

PG5600

iOS programmering

Forelesning 8

Sist gang

- Debugging
- Swift og gjenbruk av kode
 - Bundles
 - Rammeverk
- Tråder og asynkronitet
- Snakke med internett
- Cloud Kit

Agenda

App extensions

Typer app extensions

- Today
- Share
- Action
- Photo Editing
- Document Provider / Storage Provider
- Custom Keyboard

Today

I dag

Ferdig

– Oversikt for i dag

–  Påminnelser

–  Kalender

–  Kindle

– Oversikt for i morgen

IKKE INKLUDER

+  Trafikkinformasjon

+  Aksjer

+  Dine profilbesøk

+  Dropbox

+  Duolingo

+  Evernote

I dag

Varslinger



Kalender

Ingen hendelser



Kindle



The Phoenix Project: A Novel About IT, Dev...

Kim, Gene; Behr, Kevin ; Spafford, George



iOS Programming: The Big Nerd Ranch Gui...

Conway, Joe; Hillegass, Aaron; Keur, Christian



I morgen

Formiddagen ser ut til å være åpen. Du har
én oppføring i kalenderen kl. 15.00.

Rediger

Værinformasjon levert av
The Weather Channel, LLC.

open.bekk.no



BEKK OPEN

Logging for distributed systems

Keeping your system debuggable and traceable in the face of scaling, growth and a distributed architecture.



AirDrop. Del med personer i nærheten. Hvis du ikke ser dem, må du be dem slå på AirDrop i kontrollsenteret på iOS eller gå til AirDrop i Finder på en Mac.



Mail



Twitter



Facebook



Pocket



Mer

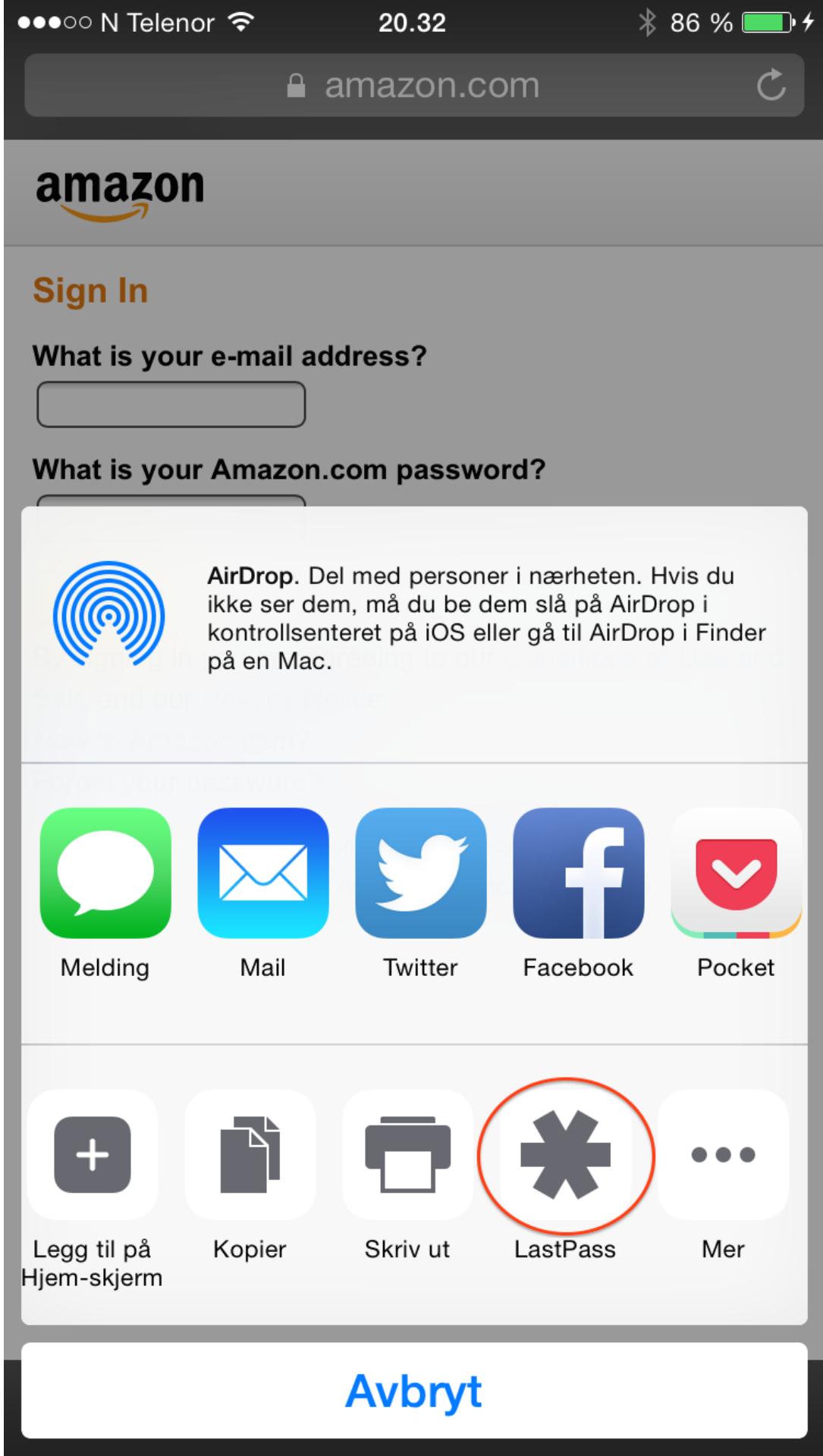
Legg til
bokmerkeLegg til i
leselistenLegg til på
Hjem-skjermLegg til i delte
koblinger

Kopier

Share

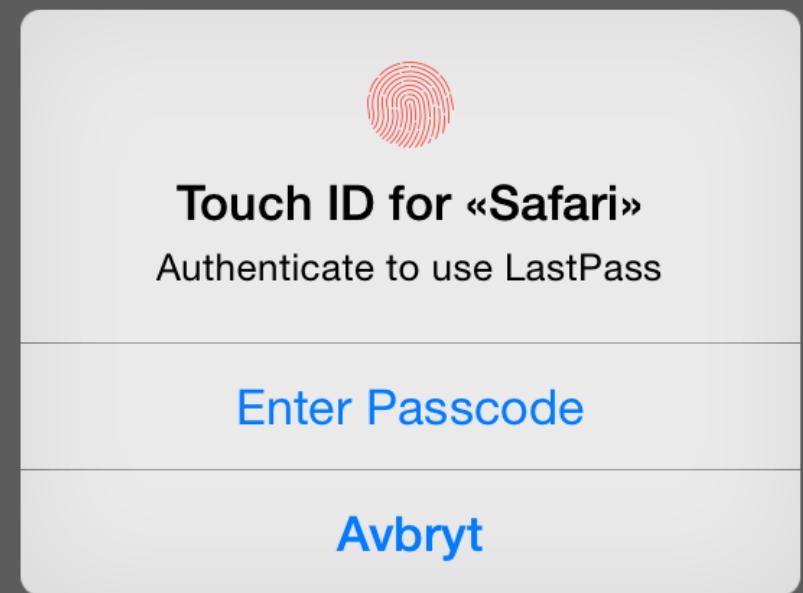
Avbryt

Action



Action

●●●○○ N Telenor ⌂ 19.38 * 40 % 🔋



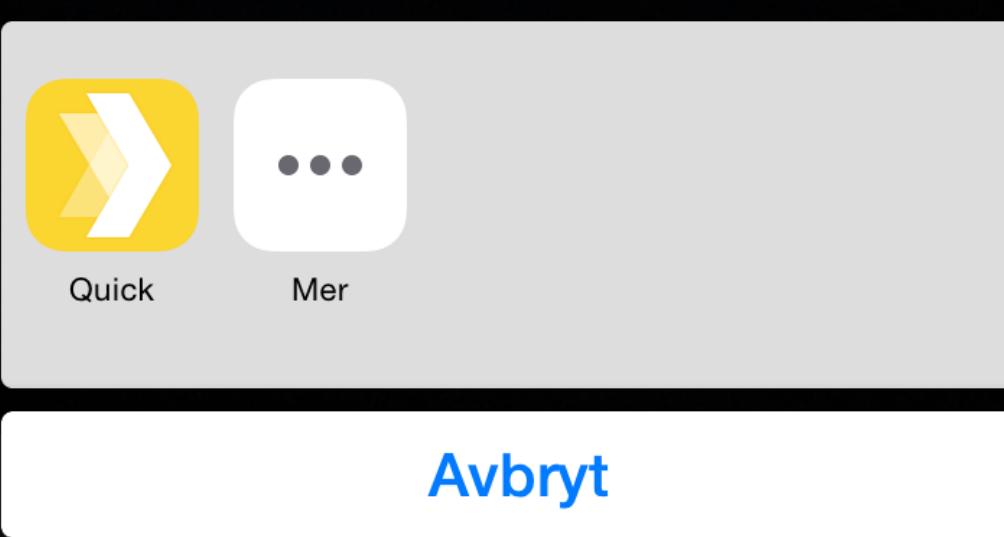
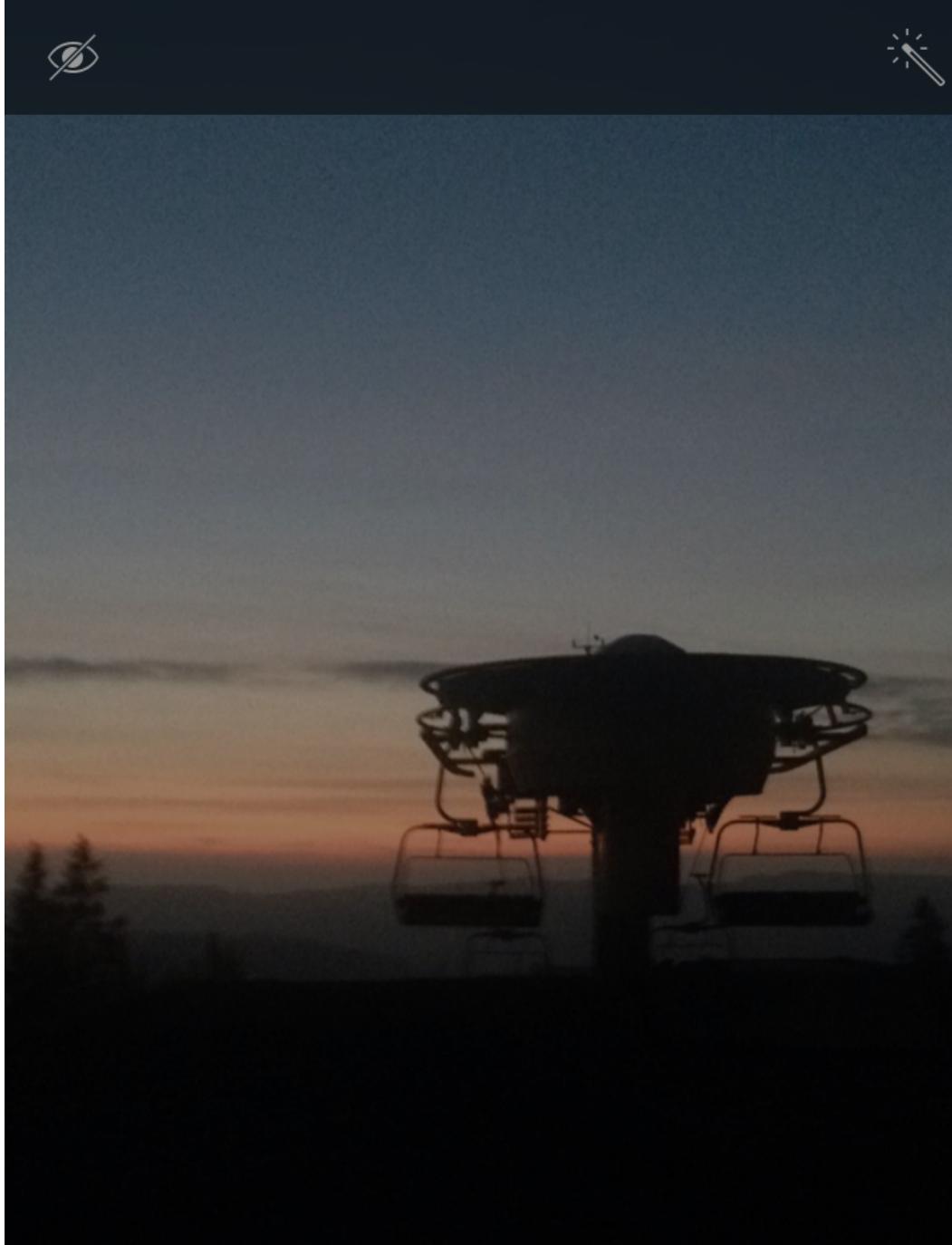
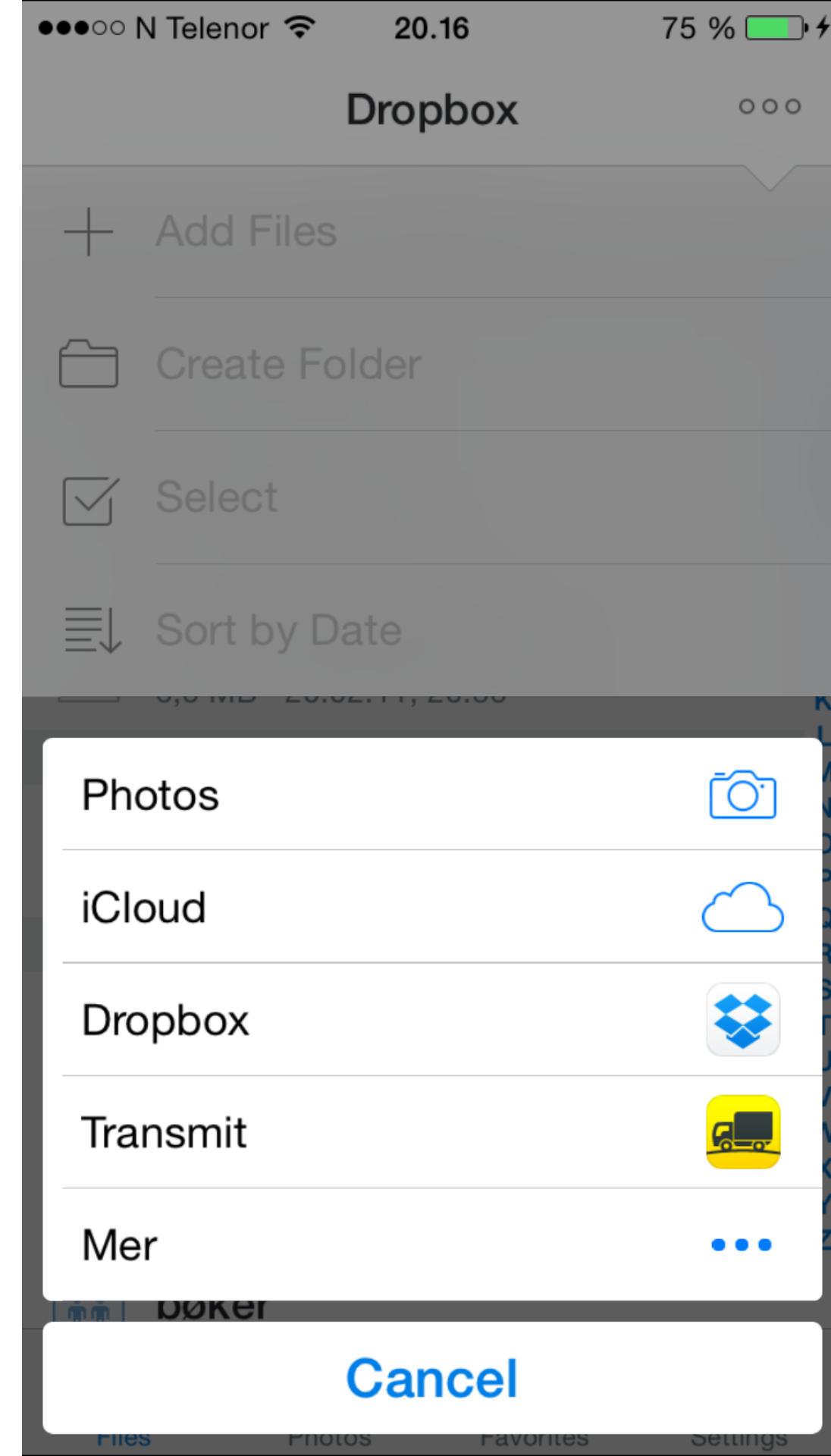


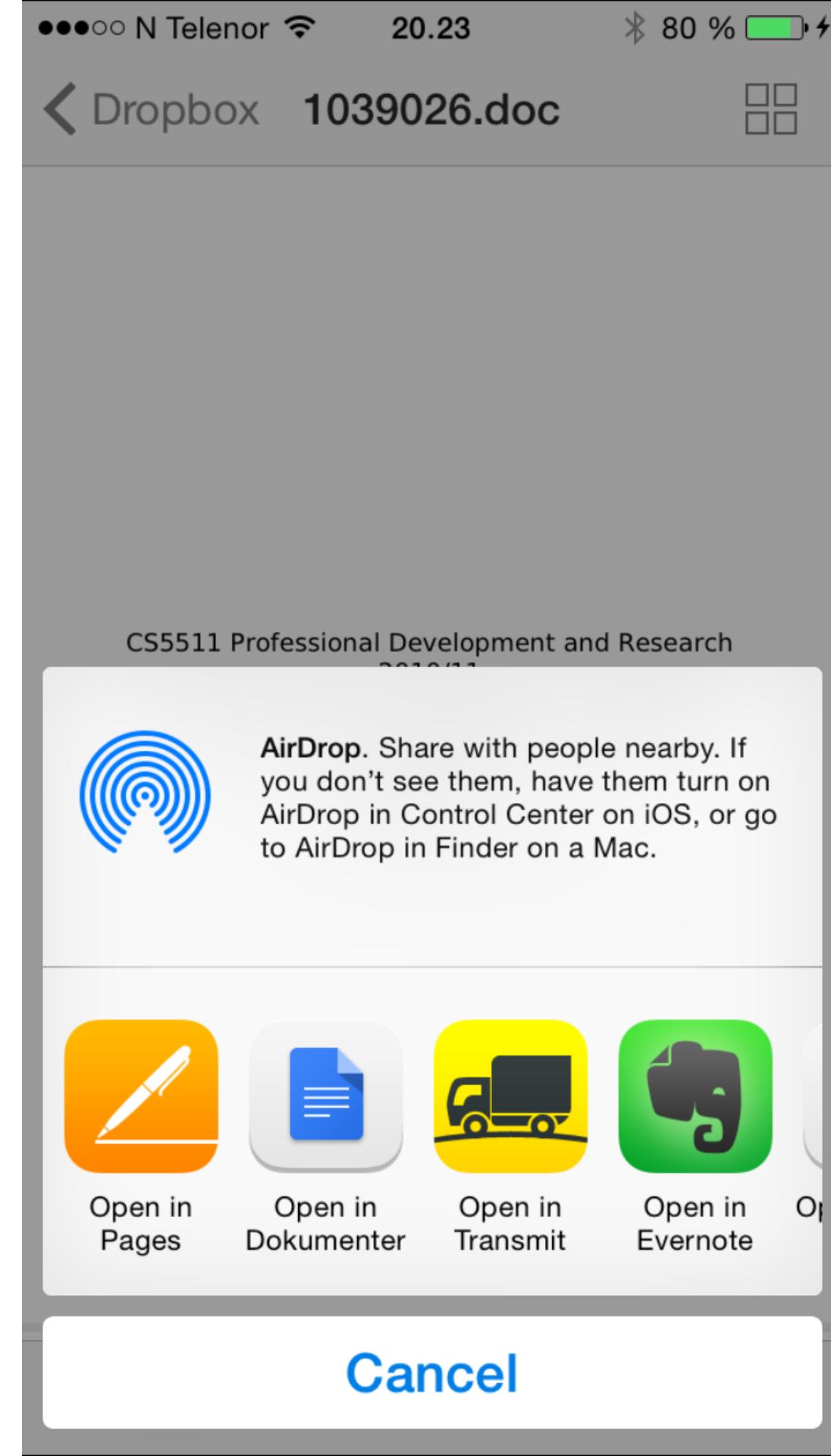
Photo editing

Document picker

(Document provider)



Storage Provider (Document provider)



< Extensions



GIF Keyboard [12+]

Riffsy, Inc >

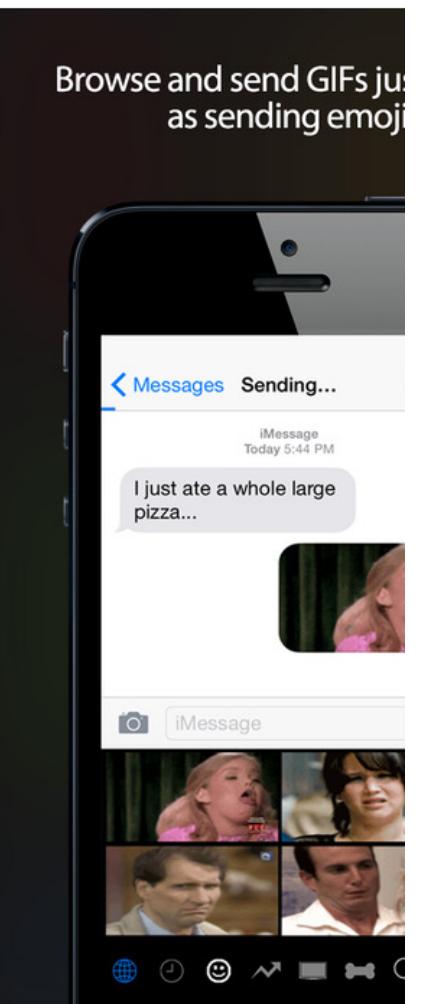
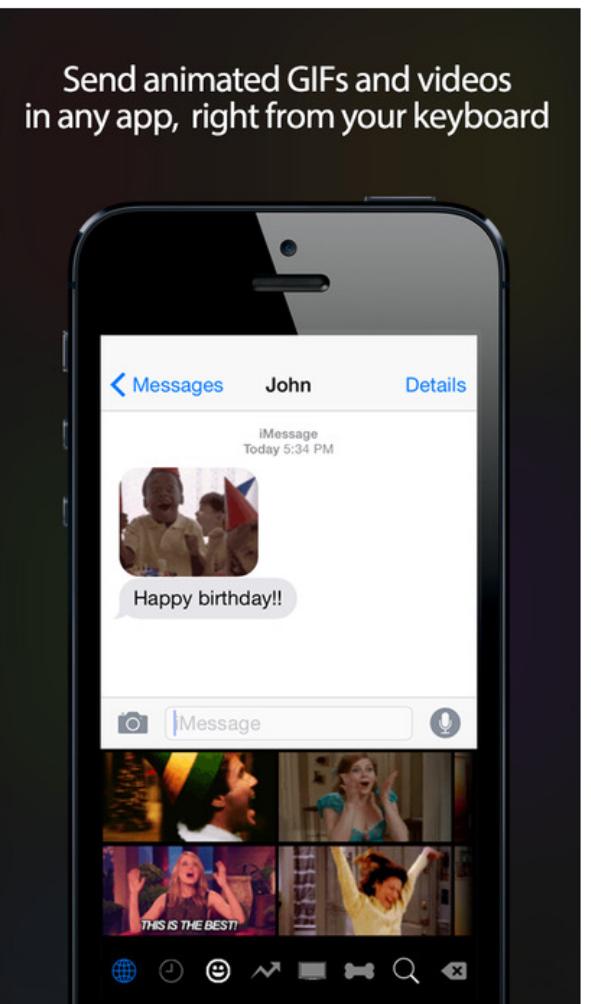
★★★★★ (5)

+ GRATIS

Detaljer

Omtaler

Relatert



Custom keyboards



Aktuelt



Topplister



Utforsk



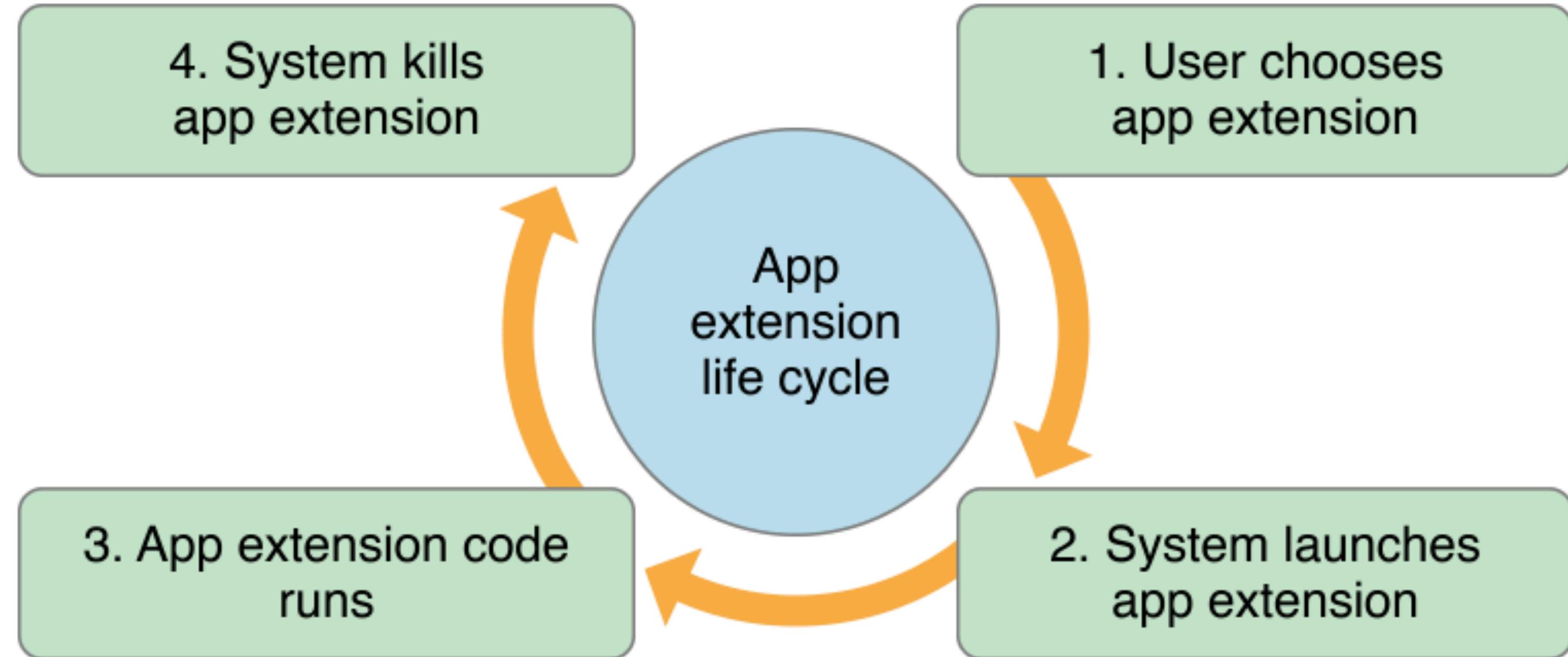
Søk



Oppdateringer

App extensions

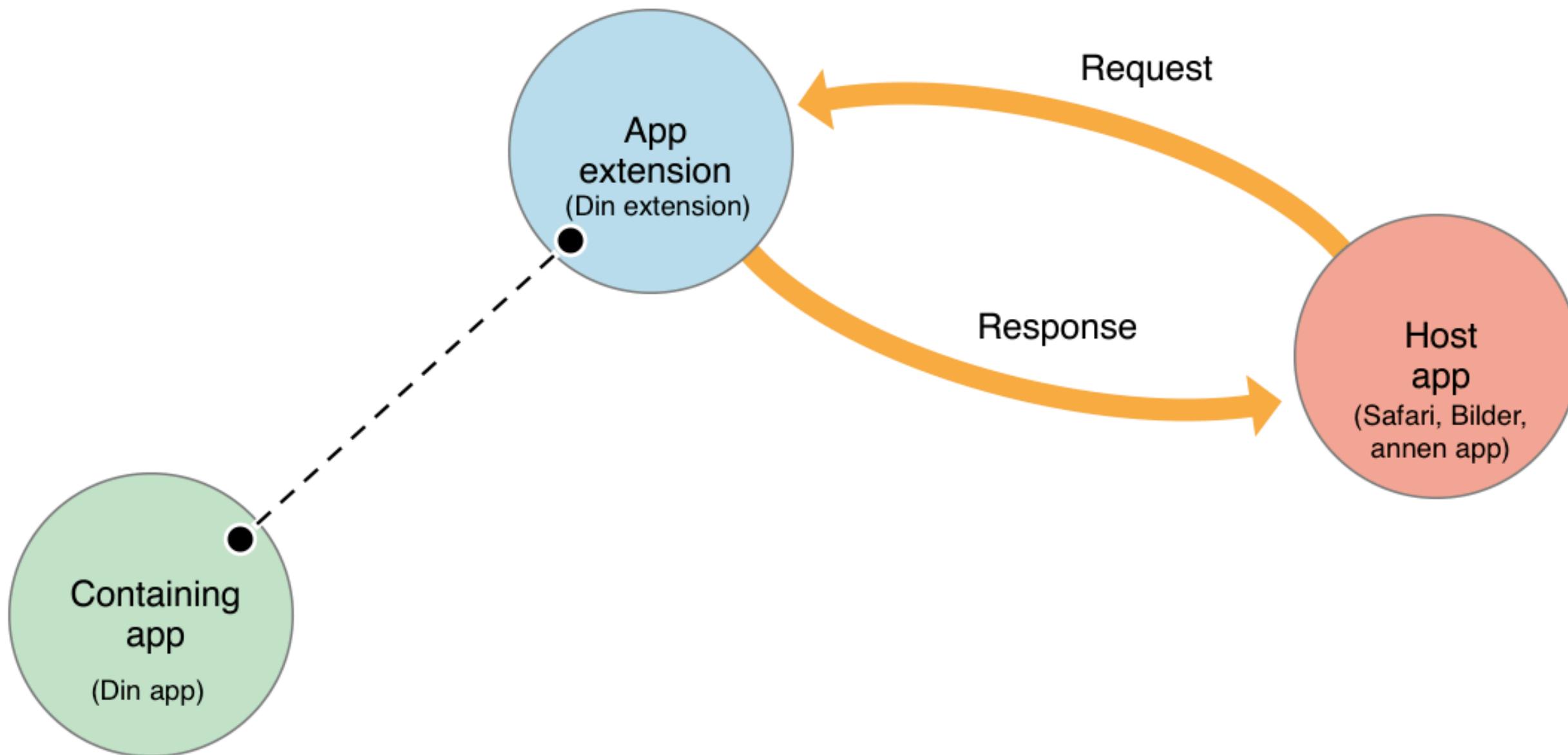
- Extensions må være del av en app
- Extensions lages som et nytt target for en eksisterende app
- Extensions blir installert parallelt med appen
- Brukerne må aktivt ta i bruk extensions
- Extensions har noen begrensninger i hvilke APIer som kan brukes
- Containing app og extensions kjøres separat og med egne livssykluser



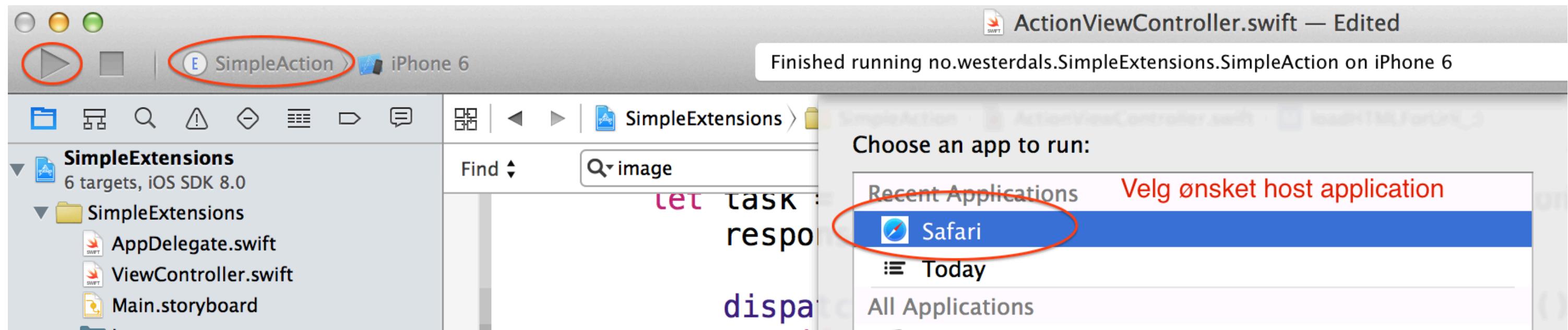
"To pass app review, your containing app must provide functionality to users; it can't just contain app extensions."

- App Extension Programming Guide

Terminologi



Kjør og debug



Today

Eksempel

Eksempel: Today

```
class TodayViewController: UIViewController, NCWidgetProviding {  
  
    @IBOutlet weak var label: UILabel!  
  
    func widgetPerformUpdateWithCompletionHandler(completionHandler: ((NCUpdateResult) -> Void)!) {  
        // Perform any setup necessary in order to update the view.  
  
        // If an error is encountered, use NCUpdateResult.Failed  
        // If there's no update required, use NCUpdateResult.NoData  
        // If there's an update, use NCUpdateResult.NewData  
  
        self.label.text = NSDate().description  
        completionHandler(NCUpdateResult.NewData)  
    }  
}
```

extensionContext

extensionContext

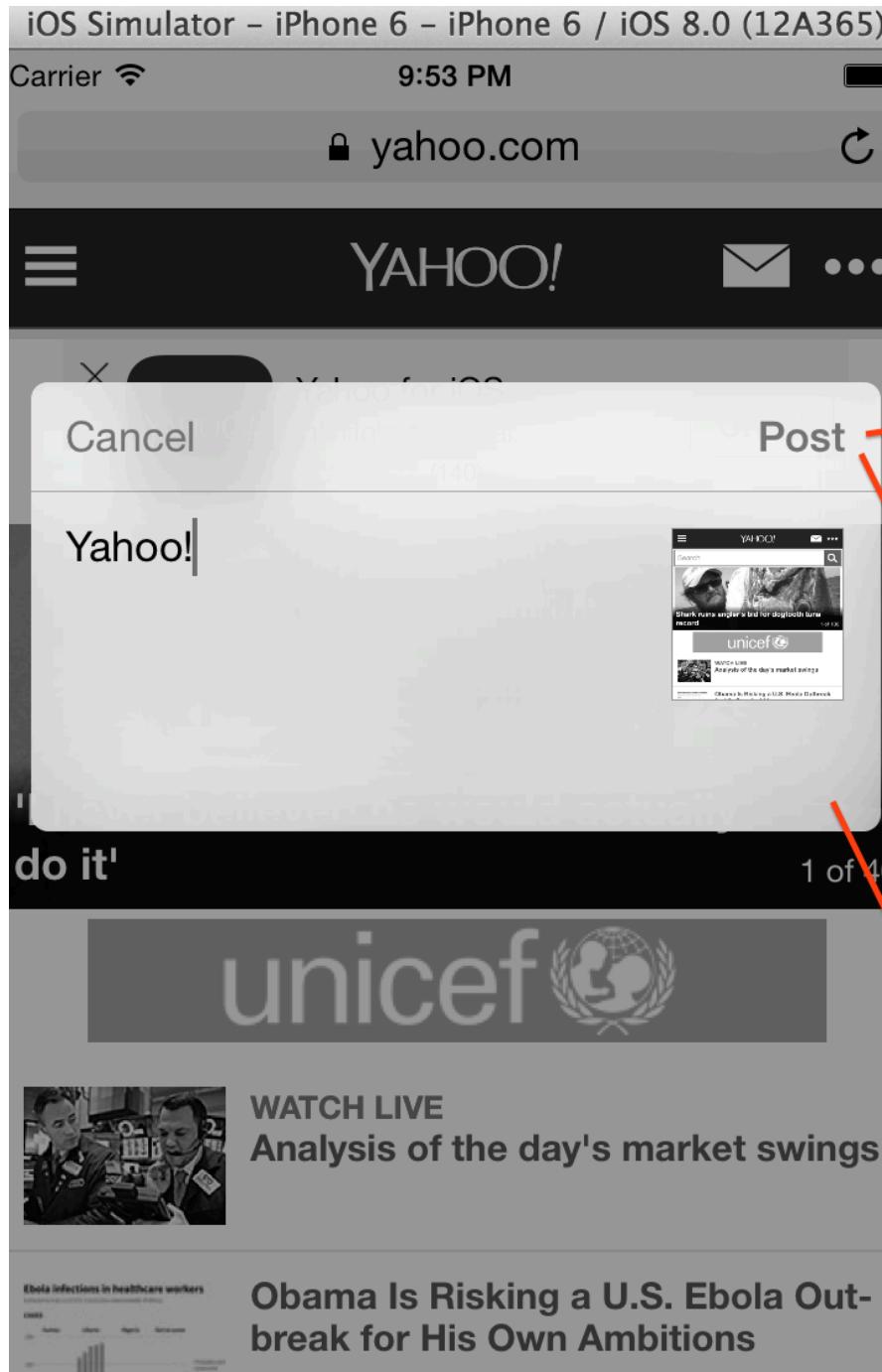
- Tilgjengelig i din app med context fra host, eks:
 - Hvilken side står man på i Safari når man trykket på action?
 - Hvilke bilde(r) er valgt når du trykker del?

share

Eksempel

Share extensions

- Presenterer en dialog hvor brukere kan dele data (redigere og poste)
- Som standard arver man fra `SLComposeServiceViewController` som gir en del gratis: preview, validering, animasjon, konfigurasjon, gui
- Kan gå helt custom med egen `ViewController`



```
Running no.westerdai.simpleExtensions.SimpleShare on iPhone 6
SimpleExtensions > SimpleShare > ShareViewController.swift > didSelectPost()

import UIKit
import Social

class ShareViewController: SLComposeServiceViewController {

    override func isContentValid() -> Bool {
        // Do validation of contentText and/or NSExtensionContext attachments
        here
        return countElements(self.contentText) > 5
    }

    override func didSelectPost() {
        println(self.contentText)

        // Inform the host that we're done, so it un-blocks its UI
        self.extensionContext!.completeRequestReturningItems([], completionHandler: nil)
    }

    override func configurationItems() -> [AnyObject]! {
        return NSArray()
    }
}
```

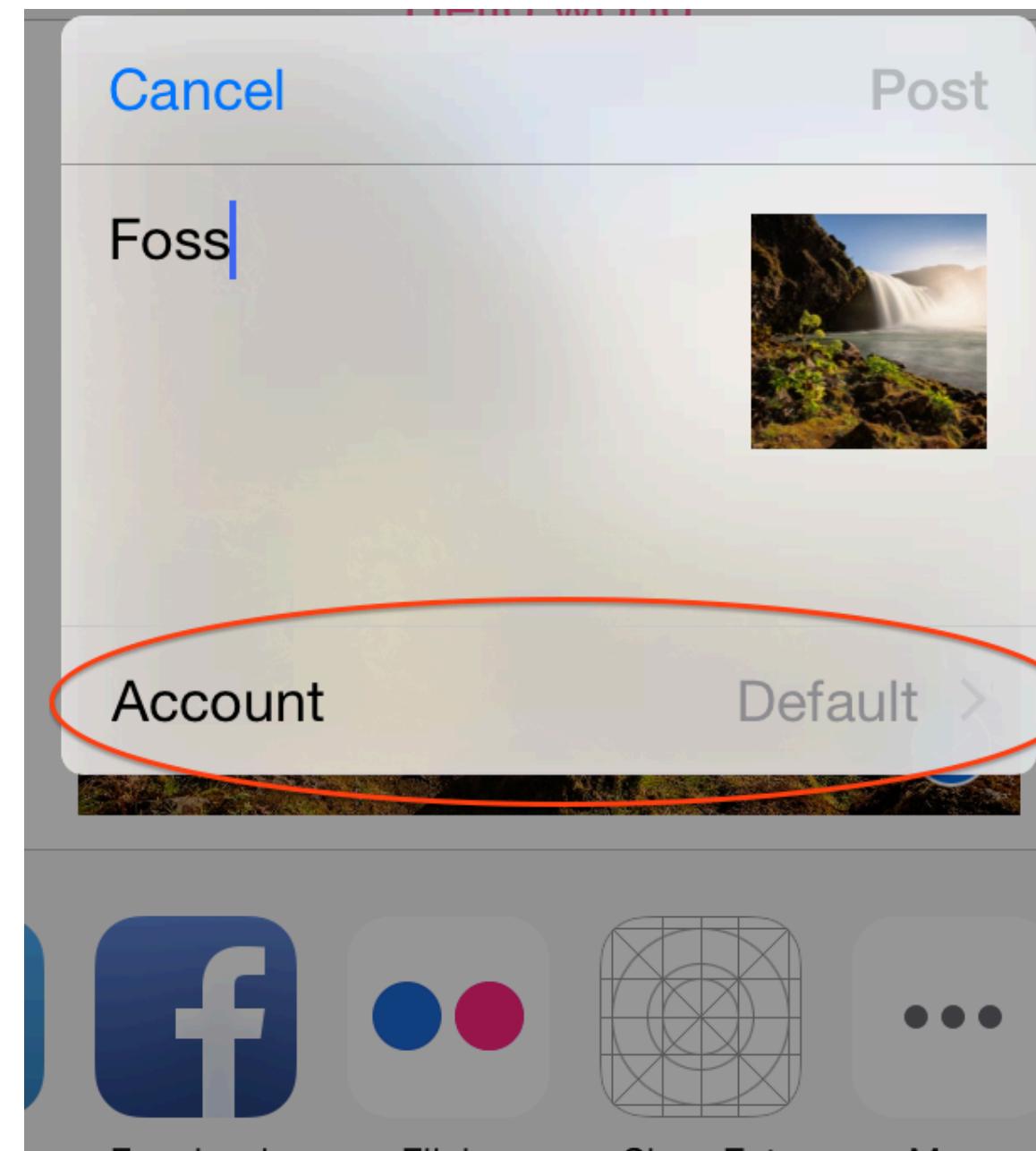
Eksempel konfigurering av options

```
override func configurationItems() -> [AnyObject]! {
    // To add configuration options via table cells at the bottom of the sheet, return an array of SLComposeSheetConfigurationItem here.

    let vc = self.storyboard?.instantiateViewController(withIdentifier("configController")) as UIViewController

    let config = SLComposeSheetConfigurationItem()
    config.title = "Account"
    config.value = "Default"
    config.tapHandler = {
        self.pushConfigurationViewController(vc)
    }
    return [config]
}
```

Eksempel konfigurerering av options



Action

Eksempel

Eksempel: Vis HTML action

```
class ActionViewController: UIViewController {

    @IBOutlet weak var textView: UITextView!

    override func viewDidLoad() {
        super.viewDidLoad()

        // Get the item[s] we're handling from the extension context.
        // TODO
    }

    func loadHTMLForUrl(url : NSURL) {
        // TODO
    }

    @IBAction func done() {
        self.extensionContext!.completeRequestReturningItems([], completionHandler: nil)
    }

}
```

```
// For alle extensionItems...
for item: AnyObject in self.extensionContext!.inputItems {
    let inputItem = item as NSExtensionItem

    // Se på alle itemProviders...
    for provider: AnyObject in inputItem.attachments! {
        let itemProvider = provider as NSItemProvider

        // Se om de er en URL
        if itemProvider.hasItemConformingToTypeIdentifier(String(kUTTypeURL)) {

            itemProvider.loadItemForTypeIdentifier(String(kUTTypeURL),
                options: nil, completionHandler: { (item, error) -> Void in

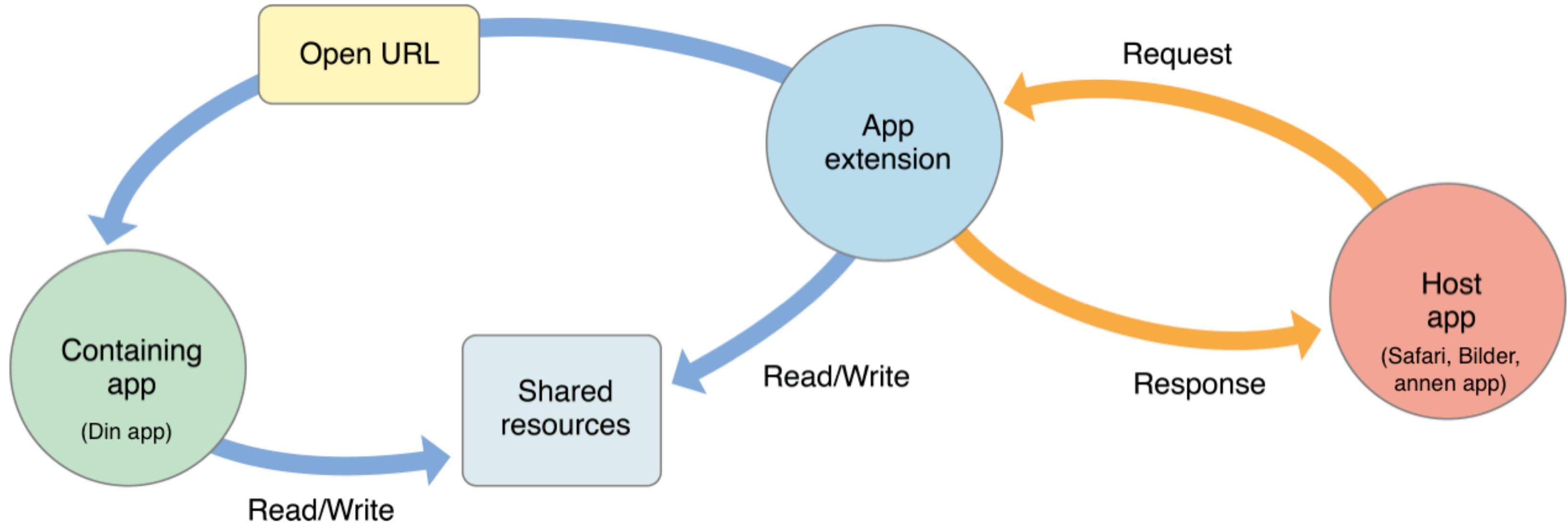
                if let url = item as? NSURL {
                    self.loadHTMLForUrl(url)
                }
            })
        }
    }
}
```

```
func loadHTMLForUrl(url: NSURL) {
    let session = NSURLSession.sharedSession()
    let task = session.dataTaskWithURL(url,
        completionHandler: { (data, response, error) -> Void in

            dispatch_async(dispatch_get_main_queue(), { () -> Void in
                if data != nil {
                    self.textView.text = NSString(data: data, encoding: NSUTF8StringEncoding)
                } else {
                    self.textView.text = error.description
                }
            })
        }
    task.resume()
}
```

Mer avanserte ting

App og extension kan ikke snakke
direkte med hverandre



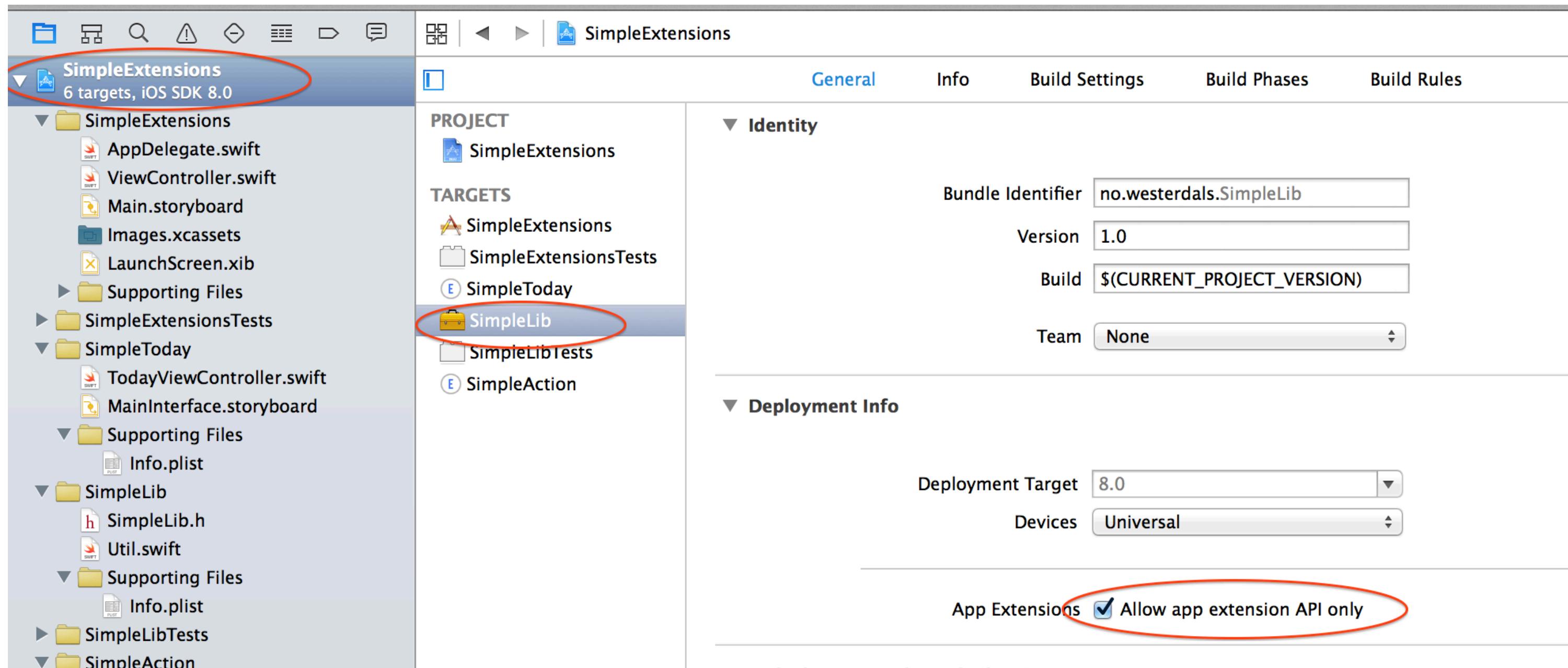
App og extension kan ikke snakke direkte med hverandre

- Shared resources
 - Må aktivere App groups og entitlements
 - Krever til Developer portal
 - Søk etter "Sharing Data with Your Containing App" i dok for detaljer
- Kodedeling: med Cocoa Touch Framework

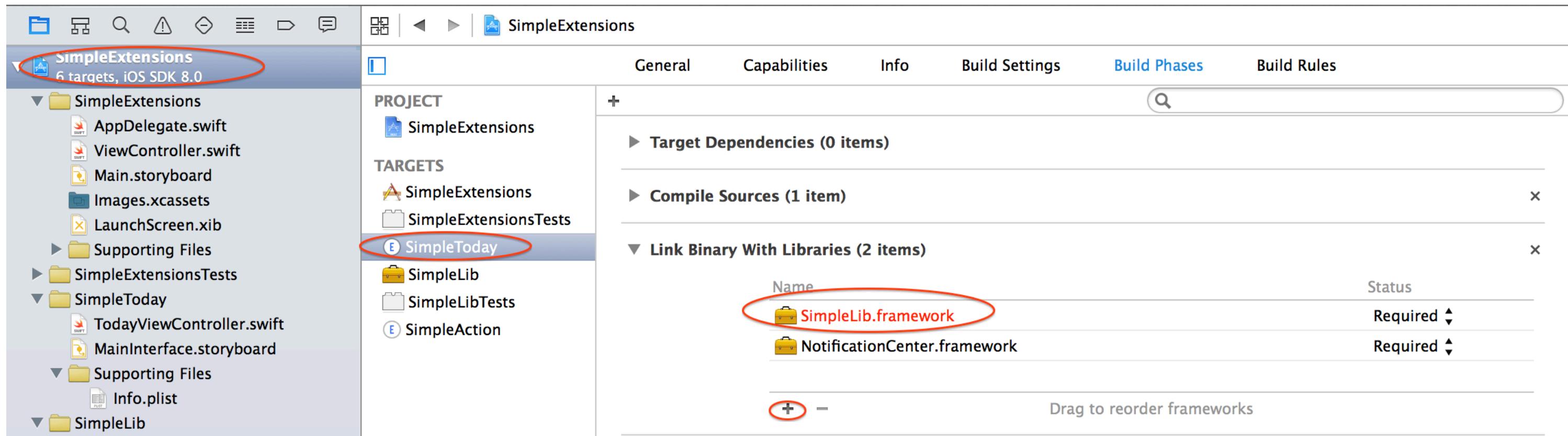
Dele kode mellom app og extension

1. File -> New -> Target -> Cocoa Touch Framework
2. Gi navn, eks SimpleLib
3. Opprett klasser i SimpleLib. Husk å sett de til public
4. Huk av "Allow app extension API only" (se neste foil)
5. Inkluder framework i extension (se to foiler frem)
6. I extension/app target: import SimpleLib og du kan bruke klassene i framework som andre klasser

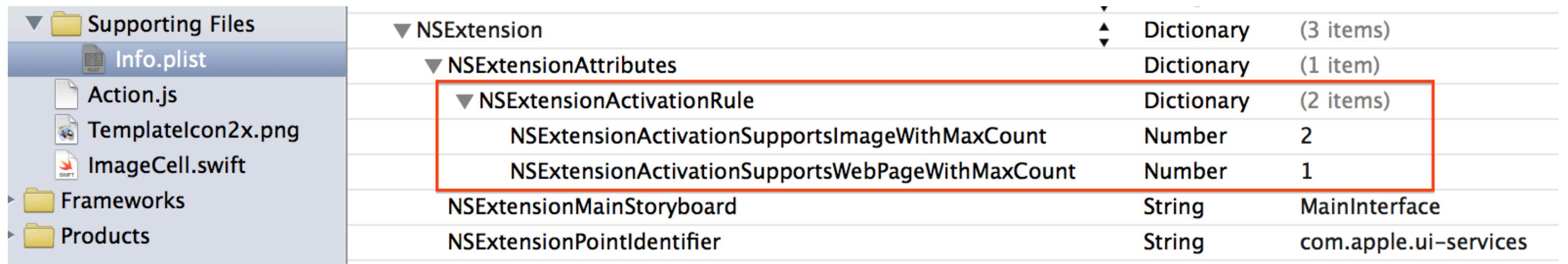
Dele kode mellom app og extension



Dele kode mellom app og extension

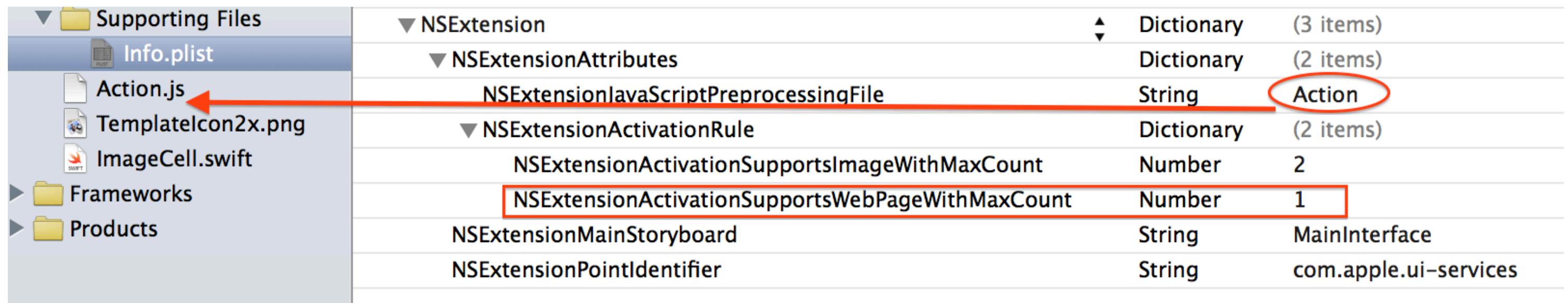


Definere hvilke typer extension skal aktiveres for



Få tilgang til nettsider
i extensions med JavaScript

Legg til js-fil og konfigurer Info.plist



Action.js

```
var MyExtension = function() {};

MyExtension.prototype = {
    run: function(arguments) { // kalles ved start
        arguments.completionFunction({
            "selection": document.getSelection().toString()
        });
    },
    finalize: function(arguments) { // kalles ved slutt
        var range = document.getSelection().getRangeAt(0)
        range.deleteContents()
        // arguments kommer fra NSExtensionContext completeRequestReturningItems:completion:
        range.insertNode(document.createTextNode(arguments["translatedText"]))
    }
};

// Må assinges til "ExtensionPreprocessingJS"
var ExtensionPreprocessingJS = new MyExtension;
```

Hent ut data sendt fra run



The screenshot shows a news article from VG (Vogts) dated Wednesday, October 15. The headline reads "Rognlien gir seg som idrettspresident". Below the headline, a quote from Truls Dæhli is shown: "VGs Truls Dæhli: - Norsk idrett trenger en som er i stand til å tenke nytt". A red arrow points from this quote to the word "selection" in the accompanying code.

```
for item: AnyObject in self.extensionContext!.inputItems {
    let inputItem = item as NSExtensionItem
    for provider: AnyObject in inputItem.attachments! {
        let itemProvider = provider as NSItemProvider
        if itemProvider.hasItemConformingToTypeIdentifier(String(kUTTypePropertyList)) {

            itemProvider.loadItemForTypeIdentifier(String(kUTTypePropertyList), options: nil,
                completionHandler: { (item, error) -> Void in
                    if(item != nil) {
                        let dictionary = item as NSDictionary
                        let data = dictionary[NSExtensionJavaScriptPreprocessingResultsKey] as
                            [NSObject: AnyObject]
                        println(data["selection"]!)
                    } else {
                        println("Error \(error), item \(item)")
                    }
                }
            }
        }
    }
}
```

The code is part of a JavaScript action named "no.westerdals.SimpleExtensions.JSAction". A red arrow also points from the "selection" line to the output window below, which displays the value "null".

```
Failed to inherit CoreMedia permissions from 4927
(null)
Truls Dæhli
```

Returner data til finalize

```
let resultsDictionary = [
    NSExtensionJavaScriptFinalizeArgumentKey: [
        "translatedText": "Tekst fra oversetting her"
    ]
]

let resultsItem = NSExtensionItem()
let attachment = NSItemProvider(item: resultsDictionary,
    typeIdentifier: String(kUTTypePropertyList))
resultsItem.attachments = [attachment]

self.extensionContext!.completeRequestReturningItems([resultsItem],
    completionHandler: nil)
```

oppgaver

Se Øvingsoppgavene

<https://github.com/hinderberg/ios-swift-kurs>