



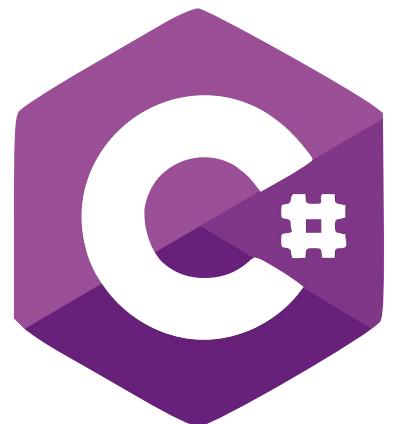
# .NET Core

LUKÁŠ VAVREK



**LUKÁŠ VAVREK**  
Software Engineer

GlobalLogic



## C #

Backend web development,  
Cross-platform mobile app development



## Swift

iOS mobile app development



## Architecture

Microservices  
Serverless



**LUKÁŠ VAVREK**

Software Engineer

GlobalLogic

# Course Outline

.NET Platform

Toolset

C# Language

---

Web API

Design Patterns

Databases

Authentication

---

Docker

Modular Architectures

# Communication



gl-dotnetcore.slack.com



# Before we start

**How many of you  
know some  
programming  
language?**

# Before we start

**How many of you have  
some experience with  
OOP language?**

# Introduction

.NET Framework

.NET Core

.NET Standard

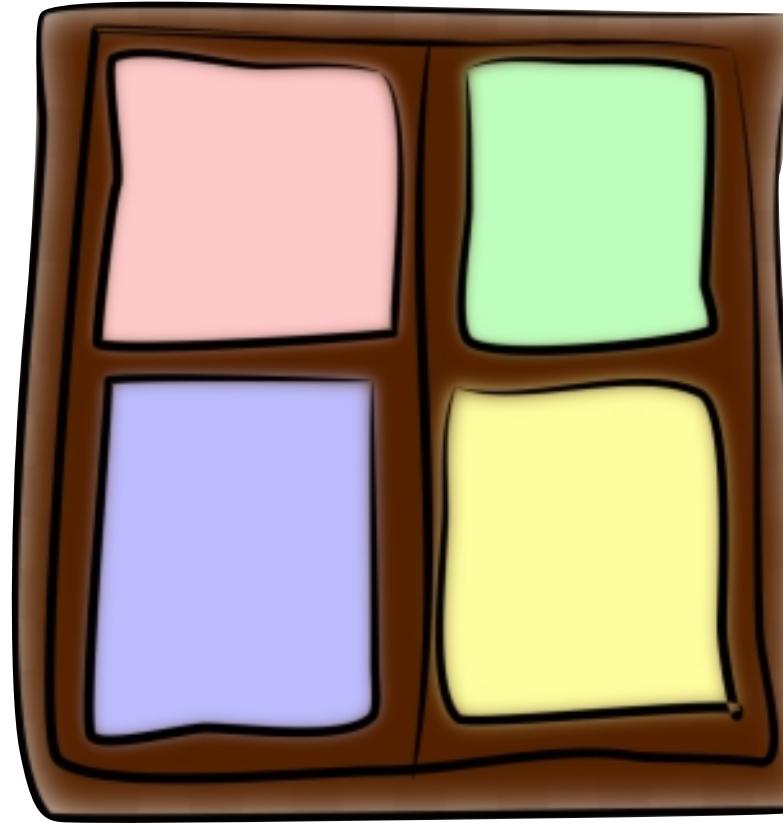
# .NET Framework



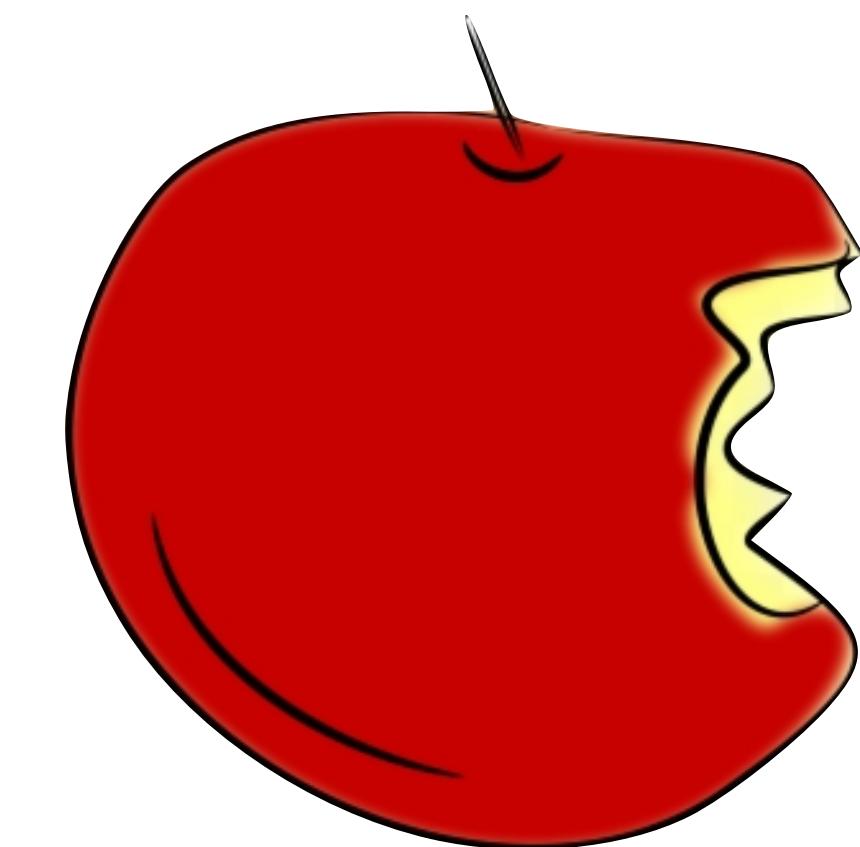
- Software Development Framework
- Framework Class Library (FCL)
- Common Language Runtime (CLR)

# Microsoft

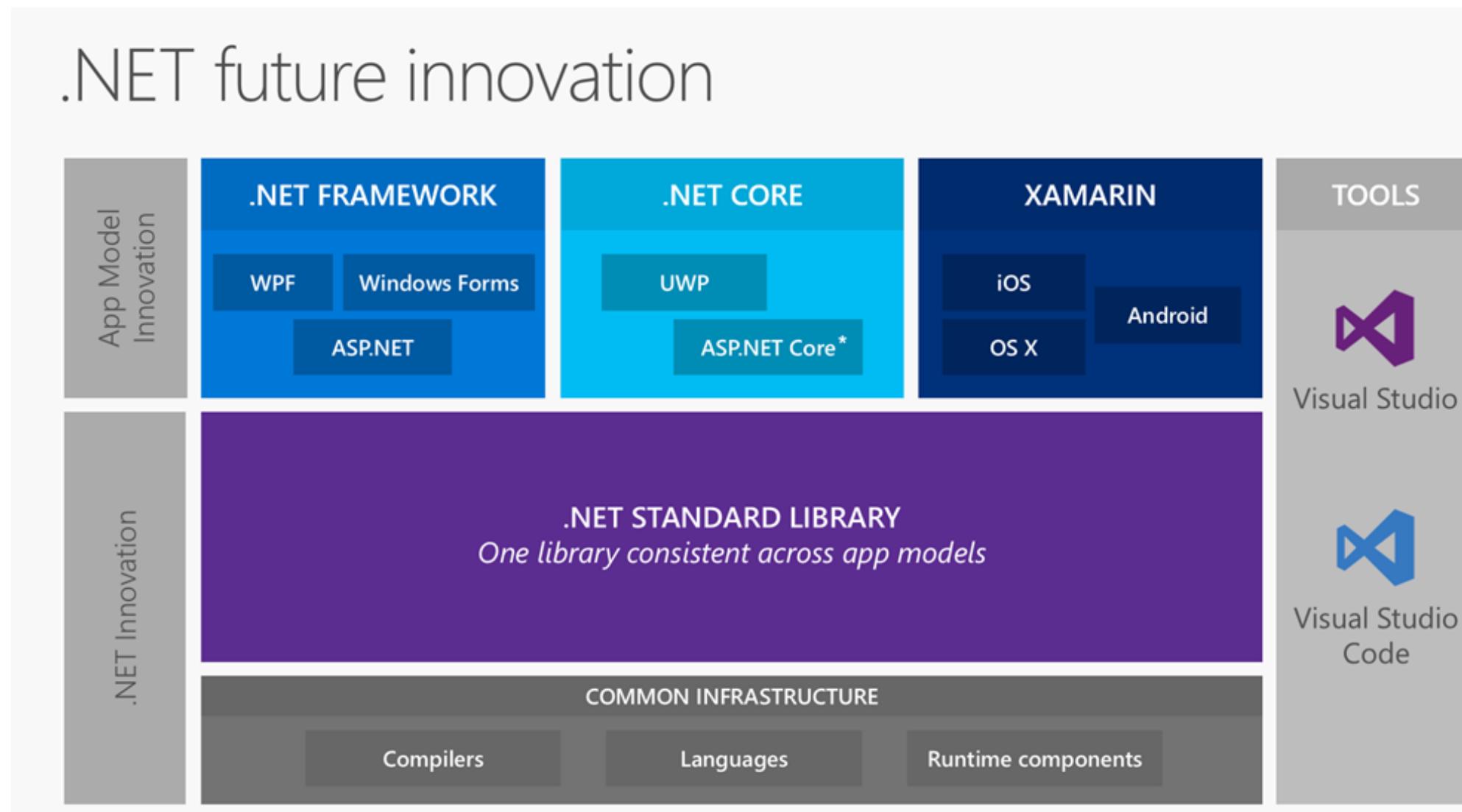
# .NET Core



- OSS Software framework
- Cross-Platform
- Fast and Modern



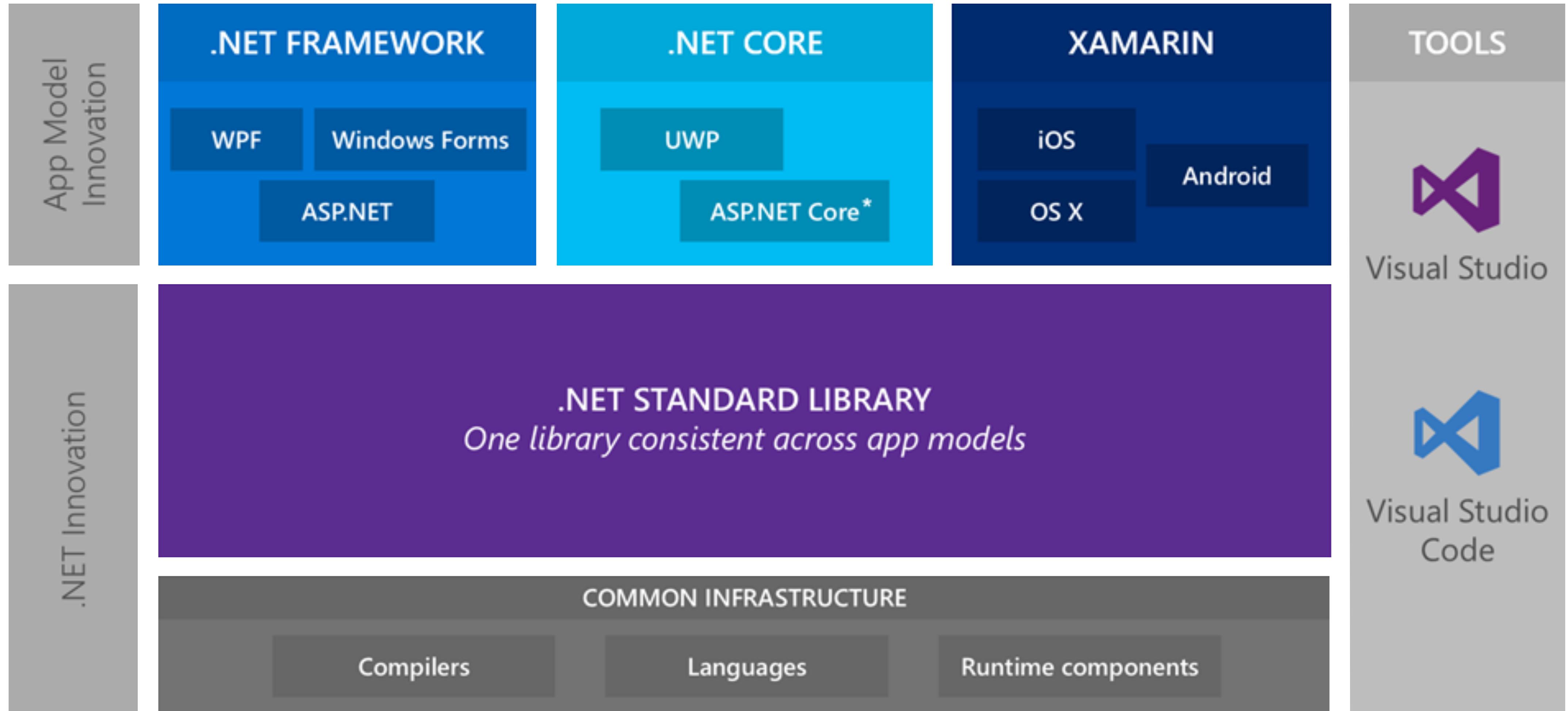
# .NET Standard



- Formal Specification
- Establish Uniformity
- Portable Library Development

<https://github.com/dotnet/standard>

# .NET Standard



# Microsoft ❤️ Open Source



<https://github.com/Microsoft/dotnet>

C#



The image shows a person working on a laptop, with a large semi-transparent overlay displaying C# code. The code is related to a 'PlaybackOverlayActivity' and includes methods for starting and stopping polling for updates. The code editor interface is visible, showing tabs for 'MainActivity.java', 'PostTV-Android [InstallAndroidTVDebug]', and 'PlaybackOverlayActivity'. The background shows a blurred view of the person's hands, a smartphone connected via USB, and a coffee cup.

```
public void startPollingForUpdates(@NotNull String uid) {
    Timber.d("startPollingForUpdates(), with this uid: %s", uid);
    long interval = 1000 * 60; // check every minute
    long start = System.currentTimeMillis() + interval;
    Intent i = LiveEventsService.getCheckLiveEventIntent(this, uid);
    pi = PendingIntent.getService(this, 0, i, 0);
    AlarmManager alarmManager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
    alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, start, interval, pi);
}

public void stopPollingForUpdates() {
    Timber.d("stopPollingForUpdates()");
    if (pi != null) {
        AlarmManager alarmManager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
        alarmManager.cancel(pi);
    }
}
```

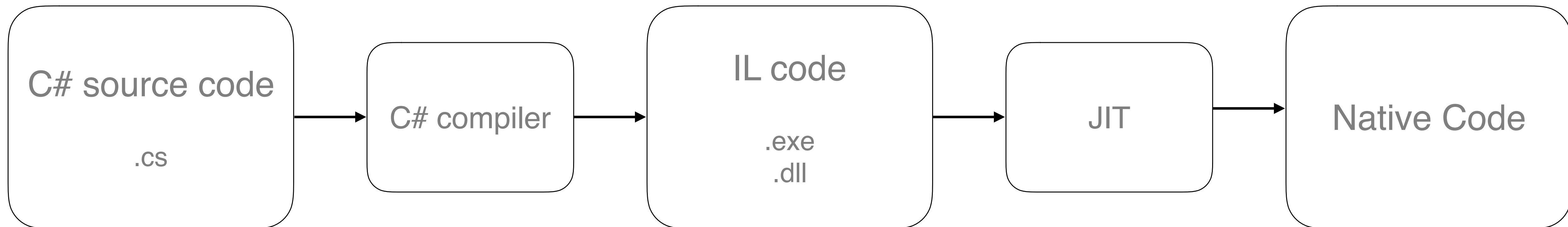
# C# Language

C# is a simple, modern, object-oriented, and type-safe programming language. It includes support for component-oriented and functional programming, automatic memory management using garbage collection and build in exception handling.

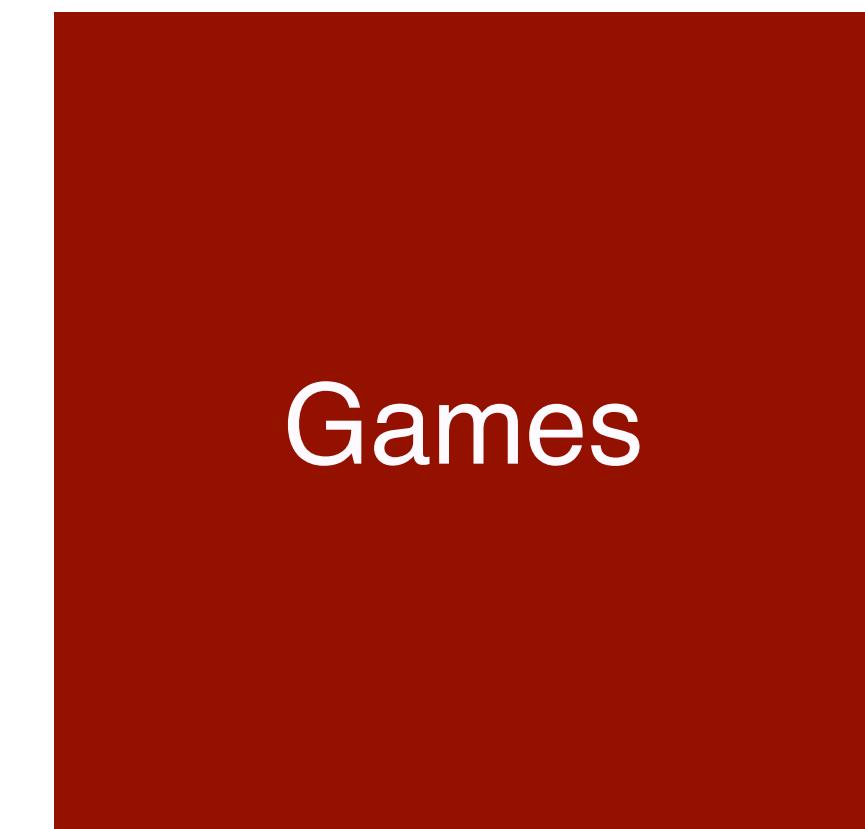
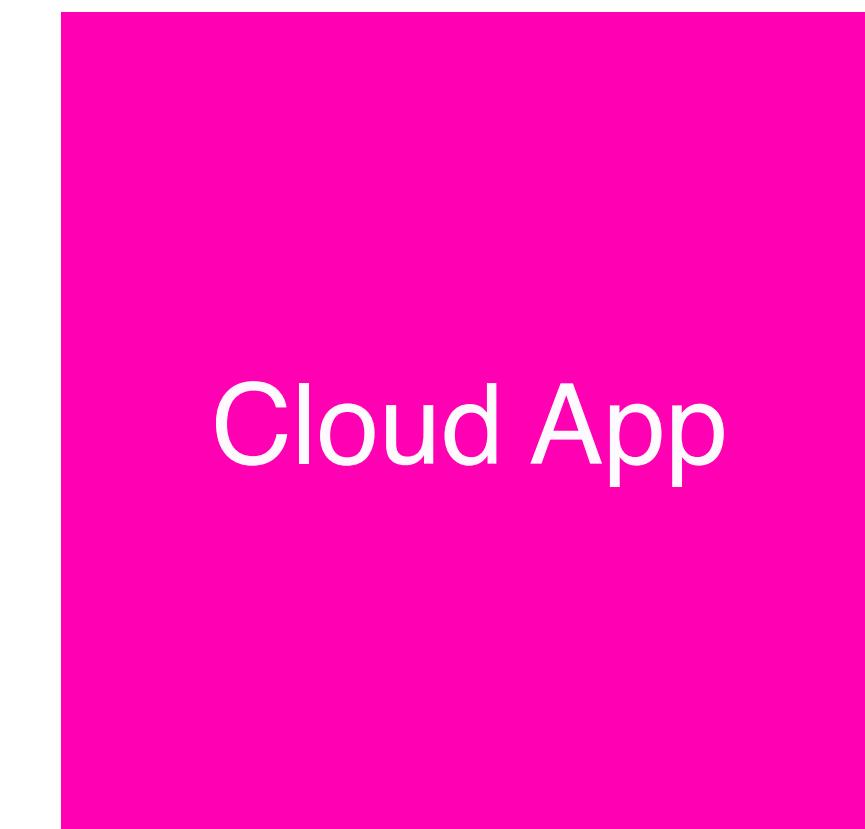
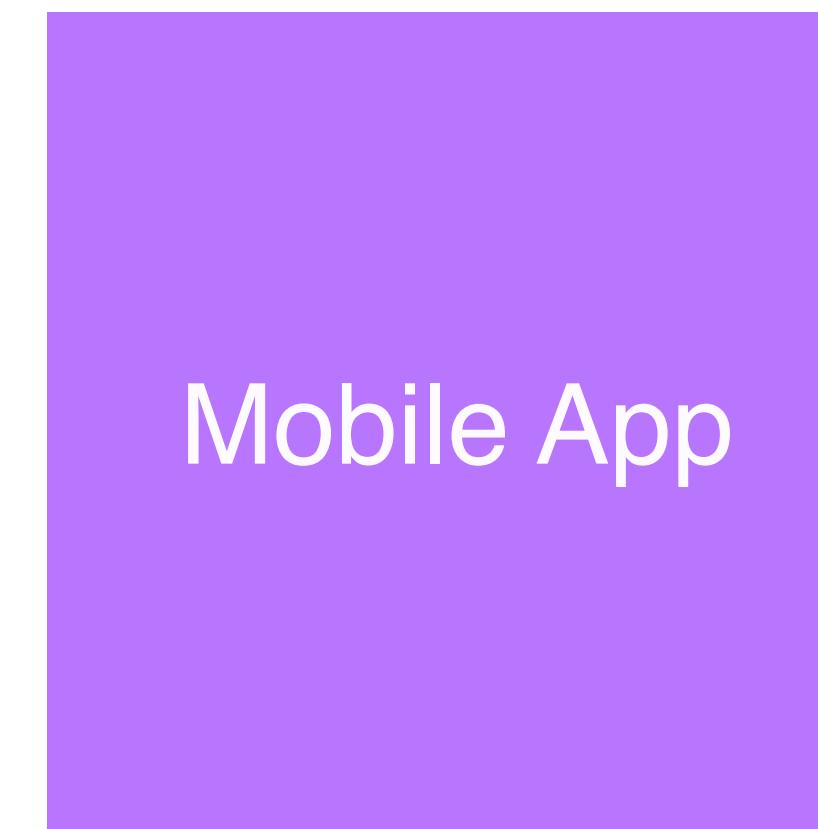
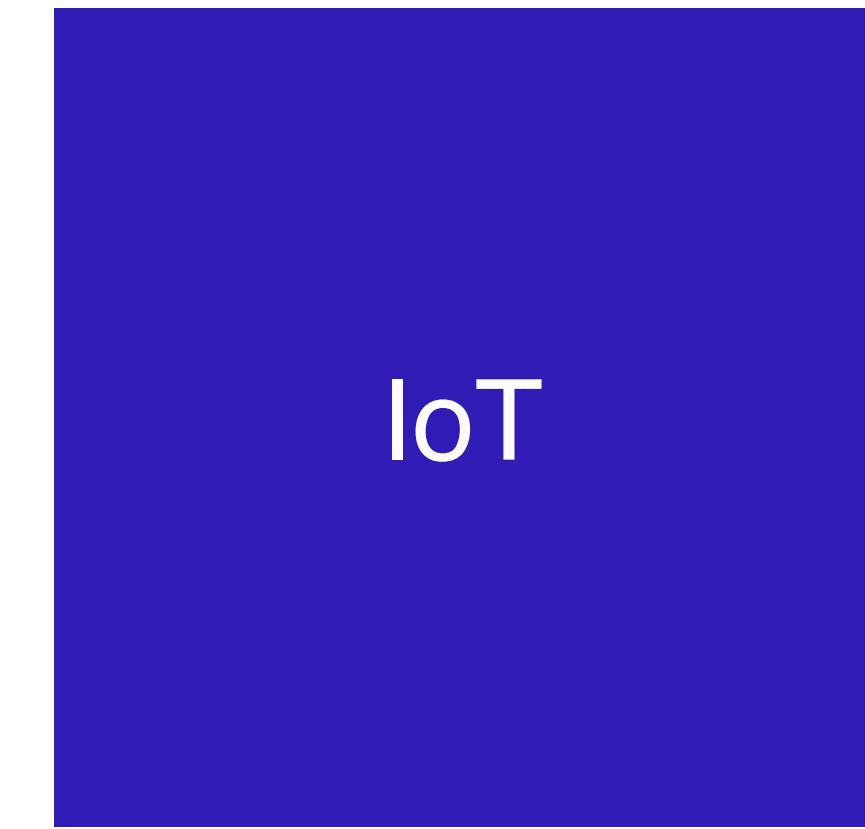
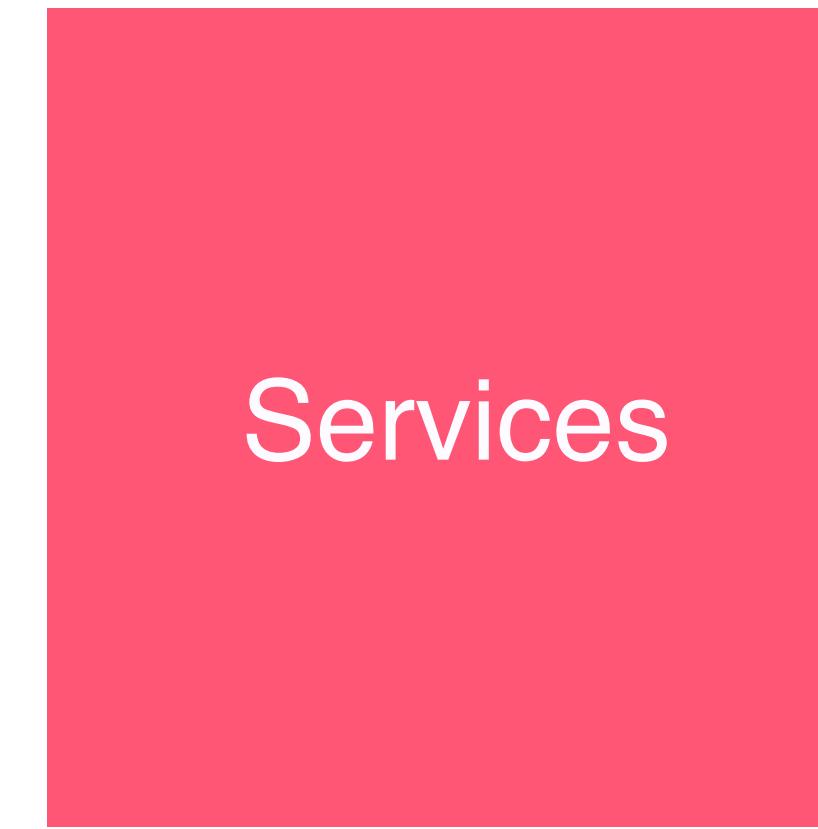
# Hello World

```
using System;
class Hello
{
    static void Main() {
        Console.WriteLine("Hello, World");
    }
}
```

# Compilation Process



# What can I build?



# DEMO

## Hello World Console Application



```
function hello_world() {
    echo "Hello, World!";
}

hello_world();
```



# Q & A

