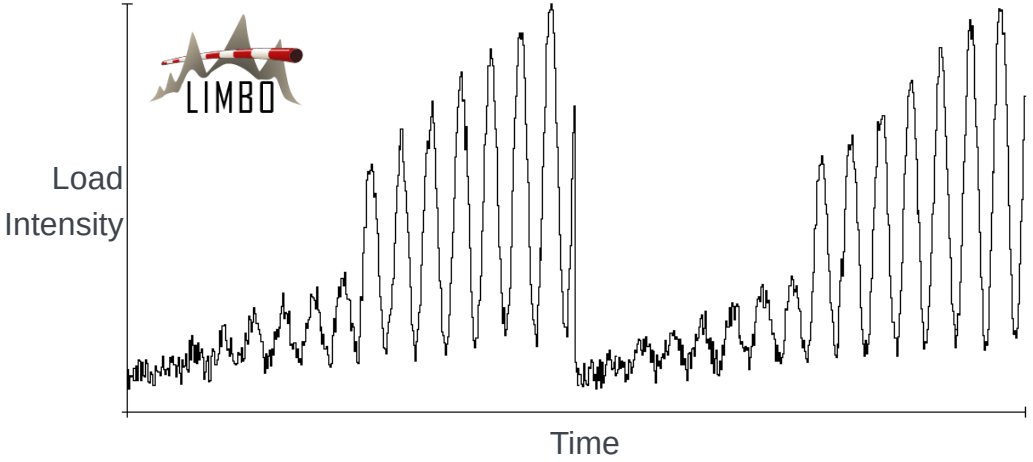
How does MiSim work?
Architecture Description



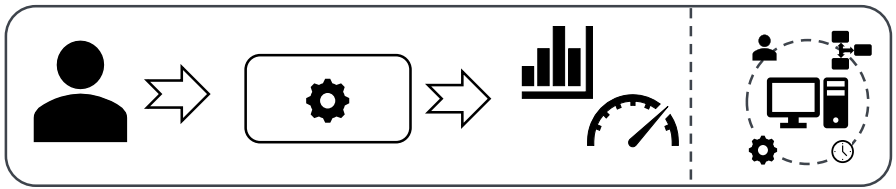

```
{
  "network_latency": ".02+.002-0.001",
  "microservices": [
    {
      "name": "gateway",
      "instances": 1,
      "capacity": 10000,
      "loadbalancer_strategy": "even",
      "operations": [...],
      "patterns": [
        {
          "type": "retry",
          "strategy": {
            "type": "linear"
          }
        }
      ]
    },
    {"name": "Service_A" ...},
    {"name": "Service_B" ...}
  ]
}
```

How does MiSim work?
Experiment Description

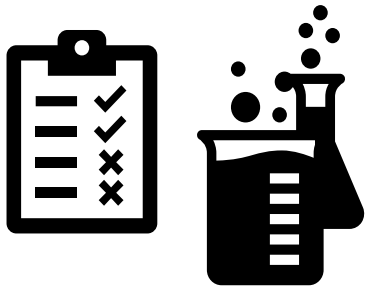




Architecture Tradeoff Analysis Method (ATAM) Scenario[3]



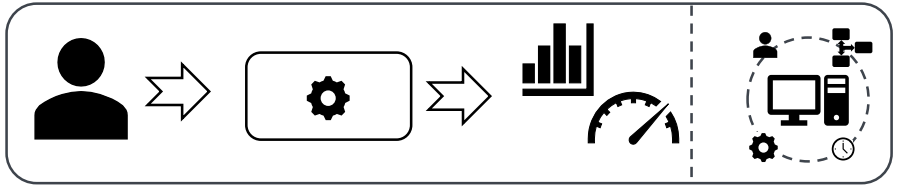
Resilience Test



How does MiSim work?
Experiment Description

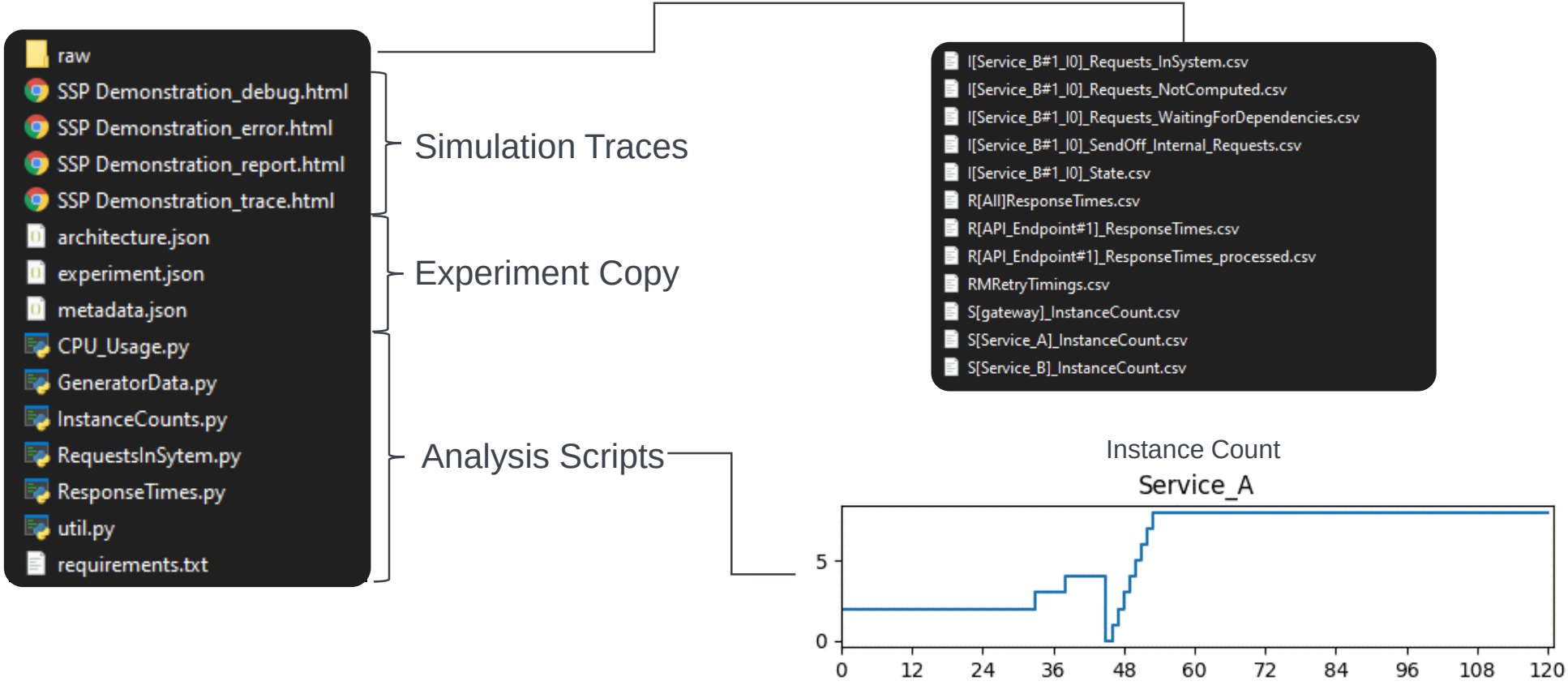
Architecture Tradeoff Analysis Method (ATAM)

Scenario

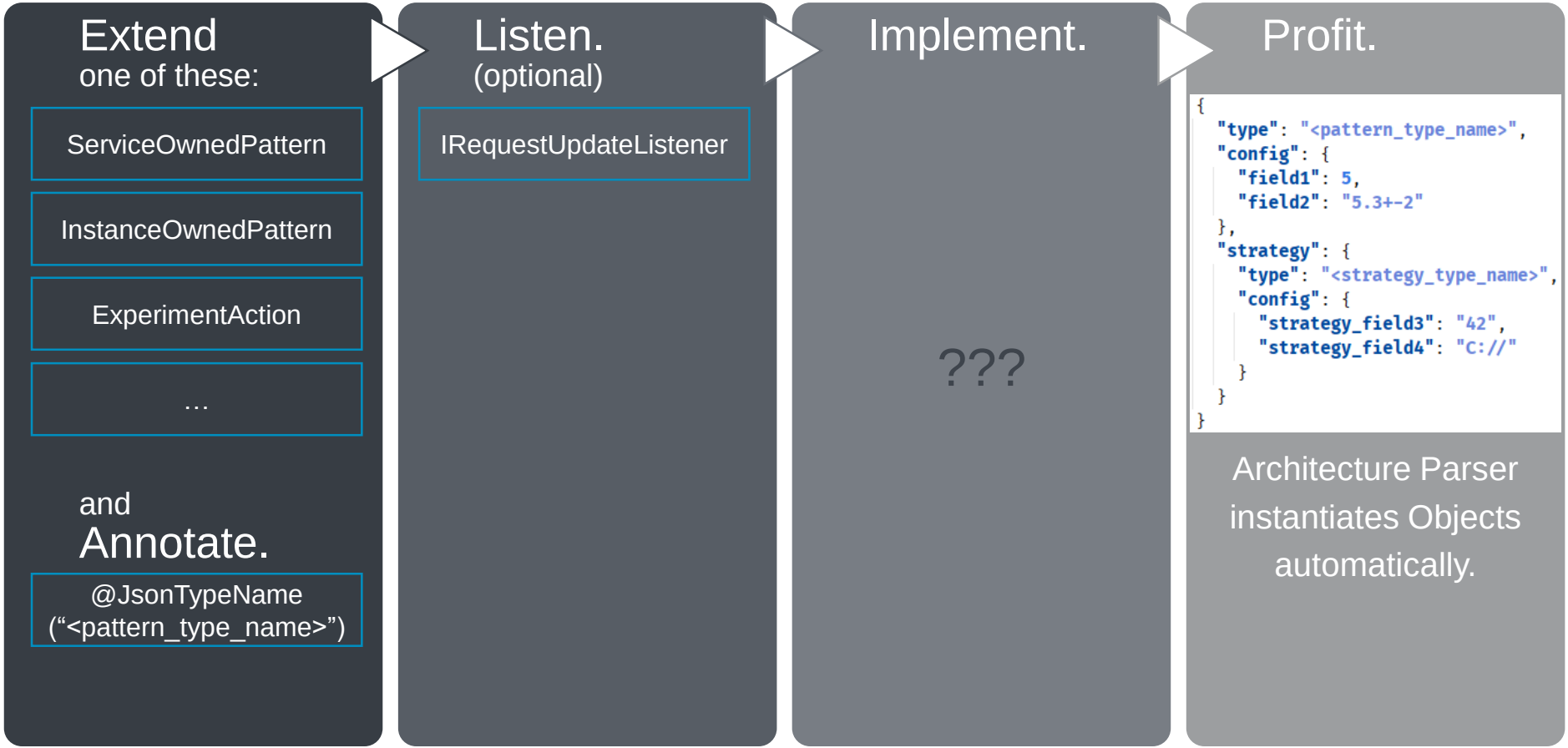


```
{
  "name": "SSP Demonstration",
  "description": "Shuts down 2 instances of Service_A at 45 STU.",
  "report_location": "SSP_Report/",
  "duration": 130,
  "seed": 42,
  "artifact": "Service_A",
  "component": "API_Endpoint",
  "stimulus": "LOAD src/test/resources/SSPExample/SSP_Limbo.csv AND KILL @45",
  ...
}
```

Output

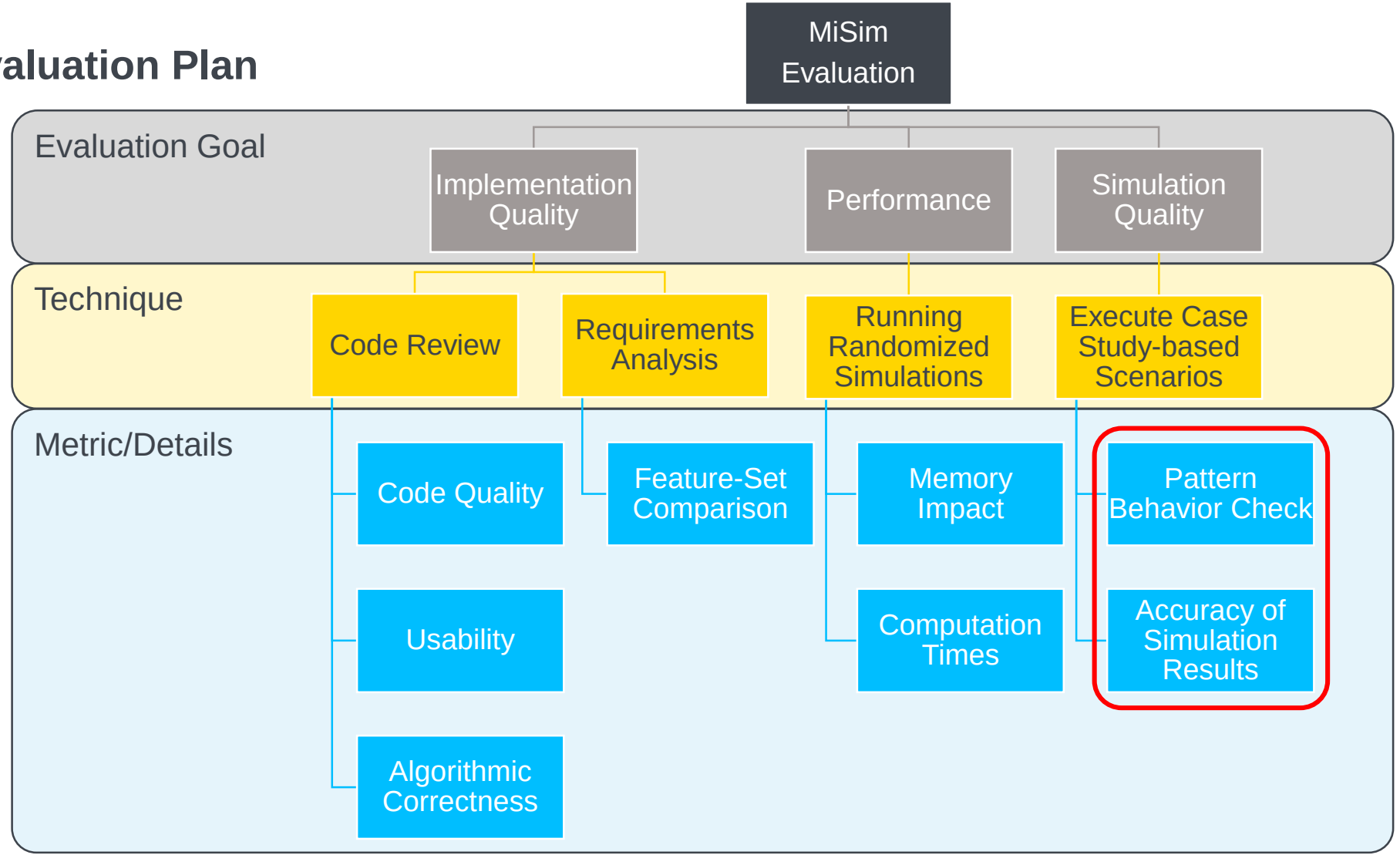

raw
SSP Demonstration_debug.html
SSP Demonstration_error.html
SSP Demonstration_report.html
SSP Demonstration_trace.html
architecture.json
experiment.json
metadata.json
CPU_Usage.py
GeneratorData.py
InstanceCounts.py
RequestsInSytem.py
ResponseTimes.py
util.py
requirements.txt
Simulation Traces
Experiment Copy
Analysis Scripts
I[Service_B#1_I0]_Requests_InSystem.csv
I[Service_B#1_I0]_Requests_NotComputed.csv
I[Service_B#1_I0]_Requests_WaitingForDependencies.csv
I[Service_B#1_I0]_SendOff_Internal_Requests.csv
I[Service_B#1_I0]_State.csv
R[All]ResponseTimes.csv
R[API_Endpoint#1]_ResponseTimes.csv
R[API_Endpoint#1]_ResponseTimes_processed.csv
RMRetryTimings.csv
S[gateway]_InstanceCount.csv
S[Service_A]_InstanceCount.csv
S[Service_B]_InstanceCount.csv
Instance Count
Service_A
5
0
0
12
24
36
48
60
72
84
96
108
120

Extensibility of MiSim



Evaluation and Conclusion

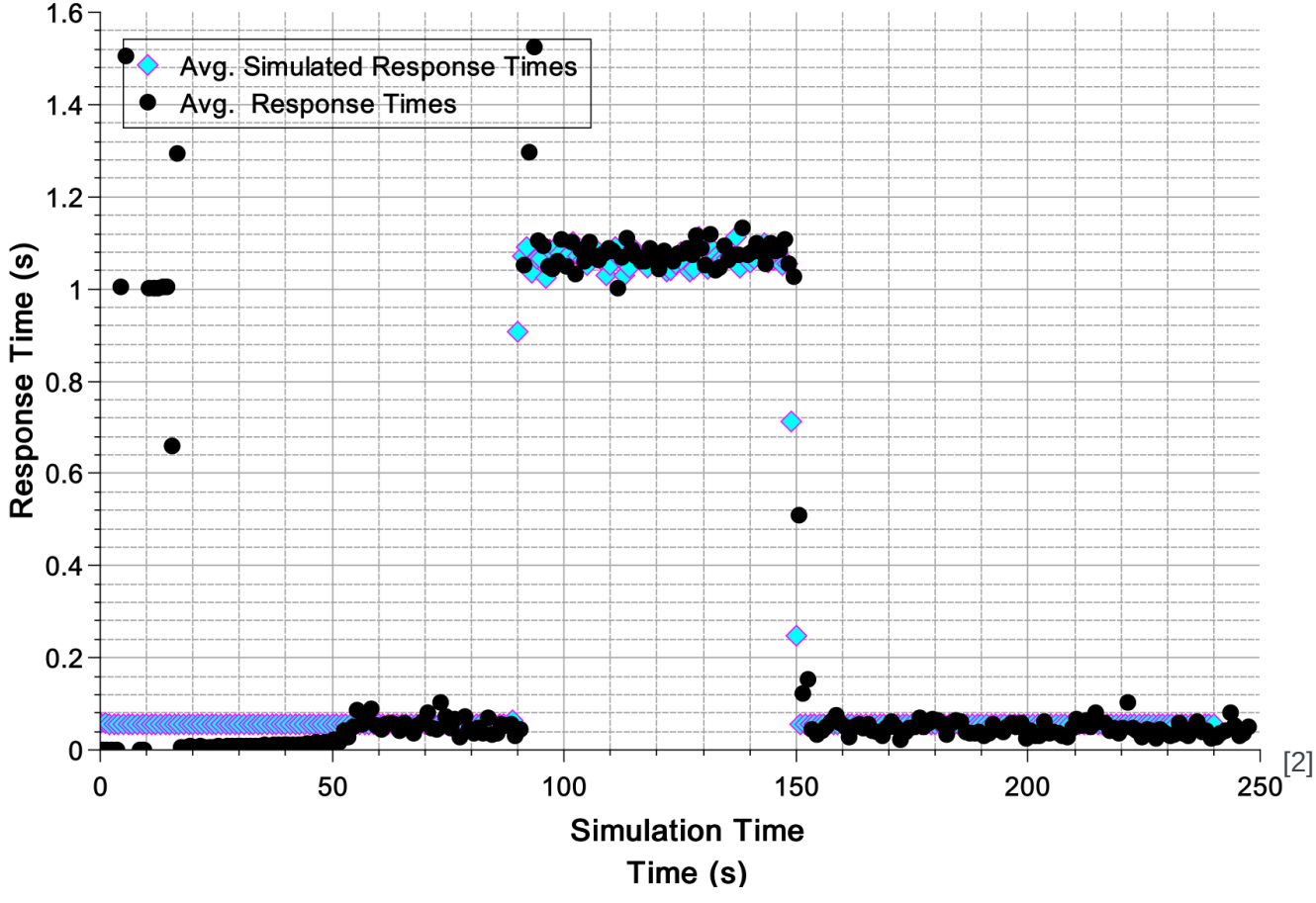
Evaluation Plan



[2]

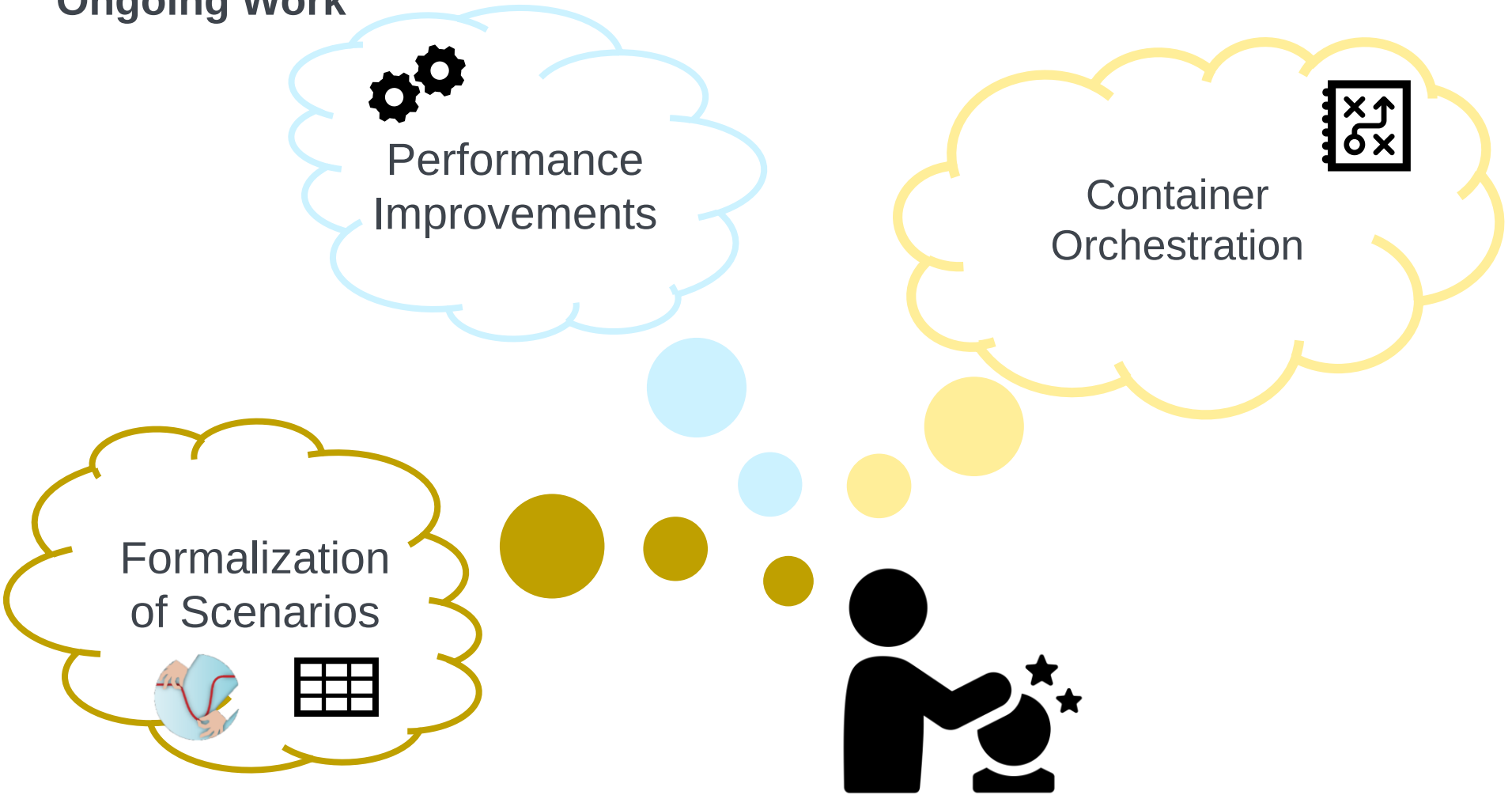
Evaluation Results Excerpt

Delay Injection



[2]

Ongoing Work



Links and References

Where to find us:

- MiSim 3.x onwards https://github.com/Cambio-Project/resilience-simulator
- Cambio-Project https://github.com/Cambio-Project

References:

[1] Sebastian Frank, Alireza Hakamian, Lion Wagner, Dominik Kesim, Jóakim von Kistowski, and André van Hoorn: *Scenario-based Resilience Evaluation and Improvement of Microservice Architectures: an Experience Report. FAACS@ECAS 2021*

[2] L. Wagner. *Simulating Scenario-based Chaos Experiments for Microservice Architectures.*

[3] L. Bass, and P. Clements, and R. Kazman. *Software Architecture in Practice.*

Additional Information

Extensibility in MiSim
Retry Example

```
{
  "type": "retry",
  "config": {
    "maxTries": 5
  },
  "strategy": {
    "type": "jittering",
    "config": {
      "baseBackoff": 0.02,
      "maxBackoff": 1,
      "base": 2
    }
  }
},
```

Expects Strategy Definition

Is registered as network listener.

```
@JsonTypeName("retry")
public class Retry extends StrategicInstanceOwnedPattern<IRetryStrategy> implements IRequestUpdateListener {

    @Expose
    private int maxTries = 5;
```

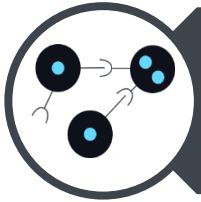
```
@JsonTypeName("jittering")
public class JitteringExponentialBackoffRetryStrategy extends ExponentialBackoffRetryStrategy {
```

What does MiSim do?

Input

Simulation

Output



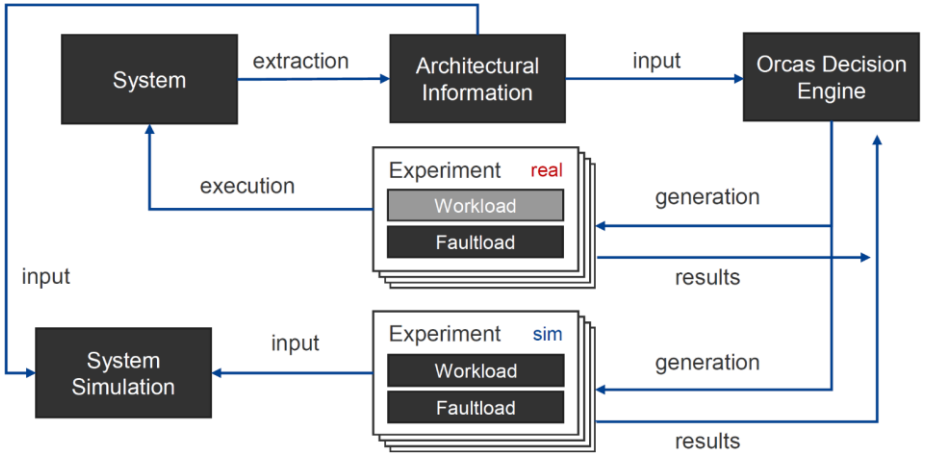
Architecture Description

Experiment Description

- Service Orientated Architectures
  - Message Traveling (Failure/Timeouts/Dependencies)
  - Separation of Service Instances
- Resilience Pattern Behavior
  - Circuit Breaker
  - Retry
- Chaos Injections
  - Chaos Monkey (Instance/Service Failure)
  - Network Delay
- (Multicore) CPU Scheduling
- LIMBO Load Models

- Simulation Trace
- Raw Metrics
  - Instance Counts
  - Response Times
  - CPU Usage
  - …
- Analysis Scripts
- Dependency Graph (soon)

Scientific Background and Motivation
Roots of MiSim

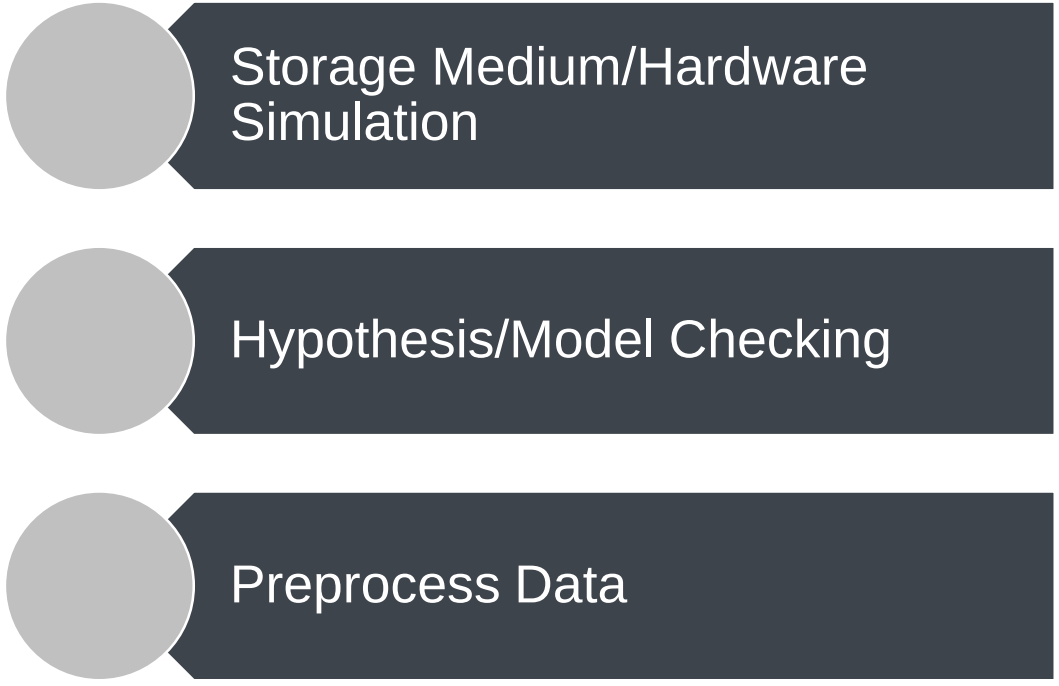


ORCAS Project:
Leverage Simulation for Efficient Resilience Evaluation

Cambio-Project:
Using ATAM Scenarios to describe Resilience requirements.

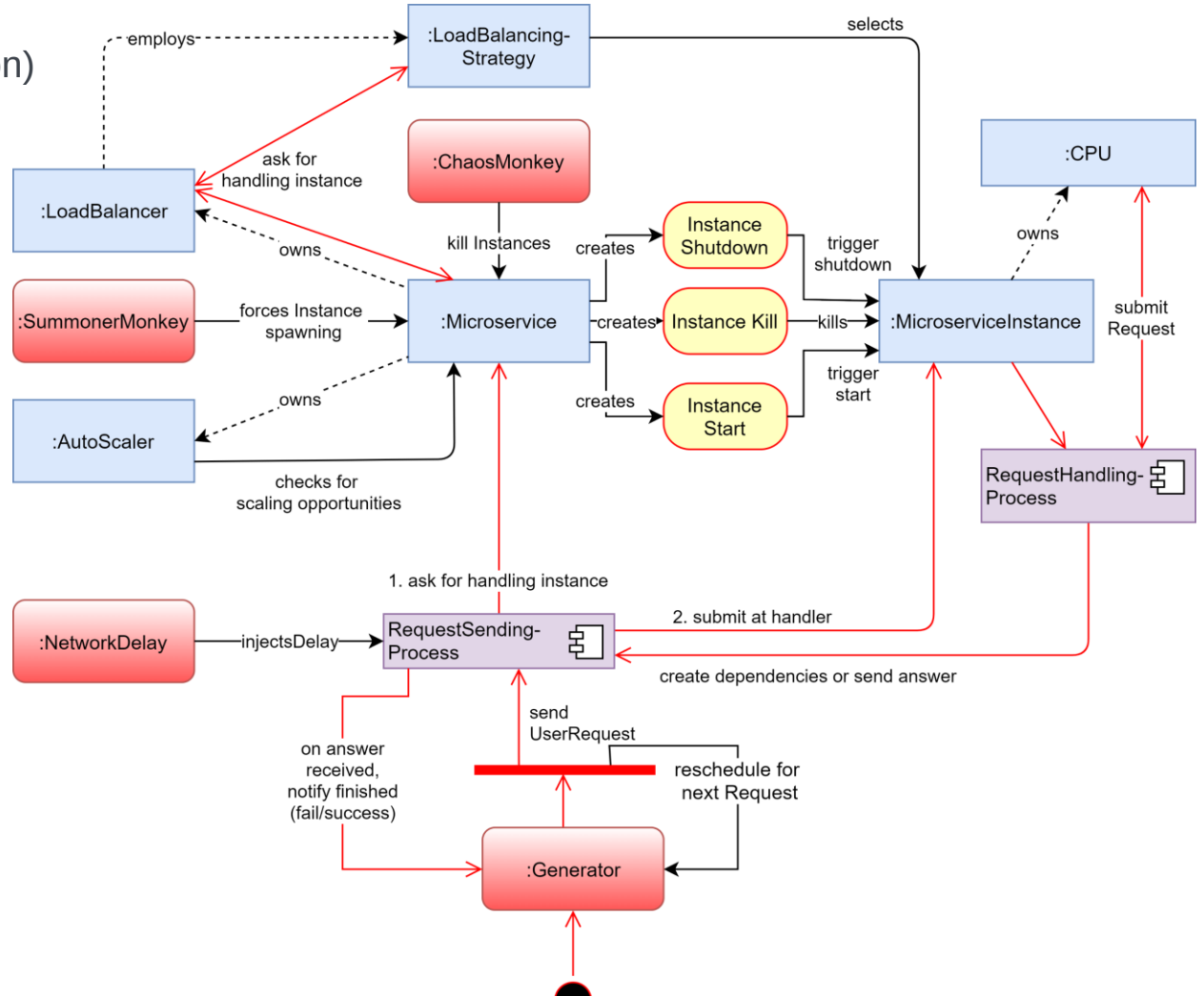
Requirements Specification
of 24 Requirements
for the Simulator

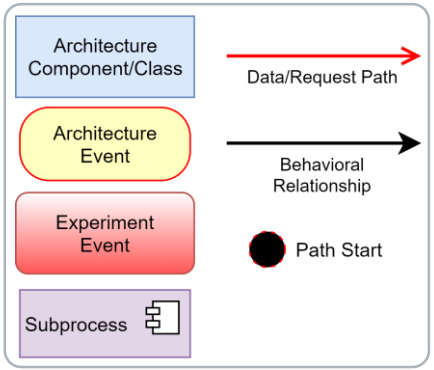
What does MiSim NOT do?



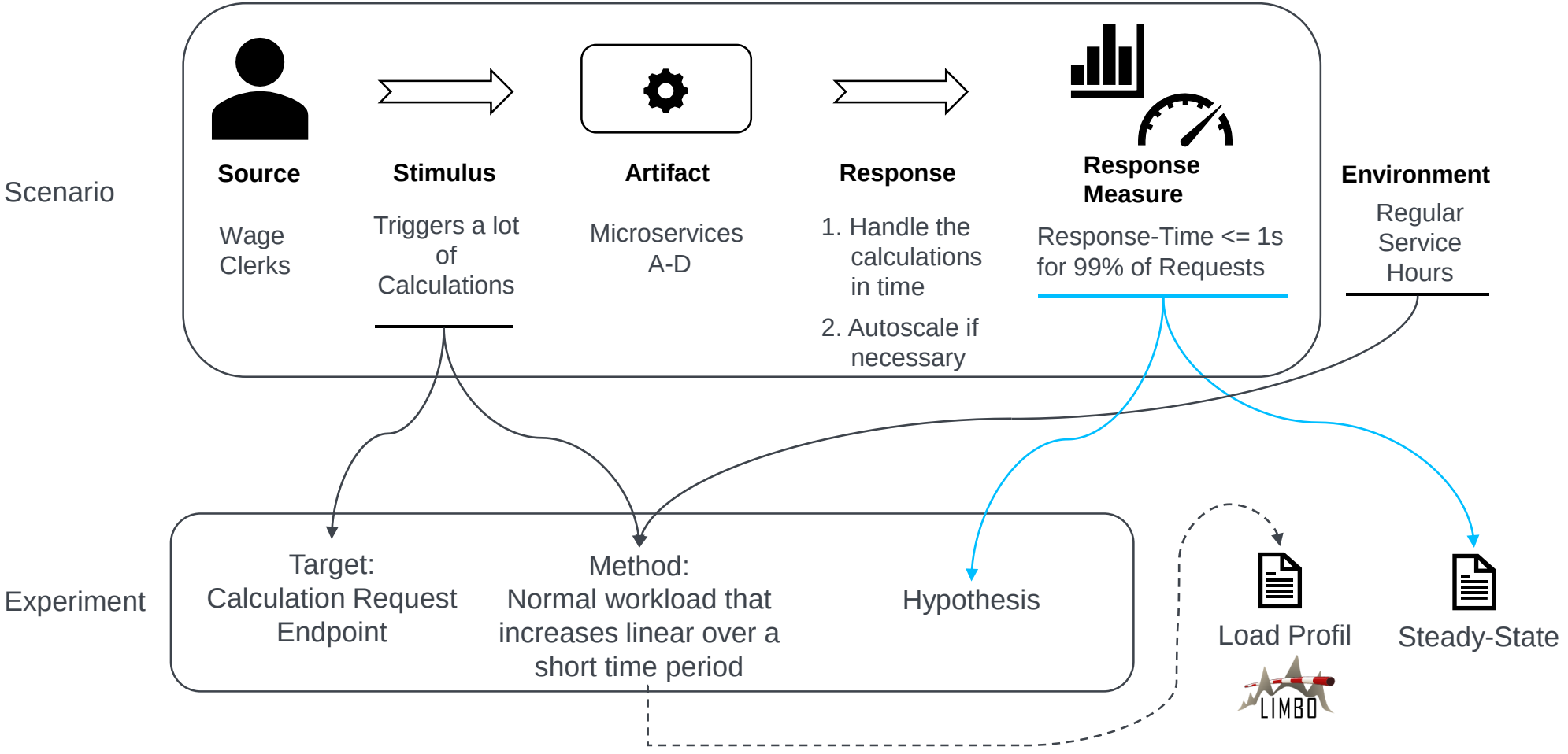
Architecture MiSim 3.0

(excluding parsing and data collection)





Software Scenarios — Example



Modification of MiSim 2.0

Architecture Improvements

- Separate unrelated Code
- Enforce Strategy Patterns
- Enforce Observer Patterns
- Network Simulation

Architecture Reengineering

Requirements Implementation

- Resilience Patterns
- Chaos Injections
- Data/Metric Collector
- Input/Scenario Parser

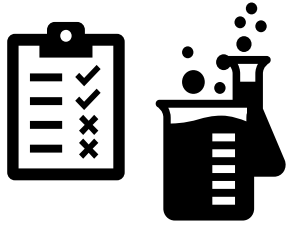
Feature-Set Comparison of MiSim 2.0 and 3.0

	MiSim 2.0	*MiSim* 3.0
R1 (uses DES)	Y	Y
R2 (headless mode)	Y	Y
R3 (lightweight)	Y	Y
R4 (parallel runs)	Y	Y
R5 (output metrics)		
R5.1 Response Times	Y	Y
R5.2 Error Rates	N	N
R5.3 Throughput	N	N
R5.4 Queue lengths	Y	Y
R5.5 Execution Traces	Y	Y
R6 (raw output)	Y	Y
R7 (common architecture desc.)	~ [1]	Y
R8 (case study faultloads)	~ [2]	Y
R9 (CTK faultloads)	~ [2]	Y
R10 (LIMBO support)	N	Y
R11 (resilience features)		
R11.1 Self-healing (restart)	N	N
R11.2 Auto Scaling	N	Y
R11.3 Load Balancing	N	Y
R11.4 Retry	N	Y
R11.5 Circuit Breaker	Y	Y
R11.6 Rate Limiter	Y	Y
R11.7 Caching	N	N
R12 (compatability)		
R12.1 Cambio Scenarios	N	Y
R12.2 Resirio [Zor21]	~	Y
R12.3 TransVis [Bec21]	~	~

- Most requirements are now satisfied
- R5.2, R5.3 and R11.1 are achievable with little additional manual effort
- R11.7 Caching was not implemented as simulated request currently don’t contain content

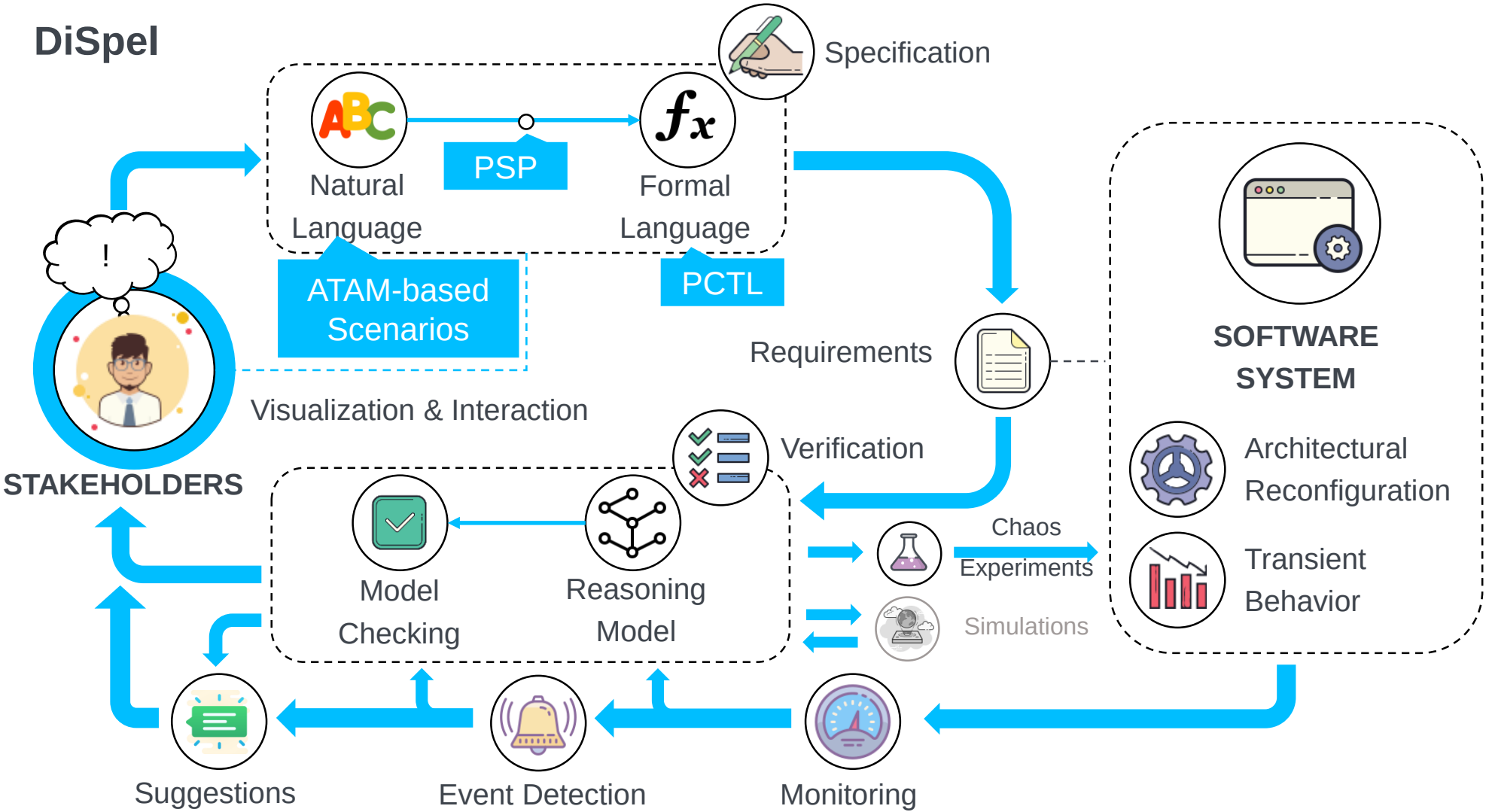
How does MiSim work?
Experiment Description

Resilience Test



```
{
  "name": "SSP Demonstration",
  "description": "Shuts down 2 instances of Service_A at 60 STU.",
  "report_location": "/SSP_Report/",
  "duration": 120,
  "seed": 42,
  "artifact": "Service_A",
  "component": "ALL ENDPOINTS",
  "stimulus": "LOAD src/test/resources/SSPExample/SSP_Limbo.csv AND KILL 2@60",
  ...
}
```

```
{
  "simulation_metadata": {
    "name": "SSP Demonstration",
    "description": "Shuts down 2 instances of Service_A at 60 STU.",
    "report_location": "SSP_Report/",
    "duration": 120,
    "seed": 42
  },
  "generators": [
    {
      "type": "limbo",
      "config": {
        "arrival_time": 0,
        "model": "src/test/resources/SSPExample/SSP_Limbo.csv",
        "target_operation": "gateway.PING"
      }
    }
  ],
  "named_monkey": {
    "type": "chaos_monkey",
    "config": {
      "microservice": "gateway",
      "instances": "2"
    }
  }
}
```


DiSpel
Specification
Natural Language
PSP
Formal Language
ATAM-based Scenarios
PCTL
!
STAKEHOLDERS
Visualization & Interaction
Requirements
SOFTWARE SYSTEM
Verification
Architectural Reconfiguration
Model Checking
Reasoning Model
Chaos Experiments
Transient Behavior
Simulations
Suggestions
Event Detection
Monitoring