



# Xamarin <3 Azure!

## Mobile Cross-Plattform-Apps mit Azure Backend

Rico Herlt

@rherlt | <http://RicoHerlt.com>

# Rico Herlt

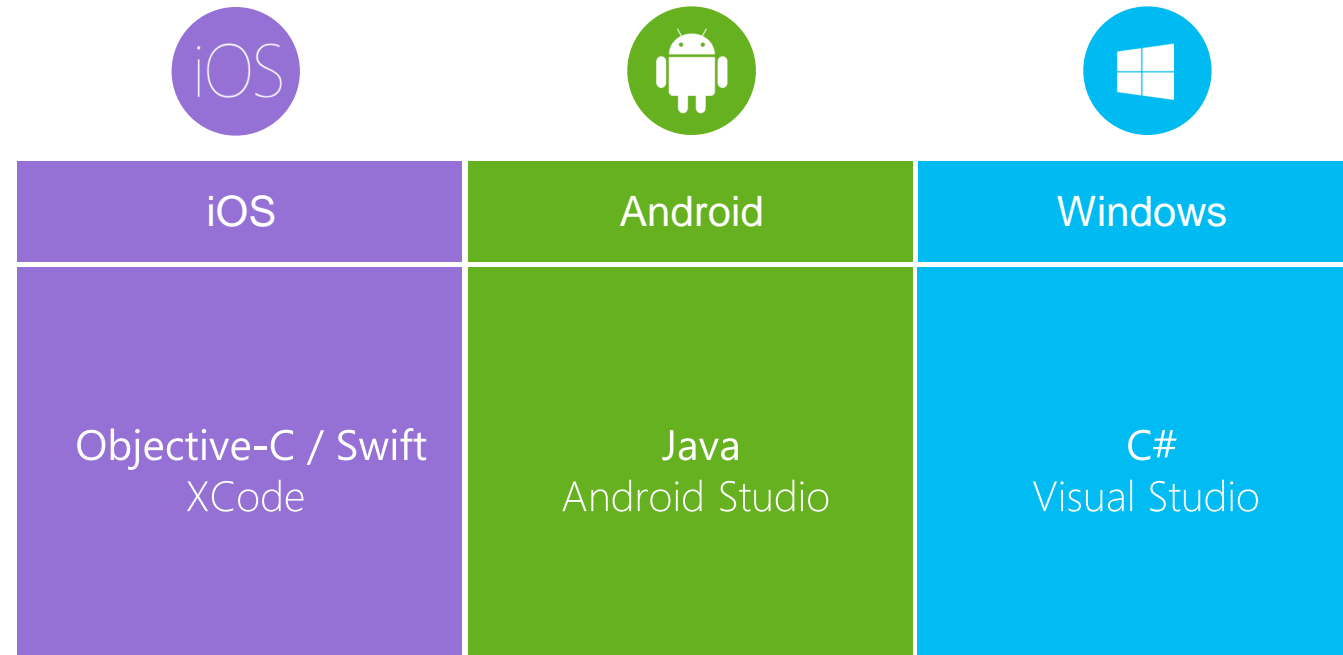
- Software-Consultant und –Entwickler
- Geschäftsführer der h3ko GmbH
- Cross-Plattform-Entwicklung mit Xamarin täglich seit fünf Jahren
- Entwicklung von Web Services und Anbindung an Apps
- Microsoft Student Partner (<http://StudentPartners.de/>)
- Veranstaltung von Programmierwettbewerben für Studenten
- Organisation von Cloud Meetups in Berlin

# Agenda

- Von der klassischen App zu Cross-Plattform
- Erhältliche Werkzeuge
- Xamarin Framework
- Code Sharing und Portable Class Libraries (PCLs)
- UI-Implementierung
- Xamarin.Forms
- Azure Cloud Services
- Xamarin <3 Azure!
- Demo Projekt - Droplt

# Mobile Apps entwerfen: bisher

Klassischer Silo-Ansatz



Keine Gemeinsame Code-Basis • Viele Sprachen und IDEs • Viele Teams

# Heute: Cross-Plattform-Apps

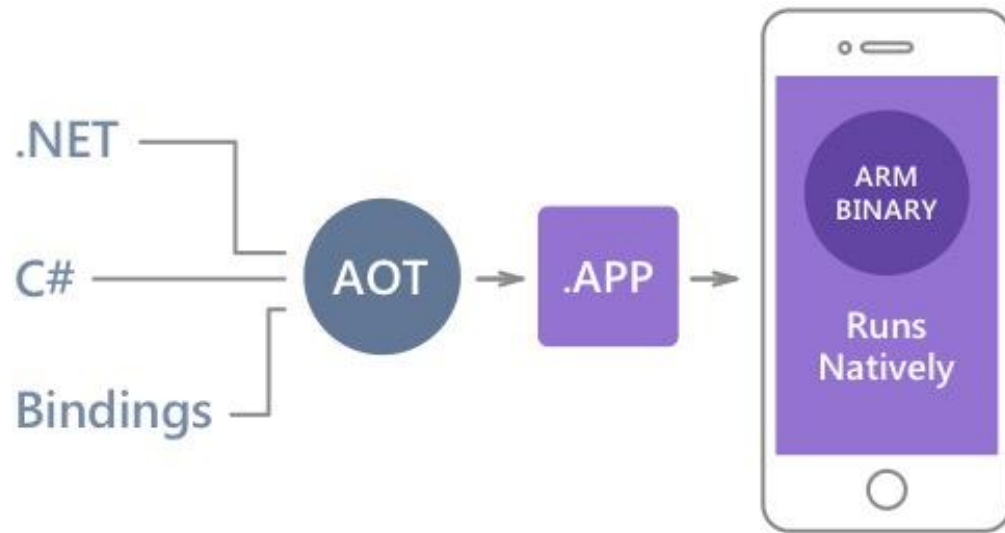
- 1 Code => Alle Plattformen
- Der Traum eines Entwicklers:
  - Teilen von Geschäftslogik und DTOs
  - Teilen von UI-Implementierung
  - Einheitlicher Zugriff auf plattformspezifische Hardware
  - Gemeinsame Programmiersprache für alle Plattformen
  - Einheitlich nutzbare Frameworks
  - ABER: Beibehalten von plattformspezifischem Look & Feel



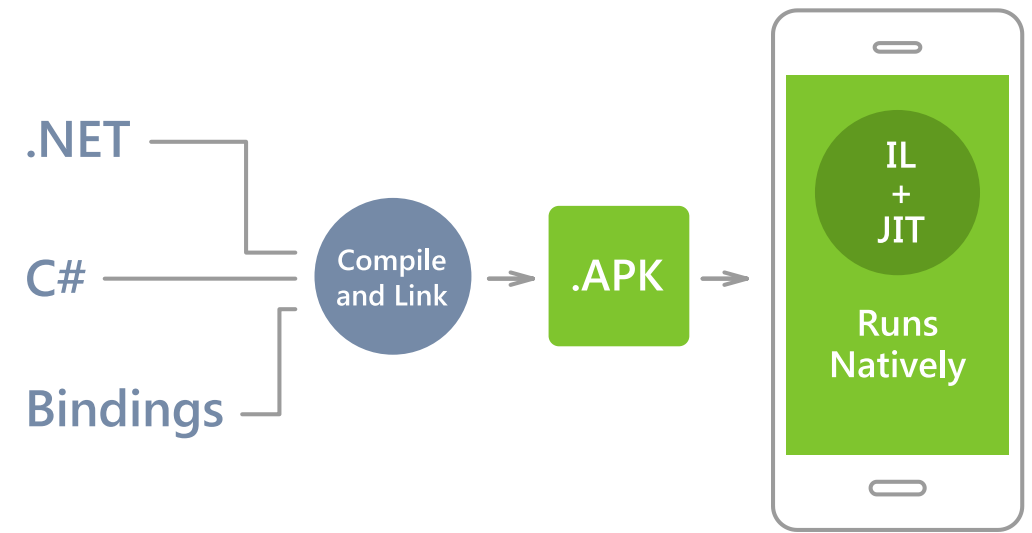
# Gut: Gemeinsamer Code bei hybriden Apps

- Durch Implementierungen in HTML5 und Java Script
- Anzeige des UIs, gerendert durch native Webview (Browser)
- ABER:
  - kein natives Benutzererlebnis
  - Oftmals Schwierigkeiten bei Hardwarezugriffen
  - Schlechte Performanz

# Besser: Native Performanz mit Xamarin



Xamarin.iOS führt vollständige Ahead Of Time (AOT) Kompilierung aus um ARM Binary für Apple's App Store zu erstellen



Xamarin.Android nutzt den Vorteil der Just In Time (JIT) Kompilierung auf Android Geräten.



# Erhältliche Werkzeuge



Visual Studio

<https://www.visualstudio.com/>  
KOSTENLOS als Community Edition



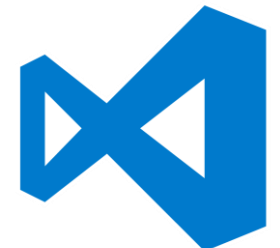
Xamarin Studio

<https://www.xamarin.com/>  
KOSTENLOS als Community Edition



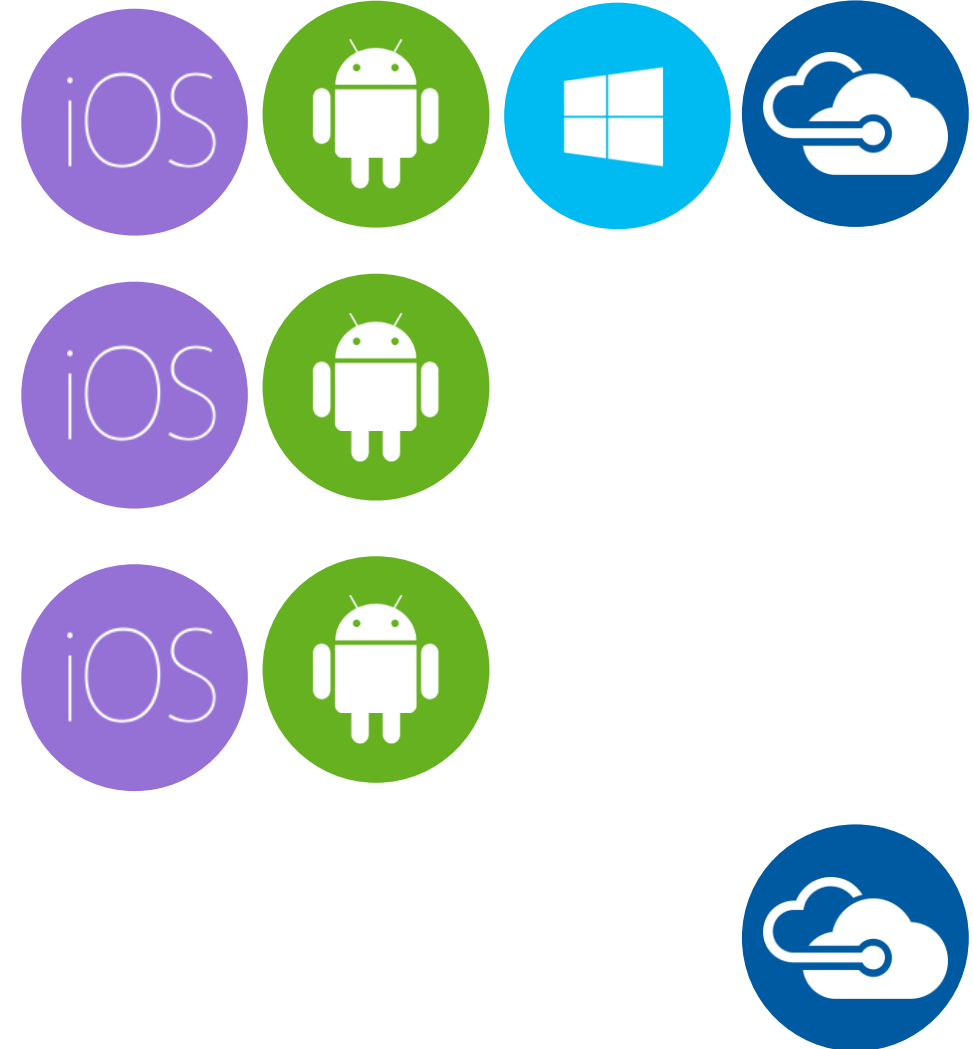
Visual Studio for Mac (Preview)

<https://www.visualstudio.com/de/vs/visual-studio-mac/>  
KOSTENLOS als Community Edition



VS Code

<https://code.visualstudio.com/>  
KOSTENLOS, Open Source





# Xamarin Framework



Native iOS App



Native Android App



Native Windows App

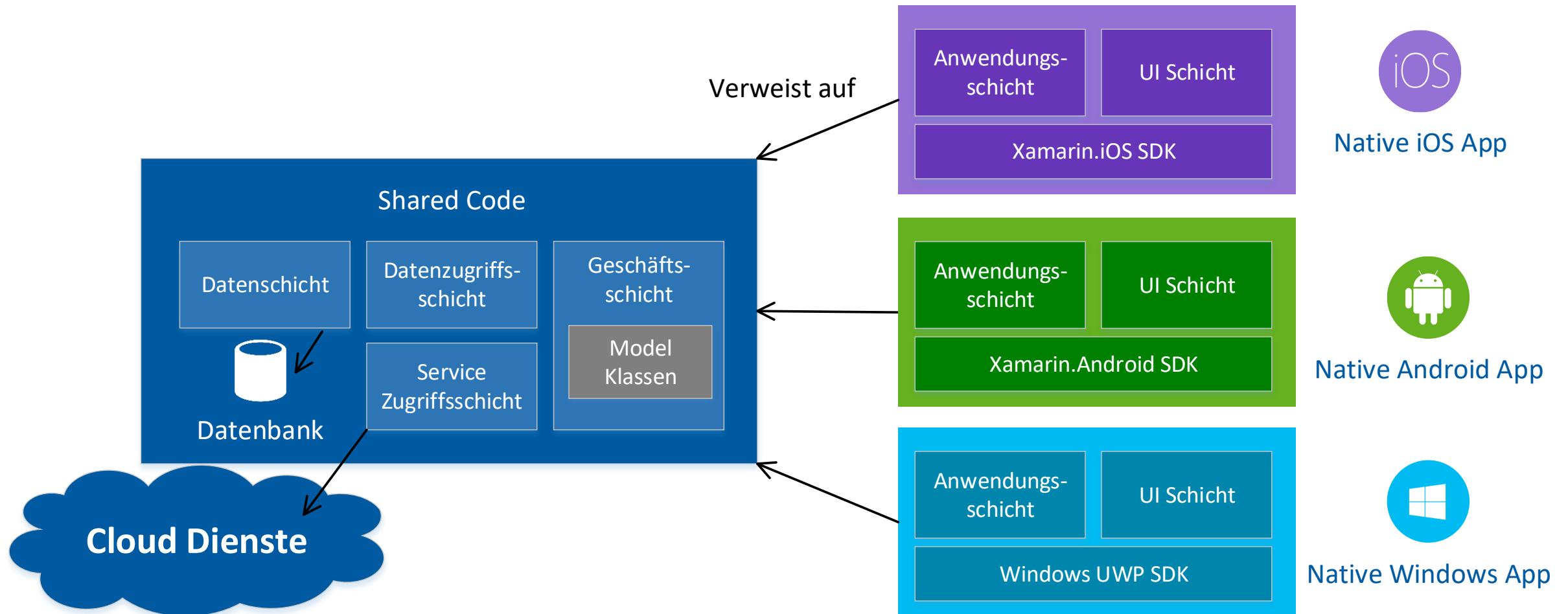
Plattformspezifischer C#  
Code

Plattformspezifischer C#  
Code

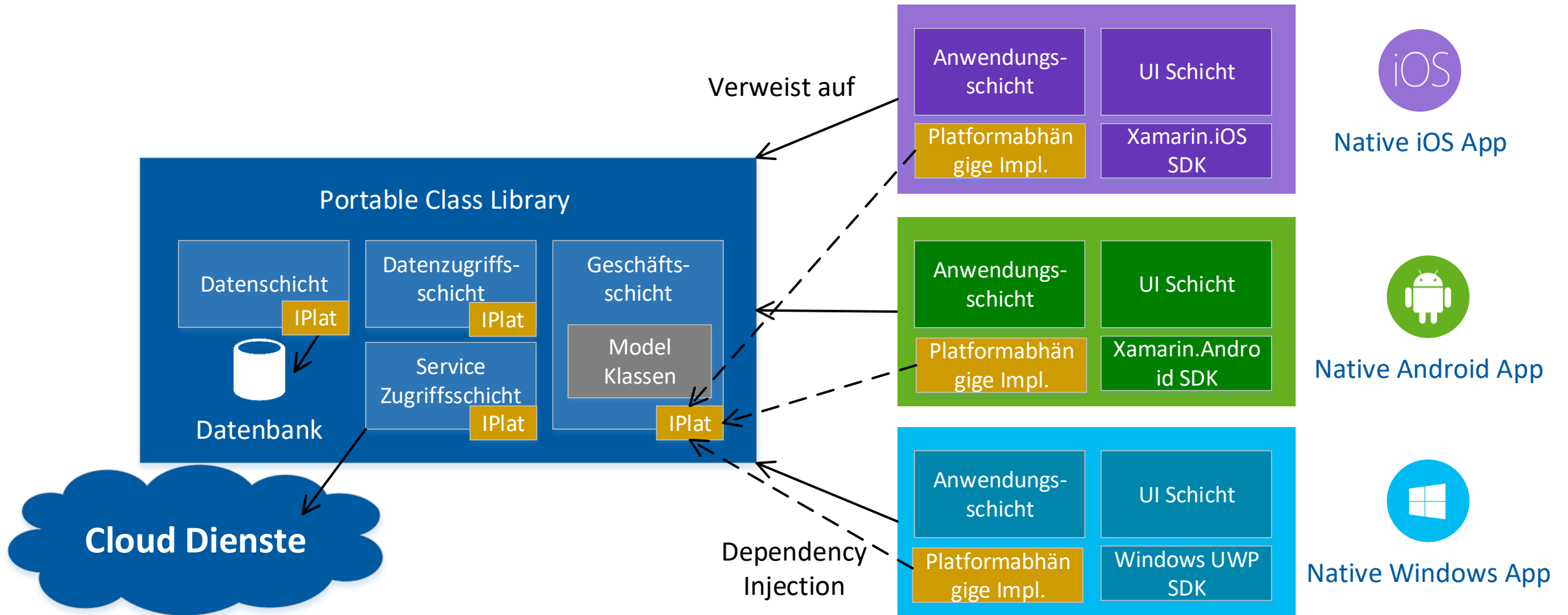
Plattformspezifischer C#  
Code

Geteilte C# Geschäftslogik

# Code Sharing



# Portable Class Libraries (PCLs)



# UI-Implementierung



## Xamarin.Forms Framework

- XAML generiert plattformabhängiges UI
- Custom Renderers
- Data Binding wie in WPF



## Plattformspezifisch

- Story Board bzw. XIB für iOS
- AXML für Android
- XAML für Windows



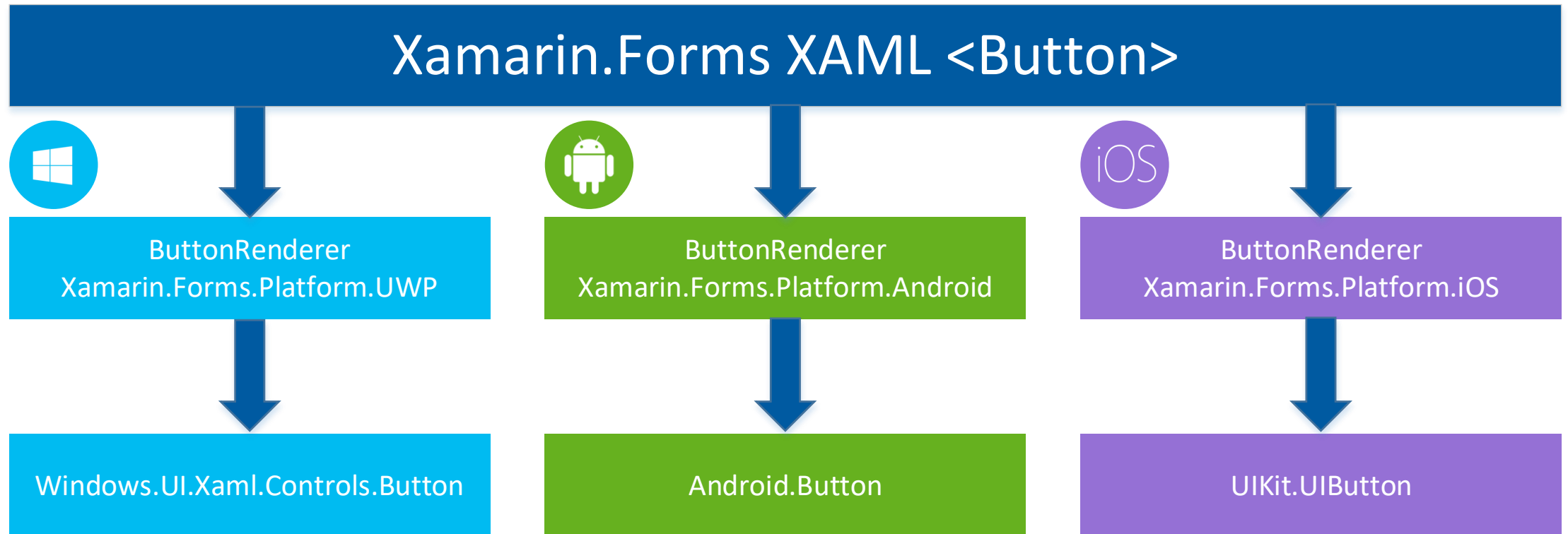
## Code

- Definition von nativen UI-Komponenten im Code
- Am wenigsten zu empfehlen

# Xamarin.Forms

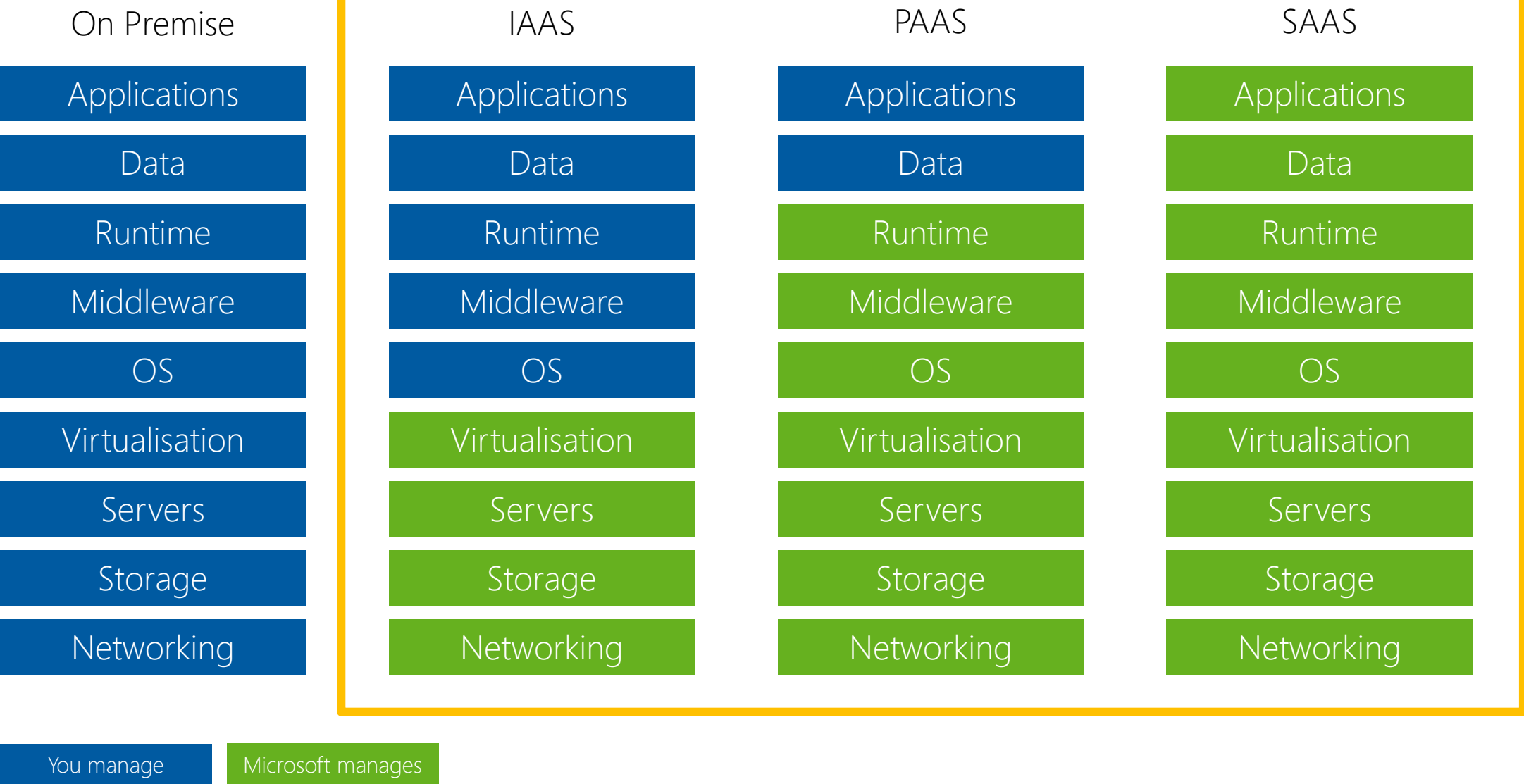
- Framework unterteilt sich in plattformunabhängigen und – spezifischen Teil
- UI Definition in XAML Syntax im plattformübergreifenden Core-Projekt
- Data Bindings verfügbar
- Generierung von nativen UI Komponenten zur Compile-Zeit oder zur Laufzeit
- Implementierungen für Standardkomponenten verfügbar
- Durch Custom Renderer vom Entwickler anpassbar / erweiterbar

# Xamarin.Forms



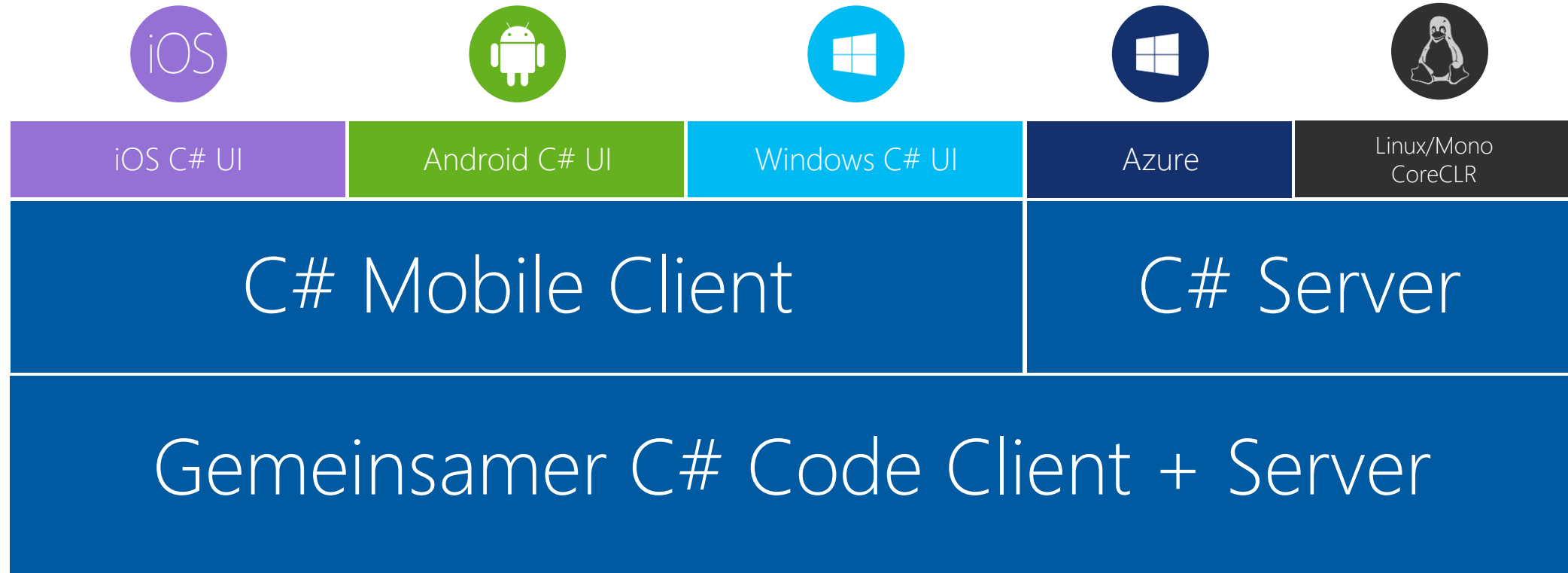
# Azure Cloud Services

Microsoft Azure



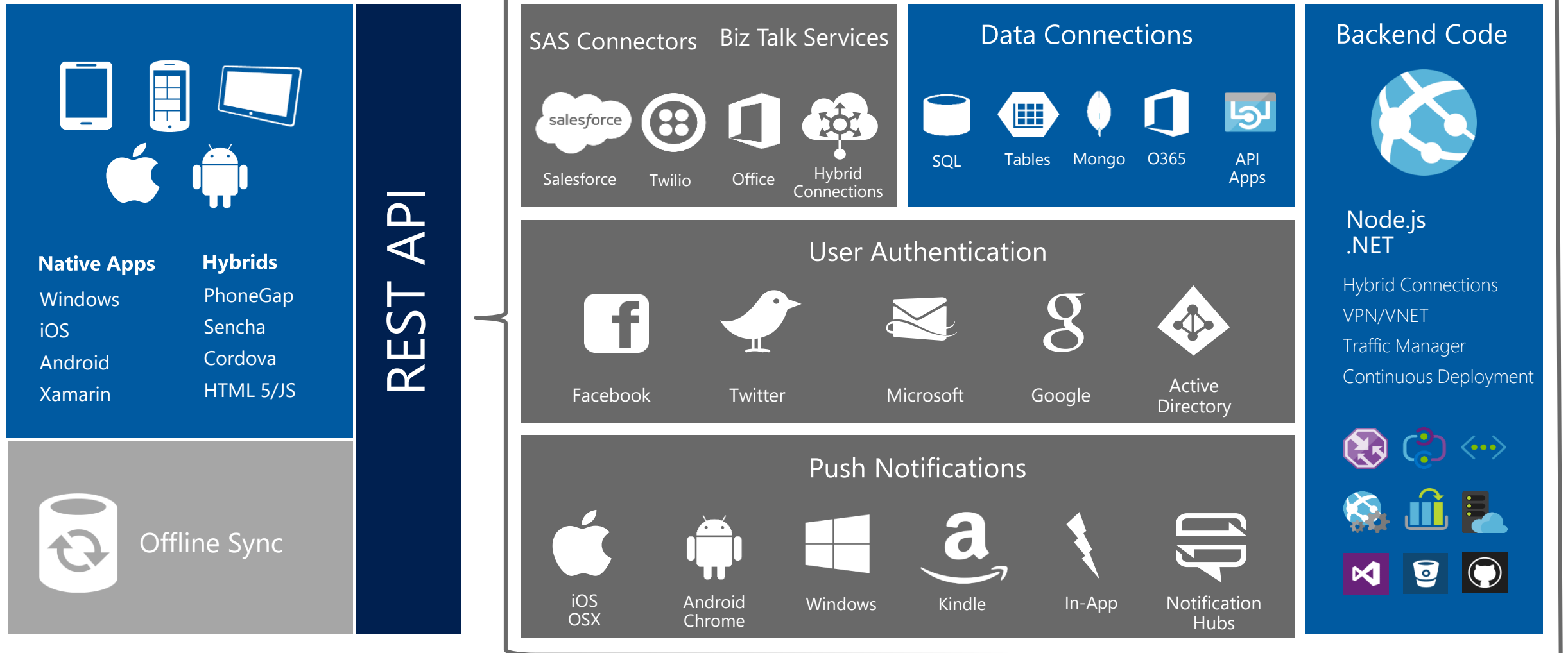


# Xamarin <3 Azure!



Gemeinsame C# Code-Basis • 100% nativer API Zugriff • Hohe Performanz

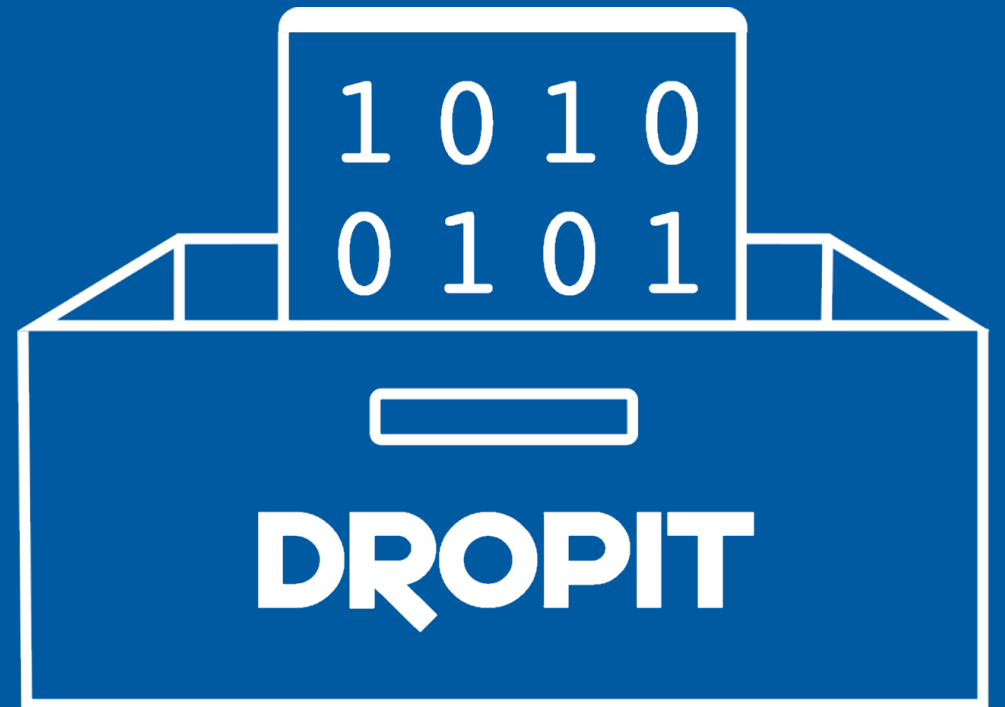
# Azure Mobile Apps



# Klassisches Beispiel



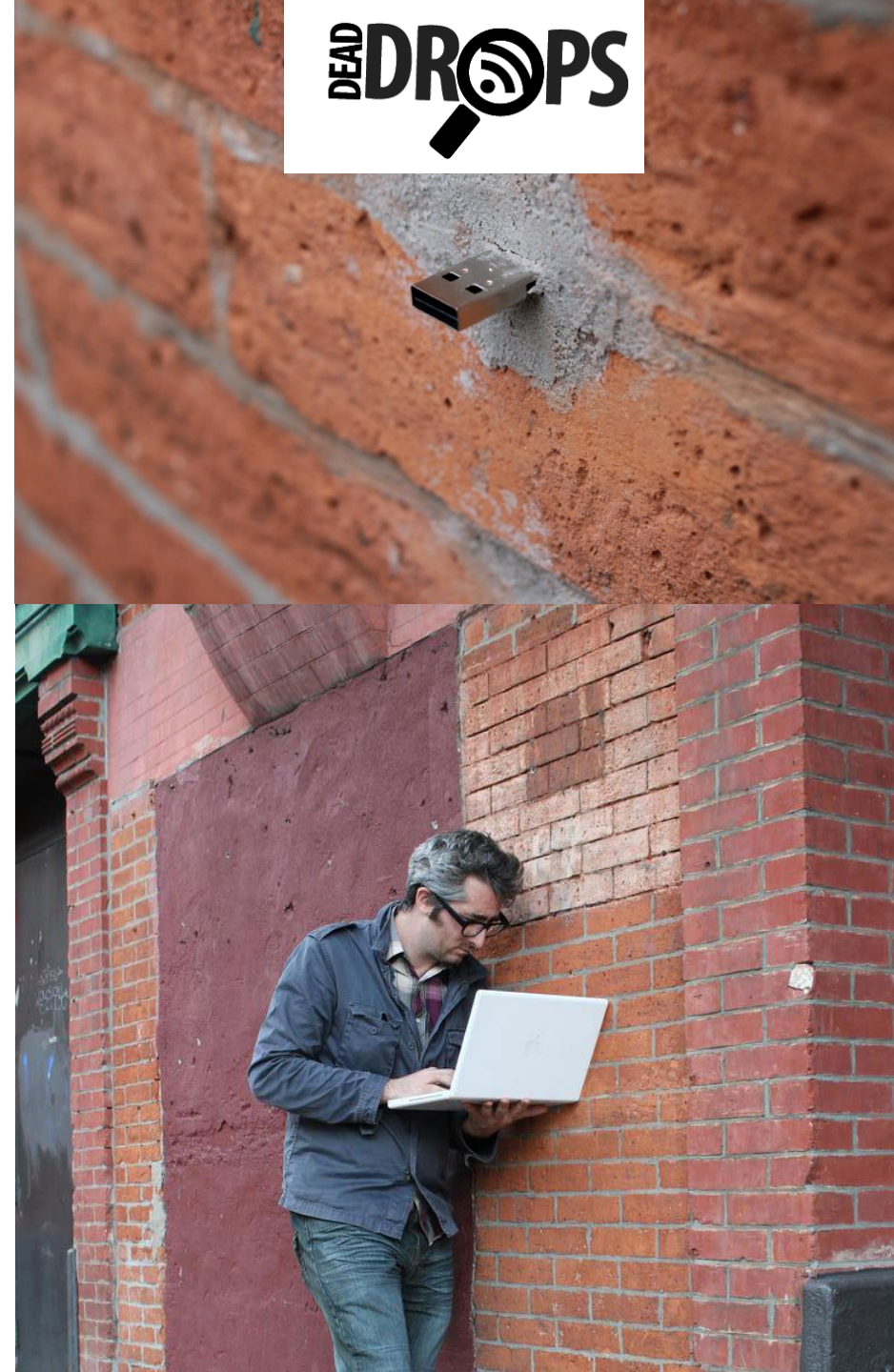
# Demo-App



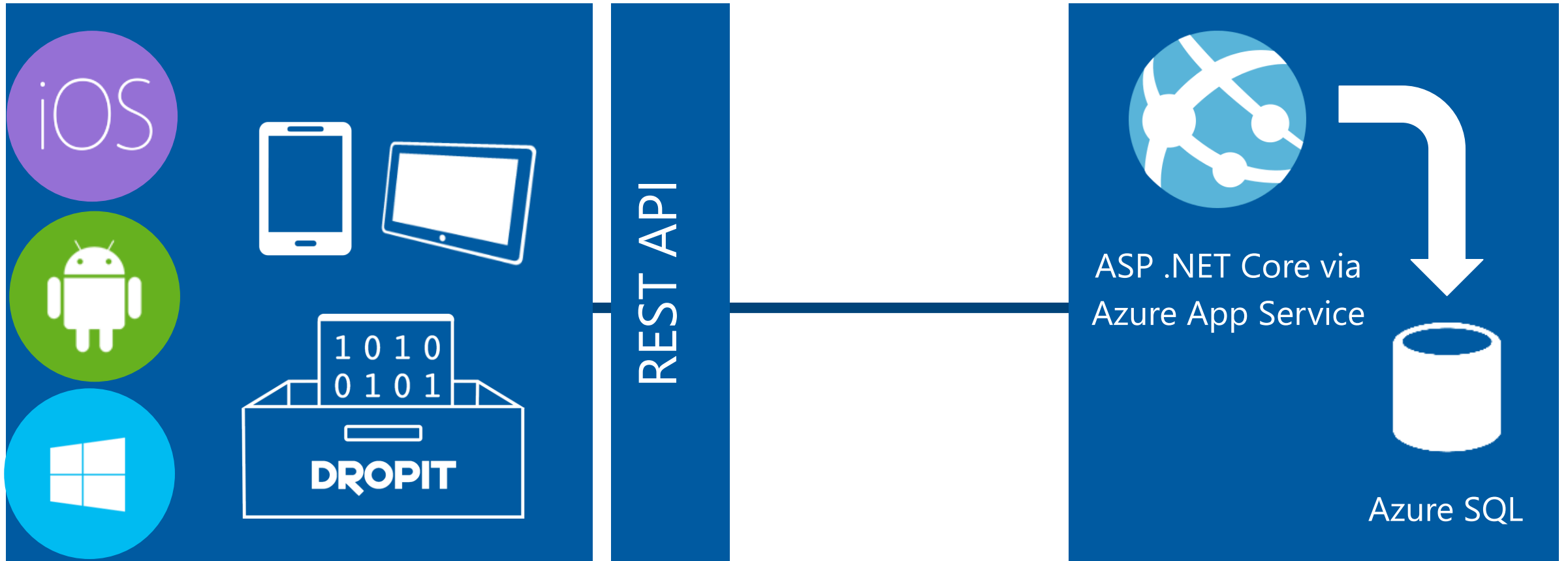
# DROPIT

- Anonym ortsbasierte Nachrichten hinterlassen
- Nachrichten können nur von Personen gelesen werden, die sich in der Nähe der Nachricht befinden
- In Anlehnung an das offline File Sharing Netzwerk „Dead Drops“

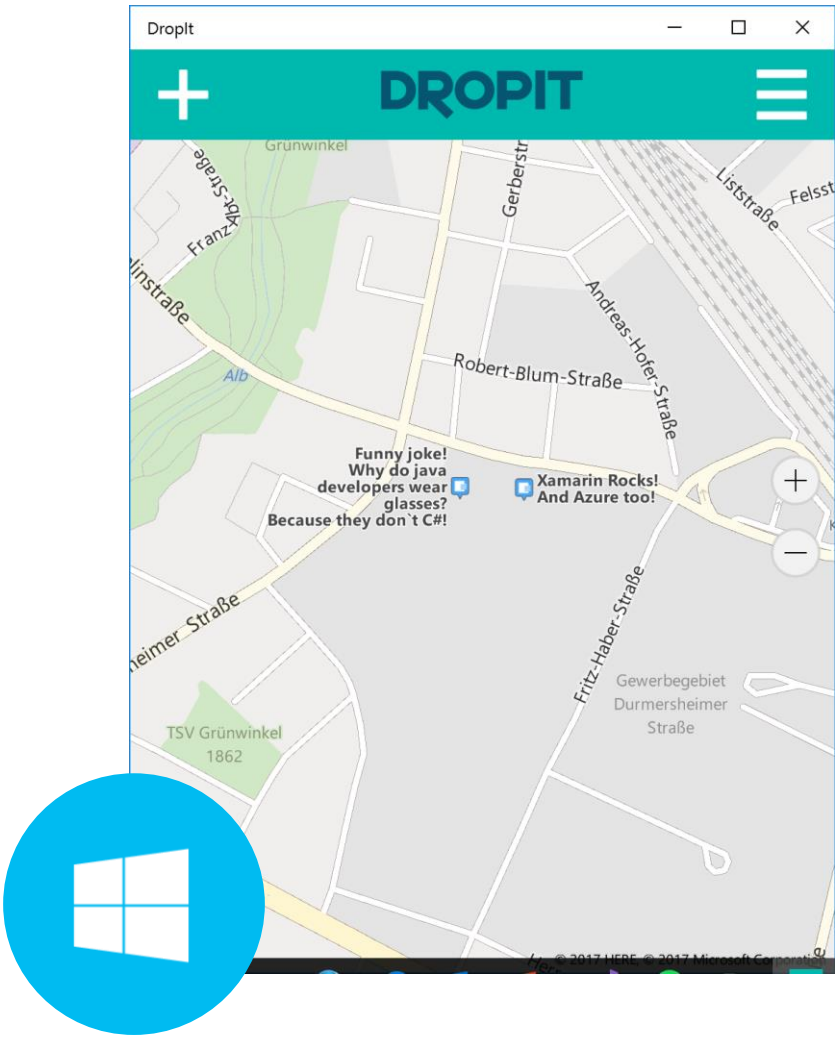
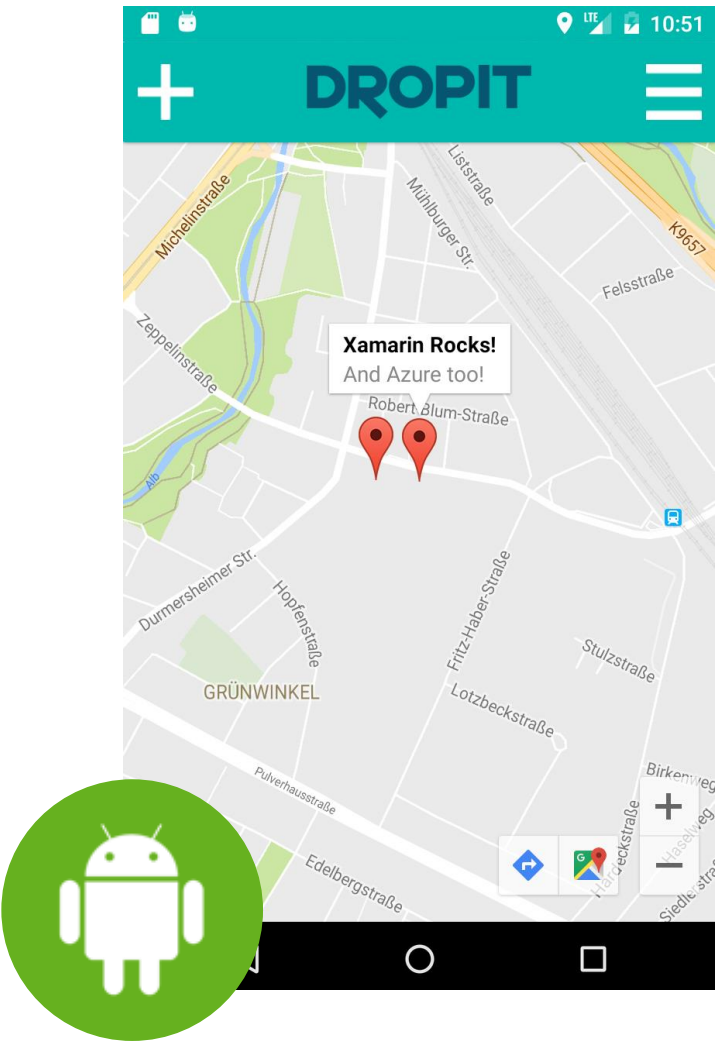
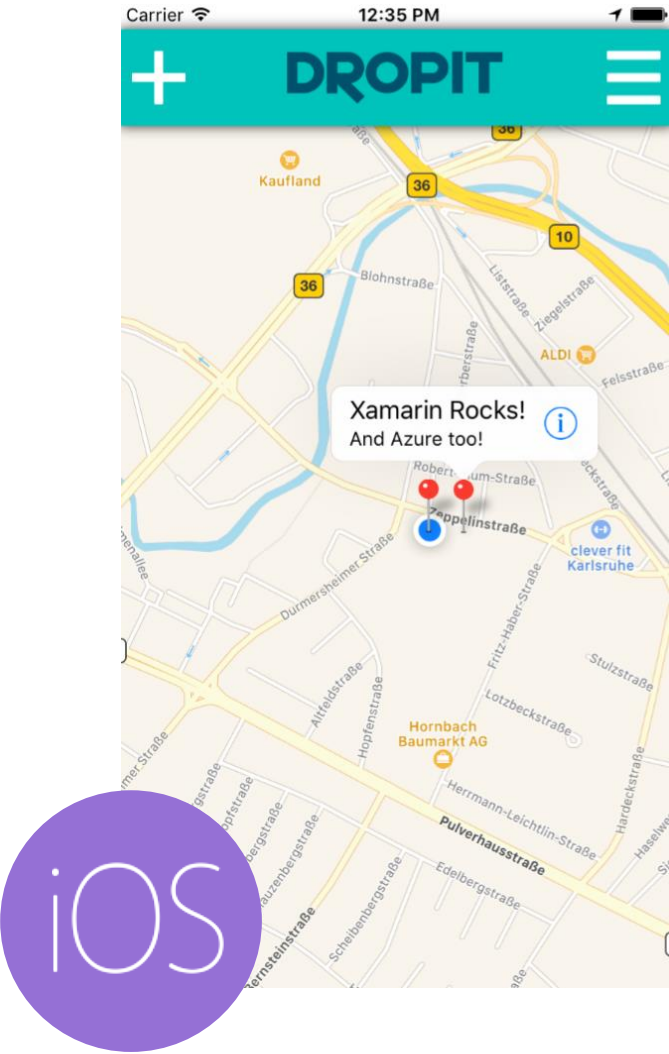
DEAD DROPS



# Technischer Aufbau



# Screenshots





# Screenshots

Carrier 12:38 PM

[Back](#)

**Add a new drop**

Drop title

Date:

Should this drop be always valid? ☐

This drop is valid until:

Drop your message here!

[Add a new Drop](#)



10:51

[Back](#)

**Add a new drop**

Drop title

Date:

Should this drop be always valid? ☐

This drop is valid until:

Drop your message here!

[Add a new Drop](#)



Droplt

**Add a new drop**

Drop title

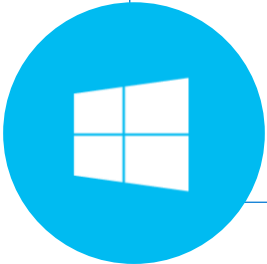
Date:

Should this drop be always valid? ☐ Off

This drop is valid until:

Place your Drop message here!

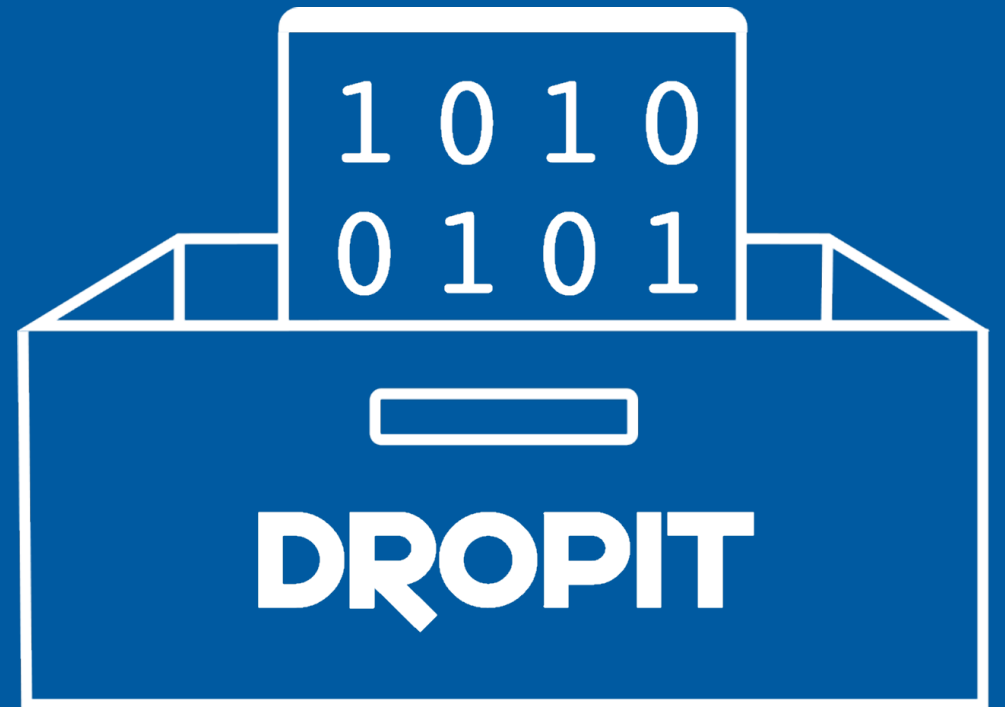
[Add a new Drop](#)



# Verwendete Frameworks (via NuGet)

- Xamarin.Forms
- Xamarin.Forms.Map
- Newtonsoft.Json (JSON Verarbeitung)
- Refit (HTTP REST Client)
- Splat (Dependency Injection)
- Xam.Plugin.GeoLocator (GPS Zugriff)
- Xam.Plugins.Notifier (App Notifications)
- Plugin.Permissions (App Rechte)
- KeyChain.Net (Sicheres Speichern von Credentials)
- PropertyChanged.Fody (Generierung von INotifyPropertyChanged-Implementierungen)

# Coding



# Links

## Demo Code Droplt

<https://github.com/rherlt/XamarinLovesAzure>

## Xamarin Workshop

<https://github.com/robinmanuelthiel/xamarinworkshop>

## Robin-Manuel Thiel

Xamarin Mobile Engineer (TSP)

Microsoft App Innovation

<https://twitter.com/robinmanuelt>

# Links

What is Azure App Service?

<http://docs.microsoft.com/en-us/azure/app-service/app-service-value-prop-what-is/>

Develop Cloud Connected Mobile Apps with Xamarin and Microsoft Azure

<http://adrianhall.github.io/develop-mobile-apps-with-csharp-and-azure/>

# Fragen?

## Vielen Dank für die Aufmerksamkeit!

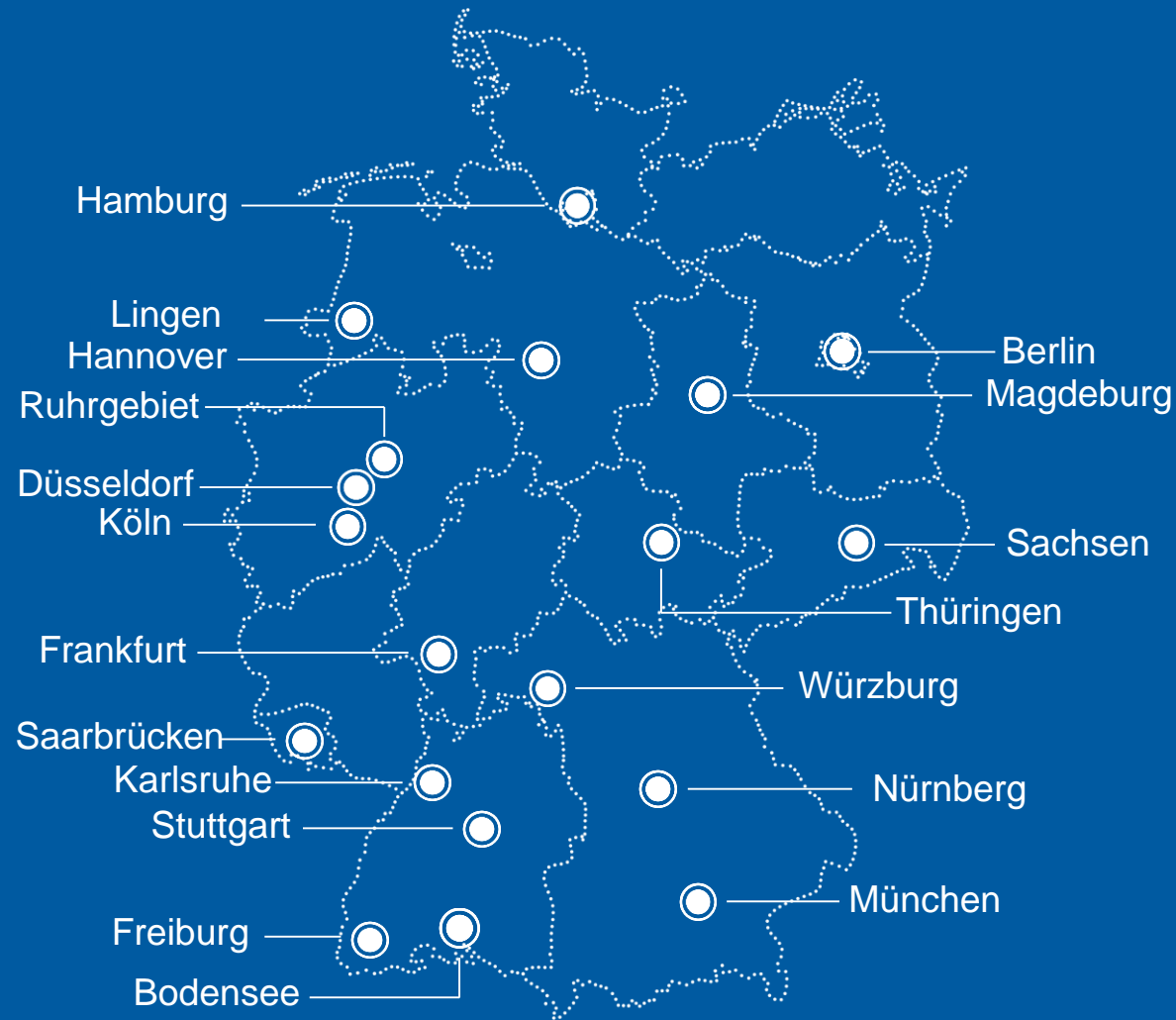


**Rico Herlt**

Software-Consultant und -Entwickler

@rherlt | <http://RicoHerlt.com>

# Microsoft Azure Meetups



[www.aka.ms/azure-meetups](http://www.aka.ms/azure-meetups)