

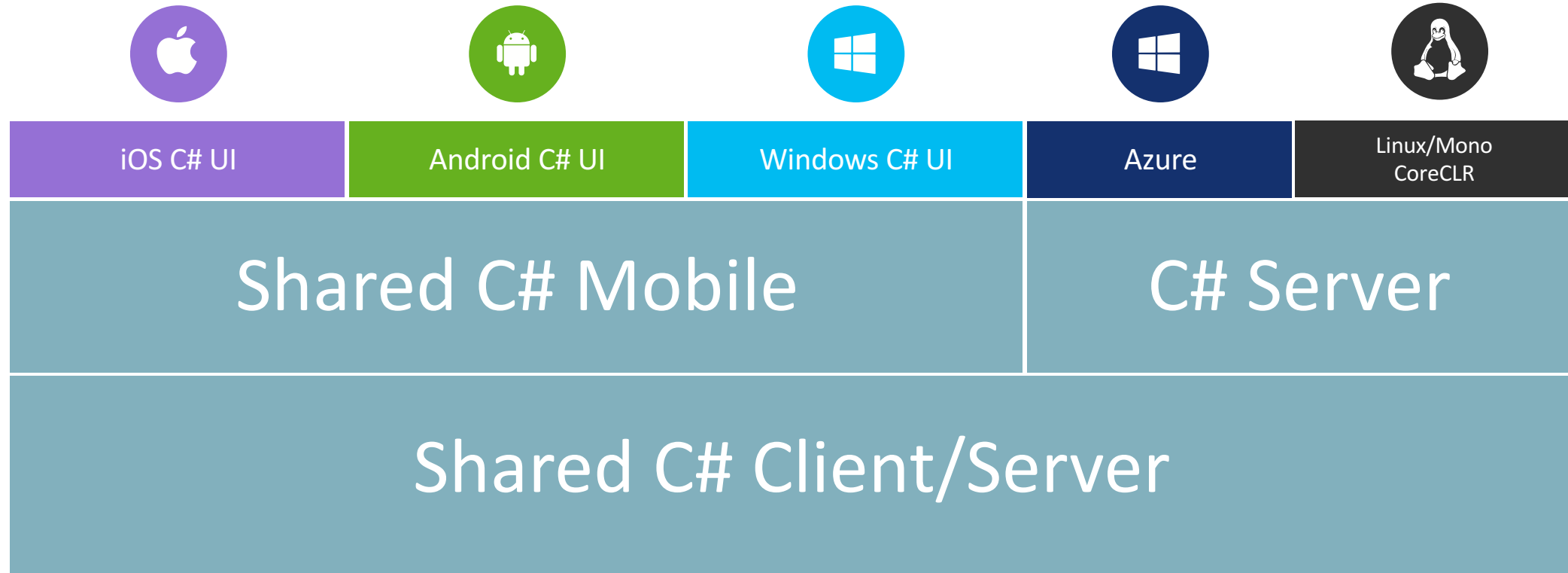
Developing Cloud-Connected Native iOS, Android & Windows Apps in Xamarin

Nish Anil, Xamarin Team @ Microsoft
@nishanil

Gaurav Seth
TSP, Azure App Services @ Microsoft

All (almost?) apps require a backend.

Xamarin Apps + Backend Services



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

Plenty of Options



Azure Mobile Apps



IBM MobileFirst



Amazon Web Services



SQLCipher



Couchbase



Realm



Oracle Mobile Cloud



SQLite-net



Azure App Service

“PaaS that developers love and your business can trust”

SECURITY AND MANAGEMENT



PLATFORM SERVICES

Compute



App Service



Azure Functions



Developer Services



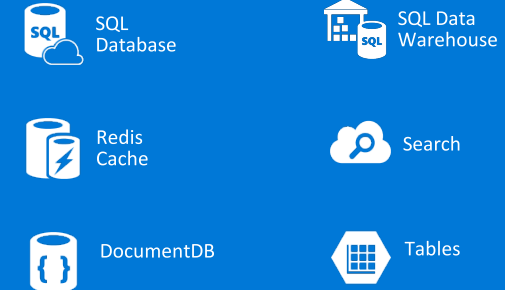
Integration



Analytics and IoT



Data



Media & CDN



HYBRID OPERATIONS



INFRASTRUCTURE SERVICES

Compute



Storage



Networking



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



Mobile Apps

Engage employees, partners and customers on any device at any time

- ✓ Enterprise Grade Apps
- ✓ Fully Managed Platform
- ✓ High Productivity Development

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

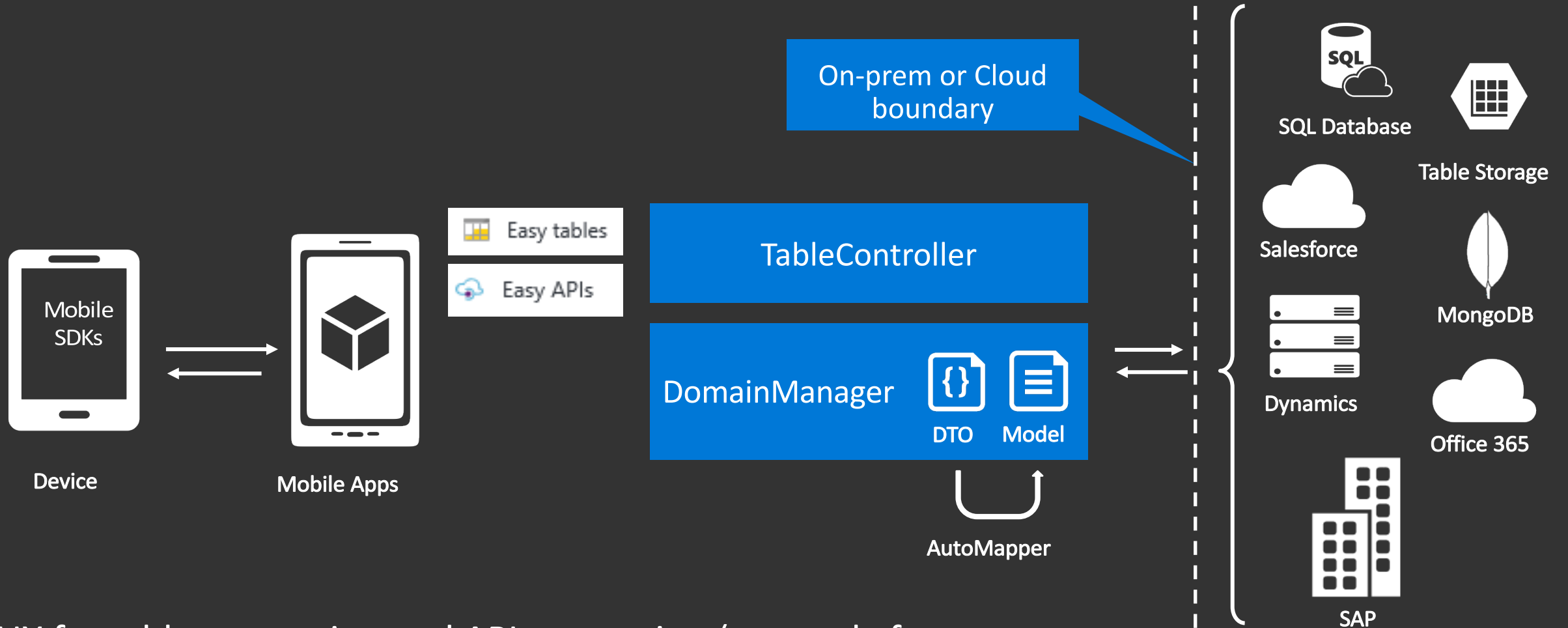
```
public async Task<IEnumerable<Store>> GetStoresAsync()
{
    await table.PullAsync("allStores", table.CreateQuery());
    return await table.ToEnumerableAsync();
}

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 Zero – Code Backend

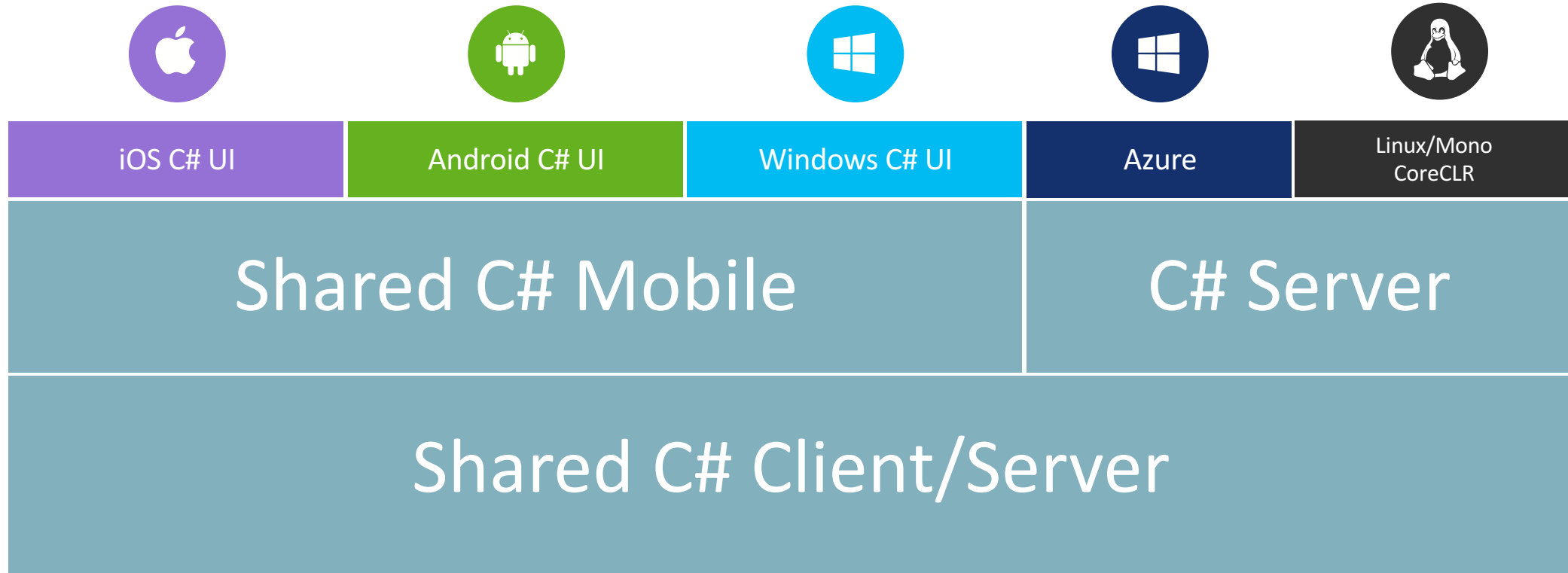
Azure - EasyTables

Data Source



UX for table generation and API generation (currently for Node.js and Azure SQL databases)

Mobile + Server



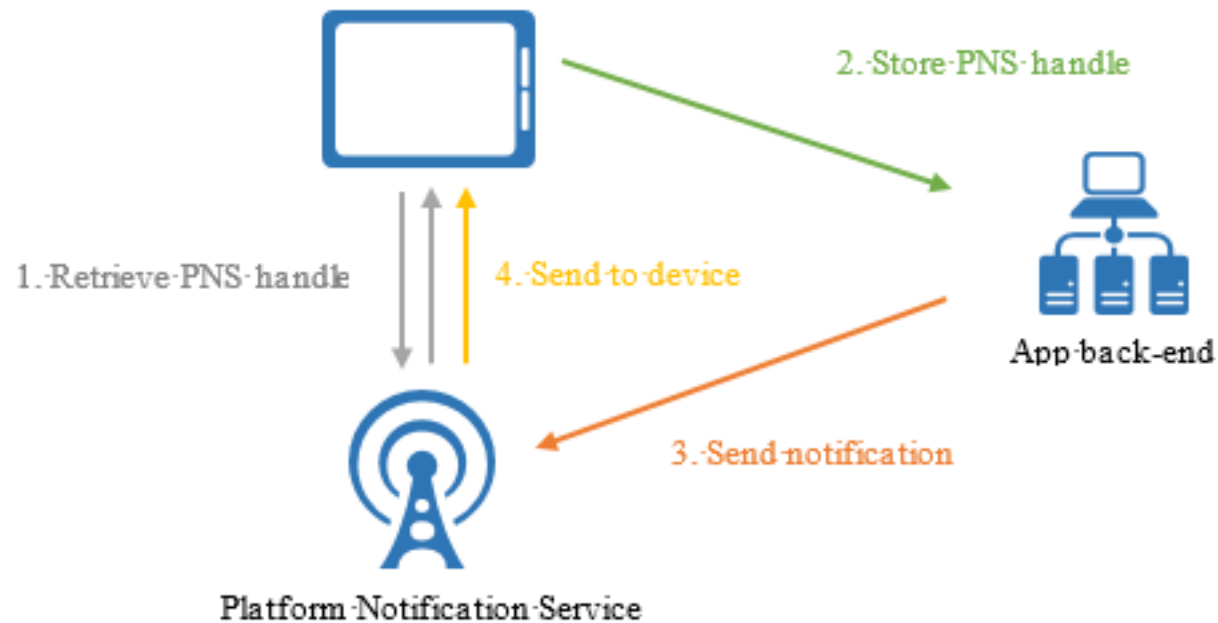
Shared C# codebase • 100% native API access • High performance

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

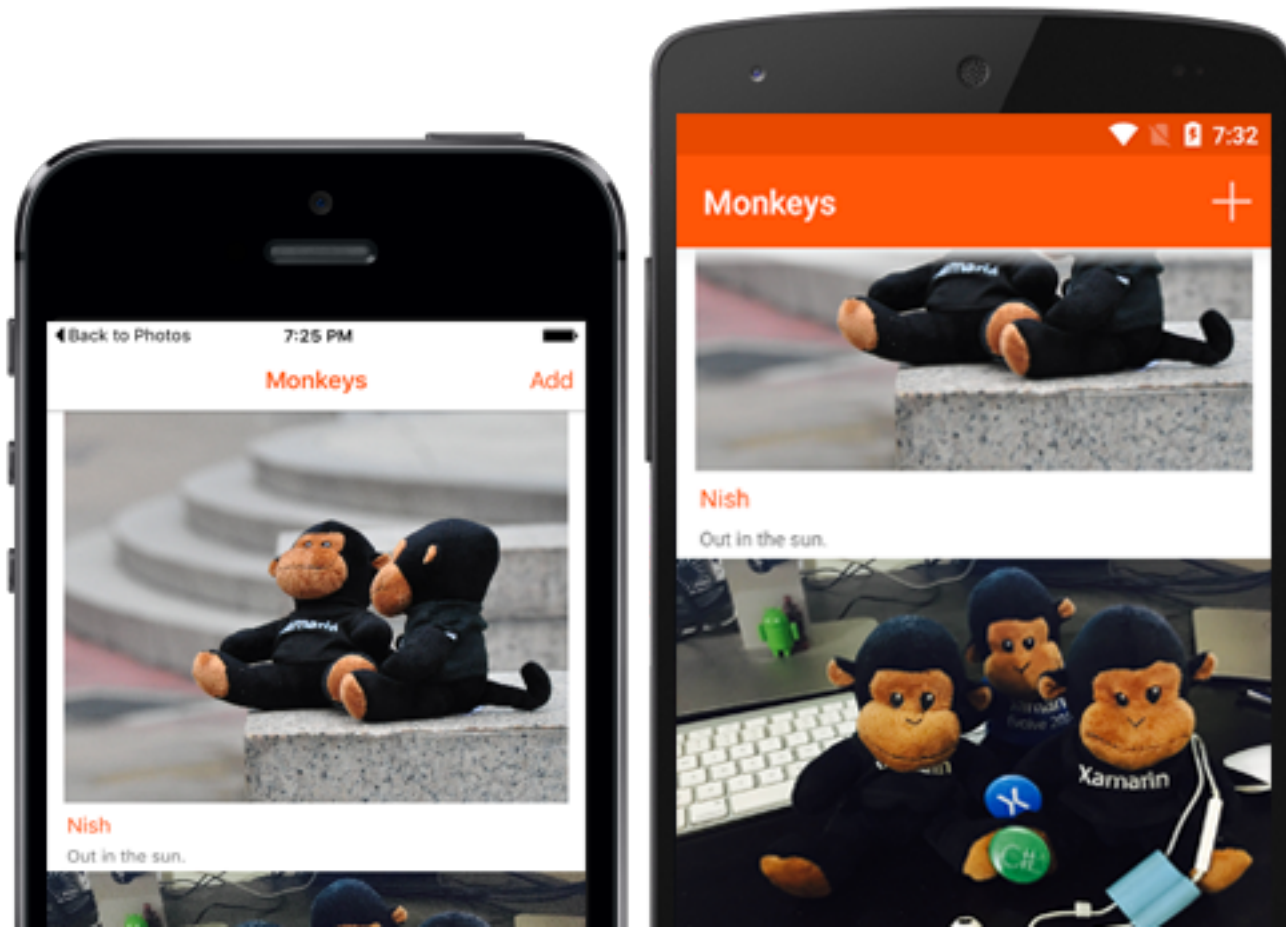
Push Notifications

- 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.



Let's add a Scalable Backend

Don't miss!

Tomorrow @ 12 noon, Hall B




Track: iOS/Android/Mobile Cloud | September 15-16, 2016

**Adding Intelligence to your Apps using
Microsoft Cognitive Services**

Mayur Tendulkar

Up Next

Hall B



Track: iOS/Android/Mobile Cloud | September 15-16, 2016

Continuous Delivery for iOS, Android & Windows Apps using Xamarin, VSTS & HockeyApp

Nish Anil



Get Started Today

xamairn.com/download
tryappservice.azure.com/

More!



aka.ms/vsandexamarin



aka.ms/gettingstartedwithxamarin



github.com/xamarin



twitter.com/msdevindia



facebook.com/MicrosoftDeveloper.India/



Booth # 10

Thank you. Questions?

Nish Anil

Senior PM– Xamarin @ Microsoft

Gaurav Seth

TSP, Azure App Service Microsoft

nish@microsoft.com

@nishanil