

Scenario: Mobile App with TrueVault

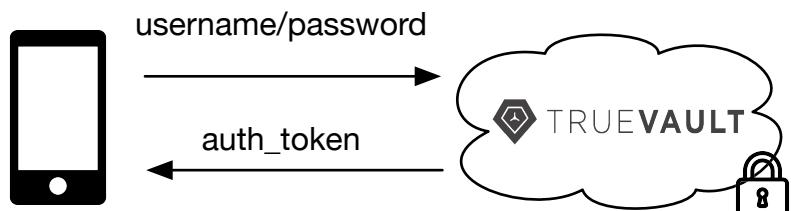
TrueVault provides a HIPAA-compliant, secure Backend as a Service. We recommend taking full advantage of this by offloading all server-side work to TrueVault, using TrueVault as your entire application-facing API.

By interacting with TrueVault directly from the app, without any interstitial API servers, you can vastly decrease your security and compliance burden. Managing an intermediate server, even just a relay — with no data stored permanently — incurs security and compliance costs. The value you are getting from TrueVault isn't just the functionality or developer convenience, but also the operational security as a service. It's our responsibility to ensure your data is encrypted and stored redundantly. It's our responsibility to prevent API-level attacks and patch our systems immediately when an issue is announced. It's our responsibility to meet all HIPAA requirements and help you pass audits with ease.

Let's explore this interaction pattern in more concrete terms.

Authentication

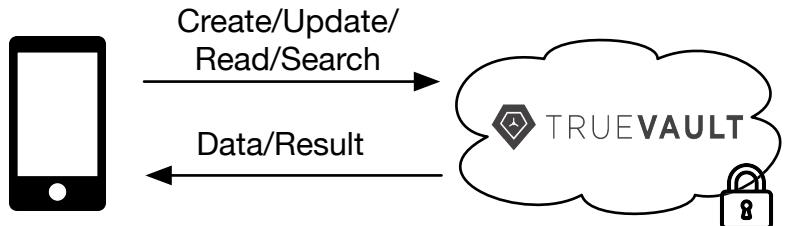
Your users authenticate with username/password directly to TrueVault, and subsequent calls use a secure access token. You do not need to embed any sort of app-wide API key in your source code.



Important: No private API keys stored on device.

Data Access

Once authenticated, your users can create, update, read (by id or search query) and delete their data. Using Groups and Group Policies, you can specify precisely which records a given user should be able to view or modify, and tightly control access to the data stored in TrueVault.

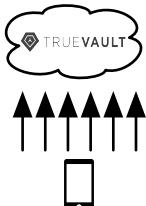


⚠ Gotchas

Local Storage



It is often desirable to cache user data locally on the device. It is possible to do this securely, but it takes **careful and precise** execution. The TrueVault platform secures your data in the cloud, but storage on the device is your responsibility. If you want help, we are happy to introduce you to one of our Trusted Partners who can help you build this functionality.



N+1 Request Problem

Structure your data in TrueVault with your access patterns in mind. There is no equivalent of the relational `join` in NoSQL stores like TrueVault. If you find yourself building a list view that forces you to make an initial request, followed by one request per list item, you've run into the N+1 Request problem. Revisit your data model; we can help!