CORE DATA

Jonathan Penn
#inspect 2013 - Brussels, Belgium

@jonathanpenn

Navel Labs

COCOA MANIFEST

Slides ‘n Sample Code

cocoamanifest.net/features

“Why would I use Core Data?”

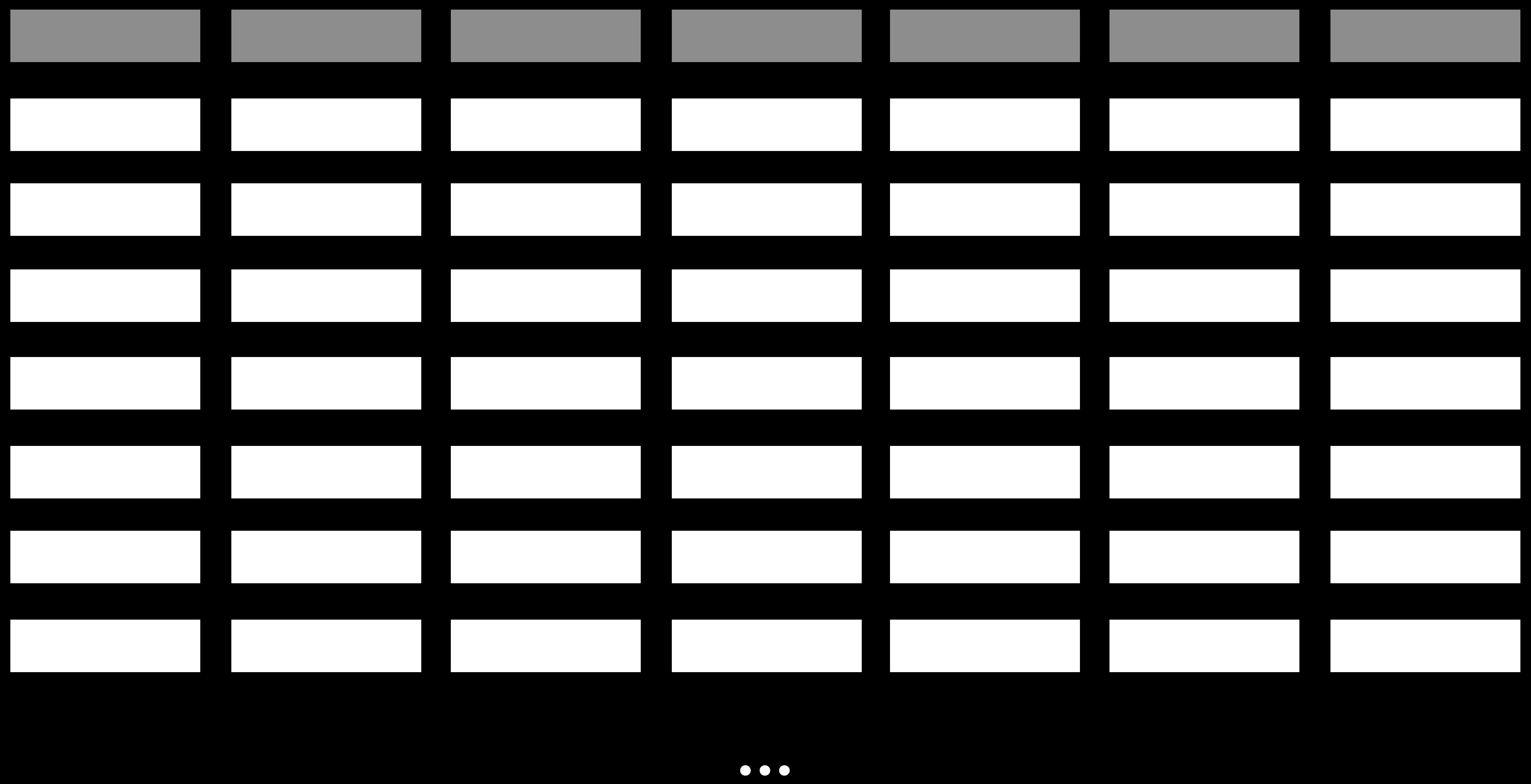
ONE DOES NOT SIMPLY
USE CORE DATA

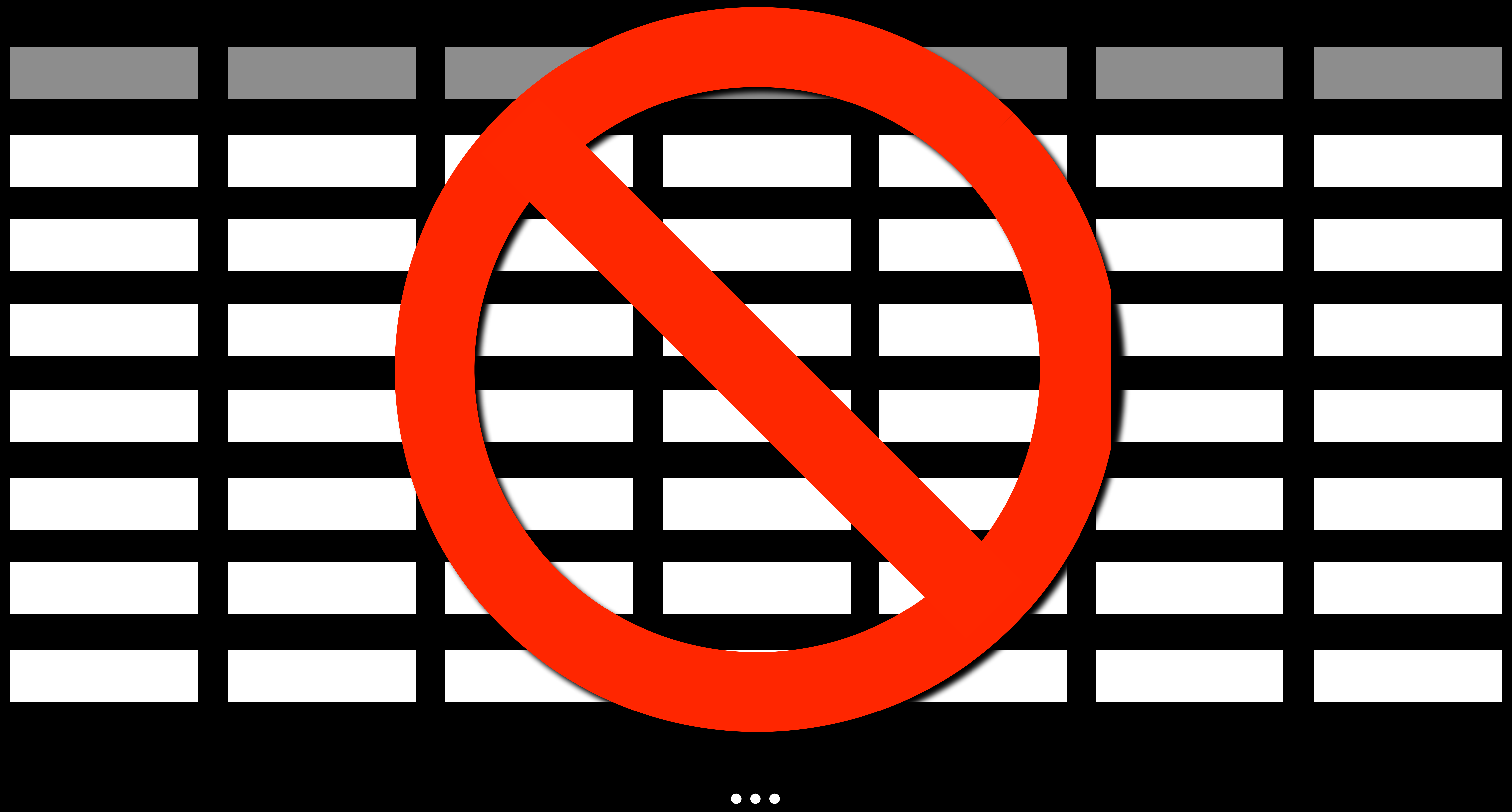
It’s a lifestyle.

Goals:

1. Help you decide
2. Help you understand

Is it a SQLite ORM?

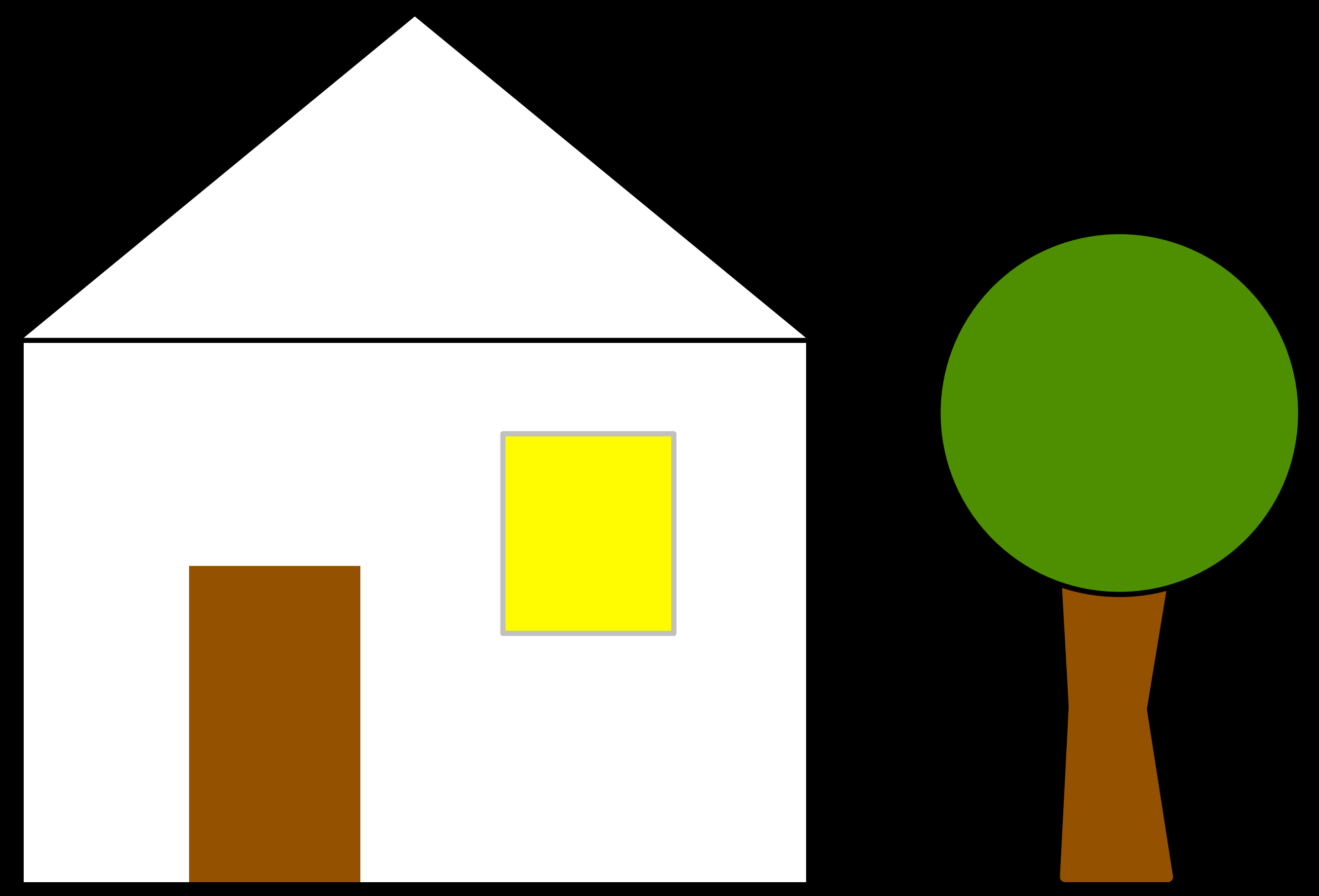


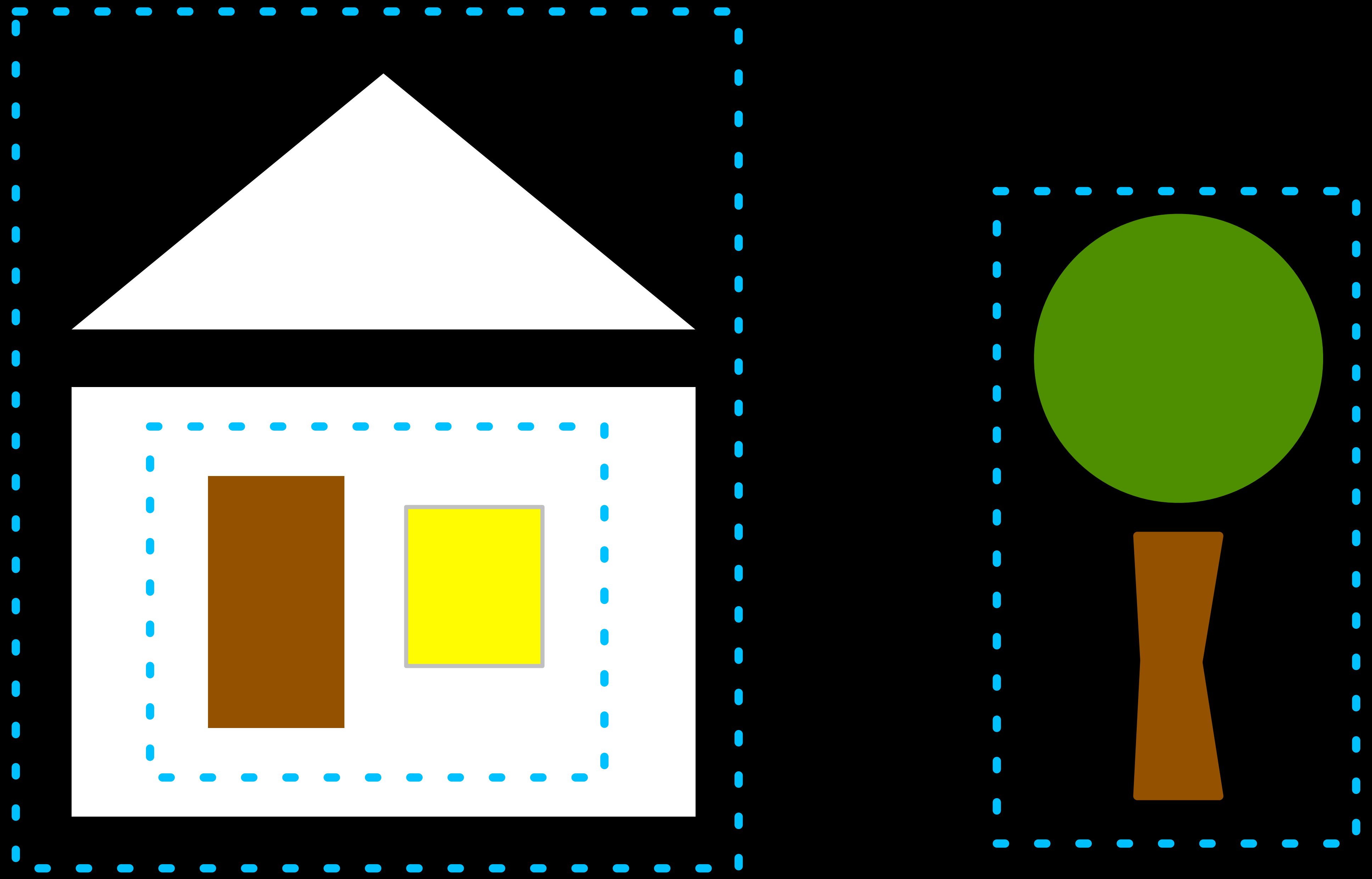


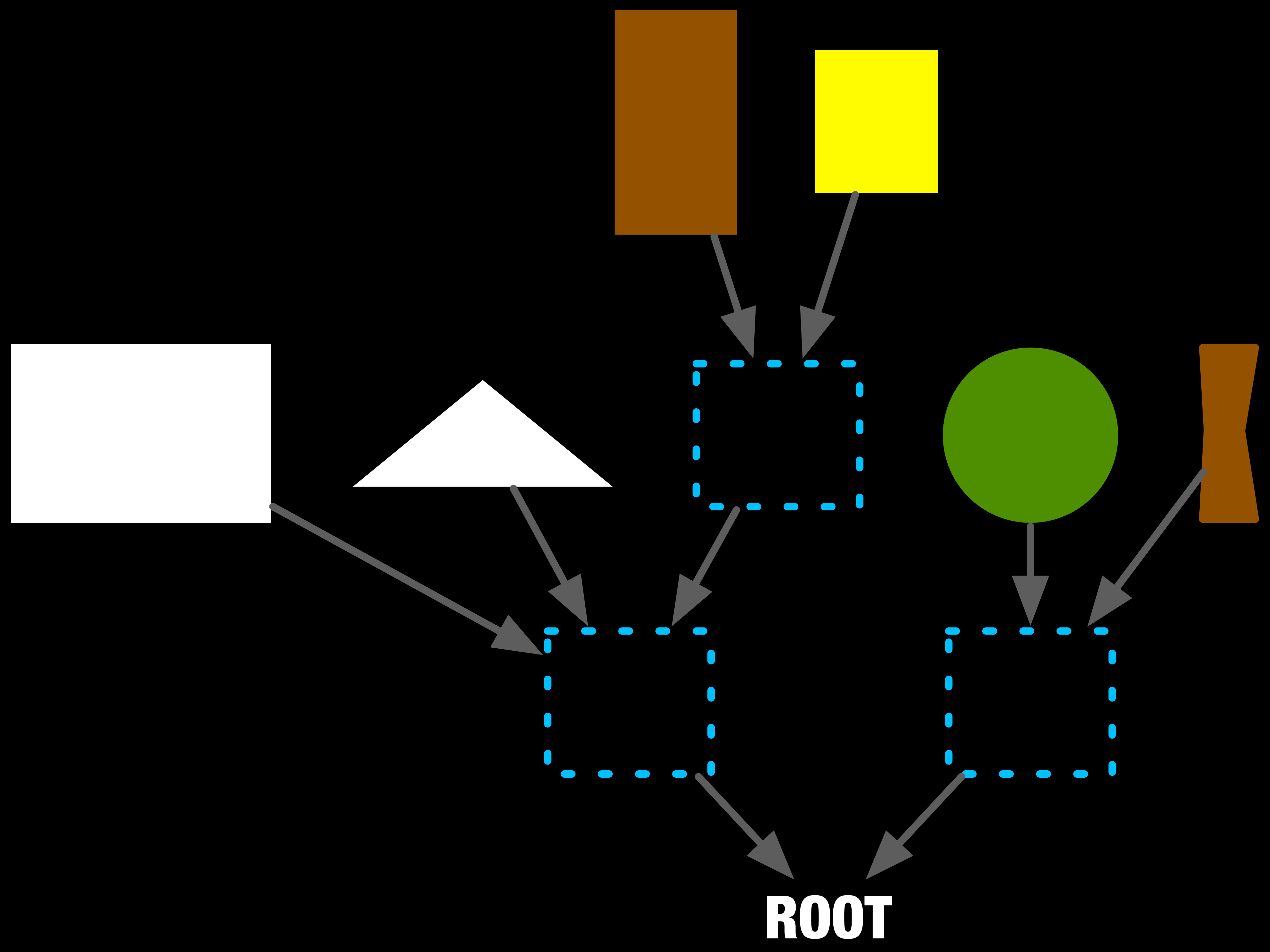
Object Graph

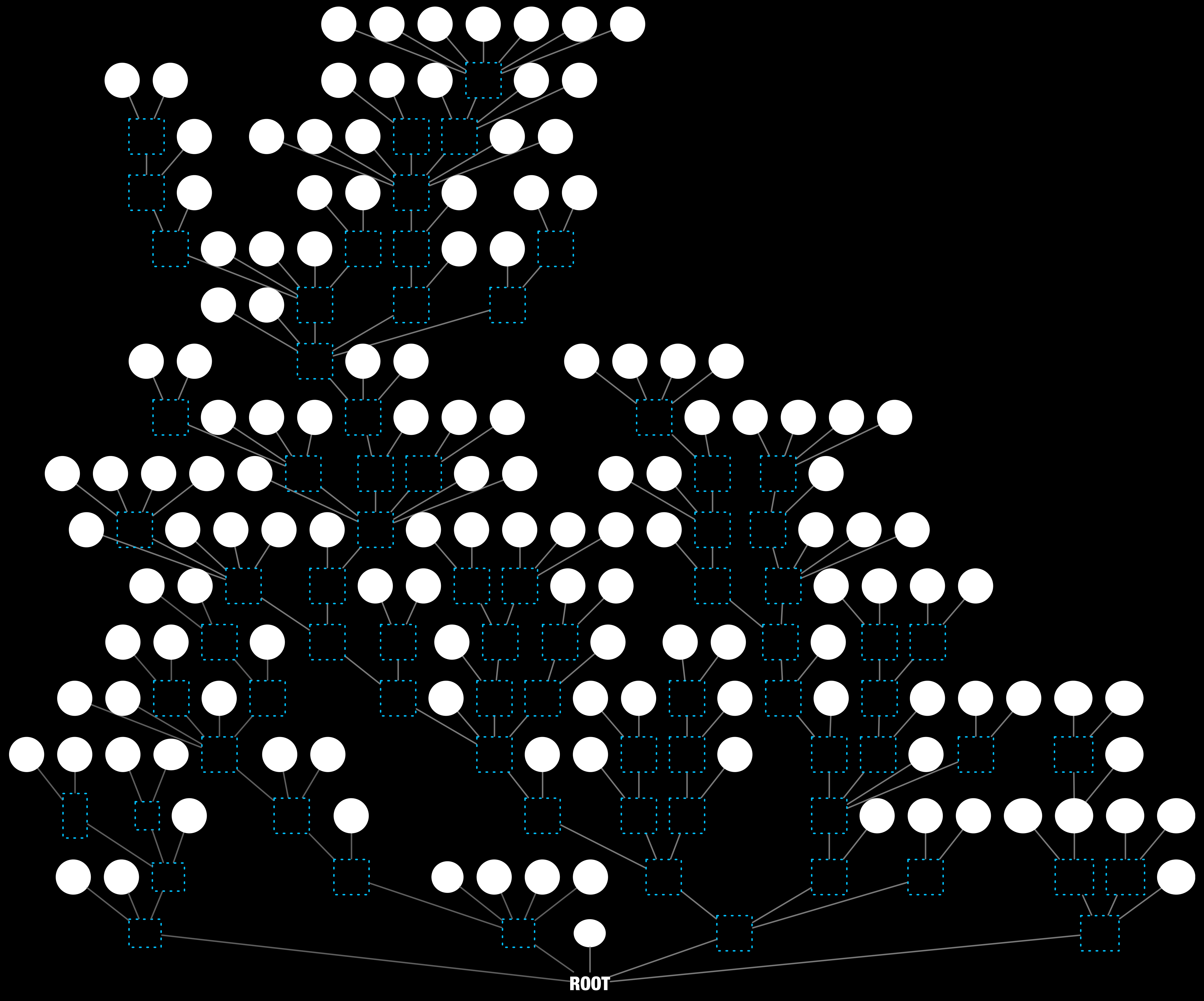
Mmmm, goodies!

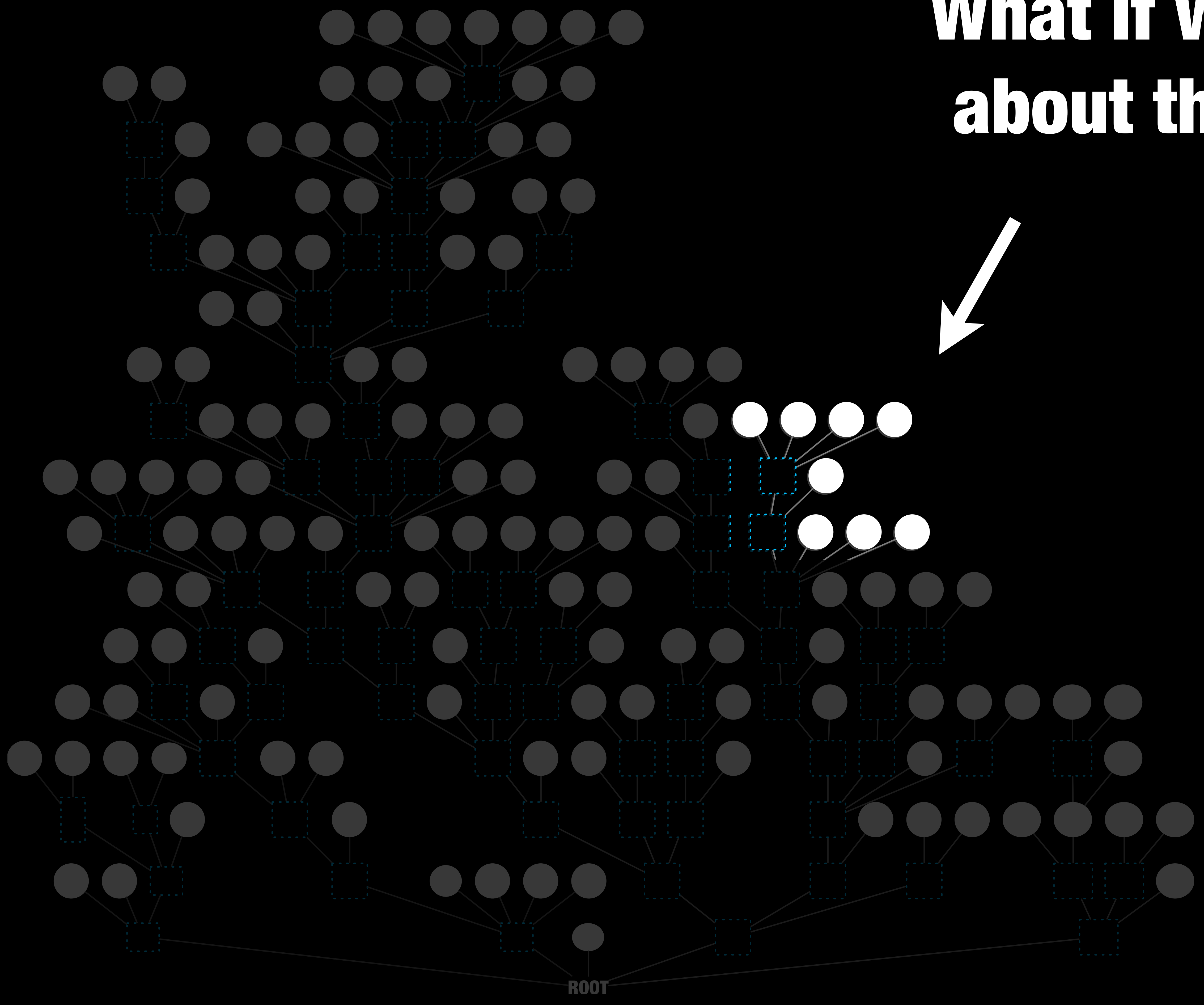
For example…






ROOT


ROOT



What if we only care about this portion?

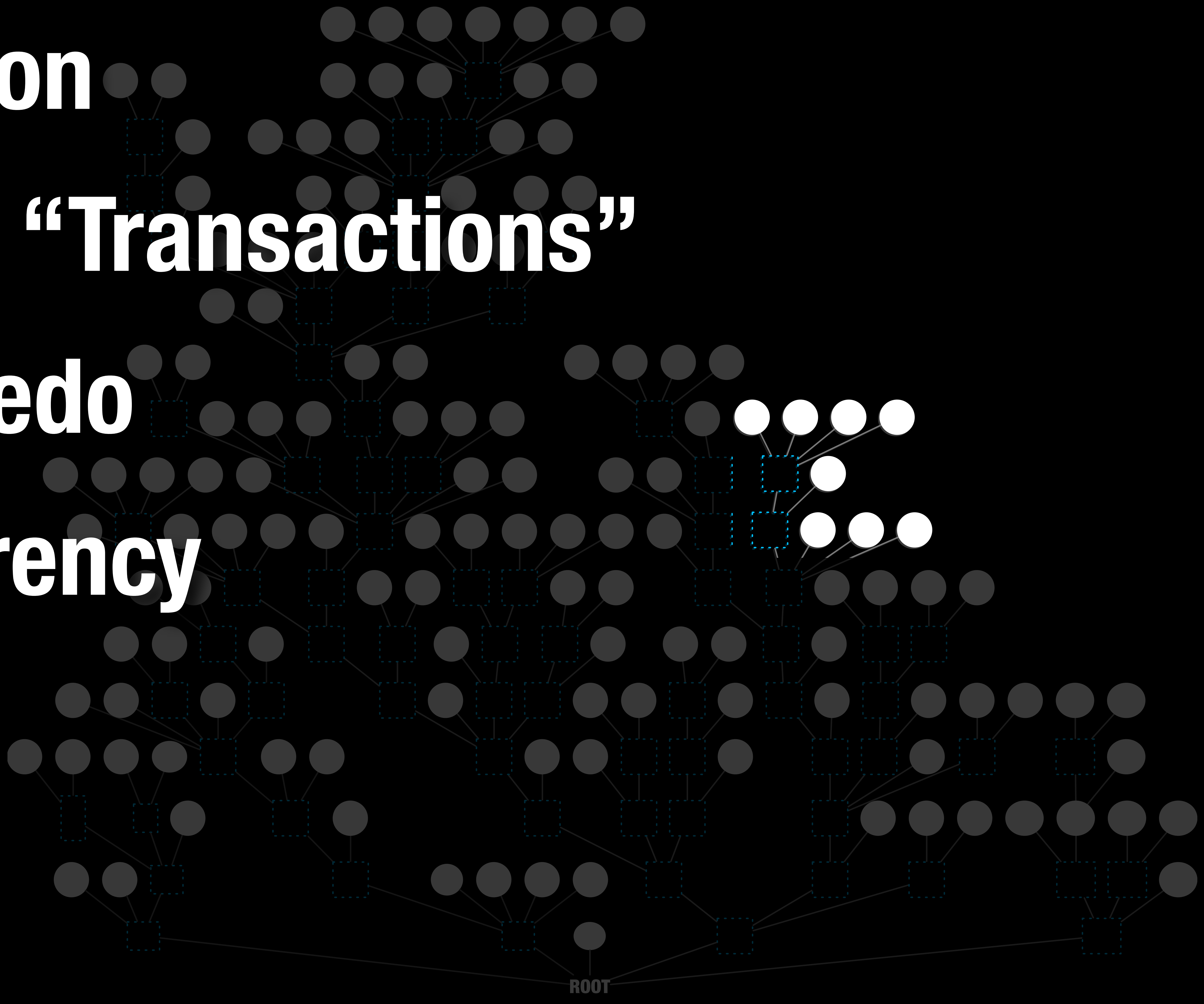
“Virtual Memory”

Validation

Context “Transactions”

Undo/Redo

Concurrency



NSManagedObjectContext

NSManagedObject

NSManagedObjectModel

NSPersistentStoreCoordinator

NSFetchedResultsController

NSManagedObjectContext

Fetches, inserts, removes objects

Represents a “transaction”

Saving “commits” the transaction

Notifies observers of changes

NSManagedObjectContext

Inserting

```
context = ... # Some context

entity = NSEntityDescription.entityForName("Shape",
           inManagedObjectContext: context)

shape  = NSManagedObject.alloc.initWithEntity(entity,
           insertIntoManagedObjectContext: context)
```

NSManagedObjectContext

Removing

```
context.deleteObject(shape)
```

NSManagedObjectContext

Saving

```
if !context.save(.........)

  NSLog("Something went wrong")
end
```

NSManagedObjectContext

Saving

```
error_ptr = Pointer.new(:object)

if !context.save(error_ptr)
  error = error_ptr[0]
  NSLog("Something went wrong: %@, %@", error, error.userInfo)
end
```

NSManagedObjectContext

Fetching

???

```
error_ptr = Pointer.new(:object)
results = context.executeFetchRequest(request,
                                      error: error_ptr)

if results.nil?
  error = error_ptr[0]
  NSLog("Something went wrong: %@, %@", error, error.userInfo)
else
  NSLog("Results: %@", results)    # Array of results
end
```

NSManagedObjectContext

Fetching

```
request = NSFetchRequest.alloc.initWithEntityName("Shape")

request.sortDescriptors = [
  NSSortDescriptor.sortDescriptorWithKey("zIndex",
                                         ascending: false)
]

request.predicate = NSPredicate.predicateWithFormat(
                                  "x < %@ and y < %@",
                                  xLimit, yLimit)
```

NSManagedObjectContext

Multiple contexts

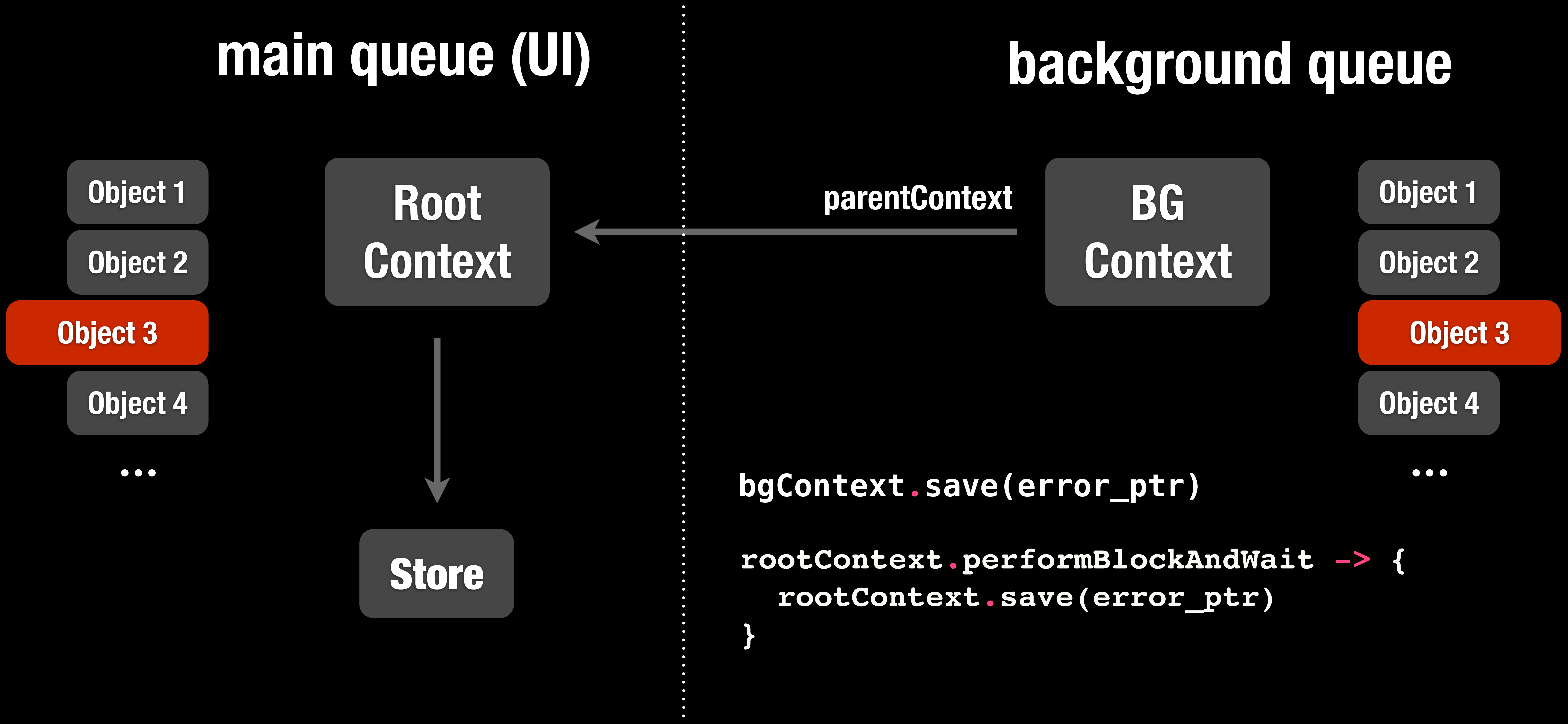
At least one per thread/concurrent queue

Parent / Child

Save pushes changes to parent

Saving on root context writes to disk

NSManagedObjectContext



main queue (UI)

background queue

Object 1

Object 2

Object 3

Object 4

…

Root Context

Store

parentContext

BG Context

Object 1

Object 2

Object 3

Object 4

…

```
bgContext.save(error_ptr)

rootContext.performBlockAndWait -> {
  rootContext.save(error_ptr)
}
```

It’s how you reach
your stuff

NSManagedObject

The tangible instances of your “entities”

Subclassed for extra behavior

Complex validations

NSManagedObject

```
shape.x

shape.zIndex = 1000

shape.parent.zIndex

shape.children
```

NSManagedObject

```
class Shape < NSManagedObject





end
```

NSManagedObject

```
class Shape < NSManagedObject

  def self.insertInContext context
    NSEntityDescription.insertNewObjectForEntityForName(
      "Shape", inManagedObjectContext: context)
  end

  def self.fetchRequest
    NSFetchRequest.alloc.initWithEntityName("Shape")
  end

end
```

NSManagedObject

Belong to a single context

Belong to a single context!!!!

Pass around NSManagedObjectIDs

When in doubt, fetch again from a context

Your stuff

NSManagedObjectModel

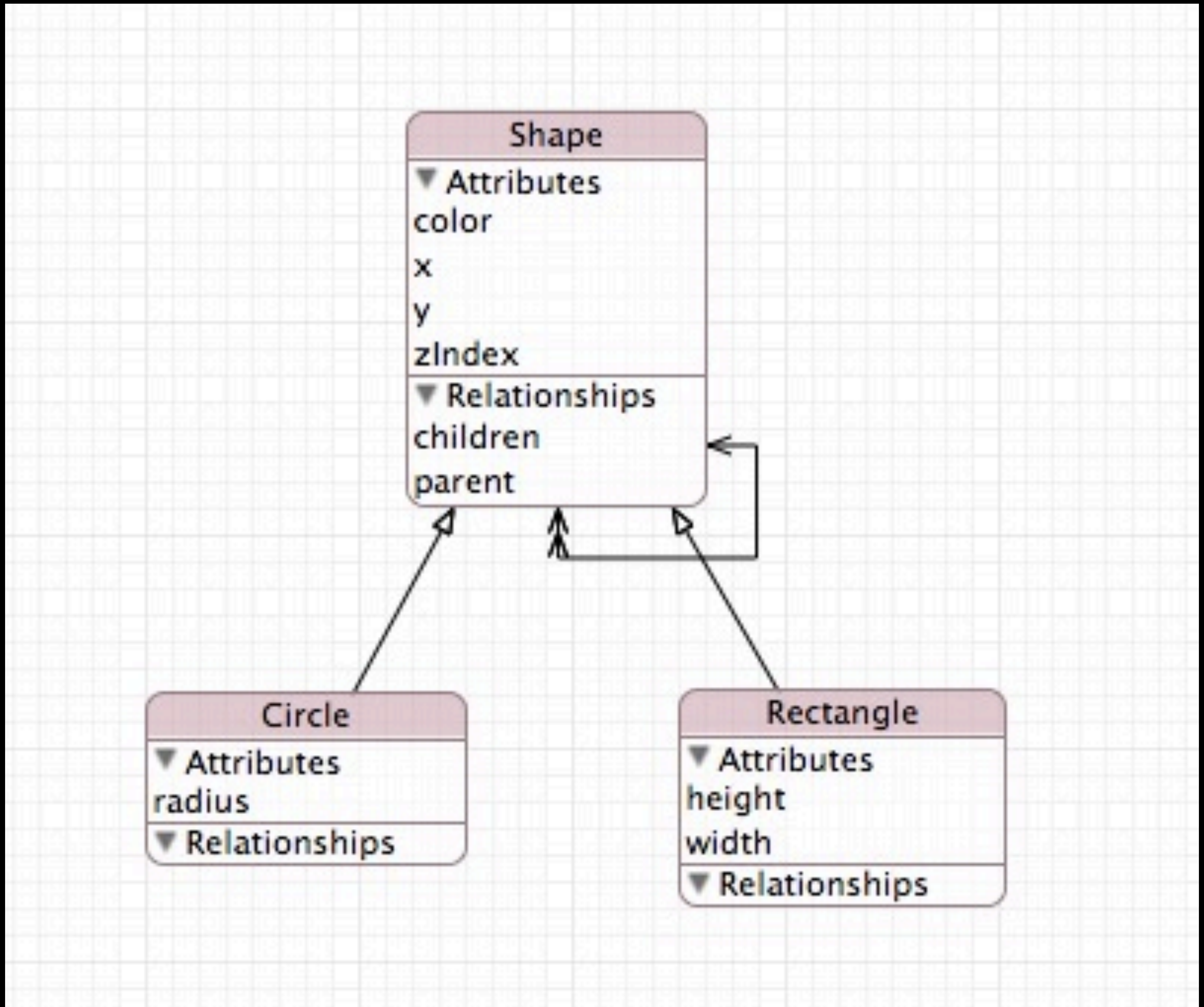
Describes entities and their relationships

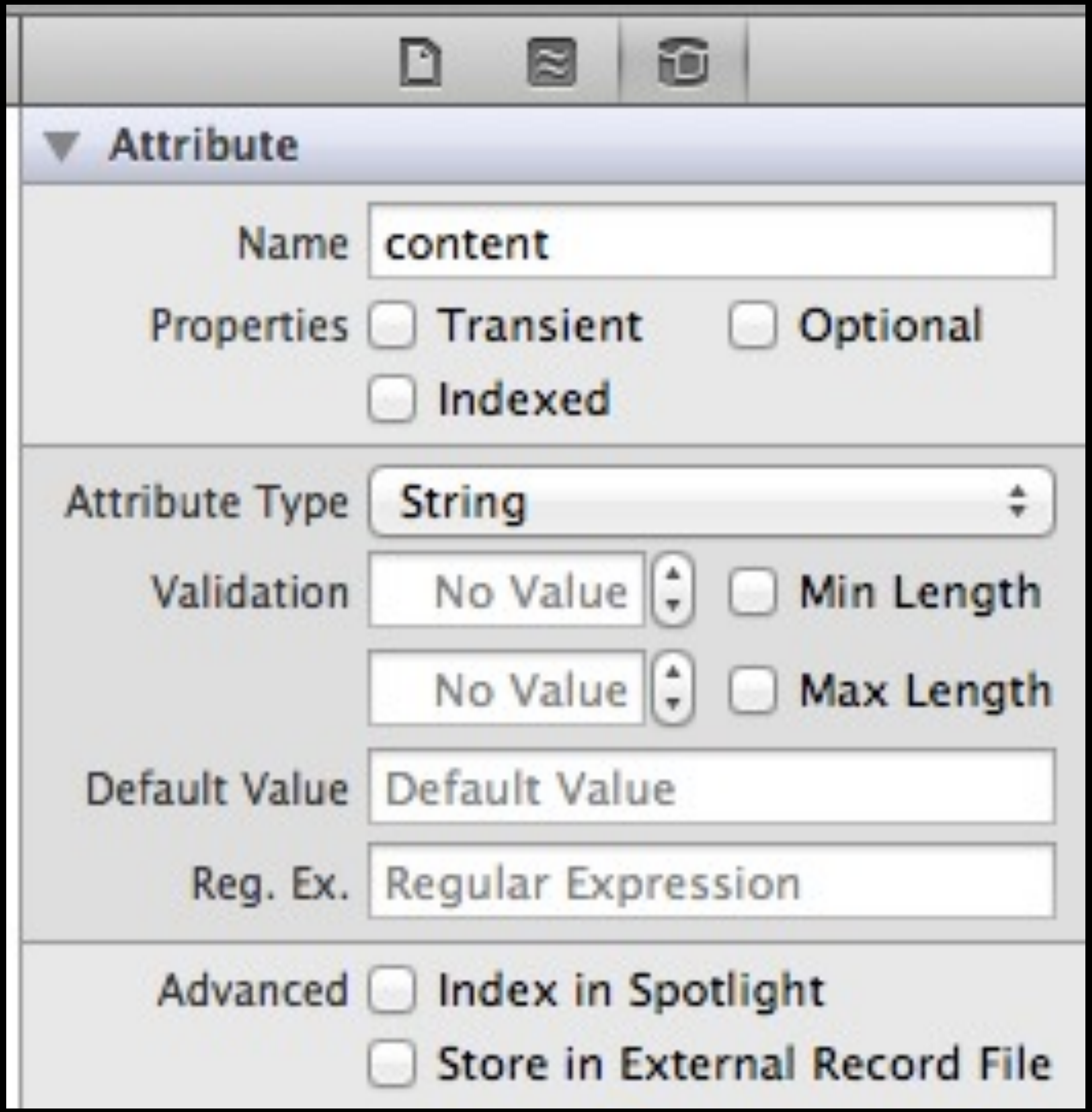
Attributes

Simple validations

Lightweight migrations

NSManagedObjectModel





Attribute

Name: content

Properties: □ Transient □ Optional □ Indexed

Attribute Type: String

Validation: No Value □ Min Length

No Value □ Max Length

Default Value: Default Value

Reg. Ex.: Regular Expression

Advanced: □ Index in Spotlight □ Store in External Record File

NSManagedObjectModel

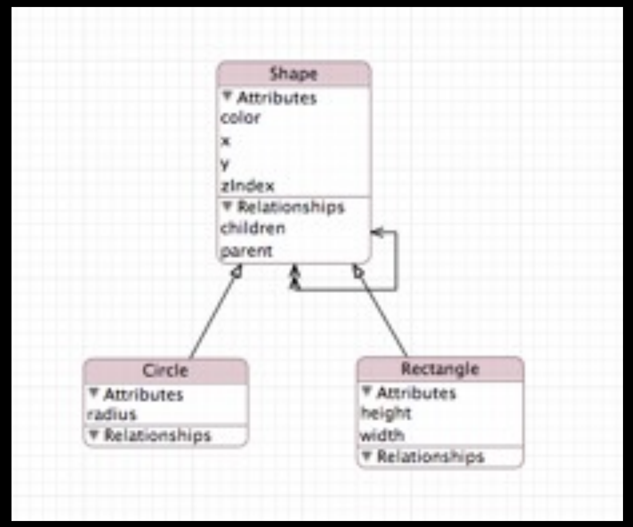
```
shape_entity = NSEntityDescription.new.tap do |entity|
  entity.name = 'Shape'
  entity.managedObjectClassName = 'Shape'
  entity.properties = [
    NSAttributeDescription.new.tap do |attr|
      attr.name = 'zIndex'
      attr.attributeType = NSDecimalAttributeType
      attr.optional = false
    end

    # ...
  ]
end

model.entities = [shape_entity]
```

NSManagedObjectModel

Motion Migrate

fousa.github.com/motion_migrate/

```ruby
class Plane < MotionMigrate::Model
  property :name,   :string
  property :multi, :boolean, :default => false

  belongs_to :pilot, :class_name => "Pilot",
                     :inverse_of => :planes
end
```



The schema
for your stuff

NSPersistentStoreCoordinator

... coordinates persistent stores!

You point it at a spot on disk

Tell it the store type you want

Contexts talk to it when necessary

NSFetchedResultsController

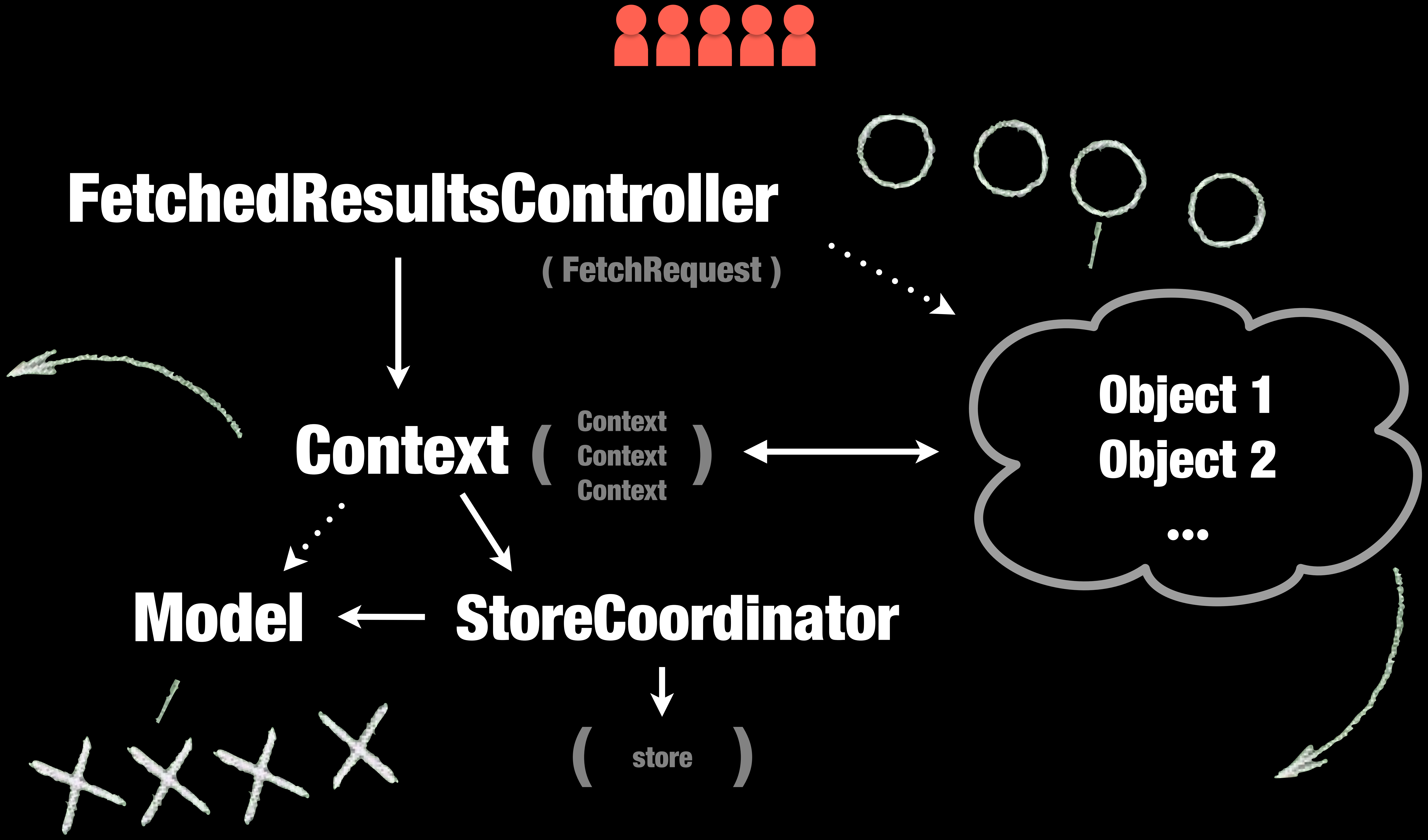
Tied to specific fetch request

Can be grouped into sections

Notifies *only* about relevant objects

Maintains ordering/grouping

Works great with table/collection views



FetchedResultsController

(FetchRequest)

Context (Context Context Context)

Object 1

Object 2

...

Model

StoreCoordinator

(store)

Let’s build our own Core Data stack!

```
class Document
  attr_reader :context, :coordinator, :model
  # ...
end
```

```
@model = NSManagedObjectModel.mergedModelFromBundles(nil)
```

or

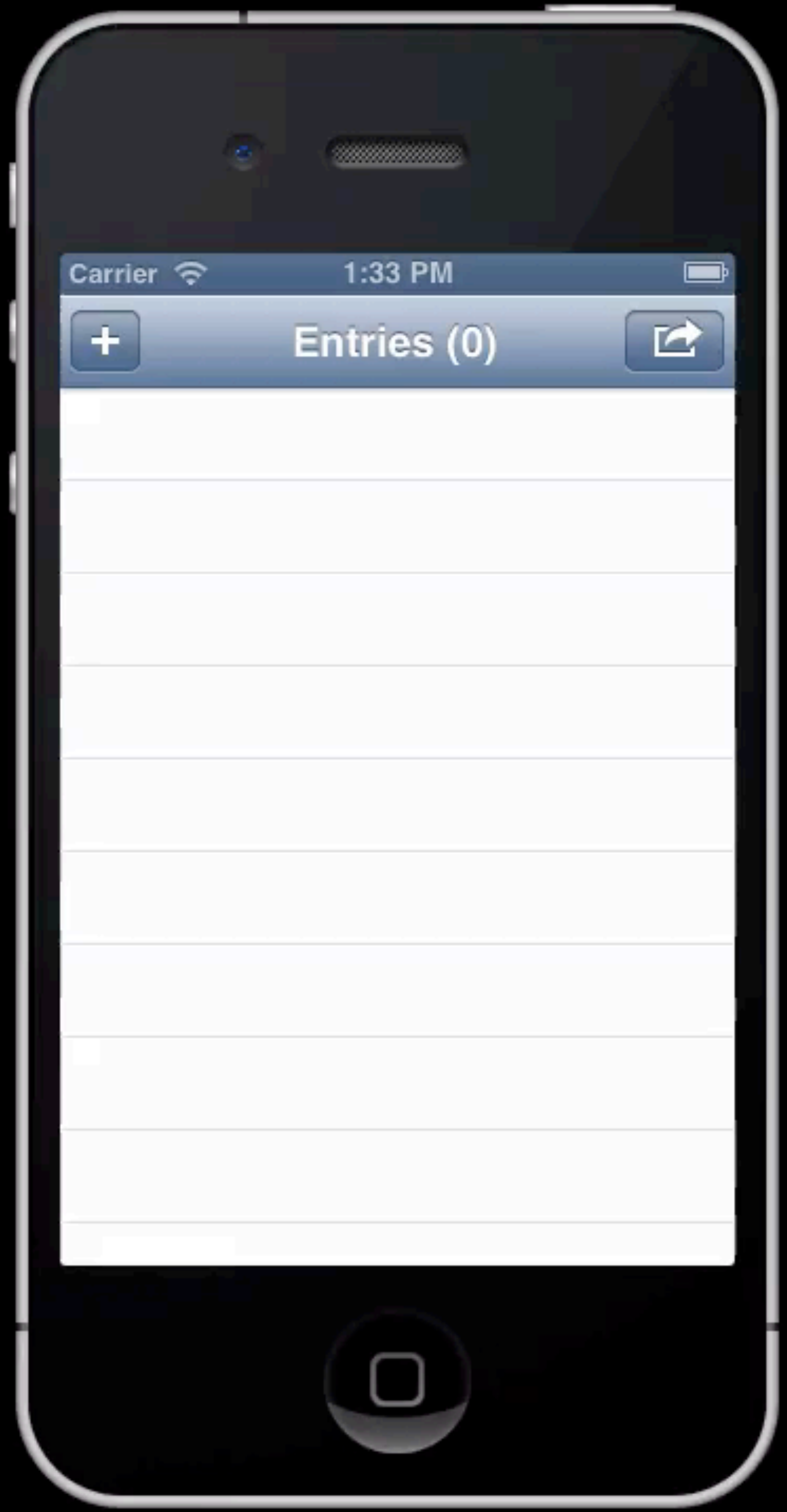
```
@model = NSManagedObjectModel.alloc.init
@model.entities = [...]
```

```
@coordinator = NSPersistentStoreCoordinator.alloc.
  initWithManagedObjectModel(@model)

url = NSURL.fileURLWithPath("some/path/in/documents")

error_ptr = Pointer.new(:object)
@coordinator.addPersistentStoreWithType(
  NSSQLiteStoreType,
  configuration: nil,
  URL: url,
  options: nil,
  error: error_ptr)
```

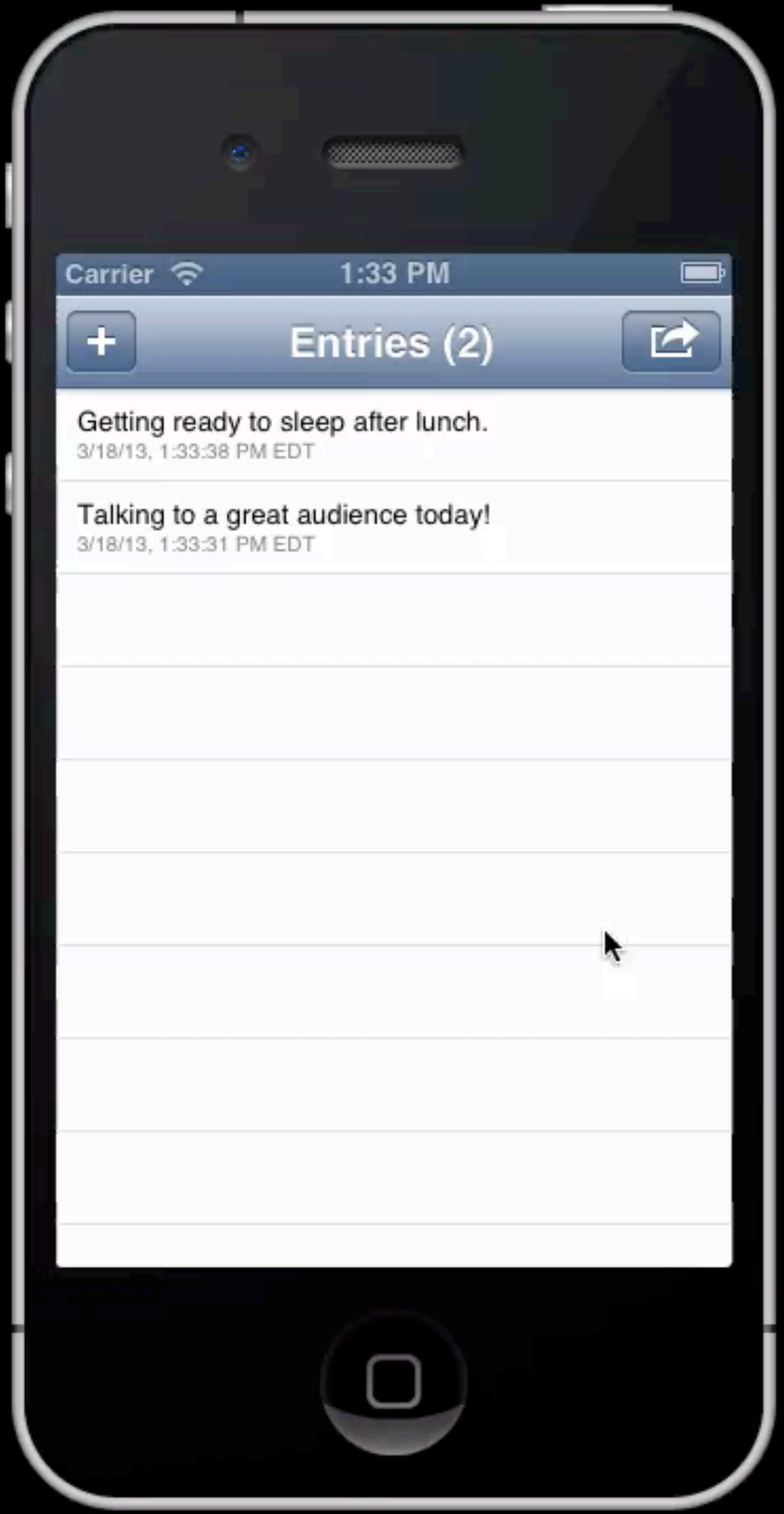
```
@context = NSManagedObjectContext.alloc.
      initWithConcurrencyType(NSMainQueueConcurrencyType)

@context.persistentStoreCoordinator = @coordinator
```

```
@doc = Document.new

@doc.context.executeFetchRequest(...)



@doc.newChildContext

@doc.saveRootContext

# ... etc
```

Demo!

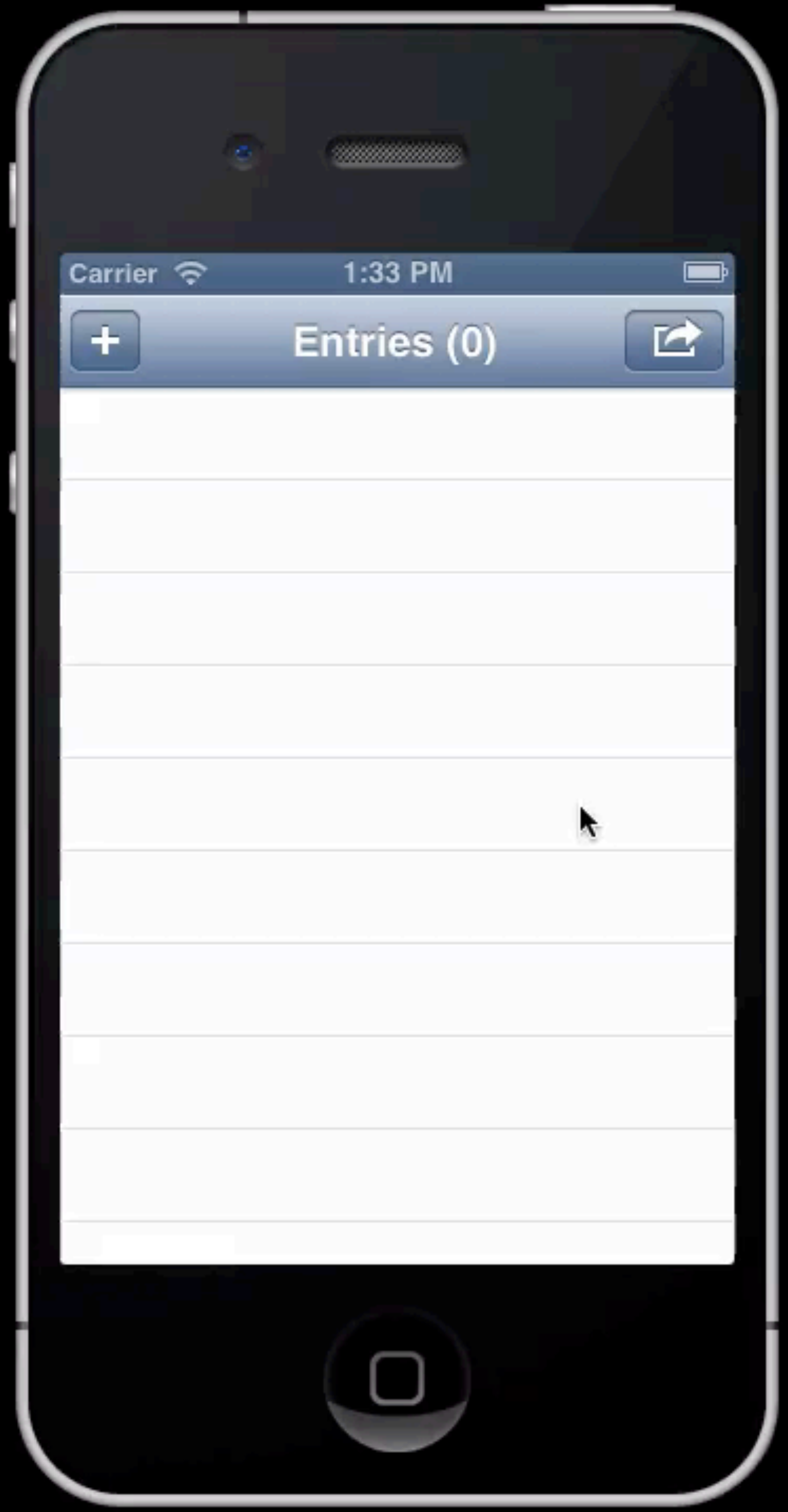


Carrier
1:33 PM
+
Entries (0)

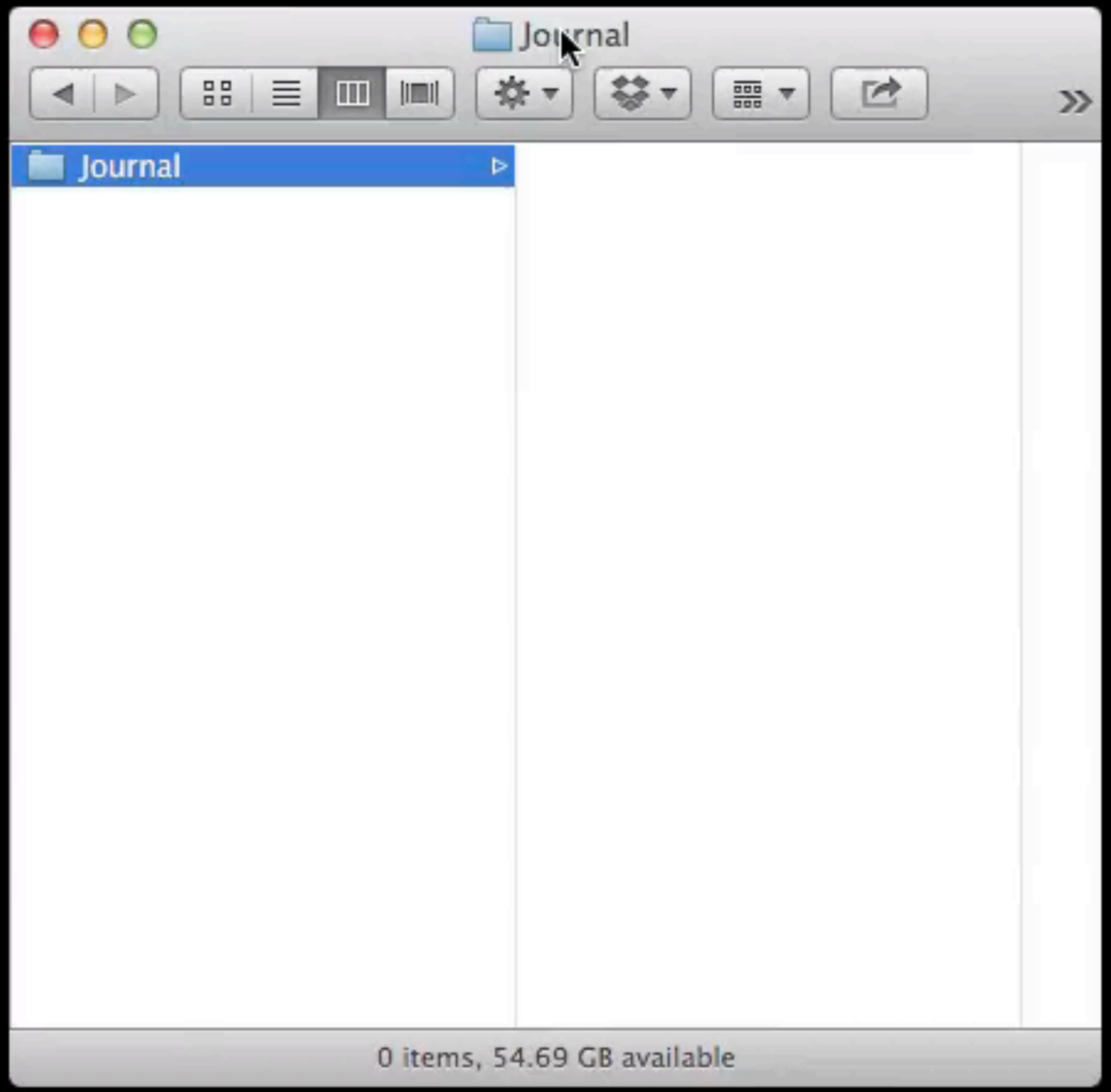


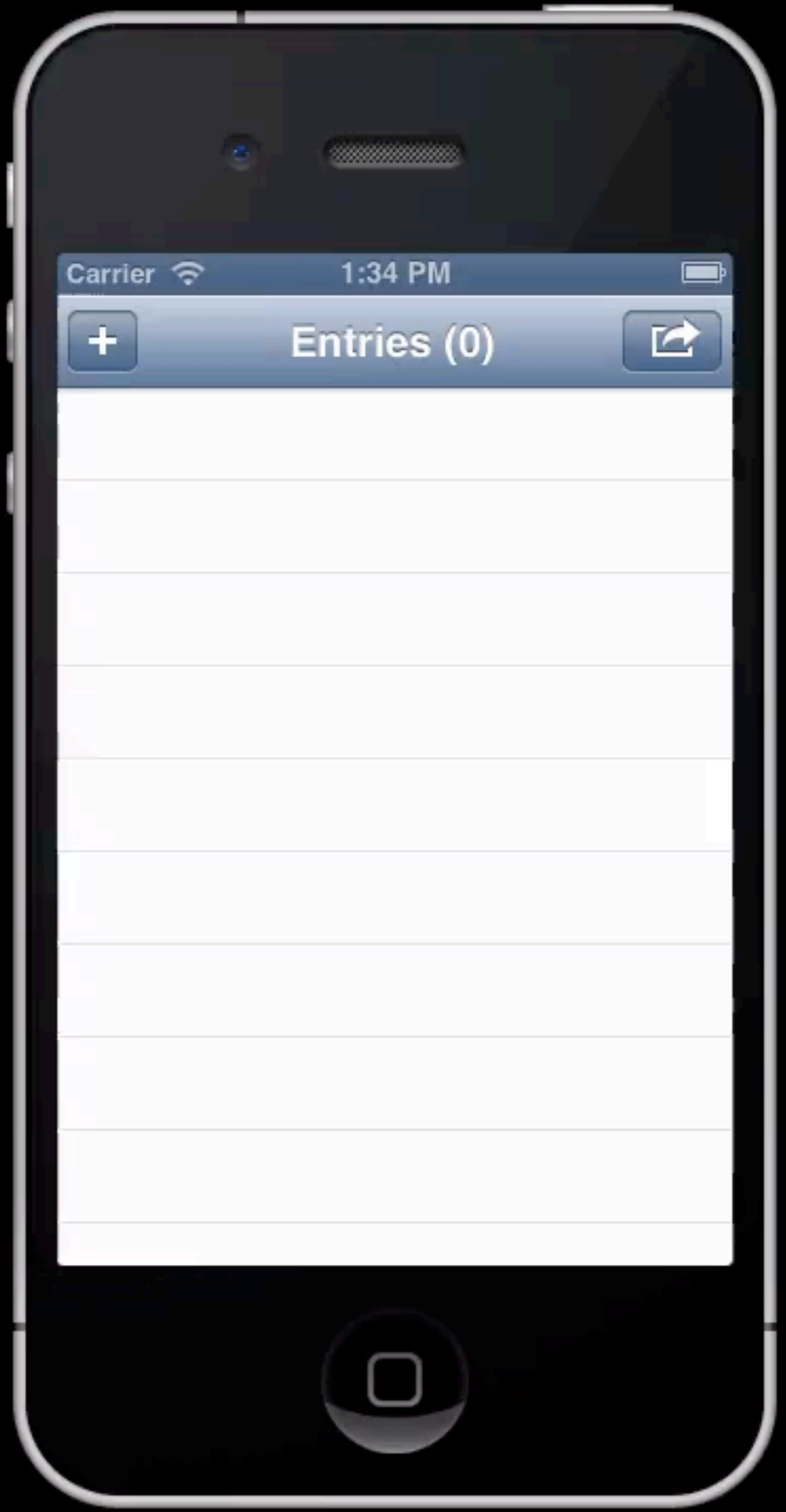
Getting ready to sleep after lunch.
3/18/13, 1:33:38 PM EDT

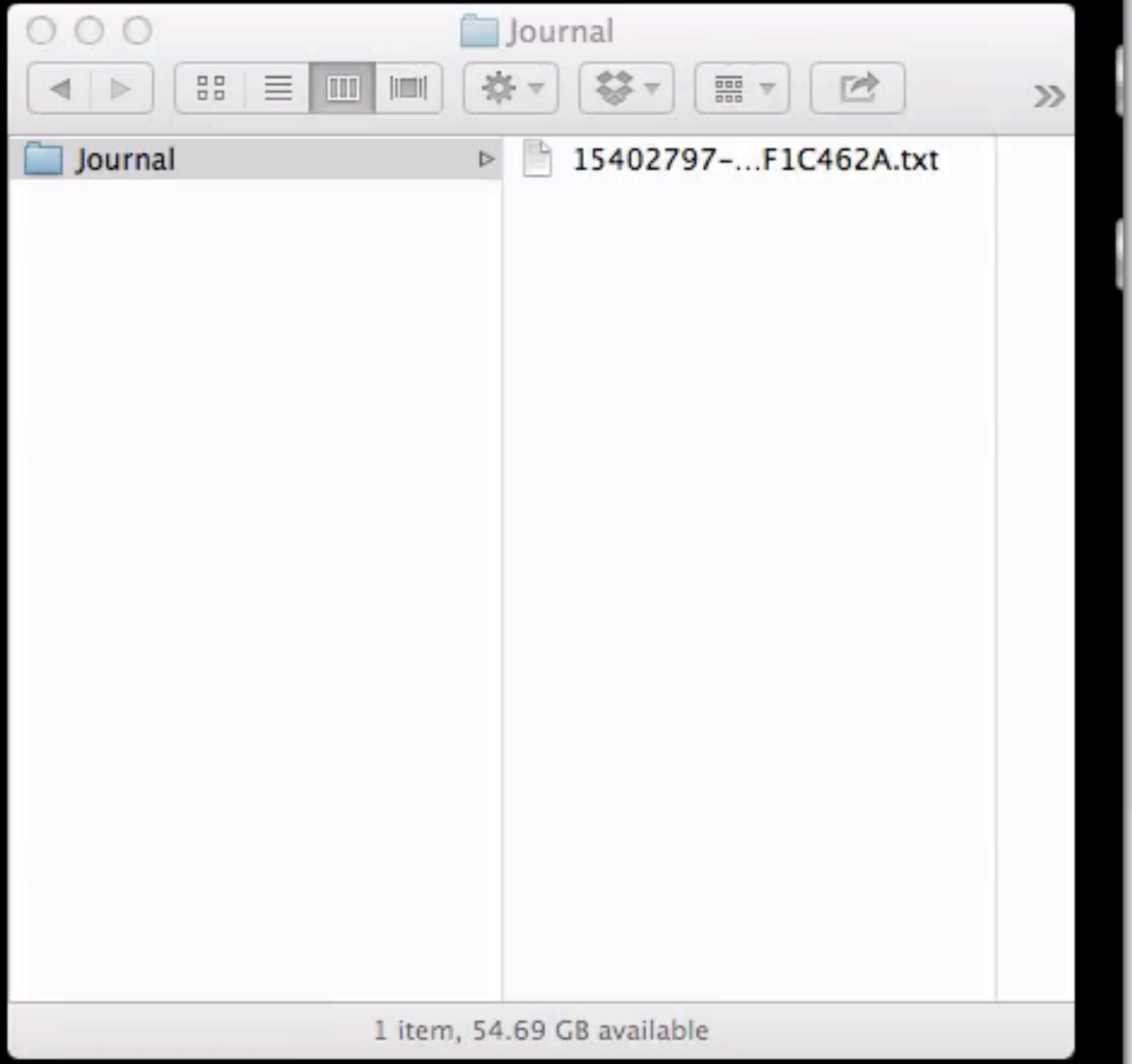
Talking to a great audience today!
3/18/13, 1:33:31 PM EDT



Carrier
1:33 PM
+
Entries (0)

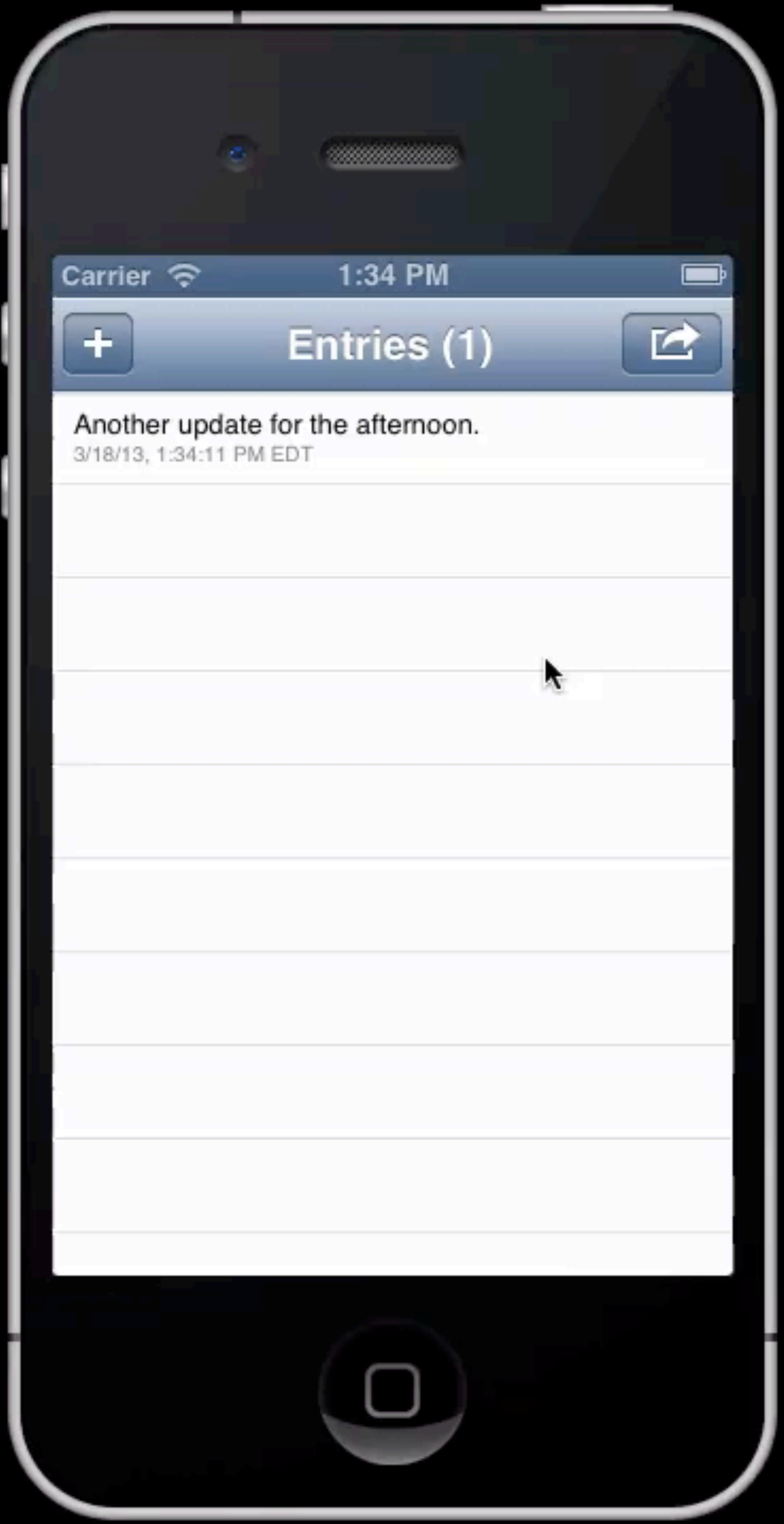

Journal
Journal
0 items, 54.69 GB available


Carrier
1:34 PM
+
Entries (0)

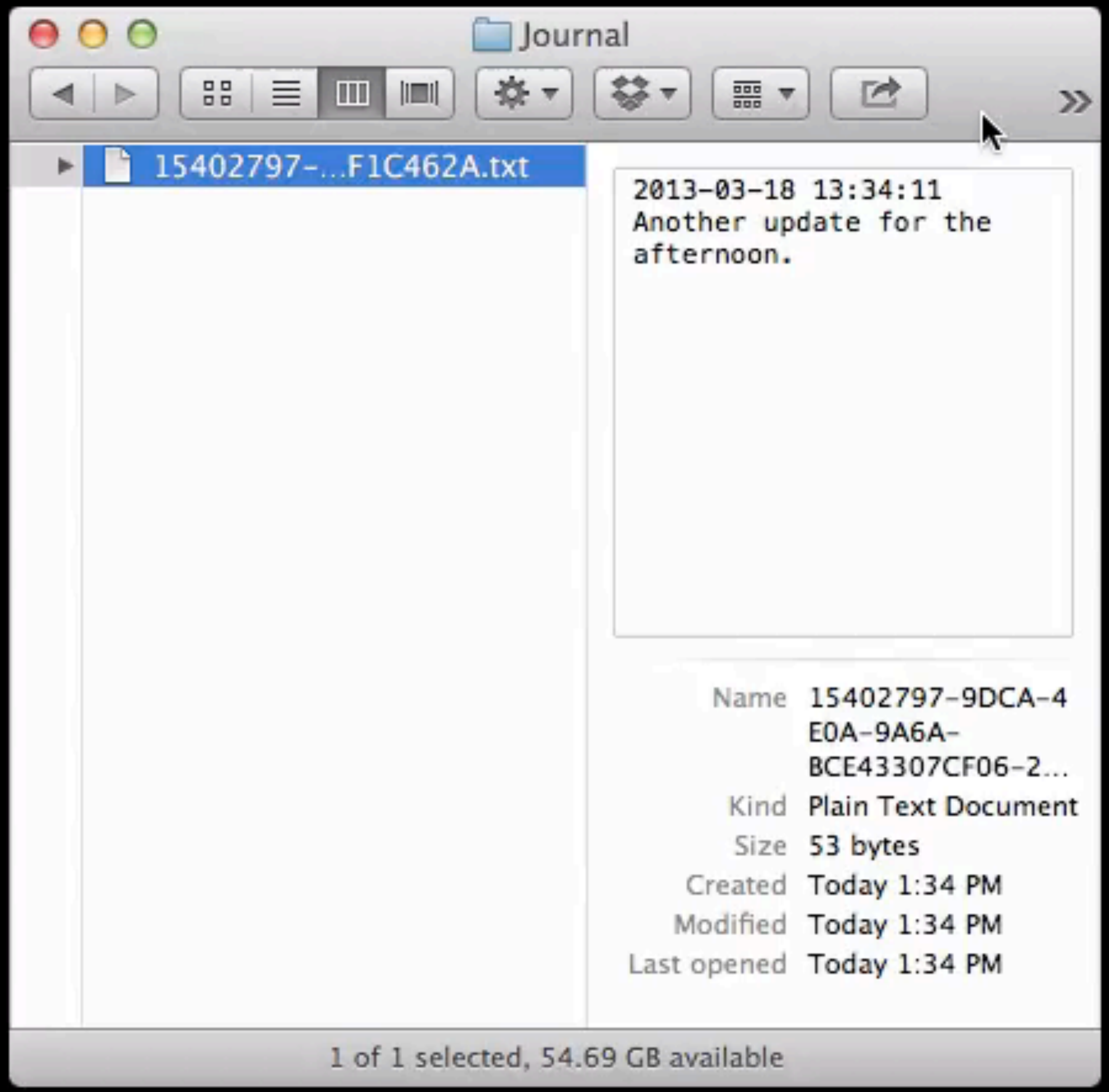


15402797-…F1C462A.txt

1 item, 54.69 GB available



Another update for the afternoon.
3/18/13, 1:34:11 PM EDT

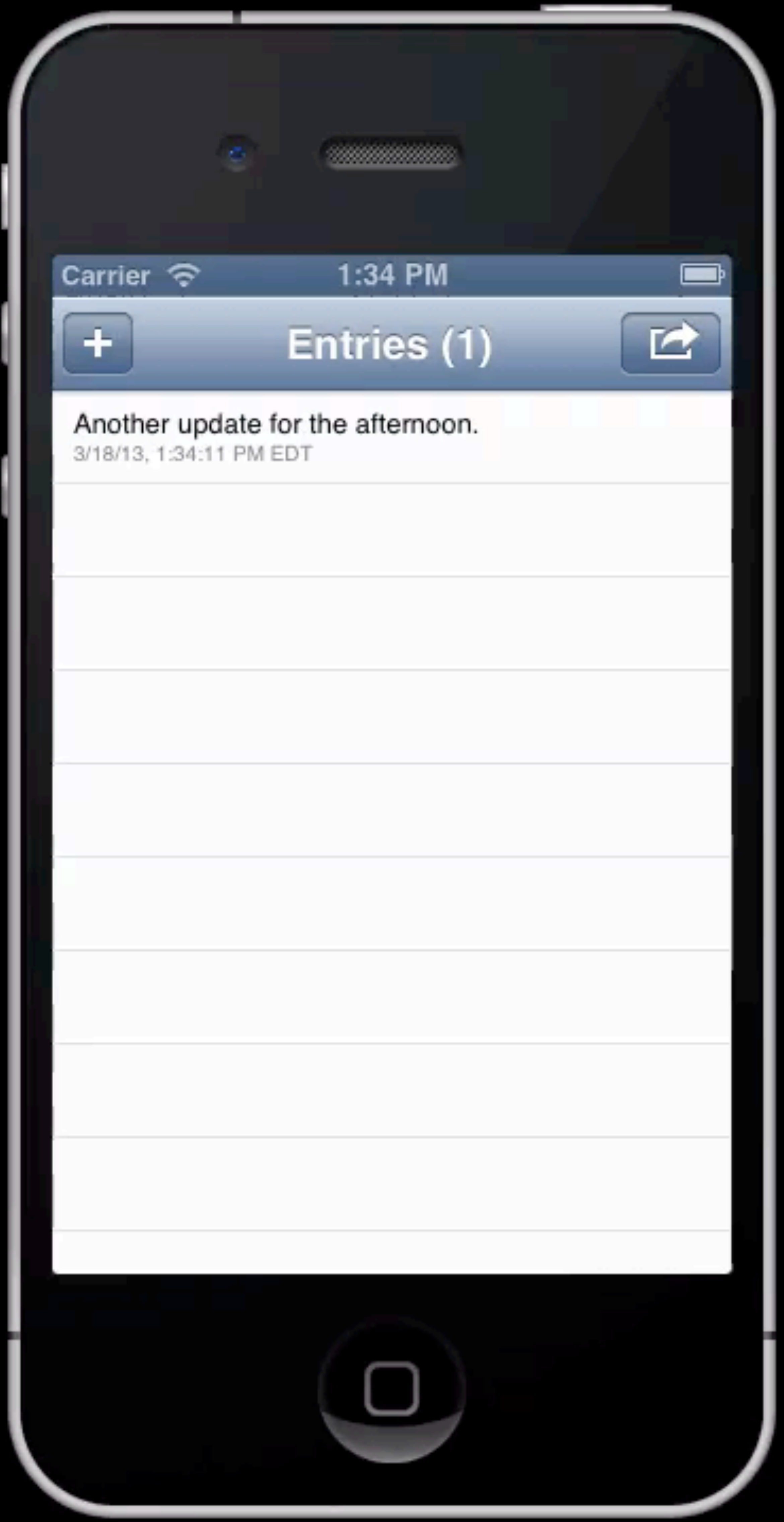


15402797-...F1C462A.txt

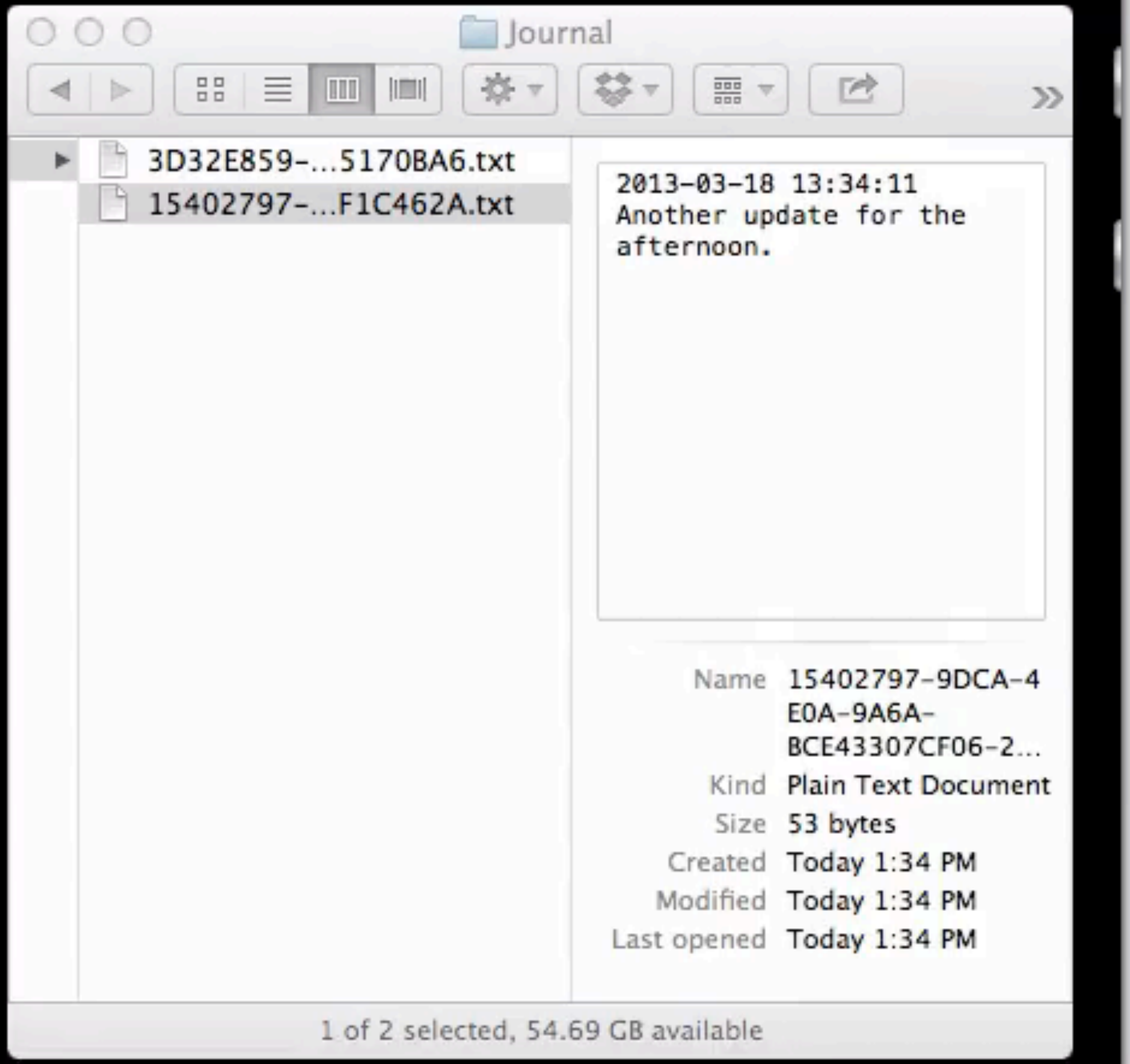
```
2013-03-18 13:34:11
Another update for the
afternoon.
```

Name	15402797-9DCA-4E0A-9A6A-BCE43307CF06-2...
Kind	Plain Text Document
Size	53 bytes
Created	Today 1:34 PM
Modified	Today 1:34 PM
Last opened	Today 1:34 PM

1 of 1 selected, 54.69 GB available



Another update for the afternoon.
3/18/13, 1:34:11 PM EDT

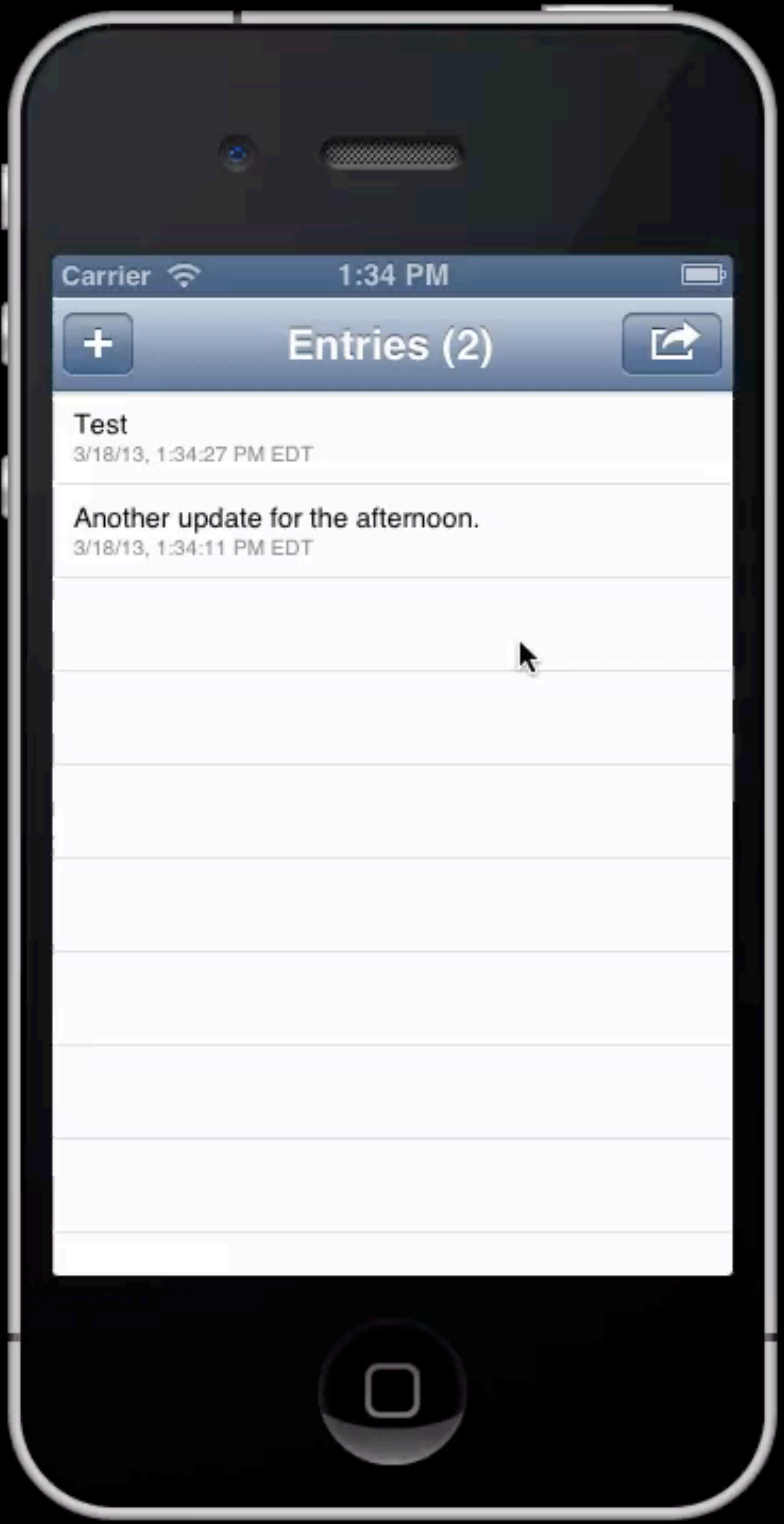


3D32E859-...5170BA6.txt
15402797-...F1C462A.txt

```
2013-03-18 13:34:11
Another update for the
afternoon.
```

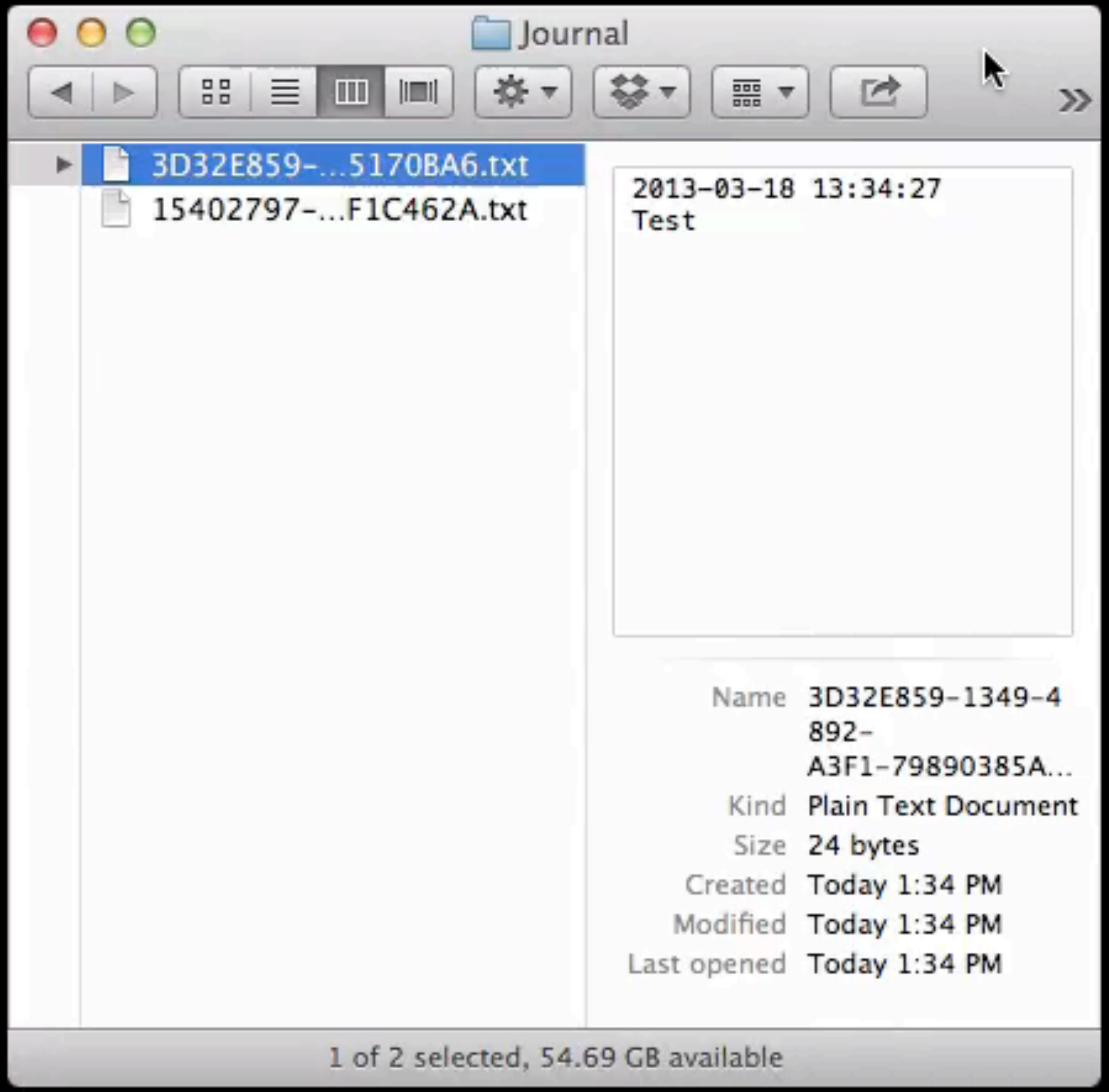
Name 15402797-9DCA-4E0A-9A6A-BCE43307CF06-2...
Kind Plain Text Document
Size 53 bytes
Created Today 1:34 PM
Modified Today 1:34 PM
Last opened Today 1:34 PM

1 of 2 selected, 54.69 GB available



Test
3/18/13, 1:34:27 PM EDT

Another update for the afternoon.
3/18/13, 1:34:11 PM EDT



3D32E859-...5170BA6.txt

15402797-...F1C462A.txt

```
2013-03-18 13:34:27
Test
```

Name 3D32E859-1349-4892-A3F1-79890385A...

Kind Plain Text Document

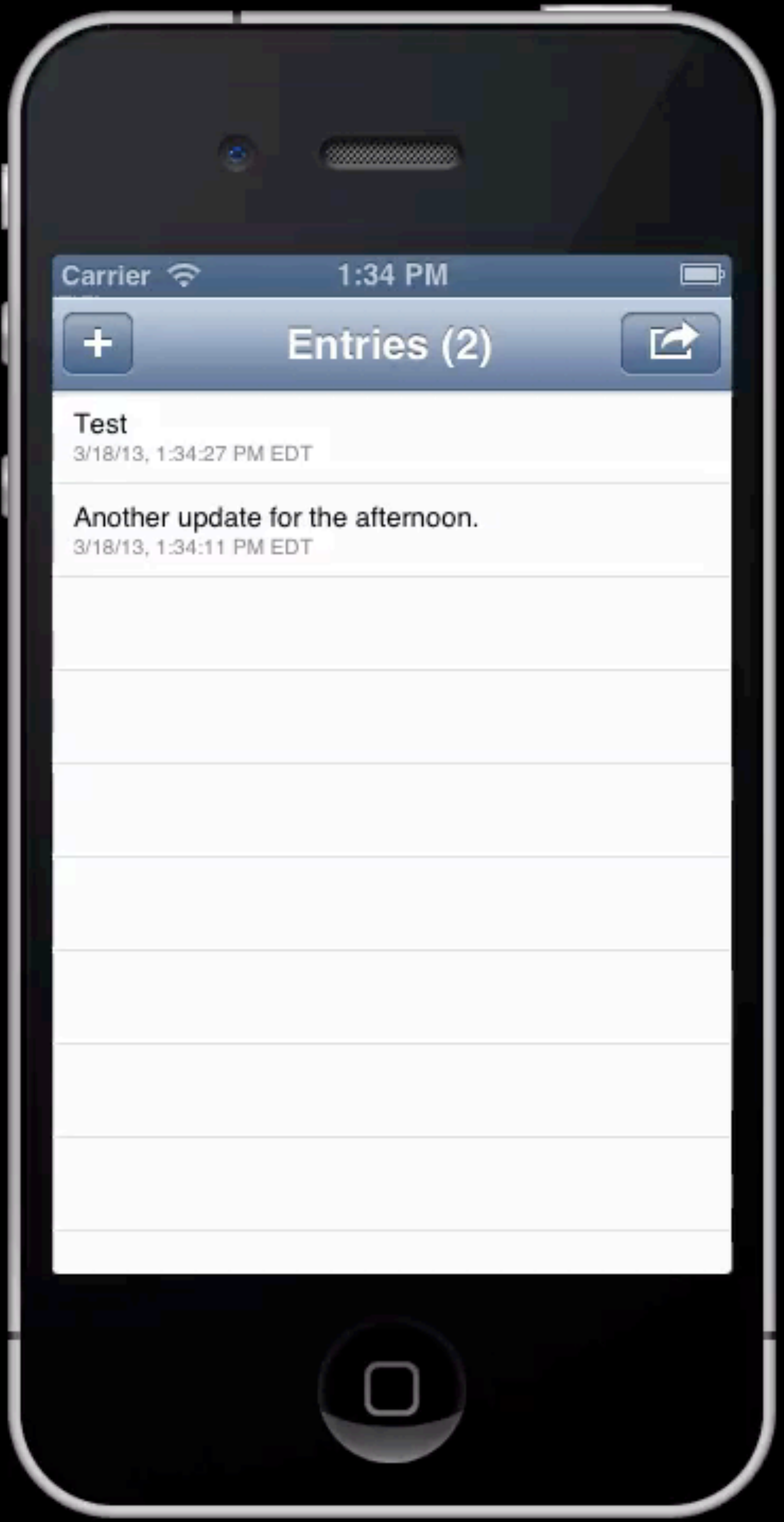
Size 24 bytes

Created Today 1:34 PM

Modified Today 1:34 PM

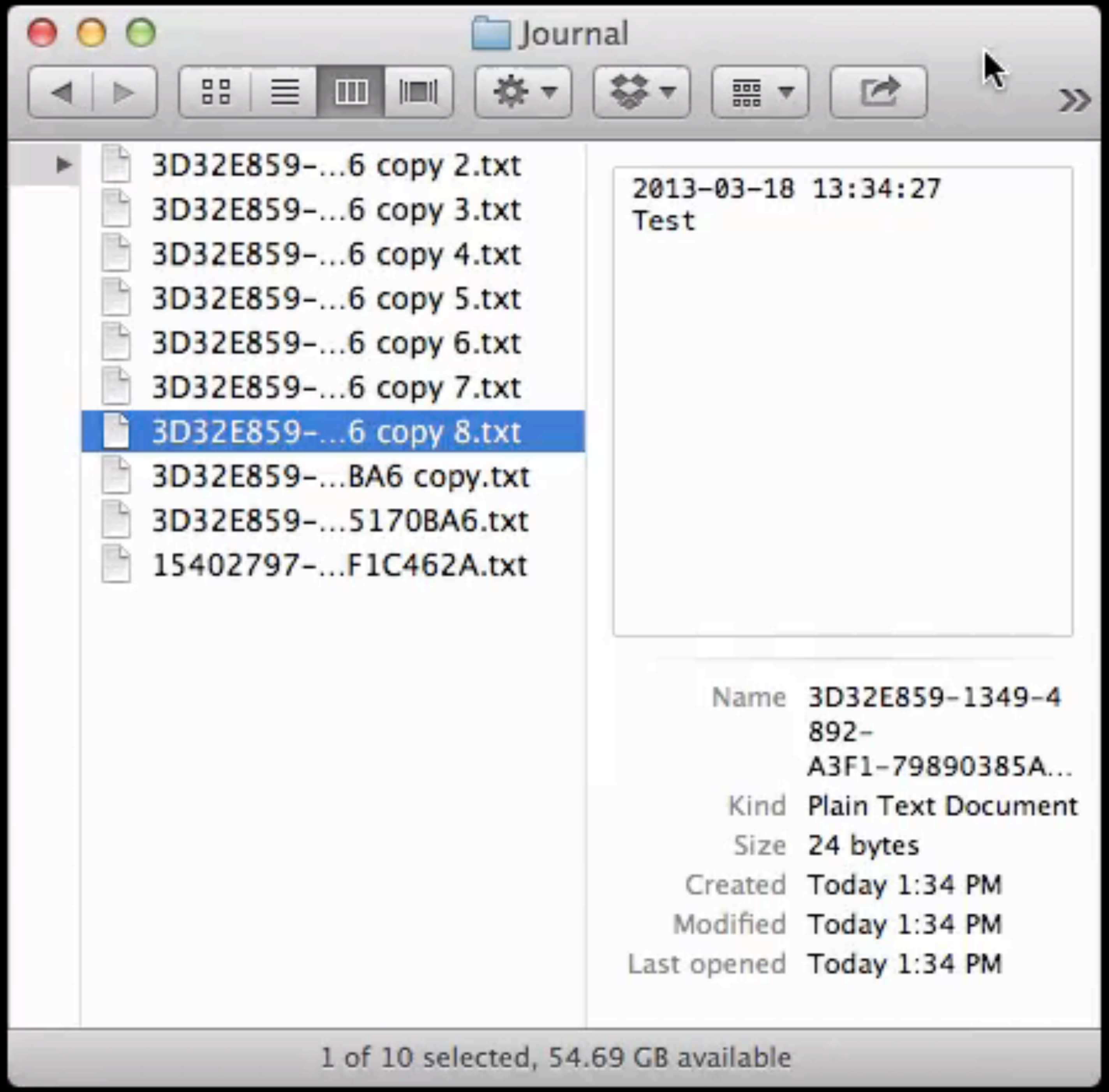
Last opened Today 1:34 PM

1 of 2 selected, 54.69 GB available



Test
3/18/13, 1:34:27 PM EDT

Another update for the afternoon.
3/18/13, 1:34:11 PM EDT

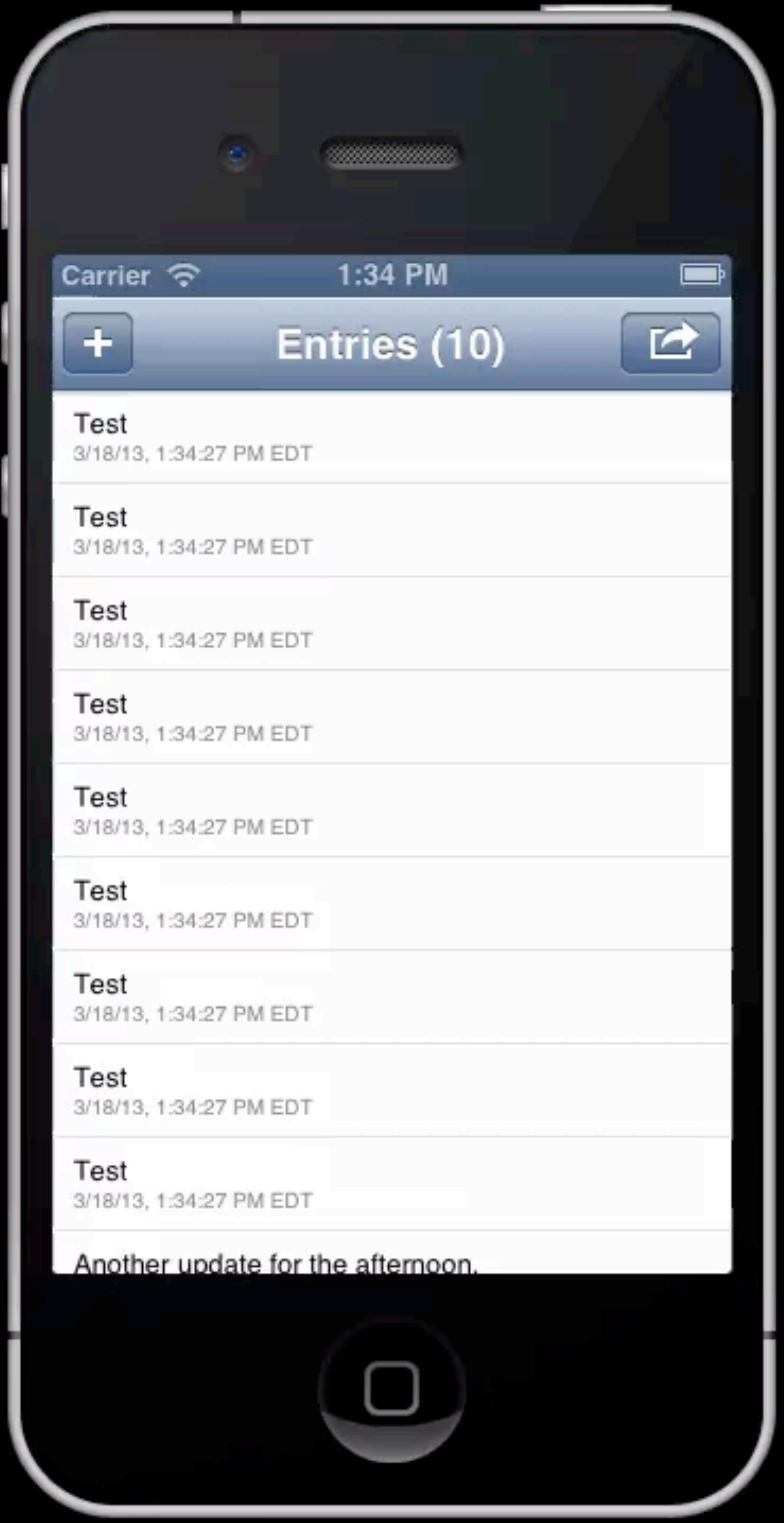


3D32E859-…6 copy 2.txt
3D32E859-…6 copy 3.txt
3D32E859-…6 copy 4.txt
3D32E859-…6 copy 5.txt
3D32E859-…6 copy 6.txt
3D32E859-…6 copy 7.txt
3D32E859-…6 copy 8.txt
3D32E859-…BA6 copy.txt
3D32E859-…5170BA6.txt
15402797-…F1C462A.txt

```
2013-03-18 13:34:27
Test
```

Name	3D32E859-1349-4892-A3F1-79890385A…
Kind	Plain Text Document
Size	24 bytes
Created	Today 1:34 PM
Modified	Today 1:34 PM
Last opened	Today 1:34 PM

1 of 10 selected, 54.69 GB available



Test
3/18/13, 1:34:27 PM EDT

Test
3/18/13, 1:34:27 PM EDT

Test
3/18/13, 1:34:27 PM EDT

Test
3/18/13, 1:34:27 PM EDT

Test
3/18/13, 1:34:27 PM EDT

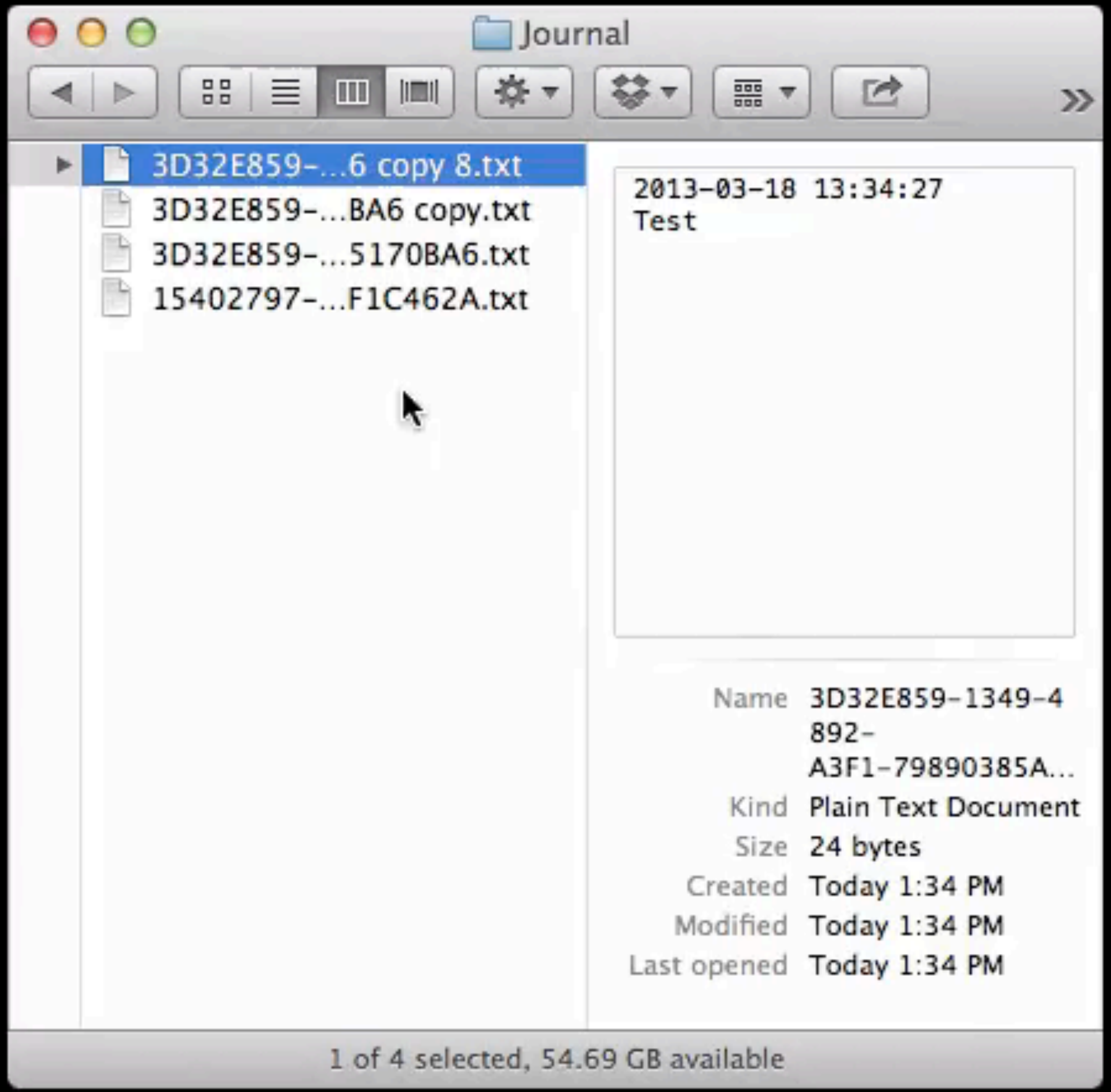
Test
3/18/13, 1:34:27 PM EDT

Test
3/18/13, 1:34:27 PM EDT

Test
3/18/13, 1:34:27 PM EDT

Test
3/18/13, 1:34:27 PM EDT

Another update for the afternoon.

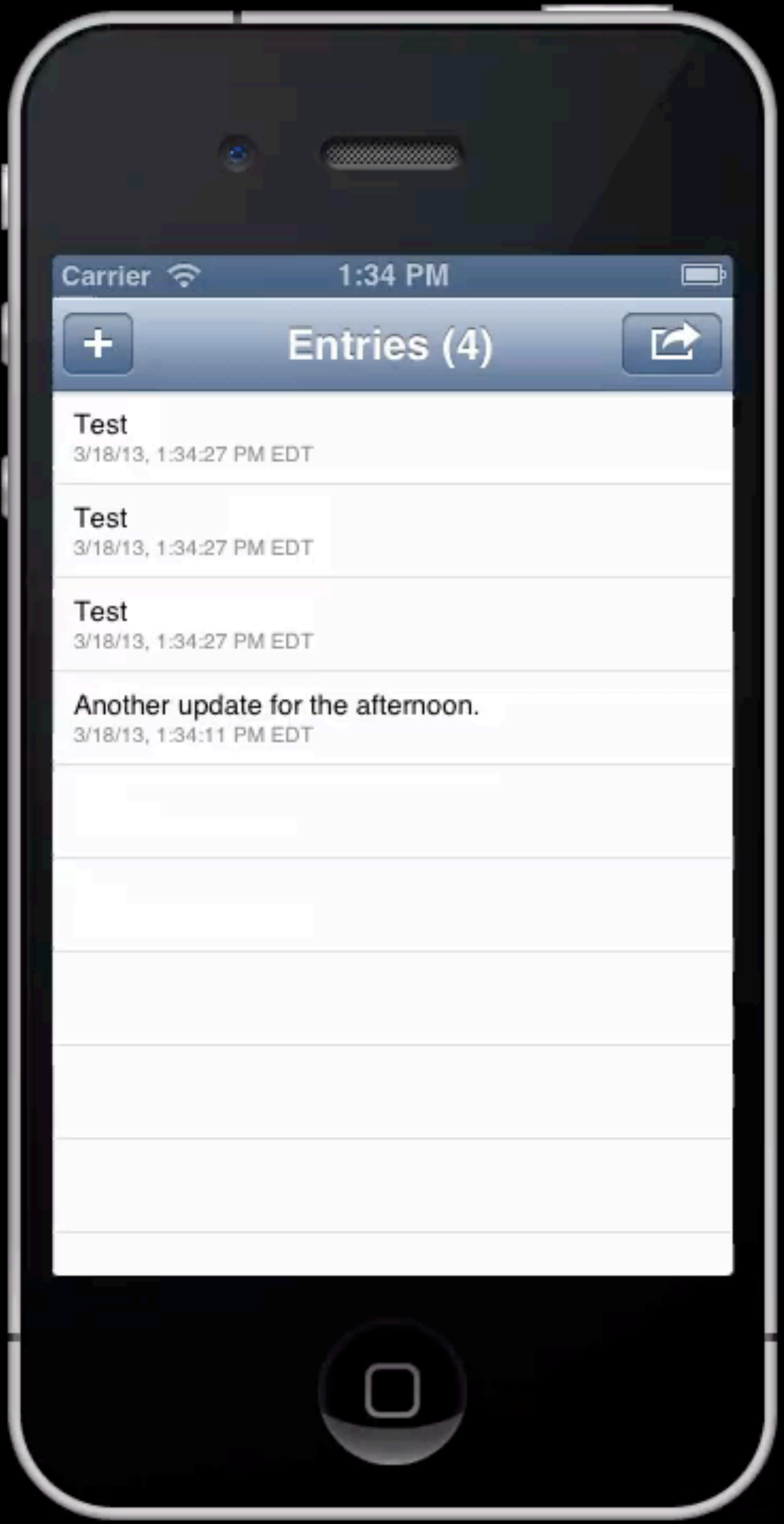


3D32E859-...6 copy 8.txt
3D32E859-...BA6 copy.txt
3D32E859-...5170BA6.txt
15402797-...F1C462A.txt

```
2013-03-18 13:34:27
Test
```

Name	3D32E859-1349-4892-A3F1-79890385A...
Kind	Plain Text Document
Size	24 bytes
Created	Today 1:34 PM
Modified	Today 1:34 PM
Last opened	Today 1:34 PM

1 of 4 selected, 54.69 GB available

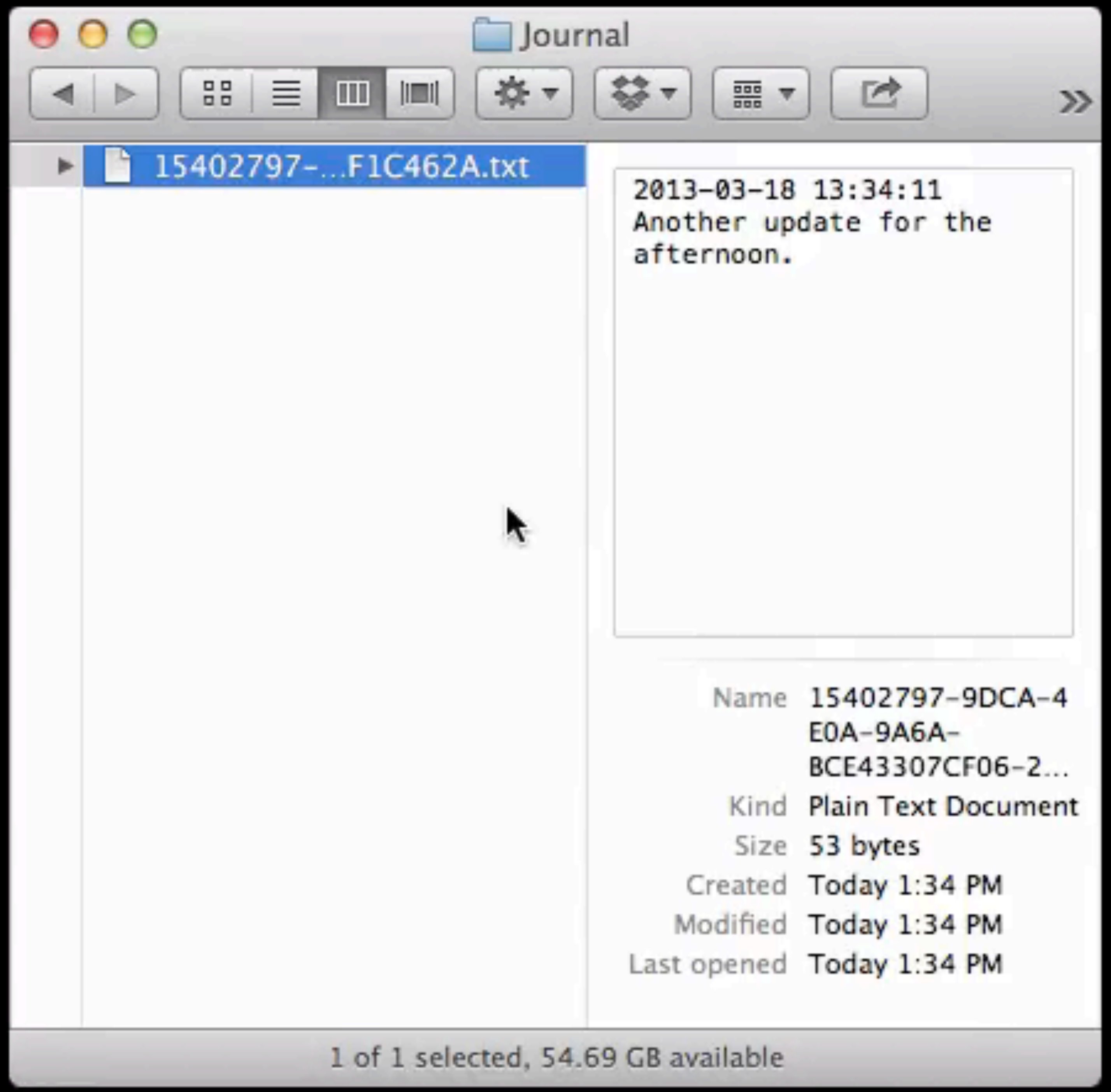


Test
3/18/13, 1:34:27 PM EDT

Test
3/18/13, 1:34:27 PM EDT

Test
3/18/13, 1:34:27 PM EDT

Another update for the afternoon.
3/18/13, 1:34:11 PM EDT



```
2013-03-18 13:34:11
Another update for the
afternoon.
```

Name 15402797-9DCA-4E0A-9A6A-BCE43307CF06-2...

Kind Plain Text Document

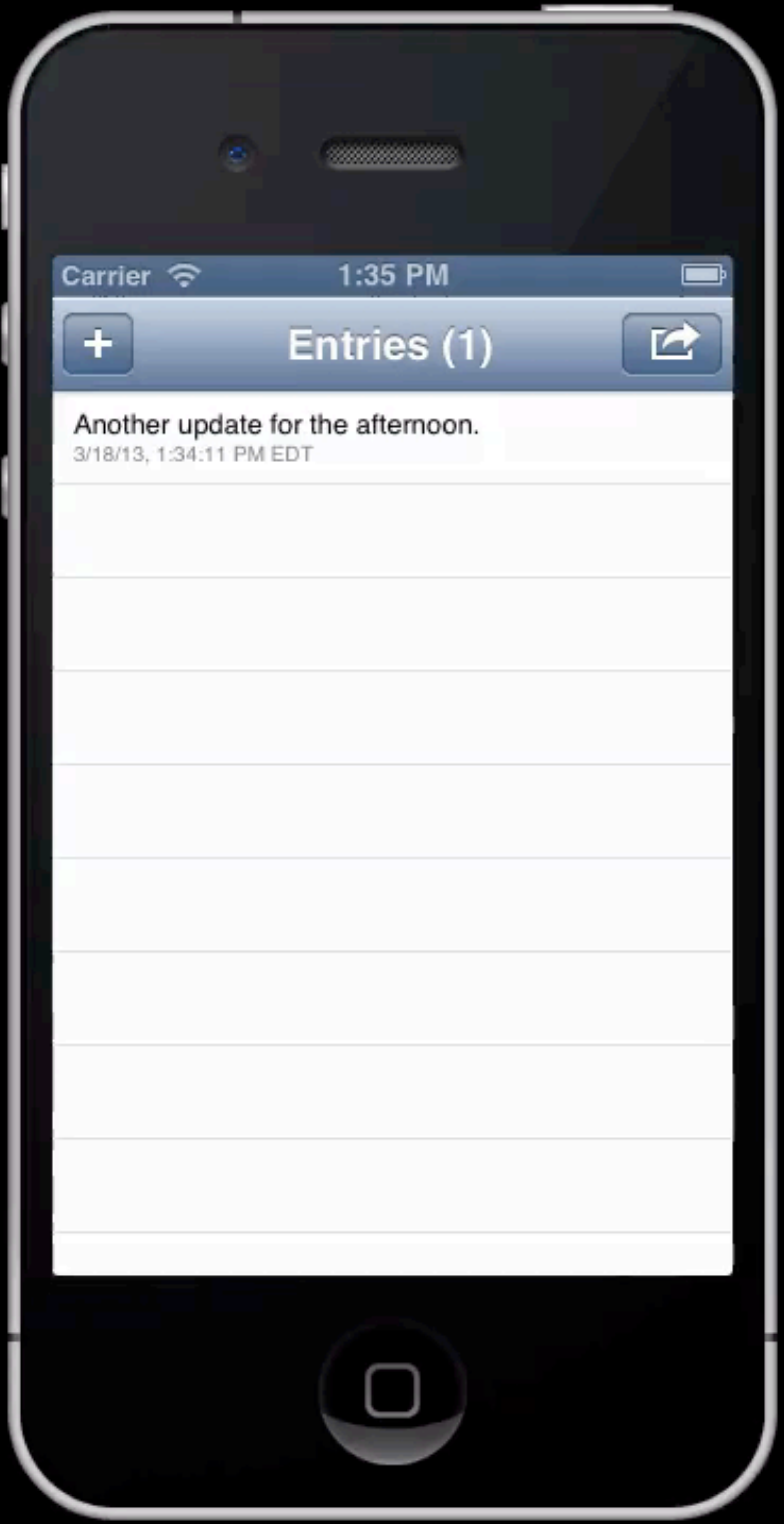
Size 53 bytes

Created Today 1:34 PM

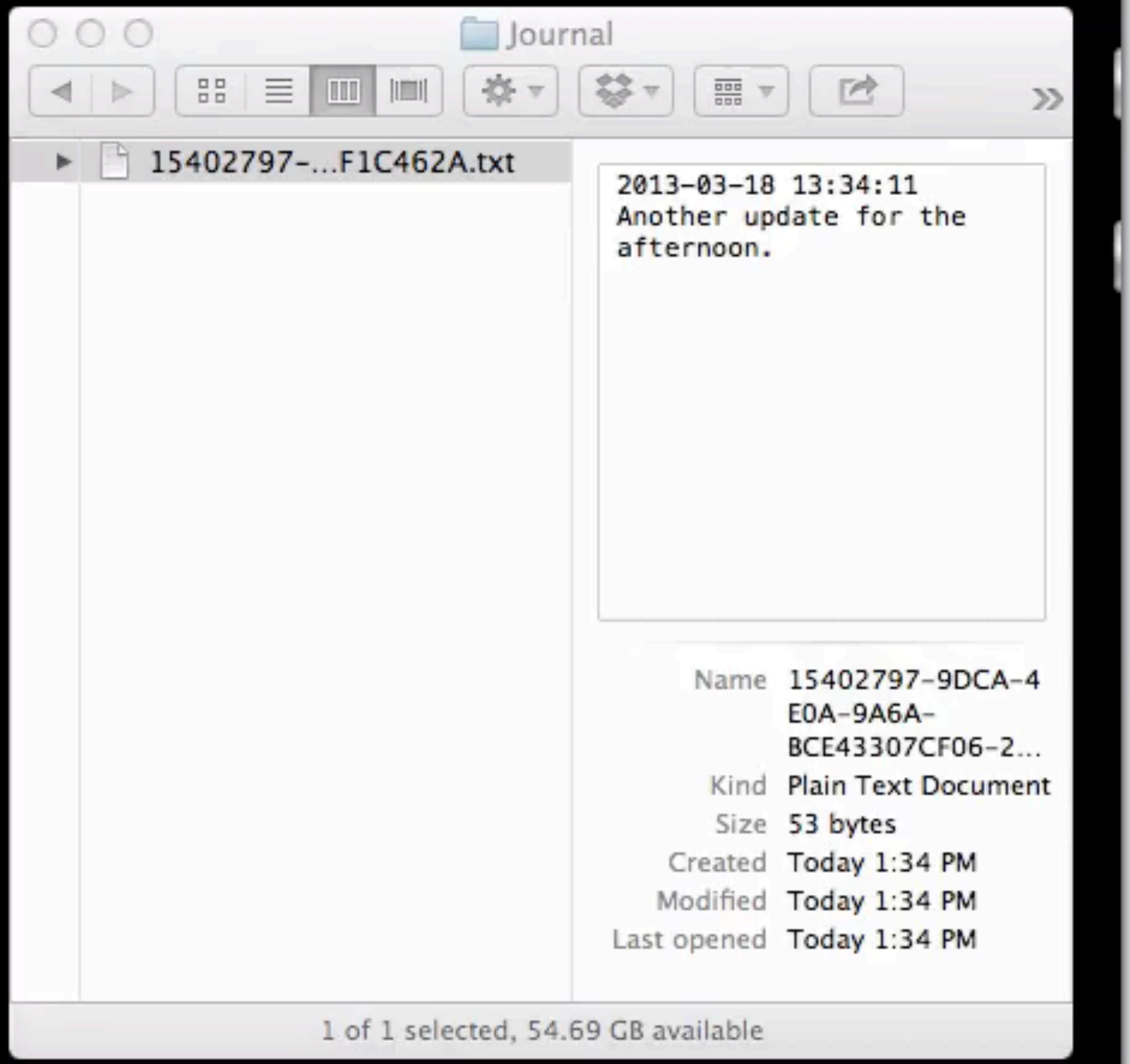
Modified Today 1:34 PM

Last opened Today 1:34 PM

1 of 1 selected, 54.69 GB available



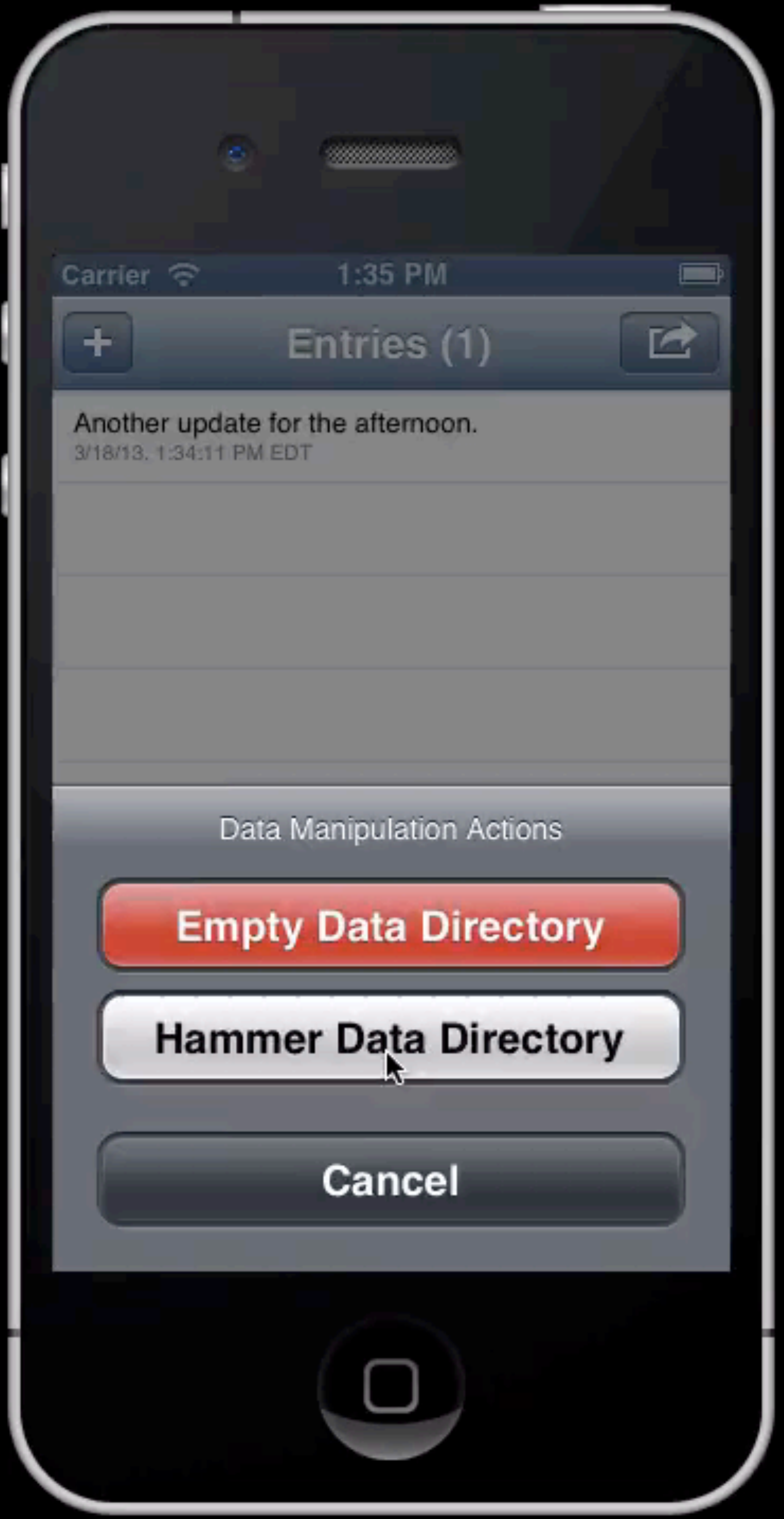
Another update for the afternoon.
3/18/13, 1:34:11 PM EDT



```
2013-03-18 13:34:11
Another update for the
afternoon.
```

Name	15402797-9DCA-4E0A-9A6A-BCE43307CF06-2...
Kind	Plain Text Document
Size	53 bytes
Created	Today 1:34 PM
Modified	Today 1:34 PM
Last opened	Today 1:34 PM

1 of 1 selected, 54.69 GB available



Another update for the afternoon.
3/18/13, 1:34:11 PM EDT

Resources

Motion Data

`github.com/alloy/MotionData`

Superbox

`github.com/awdogsgo2heaven/superbox`

Resources

Apple’s Core Data Docs

`developer.apple.com/Library/ios/documentation/Cocoa/Conceptual/CoreData/cdProgrammingGuide.html`

Marcus Zarra’s Core Data Book

`pragprog.com/book/mzcd2/core-data`

Josh Smith’s Core Data Book (coming later this year)

Follow for more info: `twitter.com/kognate`

My slides & sample code

`cocoamanifest.net/features`

CORE DATA

Win Condition Acquired!

Thanks!

(and please feedback)

@jonathanpenn

cocoamanifest.net