Fully Multiplatform Pure Java Development for Desktop, Web, Android and iOS

Kirill Prazdnikov

Huawei

@Arkanoid

pkirill+jp@gmail.com

JPoint LOGO

Kirill Prazdnikov

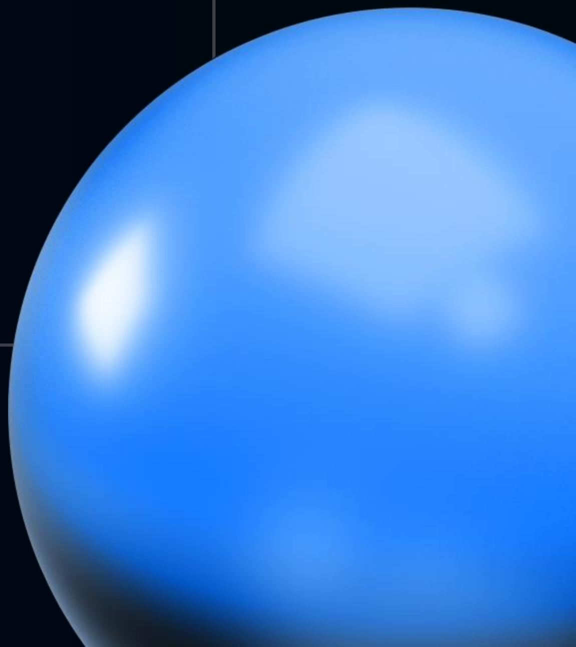
@Arkanoid

Bio

- Game development C++
- JavaFX development C++/Java
- Delightex – multiplatform Java/C++/ObjC/JavaScript
- Huawei

Outline of the talk

1. Hello world demo: desktop + web (kirillp.github.io)
2. Overview of technologies for portable development
3. Reasons for Java multiplatform
4. Rich examples
5. Commons & Differences between platforms
6. Project modules and structure
7. JNI on web and on iPhone
8. Performance
9. Alternatives overview.
10. Link to Github project
11. Q&A.



Demo time

http://kirillp.github.io

Reasons for Java multiplatform

- Reduces technology diversity
- Reduces codebase
- Requires less people: usually, classic app development process requires dedicated engineers to work on different platforms
- Improves development workflow

Overview of technologies for portable Java compilation

- Google GWT was pioneer here — it is a transpiler
- Transpiler —text to text: converts Java to JS
- Problems
    - you have to distribute sources in JARs
    - does not support Kotlin

Java AOT compilation

- AOT: Closed World model — no dynamic class loading
- TeaVM — Java / Kotlin (bytecode) AOT compiler

  targets: JavaScript, WASM, C
- GraalVM — Java / Kotlin (bytecode) AOT to

  Native exe or dll (.dylib, .so)

  Apple arm targets

Rich examples

- Delightex CoSpaces EDU — in prod for over 5 years
  - Educational app with rich 3D Graphics, Lighting, Physics and Scripting
  - 22k Classes, 145k methods
- Huawei SuduEditor – in development
  - A text editor using ANTLR parsing engine for code highlighting
  - 600 + 500 Classes (two threads), 10K methods
  - https://github.com/SuduIDE/sudu-editor

Common in all platforms

- UI and View compose
  - WebGL (angle) for composition and rendering
  - Canvas for offscreen Text and SVG shaping
  - UI events
- I/O http, file — with some limitations on WEB
- Multithreading - with some WEB specifics:
  deep copy or transferable

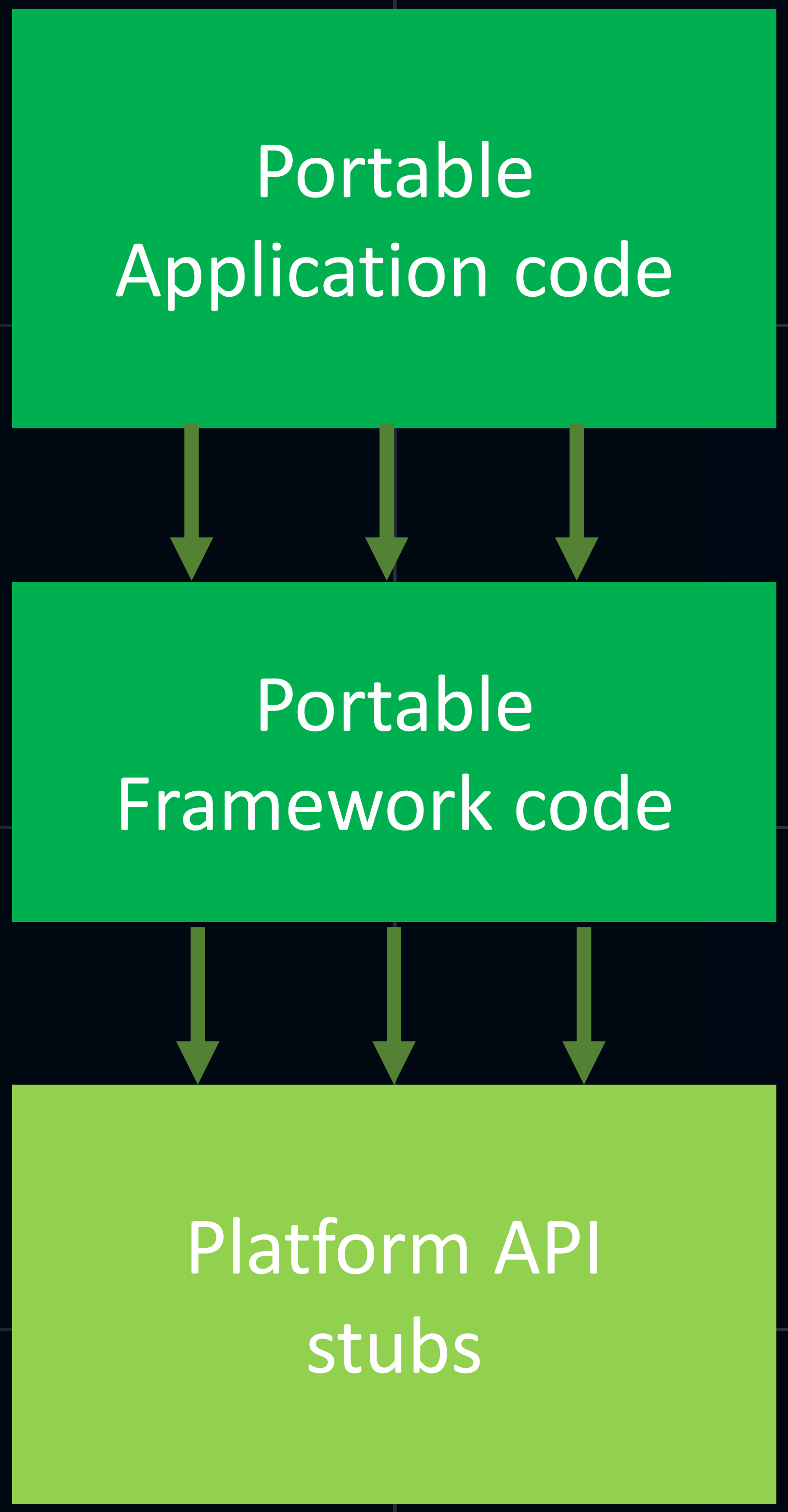
Differences between platforms

- No reflection support
  - very limited reflection capabilities on GraalVM
  - no reflection for Web in TeaVM
- no WeakHashMap on Web
- no synchronous IO API on UI thread

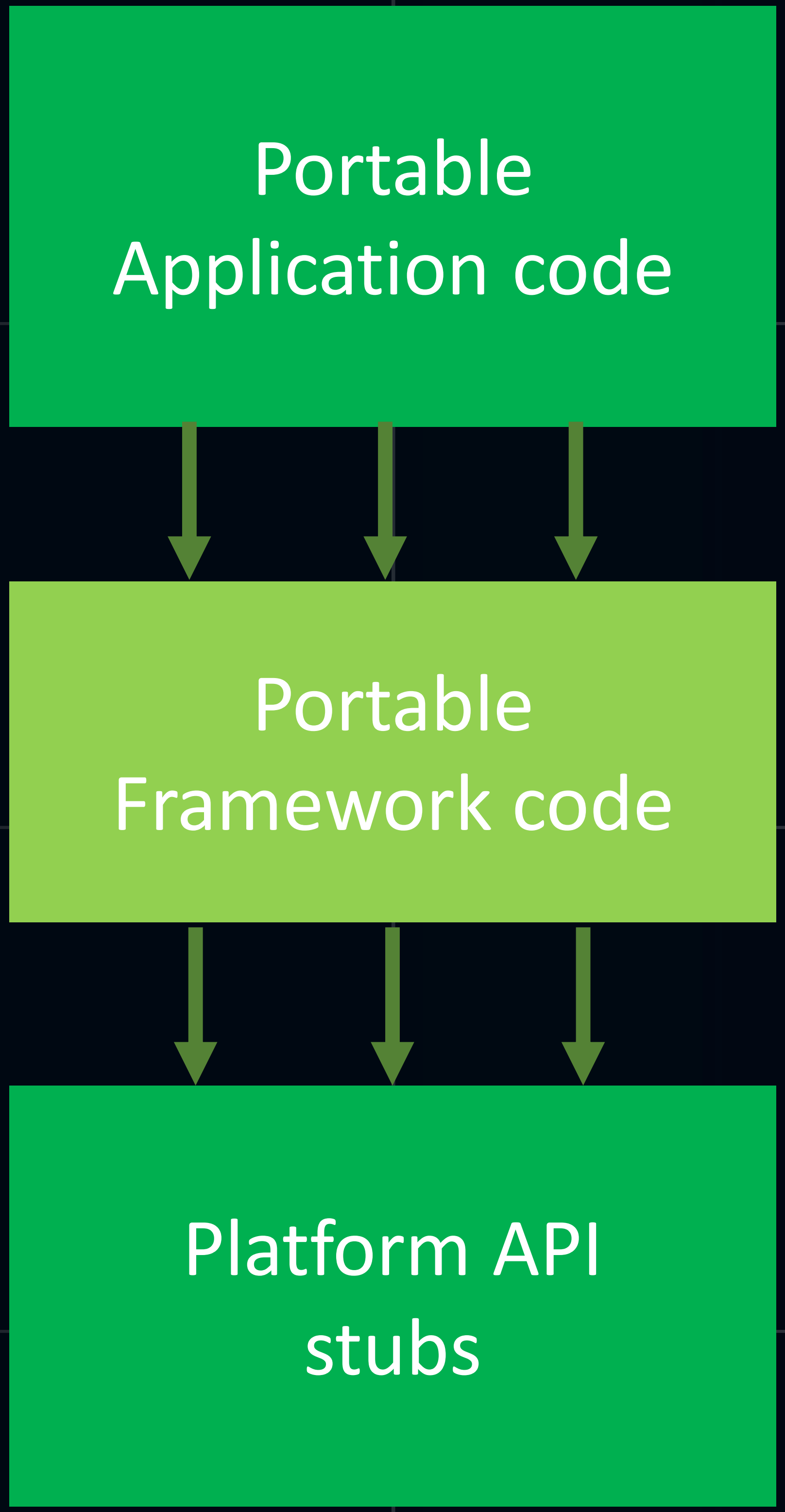
Differences between platforms

- Multithreading on Web is tricky
  - Code and classes are not shared between threads, you cannot send Runnable or a class instance
  - Messaging: deep copy is **slow,** transfer is fast **but**
  - **Sender** loses control over transferable objects:
    array.length == 0 after transfer on sender side

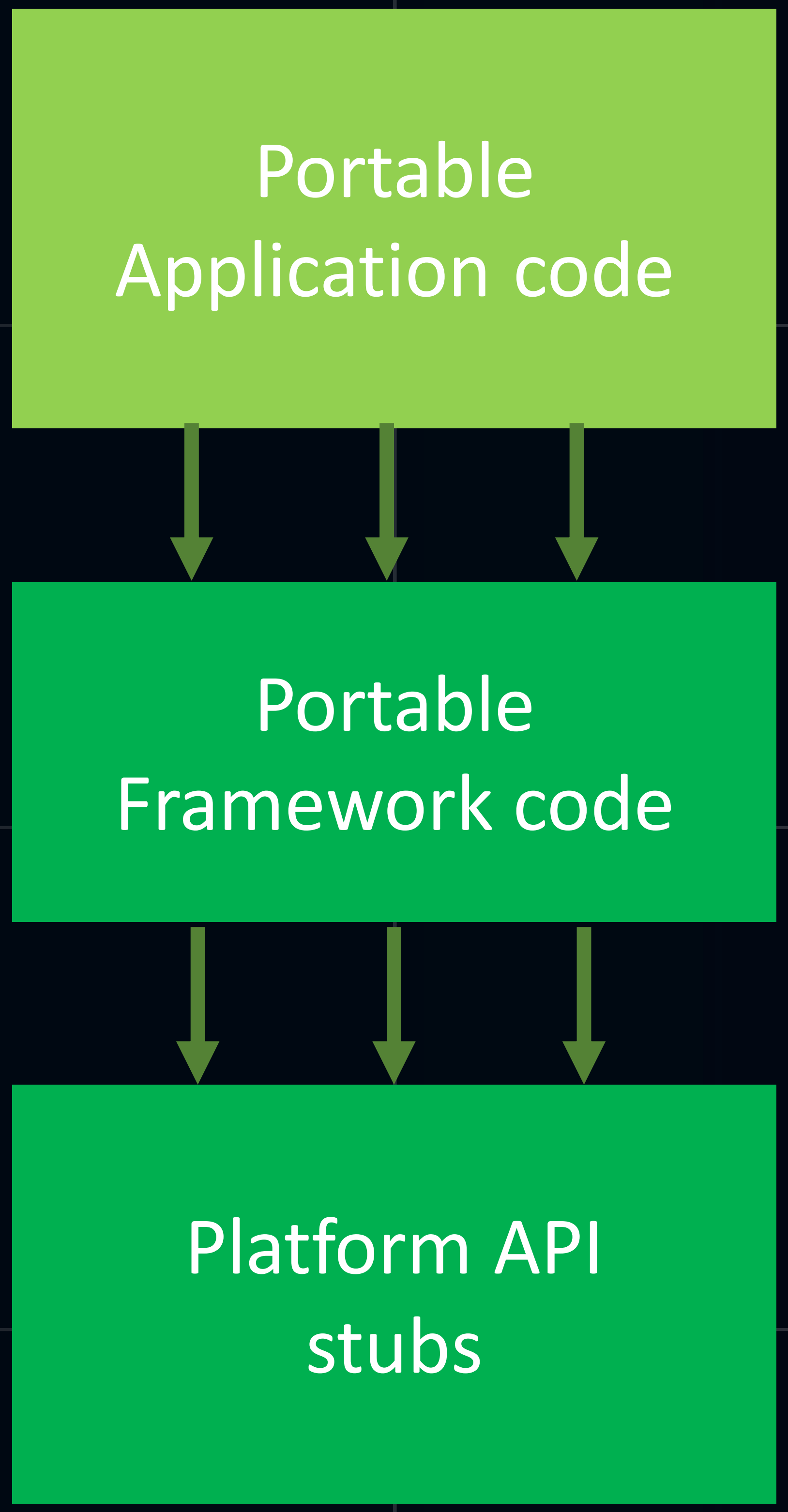
Project modules structure



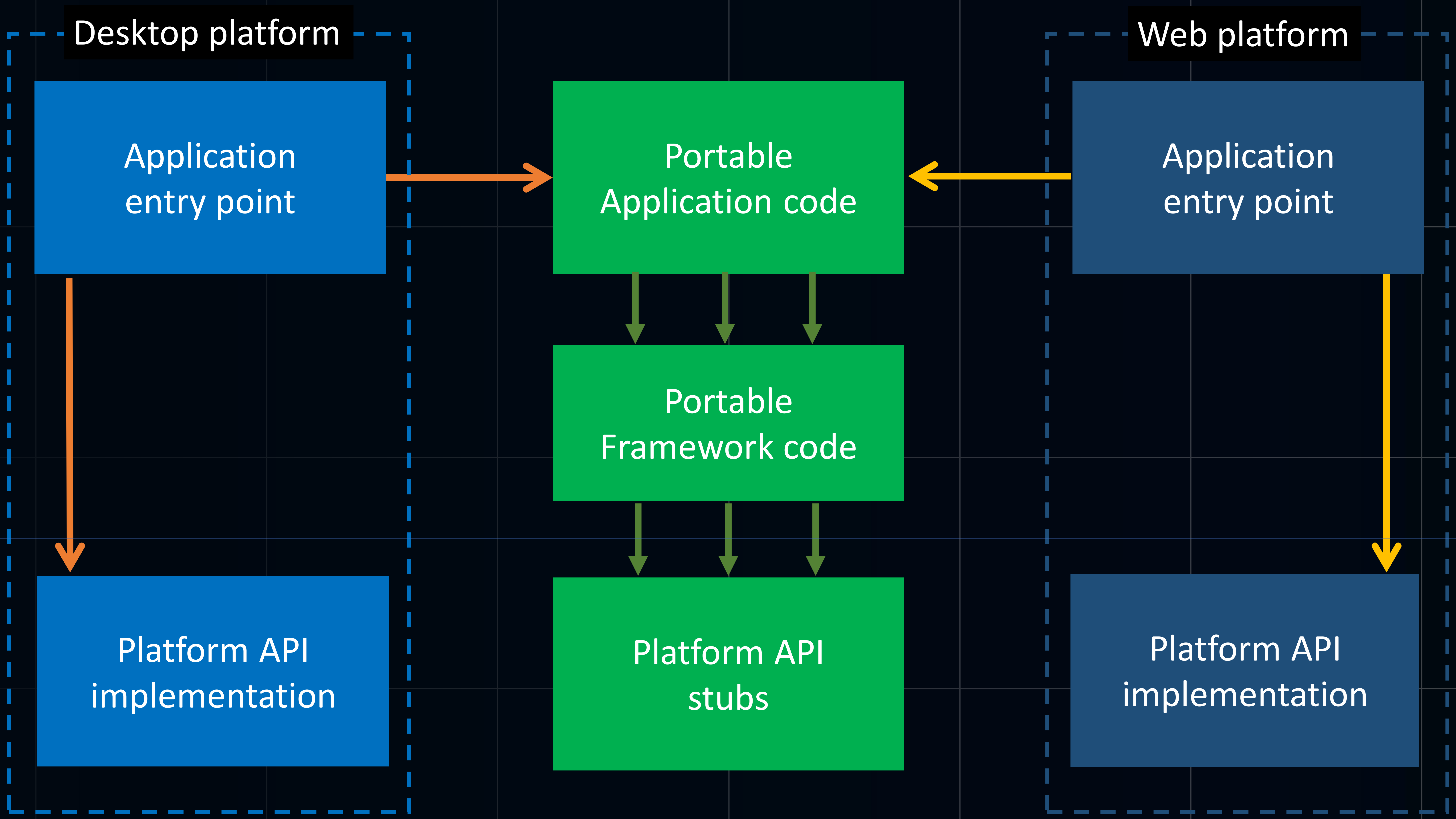
Project modules structure



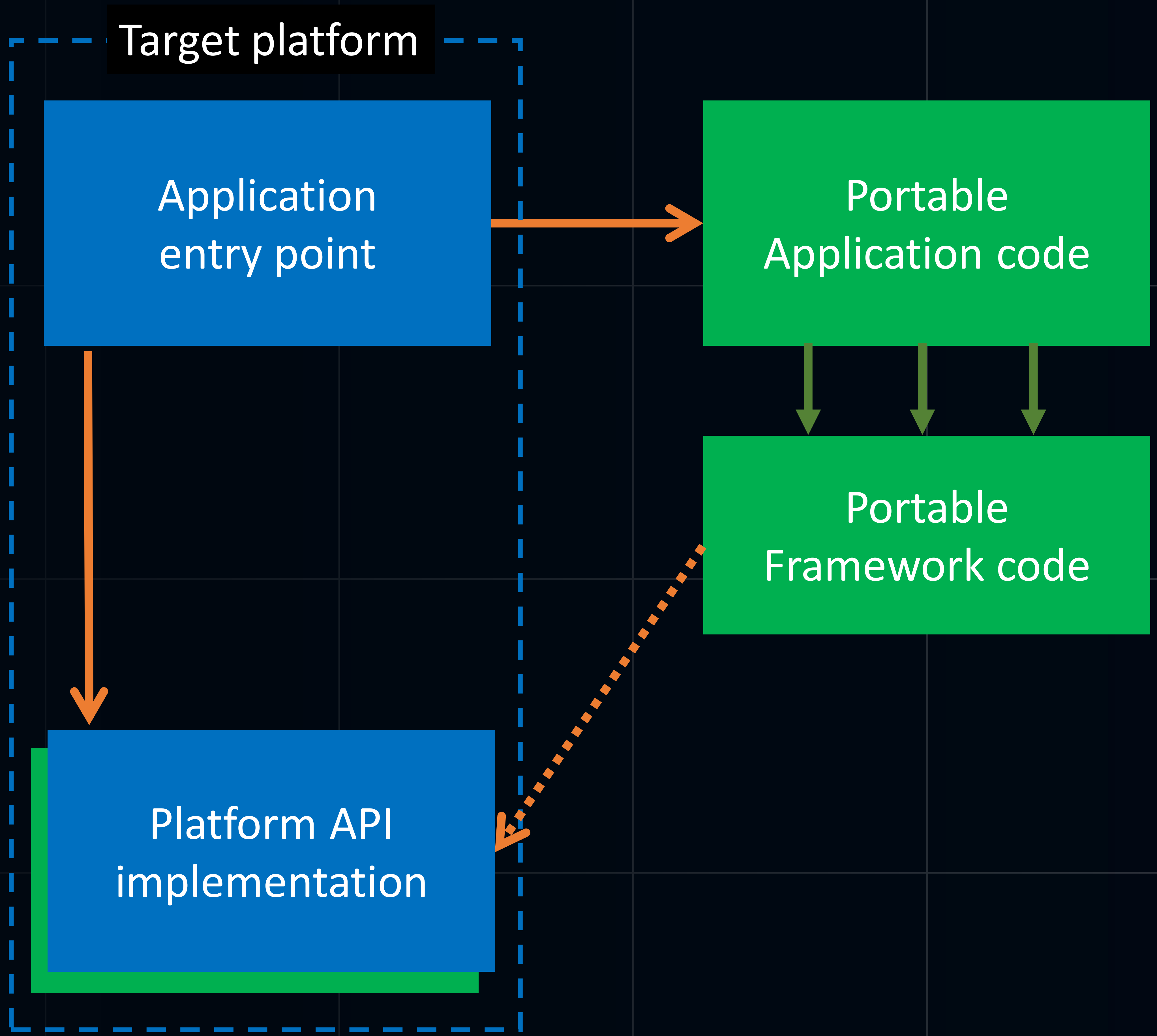
Project modules structure



Project modules structure



Linking platform app

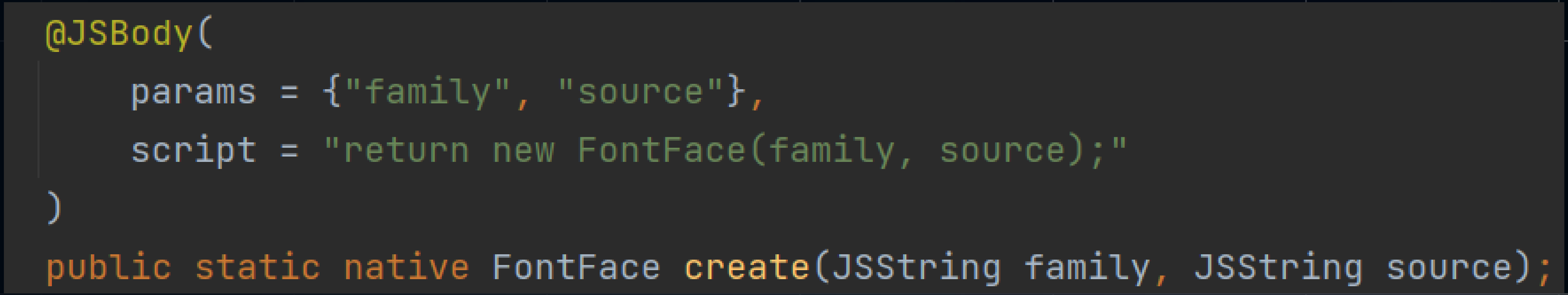


JNI and “native” with GraalVM

- GraalVM supports
    - classic JNI (slow)
    - @CFunction annotated C-functions – much faster
    - @CEntryPoint to call Java from native

JNI and “native” on web

- For Web “native” == call to JavaScript using JSObject or @JSBody



```java
@JSBody(
    params = {"family", "source"},
    script = "return new FontFace(family, source);"
)
public static native FontFace create(JSString family, JSString source);
```

- WebAssembly integration works great (C++, Rust, etc)

https://kirillp.github.io#wasm

JNI and “native” on iPhone

- On iPhone only GraalVM as JVM currently possible
  - No need to use JNI, instead we can use
    - @CFunction to call ObjC from Java
    - @CEntryPoint to call Java from ObjC
  - Objects lifetime: “Objective-C Automatic Reference Counting”
  - id objc_getClass(const char *name);
  - SEL sel_getUid(const char *str);
  - id objc_msgSend(id, SEL, ...);

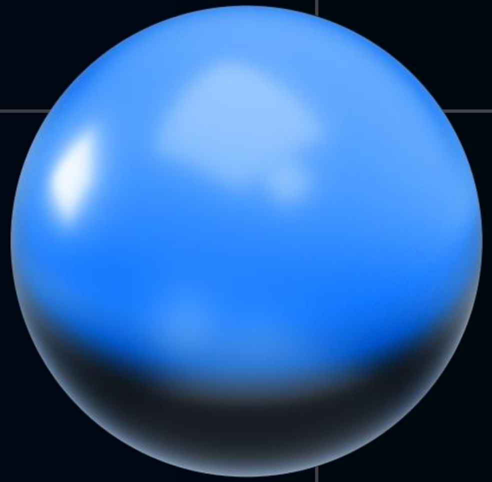
Performance

- JS is about 5x times slower on ANTLR parsing tasks
- JS is not always so slow

Alternatives overview

- GWT3
- google/j2cl
- https://github.com/mirkosertic/Bytecoder
- https://github.com/i-net-software/JWebAssembly

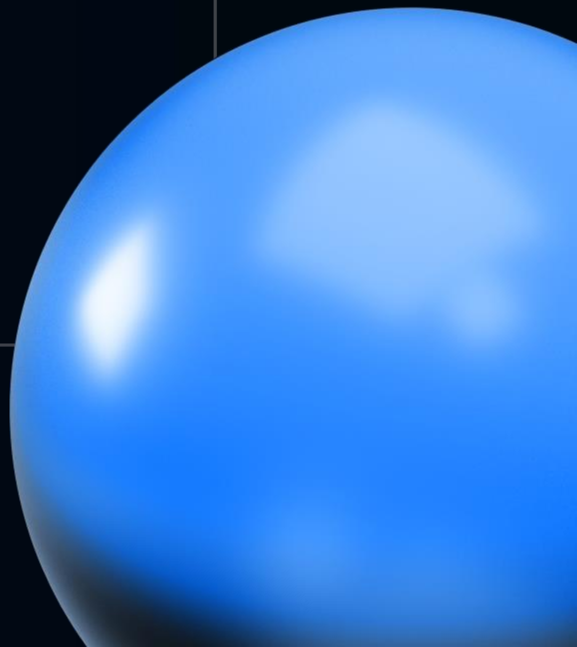
GitHub project links



Open sourced under MIT

https://github.com/SuduIDE/sudu-editor

https://kirillp.github.io



Thank you