



SEMESTRÁLNÍ PRÁCE Z PŘEDMĚTU KIV/MBKZ

# MULTIPLATFORMNÍ APLIKACE ZOBRAZUJÍCÍ PŘEDPOVĚĎ POČASÍ

STANISLAV KRÁL

A17B0260P

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA APLIKOVANÝCH VĚD

# Obsah

<b>1</b>	<b>Analýza</b>	<b>2</b>
1.1	Vývoj multiplatformních aplikací pomocí frameworku Flutter .	2
<b>2</b>	<b>Závěr</b>	<b>4</b>

# 1 Analýza

## 1.1 Vývoj multiplatformních aplikací pomocí frameworku Flutter

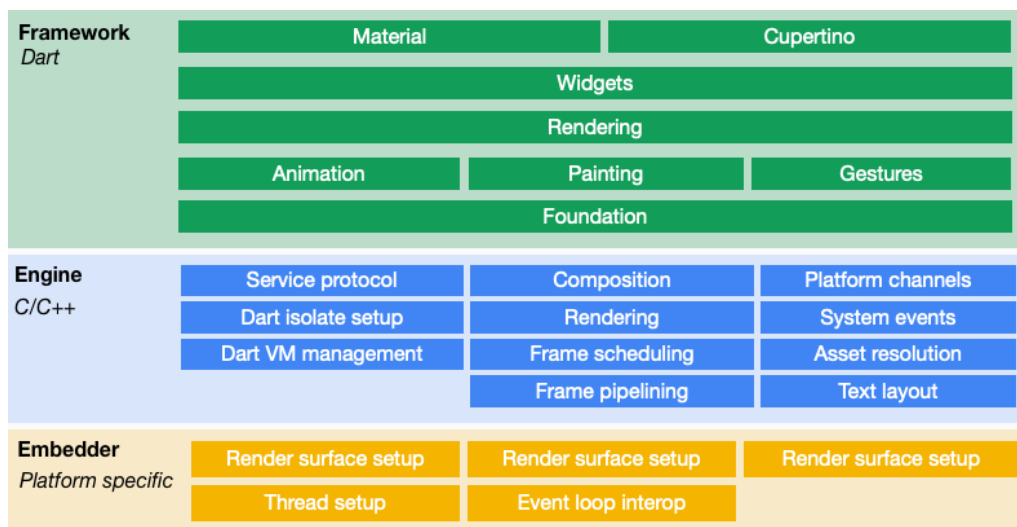
Flutter je open-source framework na jehož vývoji se podílí převážně firma Google. Cílem Flutteru je nabídnout vývojářům možnost vyvíjet výkonné aplikace, které působí nativně na všech platformách. Flutter umožňuje sdílet veškerý kód mezi všemi platformami. Toho je docíleno tak, že knihovna nepoužívá žádné nativní komponenty uživatelského rozhraní dané platformy. Okno aplikace totiž pouze slouží jako plátno a všechny komponenty si Flutter vykresluje sám. K vykreslování používá open-source knihovnu Skia, která je napsaná v jazyce C++, čímž dosahuje plynulosti ve zobrazování uživatelského rozhraní. Pro psaní Flutter aplikací se používá programovací jazyk Dart.

Flutter respektuje rozdíly v chování uživatelských rozhraní mezi mobilními operačními systémy čímž pomáhá aplikacím psaným v tomto frameworku působit tak, jako kdyby byly nativní a cílené právě pro danou platformu. Mezi hlavní rozdíly mezi platformami, jež tato knihovna implementuje na jednotlivých platformách zvláště, patří například **scrollování**, ikony nebo typografie.

Při nasazení na Android platformu je C a C++ kód enginu je kompilován pomocí Android NDK, zatímco Dart kód se s využitím kompilace typu ahead-of-time kompiluje do nativních ARM a x86 knihoven. Tyto knihovny jsou poté použity ve vygenerované nativní Android aplikaci, která slouží jako hostitel, a z celého projektu je následně vytvořen APK balík. Při vyvíjení Flutter aplikace je použit virtuální stroj, který umožňuje upravovat zdrojový kód aplikace bez nutnosti ji restartovat. Tato funkcionality má největší význam při tvorbě uživatelského rozhraní, kdy se jednotlivé změny v designu aplikace ihned projeví na zařízení. Podobně probíhá i nasazení aplikace na iOS platformu, kdy kód enginu je překládán pomocí LLVM. Kompilace do nativních knihoven znamená, že narozdíl od ostatních frameworků pro tvorbu multiplatformních aplikací, jsou Flutter aplikace zcela nativní.

Prvky uživatelského rozhraní, které Flutter nabízí, se snaží co nejvěrněji napodobit ty nativní. Prvky napodobující prvky platformy Android se nachází v balíku **Material** a prvky platformy iOS v balíku **Cupertino**. Tým vývojářů Flutteru bere ohledy na aktualizace mobilních operačních systémů a změny v uživatelských rozhraní včas implementuje do svého frameworku.

Flutter dále umožňuje psát nativní kód specifický pro danou platformu pomocí konstrukce zvané **platform channel**, která funguje na principu asynchronního předávání zpráv. Část Flutter aplikace pošle hostitelské nativní



Obrázek 1: Diagram architektury Flutter

aplikaci zprávu, která asynchronně na tuto zprávu odpoví. Toto umožňuje přístup k nativnímu API dané platformy.

Na konci roku 2019, v každoročním shrnutí služby GitHub, je programovací jazyk Dart, který se v dnešní době používá nejvíce právě ve Flutter aplikacích, označen jako jazyk s nejrychleji rostoucím počtem vývojářů, jež ho používají pro vývoj aplikací [?].

## 2 Závěr

V rámci této semestrální práce byl vytvořena dvojice programů - server a klient napodující deskovou hru Kris Kros. Důkladný a pečlivý návrh protokolu se osvědčil a v průběhu vývoje neprošel žádnými změnami. Avšak co se několikrát měnilo, byl způsob čtení zpráv, kdy nakonec jsou zprávy čteny spolehlivým automatem. Pokud bych automat navrhl hned na začátku, mohl jsem si ušetřit spoustu času laděním a upravováním parseru zpráv. Aplikace i server jsou však nyní funkční a hra je hratelná. Pokud je spojení s některým hráčem dočasně nedostupné nebo je klient násilně vypnut, tak po autorizaci je hráčovo stav korektně obnoven.