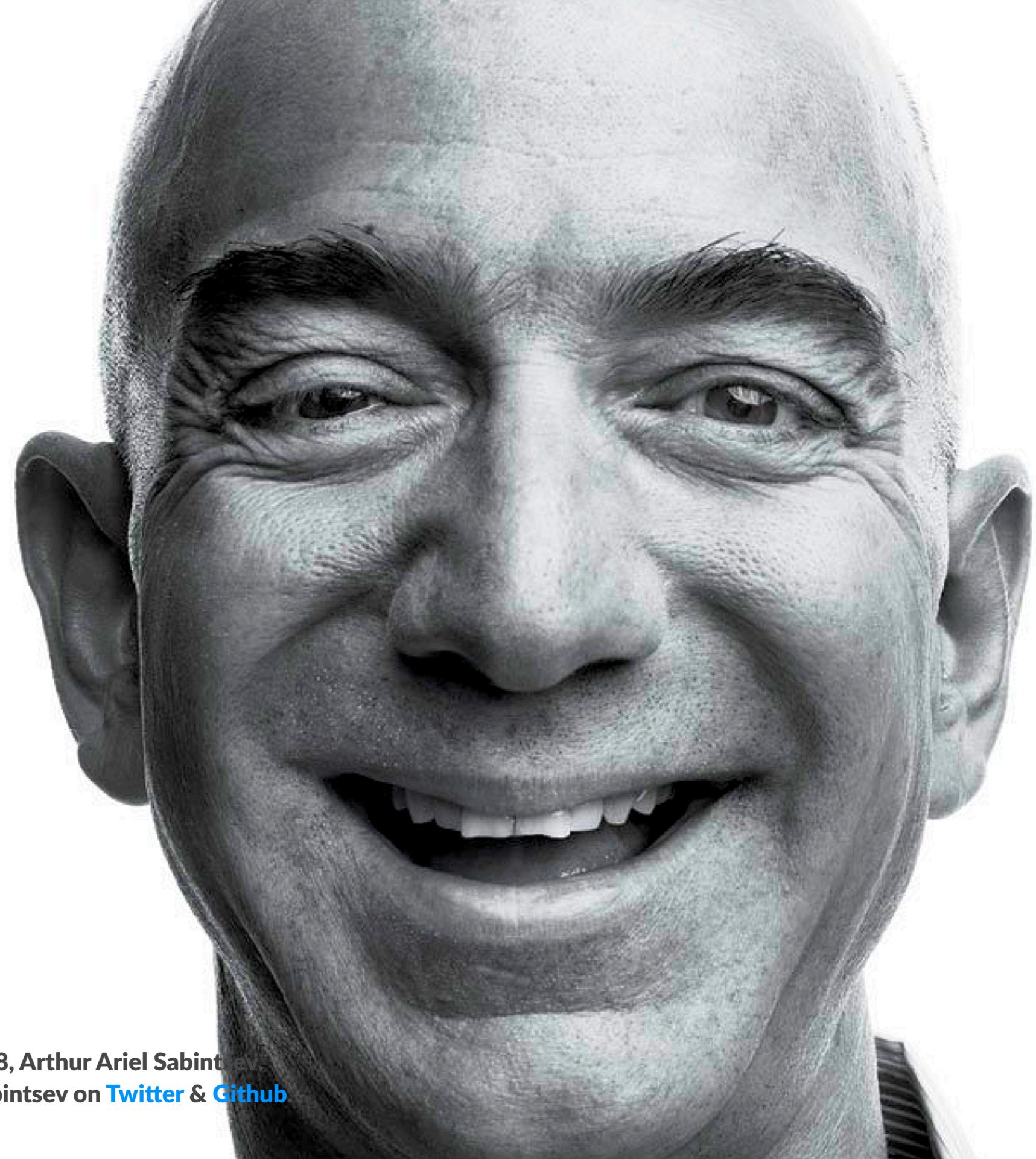


Scaling iOS Architecture

Arthur Sabintsev
Lead iOS Developer
The Washington Post



Jeff Bezos

CEO, Amazon
Founder, Blue Origin
Owner, The Washington Post

Apps (pre-Bezos Acquisition)



Apps (post-Bezos Acquisition)





The Washington Post

arc publishing



DAILY NEWS

Chicago Tribune

Los Angeles Times

The
Boston
Globe

© 2018, Arthur Ariel Sabintsev.

ArtSabintsev on [Twitter](#) & [Github](#).

THE GLOBE AND MAIL



The Salt Lake Tribune

ADVANCE NZME.

NEW ZEALAND
MEDIA AND
ENTERTAINMENT

LOCAL

Le Parisien



THE BALTIMORE SUN
baltimoresun.com

infobae

The
Philadelphia
Inquirer

*The Washington Post is now building
and maintaining native mobile apps
for other newspaper publishers.*

Project Unicorn ^{1 2}

(aka: *The White Label App, Arc Mobile, Arc Native*)



¹ Image Credit: [Unicorn King](#) by [rubymight](#)

² Unofficial (Internal) Logo

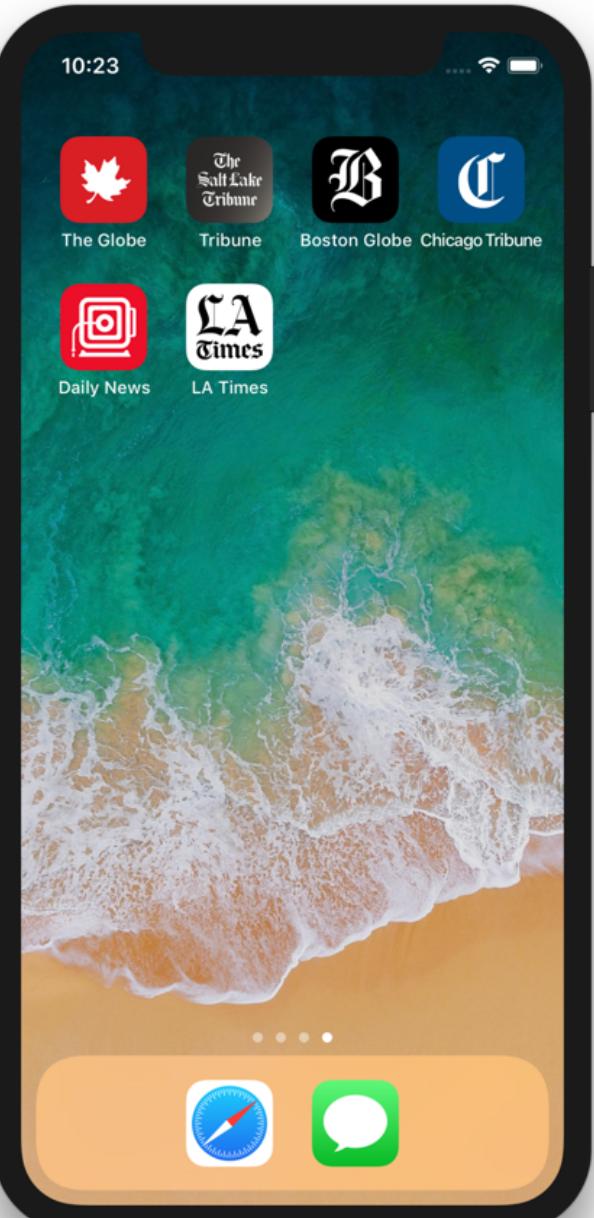
What is the Unicorn / Arc Mobile app?

Unicorn was pitched to be a clone of The Washington Post Classic app.



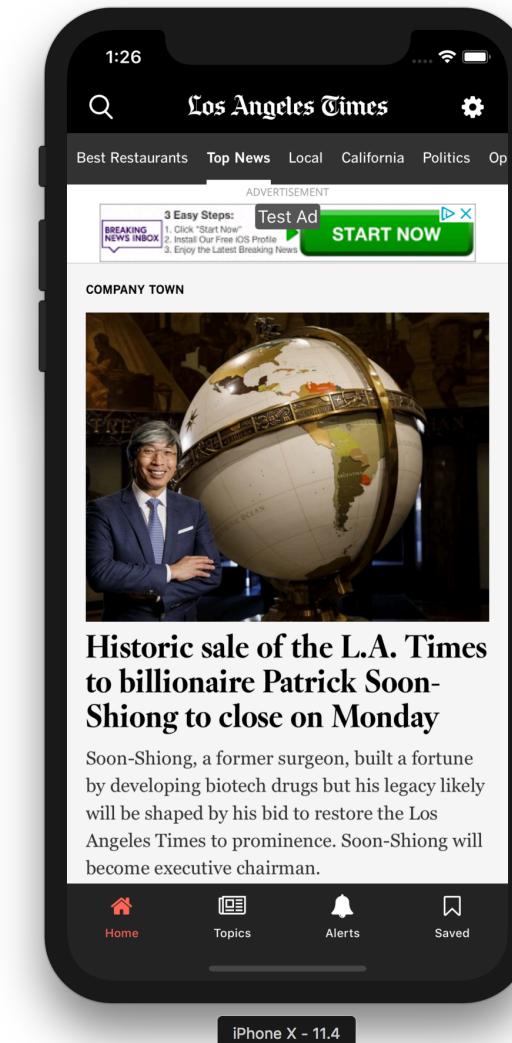
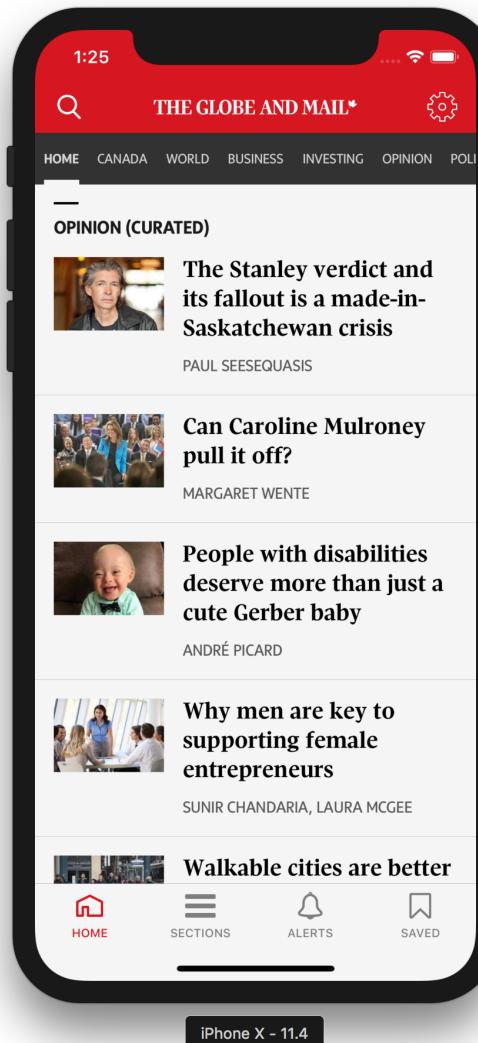
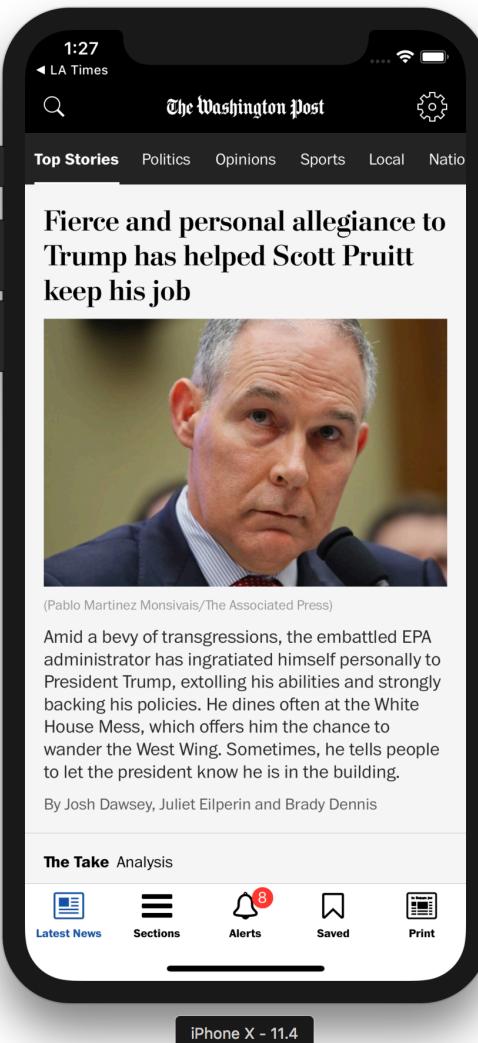


Unicorn
→
Powered



Chicago Tribune: June 2018
Boston Globe: August 2018

Unicorn-Powered Apps



We were not prepared!

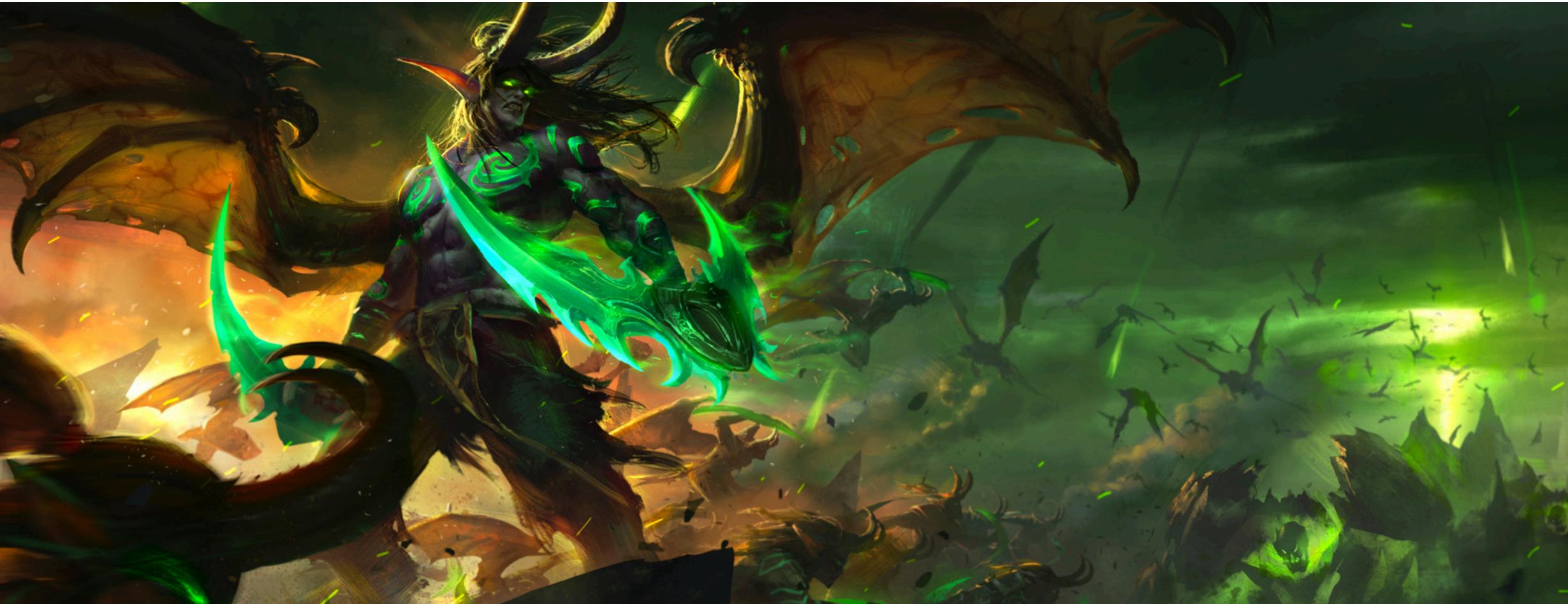


Image: [Qichao Wang on ArtStation](#)

Structure of a News App

- Section Fronts
- Articles
- Section List
- Search
- Saved Articles
- Segmented Push Notifications



POLITICS

ANALYSIS



**Trump, Kim, Erdogan, Putin:
When strongmen stick together,
democracy should watch out 🔑**

In the past decade, might-makes-right nationalism has made a comeback like the world has never seen since the 1930s. For now, some of the strongmen are praising and supporting one another – but what



HOME



SECTIONS



ALERTS



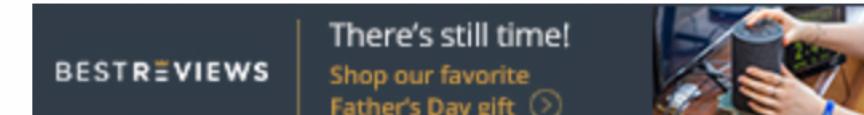
SAVED

Structure of a News App

- Section Fronts
- Articles
- Section List
- Search
- Saved Articles
- Segmented Push Notifications



ADVERTISEMENT



A mother's story of a stranger telling her to "speak English" to her daughter to avoid confusing her.

LOS ANGELES TIMES

I am raising my daughter to speak three languages. A stranger demanded I 'speak English' to her

Structure of a News App

- Section Fronts
- Articles
- Section List
- Search
- Saved Articles
- Segmented Push Notifications

Sprint 11:16 PM 100%

The Salt Lake Tribune 

RECENT

Business

Global Religion

Education

Utah Jazz

SECTIONS

Homepage

News

Business

Education

Environment



Sports



HOME



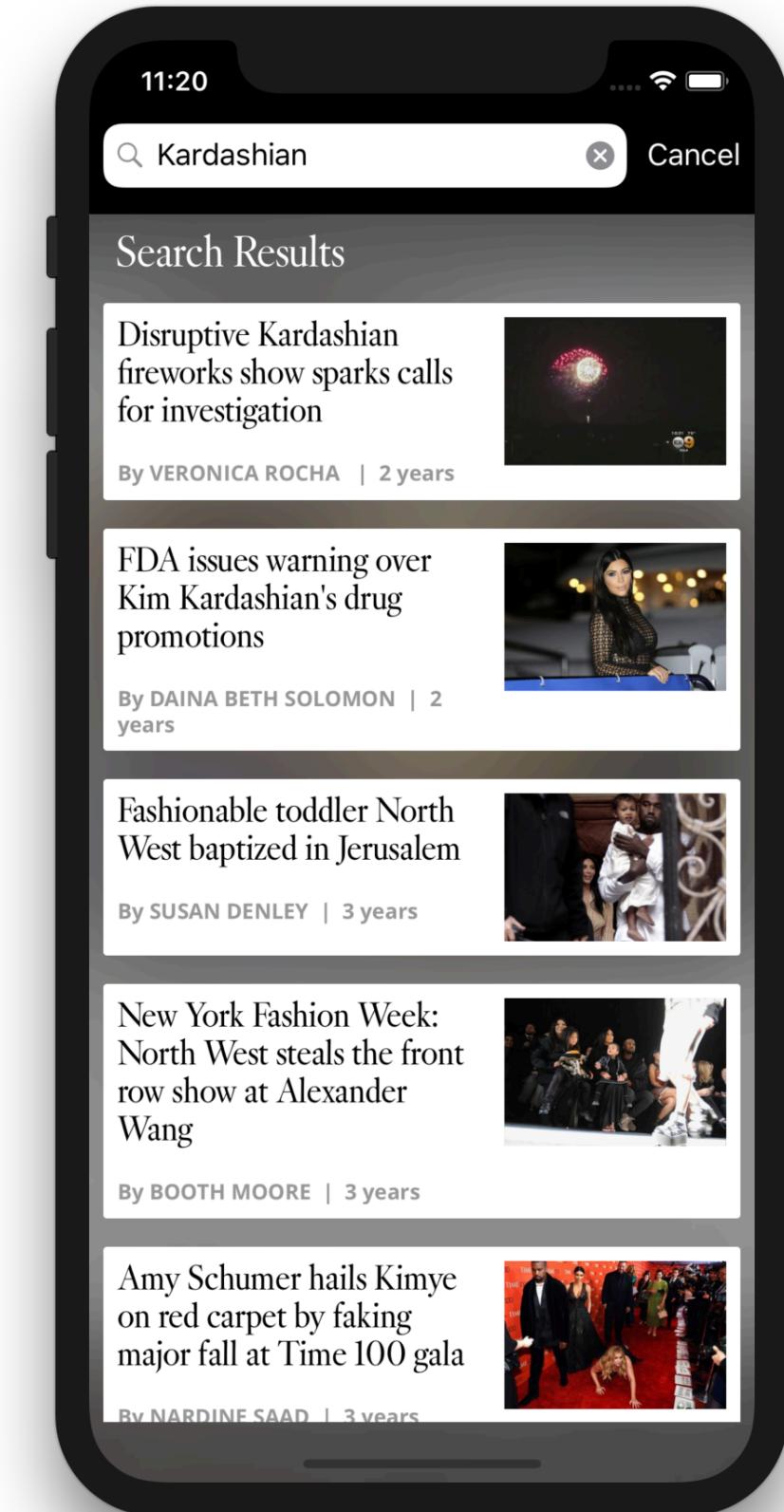
SECTIONS



SAVED

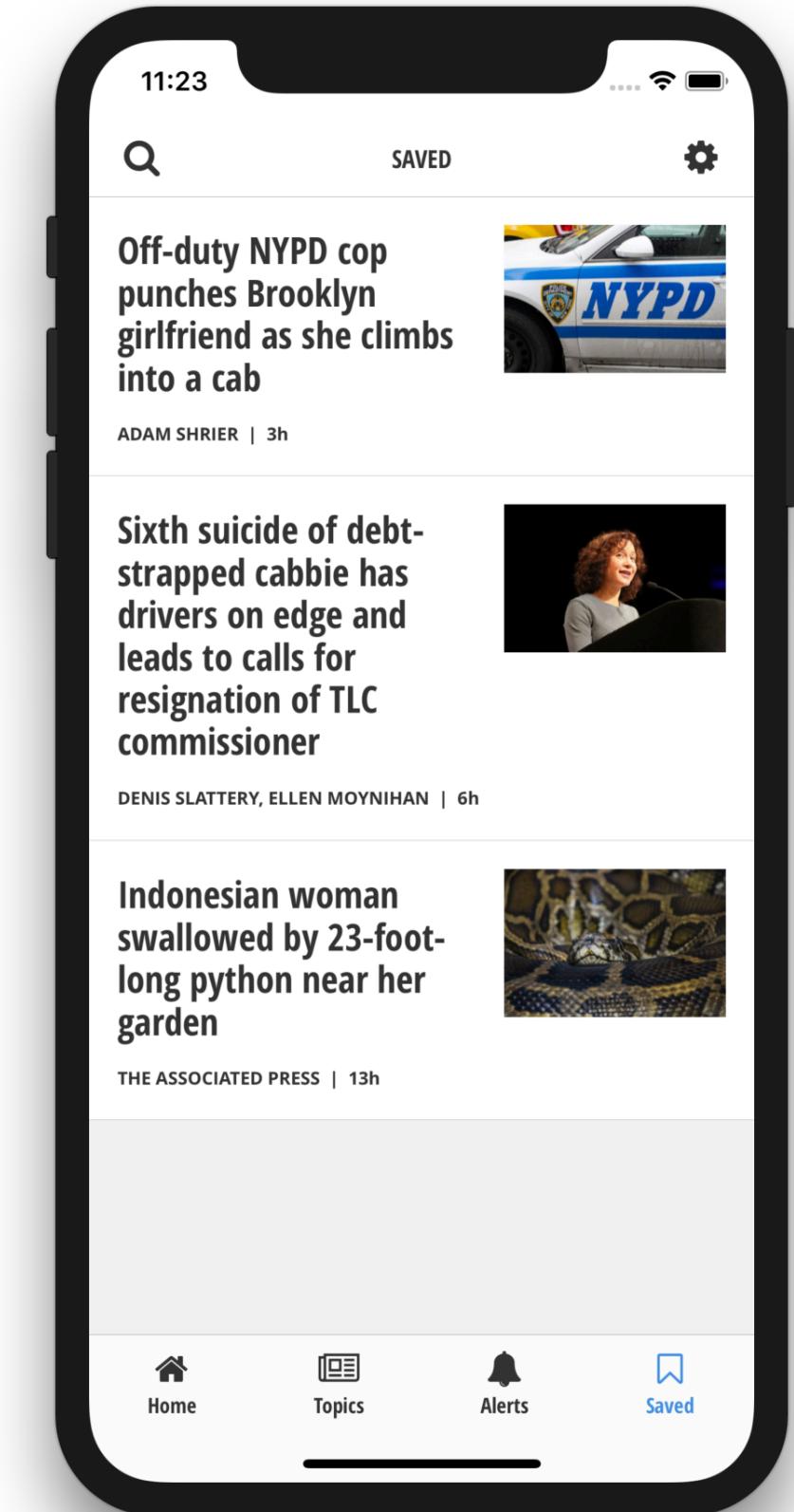
Structure of a News App

- Section Fronts
- Articles
- Section List
- Search
- Saved Articles
- Segmented Push Notifications



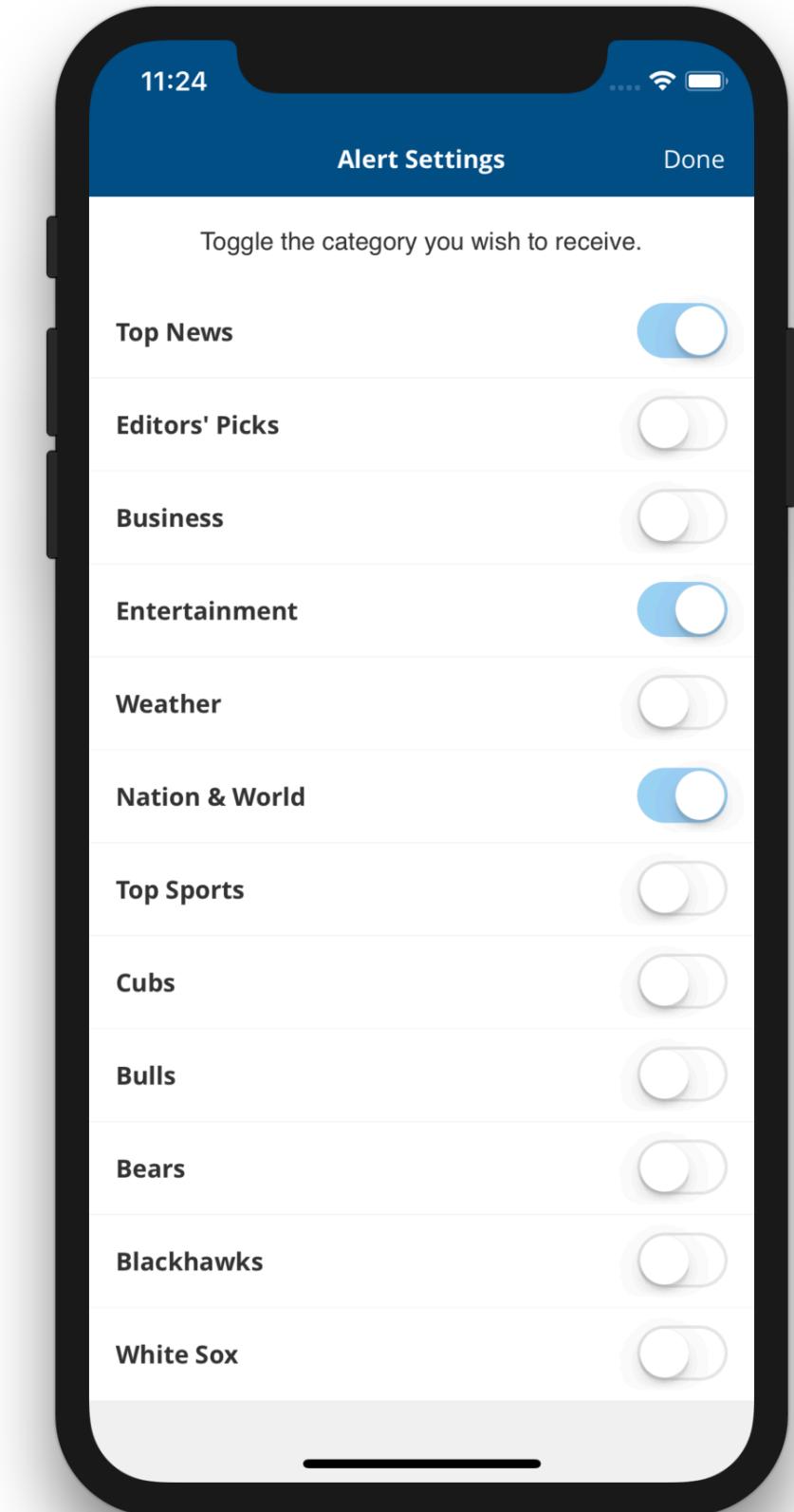
Structure of a News App

- Section Fronts
- Articles
- Section List
- Search
- Saved Articles
- Segmented Push Notifications



Structure of a News App

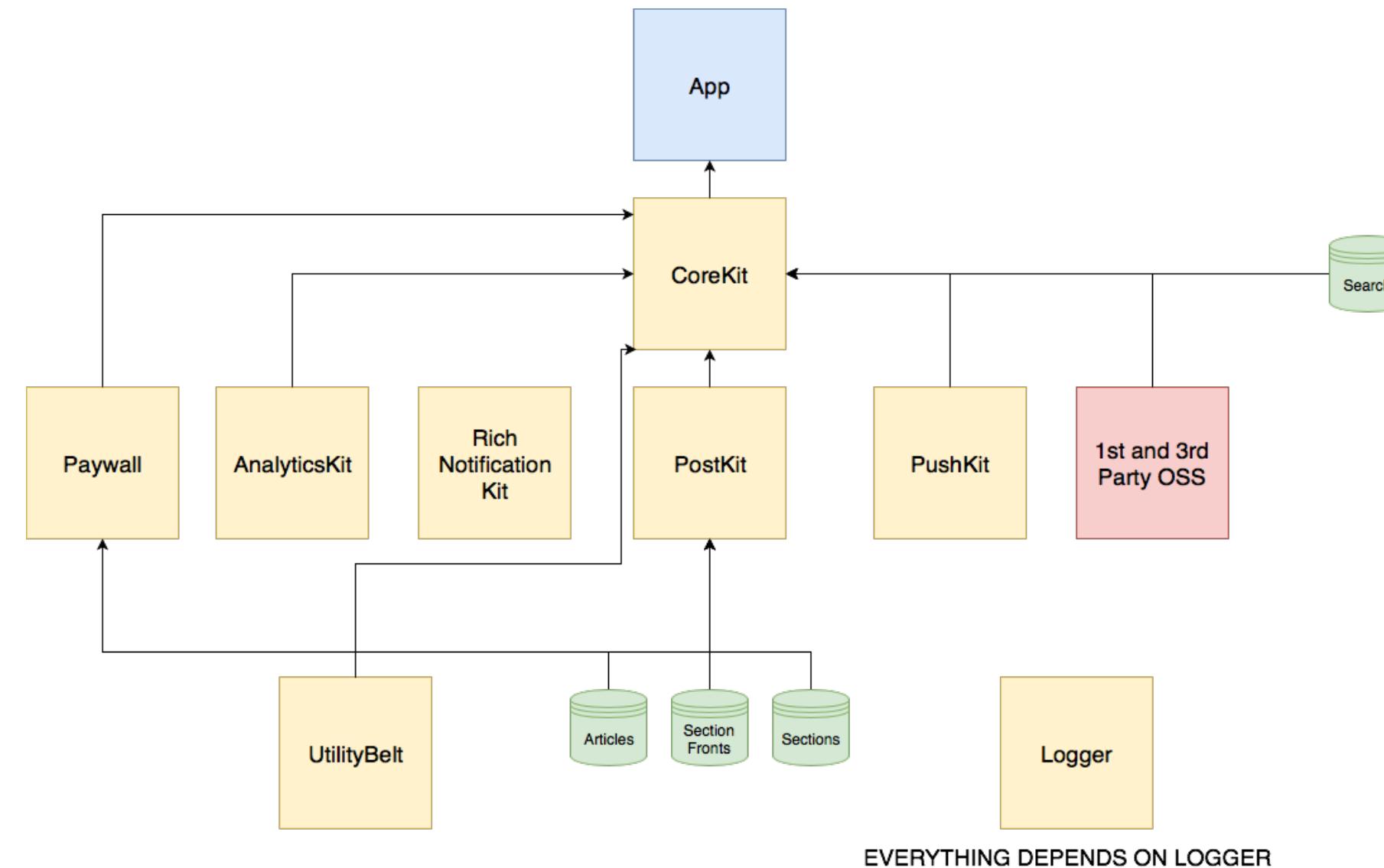
- Section Fronts
- Articles
- Section List
- Search
- Saved Articles
- Segmented Push Notifications



App Customization Options

- Ad Provider(s)
- Analytics Provider(s)
- Paywall Provider
- Push Notification Provider
- Theming (Colors, Fonts, Images, Paragraph Styles)
- Traffic Provider
- Weather Provider

High-Level Architecture of a Unicorn App



Partner.swift

```
struct Partner {  
    static let shared = Partner()  
    struct Colors {...}  
    struct Fonts {...}  
}  
  
extension Partner: AnalyticsManaging {...}  
extension Partner: AdsManging {...}  
extension Partner: PartnerManaging {...}  
extension Partner: PaywallManaging {...}  
extension Partner: PushManaging {...}  
extension Partner: Themeable {...}  
/// etc.
```

Partner+PartnerManaging.swift

```
extension Partner: PartnerManaging {  
  
    // Feature Toggles  
    var isOfflineModeEnabled: Bool { get }  
    var isPaywallEnabled: Bool { get }  
    var isSearchEnabled: Bool { get }  
  
    // Legal Toggles  
    var requiresTrackingConsent: Bool { get } // GDPR  
  
    // Cost-Saving Toggles  
    var searchOnUserInput: Bool { get } // e.g., Typeahead  
  
    /// and many more (seriously, tons of knobs to tweak)!  
}
```

Partner Setup

```
// PartnerAppDelegate.partner
override var partner: Any {
    return Partner.shared
}

// CoreAppDelegate().setupPartner()
open func setupPartner() {
    PartnerManager.currentPartner = partner as? PartnerManaging
        ?? PartnerManager.currentPartner

    // etc.
}

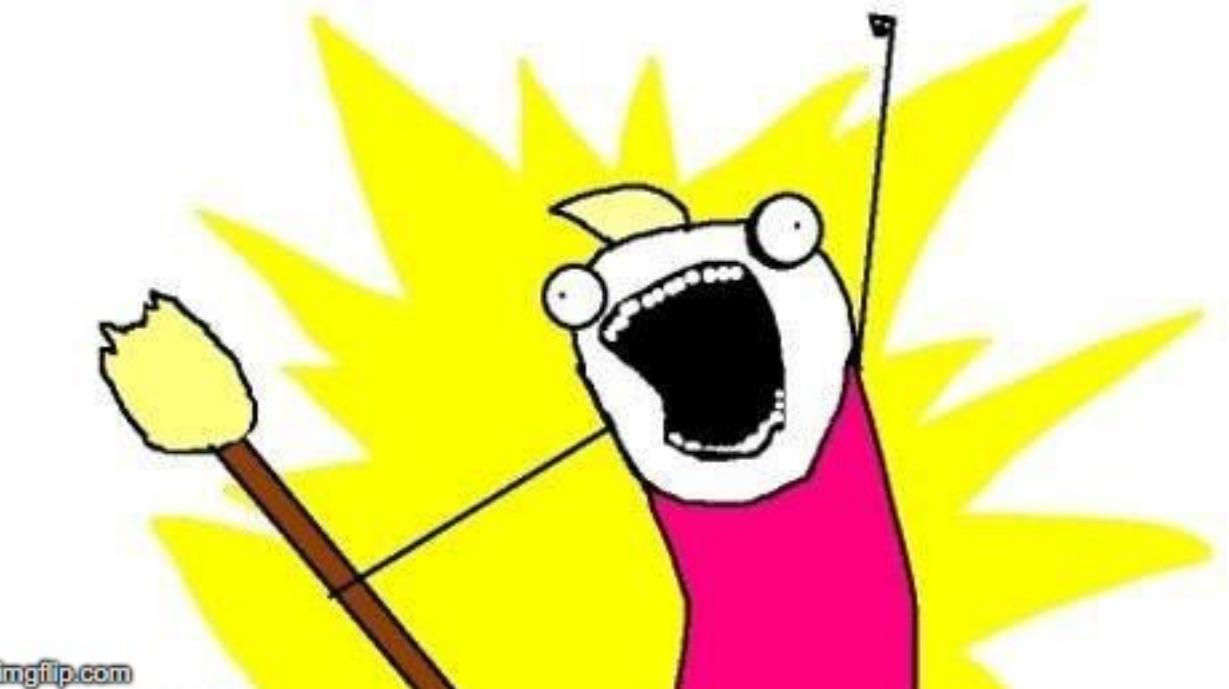
// PartnerManager.swift
public struct PartnerManager {
    public static var currentPartner: PartnerManaging = DefaultPartner()
}

// PartnerManaging.swift
public struct DefaultPartner: PartnerManaging {
    // Default PartnerManaging Implementation
}
```

Protocol Oriented Theming

- AppThemeable
- ArticleThemeable
- ImageThemable
- NavigationThemeable
- PaywallThemeable
- PushThemeable

THEME ALL THE THINGS!



imgflip.com

Problems with Existing Solution

- Current implementation does not scale as there are hundreds of knobs.
- Default protocol implementations make it hard to discover:
 - Renamed variables
 - New variables
 - Deprecated methods ([Swift Forums Post](#))

Future of Theming

Codable-backed JSON files

- **Style.json**

- A list of *all* the *tweakable* elements, broken up in a declarative way by view, VC, and element type.

- **SuperStyle.json**

- A superset of *tweakable* elements that are customized and *trickledown_* to all the values defined in *Style.json*.

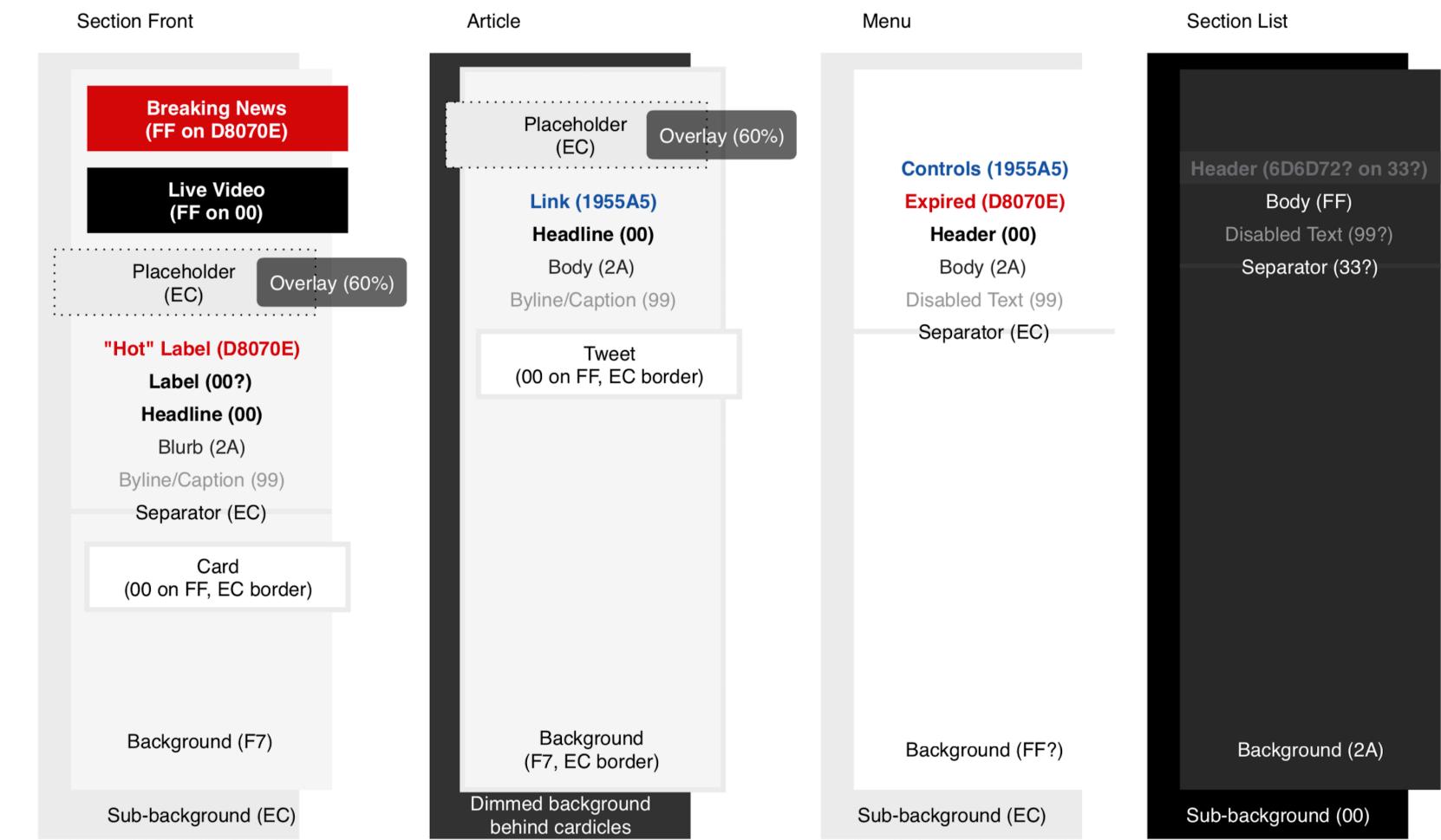
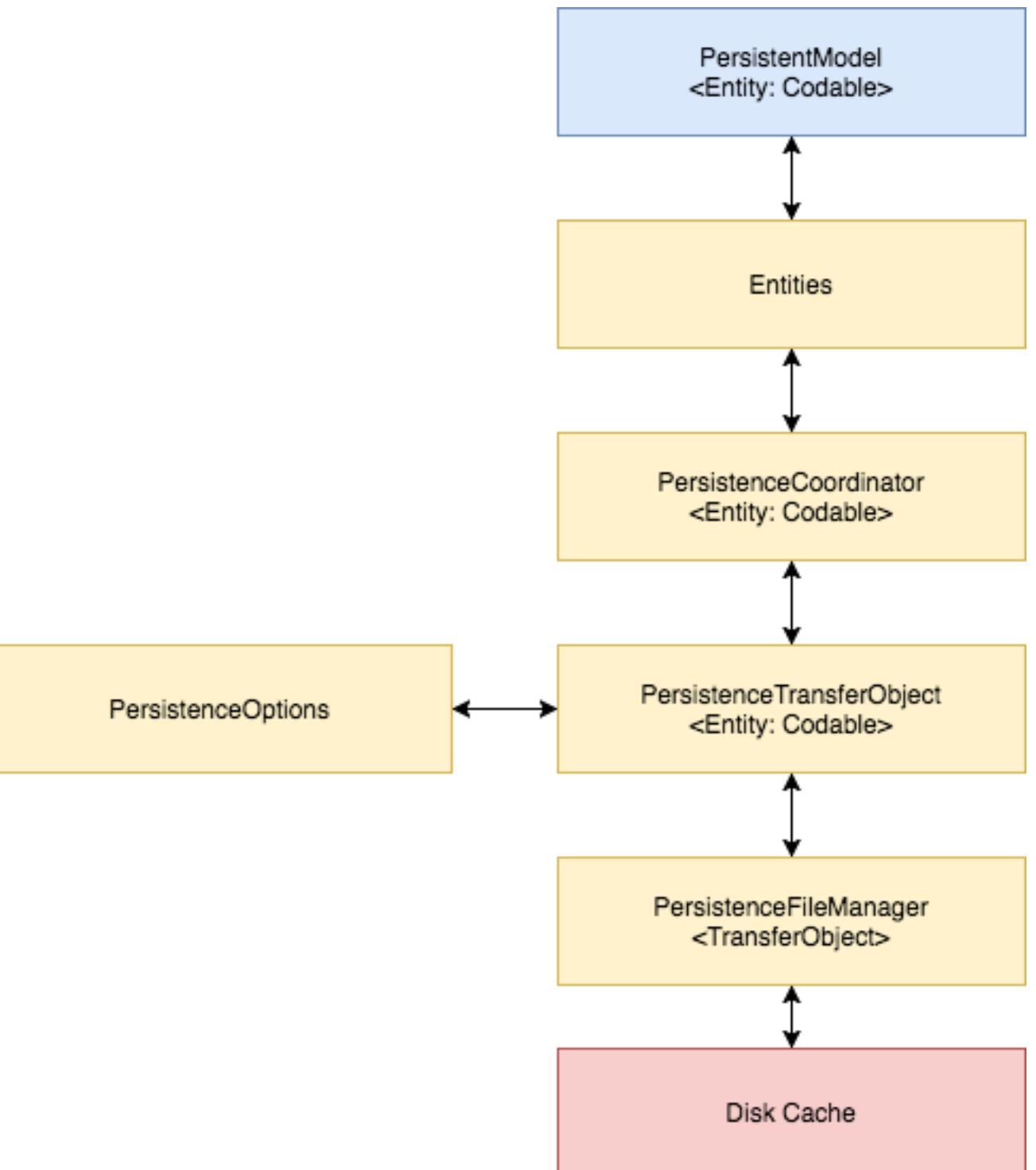


Image: Andrew Schoenfeld, Classic App Maintainer

Entities

A generic front-end that performs CRUD operations on Codable-conforming models.



Entities: Abstract Code Example

```
// Example: Main Configuration File
public var main: MainConfig? {
    get {
        return getValue(MainConfig.self, for: .main)
    } set {
        setValue(for: .main, tonewValue: newValue)
    }
}

func getValue<T: Codable>(_ type: T.Type,
                        for options: PersistenceOptions,
                        environmentOverride environment: Environment = EnvironmentManager.shared.current) -> T? {
    let key = coordinatorsKey(for: options, environment: environment)
    let coordinator = getCoordinator(for: key) as? PersistenceCoordinator<T> ??
        PersistenceCoordinator<T>(for: options, environmentOverride: environment)
    setCoordinator(coordinator, forKey: key)
    return coordinator.read()
}

func setValue<T: Codable>(for options: PersistenceOptions,
                        tonewValue newValue: T?,
                        environmentOverride environment: Environment = EnvironmentManager.shared.current) {
    let key = coordinatorsKey(for: options, environment: environment)
    let coordinator = getCoordinator(for: key) as? PersistenceCoordinator<T> ??
        PersistenceCoordinator<T>(for: options, environmentOverride: environment)
    coordinator.update(with: newValue)

    // If the `newValue` is `nil`, it has been deleted so remove the cached coordinator from the dictionary.
    setCoordinator(newValue != nil ? coordinator : nil, forKey: key)
}
```

PersistenceOptions

```
enum PersistenceOptions {
    case main
    case secure
    case paywall
    case ... // many more

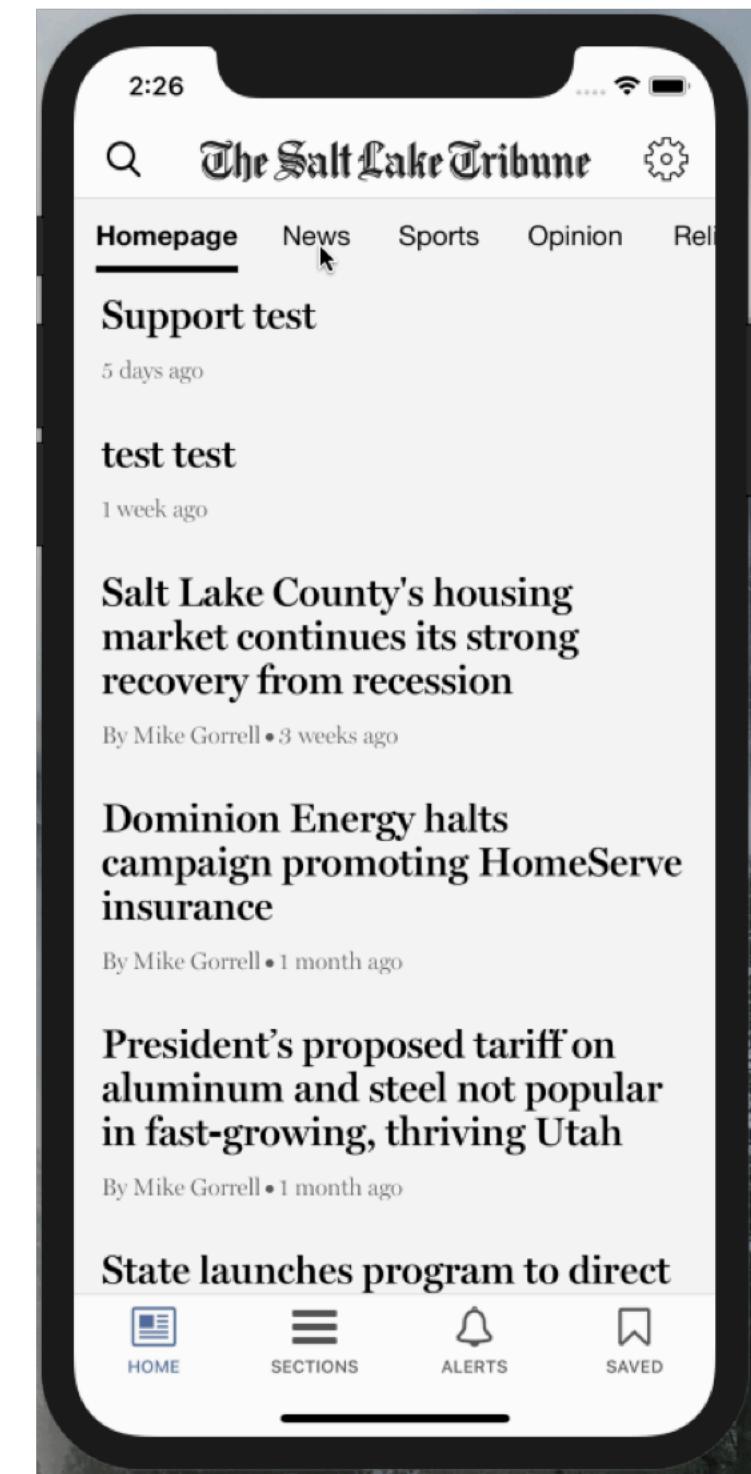
    /// Backup JSON file that is bundled with the app (may be an older version)
    var backupBundledFile: String?

    /// Decryption key for the model, if necessary.
    var decryptionKey: String?

    /// Disk Writing Options
    var fileWritingOptions: Data.WritingOptions {
        switch self {
        case .secure, .paywall:
            return [.atomic, .completeFileProtection]
        default:
            return .atomic
        }
    }
}
```

Environment Switching

- Built on top of **Entities**, environment switching allows partners to test their app against production, staging, and development environments.
- Useful for testing:
 - Section front layouts
 - Paywall API changes
- In the past we had to deploy new TestFlight builds whenever they wanted to test against a different environment.



CocoaPods - AnalyticsKit

```
s.static_framework = true
s.swift_version = '4.1'

s.ios.deployment_target = 9.0
s.tvos.deployment_target = 9.0

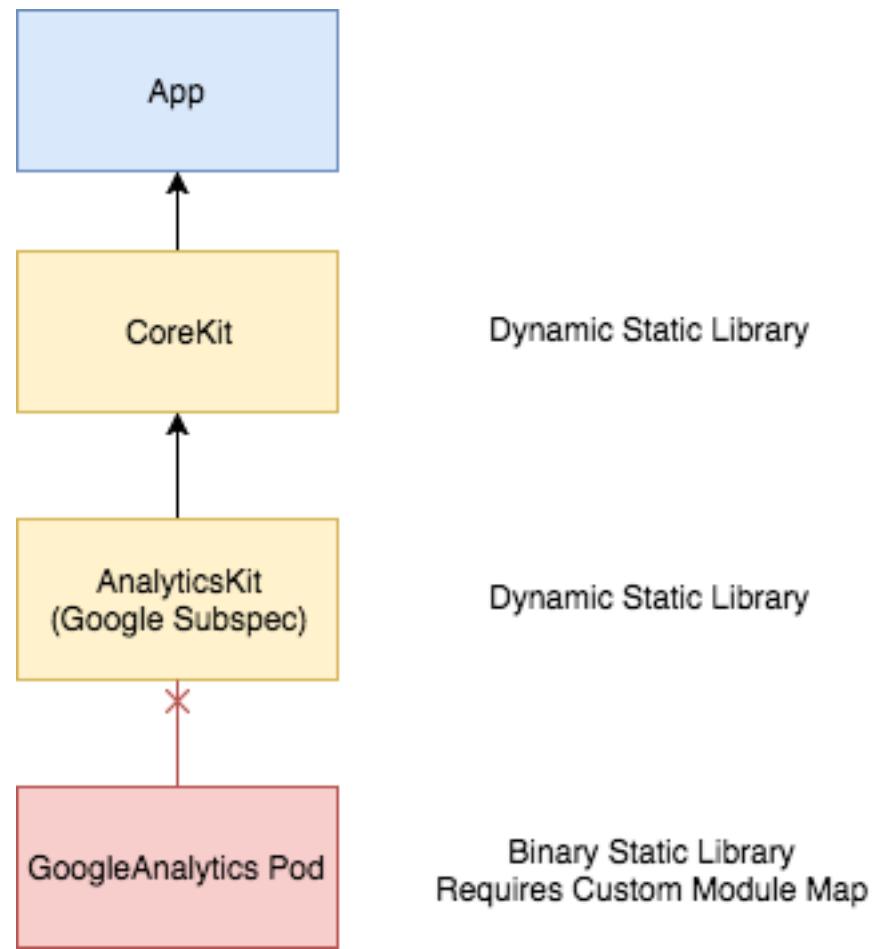
s.default_subspec = 'Core'
s.subspec 'Core' do |sub|
    sub.source_files = 'Sources/Core/**/*'
end

s.subspec 'ComScore' do |sub|
    sub.dependency 'AnalyticsKit/Core'
    sub.ios.dependency 'ComScore'
    sub.source_files = 'Sources/ComScore'
end

s.subspec 'Firebase' do |sub|
    sub.dependency 'AnalyticsKit/Core'
    sub.ios.dependency 'FirebaseAnalytics', '>= 4'
    sub.ios.source_files = 'Sources/Firebase'
end

s.subspec 'Omniture' do |sub|
    sub.dependency 'AnalyticsKit/Core'
    sub.ios.dependency 'AdobeMobileSDK', '>= 4.15.0'
    sub.tvos.dependency 'AdobeMobileSDK/TVOS', '>= 4.8.1'
    sub.source_files = 'Sources/Omniture'
end

// Any many many more!
```



BuddyBuild

- Built for *mobile* developers with both CI and CD and ***just works!*™**
- Deploys to ~~iTunes~~ App Store Connect automatically with custom rules.
- No need to maintain *fastlane* scripts or yaml files across dozens of repos.
- Only CI that did not fail during our tests (seriously - we spent a month)
 - We tested circle, travis, bitrise
 - We tested Xcode Bots on a Mac Mini (in 2015-2017) 😊 😂 😅

We ❤️ BuddyBuild!



“ For all 20 iOS applications and libraries we have under active development, we’re able to focus solely on improving our codebase and iterating on our projects, and we trust buddybuild to handle the rest.

Arthur Sabintsev, Lead iOS Engineer at the Washington Post



Thoughts on Open Source

I ❤️ contributing to Open Source.

I rarely use other people's Open Source tools.

Why?

Open Source Tools We Use ³

3rd Party

CocoaLumberjack

3rd Party

RNCryptor

1st Party

Failable

Eureka

SVProgressHUD

Freedom

Mockingjay

XCDYouTubeKit

Siren

³ Not listed here are base required SDKs (Ad SDKs, Analytics SDKs, Crash Monitoring, etc.)

Team Structure

- 1 Lead (Baltimore)
- 3 Seniors (DC, Denver, Tallahassee)
- 2 Juniors (Cincinnati, Manhattan)

Process

- GitFlow
- All code comes in via a PR
- Pair Programming for Code Reviews
- Weekly Kanban
- Daily Scrum
- Daily Check-in

What's Next?

- Arc Mobile SDK Offering
- Automating Upstream PRs
- Spin-up Apps with a push of a button via a web admin portal
- Theming Engine Rewrite

Thank You!