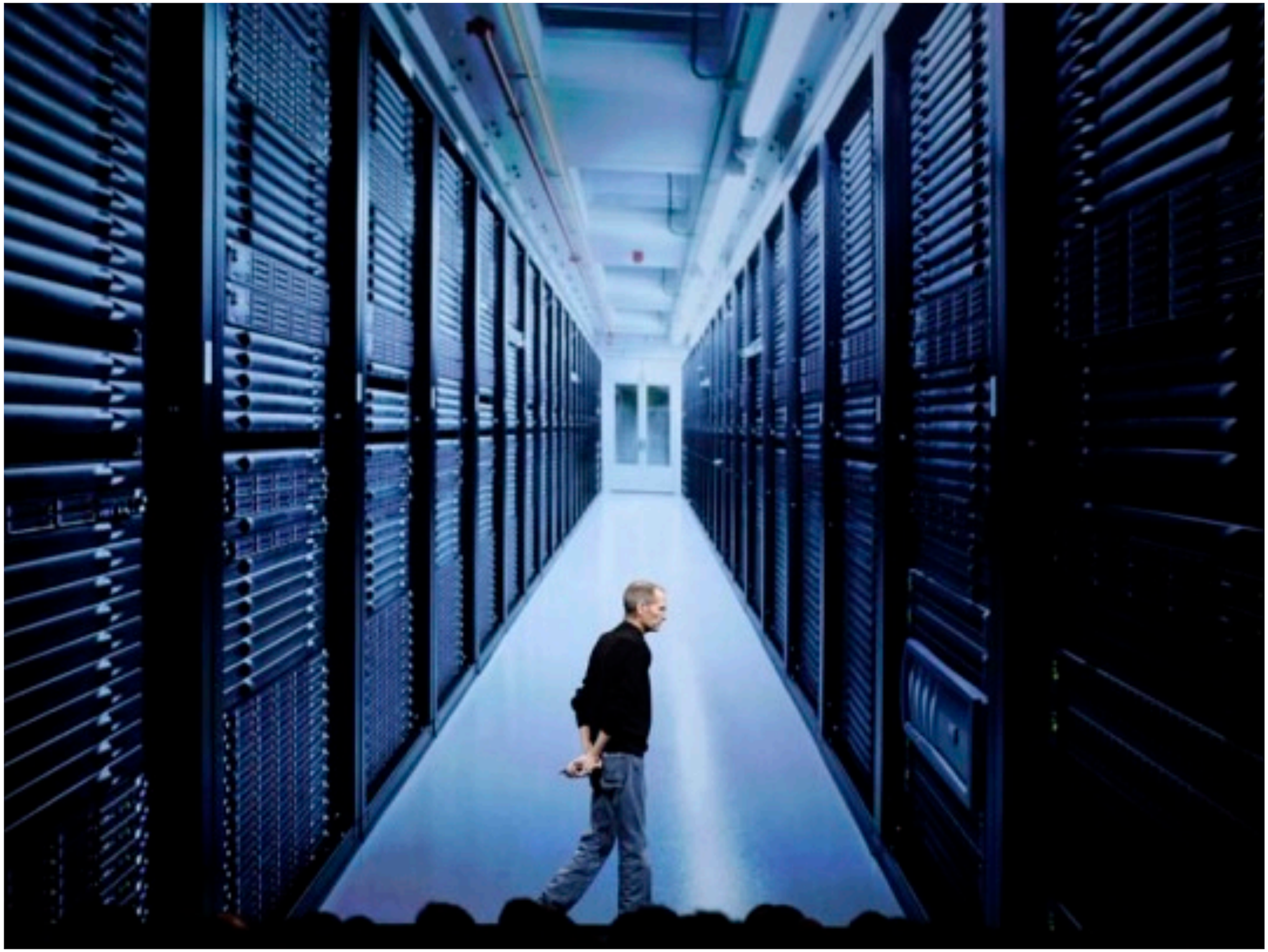
# How does iCloud work?

Christian Beer
http://chbeer.de

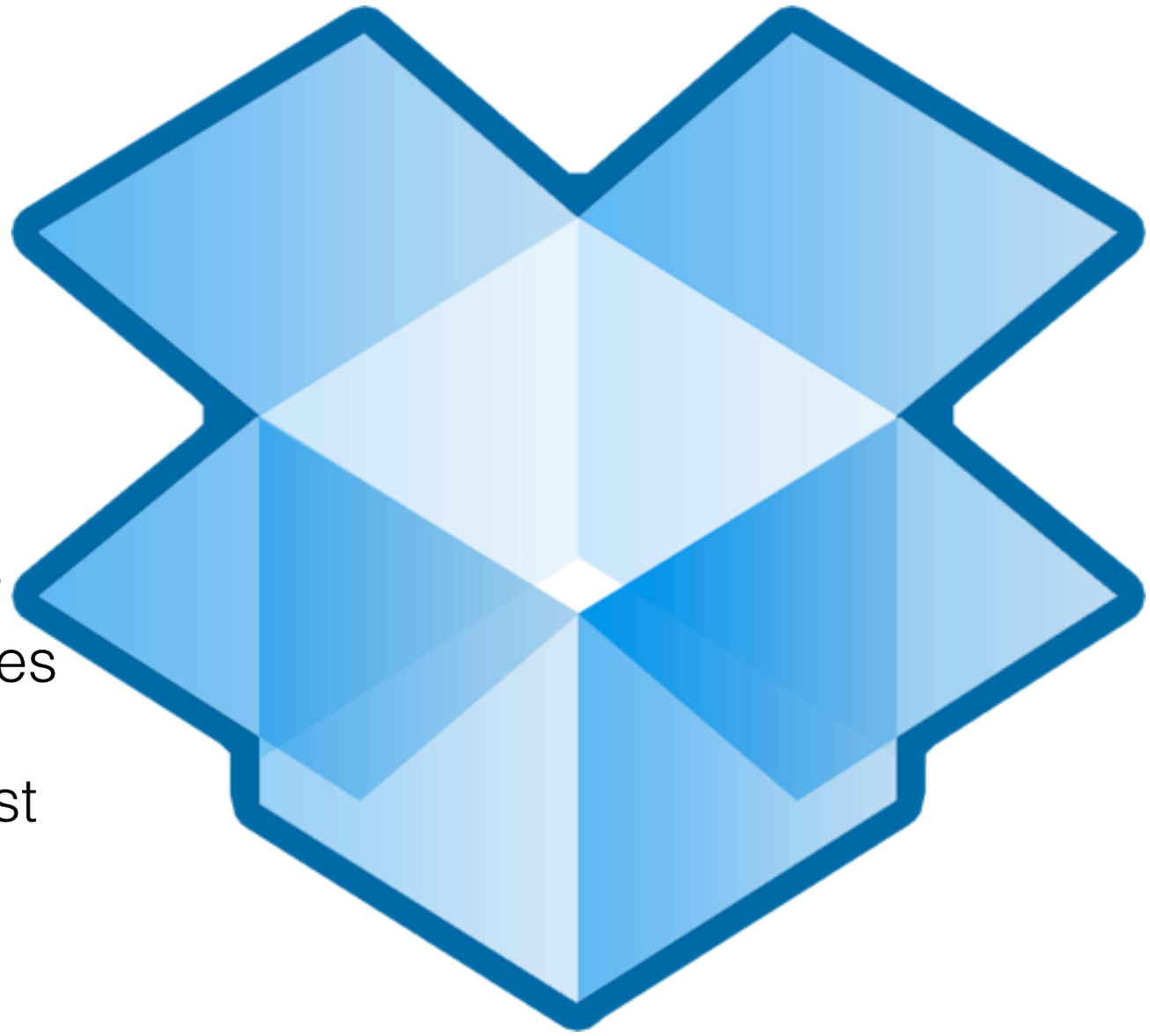
# iCloud = Datacenter

# iCloud = File sync



# Like Dropbox?

- Syncs files
- In background
- One shared directory („Ubiquitous“ container)
- Files in ubiquitous container are synched between devices
- iOS devices: sync on request
- Mac: syncs all data



# But different!

- No files accessible by user
- Only for apps
- Can only access own directory
- Conflict management
- No versioning (at least not visible / accessible)

# iCloud

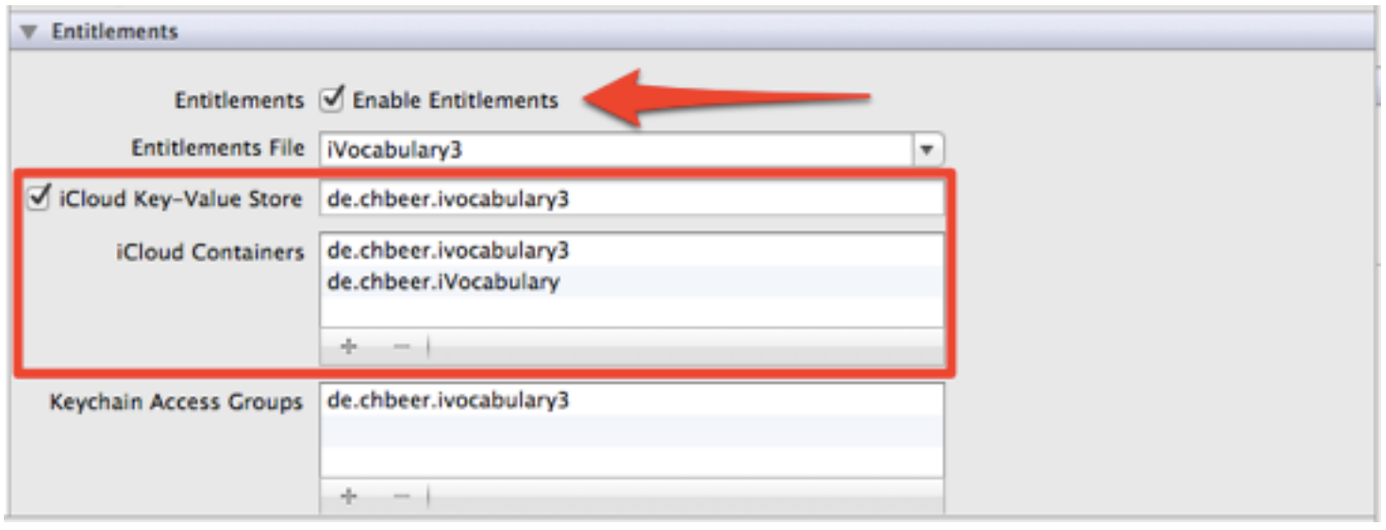
- Key-Value sync (like NSUserDefaults)
- File sync / Documents sync
- Core Data sync (Shoebox)
- Core Data Documents sync

# Gettin' started

- Activate iCloud for app:

| Description | Apple Push Notification service | In App Purchase | Game Center | iCloud | Action |
|---|---|---|---|---|---|
| **2SUZ2UZ37H.*** Xcode: Wildcard AppID | Unavailable | Unavailable | Unavailable | Configurable | Configure |
| **3DBL5M5TFG.de.chbeer.iVoc...** iVocabulary | Configurable for Development<br>Configurable for Production | Enabled | Enabled | Enabled | Configure |
| **3DBL5M5TFG.de.chbeer.ivoc...** iVocabulary 3 | Configurable for Development<br>Configurable for Production | Enabled | Enabled | Enabled | Configure |

- Add entitlements:



# Check for iCloud

- Is iCloud Document Storage available?

```
NSFileManager *fm = [NSFileManager defaultManager];
NSURL *url = [fm URLForUbiquityContainerIdentifier:nil];
BOOL iCloudAvailable = (url != nil);
```

- Do it in background!
- Do it early! (`application:didFinishLaunching...`)
- The URL is the base URL for your iCloud storage.

# Key-Value sync

- Limits:
  - 64 KB (and max 64 KB each key)
  - Not guaranteed
- Always store to NSUserDefaults as well!
- Saves to memory until synchonize

```
NSUbiquitousKeyValueStore *store;
store = [NSUbiquitousKeyValueStore defaultStore];
[store setLongLong:12 forKey:@“page“];
long page = (long)[store longLongForKey:@“page“];
[store synchronize];
```

# File sync

- Need to do conflict management yourself
- Move files explicitly in and out:

```
NSFileManager *fm = [NSFileManager defaultManager];
NSURL *iCloudURL = [ubiquitousURL URLByAppending...
BOOL success = [fm setUbiquitous:YES itemAtURL:fileURL
                  destinationURL:iCloudURL error:&error];
```

- Use a file coordinator for changes!!

```
NSFileCoordinator *fc = [...];
[fc coordinateReadingItemAtURL:fileURL [...]
                       byAccessor:^(NSURL *newURL){
    // read from newURL
}];
```

# File sync

- Download files, eventually:

```
NSFileManager *fm = [NSFileManager defaultManager];
if (![fm startDownloadingUbiquitousItemAtURL:file
                                       error:&error]) {
    // handle error
}
```

- Updates on files are recognized by NSMetadataQuery

# Get info about files

- Search files via NSMetadataQuery
- -[NSURL getResourceValue:forKey:error:]

```
NSError *error = nil;
NSNumber *isDownloaded = nil;
BOOL success = [fileURL getResourceValue:&isDownloaded
                                  forKey:NSURLUbiquitousItemIsDownloadedKey
                                                                  error:&error];
```

# UIDocument

- Like NSDocument
- Includes iCloud support
- implements NSFilePresenter
- Reading / writing: coordinated and in background
- Uses/provides NSUndoManager for auto-saving
- Conflict management still not solved

# Core Data (Shoebox)

- Like normal Core Data
- Simply add two options to NSPersistentStore:
  - ContentNameKey : Unique Name of the DB
  - ContentURLKey: URL for transaction logs (in ubiquitous container)

```
@“MyUniqueContentName“, NSPersistentStoreUbiquitousContentNameKey,
ubiquitousURL, NSPersistentStoreUbiquitousContentURLKey,

[NSNumber numberWithBool:YES], NSMigratePersistentStoresAutomaticallyOption,
[NSNumber numberWithBool:YES], NSInferMappingModelAutomaticallyOption,
nil
```

# Core Data Documents

- UIManagedDocument:
  UIDocument for databases
- Not only SQLite (also atomic store: XML, plist)
- Undo management
- managedObjectContext is a main thread context!

# Merging changes!

- Changes get merged into Core Data
- You need to update your active objects!

```
NSNotificationCenter nc = [NSNotificationCenter defaultCenter];

[nc addObserver:self
       selector:@selector(mergeChangesFrom_iCloud:)
           name:NSPersistentStoreDidImportUbiquitousContentChangesNotification
         object:doc.managedObjectContext.persistentStoreCoordinator];
```

```
- (void)mergeChangesFrom_iCloud:(NSNotification *)notification {
    NSManagedObjectContext* moc = [self managedObjectContext];
    [moc performBlock:^{
        [self mergeiCloudChanges:notification forContext:moc];
    }];
}
```

```
- (void)mergeiCloudChanges:(NSNotification*)note
                forContext:(NSManagedObjectContext*)moc {

    [moc mergeChangesFromContextDidSaveNotification:note];
}
```

# Find Files in iCloud

- NSMetadataQuery

```
NSMetadataQuery *query = [[NSMetadataQuery alloc] init];

query.searchScopes = [NSArray arrayWithObjects:
                      NSMetadataQueryUbiquitousDocumentsScope,
                      nil];
query.predicate = [NSPredicate predicateWithFormat:@"%K LIKE '*.ivoc'",
                   NSMetadataItemFSNameKey];

[[NSNotificationCenter defaultCenter] addObserver:self
                                         selector:@selector(metadataQueryNotification:)
                                             name:nil
                                           object:query];

[query startQuery];
```

# Find Files in iCloud

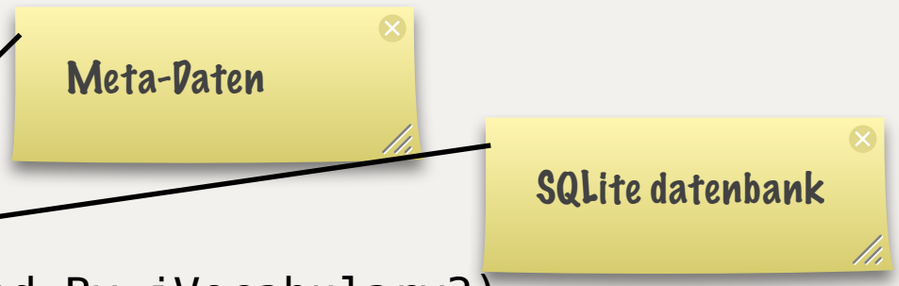
- NSMetadataQuery Notifications

```
- (void) metadataQueryNotification:(NSNotification*)notification
{
    NSString *name = notification.name;

    [self.query disableUpdates];
    if ([NSMetadataQueryDidFinishGatheringNotification isEqual:name] ||
        [NSMetadataQueryDidUpdateNotification isEqual:name]) {

        if ([NSMetadataQueryDidFinishGatheringNotification isEqual:name] {
            self.loading = NO;
        }

        [self updateItems:[_query results]];

        [self. tableView reloadData];
    } else if ([NSMetadataQueryDidStartGatheringNotification isEqual:name]) {
        self.loading = YES;
    }
    [self.query enableUpdates];
}
```

# In Depth

- UIManagedDocument bundle

```
.../Documents/
+- Botanik-Vokabular 5.ivoc/
    +-- DocumentMetadata.plist
    +-- StoreContent.nosync
        +-- persistentStore
        +-- (A Document Being Saved By iVocabulary3)
        +-- .persistentStore_SUPPORT
             +-- _EXTERNAL_DATA
```




- ...nosync = wird nicht gesynct

- DocumentMetadata.plist

```
<plist version="1.0">
    <dict>
        <key>NSPersistentStoreUbiquitousContentNameKey</key>
        <string>1F6CD4C5-31E7-4A52-8A83-9232D9F624AC</string>
```

# In Depth

- Transaction Logs

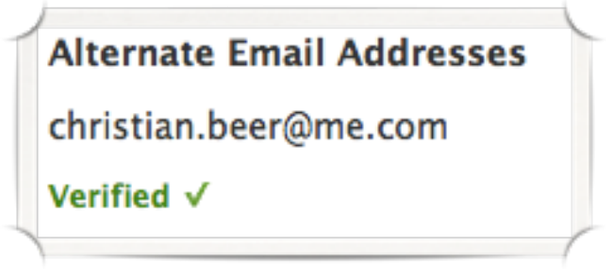
```
.../iVocabularyTransactionLogs
 +-- Botanik-Vokabular 5
      +-- .baseline,
            [...]
      +-- current.nosync
            +-- 1F6CD4C5-31E7-4A52-8A83-9232D9F624AC/[...UDID...]
                 +-- baseline.gcmodel
                 +-- baseline.meta
                 +-- baseline.model
+-- .cdmetadata,
+-- .cdmetadata.nosync,
      +-- mobile.9DFEE1D5-C465-5D7C-B478-0C404DB164D8/metadata.store
+-- mobile.9DFEE1D5-C465-5D7C-B478-0C404DB164D8,
      +-- A411676E-BCD1-419A-A1A7-18D94B3A7D9F/
           +-- CRqAE_3GRkYp_CLSWZBaSeyNHFeNEJ1QAQUdRLKCgnw=/
                +-- 97ECB2F7-886F-4C99-A54A-541523E523CB.1.cdt"
                +-- 64DA2527-09CE-43AE-B14E-A7AC59FDEFB0.1.cdt
```

Core Data Model

Transaktionen

# Problems

- Duplicate Apple ID (appleid.apple.com)




- Corrupt logs

- iCloud deletes files (Re: iCloud deleted my documents. How do I get them back) -> Duplicate Apple ID

# Debugging

- Good question!?

# iCloud Examples

- „Your Third iOS App“ (UIDocument)
  http://developer.apple.com/library/ios/#documentation/General/Conceptual/iCloud101/Introduction/Introduction.html

- Recipies sample app:
  - https://devforums.apple.com/thread/126670?tstart=0

Very helpful information!

- iCloud sample:
  - https://github.com/sonsongithub/iCloudSample

Not the nicest code.

# Helpful Resources

- iCloud forums:
  - https://devforums.apple.com/community/ios/ios5beta/icloud
  - https://devforums.apple.com/community/mac/lion/icloud
- iCloud document controller:
  - https://gist.github.com/1438679