

SCAI CSE Automation Projects: Jupyter Textbooks

Alexander Ng, Vikriti Lokegaonkar, Punith Raju Janardhana, Tanay Upreti, Anmol Rao¹, Ryan T. Woo¹, Dr. Gennaro De Luca¹, Dr. Yinong Chen¹

¹School of Computing and Augmented Intelligence

Background and Introduction

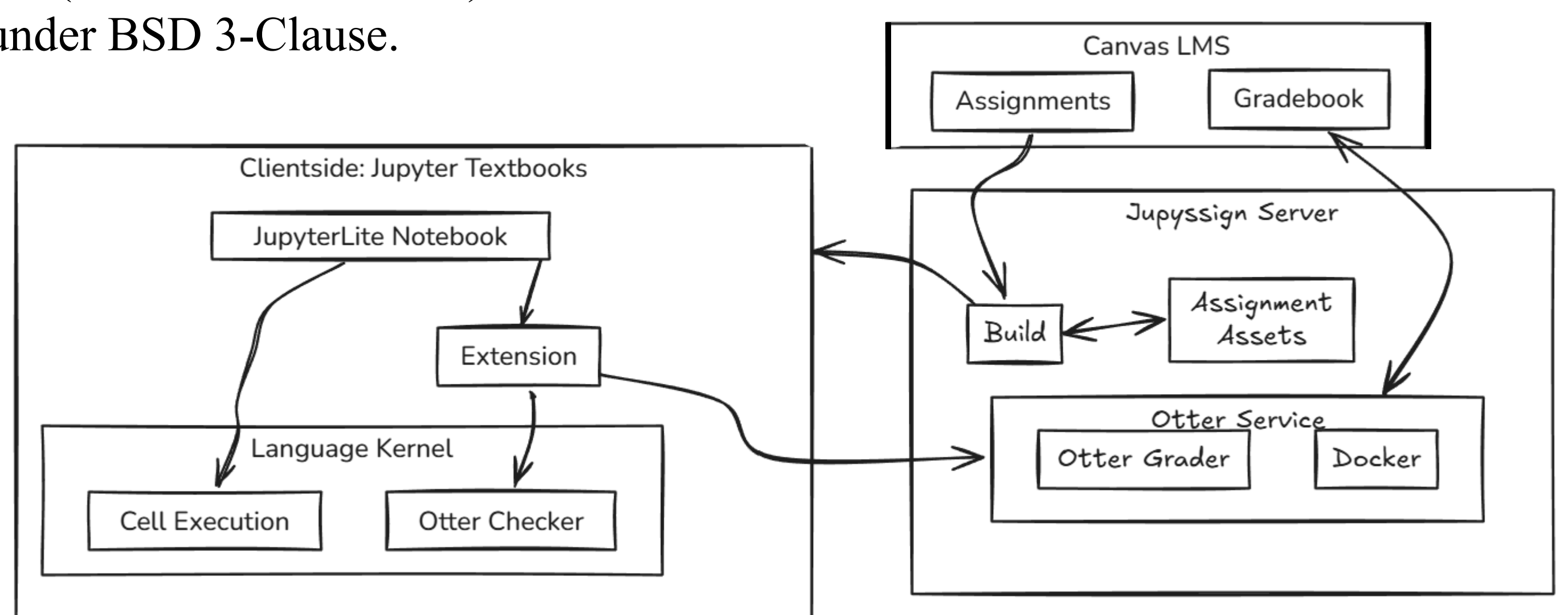
Interactive digital textbooks with code execution capabilities have proven to be useful for improving the educational experience and outcomes for computing students [1]. However, many are provided only as proprietary cloud-based software-as-a-service products. This project seeks to democratize access to this technology by better leveraging clientside compute and existing open-source software.

Project Requirements

1. Leverage existing open-source code notebook software and autograder toolchains.
2. Prioritize clientside browser-based compute to allow for code execution.
3. Integrate with existing learning management system assignment modules for seamless access by instructional staff and students.

Design Description: Approach & Methodology

1. JupyterLab [2] is an open-source project (revised BSD license) Otter-Grader [3] is similarly licensed under BSD 3-Clause.
2. JupyterLite runs entirely within the browser with Python support via the Pyodide language kernel [4] (CPython ported to WebAssembly).
3. The learning tools interoperability (LTI) 1.3 [5] standard was used for to enable learning management system integration, e.g. grade sync.



Results: Prototype Implementation

Instructor Tooling

The interface shows a code editor with the following Python code:

```
# BEGIN SOLUTION
# Your solution here
# BEGIN SOLUTION
triangle_type = {
    1: "equilateral",
    2: "isosceles",
    3: "scalene"
}

def triangle_type(items):
    return triangle_type.get(len(set(items)), " ")

# END SOLUTION

# BEGIN TESTS
points: 1
# BEGIN TEST CONFIG
# END TEST CONFIG

def test_triangle_eq():
    assert f(1, 1, 1) == "equilateral"
    assert f(1, 2, 3) == "equilateral"
    assert f(2, 2, 2) == "equilateral"

test_triangle_eq(triangle_type)

# END SOLUTION
```

Buttons for 'Add Problem', 'Add Answer', and 'Publish Assignment' are visible. The interface also includes a 'Jupyter Notebook Assignment' section with a 'Create Deep Link Resource' button.

Student Interface

The screenshot shows a student's view of an assignment in Canvas LMS. The assignment is titled 'Assignment 1: Welcome to Jupyter' and is worth 100 points. The student interface includes a 'Submit' button and a 'Check Answer' button. The assignment content includes a question about triangle types and a code editor with the same Python code as seen in the instructor tooling.

Future Work

1. Custom widgets in the textbook environment for embedded check for understanding questions
2. Interactive debugging support and question integration
3. Integration with CSEAP's other intelligent tutoring, adaptive learning, and generative tools (reading materials, assignments, question banks, etc.)

References

- [1] zyBooks, "Build Confidence and Save Time With Interactive Textbooks," zyBooks, 2025. [Online]. Available: <https://www.zybooks.com/>. Accessed: Nov. 19, 2025.
- [2] Project Jupyter, "JupyterLab: A Next-Generation Notebook Interface," jupyter.org. [Online]. Available: <https://jupyter.org/>. Accessed: Nov. 19, 2025.
- [3] Otter-Grader Developers, "Otter-Grader Documentation," Read the Docs. [Online]. Available: <https://otter-grader.readthedocs.io/en/latest/>. Accessed: Nov. 19, 2025.
- [4] Pyodide Contributors, "Pyodide — Version 0.29.0 (stable documentation)," pyodide.org. [Online]. Available: <https://pyodide.org/en/stable/>. Accessed: Nov. 19, 2025.
- [5] IMS Global Learning Consortium, "Learning Tools Interoperability (LTI) Core Specification 1.3," imsglobal.org. [Online]. Available: <https://www.imsglobal.org/spec/lti/v1p3/>. Accessed: Nov. 19, 2025.