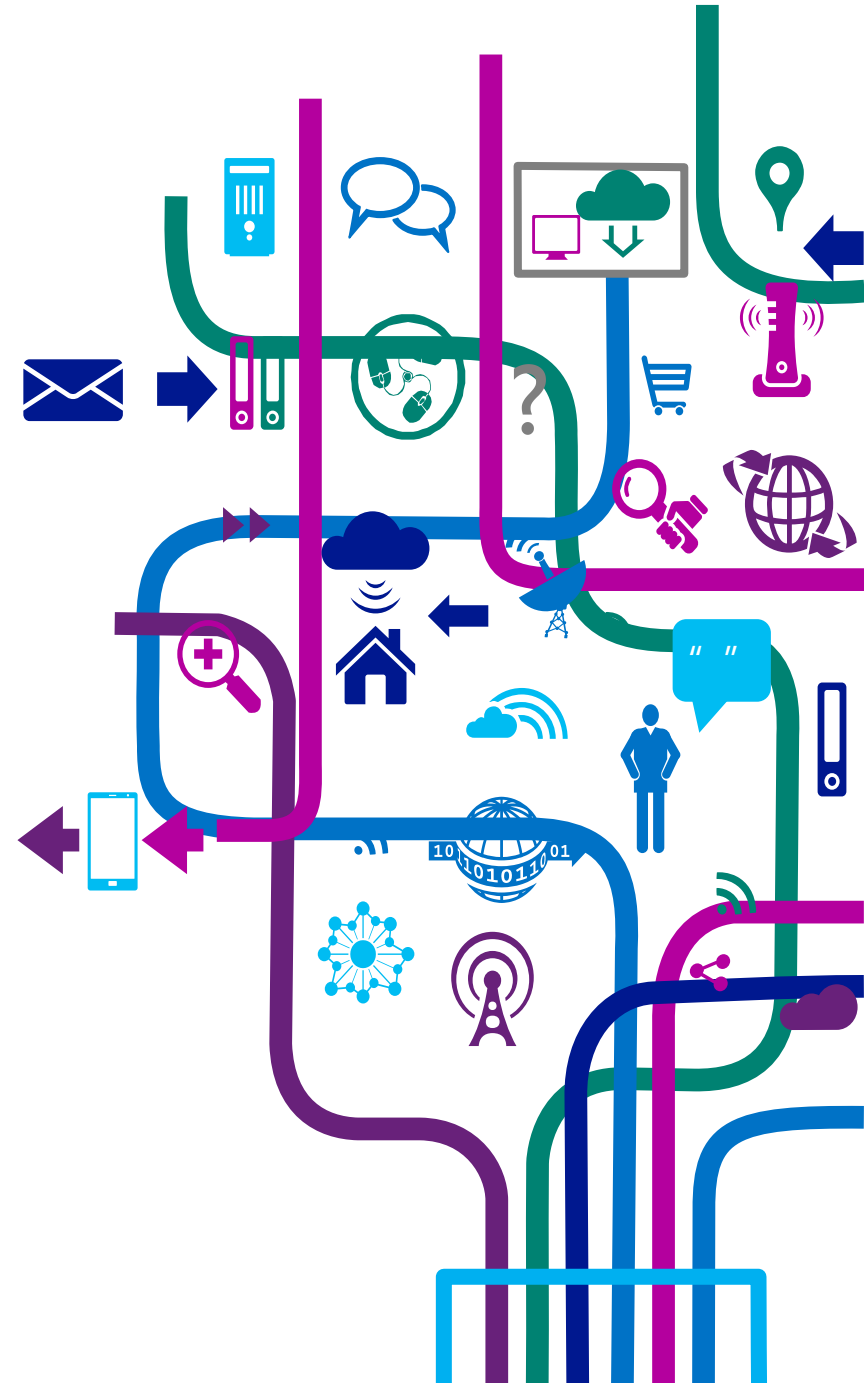


Azure Mobile App

김영욱 Technical Evangelist
부장/ DX / Microsoft

youngwook@outlook.com
Blog: Youngwook.com



Agenda

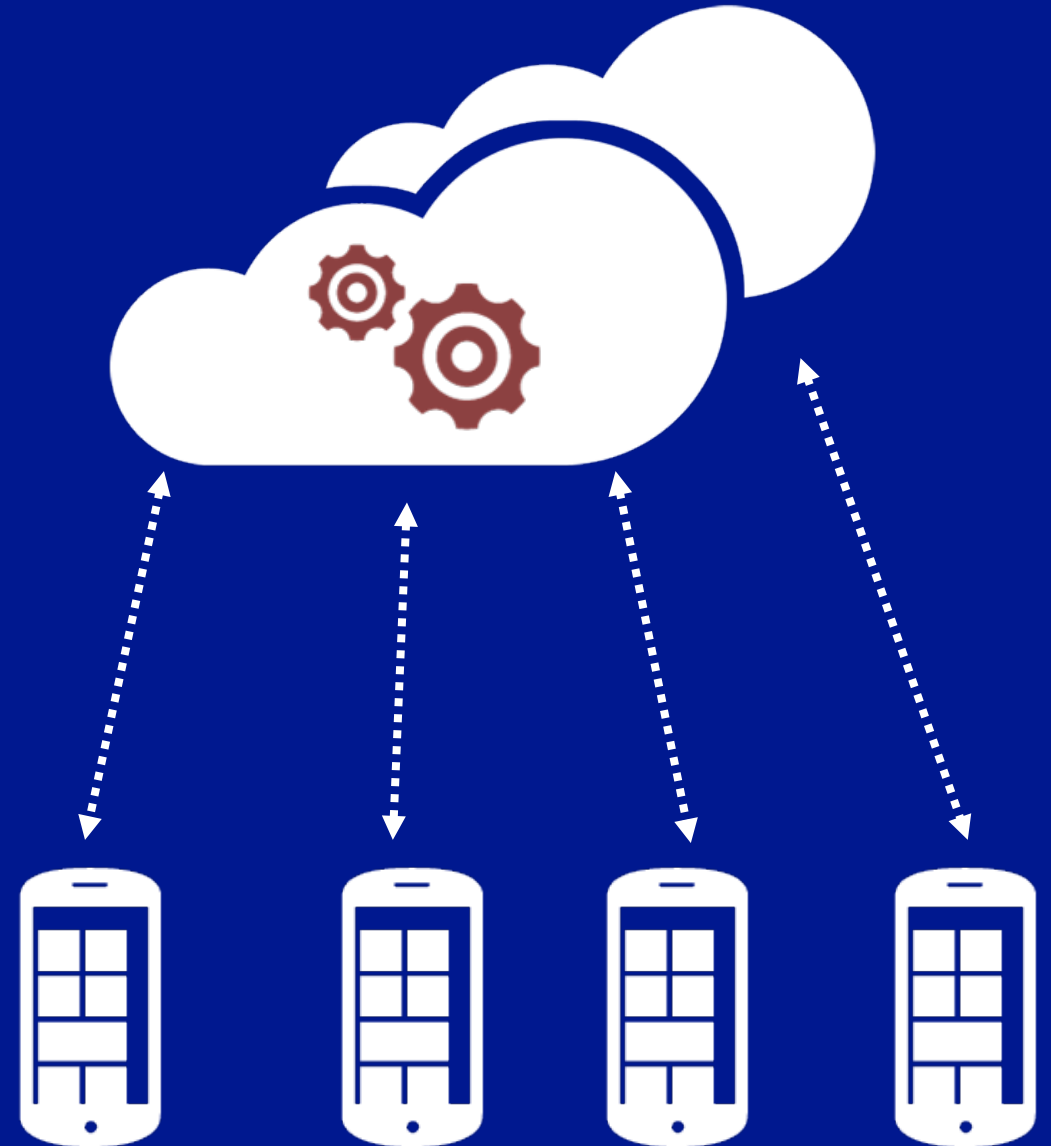
- 1st Hour: Introduction to Mobile Services (45 minutes)
 - General Architecture
 - Management Portal
 - REST API
 - Break (15 minutes)

App Development Challenges

다양한 모바일 플랫폼의 지원

한정된 개발 자원

푸시, 인증, 데이터 통신 등 백엔드 서비스의
파편화



Cross





Multiple Devices for Apps

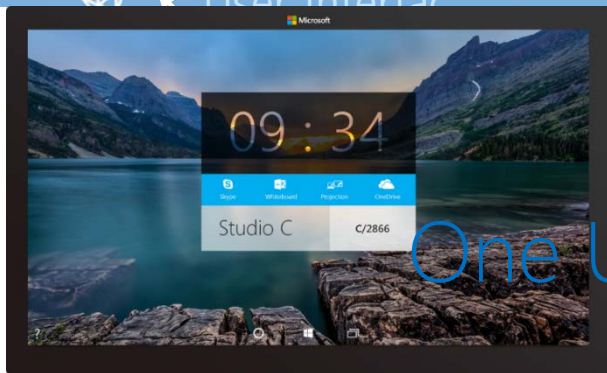


Adaptive User Interface

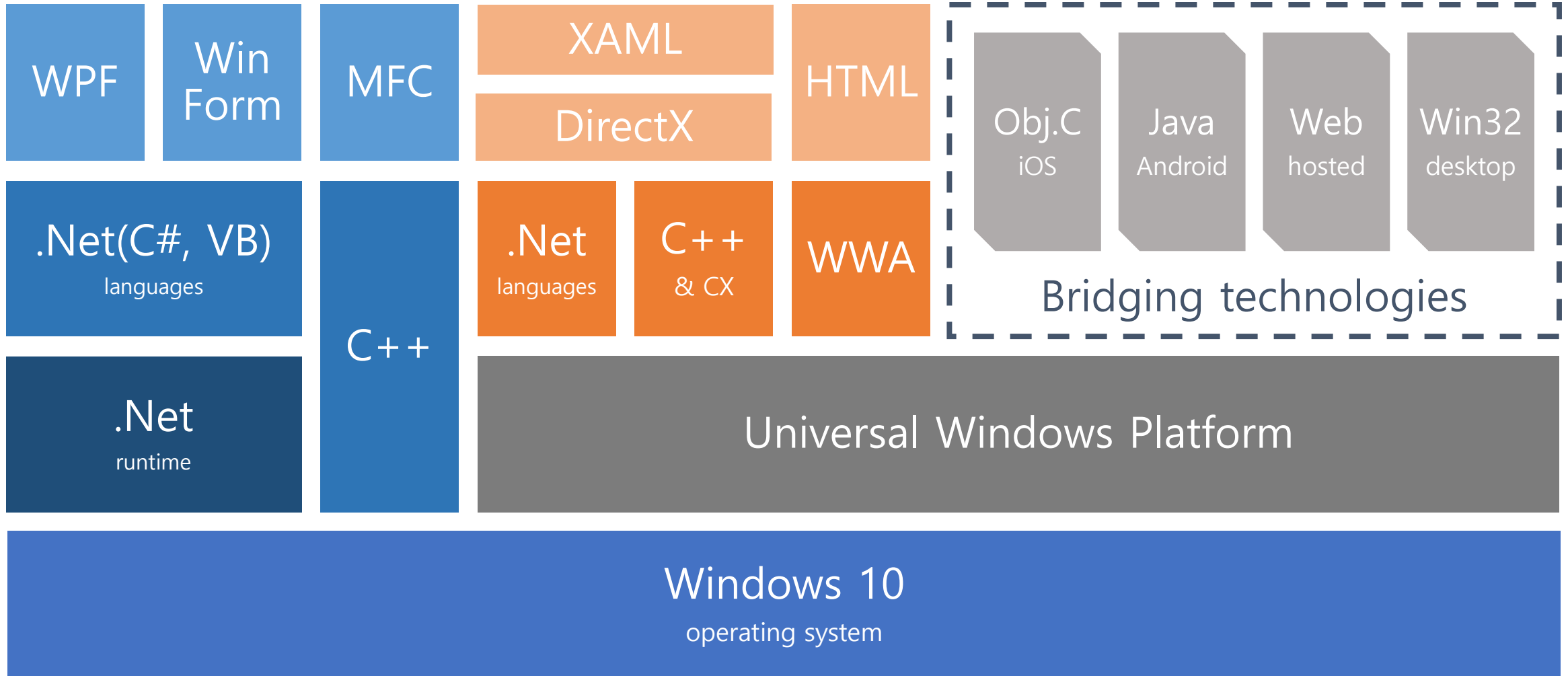
Natural
Input



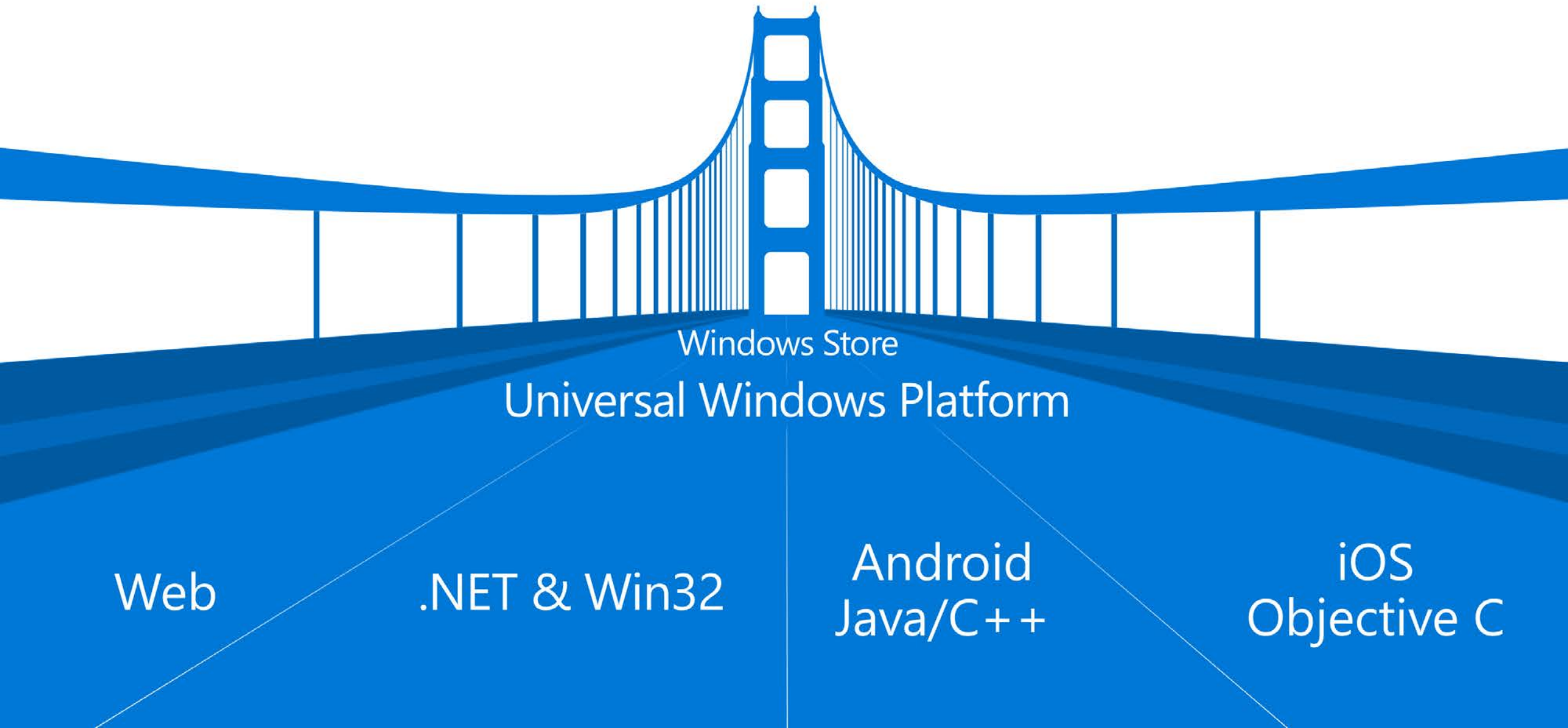
Reuse Existing Code



One Universal Windows Platform



1 Billion Windows 10 Devices



Windows Store

Universal Windows Platform

Web

.NET & Win32

Android
Java/C++

iOS
Objective C

Cross Platform



Cross Platform
MOBILE APP DEVELOPMENT



Cross Platform Development and C++

모바일 크로스 플랫폼

이전: Ship an app on Windows

지금: Ship four apps on Windows, OS X, Android, and iOS

Visual Studio에서 달라진 C++

How do we support modern application development in C++?

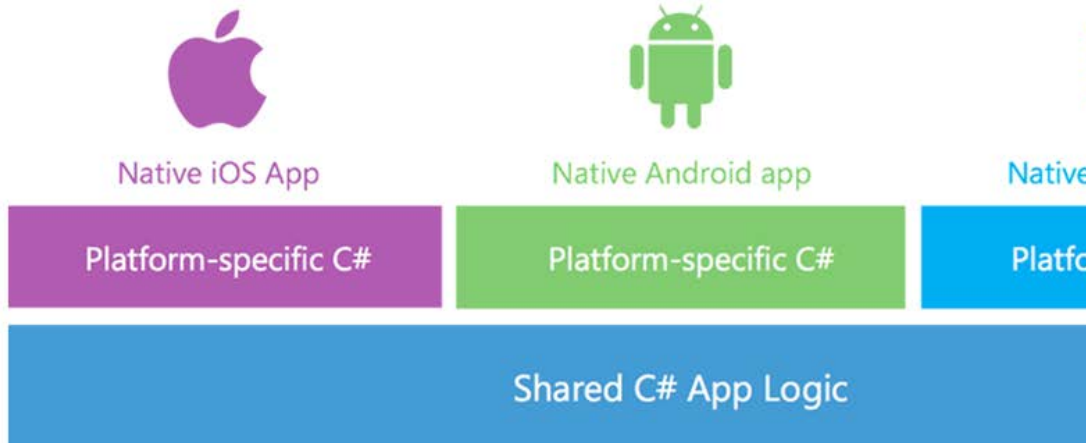
이전: Visual C++ (c1xx, c2) for targeting Windows, WP, Xbox

지금: Visual C++ (c1xx, c2) for targeting Windows, WP, Xbox
and Clang & LLVM for targeting Android and iOS





APACHE
CORDOVA™



Building Cordova Apps for Windows 10

Polita Paulus
Principal PM Lead



The Silo Approach

Build App
Multiple Times



iOS App

Objective-C
XCode



Android App

Java
Eclipse



Windows App

C#
Visual Studio

Xamarin's Unique Approach



Native With
Code Sharing

iOS C# UI

Android C# UI

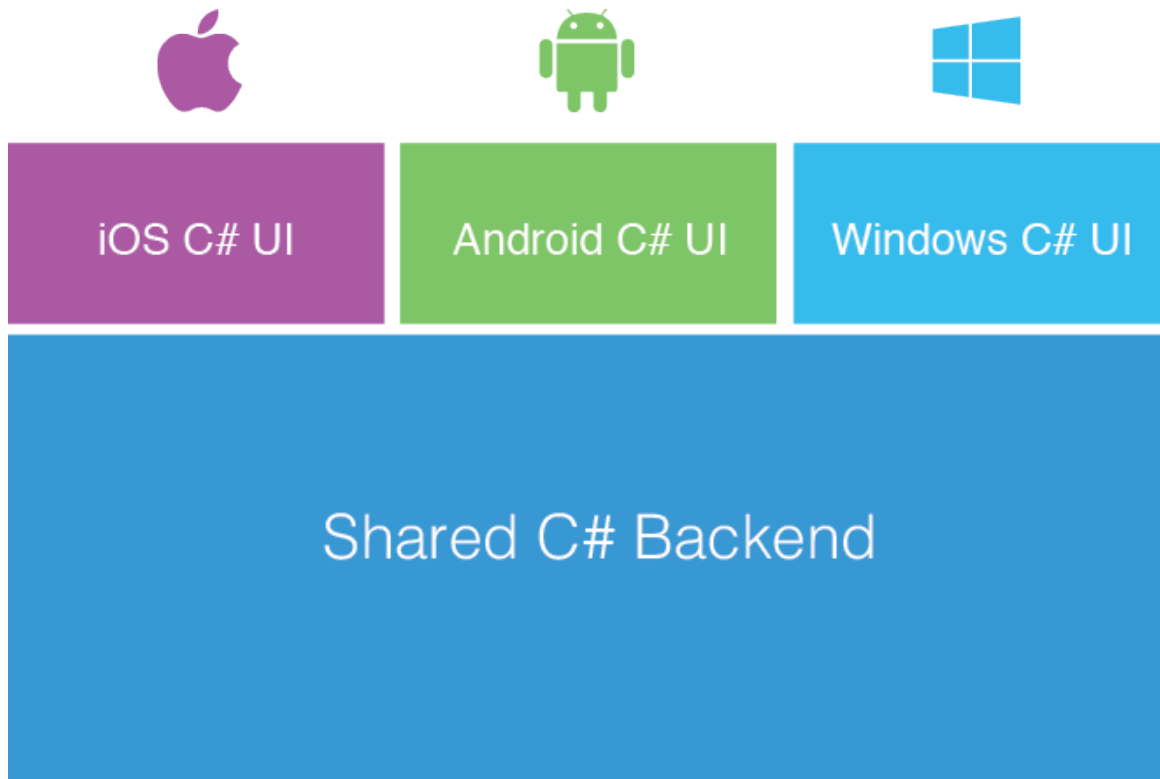
Windows C# UI

Shared C# Backend

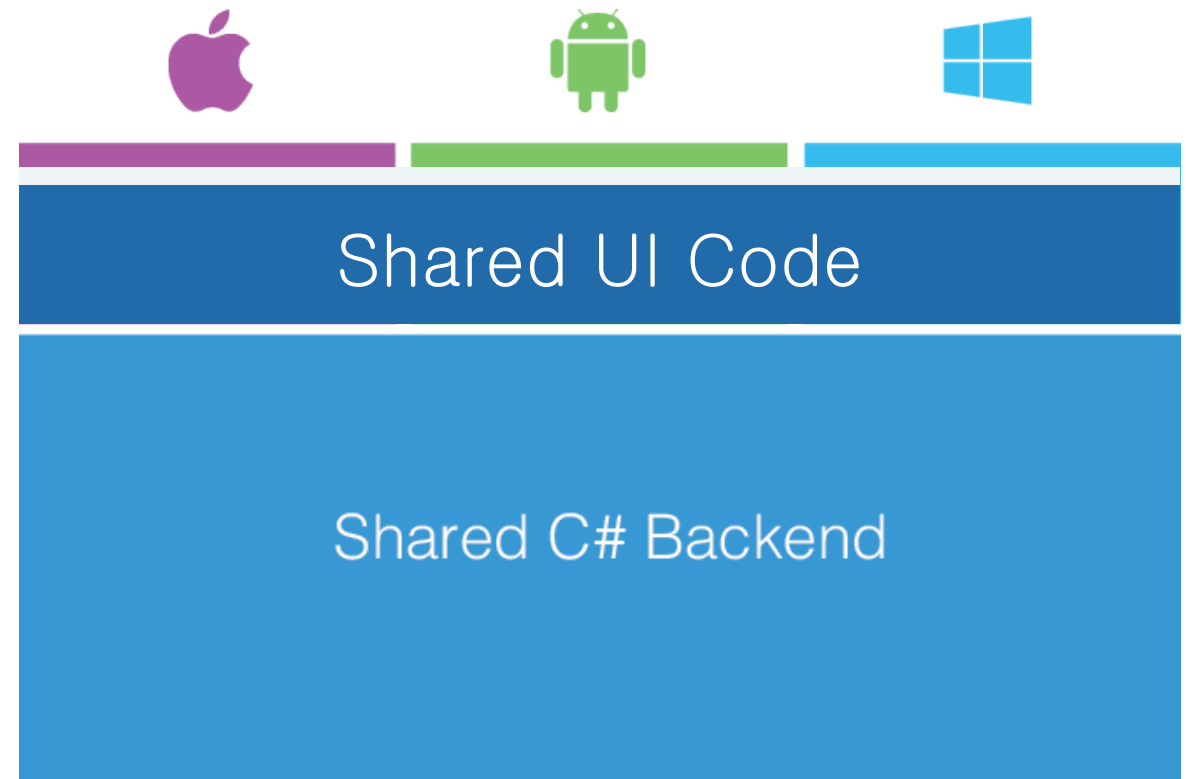
Xamarin + Xamarin.Forms



Traditional Xamarin approach



With Xamarin.Forms:
more code-sharing, native controls





Microsoft Azure Mobile Services accelerates connected client application development by streamlining common backend tasks like structuring storage, authenticating users, and sending push notifications.

Key Scenarios

Rapid Development

Time is money. Get your app up and running sooner when you use Mobile Services to configure a secure backend in less than five minutes.

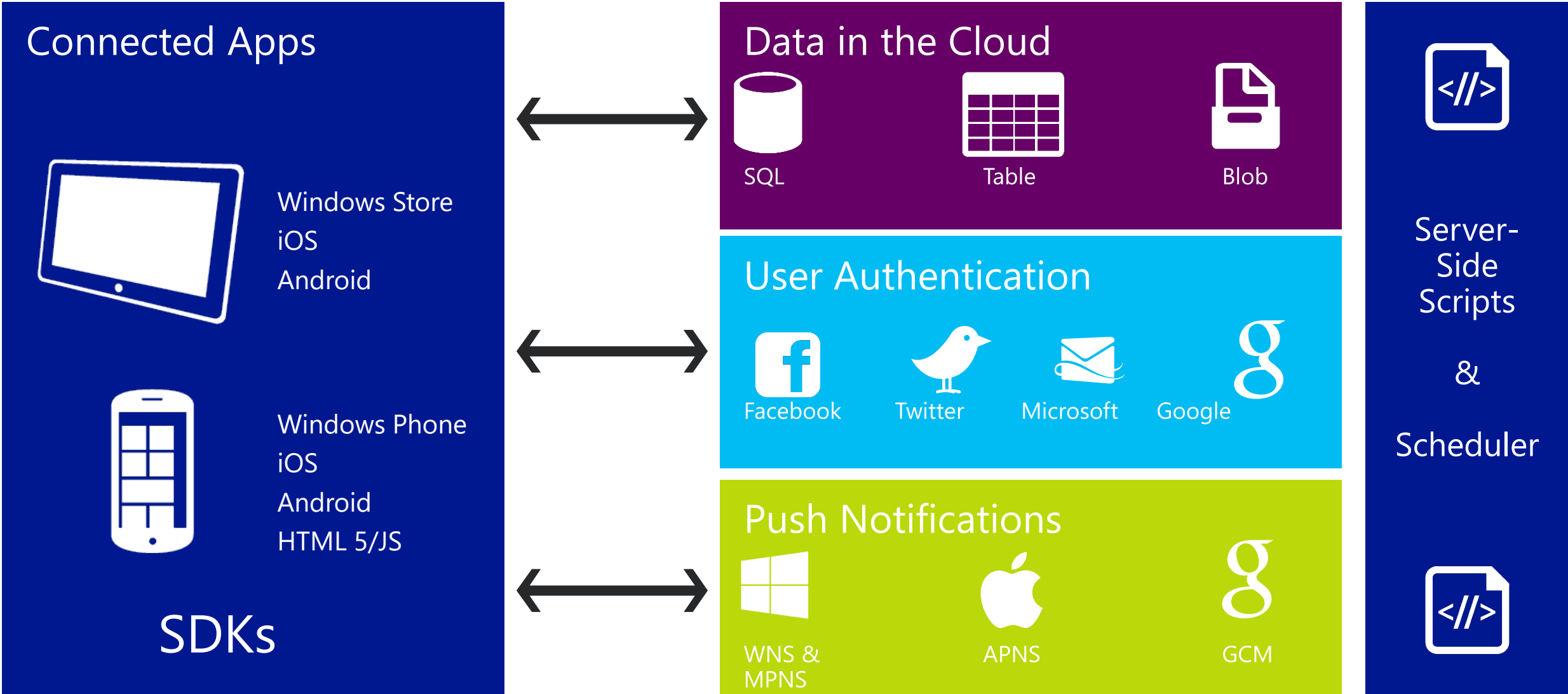
Make your app social

Whether your customers use Google, Facebook, Twitter, or Microsoft Account, no matter what devices they run your app on, with Mobile Services you can make your app social and personable fast.

Make your app engaging and dynamic

Push notifications and Live Tiles is the premier way to engage your customers. Make your app engaging and dynamic using Mobile Services Push.

Basic Features



Supported Platforms



Windows



iOS



Windows
Phone



Android



HTML5 &
Javascript



Windows Azure

Management Portal



Dashboard

■ Quick glance

- Displays relevant mobile service information, including the service URL, status, location, database and subscription

■ Usage Timeline:

- Displays a scalable chart of CPU time, API calls and data reads over a specific period of time.

■ Usage Overview:

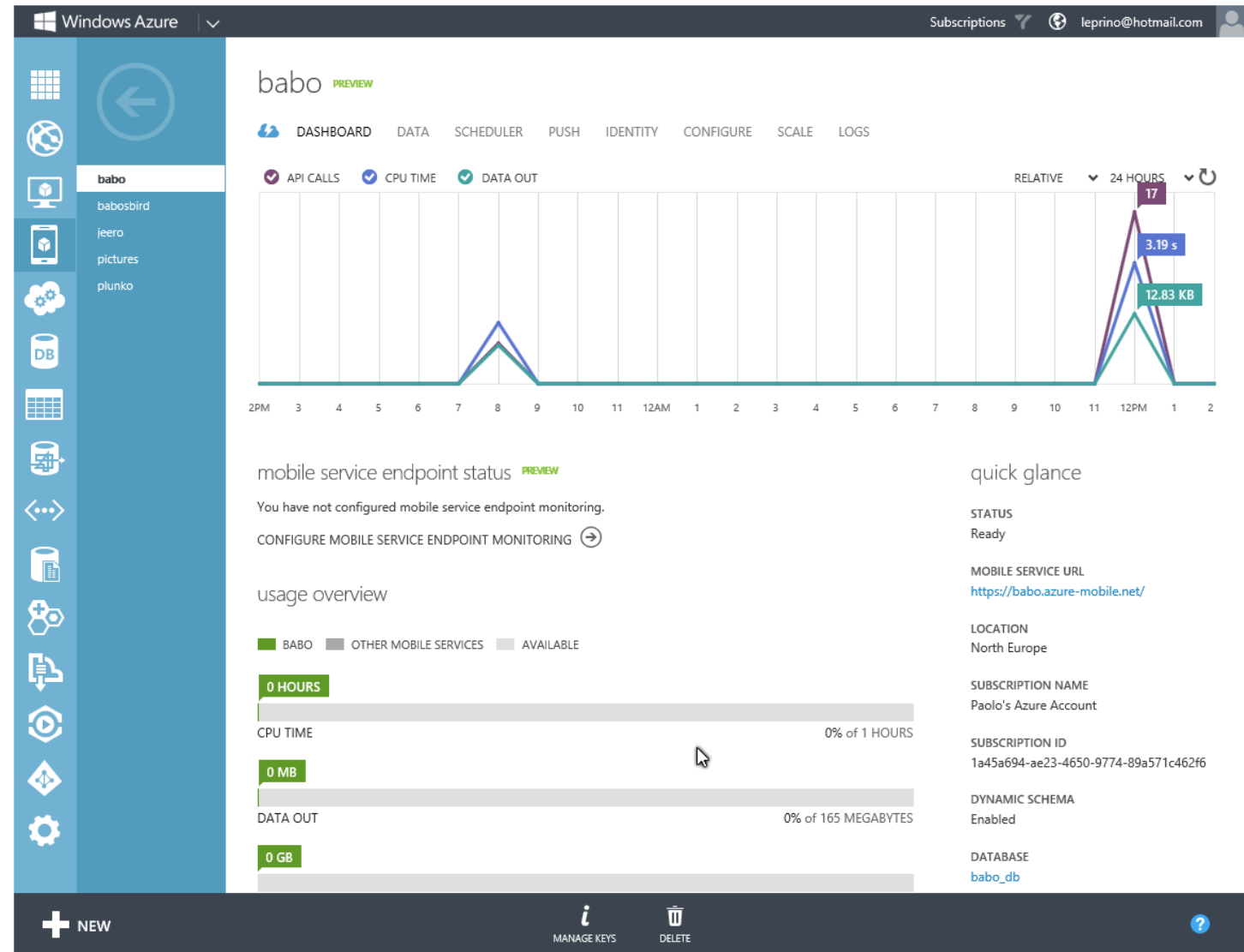
- Displays an overview of the total compute time and storage used by this mobile service

■ Manage Keys

- Manage Keys

■ Delete

- Delete the Mobile Service



Data

- In Microsoft Azure Mobile Services, data is stored in tables, which are maintained in the Microsoft Azure SQL Database that is associated with your mobile service.
- Click the Create button to add a new table to your mobile service.
- By default, you don't need to predefine the schema of tables in your database.
- Mobile Services automatically add columns to a table based on the data you insert. To change this dynamic schema behavior, use the Dynamic Schema setting on the Configure tab.
- To browse data, manage permissions and indexes on tables, or add server scripts, click on a table in the list.
- To delete a table, click to select a table row, and then click Delete.

The screenshot shows the Microsoft Azure Mobile Services interface for a service named 'babo'. The interface includes a navigation menu on the left with icons for various actions like 'NEW', 'CREATE', and 'DELETE'. The main content area displays a table with the following data:

TABLE	INDEXES	RECORDS
Channel	1	3
Counts	1	1
TodoItem	2	8

The 'Channel' row is highlighted in blue, indicating it is selected. The interface also shows navigation tabs for 'DASHBOARD', 'DATA', 'SCHEDULER', 'PUSH', 'IDENTITY', 'CONFIGURE', 'SCALE', and 'LOGS'. The user's email 'leprino@hotmail.com' is visible in the top right corner.

Browse

- The **Browse** tab lets you browse data rows in a selected table.
- Click the forward and back arrows to navigate through multiple pages of data.
- Click on table names in the left navigation bar to switch between tables.
- Click on the large left arrow to go back to the **Data** tab.

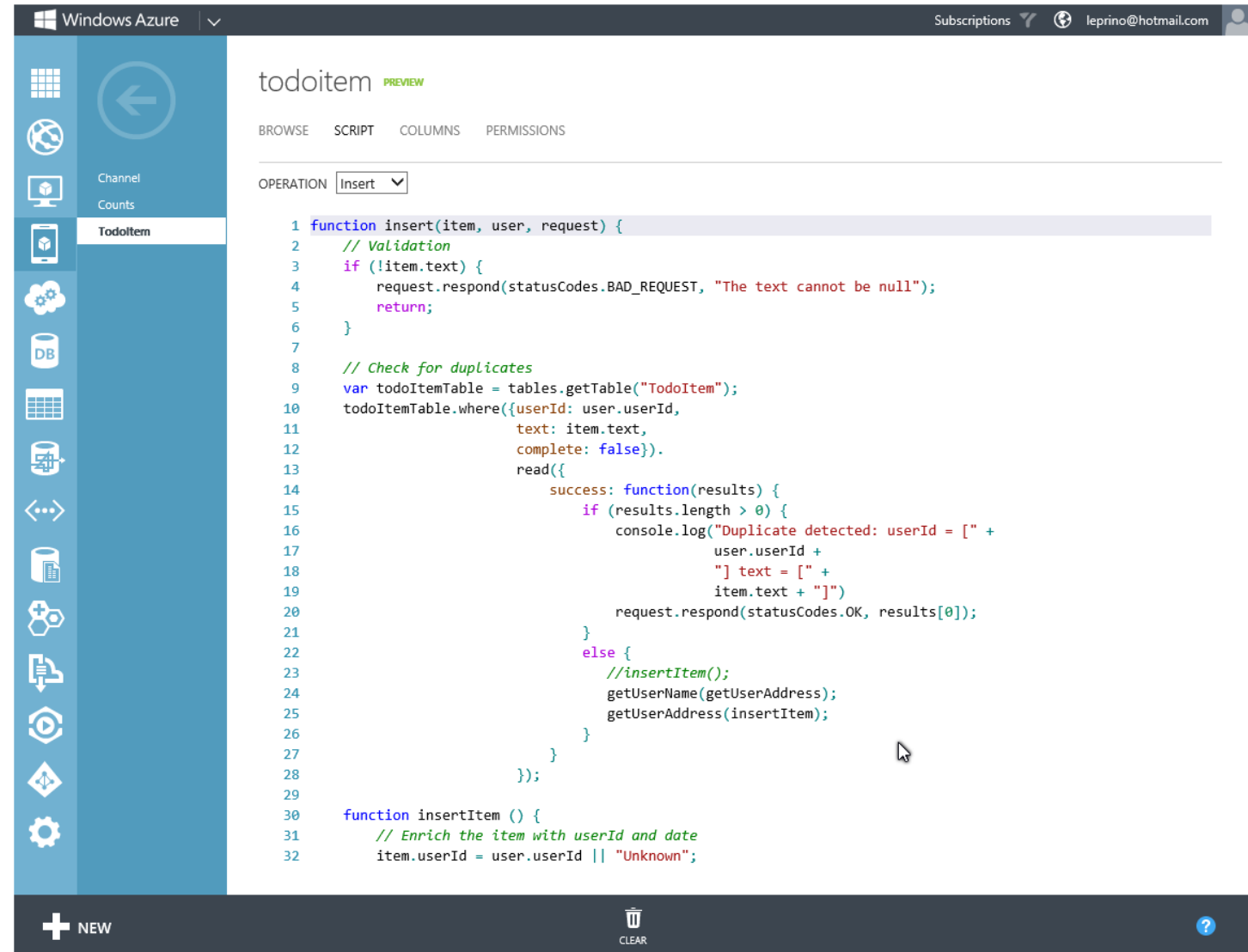
The screenshot shows the Windows Azure portal interface. The top navigation bar includes 'Windows Azure', 'Subscriptions', and the user profile 'leprino@hotmail.com'. The main content area is titled 'todoitem' with a 'PREVIEW' label. Below the title are tabs for 'BROWSE', 'SCRIPT', 'COLUMNS', and 'PERMISSIONS'. The 'BROWSE' tab is active, displaying a table with the following data:

id	text	complete	userId	createdAt	userName
2	Go to dinner	false	MicrosoftAccountbeb1f1ccd...	2013-05-10T17:06:45.596+...	Paolo Salvatori
3	Ciao	true	MicrosoftAccountbeb1f1ccd...	2013-05-10T17:08:38.826+...	Paolo Salvatori
5	Vado!	true	MicrosoftAccountbeb1f1ccd...	2013-05-10T17:12:49.691+...	Paolo Salvatori
6	Go to Moscow	false	MicrosoftAccountbeb1f1ccd...	2013-05-14T07:10:48.142+...	<unknown>
7	Go to the cinema	false	MicrosoftAccountbeb1f1ccd...	2013-05-21T13:29:51.765+...	<unknown>
8	Go to lunch	false	MicrosoftAccountbeb1f1ccd...	2013-05-23T10:43:49.833+...	<unknown>
9	Go to Pisa	false	MicrosoftAccountbeb1f1ccd...	2013-05-23T10:50:01.745+...	<unknown>
10	Go to Lucca	false	MicrosoftAccountbeb1f1ccd...	2013-05-23T10:52:49.892+...	<unknown>

The left navigation bar contains various icons for navigation, including a large left arrow at the top. The bottom of the interface features a dark bar with 'NEW', 'REFRESH', 'TRUNCATE', and a help icon.

Script

- Mobile Services enables you to register JavaScript code to be executed by the service when an insert, update, delete, or read operation occurs against a specific table.
- For each table there are 4 CRUD operations (insert, read, update, del)
- These scripts provide the ability to inject business logic into table operation, stop an operation from taking place, change or enrich values supplied to the operation or change the results.
- For authenticated clients, you can also perform user-specific authorizations based on the **userId** value of the supplied **user** object.



Windows Azure | Subscriptions | leprino@hotmail.com

todoitem PREVIEW

BROWSE SCRIPT COLUMNS PERMISSIONS

OPERATION

```
1 function insert(item, user, request) {
2   // Validation
3   if (!item.text) {
4     request.respond(statusCodes.BAD_REQUEST, "The text cannot be null");
5     return;
6   }
7
8   // Check for duplicates
9   var todoItemTable = tables.getTable("TodoItem");
10  todoItemTable.where({userId: user.userId,
11                      text: item.text,
12                      complete: false}).
13  read({
14    success: function(results) {
15      if (results.length > 0) {
16        console.log("Duplicate detected: userId = [" +
17                    user.userId +
18                    "] text = [" +
19                    item.text + "]");
20        request.respond(statusCodes.OK, results[0]);
21      }
22    }
23    //insertItem();
24    getUserAddress(insertItem);
25  });
26
27  });
28
29
30 function insertItem () {
31   // Enrich the item with userId and date
32   item.userId = user.userId || "Unknown";
```

+ NEW CLEAR ?

Columns

- To delete columns, click the column and then click **Delete**.
- To add additional columns to the table, simply send an insert request including the new properties from your app with dynamic schema enabled.
- Once a column is created, its data type cannot be changed by Mobile Services.
- Insert or update operations fail when the type of a property in the JSON object cannot be converted to the type of the equivalent column in the table.
- You can improve the performance of queries by defining an index on the table.
- To add a column to the index, select the column and click **Set as Index**.

Windows Azure | Subscriptions | leprino@hotmail.com

todoitem PREVIEW

BROWSE SCRIPT **COLUMNS** PERMISSIONS

COLUMN NAME	TYPE	INDEX
id	bigint(MSSQL)	✓ Indexed
text	string	
complete	boolean	
userid	string	
createdAt	date	✓ Indexed
userName	string	
userAddress	string	

+ NEW | DELETE | SET INDEX | ?

Permissions

- **Mobile Services** enables you to set the following permissions on table operations:
 - **Everyone:** request for the operation against the table is accepted. This option leaves your data wide-open for everyone to access.
 - **Anybody with the Application Key:** The application key is required to perform the operation. The application key is distributed with the application. **Note:** Because this key is not securely distributed, it cannot be considered a security token. To secure access to your mobile service data, you must instead authenticate users before accessing.
 - **Only Authenticated Users:** Only authenticated users are permitted to perform the operation. Scripts can be used to further restrict access to tables based on an authenticated user.
 - **Only Scripts and Admins:** The operation requires the service master key, which limits the operation only to registered scripts or to administrator accounts

The screenshot displays the Windows Azure portal interface for configuring permissions on a table named 'todoitem'. The top navigation bar includes 'Windows Azure', 'Subscriptions', and the user profile 'leprino@hotmail.com'. The main content area is titled 'todoitem PREVIEW' and has tabs for 'BROWSE', 'SCRIPT', 'COLUMNS', and 'PERMISSIONS'. The 'PERMISSIONS' tab is active, showing a section for 'table permissions'. Below this, four permission settings are listed, each with a dropdown menu set to 'Only Authenticated Users':

- INSERT PERMISSION: Only Authenticated Users
- UPDATE PERMISSION: Only Authenticated Users
- DELETE PERMISSION: Only Authenticated Users
- READ PERMISSION: Only Authenticated Users

The left sidebar contains various service icons, with 'Todoitem' selected. The bottom of the page features a '+ NEW' button and a help icon.

Scheduler

- Mobile Services enables you to register JavaScript code to be executed by the service either on a schedule that you define or on-demand from the Management Portal.
 - Removing duplicate records from a table.
 - Backing up a database.
 - Send a report via email.
- When your mobile service runs in free mode, you can create only one scheduled job at a time; in reserved mode, you can create up to ten scheduled jobs at the same time.

The screenshot shows the Windows Azure Management Portal interface for a mobile service named 'babo'. The top navigation bar includes 'DASHBOARD', 'DATA', 'SCHEDULER', 'PUSH', 'IDENTITY', 'CONFIGURE', 'SCALE', and 'LOGS'. The 'SCHEDULER' tab is active, displaying a table with the following columns: JOB, STATUS, LAST RUN, NEXT RUN, and FREQUENCY. A single job named 'twitterCount' is listed with a status of 'Disabled', a last run time of 'Tue Apr 16 2013, 6:30:00 PM', a next run time of 'Never', and a frequency of 'Every 15 minutes'. A modal dialog titled 'MOBILE SERVICES: SCHEDULER' is open, showing the 'Create new job' form. The 'JOB NAME' field contains 'BackupDatabase'. Under the 'SCHEDULE' section, the 'Every' radio button is selected, with '15' in the frequency dropdown and 'minutes' in the unit dropdown. The 'On demand' radio button is unselected. A note at the bottom of the dialog states 'You can add a schedule later.' The bottom of the portal features a dark bar with icons for '+ NEW', '+ CREATE', '▶ ENABLE', and '🗑 DELETE', along with a help icon.

JOB	STATUS	LAST RUN	NEXT RUN	FREQUENCY
twitterCount	Disabled	Tue Apr 16 2013, 6:30:00 PM	Never	Every 15 minutes

MOBILE SERVICES: SCHEDULER

Create new job

JOB NAME
BackupDatabase

SCHEDULE

Every 15 minutes

On demand

You can add a schedule later.

Push

- Mobile Services can send push notifications to your apps.
- The way that you configure and send push notifications depends on your app.
- You can send push notifications to the following applications:
 - **Windows Store app** by using the **Windows Push Notification Service (WNS)**. To send push notifications to your app, you must configure your mobile service to work with WNS.
 - **iPhone and iPad apps** by using the **Apple Push Notification Service (APNS)**. To send push notifications to your app, you must configure your mobile service to work with APNS. Do this by generating a special push notification certificate at the iOS Provisioning Portal
 - **Android:** You can send push notifications to Android apps by using the **Google Cloud Messaging (GCM)** service. To send push notifications to your app, you must configure your mobile service to work with GCM.
 - **Windows Phone app** by using the **Microsoft Push Notification Service (MPNS)**. Mobile Services does not require you to register your Windows Phone 8 app to send push notifications.

The screenshot shows the Windows Azure portal interface for a mobile service named 'babo'. The top navigation bar includes 'DASHBOARD', 'DATA', 'SCHEDULER', 'PUSH', 'IDENTITY', 'CONFIGURE', 'SCALE', and 'LOGS'. The 'PUSH' tab is selected. The main content area is divided into sections for configuring different notification services:

- windows application credentials:**
 - CLIENT SECRET:** vmcyH0yqfdjXtkN4vZ3ez4+ibmARmeKc
 - PACKAGE SID:** ms-app://s-1-15-2-749284573-2737771755-300233!
- apple push notification settings:**
 - CERTIFICATE:** [No certificate present] with an 'Upload' button.
 - FEEDBACK SCRIPT:** EDIT SCRIPT (with a right arrow icon)
- google cloud messaging settings:**
 - API KEY:** (empty text input field)

The left sidebar shows a navigation menu with icons for various services and a list of applications: 'babo', 'babosbird', 'jeero', 'pictures', and 'plunko'. The bottom of the page features a '+ NEW' button and a help icon.

Identity

- Mobile Services integrates with the following identity providers to make it easy to authenticate users from your app:
 - Microsoft Account
 - Facebook login
 - Twitter login
 - Google login
- To enable authentication, you must first register your app with one or more of these identity providers and then configure your mobile service.
- You must also add authentication code to your app.
- When a user login is authenticated by Mobile Services, the value of the **userId** property on the **user** object passed to server scripts is set to a value that uniquely identifies a user. This value can then be used to authorize access to data.

Windows Azure | Subscriptions | leprino@hotmail.com

babo PREVIEW

DASHBOARD DATA SCHEDULER PUSH **IDENTITY** CONFIGURE SCALE LOGS

microsoft account settings

CLIENT ID

CLIENT SECRET

facebook settings

APP ID/API KEY

APP SECRET

twitter settings

CONSUMER KEY

CONSUMER SECRET

+ NEW

Configure

- You can directly access to the database used by your mobile service just clicking the link.
- You can change the database used by your mobile service to store data with a new or existing DB by clicking Change DB and select one of the following options.
- Enable/disable dynamic schema.
- In reserved mode, you can configure endpoint monitoring.

The screenshot shows the Windows Azure portal interface for configuring a mobile service named 'babo'. The top navigation bar includes 'Windows Azure', 'Subscriptions', and the user email 'leprino@hotmail.com'. The left sidebar contains a navigation menu with icons for various services and a list of mobile services: 'babo', 'babosbird', 'jeero', 'pictures', and 'plunko'. The main content area is titled 'babo PREVIEW' and features a navigation menu with 'DASHBOARD', 'DATA', 'SCHEDULER', 'PUSH', 'IDENTITY', 'CONFIGURE', 'SCALE', and 'LOGS'. The 'CONFIGURE' page is divided into sections: 'database settings', 'dynamic schema', 'cross-origin resource sharing (cors)', and 'monitoring'. In the 'database settings' section, 'SQL DATABASE' is set to 'babo_db' and 'SQL SERVER' is set to 'tqv6ywwuzm'. The 'dynamic schema' section has 'ENABLE DYNAMIC SCHEMA' set to 'ON'. The 'cross-origin resource sharing (cors)' section shows 'ALLOW REQUESTS FROM HOST NAMES' with a list containing 'localhost' and 'uppy', and an input field for 'HOST NAME'. The bottom navigation bar includes 'NEW', 'CHANGE DATABASE', 'MANAGE KEYS', 'DELETE', and a help icon.

Scale

- On the Scale tab, you can scale your mobile service by switching to reserved mode and adding role instances.
- You can also scale the Microsoft Azure SQL Database instance used by the service.
 - **Note:** When you set the Mobile service mode to Reserved, all of your mobile services in the same region run in Reserved mode.
- To estimate the cost required to scale your mobile service, see the Mobile Services Pricing Calculator.

The screenshot displays the Windows Azure portal interface for a mobile service named 'babo'. The top navigation bar includes 'Windows Azure', 'Subscriptions', and the user email 'leprino@hotmail.com'. The left sidebar contains various service icons, with 'babo' selected. The main content area shows the 'SCALE' tab for the 'babo' service, which is in 'PREVIEW' status. The 'general' section includes a 'MOBILE SERVICE MODE' selector with 'FREE' and 'RESERVED' options, and a 'reserved capacity (disabled)' section. The 'INSTANCE SIZE' is set to 'Small (1 core, 1.75 GB Memory)'. The 'INSTANCE COUNT' is set to 1, with a legend showing 'BABO' (dark grey) and 'AVAILABLE' (light grey). The 'sql database' section shows 'BABO_DB SQL DATABASE' with 'WEB' and 'BUSINESS' options, and a '1 GB' dropdown menu. The bottom of the page features a '+ NEW' button and a help icon.

Log

- Mobile Services writes system and script errors to the log.
- Click on a log entry to view details and copy to the clipboard.
- Click the link in the Source column to directly jump to the script that generated the error.
 - **Note:** Existing log entries can be viewed in the Management Portal for 7 days.
- Your registered server scripts can also write information directly to the log by using the **console** object.

The screenshot displays the Windows Azure Management Portal interface for a service named 'babo'. The top navigation bar includes 'DASHBOARD', 'DATA', 'SCHEDULER', 'PUSH', 'IDENTITY', 'CONFIGURE', 'SCALE', and 'LOGS'. The 'LOGS' tab is active, showing a table of log entries. The table has four columns: LEVEL, MESSAGE, SOURCE, and TIME STAMP. The log entries include information messages (e.g., 'Sent message: [object Object]', 'Successfully inserted: { text: 'Go to Lucca', ...}') and error messages (e.g., 'Error in script \'/table/TodoItem.insert.js'. Er...'). The SOURCE column for all entries is 'TodoItem/insert'. The TIME STAMP column shows dates from May 23, 2013, ranging from 12:50:01 PM to 12:52:53 PM. A sidebar on the left contains navigation icons and a list of services: 'babo', 'babosbird', 'jeero', 'pictures', and 'plunko'. At the bottom of the interface, there are buttons for '+ NEW', 'REFRESH', 'DETAILS', and a help icon.

LEVEL	MESSAGE	SOURCE	TIME STAMP
Information	Sent message: [object Object]	TodoItem/insert	Thu May 23 2013, 12:52:53 PM
Information	{ "s:Envelope": { "s": { "xmlns:s": "http://schem...	TodoItem/insert	Thu May 23 2013, 12:52:50 PM
Information	Body: <s:Envelope xmlns:s="http://schema...	TodoItem/insert	Thu May 23 2013, 12:52:50 PM
Information	Sent push: { statusCode: 200, deviceConne...	TodoItem/insert	Thu May 23 2013, 12:52:50 PM
Error	Error in script \'/table/TodoItem.insert.js'. Er...	TodoItem/insert	Thu May 23 2013, 12:52:50 PM
Error	Error in script \'/table/TodoItem.insert.js'. Er...	TodoItem/insert	Thu May 23 2013, 12:52:50 PM
Information	Sent push: { headers: { 'x-wns-notifications...	TodoItem/insert	Thu May 23 2013, 12:52:50 PM
Information	Sent push: { statusCode: 200, deviceConne...	TodoItem/insert	Thu May 23 2013, 12:52:49 PM
Information	Successfully inserted: { text: 'Go to Lucca', ...	TodoItem/insert	Thu May 23 2013, 12:52:49 PM
Information	About to insert: { text: 'Go to Lucca', compl...	TodoItem/insert	Thu May 23 2013, 12:52:49 PM
Information	{ "s:Envelope": { "s": { "xmlns:s": "http://schem...	TodoItem/insert	Thu May 23 2013, 12:52:49 PM
Information	Body: <s:Envelope xmlns:s="http://schema...	TodoItem/insert	Thu May 23 2013, 12:52:49 PM
Information	{ "s:Envelope": { "s": { "xmlns:s": "http://schem...	TodoItem/insert	Thu May 23 2013, 12:50:02 PM
Information	Body: <s:Envelope xmlns:s="http://schema...	TodoItem/insert	Thu May 23 2013, 12:50:02 PM
Error	An error ocured creating/accessing the S...	TodoItem/insert	Thu May 23 2013, 12:50:02 PM
Information	Sent push: { statusCode: 200, deviceConne...	TodoItem/insert	Thu May 23 2013, 12:50:02 PM
Information	Sent push: { statusCode: 200, deviceConne...	TodoItem/insert	Thu May 23 2013, 12:50:02 PM
Error	Error in script \'/table/TodoItem.insert.js'. Er...	TodoItem/insert	Thu May 23 2013, 12:50:02 PM
Error	Error in script \'/table/TodoItem.insert.js'. Er...	TodoItem/insert	Thu May 23 2013, 12:50:02 PM
Information	Sent push: { headers: { 'x-wns-notifications...	TodoItem/insert	Thu May 23 2013, 12:50:01 PM

DEMO

Azure Mobile Service



References

- Microsoft Azure Mobile Services

- <http://www.windowsazure.com/en-us/develop/mobile/>

- Tutorials and Resources

- <http://www.windowsazure.com/en-us/develop/mobile/resources-html/>

- Upload File to Microsoft Azure Blob Storage using Microsoft Azure Mobile Services

- <http://code.msdn.microsoft.com/Upload-File-to-Windows-c9169190>

- Getting user information on Azure Mobile Services

- <http://blogs.msdn.com/b/carlosfigueira/archive/2012/10/25/getting-user-information-on-azure-mobile-services.aspx>

- Troubleshooting authentication issues in Azure Mobile Services

- <http://blogs.msdn.com/b/carlosfigueira/archive/2012/10/23/troubleshooting-authentication-issues-in-azure-mobile-services.aspx>

References

- Using the scheduler to backup your Mobile Service database
 - http://www.thejoyofcode.com/Using_the_scheduler_to_backup_your_Mobile_Service_database.aspx
- Getting started with the CLI and backing up your scripts
 - http://www.thejoyofcode.com/Getting_started_with_the_CLI_and_backing_up_your_scripts_Day_4_.aspx
- More CLI – changing your Mobile Services workflow
 - http://www.thejoyofcode.com/More_CLI_ndash_changing_your_Mobile_Services_workflow_Day_5_.aspx
- New Microsoft Azure Mobile Services Getting Started Content
 - <http://blogs.msdn.com/b/windowsazure/archive/2013/01/09/new-windows-azure-mobile-services-getting-started-content.aspx>
- How to use Microsoft Azure Table Storage in Microsoft Azure Mobile Services
 - <http://code.msdn.microsoft.com/windowsazure/Use-Table-Storage-in-0b6bcbc7>

References

- How to Use the Table Service from Node.js
 - <http://www.windowsazure.com/en-us/develop/nodejs/how-to-guides/table-services/>
- How to Use Service Bus Queues
 - <http://www.windowsazure.com/en-us/develop/nodejs/how-to-guides/service-bus-queues/>
- How to Use Service Bus Topics/Subscriptions
 - <http://www.windowsazure.com/en-us/develop/nodejs/how-to-guides/service-bus-topics/>
- How to Send Email Using SendGrid from Node.js
 - <http://www.windowsazure.com/en-us/develop/nodejs/how-to-guides/sendgrid-email-service/>

