Core OS

#WWDC14

Designing Accessories for iOS and OS X

Session 701

Robert Walsh
Manager, Platform Accessories

Introduction

Accessory interfaces

Accessory protocols

What's new in iOS 8 and OS X Yosemite

Program information and wrap up

Glucose meters

Health thermometers

App-enabled accessories

Wireless speakers

Light bulbs

Blood pressure monitors

Speakers

Storage devices

Audio devices

Garage door openers

Headphones

Heart rate monitors

Hearing aids

Game controllers

iBeacon

Video devices

Thermostats

Cases

Glucose meters

Health thermometers

App-enabled accessories

Wireless speakers

Light bulbs

Blood pressure monitors

Speakers

Storage devices

Audio devices

Garage door openers

Headphones

Heart rate monitors

Hearing aids

Game controllers

iBeacon

Video devices

Thermostats

Cases

Accessory Interfaces

Wireless

- Bluetooth
- Wi-Fi

Wired

- Lightning connector
- USB
- Thunderbolt 2
- Headphone remote and mic

Bluetooth

Bluetooth Low Energy

Supported features

	iOS	OS X	App framework	For more information
General communication with apps	✓	✓	Core Bluetooth	Bluetooth Design Guidelines
Apple Notification Center Service (ANCS)	✓		Notification Center	Bluetooth Design Guidelines
Hearing aid audio transport	✓		Core Audio	MFi Program
HealthKit	✓		HealthKit	Bluetooth Design Guidelines MFi Program
HomeKit	✓		HomeKit	MFi Program
iBeacon	✓		Core Location	iBeacon Licensing

Bluetooth Low Energy

Supported features

	iOS	OS X	App framework	For more information
General communication with apps	✓	✓	Core Bluetooth	Bluetooth Design Guidelines
Apple Notification Center Service (ANCS)	✓		Notification Center	Bluetooth Design Guidelines
Hearing aid audio transport	✓		Core Audio	MFi Program
HealthKit	✓		HealthKit	Bluetooth Design Guidelines MFi Program
HomeKit	✓		HomeKit	MFi Program
iBeacon	✓		Core Location	iBeacon Licensing

Bluetooth Low Energy

Supported features

	iOS	OS X	App framework	For more information
General communication with apps	✓	✓	Core Bluetooth	Bluetooth Design Guidelines
Apple Notification Center Service (ANCS)	✓		Notification Center	Bluetooth Design Guidelines
Hearing aid audio transport	✓		Core Audio	MFi Program
HealthKit	✓		HealthKit	Bluetooth Design Guidelines MFi Program
HomeKit	✓		HomeKit	MFi Program
iBeacon	✓		Core Location	iBeacon Licensing

Classic Bluetooth

Support for standard features

	iOS	OS X	App framework
Advanced Audio Distribution Profile (A2DP)	✓	✓	Advanced Audio Distribution Profile (A2DP)
Audio/Video Remove Control Profile (AVRCP)	✓	✓	Audio/Video Remove Control Profile (AVRCP)
Dial Up Network (DUN) Profile		✓	Dial Up Network (DUN) Profile
File Transfer Profile (FTP)		✓	File Transfer Profile (FTP)
Hands-Free Profile (HFP)	✓		Hands-Free Profile (HFP)
Hardcopy Cable Replacement Profile (HCRP)		✓	Hardcopy Cable Replacement Profile (HCRP)
Headset Profile (HSP)		✓	Headset Profile (HSP)
Human Interface Device (HID) Profile	✓	✓	Human Interface Device (HID) Profile
Message Access Profile (MAP)	✓		Message Access Profile (MAP)
Object Push Profile (OPP)		✓	Object Push Profile (OPP)
Personal Area Network Profile (PAN)	✓		Personal Area Network Profile (PAN)
Phone Book Access Profile (PBAP)	✓		Phone Book Access Profile (PBAP)
Serial Port Profile (SPP)		✓	Serial Port Profile (SPP)

Classic Bluetooth

Support for extended features on iOS

	For more information
Headset battery level indication	Bluetooth Design Guidelines
Siri Eyes Free	Bluetooth Design Guidelines
iAP • Access to media library • Access to location • Proprietary communication with apps • Direct users to App Store when accessory is paired • Game controllers	MFi Program

Wi-Fi

Wi-Fi Alliance certification

- 802.11a/b/g, 802.11n, 802.11ac

Dual-band operation

- 2.4 and 5 GHz

WPA2 security mode

Quality of services

- Wireless Multimedia (WMM)

IPv6

Lightning Connector

Lightning headphone module

- Richer controls, including iTunes Radio
- App-enabled headphones

New connector modules

USB

All current Macs ship with USB 3 ports

All current Macs support 5 Gbps per port

900 mA per port for device power and charging

High performance storage devices should use UAS

Built-in driver support for XHCI 1.0 compliant host controllers

Thunderbolt 2

High speed data and 4K video on one cable

PCI Express and DisplayPort protocol

Dual-channel, bi-directional 20 Gbps per port

Up to 10W for bus-powered devices

Daisy chain up to six devices per port

Certification program ensures high quality devices

Refer to *Thunderbolt Device Driver Programming Guide*

Headphone Remote and Mic

Apps can receive remote control events

Standardized microphone

What’s New in iOS 8 and OS X Yosemite

HealthKit Accessories

Terry Worley
Software Engineer, HealthKit team

HealthKit Sneak Peek

HealthKit Sneak Peek

New framework in iOS 8

HealthKit Sneak Peek

New framework in iOS 8

Save and access health and fitness data

HealthKit Sneak Peek

New framework in iOS 8

Save and access health and fitness data

Class A data security

HealthKit Sneak Peek

New framework in iOS 8

Save and access health and fitness data

Class A data security

User privacy settings

HealthKit Sneak Peek

New framework in iOS 8

Save and access health and fitness data

Class A data security

User privacy settings

Searches and statistical queries

HealthKit Sneak Peek

New framework in iOS 8

Save and access health and fitness data

Class A data security

User privacy settings

Searches and statistical queries

Unit conversions

HealthKit Sneak Peek

New framework in iOS 8

Save and access health and fitness data

Class A data security

User privacy settings

Searches and statistical queries

Unit conversions

Notifications of new data

HealthKit Sneak Peek

New framework in iOS 8

Save and access health and fitness data

Class A data security

User privacy settings

Searches and statistical queries

Unit conversions

Notifications of new data

Integration with accessories

Health and Fitness Accessories

Health and Fitness Accessories

Bluetooth Low Energy is ideal for health and fitness

Health and Fitness Accessories

Bluetooth Low Energy is ideal for health and fitness

Accessories play the leading role in providing HealthKit data

Health and Fitness Accessories

Bluetooth Low Energy is ideal for health and fitness

Accessories play the leading role in providing HealthKit data

Accessories with built-in HealthKit support

- Heart rate monitor
- Glucose sensor
- Blood pressure monitor
- Health thermometer

Health and Fitness Accessories

Bluetooth Low Energy is ideal for health and fitness

Accessories play the leading role in providing HealthKit data

Accessories with built-in HealthKit support

- Heart rate monitor
- Glucose sensor
- Blood pressure monitor
- Health thermometer

Once paired, HealthKit automatically controls the accessory

Developing Native HealthKit Accessories

Developing Native HealthKit Accessories

Adhere to Bluetooth Low Energy GATT Specifications

Developing Native HealthKit Accessories

Adhere to Bluetooth Low Energy GATT Specifications

Best practices

Developing Native HealthKit Accessories

Adhere to Bluetooth Low Energy GATT Specifications

Best practices

- Implement optional fields within the services, e.g. the heart rate field within the blood pressure service saved as its own data point in HealthKit

Developing Native HealthKit Accessories

Adhere to Bluetooth Low Energy GATT Specifications

Best practices

- Implement optional fields within the services, e.g. the heart rate field within the blood pressure service saved as its own data point in HealthKit
- Implement supported metadata, e.g. energy expended stored as metadata along with the heart rate measurement in HealthKit

Developing Native HealthKit Accessories

Adhere to Bluetooth Low Energy GATT Specifications

Best practices

- Implement optional fields within the services, e.g. the heart rate field within the blood pressure service saved as its own data point in HealthKit
- Implement supported metadata, e.g. energy expended stored as metadata along with the heart rate measurement in HealthKit
- Implement optional characteristics, e.g. sensor contact to help identify valid data

Developing Native HealthKit Accessories

Adhere to Bluetooth Low Energy GATT Specifications

Best practices

- Implement optional fields within the services, e.g. the heart rate field within the blood pressure service saved as its own data point in HealthKit
- Implement supported metadata, e.g. energy expended stored as metadata along with the heart rate measurement in HealthKit
- Implement optional characteristics, e.g. sensor contact to help identify valid data
- Implement optional services, e.g. battery Service

Developing Other Accessories

Developing Other Accessories

Custom solutions can contribute to HealthKit

Developing Other Accessories

Custom solutions can contribute to HealthKit

Use any service that makes sense for your accessory. That might include the ExternalAccessory framework, CoreBluetooth, USB, or Wi-Fi

Developing Other Accessories

Custom solutions can contribute to HealthKit

Use any service that makes sense for your accessory. That might include the ExternalAccessory framework, CoreBluetooth, USB, or Wi-Fi

Then use the HealthKit framework APIs from within your app to add your data

Developing Other Accessories

Custom solutions can contribute to HealthKit

Use any service that makes sense for your accessory. That might include the ExternalAccessory framework, CoreBluetooth, USB, or Wi-Fi

Then use the HealthKit framework APIs from within your app to add your data

We hope these custom solutions will join in

HealthKit Software Support

HealthKit Software Support

```
@property (nonatomic) HKHealthStore *healthStore;
[self setHealthStore:[[HKHealthStore alloc] init]];
```

HealthKit Software Support

```
@property (nonatomic) HKHealthStore *healthStore;
[self setHealthStore:[[HKHealthStore alloc] init]];
...
HKQuantityType *heartRateType =
  [HKObjectType quantityTypeForIdentifier:HKQuantityTypeIdentifierHeartRate];
```

HealthKit Software Support

```
@property (nonatomic) HKHealthStore *healthStore;
[self setHealthStore:[[HKHealthStore alloc] init]];
...
HKQuantityType *heartRateType =
  [HKObjectType quantityTypeForIdentifier:HKQuantityTypeIdentifierHeartRate];
```

HealthKit Software Support

```
@property (nonatomic) HKHealthStore *healthStore;
[self setHealthStore:[[HKHealthStore alloc] init]];
...
HKQuantityType *heartRateType =
  [HKObjectType quantityTypeForIdentifier:HKQuantityTypeIdentifierHeartRate];
HKObserverQuery *heartRateQuery =
  [[HKObserverQuery alloc] initWithSampleType:heartRateType predicate:nil
    updateHandler:^(HKObserverQuery *query,
                    void (^completion)(void), NSError *error) {

  }];
```

HealthKit Software Support

```
@property (nonatomic) HKHealthStore *healthStore;
[self setHealthStore:[[HKHealthStore alloc] init]];
...
HKQuantityType *heartRateType =
  [HKObjectType quantityTypeForIdentifier:HKQuantityTypeIdentifierHeartRate];
HKObserverQuery *heartRateQuery =
[[HKObserverQuery alloc] initWithSampleType:heartRateType predicate:nil
  updateHandler:^(HKObserverQuery *query,
                  void (^completion)(void), NSError *error) {
    if (error == nil) {
       // Use HealthKit to fetch the new sample(s) from the database.
       [self fetchNewSamples];
    }
 }];
```

HealthKit Software Support

```
@property (nonatomic) HKHealthStore *healthStore;
[self setHealthStore:[[HKHealthStore alloc] init]];
...
HKQuantityType *heartRateType =
  [HKObjectType quantityTypeForIdentifier:HKQuantityTypeIdentifierHeartRate];
HKObserverQuery *heartRateQuery =
  [[HKObserverQuery alloc] initWithSampleType:heartRateType predicate:nil
    updateHandler:^(HKObserverQuery *query,
                    void (^completion)(void), NSError *error) {
      if (error == nil) {
         // Use HealthKit to fetch the new sample(s) from the database.
         [self fetchNewSamples];
      }
   }];
[self.healthStore executeQuery:heartRateQuery];
```

Demo

Heart rate monitor

Jorge Moriñigo
Software Engineer, HealthKit team

Related Sessions

Introducing HealthKit	Mission	Tuesday 10:15AM
Health and Fitness Get Together	Folsom	Tuesday 4:30PM

HomeKit Accessories

Kevin McLaughlin
Wireless Software Engineering

HomeKit

HomeKit

Bring exciting new accessories to our users

HomeKit

Bring exciting new accessories to our users

Provide consistent user experience

HomeKit

Bring exciting new accessories to our users

Provide consistent user experience

Learn about APIs at HomeKit session

HomeKit Accessory Protocol

HomeKit Accessory Protocol

HomeKit

Accessory

Protocol

HomeKit Accessory Protocol

HomeKit

Accessory

Protocol

“HAP”

HomeKit Accessory Protocol

Connecting accessories to HomeKit

HomeKit Accessory Protocol

Connecting accessories to HomeKit

Transports

- Bluetooth Low Energy
- IP

HomeKit Accessory Protocol

Connecting accessories to HomeKit

Transports

- Bluetooth Low Energy
- IP

Security

- Bi-directional authentication
- Per-session encryption

HomeKit Accessory Protocol

Connecting accessories to HomeKit

Transports

- Bluetooth Low Energy
- IP

Security

- Bi-directional authentication
- Per-session encryption

Common functionality definitions

- Services
- Characteristics

HomeKit 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

HomeKit Accessory Profiles

Services

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

- Custom

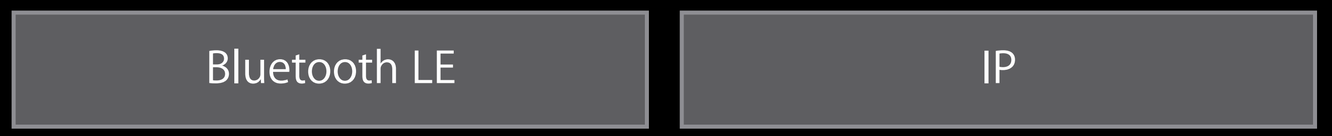
Characteristics

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

- Custom

HomeKit Accessory Protocol Layers

HomeKit Accessory Protocol Layers



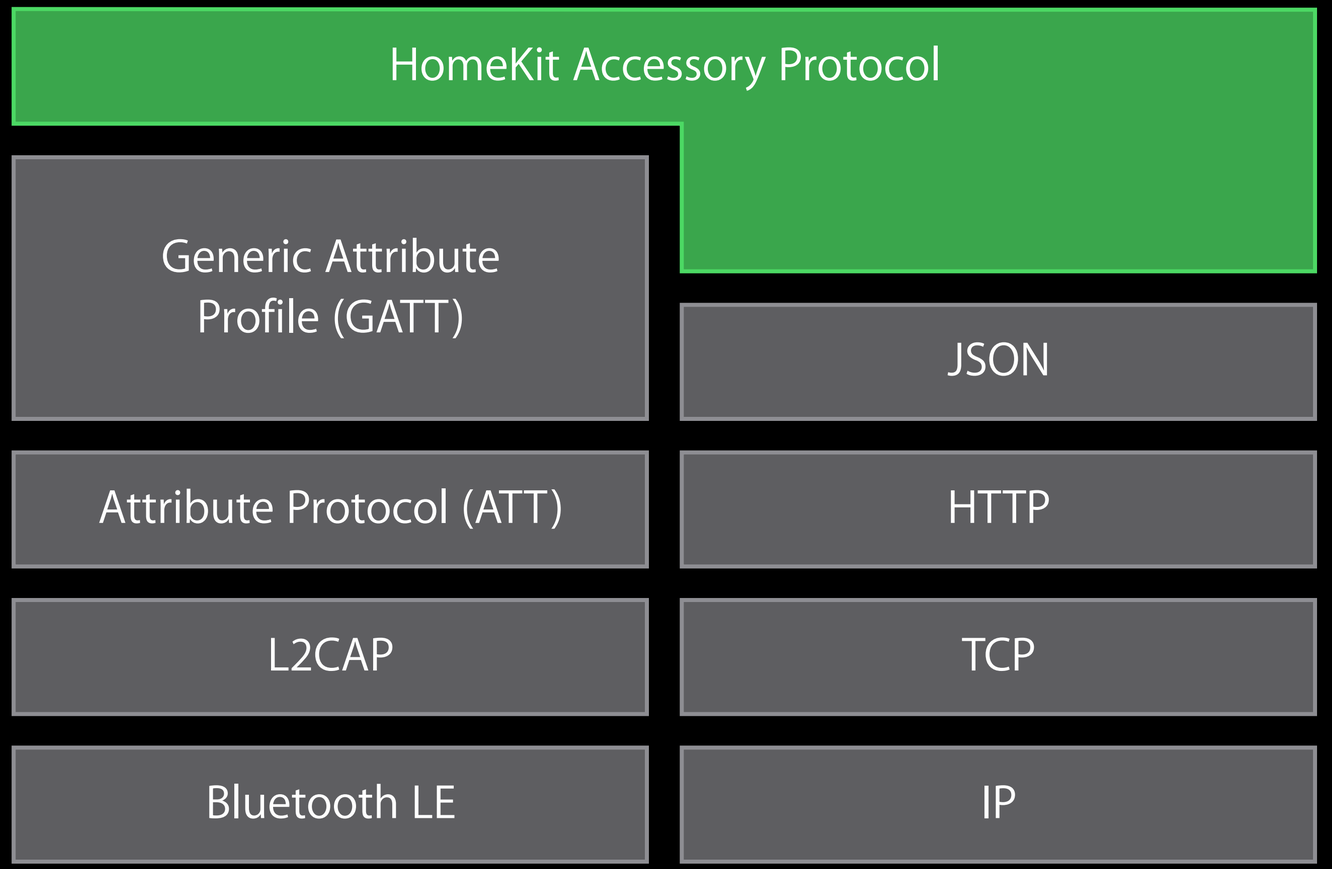
HomeKit Accessory Protocol Layers

HomeKit Accessory Protocol Layers

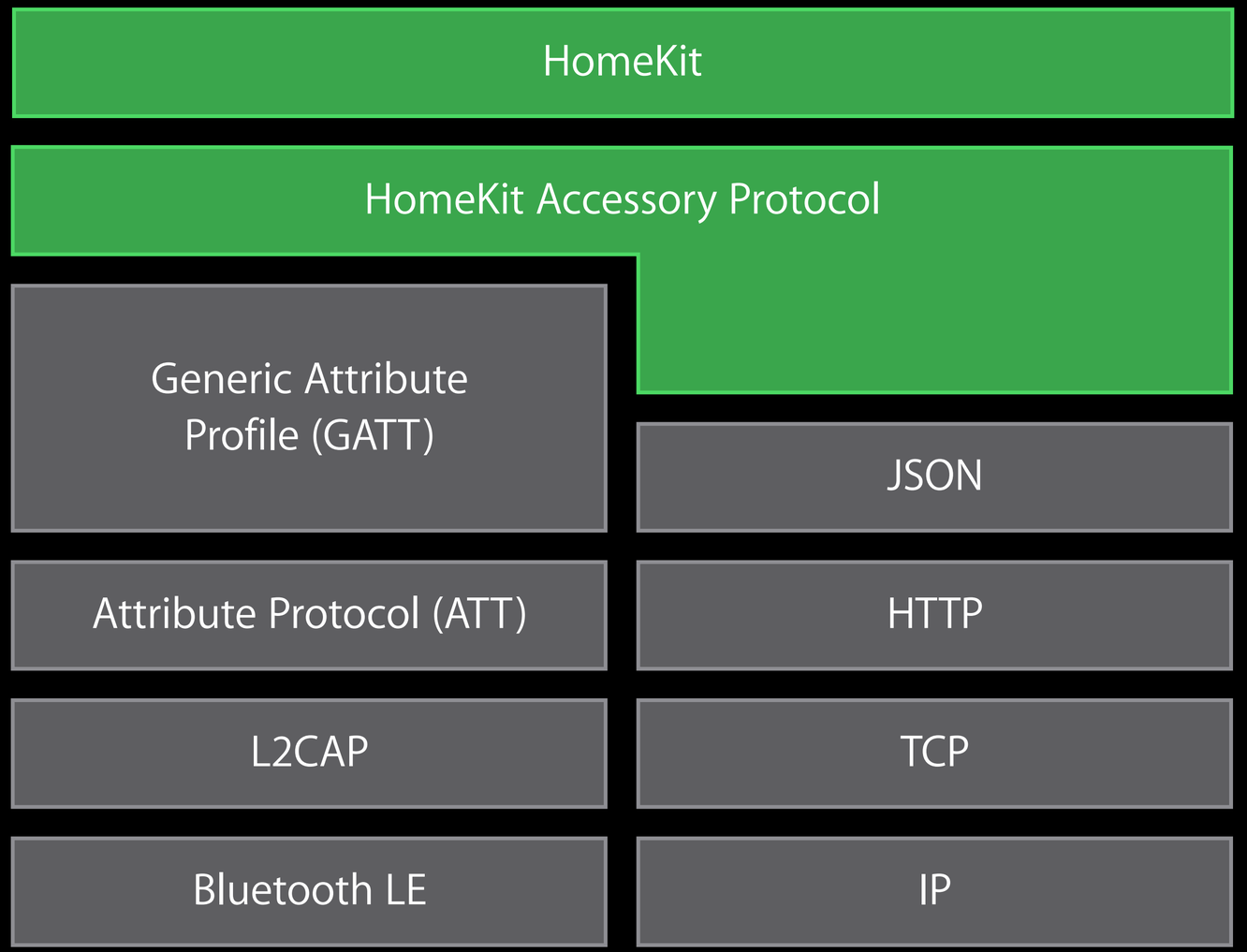
HomeKit Accessory Protocol Layers

Generic Attribute Profile (GATT)	JSON
Attribute Protocol (ATT)	HTTP
L2CAP	TCP
Bluetooth LE	IP

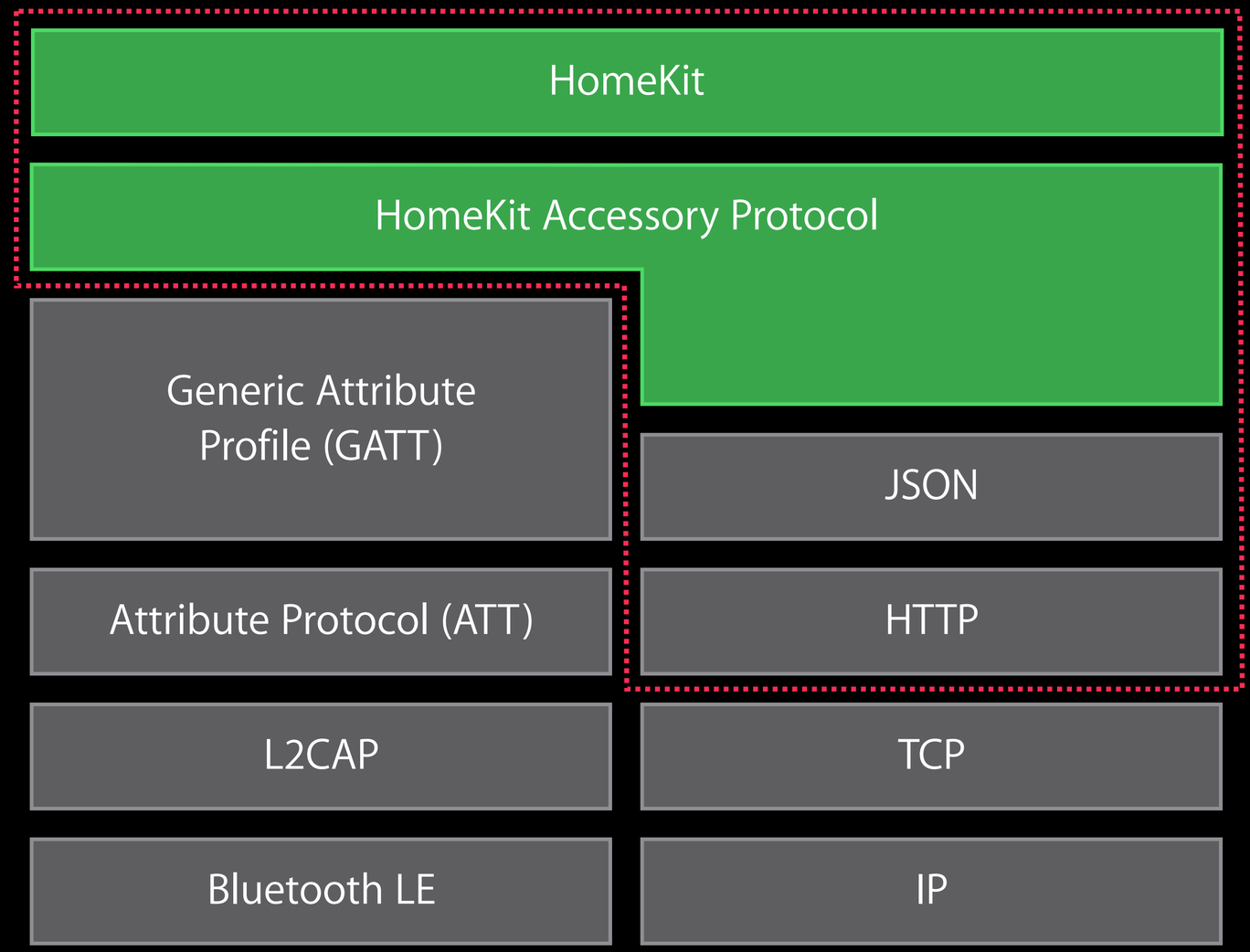
HomeKit Accessory Protocol Layers



HomeKit Accessory Protocol Layers



HomeKit Accessory Protocol Layers



Garage Door Opener

Services and characteristics example

Garage Door Opener

Services and characteristics example

```
accessory : {
```

Garage Door Opener

Services and characteristics example

```
accessory : {
    service1 : “public.hap.accessory-information” {
```

Garage Door Opener

Services and characteristics example

```
accessory : {
    service1 : “public.hap.accessory-information” {
        characteristic : “serial-number”
```

Garage Door Opener

Services and characteristics example

```
accessory : {
    service1 : “public.hap.accessory-information” {
        characteristic : “serial-number”
        characteristic : “identify”
    }
```

Garage Door Opener

Services and characteristics example

```
accessory : {
    service1 : “public.hap.accessory-information” {
        characteristic : “serial-number”
        characteristic : “identify”
    }
    service2 : “public.hap.garage-door-opener” {
```

Garage Door Opener

Services and characteristics example

```
accessory : {
    service1 : “public.hap.accessory-information” {
        characteristic : “serial-number”
        characteristic : “identify”
    }
    service2 : “public.hap.garage-door-opener” {
        characteristic : “target-state”
```

Garage Door Opener

Services and characteristics example

```
accessory : {
    service1 : “public.hap.accessory-information” {
        characteristic : “serial-number”
        characteristic : “identify”
    }
    service2 : “public.hap.garage-door-opener” {
        characteristic : “target-state”
        characteristic : “current-state”
```

Garage Door Opener

Services and characteristics example

```
accessory : {
    service1 : “public.hap.accessory-information” {
        characteristic : “serial-number”
        characteristic : “identify”
    }
    service2 : “public.hap.garage-door-opener” {
        characteristic : “target-state”
        characteristic : “current-state”
        characteristic : “obstruction-detected”
    }
```

Garage Door Opener

Services and characteristics example

```
accessory : {
    service1 : “public.hap.accessory-information” {
        characteristic : “serial-number”
        characteristic : “identify”
    }
    service2 : “public.hap.garage-door-opener” {
        characteristic : “target-state”
        characteristic : “current-state”
        characteristic : “obstruction-detected”
    }
    service3 : “public.hap.lightbulb” {
```

Garage Door Opener

Services and characteristics example

```
accessory : {
    service1 : “public.hap.accessory-information” {
        characteristic : “serial-number”
        characteristic : “identify”
    }
    service2 : “public.hap.garage-door-opener” {
        characteristic : “target-state”
        characteristic : “current-state”
        characteristic : “obstruction-detected”
    }
    service3 : “public.hap.lightbulb” {
        characteristic : “on”
    }
}
```

IP Details

IP Details

Remote access through HomeKit

IP Details

Remote access through HomeKit

Bonjour for accessory discovery

IP Details

Remote access through HomeKit

Bonjour for accessory discovery

A single IP accessory may present multiple “accessories”

- Enables bridges
- Bridge is responsible for translating to and from HAP

IP Details

Remote access through HomeKit

Bonjour for accessory discovery

A single IP accessory may present multiple “accessories”

- Enables bridges
- Bridge is responsible for translating to and from HAP

RESTful API to interact with accessories, services, and characteristics

Bluetooth Low Energy Details

Bluetooth Low Energy Details

Remote access through HomeKit

Bluetooth Low Energy Details

Remote access through HomeKit

Apple-defined advertisement data for HAP

Bluetooth Low Energy Details

Remote access through HomeKit

Apple-defined advertisement data for HAP

Accessories are not identifiable from their advertisement data

Bluetooth Low Energy Details

Remote access through HomeKit

Apple-defined advertisement data for HAP

Accessories are not identifiable from their advertisement data

HAP security instead of Bluetooth Low Energy pairing

Protocol Security

End-to-end encryption

Initial setup secured directly between iOS and accessory

Perfect forward secrecy

Standard cryptography

Security Details

Algorithm	Description
Secure Remote Password (SRP)	Encrypts and authenticates initial pairing key exchange
Ed25519	Long-term keys for pairing and authentication
Curve25519	Encrypts initial authentication for each session
HKDF-SHA-512	Per-session ephemeral encryption key derivation
ChaCha20-Poly1305	Encrypts and authenticates HAP data

Defined in pairing specification

Support available at HomeKit lab and through MFi program

Next Steps

Next Steps

MFi program

- Access to HomeKit specifications for protocol, pairing, and profiles
- Development and certification tool
- Available soon

Next Steps

MFi program

- Access to HomeKit specifications for protocol, pairing, and profiles
- Development and certification tool
- Available soon

Learn HomeKit APIs

- Attend HomeKit session and lab

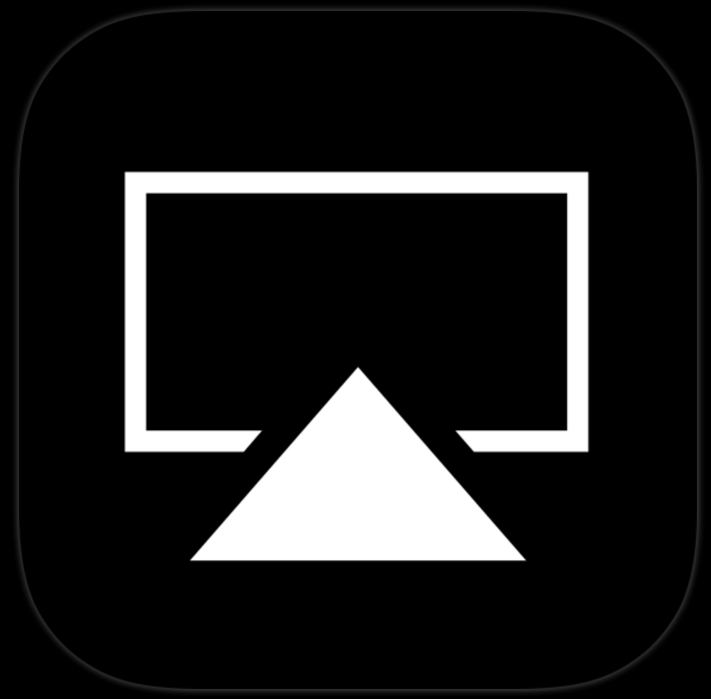
Made for iPod iPhone iPad

AirPlay

AirPlay

Audio handling changes

- Accessory side skew compensation
- Buffer underrun handling



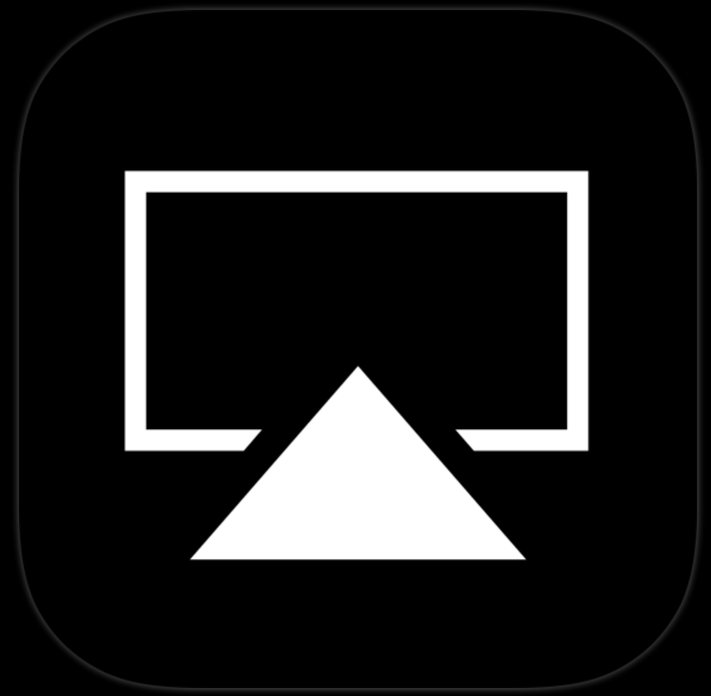
AirPlay

Audio handling changes

- Accessory side skew compensation
- Buffer underrun handling

New reference platform

- BSD on Raspberry Pi



iAP2 and External Accessory Framework

iAP2 and External Accessory Framework

Media playback enhancements for accessories

iAP2 and External Accessory Framework

Media playback enhancements for accessories

Access to playback queue

iAP2 and External Accessory Framework

Media playback enhancements for accessories

Access to playback queue

Play all tracks

iAP2 and External Accessory Framework

Media playback enhancements for accessories

Access to playback queue

Play all tracks

Seek to specific time in music track

iAP2 and External Accessory Framework

Media playback enhancements for accessories

Access to playback queue

Play all tracks

Seek to specific time in music track

Get non-localized bundle identifier for media apps

iAP2 and External Accessory Framework

Media playback enhancements for accessories

Access to playback queue

Play all tracks

Seek to specific time in music track

Get non-localized bundle identifier for media apps

Get progress information during media library synchronization

iAP2 and External Accessory Framework

Additional enhancements for accessories

iAP2 and External Accessory Framework

Additional enhancements for accessories

Simplified time synchronization

iAP2 and External Accessory Framework

Additional enhancements for accessories

Simplified time synchronization

Get detailed charge state information

iAP2 and External Accessory Framework

Additional enhancements for accessories

Simplified time synchronization

Get detailed charge state information

Smaller footprint link layer reference code

iAP2 and External Accessory Framework

Additional enhancements for accessories

Simplified time synchronization

Get detailed charge state information

Smaller footprint link layer reference code

Multiple apps can open sessions for the same External Accessory protocol simultaneously

Wireless Accessory Configuration

Wireless Accessory Configuration

Configure Wi-Fi accessories from within your app

Import External Accessory framework

Implement `EAWiFiUnconfiguredAccessoryBrowserDelegate` protocol

Instantiate `EAWiFiUnconfiguredAccessoryBrowser`

Wireless Accessory Configuration

Search for unconfigured Wi-Fi accessories

Search for all unconfigured Wi-Fi accessories

```
[browser startSearchingForUnconfiguredAccessoriesMatchingPredicate:nil];
```

or filter for a subset of accessories

```
[browser startSearchingForUnconfiguredAccessoriesMatchingPredicate:
    [NSPredicate predicateWithFormat:@"name = ‘Amalgamated Audio’"]];
```

Wireless Accessory Configuration

Delegate callbacks

New accessories were found

```
– (void)accessoryBrowser:(EAWiFiUnconfiguredAccessoryBrowser*)
    browser didFindUnconfiguredAccessories:(NSSet*)accessories
```

Previously found accessory is no longer available

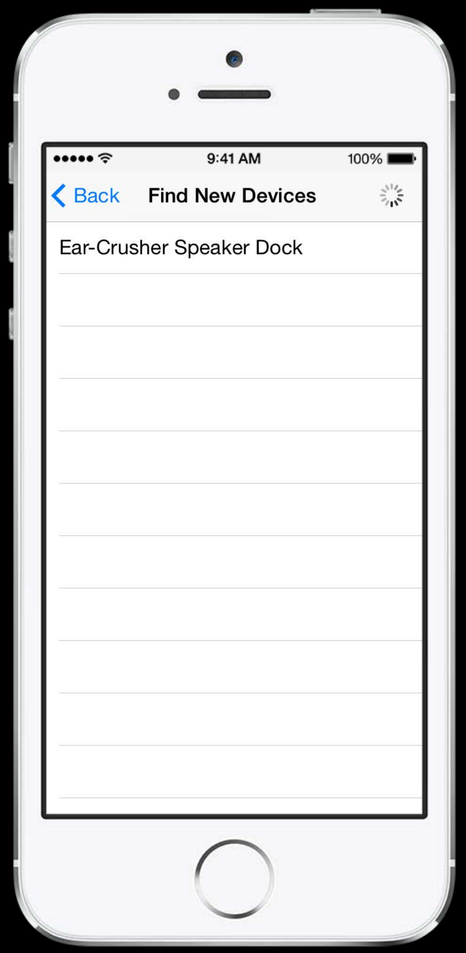
```
– (void)accessoryBrowser:(EAWiFiUnconfiguredAccessoryBrowser*)
    browser didRemoveUnconfiguredAccessories:(NSSet*)accessories
```

State of the browser has changed

```
– (void)accessoryBrowser:(EAWiFiUnconfiguredAccessoryBrowser*)
    browser didUpdateState:(EAWiFiUnconfiguredAccessoryBrowserState)state
```

Wireless Accessory Configuration

User selects an accessory



Wireless Accessory Configuration

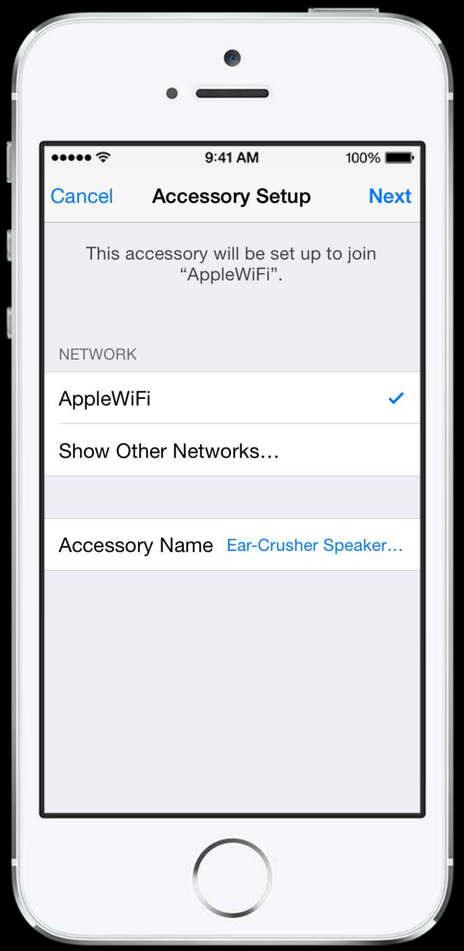
User selects an accessory

Begin configuring the accessory

```
[browser configureAccessory:accessoryToConfigure
   withConfigurationUIOnViewController:self];
```

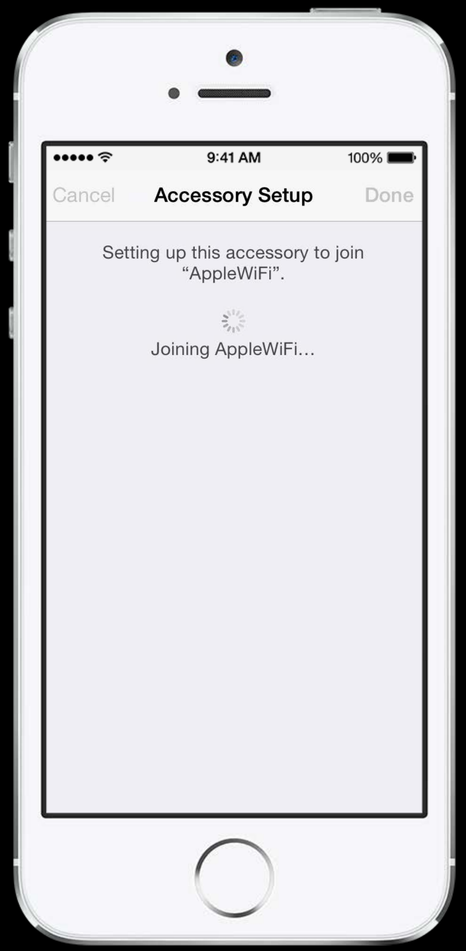
Wireless Accessory Configuration

User configures Wi-Fi accessory



Wireless Accessory Configuration

User configures Wi-Fi accessory

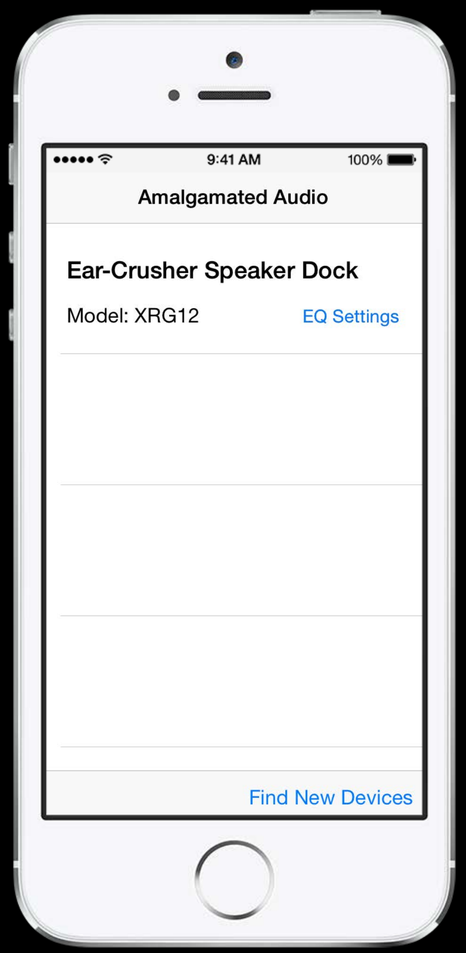


Wireless Accessory Configuration

User configures Wi-Fi accessory

Wireless Accessory Configuration

Configuration complete



Summary

HealthKit

HomeKit

Media playback enhancements for accessories

App-driven Wireless Accessory Configuration

References

MFi Program

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

iBeacon

http://developer.apple.com/ibeacon/

Bluetooth Accessory Design Guidelines for Apple Products

http://developer.apple.com/bluetooth/

Case Design Guidelines for Apple Devices

http://developer.apple.com/resources/cases/

More Information

Stephen Chick
iPhone Evangelist
chick@apple.com

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

Mark Tozer
Desktop Technologies Evangelist
tozer@apple.com

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

Related Sessions

Introducing HealthKit	Mission	Tuesday 10:15AM
What's New in Core Location	Marina	Tuesday 2:00PM
Introducing HomeKit	Marina	Tuesday 4:30PM
Adopting AirPrint	Pacific Heights	Friday 9:00AM
Designing for Game Controllers	Mission	Friday 10:15AM

Labs

• I/O Technologies Lab	Core OS Lab A	Tuesday 9:00AM
• Accessories and I/O Technologies Lab	Core OS Lab A	Tuesday 11:30AM
• I/O Technologies Lab	Core OS Lab A	Thursday 11:30AM
• Accessories and I/O Technologies Lab	Core OS Lab A	Thursday 12:45PM
• HomeKit Lab	Services Lab A	Thursday 12:45PM
• HealthKit Lab	Services Lab B	Friday 9:00AM

WWDC14