

# Load Testing and Performance Evaluation of the Theia Online IDE

Bachelor Thesis Intermediate Presentation

Author: Tobias Klingenberg

Supervisor: Prof. Dr. Stephan Krusche

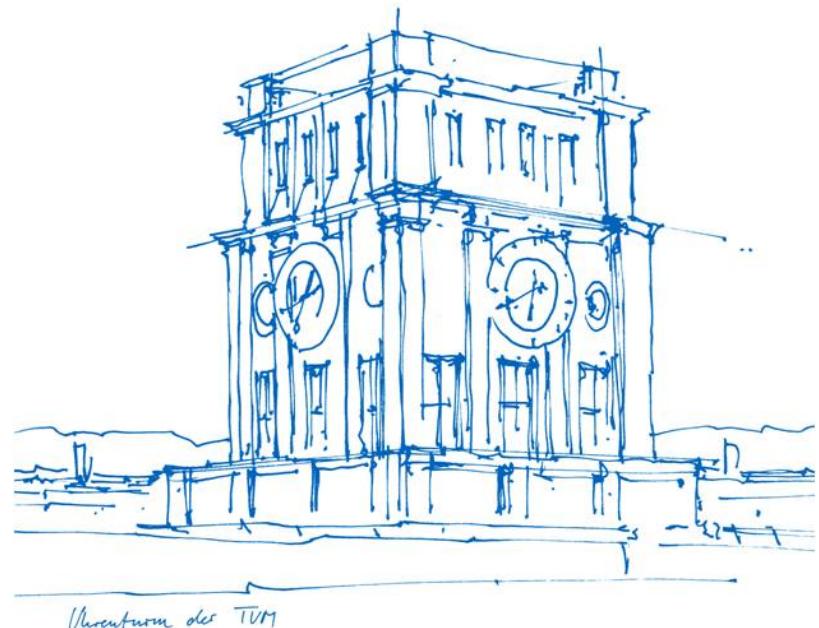
Advisor: Matthias Linhuber

Technische Universität München

TUM School of Computation, Information and Technology

Group for Applied Education Technologies

Garching, 23. October 2025



# Motivation

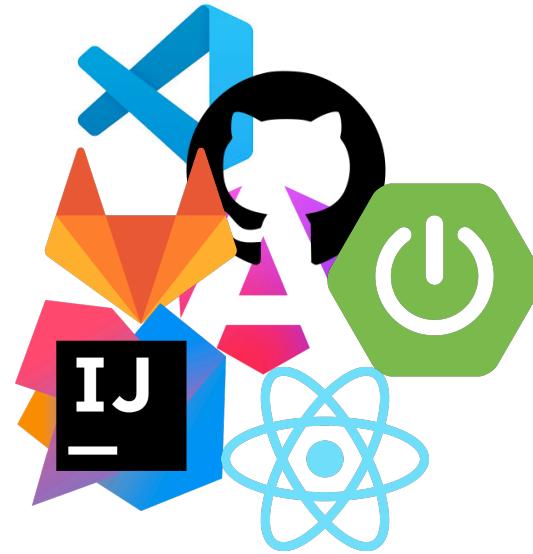


Steve

# Motivation



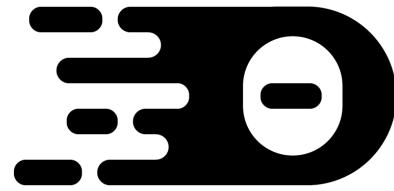
Steve



# Motivation



Steve



Theia

# Motivation



Steve  
Student



Hector  
Instructor



Jennifer  
Developer

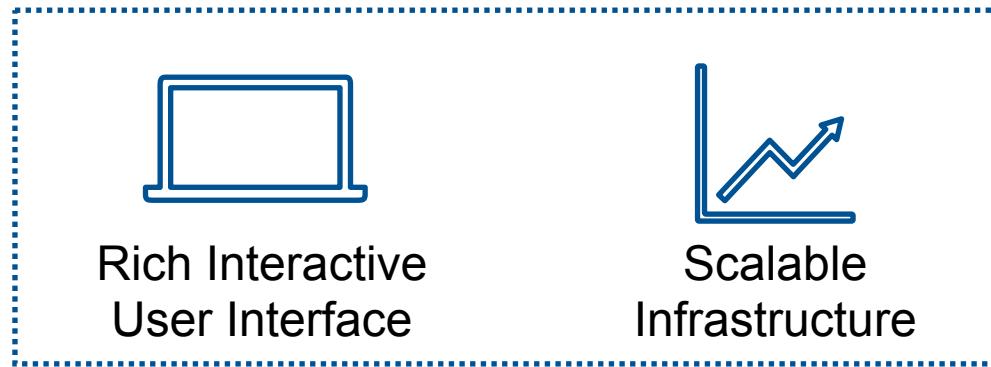


Edwin  
Admin

# Problem



Steve

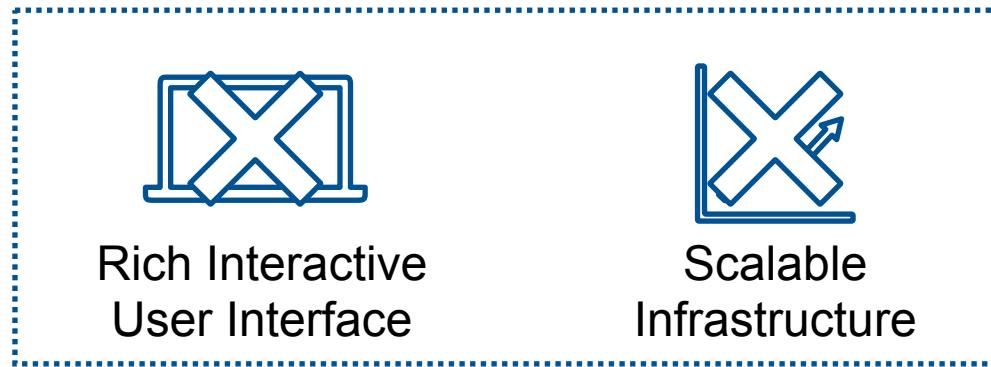


Edwin

# Problem



Steve



Not currently tested



Edwin

# Objectives

1. Developing a **Testing Suite** that tests the usability and **functional correctness** of Theia Cloud
2. Extending the Test Suite with a **Load Testing** framework that allows developers to evaluate the performance of .
3. Creating a **MCP Testing** script that allows a LLM to interact with Theia Cloud for testing purposes



# Goals

: Functional Tests 

Testing **functionality**  
using E2E Tests

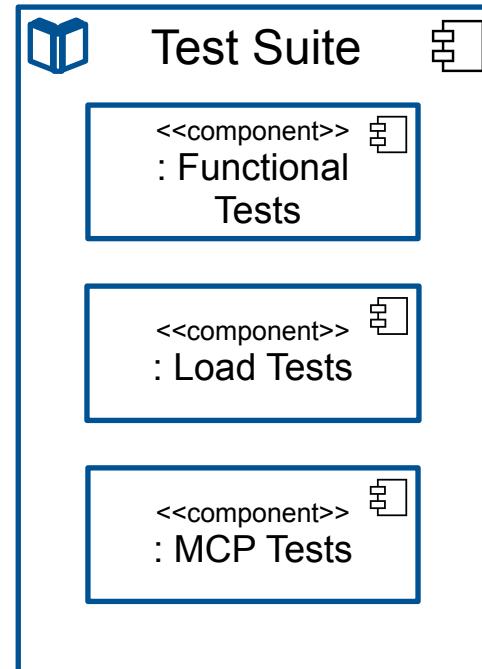
: Load Tests 

Testing **scalability**  
under load

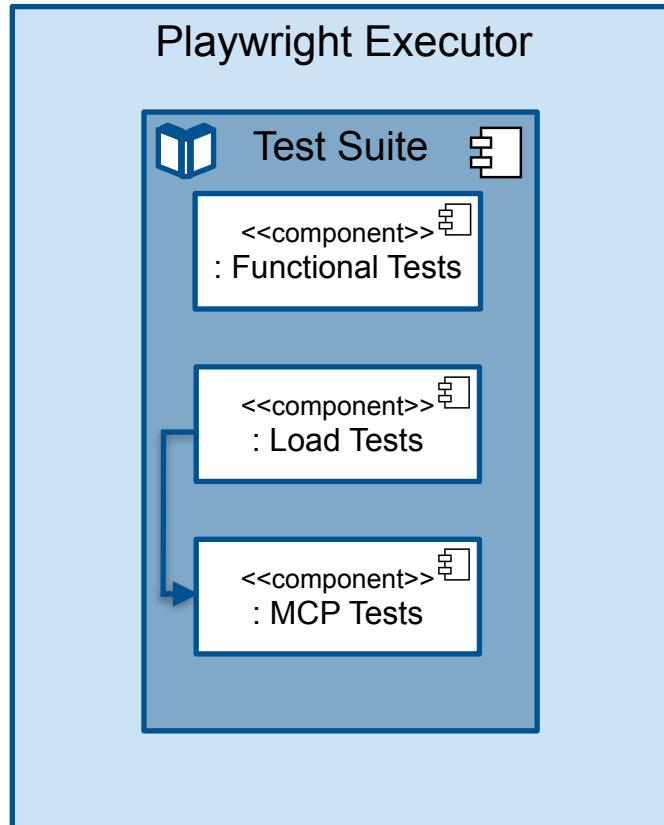
: MCP Tests 

PoF: Testing using  
**LLMs**

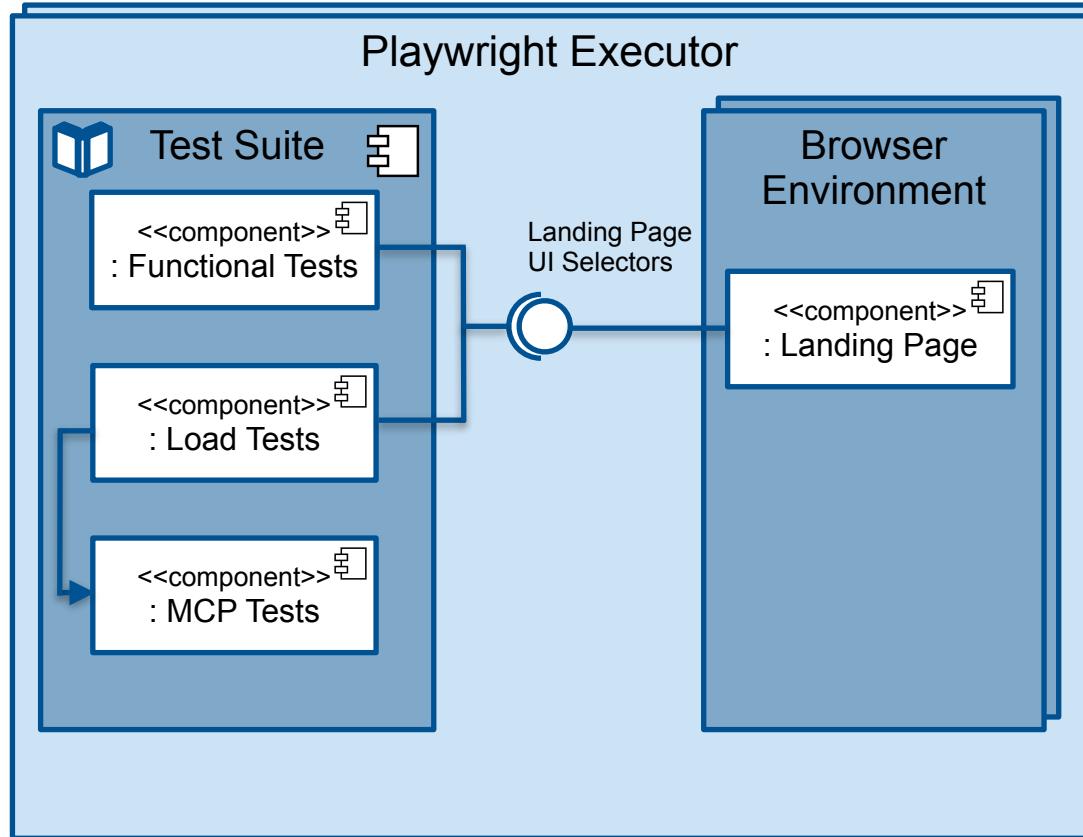
# Top Level Design



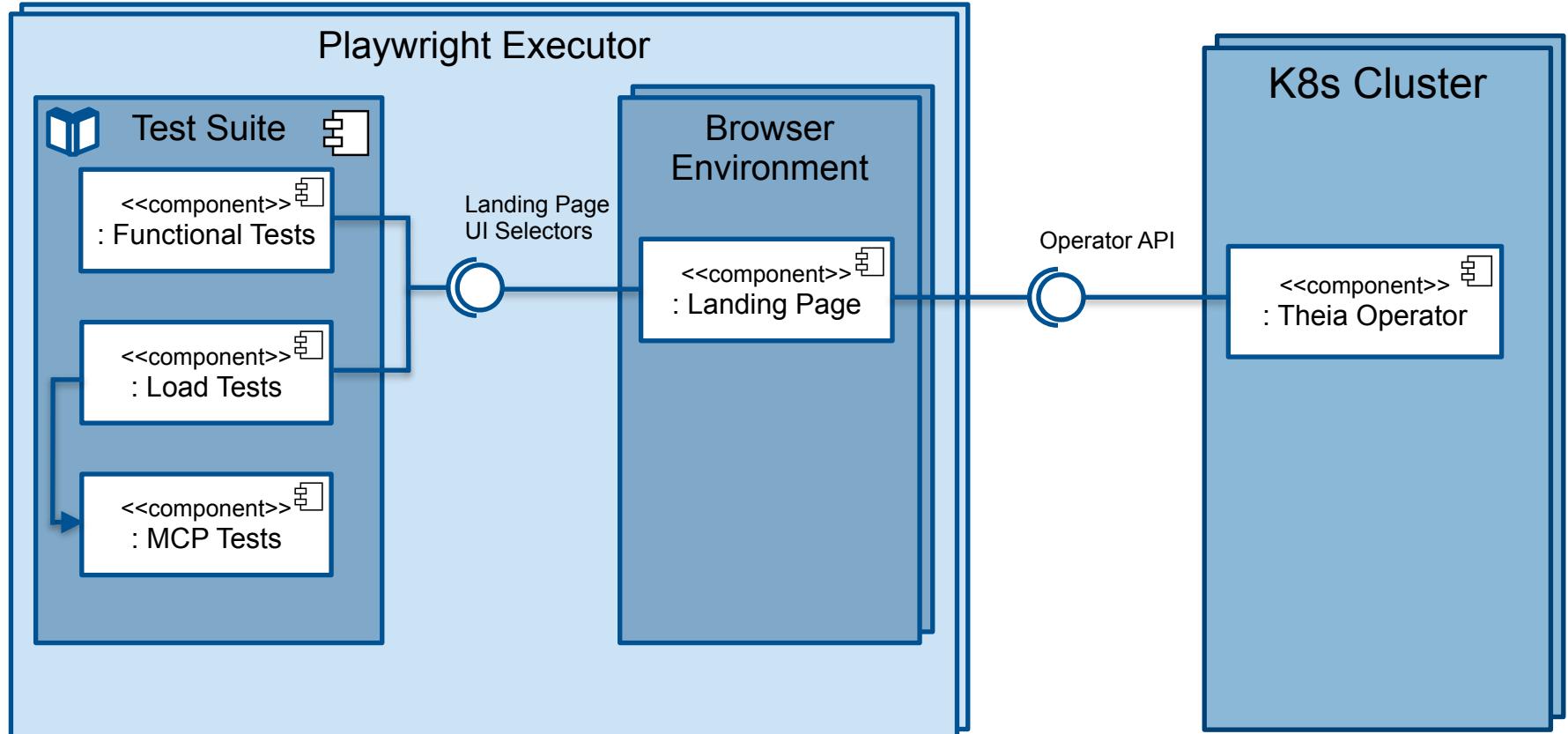
# Subsystem Decomposition



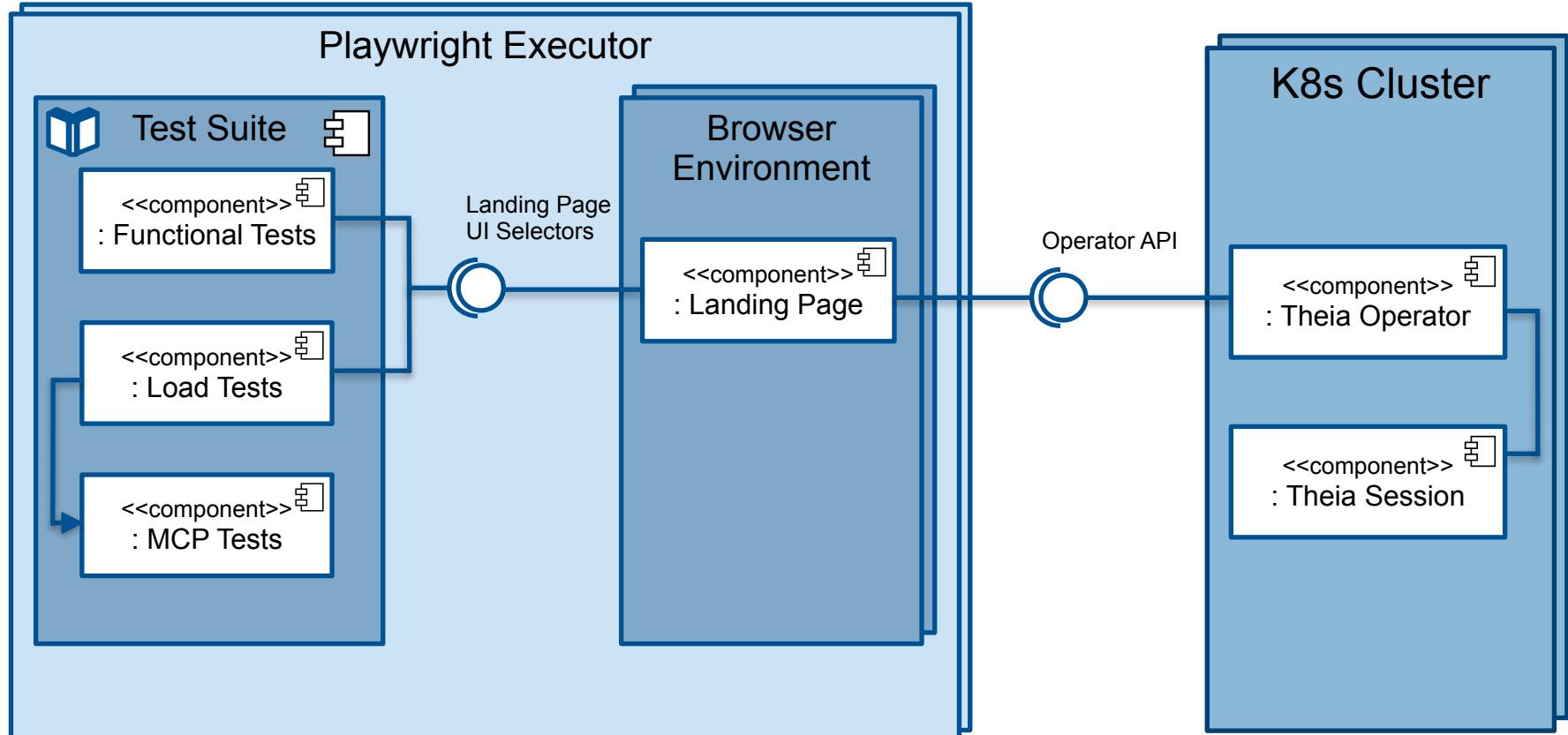
# Subsystem Decomposition



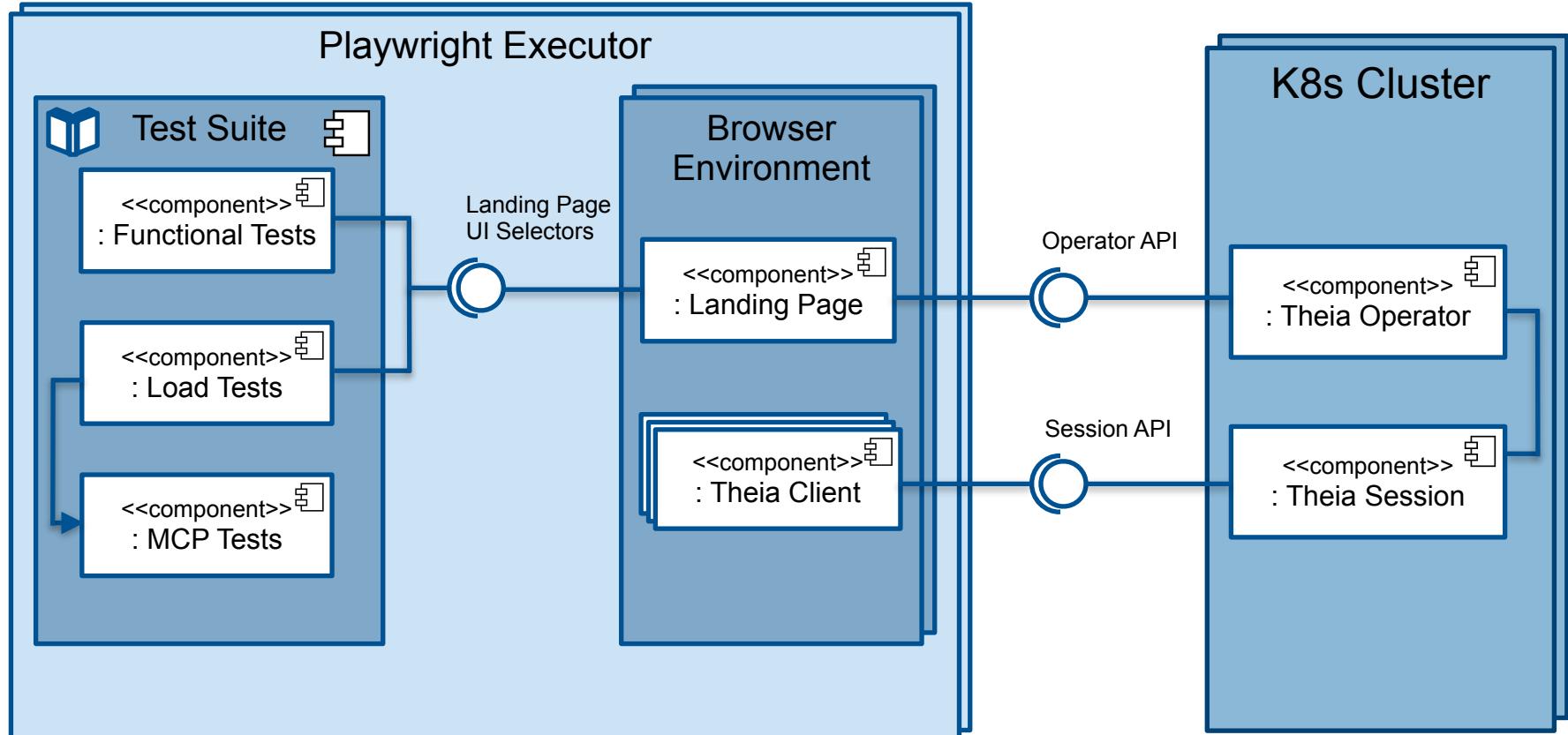
# Subsystem Decomposition



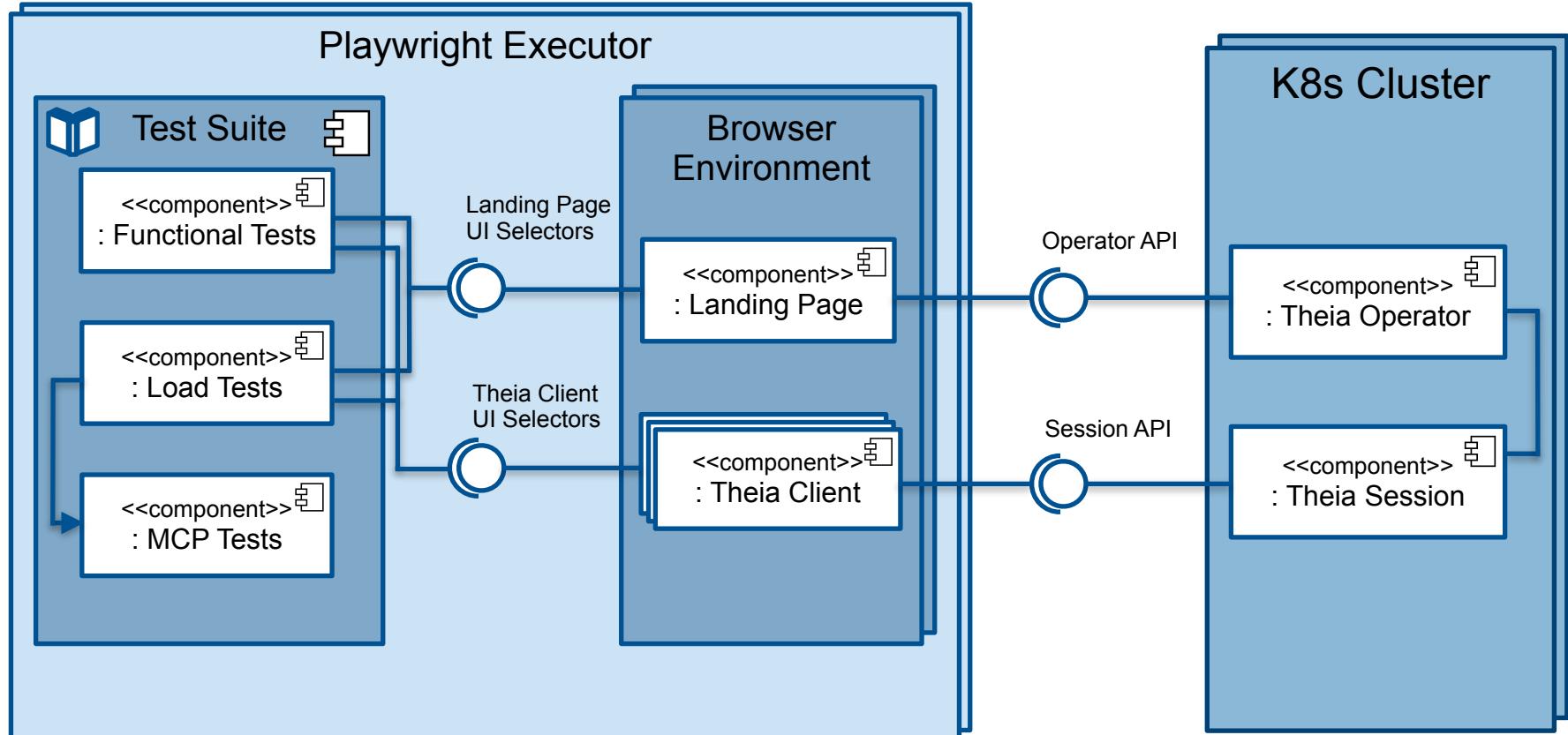
# Subsystem Decomposition



# Subsystem Decomposition



# Subsystem Decomposition



# Functionality Tests

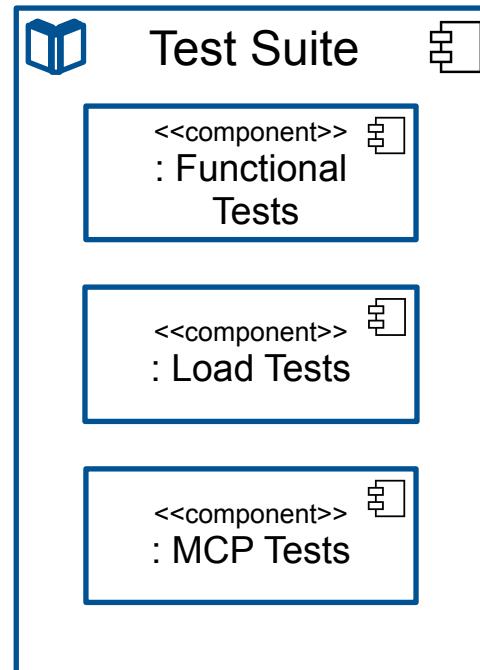


# Functionality Tests

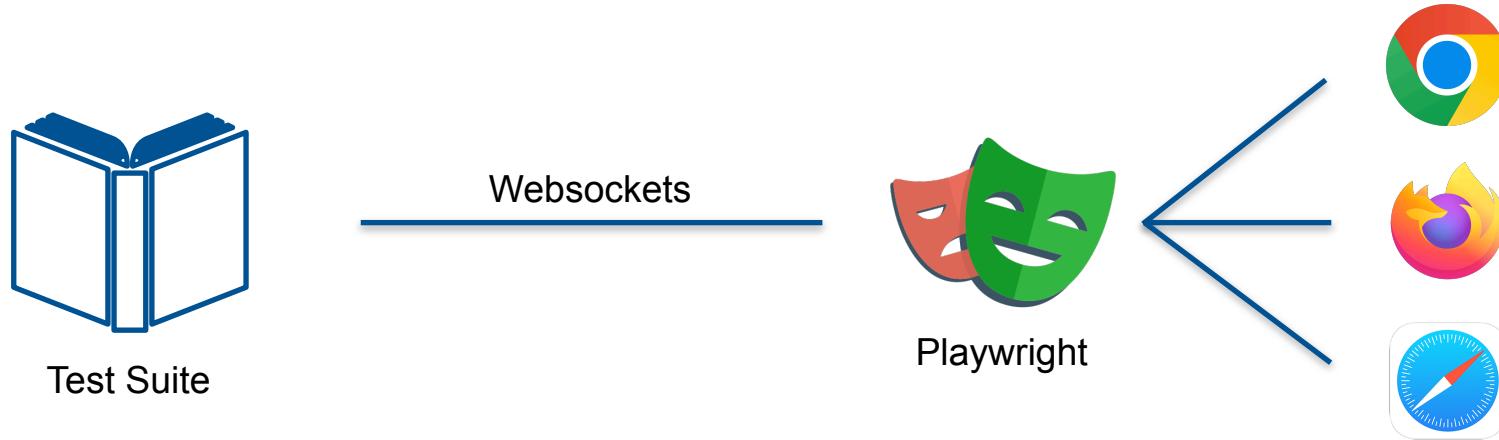


# Functionality Tests

# Functional Tests



# E2E Tests with Playwright





Playwright



Artemis

Landing Page



Keycloak

The screenshot shows a Java development environment with the following interface elements:

- Top Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Sidebar:** ARTEMIS: SCORPIO, theia, Submit Exercise, Points (No Graded Result), Submission due (No Due date).
- Section Headers:** Sorting with the Strategy Pattern, Part 1: Sorting, Part 2: Strategy Pattern.
- Text Content:** Instructions for sorting algorithms and strategy patterns.
- Code Editor:** A large window displaying the `MergeSort.java` file. The code implements a merge sort algorithm for a list of dates. A mouse cursor is positioned over the `merge` method.
- Bottom Status Bar:** main, 0 0 0 0 0 0, Collaborate, Java Ready, Line 30, Col 33, LF, UTF-8, Spaces: 2, Java, and several icons.

```
theialocaltesttheia-ge85kof > src > test > MergeSort.java > MergeSort > merge(List<Date>, int, int, int)
1
2
3 import java.util.*;
4
5 public class MergeSort {
6
7 /**
8 * Wrapper method for the real MergeSort algorithm.
9 *
10 * @param input the List of Dates to be sorted
11 */
12 public void performSort(List<Date> input) {
13     mergesort(input, 0, input.size() - 1);
14 }
15
16 private void mergesort(List<Date> input, int low, int high) {
17     if (high - low < 1) {
18         return;
19     }
20     int mid = (low + high) / 2;
21     mergesort(input, low, mid);
22     mergesort(input, mid + 1, high);
23     merge(input, low, mid, high);
24 }
25
26 private void merge(List<Date> input, int low, int middle, int high) {
27
28     Date[] temp = new Date[high - low + 1];
29     int leftIndex = low;
30     int rightIndex = middle + 1;
31     int wholeIndex = 0;
32     while (leftIndex <= middle && rightIndex <= high) {
33         if (input.get(leftIndex).compareTo(input.get(rightIndex)) <= 0) {
34             temp[wholeIndex] = input.get(leftIndex++);
35         } else {
36             temp[wholeIndex] = input.get(rightIndex++);
37         }
38         wholeIndex++;
39     }
40     if (leftIndex <= middle && rightIndex > high) {
41         while (leftIndex <= middle) {
42             temp[wholeIndex++] = input.get(leftIndex++);
43         }
44     } else {
45         while (rightIndex <= high) {
46             temp[wholeIndex++] = input.get(rightIndex++);
47         }
48     }
49     for (wholeIndex = 0; wholeIndex < temp.length; wholeIndex++) {
50         input.set(wholeIndex + low, temp[wholeIndex]);
51     }
52 }
```

The screenshot shows a Java code editor interface. The main window displays the code for `MergeSort.java`. The code implements a merge sort algorithm for a list of dates. The editor has a dark theme with syntax highlighting for Java keywords and comments.

```
a-ge85kof > src > test > MergeSort.java > MergeSort > merge(List<Date>, int, int, int)
...
java.util.*;
class MergeSort {
    ...
    // per method for the real MergeSort algorithm.
    ...
    // am input the List of Dates to be sorted
    void performSort(List<Date> input) {
        esort(input, low:0, input.size() - 1);
        ...
        e void mergesort(List<Date> input, int low, int high) {
            high - low < 1) {
                return;
            mid = (low + high) / 2;
            esort(input, low, mid);
            esort(input, mid + 1, high);
            e(input, low, mid, high);
        }
        ...
        private void merge(List<Date> input, int low, int middle, int high) {
            Date[] temp = new Date[high - low + 1];
            int leftIndex = low;
            int rightIndex = middle + 1;
            int wholeIndex = 0;
            while (leftIndex <= middle && rightIndex <= high) {
                if (input.get(leftIndex).compareTo(input.get(rightIndex)) <= 0) {
                    temp[wholeIndex] = input.get(leftIndex++);
                } else {
                    temp[wholeIndex] = input.get(rightIndex++);
                }
                wholeIndex++;
            }
            if (leftIndex <= middle && rightIndex > high) {
                while (leftIndex <= middle) {
                    temp[wholeIndex++] = input.get(leftIndex++);
                }
            } else {
                while (rightIndex <= high) {
                    temp[wholeIndex++] = input.get(rightIndex++);
                }
            }
            for (wholeIndex = 0; wholeIndex < temp.length; wholeIndex++) {
                input.set(wholeIndex + low, temp[wholeIndex]);
            }
        }
    }
}
```

The left sidebar contains a file tree and a list of tasks:

- New Text File Alt+N
- New File... Cmd+Alt+N
- New Folder... Alt+Shift+N
- Open... Cmd+Alt+O
- Open Workspace from File... Cmd+Alt+W
- Open Recent Workspace... Cmd+Alt+R
- Add Folder to Workspace...
- Save Workspace As...
- Save Cmd+S
- Save All Cmd+Alt+S
- Save As...
- ✓ Auto Save
- Upload Files...
- Preferences >
- Close Editor Alt+W
- Close Workspace

**Test Results**

Implement the method `performSort(List<Date>)` in the class `BubbleSort`. Make sure to follow the Bubble Sort algorithm exactly.

**2. ① Implement Merge Sort [No](#)**

**Test Results**

Implement the method `performSort(List<Date>)` in the class `MergeSort`. Make sure to follow the Merge Sort algorithm exactly.

**Part 2: Strategy Pattern**

We want the application to apply different algorithms for sorting a `List` of `Date` objects. Use the strategy pattern to select the right sorting algorithm at runtime.

You have the following tasks:

File Edit Selection View Go Run Terminal Help

Java Ready

Ln 30, Col 33 LF UTF-8 Spaces: 2 {} Java

The screenshot shows a Java code editor interface with a dark theme. A context menu is open over the code area, with the 'File' menu highlighted by a red box. The 'File' menu contains the following items:

- New Text File Alt+N
- New File... Cmd+Alt+N
- New Folder...
- New Window Alt+Shift+N
- Open... Cmd+Alt+O
- Open Workspace from File... Cmd+Alt+W
- Open Recent Workspace... Cmd+Alt+R
- Add Folder to Workspace...
- Save Workspace As...
- Save Cmd+S
- Save All Cmd+Alt+S
- Save As...
- ✓ Auto Save
- Upload Files...
- Preferences >
- Close Editor Alt+W
- Close Workspace

The main code editor window displays a Java file named `MergeSort.java` with the following content:

```
a-ge85kof > src > test > MergeSort.java > MergeSort > merge(List<Date>, int, int, int)
...
class MergeSort {
    ...
    //per method for the real MergeSort algorithm.
    ...
    // am input the List of Dates to be sorted
    ...
    void performSort(List<Date> input) {
        esort(input, low, 0, input.size() - 1);
    }

    void mergesort(List<Date> input, int low, int high) {
        if (high - low < 1) {
            return;
        }
        int mid = (low + high) / 2;
        esort(input, low, mid);
        esort(input, mid + 1, high);
        merge(input, low, mid, high);
    }

    private void merge(List<Date> input, int low, int middle, int high) {
        ...
    }
}

int wholeIndex = 0;
while (leftIndex <= middle && rightIndex <= high) {
    if (input.get(leftIndex).compareTo(input.get(rightIndex)) <= 0) {
        temp[wholeIndex] = input.get(leftIndex++);
    } else {
        temp[wholeIndex] = input.get(rightIndex++);
    }
    wholeIndex++;
}
if (leftIndex <= middle && rightIndex > high) {
    while (leftIndex <= middle) {
        temp[wholeIndex++] = input.get(leftIndex++);
    }
} else {
    while (rightIndex <= high) {
        temp[wholeIndex++] = input.get(rightIndex++);
    }
}
for (wholeIndex = 0; wholeIndex < temp.length; wholeIndex++) {
    input.set(wholeIndex + low, temp[wholeIndex]);
}
```

Below the code editor, there is a 'Test Results' section with the following text:

Implement the method  
performSort  
in the class  
sure to follow the Merge Sort  
algorithm exactly.

2. ① Implement Merge Sort [No](#)

Test Results  
Implement the method  
performSort(List<Date>)  
in the class [MergeSort](#). Make  
sure to follow the Merge Sort  
algorithm exactly.

**Part 2: Strategy Pattern**

We want the application to apply different algorithms for sorting a [List of Date](#) objects. Use the strategy pattern to select the right sorting algorithm at runtime.

You have the following tasks:

File Edit Selection View Go Run Terminal Help

Java Ready

Ln 30, Col 33 LF UTF-8 Spaces: 2 {} Java

# Functionality in Test



**Editor**  
8 Tests



**Search**  
5 Tests



**Terminal**  
7 Tests



**VCS**  
4 Tests



**Programming Language**  
24 Tests



**Artemis Integration**  
1 Test

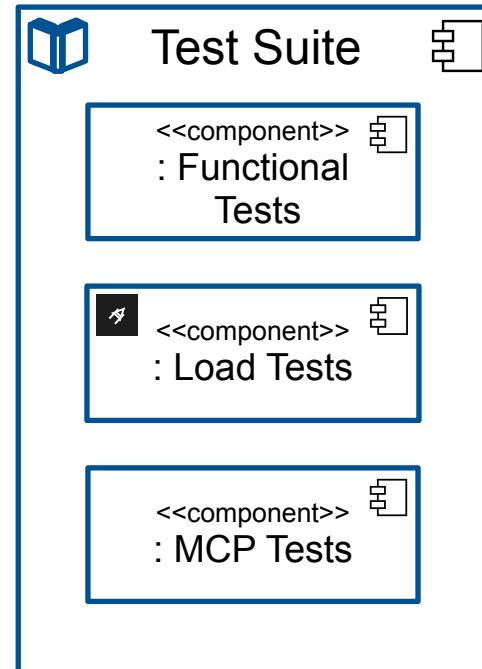
# Load Testing

# Load Testing

# Load Testing

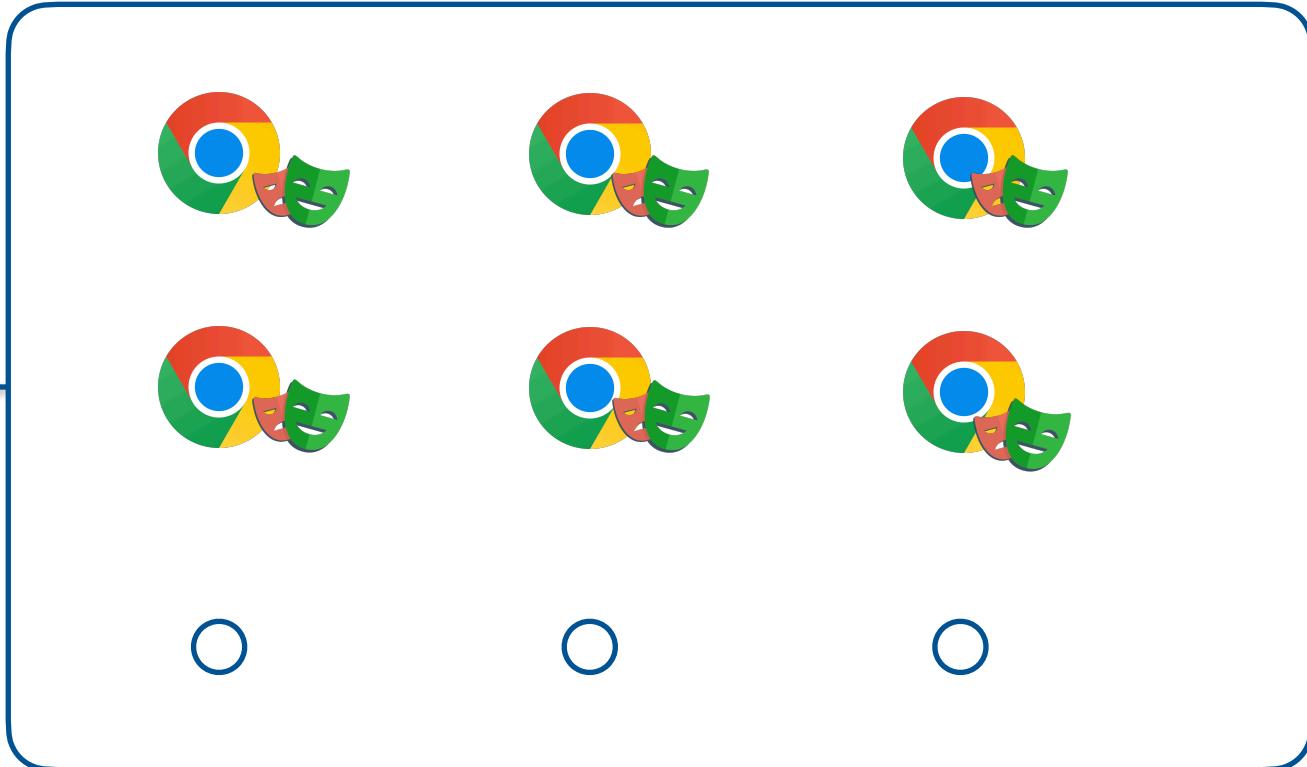


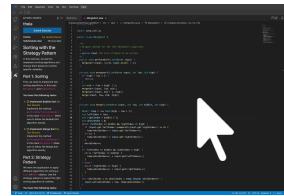
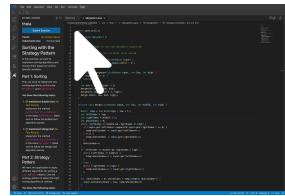
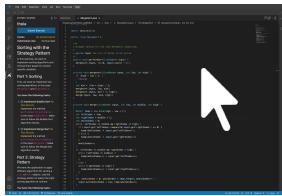
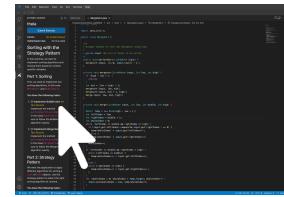
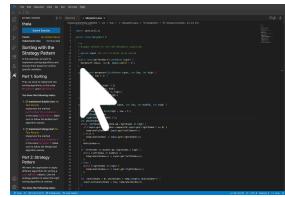
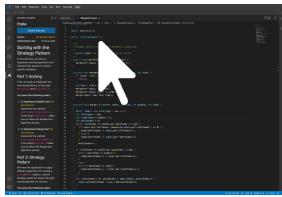
# Load Testing

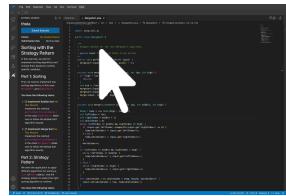
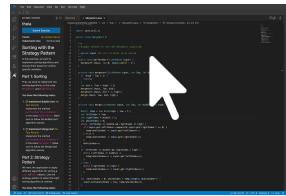
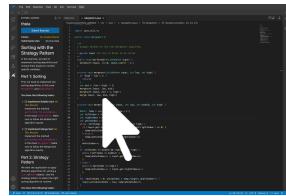
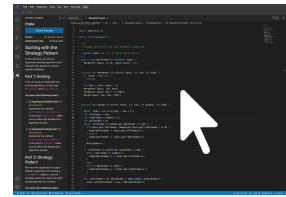
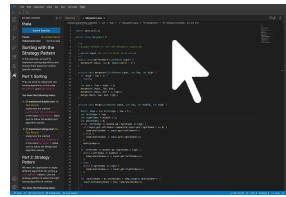
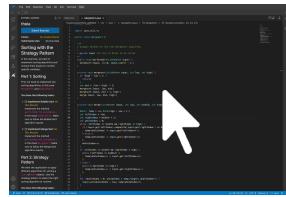




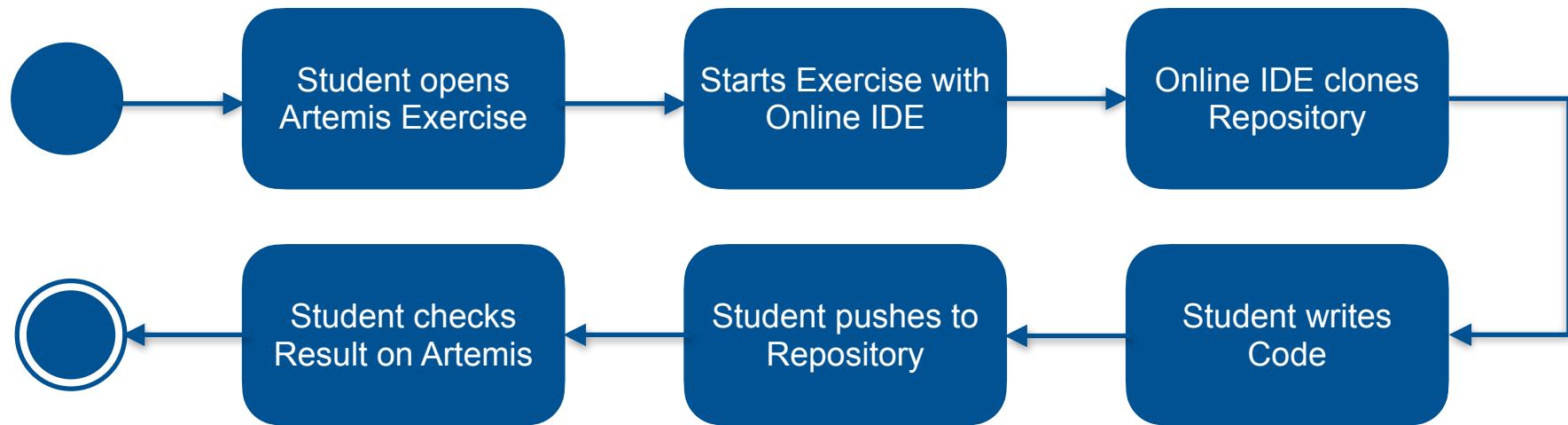
Artillery.io







# Example Workflow





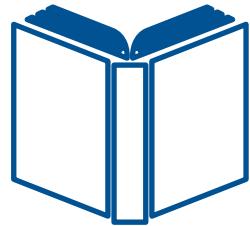
Steve

!=



Static Tests

# LLM Testing



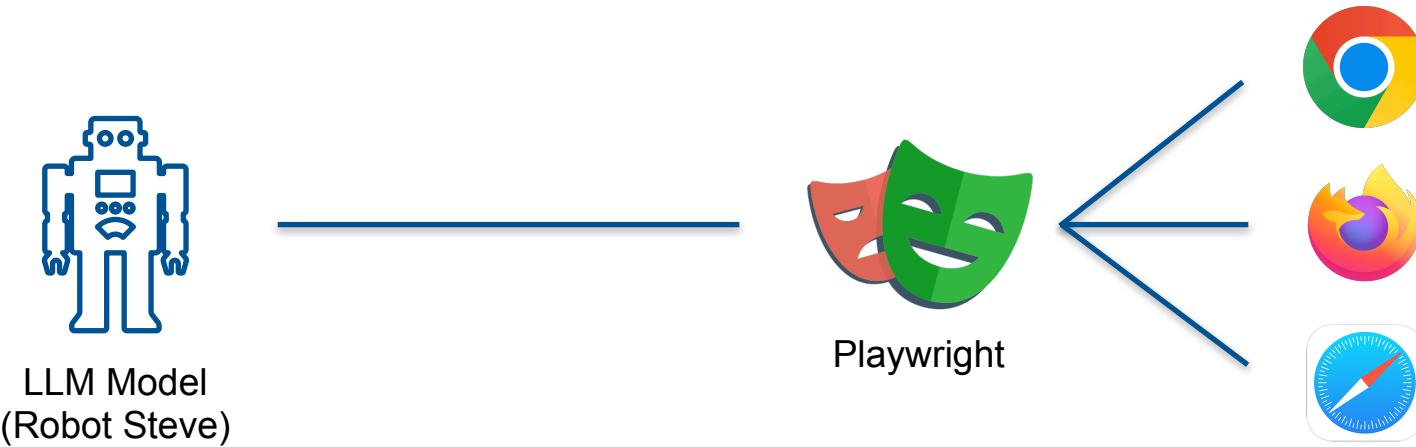
Test Suite  
(Steve)



Playwright



# LLM Testing



The screenshot shows a terminal window and a file explorer interface side-by-side.

**Terminal View:**

- Output of the command `mcp > virtualstudent.prompt`:

```
1 Imagine you are a student at a university
2 The course management provided an online platform for exams. Your task now is to explore the
3 IDE using the Playwright MCP Server, interact with the IDE.
4
5 You can find the Landing Page to start
6 https://theia.artemis.cit.tum.de
7
8 Also note the following:
9 - You will only get this single prompt
10 - Don't use any unnecessary wait states
11 - Never navigate to another page/link
```
- Output of the command `volta-shim +> volta switch 22-tests-integrate-llmcp-integration-optional`:

```
git switch 22-tests-integrate-llmcp-integration-optional
Switched to branch '22-tests-integrate-llmcp-integration-optional'
Your branch is up to date with 'origin/22-tests-integrate-llmcp-integration-optional'.
```
- Output of the command `npx tsx mcp/mcp-setup && npx @playwright/mcp@latest --config=mcp/local/config.json --storage-state .auth/mcp\_auth.json`:

```
theia-scale-tests on 22-tests-integrate-llmcp-integration-optional [§]
> Logging in and saving session...
Listening on http://localhost:8931
Put this in your client config:
{
  "mcpServers": [
    "playwright": {
      "url": "http://localhost:8931/mcp"
    }
  ]
}
For legacy SSE transport support, you can use the /sse endpoint instead.
[OK to generate a command]
```

**File Explorer View:**

- Current folder: `virtualstudent.prompt`
- Content of `virtualstudent.prompt` file:

```
1 Imagine you are a student at a university
2 The course management provided an online platform for exams. Your task now is to explore the
3 IDE using the Playwright MCP Server, interact with the IDE.
4
5 You can find the Landing Page to start
6 https://theia.artemis.cit.tum.de
7
8 Also note the following:
9 - You will only get this single prompt
10 - Don't use any unnecessary wait states
11 - Never navigate to another page/link
```
- Project structure:
  - THEIA-SCALE-TESTS
    - .auth
    - .cursor
    - .github/workflows
      - artemis-integration-tests.yml
      - artillery-tests.yml
      - functional-tests.yml
      - scalable-tests.yml
    - xscode
      - mcp.json
      - settings.json
    - fixtures
    - utils
    - example-texts
      - lorem-ipsum.ts
    - templates
    - exercise\_solutions
      - BubbleSort.txt
      - Client.txt
      - Context.txt
      - MergeSort.txt
      - Policy.txt
      - SortStrategy.txt
    - programming\_languages
      - course.json
      - exercise.json
      - commands.ts
      - constants.ts
      - global-setup.ts
      - global-teardown.ts
      - theia.fixture.ts
    - mcp
      - automated
      - local
      - mcp-setup.ts
      - README.md
    - node\_modules
    - pages
      - artemis
        - ArtemisPage.ts
        - ExercisePage.ts
    - ide
      - custom-pom
    - theia-pom
    - IDEPage.ts

virtualstudent.prompt – theia-scale-tests

Use the prompt given in the file:

I'll follow the virtual student prompt to explore the Theia Online IDE thoroughly! Let me be a curious Computer Science student and try everything the IDE has to offer.

To-dos 8

- Navigate to Theia landing page and start an IDE session
- Explore Text editor features - syntax highlighting, autocomplete, themes
- Test file creation, editing, saving, and file management
- Explore terminal capabilities - commands, navigation, multiple terminals
- Try building and running code in different languages
- Explore and change IDE preferences and settings
- Test debugging, extensions, collaboration, and other advanced features
- Explore different panels, views, and interface customization

Perfect! As a curious CS student, let me start by navigating to the Theia landing page and exploring everything this online IDE has to offer!

> Called browser.Navigate ✓

Excellent! I'm already logged in as a student ([theia-prod@linhuber.org](https://theia-prod@linhuber.org)) and

virtualstudent.prompt

```
mcp > virtualstudent.prompt
1 Imagine you are a student at a
2 The course management provided
3 exams. Your task now is to exp
4 using the terminal, running an
5 IDE using the Playwright MCP S
6 interact with the IDE.
7
8 You can find the Landing Page
9 https://theia.artemis.cit.tum.de
10
11 Also note the following:
12 - You will only get this singl
13 - Don't use any unnecessary wai
14 - Never navigate to another pa
```

... volta-shim +~ + ~ ^ x

```
Switched to branch '22-tests-integrate
at: 'l1mmpc-integration-optional'
Your branch is up to date with 'origin
/22-tests-integrate/l1mmpc-integration
-optional'.
```

theia-scale-tests on 9 22-tests-integrat
ate:l1mmpc-integration-optional [\\$]
> npx tsx mcp/mcp-setup &
> npx @playwright/acp@latest --config
=mcp/local/config.json --storage-state
\_auth/mcp\_auth.json
https://theia.artemis.cit.tum.de
Logging in and saving session...
Listening on http://localhost:8931
Put this in your client config:
{
 "mcpServers": {
 "playwright": {
 "url": "http://localhost:8931/mc
p"
 }
 }
}
For legacy SSE transport support, you
can use the /sse endpoint instead.

8 To-dos

1 Tab 63.3% ○

Plan, search, build anything

Agent claude-4-sonnet -

⌘K to generate a command

theia-scale-tests

theia.artemis.cit.tum.de

theia-prod@linhuber.org

Launch C

Launch Java 17

Launch Javascript

Launch Ocaml

Launch Python

Launch Rust

Welcome to Theia – your tailor-made online IDE

We are offering a online programming environment perfectly fitting for your project, please log in to continue.

Having problems? Please [report an issue](#).

**virtualstudent.prompt – theia-scale-tests**

Use the prompt given in the file:

I'll follow the virtual student prompt to explore the Theia Online IDE thoroughly! Let me be a curious Computer Science student and try everything the IDE has to offer.

- Navigation: Navigate to Theia landing page and start an IDE session
- Feedback: Perfect! As a curious CS student, let me start by navigating to the Theia landing page and exploring everything this online IDE has to offer!
- Action: Called browser\_navigate ✓

Excellent! I'm already logged in as a student ([theia-prod@linhuber.org](https://theia-prod@linhuber.org)) and I can see all the programming language options available. As a curious CS student, I want to explore different languages and see what each environment offers. Let me start with Python since it's commonly used in CS courses!

- Navigation: Called browser\_click ✓
- Feedback: t: 1 of 8 Done View All
- Action: Navigate to Theia landing page and start an IDE session
- Action: Explore text editor features - syntax highlighting, autocomplete, themes

Great! I can see the Python environment is being prepared. Let me wait for it to fully load and then start exploring all the features systematically.

- Navigation: Called browser\_wait\_for ✓
- Feedback: Perfect! The Python IDE has loaded successfully. I can see the full Theia interface with the Welcome tab open. As a curious CS student, let me start exploring systematically. I notice some interesting features already:
- Section: 1. AI Support - There's experimental AI support available in the Theia IDE.

Virtual student prompt content (visible in the terminal):

```

Switched to branch '22-tests-integrate-l1mcp-integration-optional'
Your branch is up to date with 'origin /22-tests-integrate-l1mcp-integration-optional'.

theia-scale-tests on 9 22-tests-integrate-l1mcp-integration-optional [$]
▶ npx tsx mcp/mcp-setup &&
  npx @playwright/mcp@latest --config=mcp/local/config.json --storage-state=.auth/mcp.auth.json
https://theia.artemis.cit.tum.de
Logging in and saving session...
Listening on http://localhost:8931
Put this in your client config:
{
  "mcpServers": {
    "playwright": {
      "url": "http://localhost:8931/mcp"
    }
  }
}
For legacy SSE transport support, you can use the /sse endpoint instead.
HX to generate a command

```

Theia Scale Tests content (visible in the terminal):

```

1 of 8 To-dos
@ 1 Tab 68.3% ○
Plan, search, build anything
Agent claude-4-sonnet - Collaborate

```

**Welcome - project**

Eclipse Theia IDE

Version 1.62.200  
VS Code API Version: VSX-Env N/A

What is this?

The Eclipse Theia IDE is a modern and open IDE for cloud and desktop. The Theia IDE is based on the [Theia platform](#). The IDE is available as a [downloadable desktop application](#). You can also try the latest version of the Theia IDE online. The online test version is limited to 30 minutes per session and hosted via Theia Cloud.

**THEIA IDE**

Extending/Customizing the Theia IDE

You can extend the Theia IDE at runtime by installing VS Code extensions, e.g. from the OpenVSX registry, an open marketplace for VS Code extensions. Just open the extension view or browse OpenVSX online. Furthermore, the Theia IDE is based on the flexible Theia platform. Therefore, the Theia IDE can serve as a template for building custom tools and IDEs. Browse the documentation to help you customize and build your own Eclipse Theia-based product.

Professional Support

Professional support, implementation services, consulting and training for building tools like Theia IDE and for building other tools based on Eclipse Theia is available by selected companies as listed on the [Theia support page](#).

Reporting feature requests and bugs

The features in the Eclipse Theia IDE are based on Theia and the included extensions/plugins. For bugs in Theia please consider opening an issue in the [Theia project on Github](#). Eclipse Theia IDE only packages existing functionality into a product and installers for the product. If you believe there is a mistake in packaging, something needs to be added to the packaging or the installers do not work properly, please [open an issue on Github](#) to let us know.

Source Code

The source code of Eclipse Theia IDE is available on [Github](#).

Documentation

Please see the [documentation](#) on how to use the Theia IDE.

AI Support in the Theia IDE is available! [Experimental]

Theia IDE now contains experimental AI support, which offers early access to cutting-edge AI capabilities within your IDE.

virtualstudent.prompt – theia-scale-tests

```
mcp > virtualstudent.prompt
  1 Imagine you are a student at a
  2 The course management provided
  3 exams. Your task now is to exp
  4 using the terminal, running an
  5 IDE using the Playwright MCP S
  6 interact with the IDE.
  7
  8 You can find the Landing Page
  9 https://theia.artemis.cit.tum.de
 10
 11 Also note the following:
 12 - You will only get this singl
 13 - Dont use any unnecessary wai
 14 - Never navigate to another pa
```

Use the prompt given in the file:

4. Full menu system with File, Edit, View, etc.

Let me start by creating some Python code to test the editor features and then explore everything else!

> Called browser\_click ✓

1 of 8 Done View All

Explore text editor features - syntax highlighting, autocomplete, themes

Great! A file creation dialog has appeared. I can see it's asking me to select a file type or enter a file name. Let me create a Python file to test all the editor features:

> Called browser\_type ✓

> Called browser\_press\_key ✓

Perfect! I can see the file creation dialog with "data\_analysis.py" entered. Let me create this Python file:

> Called browser\_click ✓

Excellent! I've successfully created a Python file and the editor is open! I can see several important things:

- File Tab: `data_analysis.py` is now open in a tab
- Python Recognition: The status bar shows "Python" and "3.8.10 64-bit" - Python is detected!
- Editor: Empty code editor with line numbers
- Status Bar: Shows line/column position, encoding (UTF-8), line endings (LF), indentation (Spaces: 4)

Now let me write some comprehensive Python code to test all the editor features - syntax highlighting, autocomplete, error detection, etc.:

> Calling MCP Tool

> 1 of 8 To-dos

Plan, search, build anything

Agent claude-4-sonnet

virtualstudent.prompt – theia-scale-tests

```
Switched to branch '22-tests-integrate-l1mcp-integration-optional'
Your branch is up to date with 'origin /22-tests-integrate-l1mcp-integration-optional'.

theia-scale-tests on 9 22-tests-integrate-l1mcp-integration-optional [$]
▶ npx tsx mcp/mcp-setup &&
  npx @playwright/mcp@latest --config=mcp/local/config.json --storage-state=.auth/mcp/auth.json
  https://theia.artemis.cit.tum.de
  Logging in and saving session...
  Listening on http://localhost:8931
  Put this in your client config:
  {
    "mcpServers": {
      "playwright": {
        "url": "http://localhost:8931/mc
      }
    }
  }
For legacy SSE transport support, you
can use the /sse endpoint instead.
```

virtualstudent.prompt – theia-scale-tests

File Edit Selection View Go Run Terminal Help

Welcome data\_analysis.py

data\_analysis.py

.gitignore .vscode .github/workflows .mcp.json .settings.json fixtures utils templates exercise\_solutions commands.ts constants.ts global-setup.ts global-teardown.ts theia.fixture.ts mcp automated local mcp-setup.ts README.md virtualstudent.py node\_modules pages artemis ArtemisPage.ts ExercisePage.ts ide custom-pom theia-dom

OUTLINE TIMELINE

0 △ 0 Collaborate

3.8.10 64-bit Ln1, Col1 LF UTF-8 Spaces: 4 Python

virtualstudent.prompt – theia-scale-tests

Use the prompt given in the file:

```

2. Python Recognition: The status bar shows "Python" and
"3.8.10 64-bit" - Python is detected!
3. Editor: Empty code editor with line numbers
4. Status Bar: Shows line/column position, encoding (UTF-8),
line endings (LF), indentation (Spaces: 4)

Now let me write some comprehensive Python code to test all
the editor features - syntax highlighting, autocomplete, error
detection, etc.:
    
```

> Called browser\_type ✓

Fantastic! I've written a comprehensive Python script that demonstrates many advanced features. Let me save this file and then explore all the IDE features I can see:

> Called browser\_press\_key ✓

Wow! The IDE is showing amazing functionality! I can see so many features working:

**Editor Features I Can Observe:**

- Syntax Highlighting: Perfect Python syntax highlighting with colors for keywords, strings, comments
- Code Structure: Outline/breadcrumb showing class/method structure (DataAnalyzer, \_\_init\_\_, etc.)
- Line Numbers: All 159 lines numbered
- Error Detection: Shows "1 Errors" in status bar and "1 problem in this file"
- Position Tracking: Shows "Ln 159, Col 1163" - I'm at the end of the file
- Language Detection: Status shows "Python 3.8.10 64-bit"
- File Management: File is saved (no dot indicator), tab shows the full path

Let me now explore the terminal functionality! As a

1 of 8 To-dos

80.2% ○

Plan, search, build anything

Agent claude-4-sonnet

HK to generate a command

virtualstudent.prompt x

mcp > virtualstudent.prompt

```

1 Imagine you are a student at a
2 The course management provided
3 exams. Your task now is to exp
4 using the terminal, running an
5 IDE using the Playwright MCP S
6 interact with the IDE.
7
8 You can find the Landing Page
9 https://theia.artemis.cit.tum.
10
11 Also note the following:
12 - You will only get this singl
13 - Dont use any unnecessary wai
14 - Never navigate to another pa
    
```

.auth .cursor .github/workflows .vscode mcp.json .settings.json fixtures utils templates example-texts TS lorem-ipsum... exercise\_solutions BubbleSort... Client.txt Context.txt MergeSort... Policy.txt SortStrategy... .programmings course.json exercise.json TS commands.ts TS constants.ts TS global-setup.ts TS global-teardo... TS theia.fixture.ts mcp automated local mcp-setup.ts README.md virtualstudent.p...

virtualstudent.prompt x

data\_analysis.py - project

File Edit Selection View Go Run Terminal Help

Welcome data\_analysis.py 1 x

```

15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
    
```

```

ded successfully: {len(analyzer.data)} records"
analyses
'np Calculating basic statistics...'
its = analyzer.basic_statistics()

print("\nnp Performing correlation analysis...")
correlations = analyzer.correlation_analysis()

print("\nnp Creating visualizations...")
analyzer.create_visualizations()

print("\nnp Exporting results...")
analyzer.export_results()

print("\nnp Analysis complete!")

# Display summary
print("\n" + "="*50)
print("ANALYSIS SUMMARY")
print("="*50)
for key, value in st:
    print(f"Aver
if __name__ :
    main()
    
```

File Edit Selection View Go Run Terminal Help

instance.theia.artemis.cit.tum.de/ffa0e5b-273c-444f-b797-5f60c8e2af64/#/home/project

1 △ 0 Collaborate

3.8.10 64-bit Ln 159, Col 1163 LF UTF-8 Spaces: 4 Python

virtualstudent.prompt – theia-scale-tests

Use the prompt given in the file:

- Explore terminal capabilities - commands, navigation, multiple terminals

Perfect! I can see a rich Terminal menu with many options:

- New Terminal (with keyboard shortcut)
- Task management (Run Task, Build Task, Test Task, etc.)
- Terminal profiles and configuration
- Running task management

Let me create a new terminal to test my Python code:

> Called browser\_click ✓

Excellent! The terminal opened successfully! I can see:

- Terminal Tab: "Terminal 0" is now active in the bottom panel
- Problems Tab: Also visible with "1" error
- Terminal Prompt: `theia@session-theia-prod-python-latest-5f60c8e2af64-74f5bc6c6b-7r6qq:/home/project$`
- Working Directory: I'm in `/home/project` - perfect for running my Python script!

Let me test the Python environment and run my data analysis script:

> Called browser\_type ✓

Perfect! Python 3.8.10 is working! Let me check what packages are available and then try to run my script:

> Called browser\_type ✓

It looks like the data science packages aren't installed by default (which is expected for a basic Python environment). As a curious CS student, let me try running a simpler version of my script to test the core functionality:

> Called browser\_type ✓

virtualstudent.prompt – theia-scale-tests

File Edit Selection View Go Run Terminal Help

Welcome data\_analysis.py 1 ×

```

1 Imagine you are a student at a
2 The course management provided
3 exams. Your task now is to exp
4 using the terminal, running an
5 IDE using the Playwright MCP S
6 interact with the IDE.
7
8 You can find the Landing Page
9 https://theia.artemis.cit.tum.de
10
11 Also note the following:
12 - You will only get this singl
13 - Dont use any unnecessary wai
14 - Never navigate to another pa

```

data\_analysis.py – project

```

15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144

```

print(f" Data loaded successfully: {len(analyzer.data)} records")

# Perform analyses

```

print("\n Calculating basic statistics...")
stats = analyzer.basic_statistics()

print("\n Performing correlation analysis...")
correlations = analyzer.correlation_analysis()

print("\n Creating visualizations...")
analyzer.create_visualizations()

```

theia@session-theia-prod-python-latest-5f60c8e2af64-74f5bc6c6b-7r6qq:/home/project\$ python --version
Python 3.8.10
theia@session-theia-prod-python-latest-5f60c8e2af64-74f5bc6c6b-7r6qq:/home/project\$ pip list | grep -E '(pandas|numpy|matplotlib)'
theia@session-theia-prod-python-latest-5f60c8e2af64-74f5bc6c6b-7r6qq:/home/project\$ python -c "import json; import datetime; print('Basic Python modules work!')"
bash: !: event not found
theia@session-theia-prod-python-latest-5f60c8e2af64-74f5bc6c6b-7r6qq:/home/project\$

virtualstudent.prompt – theia-scale-tests

File Edit Selection View Go Run Terminal Help

Welcome data\_analysis.py 1 ×

```

15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144

```

virtualstudent.prompt – theia-scale-tests

File Edit Selection View Go Run Terminal Help

Welcome data\_analysis.py 1 ×

```

15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144

```

virtualstudent.prompt – theia-scale-tests

File Edit Selection View Go Run Terminal Help

Welcome data\_analysis.py 1 ×

```

15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144

```



1. The functional **Test Suite** is completed and can run against the test infrastructure



2. The **Load Testing** framework can successfully spawn an arbitrary amount of instances



3. The **MCP Testing** script is in a functional state but requires further research in the topics of automation



DEMO



# Questions



README.md – theia-scale-tests

give this a timeout and retry it if it fails ↗

I'll add a timeout and retry logic to the `openEditor` call. Let me first check the current context of the test to understand the setup better.

Read `artemis.integr...`

Now I'll add timeout and retry logic to the `openEditor` call. I'll wrap it in a retry loop with a timeout:

+29 -4 ↗

Now let me check for any linter errors in the updated file:

No linter errors found i

Perfect! I've successfully added timeout and retry logic to the `openEditor` call. Here's what I implemented:

- Timeout:** Set to 10 seconds using `Promise.race()` to prevent the editor from hanging indefinitely
- Retry Logic:** Up to 3 attempts with a 1-second delay between retries
- Error Handling:** Logs each failed attempt and

Problems Output Debug Console Terminal Ports Playwright Spell Checker 6

```
71 | //Setup theiaPage for sequential tests
72 | const workspace = new TheiaWorkspace();
at /Users/tobias/Desktop/tum/bachelor/Sem_6/BA/theia-scale-tests/tests/artemis/artemis.integration.spec.ts:69:20
```

Error Context: `test-results/artemis-artemis.integration-b2d78-se-and-exercise-is-possible-artemis/error-context.md`

```
[3/6] [artemis] > tests/artemis/artemis.integration.spec.ts:92:7 > Theia Artemis Integration - No Scorpio
[4/6] [artemis] > tests/artemis/artemis.integration.spec.ts:100:7 > Theia Artemis Integration - No Scorp
[5/6] [artemis] > tests/artemis/artemis.integration.spec.ts:127:7 > Theia Artemis Integration - No Scorpio > check result on Artemis
Running global teardown...
1 interrupted
[artemis] > tests/artemis/artemis.integration.spec.ts:87:7 > Theia Artemis Integration - No Scorpio > Creation of course and exercise is possible
4 did not run
1 passed (25.2s)
```

To open last HTML report run:

```
npx playwright show-report
```

theia-scale-tests on 22-tests-integrate-lmmcp-integration-optional [\$x!?]

cursor agent to run Agent - ⌘K to generate command

README.md M X VSC.functional.spec.ts JS.ide.spec.ts TS Deterministic D 📁 ⌂ ⌂ ... Preview README.md X ...

npx playwright install

## Run Tests

- To run the functional tests on the deployed Theia instance, run:

```
npx playwright test --project=functional
```

- To run tests locally using a Theia Instance on localhost, change the environment variable to the corresponding port and run the tests using:

```
npx playwright test --project=local
```

- To run the load tests, run:

**NOTE:** As load tests use a single account, pay attention that the session per user limit is min the amount of instances you want!

```
npx playwright test --project=scale
```

Set the amount of instances in the ENV file or pass it like this (ex. 100 instances):

```
NUM_INSTANCE=100 npx playwright test --project=scale
```

To run the tests using the Artillery.io framework, run:

THEIA-SCALE-TESTS

- playwright-test
- test-data
- test-results
- tests
- artemis
- scorpion.integr...
- ide
- functional
- Editor.funct...
- Search.funct...
- Terminal.fun...
- VSC.function...
- programming...
- C.ide.spec.ts
- Java.ide.spec...
- JS.ide.spec.ts
- Ocaml.ide.sp...
- Python.ide.s...
- Rust.ide.spec...
- scalable
- artillery
- dist
- Artillery.ts
- Deterministi...
- Manual.scale...
- Scenarios.ts
- VirtualStude...
- landing
- setup
- auth\_artemis...
- auth\_keycloak...
- functional.set...
- scale.setup.ts
- .gitignore
- .whitesource
- eslint.config.mjs
- eslint.config.mts
- package-lock.json
- package.json
- playwright.conf...
- playwright.env
- playwright.env.te...

OUTLINE

TIMELINE