Shaped + KMM

Wrong way of using KMM

Nikolay Dmitriev
George Yemelyanov

APALON

contact@apalon.com

apalon.com

О чем будем говорить

- Спикеры

О чем будем говорить

- Спикеры
- Почему KMM

О чем будем говорить

- Спикеры
- Почему KMM
- Shaped

О чем будем говорить

- Спикеры
- Почему KMM
- Shaped
- Архитектура редактора

О чем будем говорить

- Спикеры
- Почему KMM
- Shaped
- Архитектура редактора
- Проблемы iOS
- Проблемы Android

О чем будем говорить

- Спикеры
- Почему KMM
- Shaped
- Архитектура редактора
- Проблемы iOS
- Проблемы Android
- Выводы

Дмитриев Николай

- Unity3D
- Android
- Flutter
- iOS

Георгий Емельянов

- Android
- KMM

Почему KMM

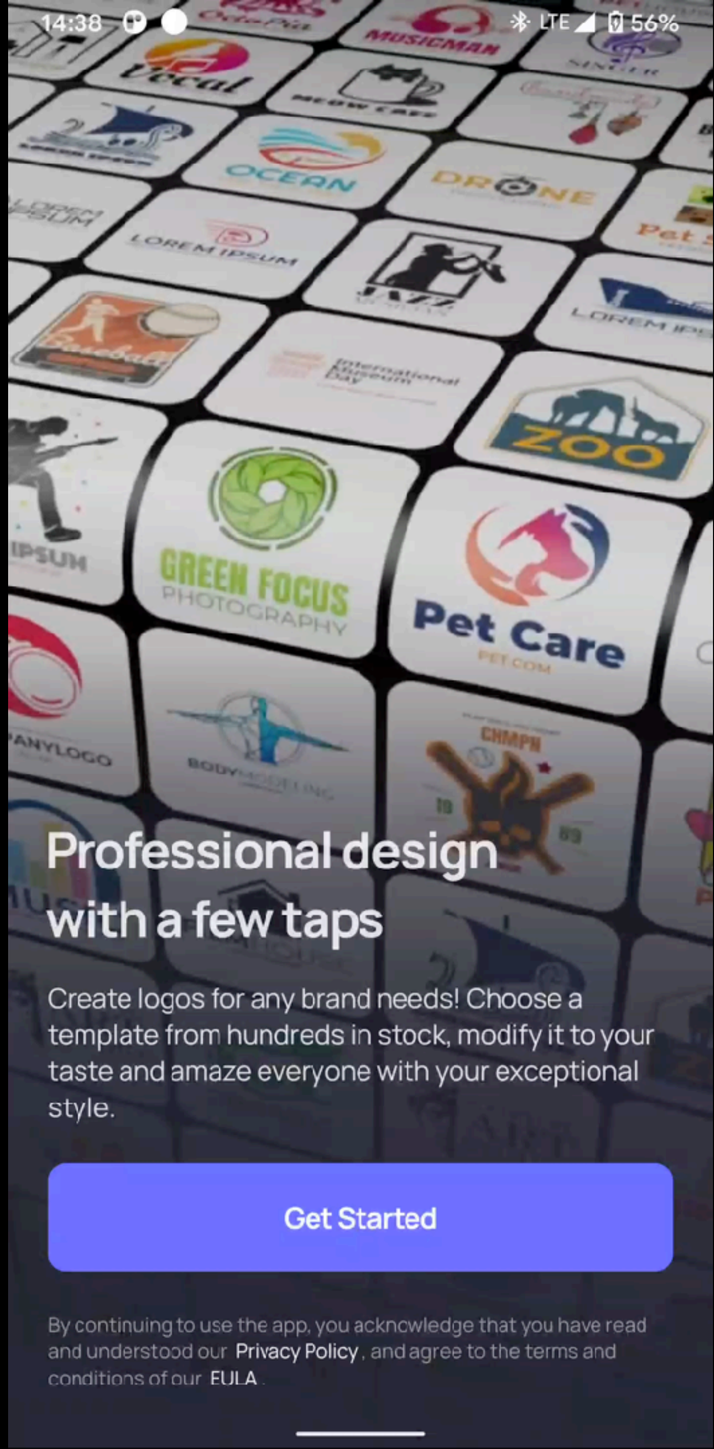
 - Apalon - продуктовая компания

 - KMM - многообещающая, но молодая технология

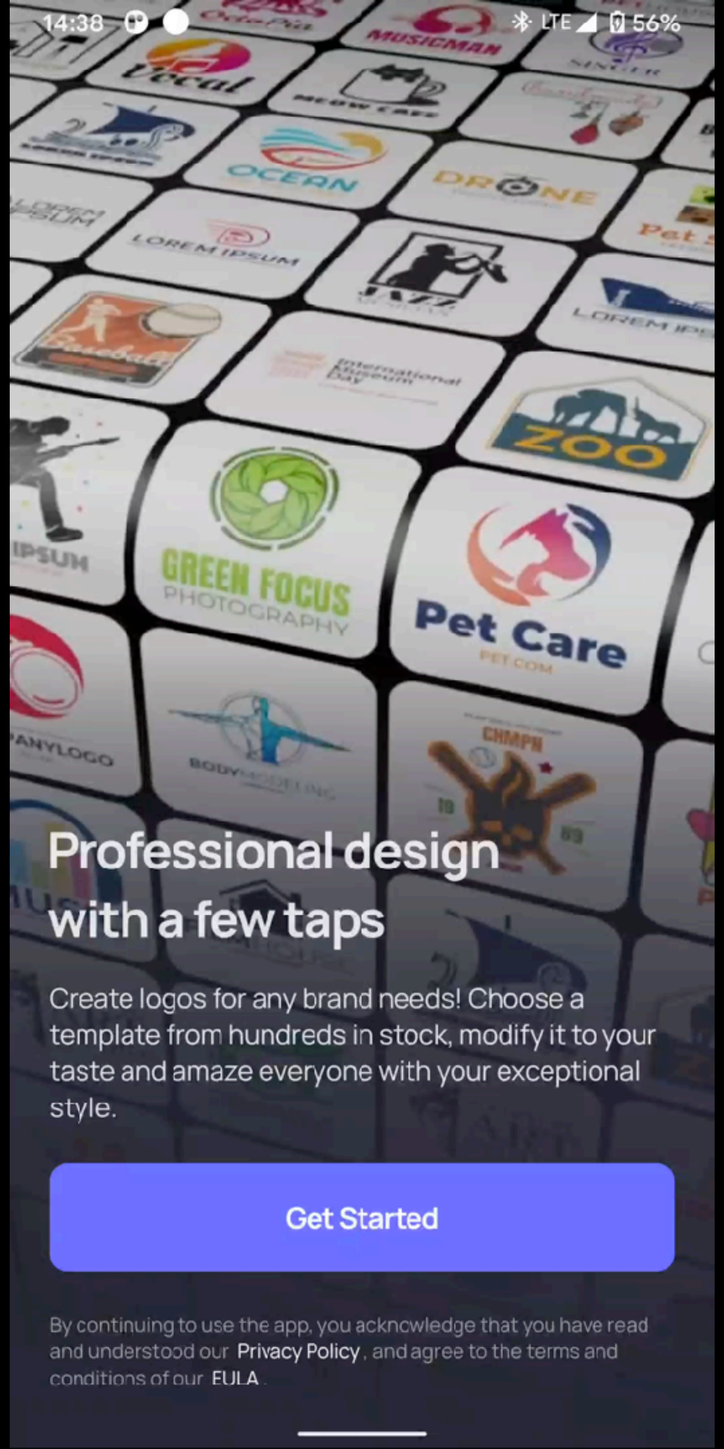
 - Shaped - подопытный кролик

О чем будем говорить

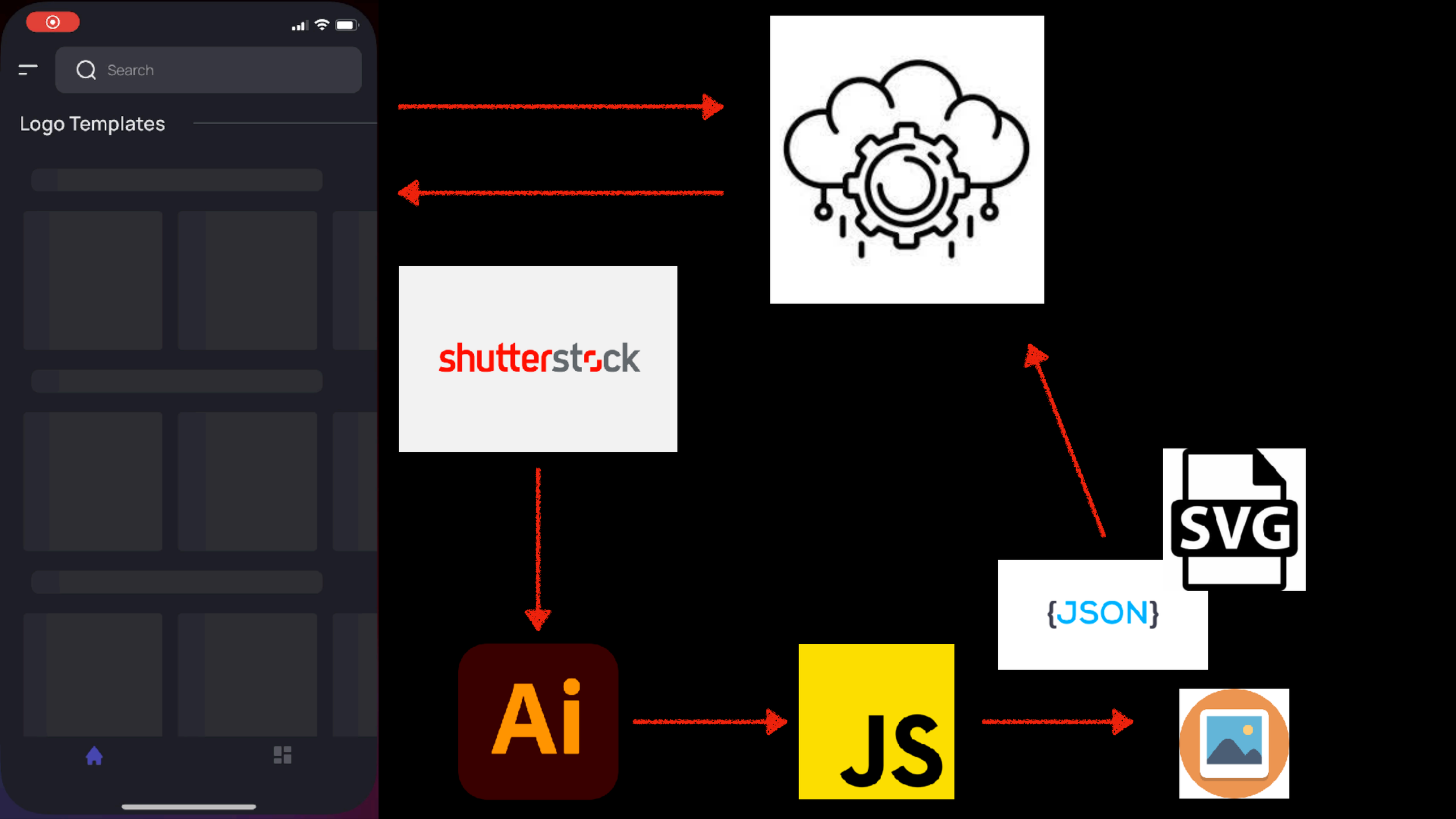
- ~~Спикеры~~
- ~~Почему KMM~~
- Shaped
- Архитектура редактора
- Проблемы iOS
- Проблемы Android
- Выводы

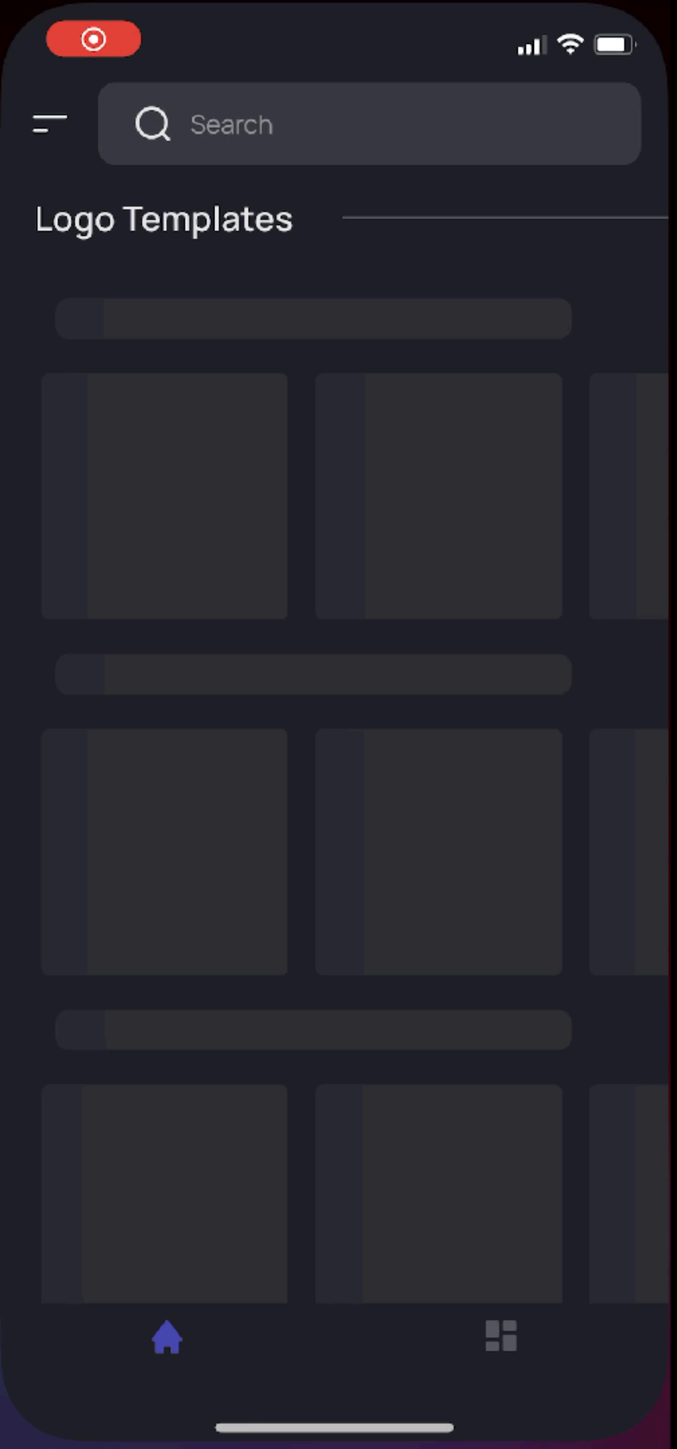


Что такое Shaped ?



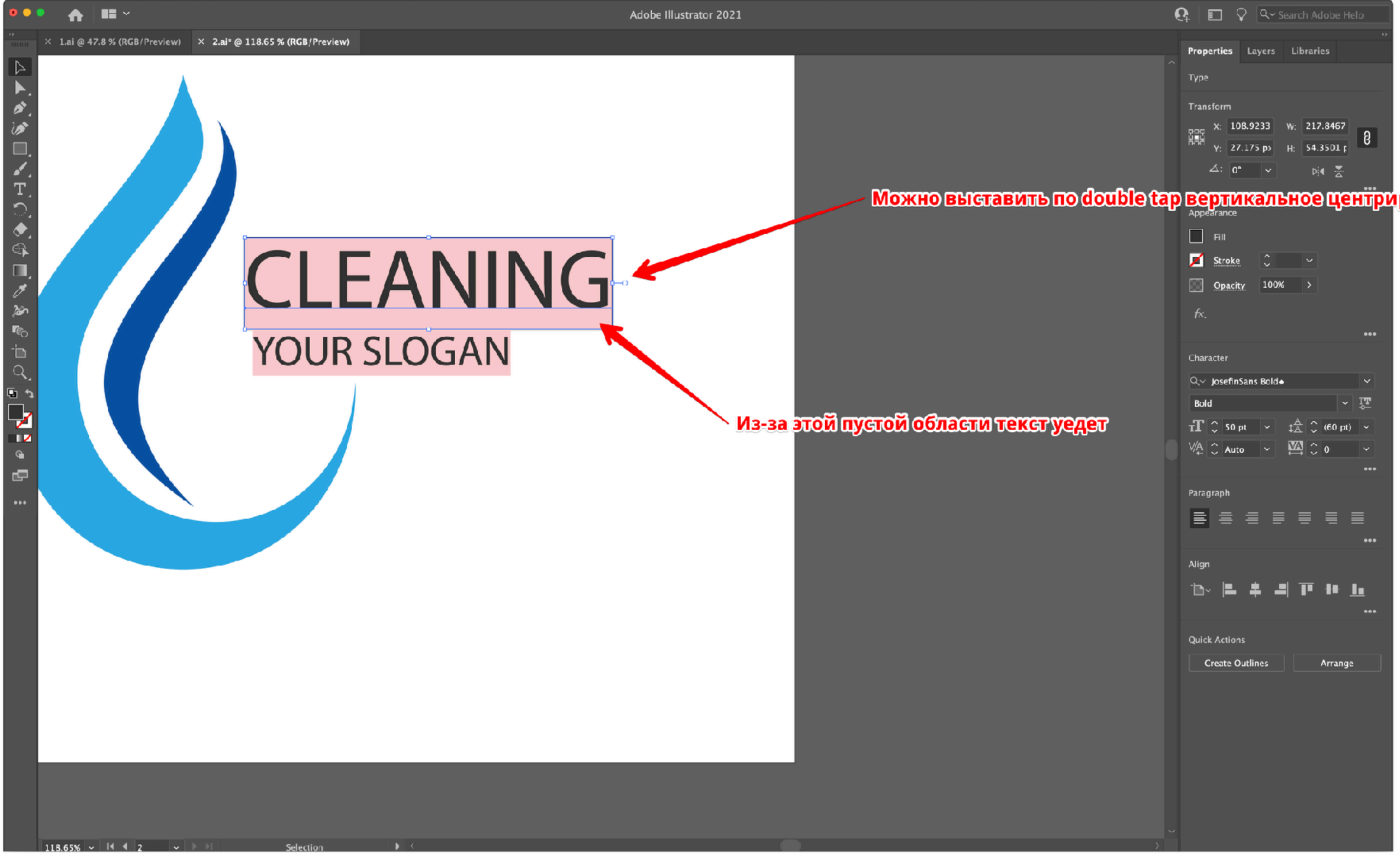
Что такое Shaped ?


Search
Logo Templates
shutterstock
Ai
JS
SVG
{JSON}



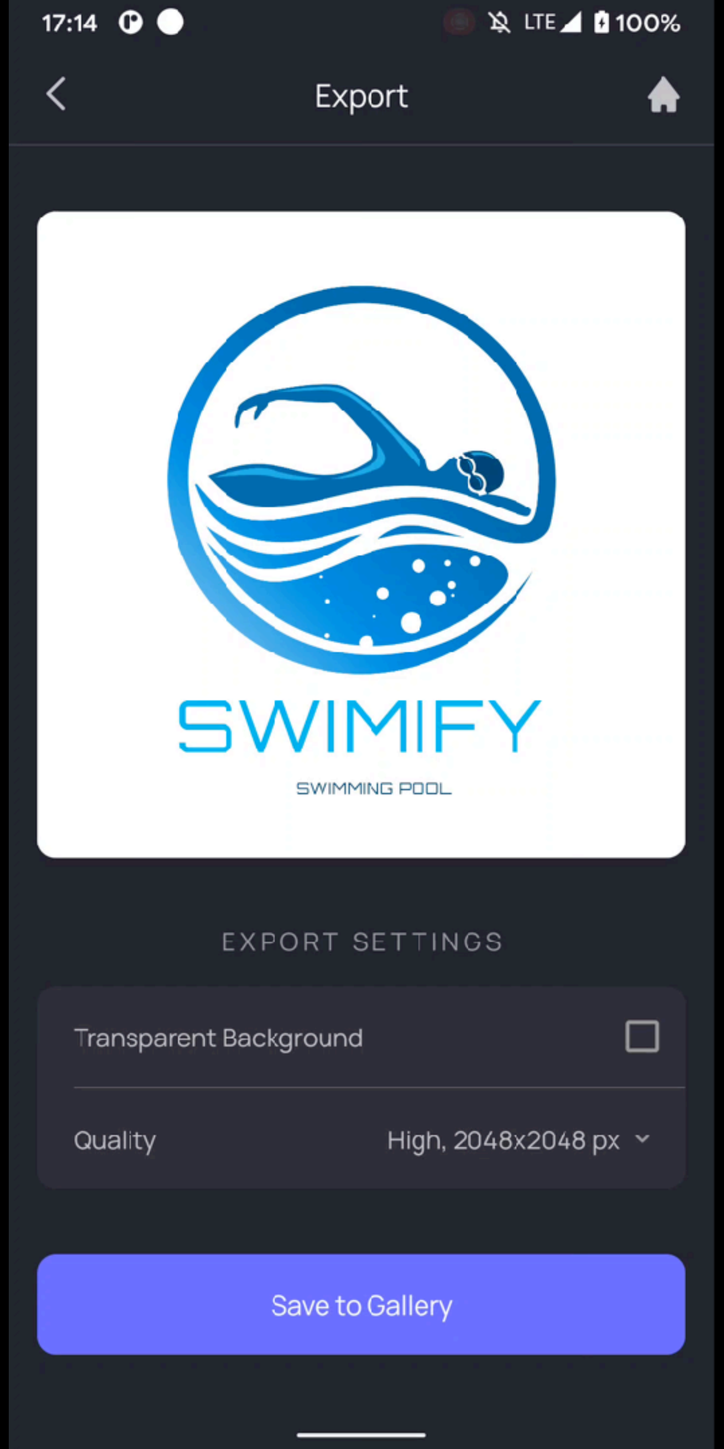
12_template.json


```
{
  "name": "12",
  "size": {
    "width": 500,
    "height": 500
  },
  "layers": [
    {
      "transform": {
        "rotation": 0,
        "rect": {
          "left": 108.369290111299,
          "top": -454.69321981391,
          "right": 393.063050506993,
          "bottom": -169.999999999998
        }
      },
      "content": {
        "type": "SVG",
        "svg": "<?xml version=\"1.0\" encoding=\"utf-8\"?>\n<!-- Generator: Adobe Illustrator 25.2.0, SVG Export Plug-In . S
      }
    },
    {
      "transform": {
        "rotation": 0,
        "rect": {
          "left": 53.69921875,
          "top": -163.71484375,
          "right": 469.1337890625,
          "bottom": -85.4501953125
        }
      },
      "content": {
        "type": "Text",
        "text": "Beautyful",
        "fontSize": 72,
        "fontName": "JosefinSans-Regular",
        "textStyles": [],
        "alignment": "CENTER",
        "fillColor": "#d87f27",
        "letterSpacing": 200,
        "lineSpacing": 0.20000002119276
      }
    },
    {
      "transform": {
        "rotation": 0,
        "rect": {
          "left": 158.4169921875,
          "top": -74.916015625,
```

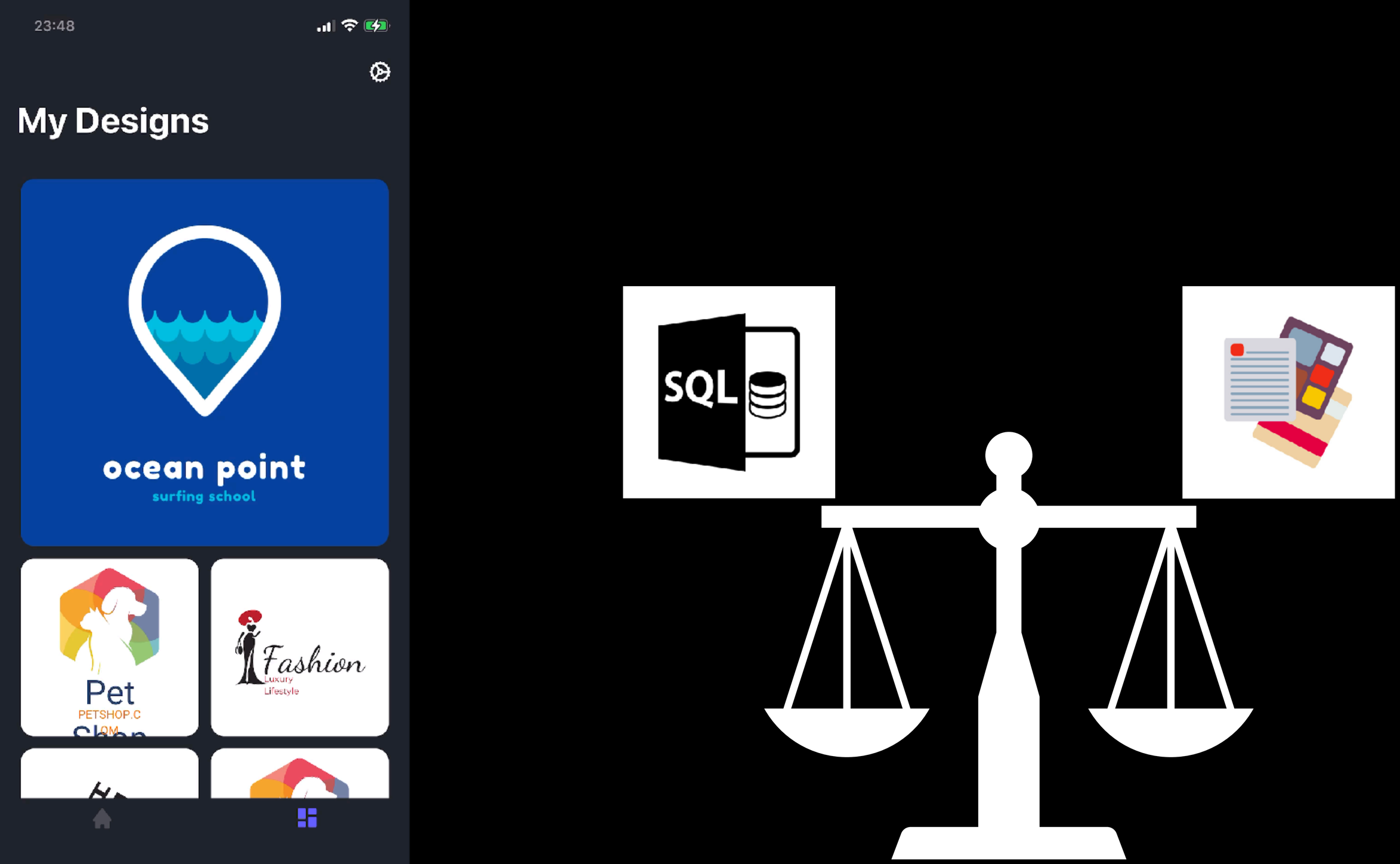

Adobe Illustrator 2021
1.ai @ 47.8 % (RGB/Preview)
2.ai* @ 118.65 % (RGB/Preview)
CLEANING
YOUR SLOGAN
Можно выставить по double tap вертикальное центрирова
Из-за этой пустой области текст уедет
Properties
Layers
Libraries
Type
Transform
Appearance
Fill
Stroke
Opacity
100%
Character
JosefinSans Bold
Bold
50 pt
(60 pt)
Auto
0
Paragraph
Align
Quick Actions
Create Outlines
Arrange
118.65%
Selection

Что такое Shaped ?

Что такое Shaped ?

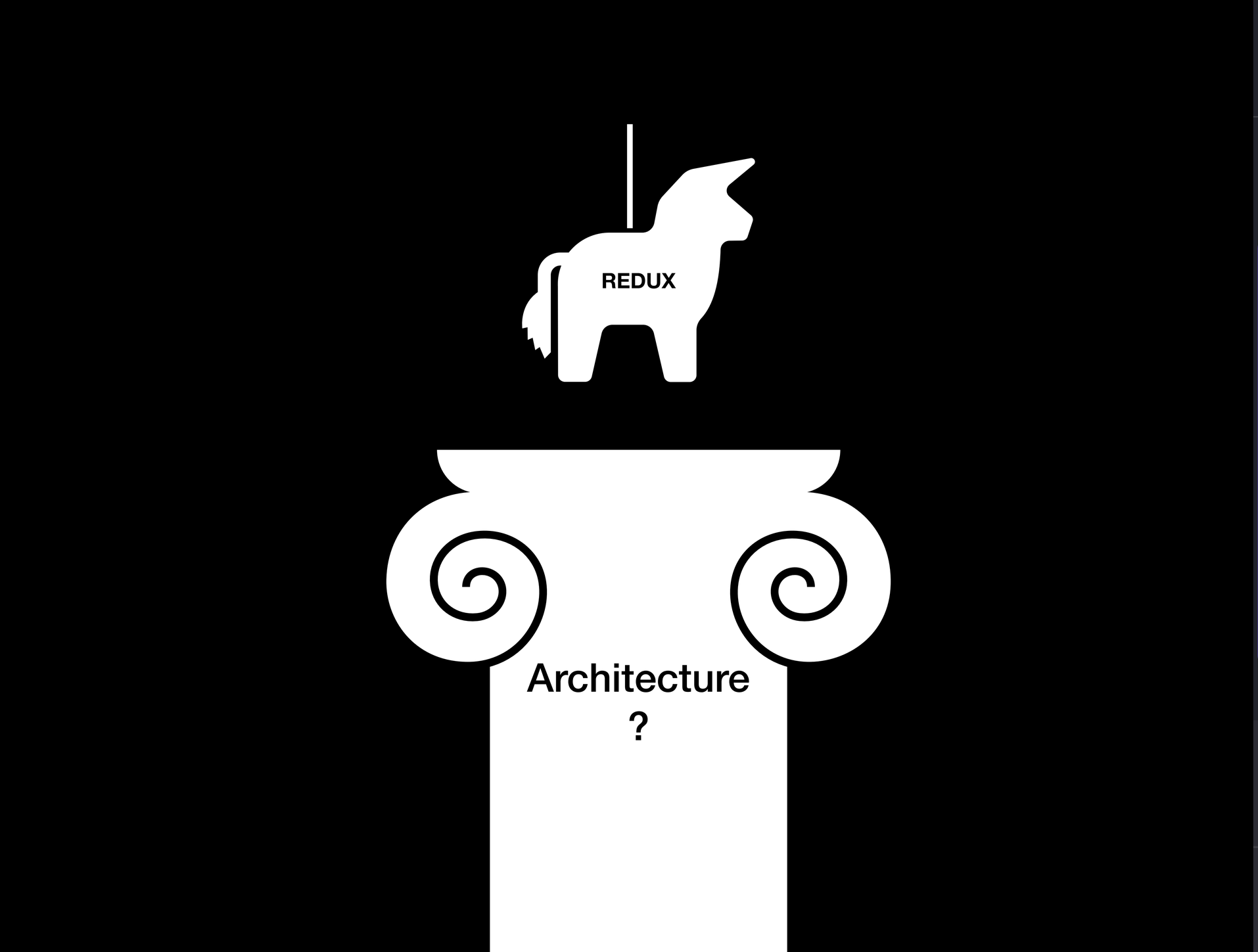


Что такое Shaped ?

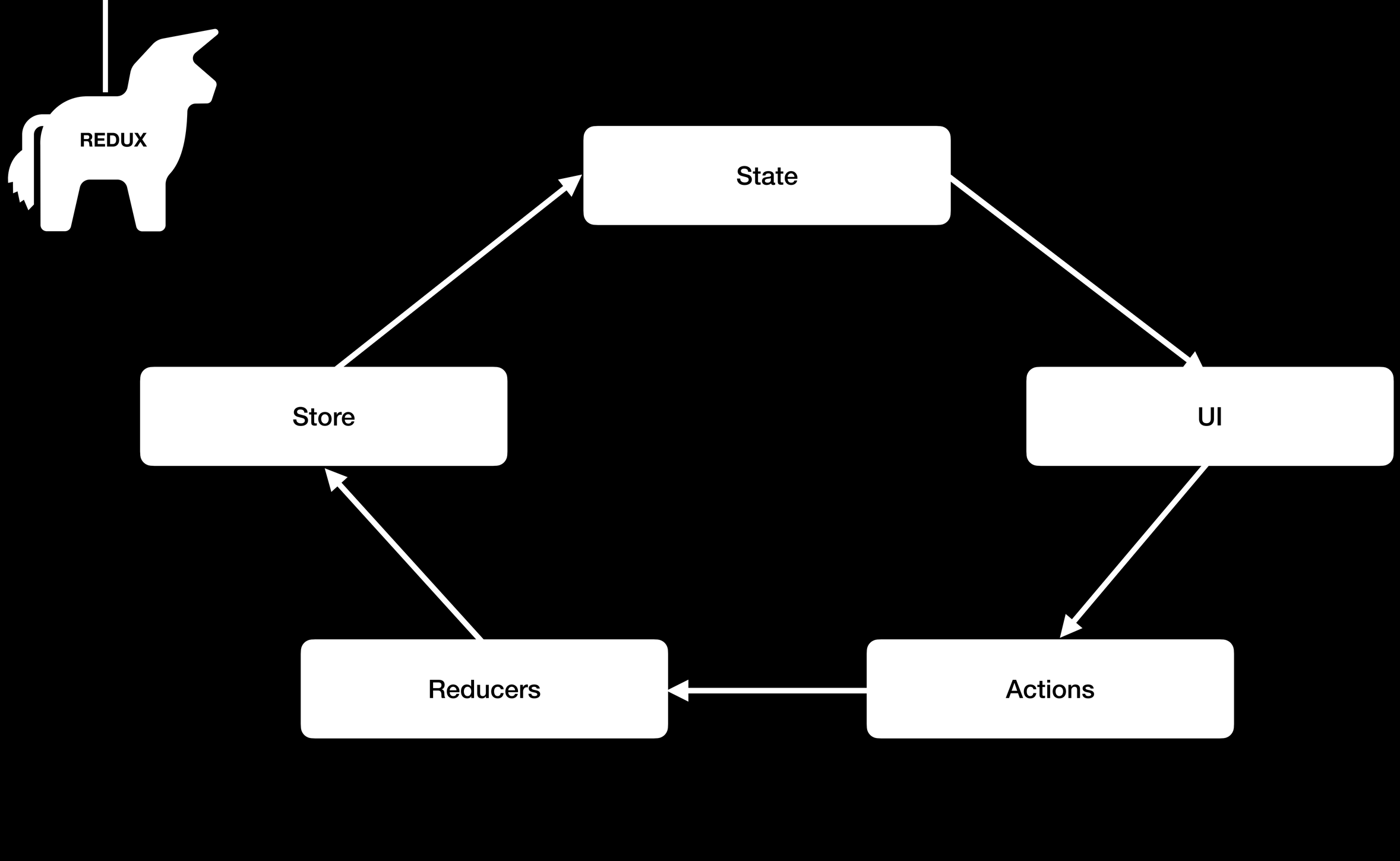

23:48
My Designs
ocean point
surfing school
Pet
PETSHOP.C
Fashion
Luxury
Lifestyle
SQL

О чем будем говорить

- ~~Спикеры~~
- ~~Почему KMM~~
- ~~Shaped~~
- Архитектура редактора
- Проблемы iOS
- Проблемы Android
- Выводы


REDUX
Architecture
?


9:41
Logo
CODEBREW
42
Font
Size
Style


REDUX
State
Store
UI
Reducers
Actions

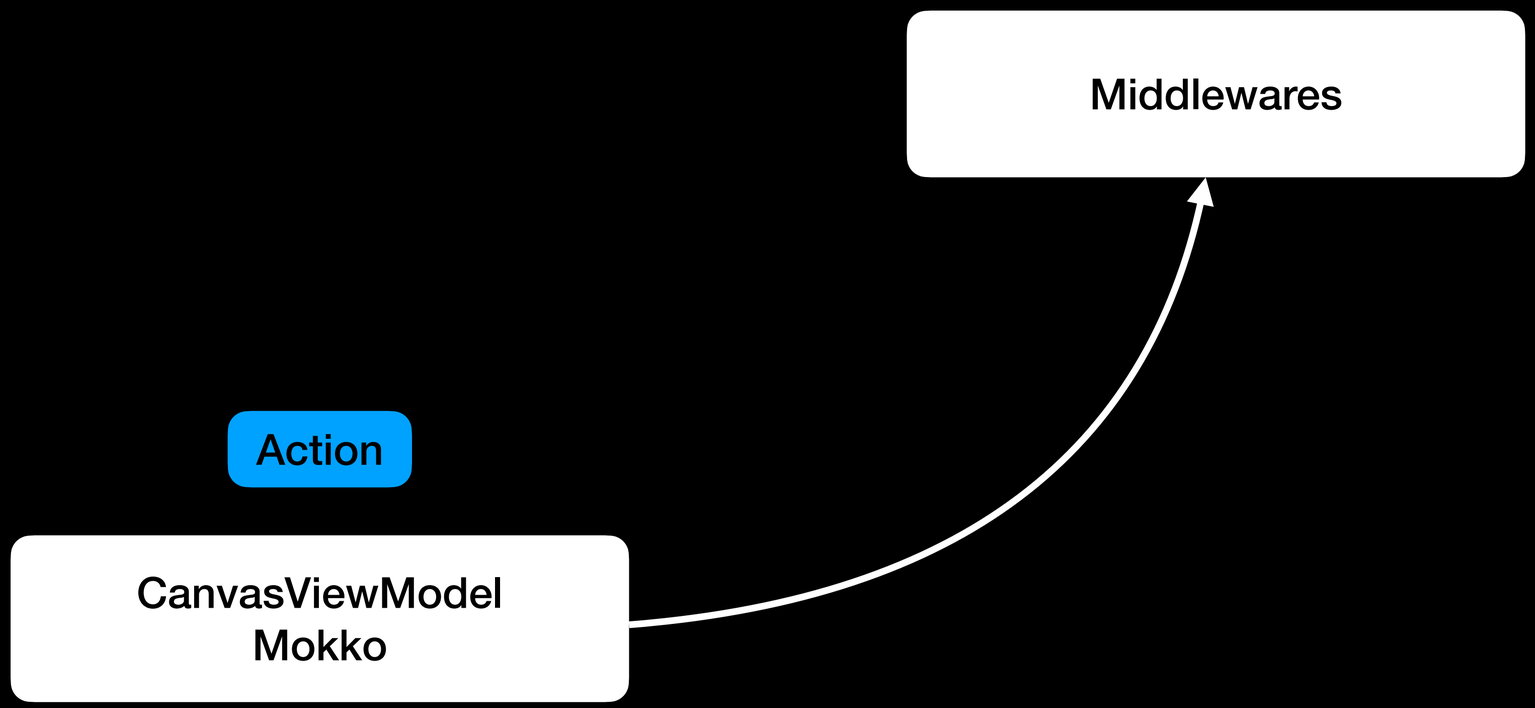
Action

```
sealed class CanvasAction

data class ForceSetStateAction(val state: CanvasState): CanvasAction()

data class SelectLayerAction(val layerID: String?): CanvasAction()
data class ChangeLayerTransformAction(val layerID: String?, val transform: Transform): CanvasAction()
data class AddLayerAction(val layer: Layer): CanvasAction()
data class ChangeOpacityAction(val layerID: String, val opacity: Float): CanvasAction()
data class IncreaseZOrderAction(val layerID: String): CanvasAction()
data class DecreaseZOrderAction(val layerID: String): CanvasAction()
data class OverrideColorsAction(val layerID: String, val colors: List<ShapeColor>): CanvasAction()
data class ChangeFontSizeAction(val layerID: String, val transform: Transform, val fontSize: Float): CanvasAc
data class ChangeLetterSpacingAction(val layerID: String, val transform: Transform, val letterSpacing: Float)
data class ChangeLineSpacingAction(val layerID: String, val transform: Transform, val lineSpacing: Float): Ca
data class ChangeTextAlignmentAction(val layerID: String, val alignment: TextAlignment): CanvasAction()
data class ChangeTextAction(val layerID: String, val text: String, val transform: Transform): CanvasAction()
data class ChangeFontAction(val layerID: String, val fontName: String, val transform: Transform): CanvasActio
data class ChangeTextStyles(val layerID: String, val textStyles: List<TextStyles>, val transform: Transform):
data class DeleteLayerAction(val layerID: String): CanvasAction()
data class DuplicateLayerAction(val layerID: String): CanvasAction()
data class ReorderLayersAction(val newOrderIDs: List<String>): CanvasAction()
data class ApplyEffect(val layerID: String, val effect: EffectKMM?): CanvasAction()
data class ChangeTextBendingAction(val layerID: String, val transform: Transform, val bending: Float, val tex
data class SetBackgroundLayerAction(val backgroundLayer: Layer?): CanvasAction()
```

Action

CanvasViewModel
Mokko

```
class CanvasViewModel(
    private val canvasStore: CanvasDispatcher,
    private val differenceDefiner: LayersDifferencesDefiner
) : ViewModel() {

    private val _state: MutableLiveData<CanvasState> = MutableLiveData(CanvasState())
    val state: LiveData<CanvasState> = _state

    fun dispatch(action: CanvasAction) {
        canvasStore.dispatch(action)
    }

    fun layersDiff(): LayersDifferences {
        return layersDiff
    }
}
```


Middlewares
Action
CanvasViewModel
Mokko

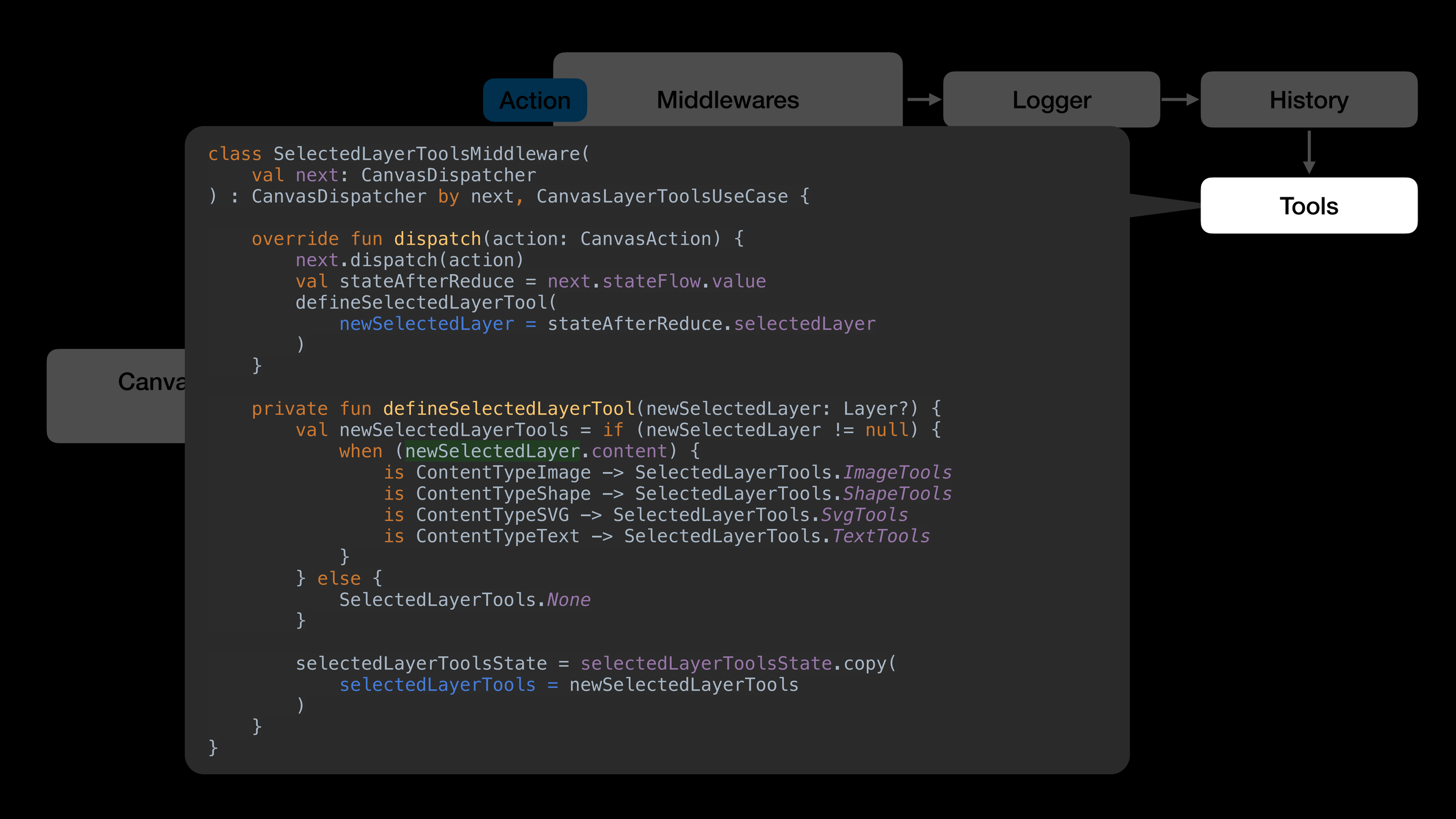
Middlewares

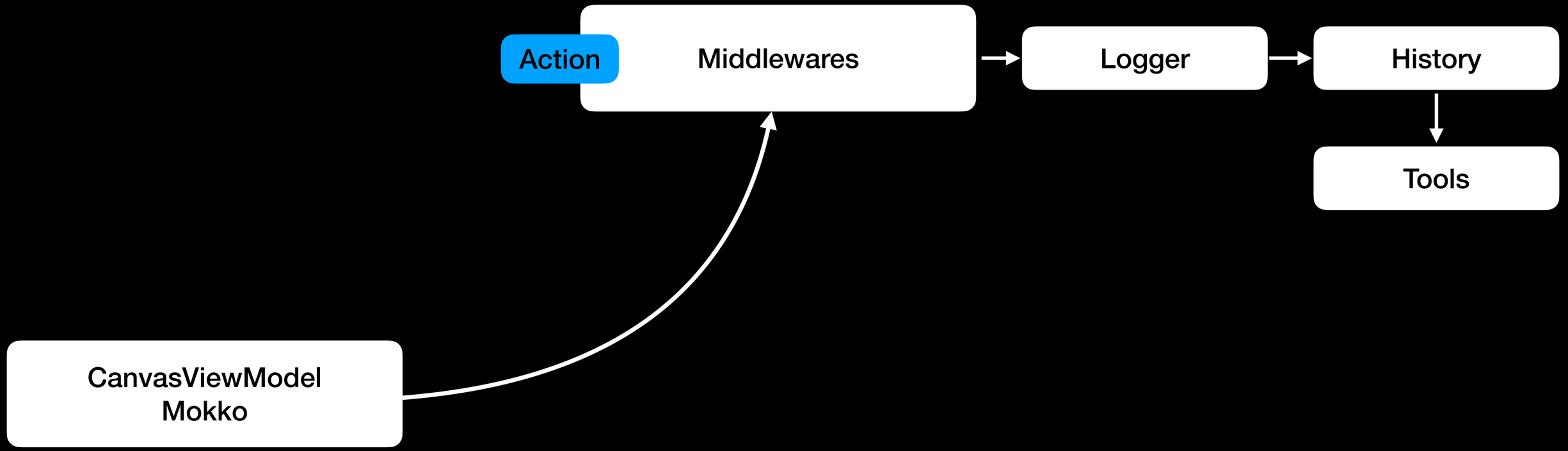
Logger

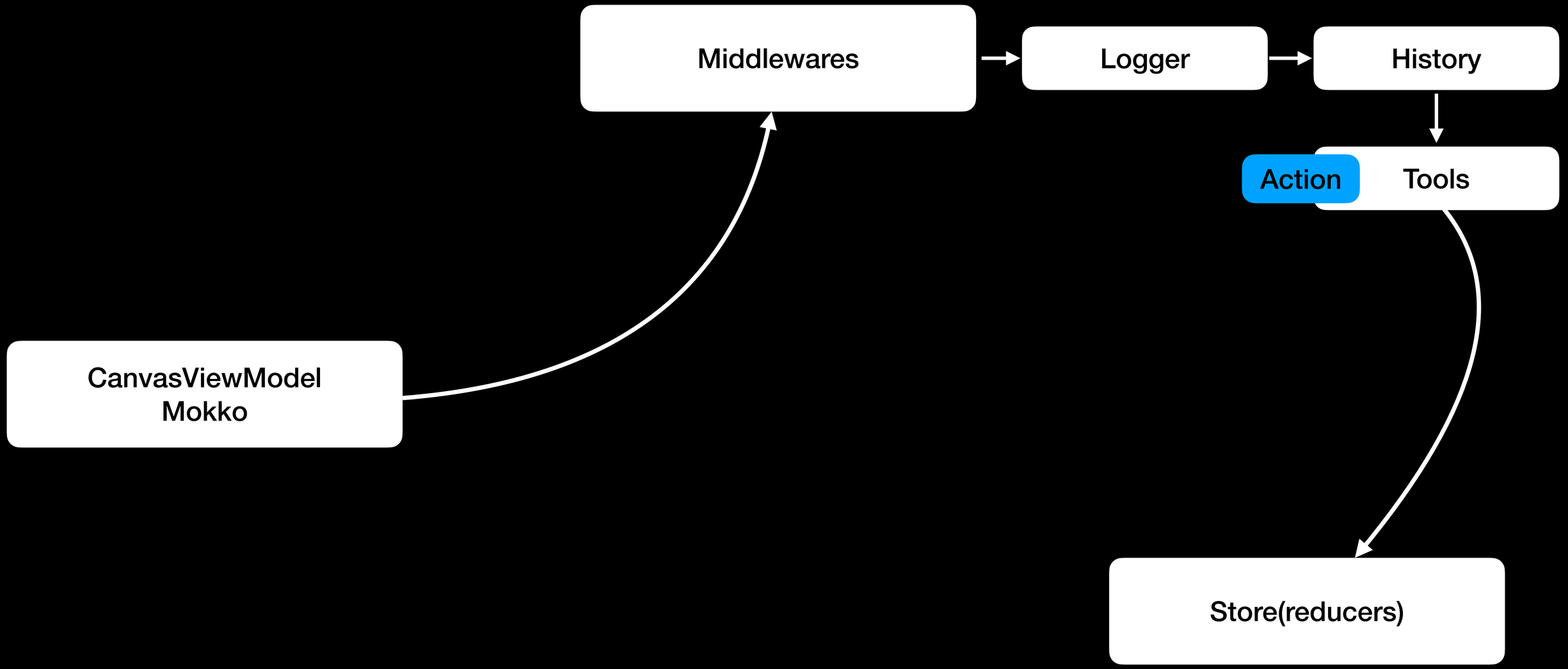
CanvasViewModel
Mokko

```
class CanvasLoggerMiddleware(
    val next: CanvasDispatcher
): CanvasDispatcher by next {
    override fun dispatch(action: CanvasAction) {
        Napier.d("Action was dispatched ${action::class}\n")
        println("Action was dispatched ${action::class}\n")
        next.dispatch(action)
    }
}
```

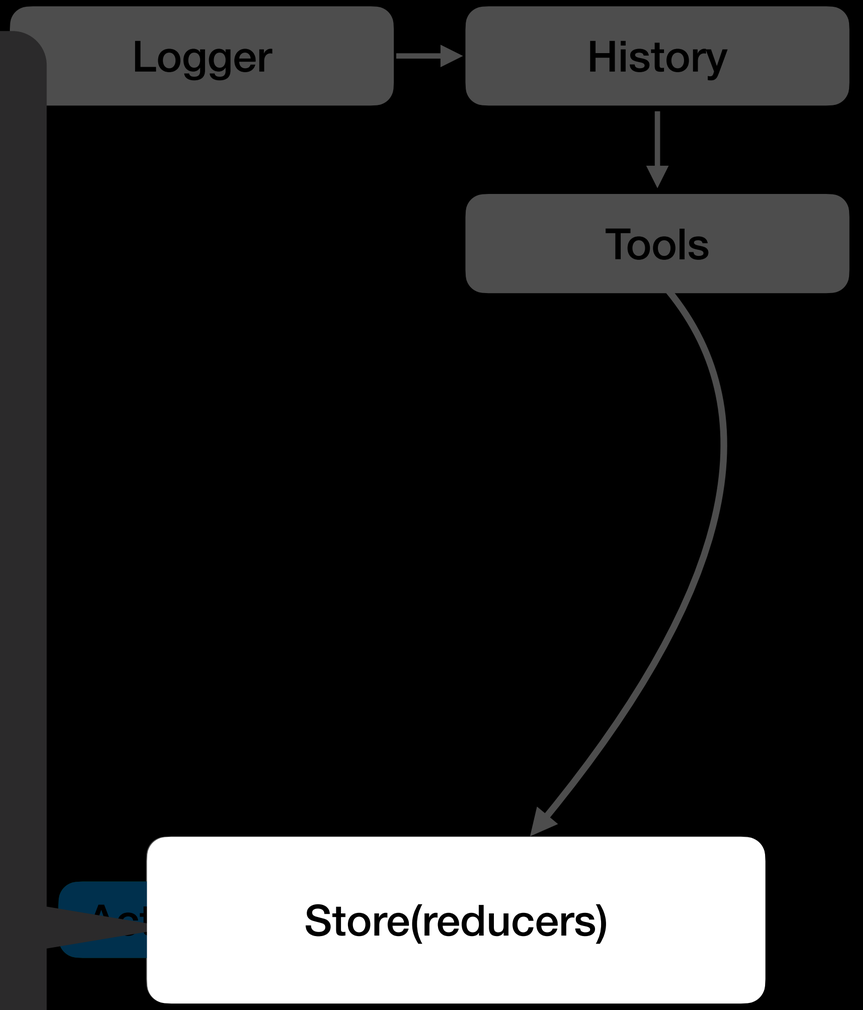

Action
Middlewares
Logger
History
Canva
class CanvasHistoryMiddleware(
val next: CanvasDispatcher
) : CanvasDispatcher by next, UndoRedoUseCase {
override fun dispatch(action: CanvasAction) {
val shouldSaveAction = shouldSave(action)
if (shouldSaveAction) {
// Some code
}
// Forward to the actual store
next.dispatch(action)
}
override fun undo() { }
override fun redo() { }
override fun dropHistory() {}
private fun forwardNewState(offset: Int) {}
private fun shouldSave(action: CanvasAction): Boolean {
return when(action) {
is ForceSetStateAction -> false
is SelectLayerAction -> false
is SetSVGColorsAction -> false
is SetTransformsAction -> false
else -> true
}
}
}

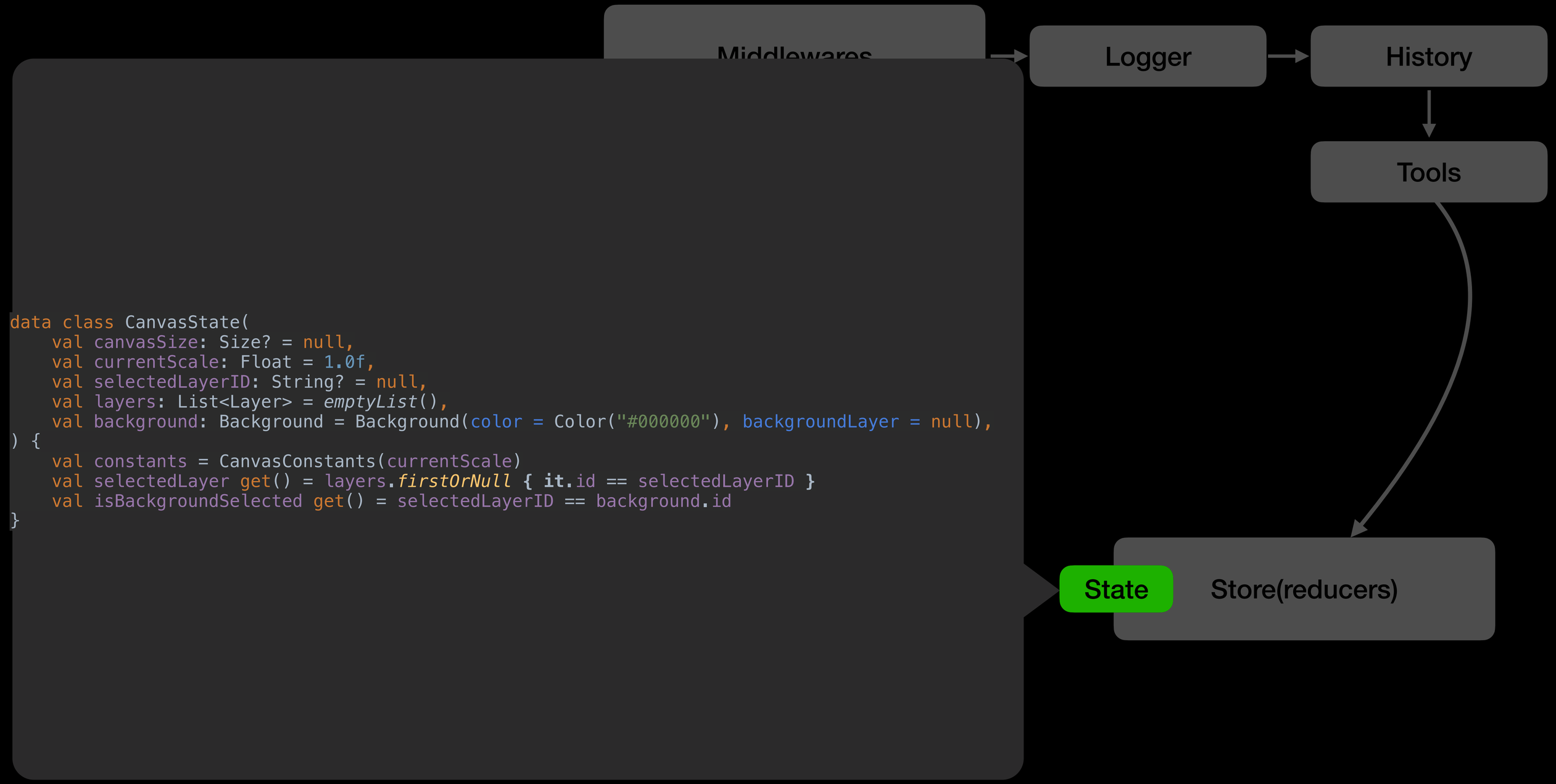

Action
Middlewares
Logger
History
Tools
Canva
class SelectedLayerToolsMiddleware(
val next: CanvasDispatcher
) : CanvasDispatcher by next, CanvasLayerToolsUseCase {
override fun dispatch(action: CanvasAction) {
next.dispatch(action)
val stateAfterReduce = next.stateFlow.value
defineSelectedLayerTool(
newSelectedLayer = stateAfterReduce.selectedLayer
)
}
private fun defineSelectedLayerTool(newSelectedLayer: Layer?) {
val newSelectedLayerTools = if (newSelectedLayer != null) {
when (newSelectedLayer.content) {
is ContentTypeImage -> SelectedLayerTools.ImageTools
is ContentTypeShape -> SelectedLayerTools.ShapeTools
is ContentTypeSVG -> SelectedLayerTools.SvgTools
is ContentTypeText -> SelectedLayerTools.TextTools
}
} else {
SelectedLayerTools.None
}
selectedLayerToolsState = selectedLayerToolsState.copy(
selectedLayerTools = newSelectedLayerTools
)
}
}


Action
Middlewares
Logger
History
Tools
CanvasViewModel
Mokko

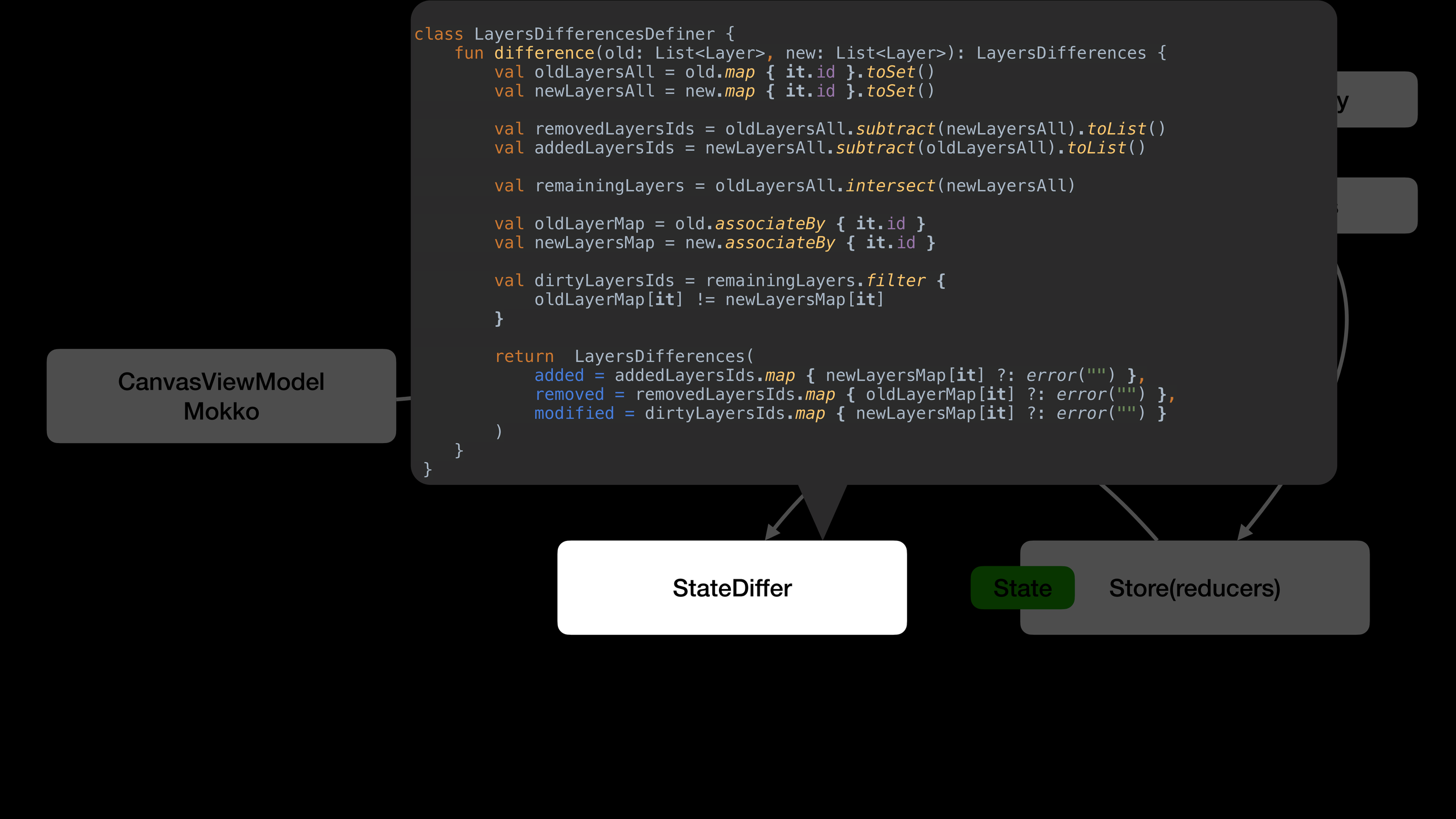

Middlewares
Logger
History
Action
Tools
CanvasViewModel
Mokko
Store(reducers)

```kotlin
class CanvasStoreDefault : CanvasStore {
    private val reducer = CanvasActionsReducer()

    private val _stateFlow = MutableStateFlow(CanvasState())
    override val stateFlow = _stateFlow

    override fun dispatch(action: CanvasAction) {
        _stateFlow.value = when (action) {
            is AddLayerAction -> reducer.reduce(_stateFlow.value, action)
            is SelectLayerAction -> reducer.reduce(_stateFlow.value, action)
            is ChangeLayerTransformAction -> reducer.reduce(_stateFlow.value, action)
            is ChangeOpacityAction -> reducer.reduce(_stateFlow.value, action)
            is ForceSetStateAction -> reducer.reduce(action)
        }
    }
}
```

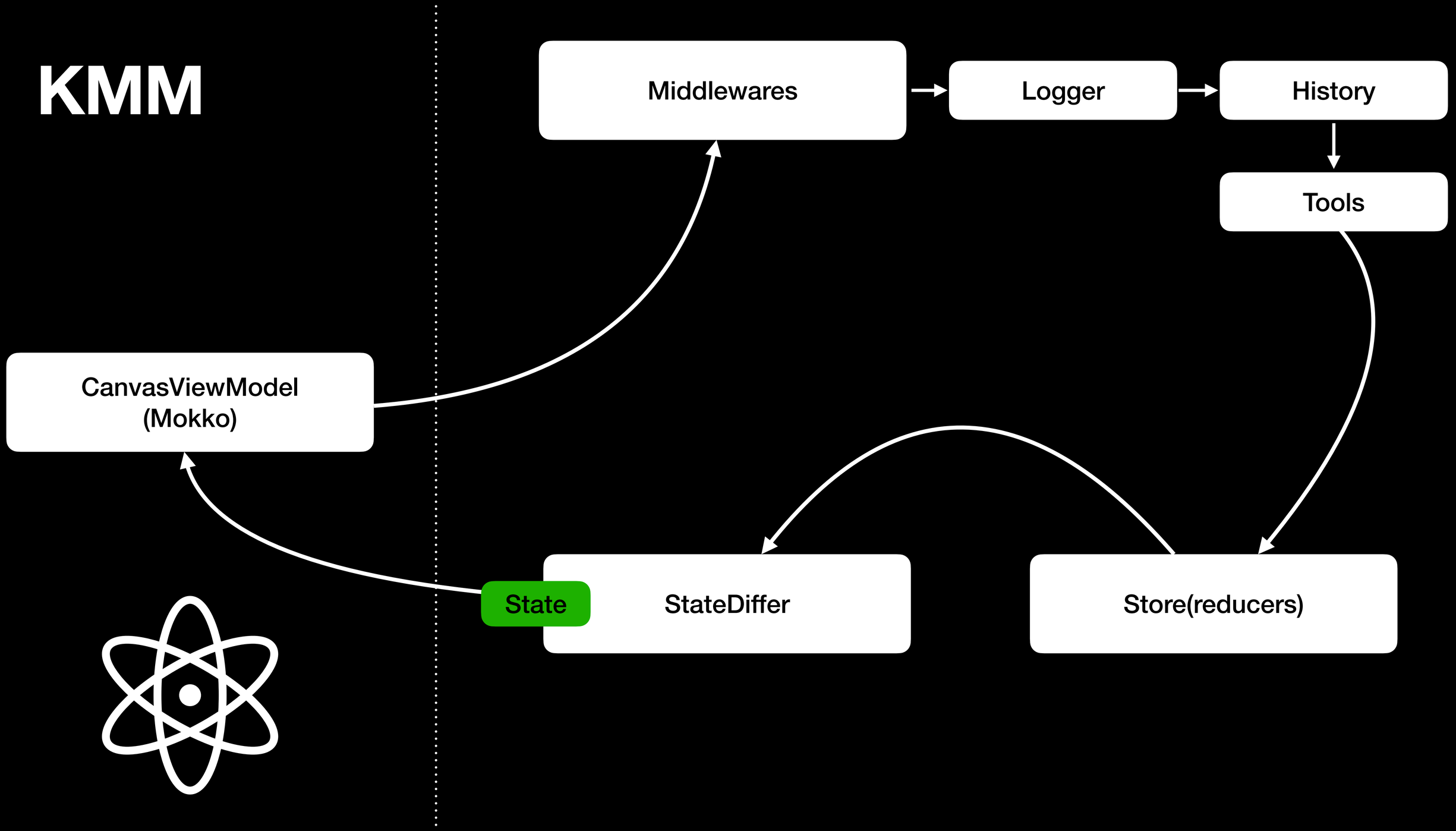


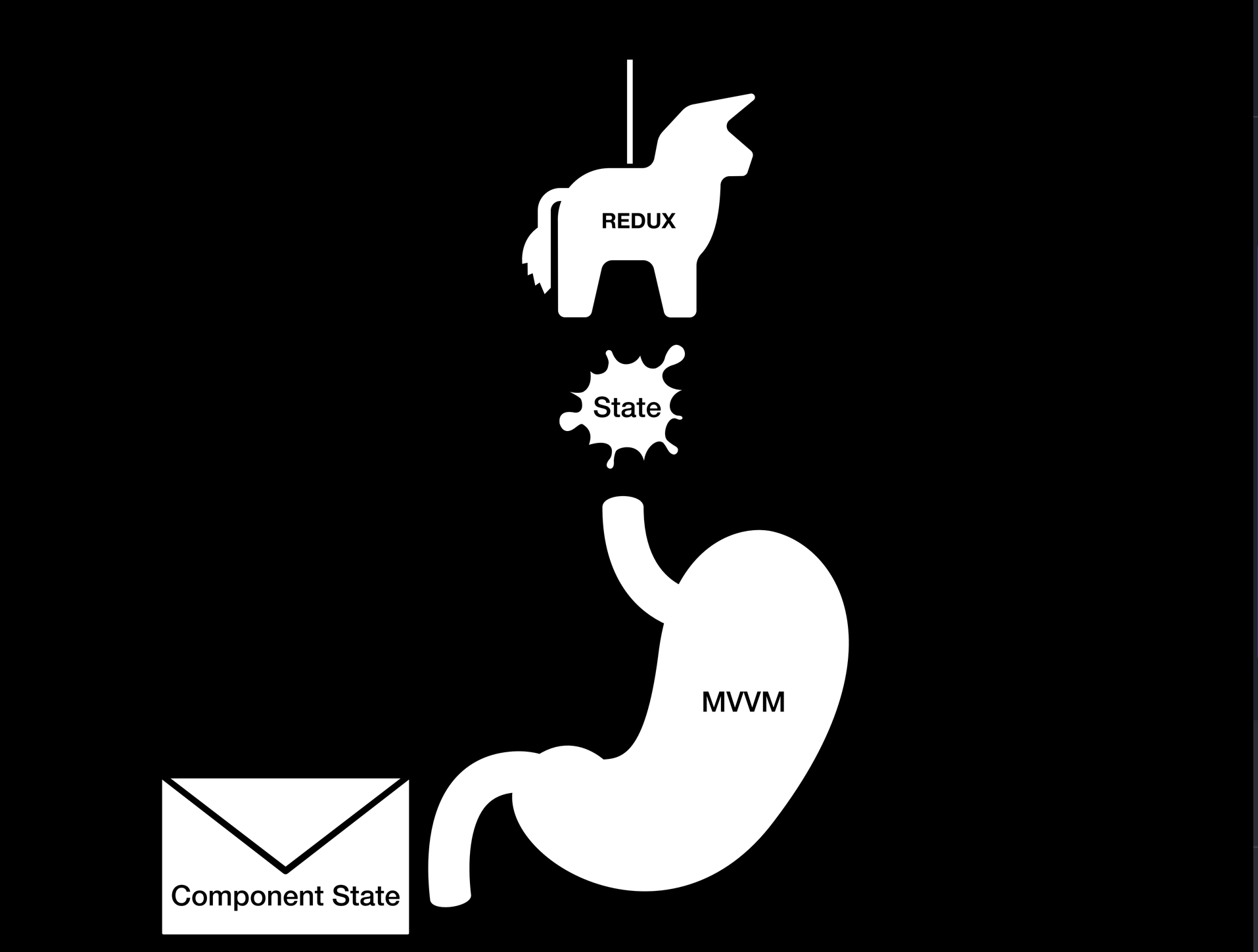



```
data class CanvasState(
    val canvasSize: Size? = null,
    val currentScale: Float = 1.0f,
    val selectedLayerID: String? = null,
    val layers: List<Layer> = emptyList(),
    val background: Background = Background(color = Color("#000000"), backgroundLayer = null),
) {
    val constants = CanvasConstants(currentScale)
    val selectedLayer get() = layers.firstOrNull { it.id == selectedLayerID }
    val isBackgroundSelected get() = selectedLayerID == background.id
}
```

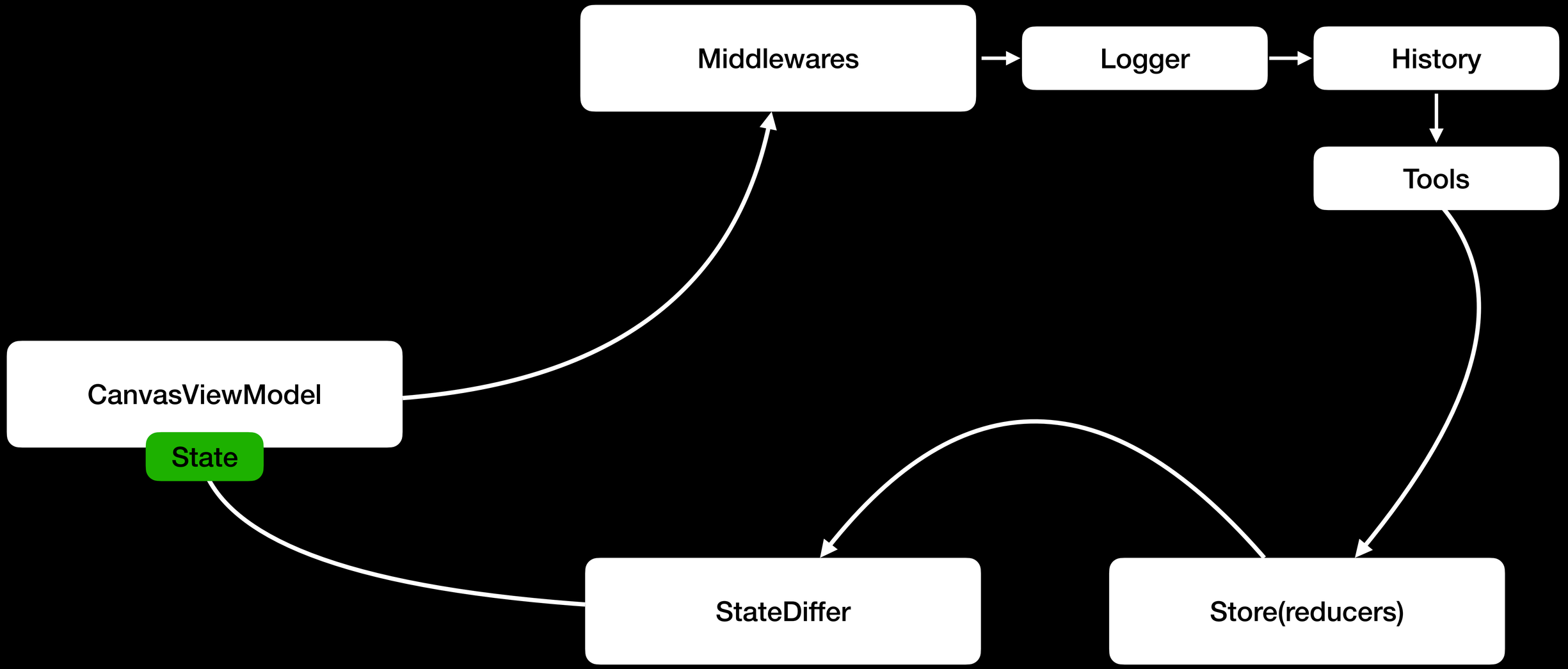

class LayersDifferencesDefiner {
fun difference(old: List<Layer>, new: List<Layer>): LayersDifferences {
val oldLayersAll = old.map { it.id }.toSet()
val newLayersAll = new.map { it.id }.toSet()
val removedLayersIds = oldLayersAll.subtract(newLayersAll).toList()
val addedLayersIds = newLayersAll.subtract(oldLayersAll).toList()
val remainingLayers = oldLayersAll.intersect(newLayersAll)
val oldLayerMap = old.associateBy { it.id }
val newLayersMap = new.associateBy { it.id }
val dirtyLayersIds = remainingLayers.filter {
oldLayerMap[it] != newLayersMap[it]
}
return LayersDifferences(
added = addedLayersIds.map { newLayersMap[it] ?: error("") },
removed = removedLayersIds.map { oldLayerMap[it] ?: error("") },
modified = dirtyLayersIds.map { newLayersMap[it] ?: error("") }
)
}
}
CanvasViewModel
Mokko
StateDiffer
State
Store(reducers)

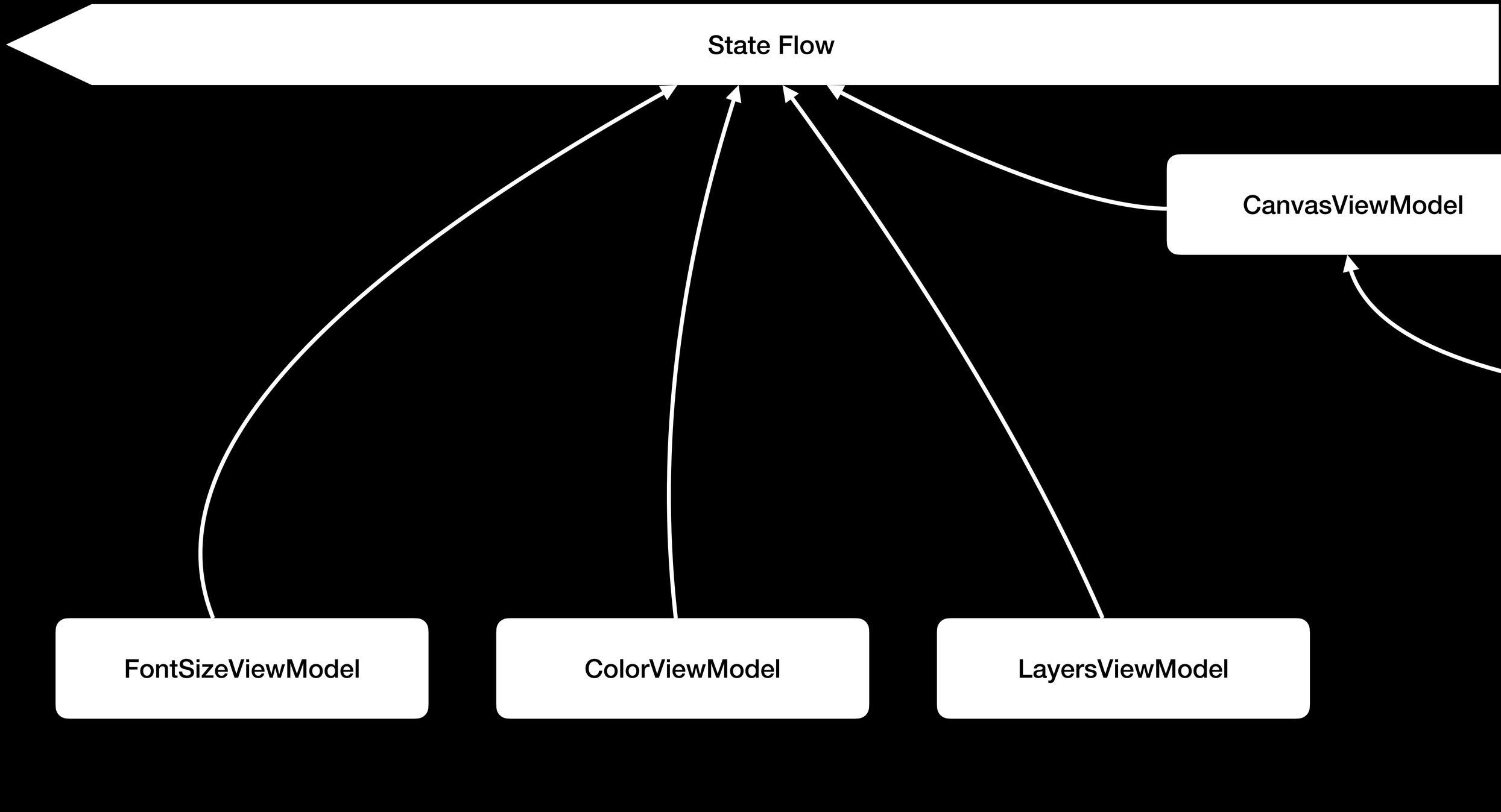
KMM


Middlewares
Logger
History
Tools
CanvasViewModel
(Mokko)
State
StateDiffer
Store(reducers)


REDUX
State
MVVM
Component State

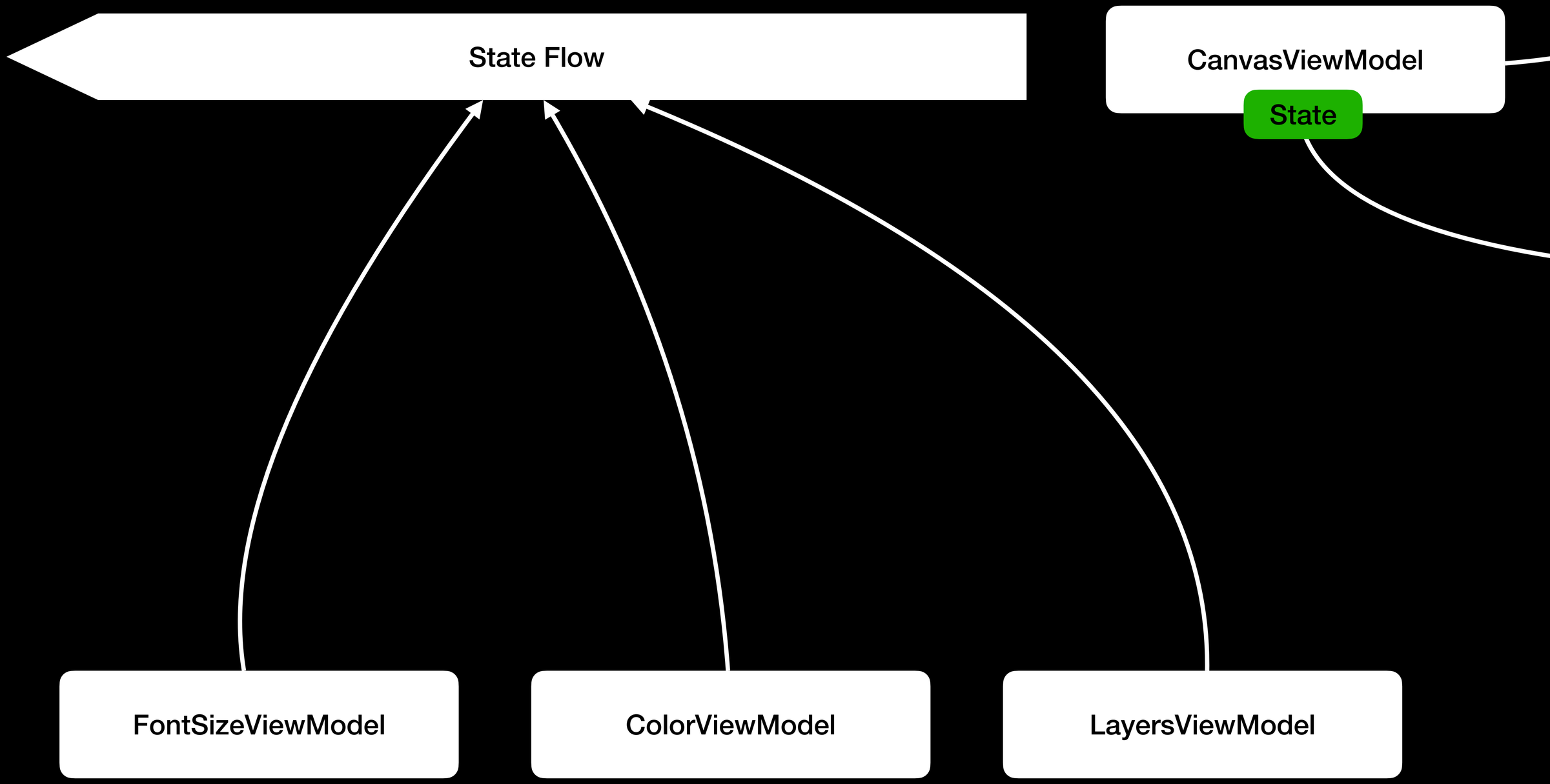

9:41
Logo
CODEBREW
42
Font
Size
Style


Middlewares
Logger
History
Tools
CanvasViewModel
State
StateDiffer
Store(reducers)


State Flow
CanvasViewModel
FontSizeViewModel
ColorViewModel
LayersViewModel

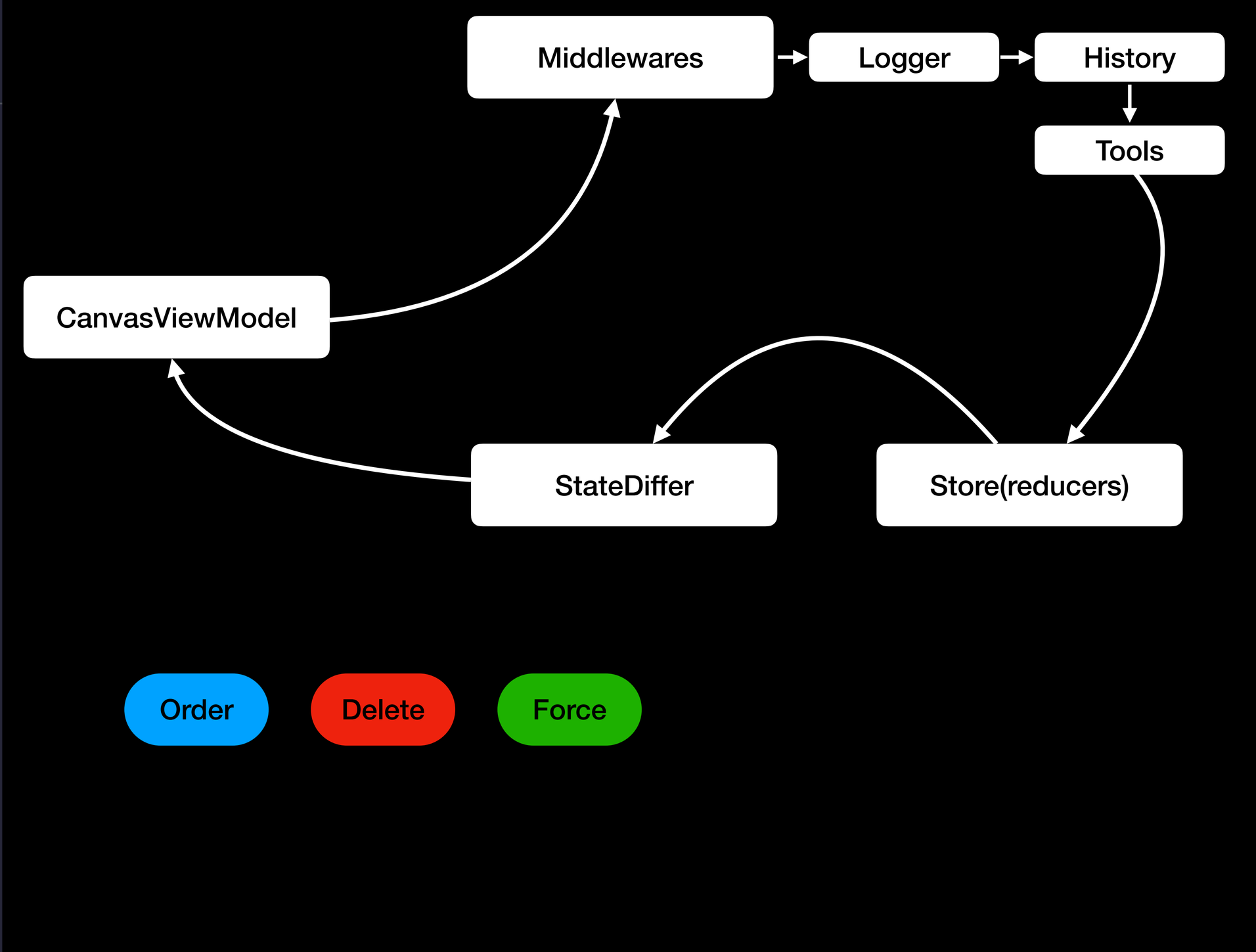
О чем будем говорить

- ~~Спикеры~~
- ~~Почему KMM~~
- ~~Shaped~~
- ~~Архитектура редактора~~
- Проблемы iOS
- Проблемы Android
- Выводы


State Flow
CanvasViewModel
State
FontSizeViewModel
ColorViewModel
LayersViewModel


Real Fans Club
BASEBALL


12:32
Real Fans Club
BASEBALL


Middlewares
Logger
History
Tools
CanvasViewModel
StateDiffer
Store(reducers)
Order
Delete
Force

Middlewares → Logger → History → Tools

CanvasViewModel

StateDiffer

Store(reducers)

Order

Delete

Force

Force

State

State

Order

Delete

Order

StateFlow

Store(reducers)

Order

Delete

Force

State

Force

State

Order

Delete

Order

StateFlow

Store(reducers)

Middlewares → Logger → History → Tools

State

Force

CanvasViewModel

StateDiffer

Store(reducers)

Mokko

- State optional

Mokko

- State optional
- Async

Mokko

```
actual open class LiveData<T>(initialValue: T) {
    private var storedValue: T = initialValue
    private val observers = mutableListOf<(T) -> Unit>()

    actual open val value: T
        get() = storedValue

    actual fun addObserver(observer: (T) -> Unit) {
        observer(value)
        observers.add(observer)
    }

    actual fun removeObserver(observer: (T) -> Unit) {
        observers.remove(observer)
    }

    protected fun changeValue(value: T) {
        storedValue = value

        observers.forEach { it(value) }
    }
}
```

SVG Colors

20:07

My Designs

SVG

20:07

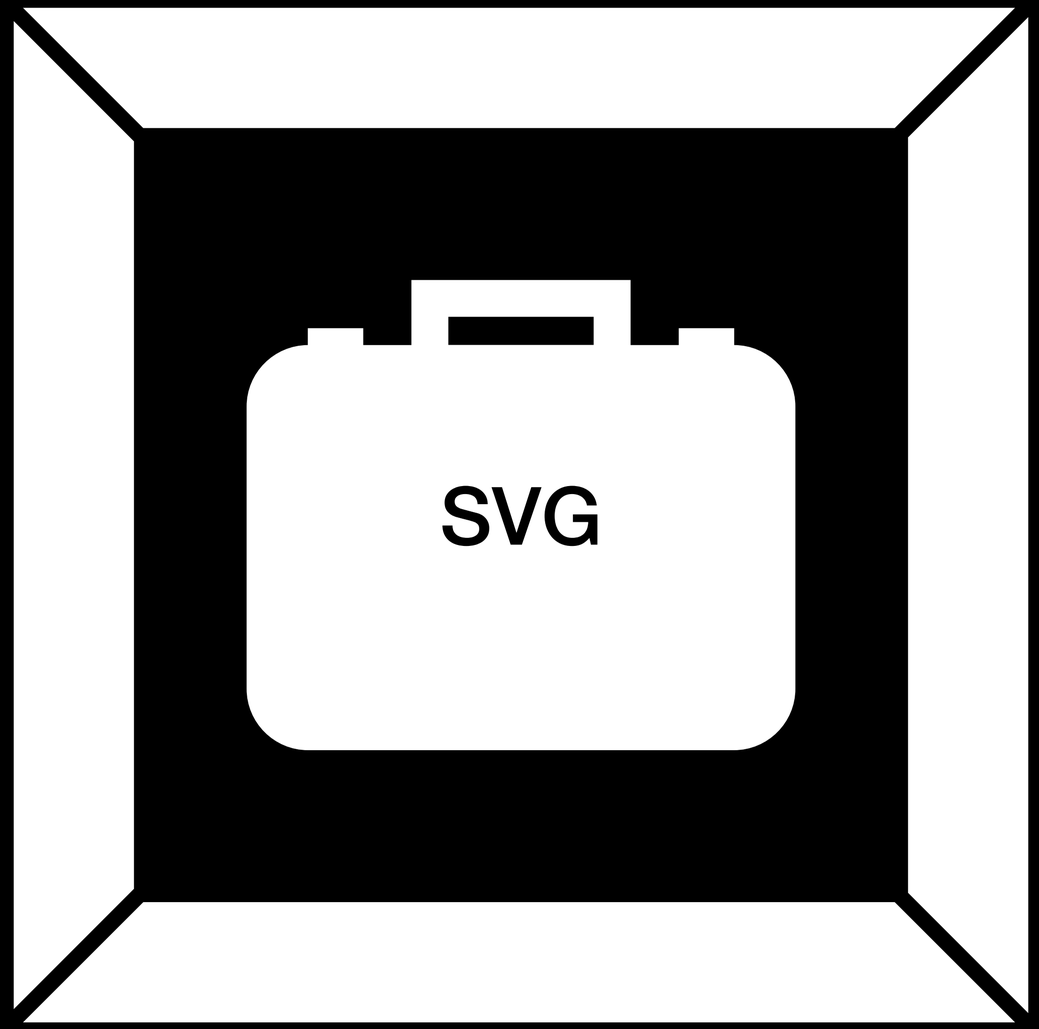
My Designs

SVG

?

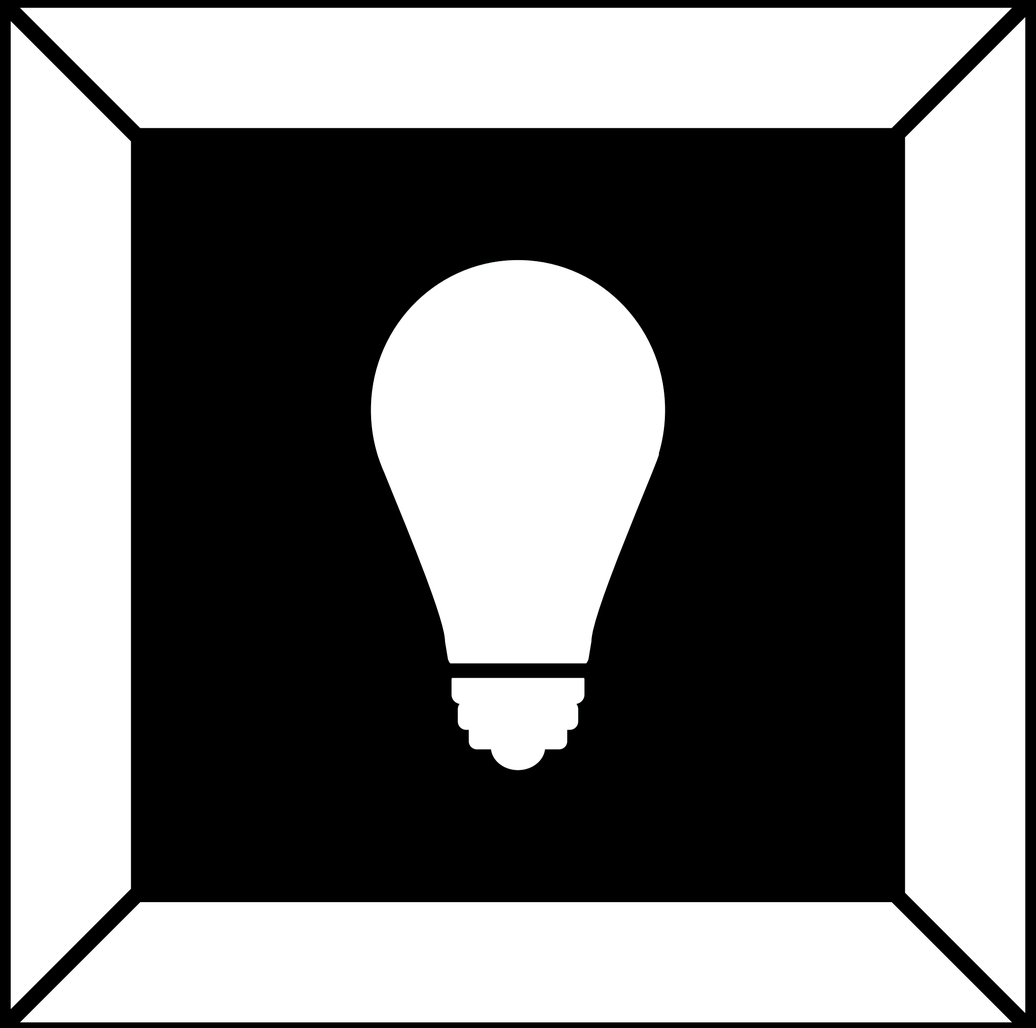
20:07

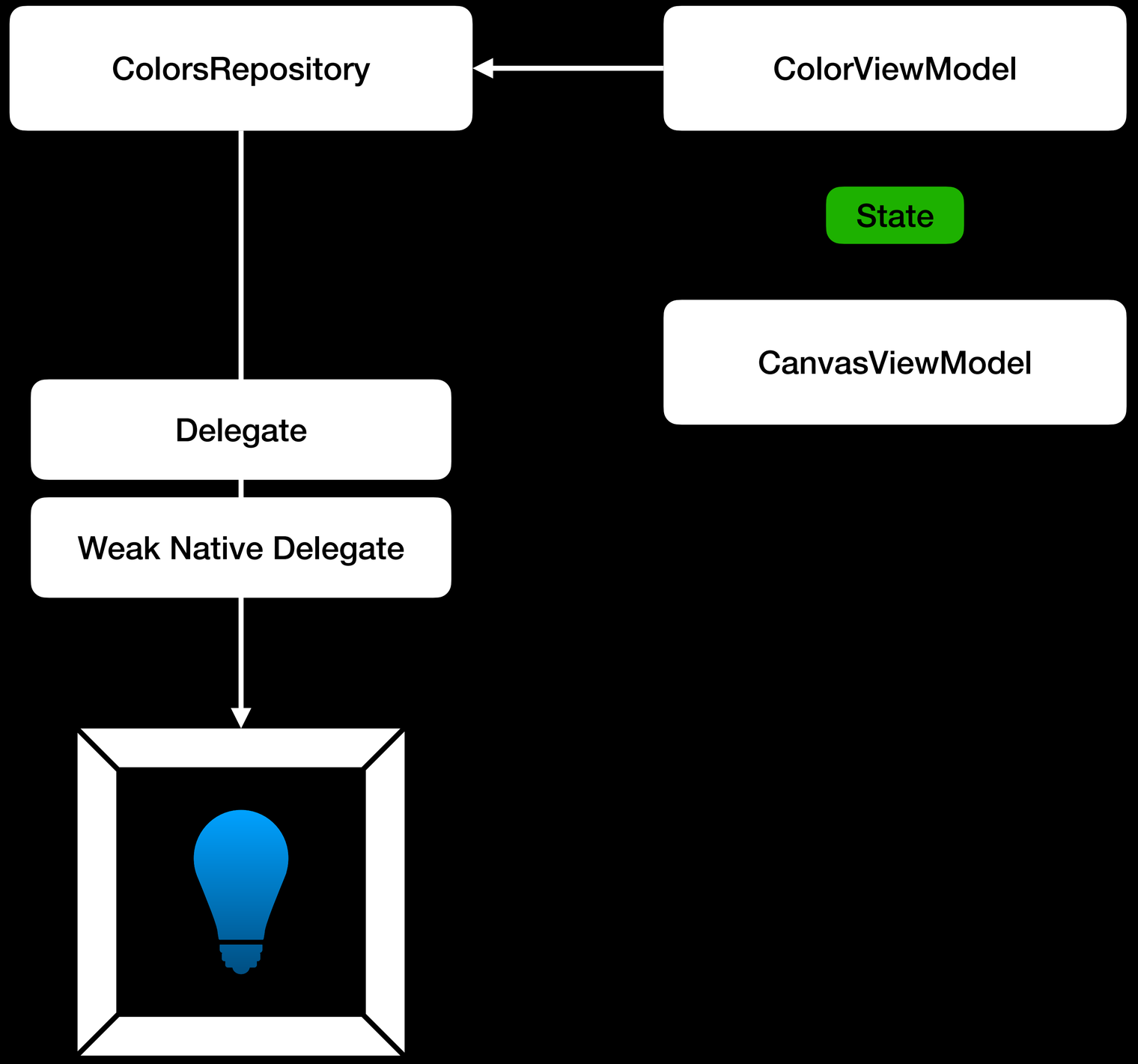
My Designs



20:07

My Designs

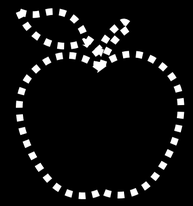



ColorsRepository
ColorViewModel
State
CanvasViewModel
Delegate
Weak Native Delegate

Presentation

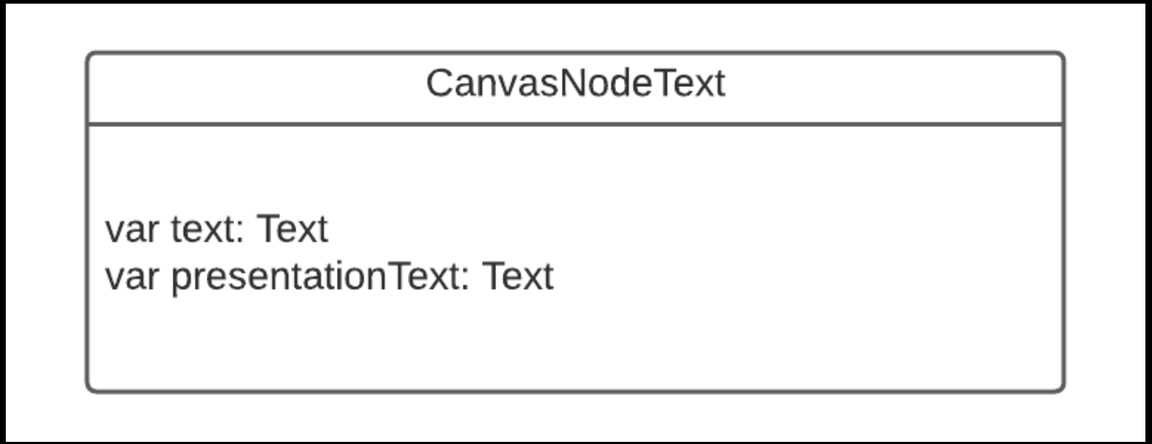
Presentation

Model



Частично решили на UI layer

- Но, нужна ссылка на View, что не очень удобно и верно



Errors

```
@Throws(Exception::class)
fun fromJSON(json: String): Document {
    return Json {
        ignoreUnknownKeys = true
        allowStructuredMapKeys = true
    }.decodeFromString(json)
}
```

Компиляция под iOS != Android

65535

Частый clean build

- CocoaPods
- SPM
- Gradle

Трансляция в Swift

- Switch vs When
- Lowercased enums
- Copy is missing

```
let kmmDocument = try? Document.Companion().fromJSON(json: jsonString)
```

- Closures vs streams
- Garbage collection

Native performance?

- При дебаг зачастую огромный стэк вызовов

Pack for Xcode

```
val packForXcode by tasks.creating(Sync::class) {
    group = "build"
    var mode = System.getenv("CONFIGURATION") ?: "DEBUG"
    // KMM supports only two modes = Debug and Release
    // Only RELEASE builds generate bitcode symbols, which are needed to achieve iOS builds
    // So, all builds except Debug fallbacks to RELEASE
    if (mode.toLowerCase() == "debug") {
        mode = "DEBUG"
    } else {
        mode = "RELEASE"
    }
    val sdkName = System.getenv("SDK_NAME") ?: "iphonesimulator"
    val targetName = "ios" + if (sdkName.startsWith("iphoneos")) "Arm64" else "X64"
    val framework = kotlin.targets.getByName<KotlinNativeTarget>(targetName).binaries.getFramework(mode)
    inputs.property("mode", mode)
    dependsOn(framework.linkTask)
    val targetDir = File(buildDir, "xcode-frameworks")
    from({ framework.outputDirectory })
    into(targetDir)
}
```

Xcode/Appcode ломаются

```
    private func loadUserDesign(trigger: AnyPublisher<Void, Never>) -> AnyPublisher<Void, Never> {
        return trigger
            .handleEvents(receiveOutput: { _ in self.stateSubject.send(.loading) })
            .setFailureType(to: AppErrors.self)
            .flatMap { [documentsRepository] _ in documentsRepository.allDocumentsInDirectory() }
            .catch { _ in Just<[URL]>([]) }
            .map { urls in urls.map { LogoDocument(fileURL: $0) }}
            .map { documents in
                documents.map { document in
                    MyDesigns.CollectionView.Cell.Form(
                        document: document,
                        thumbnailFutureProvider: { [weak self, document] in
                        return self?.documentsRepository.requestThumbnail(for: document)
                    }
                )
            }}
            .handleEvents(receiveOutput: { urls in
                if urls.isEmpty {
                    self.stateSubject.send(.empty)
                }
                else {
                    self.stateSubject.send(.userLogos(urls))
                }
            })
            .map { _ in }
            .eraseToAnyPublisher()
    }
}
```

Failed to produce diagnostic for expression; please file a bug report

Multithreading

- Mutable state == 1 thread
- Immutable state == many threads
- Immutable and frozen state
- Kotlin 1.6.0

Базовые методы - toHex, toFloat

```
private fun tryToConvertToFloat(text: String?): Float? {
    return text?.replace(",", ".")?.toFloatOrNull()
}
```

Сложно избавиться

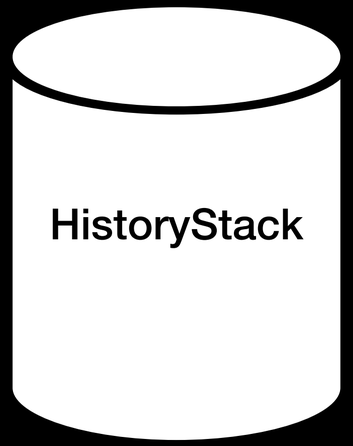
Struct StrokeEffect (enum)

State.Layers.Effect

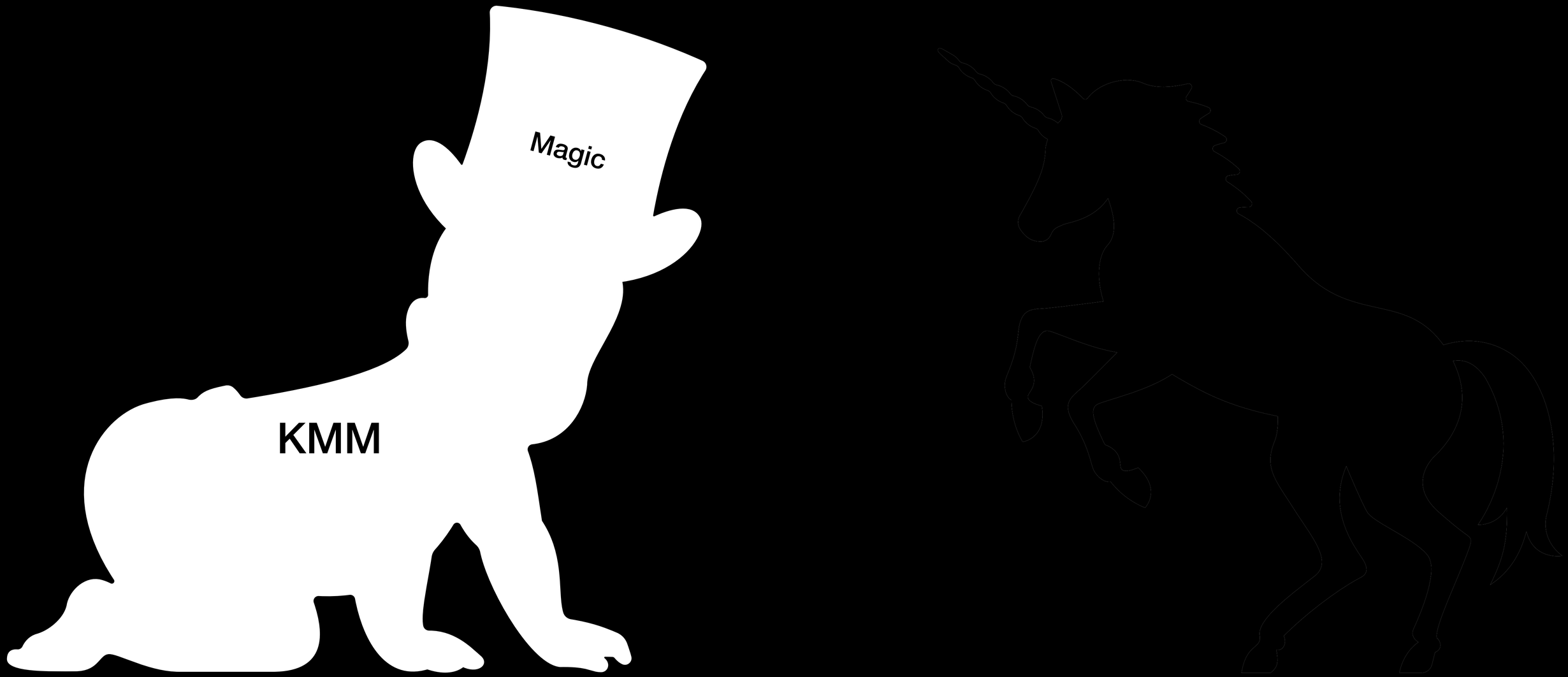
iOS

KMM

Interface Effect



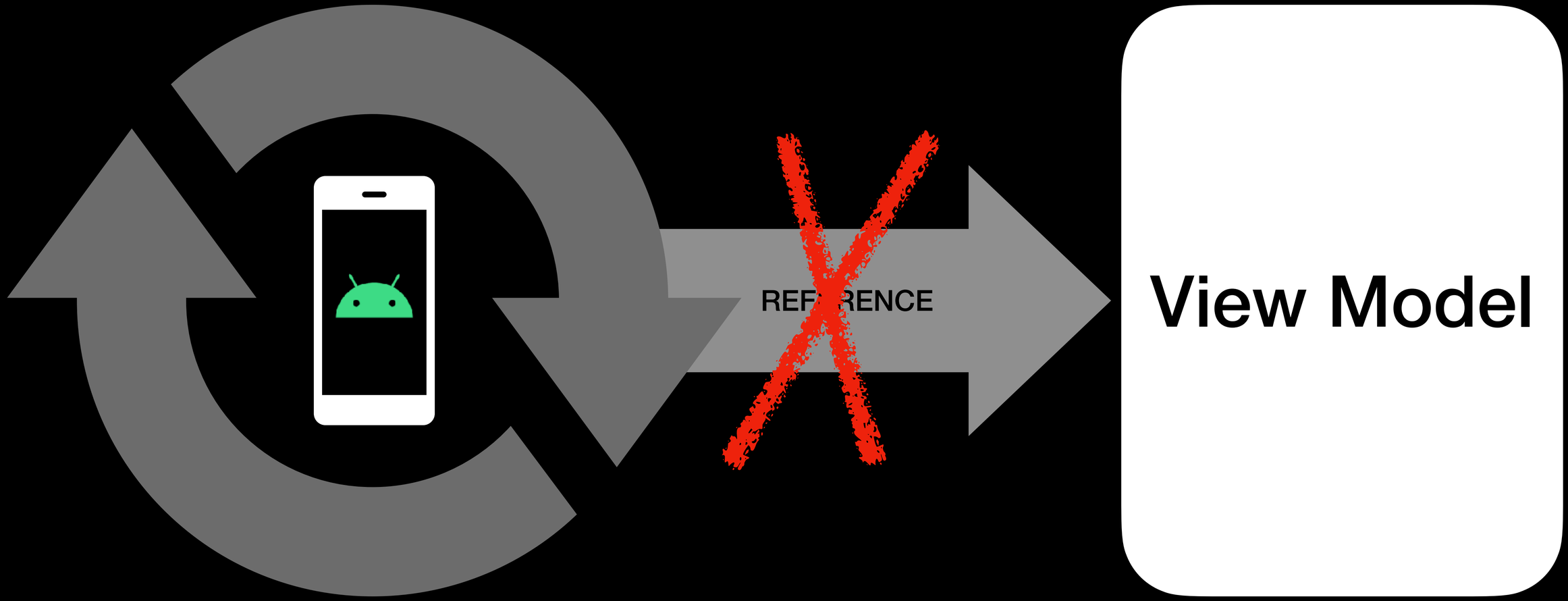
Впечатление от KMM iOS

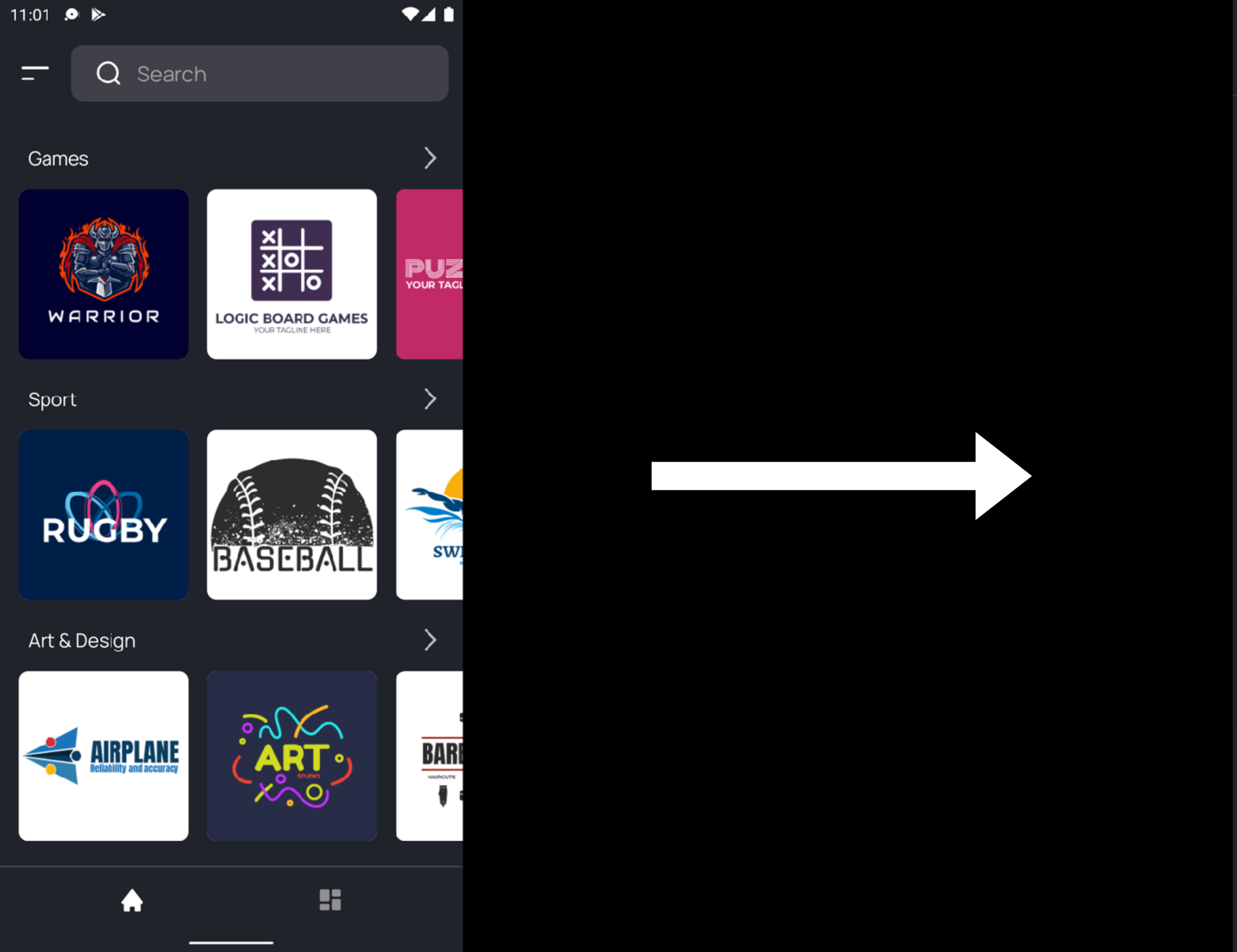


О чем будем говорить

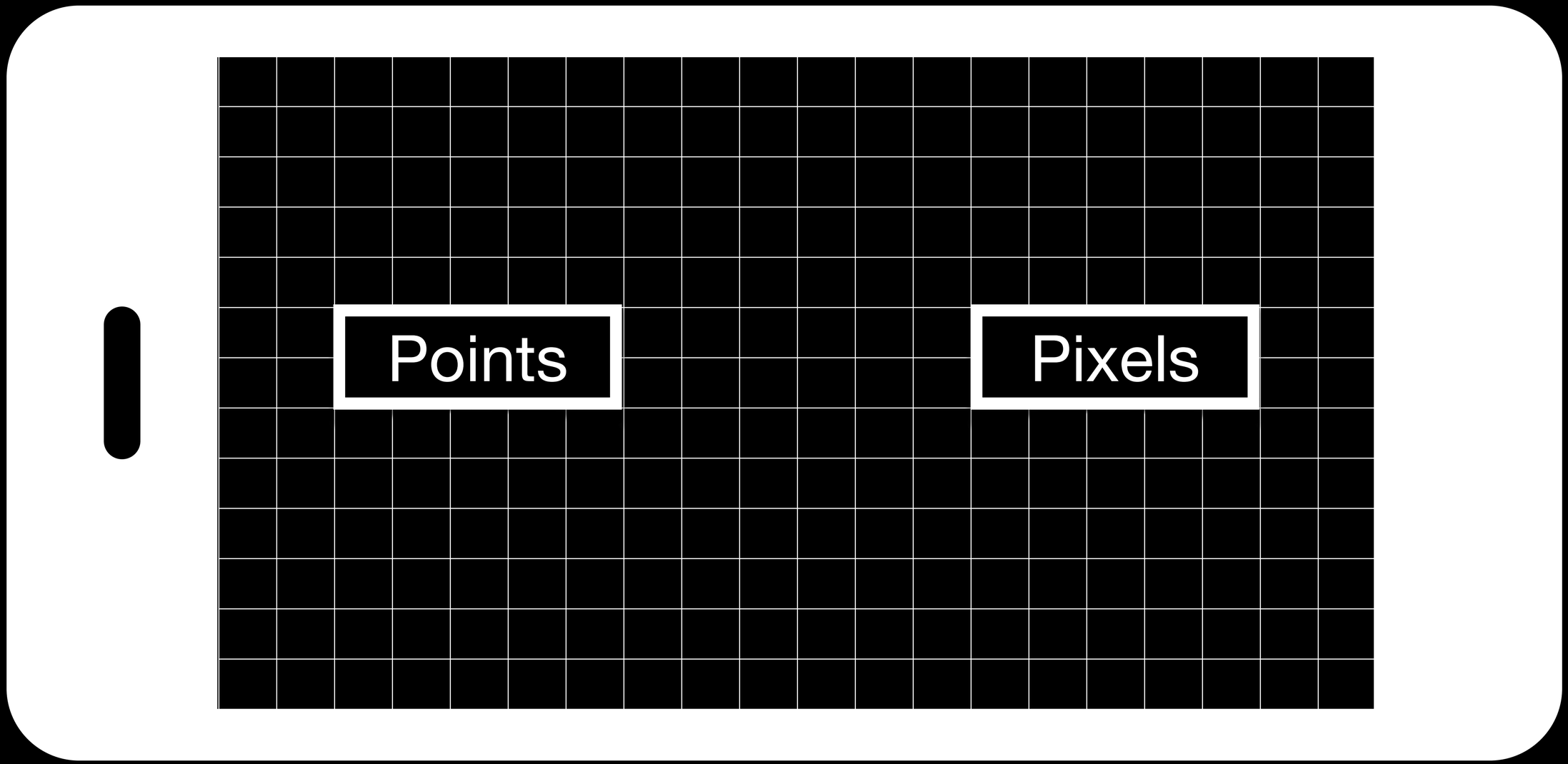
- ~~Спикеры~~
- ~~Почему KMM~~
- ~~Shaped~~
- ~~Архитектура редактора~~
- ~~Проблемы iOS~~
- Проблемы Android
- Выводы

View lifecycle differences



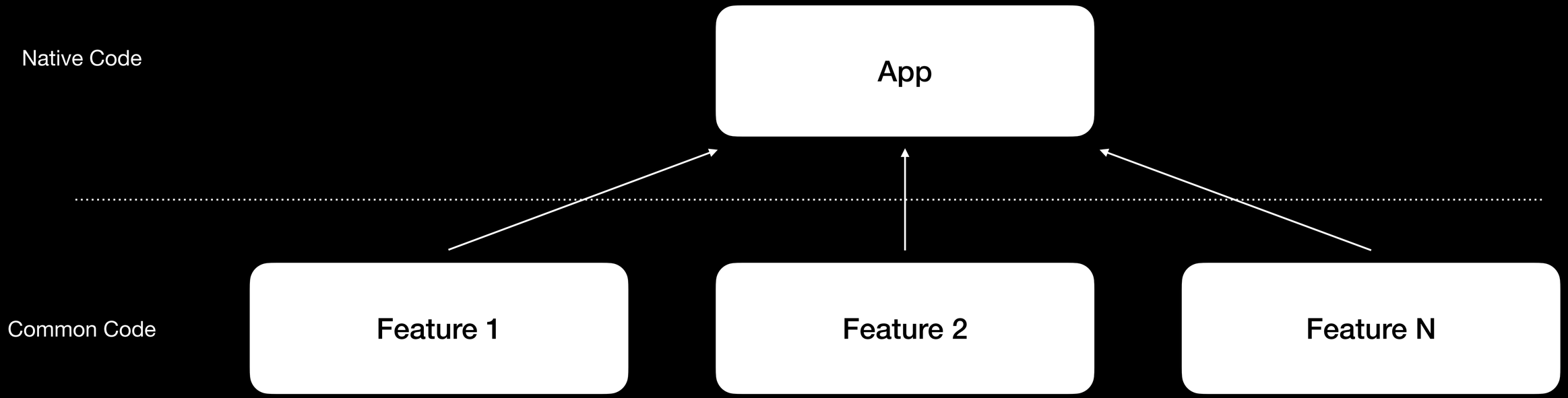

11:01
Search
Games
WARRIOR
LOGIC BOARD GAMES
YOUR TAGLINE HERE
Sport
RUGBY
BASEBALL
Art & Design
AIRPLANE
Reliability and accuracy
ART


11:02
LOGIC BOARD GAMES
YOUR TAGLINE HERE

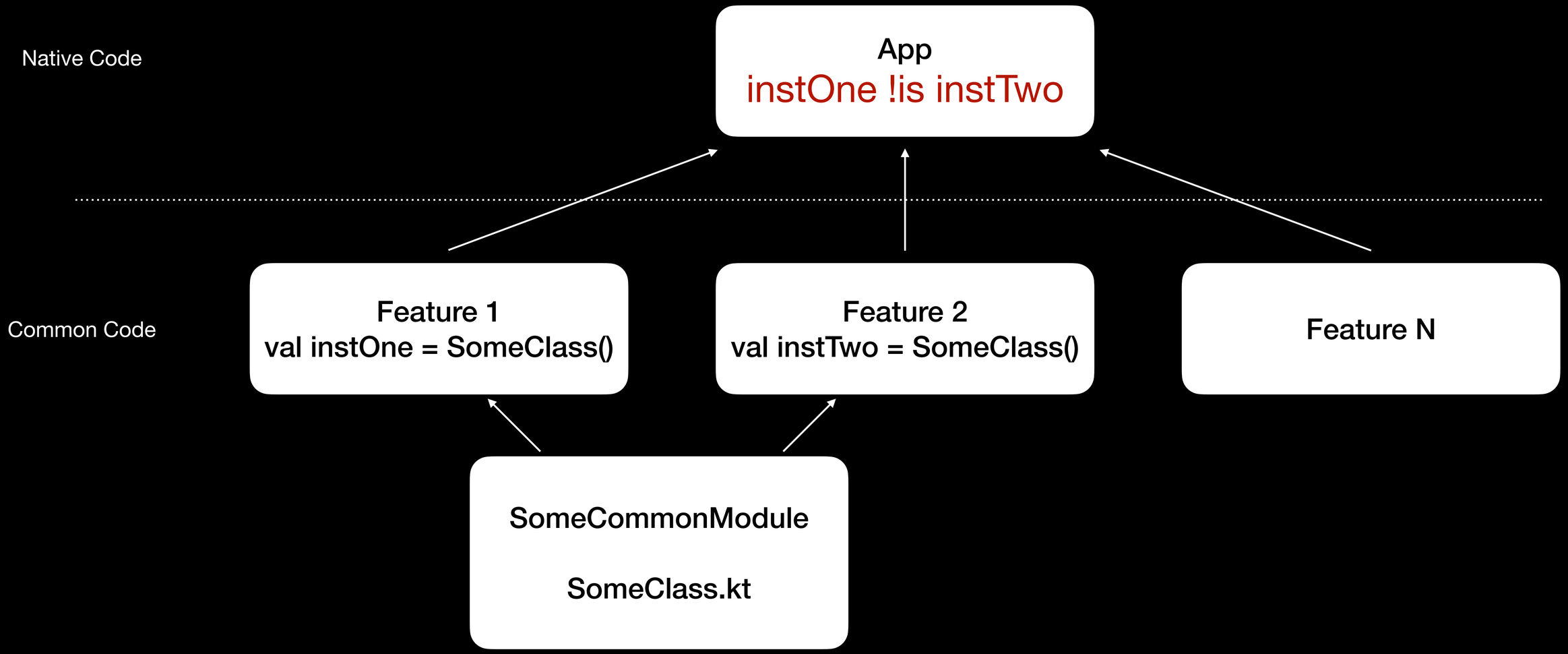

Points
Pixels

Ограничение по модульности

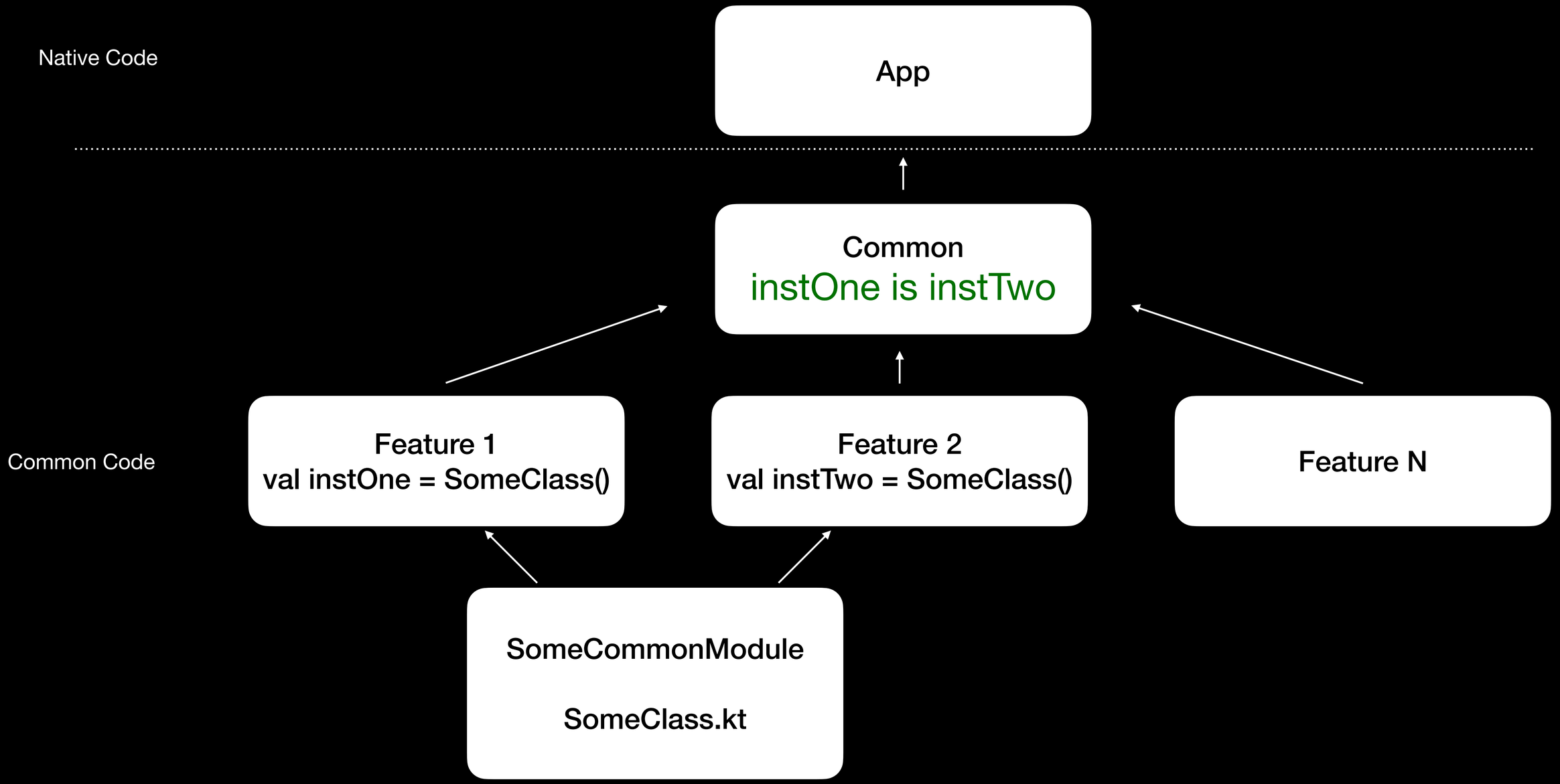
Модульность в идеальном мире



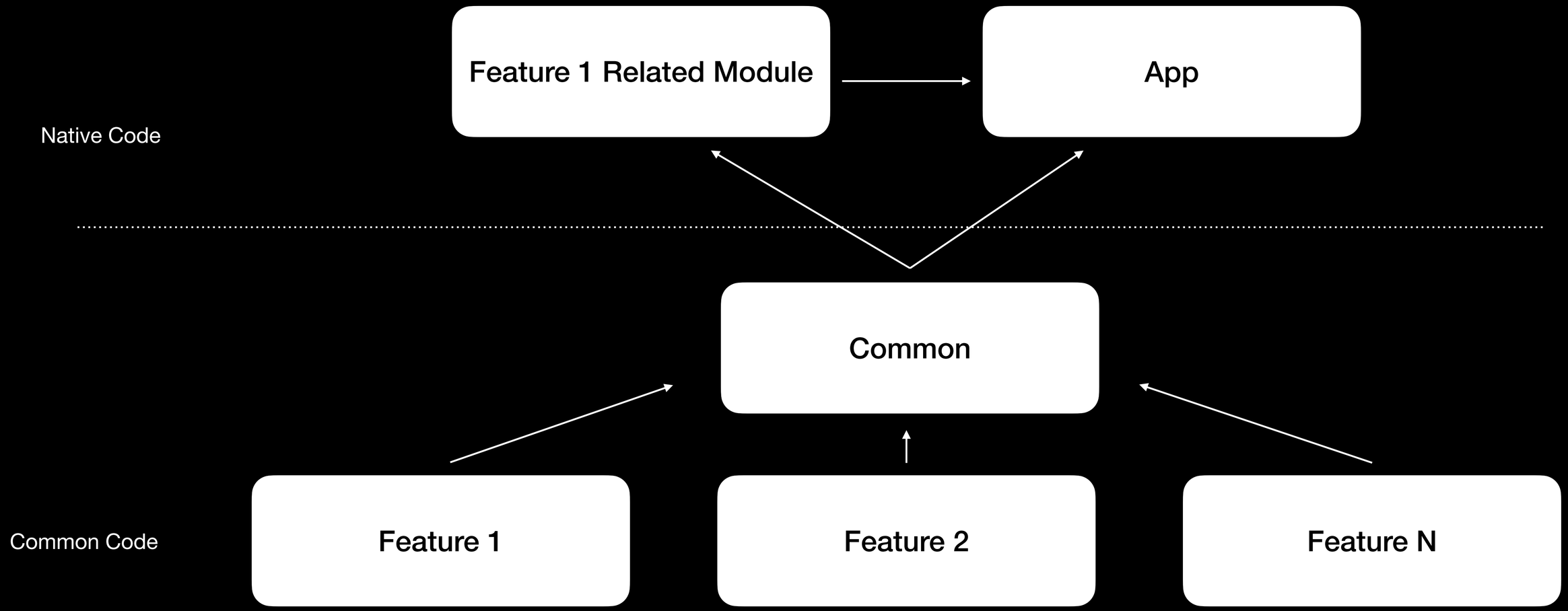
Модульность в KMM



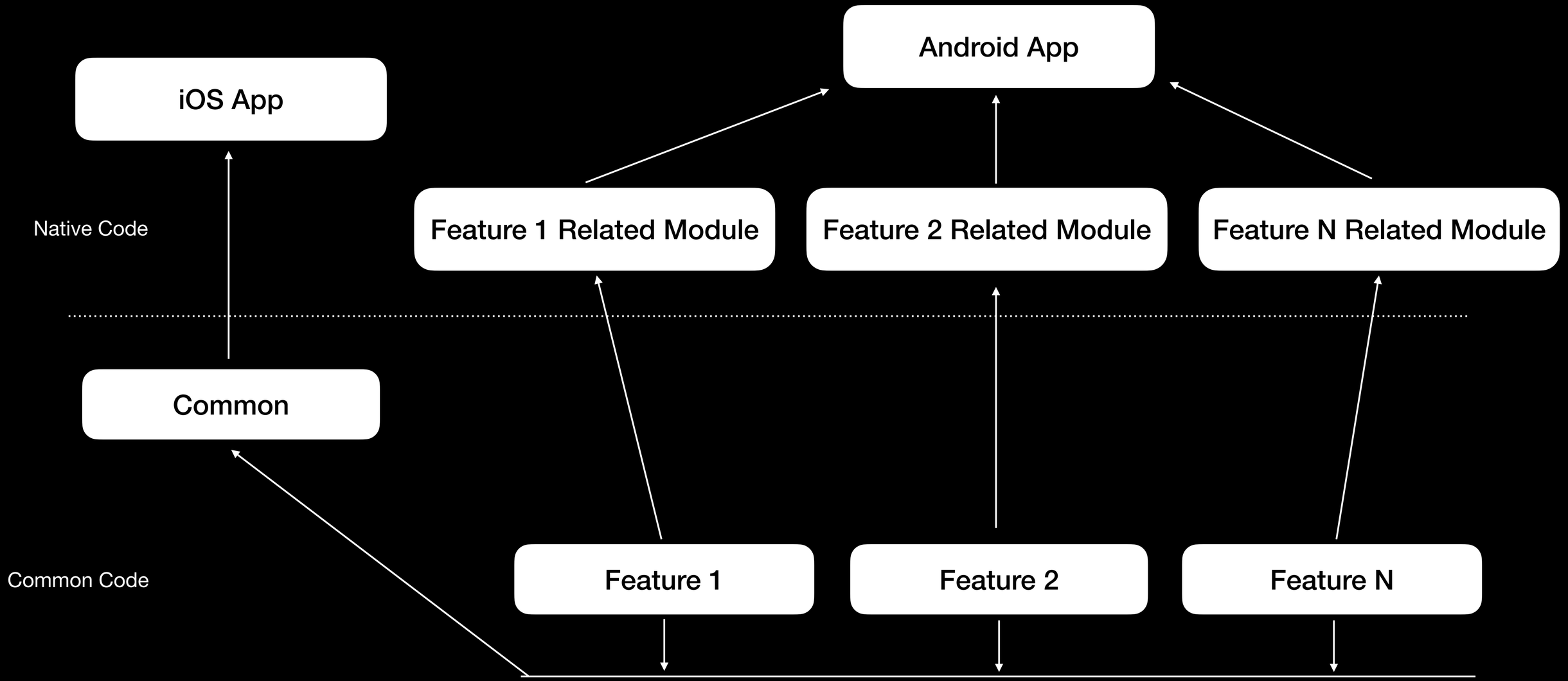
Модульность в KMM



Модульность в KMM



Как же быть?


Android App
iOS App
Native Code
Feature 1 Related Module
Feature 2 Related Module
Feature N Related Module
Common
Common Code
Feature 1
Feature 2
Feature N

Боль Android разработчика :)

О чем будем говорить

- ~~Спикеры~~
- ~~Почему KMM~~
- ~~Shaped~~
- ~~Архитектура редактора~~
- ~~Проблемы iOS~~
- ~~Проблемы Android~~
- Выводы

Выводы

- Сухую логику шарить очень удобно
- Синхронность платформ
- Сильно высокий порог входа для iOS. Очень сложно найти замену разработчику, высокие риски
- Достаточно долго собирается проект с зависимостям для iOS

Результаты

- Мы решили отойти от KMM на проекте Shaped и переписать эту часть нативно
- В будущем, когда KMM вырастет, мы планируем попробовать еще ;)

Вопросы

Nikolai Dmitriev
dnv190@gmail.com
Telegram/Phone: +375 29 544-58-22

George Emelyanov
georgiscoolman@gmail.com
Telegram: +375 29 897-47-97