Real-Time Collaborative Apps with Realm

Nikola Irinchev

realm

Agenda

- Intro to the Realm Mobile Database
- Getting started with the Realm Mobile Platform
- [Demo] Deploying on Azure
- [Demo] Implementing a messaging app

Realm Mobile Database

- On-device cross-platform **object** database
- Launched July 2014
- Developed in the open, fully open source
- 19K+ GitHub stars, 100K+ active developers

realm

How does it differ from SQLite?

Realm is an object database

- Realm is built from scratch and optimized specifically for mobile
- There is no translation layer - the objects are the database
- Realm is fast and therefore easy on the battery.

How do I use it?

Models

```
public class Dog : RealmObject
{
    public string Name { get; set; }
    public int Age { get; set; }
}
```

realm

Use **LINQ** to perform complex queries across your object graph

```
// Fluent or Extension Syntax
var oldDogs = realm.All<Dog>().Where(d => d.Age > 8);

// RealmResult collection conforms to the IQueryable
foreach (var d in oldDogs)
{
    Debug.WriteLine(d.Name);
}
```

realm

Mutate data in transactions

```
realm.Write(() =>
{
    realm.Add(new Dog { Name = "Rex", Age = 1 });
    realm.Add(new Dog { Name = "Fido", Age = 2 });
});
```

realm

Relations work like they should

```
public class Dog : RealmObject
{
    public string Name { get; set; }
    public int Age { get; set; }
}

public class Person : RealmObject
{
    public string Name { get; set; }
    public IList<Dog> Dogs { get; }
}
```

Notice that there is no ID property on Dog

realm

Liveness and reactiveness

Objects and queries represent the current state of the database. You don't need to manually refresh.

Realm objects **live-update** in response to changes

```
var puppies = realm.All<Dog>().Where(d => d.Age < 2);

puppies.Count(); // => 0 because no dogs have been added yet

// Update and persist objects with a transaction
realm.Write(() =>
{
    realm.Add(new Dog
    {
        Name = "Rex",
        Age = 1
    });
});

// Queries are updated in realtime
puppies.Count(); // => 1
```

realm

Databinding

All RealmObject's implement INotifyPropertyChanged.
Lists and query results implement INotifyCollectionChanged.

These work cross-thread and cross processes.

TwoWay databinding

All RealmObject's implement IReflectableType to handle properties being set by the binding engine

Threading

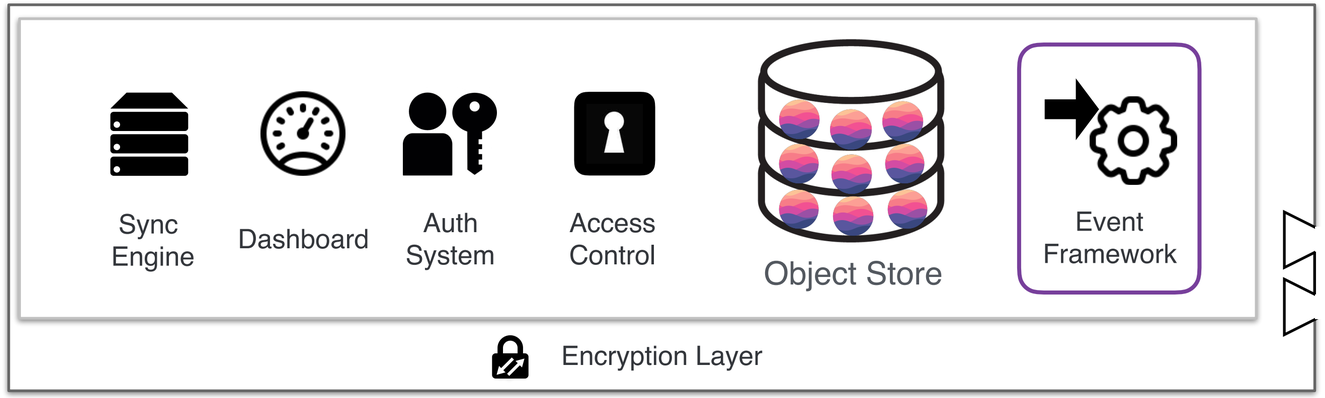
```
public class Dog : RealmObject
{
    [PrimaryKey]
    public string Id { get; set; } = Guid.NewGuid().ToString();

    public string Name { get; set; }
    public int Age { get; set; }
}

string id = null;
realm.Write(() =>
{
    var dog = realm.Add(new Dog { Name = "Rex", Age = 1 });
    id = dog.Id;
});

Task.Run(() =>
{
    var realm = Realm.GetInstance();
    var dog = realm.Find<Dog>(id);
    // compute something really expensive here
});
```

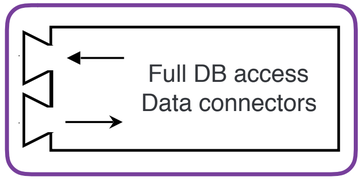
realm

Realm Mobile Platform

What is it?

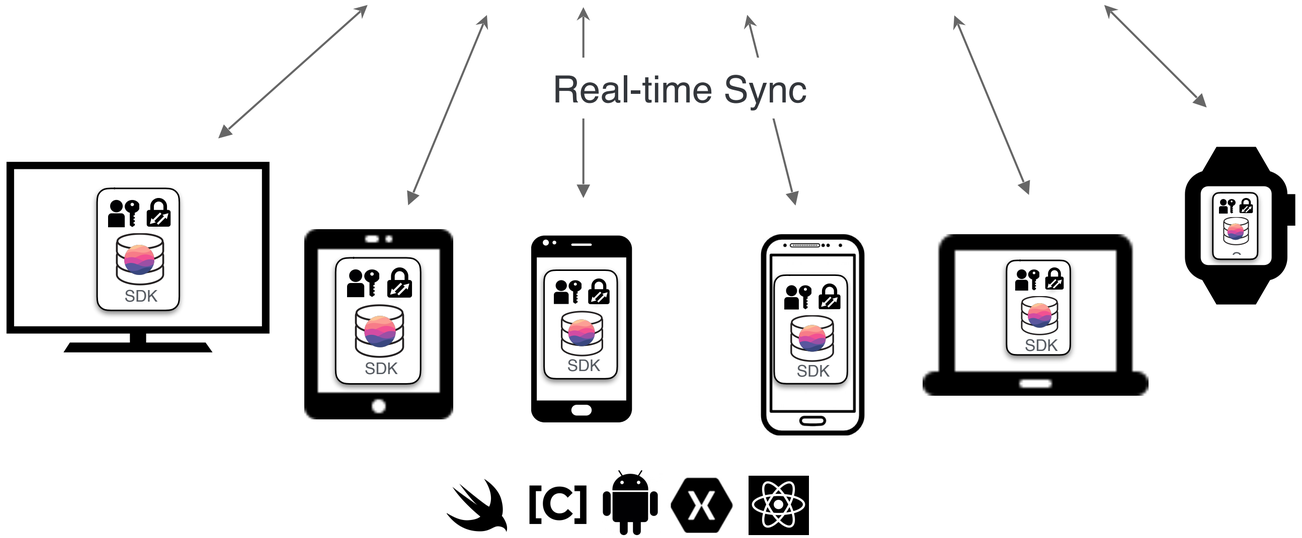
Realm Object Server NEW





Enterprise Feature

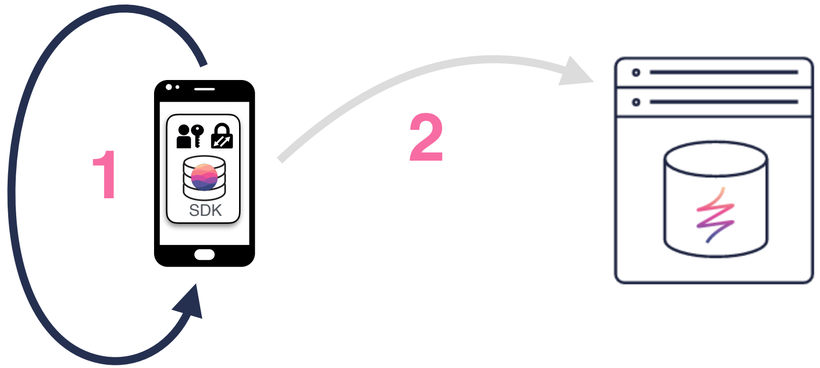
Realm Mobile Database



realm

Eliminate Network Challenges

- Offline-first
- Full database on the device
  - Queries run locally
  - Writes apply immediately
- Reactive architecture keeps UI up-to-date
- Automatic sync happens in the background
- Resilient to any network conditions



realm

Open your realm with sync

```
var credentials = Credentials.UsernamePassword(username, password, false);

var authURL = new Uri("http://my.realm-auth-server.com:9080");
var user = await User.LoginAsync(credentials, authURL);

var serverURL = new Uri("realm://my.realm-server.com/~/default");
var configuration = new SyncConfiguration(user, serverURL);

var realm = Realm.GetInstance(syncConfiguration);
```

Demo

Tips and key takeaways

- Use multiple realms in an app
- Make the most of databinding support
- Back the smallest shareable unit with a single realm

Questions?

Nikola Irinchev
nikola.irinchev@realm.io
@nirinchev

realm