

# Your App, Your Website, and Safari

Ensuring continuity of user experience

Session 506

Ricky Mondello

Safari and WebKit Engineer

# Continuity

**Continuity** |ˌkæntnɪˈtɪoʊəti|

noun (pl. continuities)

the unbroken and consistent existence or operation of something over a period of time

**Continuity** |ˌkæntɪn(ə)ˈwɪtē|

noun (pl. continuities)

the **unbroken** and **consistent** existence or operation of something over a period of time

# Handoff



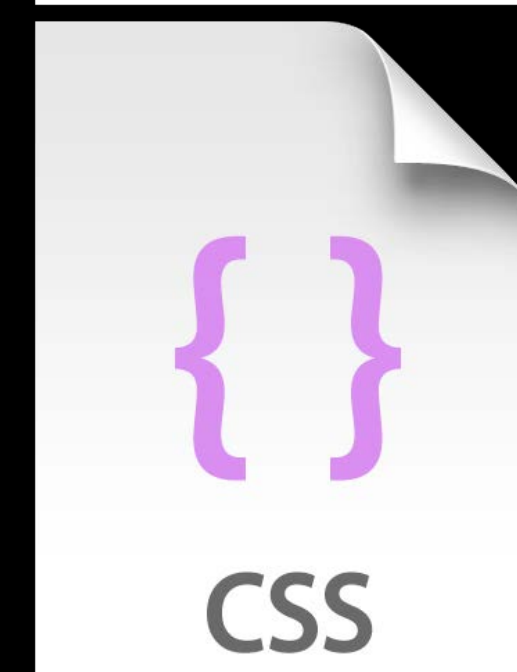
# Related Sessions

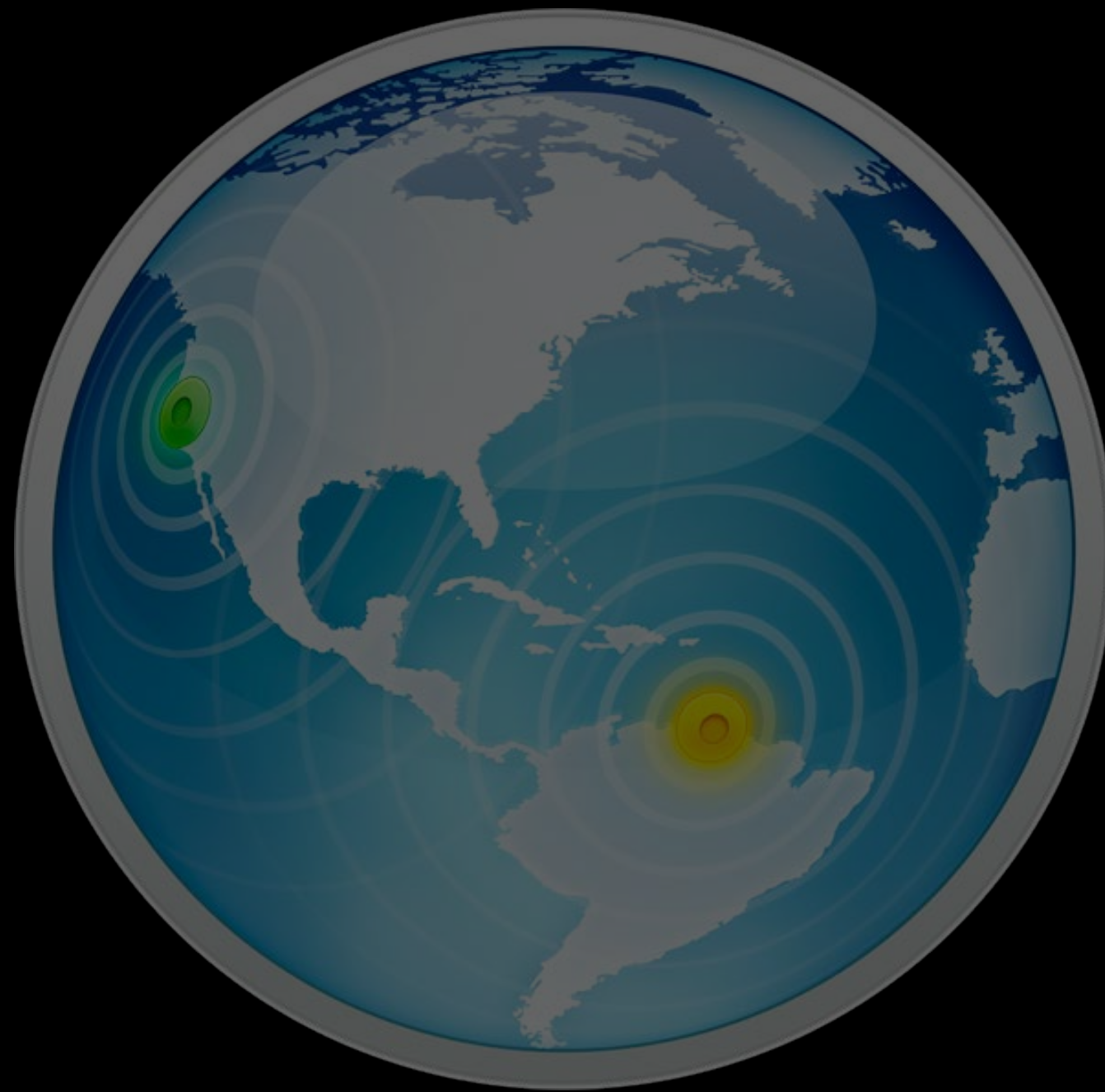
- 
- Adopting Handoff on iOS and OS X Mission Wednesday 2:00PM
-





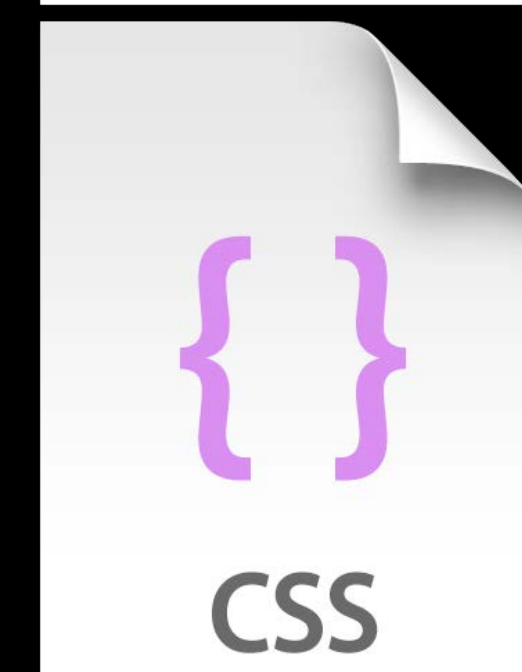












# What You Will Learn

How to:

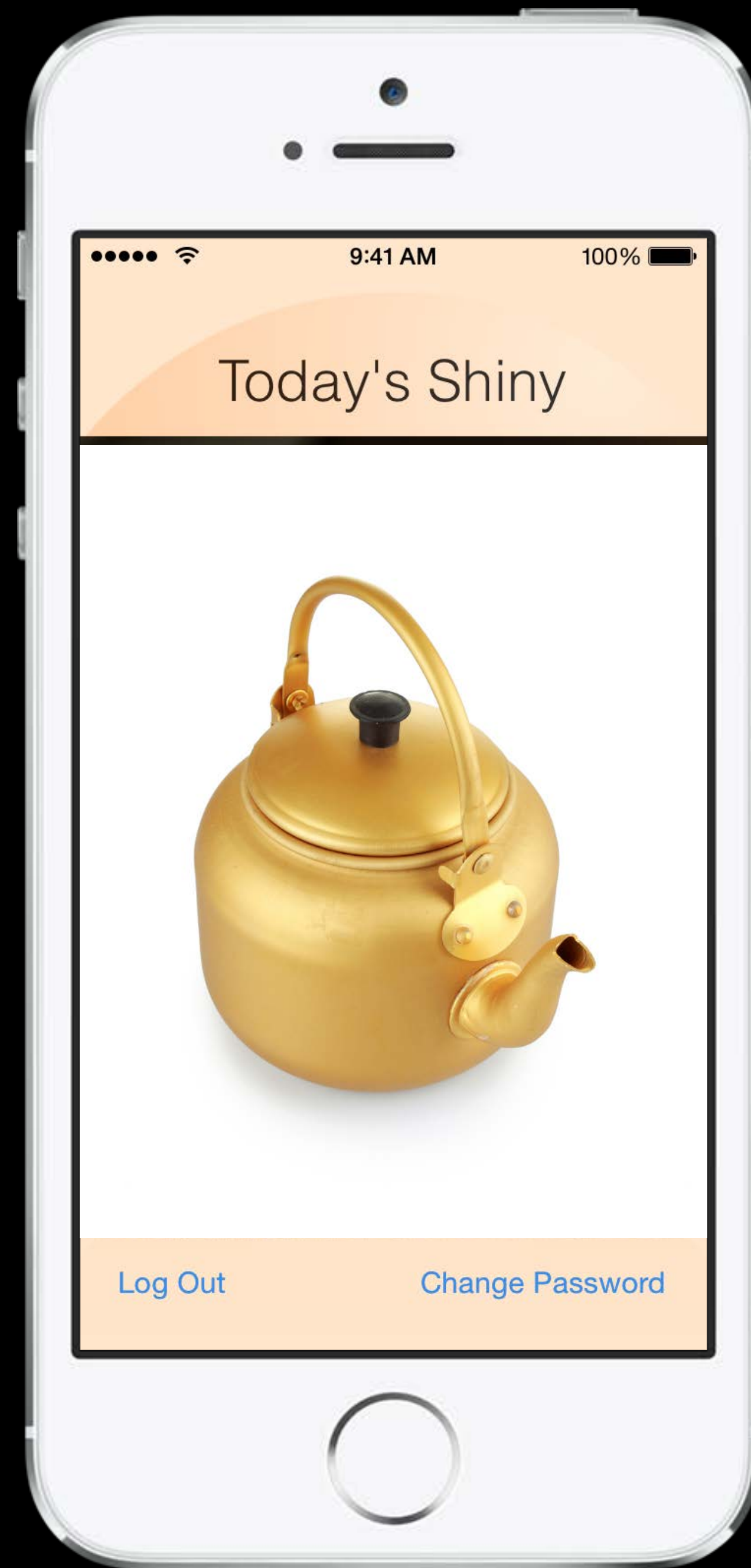
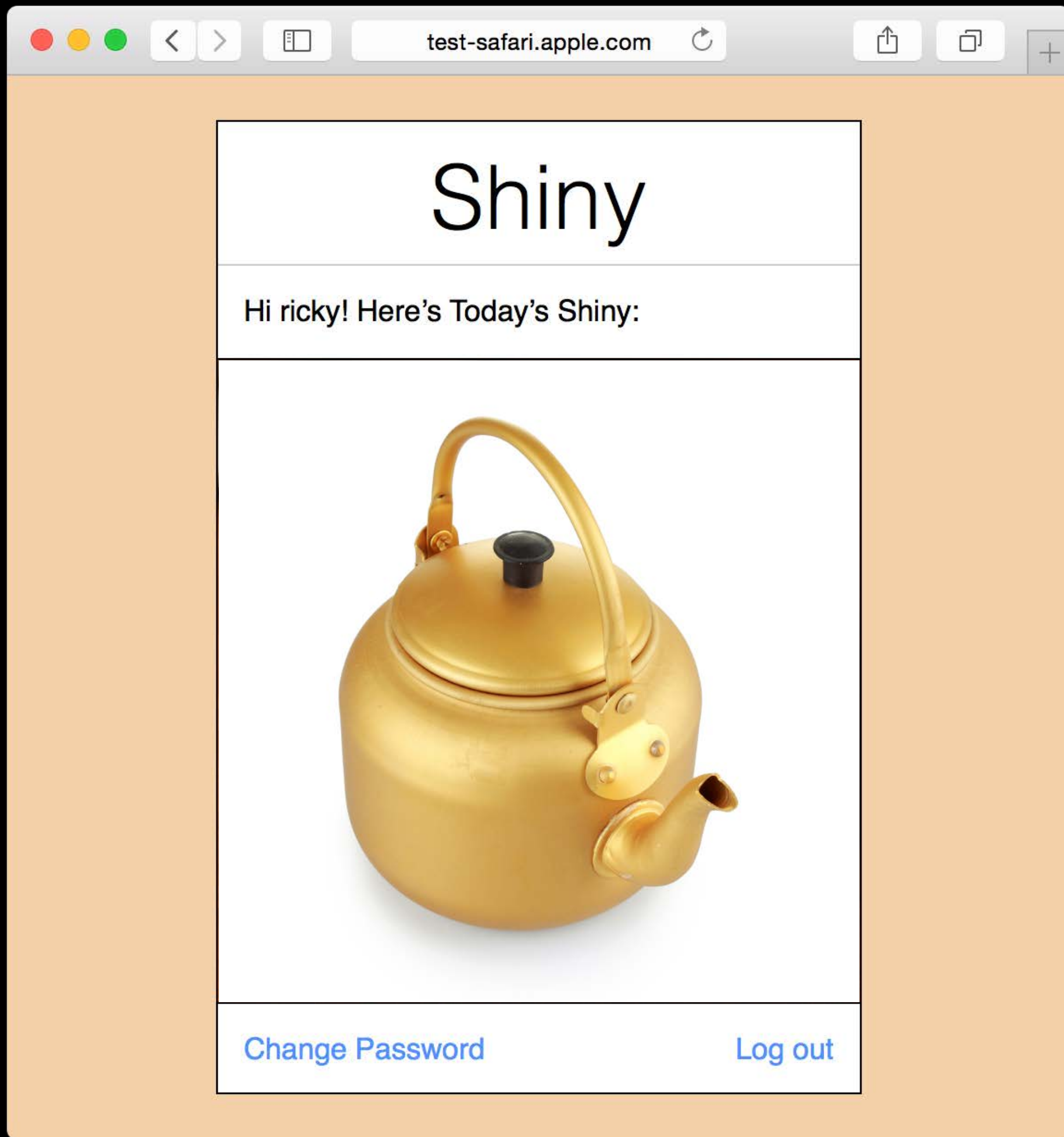
- Use Safari credentials in your iOS apps
- Make Safari AutoFill work better on your website
- Help Safari promote your website
- Make your website consistent across devices

# Credentials in Safari

And how your app can use them









# Shiny

Username

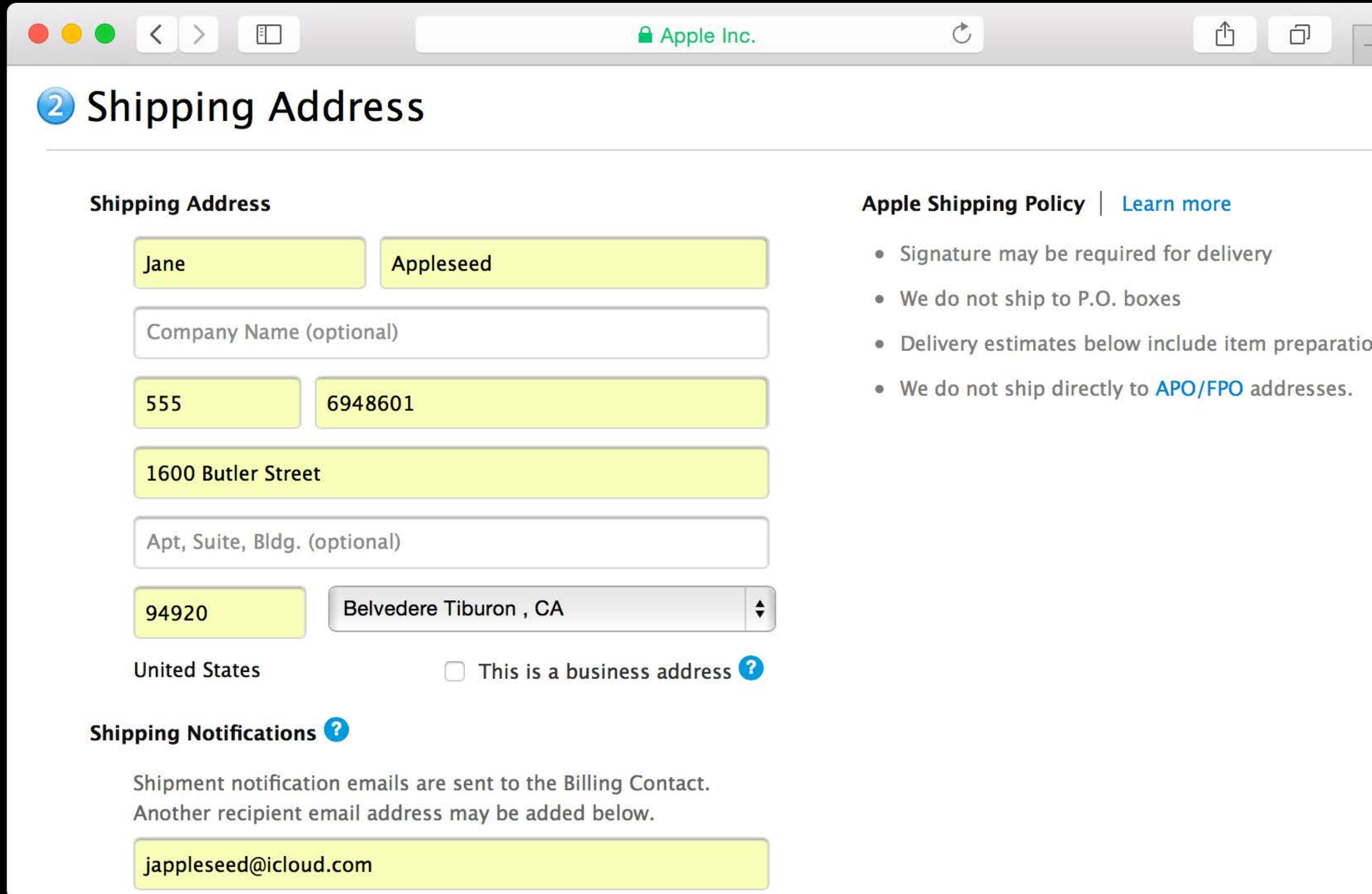
Password

[Log In](#)

[Forgot Password?](#)



# Safari AutoFill



The screenshot shows a Safari browser window with the address bar displaying "Apple Inc." and a lock icon. The page title is "2 Shipping Address". The form contains several input fields with yellow AutoFill suggestions:

- First name: Jane
- Last name: Appleseed
- Company Name (optional):
- Zip code: 555
- Postal code: 6948601
- Street address: 1600 Butler Street
- Apt, Suite, Bldg. (optional):
- City/State: 94920, Belvedere Tiburon, CA
- Country: United States
- Business address checkbox:  This is a business address ?

**Shipping Notifications ?**

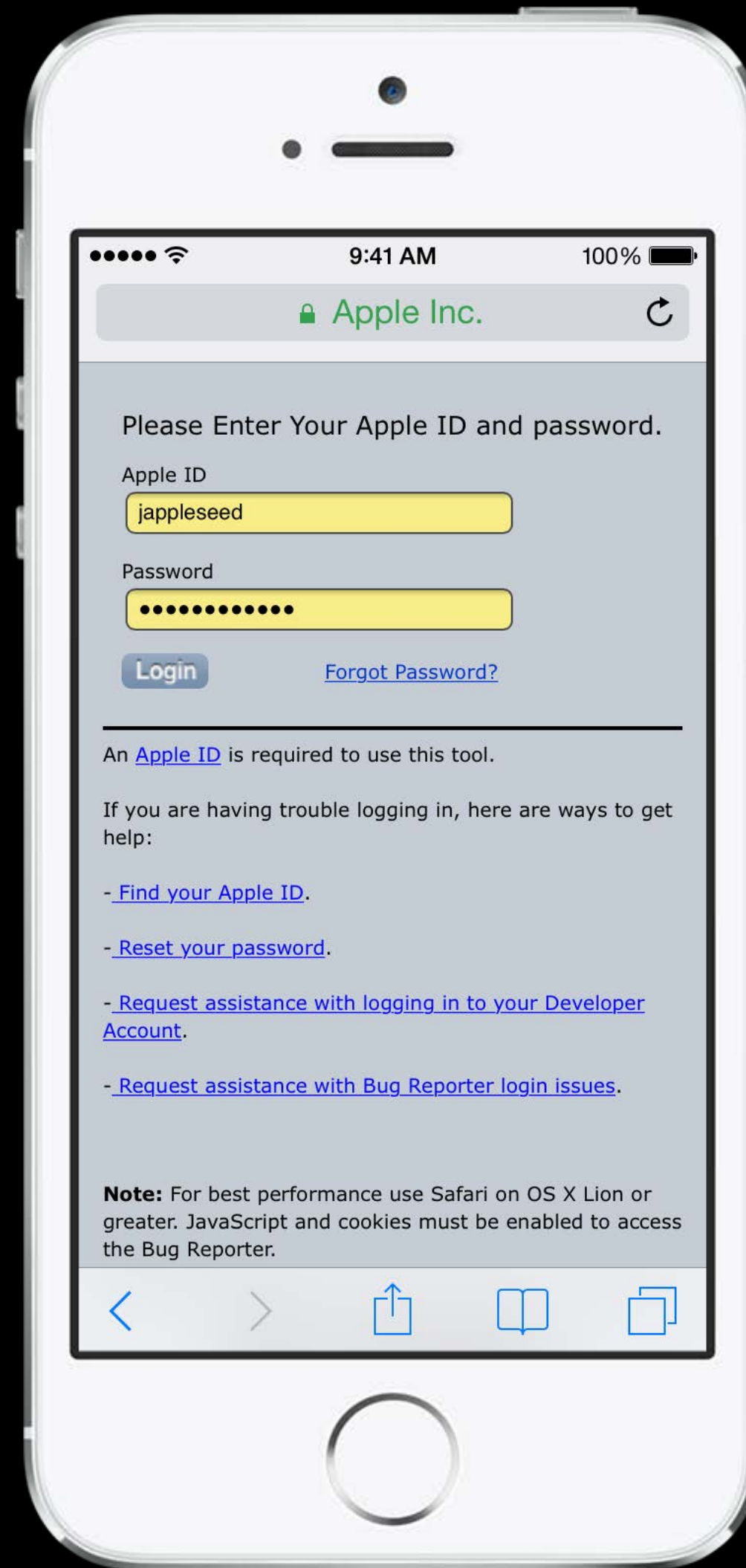
Shipment notification emails are sent to the Billing Contact. Another recipient email address may be added below.

jappleseed@icloud.com

**Apple Shipping Policy | [Learn more](#)**

- Signature may be required for delivery
- We do not ship to P.O. boxes
- Delivery estimates below include item preparation
- We do not ship directly to [APO/FPO](#) addresses.

# Safari AutoFill



# Safari AutoFill

The screenshot shows a Safari browser window with the address bar displaying 'bugs.webkit.org'. The page title is 'WebKit Bugzilla' and the main heading is 'Create a new user account for 'jappleseed@icloud.com''. The page contains a form for account creation with the following fields:

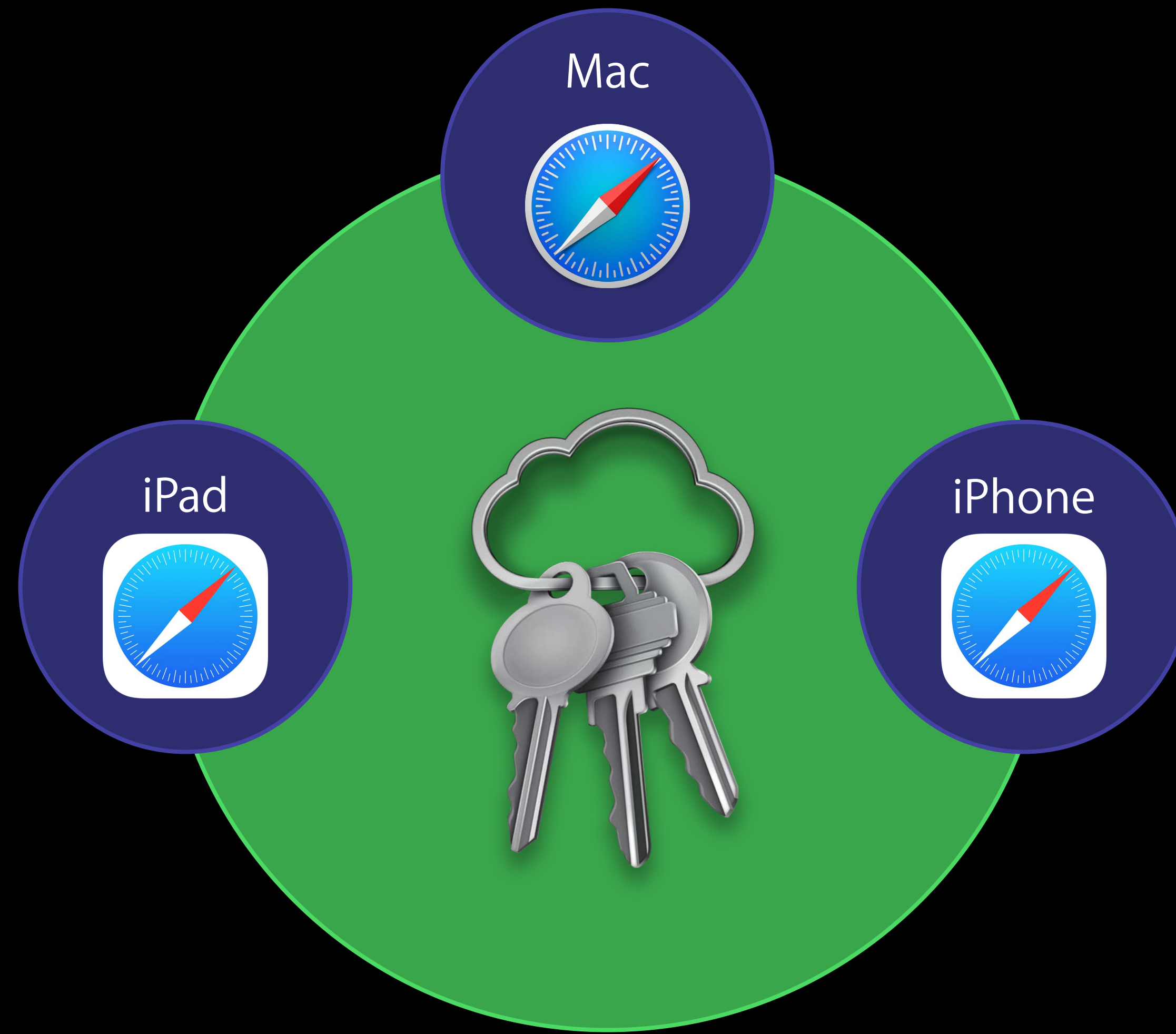
- Email Address:** jappleseed@icloud.com
- (OPTIONAL) Real Name:** Jane Appleseed
- Type your password:** [Empty password field]
- Confirm your password:** [Empty password field]

A Safari AutoFill suggestion popup is displayed over the password fields, showing a key icon and the text: 'Use Safari suggested password: KpN-HM6-SFg-y6P'. Below the popup, a message states: 'This password will be saved in your iCloud Keychain so it is available for AutoFill on all your devices.' The page also includes a 'Cancel Account' button and a footer with navigation links: Home | New | Search | [Search Box] | Find | Reports | Requests | New Account | Log In.

# Safari AutoFill

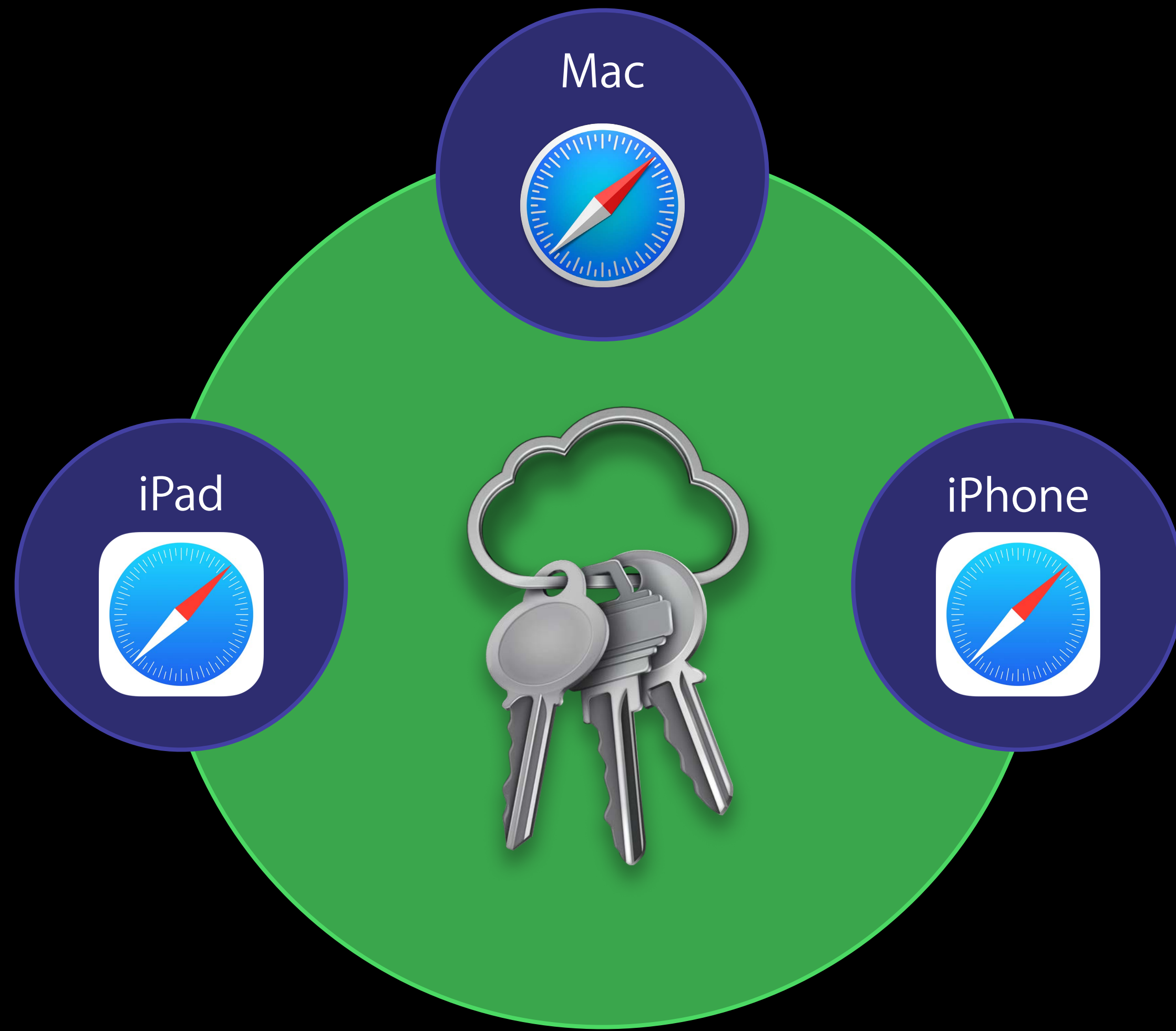


# Credentials in Safari

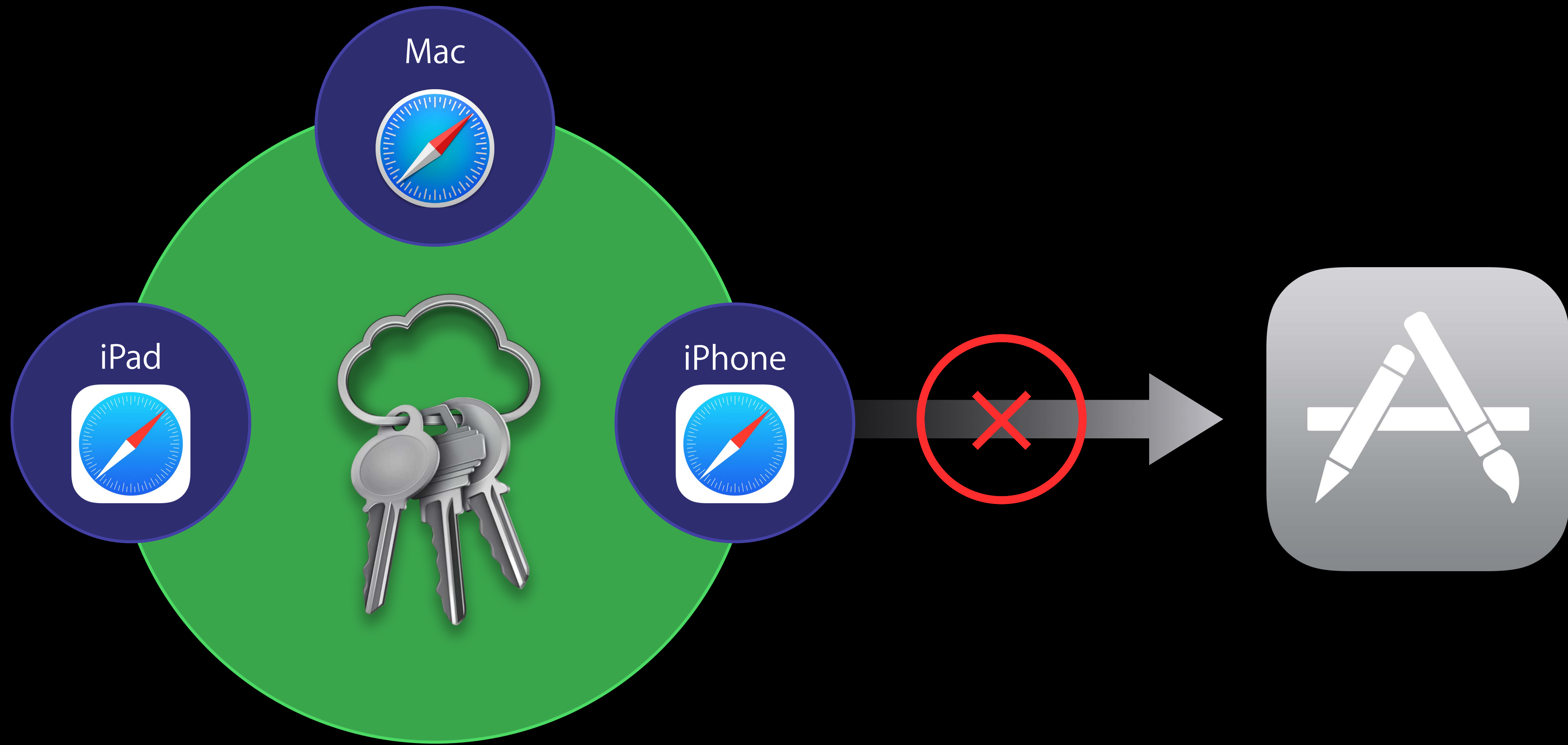




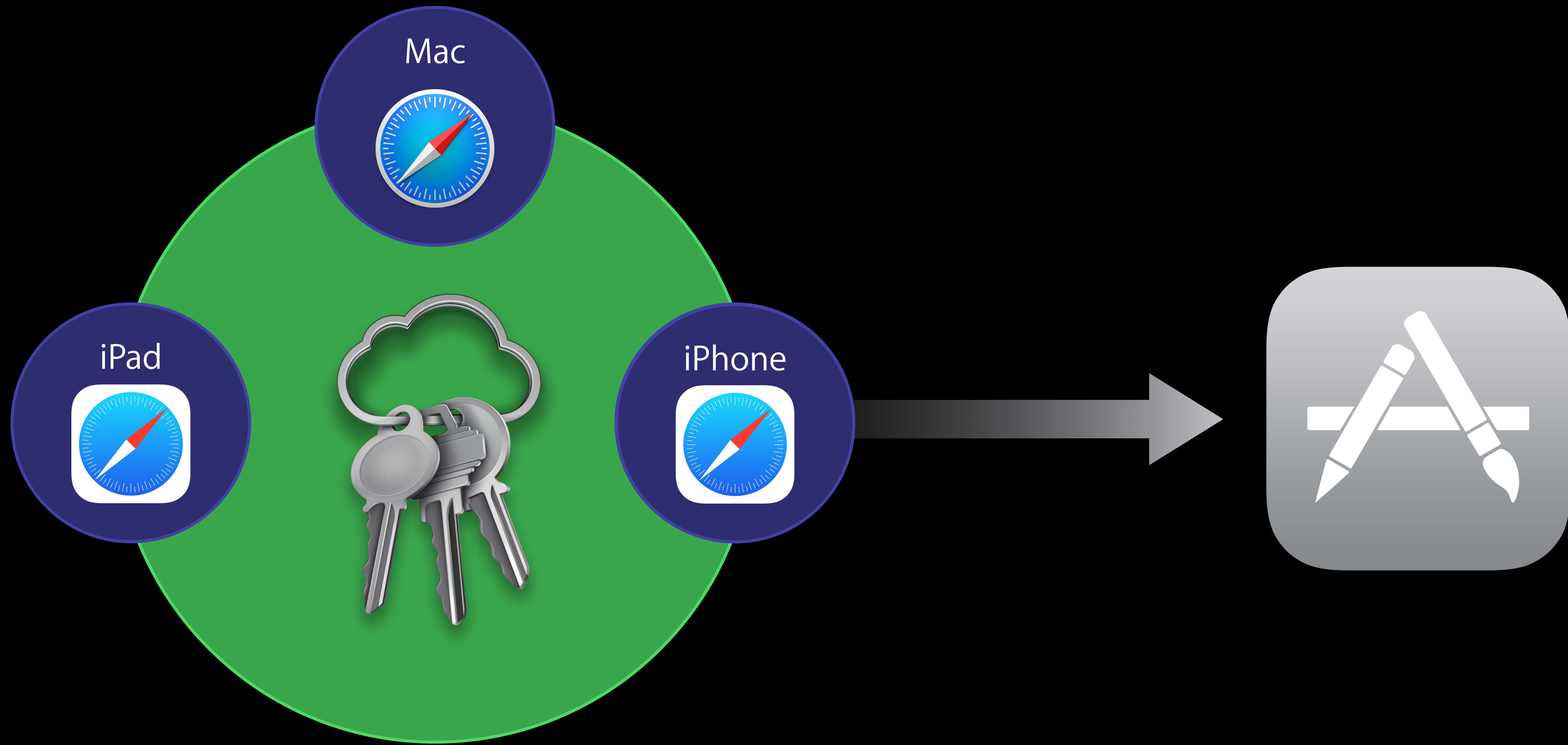
# Credentials in Safari



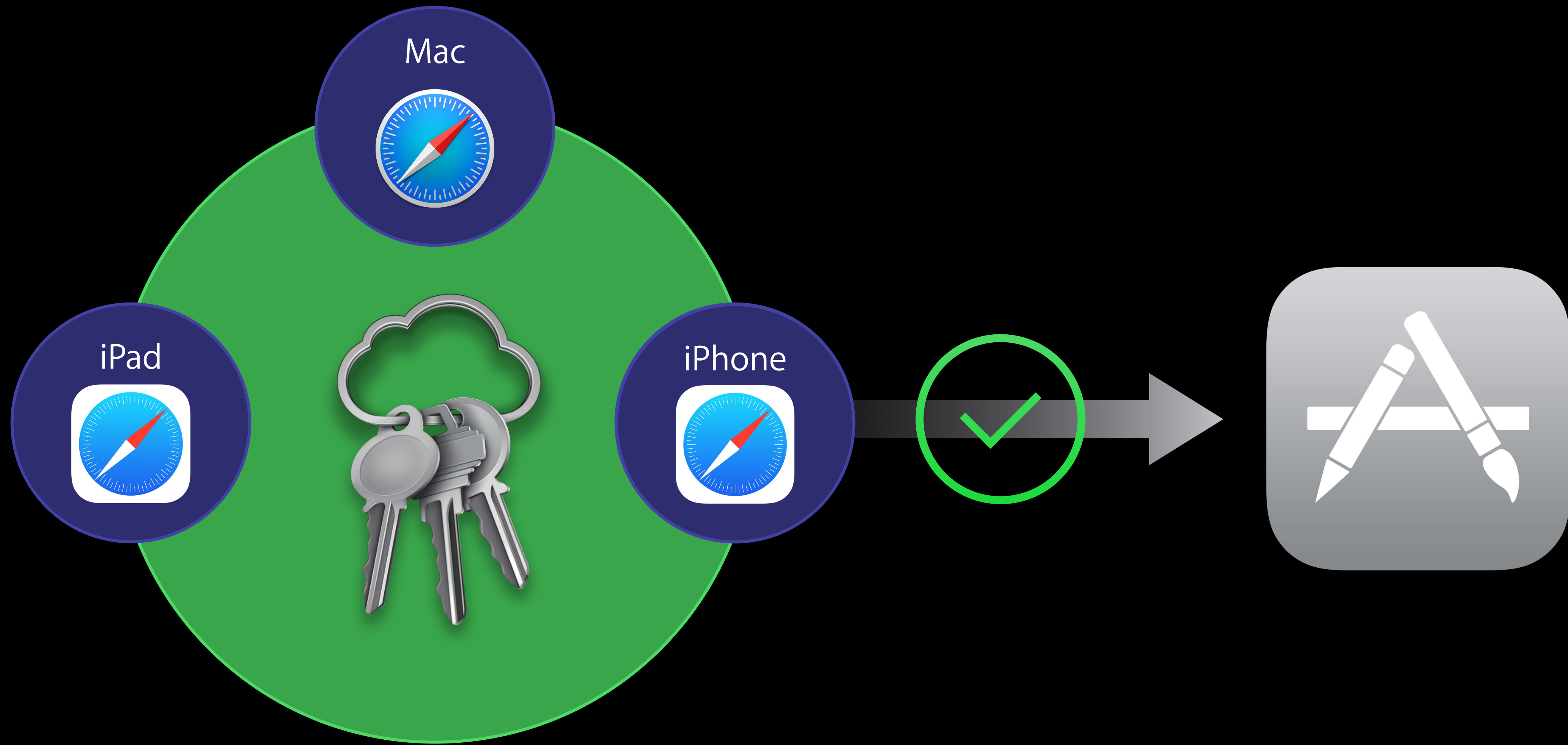
# Credentials in Safari



# Credentials in Safari



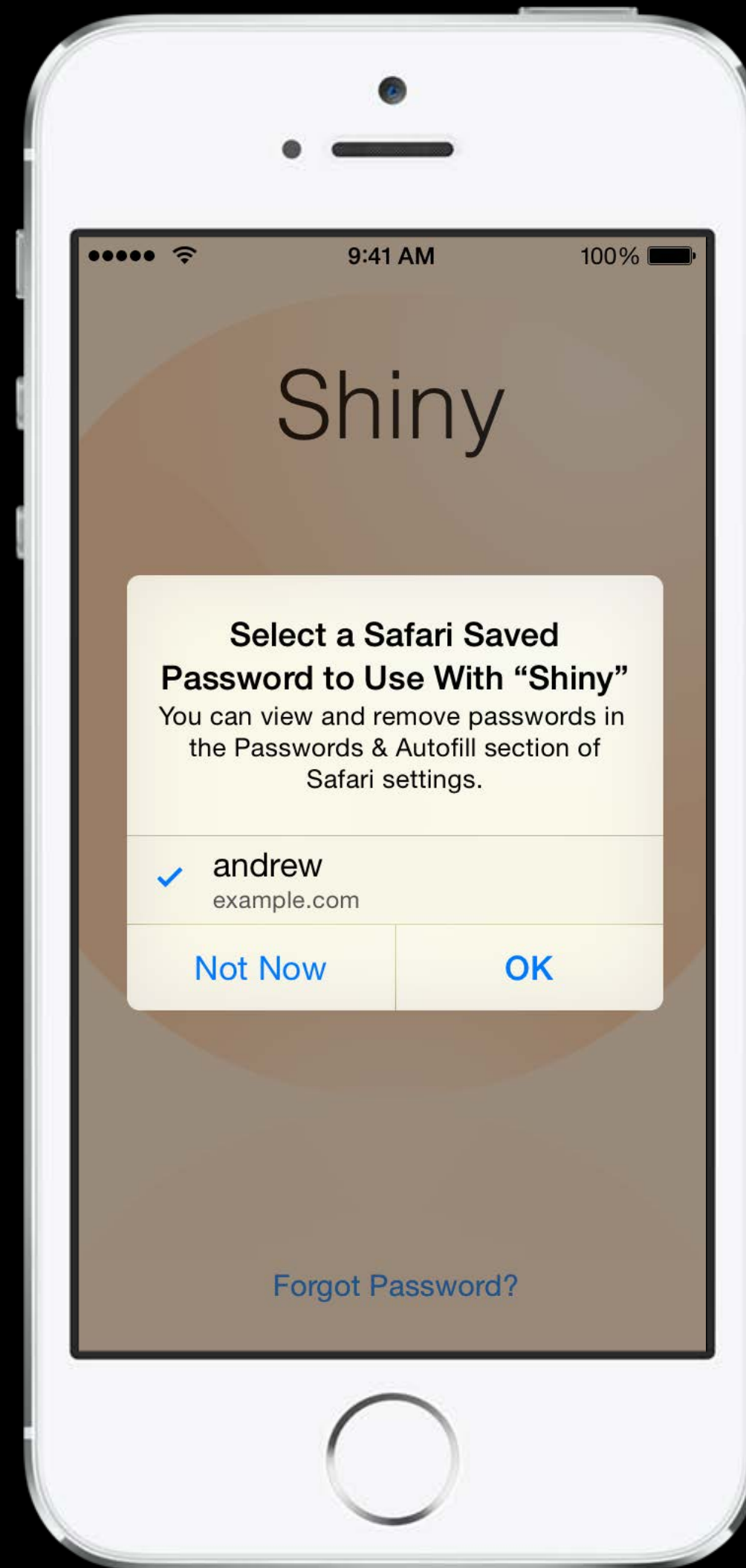
# Credentials in Safari



# Credentials in Safari



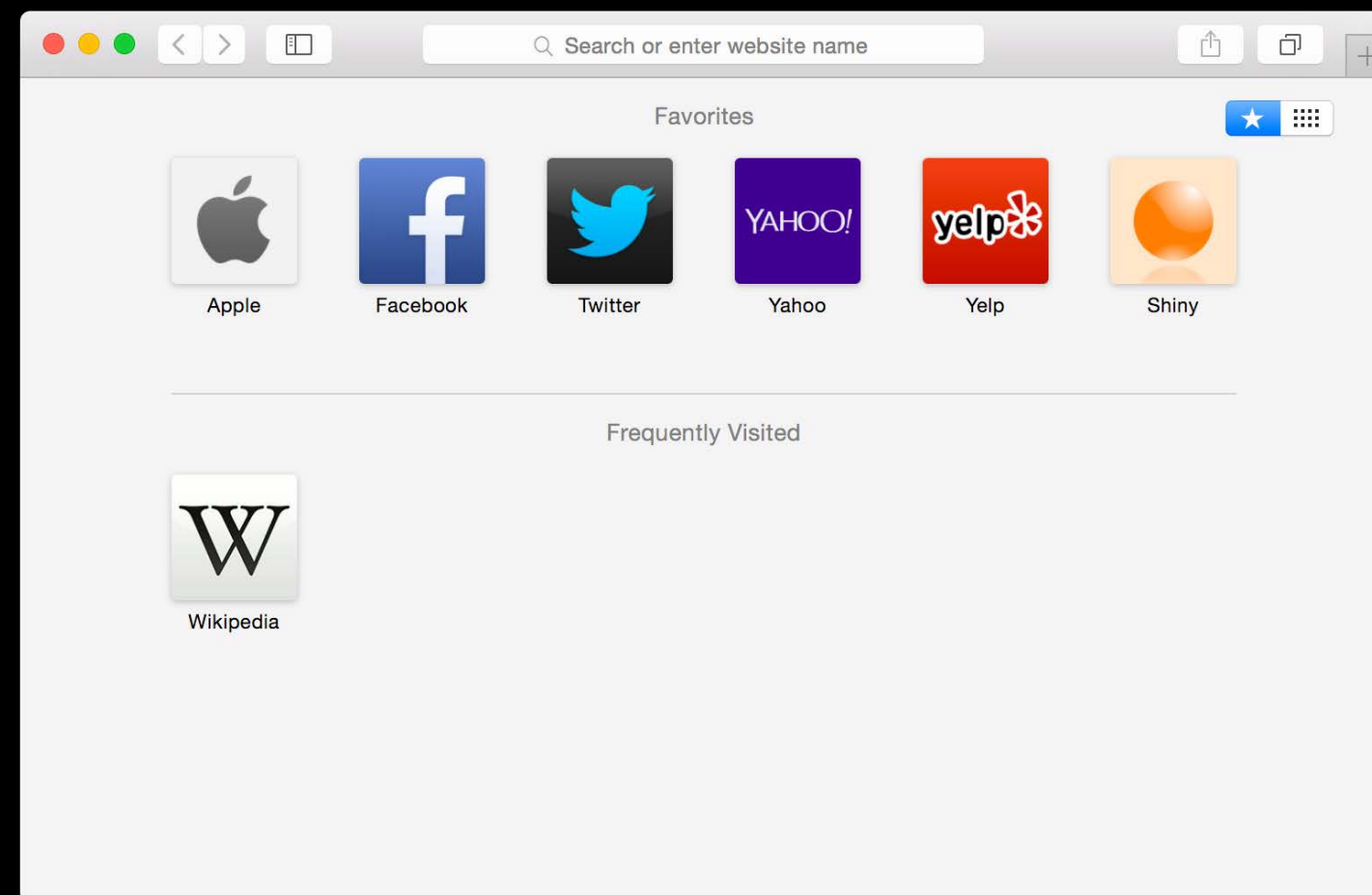
# Credentials in Safari



# Safari Credentials in Native Apps

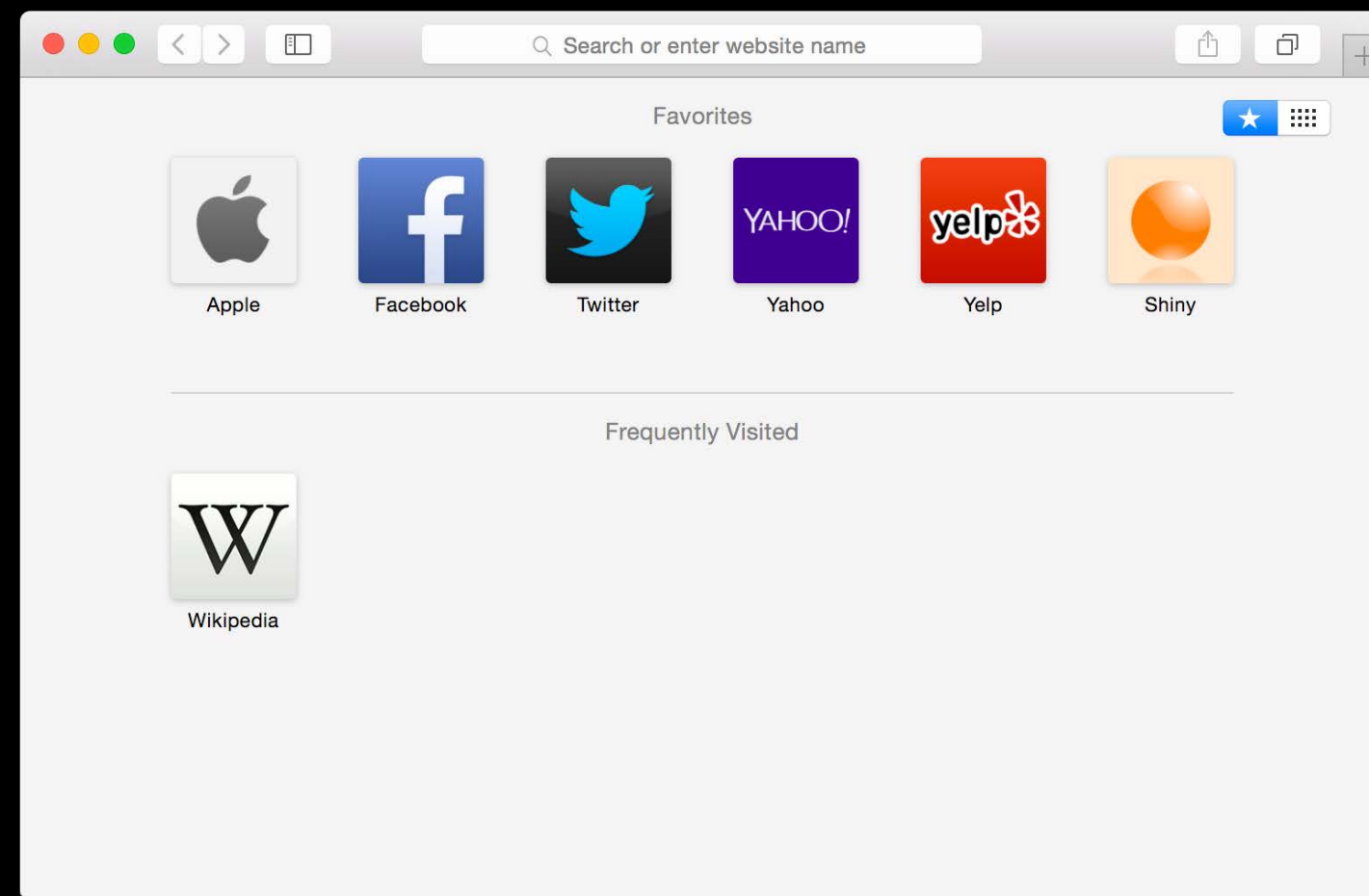
Andrew Whalley  
Core OS Security Engineering

# Password Security

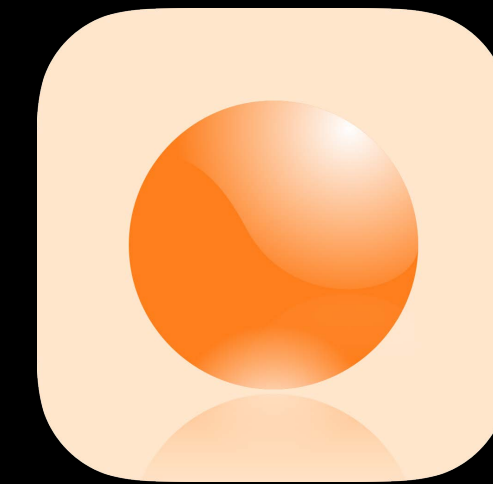
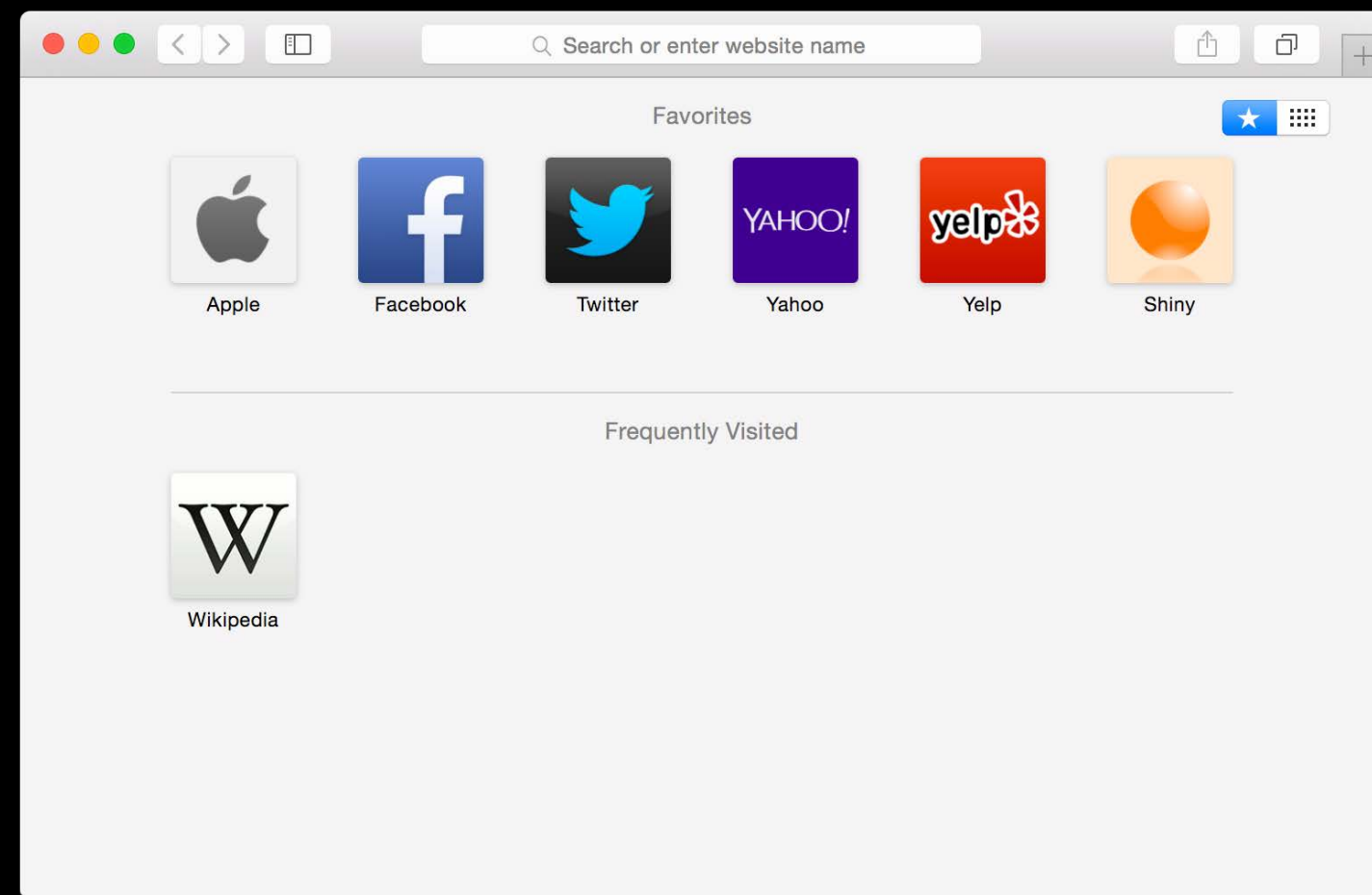




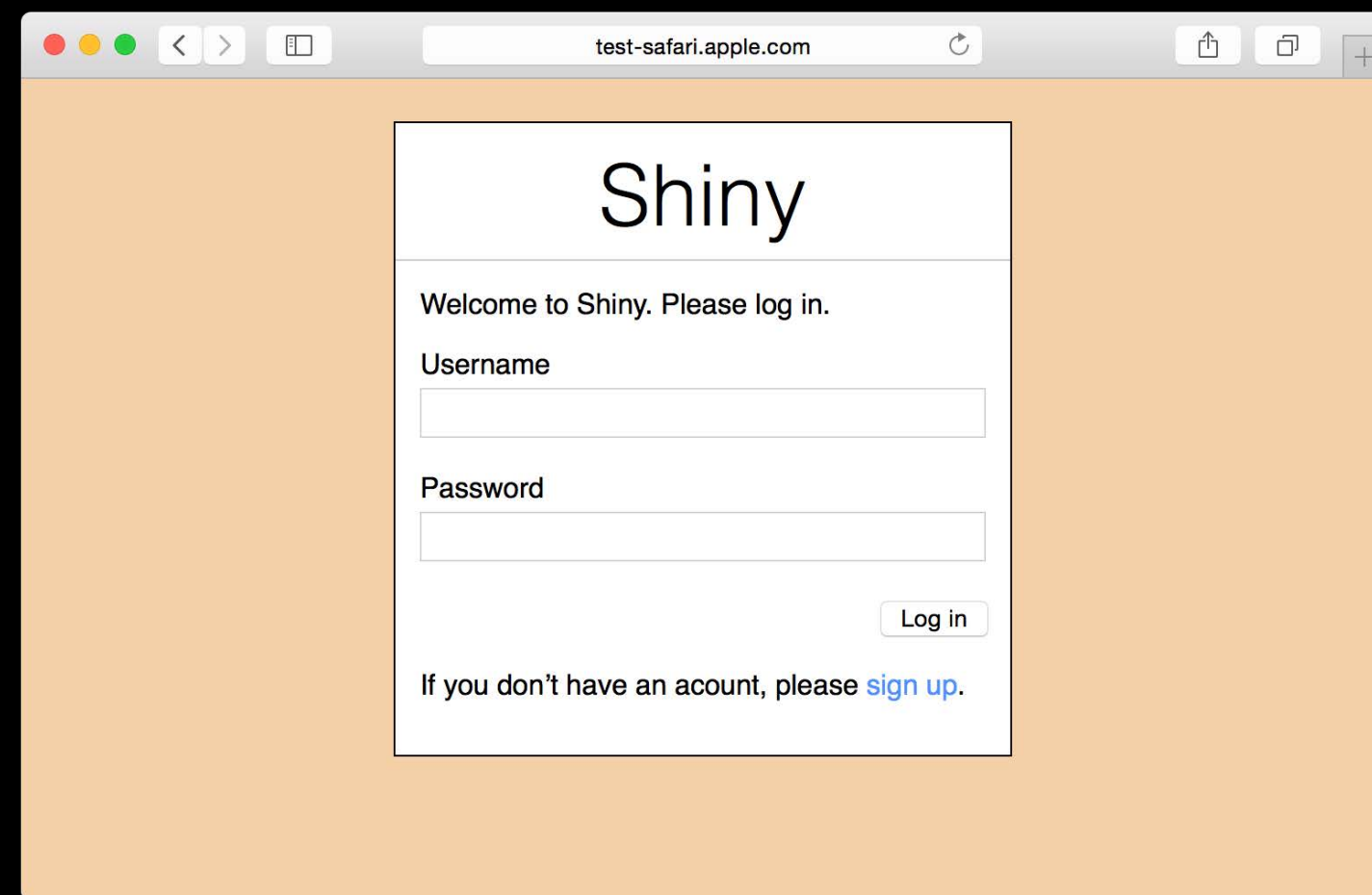
# Password Security



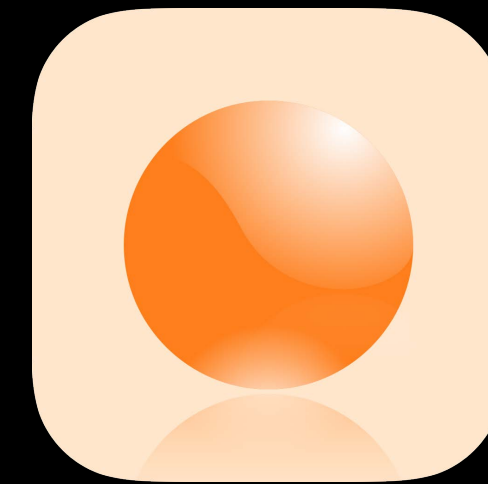
# Password Security



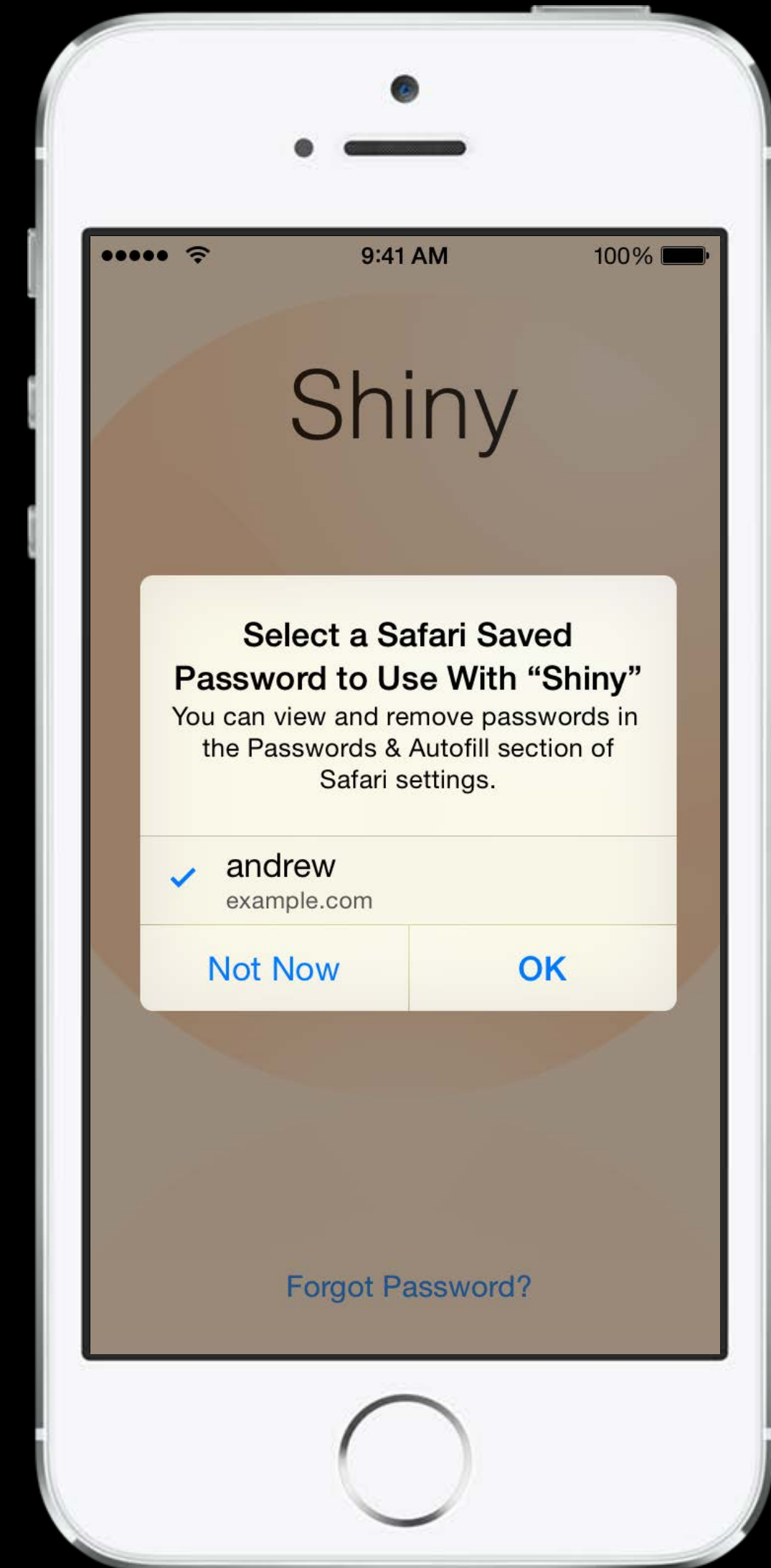
# Password Security



andrew  
afz-zma-9JX-puL

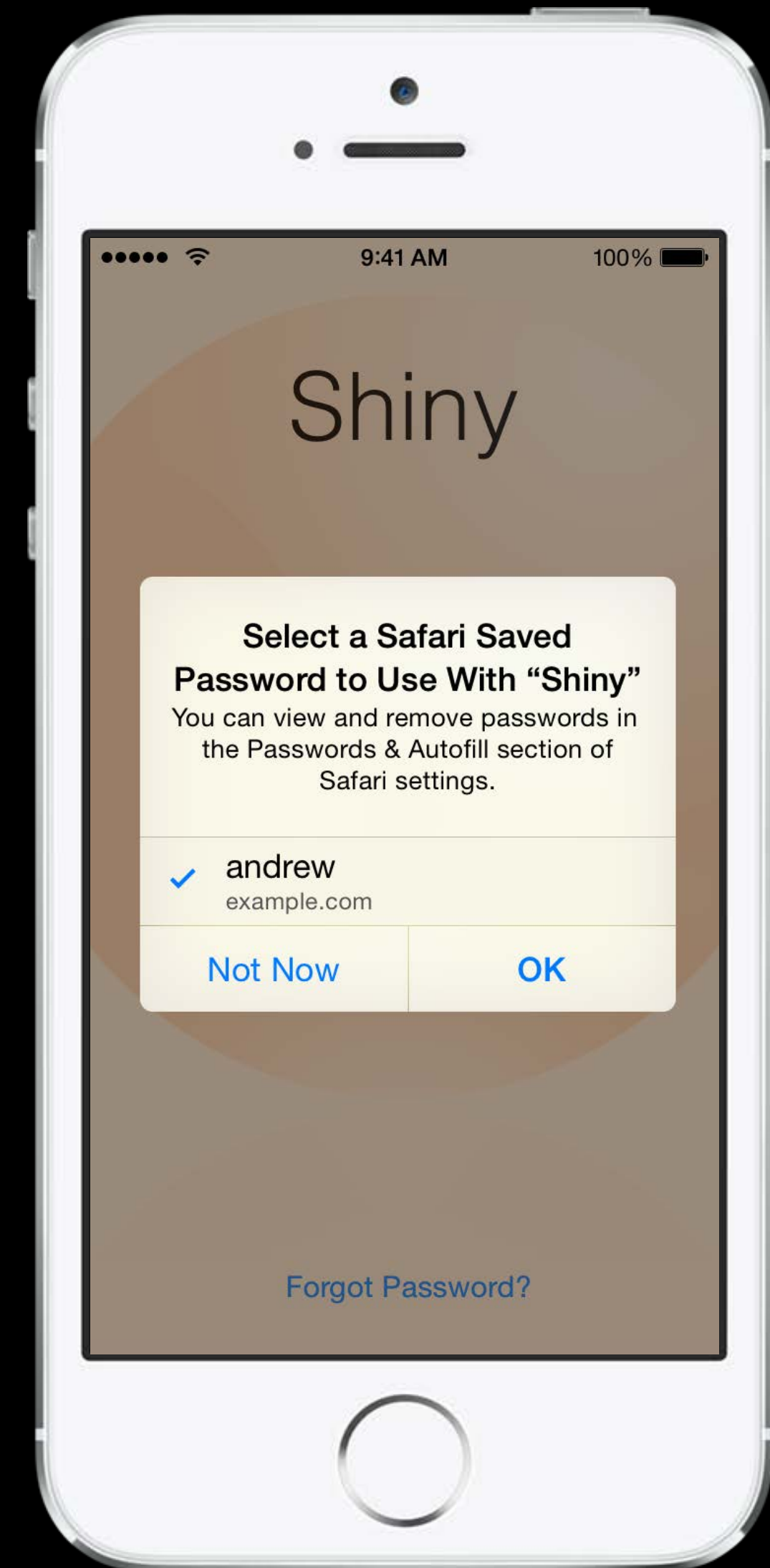


# Overview



# Overview

Associate your app and website



# Overview

Associate your app and website

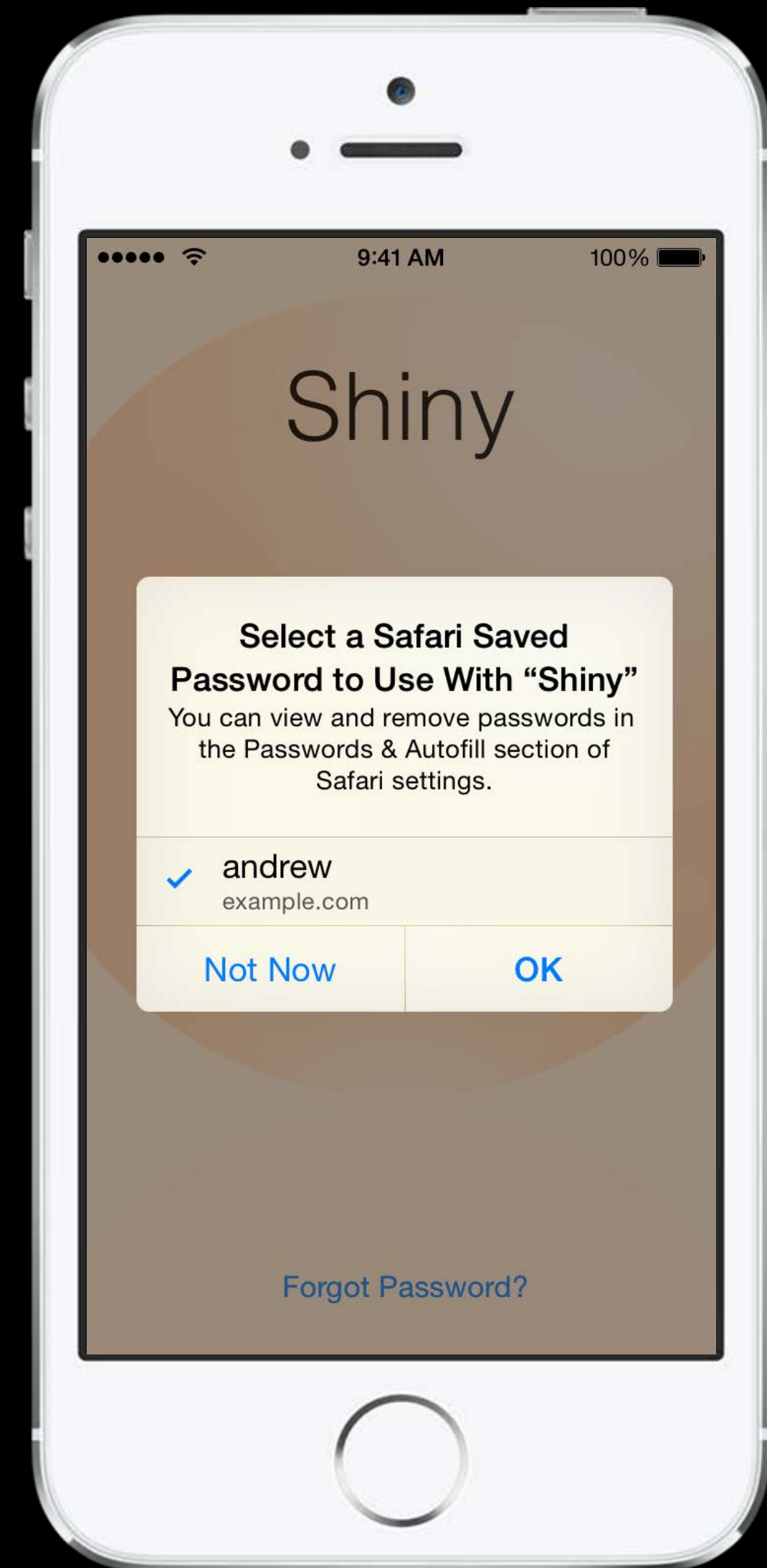
- One file on your website



# Overview

Associate your app and website

- One file on your website
- One new entitlement

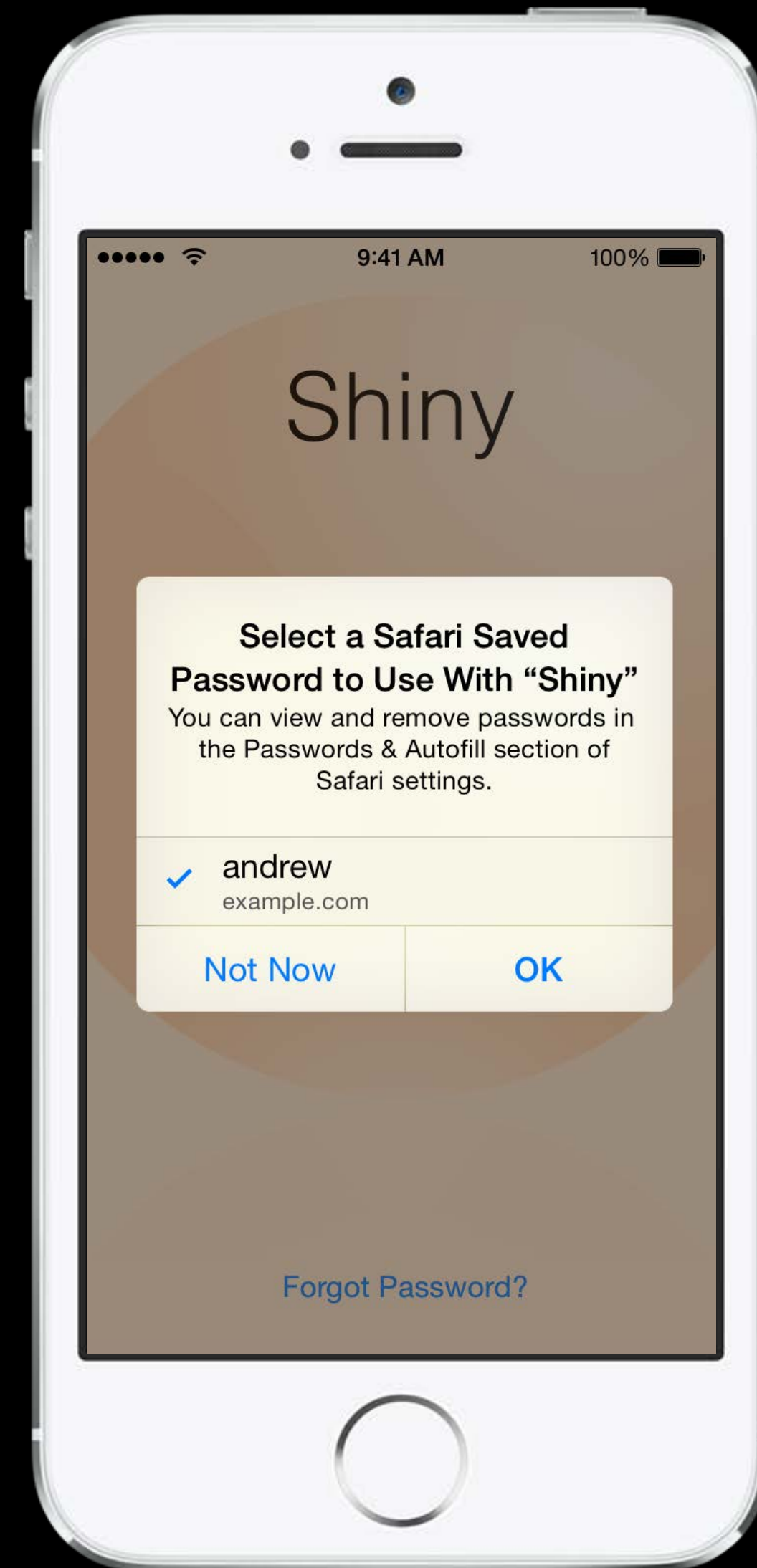


# Overview

Associate your app and website

- One file on your website
- One new entitlement

Simple APIs





# Overview

Associate your app and website

- One file on your website
- One new entitlement

Simple APIs

- Ask for credentials



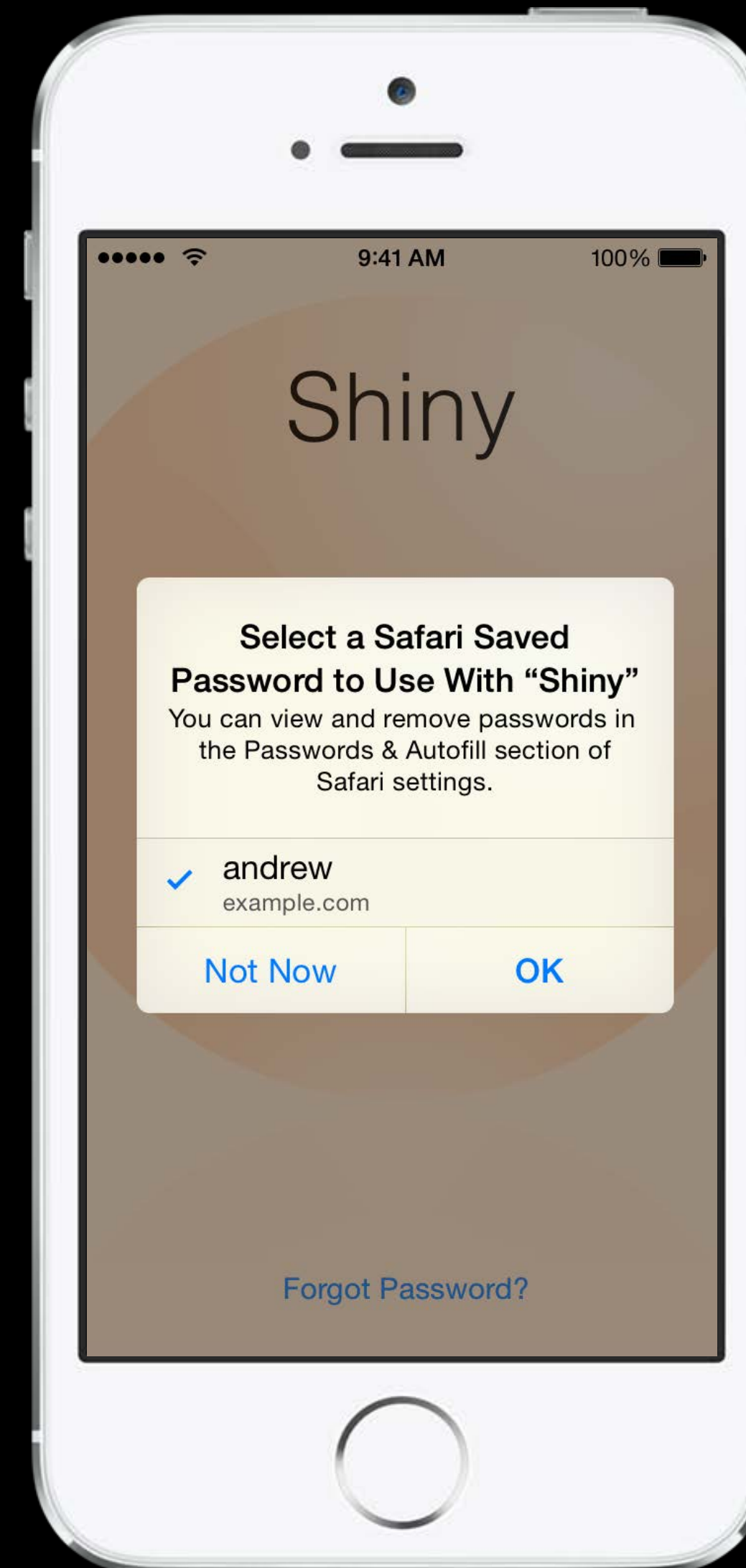
# Overview

Associate your app and website

- One file on your website
- One new entitlement

Simple APIs

- Ask for credentials
- Create or update credentials



# Associating App and Website



# Associating App and Website



# Associating App and Website



# Associating App and Website

# Associating App and Website

App	Website	Status
Shiny App	example.com	Pending
Foo Client	foo.com	Approved
Foo Admin App	foo.com	Approved

# Associating App and Website



Shiny App



example.com



# Associating App and Website



Shiny App



example.com

# Associating App and Website



Shiny App



example.com

# Associating App and Website



Shiny App



example.com

<https://example.com/apple-app-site-association>

# Associating App and Website



Shiny App



example.com

<https://example.com/apple-app-site-association>

# Associating App and Website



Shiny App



example.com

# Associating App and Website



Shiny App



example.com

# Associating App and Website



Shiny App



example.com

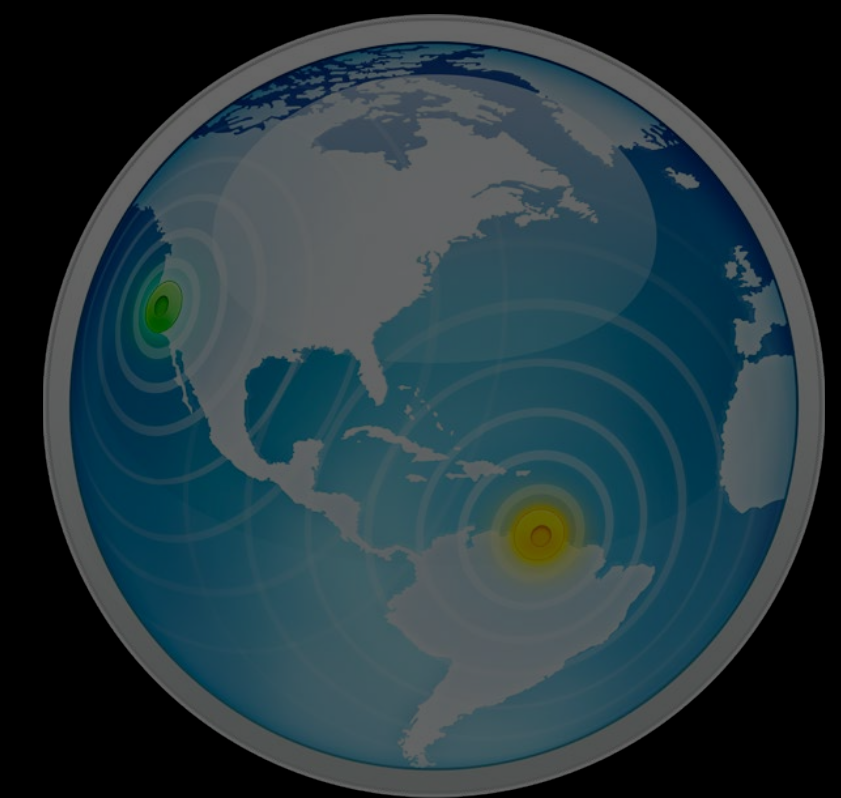
Shiny App



# Associating App and Website



Shiny App



example.com





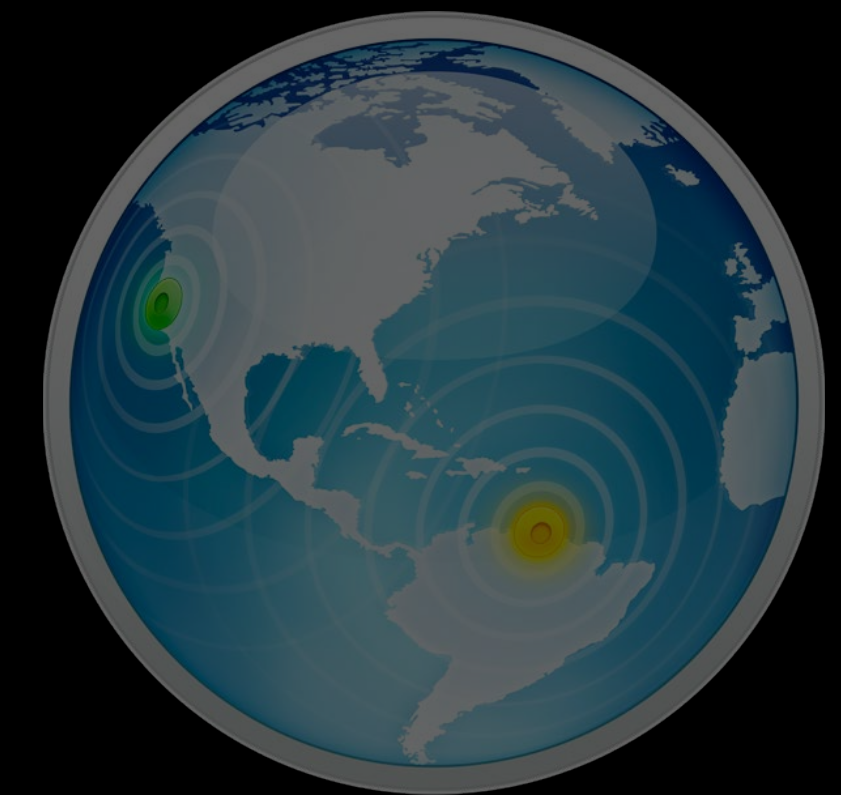
# Associating App and Website



Shiny App



Shiny App



example.com

# Associating App and Website

Shiny App

example.com

# Associating App and Website

App	Website	Status
Shiny App	example.com	Pending
Foo Client	foo.com	Approved
Foo Admin App	foo.com	Approved

# Associating App and Website

App	Website	Status
Shiny App	example.com	Approved
Foo Client	foo.com	Approved
Foo Admin App	foo.com	Approved

# Determining App Intent

Entitlements

# Determining App Intent

Entitlements

application-identifier

# Determining App Intent

## Entitlements

application-identifier

teamid.bundle-identifier

# Determining App Intent

## Entitlements

application-identifier

teamid.bundle-identifier

YWBN8XTPBJ.com.example.Shiny

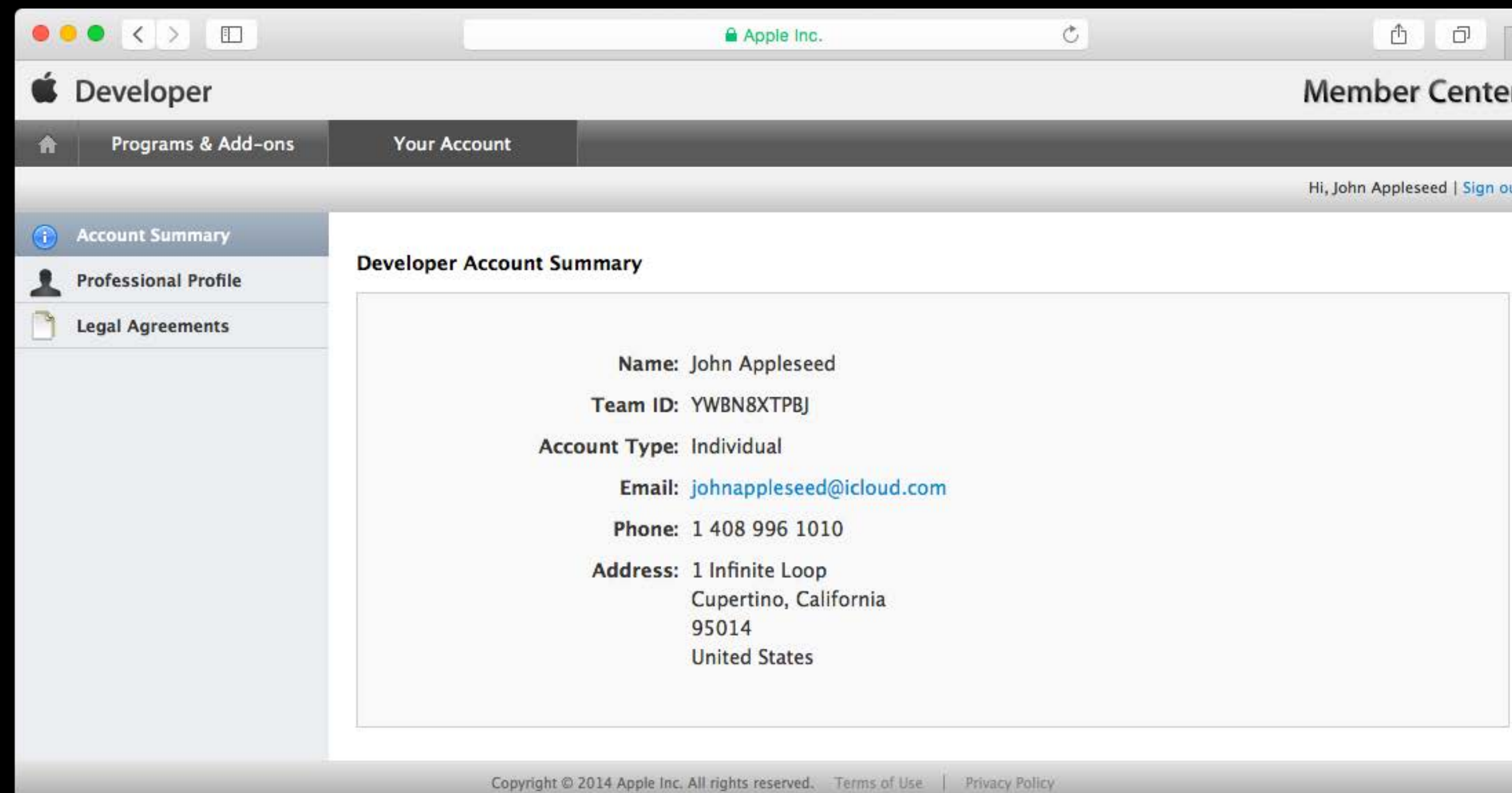


# Determining App Intent Entitlements

application-identifier

teamid.bundle-identifier

YWBN8XTPBJ.com.example.Shiny

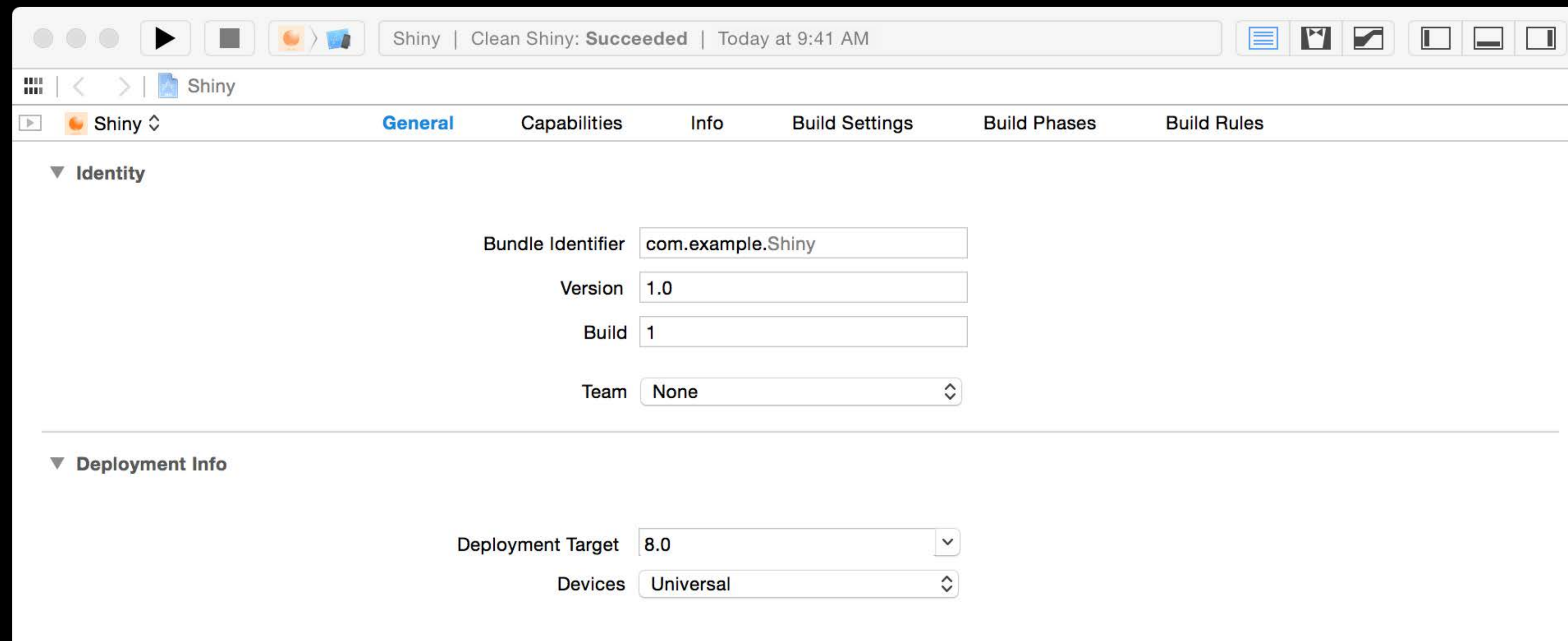


# Determining App Intent Entitlements

application-identifier

teamid.bundle-identifier

YWBN8XTPBJ.com.example.Shiny



# Determining App Intent

## Entitlements

application-identifier

teamid.bundle-identifier

YWBN8XTPBJ.com.example.Shiny

# Determining App Intent

## Entitlements

`application-identifier`

`teamid.bundle-identifier`

`YWBN8XTPBJ.com.example.Shiny`

`com.apple.developer.associated-domains`

# Determining App Intent

## Entitlements

`application-identifier`

`teamid.bundle-identifier`

`YWBN8XTPBJ.com.example.Shiny`

`com.apple.developer.associated-domains`

`service:fully.qualified.domain.name[:port number]`

# Determining App Intent

## Entitlements

`application-identifier`

`teamid.bundle-identifier`

`YWBN8XTPBJ.com.example.Shiny`

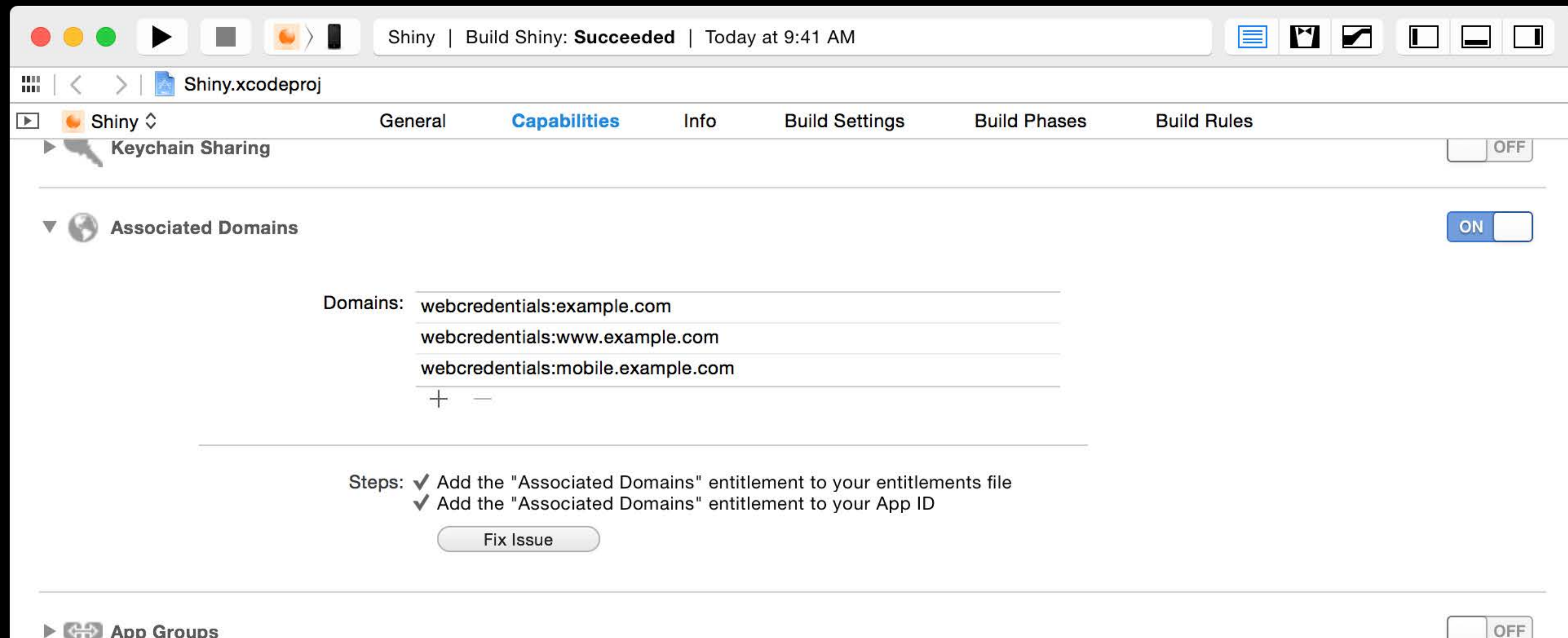
`com.apple.developer.associated-domains`

`service:fully.qualified.domain.name[:port number]`

`webcredentials:example.com`

# associated-domains Entitlement

## Xcode



# Associating App and Website





# Associating App and Website



# Determining Website Intent

<https://example.com/apple-app-site-association>

# Determining Website Intent

<https://example.com/apple-app-site-association>

```
{
  "webcredentials": {
    "apps": [
      "YWBN8XTPBJ.com.example.Shiny",
      "YWBN8XTPBJ.com.example.ShinyUploader" ]
    }
  }
```

# Determining Website Intent

<https://example.com/apple-app-site-association>

```
{
  "webcredentials": {
    "apps": [
      "YWBN8XTPBJ.com.example.Shiny",
      "YWBN8XTPBJ.com.example.ShinyUploader" ]
    }
  }
```

# Determining Website Intent

<https://example.com/apple-app-site-association>

```
{
  "webcredentials": {
    "apps": [
      "YWBN8XTPBJ.com.example.Shiny",
      "YWBN8XTPBJ.com.example.ShinyUploader" ]
  }
}
```

# Determining Website Intent

<https://example.com/apple-app-site-association>

```
{
  "webcredentials": {
    "apps": [
      "YWBN8XTPBJ.com.example.Shiny",
      "YWBN8XTPBJ.com.example.ShinyUploader" ]
    }
  }
```

# Determining Website Intent

Content-Type must be `application/pkcs7-mime`

Signed

- Cryptographic Message Syntax
- S/MIME

# apple-app-site-association File

Signing the JSON



# apple-app-site-association File

## Signing the JSON

```
echo '{"webcredentials":{"apps":["YWBN8XTPBJ.com.example.Shiny",  
"YWBN8XTPBJ.com.example.ShinyUploader"]}}' > json.txt
```

# apple-app-site-association File

## Signing the JSON

```
echo '{"webcredentials":{"apps":["YWBN8XTPBJ.com.example.Shiny",  
"YWBN8XTPBJ.com.example.ShinyUploader"]}}' > json.txt
```

# apple-app-site-association File

## Signing the JSON

```
cat json.txt | openssl smime -sign -inkey example.com.key  
-signer example.com.pem  
-certfile intermediate.pem  
-noattr -nodetach  
-outform DER > apple-app-site-association
```

# apple-app-site-association File

## Signing the JSON

```
cat json.txt | openssl smime -sign -inkey example.com.key  
-signer example.com.pem  
-certfile intermediate.pem  
-noattr -nodetach  
-outform DER > apple-app-site-association
```

# apple-app-site-association File

## Signing the JSON

```
cat json.txt | openssl smime -sign -inkey example.com.key  
-signer example.com.pem  
-certfile intermediate.pem  
-noattr -nodetach  
-outform DER > apple-app-site-association
```

# apple-app-site-association File

## Signing the JSON

```
cat json.txt | openssl smime -sign -inkey example.com.key  
-signer example.com.pem  
-certfile intermediate.pem  
-noattr -nodetach  
-outform DER > apple-app-site-association
```

# Associating App and Website



# Determining Website Intent

Return results



# Determining Website Intent

Return results

HTTP Status

Service and  
Identifier Match

Signature Is Valid

Approval State

---




# Determining Website Intent

Return results

HTTP Status	Service and Identifier Match	Signature Is Valid	Approval State
200			Approved





# Determining Website Intent

Return results

HTTP Status	Service and Identifier Match	Signature Is Valid	Approval State
200			Approved
200			Denied





# Determining Website Intent

## Return results

HTTP Status	Service and Identifier Match	Signature Is Valid	Approval State
200			Approved
200			Denied
300-499	n/a	n/a	Denied

# Determining Website Intent

## Return results

HTTP Status	Service and Identifier Match	Signature Is Valid	Approval State
200			Approved
200			Denied
300-499	n/a	n/a	Denied
500-599	n/a	n/a	Retry

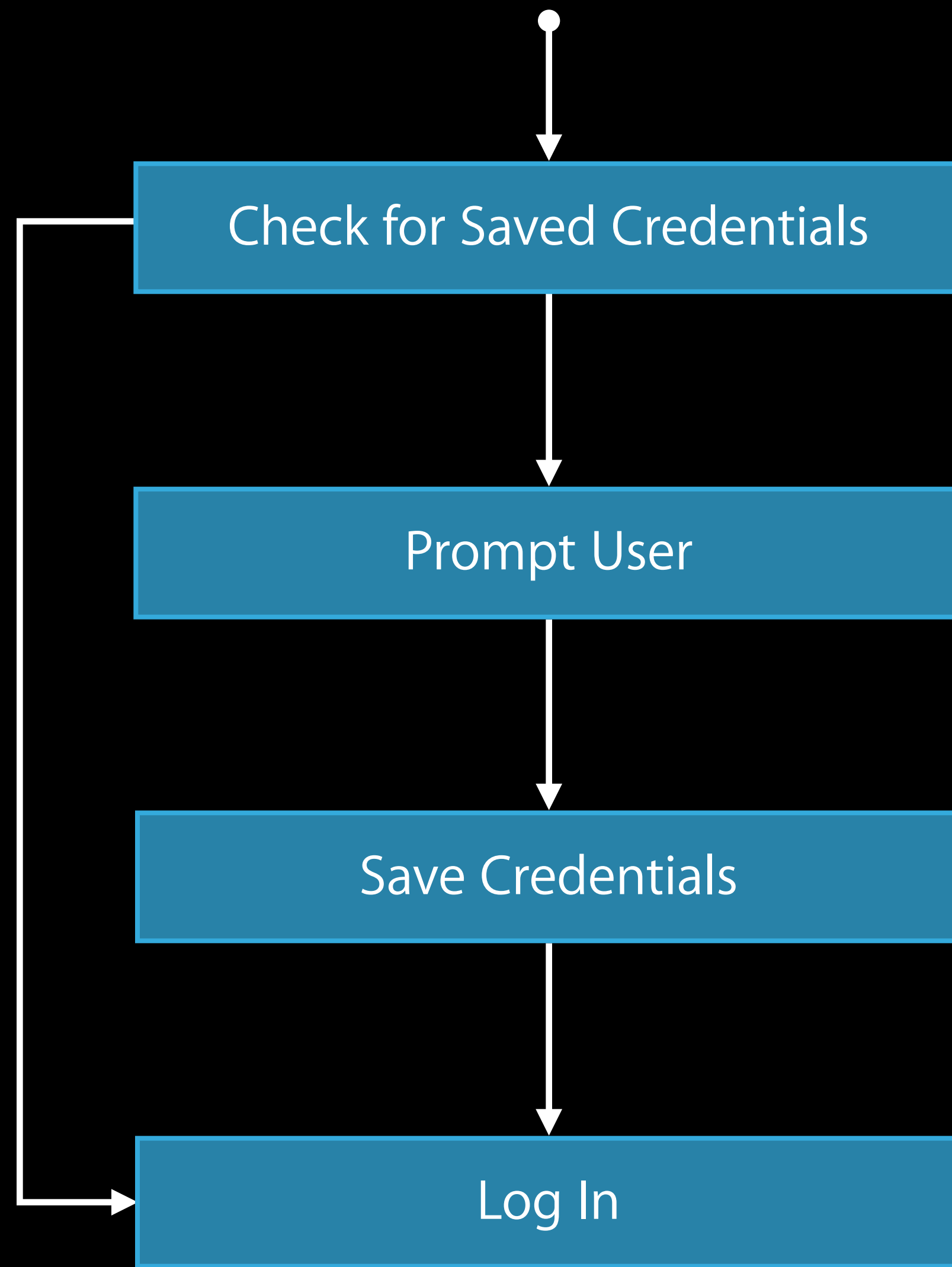
# Native Apps with Accounts



# Native Apps with Accounts

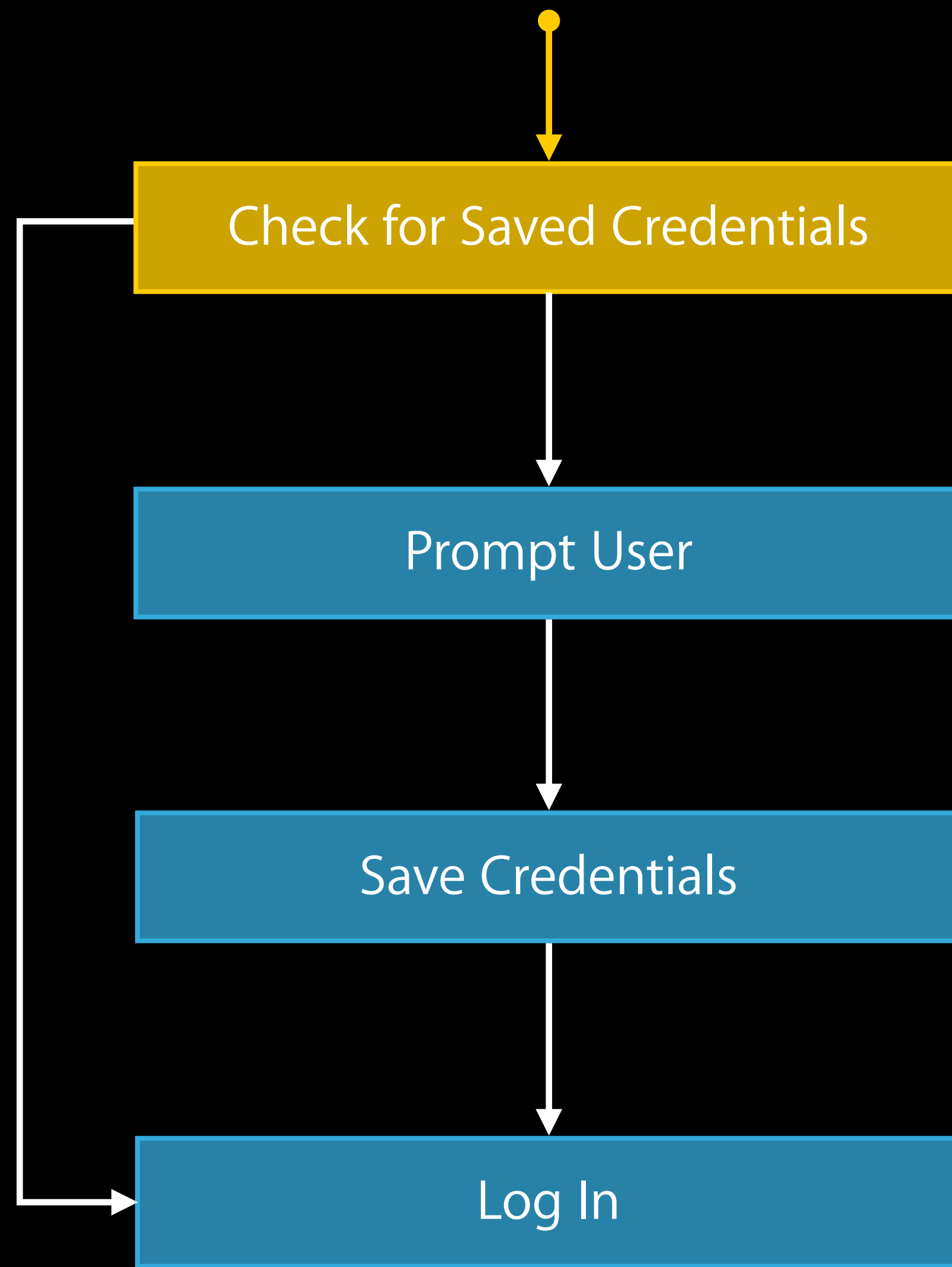


# Native Apps with Accounts

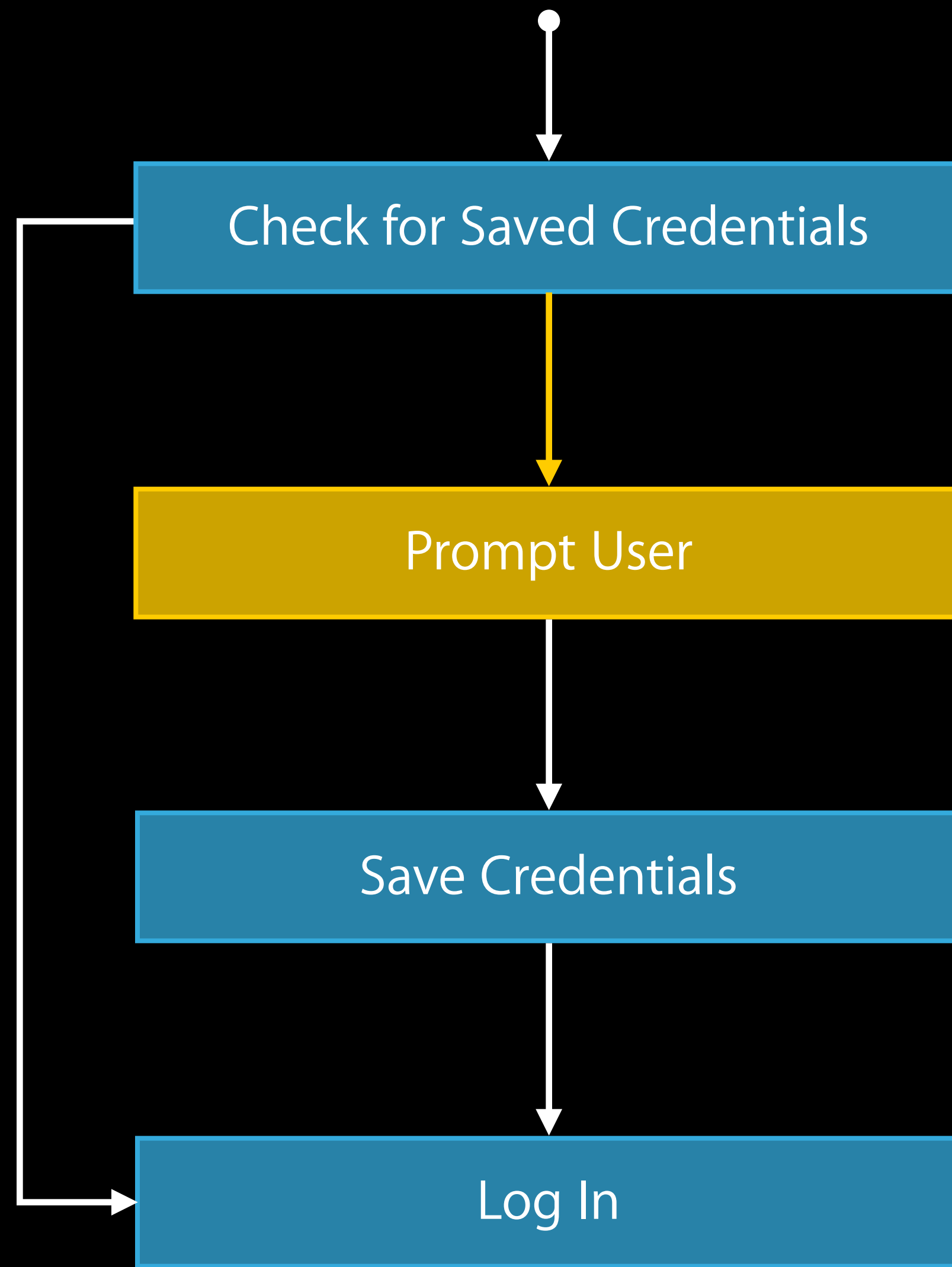




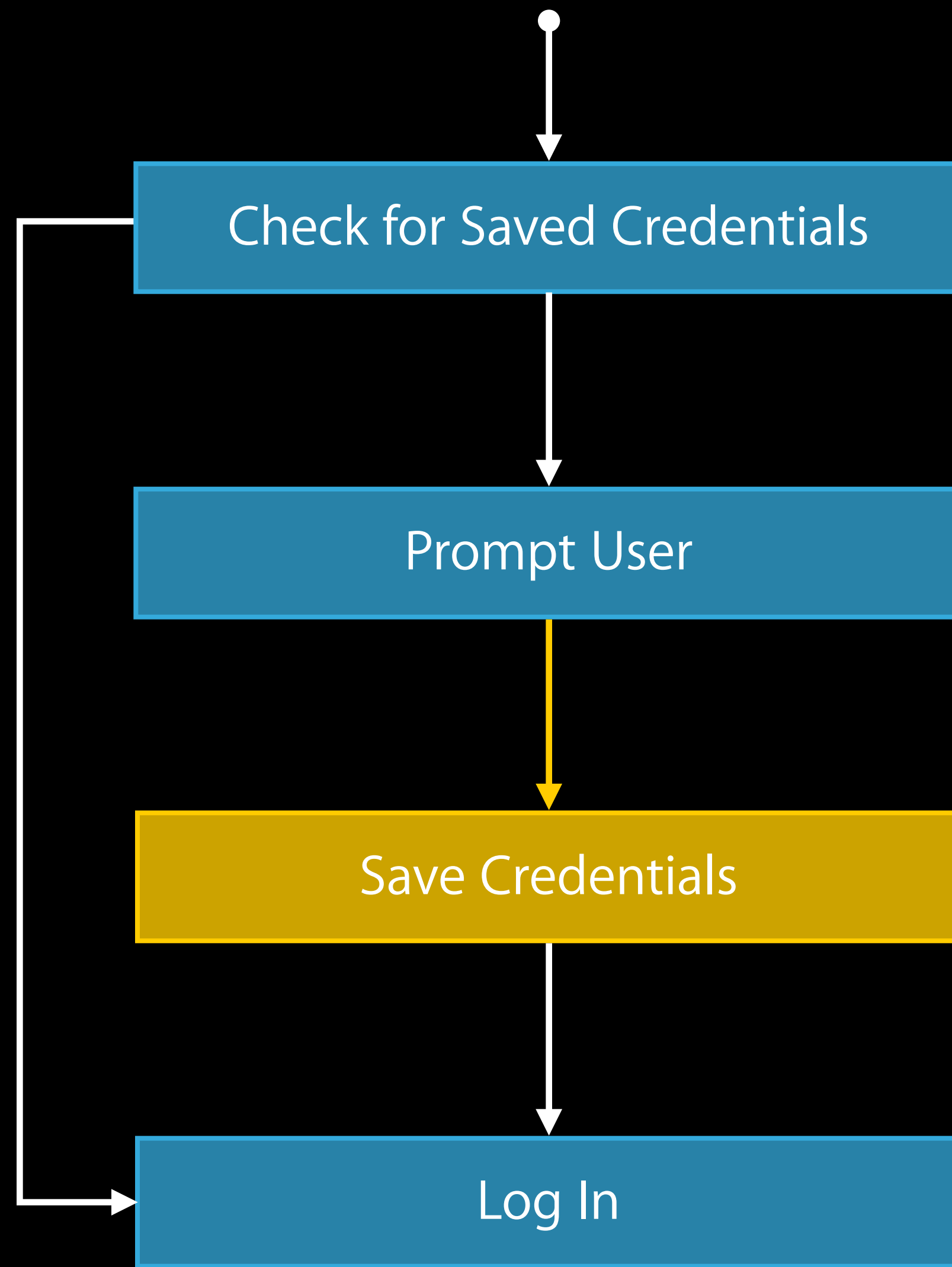
# Native Apps with Accounts



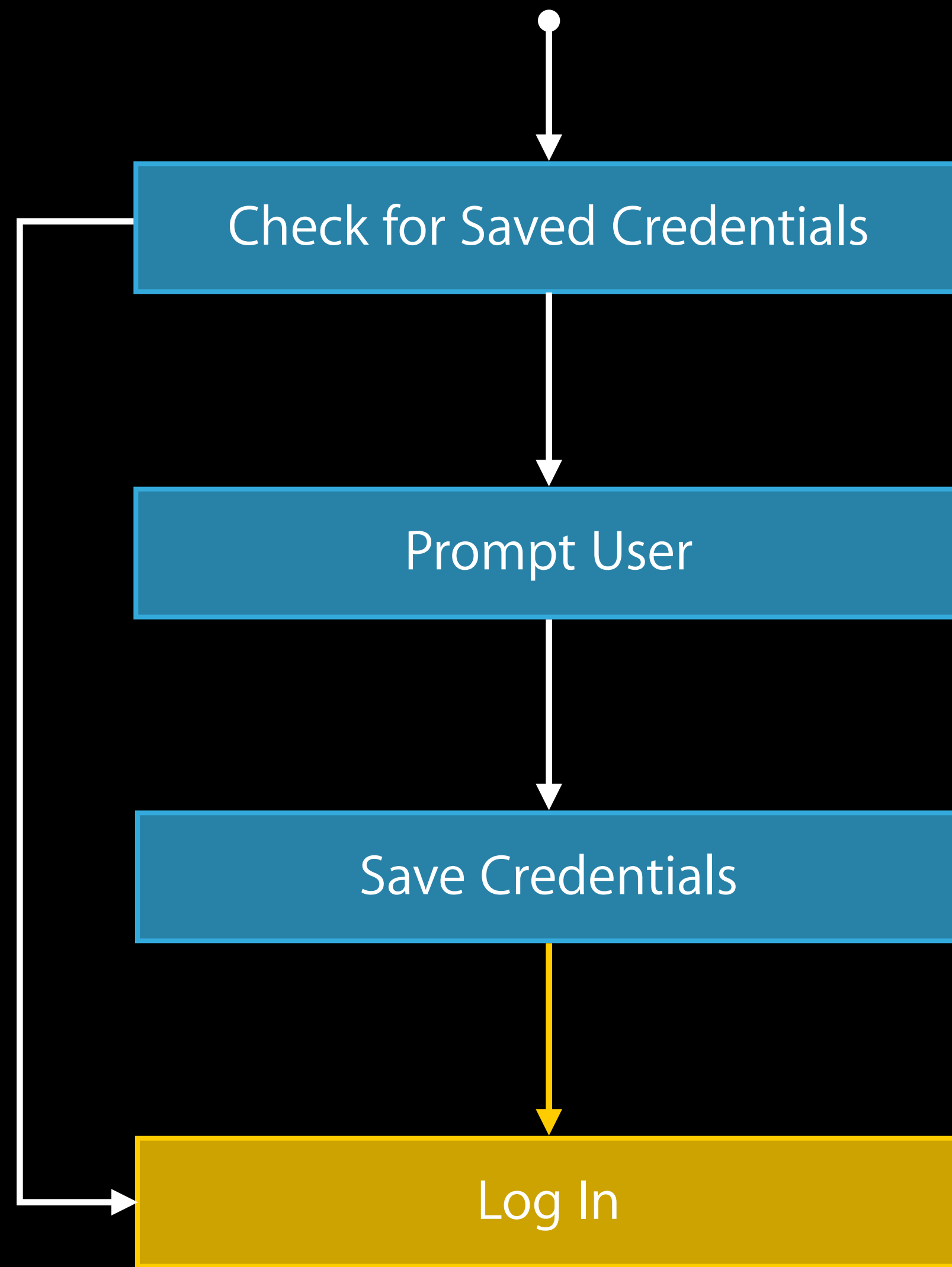
# Native Apps with Accounts



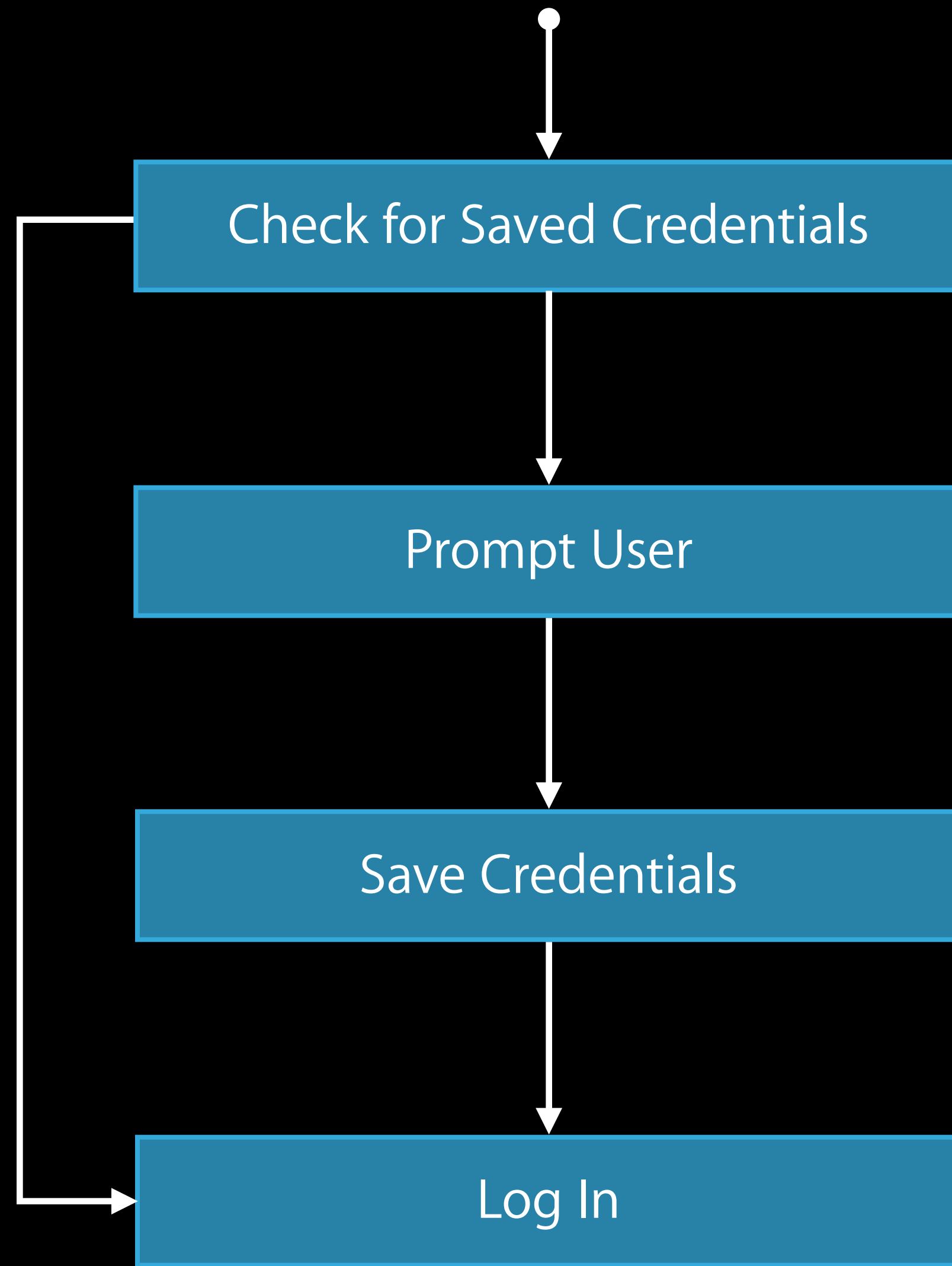
# Native Apps with Accounts



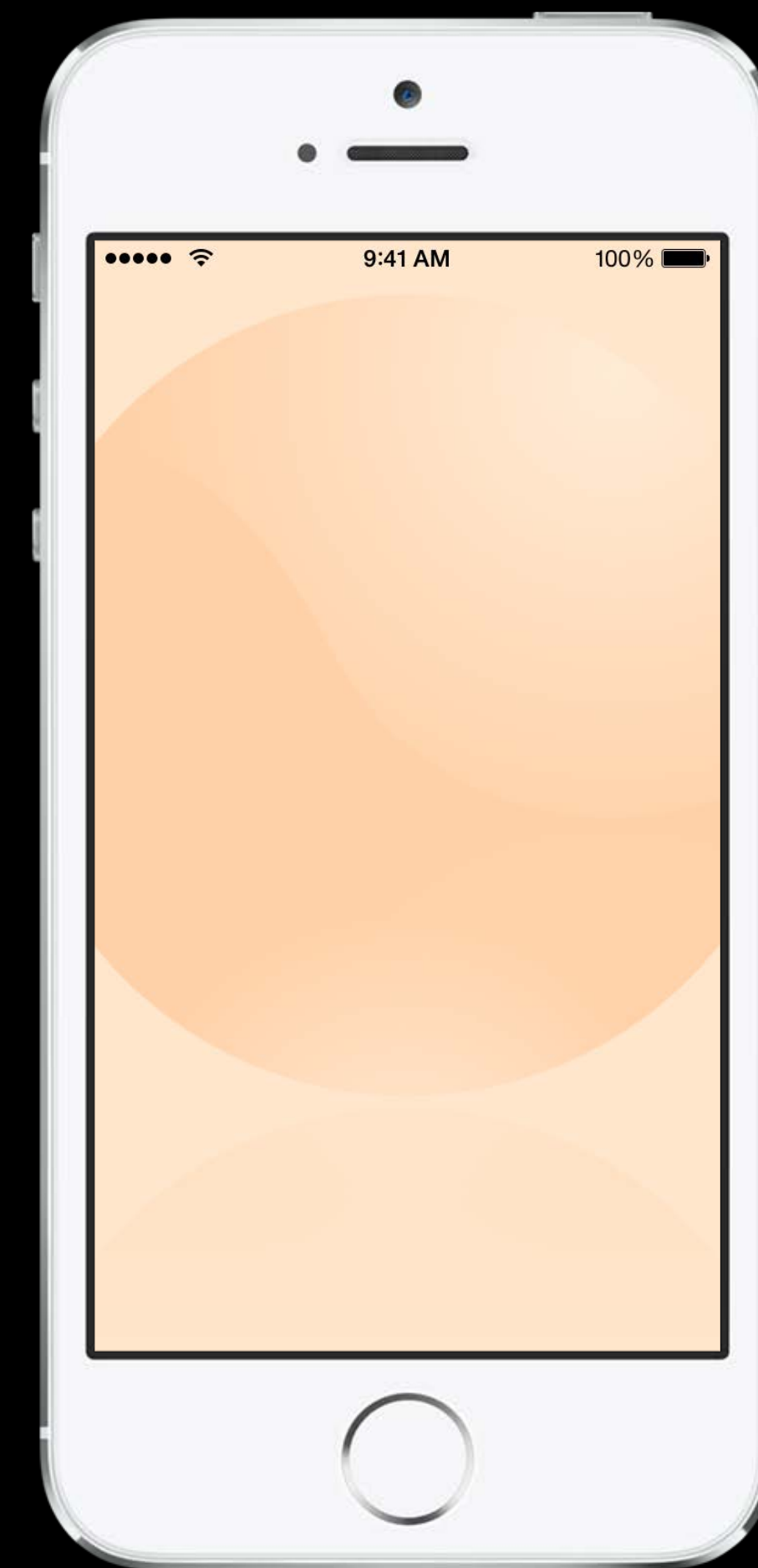
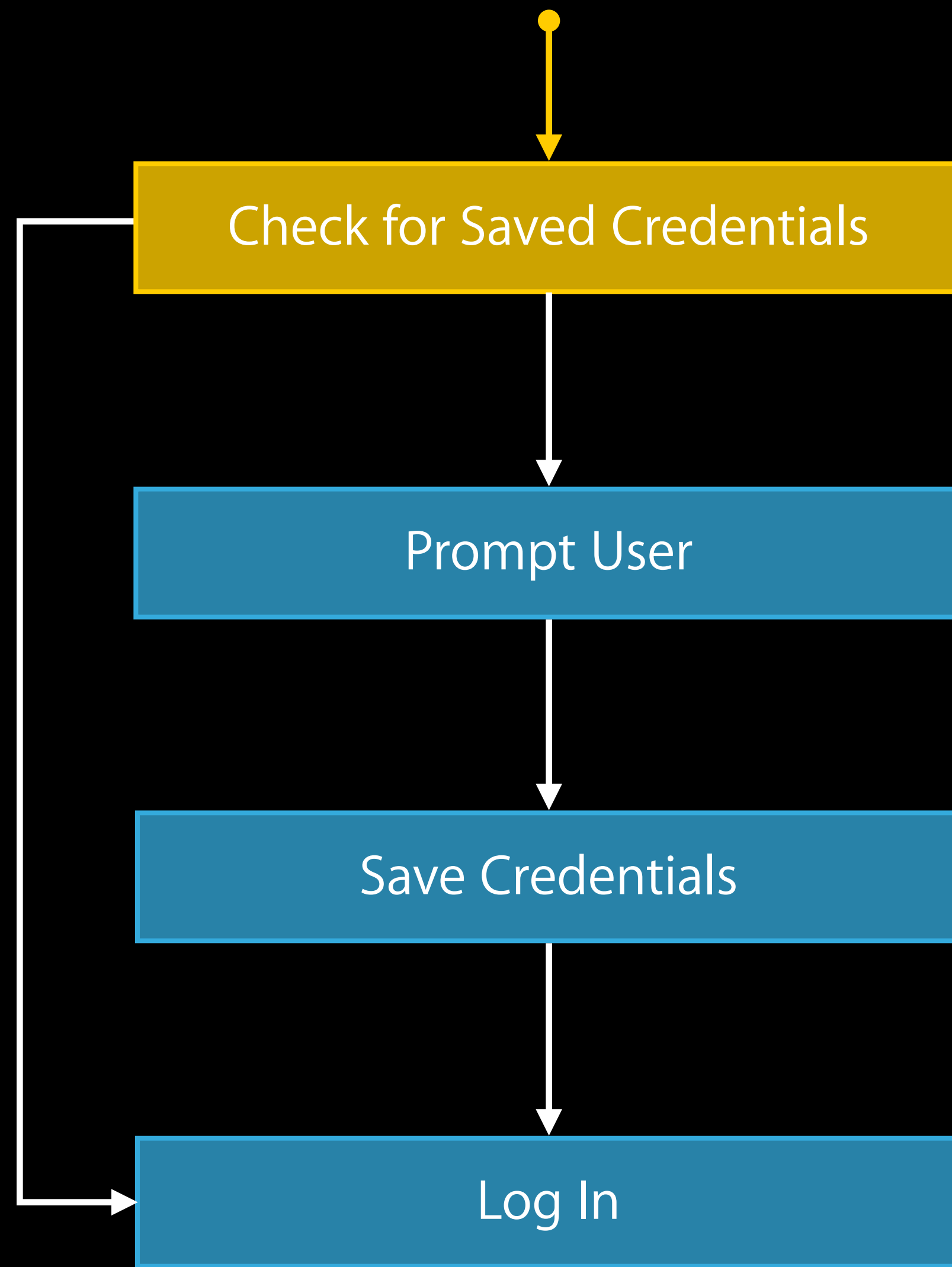
# Native Apps with Accounts



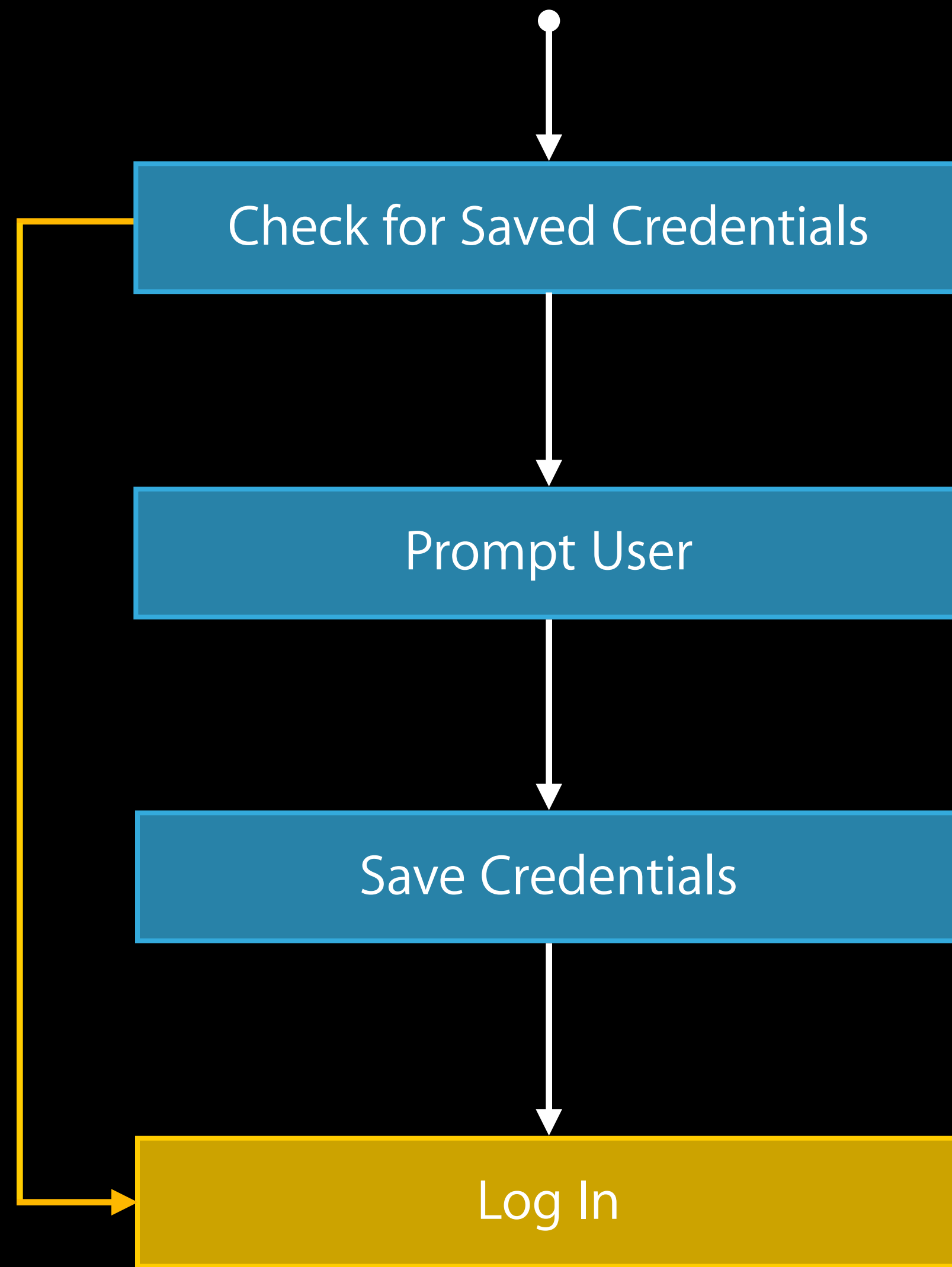
# Native Apps with Accounts



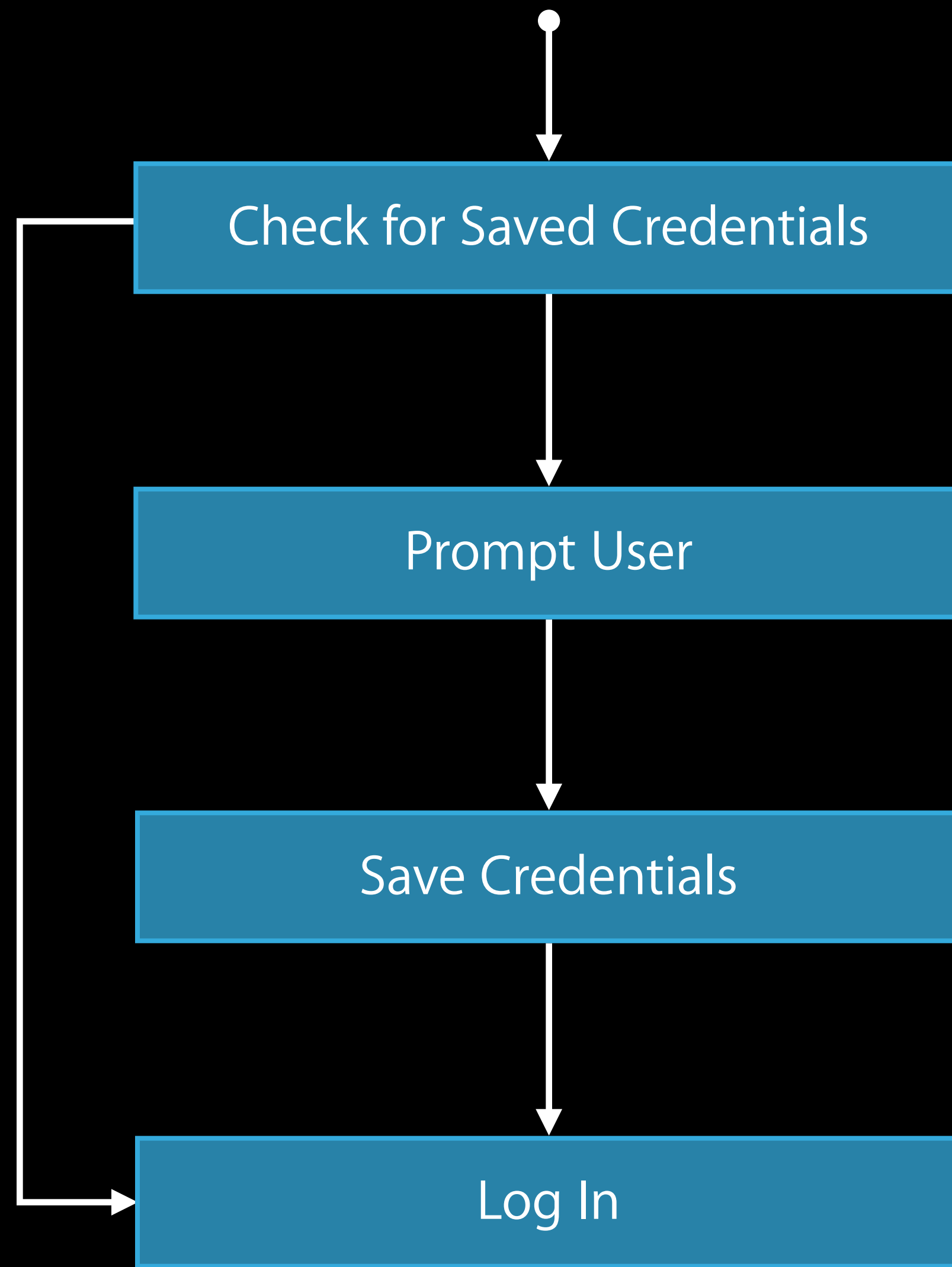
# Native Apps with Accounts



# Native Apps with Accounts

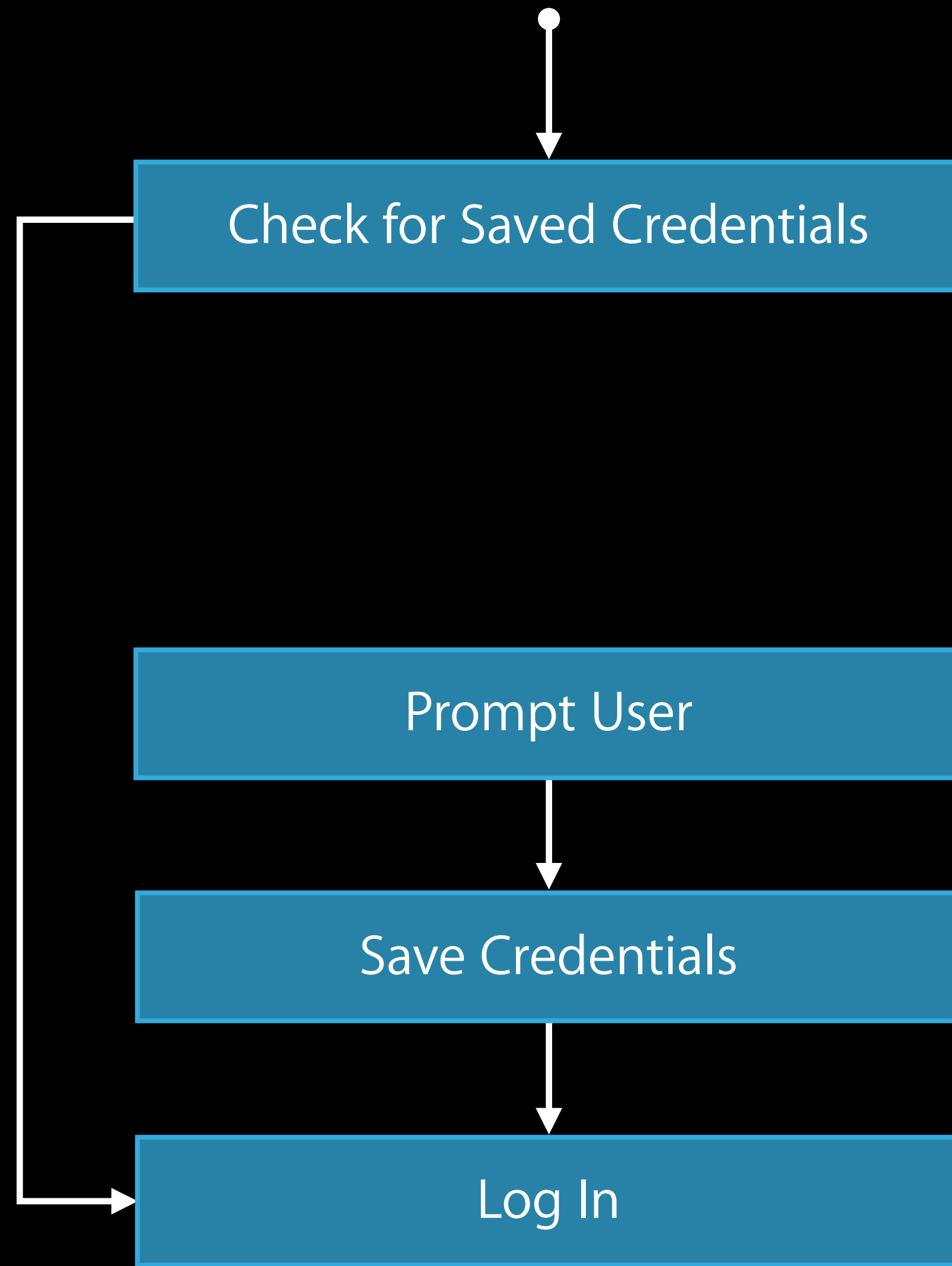


# Native Apps with Accounts

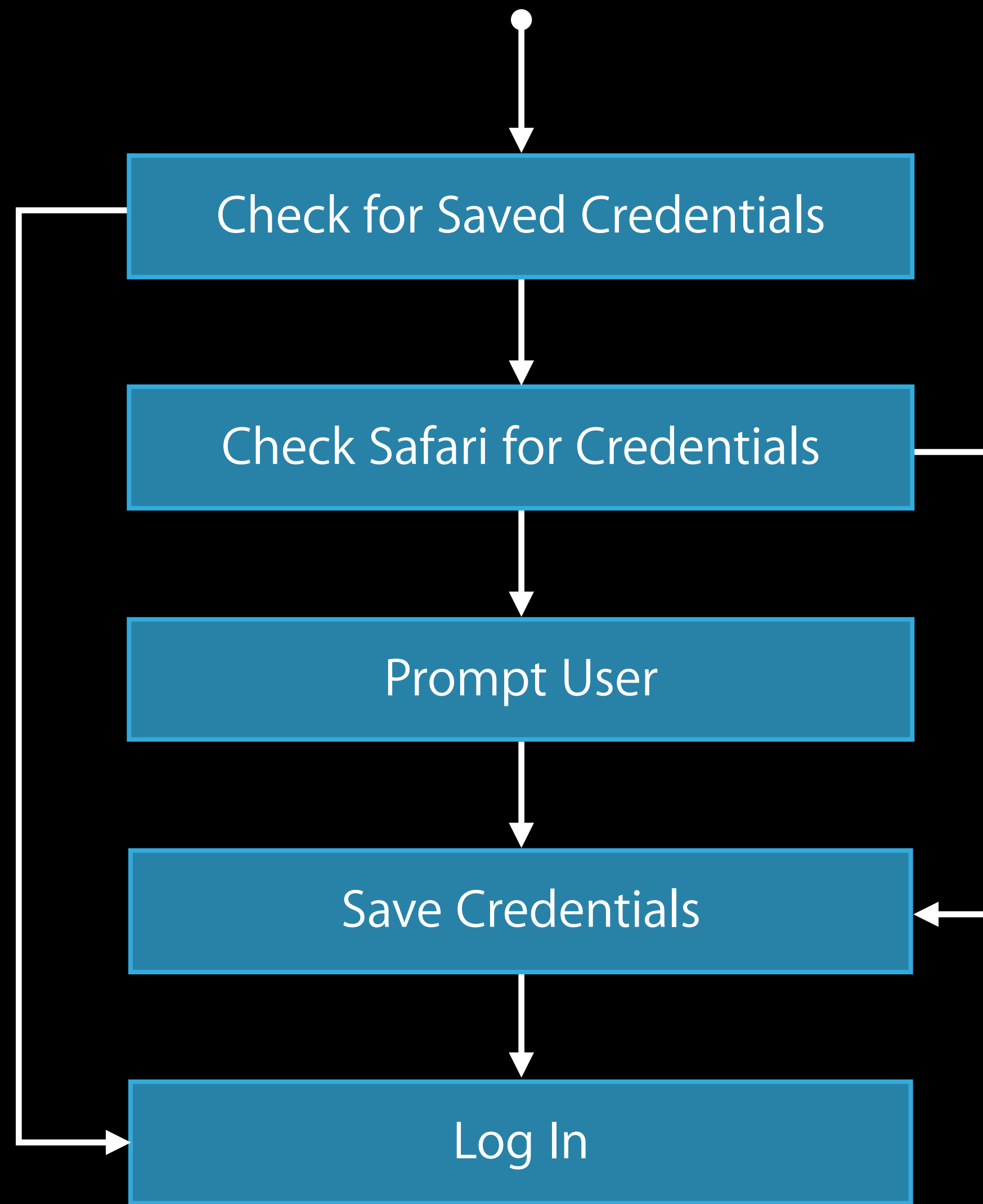




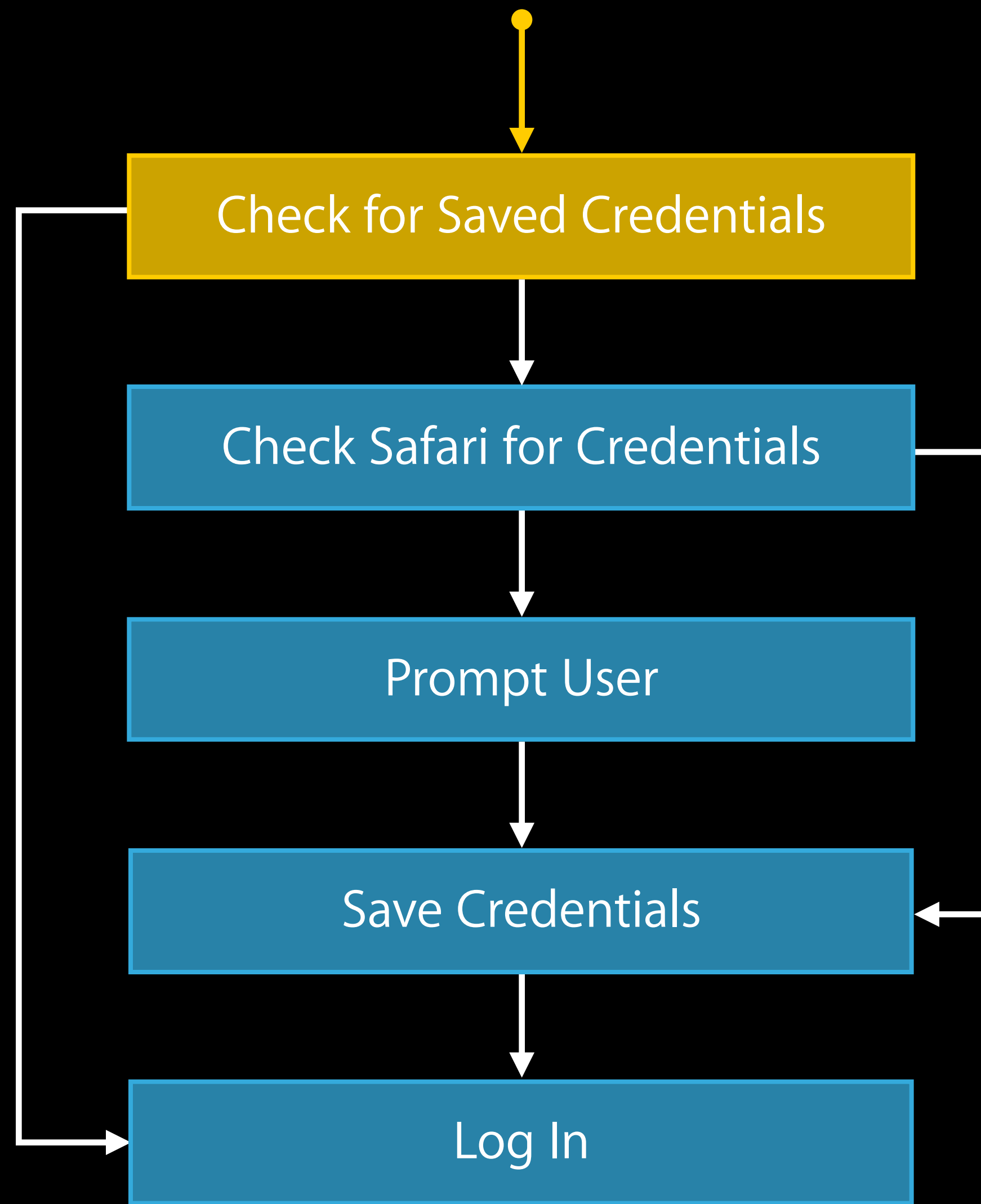
# Native Apps with Accounts



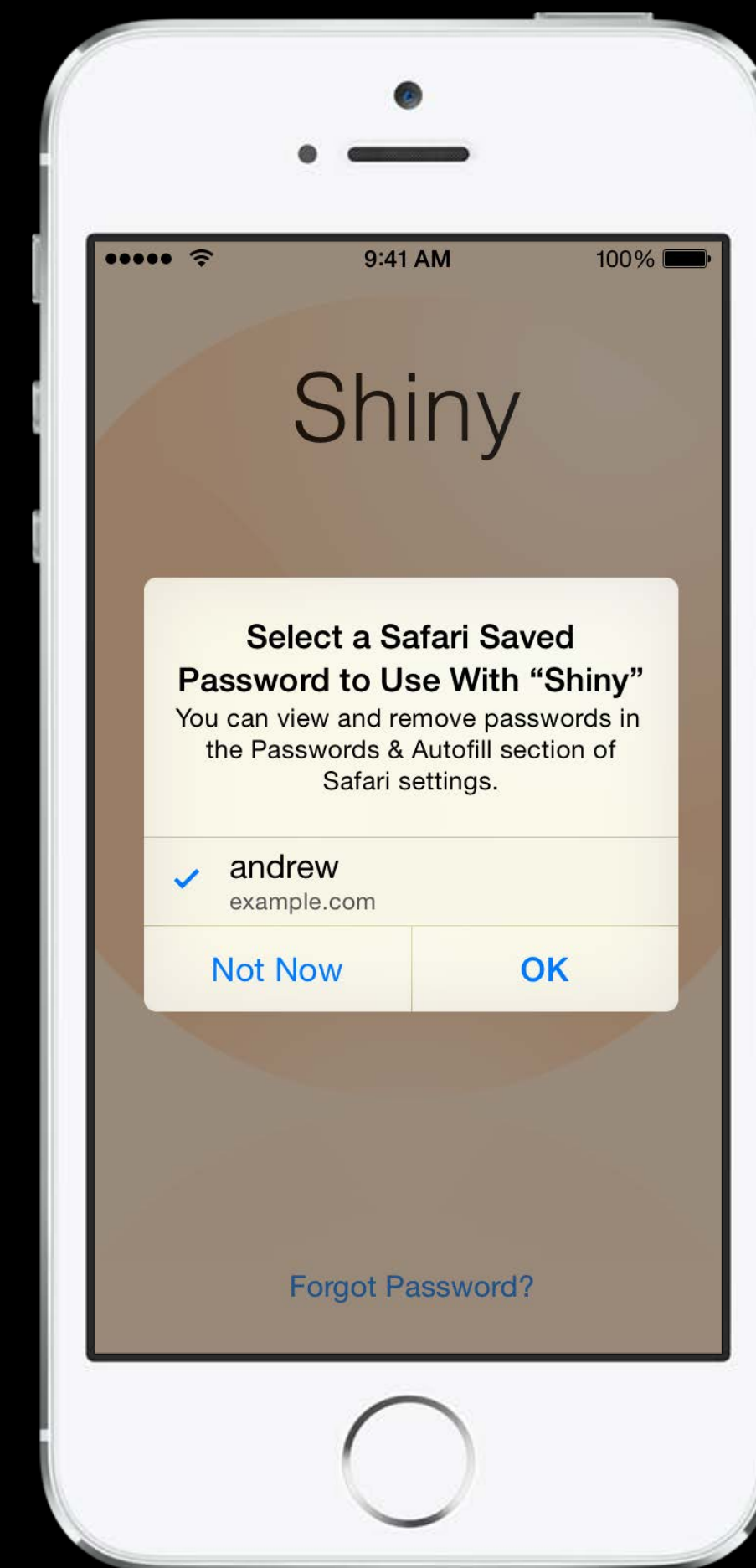
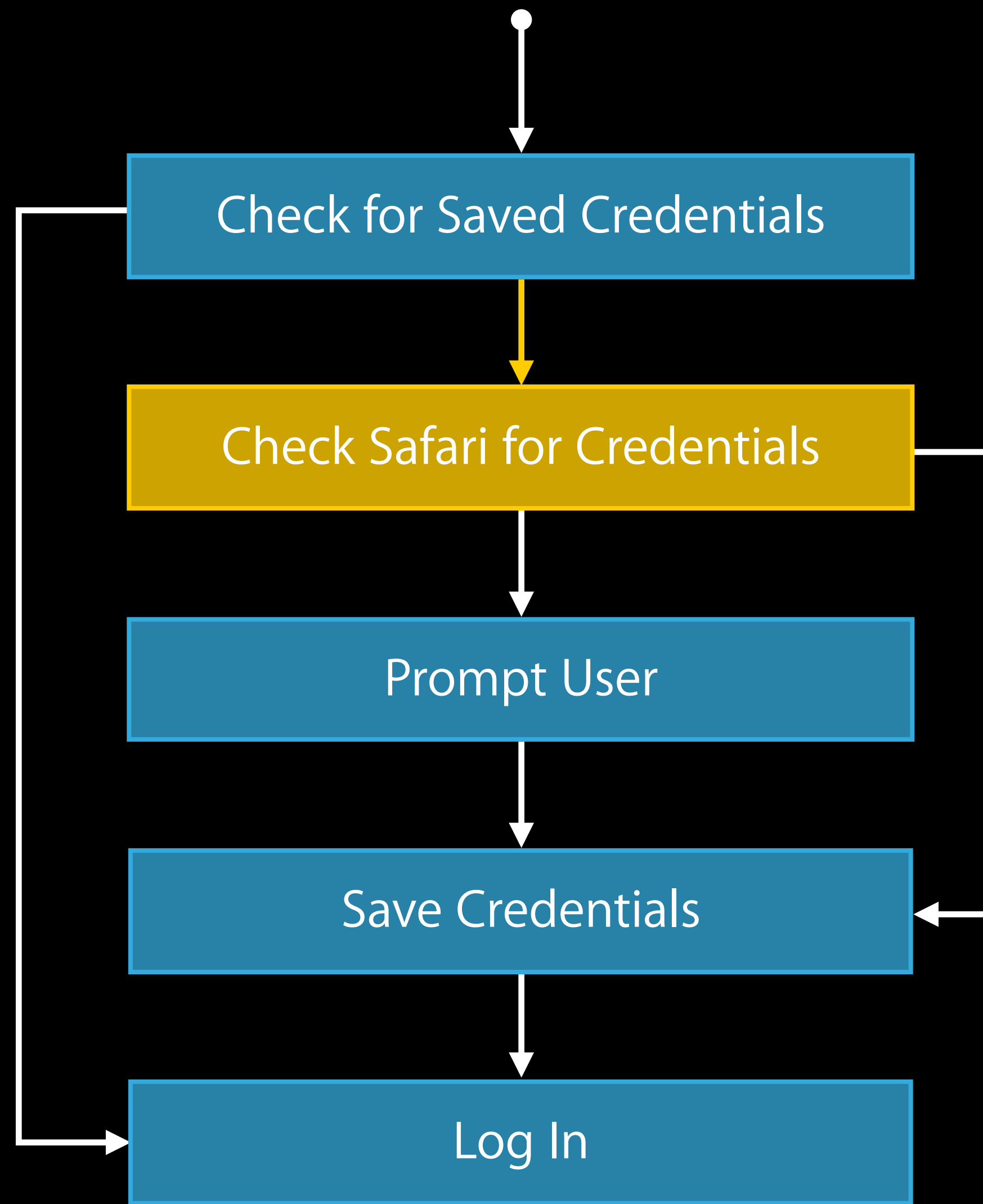
# Native Apps with Accounts



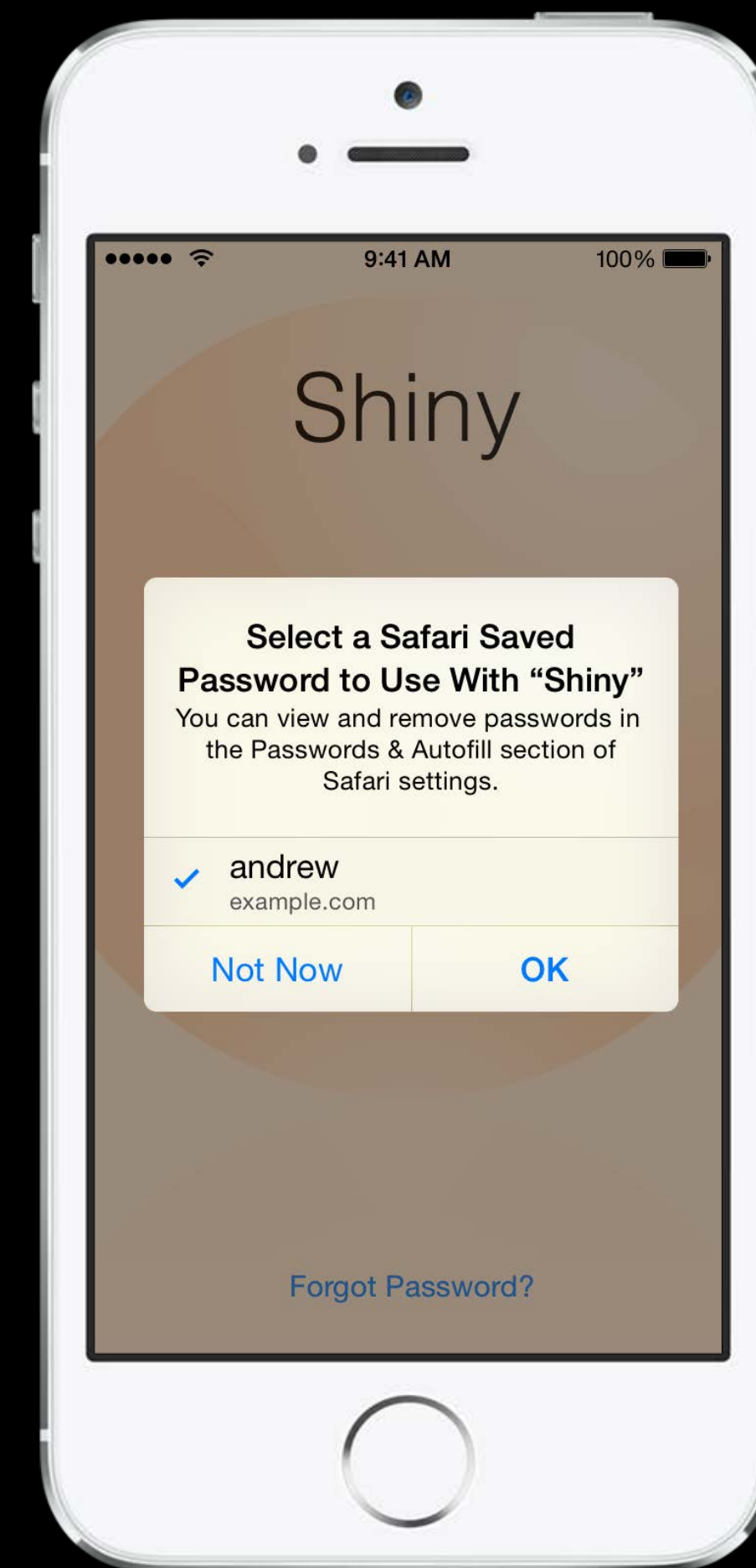
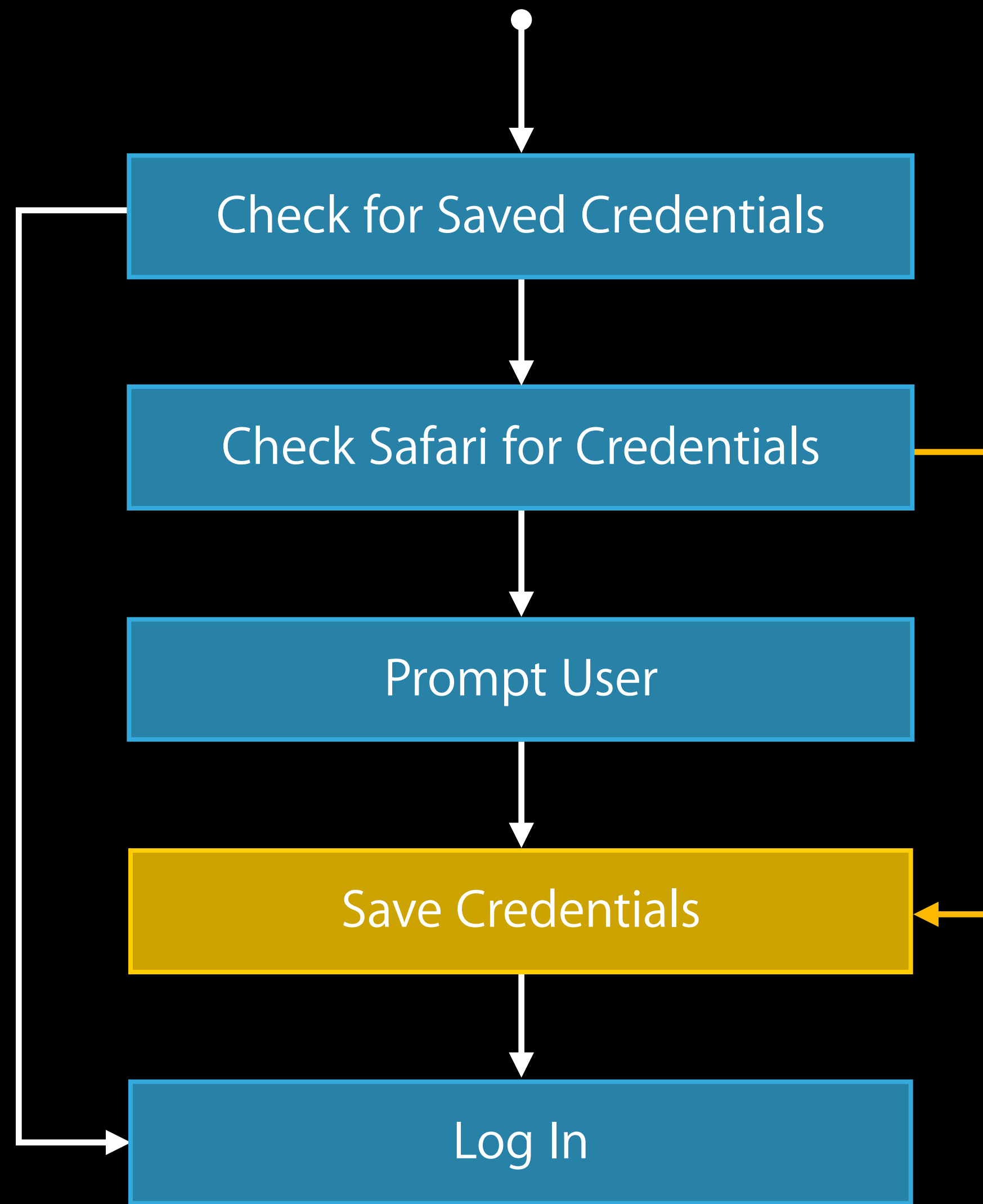
# Native Apps with Accounts



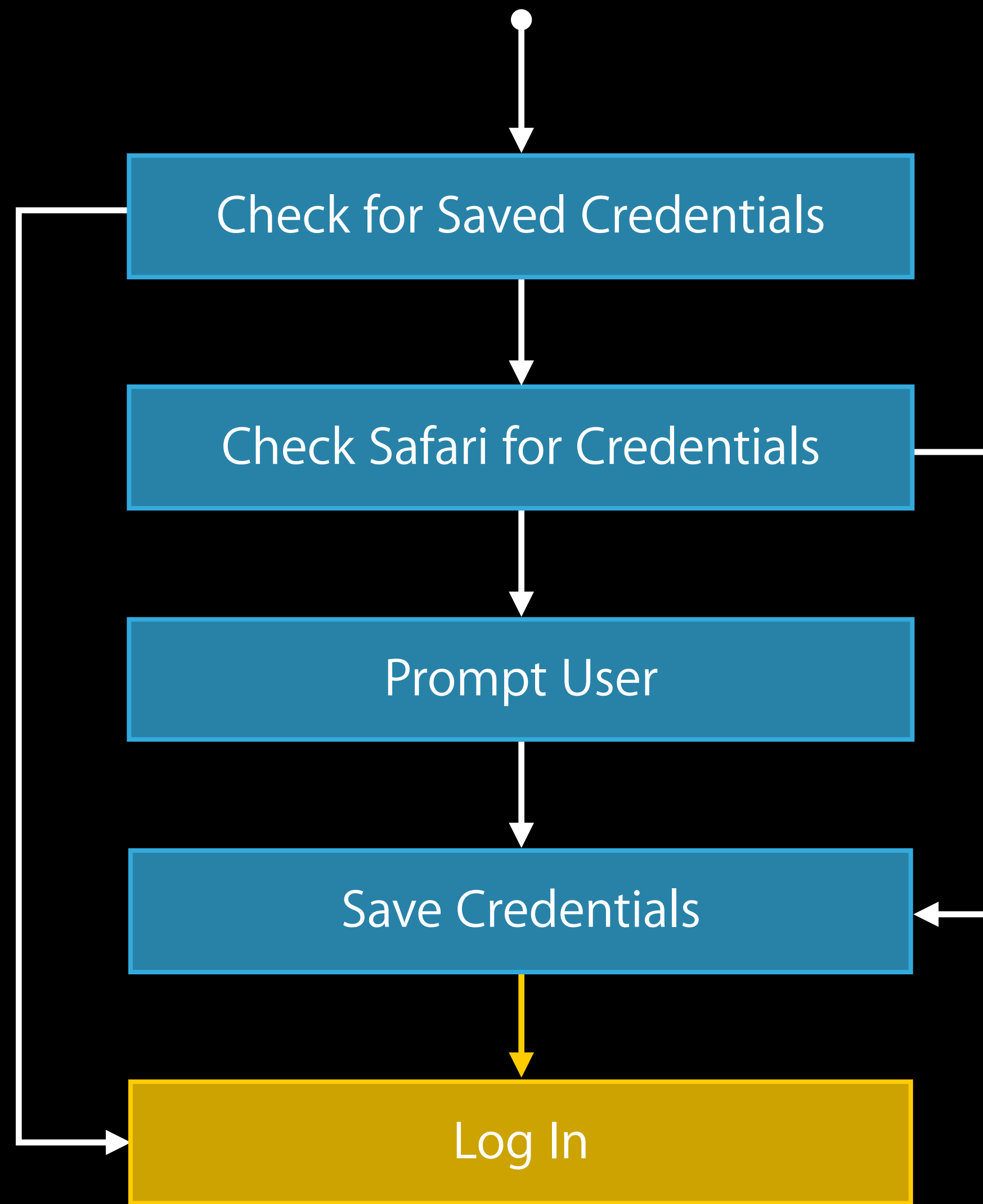
# Native Apps with Accounts



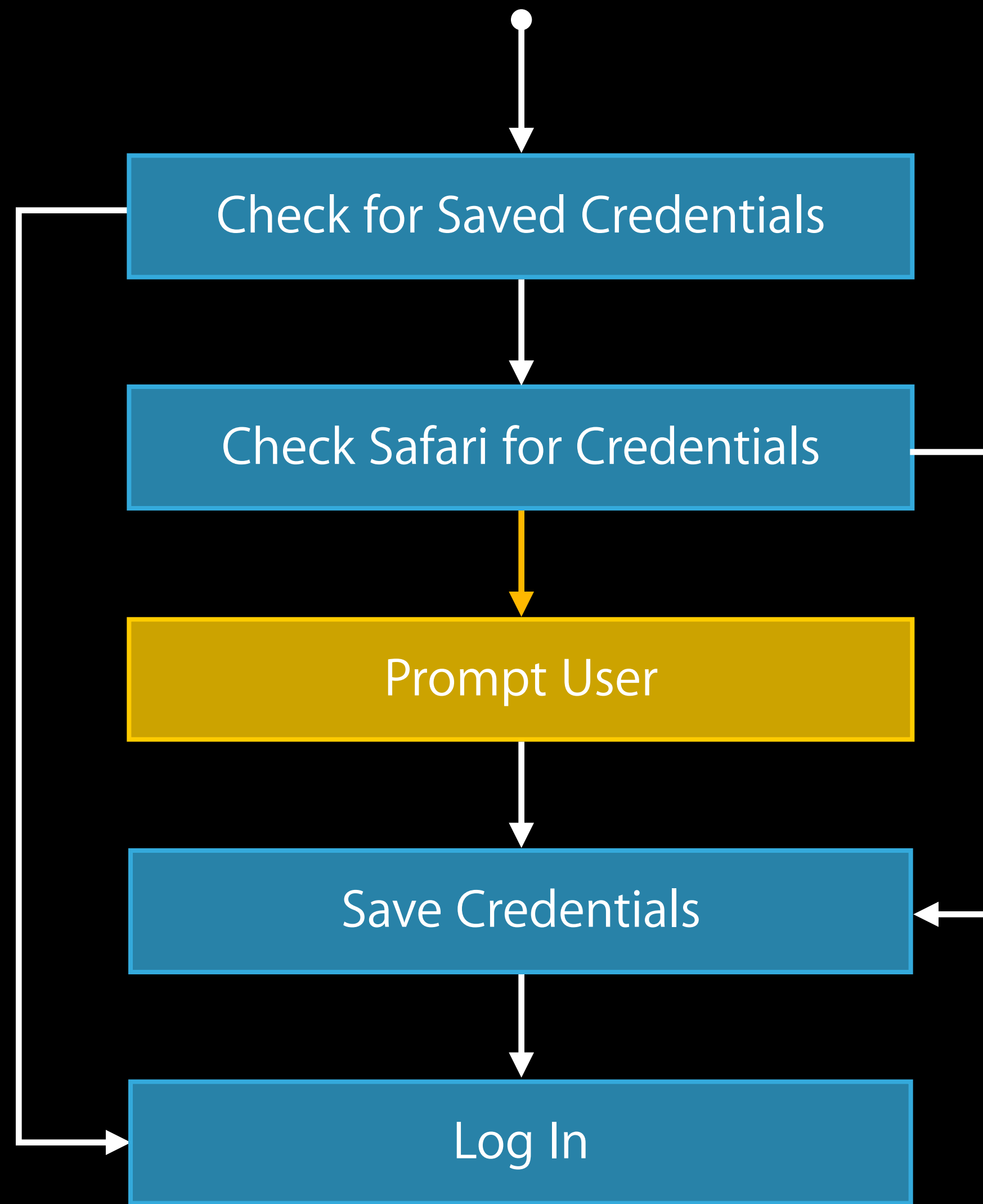
# Native Apps with Accounts



# Native Apps with Accounts



# Native Apps with Accounts



# New APIs



# New APIs

SecRequestSharedWebCredential

SecAddSharedWebCredential

SecCreateSharedWebCredentialPassword

# New APIs

SecRequestSharedWebCredential

SecAddSharedWebCredential

SecCreateSharedWebCredentialPassword

# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(CFSTR("example.com"),  
                              CFSTR("lordsecurity@icloud.com"),  
                              ^(CFArrayRef credentials, CFErrorRef error) {  
  
                              });
```

# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(CFSTR("example.com"),  
                              CFSTR("lordsecurity@icloud.com"),  
                              ^(CFArrayRef credentials, CFErrorRef error) {  
  
                              });
```

# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(CFSTR("example.com"),  
                              CFSTR("lordsecurity@icloud.com"),  
                              ^(CFArrayRef credentials, CFErrorRef error) {  
  
                              });
```

# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(CFSTR("example.com"),  
                              CFSTR("lordsecurity@icloud.com"),  
                              ^(CFArrayRef credentials, CFErrorRef error) {  
  
                              });
```

# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(NULL,  
                               NULL,  
                               ^(CFArrayRef credentials, CFErrorRef error) {  
  
                               });
```

# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(NULL, // use domains from entitlement
                               NULL,
                               ^(CFArrayRef credentials, CFErrorRef error) {
                                   });
```



# Obtaining Safari Credentials

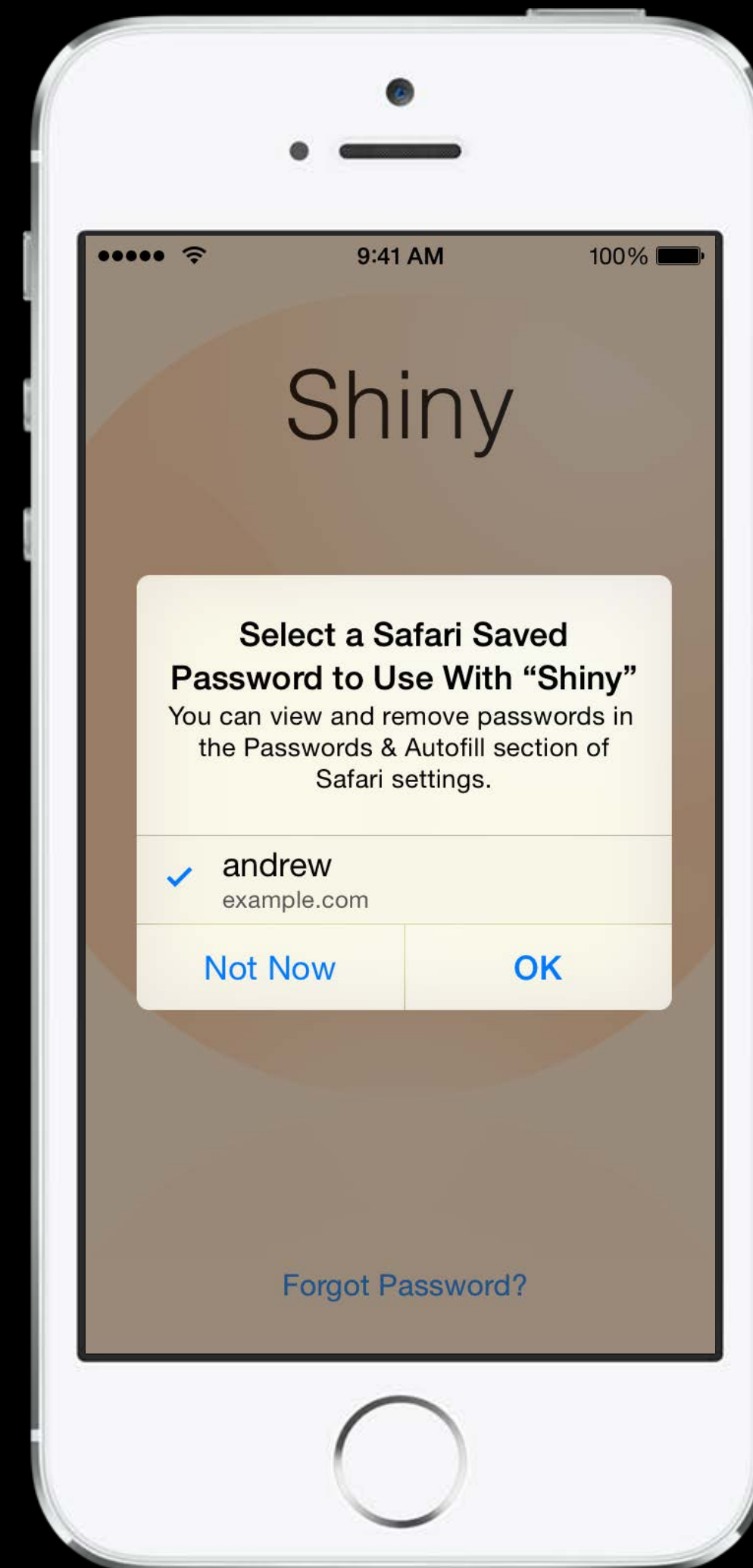
```
SecRequestSharedWebCredential(NULL, // use domains from entitlement
                               NULL, // find all matching usernames
                               ^(CFArrayRef credentials, CFErrorRef error) {
                                   });
```

# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(NULL, // use domains from entitlement
                               NULL, // find all matching usernames
                               ^(CFArrayRef credentials, CFErrorRef error) {
                                   });
```



# Obtaining Safari Credentials



# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(NULL, // use domains from entitlement
                               NULL, // find all matching usernames
                               ^(CFArrayRef credentials, CFErrorRef error) {
                                   });
```









# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(NULL, // use domains from entitlement
                              NULL, // find all matching usernames
                              ^(CFArrayRef credentials, CFErrorRef error) {

    if (!error && CFArrayGetCount(credentials)) {
        CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
        NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
        NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
        dispatch_async(dispatch_get_main_queue(), ^{
            [serverManager attemptLoginWithUsername:username password:password];
        });
    } else {
        dispatch_async(dispatch_get_main_queue(), ^{
            [self showLoginUI];
        });
    }
});
```

# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(NULL, // use domains from entitlement
                              NULL, // find all matching usernames
                              ^(CFArrayRef credentials, CFErrorRef error) {

    if (!error && CFArrayGetCount(credentials)) {
        CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
        NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
        NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
        dispatch_async(dispatch_get_main_queue(), ^{
            [serverManager attemptLoginWithUsername:username password:password];
        });
    } else {
        dispatch_async(dispatch_get_main_queue(), ^{
            [self showLoginUI];
        });
    }
});
```

# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(NULL, // use domains from entitlement
                              NULL, // find all matching usernames
                              ^(CFArrayRef credentials, CFErrorRef error) {

    if (!error && CFArrayGetCount(credentials)) {
        CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
        NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
        NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
        dispatch_async(dispatch_get_main_queue(), ^{
            [serverManager attemptLoginWithUsername:username password:password];
        });
    } else {
        dispatch_async(dispatch_get_main_queue(), ^{
            [self showLoginUI];
        });
    }
});
```

# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(NULL, // use domains from entitlement
                              NULL, // find all matching usernames
                              ^(CFArrayRef credentials, CFErrorRef error) {

    if (!error && CFArrayGetCount(credentials)) {
        CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
        NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
        NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
        dispatch_async(dispatch_get_main_queue(), ^{
            [serverManager attemptLoginWithUsername:username password:password];
        });
    } else {
        dispatch_async(dispatch_get_main_queue(), ^{
            [self showLoginUI];
        });
    }
});
```

# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(NULL, // use domains from entitlement
                              NULL, // find all matching usernames
                              ^(CFArrayRef credentials, CFErrorRef error) {

    if (!error && CFArrayGetCount(credentials)) {
        CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
        NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
        NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
        dispatch_async(dispatch_get_main_queue(), ^{
            [serverManager attemptLoginWithUsername:username password:password];
        });
    } else {
        dispatch_async(dispatch_get_main_queue(), ^{
            [self showLoginUI];
        });
    }
});
```

# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(NULL, // use domains from entitlement
                              NULL, // find all matching usernames
                              ^(CFArrayRef credentials, CFErrorRef error) {

    if (!error && CFArrayGetCount(credentials)) {
        CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
        NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
        NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
        dispatch_async(dispatch_get_main_queue(), ^{
            [serverManager attemptLoginWithUsername:username password:password];
        });
    } else {
        dispatch_async(dispatch_get_main_queue(), ^{
            [self showLoginUI];
        });
    }
});
```

# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(NULL, // use domains from entitlement
                              NULL, // find all matching usernames
                              ^(CFArrayRef credentials, CFErrorRef error) {

    if (!error && CFArrayGetCount(credentials)) {
        CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
        NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
        NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
        dispatch_async(dispatch_get_main_queue(), ^{
            [serverManager attemptLoginWithUsername:username password:password];
        });
    } else {
        dispatch_async(dispatch_get_main_queue(), ^{
            [self showLoginUI];
        });
    }
});
```

# Obtaining Safari Credentials

```
SecRequestSharedWebCredential(NULL, // use domains from entitlement
                              NULL, // find all matching usernames
                              ^(CFArrayRef credentials, CFErrorRef error) {

    if (!error && CFArrayGetCount(credentials)) {
        CFDictionaryRef credential = CFArrayGetValueAtIndex(credentials, 0);
        NSString *username = CFDictionaryGetValue(credential, kSecAttrAccount);
        NSString *password = CFDictionaryGetValue(credential, kSecSharedPassword);
        dispatch_async(dispatch_get_main_queue(), ^{
            [serverManager attemptLoginWithUsername:username password:password];
        });
    } else {
        dispatch_async(dispatch_get_main_queue(), ^{
            [self showLoginUI];
        });
    }
});
```



# Telling Safari About Credentials

# Telling Safari About Credentials

```
SecAddSharedWebCredential(CFSTR("example.com"),  
                           username,  
                           password,  
                           ^(CFErrorRef error) {  
                               NSLog(@"%@", error);  
                           });
```

# Telling Safari About Credentials

```
SecAddSharedWebCredential(CFSTR("example.com"),
                           username,
                           password,
                           ^(CFErrorRef error) {
                               NSLog(@"%@", error);
                           });
```

# Telling Safari About Credentials

```
SecAddSharedWebCredential(CFSTR("example.com"),
    username,
    password,
    ^(CFErrorRef error) {
        NSLog(@"%@", error);
    });
```

# Telling Safari About Credentials

```
SecAddSharedWebCredential(CFSTR("example.com"),
    username,
    password,
    ^(CFErrorRef error) {
        NSLog(@"%@", error);
    });
```

# Telling Safari About Credentials

```
SecAddSharedWebCredential(CFSTR("example.com"),
                          username,
                          password,
                          ^(CFErrorRef error) {
                              NSLog(@"%@", error);
                          });
```

# Telling Safari About Credentials

```
SecAddSharedWebCredential(CFSTR("example.com"),
                           username,
                           password,
                           ^(CFErrorRef error) {
                               NSLog(@"%@", error);
                           });
```

# Telling Safari About Credentials

## Updating a Credential

```
SecAddSharedWebCredential(CFSTR("example.com"),
                           username,
                           newPassword,
                           ^(CFErrorRef error) {
                               NSLog(@"%@", error);
                           });
```



# Telling Safari About Credentials

## Updating a Credential

```
SecAddSharedWebCredential(CFSTR("example.com"),  
                           username,  
                           newPassword,  
                           ^(CFErrorRef error) {  
                               NSLog(@"%@", error);  
                           });
```



# Telling Safari About Credentials

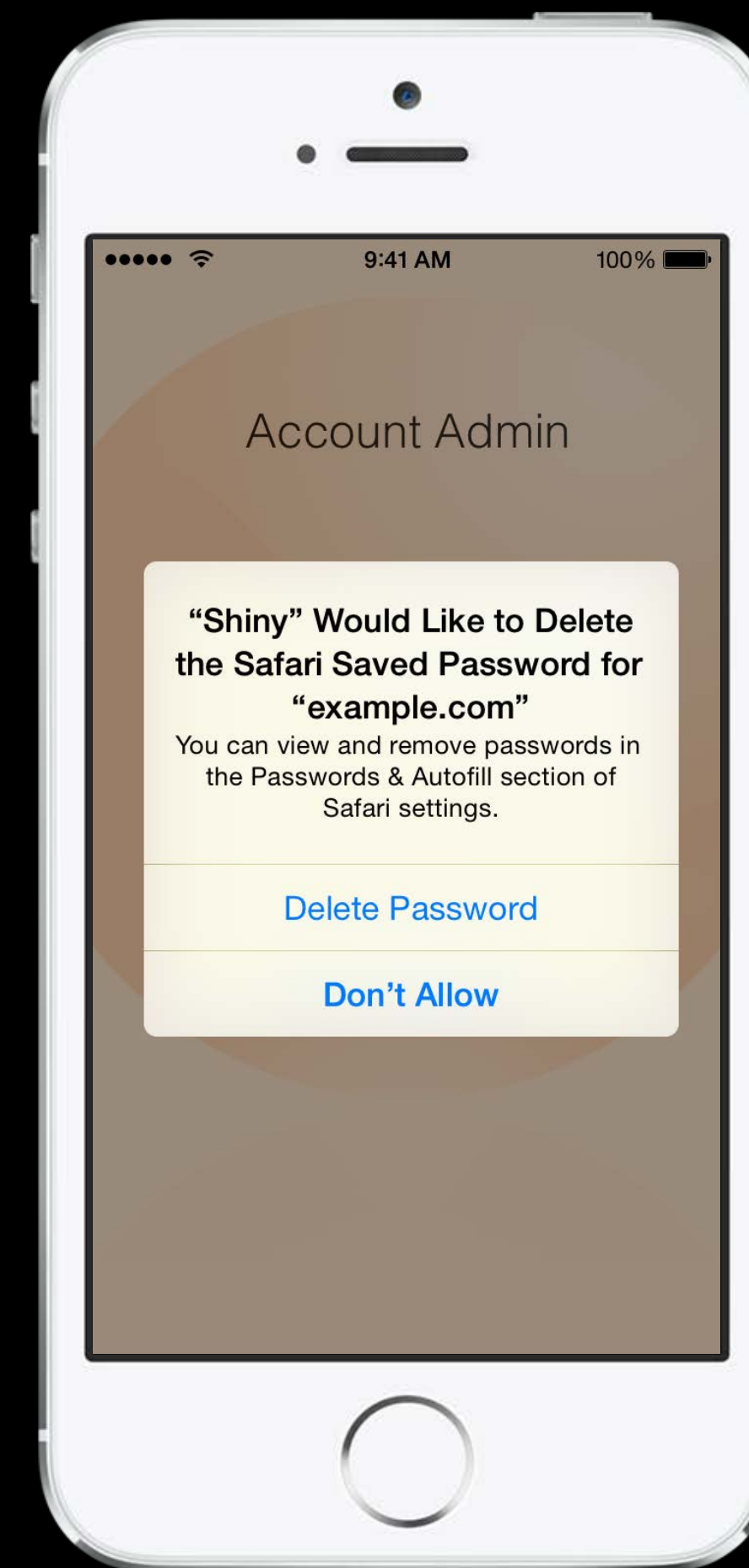
## Deleting a Credential

```
SecAddSharedWebCredential(CFSTR("example.com"),
                          username,
                          NULL,
                          ^(CFErrorRef error) {
                              NSLog(@"%@", error);
                          });
```

# Telling Safari About Credentials

## Deleting a Credential

```
SecAddSharedWebCredential(CFSTR("example.com"),  
                           username,  
                           NULL,  
                           ^(CFErrorRef error) {  
                               NSLog(@"%@", error);  
                           });
```



# Generating a Password

```
CFStringRef suggestedPwd = SecCreateSharedWebCredentialPassword();
```

# Generating a Password

```
CFStringRef suggestedPwd = SecCreateSharedWebCredentialPassword();
```

JVA-9cx-ZfT-c4H

k6S-kgS-c7E-hhK

2H4-mVC-dx3-Vbv

9uB-VC9-52L-9H5

meR-7CK-d3B-em9

mMJ-8LN-Wf4-Jsb

wu4-PGe-gEQ-xTE

645-3bY-7CG-3mn

5B8-T8E-rRG-EtB

ESB-naN-PrF-s3r

wN4-NUq-dHB-oU9

Z7k-n3r-nqB-oNP

T6h-7tS-65L-011f

# Error Conditions

# Error Conditions

Condition

CFErrorGetCode()

CFErrorCopyDescription()

---

# Error Conditions

Condition	CFErrorGetCode()	CFErrorCopyDescription()
Missing entitlement	errSecParam	"No domain provided"



# Error Conditions

Condition	CFErrorGetCode()	CFErrorCopyDescription()
Missing entitlement	errSecParam	"No domain provided"
Missing entitlement	errSecMissingEntitlement	"example.com not found in com.apple.developer.associated-domains entitlement"

# Error Conditions

Condition	CFErrorGetCode()	CFErrorCopyDescription()
Missing entitlement	errSecParam	"No domain provided"
Missing entitlement	errSecMissingEntitlement	"example.com not found in com.apple.developer.associated-domains entitlement"
Domain not approved	errSecItemNotFound	"no matching items found"

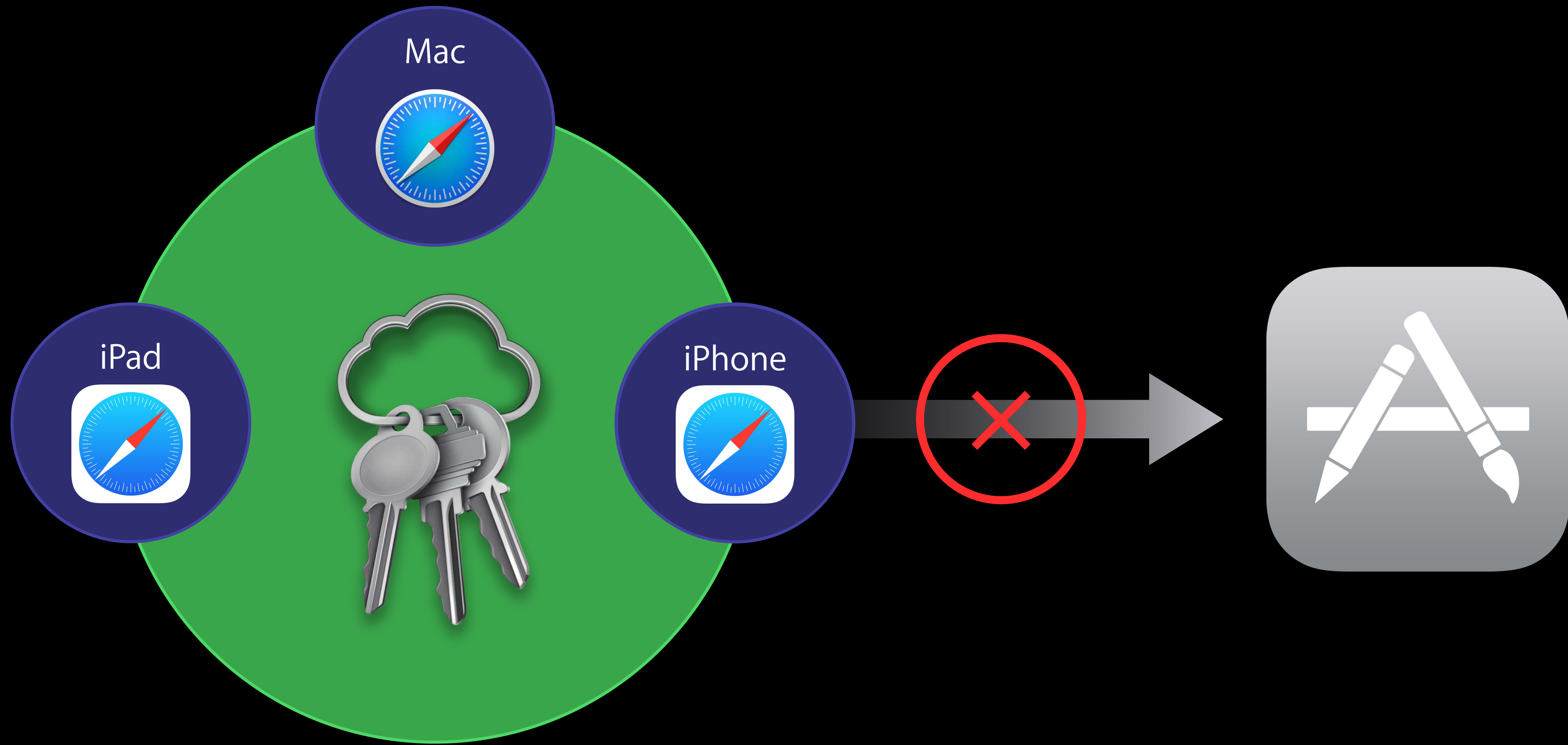
# Error Conditions

Condition	CFErrorGetCode()	CFErrorCopyDescription()
Missing entitlement	errSecParam	"No domain provided"
Missing entitlement	errSecMissingEntitlement	"example.com not found in com.apple.developer.associated-domains entitlement"
Domain not approved	errSecItemNotFound	"no matching items found"
No saved credentials	errSecItemNotFound	"no matching items found"

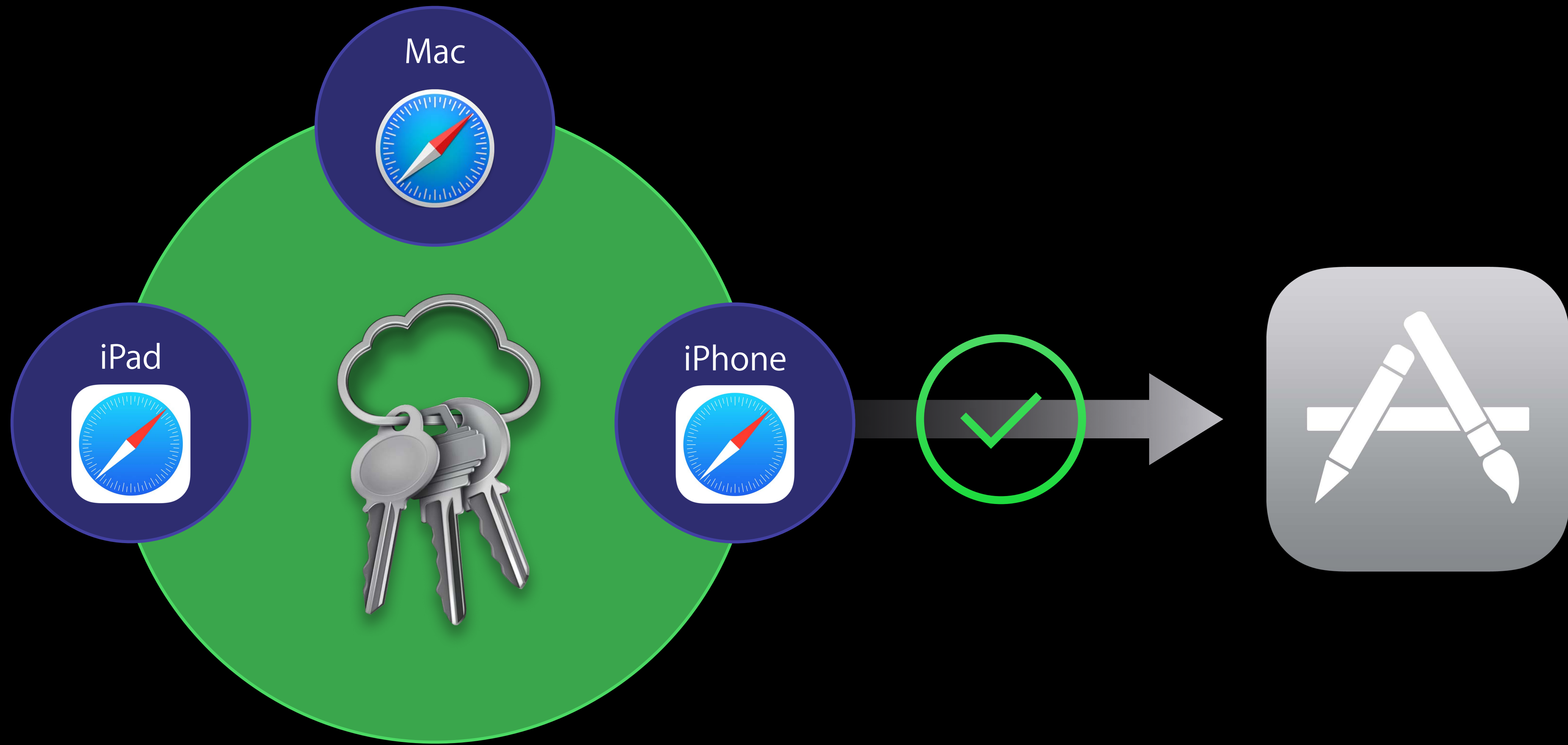
*Demo*

Putting it into practice

# Safari Credentials in Native Apps



# Safari Credentials in Native Apps



# Credentials on the Web

Ricky Mondello

Safari and WebKit Engineer

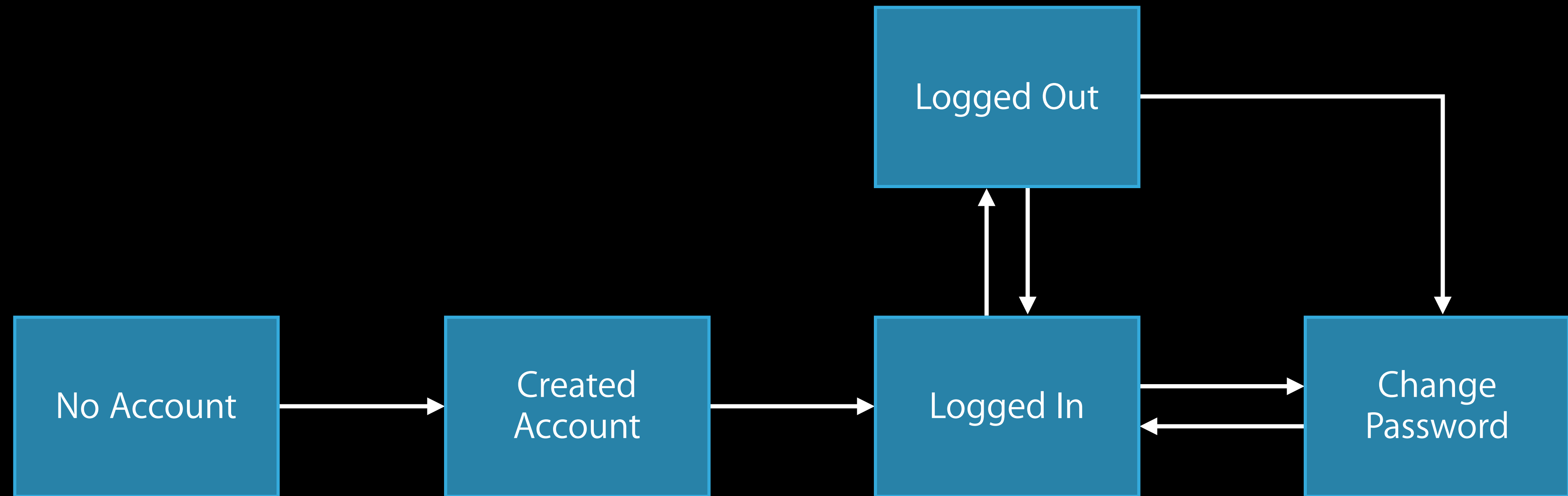




Expressing Intent

# Expressing Intent on the Web

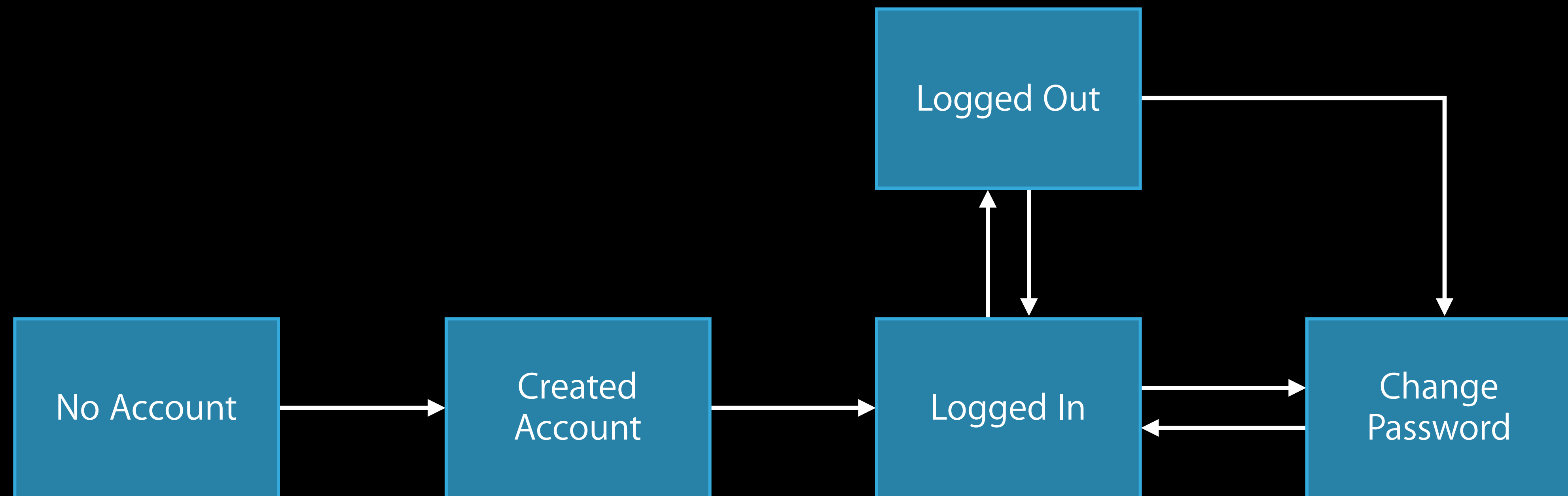
# The Password Lifecycle



# Test Your Website with Safari AutoFill

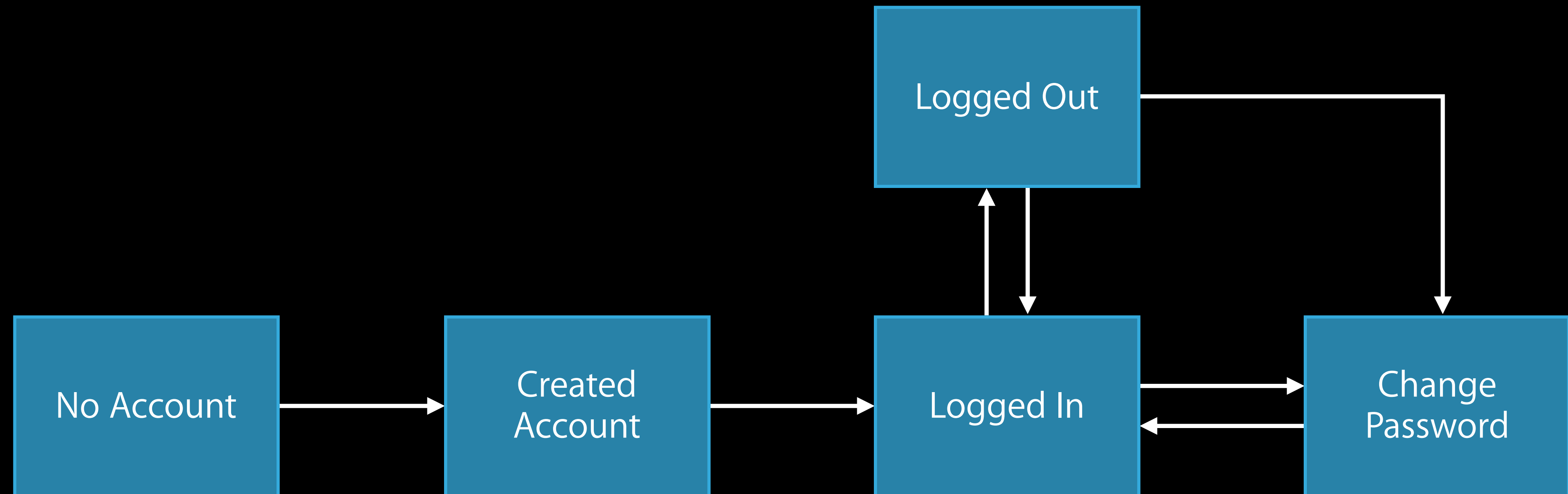
# Testing Your Site and Safari AutoFill

The password lifecycle



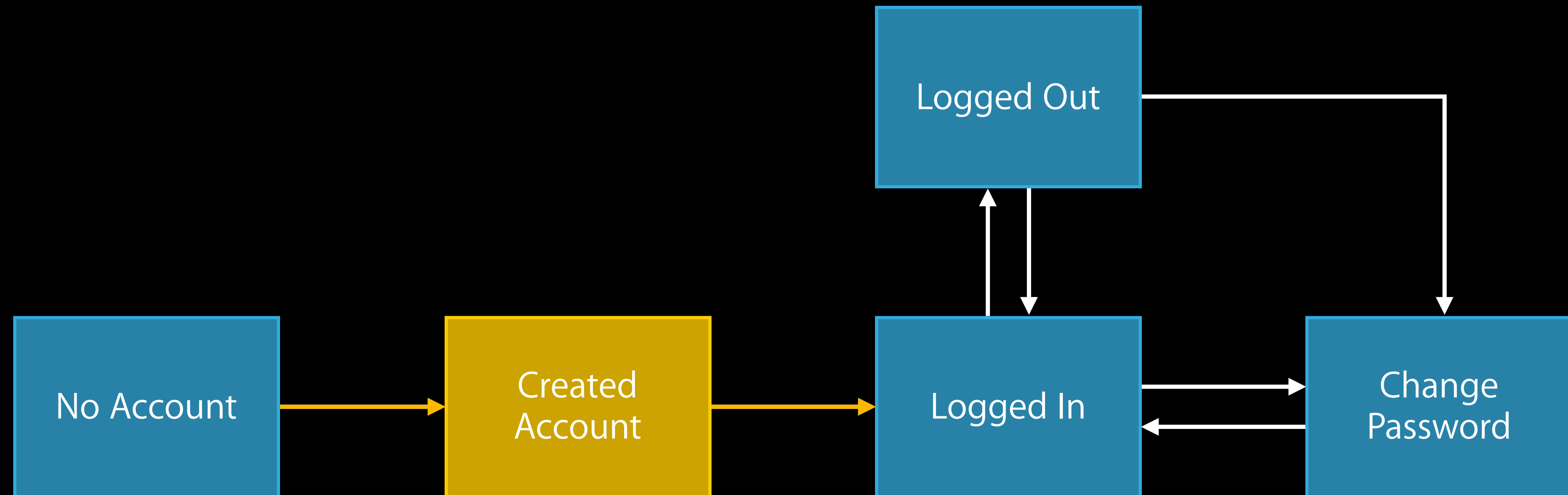
# Testing Your Site and Safari AutoFill

Creating an account



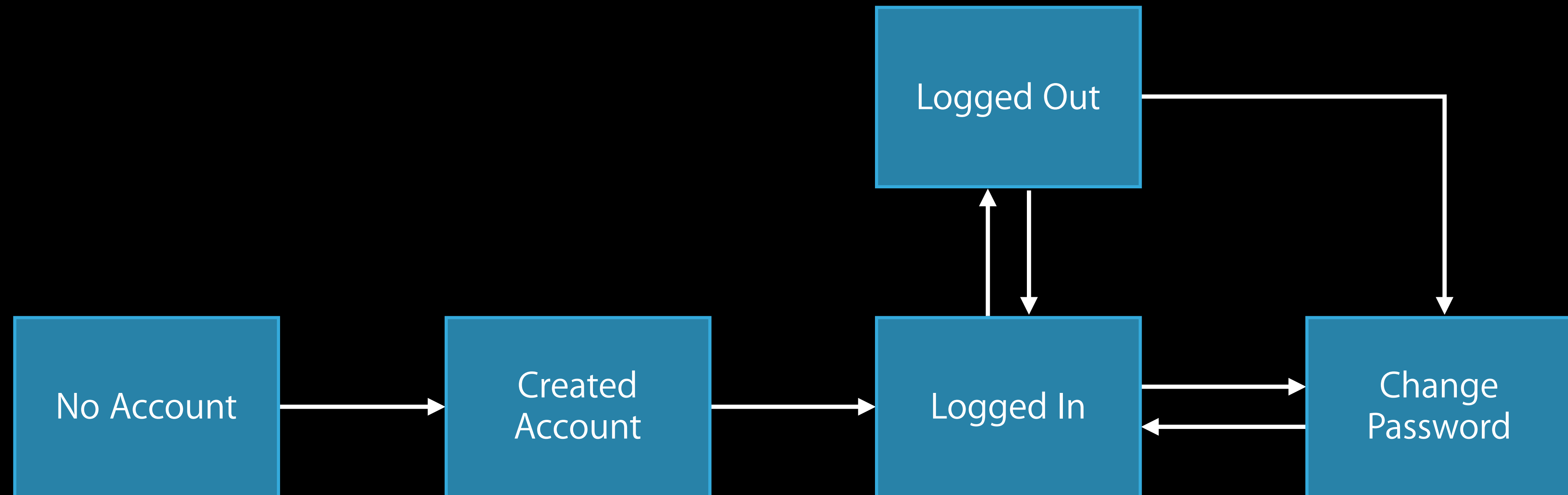
# Testing Your Site and Safari AutoFill

Creating an account



# Testing Your Site and Safari AutoFill

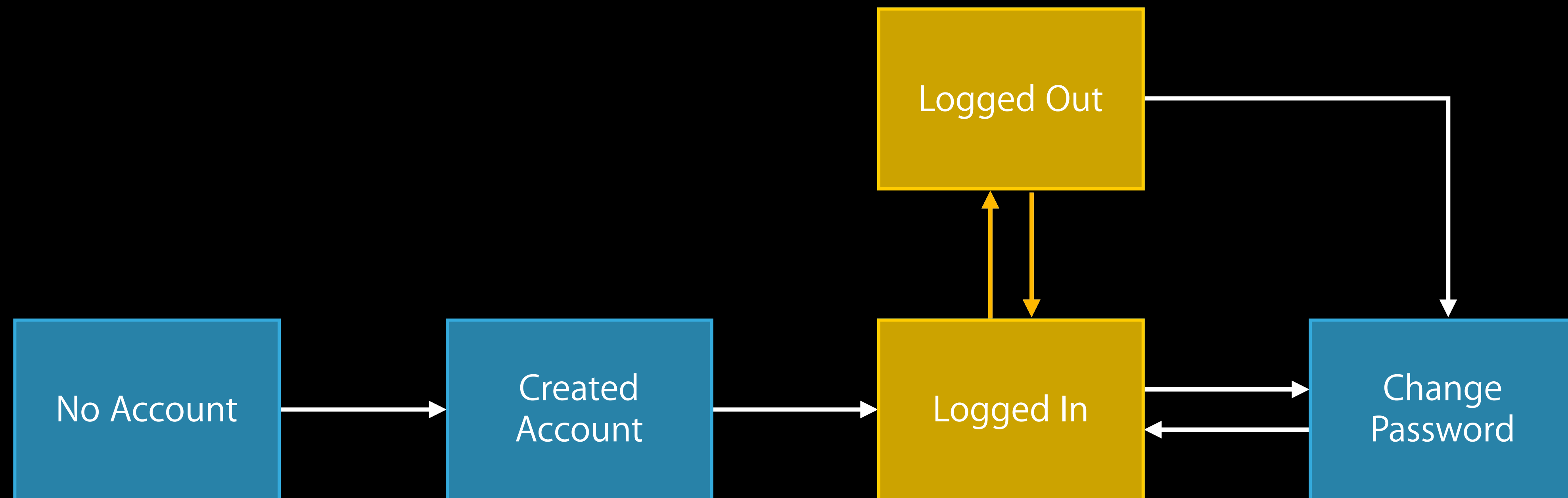
Logging in with saved credentials





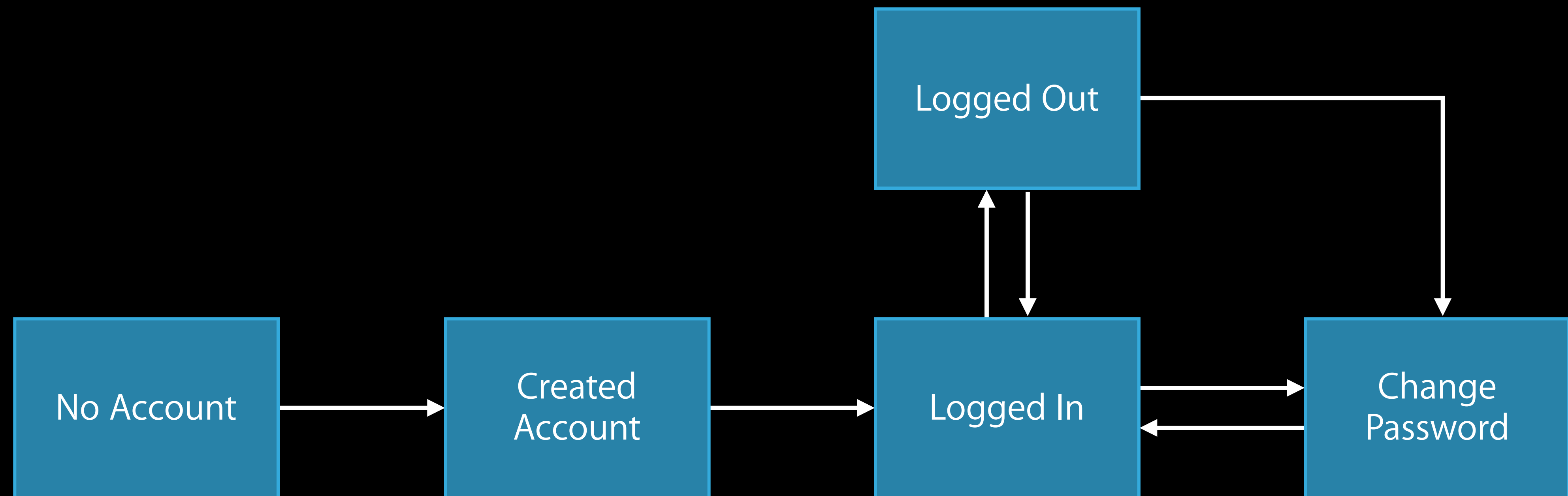
# Testing Your Site and Safari AutoFill

Logging in with saved credentials



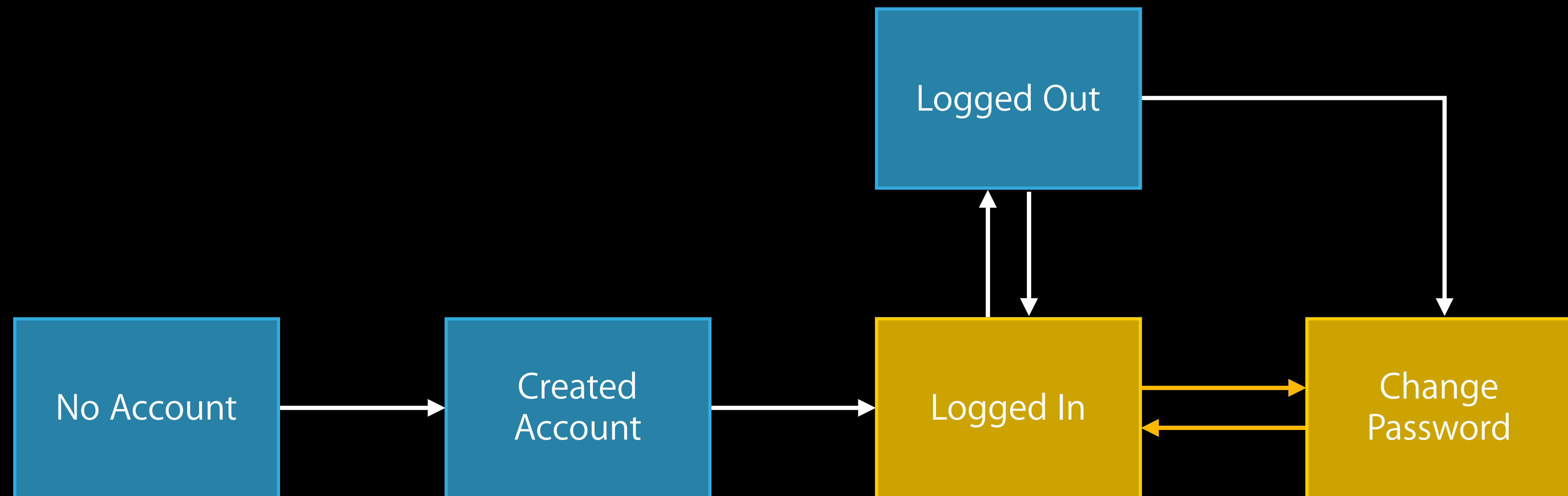
# Testing Your Site and Safari AutoFill

Changing a password



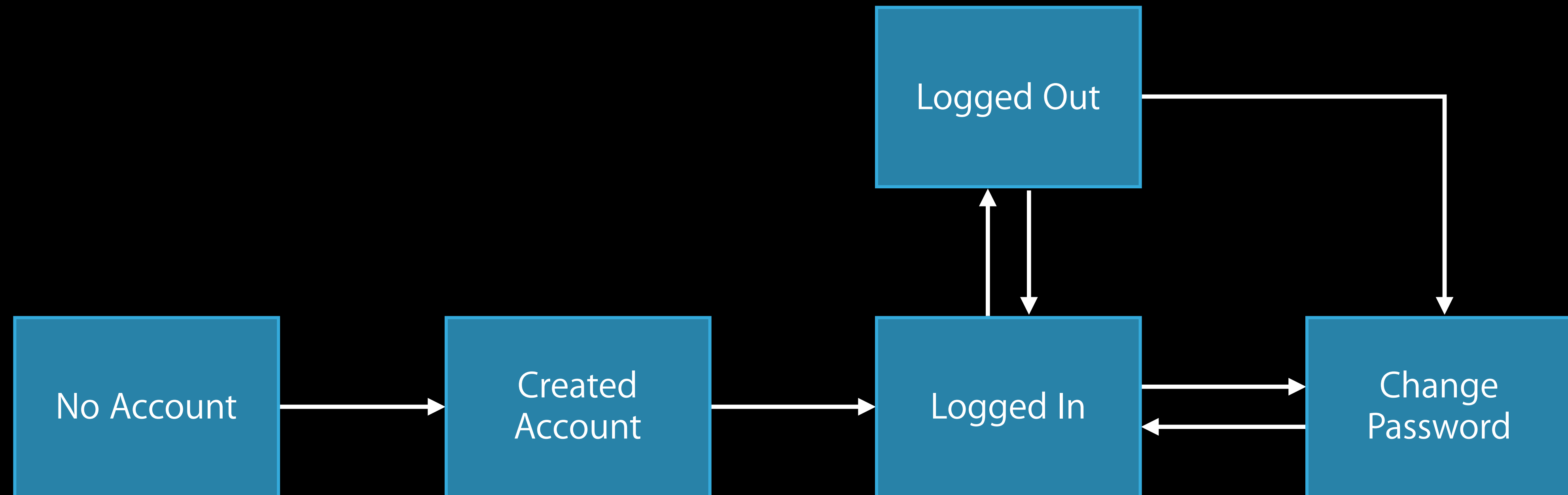
# Testing Your Site and Safari AutoFill

Changing a password



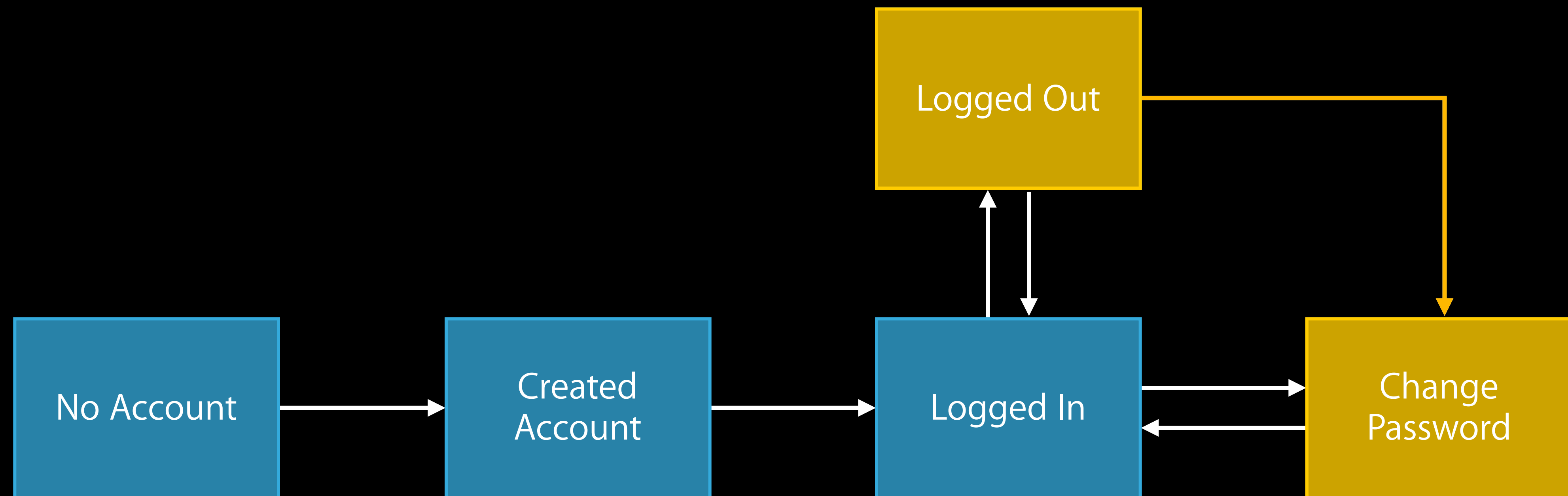
# Testing Your Site and Safari AutoFill

Resetting a password



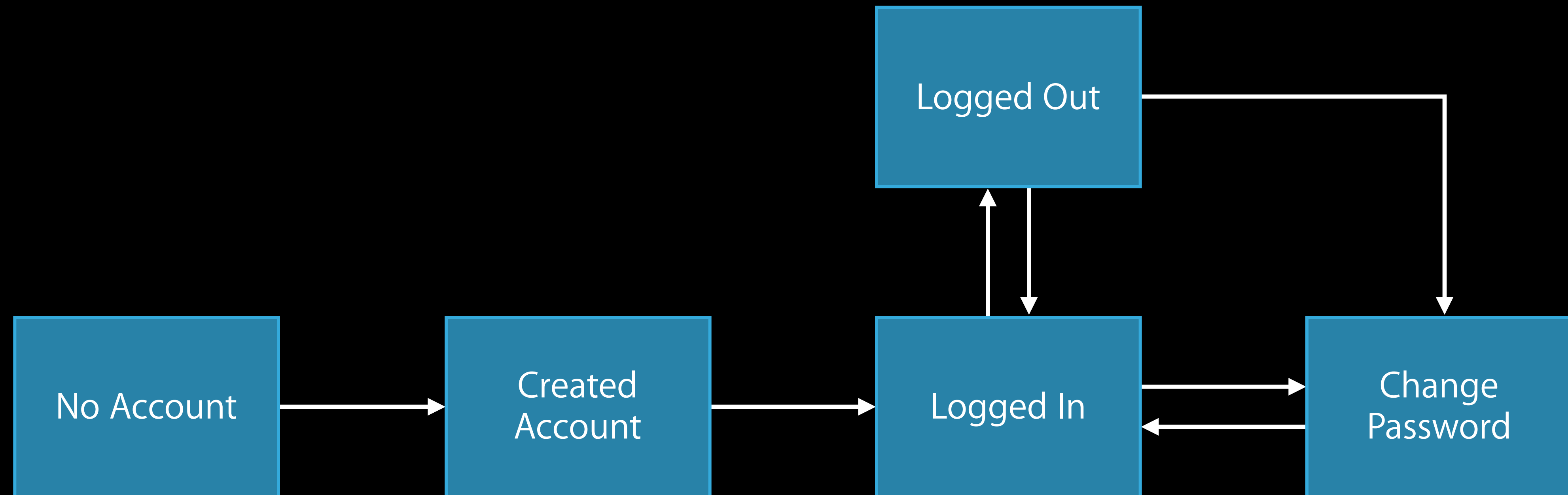
# Testing Your Site and Safari AutoFill

Resetting a password



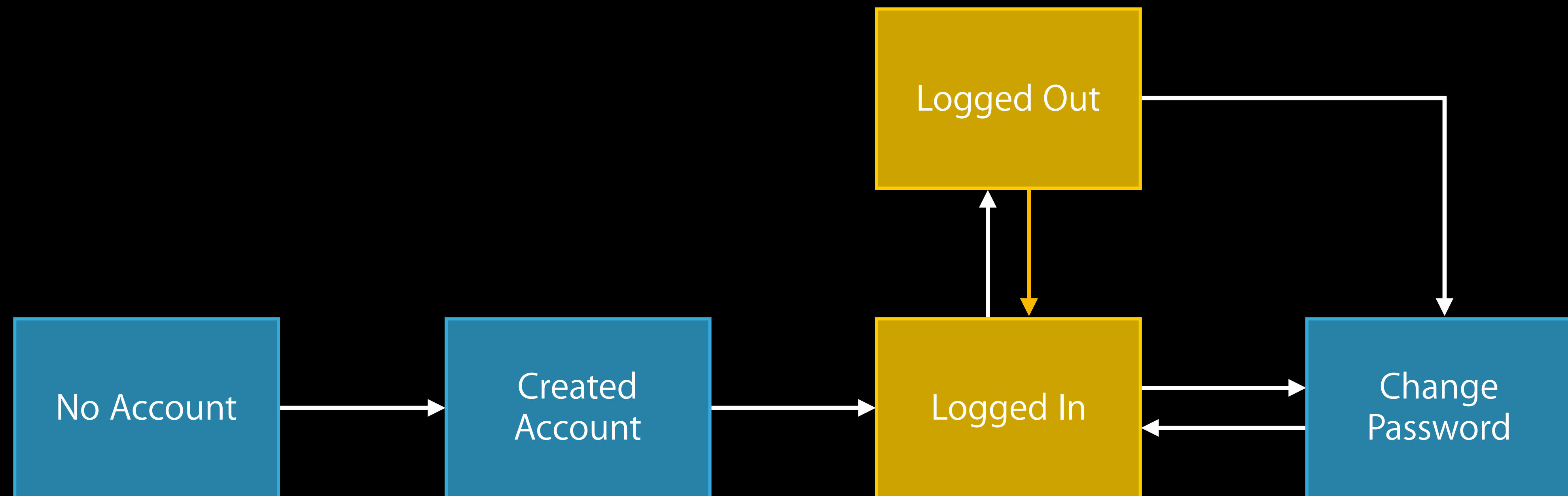
# Testing Your Site and Safari AutoFill

Logging in for the first time



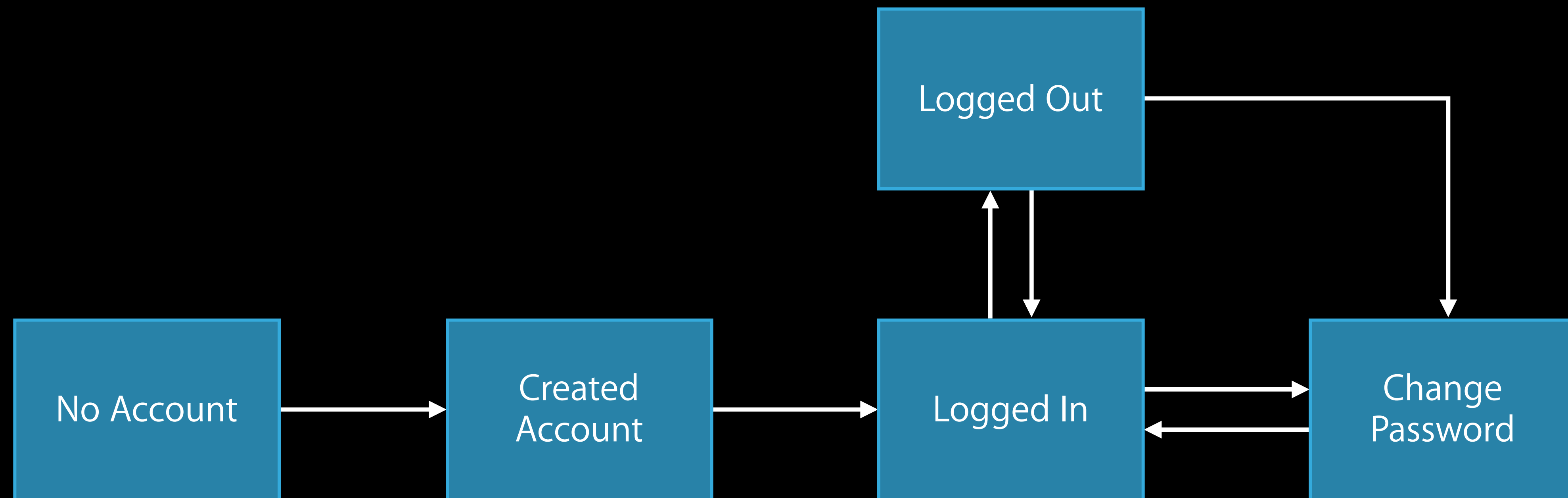
# Testing Your Site and Safari AutoFill

Logging in for the first time



# Testing Your Site and Safari AutoFill

What to do if something doesn't work





# Your Forms' Intent

# Your Forms' Intent

New values for the autocomplete attribute



```
<input type="text" autocomplete="username">
```

```
<input type="password" autocomplete="current-password">
```

```
<input type="password" autocomplete="new-password">
```

# Logging in



```
<form id="login" action="login.php"
  method="post">

  <input type="text"
    autocomplete="username">

  <input type="password"
    autocomplete="current-password">

  <input type="submit" value="Log in">
</form>
```

Username

Password

# Signing up



```
<form id="signup" action="signup.php"
  method="post">

  <input type="text"
    autocomplete="username">

  <input type="password"
    autocomplete="new-password">

  <input type="submit" value="Sign up">
</form>
```

Username

Password

# Signing up

## With confirmation



```
<form id="signup" action="signup.php"
  method="post">

  <input type="text"
    autocomplete="username">

  <input type="password"
    autocomplete="new-password">

  <input type="password"
    autocomplete="new-password">

  <input type="submit" value="Sign up">
</form>
```

Username

Password

Confirm Password

# Changing One's Password

While logged in



```
<form id="change-password"
  action="update-user.php" method="post">
  <input type="text" autocomplete="username"
    readonly value="rmondello">
  <input type="password"
    autocomplete="current-password">
  <input type="password"
    autocomplete="new-password">
  <input type="submit"
    value="Change Password">
</form>
```

Username

rmondello

Current Password

New Password

Change Password

# Changing One's Password

While logged in



```
<form id="change-password"
  action="update-user.php" method="post">
```

```
<input type="text" autocomplete="username"
  readonly value="rmondello">
```

```
<input type="password"
  autocomplete="current-password">
```

```
<input type="password"
  autocomplete="new-password">
```

```
<input type="submit"
  value="Change Password">
```

```
</form>
```

Username

Current Password

New Password

Change Password

# Changing One's Password

While logged in



```
<form id="change-password"
  action="update-user.php" method="post">
```

```
<input type="hidden" autocomplete="username"
  value="rmondello">
```

```
<input type="password"
  autocomplete="current-password">
```

```
<input type="password"
  autocomplete="new-password">
```

```
<input type="submit"
  value="Change Password">
```

```
</form>
```

Current Password

New Password

Change Password



# Changing One's Password

After forgetting the password



```
<form id="change-password"
  action="update-user.php" method="post">
  <input type="text" autocomplete="username"
    readonly value="rmondello">
  <input type="password"
    autocomplete="new-password">
  <input type="submit"
    value="Change Password">
</form>
```

Username

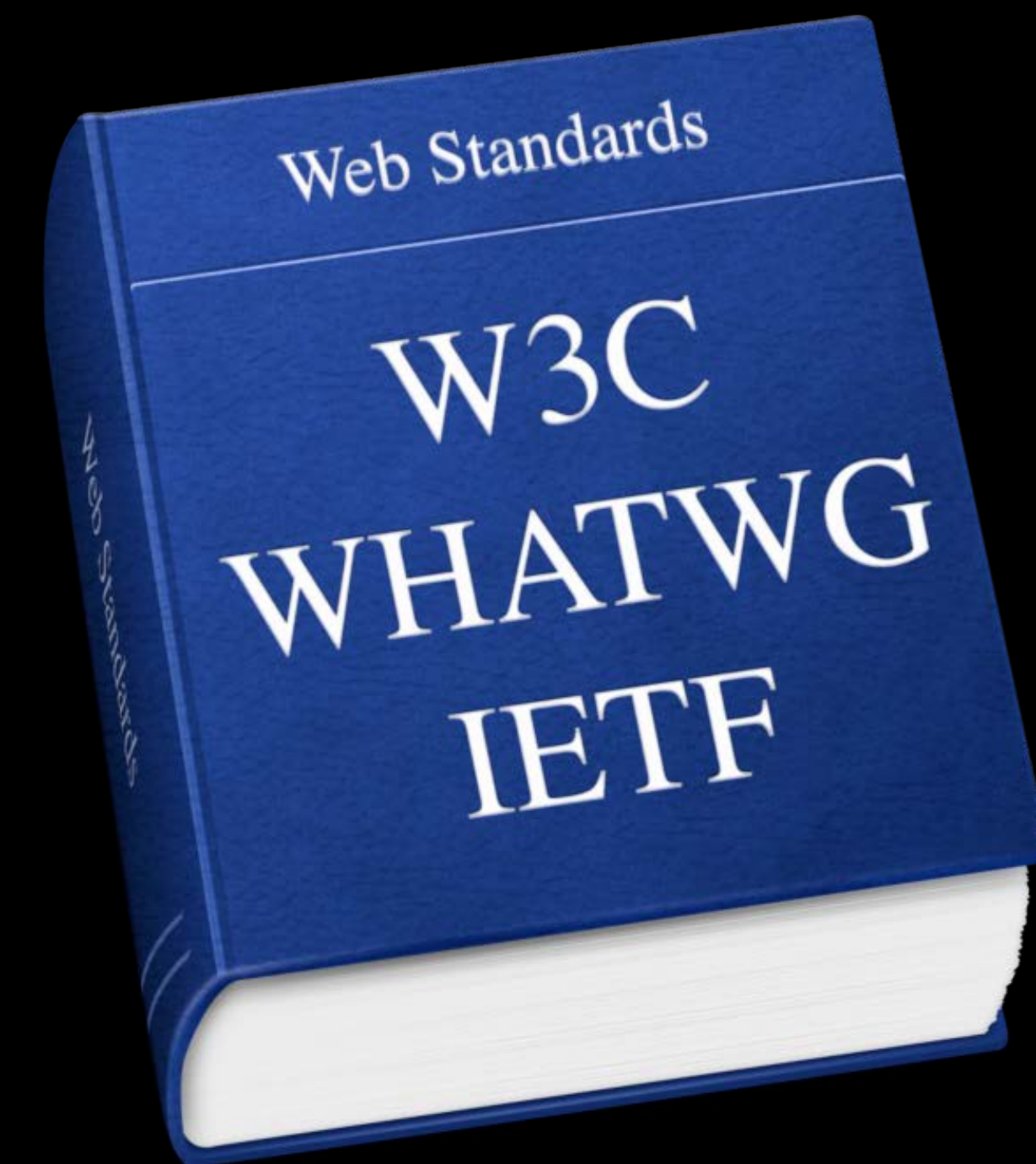
Password

# Your Forms' Intent

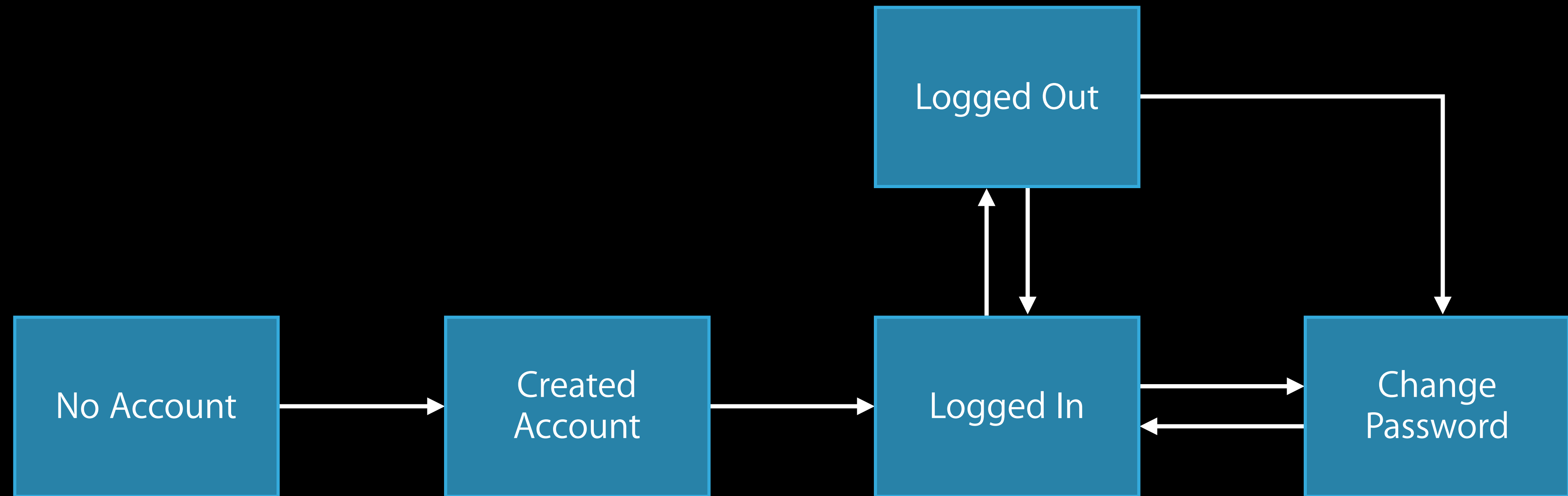
The autocomplete attribute

NEW

username	email
current-password	street-address
new-password	city
name-full	state
given-name	province
middle-name	postal-code
family-name	country
birthday-year	phone
birthday-month	phone-area-code
birthday-day	organization



# Testing Your Site and Safari AutoFill







# Declaring a State Transition

## History API

`history.pushState`

`history.replaceState`

# Declaring a State Transition

## History API

```
history.pushState(data, null, "page.html");
```

```
history.replaceState(data, null, "page.html");
```

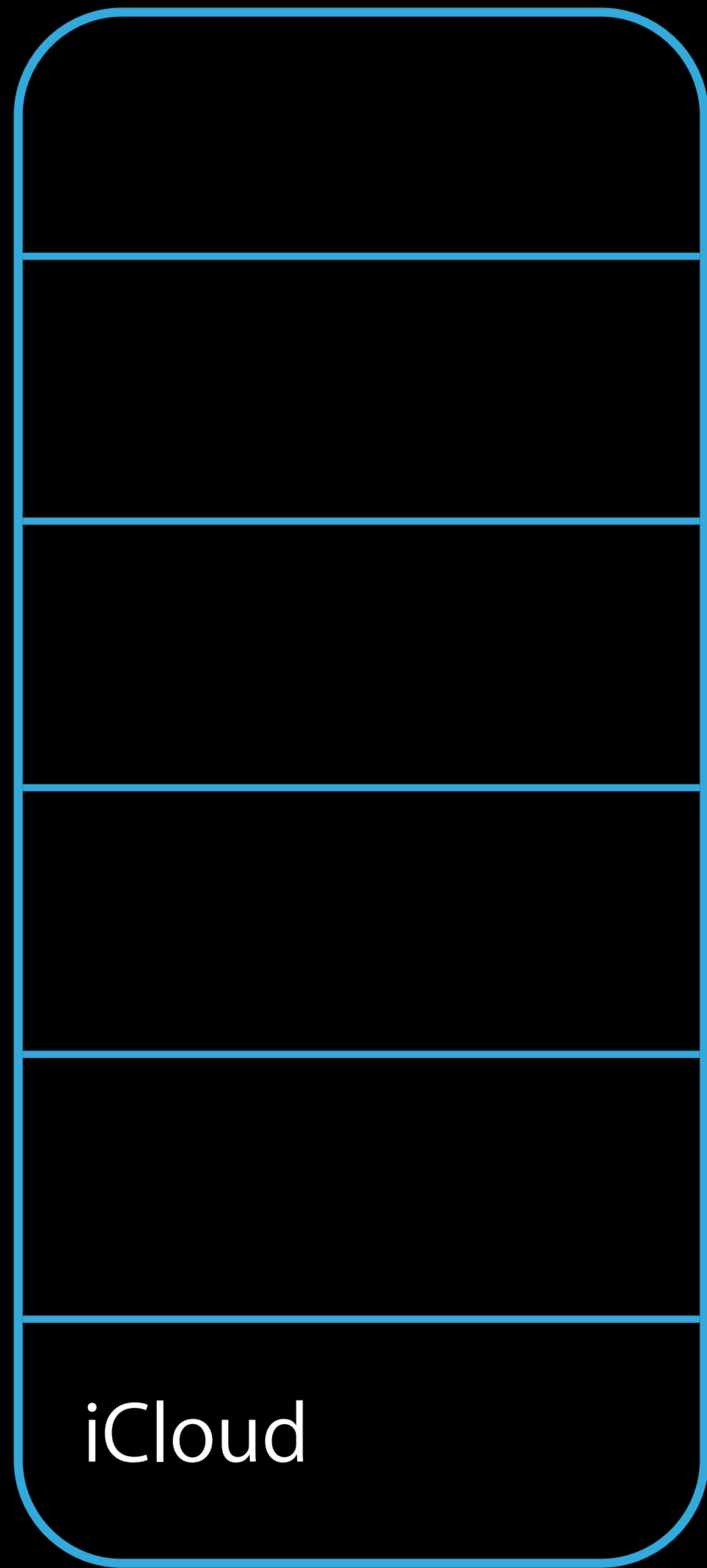




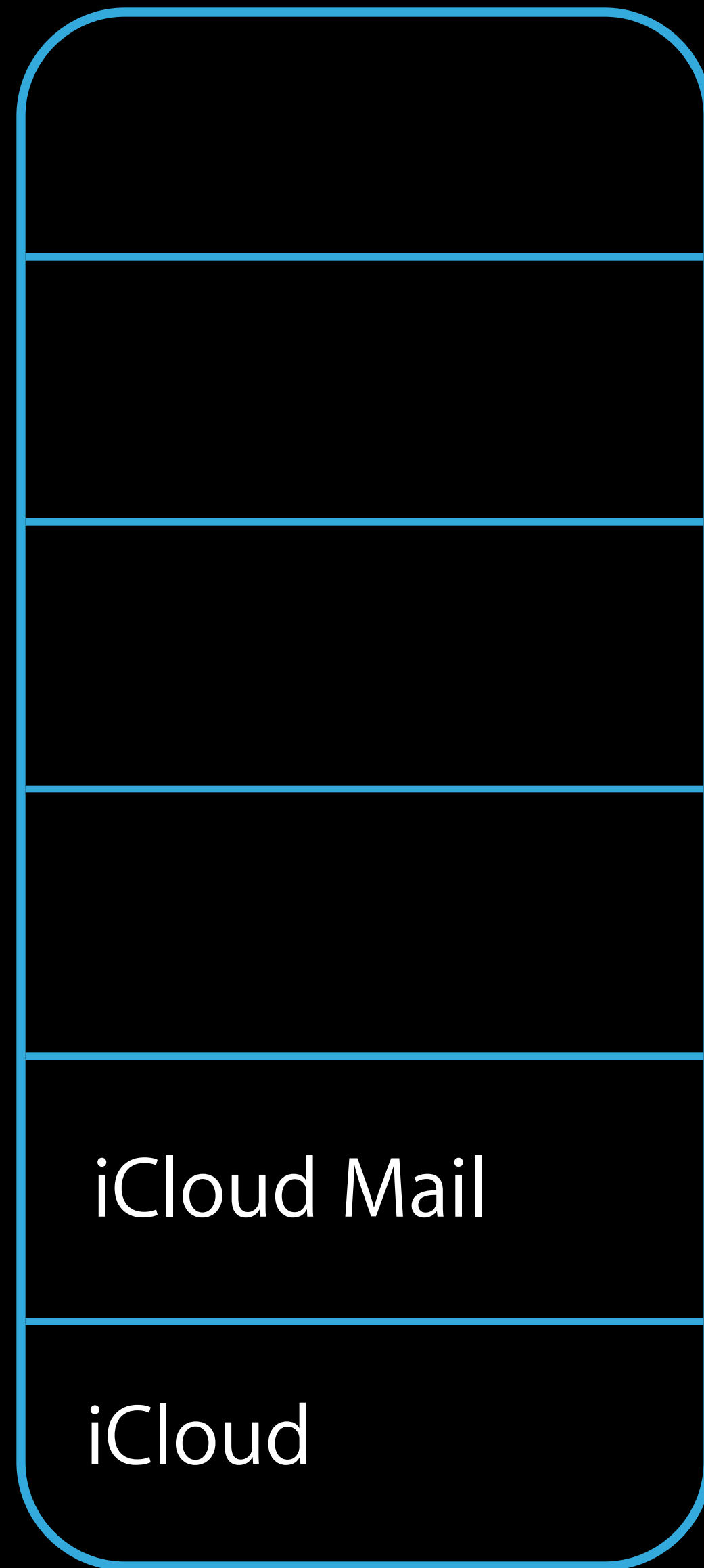






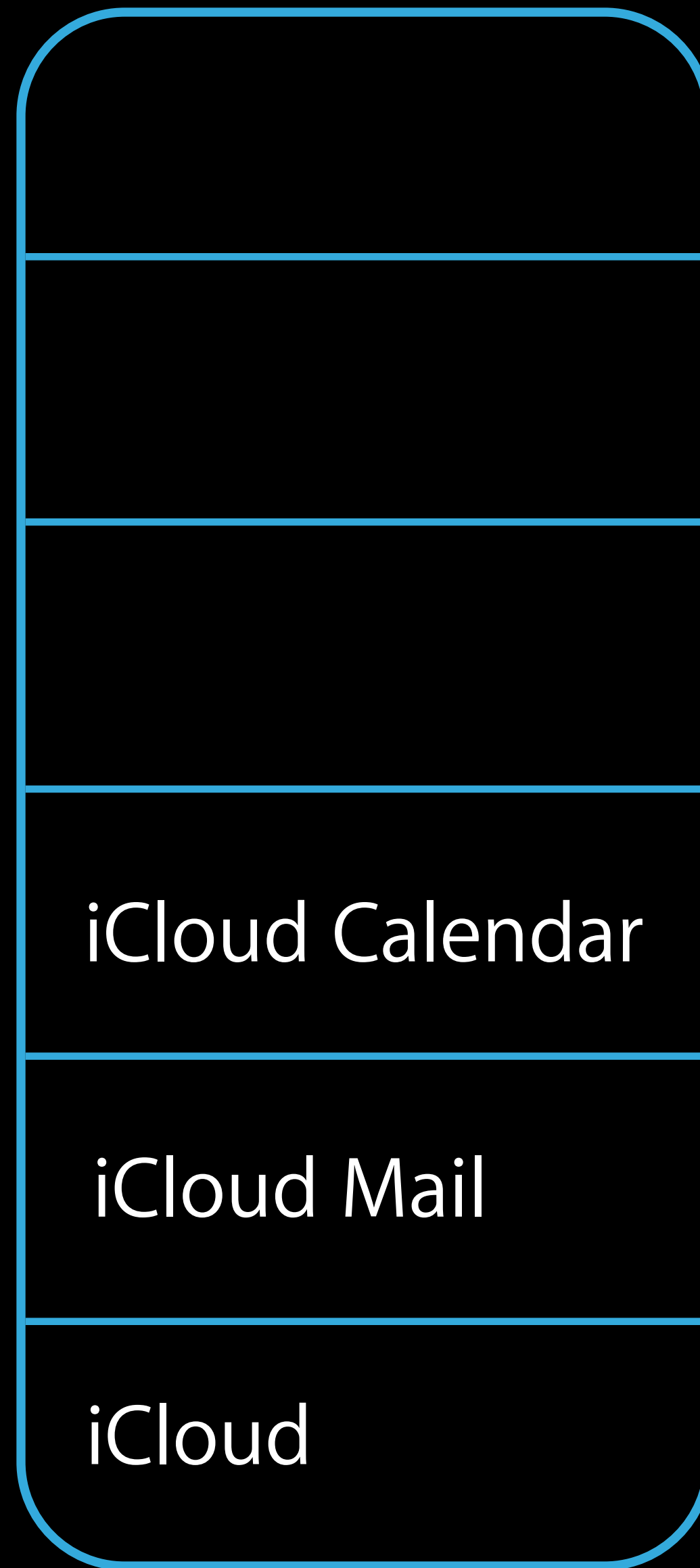


iCloud



iCloud Mail

iCloud



Twitter

iCloud Calendar

iCloud Mail

iCloud

A User Profile

Twitter

iCloud Calendar

iCloud Mail

iCloud



A Tweet

A User Profile

Twitter

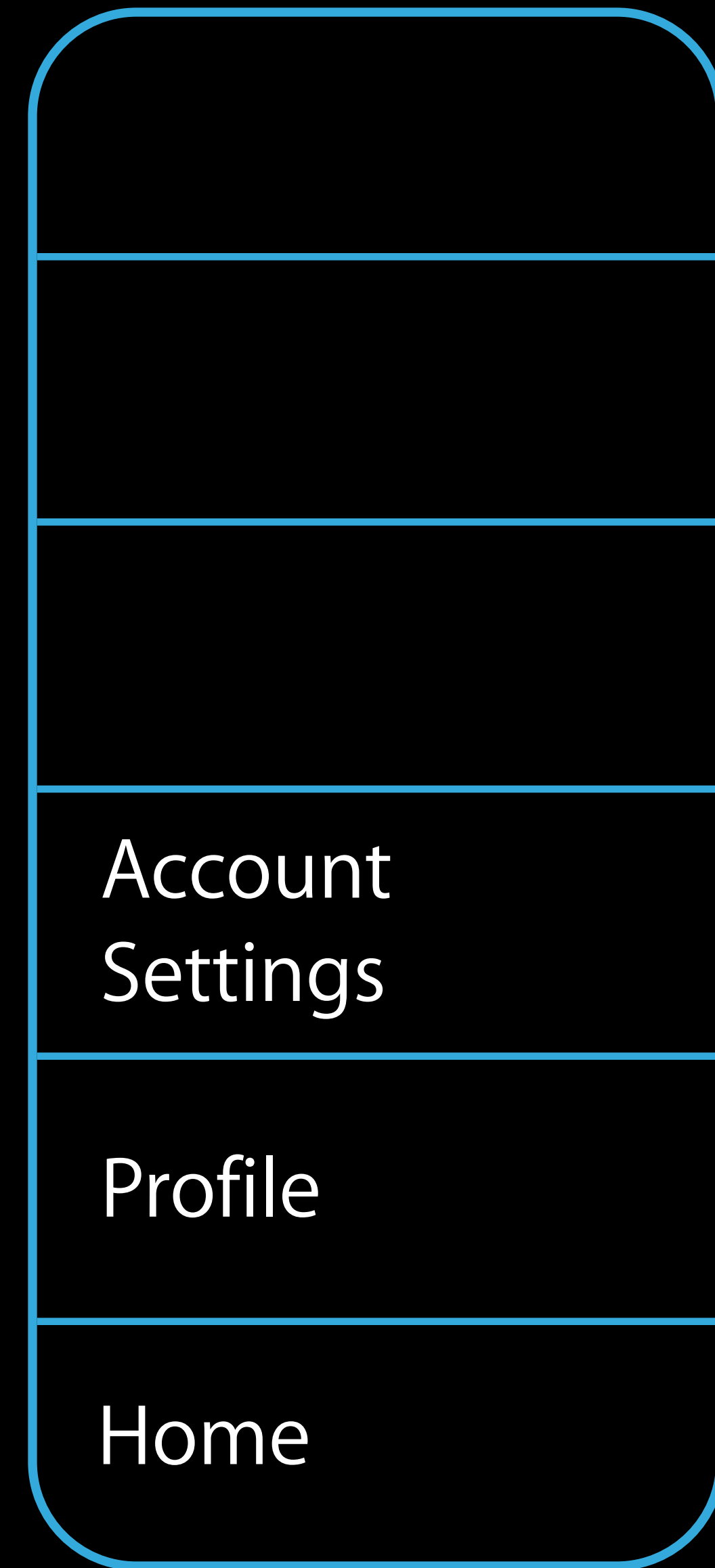
iCloud Calendar

iCloud Mail

iCloud

# Declaring a State Transition

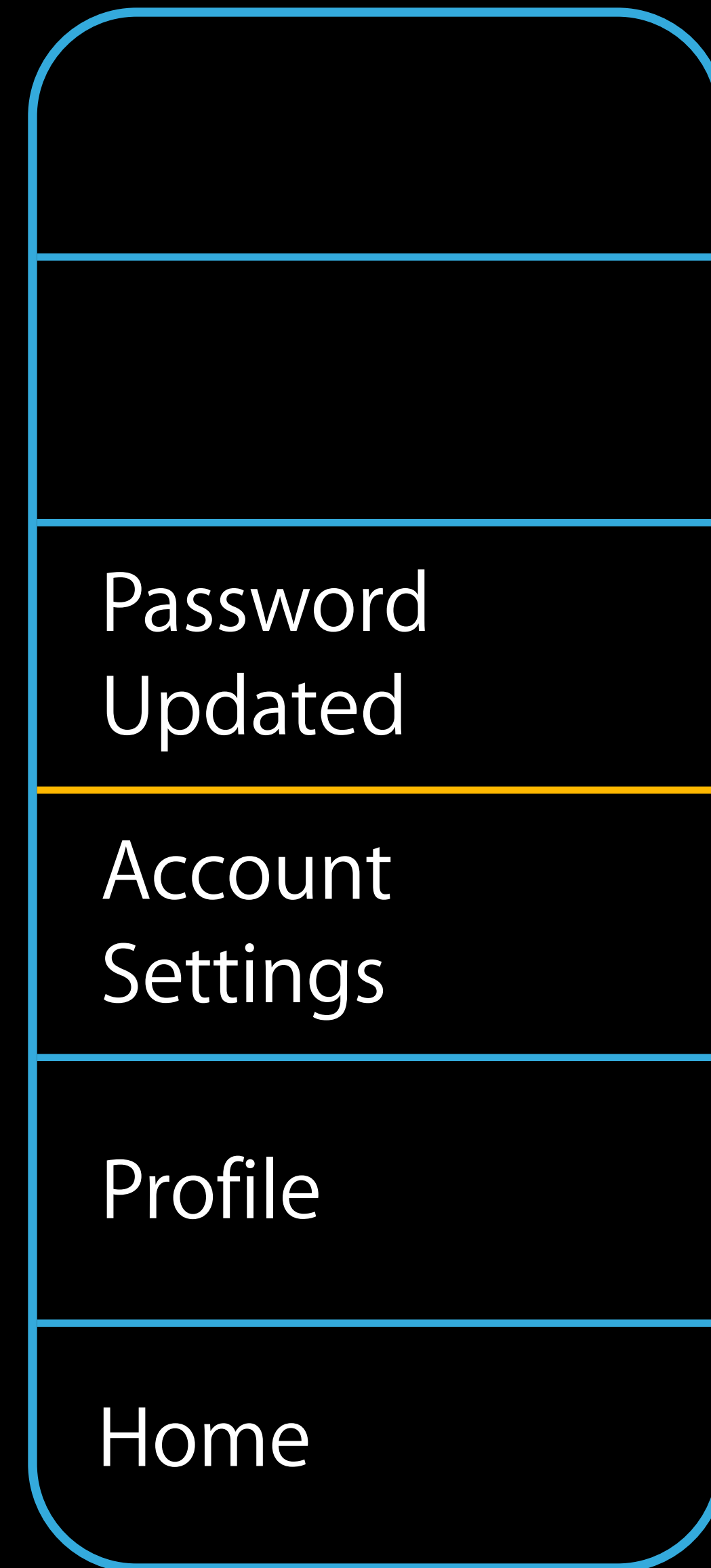
History API

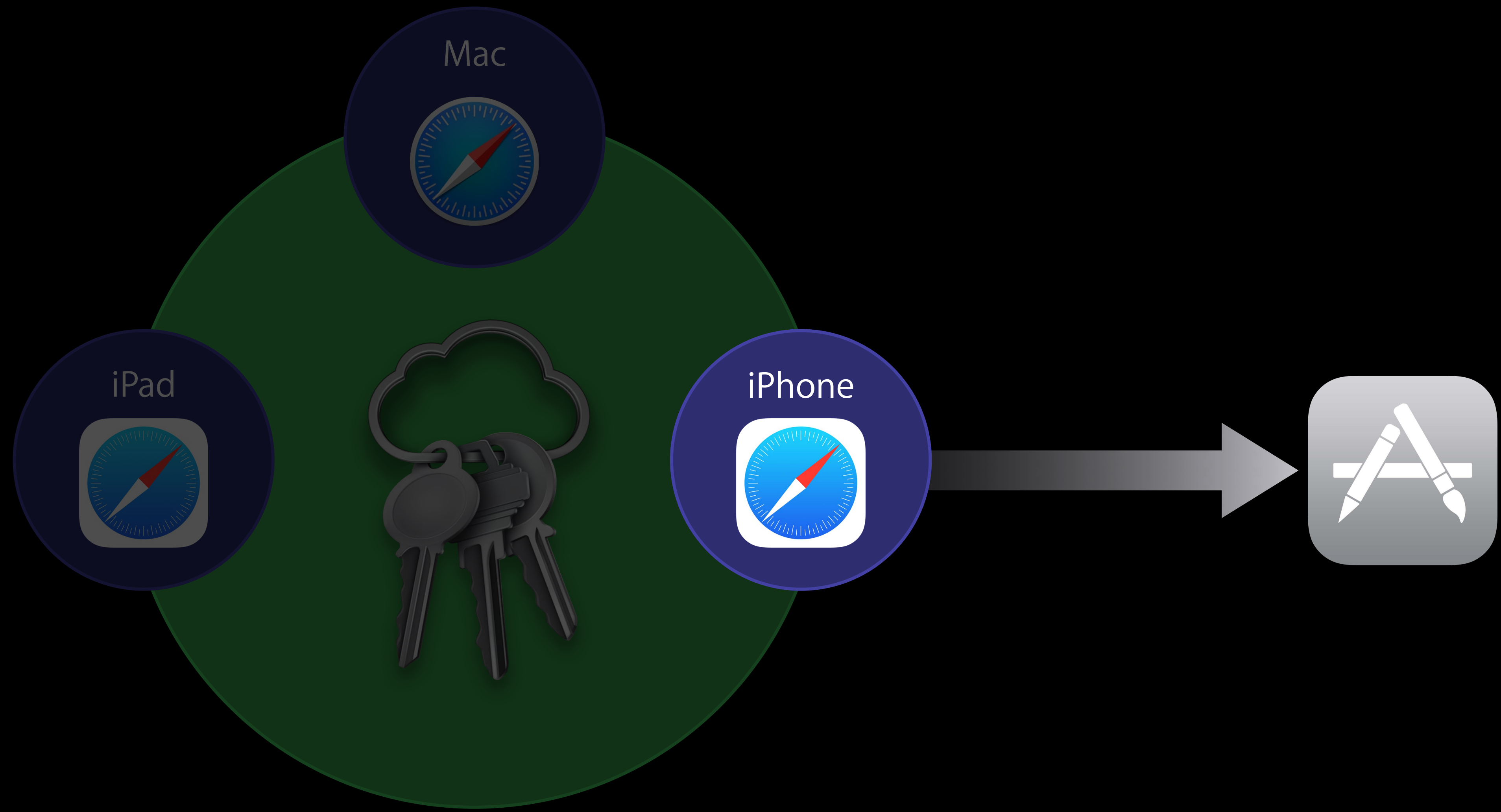


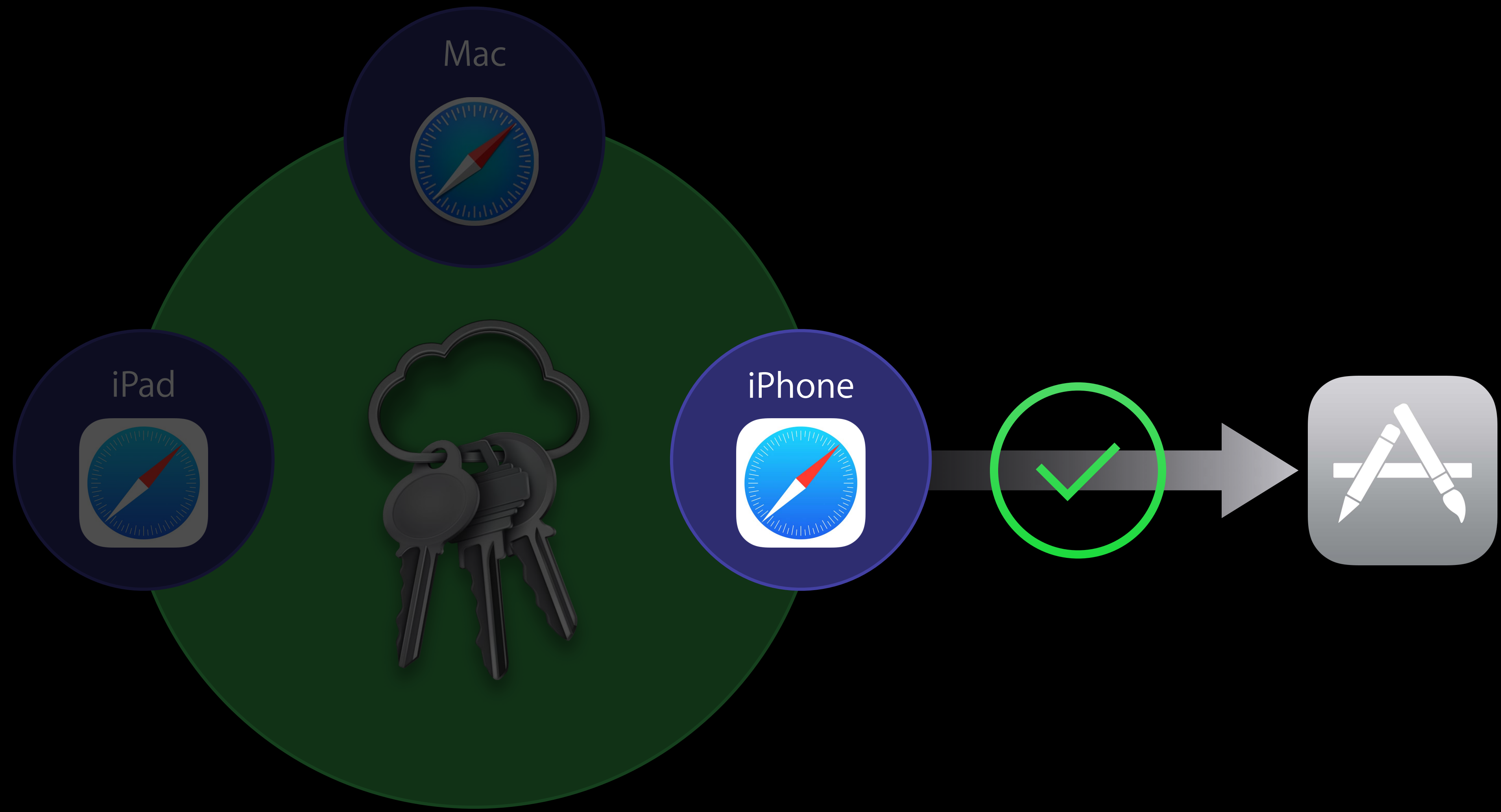
# Declaring a State Transition

## History API

```
history.pushState({ data: "yourData" }, null);
```







# Safari and Your Website's Content

Iconography and metadata



9:41 AM 100%

Search or enter website name



Apple



Facebook



Twitter

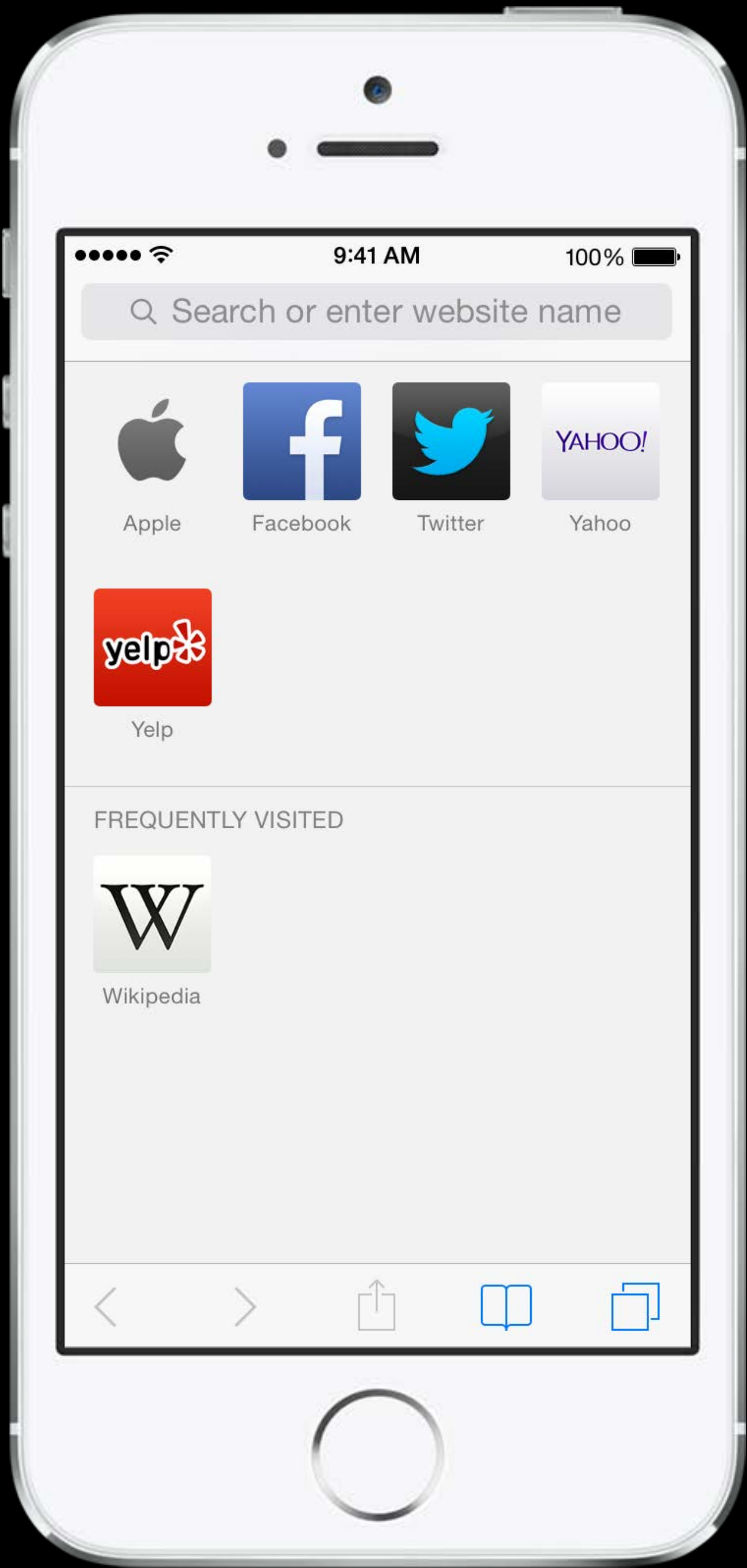


Yahoo



Yelp





9:41 AM 100%

Search or enter website name



Apple



Facebook



Twitter



Yahoo



Yelp

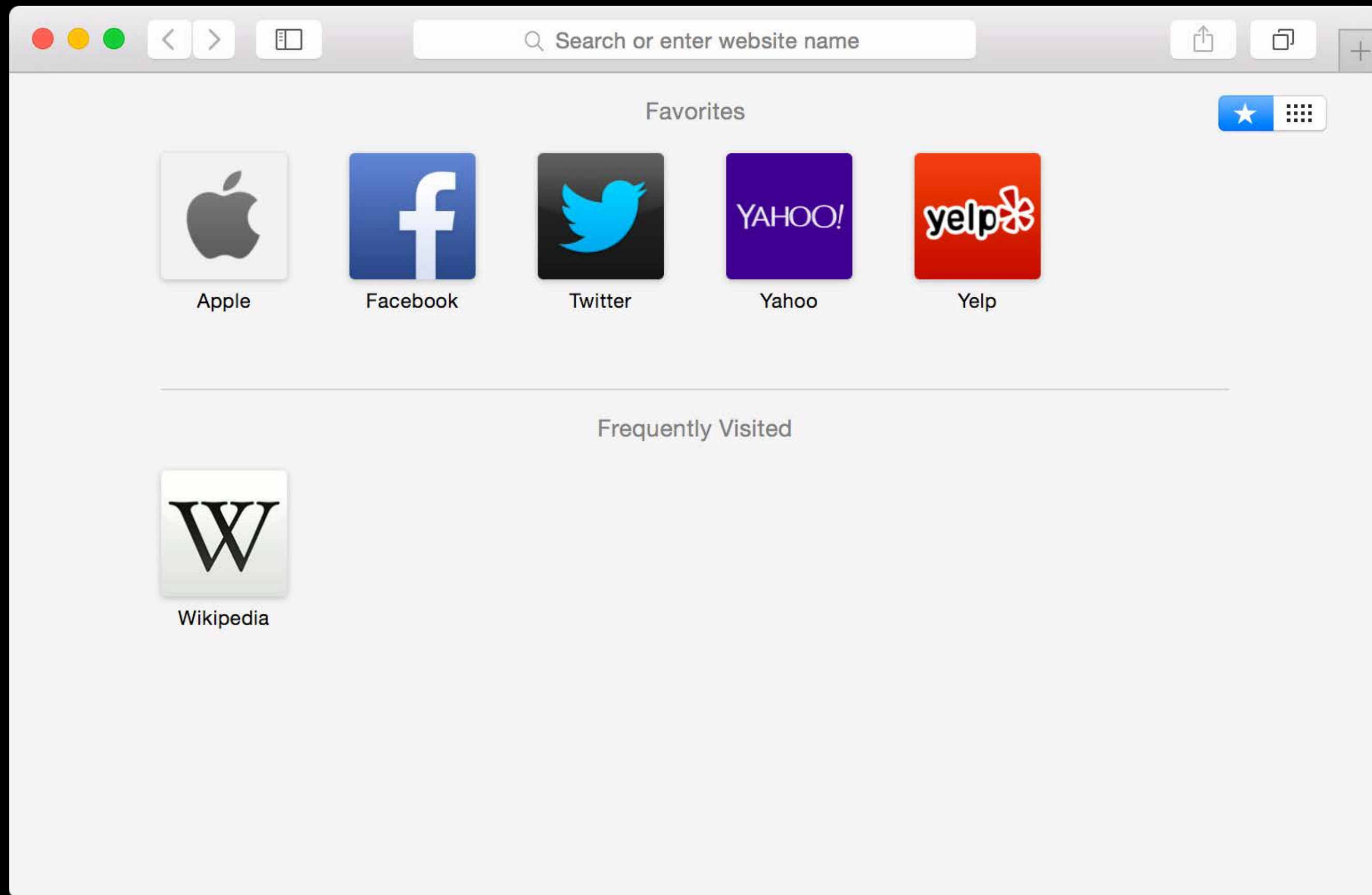
FREQUENTLY VISITED

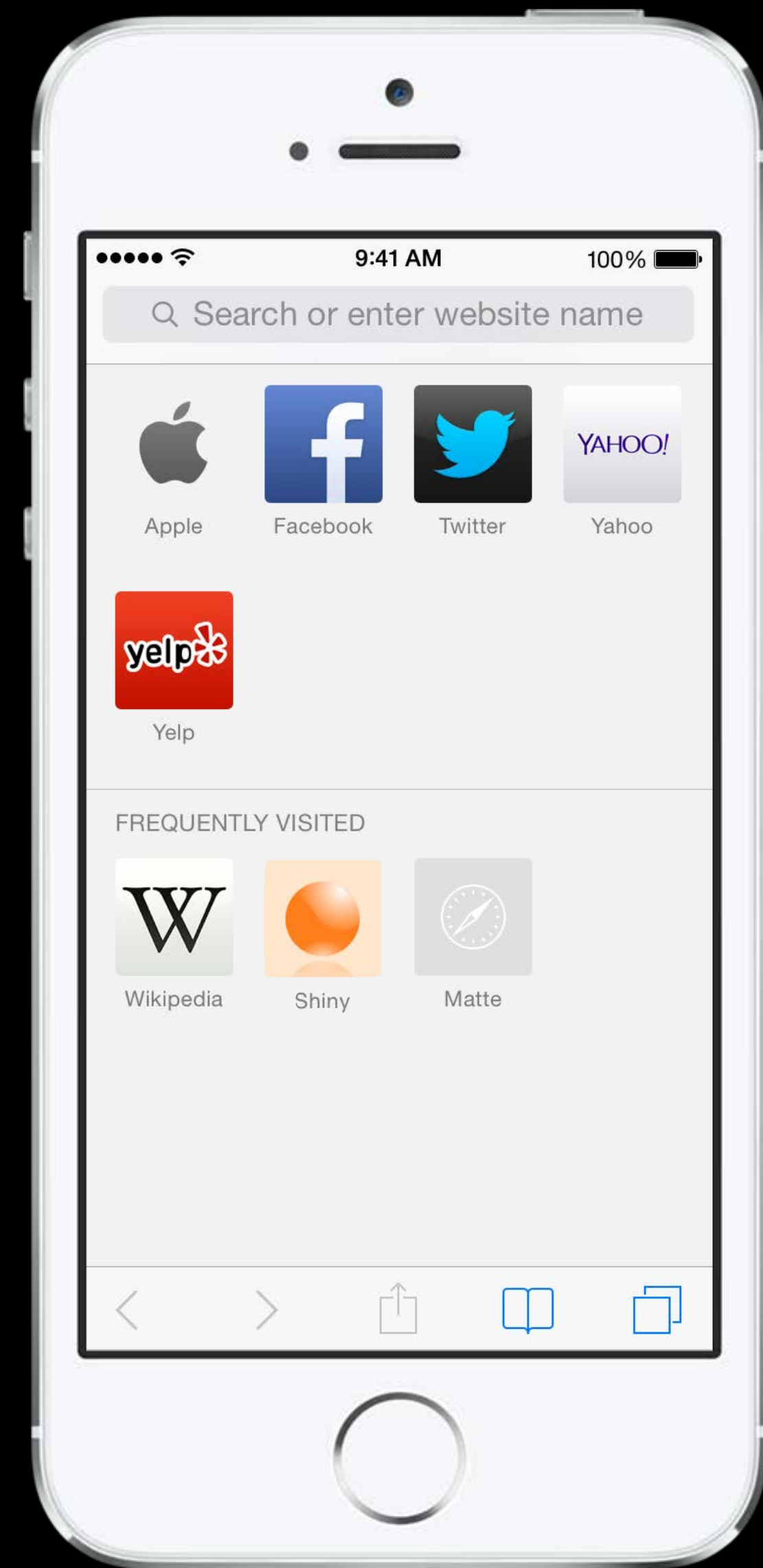
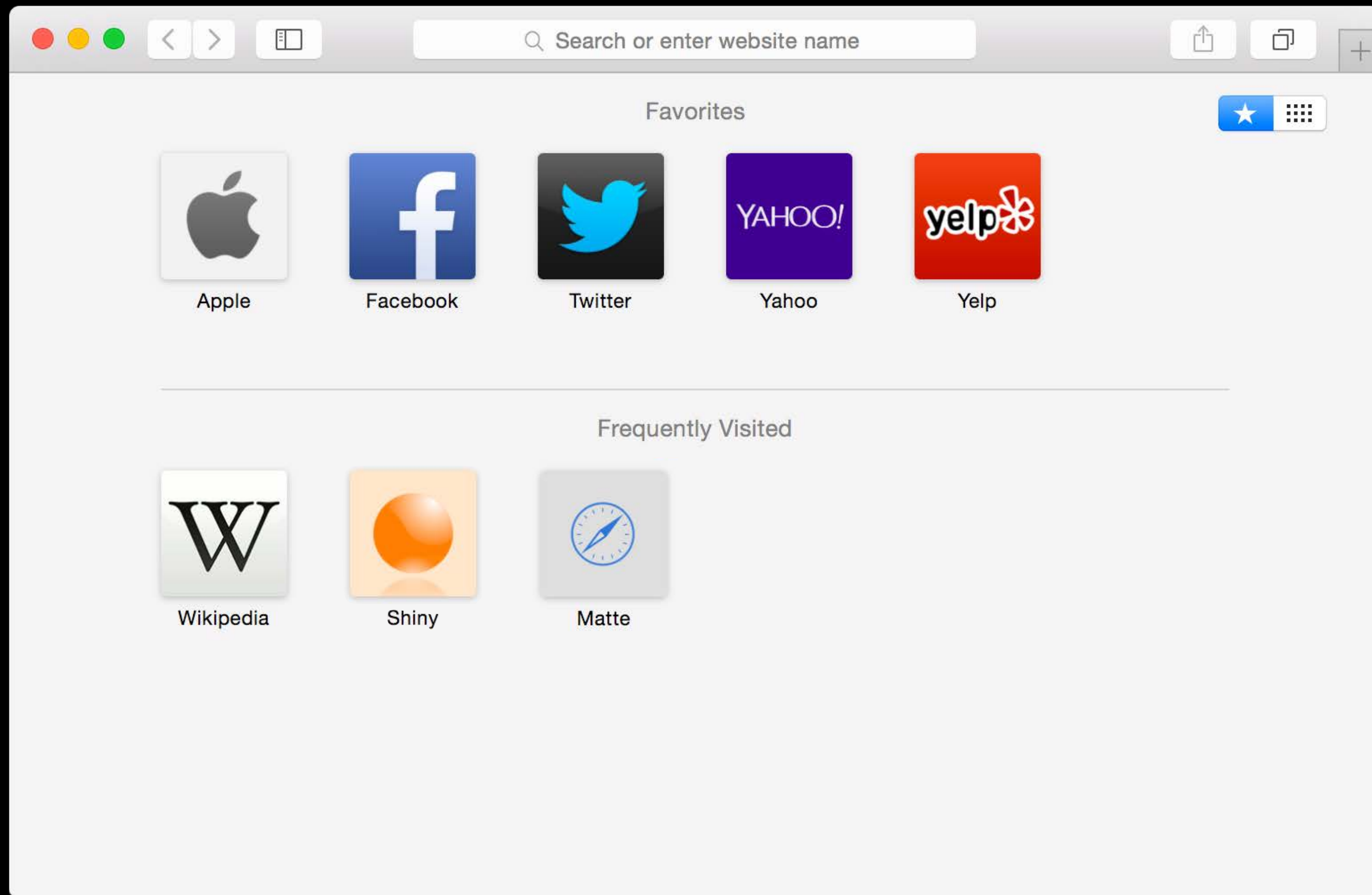


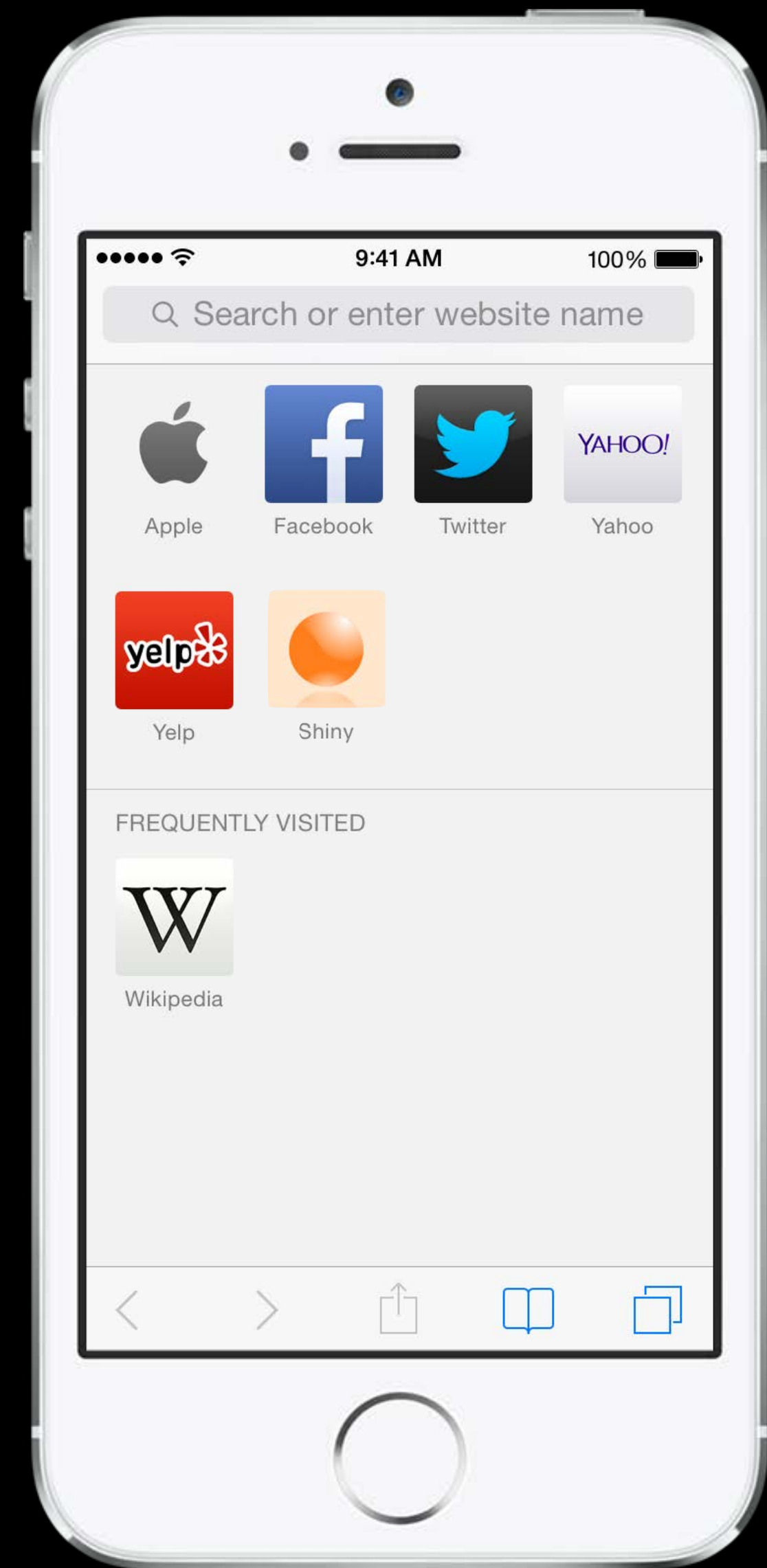
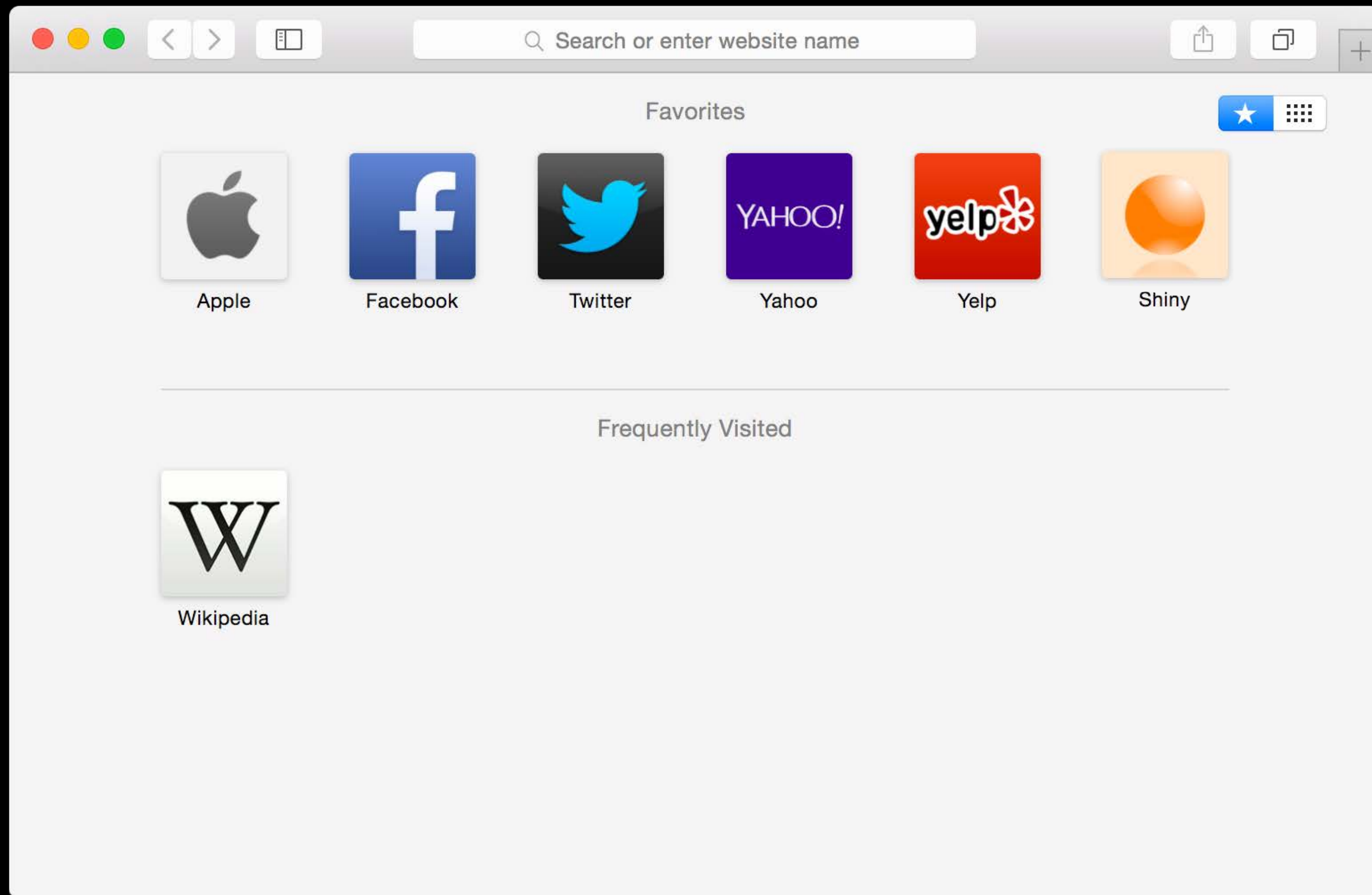
Wikipedia











# Iconography

## Favicon

In-page markup:

```
<head>  
  <link rel="icon"  
        type="image/png"  
        href="http://www.example.com/favicon.png">  
</head>
```

Standard path:

```
/favicon.ico
```

# Iconography

## Apple Touch Icon

Used on iOS home screen

In-page markup:

```
<head>  
  <link rel="apple-touch-icon" sizes="152x152"  
        href="/apple-touch-icon.png">  
</head>
```

See *Configuring Web Applications* on the Safari Web Content Guide

Apple Touch Icon

# Iconography

Apple Touch Icon



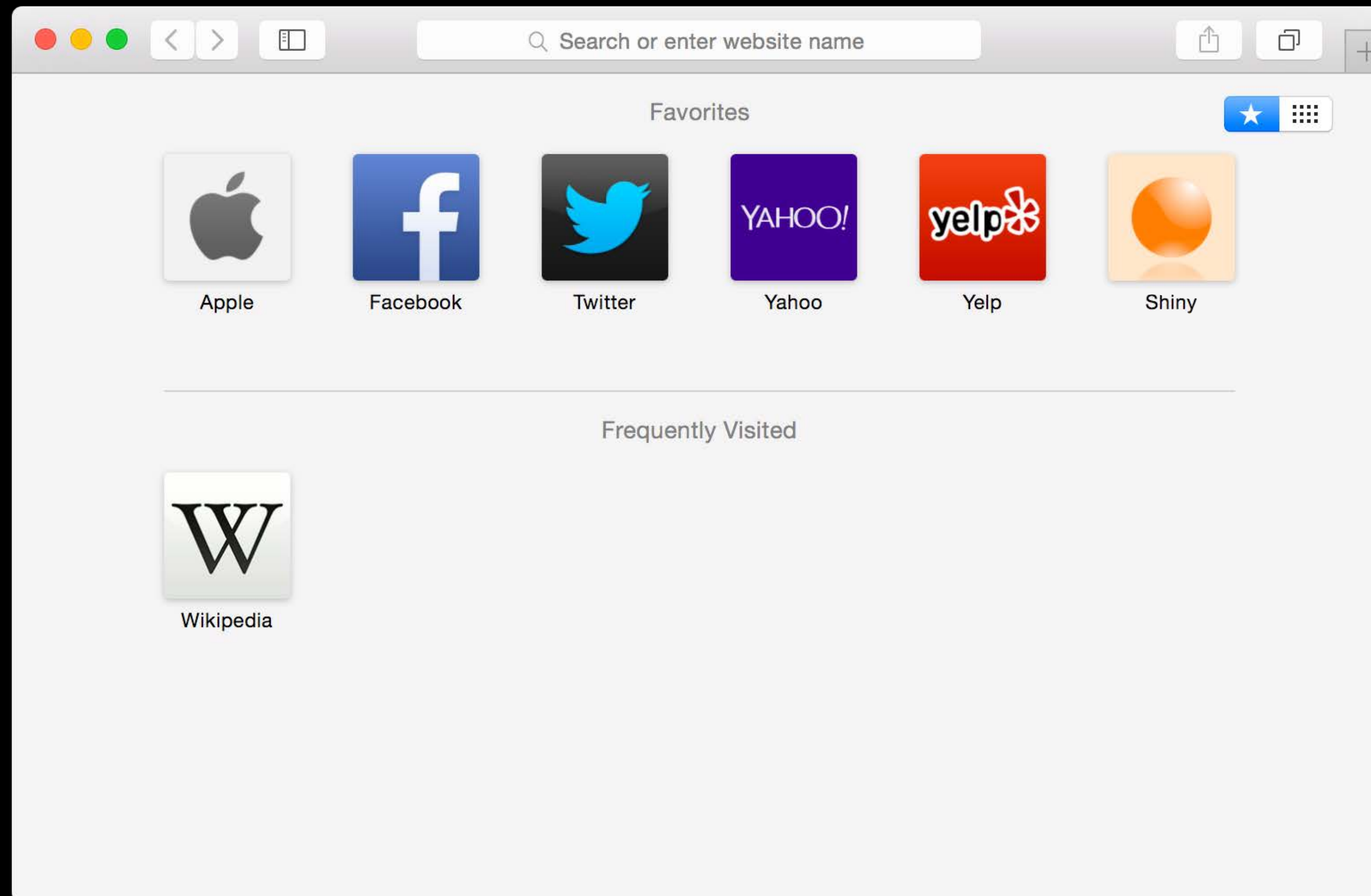
# Iconography

Apple Touch Icon



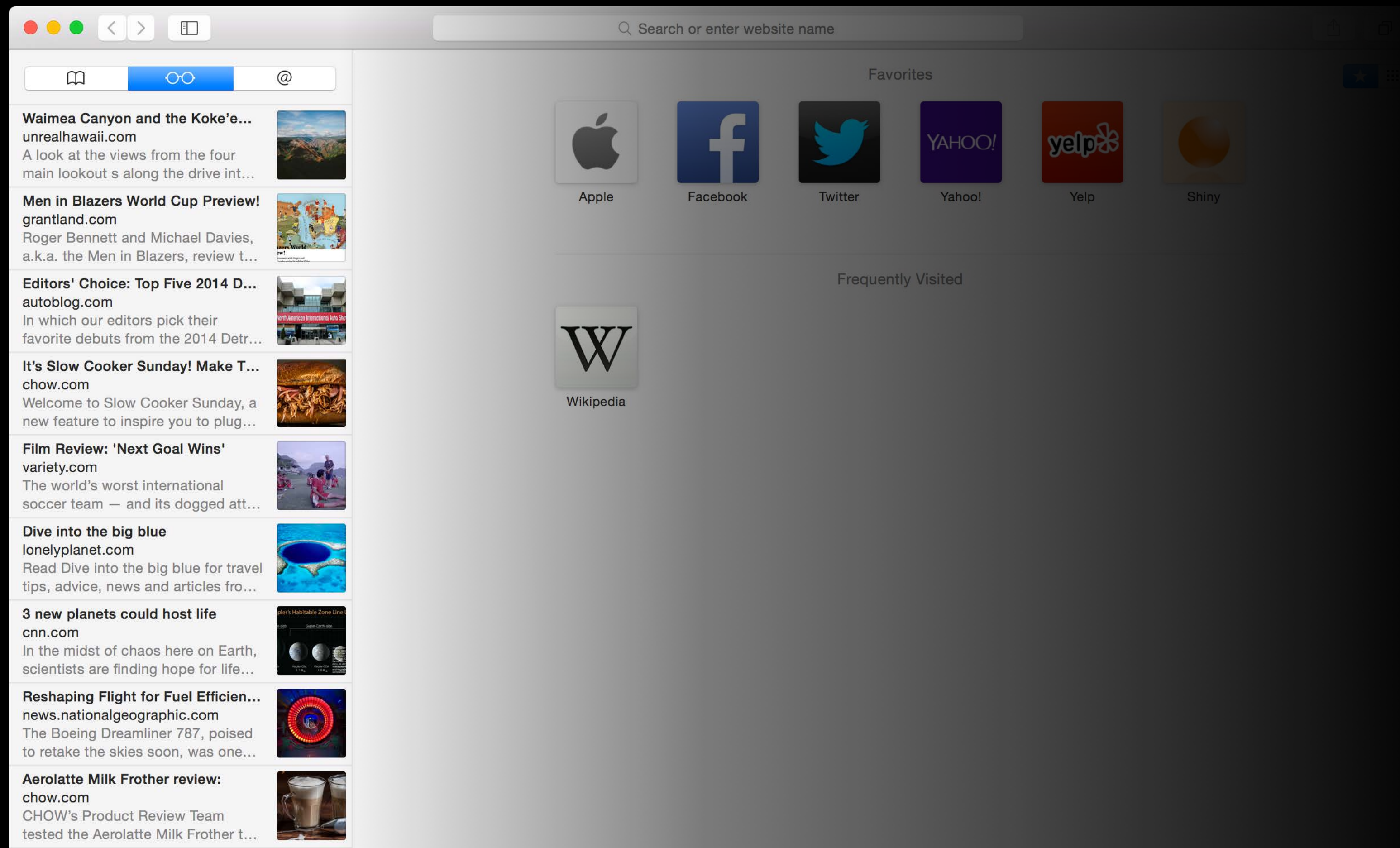


# Iconography

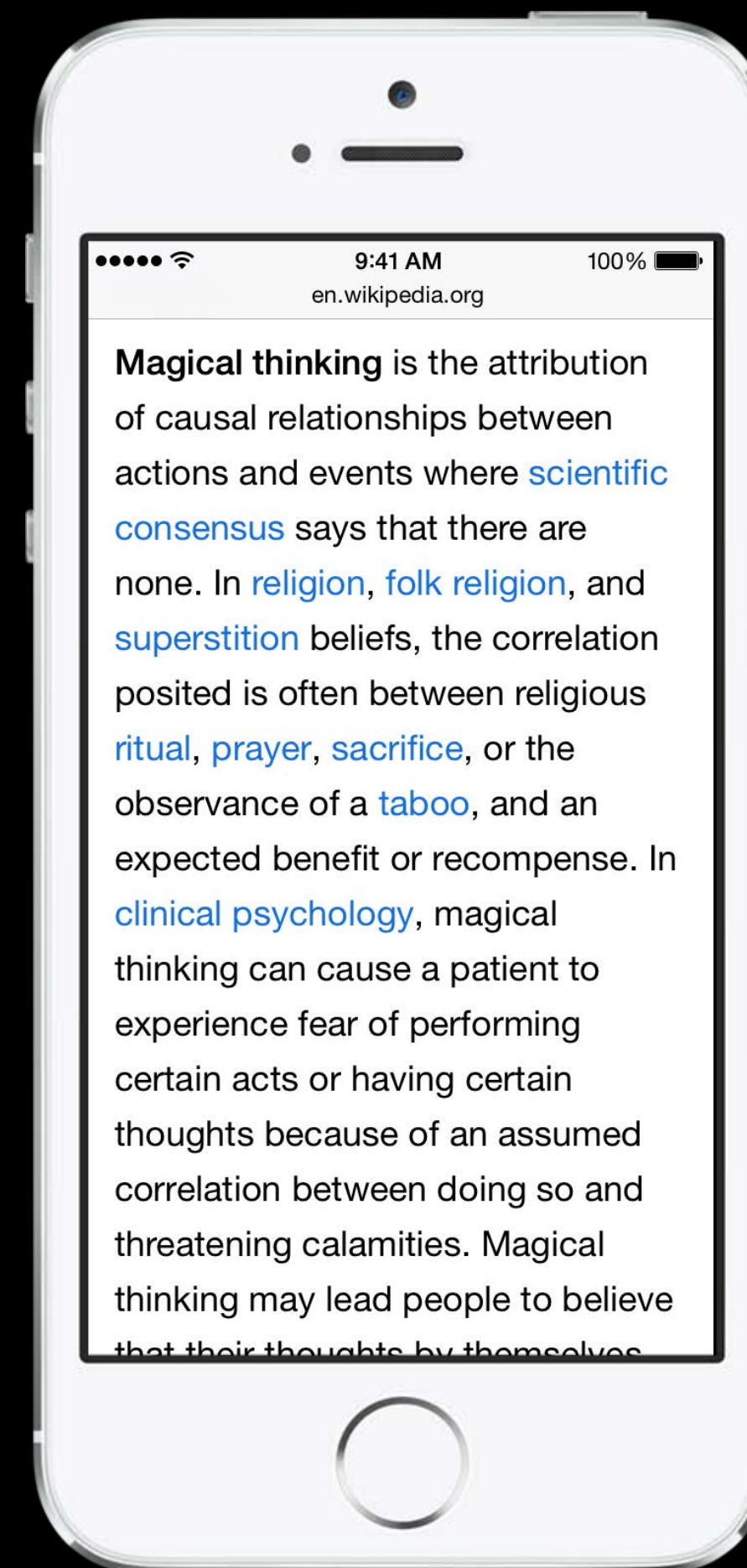
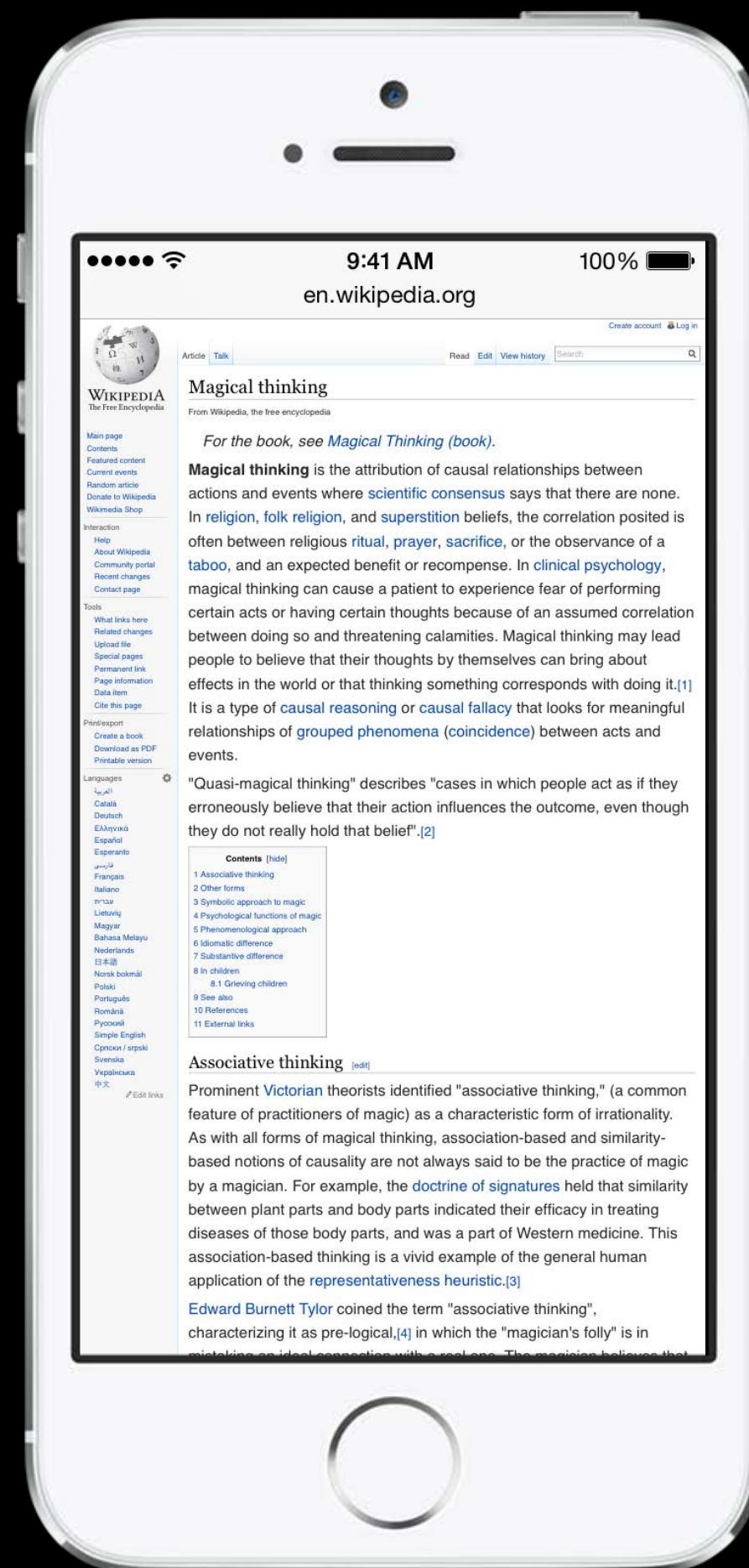




# Reading List



# Safari Reader



# Shared Links

The image shows a screenshot of a web browser window. On the left side, there is a social media feed with several items:

- Ricky Mondello**: Always be on the lookout for magical thinking. [en.wikipedia.org/wiki/Magical t...](http://en.wikipedia.org/wiki/Magical_t...)
- iBooks**: Find TFIOS and all of John Green's books on iBooks and pre-order the #TFIOS movie on iTunes. #DFTBA [tw.apple.com/TFIOS](http://tw.apple.com/TFIOS)
- iBooks**: Twenty years ago what did you imagine yourself doing in 2014? #AskJohnGreen [tw.apple.com/TFIOS](http://tw.apple.com/TFIOS)
- App Store**: Get the whole picture the whole time with The Guardian. [tw.appstore.com/6010cKgw](http://tw.appstore.com/6010cKgw) <http://t.co/6UjgwXtRT0>
- iTunes Music**: "If you feel like leaving, you know you can go / But why don't you stay until tomorrow." @halloates [tw.itunes.com/6014cyau](http://tw.itunes.com/6014cyau)
- iTunes Music**: "The hurt doesn't show / But the pain still grows / It's no stranger to you and me."  
(air drums)  
@PhilCollinsFeed [tw.itunes.com/6019cya5](http://tw.itunes.com/6019cya5)
- iTunes**: The neon lights are bright indeed.

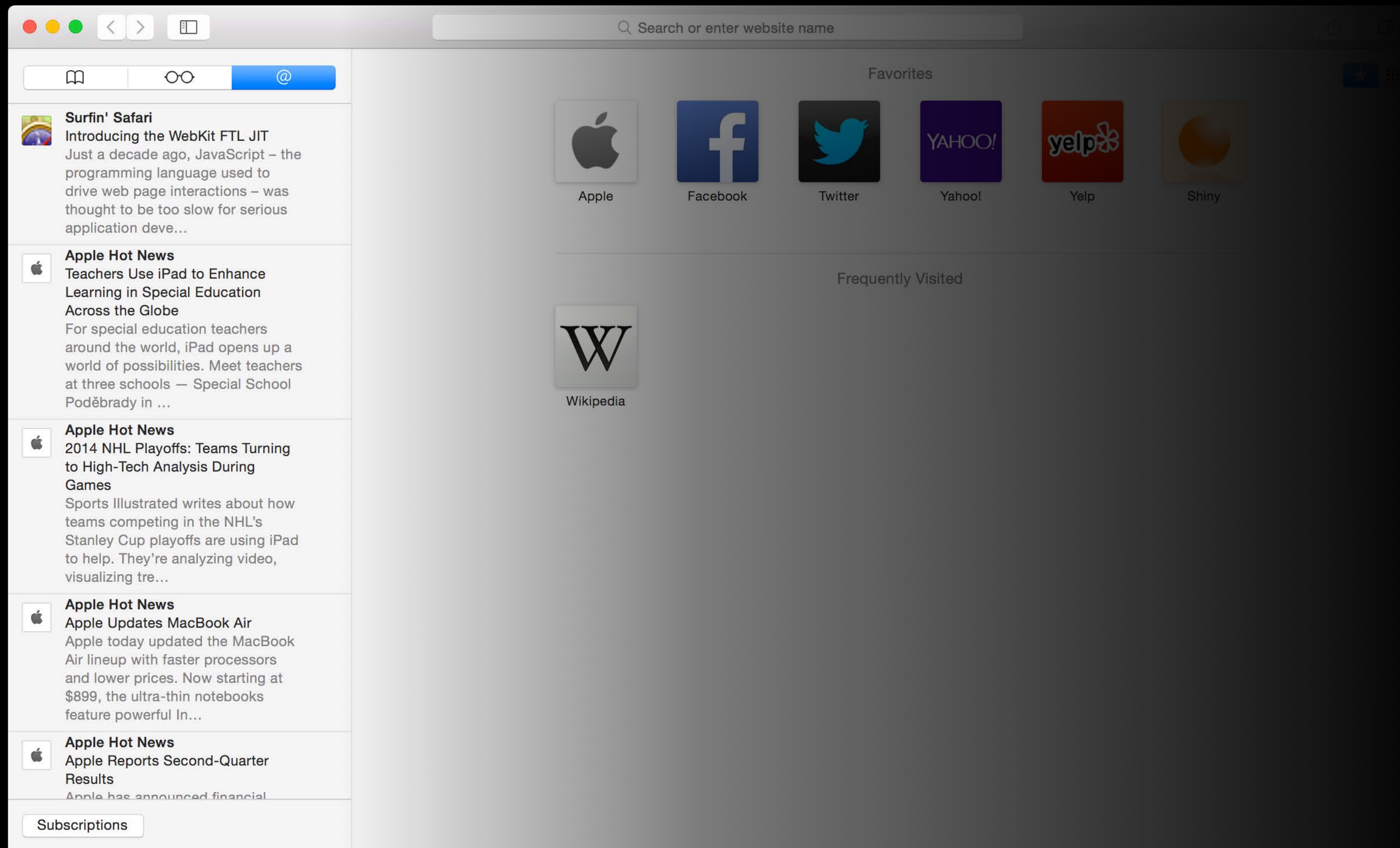
At the bottom of the feed is a "Subscriptions" button.

The main content area of the browser shows a homepage with a search bar at the top. Below the search bar, there are two sections:

- Favorites**: A row of icons for Apple, Facebook, Twitter, Yahoo!, Yelp, and Shiny.
- Frequently Visited**: A single icon for Wikipedia.

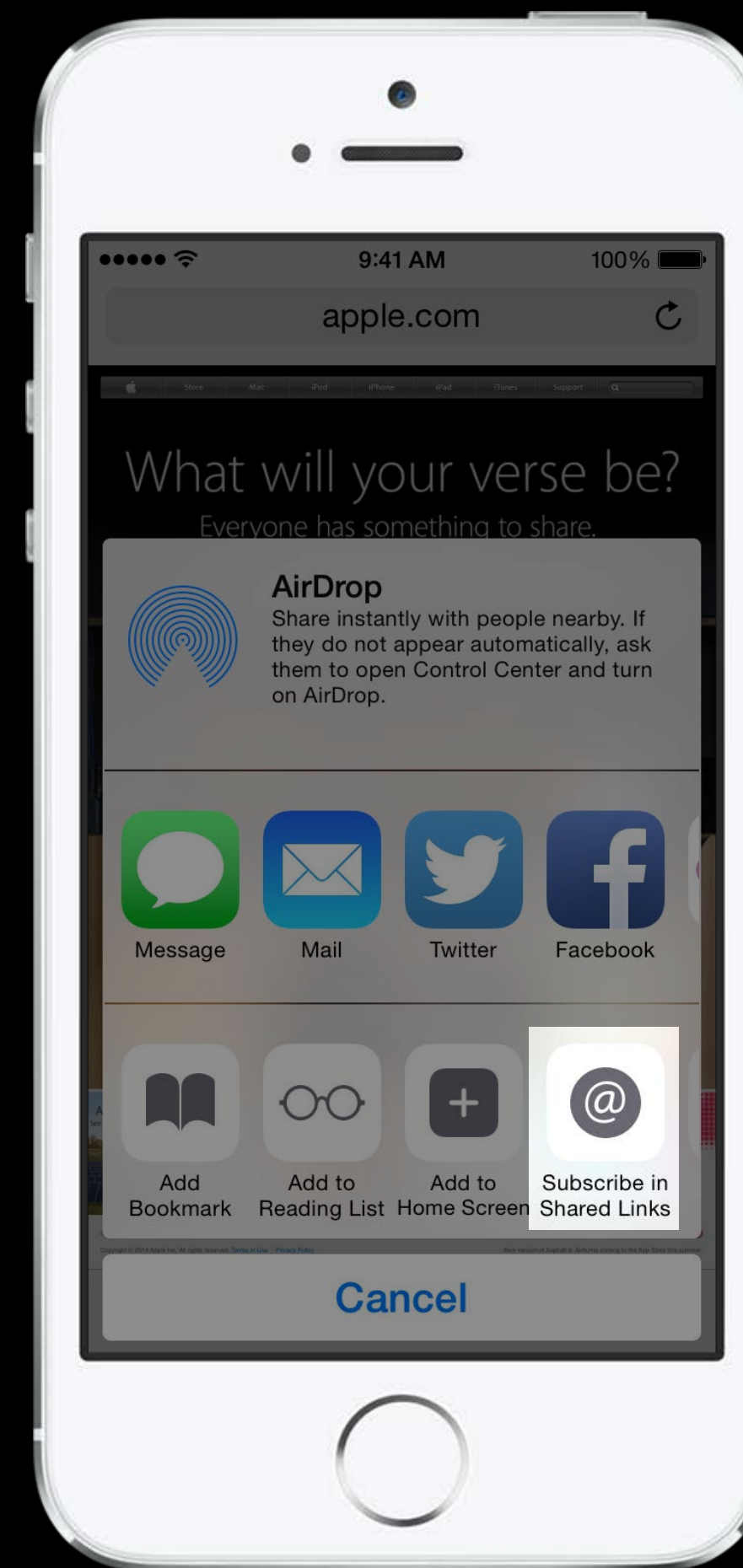
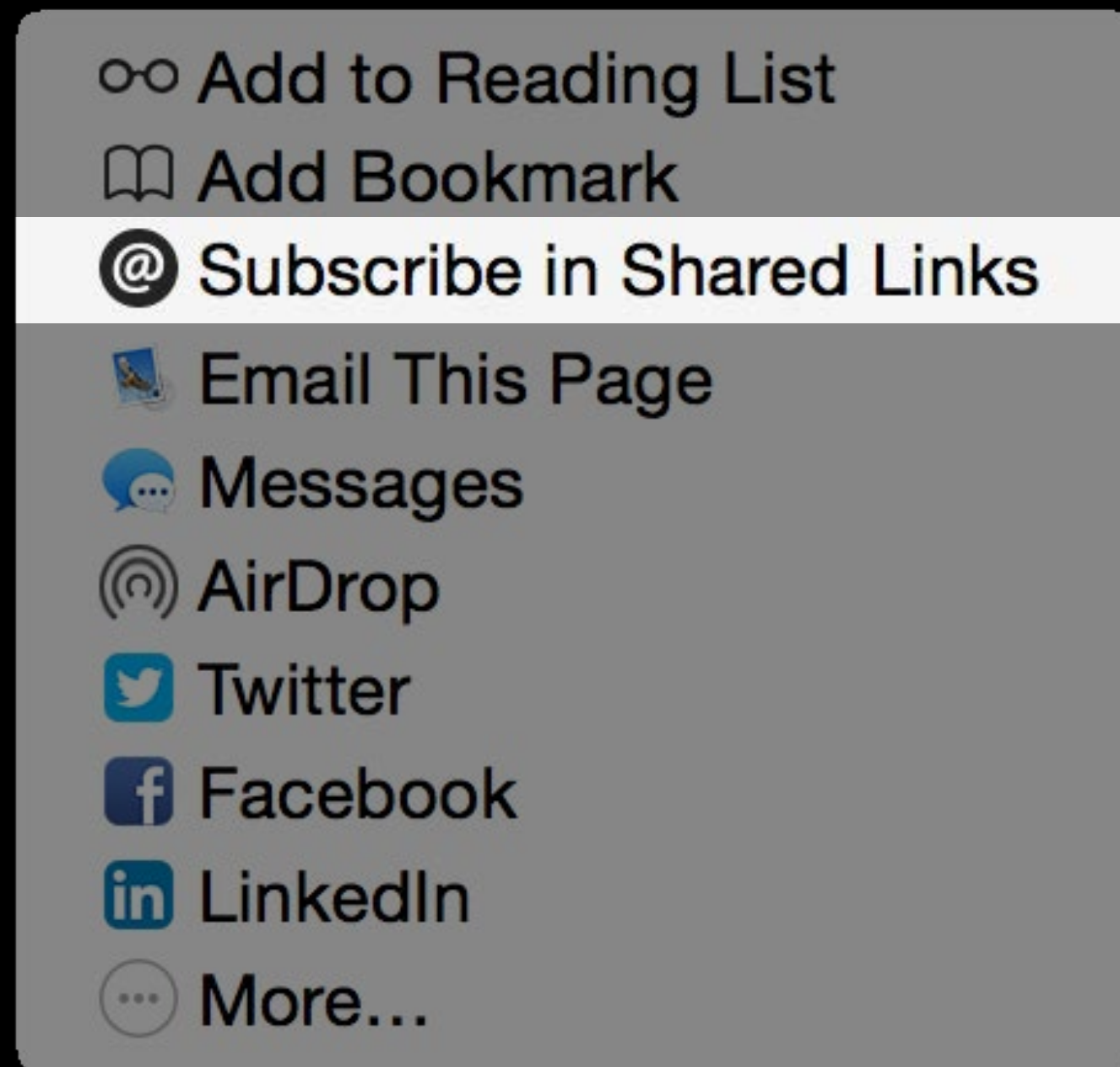
# Your Site in Shared Links

NEW



# Your Site in Shared Links

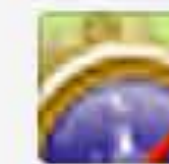
NEW



# Your Site in Shared Links

NEW

```
<channel>
  <title>Surfin' Safari</title>
  <item>
    <title>Introducing the WebKit FTL JIT</
title>
    <link>https://www.webkit.org/blog/3362/
introducing-the-webkit-ftl-jit/</link>
    <pubDate>Tue, 13 May 2014 02:52:42 GMT
</pubDate>
    <content type="html">
      <![CDATA[<p>Just a decade ago...</p>]]>
    </content>
  </item>
</channel>
```



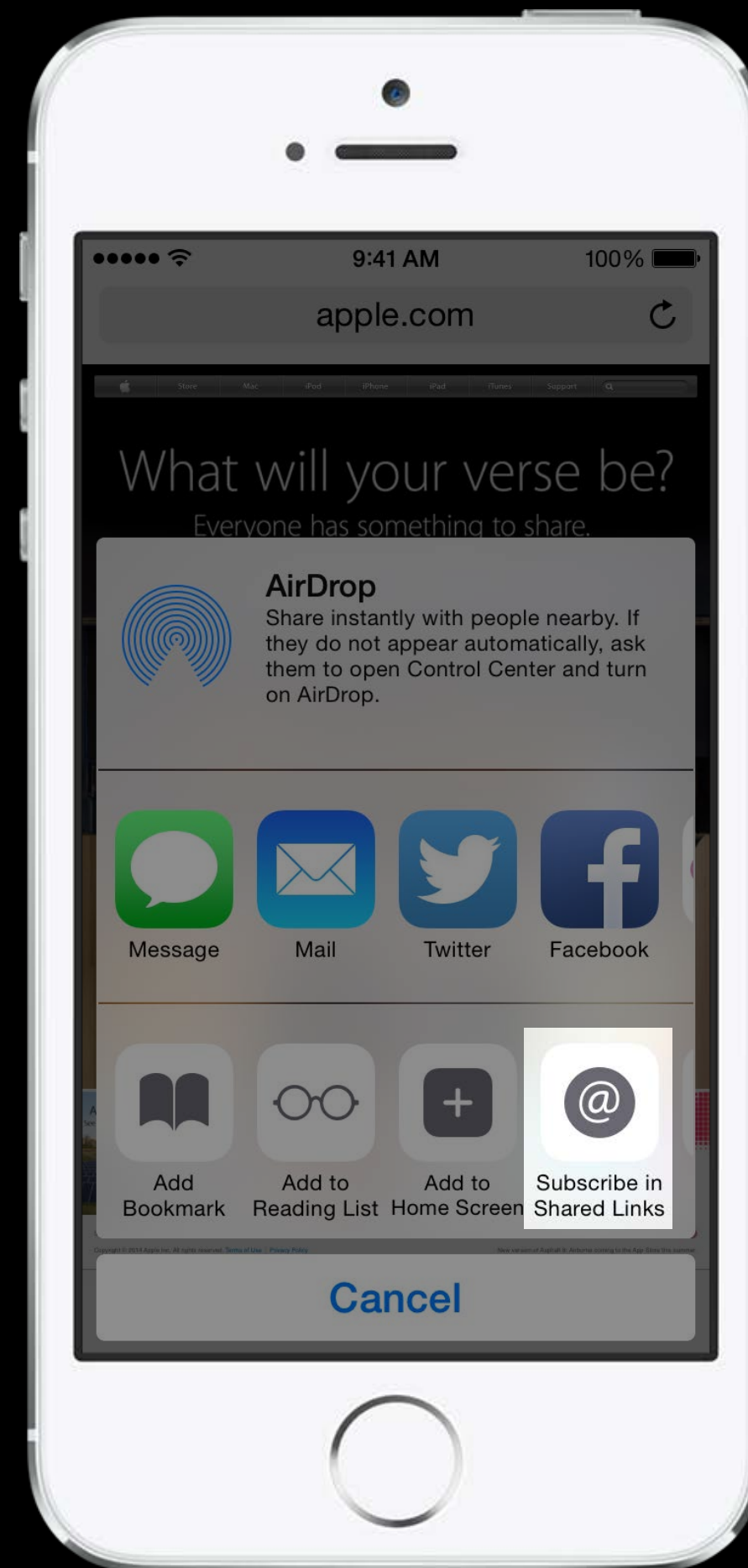
## Surfin' Safari

Introducing the WebKit FTL JIT

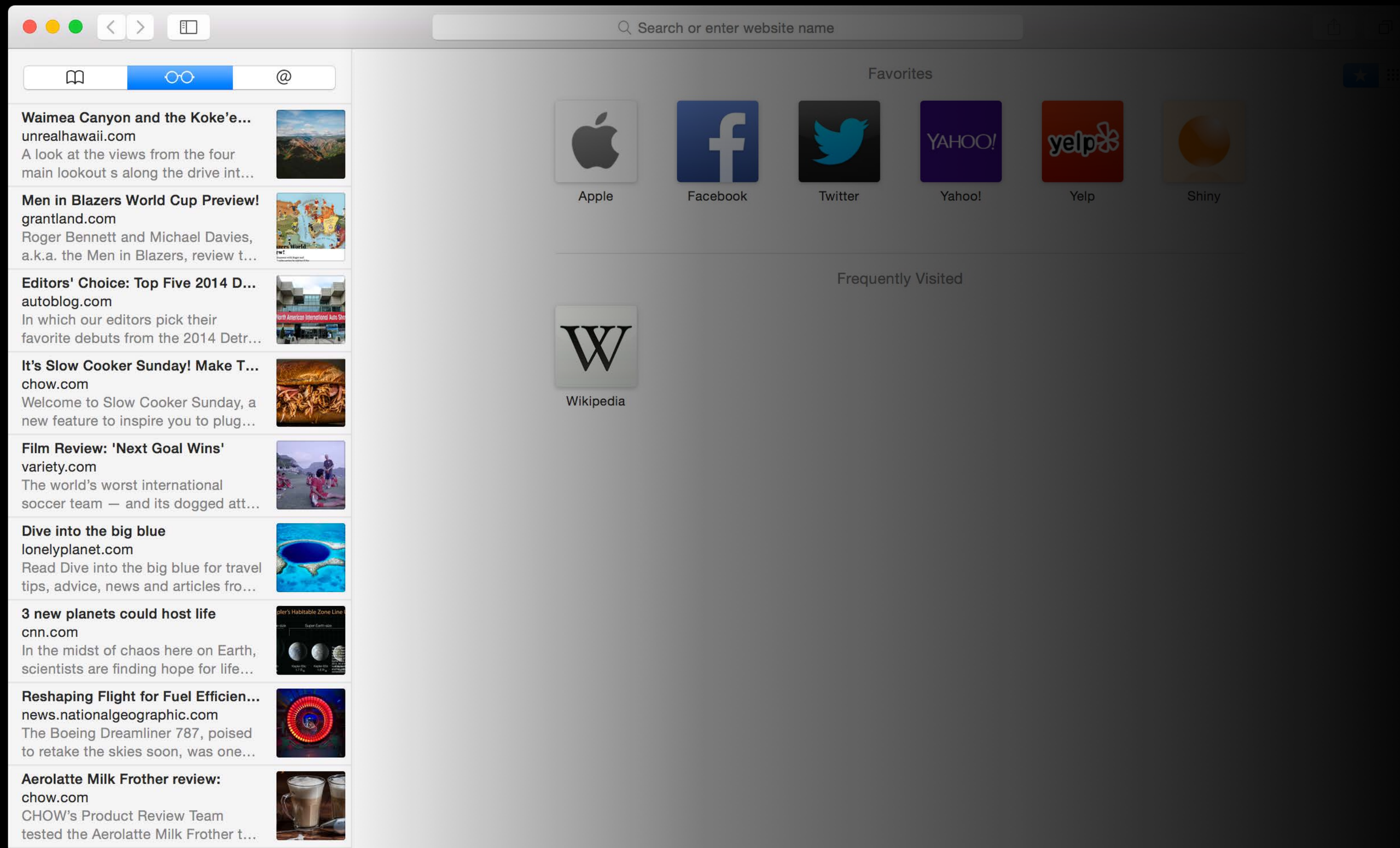
Just a decade ago, JavaScript – the programming language used to drive web page interactions – was thought to be too slow for serious application deve...



# Your Site in Shared Links



# Reading List



# Reading List

Adding from your iOS app

```
@import SafariServices;
```

```
NSURL *URL = [NSURL URLWithString:@"https://webkit.org/blog/..."];
```

```
[SSReadingList defaultReadingList] addReadingListItemWithURL:URL  
  title:@"Introducing the WebKit FTL JIT"  
  previewText:@"Just a decade ago, JavaScript – the programming language..."  
  error:NULL];
```

# Expressing Intent on the Web

# Reading List

Better titles and descriptions through metadata

```
<head>  
  <title>Surfin' Safari – Archive » Introducing the WebKit FTL JIT</title>  
</head>
```

# Reading List

## Better titles and descriptions through metadata

```
<head>  
  <title>Surfin' Safari – Archive » Introducing the WebKit FTL JIT</title>  
  
  <meta property="og:title"  
    content="Introducing the WebKit FTL JIT" />  
  <meta property="og:description"  
    content="Just a decade ago, JavaScript – the programming..." />  
</head>
```

# Reading List

## Better titles and descriptions through metadata

```
<head>
  <title>Surfin' Safari – Archive » Introducing the WebKit FTL JIT</title>

  <meta property="og:title"
        content="Introducing the WebKit FTL JIT" />
  <meta property="og:description"
        content="Just a decade ago, JavaScript – the programming..." />
</head>
```



### Surfin' Safari

#### Introducing the WebKit FTL JIT

Just a decade ago, JavaScript – the programming language used to drive web page interactions – was thought to be too slow for serious application deve...

# Reading List

Better titles and descriptions through metadata

```
<head>  
  <meta name="description"  
        content="Just a decade ago, JavaScript – the programming...">  
</head>
```



## Surfin' Safari

Introducing the WebKit FTL JIT

Just a decade ago, JavaScript – the programming language used to drive web page interactions – was thought to be too slow for serious application deve...



# Metadata Standards

## Resources

OpenGraph

<http://ogp.me>

The Dublin Core<sup>®</sup> Metadata Initiative

<http://dublincore.org>

schema.org

<http://schema.org>



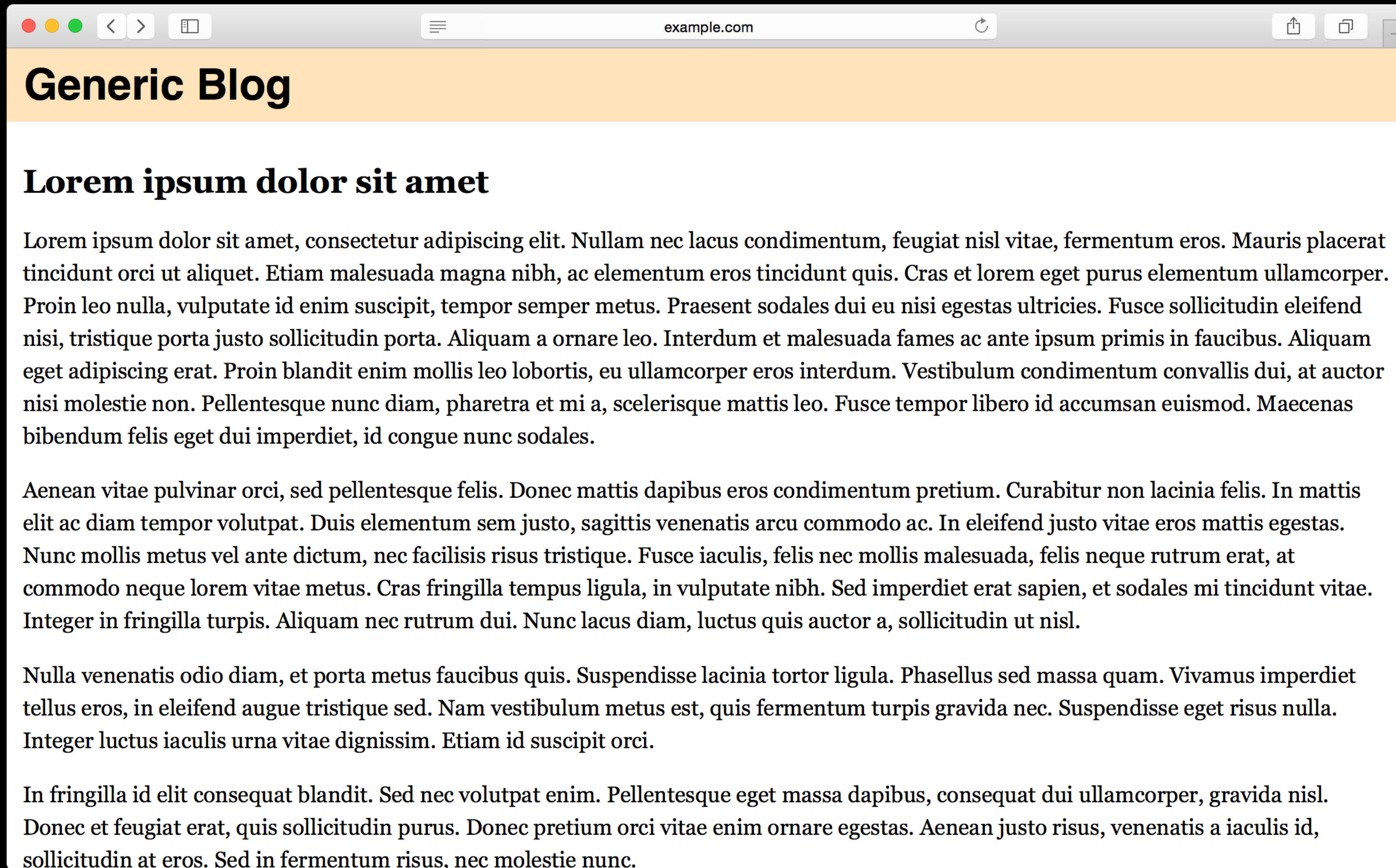


# The Sticky Mobile Website

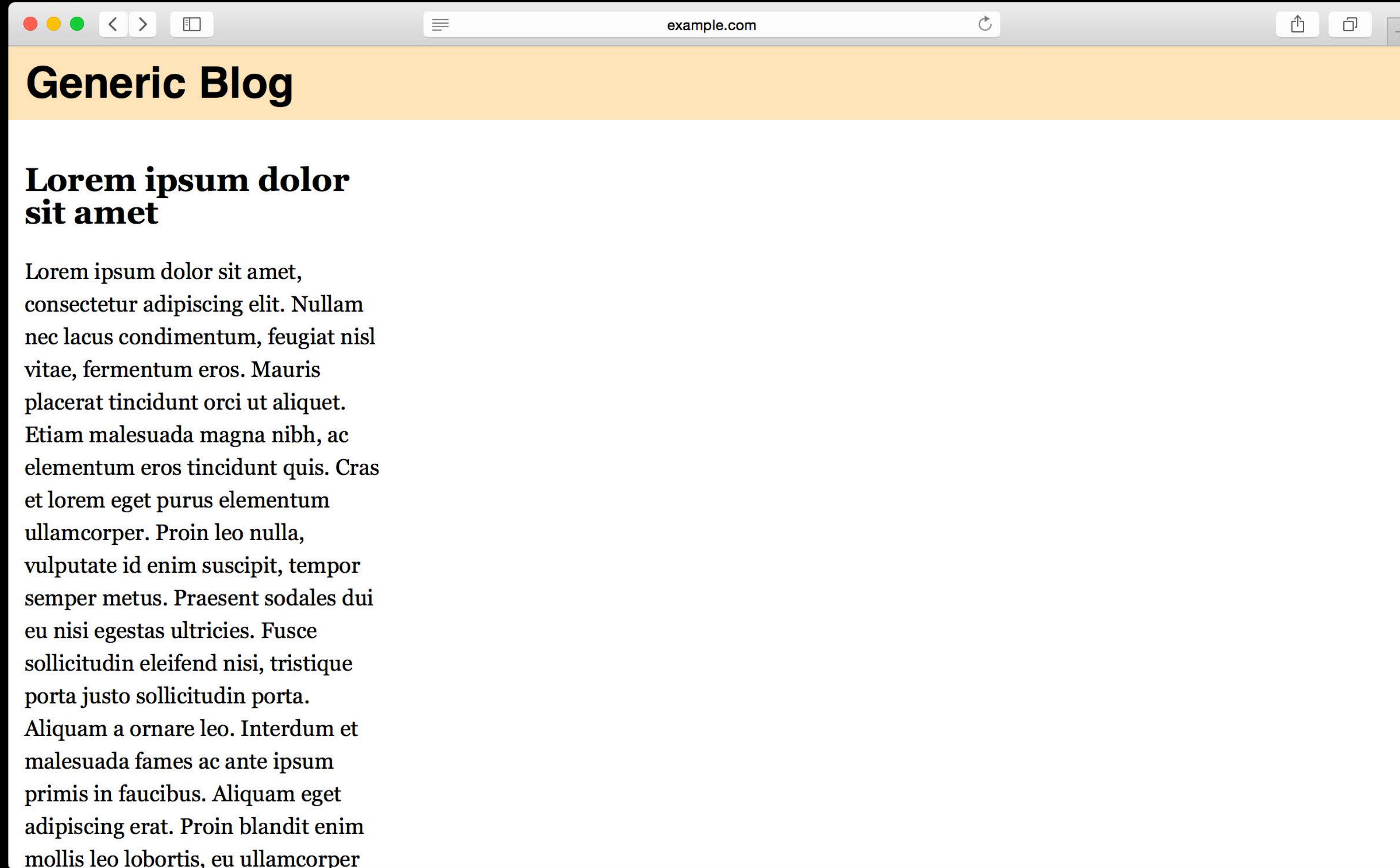
# The Sticky Mobile Website



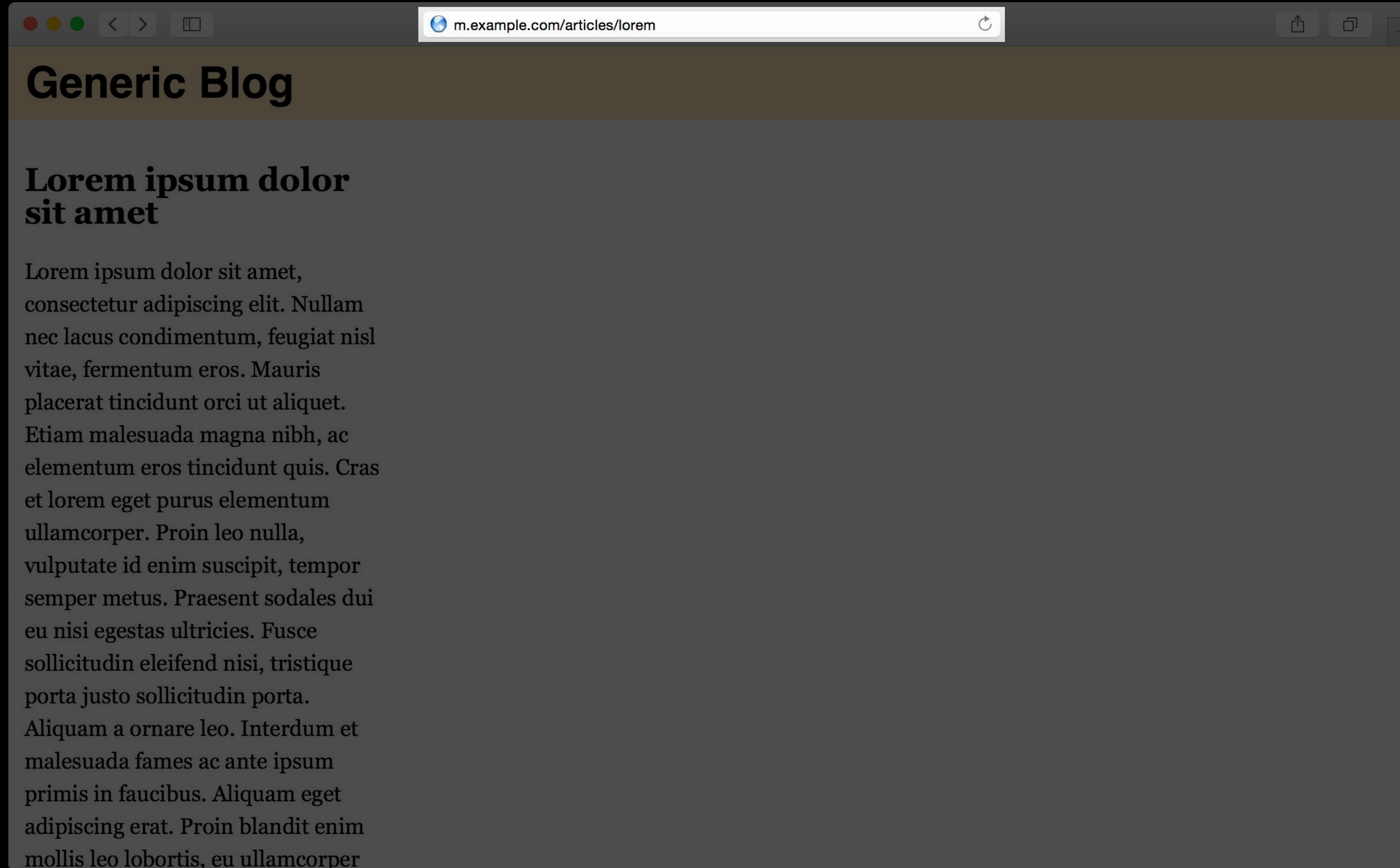
# The Sticky Mobile Website



# The Sticky Mobile Website

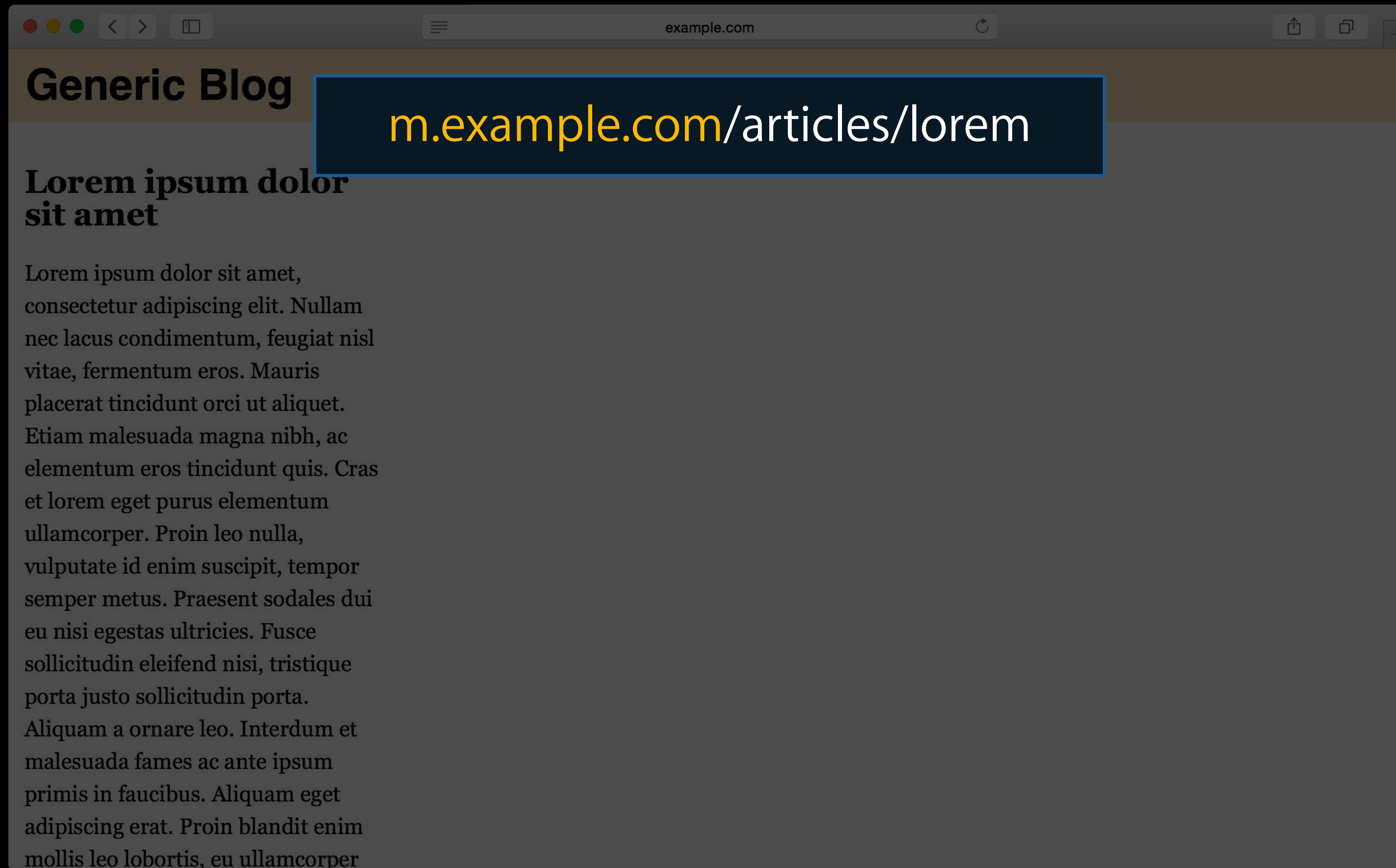


# The Sticky Mobile Website





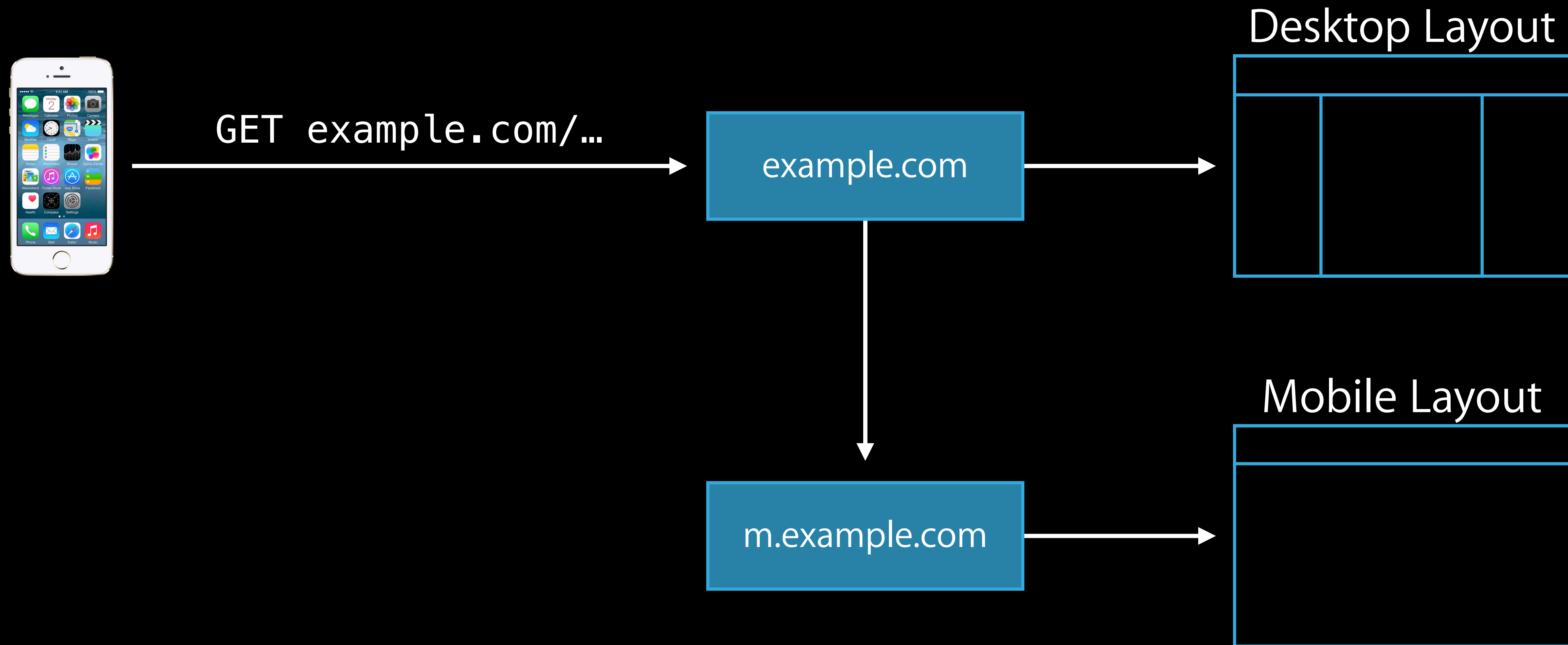
# The Sticky Mobile Website



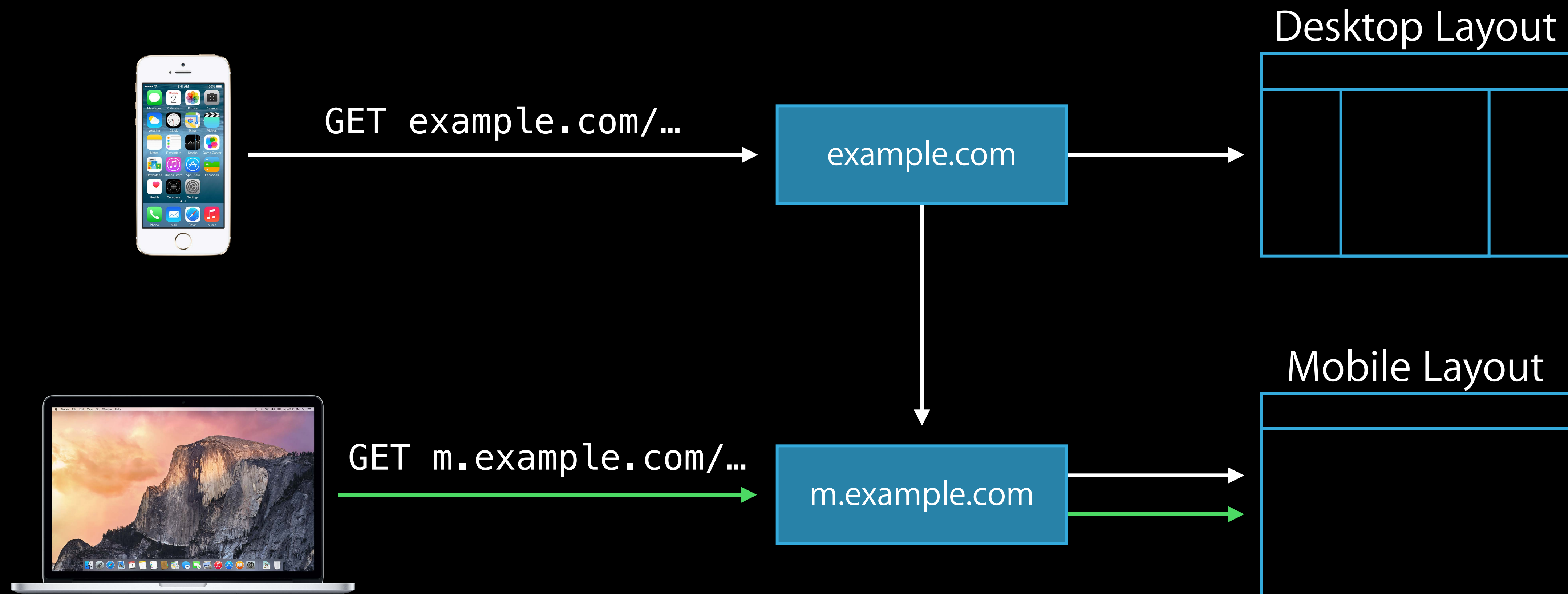
# The Sticky Mobile Website



# The Sticky Mobile Website

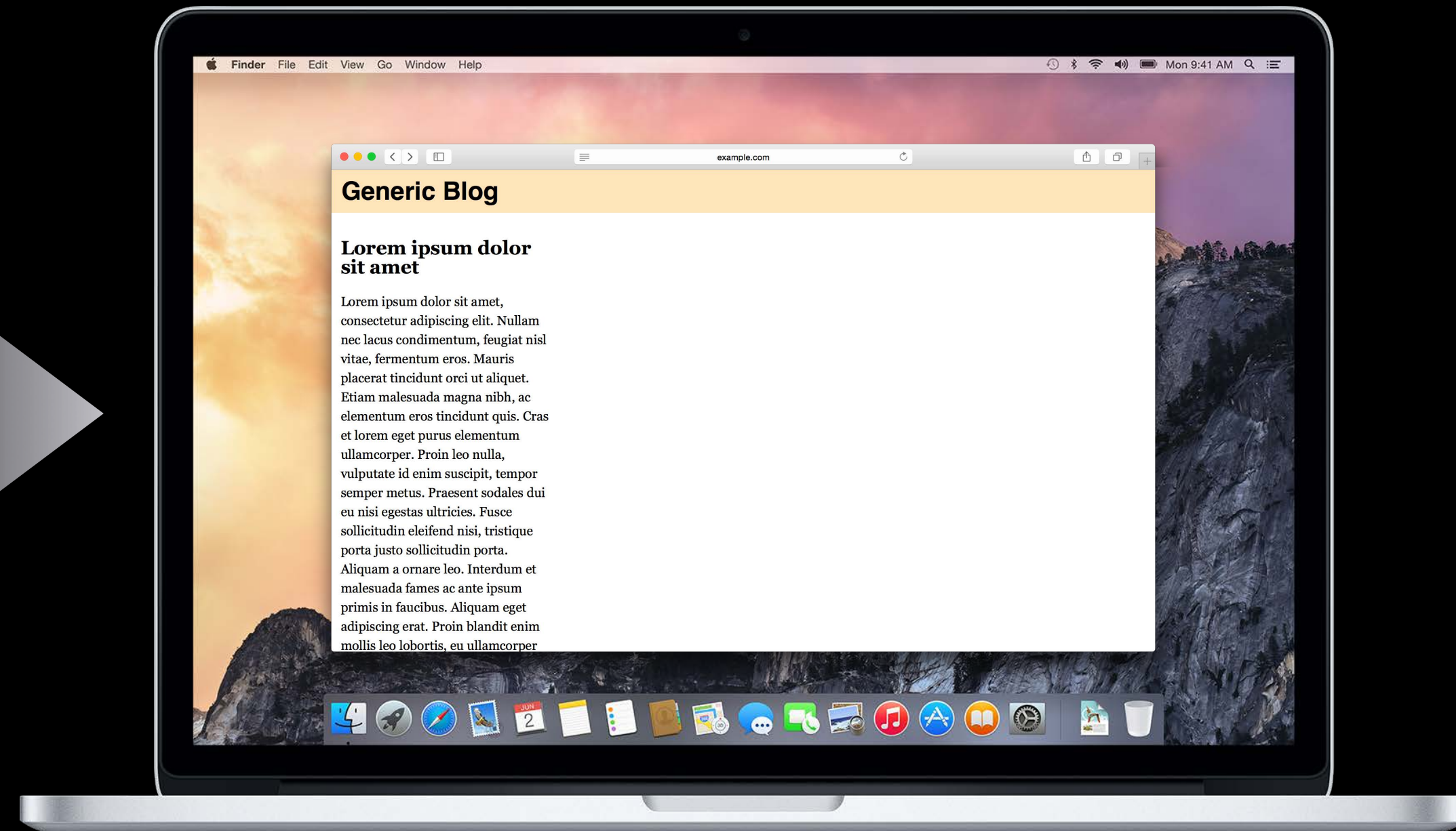


# The Sticky Mobile Website

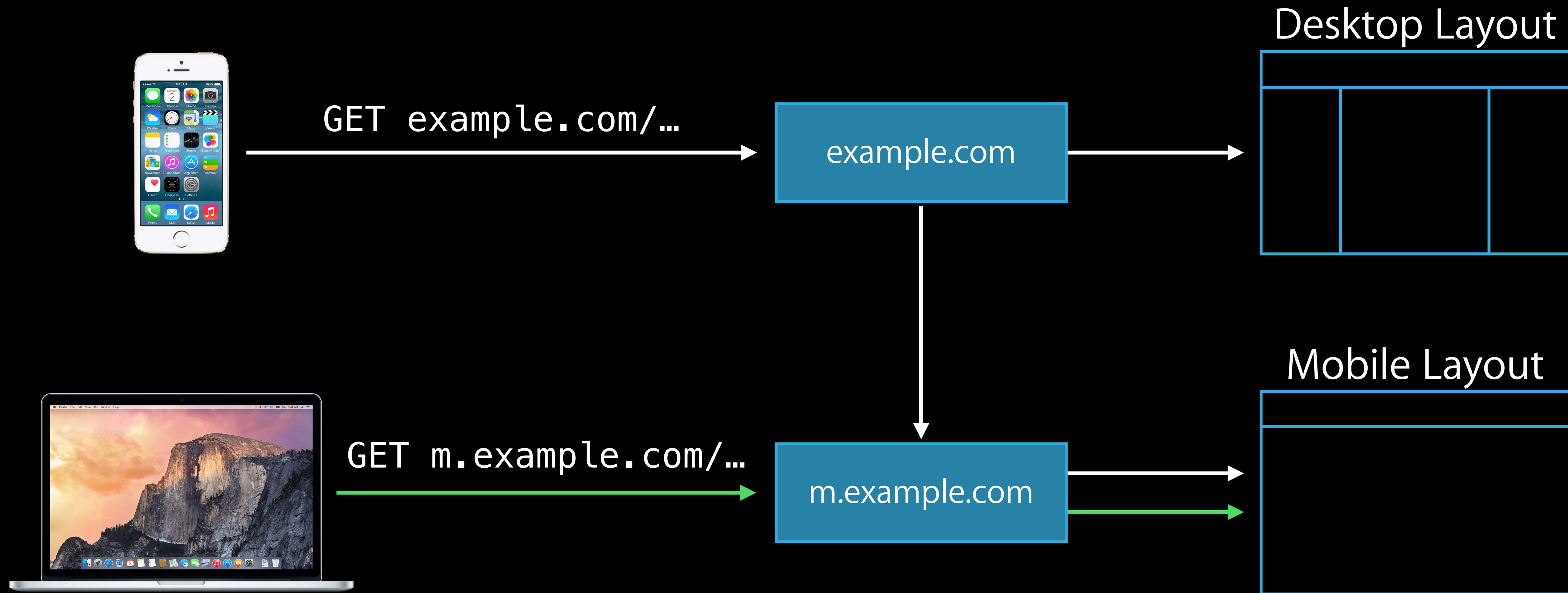


# The Sticky Mobile Website

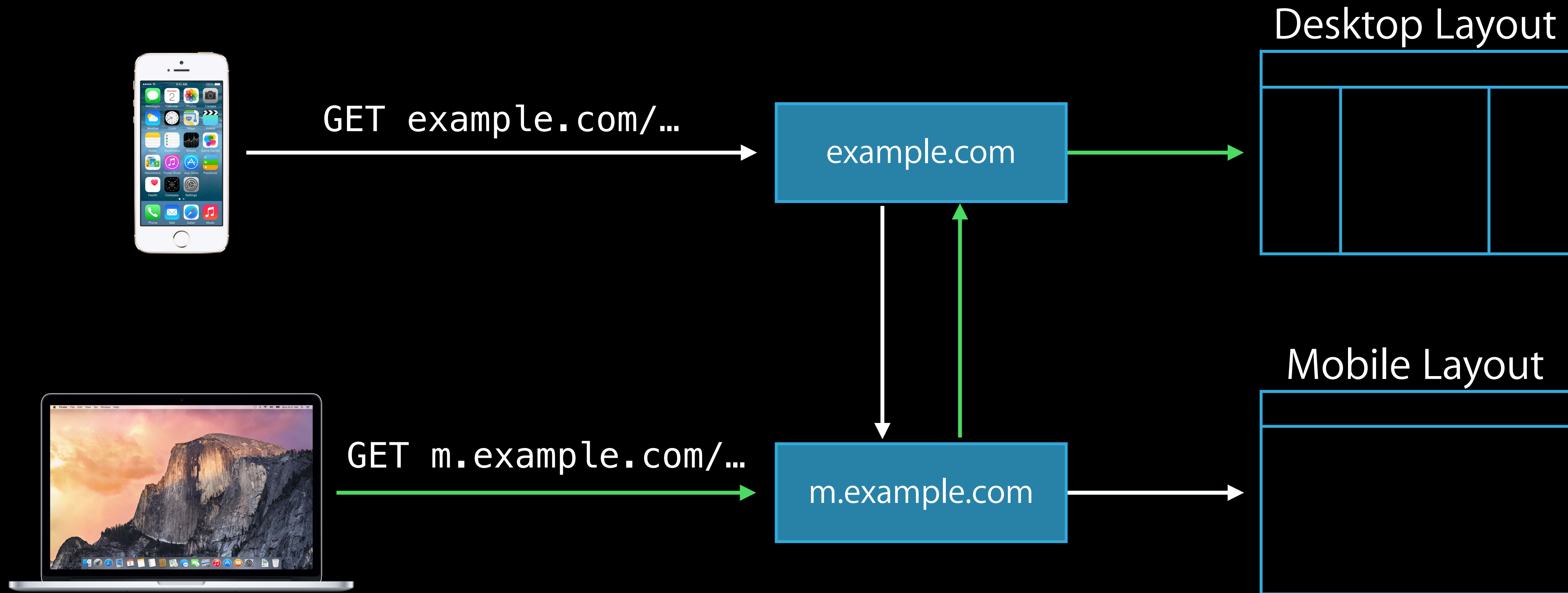
Relevant to Handoff



# The Sticky Mobile Website

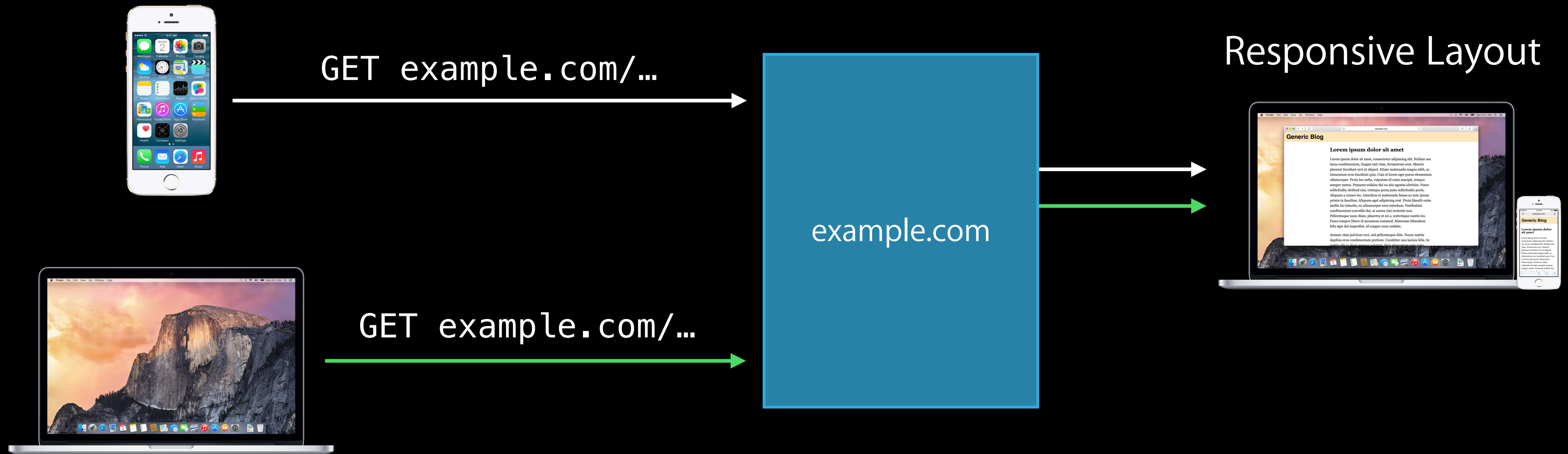


# The Sticky Mobile Website



# Responsive Design

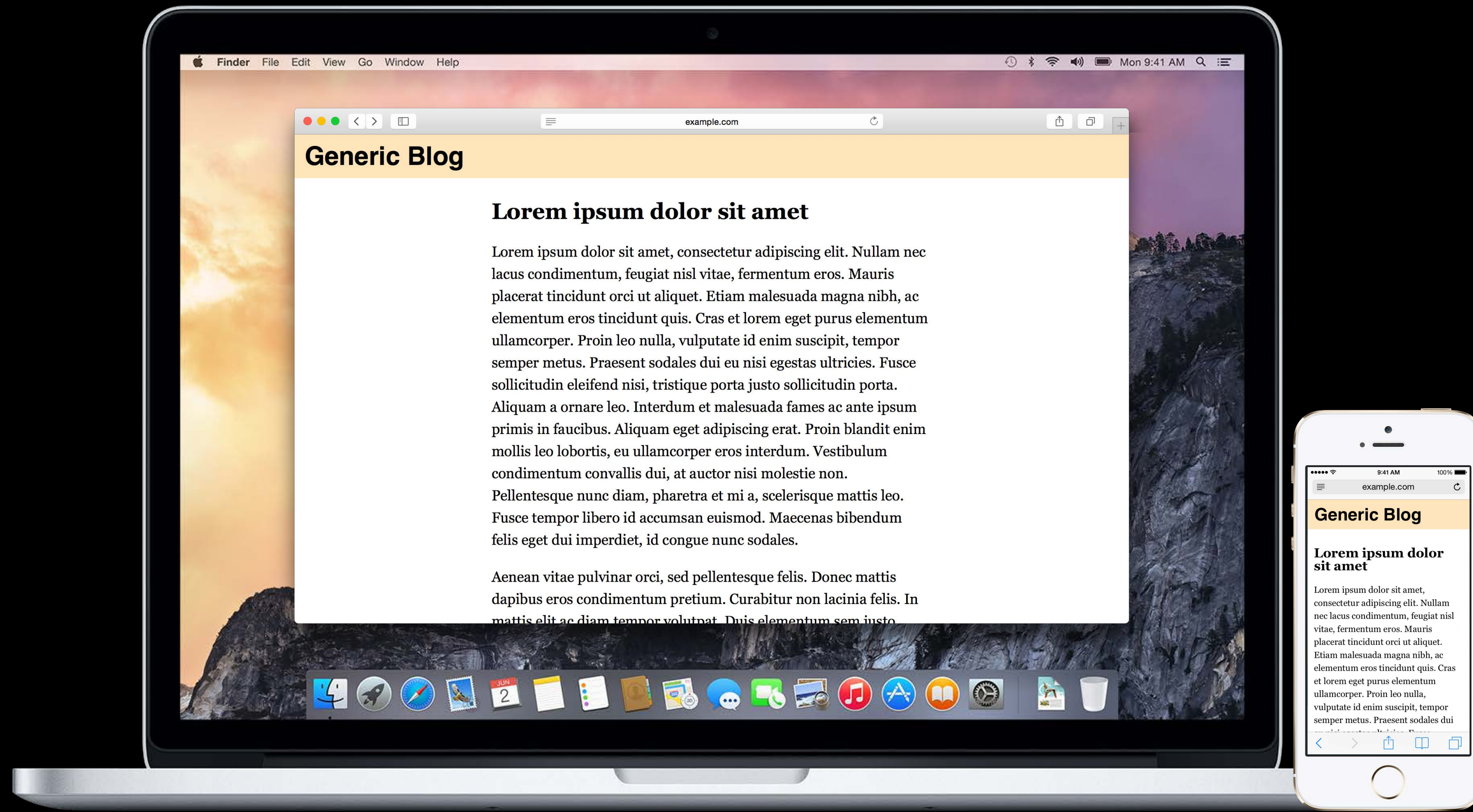
One site for everyone





# Responsive Design

One site for everyone



# Related Sessions

- 
- Designing Responsive Web Experiences Marina Friday 10:15AM
-





# Related Sessions

- 
- Advanced Media for the Web Nob Hill Tuesday 11:30AM
-



**Continuity** |ˌkæntɪn(ə)ˈwɪtē|

noun (pl. continuities)

the **unbroken** and **consistent** existence or operation of something over a period of time

# Summary



- Use Safari credentials in your apps
- Test Safari AutoFill with your website
- Use the autoComplete attribute
- Deploy high-quality website icons
- Express intent with metadata
- Unify mobile and desktop websites



# More Information

Evangelism

[evangelism@apple.com](mailto:evangelism@apple.com)

Developer Technical Support

<http://developer.apple.com/contact>

Apple Developer Forums

<http://devforums.apple.com>

# Related Sessions

- 
- Advanced Media for the Web Nob Hill Tuesday 11:30AM
  - Keychain and Authentication with Touch ID Nob Hill Wednesday 10:15AM
  - Adopting Handoff on iOS and OS X Mission Wednesday 2:00 PM
  - Designing Responsive Web Experiences Marina Friday 10:15AM
-

# Labs

- 
- Security Lab Core OS Lab B Wednesday 11:30AM
  - Safari and WebKit Lab Media Lab B Wednesday 4:30PM
  - Safari and WebKit Lab Media Lab B Thursday 2:00PM
  - Security and Privacy Lab Core OS Lab B Thursday 3:15PM
-

 WWDC14