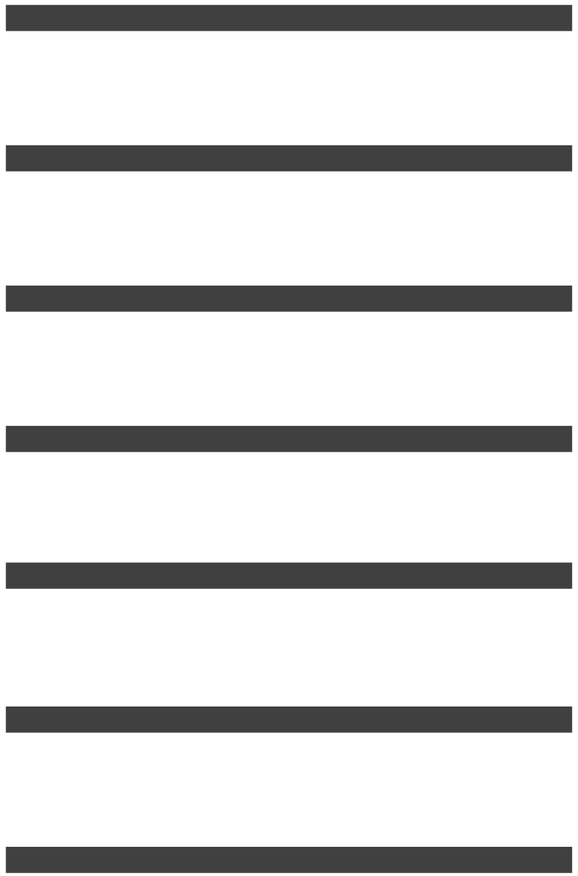
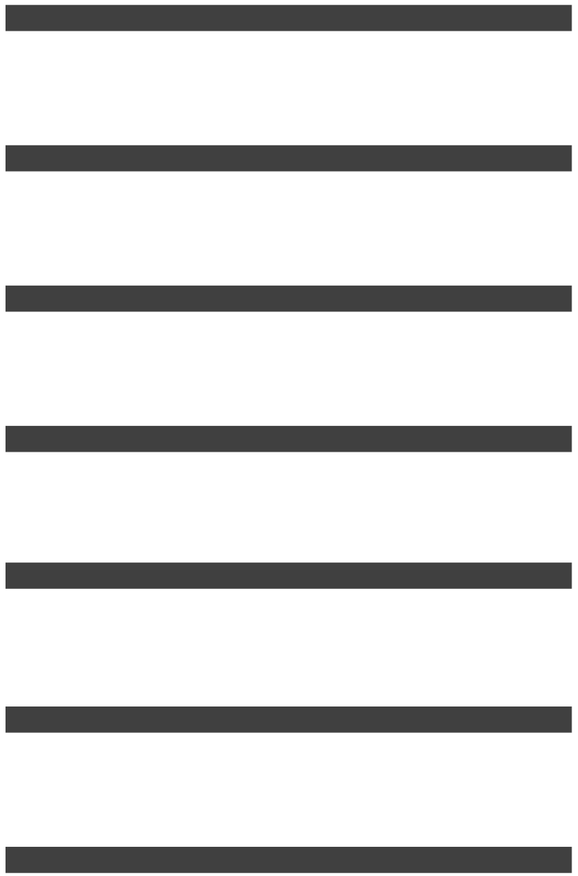
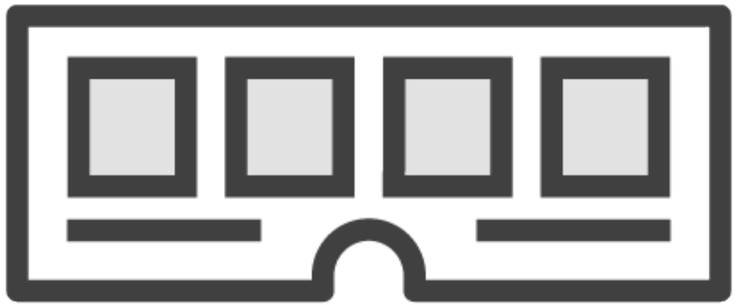
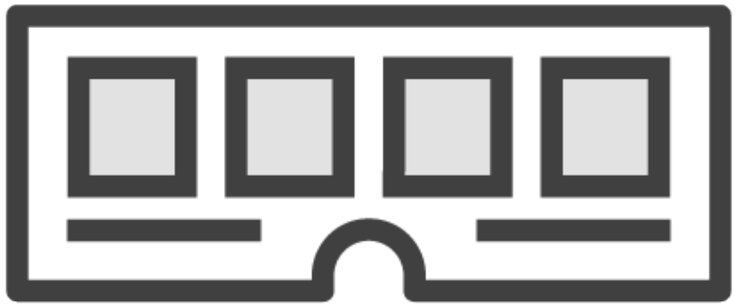
Saving Objects and Object Graphs



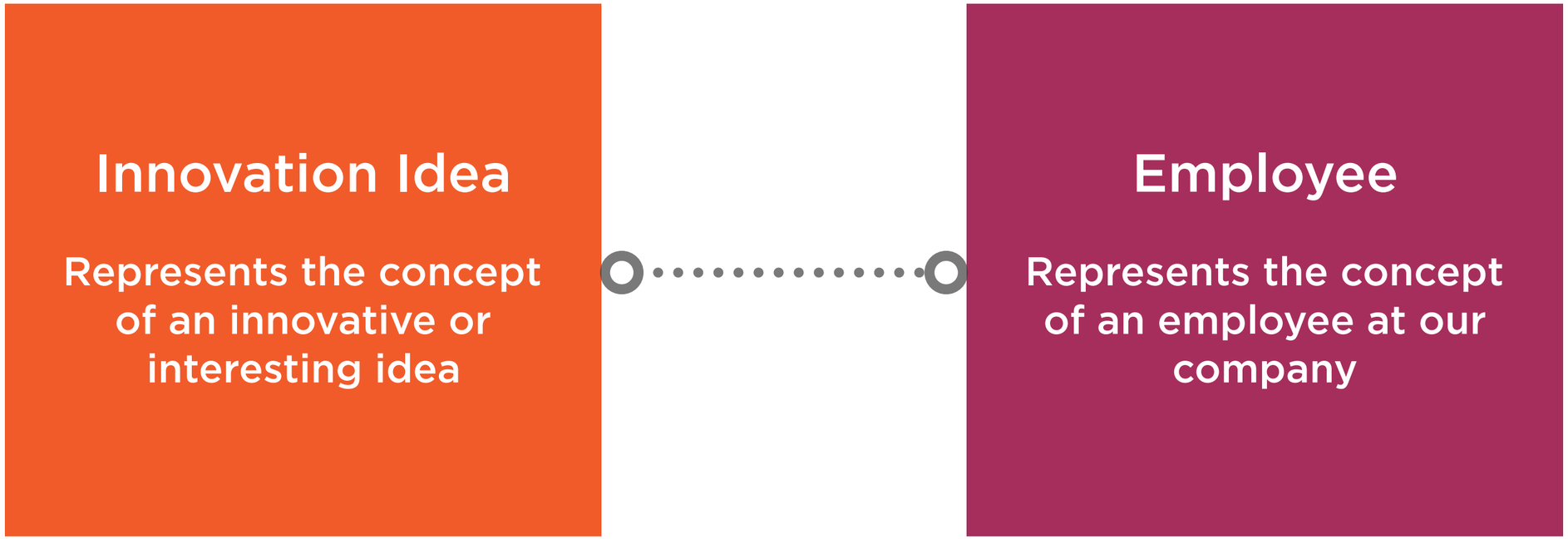
Larger data sets that could grow indefinitely



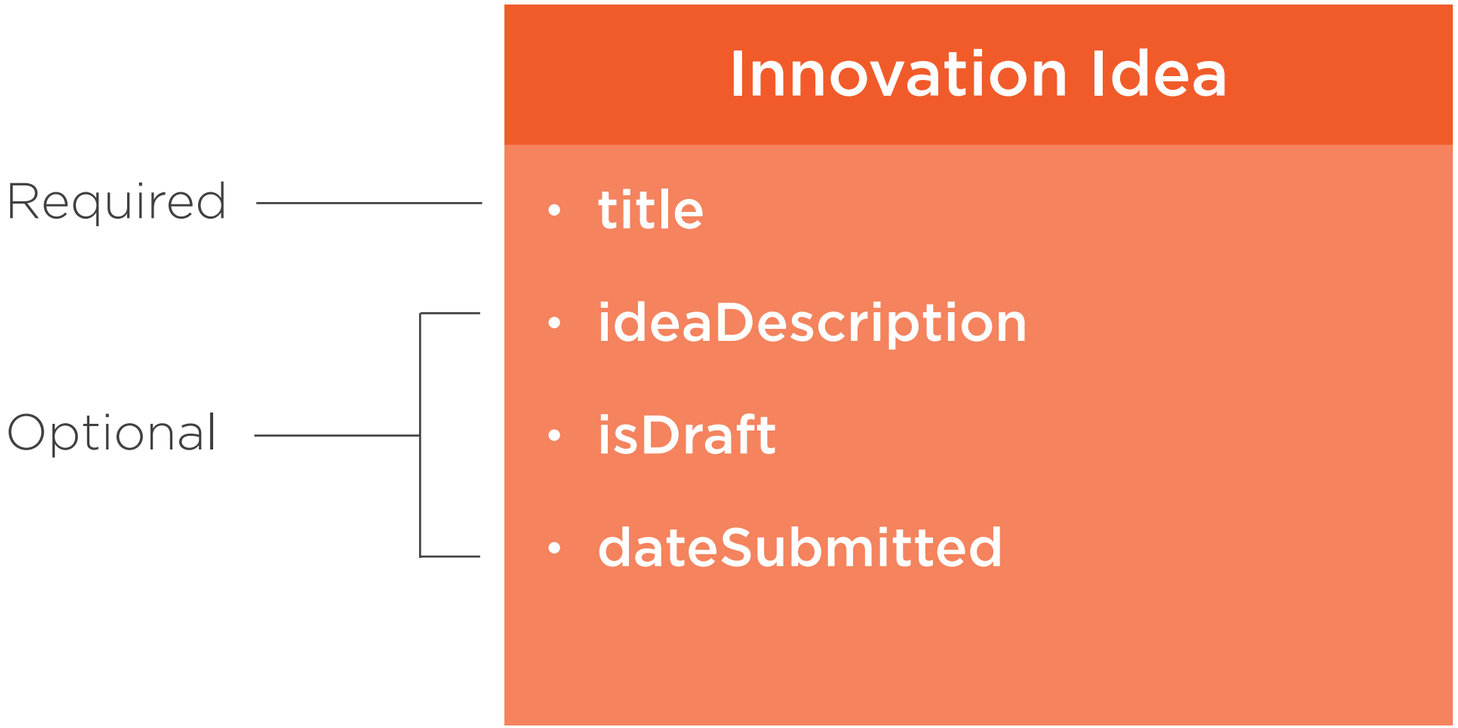




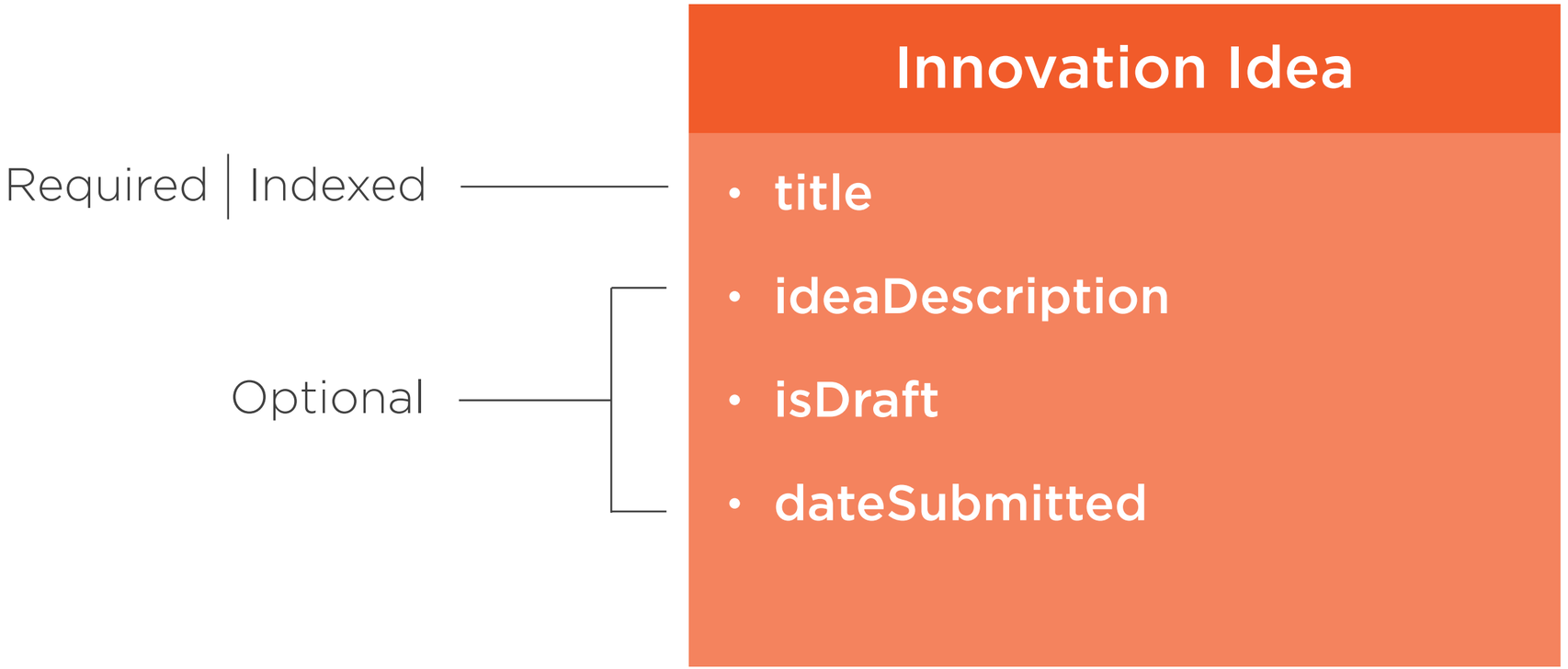
Load subsets of data into memory with querying, filtering, and sorting



Data modeling may involve relationships between objects forming object graphs



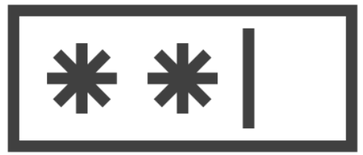
Enforce rules for required fields



Index for performance

Thinking about who leverages data may lead you to requirements about data sharing, which leads to thinking about remote storage, synchronization, and error handling

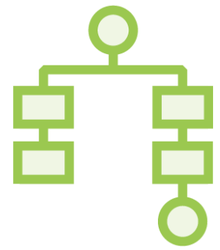




Sharing data will require you to think about users, accounts, authentication, permissions, and privacy

Settings & Configuration

- User Defaults
- iCloud Key-Value Store
- Property List (plist)



Objects & Object Graphs

- SQLite
- Core Data
- Realm Database
- CloudKit
- Firebase
- Realm Platform

Not mean to overwhelm you...

....meant to help set your expectations for the kinds of problems these technologies solve.

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**

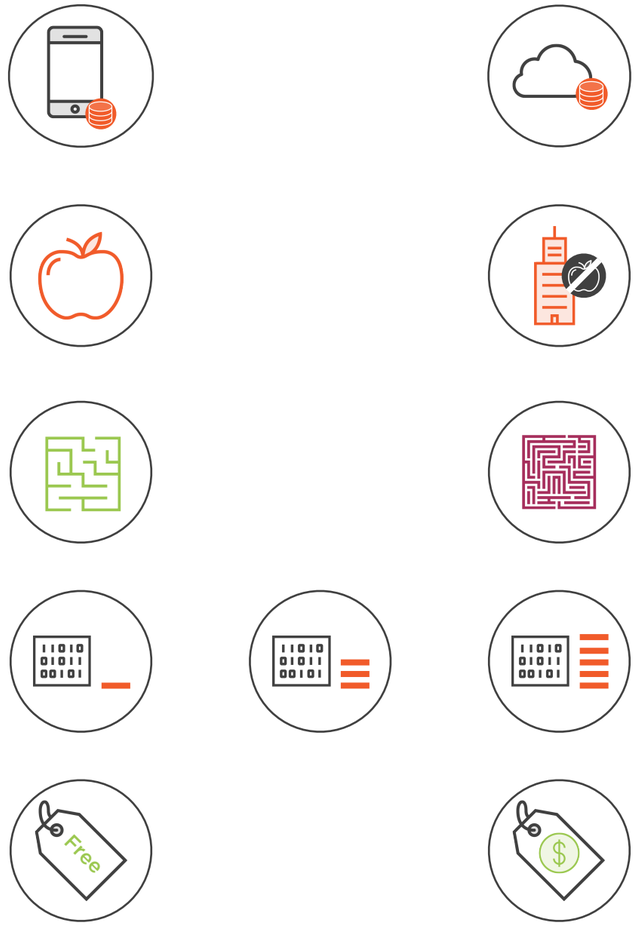
- **Data model complexity**

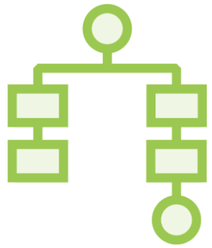
- **Data Size**

- **Price**

Angles for Analysis

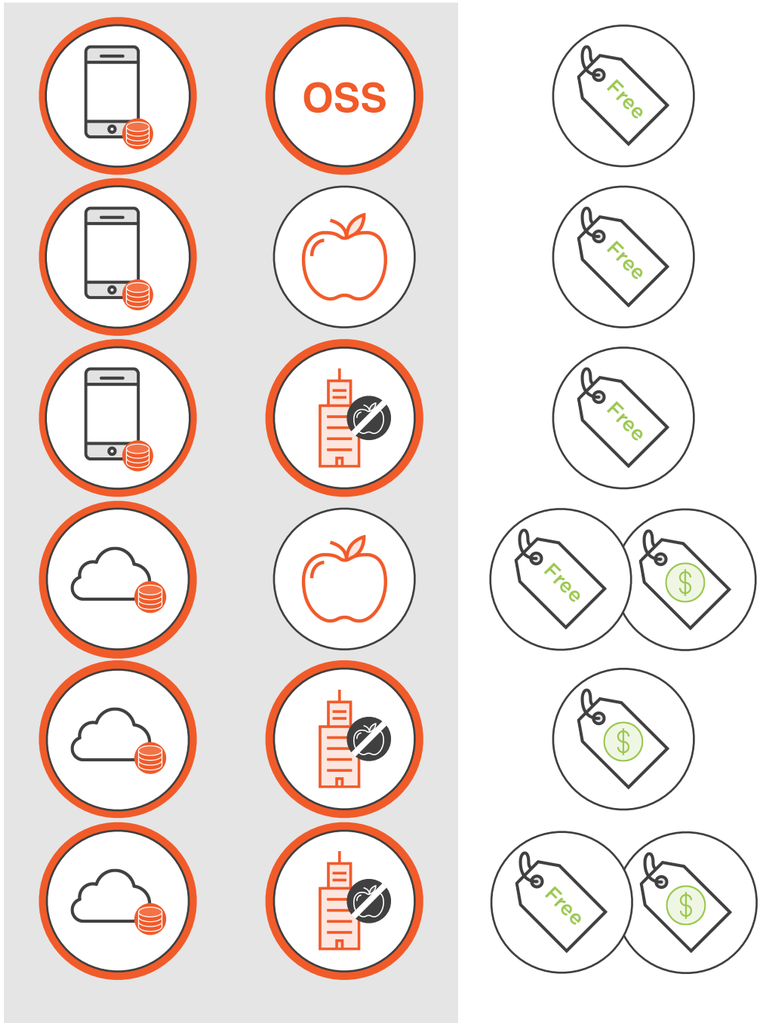
- **Where data is saved**
- **Who maintains the technology**
- **Data model complexity**
- **Data Size**
- **Price**

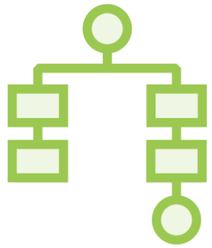




Objects & Object Graphs

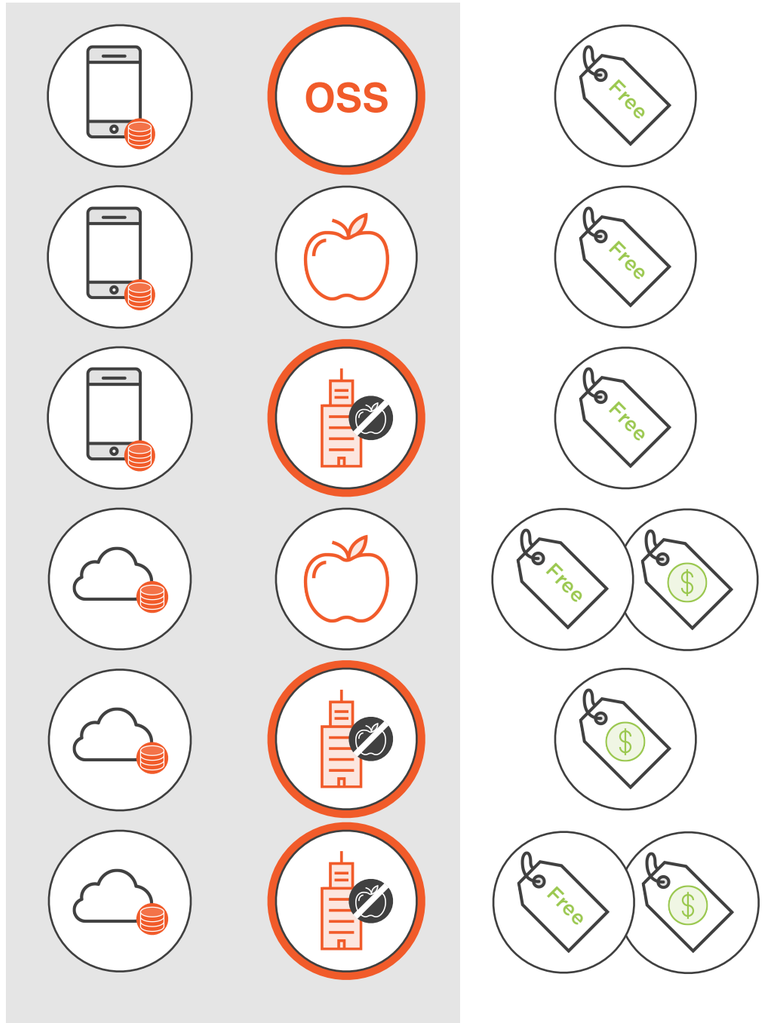
- SQLite
- Core Data
- Realm Database
- CloudKit
- Realm Platform
- Firebase

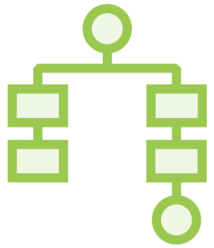




Objects & Object Graphs

- SQLite
- Core Data
- Realm Database
- CloudKit
- Realm Platform
- Firebase





Objects & Object Graphs

SQLite

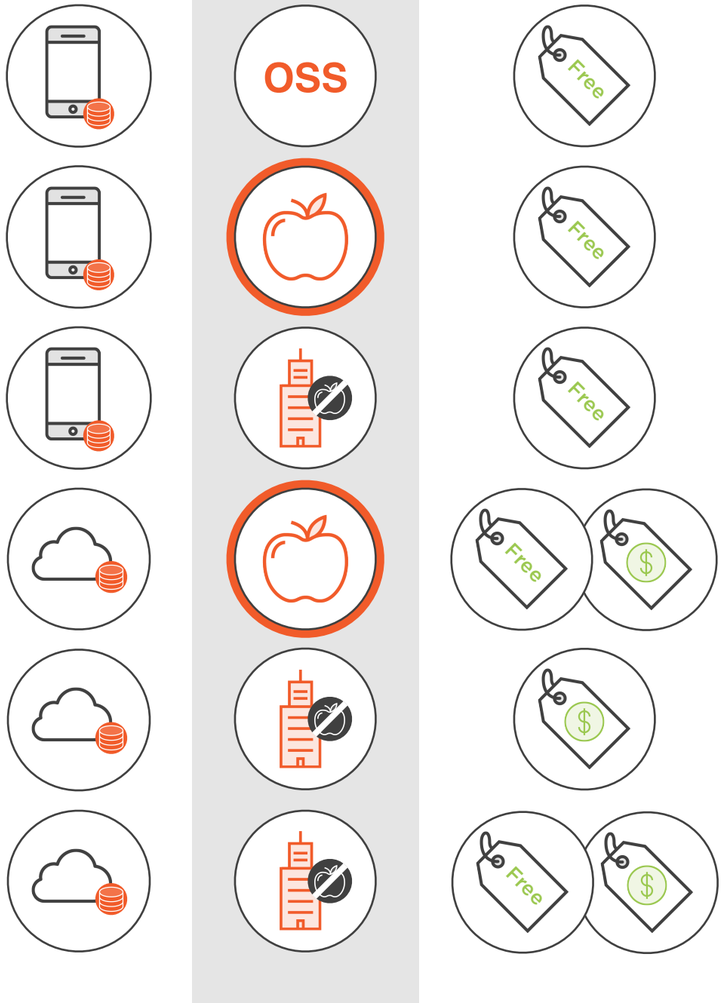
Core Data

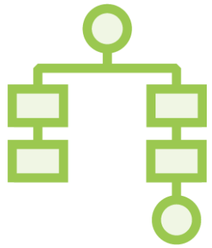
Realm Database

CloudKit

Realm Platform

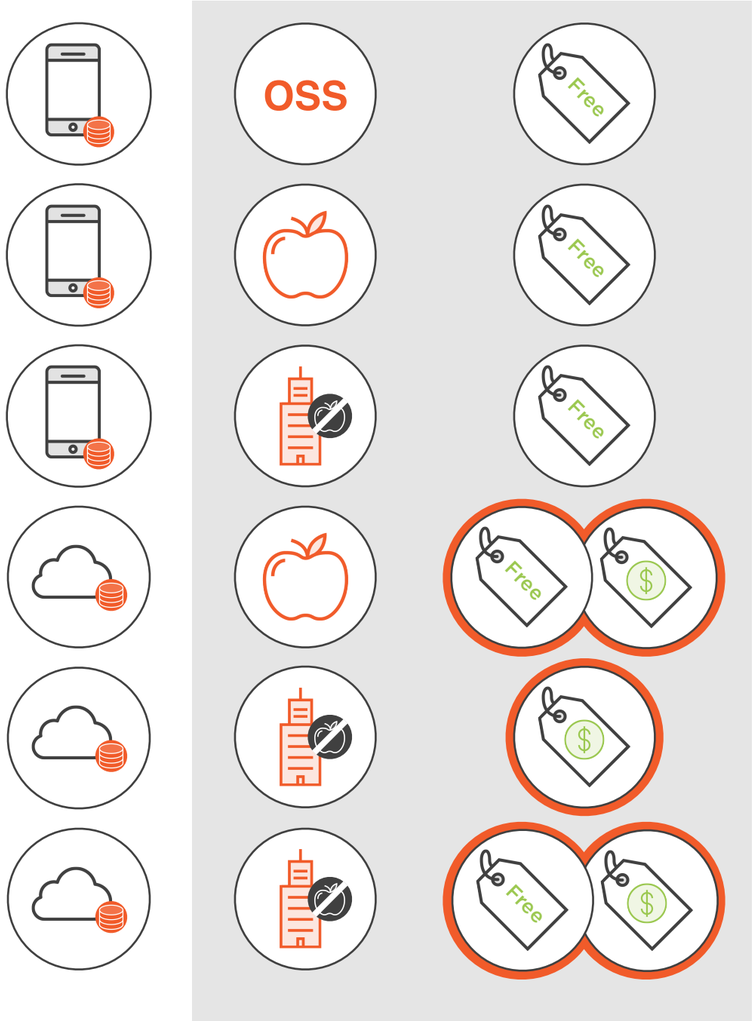
Firebase



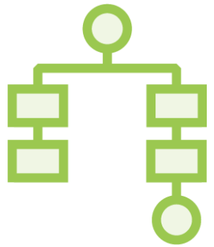


Objects & Object Graphs

- SQLite
- Core Data
- Realm Database
- CloudKit
- Realm Platform
- Firebase

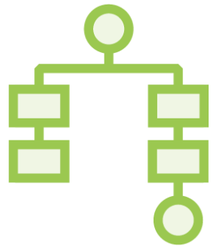


What does it take to get things up and running in an app?



Objects & Object Graphs

- SQLite
- Core Data
- Realm Database
- CloudKit
- Realm Platform
- Firebase



Objects & Object Graphs

SQLite

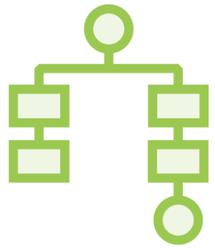
Core Data

Realm Database

CloudKit

Realm Platform

Firebase



Objects & Object Graphs

SQLite

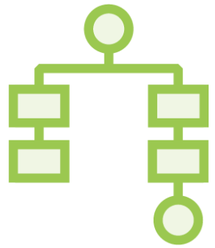
Core Data

Realm Database

CloudKit

Realm Platform

Firebase



Objects & Object Graphs

SQLite

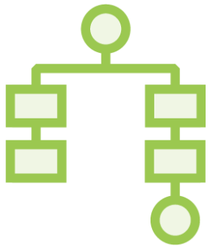
Core Data

Realm Database

CloudKit

Realm Platform

Firebase



Objects & Object Graphs

Set up

Model

Work with data

What am I getting myself into if I pick one persistence technology over another?

How does it feel to work with this persistence technology in code?

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

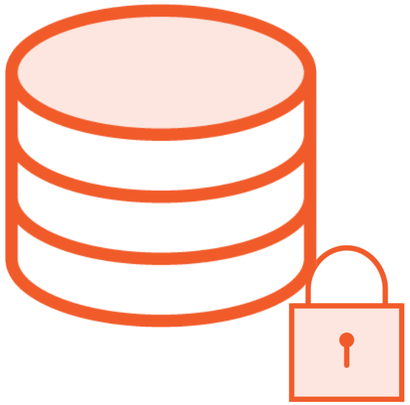
Surveying the Setup Process

CloudKit Architecture

Container

Public Database

Read access for all app users and developer; Write access for creators



Private Database

Read/write access for current user **only**

Shared Database

Read/write access for current user is specified in associated Share object

What am I getting myself into when it comes to setting up each persistence technology?

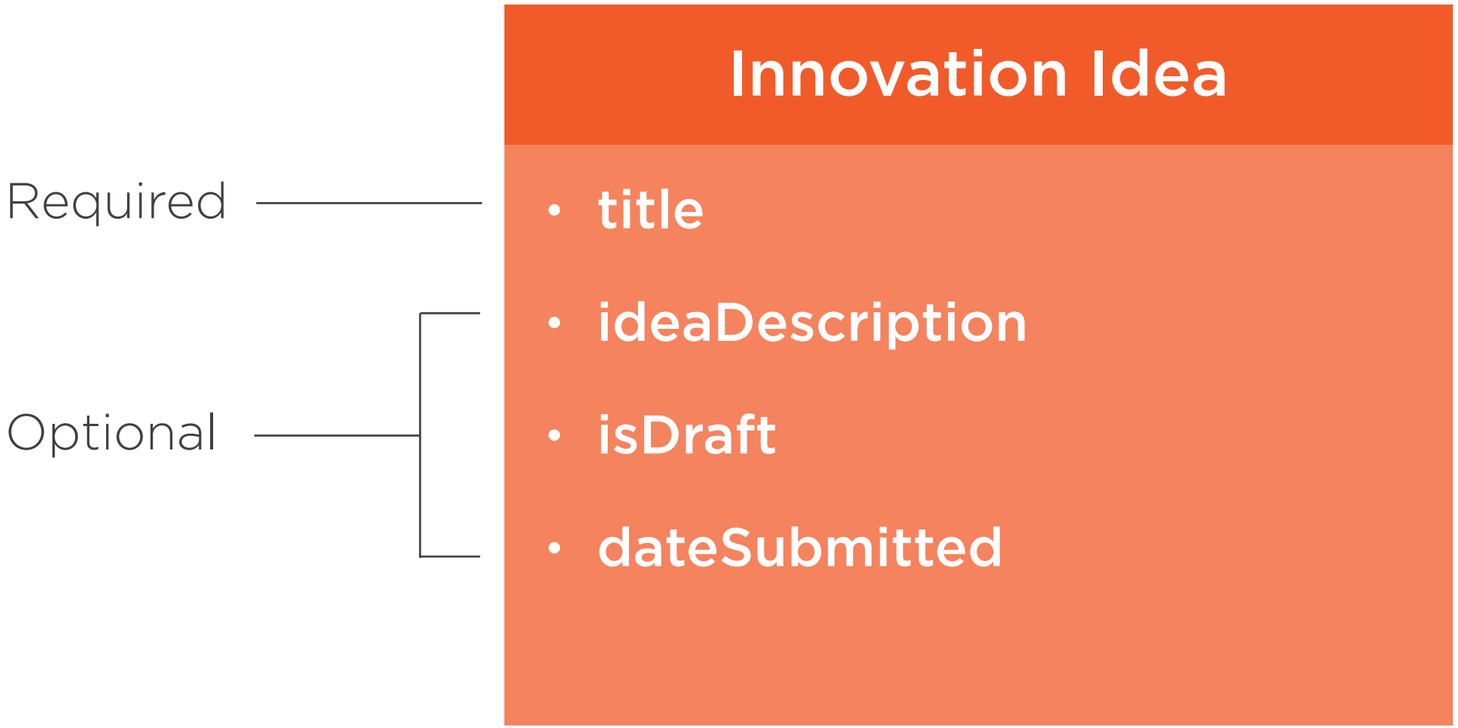
How does the data modeling process feel?

Exploring the Data Modeling Process

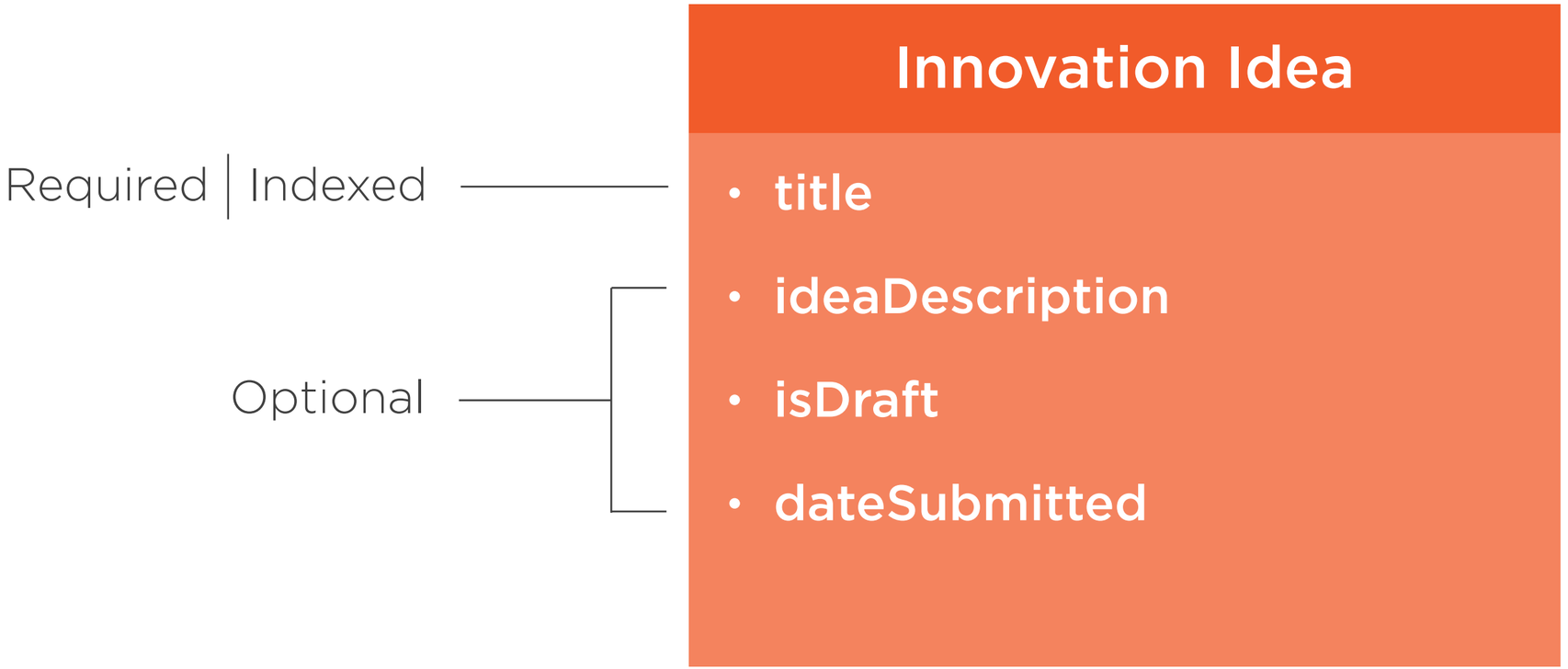
Persistence Layer

```
class InnovationIdea {
      ideaDescription: String
      submittedBy: String
}
```

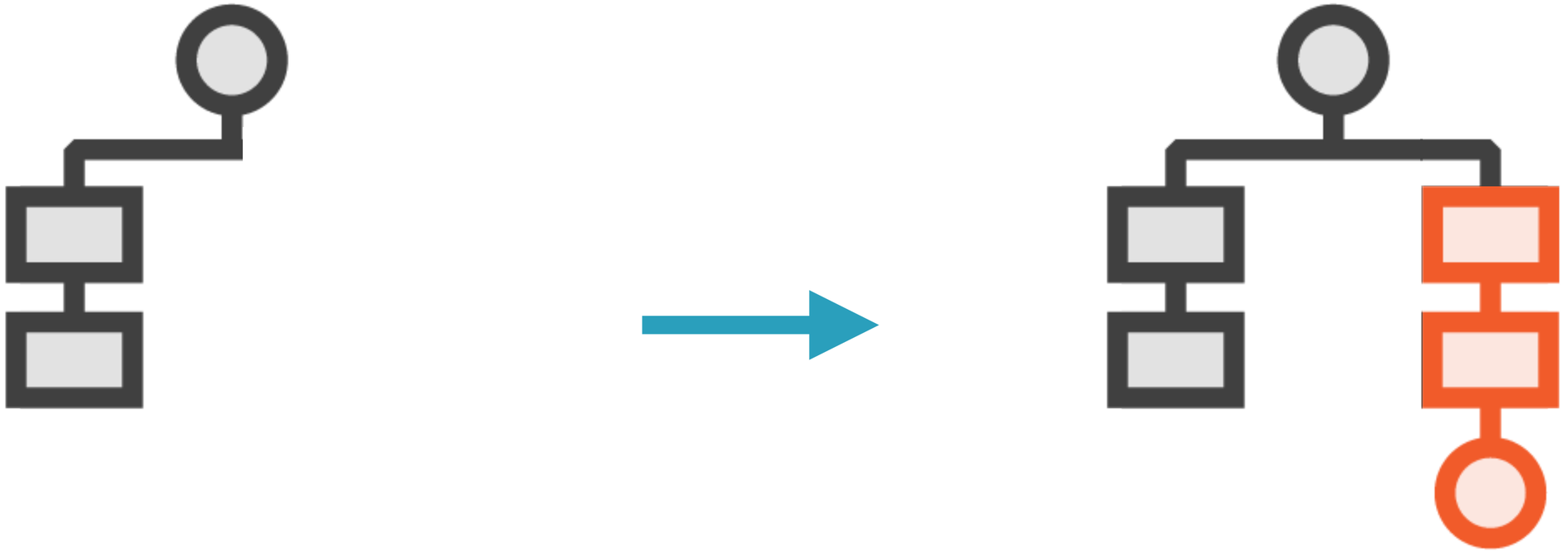
```
struct InnovationIdea {
      ideaDescription: String
      submittedBy: String
}
```



Enforce rules for required fields



Index for performance



Migrate data models to new versions after your app is launched to the App Store

How does it feel to work with data using each of these solutions?

Getting the Feel for Working with Data

Working with Data

- Create and save objects to the persistence layer
- Query with filters and sort orders
- Make updates to objects
- Delete objects
- Watch for data changes to know when and how to update the UI

Keeping the Big Picture in Mind

Keeping the Big Picture in Mind

- Always a bit of setup
- Always need to connect to your persistence layer
- Always need to define a structure for your data
- Always need to create, query, filter, sort, update, and delete data
- Always need to make sure your UI stays in sync with all of these different kinds of changes

What if you need to save data in a more "unstructured" way?

What if you need to encapsulate chunks of content together and save it all as a packaged up file...

...as a single unit with a custom file extension?

Documents & Files

UIDocument

iOS Files App