



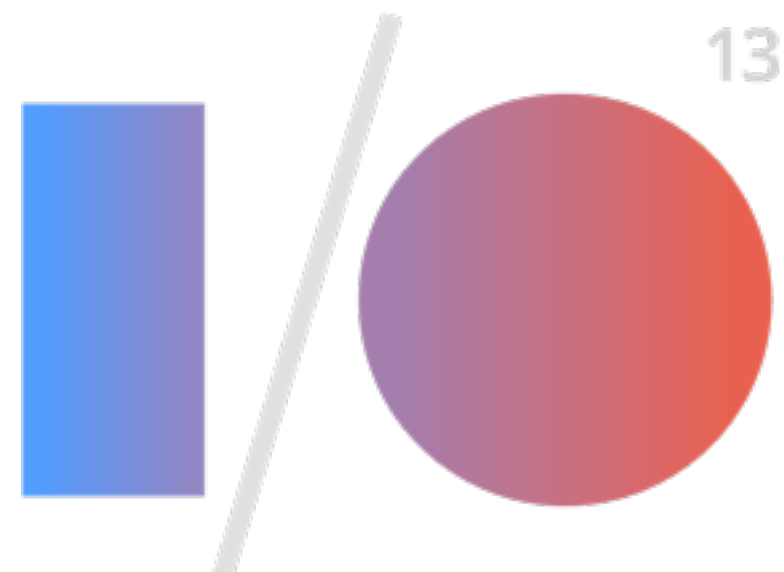
Google  
**Developers**



# How to offer Google+ alongside other social sign-in options

Managing Multiple Authentication Providers

Ian Barber  
Google+ Developer Advocate





Sign in with Google



**Log In**

**Sign In**

**Connect**



**Sign in with Google**

**Connect**

**Sign In**

**Sign Up**



**Authentication &  
Authorization**

**Capabilities**

**Data Model**

**User Story**

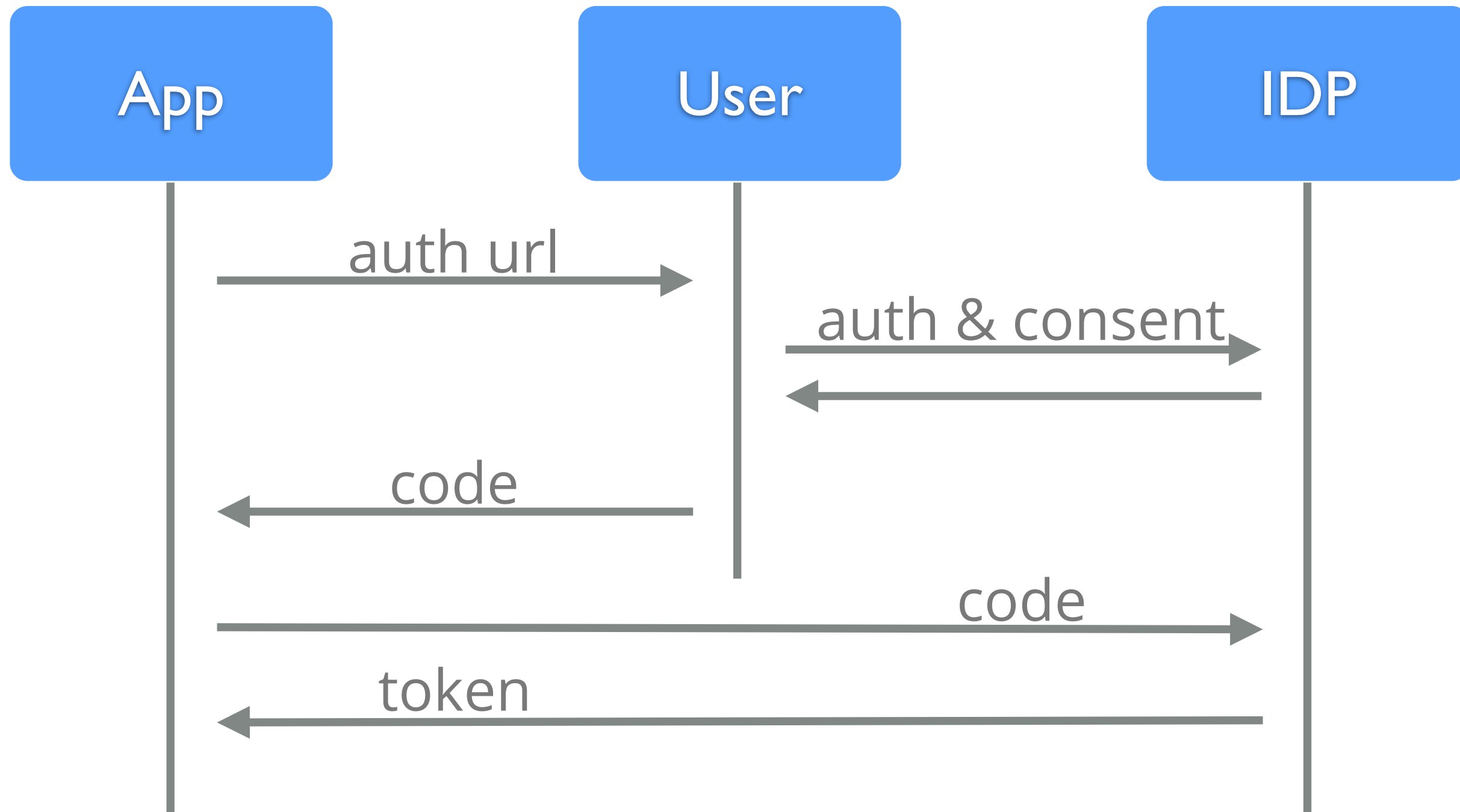


# OAuth Everywhere

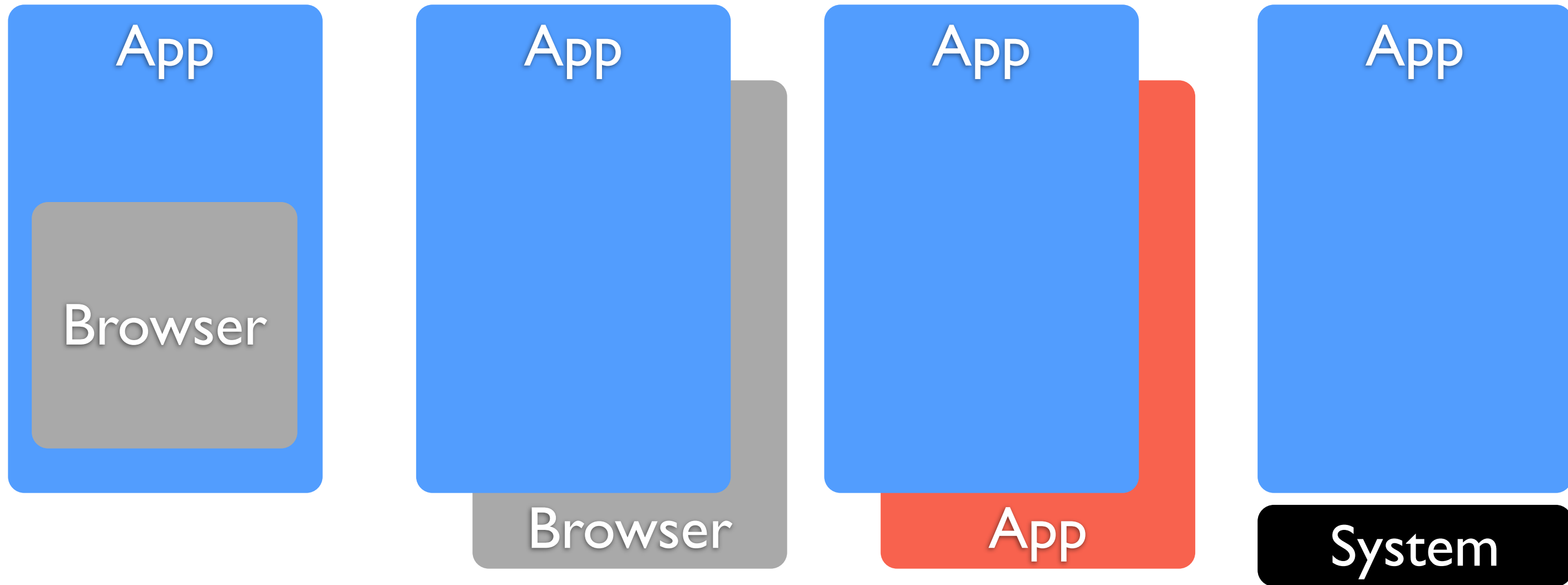


OAuth Logo: <http://wiki.oauth.net>

# Patterns of Web Authentication

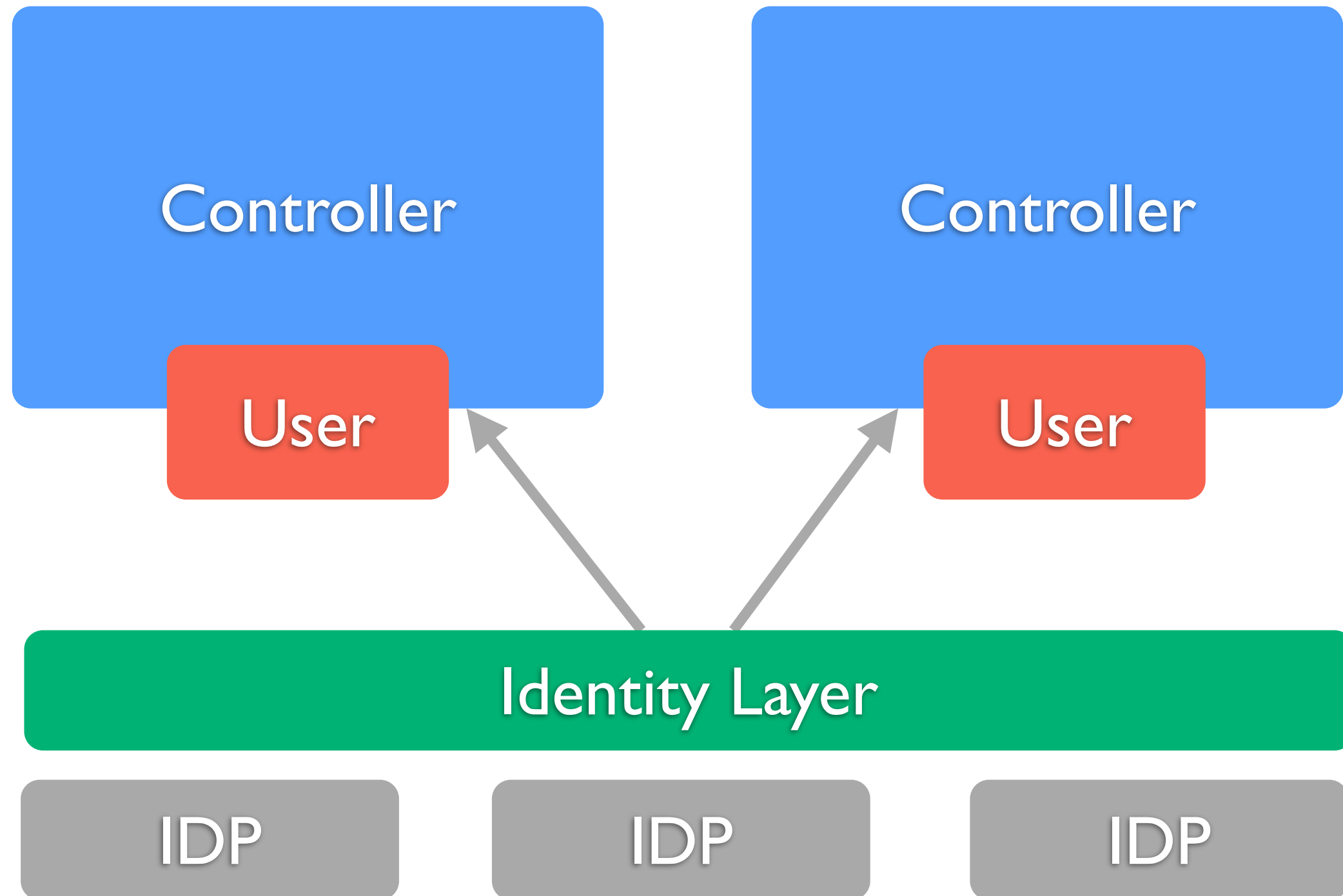


# Patterns of Mobile Authentication





# What Do We Want?



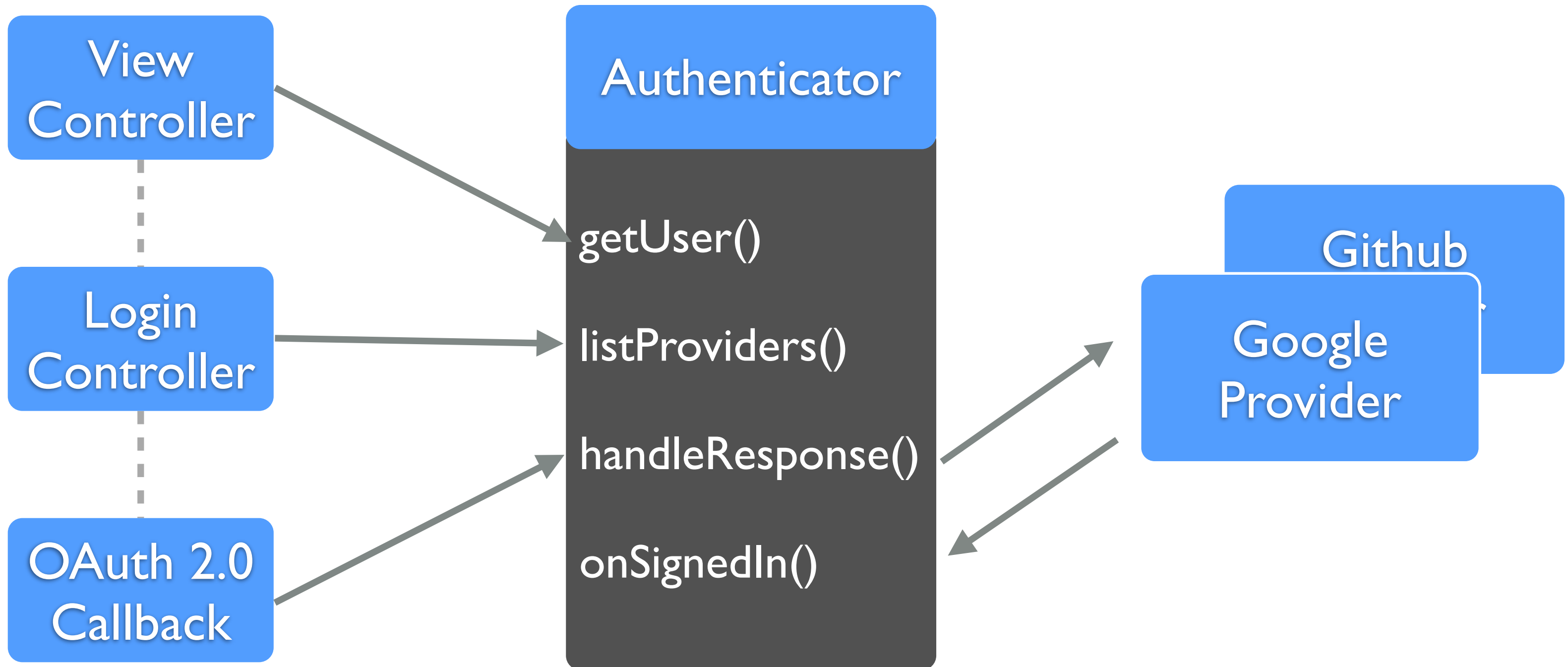
# Application Authentication Interaction

PHP

```
public function handle($path, $unsafe_request) {  
    // Redirect to login if not logged in.  
    if($this->auth->getUser() == null) {  
        header("Location: /login");  
        exit;  
    }  
    $user = $this->auth->getUser();  
    $content =  
        sprintf("<h1>Hi %s</h1>", $user->getName());  
}
```



# Mediating Access



# Authentication Strategies

Check State

Strategy

check token

Sign In

Strategy

return markup

return code

Verify

Strategy

retrieve token

verify with server

retrieve data

Provide User



# Provider

PHP

```
<?php
interface Provider {
    public function getId();
    public function checkState();
    public function getMarkup();
    public function getScript();
    public function validate($unsafe_request);
    public function setCallback($callback);
}
```



# Strategy - Github

PHP

```
private function getAuthUrl($options) {
    $bytes = openssl_random_pseudo_bytes(8);
    $state = bin2hex($bytes);
    $params = array_merge(array(
        'client_id' => $this->client_id,
        'state' => $state,
        'redirect_uri' => $this->redirect_uri
    ), $options);
    return self::AUTH_URL.http_build_query($params);
}
```



# Strategy - Github

PHP

```
public function getMarkup($options = array()) {  
    $url = $this->getAuthUrl($options);  
    return sprintf('<a class= "btn" href="%s">Sign In  
        With Github</a>', $url);  
}
```



# Strategy - Github

PHP

```
public function validate($unsafe_request) {  
    // ... get $unsafe_request['code']  
    $context = stream_context_create($opts);  
    $response = file_get_contents(self::TOKEN_URL,  
        false, $context);  
    $token = json_decode($response);  
  
    if(isset($token->access_token)) {  
        // ...  
    }  
}
```





# Strategy - Google+

PHP

```
public function getMarkup() {  
    return sprintf('<span  
        class="g-signin"  
        data-callback="signInCallback"  
        data-clientid="%s"  
        data-cookiepolicy="single_host_origin">  
        </span>', $this->client_id);  
}
```



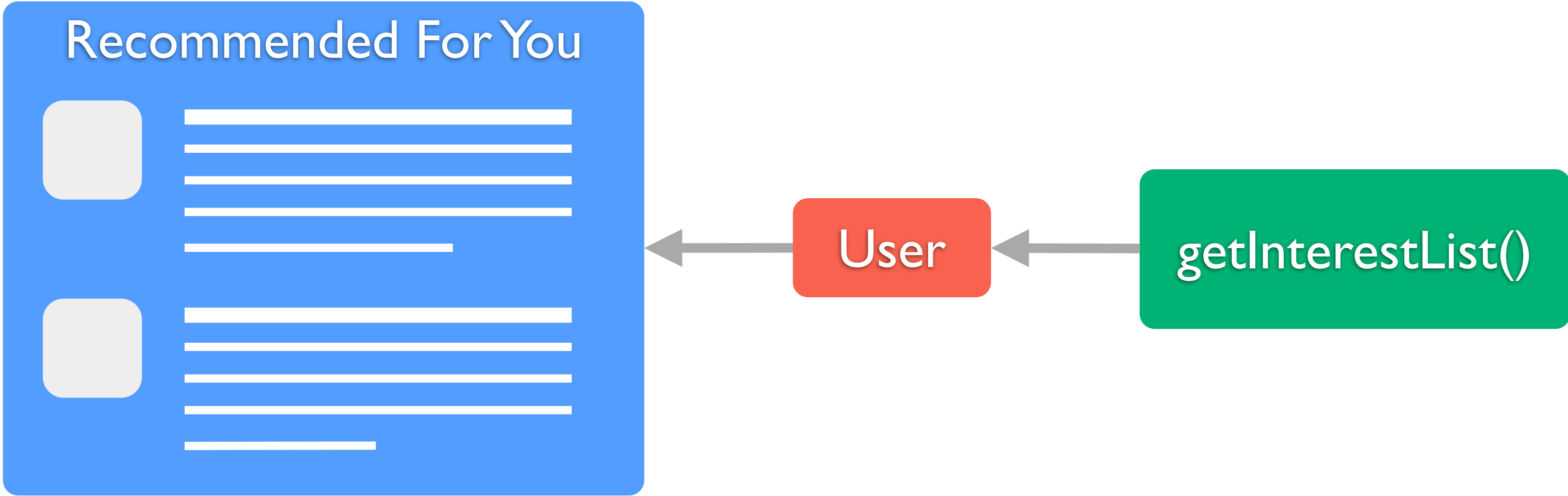
# Strategy - Google+

PHP

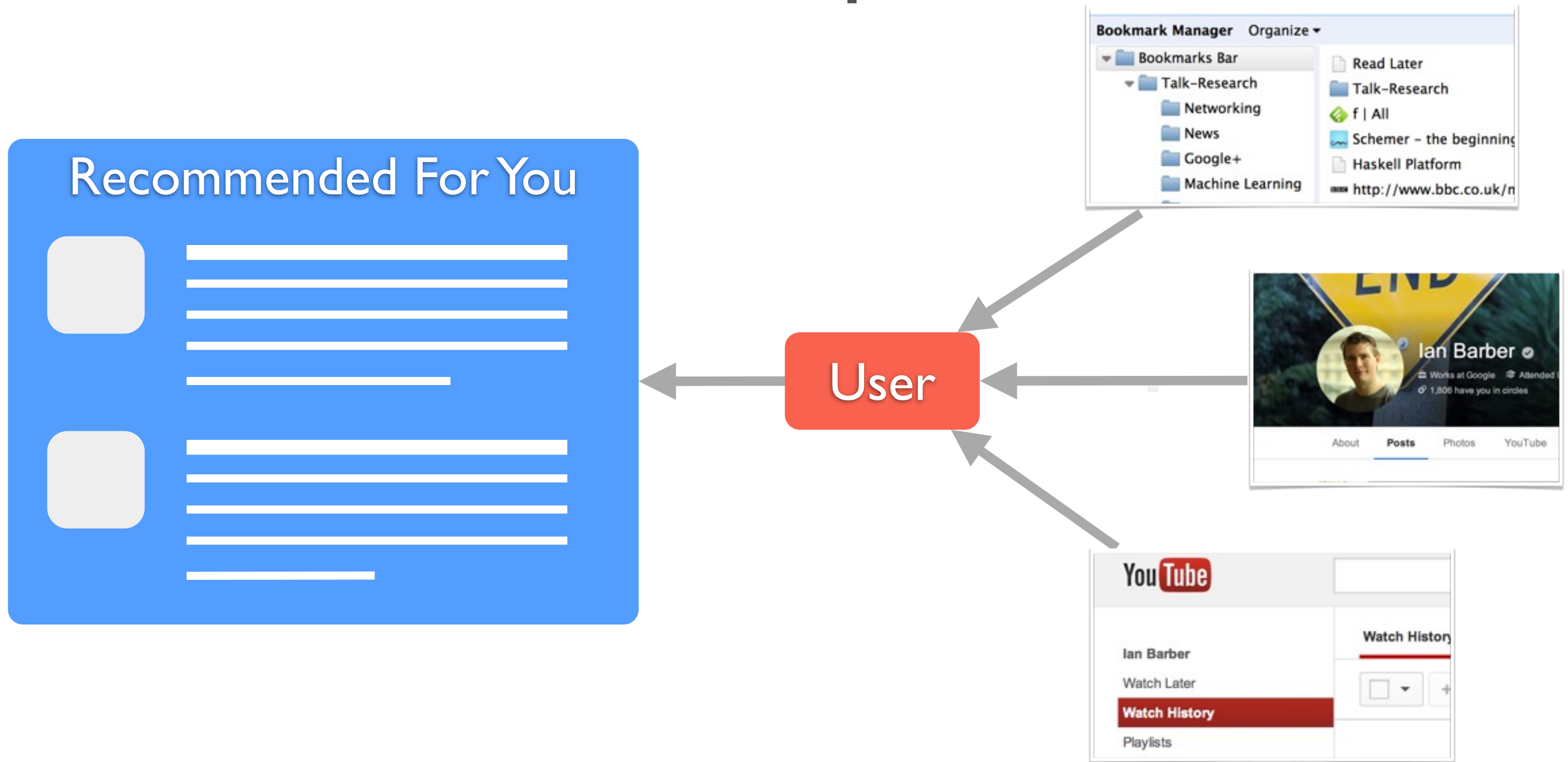
```
public function validate($unsafe_request) {  
    $this->gclient->authenticate  
        ($unsafe_request['code']);  
  
    $_SESSION[self::TAG]['token'] =  
        $this->gclient->getAccessToken();  
    $_SESSION[self::TAG]['user'] =  
        $this->plus->people->get("me");  
    // ...  
}
```



# How can we enable richer experiences?



# How can we enable richer experiences?



# Representing The User

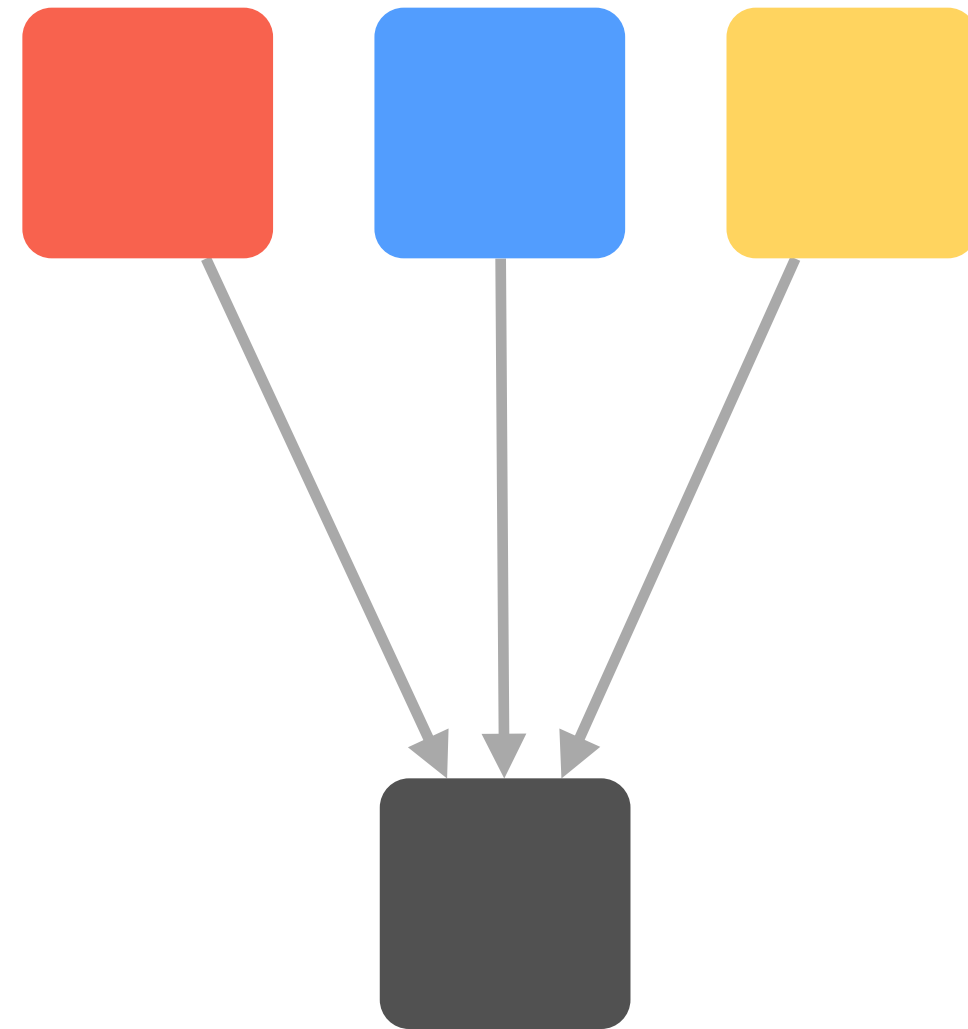
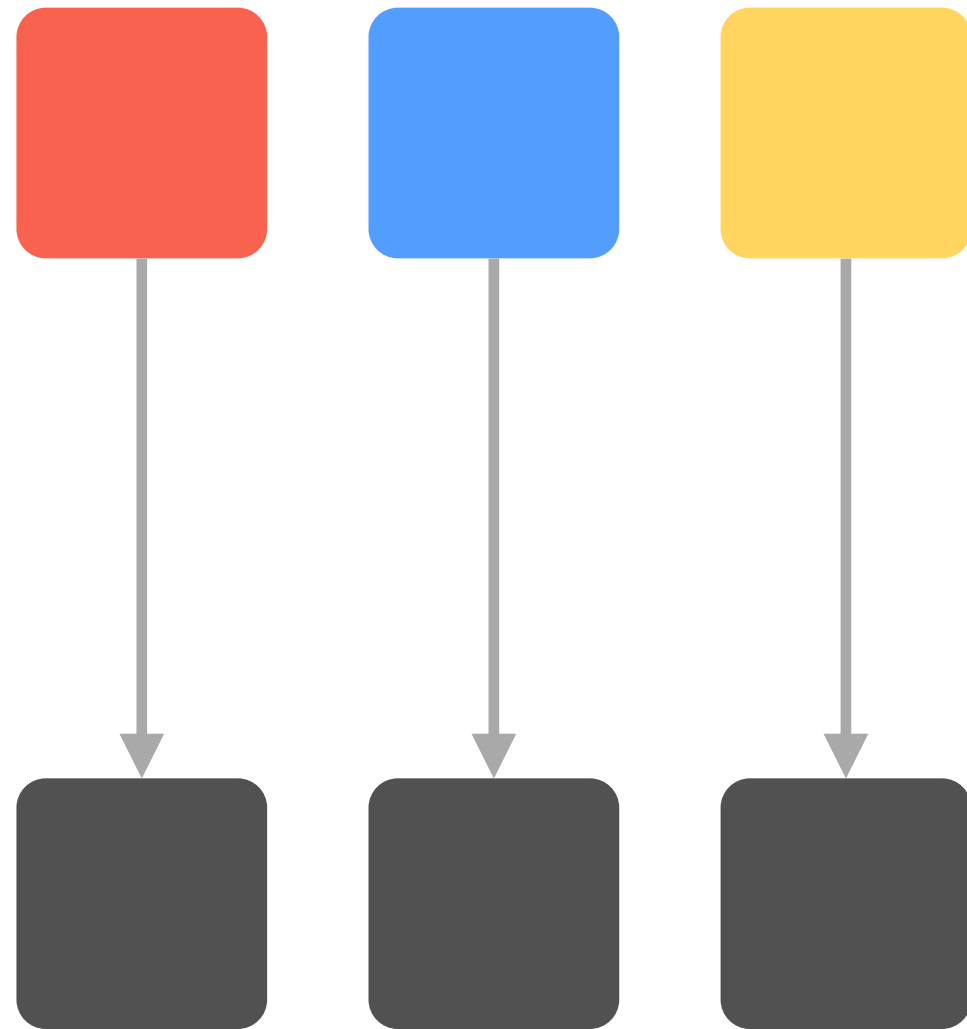
PHP

```
<?php
```

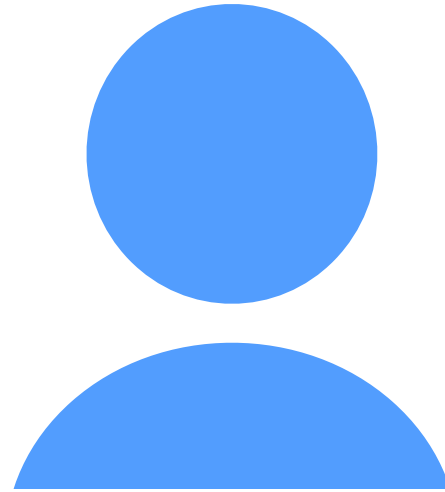
```
interface User {  
    public function getProvider();  
    public function getId();  
    public function getName();  
    public function hasFeature($feature);  
    public function signOut();  
    public function disconnect();  
}
```



# Sign In or Sign Up?



# Who Is The User?



**(service, identifier)**

(google, 104824858261236811362)

(facebook, 560561839)

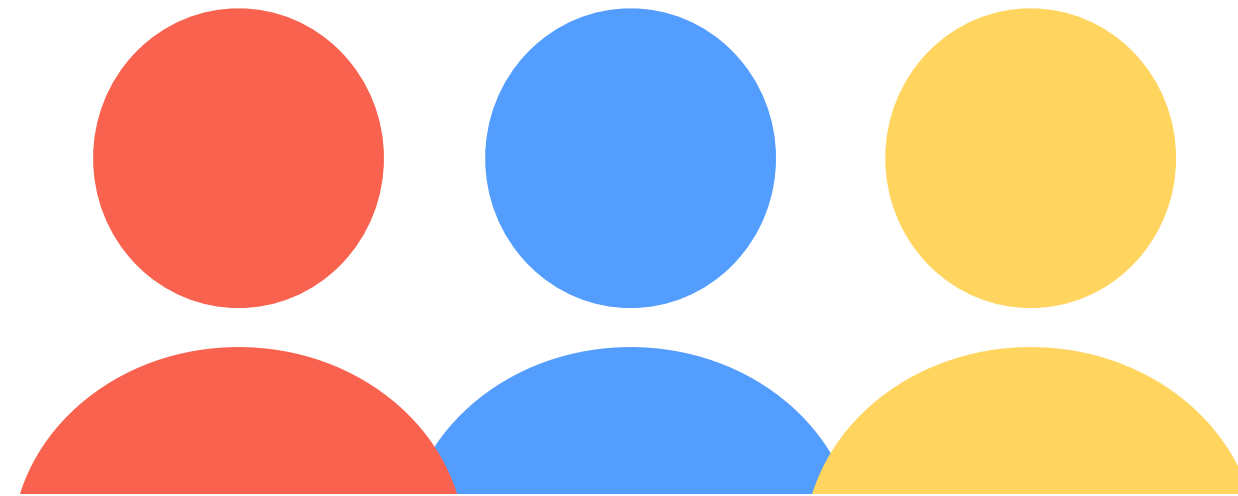
(linkedin, 10492836)



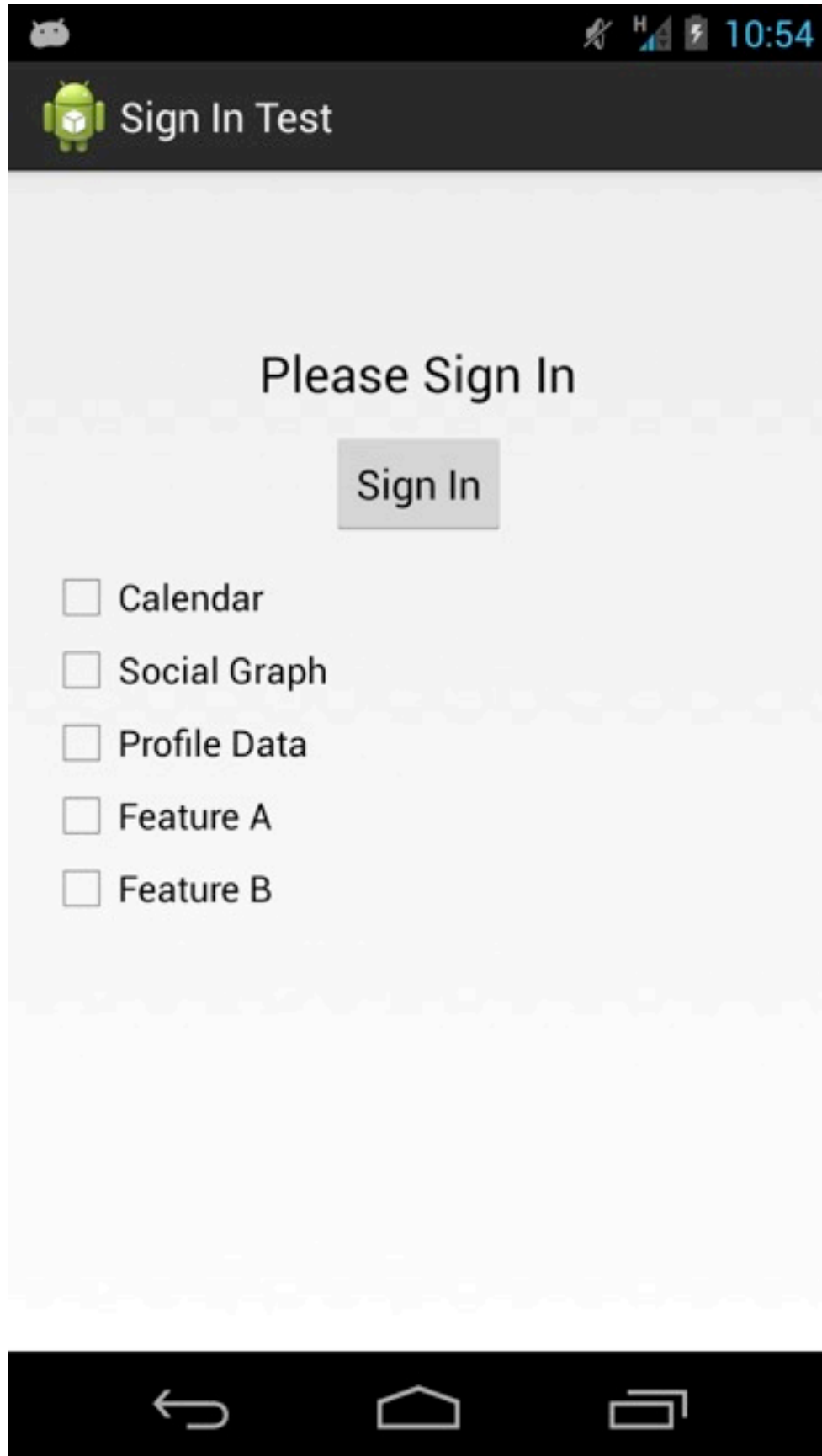
View your email address

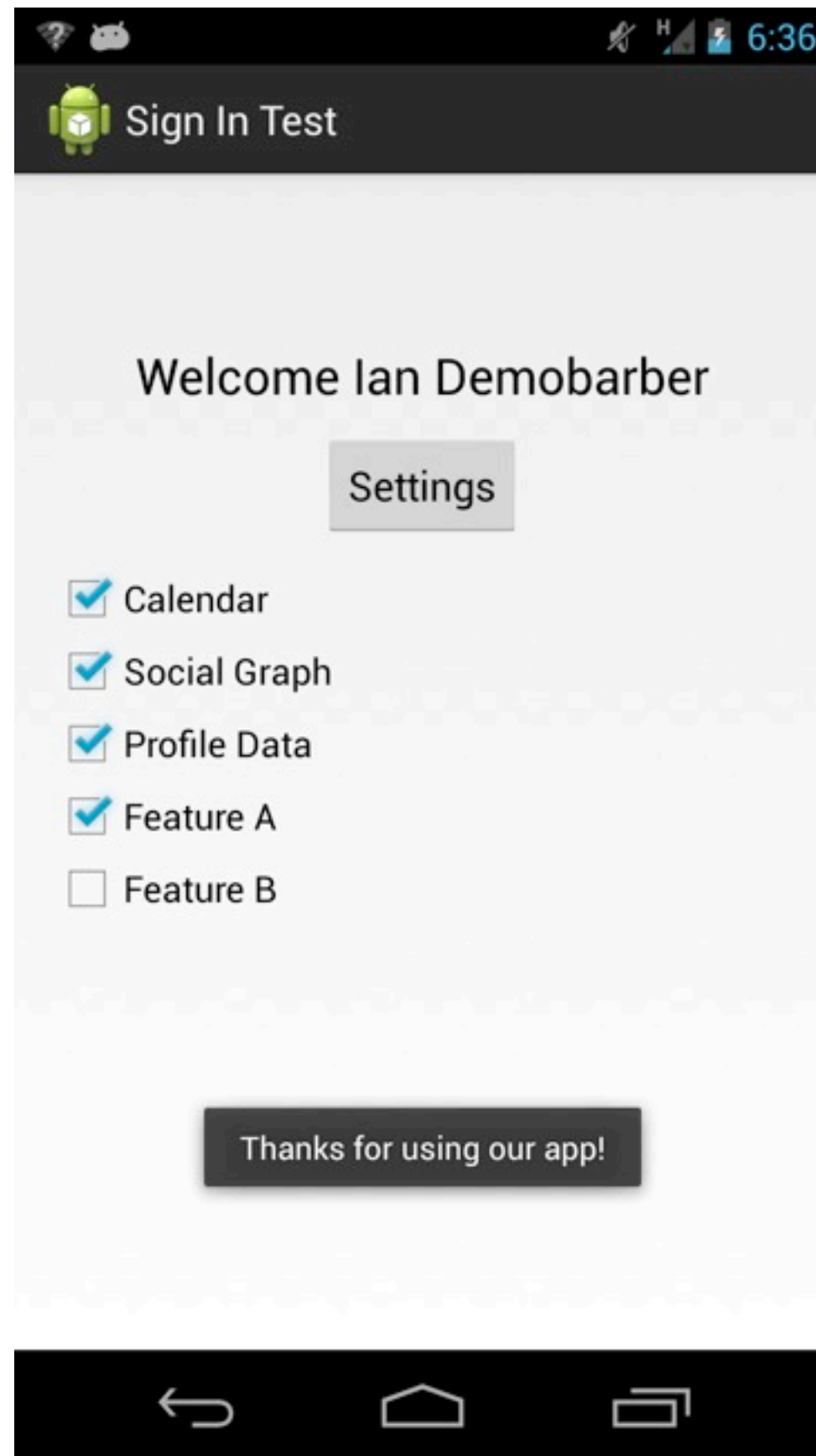
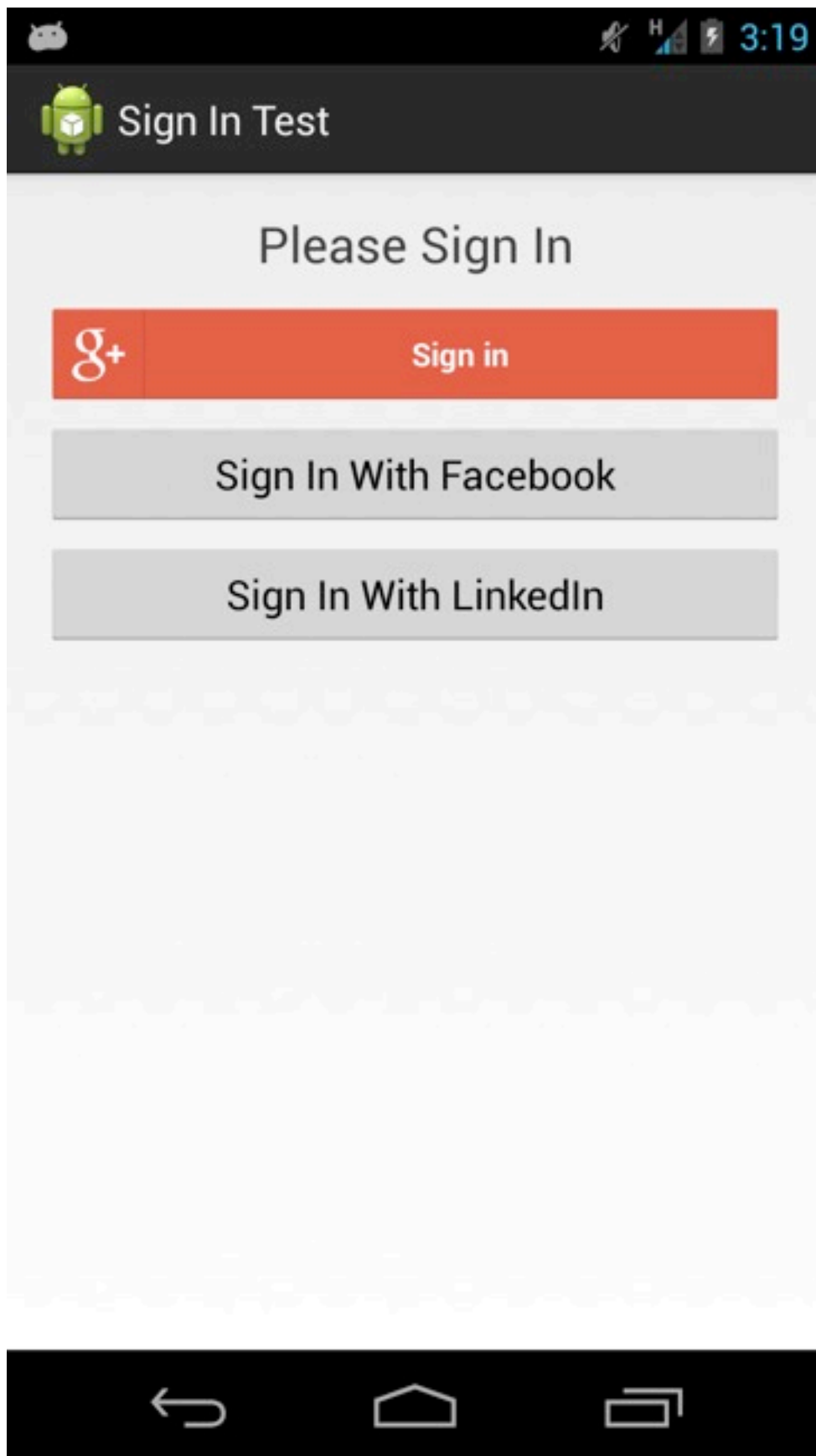


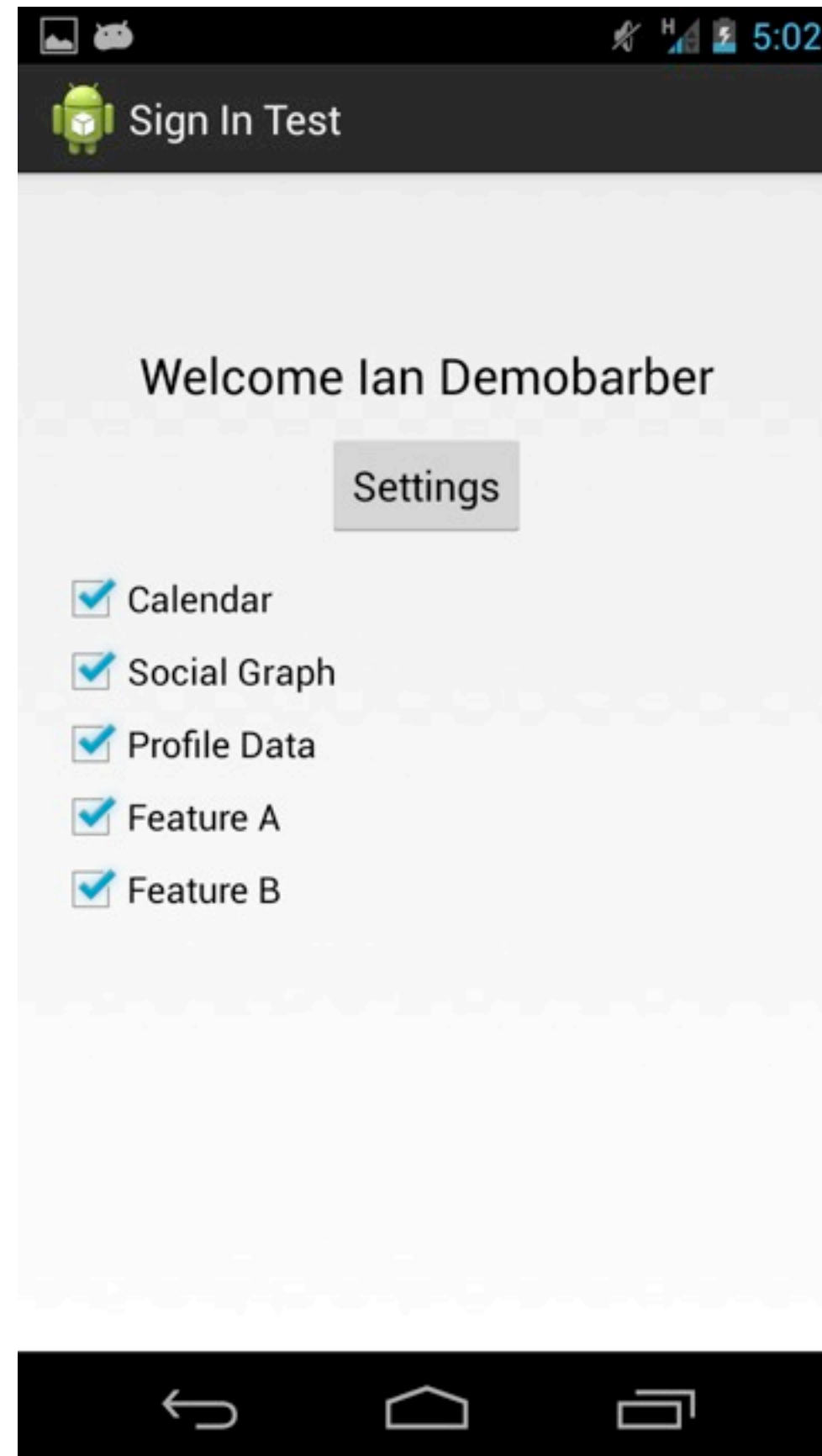
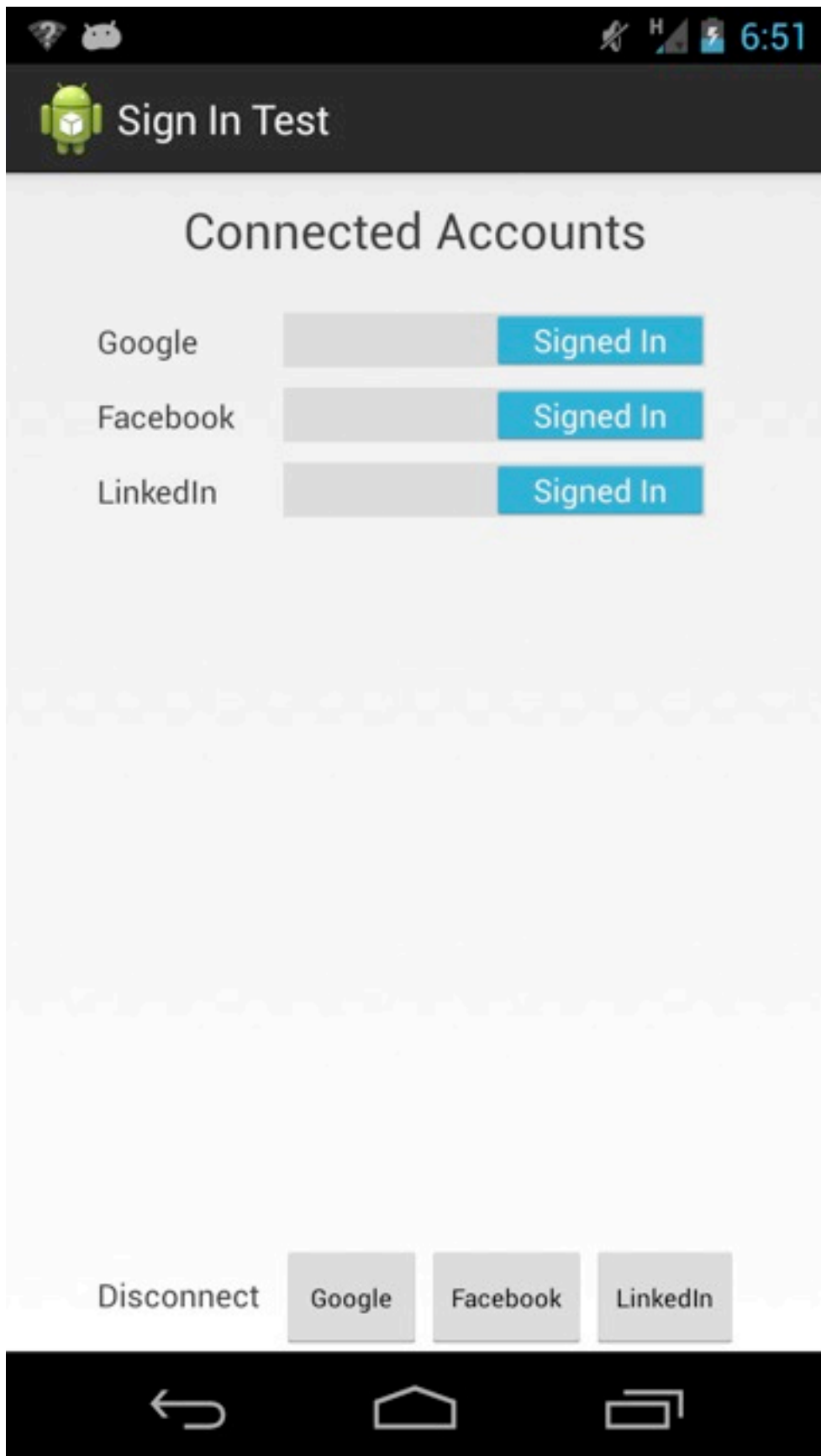
# Combining Accounts



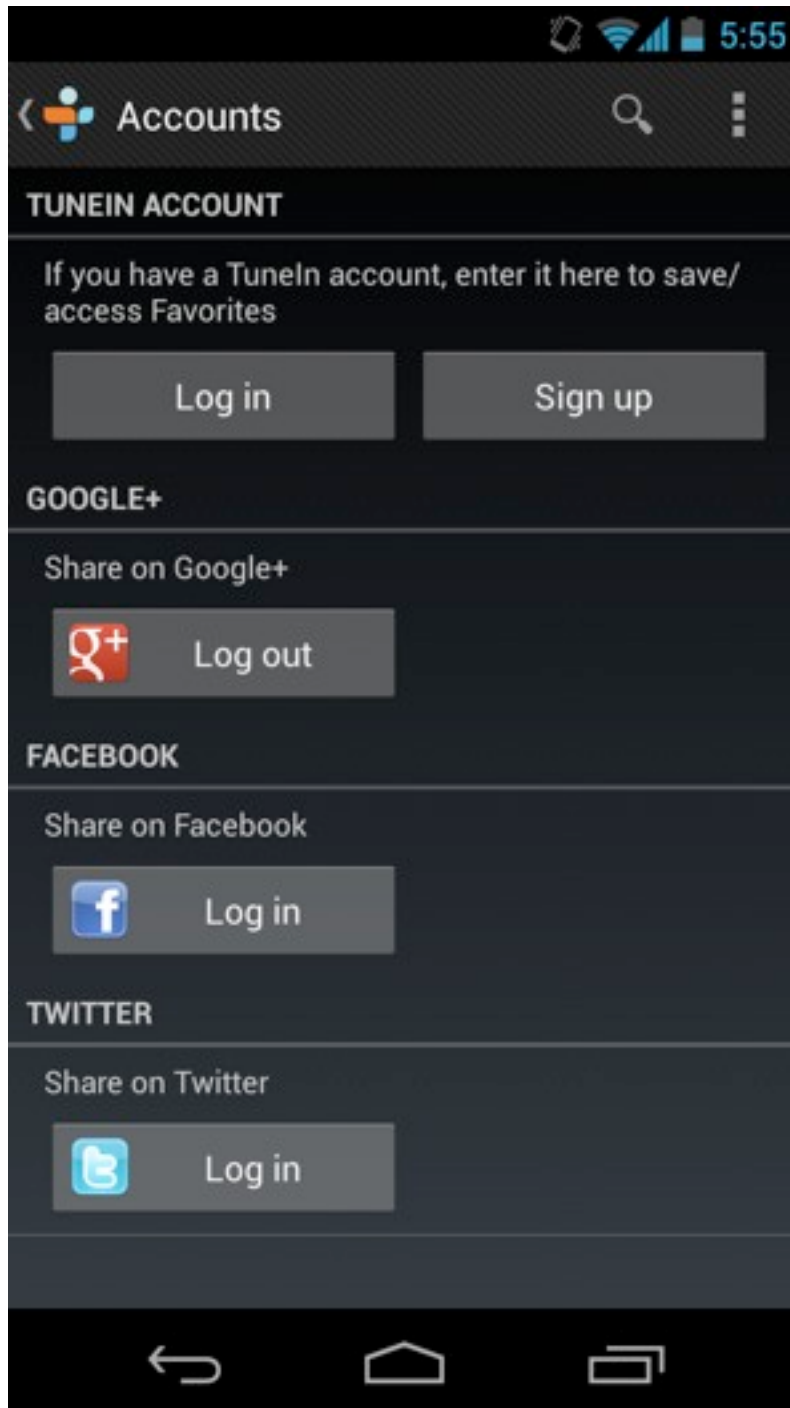




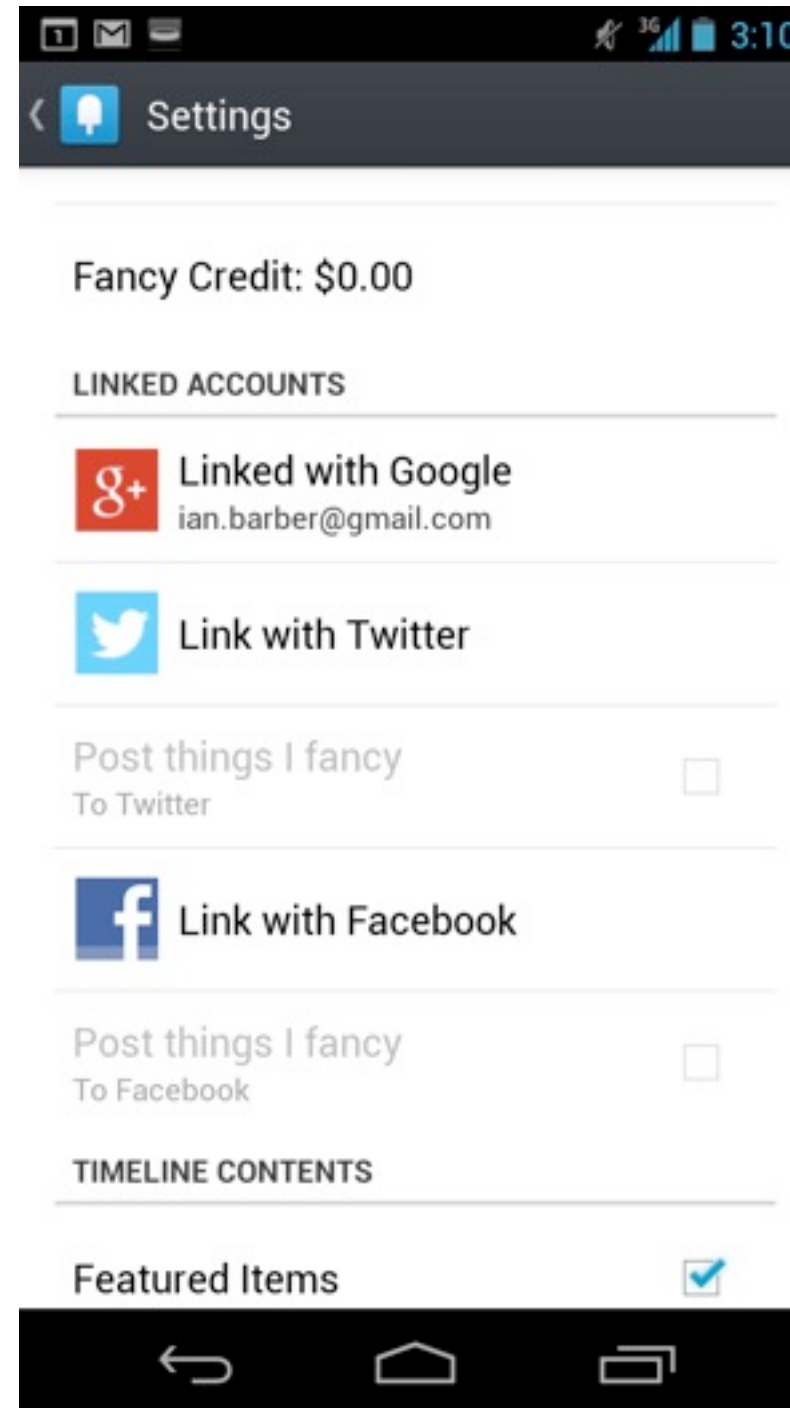




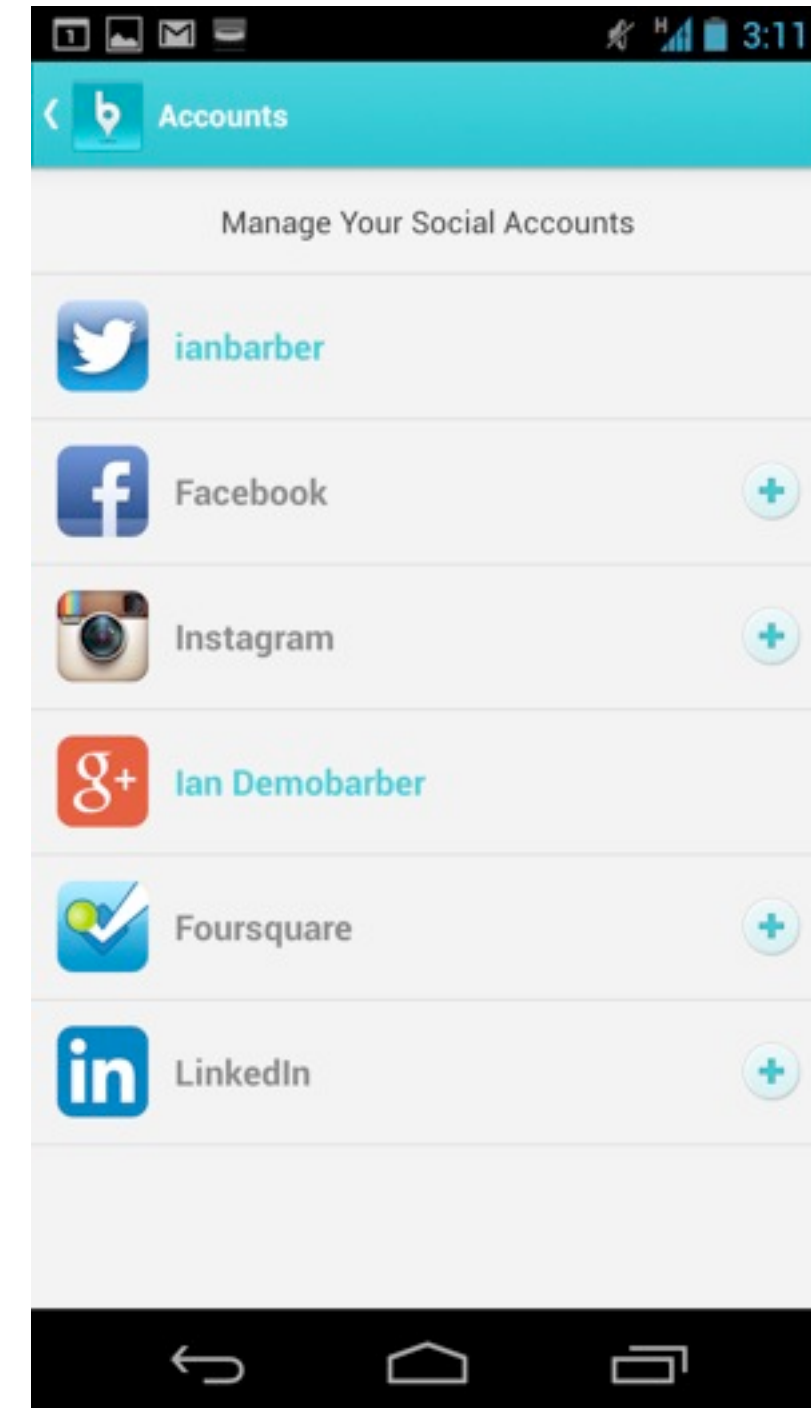
# Sharing Sessions



Tuneln Radio



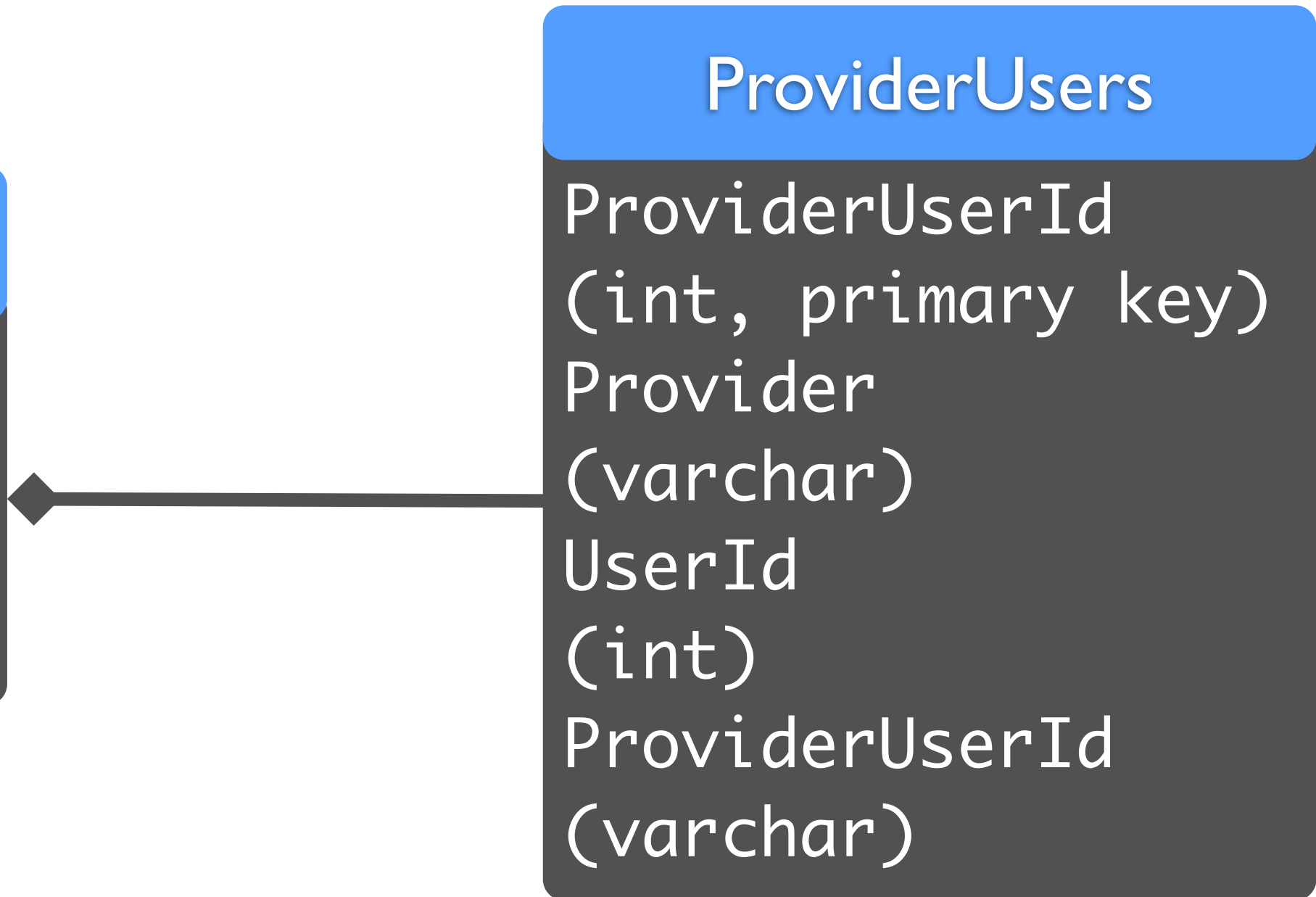
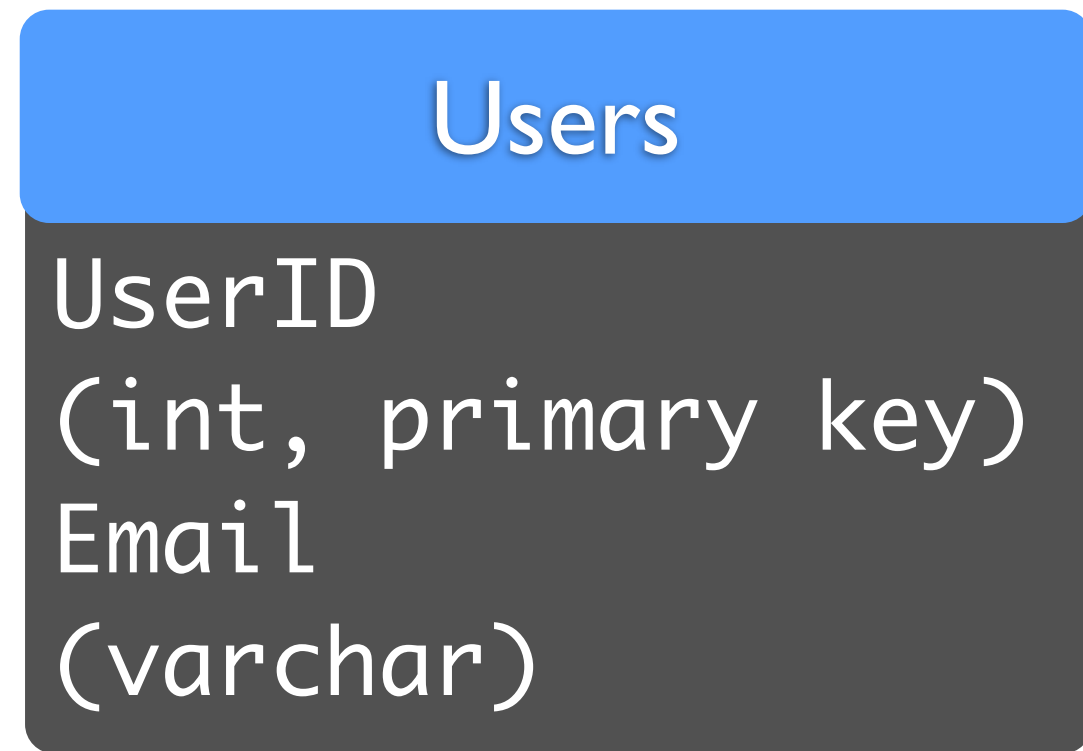
The Fancy



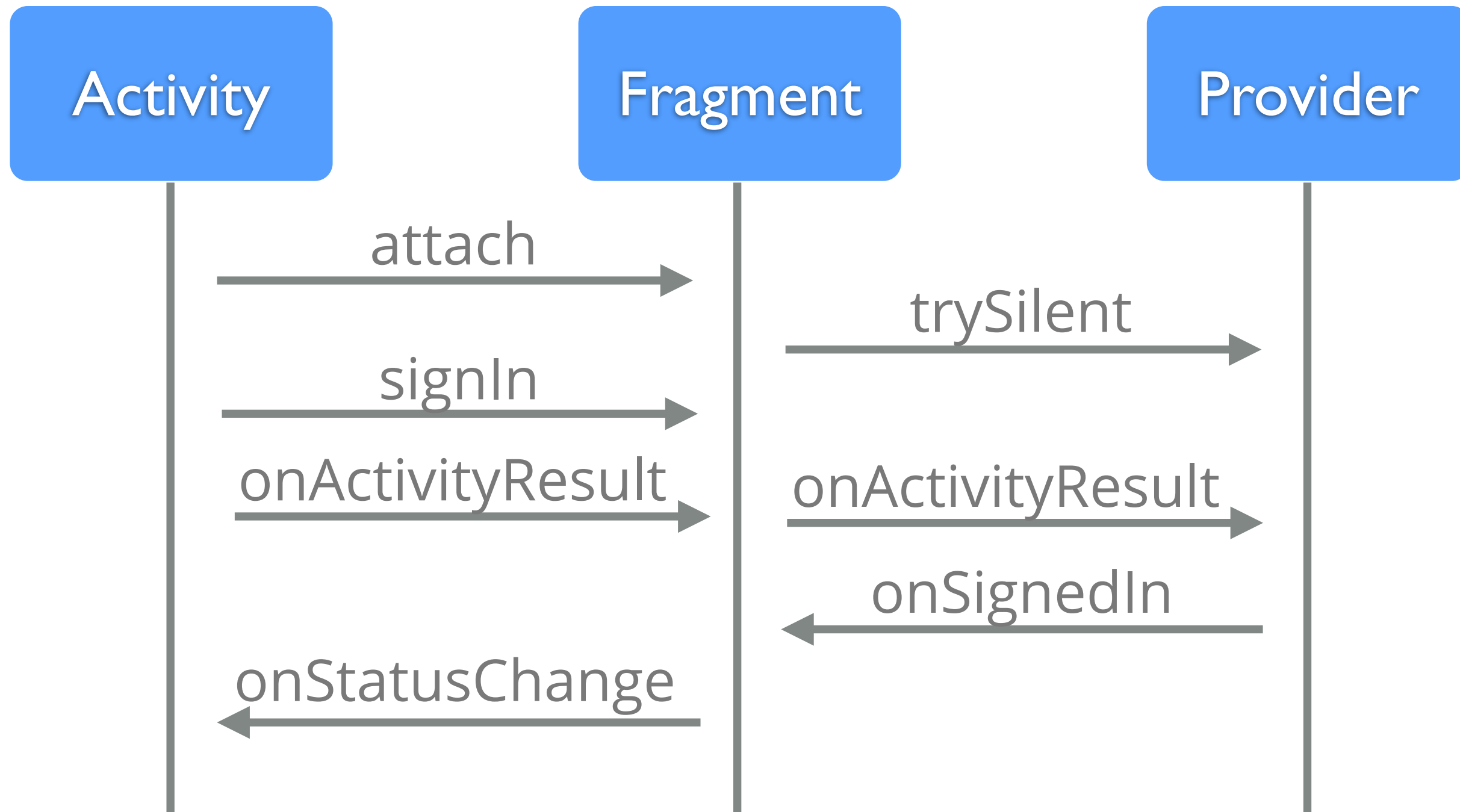
Banjo



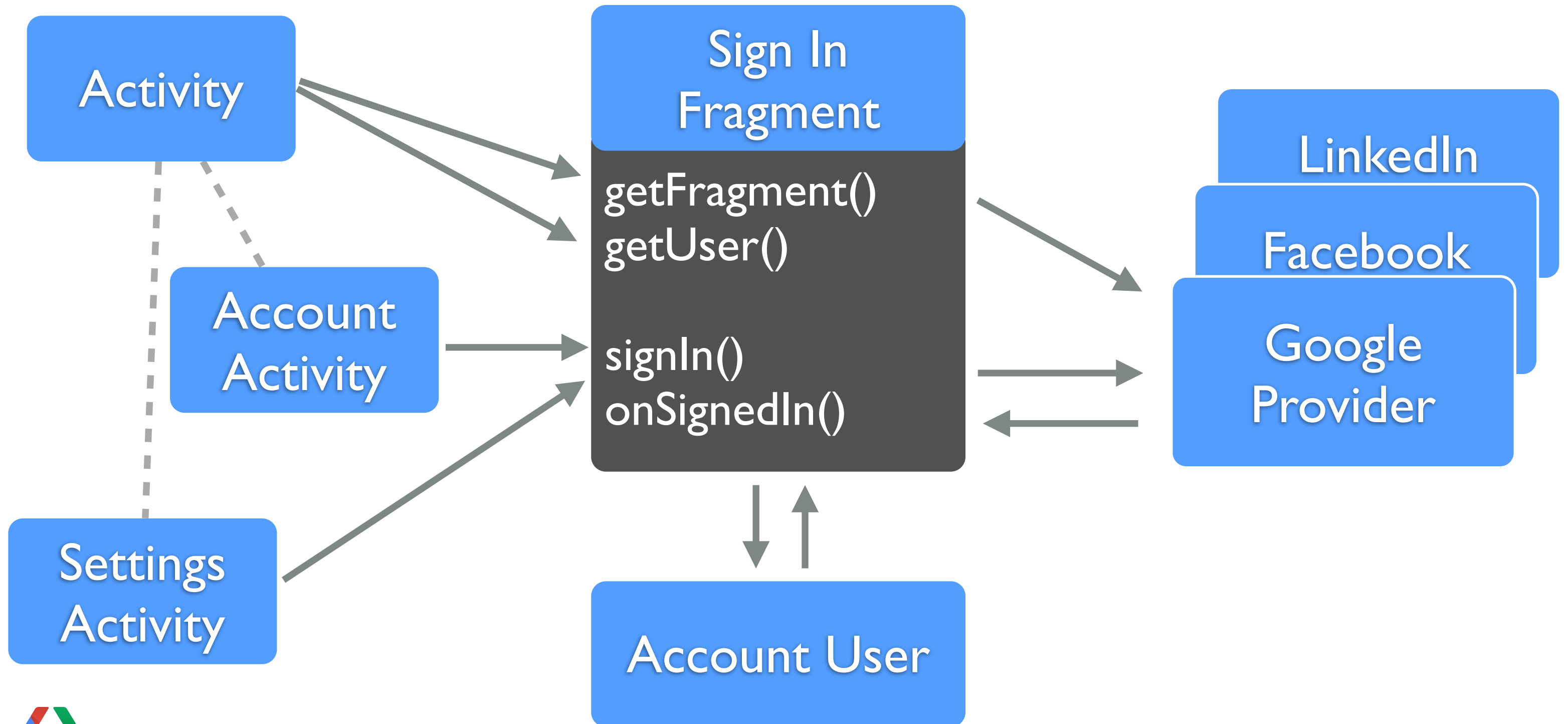
# Combining Accounts



# Authentication on Android



# Mediating Access



# Main Activity Fragment Callback

```
public void onStatusChange(SignInUser user) {  
    maintext.setText(user.getName());  
    graph.setChecked(user.hasFeature(  
        ProviderUtil.Feature.GRAPH));  
    if(user.isNew()) {  
        // For new users, give them an extra hello!  
        Toast.makeText(this,  
            getString(R.string.firstrun),  
            Toast.LENGTH_LONG).show();  
    }  
}
```

Java





# User Object

Java

```
public class SignInUser {  
  
    private HashMap<Provider, Object> mProviderData;  
    private String mName;  
    private Integer mId;  
    private boolean mIsNew;  
    private DBAdapter mDb;  
}
```



# User Object - Adding Provider Data

Java

```
public void setProviderData(
    Provider idp, Object user) {
    mProviderData.put(idp, user);
    if (mId == null) {
        mId = mDb.getUserId(idp, getProviderUserId(idp));
        if (mId == null) {
            mDb.createUser(idp, this);
            mIsNew = true;
        }
    }
}
```



# User Object - Testing For Features

Java

```
public boolean hasFeature(  
    ProviderUtil.Feature feature) {  
    for(Provider p : mProviderData.keySet()) {  
        if (p.hasFeature(feature)) {  
            return true;  
        }  
    }  
    return false;  
}
```



# Strategy - Facebook

Java

```
public void trySilentAuthentication() {  
    Session.openActiveSession(mManager.getActivity(),  
        false, this);  
}  
  
public void signIn() {  
    Session.openActiveSession(mManager.getActivity(),  
        true, this);  
}
```



# Strategy - Google+

Java

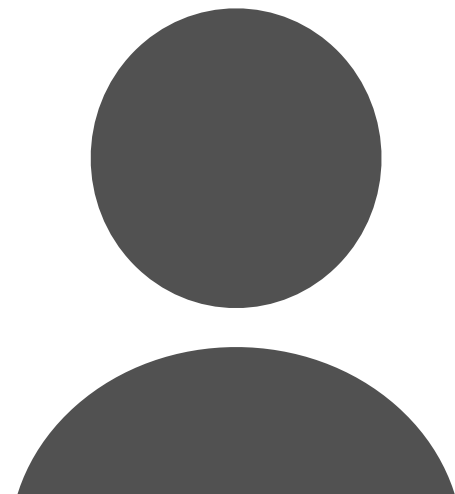
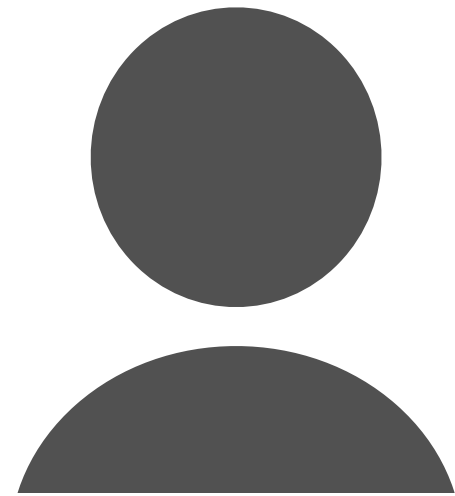
```
public void trySilentAuthentication() {  
    mPlusClient.connect();  
}
```

```
public void signIn() {  
    mRequestCode = GPLUS_REQUEST;  
    resolveLastResult();  
}
```

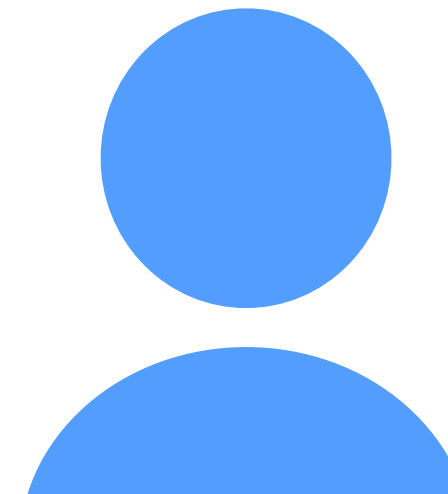
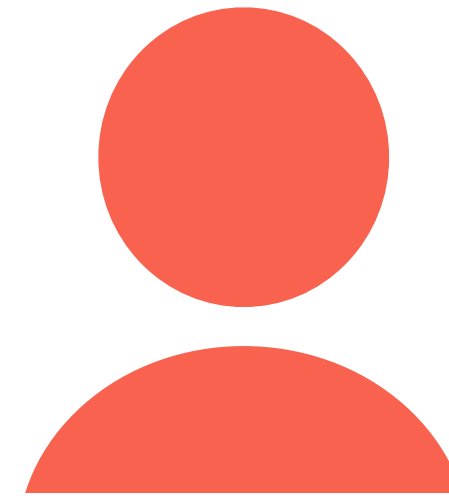


# Conflicting Accounts

App Account

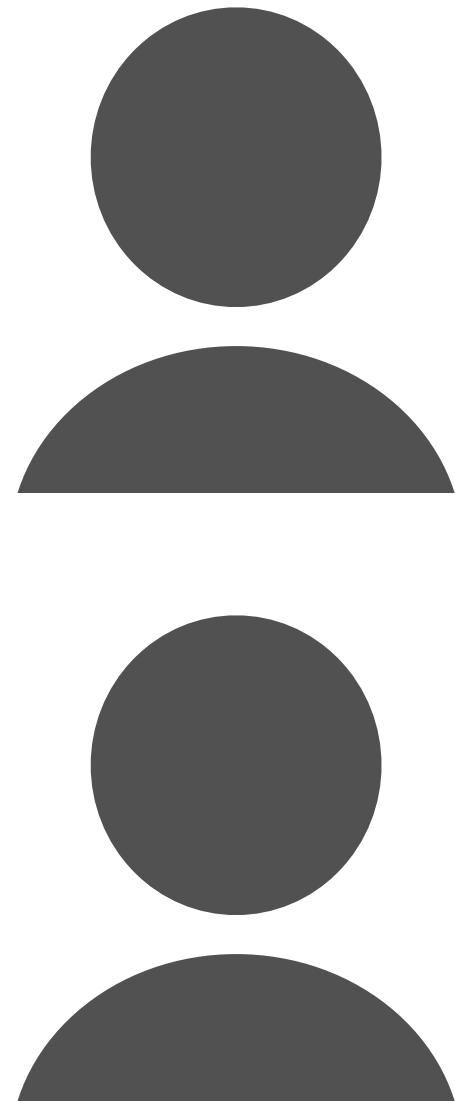


IDP Account

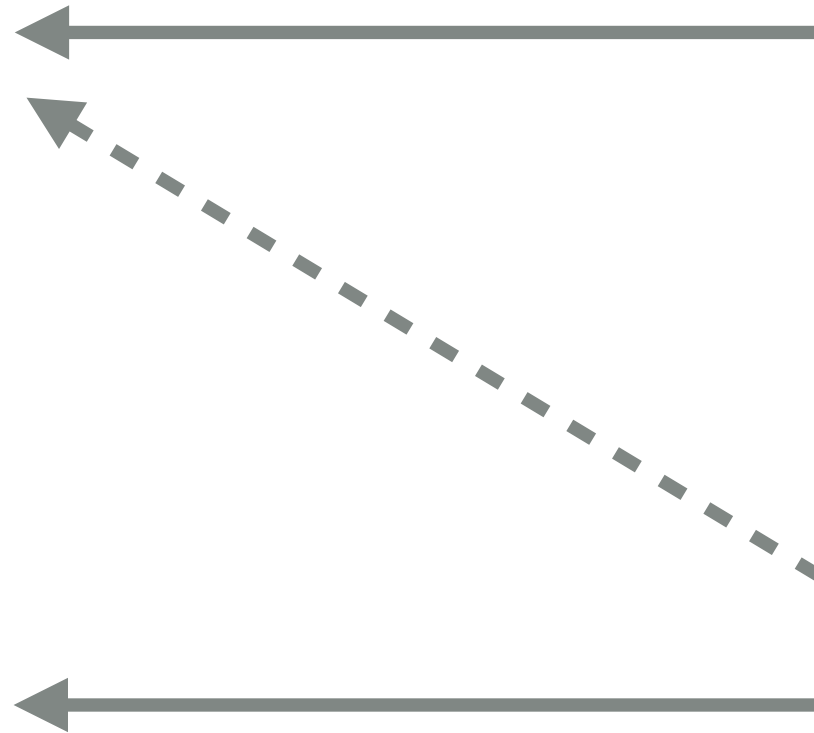
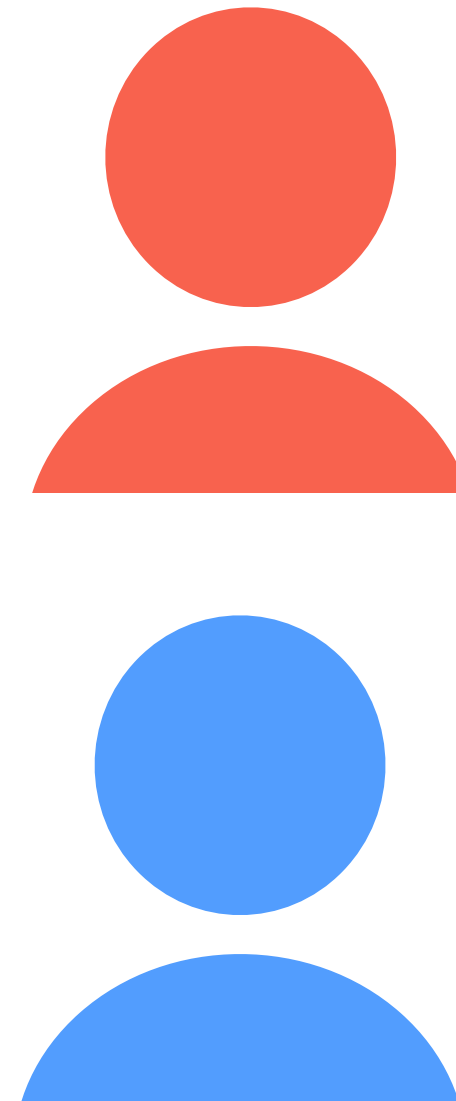


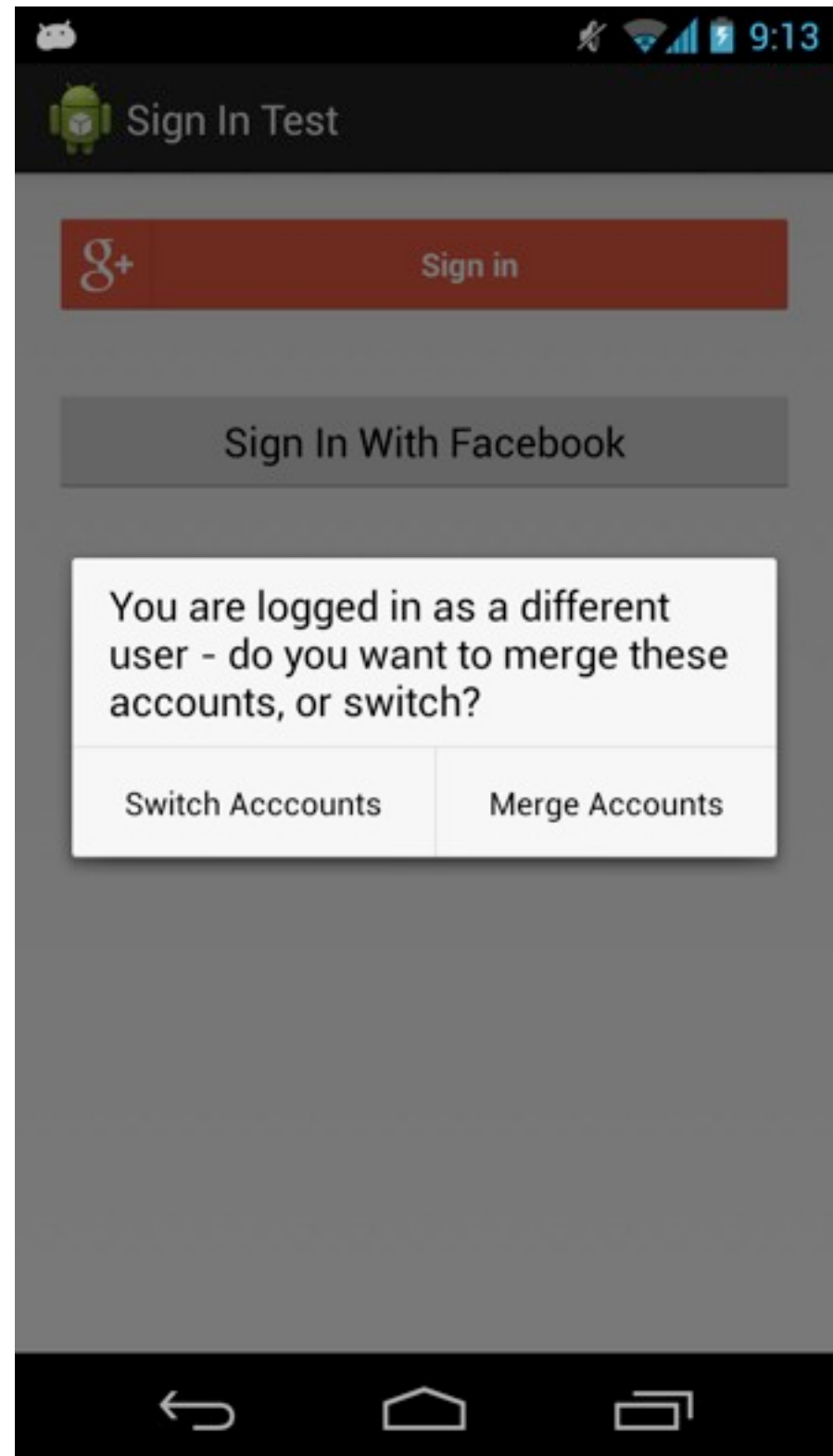
# Conflicting Accounts

App Account



IDP Account

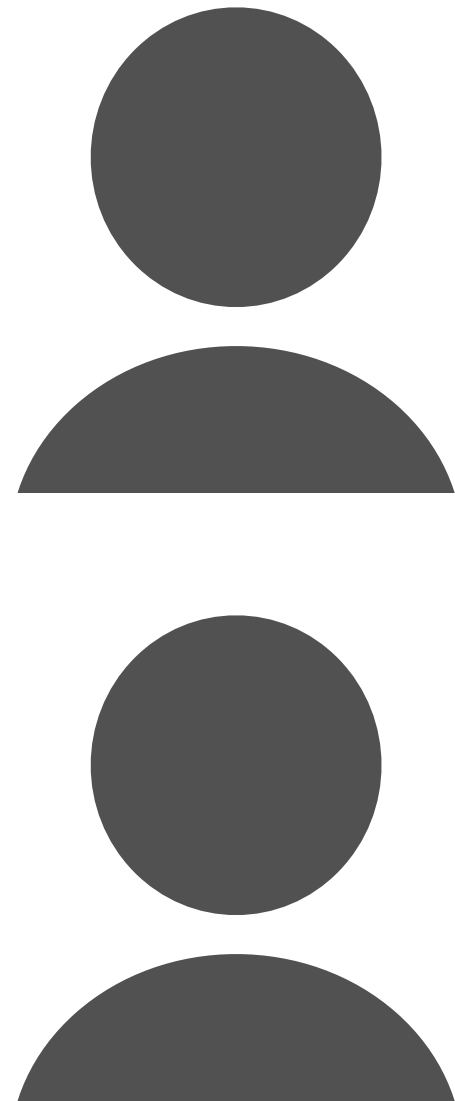




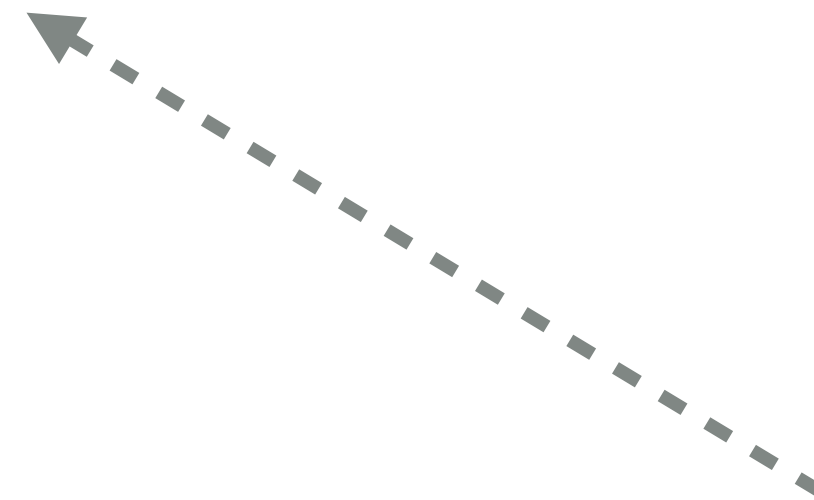
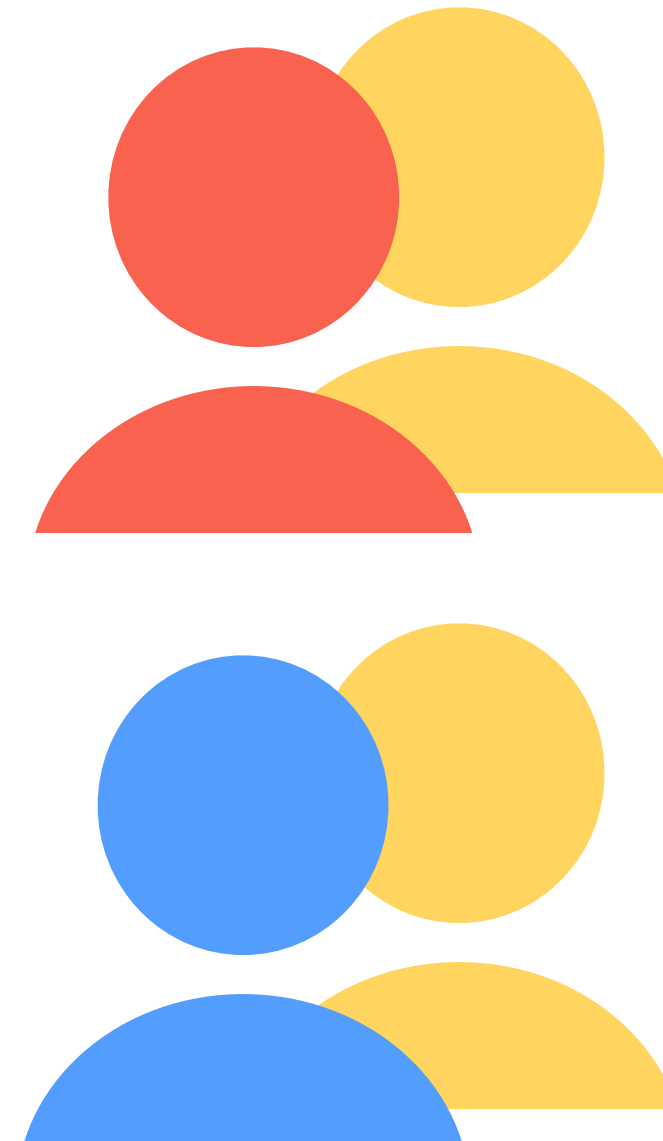


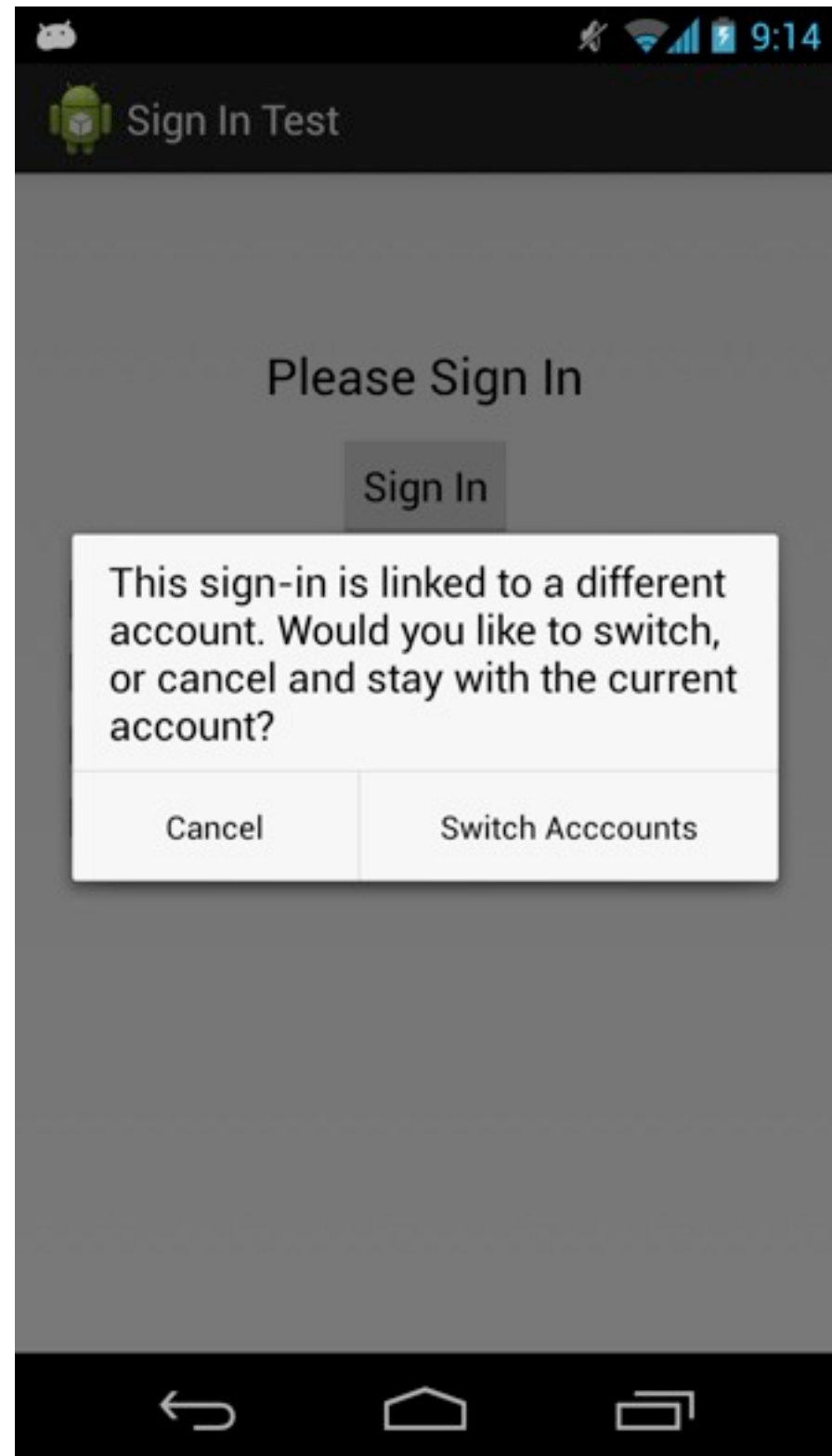
# Conflicting Accounts

App Account



IDP Account



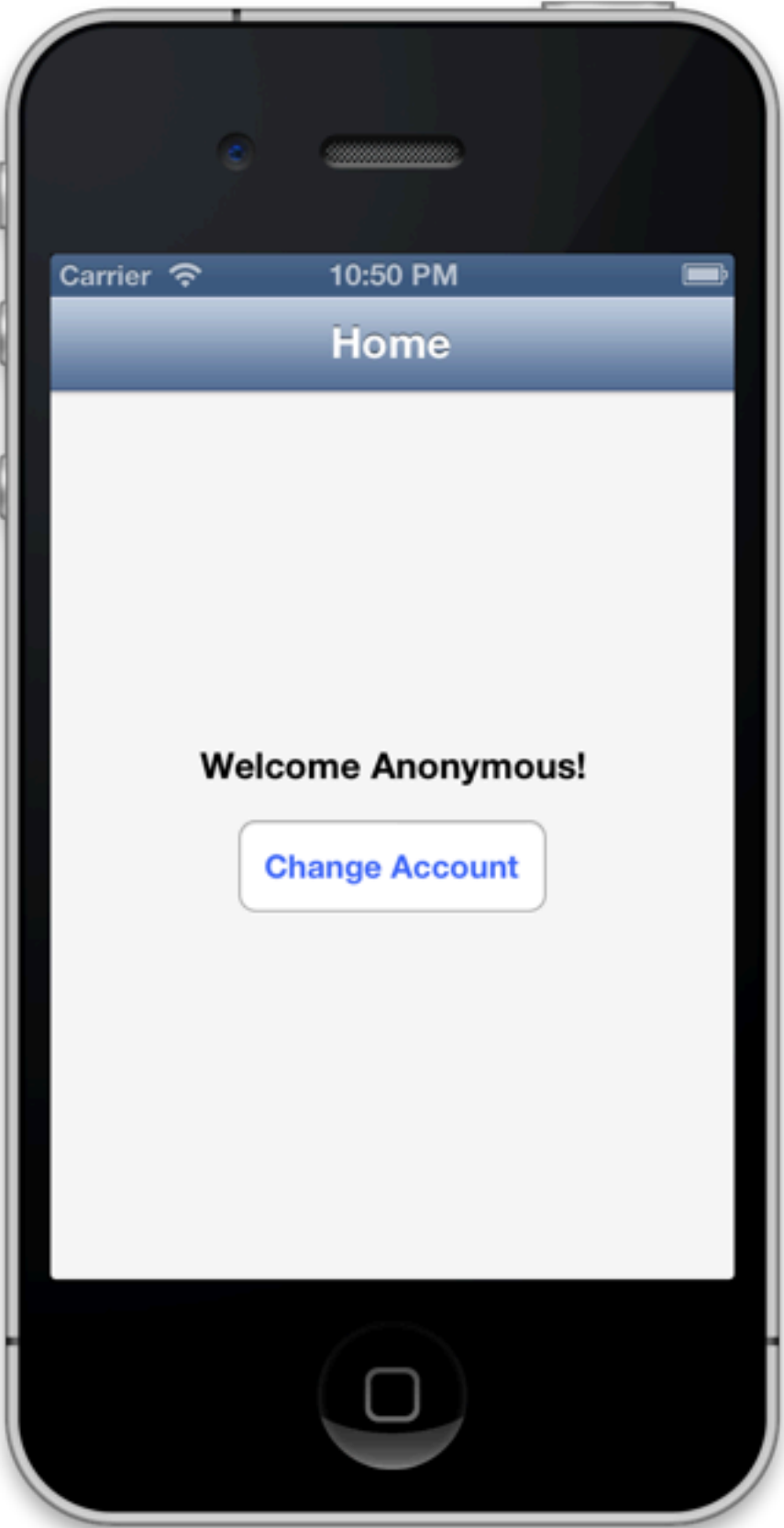


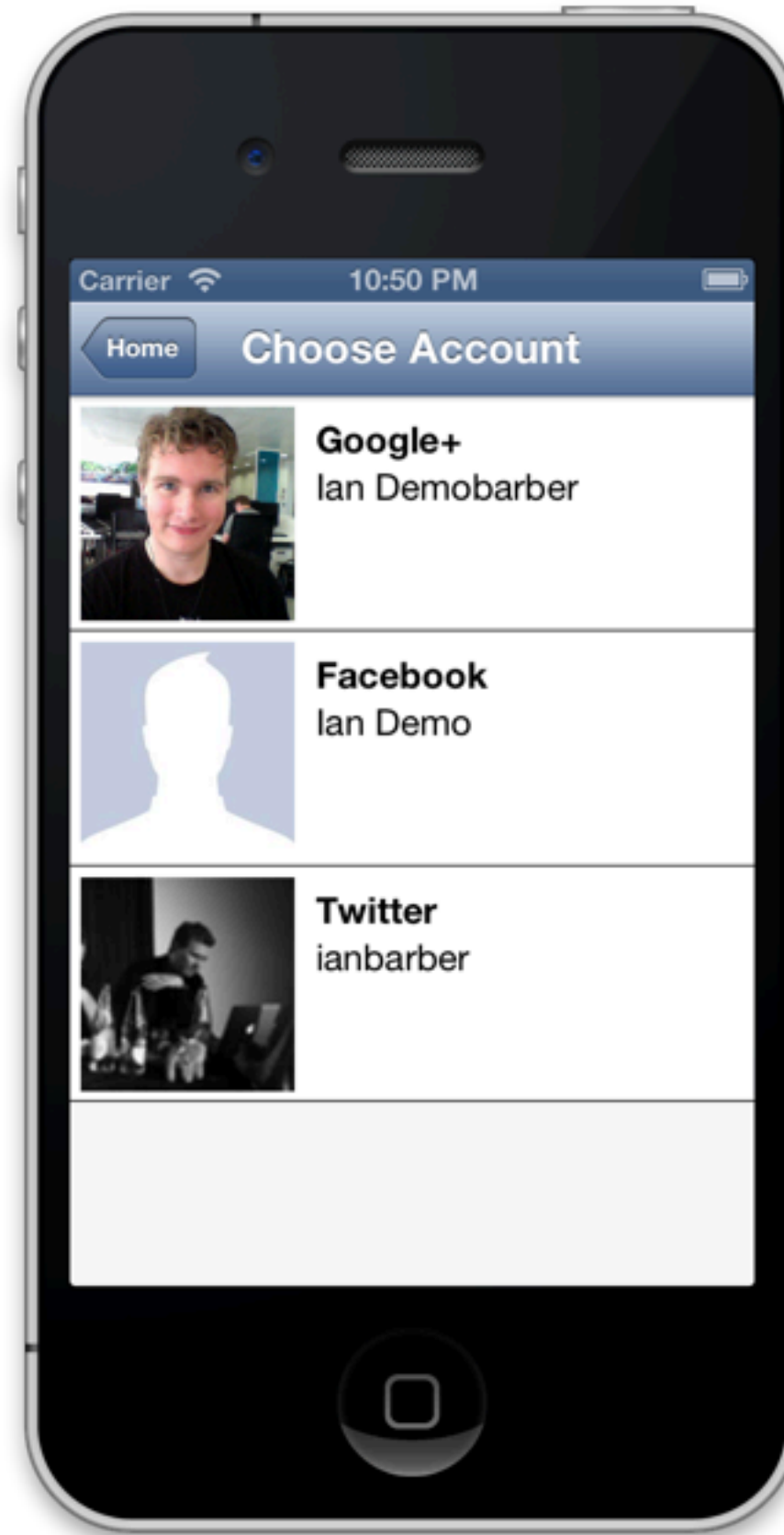
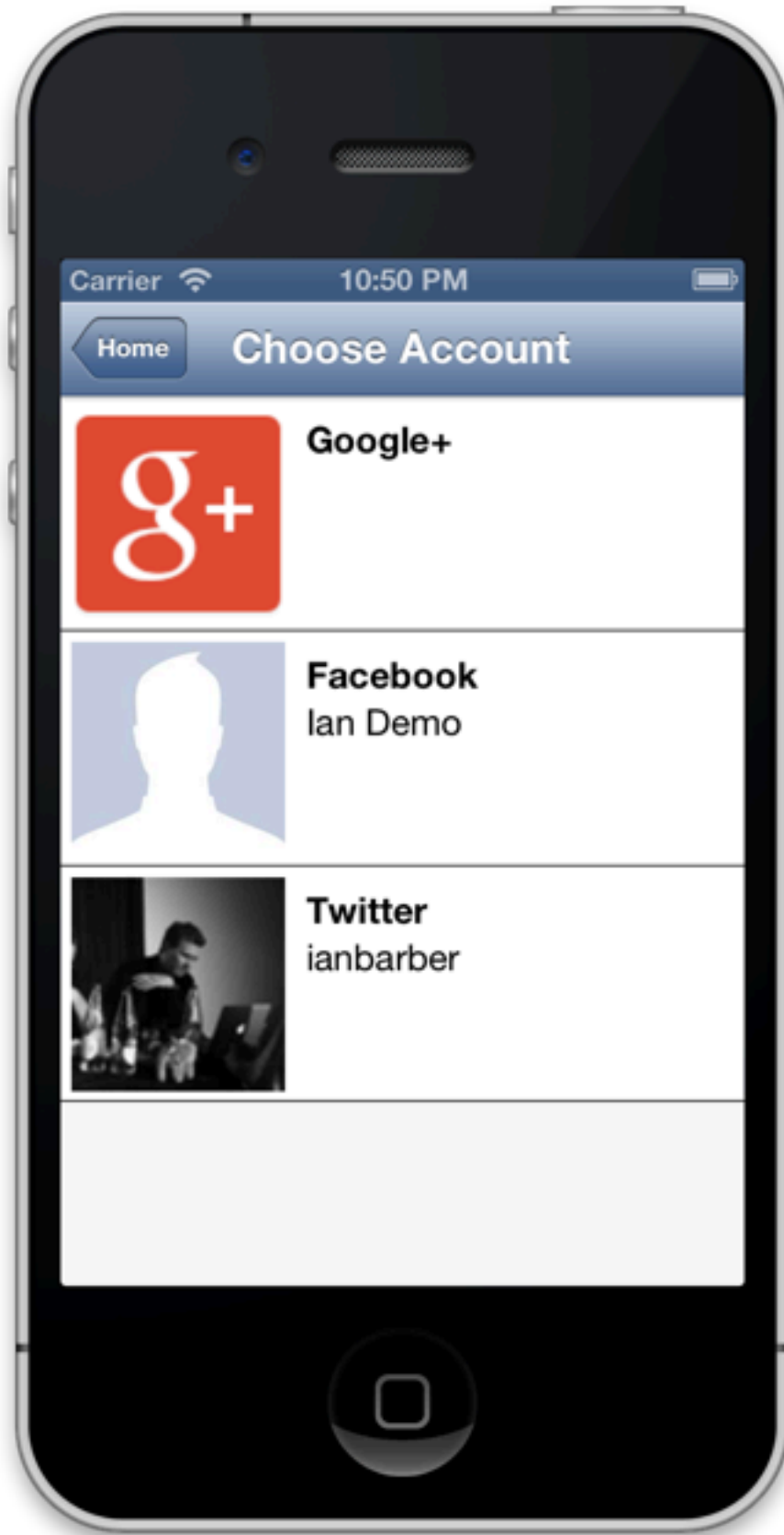
# Authenticator Mediates Merging

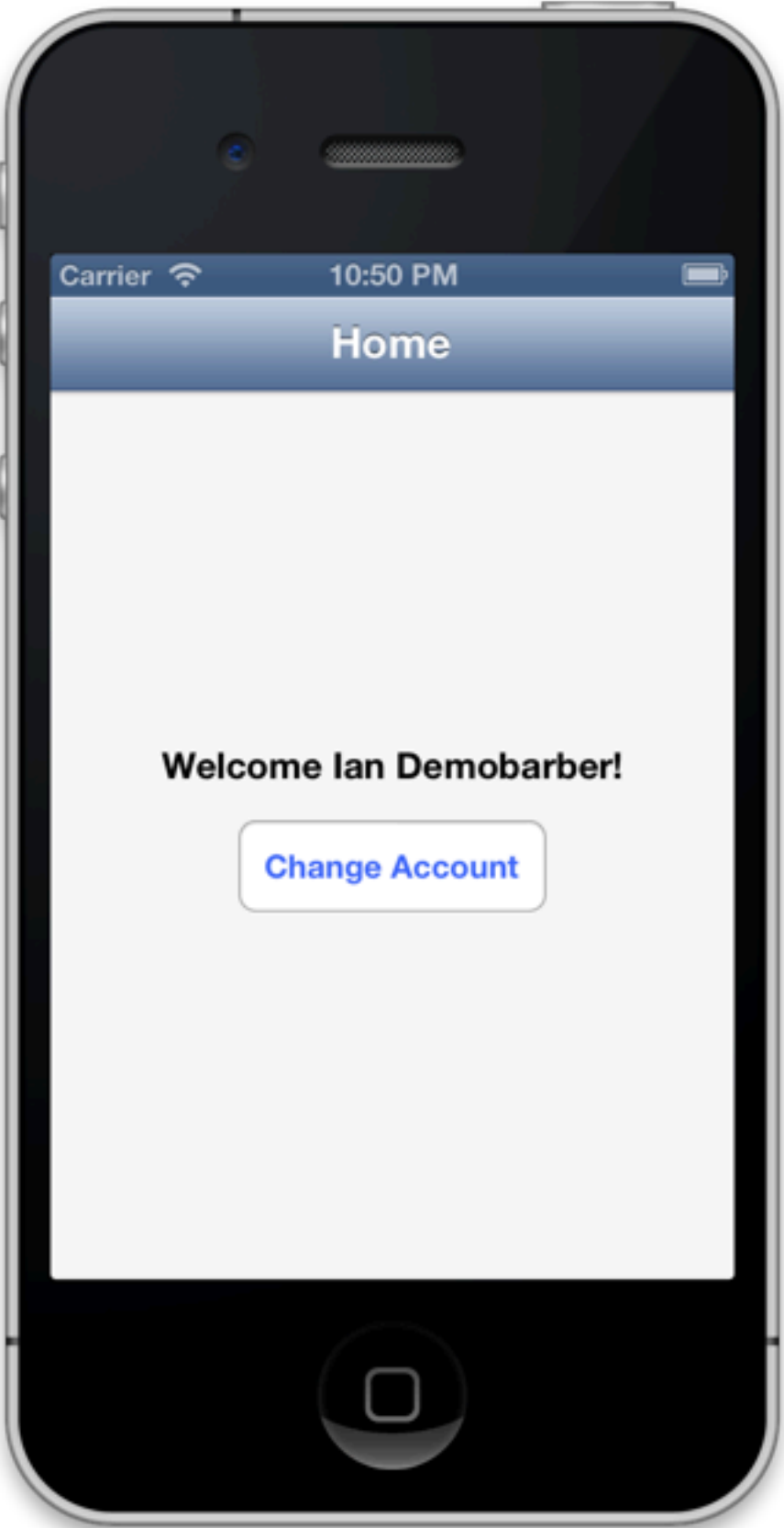
Java

```
public void onSignedIn(SignInUser user) {
    if (mUser == null) {
        mUser = user;
    } else if(mUser.getId() == user.getId()) {
        // ... merge provider data
    } else {
        if(mUser.canMerge(user)) {
            // ... show merge or switch
        } else {
            // ... how cancel or switch
        }
    }
}
```

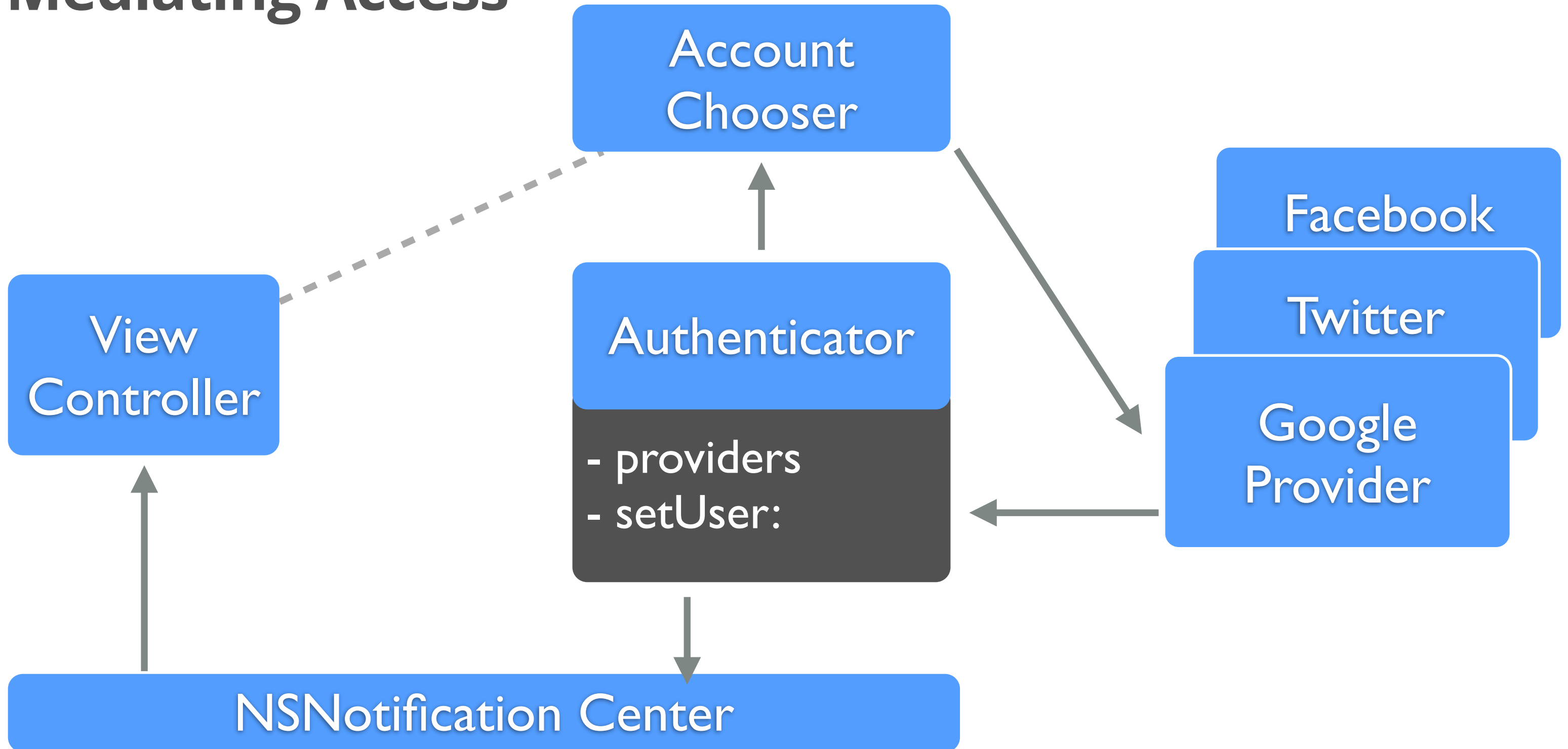








# Mediating Access



# ViewController Subscribes

Objective-C

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
    [[NSNotificationCenter defaultCenter]  
        addObserver:self  
        selector:@selector(checkUser:)  
        name:kUserStatusNotification  
        object:nil];  
    [self checkUser:nil];  
}
```





# Authenticator Publishes Message

Objective-C

```
@implementation Authenticator

// ...

- (void)setUser:(User *)user {
    _user = user;
    [[NSNotificationCenter defaultCenter]
     postNotificationName:kUserStatusNotification
     object:self];
}
```



# Individual Provider Holds IDP State

Objective-C

```
- (void)signIn {  
    if(self.user) {  
        return [self returnUser];  
    }  
    self.isUserAction = true;  
    [self retrieveUser];  
}
```



# Individual Provider Holds IDP State

Objective-C

```
- (void)retrieveUser {
    [self.accountStore
     requestAccessToAccountsWithType:self.accountType
                                options:nil
     completion:^(BOOL granted, NSError *error) {
        if(granted) {
            NSArray *accounts = [self.accountStore
                                accountsWithType:self.accountType];
            if ([accounts count] > 0) {
                self.user = [accounts objectAtIndex:0];
            }
        }
    }
}
```

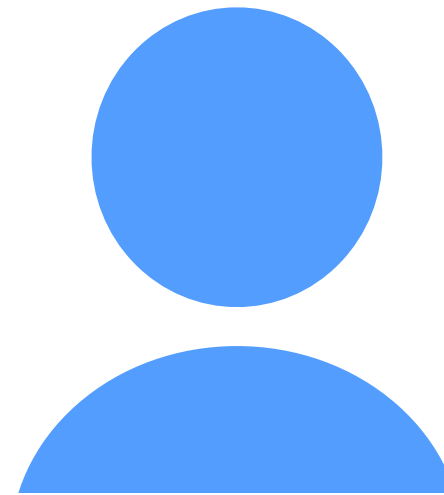


# Strategy - Google+

Objective-C

```
- (void)signIn {  
    if (self.user) {  
        return [self returnUser];  
    }  
    self.isUserAction = true;  
    [[GPPSignIn sharedInstance] authenticate];  
}
```





Authentication  
& Authorization

Data Model

User Story

Capabilities



# <Thank You!>



Ian Barber - <http://google.com/+IanBarber>

Running a login system with an account chooser - <http://goo.gl/R8pMP>

Google+ Developer Site - <http://developers.google.com/+>

Code from this talk - <https://github.com/googleplus/googleIO-811>



Google  
Developers