SAPUI5 Nested Components

Graham Robinson
Yelcho Systems Consulting

*Photo by Martin Gillet

Yelcho
Systems Consulting
Graham Robinson
Principal Consultant
Mobile +61 412 402 441
Email graham@yelcho.com.au

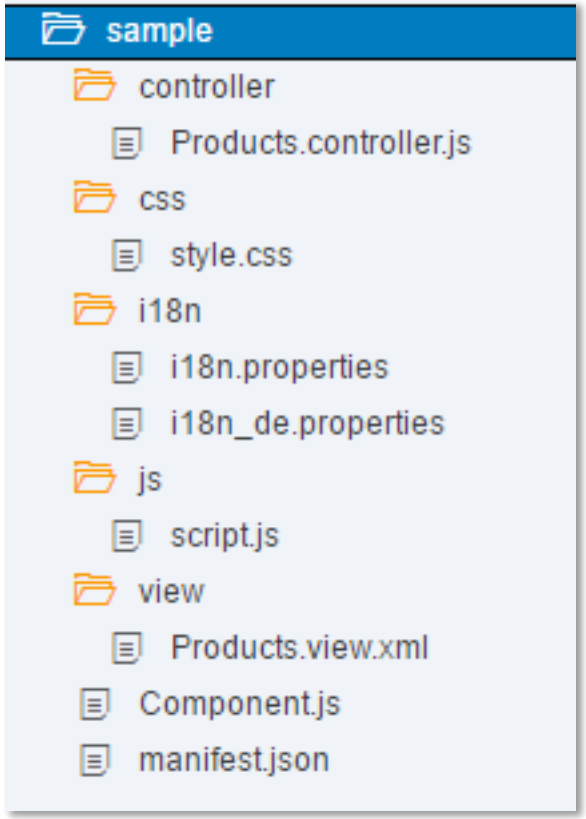
@grahamrobbo

SAP® Mentors

Photo by Frank Hurley – used with permission

SAPUI5 Components

- Components are independent and reusable parts used in SAPUI5 applications.
- An application can use components from different locations from where the application is running.
- Thus, components can be developed by different development teams and be used in different projects.
- Components also support the encapsulation of closely related parts of an application into a particular component.
- This makes the structure of an application and its code easier to understand and to maintain.



Filter
- What's New in SAPUI5
- Read Me First
- Get Started: Setup, Tutorials, and Demo Apps
- Essentials
  - Bootstrapping: Loading and Initializing
  - Structuring: Components and Descriptor
    - Components
      - Component.js File
      - Using and Nesting Components
      - Declarative API for Initial Components
      - Handling IDs in UI Components
      - The Owner Component
      - Advanced Concepts for SAPUI5 Components
    - Descriptor for Applications, Components, and Libraries (manifest.json)
  - Model View Controller (MVC)
  - Data Binding
  - Reusing UI Parts: Fragments

Components

Edit on GitHub

Components are independent and reusable parts used in SAPUI5 applications.

An application can use components from different locations from where the application is running. Thus, components can be developed by different development teams and be used in different projects. Components also support the encapsulation of closely related parts of an application into a particular component. This makes the structure of an application and its code easier to understand and to maintain.

> **Note**
> Constraints due to cross-origin issues also apply to components.

SAPUI5 provides the following two types of components:

- Faceless components (class: `sap.ui.core.Component`)

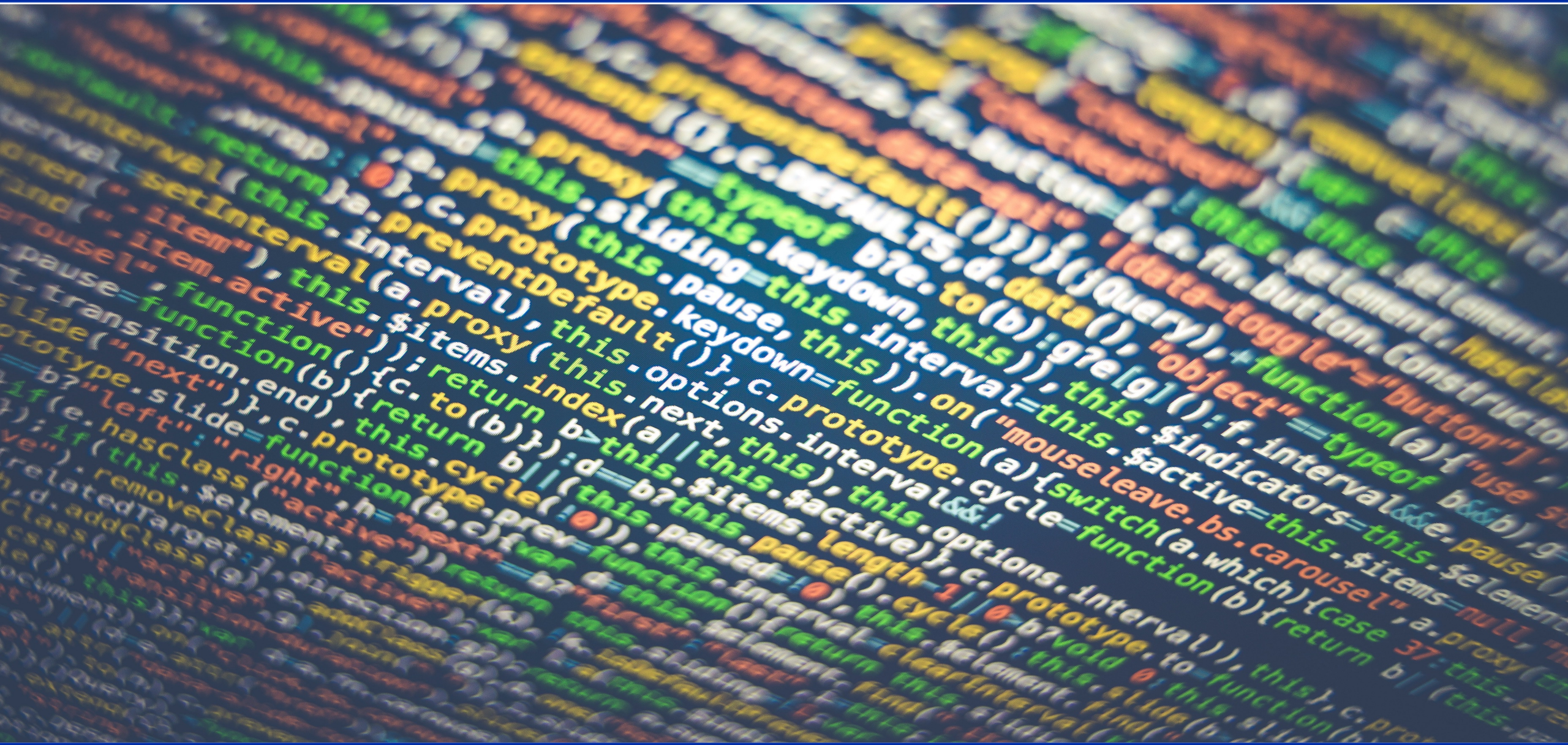
  Faceless components do **not** have a user interface and are used for coding where no UI elements are needed. Please consider that a faceless component can't be added to a `ComponentContainer`. For more information, see the API Reference: `sap.ui.core.ComponentContainer`.

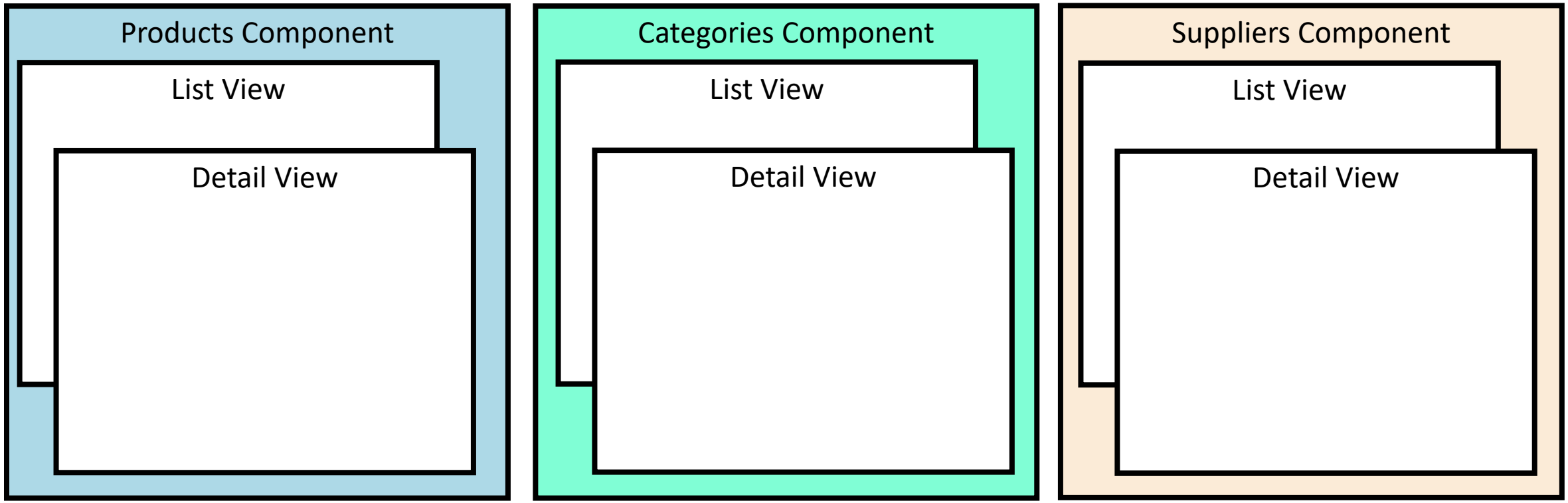
- UI components (class: `sap.ui.core.UIComponent`)

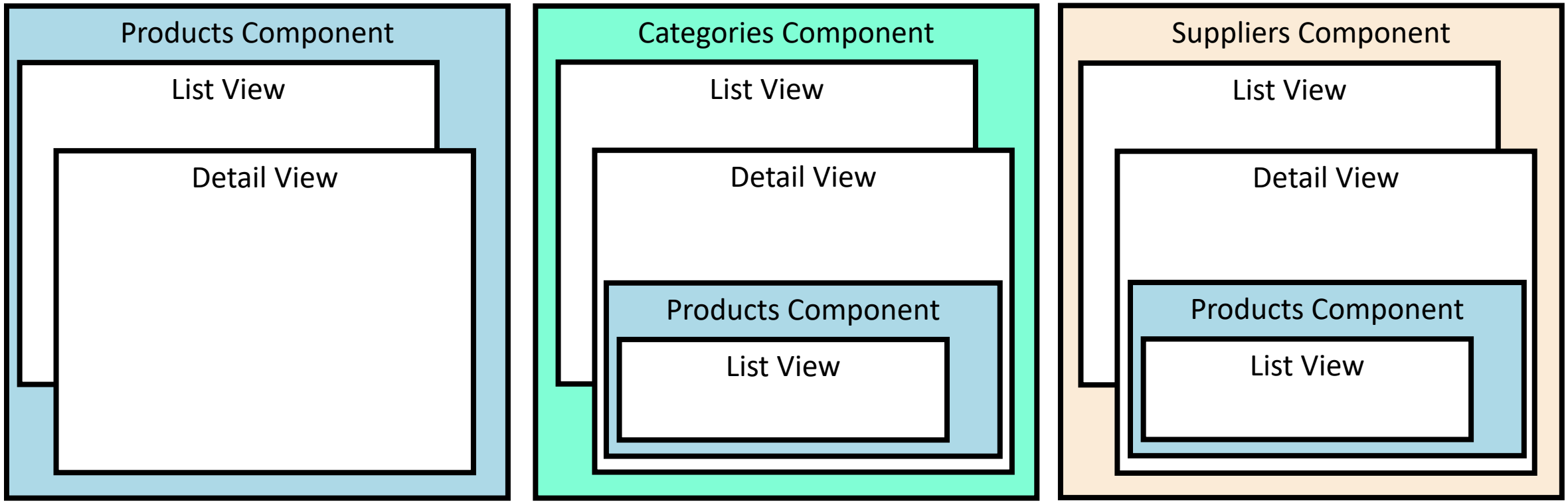
  UI components extend components and add rendering functionality to the component. They represent a screen area or element on the user interface, for example, a button or a shell, along with the respective settings and metadata. `sap.ui.core.UIComponent` extends `sap.ui.core.Component` and adds rendering functionality to the component.

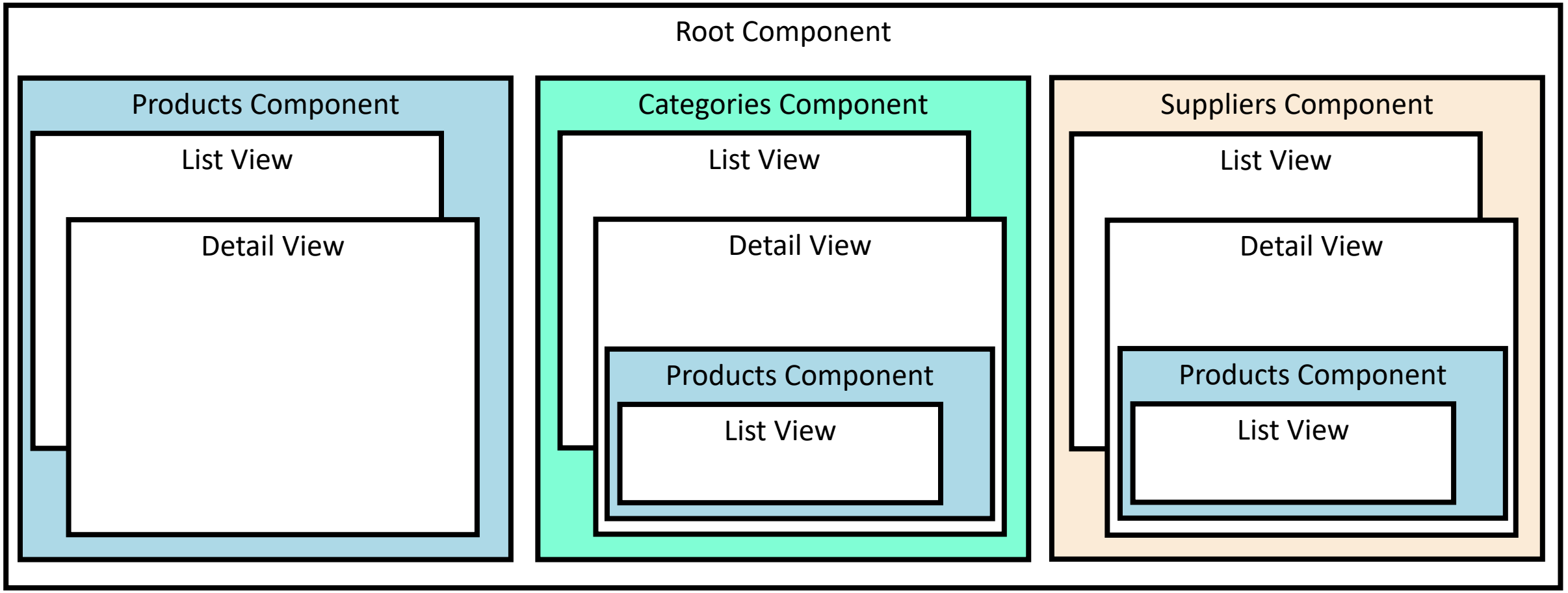
The `sap.ui.core.Component` class is the base class and provides the metadata for both types of components. To extend the functionality, components can inherit from their base class or from another component.

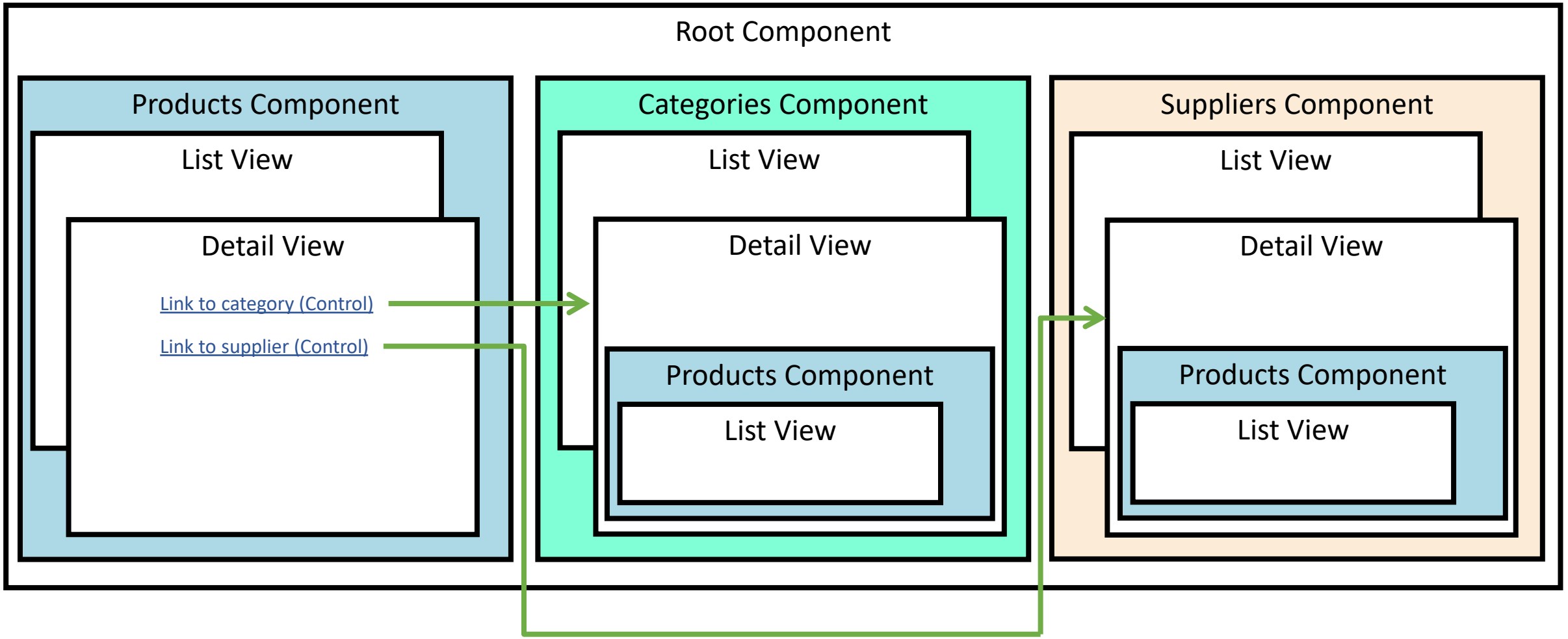
https://github.com/grahamrobbo/MST2023

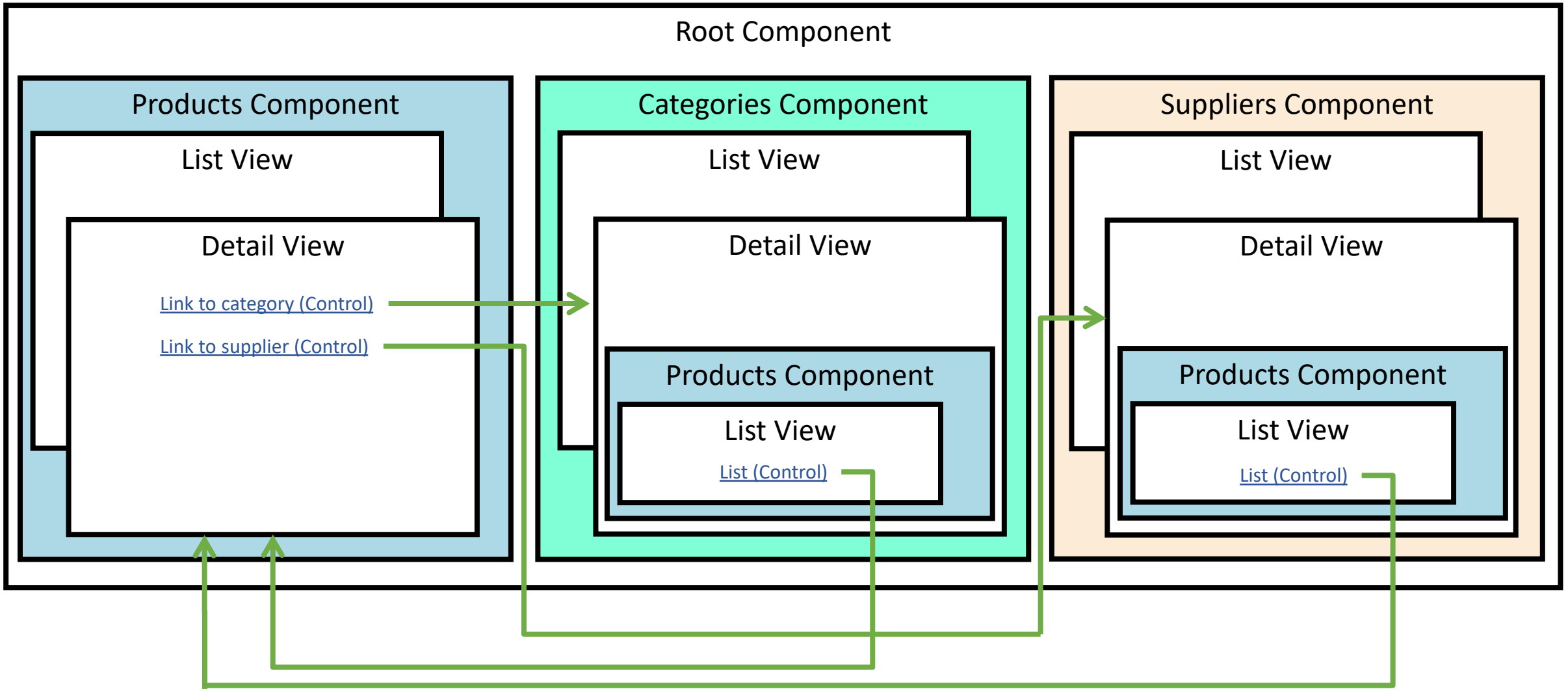



Products Component
List View
Detail View
Categories Component
List View
Detail View
Suppliers Component
List View
Detail View


Products Component
List View
Detail View
Categories Component
List View
Detail View
Products Component
List View
Suppliers Component
List View
Detail View
Products Component
List View


Root Component
Products Component
List View
Detail View
Categories Component
List View
Detail View
Products Component
List View
Suppliers Component
List View
Detail View
Products Component
List View


Root Component
Products Component
List View
Detail View
Link to category (Control)
Link to supplier (Control)
Categories Component
List View
Detail View
Products Component
List View
Suppliers Component
List View
Detail View
Products Component
List View


Root Component
Products Component
List View
Detail View
Link to category (Control)
Link to supplier (Control)
Categories Component
List View
Detail View
Products Component
List View
List (Control)
Suppliers Component
List View
Detail View
Products Component
List View
List (Control)

Filter
- What's New in SAPUI5
- Read Me First
- Get Started: Setup, Tutorials, and Demo Apps
- Essentials
  - Bootstrapping: Loading and Initializing
  - Structuring: Components and Descriptor
    - Components
      - Component.js File
      - Using and Nesting Components
      - Declarative API for Initial Components
      - Handling IDs in UI Components
      - The Owner Component
      - Advanced Concepts for SAPUI5 Components
    - Descriptor for Applications, Components, and Libraries (manifest.json)
  - Model View Controller (MVC)
  - Data Binding
  - Reusing UI Parts: Fragments

Components

Edit on GitHub

Components are independent and reusable parts used in SAPUI5 applications.

An application can use components from different locations from where the application is running. Thus, components can be developed by different development teams and be used in different projects. Components also support the encapsulation of closely related parts of an application into a particular component. This makes the structure of an application and its code easier to understand and to maintain.

> **Note**
> Constraints due to cross-origin issues also apply to components.

SAPUI5 provides the following two types of components:

- Faceless components (class: `sap.ui.core.Component`)

  Faceless components do **not** have a user interface and are used for coding where no UI elements are needed. Please consider that a faceless component can't be added to a `ComponentContainer`. For more information, see the API Reference: `sap.ui.core.ComponentContainer`.

- UI components (class: `sap.ui.core.UIComponent`)

  UI components extend components and add rendering functionality to the component. They represent a screen area or element on the user interface, for example, a button or a shell, along with the respective settings and metadata. `sap.ui.core.UIComponent` extends `sap.ui.core.Component` and adds rendering functionality to the component.

The `sap.ui.core.Component` class is the base class and provides the metadata for both types of components. To extend the functionality, components can inherit from their base class or from another component.

https://github.com/grahamrobbo/MST2023

*Photo by Martin Gillet

How to Connect with Me

E: graham@yelcho.com.au

M: +61 412 402 441

Li: linkedin.com/in/grahamrobbo/

@grahamrobbo