

A portrait photograph of Glenn Stephens, a man with short brown hair and a slight smile, wearing a dark t-shirt. He is positioned on the left side of the slide, overlaid on a blue diagonal band.

Data Caching and Synchronization

Glenn Stephens
glen.stephens@xamarin.com

 **Xamarin**
University

Agenda

1. The problem with Offline Data
2. Connecting Remote and Local Data
3. Existing Frameworks



Xamarin University

Phones lose connectivity

Xamarin University



No connection



Flight mode



Terrible Network

Offline data needs

- Regardless we need access to the data for our application
- We need access to remote servers
- We need local storage
- We need a mechanism to sync the offline data to the cloud data



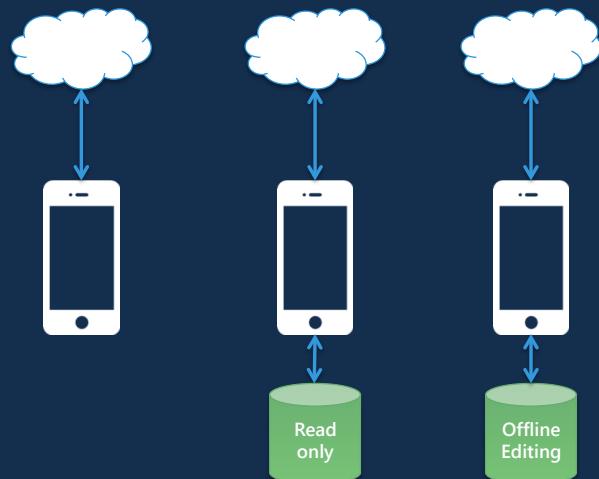
Xamarin University

Disconnection is the rule,
not the exception

Xamarin University

Types of Connected Mobile Apps

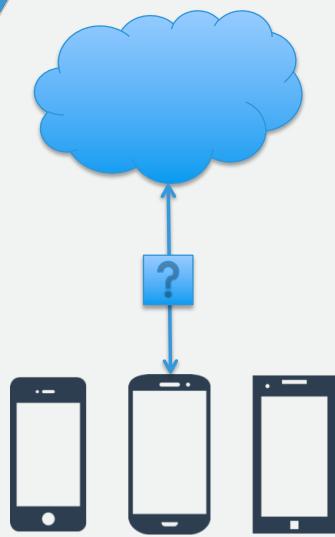
- Online only
- Online with Read-only offline
- Online with offline-editing



Xamarin University

Detecting offline mode

- You can never truly detect when you will be offline



Xamarin University

Detecting offline status - iOS

Reachability.cs is available from [Xamarin Samples](#)

```
public bool CanConnect (string host)
{
    return Reachability.IsChecked (host);
}
```

Xamarin University

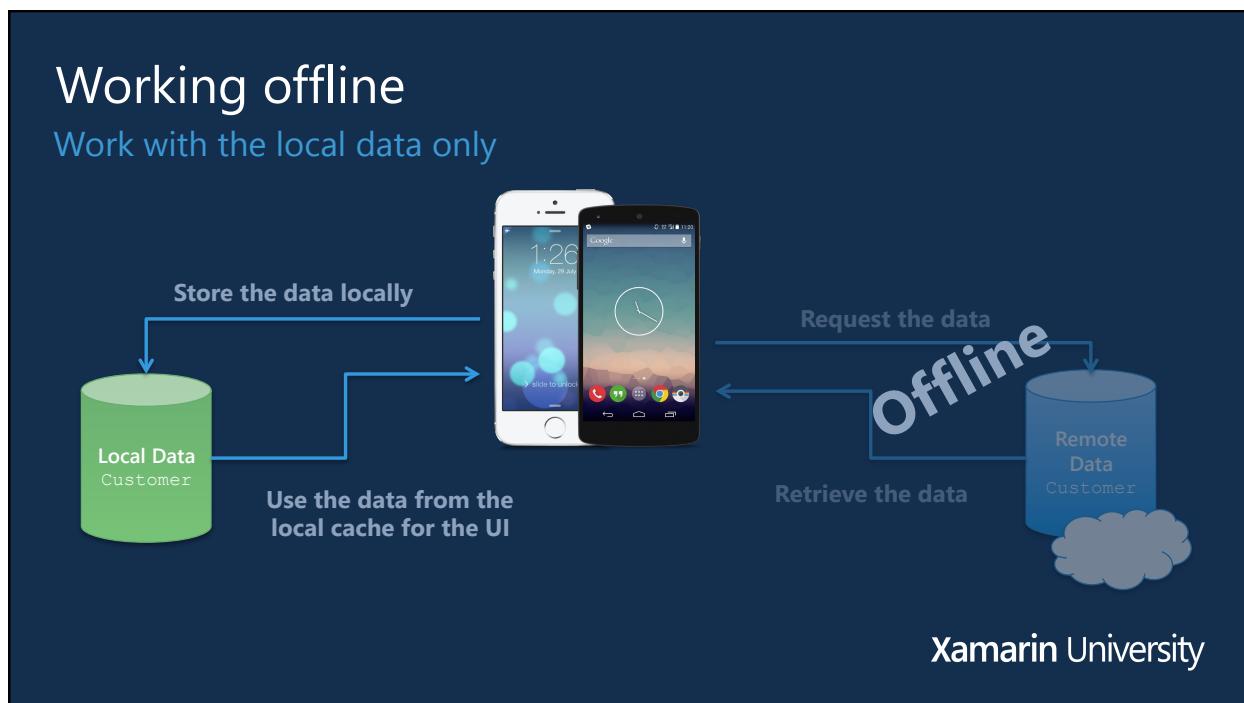
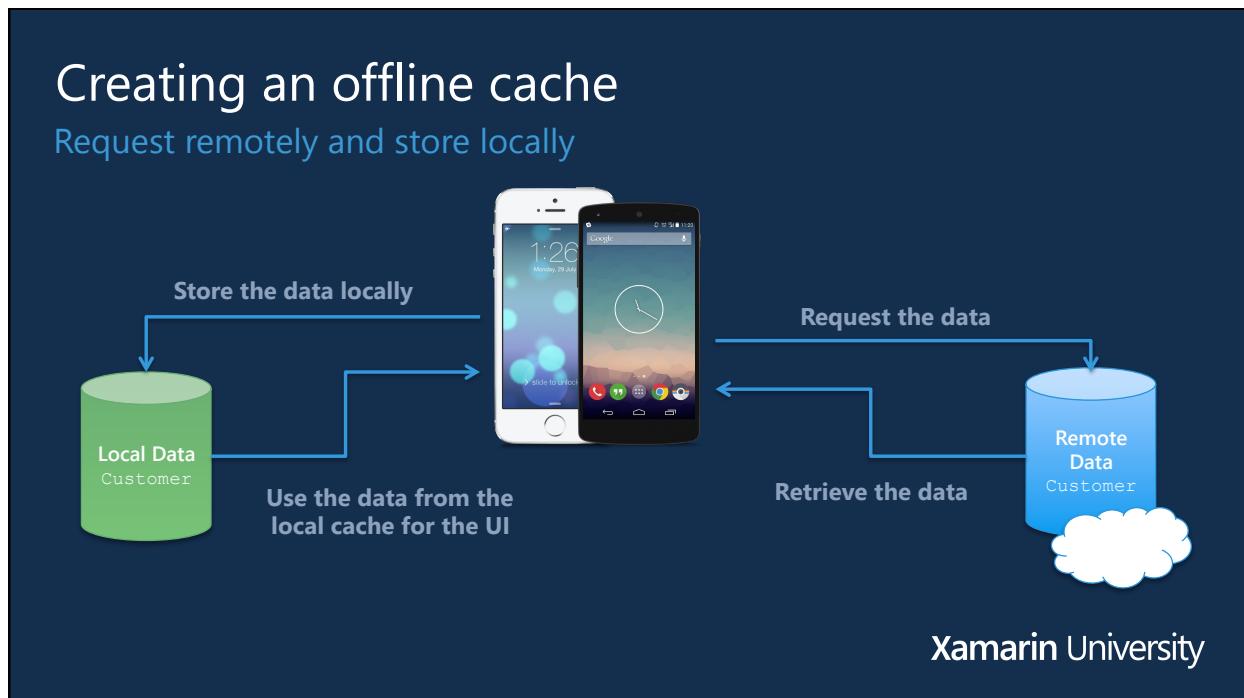
Detecting offline status - Android

Use the [ConnectivityManager](#) class

```
public bool CanConnect (string address)
{
    var mgr = Application.Context.GetSystemService(
        Context.ConnectivityService)
        as ConnectivityManager;
    var netInfo = mgr.ActiveNetworkInfo;

    return netInfo != null && netInfo.IsConnectedOrConnecting;
}
```

Xamarin University





Structuring your sync

- For offline access, define a sync class that you can call.
- Call it when required



Xamarin University

Xamarin University

Sync when you're not running

iOS7 can use
Background Fetch to
push changes and
download caches

Performing a Fetch

Retrieving new results

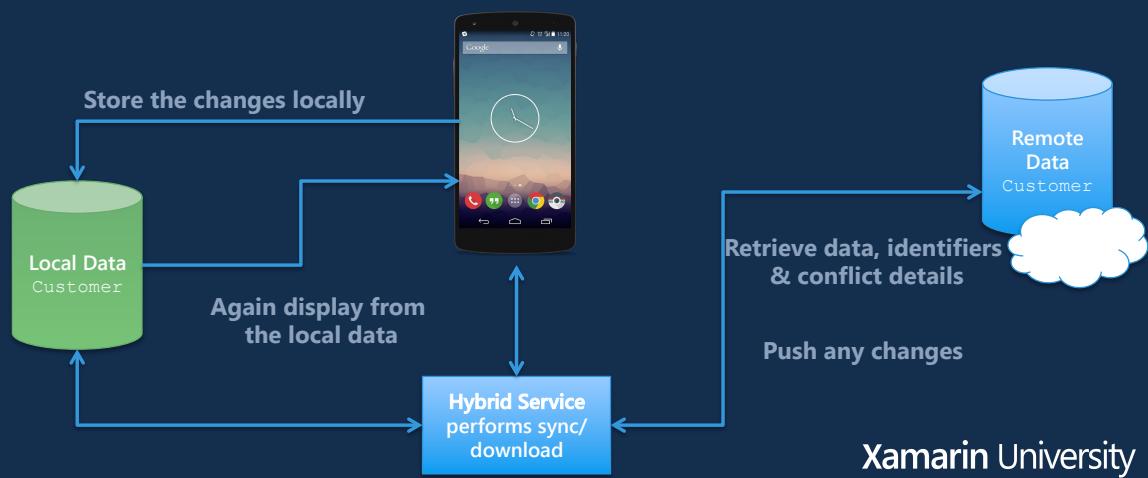
```
public async override Task PerformFetch (UIApplication application,
    Action<UIBackgroundFetchResult> completionHandler)
{
    try
    {
        // Perform the Sync
        var hasMoreData = await PerformSync();

        if (hasMoreData)
            completionHandler(UIBackgroundFetchResult.NewData);
        else
            completionHandler(UIBackgroundFetchResult.NoData);
    } catch {
        completionHandler(UIBackgroundFetchResult.Failed);
    }
}
```

Xamarin University

Android Service

Synchronising when your app is not running



Existing Frameworks

Xamarin University

Existing Tools and Frameworks

- Azure Mobile Services (beta)
- Couchbase
- EasyNetQ
- Zumero (SQL Server syncing)
- Mobilink



Xamarin University

Demo

Xamarin University

What system should I use?

- What's your starting point?
- What's your operating environment?
- What's your data co-ordination



Xamarin University

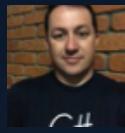
Summary

- Caching and Data Synchronization is coordinating your existing knowledge of REST/Web/Remote Services and your knowledge of local storage
- Plenty of available frameworks for Greenfield projects
- Have fun

Xamarin University



Data Caching and
Synchronizing



Glenn Stephens
glen.stephens@xamarin.com

