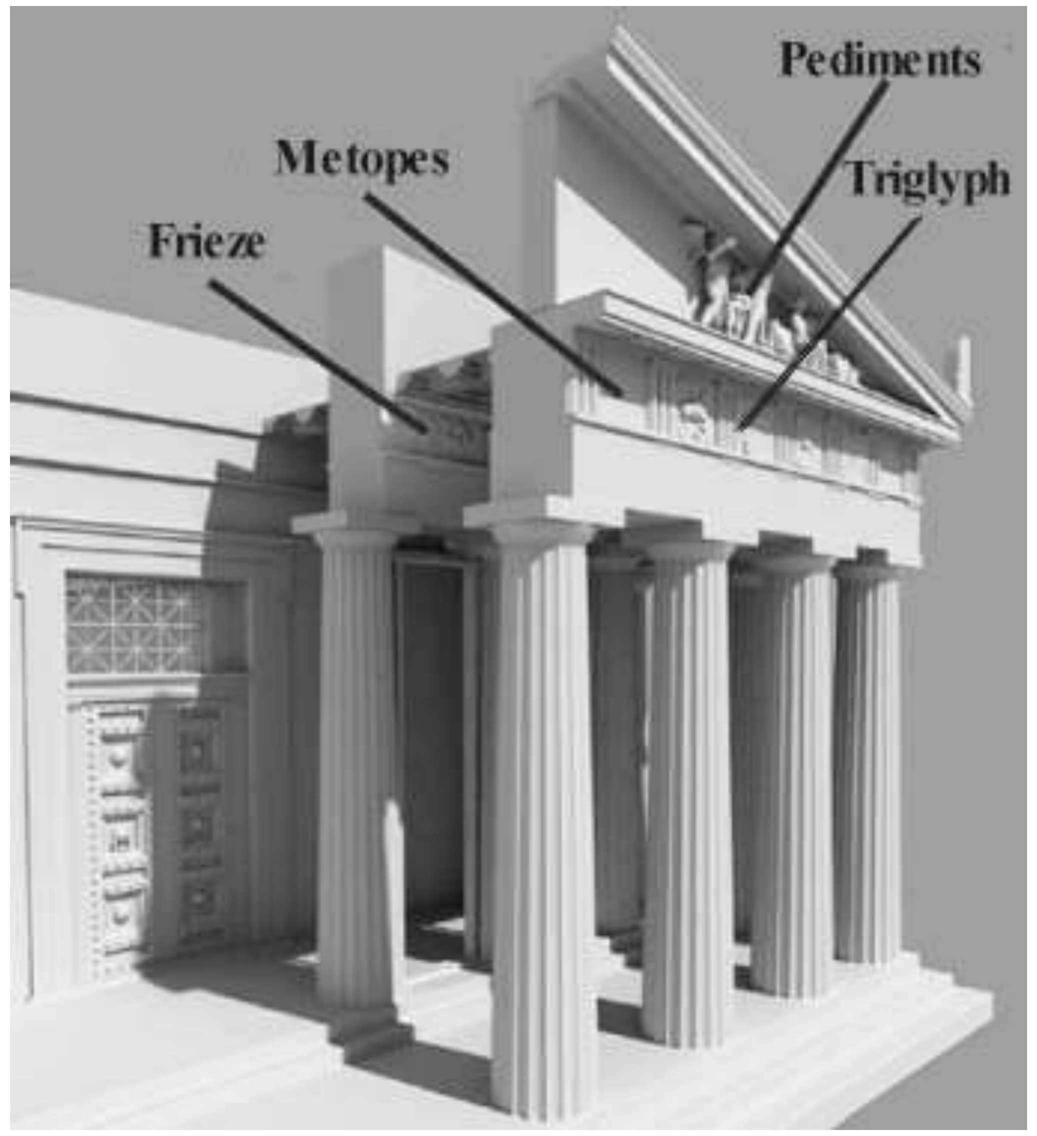
KIN+CARTA
Create

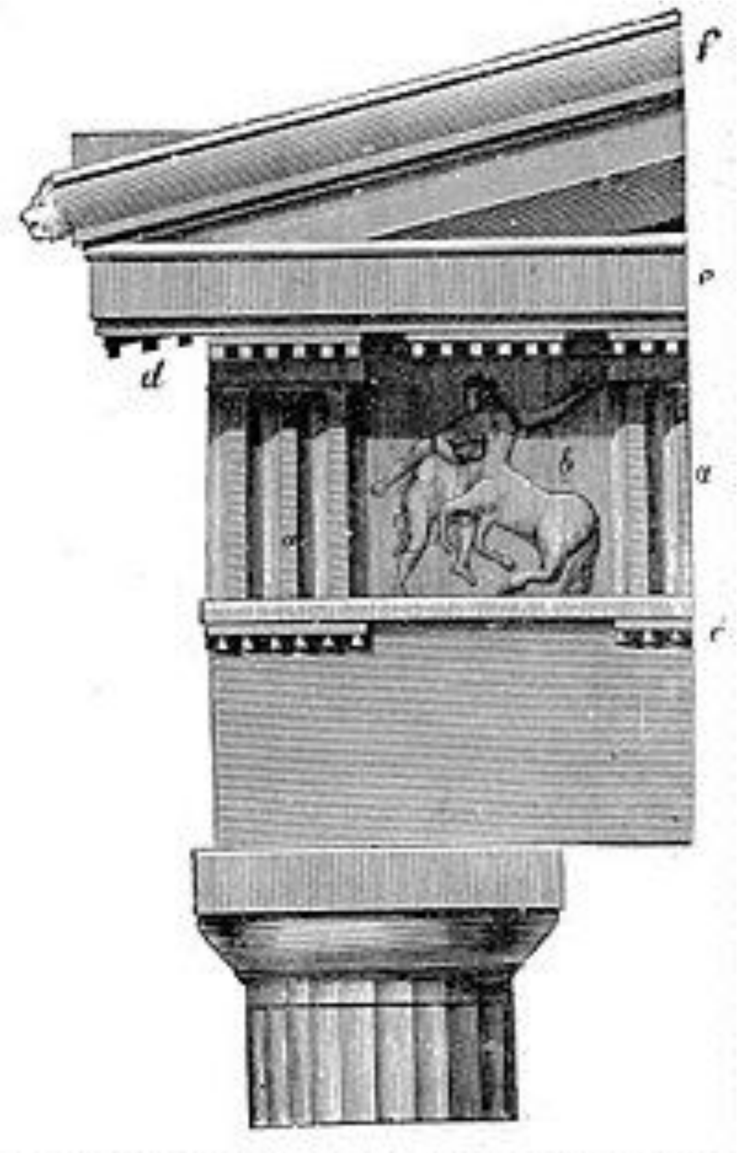
Living on a boat.

Navigator 2.0 and Riverpod hands on.
October 2021

Contents

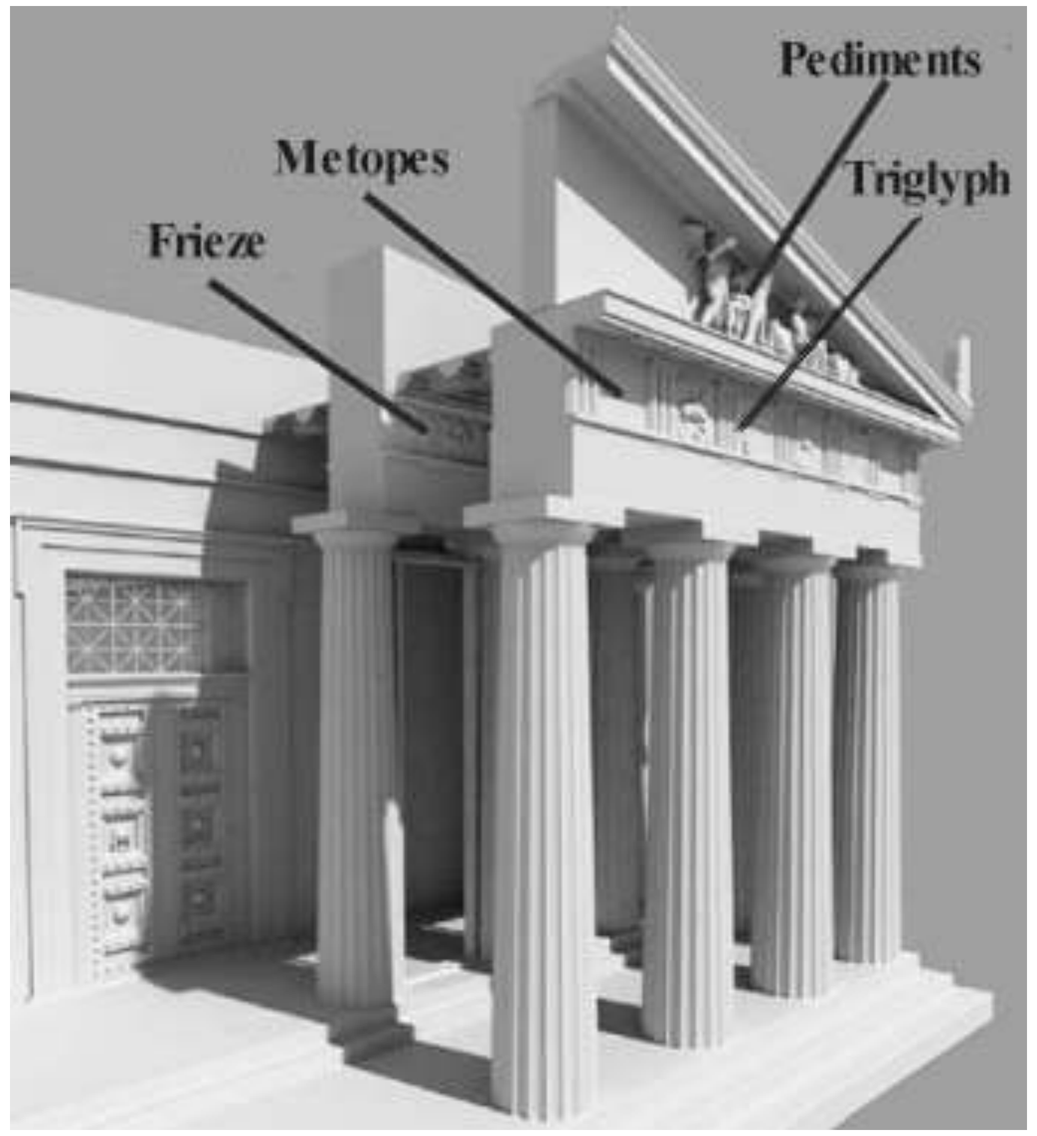
A word on Architecture

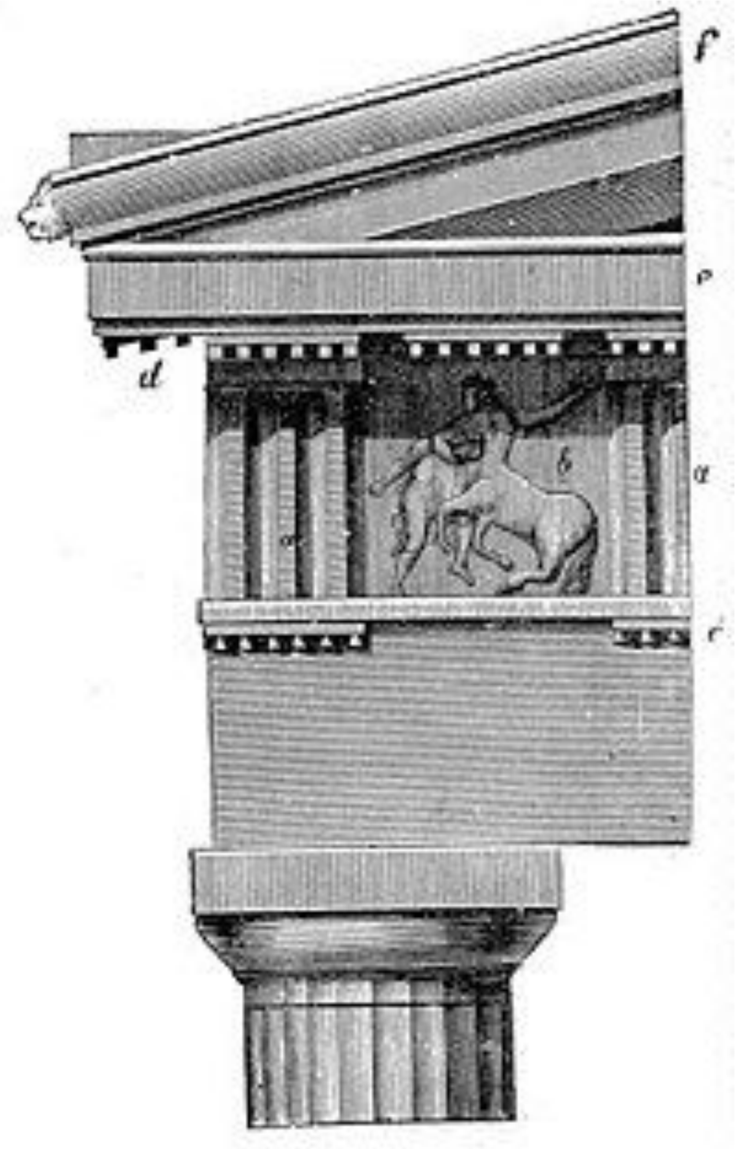




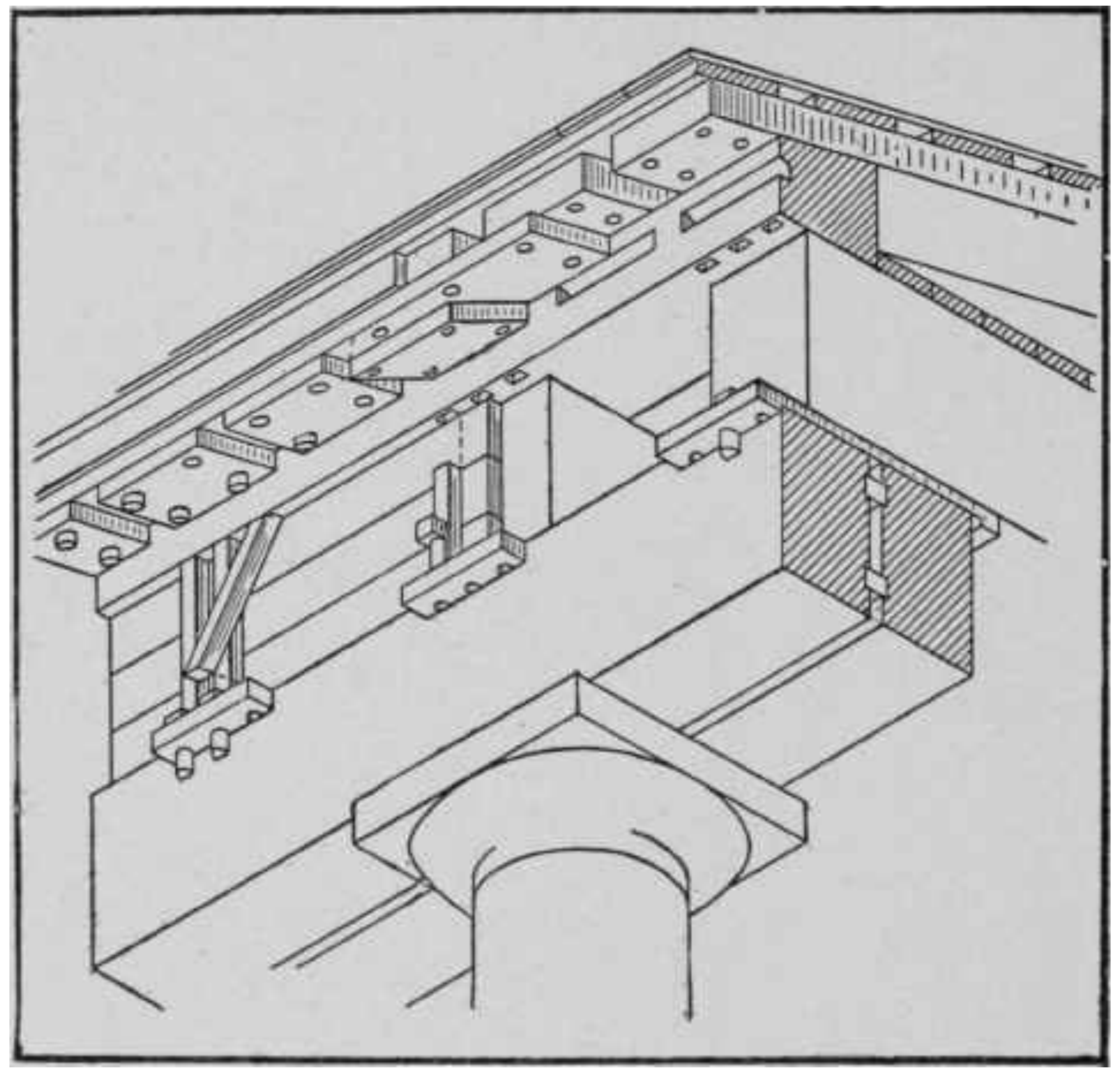
The Doric Order in the Parthenon at Athens.

A word on Architecture





The Doric Order in the Parthenon at Athens.



Doric order in timber

Riverpod, Freezed and Navigator 2.0

Riverpod

Second generation injection

Riverpod does NOT use InheritedWidget.

Riverpod does NOT force you to inject-by-class

Freezed

Distilled immutability

Freezed solves the problem of Dart lacking immutable data classes and union types (aka sealed classes)

Navigator 2.0

Open navigation

Navigator 2.0 fully controls the navigation stack while allowing the application being aware of system navigation

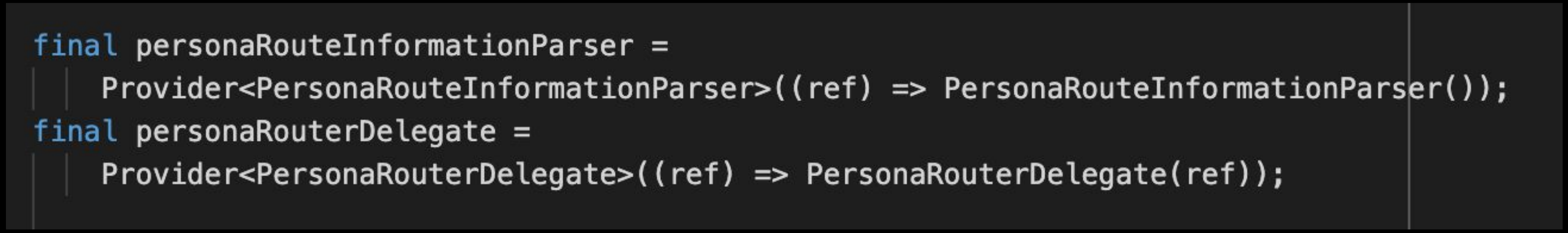
Riverpod

Provider

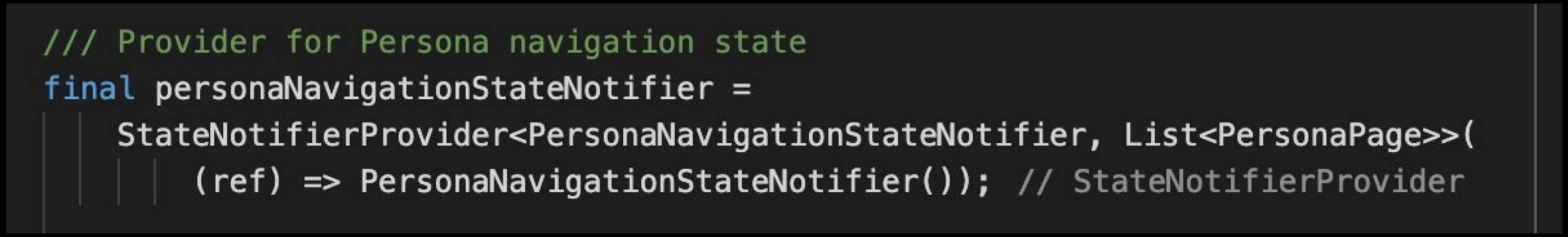
Returns instance of the object. Very useful for singletons.

StateNotifierProvider

Returns the state and the notifier of that state. Notifier is used for updating the state.



```
final personaRouteInformationParser =
    Provider<PersonaRouteInformationParser>((ref) => PersonaRouteInformationParser());
final personaRouterDelegate =
    Provider<PersonaRouterDelegate>((ref) => PersonaRouterDelegate(ref));
```



```
/// Provider for Persona navigation state
final personaNavigationStateNotifier =
    StateNotifierProvider<PersonaNavigationStateNotifier, List<PersonaPage>>(
        (ref) => PersonaNavigationStateNotifier()); // StateNotifierProvider
```

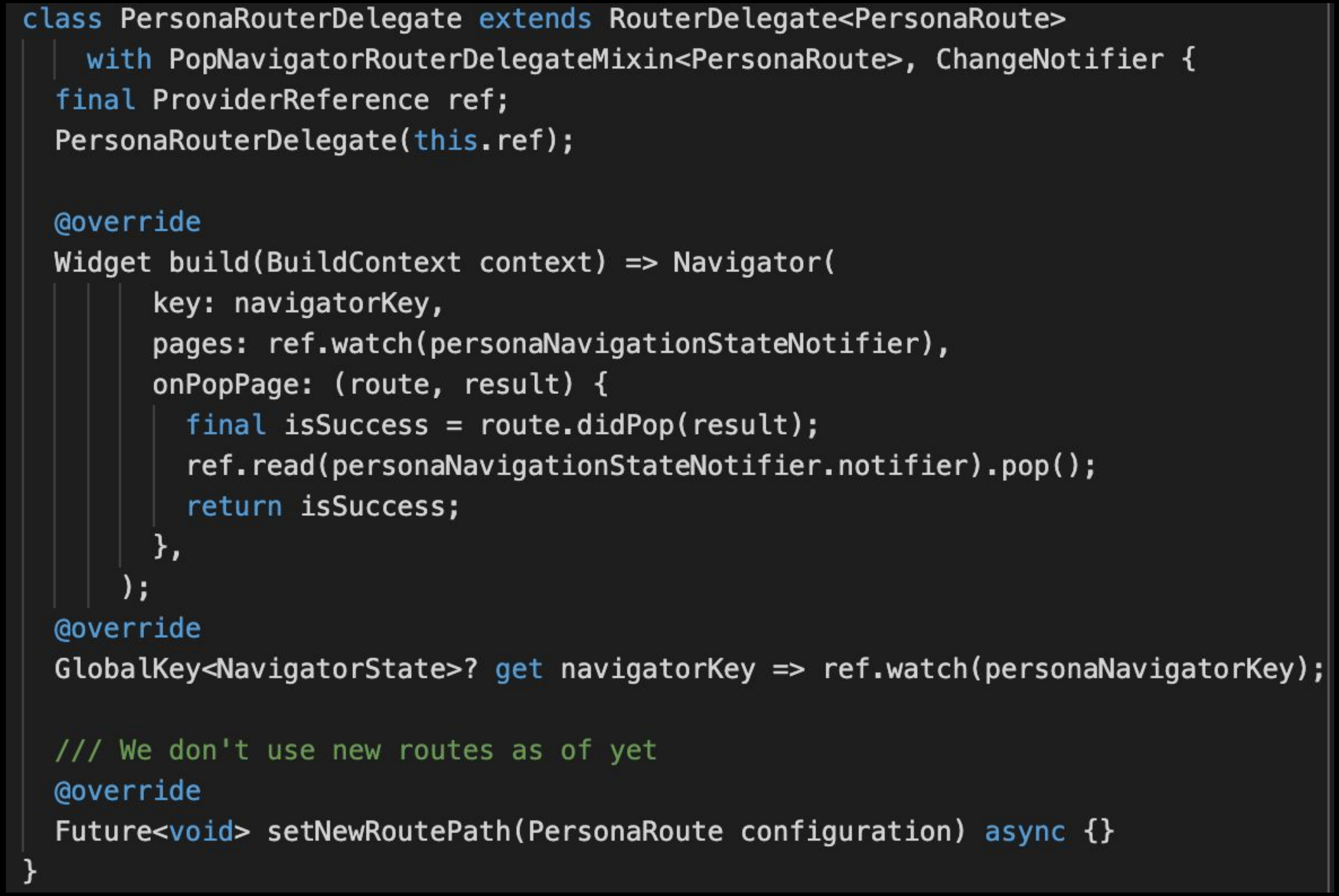
What do we need from Navigator 2.0

RouterDelegate

It’s purpose is to return Navigator widget with a stack of pages.

RouteInformationParser

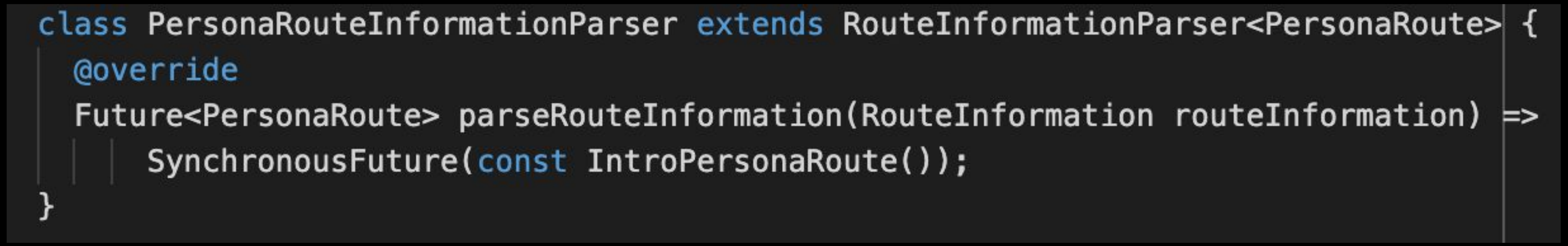
It’s purpose is to break down the RouteInformation into your route object. In our example it will be called only once when the application starts.



```
class PersonaRouterDelegate extends RouterDelegate<PersonaRoute>
    with PopNavigatorRouterDelegateMixin<PersonaRoute>, ChangeNotifier {
  final ProviderReference ref;
  PersonaRouterDelegate(this.ref);

  @override
  Widget build(BuildContext context) => Navigator(
        key: navigatorKey,
        pages: ref.watch(personaNavigationStateNotifier),
        onPopPage: (route, result) {
          final isSuccess = route.didPop(result);
          ref.read(personaNavigationStateNotifier.notifier).pop();
          return isSuccess;
        },
      );
  @override
  GlobalKey<NavigatorState>? get navigatorKey => ref.watch(personaNavigatorKey);

  /// We don't use new routes as of yet
  @override
  Future<void> setNewRoutePath(PersonaRoute configuration) async {}
}
```

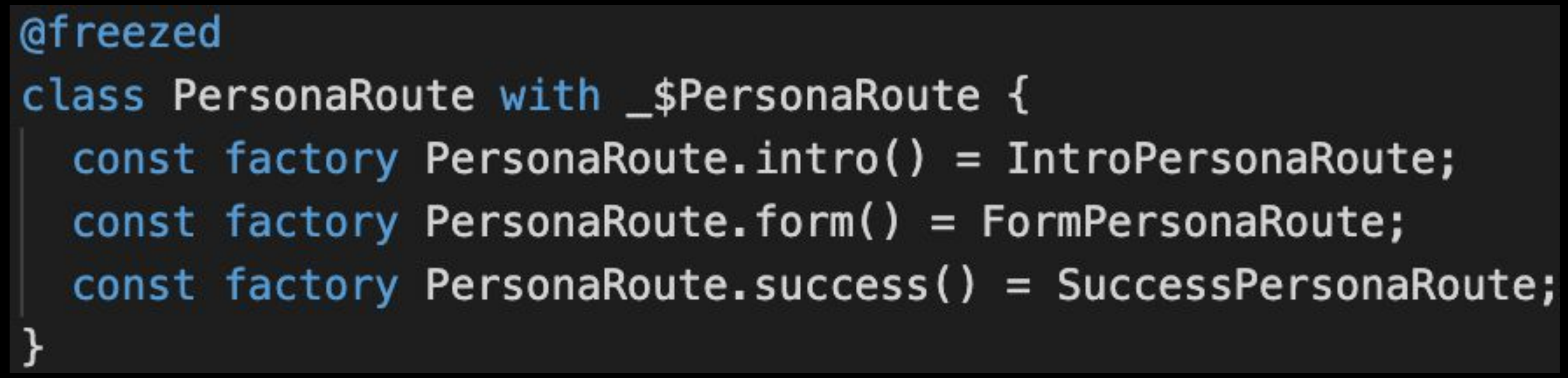


```
class PersonaRouteInformationParser extends RouteInformationParser<PersonaRoute> {
  @override
  Future<PersonaRoute> parseRouteInformation(RouteInformation routeInformation) =>
      SynchronousFuture(const IntroPersonaRoute());
}
```

Freezed

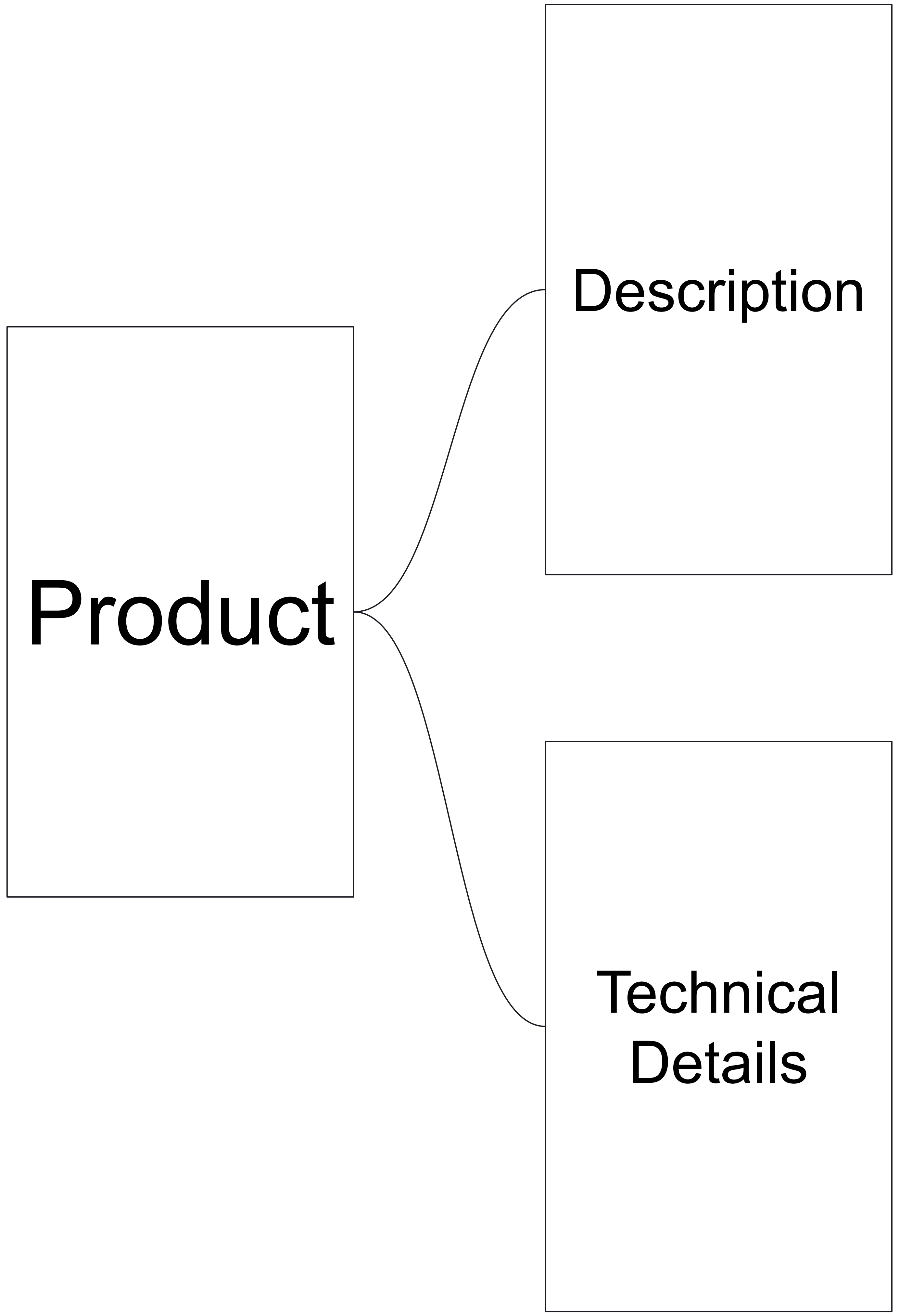
Union type allows to create closed group of subtypes which can have distinct parameters.

In our case we won’t need otherwise handy methods when() and map() which force implementation of handlers for all subtypes.



```
@freezed
class PersonaRoute with _$PersonaRoute {
  const factory PersonaRoute.intro() = IntroPersonaRoute;
  const factory PersonaRoute.form() = FormPersonaRoute;
  const factory PersonaRoute.success() = SuccessPersonaRoute;
}
```

Coding



Product

Description

Technical Details

Thanks

KIN+CARTA
Create

Swav Kulinski
Flutter GDE, Flutter Lead at Kin+Carta

swav.kulinski@gmail.com
https://www.linkedin.com/in/swavkulinski/