



The image displays a composite of various digital interfaces, all featuring the Brut brand. In the top left, a YouTube channel page for 'Brut' is shown, with a large 'Brut.' logo in white on a purple gradient background. To its right is a Twitter profile for '@brutofficiel', also featuring the 'Brut.' logo. Below these are two mobile devices: an iPhone displaying the Instagram feed of the same account (@brutofficiel), and another iPhone showing the Facebook profile for 'Brut' (@brutofficiel). The Facebook page includes a large 'Brut.' logo, and the Instagram feed shows several posts related to current events and celebrities.

YouTube FR

Rechercher

Brut.

twitter.com/brutofficiel?ref\_src=twsrctwsrc%5Egoogle%7Ctwcamp%5Eserp

Accueil À propos Recherchez sur Twitter Vous avez déjà un compte ? Se connecter

facebook

Brut.

Brut @brutofficiel Accueil Publications Vidéos À propos Photos Communauté Infos et publicités J'aime Partager Suggérer des modifications Utiliser l'appli... Envoyer un message

Brut.

Brut avec vos snaps Tous les jours, Brut raconte les grands événements avec vous ! www.brut.media

S'abonner Stories

Les 4 infos que tu as ratées Il y a 1 h

Ces snaps dans les lieux qui choquent 😱 Il y a 1 j

24h dans le monde en 2 Il y a 3 j

PNL a enfin sorti son album !

Rechercher publications sur

Communauté

2 252 757 personnes aiment ça

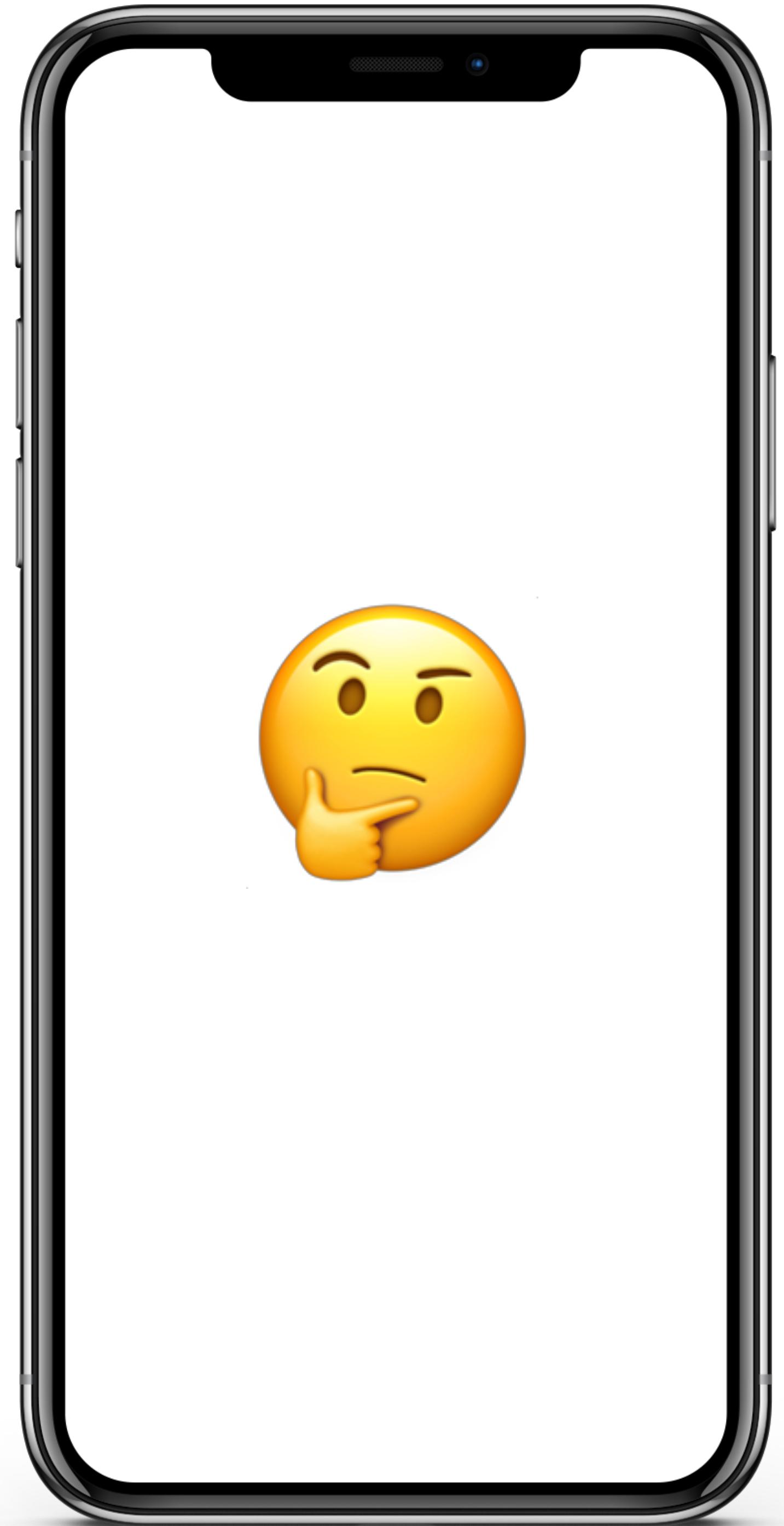
IGTV Centrafricaine L'appli Brut. Yao Tuto E-mail

brutofficiel

1 014 publications 501 K abonnés 247 abonnements Contacter

Brut Société de médias/d'actualités Brut. Info. Maintenant, partout. www.brut.media/fr Suivi(e) par harry.boh.17, jade\_lemaître, nevanoushka et 9 autres personnes

IGTV Centrafricaine L'appli Brut. Yao Tuto E-mail





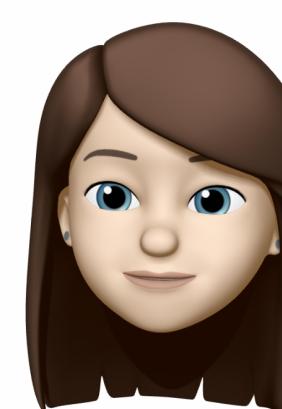
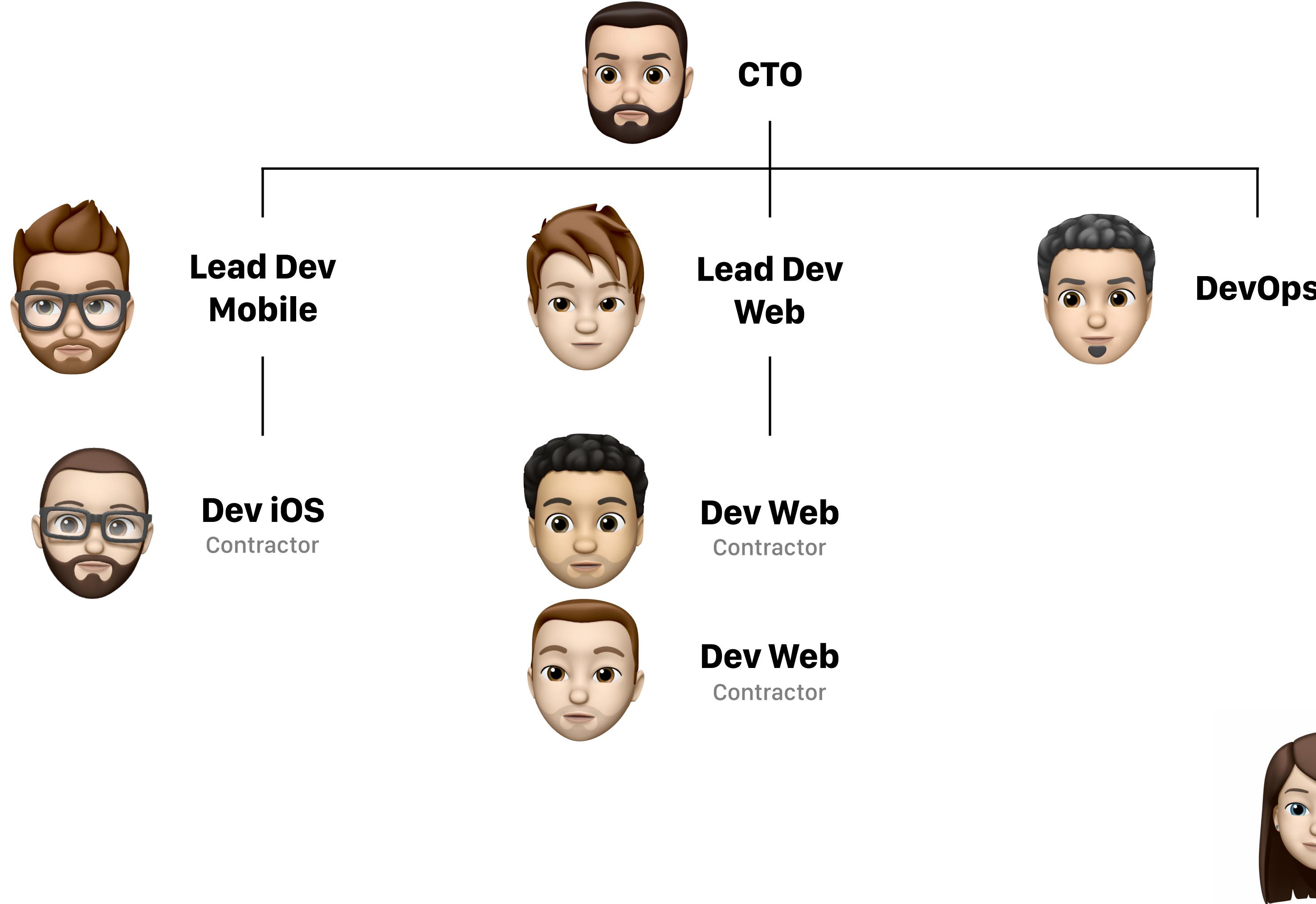


**1 000**  
**videos/month**

**800 M**  
**views/month**

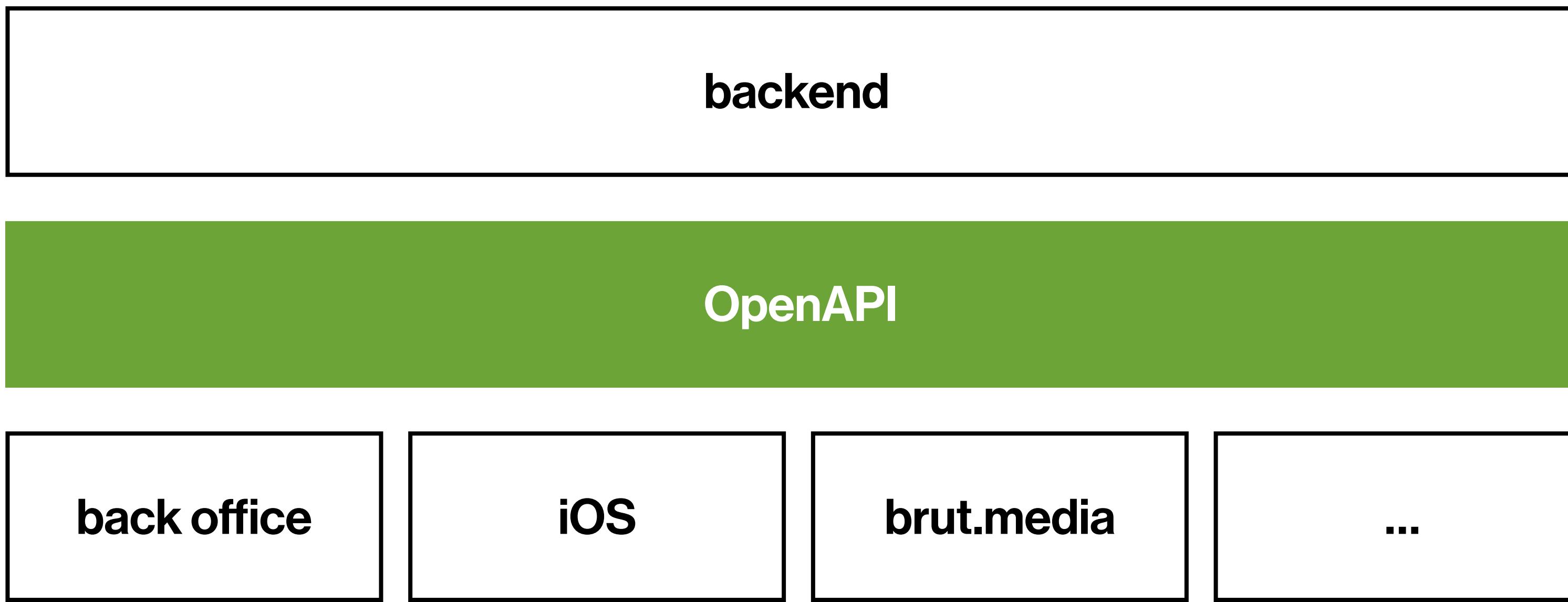
**10 M**  
**followers**

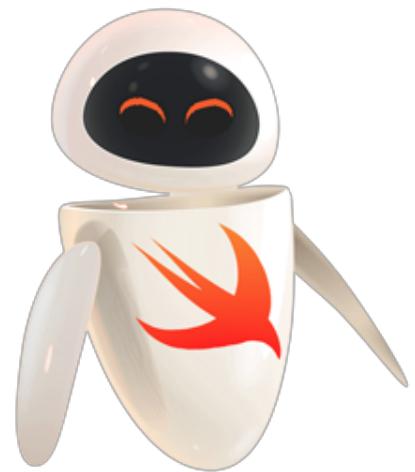
# Team



H5

**API contract**





**SwagGen**

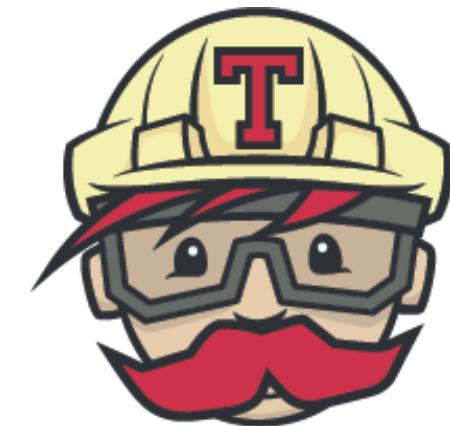


**SwiftLint**

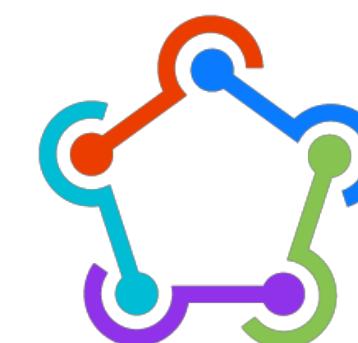


**lokalise**

**GitHub**



**Travis CI**



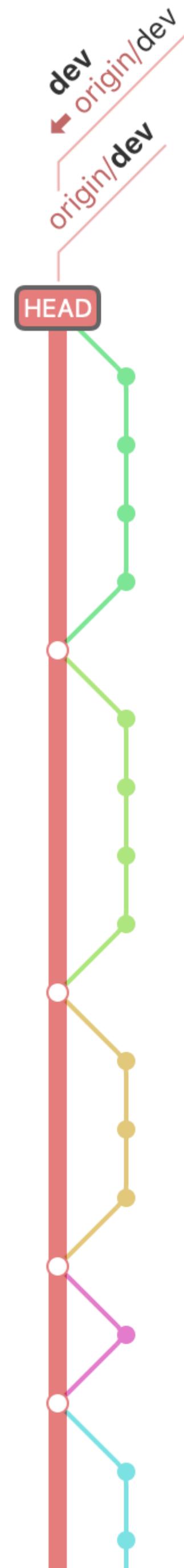
**fastlane**

**DANGER**

# SCRIPT



# ALL THE THINGS



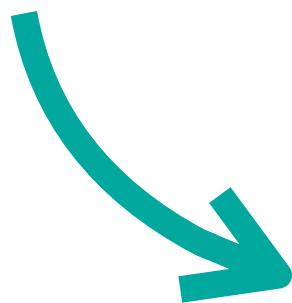
master dev

## Git strategy



- Pull requests must be rebased on dev before merging
- Merge on the master branch are done using a script to release a new build

## Release process

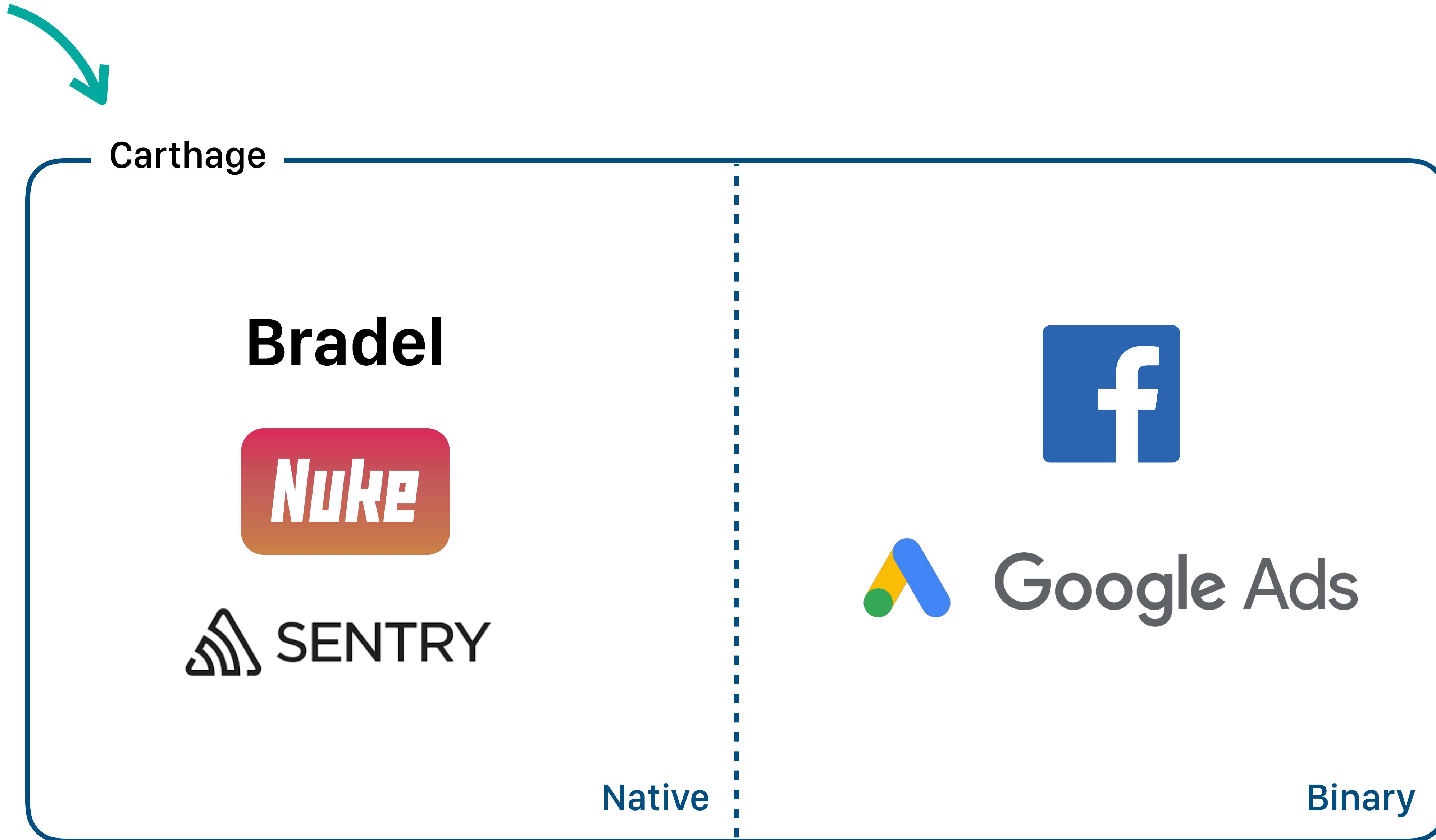


fastlane

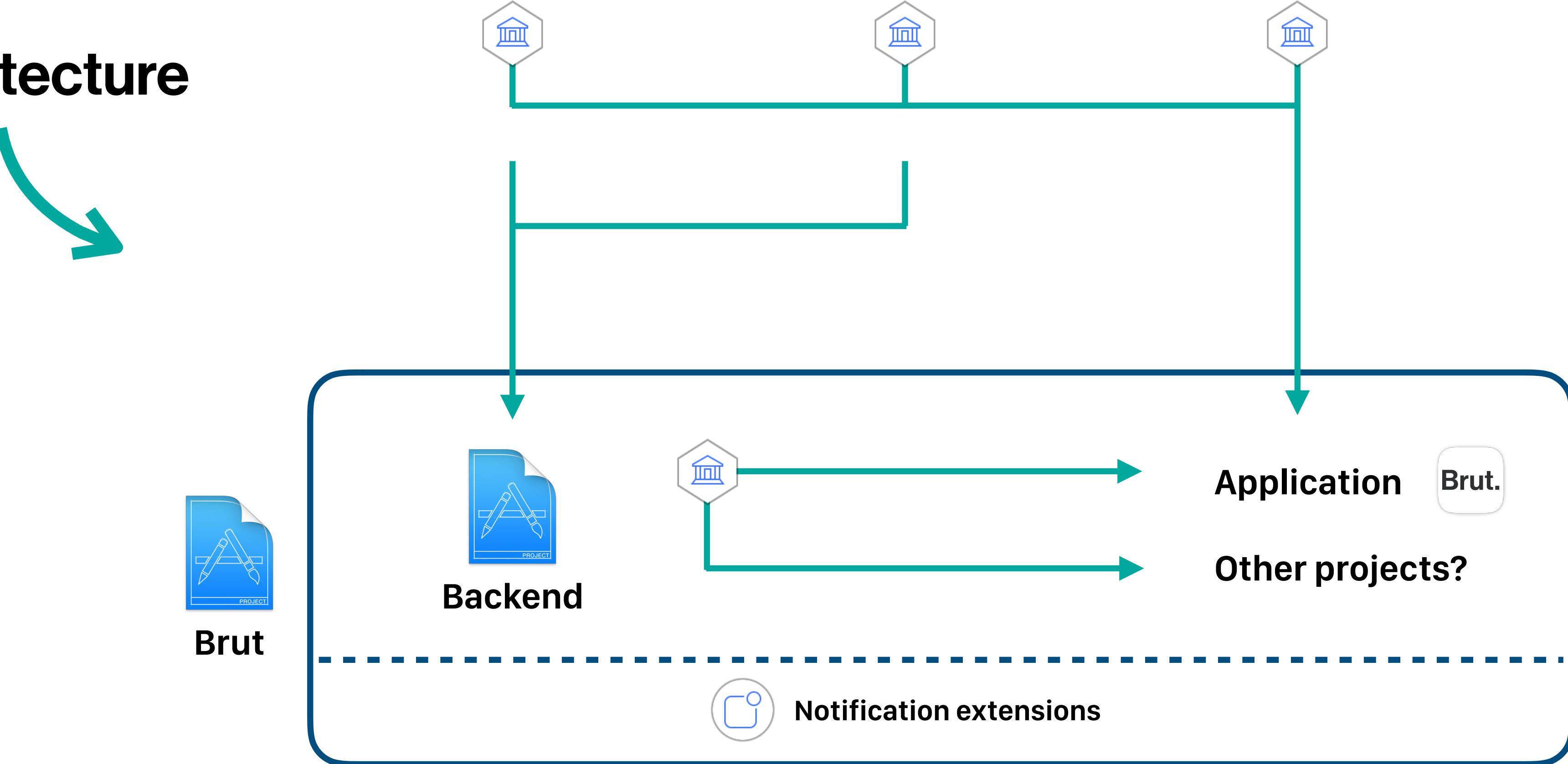


- Merge on master generates a TestFlight build
- Build number is Travis CI job number  
Version number is updated manually
- Generates a changelog with merge commits
- Uploads symbols to Sentry

# Dependencies



# Architecture



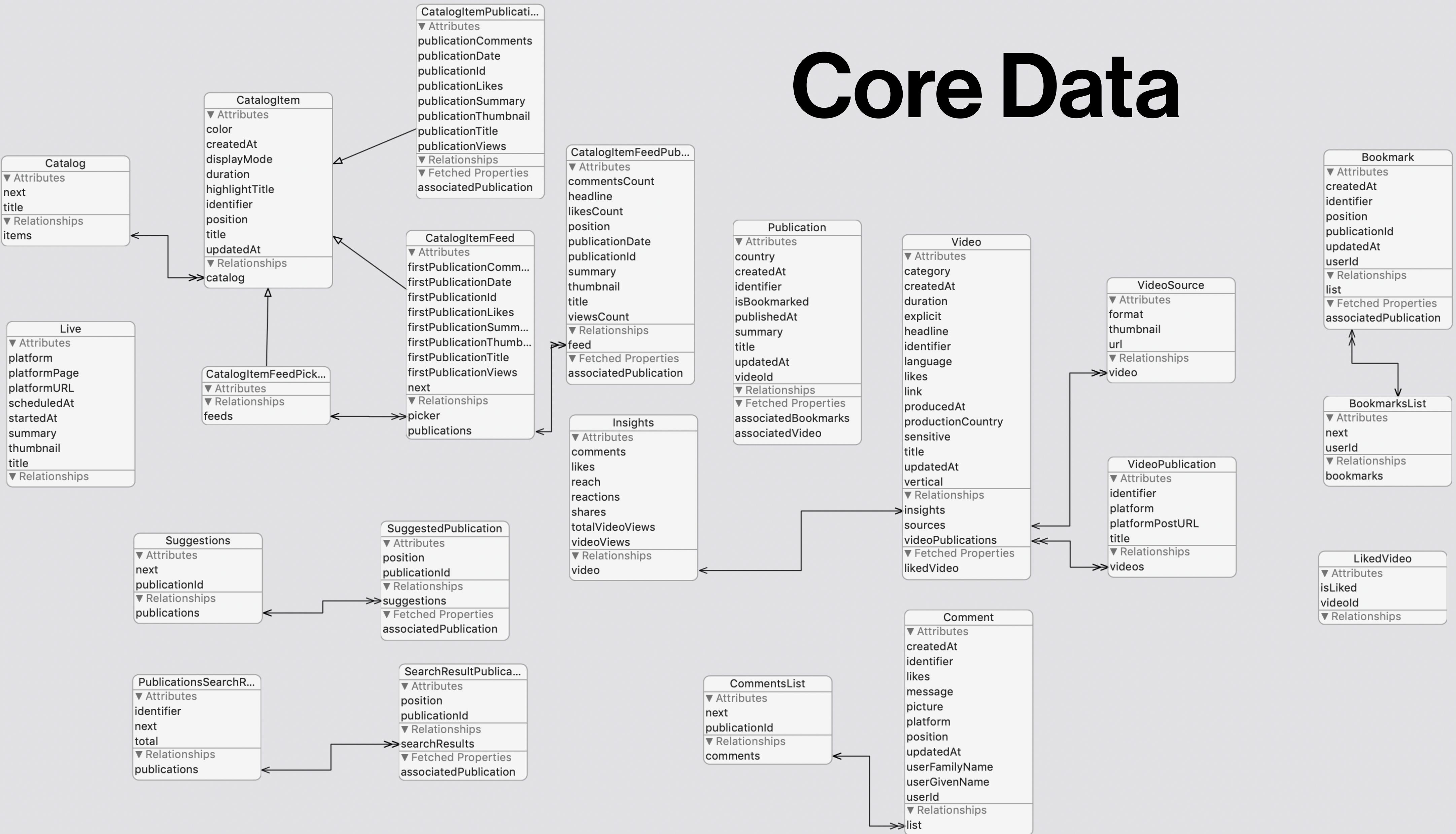
# Main project

- Views
- View controllers
- View models

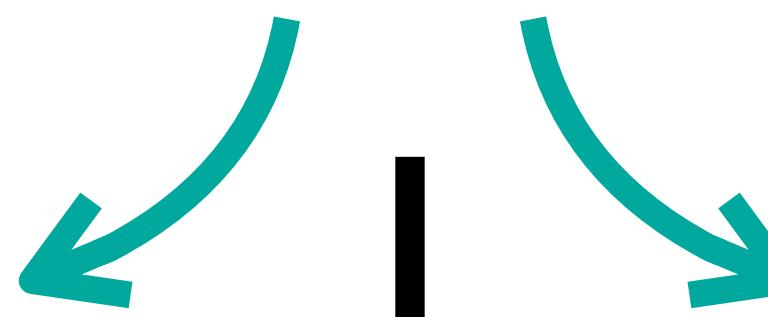
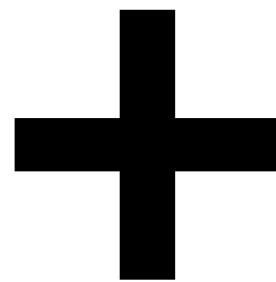
# Backend

- Networking code
- CoreData entities
- Data fetching routines

# Core Data



## Core Data



- Native solution
- UI tool to create the App model layer
- Manage caching for us

- High learning curve
- Black box with potential pitfalls

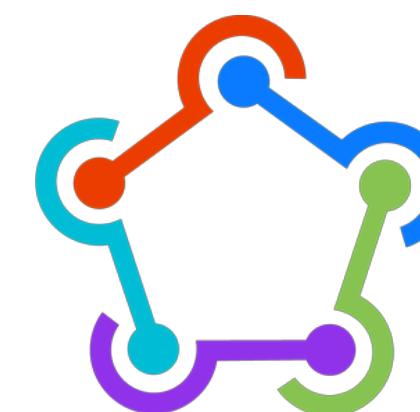


- **Open-source error reporting tool**
- **Privacy-compatible alternative to Fabric**
- **Flexible**
- **Covers all our needs**

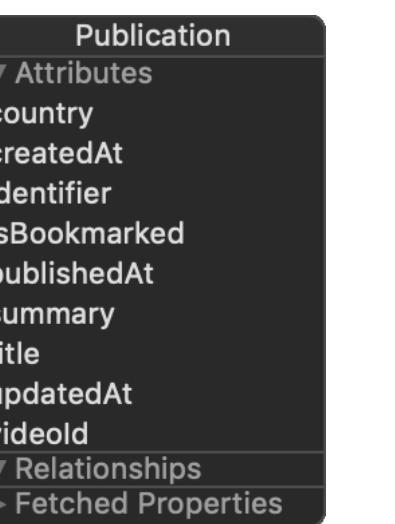
# Localization



- Lokalise to update and manage the localization strings
- fastlane lane to download the strings/stringsdict files and install them in the project
- SwiftGen to have strongly-typed identifiers



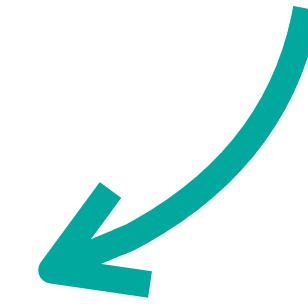
# Bindings



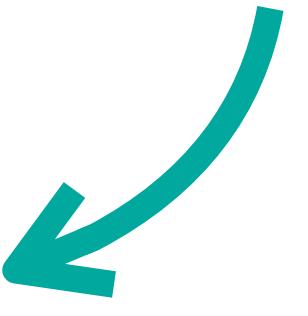
?



```
Publication:
  type: object
  required:
    - publication_id
    - title
    - media_id
    - created_at
    - updated_at
    - platform
    - country
  properties:
    publication_id:
      type: string
    title:
      type: string
    description:
      type: string
    media_id:
      type: string
    media_type:
      type: string
    media:
      $ref: '#/definitions/Video'
    created_at:
      type: string
      format: date-time
    updated_at:
      type: string
      format: date-time
    published_at:
      type: string
      format: date-time
    published_format:
      type: string
    platform:
      type: string
    platform_page:
      type: string
    platform_page_id:
      type: string
    platform_post_id:
      type: string
    platform_post_url:
      type: string
      format: uri
      pattern: ^https?://
    country:
      type: string
    is_bookmarked:
      type: boolean
```



# Bindings



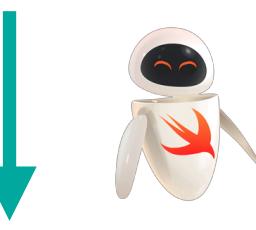
```
Publication:
  type: object
  required:
    - publication_id
    - title
    - media_id
    - created_at
    - updated_at
    - platform
    - country
  properties:
    publication_id:
      type: string
    title:
      type: string
    description:
      type: string
    media_id:
      type: string
    media_type:
      type: string
    media:
      $ref: '#/definitions/Video'
    created_at:
      type: string
      format: date-time
    updated_at:
      type: string
      format: date-time
    published_at:
      type: string
      format: date-time
    published_format:
      type: string
    platform:
      type: string
    platform_page:
      type: string
    platform_page_id:
      type: string
    platform_post_id:
      type: string
    platform_post_url:
      type: string
      format: uri
      pattern: ^https?://
    country:
      type: string
    is_bookmarked:
      type: boolean
```

?



**SwiftGen**

```
Publication
  ▼ Attributes
  country
  createdAt
  identifier
  isBookmarked
  publishedAt
  summary
  title
  updatedAt
  videoId
  ▼ Relationships
  ▶ Fetched Properties
```



```
public class Publication: NSManagedObject {
  public class func entityName() -> String { ... }

  public class func entity(in managedObjectContext: NSManagedObjectContext) -> NSEntityDescription? { ... }

  @nonobjc public class func fetchRequest() -> NSFetchedResultsController<Publication> {
    // swiftlint:disable discouraged_optional_boolean discouraged_optional_type_annotation
    let fetchRequest = NSFetchedResultsController<NSManagedObject>(entity: Publication.entity(), context: managedObjectContext)
    fetchRequest.fetchBatchSize = 20
    return fetchRequest
  }

  @NSManaged public var country: String
  @NSManaged public var created_at: Date

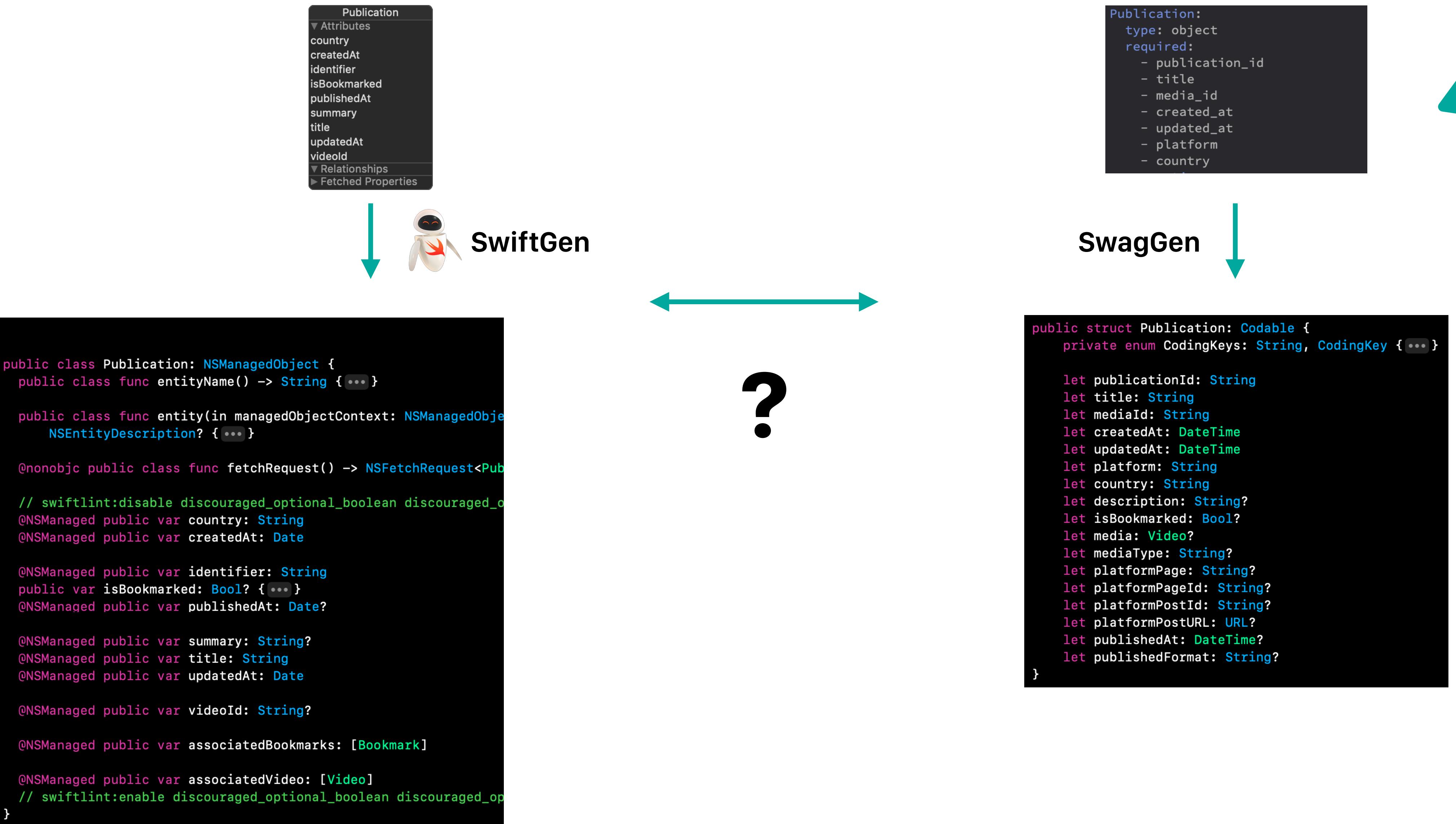
  @NSManaged public var identifier: String
  public var isBookmarked: Bool? { ... }
  @NSManaged public var publishedAt: Date?

  @NSManaged public var summary: String?
  @NSManaged public var title: String
  @NSManaged public var updatedAt: Date

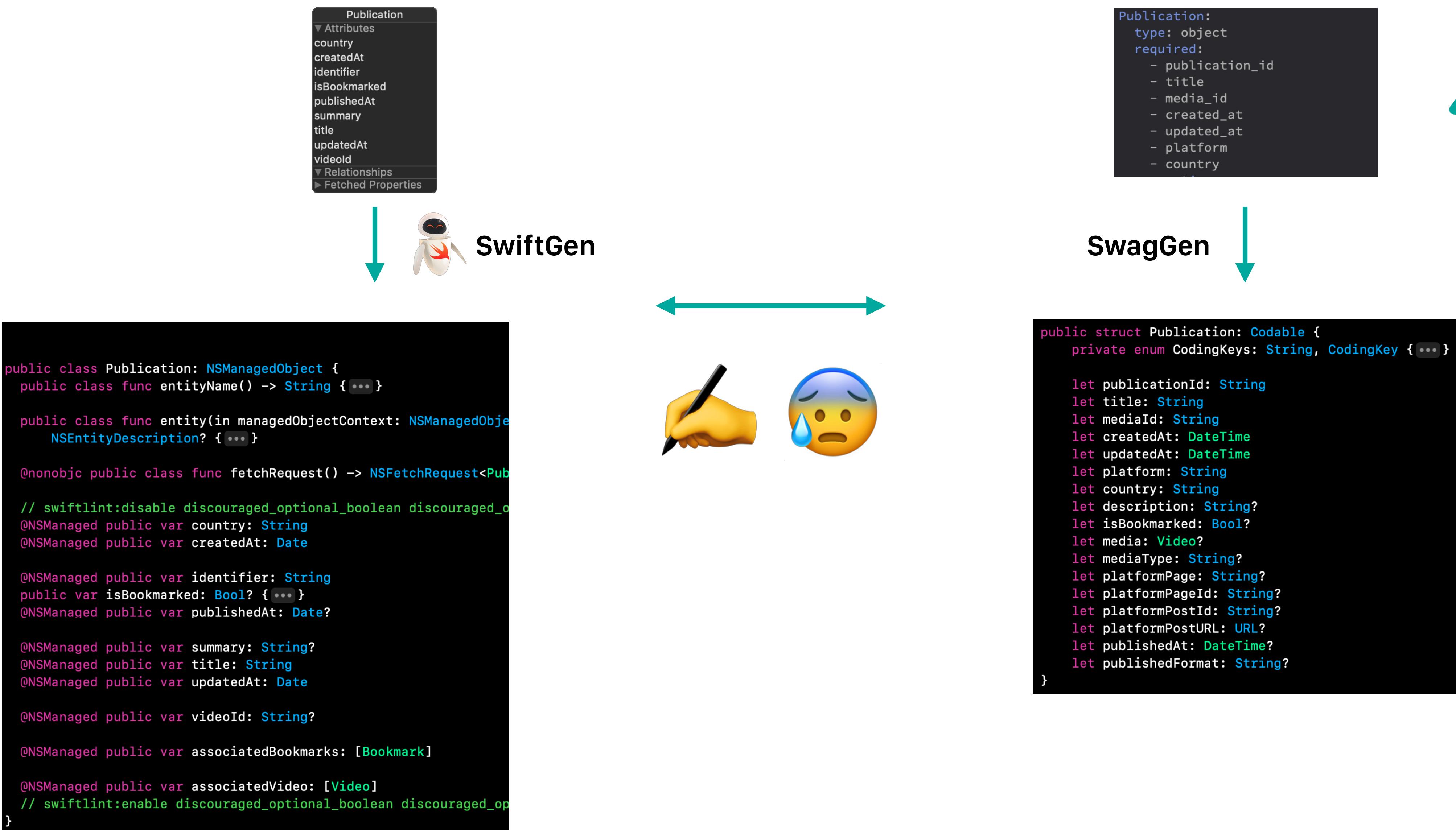
  @NSManaged public var videoId: String?

  @NSManaged public var associatedBookmarks: [Bookmark]
  @NSManaged public var associatedVideo: [Video]
  // swiftlint:enable discouraged_optional_boolean discouraged_optional_type_annotation
}
```

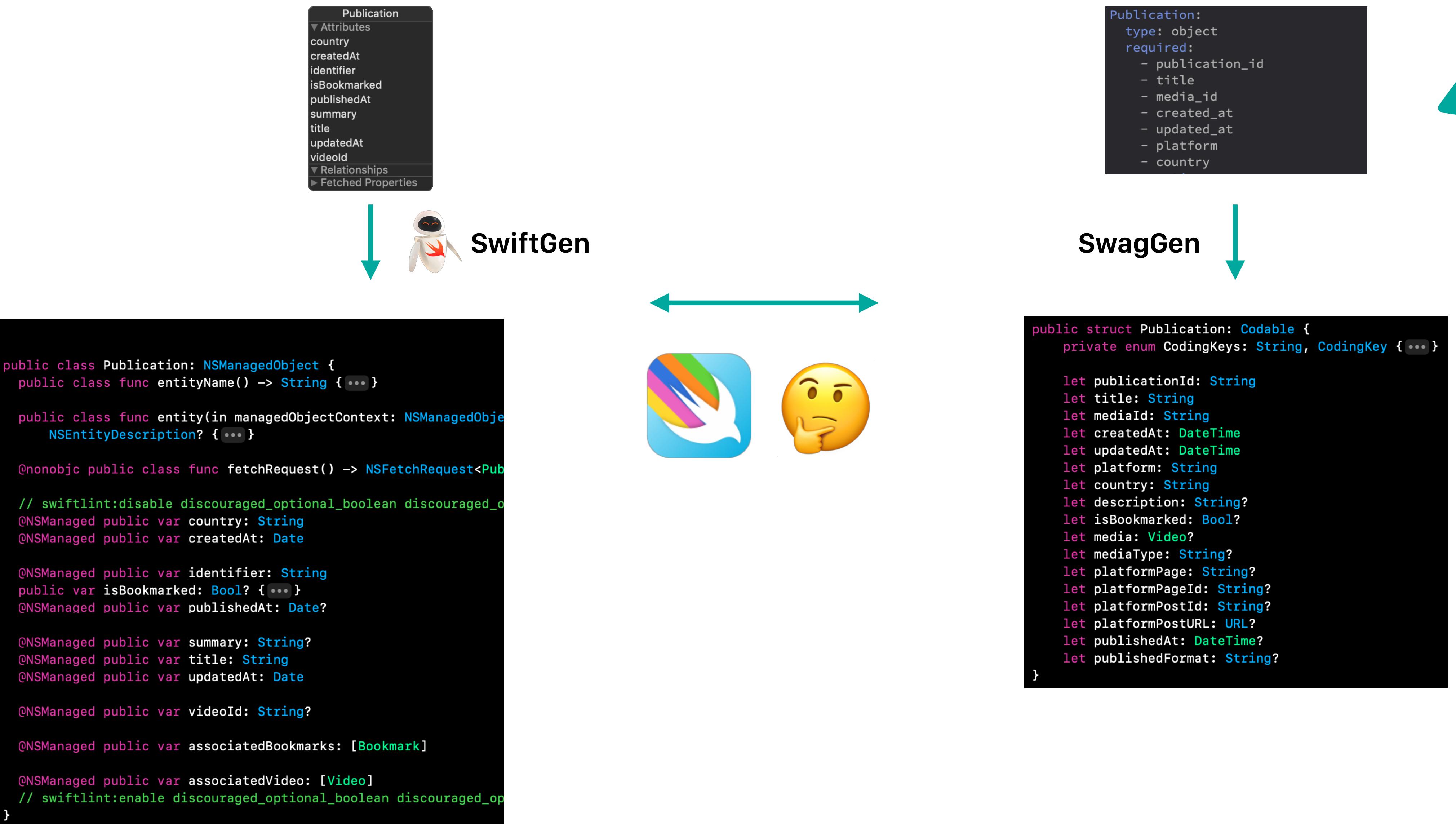
# Bindings



# Bindings



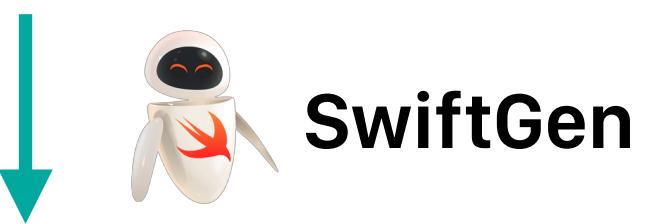
# Bindings



# Bindings

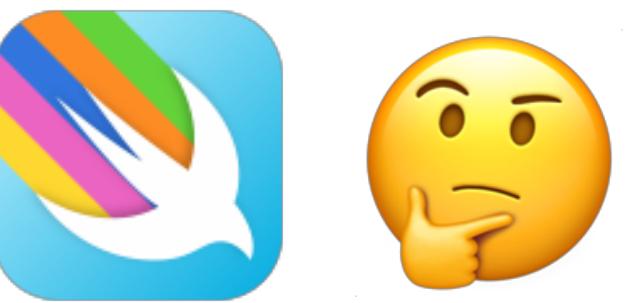
```
User Info  
Key      Value  
bindings.active true  
bindings.inline true  
+ -
```

```
Publication  
▼ Attributes  
country  
createdAt  
identifier  
isBookmarked  
publishedAt  
summary  
title  
updatedAt  
videoId  
▼ Relationships  
► Fetched Properties
```



SwiftGen

```
// sourcery:bindings:  
// sourcery:bindings: inline = true  
public class Publication: NSManagedObject {  
    public class func entityName() -> String { ... }  
  
    public class func entity(in managedObjectContext: NSManagedObjectContext) -> NSEntityDescription? { ... }  
  
    @nonobjc public class func fetchRequest() -> NSFetchedResultsController<Publication>  
  
    // swiftlint:disable discouraged_optional_boolean discouraged_optional_type  
    @NSManaged public var country: String  
    @NSManaged public var createdAt: Date  
  
    @NSManaged public var identifier: String  
    public var isBookmarked: Bool? { ... }  
    @NSManaged public var publishedAt: Date?  
  
    @NSManaged public var summary: String?  
    @NSManaged public var title: String  
    @NSManaged public var updatedAt: Date  
  
    @NSManaged public var videoId: String?  
  
    @NSManaged public var associatedBookmarks: [Bookmark]  
  
    @NSManaged public var associatedVideo: [Video]  
    // swiftlint:enable discouraged_optional_boolean discouraged_optional_type  
}
```

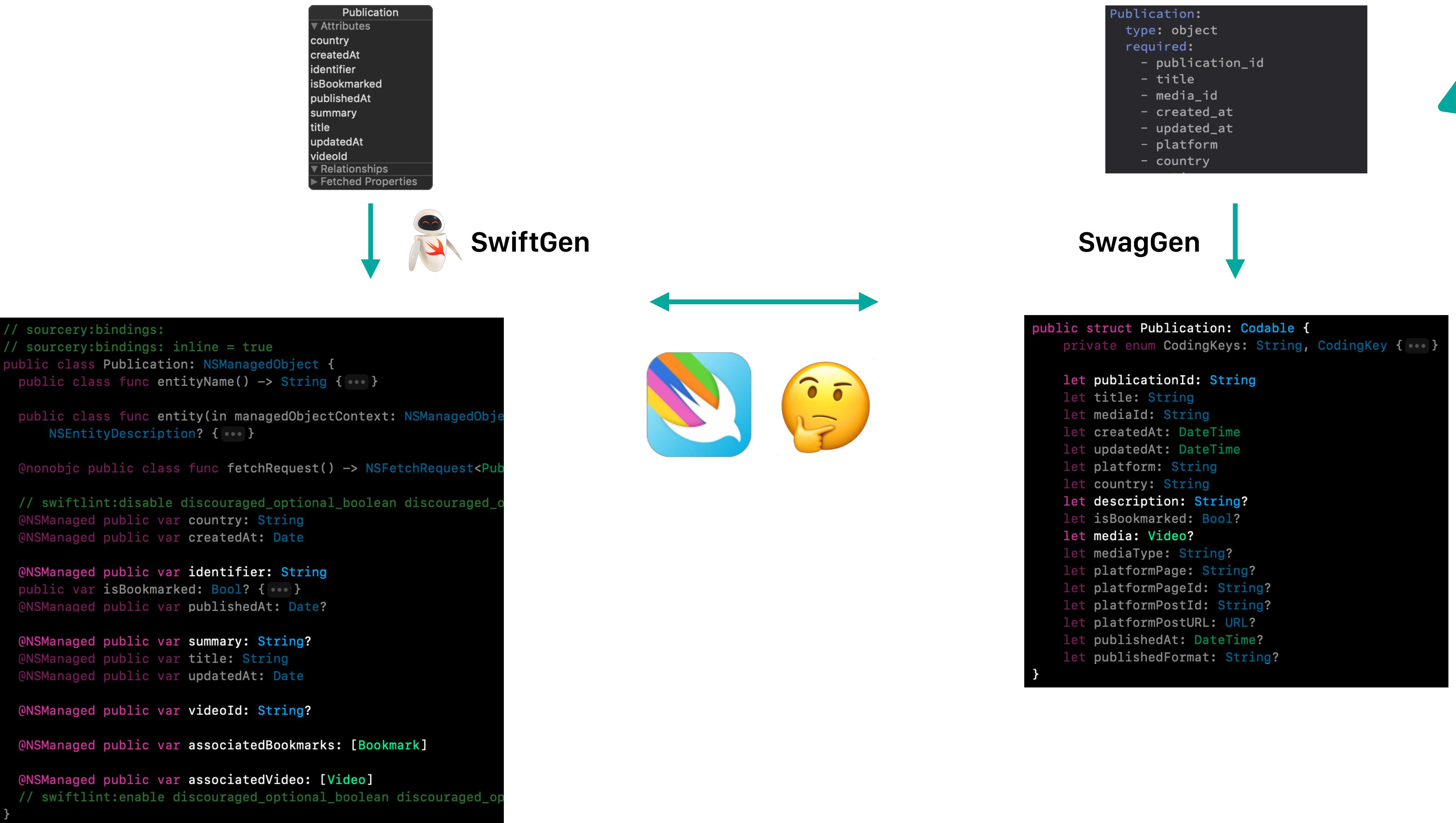


```
Publication:  
  type: object  
  required:  
    - publication_id  
    - title  
    - media_id  
    - created_at  
    - updated_at  
    - platform  
    - country
```

SwagGen

```
public struct Publication: Codable {  
    private enum CodingKeys: String, CodingKey { ... }  
  
    let publicationId: String  
    let title: String  
    let mediaId: String  
    let createdAt: DateTime  
    let updatedAt: DateTime  
    let platform: String  
    let country: String  
    let description: String?  
    let isBookmarked: Bool?  
    let media: Video?  
    let mediaType: String?  
    let platformPage: String?  
    let platformPageId: String?  
    let platformPostId: String?  
    let platformPostURL: URL?  
    let publishedAt: DateTime?  
    let publishedFormat: String?  
}
```

# Bindings



# Bindings

Attribute

Name summary

Properties  Transient  Optional

Attribute Type String

Validation No Value  Min Length  
No Value  Max Length

Default Value Default Value

Reg. Ex. Regular Expression

Advanced  Index in Spotlight  Preserve After Deletion

Deprecated

Spotlight  Store in External Record File

User Info

Key Value

bindings.key description

+ -

Publication

Attributes

- country
- createdAt
- identifier
- isBookmarked
- publishedAt
- summary
- title
- updatedAt
- videoId

Relationships

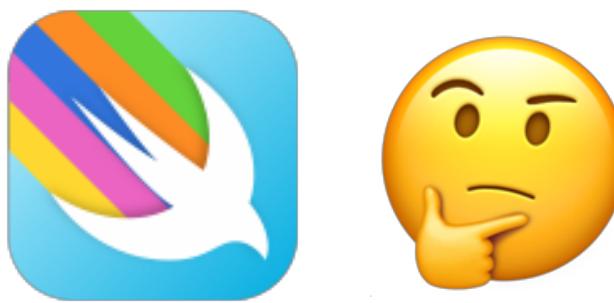
Fetched Properties

Publication:

```
type: object
required:
  - publication_id
  - title
  - media_id
  - created_at
  - updated_at
  - platform
  - country
```



## SwiftGen



## SwagGen

```
public struct Publication: Codable {
    private enum CodingKeys: String, CodingKey { ... }

    let publicationId: String
    let title: String
    let mediaId: String
    let createdAt: DateTime
    let updatedAt: DateTime
    let platform: String
    let country: String
    let description: String?
    let isBookmarked: Bool?
    let media: Video?
    let mediaType: String?
    let platformPage: String?
    let platformPageId: String?
    let platformPostId: String?
    let platformPostURL: URL?
    let publishedAt: DateTime?
    let publishedFormat: String?
}
```

```
// sourcery:bindings:
// sourcery:bindings: inline = true
public class Publication: NSManagedObject {
    public class func entityName() -> String { ... }

    public class func entity(in managedObjectContext: NSManagedObjectContext) -> NSEntityDescription? { ... }

    @nonobjc public class func fetchRequest() -> NSFetchedResultsController<Publication> {
        // swiftlint:disable discouraged_optional_boolean discouraged_optional_type_annotation
        @NSManaged public var country: String
        @NSManaged public var createdAt: Date
        // sourcery:bindings: key = "publicationId"
        @NSManaged public var identifier: String
        public var isBookmarked: Bool? { ... }
        @NSManaged public var publishedAt: Date?
        // sourcery:bindings: key = "description"
        @NSManaged public var summary: String?
        @NSManaged public var title: String
        @NSManaged public var updatedAt: Date
        // sourcery:bindings: key = "media.videoId"
        @NSManaged public var videoId: String?
        // sourcery:bindings: skip = true
        @NSManaged public var associatedBookmarks: [Bookmark]
        // sourcery:bindings: skip = true
        @NSManaged public var associatedVideo: [Video]
        // swiftlint:enable discouraged_optional_boolean discouraged_optional_type_annotation
    }
}
```

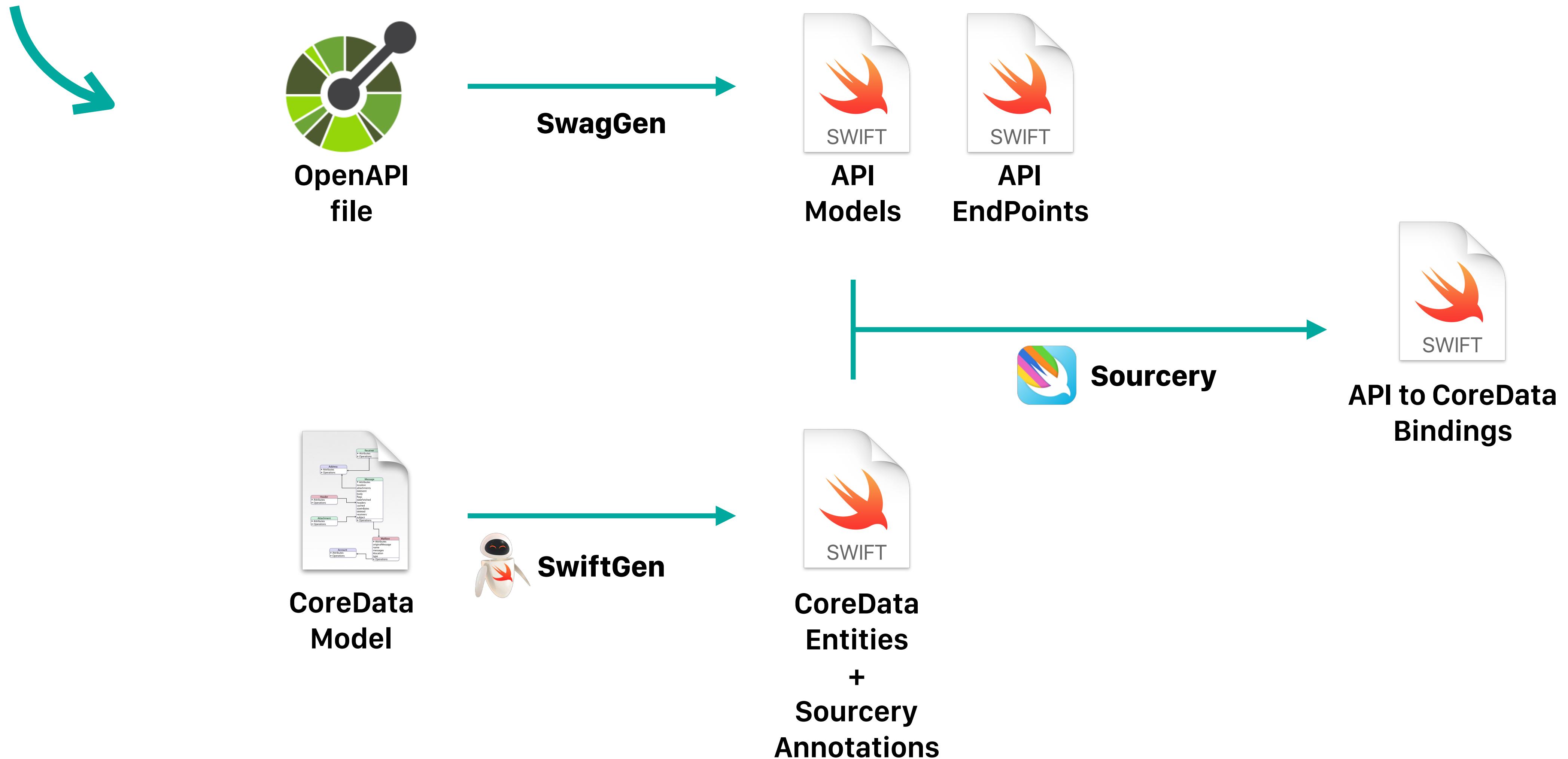
# Bindings



# Bindings



# API & model bindings





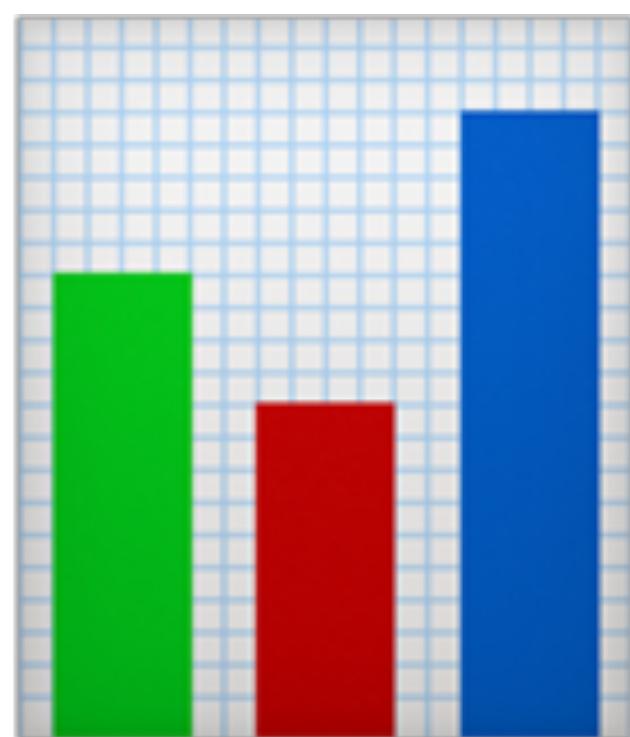
# Product

**Create a label  
with a red background  
that follows text**

**Now open-source:  
[github.com/brutmedia/labelrouge](https://github.com/brutmedia/labelrouge)**







# Questions?

