



# Hyperledger Mentorship Project Presentation

November 2021

# Hyperledger Caliper – Declarative specification of workloads

## › Introduction

- › **Name:** Aastha Bist
- › **Location:** Dehradun, Uttarakhand, India
- › **University:** Graphic Era University, Dehradun
- › **Mentor:** Attila Klenik (Budapest University of Technology and Economics, Hungary)
- › **Hyperledger Project:** Hyperledger Caliper

# Declarative specification of workloads

## › **Project Description:**

- › Hyperledger Caliper is a framework used to benchmark blockchain systems.
- › Workload modules are a Caliper component used for defining parameters passed to every transaction.
- › Currently, workload modules have to be implemented in JavaScript.
- › **Project focus area:** Developing a low-code solution to define workload modules in Hyperledger Caliper.

# Declarative specification of workloads

## JavaScript declaration

```
/**
 * Assemble TXs for creating new marbles.
 * @return {Promise<TxStatus[]>}
 */
async submitTransaction() {
  this.txIndex++;

  const marbleName = `${this.roundArguments.marblePrefix}_${this.roundIndex}_${this.workerIndex}_${this.txIndex}`;
  let marbleColor = this.colors[this.txIndex % this.colors.length];
  let marbleSize = (((this.txIndex % 10) + 1) * 10).toString(); // [10, 100]
  let marbleOwner = this.owners[this.txIndex % this.owners.length];

  let args = {
    contractId: this.txIndex % 2 === 0 ? 'mymarbles' : 'yourmarbles',
    contractFunction: 'initMarble',
    contractArguments: [marbleName, marbleColor, marbleSize, marbleOwner],
    invokerIdentity: 'client0.org1.example.com',
    timeout: 5,
    readOnly: false
  };
};
```

# Declarative specification of workloads

## YAML declaration

```
test:
  workers:
  rounds:
  - label:
    txNumber:
    rateControl:
    workload:
      module: declarative
      arguments:
        myParameter1: param1
        myParameter2: 42
      behavior:
        contractSelection:
          uniform:
          weightedUniform:
            weights: [1, 1, 2]
          roundRobin:
        contracts:
          - name: contract1
            functionSelection:
              uniform:
              weightedUniform:
                weights: [1, 1, 2]
              roundRobin:
            functions:
              - name: function1
                parameters:
                  - name: param1
                    type: uniform_random
                options:
                  min: 10
                  max: 100
```



# Declarative specification of workloads

## › Project Objectives:

- › Design a YAML schema for defining contract and parameter selection
- › Create modular implementation for supporting YAML schema
- › Integrate and test the implementation in Caliper's environment
- › Migrate an existing workload to the new schema
- › Provide user and developer level documentation

# Declarative specification of workloads

## › Project Deliverables:

- › Deliverable 1: Building out the initial schema
- › Deliverable 2: Create Value Provider classes for the schema
- › Deliverable 3: Integrate and test Value Providers
- › Deliverable 4: Integrate the schema through higher level classes
- › Deliverable 5: Migrating an existing workload to the schema
- › Deliverable 6: Creating documentation

# Declarative specification of workloads

## › **Project Execution & Accomplishments:**

- › Surveyed other similar tools to observe how declarative configuration style works
  - Common trend: code-based configurations
- › Created a YAML schema for declaring parameters
- › Implemented, integrated and tested value providers.
- › Ongoing task: Integrating the schema through higher level classes
- › Upcoming task: Testing an existing configuration using the schema, writing user and developer documentation



# Declarative specification of workloads

## › **Recommendations for future work:**

- › Adding more complex value providers.
- › Adjusting the existing schema implementing classes to support different types of SUTs.
- › Scheduling for contracts and functions.

# Declarative specification of workloads

## › **Project Output or Results:**

- › Project proposal - <https://bit.ly/caliper-project-proposal>
- › Project Plan - <http://bit.ly/caliper-project-page>
- › Declarative Workload Directory - <https://bit.ly/declarative-workload>
- › GitHub Pull Requests- <https://bit.ly/gh-project-pr>

# Declarative specification of workloads

## › Insights Gained:

- › The solution won't be perfect on the first try. Improving things iteratively.
- › Testing your code thoroughly.
- › Thinking outside the box, sometimes.



A large audience is seated in a conference hall, facing a stage where a speaker is visible. The scene is overlaid with a blue geometric pattern of lines and dots. The text "THANK YOU!" is prominently displayed in the center.

**THANK YOU!**