

2017

Global **Azure** **BOOTCAMP**



AzureHeads

 autoexec.gr

 SQLschool.gr



Azure Mobile Apps and Xamarin: From zero to hero

Nasos Loukas
Mobile Team Leader @ KYON
aloukas@outlook.com



From zero to hero

Chapter 0: Xamarin

Chapter 1: Azure Mobile Apps

Chapter 2: Offline Sync

Chapter 3: Authentication

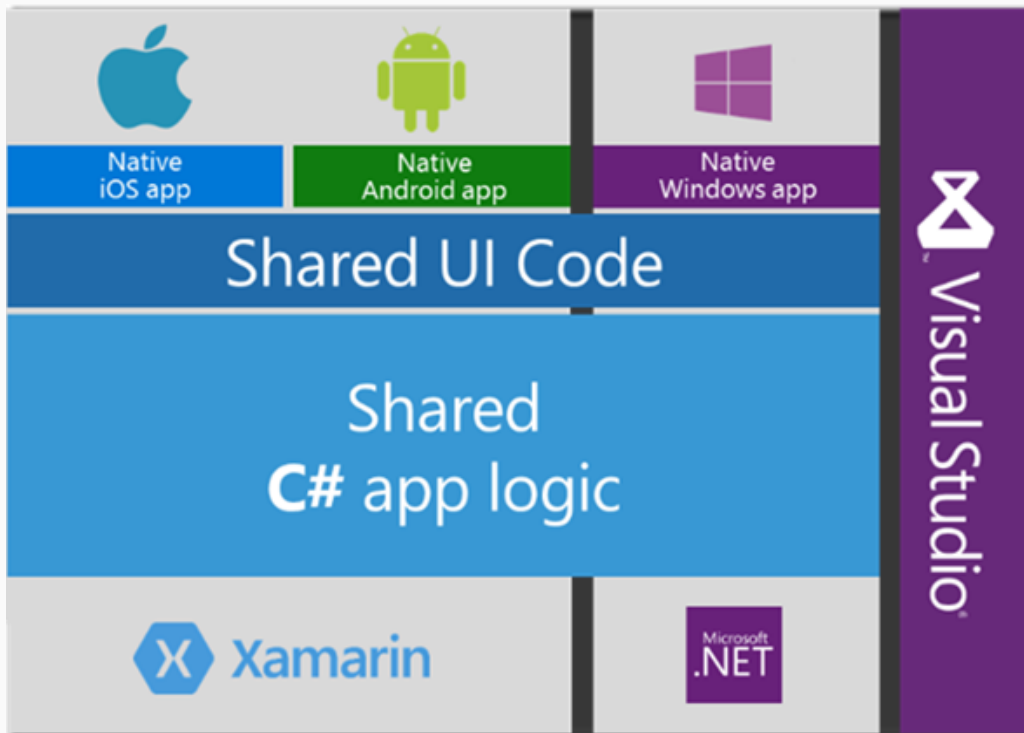
Chapter 4: Push Notifications

Chapter 5: AI with Cognitive Services

Zero or Hero? Show me the code!

Conclusions

Chapter 0: Xamarin



Chapter 0: Xamarin

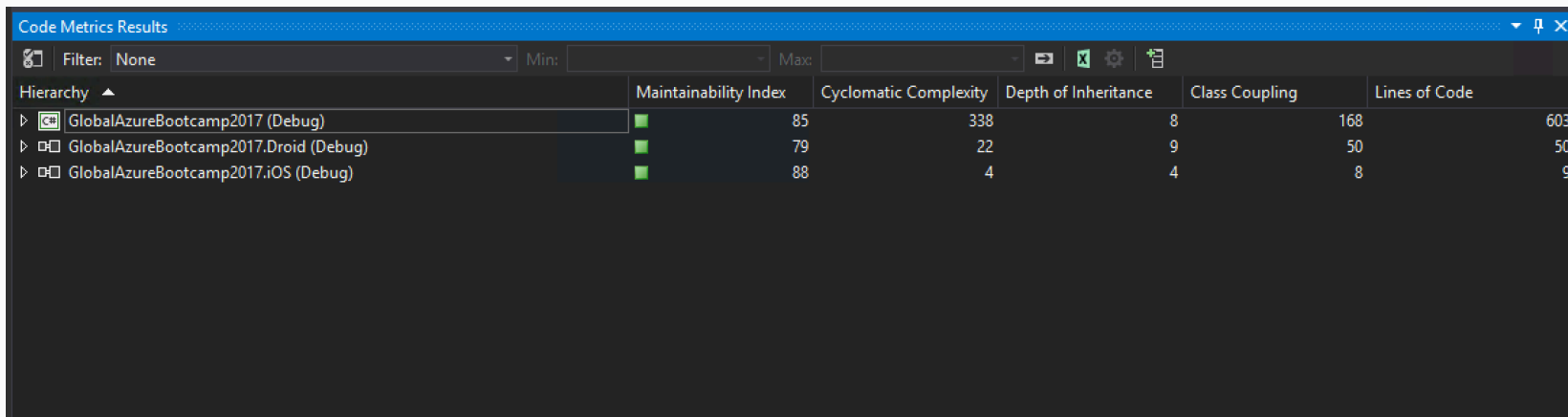
Features

- Complete binding for the underlying SDKs
 - Xamarin.iOS
 - Xamarin.Android
- Native libraries invoking
- C# with lamdas, LINQ etc
- .NET BCL
 - XML, Database, Serialization, Networking etc.
- Xamarin/Visual Studio
- Code Sharing

Chapter 0: Xamarin

Demo Metrics

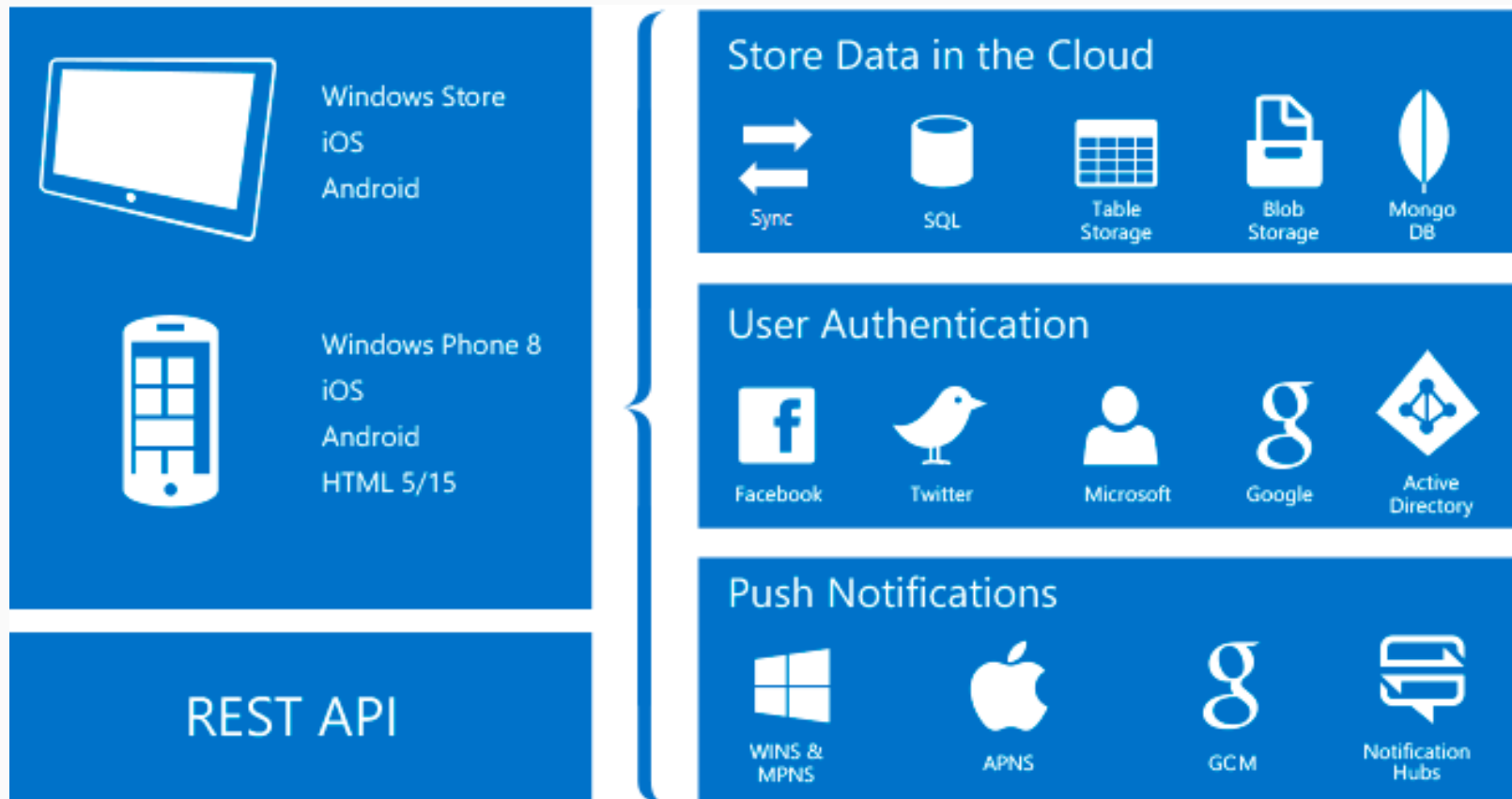
- PCL project
 - Xamarin.Forms
- Shared ~91% of code!!



The screenshot shows a 'Code Metrics Results' window with a table of metrics for three projects. The table has columns for Maintainability Index, Cyclomatic Complexity, Depth of Inheritance, Class Coupling, and Lines of Code. Each row includes a green square icon indicating a positive metric value.

Hierarchy	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling	Lines of Code
GlobalAzureBootcamp2017 (Debug)	85	338	8	168	603
GlobalAzureBootcamp2017.Droid (Debug)	79	22	9	50	50
GlobalAzureBootcamp2017.iOS (Debug)	88	4	4	8	9

Chapter 1: Azure Mobile Apps



Chapter 2: Offline Sync

- Client
 - Local store (SQLite or Core Data)
- Backend
 - Easy tables (No-Code Table)
- Benefits
 - Improve app responsiveness
 - Allow users to create/edit data offline
 - Limit network usage on high-latency or metered networks
 - Sync across multiple devices

Chapter 2: Offline Sync

Query Data

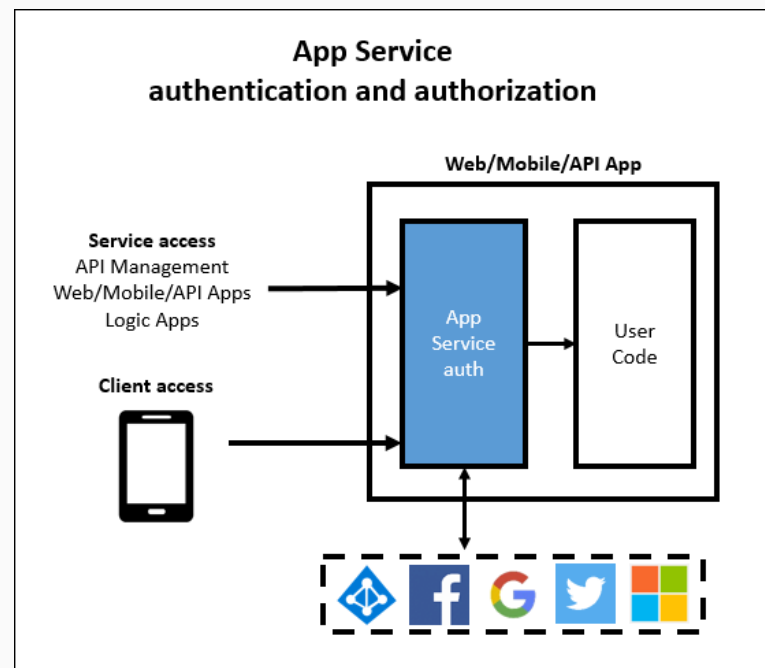
- The SDK supports all CRUD operations
- Filtering:
 - Relational operators (==, !=, <, <=, >, >=)
 - Arithmetic operators (+, -, /, *, %)
 - Number precision (Math.Floor, Math.Ceiling)
 - String functions (Length, Substring, Replace, IndexOf, StartsWith, EndsWith)
 - Date properties (Year, Month, Day, Hour, Minute, Second)
 - Access properties of an object, and
 - Expressions combining any of these operations

```
var items = await _table.Where(e => e.EventId==eventId).ToEnumerableAsync();
```

- Sorting
- Paging

Chapter 3: Authentication

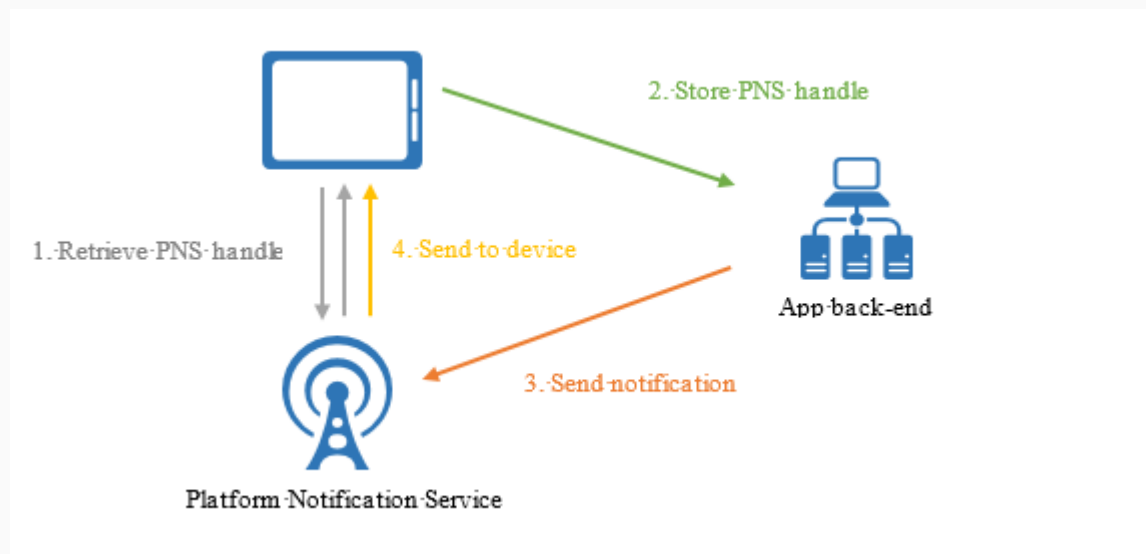
- Various external identity providers
 - Facebook
 - Google
 - Microsoft
 - Twitter
 - Azure Active Directory
- Multiple flows
 - Client-managed
 - Server-managed



Chapter 4: Push Notifications

Platform Notification Systems

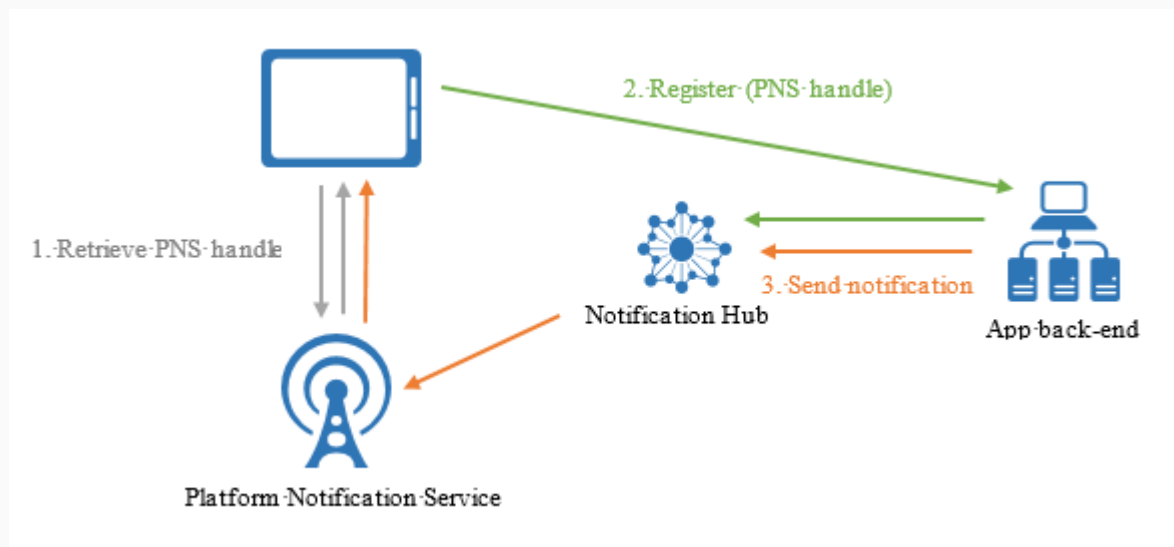
- Platform dependency
- Scale
- Routing



Chapter 4: Push Notifications

Notification Hub

- Cross platform
- Cross backend
- Rich set of delivery patterns
- Rich telemetry
- Scalability
- Security



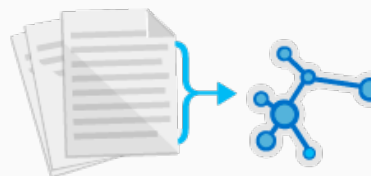
Chapter 5: AI with Cognitive Service

- Powerful AI algorithms for vision, speech, language, and knowledge.
- Text Analytics API:

Sentiment Analysis



Topic detection



Key phrase extraction



Language detection



Chapter 6: Show me the code

Demo

```
private AzureService()
{
    var url = new Uri(Constants.ApplicationUrl);

    // init service
    Client = new MobileServiceClient(url);

    // init store
    var store = new MobileServiceSQLiteStore($"{url.Host}.db");
    store.DefineTable<Event>();
    store.DefineTable<EventUpdate>();
    store.DefineTable<Speaker>();
    store.DefineTable<Activity>();
    store.DefineTable<UserActivity>();

    //Initializes the SyncContext using the default IMobileServiceSyncHandler.
    Client.SyncContext.InitializeAsync(store);

    // init repositories
    EventRepository = new EventRepository(this, Client.GetSyncTable<Event>());
    SpeakerRepository = new SpeakerRepository(this, Client.GetSyncTable<Speaker>());
    ActivityRepository = new ActivityRepository(this, Client.GetSyncTable<Activity>());
    UserActivityRepository = new UserActivityRepository(this, Client.GetSyncTable<UserActivity>(), ActivityRepository);
    EventUpdateRepository = new EventUpdateRepository(this, Client.GetSyncTable<EventUpdate>());

    TextAnalyticsRepository = new TextAnalyticsRepository();
}
```

Chapter 6: Show me the code

Resources

- Azure Mobile Apps:
<https://docs.microsoft.com/en-us/azure/app-service-mobile/app-service-mobile-value-prop>
- Azure Mobile Apps Client SDK:
<https://docs.microsoft.com/en-us/azure/app-service-mobile/app-service-mobile-dotnet-how-to-use-client-library>
- Offline Sync:
<https://docs.microsoft.com/en-us/azure/app-service-mobile/app-service-mobile-xamarin-forms-get-started-offline-data>
- Authentication:
<https://docs.microsoft.com/en-us/azure/app-service-mobile/app-service-mobile-xamarin-forms-get-started-users>
- Push notifications:
<https://docs.microsoft.com/en-us/azure/app-service-mobile/app-service-mobile-xamarin-forms-get-started-push>
- Demo code:
<https://github.com/NaSOS/azure-mobile-apps-xamarin-demo>

Final Chapter: Conclusions

Azure Mobile Apps

- Pros
 - Many features to create a great app
 - All features are covered by the SDKs
 - SDKs are easy to use
 - Not need to code backend
 - Fast development time
- Cons
 - Need to keep models flat
 - No client-side entity relations
 - Authenticated users have access to everything

Thank You



Feedback:

<https://aka.ms/cc9cf1>

