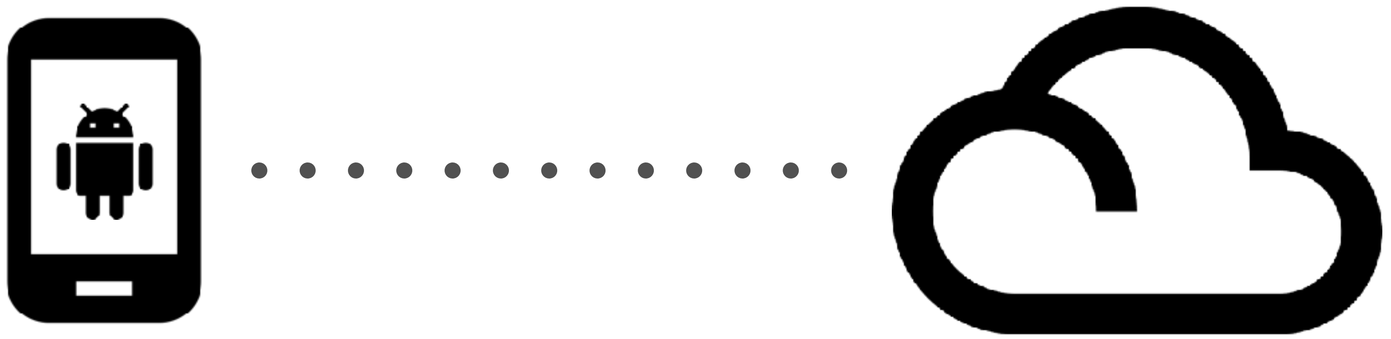
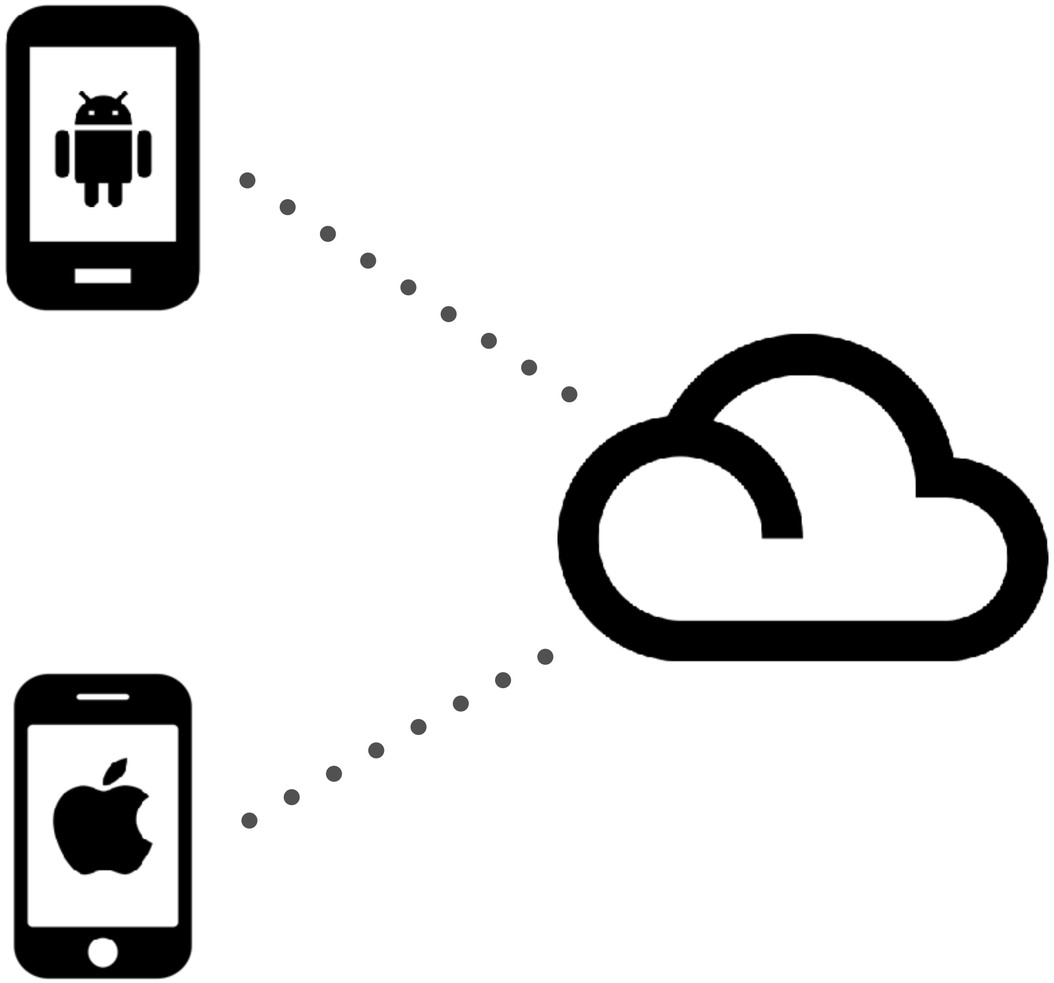
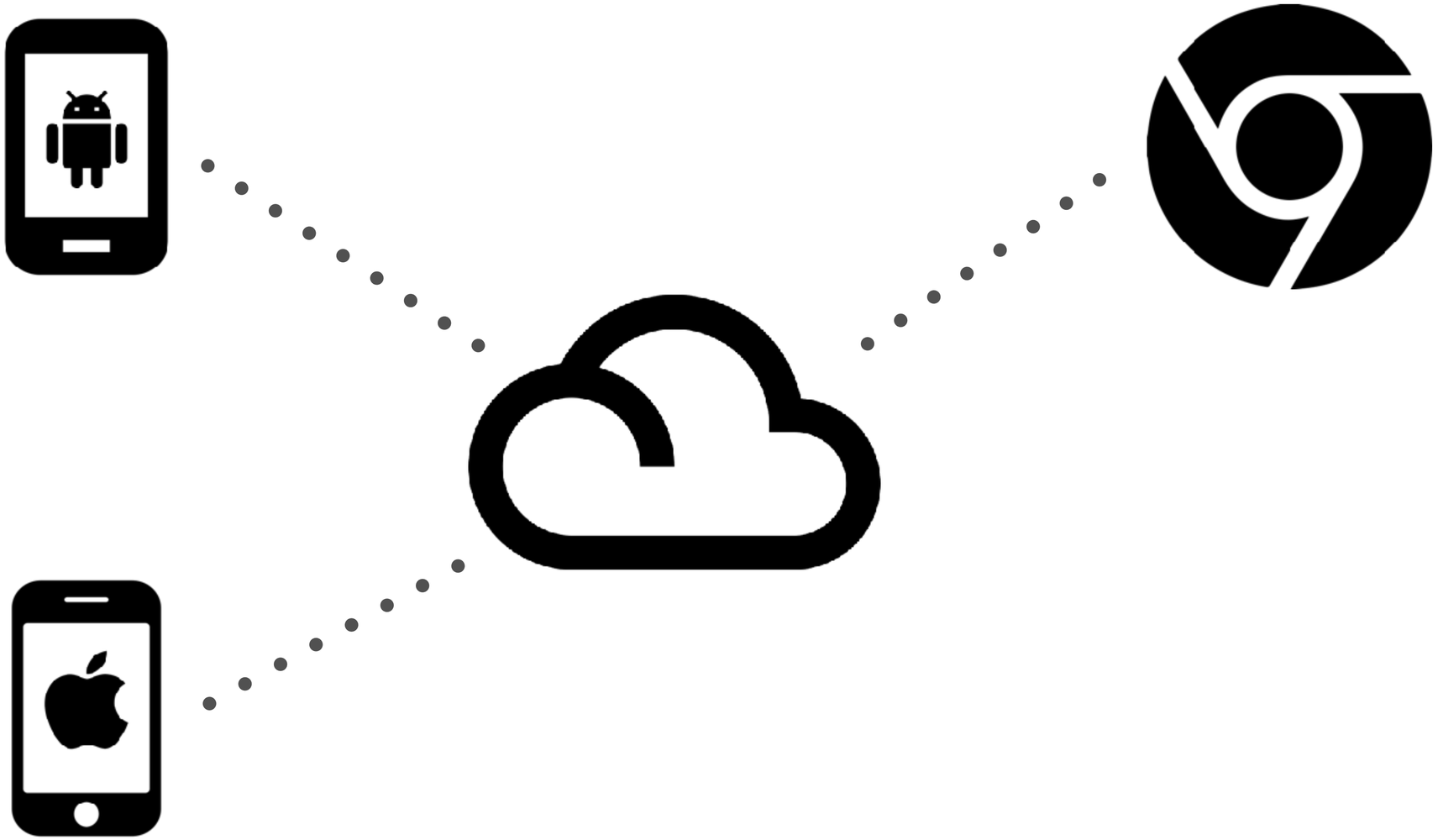
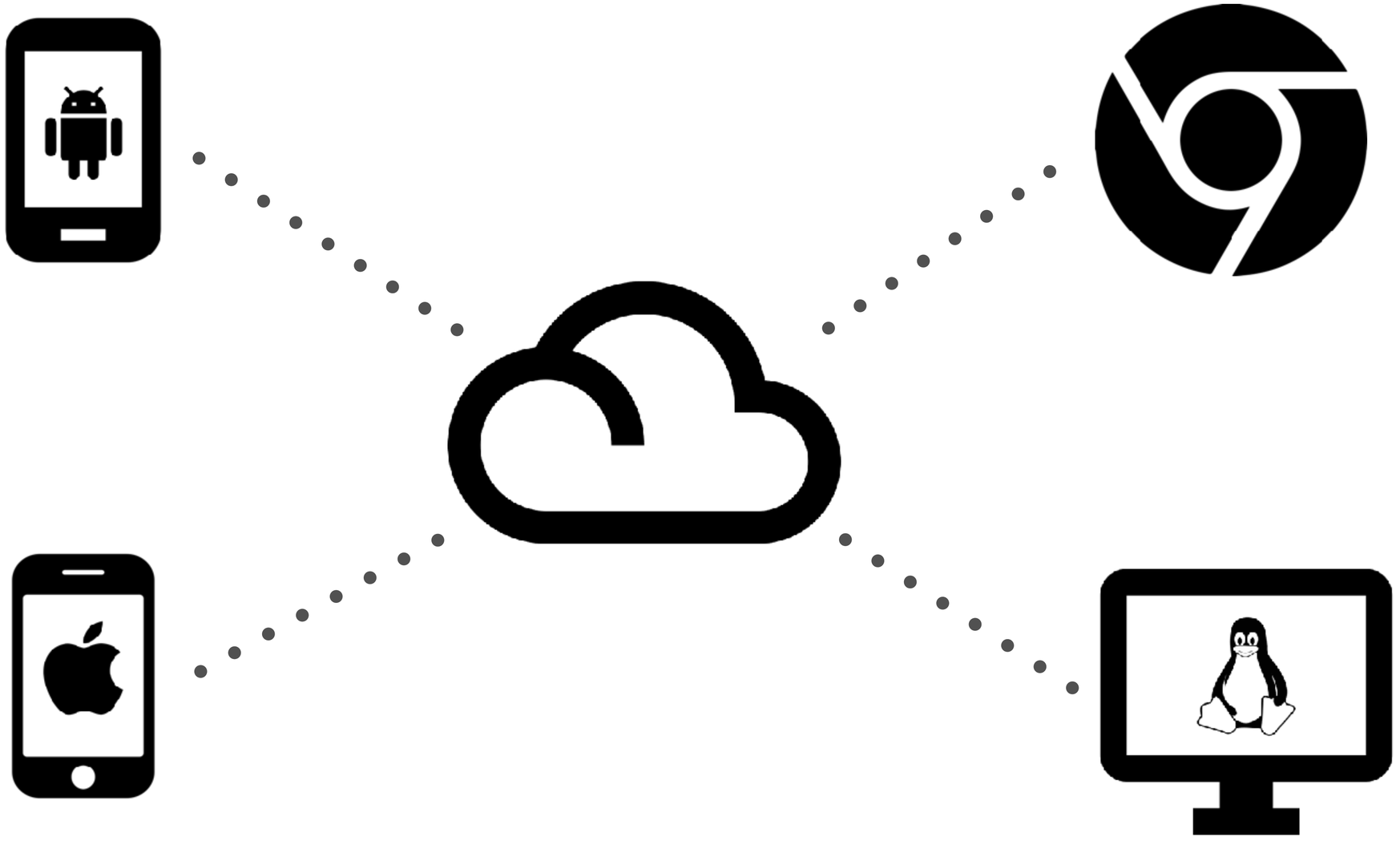
Building Multiplatform Projects with Kotlin

Sergey Ryabov



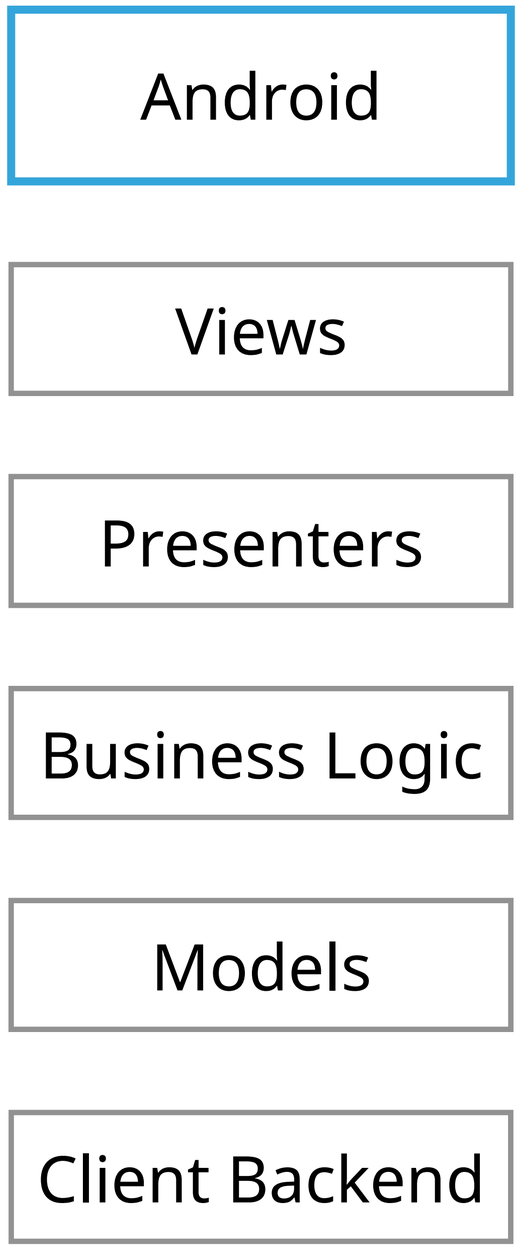


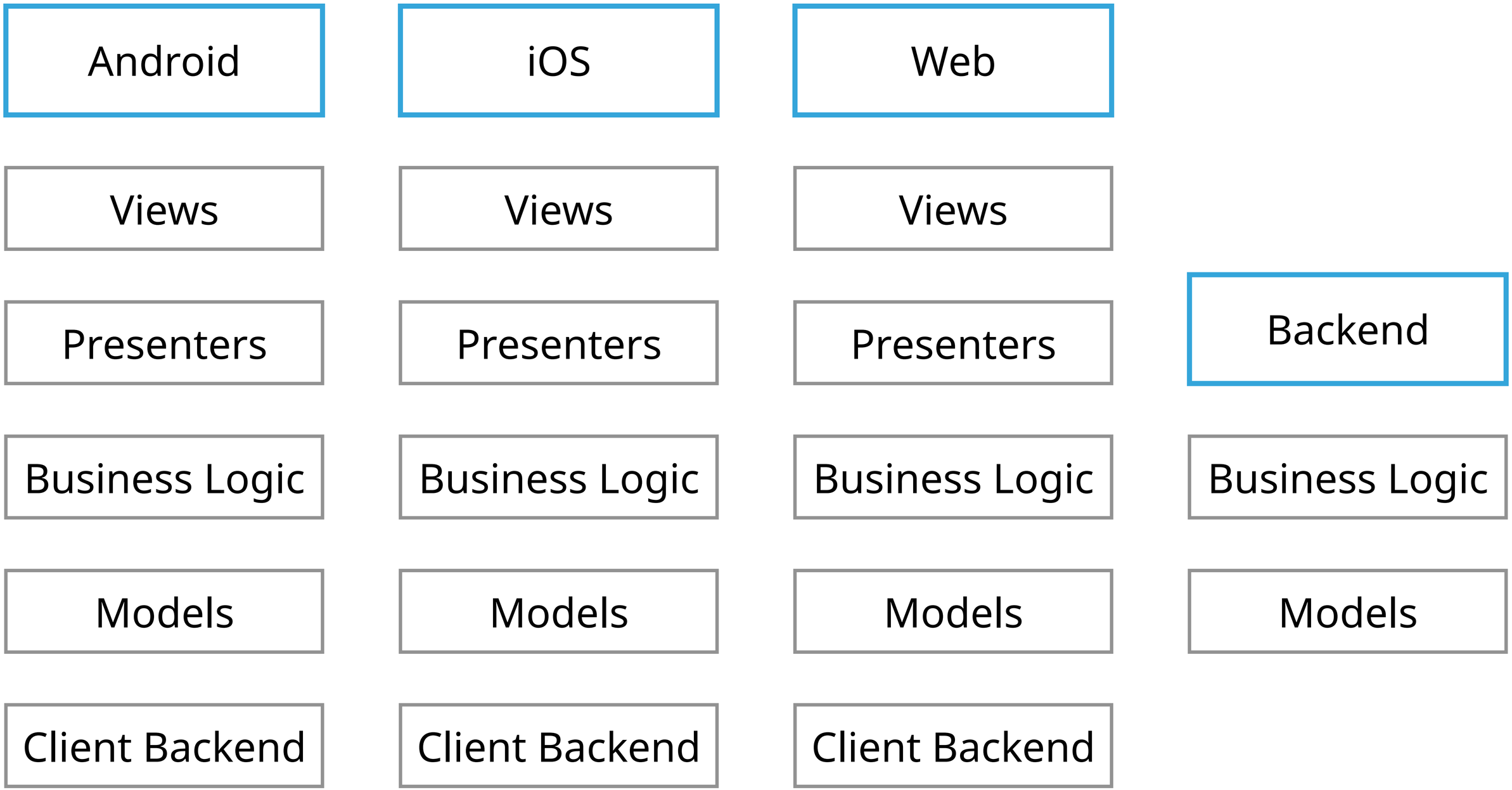


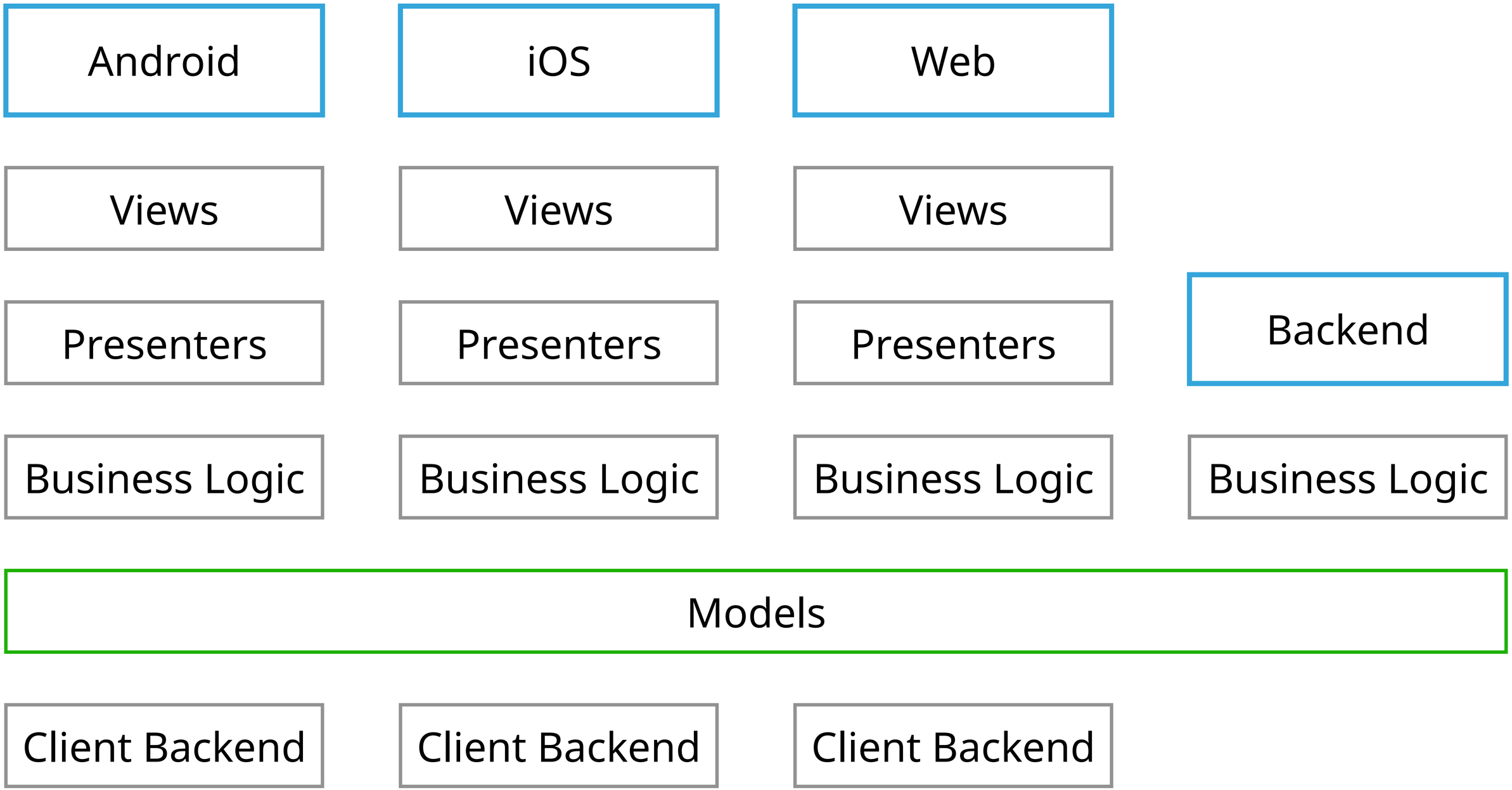


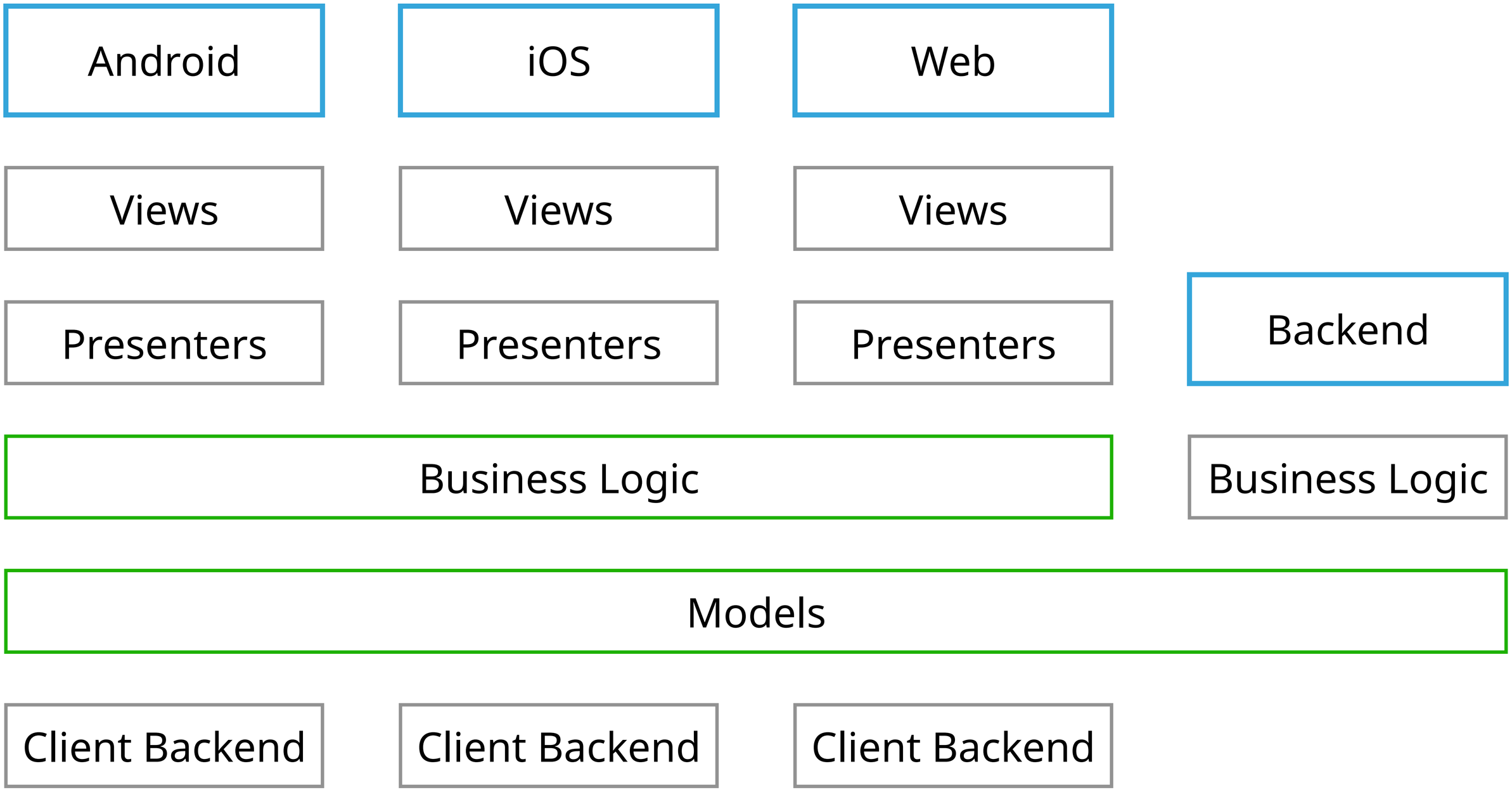
WRITE ONCE, RUN EVERYWHERE!

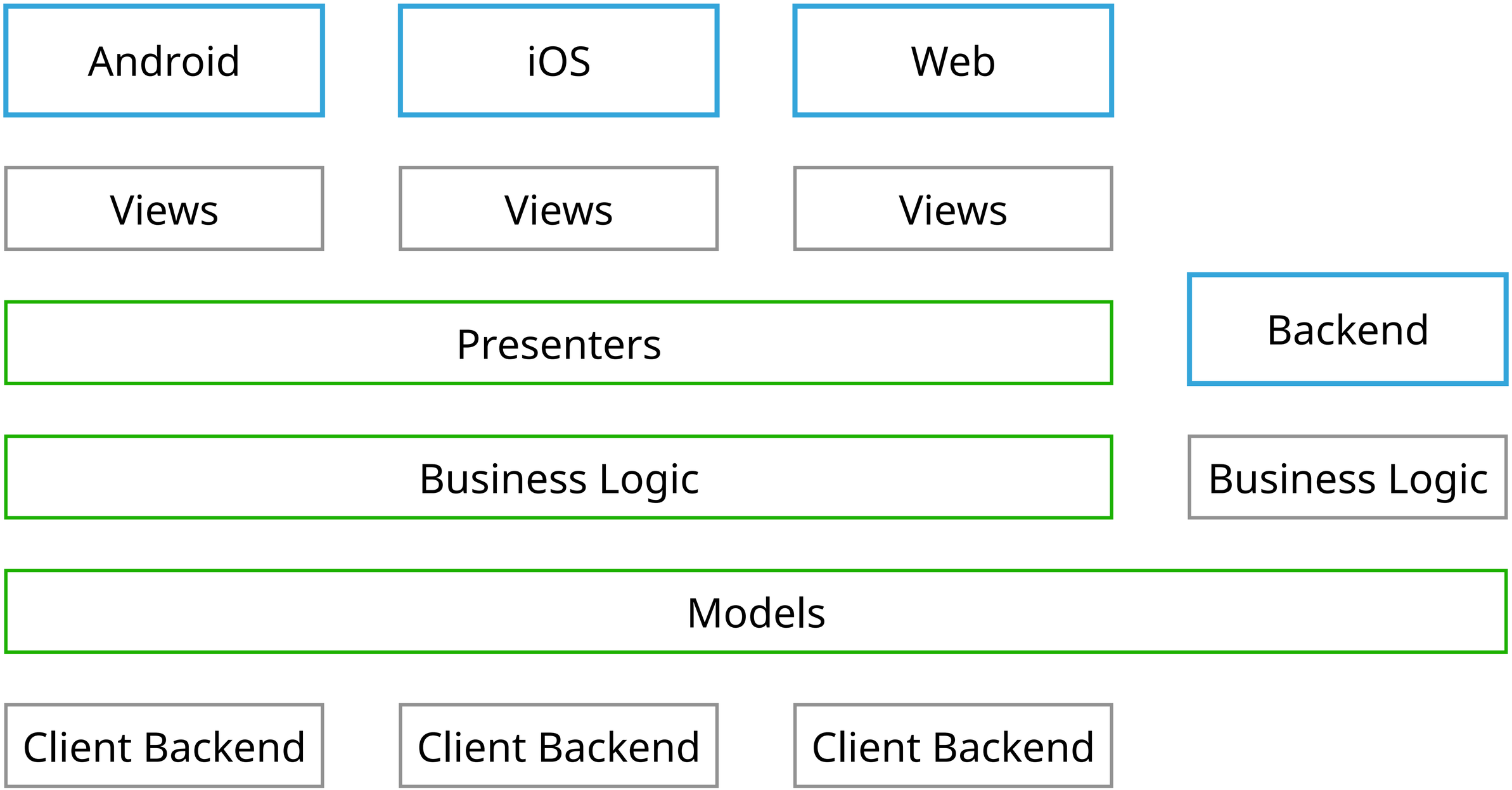
© JAVA


Android
Views
Presenters
Business Logic
Models
Client Backend


Android
iOS
Web
Views
Views
Views
Presenters
Presenters
Presenters
Backend
Business Logic
Business Logic
Business Logic
Business Logic
Models
Models
Models
Models
Client Backend
Client Backend
Client Backend


Android
iOS
Web
Views
Views
Views
Presenters
Presenters
Presenters
Backend
Business Logic
Business Logic
Business Logic
Business Logic
Models
Client Backend
Client Backend
Client Backend


Android
iOS
Web
Views
Views
Views
Presenters
Presenters
Presenters
Backend
Business Logic
Business Logic
Models
Client Backend
Client Backend
Client Backend


Android
iOS
Web
Views
Views
Views
Presenters
Backend
Business Logic
Business Logic
Models
Client Backend
Client Backend
Client Backend

WHAT DO WE ALREADY HAVE FOR THAT?

WHAT DO WE ALREADY HAVE FOR THAT?

- C++

WHAT DO WE ALREADY HAVE FOR THAT?

- C++
- J2ObjC + GWT

WHAT DO WE ALREADY HAVE FOR THAT?

- C++
- J2ObjC + GWT
- Xamarin

WHAT DO WE ALREADY HAVE FOR THAT?

- C++
- J2ObjC + GWT
- Xamarin
- ReactNative

WHAT DO WE ALREADY HAVE FOR THAT?

- C++
- J2ObjC + GWT
- Xamarin
- ReactNative
- Flutter

MAJOR PROBLEMS

MAJOR PROBLEMS

- Access to native APIs

MAJOR PROBLEMS

- Access to native APIs
- Latest SDK version support

MAJOR PROBLEMS

- Access to native APIs
- Latest SDK version support
- Modern dev stack

MAJOR PROBLEMS

- Access to native APIs
- Latest SDK version support
- Modern dev stack
- Executable size

WHY KOTLIN?

WHY KOTLIN?

Feb 2016 - Kotlin 1.0: production JVM, experimental JS

WHY KOTLIN?

Feb 2016 - **Kotlin 1.0:** production JVM, experimental JS

Mar 2017 - **Kotlin 1.1:** production JS

WHY KOTLIN?

Feb 2016 - **Kotlin 1.0:** production JVM, experimental JS

Mar 2017 - **Kotlin 1.1:** production JS

Apr 2017 - **Kotlin Native Tech Preview**

WHY KOTLIN?

Feb 2016 - **Kotlin 1.0:** production JVM, experimental JS

Mar 2017 - **Kotlin 1.1:** production JS

Apr 2017 - **Kotlin Native Tech Preview**

Nov 2017 - **Kotlin Native 0.4:** iOS support

WHY KOTLIN?

Feb 2016 - **Kotlin 1.0:** production JVM, experimental JS

Mar 2017 - **Kotlin 1.1:** production JS

Apr 2017 - **Kotlin Native Tech Preview**

Nov 2017 - **Kotlin Native 0.4:** iOS support

Nov 2017 - **Kotlin 1.2:** Multiplatform projects, JVM/JS support

WHY KOTLIN?

Feb 2016 - **Kotlin 1.0:** production JVM, experimental JS

Mar 2017 - **Kotlin 1.1:** production JS

Apr 2017 - **Kotlin Native Tech Preview**

Nov 2017 - **Kotlin Native 0.4:** iOS support

Nov 2017 - **Kotlin 1.2:** Multiplatform projects, JVM/JS support

Feb 2018 - **Kotlin Native 0.6:** Multiplatform support +

WHY KOTLIN?

Feb 2016 - **Kotlin 1.0:** production JVM, experimental JS

Mar 2017 - **Kotlin 1.1:** production JS

Apr 2017 - **Kotlin Native Tech Preview**

Nov 2017 - **Kotlin Native 0.4:** iOS support

Nov 2017 - **Kotlin 1.2:** Multiplatform projects, JVM/JS support

Feb 2018 - **Kotlin Native 0.6:** Multiplatform support + transparent ObjC/Kotlin collection types interop

WHY KOTLIN?

Feb 2016 - **Kotlin 1.0**: production JVM, experimental JS

Mar 2017 - **Kotlin 1.1**: production JS

Apr 2017 - **Kotlin Native Tech Preview**

Nov 2017 - **Kotlin Native 0.4**: iOS support

Nov 2017 - **Kotlin 1.2**: Multiplatform projects, JVM/JS support

Feb 2018 - **Kotlin Native 0.6**: Multiplatform support + transparent ObjC/Kotlin collection types interop

Jul 2018 - **Kotlin Native 0.8**: kotlin-stdlib sync with JVM/JS

WHY KOTLIN IS BETTER?

WHY KOTLIN IS BETTER?

- Access to native APIs

WHY KOTLIN IS BETTER?

- Access to native APIs
- Latest SDK version support

WHY KOTLIN IS BETTER?

- Access to native APIs
- Latest SDK version support
- Modern dev stack

WHY KOTLIN IS BETTER?

- Access to native APIs
- Latest SDK version support
- Modern dev stack
- Executable size

HOW DOES IT WORK IN KOTLIN?

HOW DOES IT WORK IN KOTLIN?

- Gradle multimodule projects

HOW DOES IT WORK IN KOTLIN?

- Gradle multimodule projects
  - Common

HOW DOES IT WORK IN KOTLIN?

- Gradle multimodule projects
  - Common
  - Platform

HOW DOES IT WORK IN KOTLIN?

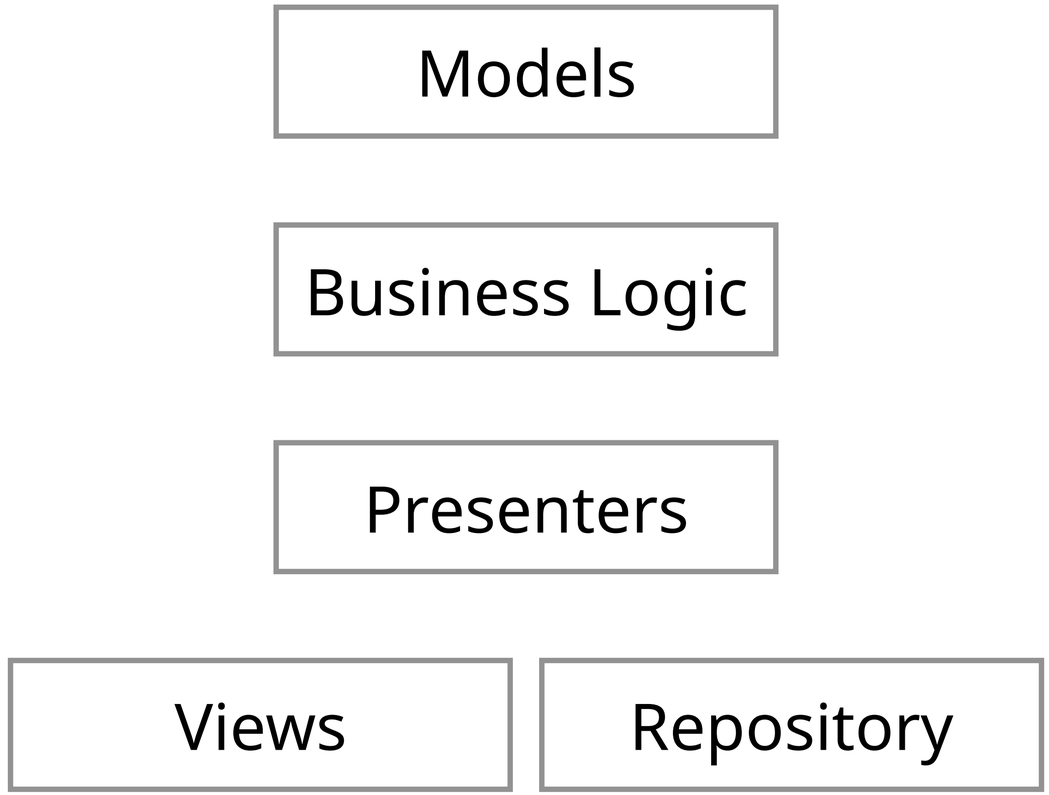
- Gradle multimodule projects
  - Common
  - Platform
  - Regular

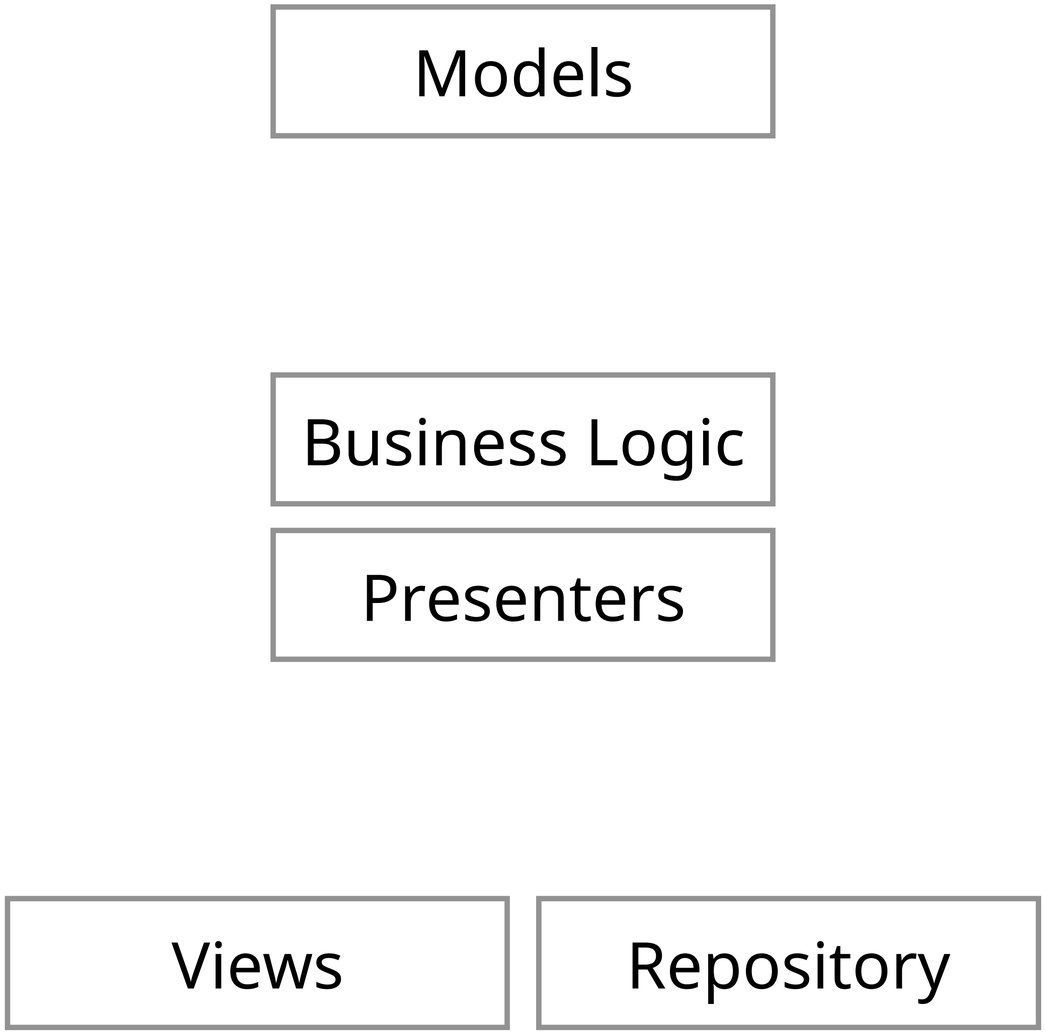
HOW DOES IT WORK IN KOTLIN?

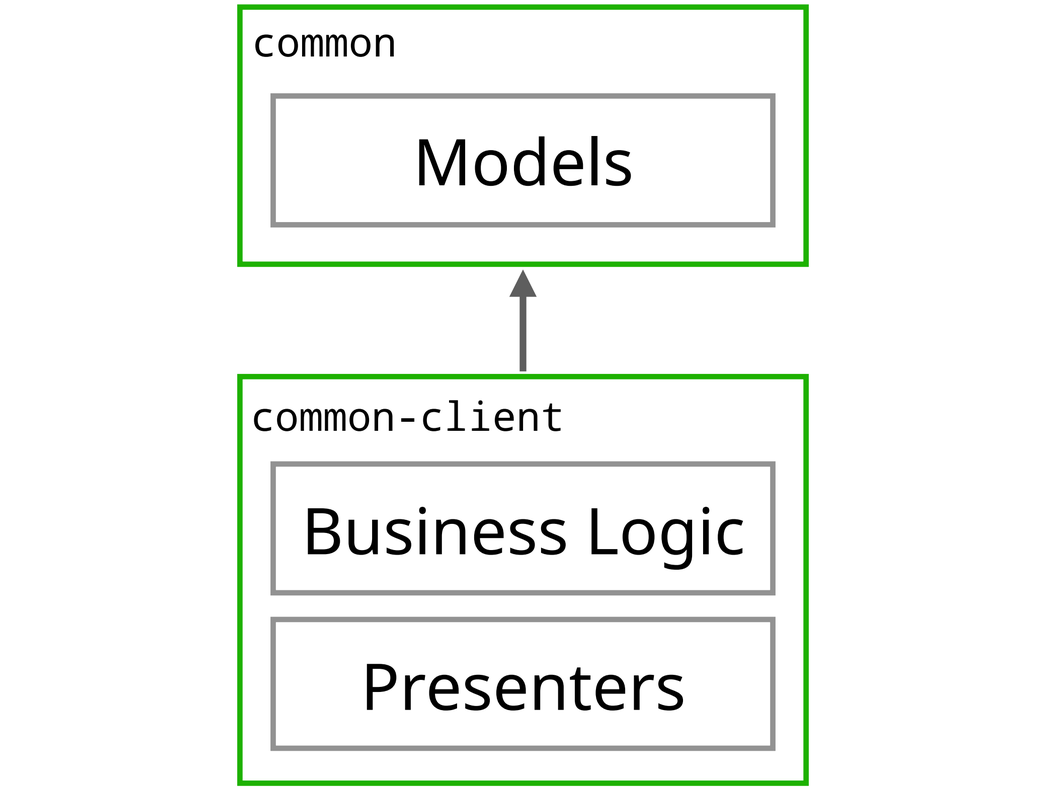
- Gradle multimodule projects
  - Common
  - Platform
  - Regular
- Multiplatform headers & implementations

HOW DOES IT WORK IN KOTLIN?

- Gradle multimodule projects
  - Common
  - Platform
  - Regular
- Multiplatform headers & implementations
  - `expect` & `actual`

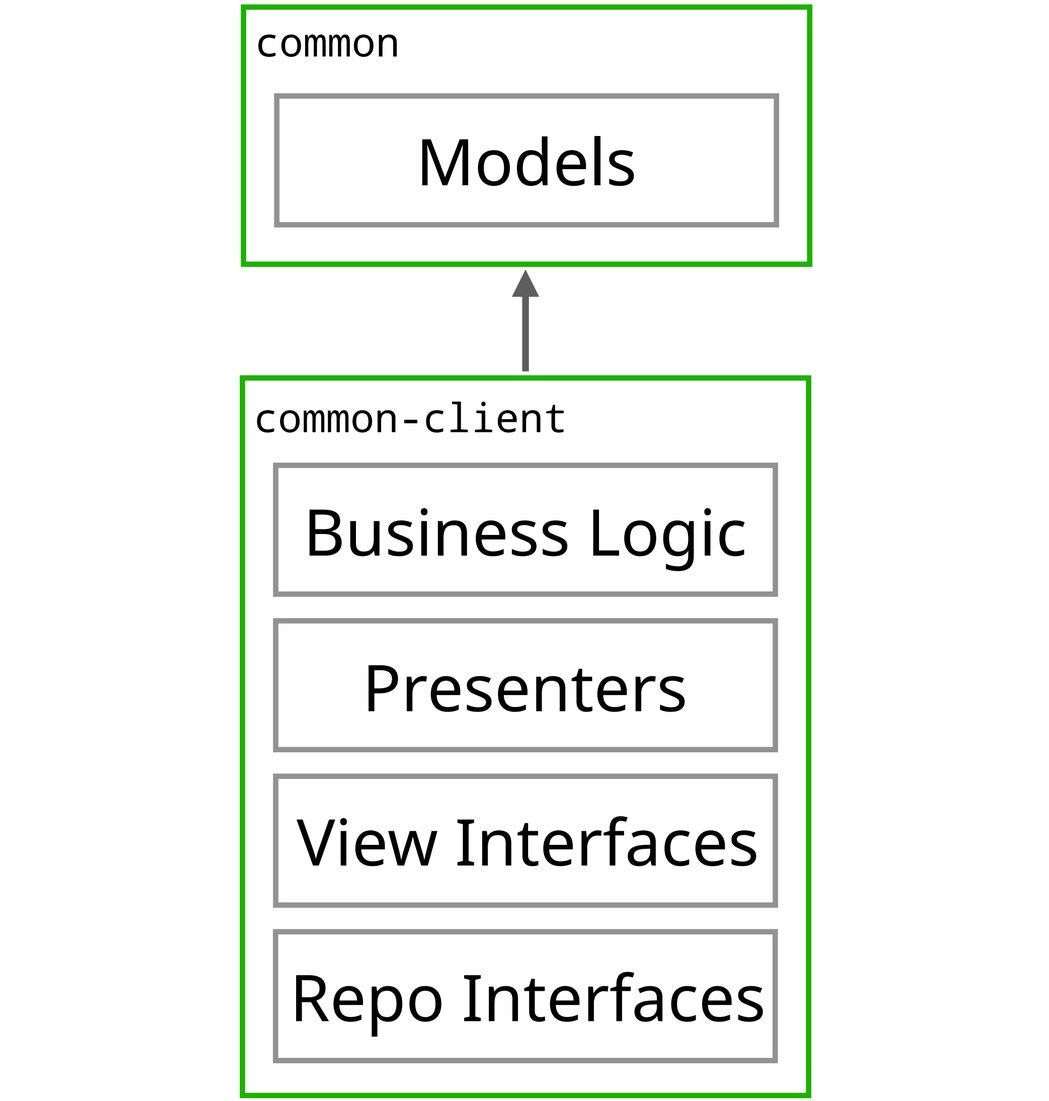

Models
Business Logic
Presenters
Views
Repository


Models
Business Logic
Presenters
Views
Repository



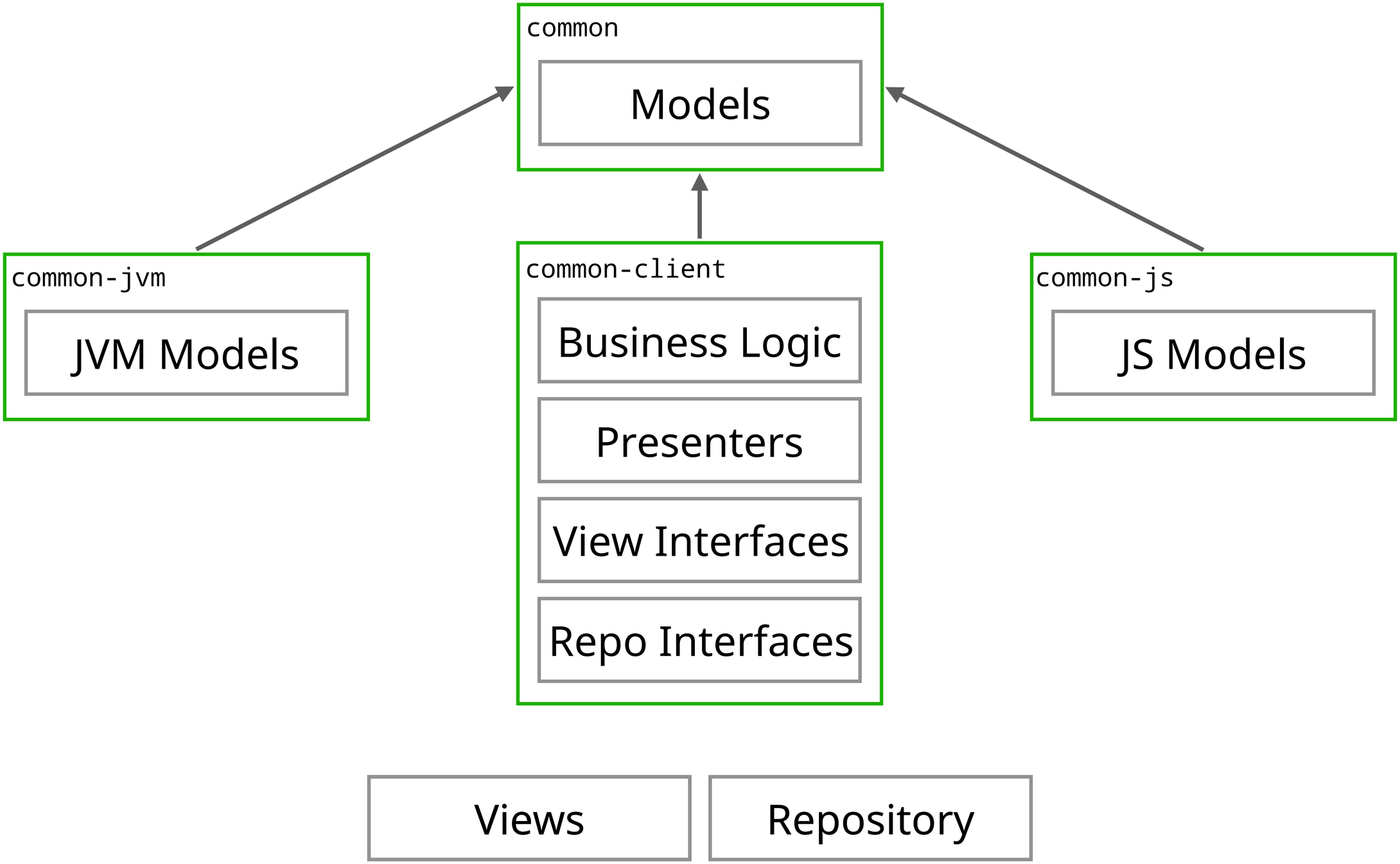
Views

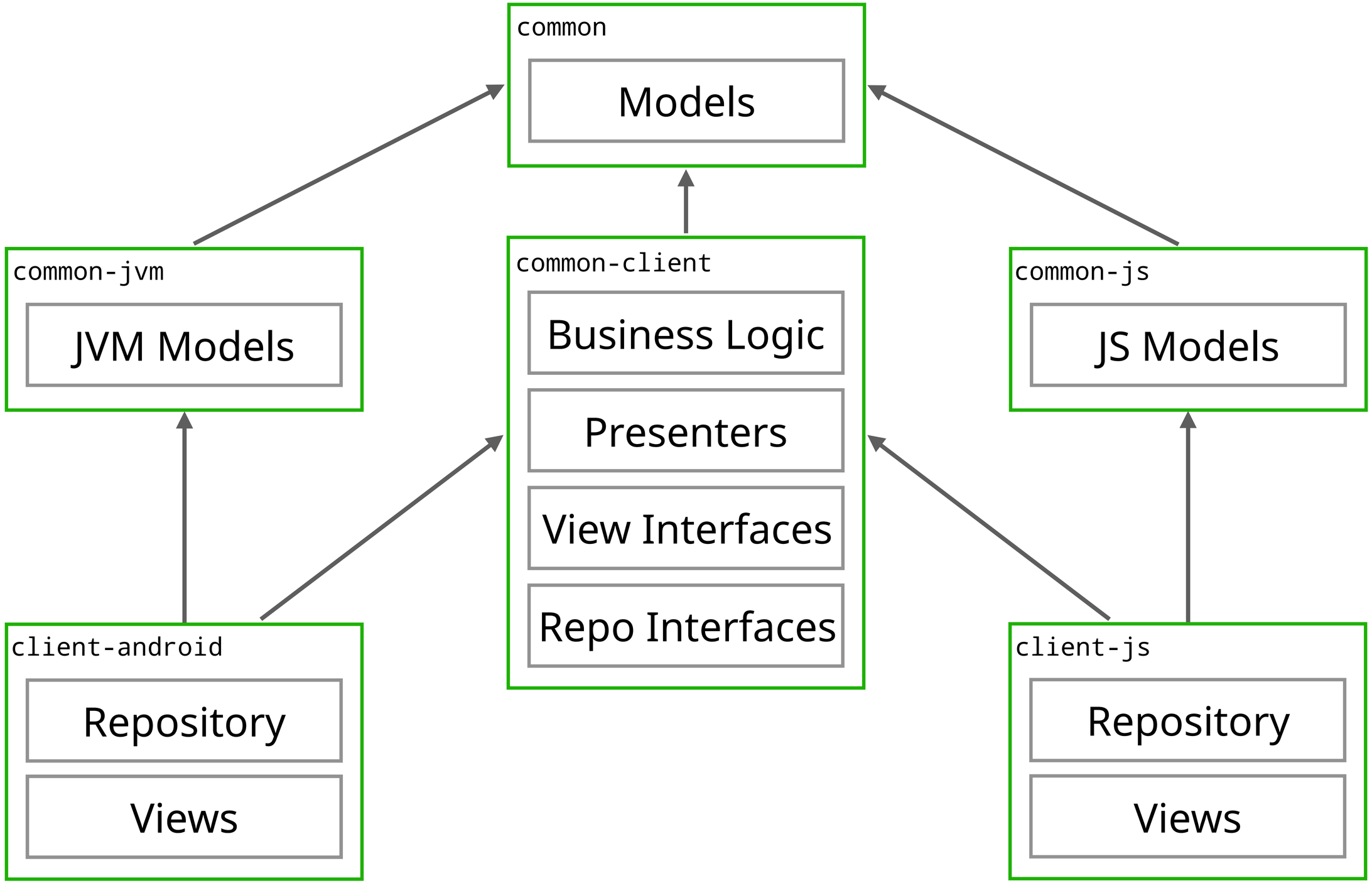
Repository

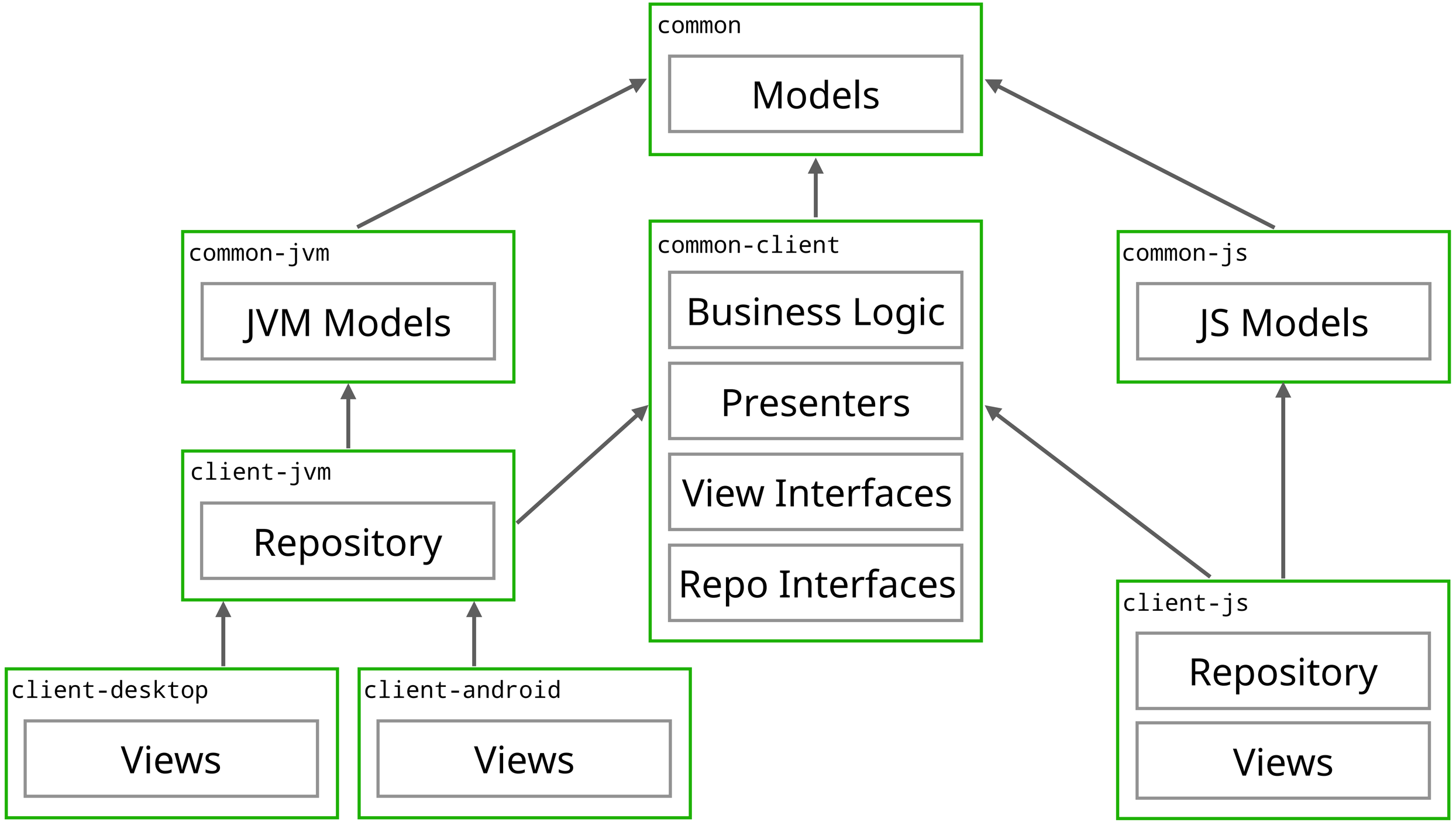


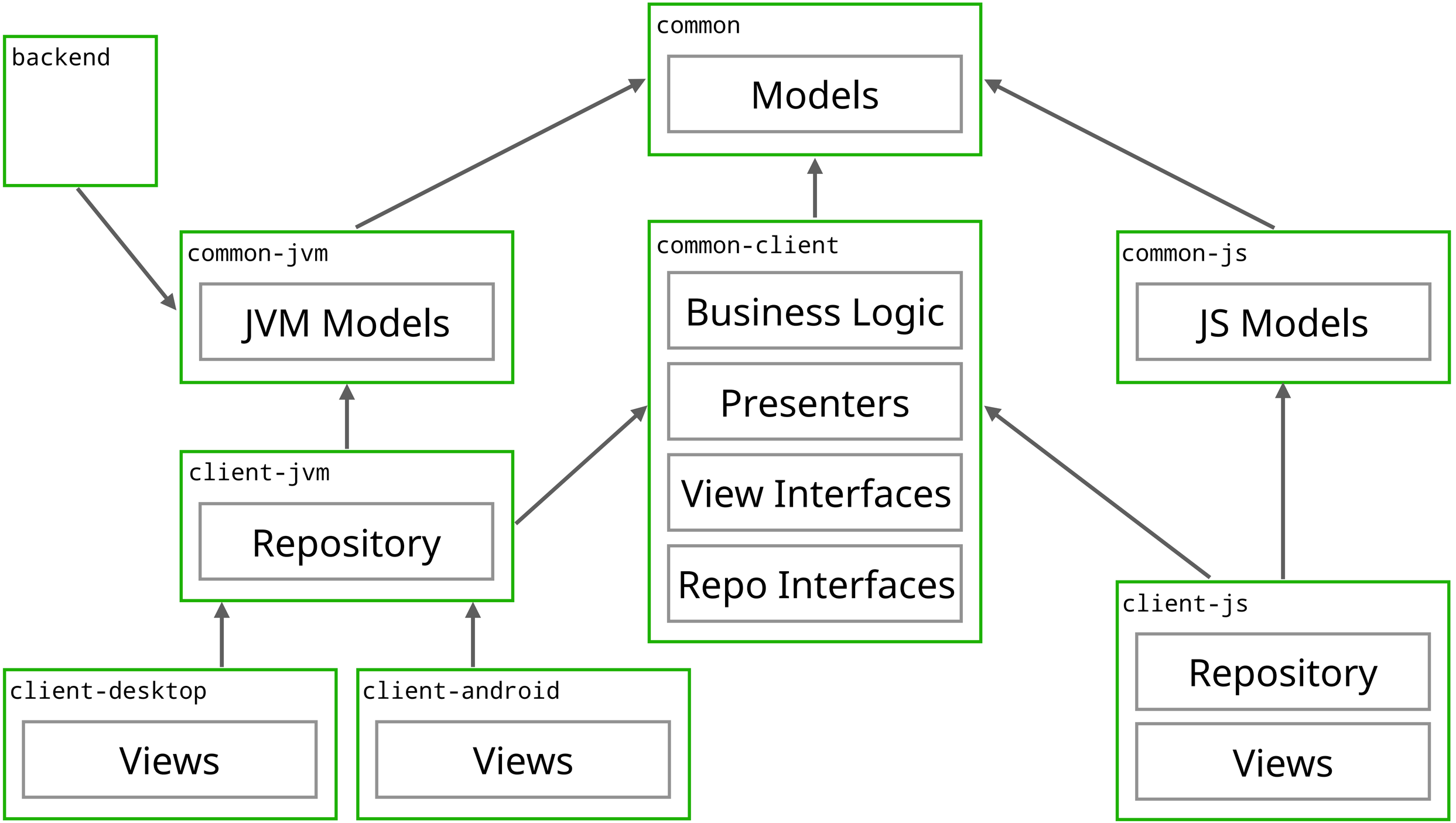
Views

Repository


common
Models
common-jvm
JVM Models
common-client
Business Logic
Presenters
View Interfaces
Repo Interfaces
common-js
JS Models
Views
Repository


common
Models
common-jvm
JVM Models
common-client
Business Logic
Presenters
View Interfaces
Repo Interfaces
common-js
JS Models
client-android
Repository
Views
client-js
Repository
Views

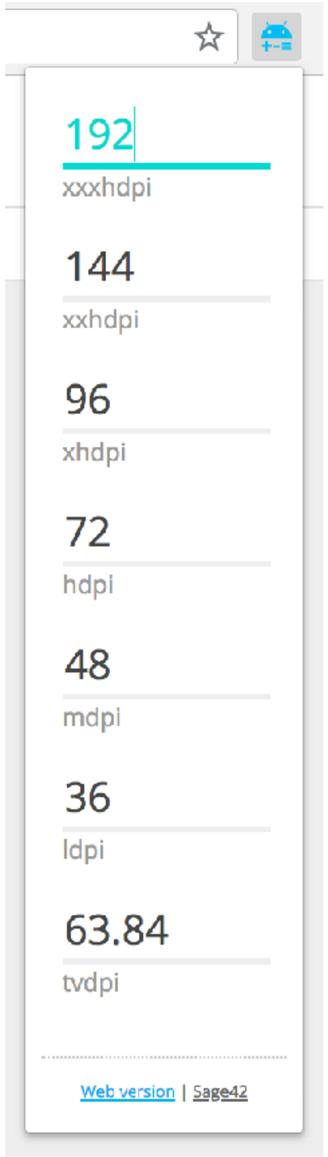

common
Models
common-jvm
JVM Models
common-client
Business Logic
Presenters
View Interfaces
Repo Interfaces
common-js
JS Models
client-jvm
Repository
client-desktop
Views
client-android
Views
client-js
Repository
Views


backend
common
Models
common-jvm
JVM Models
common-client
Business Logic
Presenters
View Interfaces
Repo Interfaces
common-js
JS Models
client-jvm
Repository
client-desktop
Views
client-android
Views
client-js
Repository
Views

EXAMPLE: ANDROID APP + FIREFOX EXTENSION

EXAMPLE: ANDROID APP + FIREFOX EXTENSION

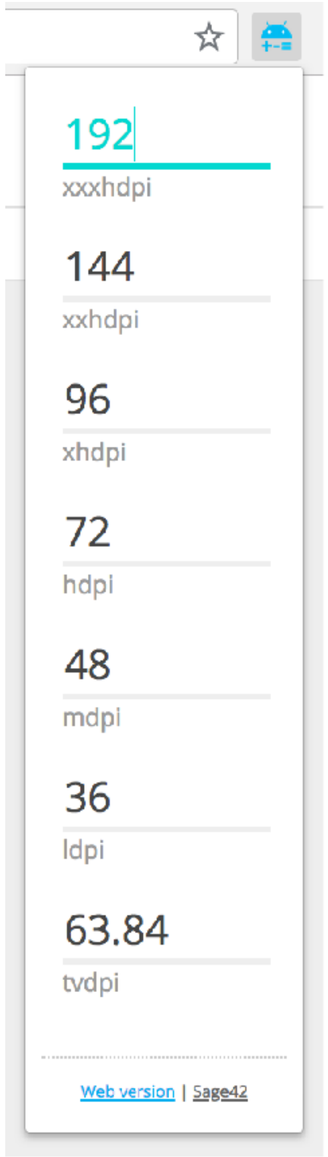
Original

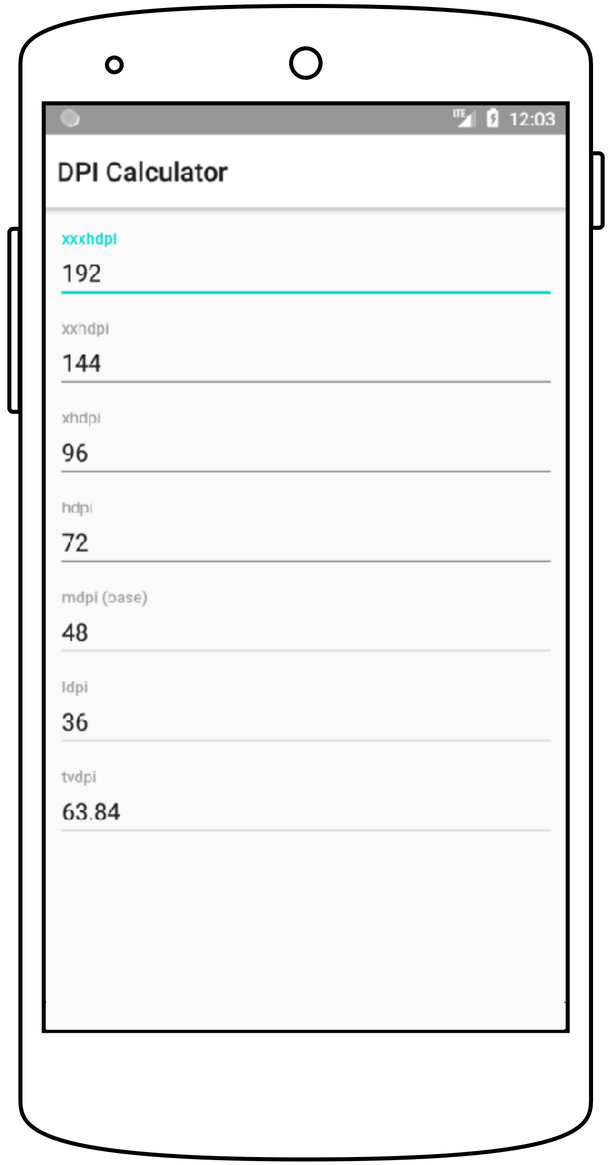


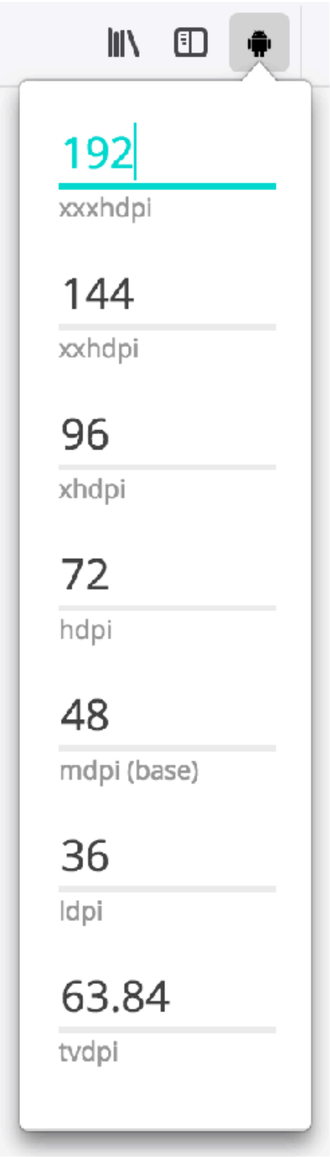
http://jennift.com/dpical.html

EXAMPLE: ANDROID APP + FIREFOX EXTENSION

Original

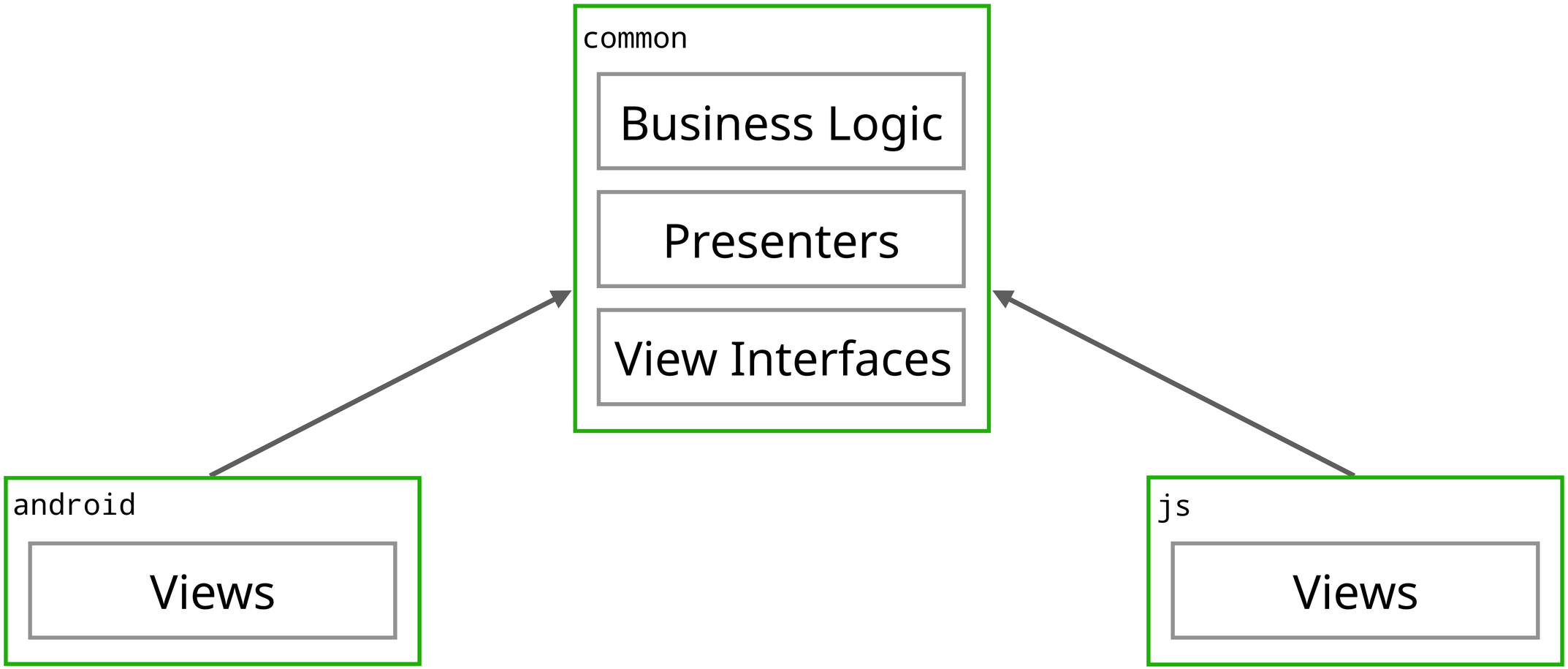






http://jennift.com/dpical.html

EXAMPLE: ANDROID APP + FIREFOX EXTENSION



CAVEATS

CAVEATS

- Drawbacks of the concept

CAVEATS

- Drawbacks of the concept
  - Lots of manual work for module configuration (KT-23930)

CAVEATS

- Drawbacks of the concept
  - Lots of manual work for module configuration (KT-23930)
  - Several platform codebases can’t sit in one module (different SourceSets)

CAVEATS

- Drawbacks of the concept
  - Lots of manual work for module configuration (KT-23930)
  - Several platform codebases can’t sit in one module (different SourceSets)
- Every platform has its own problems

CAVEATS

- Drawbacks of the concept
  - Lots of manual work for module configuration (KT-23930)
  - Several platform codebases can’t sit in one module (different SourceSets)
- Every platform has its own problems
  - Kotlin/JS

CAVEATS

- Drawbacks of the concept
  - Lots of manual work for module configuration (KT-23930)
  - Several platform codebases can’t sit in one module (different SourceSets)
- Every platform has its own problems
  - Kotlin/JS
    - One output JS-file per module

CAVEATS

- Drawbacks of the concept
  - Lots of manual work for module configuration (KT-23930)
  - Several platform codebases can’t sit in one module (different SourceSets)
- Every platform has its own problems
  - Kotlin/JS
    - One output JS-file per module
    - Manual setup for including libs in the output

CAVEATS

- Drawbacks of the concept
  - Lots of manual work for module configuration (KT-23930)
  - Several platform codebases can’t sit in one module (different SourceSets)
- Every platform has its own problems
  - Kotlin/JS
    - One output JS-file per module
    - Manual setup for including libs in the output
  - Kotlin/Native

CAVEATS

- Drawbacks of the concept
  - Lots of manual work for module configuration (KT-23930)
  - Several platform codebases can’t sit in one module (different SourceSets)
- Every platform has its own problems
  - Kotlin/JS
    - One output JS-file per module
    - Manual setup for including libs in the output
  - Kotlin/Native
    - Still experimental - some features are not supported

CAVEATS

- Drawbacks of the concept
  - Lots of manual work for module configuration (KT-23930)
  - Several platform codebases can’t sit in one module (different SourceSets)
- Every platform has its own problems
  - Kotlin/JS
    - One output JS-file per module
    - Manual setup for including libs in the output
  - Kotlin/Native
    - Still experimental - some features are not supported
    - No reuse for multiplatform (yet)

FUTURE PLANS

FUTURE PLANS

- Re-thinking Multimodule projects configuration

FUTURE PLANS

- Re-thinking Multimodule projects configuration
- Catching up Kotlin/Native with other platforms

FUTURE PLANS

- Re-thinking Multimodule projects configuration
- Catching up Kotlin/Native with other platforms
- Increasing common libs count
  - kotlin.test
  - kotlinx.html
  - kotlinx.serialization
  - kotlinx.coroutines
  - kotlinx.io
  - SQLDelight
  - ...

PRODUCTION USAGE

PRODUCTION USAGE

- KotlinConf app (https://github.com/jetbrains/kotlinconf-app)

PRODUCTION USAGE

- KotlinConf app (https://github.com/jetbrains/kotlinconf-app)
- Droidcon NYC (Sessionize) app (https://github.com/touchlab/DroidconKotlin)

PRODUCTION USAGE

- KotlinConf app (https://github.com/jetbrains/kotlinconf-app)
- Droidcon NYC (Sessionize) app (https://github.com/touchlab/DroidconKotlin)
- Revolut (in-progress)

WRITE ONCE, RUN EVERYWHERE

WRITE ONCE, SHARE EVERYWHERE

WRITE ONCE, SHARE EVERYWHERE

- Create common business logic, not UI

WRITE ONCE, SHARE EVERYWHERE

- Create common business logic, not UI
- Write every part of the app in one language (optional)

LINKS

- Code from the talk
  https://goo.gl/VYgHmr

- Serialization support for Kotlin/Native
  https://github.com/Kotlin/kotlinx.serialization/issues/86

- Channels support for Kotlin/Native
  https://github.com/Kotlin/kotlinx.coroutines/issues/462

- New Multiplatform configuration approach
  https://youtrack.jetbrains.com/issue/KT-23930

Building Multiplatform Projects with Kotlin

Sergey Ryabov
@colriot