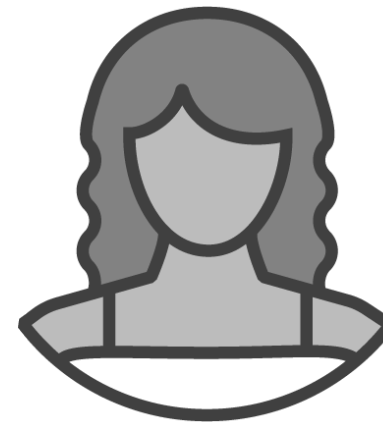
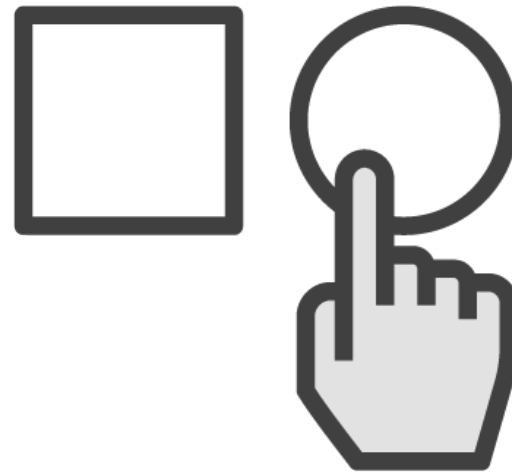
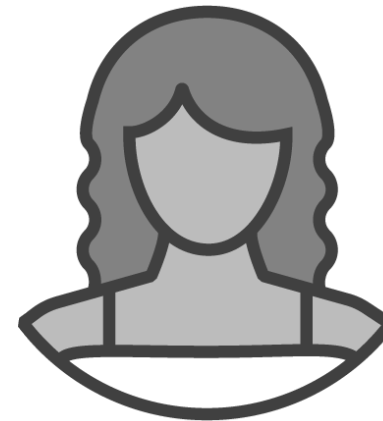
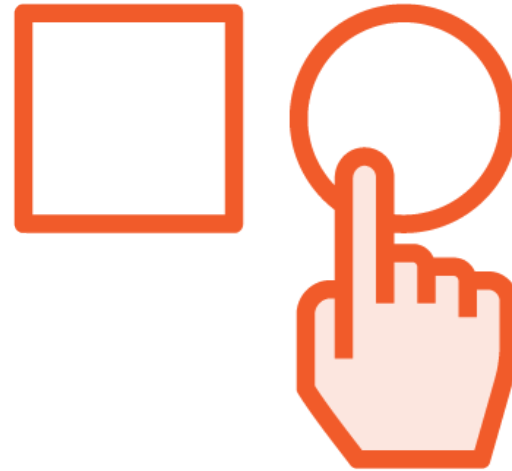
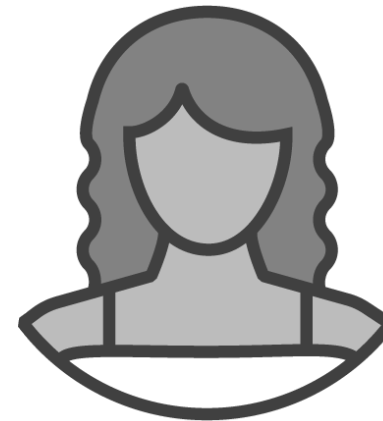
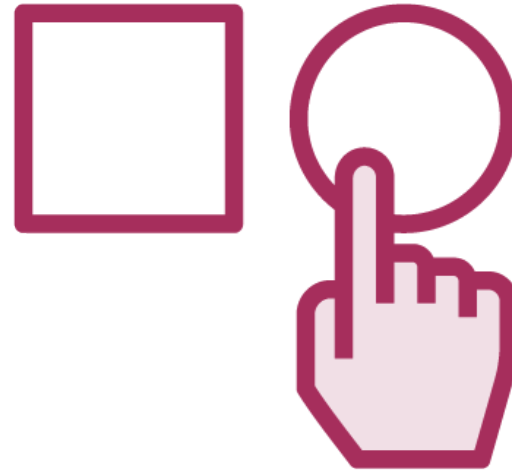
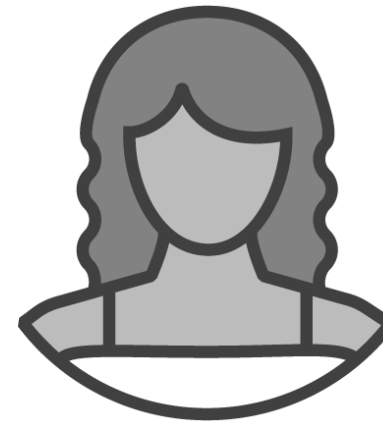
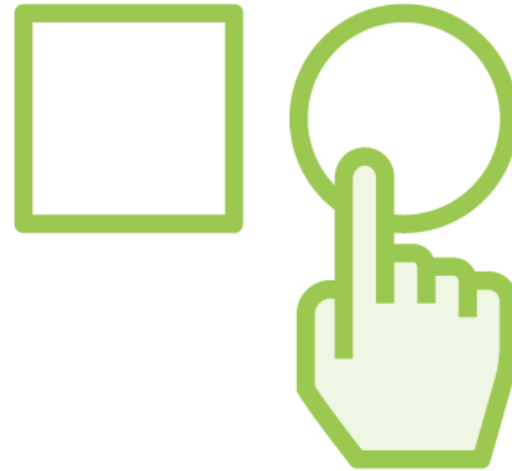


Saving Settings and Configuration







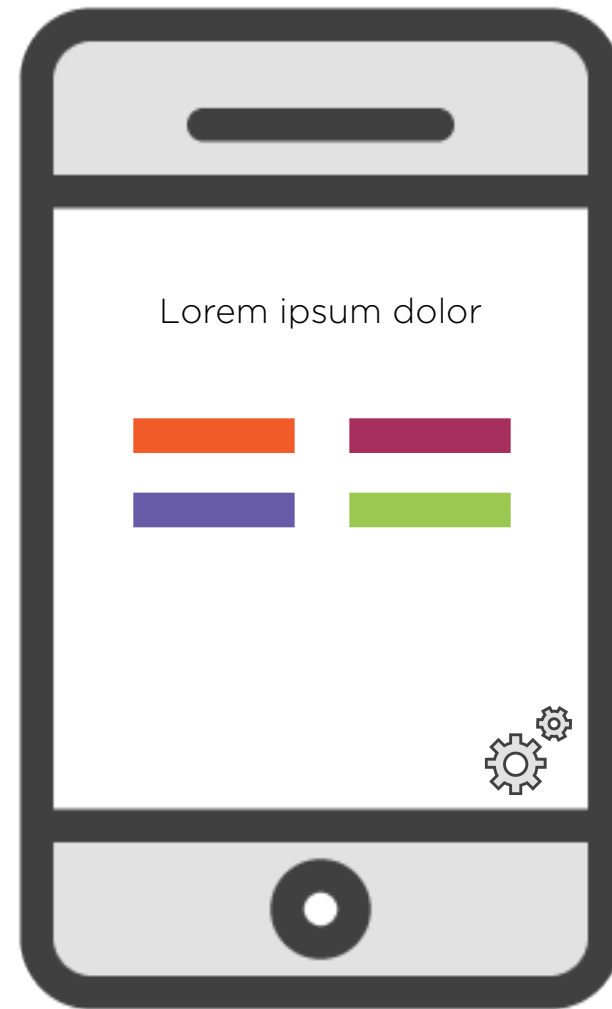


App Settings and Configuration

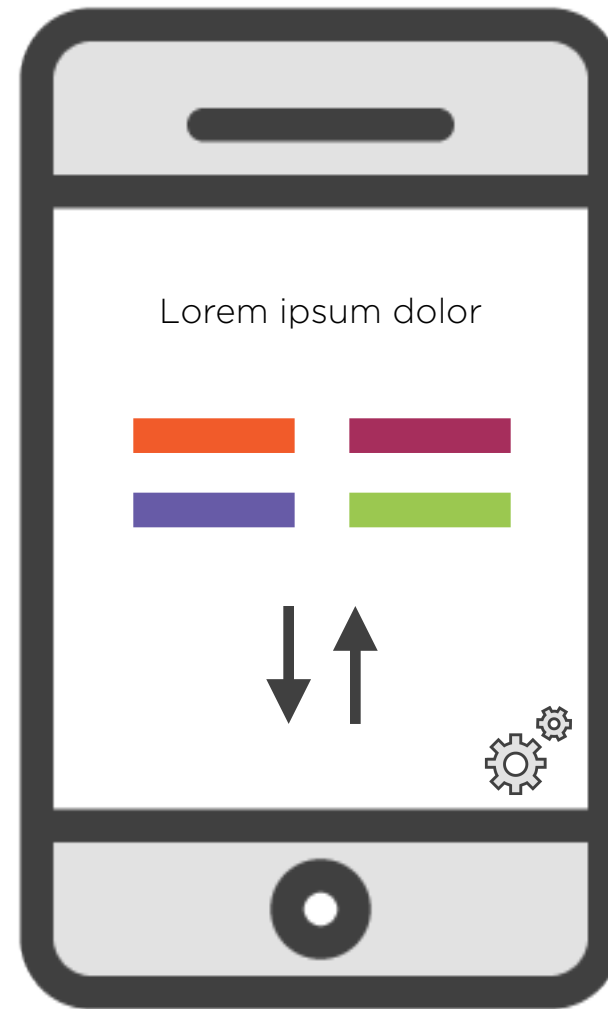
A —————● A



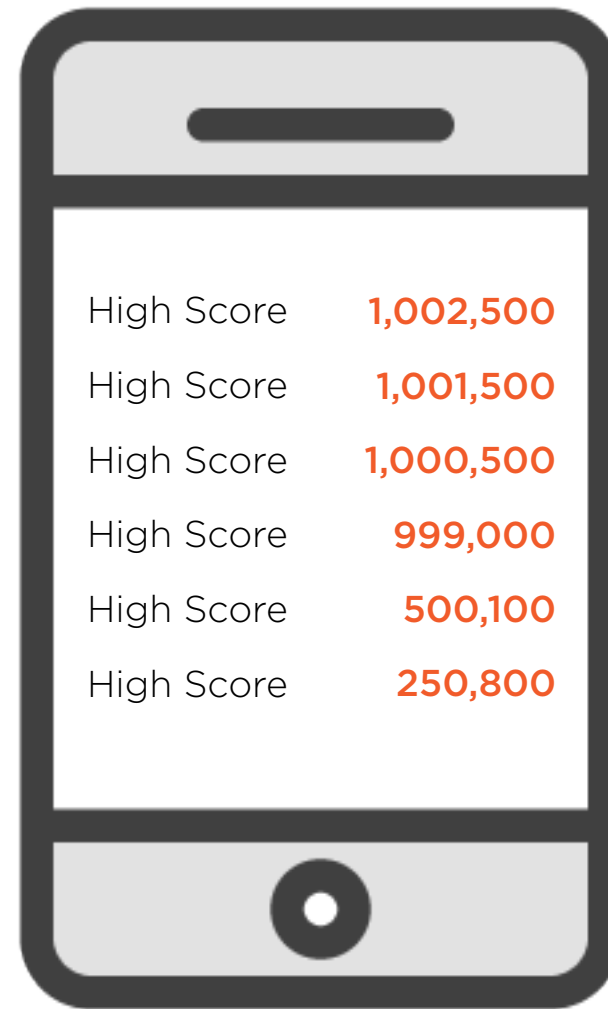
App Settings and Configuration



App Settings and Configuration



App Settings and Configuration



App Settings and Configuration

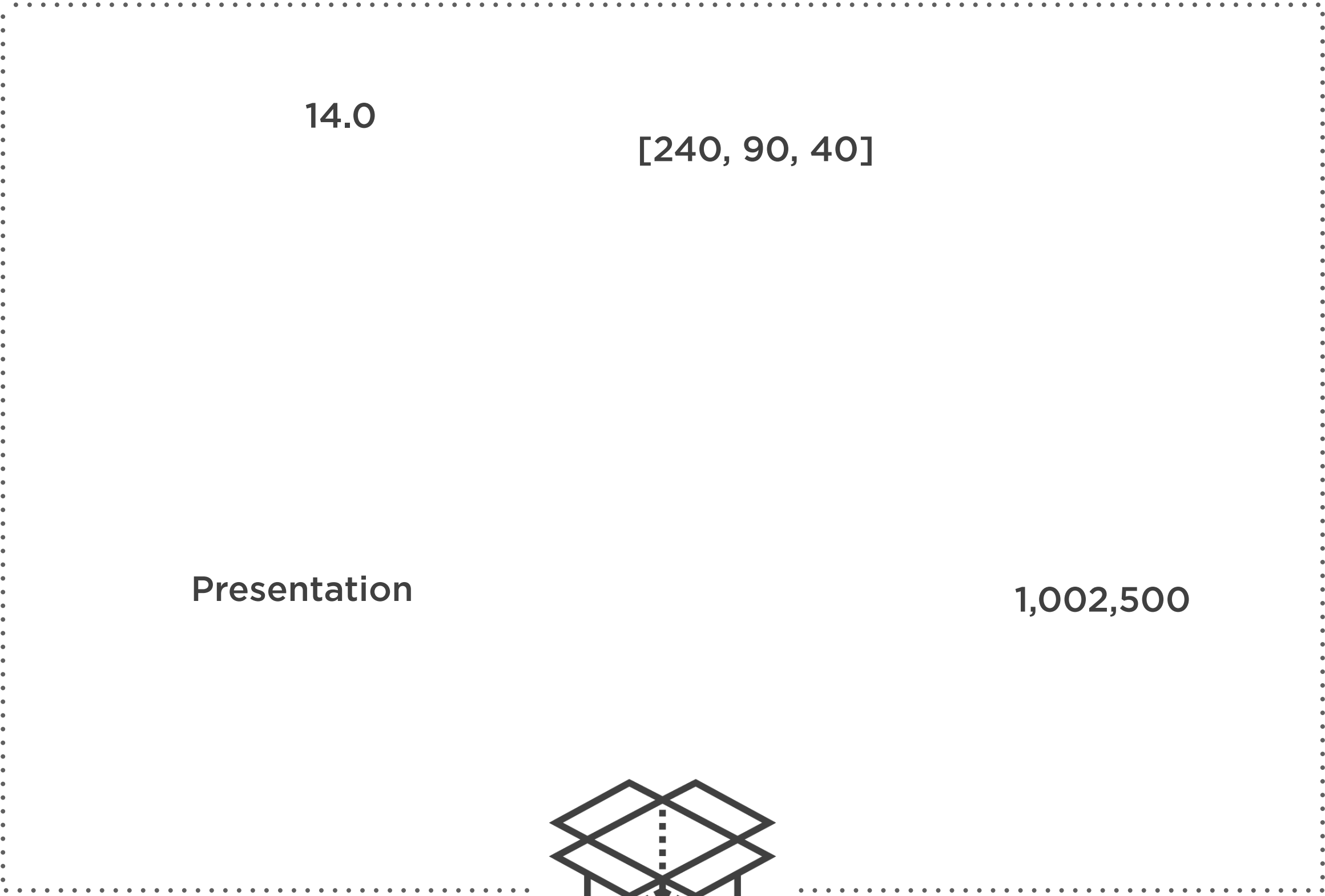


App Settings and Configuration



App Settings and Configuration





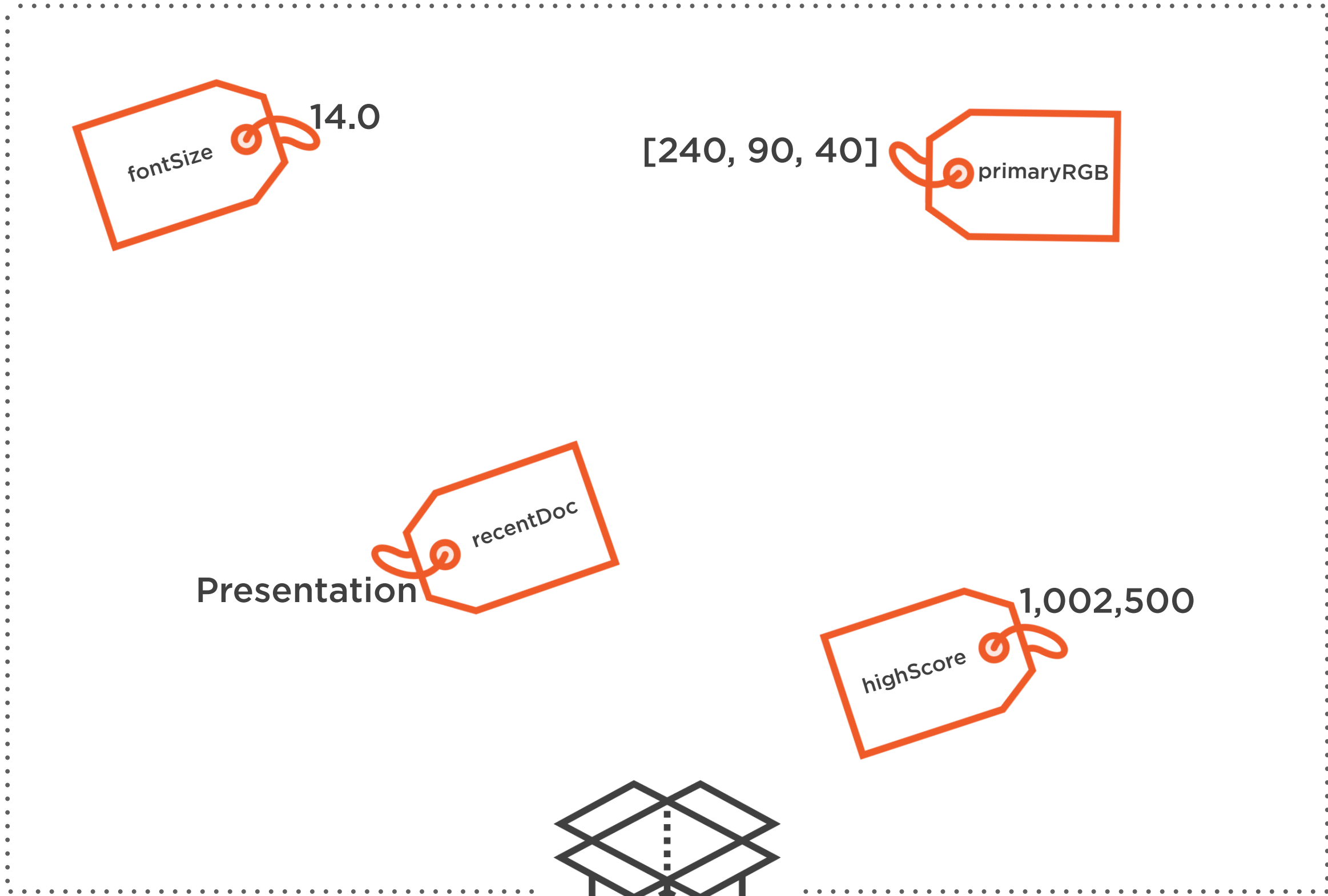
14.0

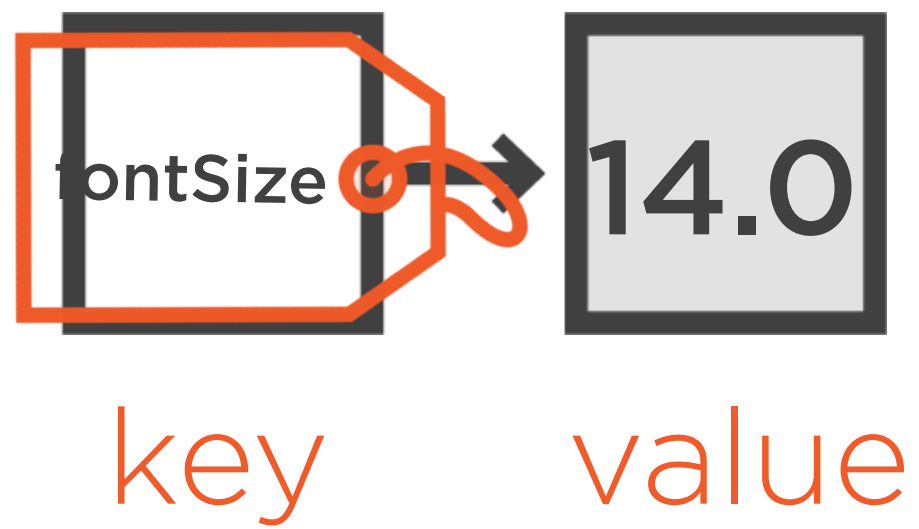
[240, 90, 40]

Presentation

1,002,500









Settings & Configuration

User Defaults

iCloud Key-Value
Store

Property List
(plist)

Angles for Analysis



Where data is saved



Who maintains the technology



Data model complexity



Data Size



Price

Angles for Analysis

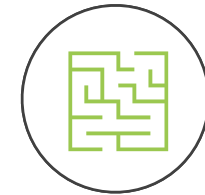
Where data is saved



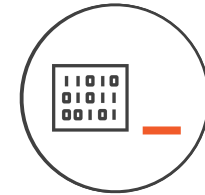
Who maintains the technology



Data model complexity



Data Size



Price



Angles for Analysis

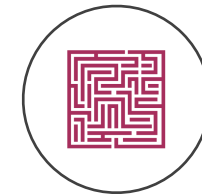
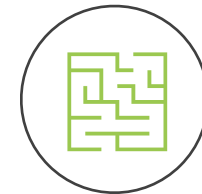
Where data is saved



Who maintains the technology



Data model complexity



Data Size



Price

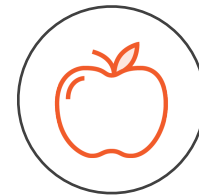


Angles for Analysis

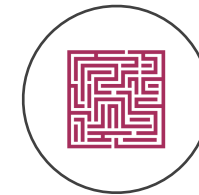
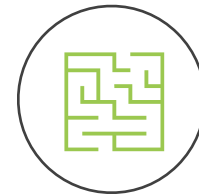
Where data is saved



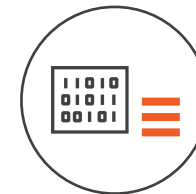
Who maintains the technology



Data model complexity



Data Size



Price

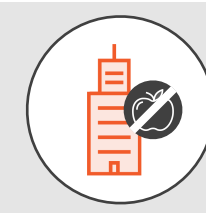


Angles for Analysis

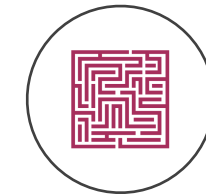
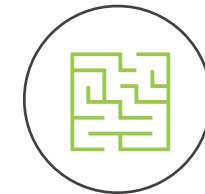
Where data is saved



Who maintains the technology



Data model complexity



Data Size



Price

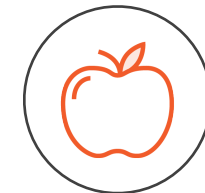


Angles for Analysis

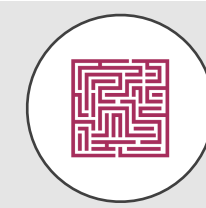
Where data is saved



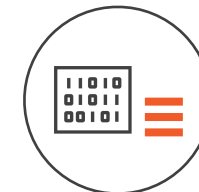
Who maintains the technology



Data model complexity



Data Size



Price



Angles for Analysis

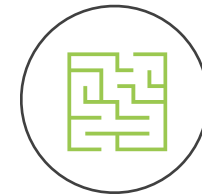
Where data is saved



Who maintains the technology



Data model complexity



Data Size



Price

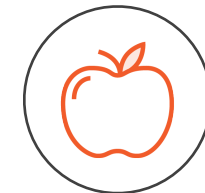


Angles for Analysis

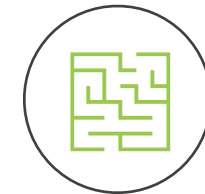
Where data is saved



Who maintains the technology



Data model complexity



Data Size



Price



Angles for Analysis

Where data is saved



Data Size



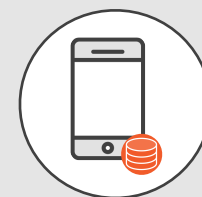


Settings & Configuration

User Defaults

iCloud Key-Value
Store

Property List
(plist)



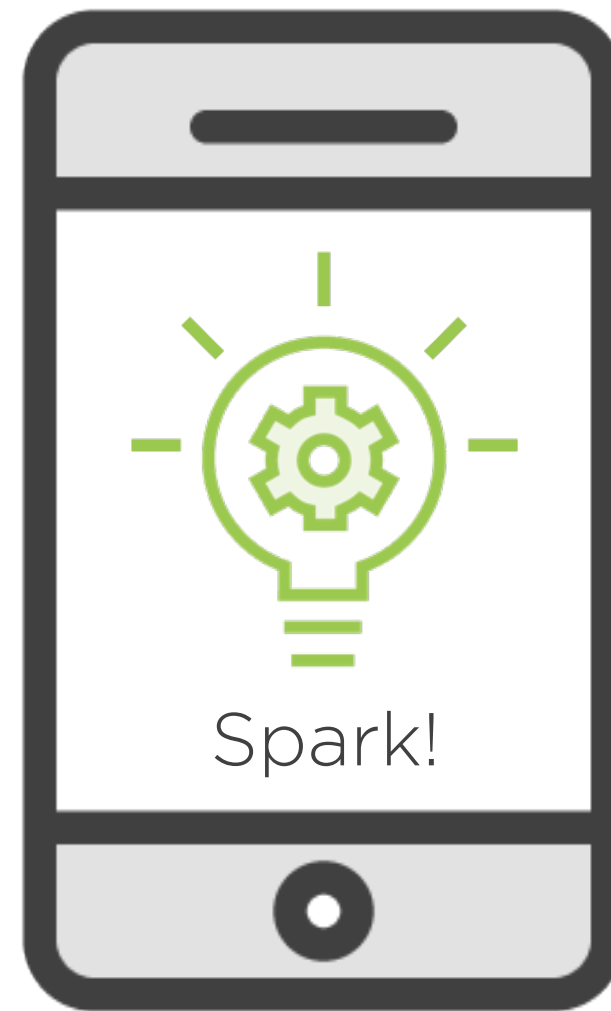


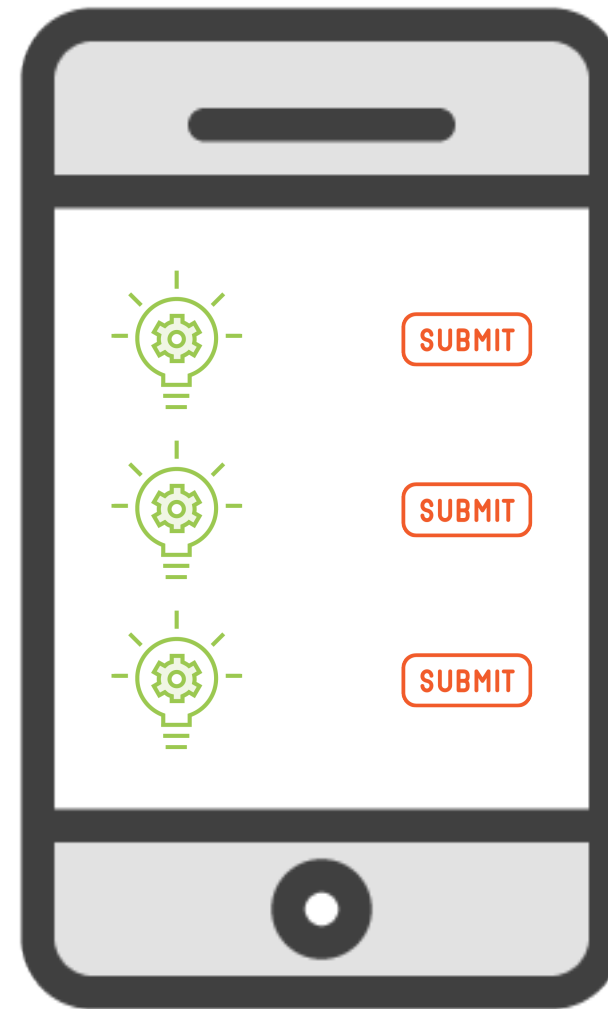
**What does it take to set these up
in an app?**

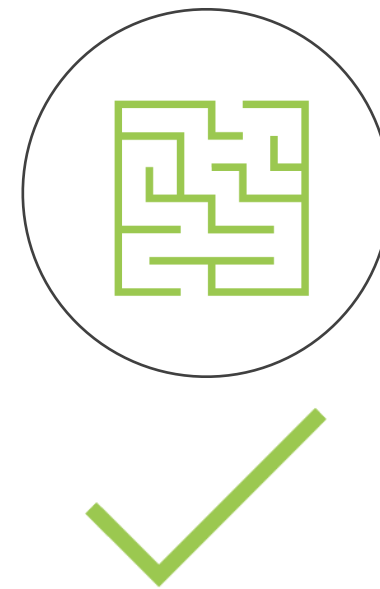
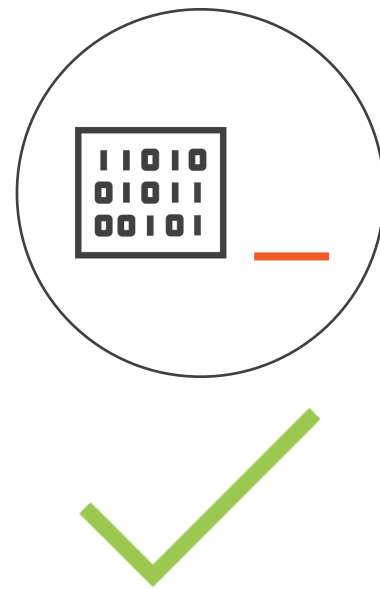


**How does it feel to use them in
code?**

Managing Preferences with UserDefaults







What would it take to persist this data with `UserDefaults`?

set()

set()

set()

12:15



Name

E-mail

Department

Save

```
forKey: "name" )
```

```
forKey: "email" )
```

```
forKey: "department" )
```

```
set(      nameTextField.text, forKey: "name"      )  
set(      emailTextField.text, forKey: "email"      )  
set( departmentTextField.text, forKey: "department" )
```

Big Picture Takeaways



Simple, small data sets



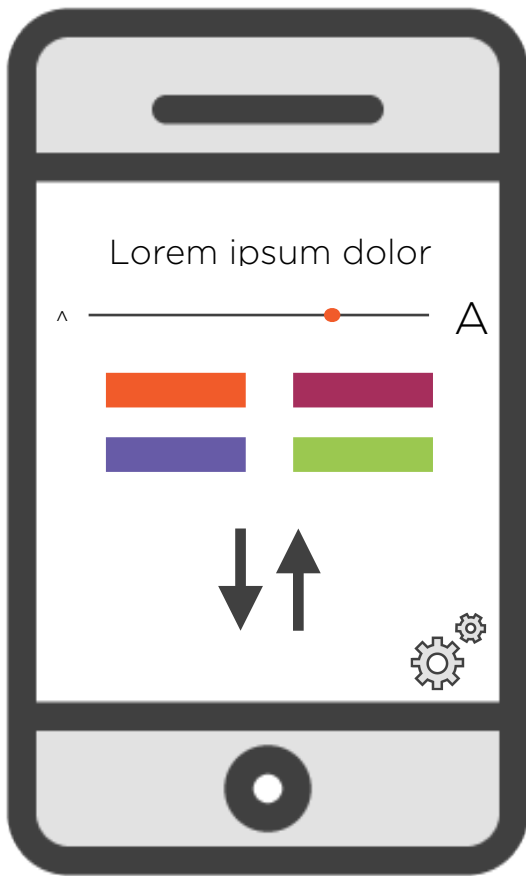
Entire UserDefaults data store loads into memory on app launch

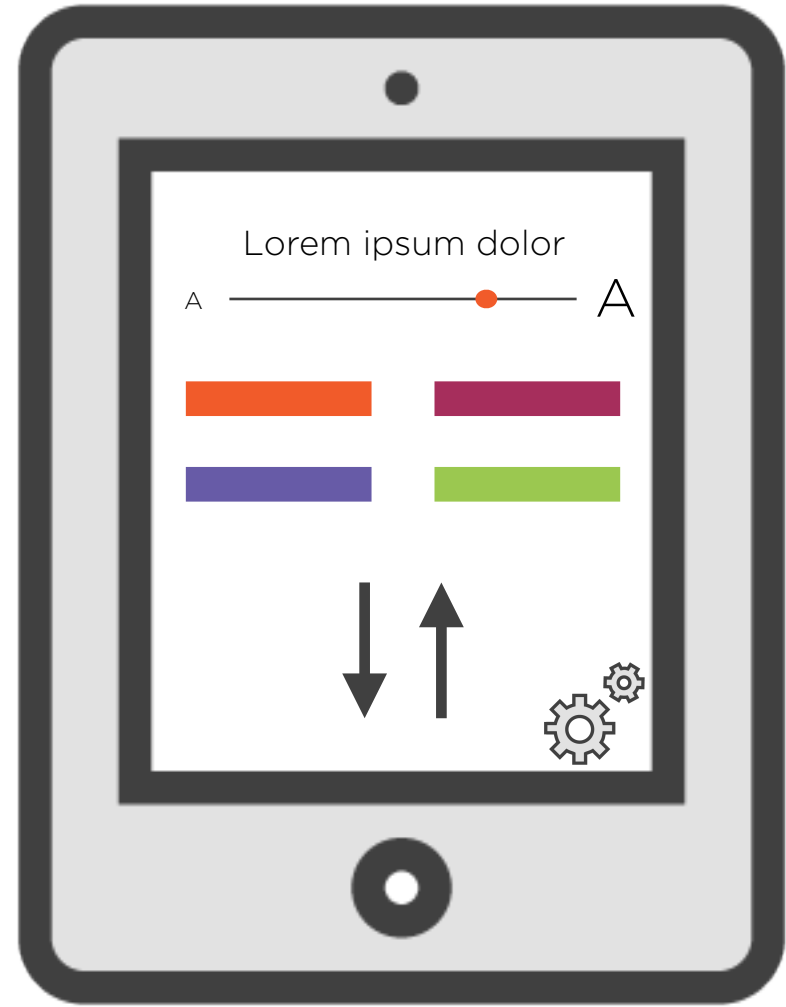
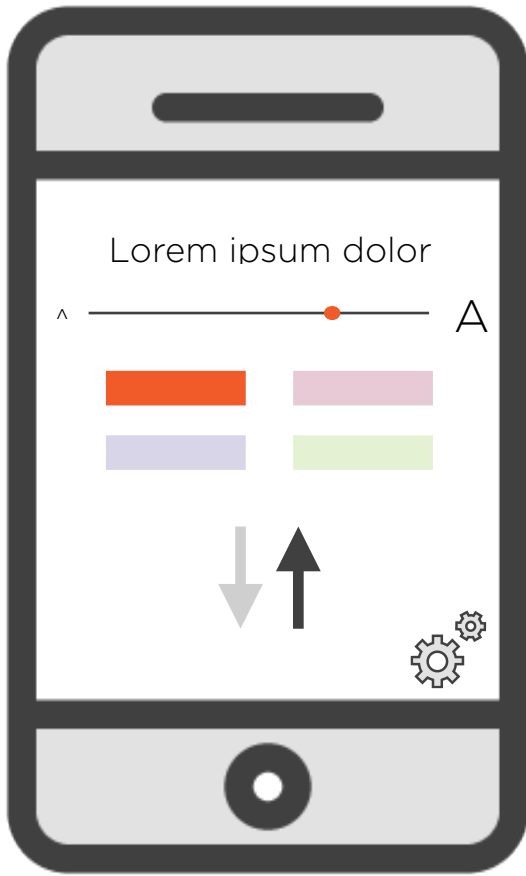


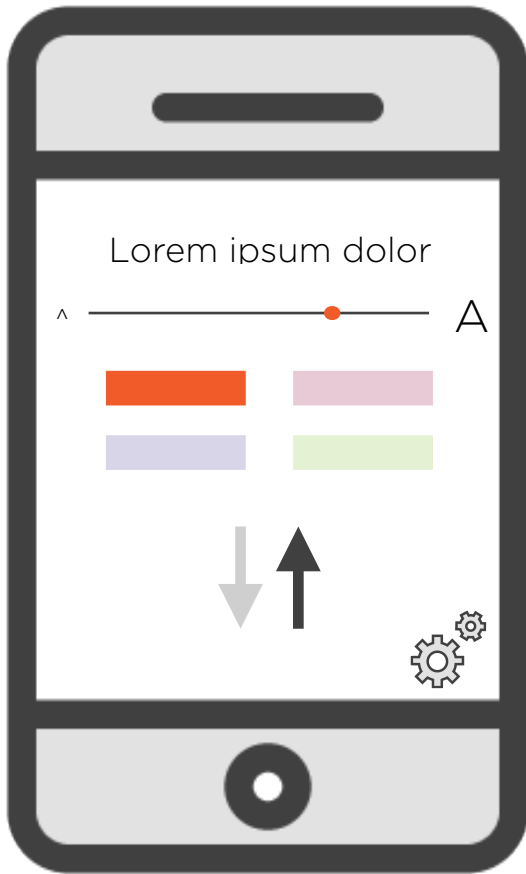
API associates values to keys and are stored locally on the user's device

What if your users are expecting the values they save to appear on their other devices automatically?

Synchronizing Settings with iCloud Key-Value Store









Settings & Configuration

User Defaults

iCloud Key-Value
Store

Property List
(plist)

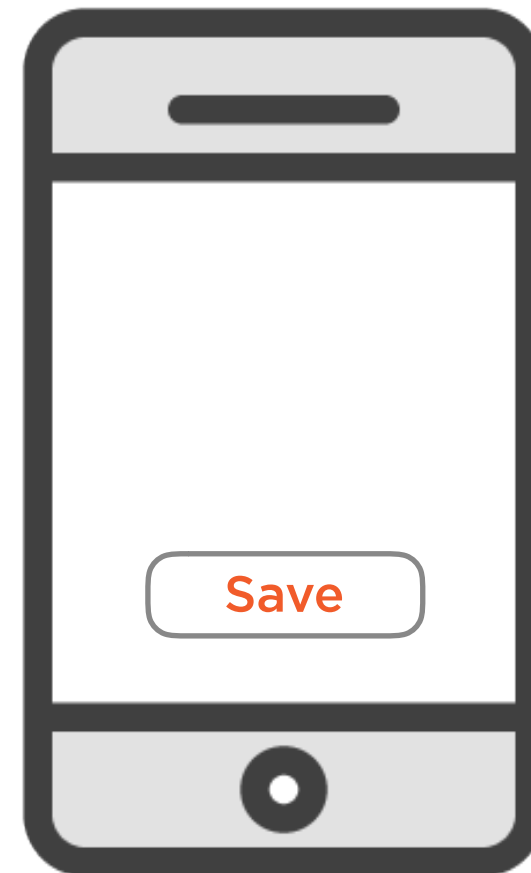


Settings & Configuration

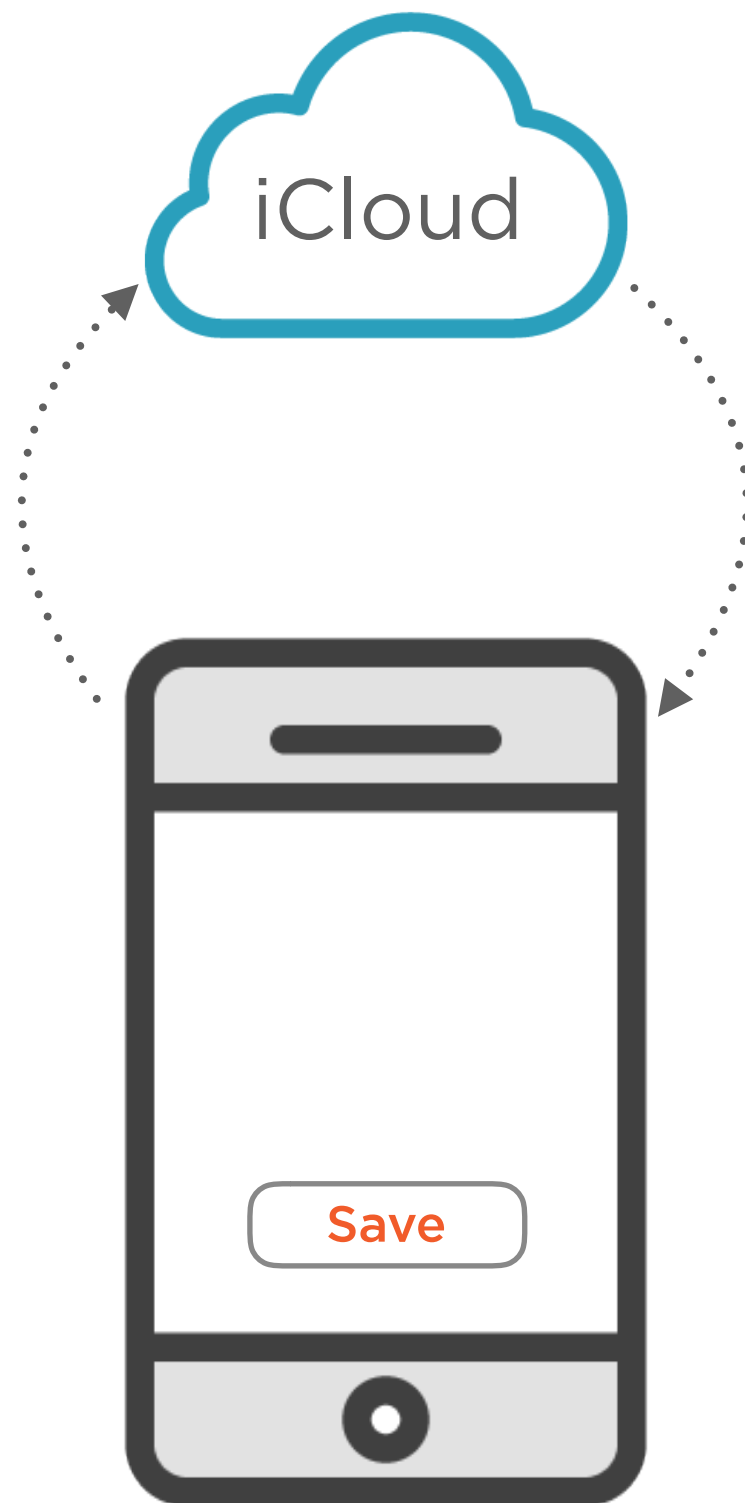
User Defaults

iCloud Key-Value
Store

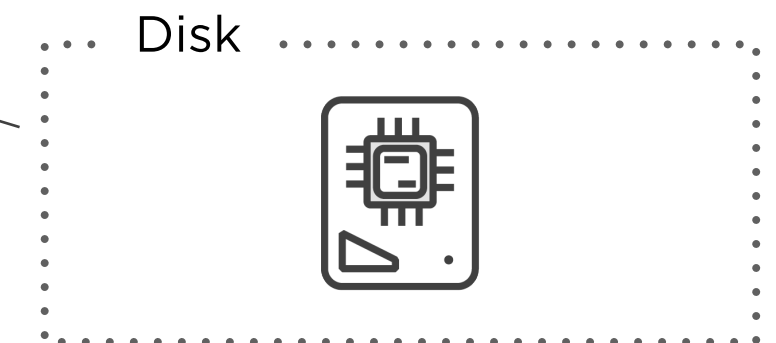
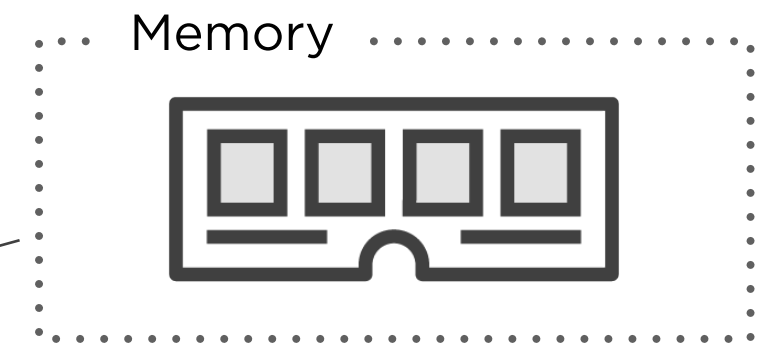
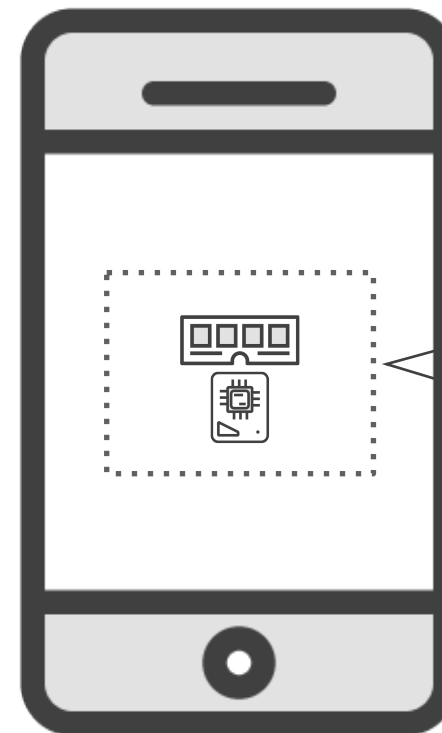
Property List
(plist)

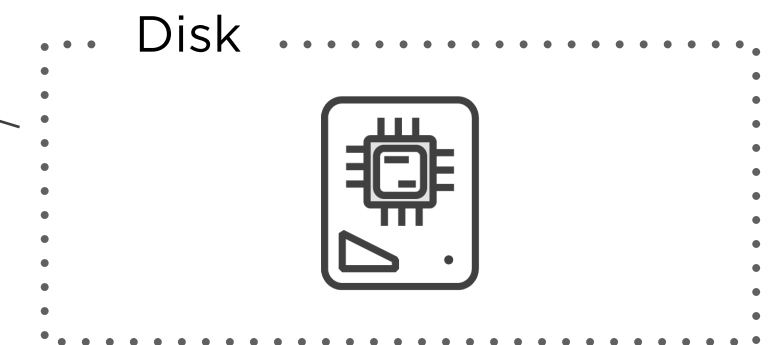
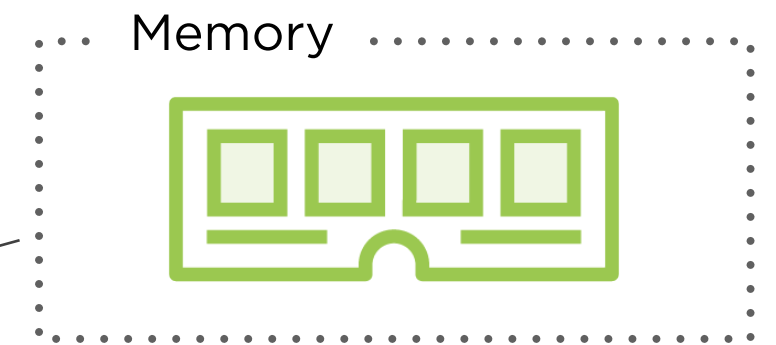
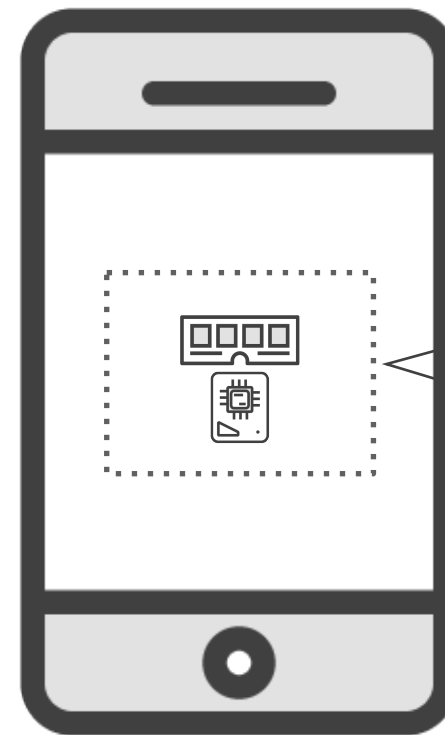


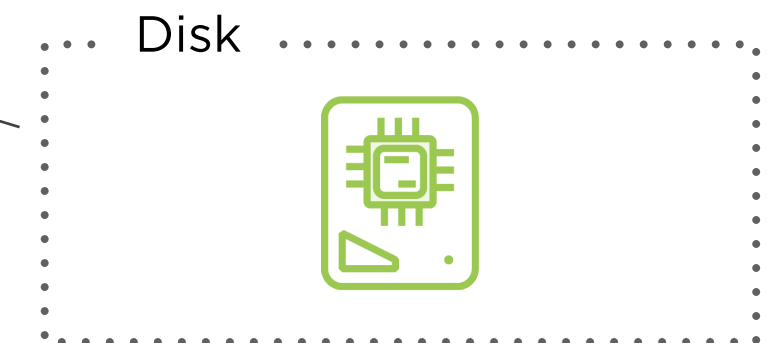
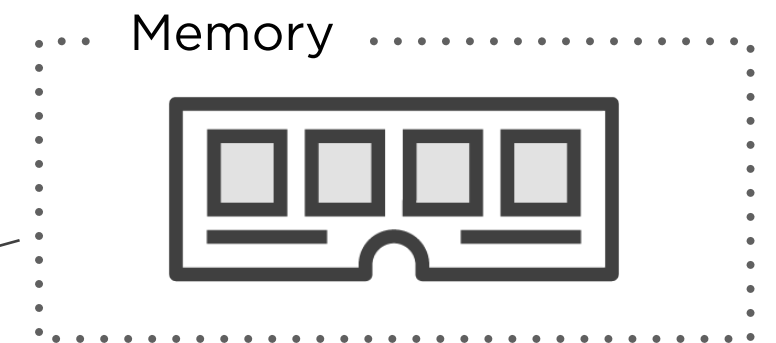
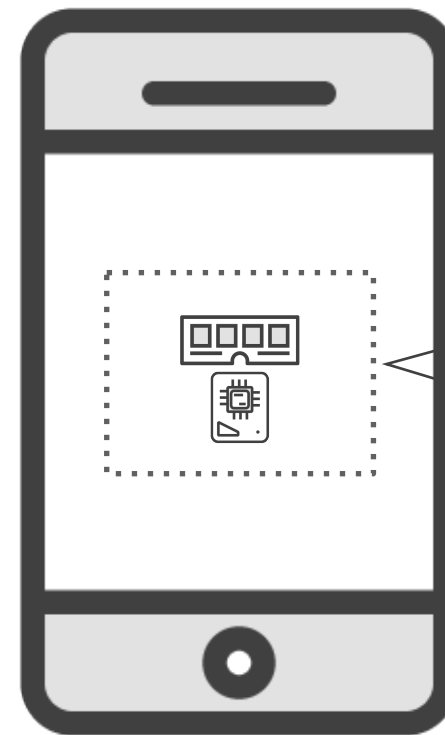


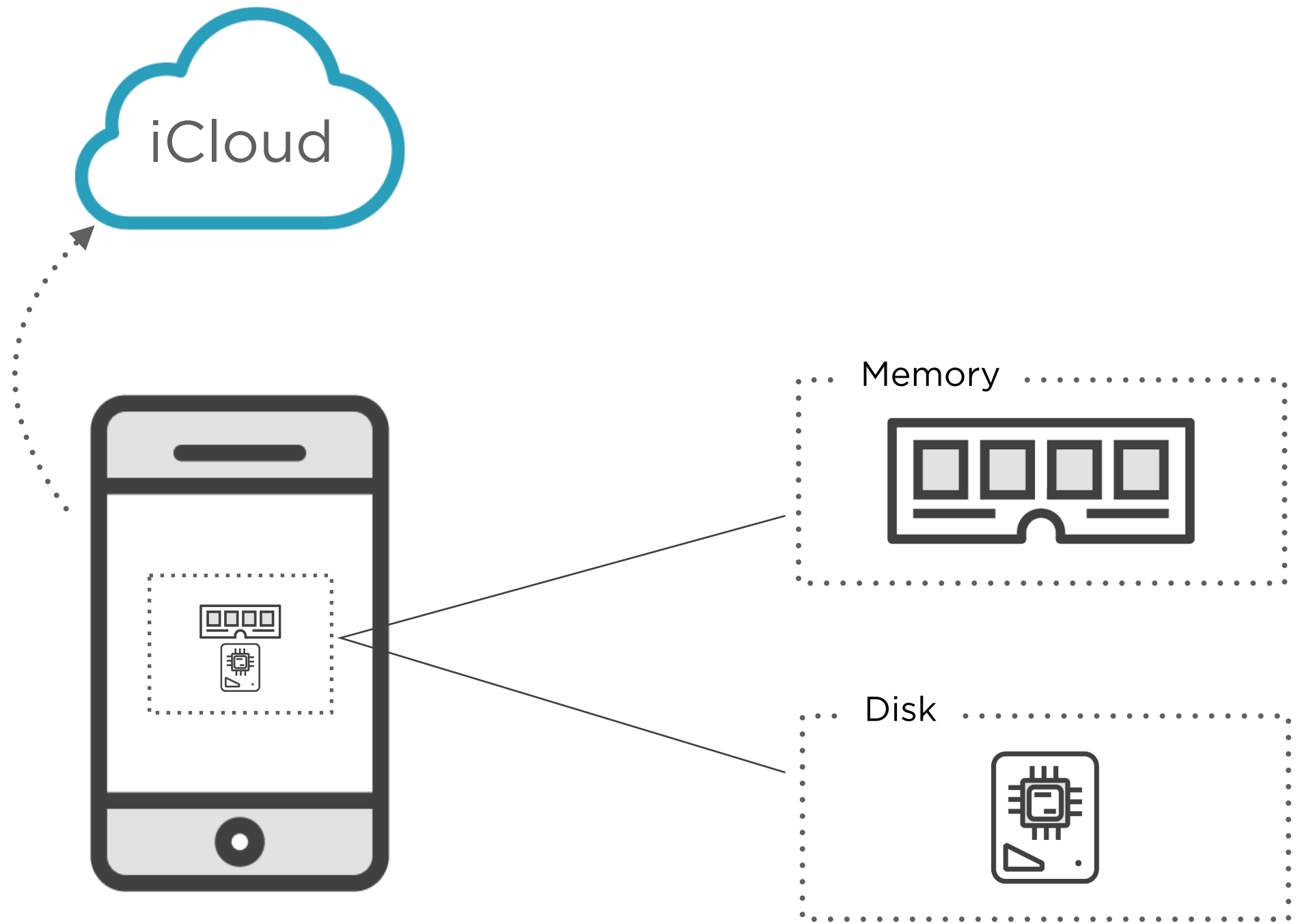


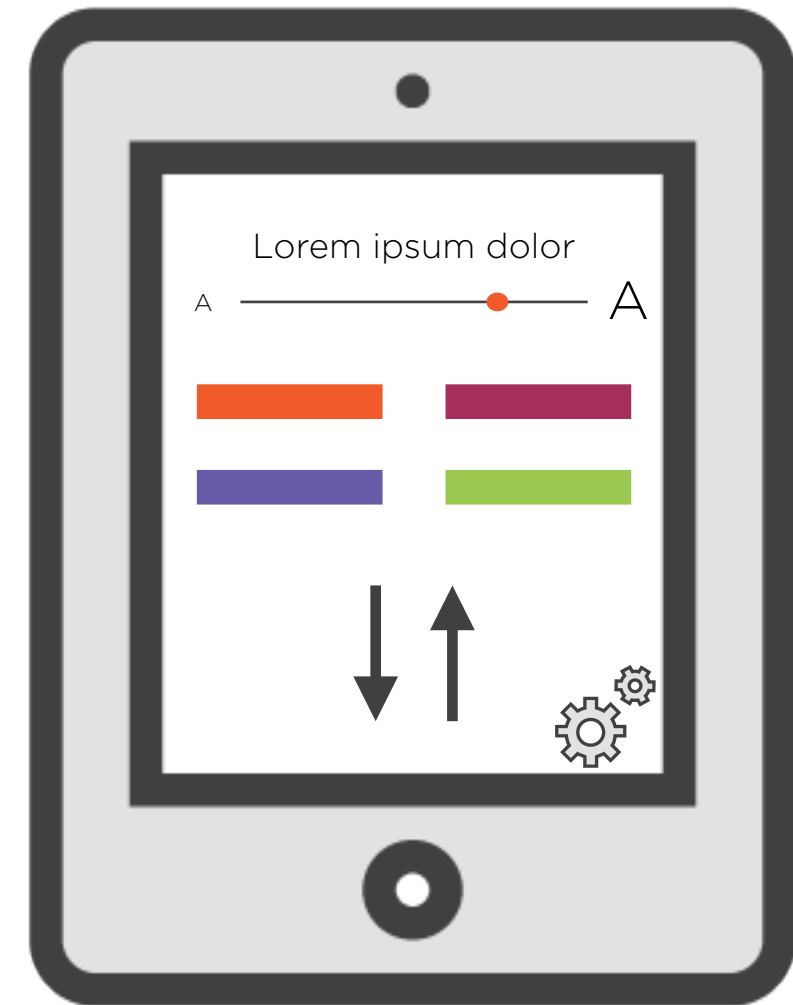
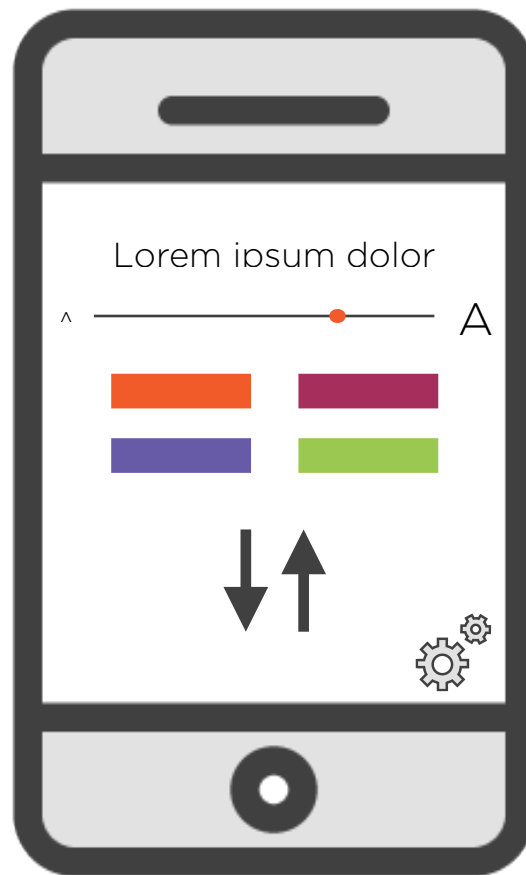
How do the user's other devices
know that there's a new value?

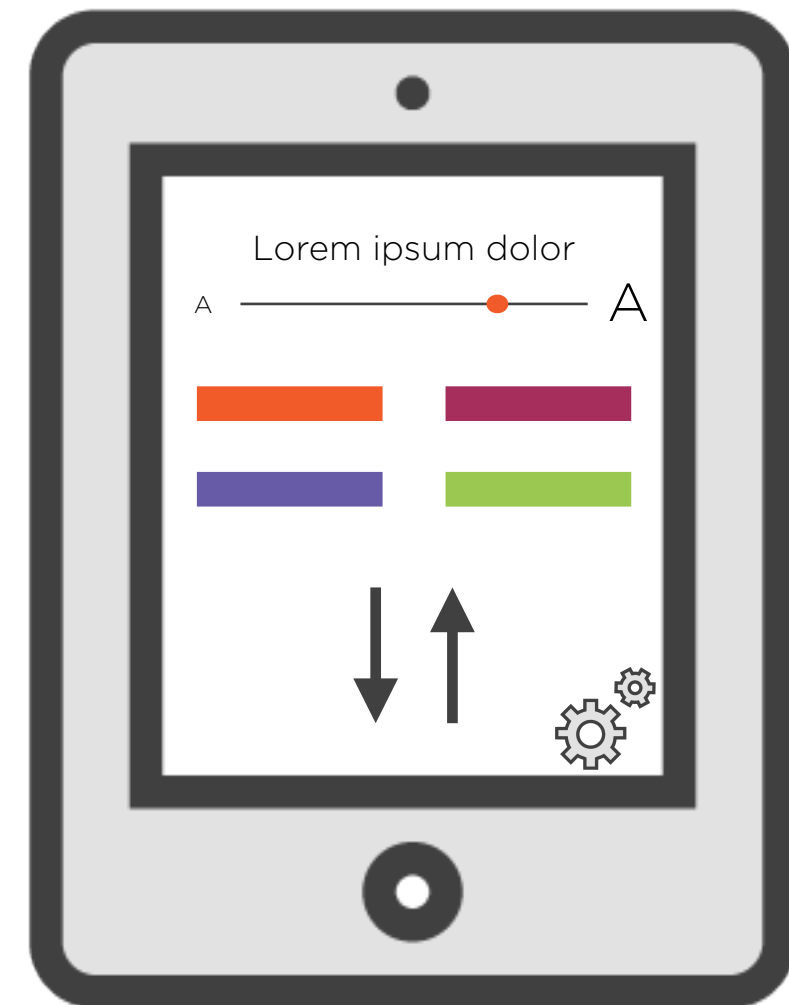
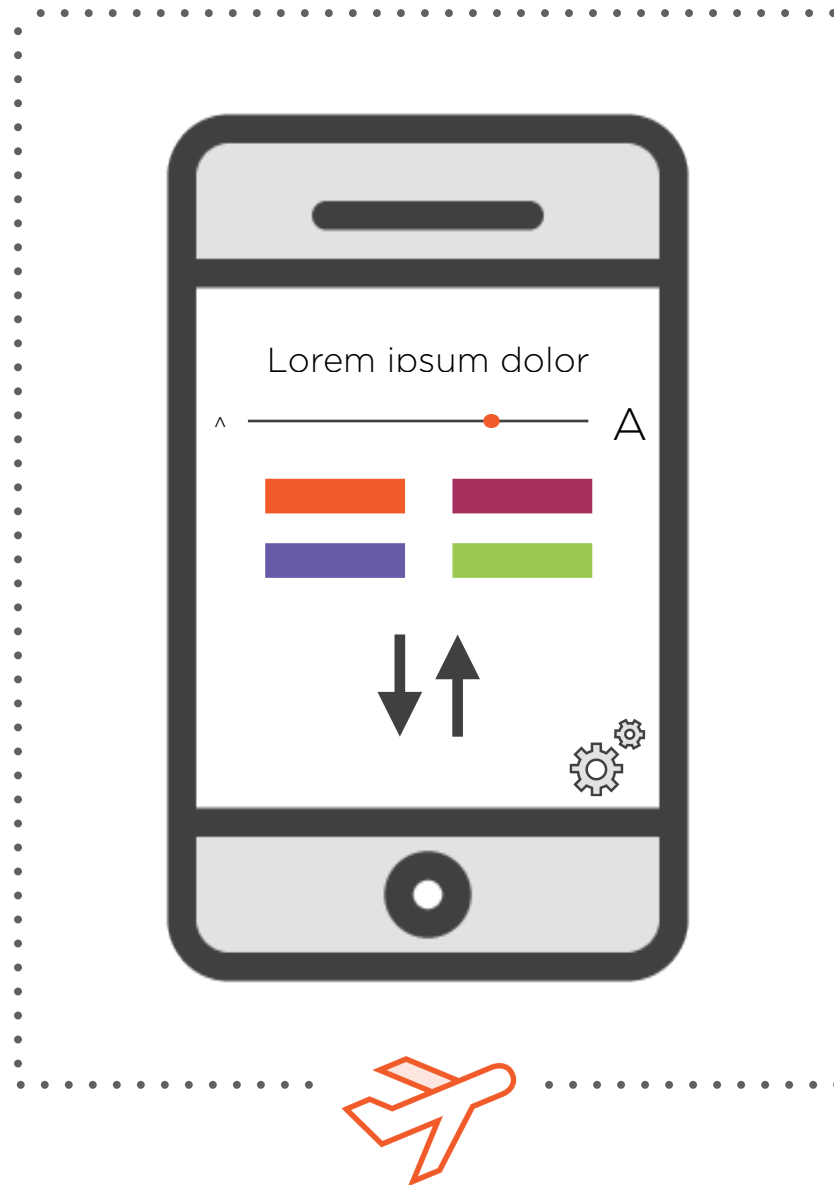


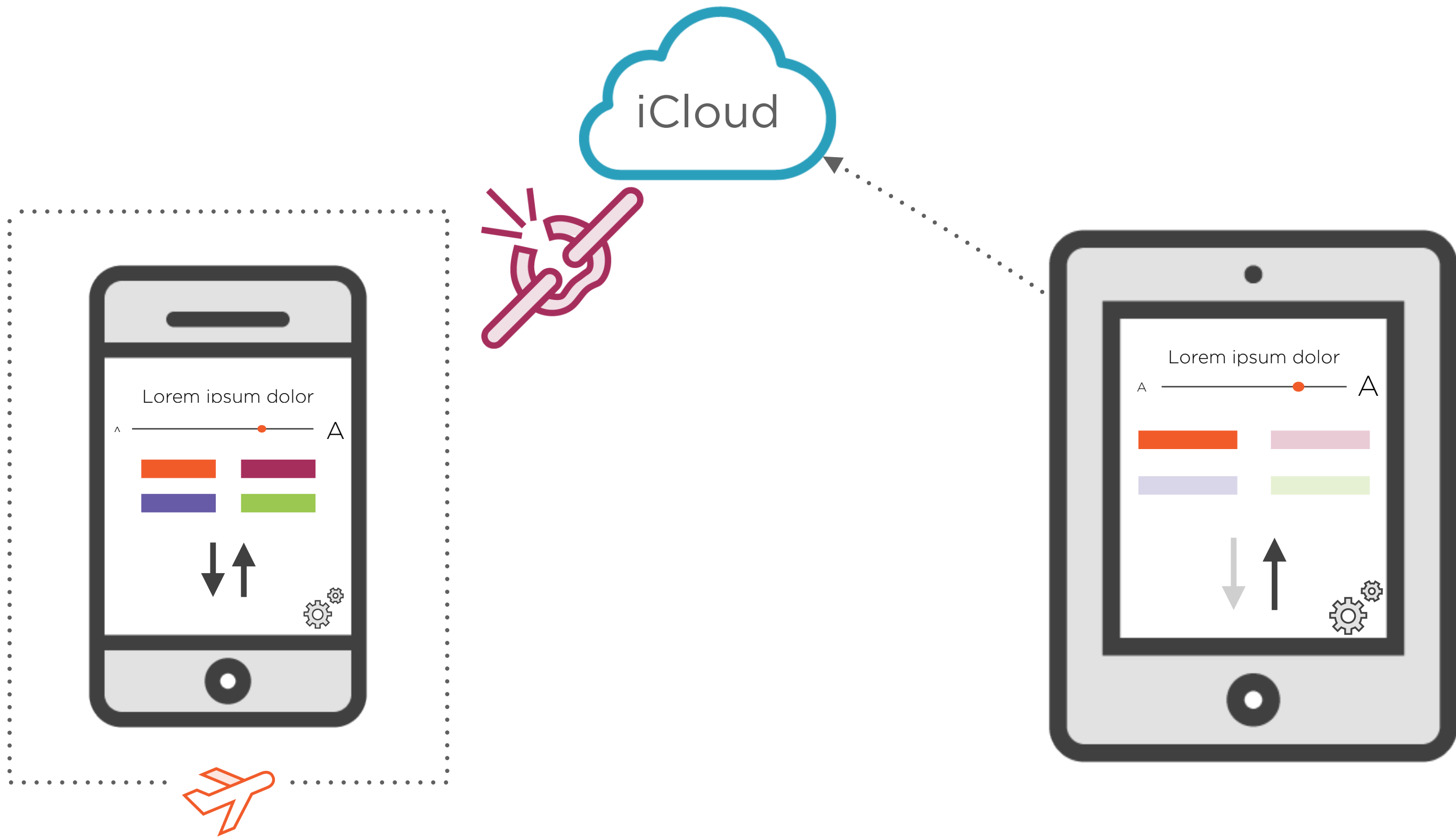


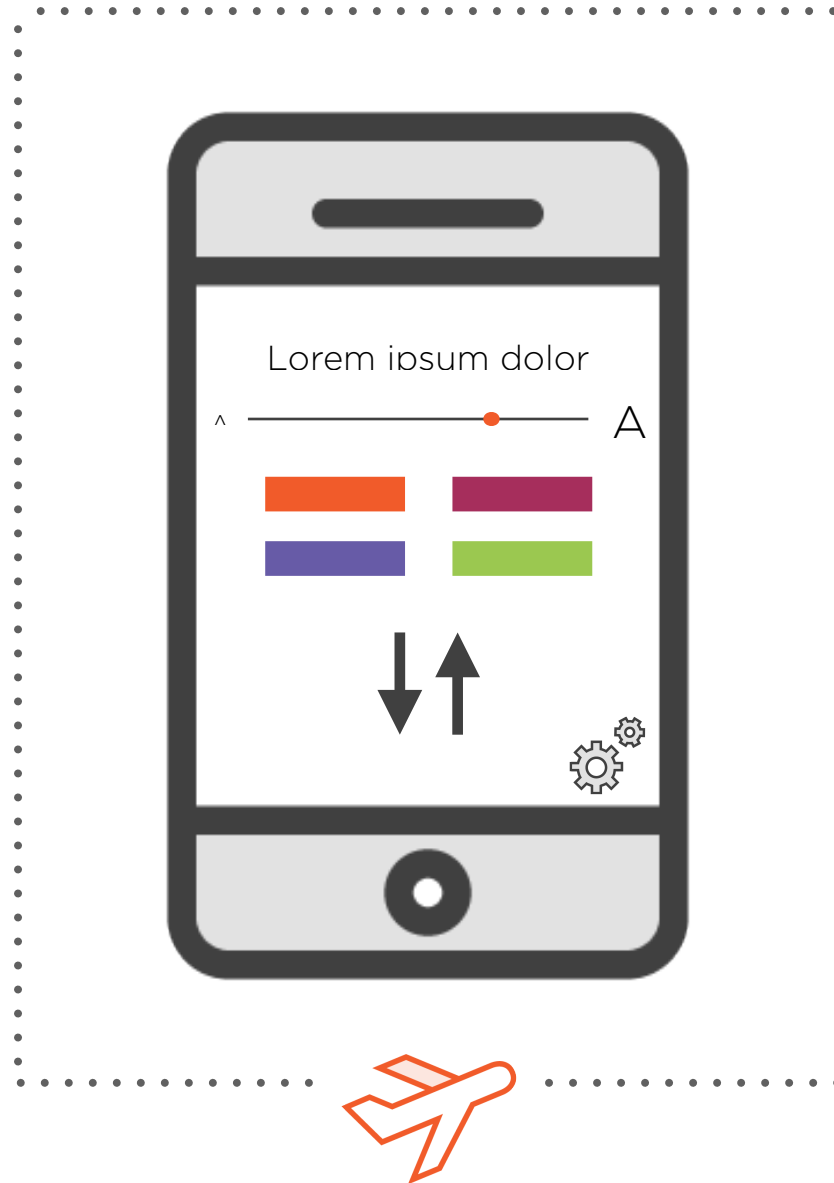


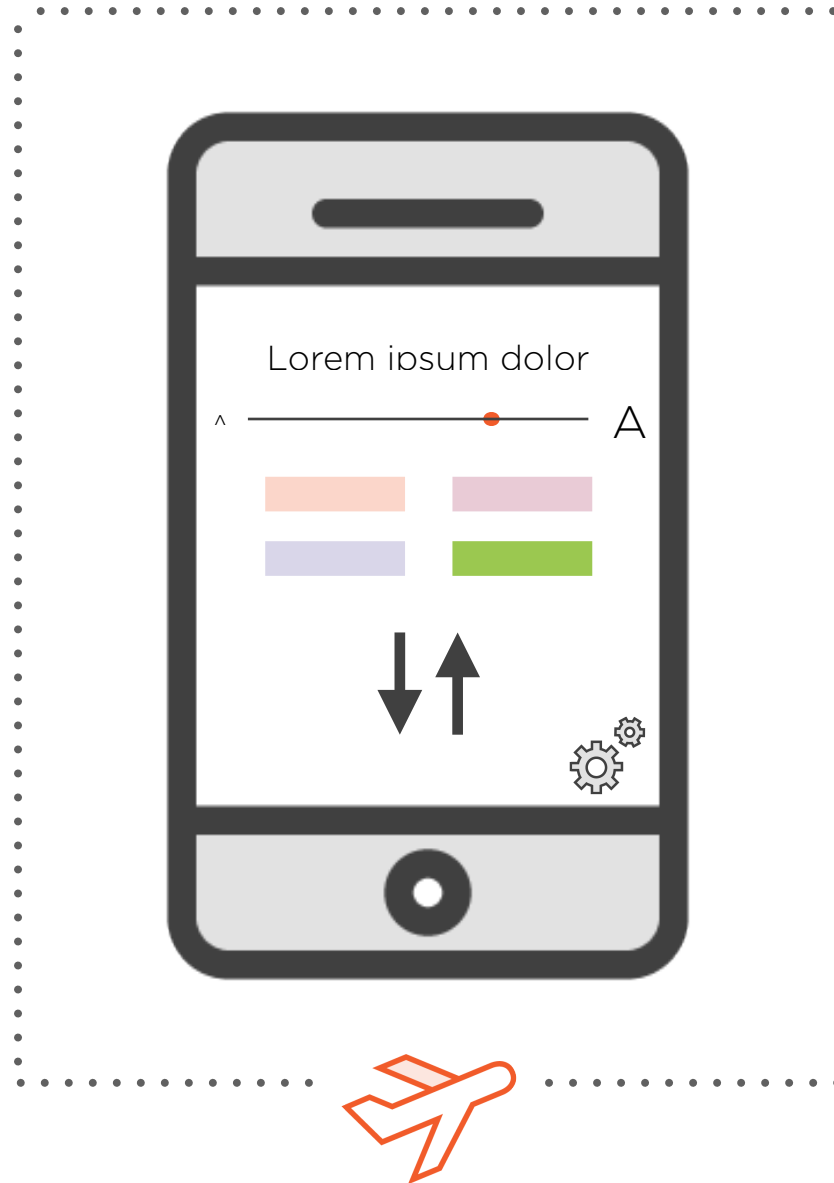


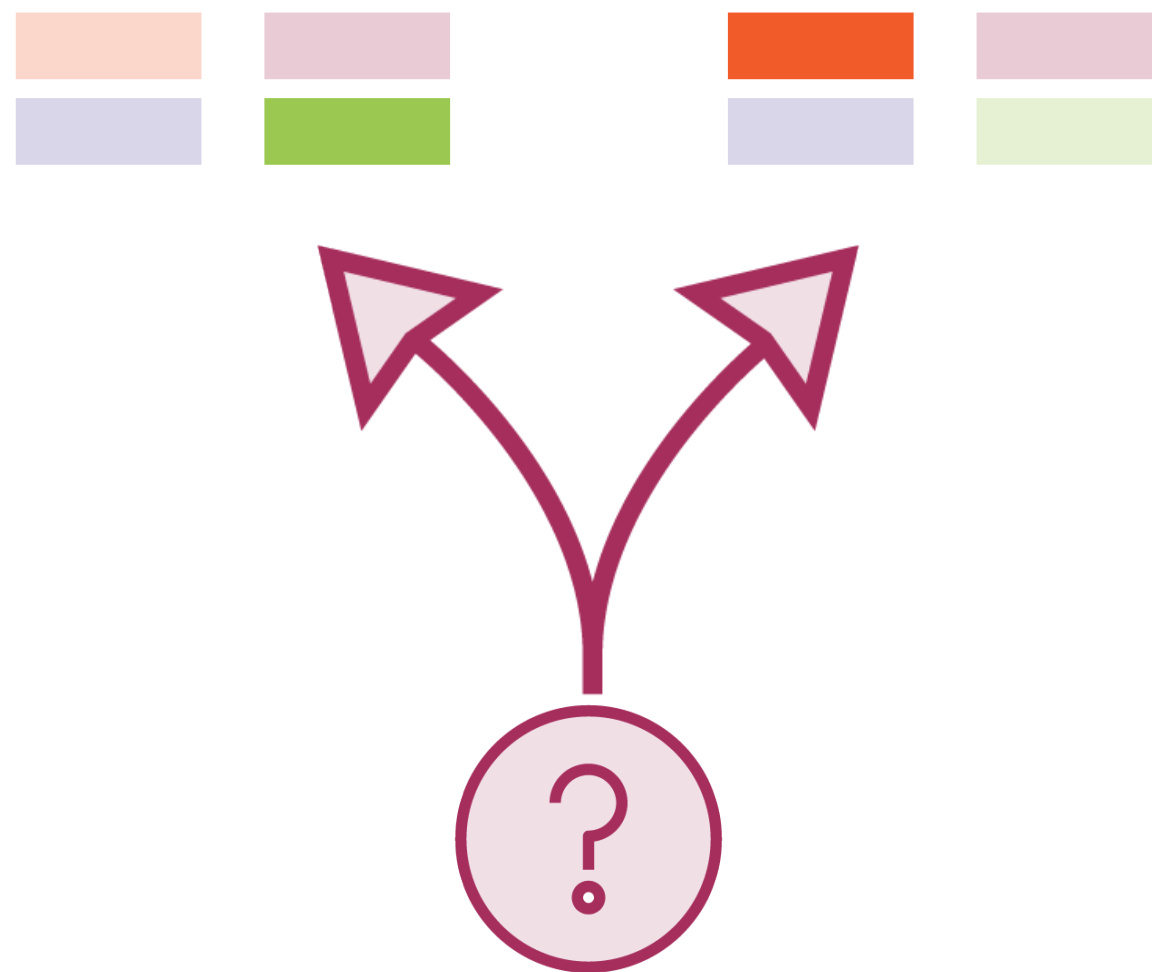













Big Picture Takeaways



Assumes connectivity to the Internet



Assumes user is using iCloud for synchronizing information across their devices



Coupling with UserDefaults is a common strategy for saving data that is essential to your app when offline



Be prepared to answer whether taking latest value for a key from iCloud is the right strategy



Total of 1 MB per app



No notifications in simulator; need iOS devices to fully test

What if you as a developer want to *provide* some
default configuration for your app?

What if you want to ship that configuration as an artifact when you publish to the App Store?

What if you want to save just a little
more information?

Configuring Apps with Property Lists



Settings & Configuration

User Defaults

iCloud Key-Value
Store

Property List
(plist)



Settings & Configuration

User Defaults

iCloud Key-Value
Store

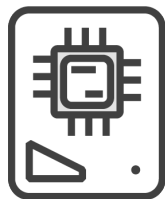
Property List
(plist)

UserDefaults

Memory



Disk

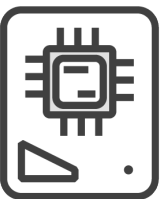


Property List

Memory



Disk



Big Picture Takeaways



File-based approach for persisting simply-structured data sets



Entire contents of a plist get loaded into memory when you access the file



Way to ship initial configuration for your app



Capable of storing lists of data with predictable size

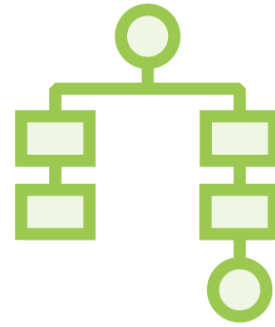


Settings & Configuration

User Defaults

iCloud Key-Value
Store

Property List
(plist)



Objects & Object Graphs

SQLite

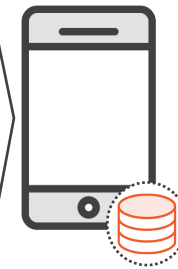
Core Data

Realm Database

CloudKit

Firebase

Realm Platform



Documents & Files

UIDocument

iOS Files App

What if you only wanted to load a subset of the data into memory by querying the data store?

What if you needed to save a **more complex data**
model that includes relationships?

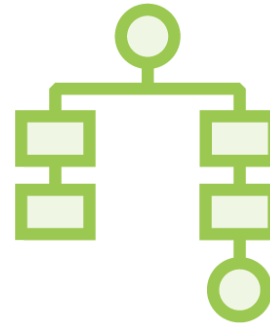


Settings & Configuration

User Defaults

iCloud Key-Value
Store

Property List
(plist)



Objects & Object Graphs

SQLite

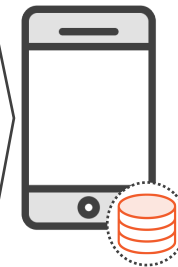
Core Data

Realm Database

CloudKit

Firebase

Realm Platform



Documents & Files

UIDocument

iOS Files App