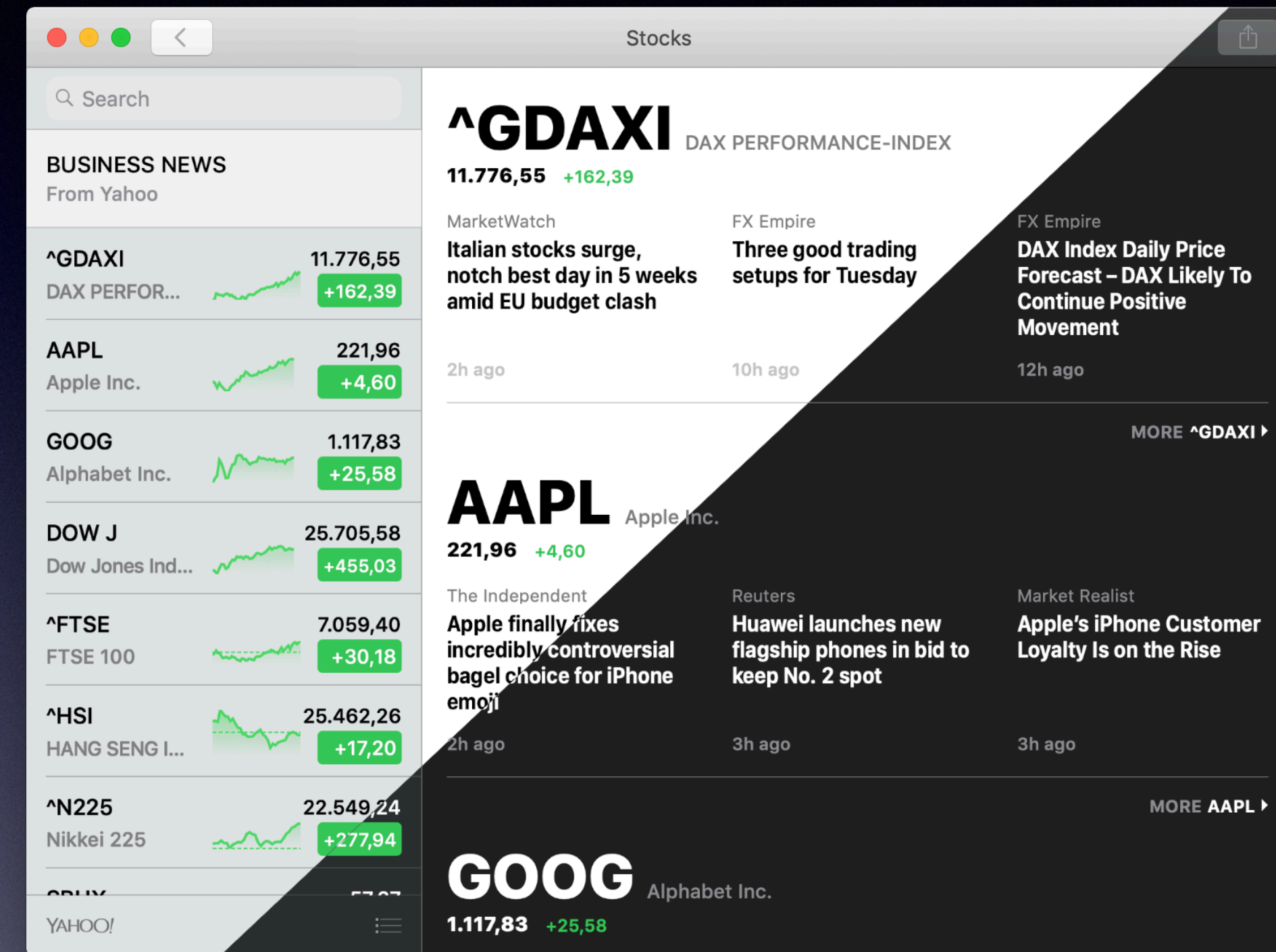
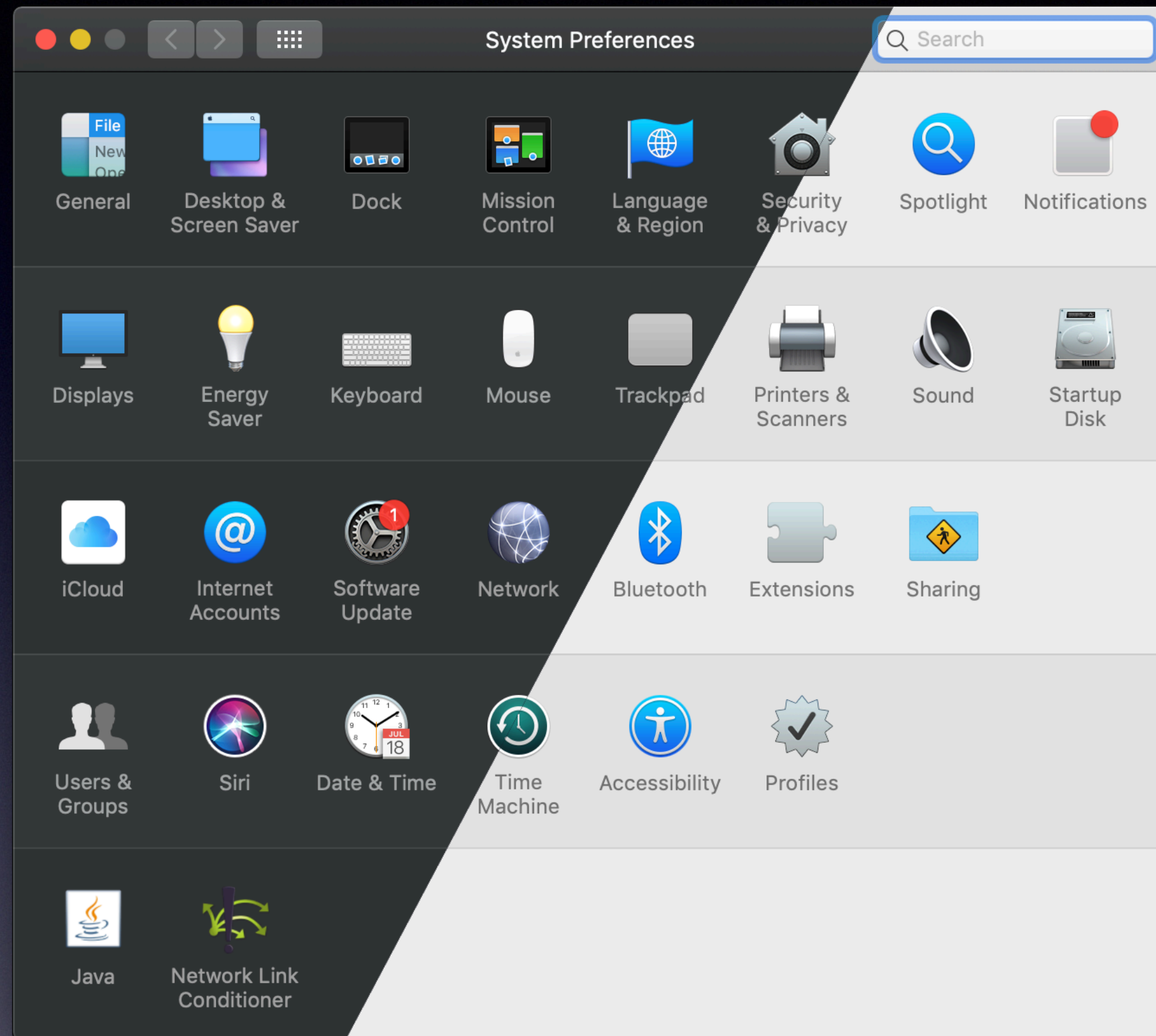


Dark Mode unter macOS

Mojave


Florian Lücke




Stocks
↑

BUSINESS NEWS
From Yahoo


^GDAXI
DAX PERFOR... 11.776,55
+162,39




AAPL
Apple Inc. 221,96
+4,60




GOOG
Alphabet Inc. 1.117,83
+25,58




DOW J
Dow Jones Ind... 25.705,58
+455,03



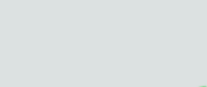
^FTSE
FTSE 100 7.059,40
+30,18




^HSI
HANG SENG I... 25.462,26
+17,20



^N225
Nikkei 225 22.549,24
+277,94



CRUX 57,07



YAHOO! ⋮

^GDAXI

DAX PERFORMANCE-INDEX

11.776,55 +162,39

MarketWatch

Italian stocks surge, notch best day in 5 weeks amid EU budget clash

2h ago

FX Empire

Three good trading setups for Tuesday

10h ago

FX Empire

DAX Index Daily Price Forecast – DAX Likely To Continue Positive Movement

12h ago

[MORE ^GDAXI ▶](#)

AAPL

Apple Inc.

221,96 +4,60

The Independent

Apple finally fixes incredibly controversial bagel choice for iPhone emoji

2h ago

Reuters

Huawei launches new flagship phones in bid to keep No. 2 spot

3h ago

Market Realist

Apple's iPhone Customer Loyalty Is on the Rise

3h ago

[MORE AAPL ▶](#)

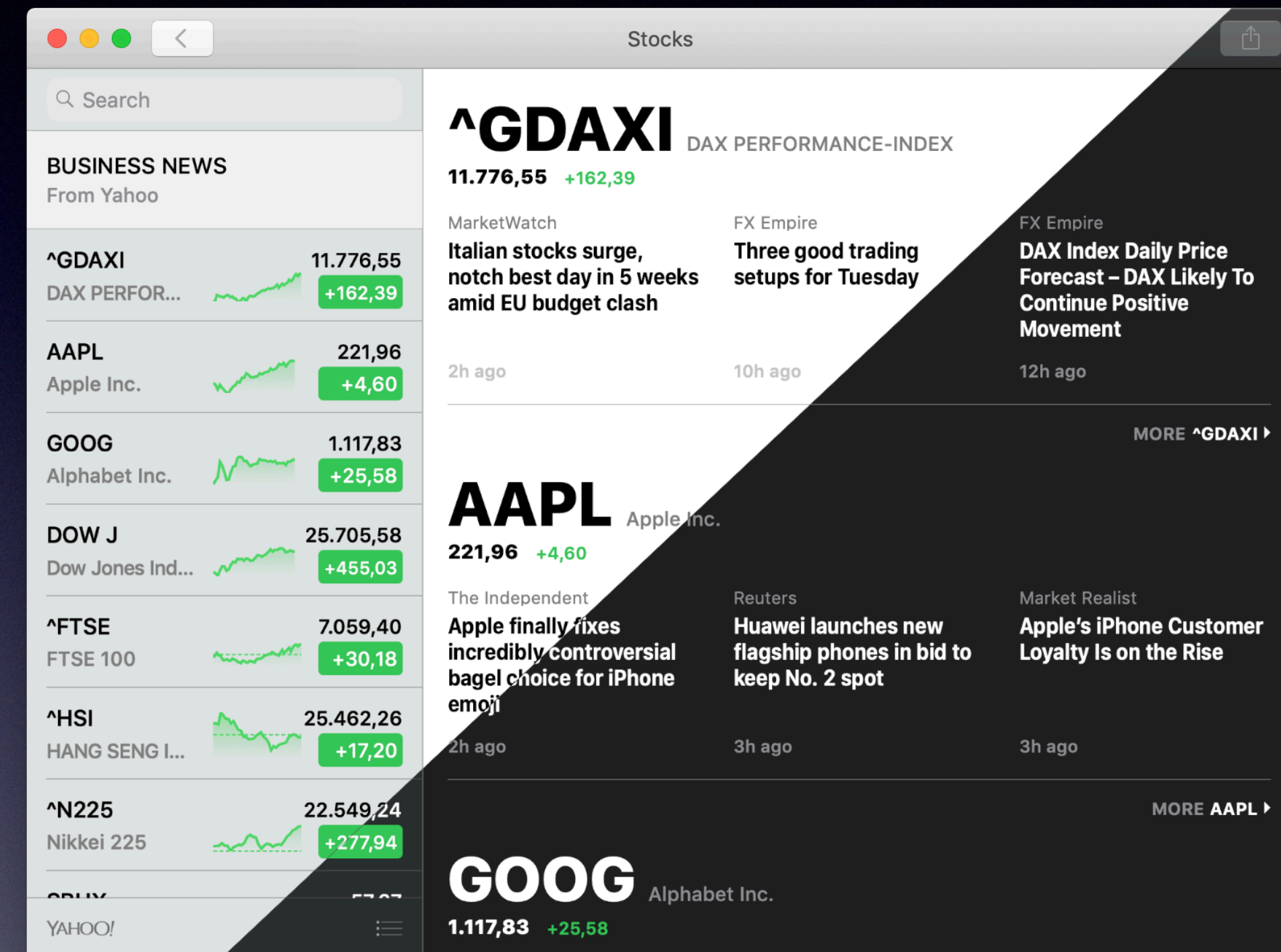
GOOG

Alphabet Inc.

1.117,83 +25,58

Dark Mode auch in UIKit für macOS

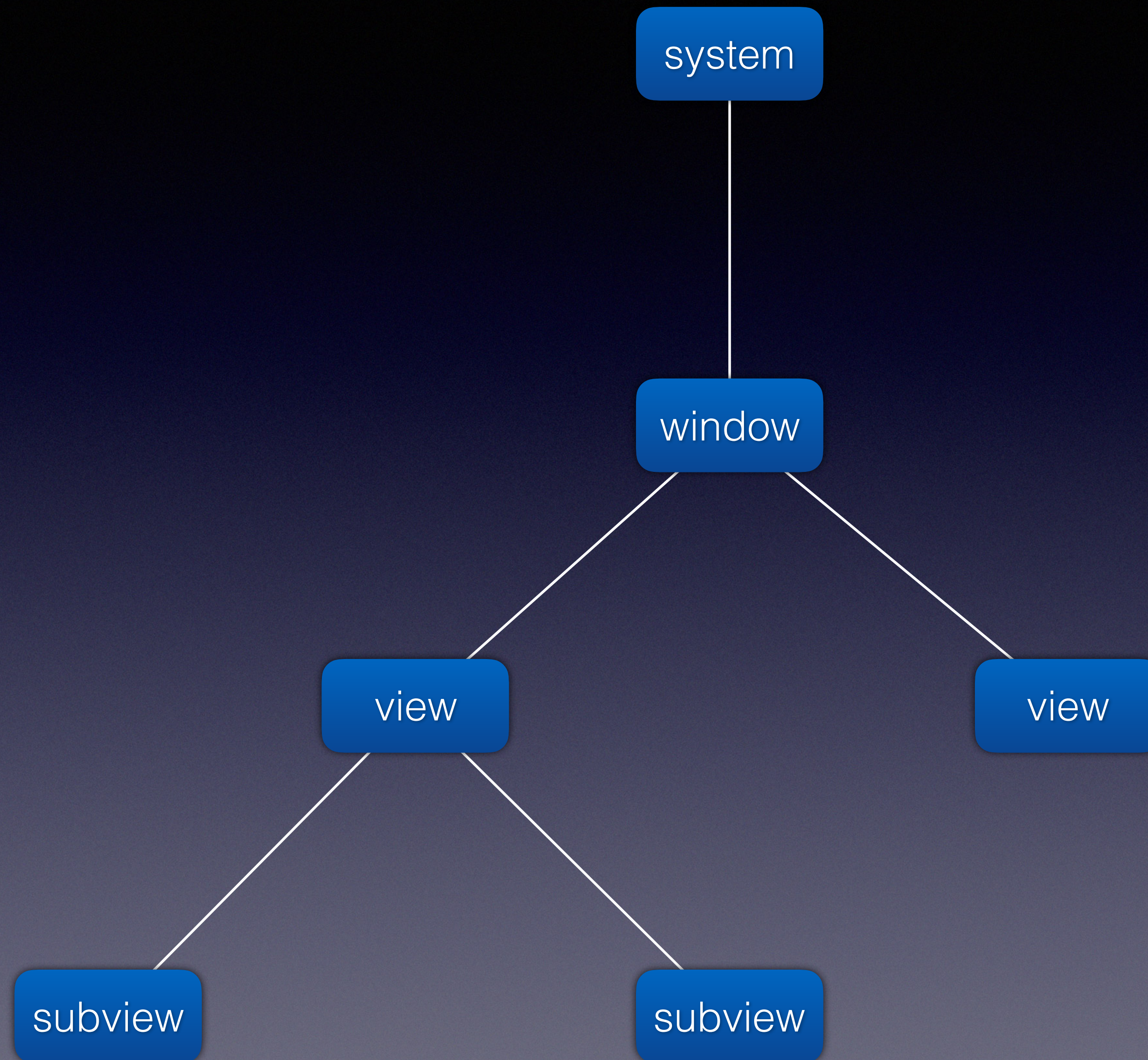
Vielleicht bald Dark Mode für iOS?

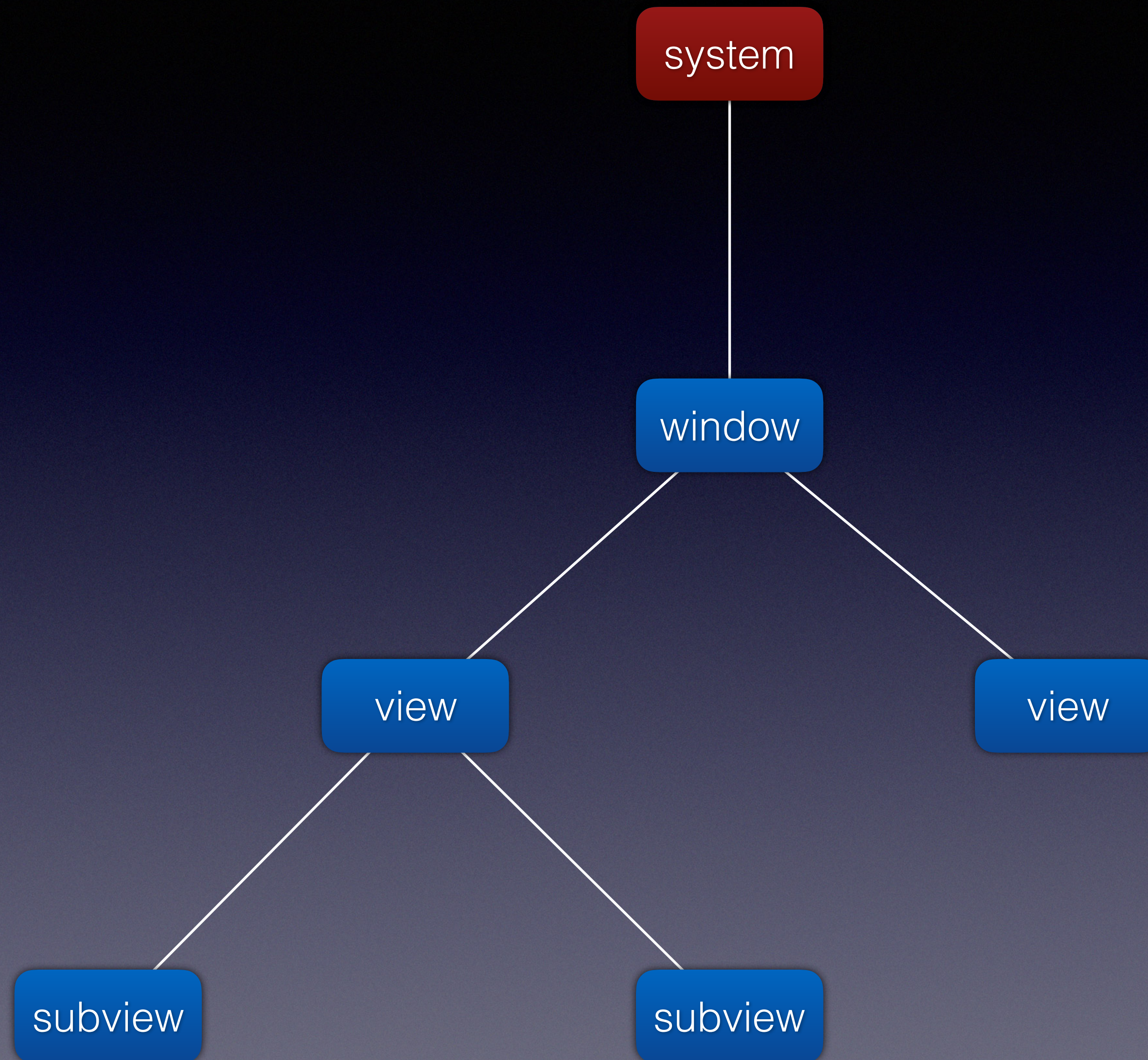


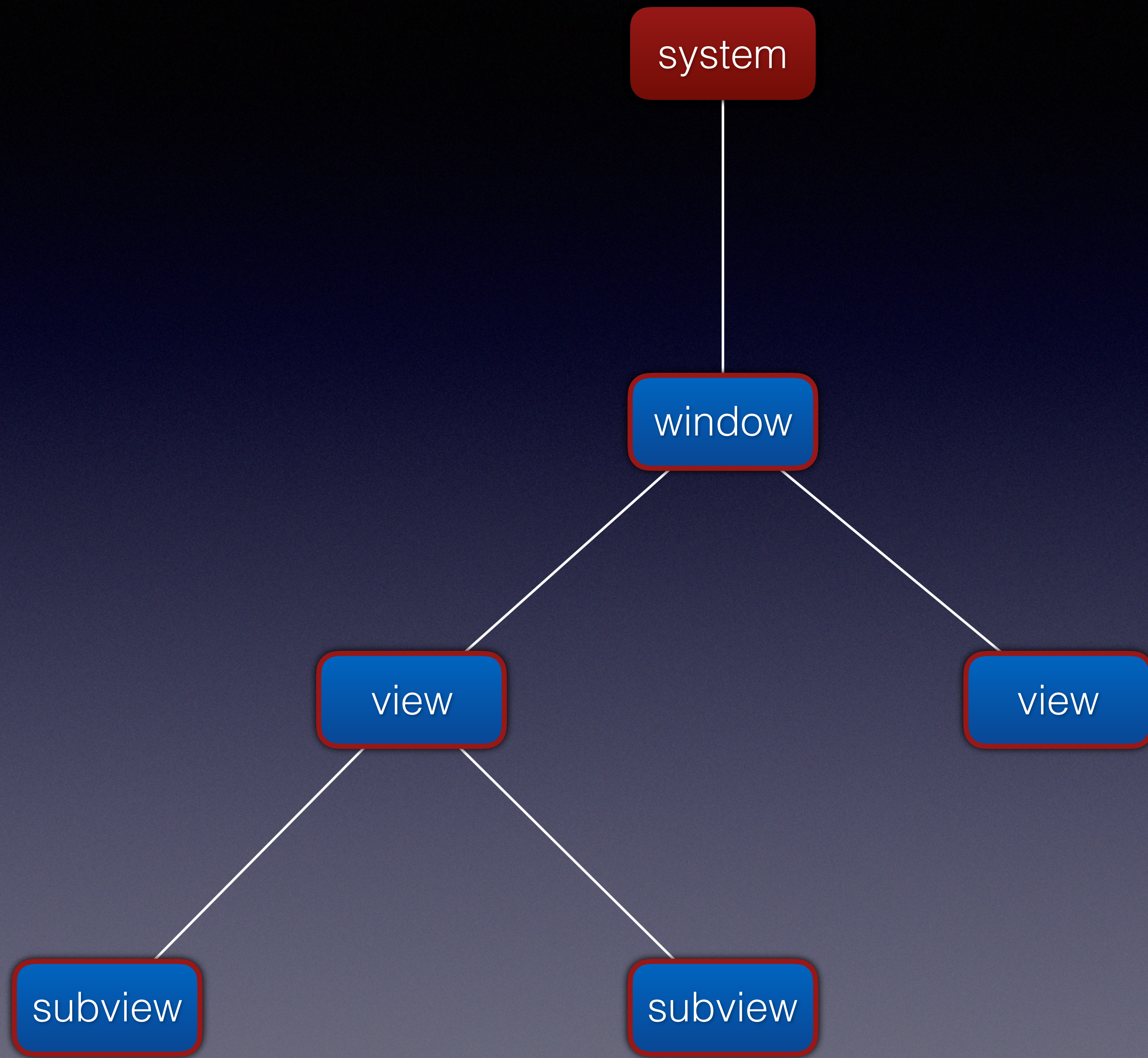
Wie funktioniert Dark Mode?

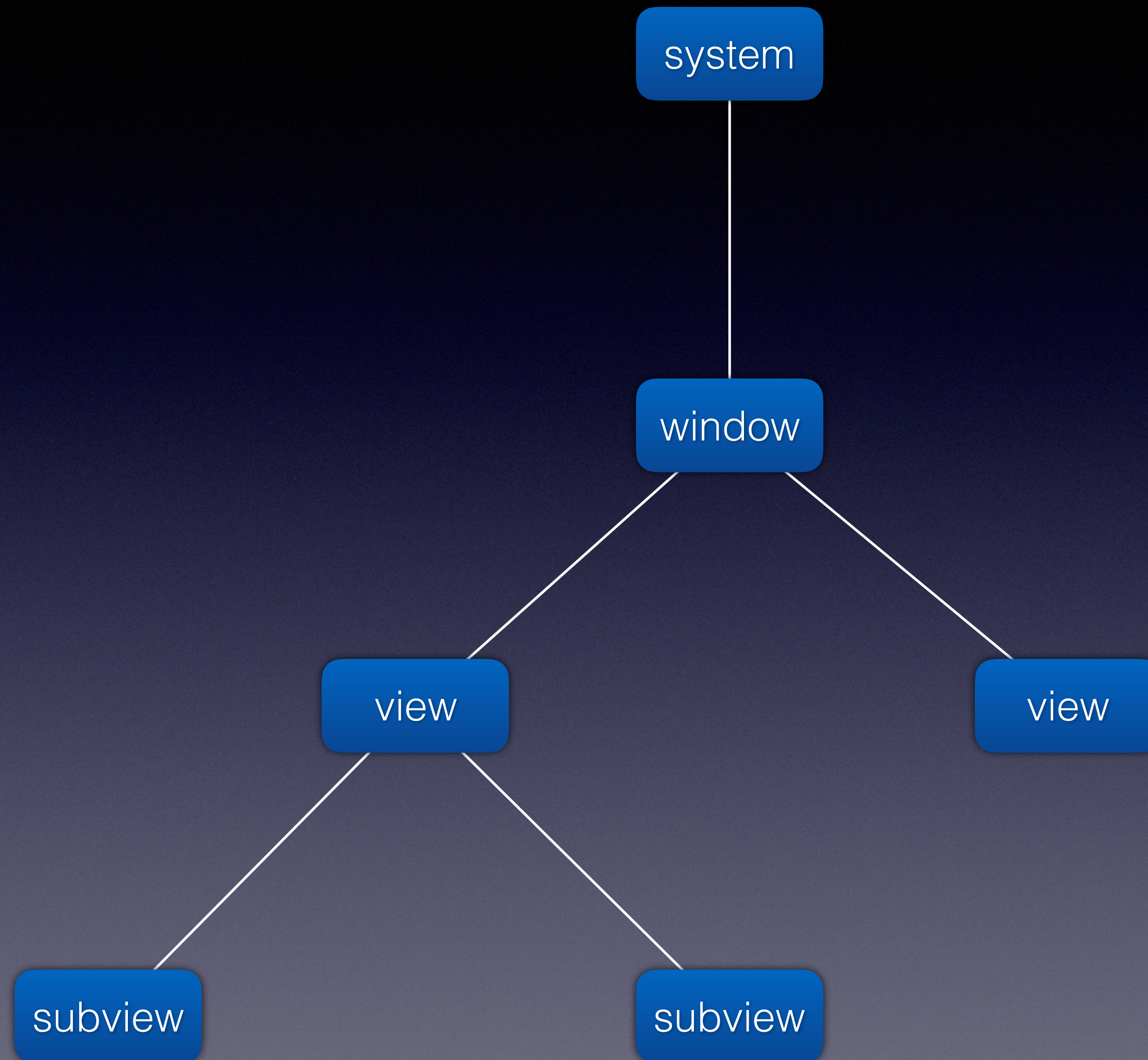
- Basiert auf **NSAppearance** (verfügbar ab macOS 10.9), “An object that manages standard appearance attributes for UI elements in an app.”
- Properties **appearance** und **effectiveAppearance** für **NSApplication**, **NSWindow**, **NSView** und **NSPopover** um Appearance zu beeinflussen.

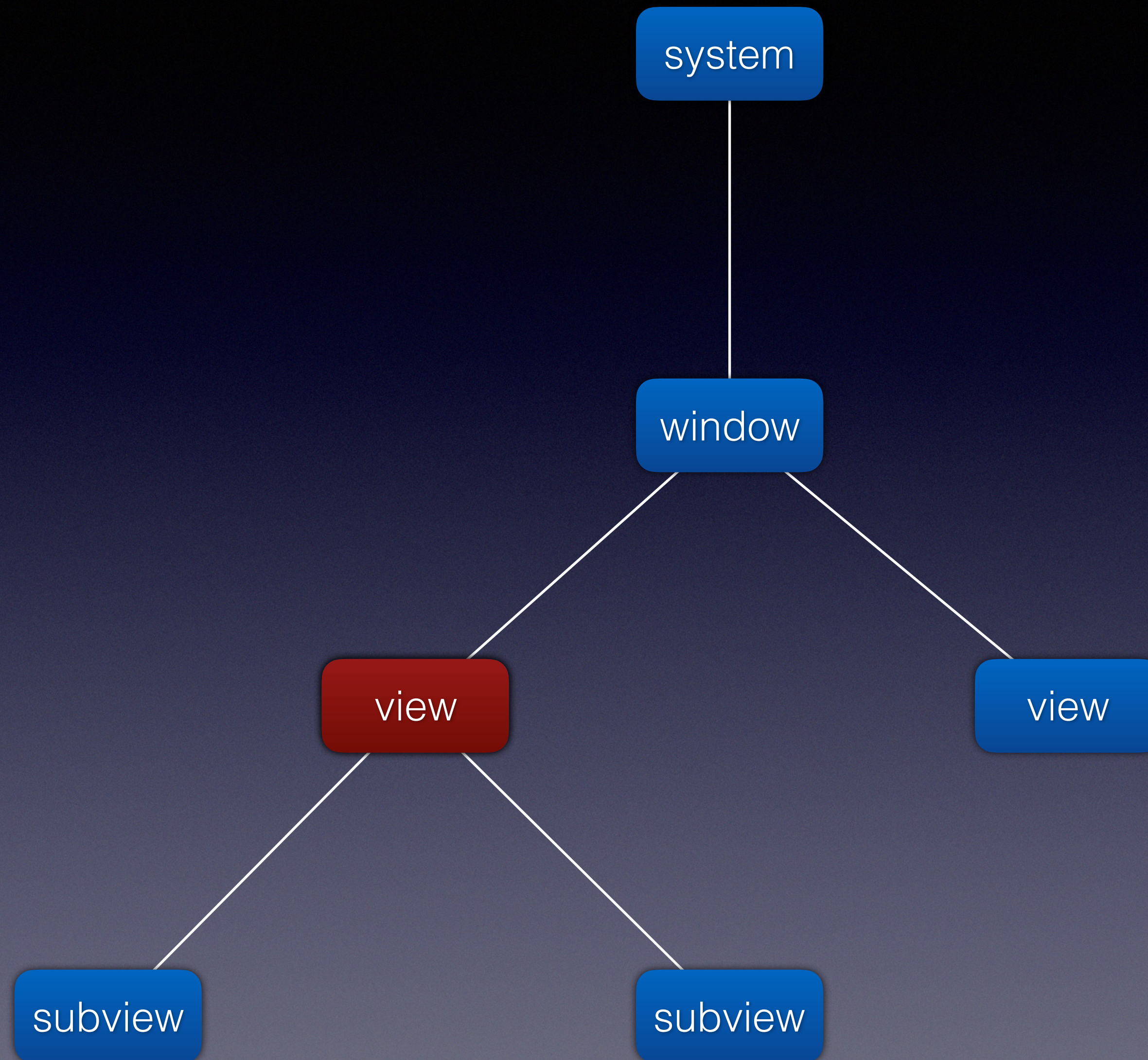
- **appearance**, “The appearance of the receiver, in an **NSAppearance** object.”, **normalerweise nil**
- **effectiveAppearance**, “The appearance that will be used when the receiver is drawn onscreen, in an **NSAppearance** object.”
- Vererbung der Appearance durch die View-Hierarchie.

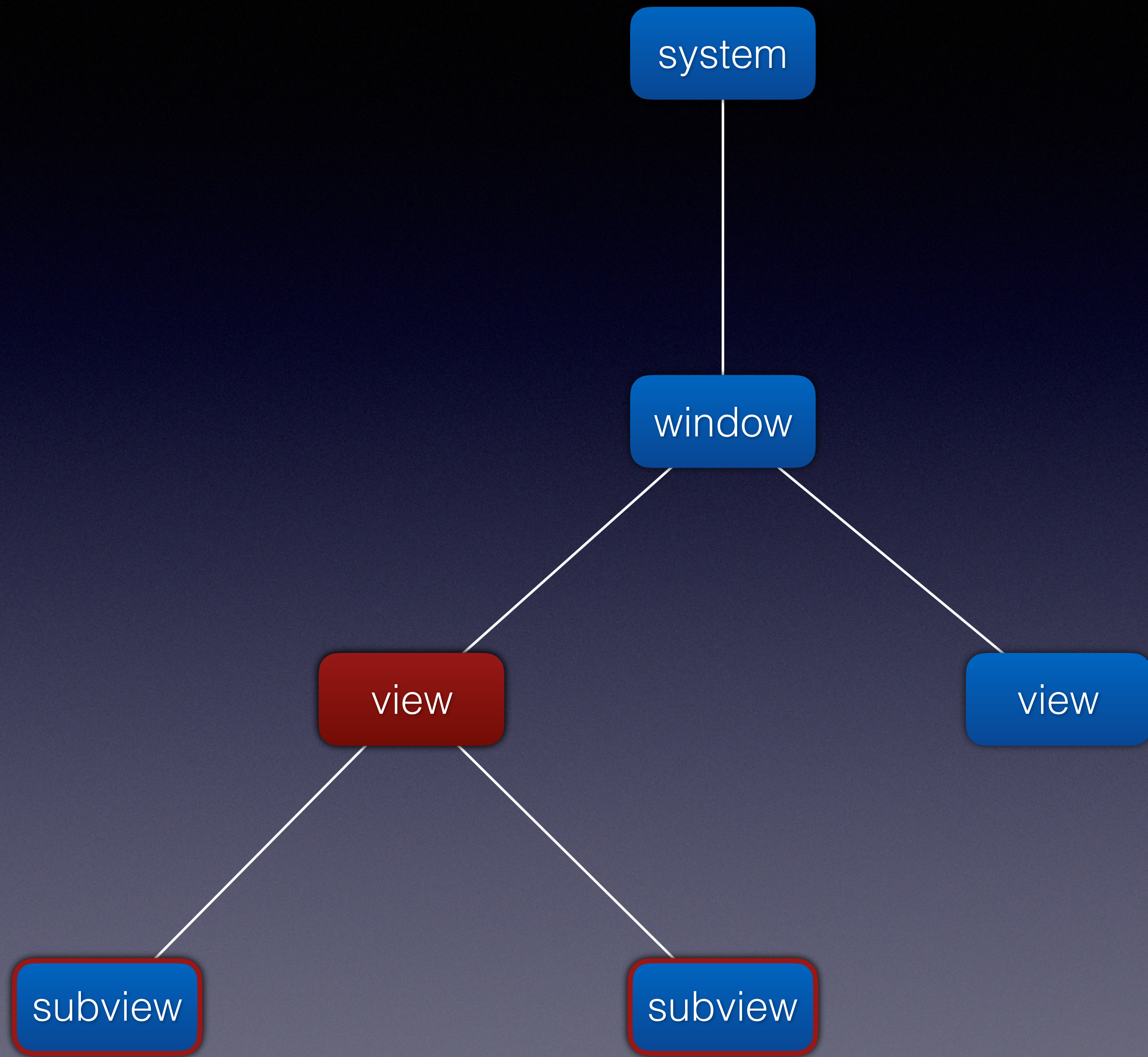












Dark Mode Implementieren

- „(fast) keine Änderungen notwendig“™
 - für Apps die ausschließlich System UI-Komponenten verwenden.
- Bei Custom UI:
 - Semantic Colors
 - Verwendung von Asset Catalogues
 - `+[UIImage imageNamed:]`
 - Template Images
 - `+[NSColor colorNamed:]`

Demo

Weitere Tipps

- Vibrancy und Farben nicht mischen
- **effectiveAppearance** unterstützt KVO
- NSVisualEffectsView als Sibling einfügen
- Mit `-[NSAppearance bestMatchFrom:]` kann man Vibrant/Accessibility Appearances “beschneiden”
- `+[NSAppearance setCurrentAppearance:]` kann verwendet werden, um Appearance beim zeichnen zu überschreiben
- CSS Media Query in Safari Technology Preview

Woher kommen Appearances?

- `+[NSAppearance appearanceNamed:]`
 - `NSAppearanceNameAqua`
 - `NSAppearanceNameDarkAqua`
 - `(NSAppearanceNameVibrantLight)`
 - `(NSAppearanceNameVibrantDark)`
 - `(NSAppearanceNameAccessibilityHighContrast...)`

- `-[NSAppearance initWithAppearanceNamed:bundle:]`
- Appearances als Bundle im Dateisystem
 - `/System/Library/SystemAppearance.bundle/Contents/Resources`
 - 25 (!) verschiedene Appearances
- Können auch selbst erstellt werden:
 - <https://github.com/insidegui/AppearanceMaker>

Demo

Quellen

- WWDC 2018, Session 210 „Introducing Dark Mode“
<https://developer.apple.com/videos/play/wwdc2018/210/>
- WWDC 2018, Session 218 „Advanced Dark Mode“
<https://developer.apple.com/videos/play/wwdc2018/218/>
- Supporting Dark Mode in Your Interface
https://developer.apple.com/documentation/appkit/supporting_dark_mode_in_your_interface?language=objc
- Providing Images for Different Appearances
https://developer.apple.com/documentation/appkit/images_and_pdf/providing_images_for_different_appearances?language=objc

```
tell application "System Events"  
    tell appearance preferences  
        set dark mode to not dark mode  
    end tell  
end tell
```