HomeKit

Session 213

Kevin McLaughlin
Wireless Software Engineering

© 2014 Apple Inc. All rights reserved. Redistribution or public display not permitted without written permission from Apple.

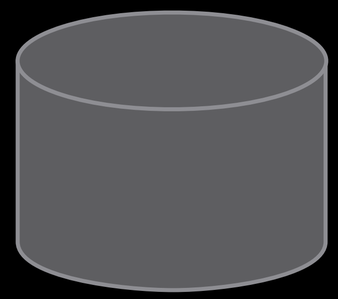
Agenda

Features

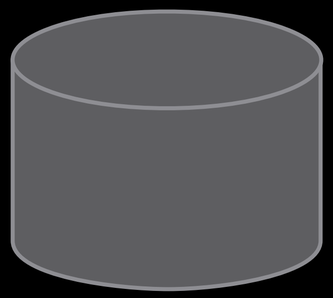
Core Concepts

Advanced Topics

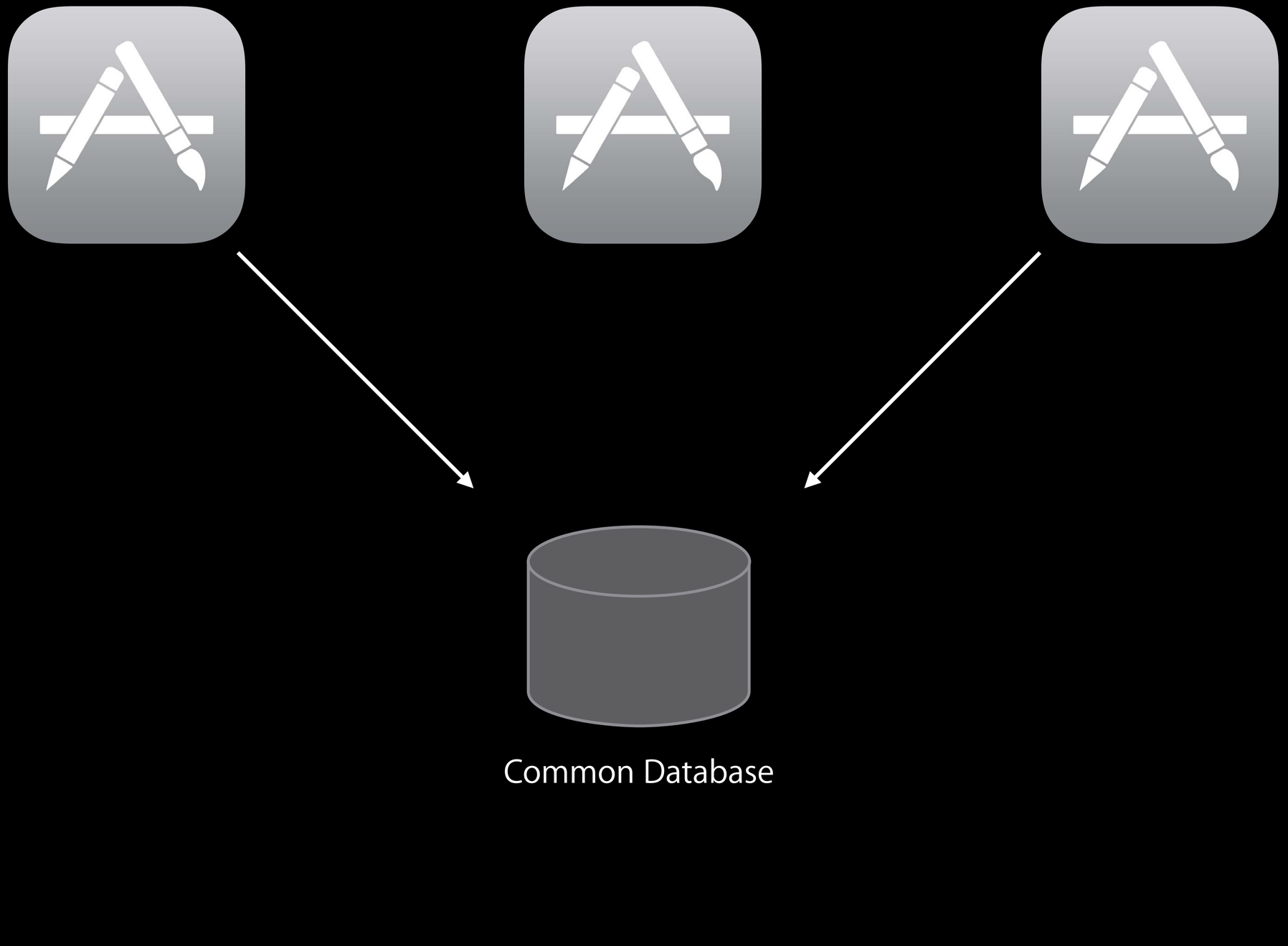
Features

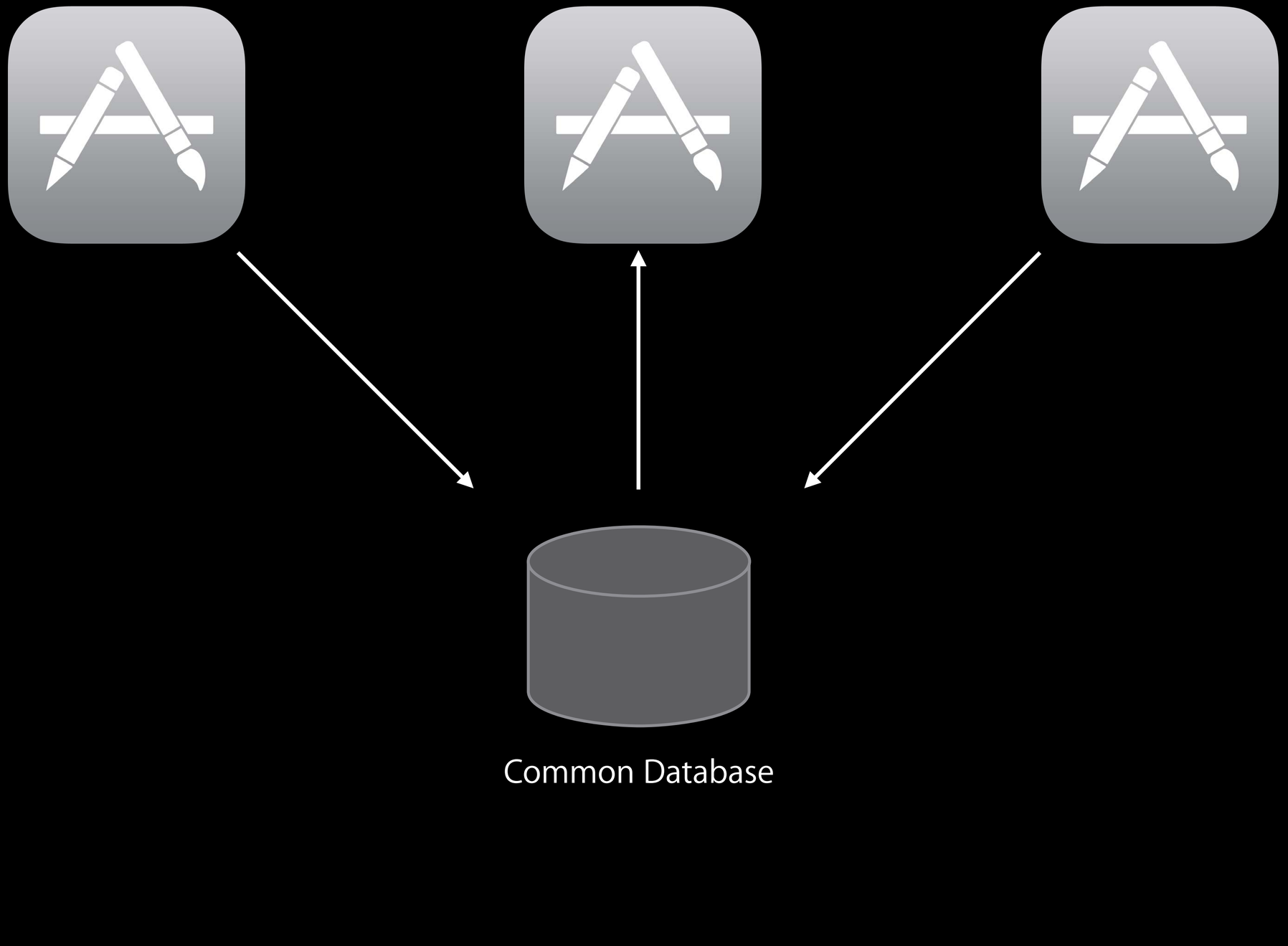


Common Database



Common Database


Common Database


Common Database

Light bulb

Garage door

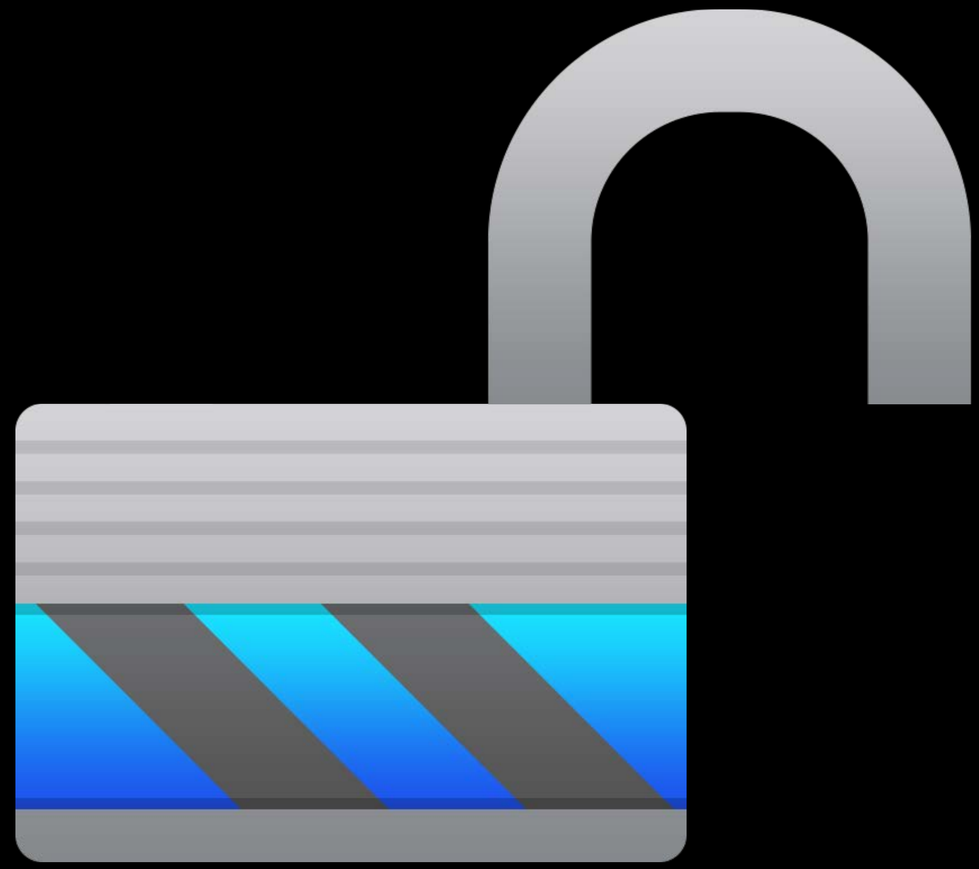
Custom

Door lock

Remote Access

Remote Access

Remote Access



Core Concepts

Home Manager

Entry point

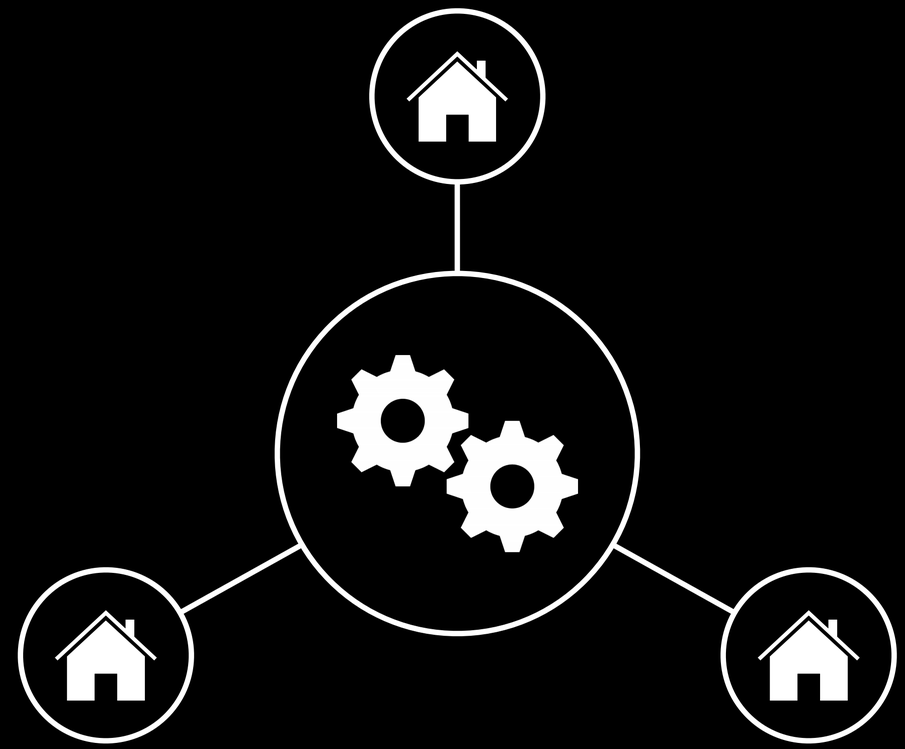
Common database

Manages homes

Primary home

Add or remove homes

Notifies of changes



HMHomeManager

```
self.homeManager = [[HMHomeManager alloc] init];
self.homeManager.delegate = self;
```

Retrieving Homes

```
- (void)homeManagerDidUpdateHomes:(HMHomeManager *)manager {
    self.homes = manager.homes;
    self.primaryHome = manager.primaryHome;
}
```

Creating a Home

```
[self.homeManager addHomeWithName:@“Secret Lair #13”
completionHandler:^(HMHome *home, NSError *error) {
    if (error != nil) {
        // adding the home failed; check error for why
    } else {
        // success!
    }
}];
```

Home

Contains rooms, accessories, etc.

Notifies of changes

Uniquely named

Recognized by Siri

HMHome

```
NSString *homeName = self.home.name;

NSArray *allRooms = self.home.rooms;

NSArray *allAccessories = self.home.accessories;
```

Creating a Room

```
[self.home addRoomWithName:@“Command Center” completionHandler:^(HMRoom
*room, NSError *error) {
    if (error != nil) {
        // unable to add room. check error for why
    } else {
        // success!
    }
}];
```

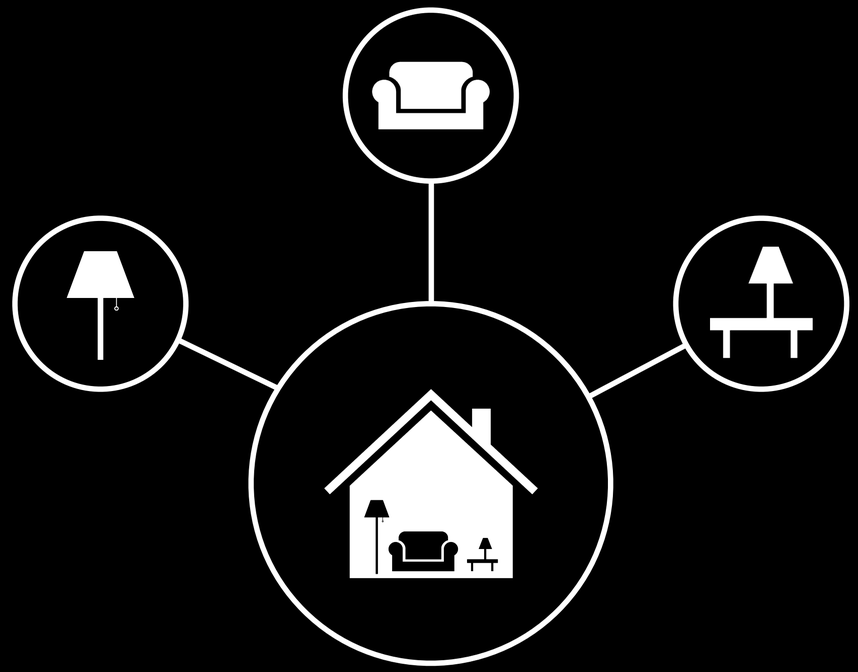
Room

Contains accessories

Notifies of changes

Uniquely named with a home

Recognized by Siri



HMRoom

```
NSArray *accessoriesInRoom = self.room.accessories;
```

Accessory

Corresponds to physical device(s)

Assigned to a room

Accesses device state

Notifies of changes

Uniquely named with a home

Recognized by Siri

Accessory

Corresponds to physical device(s)

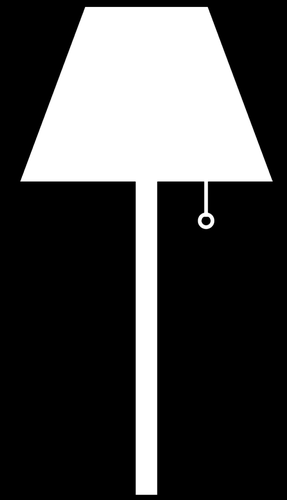
Assigned to a room

Accesses device state

Notifies of changes

Uniquely named with a home

Recognized by Siri



HMAccessory

```
HMRoom *room = self.accessory.room;

NSArray *services = self.accessory.services;
```

HMAccessory

```
- (void)accessoryDidUpdateReachability:(HMAccessory *)accessory {
    if (accessory.reachable == YES) {
        // we can communicate with the accessory
    } else {
        // the accessory is out of range, turned off, etc
    }
}
```

Services

Represents a function of an accessory

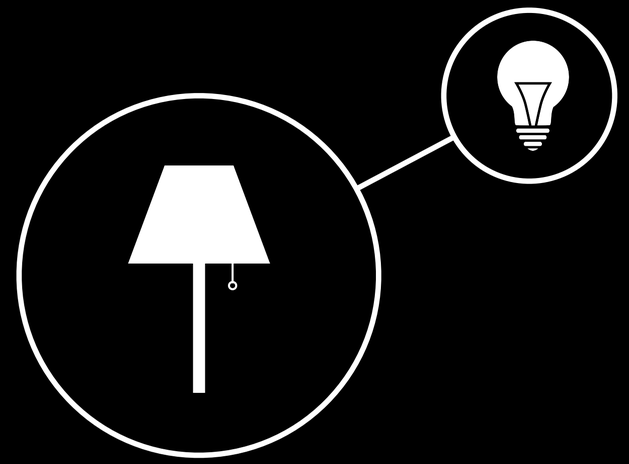
Contains characteristics of the service

May have a name

- A “light bulb” service has a name
- A “firmware update” service does not
- Don’t expose unnamed services
- Names must be unique within a home

Recognized by Siri

- Named and Apple-defined



HMService

```
NSString *name = self.service.name;

NSArray *characteristics = self.service.characteristics;

NSString *serviceType = self.service.serviceType;
```

```
HMAccessory *accessory = self.service.accessory;
```

Characteristics

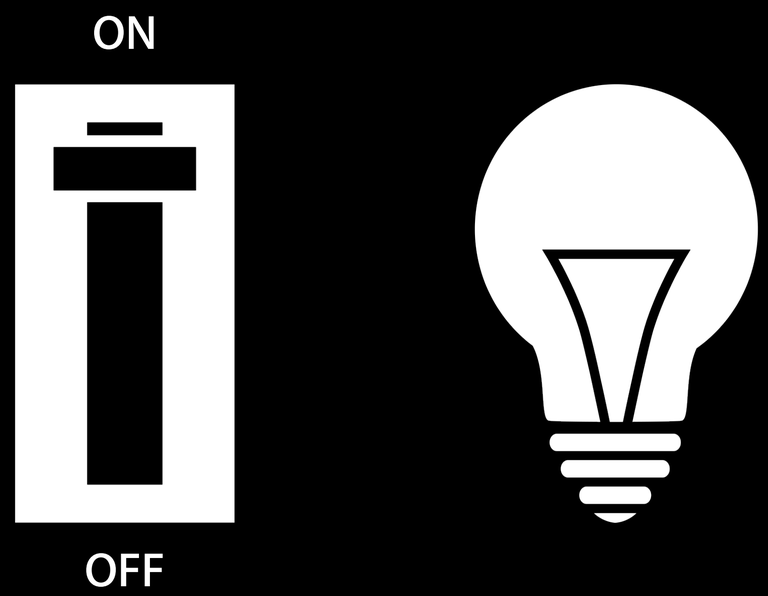
Parameter of a service that allows interaction

Provides information to describe the characteristic

Uses completion handler blocks for results

Recognized by Siri

- Apple-defined



Characteristics

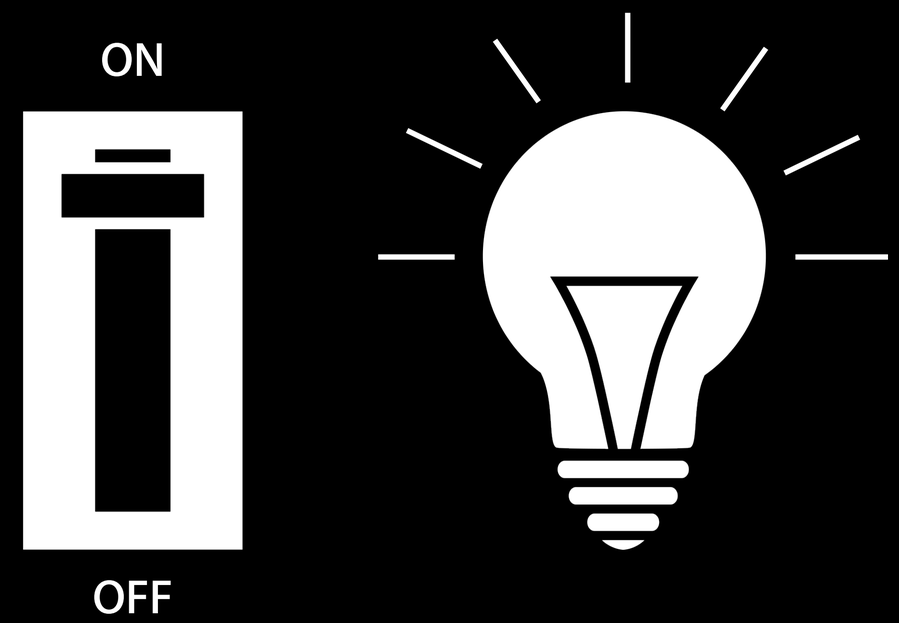
Parameter of a service that allows interaction

Provides information to describe the characteristic

Uses completion handler blocks for results

Recognized by Siri

- Apple-defined



Characteristics

Read-only

- e.g. Current temperature

Read-write

- e.g. Target temperature

Write-only

- e.g. Identify

HMCharacteristic

```
[self.characteristic readValueWithCompletionHandler:^(NSError *error) {
    if (error == nil) {
        id value = self.characteristic.value;
    }
}];
```

```
[self.characteristic writeValue:@42 withCompletionHandler:^(NSError *error) {
    if (error != nil) {
        // unable to write value. check error for why
    }
}];
```

Adding New Accessories

Accessory Browser

Finds new accessories

HMAccessoryBrowser

```
self.accessoryBrowser = [[HMAccessoryBrowser alloc] init];
self.accessoryBrowser.delegate = self;

[self.accessoryBrowser startSearchingForNewAccessories];

… later …

[self.accessoryBrowser stopSearchingForNewAccessories];
```

Finding New Accessories

HMAccessoryBrowser

```
- (void)accessoryBrowser:(HMAccessoryBrowser *)browser didFindNewAccessory:
(HMAccessory *)accessory {
    // a new accessory has been found
}
```

Adding New Accessories to a Home

HMHome

```
[self.home addAccessory:newAccessory completionHandler:^(NSError *error) {
    if (error == nil) {
        // success!
        // assign the accessory to a room and give it a name
    } else {
        // adding accessory failed; check error for why
    }
}];
```

Removing New Accessories from View

HMAccessoryBrowser

```
- (void)accessoryBrowser:(HMAccessoryBrowser *)browser didRemoveNewAccessory:
(HMAccessory *)accessory {
    // a new accessory has been added to a home
    // or has gone out of range or has been turned off
}
```

Testing Your App

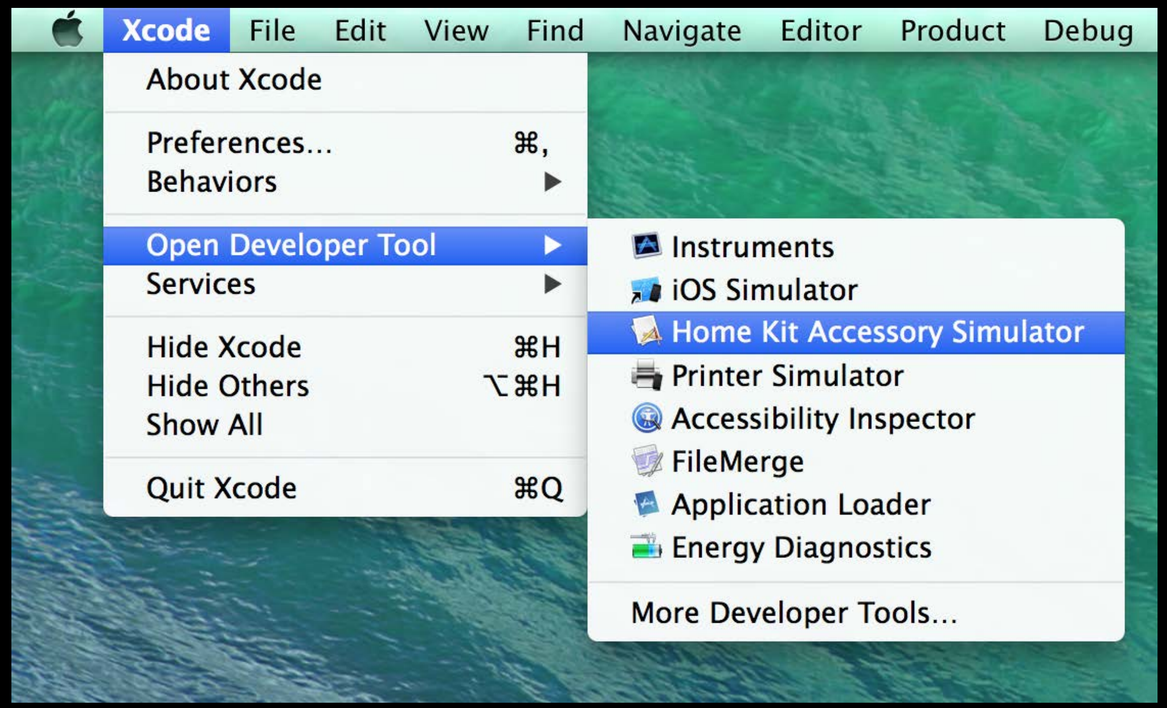
Testing Your App

How do you test?

Testing Your App

How do you test?

HomeKit Accessory Simulator!



Demo

HomeKit Accessory Simulator

HomeKit Accessory Simulator

Available in Xcode 6

Acts like real accessories

Great for developing and testing HomeKit apps

Initial Setup Review

Recommended flow

Initial Setup Review

Recommended flow

Create a home

- User provides name

Initial Setup Review

Recommended flow

Create a home

- User provides name

Add rooms to the home

- User provides names

Initial Setup Review

Recommended flow

Create a home

- User provides name

Add rooms to the home

- User provides names

Add accessories

- Use an accessory browser
- Add accessory to home
- User provides name → update name for accessory
- User chooses room → assign accessory to room

Advanced Topics

Home Kit Accessory Profiles

Services

- Garage door openers
- Lights
- Door locks
- Thermostats
- IP camera controls
- Switches
- …
- Custom

Characteristics

- Power state
- Lock state
- Target state
- Brightness
- Model number
- Current temperature
- …
- Custom

Home Kit Accessory Profiles

Services

- Garage door openers
- Lights
- Door locks
- Thermostats
- IP camera controls
- Switches
- …

- Custom

Characteristics

- Power state
- Lock state
- Target state
- Brightness
- Model number
- Current temperature
- …

- Custom

Zones

HMZone

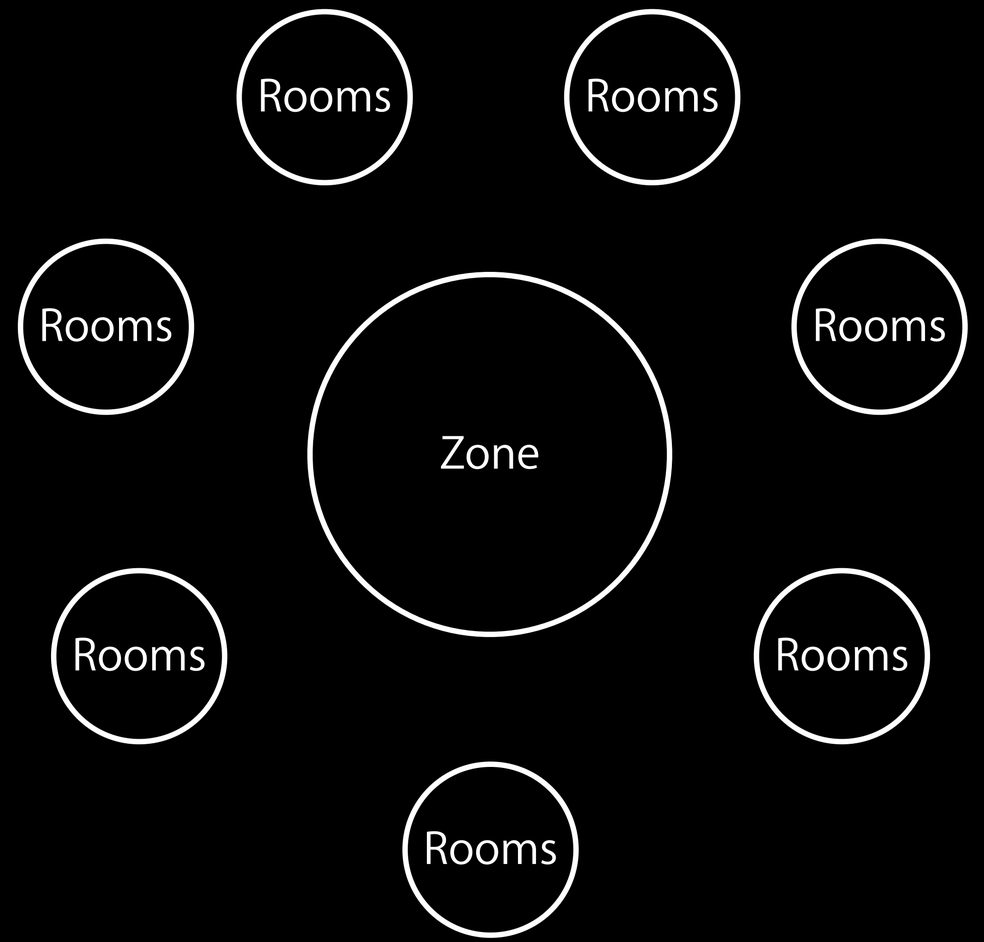
Arbitrary grouping of rooms

- Example: upstairs

Rooms can be in any number of zones

Uniquely named with a home

Recognized by Siri



Zones

HMZone

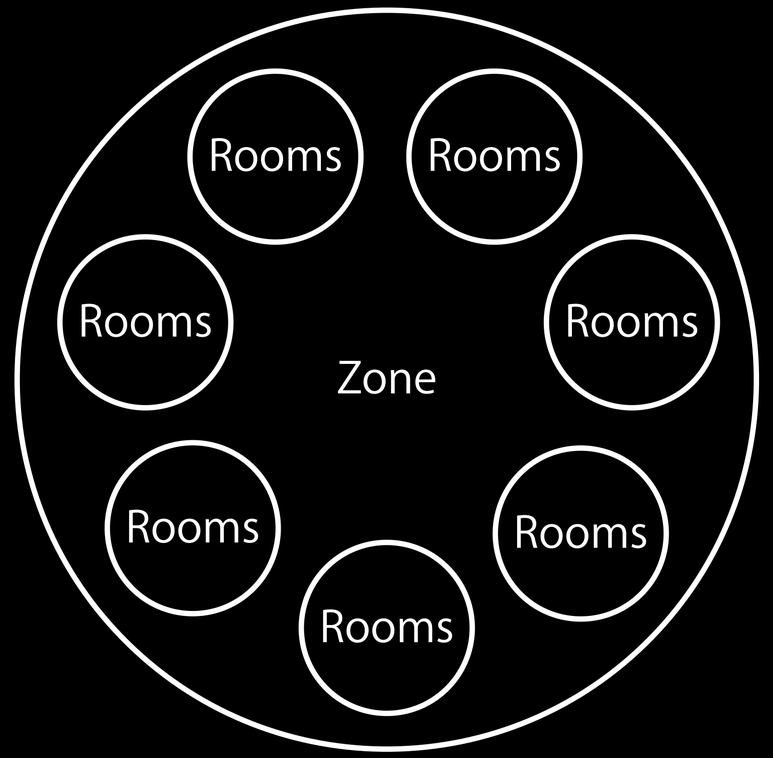
Arbitrary grouping of rooms

- Example: upstairs

Rooms can be in any number of zones

Uniquely named with a home

Recognized by Siri



Service Groups

HMServiceGroup

Arbitrary grouping of services

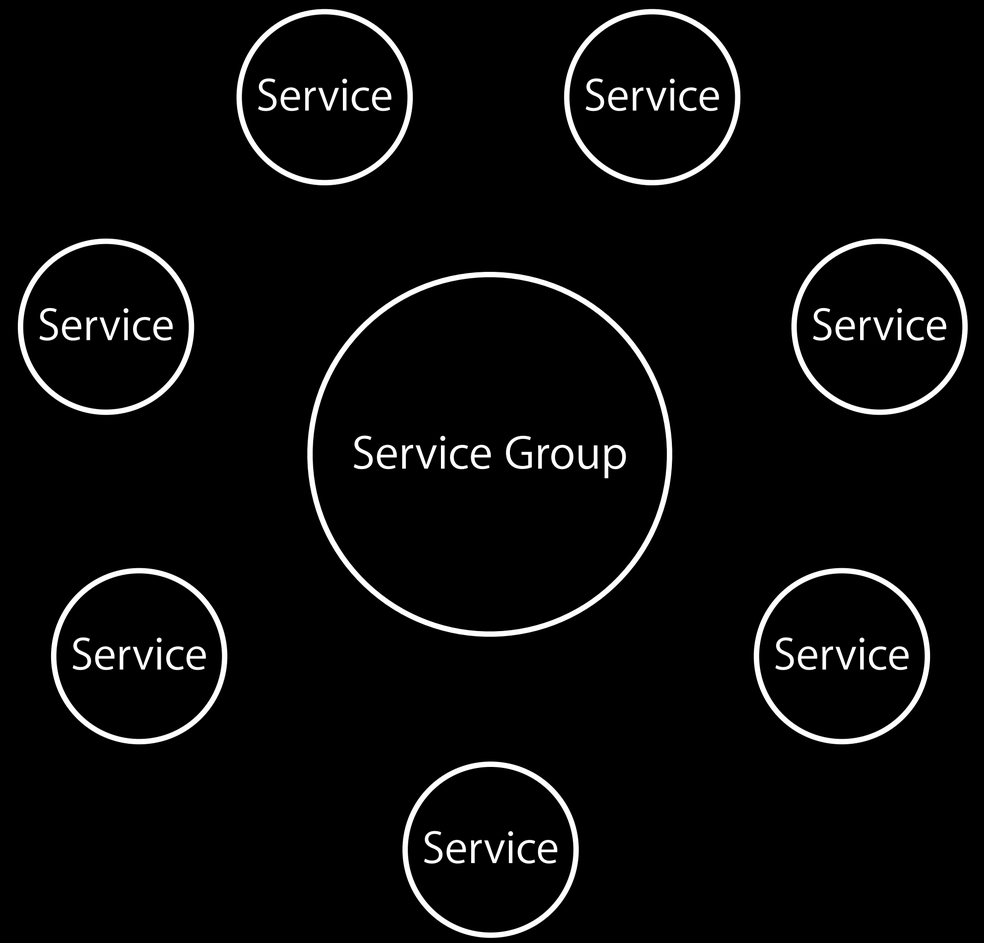
- Example: nightlights

Services can be in any number of groups

Convenient way to control services across accessories

Uniquely named with a home

Recognized by Siri



Service Groups

HMServiceGroup

Arbitrary grouping of services

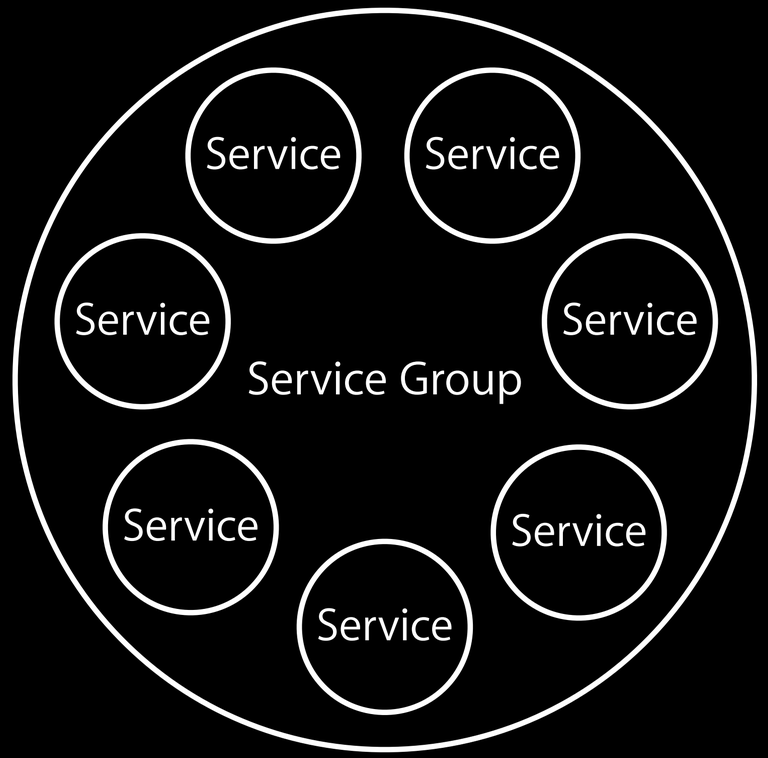
- Example: nightlights

Services can be in any number of groups

Convenient way to control services across accessories

Uniquely named with a home

Recognized by Siri



Action Sets

HMActionSet

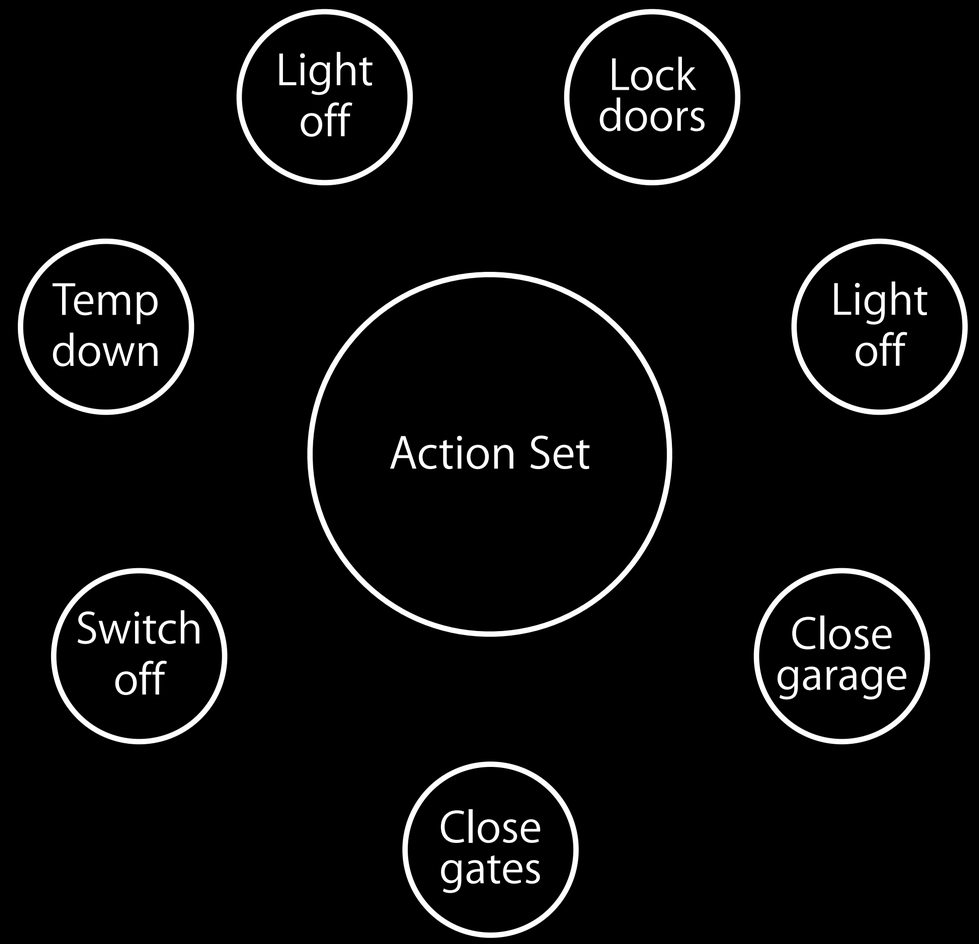
Collection of actions that are executed together

- Example: “night”

Actions executed in undefined order

Uniquely named with a home

Recognized by Siri



Action Sets

HMActionSet

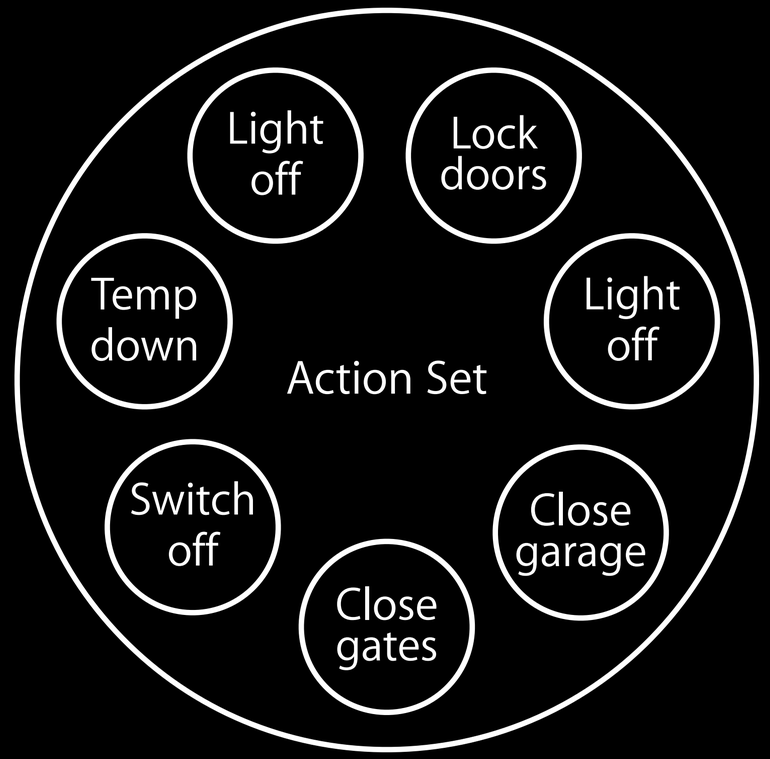
Collection of actions that are executed together

- Example: “night”

Actions executed in undefined order

Uniquely named with a home

Recognized by Siri

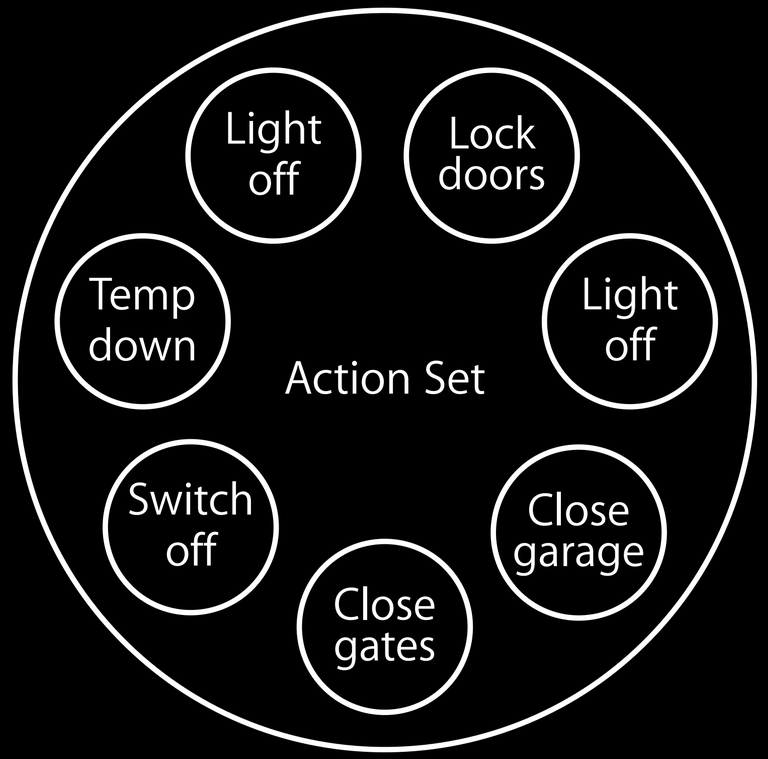


Actions

HMCharacteristicWriteAction

Added to action sets

Writes a value to a characteristic

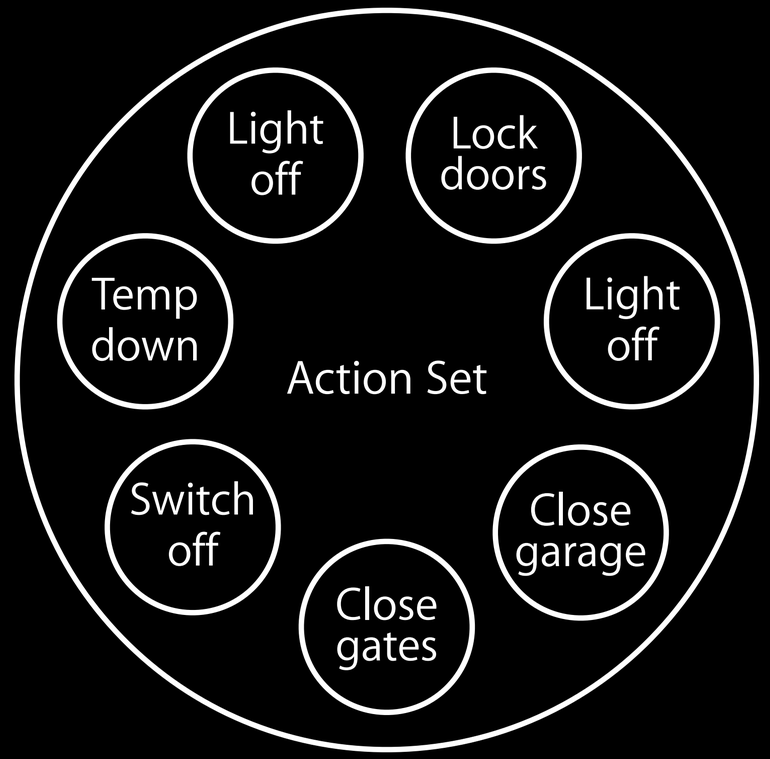


Actions

HMCharacteristicWriteAction

Added to action sets

Writes a value to a characteristic

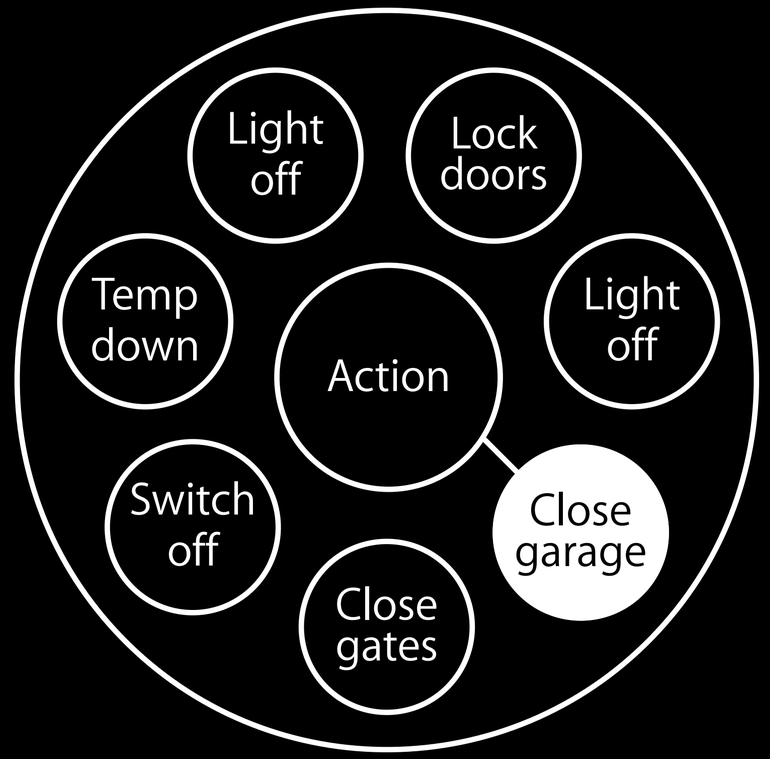


Actions

HMCharacteristicWriteAction

Added to action sets

Writes a value to a characteristic



Triggers

HMTimerTrigger

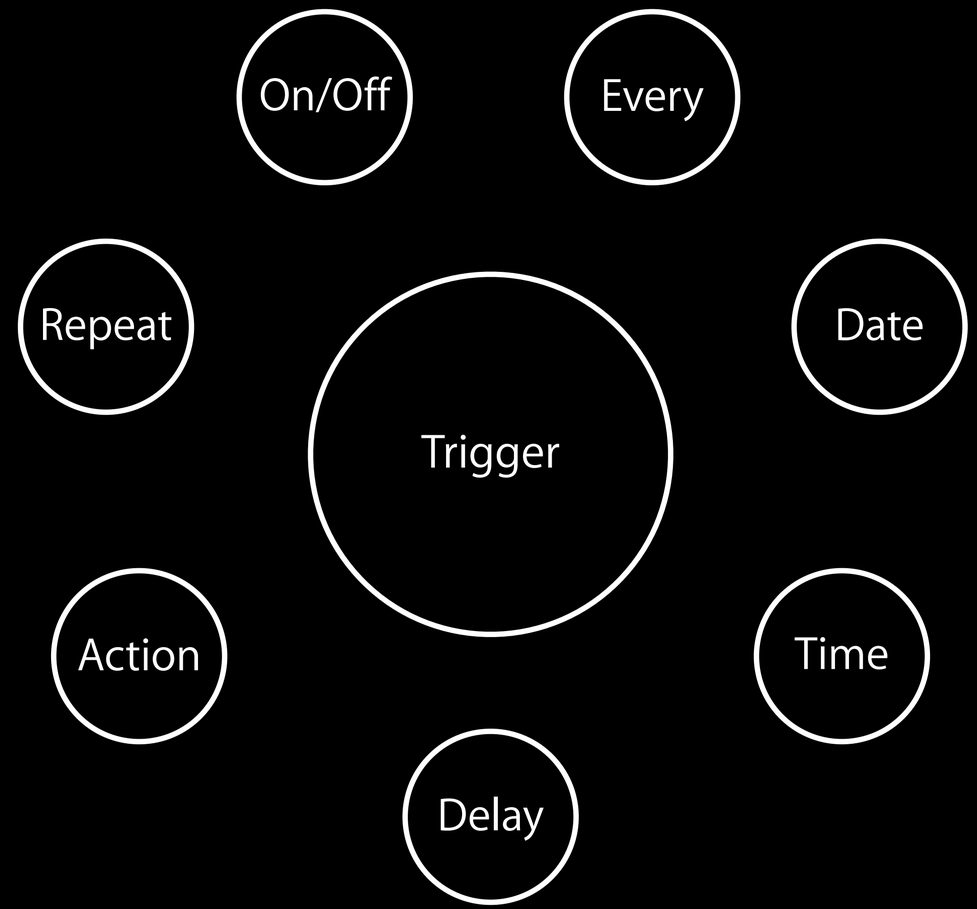
Executes an action set

Fires on a specified date

Can repeat

Uniquely named with a home

System executes in background



Triggers

HMTimerTrigger

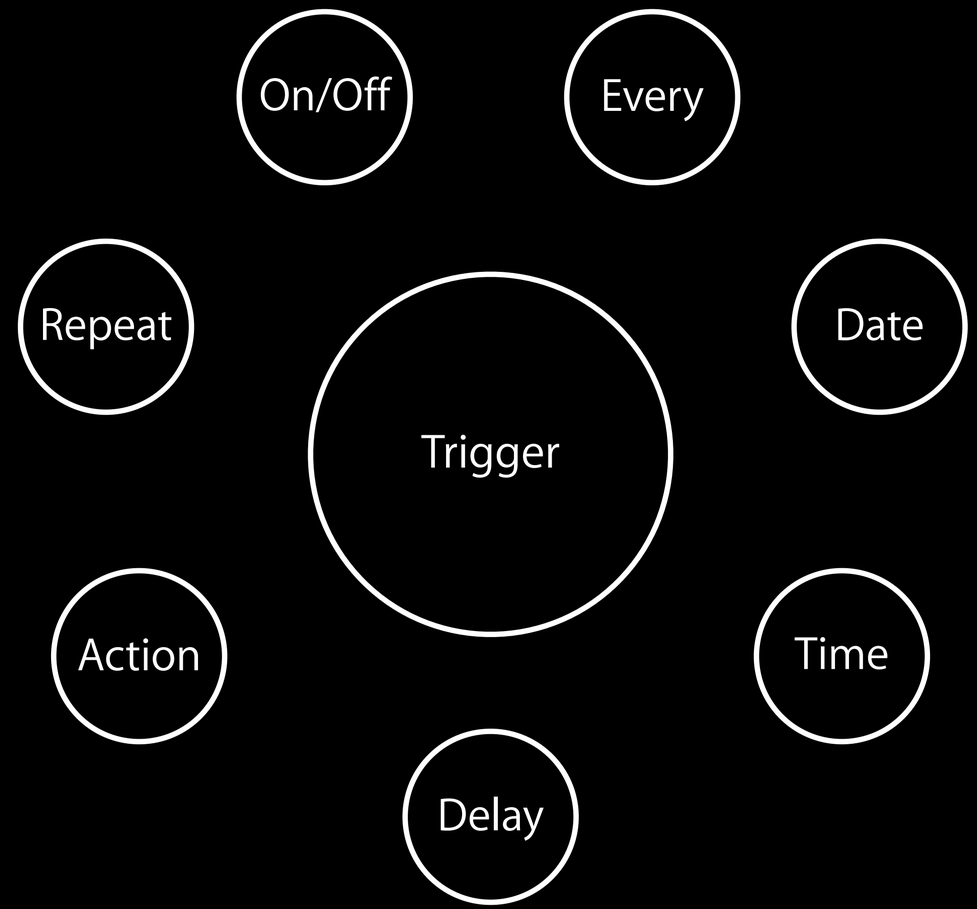
Executes an action set

Fires on a specified date

Can repeat

Uniquely named with a home

System executes in background



Triggers

HMTimerTrigger

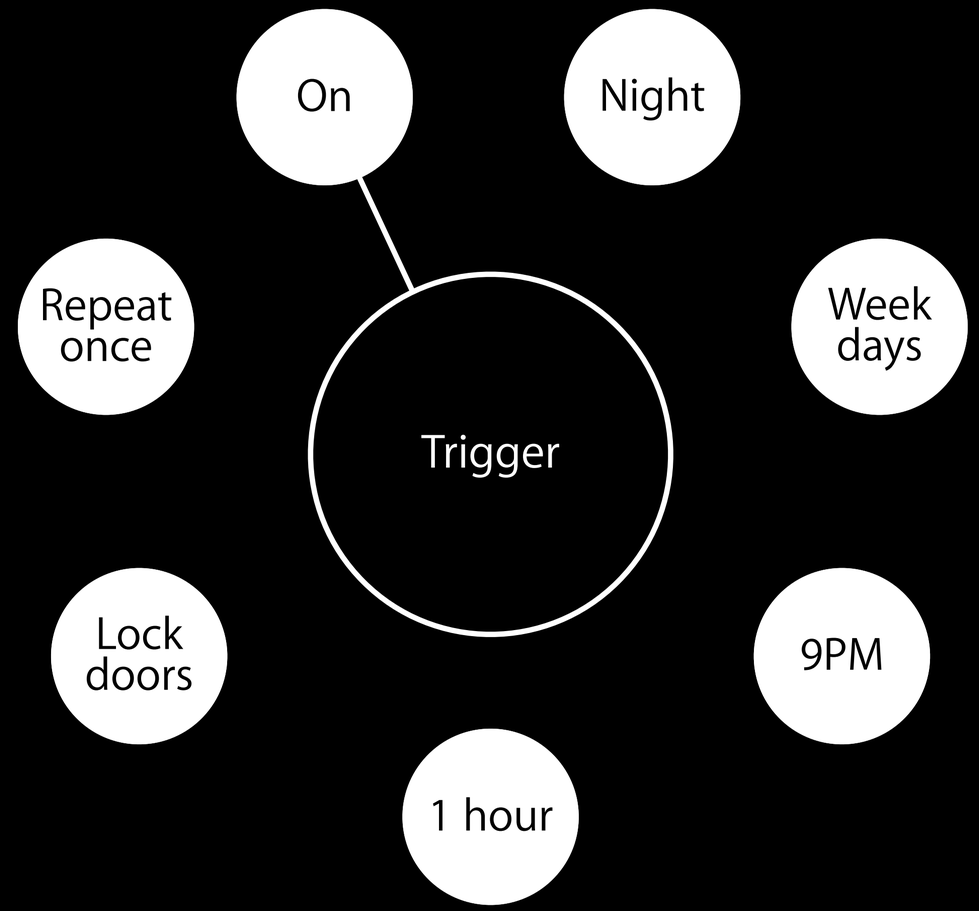
Executes an action set

Fires on a specified date

Can repeat

Uniquely named with a home

System executes in background



Demo

Action Sets and Triggers

Action Sets and Triggers

"Script" complex behavior

Quickly control many accessories

Why So Many Delegate Methods?

Why So Many Delegate Methods?

Shared database

- Other apps can cause changes
- System updates database

Accessories come and go, change state, …

Wrap Up

Consistent experience through HomeKit

Easy-to-use Framework

HomeKit Accessory Simulator

HomeKit

- http://developer.apple.com/homekit

Labs

• HomeKit Lab	Services Lab A	Thursday 12:45PM

More Information

Dave DeLong
App Frameworks Evangelist
delong@apple.com

Craig Keithley
MFi and I/O Technologies Evangelist
keithley@apple.com

Documentation
http://developer.apple.com
http://developer.apple.com/mfi/

Apple Developer Forums
http://devforums.apple.com

WWDC14