iCloud Development

Lucinda Krahl, Tim Pusateri, Victor Hawley

What is iCloud?

- Cloud storage and cloud computing
- Store, access, and share data across devices.
- Apple devices sync automatically
- PC users can access data online
- Includes “Family Sharing”
- Can be implemented in third-party applications (i.e. our demo app).

History of iCloud

June 2008 - Apple's MobileMe offered storage for contacts, calendars, etc.

June 2011 - Apple announces iCloud, which replaces MobileMe

December 2015 - iCloud Version 5.1 released

February 2016 - iCloud has 782 million users

- http://www.imore.com/history-apple-cloud-applelink-icloud
- https://en.wikipedia.org/wiki/ICloud

Use Cases and Example Apps

- Generally used when it's desirable to sync data over multiple devices and platforms
  - I.e. iPhone, Apple TV, and MacBook
- Lots of native apple apps use iCloud, but most third parties prefer to use their own cloud services

Example Services and Apps:


Photo Library


Drive


Notes


Backup


Contacts


Music

Cloud functionality in iOS

iCloud with Core Data

CloudKit

iCloud with Core Data

- Hybrid local/remote storage
- Make changes to Core Data ➔ iOS takes care of persistence/syncing
- Easy to get started:
  1. Enable iCloud in “Capabilities”
  2. Modify persistent store object to add an "NSPersistentStoreUbiquitousContentNameKey” option.
     a. `let storeOptions = [NSPersistentStoreUbiquitousContentNameKey: "SpecialTopicsDemo"]`
     b. `try coordinator.addPersistentStoreWithType(NSSQLiteStoreType, configuration: nil, URL: url, options: storeOptions)`

iCloud with Core Data - Disadvantages

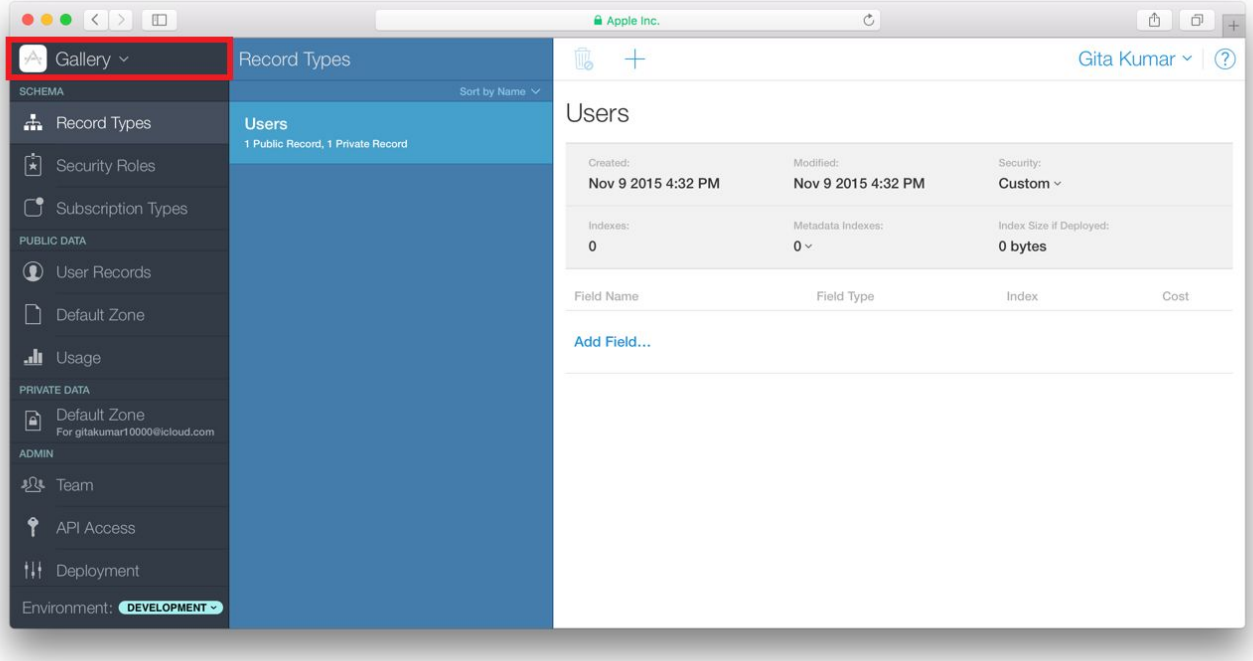
- Hard to test: Need multiple devices to accurately test iCloud sync
  - “It is especially important to test an iCloud-enabled iOS app on an actual device. iOS Simulator doesn’t accurately simulate the use of iCloud under real-world conditions.” (sou *SQLite Store with iCloud* - Apple)
- No control over iCloud syncs
  - Must react to changes in iCloud persistence store using observers
  - `NSNotificationCenter.defaultCenter().addObserver(self, selector: "storesDidChangeMethod", name: NSPersistentStoreCoordinatorStoresDidChangeNotification, object: moContext.persistentStoreCoordinator)`

CloudKit

- Apple-provided cloud database for apps
- Developer has more control than iCloud with Core Data
- Requires enrolling in Apple Developer Program, $99
- Public and Private databases available
  - Public database - information shared by all instances of the app
  - Private database - information only available to current user
- CloudKit JS provides integration with web applications
- All changes occur remotely
- Allows for apps to be notified through subscriptions when a certain action occurs (record inserts, updates, deletes, etc)
- More info: Enabling CloudKit in Your App - Apple

CloudKit - Setup

1. Design schema in online dashboard:



CloudKit - Setup

2. Specify containers your app will use in Xcode: