

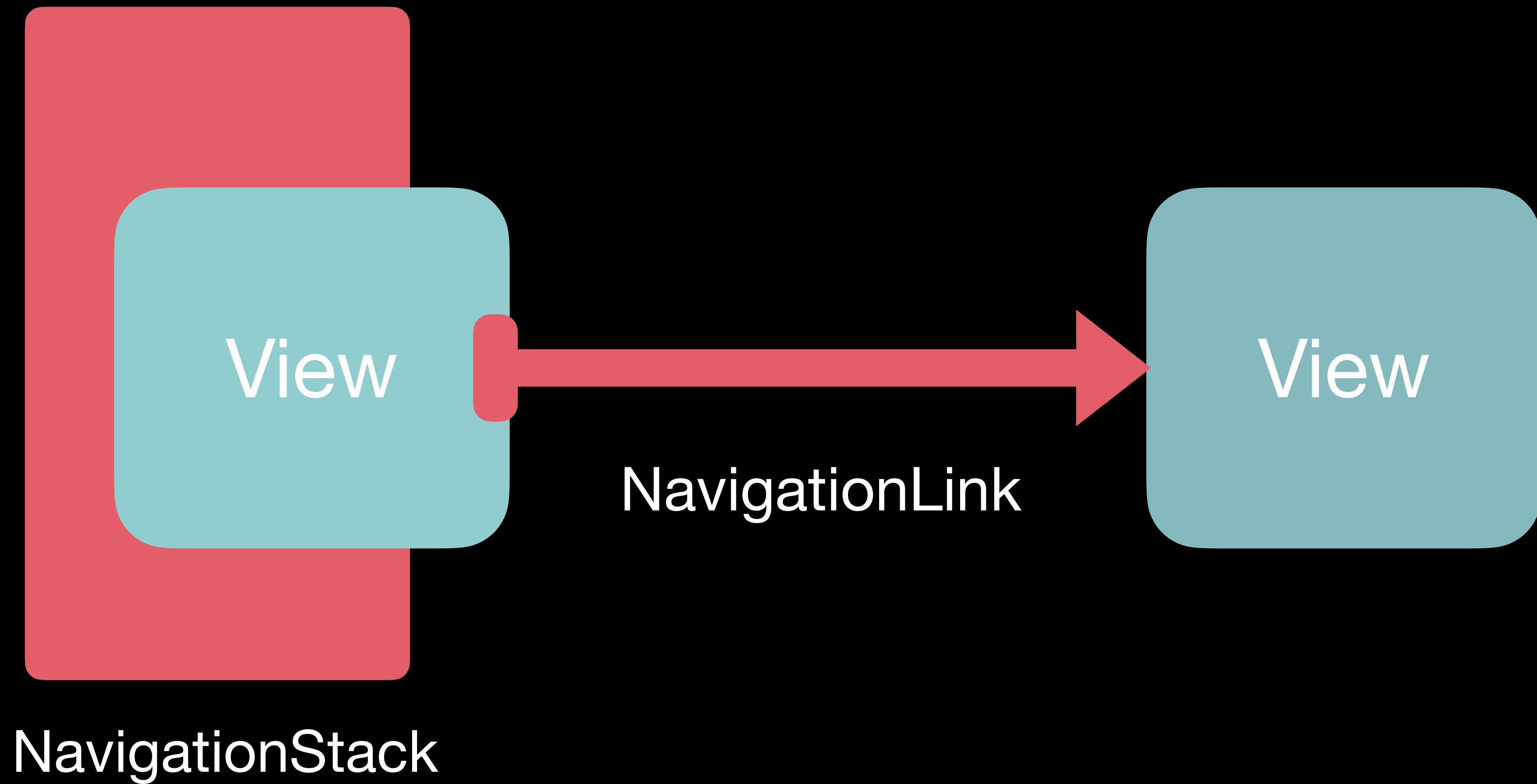
# Navigation

# Navigation

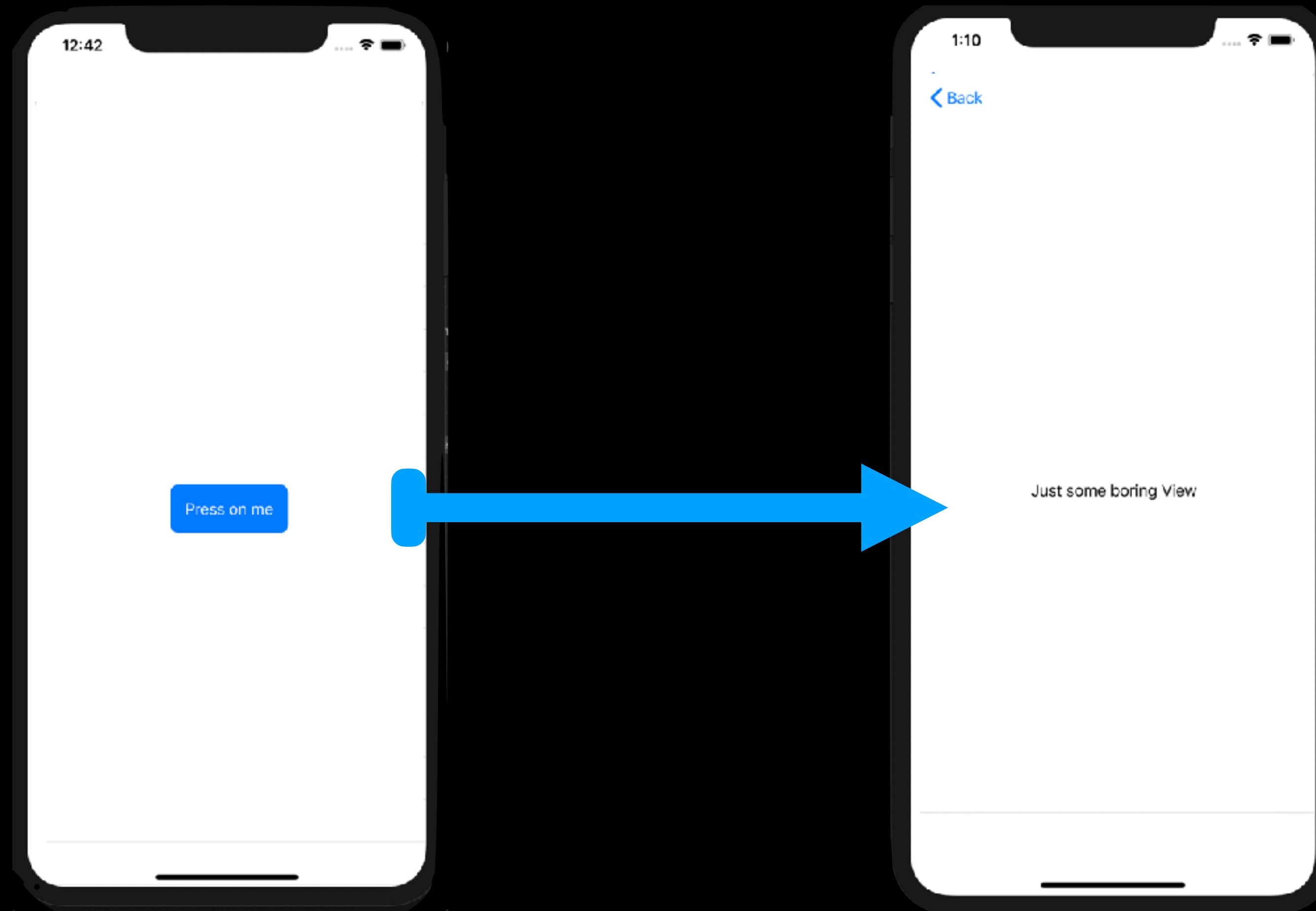
- C'est quoi ?



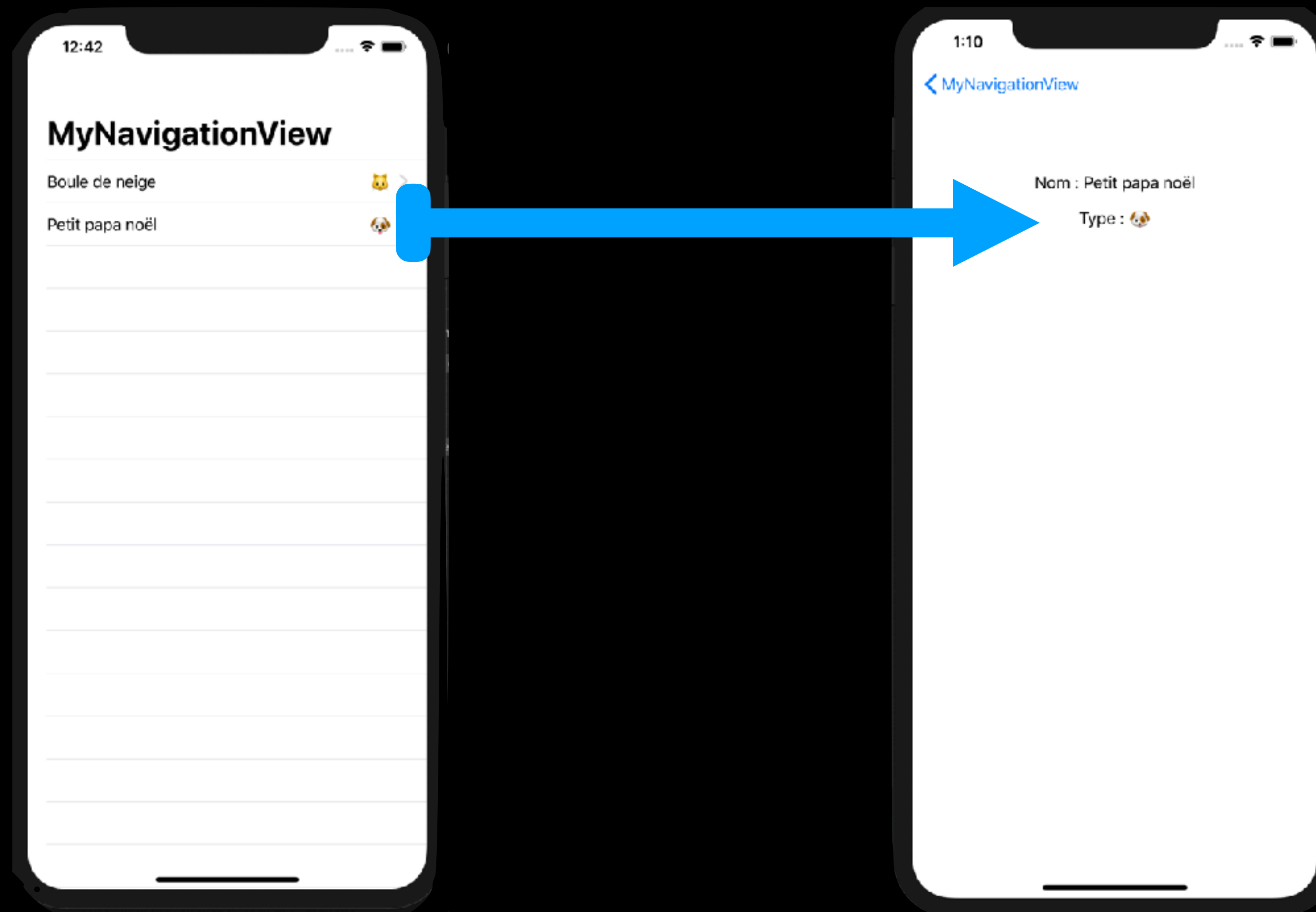
# Navigation



# Navigation



# Navigation

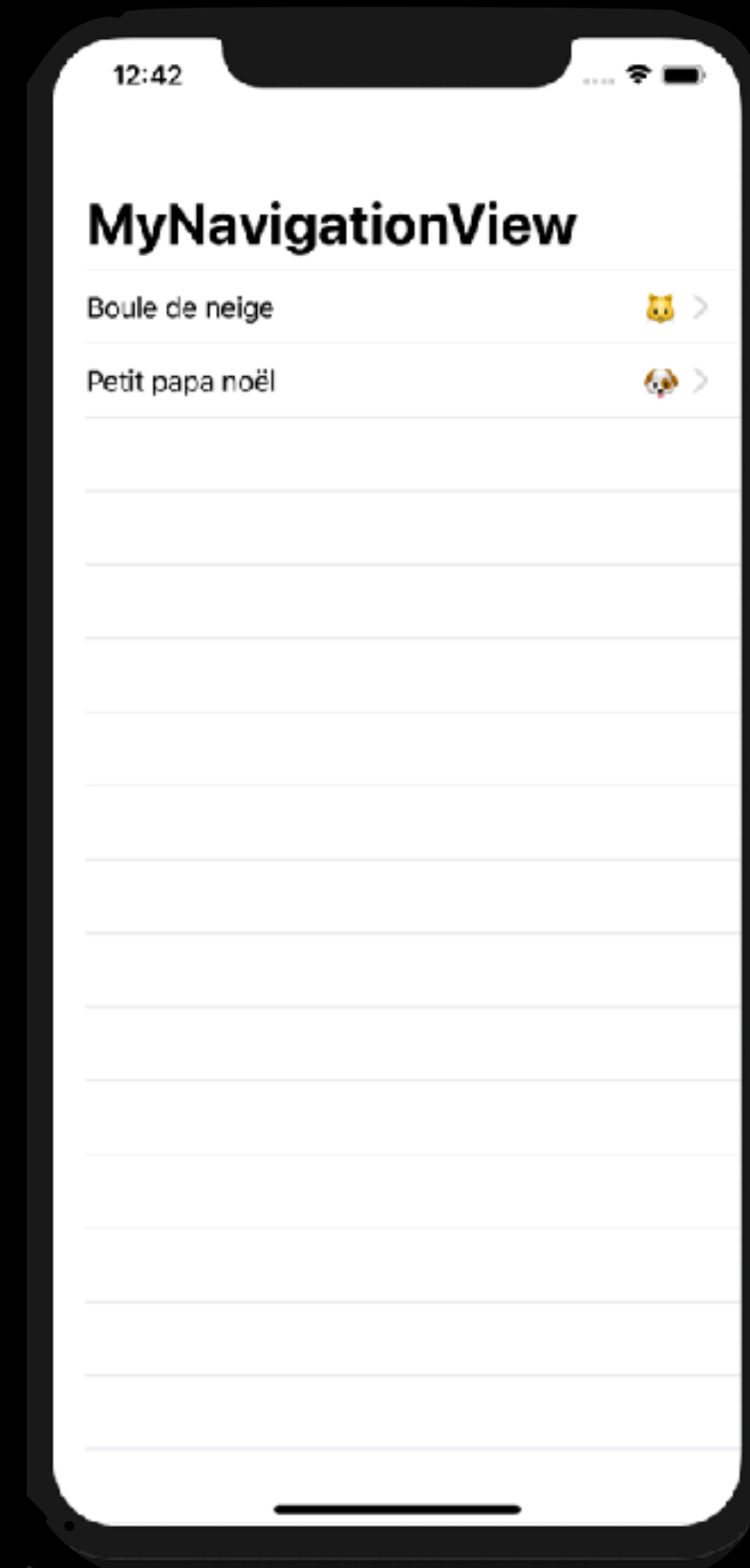


# Navigation hiérarchique

- On passe d'une vue à l'autre par un chemin
- Navigation Stack
- Navigation Link

# NavigationStack

Une vue pour présenter une stack (pile) de vues représentant un chemin dans une navigation hiérarchique.



# NavigationStack

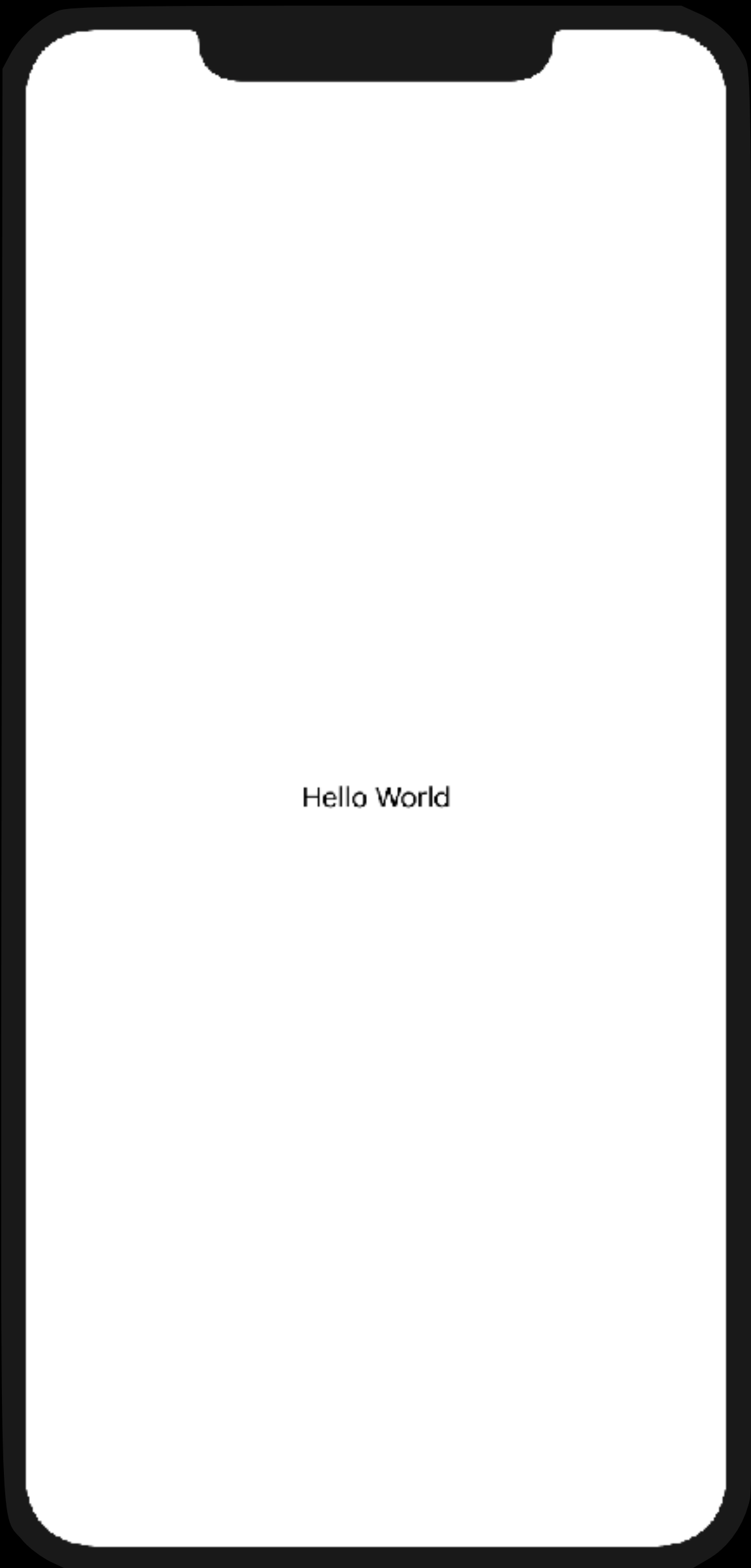
NavigationBar





# NavigationStack

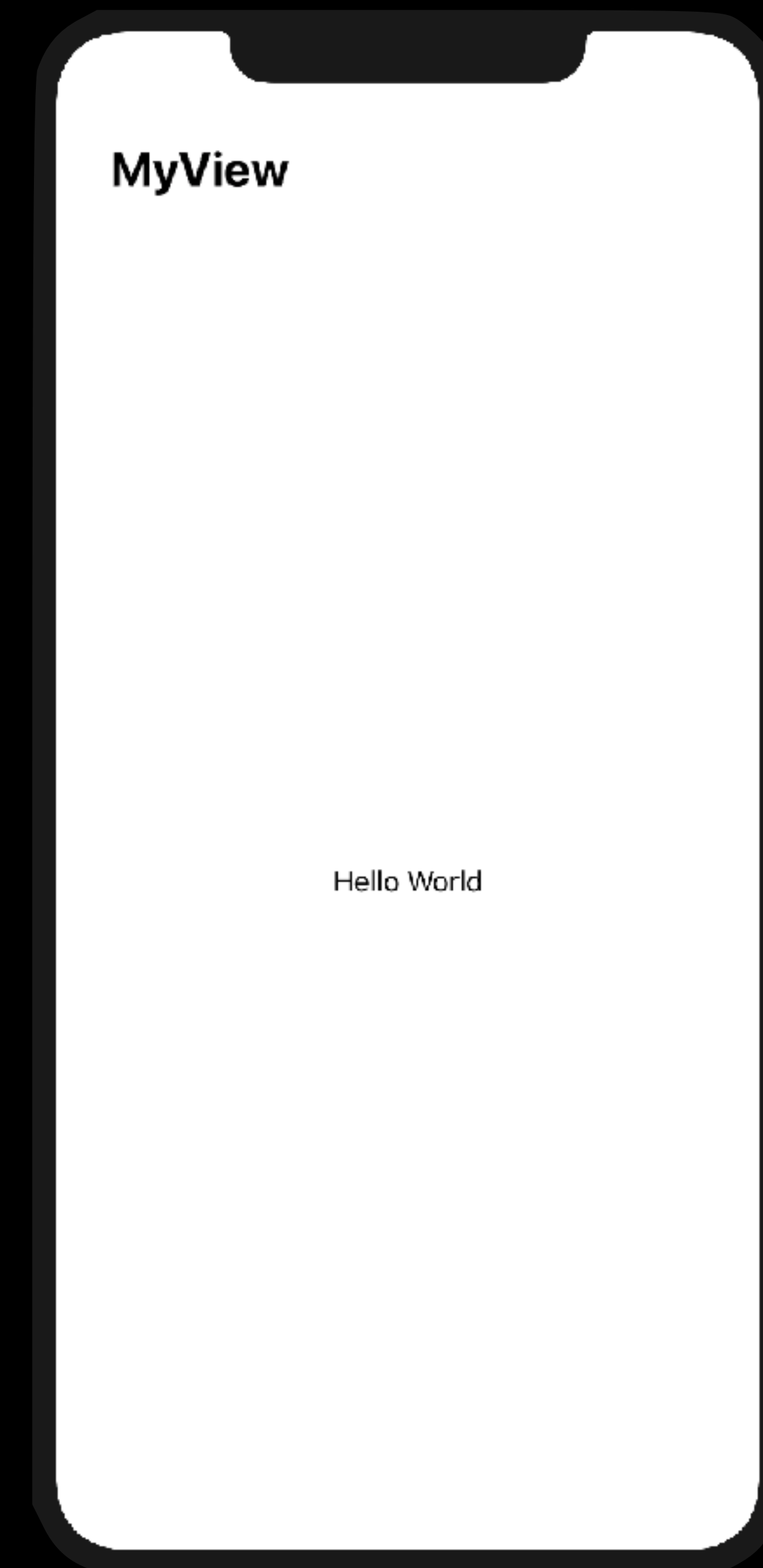
```
struct ContentView: View {  
    var body: some View {  
        Text("Hello, World!")  
    }  
}
```



Hello World

# NavigationStack

```
struct ContentView: View {  
    var body: some View {  
        NavigationStack {  
            Text("Hello, World!")  
                .navigationBarTitle("MyView")  
        }  
    }  
}
```



# NavLink

Une View qui déclenche une navigation lorsqu'on appuie dessus.

# NavLink

Une View

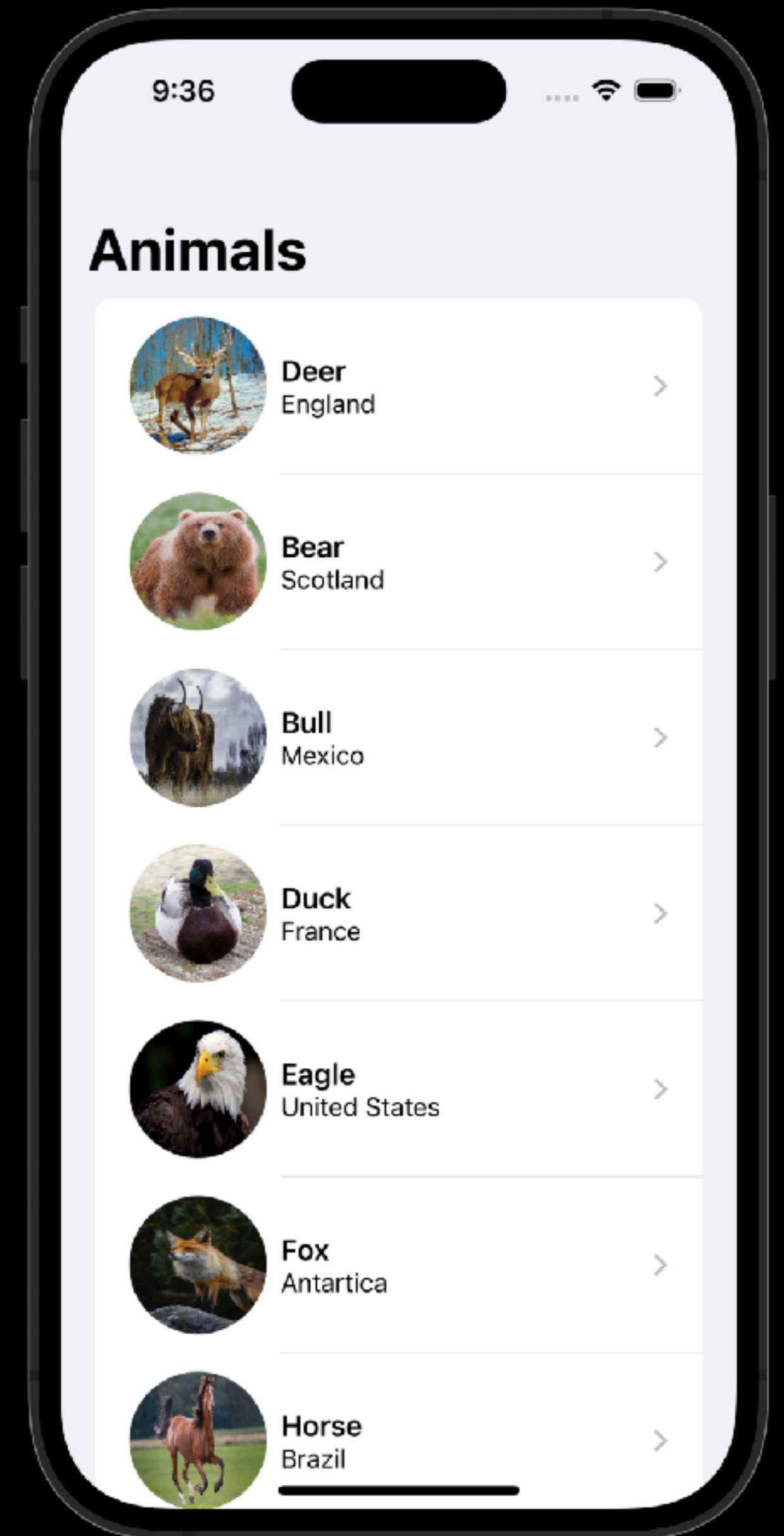
Image

List Row

Text

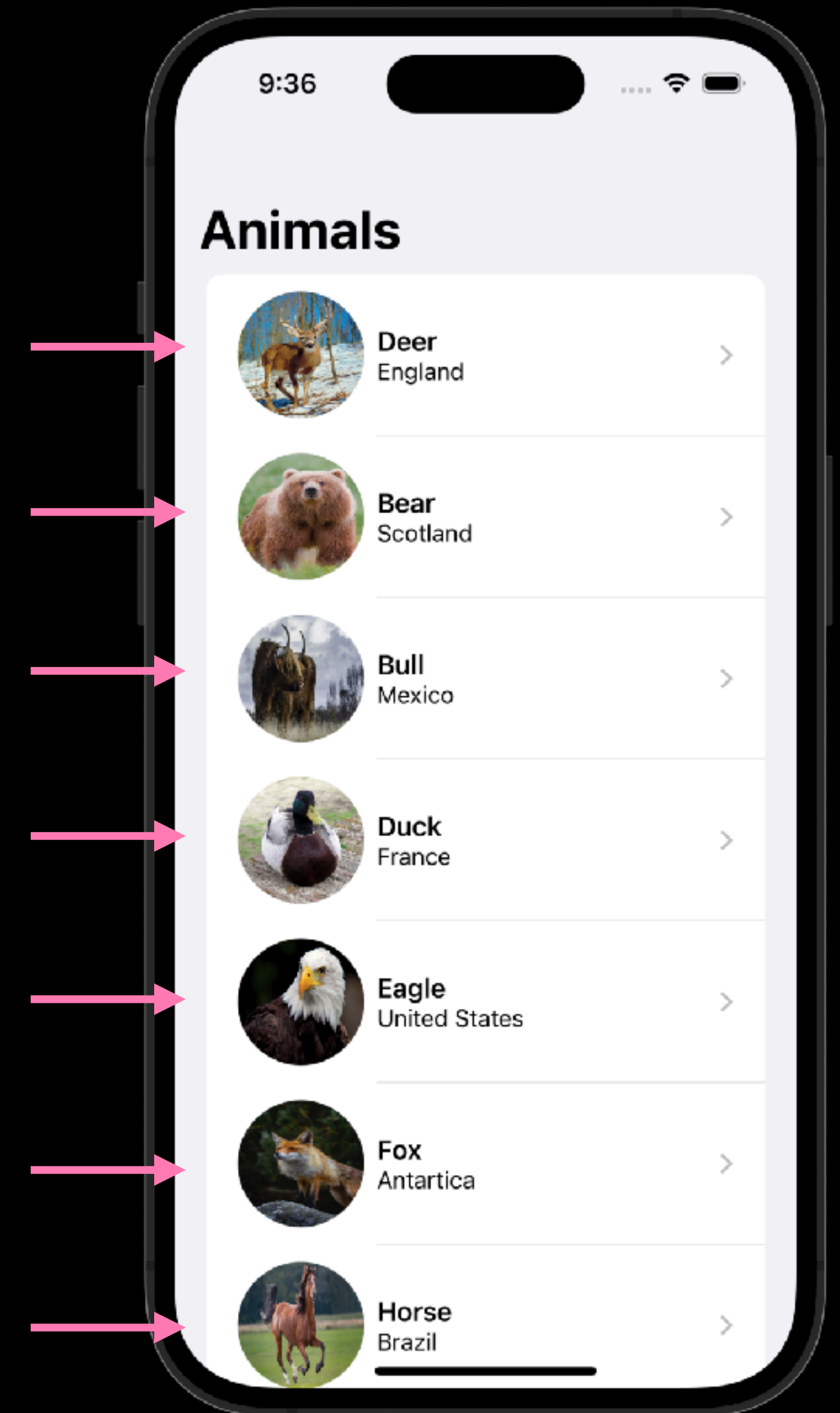
# NavLink

```
struct ListAnimals: View {  
    let animals: [Animal] = [ ... ]  
    var body: some View {  
        NavigationStack{  
            List(animals){ animal in  
                NavLink(destination: AnimalDetail()) {  
                    HStack{ ... }  
                }  
            }  
            .navigationTitle("Animals")  
        }  
    }  
}
```



# NavLink

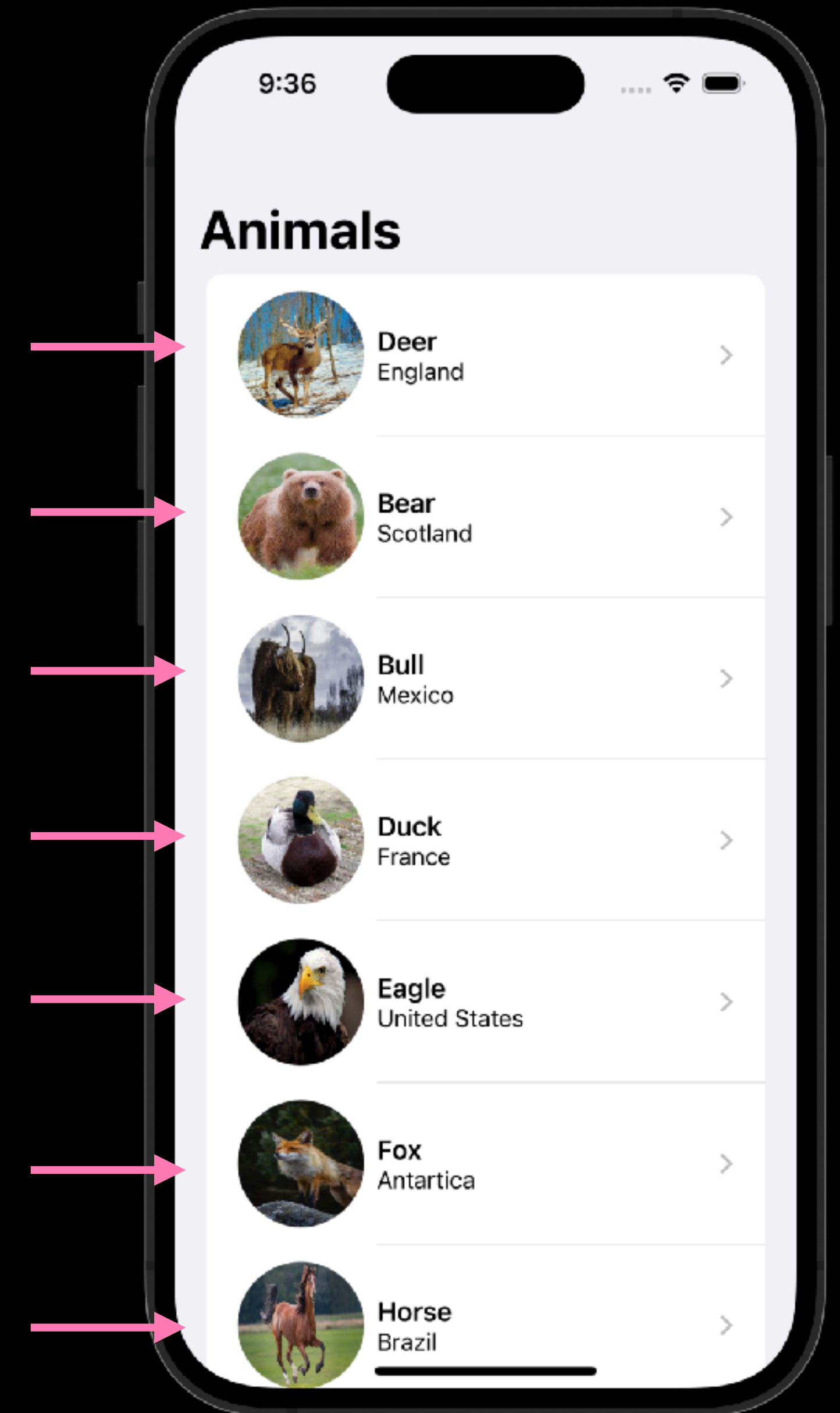
```
struct ListAnimals: View {  
    let animals: [Animal] = [ ... ]  
  
    var body: some View {  
        NavigationStack{  
            List(animals){ animal in  
                NavLink(destination: AnimalDetail()) {  
                    HStack{ ... }  
                }  
            }  
            .navigationTitle("Animals")  
        }  
    }  
}
```





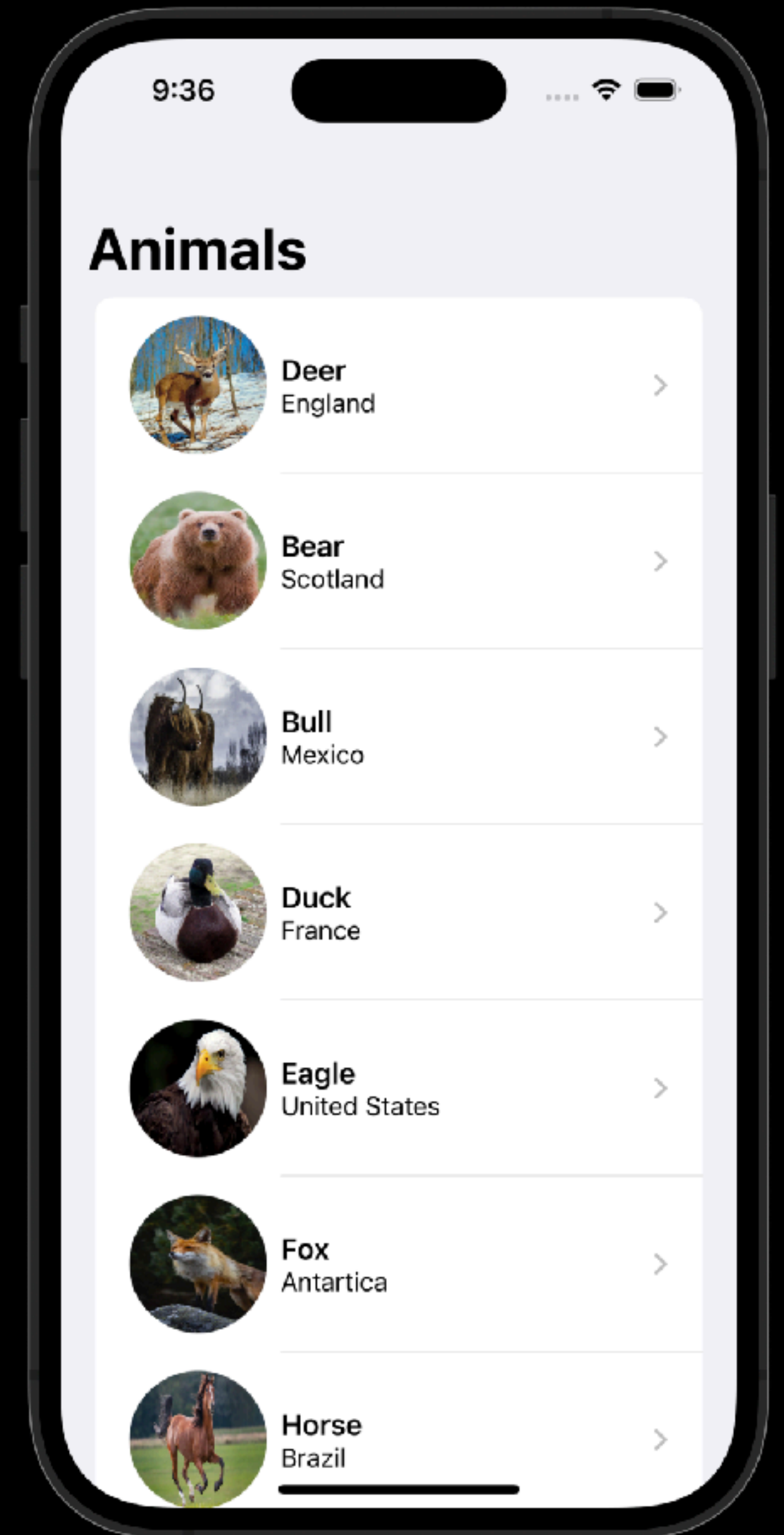
# NavLink

```
struct ListAnimals: View {  
    let animals: [Animal] = [ ... ]  
  
    var body: some View {  
        NavigationStack{  
            List(animals){ animal in  
                NavLink(destination: AnimalDetail()) {  
                    HStack{ ... }  
                }  
            }  
            .navigationTitle("Animals")  
        }  
    }  
}
```



# NavLink

```
struct ListAnimals: View {  
    let animals: [Animal] = [ ... ]  
    var body: some View {  
        NavigationStack{  
            List(animals){ animal in  
                NavLink(destination: AnimalDetail()) {  
                    HStack{ ... }  
                }  
            }  
            .navigationTitle("Animals")  
        }  
    }  
}
```



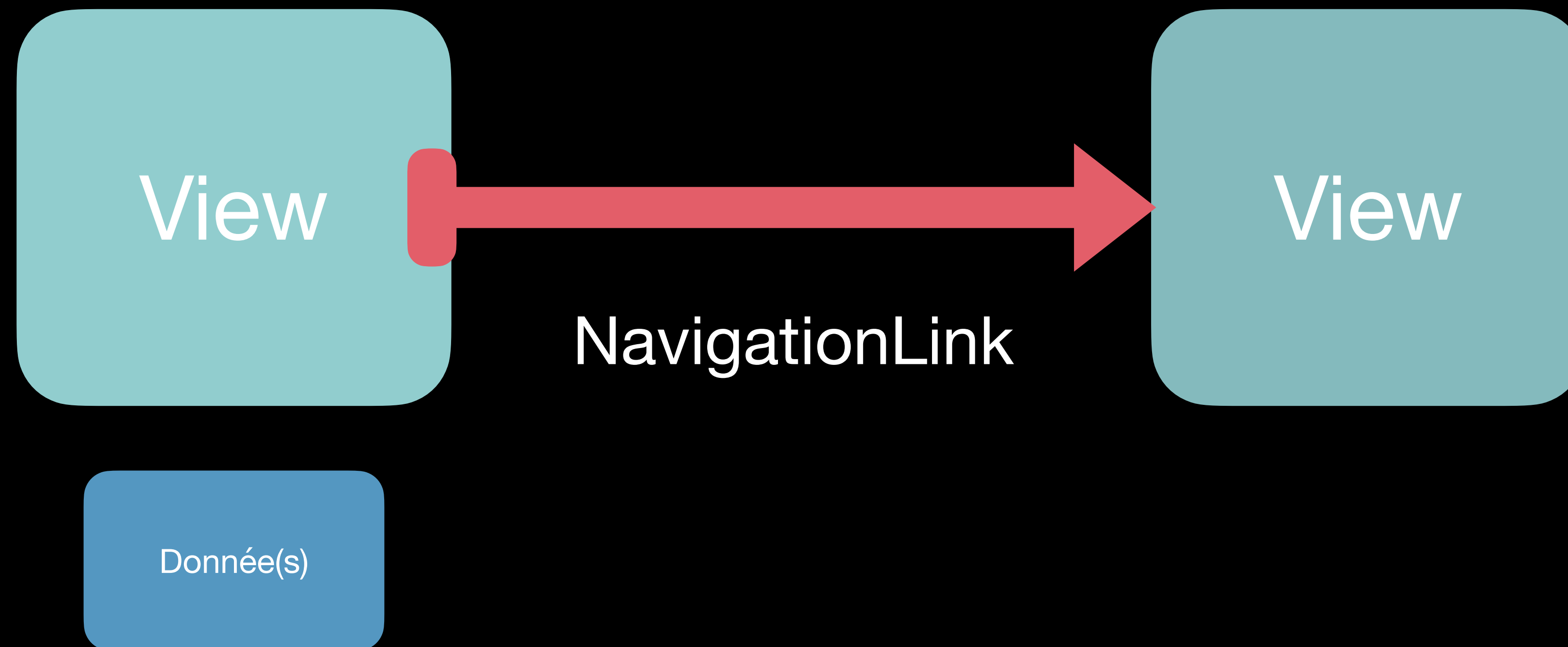


# NavigationLink

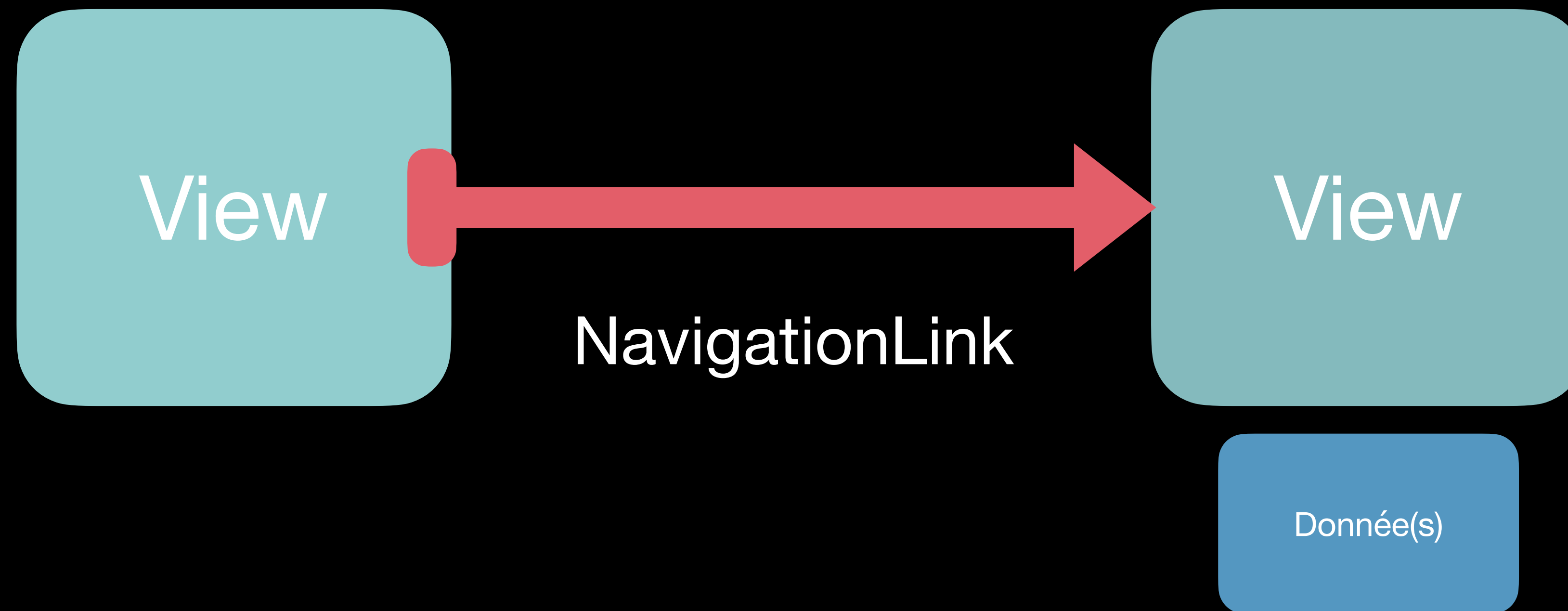
```
struct ListAnimals: View {  
    let animals: [Animal] = [ ... ]  
  
    var body: some View {  
        NavigationStack{  
            List(animals){ animal in  
                NavigationLink(destination: AnimalDetail()) {  
                    HStack{ ... }  
                }  
            }  
            .navigationTitle("Animals")  
        }  
    }  
}
```



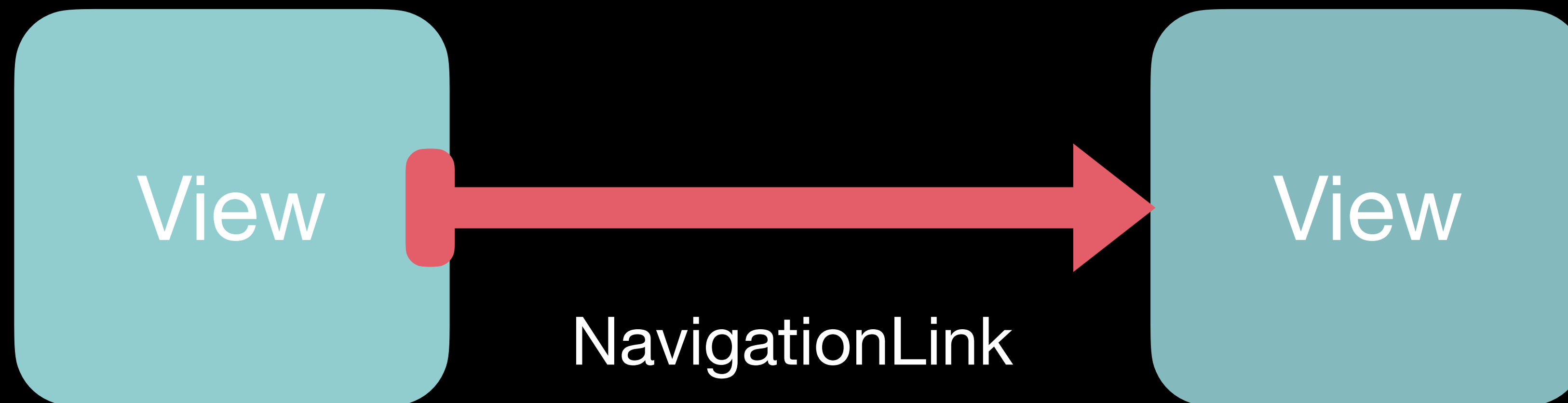
# Passage de données



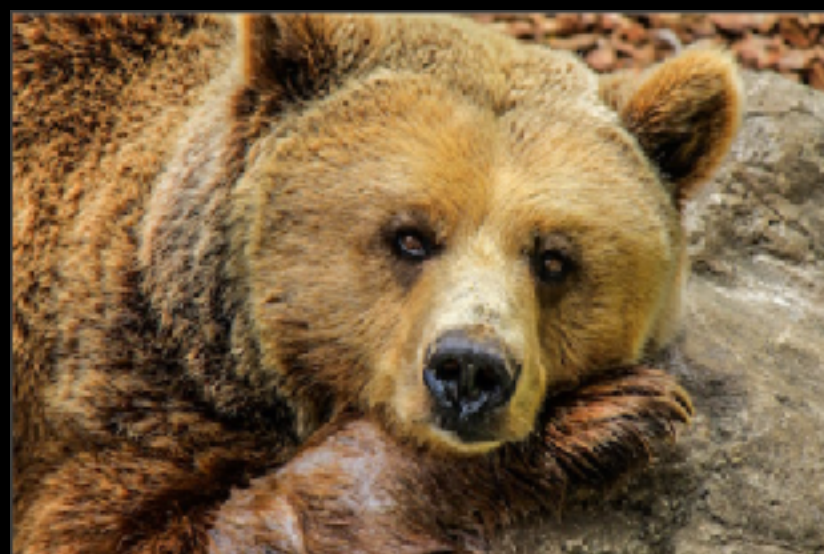
# Passage de données



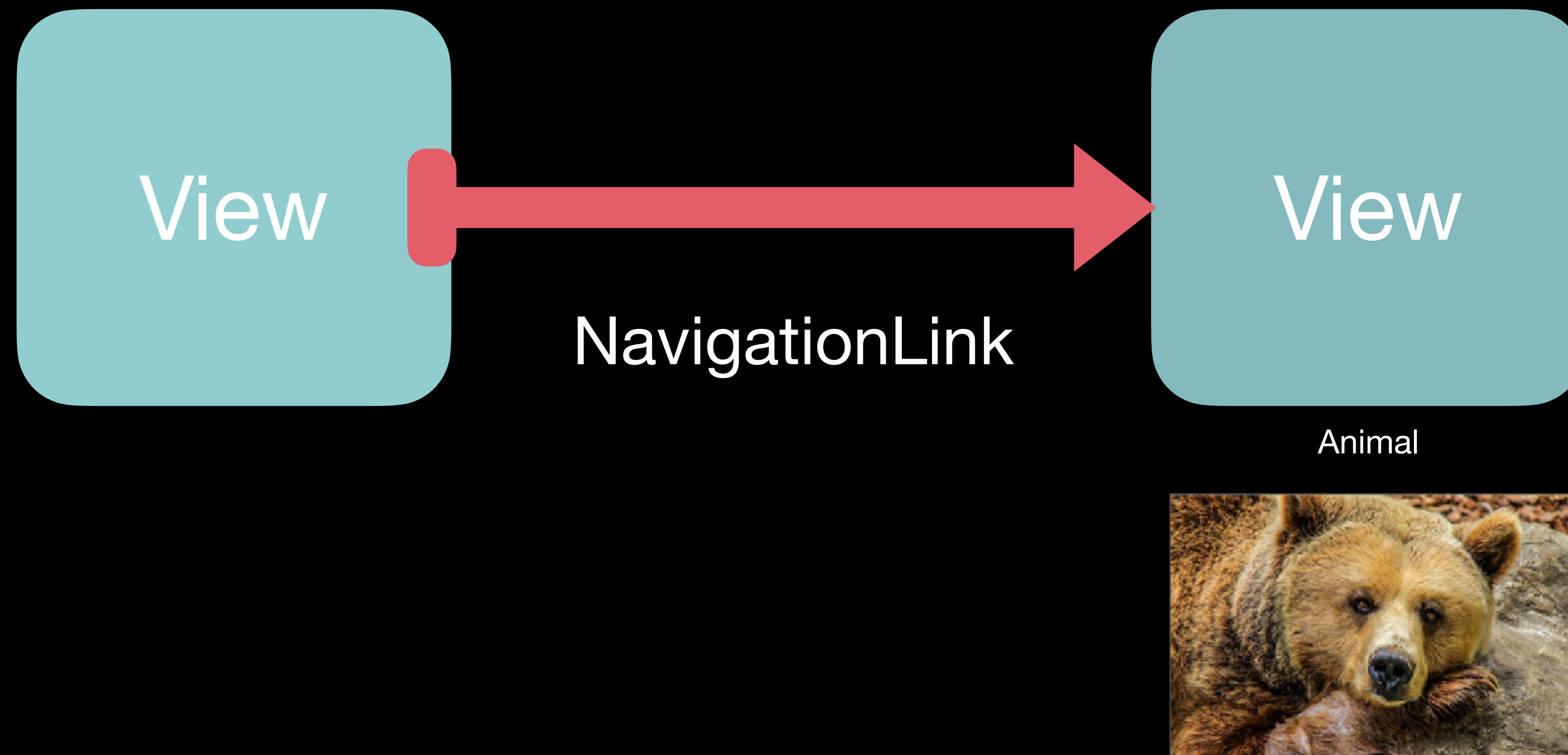
# Passage de données



Animal



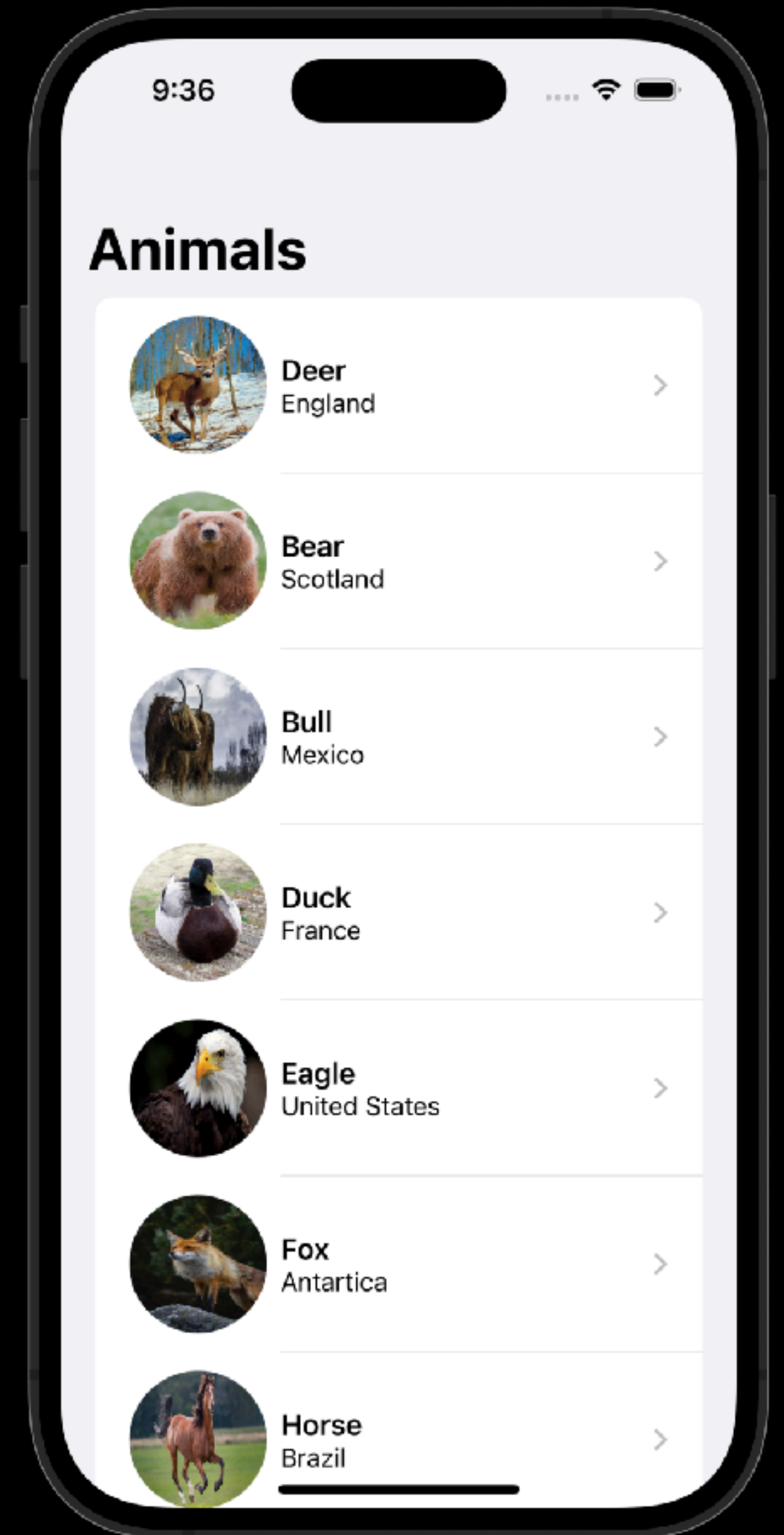
# Passage de données





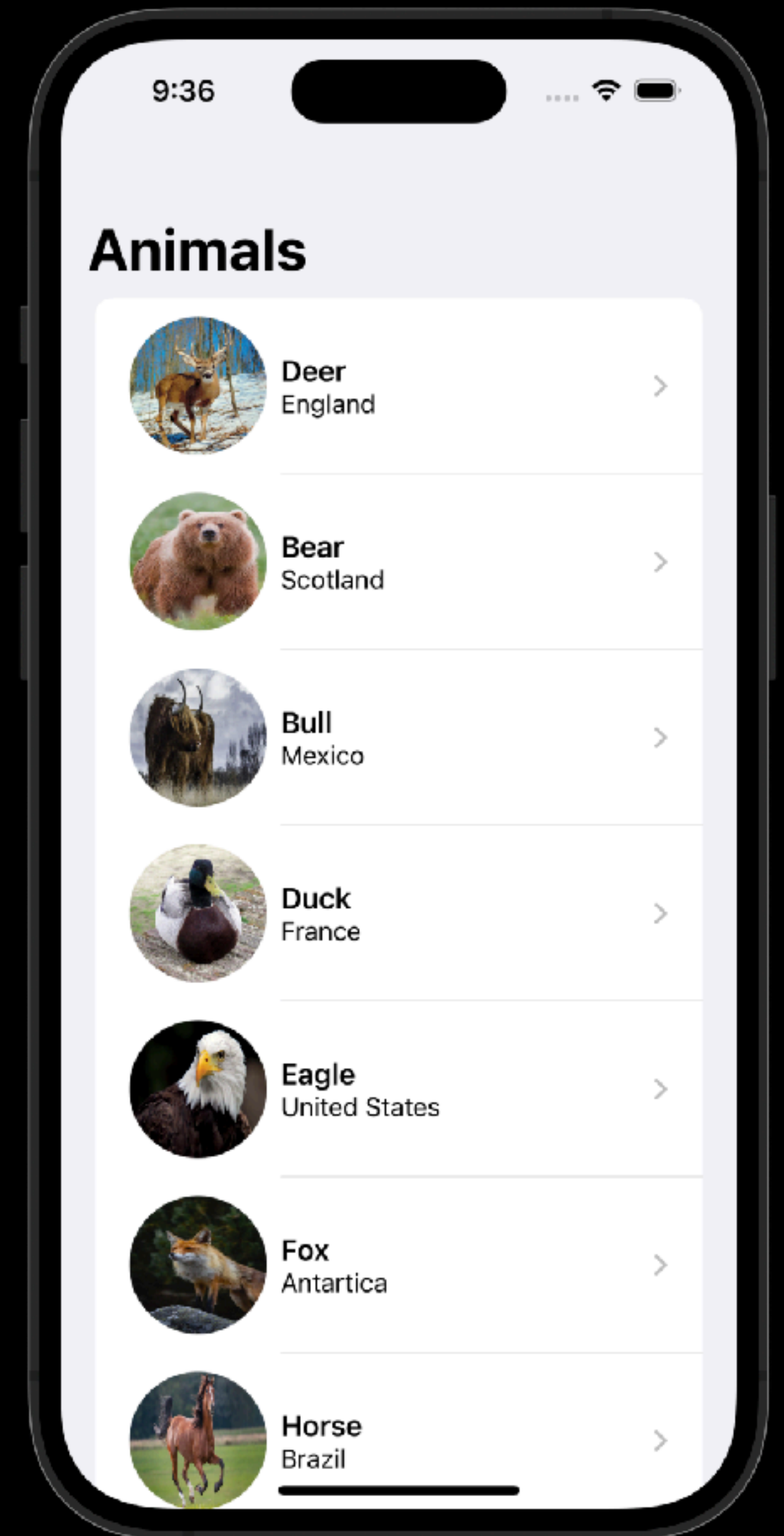
# NavLink

```
struct ListAnimals: View {  
    let animals: [Animal] = [ ... ]  
  
    var body: some View {  
        NavigationStack{  
            List(animals){ animal in  
                NavLink(destination: AnimalDetail(animal: animal)) {  
                    HStack{ ... }  
                }  
            }  
            .navigationTitle("Animals")  
        }  
    }  
}  
  
struct AnimalDetail: View {  
    var animal: Animal  
  
    var body: some View{ ... }  
}
```



# NavLink

```
struct ListAnimals: View {  
    let animals: [Animal] = [ ... ]  
  
    var body: some View {  
        NavigationStack{  
            List(animals){ animal in  
                NavLink(destination: AnimalDetail(animal: animal)) {  
                    HStack{ ... }  
                }  
            }  
            .navigationTitle("Animals")  
        }  
    }  
}  
  
struct AnimalDetail: View {  
    var animal: Animal  
  
    var body: some View{ ... }  
}
```



# NavLink

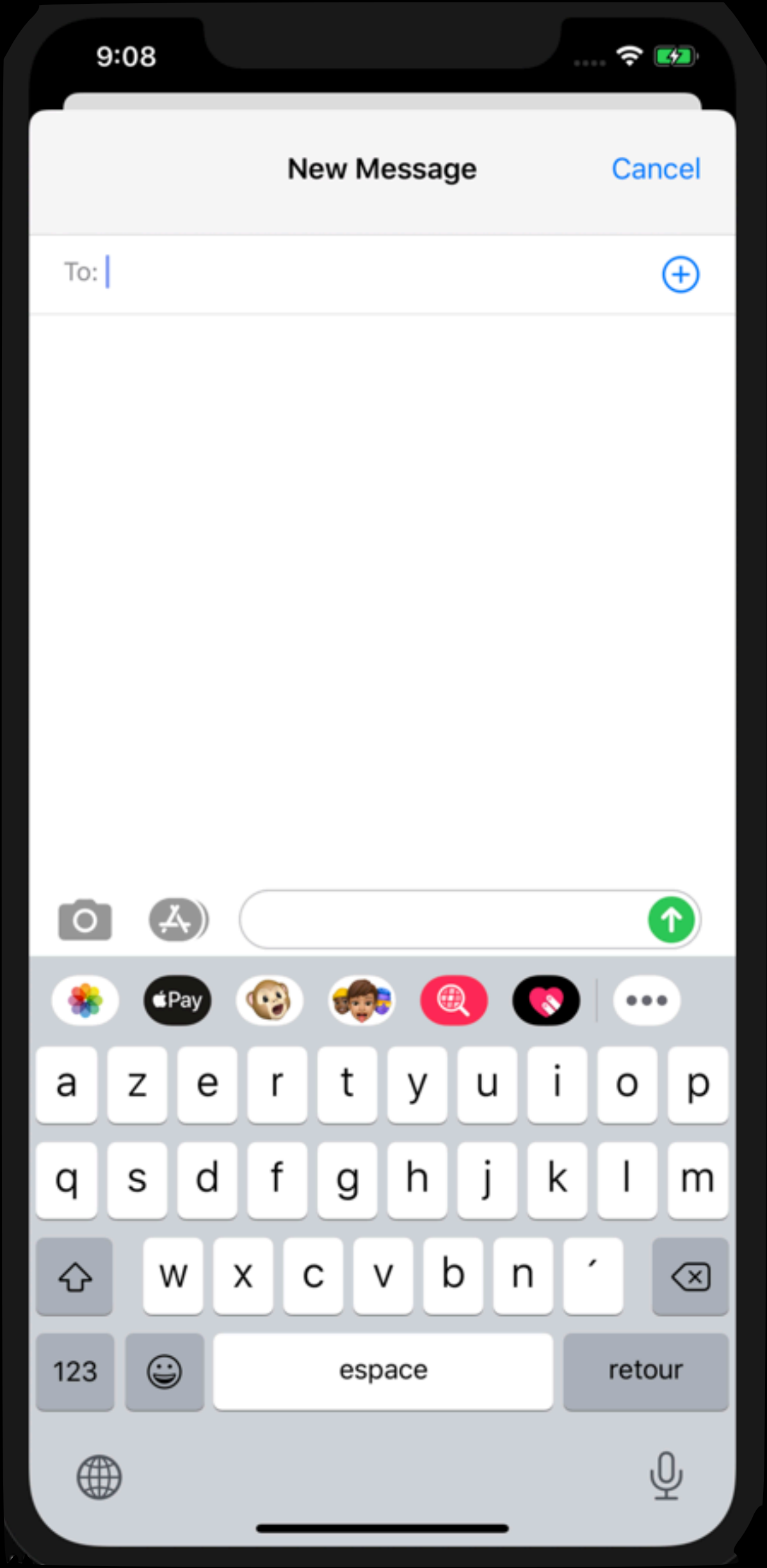
```
struct ListAnimals: View {  
    let animals: [Animal] = [ ... ]  
  
    var body: some View {  
        NavigationStack{  
            List(animals){ animal in  
                NavLink(destination: AnimalDetail(animal: animal)) {  
                    HStack{ ... }  
                }  
            }  
            .navigationTitle("Animals")  
        }  
    }  
}  
  
struct AnimalDetail: View {  
    var animal: Animal  
  
    var body: some View{ ... }  
}
```





# Navigation Modale

# Modale



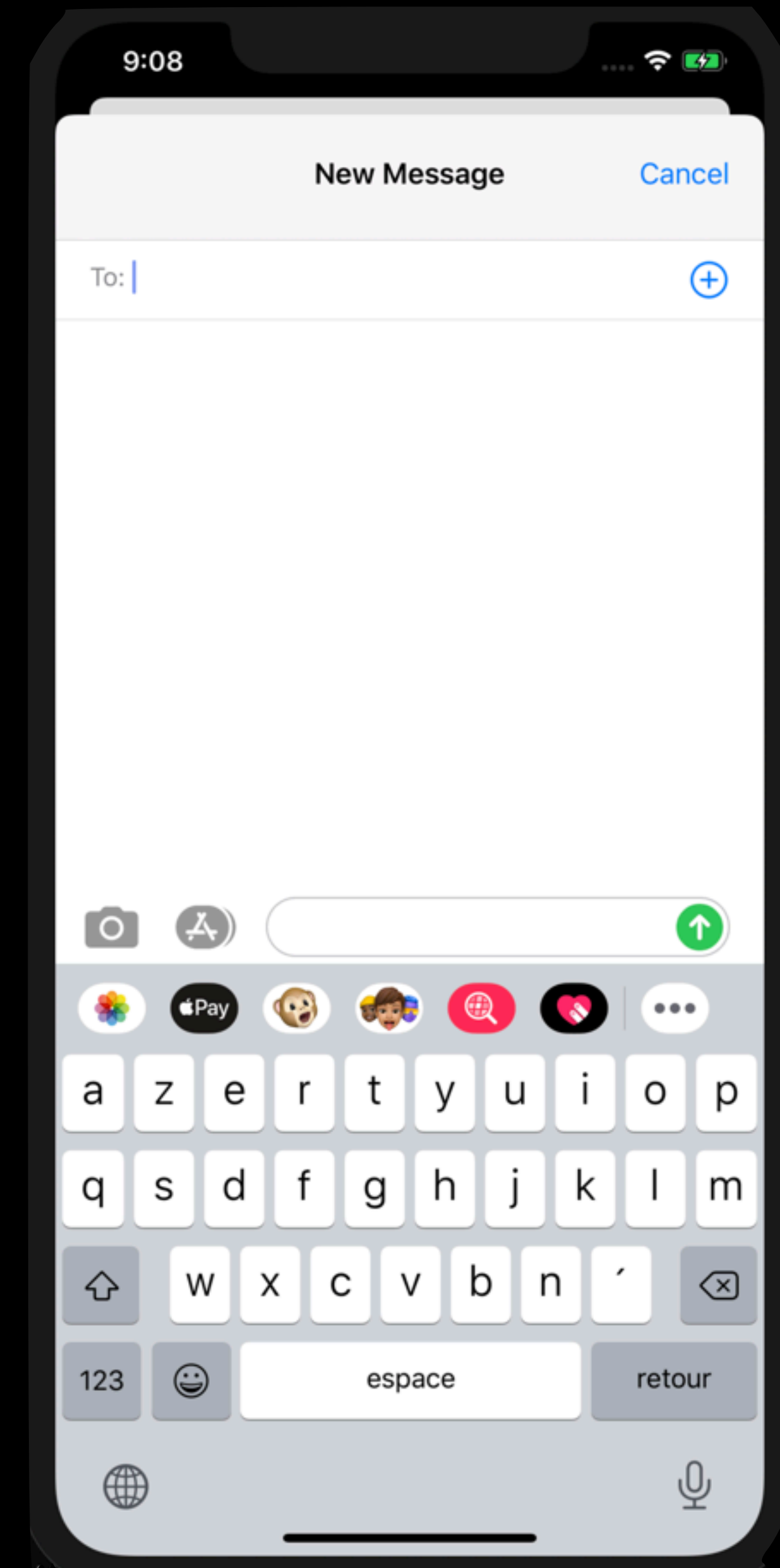
# Modale

Simple

Focus

Non lié au contexte initiale

Requiert une action pour en sortir



```
@State private var showAddView = false

var body: some View {
    NavigationView {
        List (listAnimals){ animal in
            NavigationLink(destination: AnimalDetail(animal: animal)){
                AnimalRow(data: animal)
            }
        }
        .sheet(isPresented: $showAddView){
            AnimalAdd()
        }
        .navigationBarTitle("Animals")
        .navigationBarItems(trailing: Button(action: {
            self.showAddView = true
            // self.showAddView.toggle()
        })){
            Image(systemName: "plus.circle.fill")
                .foregroundColor(.black)
                .font(Font.system(size: 25, weight: .medium))
        })
    }
}
```

```
@State private var showAddView = false
```

```
var body: some View {  
    NavigationView {  
        List (listAnimals){ animal in  
            NavigationLink(destination: AnimalDetail(animal: animal)){  
                AnimalRow(data: animal)  
            }  
        }  
        .sheet(isPresented: $showAddView){  
            AnimalAdd()  
        }  
        .navigationBarTitle("Animals")  
        .navigationBarItems(trailing: Button(action: {  
            self.showAddView = true  
            // self.showAddView.toggle()  
        })){  
            Image(systemName: "plus.circle.fill")  
                .foregroundColor(.black)  
                .font(Font.system(size: 25, weight: .medium))  
        })  
    }  
}
```



```
@State private var showAddView = false
```

```
var body: some View {  
    NavigationView {  
        List (listAnimals){ animal in  
            NavigationLink(destination: AnimalDetail(animal: animal)){  
                AnimalRow(data: animal)  
            }  
        }  
        .sheet(isPresented: $showAddView){  
            AnimalAdd()  
        }  
        .navigationBarTitle("Animals")  
        .navigationBarItems(trailing: Button(action: {  
            self.showAddView = true  
            // self.showAddView.toggle()  
        })){  
            Image(systemName: "plus.circle.fill")  
                .foregroundColor(.black)  
                .font(Font.system(size: 25, weight: .medium))  
        })  
    }  
}
```

```
@State private var showAddView = false
```

```
var body: some View {  
    NavigationView {  
        List (listAnimals){ animal in  
            NavigationLink(destination: AnimalDetail(animal: animal)){  
                AnimalRow(data: animal)  
            }  
        }  
        .sheet(isPresented: $showAddView){  
            AnimalAdd()  
        }  
        .navigationBarTitle("Animals")  
        .navigationBarItems(trailing: Button(action: {  
            self.showAddView = true  
            // self.showAddView.toggle()  
        })){  
            Image(systemName: "plus.circle.fill")  
                .foregroundColor(.black)  
                .font(Font.system(size: 25, weight: .medium))  
        })  
    }  
}
```

```
@State private var showAddView = false
```

```
var body: some View {  
    NavigationView {
```

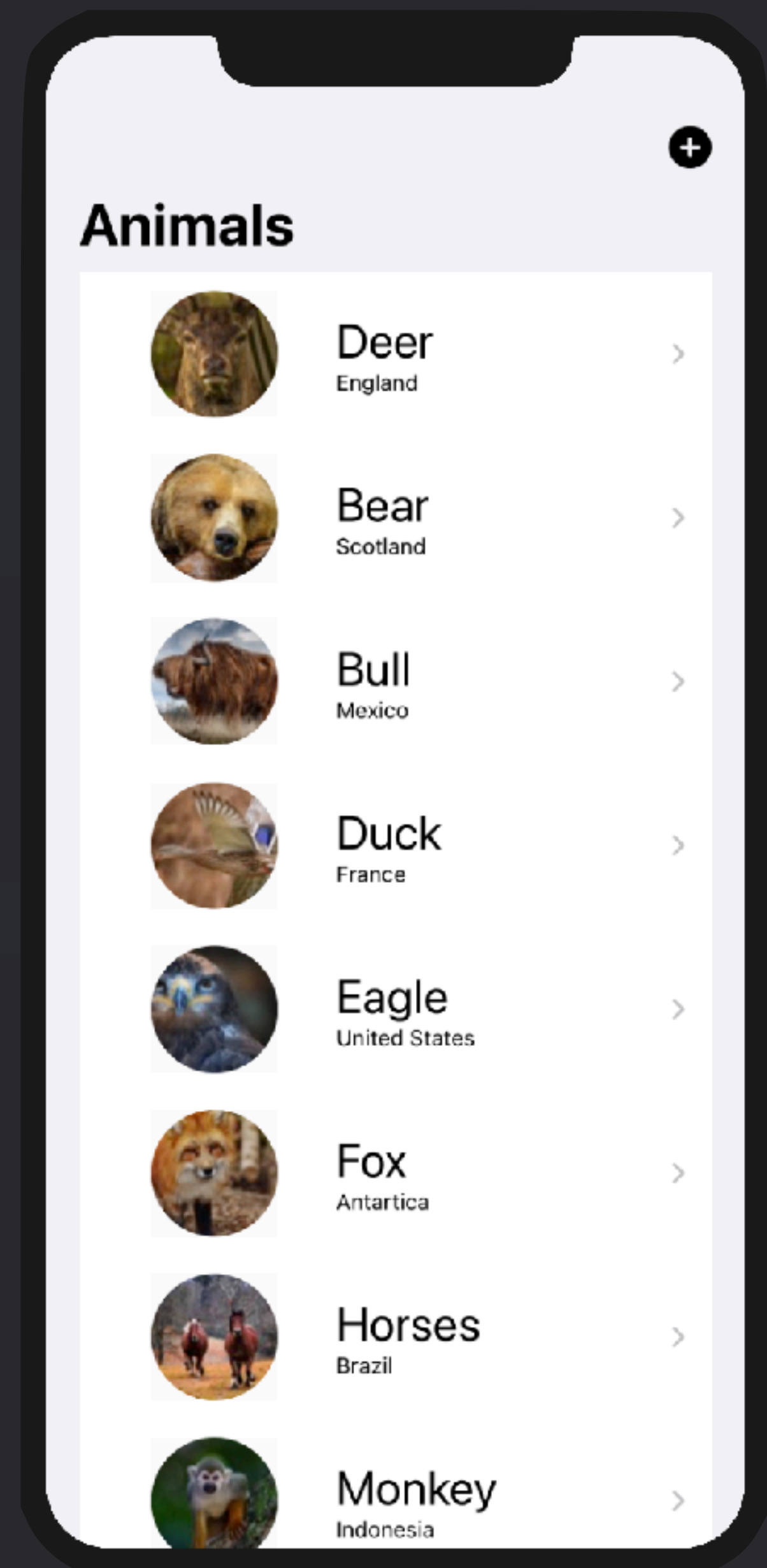
```
    }
```

```
    .sheet(isPresented: $showAddView){  
        AnimalAdd()  
    }
```

```
    .navigationBarItems(trailing: Button(action: {  
        self.showAddView = true  
        // self.showAddView.toggle()  
    })){  
        Image(systemName: "plus.circle.fill")  
            .foregroundColor(.black)  
            .font(Font.system(size: 25, weight: .medium))  
    })  
}
```

```
}
```

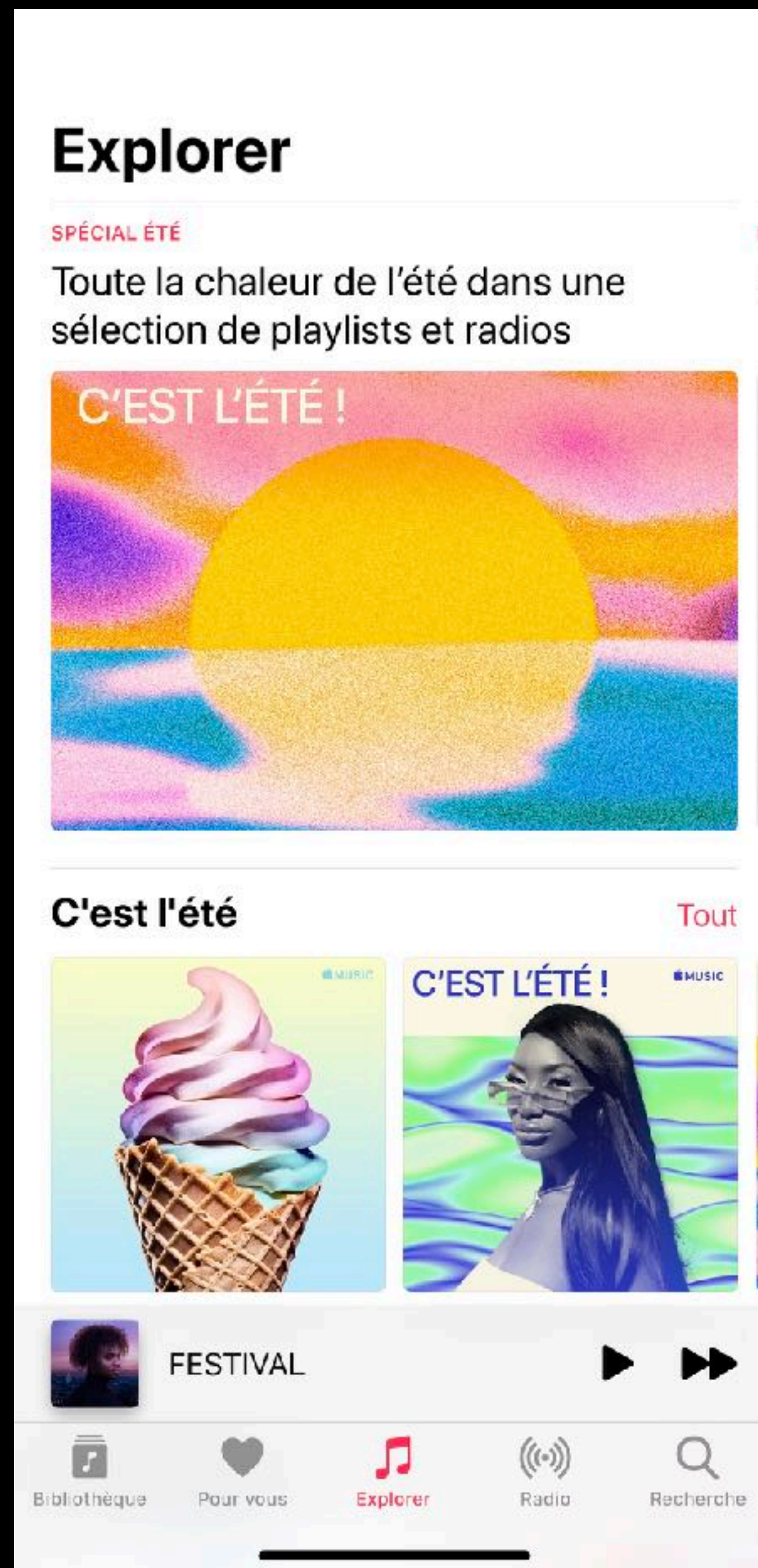
```
}
```





**One more thing ...**

# TabView



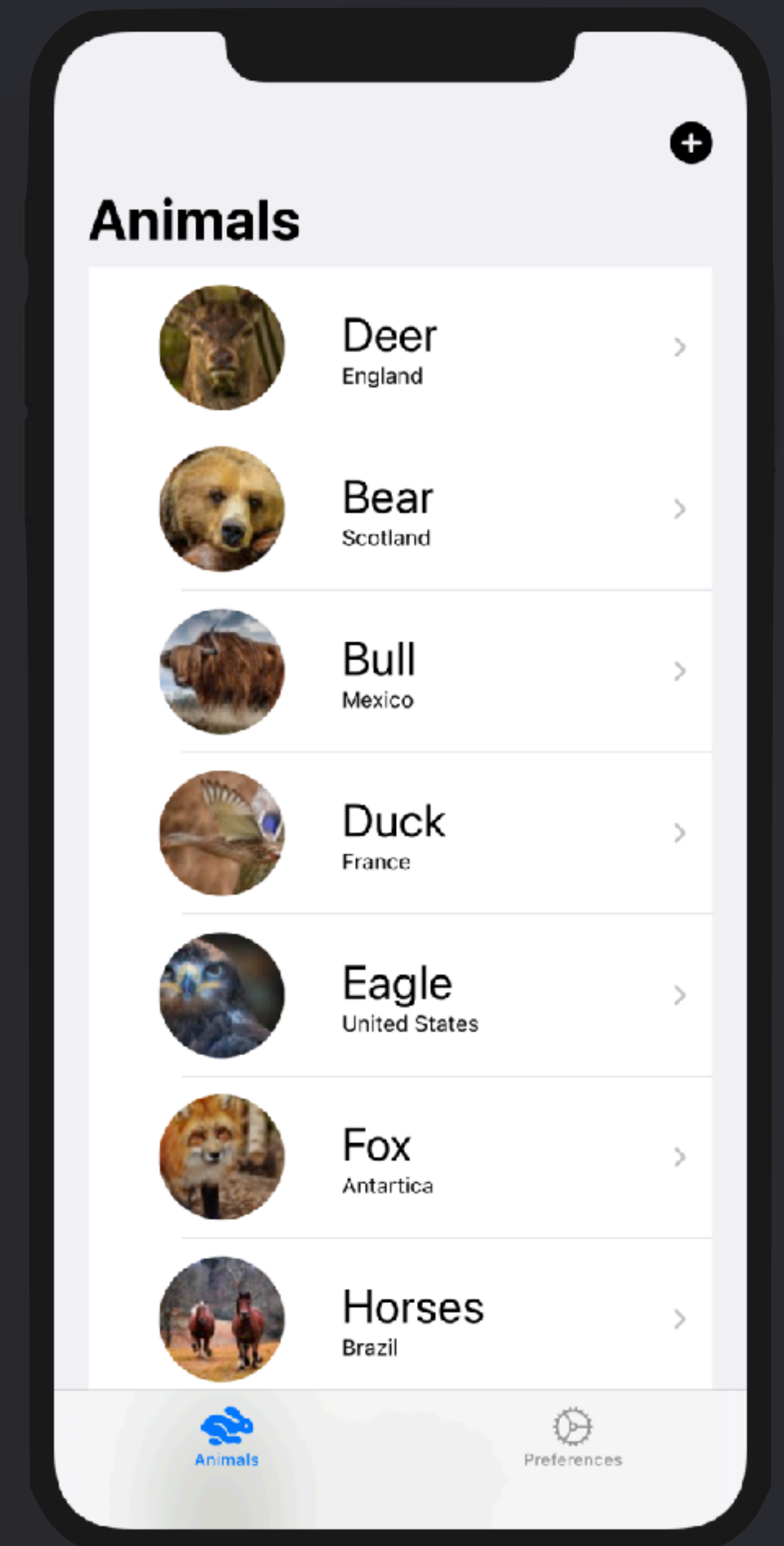
```
struct TabViewDemo: View {
    var body: some View {
        TabView{
            ListAnimals()
                .tabItem {
                    Image(systemName: "hare.fill")
                    Text("Animals")
                }
            Preferences()
                .tabItem {
                    Image(systemName: "gear")
                    Text("Preferences")
                }
        }
    }
}
```



```

struct TabViewDemo: View {
    var body: some View {
        TabView{
            ListAnimals()
                .tabItem {
                    Image(systemName: "hare.fill")
                    Text("Animals")
                }
            Preferences()
                .tabItem {
                    Image(systemName: "gear")
                    Text("Preferences")
                }
        }
    }
}

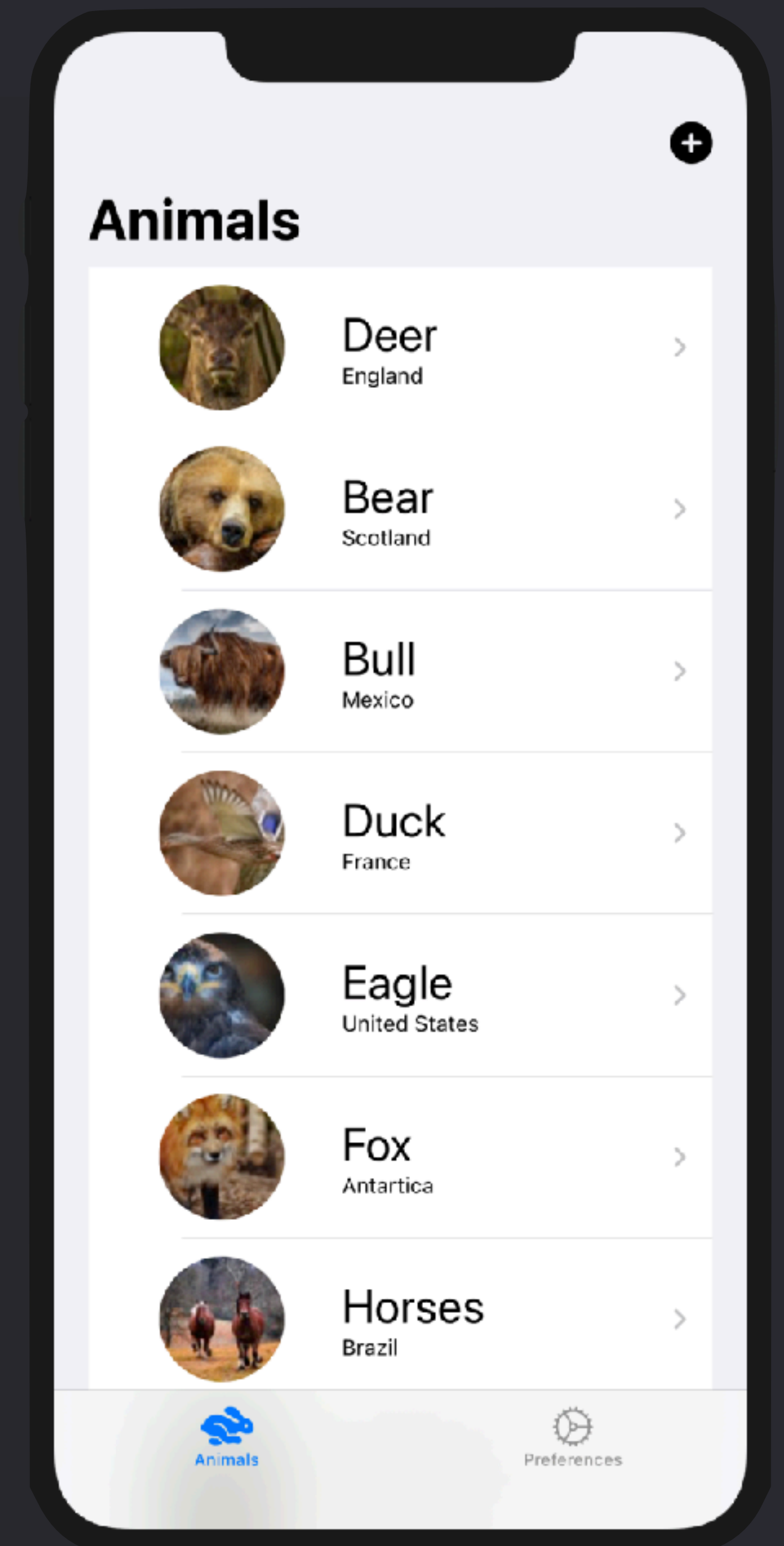
```



```

struct TabViewDemo: View {
    var body: some View {
        TabView{
            ListAnimals()
                .tabItem {
                    Image(systemName: "hare.fill")
                    Text("Animals")
                }
            Preferences()
                .tabItem {
                    Image(systemName: "gear")
                    Text("Preferences")
                }
        }
    }
}

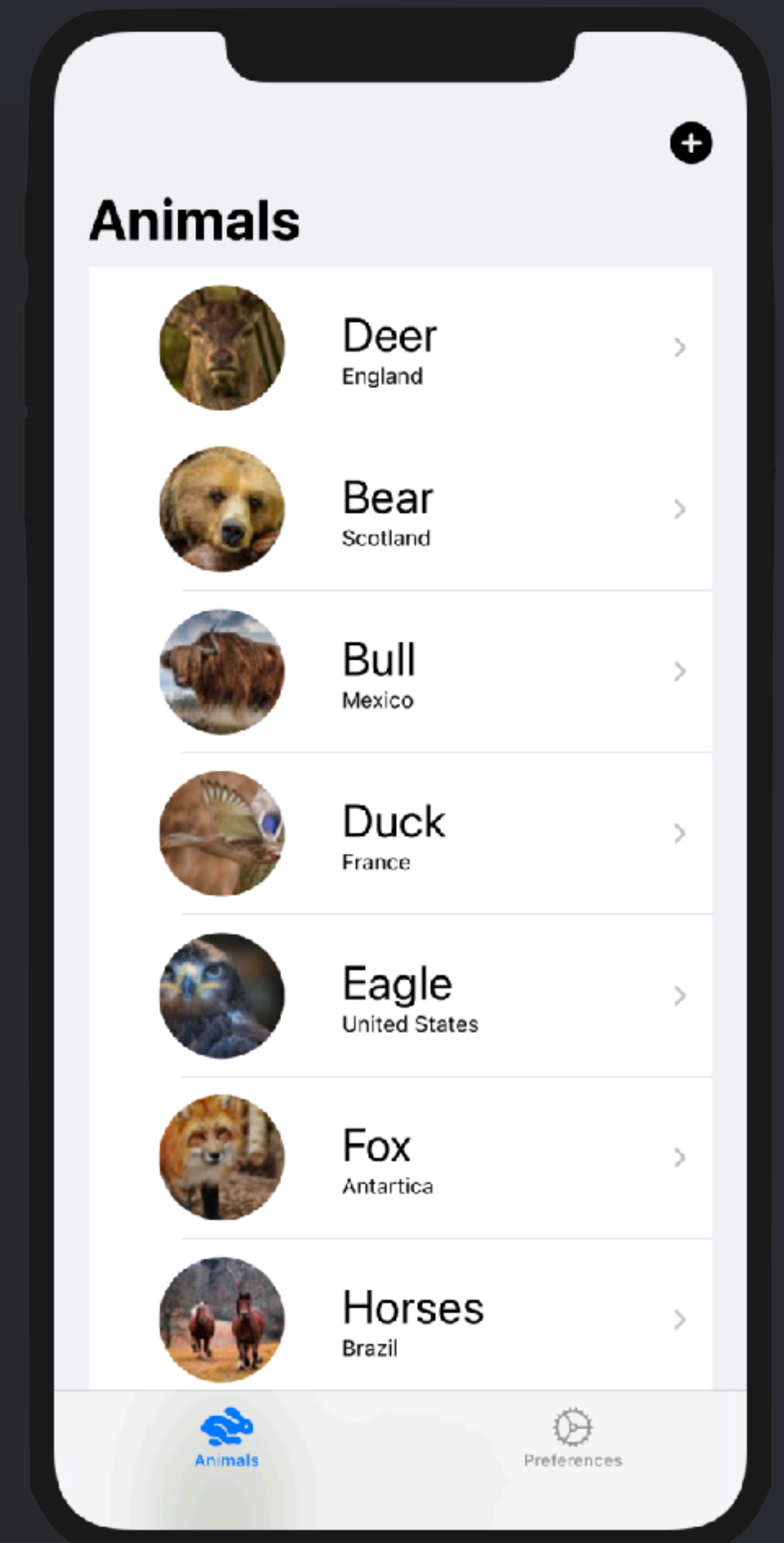
```



```

struct TabViewDemo: View {
    var body: some View {
        TabView{
            ListAnimals()
            .tabItem {
                Image(systemName: "hare.fill")
                Text("Animals")
            }
            Preferences()
            .tabItem {
                Image(systemName: "gear")
                Text("Preferences")
            }
        }
    }
}

```



# Weather list with Detail

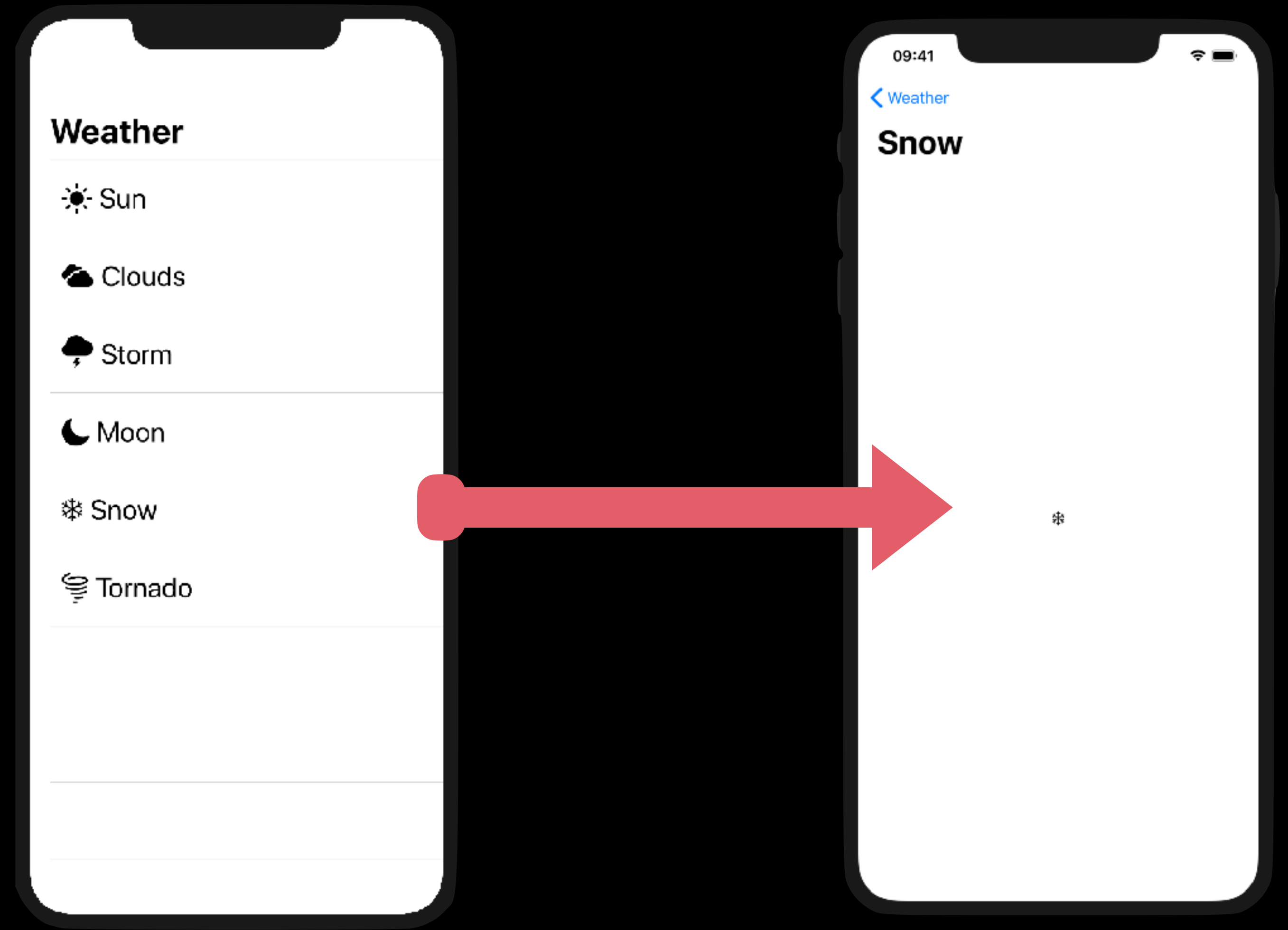
List

ExtractingView

NavigationStack

NavigationLink

Struct





# Weather list Sheet

