



**JOHANNES KEPLER
UNIVERSITÄT LINZ**

**o.Univ.-Prof. Dr.
Hanspeter Mössenböck**
Institute for System Software

T +43 732 2468 4340
F +43 732 2468 4345
hanspeter.moessenboeck@jku.at

Secretary:
Karin Gusenbauer
Ext 4342
karin.gusenbauer@jku.at

Master's Thesis

Full WASM Support for GraalVM Node.js Applications

Student: Florian Huemer
Advisor: Prof. Hanspeter Mössenböck
Co-Advisor: Dr. Christian Wirth
Begin: February 1, 2022

Graal.js [1] is a JavaScript (ECMAScript 2021+) interpreter, implemented in Java. It uses specialization to optimize its execution, e.g., specialization on actually used data types or other types of run-time feedback (profiling).

Web Assembly [3] (WASM) is a bytecode-based language with the main purpose of executing code compiled from C or other native languages in (JavaScript-based) web applications. With its focus on efficiency, safety, and portability it represents a way to improve execution performance and user comfort in these applications.

Graal.js as shipped in GraalVM 21.3 already offers basic WASM support. However, it lacks many relevant features, extensions, and tools, and suffers from poor performance and poor usability. All these aspects should be explored, implemented/improved, documented, and evaluated as part of this thesis. The goal of this work is to provide an improved WASM engine in GraalVM and a documentation of the overall state of the engine.

The scope of this thesis is as follows:

- Implement open features and relevant extensions as part of the WASM proposal process.
- Provide debugging and tooling support by implementing the relevant Truffle APIs and Nodes.
- Improve performance of the WASM interpreter and WASM Nodes. Provide relevant benchmarks and monitor improvements and regressions.
- Implement new or existing Truffle APIs relevant for the WASM engine.
- Explore the interoperability with other Truffle languages (array access, etc.).
- Provide at least one executable example that demonstrates the capabilities of the approach, i.e., a reasonably large Node.js application that can be executed, tested, benchmarked, and debugged.
- Contribute this implementation as open source to the Graal.js [1] and GraalVM [2] repositories under the UPL (requires signing the Oracle Contributor Agreement [4])

The work's progress should be discussed with the supervisor at least every 2 weeks. Please note the guidelines of the Institute for System Software when preparing the written thesis. The deadline for the written thesis is January 31, 2023.

References:

- [1] <https://www.github.com/oracle/graaljs>
[2] <https://www.github.com/oracle/graal>
[3] <https://www.webassembly.org>
[4] <https://oca.opensource.oracle.com/>

**JOHANNES KEPLER
UNIVERSITÄT LINZ**
Altenberger Straße 69
4040 Linz, Österreich
www.jku.at
DVR 0093696