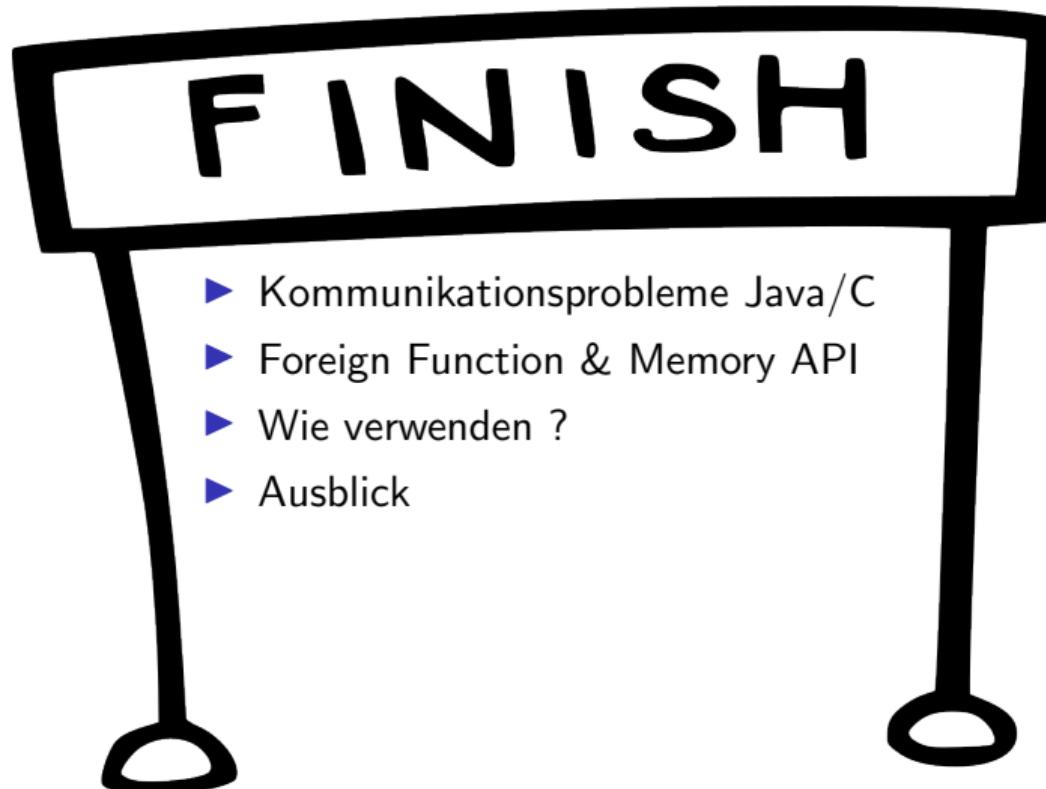


# Javas neue Gesprächskultur – ganz wie in Panama –

Bernd Müller  
Ostfalia





## Vorstellung Referent

- ▶ Prof. Informatik (Ostfalia, HS Braunschweig/Wolfenbüttel)
- ▶ Buchautor (JSF, JPA, Seam, ...)

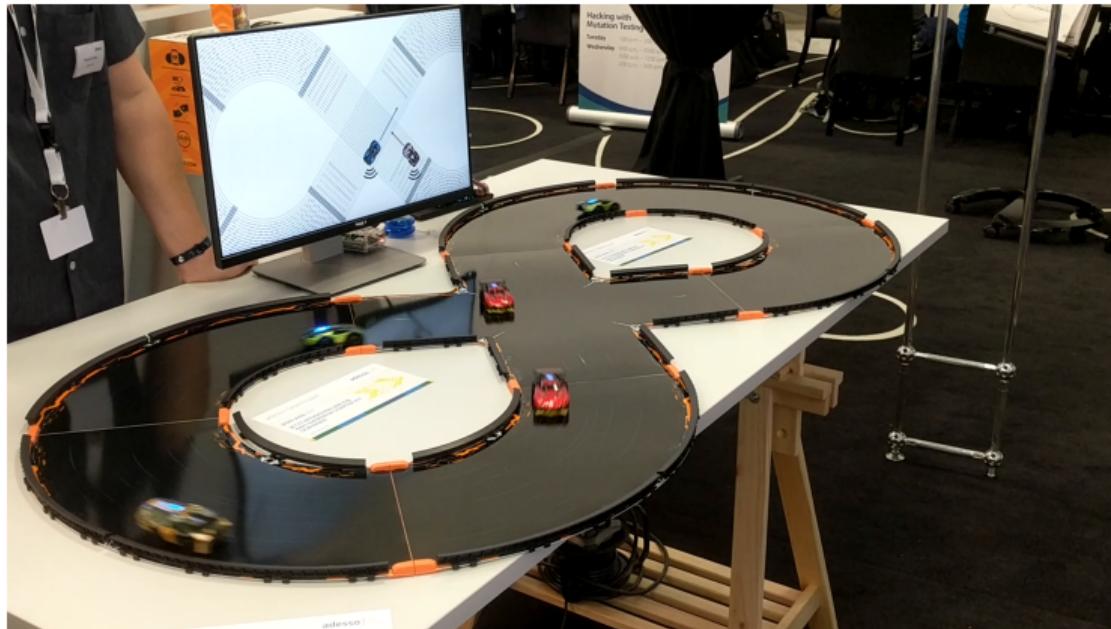


- ▶ Mitglied EGs JSR 344 (JSF 2.2) und JSR 338 (JPA 2.1)
- ▶ Geschäftsführer PMST GmbH
- ▶ JUG Ostfalen (Mitorganisator)
- ▶ Beirat der Java aktuell
- ▶ bernd.mueller@ostfalia.de
- ▶ @berndmuller    @berndmuller@fosstodon.org

 BerndMuller

## Motivation

## JavaLand 2017, Adesso-Stand



## GitHub: adessoAG/anki-drive-java

The screenshot shows the GitHub repository page for `adessoAG/anki-drive-java`. The repository is public and has 16 stars. It contains 1 branch and 1 tag. The commit history is listed, with the first commit being the initial commit. A large red oval highlights the first commit message: "Unfortunately, there is currently no cross-platform Java library to interface with Bluetooth LE devices. This project therefore requires a Node.js gateway service to handle low-level communication with the Anki vehicles. All data processing and message parsing is carried out in Java code." The repository is described as a "Java port of the Anki Drive SDK". It includes tags for `java`, `vehicle`, `java-library`, `anki`, `anki-drive`, `drive-sdk`, and `anki-vehicles`. It also lists a `Readme`, `MIT License`, 16 stars, 7 watching, and 27 forks. There is one release, the "First official release" from April 2017.

A Java port of the Anki Drive SDK

java vehicle java-library anki  
anki-drive drive-sdk anki-vehicles

Readme

MIT License

16 stars

7 watching

27 forks

Releases 1

First official release Latest  
on 4 Apr 2017

Commit	Message	Date
gradle/wrapper	Support automated builds via Travis CI (#10)	5 years ago
src/main		5 years ago
.gitignore		5 years ago
yeckey	Support automated builds via Travis CI (#10)	5 years ago
gradlew	Fix link in README (#16)	5 years ago
gradlew.bat	Initial commit	5 years ago
gradlew	Support automated builds via Travis CI (#19)	5 years ago
gradlew.bat	Initial commit	5 years ago
nakane ison	Initial commit	5 years ago

## JSR 82: Review Ballot 2000, FR3 2010

The screenshot shows the Java Community Process website with a red oval highlighting the central content area. The main navigation menu includes links for JSRs, JCP Info, and various community resources. The highlighted section displays the title "JSR-000082 Java(TM) ME for Bluetooth 1.1.1 Final Release 3" and provides download instructions. A table lists the download file, which is "JSR-000082\_Bluetooth 1.1.1 Final Release 3\_bluetooth-1.1.1-mrel2-javadoc.zip" with a size of 480.28 KB.

File Description and Name	Size
JSR-000082_Bluetooth 1.1.1 Final Release 3_bluetooth-1.1.1-mrel2-javadoc.zip	480.28 KB

If you need assistance with downloads, please contact Customer Service. For all other JCP related questions, please see our Frequently Asked Questions (FAQ).

## JAnki

The screenshot shows a GitHub repository page for the user BerndMuller named "JAnki". The repository is public and has 33 commits. A red circle highlights the "About" section, which contains the description: "A simple library to use Anki Overdrive with Java".

**About**

A simple library to use Anki Overdrive with Java

**Code**

master · 1 branch · 0 tags

Go to file · Add file · Code

BerndMuller · Minor cleanups · d440d00 · 13 Jan 2021 · 33 commits

File	Description	Time Ago
lib	Minor cleanups	13 months ago
src	Minor cleanups	13 months ago
.gitignore	notifications refactored to package notifications, added C...	4 years ago
LICENSE.txt	license added	4 years ago
README.md	license added	4 years ago
backlog.txt	Minor cleanups	13 months ago
pom.xml	Minor cleanups	13 months ago
run-cli.sh	tracking added	4 years ago
run-tracking.sh	tracking added	4 years ago

**Readme**

**MIT License**

**4 stars**

**3 watching**

**0 forks**

**Releases**

No releases published  
Create a new release

**Packages**

## TinyB von Intel

The screenshot shows the GitHub repository page for `intel-iot-devkit / tinyb`. The repository is public and has 33 watchers, 105 forks, and 229 stars. The main navigation tabs include Code, Issues (82), Pull requests (4), Actions, Projects, Wiki, Security, and three more options. Below the tabs, there's a dropdown for the branch (master) and buttons for Go to file, Add file, and Code. A red oval highlights the **About** section, which contains a brief description of the project: "TinyB exposes the BLE GATT API for C++, Java and other languages, using BlueZ over DBus." To the right of the About section is a sidebar with links to Readme, MIT License, stars (229), watching (33), forks (105), and a Release section. Another red oval highlights the **Releases** section, which shows one release: v0.5.1 (Latest), released on 19 Jan 2018.

Code

Issues 82

Pull requests 4

Actions

Projects

Wiki

Security

...

master

Go to file

Add file

Code

About

vkołotov and petreetime Implementing discovery filter... on 14 Oct 2017 131

api Implementing a generic method to set discover fi... 4 years ago

cmake Add additional version information, including in j... 6 years ago

examples examples: list\_mfg also lists advertised service ... 5 years ago

include Adding support for setting RSSI discovery filter 4 years ago

java Implementing discovery filter by UUIDs (java) an... 4 years ago

src Implementing discovery filter by UUIDs (java) an... 4 years ago

.gitignore .gitignore: Now all build\* directories are excluded 6 years ago

.travis.yml Add checkinit which tests if library loads ok and ... 6 years ago

TinyB exposes the BLE GATT API for C++, Java and other languages, using BlueZ over DBus.

Readme

MIT License

229 stars

33 watching

105 forks

Releases 1

v0.5.1 Latest

on 19 Jan 2018

Selbst ist der Mann  
und das Ergebnis ist immer ein Erlebnis ;-)



Ziel: Keine Bibliothek, nur JDK + BlueZ

## Verallgemeinerung

Keine Bibliothek, nur JDK + <your choice>

## Das Problem

## Das Problem – und hoffentlich die Lösung

- ▶ Manchmal genügt JDBC, HTTP, NIO, UNIX-Domain-Sockets, ... nicht
- ▶ Sogenannter *Off-Heap-Speicher* (außerhalb JVM) soll direkt zugegriffen werden
- ▶ Machen z.B. Tensorflow, Ignite, Lucene, Netty und vielen anderen
- ▶ Es gibt: JNI, Byte-Buffer-API, sun.misc.Unsafe, ...
- ▶ Macht aber auf Dauer nicht glücklich ...
- ▶ JEPs 191, 370, 383, 389, 393, 412, 419, 424, 434
- ▶ Mit **Project Panama: Interconnecting JVM and native code**

## JEP 419: Foreign Function & Memory API



OpenJDK FAQ  
Installing  
Contributing  
...  
...

### JEP 419: Foreign Function & Memory API (Second Incubator)

Owner Maurizio Cimadamore

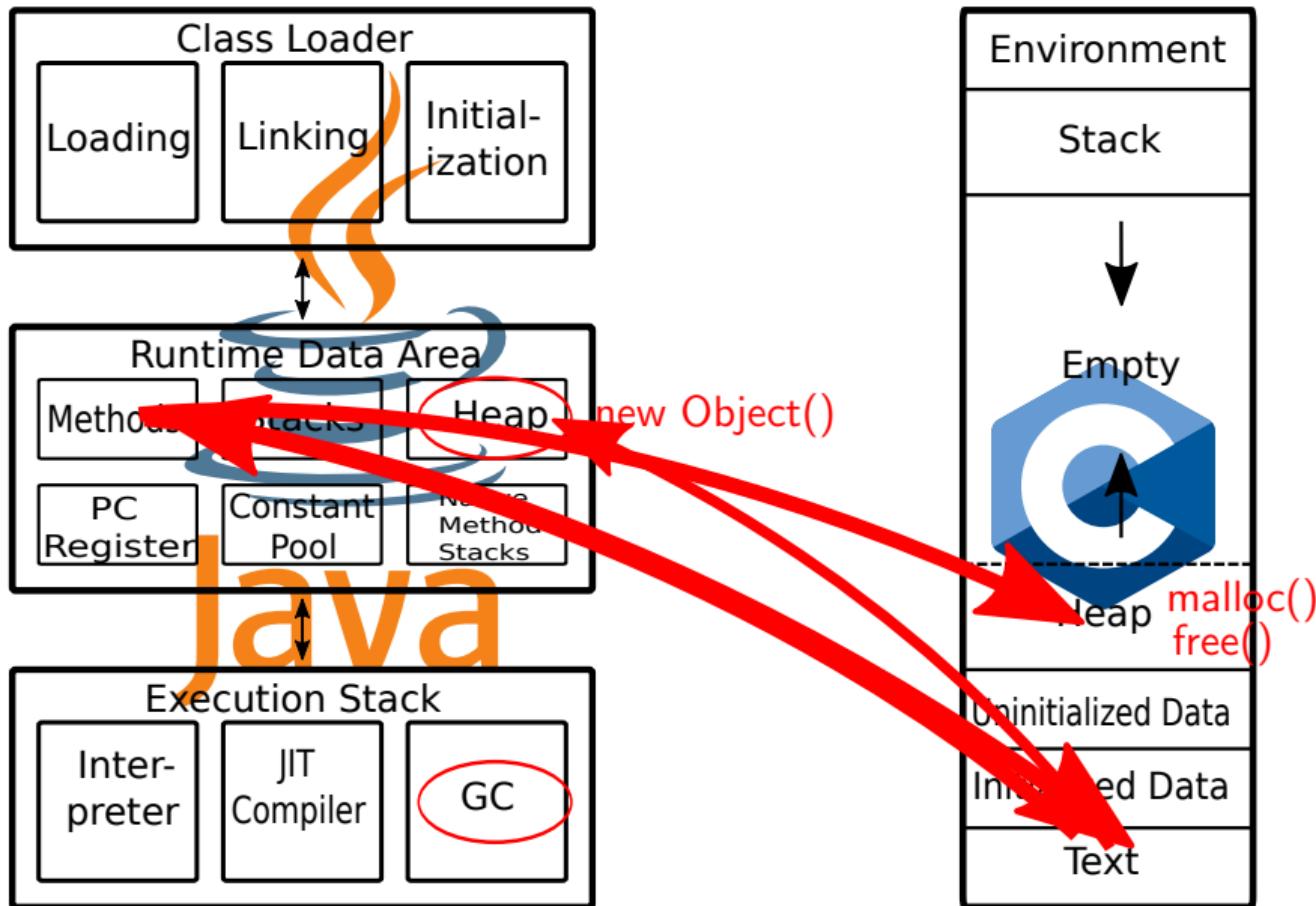
#### Summary

Introduce an API by which Java programs can interoperate with code and data outside of the Java runtime. By efficiently invoking foreign functions (i.e., code outside the JVM), and by safely accessing foreign memory (i.e., memory not managed by the JVM), the API enables Java programs to call native libraries and process native data without the brittleness and danger of JNI.

Groups  
(overview)  
Adoption  
Build  
Client Libraries  
Compatibility & Specification Review  
Compiler  
Conformance  
Core Libraries  
Governance Board

#### Summary

Introduce an API by which Java programs can interoperate with code and data outside of the Java runtime. By efficiently invoking foreign functions (i.e., code outside the JVM), and by safely accessing foreign memory (i.e., memory not managed by the JVM), the API enables Java programs to call native libraries and process native data without the brittleness and danger of JNI.



Native Method Interface + Libraries

## Lösungsversuch FFM: Konzepte und Klassen (Java 20)

- ▶ Allokation von Speicher  
MemorySegment, SegmentAllocator, ...
- ▶ Lesen und Schreiben von (strukturiertem) Speicher  
MemoryLayout, VarHandle (java.lang.invoke), ...
- ▶ Regeln des Lebenszyklus von Ressourcen  
SegmentScope
- ▶ Finden und Aufruf von Funktionen  
Linker

## Beispiel Memory Segment

- ▶ Zusammenhängender Speicherbereich
- ▶ On- oder Off-Heap
- ▶ Nativ, mapped (mmap), Array oder Buffer
- ▶ Garantien für räumliche, zeitliche sowie Thread-bezogene Beschränkungen
- ▶ ...

## Beispiele

## Prozess-Id (C)

Prozess-Id des Prozesses: keine Parameter, Integer-Rückgabe (jps-Clone)

```
#include <unistd.h>

pid_t getpid(void);
```

## Prozess-Id (Java)

```
MethodHandle getpid = Linker.nativeLinker().downcallHandle(  
    Linker.nativeLinker().defaultLookup().find("getpid").get(),  
    FunctionDescriptor.of(JAVA_INT));  
  
System.out.println("Process Id: " + (int) getpid.invokeExact());
```

## Quicksort

```
#include <stdlib.h>

void qsort(void *base, size_t nmemb, size_t size,
           int (*compar)(const void *, const void *));
```

## Vergleichsmethode/-klasse (à la Comparator#compare())

```
static class IntComparator {  
  
    static int compare(MemorySegment addr1, MemorySegment addr2) {  
        return addr1.get(JAVA_INT, 0) - addr2.get(JAVA_INT, 0);  
    }  
  
}
```

## Method-Handle auf qsort

```
MethodHandle qsort = Linker.nativeLinker().downcallHandle(  
    Linker.nativeLinker().defaultLookup().find("qsort").get(),  
    FunctionDescriptor.ofVoid(ADDRESS, JAVA_LONG, JAVA_LONG, ADDRESS)  
) ;
```

## Method-Handle auf Vergleichsmethode

```
FunctionDescriptor compareDescriptor =
    FunctionDescriptor.of(JAVA_INT, ADDRESS.asUnbounded(),
                          ADDRESS.asUnbounded());  
  
MethodHandle compareHandle =
    MethodHandles.lookup().findStatic(IntComparator.class,
                                       "compare",
                                       compareDescriptor.toMethodType());
```

## Allokation des Arrays und eigentlicher Aufruf

```
try (Arena arena = Arena.openConfined()) {
    MemorySegment comparFunc = Linker.nativeLinker().upcallStub(
        compareHandle, compareDescriptor, arena.scope());

    MemorySegment array = arena.allocateArray(JAVA_INT,
        new int[] { 0, 9, 3, 4, 6, 5, 1, 8, 2, 7 });

    qsort.invokeExact(array, 10L, 4L, comparFunc);

    int[] sorted = array.toArray(JAVA_INT);

    ...
}
```

## Demo Time



Sleepy from slides, we are !

## Werkzeugunterstützung

## JExtract

- ▶ Reduktion des Overheads durch
  - ▶ Analyse einer Include-Datei
  - ▶ und Generierung der entsprechenden Methoden und Typen für den Aufruf
- ▶ Nicht im JDK, evtl. später separat verfügbar

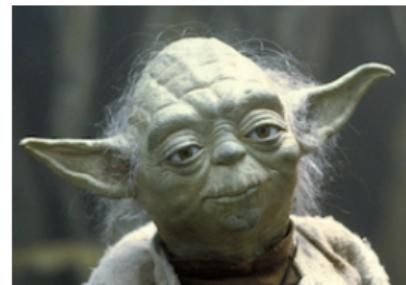
## getpid ohne jextract

```
MethodHandle getpid = Linker.nativeLinker().downcallHandle(  
    Linker.nativeLinker().defaultLookup().find("getpid").get(),  
    FunctionDescriptor.of(JAVA_INT));  
  
System.out.println("Process Id: " + (int) getpid.invokeExact());
```

## getpid mit jextract

```
System.out.println("Process Id: " + getpid());
```

## Demo Time



Sleepy from slides, we are !

Es entwickelt sich ... immer noch ...

## Quicksort mit Java 17 (Ausschnitt)

```
MethodHandle qsort = CLinker.getInstance().downcallHandle(
    CLinker.systemLookup().lookup("qsort").get(),
    MethodType.methodType(void.class, MemoryAddress.class,
                          long.class, long.class, MemoryAddress.class),
    FunctionDescriptor.ofVoid(C_POINTER, C_LONG, C_LONG, C_POINTER)
);
```

## Quicksort mit Java 18 (Ausschnitt)

```
MethodHandle qsort = CLinker.systemCLinker().downcallHandle(  
    CLinker.systemCLinker().lookup("qsort").get(),  
    FunctionDescriptor.ofVoid(ADDRESS, JAVA_LONG, JAVA_LONG, ADDRESS)  
) ;
```

## Quicksort mit Java 19 (Ausschnitt)

```
MethodHandle qsort = Linker.nativeLinker().downcallHandle(  
    Linker.nativeLinker().defaultLookup().lookup("qsort").get(),  
    FunctionDescriptor.ofVoid(ADDRESS, JAVA_LONG, JAVA_LONG, ADDRESS)  
) ;
```

## Quicksort mit Java 20 (Ausschnitt)

```
MethodHandle qsort = Linker.nativeLinker().downcallHandle(  
    Linker.nativeLinker().defaultLookup().find("qsort").get(),  
    FunctionDescriptor.ofVoid(ADDRESS, JAVA_LONG, JAVA_LONG, ADDRESS)  
) ;
```

## Fazit und Ausblick

## Persönliche – und damit subjektive – Einschätzung

- ▶ Eine wirklich gute Idee
- ▶ Mit Java 19: von Inkubator nach Preview. Evtl. GA mit Java 21?
- ▶ Versuche JAnki zu migrieren. Schwierigkeit: Komplexität des D-Bus-Interface und eigene Unzulänglichkeiten ;-)
- ▶ Dominik Martens hat es geschafft (in Master-Arbeit)
- ▶ Wenn GraalVMs Native-Image-Erzeugung FFM unterstützt: 😊
- ▶ und was machen andere ?

## Tomcat SSL

Tomcat 9.0.55 (remm)

2021-11-10

Coyote

① Improve performance of Connector shutdown - primarily to reduce the time it takes to run the test suite. (markt)

🔧 Refactor the APR/native connector shutdown to reduce the possibility of a JVM crash during the connector shutdown. (markt)

🔧 [#457](#): Add a `toString()` method to `MimeHeader` to aid debugging. (dblevins)

🔧 Add experimental OpenSSL support through the Panama API incubating in Java 17, with support for OpenSSL 1.1. This no longer requires tomcat-native or APR. Please refer to the `openssl-java17` module from the `main` branch for more details. (remm)

🔧 Fix APR connector stop so it correctly waits for the sendfile thread, if any, to exit. (markt)

🔧 Do not ignore the error condition if the APR connector is not able to open a server socket as continuing in this case will trigger a JVM crash. (markt)

## Elasticsearch

### Some scattered feedback on the Foreign Linker API

Chris Hegarty [chegar999@gmail.com](mailto:chegar999@gmail.com)

Wed Dec 22 09:19:40 UTC 2021

- Previous message (by thread): [Calling the Port Audio C API via Panama FFI APIs](#)
- Next message (by thread): [Some scattered feedback on the Foreign Linker API](#)
- **Messages sorted by:** [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#)

---

Hi,

As part of a recent prototyping effort, we evaluated replacing the usage of JNA in the core of the Elasticsearch server with the incubating Foreign Linker API.

TL;DR things just worked, perf improved and we really like the restricted native access.

Our usage (at least in this particular case) is not really performance sensitive, in that we reach into native to setup syscall filtering and check resource limits - but hey, faster is always better! We measured approximate perf improvements between 8 and 20 times faster for simple downcalls. The reason the numbers vary so much is more to do with JNA rather than Panama since the Panama times from our to our tests

## Fragen und Anmerkungen



## Vortrag und Code

<https://github.com/BerndMuller/panama-javaland-2023>

