MongoDB Stitch

The Serverless Platform From MongoDB
June 2018

Table of Contents

Introduction

MongoDB Stitch is a serverless platform which accelerates application development with simple, secure access to data and services from the client – getting your apps to market faster while reducing operational costs and effort.

Stitch represents the next stage in the industry's migration to a more streamlined, managed, agile infrastructure. Virtual Machines running in public clouds (notably AWS EC2) led the way, removing the need to manage physical hardware. Hosted VMs were followed by hosted containers and serverless offerings such as AWS Lambda and Google Cloud Functions which removed the need to think about the environment your code ran in at all. However, traditional serverless offerings still required backend developers to implement and manage access controls and create REST APIs to provide access to microservices, public cloud services, and of course data. Frontend developers were held back by needing to work with simplistic APIs that weren't suited to rich data queries.

Another trend has been the shifting of the application's business logic from the backend to the frontend (i.e., the browser, IoT asset, or mobile device). There are several incentives for this move of work to the edge:

- It enables your app to tap into the increasing processing power of your users' devices; reducing the need for expensive resources to power your backend. This provides a more scalable architecture, where every new user brings their computing resources with them.
- By performing most of the work at the edge, users experience faster response times and more immersive experiences – this is even more significant if the data can be stored locally on the device as it reduces data movement between tiers.
- Progressive apps continue to provide critical features when the client application cannot contact the backend, for example when the user has no WiFi or cellular coverage.

Of course, some work still sits best in the backend, for example:

- Code that makes many calls to backend services (e.g., the database) can run with less latency if close to that service.
- Sensitive data can be kept more secure in the backend environment.

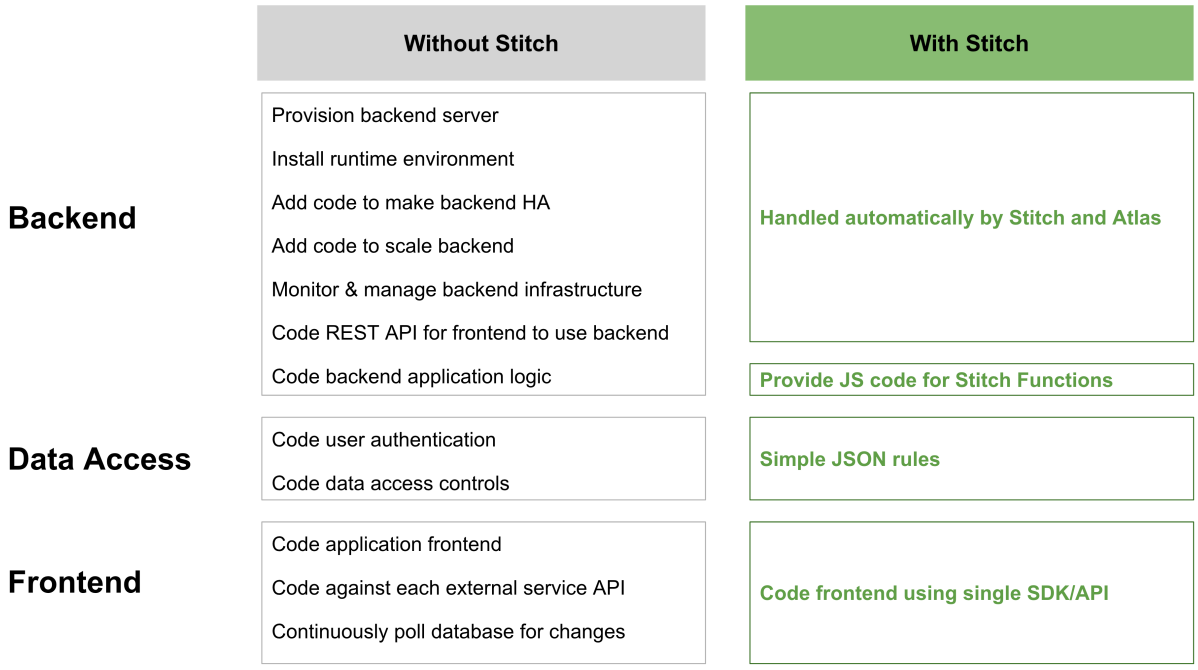

Figure 1: Work to develop app with and without MongoDB Stitch

- Application code is hidden from the user, protecting your intellectual property.

> One of the most important and universal trends in modern software application development is a focus on value, specifically where developers can add value versus where they are merely reimplementing common scaffolding. Platforms that understand this are responding by automating and abstracting undifferentiated work away from those developing software, allowing them to focus on the application itself. MongoDB's Mobile and Stitch offerings are examples of precisely this trend at work.
>
> — Stephen O'Grady, Principal Analyst with RedMonk

You need your application backend platform to interact with your services (whether cloud or microservices) and database, and to execute relatively small pieces of custom code. This is what we built the MongoDB Stitch platform to provide.

Whether building a mobile, IoT, or web app from scratch, adding new features or integrations to an existing app, or safely exposing your data to new users, Stitch can take the place of your application server and save you writing thousands of lines of boilerplate code.

Developing an Application With and Without MongoDB Stitch

You can save many weeks of development and thousands of lines of code by developing your app using Stitch. Figure 1 presents some of the efficiencies you can achieve when developing with Stitch.

Before the move to cloud services, you had to build and support the entire hardware and software stack below your application. This effort was wasteful and meant that you were spending lots of time and resources on infrastructure that didn't add any unique value to your business.

The introduction of cloud services such as AWS EC2 instances improved things but did little to simplify the data management layer.

When MongoDB launched Atlas, you could replace your database infrastructure with this service – freeing you from

having to manage your data infrastructure and making it on-demand and elastic.

Stitch takes this a step further – taking care of the backend infrastructure and the thousands of lines of boilerplate code customarily needed to control user access to data, and to integrate with services. This new stack (Figure 2) leaves you completely free to focus on the code that's unique to your business and delivers value to your users.

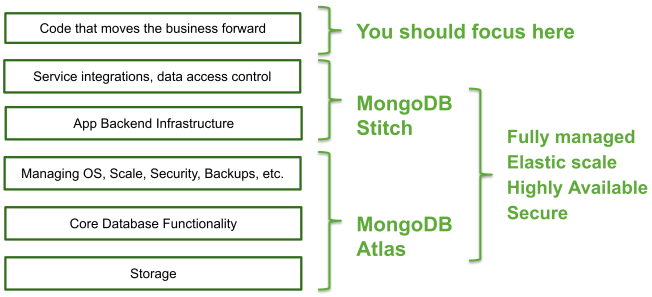

Figure 2: Focus development effort where it matters with MongoDB Stitch

Stitch Services – The Best Way to Work with Data

The Stitch serverless platform addresses the challenges faced by the developers of modern applications by providing four services:

Stitch QueryAnywhere

Stitch QueryAnywhere brings MongoDB's rich query language safely to the edge. Intuitive SDKs provide full access to your MongoDB database from the browser and mobile and IoT devices. Authentication and declarative or programmable access rules empower you to control precisely what data your users and devices can access.

SDKs (client libraries) are available for iOS, Android, and JavaScript. The SDKs provide the same query language used by backend applications accessing MongoDB directly. Your frontend application has access to all of MongoDB's functionality and scalability.

Choose from a set of access rule templates or build more complex, custom access patterns with JSON or JavaScript.

For example, a medical application could:

- Allow a doctor to only read and write medical records for patients under their care.
- Allow a nurse to see a partial view of patients records, check for allergies, and record test results.
- Allow an administrator to execute a real-time aggregation to find which doctors or services are keeping patients waiting for the longest – without being able to access any sensitive patient details.

With Stitch, these access patterns can be built out using generic MongoDB queries on the frontend, backed by a set of comprehensive access rules in Stitch.

Stitch Functions

The Stitch serverless platform also lets you host and run Stitch Functions – implemented as full JavaScript functions. These functions have access to application context such as information about users, requests, services and globally defined values.

Stitch's custom HTTP service lets you integrate microservices and cloud services with your application logic. The HTTP Service can also map incoming requests

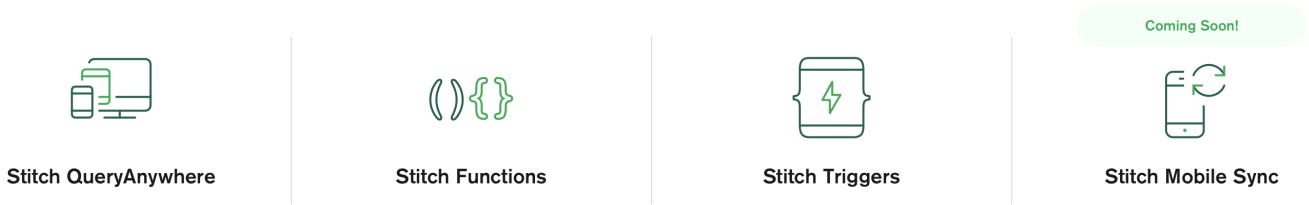

Figure 3: MongoDB Stitch Services

to functions, enabling you to create secure, custom application APIs.

As an example, you can create a function that accepts an expense claim, checks its validity, adds it in a pending state to your database, and sends a text notification to the approver. Then you can integrate it directly into your frontend with a single line of code.

Stitch Triggers

Stitch Triggers provide real-time notifications to your application functions, letting them react in response to database changes, as they happen, without the need for wasteful, laggy polling back to the database.

Stitch Triggers are built on top of MongoDB Change Streams, providing a simple, efficient way to consume and act on changes to your data.

Unlike RDBMS triggers, the logic runs on Stitch's infrastructure rather than inside the database, where it competes for resources with other database operations.

For example, you could create a trigger so that when a customer's account balance falls below a pre-defined threshold Stitch calls a function which sets a warning flag in their account in the database, and sends them a notification email.

Stitch Mobile Sync (coming soon)

With the addition of MongoDB Mobile, MongoDB lets you store data where you need it, from IoT, iOS, and Android mobile devices to MongoDB Atlas. Wherever you store your data, you use a single database and query language. Applications can use the Stitch SDK to access data, whether it's held on the mobile client or the backend. With native JSON storage, indexing, and aggregations, users can query data any way they want. With local reads and

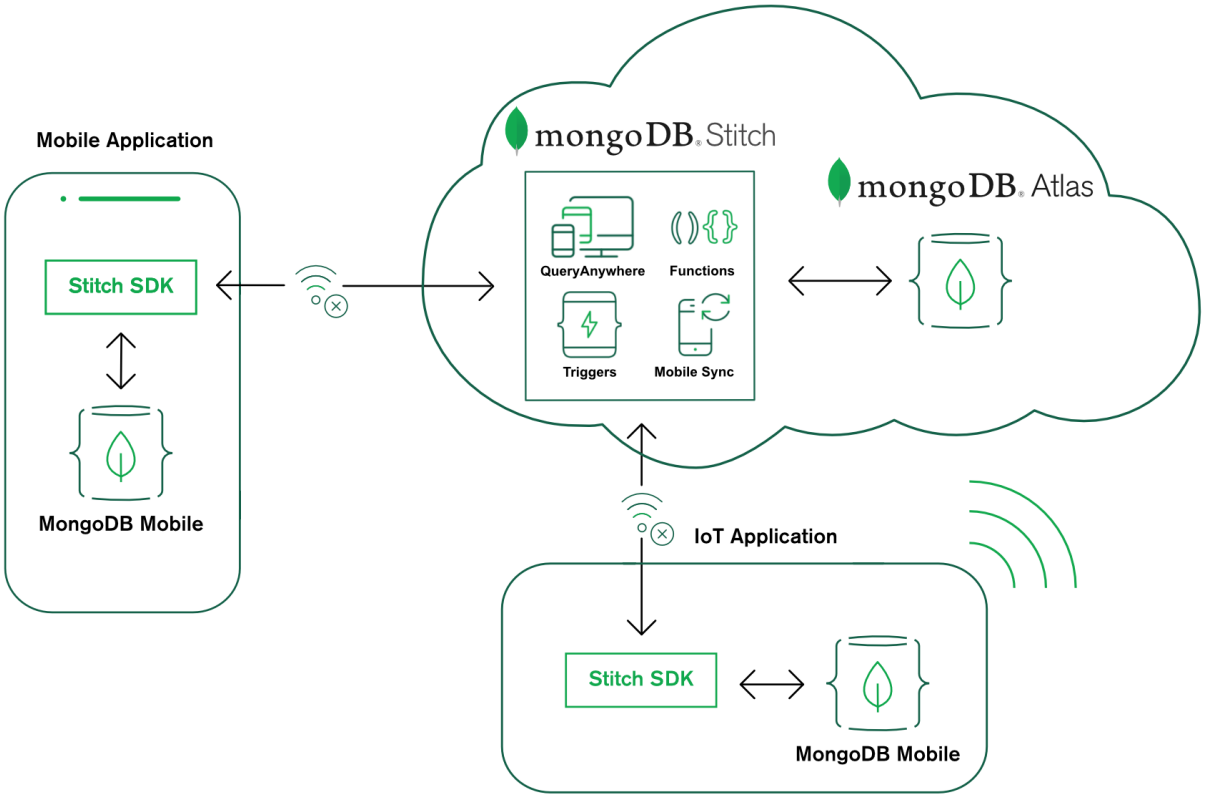

Figure 4: Syncing of data between MongoDB Atlas and MongoDB Mobile with Stitch Mobile Sync

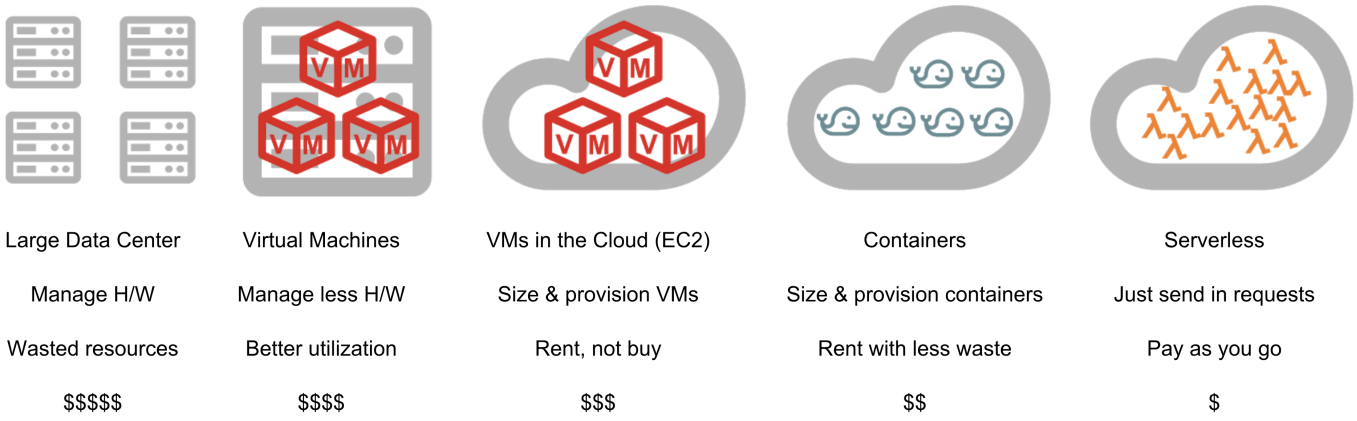

Figure 5: The evolution to serverless and MongoDB Stitch

writes, MongoDB Mobile lets you build fast, reactive apps, regardless of your connectivity.

Stitch Mobile Sync automatically synchronizes data between documents held locally in MongoDB Mobile and your backend database, helping resolve any conflicts – even after the mobile device has been offline.

This capability is the perfect way to build a new IoT or reactive, progressive mobile application. For example, the apps provided by airlines provide access to valuable information when traveling or planning future trips. Stitch Mobile Sync ensures that critical data is always available. When connected to the internet, you have access to all of the features, including finding and booking flights, checking in, retrieving your boarding pass, checking your flight status and boarding gate, and even checking the weather at your destination.

Of course, when traveling, you're often without internet connectivity, and so without a database embedded in your phone, you lose all of those features. With MongoDB Mobile, the app can store essential data on your phone, letting you use critical functions such as viewing your boarding pass and upcoming flight details – even when offline. We refer to apps that continue to offer an offline, albeit less feature-rich experience like this as "progressive".

When the underlying data in the backend database changes (for example, a flight's gate changes), you want the update to be available in the mobile app as soon as it happens. Once the application is aware of the change, it can update the information displayed and alert the user.

One way to implement this is to have the app periodically poll the backend database for changes, but this is wasteful of resources and can lead to long delays in discovering a change. The better option is to use MongoDB Stitch Mobile Sync to synchronize the changes between the Mobile database and the backend database and send a notification to the traveler as soon as the data changes.

What is Serverless?

As noted, Stitch represents the next stage in the industry's migration to a more streamlined, managed, agile infrastructure. Virtual Machines running in public clouds led the way, followed by hosted containers, and serverless offerings such as AWS Lambda, Azure Functions, and Google Cloud Functions. Each stage made it even faster for developers to get access to an environment to run their code. But they still required backend developers to implement and manage access controls and REST APIs to provide access to microservices, public cloud services, and of course data.

There are many similarities between Stitch and services such as AWS Lambda, Azure Functions, and Google Cloud Functions, but there is a fundamental difference in their implementation. Stitch receives requests for many applications and runs the relevant logic (e.g., database access or function call) within a shared environment. In contrast, the other services run a container for each function. While this bulkier, more isolated container approach adds flexibility (such as embedding third-party Node.js modules), it uses more resources. Hosted

containers can also hurt latency and impact user experience during cold starts as the function can't execute its logic until its container has spun up.

Stitch is the best serverless solution for working with MongoDB. For example, Stitch rules provide a level of data access control granularity and flexibility that isn't available anywhere else, and if you're using Atlas, then you manage your functions and database through the same UI.

These are complementary technologies, and people use them together, for example, the simplest way to have Lambda access MongoDB through a persistent connection is to have the Lambda function send requests to Stitch rather than accessing MongoDB directly. In the opposite direction, there are cases where it makes sense for a Stitch function to call a Lambda function, such as for a computationally expensive operation that makes use of libraries.

Lambda et al.	MongoDB Stitch
Serverless platform	Serverless platform
General purpose	The best way to work with MongoDB
DB connections are closed after the function completes	Optimized, perpetual connection to the database
Blank sheet of paper – there are few constraints on what you can build, but you need to write extra code for data management	Value-added features (triggers, rules, mobile sync etc.)
Tightly integrated first-party services	Tightly integrated with MongoDB, looser integration with other services
Dedicated container image with freedom to bring your own software libraries	Shared, always ready infrastructure – fast and efficient (no cold start, low memory use)
Perpetual free tier (with limitations)	Perpetual free tier
The platforms are more complementary than competitive Stitch Functions will call Lambda functions Lambda functions will call Stitch Functions	

Stitch Benefits

MongoDB Stitch gets your application to market faster

- **Serverless platform**: You don't need to provision and manage servers, VMs, or even containers – we'll execute your requests without the overhead of other serverless platforms. Focus on implementing your business logic and UI, and let us look after the infrastructure.
- **Stitch QueryAnywhere & Stitch Functions**: Provide secure access to all the features and scalability of MongoDB while making it easy to coordinate between services, data, and clients securely. All through a single, intuitive SDK.
- **Stitch Triggers**. Lets your application react to database changes without you having to write cumbersome polling software.
- **Stitch Mobile Sync (coming soon)**: Ensures that the local documents stored in MongoDB Mobile are kept in sync with your backend database, without you having to write highly complex syncing software that has to work across multiple devices and network connections.
- **Cross-platform**: Create your Stitch backend application only once, and then access it from all of your web, iOS, and Android client apps.
- **Existing apps untouched**: Your existing applications have full access to the data through any of the MongoDB drivers. Simply and securely make that same data available to new applications using Stitch.

MongoDB Stitch reduces your operational costs

- **Serverless platform**: We look after managing the platform – provisioning, upgrades, security, and monitoring. You focus on developing your application.
- **Pay as you go**: Pay only for what you consume when you consume it. There's a generous, perpetual free tier and then a low-cost, consumption-based pricing model. Refer to the documentation for details.
- **Capacity on demand**: You don't need to predict levels of user demand. MongoDB Stitch intelligently provisions and manages compute across tens of thousands of applications making it simple for Stitch to transparently scale to your workload at any point in time. Stitch scales automatically to cope with any workload that you throw at it.
- **Secure by default**: Stitch has built-in authentication (Google, Facebook, JWT, anonymous, username & password, and API keys) and authorization. With powerful, user-defined rules, Stitch acts as your secure access gateway for applications to access your data and services.
- **Automate application versioning and deployment**: The CLI automates the export and import of Stitch applications while ensuring that any secrets (such as keys) are protected.

MongoDB Stitch reduces your development effort

- **Avoid generic backend code**: 41% of developer time is spent working with backend services[1] – you shouldn't waste your effort writing that boilerplate code when you could be focused on value-add application code instead.
- **HA & scalability built in**: The most significant two challenges in building any application backend are high availability and scalability. The Stitch serverless platform is built from the ground up to provide both.
- **Orchestrate services**: Stitch Functions integrate with public cloud services as well as internal microservices – controlling what data is passed between them and who can access which services. Once you've configured your services and defined rules, your frontend application can access them all using a single, simple SDK.
- **Stitch runs your code**: You maintain the flexibility to customize Stitch behavior by writing custom JavaScript (ES6) functions. These functions are hosted on Stitch and can be invoked by your webhooks, other Stitch functions, or your frontend application through the Stitch SDK.

1. https://www.mongodb.com/blog/post/stack-overflow-and-mongodb-research-unveils-developer-productivity-struggles

MongoDB Stitch & Stitch Mobile let you build richer apps

- **Progressive, always available apps**: MongoDB Mobile (beta) extends developers' ability to put data where they need it, using a single database and API that accesses data locally on mobile devices, and centrally in the backend server. Your application frontend can continue to read and write data when disconnected from the network, with Stitch Mobile Sync automatically reconciling changes between the Mobile database and a backend MongoDB cluster once it reconnects.
- **Reactive apps**: Stitch Triggers and Stitch Functions let your application react in real time to authentication activity (coming soon), webhook calls, timers, and database changes (coming soon).
- **Capitalize on all of your existing data**: Stitch lets your application safely access and utilize any of your organization's data stored in MongoDB.

MongoDB Stitch secures access to your data

- **Authenticate anything and anyone**: Stitch will authenticate and authorize applications, DBAs, partners, or millions of end users – all based on the rules you declare.
- **Protect your entire application**: Fine-grained authorization – you can define rules to permit read or write access to complete documents, down to individual fields, services, and functions. Stitch adds secure default permissions to protect your data – overwrite these rules to change permissions where needed.
- **Flexible rules**: Your Stitch rules can draw on information on the end user, from your database, or from external systems.
- **Flexible authentication options**: You can authenticate users or applications using your existing systems or using built-in options: Facebook, Google, username & password, JWT, anonymous, and API keys.

Core Stitch Features

User & App Authentication

Just about every app needs to authenticate users, or at least uniquely identify them. While essential, this is the very definition of functionality that is repetitive, mundane, and does nothing to make your app stand out. It takes you minutes to configure Stitch to work with popular cloud authentication services, to work with your existing authentication system, or to implement a username/password scheme inside Stitch.

Once you've configured an authentication service through the Stitch UI, it typically requires a single method call from the app frontend to authenticate a user.

Stitch currently supports five user authentication methods: anonymous, email/password, Google, Facebook, and custom/Java Web Tokens. The documentation provides details for each.

Once Stitch has authenticated a user, they're assigned a unique object. Your Stitch application can then use the contents of that object – for example, to control what data they can access.

It's also possible to securely authenticate external applications rather than users. Application authentication is based on a shared secret.

Authorization – Rules

Stitch provides an unprecedented level of granularity and flexibility when you need to control what each user can access – Stitch achieves this using rules. You can use rules to control access to fields, documents, services, or hosted functions. You can define rules declaratively, using a JSON syntax which is very similar to the MongoDB query language.

For even greater flexibility, your rules may invoke JavaScript functions hosted within Stitch. You can write those functions to perform whatever checks you can dream up.

Rules are incredibly powerful, a few clicks or lines of code in Stitch can replace months' of effort implementing access controls within your application code.

MongoDB Atlas Integration

Stitch lets your application exploit the full power and scalability of MongoDB – whether using your existing database or building an entirely new application from scratch. If you're working with an Atlas database that is in use by existing applications, those applications can continue to access their data through any of the supported MongoDB drivers. Tools such as MongoDB Compass and the mongo shell can access your database directly.

There is a tight integration between Stitch and any MongoDB Atlas cluster (including those in the free tier) which is part of your Atlas organization. This integration means that you don't need to take any extra security or configuration steps to allow Stitch to access your data. While your data is automatically available to your Stitch platform, applications (and end-users) cannot access a collection until you add a read and or write-rule for that collection – ensuring that it's secure by default.

Database Access

Your app frontend can access collections using the database actions available in the Stitch SDK – the functionality and syntax of these queries match those of backend MongoDB drivers. Alternatively, you can implement Stitch functions which access the database, the app frontend then invokes those functions through the Stitch SDK. There are three advantages to using functions rather than allowing frontend apps to access the collections directly:

- Security through obfuscation. Not allowing the frontend application to know the database or collection names reduces the surface area for any potential security attack.
- Data enrichment. Rather than just sending or writing the raw data, the function can enrich it – for example, adding extra information downloaded from a cloud service.
- Side-effects. In addition to reading or writing the data, you could have your function write an audit log to a collection, or send a text message via Twilio. Note that this can also be achieved using Stitch Triggers.

Service Integrations

Stitch comes with built-in integrations with popular cloud services such as Twilio, Github, and Amazon services[2] such as Lambda, Kinesis, EC2, Machine Learning, Rekognition, SES and S3. You configure these services through the Stitch UI (typically providing your id and key for the service). As with other resources in Stitch, you can add rules to control how and when a service can be run.

Once configured, you can invoke a service with a single line of code in the app frontend or from a Stitch function.

By invoking cloud services through Stitch, you don't need to include any secrets for that service in your application frontend code – increasing security.

For those third-party services not built into Stitch, it is straightforward to configure a custom integration. Stitch's HTTP service enables you to integrate Stitch apps with any service that provides a REST API, such as Slack, or your microservices.

You can use the HTTP service to make outgoing HTTP requests to these services, or you can create incoming webhooks that respond to incoming HTTP requests.

If you're working with microservices, the HTTP service and webhooks allow you to orchestrate and enrich the flow of data throughout your system.

Functions

Stitch functions let you run your JavaScript code server-side, within the Stitch execution layer.

Functions can access data on the user, documents from MongoDB, variables stored as Stitch values, and more. Functions can call other functions, invoke services, and write to MongoDB. Functions can be called from the frontend through the SDK, by other functions, from rules, and from webhooks.

2. Find a full list of supported AWS service in the docs.

The range of things you can do with functions is infinite, but a few examples are:

- Execute a complex MongoDB aggregation, which you can invoke from your application frontend with a single, simple SDK call
- Obfuscate your schema from the front-end app, increasing security
- Execute multiple services through a single SDK call. For example:
  1. Receive a purchase request from the frontend
  2. Access the database to confirm that the item is in stock
  3. Read the user's name, email, and address from MongoDB
  4. Use the HTTP service to message the delivery microservice (with the user's details) to ship the order
  5. Update stock levels in MongoDB
  6. Email the user with their order details.
- Perform sophisticated checks to decide if a user can access a piece of data (with the collection's read-rule calling the function

Functions are the final piece that makes it possible to build an entire application using just Stitch and the code running on your frontend device.

Stitch Triggers

MongoDB change streams enable developers to more easily build reactive, real-time, web and mobile apps that can view, filter, and act on data changes as they occur in the database. Stitch Triggers build upon change streams by digesting the stream of changes and surfacing them as discrete events. Each event can invoke a Stitch function to perform actions such as sending a text message or email or applying a side effect such as updating another document in the database.

These triggers can notify your application of all writes to documents (including deletes) and provide access to all available information as changes occur. This is an efficient alternative polling that can introduce delays, incur higher overhead (due to regularly checking the database, even if nothing has changed), and lead to missed opportunities.

Triggers let your application become genuinely reactive to events, as they happen.

Stitch Mobile Sync with MongoDB Mobile (Coming Soon)

Stitch Mobile Sync will automatically synchronize data changes between data held locally, in MongoDB Mobile, and your backend database, helping resolve any conflicts – even after the mobile device has been offline.

A single application frontend could choose to work with data in three different ways, all through similar methods in the Stitch SDK:

- Data stored only in MongoDB Mobile and used only within the device, accessed through the Stitch SDK
- Data stored only in MongoDB Atlas and accessed through Stitch
- Data stored in both MongoDB Mobile and Atlas and accessed locally through the Stitch SDK – this is the data that Stitch Mobile Sync ensures stays consistent

In the case that the same document is independently updated in both Atlas and MongoDB Mobile, Stitch detects the conflict and lets you decide how to resolve it.

Values

Values are named constants that you can use in MongoDB Stitch functions and rules. They have several benefits over hard-coding actual values in your functions and rules:

- It makes your rules and functions more readable.
- A developer or operations person can find all of the values they might want to change in one place – this is especially valuable when deploying the same Stitch application more than once.
- You can update a value across all functions and rules in a single operation

Push Notifications

MongoDB Stitch supports integrating mobile apps with the Google Cloud Messaging (GCM) service to provide push notifications to the end-user. You configure notification messages from the Stitch UI, while clients register with GCM for messages sent to specific topics.

Stitch Administration API

The simplest way to start using Stitch is through its graphical UI, but at some point, you may want to integrate the configuring of Stitch into your DevOps workflow. Stitch enables this by providing an administrative API, which provides all of the same functionality as its UI. Access to the API is secured using API keys.

Logging & Debugging

Debugging application software running on a serverless platform can be tricky as you don't have access to your favorite tools or even the console. Stitch makes it simple by providing the tools you need, with everything made available through the Stitch UI:

- **Logging**. Stitch logs all requests, function calls, and more – for both successes and failures. Each log includes all of the details you need to understand what your application frontend has asked of Stitch and how Stitch responded.
- **Console**. You can write debug messages to the Stitch console from your functions, and Stitch displays your messages in the UI. When debugging your functions, you can manually invoke them from the Stitch console, passing any desired parameters.

SDKs for iOS, Android, and JavaScript

Your application accesses your Stitch resources (database, functions, values, authentication providers, and services) through a native client library/SDK. The SDKs then send requests to your backend Stitch application using Stitch's API. SDKs are available for iOS, Java, Node.js, and Android. IoT vendors (e.g., Electric Imp) are building Stitch client libraries for their frameworks, making it simple for IoT apps to work with Stitch.

If you've used any MongoDB driver in the past, then you'll feel at home working with the Stitch SDKs – a single syntax for front and backend application code.

Import/Export CLI

MongoDB Stitch applications aren't limited to living in the Stitch UI. With the Stitch Import/Export API, you can develop Stitch applications locally and import them into Stitch, export existing applications to a local directory, and easily share components between Stitch apps.

The CLI automates the export and import of Stitch applications while also protecting any secrets (such as keys). You can also export an application as a template – excluding additional, deployment-specific values.

The CLI makes it simple to manage the lifecycle of your Stitch application – build your app, export it, modify it offline, use GitHub for source control, and deploy it wherever you need it to run.

Where to Use Stitch

Full Mobile/Web Backend

Perhaps the most obvious use case for Stitch is to use it as the complete backend when you're building an entirely new mobile, IoT, or web app.

Stitch is the fastest way to develop your app – it handles the backend work so you can focus on developing the application UI and business logic. With native SDKs for iOS, Android, and JavaScript, Stitch has the critical platforms covered. You don't need to provide any backend infrastructure and Stitch scales automatically as application usage grows.

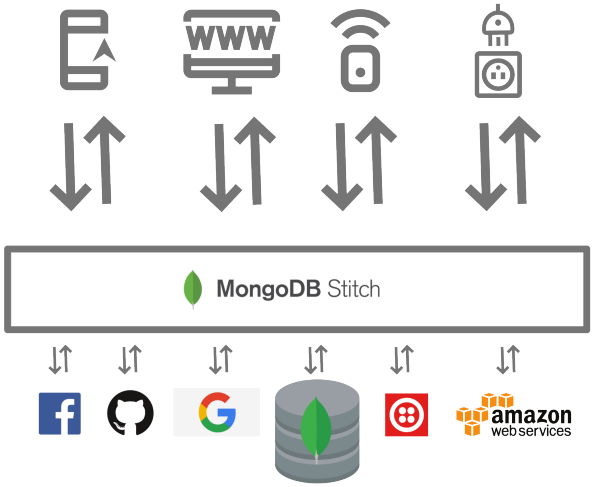

Figure 6: MongoDB Stitch as your application backend

Stitch provides end-to-end per-user context & security for your application. Data access rules are defined using simple JSON documents, or functions for more complicated checks. Your application frontend code can query MongoDB, just as it would if it were running on an app server.

Built-in integrations with leading services (including AWS, Twilio, and Github) let you enrich your application with a single line of code.

JSON is the universal data format for applications, RESTful APIs, MongoDB, and MongoDB Stitch. Because of this, there is no need to waste development and runtime resources converting between formats.

The MongoDB Stitch Mobile database (coming soon) extends developers ability to put data where they need it, using a single database and API that accesses data locally on mobile devices, and centrally in the backend server. Your application frontend can continue to read and write data when not connected to the network, confident that Stitch will sync data changes between the Mobile database and a backend MongoDB cluster when connectivity is restored.

This blog post illustrates how MongoDB Stitch can drastically reduce the work needed to build an application.

Internet of Things

When building an IoT application, Stitch serves as the middle tier between backend services (including the database) and IoT devices (directly or through an IoT gateway). No additional infrastructure is required, and Stitch automatically scales as you introduce additional devices.

Figure 7: MongoDB Stitch as an intelligent IoT hub

Devices can write data to the database using the MongoDB Query Language, or they can invoke Stitch Functions that can enrich the data before writing it – for example adding weather report data for that location. With either of those approaches, latency is low as Stitch maintains open connections to the database. Stitch assures security through the use of API keys and access rules.

Again, JSON is the universal data format in this IoT architecture, making development more efficient.

Enabling Microservices

A vital principle of a microservice architecture is that each microservice should be independent of the others, but they can only achieve anything useful by interacting with each other. That interaction takes the form of data (JSON) passing between microservices. Stitch's role is to act as the intelligent data hub.

Stitch receives data from one microservice through an incoming webhook and sends data to other microservices through the HTTP service. Access rules ensure that each microservice access only the data to which it's entitled. Your functions can execute business logic on the data, enrich it, store it in MongoDB, and determine to which which microservice(s) it should send it.

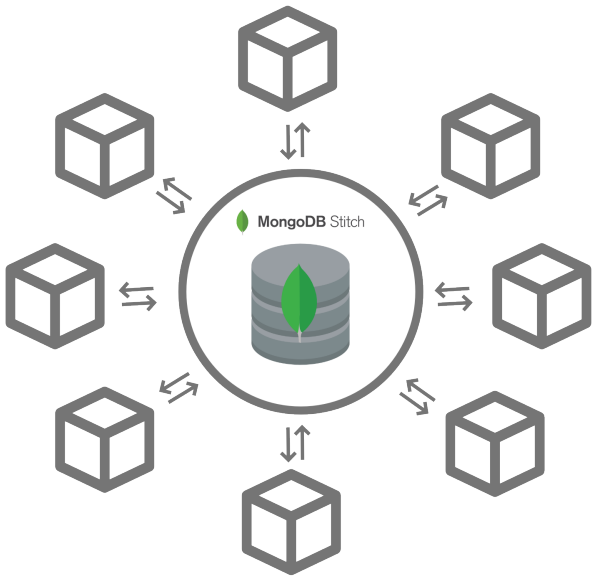

Figure 8: MongoDB Stitch orchestrating data flow between microservices

Part of the principle of keeping microservices independent involves how they store their data – the best practice is for each microservice to store its data independently of the others. Architects typically implement this policy by dedicating a database (cluster) to each microservice – this maintains the desired isolation but means provisioning and managing multiple databases. Stitch provides an alternative approach to achieve the same objective. With Stitch, multiple microservices can safely use a single, central database (MongoDB Atlas) once you have defined Stitch rules to impose the desired isolation. If taking this alternate approach, it opens up the opportunity to find additional value by looking at the data in aggregate.

Data as a Service

Your company's data should be its greatest asset. It ought to be easy to develop new applications based on your data and to generate essential business insights – but for too many, legacy systems and databases make this difficult or impossible.

Organizations are turning to a new approach: Data as a Service (DaaS). This strategic initiative is an investment in consolidating and organizing your enterprise data in one place, then making it available to serve new and existing digital initiatives. DaaS becomes a system of innovation, exposing data as a cross-enterprise asset. It unlocks data from legacy systems to drive new applications and digital systems, without the need to disrupt existing backends.

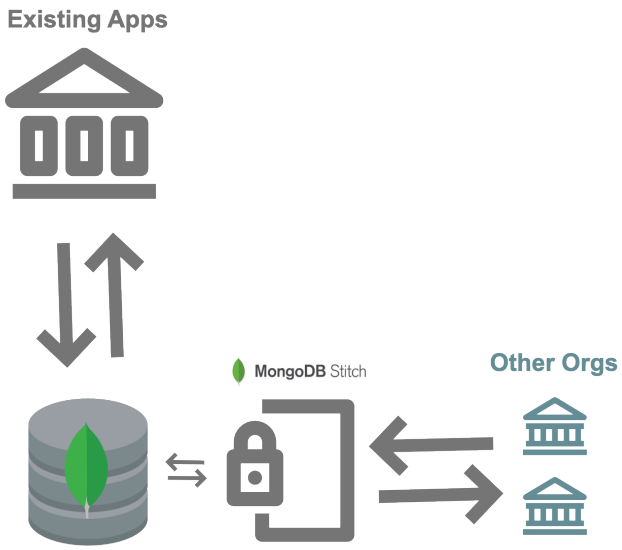

Figure 9: Building Data as a Service with MongoDB Stitch

Adding DaaS to provide selective access to your existing data safely is a perfect use case for Stitch. Your existing applications access their data using the same MongoDB drivers, but other applications, organizations, and companies use new RESTful APIs built using features of Stitch:

- Create the API using Stitch's HTTP service and webhooks.
- Control data access using rules; these rules can easily interact with existing authentication systems.
- Functions can be used to enrich the data and to record an audit trail.

As with the other use cases, no additional infrastructure is required, and Stitch automatically scales as the use of the DaaS APIs increases.

Moreover, APIs use JSON to represent data, and so transforming data is unnecessary.

MongoDB Stitch in the Wild – Acxiom Building Custom APIs With MongoDB Stitch

> Stitch has been fantastic for us. We've cut the time to develop an API for our customers in half, and by combining MongoDB Atlas and Stitch, our teams now have more time to solve business problems versus focusing on the management or operational overhead. That combination has become a key part of our cutting-edge cloud architecture, helping us design, build and manage omnichannel solutions that power exceptional consumer experiences.
>
> — Chris Lanaux, Vice President of Product and Engineering at Acxiom

Acxiom, the data foundation for the world's best marketers, uses MongoDB to build next-generation data environment solutions for Fortune 100 brands across various industries leveraging its proprietary Unified Data Layer framework – an open, trusted data framework for the modern enterprise – that powers a connected martech and adtech ecosystem. Over the past year, Acxiom has also been using MongoDB Stitch to re-platform its Real-time Operational Data Store to a scalable cloud-first approach.

MongoDB Atlas: Database as a Service For MongoDB

MongoDB Atlas is a cloud database service that makes it easy to deploy, operate, and scale MongoDB in the cloud by automating time-consuming administration tasks such as database setup, security implementation, scaling, patching, and more.

MongoDB Atlas is available on-demand through a pay-as-you-go model and billed on an hourly basis.

It's easy to get started – use a simple GUI to select the public cloud provider, region, instance size, and features you need. MongoDB Atlas provides:

- Security features to protect your data, with fine-grained access control and end-to-end encryption
- Built in replication for always-on availability
- Geographically dispersed database that can provide low-latency writes from anywhere in the world. Data can also be easily replicated and geographically distributed, enabling multi-region fault tolerance and fast, responsive reads
- Fully managed, continuous and consistent backups with point in time recovery to protect against data corruption, and the ability to query backups in-place without full restores
- Fine-grained monitoring and customizable alerts for comprehensive performance visibility
- One-click scale up, out, or down on demand. MongoDB Atlas can provision additional storage capacity as needed without manual intervention
- Automated patching and single-click upgrades for new major versions of the database, enabling you to take advantage of the latest and greatest MongoDB features
- Live migration to move your self-managed MongoDB clusters into the Atlas service with minimal downtime

MongoDB Atlas can be used for everything from a quick Proof of Concept, to test/QA environments, to powering production applications. The user experience across MongoDB Atlas, Cloud Manager, and Ops Manager is consistent, ensuring that disruption is minimal if you decide

to manage MongoDB yourself and migrate to your own infrastructure.

MongoDB Mobile (Beta): Brings the power of MongoDB to your device

MongoDB Mobile will bring the power of MongoDB to your frontend.

MongoDB Mobile lets you store data where you need it, from IoT, iOS, and Android mobile devices to your backend in the cloud – using a single database and the same access patterns. MongoDB Mobile lets you build the fastest, most reactive, always-on apps.

Coming soon, Stitch Mobile Sync will automatically synchronize data changes between data held locally and your backend database – even after the mobile device has been offline.

Conclusion & Next Steps

Whether building a mobile, IoT, or web app from scratch, adding a new feature to an existing app, safely exposing your data to new users, or adding service integrations, Stitch can take the place of your application server and save you writing thousands of lines of boilerplate code.

Data, functions, and API orchestration as a service. Stitch services let developers focus on building applications rather than on managing data manipulation code, access rules, or service integrations. Stitch's serverless model provides maximum productivity at the lowest cost.

Register for the free tier and build your first Stitch app.

Safe Harbor

The development, release, and timing of any features or functionality described for our products remain at our sole discretion. This information is merely intended to outline our general product direction, and it should not be relied on in making a purchasing decision nor is this a commitment, promise or legal obligation to deliver any material, code, or functionality.

We Can Help

We are the MongoDB experts. Over 6,600 organizations rely on our commercial products. We offer software and services to make your life easier:

MongoDB Enterprise Advanced is the best way to run MongoDB in your data center. It's a finely-tuned package of advanced software, support, certifications, and other services designed for the way you do business.

MongoDB Atlas is a database as a service for MongoDB, letting you focus on apps instead of ops. With MongoDB Atlas, you only pay for what you use with a convenient hourly billing model. With the click of a button, you can scale up and down when you need to, with no downtime, full security, and high performance.

MongoDB Stitch is a serverless platform which accelerates application development with simple, secure access to data and services from the client – getting your apps to market faster while reducing operational costs and effort.

MongoDB Mobile (Beta) MongoDB Mobile lets you store data where you need it, from IoT, iOS, and Android mobile devices to your backend – using a single database and query language.

MongoDB Cloud Manager is a cloud-based tool that helps you manage MongoDB on your own infrastructure. With automated provisioning, fine-grained monitoring, and continuous backups, you get a full management suite that reduces operational overhead, while maintaining full control over your databases.

MongoDB Consulting packages get you to production faster, help you tune performance in production, help you scale, and free you up to focus on your next release.

MongoDB Training helps you become a MongoDB expert, from design to operating mission-critical systems at scale. Whether you're a developer, DBA, or architect, we can make you better at MongoDB.

Resources

For more information, please visit mongodb.com or contact us at sales@mongodb.com.

Case Studies (mongodb.com/customers)
Presentations (mongodb.com/presentations)
Free Online Training (university.mongodb.com)
Webinars and Events (mongodb.com/events)
Documentation (docs.mongodb.com)
MongoDB Enterprise Download (mongodb.com/download)
MongoDB Atlas database as a service for MongoDB (mongodb.com/cloud)
MongoDB Stitch backend as a service (mongodb.com/cloud/stitch)

mongoDB

US 866-237-8815 ▪ INTL +1-650-440-4474 ▪ info@mongodb.com