



Dev Days

(X) Dev Days

#XamarinDevDays

Connected & Disconnected Apps with Azure Mobile Apps

Roy Cornelissen
@roycornelissen

Lead Consultant, Xpirit

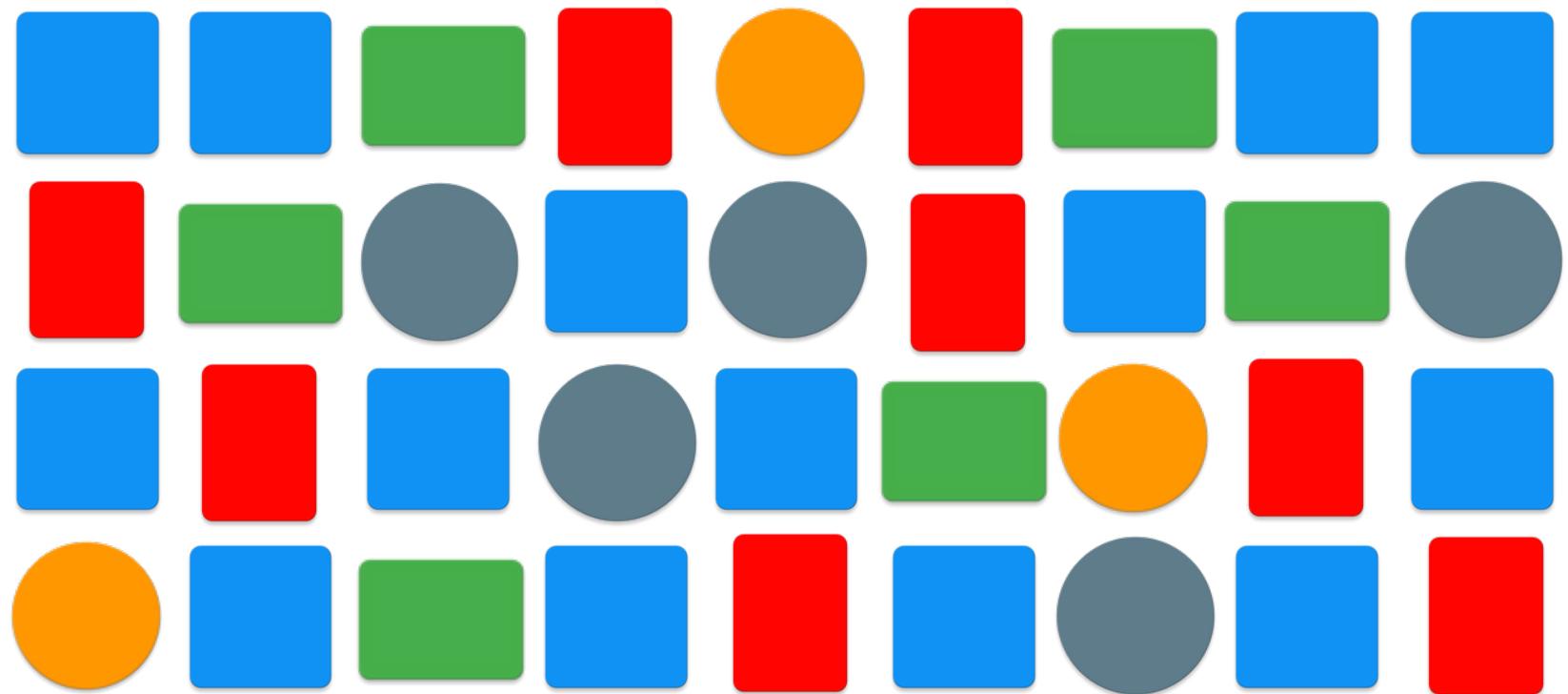
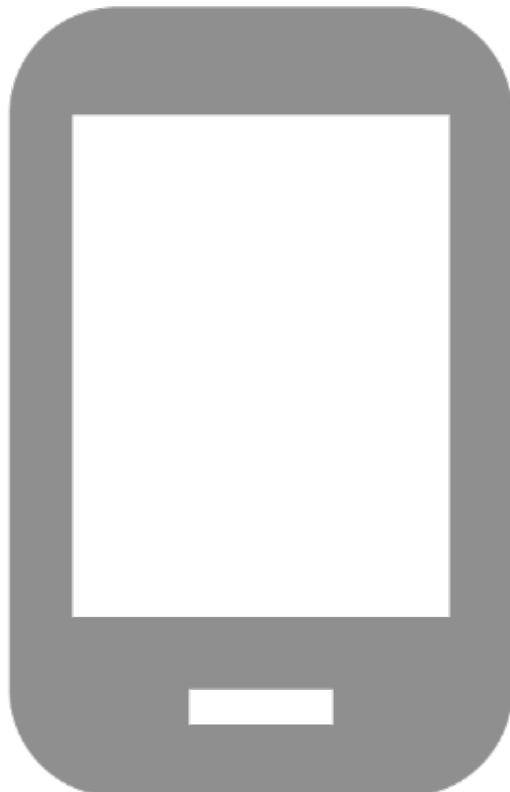
We ❤️ Apps!

189M
downloads
a day

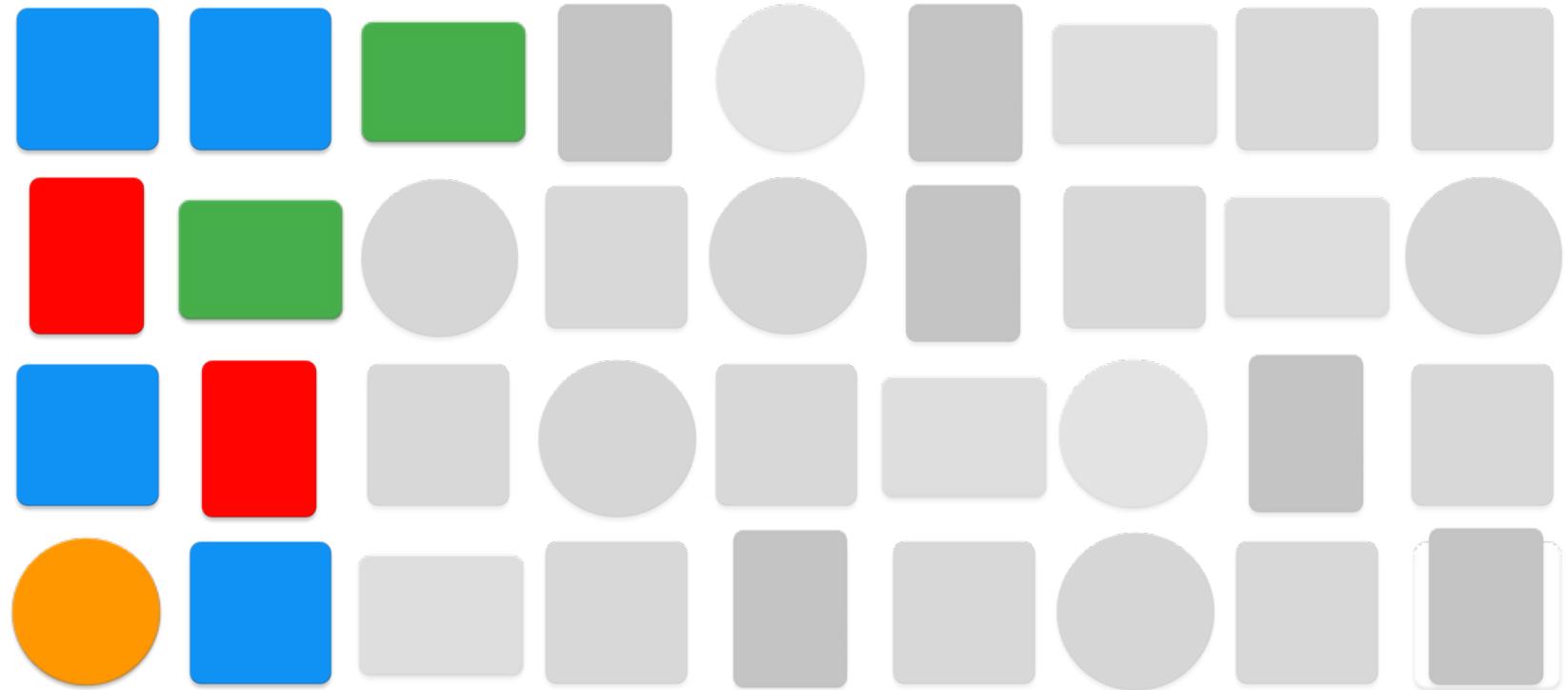
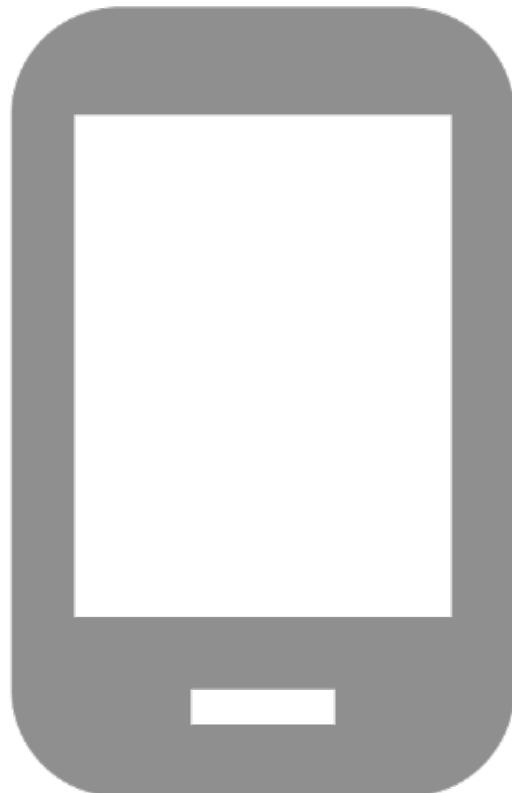
200
mins on
phone

127
mins in
apps

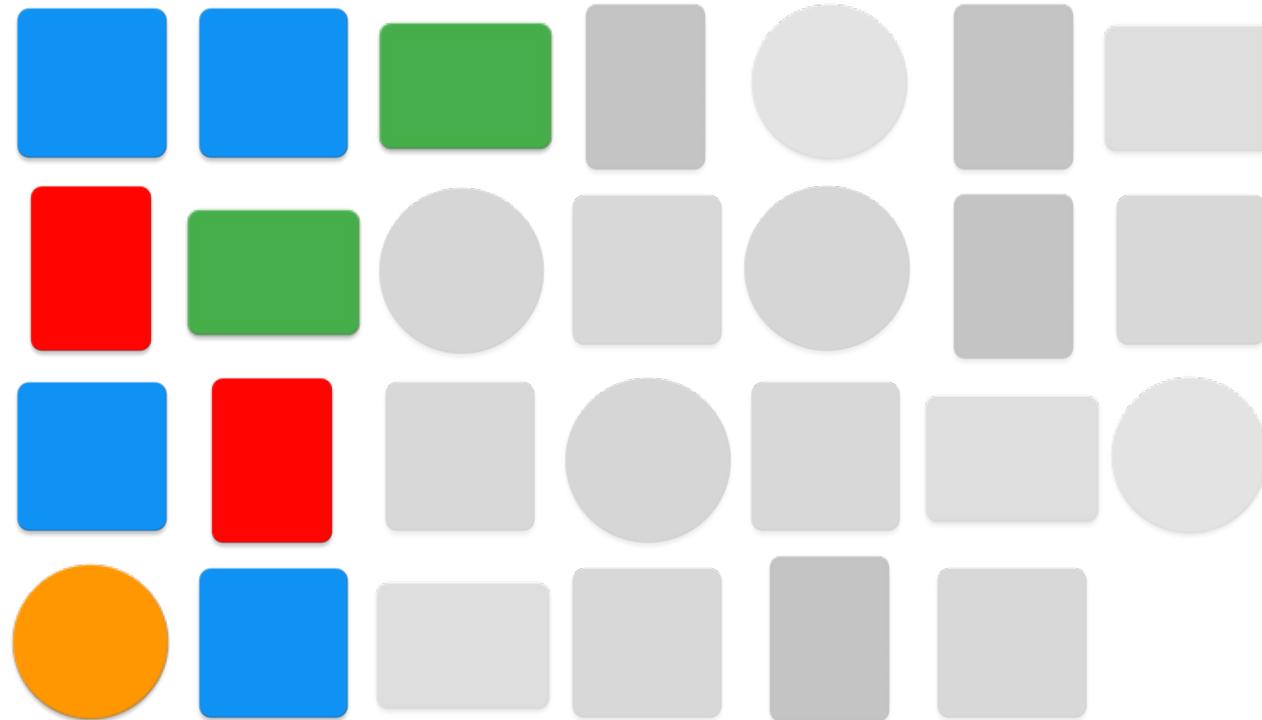
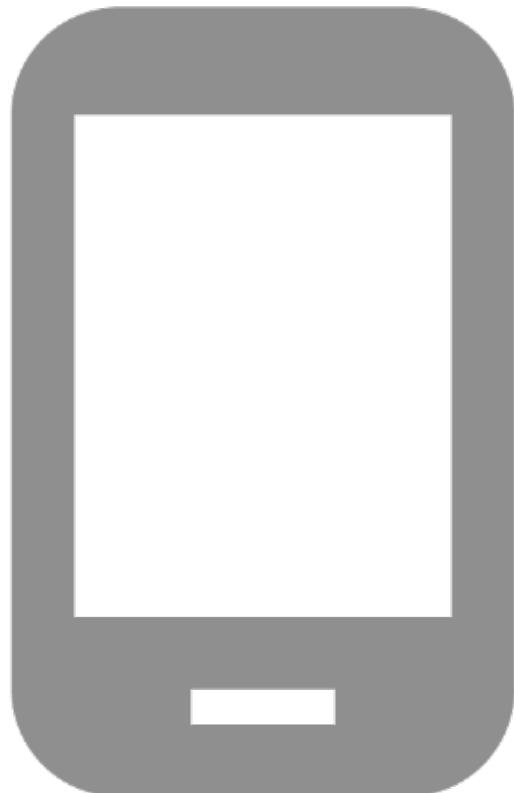
The average app user has **36** apps installed on his or her phone.



Only 1/4 are **used daily**:



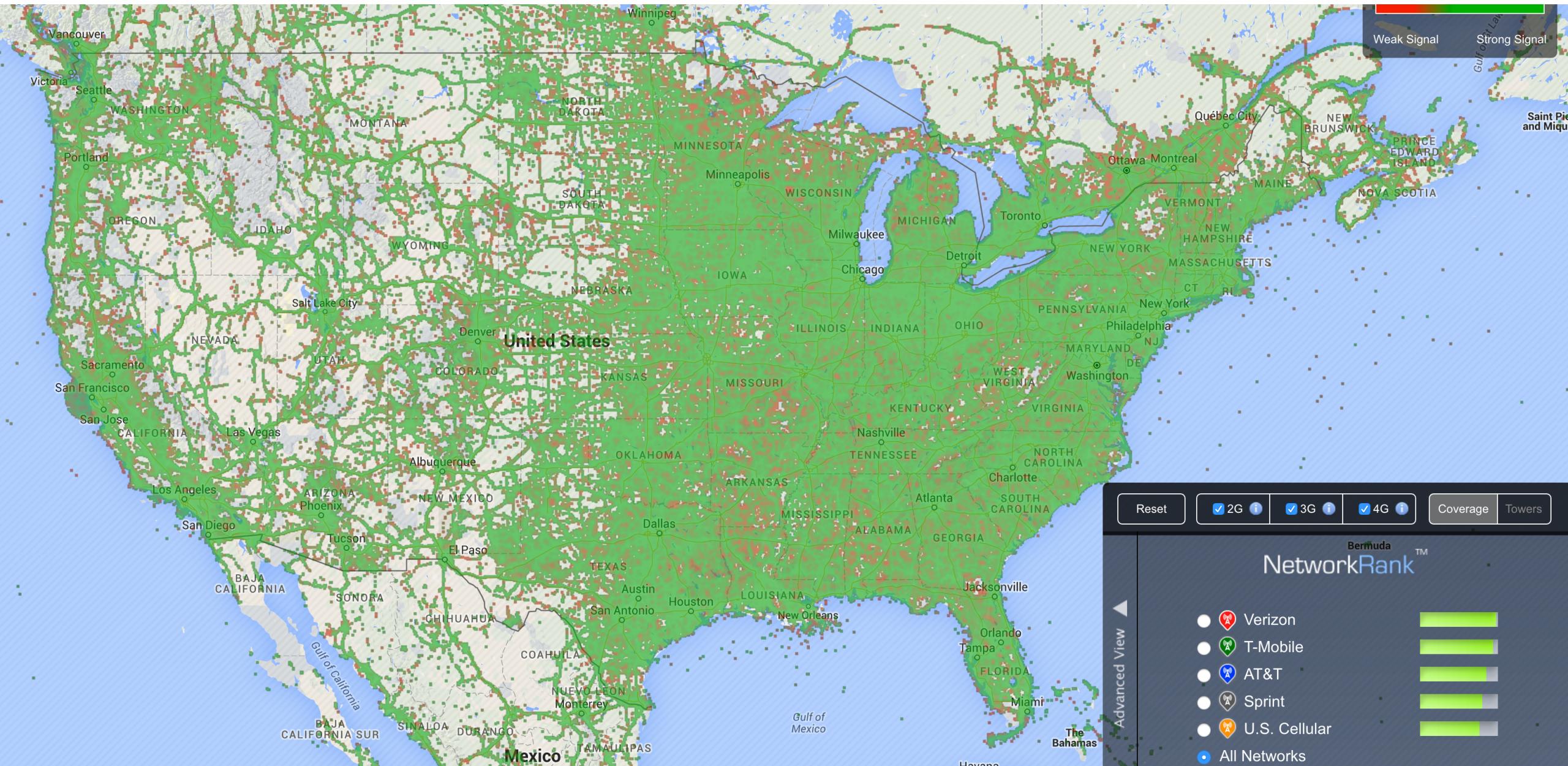
1/4 of apps are never used!



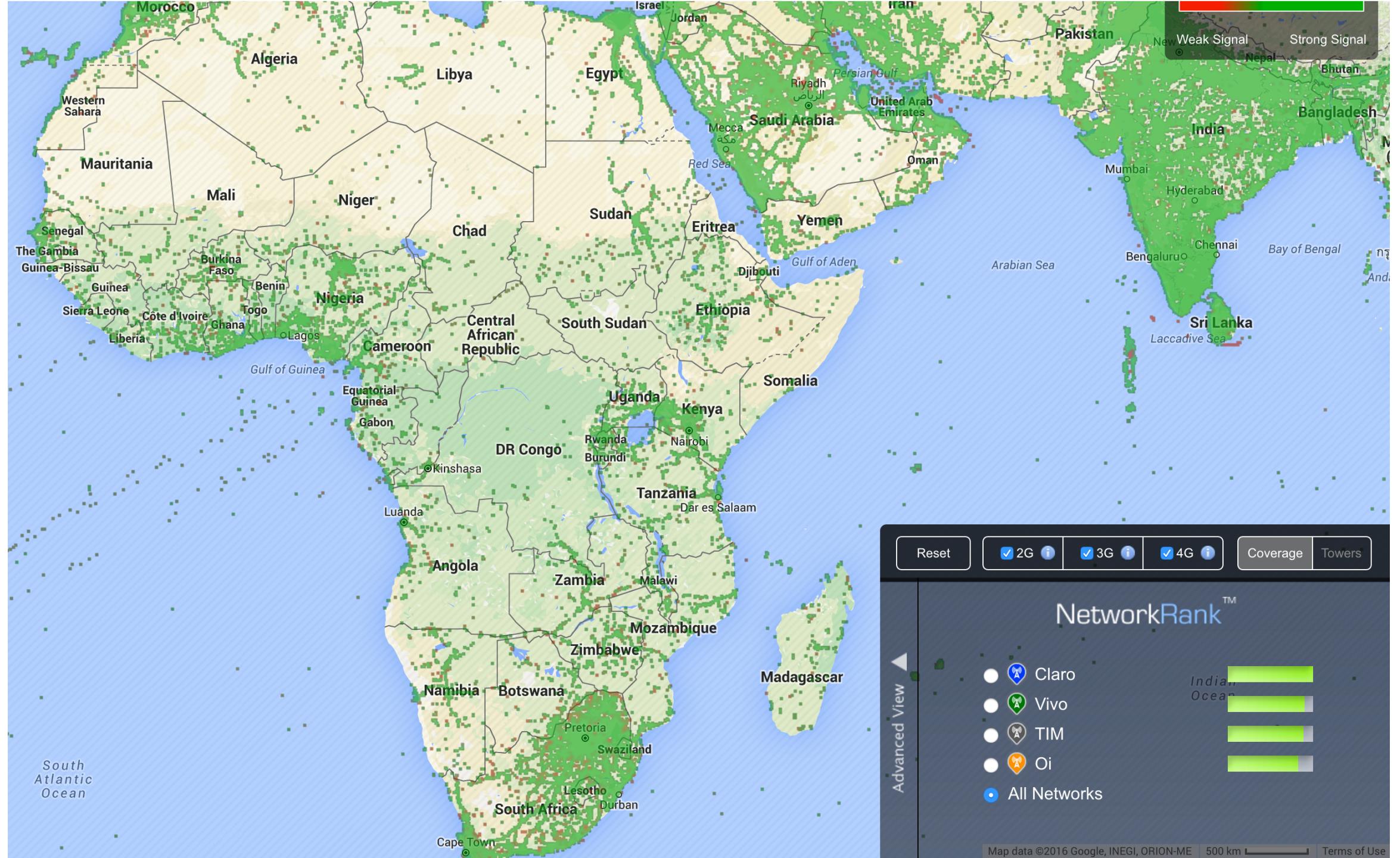
Bad App Experiences

- Slow or laggy experience
- Crashes
- Not intuitive & bad user experience
- Features not as advertised
- Data not available when you need it

Always connected?



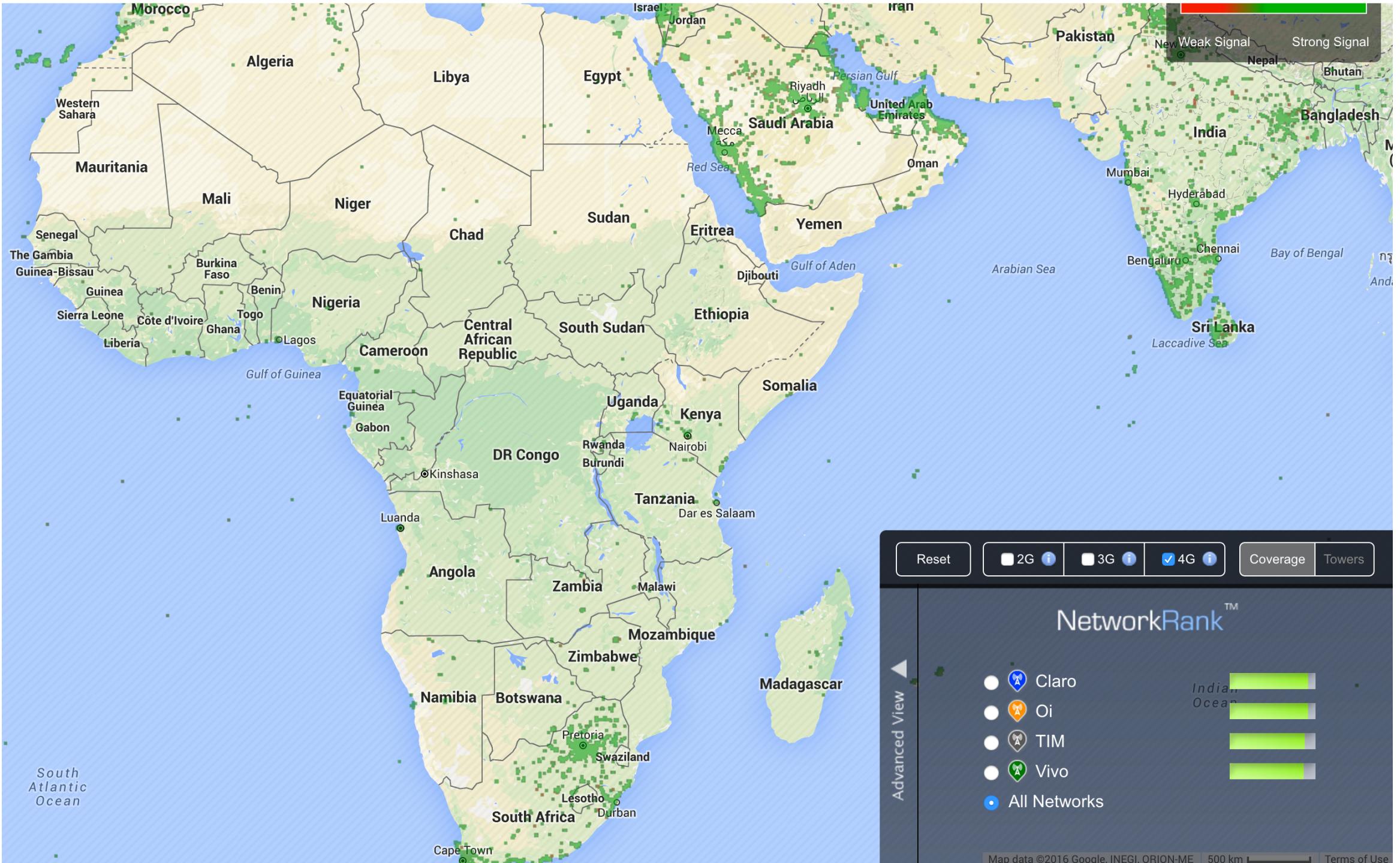
<http://opensignal.com/coverage-maps>



Reset 2G 3G 4G Coverage Towers

NetworkRank™





What about a backend?

Plenty of Options



Azure Mobile Apps



IBM MobileFirst



Amazon Web Services



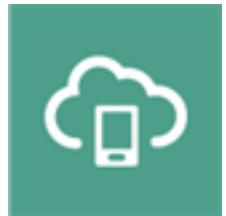
SQLCipher



Couchbase



Realm



Oracle Mobile Cloud

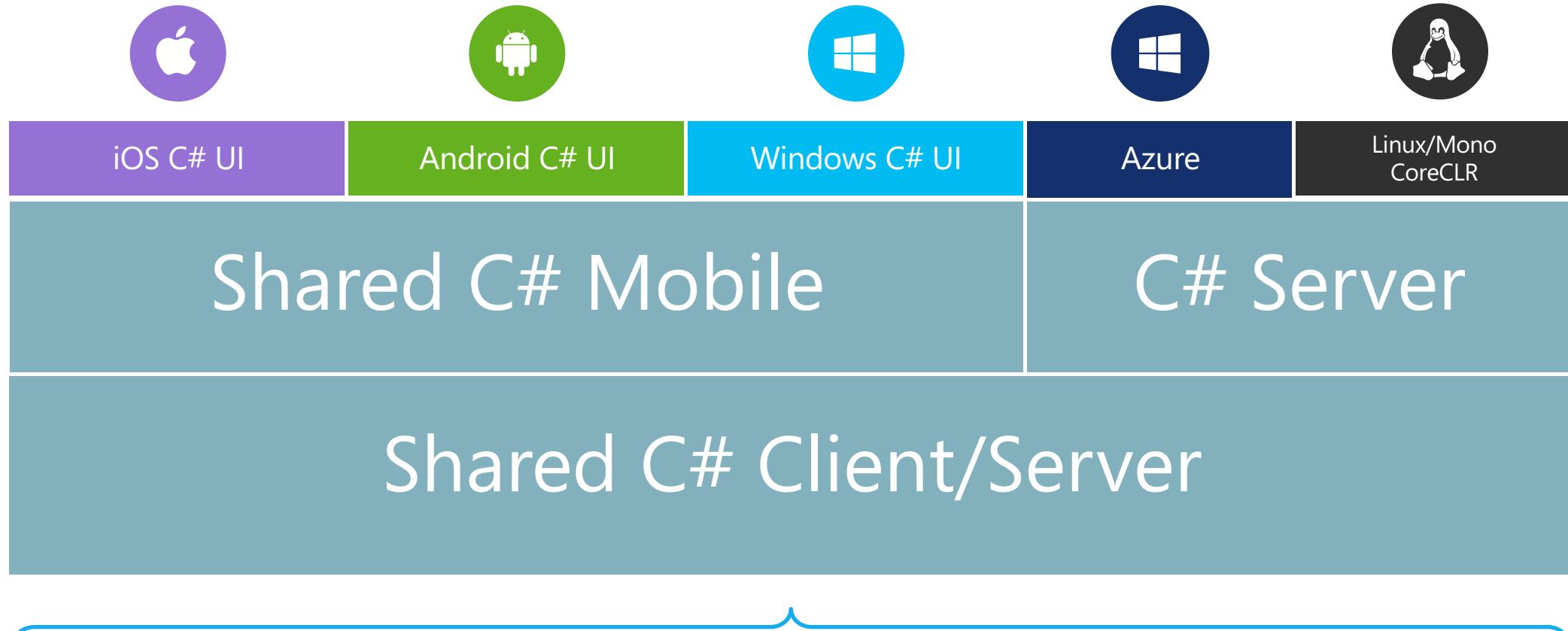


SQLite-net

Why Azure?

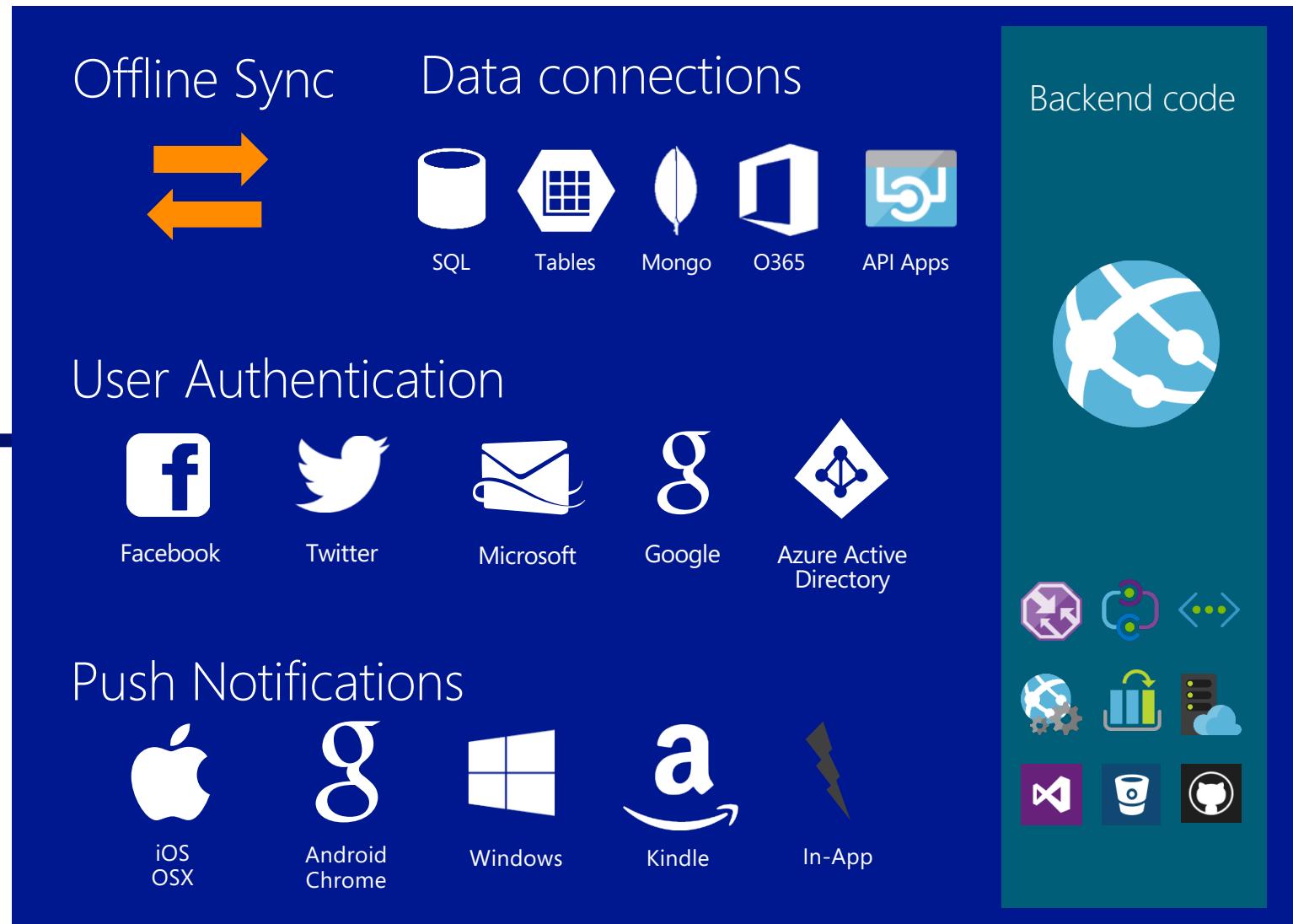
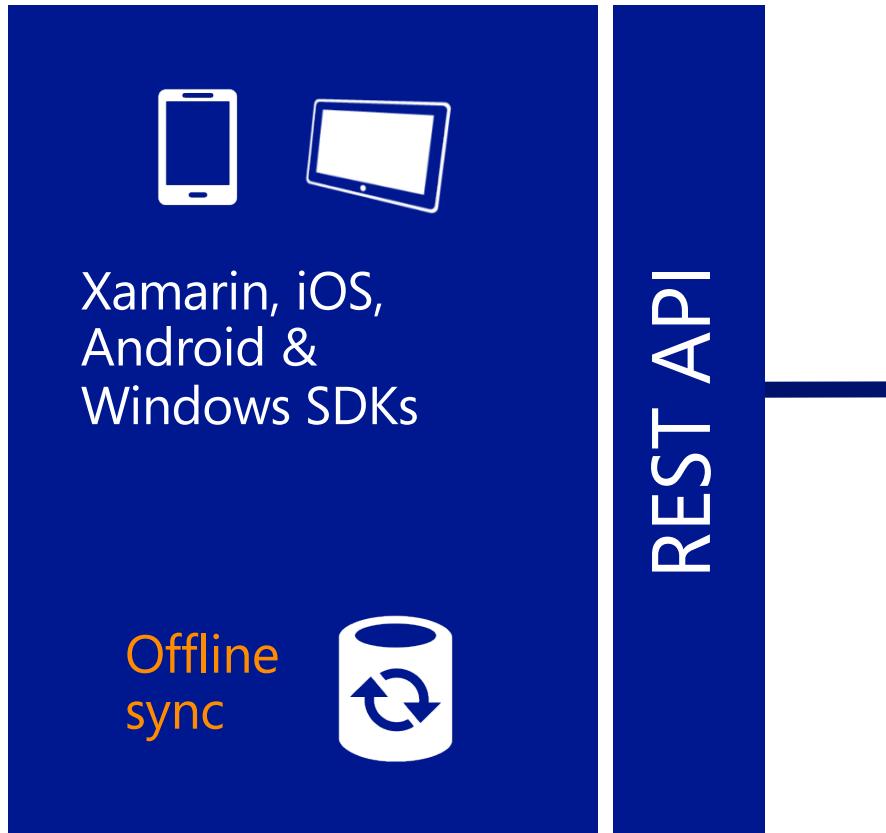
- *Extremely* powerful
- Flexible
 - Easy Tables
 - App Service
- C# SDKs available everywhere:
 - C#- iOS, Android, & Windows with Xamarin
 - C# clients, written by C# developers (open source)
 - C# backend with ASP.NET

Xamarin Apps + Backend Services



Shared C# codebase • 100% native API access • High performance

Azure Mobile Apps



Create a Mobile Service

```
MobileService = new MobileServiceClient(  
    "https://myapp.azurewebsites.net");
```

Create Tables

```
IMobileServiceSyncTable<Store> table;  
public async Task Init()  
{  
    const string path = "syncstore.db";  
    var db = new MobileServiceSQLiteStore(path);  
    db.DefineTable<Store>();  
  
    var handler = new MobileServiceSyncHandler();  
    await MobileService.SyncContext.InitializeAsync(db, h);  
    table = MobileService.GetSyncTable<Store>();  
}
```

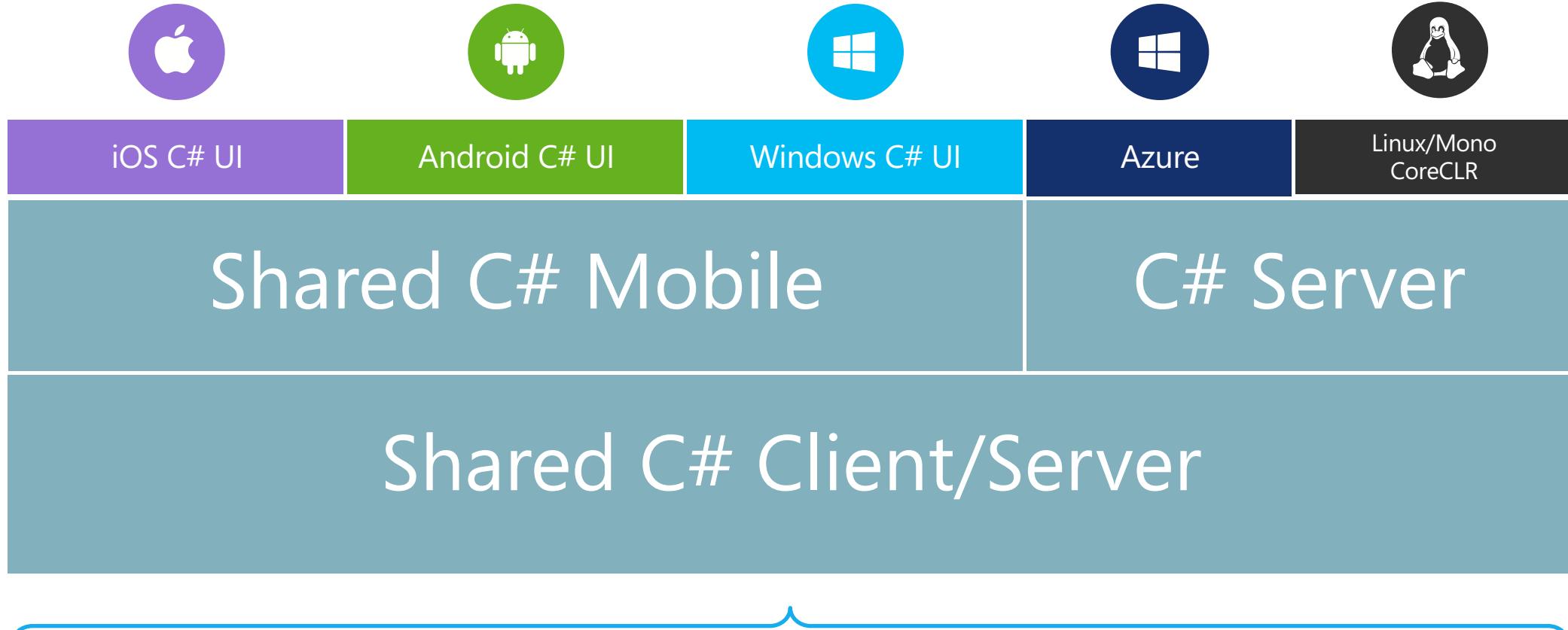
Get and Modify Data

```
public async Task<IEnumerable<Store>> GetStoresAsync()
{
    await table.PullAsync("allStores", table.CreateQuery());
    return await table.ToEnumerableAsync();
}

public async Task<Store> AddStoreAsync (Store store)
{
    await table.InsertAsync (store);
    await table.PullAsync("allStores", table.CreateQuery());
    await MobileService.SyncContext.PushAsync();
    return store;
}
```

Let's add a backend

Mobile + Server



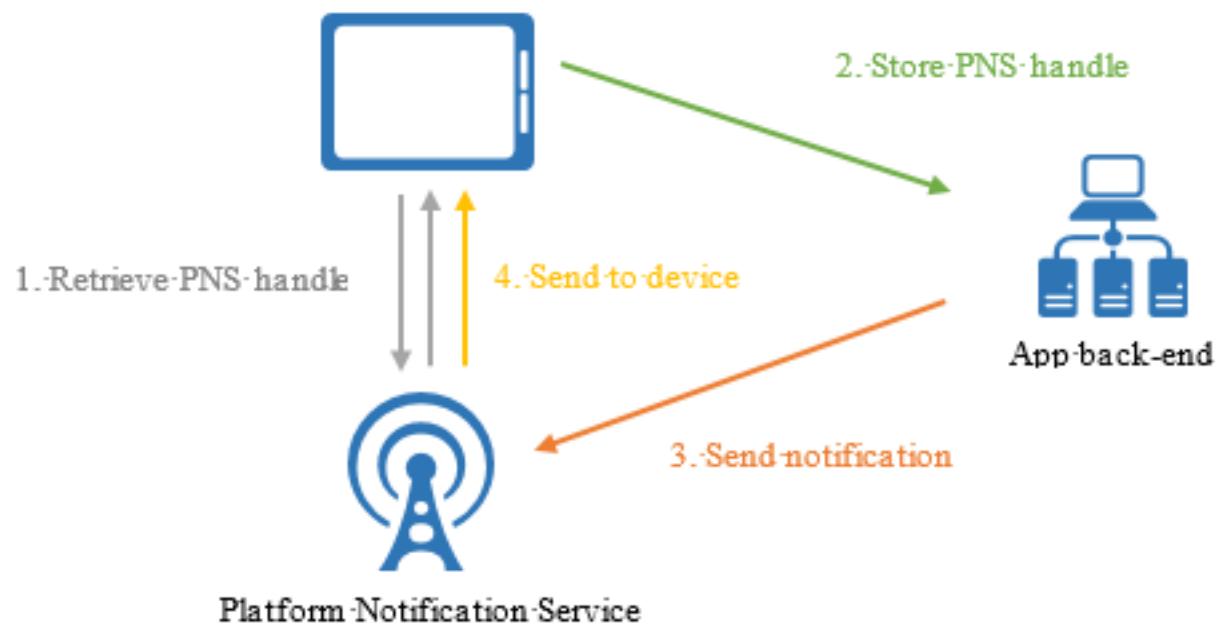
Shared C# codebase • 100% native API access • High performance

Authentication

- Rolling your own account infrastructure is difficult and time-consuming
- Secure your app with prebuilt authentication providers
 - Facebook
 - Twitter
 - Google
 - Microsoft
 - Azure AD
 - Anything OAuth 2

Push Notifications

- Easy-to-use, multiplatform scaled push infrastructure that allows you to send push notifications almost anywhere.



File Sync

- Sync files to Azure Storage, just like you did for structured data.





App Service Helpers

Add Azure to your .NET app with 4 lines of code

App Service Helper

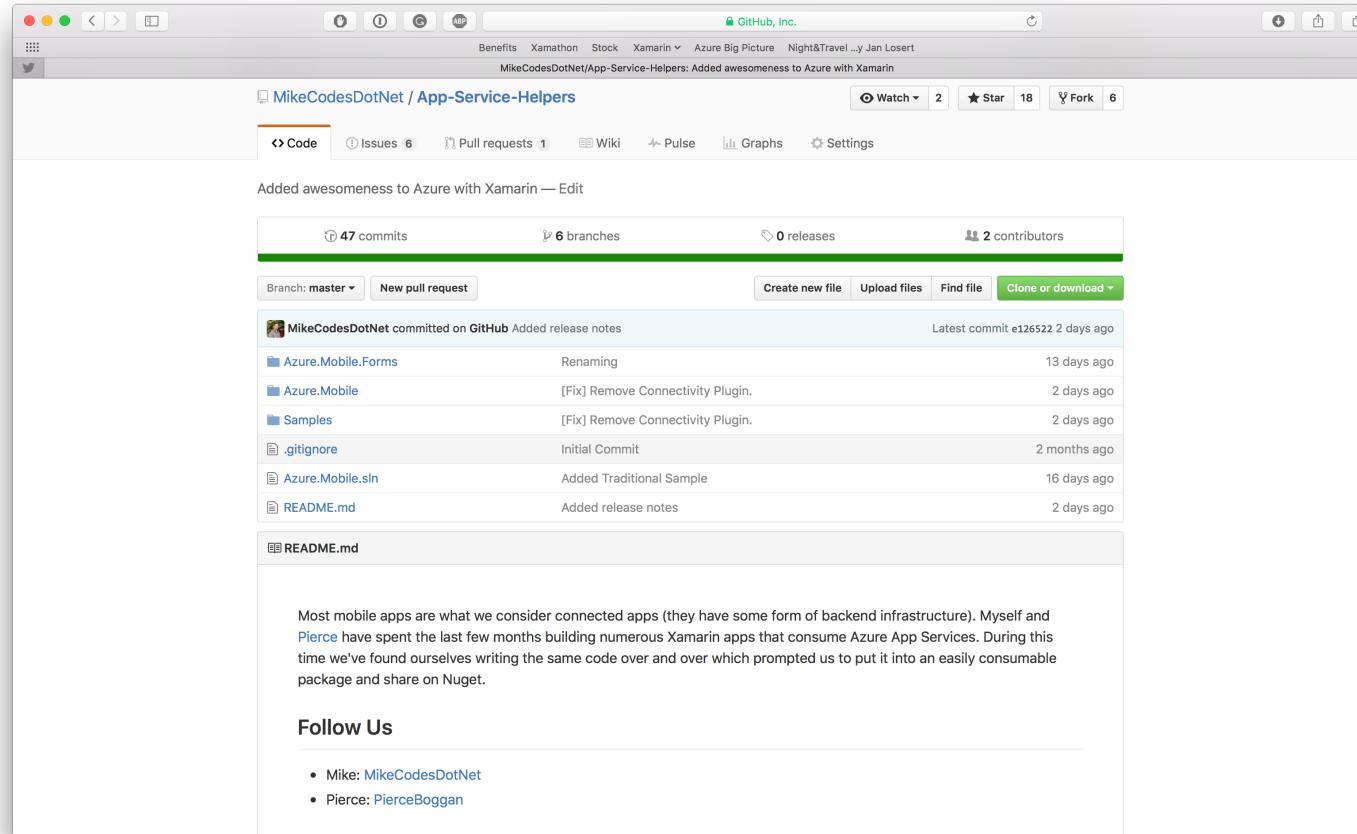
- Built on top of Azure Mobile SDK
- Modular architecture
- Available on Nuget
- Open Source

Getting Started

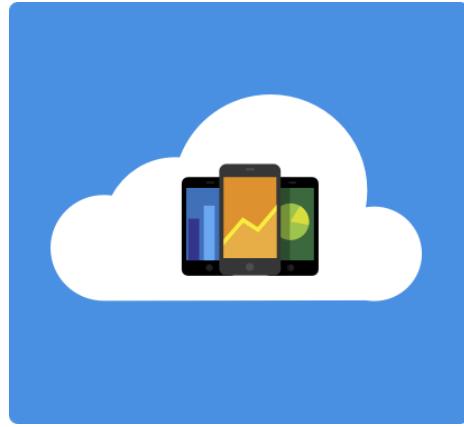
```
var client = new AppServiceHelpers.EasyMobileServiceClient();
client.Initialize(Helper.Keys.AzureServiceUrl);

client.RegisterTable<Headline>();
client.FinalizeSchema();
```

Download today



github.com/MikeCodesDotNet/App-Service-Helpers



App Service Helpers

Hands on during lab after lunch!

Lunch!

Roy Cornelissen

Lead Consultant, Xpirit

rcornelissen@xpirit.com

roycornelissen.wordpress.com

@roycornelissen