Flutter Framework

PV239

Faculty of Informatics, Masaryk University

Rastislav Mirek,
Vural a. s. & TypeSoft s. r. o. & CzechInvest

Flutter Introduction & Motivation

Framework Characteristics

- UI Framework built on top of Dart language
- Made & supported by Google
- “Very” and truly multi-platform
- Declarative paradigm
  - no “build in” MVC, MVVM, …
- Relatively new, increasingly popular
- Very fast adoption rates
- “Everything is a Widget”
- Fallback to native
- No separate script, Dart code is the script

Main Selling Points

- Code once, run **everywhere**
- Fast development
- Declarative (function of state) → fewer bugs
- Rendering & other performance on pair with native, even for animations
- Developers love it, “It’s a very satisfying tech”
- Evolves fast, e.g. null-safety
- Already a good package ecosystem

Trend

Huge hype: There are over 400,000 Flutter apps in the Play Store alone. More half of that number was added in last 6 months.

Demo

Ultimate Framework for all UI work?

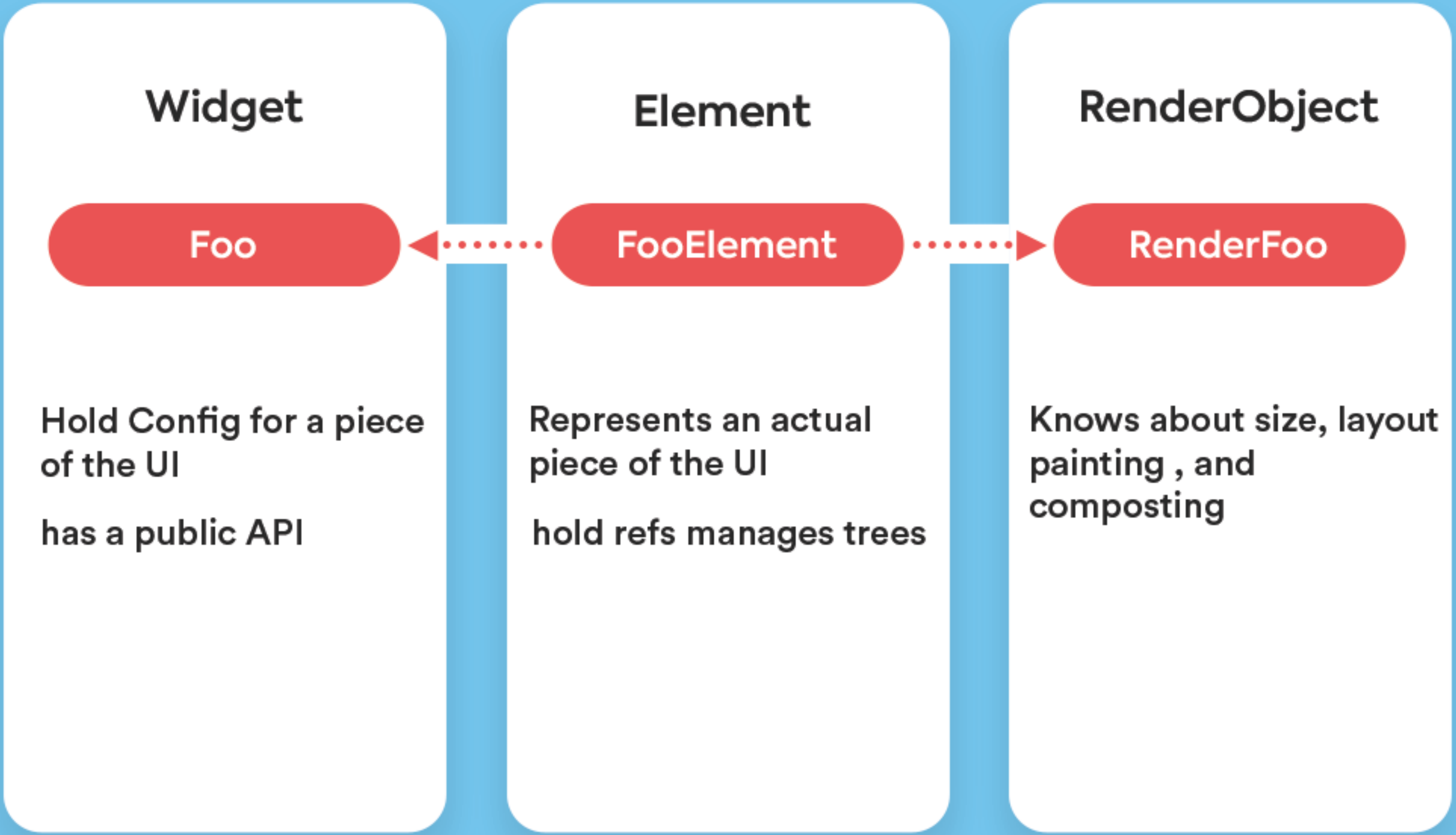
Biggest Downsides

- Higher minimal memory consumption
  - Due to Skia and framework memory footprint
  - Improves over time
- No 3D rendering
  - Native 3D rendering needs to be used for each target platform
- Only affine render transforms, no bending
  - Few other framework support it
- Desktop platforms are still in beta
- Awkward approach to text selection on web

How Flutter works

Architecture Principles

- Dart compiles to JavaScript, C, …
- 3 parallel trees represent views hierarchy
  - Widget tree
  - Elements tree
  - Render tree
- Rendering via Skia
  - Cross-platform 2D library used e.g. for rendering in Chrome
  - Field tested, very fast
- “Flutter is just a library of widgets on top of Dart and Skia"



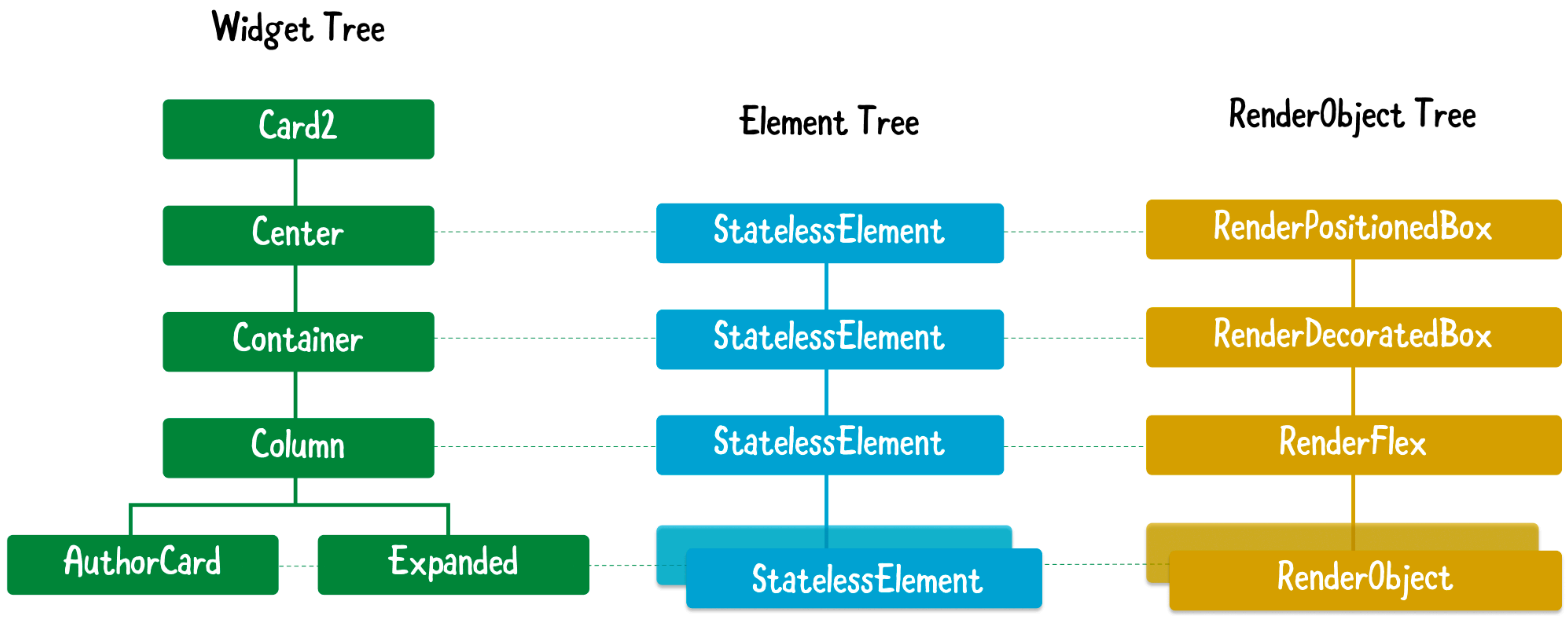
Hold Config for a piece of the UI

has a public API

Represents an actual piece of the UI

hold refs manages trees

Knows about size, layout painting , and composting



Widget Tree
Card2
Center
Container
Column
AuthorCard
Expanded
Element Tree
StatelessElement
StatelessElement
StatelessElement
StatelessElement
RenderObject Tree
RenderPositionedBox
RenderDecoratedBox
RenderFlex
RenderObject

Widgets

- 3 types
  - Stateless
  - Statefull
  - Inherited
- Easier customisation compared to native platforms
- Very natural composition and code reuse

- Ready to use widgets:
  - Material
  - “Cupertino”
  - **Many** packages

Info Mix

- Cooperation with hosting platform (native interoperability) is via Platform channels
- Complexity, as with any declarative framework is in State Management
  - Many state management packages: Notifier with Provider, Block, GetX, Redux, …
  - Most use inherited widget
- Easy theming
- Single threaded but has Isolates
- Perfect integration with other Google dev tools & APIs

Thank you