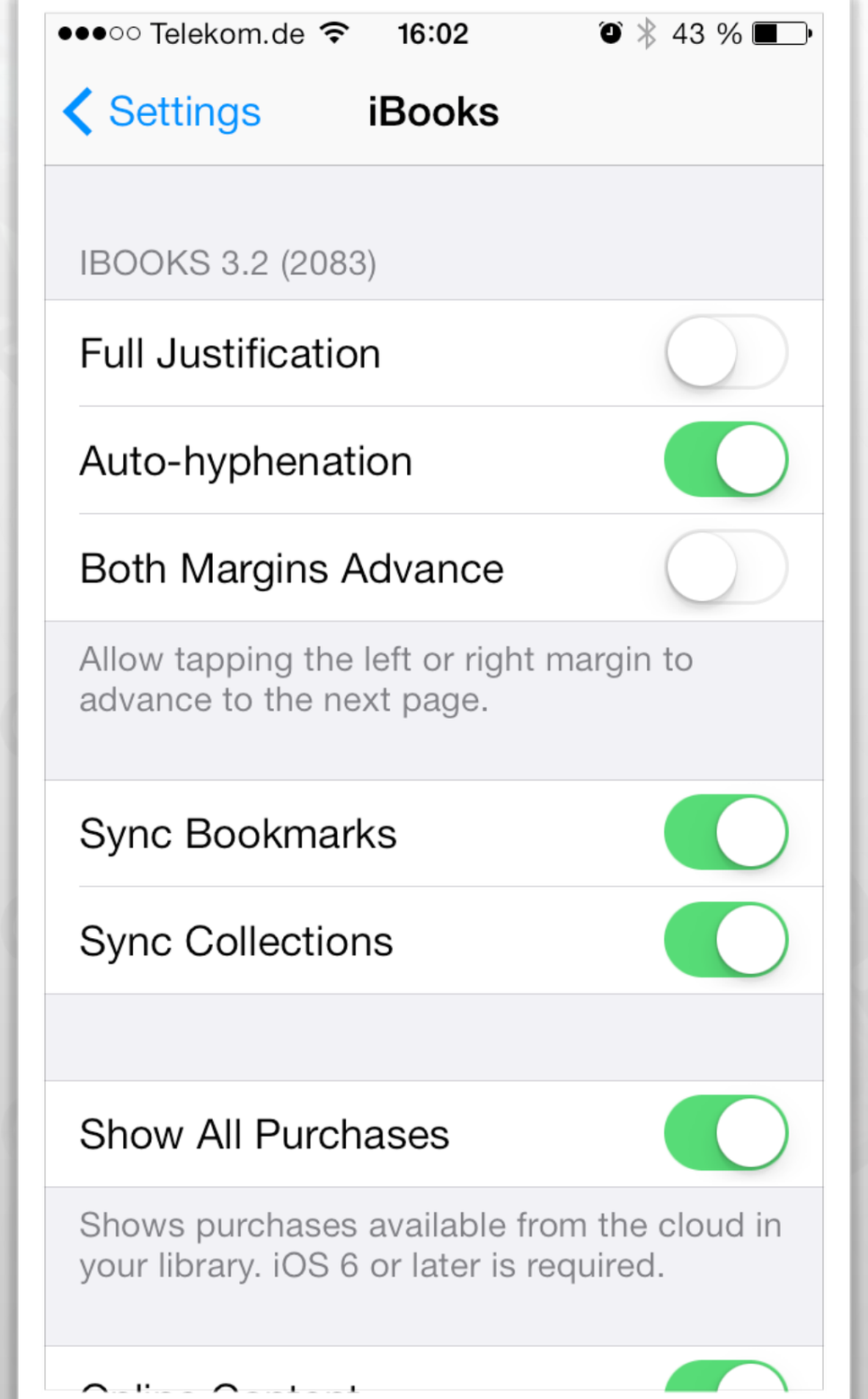
iPhone Application Programming
L09: Data Persistence

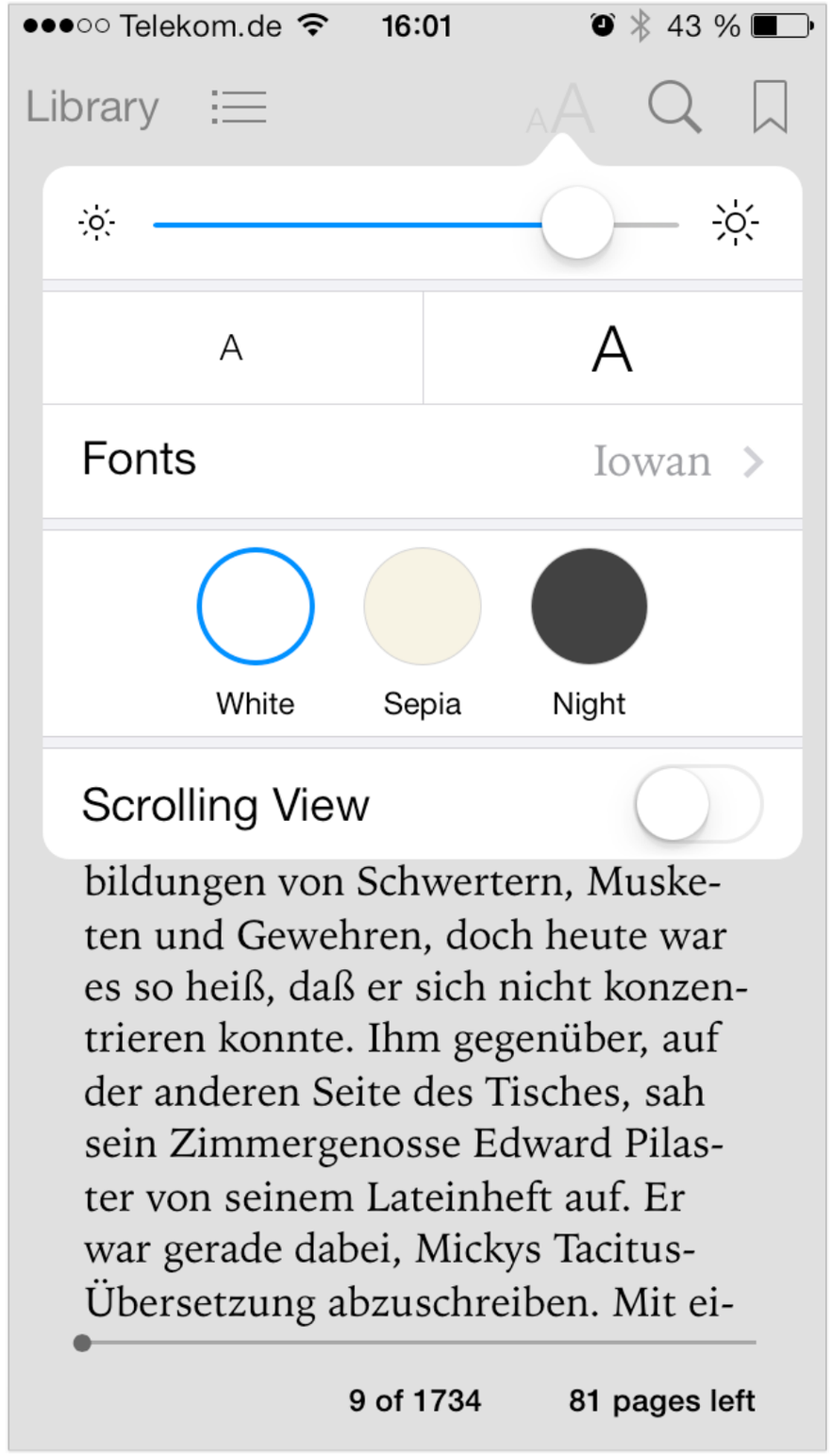
Jan-Peter Krämer
Media Computing Group, RWTH Aachen

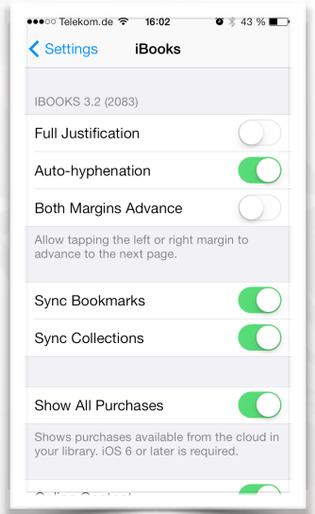
WS 2015/2016
http://hci.rwth-aachen.de/iPhone

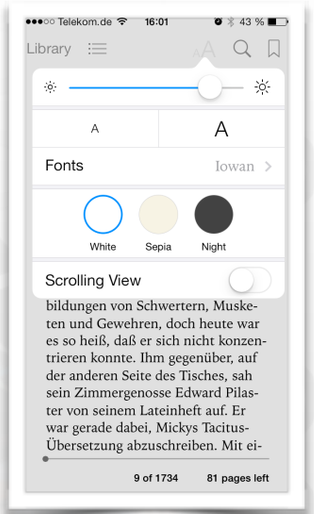
What data do we have on devices?

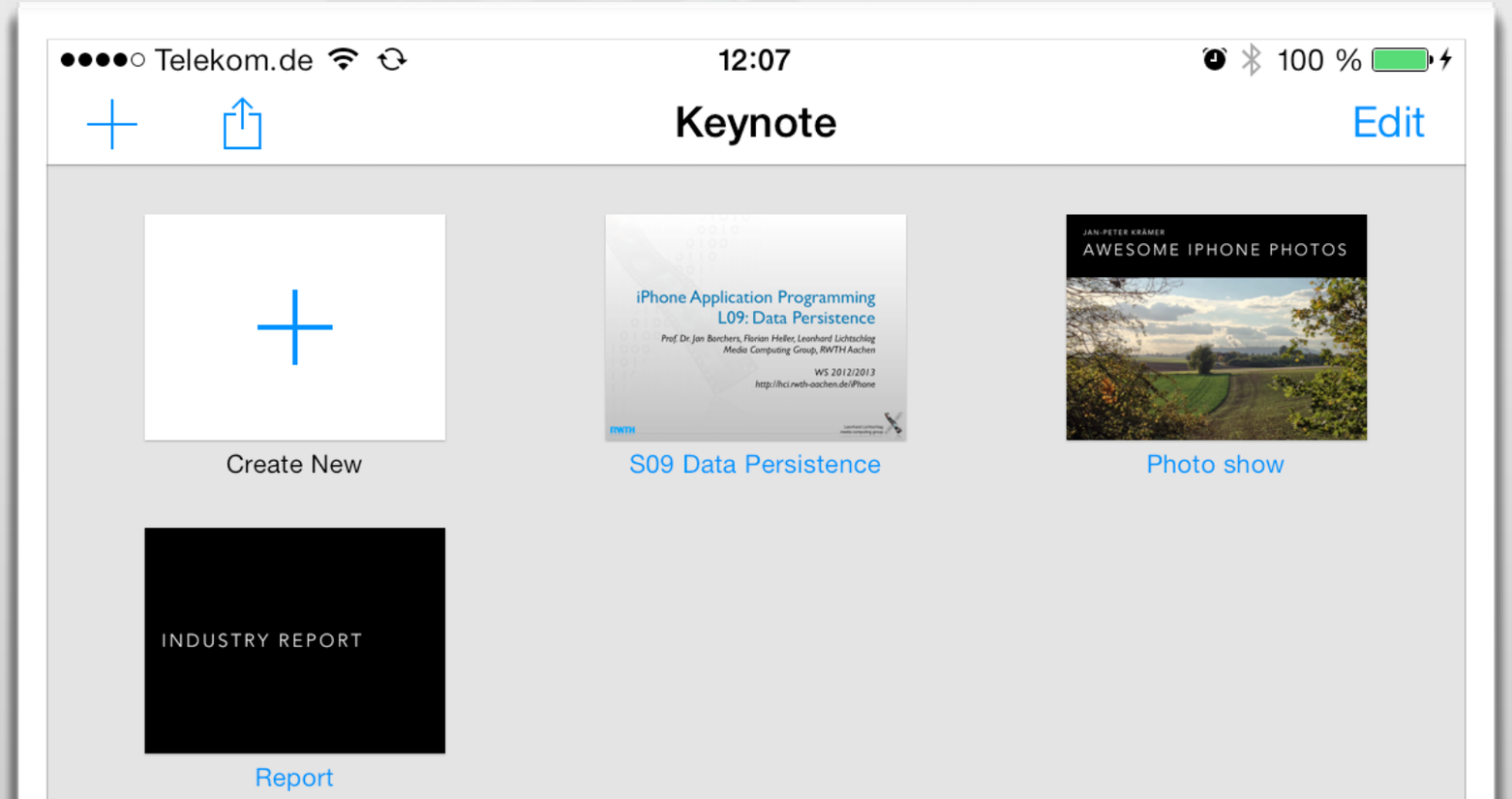
How do we present data to the user?

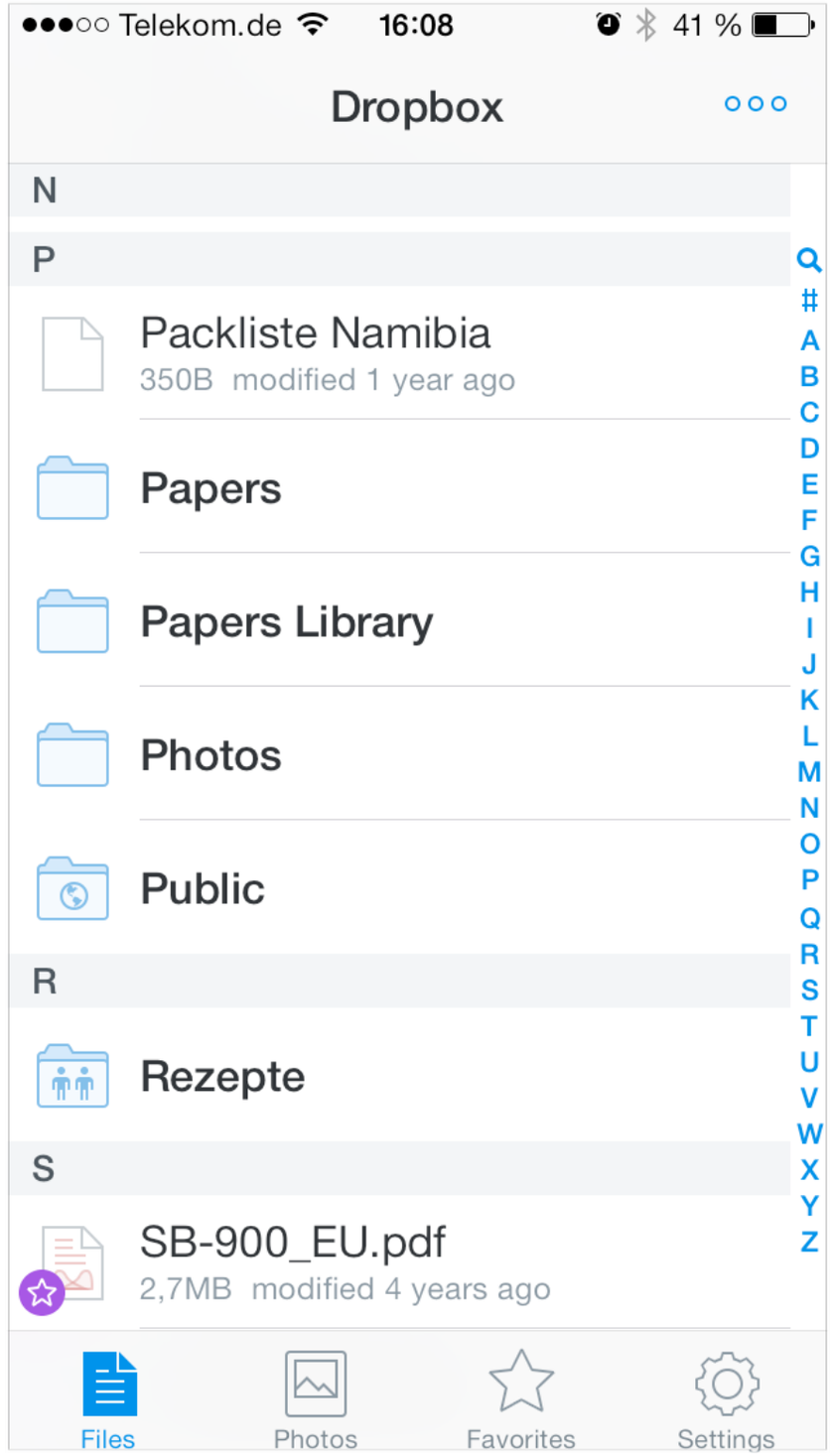

Telekom.de 16:02 43 %
Settings
iBooks
IBOOKS 3.2 (2083)
Full Justification
Auto-hyphenation
Both Margins Advance
Allow tapping the left or right margin to advance to the next page.
Sync Bookmarks
Sync Collections
Show All Purchases
Shows purchases available from the cloud in your library. iOS 6 or later is required.

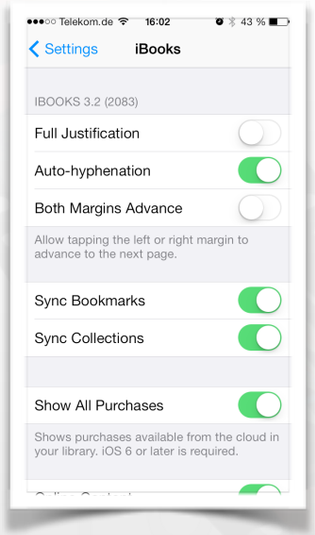

Telekom.de 16:01 43 %
Library
A
A
Fonts
Iowan
White
Sepia
Night
Scrolling View
bildungen von Schwertern, Musketen und Gewehren, doch heute war es so heiß, daß er sich nicht konzentrieren konnte. Ihm gegenüber, auf der anderen Seite des Tisches, sah sein Zimmergenosse Edward Pilaster von seinem Lateinheft auf. Er war gerade dabei, Mickys Tacitus-Übersetzung abzuschreiben. Mit ei-
9 of 1734
81 pages left

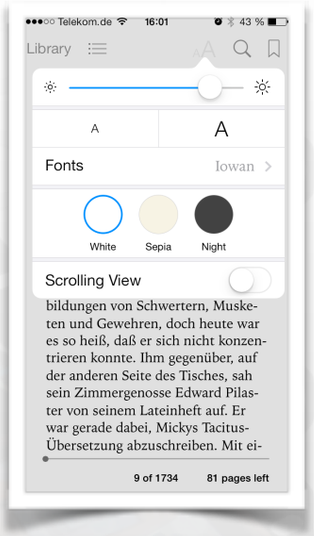

Telekom.de 16:02 43 %
Settings
iBooks
IBOOKS 3.2 (2083)
Full Justification
Auto-hyphenation
Both Margins Advance
Allow tapping the left or right margin to advance to the next page.
Sync Bookmarks
Sync Collections
Show All Purchases
Shows purchases available from the cloud in your library. iOS 6 or later is required.

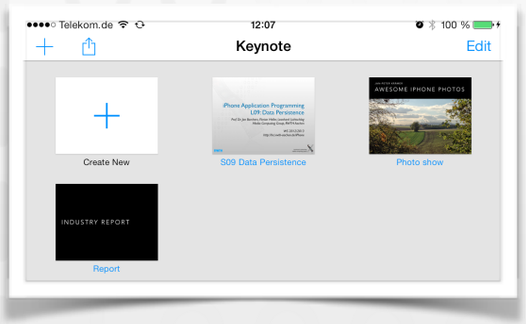

Telekom.de 16:01 43 %
Library
A
A
Fonts
Iowan
White
Sepia
Night
Scrolling View
bildungen von Schwertern, Musketen und Gewehren, doch heute war es so heiß, daß er sich nicht konzentrieren konnte. Ihm gegenüber, auf der anderen Seite des Tisches, sah sein Zimmergenosse Edward Pilaster von seinem Lateinheft auf. Er war gerade dabei, Mickys Tacitus-Übersetzung abzuschreiben. Mit ei-
9 of 1734
81 pages left

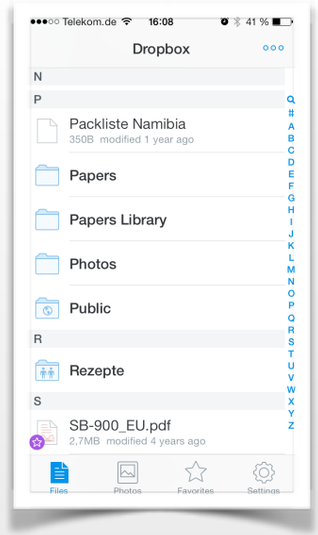

Telekom.de 12:07 100 %
Keynote
Edit
Create New
iPhone Application Programming
L09: Data Persistence
Prof. Dr. Jan Borchers, Florian Heller, Leonhard Lichtschlag
Media Computing Group, RWTH Aachen
WS 2012/2013
http://hci.rwth-aachen.de/iPhone
S09 Data Persistence
JAN-PETER KRÄMER
AWESOME IPHONE PHOTOS
Photo show
INDUSTRY REPORT
Report

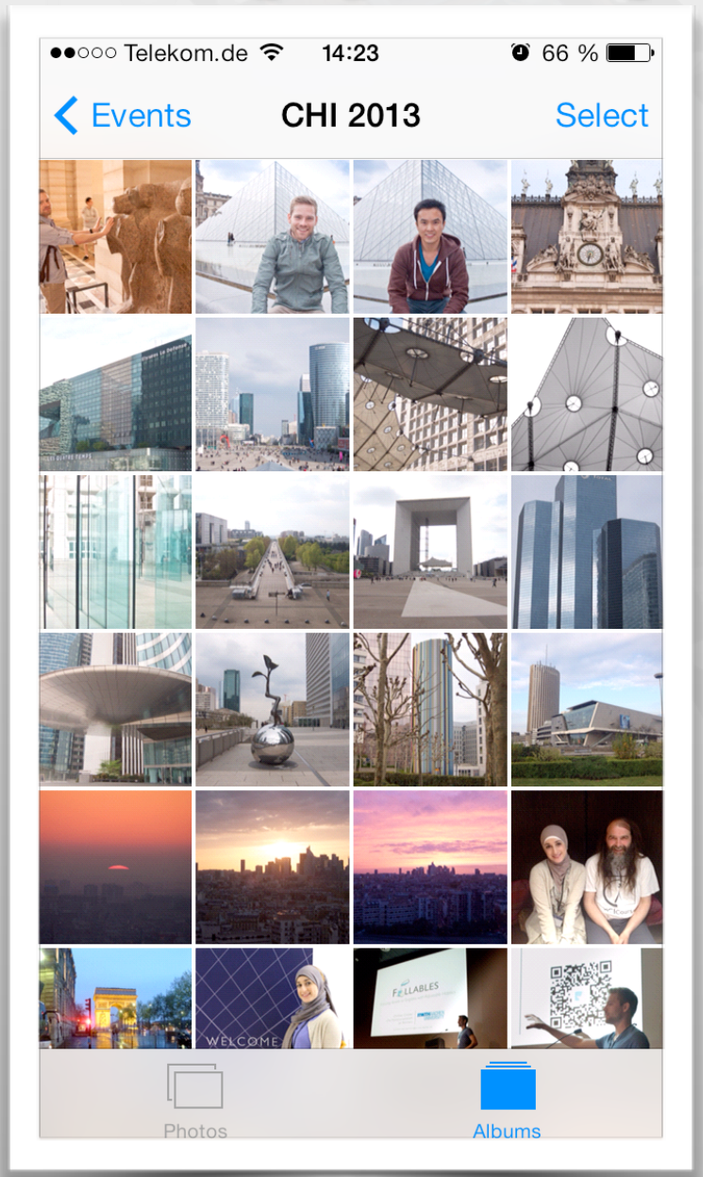

Telekom.de 16:08 41 %
Dropbox
N
P
Packliste Namibia
350B modified 1 year ago
Papers
Papers Library
Photos
Public
R
Rezepte
S
SB-900_EU.pdf
2,7MB modified 4 years ago
Files
Photos
Favorites
Settings
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

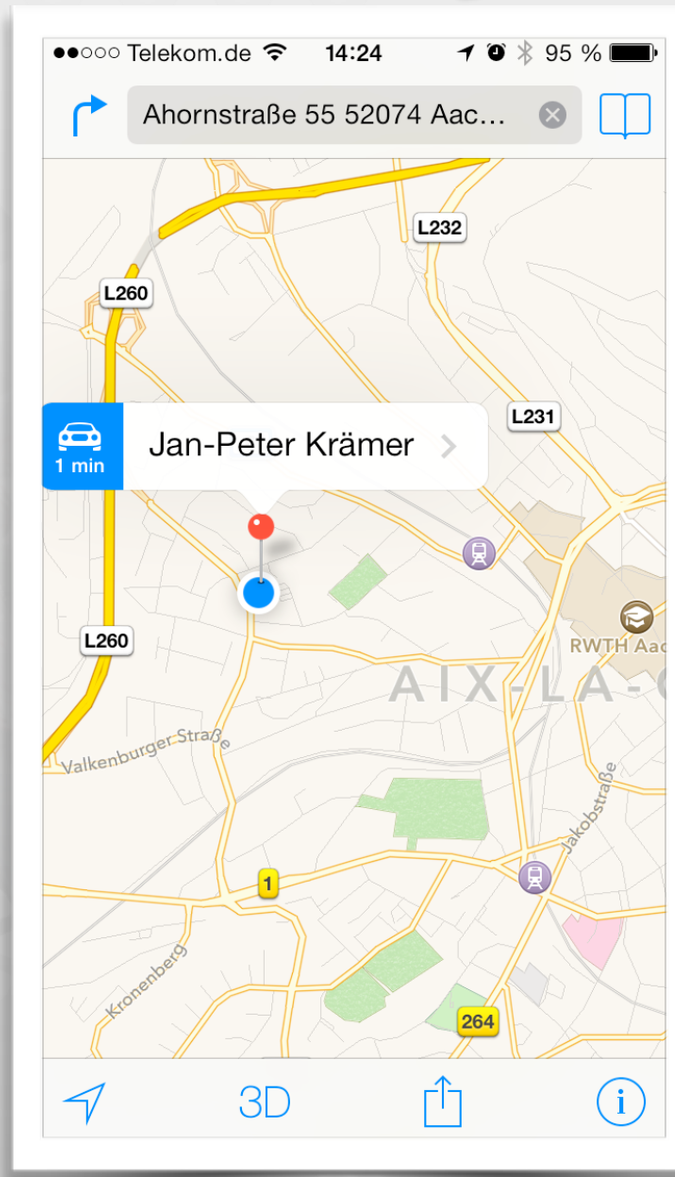

Telekom.de 16:02 43 %
Settings
iBooks
IBOOKS 3.2 (2083)
Full Justification
Auto-hyphenation
Both Margins Advance
Allow tapping the left or right margin to advance to the next page.
Sync Bookmarks
Sync Collections
Show All Purchases
Shows purchases available from the cloud in your library. iOS 6 or later is required.

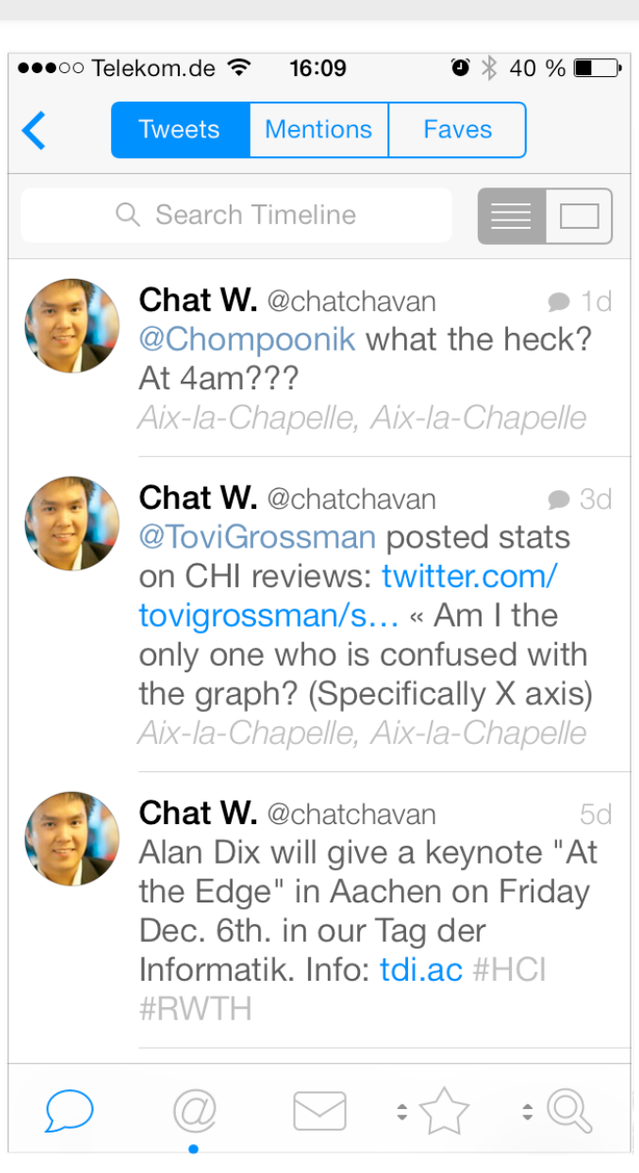

Telekom.de 16:01 43 %
Library
A
A
Fonts
Iowan
White
Sepia
Night
Scrolling View
bildungen von Schwertern, Musketen und Gewehren, doch heute war es so heiß, daß er sich nicht konzentrieren konnte. Ihm gegenüber, auf der anderen Seite des Tisches, sah sein Zimmergenosse Edward Pilaster von seinem Lateinheft auf. Er war gerade dabei, Mickys Tacitus-Übersetzung abzuschreiben. Mit ei-
9 of 1734
81 pages left


Telekom.de 12:07 100 %
Keynote
Edit
Create New
S09 Data Persistence
Photo show
INDUSTRY REPORT
Report

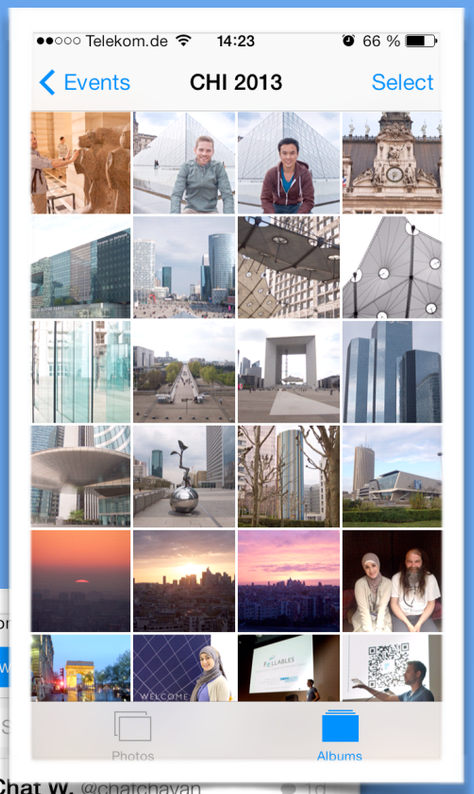

Telekom.de 16:08 41 %
Dropbox
Packliste Namibia
350B modified 1 year ago
Papers
Papers Library
Photos
Public
Rezepte
SB-900_EU.pdf
2,7MB modified 4 years ago
Files
Photos
Favorites
Settings


Telekom.de 14:23 66 %
Events
CHI 2013
Select
Photos
Albums

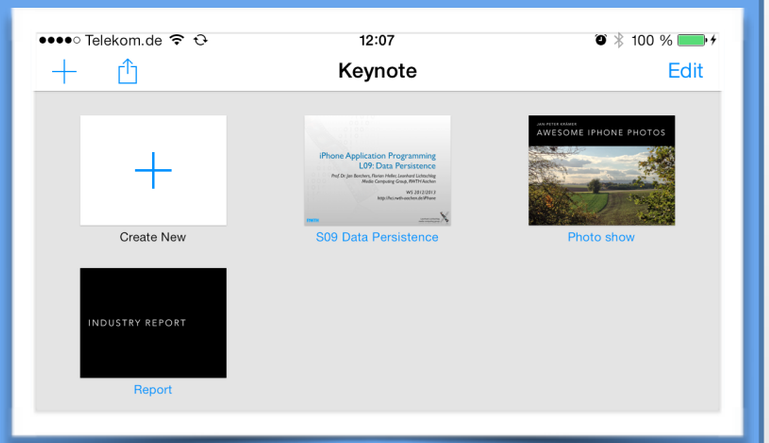

Telekom.de 14:24 95 %
Ahornstraße 55 52074 Aac…
L232
L260
L231
1 min
Jan-Peter Krämer
RWTH Aac
AIX-LA-C
Valkenburger Straße
Jakobstraße
Kronenberg
1
264
3D

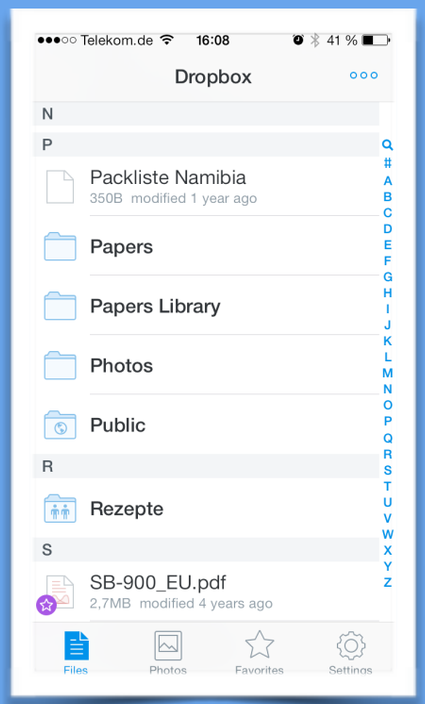

Telekom.de 16:09 40 %
Tweets
Mentions
Faves
Search Timeline
Chat W. @chatchavan 1d
@Chompoonik what the heck? At 4am???
Aix-la-Chapelle, Aix-la-Chapelle
Chat W. @chatchavan 3d
@ToviGrossman posted stats on CHI reviews: twitter.com/tovigrossman/s… « Am I the only one who is confused with the graph? (Specifically X axis)
Aix-la-Chapelle, Aix-la-Chapelle
Chat W. @chatchavan 5d
Alan Dix will give a keynote "At the Edge" in Aachen on Friday Dec. 6th. in our Tag der Informatik. Info: tdi.ac #HCI #RWTH

“Shoebox”

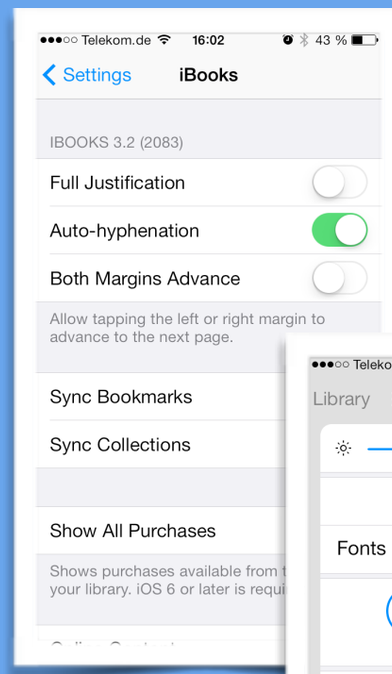


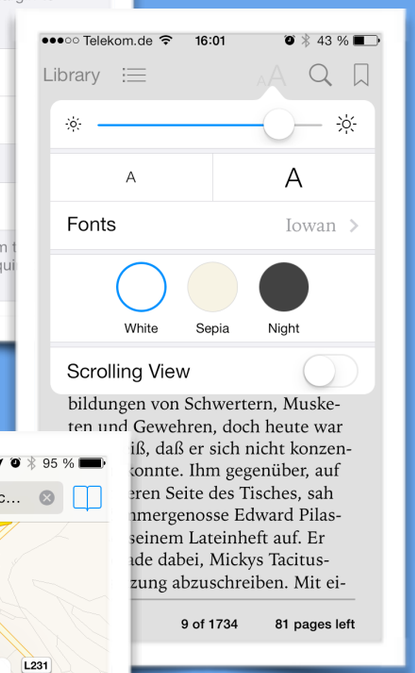
“Documents”

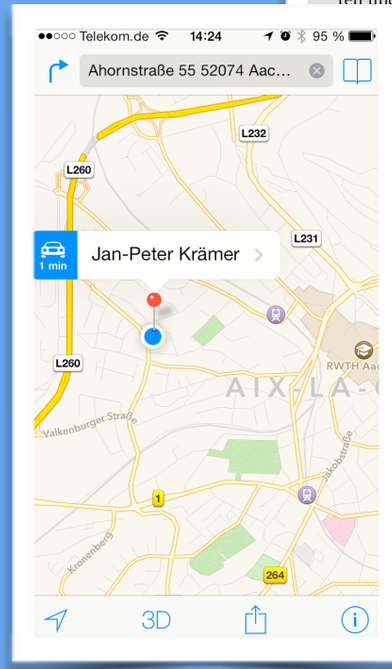




“Settings”







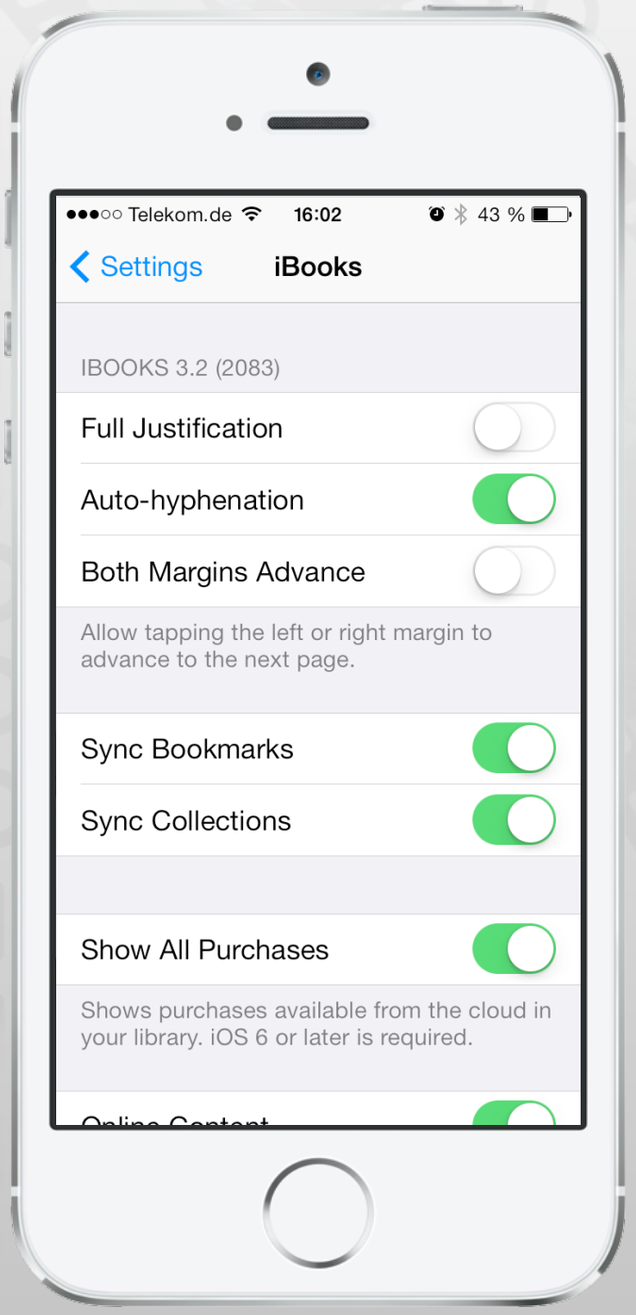
Data Handling Overview

	"Shoebox"	Document	App Data "Settings"
Local	Database	UIDocument	NSUserDefaults
Synced	Database	UIDocument with iCloud	iCloud Key/Value GameCenter

CoreData

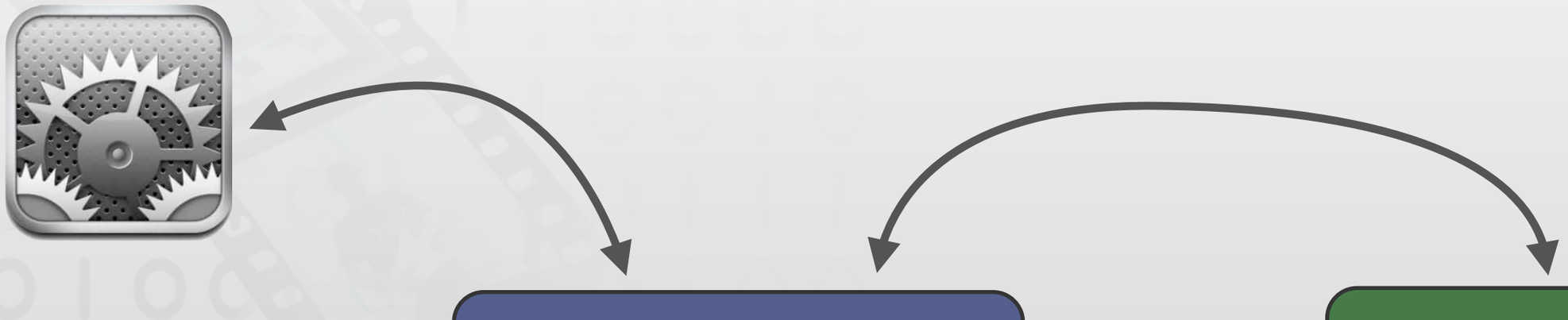
Online — Communication with a web server

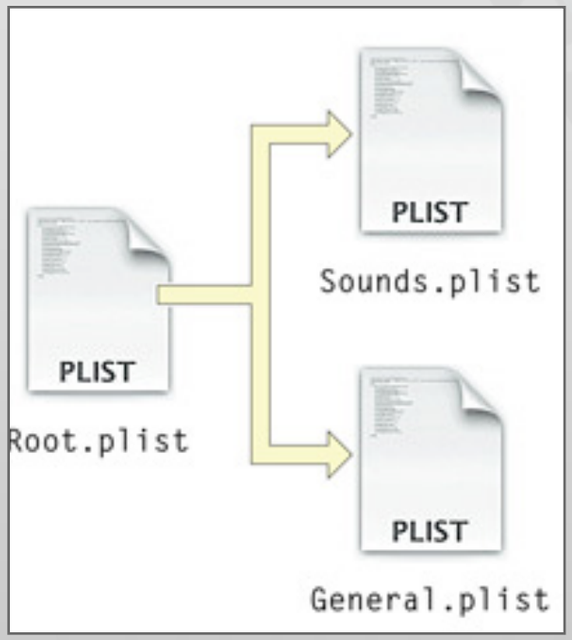
Preferences and Settings



- NSUserDefaults
  - Singleton
  - Key/value pairs
  - Provides standard values (factory defaults)
- Settings Bundle
  - Describes preferences managed in settings
  - Same keys as in NSUserDefaults

Preferences and Settings





NSUserDefaults
key/value pairs

Application

Demo

Data Handling Overview

	"Shoebox"	Document	App Data "Settings"
Local	Database	UIDocument	NSUserDefaults
Synced	Database	UIDocument with iCloud	iCloud Key/Value GameCenter
Online	Communication with a web server		

Where to put files?

```
func localDocumentDirectoryURL() -> NSURL {
  let documentsDirectoryPath =
NSSearchPathForDirectoriesInDomains(.DocumentDirectory, .UserDomainM
ask, true)[0]

  return NSURL(fileURLWithPath: documentsDirectoryPath)
}
```

UIDocument

- High level API for file access
- Always represents a user document
- Can handle file packages
- Autosaving
- Undo
- Handles remote changes from iCloud
- UIManagedDocument for CoreData

Creating a new document

```
self.document = UIDocument(fileURL: fileURL)
   self.document.saveToURL(self.document.fileURL,
forSaveOperation: .ForCreating) { (success) -> Void in
      if (success == true) {
         self.displayFileContents()
      }
   }
```

```
override init(fileURL url: NSURL) {
   self.textContents = "Hello World"
   super.init(fileURL: url)
}
```

Saving a UIDocument

- gets saved when file is closed *or*
- autosaved when changes are pending
  - NSUndoManager
  - or call updateChangeCount:

```
// Return NSData representation for saving
- (id) contentsForType:(NSString *)typeName
                 error:(NSError **)outError
{
  NSData *data = [self.contents dataUsingEncoding:
                                NSUTF8StringEncoding];
  return data;
}
```


Jan-Peter Krämer
media computing group

Data Handling Overview

	"Shoebox"	Document	App Data "Settings"
Local	Database	UIDocument	NSUserDefaults
Synced	Database	UIDocument with iCloud	iCloud Key/Value GameCenter
Online	Communication with a web server		

SQLite

Data Handling Overview

	"Shoebox"	Document	App Data "Settings"
Local	Database	UIDocument	NSUserDefaults
Synced	Database	UIDocument with iCloud	iCloud Key/Value GameCenter

CoreData

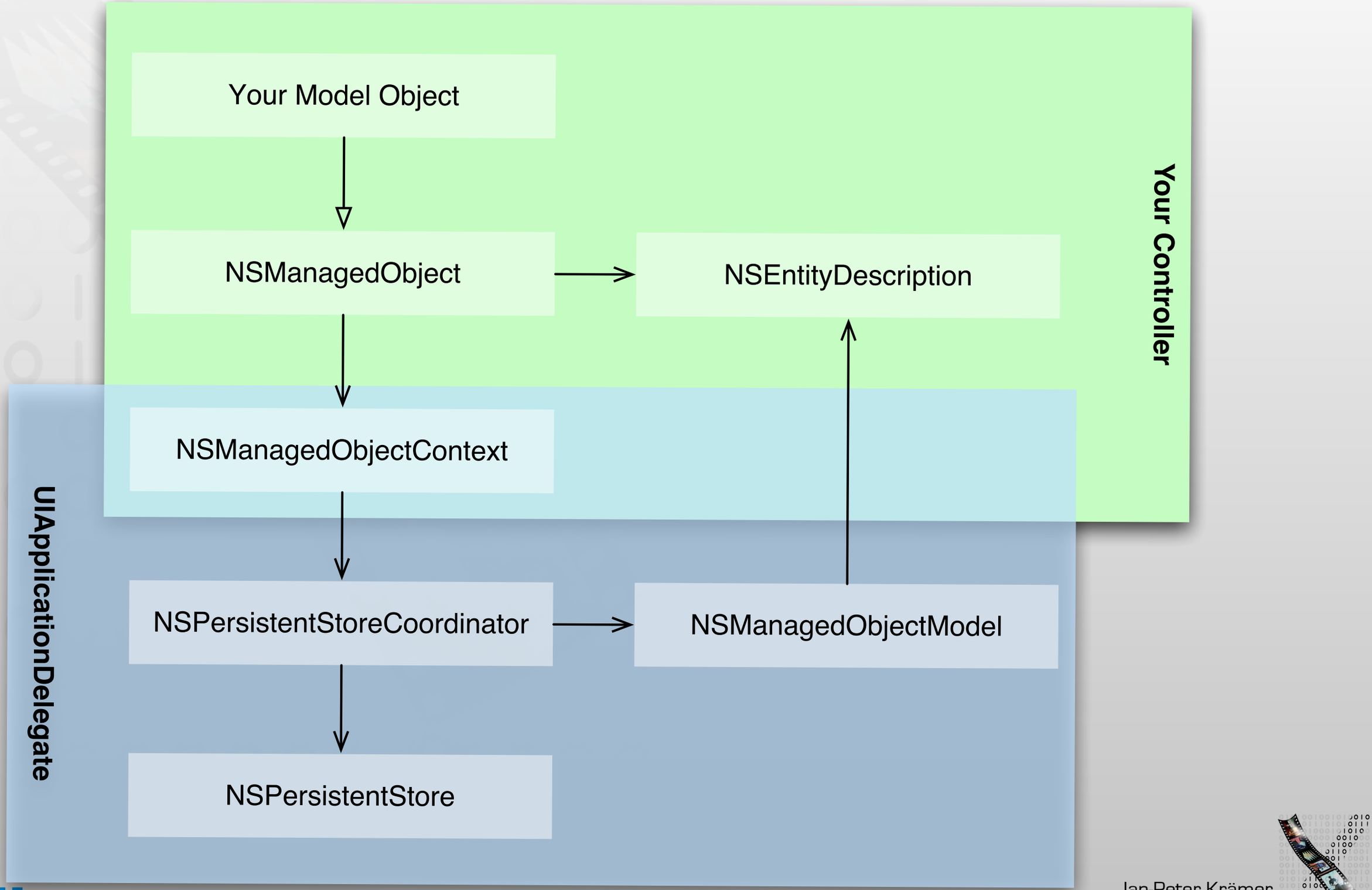
Online — Communication with a web server

Core Data

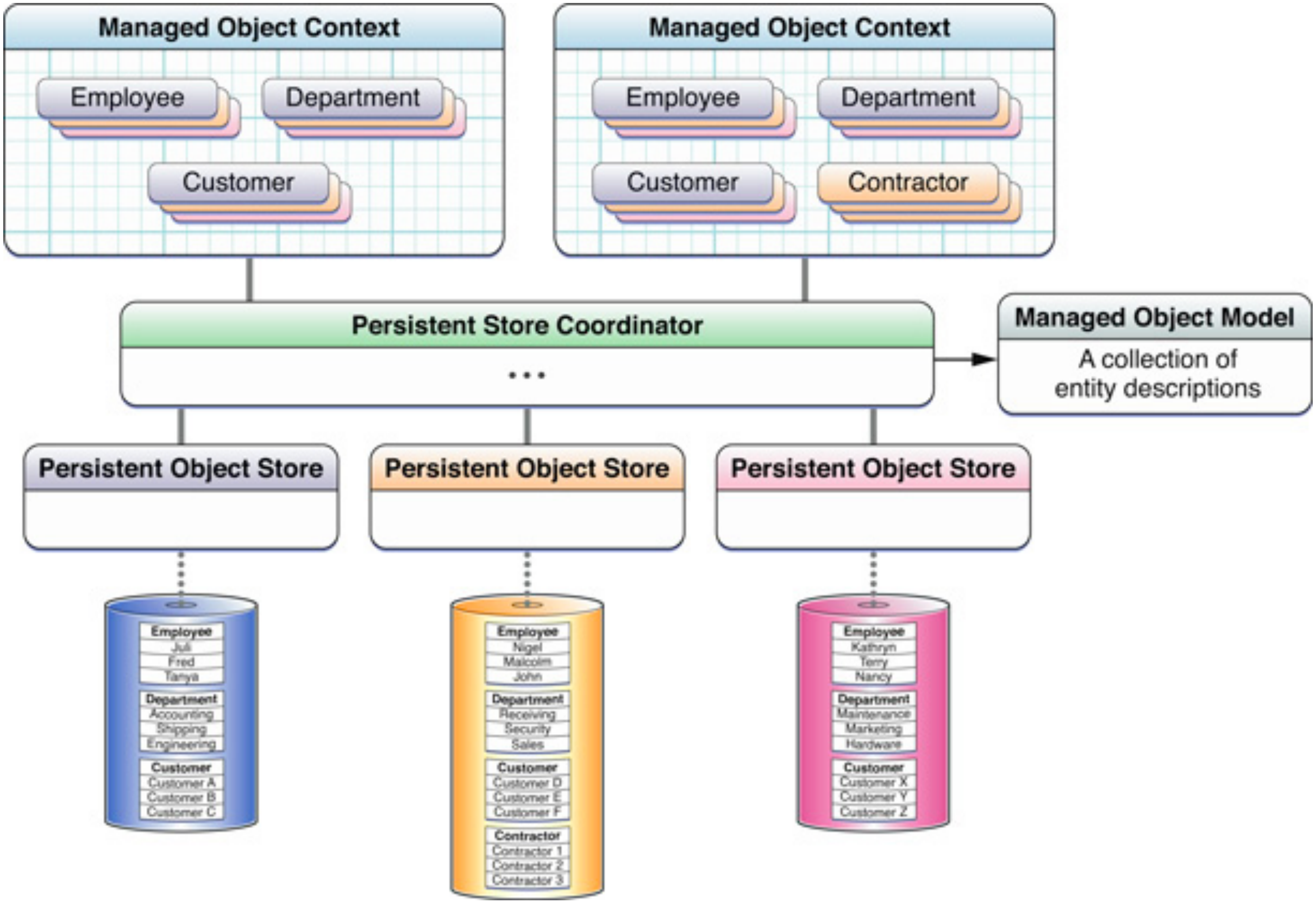
Core Data

- High level relational data storage
- Automatic persistence
- High performance access
- Get undo/redo for free
- Available data stores
  - Memory
  - Binary file
  - SQLite

Core Data Stack



Jan-Peter Krämer
media computing group


Managed Object Context
Employee
Department
Customer
Managed Object Context
Employee
Department
Customer
Contractor
Persistent Store Coordinator
...
Managed Object Model
A collection of entity descriptions
Persistent Object Store
Persistent Object Store
Persistent Object Store
Employee
Juli
Fred
Tanya
Department
Accounting
Shipping
Engineering
Customer
Customer A
Customer B
Customer C
Employee
Nigel
Malcolm
John
Department
Receiving
Security
Sales
Customer
Customer D
Customer E
Customer F
Contractor
Contractor 1
Contractor 2
Contractor 3
Employee
Kathryn
Terry
Nancy
Department
Maintenance
Marketing
Hardware
Customer
Customer X
Customer Y
Customer Z

Recipes.xcdatamodel

Recipes.xcdatamodel ▸ Recipe

Entity	Abs	Class
Image	☐	NSManage
Ingredient	☐	Ingredient
Recipe	☐	Recipe
RecipeType	☐	NSManage

Property	Kind	Type or
image	Relationship	Image
ingredients	Relationship	Ingredient
instructions	Attribute	String
name	Attribute	String
overview	Attribute	String
prepTime	Attribute	String
thumbnailImage	Attribute	Transformable
type	Relationship	RecipeType

Relationship

Name: image

☑ Optional ☐ Transient

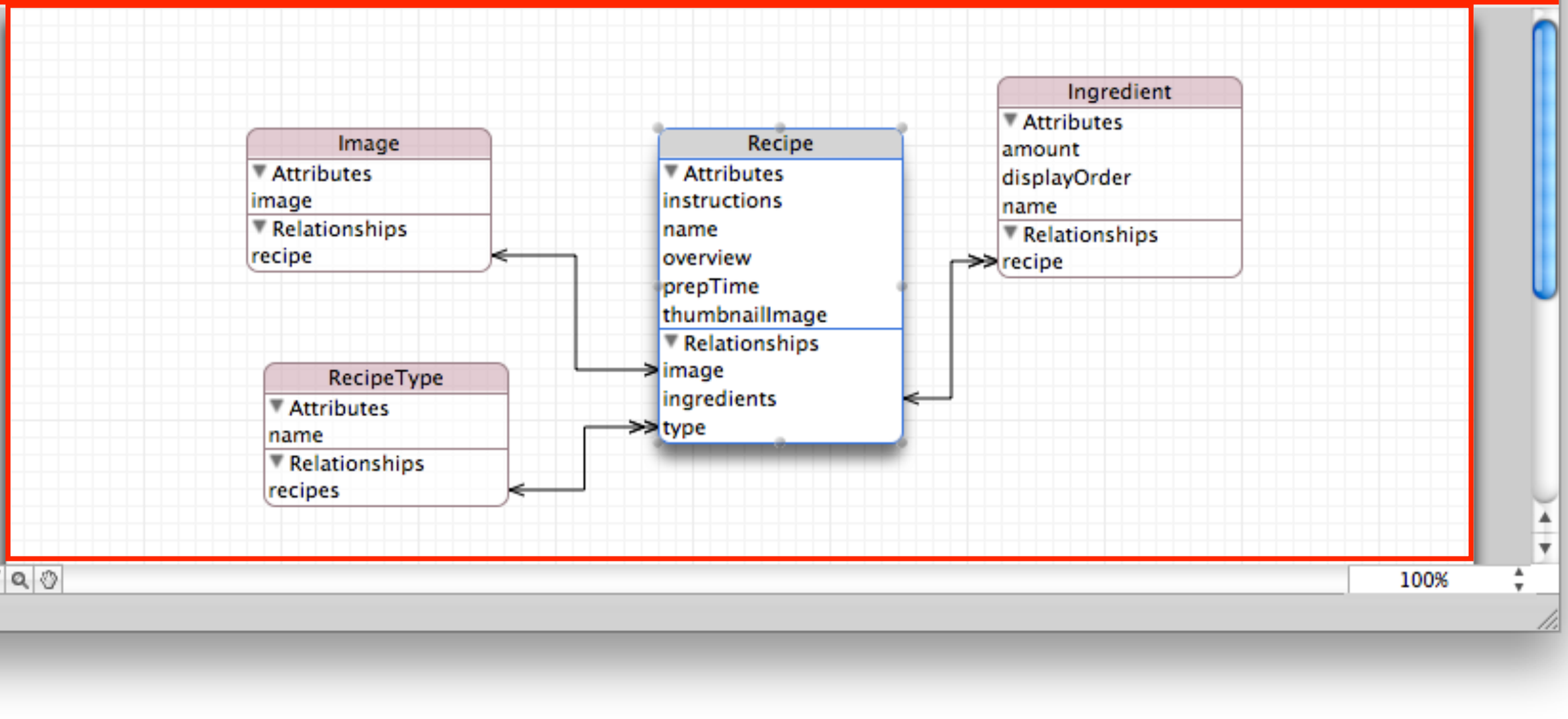
Destination: Image

Inverse: recipe

☐ To-Many Relationship

Min Count: 1 Max Count: 1

Delete Rule: Cascade



100%

Custom Model Class

```
// Recipe.swift
class Recipe: NSManagedObject {

}

// Recipe+CoreDataProperties.swift
extension Recipe {
    @NSManaged var title: String?
}
```

Creating / Deleting Managed Objects

- Create (Insert)
  - NSManagedObjectContext
  - Entity name

```
NSEntityDescription.insertNewObjectForEntityForName(entity.
    name, inManagedObjectContext: managedObjectContext)
```

- Delete
  - NSManagedObjectContext
  - NSManagedObject

```
managedObjectContext.deleteObject(object)
```

- Save context to make change persistent

Retrieving Managed Objects

- NSFetchRequest
  - Entity: NSEntityDescription
  - Sorting criteria: NSSortDescriptors
  - Search criteria: NSPredicate
- NSFetchedResultsController
  - Tailored to provide data for UITableViews
  - Requires: NSFetchRequest, NSManagedObjectContext
  - Change tracking via delegate

Fetch Data

```
//Set up the fetch request
let fetchRequest = NSFetchRequest(entityName: "Receipt")

//Set up the sort descriptor
let sortDescriptor = NSSortDescriptor(key: "creationDate", ascending: false)
fetchRequest.sortDescriptors = [sortDescriptor]


// Execute the request
if let result = try? managedObjectContext.executeFetchRequest(fetchRequest) {
    // Do stuff
}
```

Undo Manager

```
// attach undo manager
managedObjectContext.undoManager = NSUndoManager()

//perform undo
managedObjectContext.undo()
```

Demo

Performance

- Use batch fetches
- Just fetch needed values
- Use predicates to filter results
- Use SQLite for

```
let ed = NSExpressionDescription()
ed.name = "minimum"
```

```
-com.apple.CoreData.SQLDebug 1
```

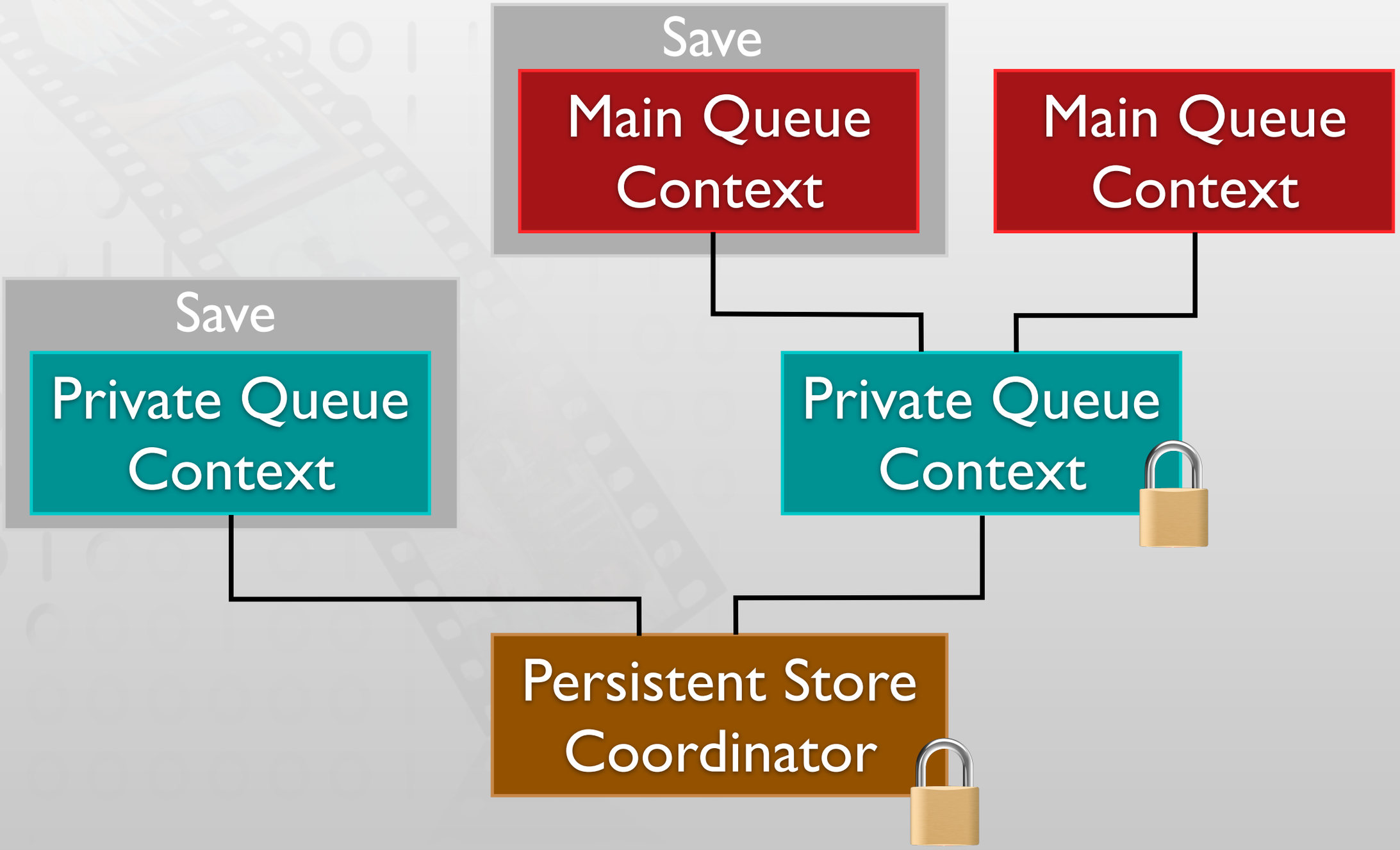
```
fetchRequest.propertiesToFetch = [ed]
```

```
fetchRequest.propertiesToFetch = ["name"]
fetchRequest.propertiesToGroupBy = ["major"]
fetchRequest.relationshipKeyPathsForPrefetching = ["course"]
```

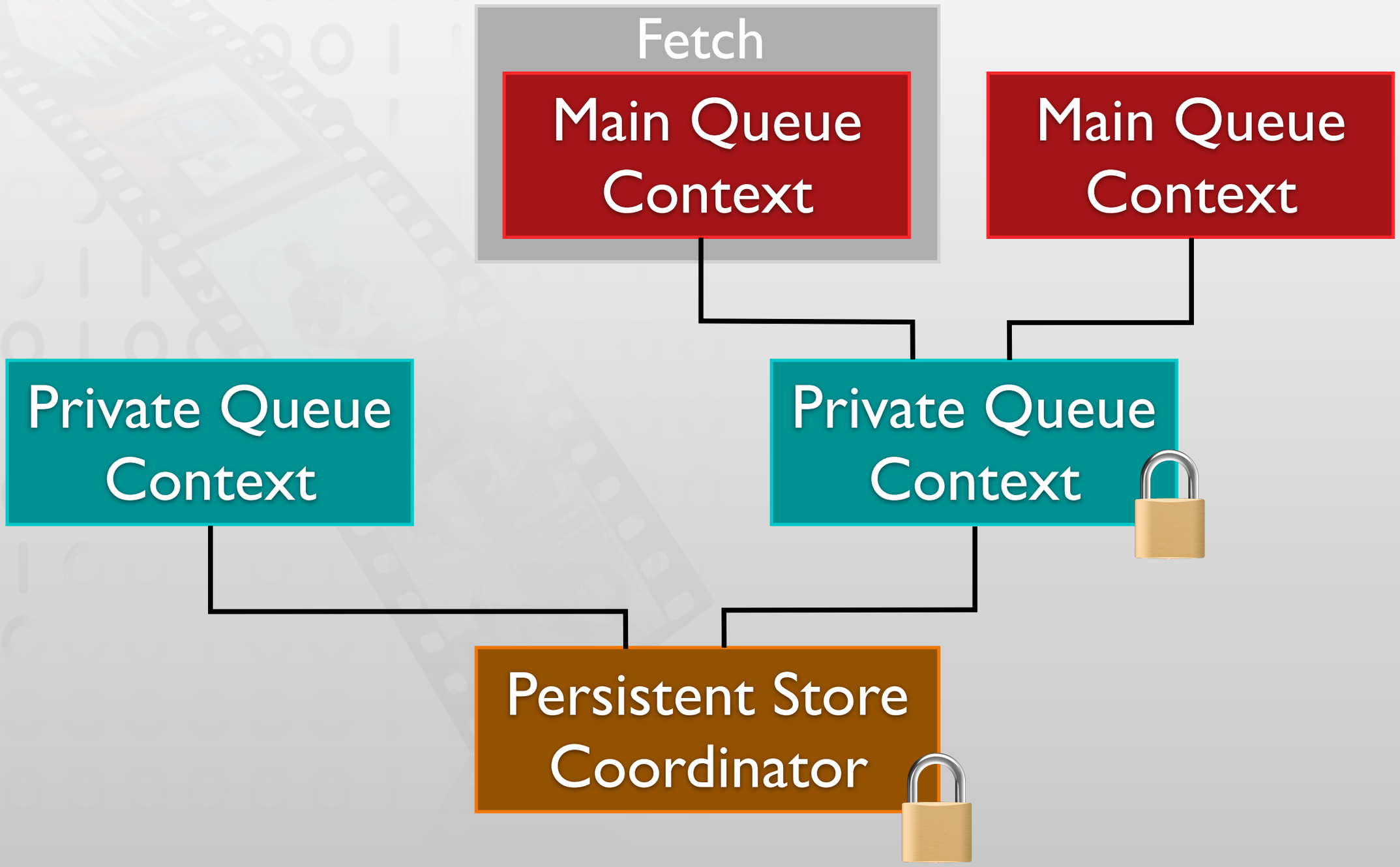
Numeric comparison is cheap

Text comparison is expensive

Concurrency



Concurrency



Data Handling Overview

	"Shoebox"	Document	App Data "Settings"
Local	Database	UIDocument	NSUserDefaults
Synced	Database	UIDocument with iCloud	iCloud Key/Value GameCenter

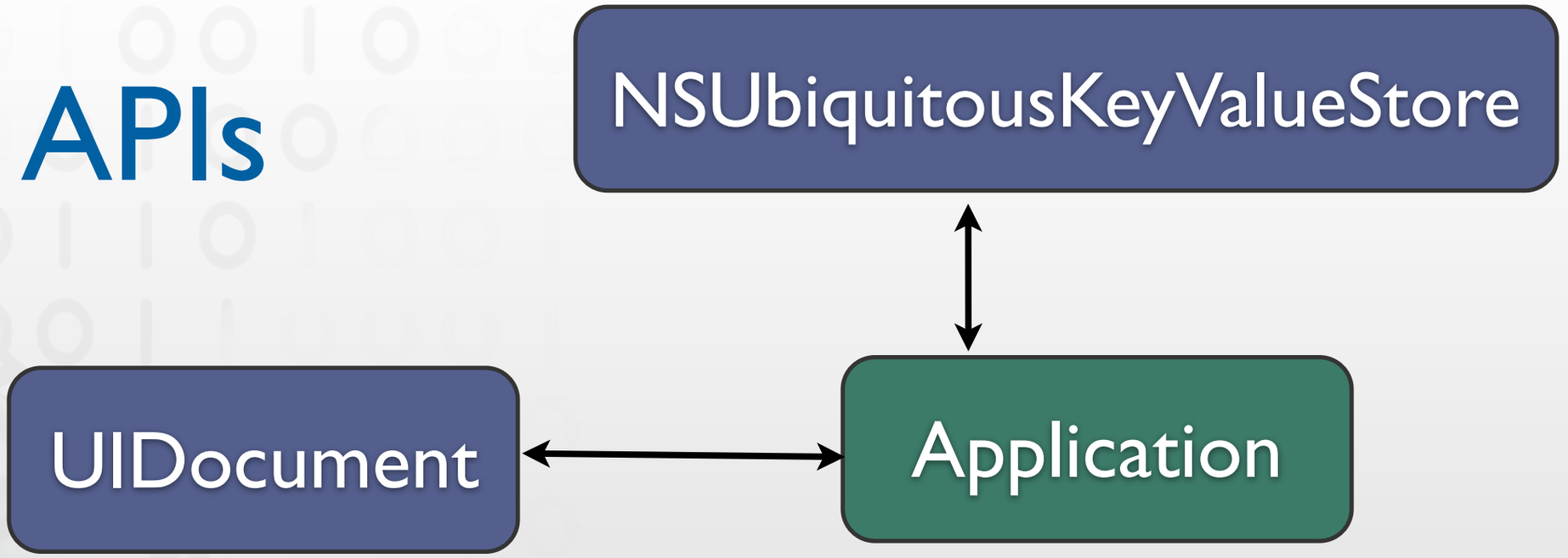
Online: Communication with a web server

iCloud

iCloud Basics

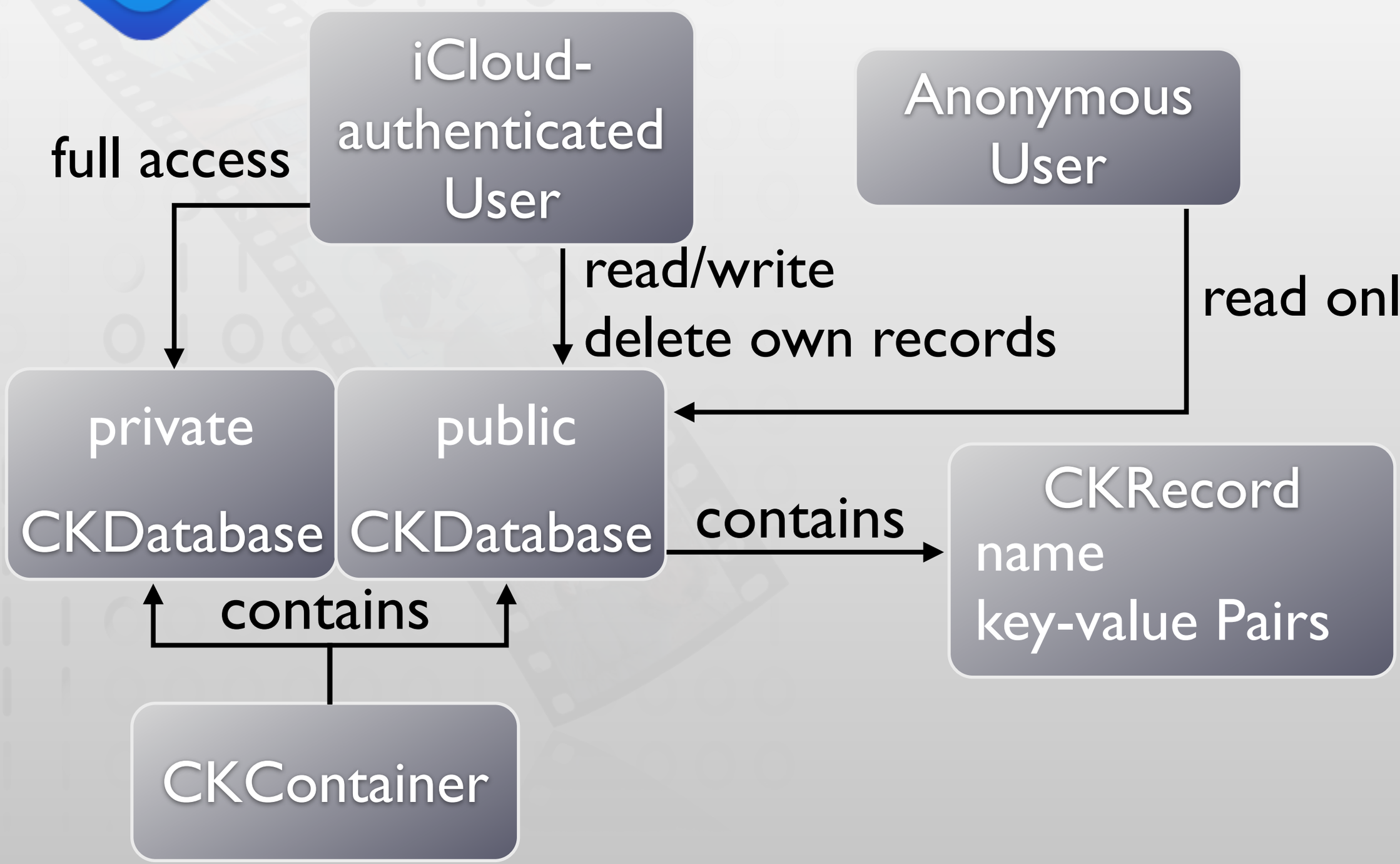
- Changes on one device are automatically pushed to other devices
- User can pick up any device and will always have access to her data
- Sync is per app and per user
- Free subscription is 5GB shared among all apps

iCloud APIs



- Settings: use NSUbiquitousKeyValueStore (similarly to NSUserDefaults)
  - 1MB limitation
  - Syncs within minutes
  - Last value always wins
- Documents: UIDocument on a URL in a special folder
  - No limit
  - Syncs within seconds
  - Can merge conflicting files

CloudKit

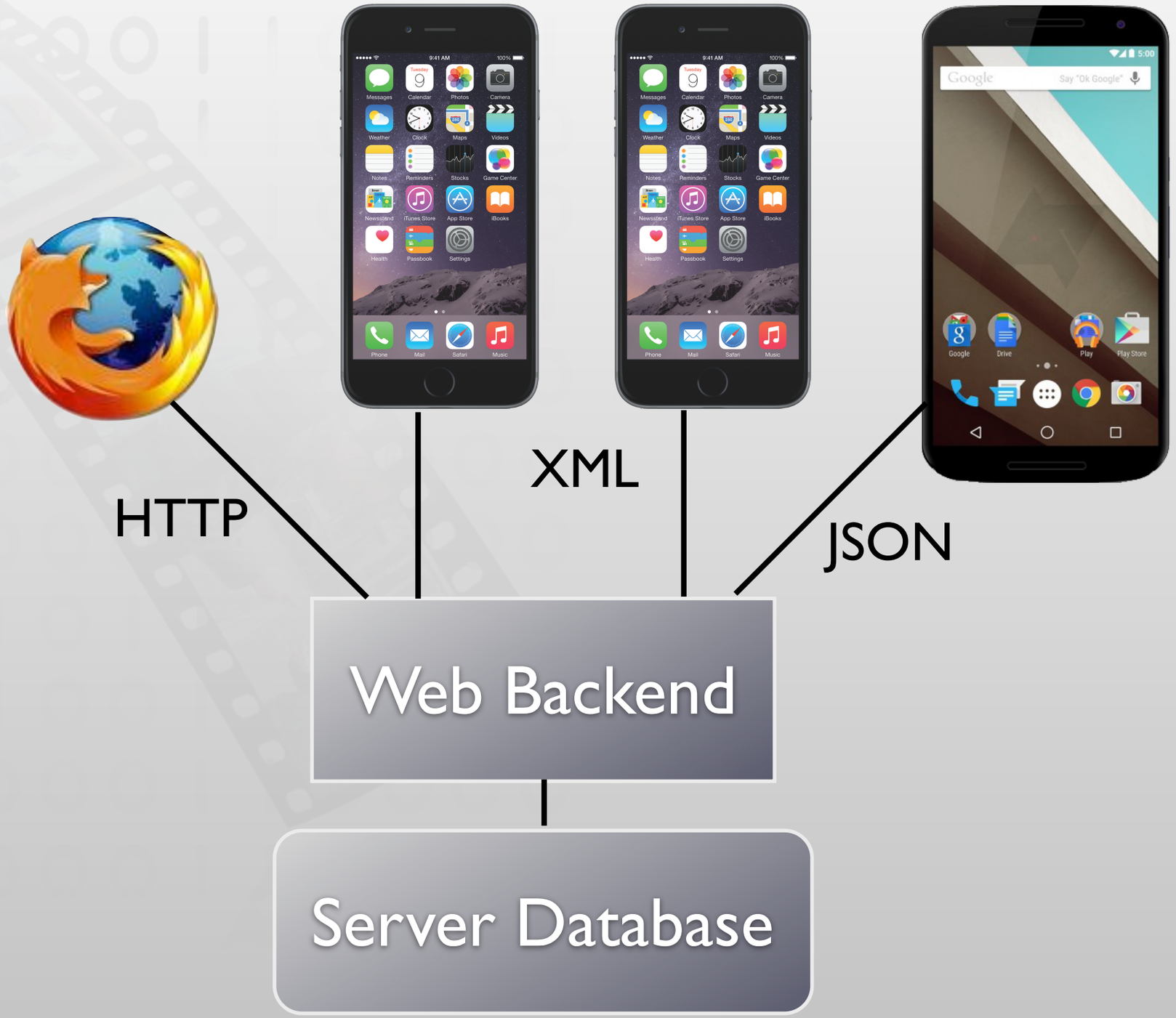

iCloud-authenticated User
Anonymous User
full access
read/write
delete own records
read only
private CKDatabase
public CKDatabase
contains
CKRecord
name
key-value Pairs
contains
CKContainer

Data Handling Overview

	"Shoebox"	Document	App Data "Settings"
Local	Database	UIDocument	NSUserDefaults
Synced	Database	UIDocument with iCloud	iCloud Key/Value GameCenter

Online — Communication with a web server

Web Backends



Representational State Transfer (REST)

- Resource manipulation via HTTP operations
- URL describes resource
  - http://my.server.com/person/127.xml

Server Address

Resource Name

Resource Id

Format

- Data encoded in XML or JSON
- GET, PUT, POST, DELETE operations are used to manipulate resources

iPhone HTTP Requests

```
let session = NSURLSession()
let url = NSURL(string: "http://hci.rwth-aachen.de")!

session.dataTaskWithURL(url) { (data, response, error) -> Void in
    print(data)
}
```

Remote (Web) Objects

Server Database

Web Backend

NSURLConnection
NSURLSession

NSData

NSXMLParser

NSJSONSerialization

Basic Data Types

RestKit Basics

- RestKit knows RESTful Webservices
- Can interact with CoreData
- Key paradigm: Mapping response from the web to Obj-C objects

RestKit Structure

Entity/Object Mapping

Request Descriptor

Response Descriptor

Object Manager

Managed Object Store

Demo

Data Handling Overview

	"Shoebox"	Document	App Data "Settings"
Local	Database	UIDocument	NSUserDefaults
Synced	Database	UIDocument with iCloud	iCloud Key/Value GameCenter

CoreData

Online — Communication with a web server

More Information

- SQLite Online Documentation
- Apple Guides
  - Archives and Serialization Programming Guide
  - Core Data Programming Guide
  - Document-Based Application Programming Guide
  - URL Loading System Programming Guide
- restkit.org
- github.com/lichtschlag/iCloudPlayground

Epilogue: Sharing data locally

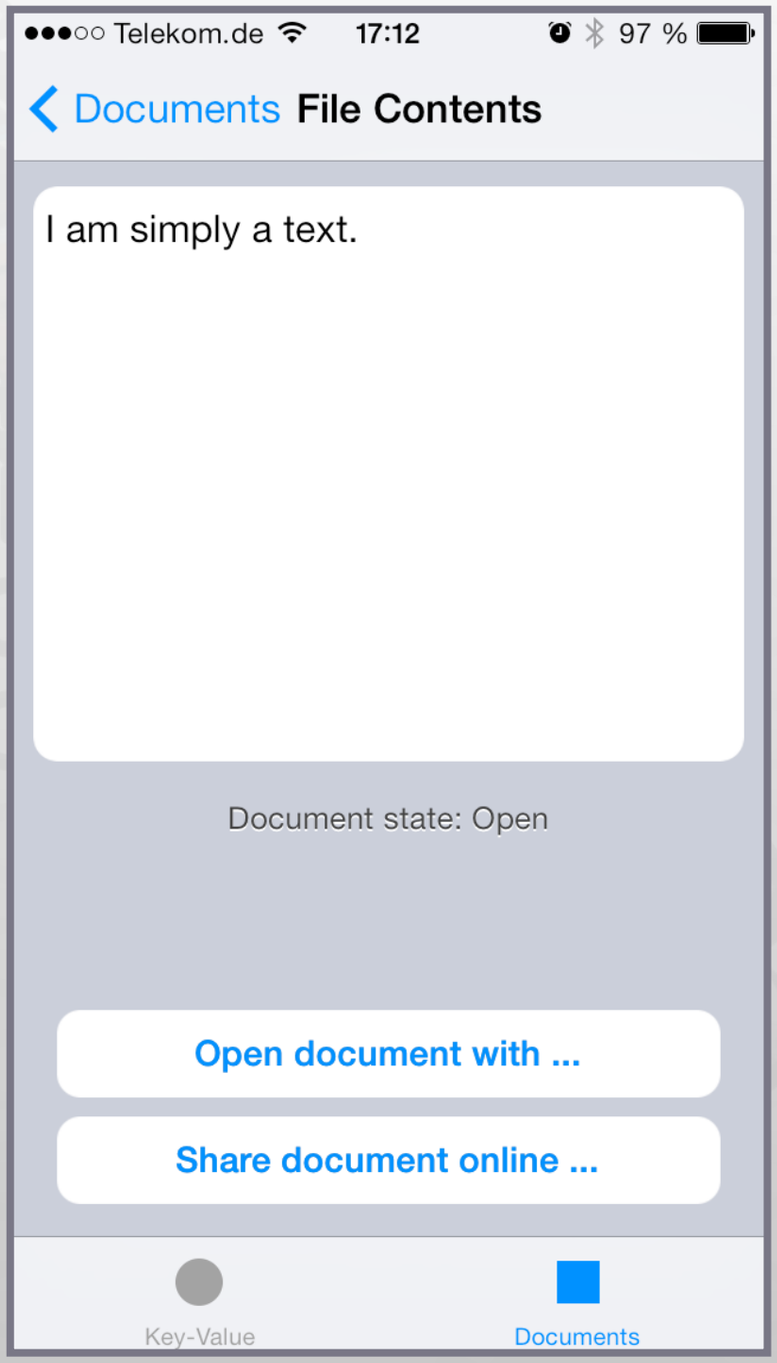
- How to get data from app to app?
- “Open with...” ?

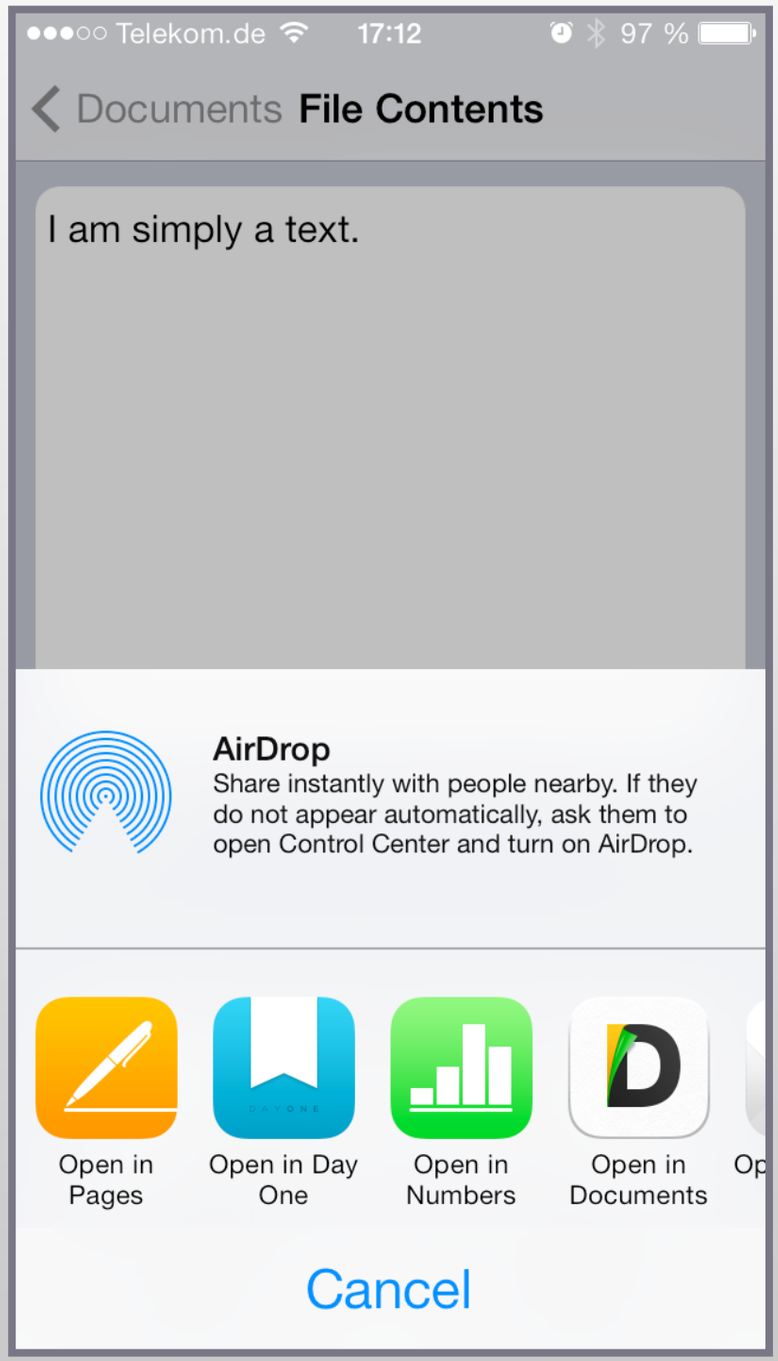
- UIDocumentInteractionController
  - Apps publish what files they can open
  - Sender app pushes document, user selects target app
  - Data is copied between app sandboxes
  - No way to track files

UIDocumentInteractionController

```
func openExternally() {
    guard let url = self.document?.fileURL else { return }
    self.document?.saveToURL(url, forSaveOperation: .ForOverwriting,
completionHandler: { (success) -> Void in
        self.docController = UIDocumentInteractionController(URL: url)
        if let success =
self.docController?.presentOpenInMenuFromRect(CGRectZero, inView:
self.openButton!, animated: true) {
            if (!success) {
                let alertController =  UIAlertController(title: "Cannot open
file in other apps", message: "No App can handle", preferredStyle: .Alert)
                let defaultAction = UIAlertAction(title: "OK", style: .Default,
handler: nil)
                alertController.addAction(defaultAction)
                self.presentViewController(alertController, animated: true,
completion: nil)
            }
        }
    })
}
```

UIDocumentController in Action





Check the iCloudPlayground demo code for more info