# FHIRBlocks iOS SDK Test App

## Introduction

Patient health information is confidential and must be protected always. There are several compliance standards to protect patient health data such as HIPAA in the US, BDSG in Germany and so on. FHIRBlocks makes it possible for Healthcare providers to comply to such standards by leveraging Blockchain.

The goal of the FHIRBlocks iOS SDK Test App is to help iOS developers understand the orchestration of REST APIs/services between the core elements of the FHIRBlocks ecosystem. The three main systems of the FHIRBlocks ecosystem are as follows:

1. **FHIRBlocks backend (FBC)** – The heart of the FHIRBlocks ecosystem, this hosts the block chain environment commonly referred to as the “FBC backend”.
2. **Fast Health Interoperability Resources(FHIR) Server** – A server that is compliant with the FHIR standard. This system stores the patient medical records and this can be referred to as the Electronic Health Record (EHR) system. The FHIR server trusts the token issued by the FBC backend.
3. **Identity Provider (IdP)** – The main responsibility of this system is to authenticate and authorize users. The FBC backend trusts the IdP.

In addition to these systems, the patient uses a mobile application (referred to as the client) protected by biometrics to access data stored by a healthcare provider on the FHIR server.

## Overview of the FHIRBlocks ecosystem

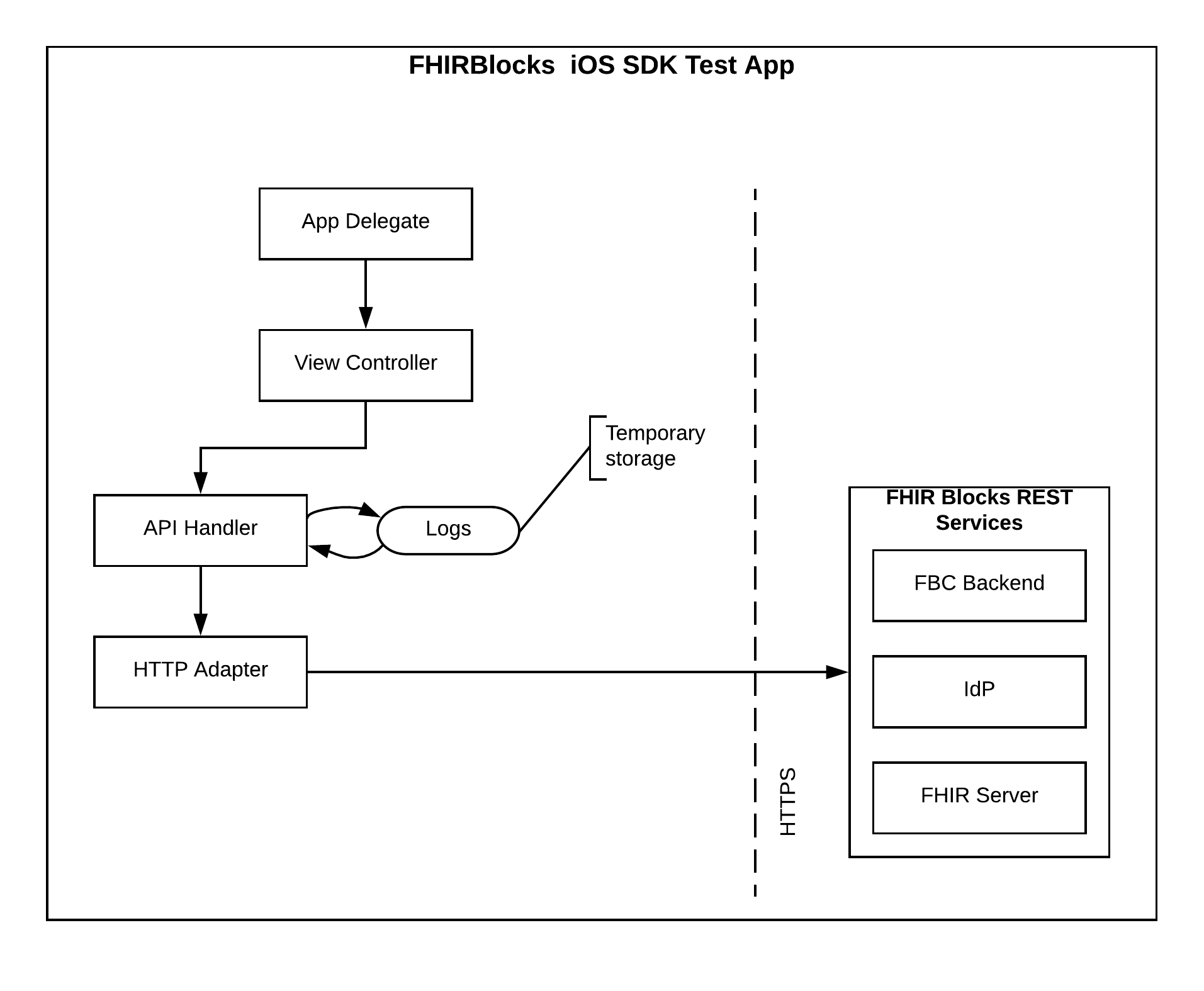
A digital identity is essential in the current day and age for the patient to carry medical records and share them between multiple healthcare providers. The FHIRBlocks ecosystem addresses this point and provides a digital identity to a patient through Blockchain.

In addition to this, an action performed by the patient is audited and stored in the FBC backend making the transaction tamper proof. Such audit entries safeguard the patient as well as the healthcare providers especially when a patient provides consent to share the medical data or to undergo any kind of treatment.

### Signing data with Json Web Tokens(JWTs)

The communication between all the systems in the ecosystem is very secure. Data exchanged between the systems conforms to the Json Web Token (JWT) standard. The client exchanges a public key with the FBC backend and signs all claims with the private key. The FBC backend and the FHIR server verify the signature with this public key and this makes it very hard to tamper the data. The algorithm chosen to sign JWT claims is EC-256.

## Design

The block diagram below indicates the core components of the test app and the interaction between them.

1. App Delegate – The iOS app life cycle delegate. Delegates app lifecycle events to the controller.
2. View Controller – The controller that responds to the events on each view, for example: API list view.
3. API Handler – Manages the orchestration and data management between different APIs. For example: After the DCR registration response, the clientID is made use of in further API calls.
4. HTTP Adapter – An adapter to manage HTTP communication with the FHIRBlocks REST services.

### FHIR resources and their profiles

The FHIR resources currently being written to the FHIR backend follow the profiles given below.

|  |  |
| --- | --- |
| **FHIR Resource** | **Profile** |
| Consent | <http://hl7.org/fhir/StructureDefinition/Consent> |
| DocumentReference | <http://hl7.org/fhir/StructureDefinition/DocumentReference> |
| Provenance | <http://hl7.org/fhir/StructureDefinition/Provenance> |
| Patient | <http://hl7.org/fhir/StructureDefinition/Patient> |

The test app adheres to these profiles by default. It must be adapted to conform to other FHIR profiles in future.

## Testing FBC REST APIs

The test app doesn’t require any specific configuration to start checking the REST APIs. It is designed to work with a demo IdP as well as an actual IdP.

### Working with a Demo IdP

A demo IdP has been setup to understand the interaction between the client, the IdP server, the FBC backend and a FHIR server. The FHIR server in this case is the open source HAPI FHIR server <http://hapi.fhir.org/>. The name assigned to the demo IdP is **“Apollo”**. The credentials of the demo user are:

**UserName:** abc (lowercase)

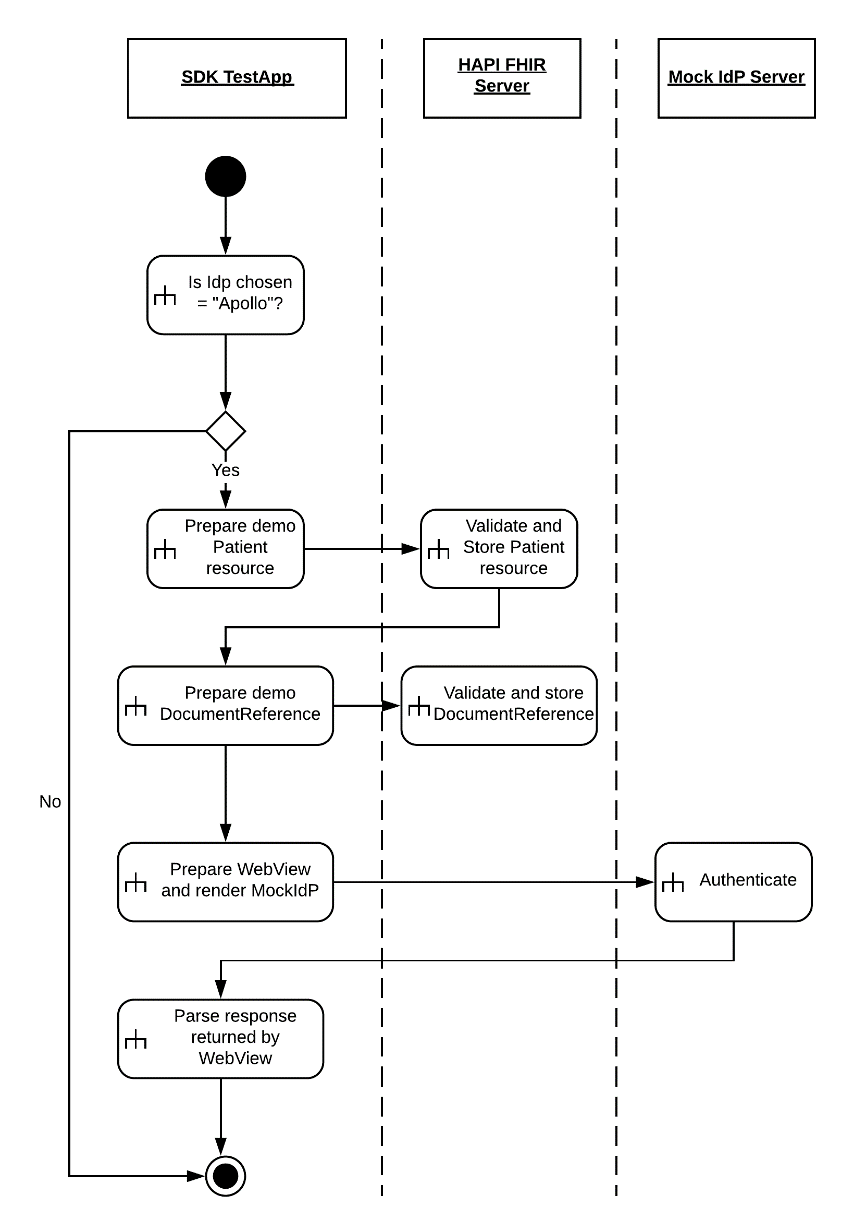
**Password:** 123

### Demo data

The HAPI FHIR server is an open source server that offers to store FHIR resources. But, this server doesn’t retain data permanently and is purged frequently. Therefore, the data must be inserted into the server before the End User License Agreement (EULA) is reviewed and the Consent and Provenance resources are stored.

The test App inserts a sample “Patient” and “DocumentReference” resource conforming to the profiles mentioned above.

The activity diagram below illustrates the case when demo data will be inserted into the system.



### Working with an actual IdP

The test app can work with an actual IdP too (Ex: Duke Resource Server). This case is straightforward, the IdP is rendered in the Webview and once authentication happens the FHIR Identifier is fetched by the app from the FBC backend and this is then used to create a Consent and Provenance record in the FHIR resource server.

## REST API Catalogue

The following REST end points are invoked in the test app. A brief description of these end points are provided below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **API Group** | **HTTP Method** | **Host** | **Endpoint** | **Description** |
| Utilitles | GET | poc-node-1.fhirblocks.io | /smoac/util/ping | Ping the server |
| /smoac/util/getTime | Get time from the server |
| /smoac/util/getTopology | Get FHIRBlocks nodes |
| /smoac/organization | Get the list of healthcare institutions |
| Dynamic Client Registration | POST | mitre.fhirblocks.io:444 | /uma-server-webapp/register | Register the current client with the FHIRBlocks ecosystem. If the response is 200 OK, then a client ID is issued. |
| FBC EULA | GET | hapi.fhir.org | /baseDstu3/DocumentReference | Get the document reference and then query the exact EULA location URI. |
| Patient Resource | GET | hapi.fhir.org | /baseDstu3/Patient | Get the Patient reference. |
| FHIRIdentifier | GET | poc-node-1.fhirblocks.io | /smoac/csi/signedJwtMessage | Pass a JWT token with the command set to the "getFhirIdentifier" |
| Get Oauth token | POST | poc-node-1.fhirblocks.io | /smoac/oauth2/token | Get the Oauth token from the FBC backend. |
| GET | poc-node-1.fhirblocks.io | /smoac/oauth2/beginGetAssertion | Begin assertion. |
| POST | poc-node-1.fhirblocks.io | /smoac/oauth2/finishGetAssertion | Finish assertion. |
| Write Consent resource | POST | hapi.fhir.org | /baseDstu3/Consent | Create a consent record. |
| Write Provenance resource | POST | hapi.fhir.org | /baseDstu3/Provenance | Create a provenance record. |
| FBC Audit | POST | poc-node-1.fhirblocks.io | /audit/signedResource | Audit patient acceptance or rejection of consent. |

**Note:** At present, the app interacts with a fixed set of REST endpoints for the FBC backend. In future, this could be made configurable.

## REST API orchestration

The sequence diagram below provides an overview of the interaction between the test app and all the core components of FHIRBlocks.

