

# Firebase Beta features use-case Analysis

Tópicos Avançados  
Senac - Porto Alegre  
Guilherme Matuella



# Guilherme Matuella



Fullstack Developer - **UI up to DevOps**;  
iOS and watchOS enthusiast;  
Apple Developer Academy and **Hercules**.

---

[linkedin.com/in/gmatuella/](https://linkedin.com/in/gmatuella/)

[github.com/matuella/](https://github.com/matuella/)

The Pragmatic Programmer: From Journeyman to Master, **Andy Hunt**



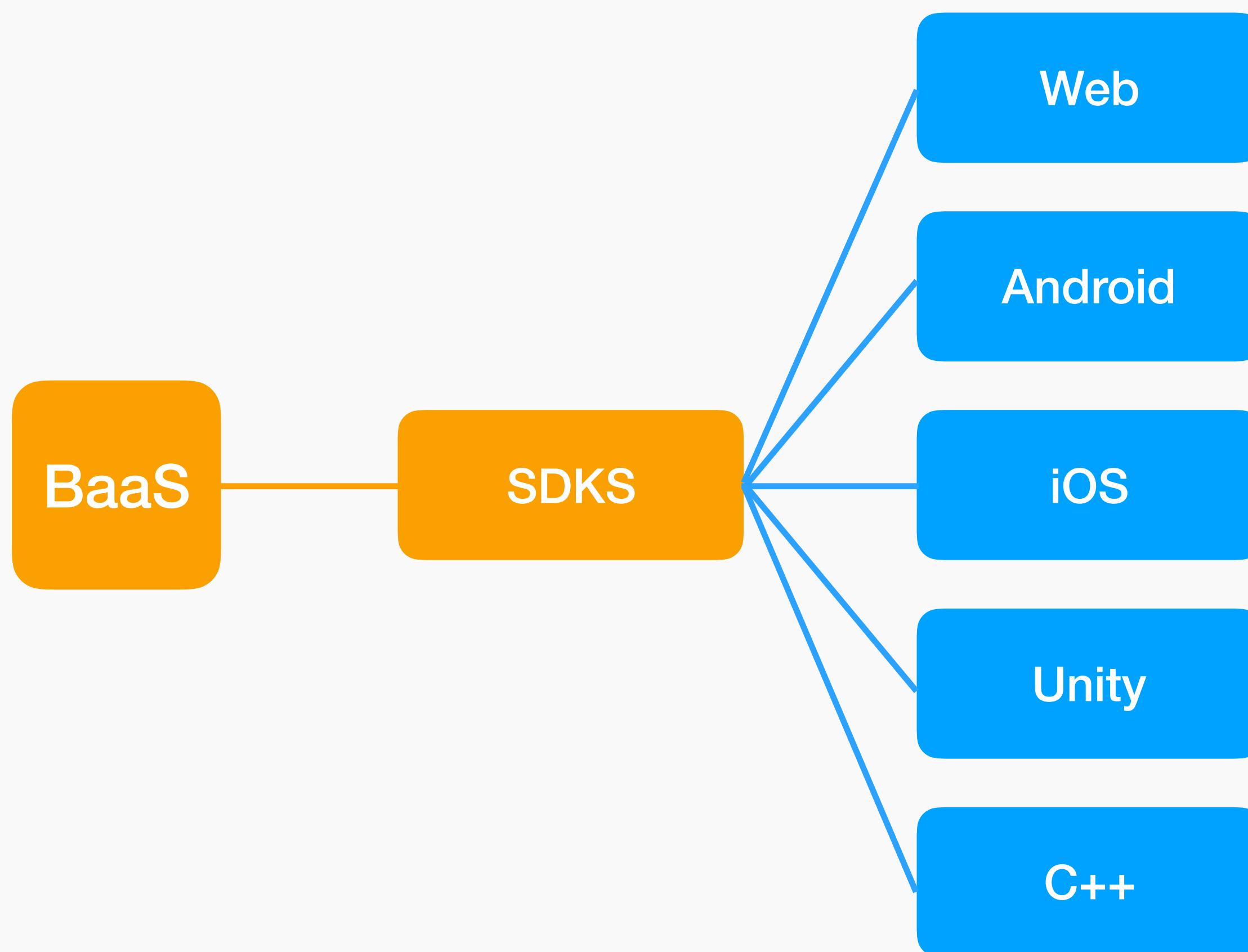
# What is it?



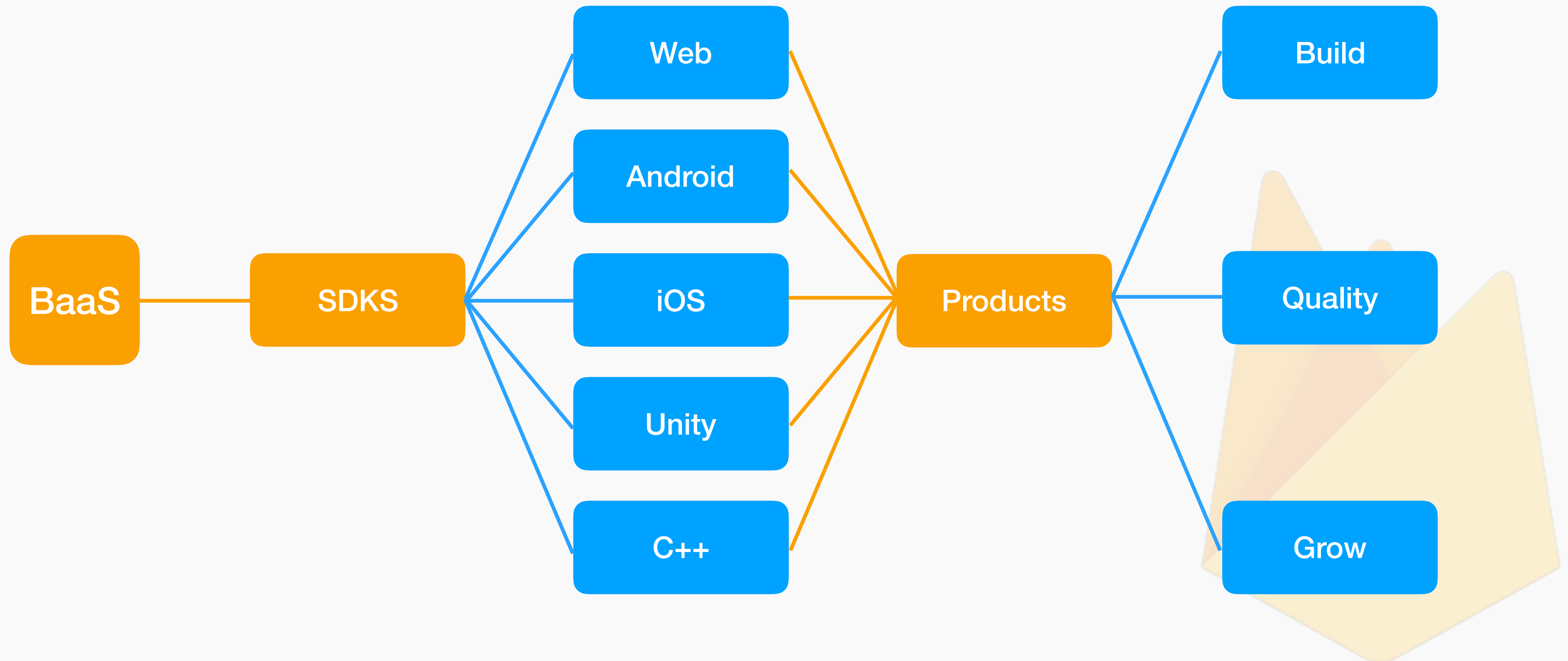
# What is it?

 BaaS

# What is it?



# What is it?



# Martian Restaurant Use-Case



# Martian Restaurant Use-Case



Validate Beta  
Features





# Martian Restaurant Use-Case



Validate Beta  
Features

Git-Branch  
Tutorial



# Martian Restaurant Use-Case



Validate Beta  
Features

Git-Branch  
Tutorial

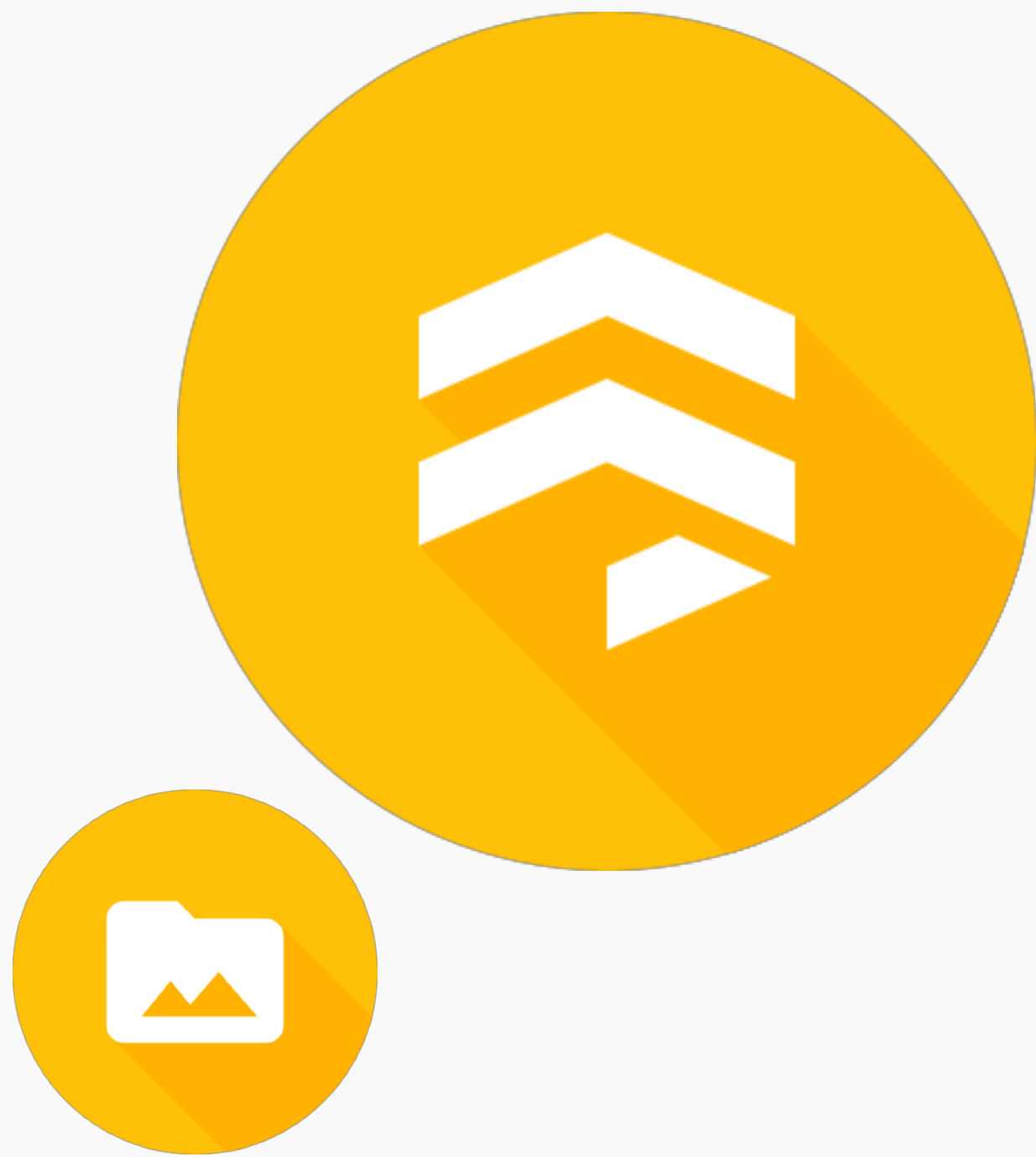
Medium Posts  
(Gitbook?)



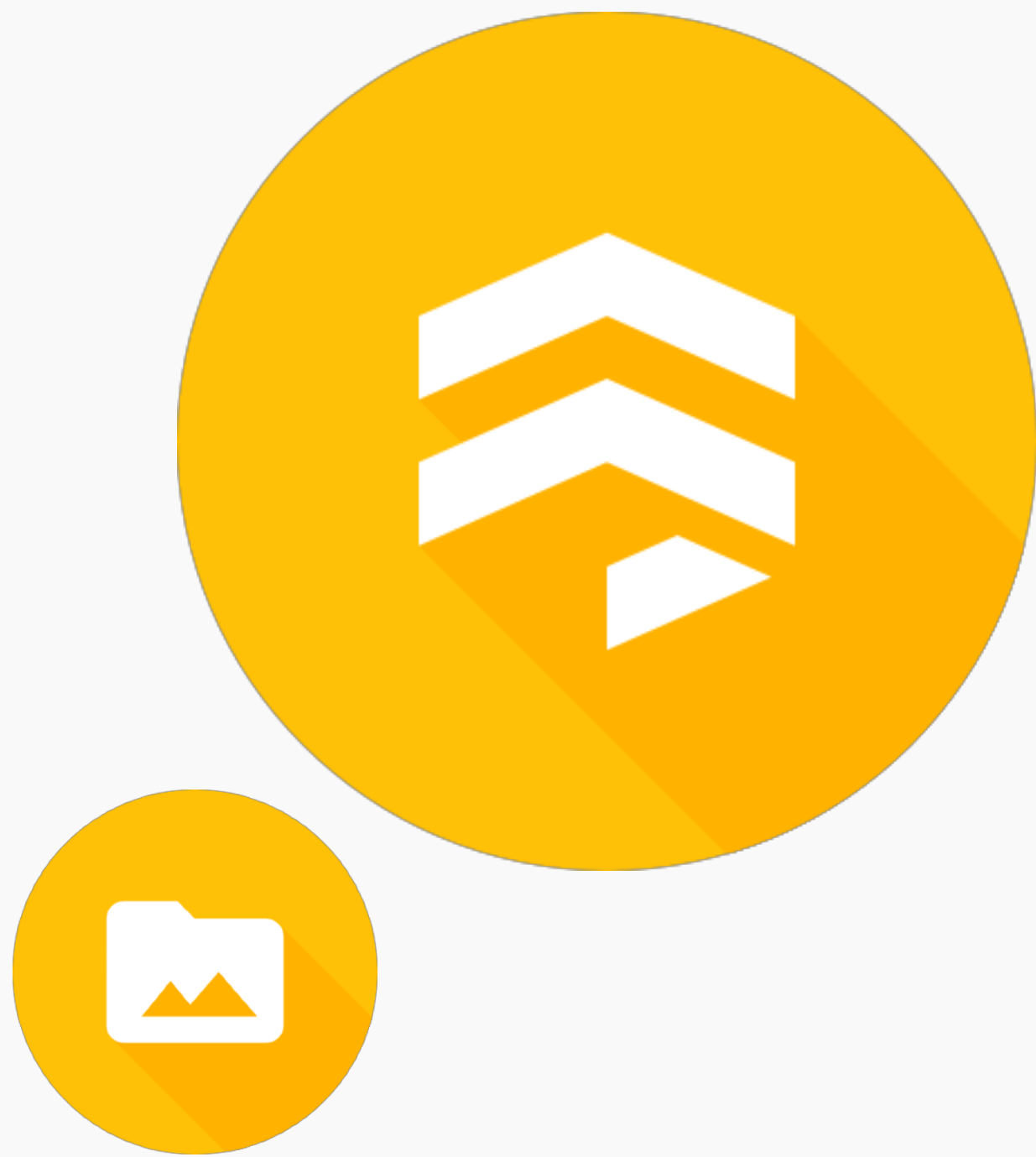
# Beta Features



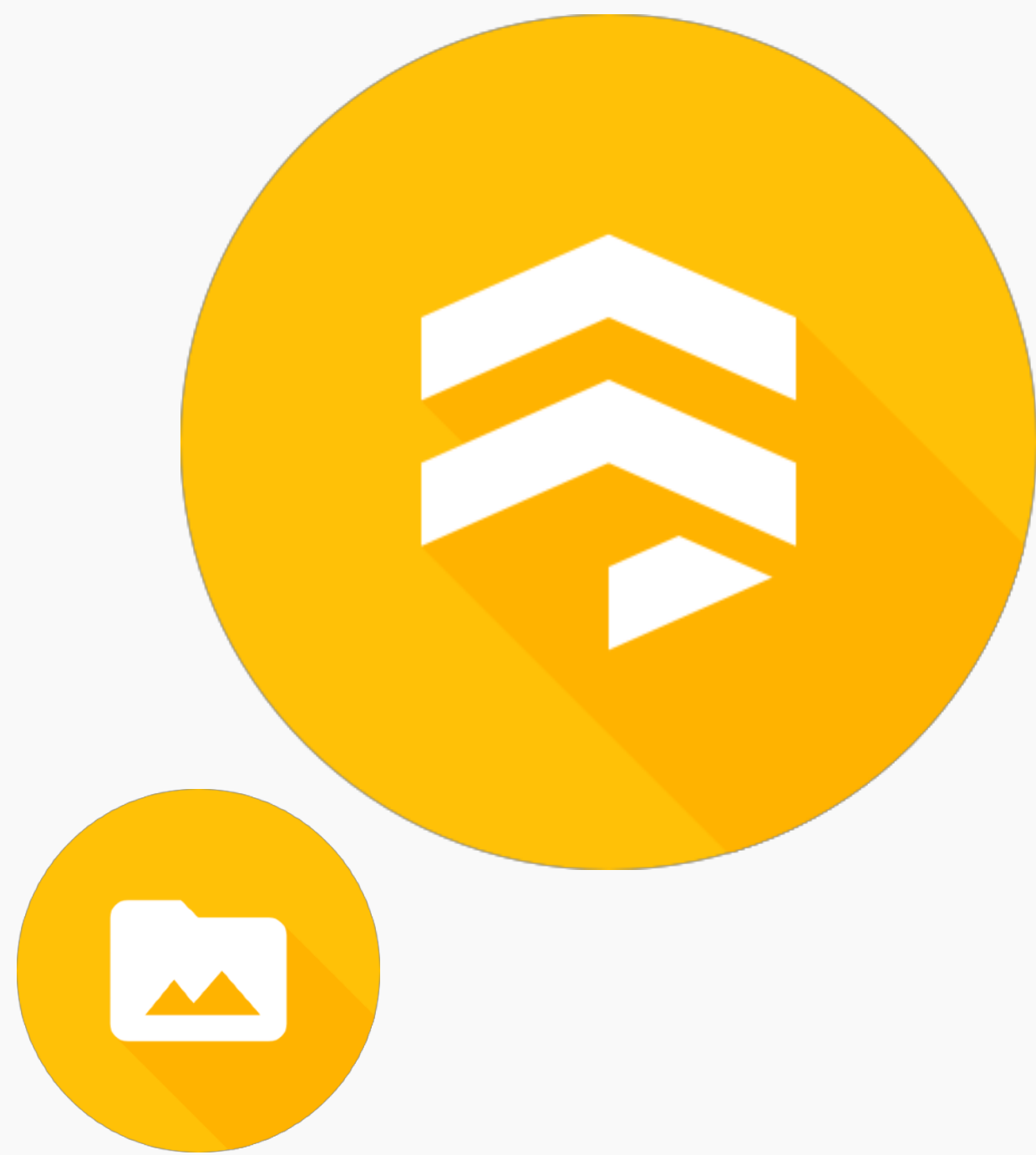
# Beta Features



# Beta Features

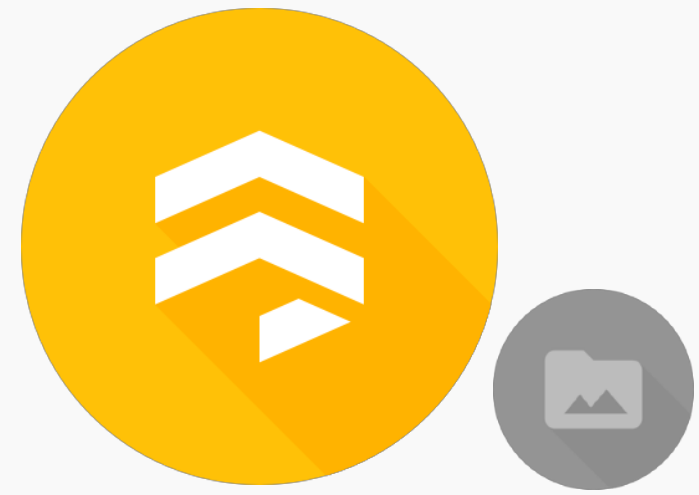


# Beta Features



# Beta Features



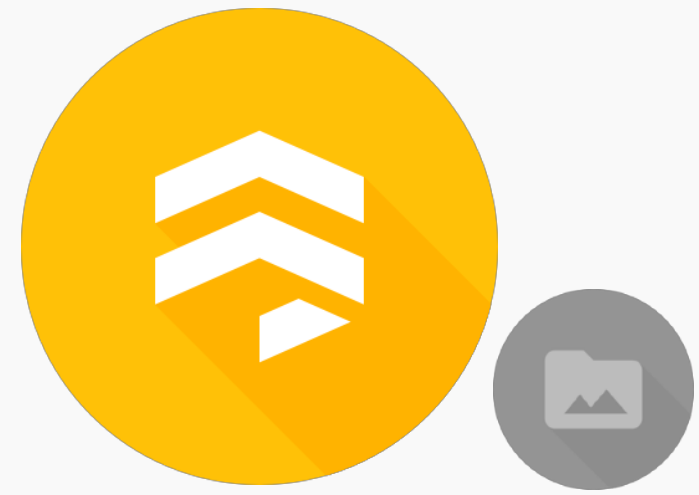


## Swift Implementation

# Firestore







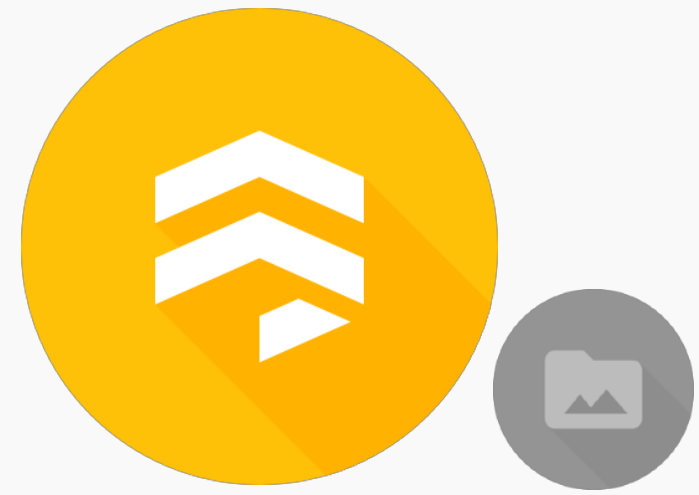
## Swift Implementation

# Firestore

Documents &  
Collections

Shallow queries



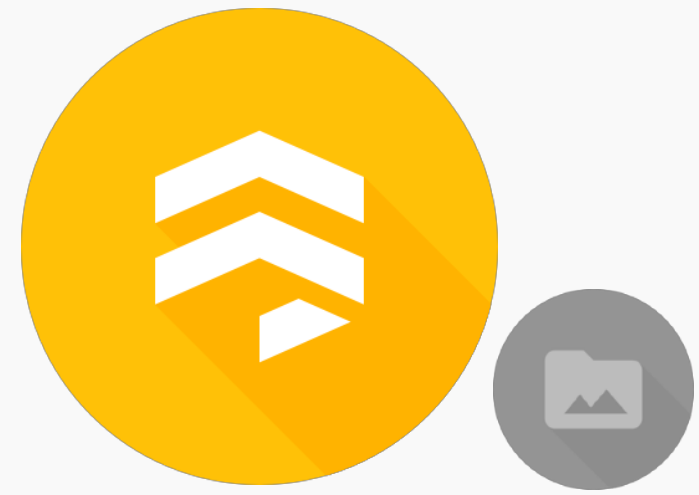


## Swift Implementation

# Firestore

```
service cloud.firestore {  
  match /databases/{database}/documents {  
  
    // Match any document in the 'cities' collection  
    match /cities/{city} {  
      allow read: if <condition>;  
      allow write: if <condition>;  
    }  
  }  
}
```



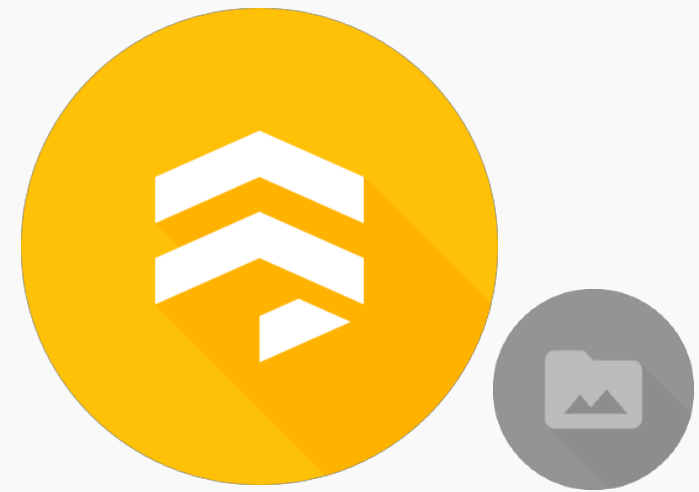


# Firestore

## Swift Implementation

```
private let db: Firestore = {  
    let db = Firestore.firestore()  
    let settings = db.settings  
    settings.areTimestampsInSnapshotsEnabled = true  
    settings.isPersistenceEnabled = false  
    db.settings = settings  
  
    return db  
}()
```





# Firestore

## Swift Implementation

```
db.collection(V.mealsPath).addSnapshotListener { [unowned self] (snapshot, error) in
    guard let snapshot = snapshot else {
        self.delegate?.mealsManager(self, failedUpdateWithErrorDescription: error!.localizedDescription)
        return
    }

    do {
        try self.notificateUpdate(withSnapshot: snapshot)
    } catch {
        //Now the App UI will probably enter in an inconsistent state, when comparing to the Firestore contents
        self.delegate?.mealsManager(self, failedUpdateWithErrorDescription: error.localizedDescription)
    }
}
```

```
private func notificateUpdate(withSnapshot snapshot: QuerySnapshot) throws {  
    let removedDocuments = snapshot.documentChanges.filter { $0.type == .removed }  
    let removedMealsIndexes = try mealsIndexes(withRemovedDocuments: removedDocuments)  
  
    let modifiedDocuments = snapshot.documentChanges.filter { $0.type == .modified }  
    let modifiedMealsIndexes = try mealsIndexes(withModifiedDocuments: modifiedDocuments)  
  
    let addedDocuments = snapshot.documentChanges.filter { $0.type == .added }  
    let addedMealsIndexes = try mealsIndexes(withAddedDocuments: addedDocuments)  
  
    self.delegate?.mealsManager(self,  
                                updatedStateWithAddedMeals: addedMealsIndexes,  
                                modifiedMeals: modifiedMealsIndexes,  
                                removedMeals: removedMealsIndexes)  
}
```



## Swift Implementation

# Storage







Swift Implementation

# Storage

AWS S3





# Storage

## Swift Implementation

```
private let mealsStorage: StorageReference = {  
    let storage = Storage.storage()  
    let mealsRef = storage.reference(withPath: V.mealsPath)  
  
    return mealsRef  
}()
```



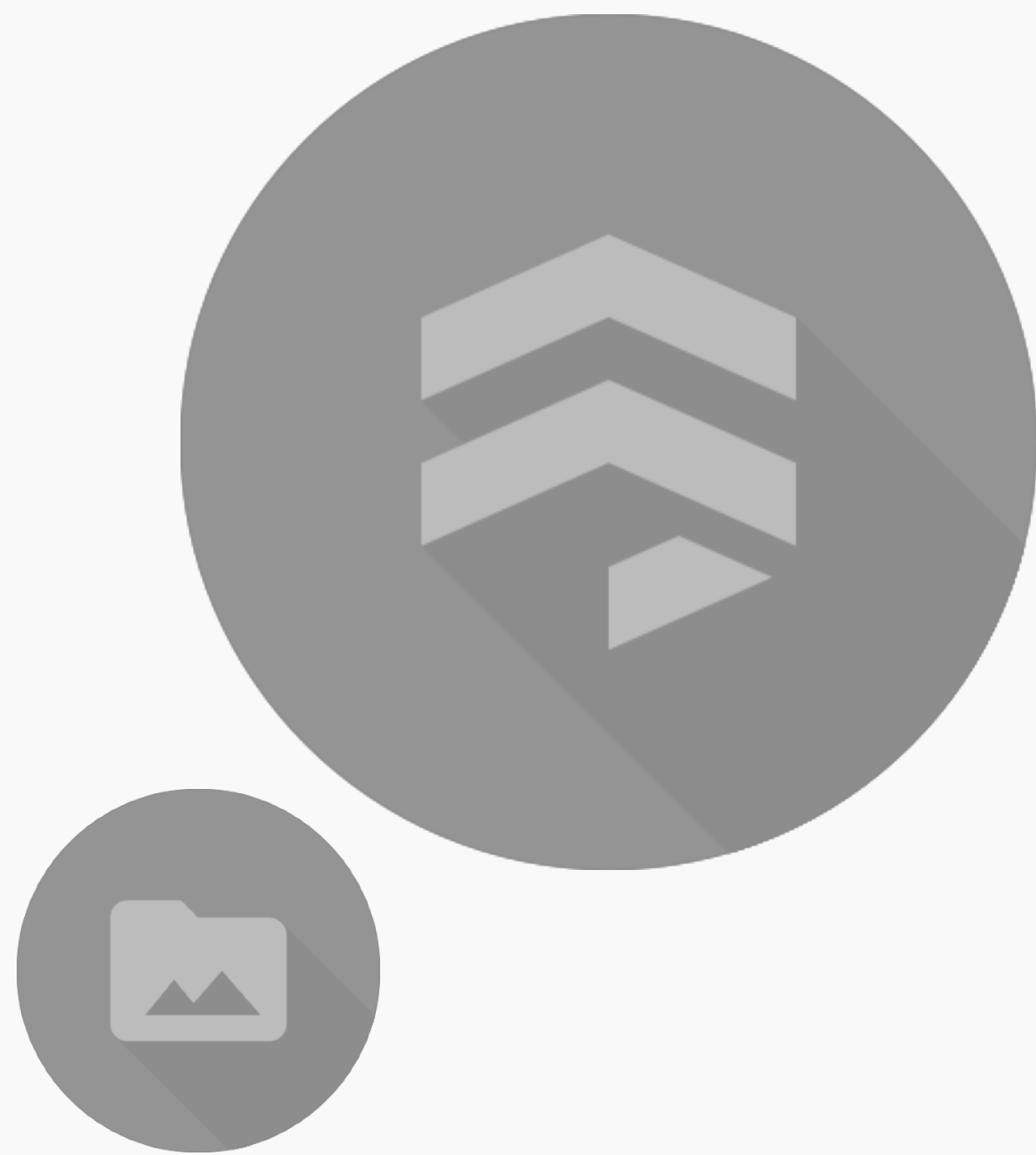


# Storage

## Swift Implementation

```
func getMealData(withRef ref: String, completion: @escaping (Data?, Error?) -> Void) {  
    mealsStorage.child(ref).getData(maxSize: 1 * 1024 * 1024) { data, error in  
        if let error = error {  
            completion(nil, error)  
        } else {  
            completion(data, nil)  
        }  
    }  
}
```

# Beta Features





# In App Messaging (IAM)

## Setup

1

Estilo e conteúdo

Layout da mensagem

Modal

Somente imagem

Banner superior

Plano de fundo

#ffffff

Cor do texto

#000000

Título da mensagem

Suspicious Behavior

Corpo (opcional)

We don't agree with eating humans

URL da imagem (opcional)

https://upload.wikimedia.org/wikipedia/commons/f/

Texto do botão (opcional)

Texto do botão

Próxima

Visualização do dispositivo

Por meio dessa visualização, é possível ter uma ideia geral de como sua mensagem será exibida em um dispositivo móvel. Haverá variação na renderização real da mensagem dependendo do dispositivo. Faça o teste com um dispositivo real para resultados reais.

Smartphone Tablet






# In App Messaging (IAM)

## Setup

2
Segmentação

Nome da campanha ?

Suspicious Behavior

Descrição da campanha (opcional) ?

Segmente usuários que correspondam a **todos** os itens a seguir...

Aplicativo	iOS	com.gmatuella.MartianRestaurant	▼	e
------------	-----	---------------------------------	---	---

Esta campanha segmenta 5 usuário(s) do app (100% dos usuários em potencial). ?





# In App Messaging (IAM)

## Setup

3

Programação

Início ?

Programada ▼

01/12/2018

22:02

Fuso horário

(GMT-08:00) Horário Los Angeles ▼

Fim ?

Sem data d... ▼

Acionar esta mensagem no app sempre que **qualquer um** dos seguintes eventos ocorrer

Saiba mais

Suspicious\_Behavior ✕

Evento +

Limite de frequência por dispositivo ?

Uma vez por dispositivo para esta campanha ▼





# In App Messaging (IAM)

## Swift Implementation

```
//futurelly store meal into cell to navigate
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let labeledVisionCell = tableView.dequeueReusableCell(withIdentifier: V.labeledVisionCellID) as! LabeledVisionCell
    let labeledVisionInIndex = estimatedLabels[indexPath.row]

    labeledVisionCell.contentLabel.text = labeledVisionInIndex.formatted

    mealsManager.fetchMeal(named: labeledVisionInIndex.name) { meal in
        DispatchQueue.main.async { labeledVisionCell.isAvailable = meal != nil }
    }

    analytics.checkForSuspiciousBehavior(withVisionLabel: labeledVisionInIndex.name)

    return labeledVisionCell
}
```





# In App Messaging (IAM)

## Swift Implementation

```
private enum V {
    static let suspiciousLabels = ["skin", "flesh", "hand", "nail", "selfie", "hair"]

    static let suspiciousBehaviorID = "Suspicious_Behavior"
}

final class MartianAnalytics {

    func checkForSuspiciousBehavior(withVisionLabel visionLabel: String) {
        if V.suspiciousLabels.contains(visionLabel.lowercased()) {
            Analytics.logEvent(V.suspiciousBehaviorID, parameters: nil)
        }
    }
}
```



# In App Messaging (IAM)

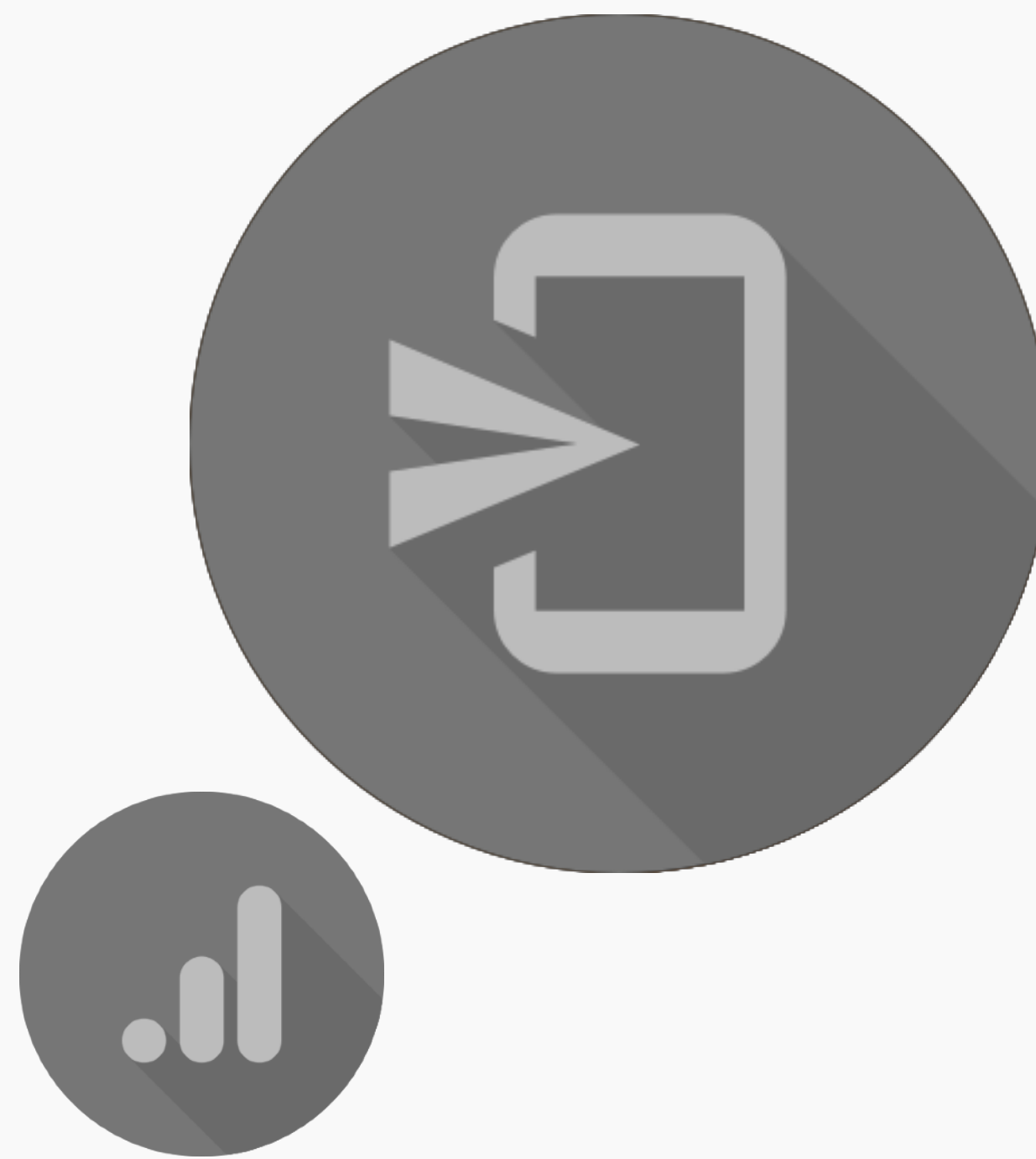
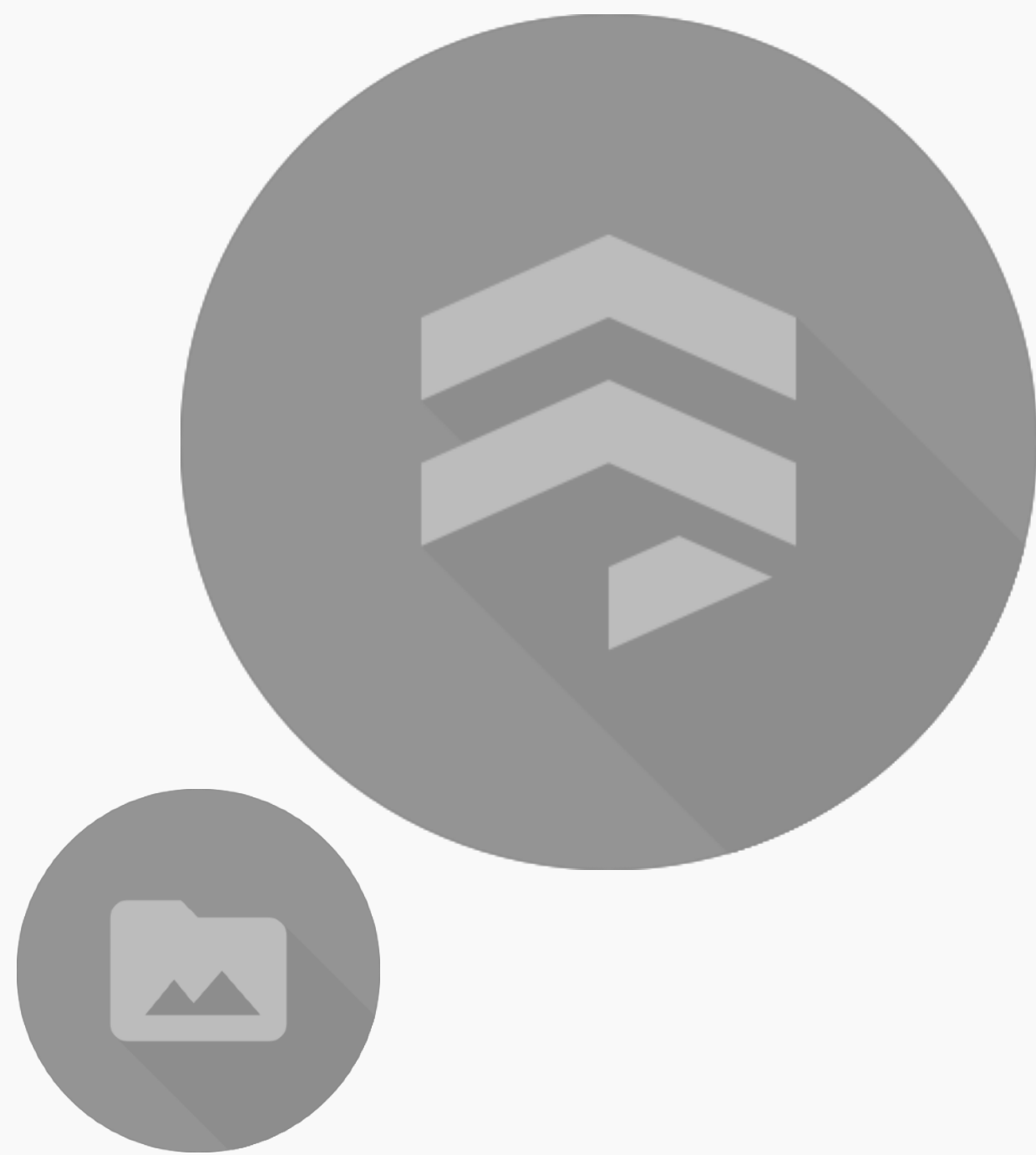
Swift Implementation

Simple (shallow)





# Beta Features





# Machine Learning Kit

## Swift Implementation

```
private let detector: VisionLabelDetector = {  
    let options = VisionLabelDetectorOptions(confidenceThreshold: Consts.detectorConfidence)  
    let vision = Vision.vision()  
  
    return vision.labelDetector(options: options)  
}()
```



# Machine Learning Kit

## Swift Implementation

```
func estimate(image: UIImage, completion: @escaping AppraisalHandler) {
    detector.detect(in: VisionImage(image: image)) { labels, error in
        guard error == nil, let labels = labels, !labels.isEmpty else {
            completion(nil, "Sorry, couldn't find anything with enough confidence 😞")
            return
        }

        let appraisals = labels.map { LabeledVision(name: $0.label, _confidence: $0.confidence) }
        completion(appraisals, nil)
    }
}
```



# Machine Learning Kit

Swift Implementation

Local Validation = meh

Cloud (expensive)

Tensorflow?



DEMO  

[github.com/matuella/MartianRestaurant](https://github.com/matuella/MartianRestaurant)



# Tools - When and why



# Tools - When and why

SQL vs NoSQL  
databases

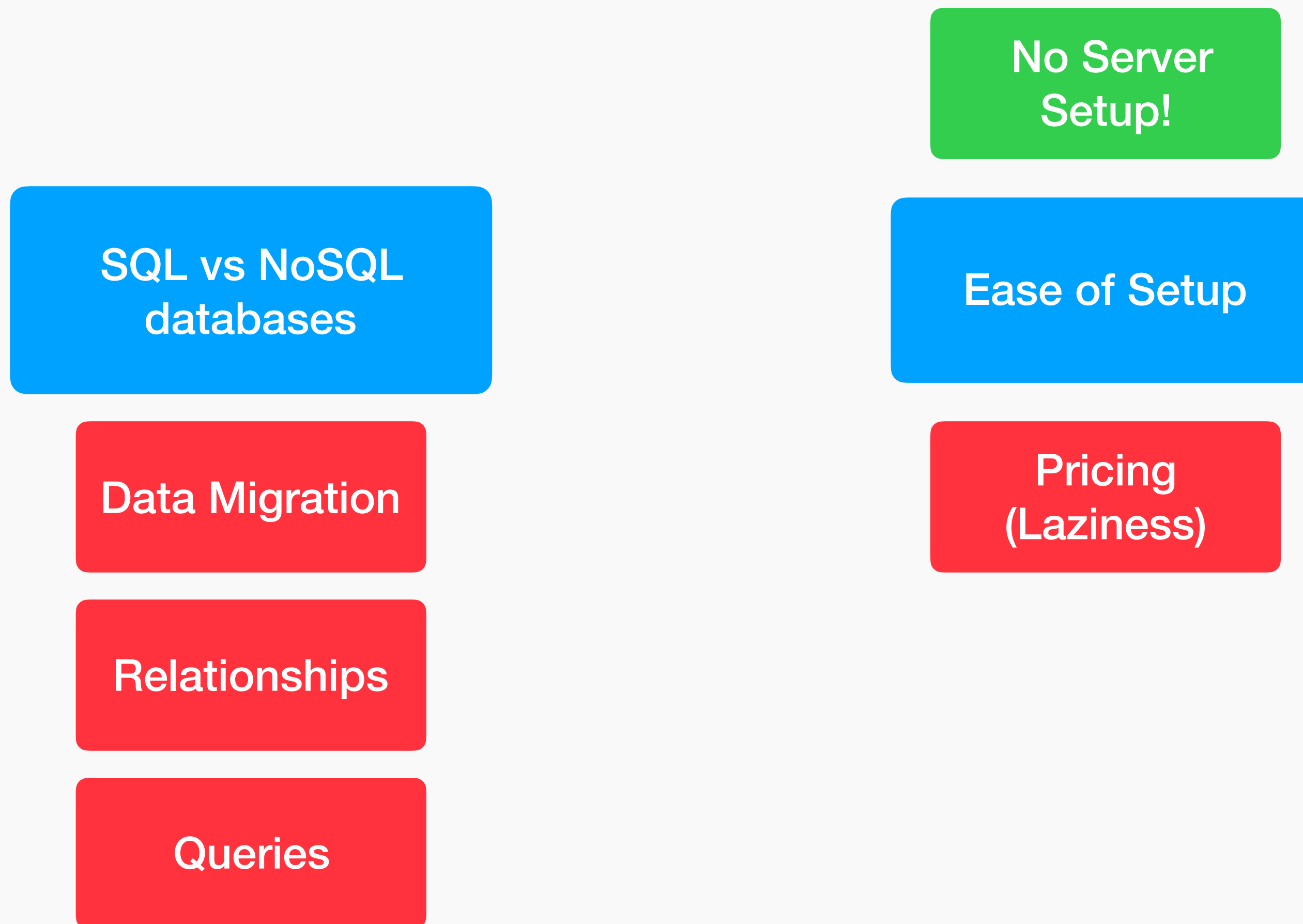
Data Migration

Relationships

Queries



# Tools - When and why





# Firestore - When and why



# Firestore - When and why



Real Time Sync

MVPs

Realtime Notifications,  
Messaging, Chat



# Firebase - When and why



Real Time Sync

MVPs

Realtime Notifications,  
Messaging, Chat

Big Query

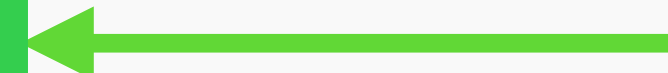
BaaS (Mobile first)



+ Complex Analytics

A little + complex  
Backend

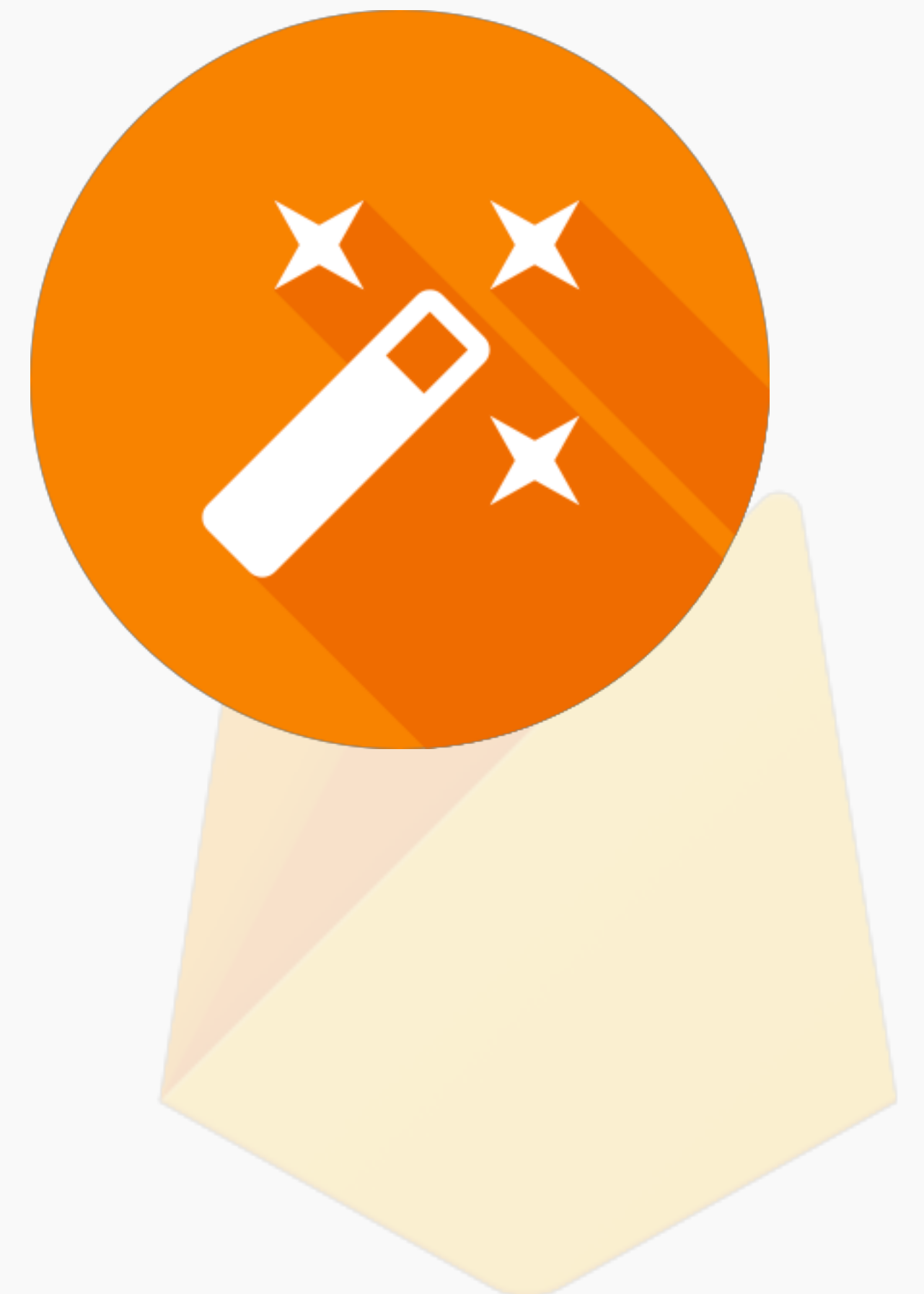
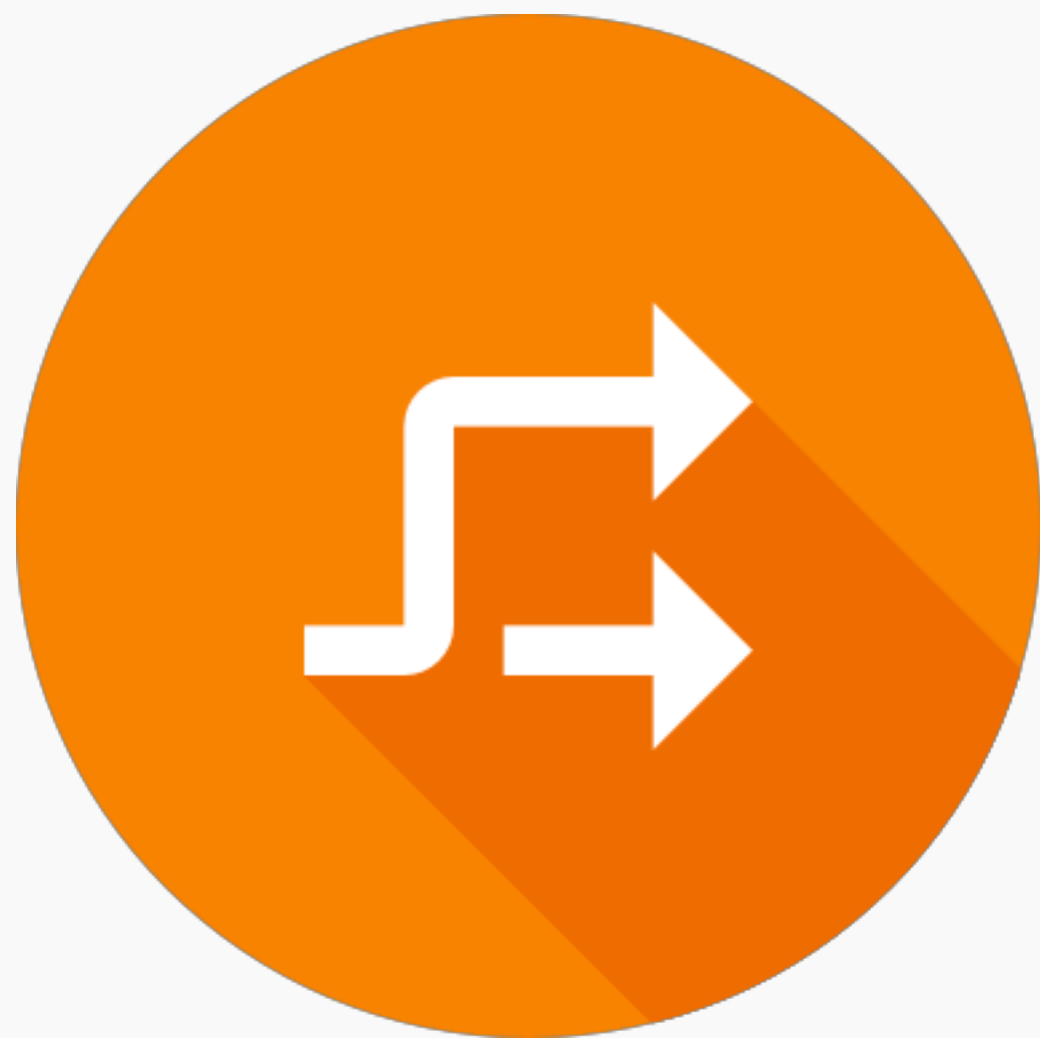
Server-side  
Integrations



# Coming Next



# Coming Next



# Summing up



# Summing up

- What is Firebase;



# Summing up

- What is Firebase;
- Martian Restaurant Use Case;





# Summing up

- What is Firebase;
- Martian Restaurant Use Case;
- Use Case Products explanation + Swift SDK implementation;



# Summing up

- What is Firebase;
- Martian Restaurant Use Case;
- Use Case Products explanation + Swift SDK implementation;
- Martian Restaurant Demo;



# Summing up

- What is Firebase;
- Martian Restaurant Use Case;
- Use Case Products explanation + Swift SDK implementation;
- Martian Restaurant Demo;
- Firebase feedback.



# Thank you!

## Questions?

