Introspection into HomeKit
The Kit-pattern:

- Functionality
- Database
- Application

HealthKit, AddressBook etc.
Introspection into HomeKit

The Kit-pattern:

- Container
- Unit

AddressBook: book and person

Homekit: room and device
The Kit-pattern (developer’s perspective):

- manager (singleton instance)
- class-tree mirrors devices (abstract -> real)
- Database reflects instances of real objects with instances of classes

as always: Maybe point to discussions
Introspection into HomeKit

The Kit-pattern (missing components):

- There is no default application.
- Every single device needs to be acquired.
- Devices are not abstract.

Missing abstraction -> SIRI-commands
and so on:

- Certification is mandatory for devices.
- WiFi-devices are not generally supported (?).
- Secure Bluetooth-LE causes serious lags.
- Apple developer program is required, even for simulations.

Some kind of conclusion: Developing with HomeKit requires a collaboration with manufacturers of certified devices.
Bridges

Bridging enables uncertified devices as HomeKit-devices.

- **HomeBridge**
  Open source application, based on nodeJS
  
  [https://github.com/nfarina/homebridge](https://github.com/nfarina/homebridge)

- **Broadcom-WICED**
  Wireless Internet Connectivity for Embedded Devices.
  
The HomeKit Accessory Simulator is an app that provides virtual home accessories that can be used to test the communication of an HomeKit-application to an accessory or a collection of accessories.

The app is not included in Xcode, it is a separate download.

Once running, the app distributes its simulated devices to real iOS-devices.
Some links:

Devices:  
https://support.apple.com/en-us/HT204903

Example code:  

No presentation of code examples at all.

The example from Apple is sufficient.  
Instead some slides out of a previous talk demonstrate an alterate use case as example with Particle cores and iOS instead of HM.
Alternate approach

As example: Particle (formerly Spark) core

HomeKit: Works with HomeBridge
+ More available interfaces
More available protocols (e.g. MQTT)
+ Different clouds (local & global)
Web, IFTTT etc.

Core is not a home-device.
Some programming and making skills are required.
Introspection into HomeKit

Cloud access:

curl https://api.spark.io/v1/devices/0123456789abcdef01234567/brew \
-d access_token=98769876987698769876987698769876987698769876987698769876
Introspection into HomeKit

**General**

Main Screen

Spark Cores

Prefs for Social
Introspection into HomeKit

Colors

All LED colors

single Color

Pattern
Introspection into HomeKit

FTP upload

FTP Server

FTP Directories
Introspection into HomeKit

Social

Facebook

MailComposer

Twitter
Introspection into HomeKit

Points of discussion

- Home vs. real estate
  - user vs. location
- Security
  - world-wide vs. local

- Database
  - changes, triggers, different apps and different users
- Application
  - Third party apps are available on the app-store.
  - The new demo-example code from Apple works.
  - eventual in iOS 9
Introspection into HomeKit

Thank you!