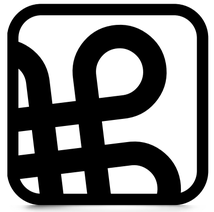
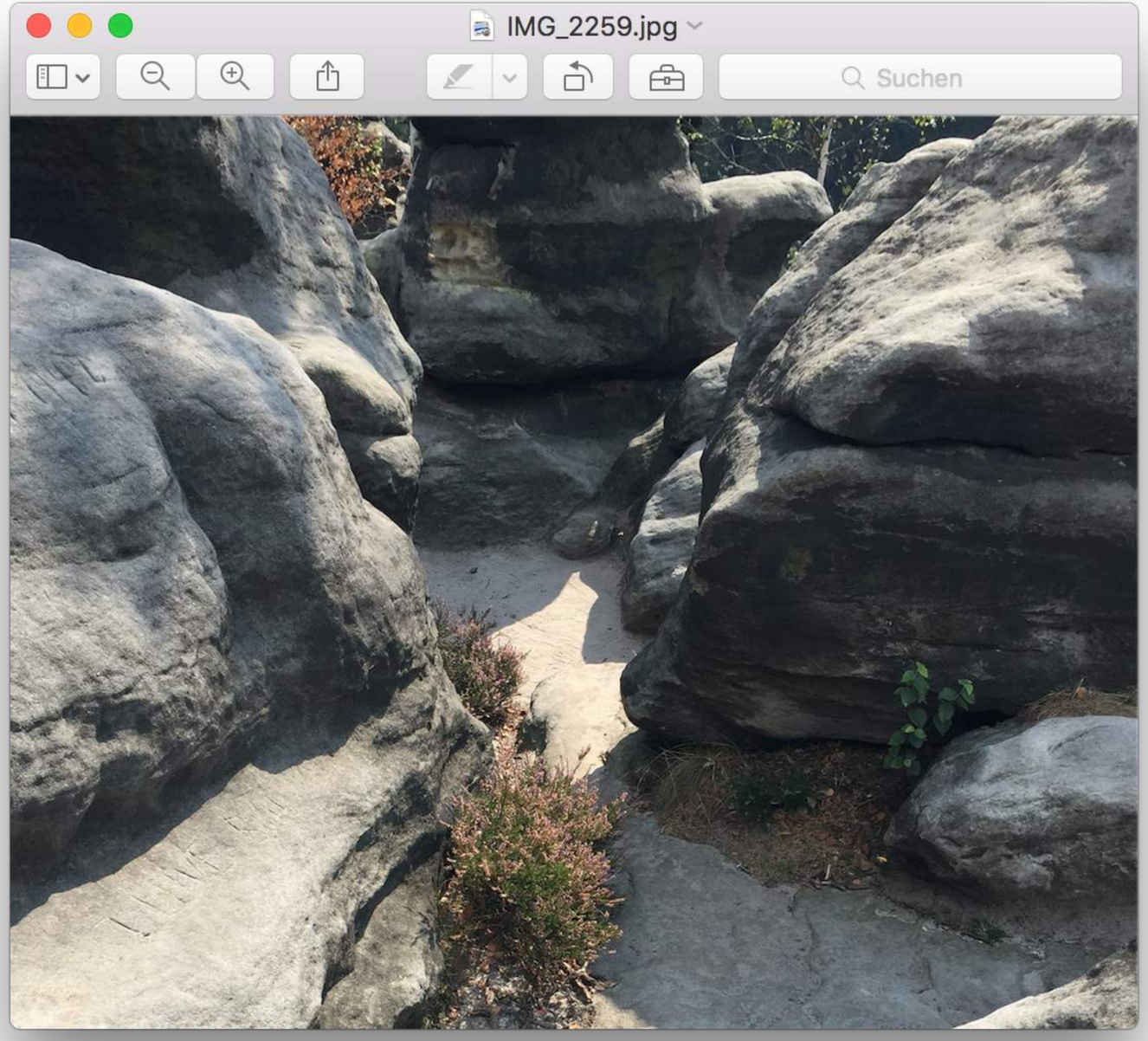


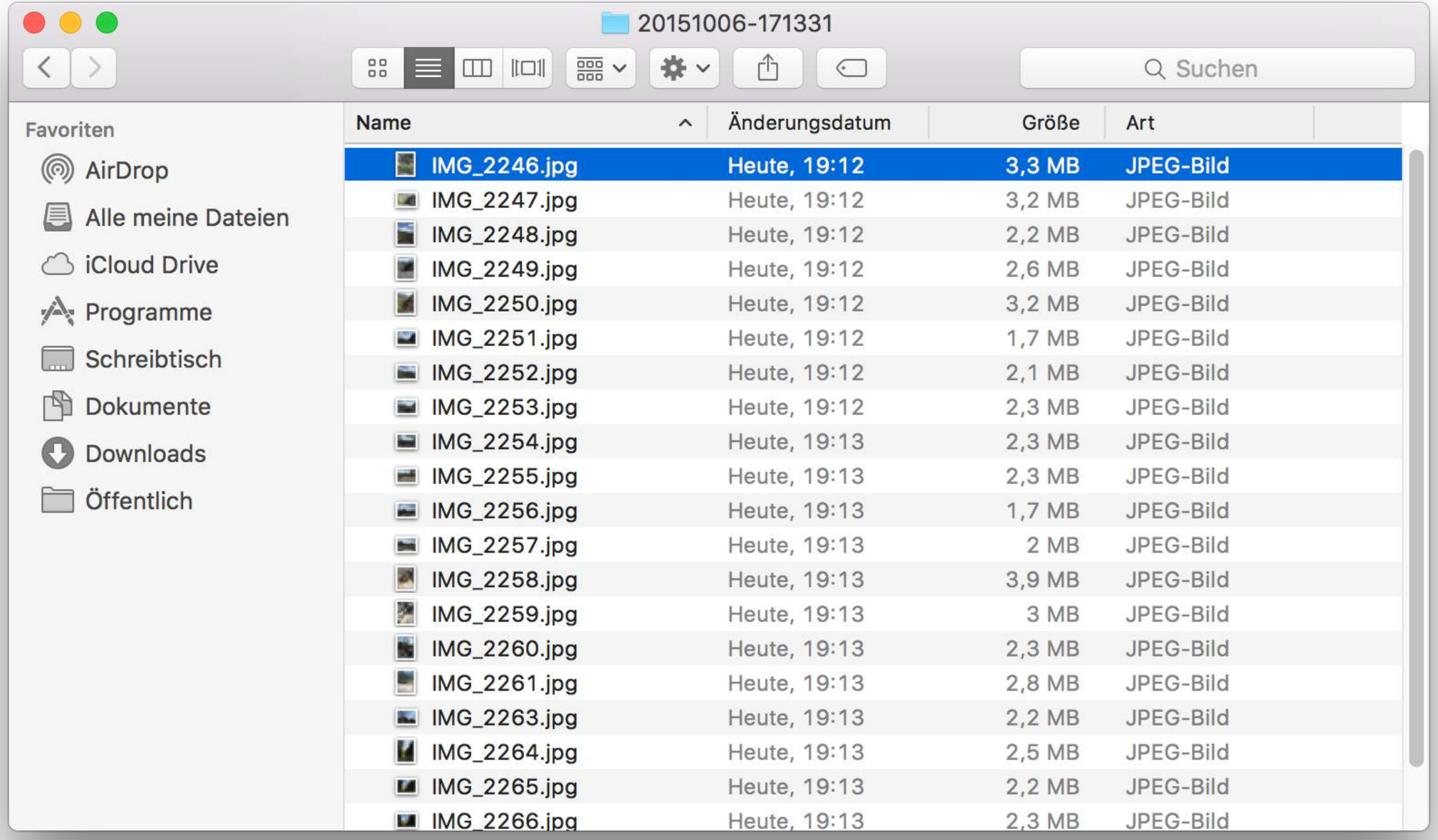
Macoun

Geschichten aus der Shoebox

Friedrich Gräter, @hdrxs

Shoebox?


IMG_2259.jpg
Suchen


20151006-171331
Suchen
Favoriten
AirDrop
Alle meine Dateien
iCloud Drive
Programme
Schreibtisch
Dokumente
Downloads
Öffentlich
Name
Änderungsdatum
Größe
Art
IMG_2246.jpg Heute, 19:12 3,3 MB JPEG-Bild
IMG_2247.jpg Heute, 19:12 3,2 MB JPEG-Bild
IMG_2248.jpg Heute, 19:12 2,2 MB JPEG-Bild
IMG_2249.jpg Heute, 19:12 2,6 MB JPEG-Bild
IMG_2250.jpg Heute, 19:12 3,2 MB JPEG-Bild
IMG_2251.jpg Heute, 19:12 1,7 MB JPEG-Bild
IMG_2252.jpg Heute, 19:12 2,1 MB JPEG-Bild
IMG_2253.jpg Heute, 19:12 2,3 MB JPEG-Bild
IMG_2254.jpg Heute, 19:13 2,3 MB JPEG-Bild
IMG_2255.jpg Heute, 19:13 2,3 MB JPEG-Bild
IMG_2256.jpg Heute, 19:13 1,7 MB JPEG-Bild
IMG_2257.jpg Heute, 19:13 2 MB JPEG-Bild
IMG_2258.jpg Heute, 19:13 3,9 MB JPEG-Bild
IMG_2259.jpg Heute, 19:13 3 MB JPEG-Bild
IMG_2260.jpg Heute, 19:13 2,3 MB JPEG-Bild
IMG_2261.jpg Heute, 19:13 2,8 MB JPEG-Bild
IMG_2263.jpg Heute, 19:13 2,2 MB JPEG-Bild
IMG_2264.jpg Heute, 19:13 2,5 MB JPEG-Bild
IMG_2265.jpg Heute, 19:13 2,2 MB JPEG-Bild
IMG_2266.jpg Heute, 19:13 2,3 MB JPEG-Bild

Momente

Suchen

Fotos

Alben

Alle Fotos

Letzter Import

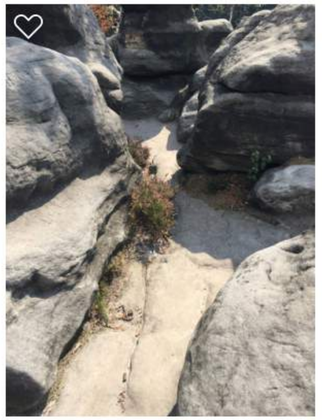
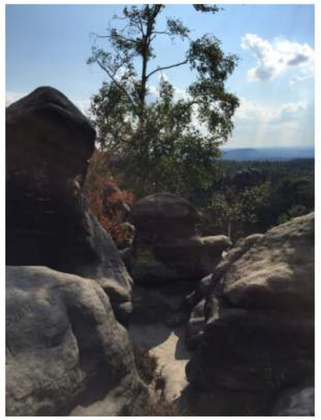
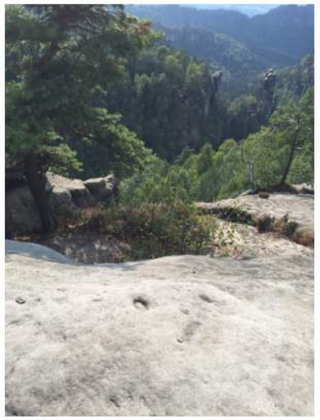
Projekte

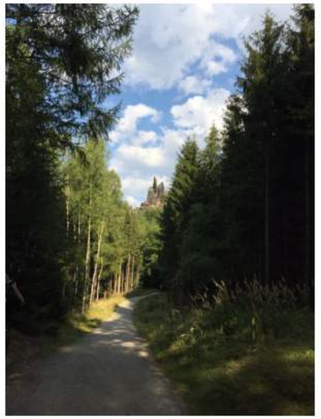
Fotokalender

Sächsisch-Böhmische Schweiz Reinhardtsdorf-Schöna, Sachsen · 23. Aug. ›





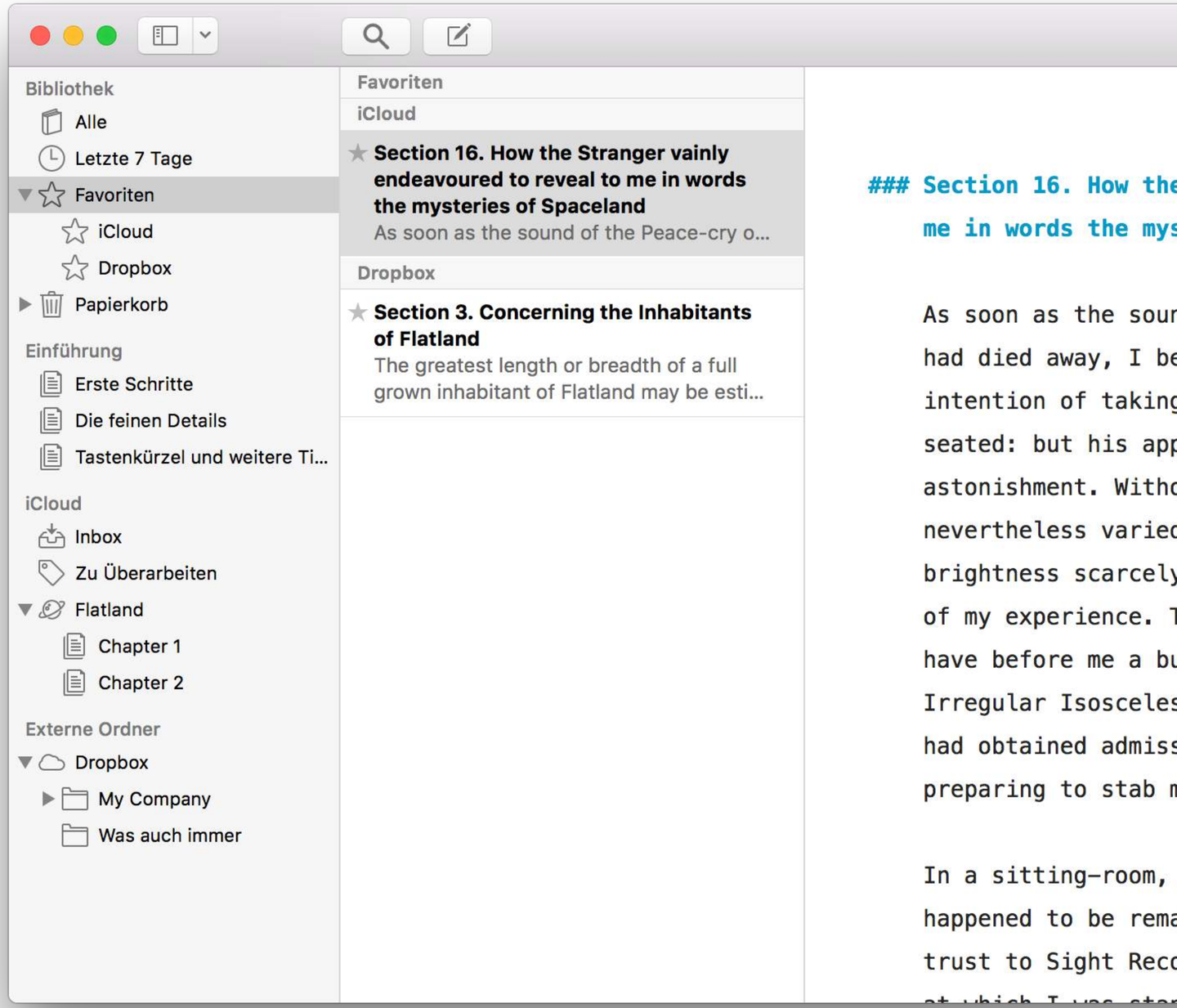






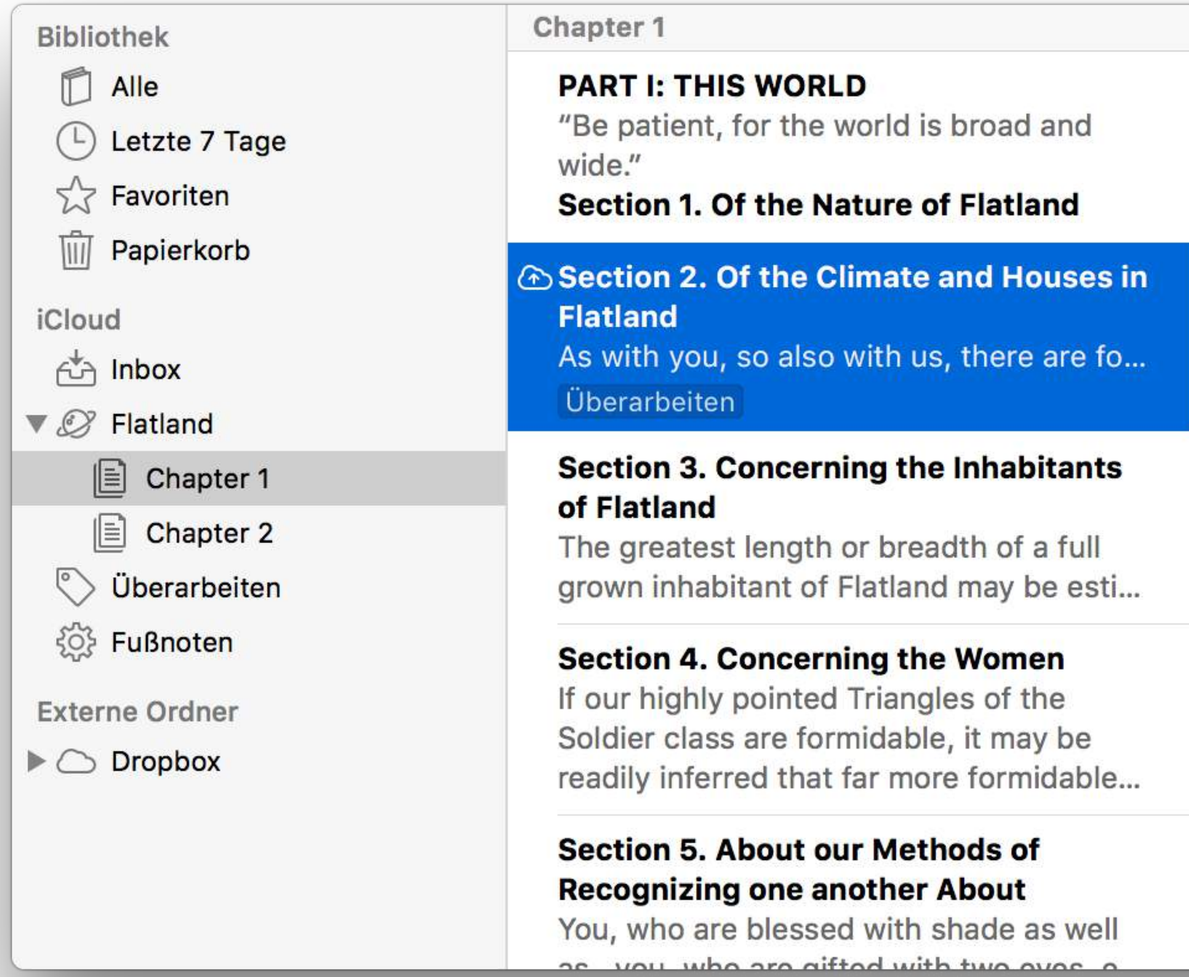
Ulysses

- Schreibprogramm
- Lebenswerk an einem Ort
- Verwaltung speziell für Texte
- Sync über iCloud
- Externe Ordner



Anforderungen

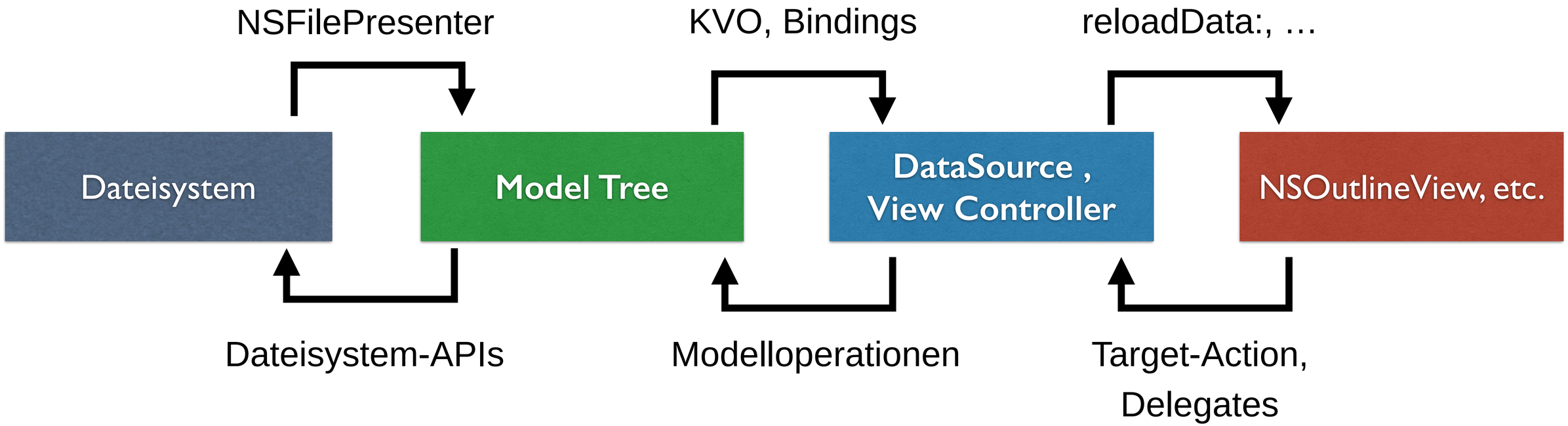
- Virtuelle Ordnerhierarchie
- Spezielle Operationen und Validierung
- Metadaten: Reihenfolge, Icons, …
- Integration mit Sync, Anwendungen, …
- …



Framework?

- **Dokument-basierte Apps:** NSDocument
- **Shoeboxing:**
  - Kein Äquivalent
  - Noch nicht mal NSDocument

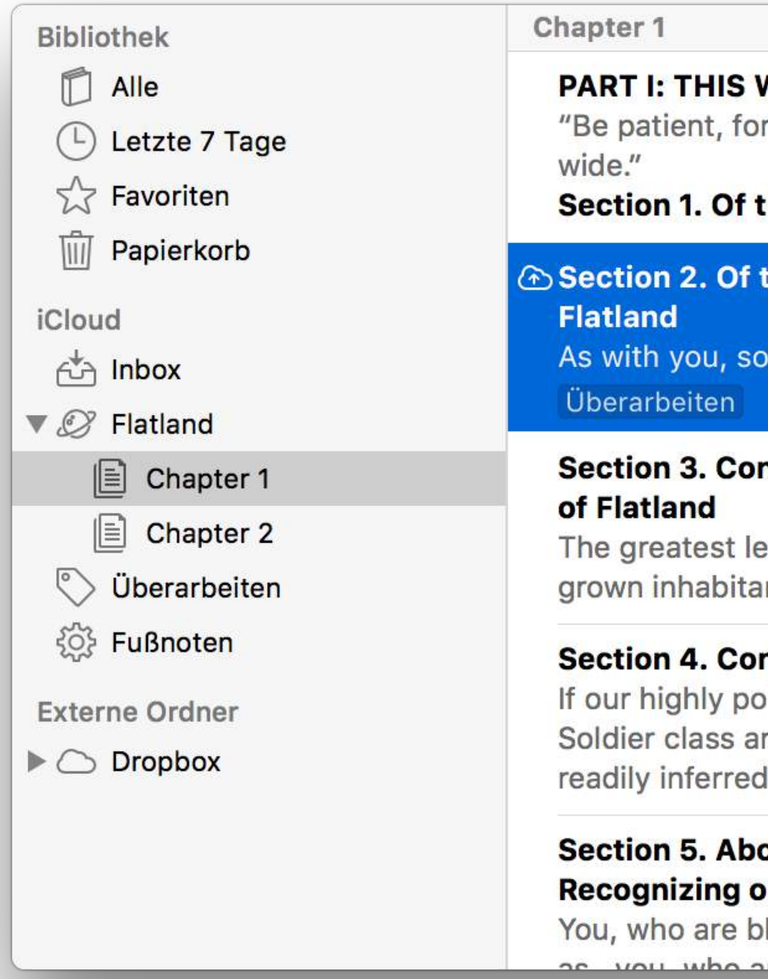
1. Versuch: MVC



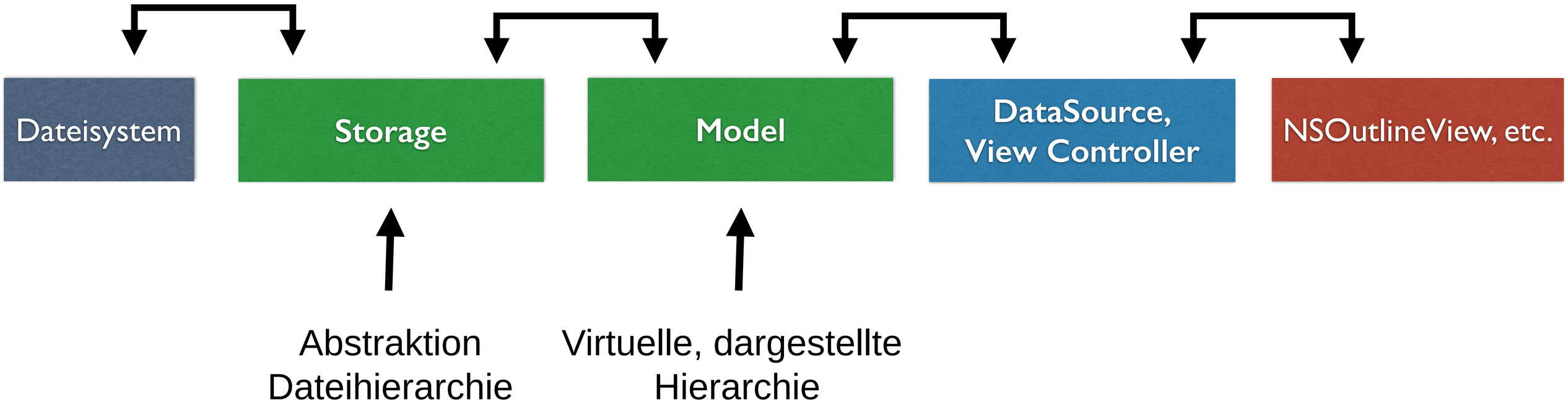
Dateien != Darstellung

- **Dargestelltes Datenmodell**
  - 20 Virtuelle Ordner- / Dateitypen
  - ca. 40 unterschiedliche Operationen mit Validieren

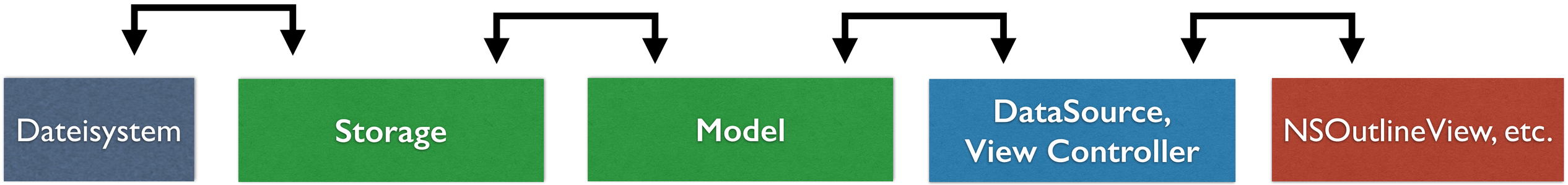
- **Datei-Hierarchie:**
  - Mehrere Backends: Nativ, Markdown, Daedalus, …
  - Dateisystem-Notifikationen, Locking, Indexing, …



2. Versuch

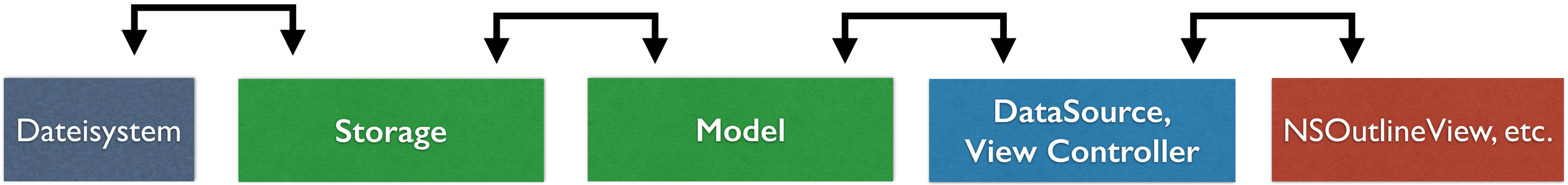

Dateisystem
Storage
Model
DataSource,
View Controller
NSOutlineView, etc.
Abstraktion
Dateihierarchie
Virtuelle, dargestellte
Hierarchie

2. Versuch

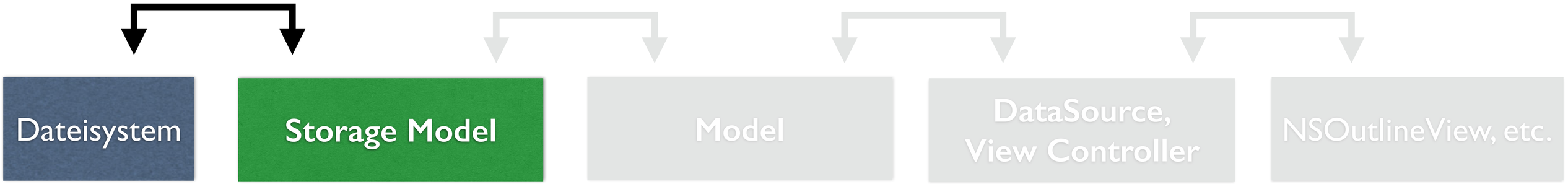


State → State → State

2. Versuch



Asynchron → Asynchron → Asynchron



- **Dateisystem:** Jederzeit Änderungen durch iCloud, andere Apps

- **Storage Model:**
  - Operationen auf veralteten Daten
  - Aktualisierung von ungeschrieben Daten

- Locking zwischen Anwendungen?

File Presentation

- **NSFileCoordinator:** Locks mit Aktualitätsgarantie

```
coordinateReadingItemAtURL: options: error: byAccessor:^{
   /* Block mit geschütztem Zugriff */
 }
```

- **NSFilePresenter:** „Geöffnete Dateien”
  - Änderungsnachrichten: `presentedItemDidChange, …`
  - `savePresentedItemChangesWithCompletionHandler, …`

NSFilePresenter: Nachteile

- Nicht verpflichtend (Nicht-NSDocument, Legacy, Cross-Plattform, …)
- Kooperativ, aber schwer zu benutzen

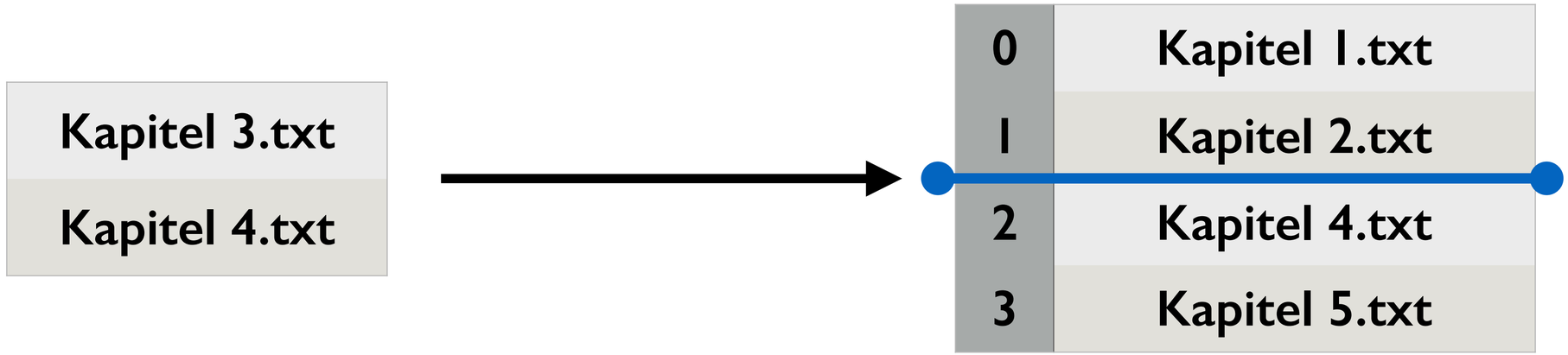
- **Shoeboxing:** Anzahl NSFilePresenter begrenzt
  - Effektiv nur ein Presenter pro Dateibaum
  - Keine Aktualitätsgarantie

Locks vermeiden

- Nicht-persistenten State vermeiden
  - Änderungen möglichst schnell persistieren
  - Speicherzustand immer aus dem Dateisystem herleiten
- Merge-Algorithmus zwischen Speicher und Dateisystem
- Algorithmen auf kurzzeitiges Locking anpassen

Beispiel: Sortiertes Kopieren

Beispiel: Sortiertes Kopieren



- Parallele Operationen innerhalb der App
- Änderungen durch andere Apps, iCloud

Beispiel: Sortiertes Kopieren

Kapitel 2.txt	*Sentinel 1*
Kapitel 3.txt	*Sentinel 2*

0	**Kapitel 1.txt**
1	**Kapitel 2.txt**
2	**Sentinel 1**
3	**Sentinel 2**
4	**Kapitel 4.txt**
5	**Kapitel 5.txt**

Lock: Sentinels einfügen

Beispiel: Sortiertes Kopieren

Kapitel 2.txt	*Sentinel 1*
Kapitel 3.txt	*Sentinel 2*

0	Kapitel 1.txt
1	Kapitel 2.txt
2	Sentinel 1
3	Kapitel 3.txt
4	Kapitel 4.txt
5	Kapitel 5.txt

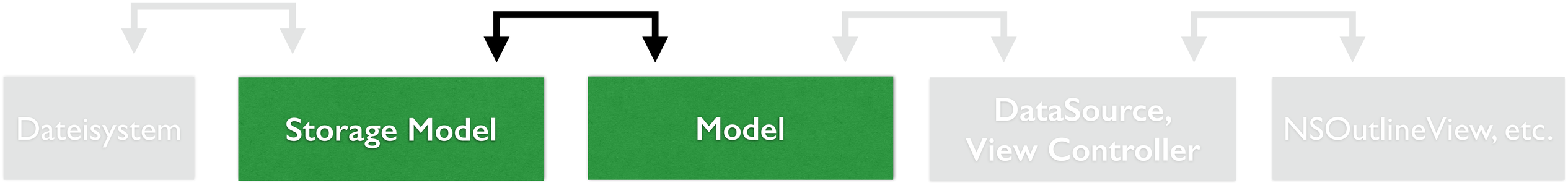
Lock: Zieldatei; Sentinel ersetzen

Beispiel: Sortiertes Kopieren

Kapitel 2.txt	*Sentinel 1*
Kapitel 3.txt	*Sentinel 2*

0	Kapitel 1.txt
1	Kapitel 2.txt
2	Kapitel 2 – 1.txt
3	Kapitel 3.txt
4	Kapitel 4.txt
5	Kapitel 5.txt

Lock: Zieldatei; Sentinel ersetzen



- **Storage Model:** Aktualisierung im Hintergrund
  - NSFilePresenter-Notifications
  - Dateioperationen

- **Model Layer:** Aktualisierung auf Main-Threads
  - Bindings zu UI-Komponenten
  - KVO, Delegation, Threads und Locks: 😱 😱 😱

KVO + Locks

```
dispatch_sync(someQueue, ^{
   self.displayName = someDisplayName;
   …
});
```

observeValueForKeyPath:

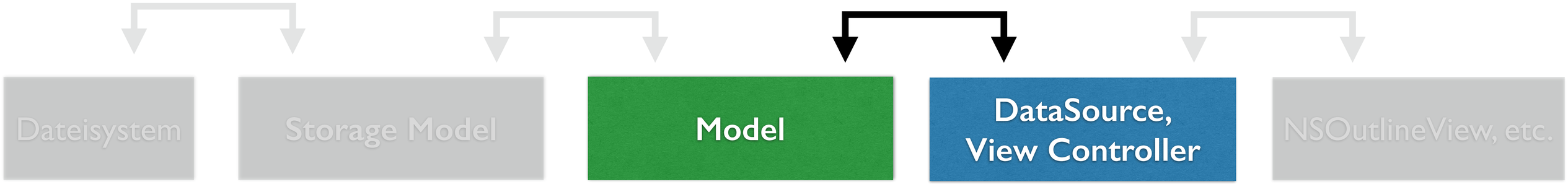
Deadlock

```
dispatch_sync(someQueue, ^{
   …
});
```

Asynchrone Observierung

- Storage Layer aktualisiert im Hintergrund
- Properties mit speziellem Kontext observiert
- Standard-Handler vollzieht Änderung auf Main Queue nach

```
- (void)observeValueForKeyPath:(NSString *)keyPath ofObject:(id)object
  …
   if (context == MirroredPropertyContext) {
    dispatch_async(dispatch_get_main_queue(), ^{
       id newValue = [object valueForKey: keyPath];
       [self setValue:newValue forKey:keyPath];
    });
```



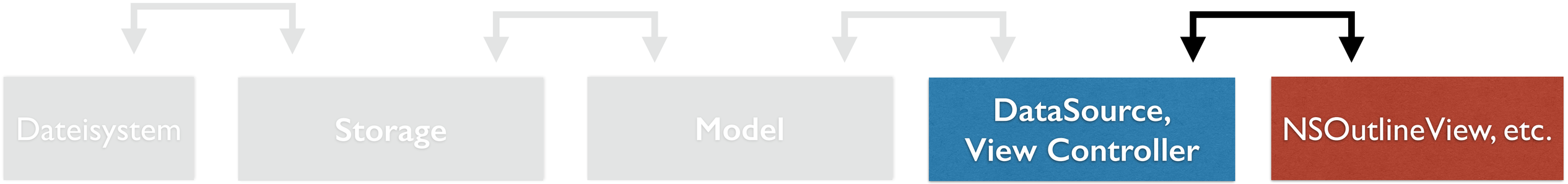
- **Modell:** Instanzen (Verschieben, Kopieren etc.)

- **UI:** Referenziert Modell-Objekte
  - Selektionen, Expansionen, Scroll-Positionen
  - State Restoration
  - Lazy-Loading

Persistent Objects

- **Persistente Referenzen:** z.B. UUIDs oder Aliase
- **Data Source**: persistentSelectedNodes, persistentExpandedNodes

- **Modell-Update:**
  - Inhalt der Outline-View aktualisieren
  - Persistente Referenzen auflösen
  - Selektion, Expansion aktualisieren



- Weitere Zwischenrepräsentation für Outline View
- 1:1 Abbildung des Modells
- Animationen vs. vollständiges Neuladen

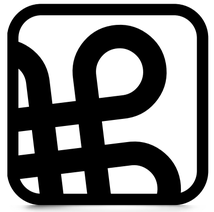
Zusammenfassung

- **Shoeboxing:** Dokumente anwendungsorientiert verwalten

- Keine Framework-Unterstützung
- Sorgfältige Modellierung erforderlich
- Asynchrone Programmierung: Grenzen von Cocoa

- **Blick über den Tellerrand:** Reactive Programming, …

Fragen?



Macoun