

# CKSyncEngine

Erste Erfahrungen

Friedrich Ruynat, 14.11.2024

# CloudKit Recap

# CloudKit Recap

- CKRecord:

# CloudKit Recap

- CKRecord:
  - Dictionary mit PList-Values (+ CKAsset, + CKReference)

# CloudKit Recap

- CKRecord:
  - Dictionary mit PList-Values (+ CKAsset, + CKReference)
  - parent-Relationship

# CloudKit Recap

- CKRecord:
  - Dictionary mit PList-Values (+ CKAsset, + CKReference)
  - parent-Relationship
  - Konflikterkennung über *changeTag*

# CloudKit Recap

- CKRecord:
  - Dictionary mit PList-Values (+ CKAsset, + CKReference)
  - parent-Relationship
  - Konflikterkennung über *changeTag*
- CKSubscription

# CloudKit Recap

- CKRecord:
  - Dictionary mit PList-Values (+ CKAsset, + CKReference)
  - parent-Relationship
  - Konflikterkennung über *changeTag*
- CKSubscription
  - Observierung der Datenbank



# CloudKit Recap

- CKRecord:
  - Dictionary mit PList-Values (+ CKAsset, + CKReference)
  - parent-Relationship
  - Konflikterkennung über *changeTag*
- CKSubscription
  - Observierung der Datenbank
  - Cursor um Delta herunterzuladen

# CloudKit Recap

- CKRecord:
  - Dictionary mit PList-Values (+ CKAsset, + CKReference)
  - parent-Relationship
  - Konflikterkennung über *changeTag*
- CKSubscription
  - Observierung der Datenbank
  - Cursor um Delta herunterzuladen
- Operationsbasierte-API: *modifyRecords, fetchRecords, ...*

# Sync

# Sync

- Szenario: Addressbuch

# Sync

- Szenario: Addressbuch
  - Lokale Kopie auf dem Gerät

# Sync

- Szenario: Addressbuch
  - Lokale Kopie auf dem Gerät
  - Zentrale Wahrheit in CloudKit

# Sync

- Szenario: Addressbuch
  - Lokale Kopie auf dem Gerät
  - Zentrale Wahrheit in CloudKit
  - Hochladen bei lokalen Änderungen

# Sync

- Szenario: Addressbuch
  - Lokale Kopie auf dem Gerät
  - Zentrale Wahrheit in CloudKit
  - Hochladen bei lokalen Änderungen
  - Herunterladen von Remote-Änderungen



# Sync

- Szenario: Addressbuch
  - Lokale Kopie auf dem Gerät
  - Zentrale Wahrheit in CloudKit
  - Hochladen bei lokalen Änderungen
  - Herunterladen von Remote-Änderungen
  - Merge von Konflikten

# Probleme beim Sync

# Probleme beim Sync

- Merge-Logik

# Probleme beim Sync

- Merge-Logik
- Wann starten Upload/Download-Operationen?

# Probleme beim Sync

- Merge-Logik
- Wann starten Upload/Download-Operationen?
  - Neue Daten verfügbar

# Probleme beim Sync

- Merge-Logik
- Wann starten Upload/Download-Operationen?
  - Neue Daten verfügbar
  - Netzwerk / Mobile Daten / Batterie bereit

# Probleme beim Sync

- Merge-Logik
- Wann starten Upload/Download-Operationen?
  - Neue Daten verfügbar
  - Netzwerk / Mobile Daten / Batterie bereit
  - CloudKit Transfer Quota

# Probleme beim Sync

- Merge-Logik
- Wann starten Upload/Download-Operationen?
  - Neue Daten verfügbar
  - Netzwerk / Mobile Daten / Batterie bereit
  - CloudKit Transfer Quota
  - Batching: Maximale Request-Länge



# Probleme beim Sync

- Merge-Logik
- Wann starten Upload/Download-Operationen?
  - Neue Daten verfügbar
  - Netzwerk / Mobile Daten / Batterie bereit
  - CloudKit Transfer Quota
  - Batching: Maximale Request-Länge
- Komplexe Fehlerbehandlung

# CKSyncEngine

# CKSyncEngine

- Neue API seit 2023

# CKSyncEngine

- Neue API seit 2023
- Eher: **CKOperationController**

# CKSyncEngine

- Neue API seit 2023
- Eher: **CKOperationController**
- Keine Merge-Logik

# CKSyncEngine

- Neue API seit 2023
- Eher: **CKOperationController**
- Keine Merge-Logik
- Steuert Up-/Download-Operationen

# CKSyncEngine

- Neue API seit 2023
- Eher: **CKOperationController**
- Keine Merge-Logik
- Steuert Up-/Download-Operationen
- Private-Database

# CKSyncEngine



# CKSyncEngine

- State

# CKSyncEngine

- **State**
  - Pending Changes (Record.ID + Save/Delete)

# CKSyncEngine

- **State**
  - Pending Changes (Record.ID + Save/Delete)
  - Letztes Server Change Token

# CKSyncEngine

- **State**
  - Pending Changes (Record.ID + Save/Delete)
  - Letztes Server Change Token
- **Delegate:** Event-Handling und Datenquelle

# CKSyncEngine

- **State**
  - Pending Changes (Record.ID + Save/Delete)
  - Letztes Server Change Token
- **Delegate:** Event-Handling und Datenquelle
  - Datenquelle für Pending Records

# CKSyncEngine

- **State**
  - Pending Changes (Record.ID + Save/Delete)
  - Letztes Server Change Token
- **Delegate:** Event-Handling und Datenquelle
  - Datenquelle für Pending Records
  - Event-Handling:

# CKSyncEngine

- **State**
  - Pending Changes (Record.ID + Save/Delete)
  - Letztes Server Change Token
- **Delegate:** Event-Handling und Datenquelle
  - Datenquelle für Pending Records
  - Event-Handling:
    - Neue Records vom Server

# CKSyncEngine

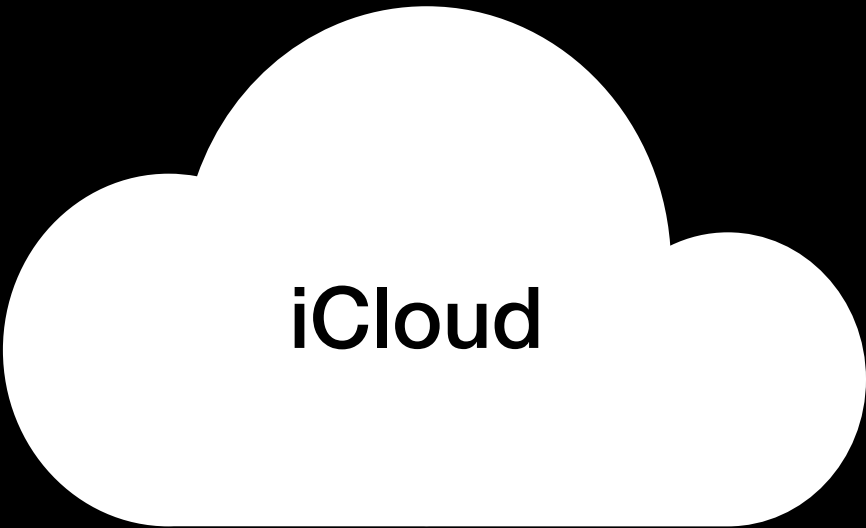
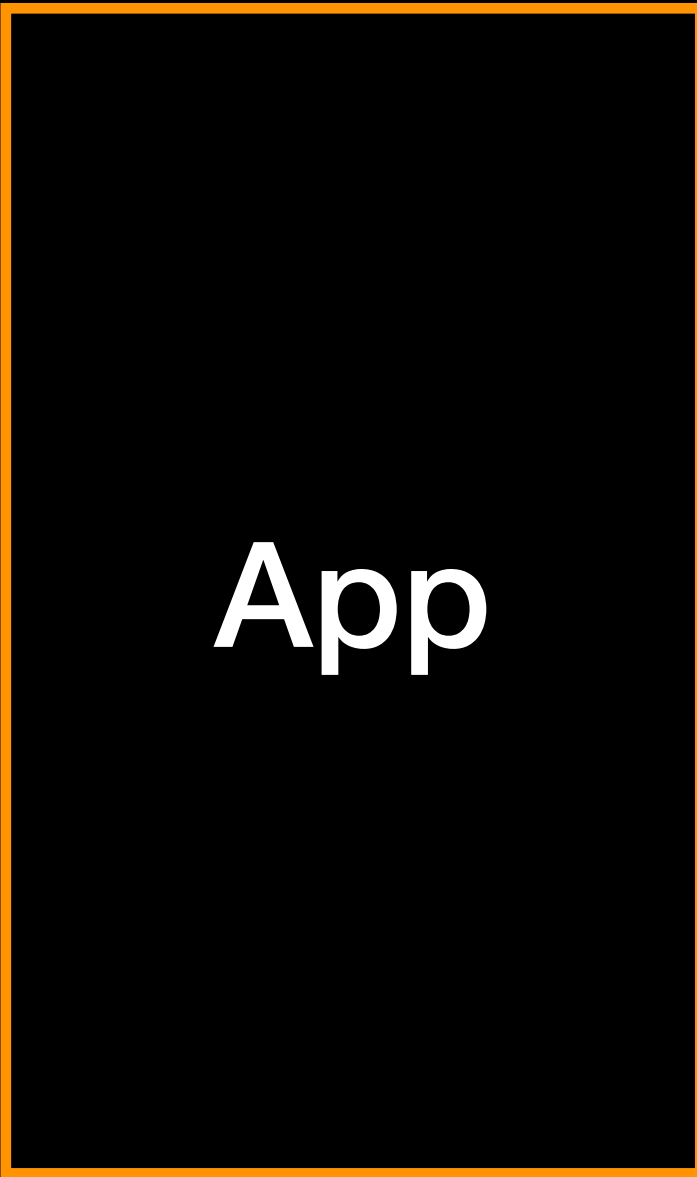
- **State**
  - Pending Changes (Record.ID + Save/Delete)
  - Letztes Server Change Token
- **Delegate:** Event-Handling und Datenquelle
  - Datenquelle für Pending Records
  - Event-Handling:
    - Neue Records vom Server
    - Fehlgeschlagene Uploads (z.B. Konflikte)



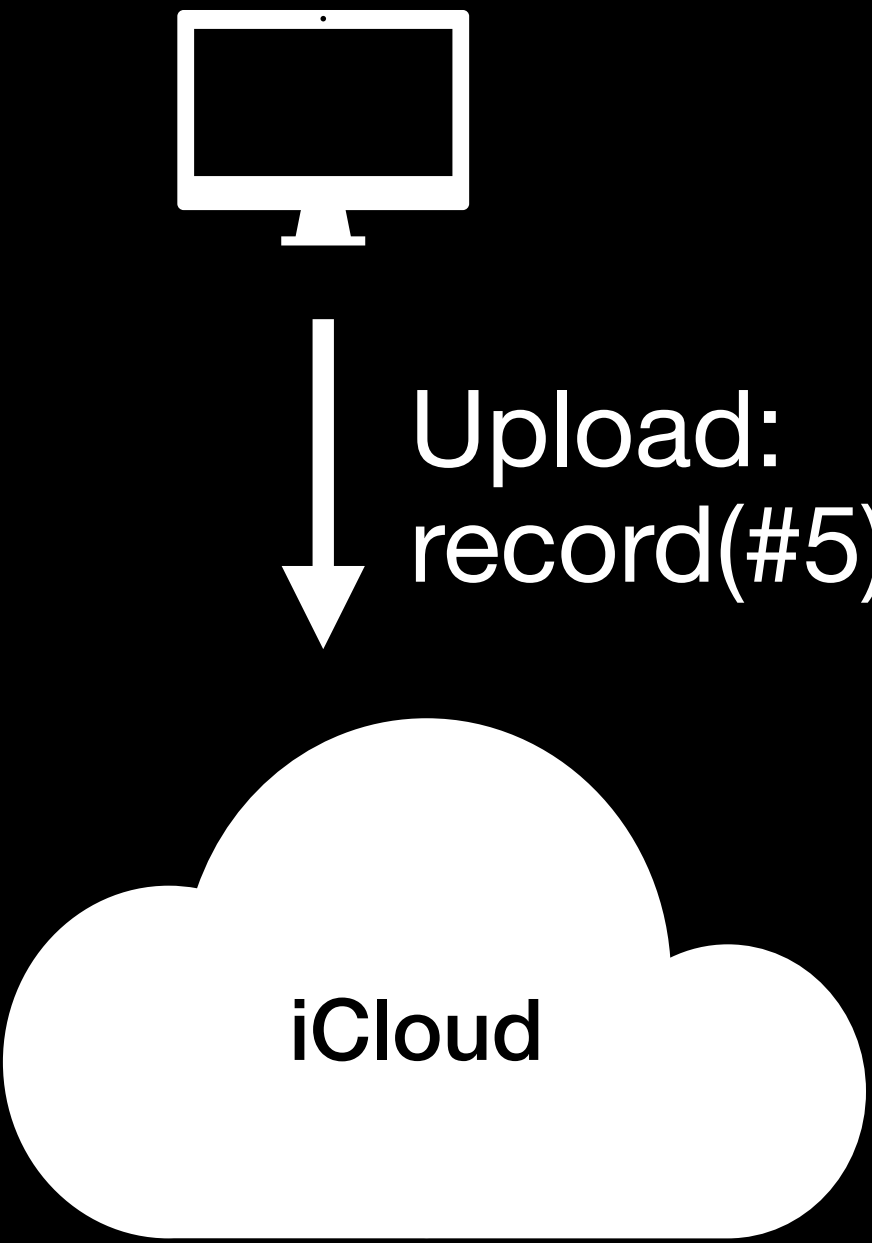
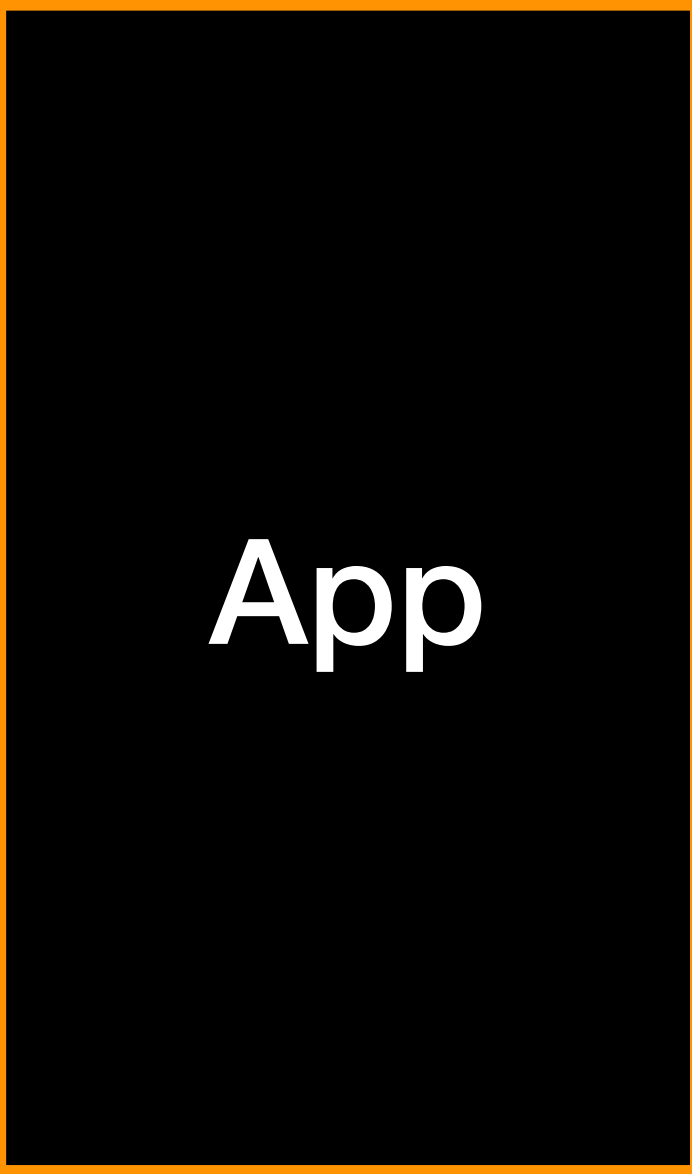
# CKSyncEngine

- **State**
  - Pending Changes (Record.ID + Save/Delete)
  - Letztes Server Change Token
- **Delegate:** Event-Handling und Datenquelle
  - Datenquelle für Pending Records
  - Event-Handling:
    - Neue Records vom Server
    - Fehlgeschlagene Uploads (z.B. Konflikte)
    - aber auch: Account-Änderung, State-Serialisierung

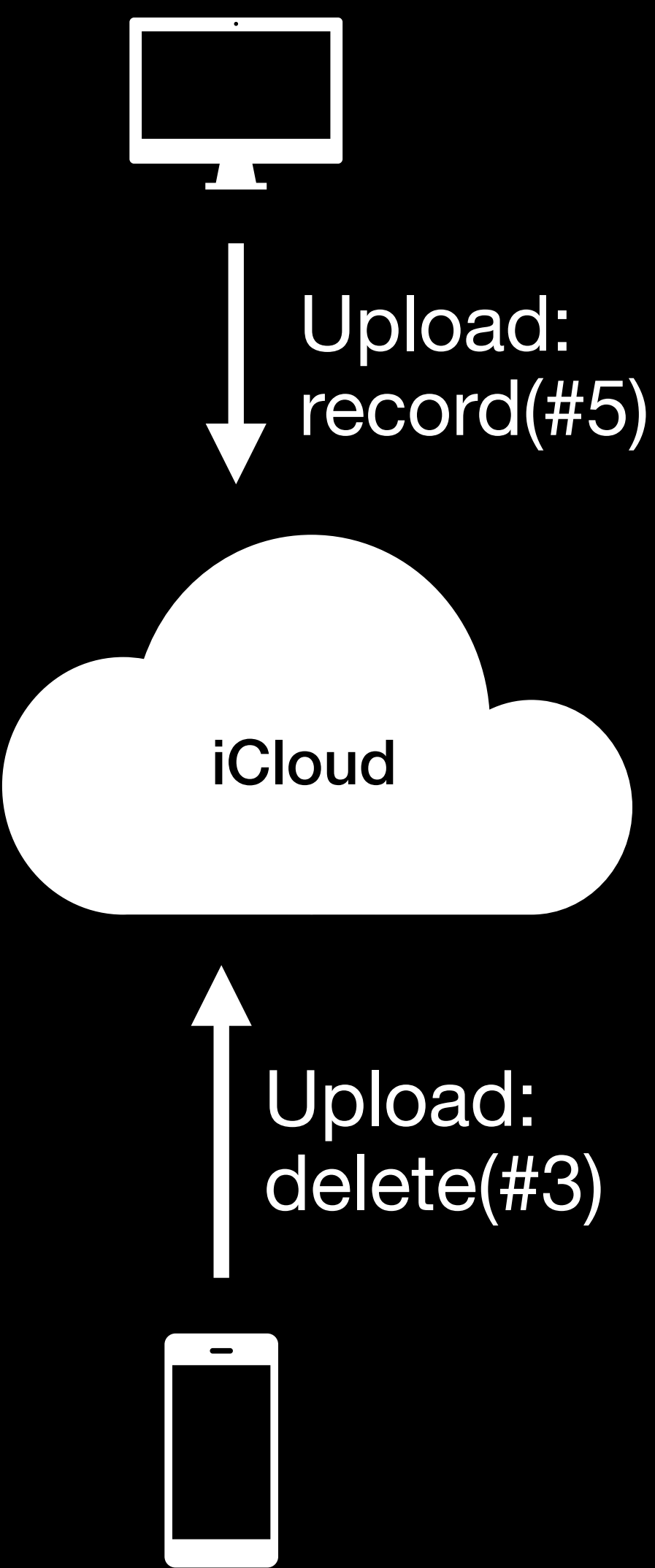
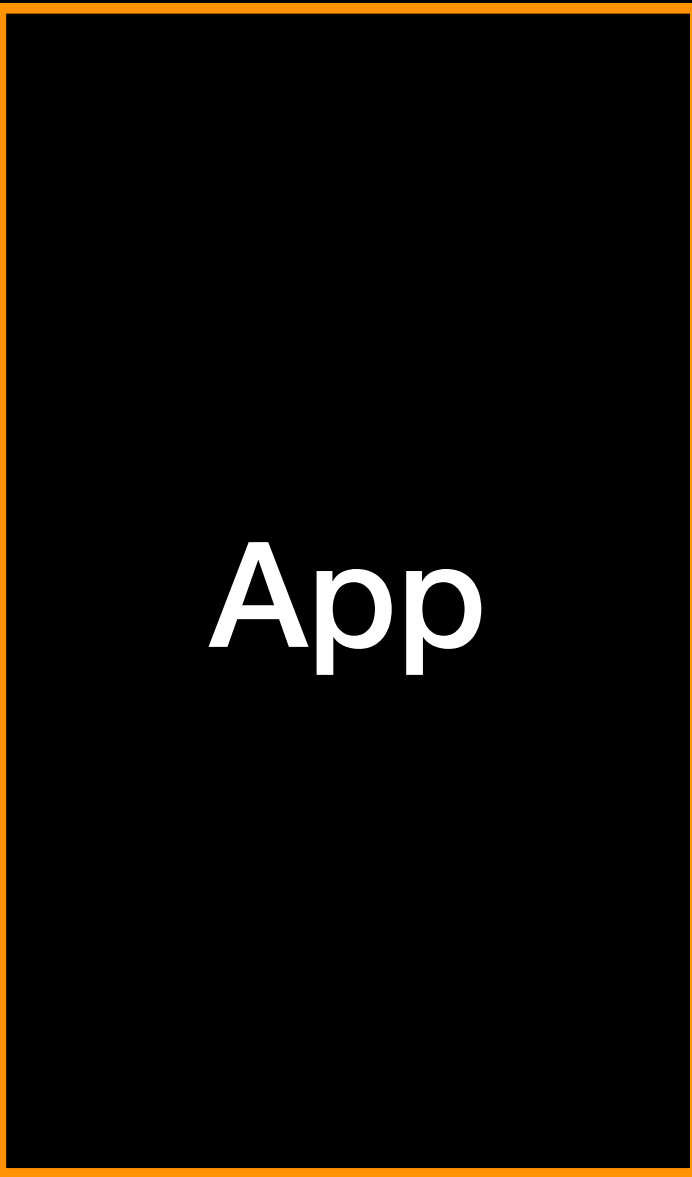
# Download



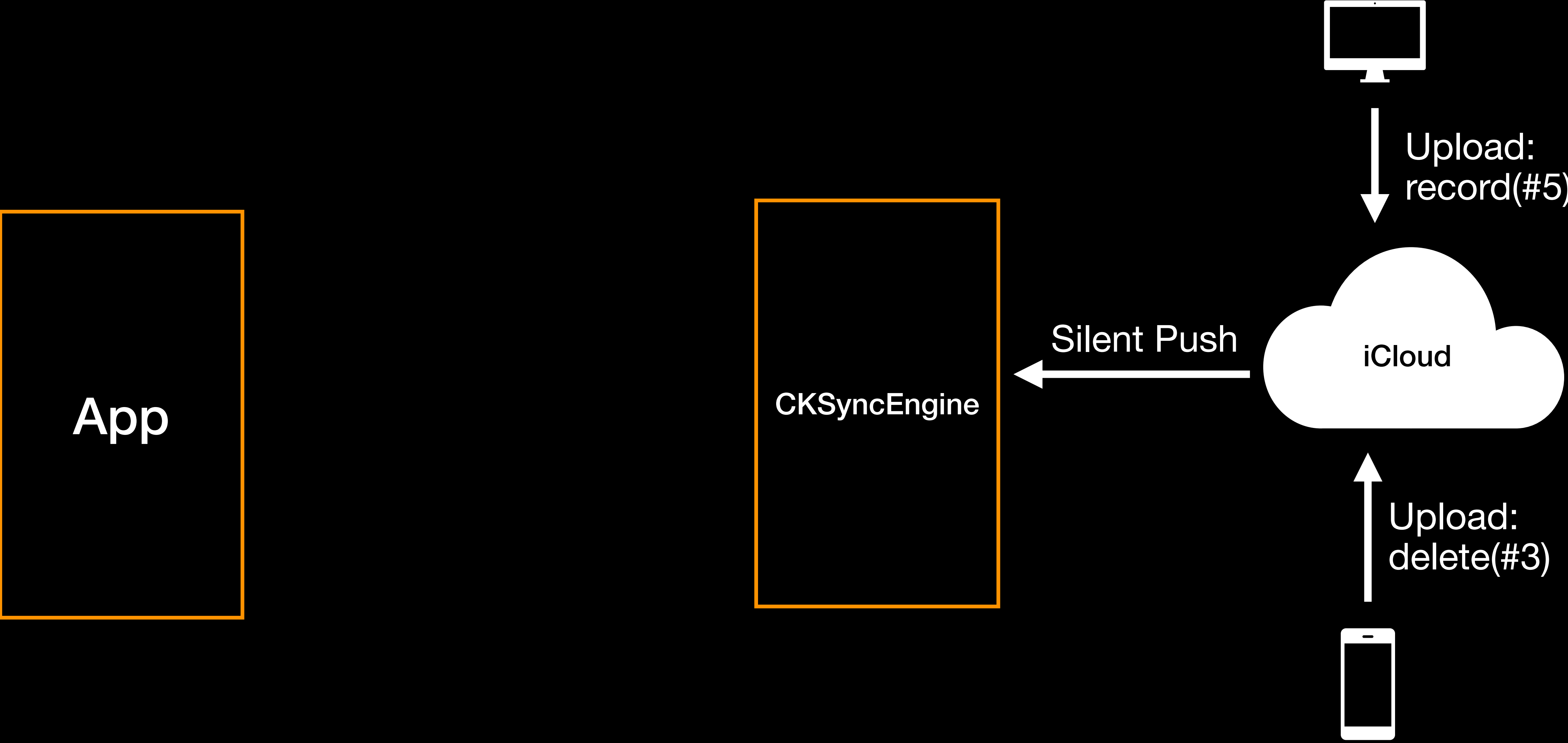
# Download



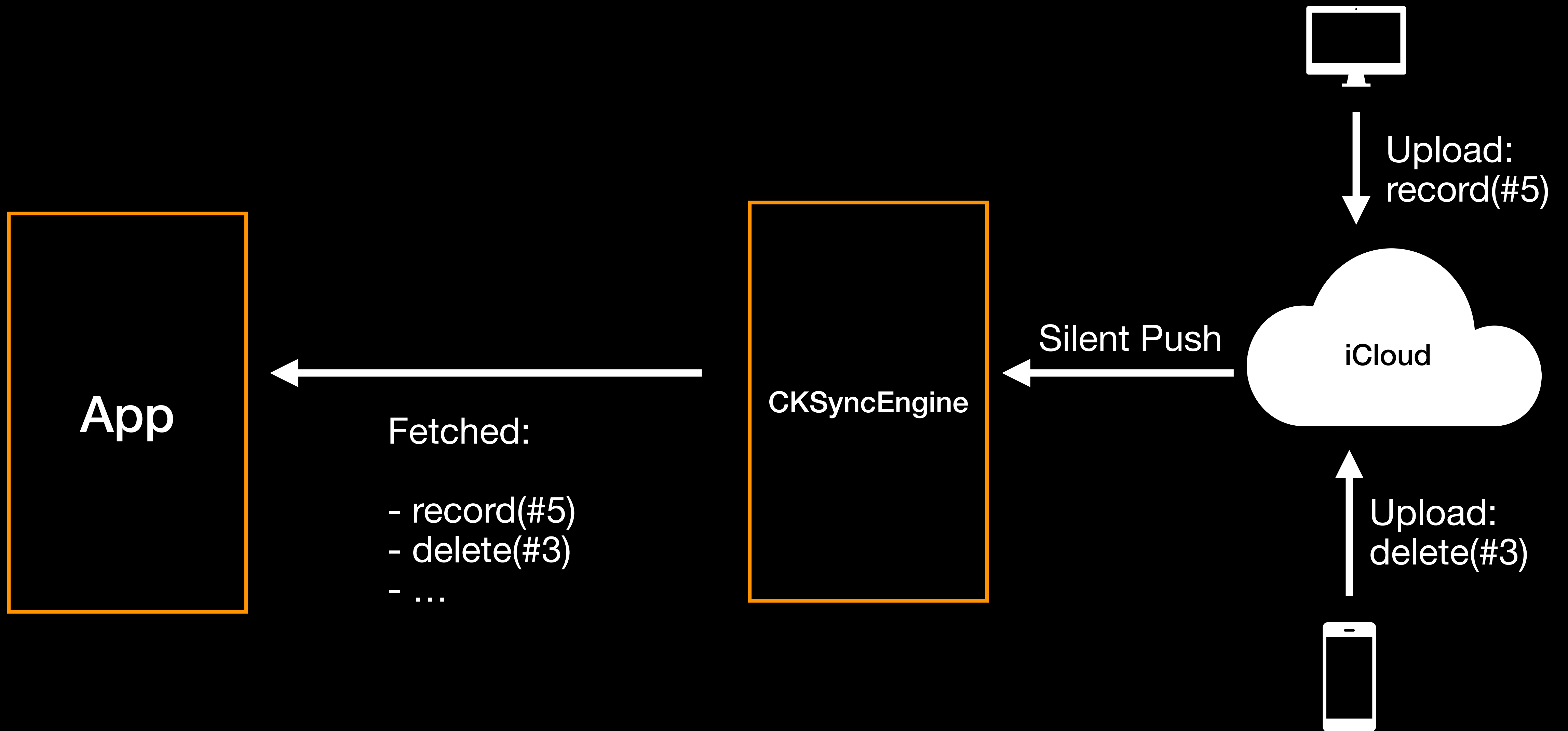
# Download



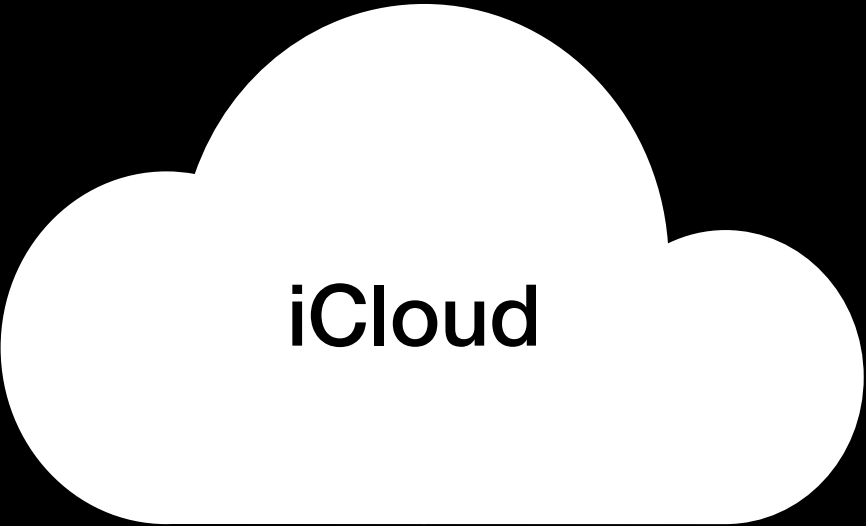
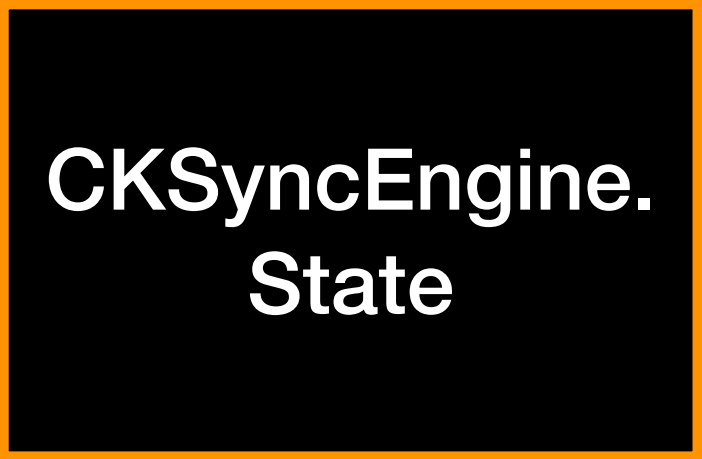
# Download



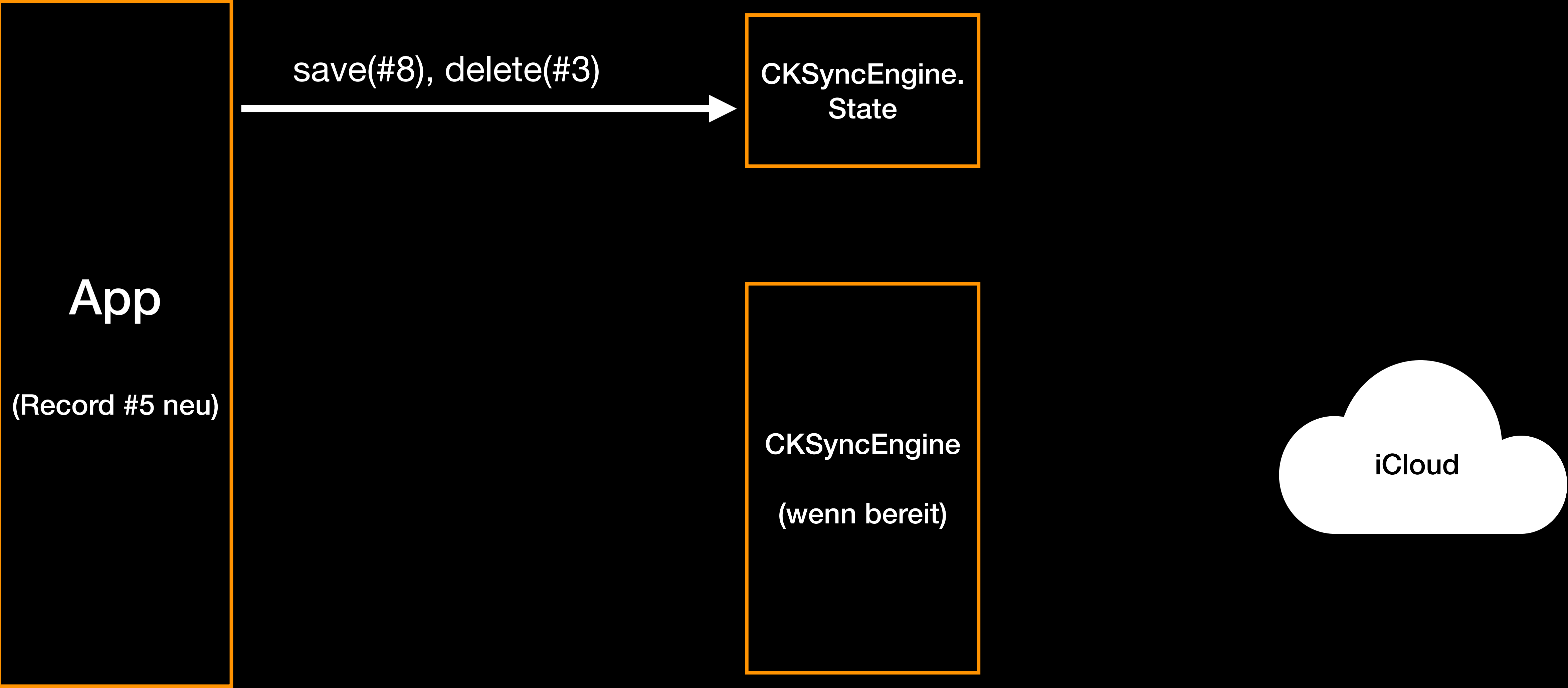
# Download



# Upload

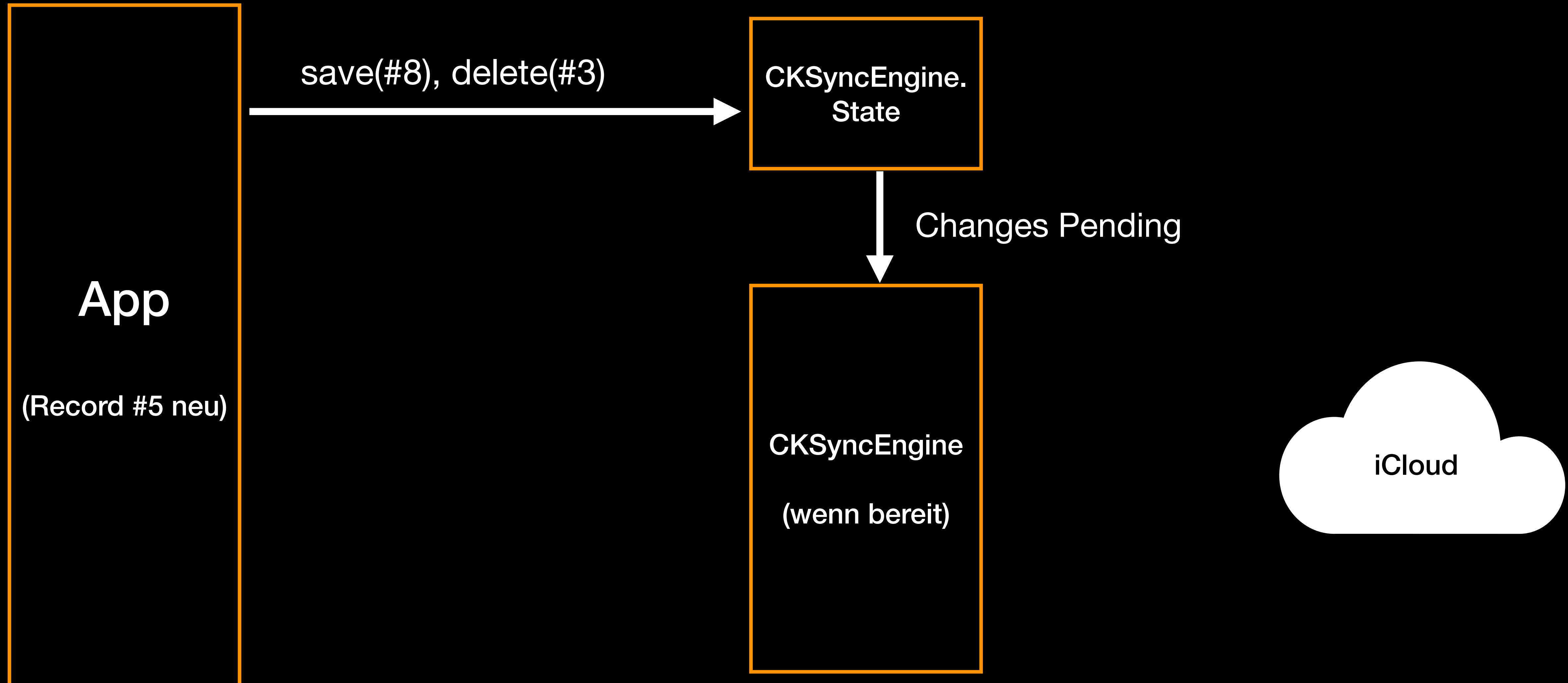


# Upload

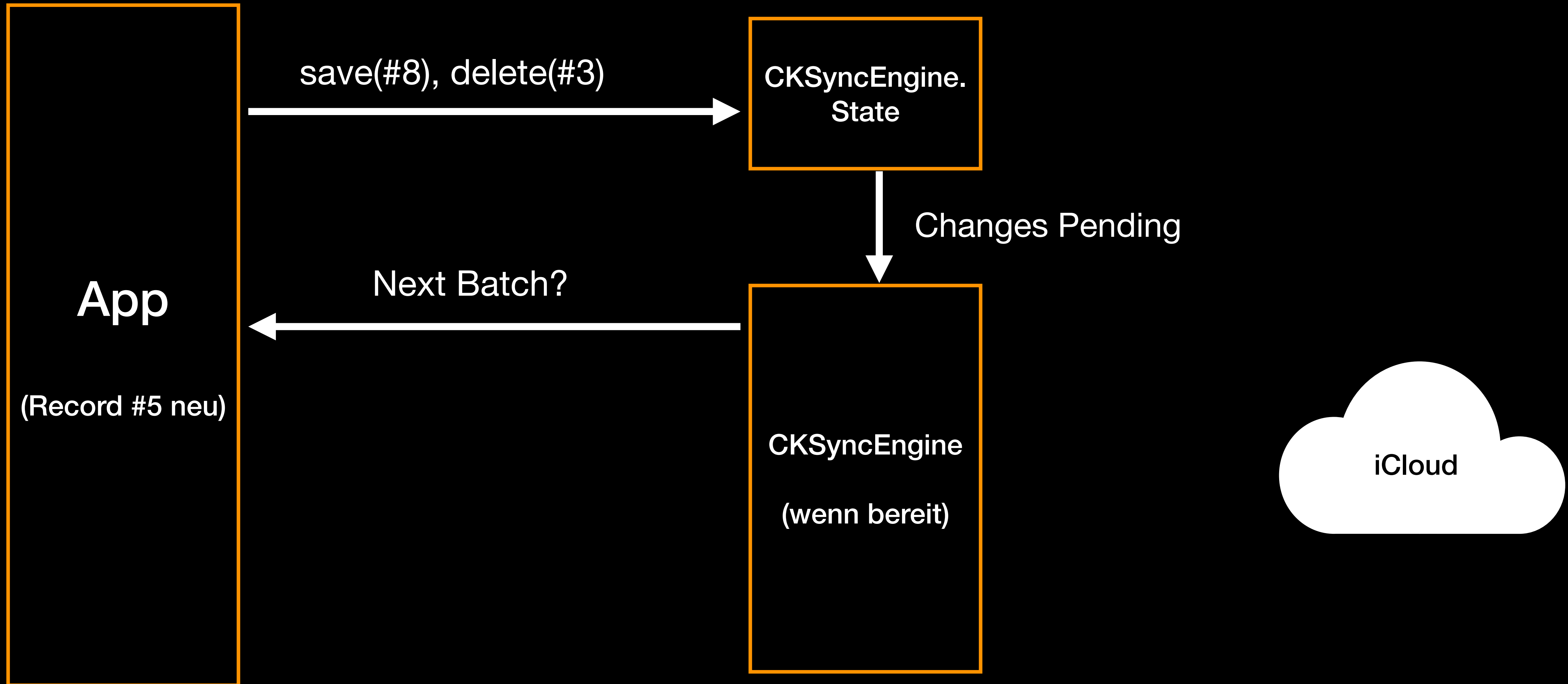




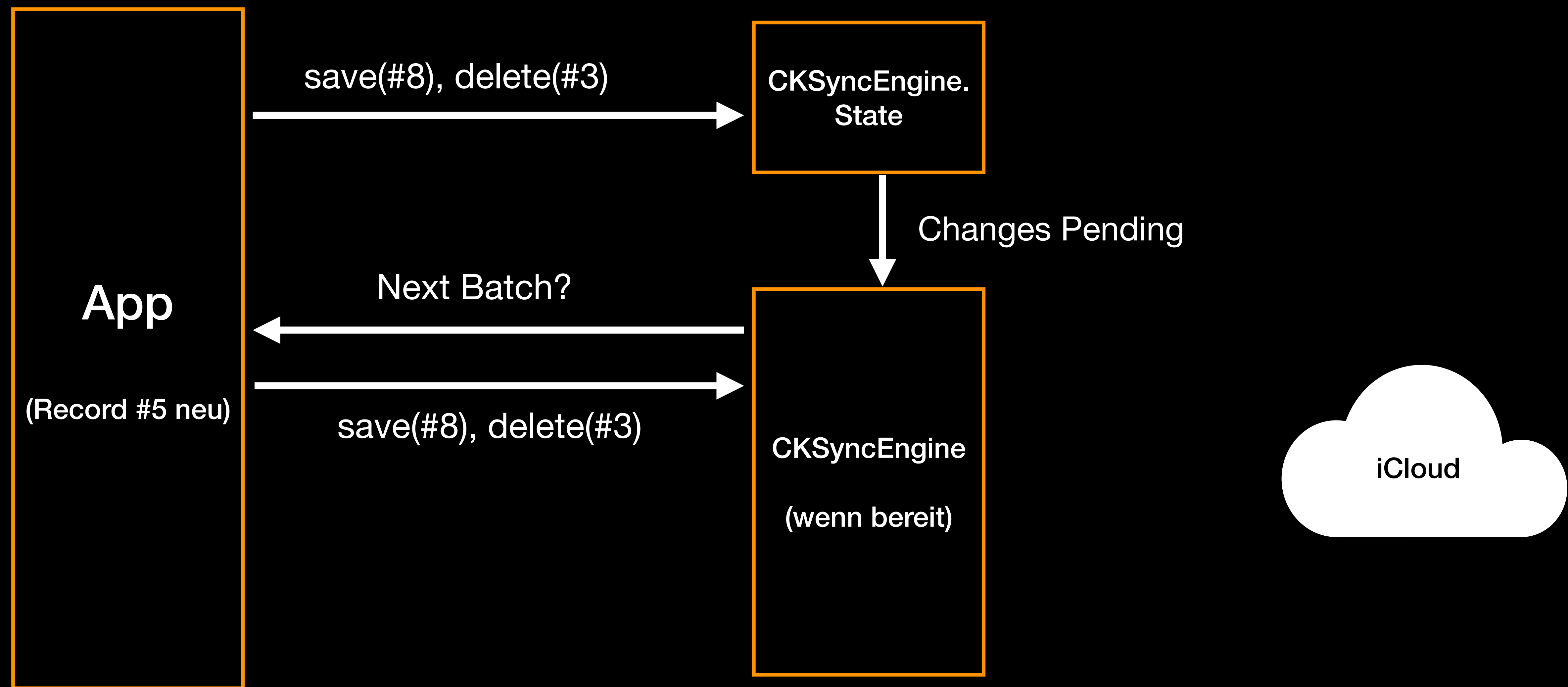
# Upload



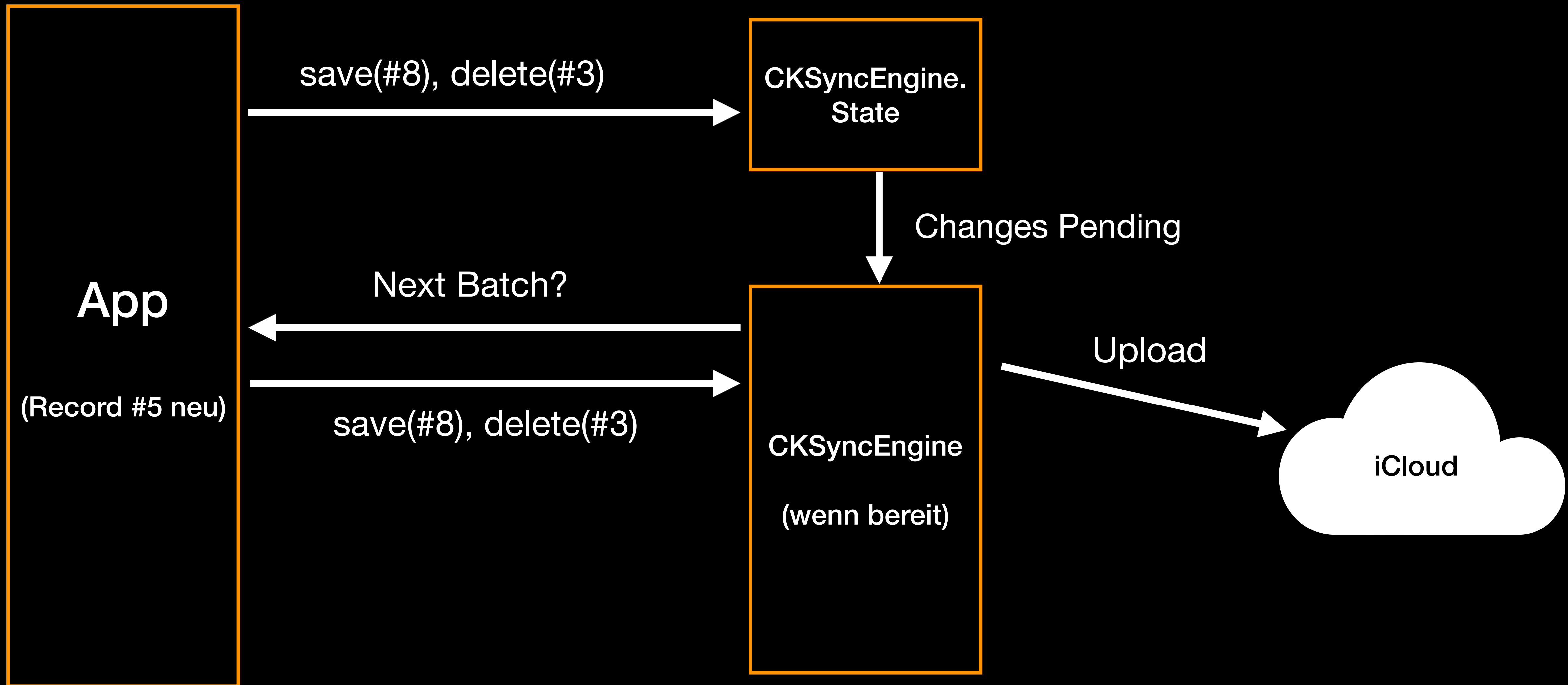
# Upload



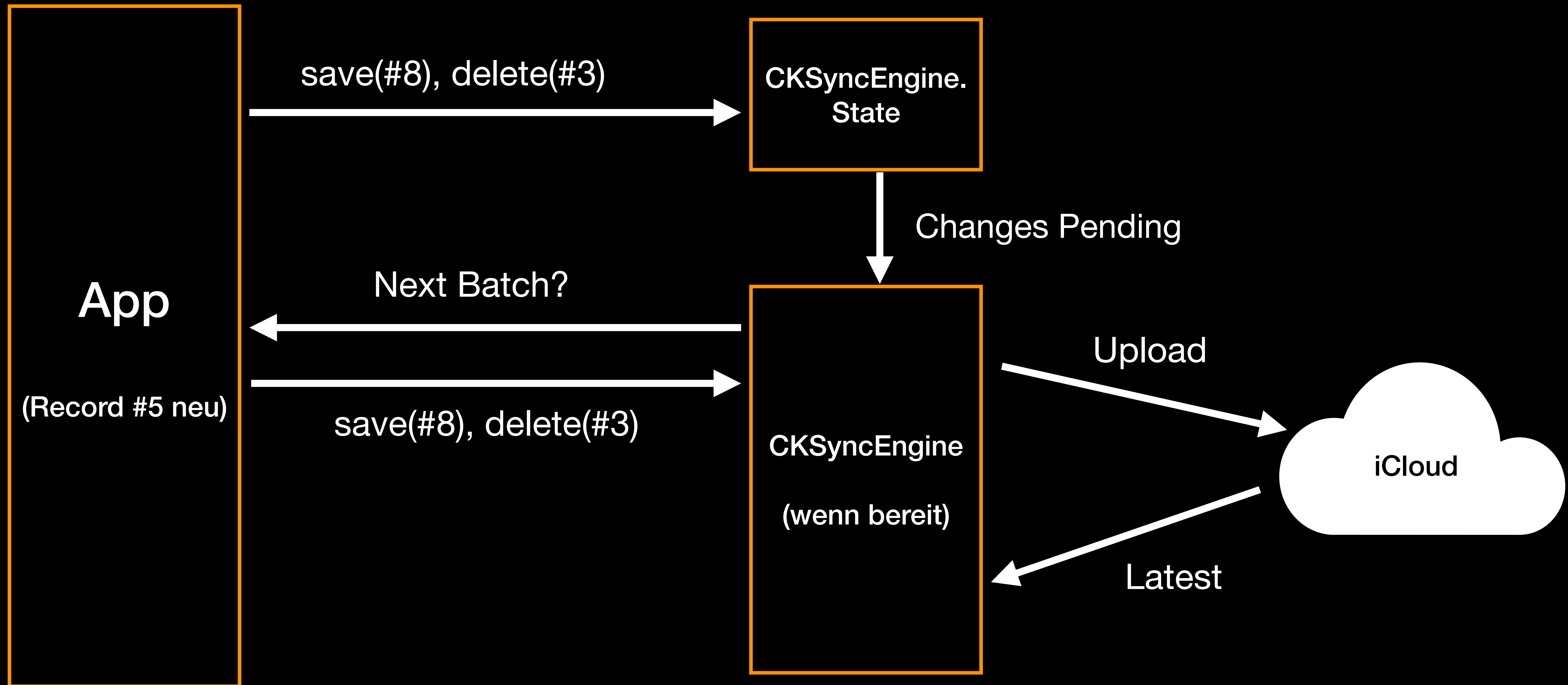
# Upload



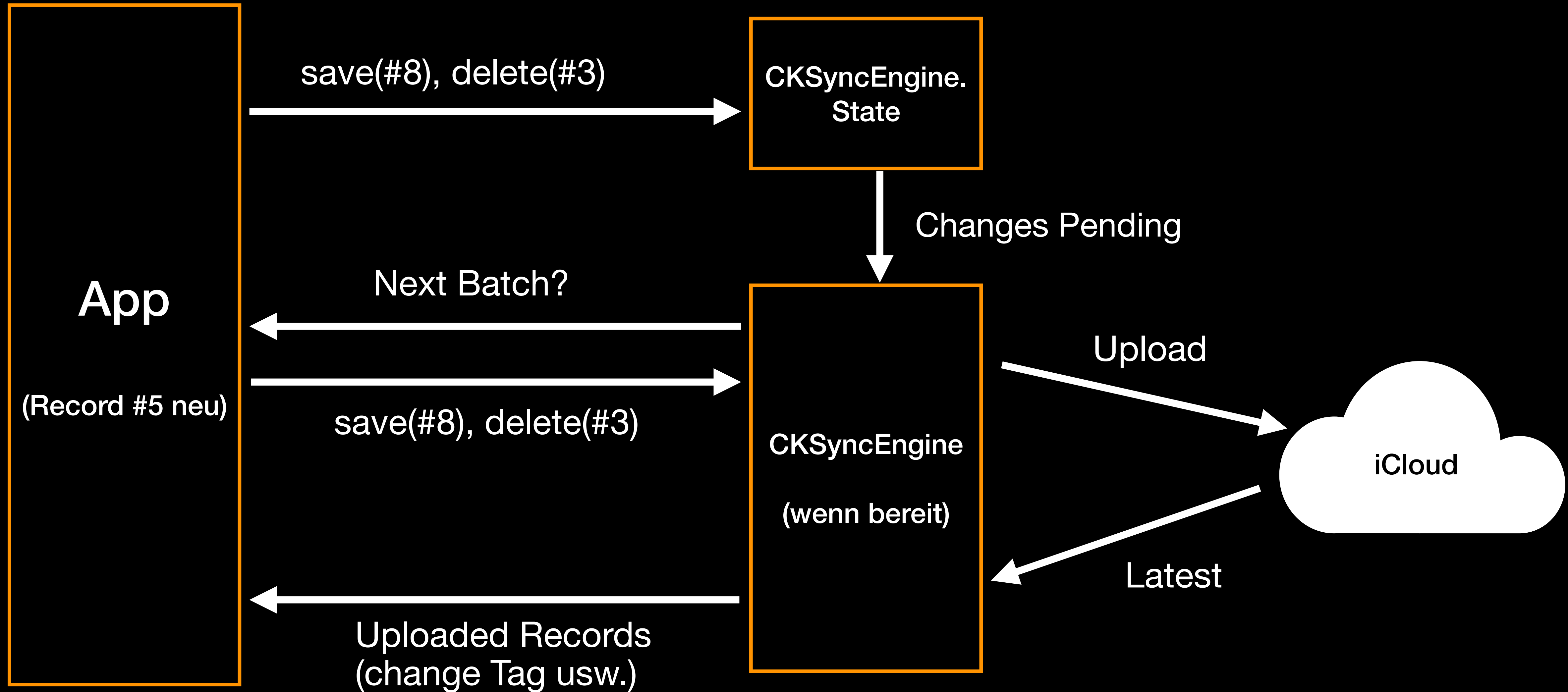
# Upload



# Upload



# Upload



# Sync-Konflikte

# Sync-Konflikte

- CloudKit: Konflikterkennung auf Record-Ebene durch *changeTag*



# Sync-Konflikte

- CloudKit: Konflikterkennung auf Record-Ebene durch *changeTag*
- Download-Konflikt:

# Sync-Konflikte

- CloudKit: Konflikterkennung auf Record-Ebene durch *changeTag*
- Download-Konflikt:
  - Server-Änderung trifft auf lokale Änderung

# Sync-Konflikte

- CloudKit: Konflikterkennung auf Record-Ebene durch *changeTag*
- Download-Konflikt:
  - Server-Änderung trifft auf lokale Änderung
  - Erkennung und Merge durch App

# Sync-Konflikte

- CloudKit: Konflikterkennung auf Record-Ebene durch *changeTag*
- Download-Konflikt:
  - Server-Änderung trifft auf lokale Änderung
  - Erkennung und Merge durch App
- Upload-Konflikt:

# Sync-Konflikte

- CloudKit: Konflikterkennung auf Record-Ebene durch *changeTag*
- Download-Konflikt:
  - Server-Änderung trifft auf lokale Änderung
  - Erkennung und Merge durch App
- Upload-Konflikt:
  - Lokales Change Tag  $\neq$  Remote Change Tag

# Sync-Konflikte

- CloudKit: Konflikterkennung auf Record-Ebene durch *changeTag*
- Download-Konflikt:
  - Server-Änderung trifft auf lokale Änderung
  - Erkennung und Merge durch App
- Upload-Konflikt:
  - Lokales Change Tag != Remote Change Tag
  - Fehler beim sentChanges-Event

# Sync-Konflikte

- CloudKit: Konflikterkennung auf Record-Ebene durch *changeTag*
- Download-Konflikt:
  - Server-Änderung trifft auf lokale Änderung
  - Erkennung und Merge durch App
- Upload-Konflikt:
  - Lokales Change Tag != Remote Change Tag
  - Fehler beim sentChanges-Event
  - Client merged und lädt Version mit Remote Change Tag hoch

# Sample Code

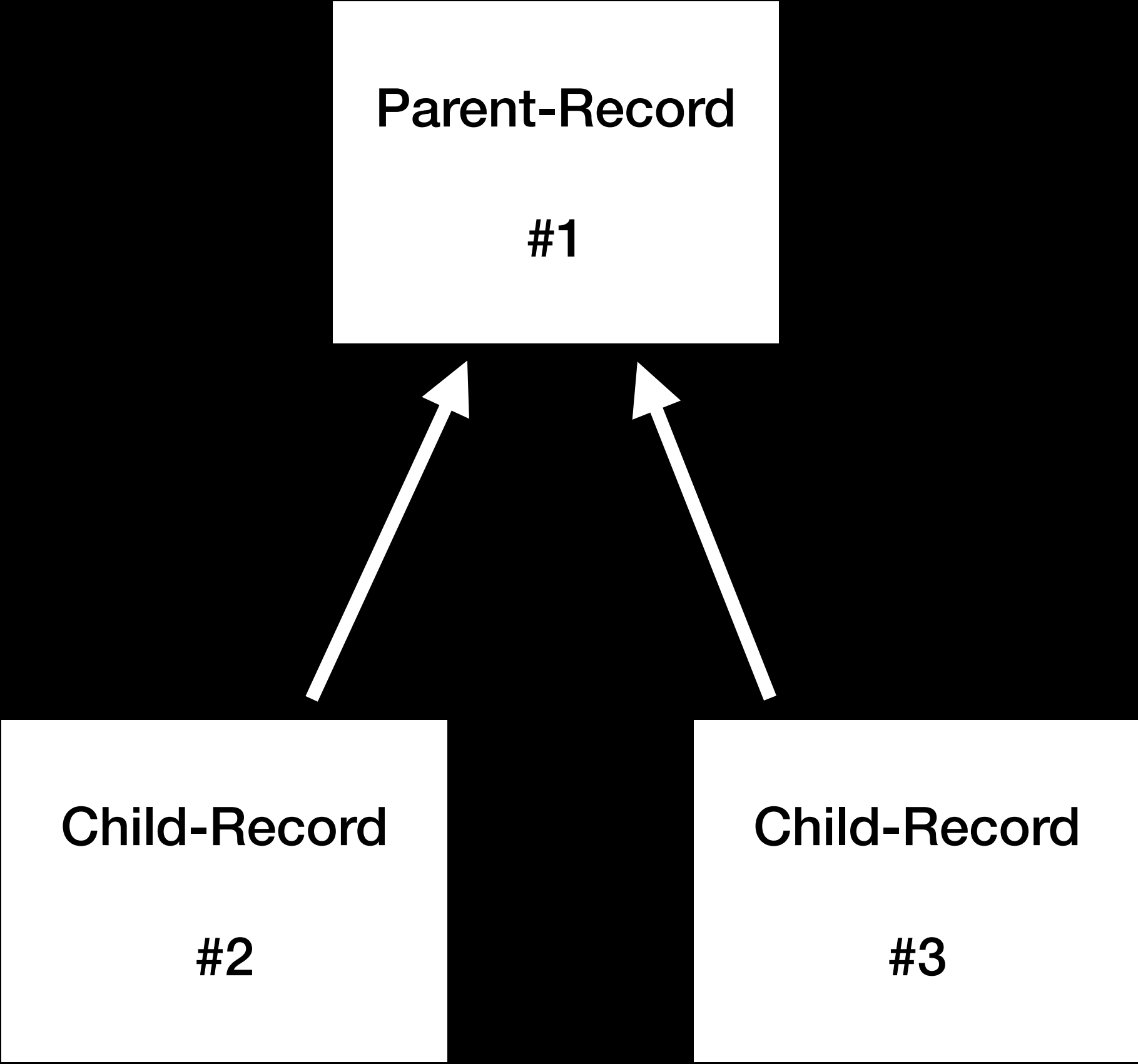
<https://github.com/apple/sample-cloudkit-sync-engine>



**Sonderfälle...**

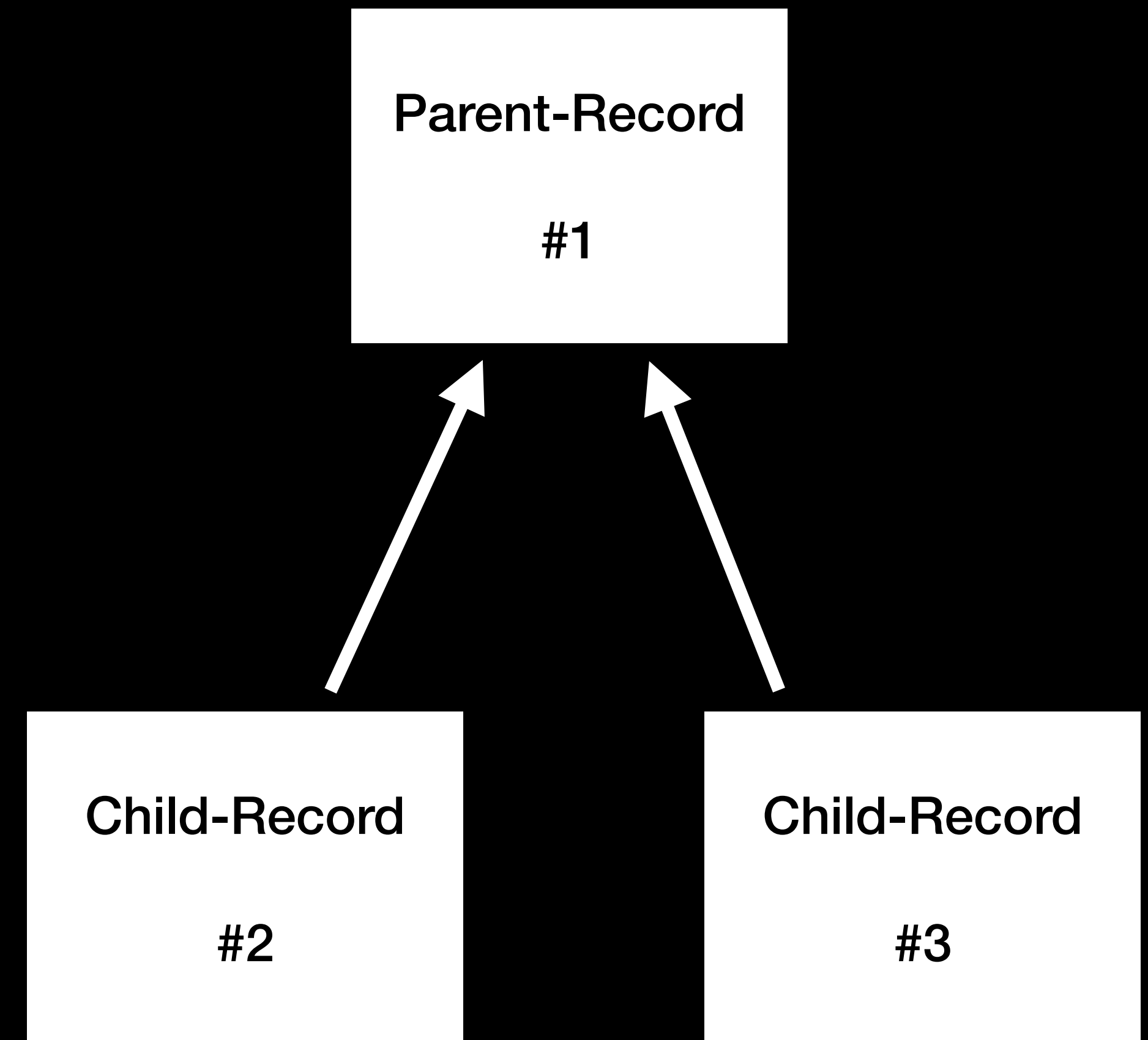
# Parent-Records

# Parent-Records



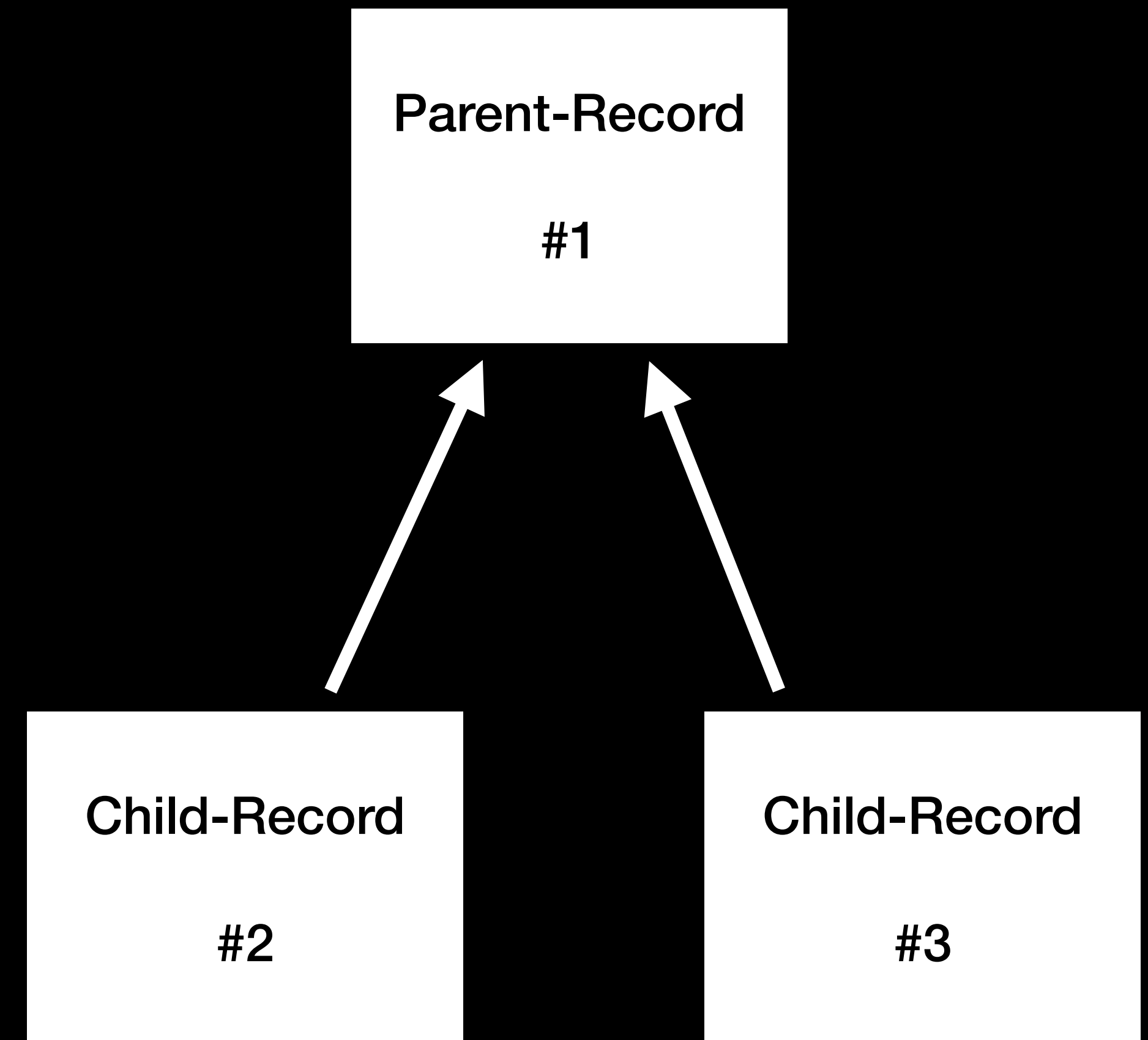
# Parent-Records

- CloudKit fordert:



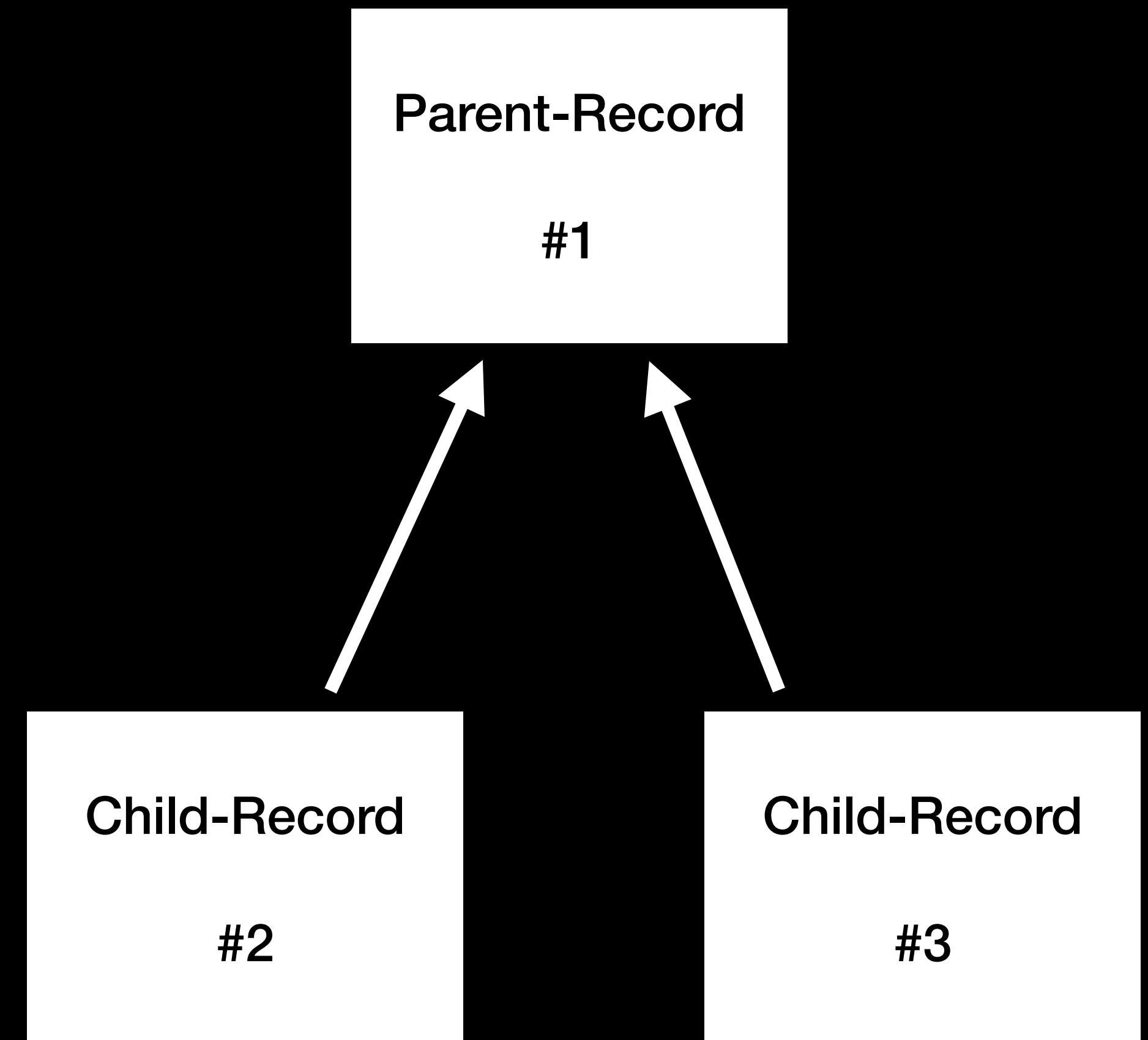
# Parent-Records

- CloudKit fordert:
  - Zyklenfreiheit



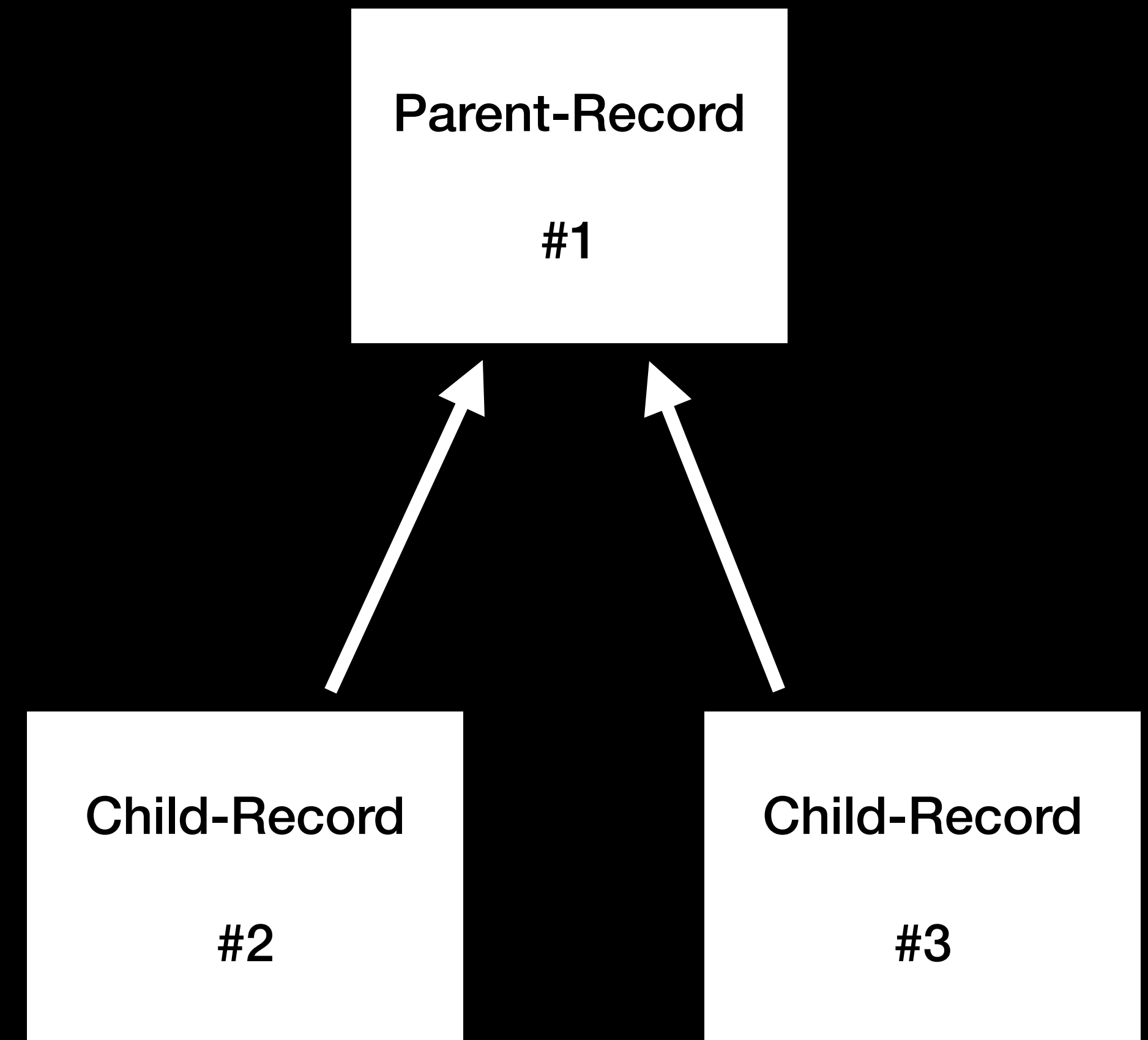
# Parent-Records

- CloudKit fordert:
  - Zyklenfreiheit
  - Gültige Parent Referenz



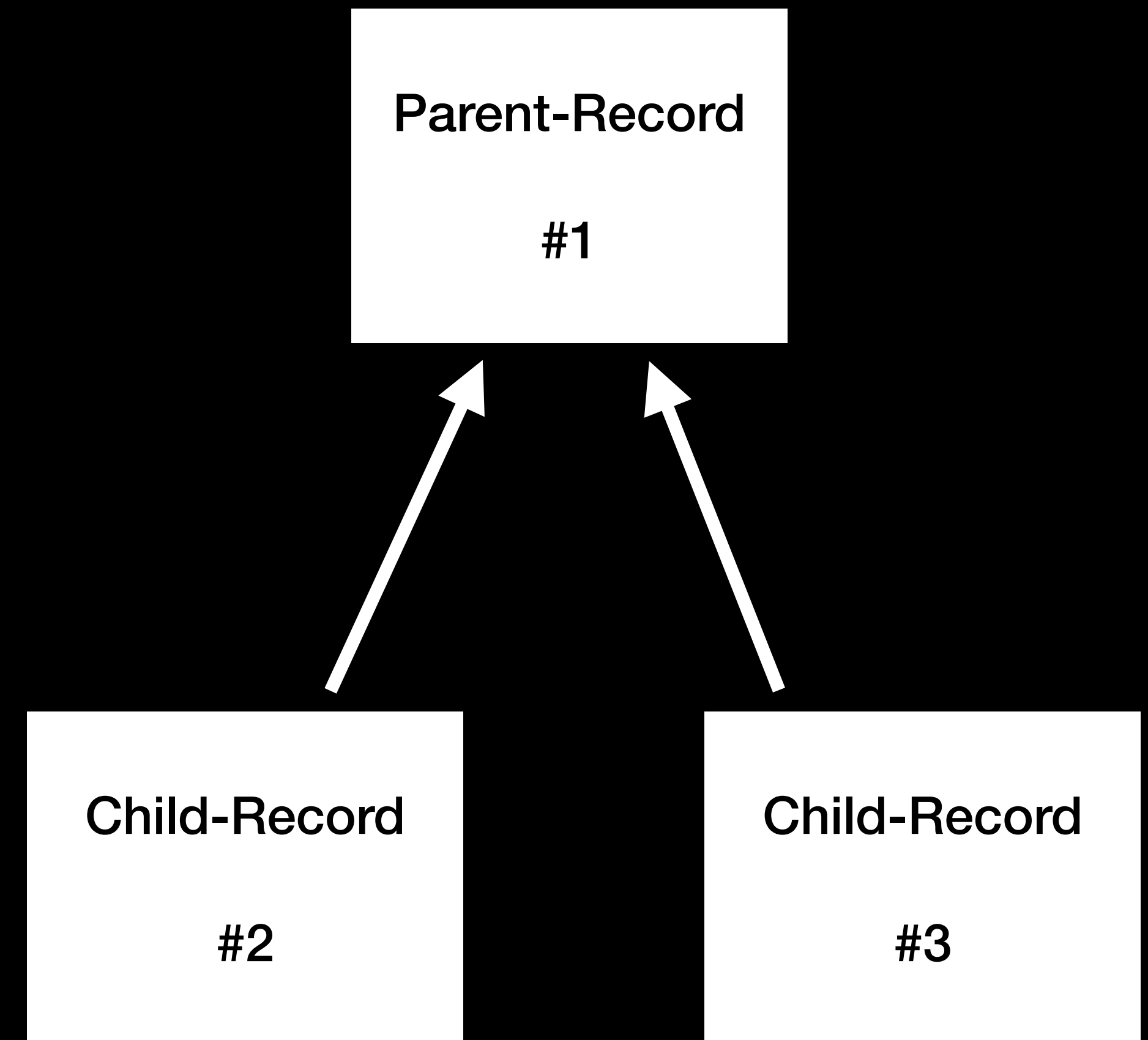
# Parent-Records

- CloudKit fordert:
  - Zyklenfreiheit
  - Gültige Parent Referenz
- Konflikt z.B.:



# Parent-Records

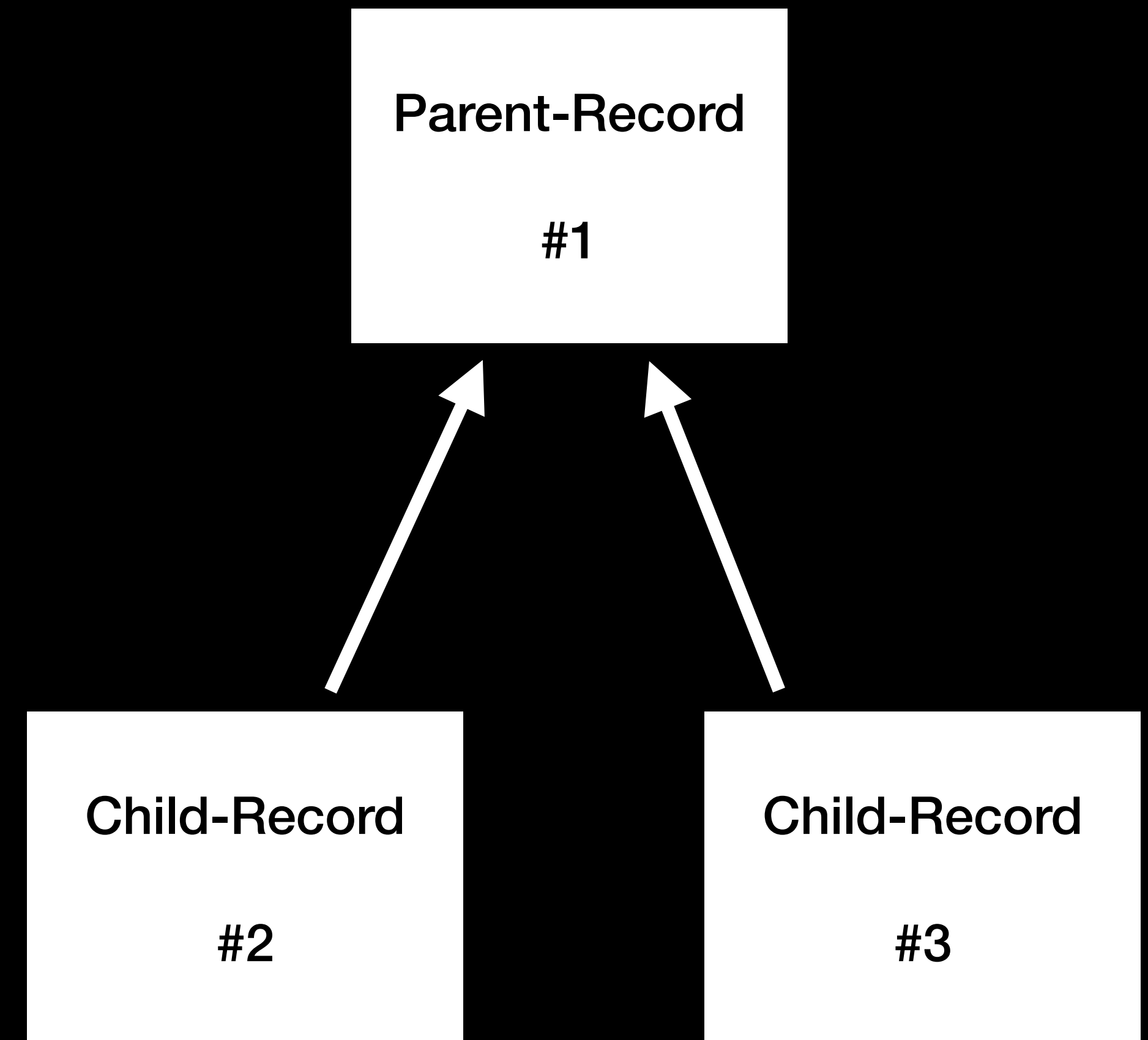
- CloudKit fordert:
  - Zyklenfreiheit
  - Gültige Parent Referenz
- Konflikt z.B.:
  - iPhone: Löscht Parent





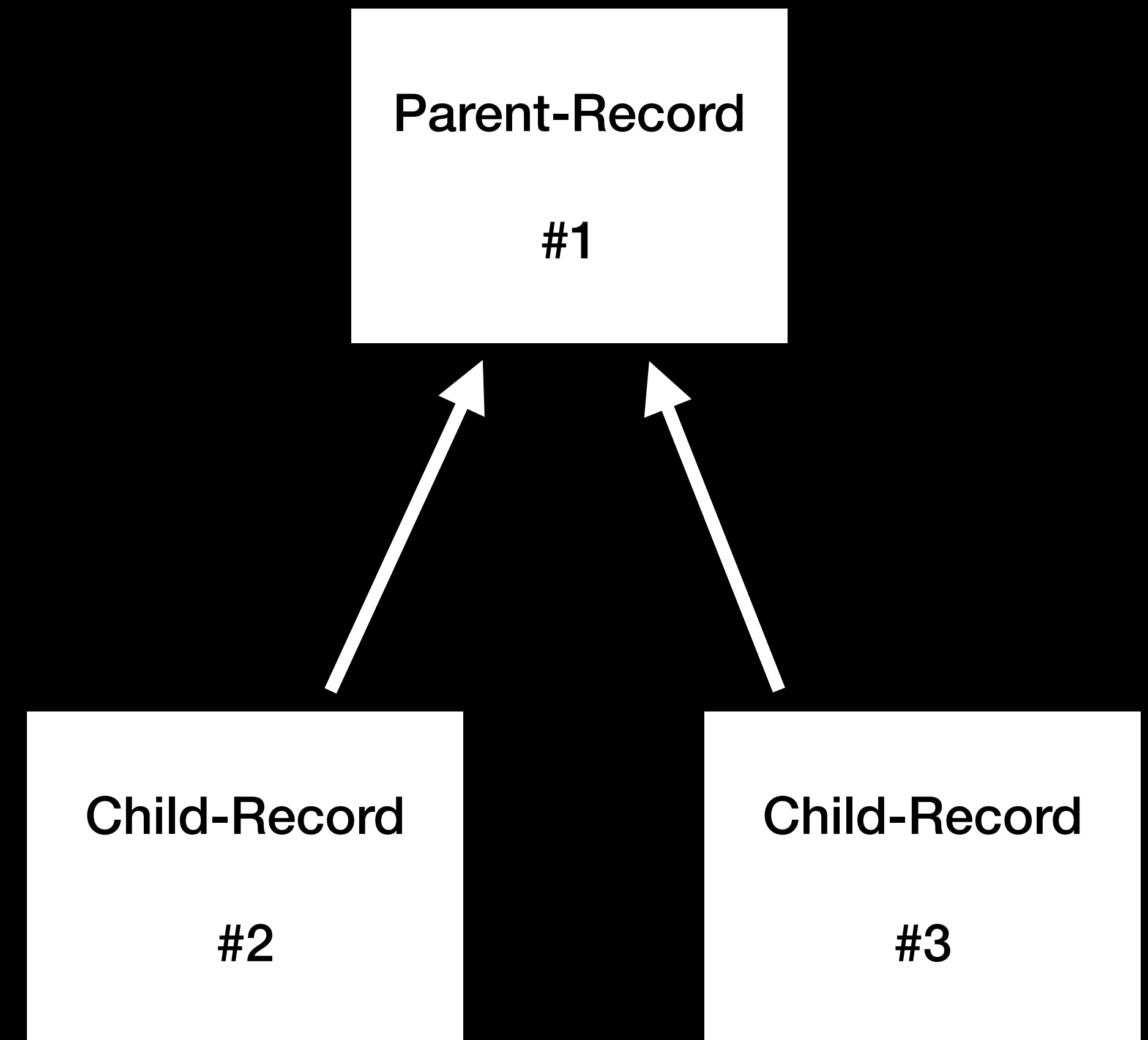
# Parent-Records

- CloudKit fordert:
  - Zyklentreiheit
  - Gültige Parent Referenz
- Konflikt z.B.:
  - iPhone: Löscht Parent
  - Mac: Fügt Kind hinzu



# Parent-Records

- CloudKit fordert:
  - Zyklenfreiheit
  - Gültige Parent Referenz
- Konflikt z.B.:
  - iPhone: Löscht Parent
  - Mac: Fügt Kind hinzu
- Fehlermeldung beim Upload



# Batching

# Batching

- CloudKit lädt Records in mehreren Batches

# Batching

- CloudKit lädt Records in mehreren Batches
- Sortierung wichtig (z.B. Parent vor Child anlegen)

# Batching

- CloudKit lädt Records in mehreren Batches
- Sortierung wichtig (z.B. Parent vor Child anlegen)
- Bündelung wichtig (z.B. atomare Änderungen mehrerer Records)

# Batching

- CloudKit lädt Records in mehreren Batches
- Sortierung wichtig (z.B. Parent vor Child anlegen)
- Bündelung wichtig (z.B. atomare Änderungen mehrerer Records)
- Download:

# Batching

- CloudKit lädt Records in mehreren Batches
- Sortierung wichtig (z.B. Parent vor Child anlegen)
- Bündelung wichtig (z.B. atomare Änderungen mehrerer Records)
- Download:
  - Beliebige Reihenfolge innerhalb eines Batches



# Batching

- CloudKit lädt Records in mehreren Batches
- Sortierung wichtig (z.B. Parent vor Child anlegen)
- Bündelung wichtig (z.B. atomare Änderungen mehrerer Records)
- Download:
  - Beliebige Reihenfolge innerhalb eines Batches
  - Modell muss temporäre Inkonsistenzen abkönnen (z.B. "Bild fehlt noch")

# Batching

- CloudKit lädt Records in mehreren Batches
- Sortierung wichtig (z.B. Parent vor Child anlegen)
- Bündelung wichtig (z.B. atomare Änderungen mehrerer Records)
- Download:
  - Beliebige Reihenfolge innerhalb eines Batches
  - Modell muss temporäre Inkonsistenzen abkönnen (z.B. "Bild fehlt noch")
  - Parent/Child-Reihenfolge über mehrere Batches konsistent

# Batching

- CloudKit lädt Records in mehreren Batches
- Sortierung wichtig (z.B. Parent vor Child anlegen)
- Bündelung wichtig (z.B. atomare Änderungen mehrerer Records)
- Download:
  - Beliebige Reihenfolge innerhalb eines Batches
  - Modell muss temporäre Inkonsistenzen abkönnen (z.B. "Bild fehlt noch")
  - Parent/Child-Reihenfolge über mehrere Batches konsistent
- Kein atomares Update-Delete (z.B. für Reference Counts)

# Assets

# Assets

- CKAsset-Values für große Dateien

# Assets

- CKAsset-Values für große Dateien
- Beispiel: Record mit Titel + Bild

# Assets

- CKAsset-Values für große Dateien
- Beispiel: Record mit Titel + Bild
- Change-Notification immer gesamter Record

# Assets

- CKAsset-Values für große Dateien
- Beispiel: Record mit Titel + Bild
- Change-Notification immer gesamter Record
  - Problem: Bild wird neu runtergeladen



# Assets

- CKAsset-Values für große Dateien
- Beispiel: Record mit Titel + Bild
- Change-Notification immer gesamter Record
  - Problem: Bild wird neu runtergeladen
  - Lokaler CloudKit-Cache nur temporär

# Assets

- CKAsset-Values für große Dateien
- Beispiel: Record mit Titel + Bild
- Change-Notification immer gesamter Record
  - Problem: Bild wird neu runtergeladen
  - Lokaler CloudKit-Cache nur temporär
- Klassisches CloudKit: **desiredKeys**

# Assets

- CKAsset-Values für große Dateien
- Beispiel: Record mit Titel + Bild
- Change-Notification immer gesamter Record
  - Problem: Bild wird neu runtergeladen
  - Lokaler CloudKit-Cache nur temporär
- Klassisches CloudKit: **desiredKeys**
  - Nicht mit CKSyncEngine...

# Fazit: CKSyncEngine

# Fazit: CKSyncEngine

- K kümmert sich um Steuerung von Up/Downloads

# Fazit: CKSyncEngine

- Kümmert sich um Steuerung von Up/Downloads
- Merge-Logik in der App

# Fazit: CKSyncEngine

- Kümmert sich um Steuerung von Up/Downloads
- Merge-Logik in der App
- Auf Batching achten

# Fazit: CKSyncEngine

- Kümmert sich um Steuerung von Up/Downloads
- Merge-Logik in der App
- Auf Batching achten
- Kleinere Limitierungen gegenüber CloudKit (z.B. *desiredKeys*)



# Weitere Features

# Weitere Features

- Handling Account-Wechsel

# Weitere Features

- Handling Account-Wechsel
- Manueller Sync

# Weitere Features

- Handling Account-Wechsel
- Manueller Sync
- Scoping auf bestimmte Zones

# Weitere Features

- Handling Account-Wechsel
- Manueller Sync
- Scoping auf bestimmte Zones
- Pending-Marker ohne IDs