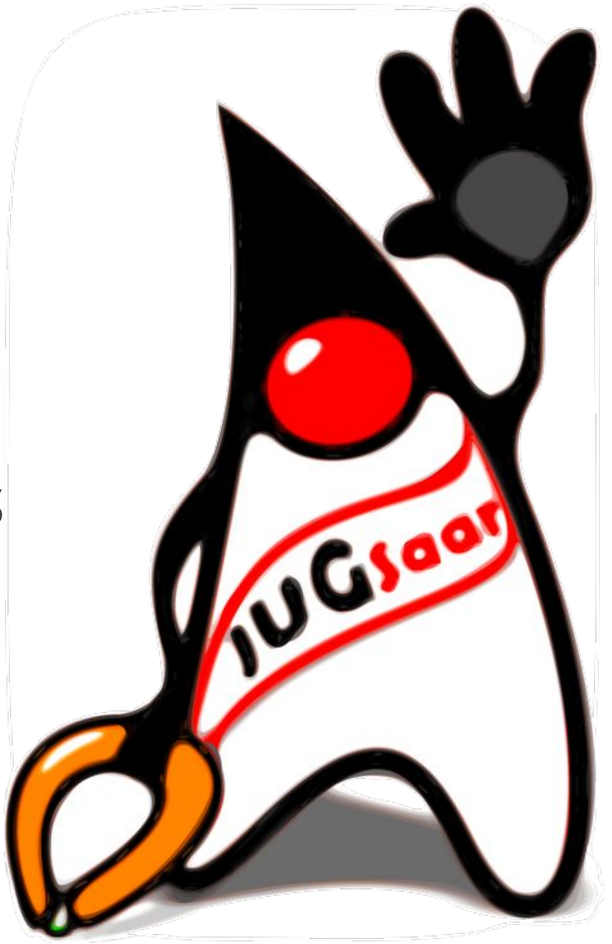
Java User Group Saarland

WebAssembly for Java Developers

Thomas Darimont & Florian Fromm

61. Meeting

07. Mar 2023



Sponsored by

codecentric   >eurodata   INFOSERVE >eurodata-Gruppe

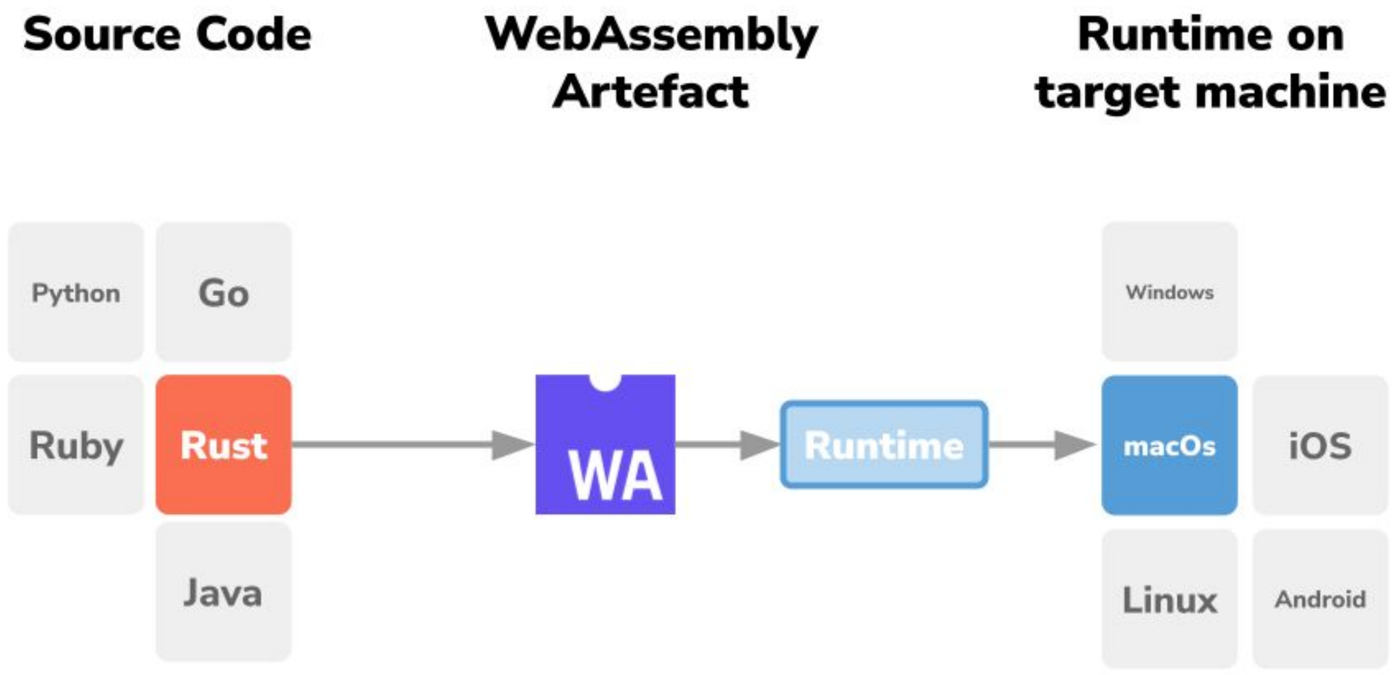
WA

WEBASSEMBLY

WEBASSEMBLY

”**WebAssembly** *or* **Wasm** *is a* ***binary*** *instruction* ***format*** *for a stack-based* ***"Virtual-machine."***

”Wasm is ***designed*** *as a* ***portable compilation target*** *for programming languages, enabling* ***deployment*** *on the web for* ***client*** *and* ***server*** *applications.”*



Source: (Slightly adjusted) https://b-nova.com/en/home/content/how-containerless-works-thanks-to-web-assembly-runtimes

```java
int add(int a, int b) {
    return a + b;
}
```

Java

```
(module
 (func $add (param $a i32) (param $b i32) (result i32)
   local.get $a
   local.get $b
   i32.add)
  ...
)
```

WAT (Web Assembly Text)

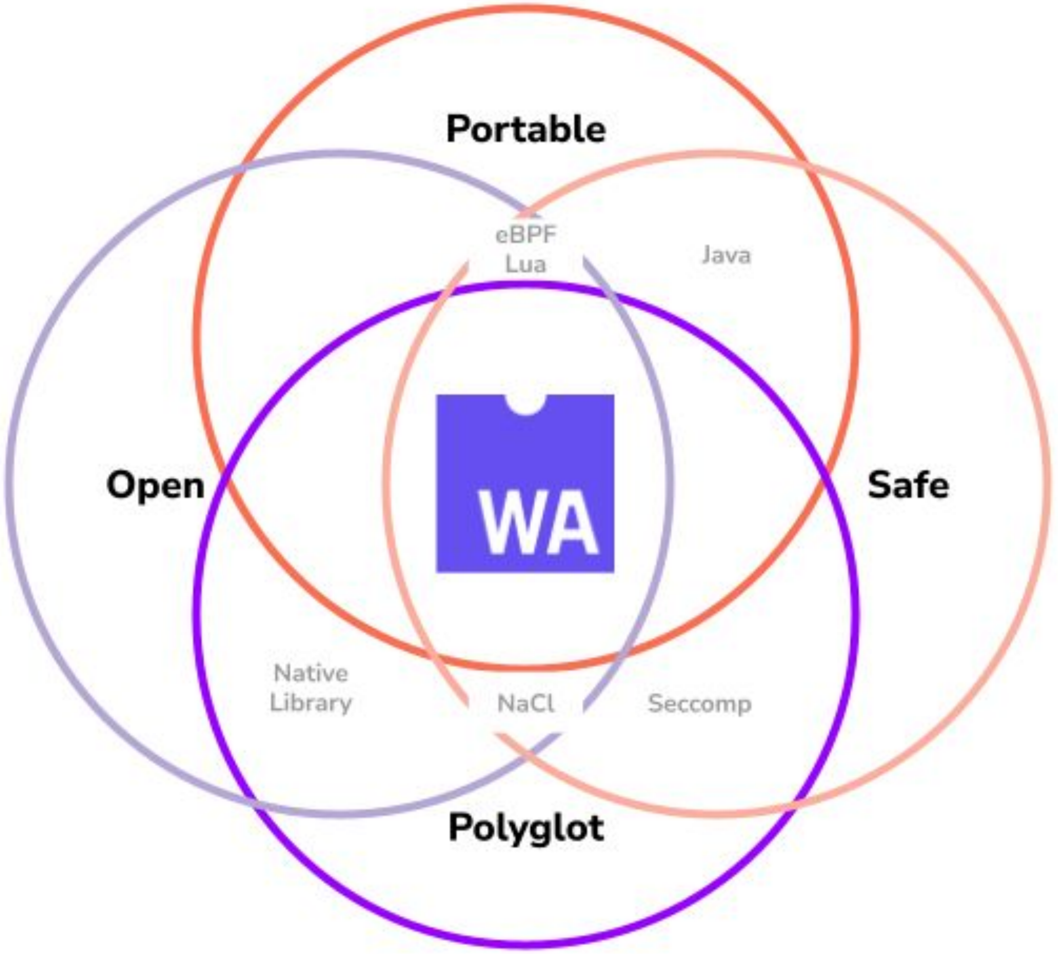
```
00000000  00 61 73 6d 0b 00 00 00  04 74 79 70 65 87 80 80  |.asm.....type...|
00000010  80 00 01 40 02 01 01 01  01 08 66 75 6e 63 74 69  |...@......functi|
00000020  6f 6e 82 80 80 80 00 01  00 06 6d 65 6d 6f 72 79  |on........memory|
00000030  85 80 80 80 00 80 02 80  02 01 06 65 78 70 6f 72  |...........expor|
00000040  74 86 80 80 80 00 01 00  03 61 64 64 04 63 6f 64  |t.......add.cod|
00000050  65 8c 80 80 80 00 01 86  80 80 80 00 00 14 00 14  |e...............|
00000060  01 40 04 6e 61 6d 65 86  80 80 80 00 01 03 61 64  |.@.name.......ad|
00000070  64 00                                             |d.|
```

WASM (Web Assembly)

Benefits of Web Assembly

Runs on “every” Platform

Open Standard
Wide adoption



Sandbox
Confined memory
Capability config

Many languages compile to Web Assembly

Source: (Slightly adjusted) https://b-nova.com/en/home/content/how-containerless-works-thanks-to-web-assembly-runtimes

BYTECODE ALLIANCE

About the Bytecode Alliance

The Bytecode Alliance is a nonprofit organization dedicated to creating secure new software foundations, building on standards such as WebAssembly and WebAssembly System Interface (WASI).

The Bytecode Alliance is committed to establishing a capable, secure platform that allows application developers and service providers to confidently run untrusted code, on any infrastructure, for any operating system or device, leveraging decades of experience doing so inside web browsers.

We have a vision for a secure-by-default WebAssembly ecosystem for all platforms.

Where can Web Assembly be used? *

Language Interoperability
Write library once; use with other languages
Figma, Google Earth, Adobe Photoshop

*) outside the Browser

Where can Web Assembly be used? *

Language Interoperability	**Plugin Systems**	**Embedded Sandboxing**	**Containerization**	**Serverless**
Write library once; use with other languages	Flexible & secure plugin systems	Guard yourself against bugs in 3rd-party libraries	Universal Runtime, capability based security model	Minimal startup time, maximum isolation
Figma, Google Earth, Adobe Photoshop	Envoy / Istio, Kubewarden, Minecraft, MS Flight Simulator	Firefox, HttpServers	Kurstlet, Hippo, WasmCloud, WasmEdge	CloudFlare Workers, AWS Lambda, Fastly, Fermyon Spin

Source: (Slightly Adjusted) Think container orchestration different - WASM is coming | Max Körbächer | Christoph Voigt
https://youtu.be/hZw8xmqkKwU?t=714

*) outside the Browser

What’s in for Java Developers?

- **Polyglot**
  - Run code written in other languages
  - Allow programmers to provide features with preferred language
- **Open and Extensible**
  - Make existing programs extensible
- **Efficient and fast**
  - Fast start-times
  - Can be faster than JavaScript
- **Secure**
  - Sandbox model built-in
  - Restricted memory
  - Explicit capability mapping (FS / NET / OS access)

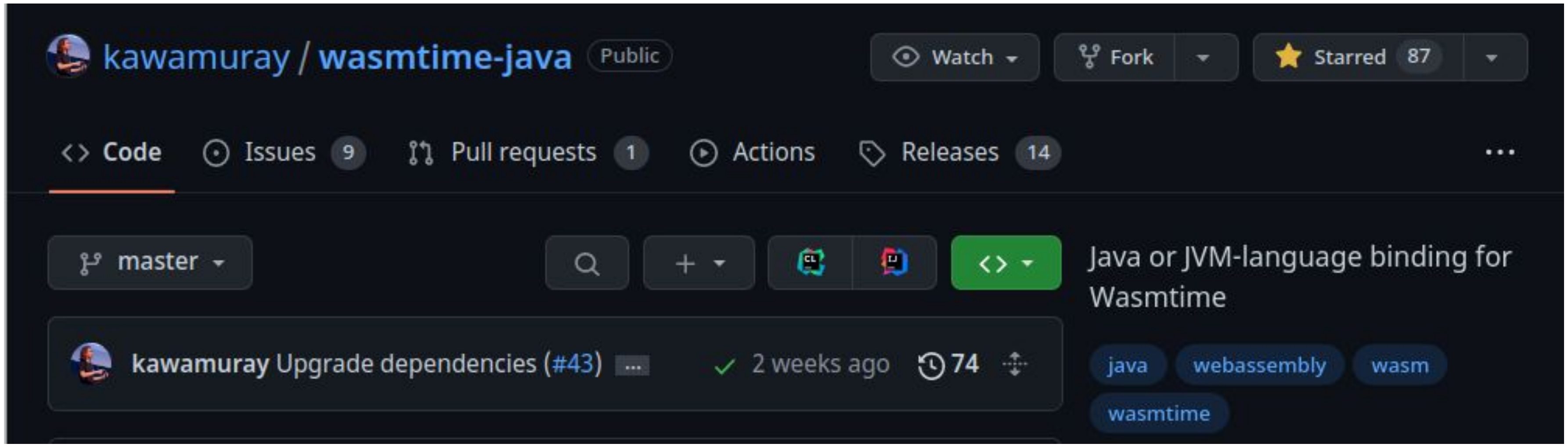
Java and Web Assembly

Wasmtime

A fast and secure runtime for WebAssembly

A Bytecode Alliance project

- wasmtime-java unofficial Java Support
- Calls wasmtime (rust) native library via Java Native Interface (JNI)
- Low-level interface



kawamuray / wasmtime-java Public

Watch · Fork · Starred 87

Code · Issues 9 · Pull requests 1 · Actions · Releases 14

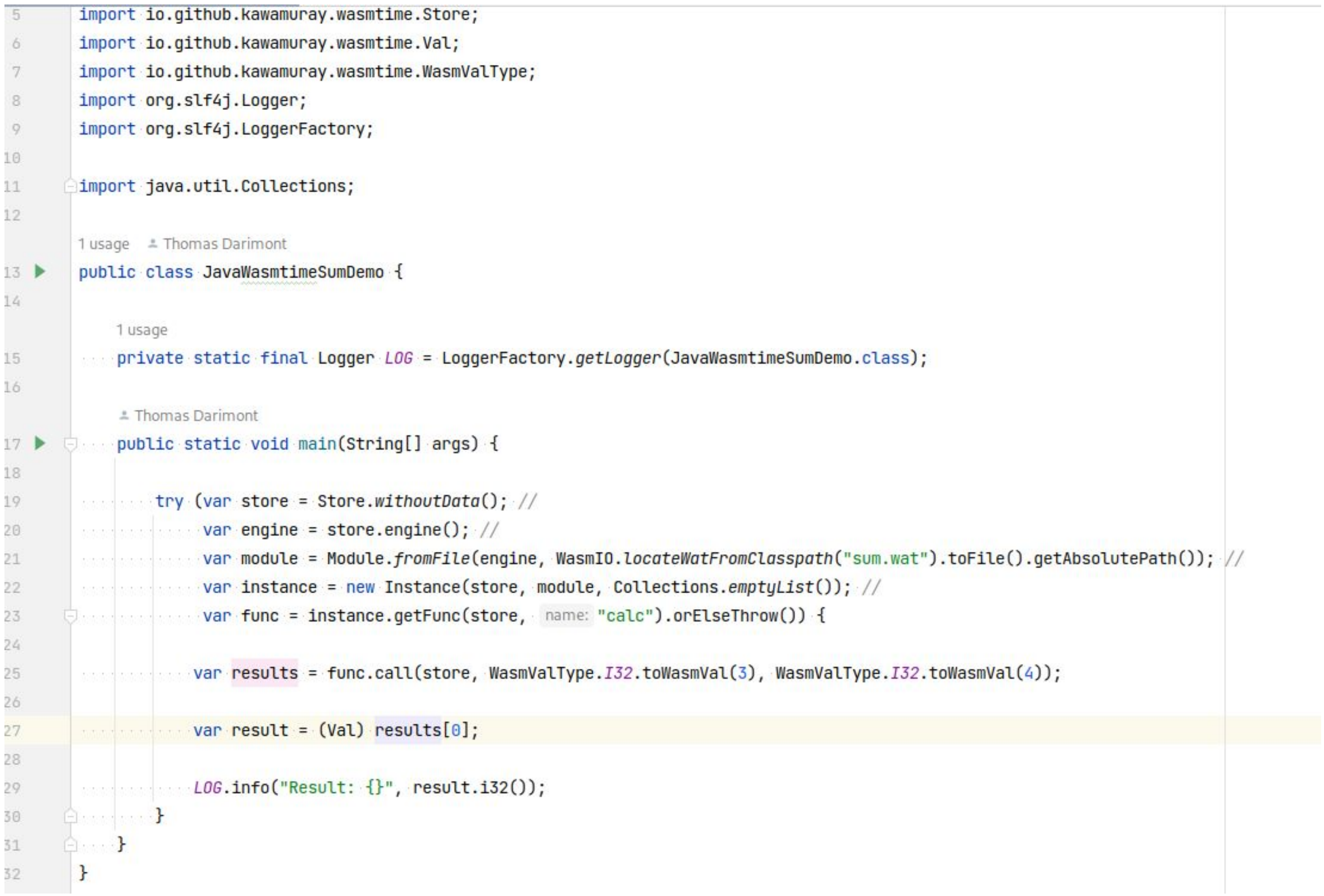
master

kawamuray Upgrade dependencies (#43) ... ✓ 2 weeks ago 74

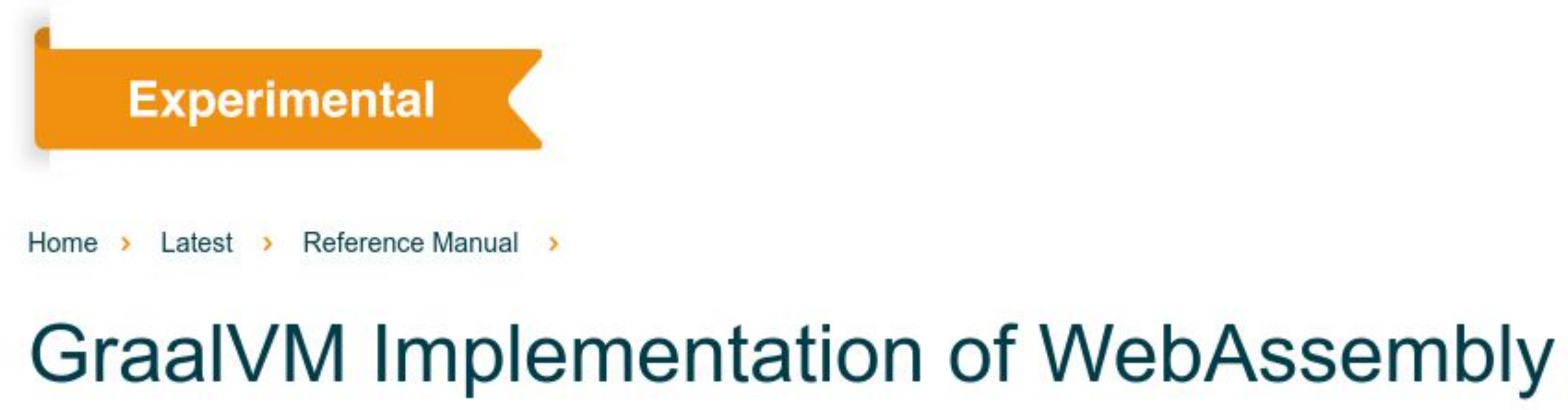
Java or JVM-language binding for Wasmtime

java · webassembly · wasm · wasmtime

wasmtime-java Demo



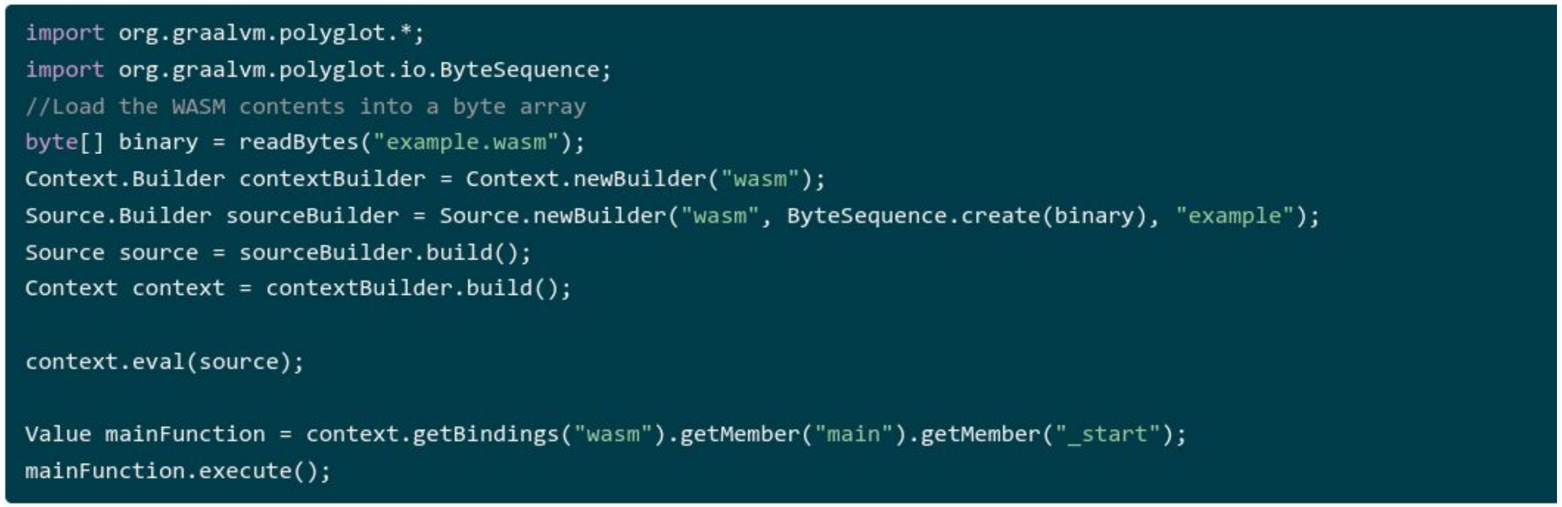
```java
import io.github.kawamuray.wasmtime.Store;
import io.github.kawamuray.wasmtime.Val;
import io.github.kawamuray.wasmtime.WasmValType;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import java.util.Collections;

1 usage  Thomas Darimont
public class JavaWasmtimeSumDemo {

    1 usage
    private static final Logger LOG = LoggerFactory.getLogger(JavaWasmtimeSumDemo.class);

    Thomas Darimont
    public static void main(String[] args) {

        try (var store = Store.withoutData(); //
             var engine = store.engine(); //
             var module = Module.fromFile(engine, WasmIO.locateWatFromClasspath("sum.wat").toFile().getAbsolutePath()); //
             var instance = new Instance(store, module, Collections.emptyList()); //
             var func = instance.getFunc(store, name: "calc").orElseThrow()) {

            var results = func.call(store, WasmValType.I32.toWasmVal(3), WasmValType.I32.toWasmVal(4));

            var result = (Val) results[0];

            LOG.info("Result: {}", result.i32());
        }
    }
}
```


Experimental

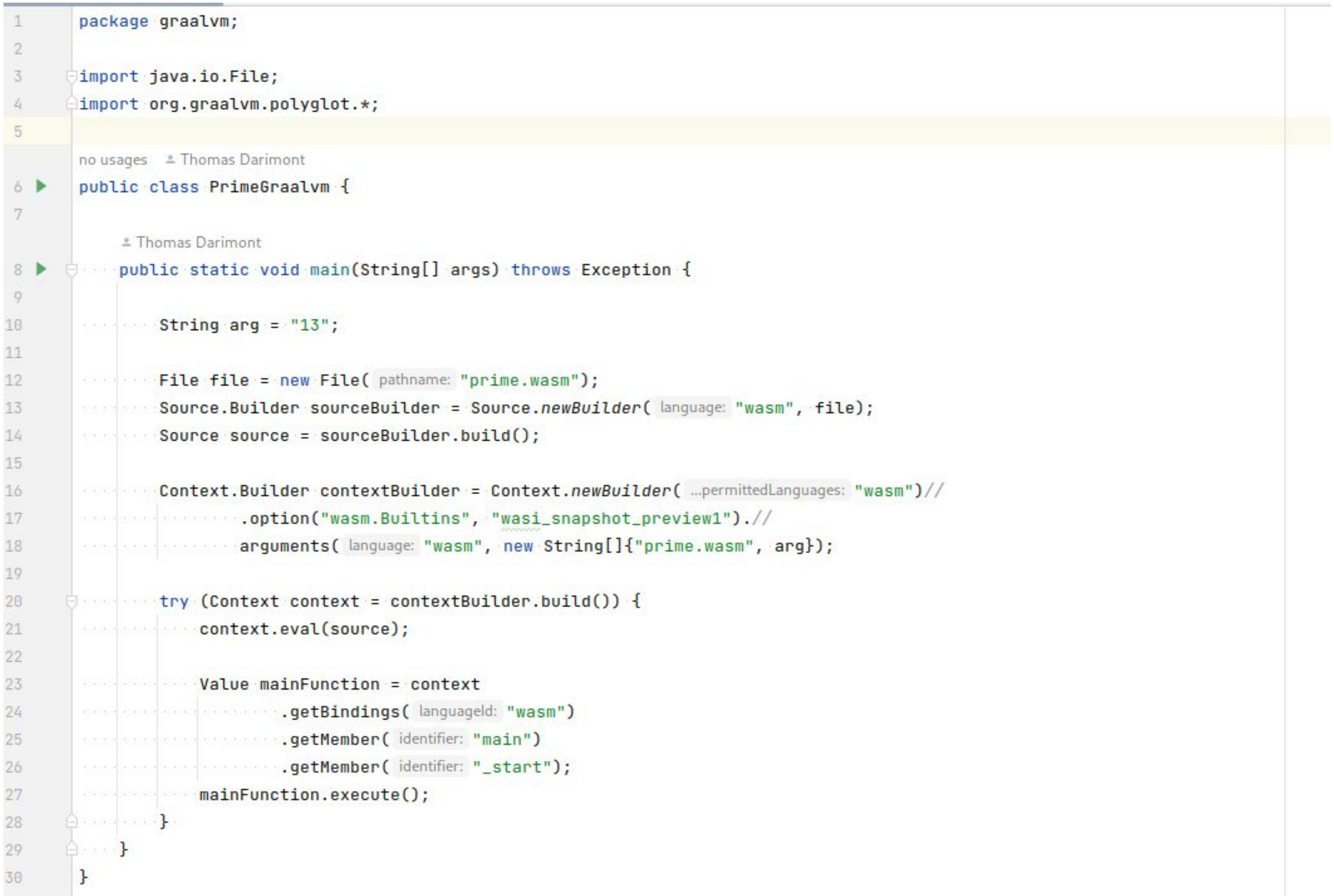
Home › Latest › Reference Manual ›

GraalVM Implementation of WebAssembly

- GraalVM Polyglot supports WASM
- Interpret and compile WebAssembly code to binary format.
- Running WebAssembly Programs
- Embedding WebAssembly Programs

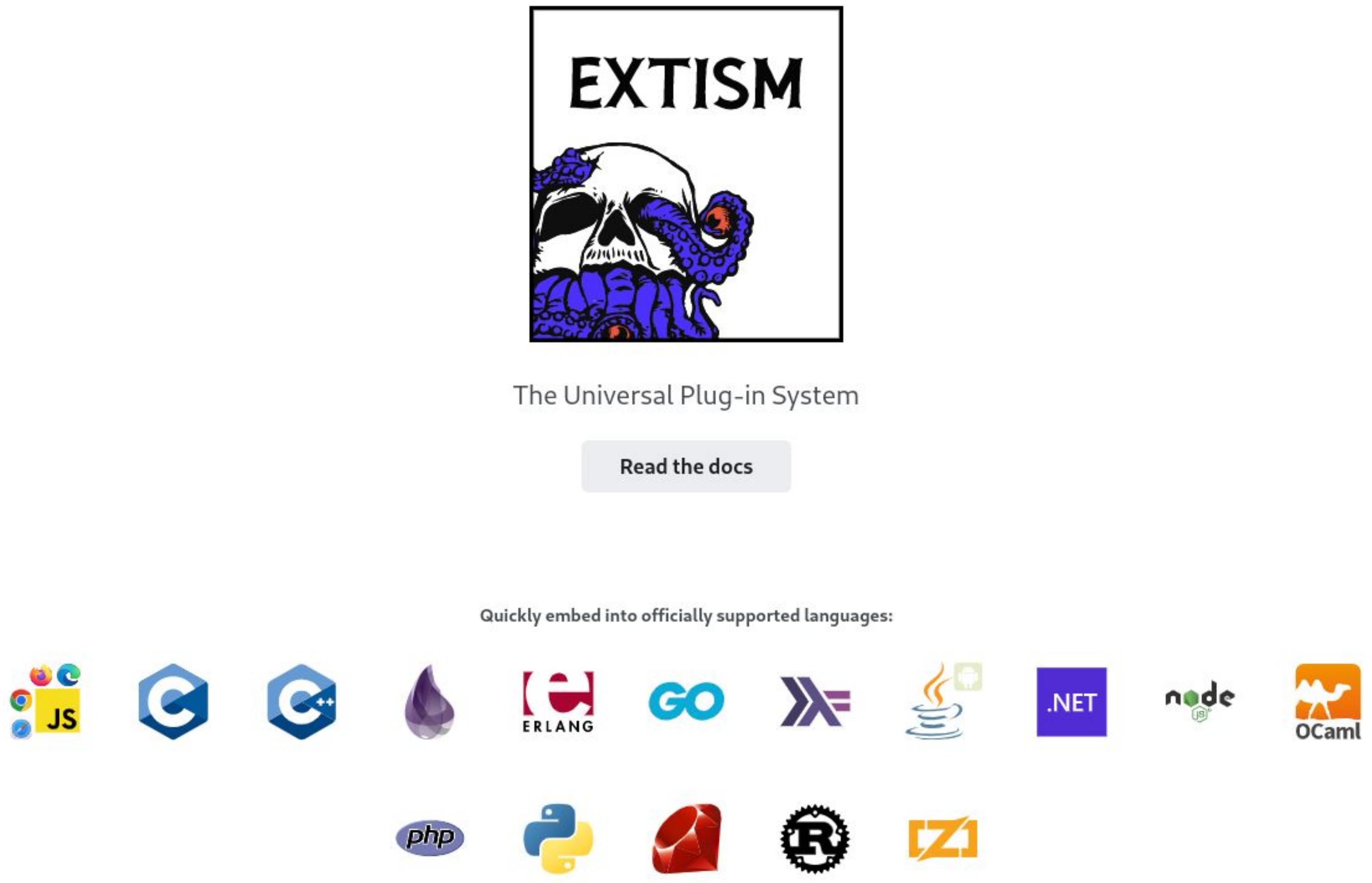

```java
import org.graalvm.polyglot.*;
import org.graalvm.polyglot.io.ByteSequence;
//Load the WASM contents into a byte array
byte[] binary = readBytes("example.wasm");
Context.Builder contextBuilder = Context.newBuilder("wasm");
Source.Builder sourceBuilder = Source.newBuilder("wasm", ByteSequence.create(binary), "example");
Source source = sourceBuilder.build();
Context context = contextBuilder.build();

context.eval(source);

Value mainFunction = context.getBindings("wasm").getMember("main").getMember("_start");
mainFunction.execute();
```

GraalVM WASM Demo



```java
package graalvm;

import java.io.File;
import org.graalvm.polyglot.*;

no usages   Thomas Darimont
public class PrimeGraalvm {

      Thomas Darimont
    public static void main(String[] args) throws Exception {

        String arg = "13";

        File file = new File( pathname: "prime.wasm");
        Source.Builder sourceBuilder = Source.newBuilder( language: "wasm", file);
        Source source = sourceBuilder.build();

        Context.Builder contextBuilder = Context.newBuilder( ...permittedLanguages: "wasm")//
                .option("wasm.Builtins", "wasi_snapshot_preview1").//
                arguments( language: "wasm", new String[]{"prime.wasm", arg});

        try (Context context = contextBuilder.build()) {
            context.eval(source);

            Value mainFunction = context
                    .getBindings( languageId: "wasm")
                    .getMember( identifier: "main")
                    .getMember( identifier: "_start");
            mainFunction.execute();
        }
    }
}
```

Extism Universal Plug-in System



The Universal Plug-in System

Read the docs

Quickly embed into officially supported languages:

Extism

- **Plugin SDKs → Java SDK**
  Wrapper around wasmtime via JNA
  Support for WASI (Web Assembly System Interface)

- **Flexible Data-Exchange**
  Data-exchange between Host and WASM module via JSON

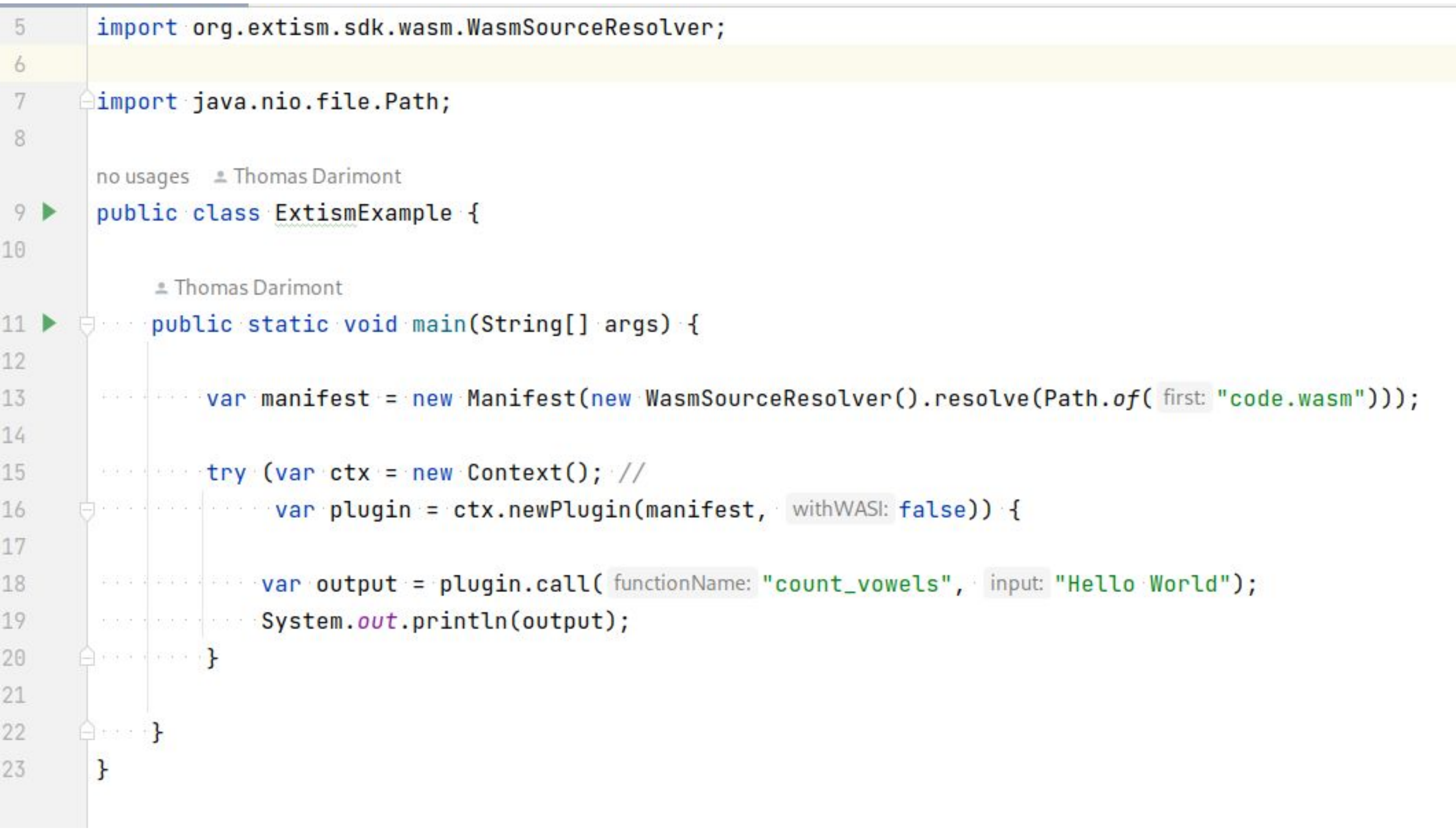
- **Easy to Use**
  Leveraging the power and portability of WebAssembly, Extism is an off-the-shelf plug-in system just a library import away. Ship in days, not weeks or months.

- **Secure by Default**
  Don't worry about what some plug-in code might do to your program. Extism is built with security as a core principle, and fully sandboxes the execution of all plug-in code.

Use-cases of a plug-in system

- Adding functionality to command-line tools
- Enabling users to "mod" a game
- Simplify "webhooks" to run event-driven logic in vendor system
- User-defined functions in a database
- No-code application extensions
- Content management system extensions

Extism Plugin SDK Demo



```
import org.extism.sdk.wasm.WasmSourceResolver;

import java.nio.file.Path;

no usages   Thomas Darimont
public class ExtismExample {

        Thomas Darimont
    public static void main(String[] args) {

        var manifest = new Manifest(new WasmSourceResolver().resolve(Path.of( first: "code.wasm")));

        try (var ctx = new Context(); //
             var plugin = ctx.newPlugin(manifest,  withWASI: false)) {

            var output = plugin.call( functionName: "count_vowels",  input: "Hello World");
            System.out.println(output);
        }

    }
}
```

Summary

- Web Assembly support in Java is currently still in its infancy
- Usage of Web Assembly from Java *currently* VERY low-level
- Enables robust extensibility for existing programs
- Better developer experience will be a game changer
- Web Assembly has potential beyond the browser!

Nächste Veranstaltungen

- ✓ 07 Mar **Web Assembly für Java Entwickler** **Thomas & Florian**
- 14 Mar **Immutable Objects meet mutable Force** **Lombok Guys**
- XX Apr **Awesome Talk TBA**
- XX Mai **Awesome Talk TBA**
- XX Jun **Awesome Talk TBA**
- XX Jul **Awesome Talk TBA**
- XX Aug **Awesome Talk TBA**
- XX Sep **Awesome Talk TBA**
- XX Okt **Awesome Talk TBA**
- XX Nov **Awesome Talk TBA**

http://www.meetup.com/de-DE/Java-User-Group-Saarland-jugsaar/#upcoming