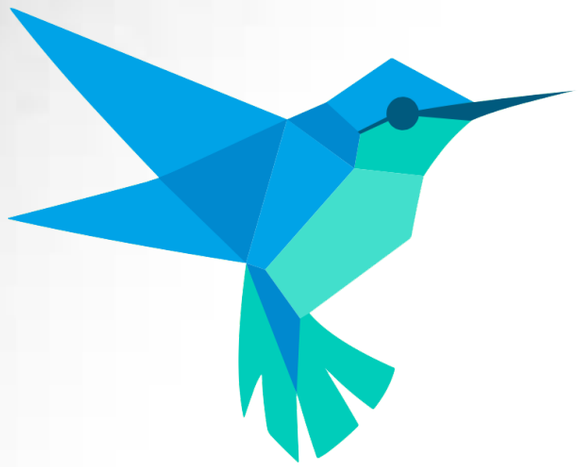


Inside Flutter

CIKLUM

EMPOWERING COLLABORATION

Hello 🙌

New format ⌨️

- Main rule: camera on, mic off
- Question? Raise a hand or ask in the chat
- Quick comment: press “space” to unmute (settings -> audio)
- Two breaks (5 min)

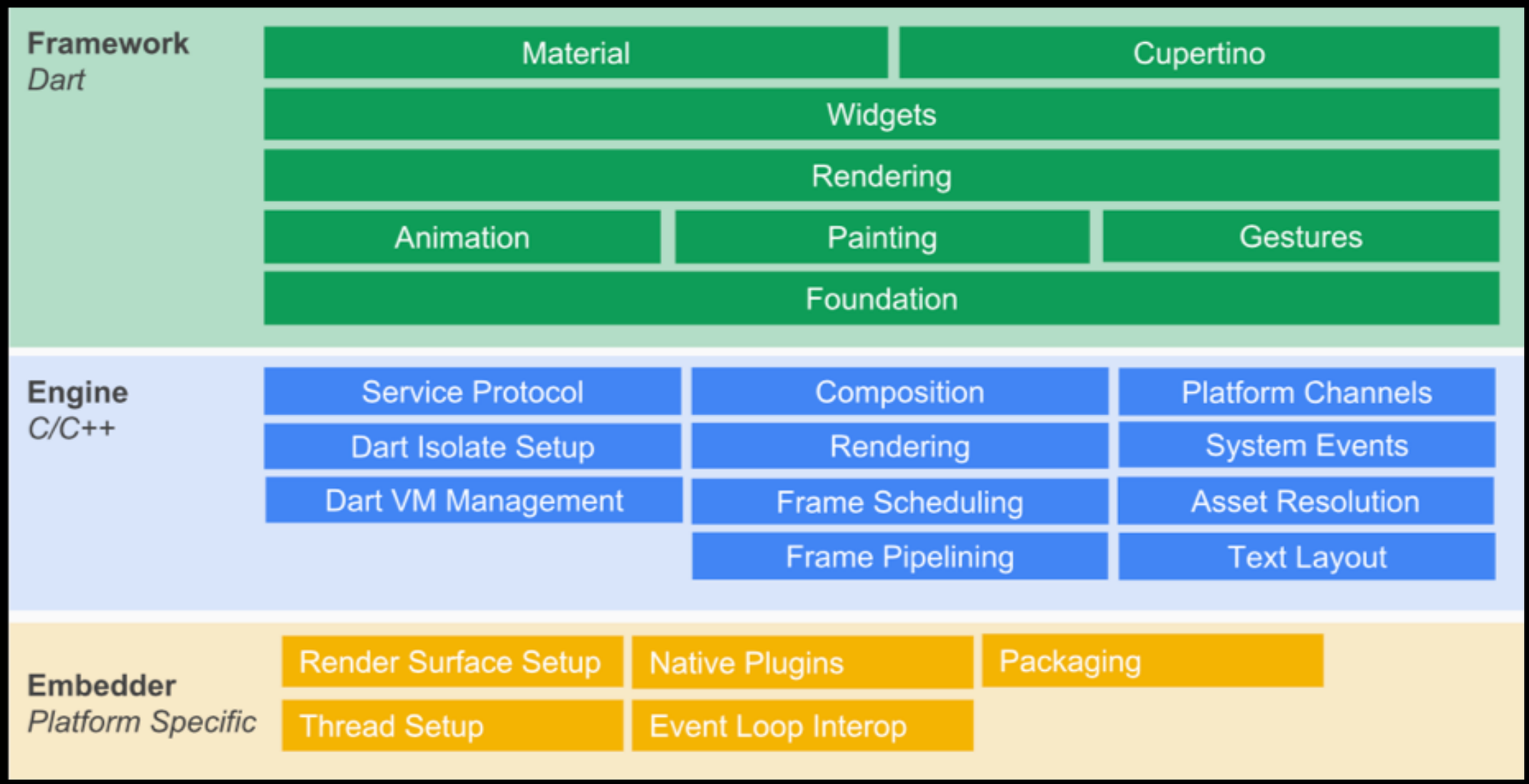
Homework

- In groups (3 students)
- Cross-group review
- All results are open
- Let me show you!

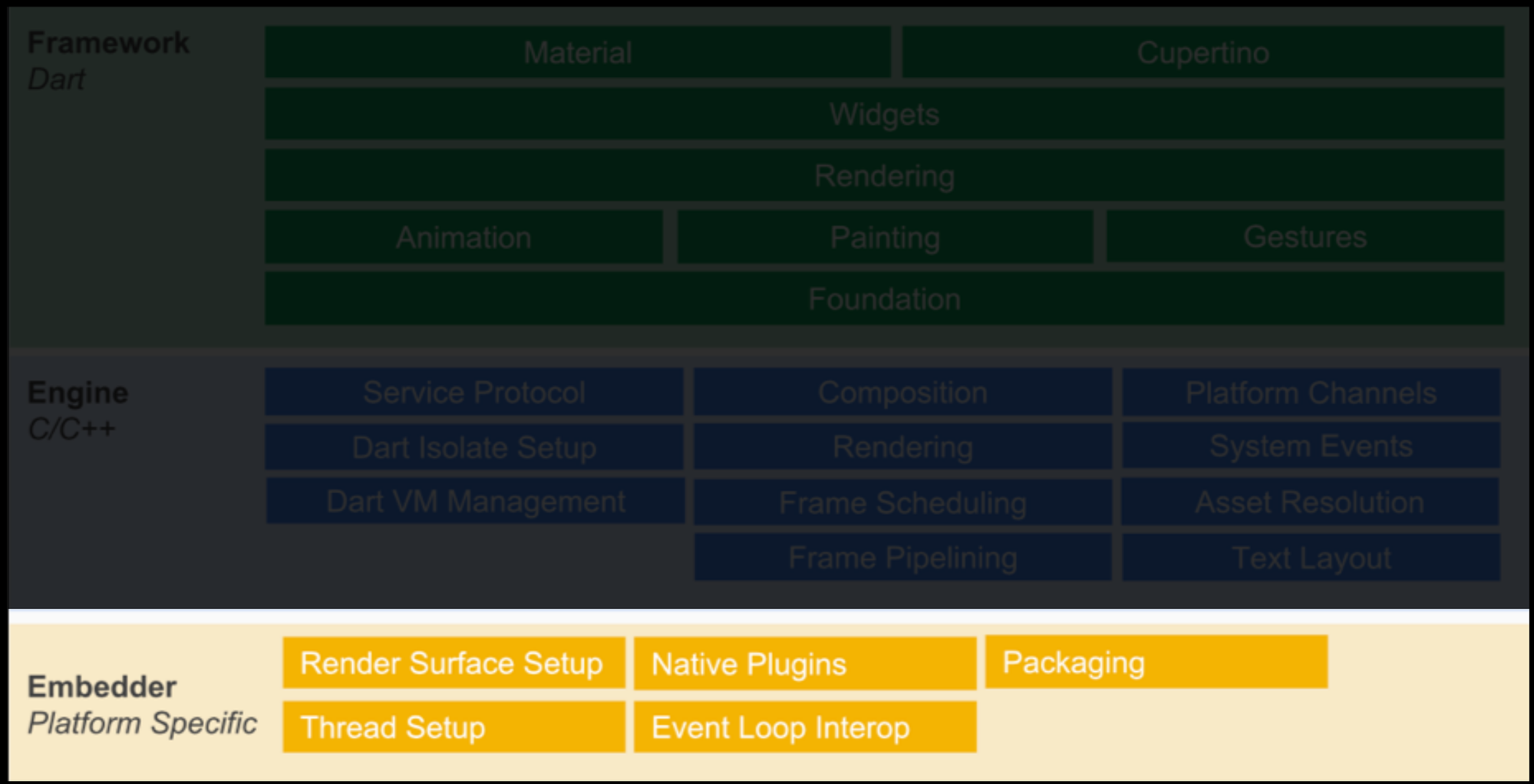
Inside Flutter

- Flutter Architecture
- Rendering
- Keys

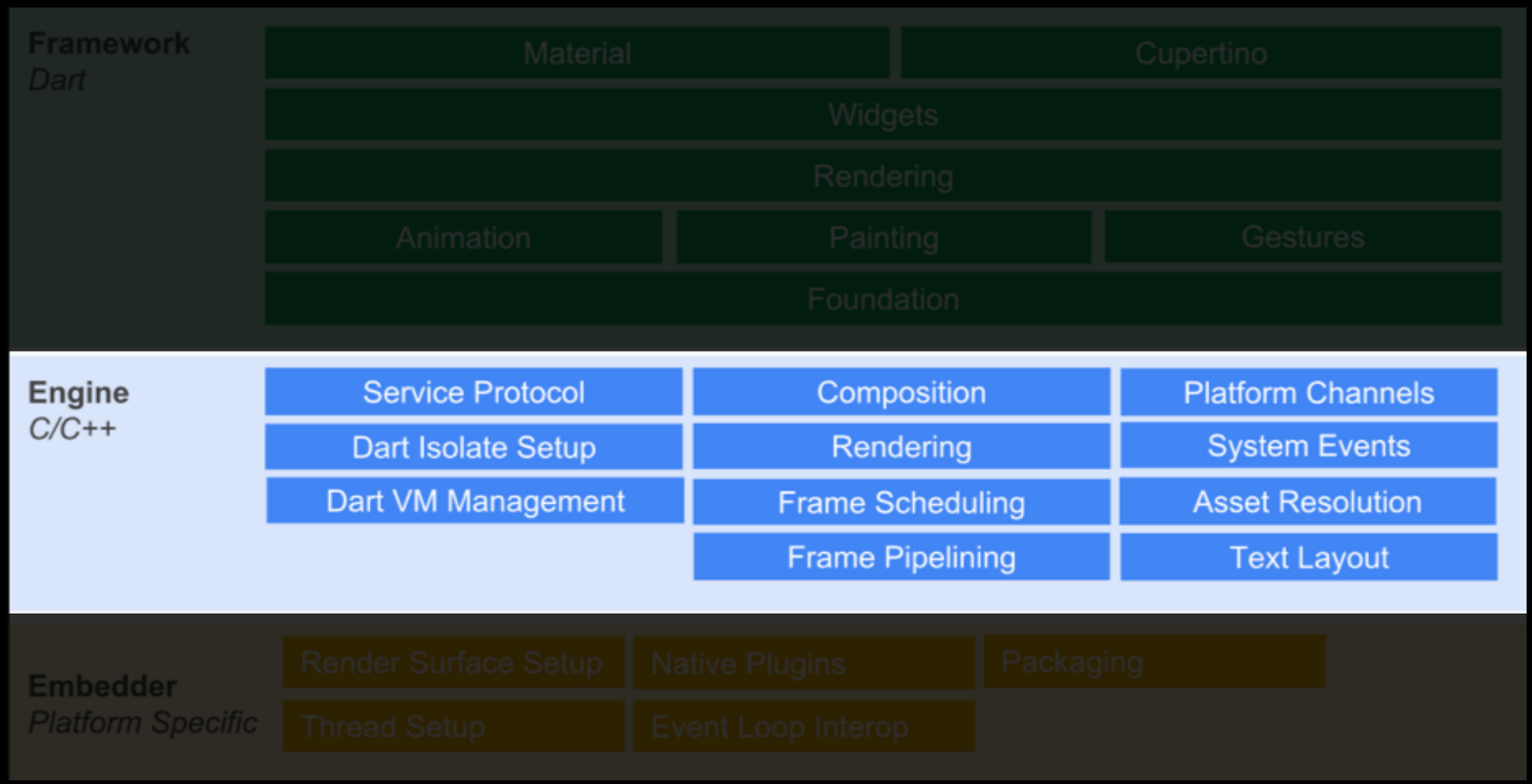
General Architecture



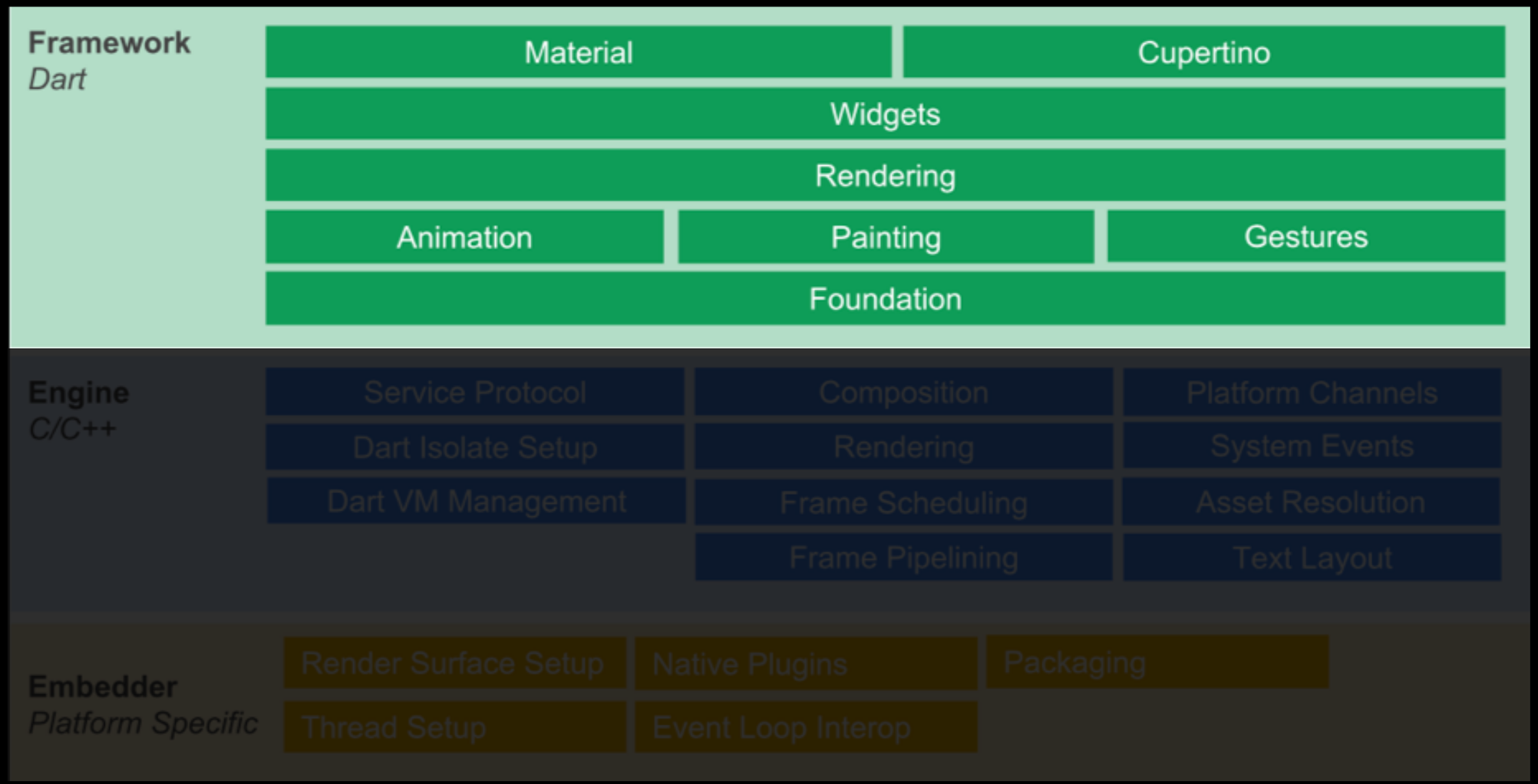
Embedder (Shell)



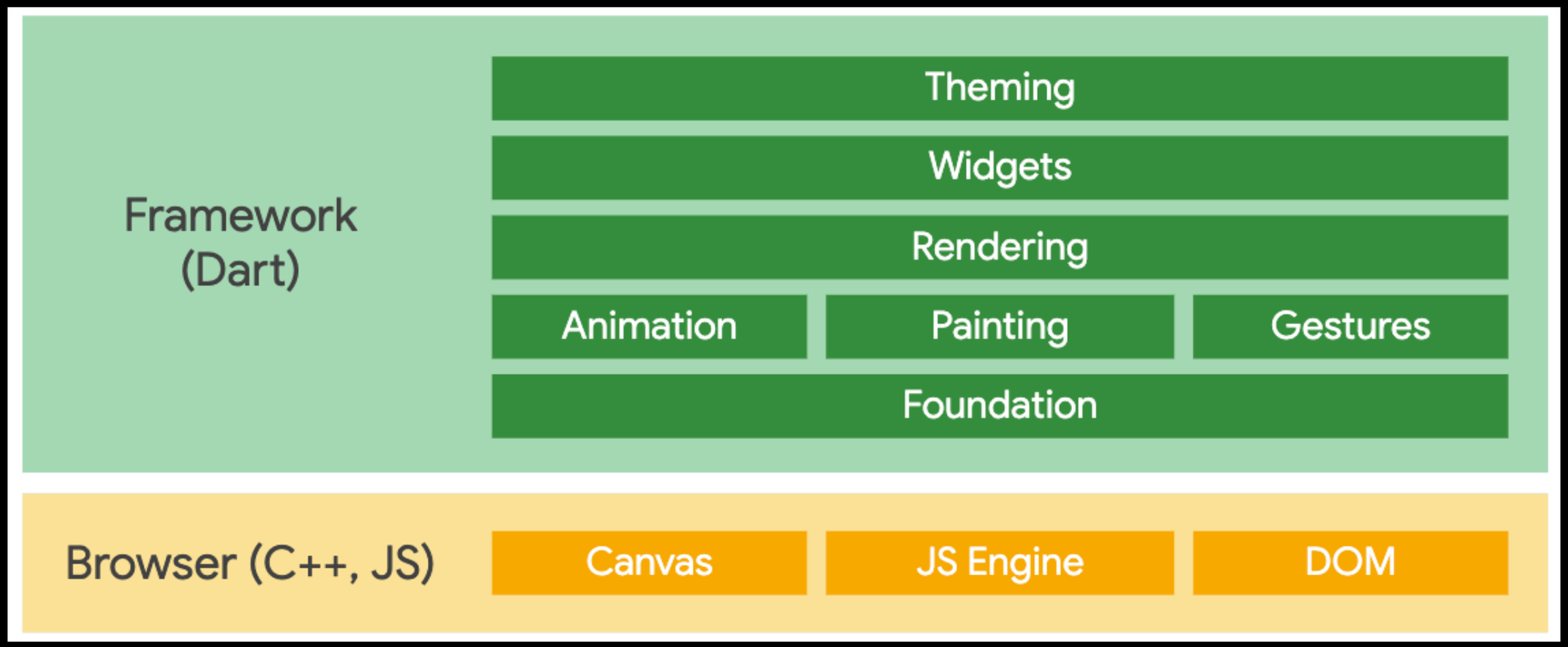
Engine


Framework
Dart
Material
Cupertino
Widgets
Rendering
Animation
Painting
Gestures
Foundation
Engine
C/C++
Service Protocol
Composition
Platform Channels
Dart Isolate Setup
Rendering
System Events
Dart VM Management
Frame Scheduling
Asset Resolution
Frame Pipelining
Text Layout
Embedder
Platform Specific
Render Surface Setup
Native Plugins
Packaging
Thread Setup
Event Loop Interop

Framework

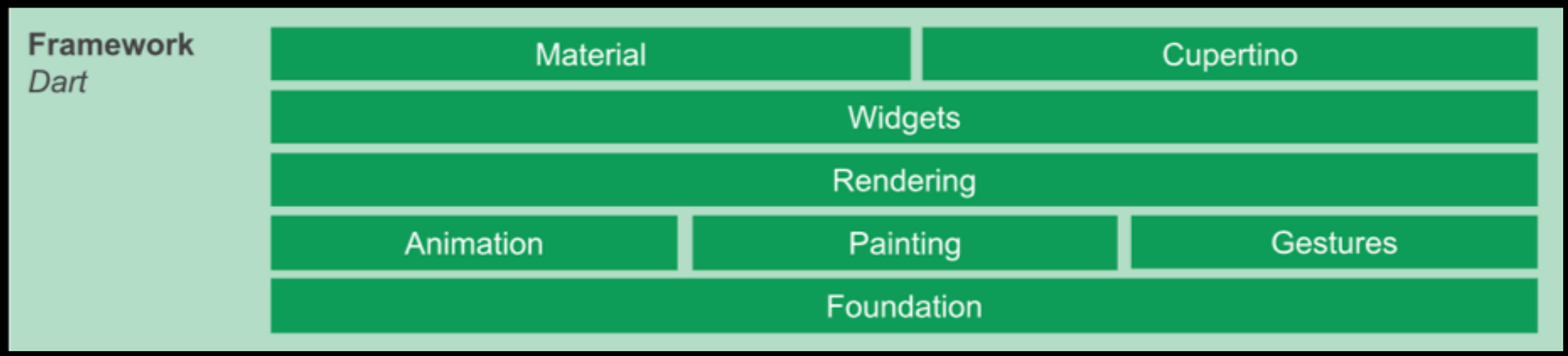

Framework
Dart
Material
Cupertino
Widgets
Rendering
Animation
Painting
Gestures
Foundation
Engine
C/C++
Service Protocol
Composition
Platform Channels
Dart Isolate Setup
Rendering
System Events
Dart VM Management
Frame Scheduling
Asset Resolution
Frame Pipelining
Text Layout
Embedder
Platform Specific
Render Surface Setup
Native Plugins
Packaging
Thread Setup
Event Loop Interop

Flutter for web



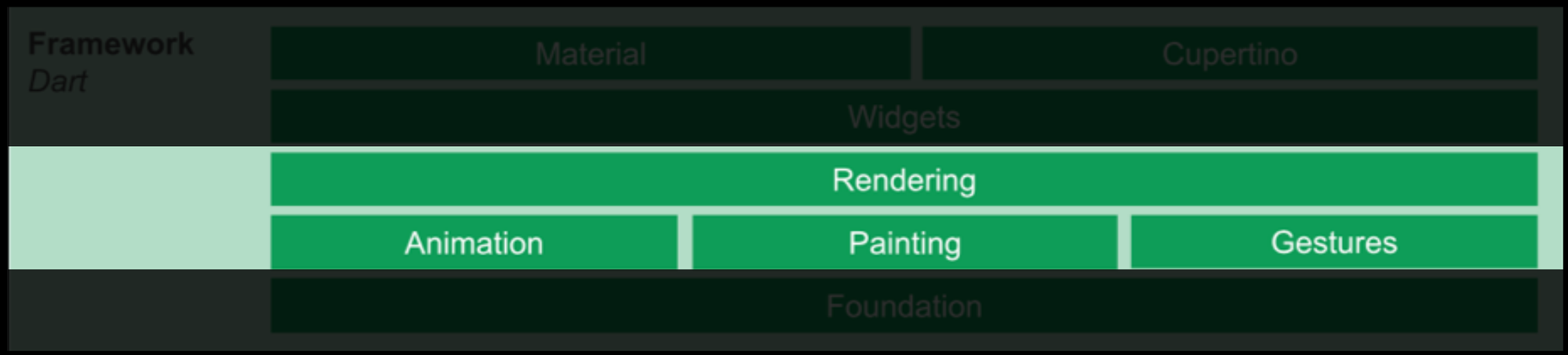
96% of your time you use the Framework layer only

Framework in details

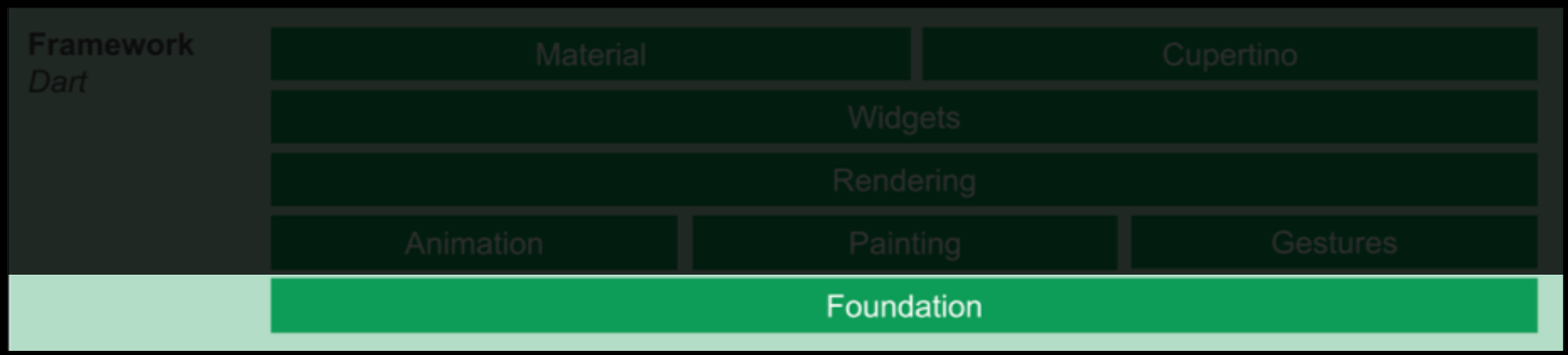


Framework in details

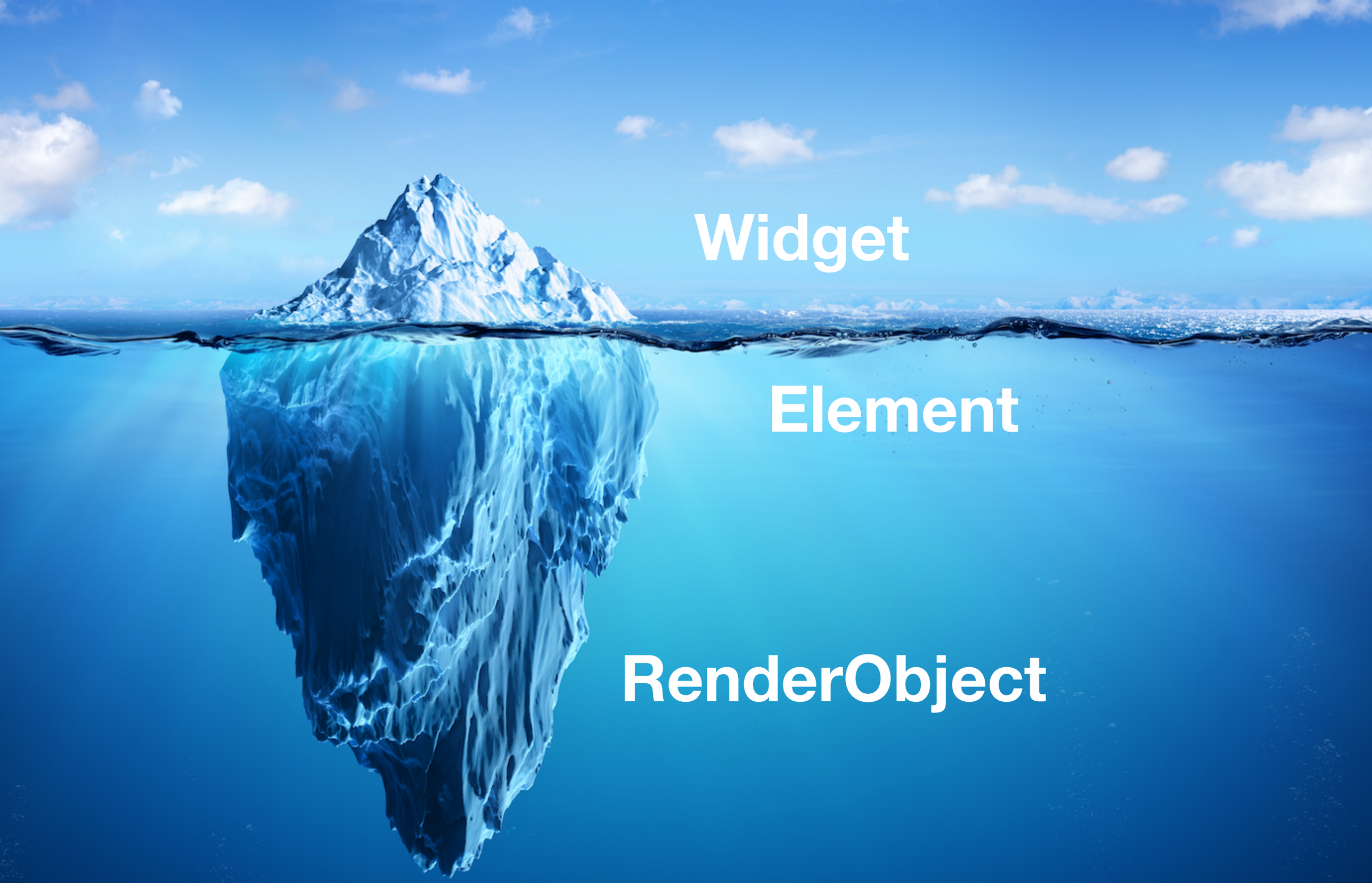
Framework in details



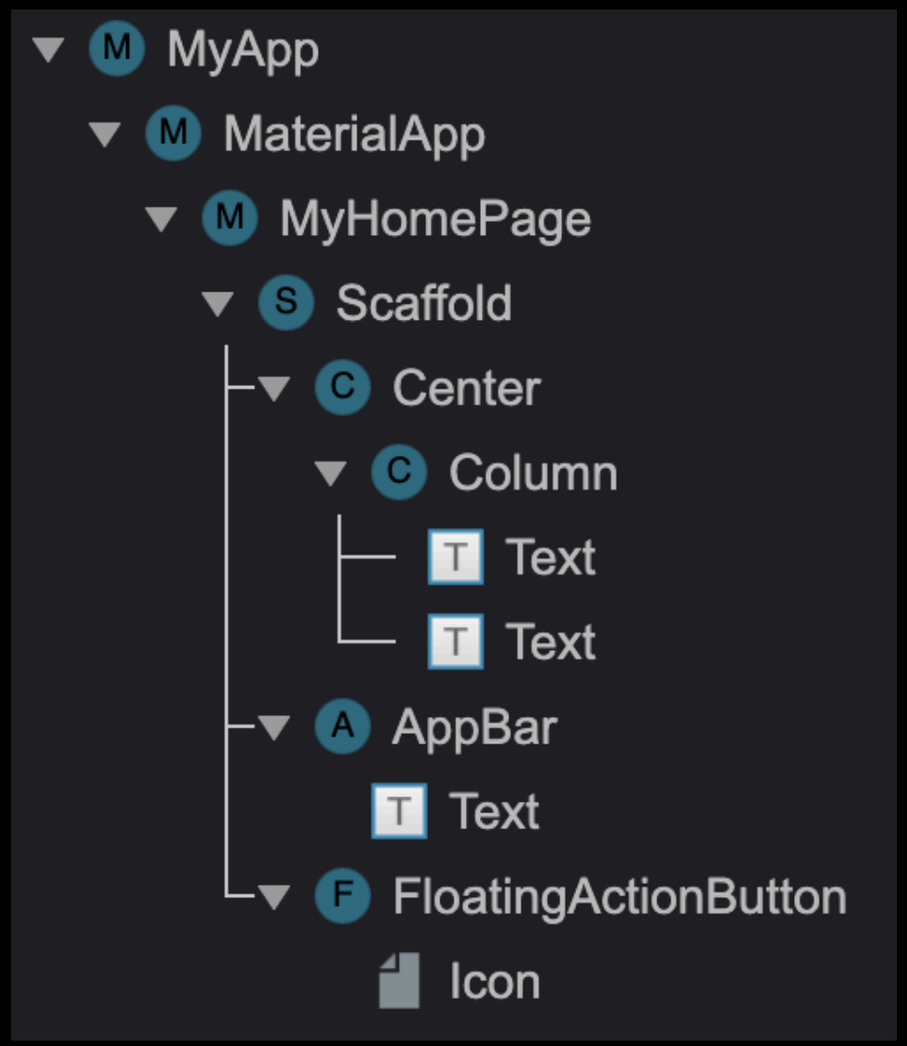
Framework in details



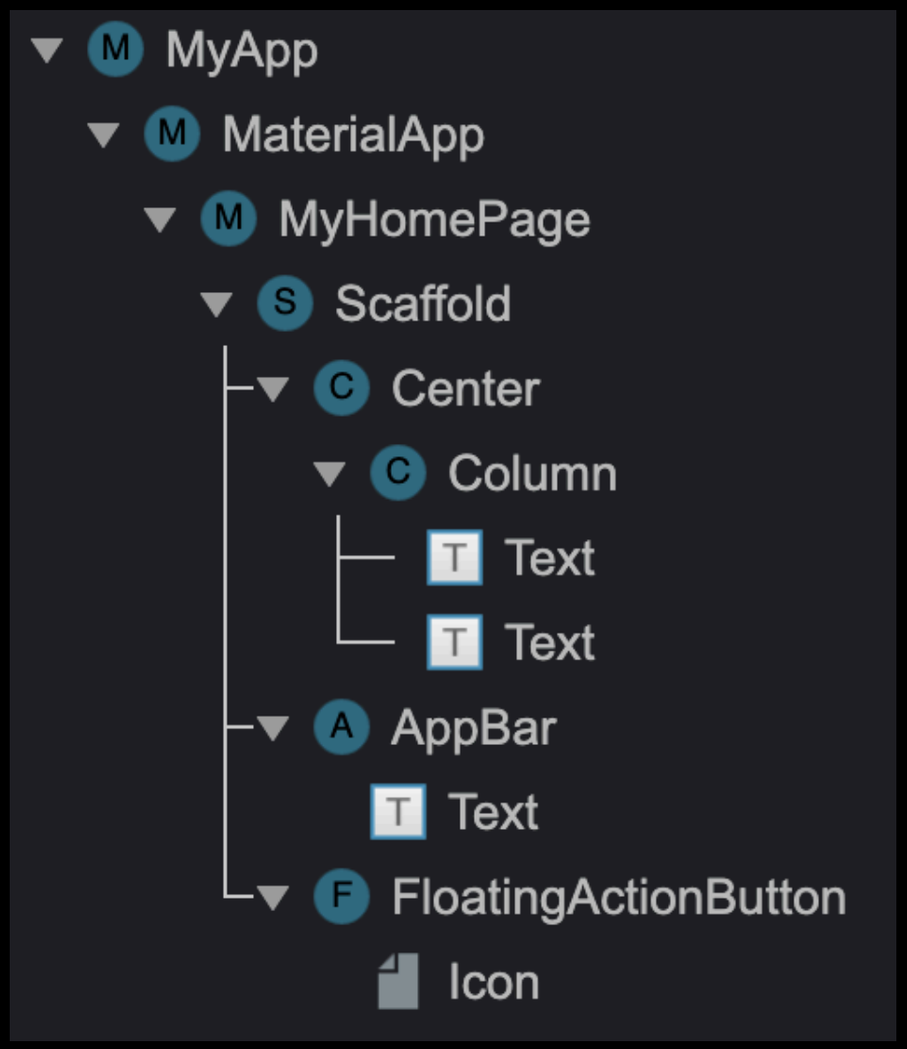
Flutter Entities

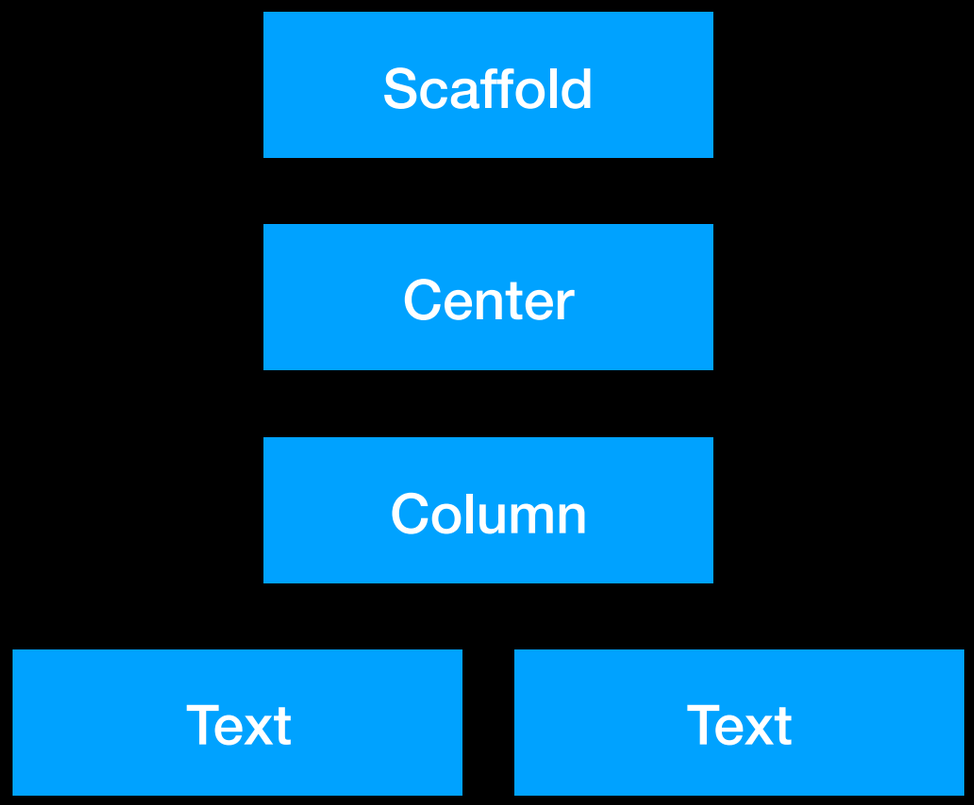

Widget
Element
RenderObject

Hierarchy

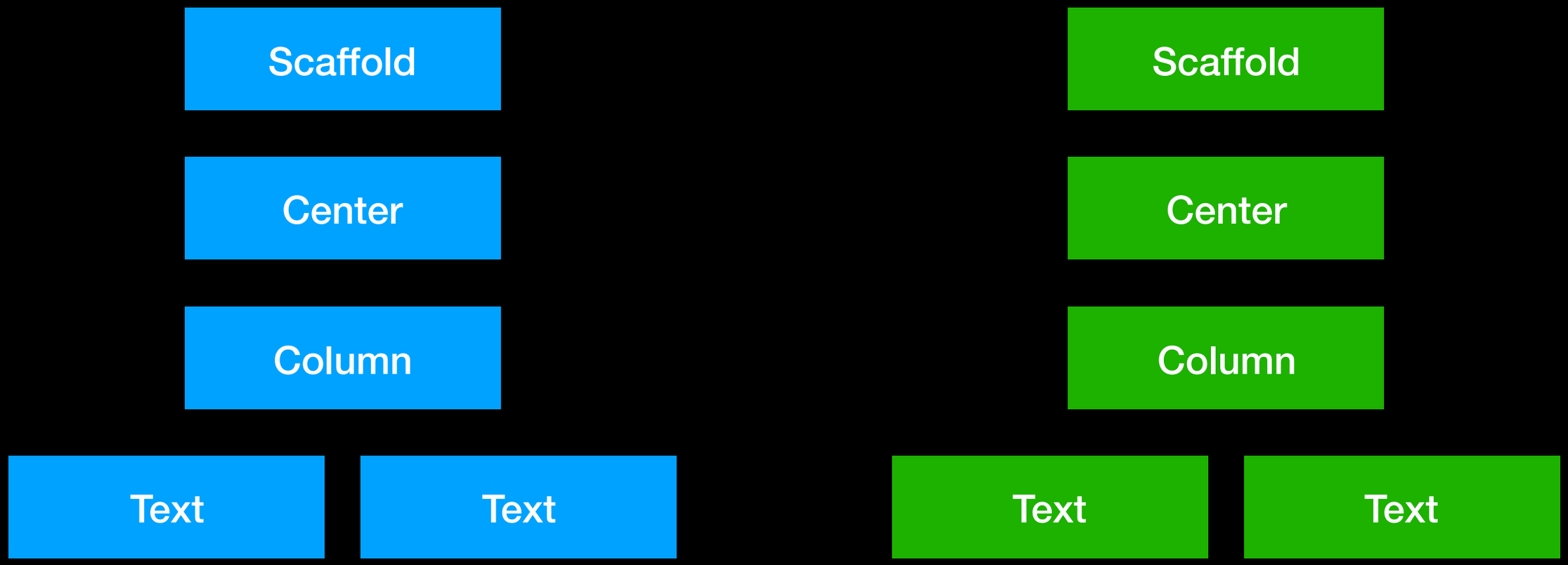

MyApp
MaterialApp
MyHomePage
Scaffold
Center
Column
Text
Text
AppBar
Text
FloatingActionButton
Icon

Hierarchy

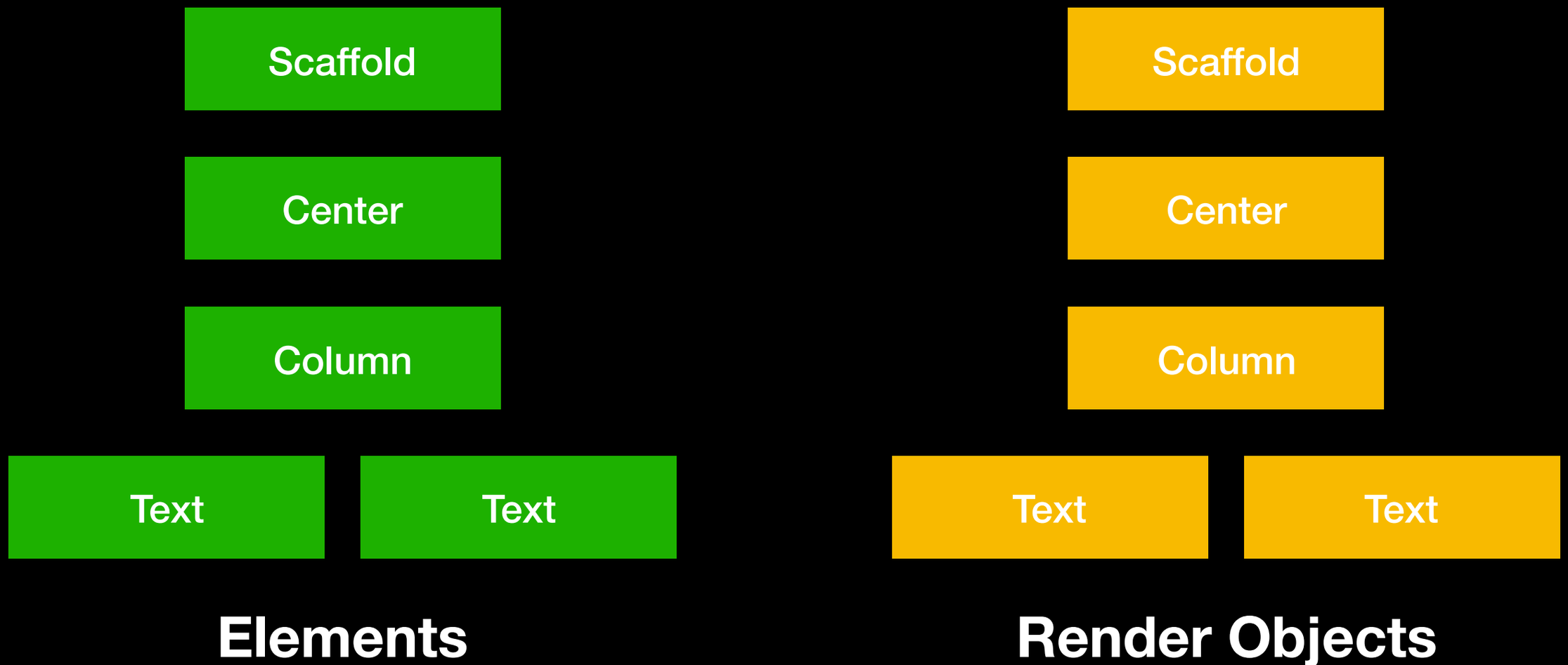

MyApp
MaterialApp
MyHomePage
Scaffold
Center
Column
Text
Text
AppBar
Text
FloatingActionButton
Icon


Scaffold
Center
Column
Text
Text

Hierarchy


Scaffold
Center
Column
Text
Text
Widgets
Scaffold
Center
Column
Text
Text
Elements

Hierarchy


Scaffold
Center
Column
Text
Text
Elements
Scaffold
Center
Column
Text
Text
Render Objects

Each control is represented in 3 trees

- Widgets tree
- Elements tree (by createElement() on the widget)
- RenderObjects tree (by createRenderObject() on the element)

What is Widget?

- A configuration for Element
- Always Stateless **!!!**
- Tiny and cheap for creation (still use **const**)

What is Element?

- Hold a reference to a Widget and the actual RenderObject
- Configure a specific location in the tree
- Manage Widget's lifecycle

You already work with Elements!

BuildContext

What is RenderObject?

- Contains all the logic for rendering
- Is abstracted all the way (can work with cartesian coordinates, yet with polar, 3D, 100500D, etc.)
- Quite expensive to create

The whole Flutter app acts like a huge RecyclerView or UICollectionView

5 min break ⌚
and let’s dive into layouts

Let’s talk about

RenderBox

What is RenderBox?

- A render object in a 2D cartesian coordinate system.
- The size of each box is expressed as a width and a height.
- Each box has its own coordinate system in which its upper left corner is placed at (0,0).
- Box layout is performed by passing a BoxConstraints down the tree.

Parent provide constraints,
child respond with the size.

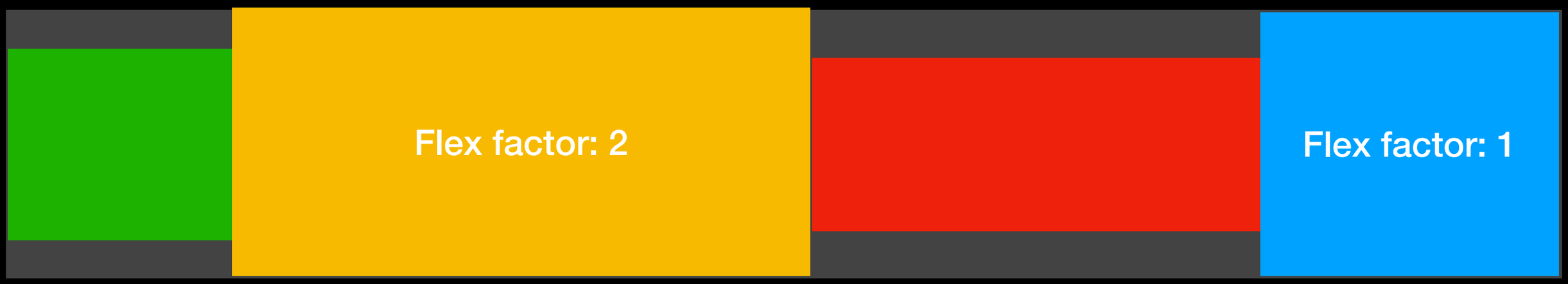
And that is how Flutter can build a screen in one step!

Layout algorithm is easy!

1. Layout Inflexible Children
2. Compute Free Space
3. Layout Flexible Children
4. Position Children

Row

- Inputs: overall minWidth, minHeight, and max Height constraints
- Outputs: overall width and height, sizes and positions for each child



Step1: Layout Inflexible Children

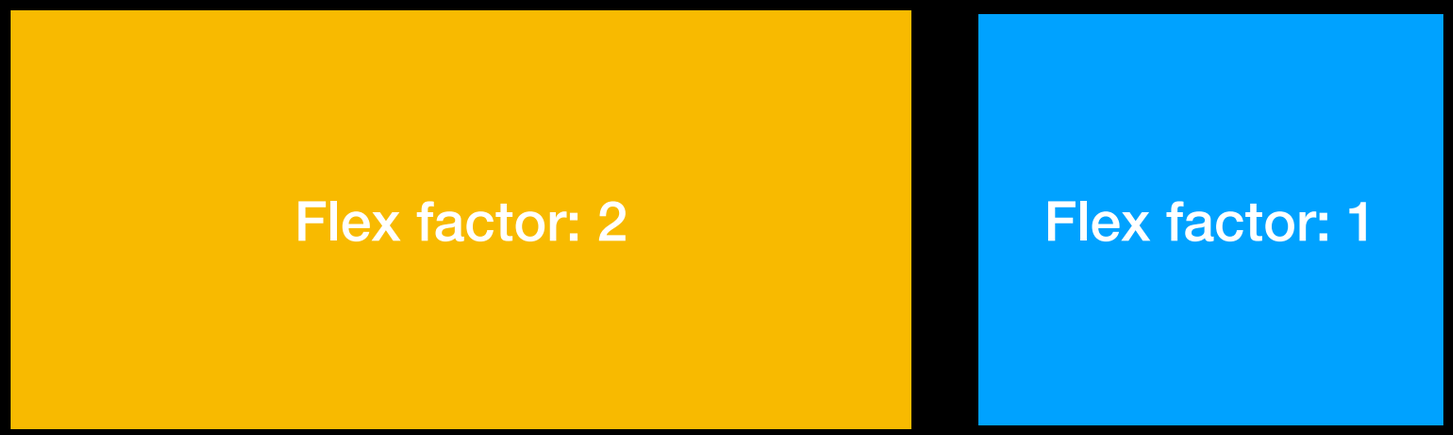
- Inputs for each inflexible child: minWidth=0, maxWidt=+inf, minHeight=0, maxHeight = incoming maxHeight

Step 2: Compute Free Space

- overall width = incoming maxWidth
- freeSpace = overall width - sum(width of inflexible children)
- spacePerFlex = freeSpace / sum(flex factor of flexible children)

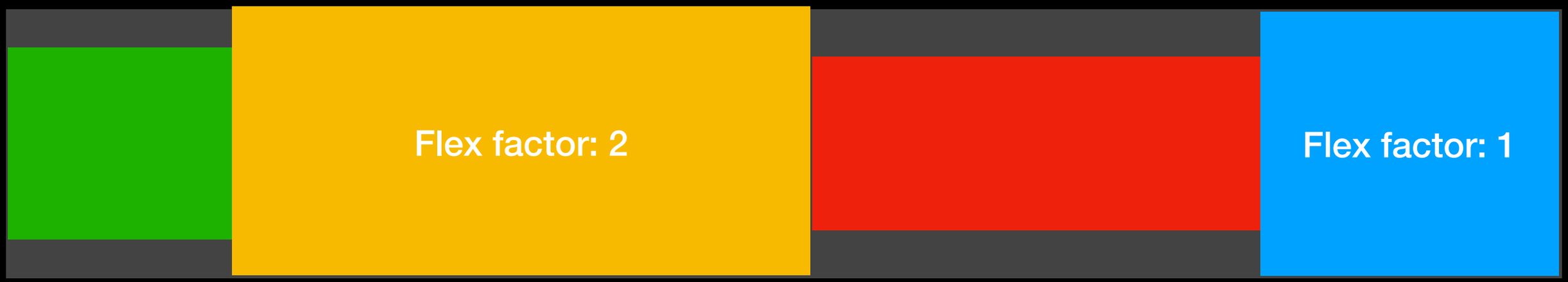
Step 3: Layout Flexible Children

- Inputs for each flexible child: minWidth = flexFactor * spacePerFlex, maxWidth = flexFactor * spacePerFlex, minHeight = 0, maxHeight = incoming maxHeight

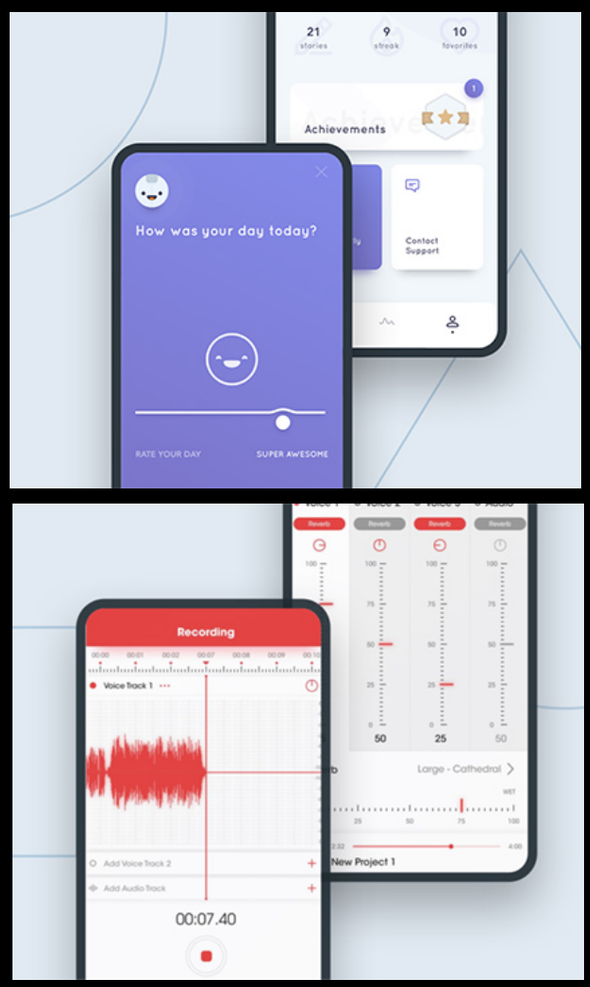


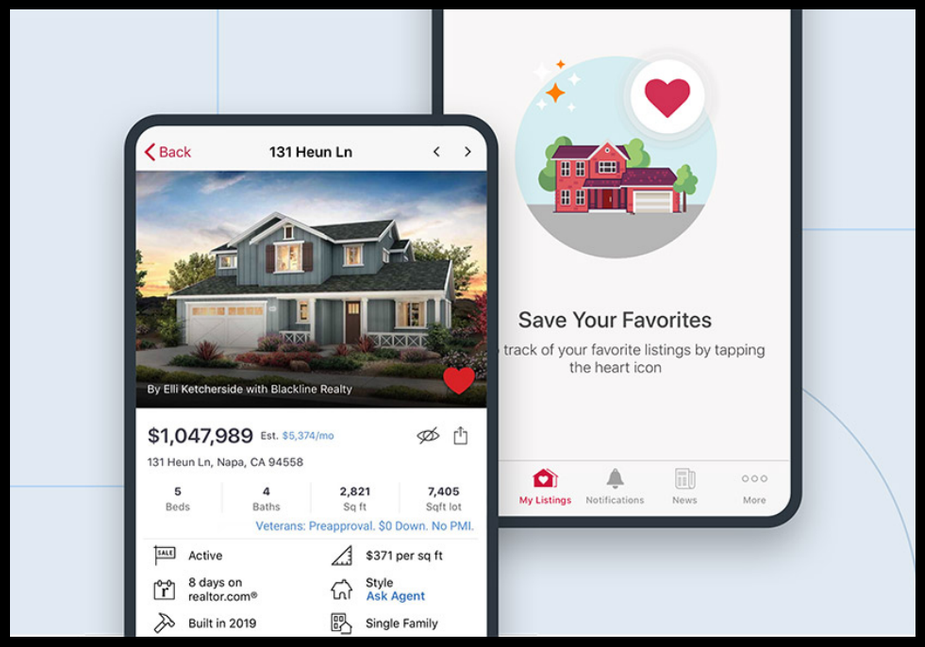
Step 4: Position Children

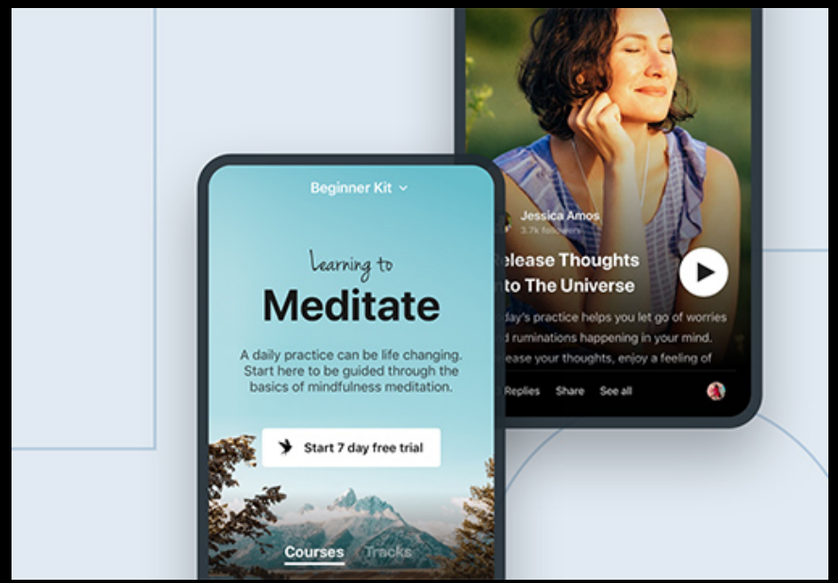
- Overall width = incoming maxWidth, overall height = max(height of children, incoming minHeight)



RenderBox + RenderSliver = 99% of all apps!







Each time you see such overflow, think of the algorithm!

More details here

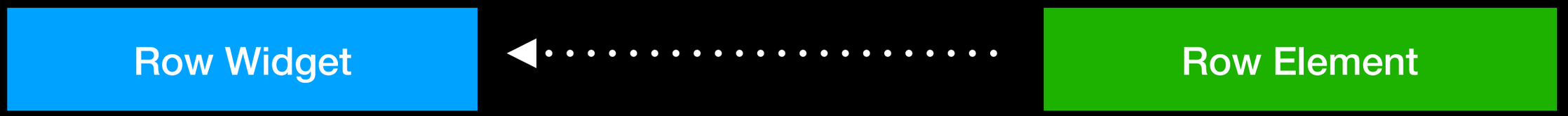
- https://flutter.dev/docs/resources/rendering

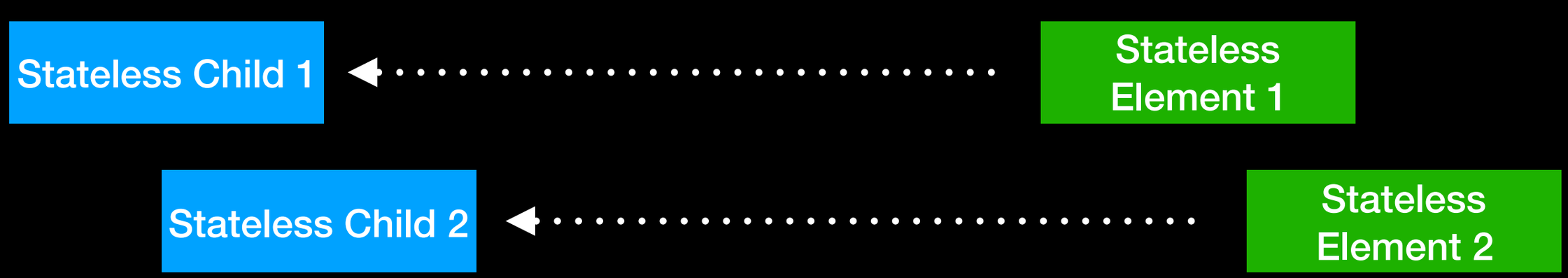
- https://youtu.be/UUfXWzp0-DU

5 min break ⌚

and let’s get back to Elements

Why care about Element?

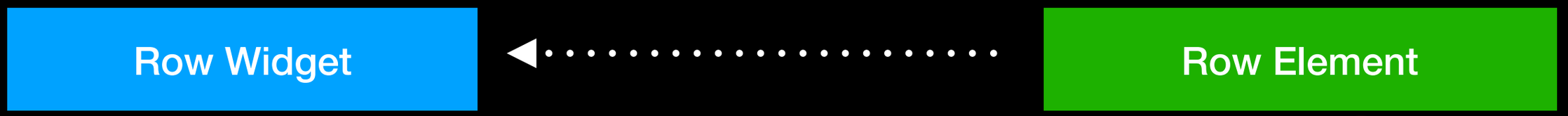


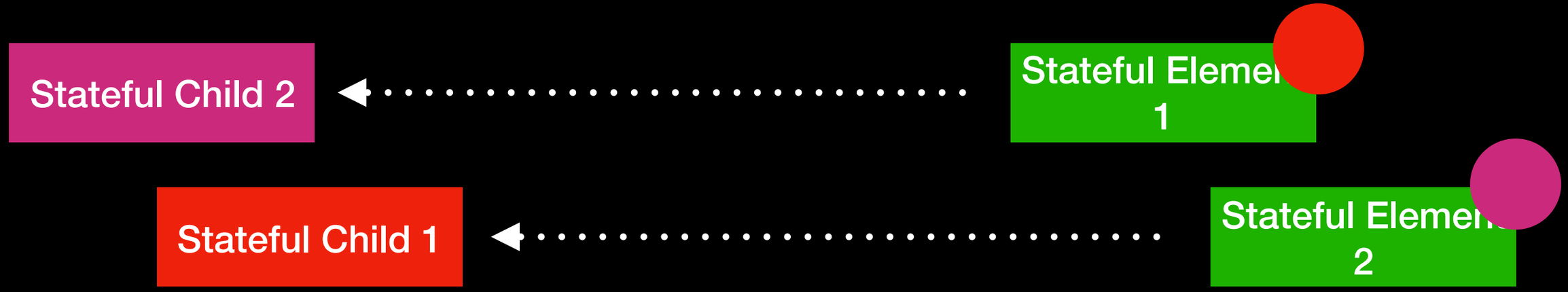


Flutter compares types and keys at one level of the tree at

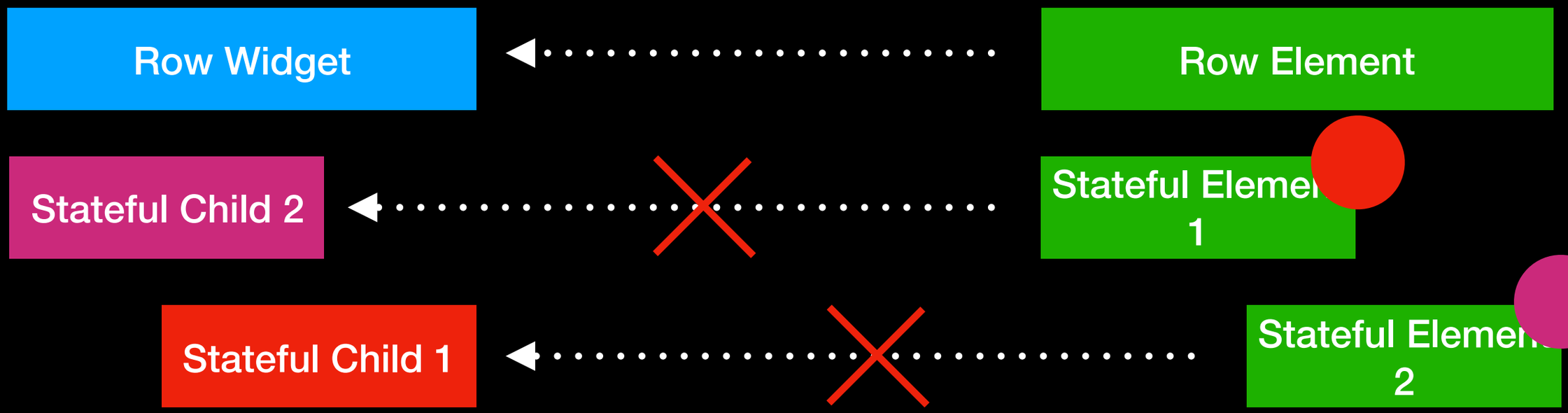
Why care about Element?

Why care about Element?

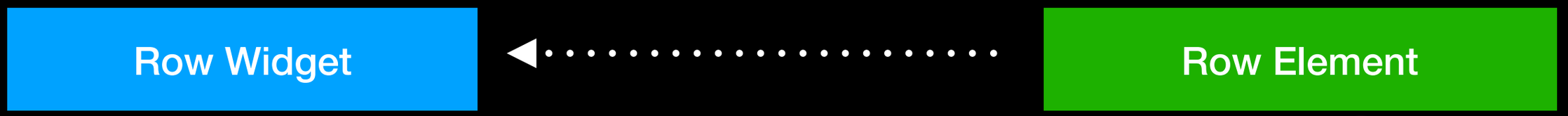


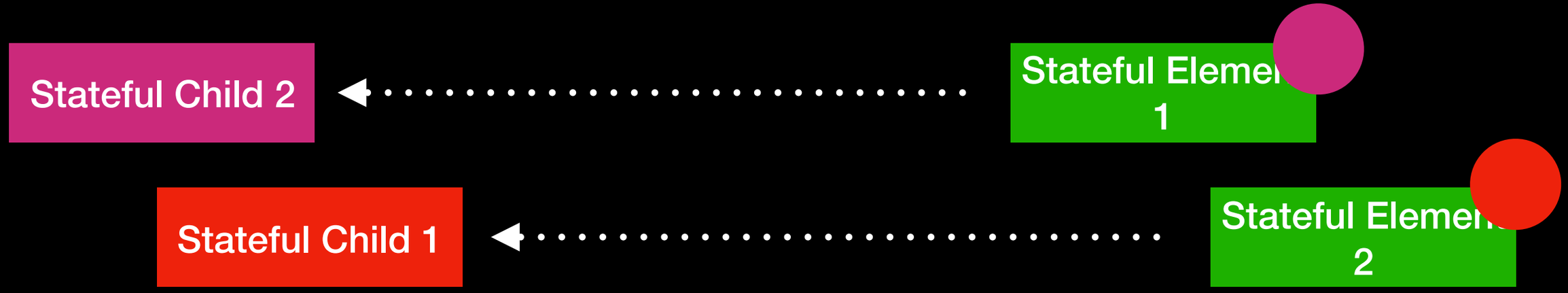


Why care about Element?



Why care about Element?





Flutter compares types and keys at one level of the tree at a time.

Let’s find the problem together!

Keys

- ValueKey
- ObjectKey
- UniqueKey
- LocalKey
- GlobalKey

Homework

- Fix the app!

- Communicate with your group, you'll have one mark

Links

- https://flutter.dev/docs/resources/inside-flutter
- https://github.com/flutter/flutter/wiki/The-Engine-architecture
- https://medium.com/flutter-community/the-layer-cake-widgets-elements-renderobjects-7644c3142401
- https://youtu.be/kn0EOS-ZiIc