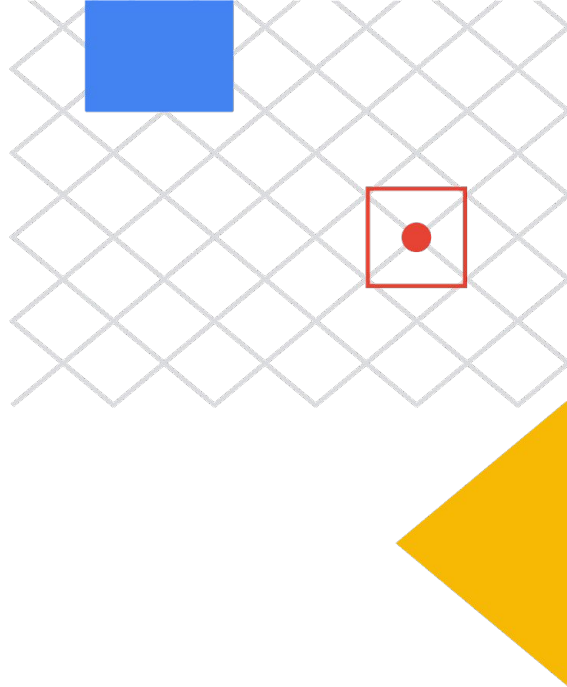


特别提醒: 由于本幻灯片和活动举办时间为 2018 年, 其中的内容无法保证最新和仍然可用。
如果有任何问题, 请访问 flutter.cn 或其他方式与我们联系。

Flutter Performance Profiling and Theory Flutter的性能测试与理论

Xiao Yu
于潇



Talk Goals

-

前言

Preface

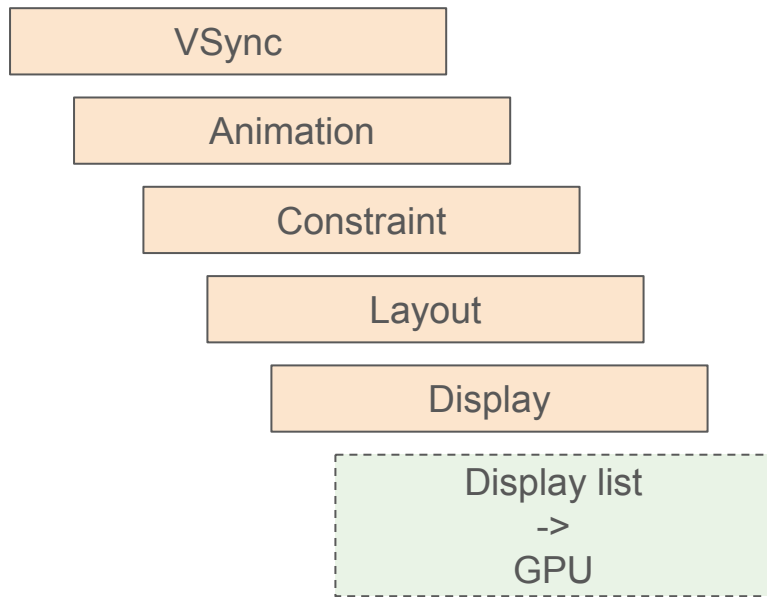
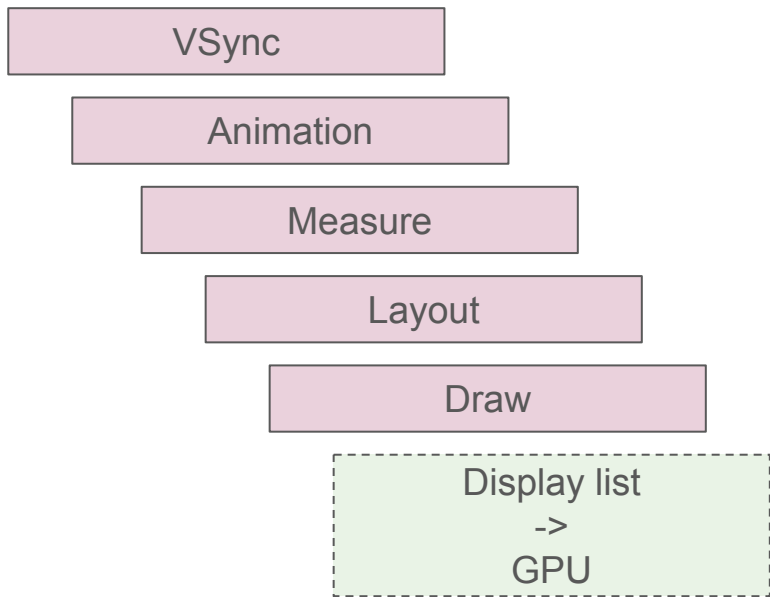
- 不是入门题材
- 同场5:30请关注Flutter图像系统性能论坛

什么是应用性能

What is Rendering Performance

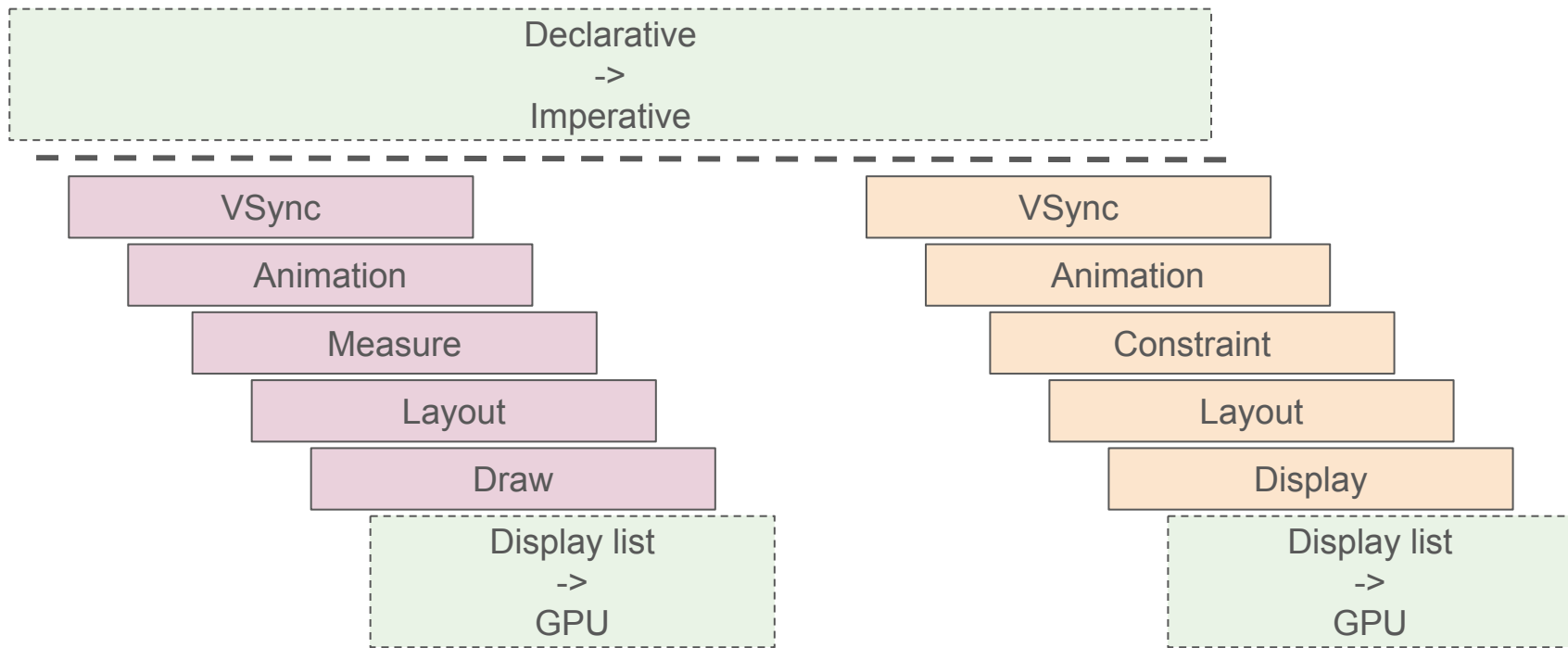
什么是应用性能

What is Rendering Performance



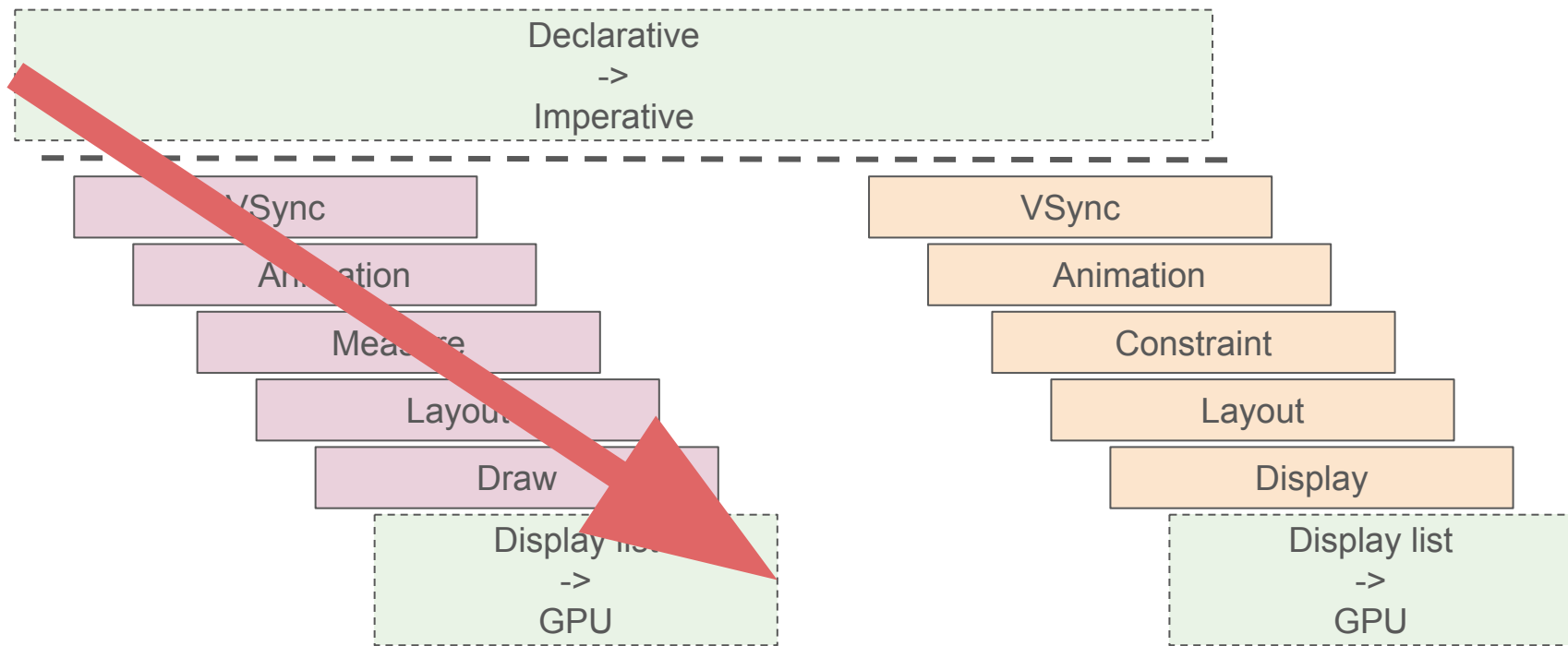
什么是应用性能

What is Rendering Performance



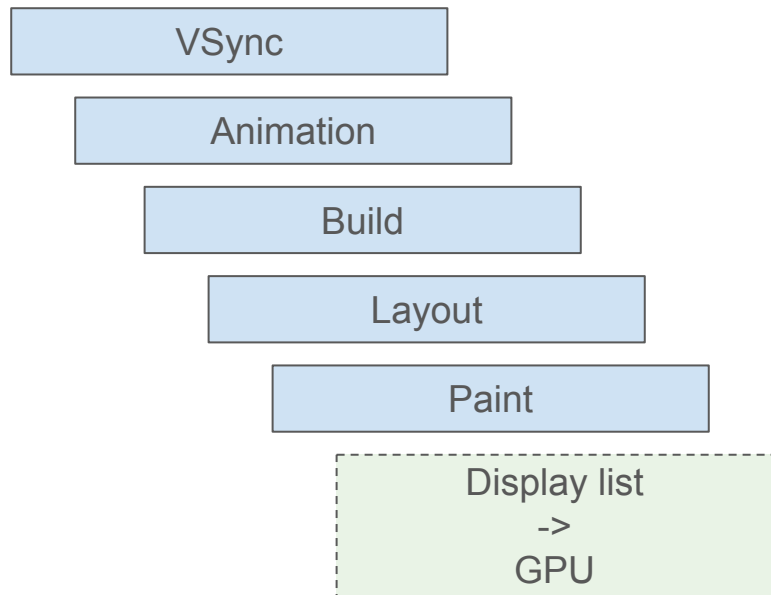
什么是应用性能

What is Rendering Performance



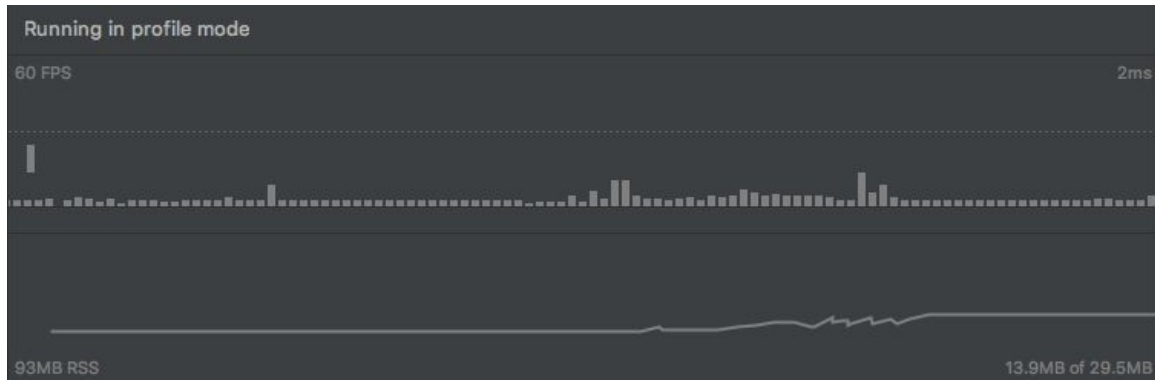
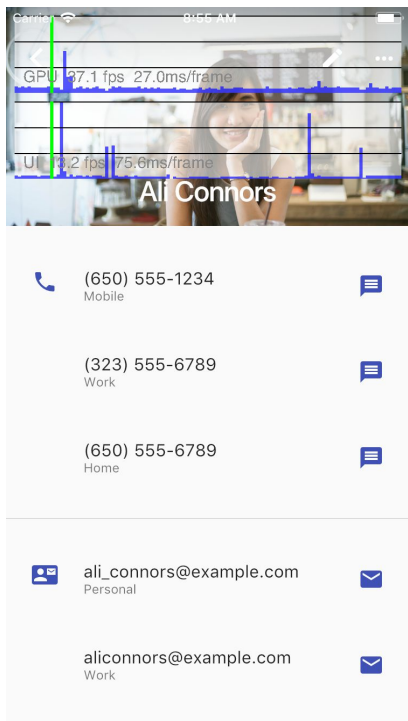
Flutter的渲染阶段

Flutter's Rendering Pipeline



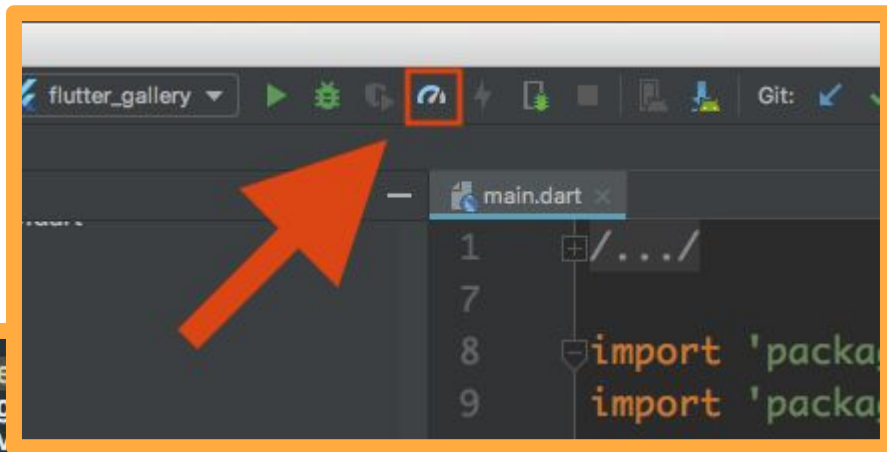
检测Flutter的渲染

Instrumenting Flutter's Rendering



Profile模式

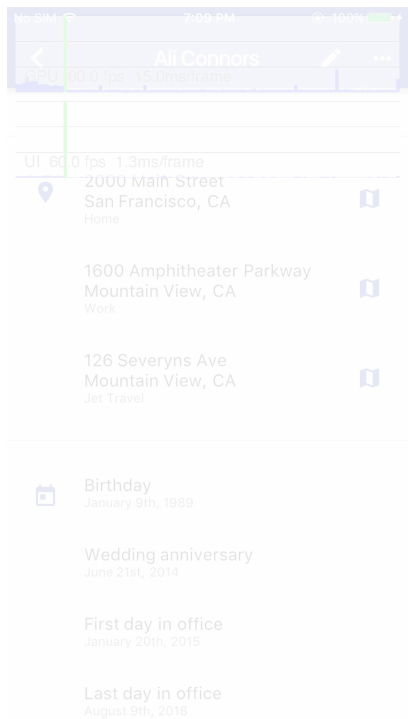
Profile Mode



```
1 ~/de
Launching
Found sav
Signing iOS app for device deployment using developer identity: "iPhone Developer Google Development (3
Manually selecting Google's mobile provisioning profile (see go/google-flutter-signing).
Running pod install...
Starting Xcode build...
  Building Dart code...
flutter_gallery flutter run --profile
```

检测Flutter的渲染

Instrumenting Flutter's Rendering

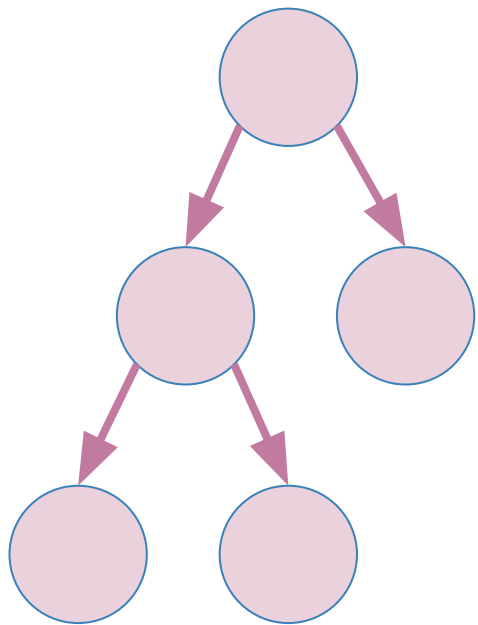


DEMO



渲染责任部件

Rendering Actors

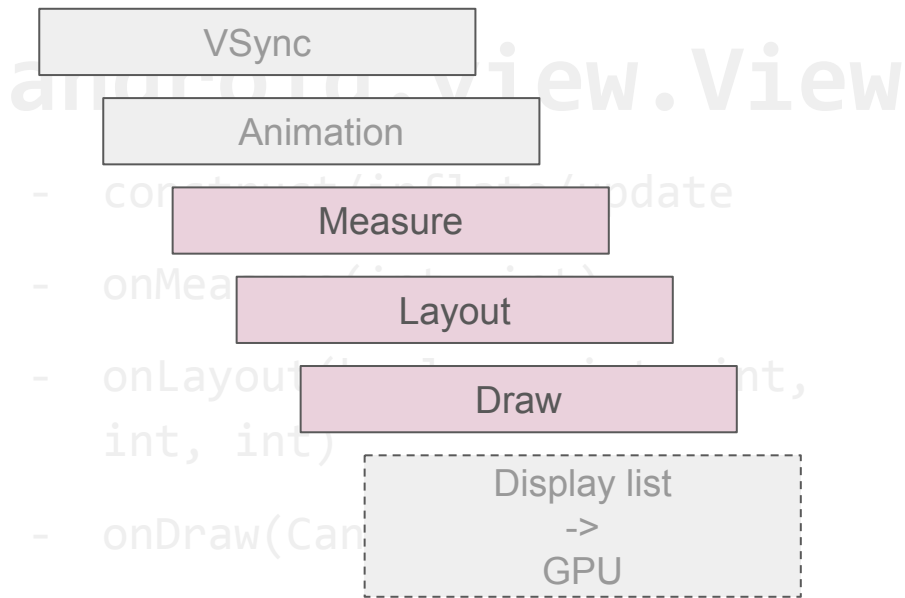
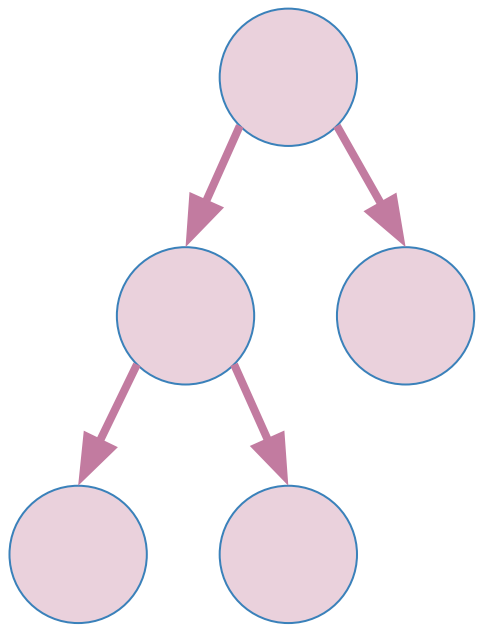


android.view.View

- construct/inflate/update
- onMeasure(int, int)
- onLayout(boolean, int, int, int, int)
- onDraw(Canvas)

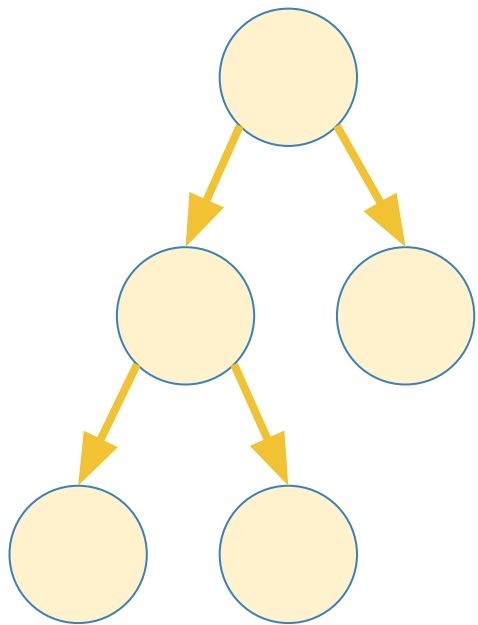
渲染责任部件

Rendering Actors



渲染责任部件

Rendering Actors

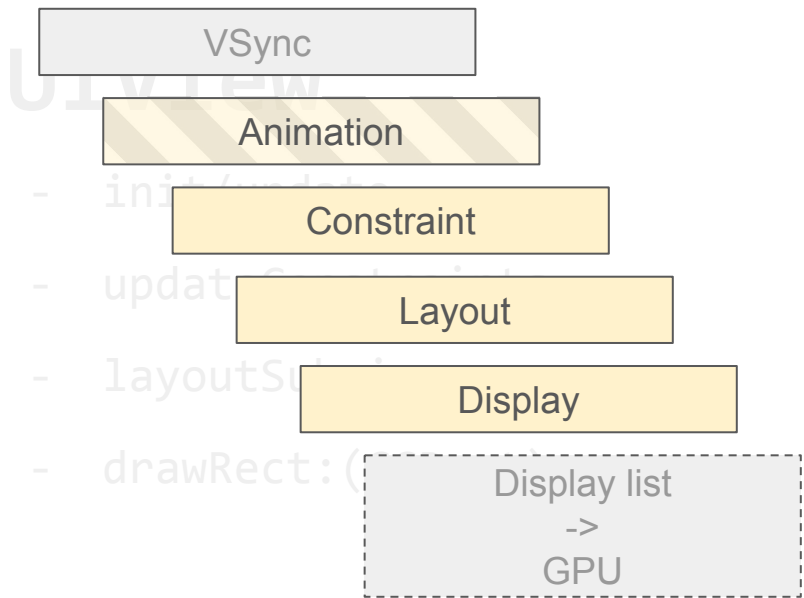
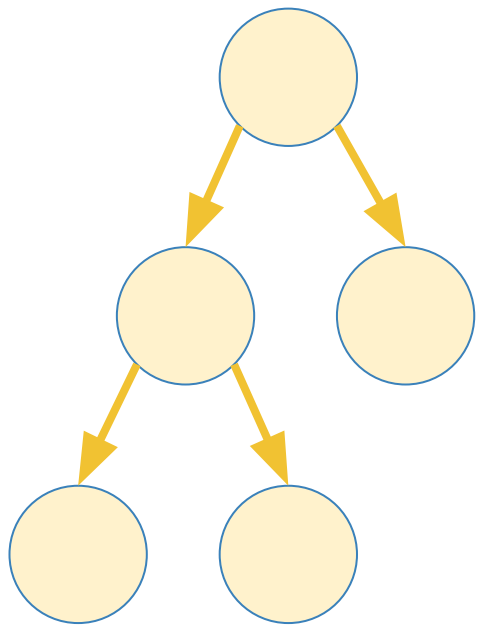


UIView

- `init/update`
- `updateConstraints`
- `layoutSubviews`
- `drawRect:(CGRect)`

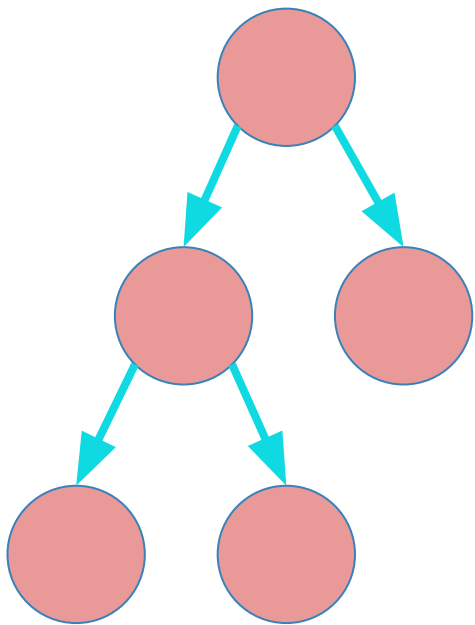
渲染责任部件

Rendering Actors



Flutter的渲染责任部件

Flutter's Rendering Actors

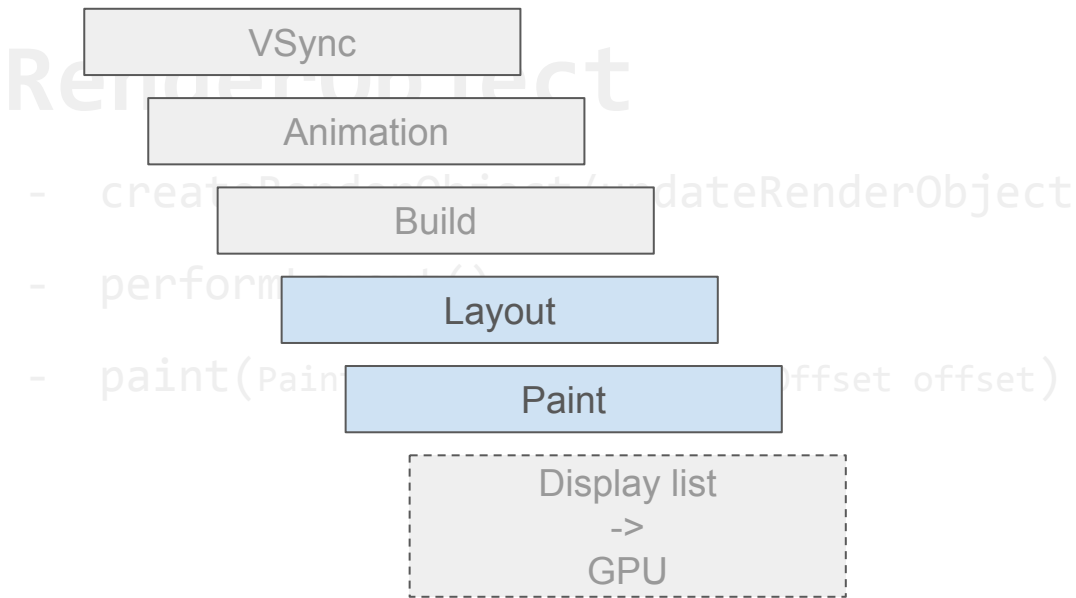
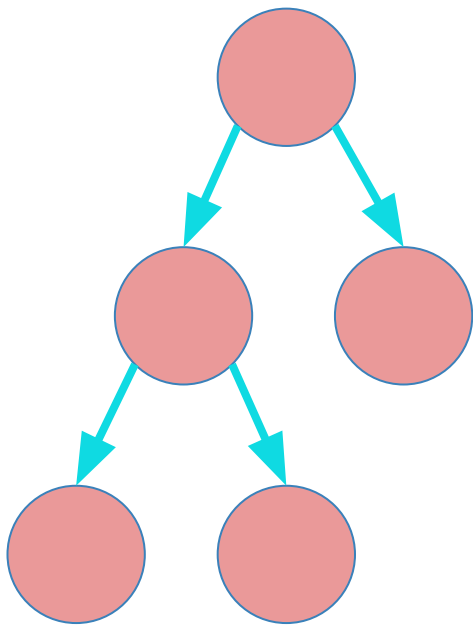


RenderObject

- createRenderObject/updateRenderObject
- performLayout()
- paint(PaintingContext context, Offset offset)

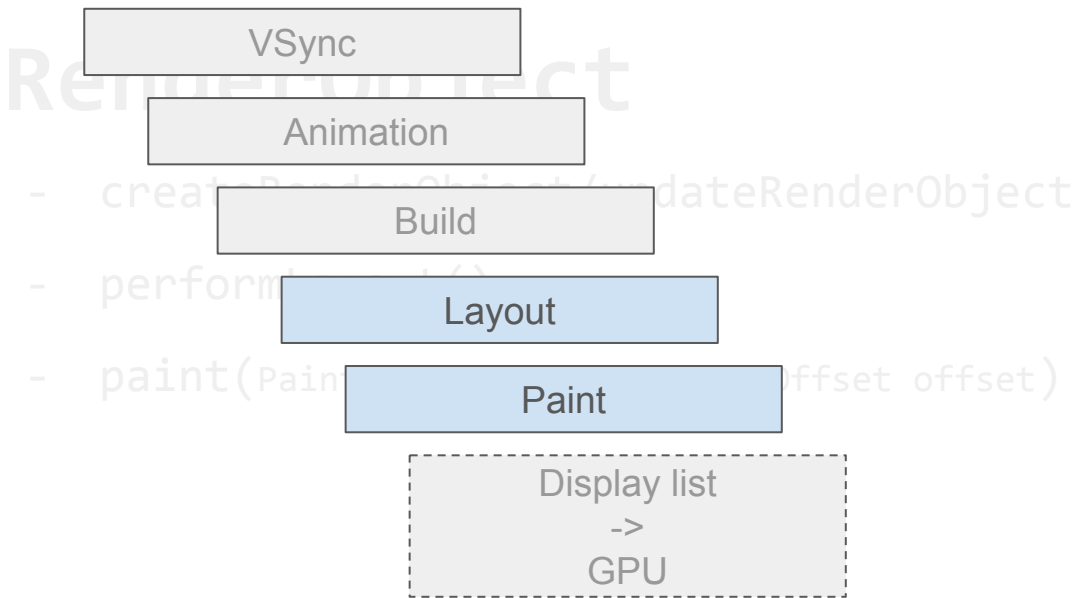
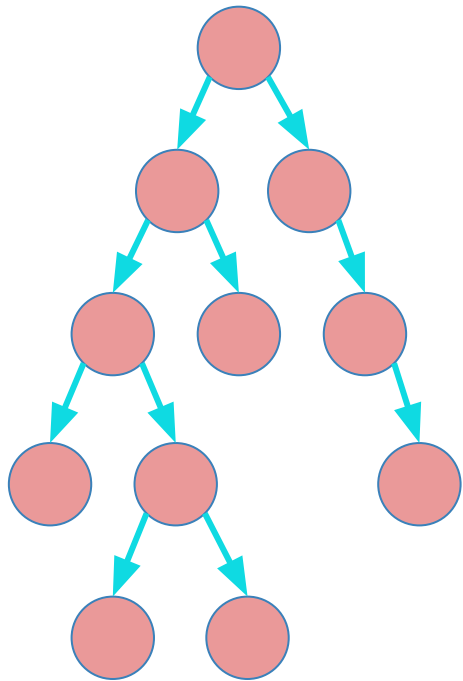
Flutter的渲染责任部件

Flutter's Rendering Actors



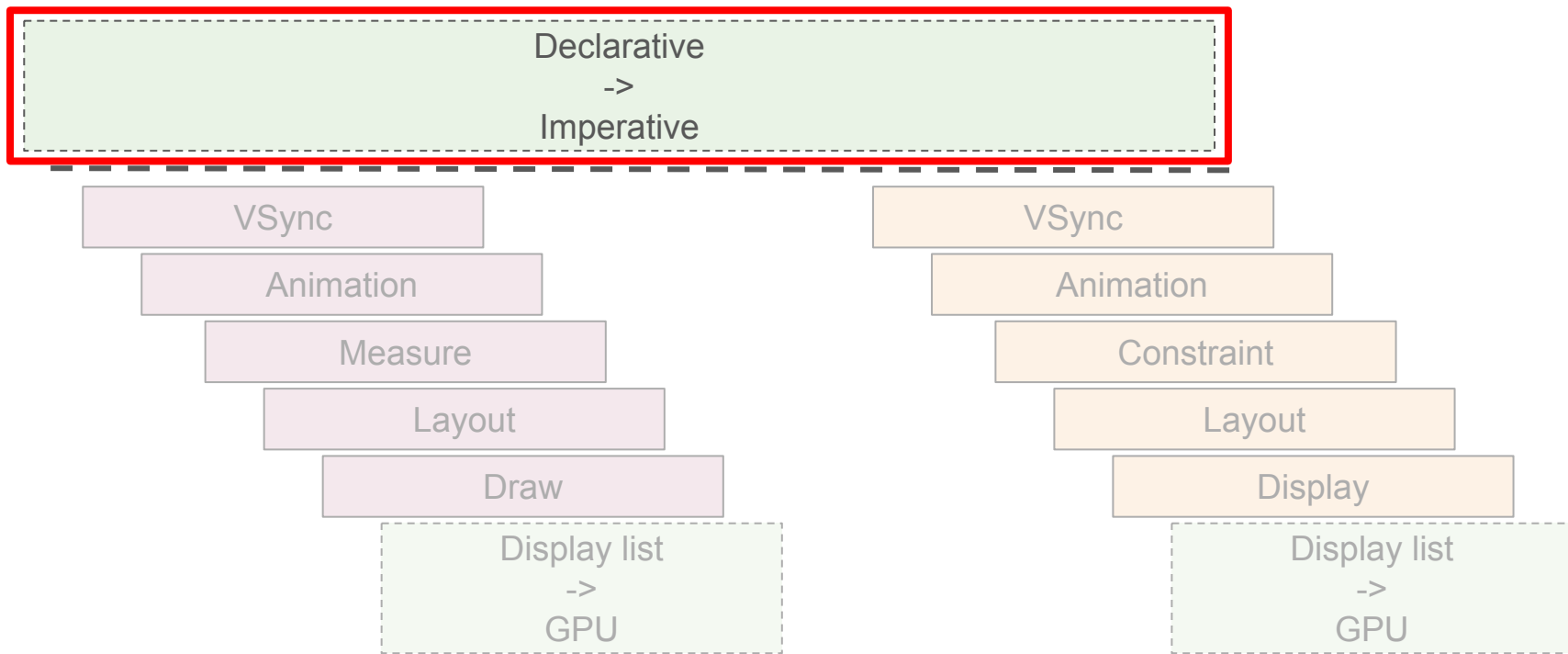
Flutter的渲染责任部件

Flutter's Rendering Actors



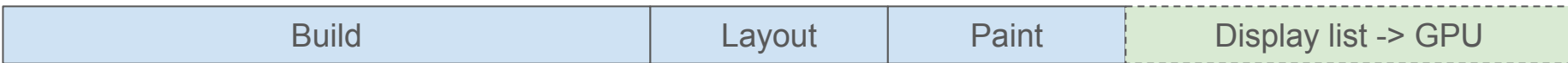
Flutter的声明式UI

Flutter's Declarative UI



Flutter的树与渲染阶段

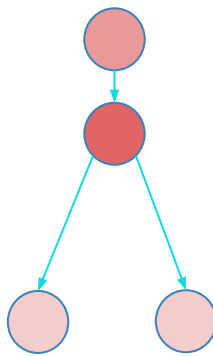
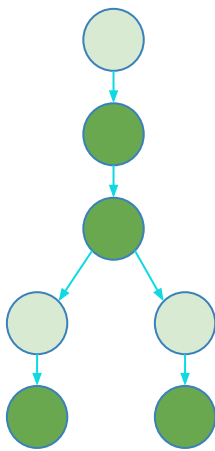
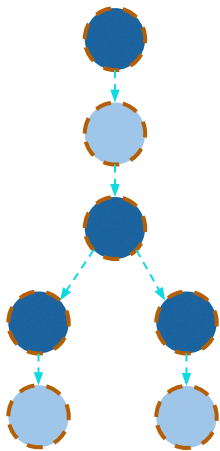
Flutter's Trees and Rendering Phases



Flutter的树与渲染阶段

Flutter's Trees and Rendering Phases

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer



Flutter的树与渲染阶段

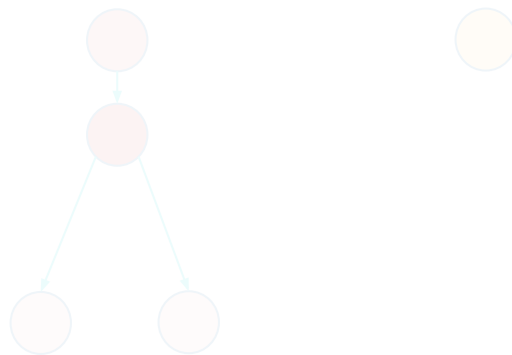
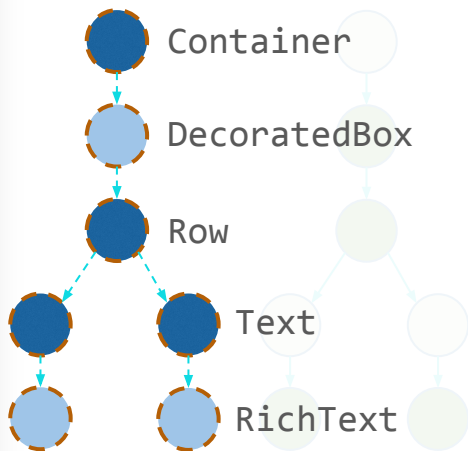
Flutter's Trees and Rendering Phases

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        './1.png'  
      ),  
      Text('A'),  
    ],  
  ),  
)
```

Image

RawImage

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer

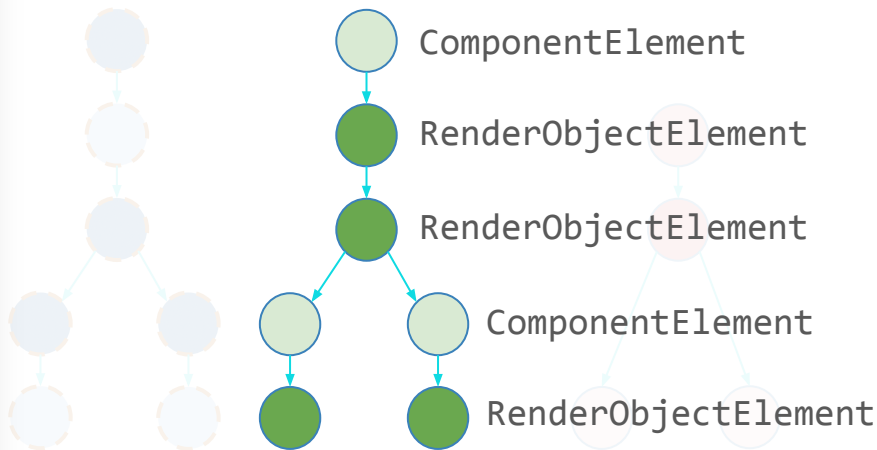


Flutter的树与渲染阶段

Flutter's Trees and Rendering Phases

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        './1.png'  
      ),  
      Text('A'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer

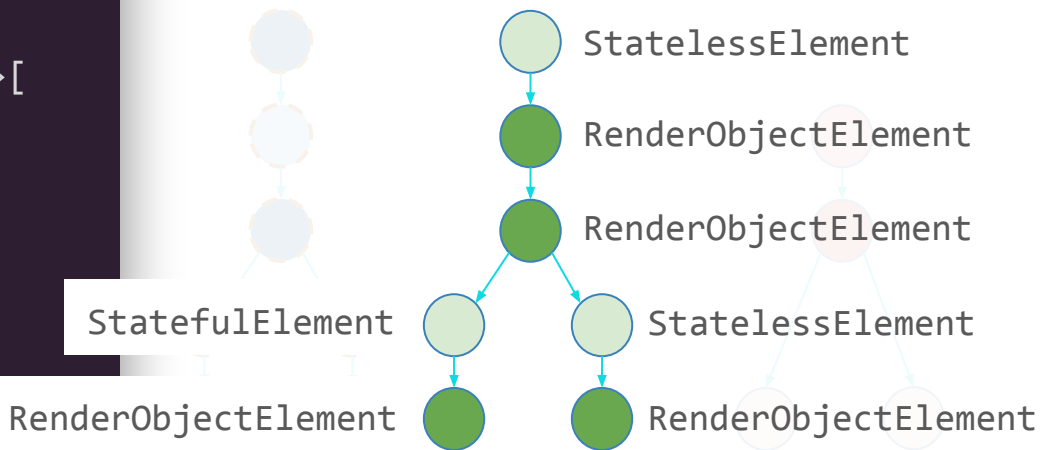


Flutter的树与渲染阶段

Flutter's Trees and Rendering Phases

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        './1.png'  
      ),  
      Text('A'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer



Flutter's Trees and Rendering Phases

RenderObjectWidget

The diagram illustrates the relationship between different widget types in Flutter. It shows a hierarchy of widgets: StatelessWidget, RenderObjectWidget, and RenderObjectWidget. The diagram uses colored circles (blue, green, red, yellow) to represent different widget types and dashed arrows to show relationships.

- StatelessWidget** (Blue circle)
- RenderObjectWidget** (Green circle)
- RenderObjectWidget** (Red circle)
- StatelessWidget** (Blue circle)
- RenderObjectWidget** (Green circle)
- RenderObjectWidget** (Yellow circle)

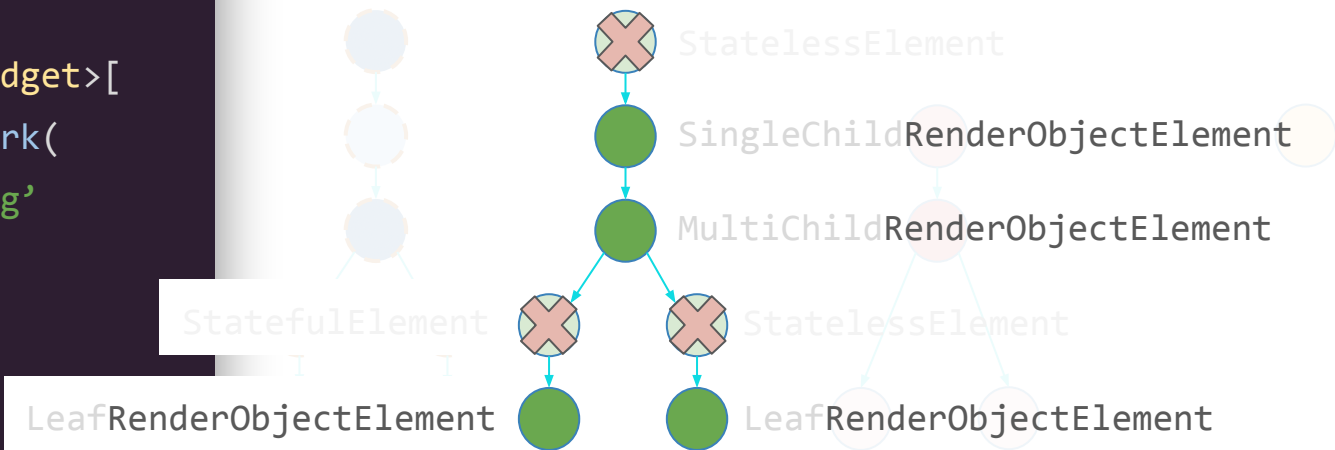
The diagram shows a hierarchy of widgets. The top level is a StatelessWidget (blue circle). It has two children: a RenderObjectWidget (green circle) and a RenderObjectWidget (red circle). The RenderObjectWidget (green circle) has two children: a StatelessWidget (blue circle) and a RenderObjectWidget (green circle). The StatelessWidget (blue circle) has two children: a StatelessWidget (blue circle) and a RenderObjectWidget (green circle). The RenderObjectWidget (green circle) has two children: a StatelessWidget (blue circle) and a RenderObjectWidget (green circle). The RenderObjectWidget (red circle) has two children: a StatelessWidget (blue circle) and a RenderObjectWidget (green circle). The RenderObjectWidget (yellow circle) has two children: a StatelessWidget (blue circle) and a RenderObjectWidget (green circle).

Flutter的树与渲染阶段

Flutter's Trees and Rendering Phases

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        './1.png'  
      ),  
      Text('A'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer

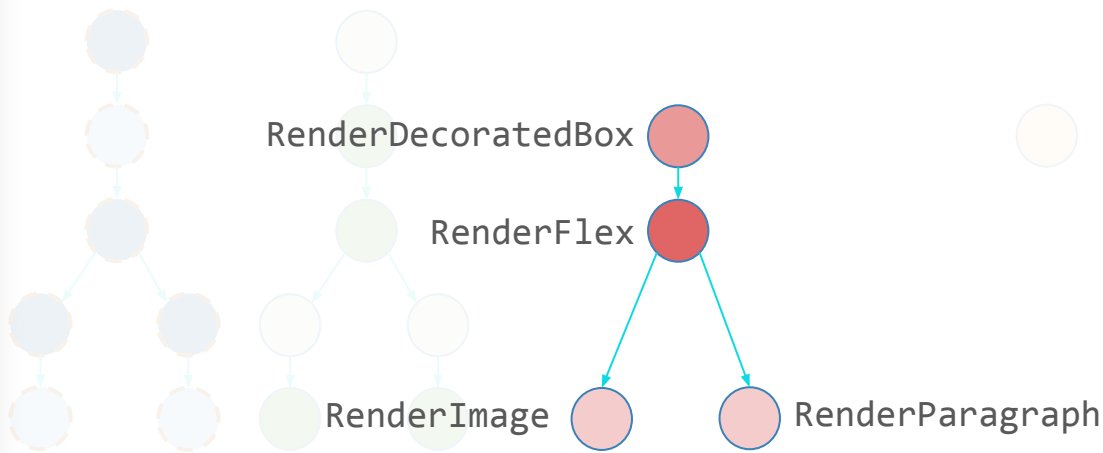


Flutter的树与渲染阶段

Flutter's Trees and Rendering Phases

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        '../1.png'  
      ),  
      Text('A'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer

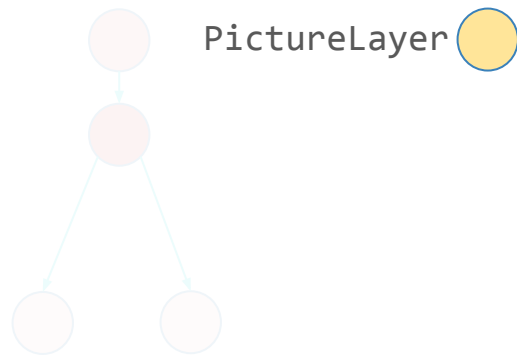
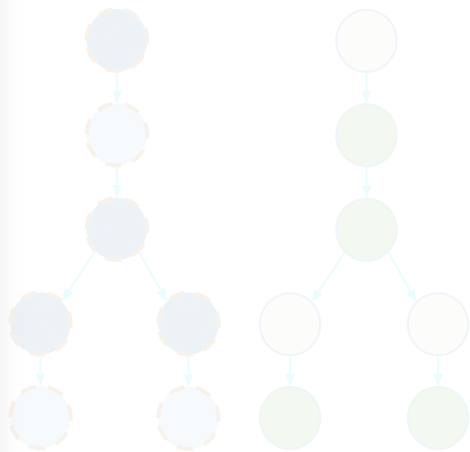


Flutter的树与渲染阶段

Flutter's Trees and Rendering Phases

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        '../1.png'  
      ),  
      Text('A'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer

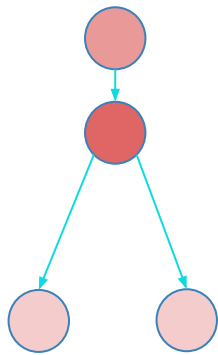
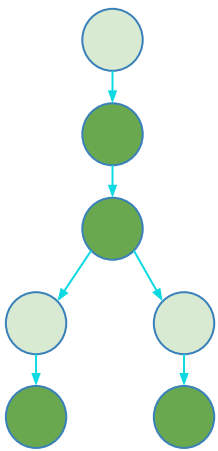
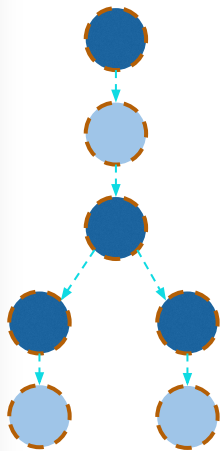


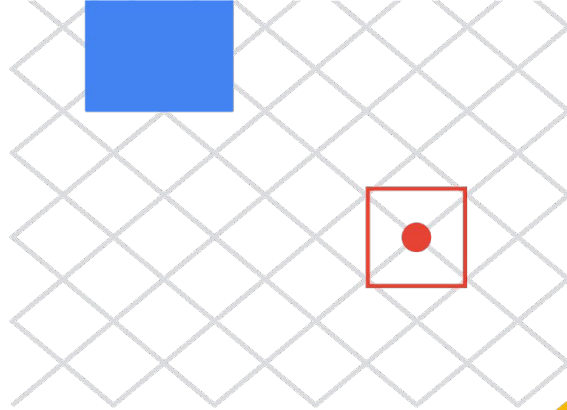
树工程

Tree Operations

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        './1.png'  
      ),  
      Text('A'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer





了解build阶段

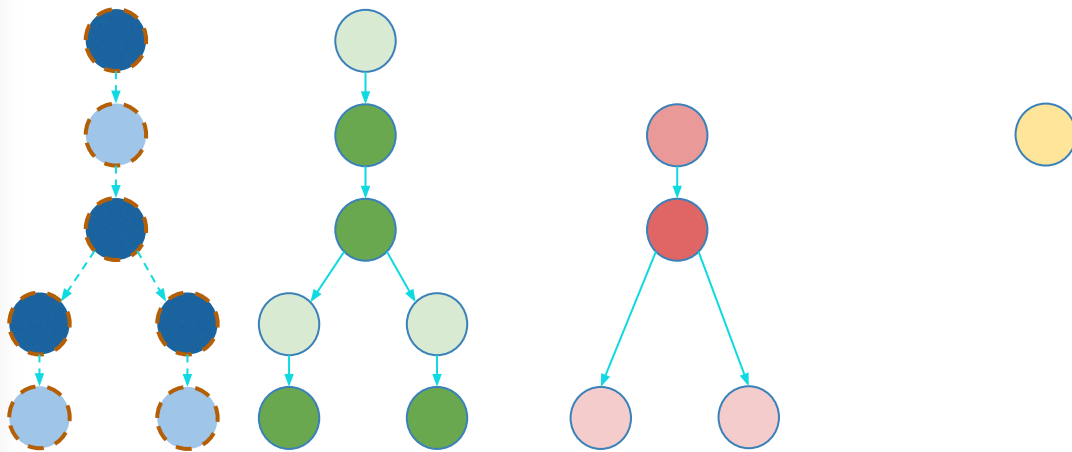
Understanding the Build Phase

同类型更新

Same Type Update

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        './1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```

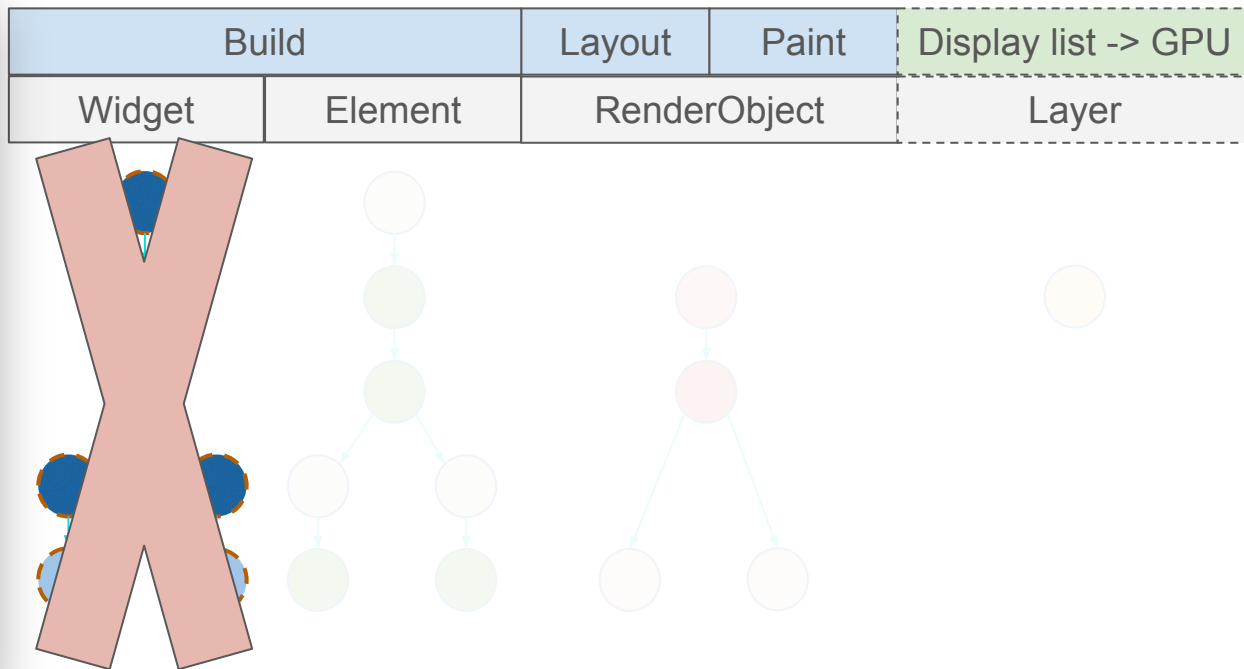
Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer



同类型更新

Same Type Update

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        '../1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```

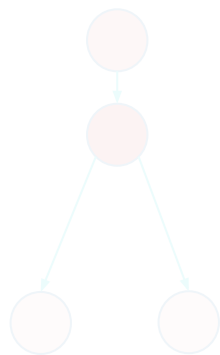
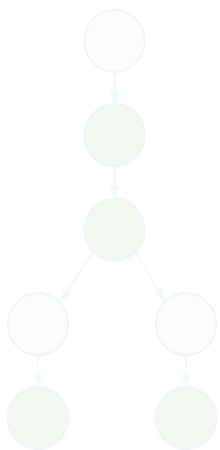
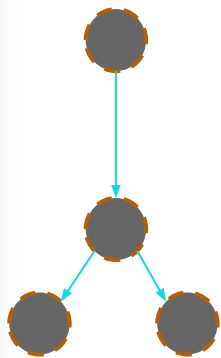


同类型更新

Same Type Update

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        '../1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer

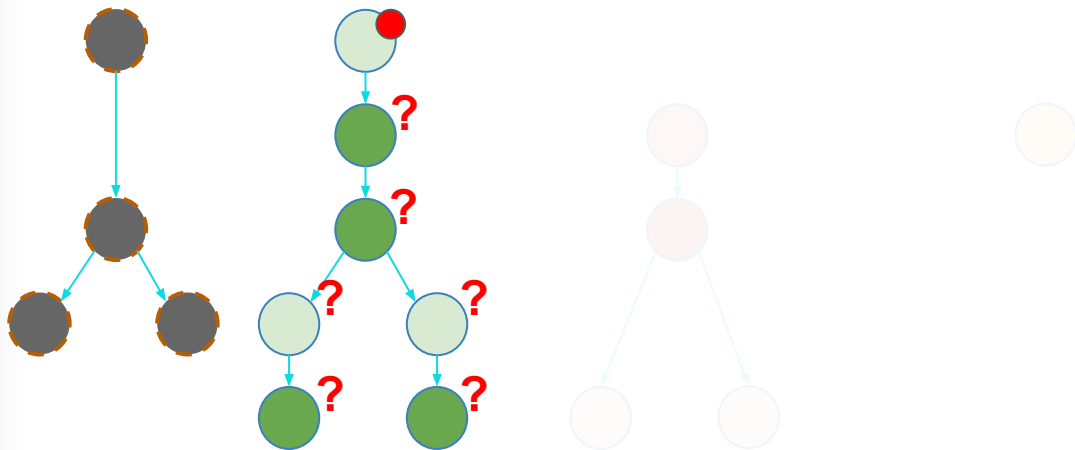


同类型更新

Same Type Update

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        './1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```

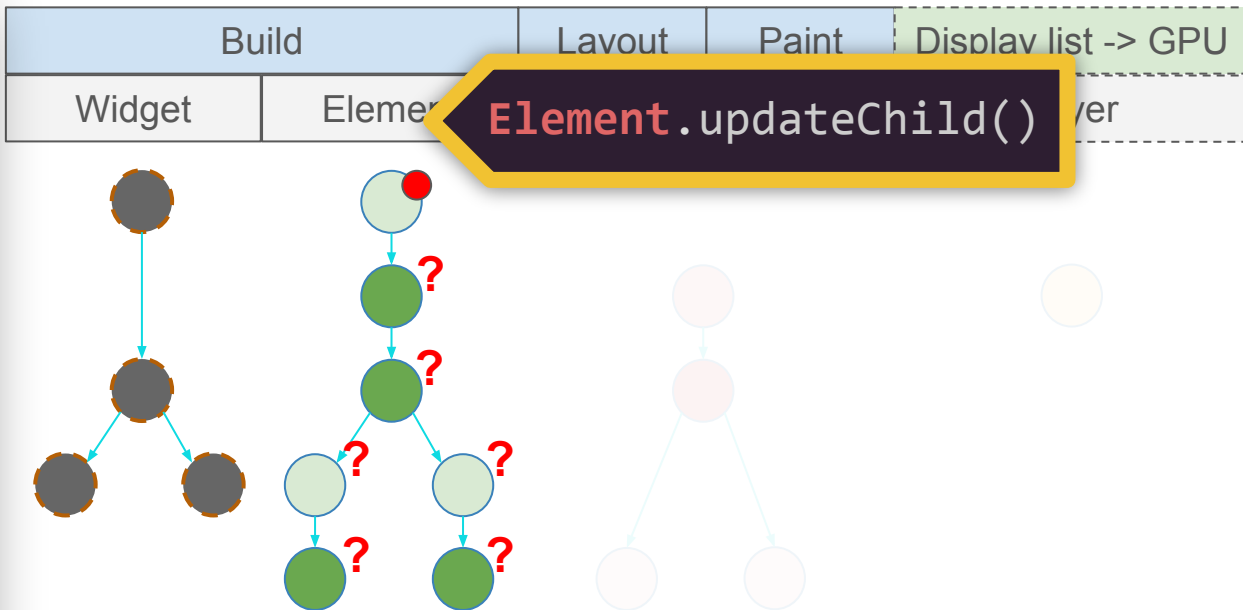
Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer



同类型更新

Same Type Update

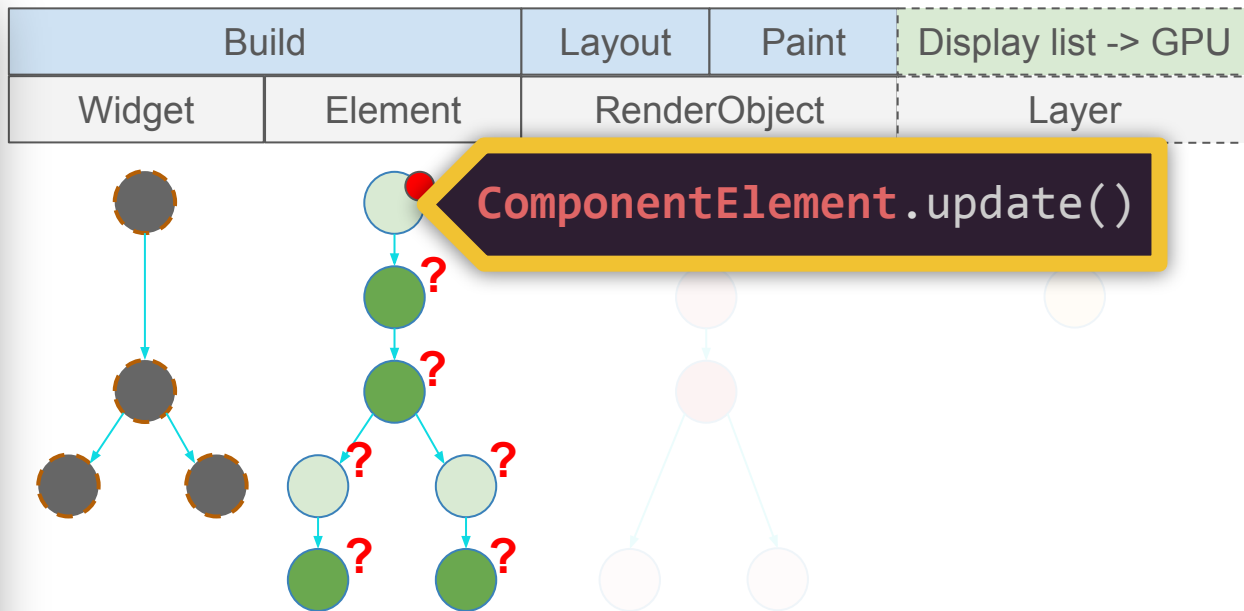
```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        './1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```



同类型更新

Same Type Update

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        './1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```

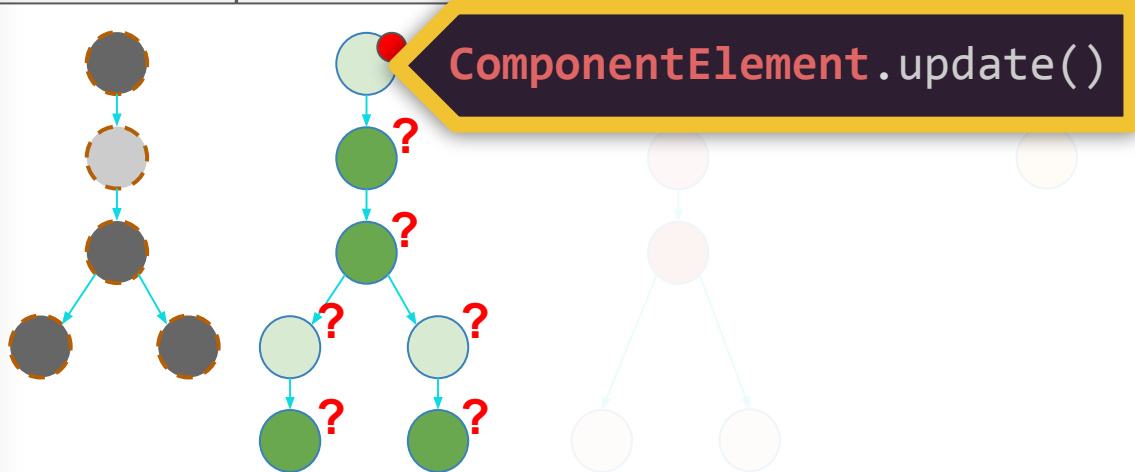


同类型更新

Same Type Update

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        './1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer

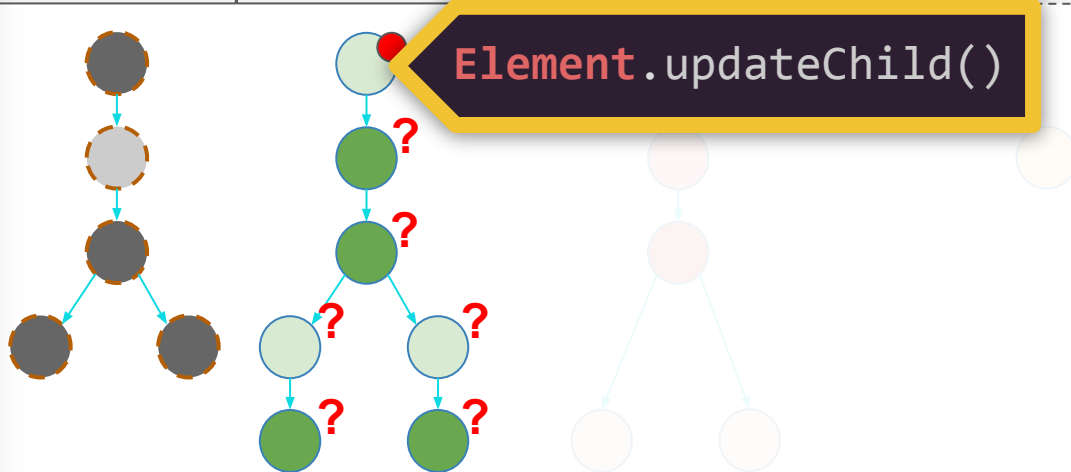


同类型更新

Same Type Update

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        '../1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer

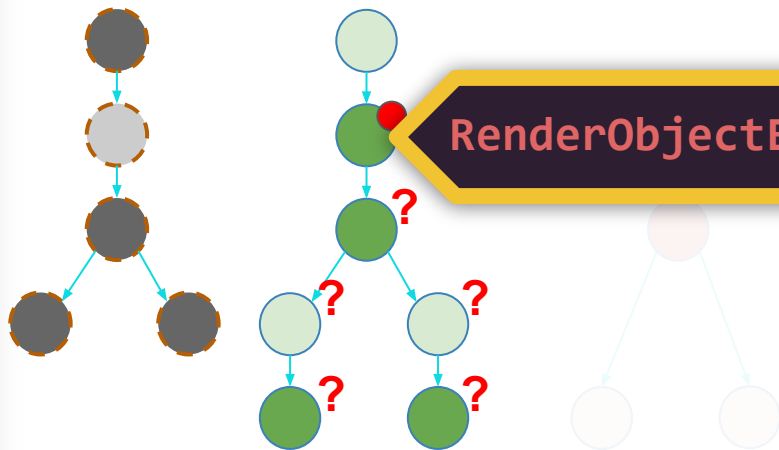


同类型更新

Same Type Update

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        '../1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```

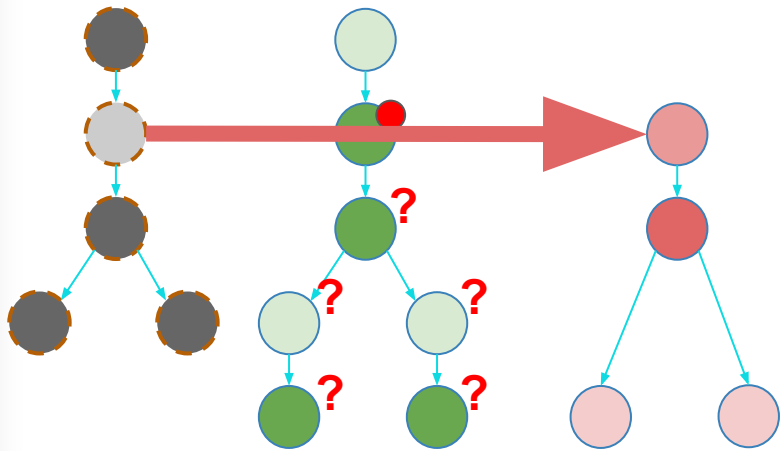
Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer



Same Type Update

Text('B'),

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer



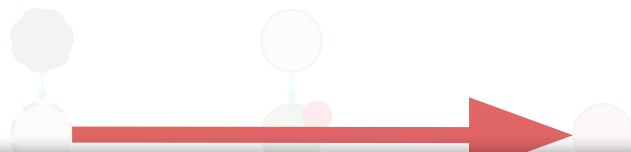
同类型更新

Same Type Update

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  

```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer



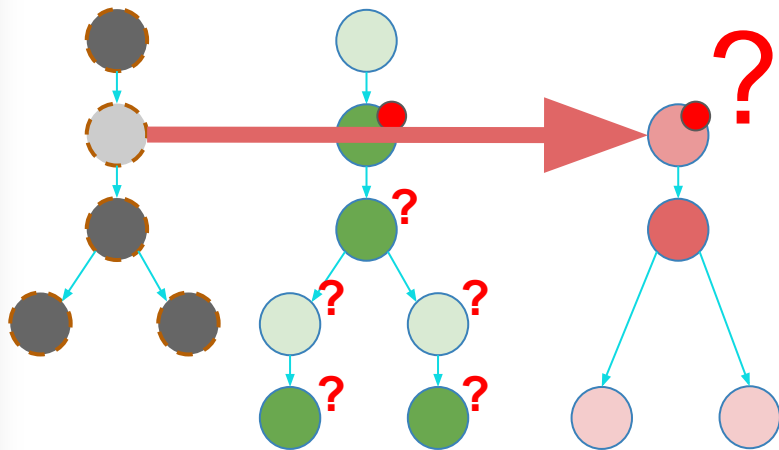
```
@override  
void updateRenderObject(BuildContext context, RenderDecoratedBox renderObject) {  
  renderObject  
    ..decoration = decoration  
    ..configuration = createLocalImageConfiguration(context)  
    ..position = position;  
}
```

同类型更新

Same Type Update

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        '../1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer

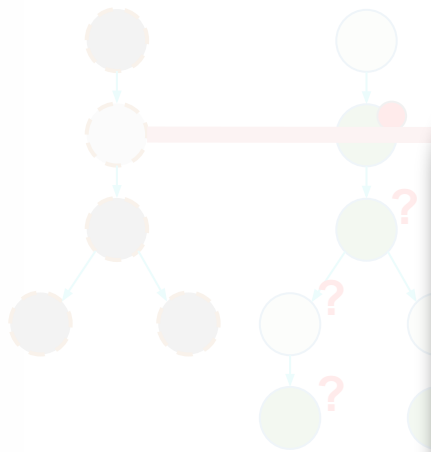


同类型更新

Same Type Update

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        '../1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer



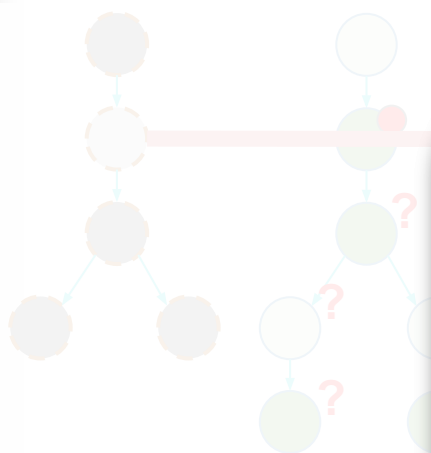
```
set decoration(Decoration value) {  
  assert(value != null);  
  if (value == _decoration)  
    return;  
  _painter?.dispose();  
  _painter = null;  
  _decoration = value;  
  markNeedsPaint();  
}
```

同类型更新

Same Type Update

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        '../1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer



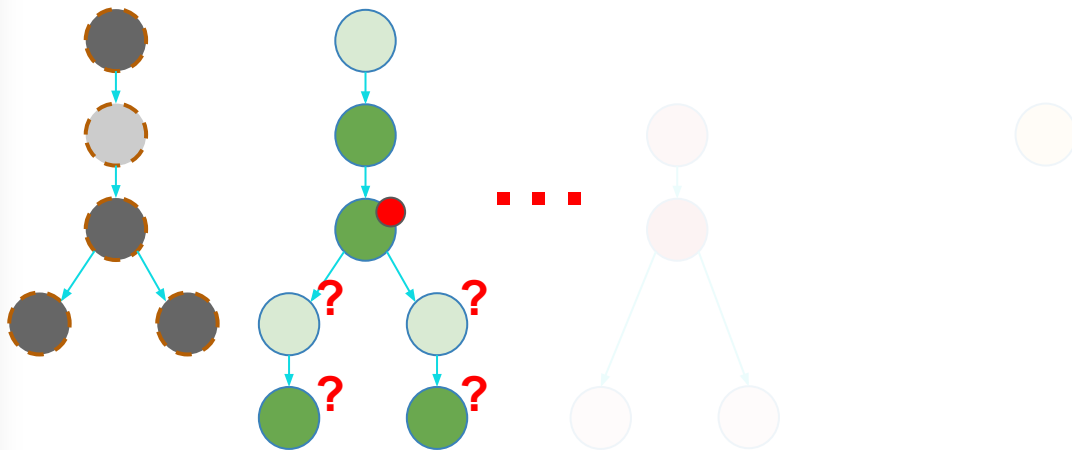
```
set decoration(Decoration value) {  
  assert(value != null);  
  if (value == _decoration)  
    return;  
  _painter?.dispose();  
  _painter = null;  
  _decoration = value;  
  markNeedsPaint();  
}
```

同类型更新

Same Type Update

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        '../1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer

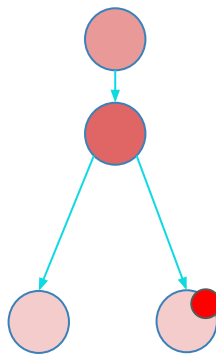
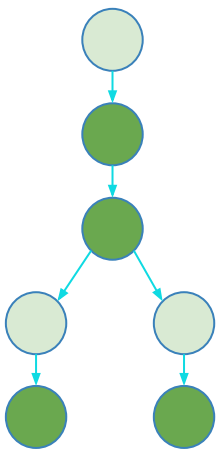
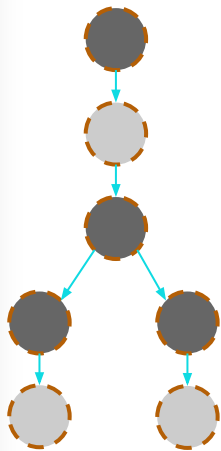


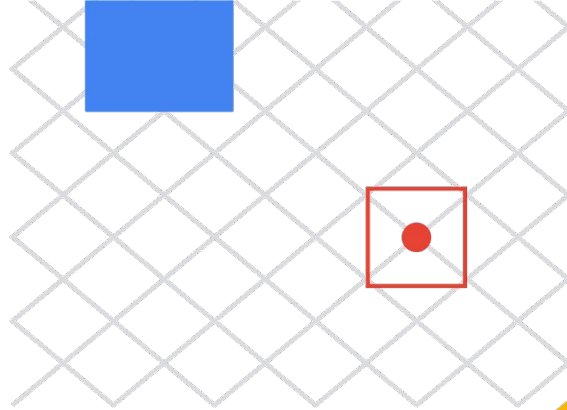
同类型更新

Same Type Update

```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        '../1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer





如何提高build效率

How to Improve Build Efficiency

调试工具

Performance Debugging Tools

- debugPrintBeginFrameBanner/debugPrintEndFrameBanner
 - 每帧开始/结束
- debugPrintScheduleBuildForStacks
 - 为什么被构建
- debugPrintRebuildDirtyWidgets
 - 什么组件被重新构建了
- debugProfileBuildsEnabled
 - 在观测台里显示构建树

调试工具

Performance Debugging Tools

- debugPrintBeginFrameBanner/debugPrintEndFrameBanner
 - 每帧开始/结束
- debugPrintScheduleBuildForStacks
 - 为什么被构建
- debugPrintRebuildDirtyWidgets
 - 什么组件被重构建了
- debugProfileBuildsEnabled
 - 在观测台里显示构建树

调试工具

Performance Debugging Tools

- debugPrintBeginFrameBanner/debugPrintEndFrameBanner
 - 每帧开始/结束
- debugPrintScheduleFrameForStacking
 - 为什么被构建
- debugPrintRebuildDirtyWidgets
 - 什么组件被重构建了
- debugProfileBuildsEnabled
 - 在观测台里显示构建树

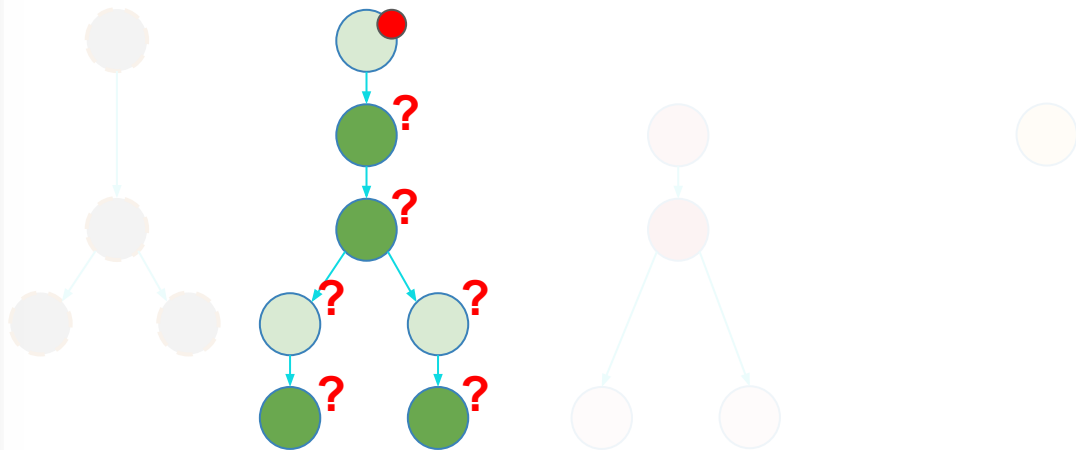
DEMO

如何提高build效率

How to Improve Build Efficiency

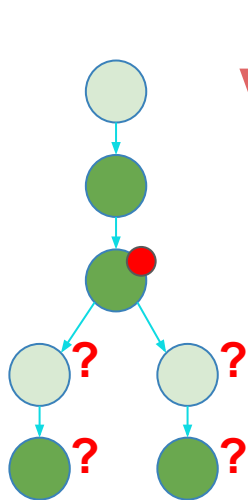
```
Container(  
  color: Colors.blue,  
  child: Row(  
    children: <Widget>[  
      Image.network(  
        './1.png'  
      ),  
      Text('B'),  
    ],  
  ),  
)
```

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer



如何提高build效率

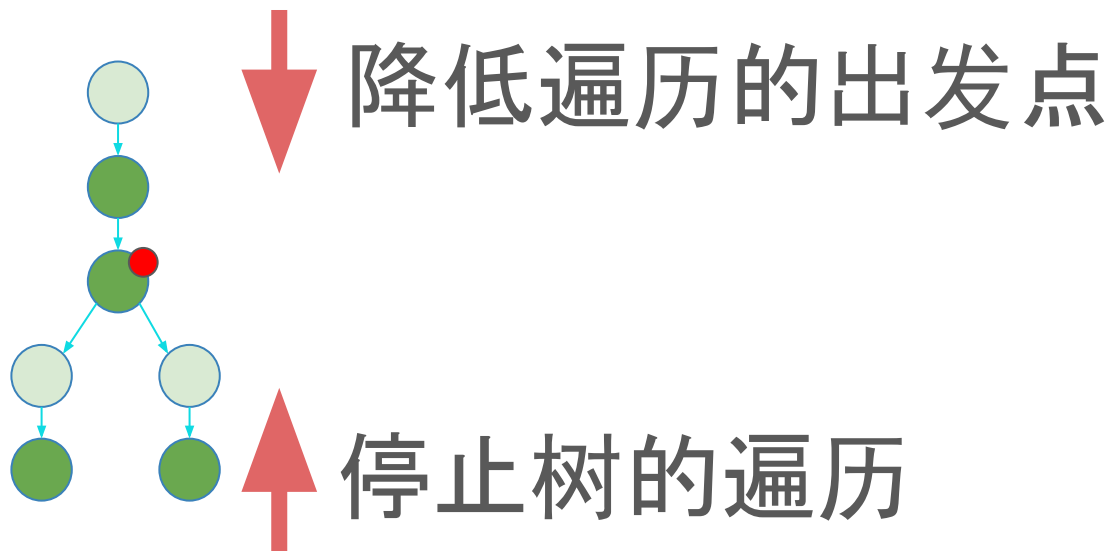
How to Improve Build Efficiency



降低遍历的出发点

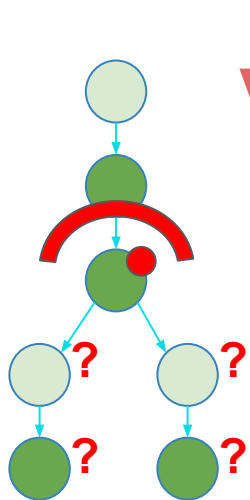
如何提高build效率

How to Improve Build Efficiency



如何提高build效率

How to Improve Build Efficiency



降低遍历的出发点

1. setState
2. InheritedWidget
3. 热重载

如何提高build效率

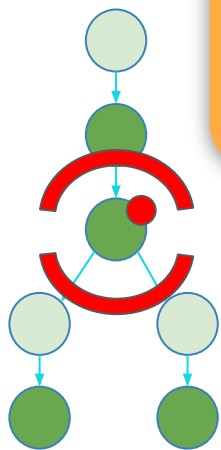
How to Improve Build Efficiency



如何提高build效率

How to Improve Build Efficiency

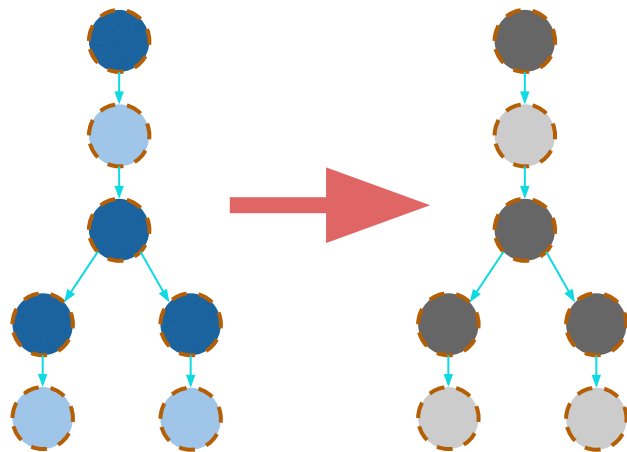
1. 重用同一子组件实例



停止树的遍历

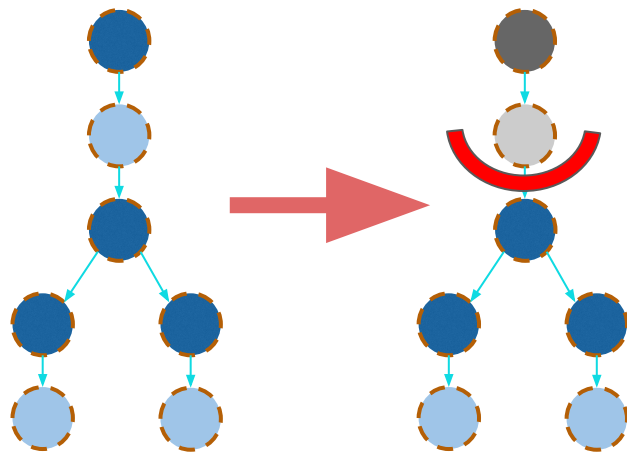
如何提高build效率

How to Improve Build Efficiency



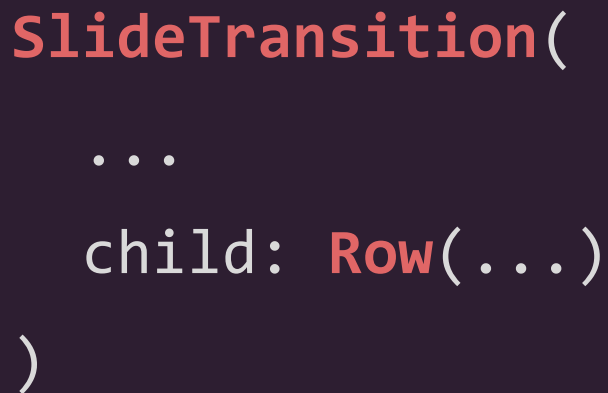
如何提高build效率

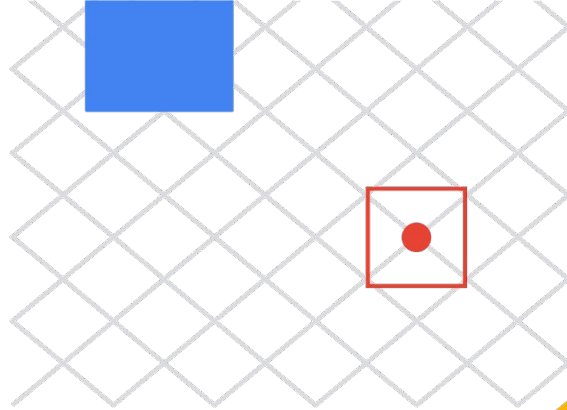
How to Improve Build Efficiency



How to Improve Build Efficiency

How to Improve Build Efficiency



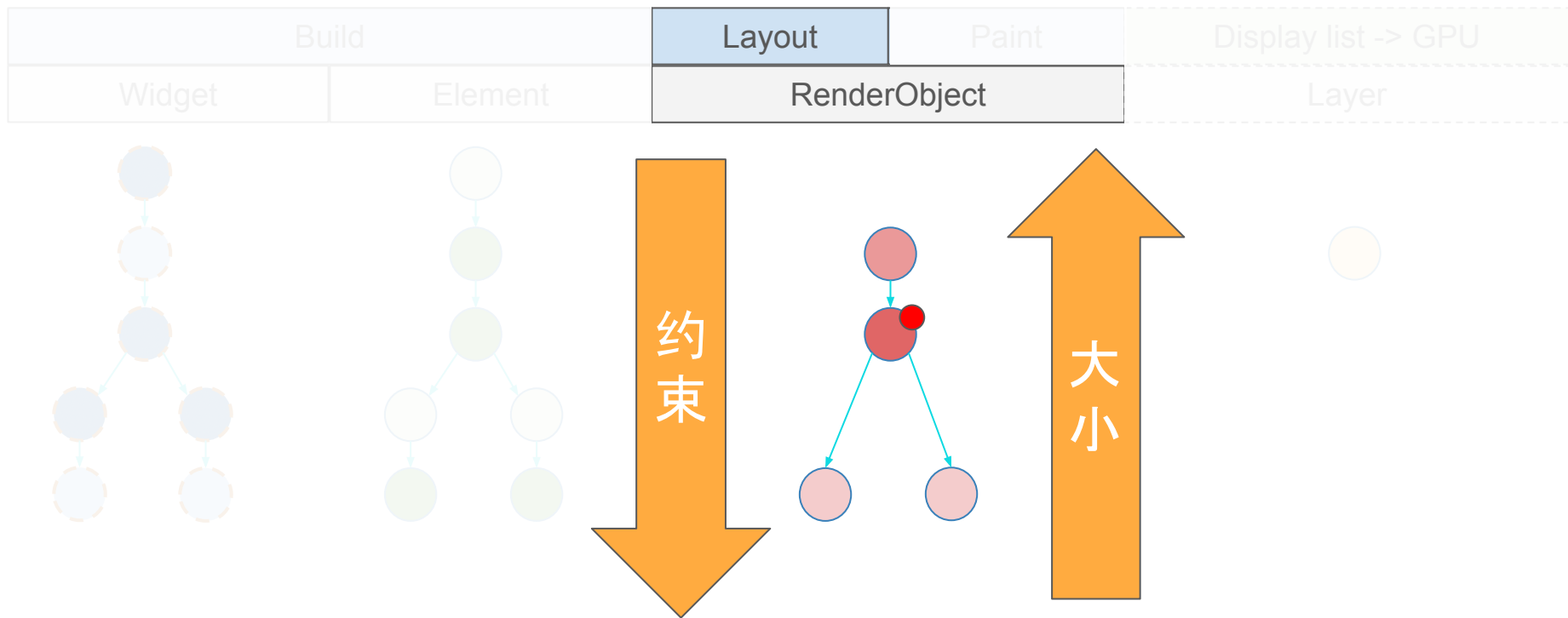


了解layout阶段

Understanding the Layout Phase

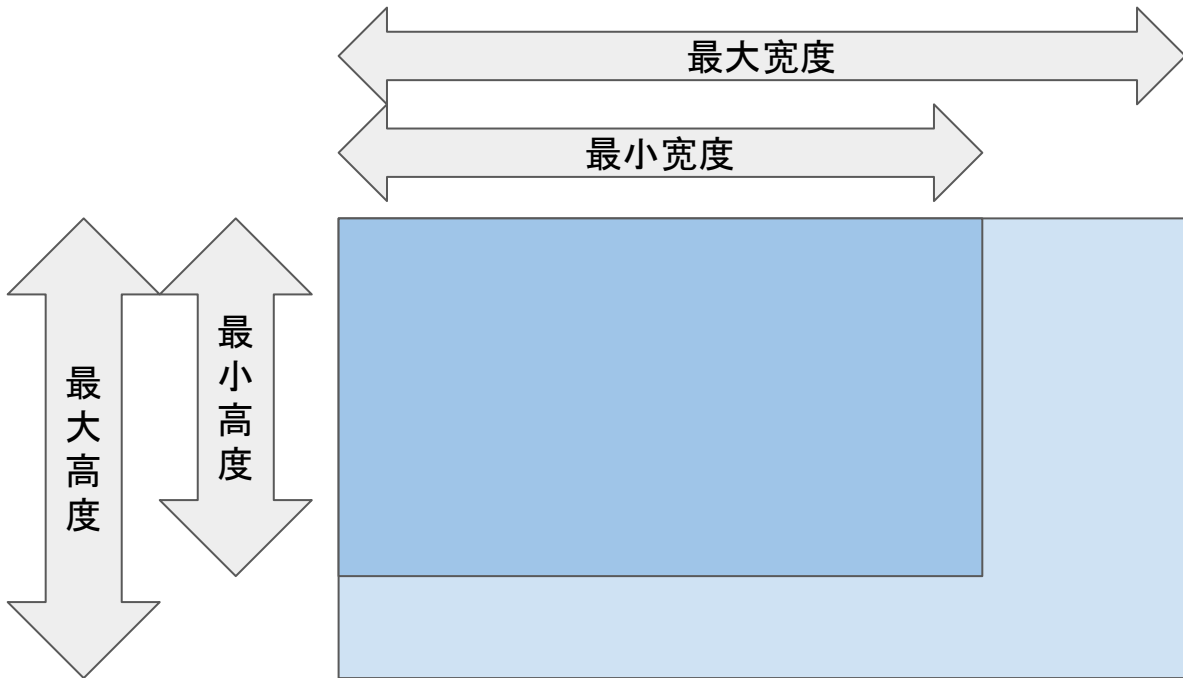
布局阶段

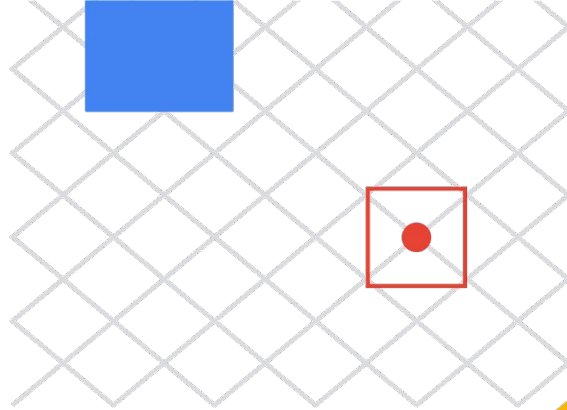
Layout Phase



布局阶段

Layout Phase



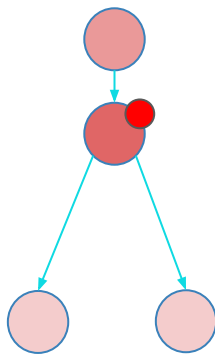
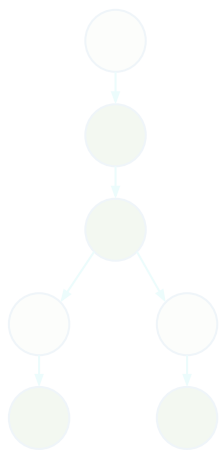
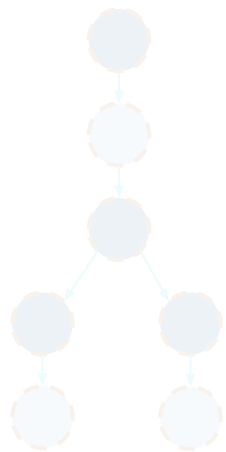
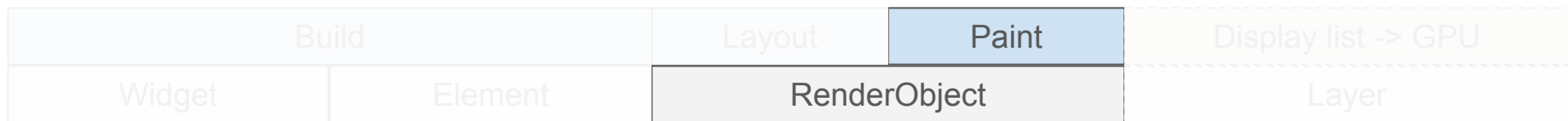


了解paint阶段

Understanding the Paint Phase

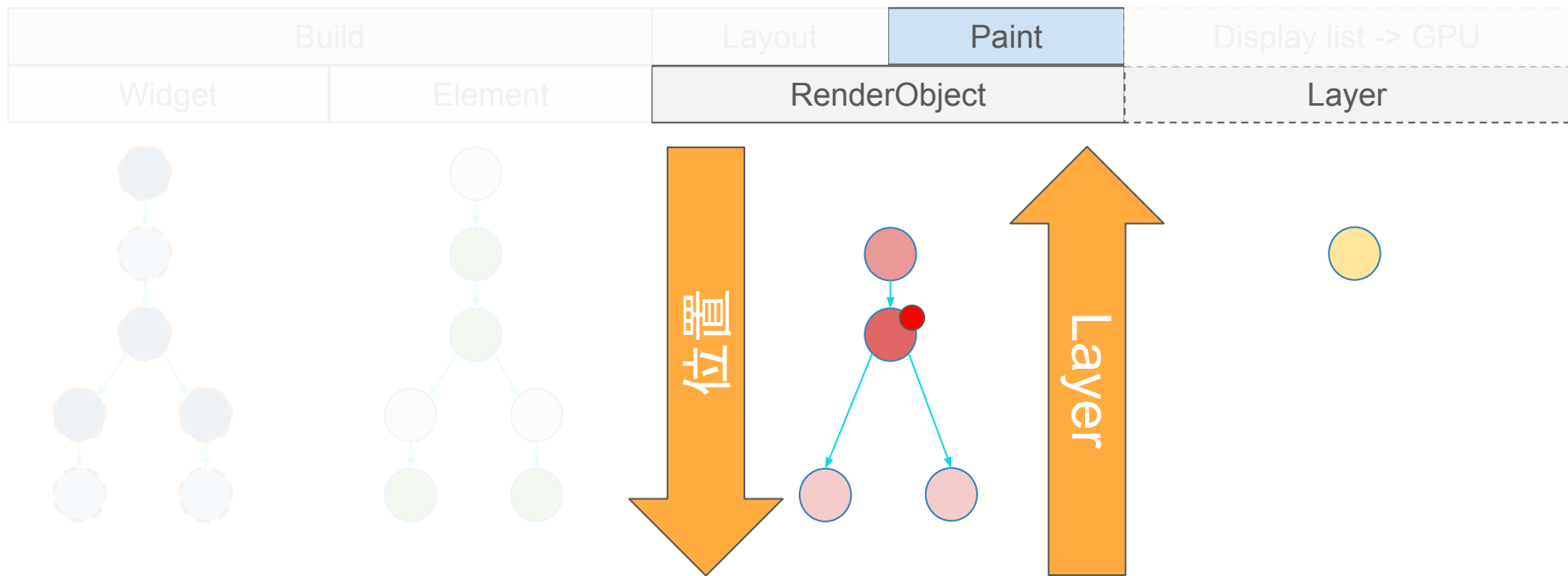
绘制阶段

Paint Phase



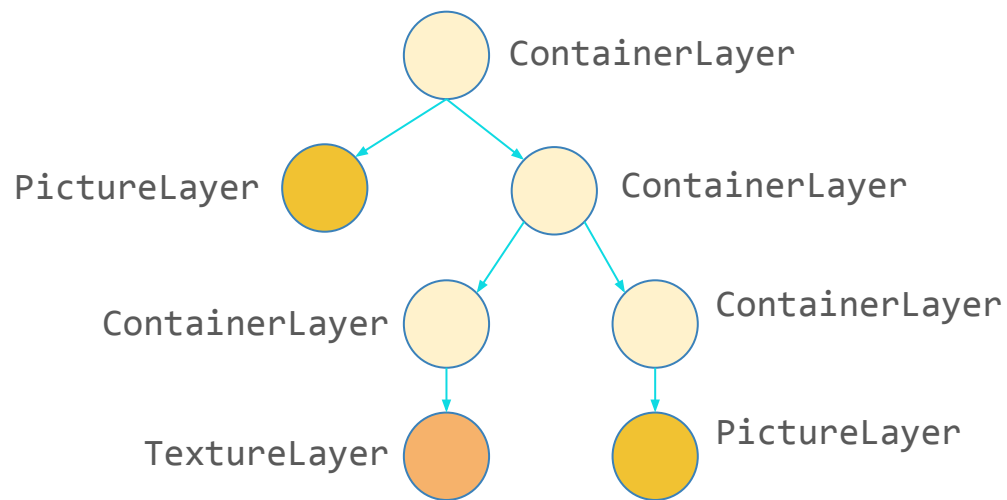
绘制阶段

Paint Phase



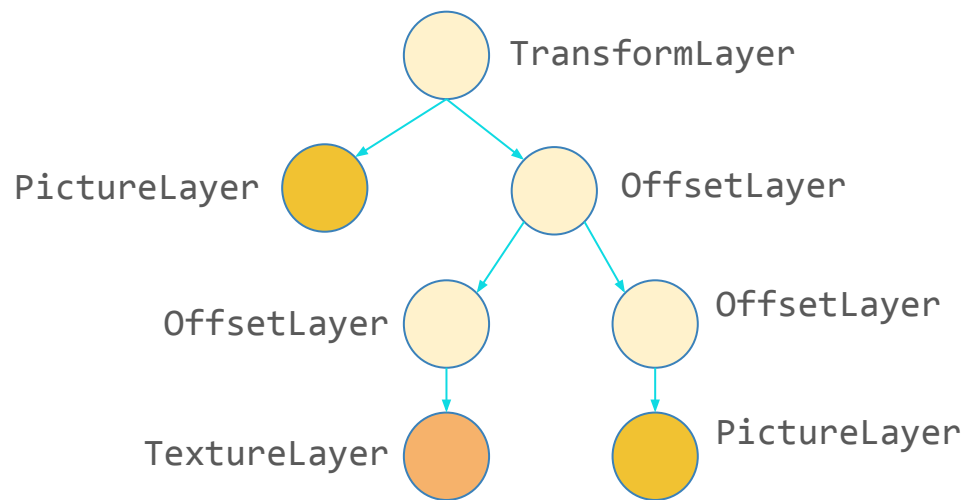
Layer种类

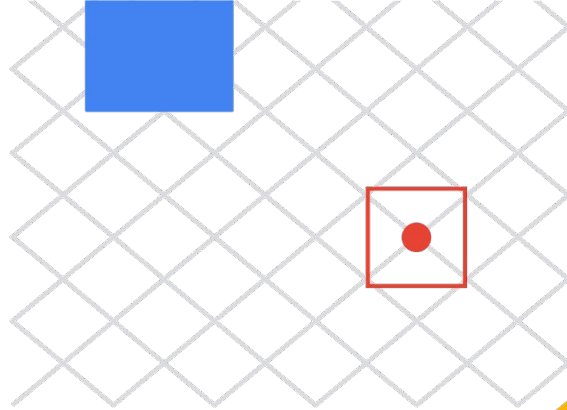
Layer Types



Layer种类

Layer Types



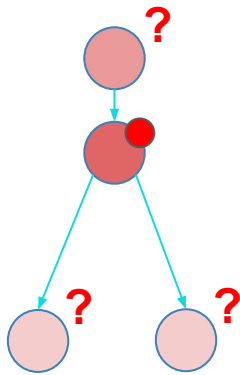
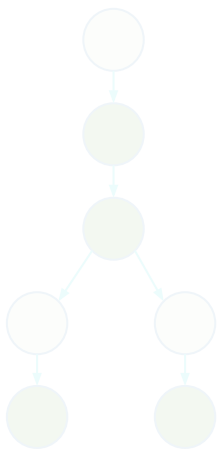
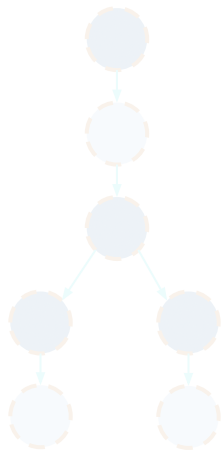
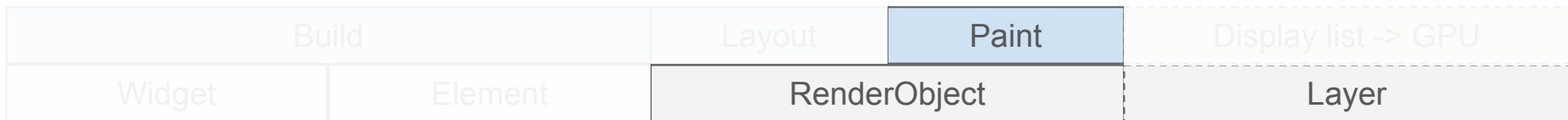


如何提高paint效率

How to Improve Paint Efficiency

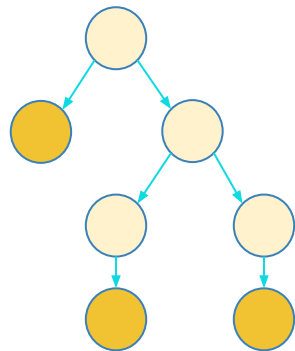
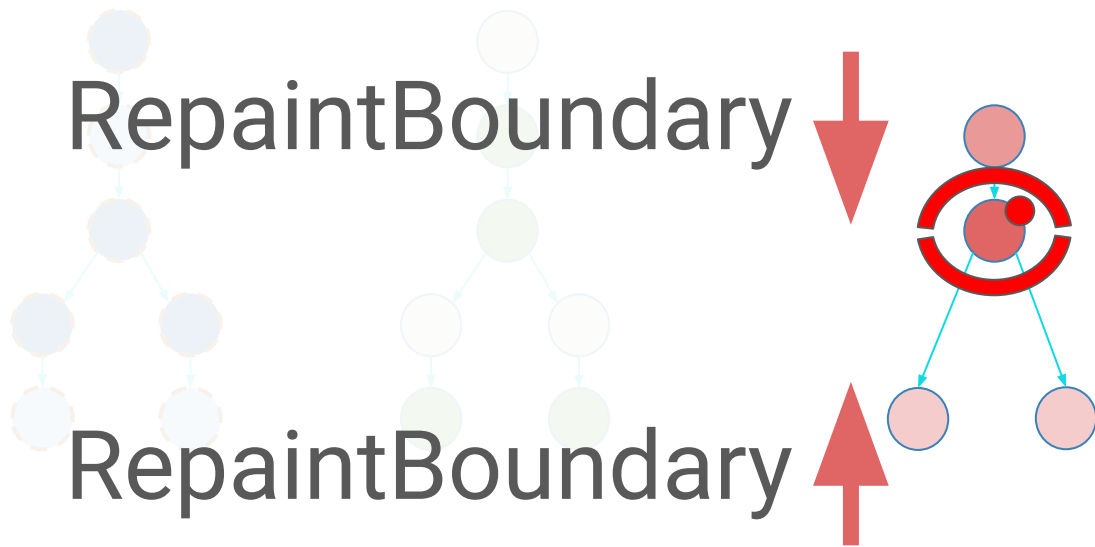
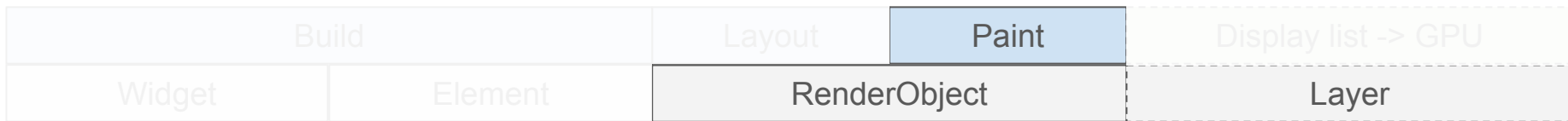
如何提高paint效率

How to Improve Paint Efficiency



如何提高paint效率

How to Improve Paint Efficiency



调试工具

Performance Debugging Tools

- debugDumpLayerTree
 - 查看layer树
- debugPaintLayerBordersEnabled
 - 查看layer界限
- debugRepaintRainbowEnabled
 - 被重新绘制的RenderObject
- debugProfilePaintsEnabled
 - 在观测台里显示绘制树

调试工具

Performance Debugging Tools

- debugDumpLayerTree
 - 查看layer树
- debugPaintLayerBordersEnabled
 - 查看layer界限
- debugRepaintRainbowEnabled
 - 被重新绘制的RenderObject
- debugProfilePaintsEnabled
 - 在了望台里显示绘制树

DEMO

xster@google.com



@xster

Build		Layout	Paint	Display list -> GPU
Widget	Element	RenderObject		Layer

