

Macoun

iCloud unter der Haube

Max Seelemann

A **SOULMEN**TALK

Kennenlernrunde

Wer hat schon mal...

eine App mit iCloud gebaut?

Wer hat schon mal...

den Document Store genutzt?

Wer hat schon mal...

Spotlight für iCloud verwendet?

Wer hat schon mal...

File Coordination verwendet?

Wer hat schon mal...

File Presenter verwendet?

Wer hat schon mal...

**verschachtelte File Presenter
asynchron auf mehreren Queues
zusammenarbeiten lassen?**

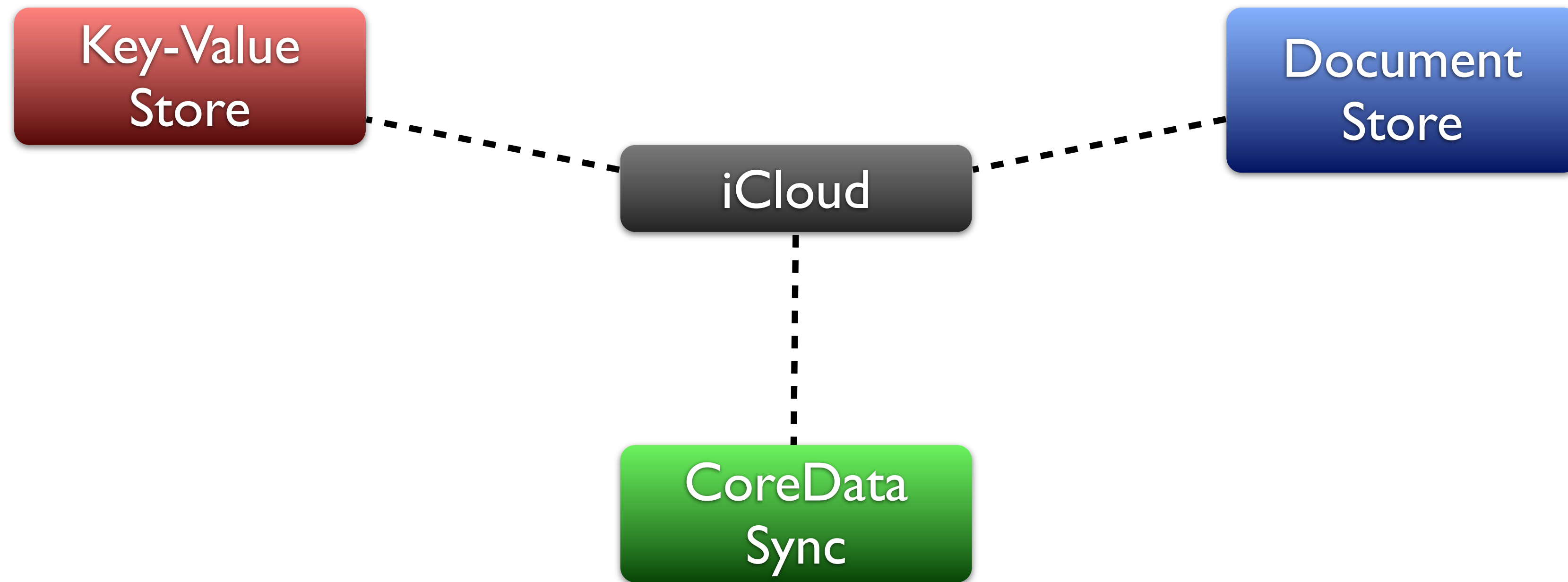
Allgemeine Ablauf-Theorie

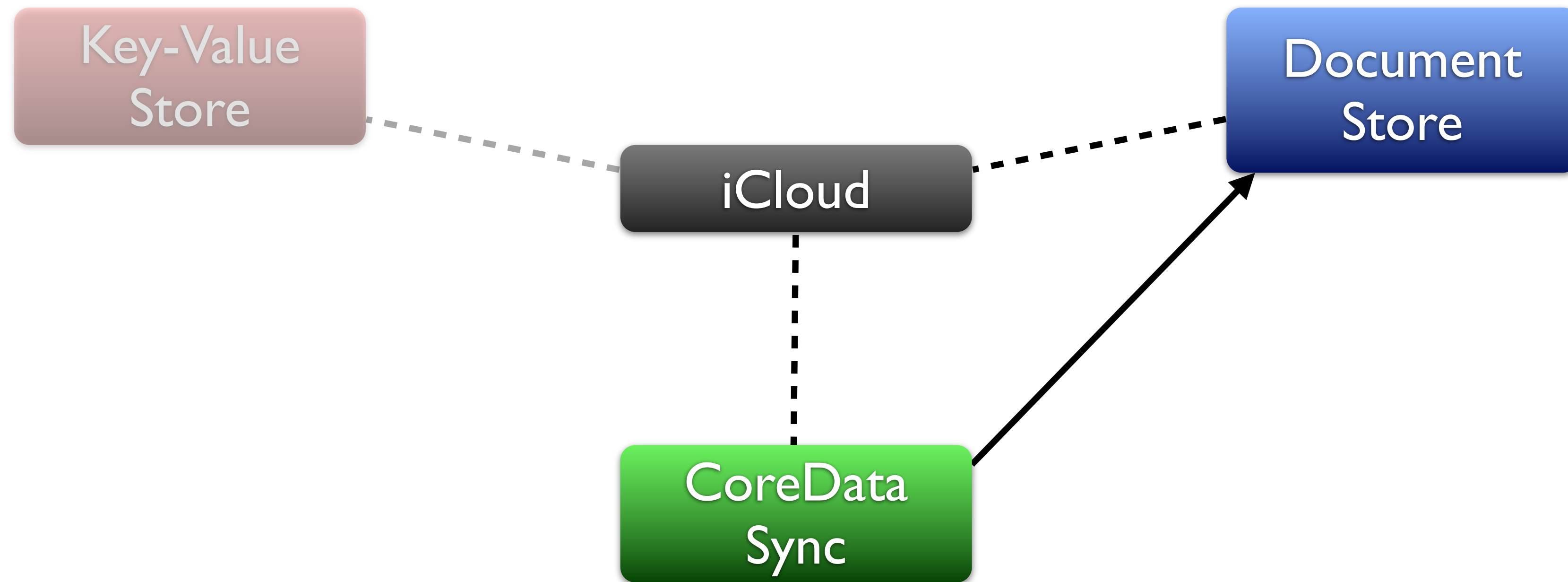
- Kein Fließtext
- Stichpunkte
- Fakten, Beobachtungen, Theorien, Vermutungen, Voodoo, etc.

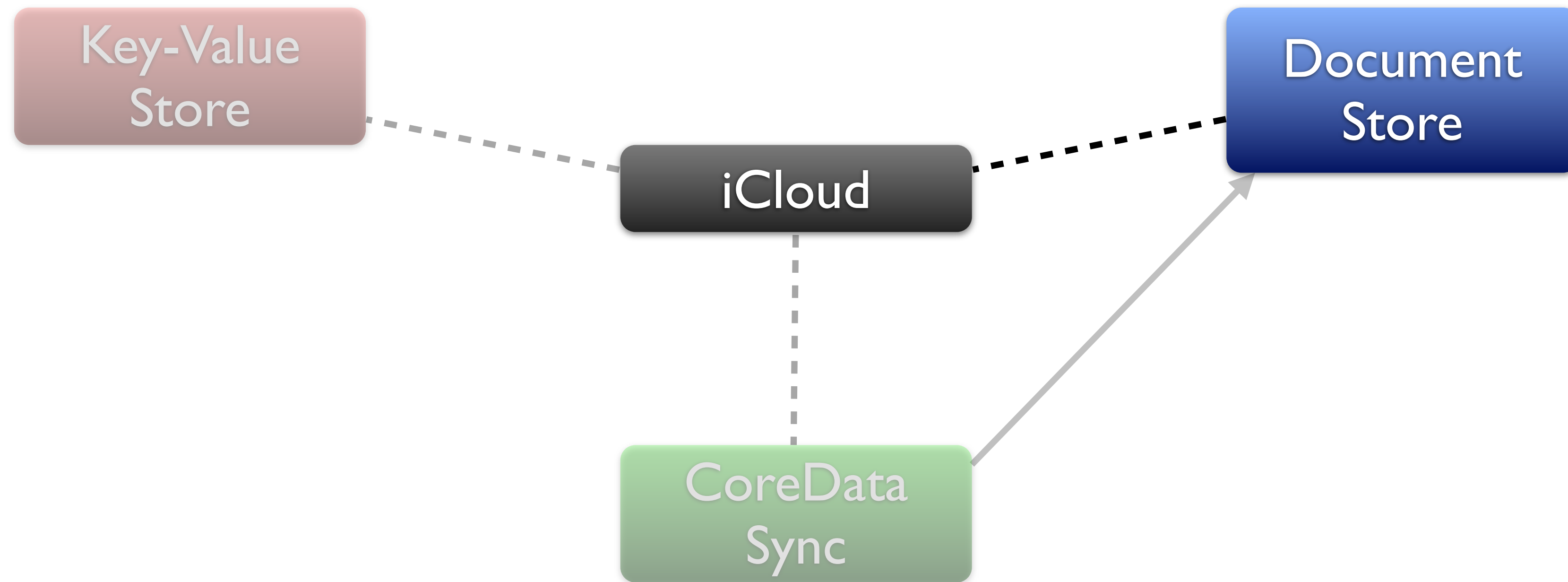
Spezielle Ablauf-Theorie

- Überblick
- Umsetzung auf dem Gerät
- Frameworks
- Manuelle Anbindung
- Probleme

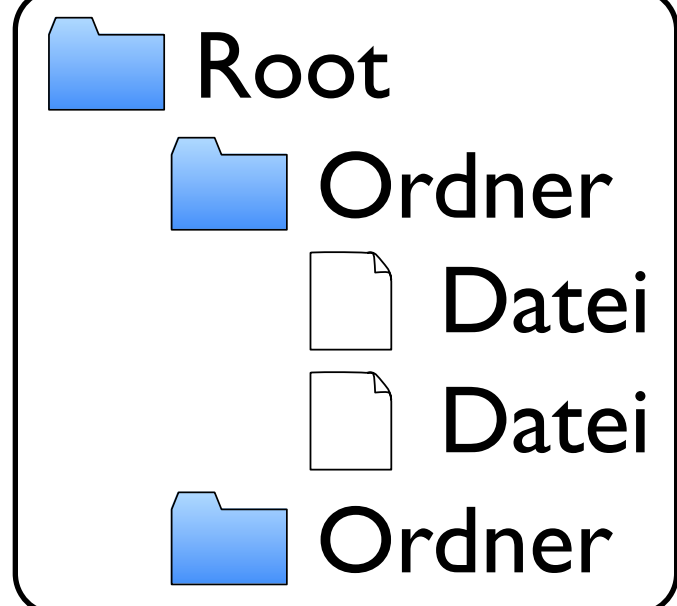
Überblick





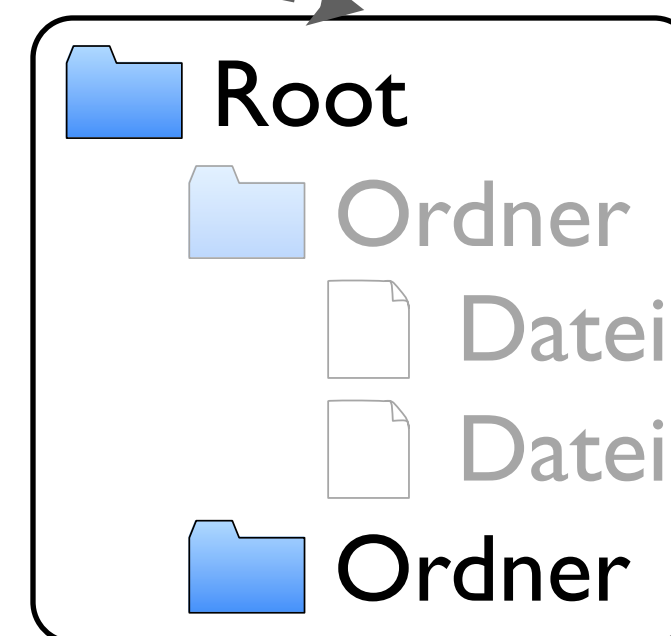
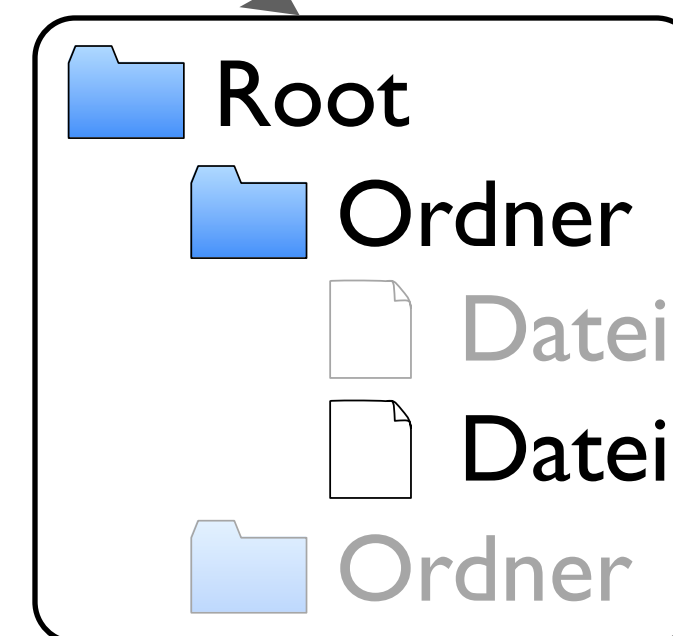
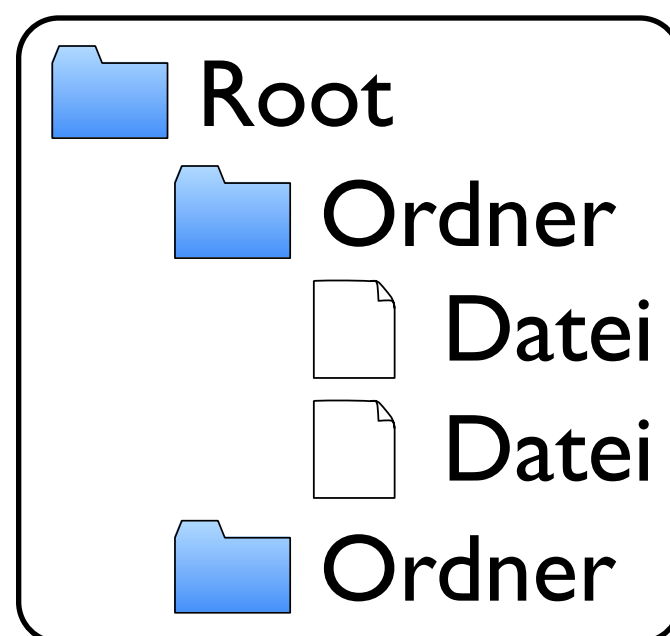


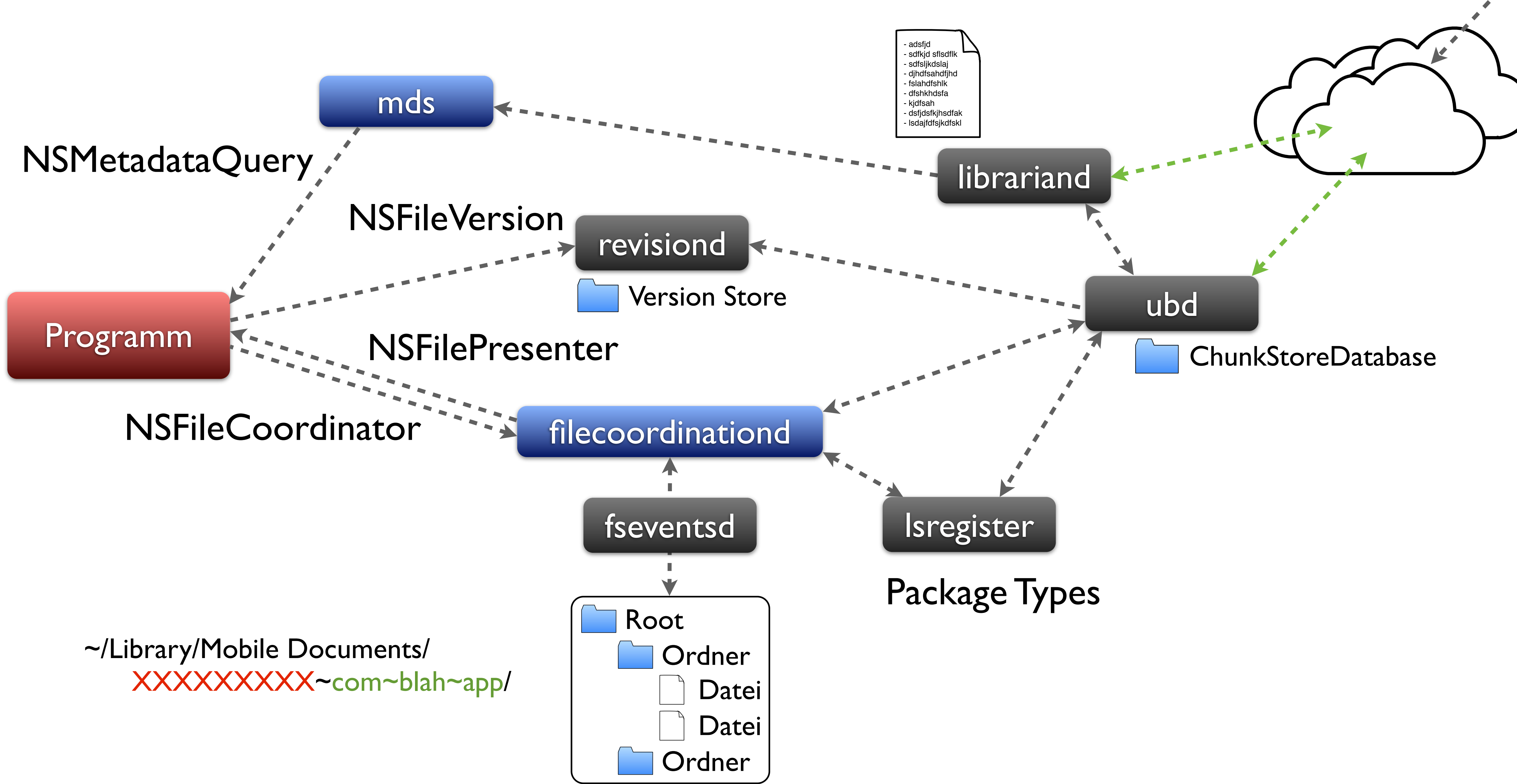
Document
Store

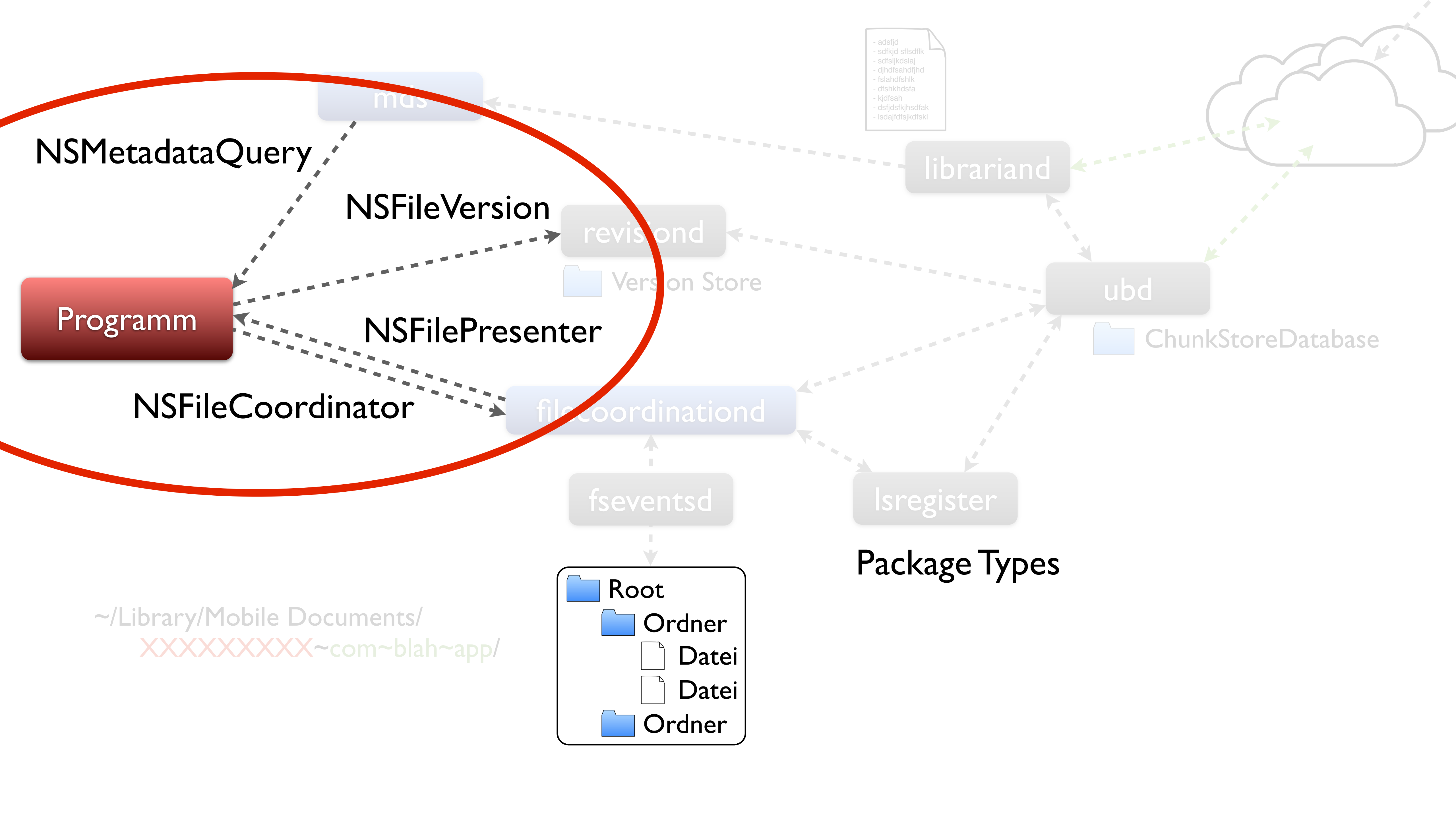


Automatisches
Download

Manuelles
Download

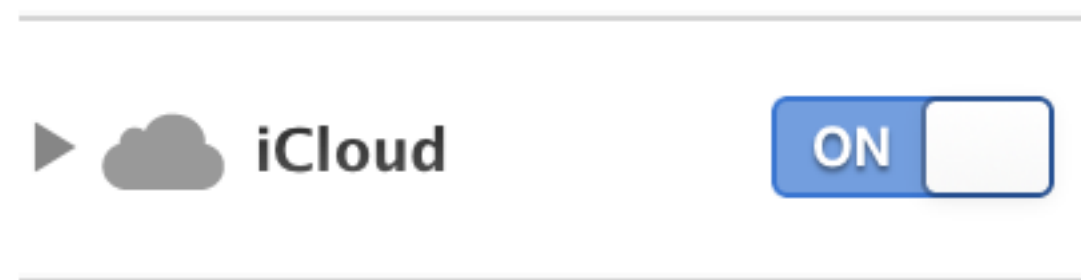






Framework

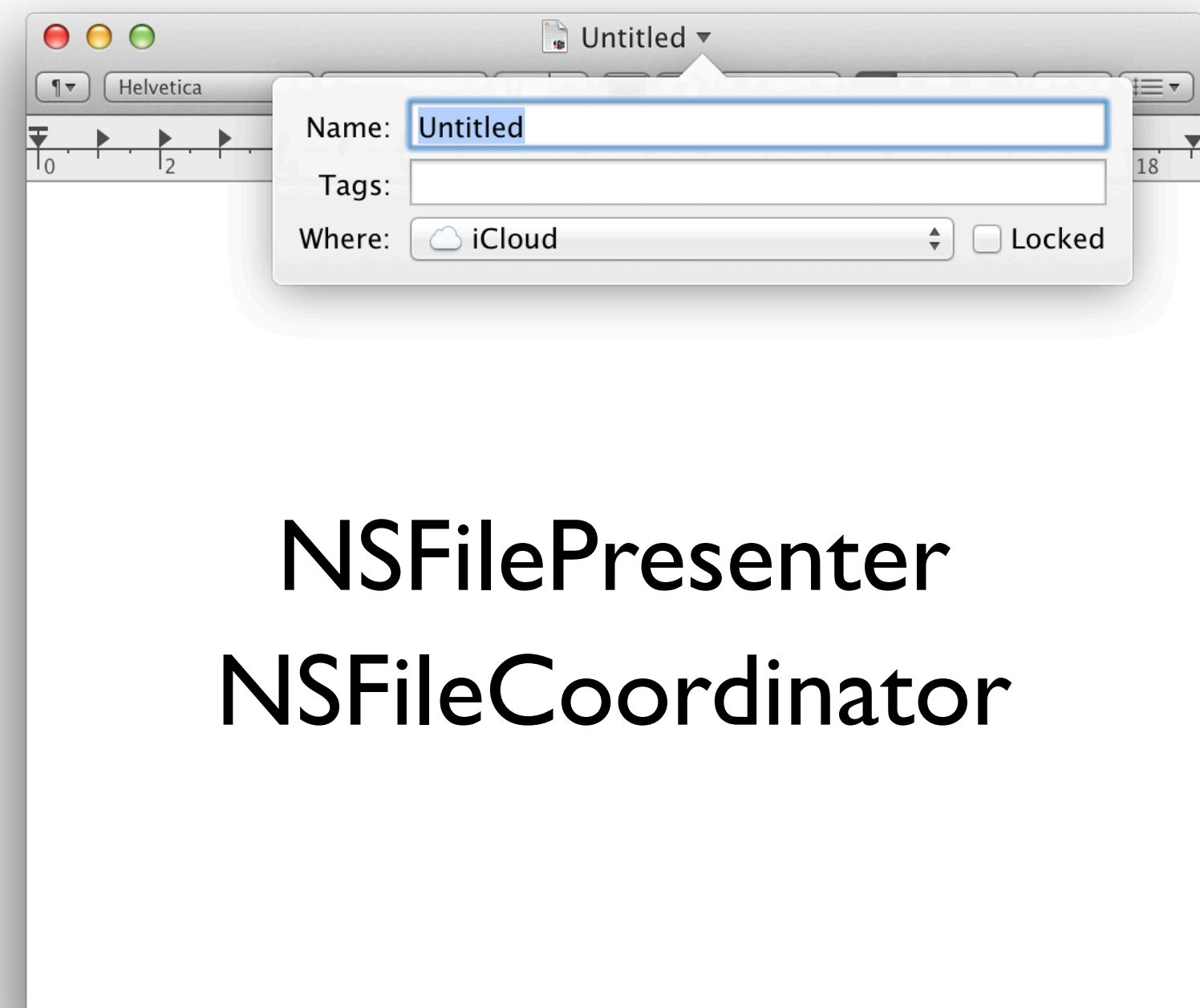
Xcode:



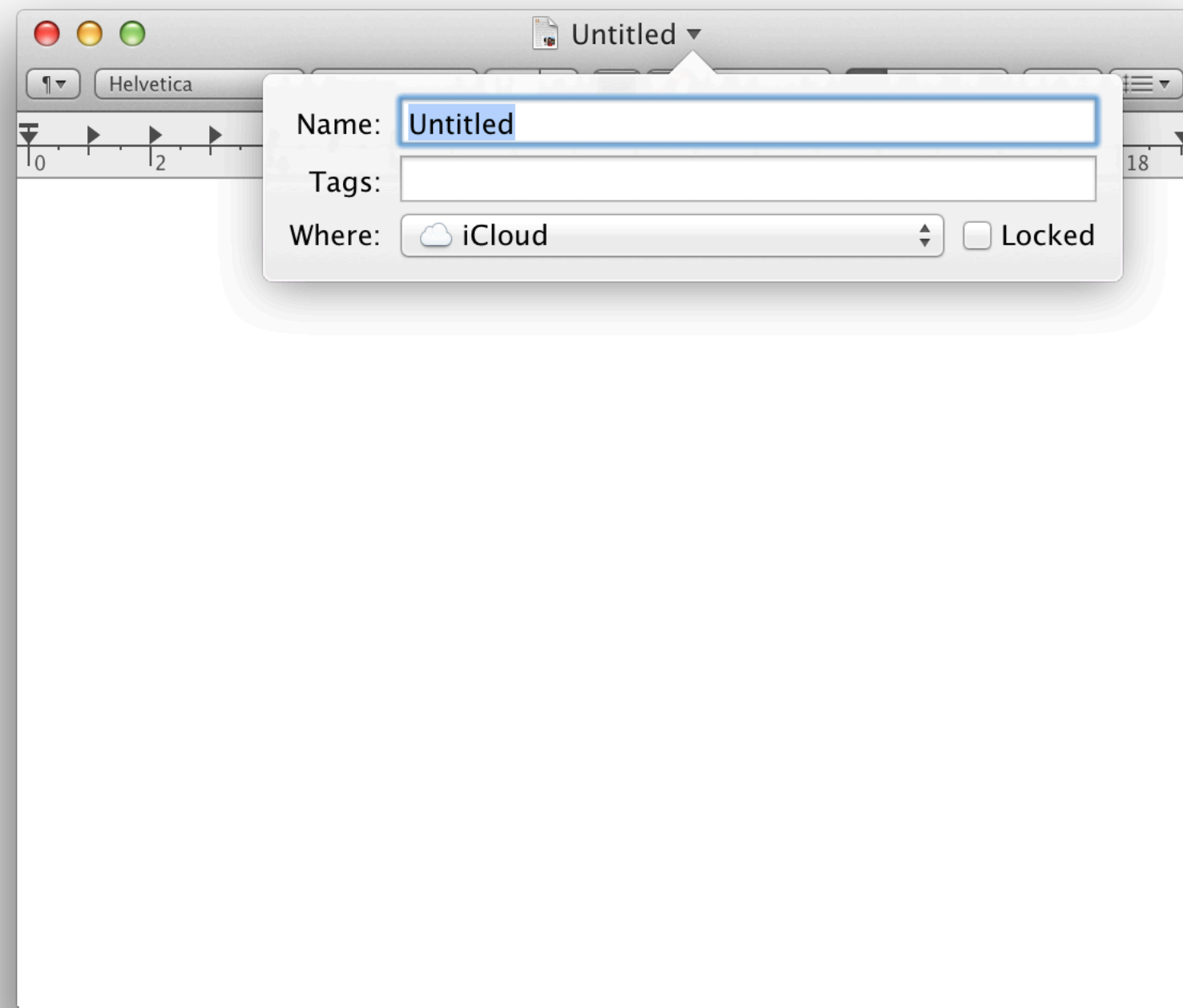
NSOpenPanel



NSDocument



NSDocument



Probleme

- Benötigt ein Fenster
- Deadlock-Gefahr
 - Nicht Unit-Testbar
- Schwergewichtig
- Keine Kontrolle
- Nicht erweiterbar
- Andere Klasse auf iOS

Selber machen?

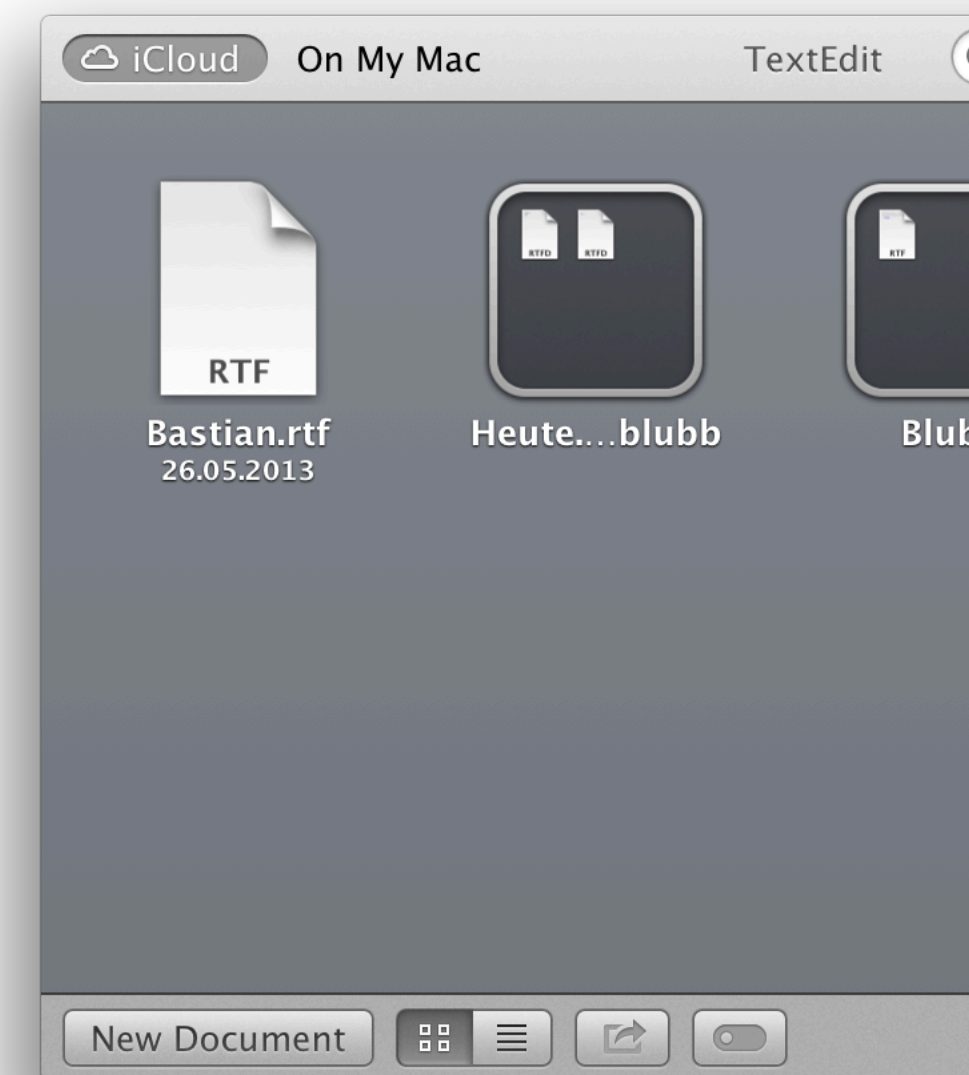
- File Presentation
- File Coordination
- Sicheres Speichern
- Asynchrones Laden, Speichern
- Konflikte behandeln
- Versionen verwalten
 - Versions-Browser



Selber Machen!

NSMetadataQuery

- Liste aller Dateien
- Regelmäßige Updates
- Keine Garantien
- Zustand (Verfügbar / Download / Upload)



NSMetadataQuery – Start

```
NSMetadataQuery *query = [NSMetadataQuery new];
query.searchScopes = @[NSMetadataQueryUbiquitousDocumentsScope];

[NSNotificationCenter defaultCenter addObserver:...
    name:NSMetadataQueryDidFinishGatheringNotification object:query];
[NSNotificationCenter defaultCenter addObserver:...
    name:NSMetadataQueryDidUpdateNotification object:query];

[query startQuery];
```

NSMetadataQuery – Update

```
- (void)syncStatusQueryChanged:(NSNotification *)notification {  
    [query disableUpdates];  
  
    for (NSUInteger index = 0; index < query.resultCount; index++) {  
        NSMetadataItem *result =  
            [_syncStatusQuery resultAtIndex: index];  
        NSURL *fileURL =  
            [result valueForKey: NSMetadataItemURLKey];  
  
        // Your code here ...  
    }  
  
    [query enableUpdates];  
}
```

NSFilePresenter

- Zeigt „**aktive Anzeige / Bearbeitung**“ einer Datei an
 - Überwachung einer Datei / eines Ordners
- Starke Garantien auf Inhalt & Integrität
- Notifikation bei Änderungen
- Synchrones Warten vor anderen Zugriffen

NSFilePresenter

- Protokoll implementieren

```
@protocol NSFilePresenter <NSObject>

@property (readonly) NSURL *presentedItemURL;
@property (readonly) NSOperationQueue *presentedItemOperationQueue;
```

- Presenter registrieren

```
[NSFileCoordinator addFilePresenter: self];
// ...
[NSFileCoordinator removeFilePresenter: self];
```


NSFilePresenter – Dateien

- Änderungen erhalten

```
- (void)presentedItemDidChange;  
- (void)presentedItemDidMoveToURL:(NSURL *)newURL;  
- (void)presentedItemDidGainVersion:(NSFileVersion *)version; ...
```

- Zugriffe erhalten

```
- (void)relinquishPresentedItemToReader:(void (^)(void (^__reacquirer)(void)))reader;  
- (void)relinquishPresentedItemToWriter:(void (^)(void (^__reacquirer)(void)))writer;  
- (void)savePresentedItemChangesWithCompletionHandler:  
    (void (^)(NSError *errorOrNil))completionHandler;  
- (void)accommodatePresentedItemDeletionWithCompletionHandler:  
    (void (^)(NSError *errorOrNil))completionHandler;
```

NSFileCoordinator

- **Geschützter Dateizugriff** auf Dateien und Ordner
- Starke Garantien auf Integrität
- Verbietet fremde bei Änderungen
- **Synchrones Warten** auf andere Zugriffe

NSFileCoordinator

- Erstellen

```
- (id)initWithFilePresenter:(id<NSFilePresenter>)filePresenterOrNil;
```

- Zugriffe ausführen

```
- (void)coordinateWritingItemAtURL:(NSURL *)url  
    options:(NSFileCoordinatorWritingOptions)options  
    error:(NSError **)outError  
    byAccessor:(void (^)(NSURL *newURL))writer;
```

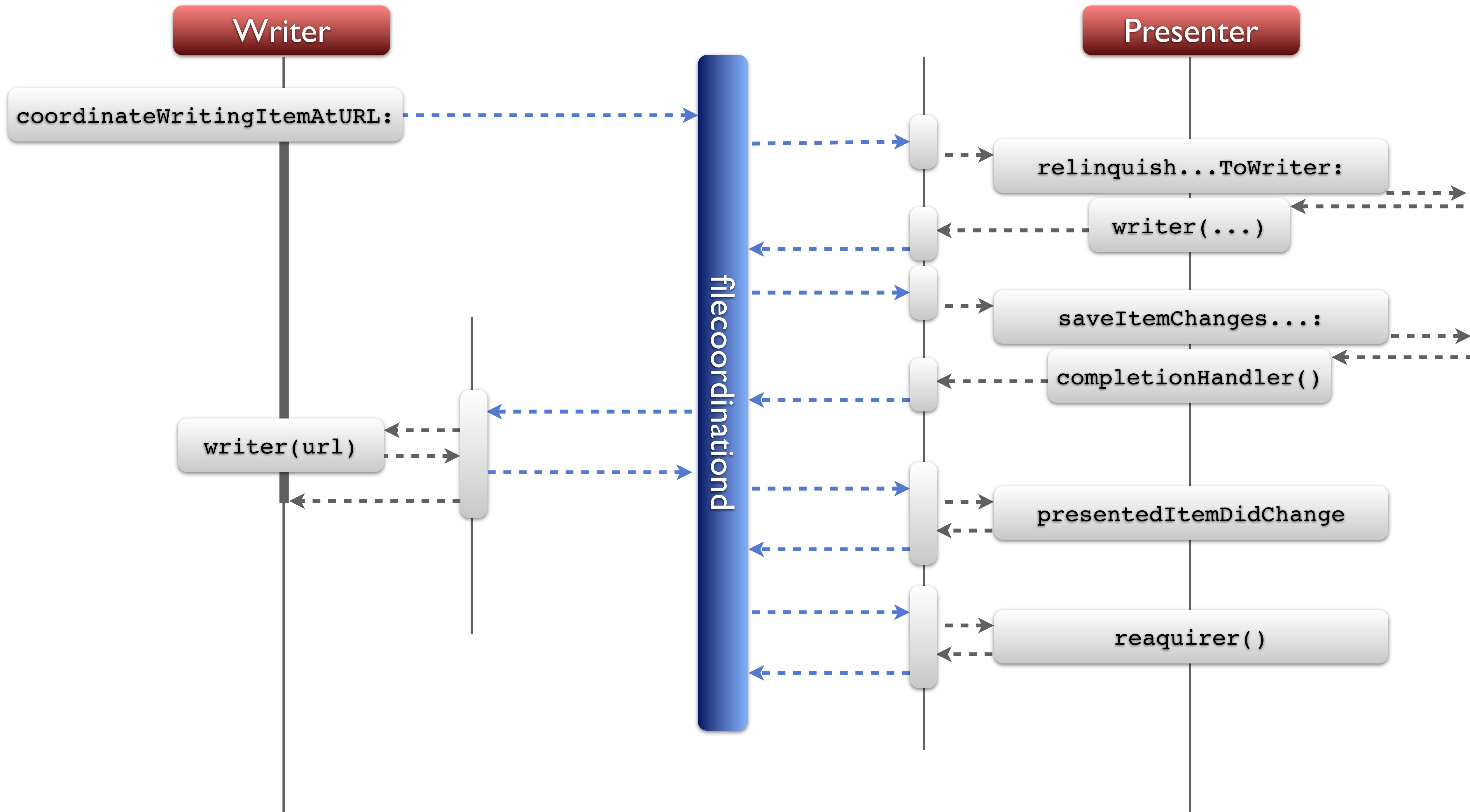
NSFileCoordinator

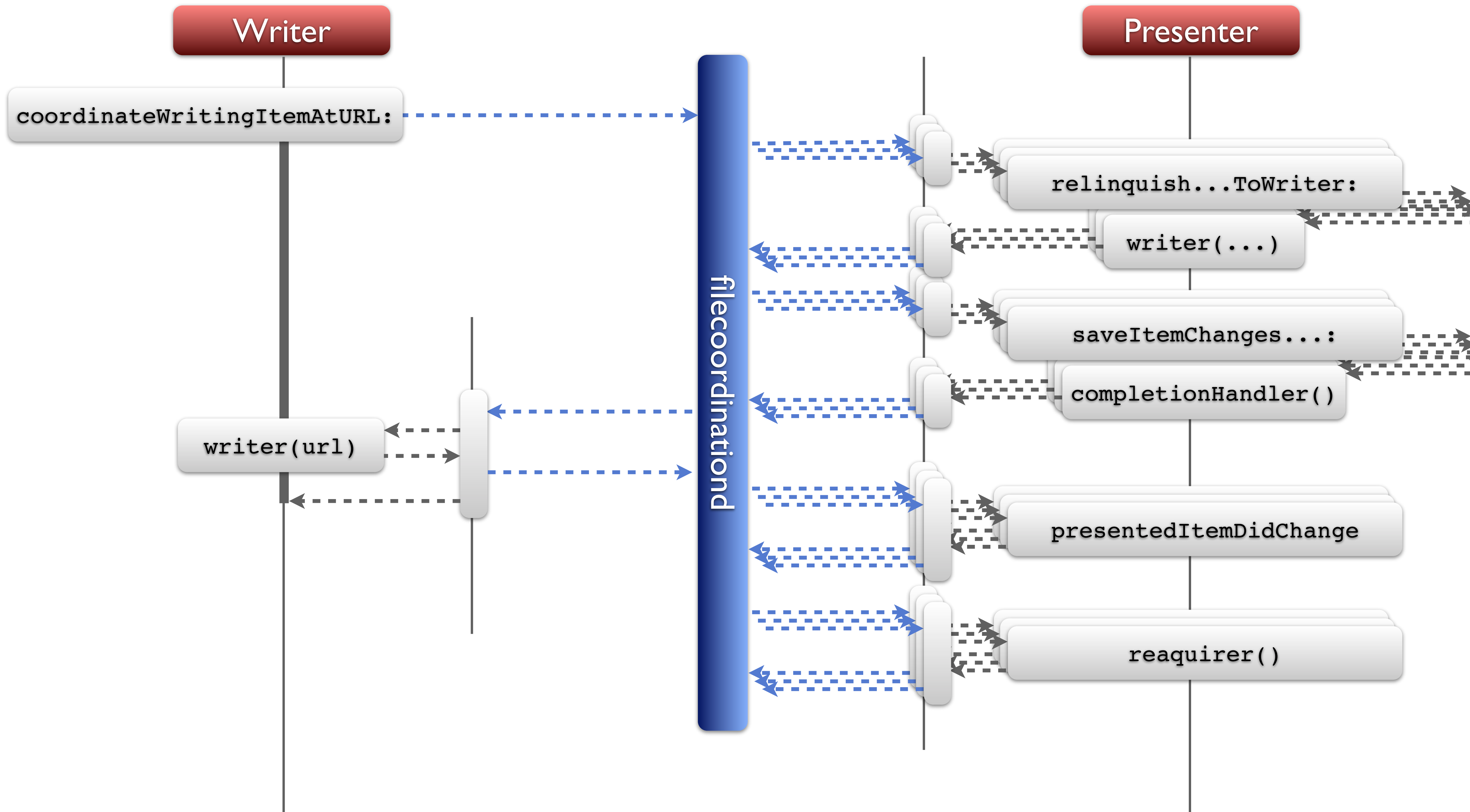
- Erstellen

```
- (id)initWithFilePresenter:(id<NSFilePresenter>)filePresenterOrNil;
```

- Zugriffe ausführen

```
- (void)coordinateWritingItemAtURL:(NSURL *)url1  
    options:(NSFileCoordinatorWritingOptions)options1  
    writingItemAtURL:(NSURL *)url2  
    options:(NSFileCoordinatorWritingOptions)options2  
    error:(NSError **)outError  
    byAccessor:  
    (void (^)(NSURL *newURL1, NSURL *newURL2))writer;
```



NSFilePresenter – Ordner

- Änderungen erhalten

```
- (void)presentedSubitemDidChangeAtURL:(NSURL *)url;  
- (void)presentedSubitemAtURL:(NSURL *)oldURL didMoveToURL:(NSURL *)newURL;  
- (void)presentedSubitemAtURL:(NSURL *)url didGainVersion:(NSFileVersion *)version;  
- (void)presentedSubitemDidAppearAtURL:(NSURL *)url;
```

- Zugriffe erhalten

```
-(void)relinquishPresentedItemToReader  
-(void)relinquishPresentedItemToWriter  
-(void)savePresentedItemChangesWithCompletionHandler  
- (void)accommodatePresentedSubitemDeletionAtURL:(NSURL *)url  
    completionHandler:(void (^)(NSError *errorOrNil))completionHandler;
```


NSFilePresenter – Ordner

- Änderungen erhalten

```
- (void)presentedSubitemDidChangeAtURL:(NSURL *)url;  
- (void)presentedSubitemAtURL:(NSURL *)oldURL didMoveToURL:(NSURL *)newURL;  
- (void)presentedSubitemAtURL:(NSURL *)url didGainVersion:(NSFileVersion *)version;  
-(void)presentedSubitemDidAppendURL:(NSURL *)url;
```

- Zugriffe erhalten

```
-(void)relinquishPresentedItemToReader  
-(void)relinquishPresentedItemToWriter  
-(void)savePresentedItemChangesWithCompletionHandler  
-(void)accommodatePresentedSubitemDeletionAtURL:(NSURL *)url  
completionHandler:(void (^)(NSError *errorOrNil))completionHandler;
```

NSFilePresenter – Ordner

- Änderungen an Dateien

- (void)`presentedSubitemDidChangeAtURL:` (NSURL *)url;

- Verschieben von Dateien

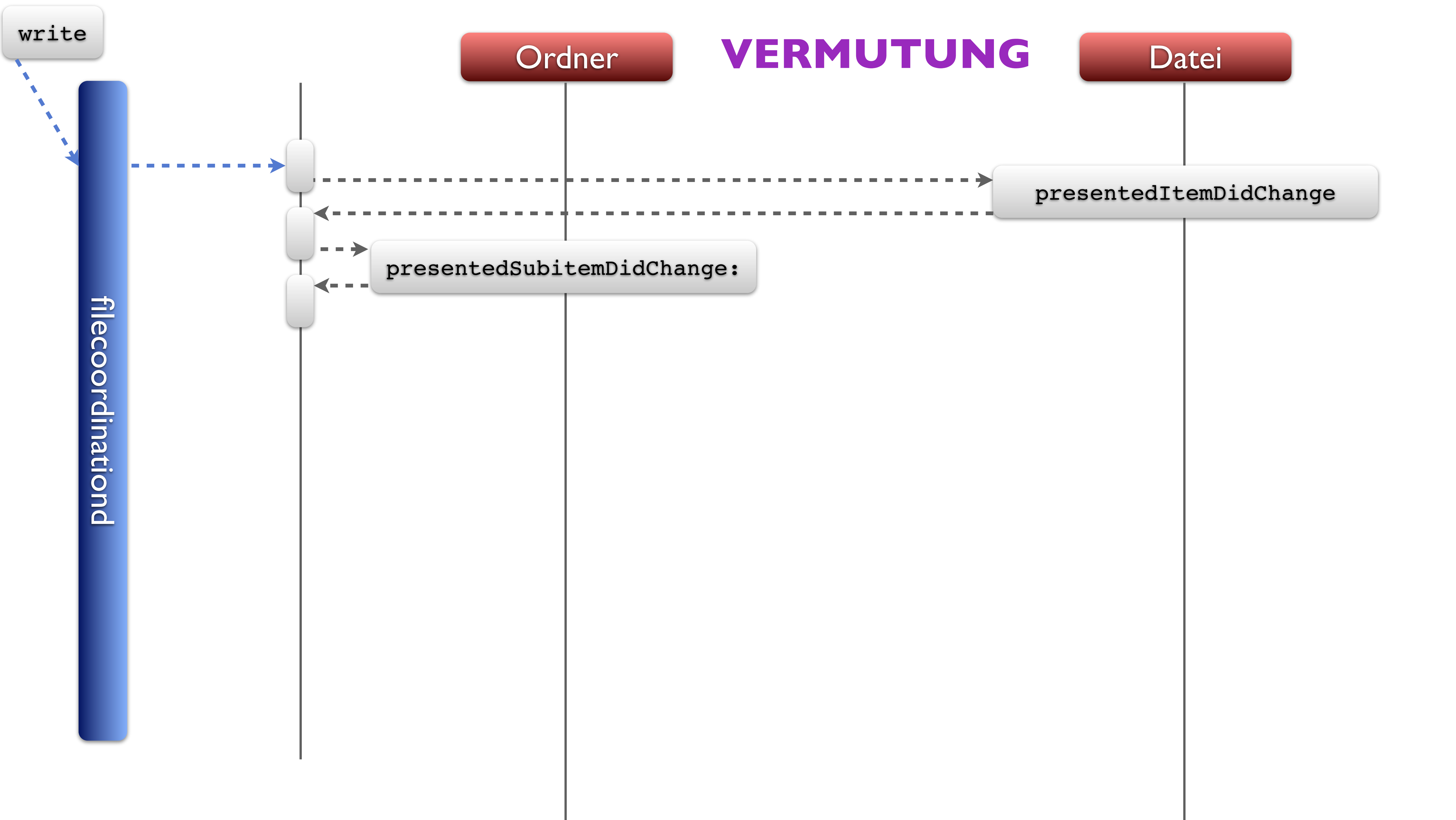
- (void)`presentedSubitemAtURL:` (NSURL *)oldURL `didMoveToURL:` (NSURL *)newURL;

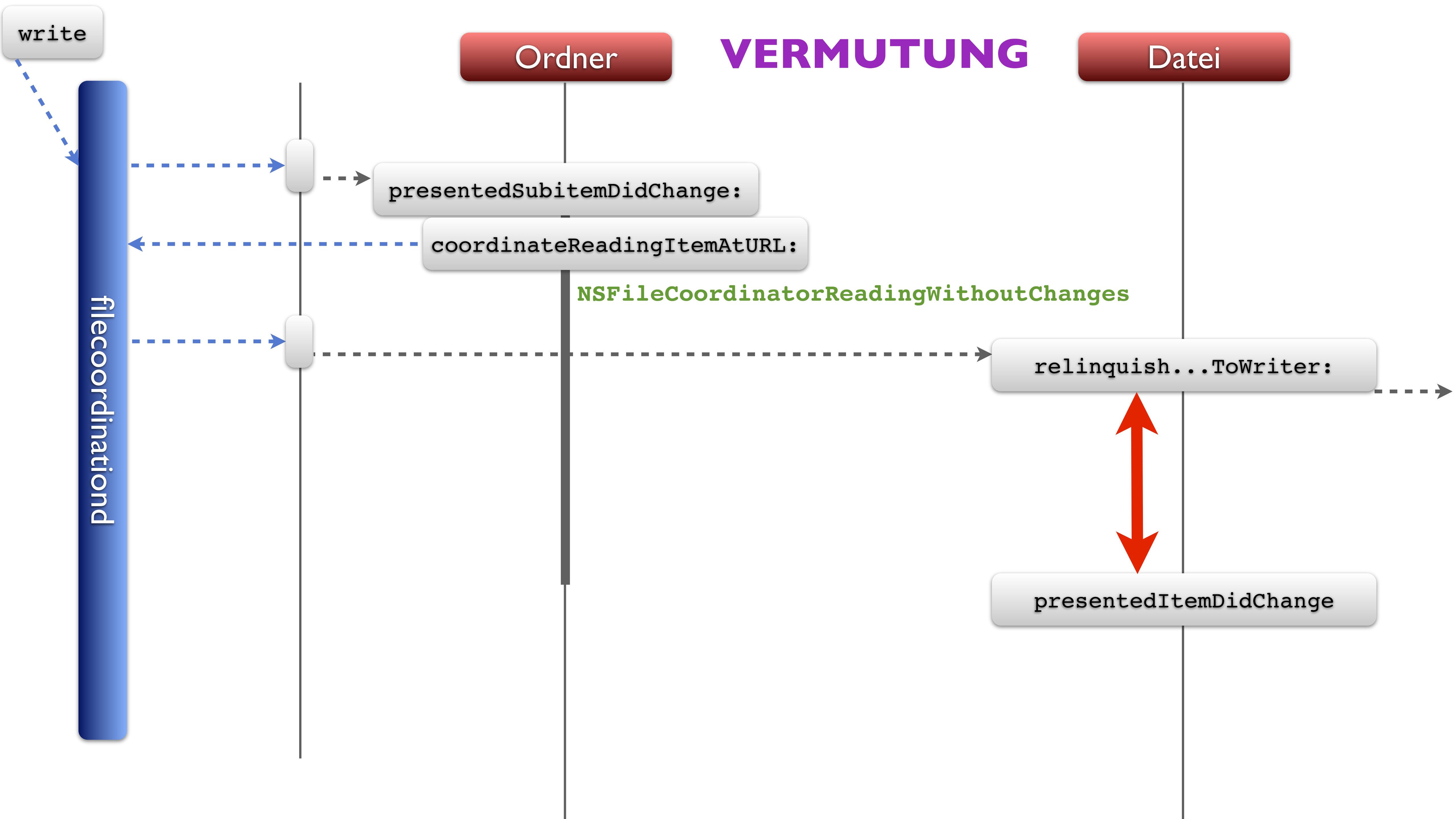
- Anlegen von Dateien

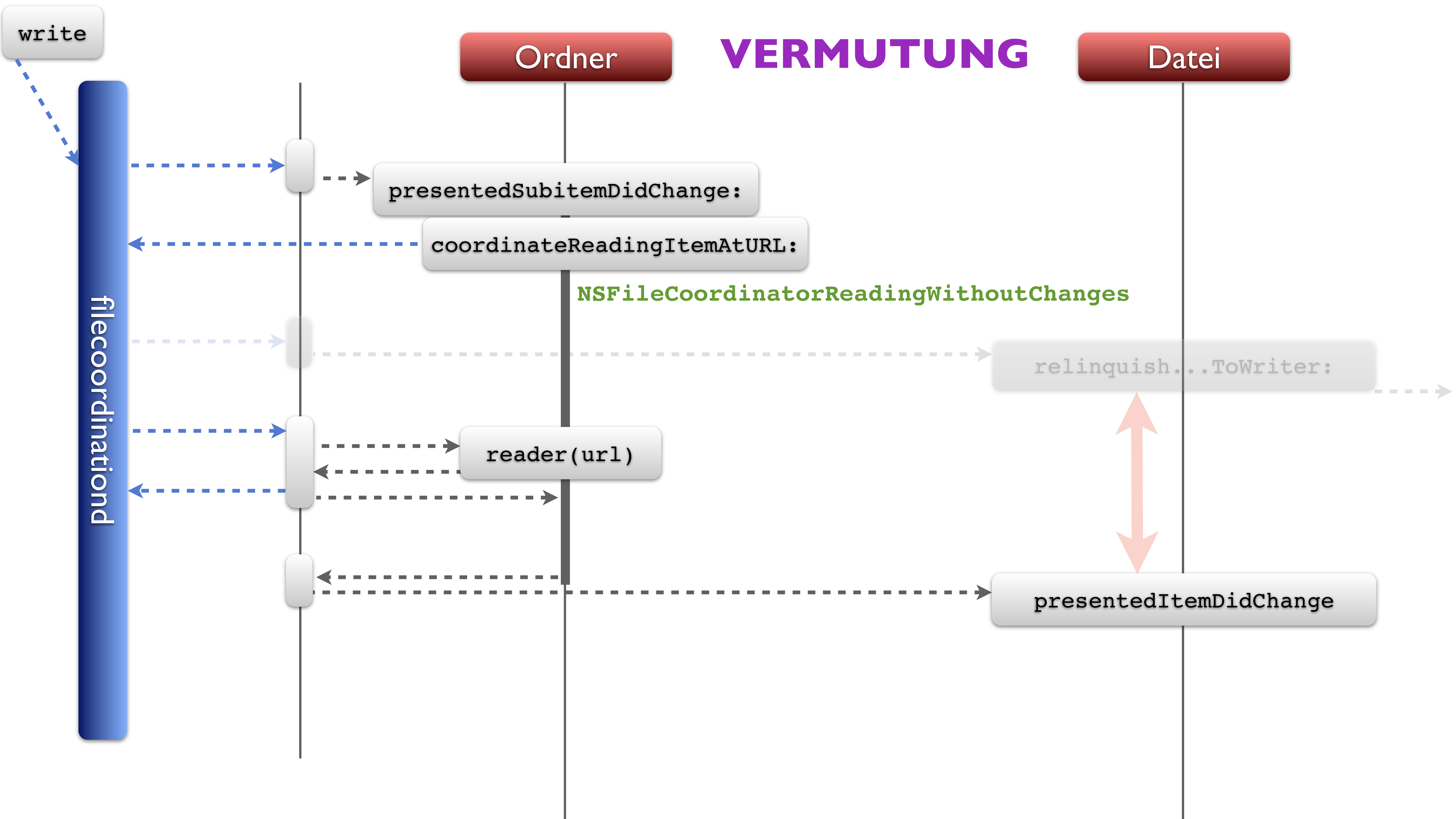
- ~~– (void)`presentedSubitemDidAppearAtURL:` (NSURL *)url;~~

- Löschen von Dateien

- ~~(void)`accommodatePresentedSubitemDeletionAtURL:` (NSURL *)url
`completionHandler:` (void (^)(NSError *errorOrNil))completionHandler;~~







Probleme

iCloud

A full-page background image of a bright blue sky filled with numerous white, fluffy clouds of varying sizes and densities. The clouds are scattered across the entire frame, creating a textured and dynamic visual.

iCloud

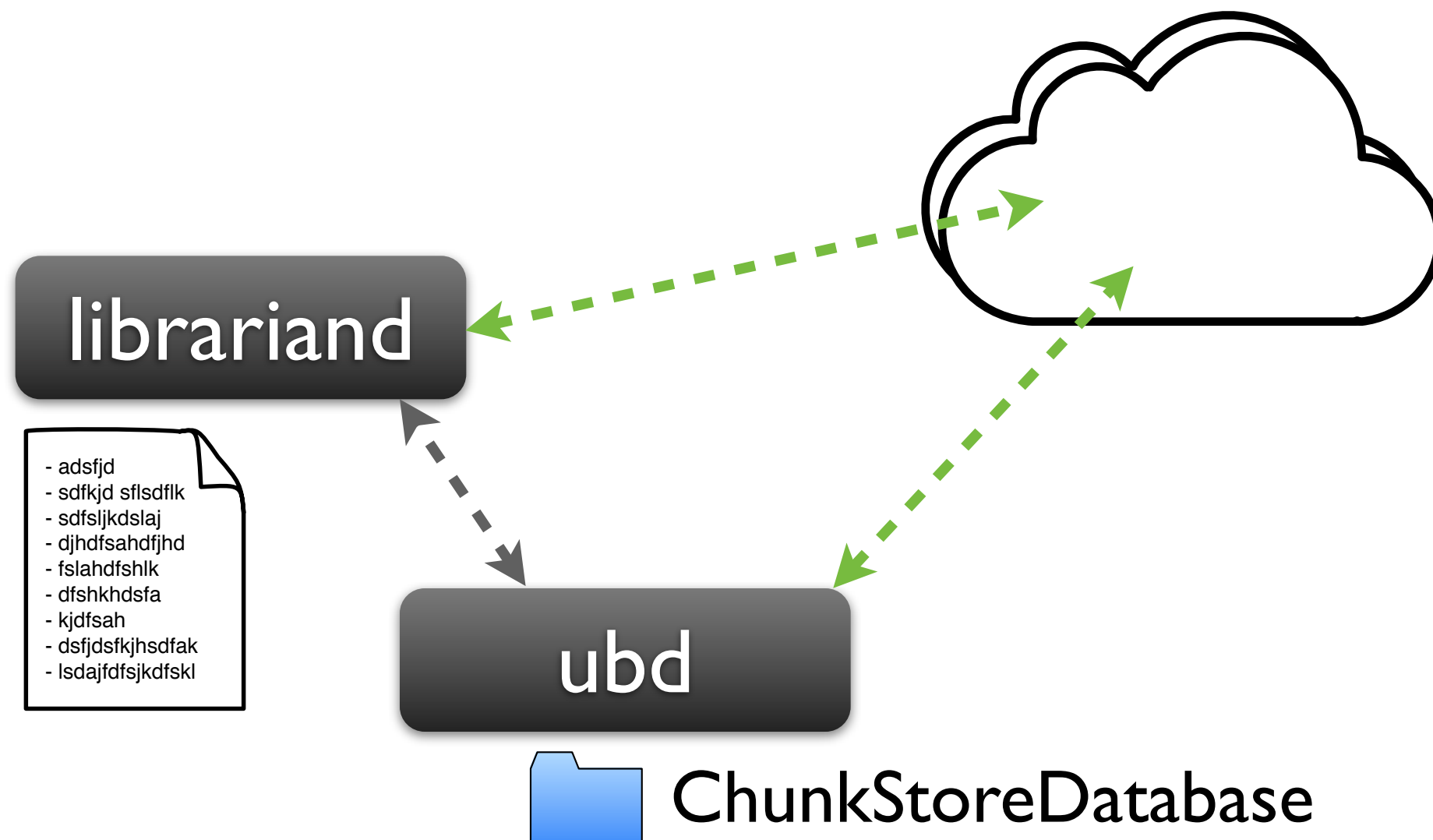


File Coordination

NSFileAccessArbiter
NSFileAccessArbiterProxy
NSFileAccessClaim
NSFileAccessNode
NSFilePresenterProxy
NSFilePresenterRelinquishment
NSFileProviderProxy
NSFileReactorProxy
NSFileReadingClaim
NSFileReadingWritingClaim
NSFileSubarbitrationClaim
NSFileWatcher
NSFileWatcherObservations
NSFileWritingClaim
NSFileWritingWritingClaim

- Buggy, Intransparent
- Schlecht dokumentiert
- Steigt gerne aus
 - Limit auf überwachte iNodes
- Gelegentlich mal ein Deadlock

iCloud Daemons



- Steigen gelegentlich aus
- Gelegentlich tote Dateien
 - Defekter Cache
- Gelegentlich hängende Server
 - Tote Dateien
- Ausführliche Logs: `~/Library/Logs/Ubiquity/`

Stufen der iCloud-Eskalation

Stufen der iCloud-Eskalation

Stufe I

- Programm neu starten

Stufen der iCloud-Eskalation

Stufe 2

- Rechner neu starten

Stufen der iCloud-Eskalation

Stufe 3

- Alle Apps beenden
- iCloud ausschalten
- 1 Minute warten
- iCloud einschalten

Stufen der iCloud-Eskalation

Stufe 4

- Alle Apps beenden
- iCloud ausschalten
- Rechner neu starten
- iCloud einschalten

Stufen der iCloud-Eskalation

Stufe 5

- Alle Apps beenden
- iCloud ausschalten
- Rechner neu starten
- iCloud einschalten
- TextEdit starten
- 1 Minute warten

Stufen der iCloud-Eskalation

Stufe 6

```
$ ubcontrol -x
```

Stufen der iCloud-Eskalation

**Der einzig
richtige Weg**

```
$ ubcontrol -x
```

Stufen der iCloud-Eskalation

Stufe 6 (iOS)

Hard Reset

(Home & Power Button 10 Sekunden drücken)

Stufen der iCloud-Eskalation

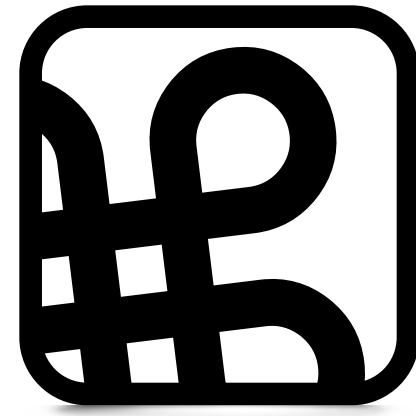
**Der einzige
Weg (iOS)**

Hard Reset

(Home & Power Button 10 Sekunden drücken)

Fragen?

Vielen Dank



Macoun