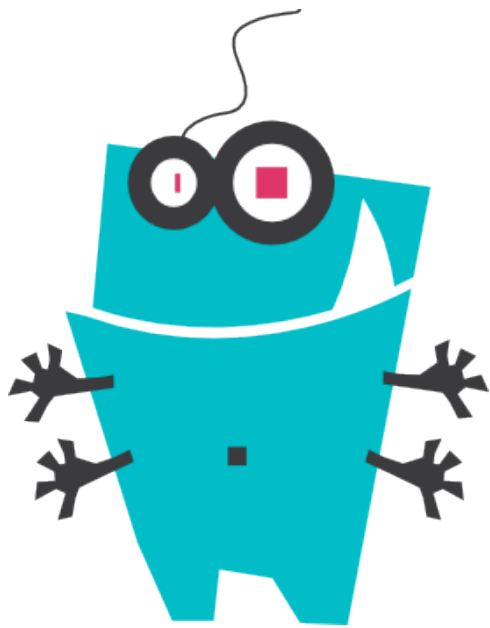


Macoun



Willkommen in der Hölle

Alexander von Below

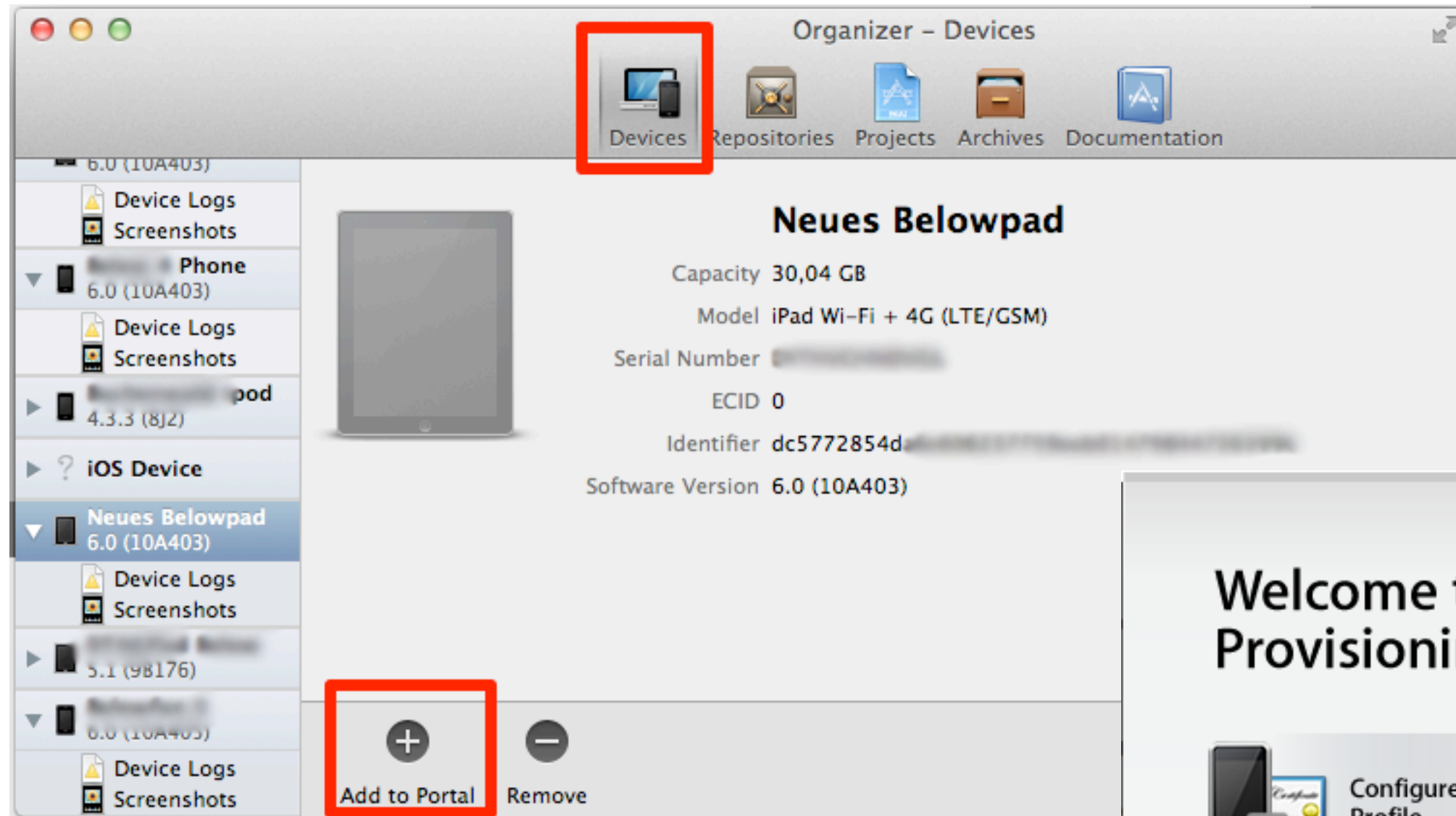
Was ist das schrecklichste an
der iOS Entwicklung?

Code Signing /
Provisioning

Ablauf

- Was macht es so schrecklich?
- Elemente des Code Signing / Provisioning
- Leider keine Patentlösungen
- Ein paar Tips

Ist doch ganz leicht...



Probleme...

- Code Signing ist ein Sicherheitsmechanismus
- Der Natur nach daher nicht fehlertolerant
- Viele Teile
- Identitätskonflikte

Ein Überblick



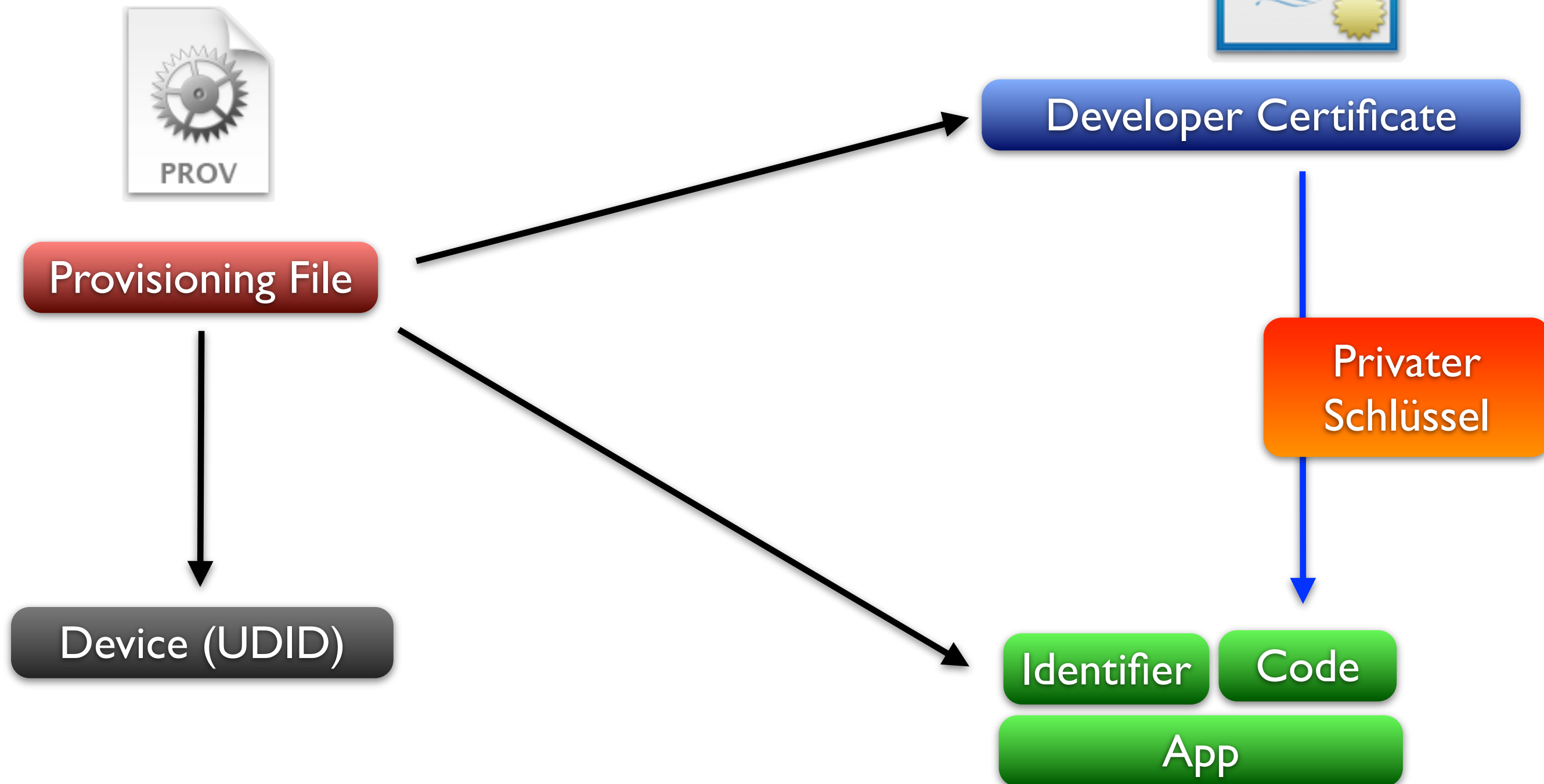
Quelle: Apple, Inc.

Ein Überblick



Quelle: Hieronymus Bosch

Mal im Ernst ...



Certificate

- Digitale Identität
- Ermöglicht eine digitale Unterschrift
- Implementiert durch „Chain of Trust“



iPhone Developer: Alexander G. von Below (blurred)

Ausgestellt von: Apple Worldwide Developer Relations Certification Authority

Ablaufdatum: Donnerstag, 18. April 2013 00:34:10 Mitteleuropäische Sommerzeit

✓ Dieses Zertifikat ist gültig.



Worldwide Developer Relations

[Download certificate ▶](#) [Download CPS ▶](#) [Download CRL ▶](#)



Apple Inc. Root Certificate

Root certificate authority

Expires: Friday, February 9, 2035 1:40:36PM US/Pacific

✓ This certificate is valid

.mobileprovision

- Certificate
- application-identifier
- Entitlements
 - iCloud
 - Push Notifications
 - ...
- Devices

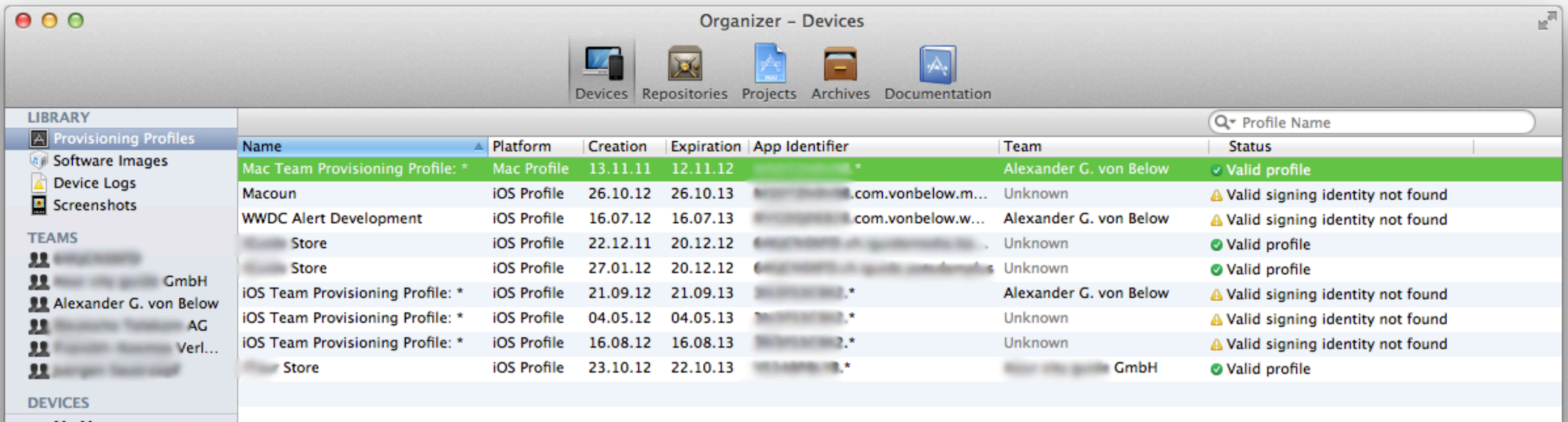
Auswahl in Xcode

- Build Settings: Code Signing Identity
- Automatic Selector wenn möglich verwenden
- Gewählt wird sowohl ein Profil wie ein Zertifikat
- Erster Schritt der Fehlersuche



Organizer: Profiles

- Window: Organizer (⌘⇧2), Devices:Provisioning Profiles
- Zweiter Schritt der Fehlersuche



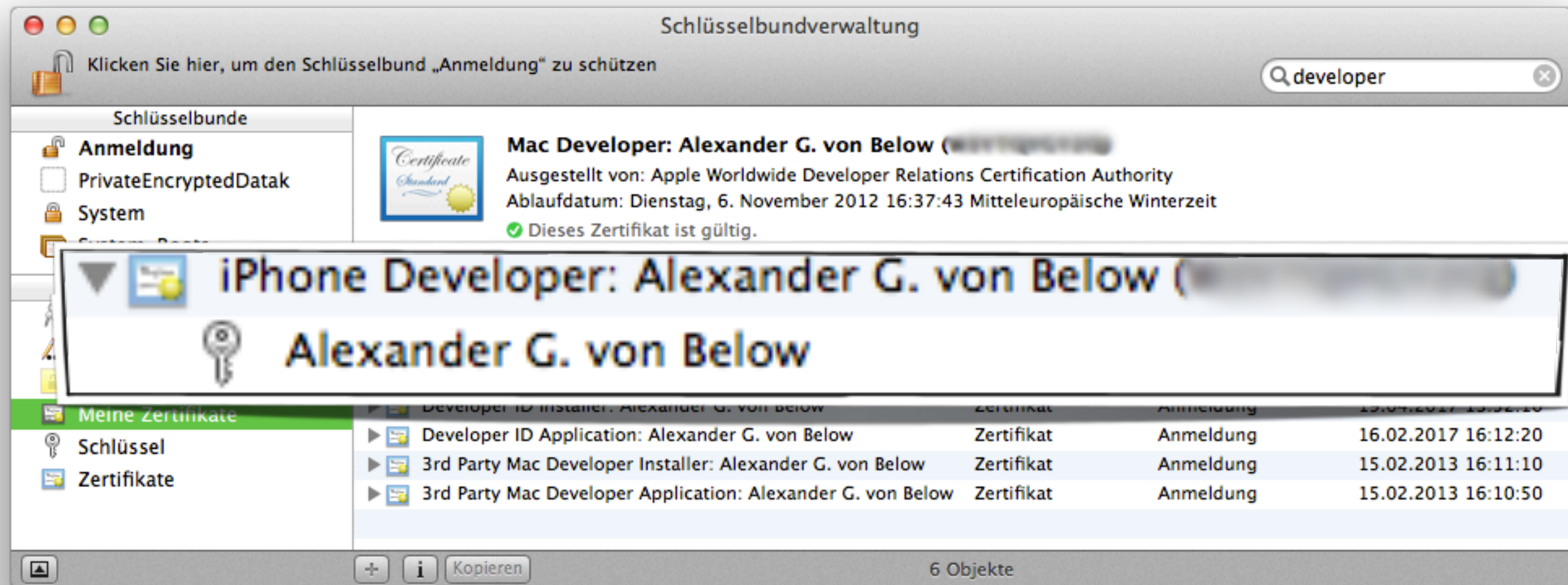
Schlüsselbund: Certificates

- „Zertifikate“ enthält alle Zertifikate auf dem Rechner
- „Meine Zertifikate“ enthält nur die Zertifikate mit Private Key





Schlüsselbund: Certificates

- „Zertifikate“ enthält alle Zertifikate auf dem Rechner
- „Meine Zertifikate“ enthält nur die Zertifikate mit Private Key





Identitätskonflikte

  Check dependencies 1

Check dependencies

Code Sign error: Certificate identity 'iPhone Distribution: [REDACTED] GmbH' appears more than once in the keychain. The codesign tool requires there only be one.

 Certificate identity 'iPhone Distribution: [REDACTED] GmbH' appears more than once in the keychain. The codesign tool requires there only be one.
[less](#)

 Activity Log Complete 23.10.12 16:14
1 error

Untote Zertifikate!

- Löschen aus der Keychain löscht sie nicht
- Xcode stellt sie aus den Profilen immer wieder her!
- Insb. für Enterprise Zertifikate, auch abgelaufene Zertifikate

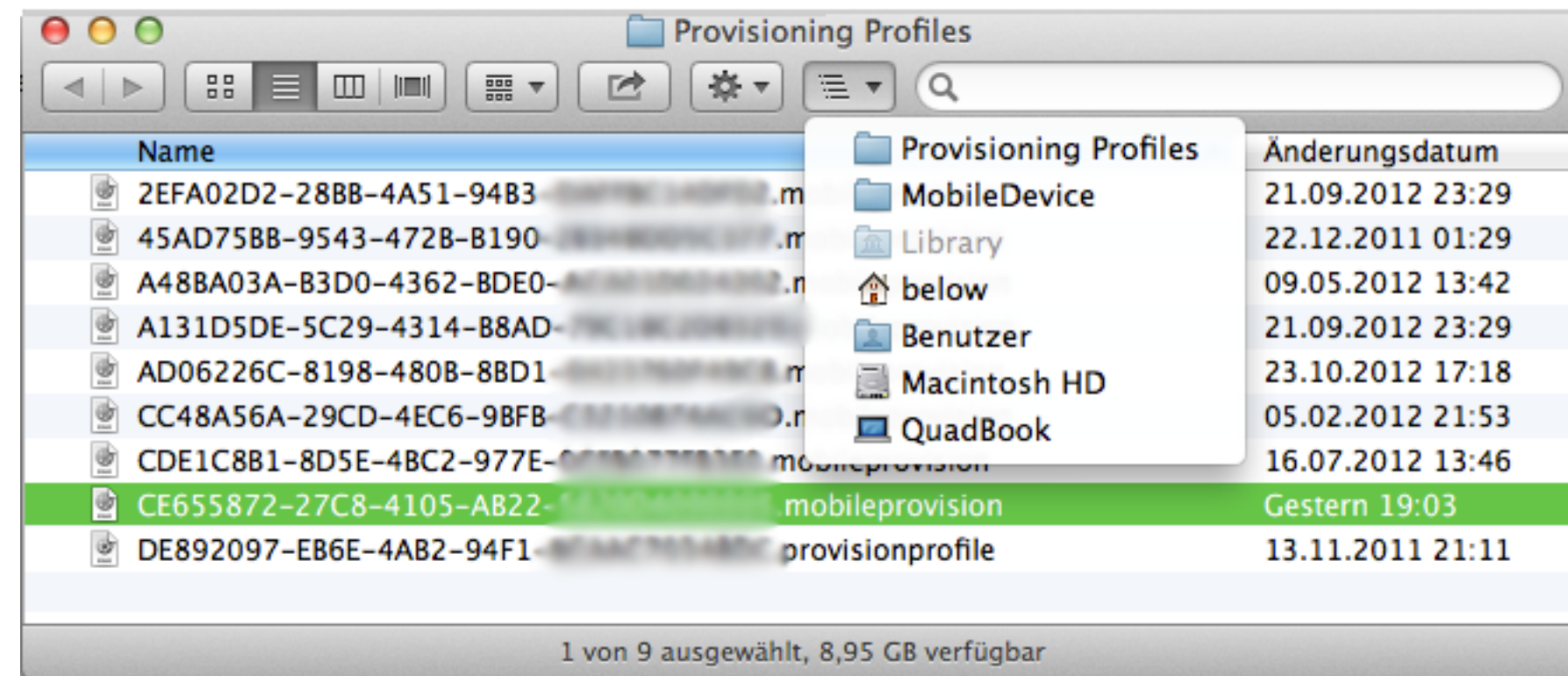
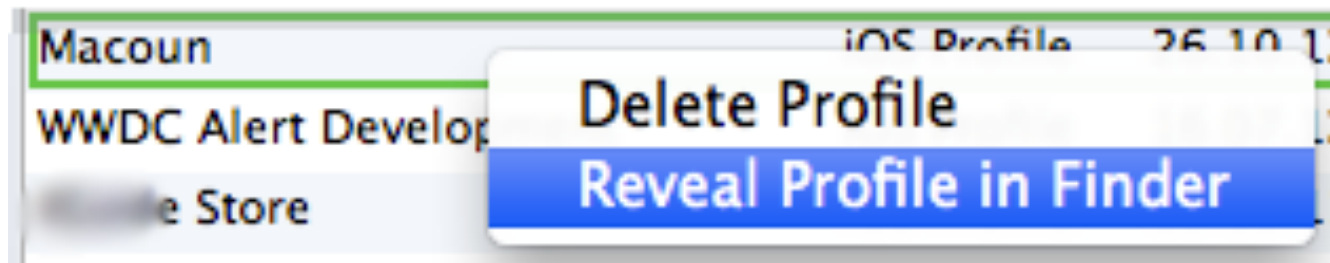
Excurs: Enterprise Certificates

- Erlauben das Deployment auf beliebig vielen Geräten
- UDID dieser Geräte muss nicht bekannt sein
 - Interne Apps, Tests, Museen, etc.
- Kann **nicht** für den App Store verwendet werden
- AppStore und Enterprise Zertifikat werden auf den selben Firmennamen ausgestellt

Profile in Xcode löschen

1. „Falsche“ Profile im Schlüsselbund löschen
2. Schlüsselbund schliessen
3. Profile im Xcode Organizer löschen

Profile verschieben



~/Library/MobileDevice/ProvisioningProfiles

Permanenterer Lösung

- Für Enterprise Zertifikate zwei User Accounts
- Möglicherweise bald ohnehin erledigt:

```
codesign --force  
--sign 1cccef965fda2fbec0fc668f15ae4fc2ca35c131  
--resource-rules=${BUILD_PATH}/CSTest.app/ResourceRules.plist  
--entitlements ${BUILD_PATH}/CSTest.build/CSTest.xcent
```



Hash statt Name

Protip: Export

- Über Schlüsselbund: Als .p12 (PKCS#12) Datei
- Über Xcode: Export aus dem Organizer

Protip 2: Kommandozeile

- codesign (l) braucht Zugriff auf die Keychain
- Leider wird das nicht als Fehlermeldung bei reinem SSH Zugriff aufgeführt

Abschluss

DU BIST NICHT ALLEIN!

DU BIST NICHT DOOF!

Codesigning schon...

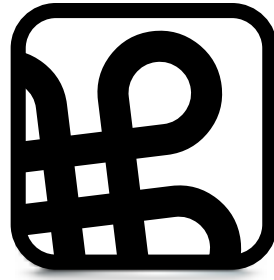
Apple Tech Note TN2250

Code Signing Troubleshooting

Die Werkstatt

Fragen?

Vielen Dank



Macoun