

Úvod do Androidu

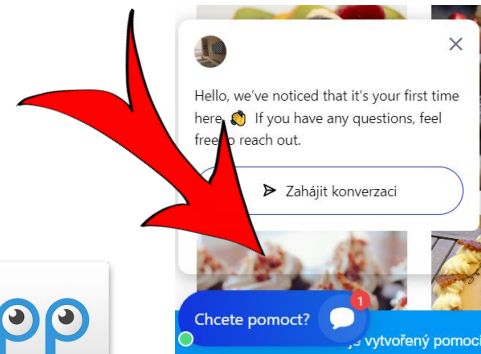
a možná i trochu víc

Jan Maděra



👋 Ahoj

- Android developer ve společnosti **Smartsupp**
- Vývoj pro Android přes 7 let
- [LinkedIn](#)





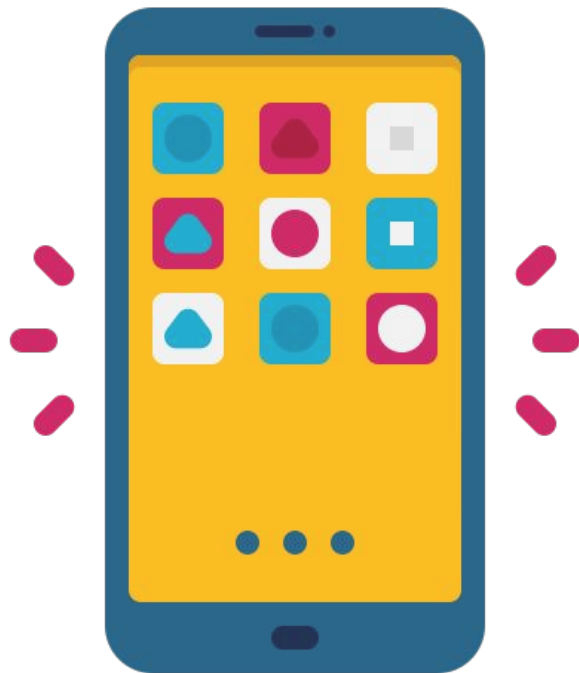
Osnova

- Nebojte se na cokoliv zeptat 😊
- Kde a jak se Android vyskytuje
- Jak vypadá vývoj pro Android
- Cross-platform
 - Kotlin Multiplatform Mobile
- Android vývoj v praxi
 - Architektury
 - Knihovny
 - ...
- Koho sledovat a kam dál?

 **Android okolo nás**

Kde všude dnes najdeme Android?

- Vlastně v podstatě všude
 - Telefony 📱
 - Tablety 📺
 - AndroidAuto 🚗
 - Brýle 🕶️

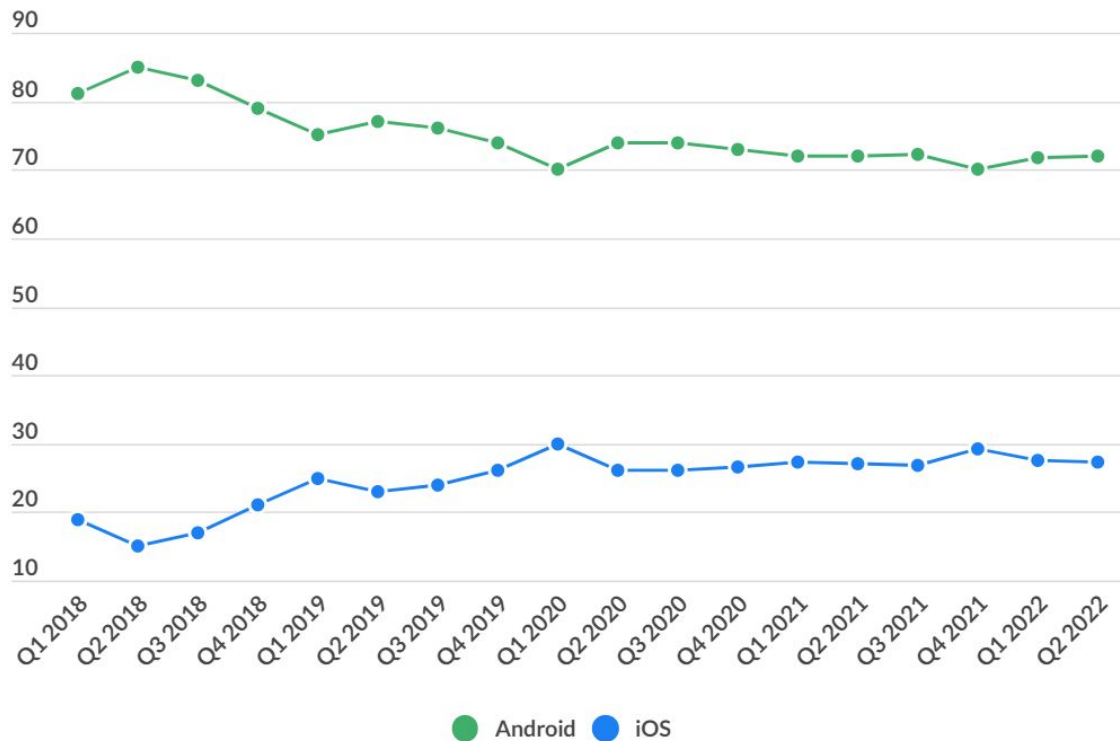


Android najdete i tam kde byste ho nečekali



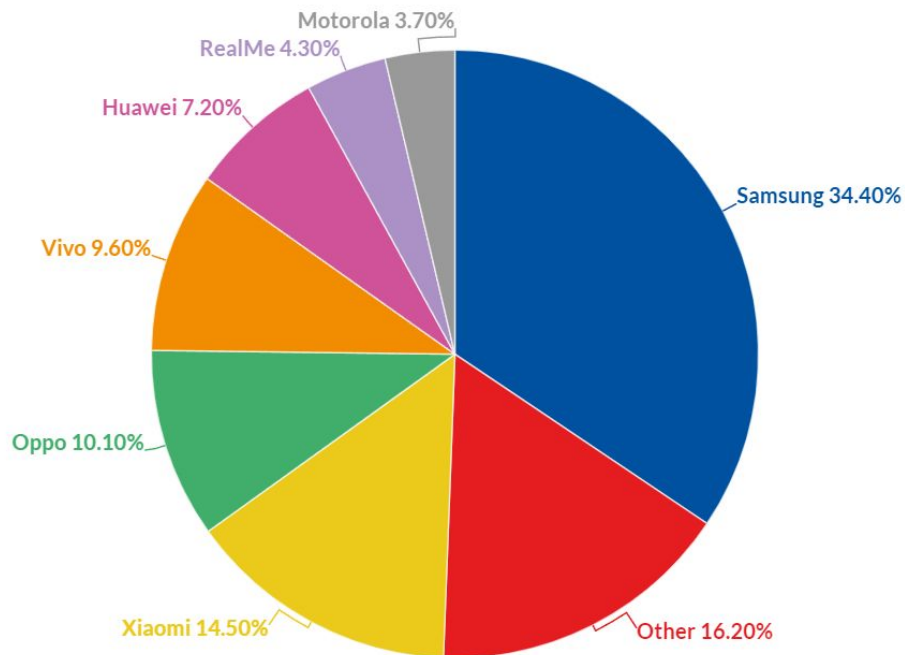
Android v číslech vs. iOS

Android vs iOS global market share (%)



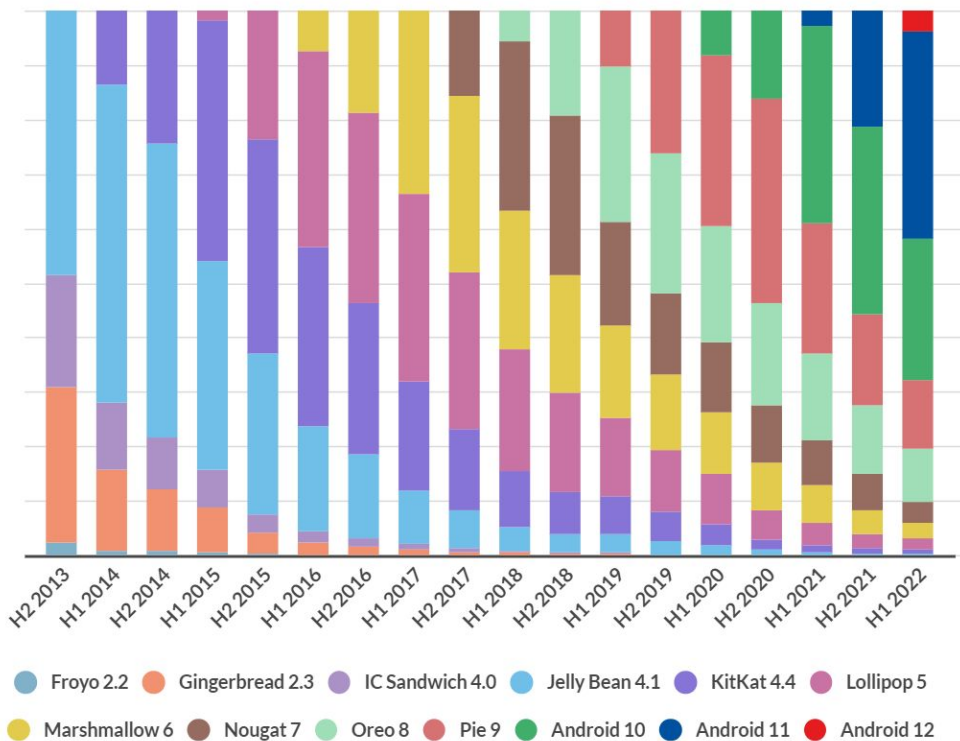
Android v číslech dle výrobce

Android vendor market share in 2022 (%)



Android v číslech dle verze

Android version market share 2013 to 2022 (%)



🤔 Vývoj s tolika verzemi je složitý!

- **Ano** i **NE**
- Google vývojáři se velmi snaží o zpětnou kompatibilitu
- Problém nastává pokud nová verze přinese nějakou **novou funkci** nebo naopak přinese nějaké **nové omezení** (zvýšení bezpečnosti)
- 🍏 *má podobný problém (né tak častý), ale Apple nehledí na zpětnou kompatibilitu, ale tlačí uživatele do updatu verze (tzn. může znamenat koupi nového modelu)*

**? Jak se takový problém řeší v
praxi ?**

❌ Problém - Zpětná kompatibilita



✓ Řešení - Zpětná kompatibilita

```
39  if (Build.VERSION.SDK_INT ≥ Build.VERSION_CODES.O) {  
40      someMethod()  
41  } else {  
42      someMethodWithLegacyStuff()  
43  }  
44
```

A code editor snippet showing a version check. The code is as follows:

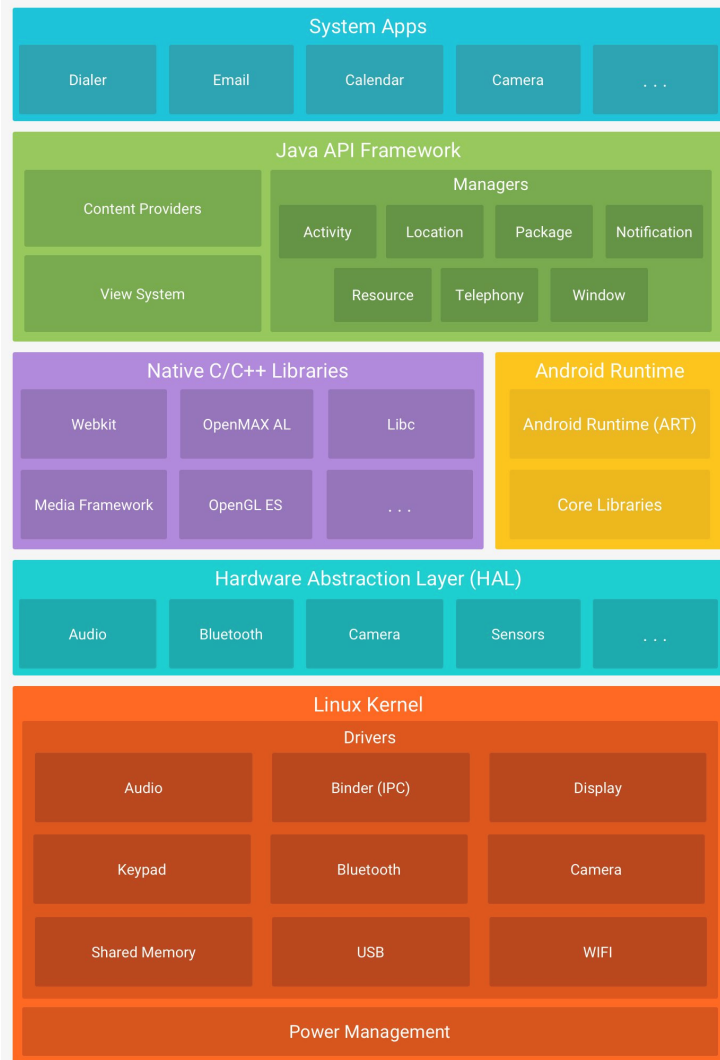
```
39  if (Build.VERSION.SDK_INT ≥ Build.VERSION_CODES.O) {  
40      someMethod()  
41  } else {  
42      someMethodWithLegacyStuff()  
43  }  
44
```


Three red arrows are present:
1. One arrow points from the top left towards the condition `Build.VERSION.SDK_INT ≥ Build.VERSION_CODES.O` on line 39.
2. Another arrow points from the top right towards the same condition on line 39.
3. A third arrow points from the bottom center towards the `else` block, specifically towards the `someMethodWithLegacyStuff()` call on line 42.

Vývoj pro Android

🔍 Android pod pokličkou

- Založen na Linuxu
- Open Source Projekt (AOSP)
 - source.android.com/
 - neobsahuje kritické komponenty jako **Play Store**, **Google Play Service** a další Google aplikace
 - Pro vytvoření funkčního OS z AOSP jsou potřeba úpravy





Jak vypadá vývoj na Android?

- Android studio (IntelliJ)
 - Plně zdarma
 - Časté aktualizace (Beta, Canary)
- Android SDK
 - Podmnožina JDK + **Android** “classy” + knihovny (Apache, ...)
- **Kotlin**
 - Dříve Java (do května 2017), ale starší code base se píše stále v Javě, ale firmy se snaží o přepis
 - 6
 - 7 od 4.4 - syntaxe, ne plné API, (desugar)
 - 8 od 7.0 - syntaxe, ne plné API, (D8, desugar)
- Gradle build system
 - Groovy, **Kotlin DSL**

Kotlin (<https://kotlinlang.org/>)

- *Moderní, stručný a bezpečný*
- JetBrains
- Plně kompatibilní s Javou
- Cross-Platform
 - **Android**
 - Kotlin-Multiplatform - **Android** + iOS
 - ServerSide
 - Spring
 - Ktor





vs.



Java vs. Kotlin - Metody, Signatury, zápis proměnných

```
public String foo(int input) {  
    String testText = "";  
    boolean testBoolean = false;  
    int testInt = 5;  
    double testDouble = 5.5;  
  
    return "5";  
}
```

```
fun foo(input: Int): String {  
    val testText1 = ""  
    val testText2: String = ""  
    val testTextBoolean = ""  
    val testTextInt = ""  
    val testTextDouble = ""  
  
    return "5"  
}
```

Java vs. Kotlin - Třídý

- Java

```
class User {  
    private String name;  
    private String age;  
}
```

- Kotlin

- **data class** - automaticky vygenerované `hashCode` a `copy`
- **copy** - metoda vrací kopírovaný objekt v těle lze měnit parametry (vhodné pro immutable)

```
class User(val name: String, val age: Int)  
  
data class DataUser(val name: String, val age: Int)
```

? Jak se píše UI pro Android ?

Staré dobré `.xml`

- Způsob používaný od prvopočátku
- Dnes se od něj pomalu odstupuje

```
<androidx.constraintlayout.widget.ConstraintLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">
```



```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/recycler_view"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:listitem="@layout/item_measured_consumption" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

Consumption

GAS

0.3 m3

28. 02. 2023

ELECTRICITY

1.4 kw/h

28. 02. 2023

GAS

2.4 m3

28. 02. 2023

ELECTRICITY

3.9 kw/h

28. 02. 2023

GAS

4.2 m3

28. 02. 2023

ELECTRICITY

5.4 kw/h

28. 02. 2023

GAS



JetPack Compose

- Dnes už doporučený způsob jak psát moderní UI

```
@Composable
fun AlphabetLetterListItem(item: AlphabetLetter) {
    Text(
        text = item.letter.toString(),
        style = MmlTypography.subtitle1,
        modifier = Modifier
            .fillMaxWidth()
            .background(Color.Black)
            .padding(start = 32.dp, top = 4.dp, bottom = 4.dp)
    )
}
```



```
@Preview
@Composable
fun PreviewAlphabetLetterListItem() {
    AlphabetLetterListItem(AlphabetLetter('A'))
}
```



Flutter vs. Android vs. iOS

```
class MyCard extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Row(children: [
      CircleAvatar(
        backgroundImage: AssetImage('assets/sundar_pichai.jpeg'),
        radius: 30,
      ), // CircleAvatar
      SizedBox(
        width: 10,
      ), // SizedBox
      Column(
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Text(
            "Sundar Pichai",
            style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
          ), // Text
          Text(
            "Chief Executive Officer of Alphabet",
            style: TextStyle(
              fontSize: 16,
            ), // TextStyle
          ), // Text
        ],
      ) // Column
    ]); // Row
  }
}
```

Flutter

```
@Composable
fun MyCard() {
  Row(modifier = Modifier.fillMaxWidth().height(60.dp)) { this: RowScope
    Image(painter = painterResource(
      id = R.mipmap.sundar_pichai,
      contentDescription = null,
      modifier = Modifier
        .size(60.dp)
        .clip(CircleShape)
        .border(2.dp, Color.Gray, CircleShape)
    )
    Column(Modifier.fillMaxHeight().padding(horizontal = 10.dp),
      verticalArrangement = Arrangement.Center
    ) { this: ColumnScope
      Text(
        text = "Sundar Pichai"
      )
      Text(
        text = "Chief Executive Officer of Alphabet"
      )
    }
  }
}
```

Jetpack Compose

```
import SwiftUI


struct ContentView: View {
  var body: some View {
    HStack (
      spacing: 10
    ){
      Image("sundar_pichai_img").resizable().frame(width: 60, height: 60)
        .clipShape(Circle()).overlay(Circle().stroke(Color.gray, lineWidth: 2)
        )
      VStack(
        alignment: .leading,
        spacing:0
      ){
        Text("Sundar Pichai").font(.title2)
        Text("Chief Executive Officer of
          Alphabet").font(.body).foregroundColor(.gray)
      }
    }
  }
}
```

Swift UI



Vydávání aplikace

Google Play/Play Console/Vydání aplikace

- Jednorázový poplatek **\$25** 
- Co získáte?
 - Analytics v rámci vaší aplikace
 - rating, instalace, aktivní uživatelé, atd...
 - Delivery přes Google Console až do Google play
 - testovací kanály, inkrementální delivery
 - Správa placených produktů a subscriptions
 - *ale taky platíte 15% z částky Googlu*
 - API
 - např. pro Continuous delivery

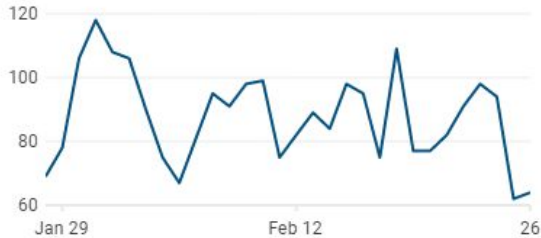


Google Play Console

Store listing performance

Store listing acquisitions ⓘ

2.63K +3.95% ▲



— Store listing acquisitions

[Explore →](#)

Your KPIs

New users acquired ⓘ

New users acquired ⓘ

2.2K +5.61% ▲



— New users acquired
— New users acquired (30 days rolling average)

[Explore →](#)

User loss ⓘ

User loss ⓘ

3.35K -7.24% ▼



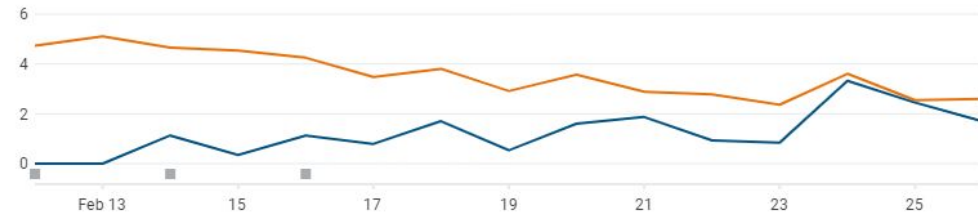
— User loss — User loss (30 days rolling average)

[Explore →](#)

Latest release performance

Crashes per 1,000 devices ⓘ

1.412 -60.68% ▼ vs all releases



— Crashes per 1,000 devices (3.35.9) — Crashes per 1,000 devices (all releases)

Android^(Native) – Výhody a Nevýhody

Výhody

- Snadný vstup do vývoje
- Skvělá komunita a dokumentace
- Mladé a stále se vyvíjející odvětví
- **Kotlin**
- Oproti iOS 🍏 větší možnost využití v “průmyslu”
 - Kasy, Scannery do skladu etc.

Nevýhody

- Výrobci si přizpůsobují/optimalizují své operační systémy, které potom vedou k chybám na produkci
 - Typický příklad notifikace
- Mnoho verzí Androidu občas komplikuje code-base



Vývoj pro Android dnes



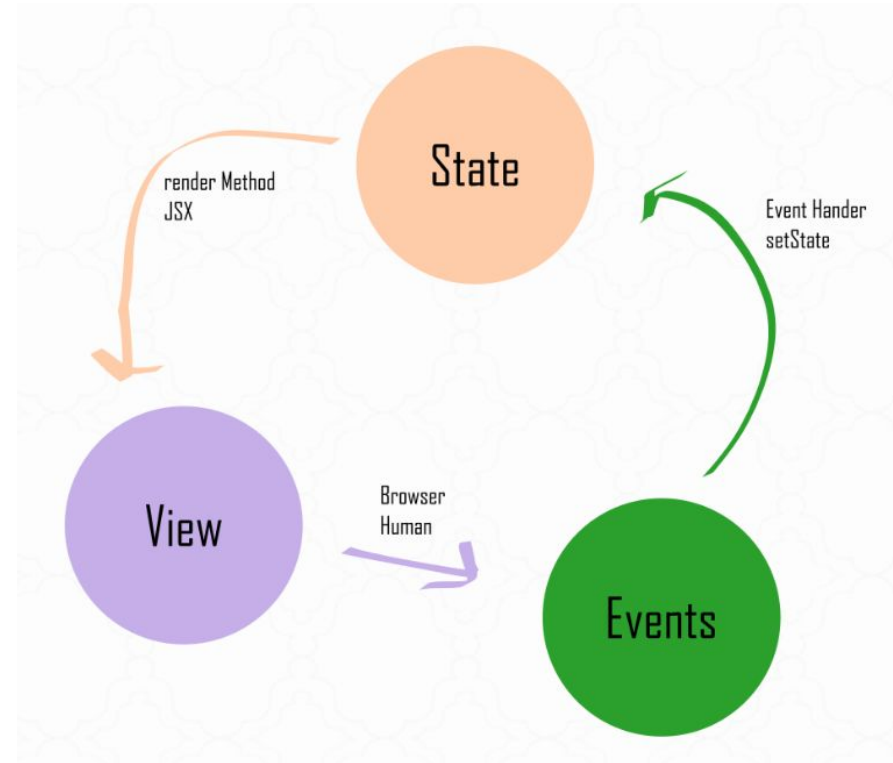
Směr vývoje pro Android

- Na co můžete narazit, když budete hledat v práci jako **Android vývojář/ka?**
 - Filozofie UI
 - Architektura aplikace
 - Knihovny
- Kterým **směrem** se pohybuje Android vývoj?
- **Má smysl se ještě učit Nativní Android vývoj**, když existují multiplatformní řešení
 - [AirBnb blog o vývoj v ReactNative](#)



Filozofie UI

- State rendering
- Programově deklarovatelné UI
- Podobný přístup lze vidět napříč všemi FrontEndovými frameworky
 - Flutter - Všechno je Widget
 - iOS - SwiftUI
 - React - Redux





Architektura aplikace

- V dnešní době se používají převážně dvě architektury
 - **MVVM** - Model-View-ViewModel
 - **MVI** - Model-View-Intent
- Využívají ViewModel pro práci se Statem, který je následně renderován
- Pracují s Observable patternem



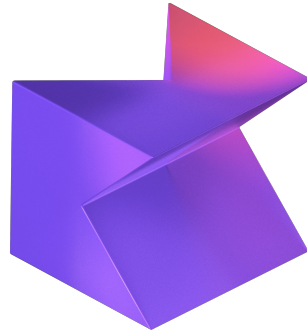
Knihovny

- [Retrofit](#) | [Fuel](#)
 - HTTP client
- [Room](#) | [SQLDelight](#)
 - Databáze nad SQLite
- [Hilt](#) | [Koin](#)
 - Dependency Injection
- [Coroutines](#) | [RxAndroid](#)
 - Asynchronicita
- [Glide](#) | [Picasso](#) | [Coil](#)
 - Práce s Obrázky
- [Timber](#)
 - Logování

👁️👁️ **Koho sledovat**

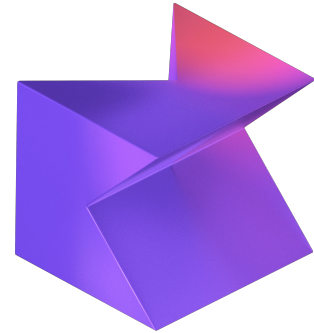
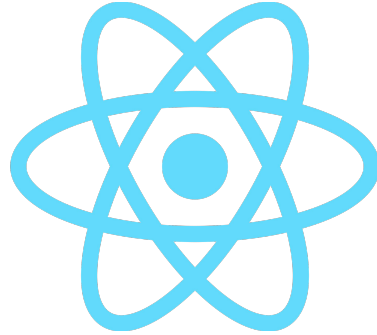
- **Jake Wharton**
 - jakewharton.com
- **Donn Felker and Kaushik Gopal**
 - <https://fragmentedpodcast.com/>
- **Adam McNeilly**
 - @AdamMc331
- **Chris Banes**
 - @chrisbanes
- **Alexey Bykov**
 - @nonewsss

Kotlin Mutlplatform Mobile



Cross-platform vývoj

- **Kotlin-Multiplatform**
 - Kotlin
- MAUI (Xamarin)
 - C#
- Flutter
 - Dart
- React Native
 - JavaScript
- Cordova
 - JavaScript





Kotlin-Multiplatform Mobile (Beta)

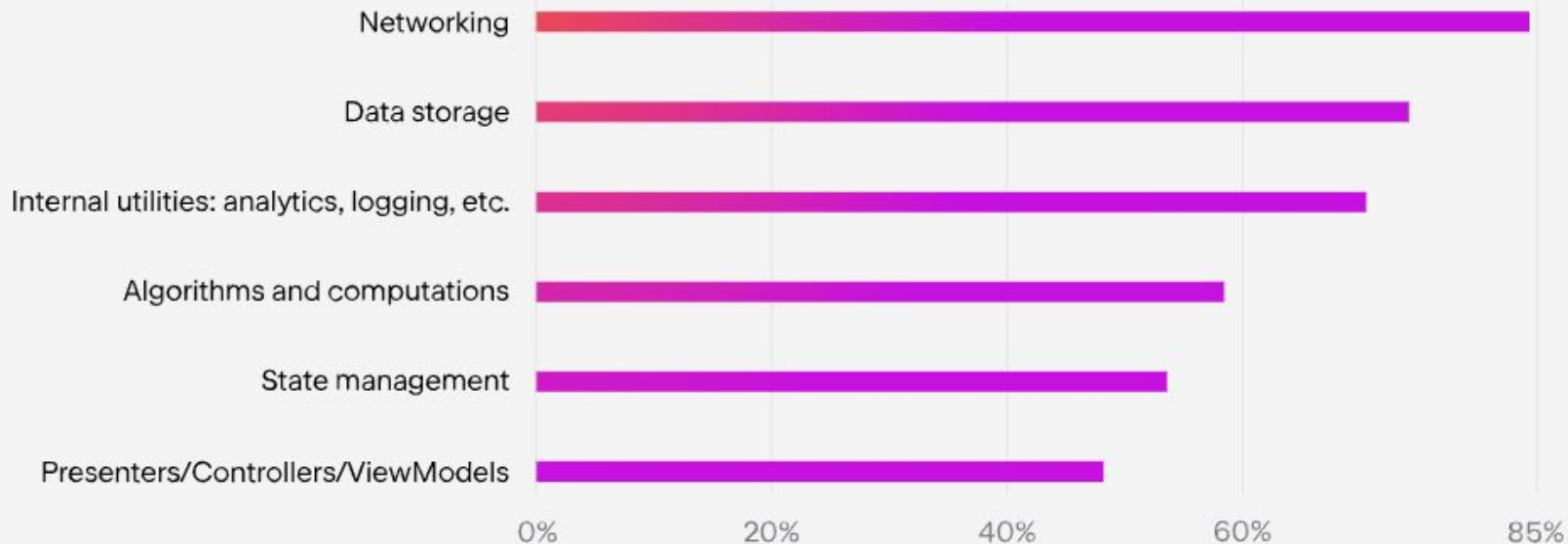
- Mobilní SDKčko
 - dnes v Betě
- Sdílení business logiky aplikace
- UI se píše nativně (Compose + SwiftUI)
 - UX uživatele stejná jako v případě nativního vývoje
- V současnosti používají KMM i velké firmy jako
 - Netflix
 - VMWare
 - etc...
- [Samples GIT](#)

KMM - Sdílení logiky



KMM - které části kódu nejčastěji?

What parts of your code were you able to share between platforms?



KMM – Výhody a Nevýhody

Výhody

- **Sdílení business logiky**, tzn. menší prostor pro chyby (menší duplicita kódu)
- UX uživatele stejné jako při nativním vývoji
- **Code-Review** napříč týmy
- **Menší vývojová kapacita ==** menší nárok na rozpočet projektu
- Lze zavést postupně pomocí **modularizace** projektu

Nevýhody

- Stále jsou zapotřebí platform specific vývojáři
 - **Ale tomu se u cross-platform vývoje často nevyhneme**
- KMM třídy
 - Je zapotřebí používat specifické třídy, které podporují KMM
- Stále beta

Děkuji za pozornost 😊