Java(FX) on Mobile Tutorial

Tom Schindl <tom.schindl@bestsolution.at>

About Me

- CTO BestSolution.at Systemhaus GmbH
- Eclipse Committer
  - e4
  - Platform
    - EMF
- Project lead
  - e(fx)clipse

Why Java??

- Embedded is getting big
  - Home Automation
  - IoT
  - Resource Problem. Not enough C/C++ devs
- For mobile. WORA?
  - Not the right argument
  - The correct one: LOAA (Learn Once Apply Anywhere)
- OpenJDK8 available for ARM
  - JDK8 profiles help to strip down

What are problems??

- CPU
  - Problems
    - Only really necessary stuff should be run on the CPU
  - Solution
    - UIs you need to render through GPU

What are problems??

- GPU
  - Problem
    - is a lot less powerful than your Desktop GPU
  - Solution
    - use as view layers and objects as possible
    - make sure to e.g. use hard float binaries when possible

What are problems??

- Memory
  - Problems
    - Diskspace ok but not comparable to Desktop
    - RAM ok but not comparable to Desktop
  - Solution
    - JDK8-Profiles (think mobile, each app has its distinct VM)
    - Jigsaw in JDK9?

What are problems??

- No JIT on Mobile
  - Problem
    - Vendors don‘t allow to generate executable code on the fly
  - Solution
    - AOT your Java-Code

How to get on Embedded

Download OpenJDK8 for ARM
from
http://jdk8.java.net

Demo Time

How to get on Mobile

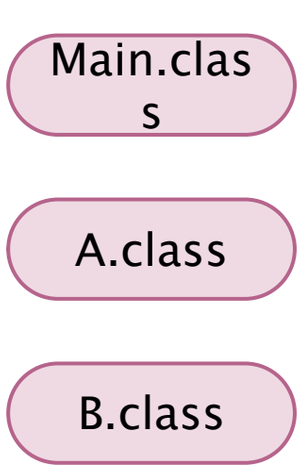
- Android
  - Dalvik-VM
    - Pros: it‘s stable and preforms well
    - Disadvantage:
      - it‘s only available for android
      - it‘s stuck in the JDK6 days of the classlibrary and language features
- iOS
  - ????

How to get on Mobile

- Introducting RoboVM for iOS
  - Makes use of Android-Classlibrary
  - Provides access to Native iOS-Frameworks including graphics
  - Is AOTed

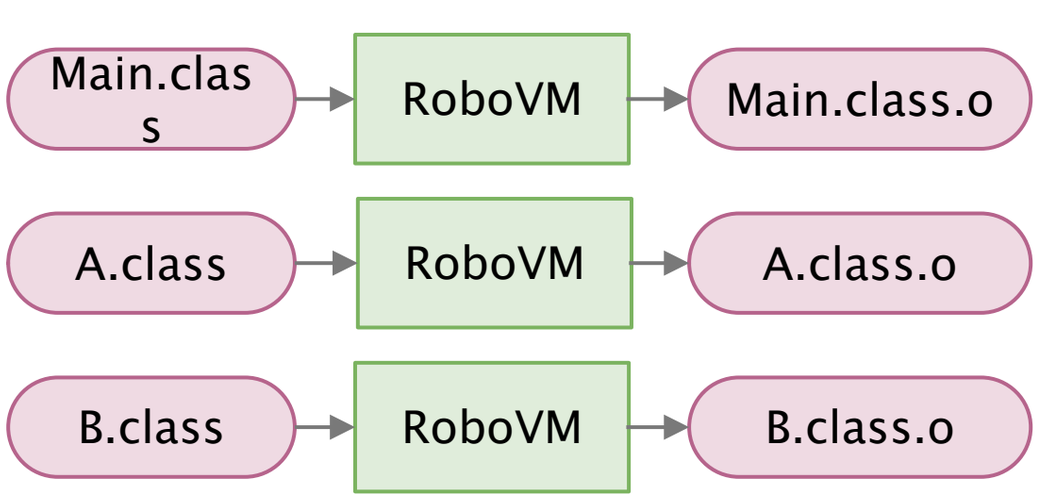
How to get on Mobile

- Introducting RoboVM for iOS
  - Makes use of Android-Classlibrary
  - Provides access to Native iOS-Frameworks including graphics
  - Is AOTed



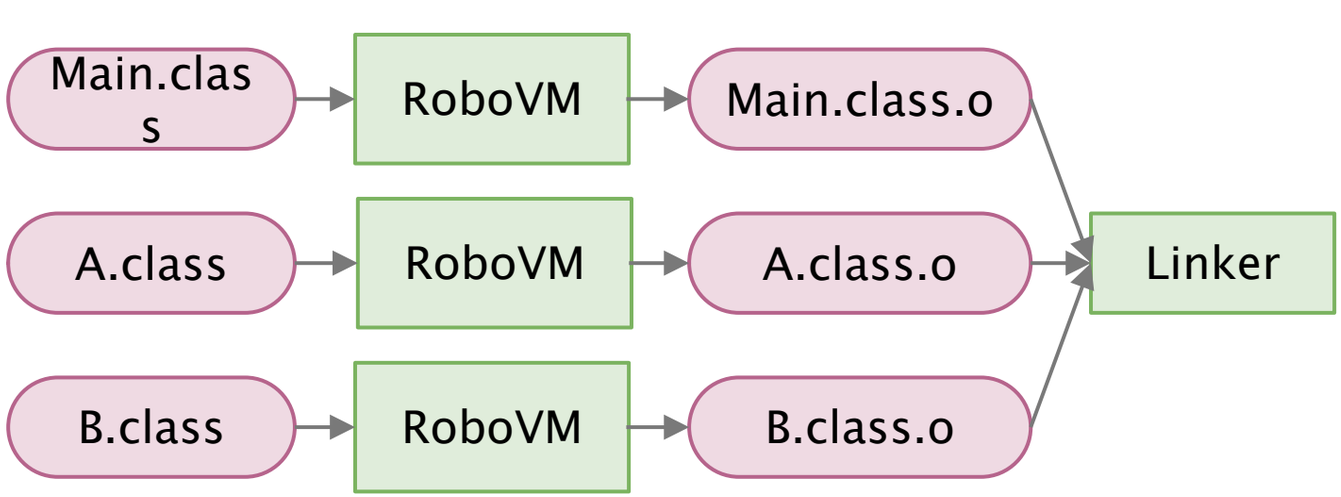
How to get on Mobile

- Introducting RoboVM for iOS
  - Makes use of Android-Classlibrary
  - Provides access to Native iOS-Frameworks including graphics
  - Is AOTed



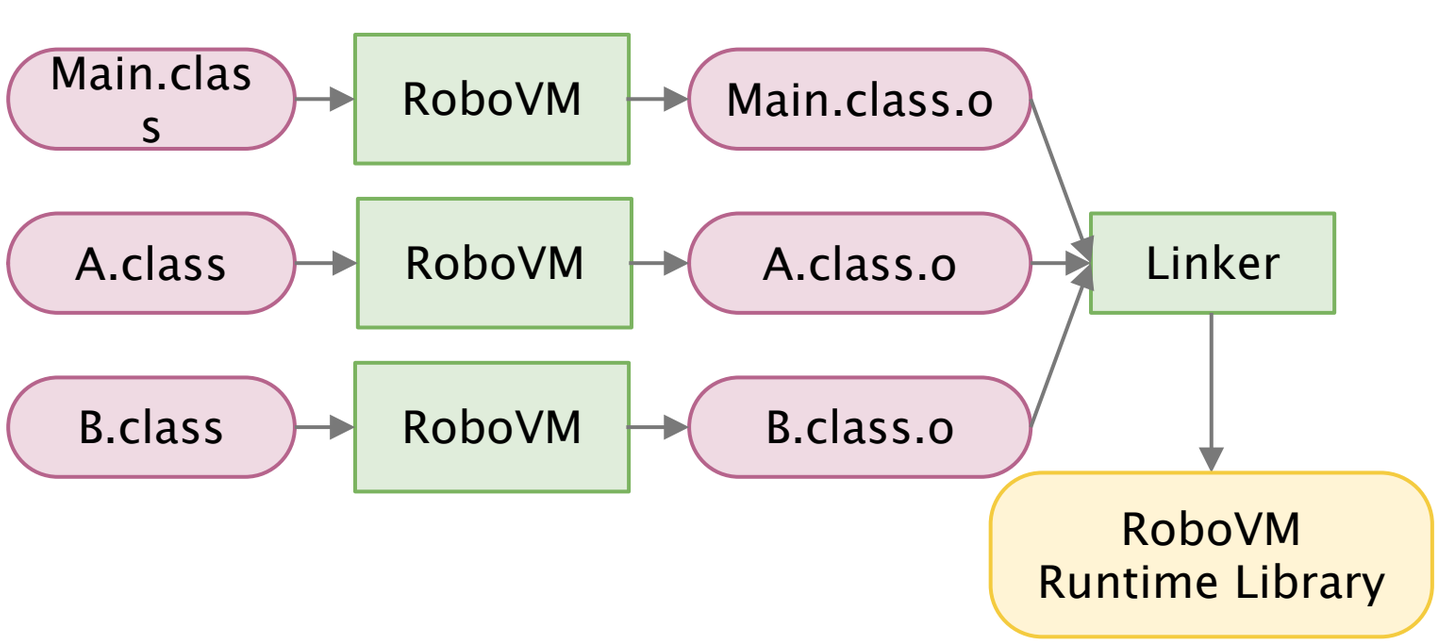
How to get on Mobile

- Introducting RoboVM for iOS
  - Makes use of Android-Classlibrary
  - Provides access to Native iOS-Frameworks including graphics
  - Is AOTed



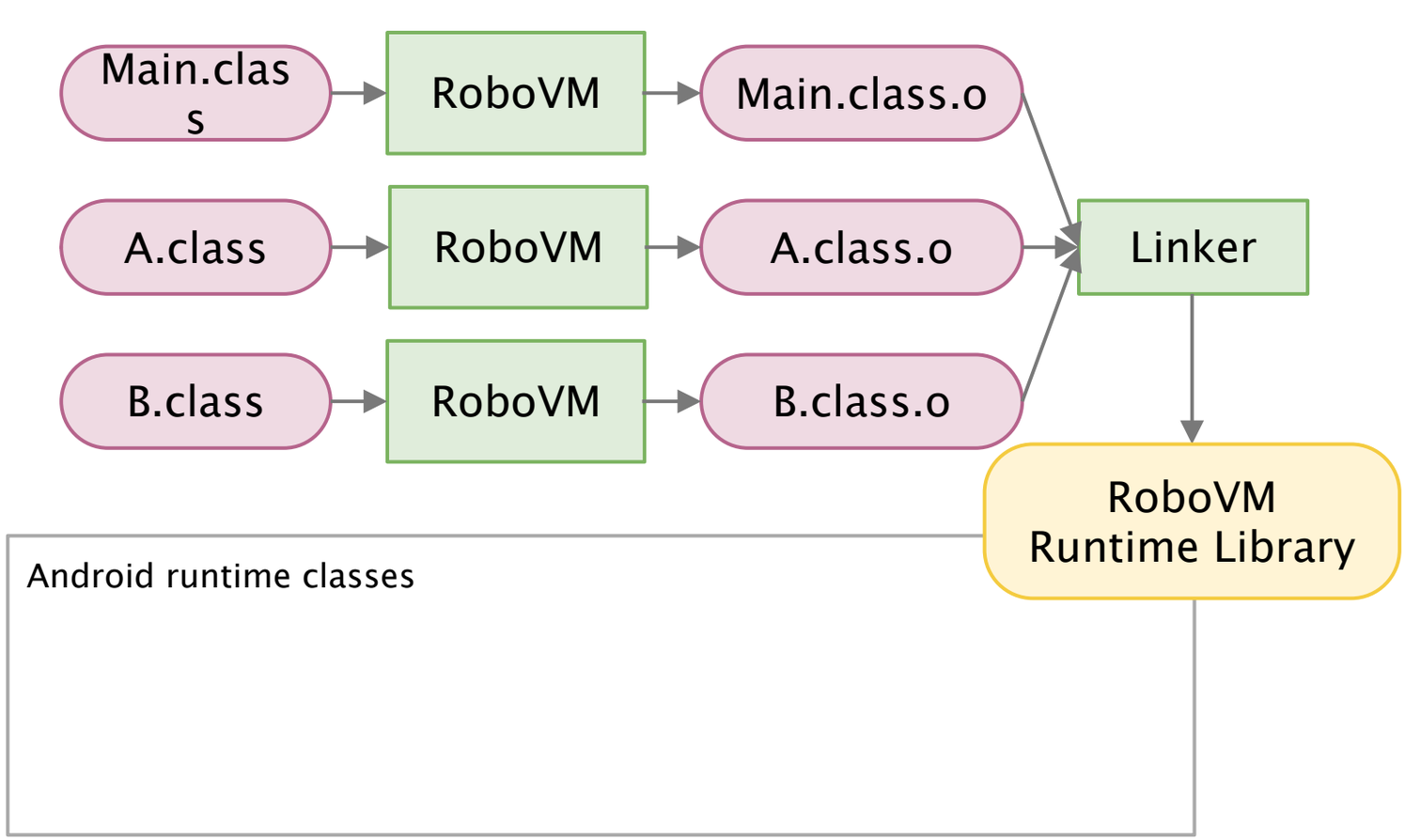
How to get on Mobile

- Introducting RoboVM for iOS
  - Makes use of Android-Classlibrary
  - Provides access to Native iOS-Frameworks including graphics
  - Is AOTed



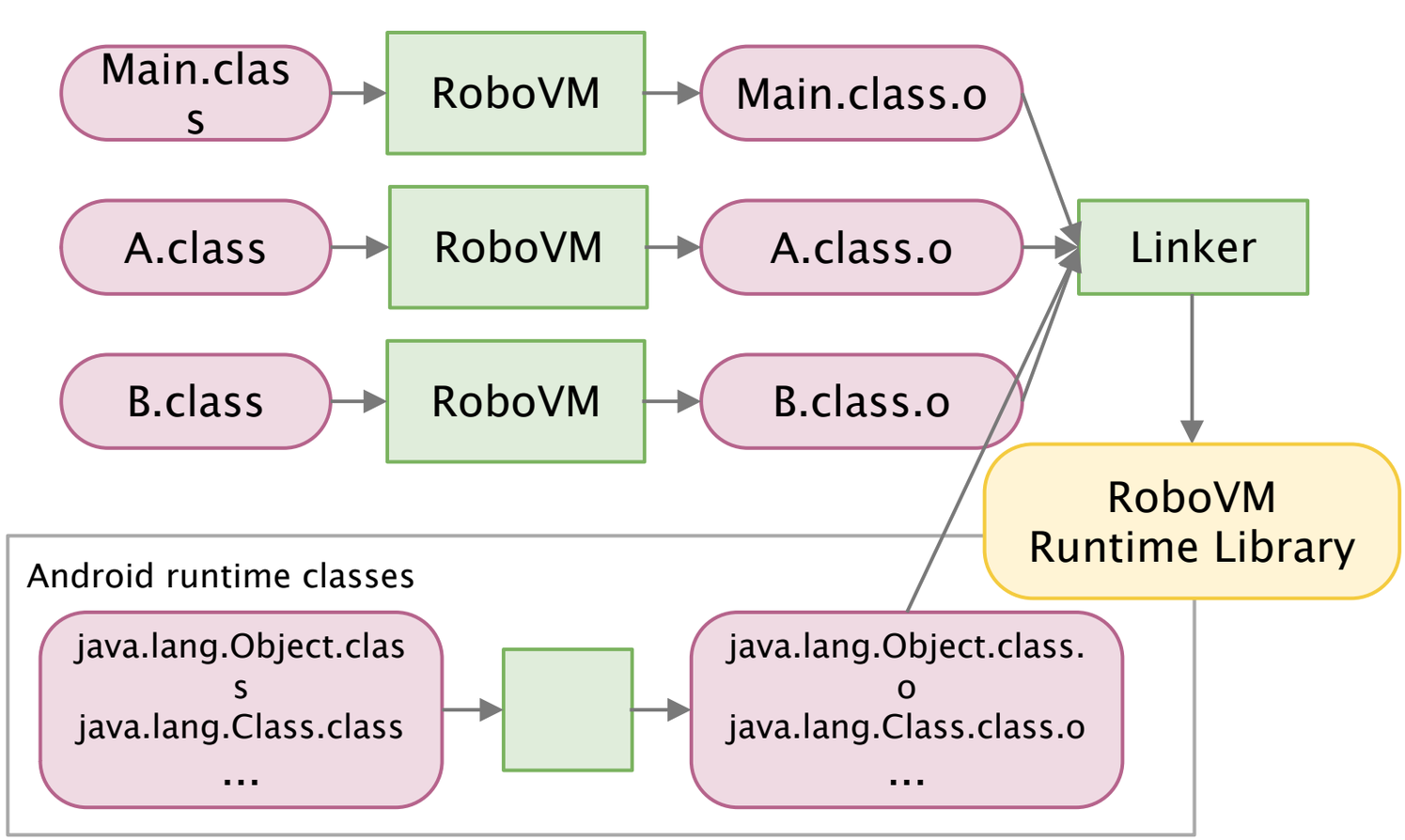
How to get on Mobile

- Introducting RoboVM for iOS
  - Makes use of Android-Classlibrary
  - Provides access to Native iOS-Frameworks including graphics
  - Is AOTed



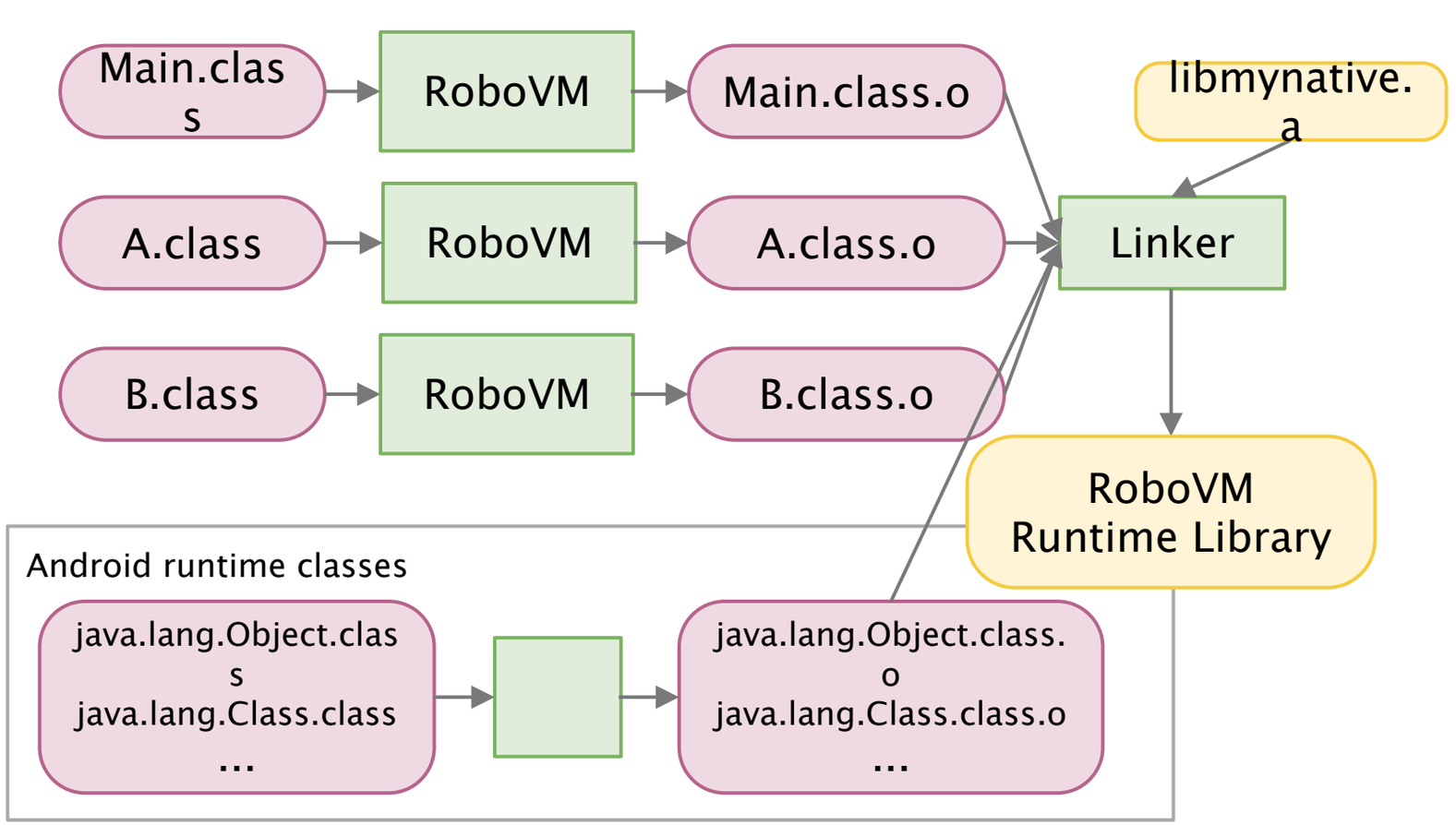
How to get on Mobile

- Introducting RoboVM for iOS
  - Makes use of Android-Classlibrary
  - Provides access to Native iOS-Frameworks including graphics
  - Is AOTed



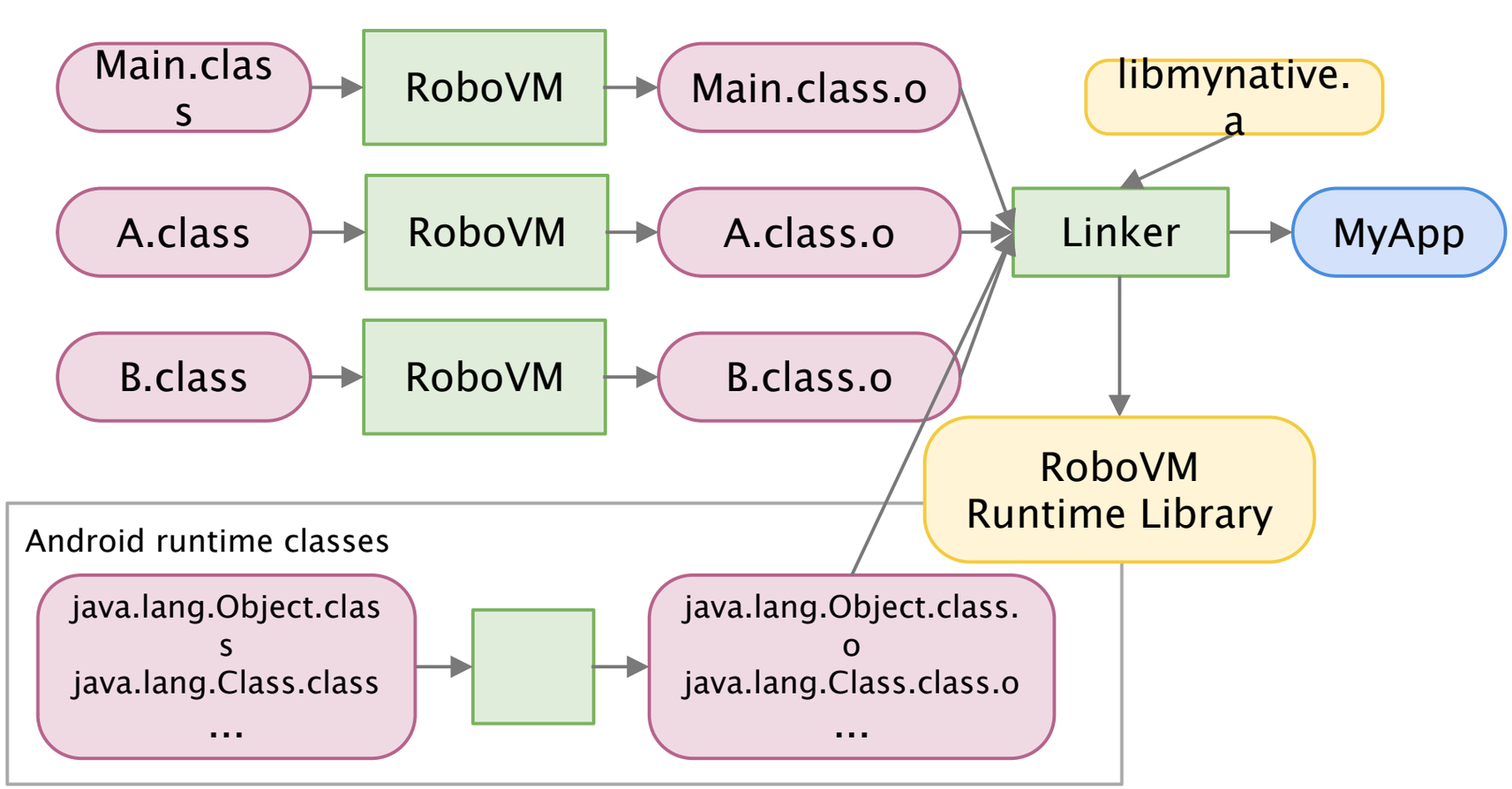
How to get on Mobile

- Introducting RoboVM for iOS
  - Makes use of Android-Classlibrary
  - Provides access to Native iOS-Frameworks including graphics
  - Is AOTed



How to get on Mobile

- Introducting RoboVM for iOS
  - Makes use of Android-Classlibrary
  - Provides access to Native iOS-Frameworks including graphics
  - Is AOTed



How to get on Mobile - UI Libs

- libgdx
  - Pros
    - gaming library but has a cross-platform version of the most important controls
    - cross-platform: Android, iOS (via RoboVM), Web (via GWT)
    - first app is in the iOS-App-Store!
  - Cons:
    - It‘s a gaming library

How to get on Mobile - UI Libs

- Android-UI
  - Pros
    - provides all UI-Elements
    - it‘s proven and performs
  - Cons:
    - It only works on Android
    - No LOAA

How to get on Mobile - UI Libs

- RoboVM-Native-UI
  - Pros
    - provides all UI-Elements
    - it‘s native so it performs well
  - Cons:
    - It only works on iOS
    - No LOAA

How to get on Mobile - UI Libs

- JavaFX
  - Pros
    - provides all UI-Elements
    - it is cross platform: Desktop, Embedded, Mobile
    - uses OpenGL which is available on iOS and Android
  - Cons:
    - It needs a VM or AOT compiler
    - Fonts:
      - ok on iOS == same lib as on desktop
      - nok on Android == does not use pango like desktop

JavaFX + RoboVM

Combine RoboVM AOT + JavaFX

- The idea
  - Take the JavaFX iOS port
  - Combine it with RoboVM
  - Result is a Runnable iOS application

Demo Time