flight

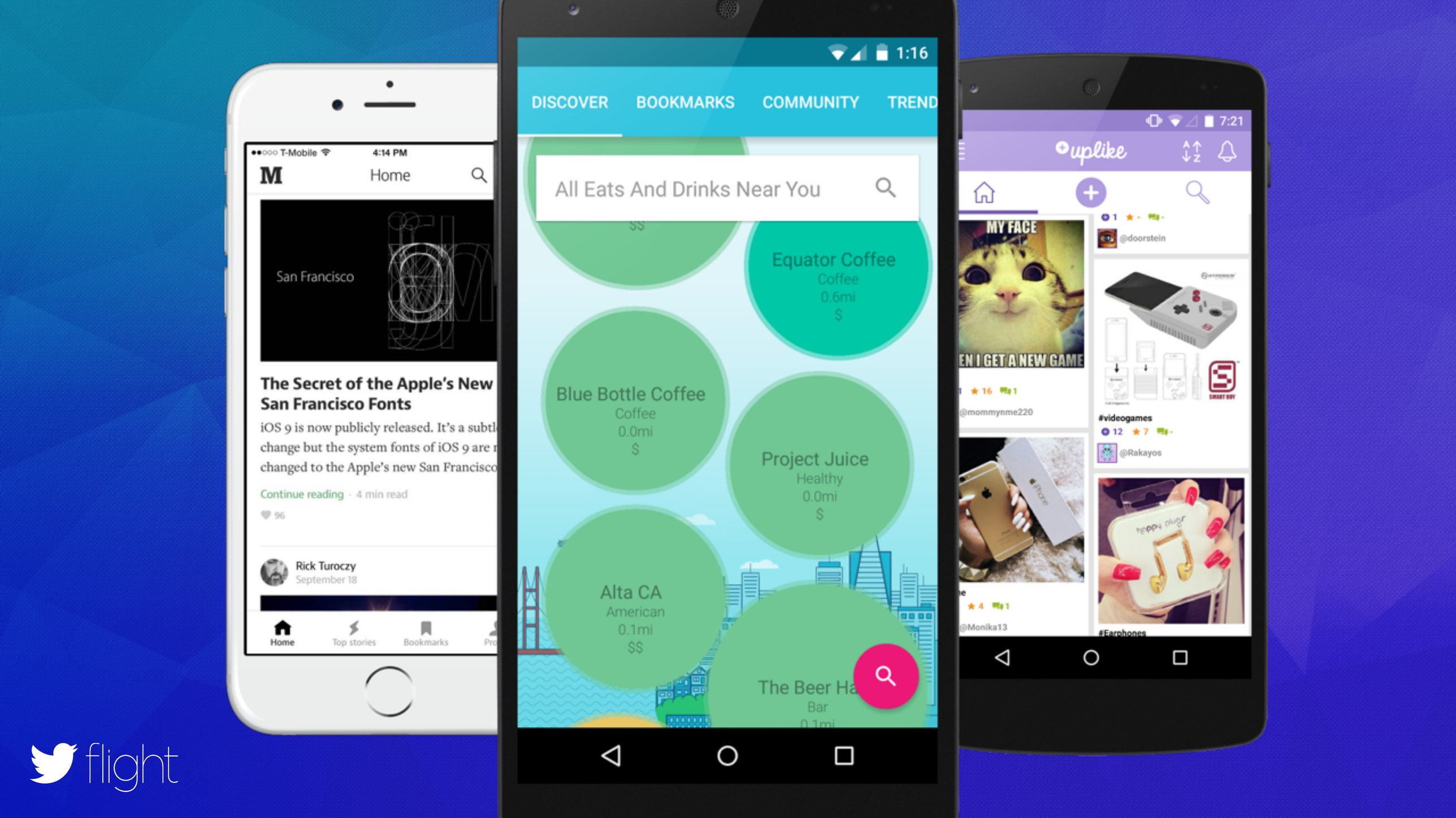
flight

Mobile Authentication Services

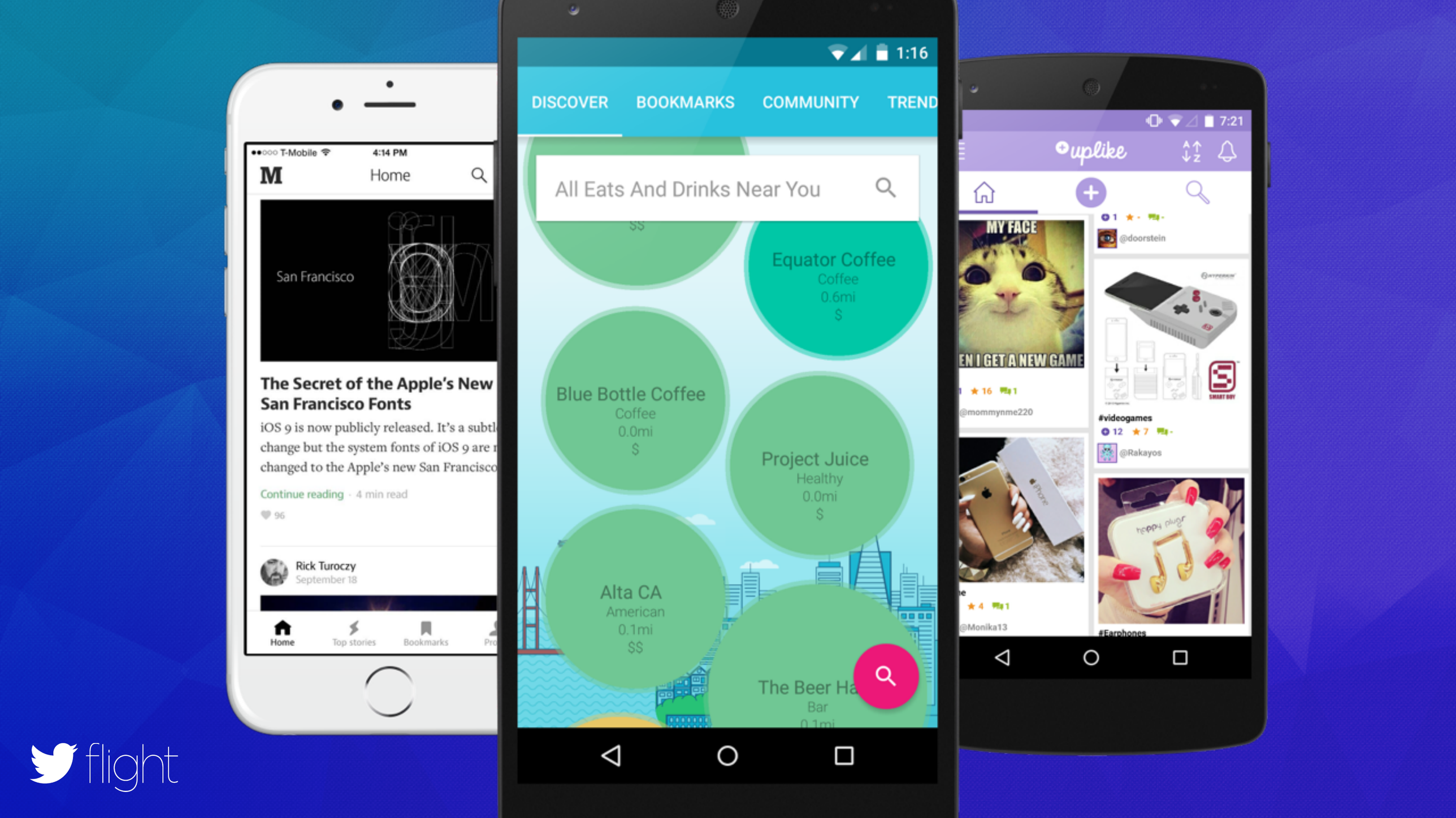
Dalton Hubble

Engineer, Twitter Kit Android |
@dghubble

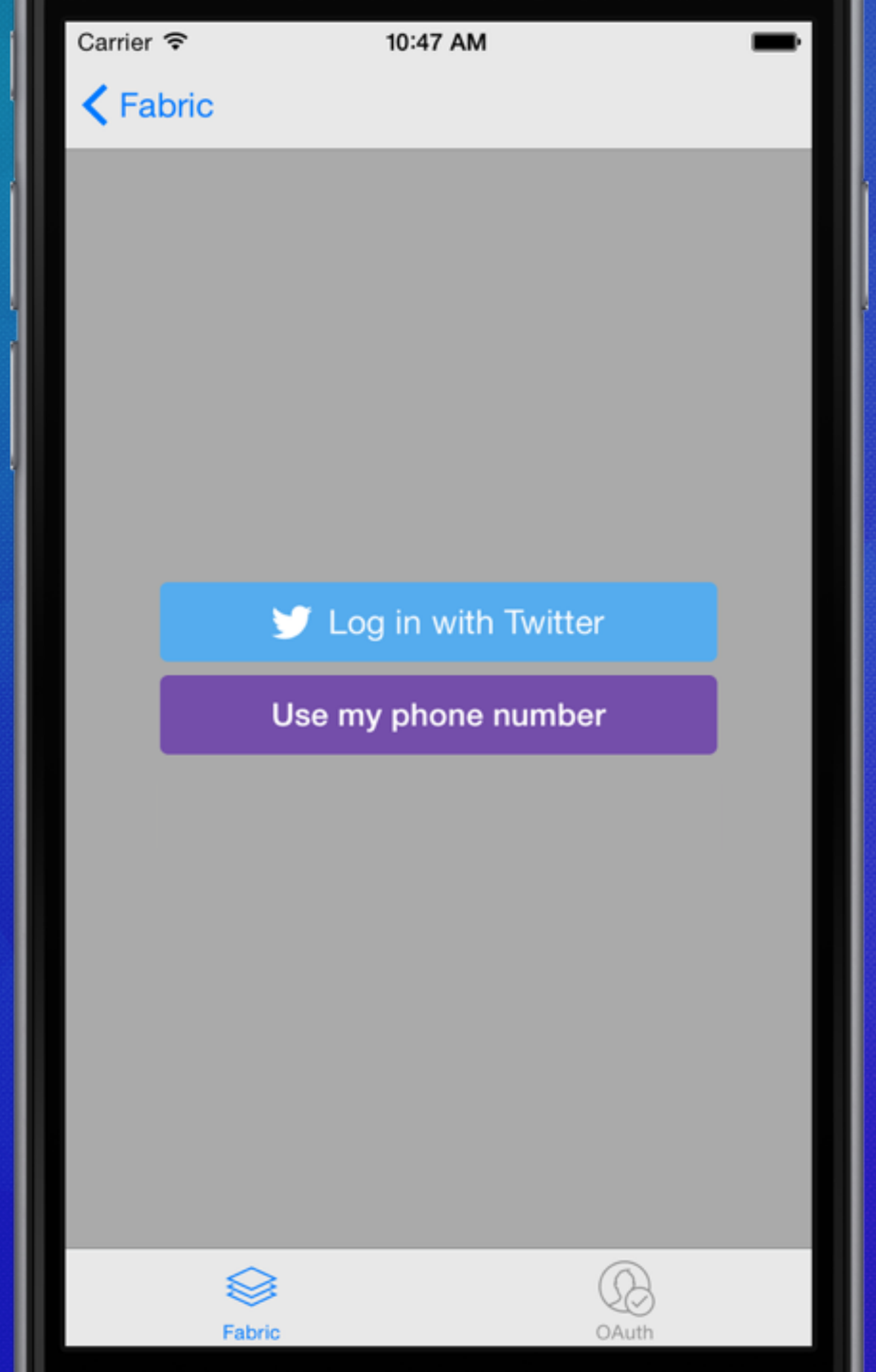
- When and Why?
- Fabric: Out of the Box
- Building an Auth Service

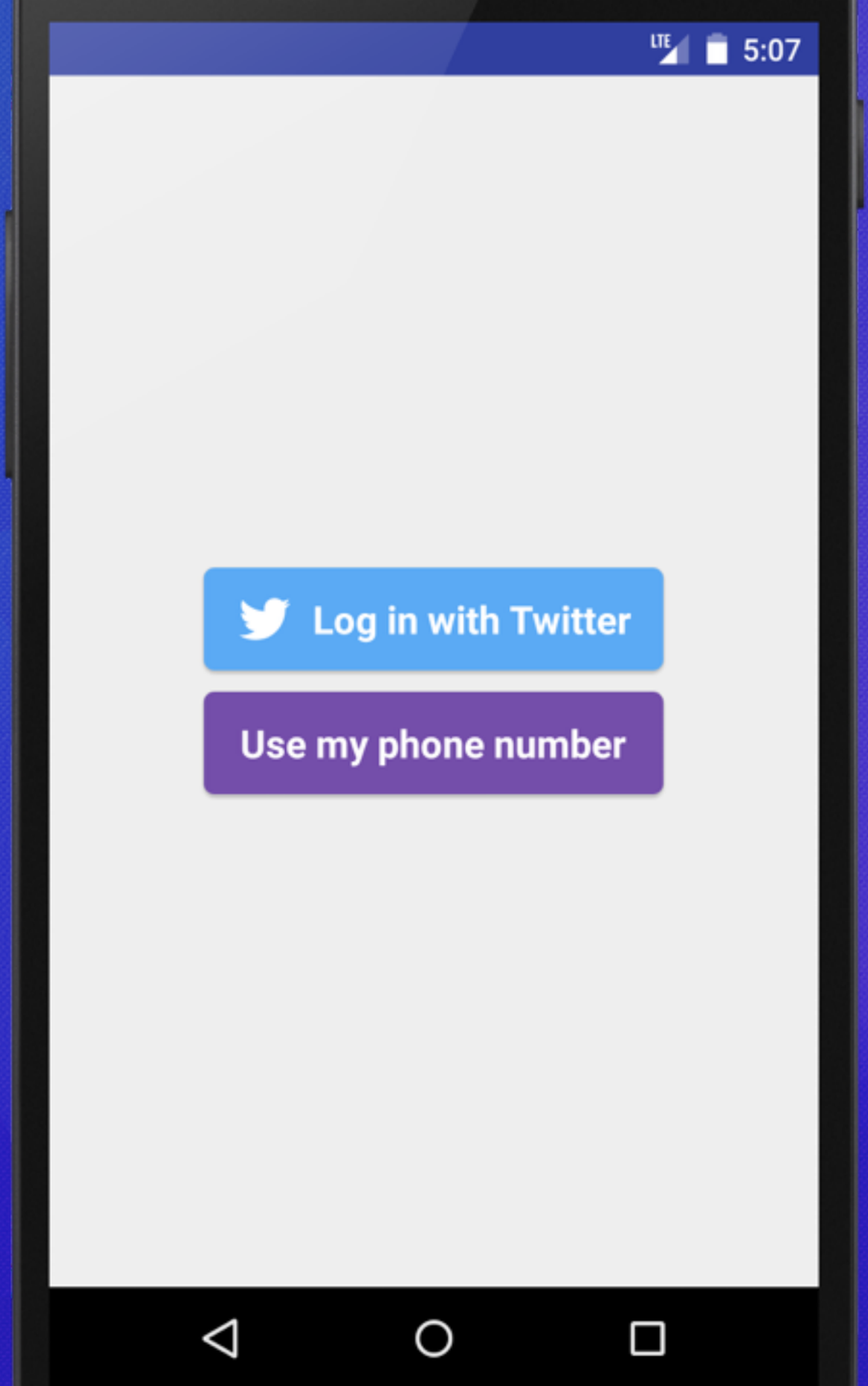

T-Mobile
4:14 PM
Home
San Francisco
The Secret of the Apple's New San Francisco Fonts
iOS 9 is now publicly released. It's a subtl
change but the system fonts of iOS 9 are
changed to the Apple's new San Francisco
Continue reading · 4 min read
96
Rick Turoczy
September 18
Home
Top stories
Bookmarks
1:16
DISCOVER
BOOKMARKS
COMMUNITY
TREND
All Eats And Drinks Near You
Equator Coffee
Coffee
0.6mi
$
Blue Bottle Coffee
Coffee
0.0mi
$
Project Juice
Healthy
0.0mi
$
Alta CA
American
0.1mi
$$
The Beer Ha
Bar
7:21
uplike
MY FACE
@doorstein
@mommynme220
#videogames
@Rakayos
@Monika13
#Earphones

flight

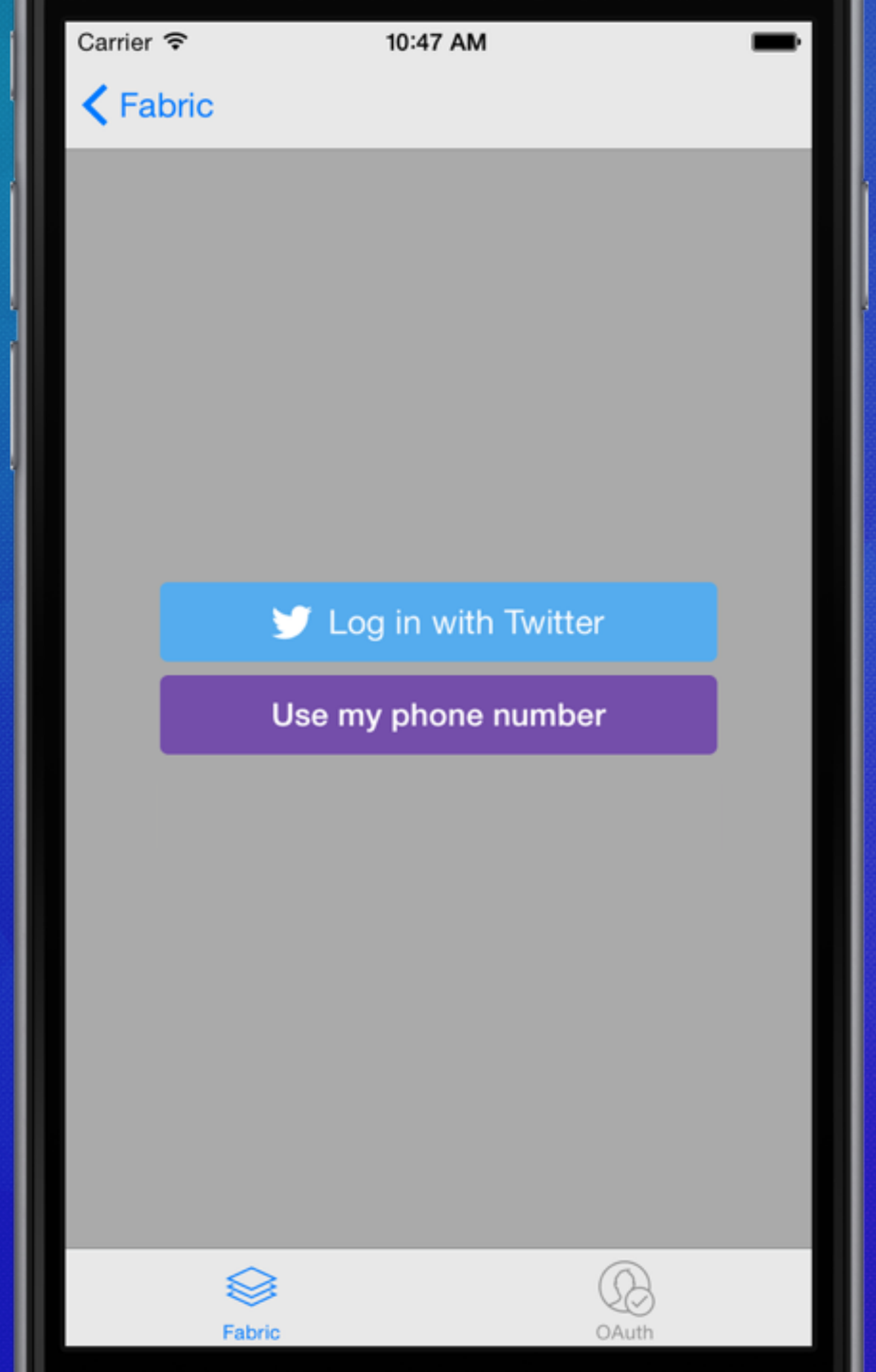

T-Mobile
4:14 PM
Home
San Francisco
The Secret of the Apple's New San Francisco Fonts
iOS 9 is now publicly released. It's a subtl
change but the system fonts of iOS 9 are
changed to the Apple's new San Francisco
Continue reading · 4 min read
96
Rick Turoczy
September 18
Home
Top stories
Bookmarks
1:16
DISCOVER
BOOKMARKS
COMMUNITY
TREND
All Eats And Drinks Near You
Equator Coffee
Coffee
0.6mi
$
Blue Bottle Coffee
Coffee
0.0mi
$
Project Juice
Healthy
0.0mi
$
Alta CA
American
0.1mi
$$
The Beer Ha
Bar
7:21
uplike
MY FACE
@doorstein
@mommynme220
#videogames
@Rakayos
@Monika13
#Earphones

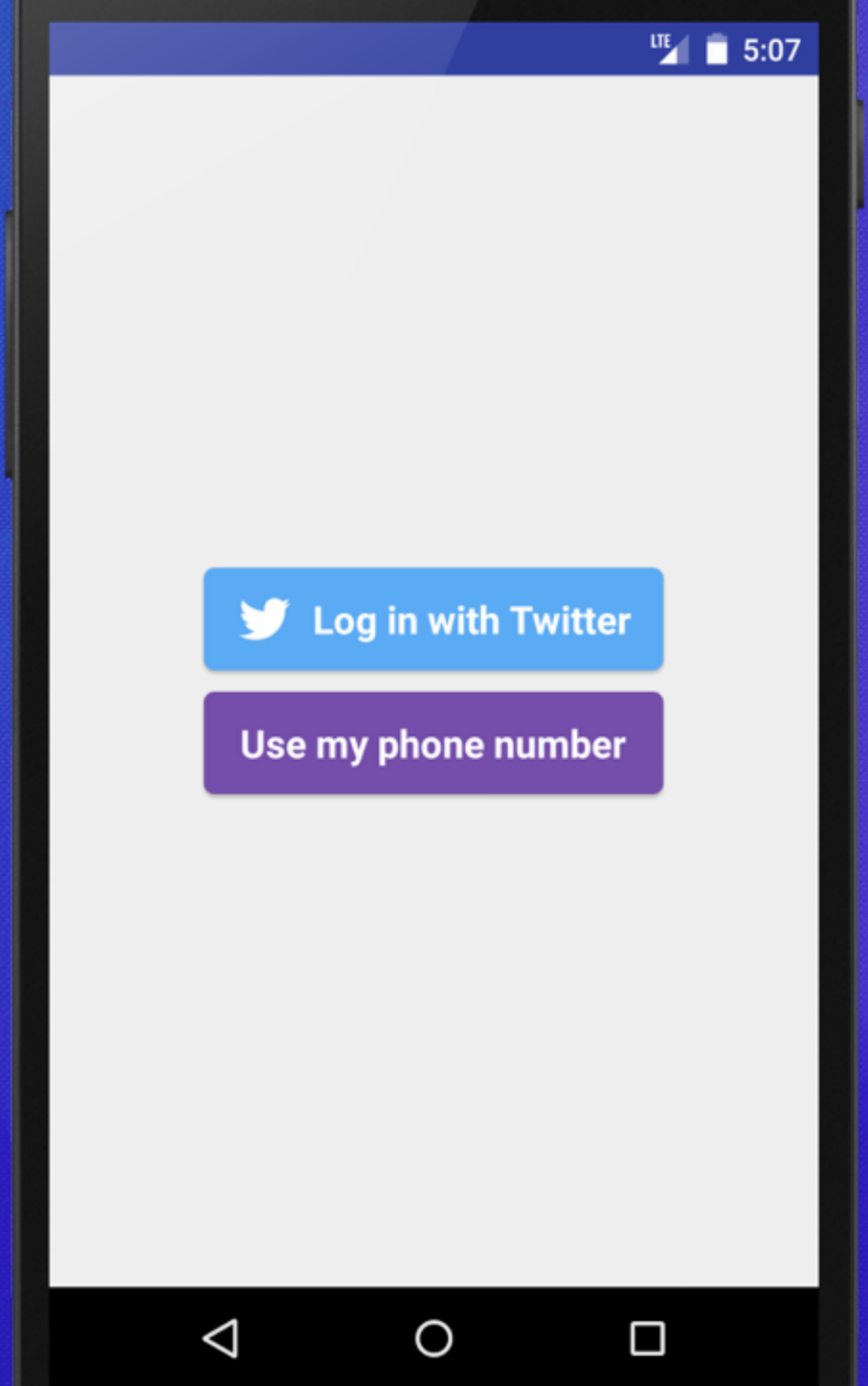
flight


Carrier
10:47 AM
Fabric
Log in with Twitter
Use my phone number
Fabric
OAuth


LTE
5:07
Log in with Twitter
Use my phone number

flight

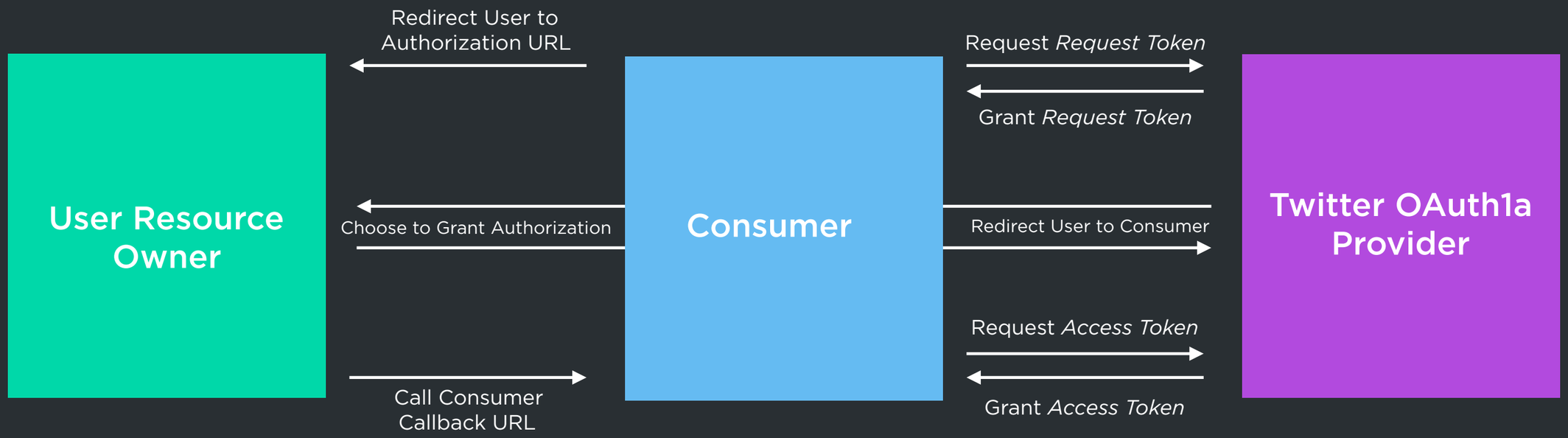

Carrier
10:47 AM
Fabric
Log in with Twitter
Use my phone number
Fabric
OAuth

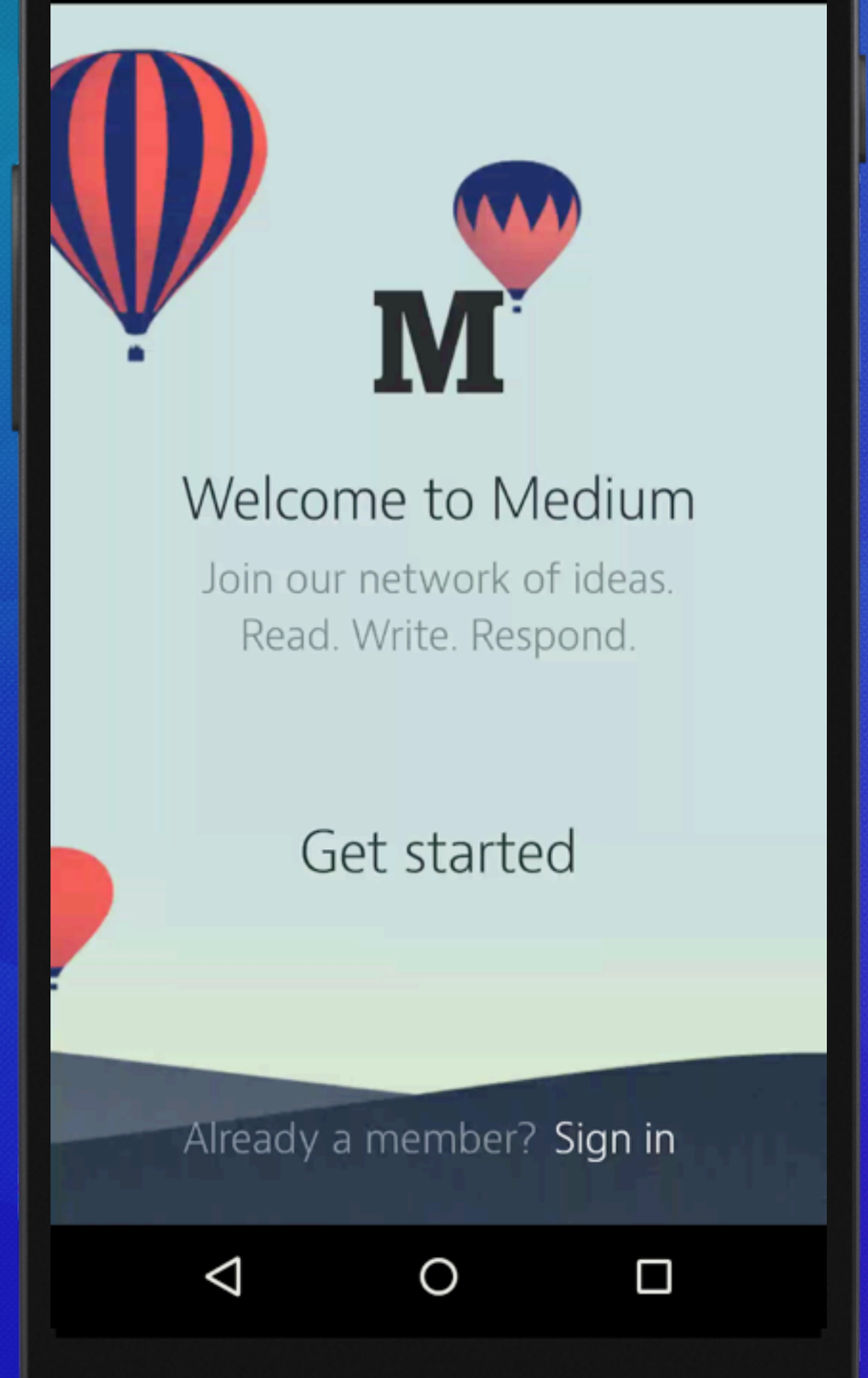

LTE
5:07
Log in with Twitter
Use my phone number

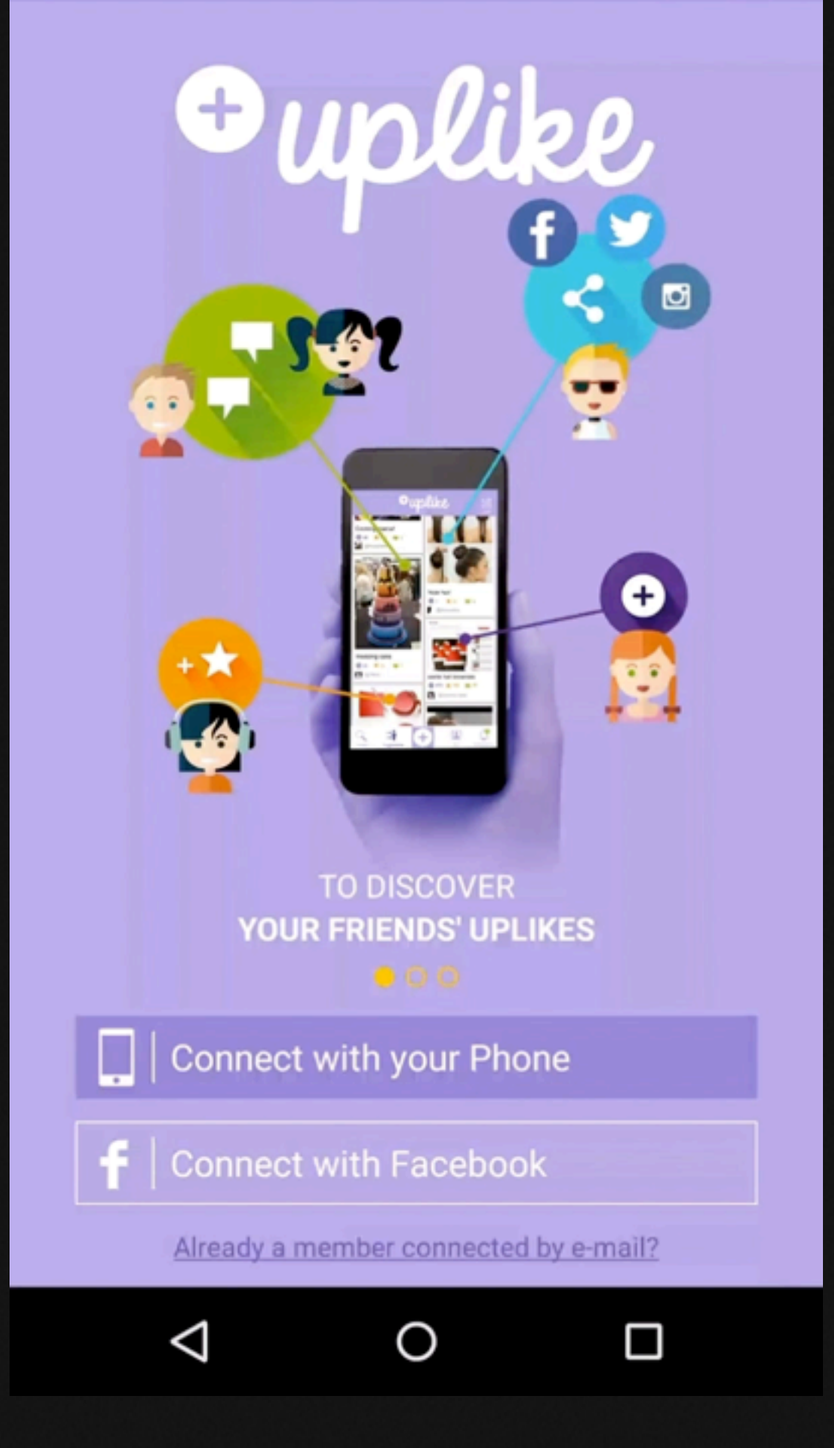
flight

Mobile Single Sign-On

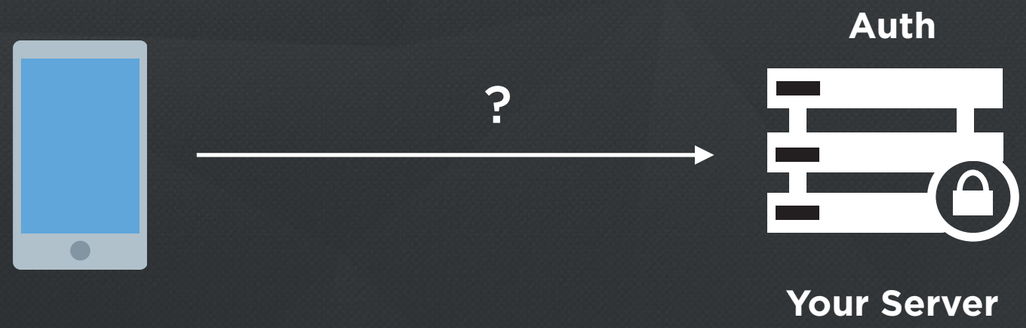
OAuth1a Authorization Flow




Welcome to Medium
Join our network of ideas.
Read. Write. Respond.
Get started
Already a member? Sign in


uplike
TO DISCOVER
YOUR FRIENDS' UPLIKES
Connect with your Phone
Connect with Facebook
Already a member connected by e-mail?

flight


Auth
?
Your Server

Building an Authentication Service

Access Token

Access Token

```
type Token struct {
  Token  string
  Secret string
}
```

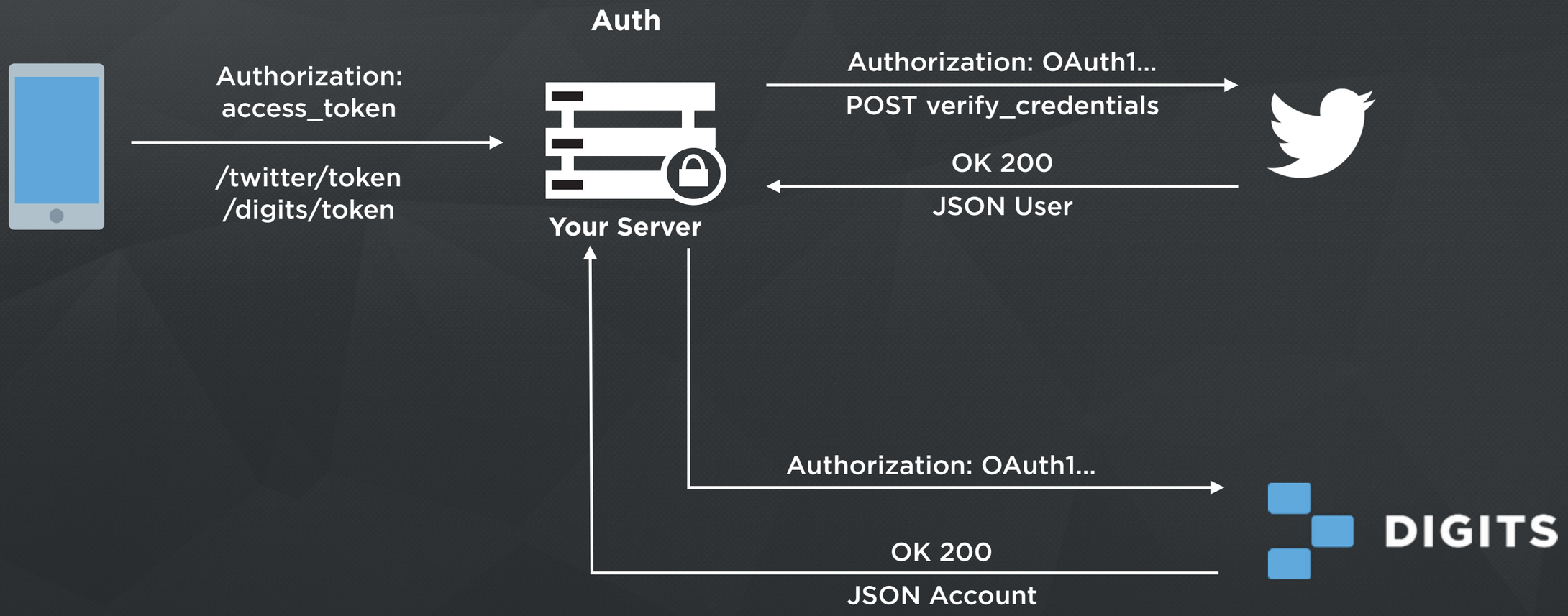
Android

```
TwitterSession.getAuthToken();

DigitsSession.getAuthToken();
```

iOS

```
TWTRSession.authToken
TWTRSession.authTokenSecret

DGTSession.authToken
DGTSession.authTokenSecret
```

Verifying an Identity


Auth
Authorization:
access_token
/twitter/token
/digits/token
Your Server
Authorization: OAuth1...
POST verify_credentials
OK 200
JSON User
Authorization: OAuth1...
OK 200
JSON Account
DIGITS

flight

GET https://api.twitter.com/1.1/account/verify_credentials.json

```
type User struct {
    ID          int64  `json:"id"`
    IDStr       string `json:"id_str"`
    Name        string `json:"name"`
    ScreenName string `json:"screen_name"`
    Status      *Tweet `json:"status"`
    ...
}
```

GET https://api.digits.com/1.1/sdk/account.json

```
type Account struct {
    AccessToken AccessToken `json:"access_token"`
    ID          int64       `json:"id"`
    IDStr       string      `json:"id_str"`
    CreatedAt   string      `json:"created_at"`
    PhoneNumber string      `json:"phone_number"`
}
```

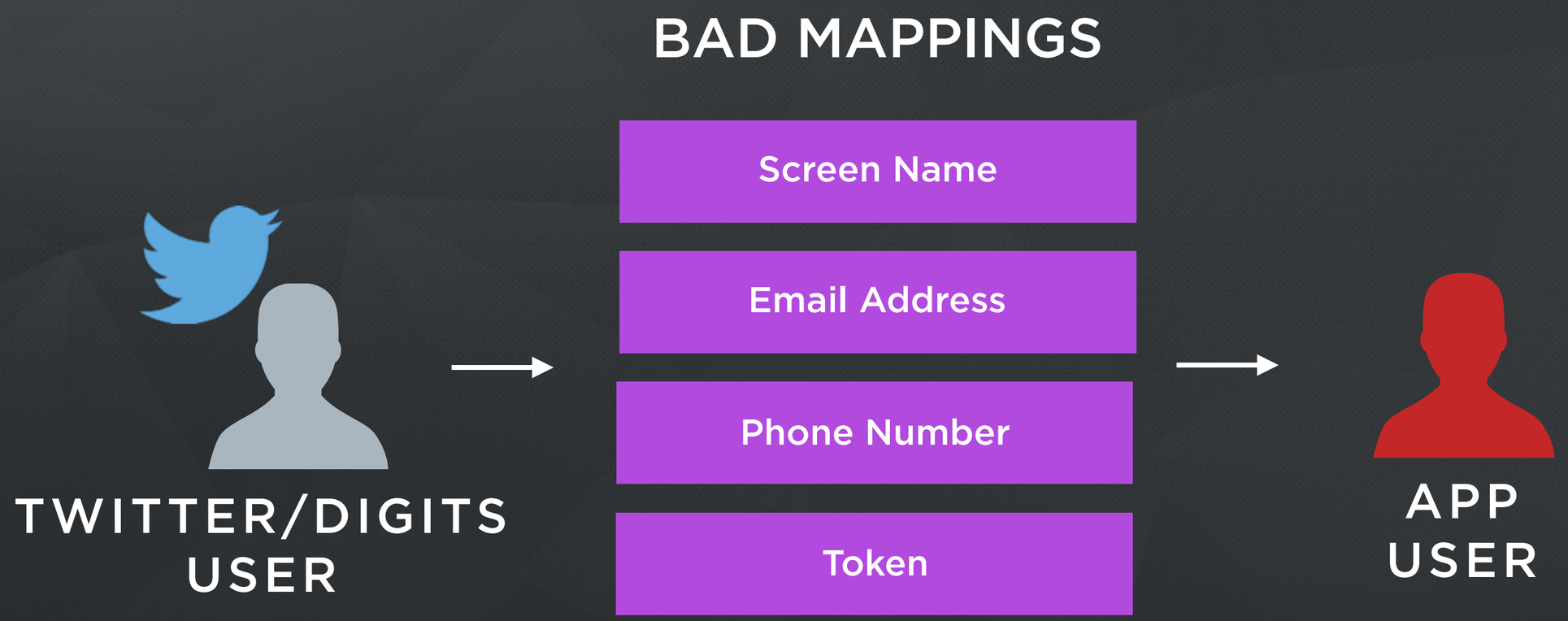
```
type ContextHandler interface {
   ServeHTTP (ctx context.Context, w http.ResponseWriter, req *http.Request)
}

func NewHandler(h ContextHandler) http.Handler
```

```
func TokenHandler(config *oauth1.Config, success, failure ContextHandler) ContextHandler {
    fn := func(ctx context.Context, w http.ResponseWriter, req *http.Request) {
        // require POST
        req.ParseForm()
        accessToken := req.PostForm.Get(accessTokenField)
        accessSecret := req.PostForm.Get(accessTokenSecretField)
        err := validateToken(accessToken, accessSecret)
        if err != nil {
            failure.ServeHTTP(WithError(ctx, err), w, req)
            return
        }
        ctx = WithAccessToken(ctx, accessToken, accessSecret)
        success.ServeHTTP(ctx, w, req)
    }
    return ContextHandlerFunc(fn)
}
```

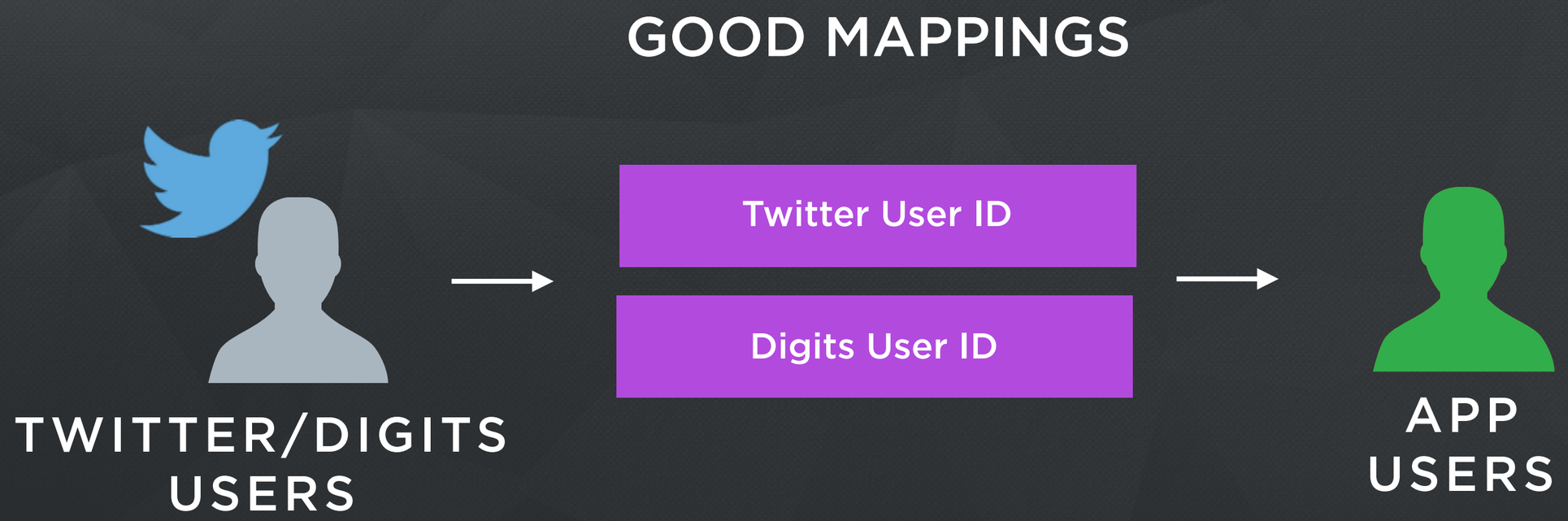
```
func verifyUser(config *oauth1.Config, success, failure ContextHandler) ContextHandler {
    fn := func(ctx context.Context, w http.ResponseWriter, req *http.Request) {
        accessToken, accessSecret, err := AccessTokenFromContext(ctx)
        // handle err

        }
        ctx = WithTwitterUser(ctx, user)
        success.ServeHTTP(ctx, w, req)
    }
    return ContextHandlerFunc(fn)
}
```

```
func verifyUser(config *oauth1.Config, success, failure ContextHandler) ContextHandler {
  fn := func(ctx context.Context, w http.ResponseWriter, req *http.Request) {
    accessToken, accessSecret, err := AccessTokenFromContext(ctx)
    // handle err
    httpClient := config.Client(ctx, oauth1.NewToken(accessToken, accessSecret))
    twitterClient := twitter.NewClient(httpClient)
    user, resp, err := twitterClient.Accounts.VerifyCredentials()
    err = validateResponse(user, resp, err)
    if err != nil {
      failure.ServeHTTP(WithError(ctx, err), w, req)
      return
    }
    ctx = WithTwitterUser(ctx, user)
    success.ServeHTTP(ctx, w, req)
  }
  return ContextHandlerFunc(fn)
}
```

Mapping To App Users



Mapping To App Users

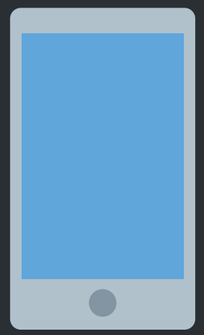


```
user, err := TwitterUserFromContext(ctx)

// lookup Notes App User by Twitter ID
account, err := accountClient.GetByTwitterID(ctx, user.ID)
if err != nil {
   // valid Twitter user, but does not correspond to an account
}
```

```
digitsAccount, err := DigitsUserFromContext(ctx)

// lookup Notes App User by Digits ID
account, err := accountClient.GetByDigitsID(ctx, digitsAccount.ID)
if err != nil {
   // valid Digits account, but does not correspond to an account
}
```

Pitfalls

- Use the Response



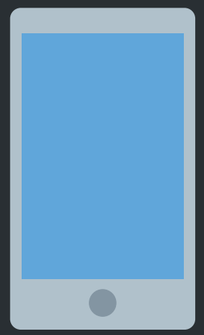
POST api.noteapp.com/digits/token

user_id=123
digit_token=token
digit_secret=secret

Auth

Your Server

flight



POST api.noteapp.com/digits/token

~~user_id=123~~
digit_token=token
digit_secret=secret

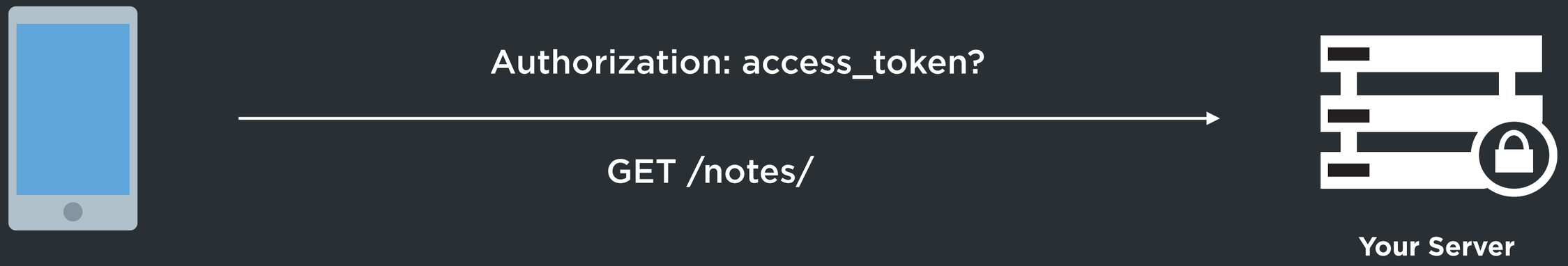
Auth

Your Server

flight

Pitfalls

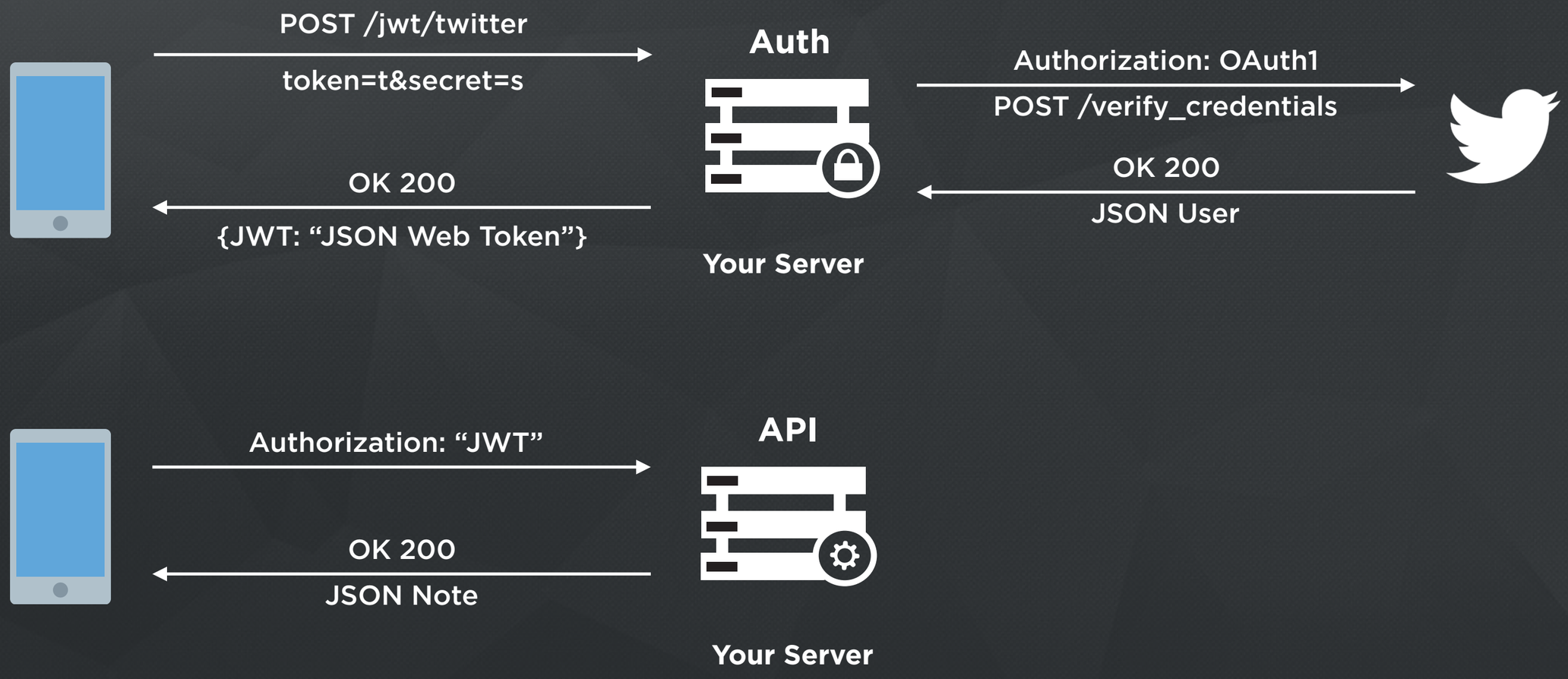
- Use the Response
- HTTPS/TLS
- API is not Private
- Digits Web Verifications


Auth
Authorization: access_token?
GET /notes/
Your Server

Goals

- Low Latency
- Expiration
- Revocation
- Metadata

Token Session



flight

Header

```
{"alg": “HS256", "typ": "JWT"}
```

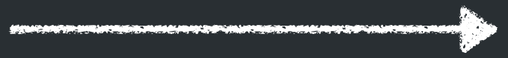
```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
```

Payload

```
{“id”: “1234”, “iat”: 123456789, “exp”: 123456789}
```

```
eyJhdWQiOiJtb2JpbGUiLCJleHAiOjE0NzA2NDY4ND
gsImlzcyI6InRva2VuLnNlcnZpY2UiLCJzdWIiOiJk
Z2h1YmJsZSJ9
```

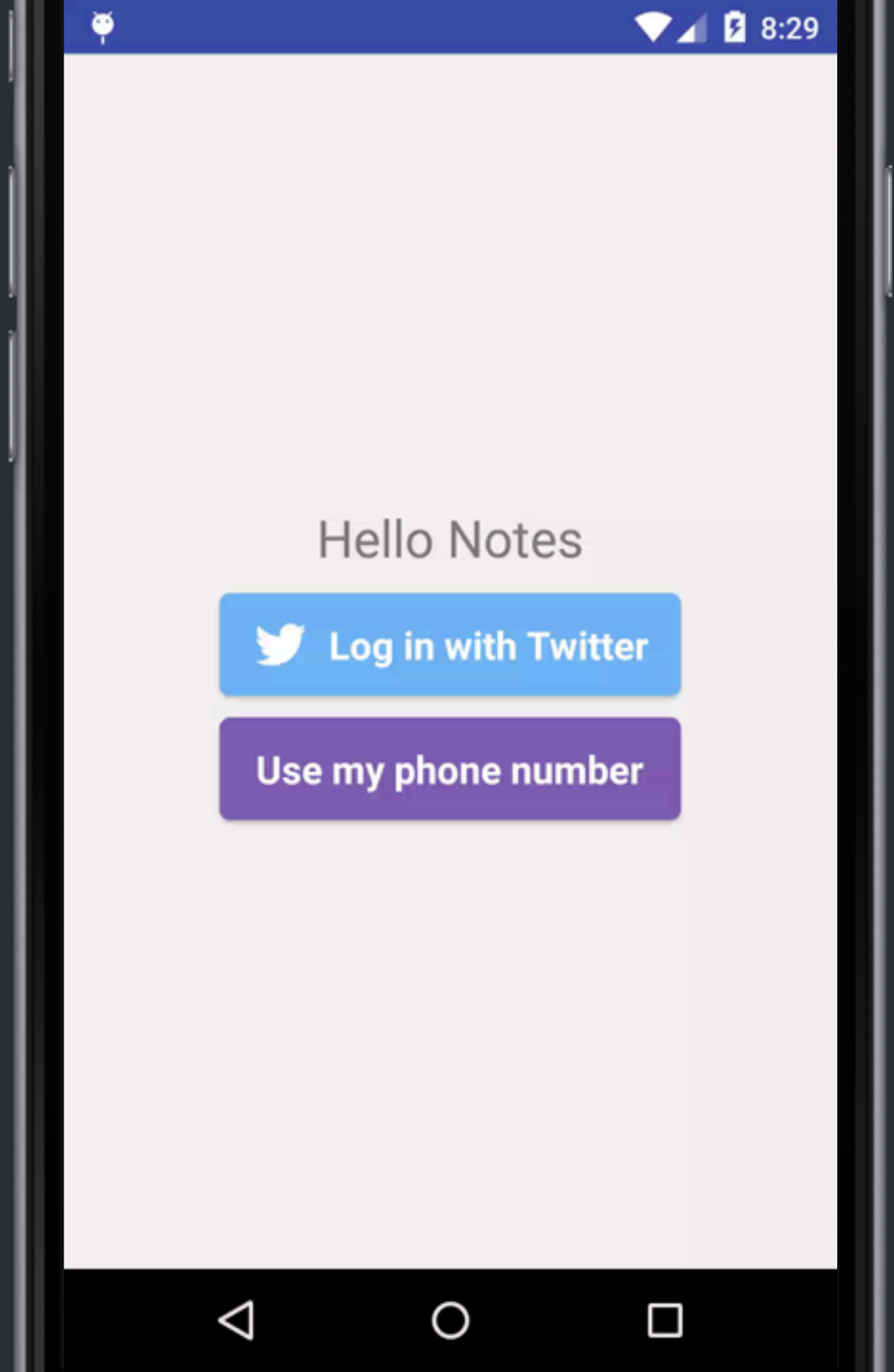
Signature

signature 

```
NBh0NUcOg34MoszOuyG-rAkskatslNwjKNsiSuG4D8U
```

flight

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJtb2JpbGUiLCJleHAiOjE0NzA2ND
Y4NDgsImlzcyI6InRva2VuLnNlcnZpY2UiLCJzdWIiOiJkZ2h1YmJsZSJ9.NBh0NUcOg34MoszO
uyG-rAkskatslNwjKNsiSuG4D8U
```



Auth Service

Account Service

Note Service

flight

Resources

- docs.fabric.io/web/digits/getting-started.html
- github.com/dghubble/gologin
- github.com/dgrijalva/jwt-go

Thank You

@dghubble

flight