

Extending Your Apps for Enterprise and Education Use

Session 301

Dave Rahardja

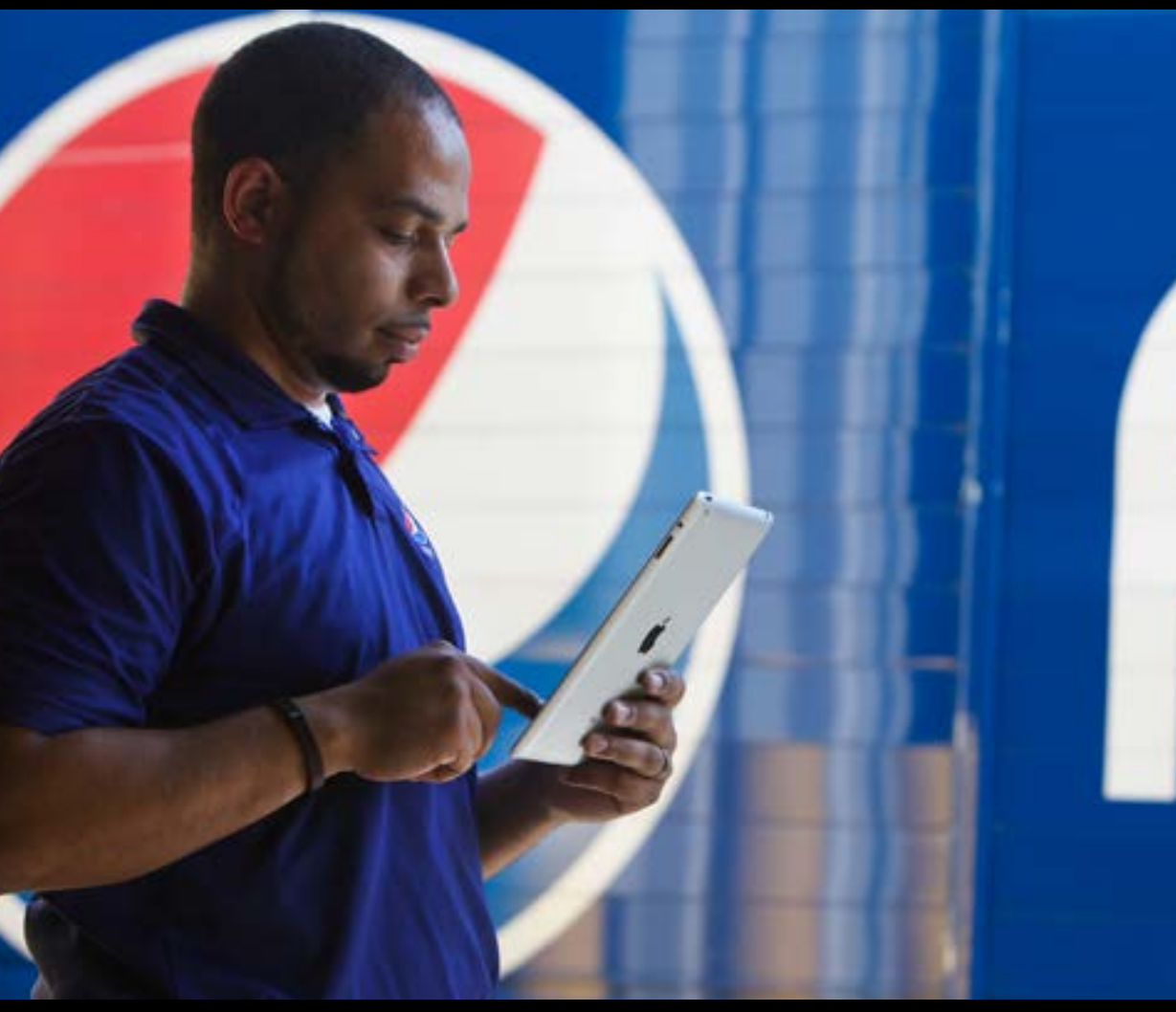
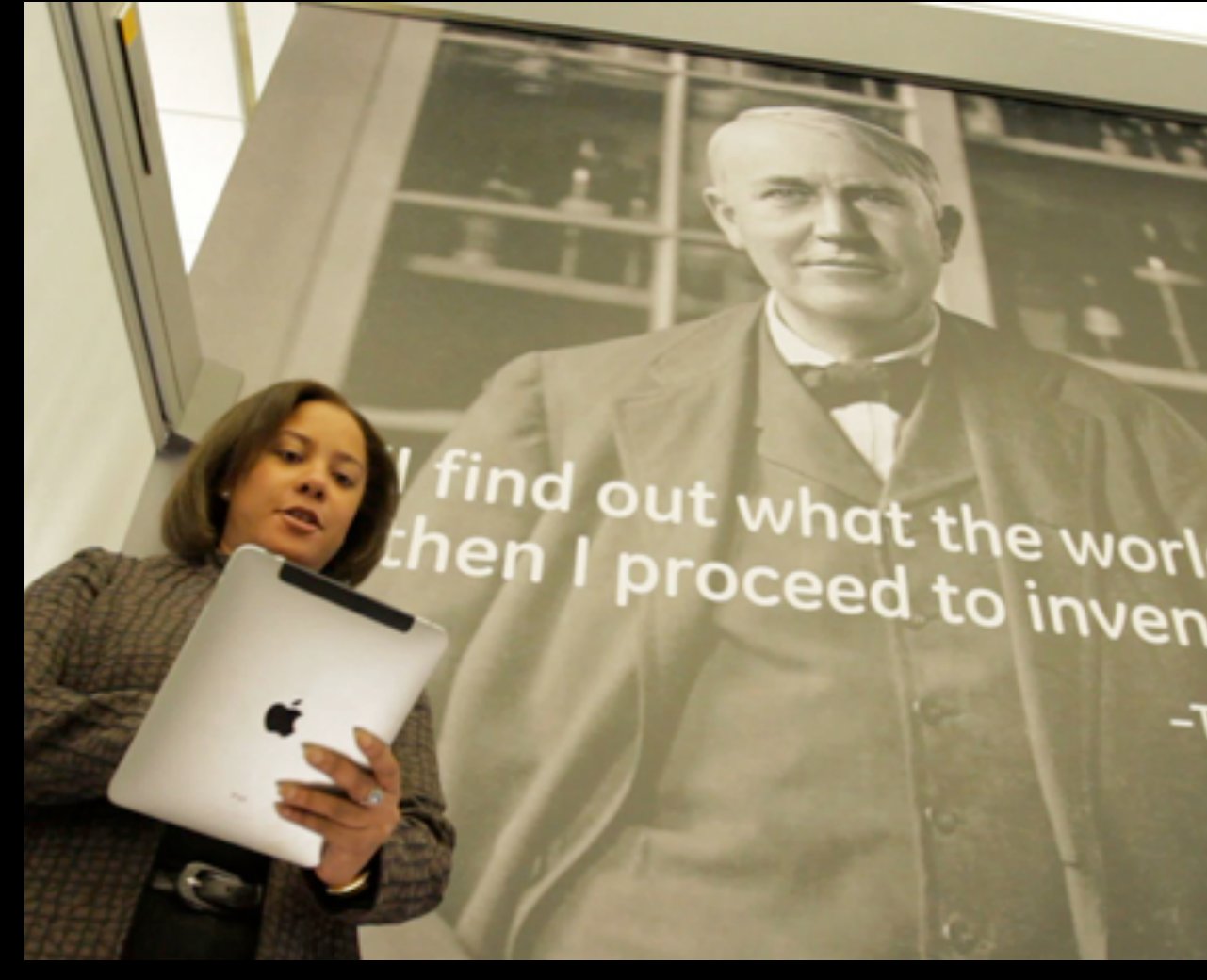
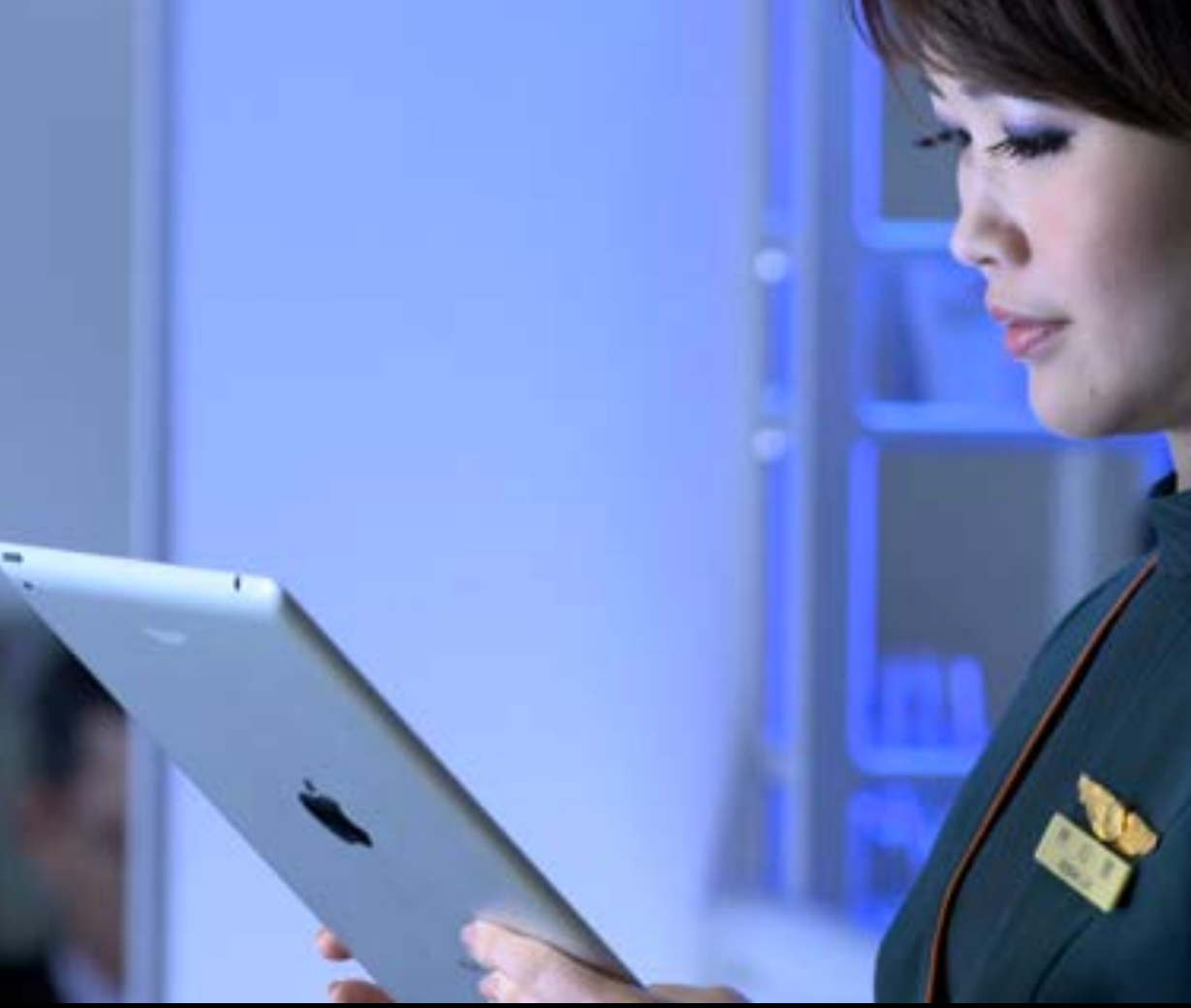
iOS Device Management Engineer

Victor Alexander

iOS Data Access Engineer

These are confidential sessions—please refrain from streaming, blogging, or taking pictures





Apps in Enterprise



Authentication

Networking

Data Security

Management

Authentication

Authentication

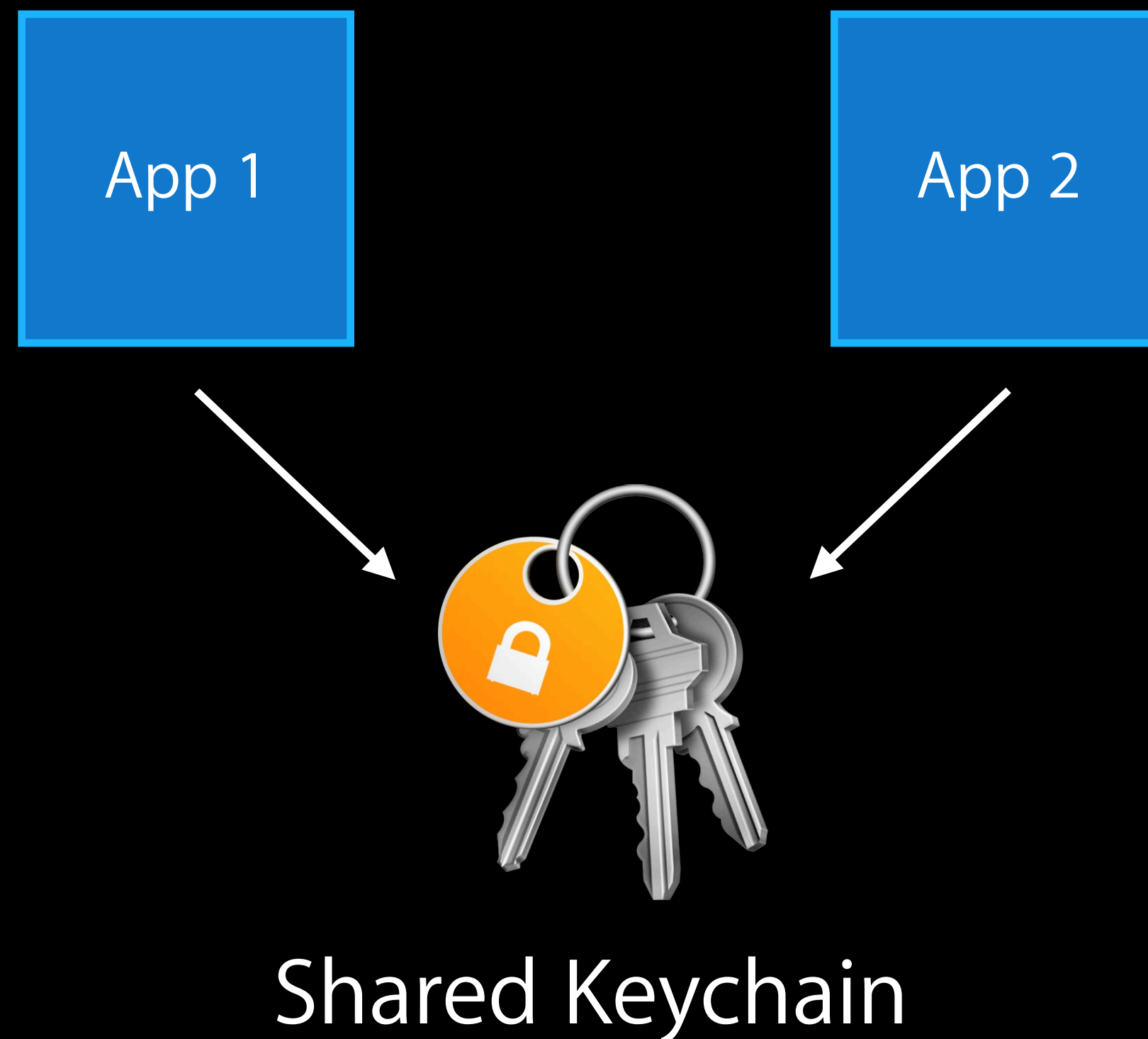
Built into iOS

- Shared keychain—great for apps from same developer

Authentication

Built into iOS

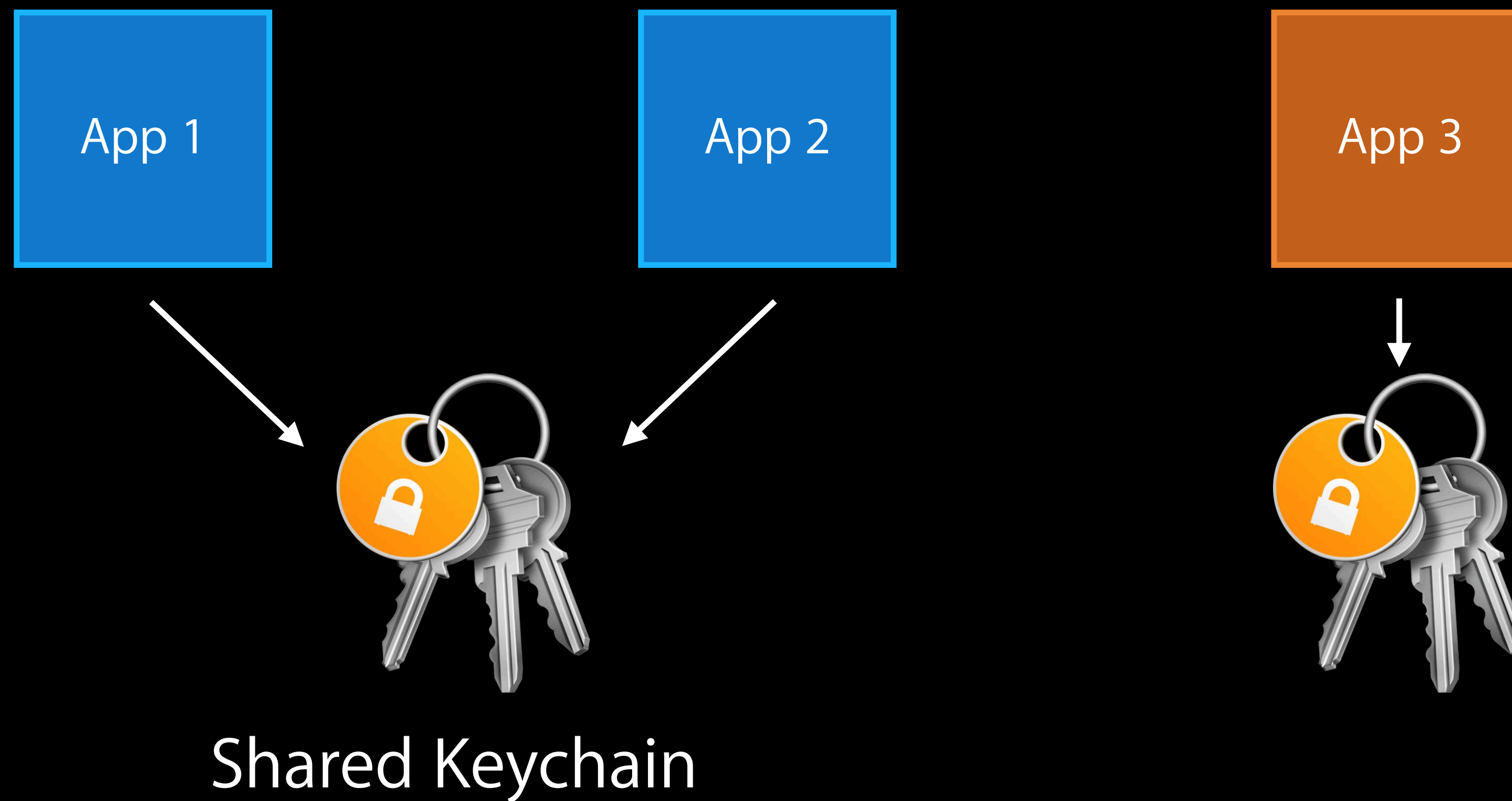
- Shared keychain—great for apps from same developer



Authentication

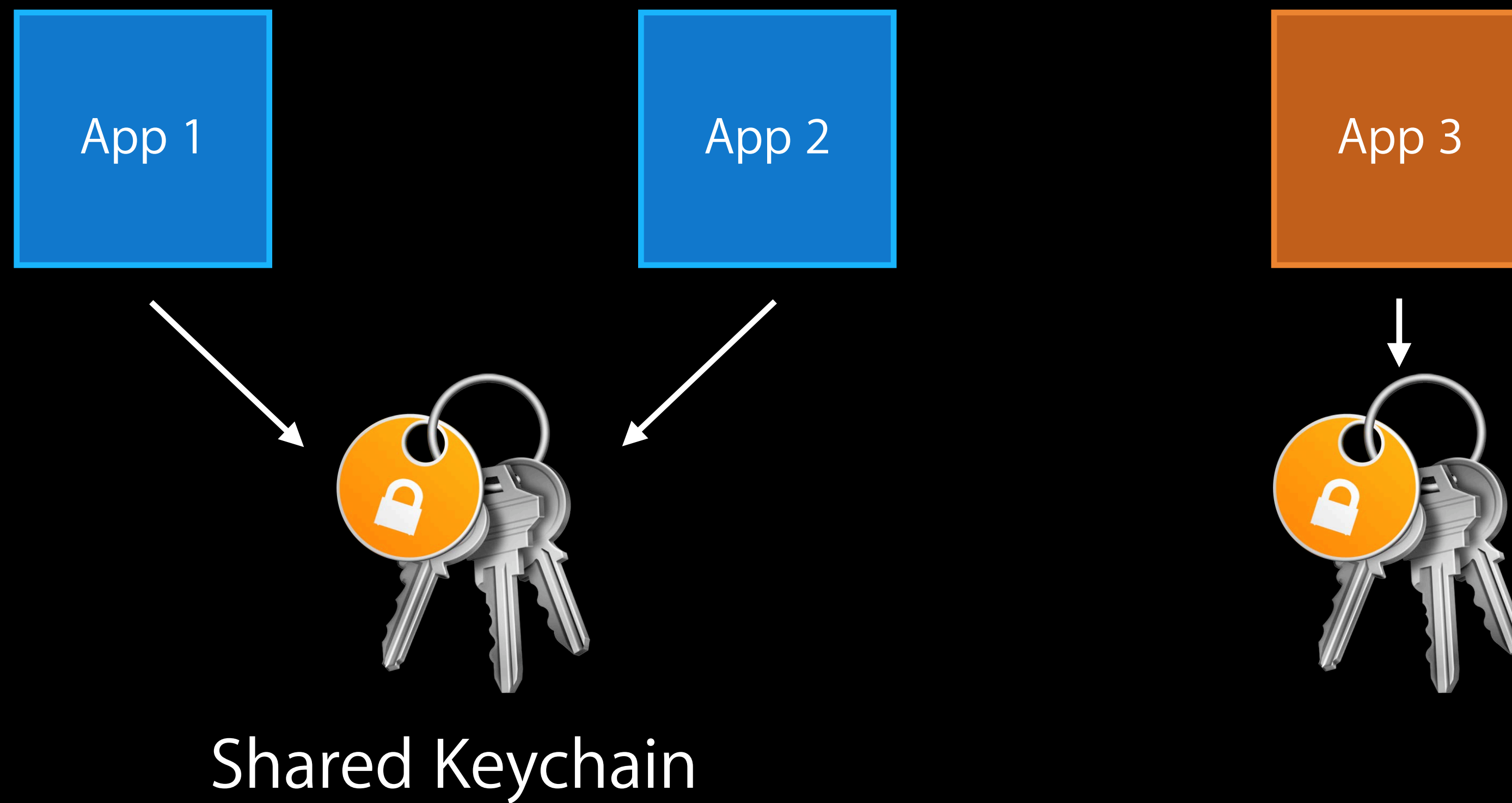
Built into iOS

- Shared keychain—great for apps from same developer



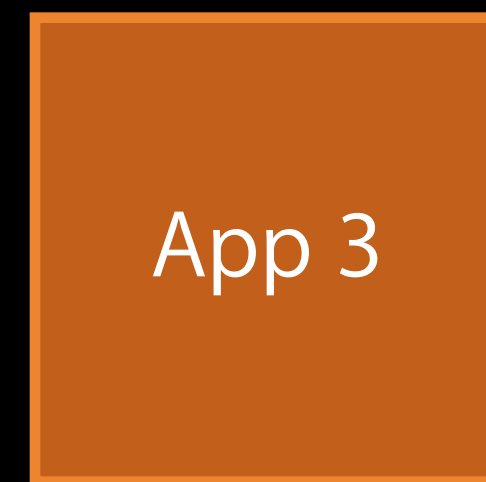
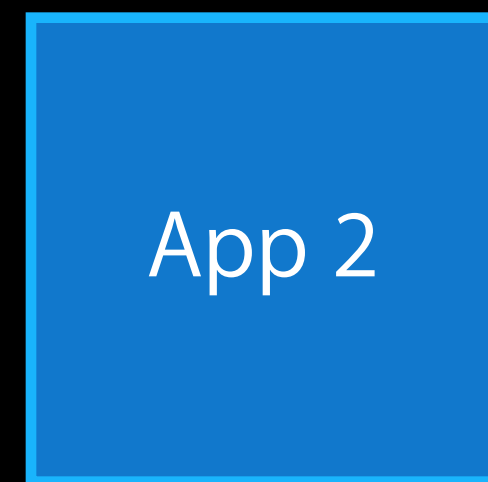
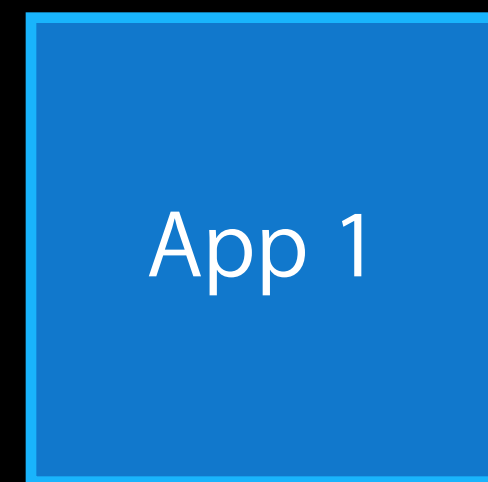
Single Sign-On

Built into iOS



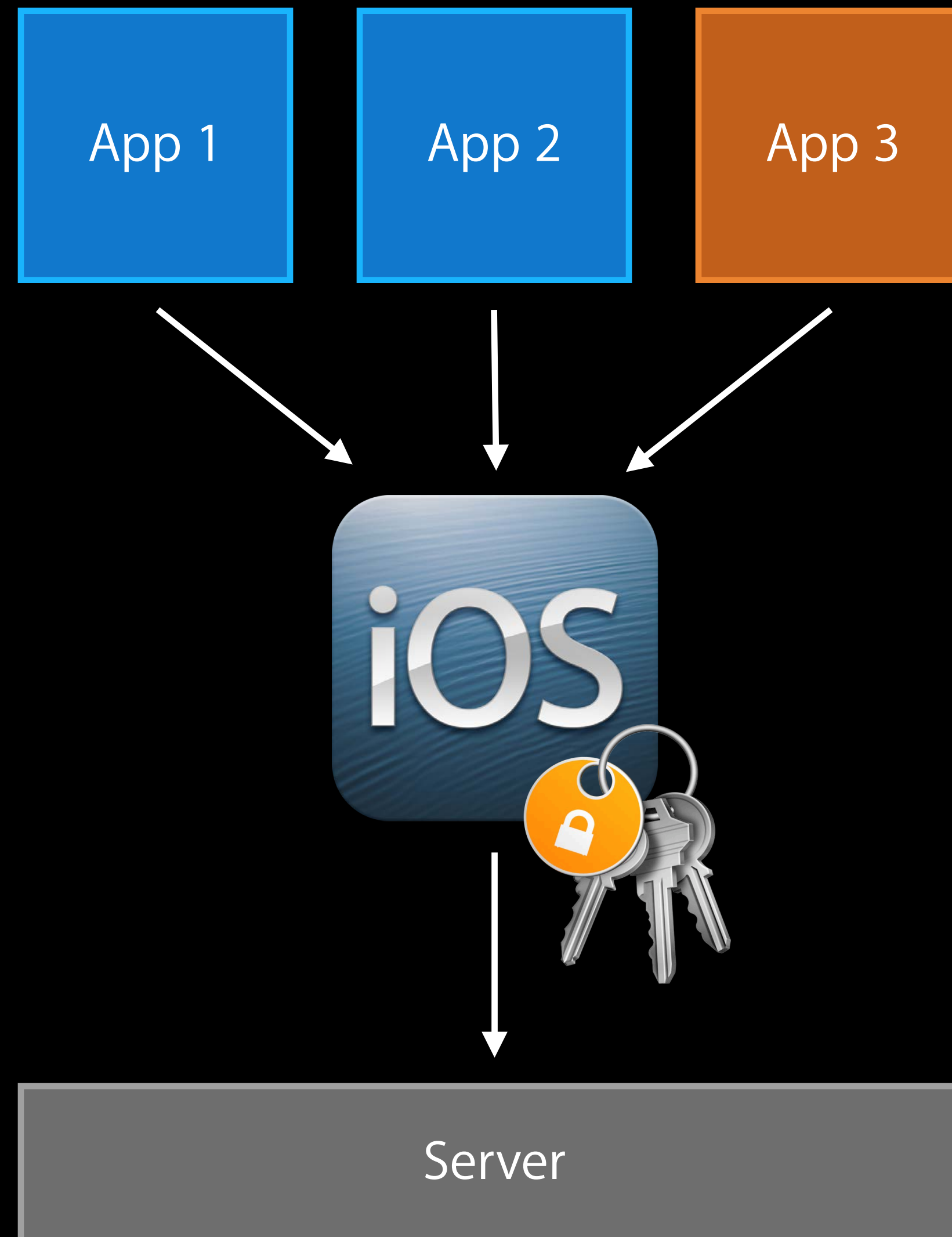
Single Sign-On

Built into iOS



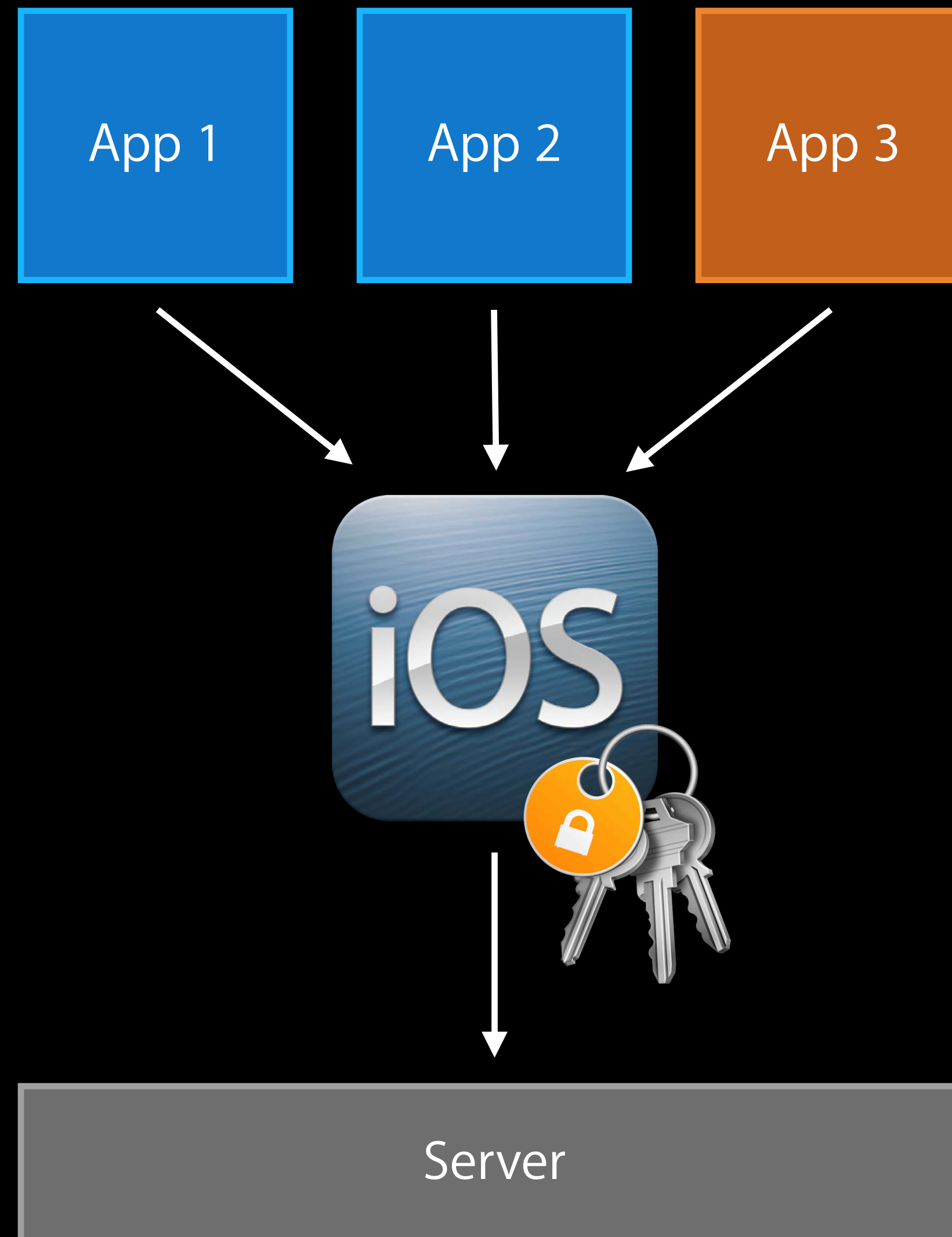
Single Sign-On

Built into iOS



Single Sign-On

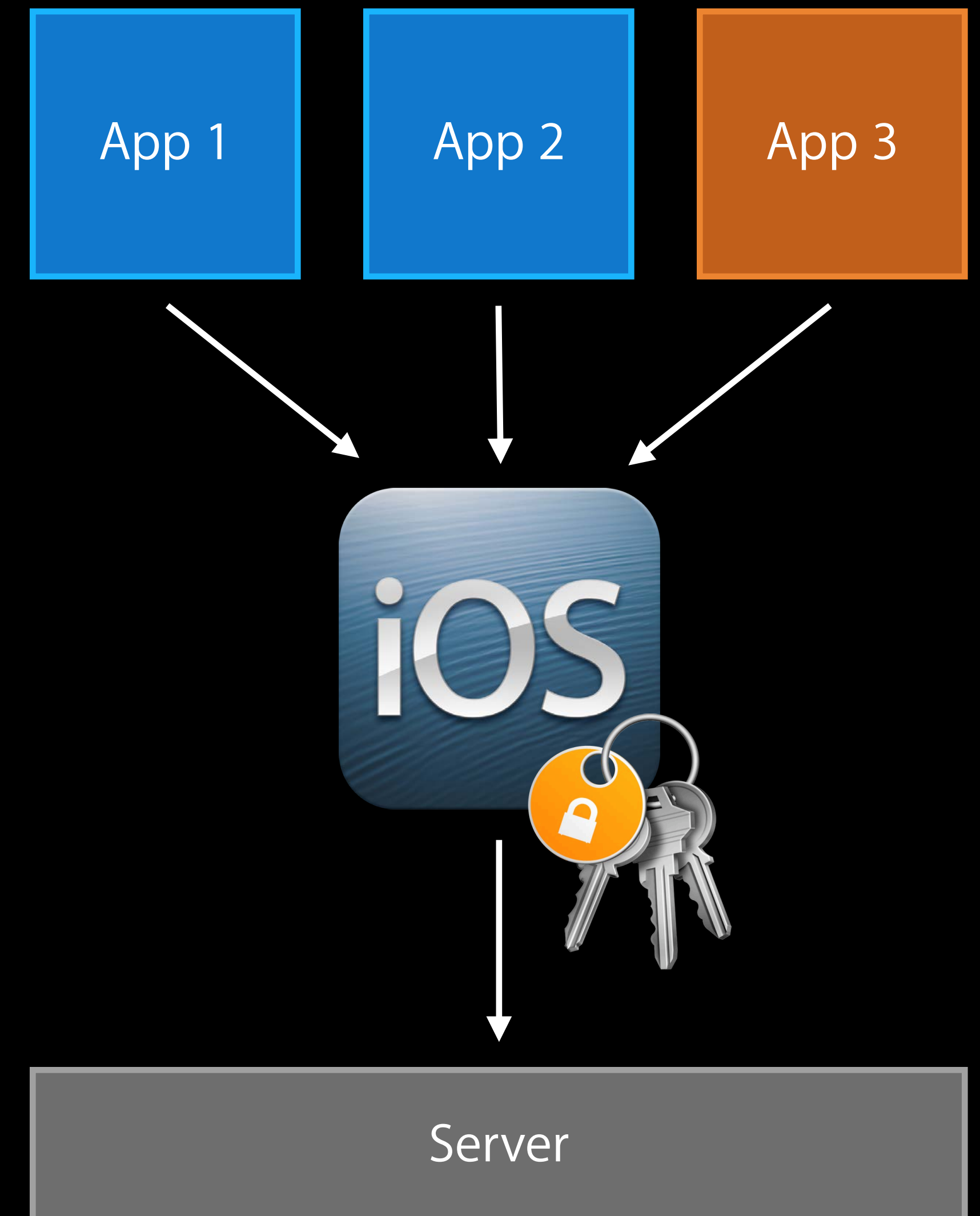
Built into iOS



Single Sign-On

Built into iOS

- Kerberos
- Grant access to specific apps only
- Password not stored inside apps



Single Sign-On

App developers

- No code changes (if you use NSURLConnection)
- Use NSURLConnection or NSURLSession
- No authentication callback to your app
~~connection:willSendRequestForAuthenticationChallenge~~

Single Sign-On

IT integrators

- Configuration profile
 - Define account
 - Define apps
 - Define URL prefixes
- Works with third-party apps and Safari

Authentication

Networking

Data Security

Management

Networking

Networking

Built into iOS

- SSL/TLS
- Proxies—per-SSID or global
- Wi-Fi
 - 802.1X
 - WPA/WPA2
- VPN

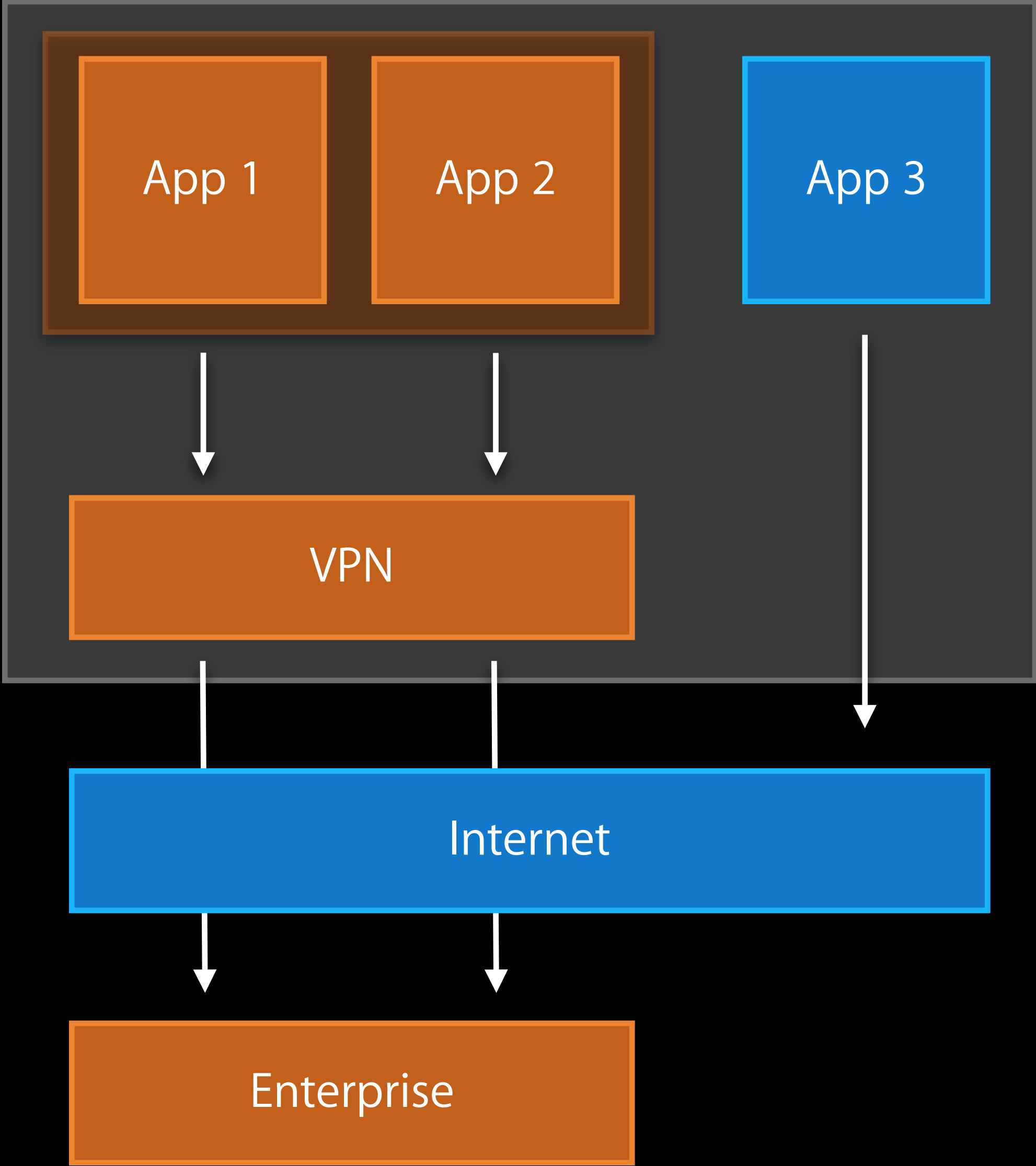
Per-App VPN

Built into iOS



Per-App VPN

Built into iOS



Per-App VPN

Built into iOS

Per-App VPN

Built into iOS

- Limit VPN access to specific apps
- Security + privacy

Per-App VPN

App developers

Per-App VPN

App developers

- No code changes required

Per-App VPN

IT integrators

- VPN plugin support
- Configured using MDM

Networking

App developers

- Use `NSURLConnection/NSURLSession`
- Be aware of proxies if you use lower level API
- Be conscious of cellular data usage
`URLRequest.allowsCellularAccess = NO;`

Authentication

Networking

Data Security

Management

Data Security

Data Security

Built into iOS

- Data protection—protects data at rest with passcode

Data Security

Built into iOS

- Data protection—protects data at rest with passcode

None

Complete with
Authorization

Complete

Data Security

Built into iOS

- Data protection—protects data at rest with passcode



Data Security

Built into iOS

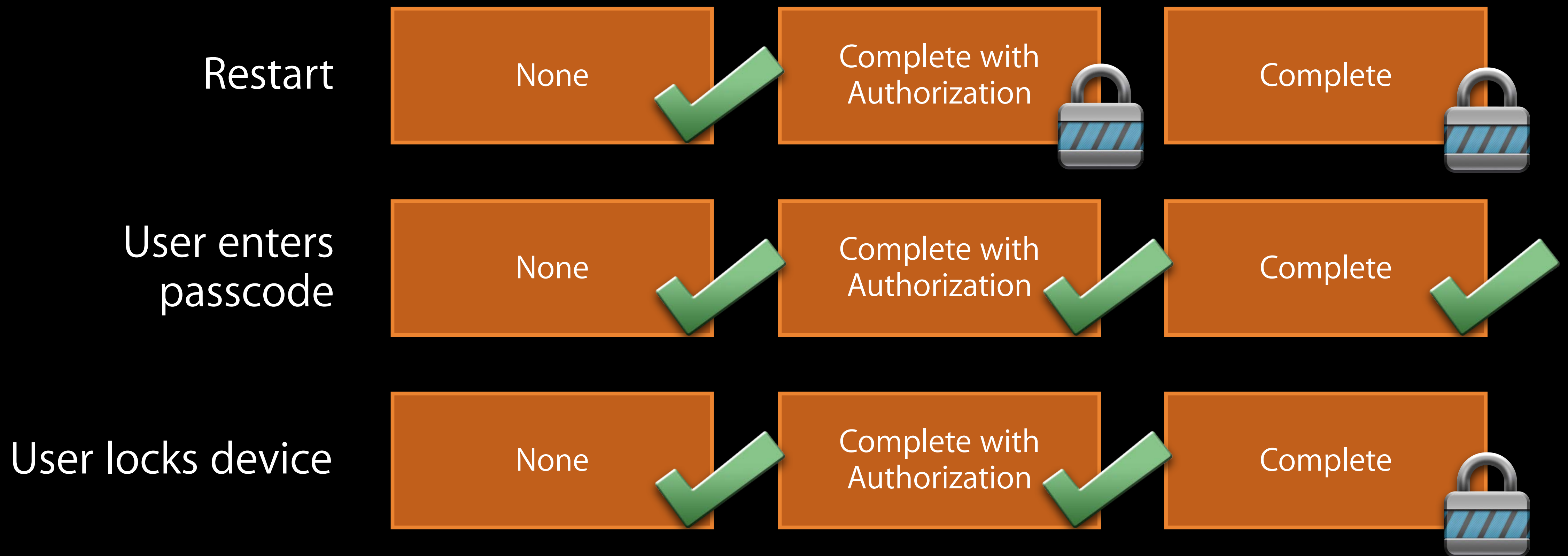
- Data protection—protects data at rest with passcode



Data Security

Built into iOS

- Data protection—protects data at rest with passcode



Data Security

Built into iOS

- Data protection—protects data at rest with passcode
- Installed apps automatically protected with `NSFileProtectionCompleteUntilFirstUserAuthentication`

Data Security

Built into iOS

- Data protection—protects data at rest with passcode
- Installed apps automatically protected with `NSFileProtectionCompleteUntilFirstUserAuthentication`



Data Security

Built into iOS

- Data protection—protects data at rest with passcode
- Installed apps automatically protected with `NSFileProtectionCompleteUntilFirstUserAuthentication`
- FIPS 140-2 certification



Data Security

App developers

Data Security

App developers

- Use `NSFileProtectionComplete` when appropriate

Data Security

App developers

- Use `NSFileProtectionComplete` when appropriate
- Write data to buffer if device is locked

Data Security

App developers

- Use `NSFileProtectionComplete` when appropriate
- Write data to buffer if device is locked
- Keychain data protection keep secrets on one device
`kSecAttrAccessibleAfterFirstUnlockThisDeviceOnly`

Data Security

App developers

- Use `NSFileProtectionComplete` when appropriate
- Write data to buffer if device is locked
- Keychain data protection keep secrets on one device
`kSecAttrAccessibleAfterFirstUnlockThisDeviceOnly`
- iCloud document sync

Authentication

Networking

Data Security

Management

Management

Management

Built into iOS

- Mobile device management
 - Passcode and security policies
 - Account installation
 - Remote erase
 - App installation and removal



Managed Open In

Built into iOS



Managed Open In

Built into iOS



- Prevent unintentional data movement

Managed Open In

Built into iOS



- Prevent unintentional data movement
- Managed data stay in managed apps and accounts

Demo

Managed open in

App Configuration and Feedback

Built into iOS



App Configuration and Feedback

Built into iOS



- Configuration
 - Send configuration dictionary to managed app
 - Appears in NSUserDefaults

App Configuration and Feedback

Built into iOS



- Configuration
 - Send configuration dictionary to managed app
 - Appears in NSUserDefaults
- Feedback
 - App writes to NSUserDefaults
 - Read feedback dictionary from managed app

App Configuration and Feedback

App developers

- Read configuration dictionary using
[[NSUserDefaults standardUserDefaults]
objectForKey:@"com.apple.configuration.managed"]
- Write feedback dictionary to NSUserDefaults key
com.apple.feedback.managed
- Listen for changes using NSUserDefaultsDidChangeNotification
- Your app might not be running

Demo

App configuration and feedback

App Configuration

```
- (void) viewDidLoad {
    [super viewDidLoad];

    //...

    [[NSNotificationCenter defaultCenter]
     addObserverForName:NSUserDefaultsDidChangeNotification
                    object:nil
                    queue:[NSOperationQueue mainQueue]
                    usingBlock:^(NSNotification *note) {
        [self readDefaultsValues];
    }];

    [self readDefaultsValues];
}
```


App Configuration

```
- (void)readDefaultsValues {
    NSDictionary *serverConfig =
        [[NSUserDefaults standardUserDefaults]
         dictionaryForKey:@"com.apple.configuration.managed"];

    NSString *serverURLString = serverConfig[@"serverURL"];
    if (serverURLString &&
        [serverURLString isKindOfClass:[NSString class]]) {
        self.serverURLTextField.text = serverURLString;
    } else {
        self.serverURLTextField.text = @"http://foo.bar";
    }

    //...
}
```

App Feedback

```
- (void)incrementSuccessCount {
    self.successCount += 1;

    //...

    NSMutableDictionary *feedback =
        [[[NSUserDefaults standardUserDefaults]
         dictionaryForKey:@"com.apple.feedback.managed"] mutableCopy];
    if (!feedback) feedback = [NSMutableDictionary dictionary];

    feedback[@"successCount"] = @(self.successCount);
    [[[NSUserDefaults standardUserDefaults]
     setObject:feedback forKey:@"com.apple.feedback.managed"];
}
```

App Configuration

App developers

App Configuration

App developers

- Settings and preferences
 - App UI settings
 - URLs
 - Disable iCloud document sync

App Configuration

App developers

- Settings and preferences
 - App UI settings
 - URLs
 - Disable iCloud document sync
- Stored as `NSFileProtectionNone`—no passwords or private keys!

App Configuration

App developers

- Settings and preferences
 - App UI settings
 - URLs
 - Disable iCloud document sync
- Stored as `NSFileProtectionNone`—no passwords or private keys!
- Document your dictionary keys

App Configuration

App developers

- Settings and preferences
 - App UI settings
 - URLs
 - Disable iCloud document sync
- Stored as `NSFileProtectionNone`—no passwords or private keys!
- Document your dictionary keys
- Validate all input types and values

App Configuration

App developers

- Settings and preferences
 - App UI settings
 - URLs
 - Disable iCloud document sync
- Stored as `NSFileProtectionNone`—no passwords or private keys!
- Document your dictionary keys
- Validate all input types and values
- Keep it small

App Configuration

App developers

- Settings and preferences
 - App UI settings
 - URLs
 - Disable iCloud document sync
- Stored as `NSFileProtectionNone`—no passwords or private keys!
- Document your dictionary keys
- Validate all input types and values
- Keep it small
- App might not be running

App Feedback

App developers

- Errors and usage statistics
- Aggregate, don't log
- Document your dictionary keys
- Keep it small
- App might not be running
- Respect privacy

App Configuration and Feedback

IT integrators

- Provide configuration when installing app
- Use feedback to detect server connection errors

Fonts

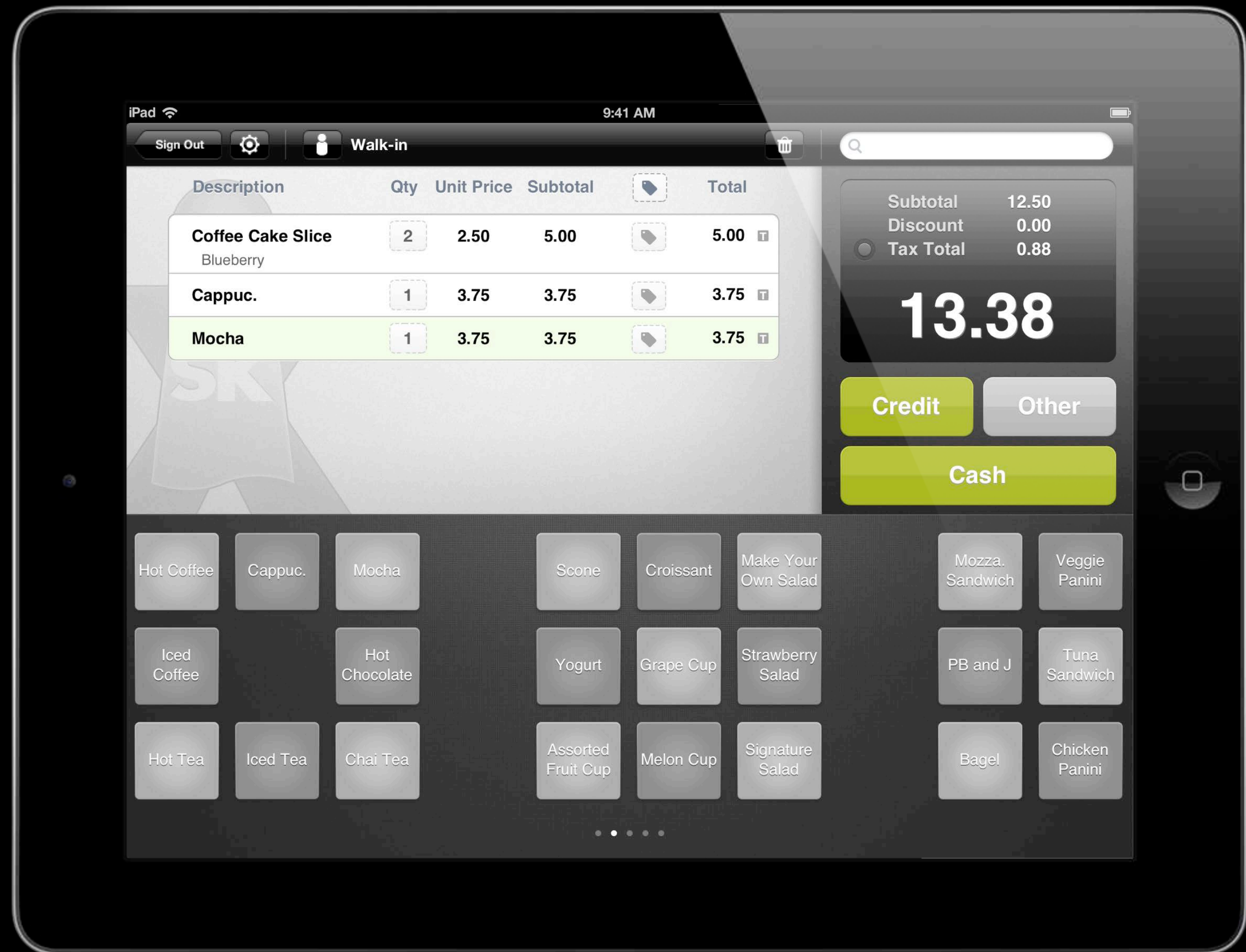
App developers

- Installed using configuration profiles
- Font list may change
- `kCTFontManagerRegisteredFontsChangedNotification`



Single App Mode

- MDM-controlled



App-Requested Single App Mode



App-Requested Single App Mode



- MDM-authorized

`UIAccessibilityRequestGuidedAccessSession()`

`UIAccessibilityIsGuidedAccessEnabled()`

`UIAccessibilityGuidedAccessStatusDidChangeNotification`

App Revocation

App developers

- App licenses can expire
- Expiry date available on receipt
- See session on receipts

Authentication

Networking

Data Security

Management

Authentication

Networking

Data Security

Management

Summary

Summary

- Build the next generation of enterprise apps

Summary

- Build the next generation of enterprise apps
- Use iOS 7 features

Summary

- Build the next generation of enterprise apps
- Use iOS 7 features
- Support app configuration and feedback

Summary

- Build the next generation of enterprise apps
- Use iOS 7 features
- Support app configuration and feedback
- Use `NSURLConnection` / `NSURLSession`

Summary

- Build the next generation of enterprise apps
- Use iOS 7 features
- Support app configuration and feedback
- Use `NSURLConnection` / `NSURLSession`
- Build in data protection and security

More Information

Paul Marcos

Evangelist

pmarcos@apple.com

Apple in Business

<http://www.apple.com/business/>

Apple in Education

<http://www.apple.com/education/>

Apple Developer Forums

<http://devforums.apple.com>

Related Sessions

Managing Apple Devices

Pacific Heights
Tuesday 11:30AM

Using Receipts to Protect Your Digital Sales

Presidio
Thursday 2:00PM

Labs

Apps for Enterprise and Education Lab

Services Lab B
Tuesday 4:30PM

 WWDC2013