



SiriKit iOS 13

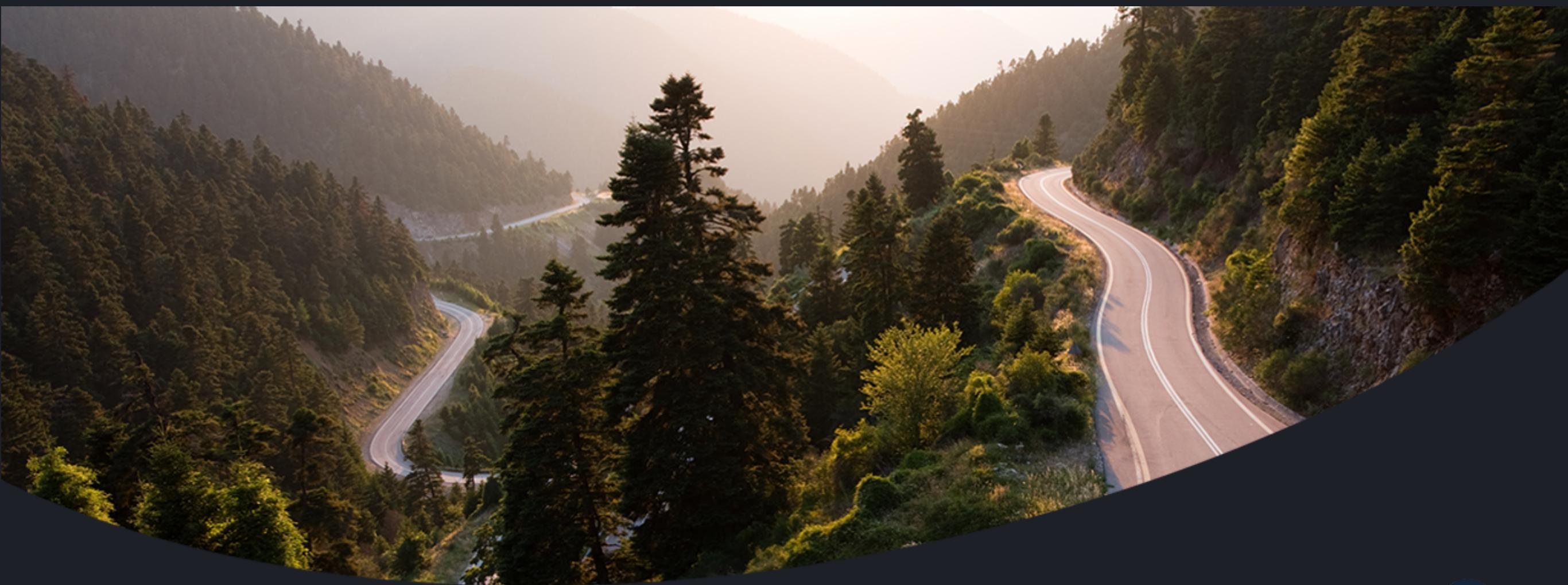
Coyote Lab – 27 Septembre 2019
Stephan Yannick





1. Intents app extension and UI
2. Intent definition
3. Sample application with Intent and UI extension

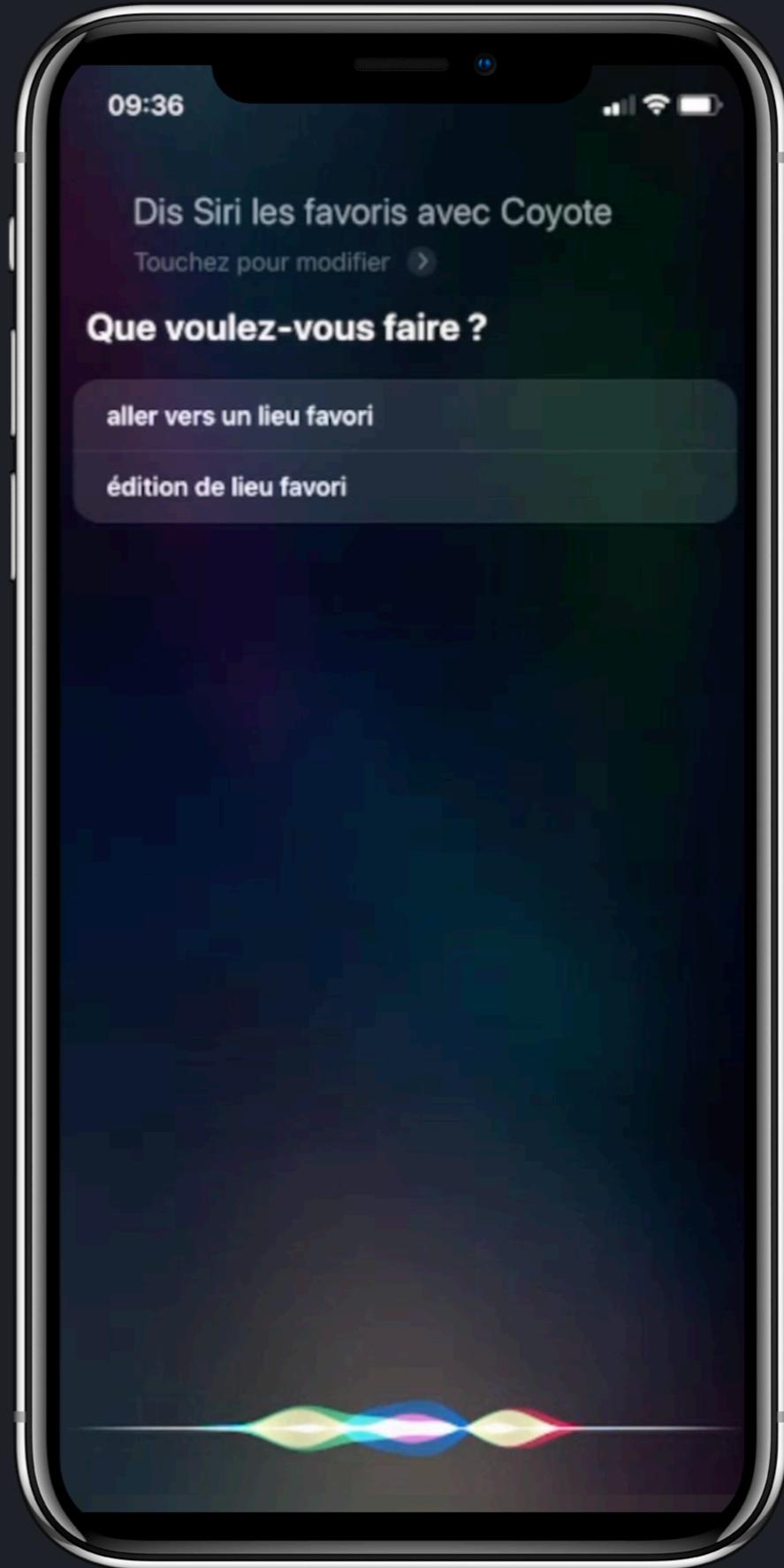




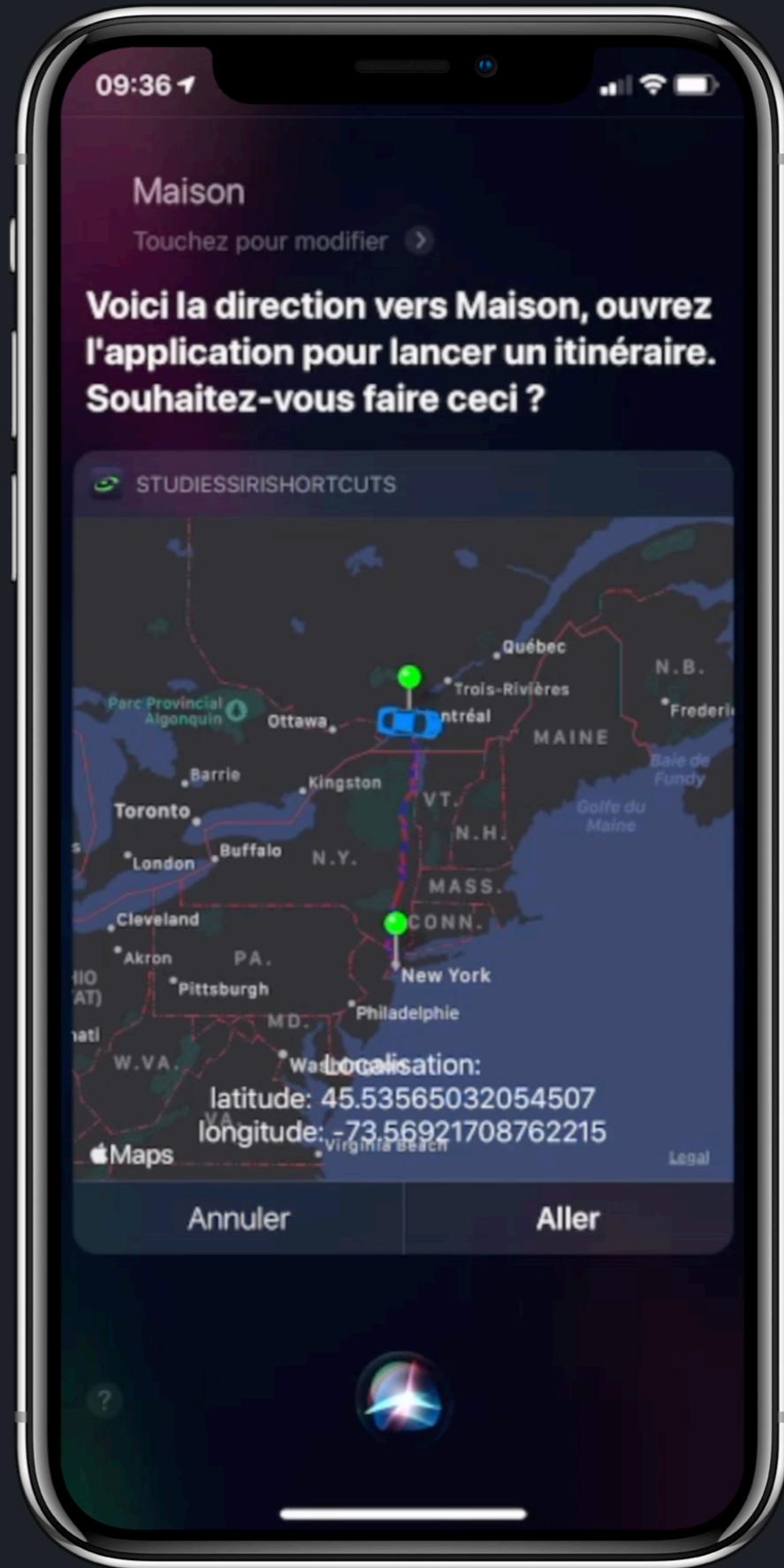
Intents app extension
Intents UI app extension
Domains and Intents



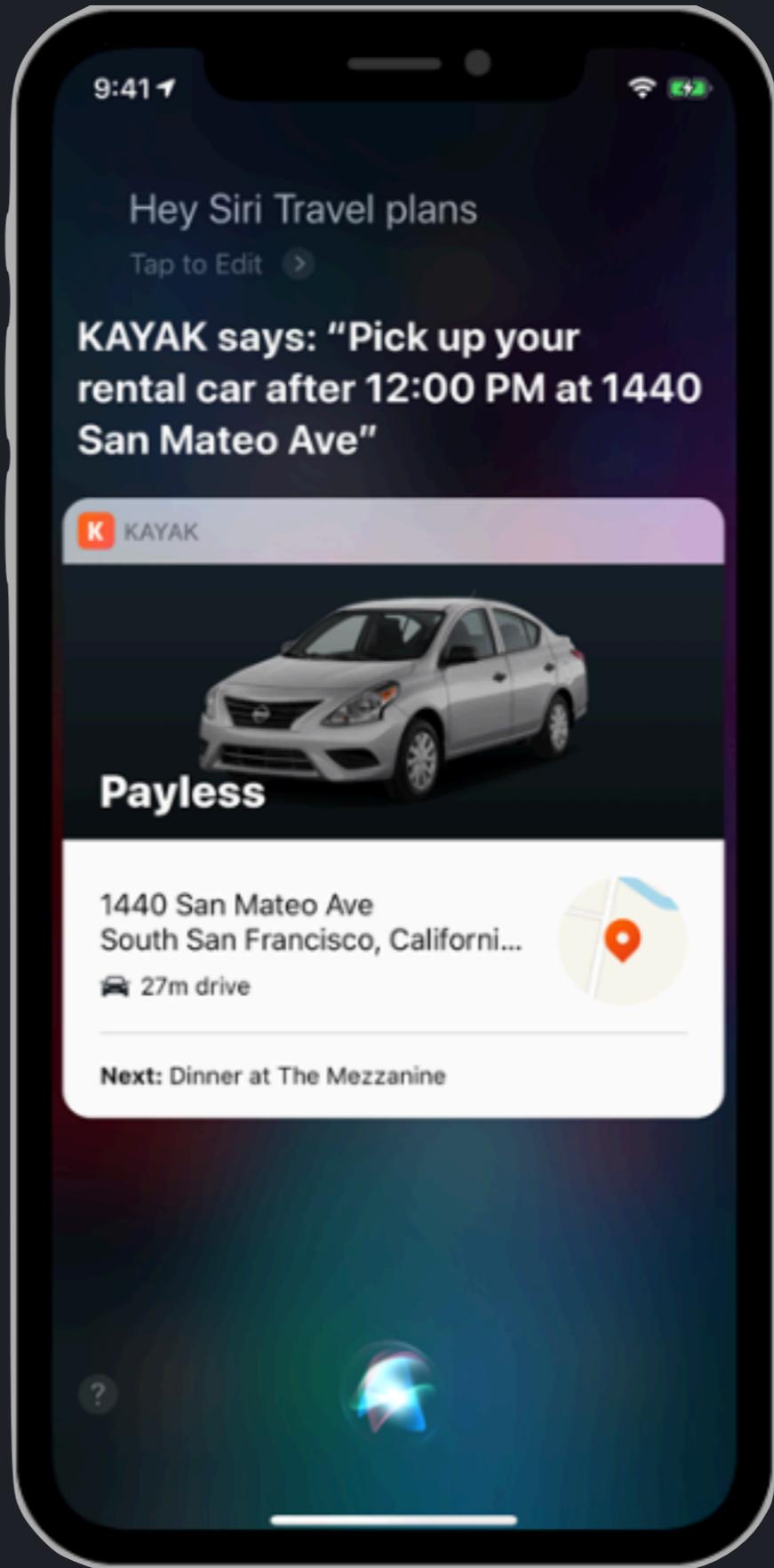
Intents app extension



Intents UI app extension



Domains and Intents



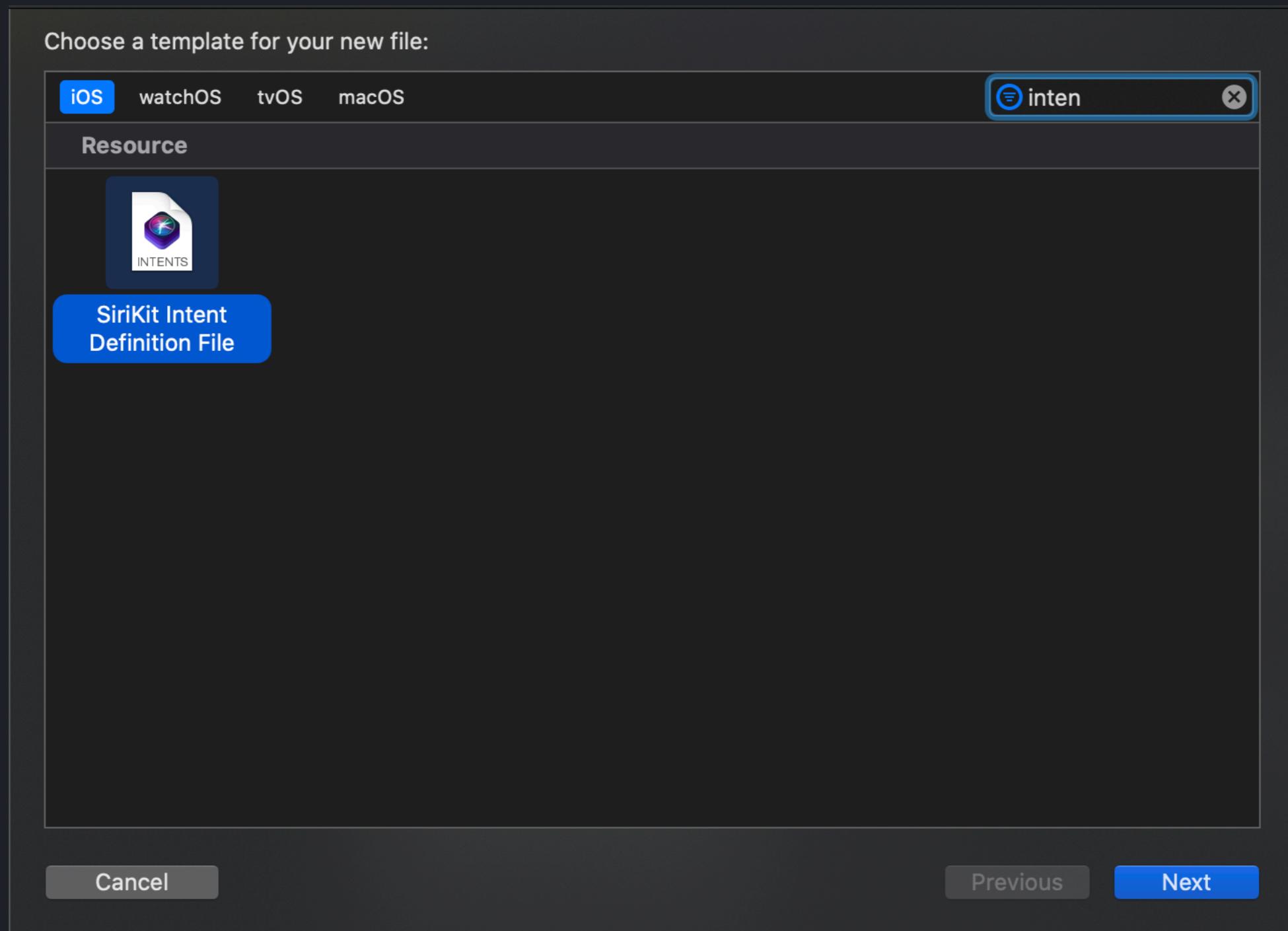
 COYOTE



Intent Definition



Intent Definition



Intent Definition

StudiesSiriShortcuts > IntentSiriKit > Modules > Intents > Intents.intentdefinition > Custom Intents > GoToFavorite

CUSTOM INTENTS

- I GoToFavorite
- R Response
- I RequestAlert
- R Response

ENUMS

- E DirectionType

TYPES

- T Favorite
- T Alert

Custom Intent

Category: Go

Title: Aller vers un lieu favori

Description: Direction vers lieu un favori avec la Map

Default Image: iconLocation

Confirmation: User confirmation required

Parameters

Parameter	Display Name	Type	Array	User-facing	Dynamic Options
T favorite	Lieu favori	favori	<input type="checkbox"/> Supports multiple values	<input checked="" type="checkbox"/> User can supply value in Siri and Shortcuts app	<input type="checkbox"/> Valid values are provided dynamically

Relationship

Parent Parameter: None

Siri Dialog

Prompt: Vers quel lieu favori ?

Disambiguation: Customize disambiguation dialog

Disambiguation Introduction: Vous avez `count` options de disponible.

Disambiguation Prompt: Vers quel lieu favori voulez-vous aller ?

Parameter Confirmation:

Validation Errors (0)

Code	Template

Type

CUSTOM INTENTS

- I GoToFavorite
- R Response
- I RequestAlert
- R Response

ENUMS

- E DirectionType

TYPES

- T Favorite
- T Alert

Type

Display Name favori

Properties

Property	Display Name	Type	Array	Supports multiple values
S identifier	localisation	Location		
S displayString				
L location				

+ -



Enum

CUSTOM INTENTS

- I GoToFavorite
- R Response
- I RequestAlert
- R Response

ENUMS

- E DirectionType

TYPES

- F Favorite
- A Alert

▼ Enum

Display Name	Direction
unknown	
same	
opposite	

▼ Cases

Case
unknown
same
opposite

Index 1

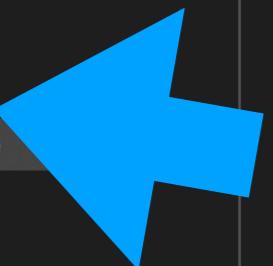
Display Name mon sens

Pronunciation Hint mon sens

▼ Alternative Speakable Matches (5)

Speakable Match	Pronunciation Hint
Dans mon sens	Dans mon sens
Dans notre sens	Dans notre sens
Dans le sens direct	Dans le sens direct
De mon côté	De mon côté
De notre côté	De notre côté
+ -	

+ -



Custom Intents

CUSTOM INTENTS

- I GoToFavorite
- R Response
- I RequestAlert
- R Response

ENUMS

- E DirectionType

TYPES

- T Favorite
- T Alert

▼ Custom Intent

Category	Go
Title	Aller vers un lieu favori
Description	Direction vers lieu un favori avec la Map
Default Image	iconLocation
Confirmation	<input checked="" type="checkbox"/> User confirmation required

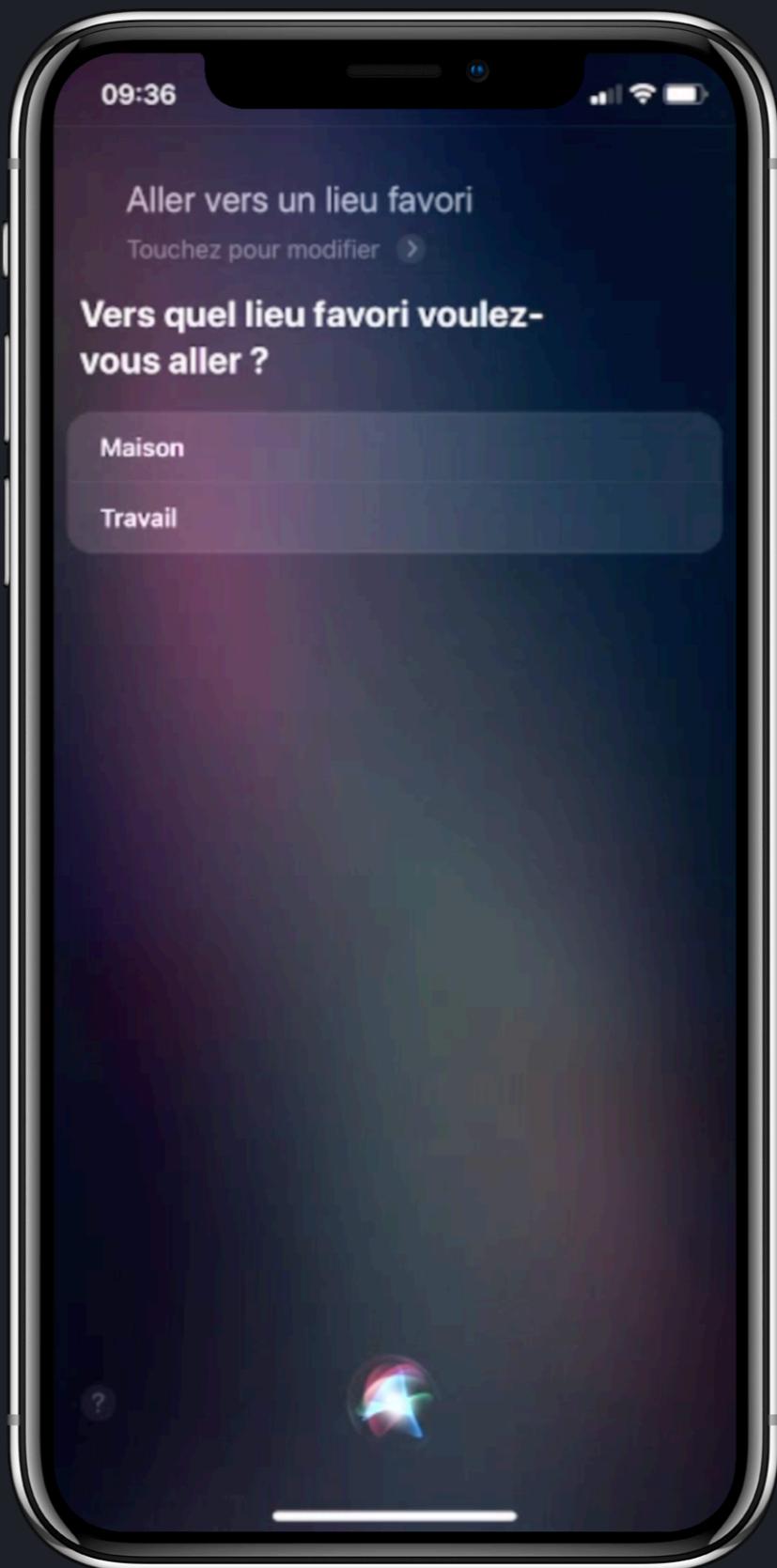
Custom Intent Parameters

▼ Parameters

Parameter					
T favorite	<div style="display: flex; justify-content: space-between;"><div style="flex: 1;"><p>Display Name <input type="text" value="Lieu favori"/></p><p>Type <input type="text" value="favori"/></p><p>Array <input checked="" type="checkbox"/> Supports multiple values</p><p>User-facing <input checked="" type="checkbox"/> User can supply value in Siri and Shortcuts app</p><p>Dynamic Options <input checked="" type="checkbox"/> Valid values are provided dynamically</p></div><div style="flex: 1; text-align: right;">▼ Relationship</div></div> <p>Parent Parameter <input type="text" value="None"/></p> <div style="margin-top: 10px;"><p>▼ Siri Dialog</p><p>Prompt <input type="text" value="Vers quel lieu favori ?"/></p><p>Disambiguation <input checked="" type="checkbox"/> Customize disambiguation dialog</p><p>Disambiguation Introduction <input type="text" value="Vous avez count options de disponible."/></p><p>Disambiguation Prompt <input type="text" value="Vers quel lieu favori voulez-vous aller ?"/></p><p>Parameter Confirmation <input type="text"/></p></div> <div style="margin-top: 10px;"><p>▼ Validation Errors (0)</p><table border="1" style="width: 100%; border-collapse: collapse;"><thead><tr><th style="width: 15%;">Code</th><th style="width: 85%;">Template</th></tr></thead><tbody><tr><td style="text-align: center; padding: 5px;">+</td><td style="text-align: center; padding: 5px;">-</td></tr></tbody></table></div>	Code	Template	+	-
Code	Template				
+	-				



Custom Intent Parameters



Custom Intent Shortcuts App

▼ Shortcuts app

Intent is user-configurable
Define how this intent appears in the Shortcuts app and Add to Siri.

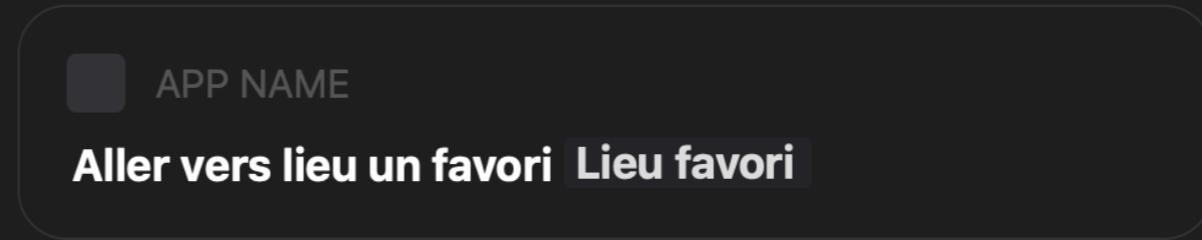
Input Parameter **None** 
If compatible, the output of the preceding action will be filled into this parameter.

Key Parameter **None** 
Defines which parameter to show suggestions for in the "Add Action" sheet.

Supported Combinations
favorite

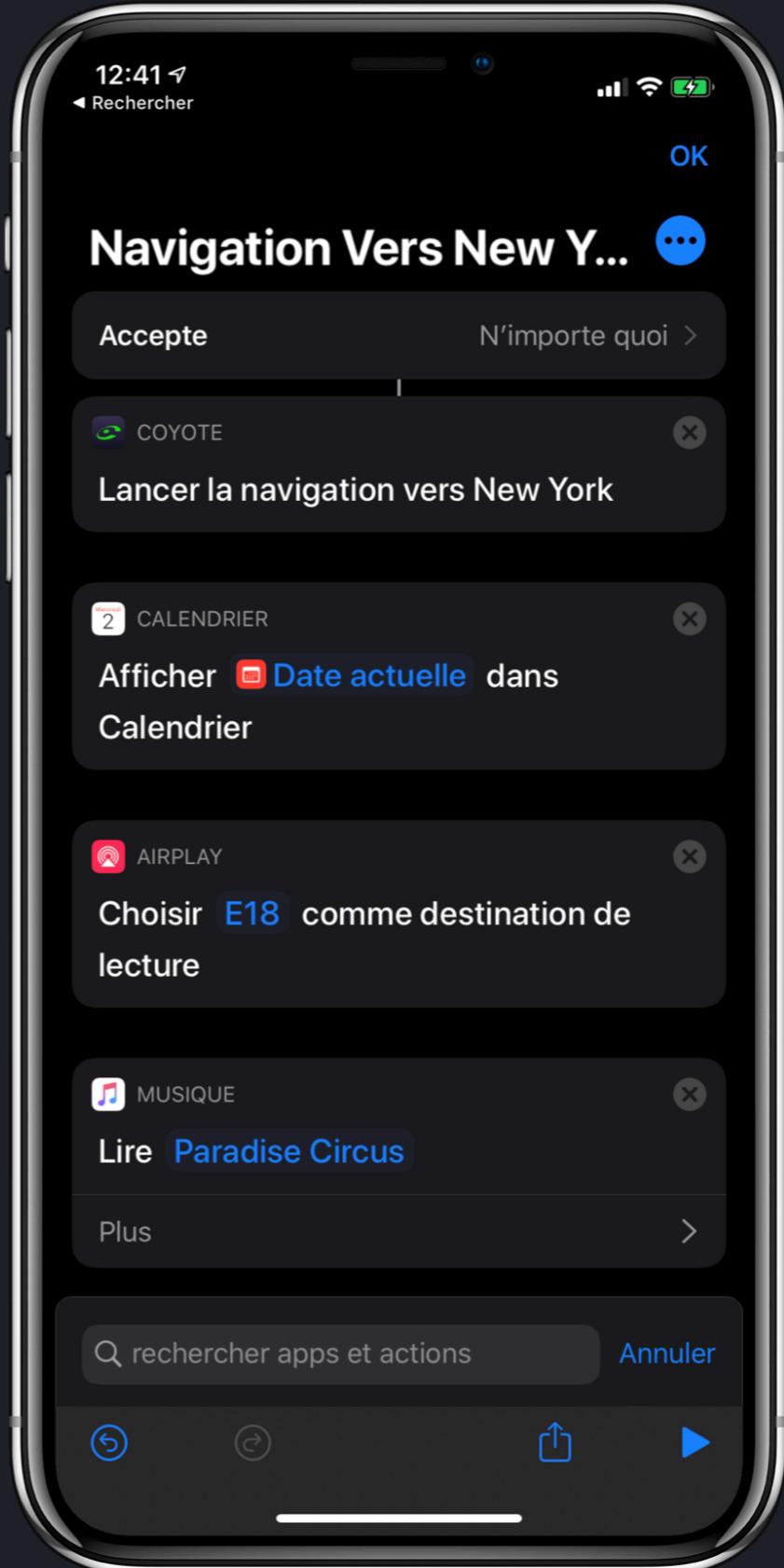
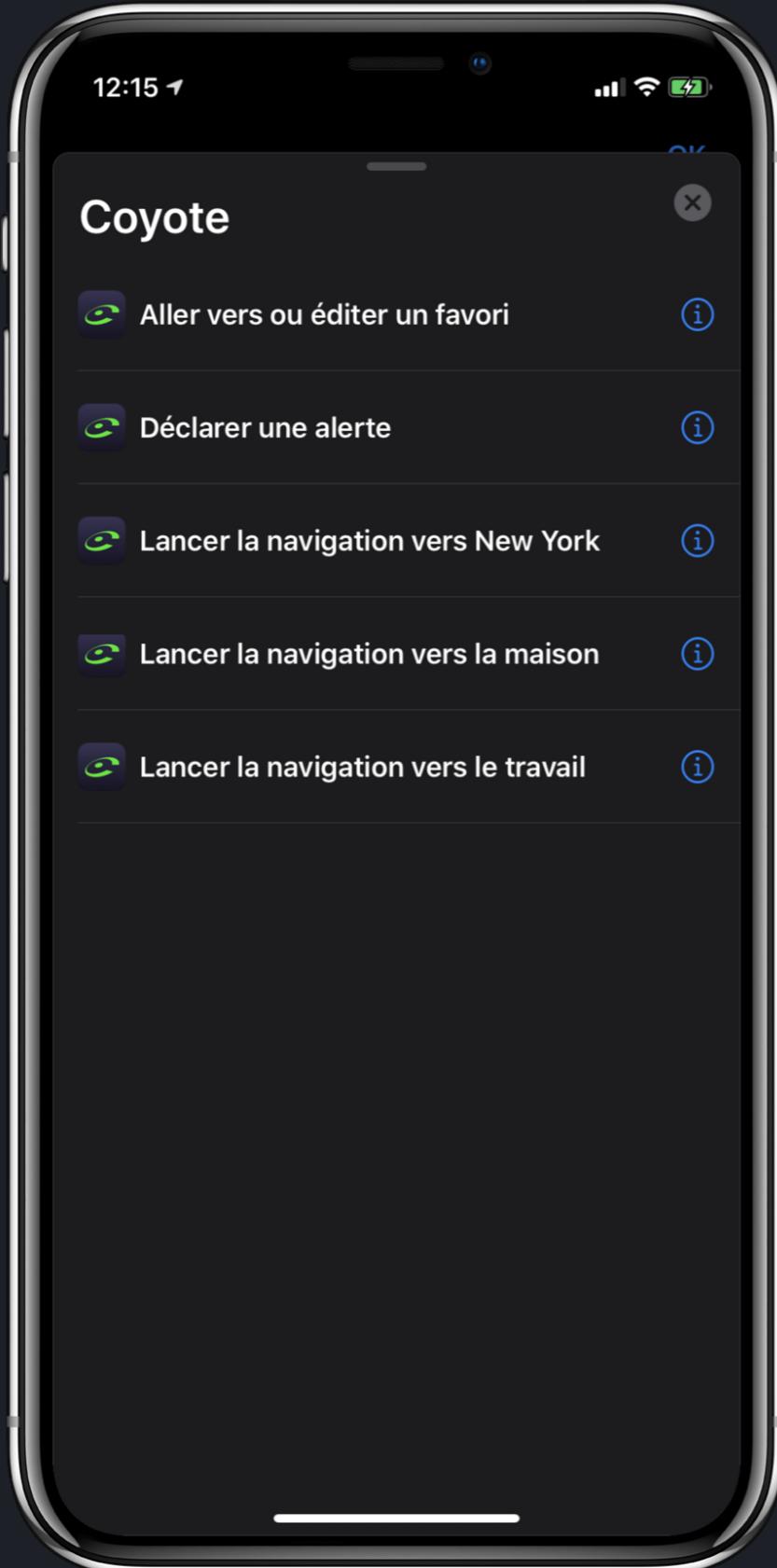
Summary **Aller vers lieu un favori favorite**
Parameters not included in the summary will be available to the user under "Show More".

Preview



APP NAME
Aller vers lieu un favori Lieu favori

Shortcuts App



COYOTE

Shortcuts App



COYOTE

Custom Intent Suggestions

▼ Suggestions

Intent is eligible for Siri Suggestions

Summary Aller vers lieu un favori favorite

Description Lancer la direction vers le lieu favori favorite

Background Supports background execution

Preview



Shortcuts



Custom Intent Response

CUSTOM INTENTS

- I GoToFavorite
- R Response
- I RequestAlert
- R Response

ENUMS

- E Direction Modified

TYPES

- T Favorite
- T Alert

Properties

Property
S failureMessageDisplay
S successLocationDisplay

Display Name: Failure Message Display
Type: String
Array: Supports multiple values

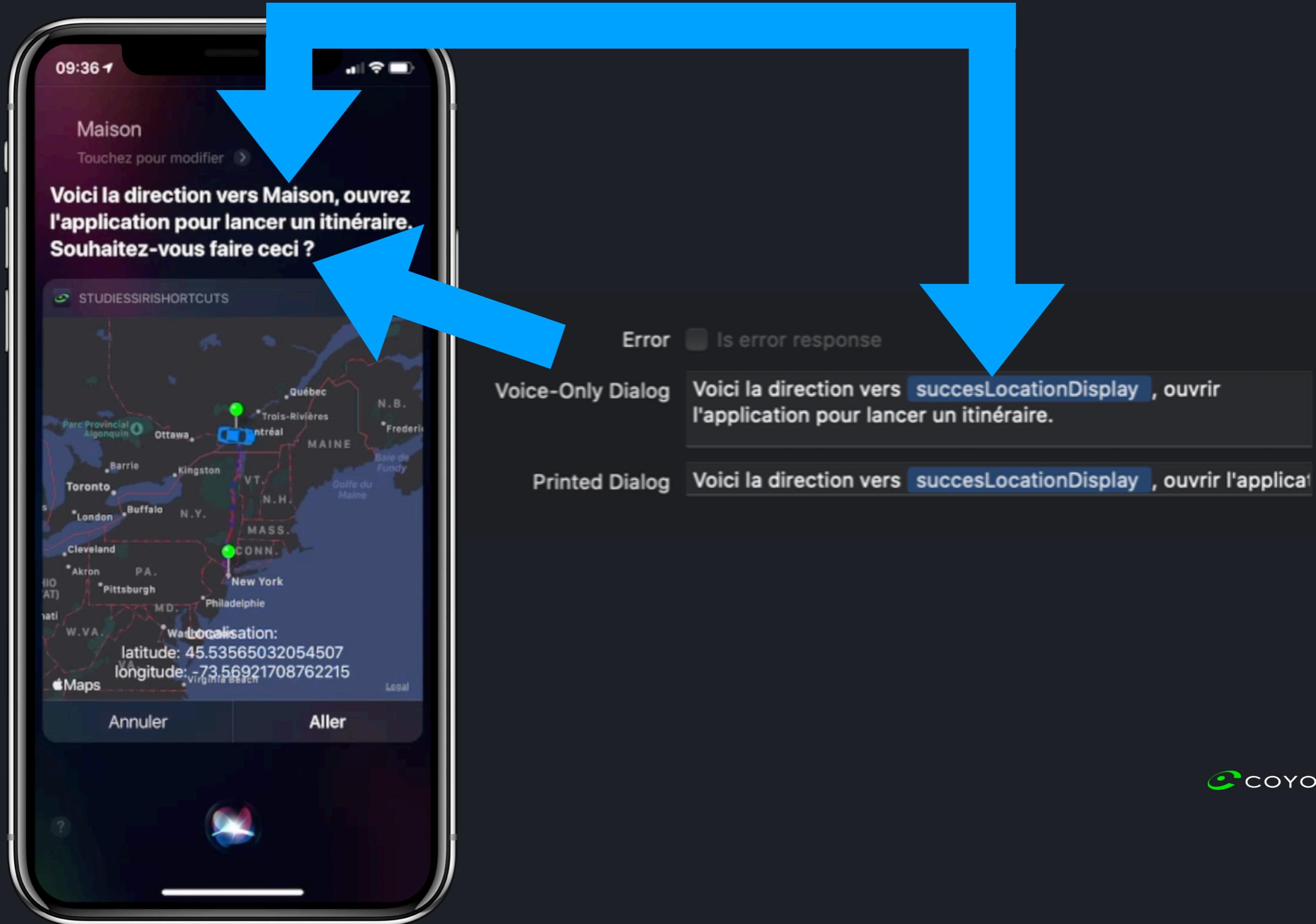
Output: None

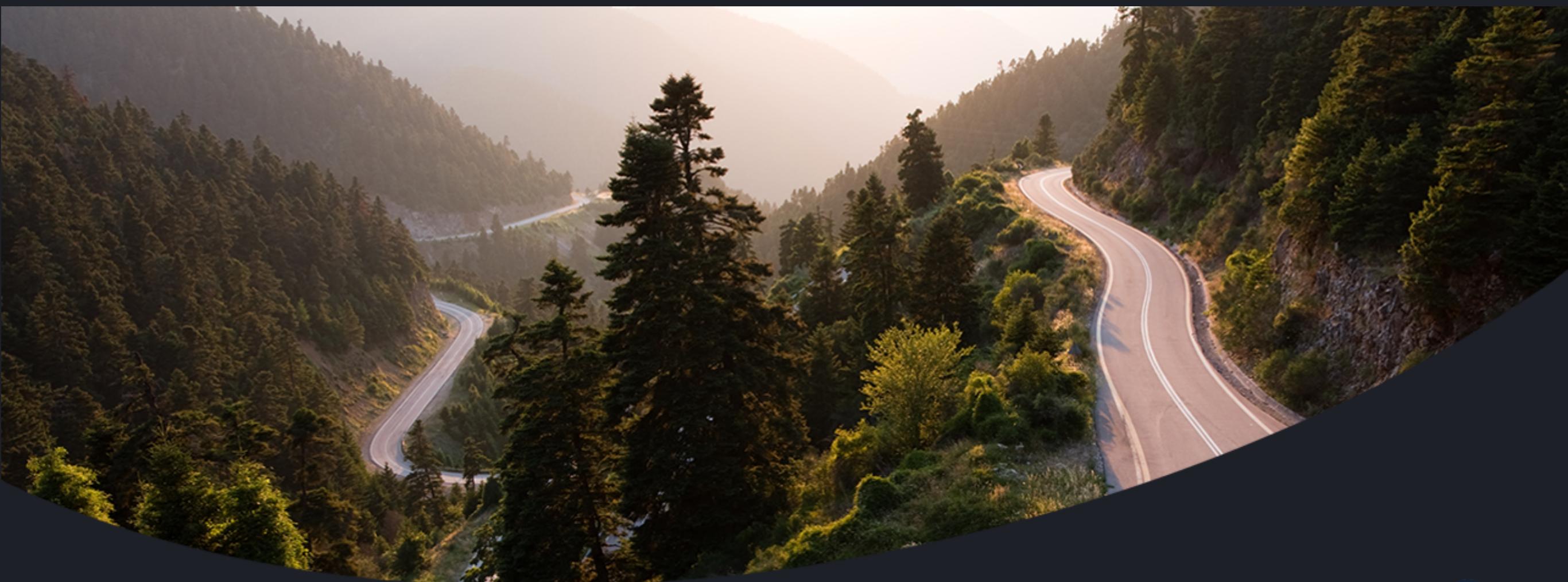
Response Templates

Code
success
failure
failureMapUserActivity

Error: Is error response
Voice-Only Dialog: Voici la direction vers `successLocationDisplay`, ouvrir l'application pour lancer un itinéraire.
Printed Dialog: Voici la direction vers `successLocationDisplay`, ouvrir l'application pour lancer un itinéraire.

Custom Intent Response



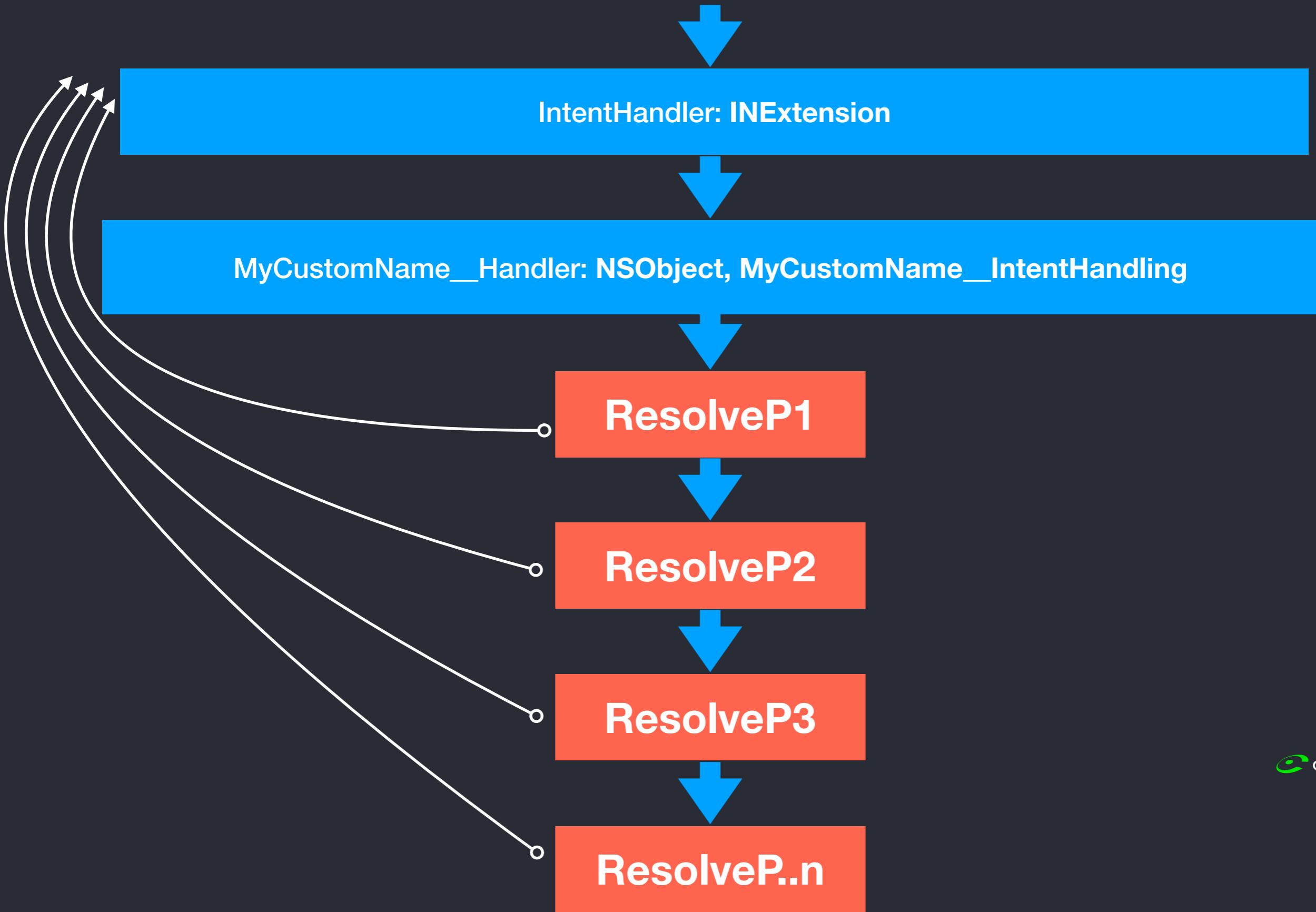


Life Cycle

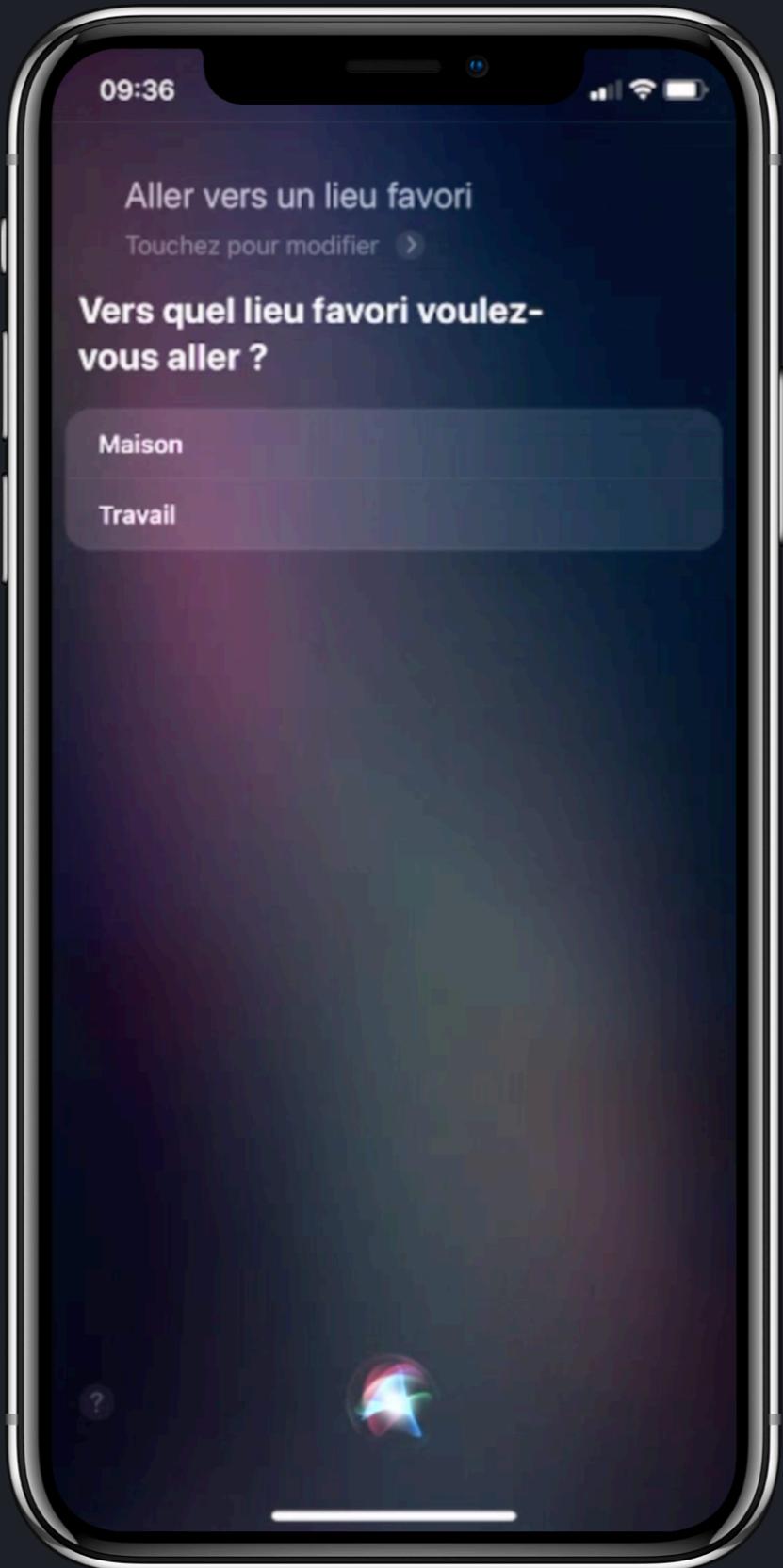


COYOTE

Life Cycle Intent extension



Revolve Parameters



Life Cycle Intent extension



IntentHandlerProviding: INExtension, INIntentHandlerProviding, NSObject



MyCustom_IntentHandler: MyCustom_IntentHandling, NSObject



Optional

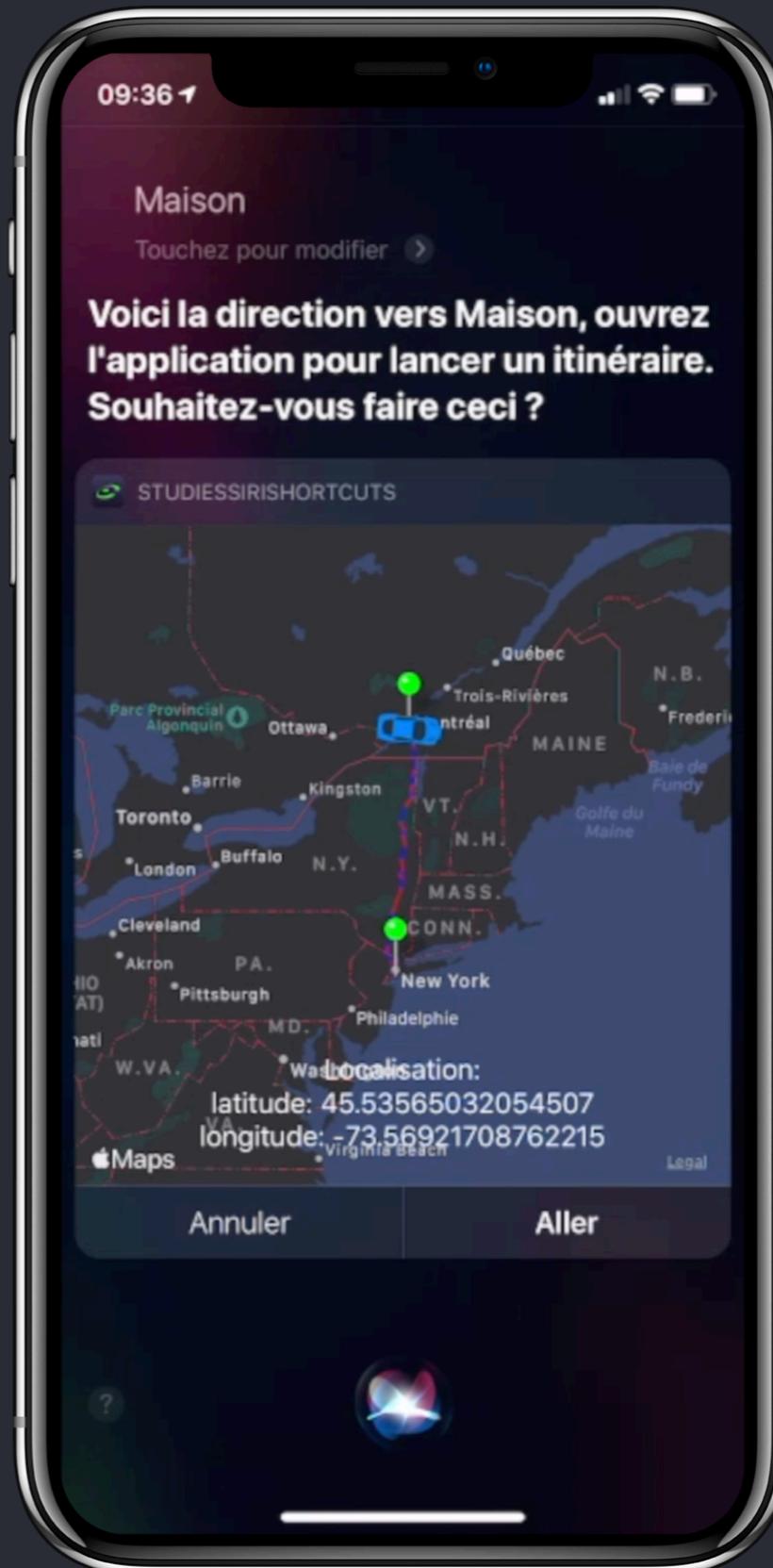
Confirm



IntentHandlerUIHosterViewController: UIViewController, INUIHostedViewControlling

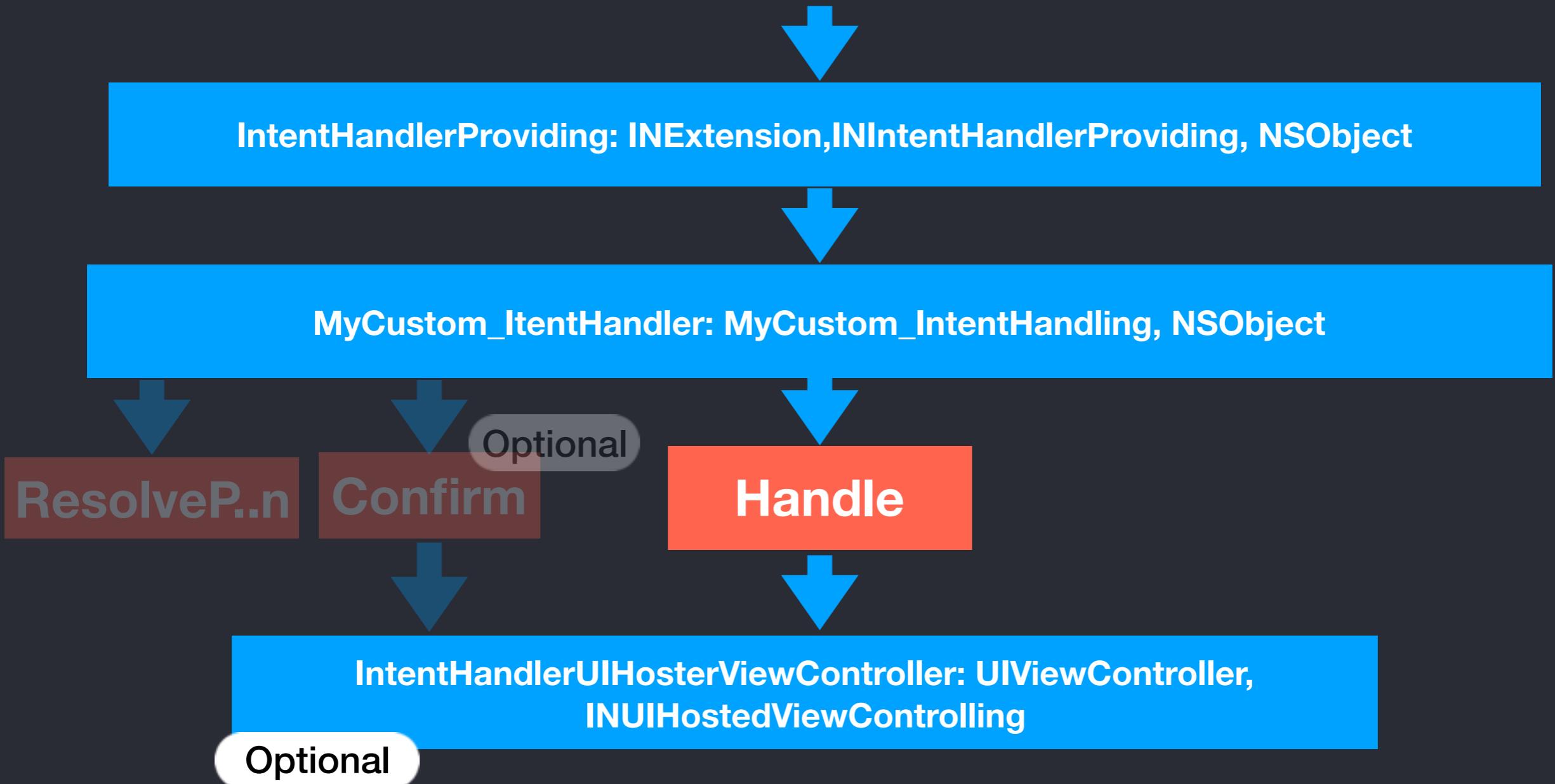
Optional

Confirm

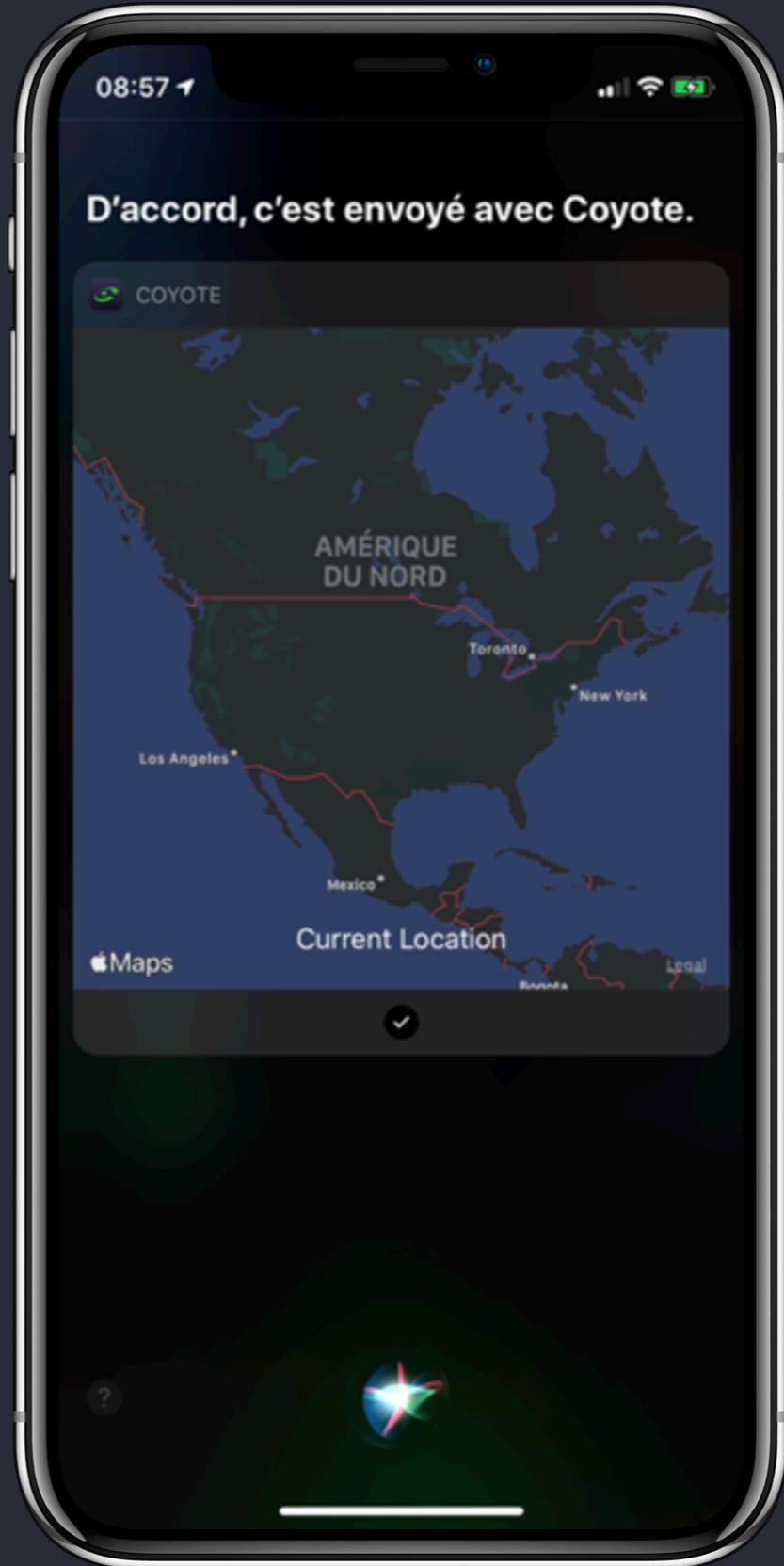


COYOTE

Life Cycle Intent extension



Handle



COYOTE

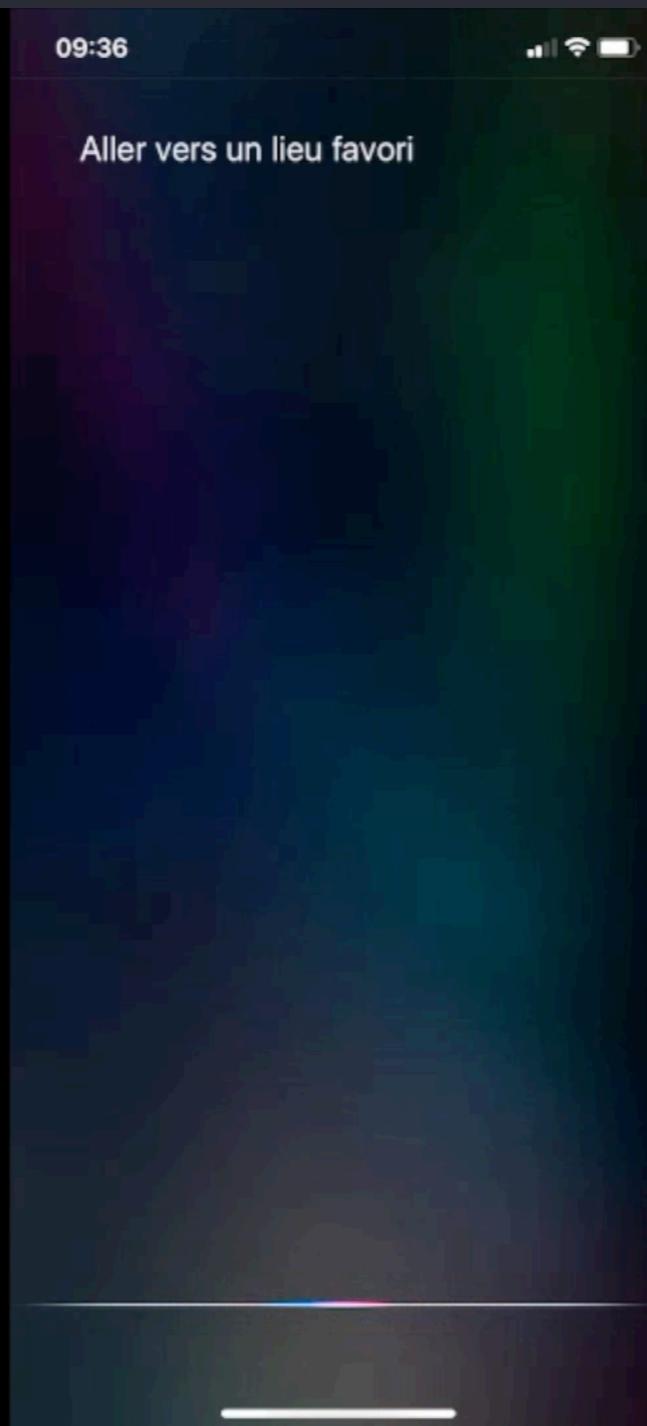


Sample application with Intent and UI extension

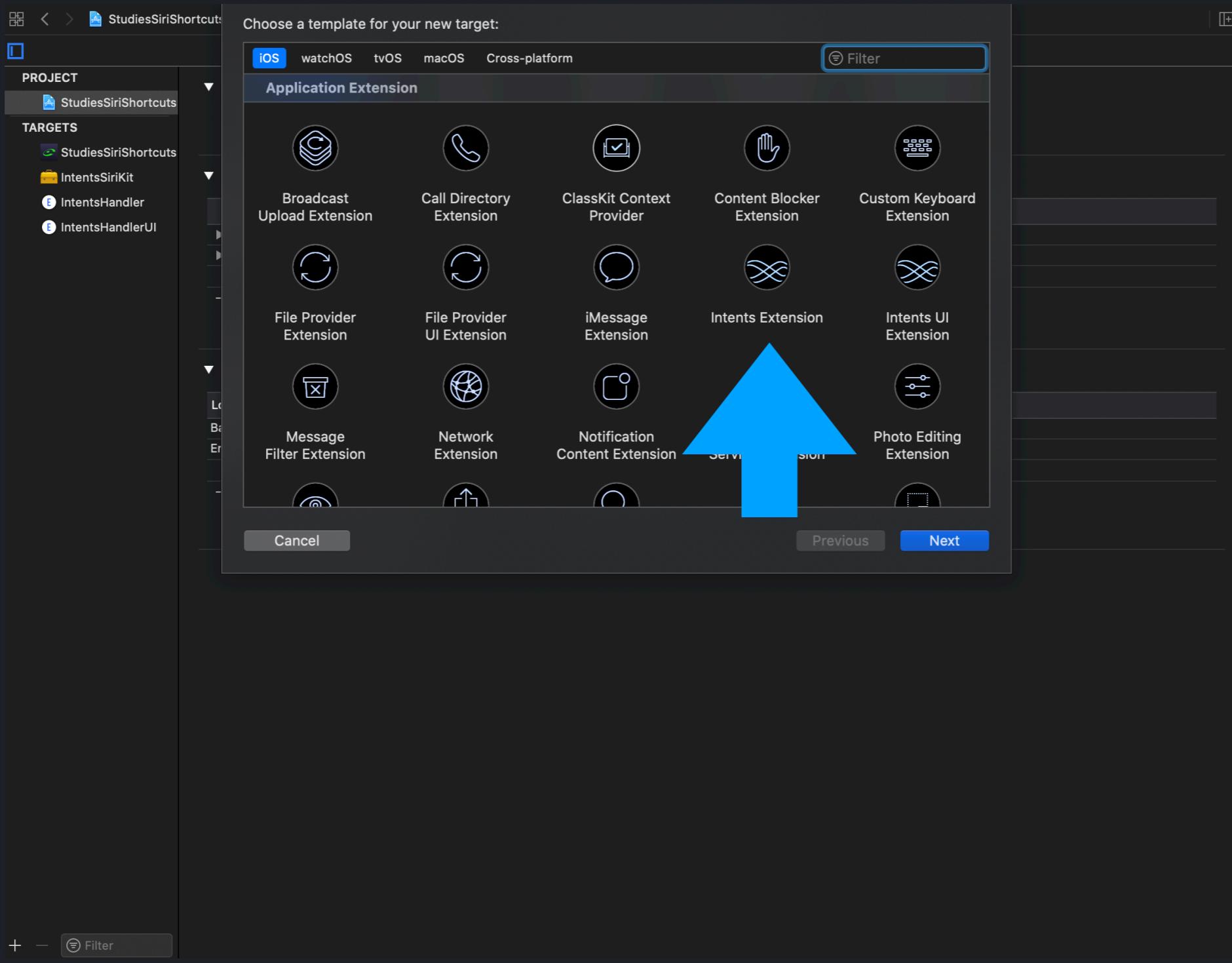


COYOTE

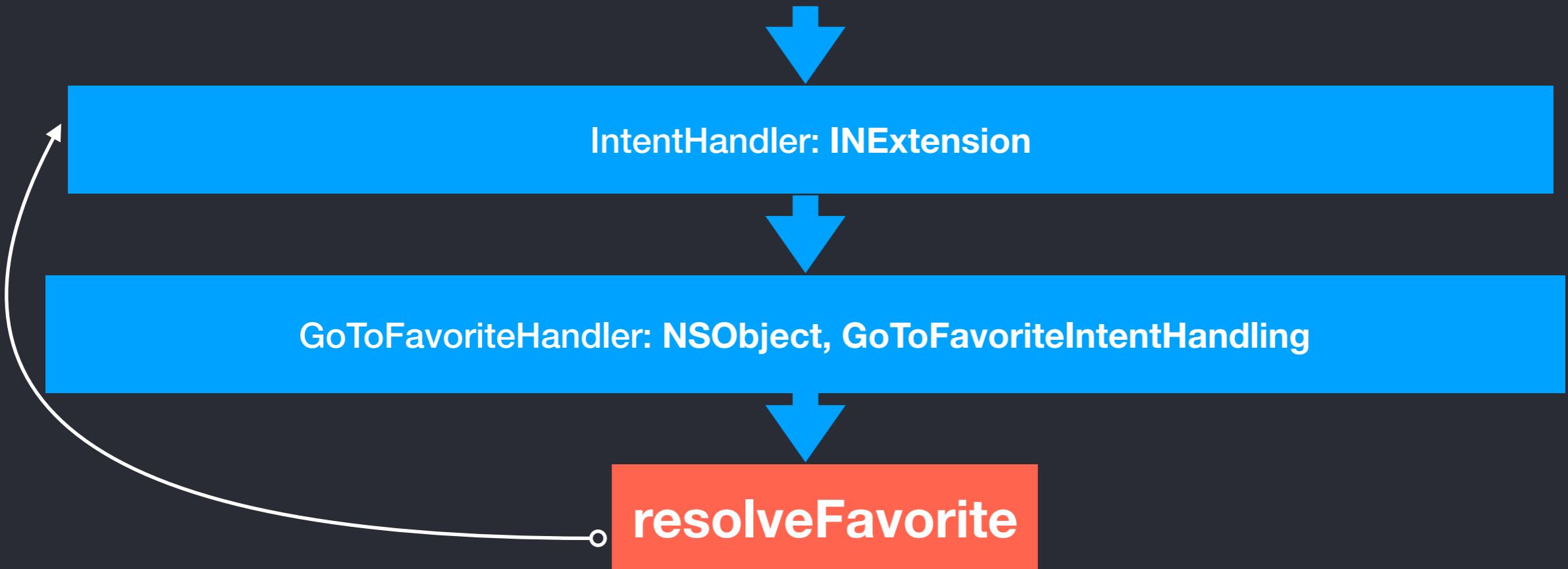
Sample App



Intents Extension



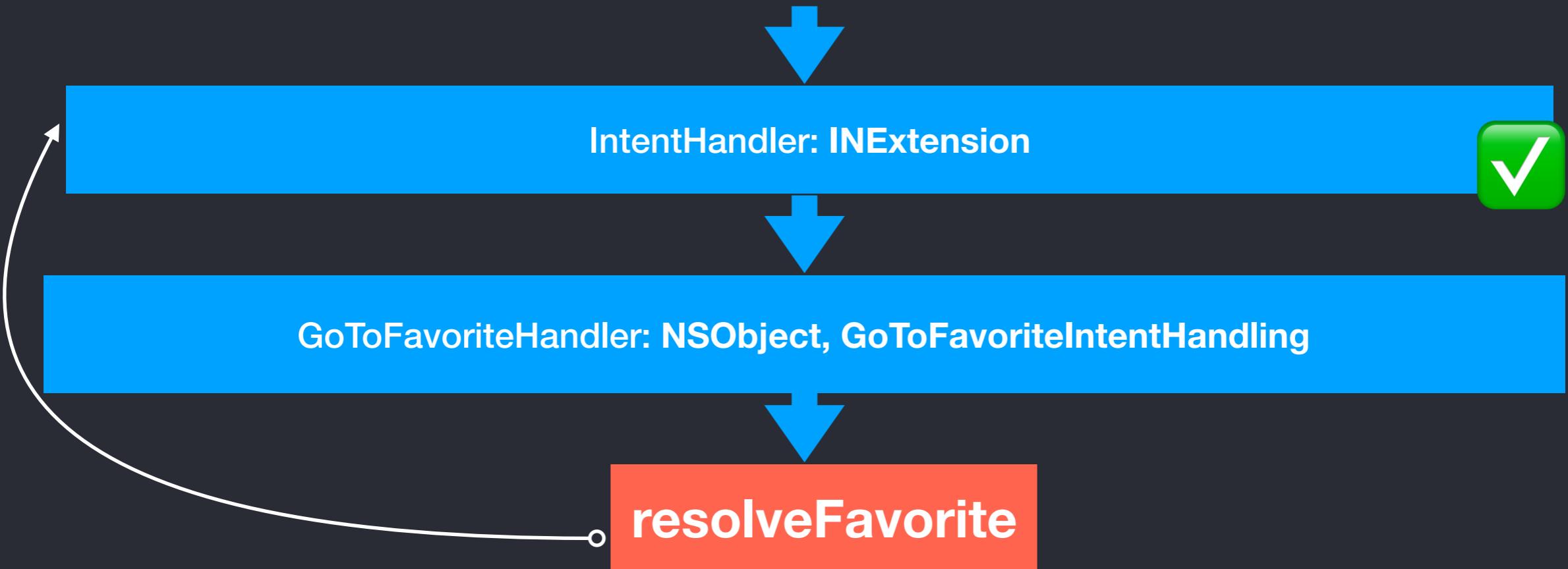
IntentHandler



IntentHandler

```
final class IntentHandler: INExtension {  
  
    override func handler(for intent: INIntent) -> Any {  
        switch intent {  
            case is GoToFavoriteIntent:  
                return GoToFavoriteHandler()  
  
            // ...  
        }  
    }  
}
```

GoToFavoriteHandler



Custom Intents

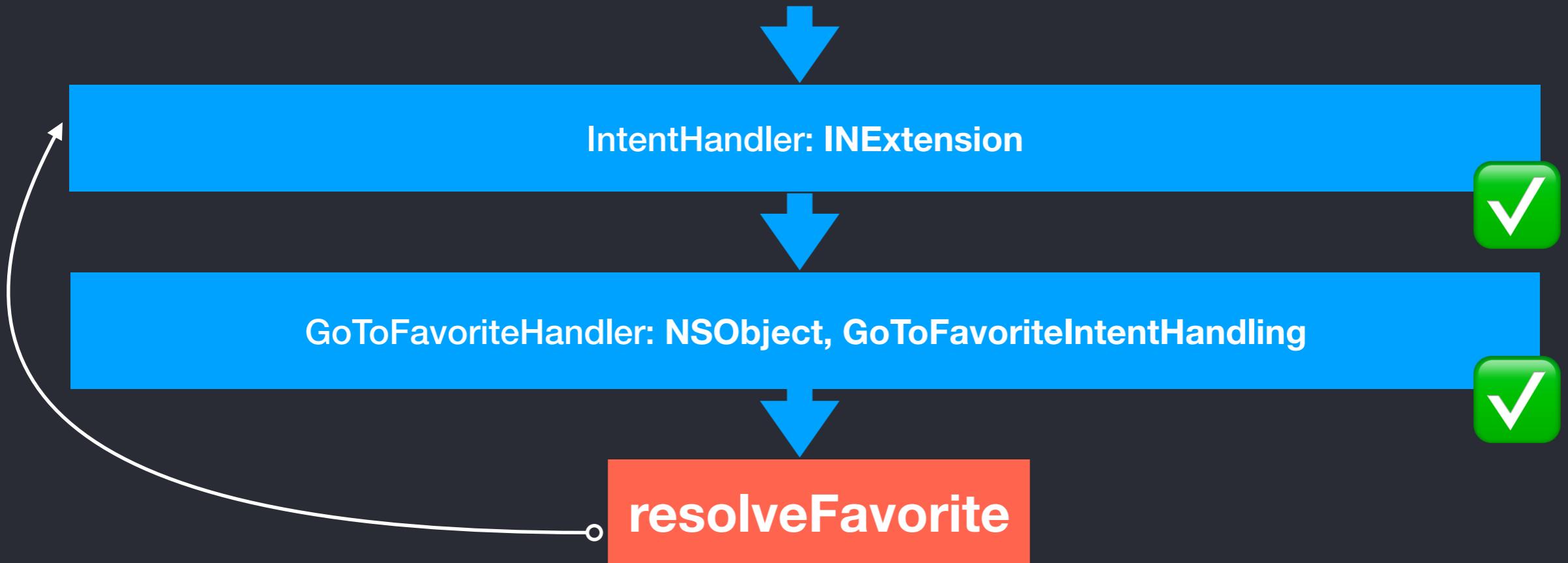
GoToFavoriteIntentHandling

```
public protocol GoToFavoriteIntentHandling: NSObjectProtocol {  
  
    func resolveFavorite(for intent: GoToFavoriteIntent,  
                        with completion: @escaping (FavoriteResolutionResult) ->  
Swift.Void)  
  
    func confirm(intent: GoToFavoritesIntent,  
                completion: @escaping (GoToFavoriteIntentResponse) -> Swift.Void)  
  
    func handle(intent: GoToFavoriteIntent,  
                completion: @escaping (GoToFavoriteIntentResponse) -> Swift.Void)  
}
```

GoToFavoriteHandler

```
public class GoToFavoriteHandler: NSObject, GoToFavoriteIntentHandling {  
    // ...  
}
```

Resolve Parameters



GoToFavoriteHandler

```
public class GoToFavoriteHandler: NSObject,  
GoToFavoriteIntentHandling {  
  
    public func resolveFavorite(for intent: GoToFavoriteIntent,  
                                with completion: @escaping  
(FavoriteResolutionResult) -> Void) {  
        // ...  
    }  
}
```



GoToFavoritIntent

```
@available(iOS 12.0, watchOS 5.0, *) @available(macOS, unavailable) @available(tvOS, unavailable)
@objc(GoToFavoriteIntent)
public class GoToFavoriteIntent: INIntent {
    @NSManaged public var favorite: Favorite?
}
```



Custom Intents

The screenshot shows the 'CUSTOM INTENTS' screen in the Apple Shortcuts editor. A specific custom intent, 'GoToFavorite', is selected and highlighted with a blue background. The right-hand panel displays the configuration for this intent.

Custom Intent Configuration:

- Category:** Go
- Title:** Aller vers un lieu favori
- Description:** Direction vers lieu un favori avec la Map
- Default Image:** iconLocation
- Confirmation:** User confirmation required

Parameters:

- Parameter:** favorite
- Display Name:** Lieu favori
- Type:** favori
- Array:** Supports multiple values
- User-facing:** User can supply value in Siri and Shortcuts app
- Dynamic Options:** Valid values are provided dynamically

Relationship:

- Parent Parameter:** None

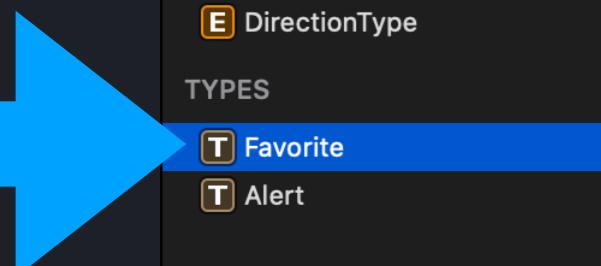
Siri Dialog:

- Prompt:** Vers quel lieu favori ?
- Disambiguation:** Customize disambiguation dialog
- Disambiguation Introduction:** Vous avez `count` options de disponible.
- Disambiguation Prompt:** Vers quel lieu favori voulez-vous aller ?
- Parameter Confirmation:** [Empty field]

Validation Errors (0)

Code	Template
+	-

Types



The screenshot shows a user interface for managing types in a system. On the left, there's a sidebar with sections for CUSTOM INTENTS, ENUMS, and TYPES. The TYPES section contains three items: 'Favorite' (selected and highlighted with a blue background), 'Alert', and 'DirectionType'. A large blue arrow points from the left towards the 'Favorite' item.

CUSTOM INTENTS

- I GoToFavorite
- R Response
- I RequestAlert
- R Response

ENUMS

- E DirectionType

TYPES

- T Favorite**
- T Alert

Type

Display Name favori

Properties

Property
S identifier
S displayString
L location

+ -

Display Name Identifier

Type String

Array Supports multiple values

Favorite

```
public class Favorite: INObject {  
    @available(iOS 13.0, watchOS 6.0, *)  
    @NSManaged public var location: CLPlacemark?  
}
```

GoToFavoriteIntentHandler

```
public class GoToFavoriteIntentHandler: NSObject, GoToFavoriteIntentHandling {  
    public func resolveFavorite(for intent: GoToFavoriteIntent,  
                               with completion: @escaping  
(FavoriteResolutionResult) -> Void) {  
  
    guard let favoris = intent.favorite else {  
        // ... Call API  
        completion(.disambiguation(with: favorites))  
        // ...  
    }  
  
    completion(.success(with: favoris))  
}  
}
```

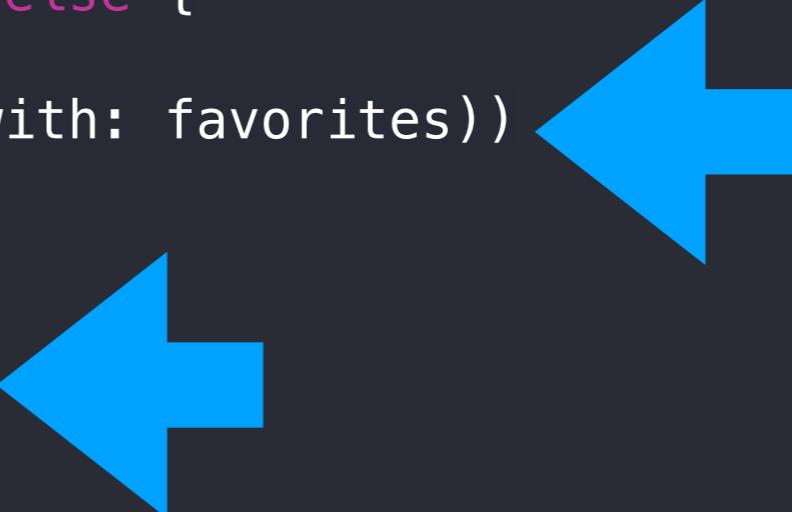


FavoriteResolutionResult / Parameters_ResolutionResult

```
public class FavoriteResolutionResult: INObjectResolutionResult {  
  
    public class func success(with resolvedObject: Favorite) -> Self { ... }  
  
    public class func disambiguation(with objectsToDisambiguate: [Favorite]) -> Self { ... }  
  
    public class func confirmationRequired(with objectToConfirm: Favorite?) -> Self { ... }  
  
    public class func needsValue() { ... }  
  
    public class func notRequired() { ... }  
  
    public class func unsupported() { ... }  
  
}
```

GoToFavoriteIntentHandler

```
public class GoToFavoriteIntentHandler: NSObject, GoToFavoriteIntentHandling {  
    public func resolveFavorite(for intent: GoToFavoriteIntent,  
                               with completion: @escaping  
(FavoriteResolutionResult) -> Void) {  
  
    guard let favoris = intent.favorite else {  
        // ... Call API  
        completion(.disambiguation(with: favorites))  
        // ...  
    }  
  
    completion(.success(with: favoris))  
}
```



Custom Intents

The screenshot shows the Apple Shortcuts editor interface for creating a custom intent. A large blue arrow points from the title 'Custom Intents' down to the 'Custom Intent' section.

CUSTOM INTENTS

- I GoToFavorite
- R Response
- I RequestAlert
- R Response

ENUMS

- E DirectionType

TYPES

- T Favorite
- T Alert

Custom Intent

Category: Go

Title: Aller vers un lieu favori

Description: Direction vers lieu un favori avec la Map

Default Image: iconLocation

Confirmation: User confirmation required

Parameters

Parameter
T favorite

Display Name: Lieu favori

Type: favori

Array: Supports multiple values

User-facing: User can supply value in Siri and Shortcuts app

Dynamic Options: Valid values are provided dynamically

Relationship

Parent Parameter: None

Siri Dialog

Prompt: Vers quel lieu favori ?

Disambiguation: Customize disambiguation dialog

Disambiguation Introduction: Vous avez `count` options de disponible.

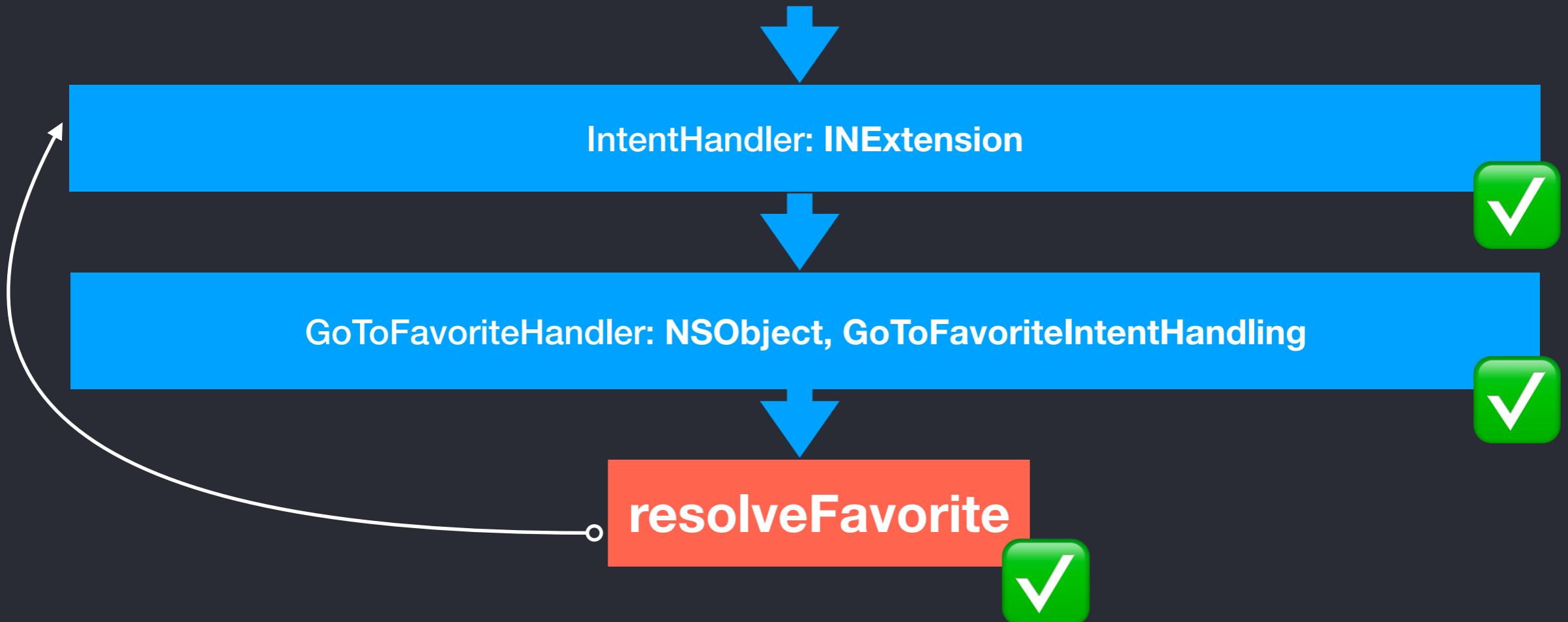
Disambiguation Prompt: Vers quel lieu favori voulez-vous aller ?

Parameter Confirmation:

Validation Errors (0)

Code	Template
+	-

Resolve Parameters



Confirm



IntentHandler: INExtension, INIntentHandlerProviding, NSObject



GoToFavoriteIntentHandler: GoToFavoriteIntentHandling, NSObject



resolveFavorite



Confirm



IntentHandlerUIHosterViewController: UIViewController, INUIHostedViewControlling

Custom Intents

Category	Go
Title	Aller vers un lieu favori
Description	Direction vers lieu un favori avec la Map
Default Image	iconLocation
Confirmation	<input type="checkbox"/> User confirmation required

Category	Go
Title	Aller vers un lieu favori
Description	Direction vers lieu un favori avec la Map
Default Image	iconLocation
Confirmation	<input checked="" type="checkbox"/> User confirmation required

GoToFavoriteIntentHandler

```
public class GoToFavoriteIntentHandler: NSObject, GoToFavoriteIntentHandling {  
    public func confirm(intent: GoToFavoriteIntent, completion: @escaping  
    (GoToFavoriteIntentResponse) -> Swift.Void) { }  
  
    public func handle(intent: GoToFavoriteIntent, completion: @escaping  
    (GoToFavoriteIntentResponse) -> Void) { }  
}
```



Custom Intents Response

The screenshot shows a software interface for managing custom intents and their responses. On the left, there's a sidebar with categories: CUSTOM INTENTS, ENUMS, and TYPES. Under CUSTOM INTENTS, there are several items: GoToFavorite, Response (which is selected), RequestAlert, and another Response item. Under ENUMS, there's a DirectionType item. Under TYPES, there are Favorite and Alert items.

The main area displays the properties of the selected 'Response' intent. It has a 'Properties' section with a table:

Property	Type	Display Name
S failureMessageDisplay	String	Failure Message Display
S successLocationDisplay	String	Success Location Display

Below this, there's an 'Output' dropdown set to 'None'. There are also '+' and '-' buttons for adding or removing properties.

At the bottom, there's a 'Response Templates' section with a 'Code' tab active. It contains the text 'readyOpenMap'. To the right of this, there are sections for 'Error' (checkbox 'Is error response' is unchecked), 'Voice-Only Dialog' (text: 'Voici la direction vers displaySuccessOpenMapFavoriteName , ouvrir l\'application'), and 'Printed Dialog' (text: 'Voici la direction vers displaySuccessOpenMapFavoriteName').

Large blue arrows are overlaid on the interface: one pointing down from the top-left towards the 'Properties' table, one pointing down from the top-right towards the 'Success Location Display' row, one pointing up from the bottom-left towards the 'Code' tab, and one pointing down from the bottom-right towards the 'Printed Dialog' text.

GoToFavoriteIntentResponse

```
public class GoToFavoriteIntentResponse: INIntentResponse {  
    /// ...  
  
    public static func readyOpenMap(displaySuccessOpenMapFavoriteName:  
String) -> GoToFavoriteDirectionsIntentResponse { }  
  
    /// ...  
}
```

Titre

```
@interface INIntentResponse : NSObject <NSCopying, NSSecureCoding>  
@property (copy, readwrite, nullable, nonatomic) NSUserActivity *userActivity;  
@end
```



Custom Intents Response

The image shows a smartphone screen displaying a Siri card. The card has a dark background with white text. At the top, it says "Maison" and "Touchez pour modifier". Below that, it says "Voici la direction vers Maison, ouvrez l'application pour lancer un itinéraire. Souhaitez-vous faire ceci ?". At the bottom are two buttons: "Annuler" on the left and "Aller" on the right. A large blue arrow points downwards from the top of the image towards the Siri card. Another large blue arrow points to the left from the right side of the Siri card towards a screenshot of the Coyote app interface.

▼ Response Templates

Code

readyOpenMap

+ -

Error Is error response

Voice-Only Dialog

Voici la direction vers `successLocationDisplay`, ouvrir l'application pour lancer un itinéraire.

Printed Dialog

Voici la direction vers `successLocationDisplay`, ouvrir l'application pour lancer un itinéraire.

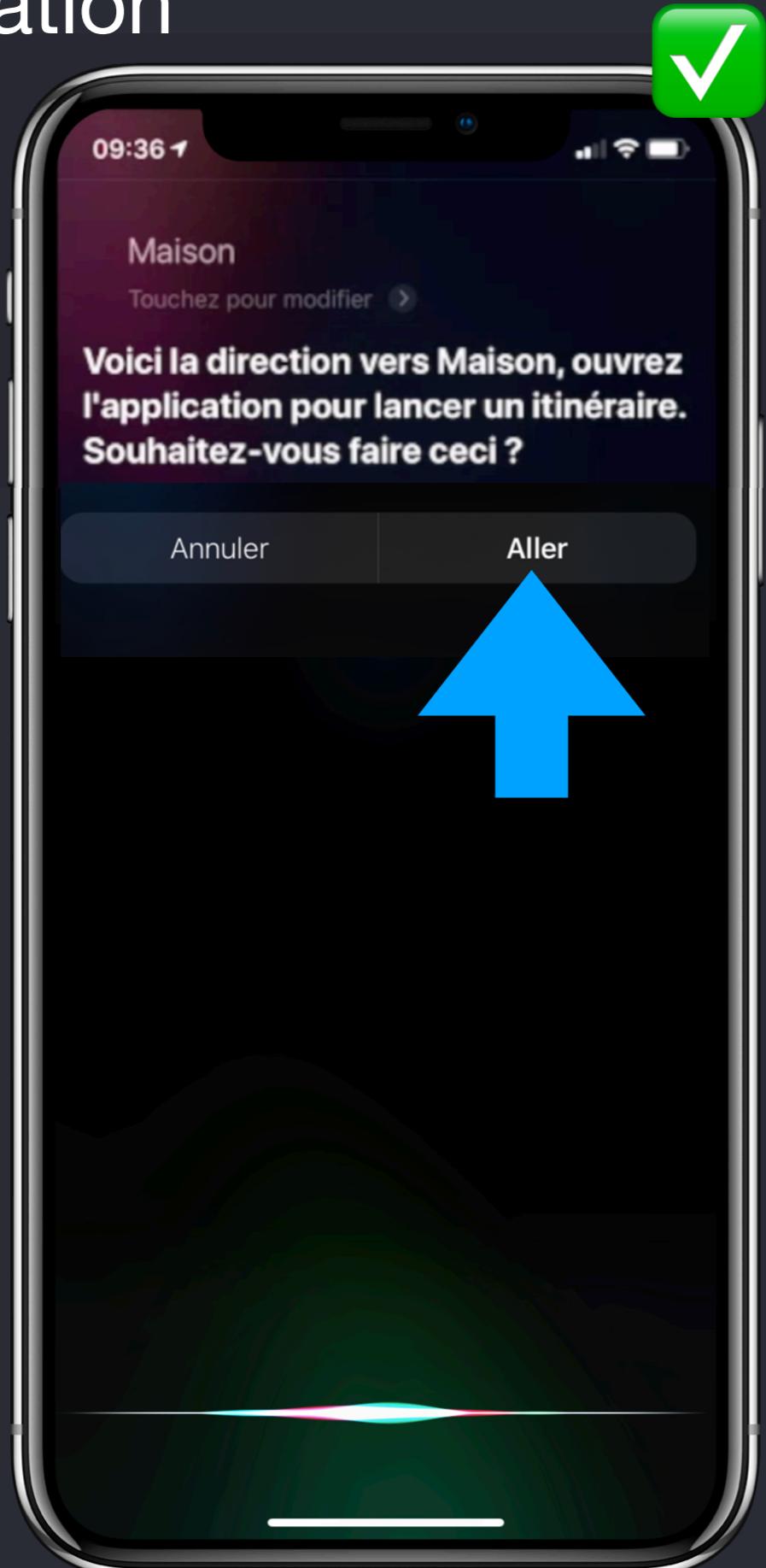
GoToFavoriteIntentResponseCode

```
public enum GoToFavoriteIntentResponseCode: Int {  
    /// Natif  
    case unspecified = 0  
    case ready  
    case continueInApp  
    case inProgress  
    case success  
    case failure  
    case failureRequiringAppLaunch  
    /// Custom  
    case readyOpenMap  
}
```

GoToFavoriteIntentHandler

```
public class GoToFavoriteIntentHandler: NSObject, GoToFavoriteIntentHandling {  
    // ...  
  
    public func confirm(intent: GoToFavoriteIntent,  
                        completion: @escaping (GoToFavoriteIntentResponse) ->  
Swift.Void) {  
  
    // ...  
    let response =  
GoToFavoriteIntentResponse(code: .failureRequiringAppLaunch,  
                           userActivity: nil)  
  
    // ...  
    let response = GoToFavoriteIntentResponse(code: .readyOpenMap,  
                                              userActivity: nil)  
  
    // ...  
    completion(response)  
}  
}
```

Confirmation



The Shortcuts editor interface is shown on the right. It has a sidebar with sections like "Custom Intent", "Parameters", "Relationship", "Siri Dialog", and "Validation Errors (0)".
Custom Intent:

- Category: Go
- Title: Aller vers un lieu favori
- Description: Direction vers lieu un favori avec la Map
- Default Image: iconLocation
- Confirmation: User confirmation required

Parameters:

Parameter	Display Name	Type	Dynamic Options
favorite	Lieu favori	favori	<input type="checkbox"/> Supports multiple values <input checked="" type="checkbox"/> User can supply value in Siri and Shortcuts app <input type="checkbox"/> Valid values are provided dynamically

Siri Dialog:

- Prompt: Vers quel lieu favori ?
- Disambiguation: Customize disambiguation dialog
- Disambiguation Introduction: Vous avez `count` options de disponible.
- Disambiguation Prompt: Vers quel lieu favori voulez-vous aller ?
- Parameter Confirmation:

Validation Errors (0):

Code	Template
+	-

IntentHandlerUIHosterViewController



IntentHandler: INExtension, INIntentHandlerProviding, NSObject



GoToFavoriteIntentHandler: GoToFavoriteIntentHandling, NSObject



resolveFavorite

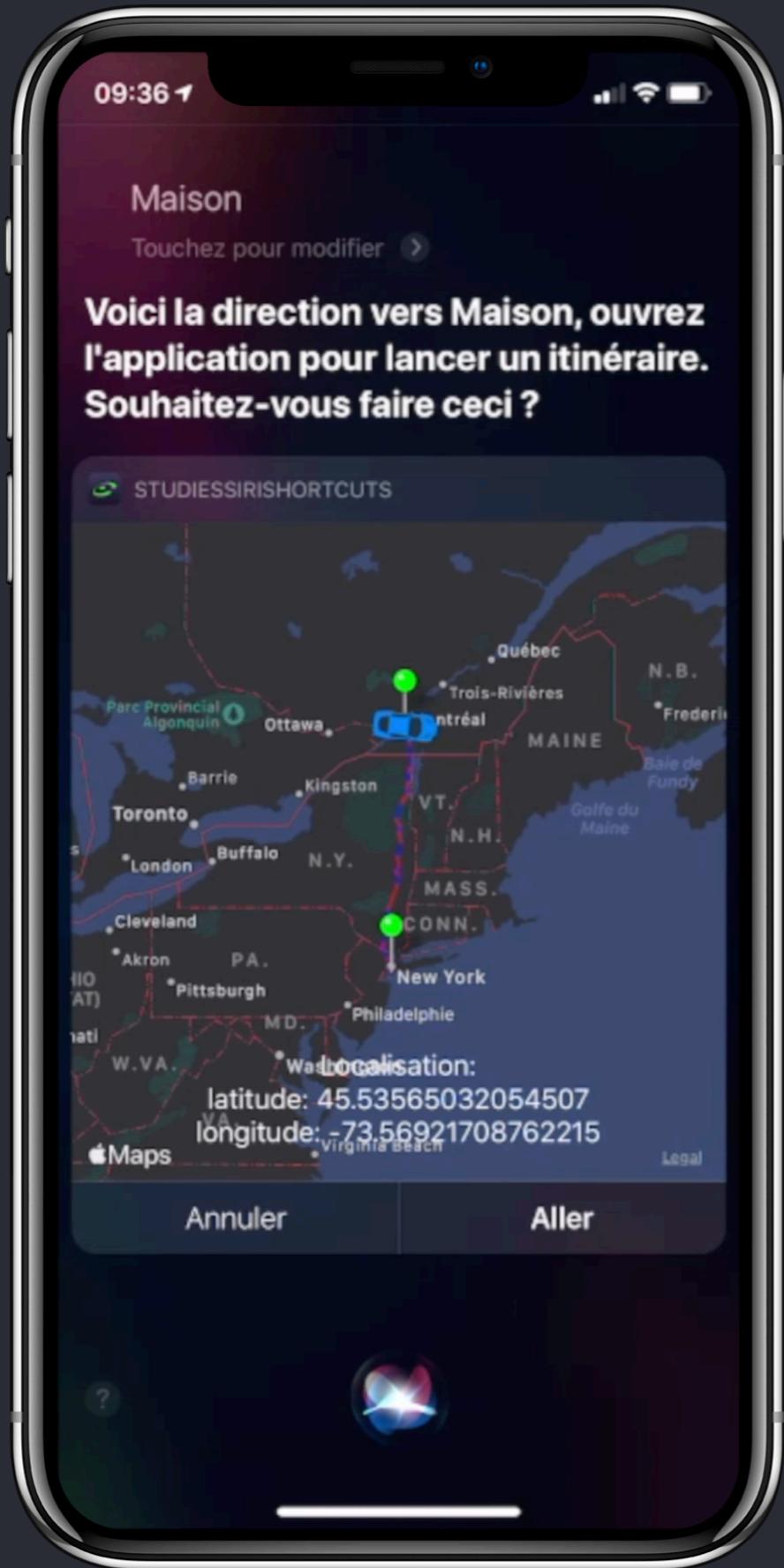


Confirm

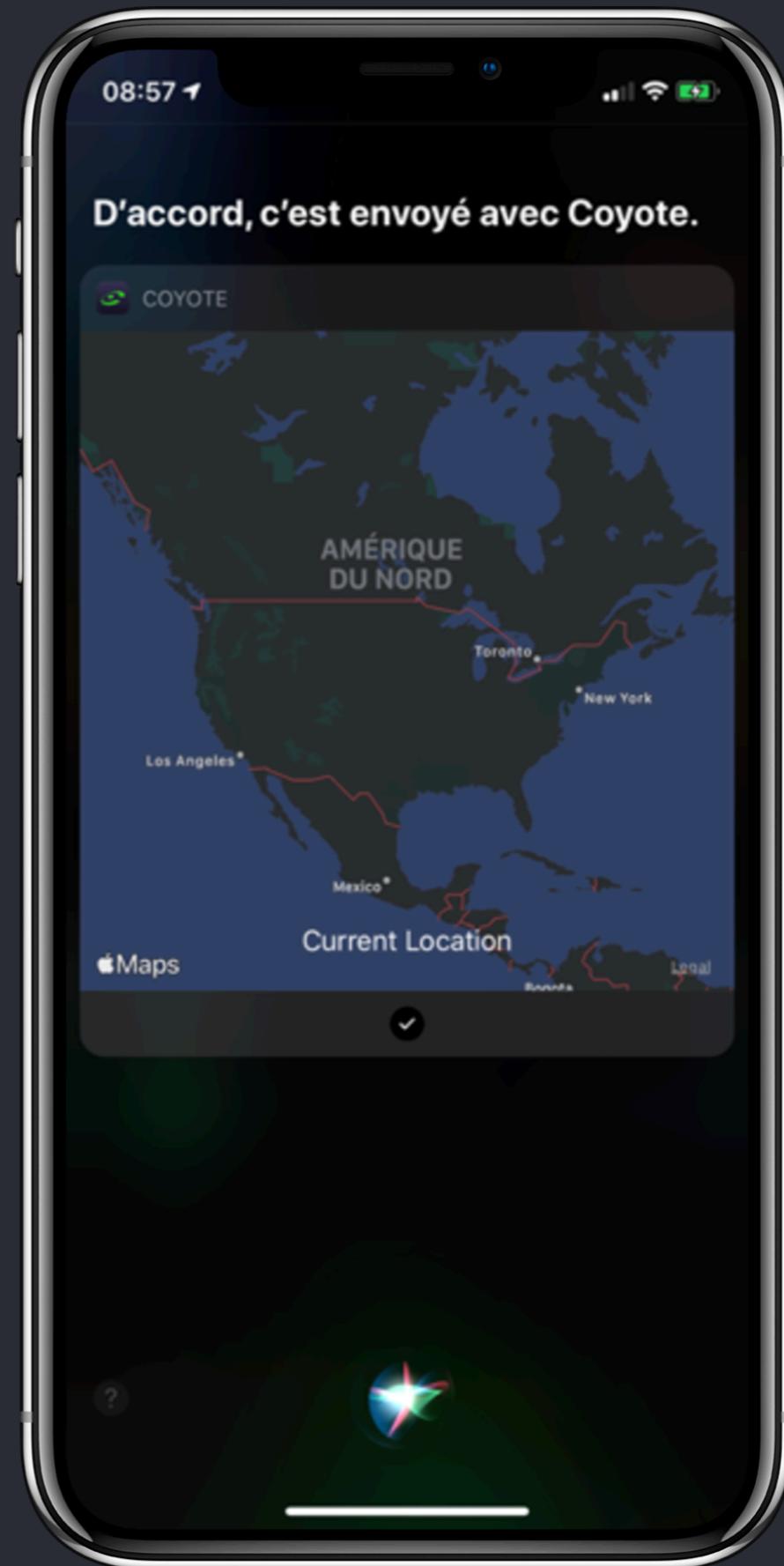


IntentHandlerUIHosterViewController: UIViewController,
INUIHostedViewControlling

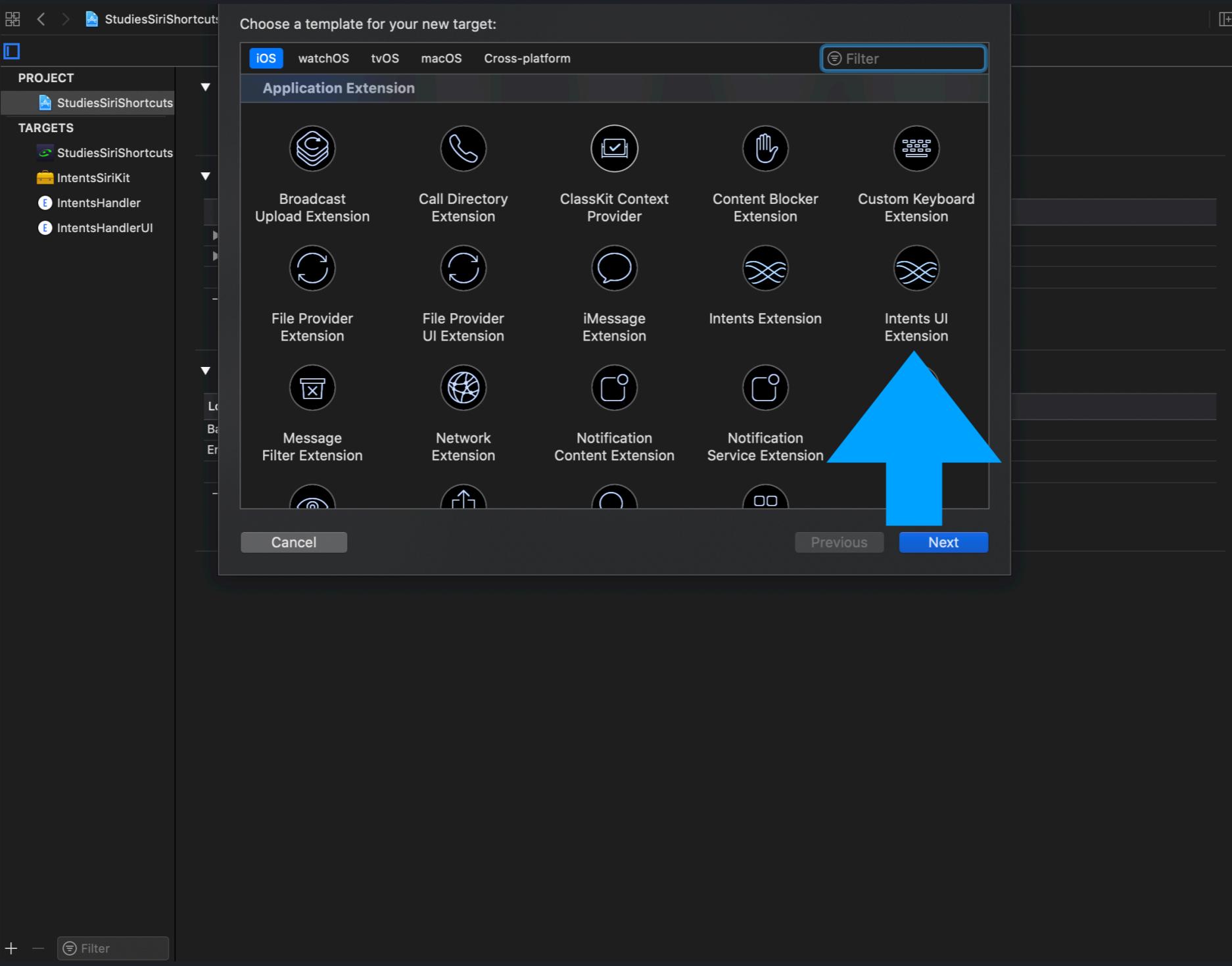
Confirm



Handle



Intents UI Extension



INUIHostedViewControlling

```
public protocol INUIHostedViewControlling : NSObjectProtocol {  
    /// ...  
    func configureView(for parameters: Set<INParameter>,  
                      of interaction: INInteraction,  
                      interactiveBehavior: INUIInteractiveBehavior,  
                      context: INUIHostedViewContext,  
                      completion: @escaping (Bool, Set<INParameter>, CGSize) -> Void)  
}
```

IntentsHandlerUIViewController

```
final class IntentsHandlerUIViewController: UIViewController,  
INUIHostedViewControlling {  
  
    /// ...  
    func configureView(for parameters: Set<INParameter>,  
                       of interaction: INInteraction,  
                       interactiveBehavior: INUIInteractiveBehavior,  
                       context: INUIHostedViewContext,  
                       completion: @escaping (Bool, Set<INParameter>, CGSize) -> Void)  
  
        switch (interaction.intentResponse) {  
        case (let response as GoToFavoriteLocationIntentResponse  
              where response.code == .readyOpenMap:  
  
            let viewController = CustomViewController()  
            attachChild(viewController)  
            completion(true, parameters, desiredSize)  
            /// ...  
        }  
    }  
    /// ...
```

GoToFavoriteIntentHandler

```
public class GoToFavoriteIntentHandler: NSObject, GoToFavoriteIntentHandling {  
    // ...  
    public func confirm(intent: GoToFavoriteIntent, completion: @escaping  
(GoToFavoriteIntentResponse) -> Swift.Void) {  
        // ...  
        let response = GoToFavoriteIntentResponse(code: .readyOpenMap,  
                                                userActivity: userActivity)  
        completion(response)  
    }  
}
```



IntentsHandlerUIViewController

```
final class IntentsHandlerUIViewController: UIViewController,  
INUIHostedViewControlling {  
  
    /// ...  
    func configureView(for parameters: Set<INParameter>,  
                      of interaction: INInteraction,  
                      interactiveBehavior: INUIInteractiveBehavior,  
                      context: INUIHostedViewContext,  
                      completion: @escaping (Bool, Set<INParameter>, CGSize) -> Void)  
  
        /// ...  
        let userActivity = interaction.intentResponse?.userActivity  
        viewController.restoreUserActivityState(userActivity)  
        /// ...  
    }  
    /// ...
```



Confirm



IntentHandlerUIHosterViewController



IntentHandler: INExtension, INIntentHandlerProviding, NSObject



GoToFavoriteIntentHandler: GoToFavoriteIntentHandling, NSObject



resolveFavorite



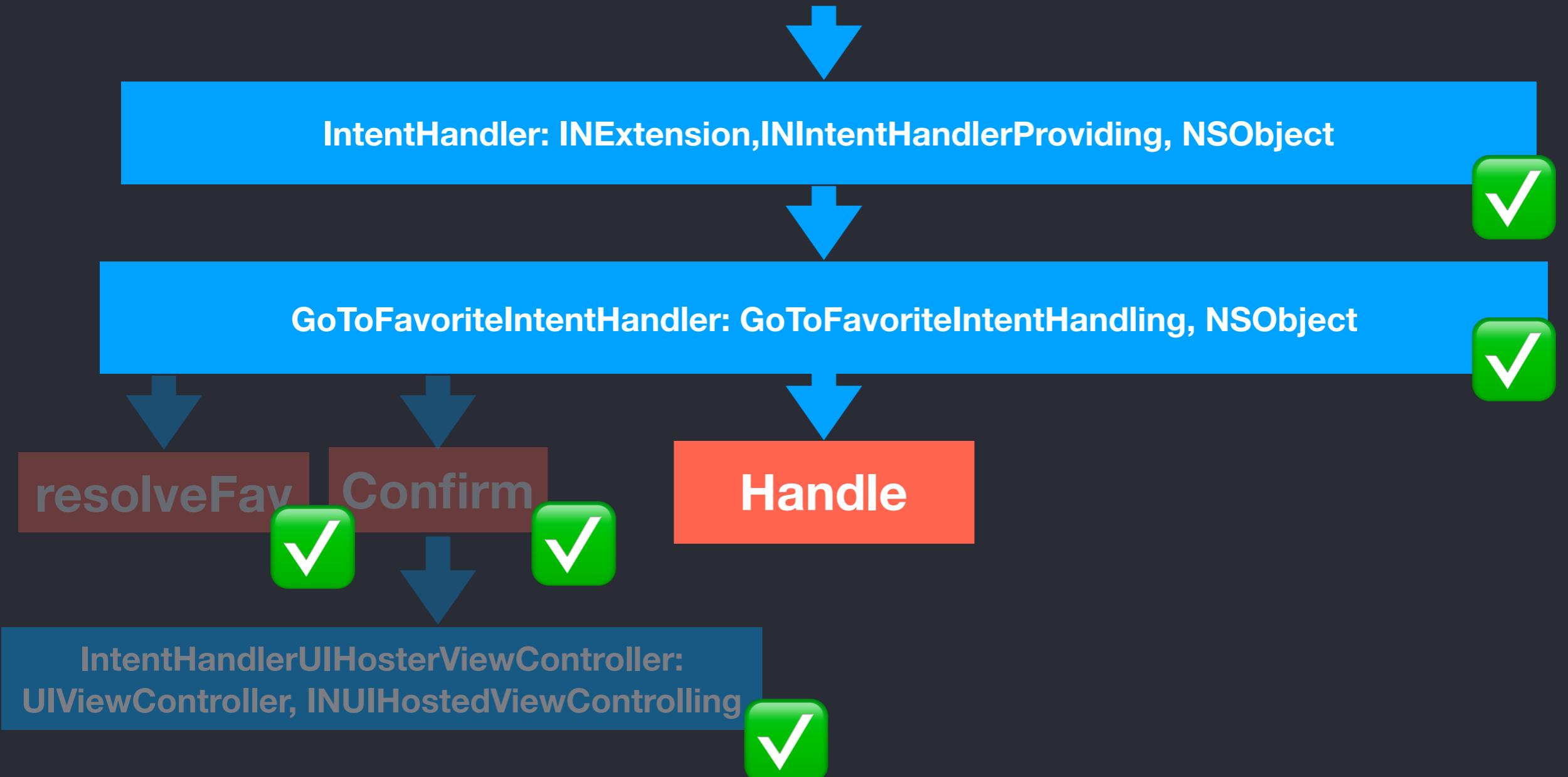
Confirm



IntentHandlerUIHosterViewController: UIViewController,
INUIHostedViewControlling



Handle



GoToFavoriteIntentHandler

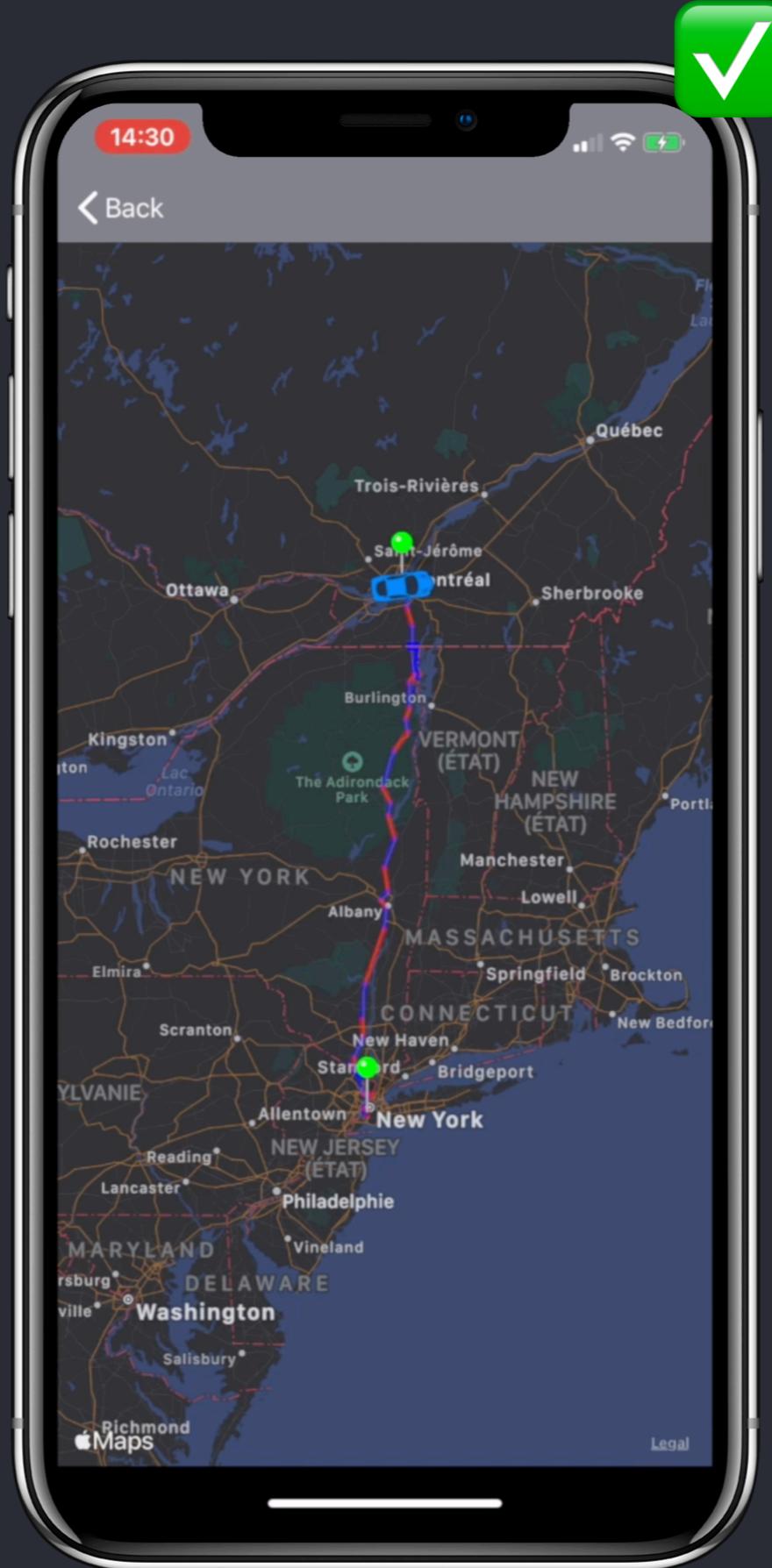
```
final class GoToFavoriteIntentHandler: NSObject, GoToFavoriteIntentHandling {  
    /// ...  
    public func handle(intent: GoToFavoriteIntent,  
                      completion: @escaping (GoToFavoriteIntentResponse) -> Void) {  
        /// ...  
  
        let response = GoToFavoriteIntentResponse(code: .continueInApp,  
                                                userActivity: userActivity)  
        completion(response)  
    }  
}
```

UIApplicationDelegate

```
/// ...
func application(_ application: UIApplication,
                  continue userActivity: NSUserActivity,
                  restorationHandler: @escaping ([UIUserActivityRestoring]?) ->
Void) -> Bool {
    guard
        userActivity.activityType == NSStringFromClass(MyCustomActivity.self),
        /// ...
    else {
        return false
    }
    /// ...
    restorationHandler(rootViewController.viewControllers)
    return true
}
```

ViewController

```
final class ViewController: UIViewController {  
    // ...  
    override func restoreUserActivityState(_ activity: NSUserActivity) {  
        super.restoreUserActivityState(activity)  
        // activity.activityType  
        // activity.userInfo  
        // ...  
    }  
}
```



COYOTE

Ending

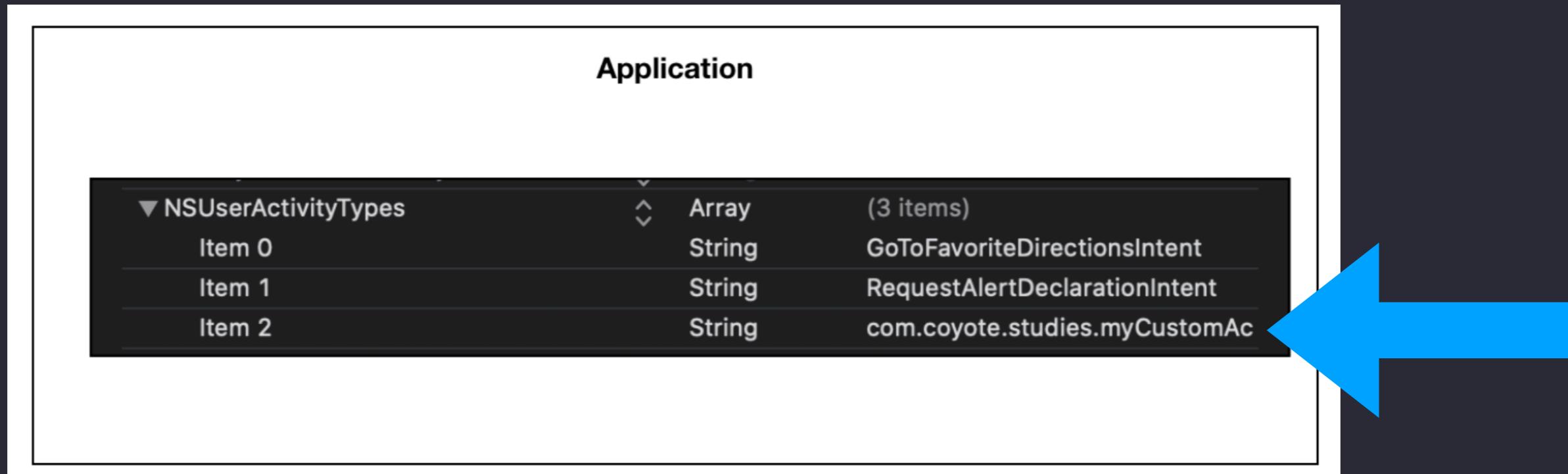




Merci

1. <https://github.com/CoyoteLab/Studies-Siri-Shortcut>
2. ystephan@coyote-group.com

Définition d'une *Activity*



Application

▼ NSUserActivityTypes	Array (3 items)
Item 0	String GoToFavoriteDirectionsIntent
Item 1	String RequestAlertDeclarationIntent
Item 2	String com.coyote.studies.myCustomAc

Extensions d'intentions (Logique)

▼ Supported Intents

Class Name	Authentication
GoToFavoriteDirection...	None ◊
RequestAlertDeclarati...	None ◊
+	-

▼ Frameworks and Libraries

Name	Embed
IntentsSiriKit.framework	Do Not Embed ◊
+	-

Extension d'interface utilisateur (UI)

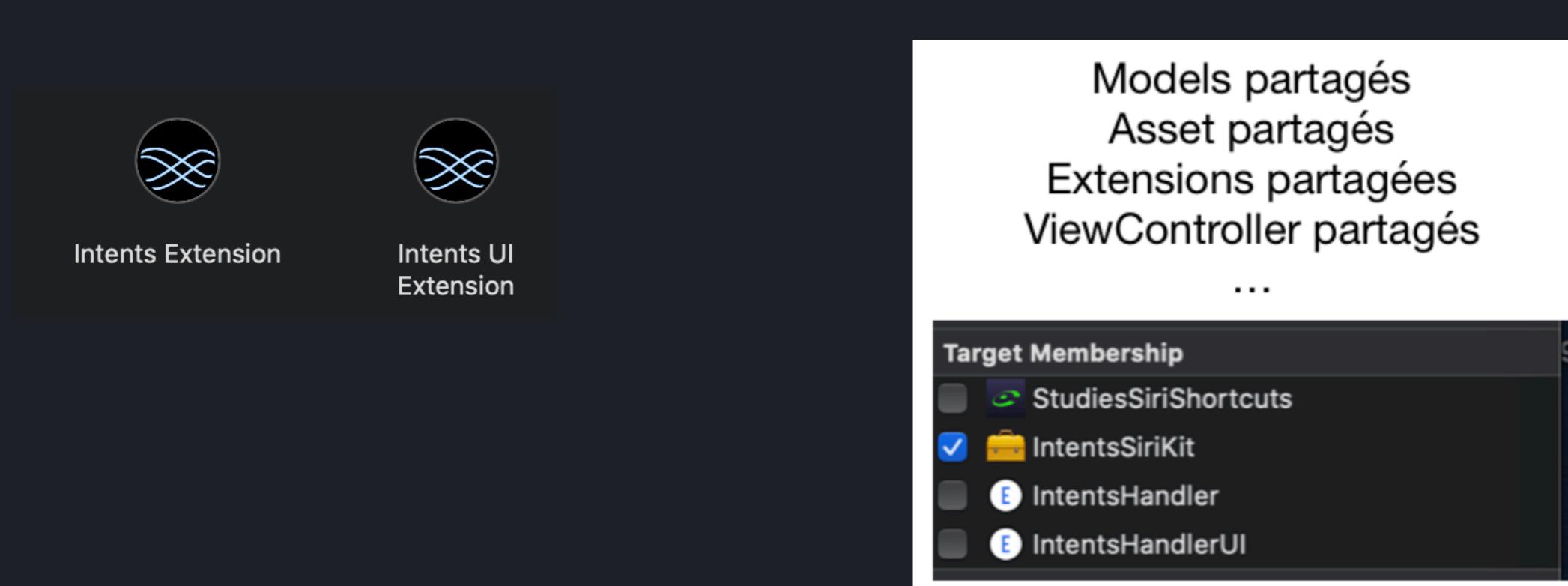
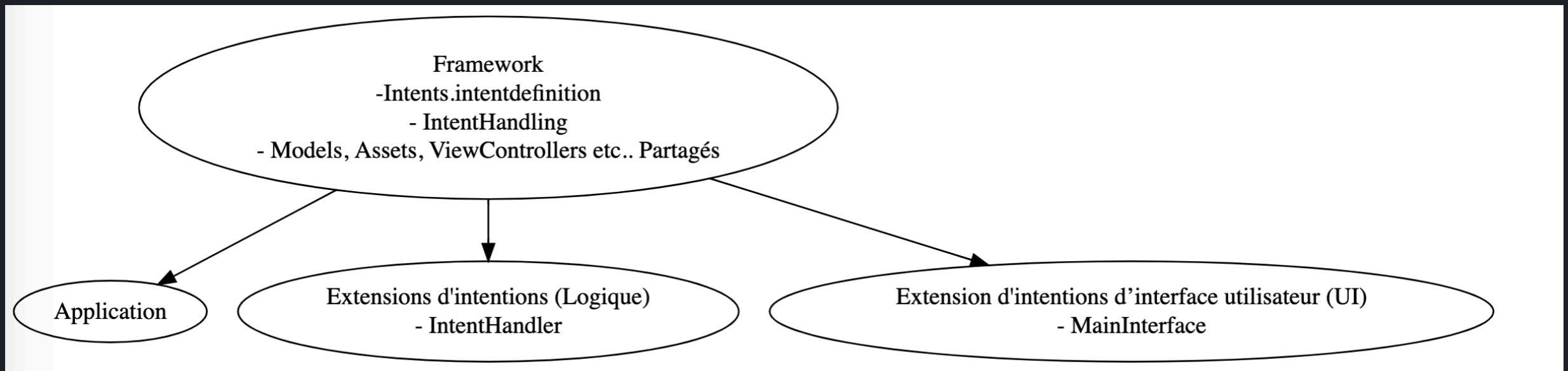
▼ Supported Intents

Class Name	
GoToFavoriteDirectionsIntent	
RequestAlertDeclarationIntent	
+	-

▼ Frameworks and Libraries

Name	Embed
IntentsSiriKit.framework	Do Not Embed ◊
IntentsUI.framework	Do Not Embed ◊
MapKit.framework	Do Not Embed ◊
+	-

Architecture Framework



Création de l'intents

