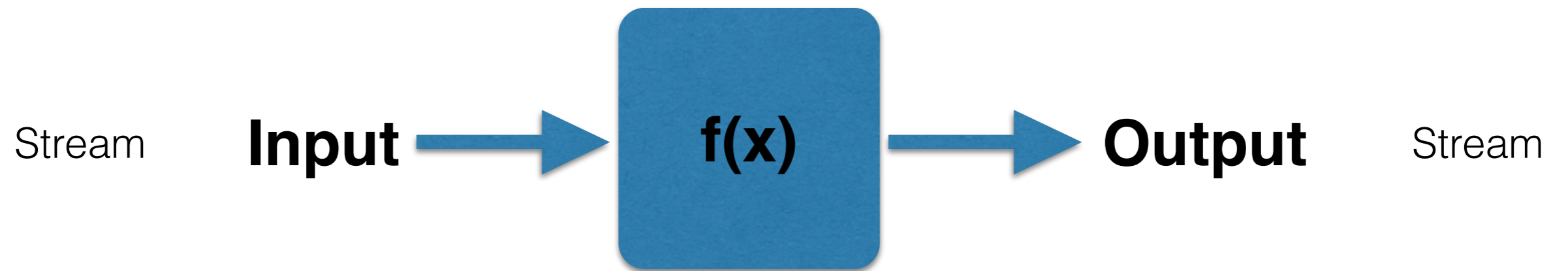


ReactiveCocoa

Bessere Abstraktionen für interaktive Programme.

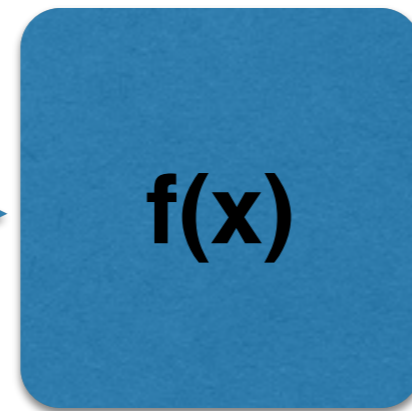
Damals...



...wie heute?

UI Event
Webservice
Datenbank
iCloud
File Presenter
GPS
...

Input



Output

UI State
Webservices
Datenbank
Dateien
...

KVO
Delegates

Bindings

GCD
Notifications

- Konzeptionelle Unterschiede
- Schwer zu kombinieren
- *Irgendwie* immer Stateful
- Code-Lokalität

Reactive Programming

Zeitlich veränderliche Variablen.

a = 2

b = 2

c = a + b = 4

b = 3

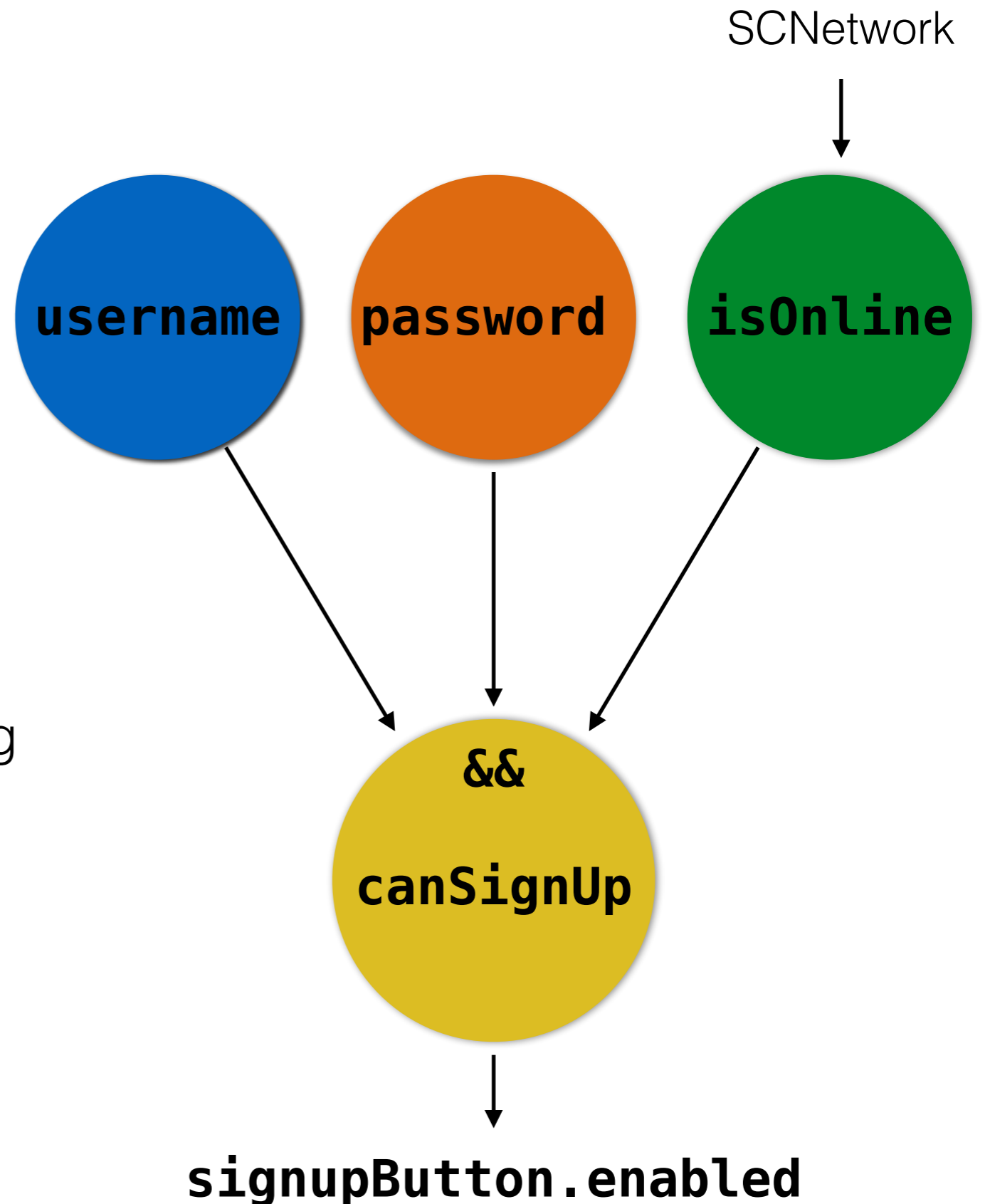
c = ~~4~~ 5

Spreadsheet-Programming

ReactiveCocoa

Signale

- Zeitlich veränderliche Variablen
- Beliebig kombinierbar
- Automatische Neuberechnung
- Binding an Properties, Methoden etc. möglich



Einfacher Observer

```
// Neues Signal aus Property anlegen  
RACSignal *mySignal = RACObserve(self, username);
```

```
// Log, wann immer die Property sich ändert  
[mySignal subscribeNext:^(NSString *newName) {  
    NSLog(@"%@@", newName);  
}];
```

Kombinieren

```
RACSignal *confirmedSignal =
```

```
[RACSignal
```

```
    combineLatest:
```

```
        @[ RACObserve(self, password),  
          RACObserve(self, confirmation) ]
```

```
    reduce:^(NSString *password, NSString *confirm) {  
        return @[confirm isEqualToString: password];  
    }];
```

Binden

```
RACSignal *confirmedSignal = ...;
```

```
RAC(self.signupButton, enabled) = confirmedSignal;
```

u.v.m.

- Viele weitere Kombinationen:
map, flattenMap, ...
- Asynchrone Programmierung:
Chaining und Splitting
- Subjekte:
Brücke zur nicht-reaktiven Welt
- Fehlerbehandlung

Demo