



Flutter  
Talks 2019

# State Management with Provider

---

Improve your app using Provider

# Sobre o Palestrante



Mobile Software Architect  
<https://aeyrium.com>

- **Diego Velásquez López**
- **Google Developer Expert Flutter**
- **Software engineer**
- **Android / iOS / Flutter**
- **Creator of 2 Android apps (+100k downloads):**
  - **Pseudocode**
  - **Quick Printer**
- **Founder & Organizer @ Flutter Peru**

@diegoveloper

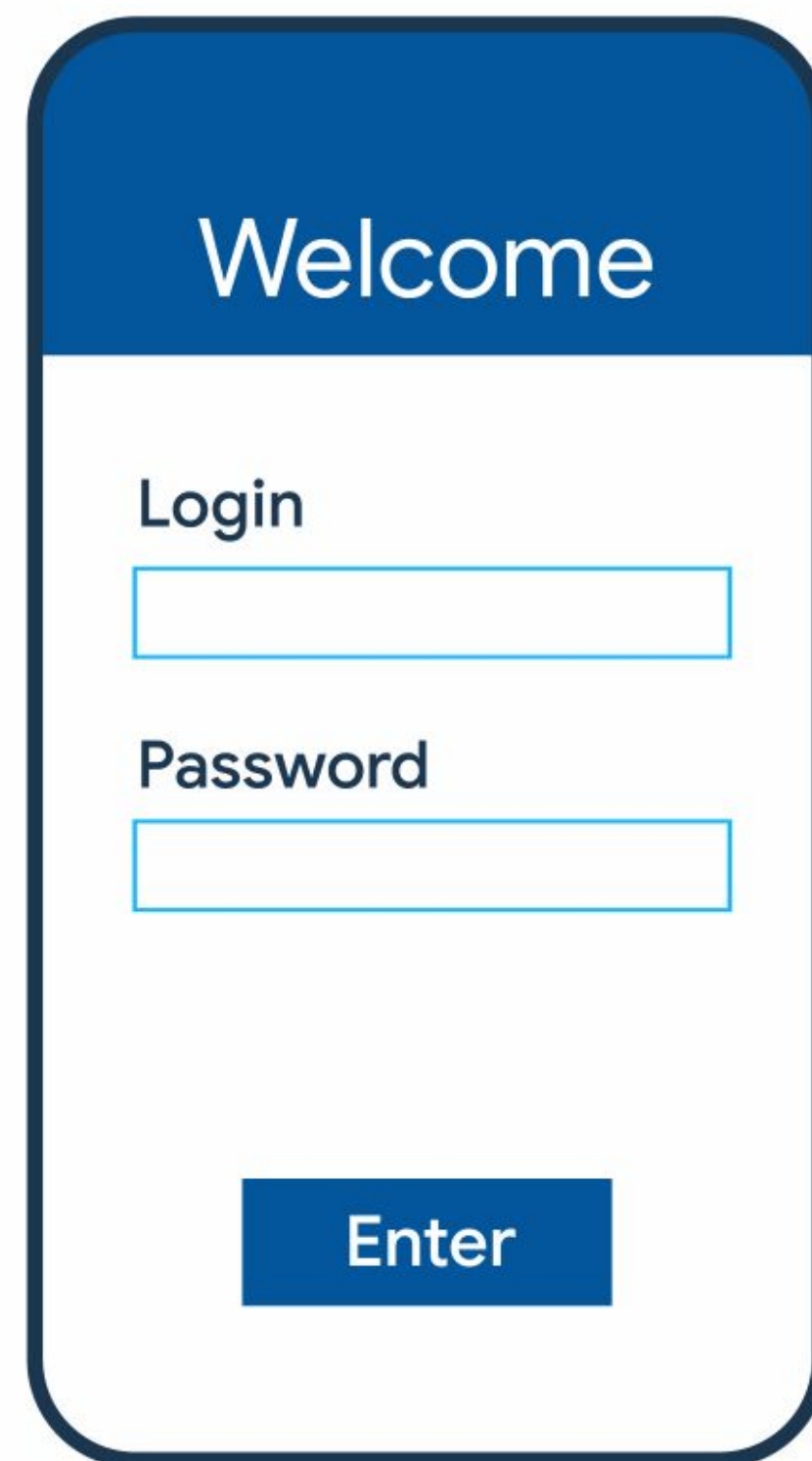




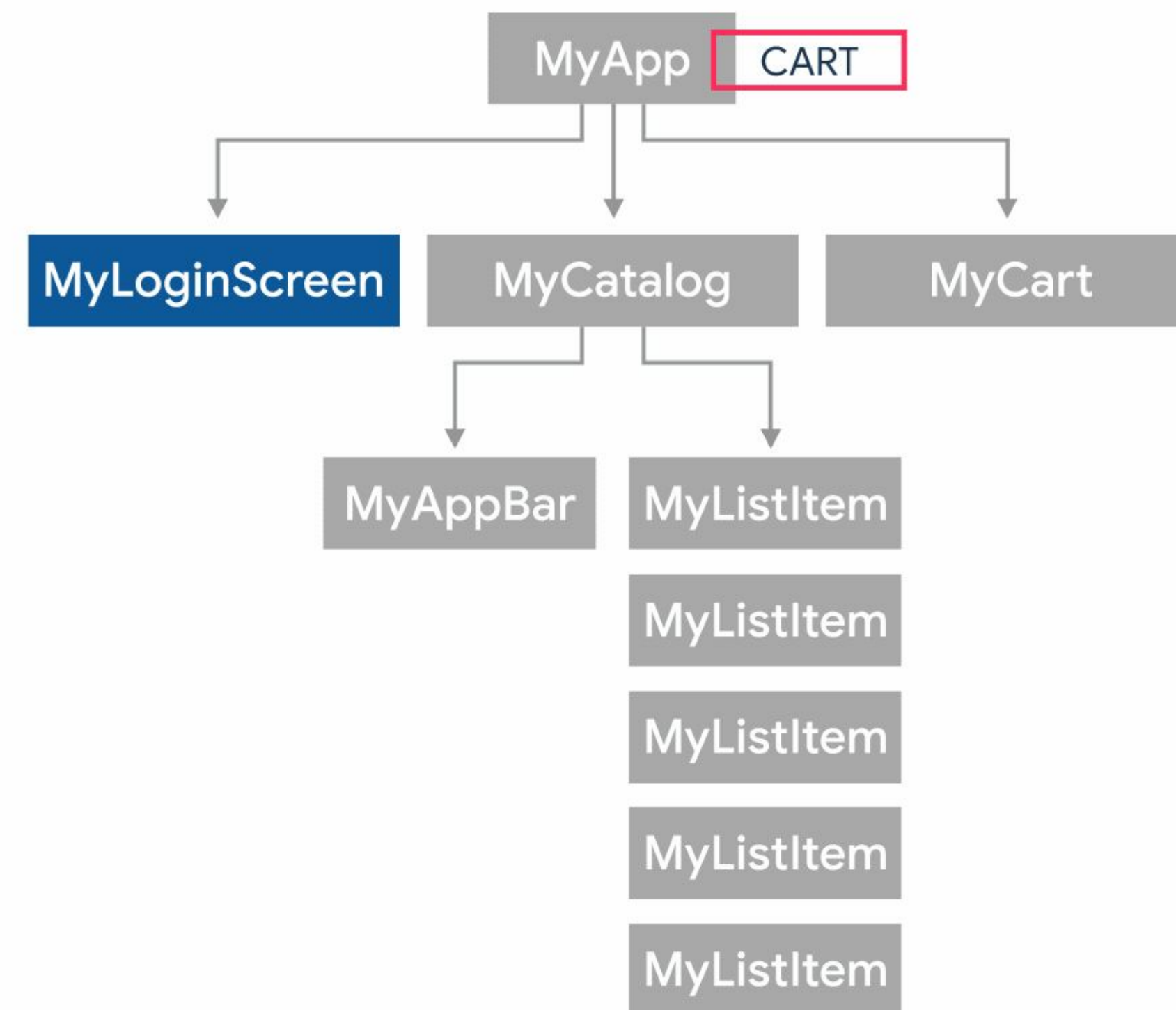
# Cronograma

- What, Why, How
- Live Code
- Provider
- Summary

# What is State Management?



A mobile app UI mockup with a dark blue header containing the word "Welcome". Below the header, there is a "Login" section with two text input fields labeled "Login" and "Password". At the bottom of the form is a dark blue button labeled "Enter".



# Why?

- **Keep UI separated from business logic.**
- **Increase the performance of your app.**
- **Allows adding features quickly without breaking the code neither sacrificing code quality.**
- **Easy to understand; hard to break.**
- **Easy to test.**



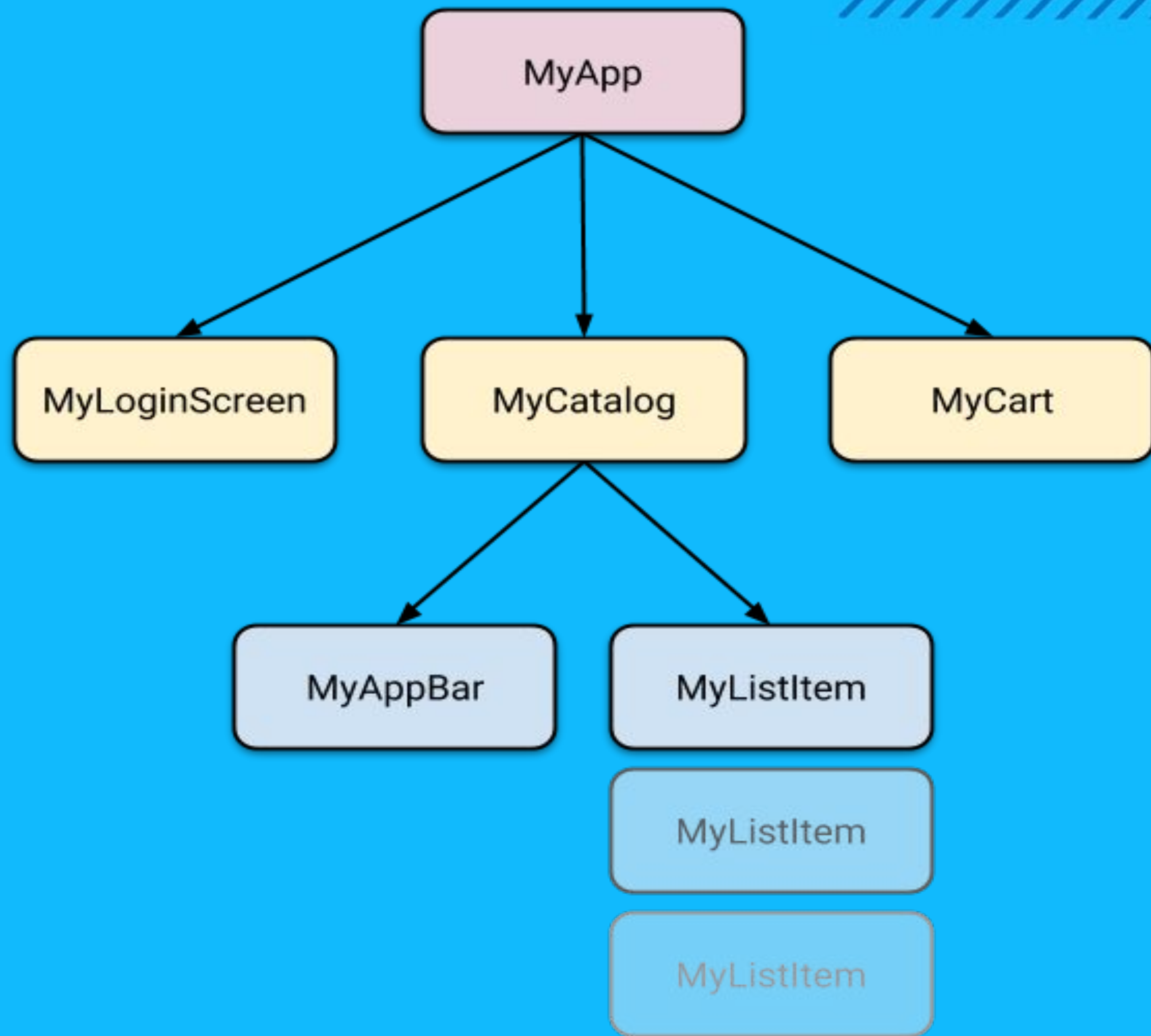
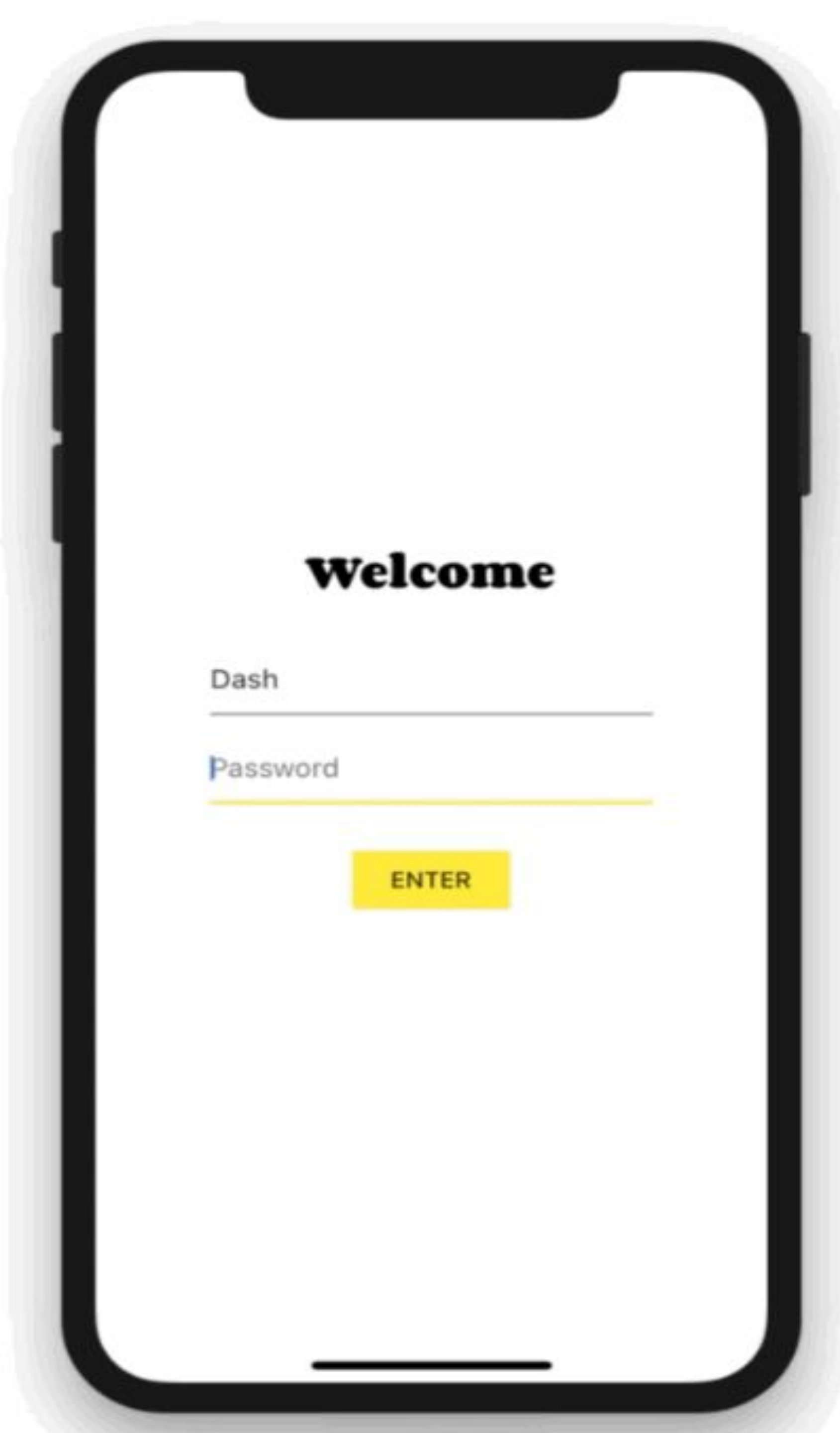
# How

- **Using setState() and StatefulWidget**
- **ScopedModel**
- **Provider**
- **BLoC (Business Logic Component)**
- **Redux**
- **MobX**

# Let's code









# Provider

<https://pub.dev/packages/provider>

**Expose/Provide**

**Read/Listen**

# Expose/Provide

name	description
Provider	The most basic form of provider. It takes a value and exposes it, whatever the value is.
ListenableProvider	A specific provider for Listenable object. ListenableProvider will listen to the object and ask widgets which depend on it to rebuild whenever the listener is called.
ChangeNotifierProvider	A specification of ListenableProvider for ChangeNotifier. It will automatically call <code>ChangeNotifier.dispose</code> when needed.
ValueListenableProvider	Listen to a ValueListenable and only expose <code>ValueListenable.value</code> .
StreamProvider	Listen to a Stream and expose the latest value emitted.
FutureProvider	Takes a <code>Future</code> and updates dependents when the future completes.

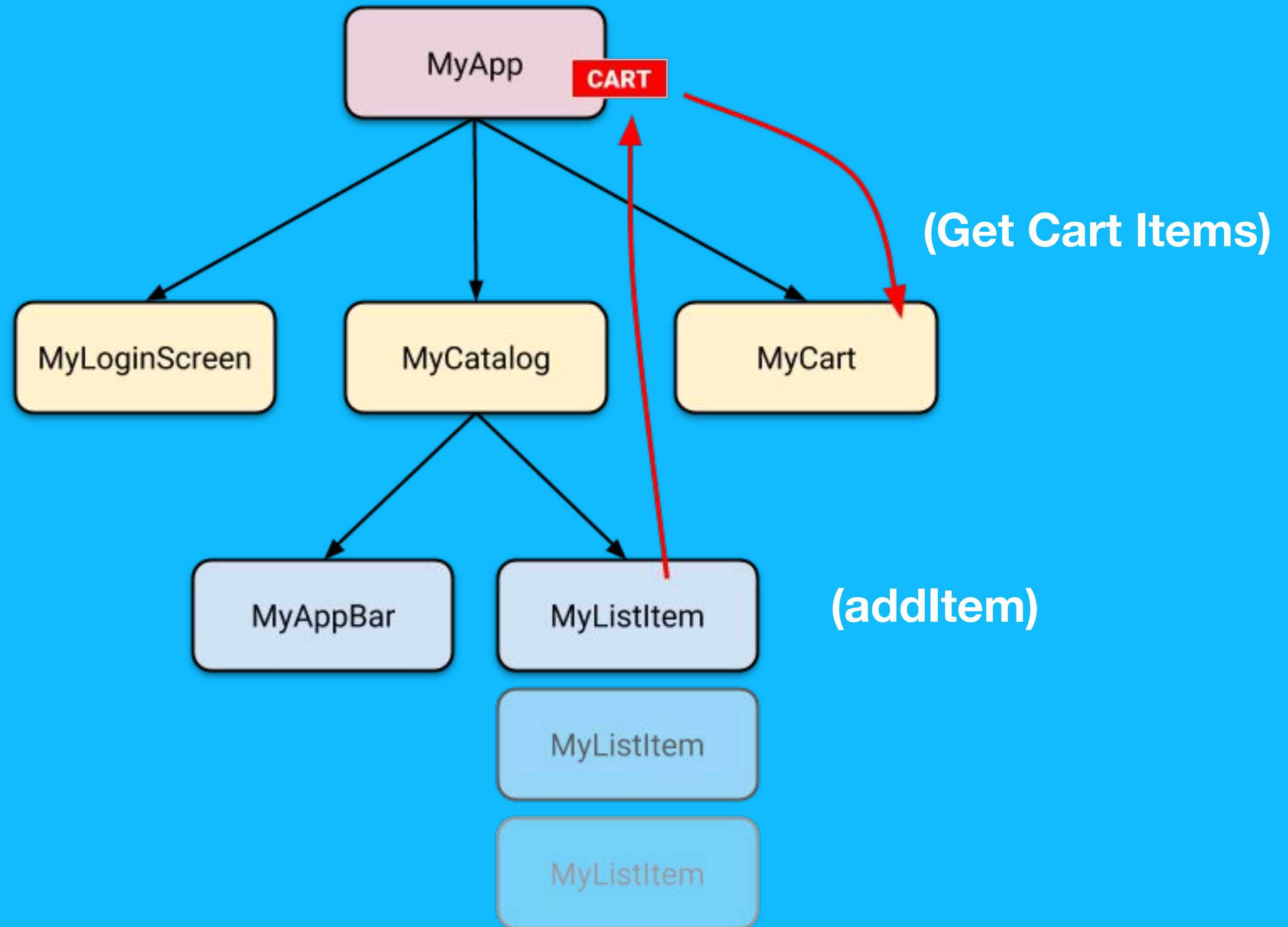


# Read/Listen

```
Provider.of<String>(context)
```

```
Foo(  
  child: Consumer<A>(  
    builder: (_, a, child) {  
      return Bar(a: a, child: child);  
    },  
    child: Baz(),  
  ),  
)
```

## ChangeNotifierProvider (CartModel)





# Summary

- **There's not right or wrong solution.**
- **Choose the one that best suits your needs.**

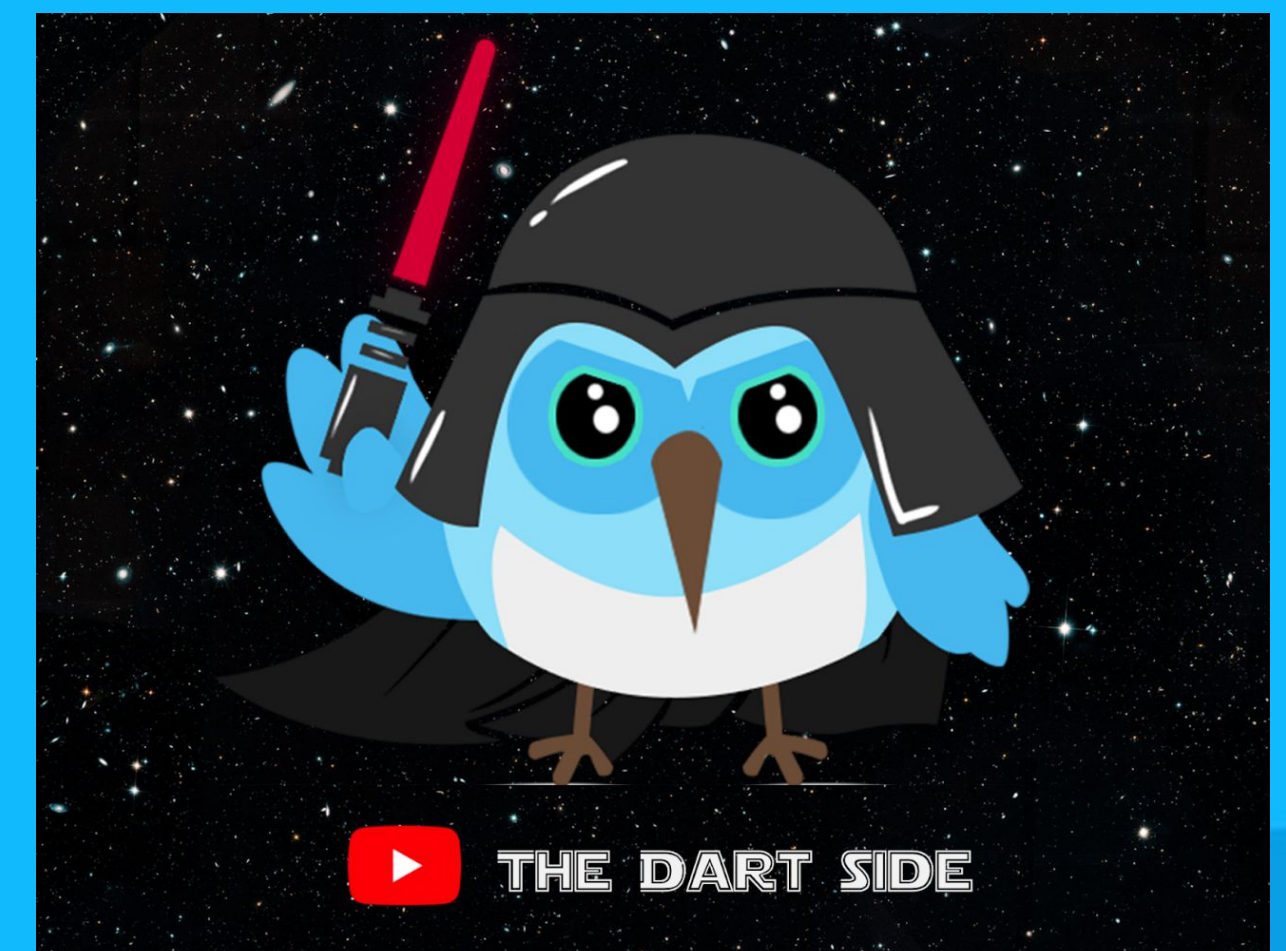




# Perguntas e Respostas

---

- [diegoverloper@gmail.com](mailto:diegoverloper@gmail.com)
- <https://github.com/diegoverloper/>
- <https://medium.com/@diegoverloper>
- <https://twitter.com/diegoverloper>







Flutter  
Talks 2019

Obrigado!