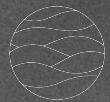


BUILD BETTER APPS, FASTER WITH REALM

An Overview of the Realm Platform



realm

CONTENTS

3 Introduction

4 Conceptual Foundations of Realm

5 Components of the Realm Platform

6 Key Features of the Realm Platform

8 How the Realm Platform Works

9 Implementing the Realm Platform

10 Key Mobile Use Cases

15 Ready for Enterprise

Introduction

Mobile apps today have to work harder than ever before. Consumers and enterprise users alike demand top performance from all apps at all times. Users also depend on mobile apps to power their personal mobility, allowing them connect and do more, anytime, anywhere. To meet these needs, apps are becoming more and more sophisticated. For example, today's top apps are able to respond immediately to updates, continue to function while offline, enable realtime interaction with other users, and support a seamless cross-platform experience. In addition, many enterprise apps connect to legacy back-end systems in order to share data and enable business-critical use cases.

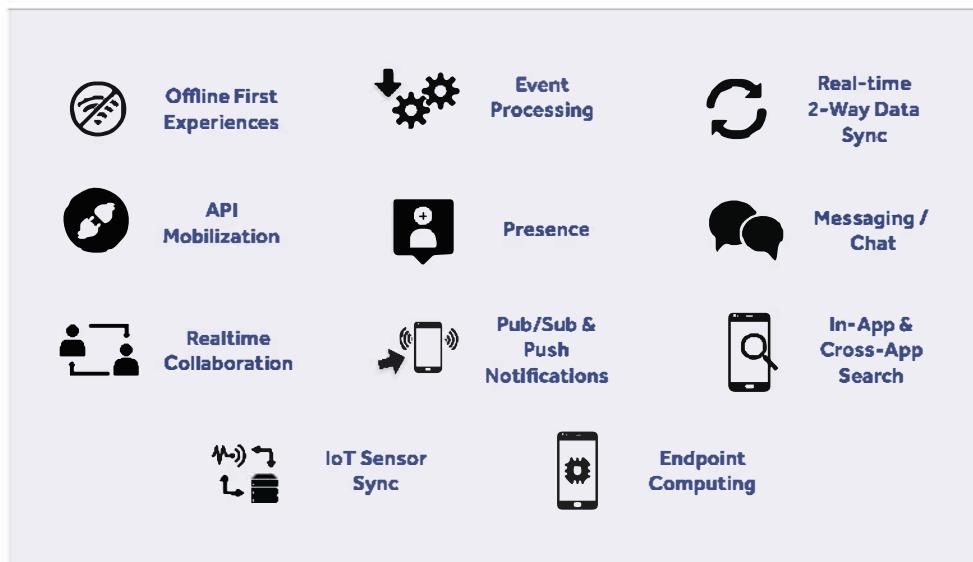
To build such high-performance apps, developers are also working harder than ever. Many engineering organizations are all too familiar with the pain of increased development complexity that results in longer cycles, overburdened resources, and an unwieldy code base. For their stakeholders, slow delivery can put the business at risk in the marketplace.

Realm was founded to help developers build today's advanced mobile experiences with greater speed and efficiency. Our set of tools and technologies are designed to make it easy for your team to build powerful, data-driven features that would otherwise be difficult using traditional approaches. The Realm Platform brings together our client-side Realm Database and server-side Realm Object Server to deliver automated, realtime data synchronization, conflict resolution, and event handling — key ingredients for today's reactive apps.

The Realm Platform simplifies application architecture and greatly reduces complexity for developers, allowing them to bypass more time-consuming and cumbersome tasks, such as writing and maintaining networking code. The platform also solves some of the thornier problems facing IT organizations, such as how to mobilize legacy APIs or how to connect multiple API sources, including third-party sources, and make them accessible to mobile app developers using Java or Swift.

Your development team is likely building mobile apps with advanced data handling capabilities, such as realtime collaboration, messaging, offline experiences, or legacy API integration. The Realm Platform offers them the fastest path to building reactive apps that can keep pace with your users and competitors.

Hard Features Made Easy



CONCEPTUAL FOUNDATIONS OF REALM

When Realm launched in 2014, our goal was to help mobile developers build better apps faster by giving them a powerful data management alternative to SQLite and Core Data. As such, we take a fundamentally different approach to data storage and synchronization, and we provide a much faster, richer data experience than standard models.

Realm: A New Kind of Database

A "Realm" is a lightweight object container that acts like a database, yet has a few key differences. Like with any other database, data in Realms can be queried, filtered, interconnected, and persisted. Unlike a conventional database, data is represented as objects that are fully reactive, enabling safe access across threads and making seamless synchronization possible. A Realm can contain multiple types of objects, each with their own enforced schema. Your application can use multiple Realms with different permissions for different users. Local Realms are an exact copy of the remote Realms, and vice versa.

Client-Side, Embedded Data Storage

Realm removes the latency inherent in the traditional database model that depends heavily on API calls over mobile networks to keep data up to date. Instead, Realm embeds a full-featured database on device for fast access and maximum efficiency, even in the case of very large data sets. Queries need only fractions of a second to complete, which provides a smooth, highly-responsive user experience. Client-side data storage also supports offline-first use cases, enabling your app to continue to function offline, then sync data later when network connectivity is available.

Server-Side Business Logic

Realm builds an event handling framework into the server, making it easy to create triggers that are tied directly to mobile data changes. The framework includes a global listener that can be configured to track object changes from the device side and then instantly run server code in response. This allows you to run sophisticated business logic against your device-to-device interactions and

device-to-server integrations. Moreover, running logic on the server means you can more easily manage updates from one centralized place.

Cross-Platform, Object Database (Not an ORM)

Realm stores data as native objects in a format that is easily consumed by a mobile app using a language binding of the developer's choice. This means that you do not need to write additional bandwidth-consuming code to serialize or deserialize data, or deal with complex object relational mapping. With the traditional model, the database administrator sets the data model in the back-end database. With Realm's simple object-oriented structure, the developer sets the data model in code, which effectively means that your database is your data model. In addition, Realm is cross-platform, so the same data layer can be shared across all your apps running on any major mobile platform.

Live Objects

At the heart of Realm's approach is the concept of "live objects." Normally you would have to build your app to make frequent calls that check for, and fetch, data updates in order to make sure that you have the latest version of the data. However, with Realm, every object is "live," meaning that the variable is always the correct representation of the underlying data. Realm attaches "listeners" to data that react (or file callback methods) when that data changes. Realm automatically initiates data synchronization and updates the object in realtime. Data sync is fast and efficient — only incremental or additive changes are transmitted, not entire objects. The entire data flow is handled seamlessly by Realm, with no additional networking code needed. With

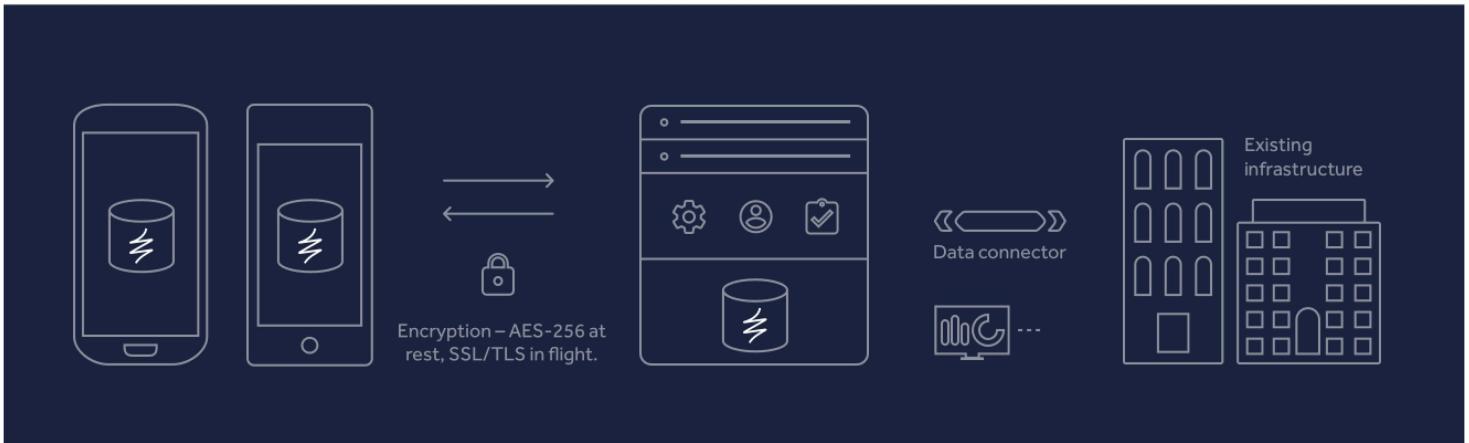
live objects, your users always have the latest data at hand and will appreciate a smooth, highly performant app experience.

Reactive Architecture

With Realm's unique architecture model, multiversion concurrency control provides concurrent access to the Realm Database across threads. This means that data readers are never blocked against a data writer, and each has a consistent view of the database, thus ensuring Realm's ACID compliance. A notification system provides updates when a write transaction completes, enabling you to write logic that is triggered by such changes and supports a reactive user experience.

Open Source Roots

Realm is grounded by a philosophy of openness. Our products are developed in the open, and we share many of our processes and projects on [GitHub](#). The Realm Database is fully open source and distributed under the Apache 2.0 license. Our engineering teams are dedicated to improving and maintaining our many language bindings and providing a high level of technical support. A community of thousands of developers actively participate with us on GitHub and contribute their own tools, code, and docs, as well as technical support. To date, Realm repos have earned 33k+ stars and included 4K+ forks. In addition, hundreds of community-built [add-ons](#) help Realm developers do more with our products.



COMPONENTS OF THE REALM PLATFORM

Realm Database

Embedded on the client, the [Realm Database](#) is a full-featured, object-oriented, cross-platform database that persists data locally on device. It's available for major mobile languages, such as Swift and Objective-C (iOS), Java (Android), C# (Xamarin, .NET), and JavaScript (React Native and Node.js). See our [current language support](#) for more information. The Realm Database is lightweight and highly performant, capable of handling very large data loads and running queries in fractions of a second. Based on shared live objects, it syncs data seamlessly in realtime with the Realm Object Server without the need to write networking, serialization, or object-relational mapping code. This means that your app will be able to refresh data as fast as needed to provide an enjoyable, engaging user experience.

Realm Object Server

Realm's unified data model extends to the [Realm Object Server](#), which mirrors the Realm Database on device. It functions as a middleware component in the mobile app architecture and manages data synchronization, event handling, and integration with legacy systems. The Realm Object Server can efficiently and simultaneously sync data across 1m+ devices and automatically resolve conflicts — all in realtime. Moreover, it provides a single place to manage all communications, including legacy API transactions, that may otherwise be subject to mobile network latency and other issues. Realm Object Server can be deployed on-premises or in the public cloud and is available as both a Linux and macOS install package.

Realm Node.js SDK

A key component of the Realm Object Server is the [Realm Node.js SDK](#). It enables developers to write logic that governs how data flows between apps and the Realm Object Server. Most importantly, it enables integration into the backend through various API connectors, thus establishing connections with legacy systems.

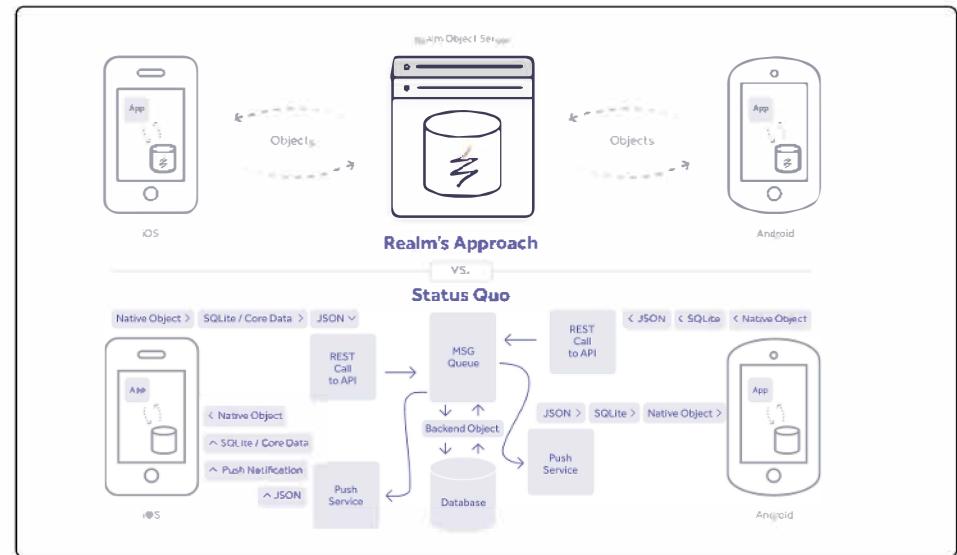
REALM FOCUSES ON EASING POINTS OF FRICTION ALONG THE FLOW OF DATA BETWEEN APP USERS AND THE BUSINESSES THAT SERVE THEM

KEY FEATURES

Realtime Data Synchronization

The core power of Realm lies in our reactive architecture that enables realtime data synchronization between data stored on the Realm Object Server and data stored on any one of your fleet of apps running on numerous mobile platforms. Once an update is detected, Realm only syncs the changes to the remote end for maximum speed and efficiency.

Example: A municipal transportation app needs to keep passengers informed of the latest updates to bus timetables. Using Realm's API, the service pushes new information stored in their corporate systems to the Realm Object Server, which updates the timetable data object. When a user opens the app, the timetable object stored in the Realm Database on device immediately detects that it is out of date and initiates the synchronization process with the Realm Object Server, which sends only the marginal changes. The entire update process happens automatically and feels instantaneous to the user.



Scalability

Depending on the app and hardware environment, a single Realm Object Server can handle well over ten thousand concurrent users. To reach massive scale, the Enterprise Edition makes it possible to deploy multiple instances of the Realm Object Server in parallel, so your app can support more than one million concurrent users and deliver the same realtime experience to every one. The server includes a built-in load balancer that automatically handles distributing connections.

Cross-Platform Data Model

Realm's approach ensures cross-platform compatibility across your entire app install base. With Realm's live object data model, the Realm Object Server and all user devices store the same data objects in the app's native language. All objects are listening for the same updates and will automatically sync the same data in the same way. This unified model means that your development team designs and builds your app's data model only once, and then simply rolls it out across your app variants.

Example: A media company needs to develop an Android version of an app that they have successfully deployed for iOS. Realm's cross-platform data model allows the Android team to develop the app knowing that they won't face the type of complexity that comes from converting CoreData activities into SQLite activities. Without Realm, this process would have required the Android team to request alterations to data types and models as specified by the system's APIs. Such platform specific customizations lead to divergence in the data model and slow development.

Event Handling

The Realm Object Server provides an event handling framework that can trigger server-side logic whenever an object changes. After data changes on the client side, it gets synced to the server, which then executes code to handle the change. Event handling also enables the Realm Object Server to trigger data flows between your app and legacy APIs and systems. You can choose to design your app around "serverless" computing principles and write discrete functions for each specific trigger.

Example: A mobile shopping app allows a user to enter a coupon code while filling out an order form. This is captured locally as a change to the client-side coupon object, which gets instantly synced to the server. This change signals the event handling framework to run validation checks, update the server-side coupon object with a response, then sync back down to the client-side coupon object to update the app UI. The entire process happens in realtime, ensuring that the user remains engaged with the order process.

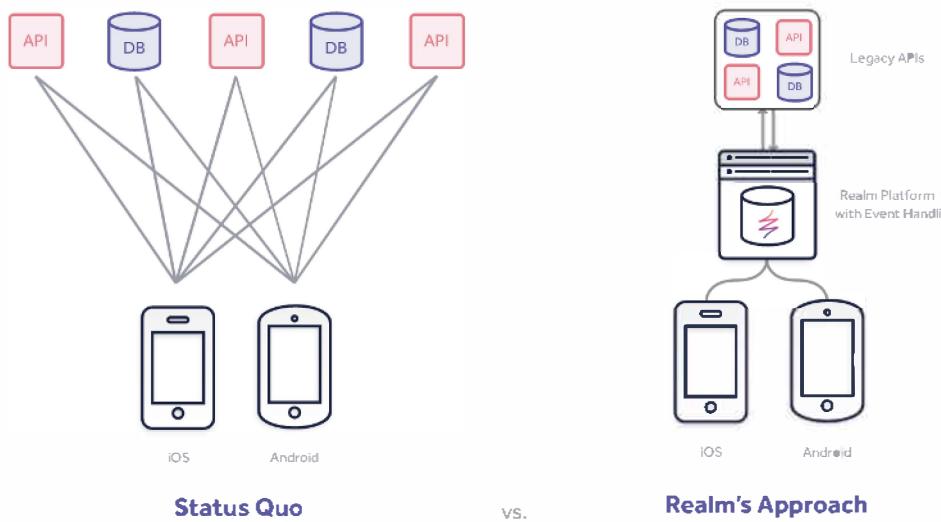
Continuous Backups

Realm provides failure recovery resiliency for your critical business data. The platform enables you to create automatic backups of your production data on the Realm Object Server and save it to a secondary server at the location of your choice. In the event of a power outage, hardware corruption, natural disaster, or similar event, you can continue to operate your application seamlessly from the secondary server.

Data Integration API

Using the Data Integration API, developers can easily connect Realm Object Server with any existing data sources or services, either internal or third-party, with no extra code needed on the client side. This allows you to surface core business data within the app experience and funnel user-generated data back into legacy systems of record. Our API makes it easy to build your own custom connectors or use one of our pre-built connectors, such as for PostgreSQL.

Example: A brick and mortar retailer wants to launch a mobile shopping app tied to an existing payment transaction database. Realm's Data Integration API connects the legacy database to the Realm Object Server, which is already synced with the app. With each transaction, the Realm Object Server manages seamless data synchronization between the app and the legacy database. To ensure accuracy between systems, the API is designed to identify broken connections and restart interrupted transactions, which then triggers the Realm Object Server to resolve any data conflicts.



Data Security and Compliance

Enterprise-grade security features, such as AES-256 encryption, ensure that your sensitive data is always kept secure. The Realm Platform uses industry standard data encryption at rest on device and server, and encryption in flight during data synchronization.

HOW IT WORKS

One of the most prominent components is the **Sync Engine**, which powers the realtime synchronization process between the server and client-side data stores.

When you install the Realm Object Server, you can log into a **Dashboard** to view metrics on server operations.

An **Authentication System** allows your users to identify themselves and can be integrated with your existing authentication systems.

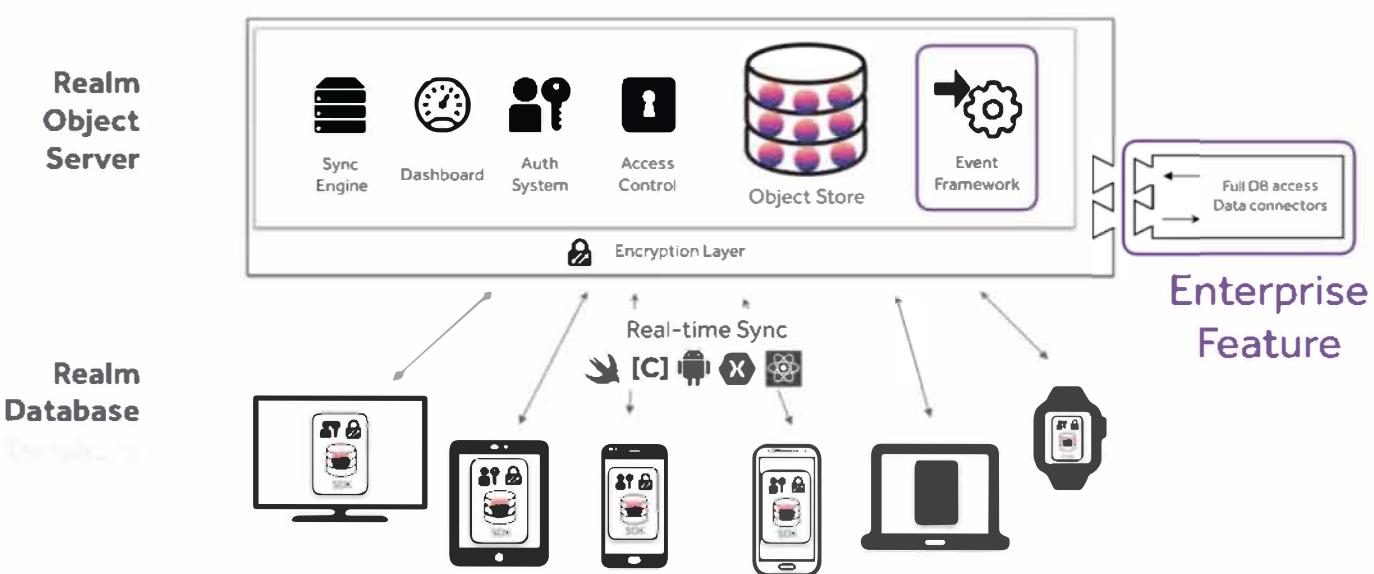
Once authenticated, various **Access Controls** can serve data to particular users, allowing you to have more fine-grained control over who gets what data in your app.

Another prominent component of the server is the **Object Store**. It is a common set of APIs that enables cross-platform compatible data storage. All data represented on the client devices is mirrored on the server in the exact same live object format.

The **Event Framework** allows you to create server side logic that responds to data coming in from the client.

Data connectors give you the ability to access all data server side and integrate it into existing infrastructure.

All data is enclosed in a **Layer of Encryption**, the services are locked down according to security best practices.



OVER 100K+ MOBILE DEVELOPERS TO DATE ACTIVELY USE REALM. ONE OF THE REASONS WHY REALM IS SO POPULAR WITH DEVELOPERS IS THAT IT IS EXCEPTIONALLY FLEXIBLE AND EASY TO USE. THE REALM MOBILE PLATFORM IS DESIGNED TO BE ADAPTABLE TO YOUR SPECIFIC ORGANIZATIONAL REQUIREMENTS.

IMPLEMENTING THE REALM PLATFORM

Installation Options

On the client side, Realm supports all major mobile platforms and provides a range of language variants to choose from. On the server side, Realm is available as both a Linux and macOS install package. Developer will typically use the macOS install for easy and convenient exploration or proof-of-concept projects, and they'll use the Linux install for long-term operation of their apps.

[See more examples of common architectures with the Realm Platform.](#)

Integrating into Your Systems Architecture

The Realm Platform can easily fit into your existing system architecture as a lightweight, modular component. Our Data Integration API allows you to connect the Realm Object Server to your APIs and backend systems, enabling you to unlock core business data and enhance the range of services provided by your apps. With event handling, you can write logic that triggers these data flows and responds to a user's interaction with your data.

The Realm Platform is versatile and flexible, allowing for easy integration under multiple architectural patterns including:

- Realm Platform alone: use the Realm Node.js SDK and contain all your business logic in JavaScript.
- Realm Platform coupled with APIs: use REST APIs to integrate with other services on your backend.
- Realm Platform coupled with external databases: use the Realm Data Connector to keep tables in a relational database in sync with realms on the ROS.

Deploy On-Premise or in the Cloud

Because Realm is installable software, not a cloud service, you can choose where you want to deploy the Realm Object Server to fit your business needs. Some businesses need to run apps in remote locations with little to no mobile network coverage. For example, a cruise ship may serve news and information to passengers through a custom app. Realm's flexibility means that it is just as easy to run Realm on existing on-premise servers behind your firewall, in your private cloud, or on your AWS or Azure instances in the public cloud.

Great Developer Experience

Realm's simplicity frees developers from worrying about the complexities of moving data across networks and legacy systems. In a large engineering organization, the Realm Object Server and API integration can easily be set up and maintained by a back-end services team using JavaScript and Node.js. This allows the mobile development team to focus on building a great user experience using their native language of choice and a familiar object-oriented development approach. Realm's simplicity means that app code stays lean, performant, and easier to debug and maintain. The organization as a whole benefits from producing better quality, more engaging apps with greater speed and efficiency.

KEY MOBILE USE CASES

To date, there have been more than 1 billion installs of the Realm Database worldwide, and this base is growing rapidly. Although Realm powers data flows in apps of every type and size, there are four use cases for which the Realm Platform is particularly well suited.

REACTIVE APP ARCHITECTURES

The hallmark of sophistication in today's mobile apps is responsiveness. Whenever a user interacts with an app, UI navigation must be fluid, data input and display must be immediate, and overall functionality must feel snappy and keep pace with the user's intentions. For businesses, app responsiveness is not only a brand concern, but also a critical driver of business. With slow or frustrating apps, users will likely drop out of key processes, such as sharing information or making a purchase. They may stop using the app altogether, which gives competitors who build reactive apps an advantage.

Reactive programming methodologies look to architectures and patterns that support such a streamlined user experience and enable a frictionless flow of data between app and server or back-end systems. Such architectures must also ensure app resiliency against failures, as well as seamless scalability with user demand.

Realm provides a simplified, reactive data architecture that improves data interactions and data handling on the edge device. The full-featured Realm Database sits on the client side and stores data locally as live objects, which are continually listening and reacting to updates by triggering the data synchronization process. The Realm Object Server enables developers to write event handling and business logic against data flows in order to power specific features or use cases. Combined in the Realm Platform, your developers can leverage these tools to quickly and easily build reactive apps that will delight and empower your users.

[Watch a demo](#) and learn more about building reactive apps with Realm.

OFFLINE-FIRST EXPERIENCES

Mobile networks are notoriously inconsistent. App users regularly experience spotty coverage, congestion, or intermittent network fluctuations, even in an urban environment. Apps that rely on the transfer of data are particularly at risk for providing a poor user experience in the face of disrupted connectivity. Sophisticated apps today are designed from the ground up to handle offline use cases and enable much of the app's functionality to persist.

Taking an offline-first approach, developers write massive amounts of code to address the array of possible network failure opportunities, handle data conflicts, and ensure app resiliency. This adds an inordinate amount of complexity to app development and results in bloated code that is more challenging to test, debug, and maintain. It also diverts developer time and attention away from building core features and a differentiated experience.

The Realm Platform abstracts away the most challenging aspects of offline-first development. Our client-side database makes your app's data accessible at all times, allowing it to function smoothly offline. When back online, Realm triggers the data synchronization process and seamlessly update your app with any incremental changes. The Realm Object Server handles data conflict resolution deterministically, using preset or custom rules. With our platform approach, your app developers are free to focus on the UI/UX while keeping code lean and performant.

[Learn more](#) about supporting offline-first use cases with Realm.

MOBILIZING LEGACY APIs

When working with APIs in mobile apps, developers face numerous challenges. The brittleness of APIs in the mobile context requires significant development time and effort to write code to ensure resiliency and seamless integration. Mobile networks regularly experience high latency and unstable connectivity that impedes an API's request-response-update process, resulting in a slow user experience. Often APIs are synchronous, which can be even more problematic for developers. Moreover, devices themselves present unpredictable conditions that can force an app to suddenly restart in the middle of an API call.

When it comes to connecting mobile apps with back-end systems, APIs pose further issues. Legacy APIs are often written in outdated protocols that make them difficult to integrate using modern app languages. In large organizations, front-end developers must request and wait for any API changes needed, or conversely, respond to app crashes due to unexpected updates to the APIs.

Realm's solution is to move all API handling from the client to the server. The Realm Object Server becomes your single place to write manage all API integrations and associated business logic. This reduces the amount of code and complexity on the client side, and centralizes your API operations across device platforms. Realm's live object model effectively makes objects act like APIs as they sync data with the server in realtime and replace the slower, failure-prone API call-and-response model. In this way, Realm helps you unlock your existing infrastructure by more easily mobilizing your APIs.

[Learn more](#) about mobilizing APIs and legacy systems with Realm.

REALTIME COLLABORATION

Some of today's most successful mobile businesses use realtime collaboration to make their apps stand out. For example, messaging apps like WhatsApp depend on continuous conversation between users. Uber visualizes the position and movement of their available fleet. Instagram enables instant photo sharing and engagement. For these apps, immediacy is fundamental to the app experience itself, as well as to the mobile business. If data synchronization is slow or bumpy, users will become easily frustrated and abandon the app.

Many developers sync data using REST calls. REST is a well-established protocol and follows a fairly straightforward call-response-update pattern. However, beneath the surface there is significant complexity that developers must address to build a resilient REST-based solution in the mobile environment. Failures can happen due to a number of network issues, such as latency or unpredictability, and data conflict can arise with online/offline usage and collaborative experiences between multiple users.

Realtime data synchronization is the foundation of the Realm Platform. Realm's "live objects" model automatically syncs data across devices. This means that if a user inputs or changes data in an iPhone app, that data is synced in realtime with your entire install base of apps, whether they be iOS or Android. Data is also synced with the Realm Object Server on the back-end, which runs your business logic and processes. Realm's automatic conflict resolution system ensures predictable merging of record data, even with multiple writers. Because Realm handles the complexity of data flows, your developers can focus their expertise on what drives the most value for your business — building an exceptional collaborative UX.

[Learn more](#) about using Realm to build realtime collaboration into your apps.

Ready for Enterprise

A broad spectrum of organizations, from startups to Fortune 500 giants use Realm to power their data-centric experiences. Because nearly every company today needs an app strategy, our customers span a wide spectrum of industries. You'll find Realm behind apps from world-renowned brands in technology, media and entertainment, consumer goods, travel and transportation, healthcare, food service, pharmaceuticals, financial services, and many more.

We recognize that enterprise companies look to adopt enabling technologies that can help them become more agile, more efficient, and ultimately more competitive. Realm meets enterprise priorities head on with a high-performance platform that enables:

- Faster time-to-market, responsive to market demands
- Higher quality, more engaging app experiences
- Increased development capacity to build advanced features
- Greater audience reach, seamless cross-platform apps
- Standardization of data management across apps and devices

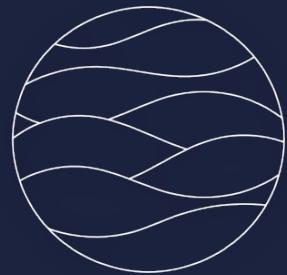
The Realm Platform is available in multiple configurations to support the needs of any developer. The Enterprise Edition offers premium features and custom support, providing deeper system integration capabilities and the scale needed to drive app development success at the enterprise level.

Learn More About Realm

As the world becomes ever more connected, the mobile internet is evolving into an open network of billions of users and trillions of devices. At Realm, we're dedicated to building a data layer that can deliver the full potential of this mobile-first world. We'll continue to support developers as they build innovative new apps that offer realtime, interactive experiences across them all.

To learn more about the Realm Platform, we encourage you and your team to try it for yourselves. Download the free [Developer Edition](#), the 60-day trial of the [Professional Edition](#), or contact us for a free demo of the [Enterprise Edition](#).

www.realm.io



realm

www.realm.io

© 2017 Realm Inc.