

# Persistence: Core Data

Stop trying to make Core Data a database. It's not a database.

by Fernando Olivares






# Agenda

## 1.5-hour session

- Author Introduction
- Live coding
- Core Data Theory
- Conclusion & Where to go from here
- Q&A

# Author Introduction

## Fernando

- ~10 years of experience
- Worked at small startups (  1SecondEveryday) to publicly traded companies (  J2 Global Inc.)
- Instructor at  Big Nerd Ranch,  [bloc.io](https://bloc.io),  Lambda School
- Won a few awards: The Storyteller Within (Apple), ERA Accelerator Top 10 (ERA NY)
- Product and Project experience
- iOS-only
- @fromJrToSr

# Core Data

## What exactly is Core Data?

- Object-graph persistence framework
- Core Data "saves your application's permanent data for offline use, to cache temporary data, and to add undo functionality to your app on a single device."
- Extremely powerful and efficient: complex queries, batch sizing, memory faulting, background execution, relationships between models.
- Based in Objective-C.

# Core Data

Radio is our music source.



Radio

FM 101.3  
FM 99.3



Songs

"Everybody wants to  
rule the world"

"Everybody"

"Gonna make you  
sweat  
(Everybody dance  
now)"



Person who likes songs

Listens for great songs  
to cover

# Core Data

No radio means no music.



Radio



Person who likes songs

Listens for great songs  
to cover

# Core Data

We'd really like to listen to those songs again.



Radio

FM 101.3  
FM 99.3



Songs

"Everybody wants to  
rule the world"

"Everybody"

"Gonna make you  
sweat  
(Everybody dance  
now)"



Person who likes songs

Listens for great songs  
to cover



Artist

Can sing and perform as  
well as write music in  
musical notation

# Core Data

## How to record a great song.



Radio



Person who likes songs



Artist

Playing "Everybody" by the Backstreet Boys



Musical Notation  
Reference Guide

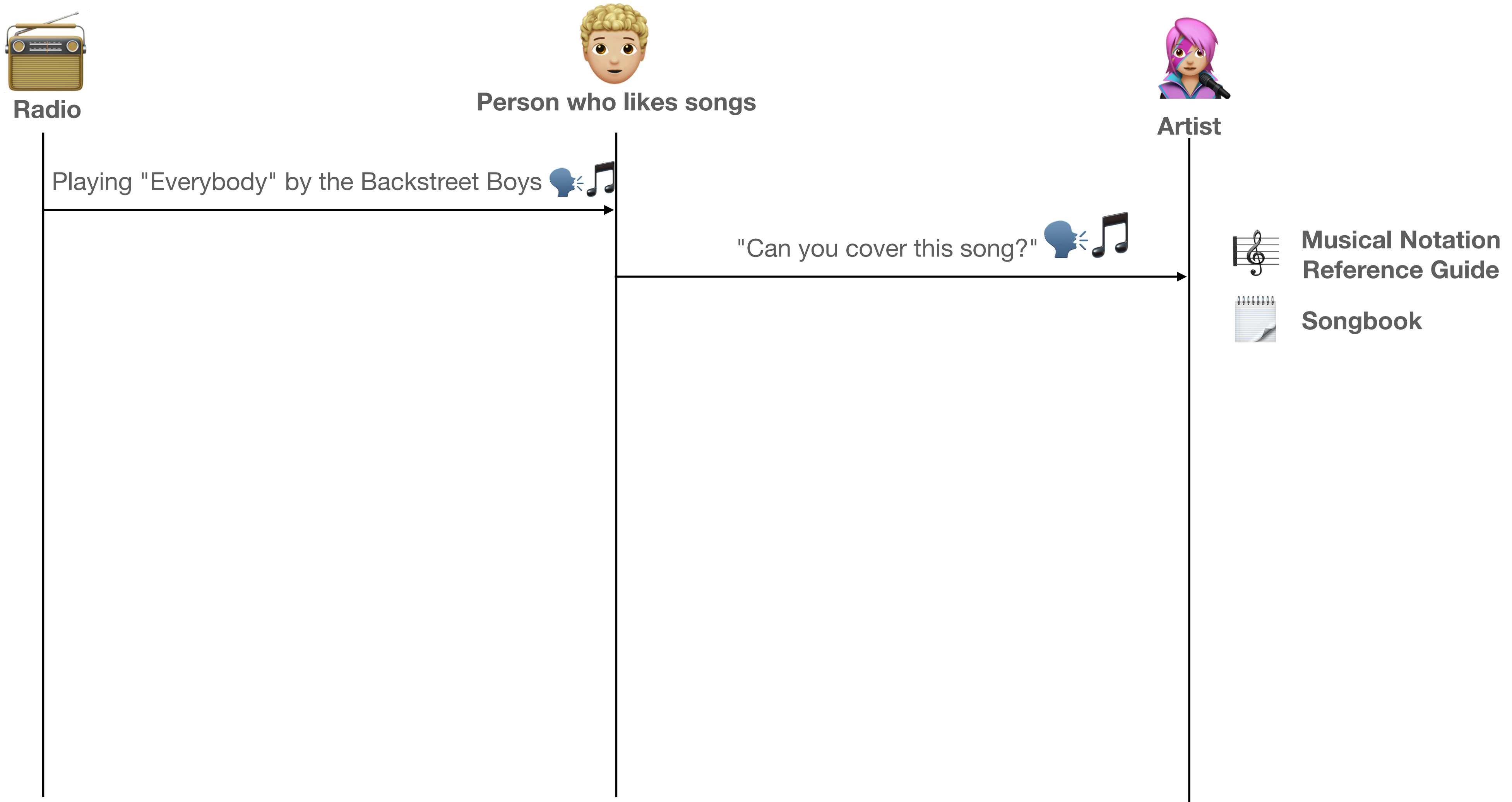


Songbook



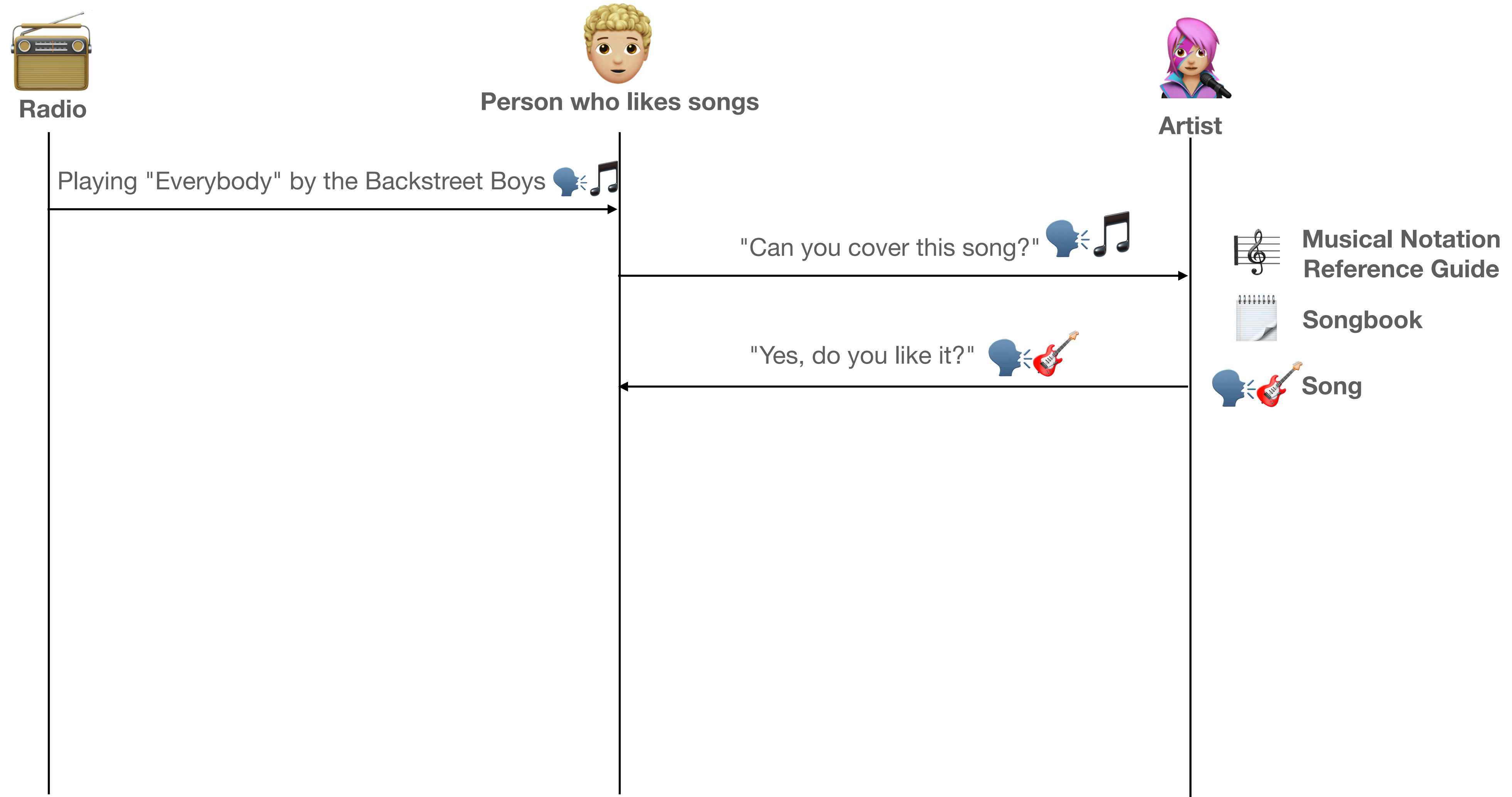
# Core Data

## How to record a great song.



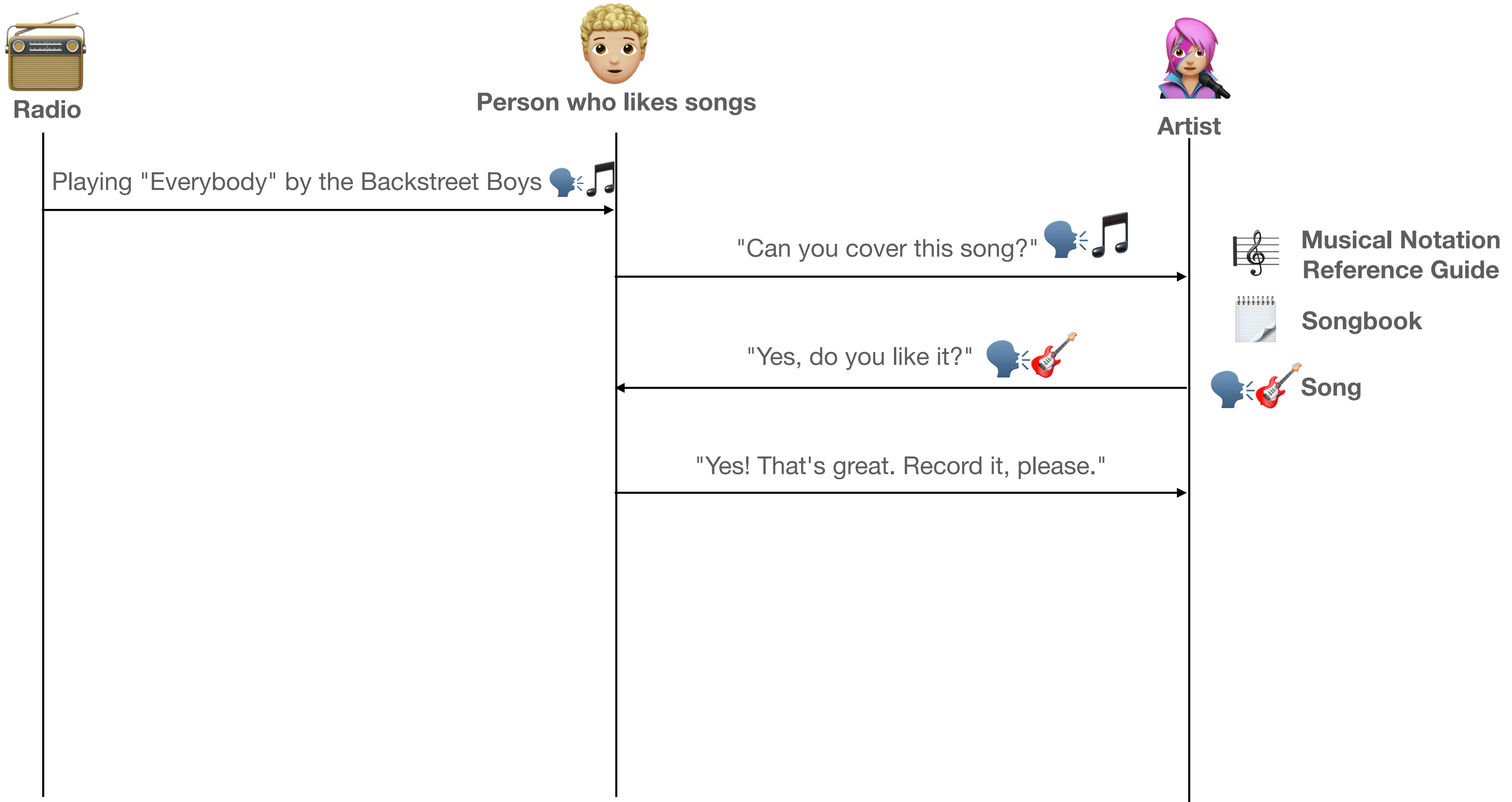
# Core Data

## How to record a great song.



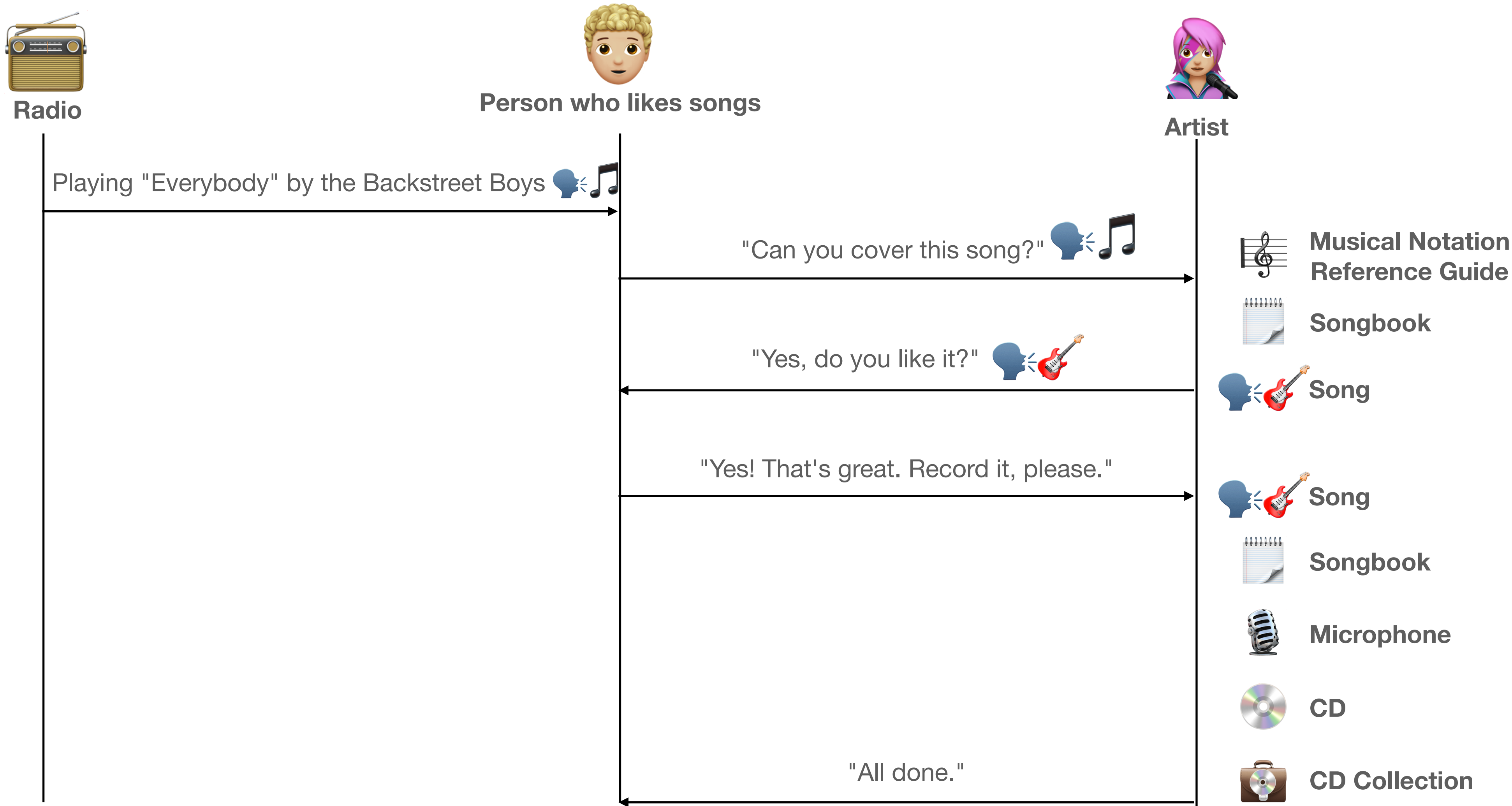
# Core Data

## How to record a great song.



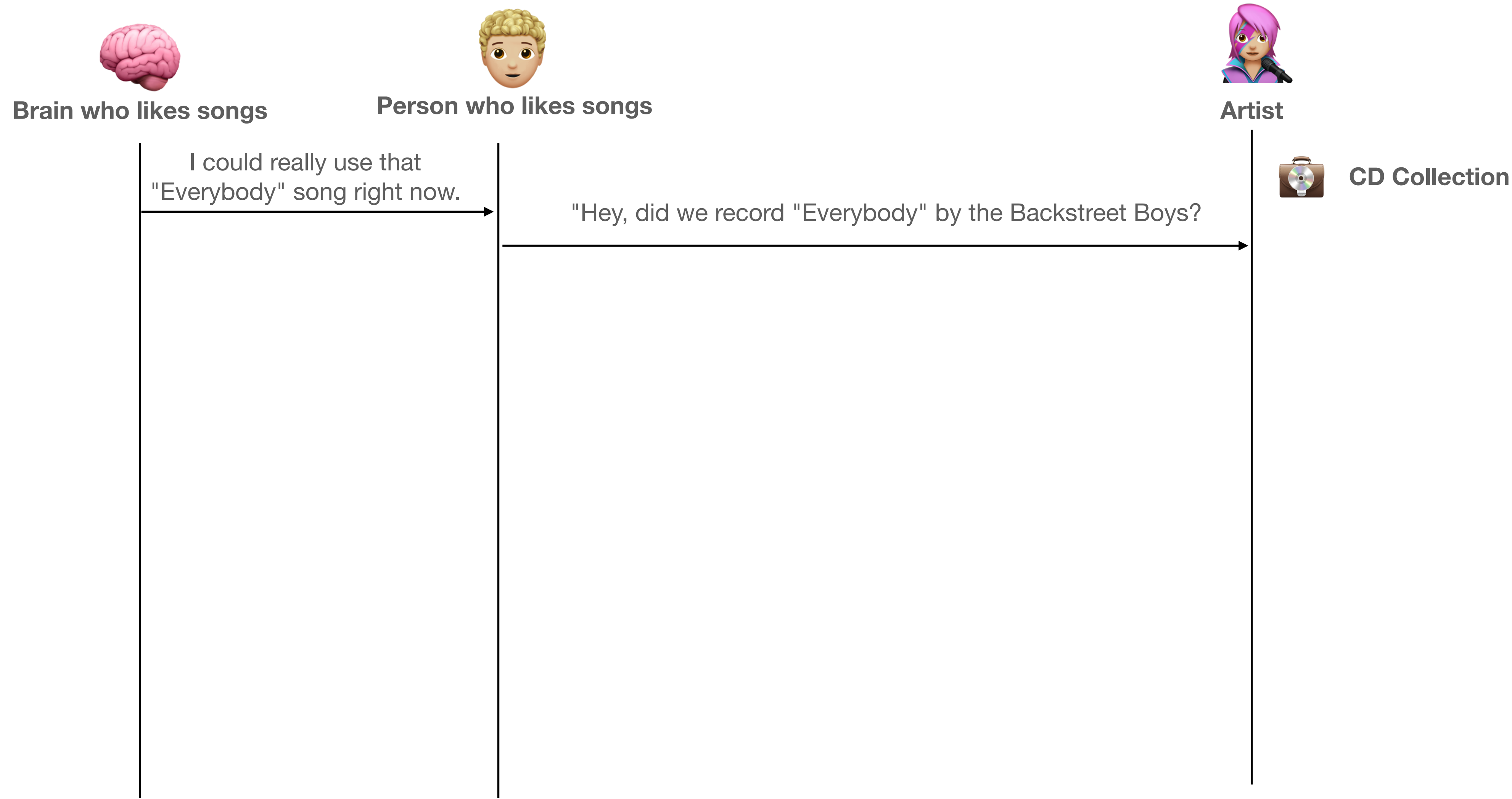
# Core Data

## How to record a great song.



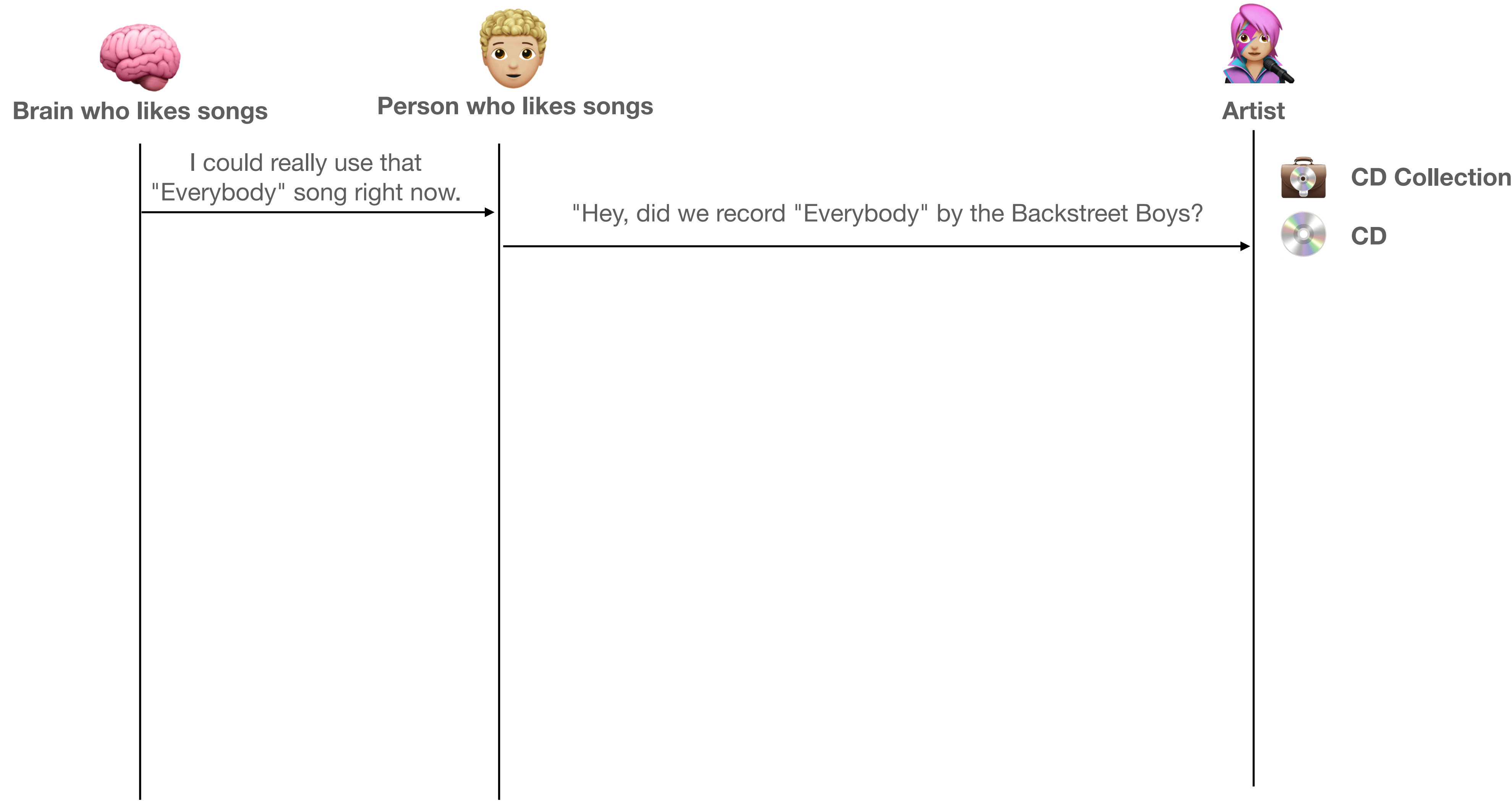
# Core Data

I want to listen to that song.



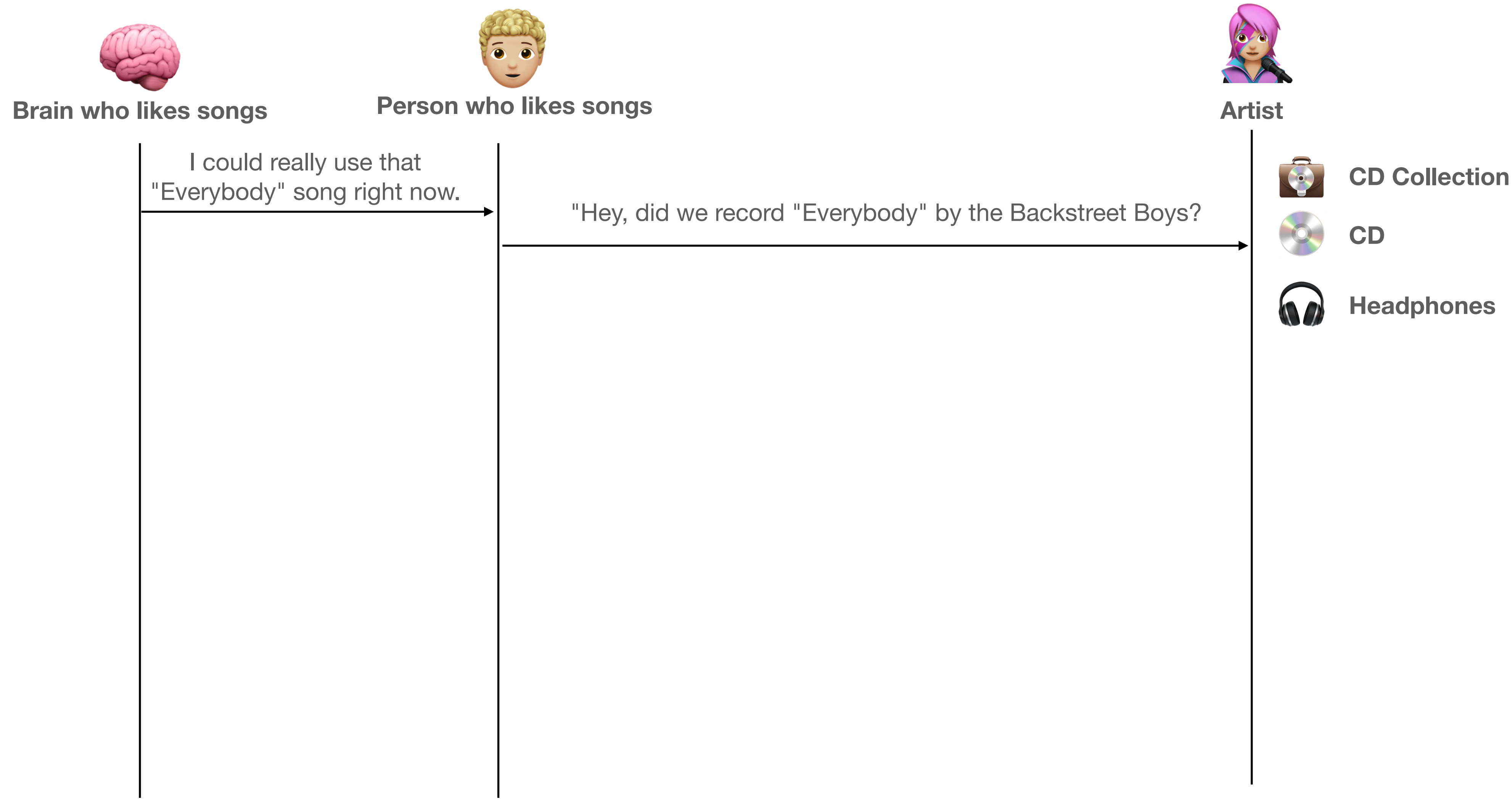
# Core Data

I want to listen to that song.



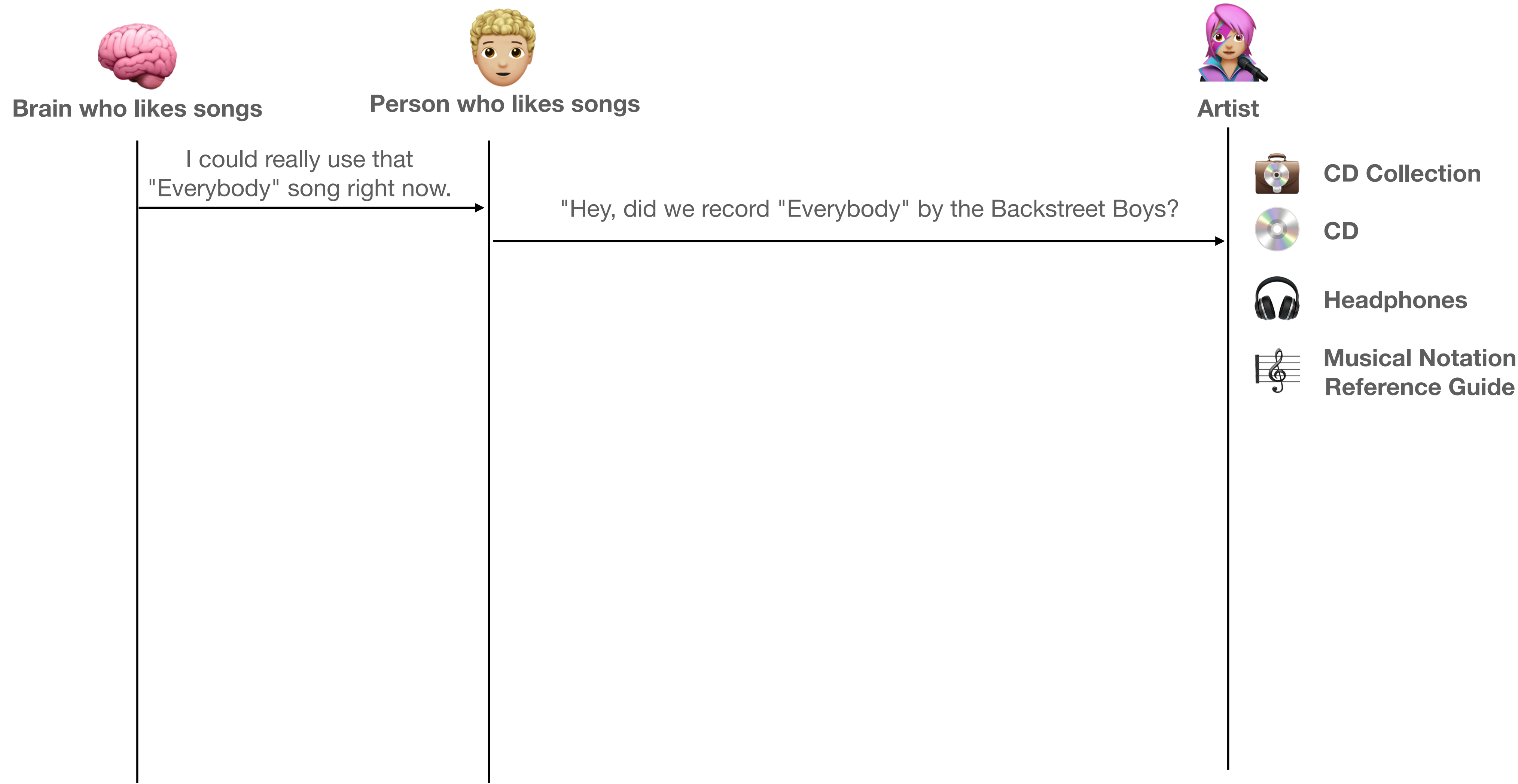
# Core Data

I want to listen to that song.



# Core Data

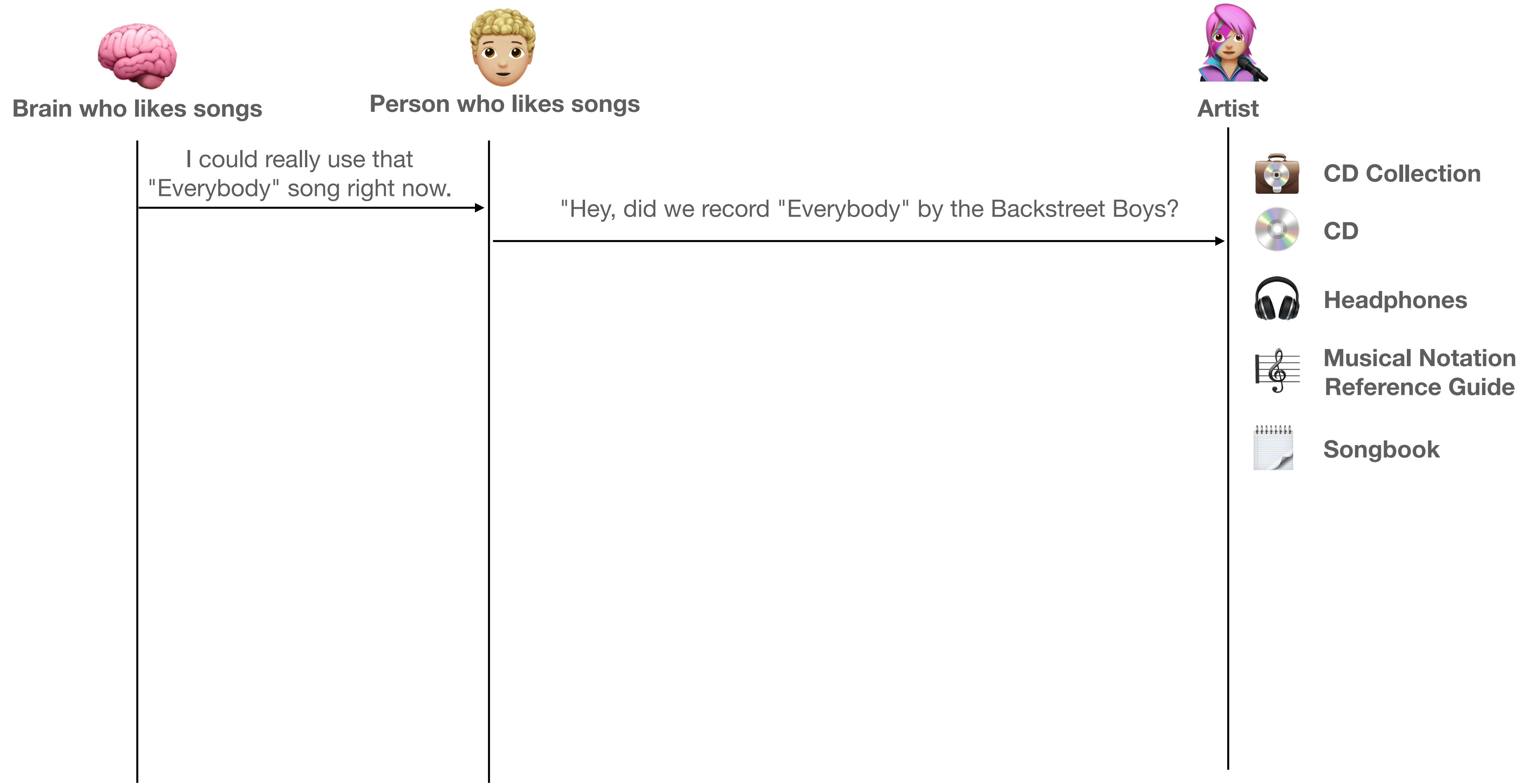
I want to listen to that song.





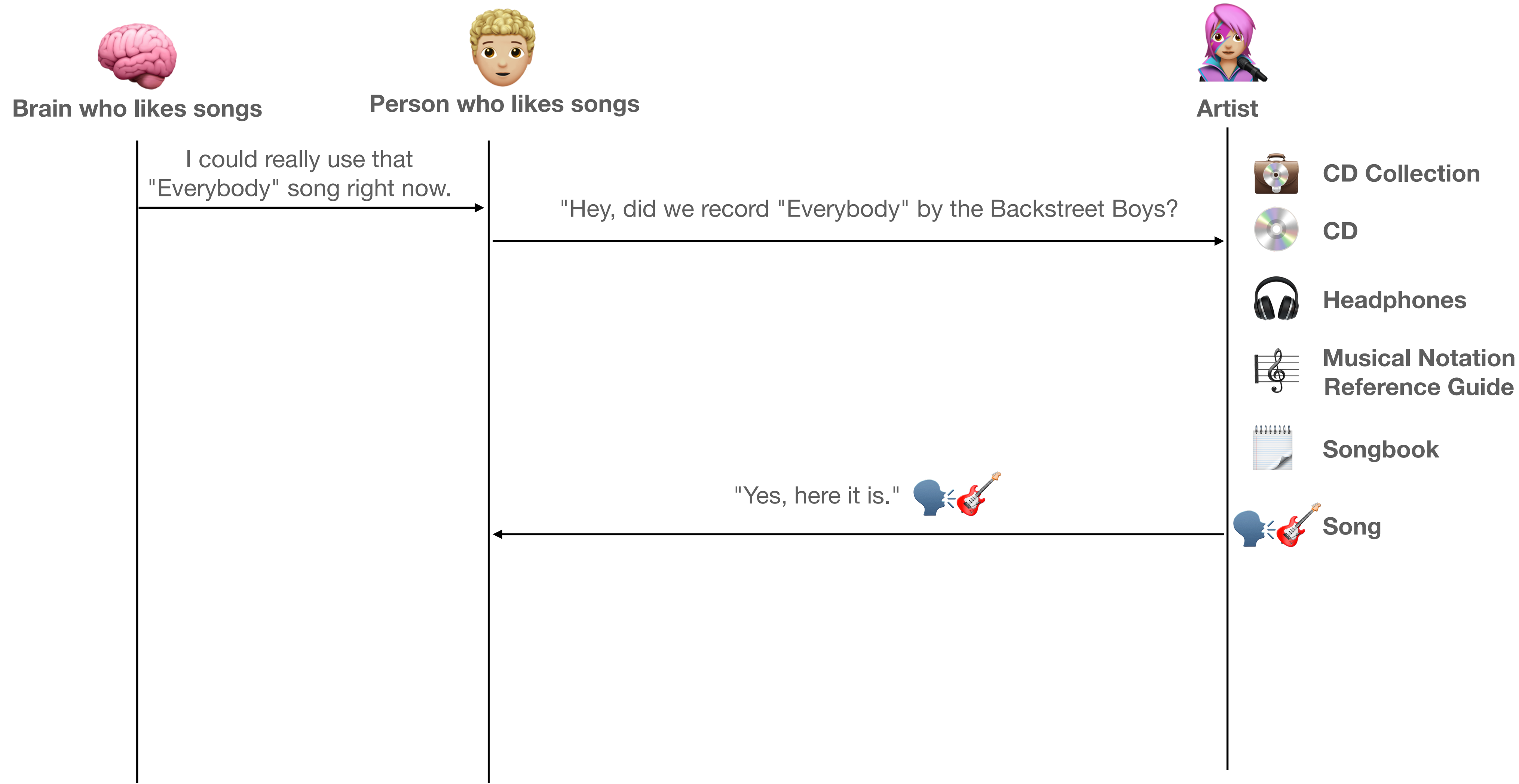
# Core Data

I want to listen to that song.



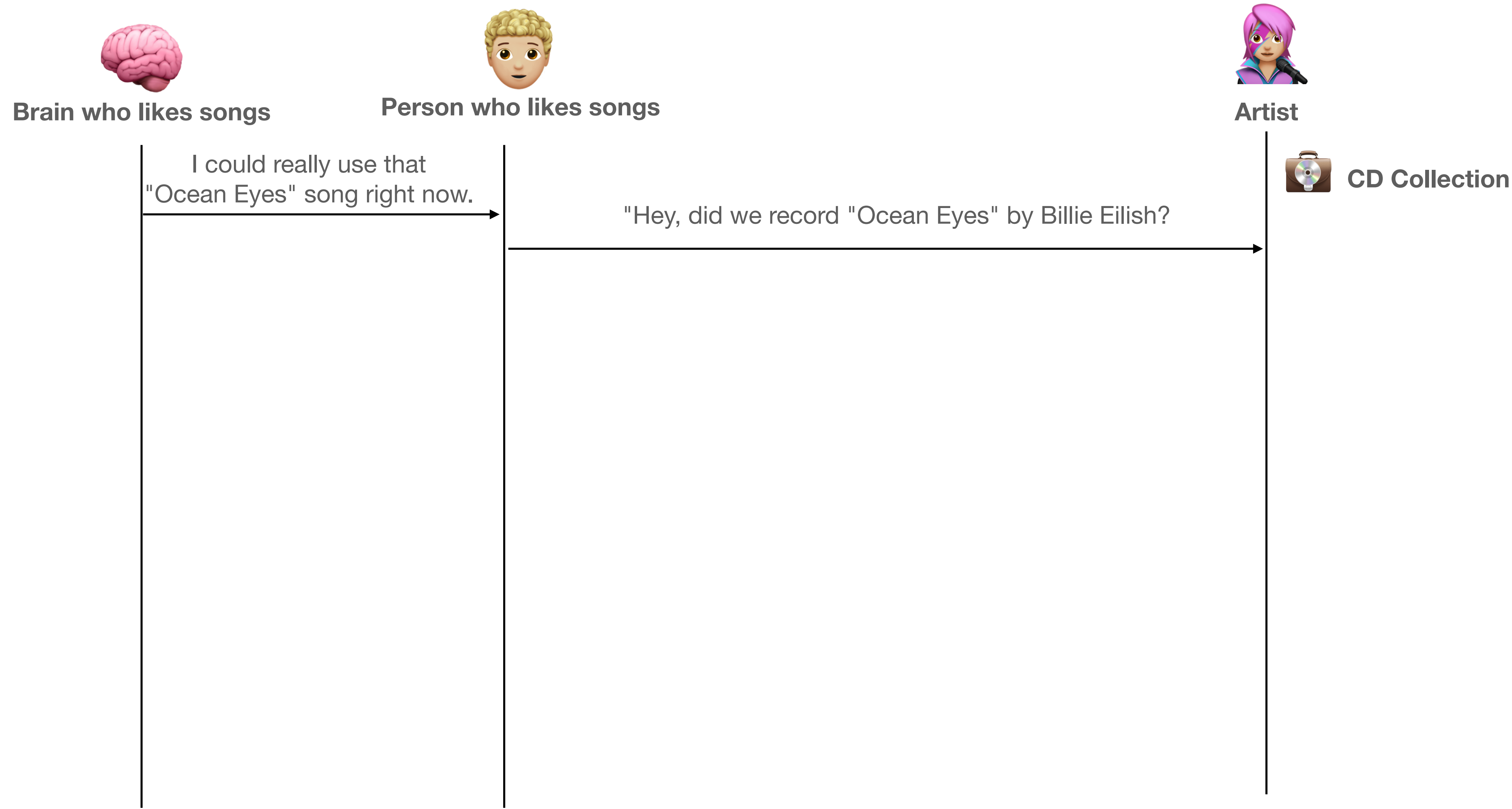
# Core Data

I want to listen to that song.



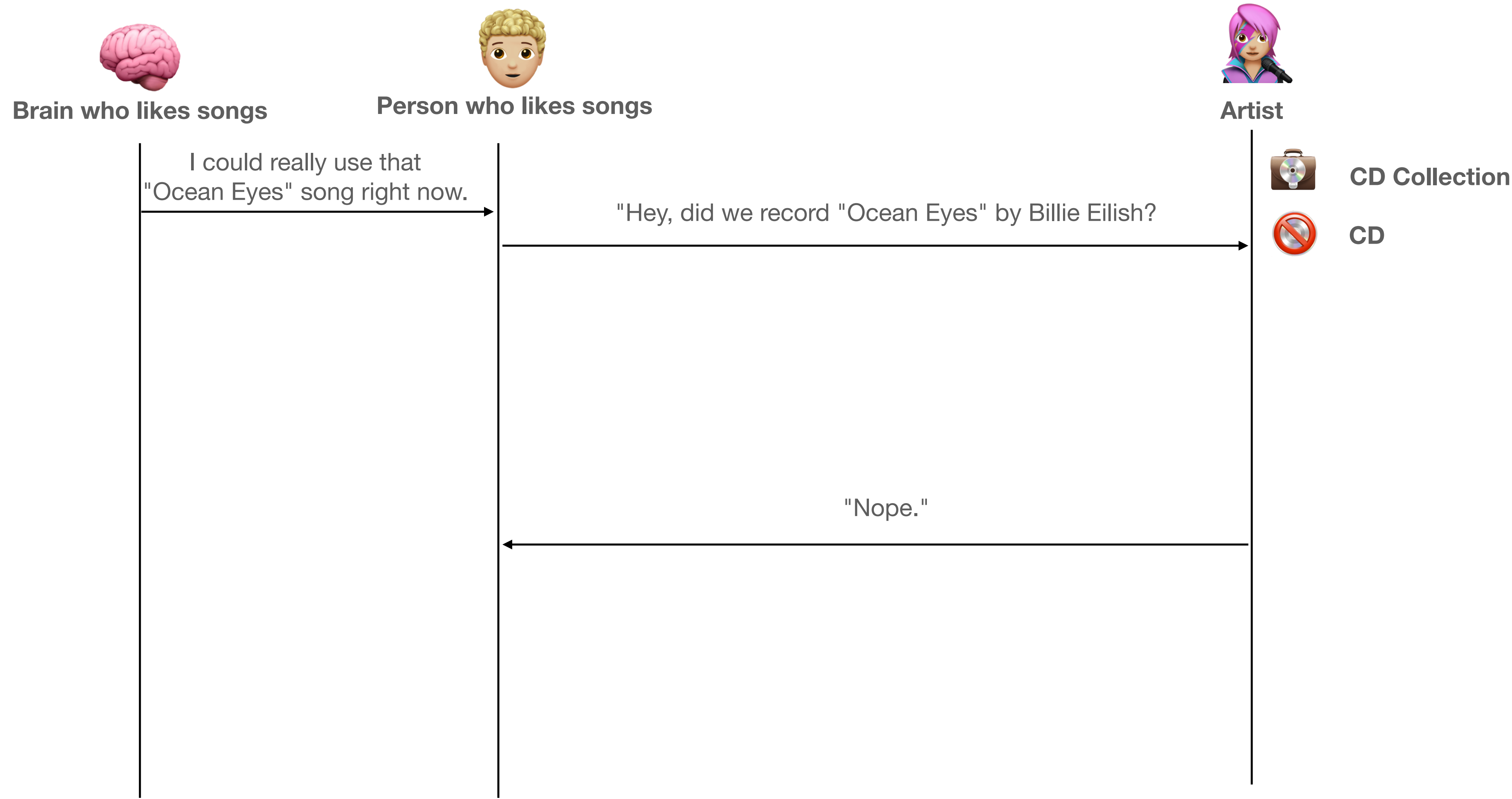
# Core Data

I want to listen to that song.



# Core Data

I want to listen to that song.



# Core Data

In practice, it's simple.



API

iTunes API



Podcasts JSON

Top Podcasts JSON



ViewController

Displays struct Podcasts



NSPersistenceContainer

Can read/write  
CoreDataPodcasts



Database/Plist/Binary File



Row/XML entry/Binary



NSPersistentStoreCoordinator



CoreDataPodcast  
NSManagedObject



NSManagedObjectContext



xcdatamodeld

# Live Demo

NSPersistenceContainer

# Core Data Stack

## NSPersistenceContainer's tools.



API

iTunes API



Podcasts JSON

Top Podcasts JSON



ViewController

Displays struct Podcasts



NSPersistenceContainer

Can read/write  
CoreDataPodcasts



**CoreDataPodcast**  
**NSManagedObject**

Individual objects



**NSManagedObjectContext**

Contains the individual  
objects that represent the  
graph



**xcdatamodeld**

Contains the  
object graph  
description



# Core Data Stack

Instructions on building an object graph.



API

iTunes API



Podcasts JSON

Top Podcasts JSON



ViewController

Displays struct Podcasts



NSPersistenceContainer

Can read/write  
CoreDataPodcasts



**CoreDataPodcast**  
**NSManagedObject**

Individual objects



**NSManagedObjectContext**

Contains the individual  
objects that represent the  
graph



**xcdatamodeld**

Contains the  
object graph  
description



# Core Data Stack

The object graph.



API

iTunes API



Podcasts JSON

Top Podcasts JSON



ViewController

Displays struct Podcasts



NSPersistenceContainer

Can read/write  
CoreDataPodcasts



**CoreDataPodcast**  
**NSManagedObject**

Individual objects



**xcdatamodeld**

Contains the  
object graph  
description



**NSManagedObjectContext**

Contains the individual  
objects that represent the  
graph

# Core Data Stack

The individual objects, inheriting from `NSManagedObject`.



API

iTunes API



Podcasts JSON

Top Podcasts JSON



ViewController

Displays struct Podcasts



NSPersistenceContainer

Can read/write  
CoreDataPodcasts



**CoreDataPodcast**  
**NSManagedObject**  
Individual objects



**NSManagedObjectContext**

Contains the individual  
objects that represent the  
graph



**xcdatamodeld**

Contains the  
object graph  
description

# Core Data Stack

There are other tools...



API

iTunes API



Podcasts JSON

Top Podcasts JSON



ViewController

Displays struct Podcasts



NSPersistenceContainer

Can read/write  
CoreDataPodcasts



xcdatamodeld

Contains the  
object graph  
description



NSPersistentStore



NSPersistentStoreCoordinator



CoreDataPodcast  
NSManagedObject

Individual objects



NSManagedObjectContext

Contains the individual  
objects that represent the  
graph

# Core Data Stack

## How to go from context to disk.



API

iTunes API



Podcasts JSON

Top Podcasts JSON



ViewController

Displays struct Podcasts



NSPersistenceContainer

Can read/write  
CoreDataPodcasts



xcdatamodeld

Contains the  
object graph  
description



NSPersistentStore



NSPersistentStoreCoordinator

Transforms from  
NSManagedObjects to  
whatever NSPersistentStore  
handles (and viceversa)



CoreDataPodcast  
NSManagedObject

Individual objects



NSManagedObjectContext

Contains the individual  
objects that represent the  
graph



# Core Data Stack

The pure, persisted data.



API

iTunes API



Podcasts JSON

Top Podcasts JSON



ViewController

Displays struct Podcasts



NSPersistenceContainer

Can read/write  
CoreDataPodcasts



xcdatamodeld

Contains the  
object graph  
description



**NSPersistentStore**

Contains the data  
at rest



**NSPersistentStoreCoordinator**

Transforms from  
NSManagedObjects to  
whatever NSPersistentStore  
handles (and viceversa)



**CoreDataPodcast  
NSManagedObject**

Individual objects



**NSManagedObjectContext**

Contains the individual  
objects that represent the  
graph

# Core Data Stack

With our powers combined...



API

iTunes API



Podcasts JSON

Top Podcasts JSON



ViewController

Displays struct Podcasts



**NSPersistentStore**

Contains the data  
at rest



**NSPersistentStoreCoordinator**

Transforms from  
NSManagedObjects to  
whatever NSPersistentStore  
handles (and viceversa)



NSPersistenceContainer

Can read/write  
CoreDataPodcasts



**CoreDataPodcast  
NSManagedObject**

Individual objects



**NSManagedObjectContext**

Contains the individual  
objects that represent the  
graph



**xcdatamodeld**

Contains the  
object graph  
description

# Core Data

**It's not a database.**

- 🎵 How to build CoreDataPodcast
- 🗣️🎸 CoreDataPodcast
- 📅 CoreDataPodcast objects exist in the managed object context
- 🎙️ Until the objects are saved using the persistent store coordinator, they are not persistent.
- 💿 Once the objects are persisted, they can be reconstituted back into CoreDataPodcasts.

# Core Data

## What exactly is Core Data?

- It's a framework that gives you extremely powerful persistence at the cost of complexity.
- Core Data has many, many moving parts that you can configure: the objects themselves (and their relations), fetch requests, the contexts, the persistent store coordinator, the persistent store.
- Do not use for simple apps. Saving a "small" JSON to disk using Codable is going to be easier and faster.
- Based in Objective-C.



# Core Data

## More Advanced Concepts

- **NSFetchedResultsController**  
<https://cocoacasts.com/populate-a-table-view-with-nsfetchedresultscontroller-and-swift-3>
- **NSPredicate**  
<https://nshipster.com/npredicate/>
- **BackgroundContexts**  
<https://www.raywenderlich.com/7586-multiple-managed-object-contexts-with-core-data-tutorial>
- **Synchronization between Core Data and remote data source**  
<https://www.donnywals.com/implementing-a-one-way-sync-strategy-with-core-data-urllsession-and-combine/>
- **Performing migrations**  
<https://www.raywenderlich.com/7585-lightweight-migrations-in-core-data-tutorial>
- **Debugging SQLite issues**  
<https://www.avanderlee.com/debugging/core-data-debugging-xcode/>
- **Listening for changes**  
[https://developer.apple.com/documentation/coredata/consuming\\_relevant\\_store\\_changes](https://developer.apple.com/documentation/coredata/consuming_relevant_store_changes)

# Next Class

Coming Soon via the mailing list!

# Support your friendly neighborhood instructor

**I'm no Spiderman but I try**

- **Practice Swift weekly with a 15-minute exercise.**

<https://mailchi.mp/hey/weekly-swift-exercise-signup>

- **Donations are welcome and include the recorded session.**

<https://paypal.me/fromjuniortosenior>

- **Private 1-1 sessions (\$49)**

[https://fernando-olivares-s-school.teachable.com/purchase?product\\_id=2927003](https://fernando-olivares-s-school.teachable.com/purchase?product_id=2927003)

- **2-week course with code reviews (\$299)**

<https://fernando-olivares-s-school.teachable.com/p/from-junior-to-senior>

Supporting me helps me keep classes free.

**Q&A**