АВРОРА
СВОЯ СИСТЕМА

ОТКРЫТАЯ
МОБИЛЬНАЯ
ПЛАТФОРМА

# CMP НА ОС АВРОРА

## С нуля до альфы


**Глазков Денис**

Старший инженер-разработчик отдела разработки ОС

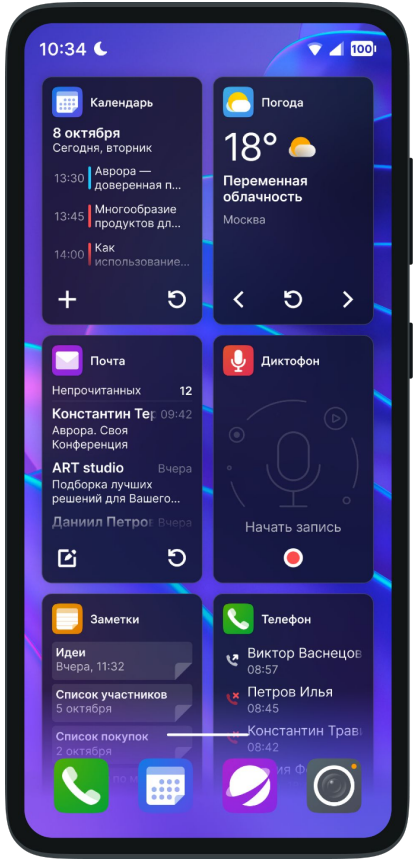
# План доклада

1. Мобильная ОС Аврора
2. В предыдущих сериях …
3. Архитектура CMP с точки зрения портирования
4. Разработка на CMP под ОС Аврора
5. Будущие планы

# Мобильная операционная система «Аврора»

Отечественная мобильная операционная система, основанная на ядре Linux, с фокусом на **безопасность и эргономику**

ОТКРЫТАЯ МОБИЛЬНАЯ ПЛАТФОРМА

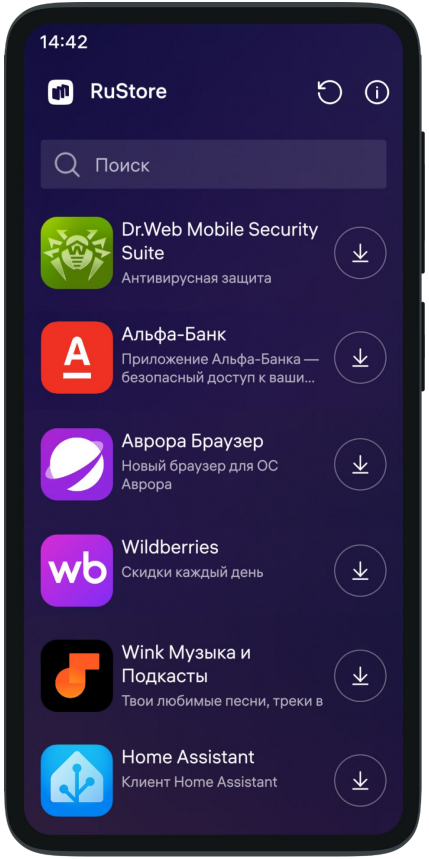


# Экосистема приложений

С 2023 года ведется активная работа в развитие поддержки кроссплатформенных фреймворков на ОС Аврора.

Поддерживаемые фреймворки

В процессе адаптации



# В предыдущих сериях …

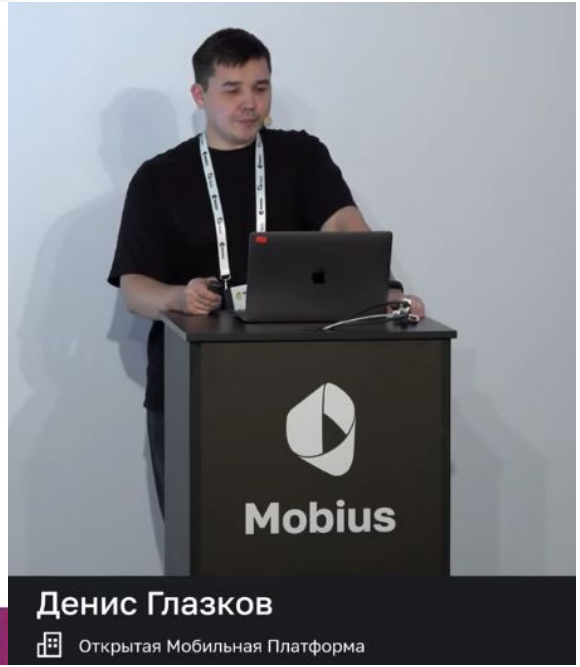
# Mobius 2025

## Текущий статус адаптации CMP

Демо приложение

**На данный момент реализовано:**

- Создание графического окна
- Сборку приложения под ОС Аврора
- Сборку итогового RPM пакета
- Запуск приложения в эмуляторе

# Mobius 2025

## Текущий статус адаптации CMP

Демо приложение

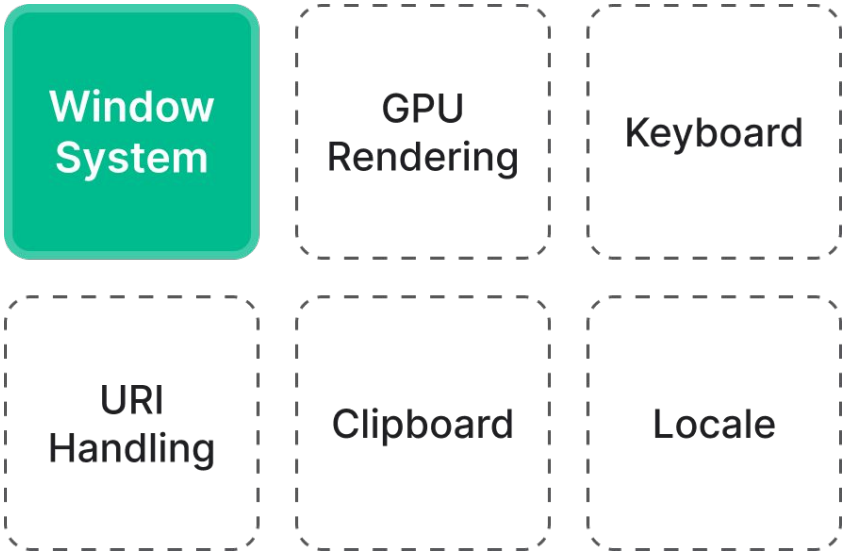
**На данный момент реализовано:**

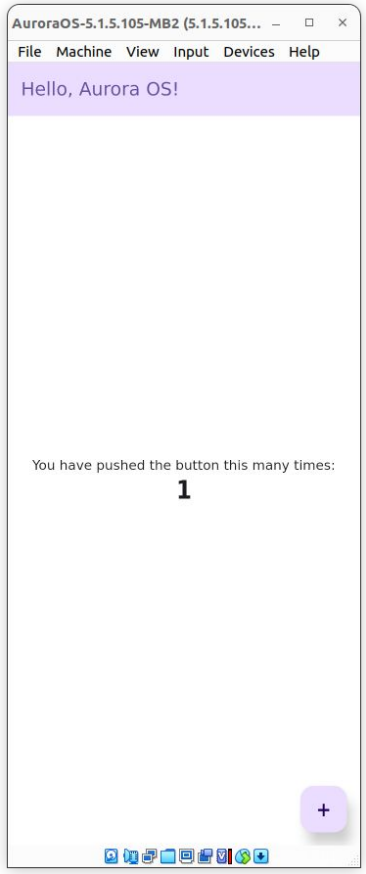
- Создание графического окна
- Сборку приложения под ОС Аврора
- Сборку итогового RPM пакета
- Запуск приложения в эмуляторе




АВРОРА | СВОЯ СИСТЕМА

# Статус CMP на конец 2025 года



# Статус CMP в середине 2026 года

Window System

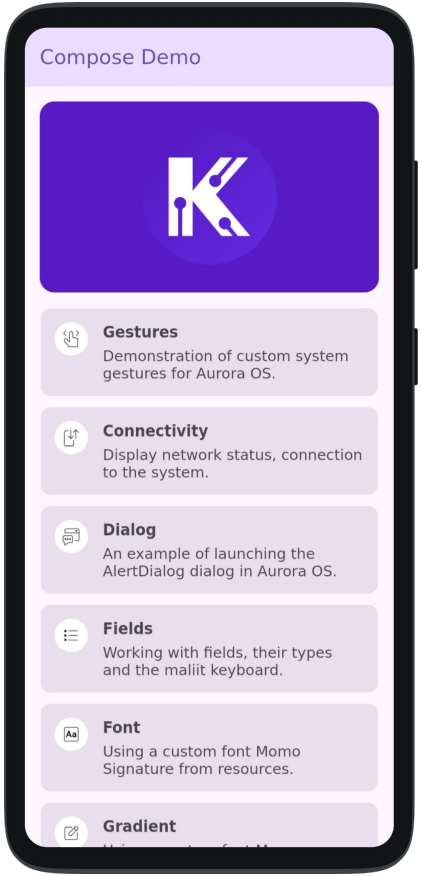
GPU Rendering

Keyboard

URI Handling

Clipboard

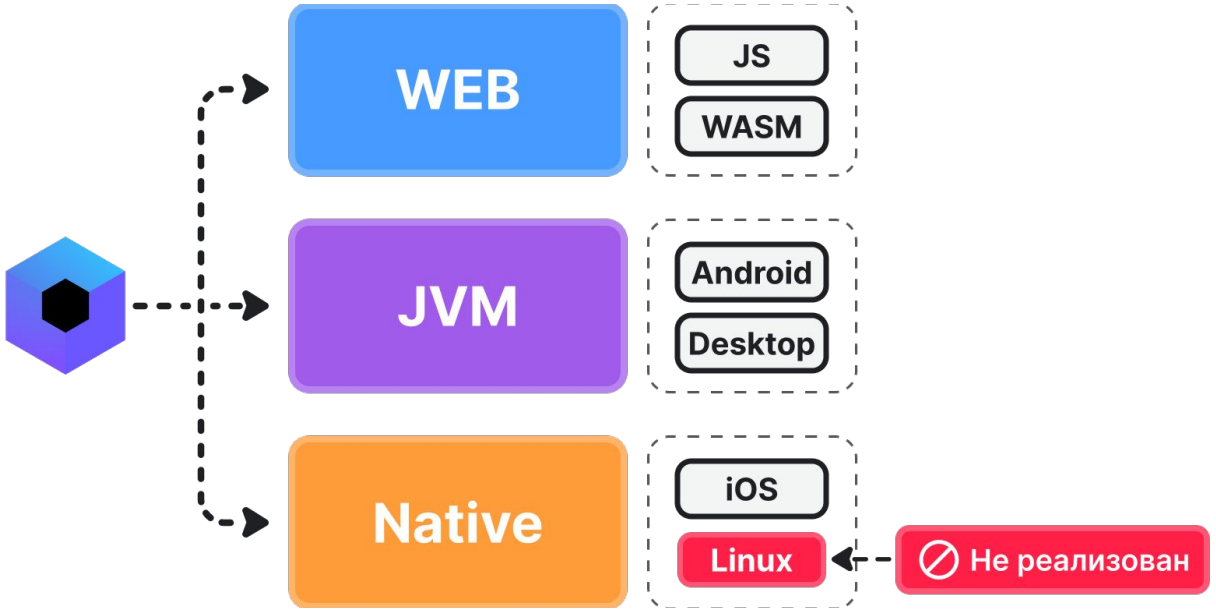
Locale



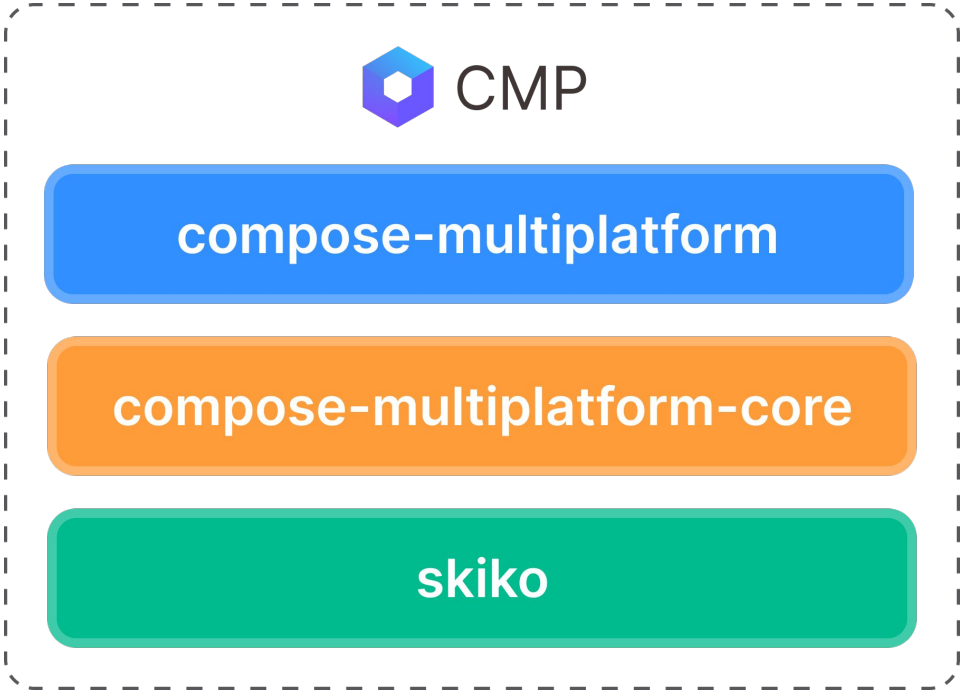

Источник:https://imgur.com/gallery/this-is-fine-Jqtp4BI

# Архитектура CMP с точки зрения портирования

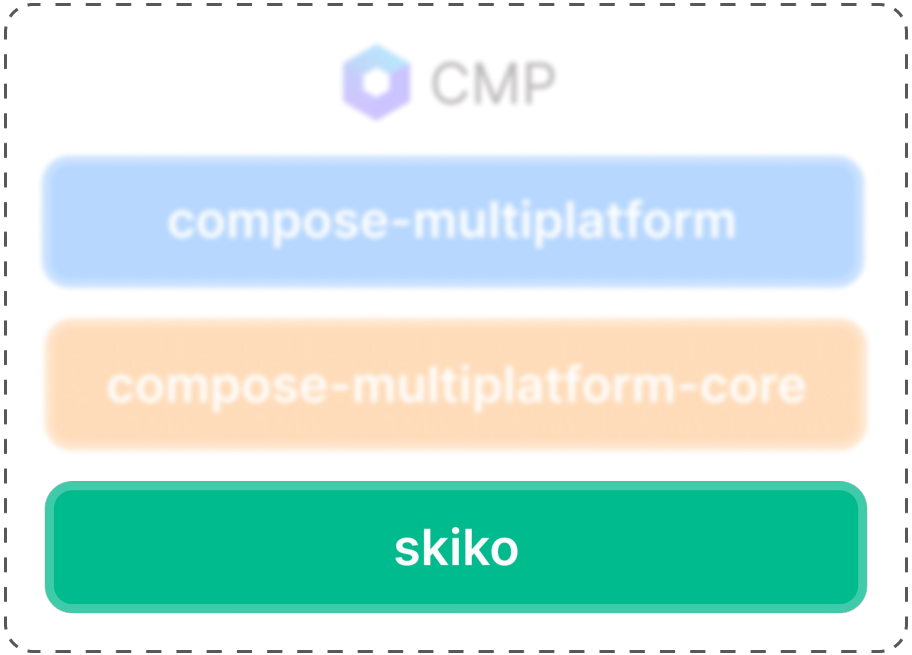
# CMP не поддерживает нативный Linux таргет из коробки



# Три столпа CMP

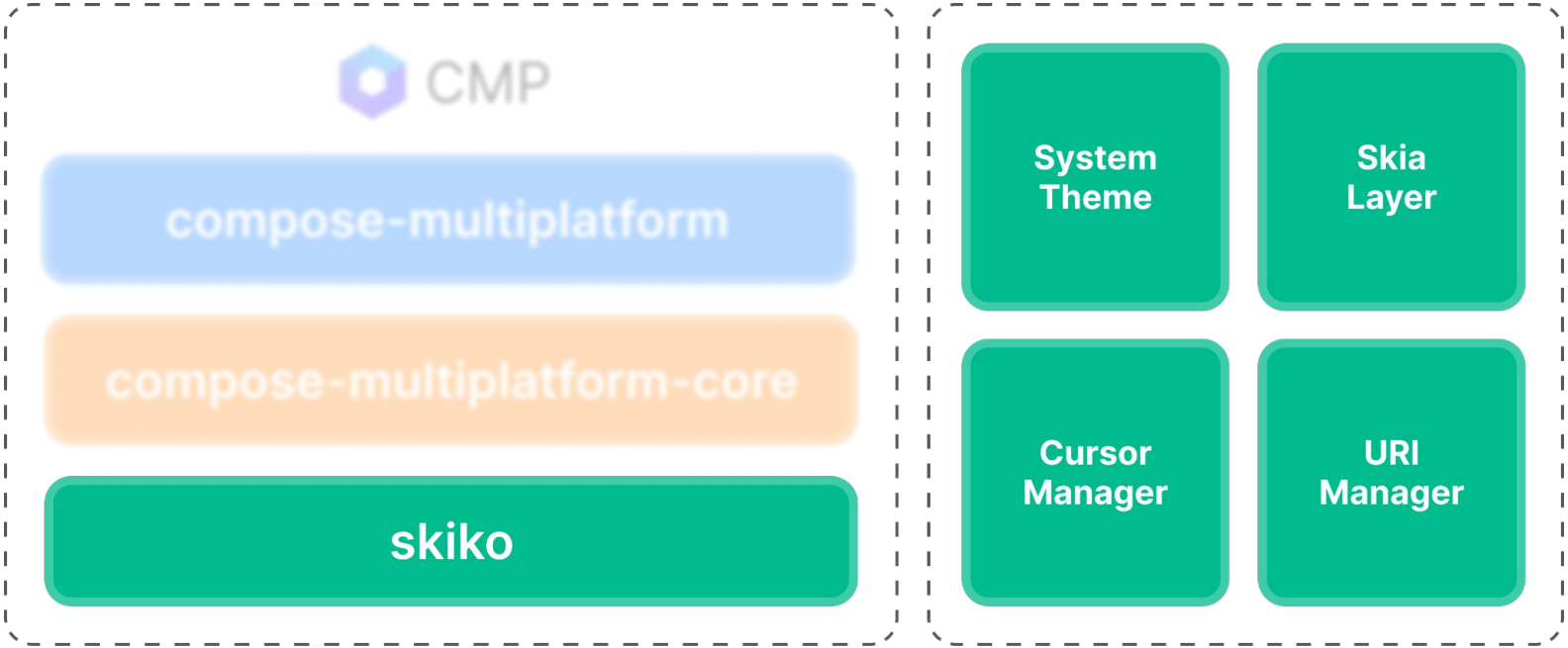


# Библиотека skiko

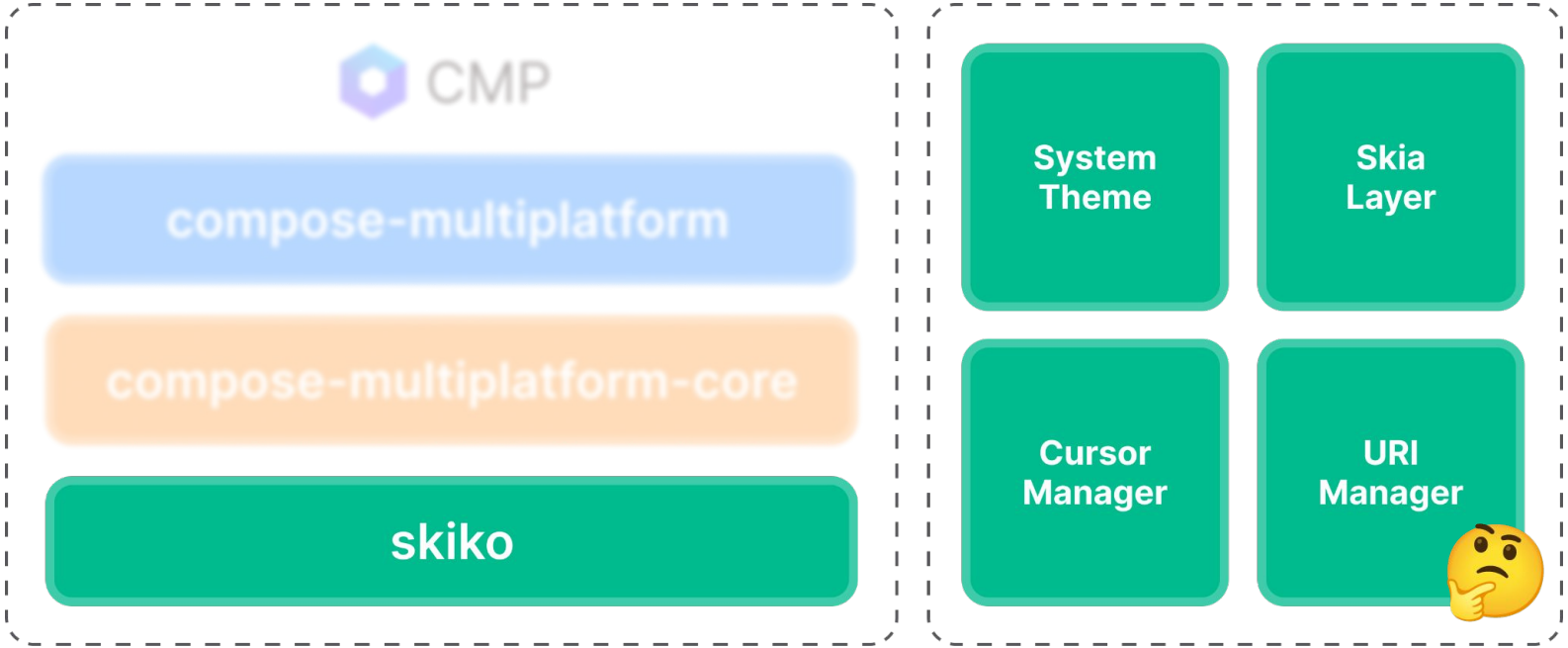



Библиотека, предоставляющая **Kotlin биндинги для** API библиотеки 2D-графики **Skia**

# Библиотека skiko



# Библиотека skiko

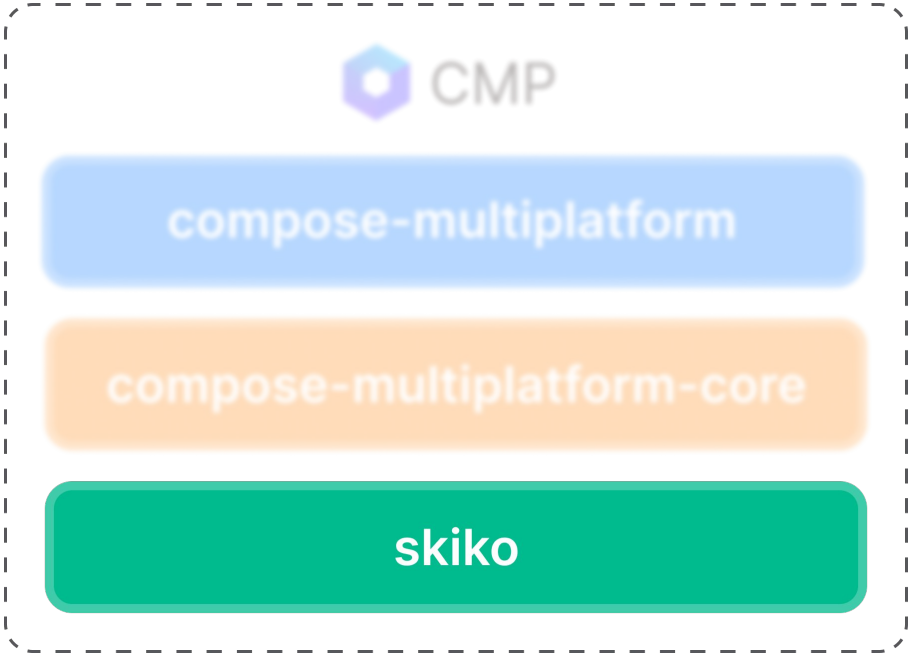


# Библиотека skiko

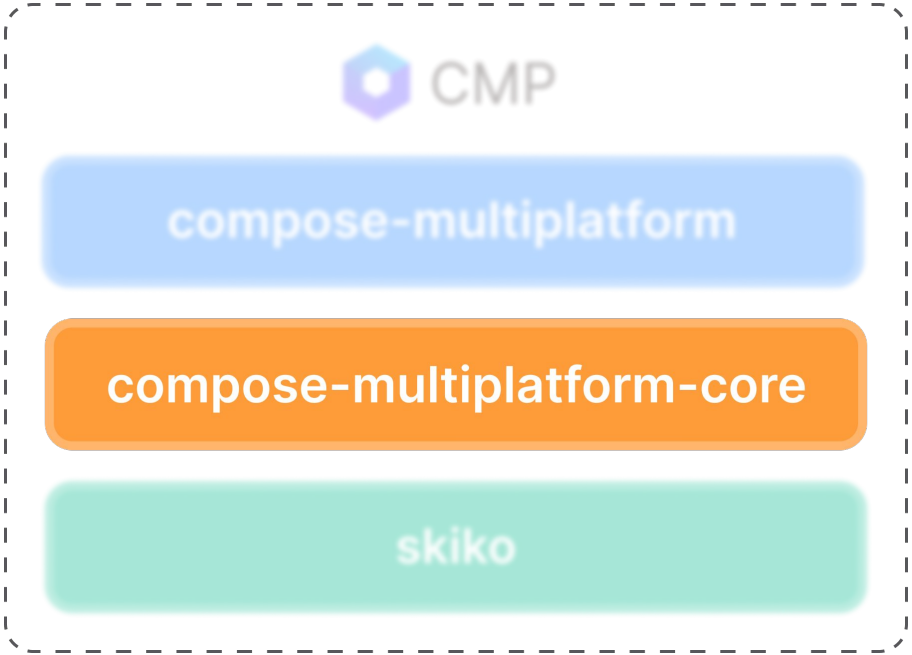
Интерфейс для реализации рендеринга на платформе

```
expect open class SkiaLayer {
    var renderApi: GraphicsApi
    val contentScale: Float
    val pixelGeometry: PixelGeometry
    var fullscreen: Boolean
    var transparency: Boolean
    val component: Any?
    var renderDelegate: SkikoRenderDelegate?
    fun attachTo(container: Any)
    fun detach()
    fun needRedraw()
    internal fun draw(canvas: Canvas)
}
```

# Библиотека skiko

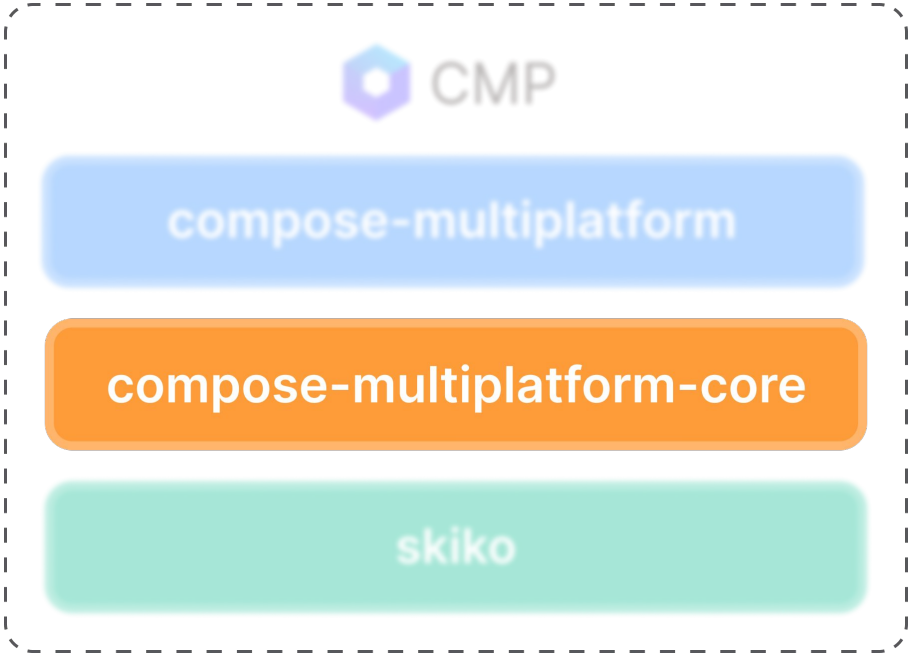


# Набор библиотек compose-multiplatform-core



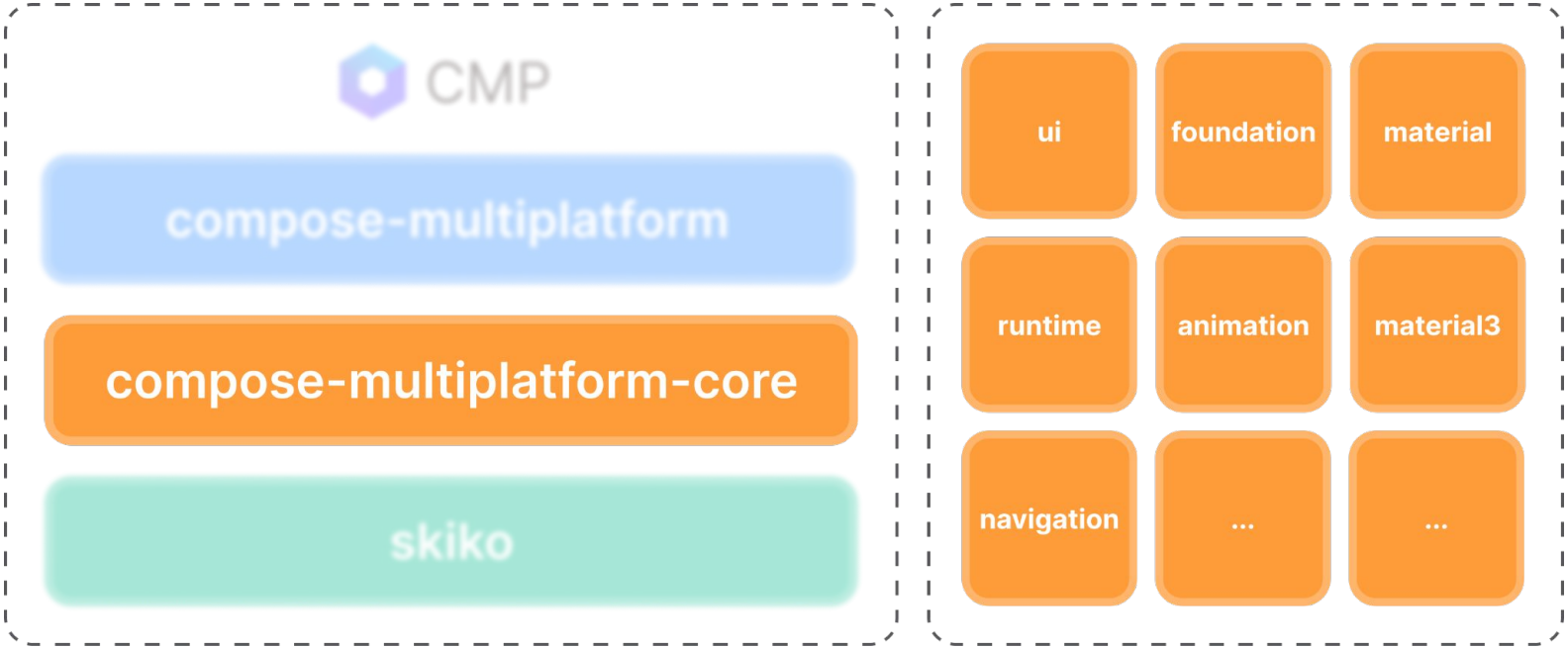

Набор KMP библиотек реализующий **основной функционал фреймворка CMP** (ui, runtime, material, ...)

# Набор библиотек compose-multiplatform-core




Огромная монорепа, которая индексируется пол часа, из-за которой мне пришлось купить 64 гб оперативки, и в которой ничего не понятно, особенно как добавить новый таргет 😁

# Набор библиотек compose-multiplatform-core



# Набор библиотек compose-multiplatform-core

## Создание окна и настройка рендеринга для платформы

```
fun Window(content: @Composable () → Unit) {
    private val layer = SkiaLayer()
    private val scene = CanvasLayersComposeScene(invalidate = layer::needRedraw)

    init {
        layer.renderDelegate = object : SkikoRenderDelegate {
            override fun onRender(canvas: Canvas, w: Int, h: Int, time: Long) {
                scene.render(canvas.asComposeCanvas(), time)
            }
        }
        scene.setContent { content() }
    }
}
```

# Набор библиотек compose-multiplatform-core

## Интеграция с виртуальной клавиатурой платформы

```
internal class AuroraTextInputService : PlatformTextInputService {
    override fun showSoftwareKeyboard(ime: ImeOptions) {
        when(ime.keyboardType) {
            KeyboardType.Text   → AuroraKeyboard.openType(AuroraKeyboardType.Text)
            KeyboardType.Number → AuroraKeyboard.openType(AuroraKeyboardType.Number)
            else                → AuroraKeyboard.open()
        }
    }

    override fun hideSoftwareKeyboard() {
        AuroraKeyboard.close()
    }
}
```

# Набор библиотек compose-multiplatform-core

Конфигурация эффекта скролла для платформы

```
class AuroraOverscrollEffect() : OverscrollEffect {
    override val isInProgress: Boolean
    override fun applyToScroll(...): Offset { ... }
    override suspend fun applyToFling(...)  { ... }
}
```

# Набор библиотек compose-multiplatform-core

Реализация отрытия URI для платформы

```
class AuroraUriHandler() : UriHandler {
    override fun openUri(uri: String) {
        if (!UriLauncher.open(uri)) {
            throw IllegalArgumentException("Can't open $uri.")
        }
    }
}
```

# Набор библиотек compose-multiplatform-core

И много интересных флагов, разбросанных по проекту …

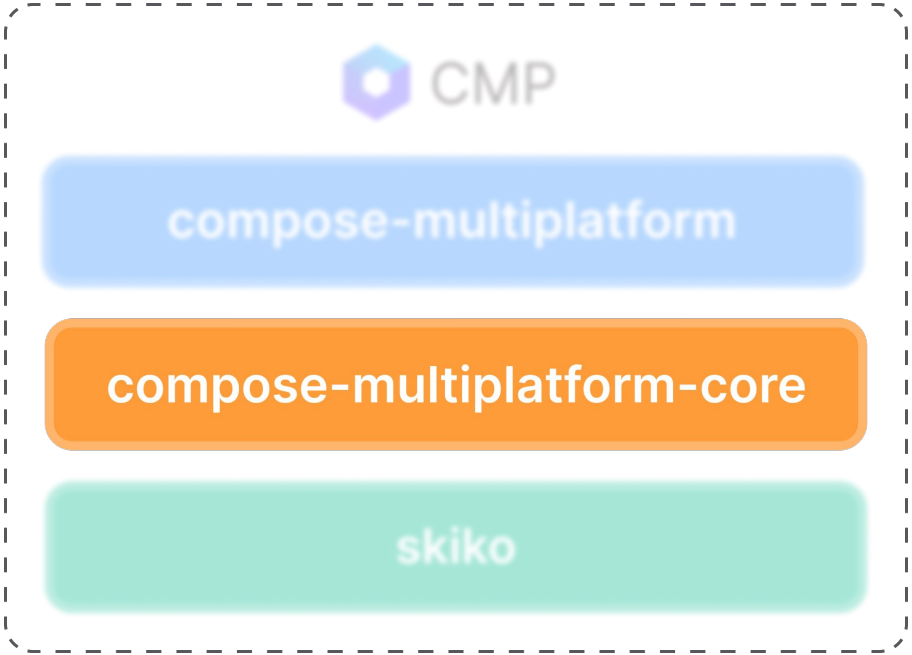
```
/**
 * This is a temporary workaround and should be removed
 * after proper mouse handling is settled (b/171402426).
 */
internal actual val isInTouchMode: Boolean = true
```

# Набор библиотек compose-multiplatform-core

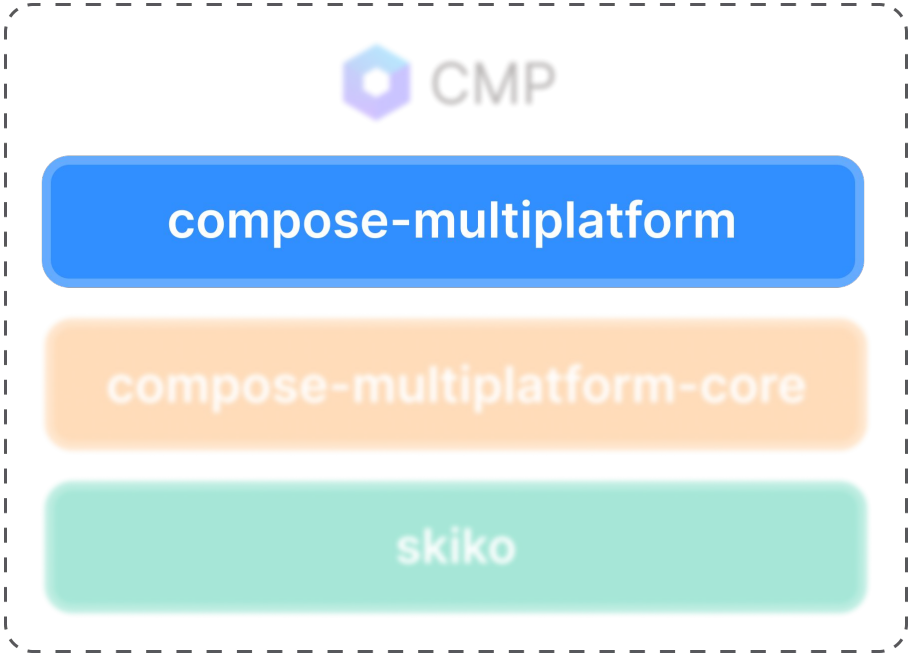
И много интересных флагов, разбросанных по проекту …

```
// TODO https://youtrack.jetbrains.com/issue/CMP-5814/...
internal actual fun isRequestFocusOnClickEnabled(): Boolean = false
```

# Набор библиотек compose-multiplatform-core

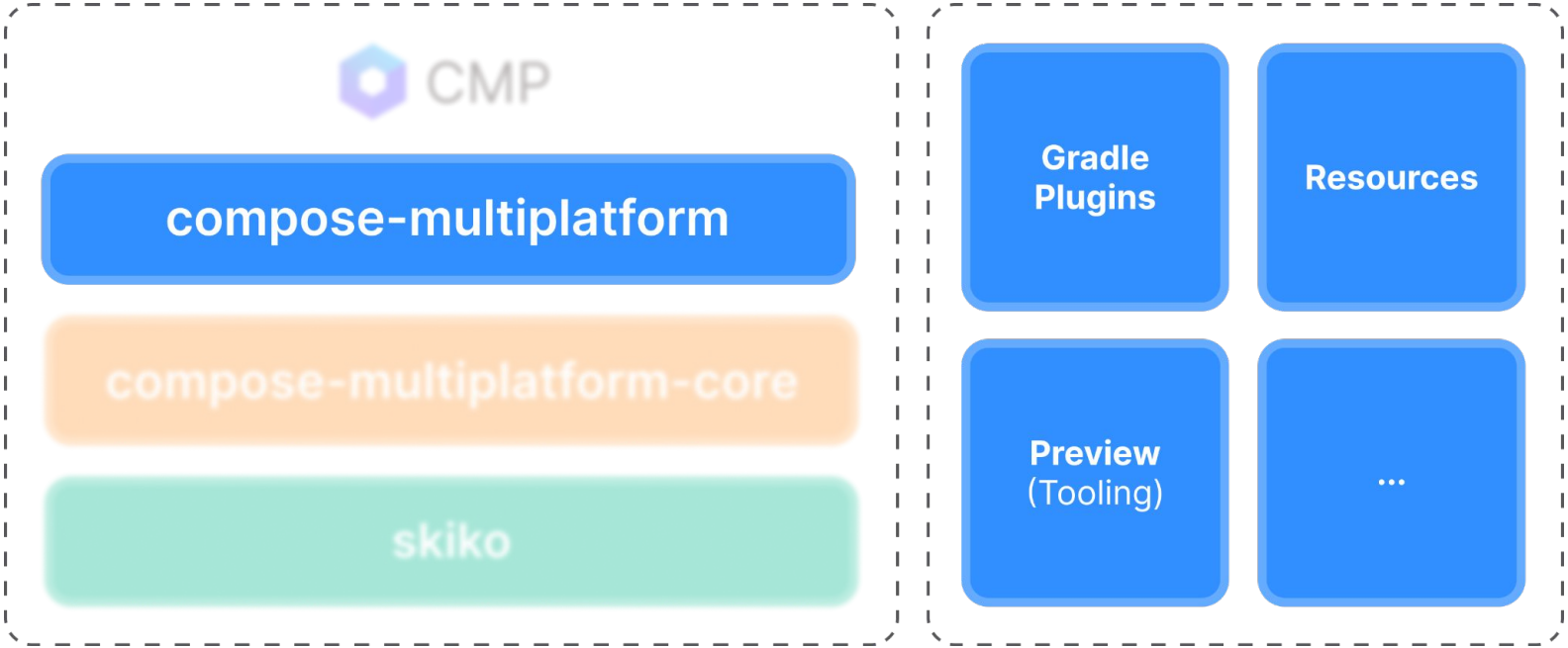


# Проект compose-multiplatform

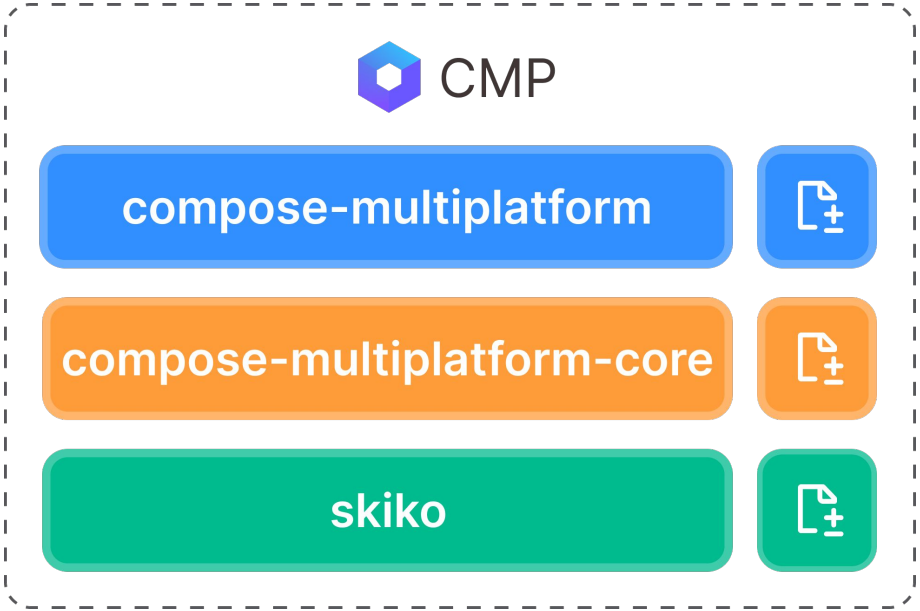



**Gradle плагины и тулинг** для удобства сборки проектов и разработки на CMP

# Проект compose-multiplatform



# Интеграция новой платформы



# Интеграция новой платформы

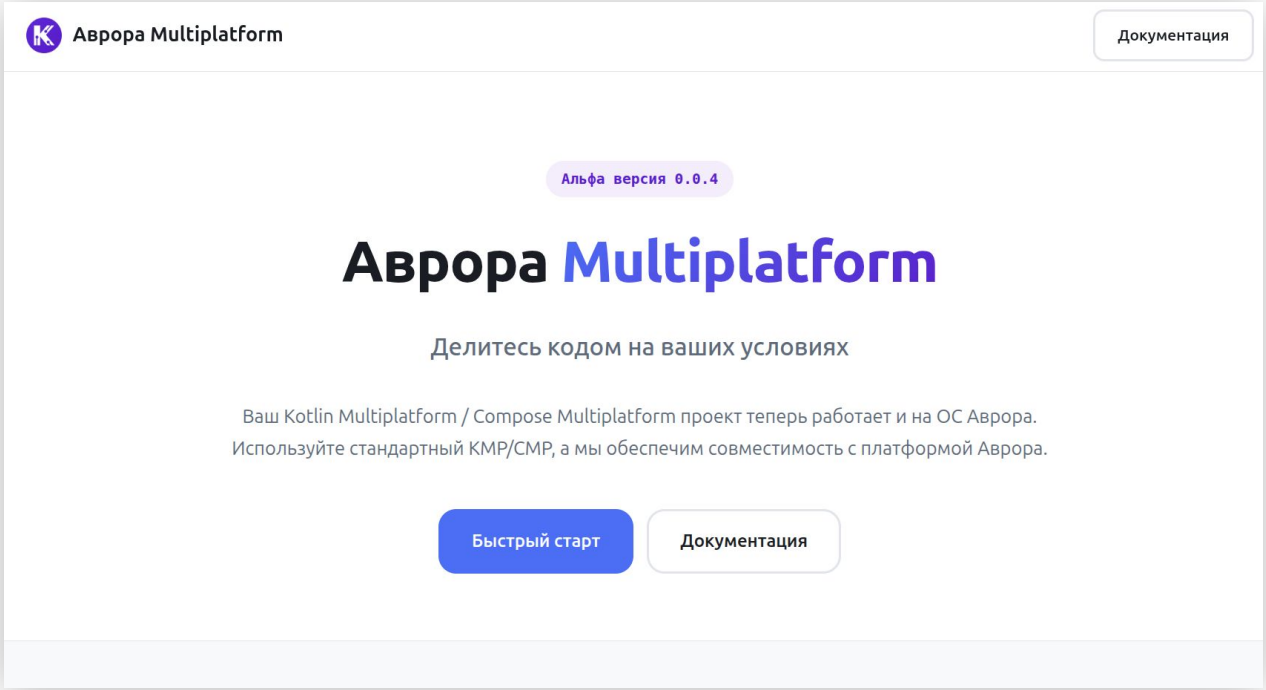
# Альфа версия СМР для ОС Аврора

# Альфа версия СМР для ОС Аврора

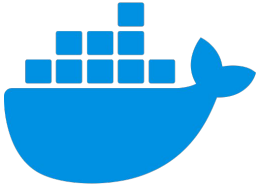
Бета программа

Aurora Developers

Аврора SDK




Документация



# Шаг 1

## Установка Docker Desktop

https://docs.docker.com/get-started/get-docker

# Install Docker Desktop on Mac

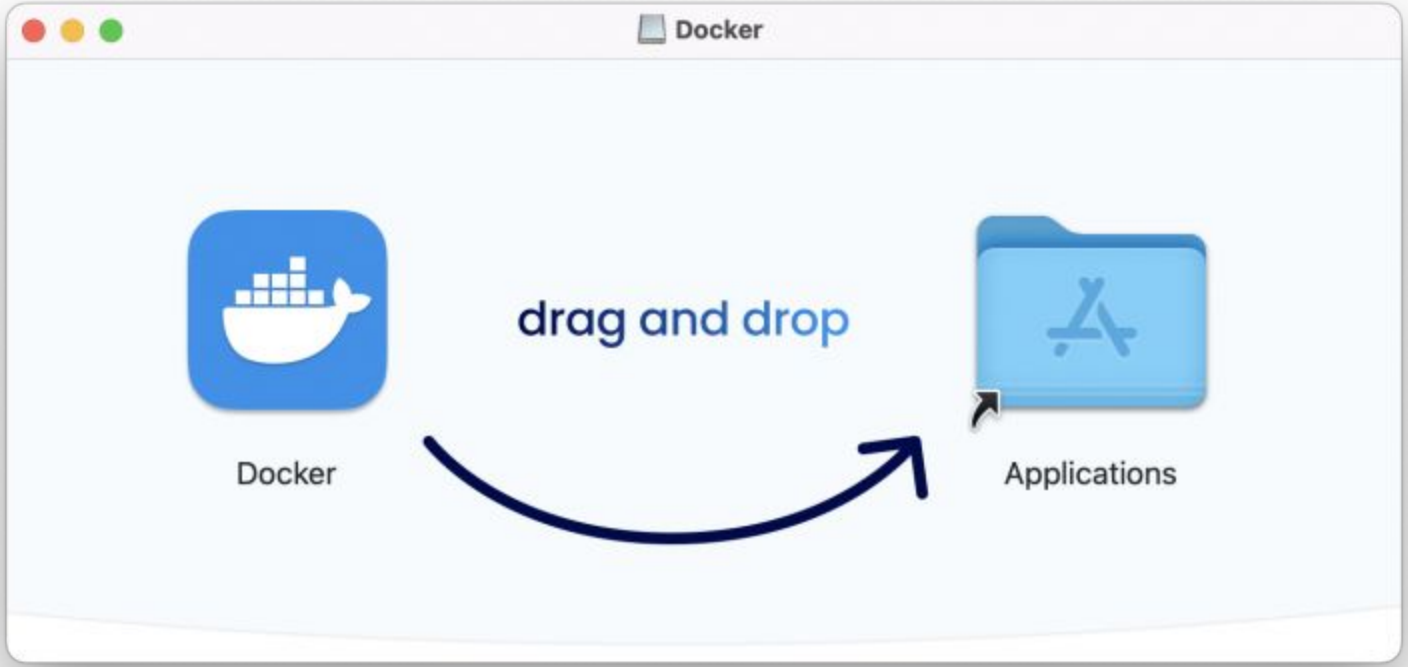
Copy as Markdown

**Docker Desktop terms**

Commercial use of Docker Desktop in larger enterprises (more than 250 employees or more than $10 million USD in annual revenue) requires a paid subscription.

This page provides download links, system requirements, and step-by-step installation instructions for Docker Desktop on Mac.

Docker Desktop for Mac with Apple silicon | Docker Desktop for Mac with Intel chip


Docker
drag and drop
Docker
Applications

# Шаг 2

## Установка Аврора SDK (ВТ)

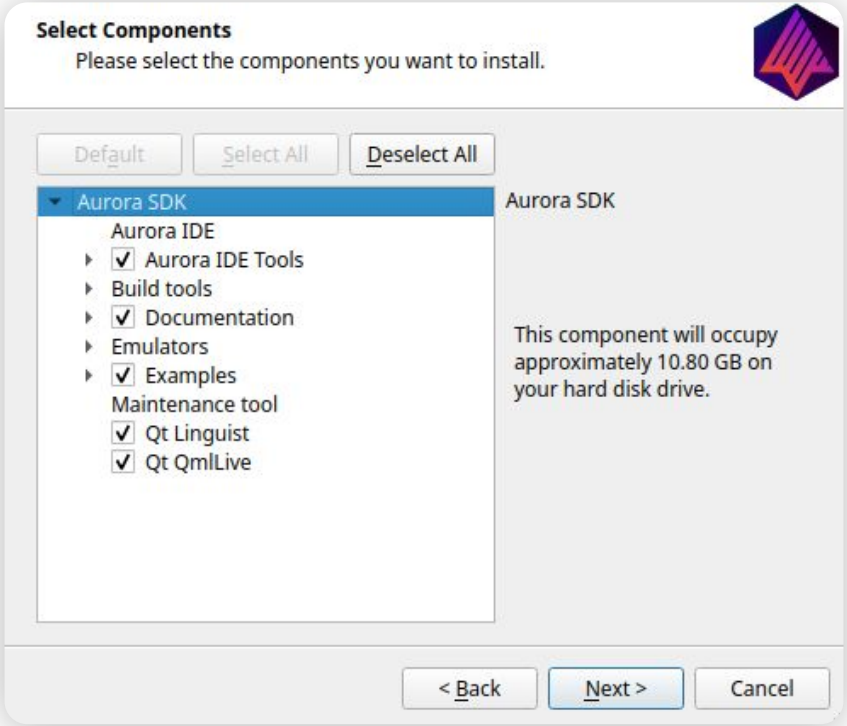
# Шаг 2

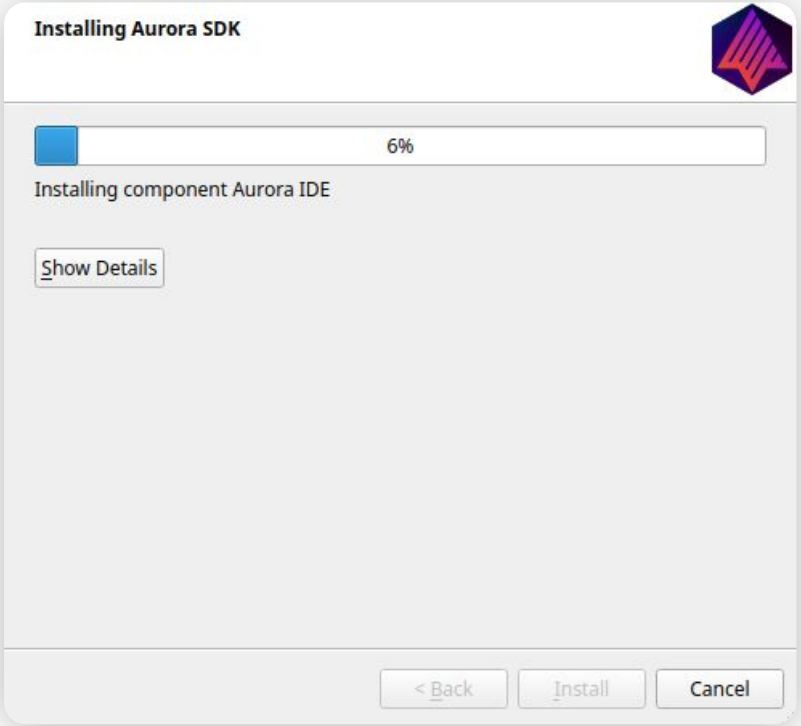
## Установка Аврора SDK (~~MB~~2)

# Шаг 2

## Установка Аврора SDK (ВТ)

https://developer.auroraos.ru/downloads/archive/sdk_bt


Select Components
Please select the components you want to install.
Default
Select All
Deselect All
Aurora SDK
Aurora IDE
Aurora IDE Tools
Build tools
Documentation
Emulators
Examples
Maintenance tool
Qt Linguist
Qt QmlLive
Aurora SDK
This component will occupy approximately 10.80 GB on your hard disk drive.
< Back
Next >
Cancel


Installing Aurora SDK
6%
Installing component Aurora IDE
Show Details
< Back
Install
Cancel

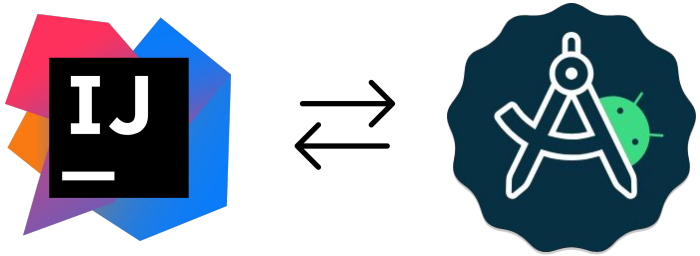
# Шаг 3

## Подготовка Maven Local

https://hub.mos.ru/auroraos/kotlin-multiplatform/aurora-maven

```
# Download
$ wget -O aurora-maven.tar.gz https://hub.mos.ru/auroraos/kotlin-multiplatform/aurora-maven/-/archive/aurora-0.0.4/aurora-maven-aurora-0.0.4.tar.gz

# Extract
$ tar -xzf aurora-maven.tar.gz
$ mkdir -p ~/.m2/
$ mv aurora-maven-* ~/.m2/repository
```

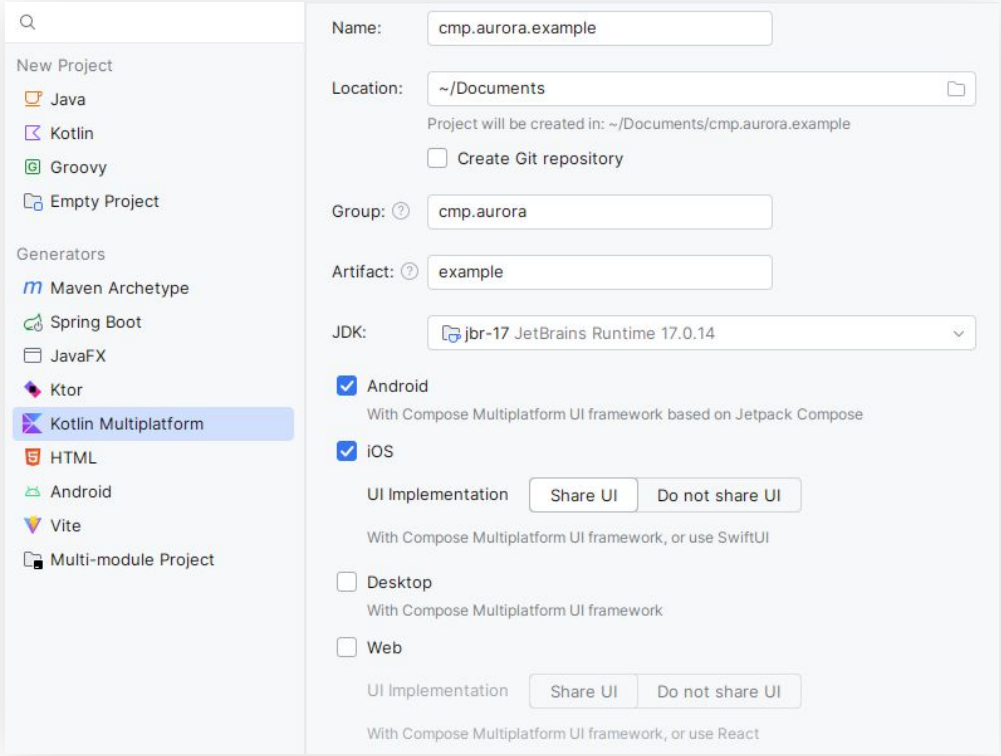



Для разработки можно использовать любимую IDE

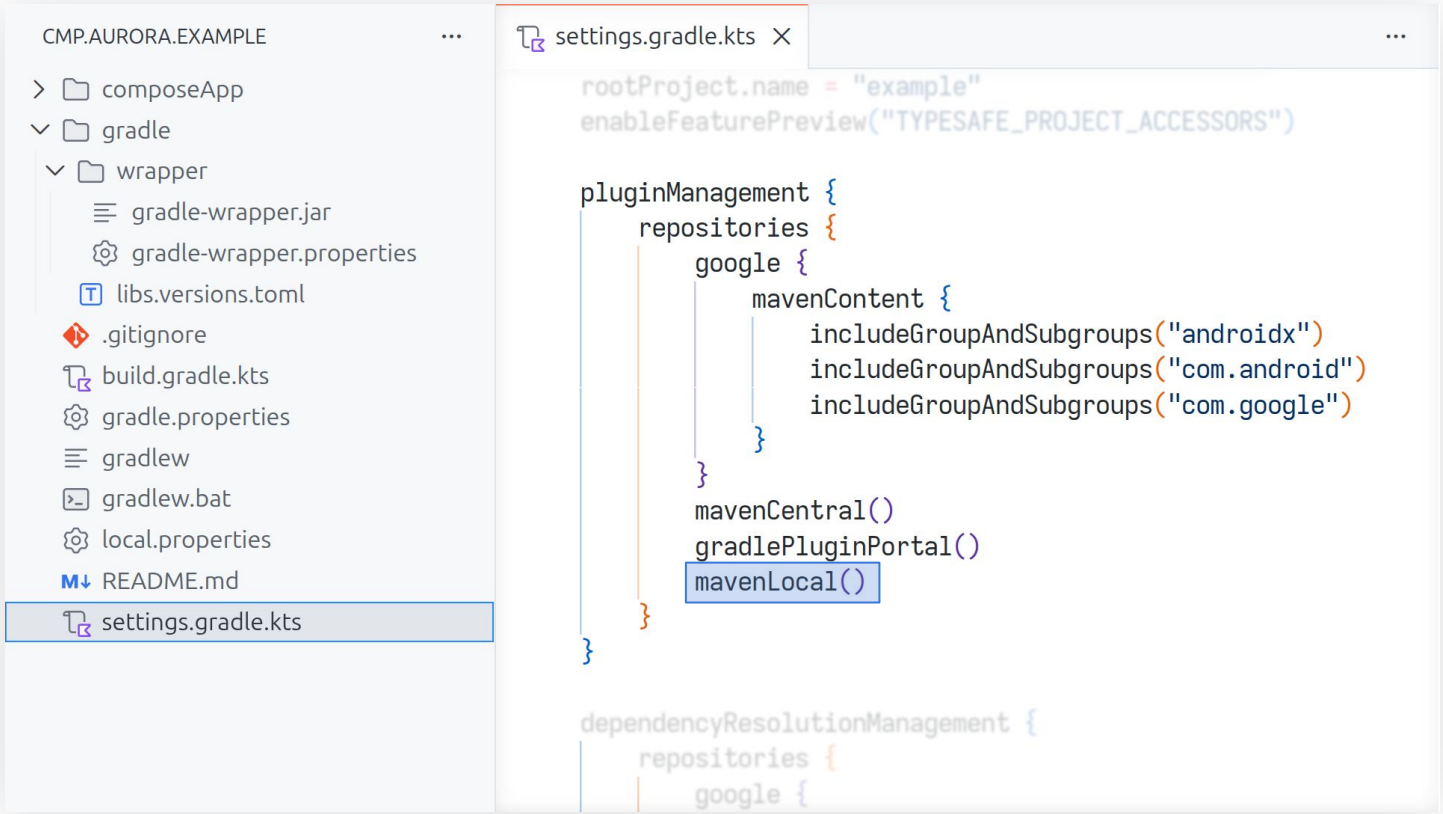
CMP для ОС Аврора подходит только для разработки под ОС Аврора

# Процесс разработки CMP приложения с единственным таргетом - ОС Аврора

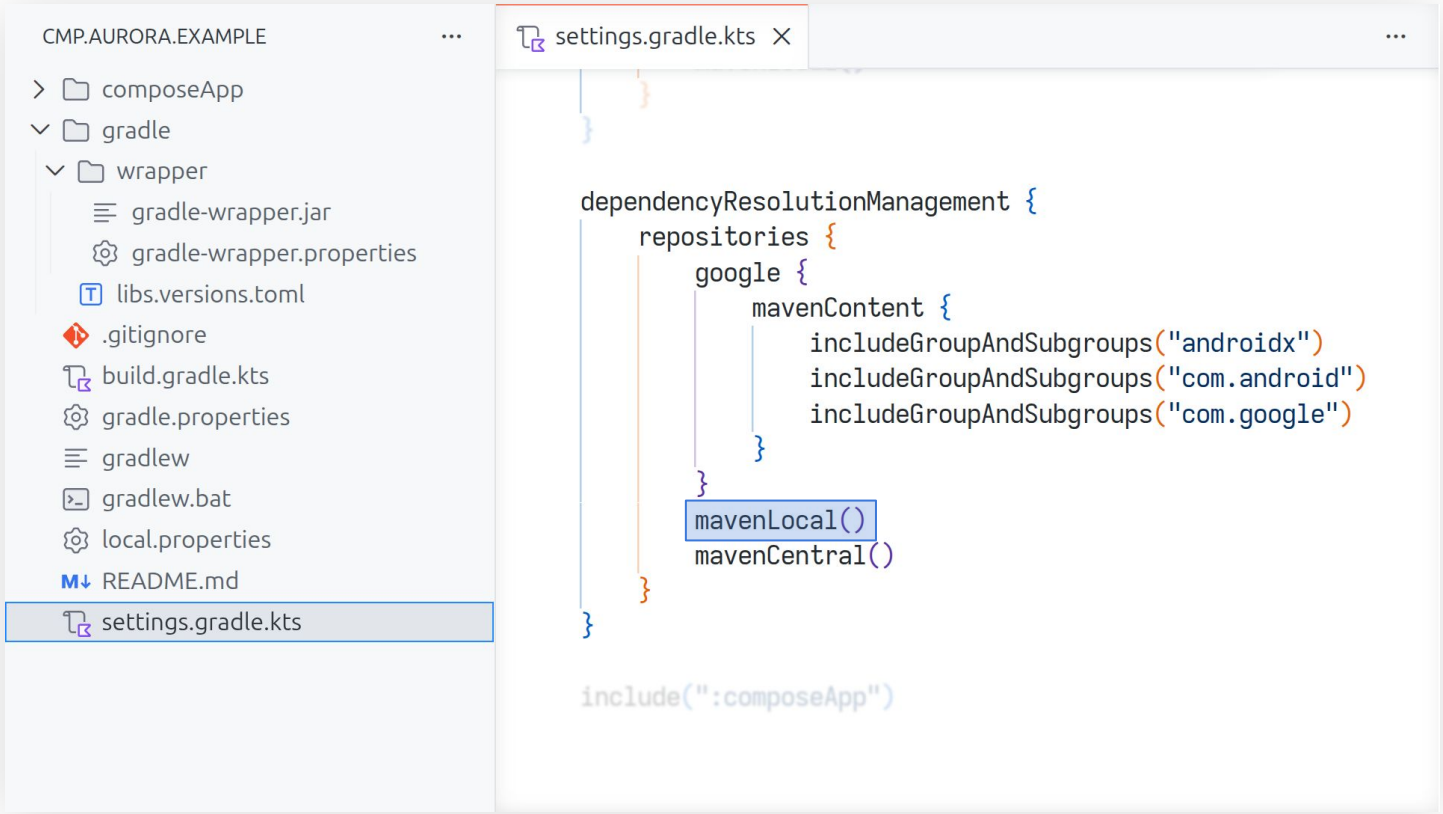
# **Шаг 4**. Создание нового проекта



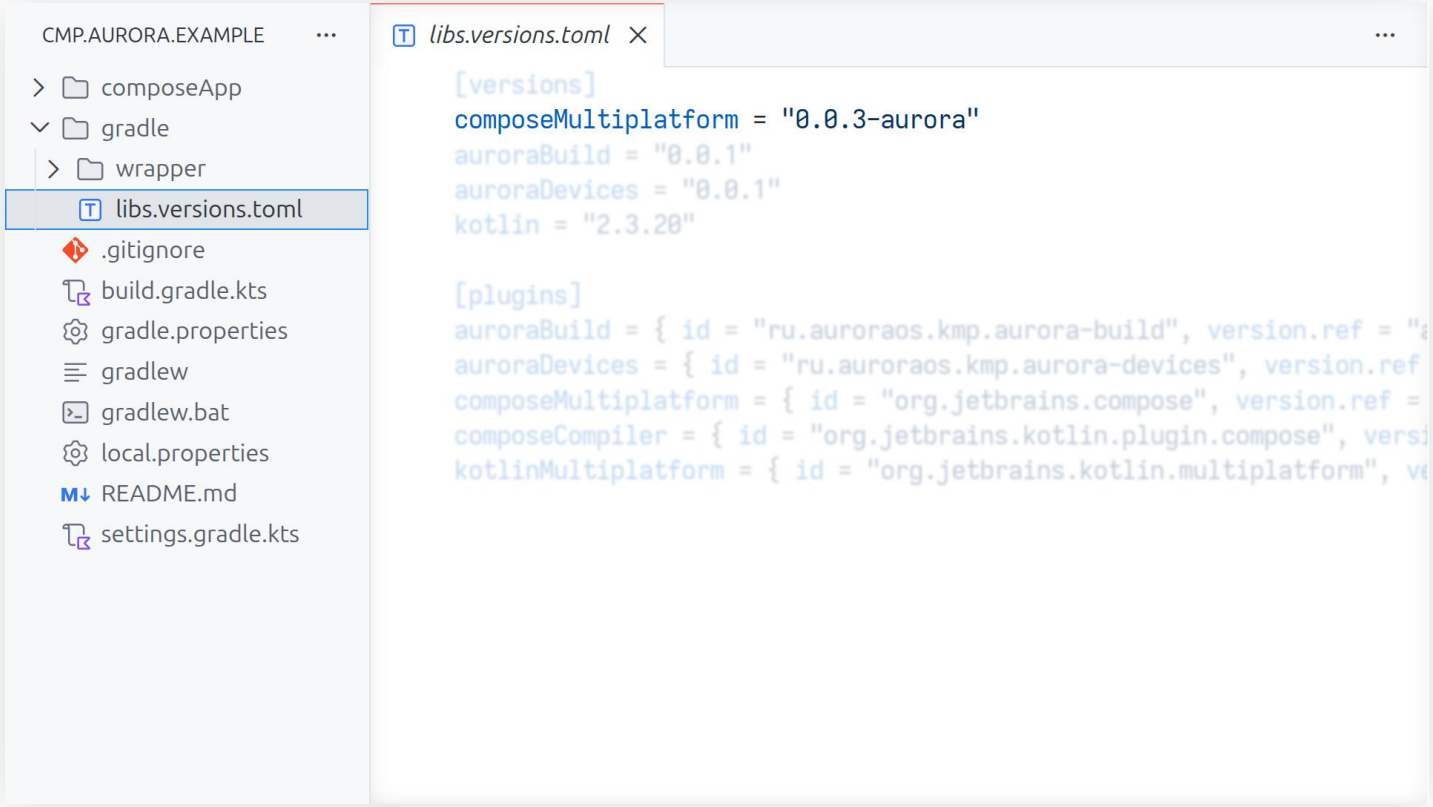
# **Шаг 5**. Подключение Maven Local



# **Шаг 5**. Подключение Maven Local

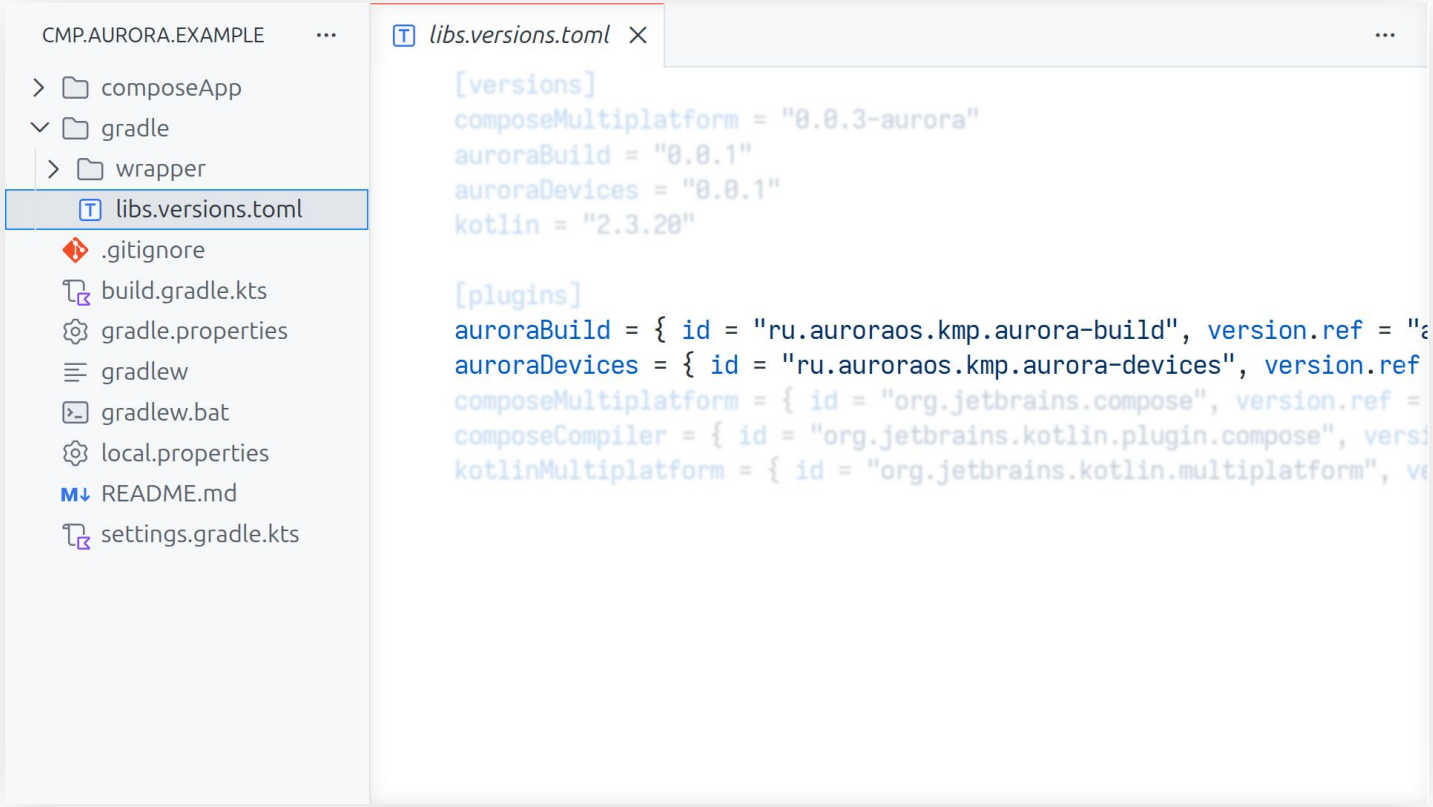


# **Шаг 6**. Изменение версии CMP

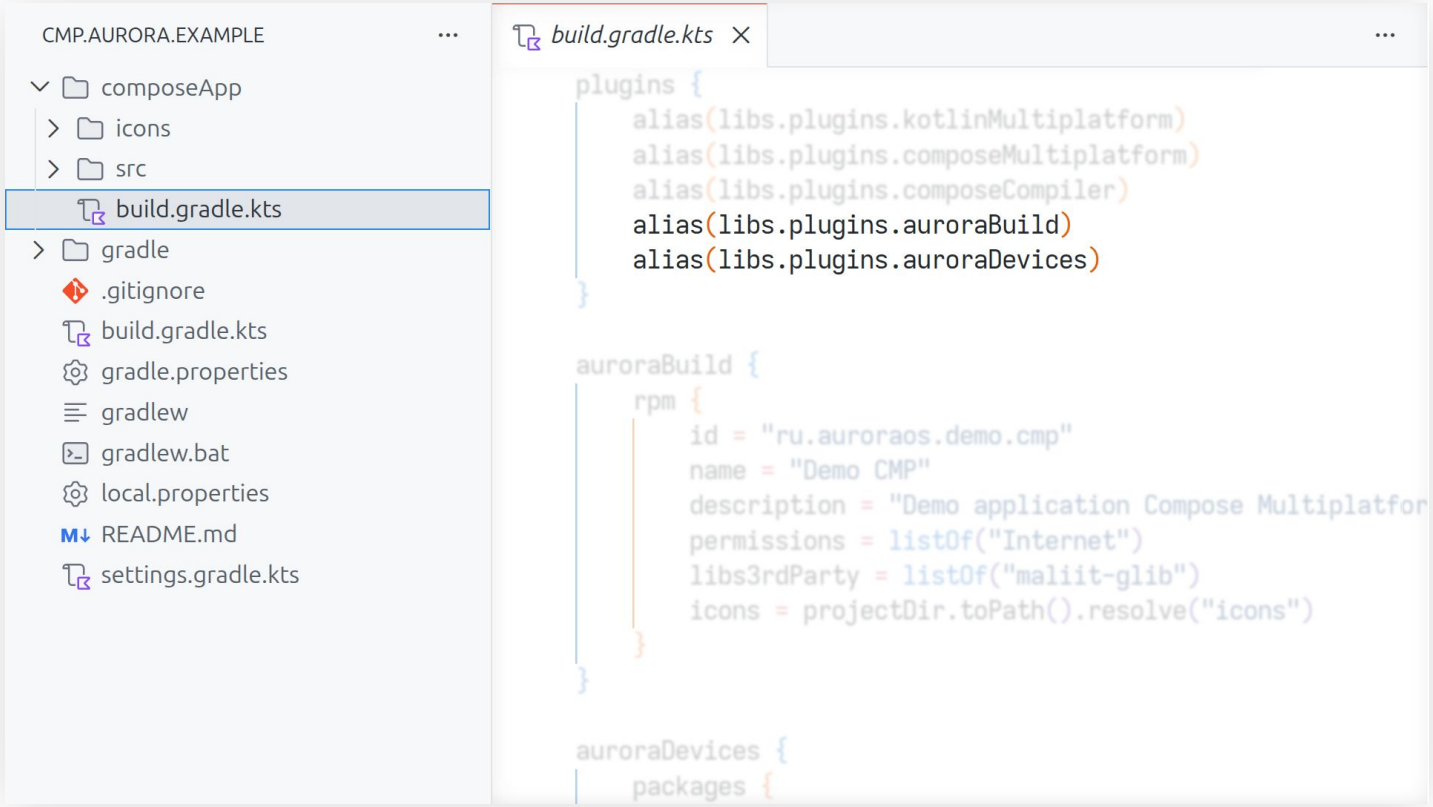


# **Шаг 7**. Добавление плагинов для ОС Аврора

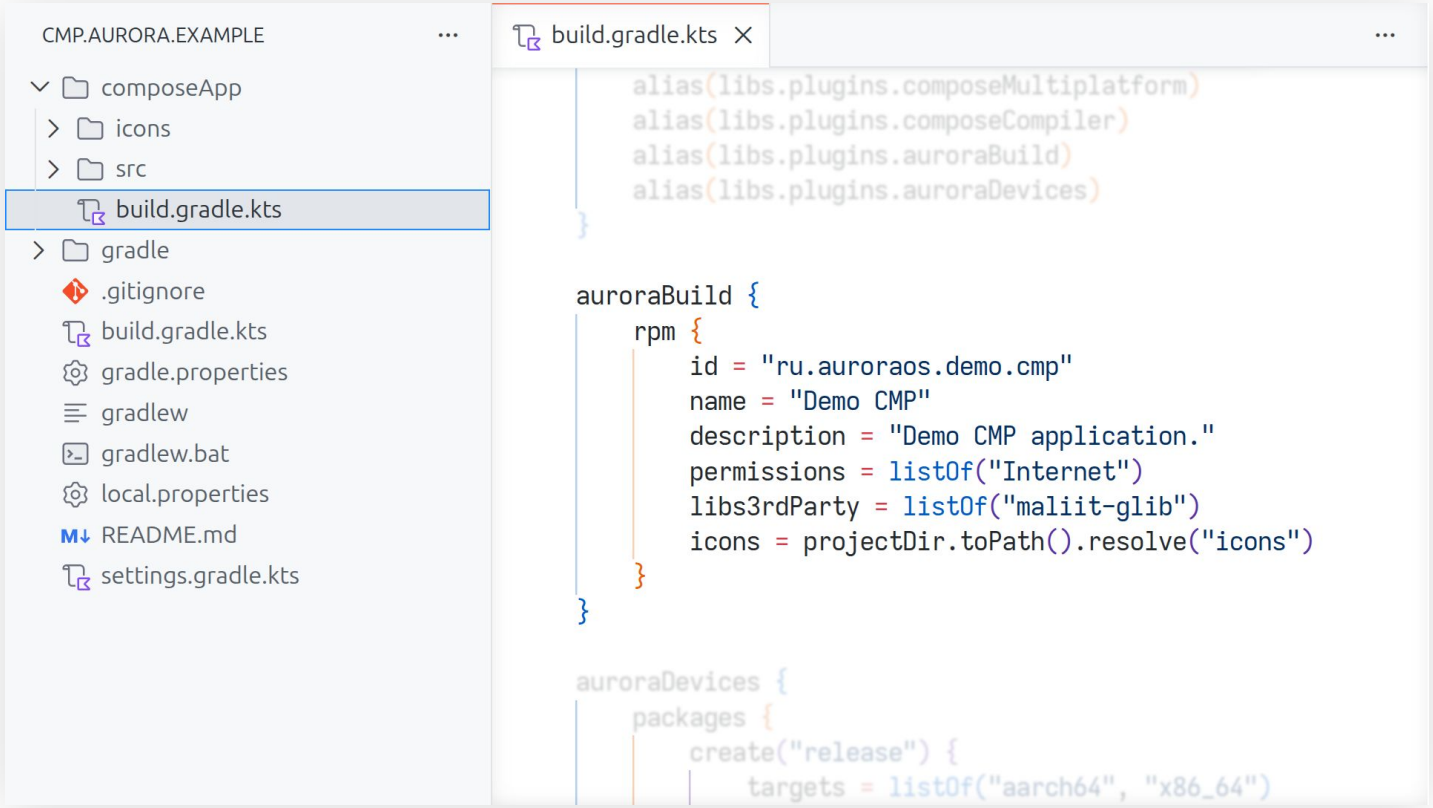
# **Шаг 7**. Добавление плагинов для ОС Аврора



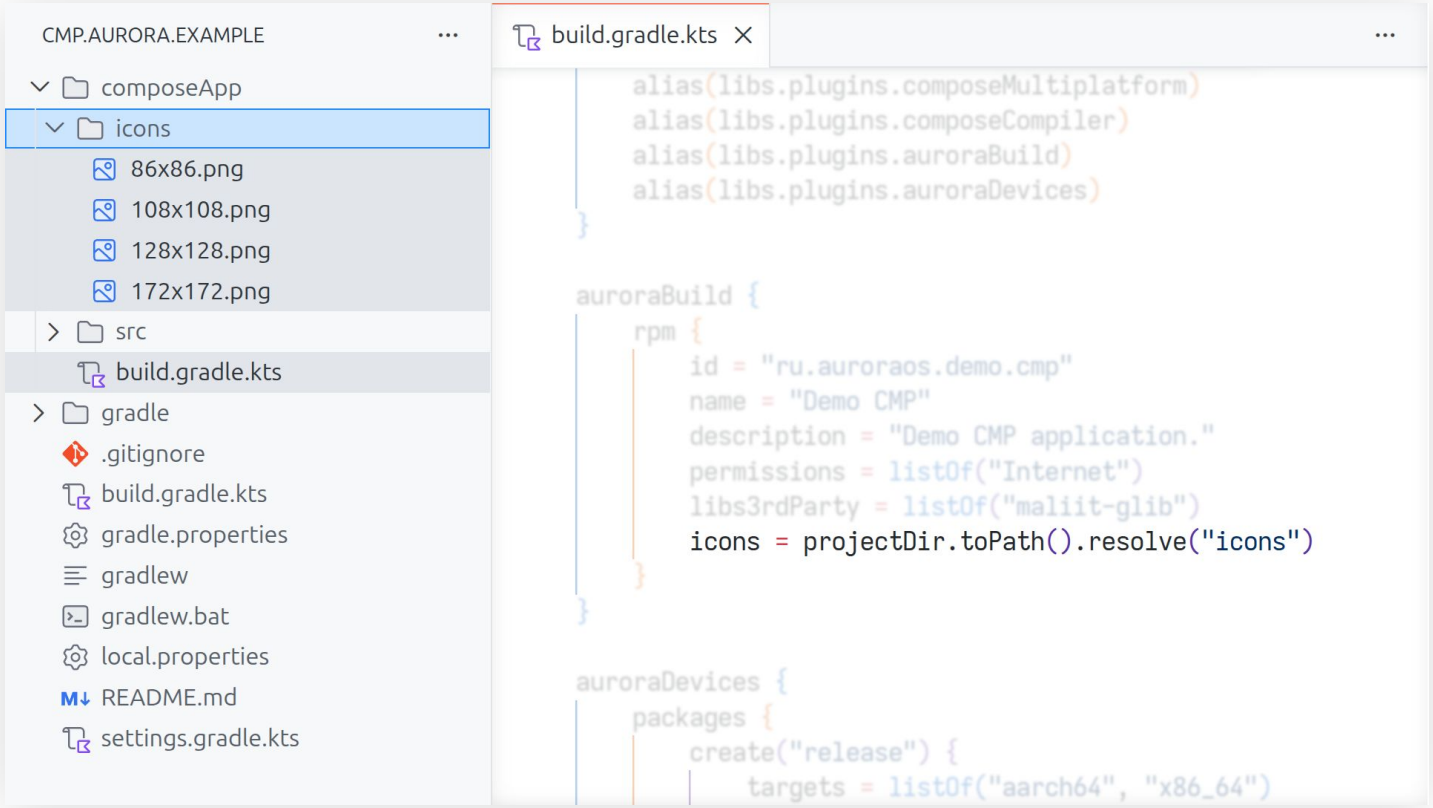
# **Шаг 7**. Добавление плагинов для ОС Аврора



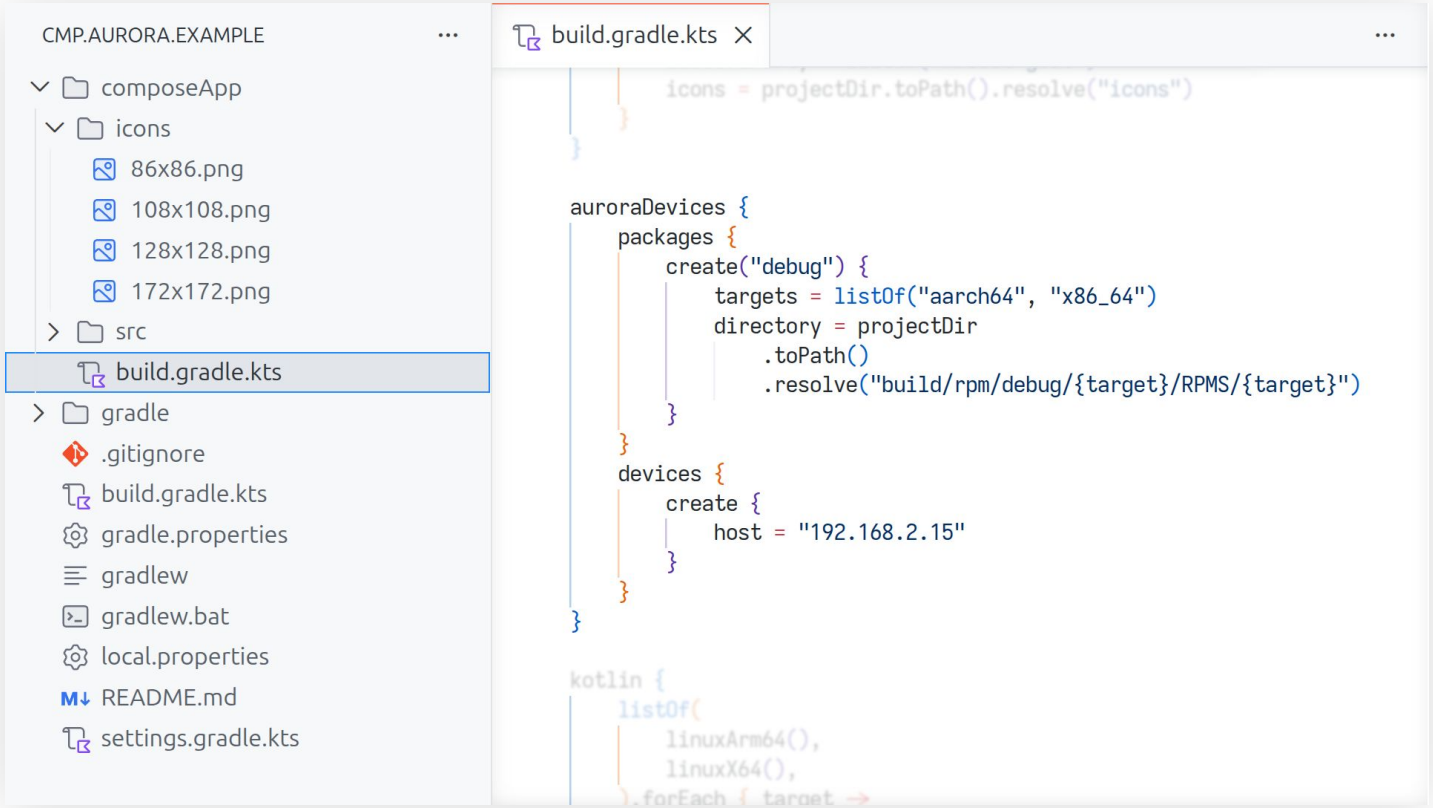
# **Шаг 8**. Конфигурация сборки приложения



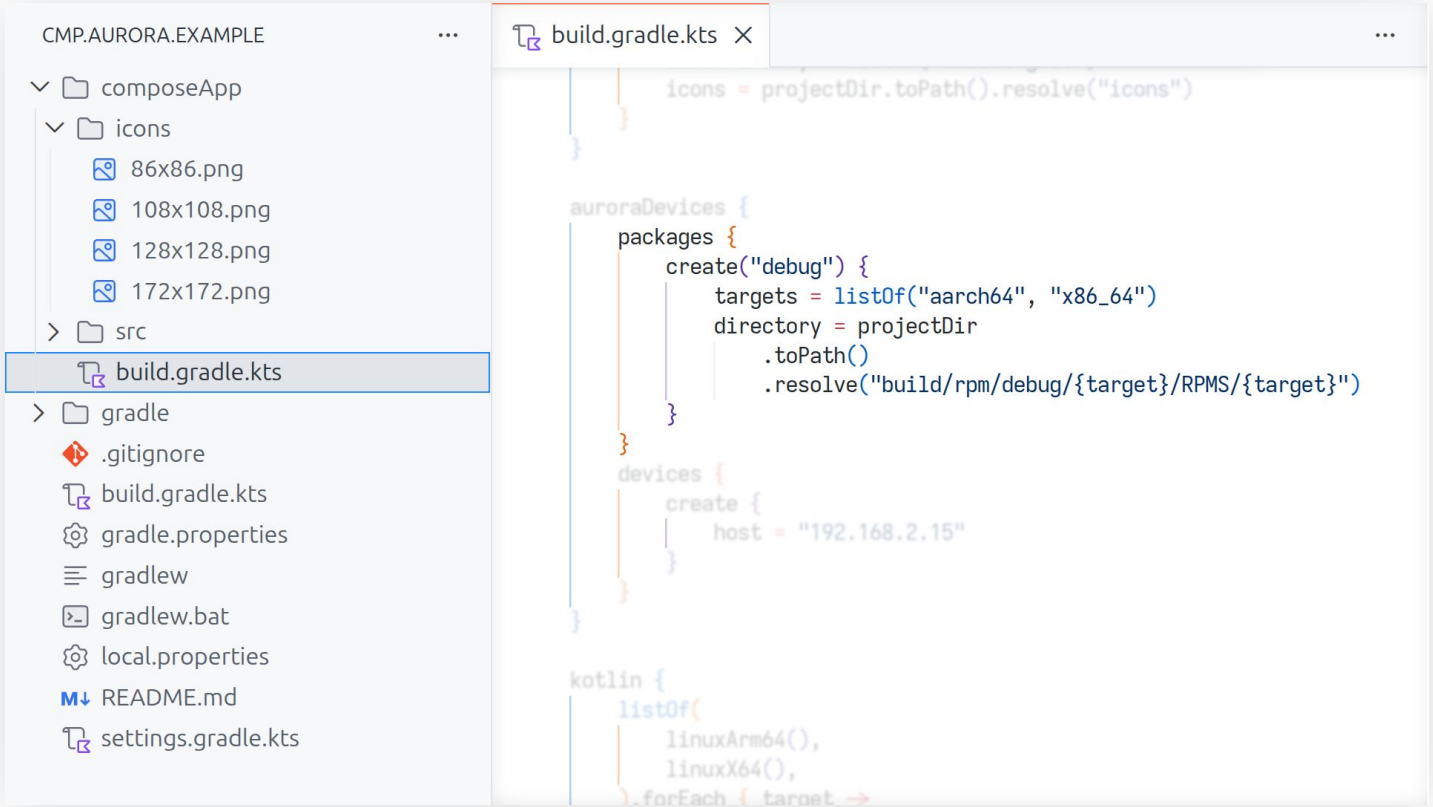
# **Шаг 8**. Конфигурация сборки приложения



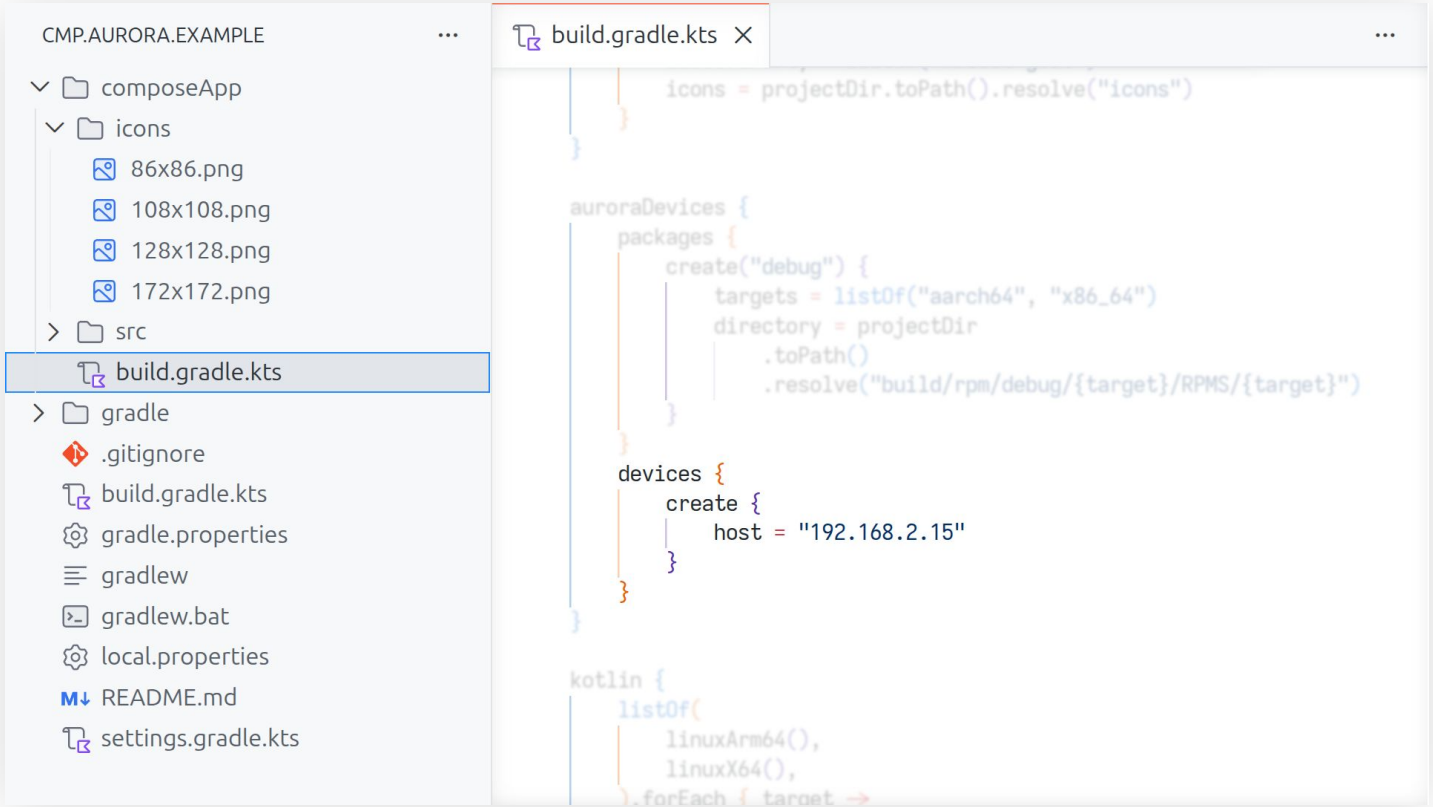
# **Шаг 9**. Конфигурация девайсов



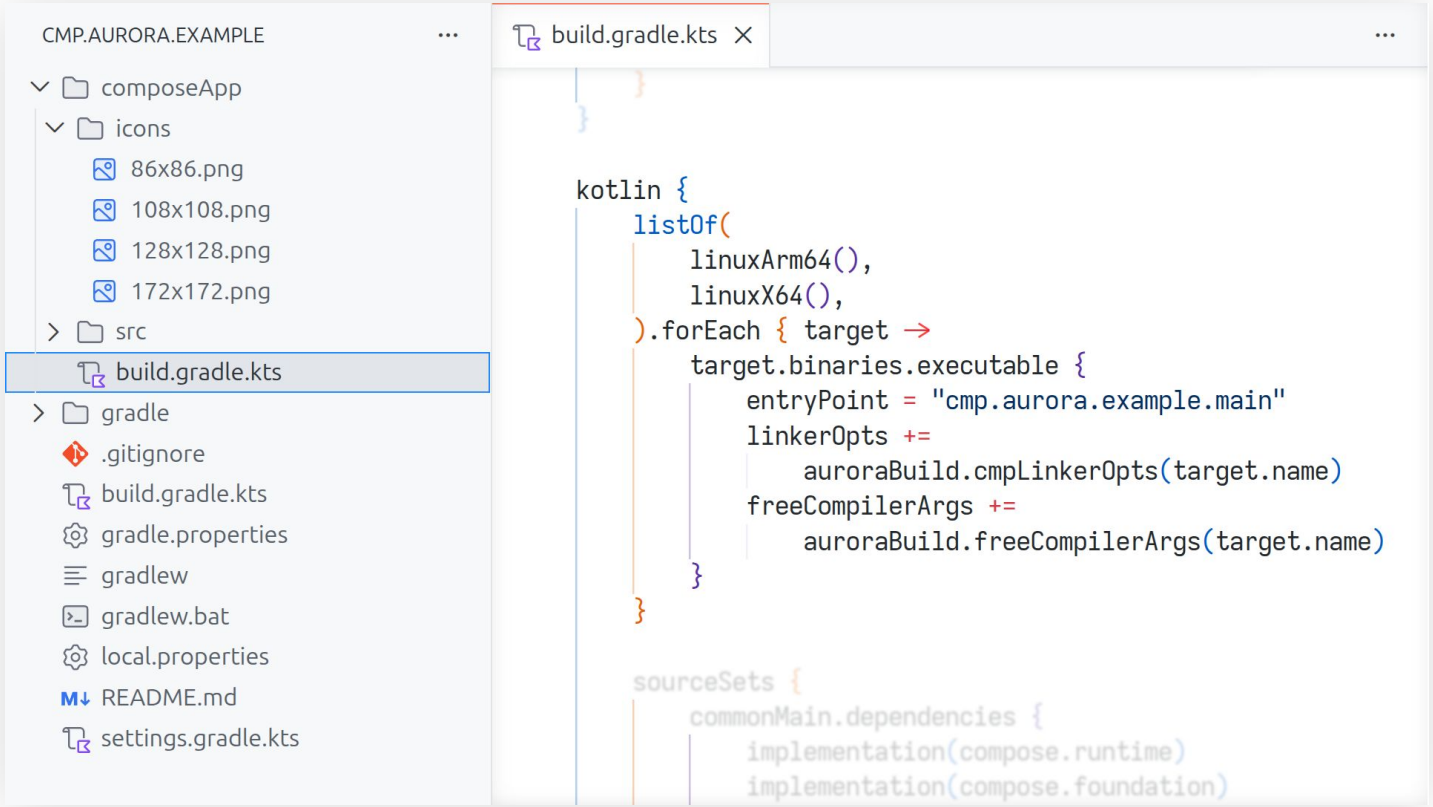
# **Шаг 9**. Конфигурация девайсов



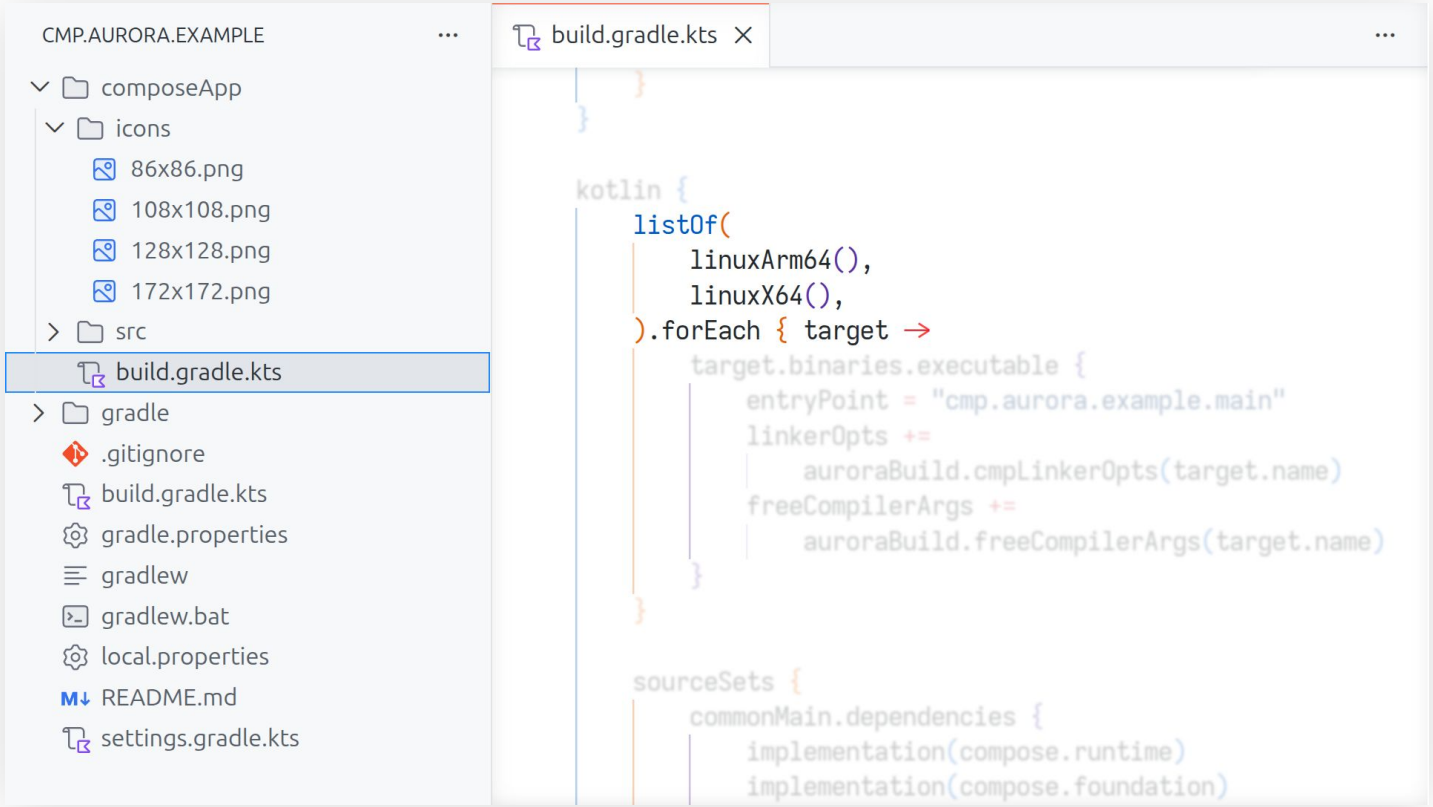
# **Шаг 9**. Конфигурация девайсов



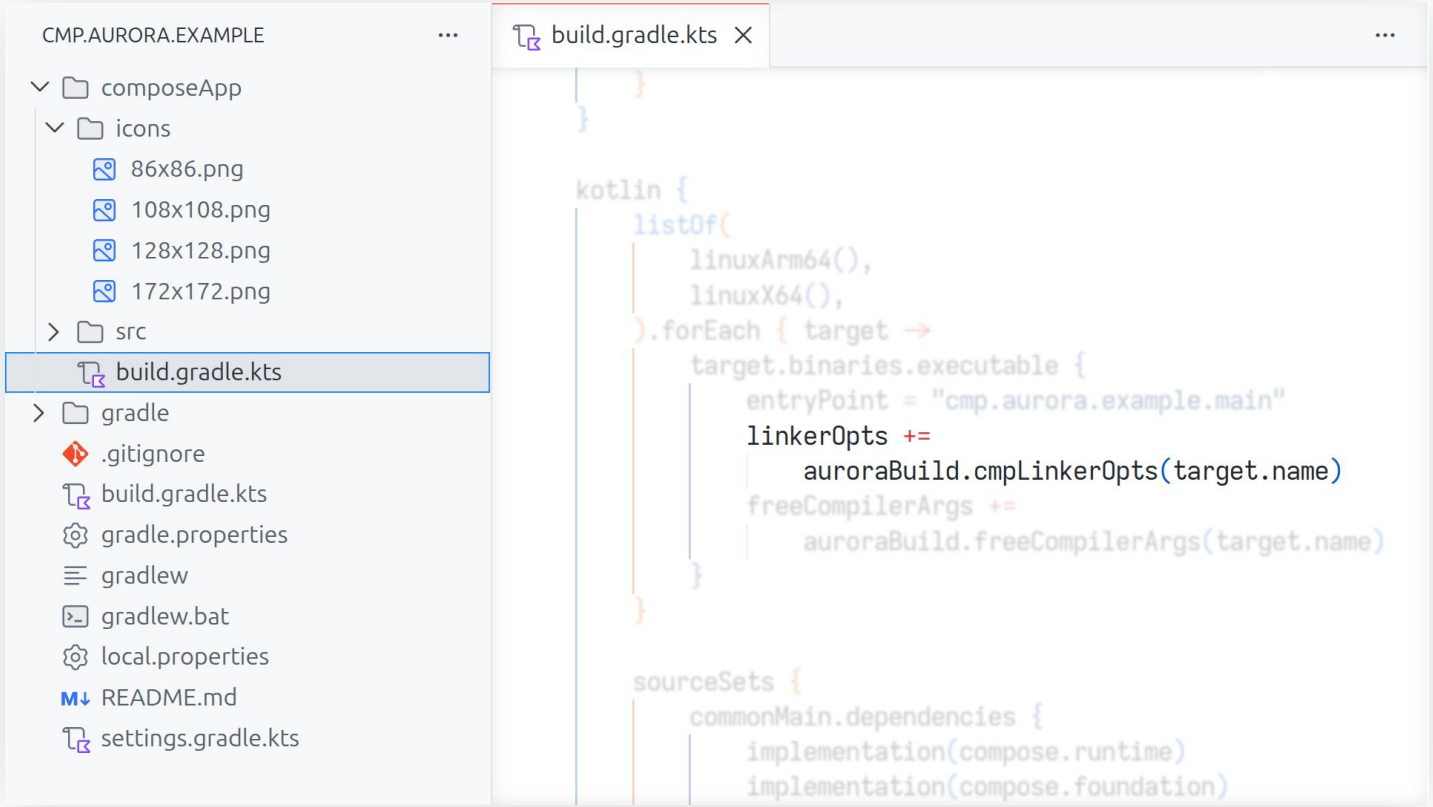
# **Шаг 10**. Конфигурация тергетов



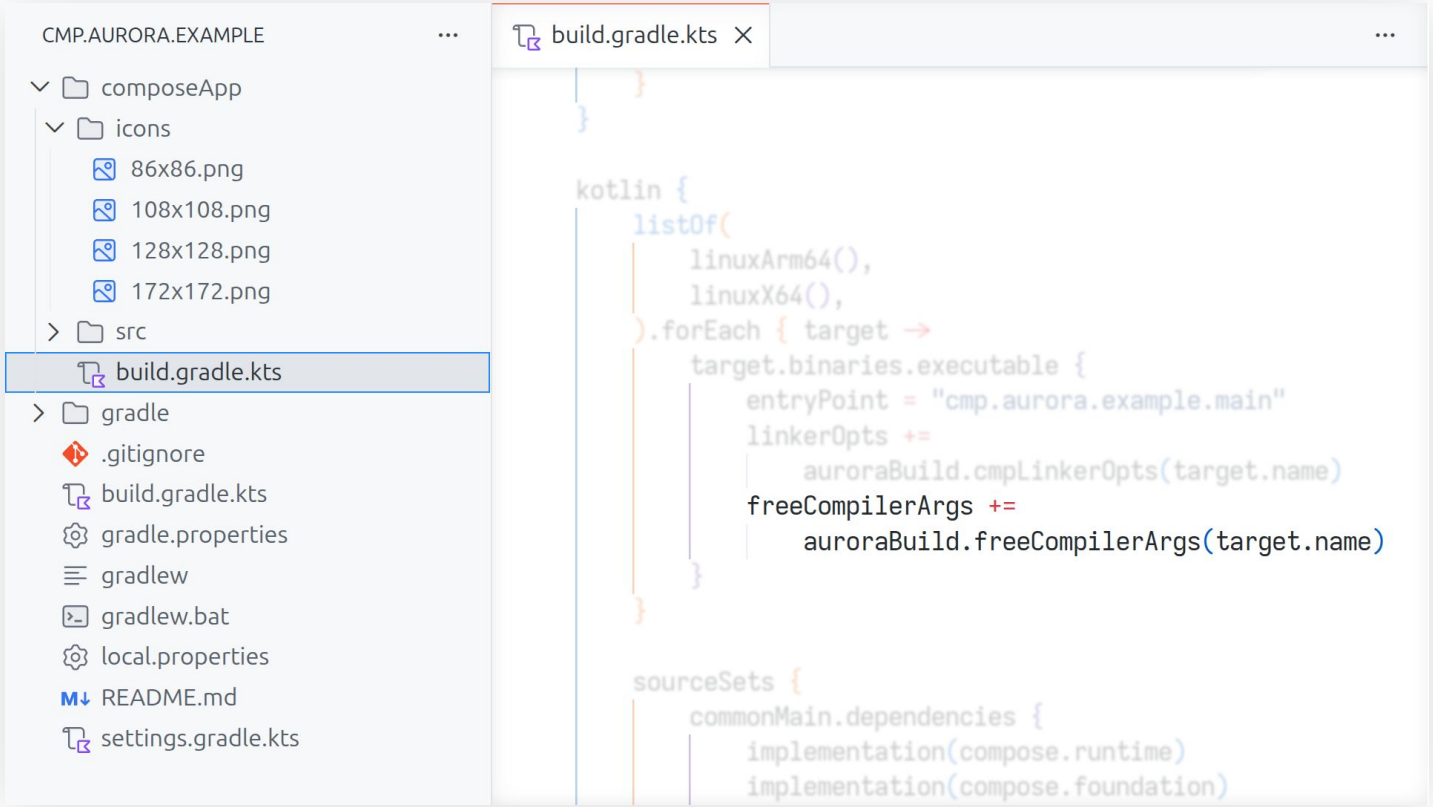
# **Шаг 10**. Конфигурация тергетов



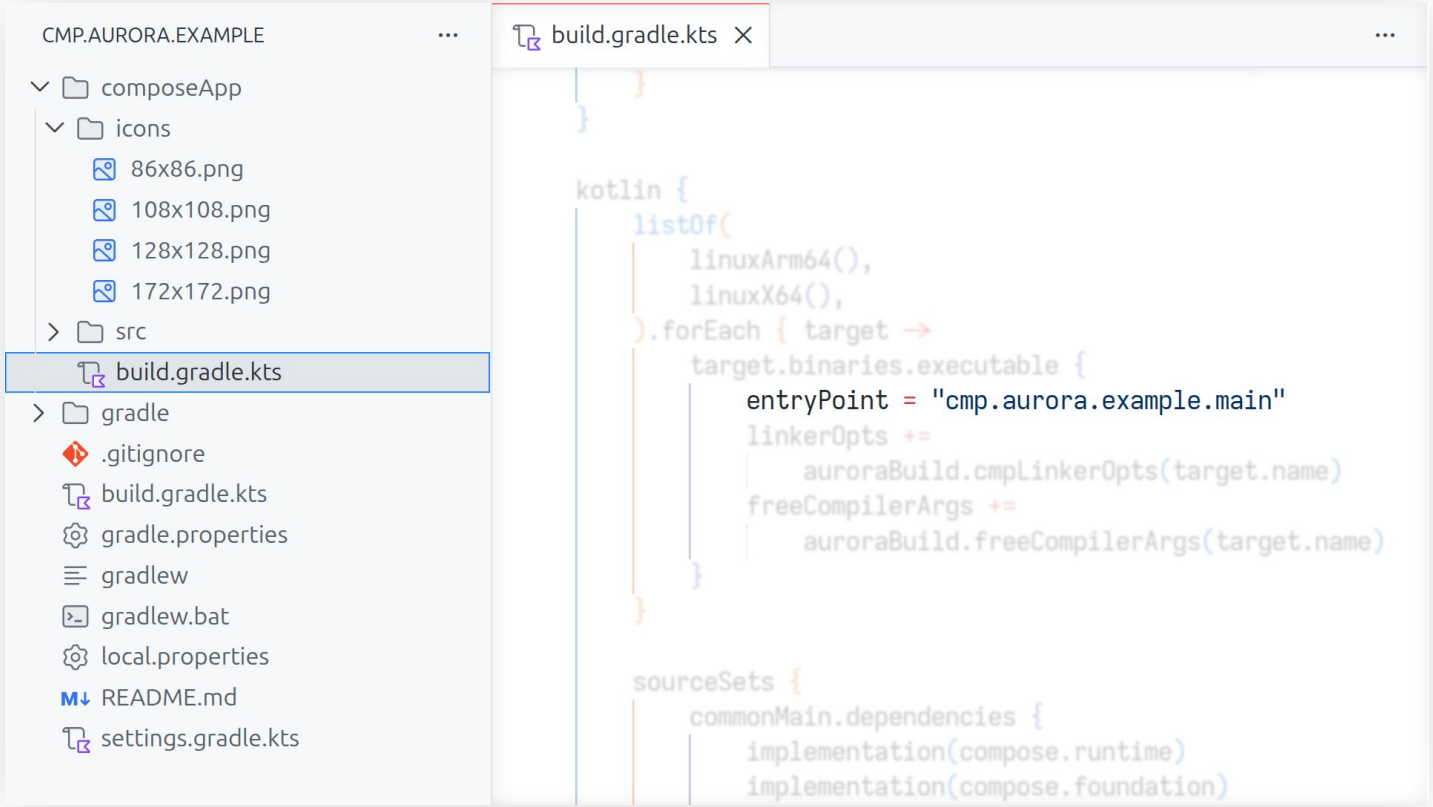
# **Шаг 10**. Конфигурация тергетов



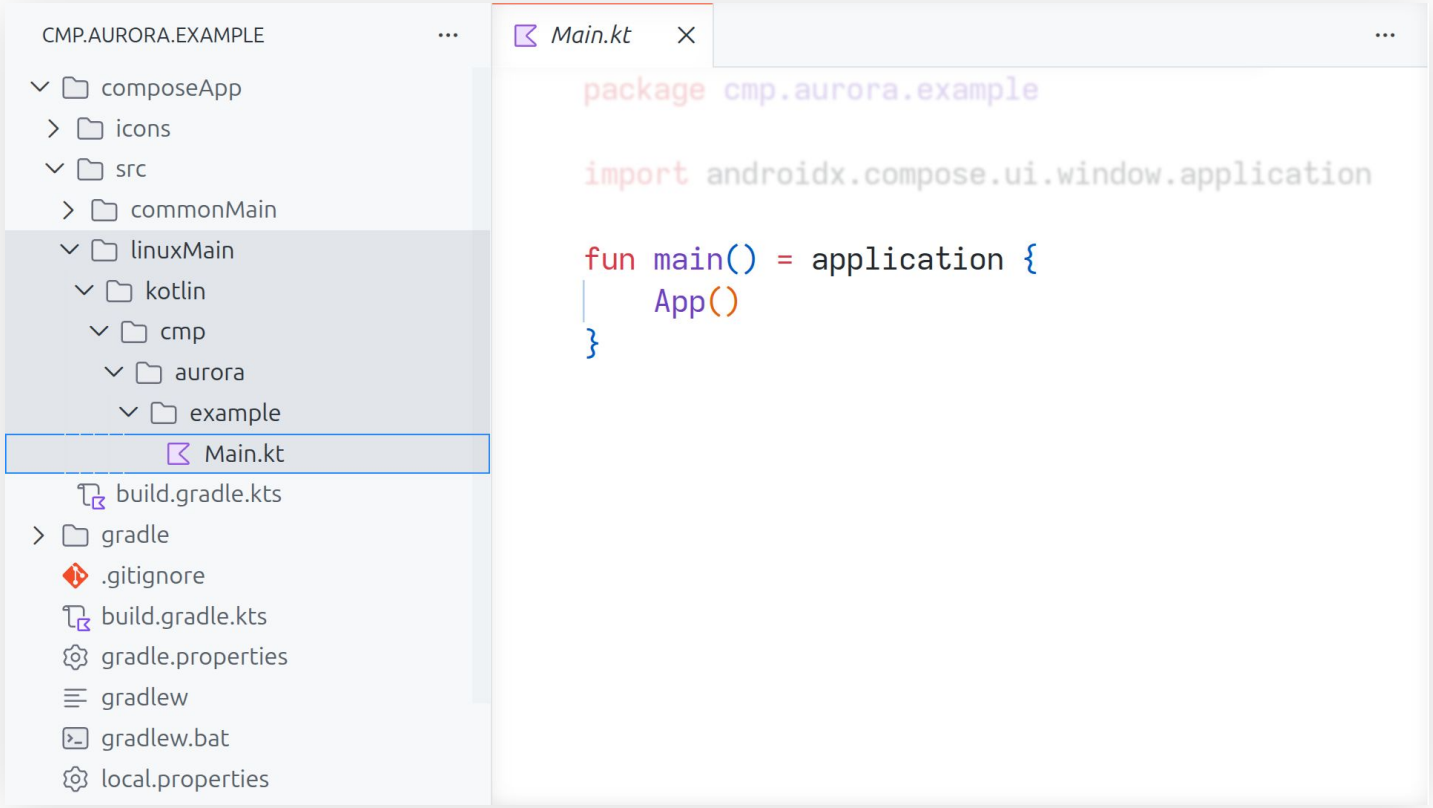
# **Шаг 10**. Конфигурация тергетов



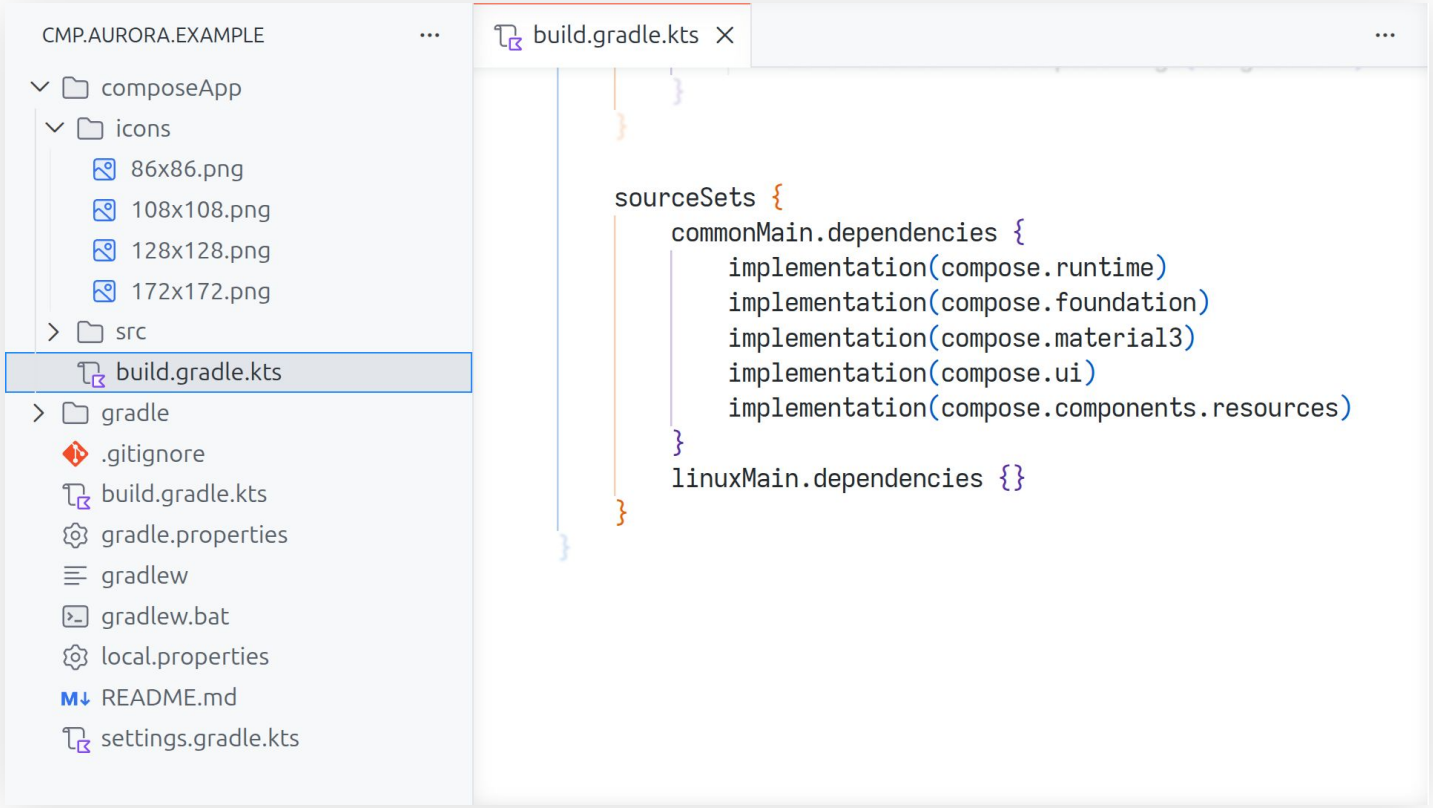
# **Шаг 10**. Конфигурация тергетов



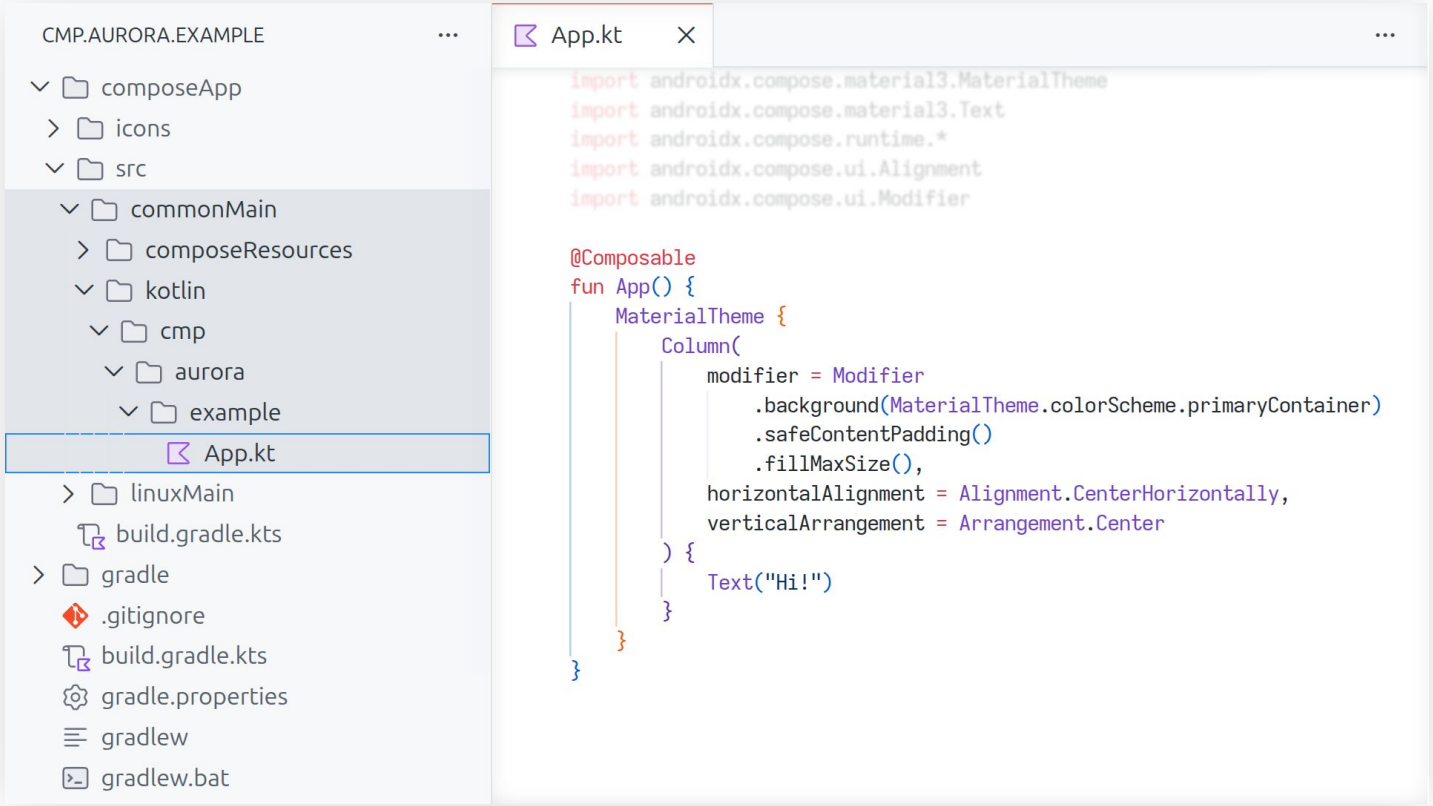
# **Шаг 10**. Конфигурация тергетов



# **Шаг 11**. Конфигурация зависимостей



# **Шаг 12**. Реализация приложения

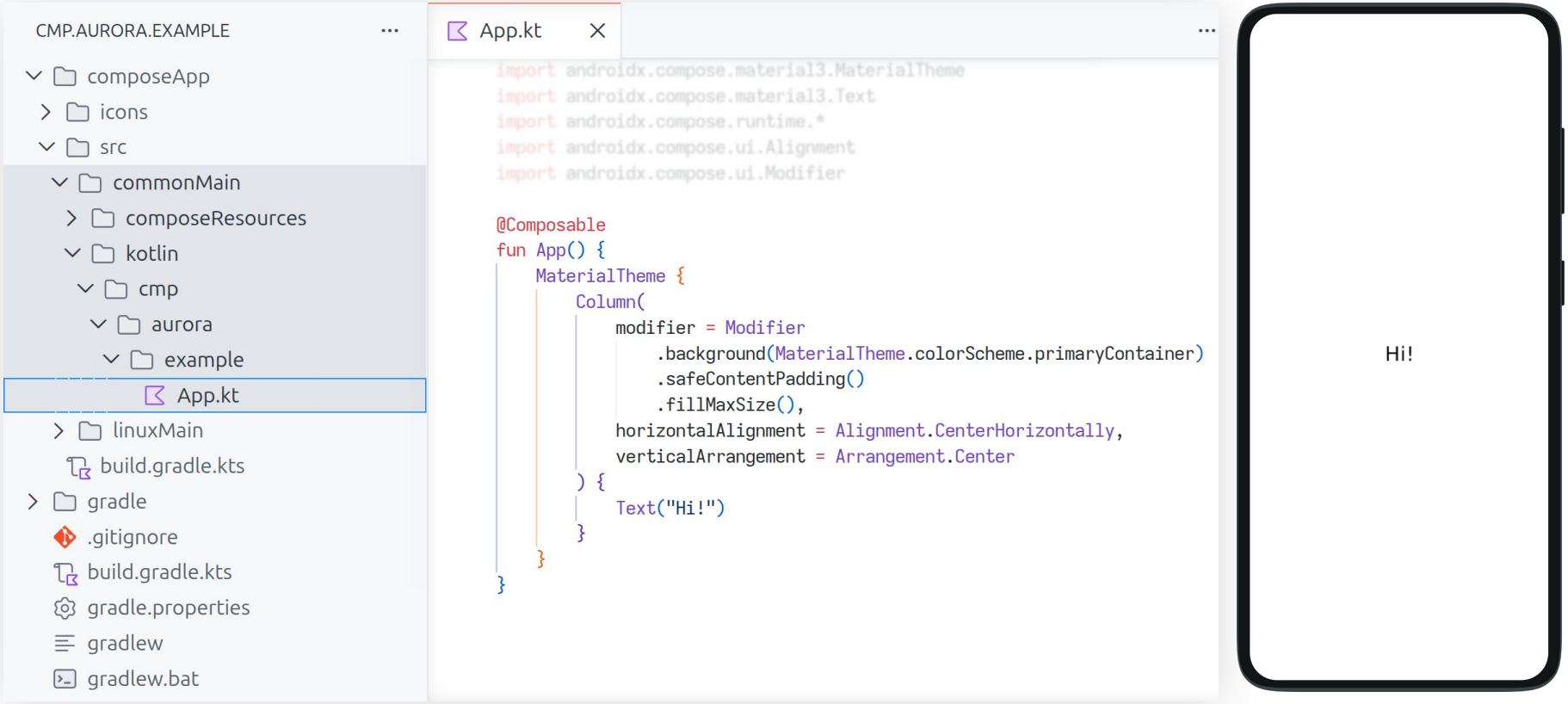


# **Шаг 13**. Сборка и запуск приложения

Сборка и запуск проекта

```
./gradlew :composeApp:buildDebugPipeline
./gradlew :composeApp:runDebugOnDevice
./gradlew :composeApp:runDebugOnEmulator
```

# **Шаг 13**. Сборка и запуск приложения



# **Шаг 13**. Сборка и запуск приложения

CMP.AURORA.EXAMPLE

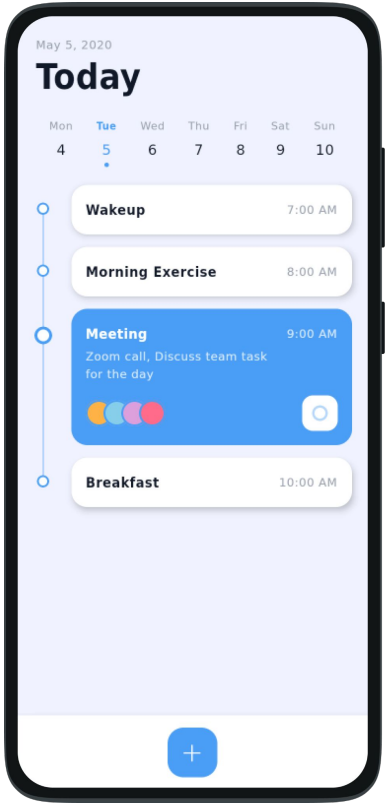
- composeApp
  - icons
  - src
    - commonMain
      - composeResources
      - kotlin
        - cmp
          - aurora
            - example
              - App.kt
      - linuxMain
      - build.gradle.kts
  - gradle
  - .gitignore
  - build.gradle.kts
  - gradle.properties
  - gradlew
  - gradlew.bat

App.kt

```
@Composable
fun App() {
    var selectedDay by remember { mutableStateOf(5) }
    var selectedEvent by remember { mutableStateOf(2) }

    val days = listOf(
        DayItem("Mon", 4), DayItem("Tue", 5),
        DayItem("Wed", 6), DayItem("Thu", 7),
        DayItem("Fri", 8), DayItem("Sat", 9),
        DayItem("Sun", 10),
    )
    val events = listOf(
        Event("Wakeup", "Early wakeup from bed and fresh", "7:00 AM
        Event("Morning Exercise", "4 types of exercise", "8:00 AM")
        Event("Meeting", "Zoom call, Discuss team task\nfor the day
        Event("Breakfast", "Morning breakfast with bread, banana\ne
    )

    MaterialTheme {
        Box(
            modifier = Modifier
                .fillMaxSize()
                .background(PageBg)
        ) {
```

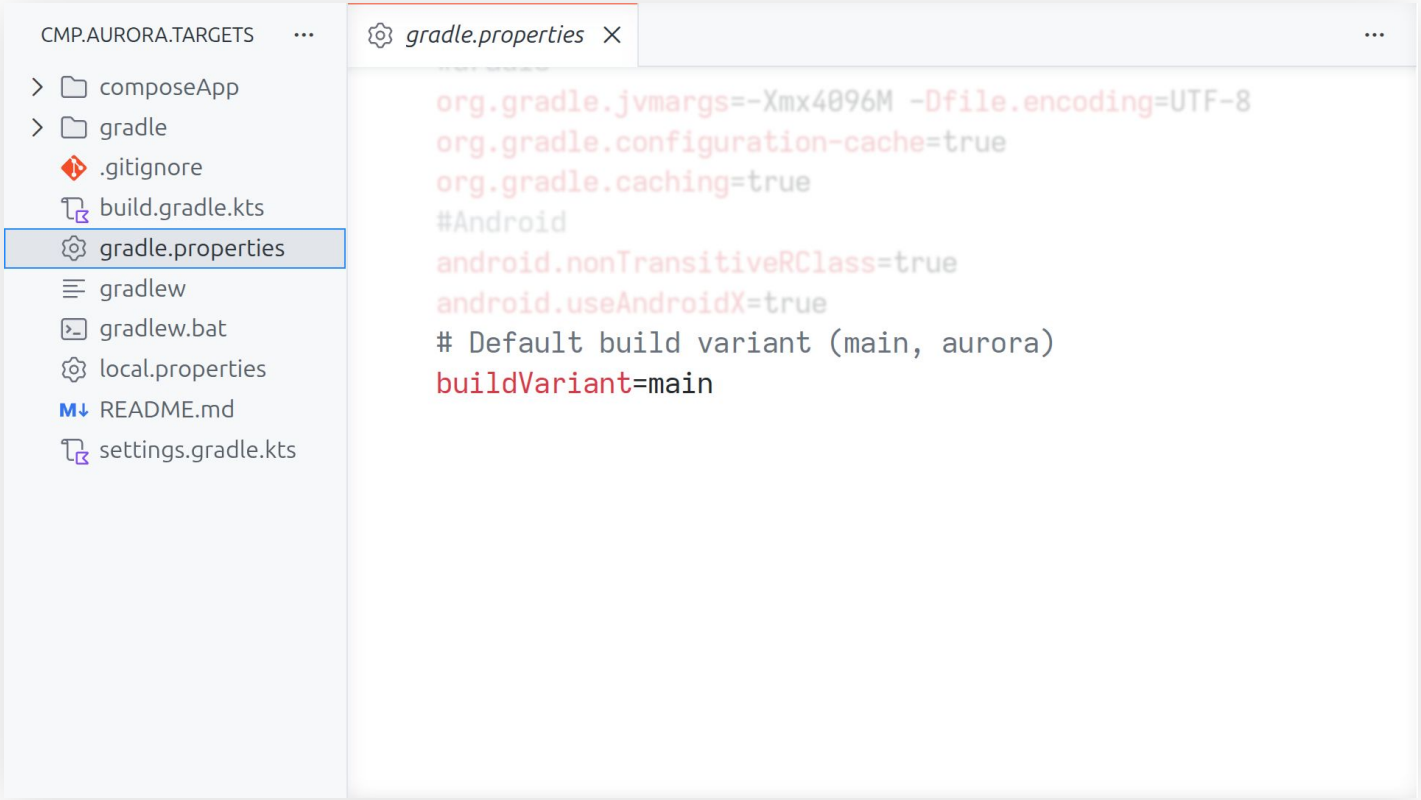


# Demo CMP

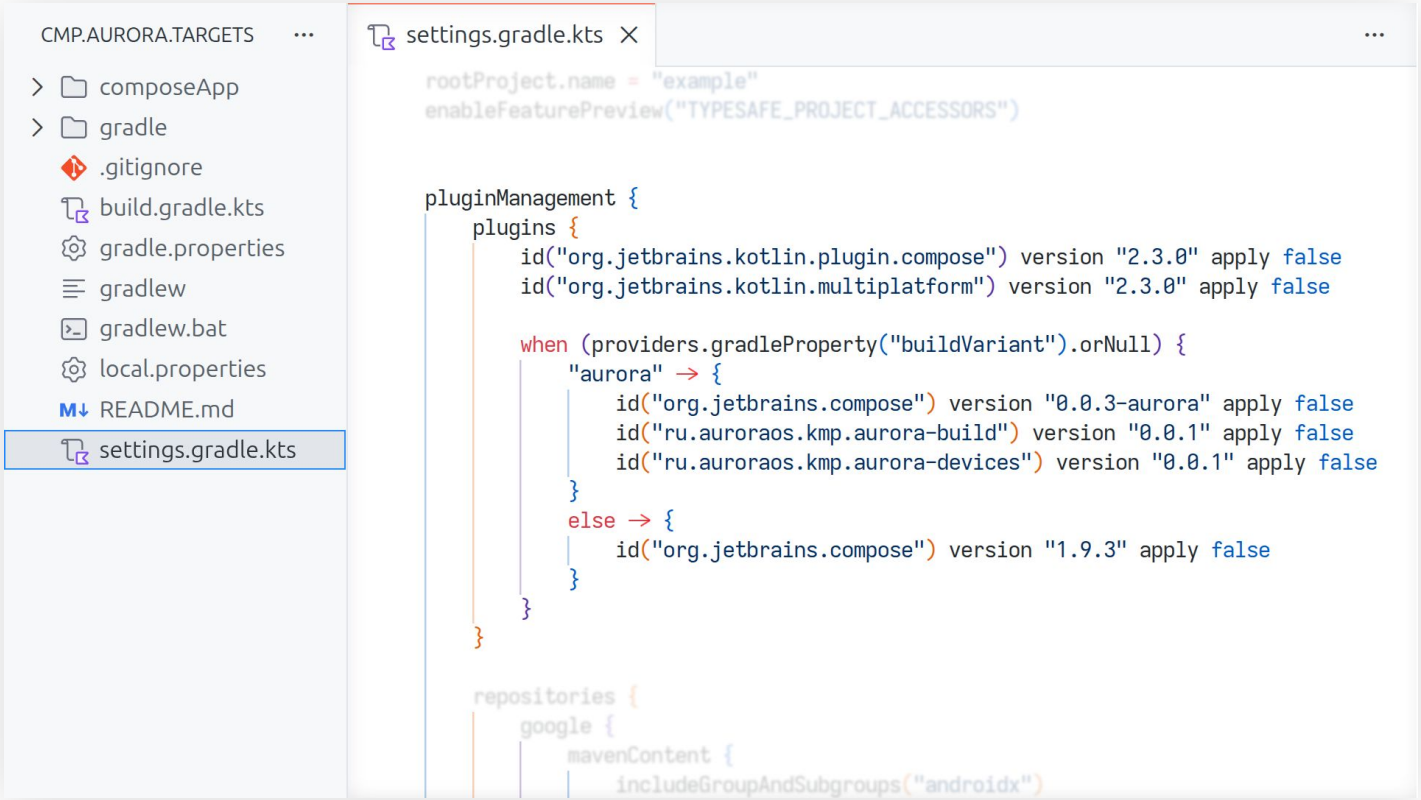
https://hub.mos.ru/auroraos/kotlin-multiplatform/demos/demo-cmp

# Как добавить поддержку ОС Аврора в существующий проект?

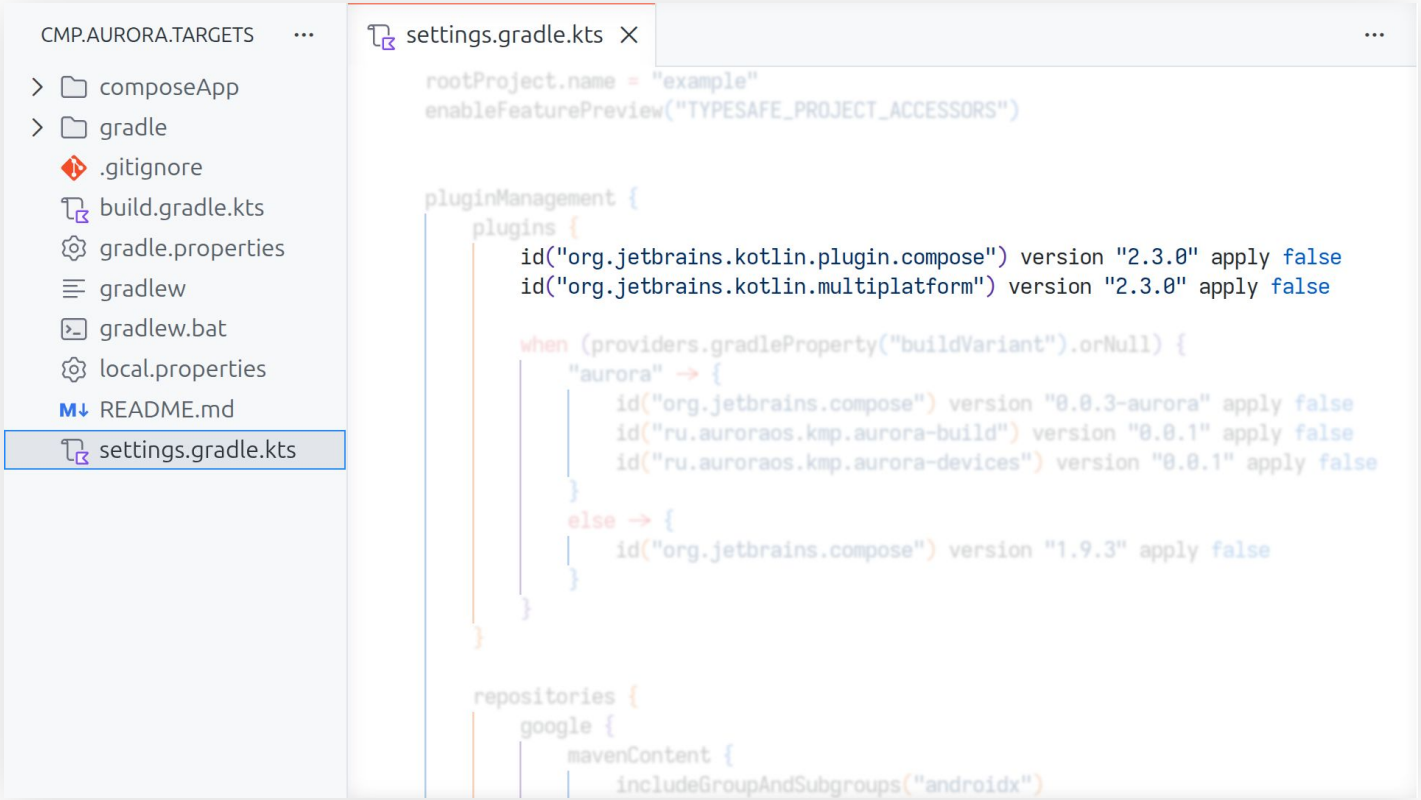
# **Шаг 1**. Добавление свойства Gradle



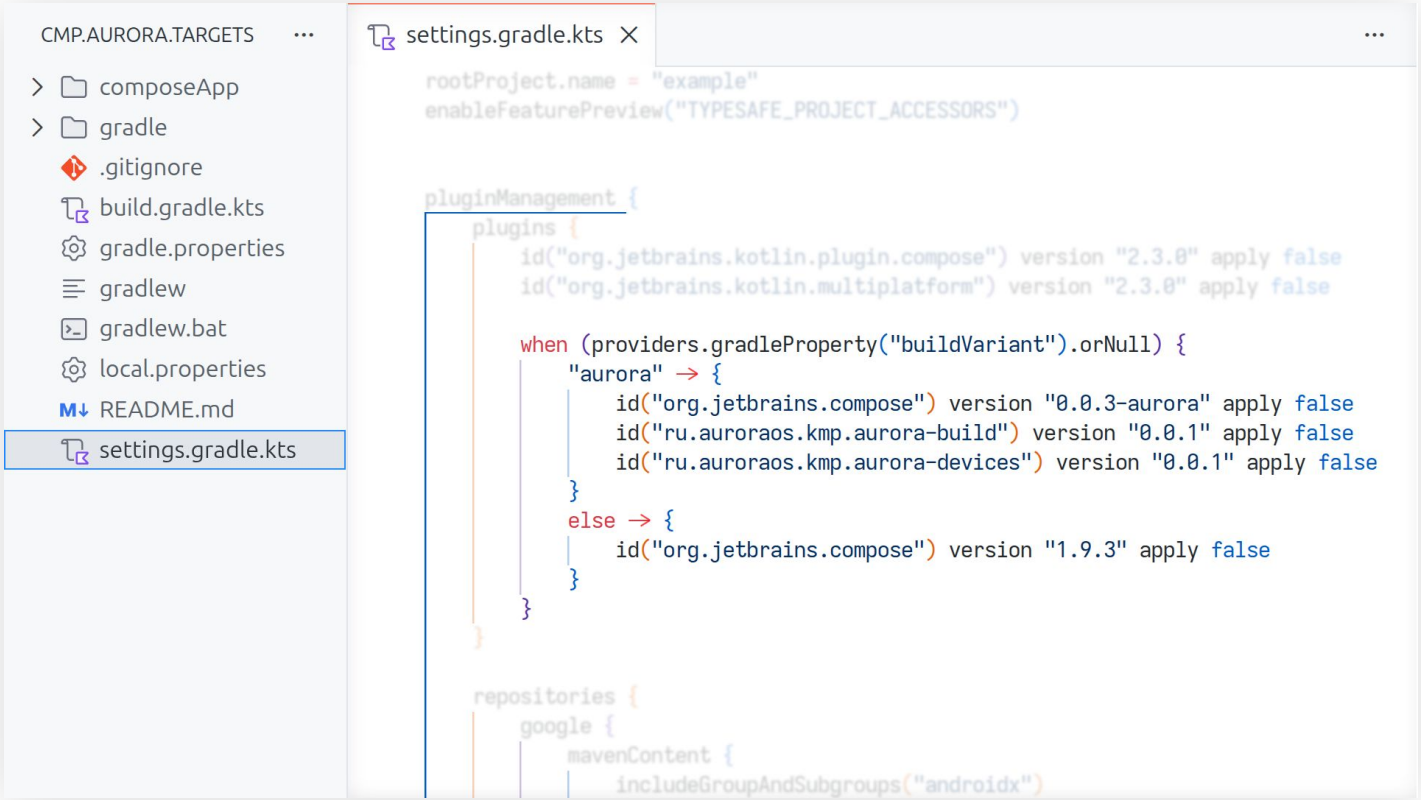
# **Шаг 2**. Настройка параметризированной сборки



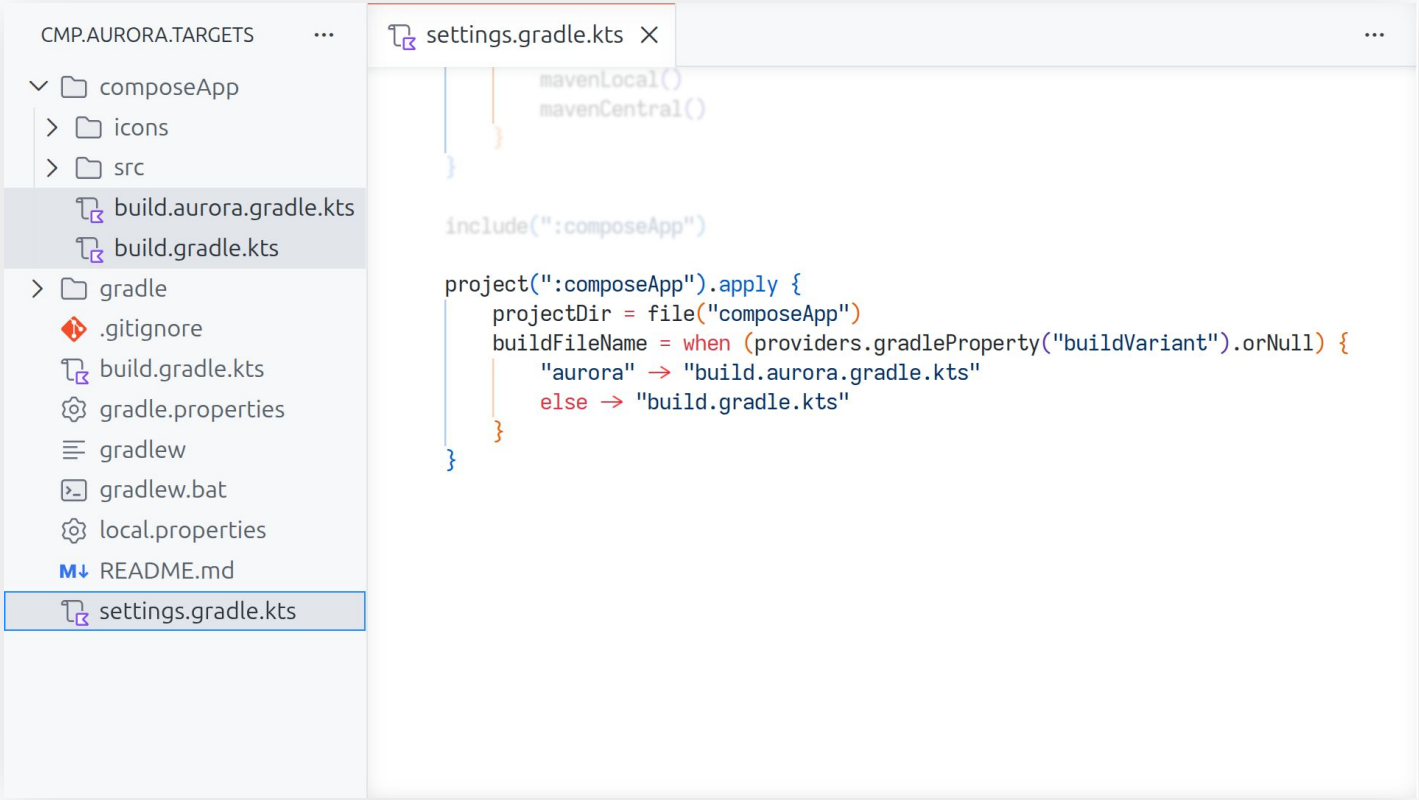
# **Шаг 2**. Настройка параметризированной сборки



# **Шаг 2**. Настройка параметризированной сборки



# **Шаг 2**. Настройка параметризированной сборки



# **Шаг 3**. Параметризованная сборка проекта

Параметризированная сборка проекта

```
./gradlew :composeApp:buildDebugPipeline -PbuildVariant=aurora
```

# Demo Targets CMP

https://hub.mos.ru/auroraos/kotlin-multiplatform/demos/demo-targets-cmp

# Roadmap по адаптации СМР для ОС Аврора

**4 квартал 2025 года**

1. Адаптация KMP
2. Адаптация KMP библиотек
3. Начало работ над CMP

**2 квартал 2026 года**

1. Стабилизация CMP
2. Развитие поддержки CMP
3. Работа с комьюнити
4. Адаптация популярных библиотек

2025 — 2026

**1 квартал 2026 года**

1. 🎉 Альфа версия CMP

**3 квартал 2026 года**

1. Развитие тулинга
2. Maven репозиторий
3. Поддержка
4. Работа с партнерами
5. Работа с комьюнити

### 4 квартал 2025 года

1. Адаптация KMP
2. Адаптация KMP библиотек
3. Начало работ над CMP

### 2 квартал 2026 года

1. Стабилизация CMP
2. Развитие поддержки CMP
3. Работа с комьюнити
4. Адаптация популярных библиотек

2025 — 2026

### 1 квартал 2026 года

1. 🎉 Альфа версия CMP

### 3 квартал 2026 года

1. Развитие тулинга
2. Maven репозиторий
3. Поддержка
4. Работа с партнерами
5. Работа с комьюнити

### 4 квартал 2025 года

1. Адаптация KMP
2. Адаптация KMP библиотек
3. Начало работ над CMP

### 2 квартал 2026 года

1. Стабилизация CMP
2. Развитие поддержки CMP
3. Работа с комьюнити
4. Адаптация популярных библиотек

2025 — 2026

### 1 квартал 2026 года

1. 🎉 Альфа версия CMP

### 3 квартал 2026 года

1. Развитие тулинга
2. Maven репозиторий
3. Поддержка
4. Работа с партнерами
5. Работа с комьюнити

# Список адаптированных библиотек

2025

2026

### 4 квартал 2025 года

1. Адаптация KMP
2. Адаптация KMP библиотек
3. Начало работ над CMP

### 1 квартал 2026 года

1. 🎉 Альфа версия CMP

### 2 квартал 2026 года

1. Стабилизация CMP
2. Развитие поддержки CMP
3. Работа с комьюнити
4. Адаптация популярных библиотек

### 3 квартал 2026 года

1. Развитие тулинга
2. Maven репозиторий
3. Поддержка
4. Работа с партнерами
5. Работа с комьюнити

Бета программа

Aurora Developers

Документация

# Заключение

- Аврора — современная быстро развивающиеся мобильная операционная система
- Мы нацелены на предоставление удобных инструментов для экосистемы
- Мы приветствуем партнеров в совместном развитии платформы
- Фреймворк KMP/CMP уже находится в альфа версии и продолжает развиваться