

Macoun

Backends aus der Dose

Katrin Apel

Segen oder Büchse der Pandora?



Über mich

- Früher Produktmanager

Über mich

- Früher Produktmanager
- Konvertiert zu iOS Entwicklung vor 3 Jahren

Über mich

- Früher Produktmanager
- Konvertiert zu iOS Entwicklung vor 3 Jahren
- 2 Apps mit Parse entwickelt

Über mich

- Früher Produktmanager
- Konvertiert zu iOS Entwicklung vor 3 Jahren
- 2 Apps mit Parse entwickelt
- Hoodie für iOS portiert

Was ist eigentlich ein BaaS?

Warum ein BaaS?

Fokus



Fehlende Erfahrung



Fehlende Zeit



Kleines Budget

\$\$\$

Funktionalität wiederholt sich



Usermanagement



BaaS Ökosystem

Kinvey

Firebase

Flox

Parse

CloudKit

Hoodie

Apiomat

FatFractal

BaaSBox

3 Lösungen im Fokus

Parse

CloudKit

Hoodie

Parse

Parse in a Nutshell

- Gehört Facebook seit April 2013

Parse in a Nutshell

- Gehört Facebook seit April 2013
- Cloud Funktionen

Parse in a Nutshell

- Gehört Facebook seit April 2013
- Cloud Funktionen
- Module für Twilio, Stripe, Mailgun, etc.

Parse in a Nutshell

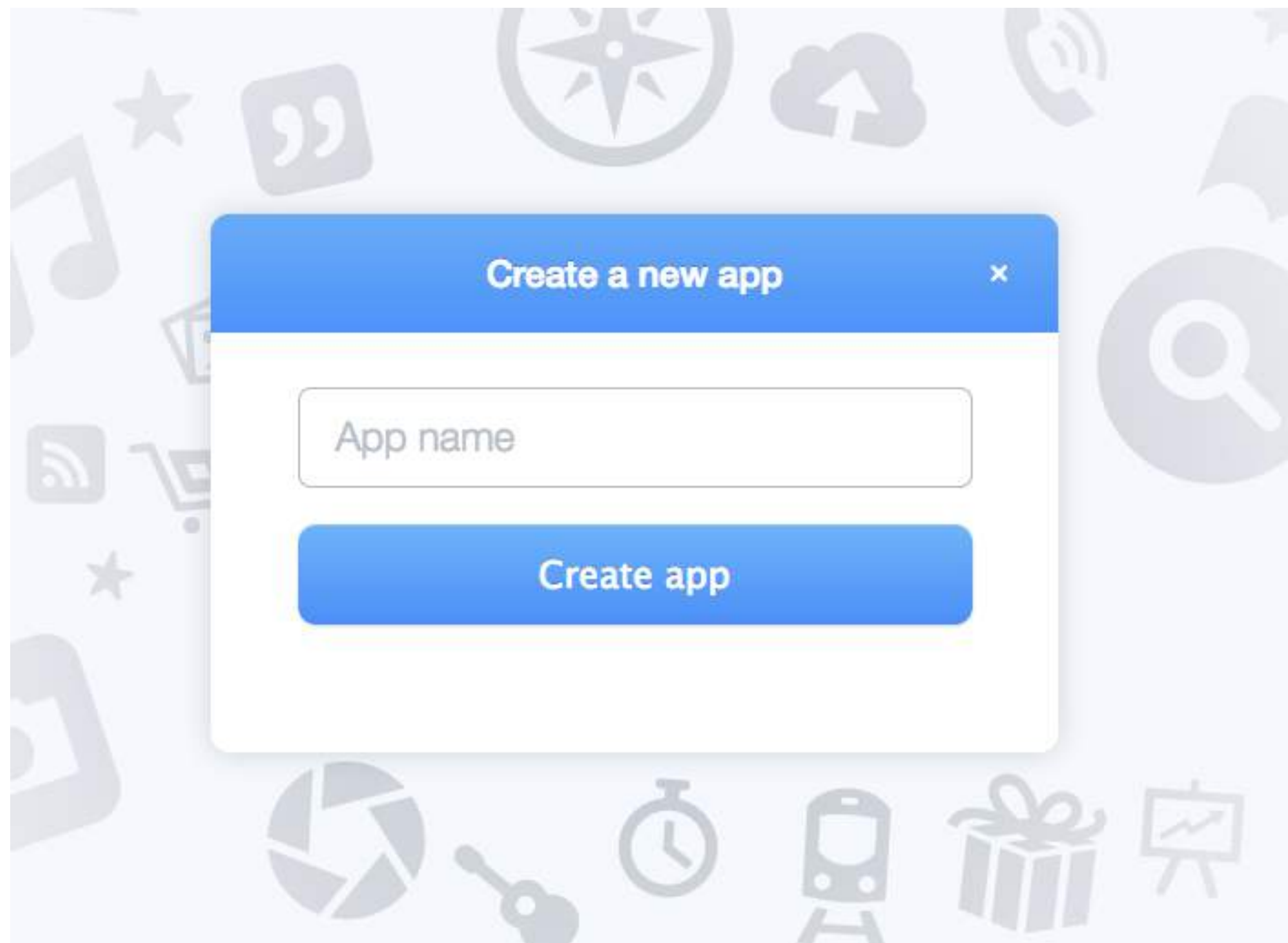
- Gehört Facebook seit April 2013
- Cloud Funktionen
- Module für Twilio, Stripe, Mailgun, etc.
- Push Notifications

Parse in a Nutshell

- Gehört Facebook seit April 2013
- Cloud Funktionen
- Module für Twilio, Stripe, Mailgun, etc.
- Push Notifications
- Analytics

Parse in a Nutshell

- Gehört Facebook seit April 2013
- Cloud Funktionen
- Module für Twilio, Stripe, Mailgun, etc.
- Push Notifications
- Analytics
- SDKs für iOS/OS X, Android, Javascript, PHP, Unity, Rest API



Parse

```
#import <Parse/Parse.h>
...

// Parse initialisieren
[Parse setApplicationId:@"APPLICATION_ID" clientKey:@"CLIENT_KEY"];
...
```

Parse - Nutzer registrieren

```
// Neuen Nutzer erstellen
```

```
PFUser *user = [PFUser user];  
user.username = @"kaalita";  
user.password = @"secretpassword";  
user.email = @"email@example.com";
```

```
[user signUpInBackgroundWithBlock:^(BOOL succeeded, NSError *error) {  
  
}];
```

Parse - Nutzer einloggen

```
// Nutzer einloggen

[PFUser logInWithUsernameInBackground:@"kaalita"
                                     password:@"secretpassword"
                                     block:^(PFUser *user, NSError *error) {
    if (user) {
        // Nutzer eingeloggt
    } else {

    }
}];
```

Parse - Objekt anlegen

```
// Neues Objekt anlegen
```

```
PFObject *blogPost = [PFObject objectWithClassName:@"BlogPost"];  
blogPost[@"title"] = @"Neuer Eintrag";  
blogPost[@"description"] = @"Hallo Welt. Willkommen";
```

Parse - Objekt speichern

```
// Objekt speichern
```

```
PFObject *blogPost = [PFObject objectWithClassName:@"BlogPost"];  
blogPost[@"title"] = @"Neuer Eintrag";  
blogPost[@"description"] = @"Hallo Welt. Willkommen";
```

```
[postObject saveInBackgroundWithBlock:^(BOOL succeeded, NSError *error) {  
  
}];
```


Parse - Datenabfrage

Parse - Segen

- Sehr großer Funktionsumfang

Parse - Segen

- Sehr großer Funktionsumfang
- Einfaches Setup

Parse - Segen

- Sehr großer Funktionsumfang
- Einfaches Setup
- Unterstützung diverser Plattformen

Parse - Segen

- Sehr großer Funktionsumfang
- Einfaches Setup
- Unterstützung diverser Plattformen
- Push Notifications

Parse - Büchse der Pandora

- Schlechte Erweiterbarkeit

Parse - Büchse der Pandora

- Schlechte Erweiterbarkeit
- Debugging

Parse - Büchse der Pandora

- Schlechte Erweiterbarkeit
- Debugging
- Festgelegt auf Parse Server

CloudKit

CloudKit in a Nutshell

- Basiert auf iCloud

CloudKit in a Nutshell

- Basiert auf iCloud
- Push Notifications

CloudKit in a Nutshell

- Basiert auf iCloud
- Push Notifications
- iOS und OS X

General

Capabilities

Info

Build Settings

Build Phases

Build Rules

▼ iCloud

Services: ☒ Key-value storage
☐ iCloud Documents
☒ CloudKit

Containers: ☒ Use default container
☐ Specify custom containers

☒ iCloud.de.kaalita.CloudKitDemo iCloud.5(CFBundleIdentifier)

☐ iCloud.de.heinemann.CloudKitApp

☐ iCloud.de.kaalita.CloudKitApp

+ ↺

CloudKit Dashboard

▶ Target Dependencies (0 items)

▶ Compile Sources (11 items) ×

▼ Link Binary With Libraries (1 item) ×

Name	Status
 CloudKit.framework	Required 

  Drag to reorder frameworks

▶ Headers (10 items) ×

▶ Copy Bundle Resources (4 items) ×

▶ Run Script ×

CloudKit

App User = iCloud User

CloudKit - Nutzerfreigabe

```
CKContainer *c = [CKContainer defaultContainer];  
  
[c requestApplicationPermission:CKApplicationPermissionUserDiscoverability  
    completionHandler:^(  
        CKApplicationPermissionStatus applicationPermissionStatus, NSError *error) {  
    }];
```

**Dürfen Personen, die
„de.kaalita.CloudKitDemo“
verwenden, Sie per E-Mail
finden?**

Personen, die Ihre E-Mail-Adresse
kennen, können sehen, dass Sie diese
App benutzen.

Nicht erlauben

OK

CloudKit - Nutzerinformation

```
CKContainer *c = [CKContainer defaultContainer];

[c fetchUserRecordIDWithCompletionHandler:^(CKRecordID *recordID,
                                           NSError *error) {

    if (!error) {
        [c discoverUserInfoWithUserRecordID:recordID
            completionHandler:^(CKDiscoveredUserInfo *user,
                                NSError *error) {

        }];
    }
}];
```

CloudKit - Nutzer finden

```
self.container = [CKContainer defaultContainer];

CKDiscoverAllContactsOperation *op =
    [[CKDiscoverAllContactsOperation alloc] init];

[op setDiscoverAllContactsCompletionBlock:^(NSArray *userInfos,
                                             NSError *error) {

}];

[self.container addOperation:op];
```

App Container

Private Datenbanken



Öffentliche DB



CloudKit - Objekt anlegen

```
// Neues Objekt erstellen
```

```
CKRecord *blogPost = [[CKRecord alloc] initWithRecordType:@"Post"];  
blogPost[@"title"] = @"Neuer Eintrag";  
blogPost[@"location"] = ;  
blogPost[@"description"] = @"Hallo Welt. Willkommen";
```

CloudKit - Objekt speichern (I)

```
// In öffentlicher Datenbank speichern  
  
CKDatabase *publicDatabase = [CKContainer publicCloudDatabase];  
  
[publicDatabase saveRecord:record  
    completionHandler:^(CKRecord *record, NSError *error) {  
  
}];
```


CloudKit - Objekt speichern (II)

```
// In privater Datenbank von iCloud Nutzer speichern  
CKDatabase *privateDatabase = [CKContainer privateCloudDatabase];  
  
[privateDatabase saveRecord:record  
    completionHandler:^(CKRecord *record, NSError *error) {  
  
}];
```

CloudKit - Datenabfrage

Subscriptions

Push Notifications basierend
auf Query

CloudKitApp

SCHEMA

Record Types

Security Roles

Subscription Types

PUBLIC DATA

User Records

Default Zone

PRIVATE DATA

Default Zone

ADMIN

Team

Deployment

Environment: DEVELOPMENT

Record Types

Sort by Name

Items

2 Attributes

Photos

1 Attribute

Users

1 Attribute

+

Katrin Apel

Items

Created:Sep 24 2014 17:38

Modified:Sep 24 2014 18:03

Security:Default

Metadata Index:Fields

Attribute Name	Attribute Type	Index
location	Location	Sort, Query
name	String	Sort, Search, Query

Add Attribute...

CloudKit - Segen

- Beste Integration / API

CloudKit - Segen

- Beste Integration / API
- Adressbuch Matching inklusive

CloudKit - Segen

- Beste Integration / API
- Adressbuch Matching inklusive
- Push Notifications basierend auf Queries

CloudKit - Büchse der Pandora

- Apple einzige Plattform

CloudKit - Büchse der Pandora

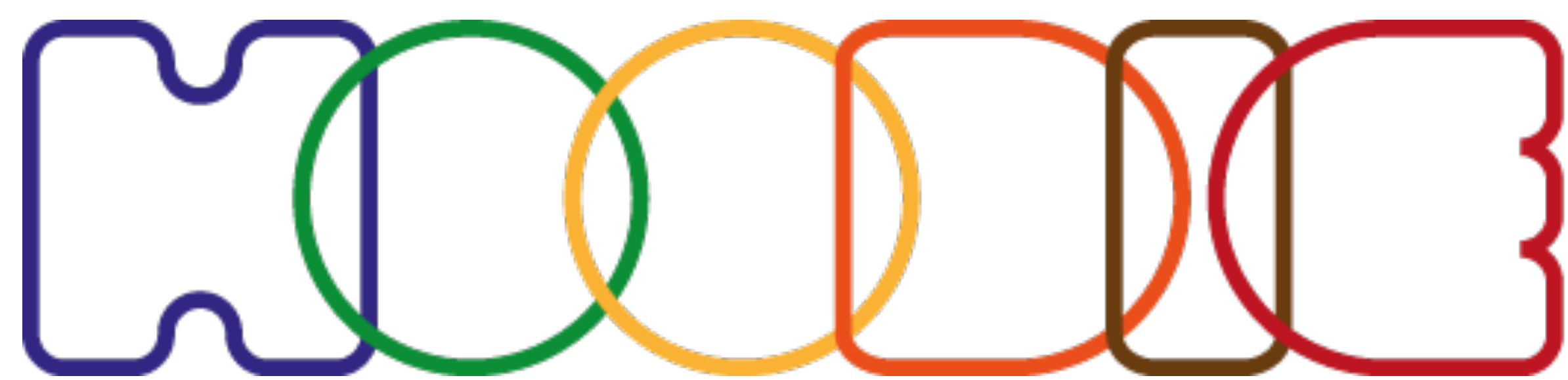
- Apple einzige Plattform
- Daten Lock-In

CloudKit - Büchse der Pandora

- Apple einzige Plattform
- Daten Lock-In
- Schlechte Erweiterbarkeit

CloudKit - Büchse der Pandora

- Apple einzige Plattform
- Daten Lock-In
- Schlechte Erweiterbarkeit
- Kein lokaler Cache out of the box



in a Nutshell

- Open Source

in a Nutshell

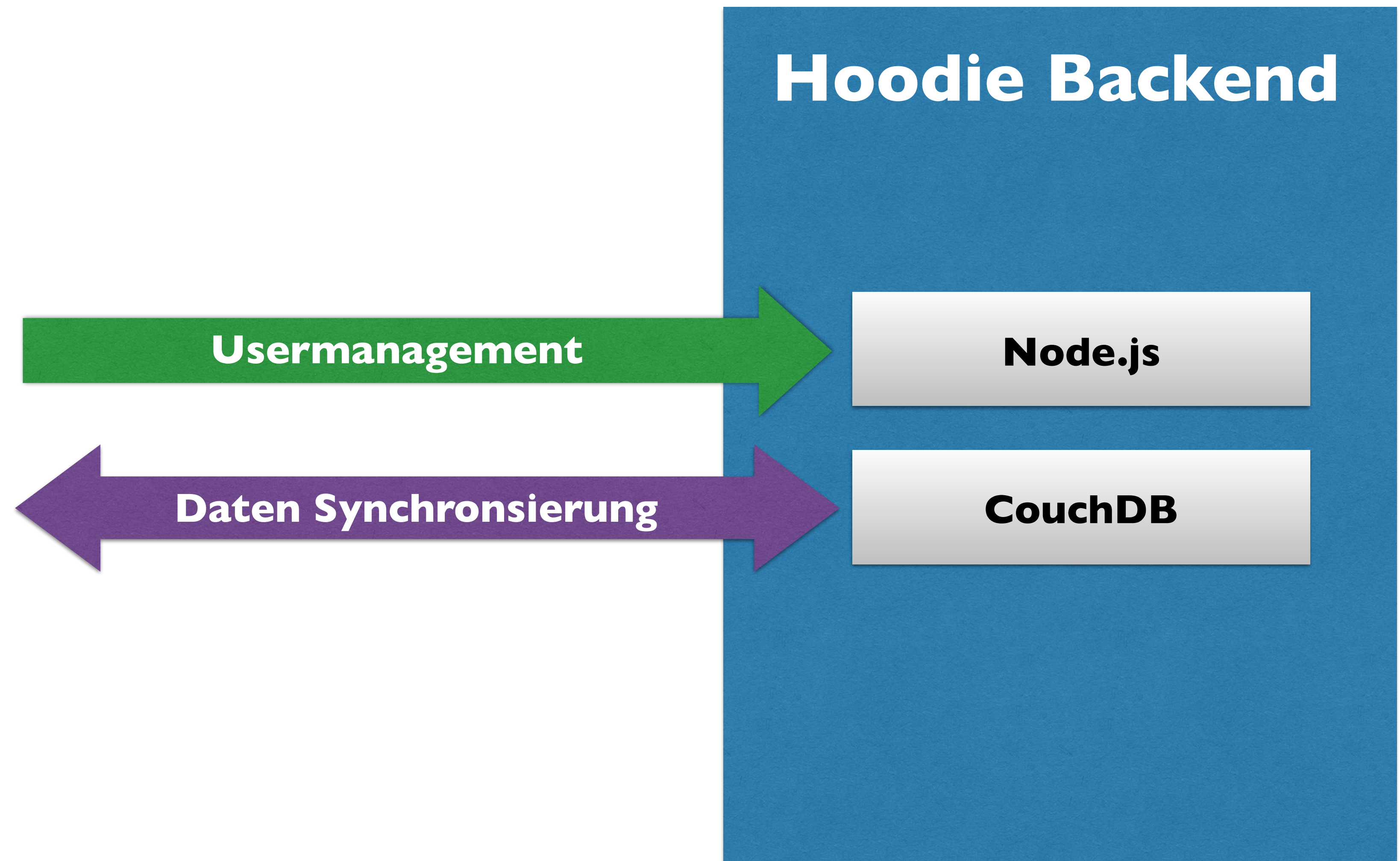
- Open Source
- Offline First

in a Nutshell

- Open Source
- Offline First
- Kein Hosting integriert

in a Nutshell

- Open Source
- Offline First
- Kein Hosting integriert
- Javascript, iOS (alpha), Rest API



Demo

Hoodie Initialisierung

```
#import "HOOHoodie.h"
...

@property (strong, nonatomic) HOOHoodie *hoodie;
...

// Hoodie initialisieren
NSString *baseURLString = @"http://127.0.0.1:6004";
self.hoodie = [[HOOHoodie alloc] initWithBaseURLString:baseURLString];

...
```

Hoodie - Nutzer registrieren

Hoodie - Objekt anlegen

```
// Neues Objekt anlegen

NSDictionary *blogPost = @{@"title": @"Neuer Eintrag",
                           @"description": @"Hallo Welt. Willkommen"};

[self.hoodie.store saveObject:blogPost
                      withType:@"post"
                      onSave:^(NSDictionary *savedObject, NSError *error) {

}];
```

Hoodie - Datenabfrage

```
// Daten abfragen
```

```
[self.store findAllObjectsWithType:@"blogPost"]
```

Hoodie - Segen

- Synchronisierungs-Lösung

Hoodie - Segen

- Synchronisierungs-Lösung
- Gute Erweiterbarkeit

Hoodie - Segen

- Synchronisierungs-Lösung
- Gute Erweiterbarkeit
- Open Source

Hoodie - Segen

- Synchronisierungs-Lösung
- Gute Erweiterbarkeit
- Open Source
- Verwendung kostenlos

Hoodie - Büchse der Pandora

- geringer Feature Umfang

Hoodie - Büchse der Pandora

- geringer Feature Umfang
- Noch nicht so stabil

Hoodie - Büchse der Pandora

- geringer Feature Umfang
- Noch nicht so stabil
- Hosting nicht inbegriffen

	Parse	CloudKit	Hoodie
Usermanagement	X	(X)	X
Daten speichern	X	X	X
Business Logik	X	-	(X)
Hosting	X	X	-
Offline Caching	(X)	-	X
Push Notifications	X	X	-

Parse - Mainstream All-Inclusive Lösung

CloudKit - Exklusiv Lösung

Hoodie - Indie Lösung

BaaS für meine App?

App Anforderungen



App Anforderungen



Abwägen

Erweiterbarkeit

Daten Lock In

Preis

App Anforderungen



App Anforderungen



App Anforderungen



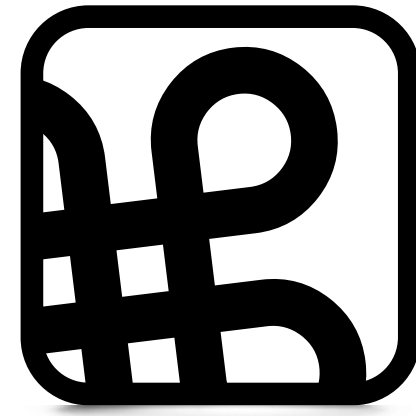
Finger weg!

Mehr Infos

- Parse: <http://www.parse.com>
- CloudKit
 - https://developer.apple.com/library/ios/documentation/CloudKit/Reference/CloudKit_Framework_Reference/
 - WWDC 2014 Sessions: Introducing CloudKit, & Advanced CloudKit
- Hoodie: <http://hood.ie>

Fragen?

Vielen Dank



Macoun