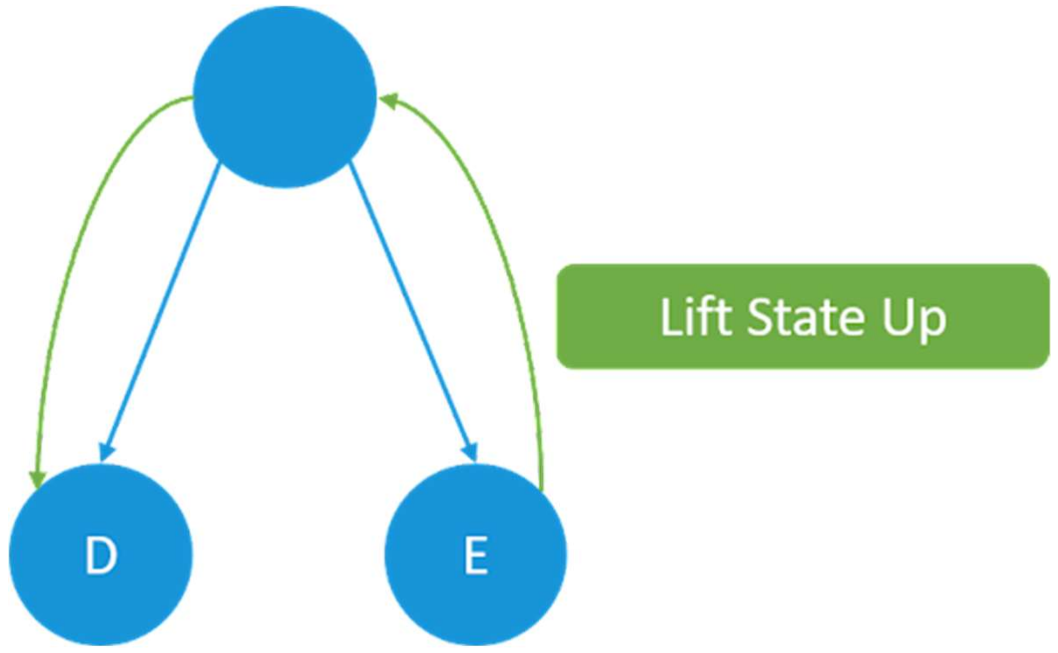
Flutter Fundamentals
Short recap day #2

Peter Kassenaar –
info@kassenaar.com

Yesterday:

- State management in various ways
  - Stateful widgets – using models/custom classes
  - passing parameters, passing functions
  - Design principle: “Lifting state up”

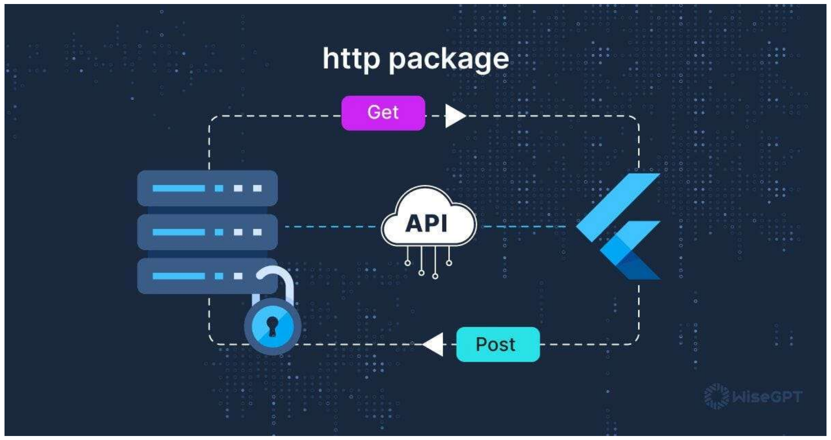


two types of widgets

- `Stateless` Widgets, `Stateful` Widgets
  - Stateful Widgets are actually two classes:
- 1. Retain state between repaints, so widget's internal state survives hot reloads and rebuilds.
- 2. Optimize runtime performance, immutable widgets are fast to diff and rebuild;

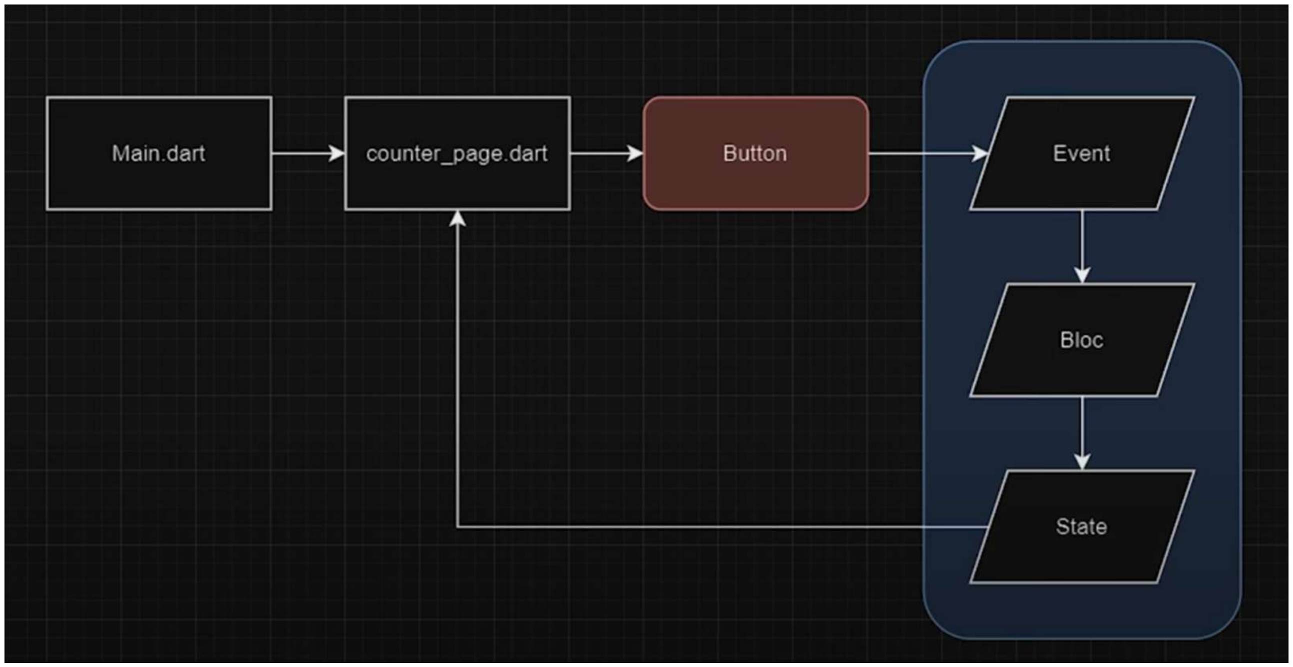
Yesterday:

- Creating and using Custom Classes, Extracting Widgets
  - Passing parameters
  - Passing functions
- Communicating with external API's
  - http / other methods

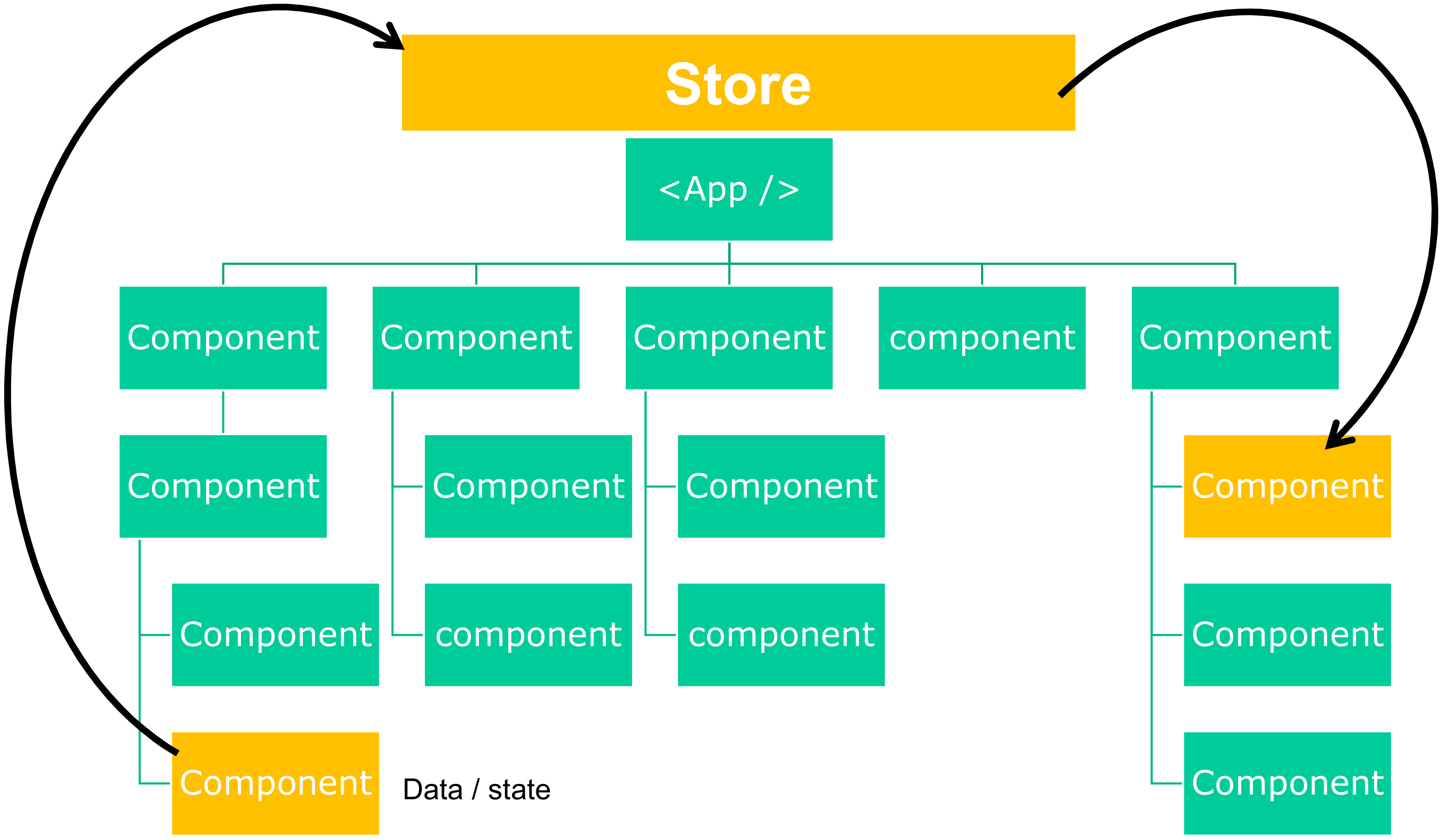


Yesterday

- State management: share application state between unconnected Widgets/components
- Bloc pattern
  - State
  - Events
  - Bloc
  - Page



State management *with* a store



Bloc pattern

- Using `context.read<T>();`
  - Listen to state changens AND update the state
- Using `context.watch<T>().state`
  - listens to `<T>` state changes.
  - This way you can use (but then: not update) the state in other widgets (think: logged in/out notification, shopping cart)
- Also:
  - multiple state properties
  - Updating the state with a payload

Questions?

Today

- Using `TextFields`
- Routing / Navigation
- Complete applications using bloc/navigation/state
- gRPC
- Gestures
- …
- Evals & goodbye