软件绿色联盟 | Alibaba Group 阿里巴巴集团 | HUAWEI

软件绿色联盟技术沙龙第28期·走进阿里
「淘系技术嘉年华 杭州站」

Flutter在闲鱼的技术演进和创新

2019 / 8 / 3 杭州 · 阿里巴巴

吉丰

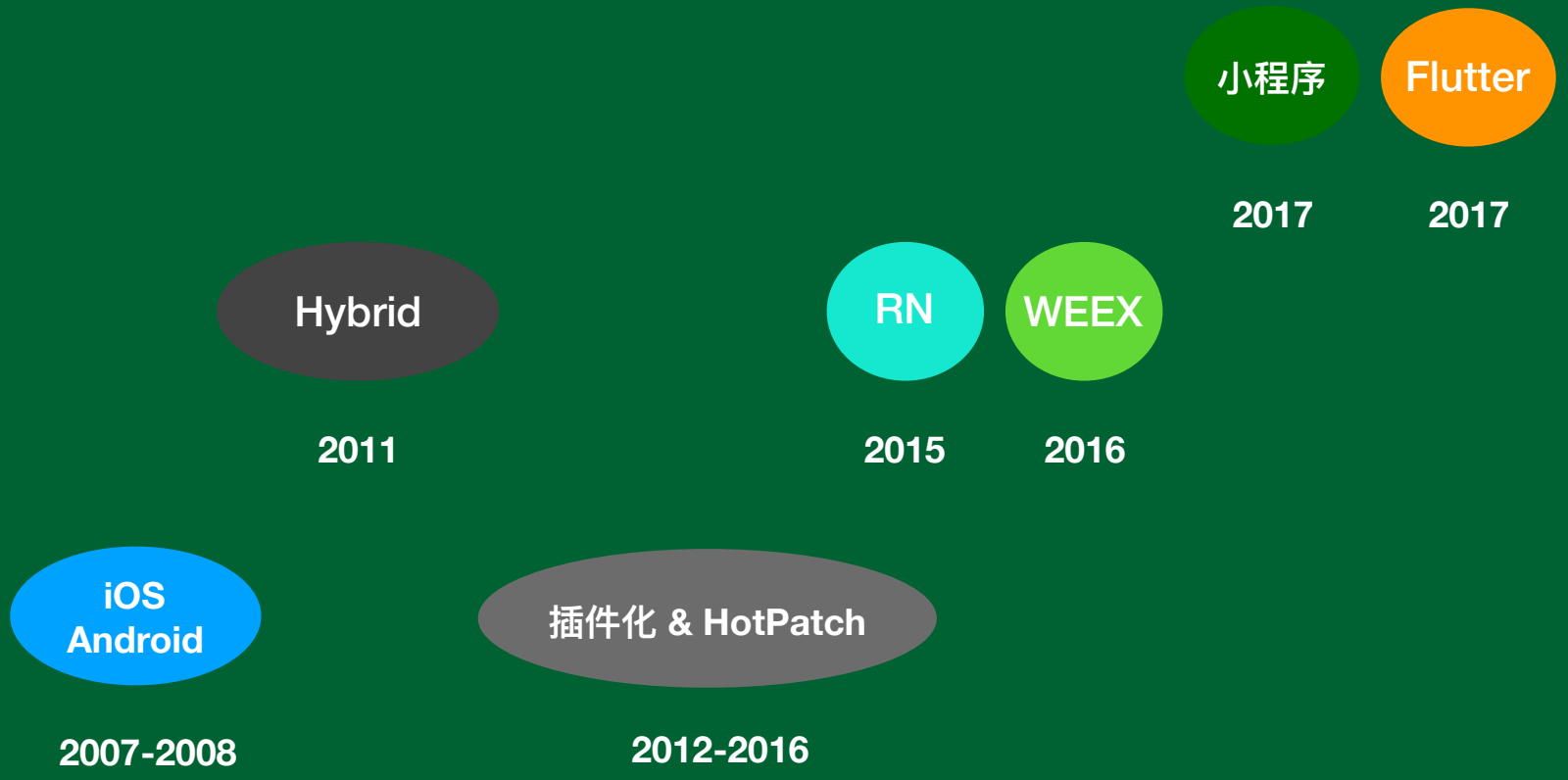
阿里巴巴淘系技术部无线技术专家

毕业于浙江大学，目前负责闲鱼客户端架构工作，主要研究方向包括跨端渲染、应用框架、混合技术等。

目录

目录

移动端动态化&跨端技术发展史



一云多端、高性能、高质量、轻量化

Flutter一瞥 – 跨端便携UI工具包

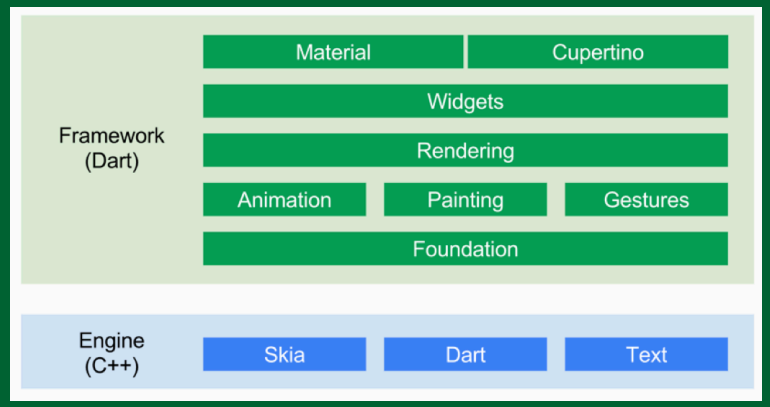
关键词

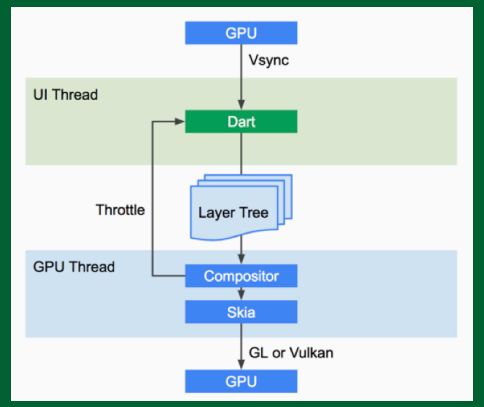
Fast Development

Expressive and Flexible UI

Native Performance

设计原则





重要节点

Alpha 2017.5

Beta 3 2018.5

Release Preview 2 2018.9

v1.0 2018.12

v1.2 2019.2

高性能跨终端引擎

Google下一代操作系统的内置 UI SDK

研发效率高/高性能/多端强一致性

目录

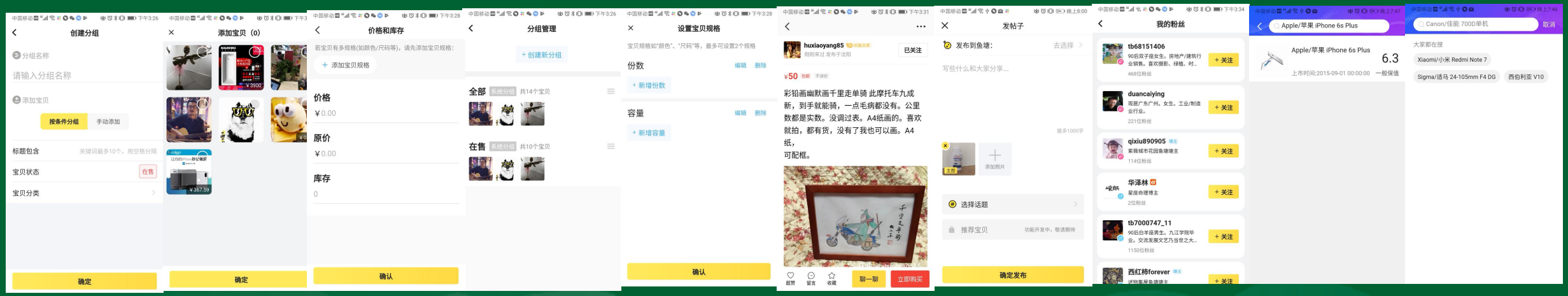
三亿人都在用的闲置交易社区

主链路

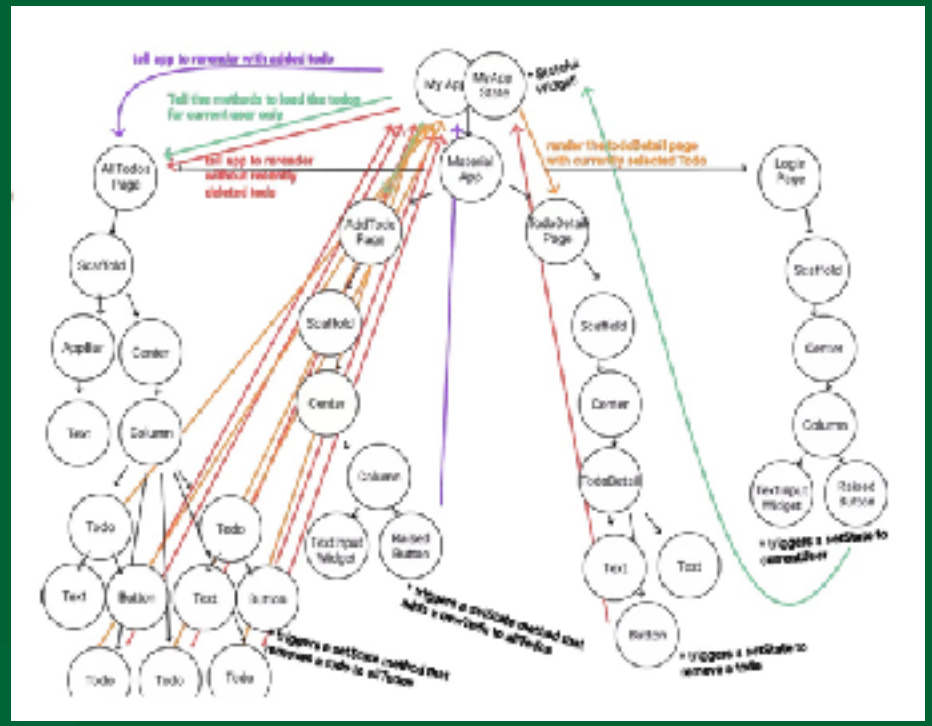
20+页面

覆盖音视频/富文本/
键盘等复杂场景

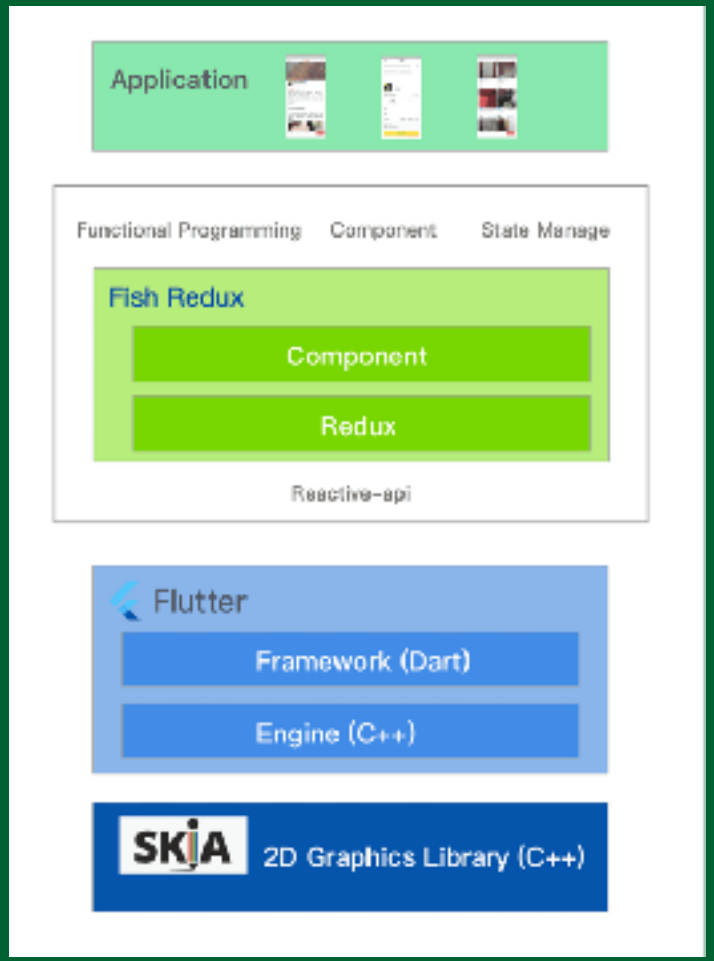
玩家 + 社区 + 闲鱼指数



规模化应用 - fish-redux 框架



耦合度高　　复用度低



fish-redux

An assembled flutter application framework.

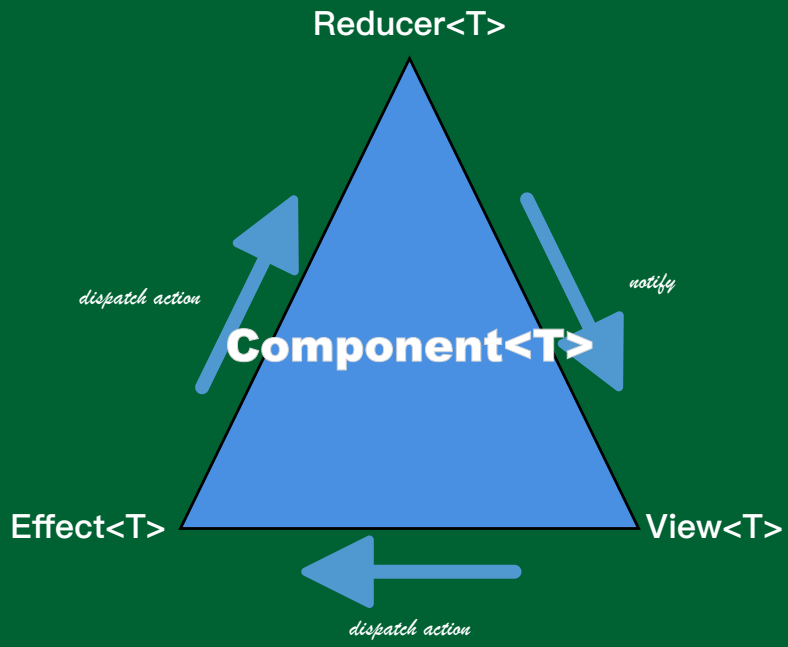
Functional Program

UI = f(state)

State = f(state, action)

SideEffect = f(action)

f = [middleware,...].reduce(f)

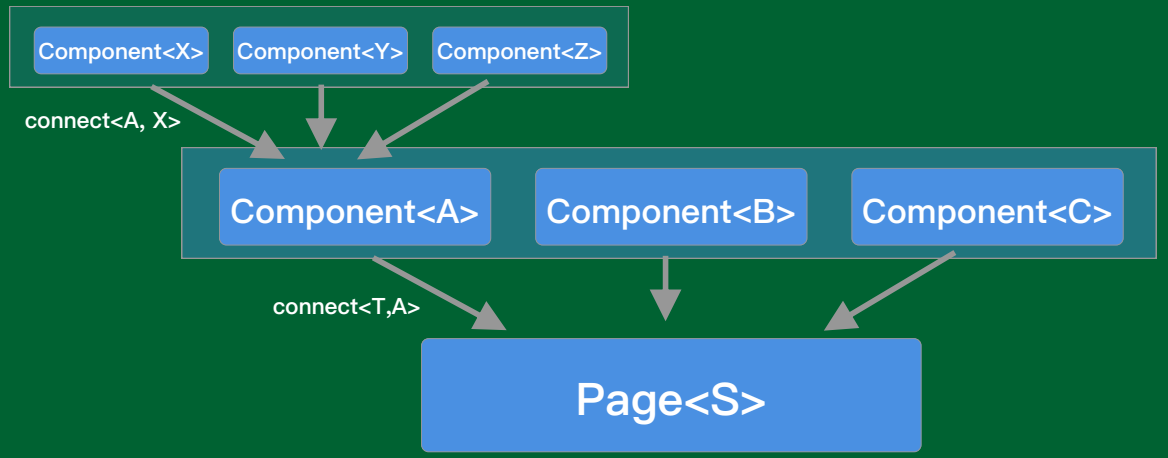


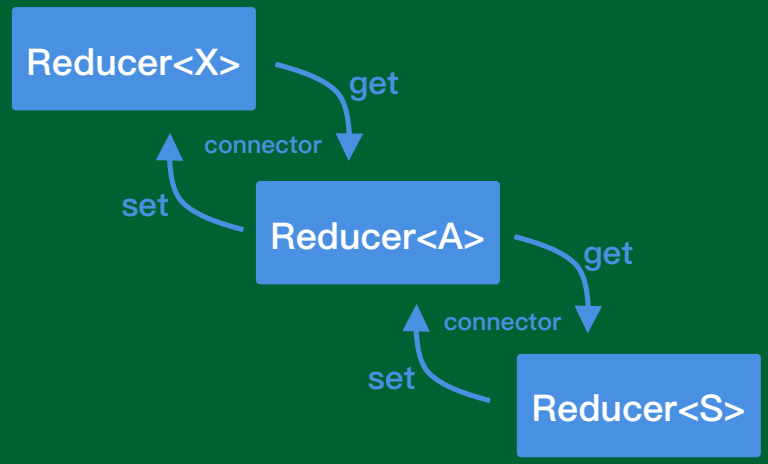
Functional Program

✓ 易编写　✓ 易测试　✓ 易维护

低耦合　易组合　易切面

Redux状态管理

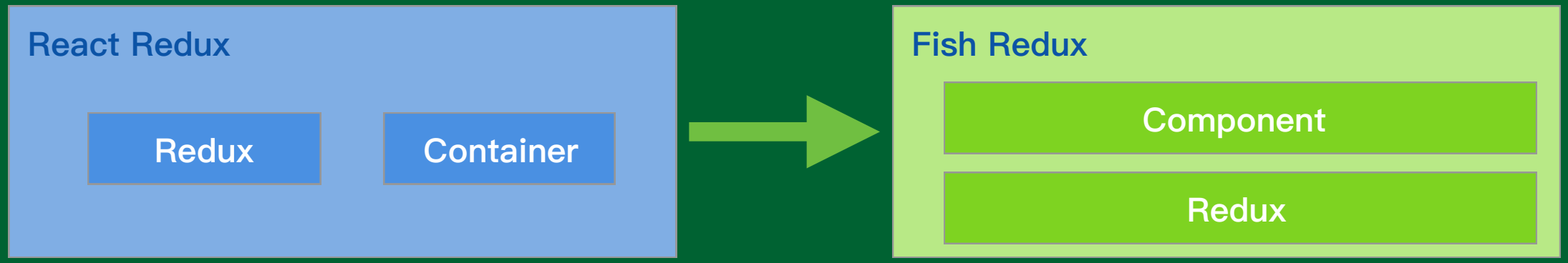




UI树和状态管理树的重合

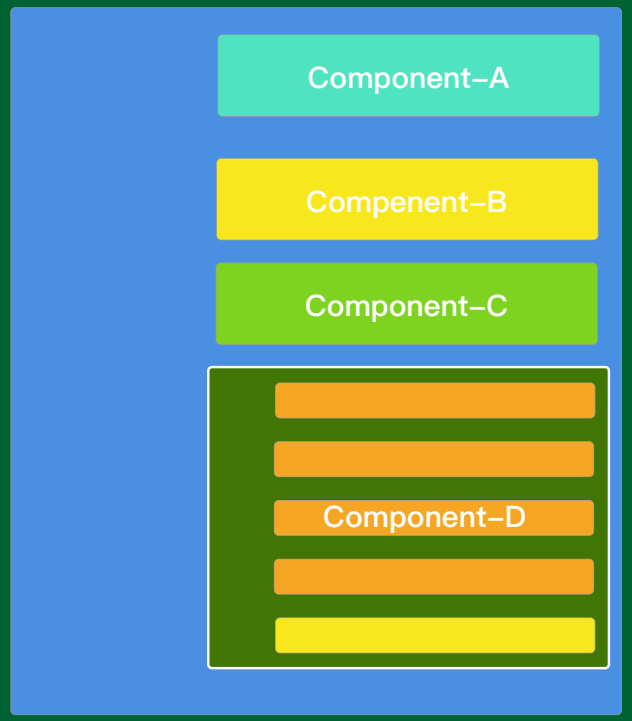
结构复用

Redux状态管理



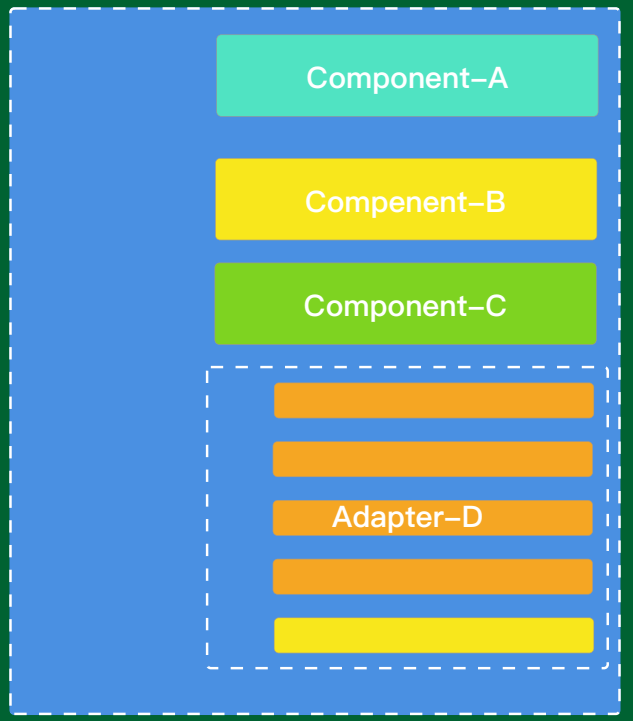
✓ 大大降低Redux的使用复杂度

列表下的组件化抽象



ListView

Component是一种的具体组件化



ListView-Adapter

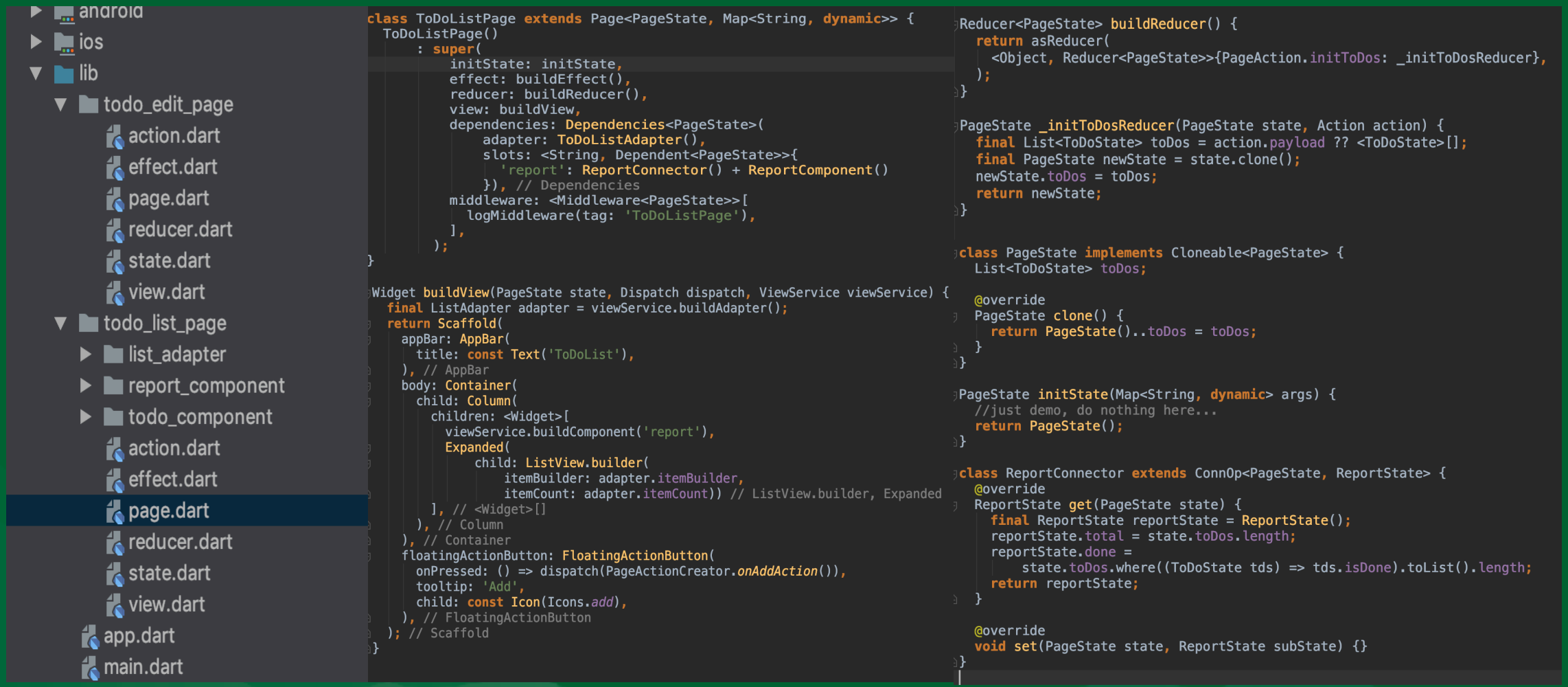
Adapter是一种虚拟的组件化

Like flatMap api

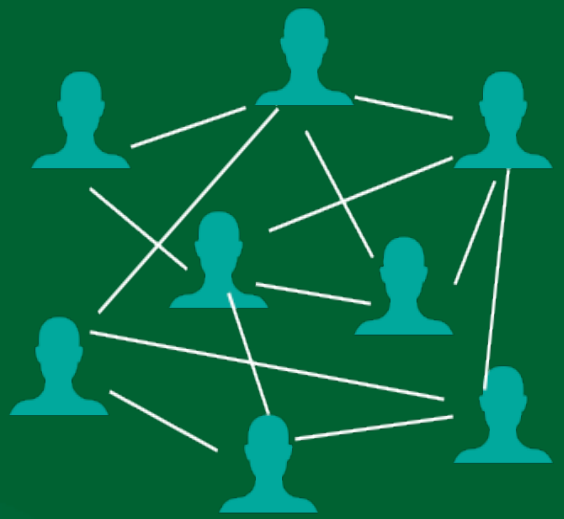
列表下的组件化抽象

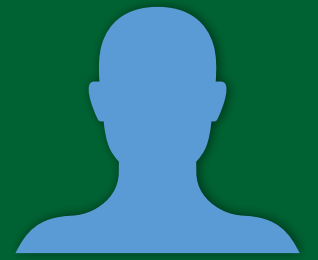
- ✓ No Big-Cell
- ✓ 更独立
- ✓ 更多的生命周期

ShowCase


android
ios
lib
todo_edit_page
action.dart
effect.dart
page.dart
reducer.dart
state.dart
view.dart
todo_list_page
list_adapter
report_component
todo_component
action.dart
effect.dart
page.dart
reducer.dart
state.dart
view.dart
app.dart
main.dart
class ToDoListPage extends Page<PageState, Map<String, dynamic>> {
ToDoListPage()
: super(
initState: initState,
effect: buildEffect(),
reducer: buildReducer(),
view: buildView,
dependencies: Dependencies<PageState>(
adapter: ToDoListAdapter(),
slots: <String, Dependent<PageState>>{
'report': ReportConnector() + ReportComponent()
}), // Dependencies
middleware: <Middleware<PageState>>[
logMiddleware(tag: 'ToDoListPage'),
],
);
}
Widget buildView(PageState state, Dispatch dispatch, ViewService viewService) {
final ListAdapter adapter = viewService.buildAdapter();
return Scaffold(
appBar: AppBar(
title: const Text('ToDoList'),
), // AppBar
body: Container(
child: Column(
children: <Widget>[
viewService.buildComponent('report'),
Expanded(
child: ListView.builder(
itemBuilder: adapter.itemBuilder,
itemCount: adapter.itemCount)) // ListView.builder, Expanded
], // <Widget>[]
), // Column
), // Container
floatingActionButton: FloatingActionButton(
onPressed: () => dispatch(PageActionCreator.onAddAction()),
tooltip: 'Add',
child: const Icon(Icons.add),
), // FloatingActionButton
); // Scaffold
}
Reducer<PageState> buildReducer() {
return asReducer(
<Object, Reducer<PageState>>{PageAction.initToDos: _initToDosReducer},
);
}
PageState _initToDosReducer(PageState state, Action action) {
final List<ToDoState> toDos = action.payload ?? <ToDoState>[];
final PageState newState = state.clone();
newState.toDos = toDos;
return newState;
}
class PageState implements Cloneable<PageState> {
List<ToDoState> toDos;
@override
PageState clone() {
return PageState()..toDos = toDos;
}
}
PageState initState(Map<String, dynamic> args) {
//just demo, do nothing here...
return PageState();
}
class ReportConnector extends ConnOp<PageState, ReportState> {
@override
ReportState get(PageState state) {
final ReportState reportState = ReportState();
reportState.total = state.toDos.length;
reportState.done =
state.toDos.where((ToDoState tds) => tds.isDone).toList().length;
return reportState;
}
@override
void set(PageState state, ReportState subState) {}
}

Dart一体化





Flutter&Dart技术栈统一

消除云端技术壁垒

客户端同学写服务端

- Serverless化
  （FaaS，Service Mesh）
- 公用组件
  （消息、缓存、KV、关系DB）

服务端同学写客户端

- UI2Code
  (根据图片生成UI代码)
- 页面代码模板化
  (页面容器，数据管理)

Dart一体化架构效果

资源均衡

Android、iOS配比不再关心
业务研发团队垂直化

协同高效

协同边数大幅减少

关注点聚焦

业务开发100%关注PRD、业务领域层开发20%关注PRD

多端一致

Android、iOS单端Bug减少
Android、iOSUI、逻辑一致性提升

不再有协议约定

无协议约定
DO、协议接口一份

目录

回顾

Flutter的优势与挑战

闲鱼的Flutter技术体系

思考

https://github.com/alibaba/fish-redux

http://github.com/alibaba/flutter_boost

https://github.com/alibaba-flutter