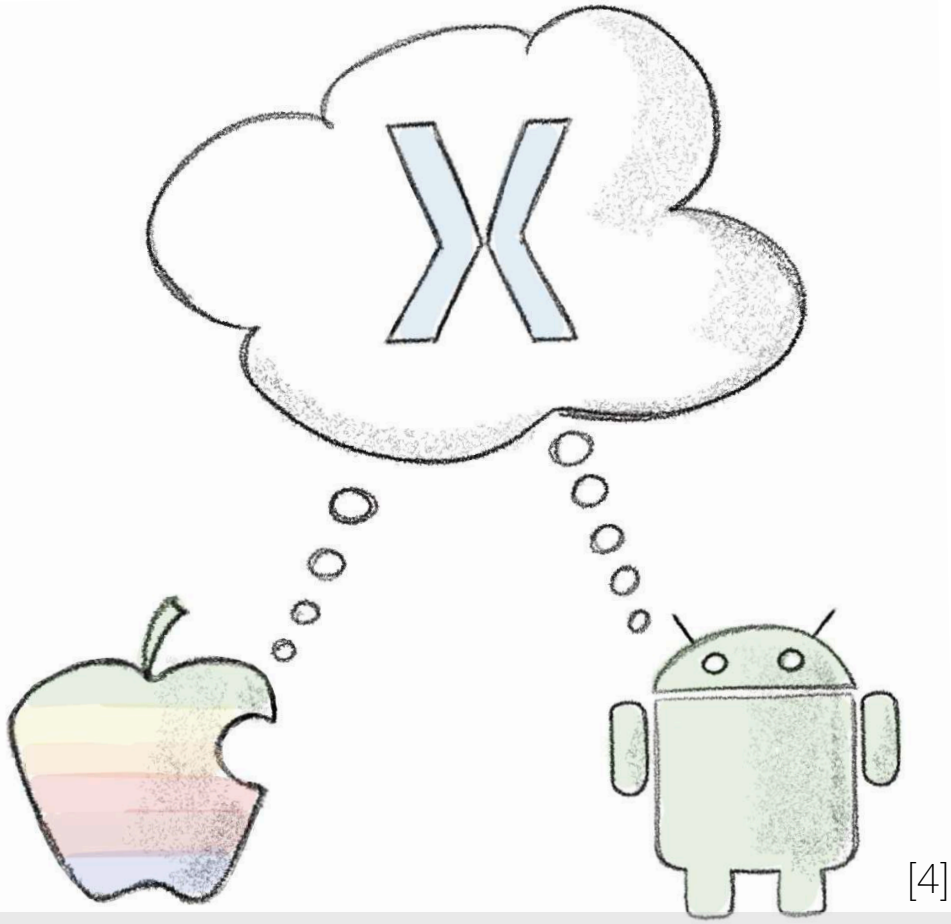
Der mobile Grabenkampf



[4]

Manu Rink
Tech Evangelist

Kerry Lothrop
Principal Consultant

DISCLAIMER

- 😳 Kein reiner iOS-Fokus
- 😏 Man wird Code sehen

- 👋 Jetzt gehen
- 🤘 oder bis zum Ende bleiben

EMOTIONAL

RATIONAL

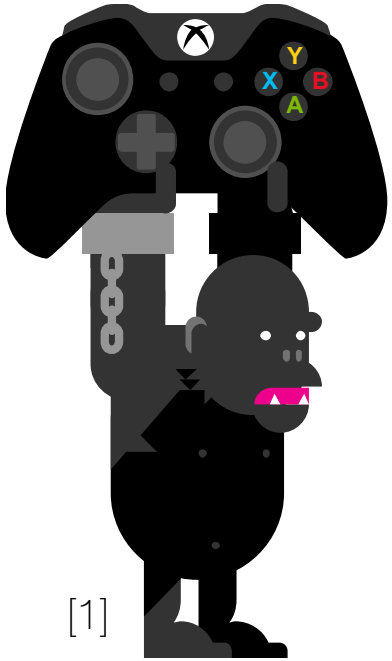
UNBEKANNT?!

CODE

Kapitel 0: Der EMOTIONALE Grabenkampf

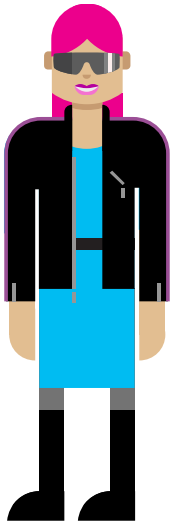
Echt jetzt… ein Grabenkampf?

Programmierung ist `$religiös, $emotional`…



[1]


[1]


[1]


[4]

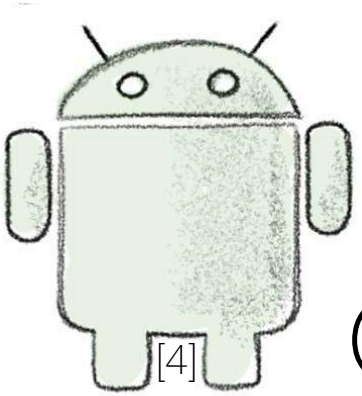
Coder

liebt seine macOS/iOS-“Welt”

“Die Grünen sind doof”

“Android Coder sind umkreative Bastler”

“Die kennen ja nicht mal ruckelfreies Scrolling”


[4]

Coder

stolz auf seine Techie-Plattform

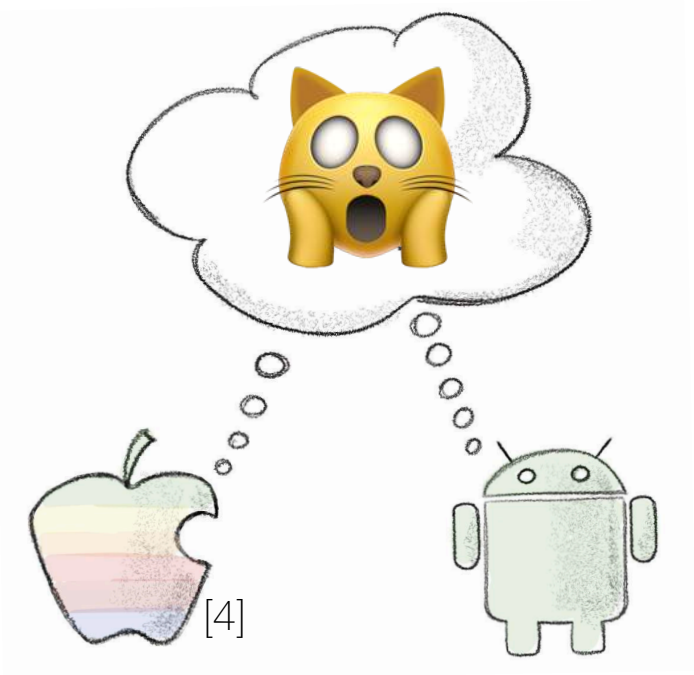
“Alles ist möglich! Auch wenns hart ist…”

“Ich liebe alle 374.173.738 Devicegrößen“

„APK ausliefern – einfach so!“

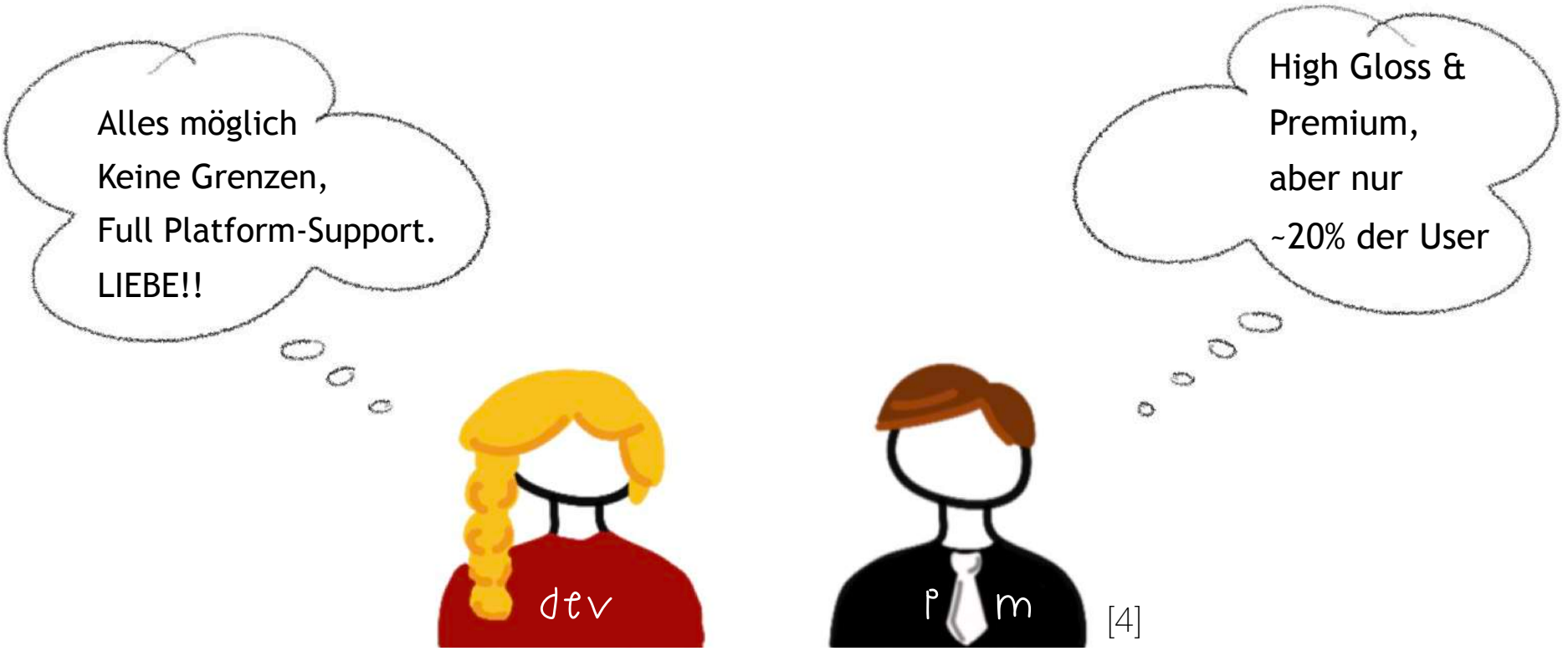
Cross-Plattform ist UNSEXY!

- 😋 nur hässliche Listen-Apps möglich
- 😣 Verzicht auf Features ist No-Go
- 😮 “High Gloss” wieder plattform-spezifisch
- 🤢 C# stinkt!


[4]

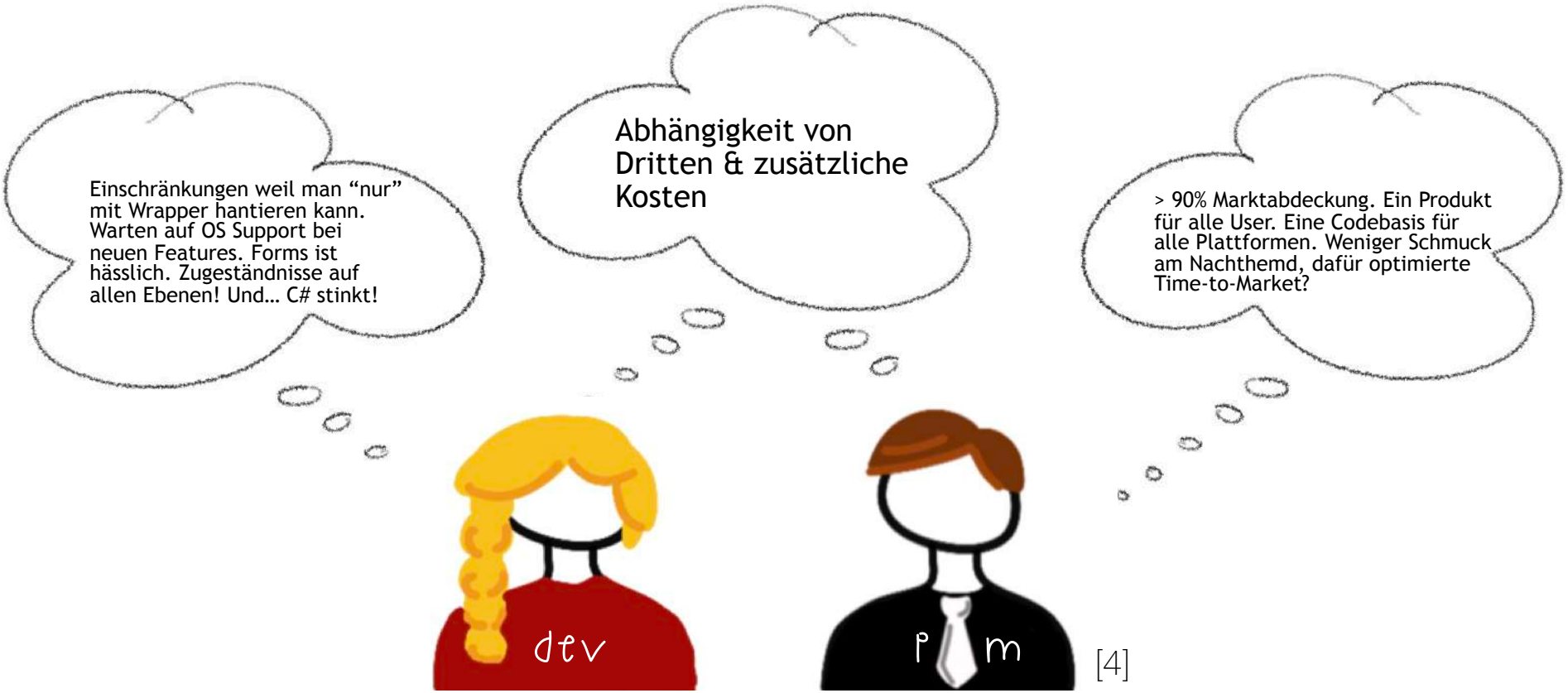
Kapitel 1: Der RATIONALE Grabenkampf

NATIVE!

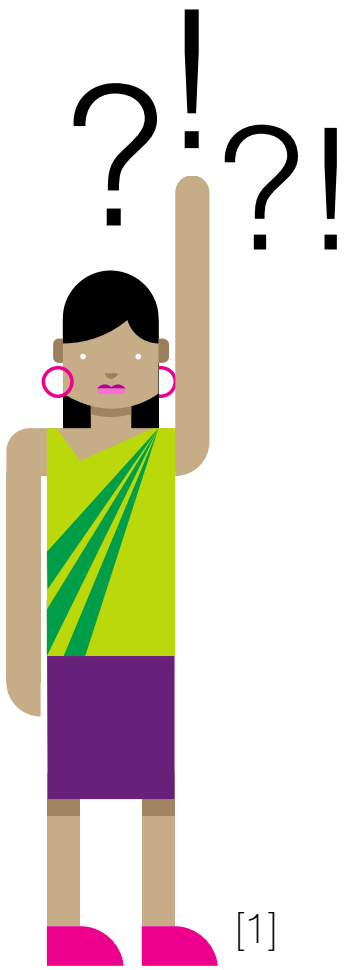

Alles möglich
Keine Grenzen,
Full Platform-Support.
LIEBE!!
High Gloss &
Premium,
aber nur
~20% der User
dev
pm

[4]

CROSS PLATFORM?



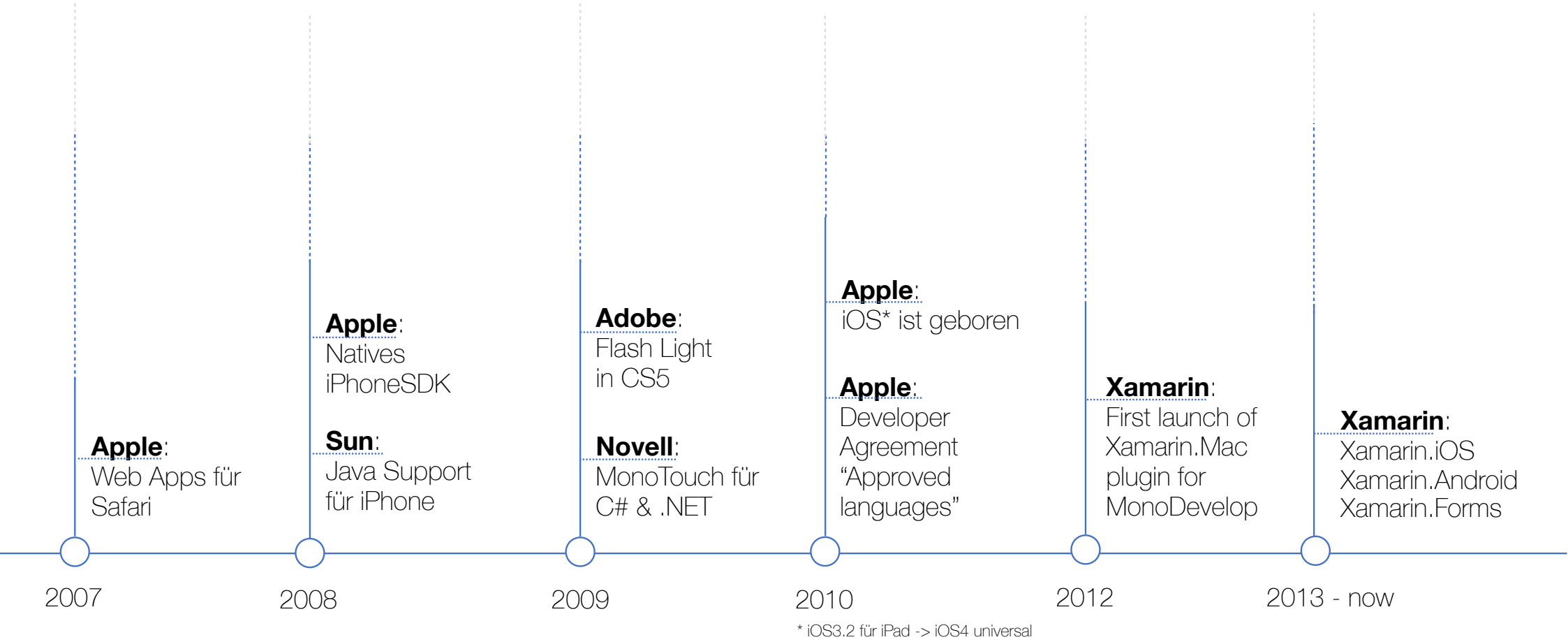
[4]



[1]

Richtiges 🔨 für das bestehende 🤦 finden

. . . WO KOMMT'S DENN HER?



* iOS3.2 für iPad -> iOS4 universal

Kapitel 2: Vereint gegen das UNBEKANNTE

NATIVE ↓
CROSS-PLATTFORM

Mono -> Ximian

↓

Xamarin = Tamarin Äffchen + X von Ximian

↑

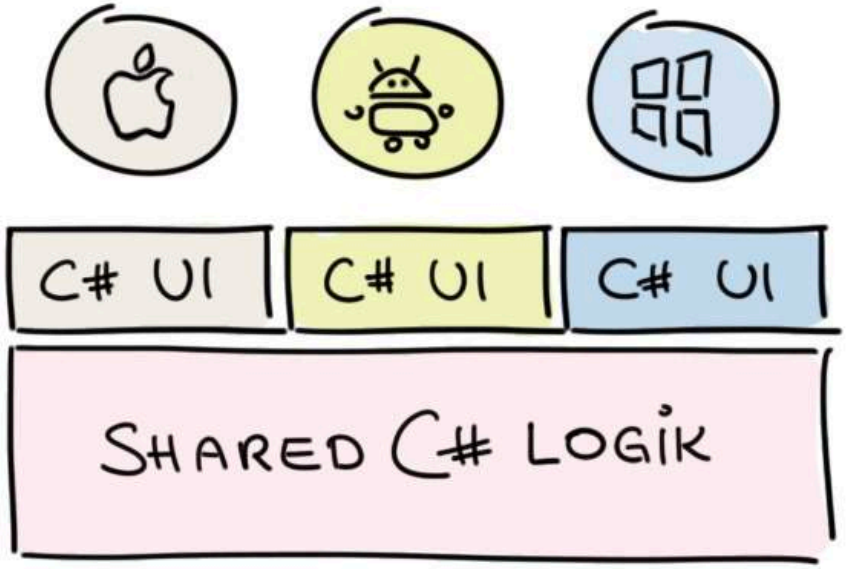
C#

[2]

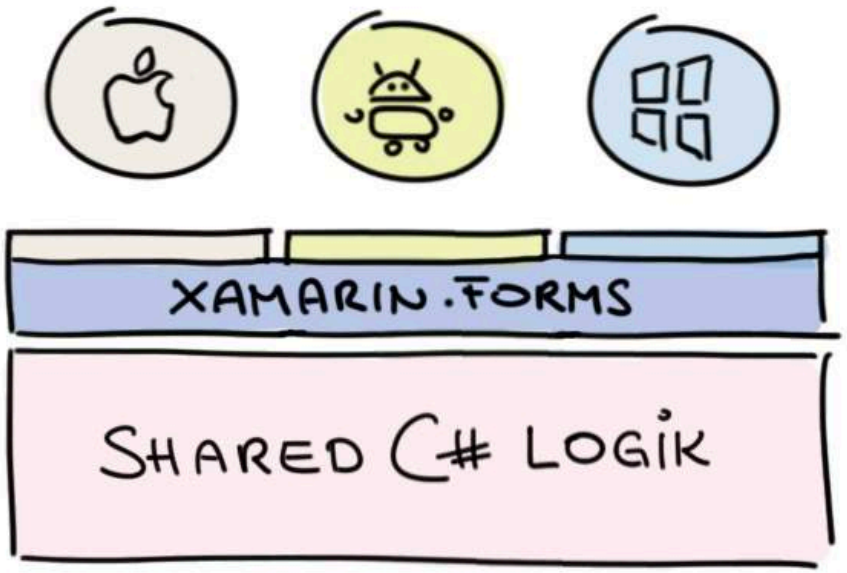
[3]

XAMARIN

Natives Cross-Platform mit zwei Ansätzen



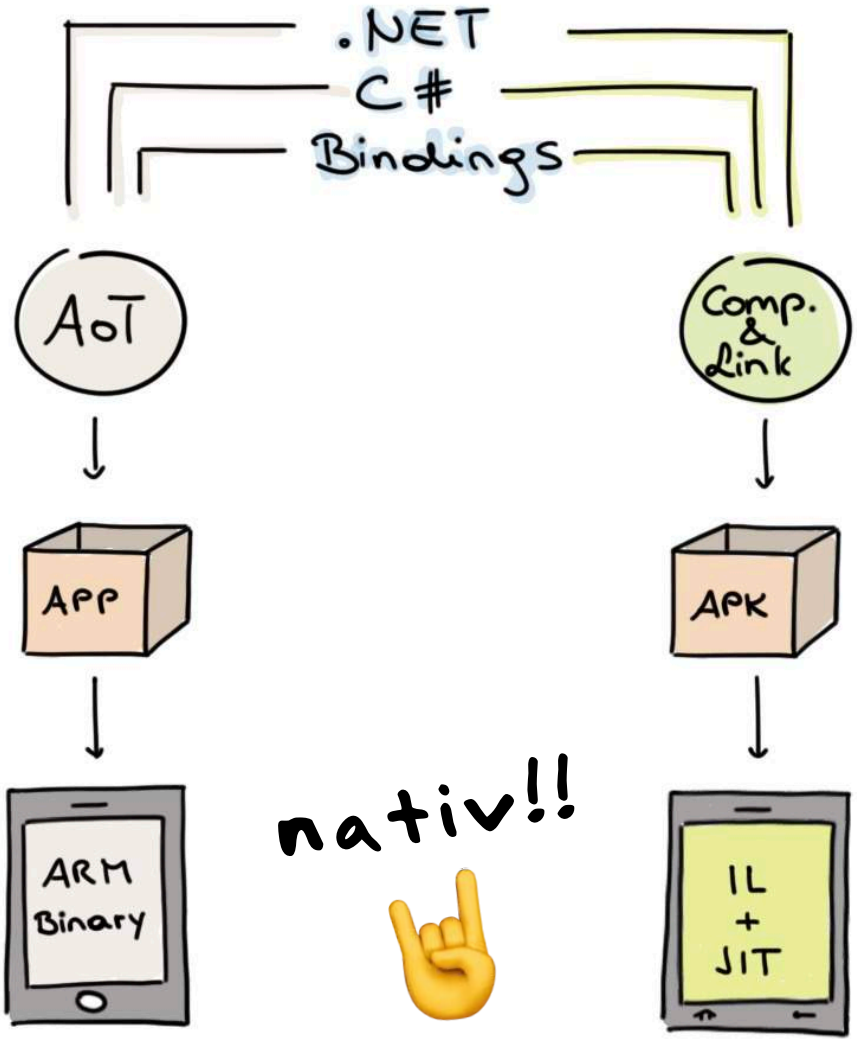
[4]



[4]

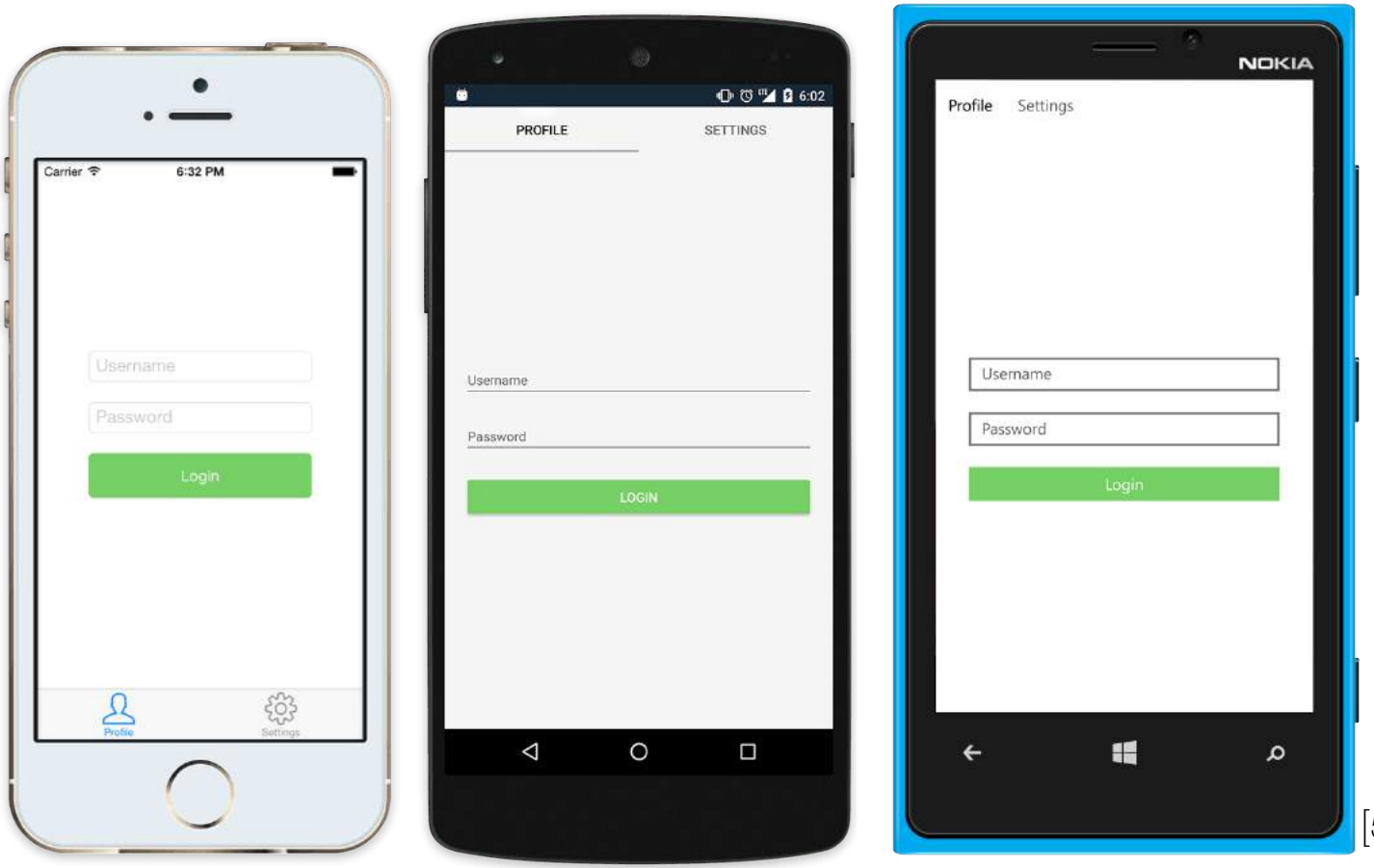
C# / .NET Apps auf
iOS und Android
...

NATIV ?!



[4]

XAMARIN.FORMS



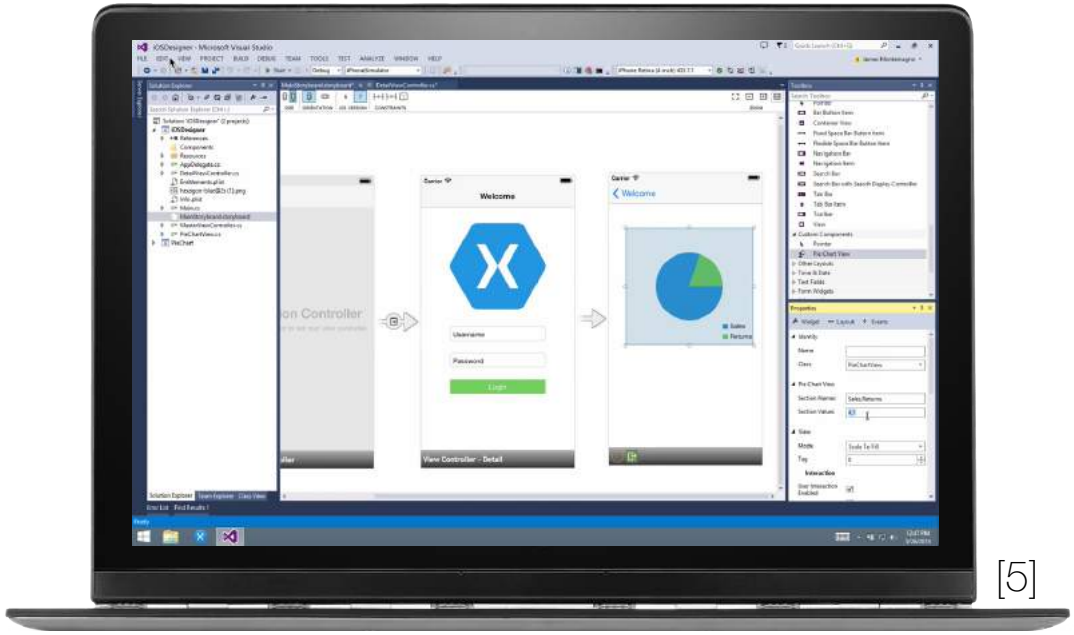
nativ!!

[5]

Kapitel 3: Und mein ❤️ Werkzeug?!

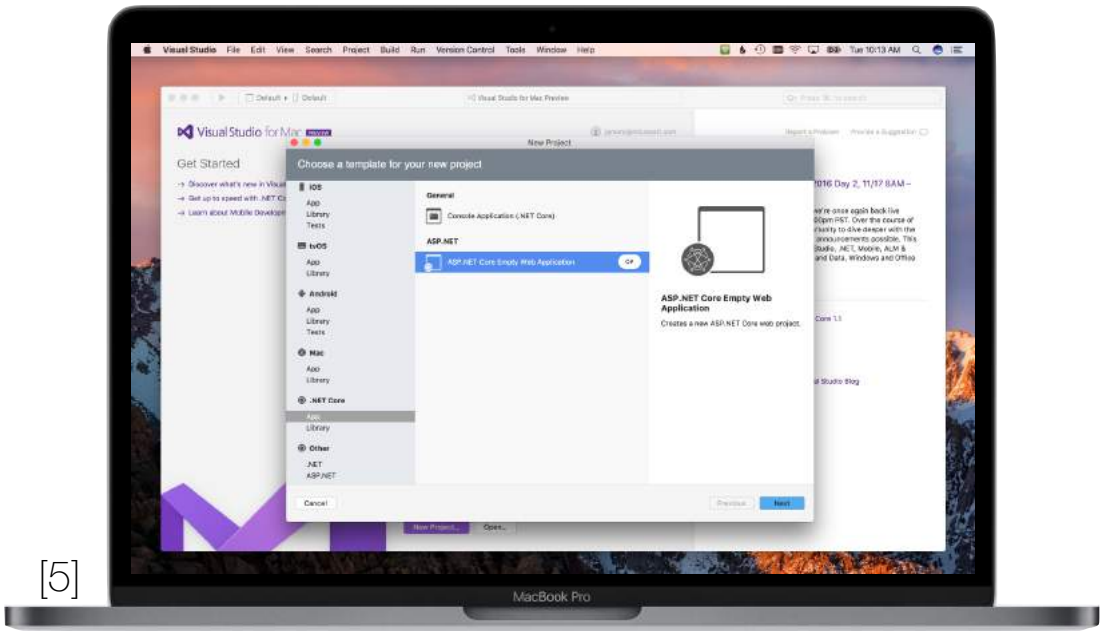
Visual Studio

for Windows



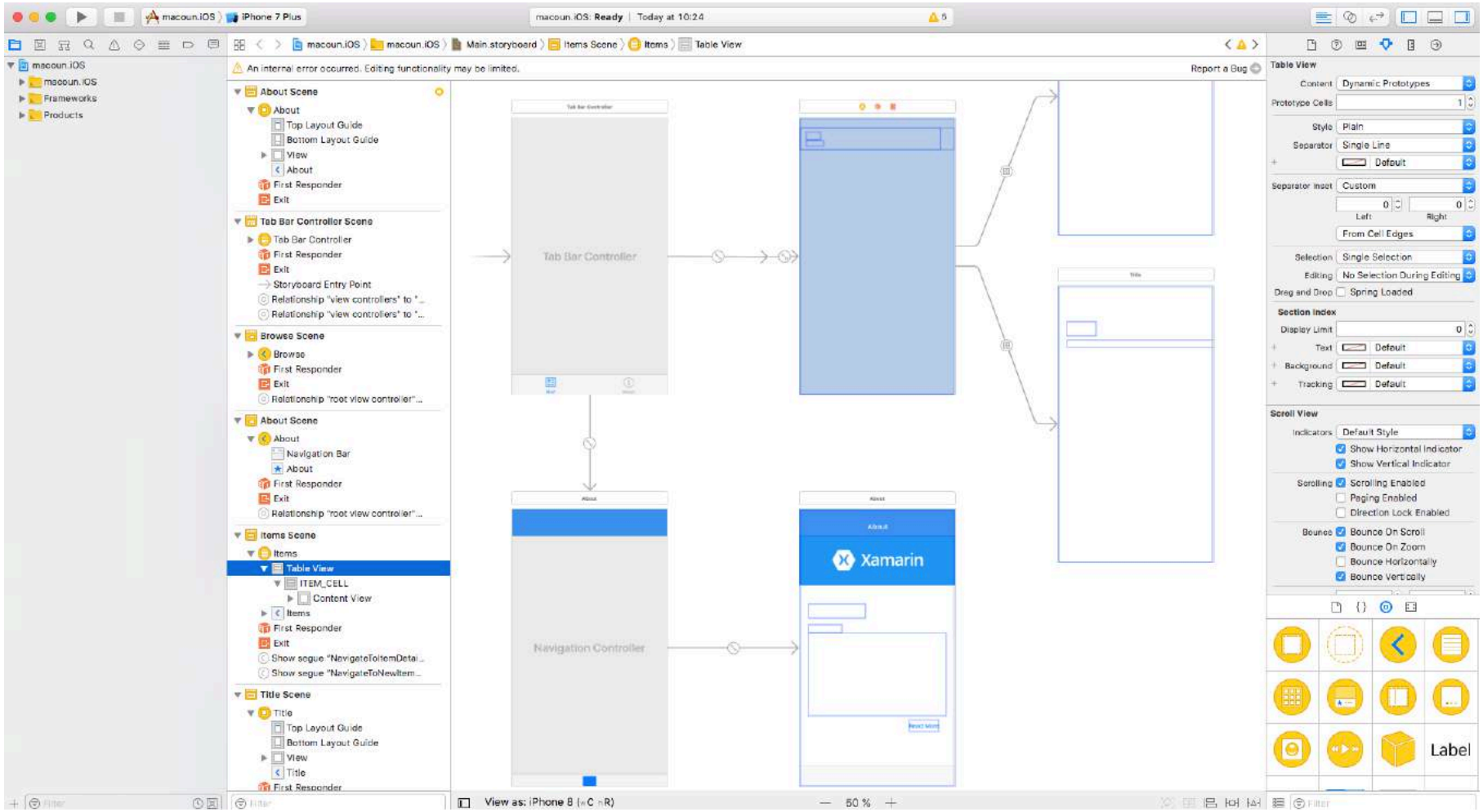
[5]

for Mac



[5]

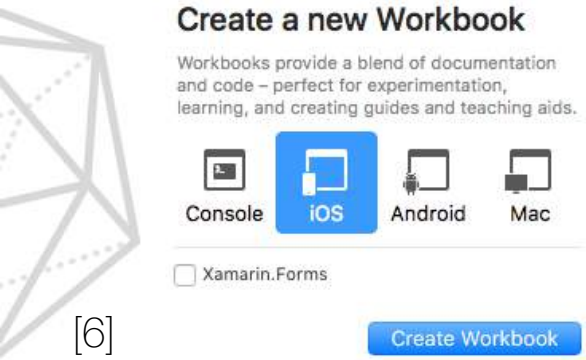
Xcode für IB



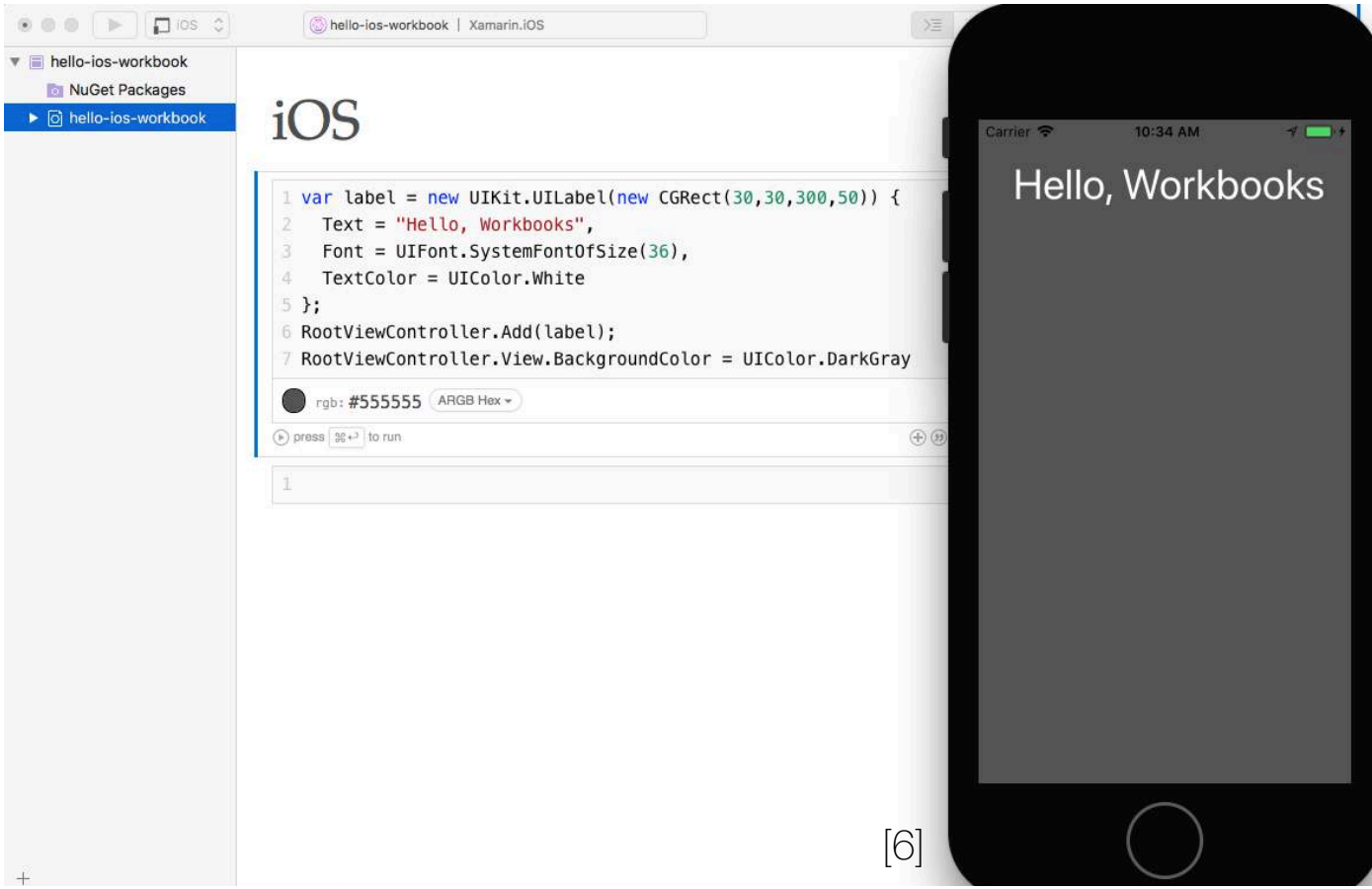
[6]

Xamarin Workbooks

Playgrounds für C#



[6]

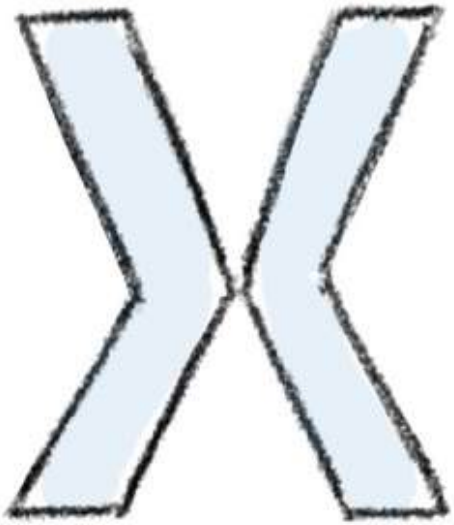


[6]

Kapitel 4: CODE als Friedensangebot!

[4]

VS.



[4]

ImageView & Images

C#

```
var image = UIImage.FromFile("monkeys.jpg")
var imageView = new UIImageView { Image = image };
View.AddSubview(imageView);
```

Swift

```
let myImage = UIImage(named:"appledroid")
let myImageView = UIImageView(image: myImage)
view.addSubview(myImageView)
```

Networking

C#

```
try
{
    var httpClient = new HttpClient();
    string result = await httpClient
        .GetStringAsync("http://whatthecommit.com/index.txt");
    TextView.Text = result;
}
catch (Exception exception)
{
    TextView.Text = $"Big UPS: {exception}";
}
```

Swift

```
let url = URL(string: "http://whatthecommit.com/index.txt")

let request = NSMutableURLRequest(url: url!)
let session = URLSession.shared

let task = session.dataTask(with: request as URLRequest) { (data, response, error) in
    if let error = error {
        print("Big UPS: \(error)")
    }

    if let data = data,
        let myData = String(data: data, encoding:String.Encoding.utf8) {
            DispatchQueue.main.async {
                self.textView.text = myData
            }
        }
    }
}
```

Animation

C#

```
public static async Task AnimateShakeAsync(UIView view)
{
    await AnimateHorizontalMovementAsync(view, -10);
    await AnimateHorizontalMovementAsync(view, 20);
    await AnimateHorizontalMovementAsync(view, -20);
    await AnimateHorizontalMovementAsync(view, 20);
    await AnimateHorizontalMovementAsync(view, -15);
    await AnimateHorizontalMovementAsync(view, 10);
    await AnimateHorizontalMovementAsync(view, -5);
}

private static async Task AnimateHorizontalMovementAsync(UIView view, nfloat horizontalOffset)
{
    await UIView.AnimateAsync(0.065, () => view.Frame =
        new CGRect(new CGPoint(view.Frame.Left + horizontalOffset, view.Frame.Top),
                   view.Frame.Size));
}
```

Swift

```
func animateTheAppleDroid () {
    UIView.animate(withDuration: 0.3, animations: {
        self.animating = true
        self.imageViewBottom.frame.origin.x -= 10
    }) { completed in
        UIView.animate(withDuration: 0.3, animations: {
            self.imageViewBottom.frame.origin.y -= 5
        }) { completed in
            UIView.animate(withDuration: 0.3, animations: {
                self.imageViewBottom.frame.origin.x += 5
            }) { completed in
                UIView.animate(withDuration: 0.3, animations: {
                    self.imageViewBottom.frame.origin.y += 15
                }) { completed in
                    UIView.animate(withDuration: 0.3) {
                        self.imageViewBottom.frame.origin.x += 5
                        self.imageViewBottom.frame.origin.y -= 10

                        self.animating = false
                    }
                }
            }
        }
    }
}
```

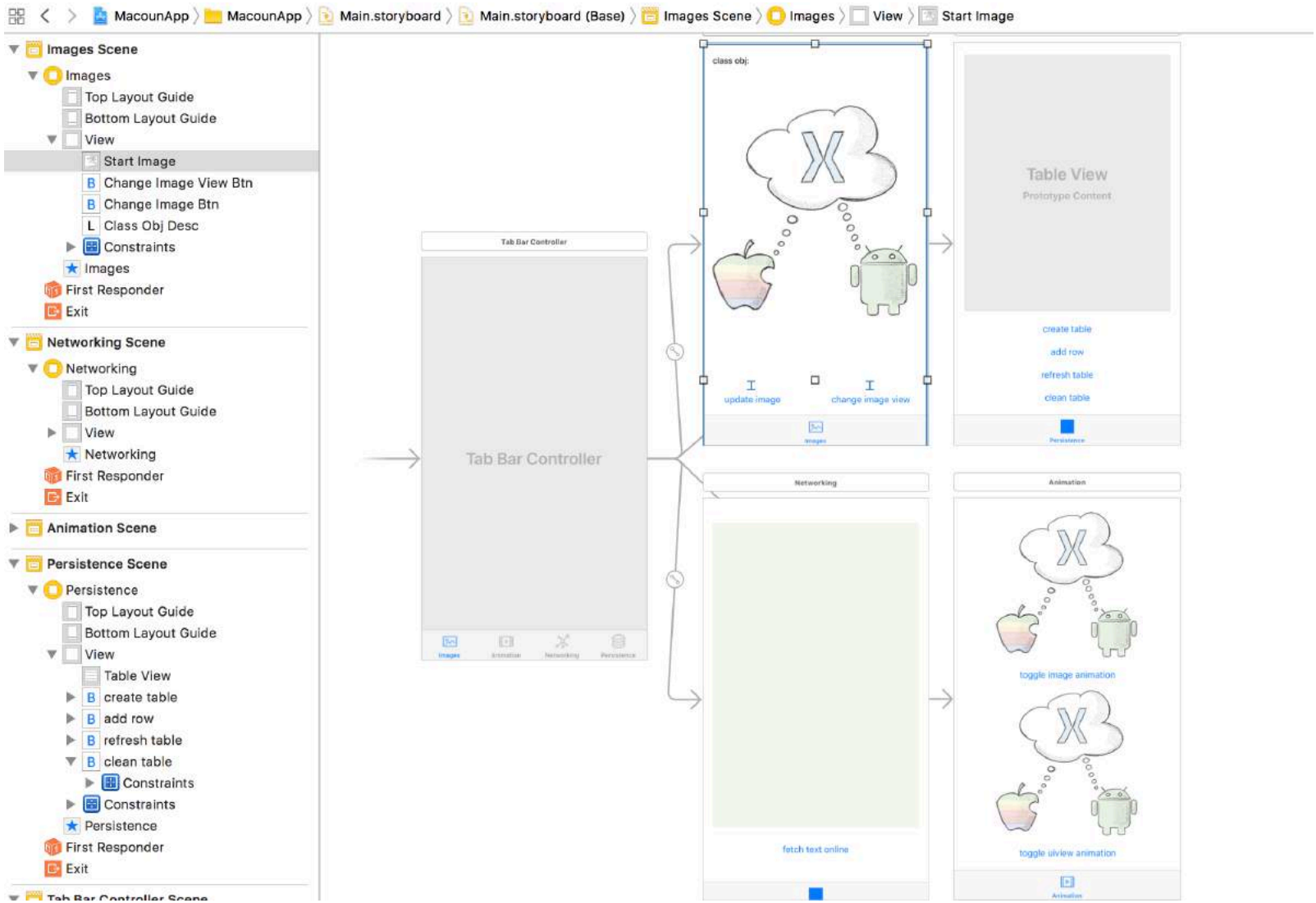
Persistence

C# with sqlite-net-pcl [7]

```
public class Stock
{
    [PrimaryKey, AutoIncrement]
    public int Id { get; set; }
    public string Symbol { get; set; }
}

...

var conn = new SQLiteConnection(databasePath);
conn.CreateTable<Stock>();

...

var query = conn.Table<Stock>().Where(v => v.Symbol.StartsWith("A"));
string stocks = string.Join(", ", query.Select(stock => '$' + stock));
```

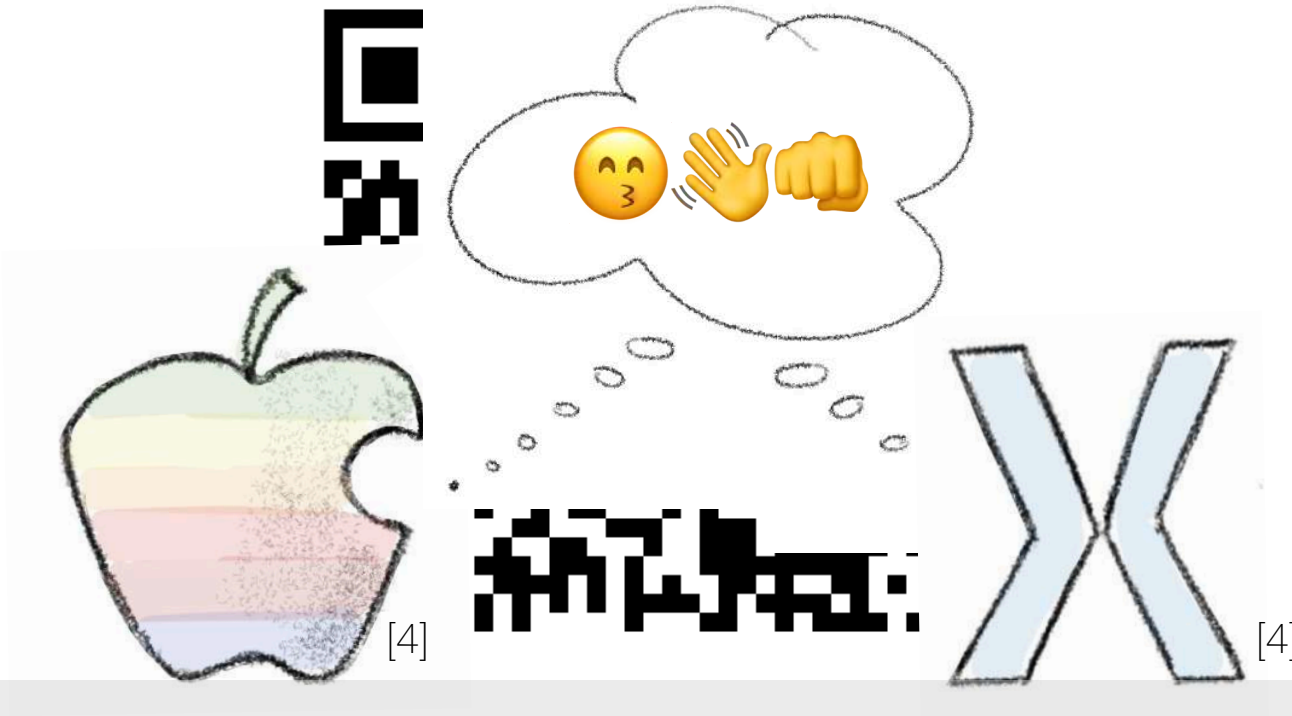
Swift with SQLite [8]

```
var db = try! Connection()
let users = Table("users")
let id = Expression<Int64>("id")
let name = Expression<String?>("name")
let email = Expression<String?>("email")

...

try! db.run(users.create(ifNotExists: true) { t in
    t.column(id, primaryKey: true)
    t.column(name)
    t.column(email)
})

...

for user in try! db.prepare(users) {
    let entry = "id: \(user[id]), name: \(user[name]!),
        mail: \(user[email]!)"
    items.append(entry)
}
```



https://github.com/codePrincess/macoun2017

Kapitel 5: Ein PROOOOSIT ! Ein Fazit!



Manu Rink
Tech Evangelist
@codeprincess

Kerry Lothrop
Principal Consultant
@kwlothrop

Quellen

[1] Microsoft Illustration Library ’16
verfügbar für Mitarbeiter und freigegeben für Präsentationszwecke

[2] Tamarin Äffchen
https://en.wikipedia.org/wiki/Tamarin
Brocken Inaglory, edited by Fir0002, edited by Brocken Inaglory - Own work

[3] Xamarin Äffchen
Photo von Manuela Rink

[4] Doodles
Handgezeichnet von Manuela Rink

[5] Tooling Images
Aus der Dokumentation von Micosoft/Xamarin

[6] Tooling Screenshots
Screenshots von Microsoft Tools von Manuela Rink

[7] SQLite.Net
https://www.nuget.org/packages/sqlite-net-pcl/

[8] SQLite Repository Link
https://github.com/stephencelis/SQLite.swift