



R lab 移动端团队技术架构体系及演进

R-Lab 2018年02月28日

Peter (子成)

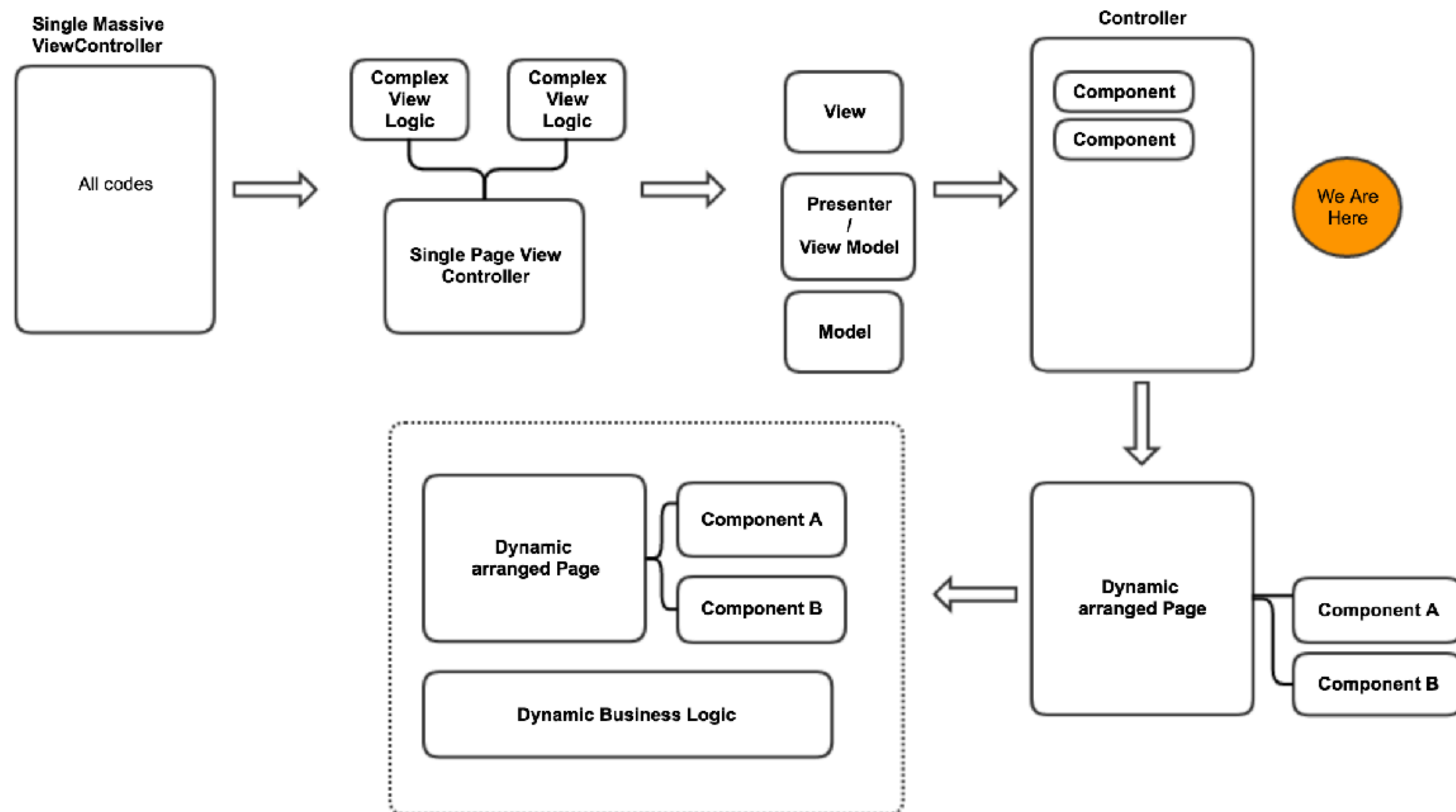


目录

- 1 | History
技术演进的历史
- 2 | Nova
自己造个轮子
- 3 | Skeleton
规范代码
- 4 | Thinking
我们对移动端架构的思考
- 5 | Goto
下一步做什么
- 6 | Others
怎么做好业务开发



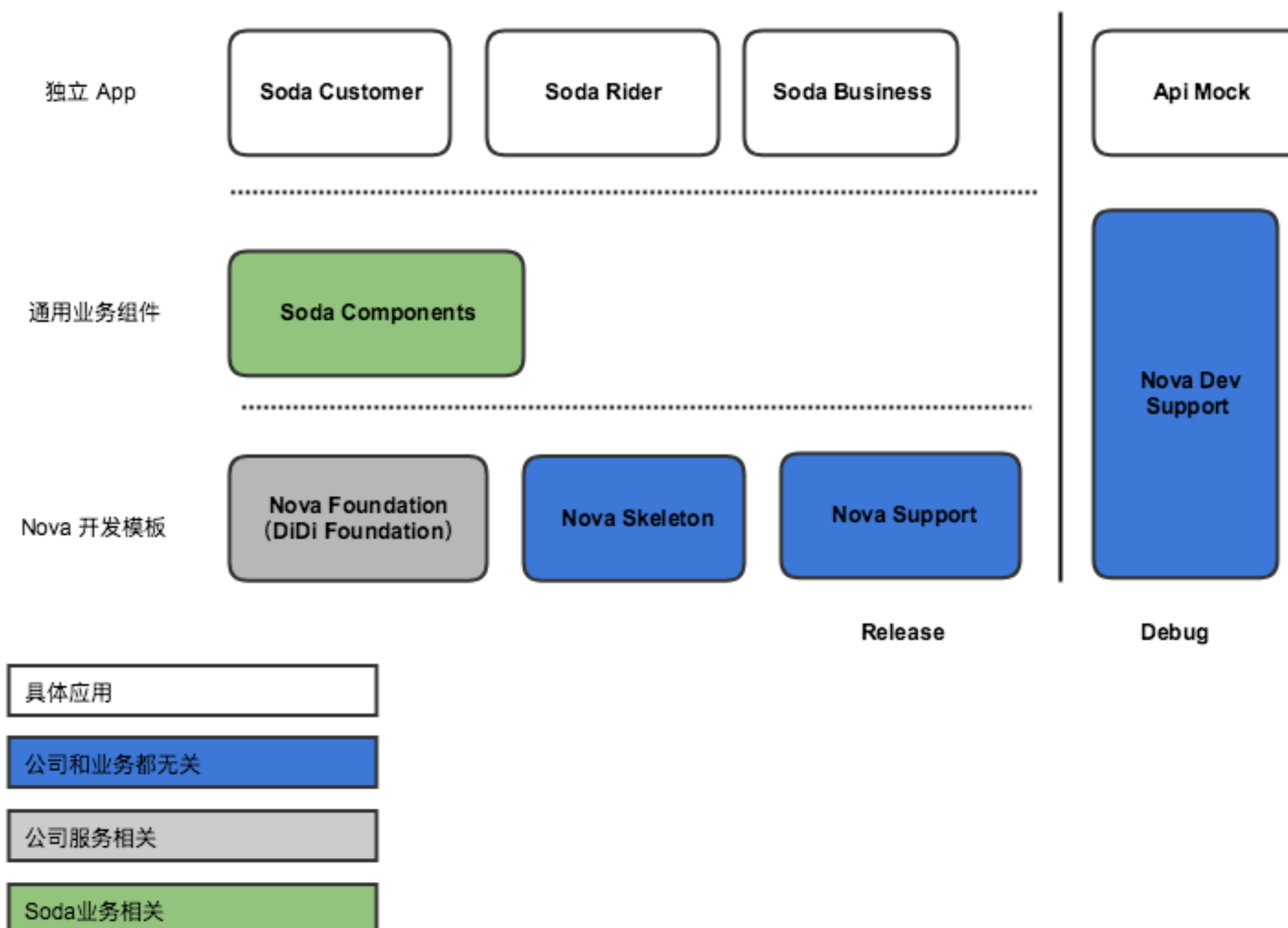
CHAPTER – 01 History





CHAPTER – 02

Nova





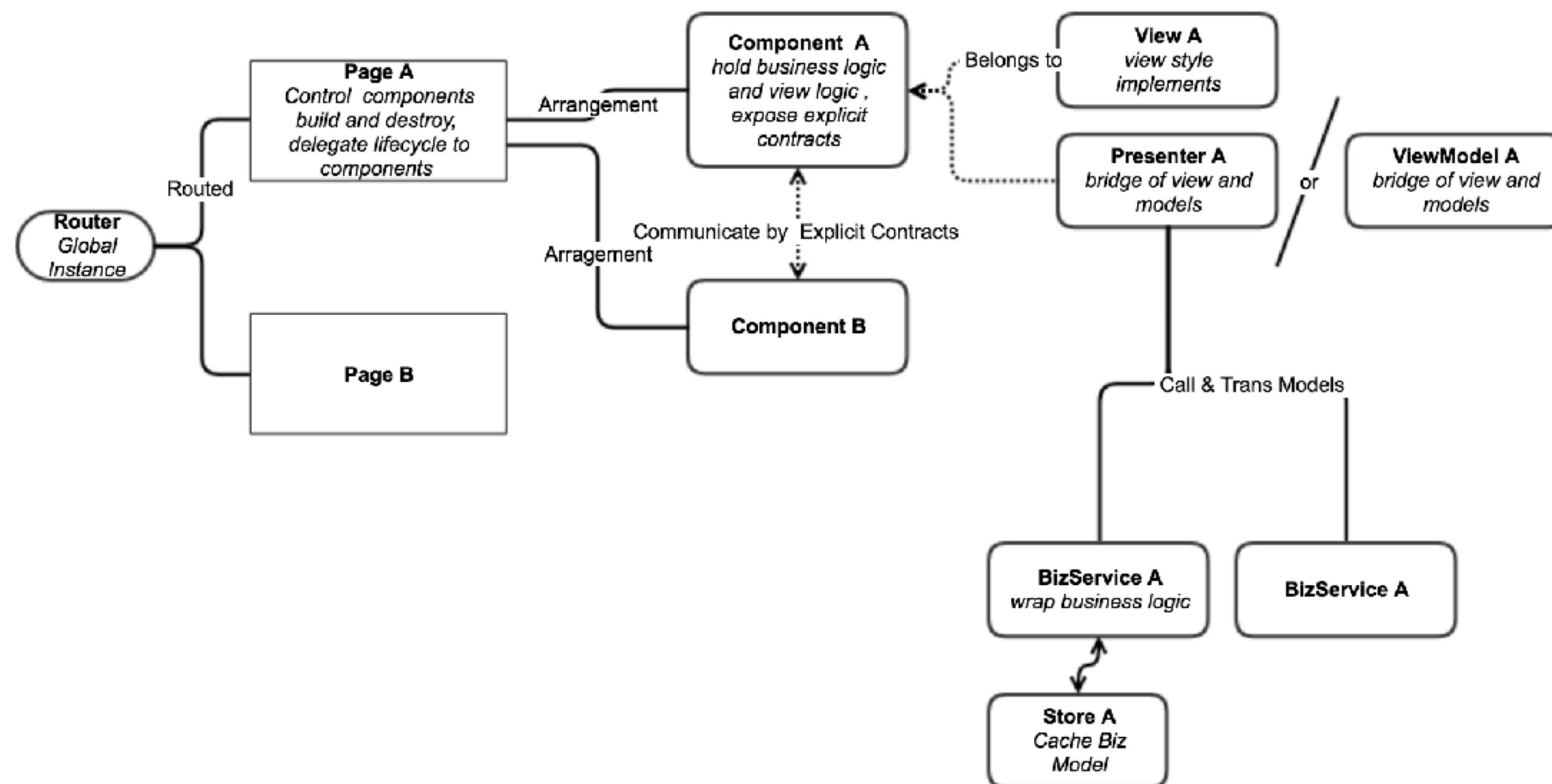
CHAPTER – 02

Nova Skeleton

- 1 分层
- 2 概念
- 3 Demo

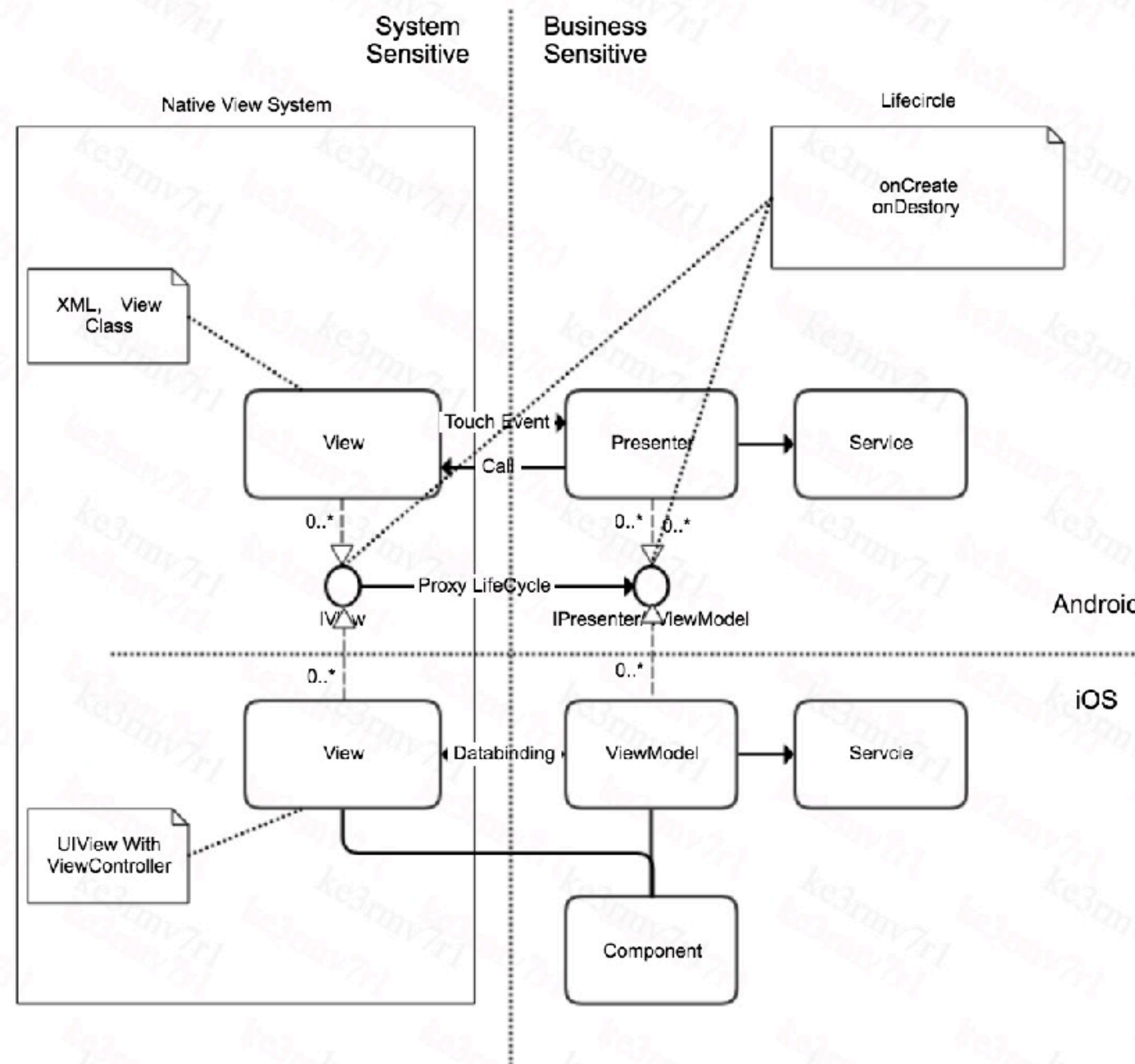
CHAPTER - 03

Skeleton 概念



CHAPTER - 03

Skeleton 概念





CHAPTER – 03

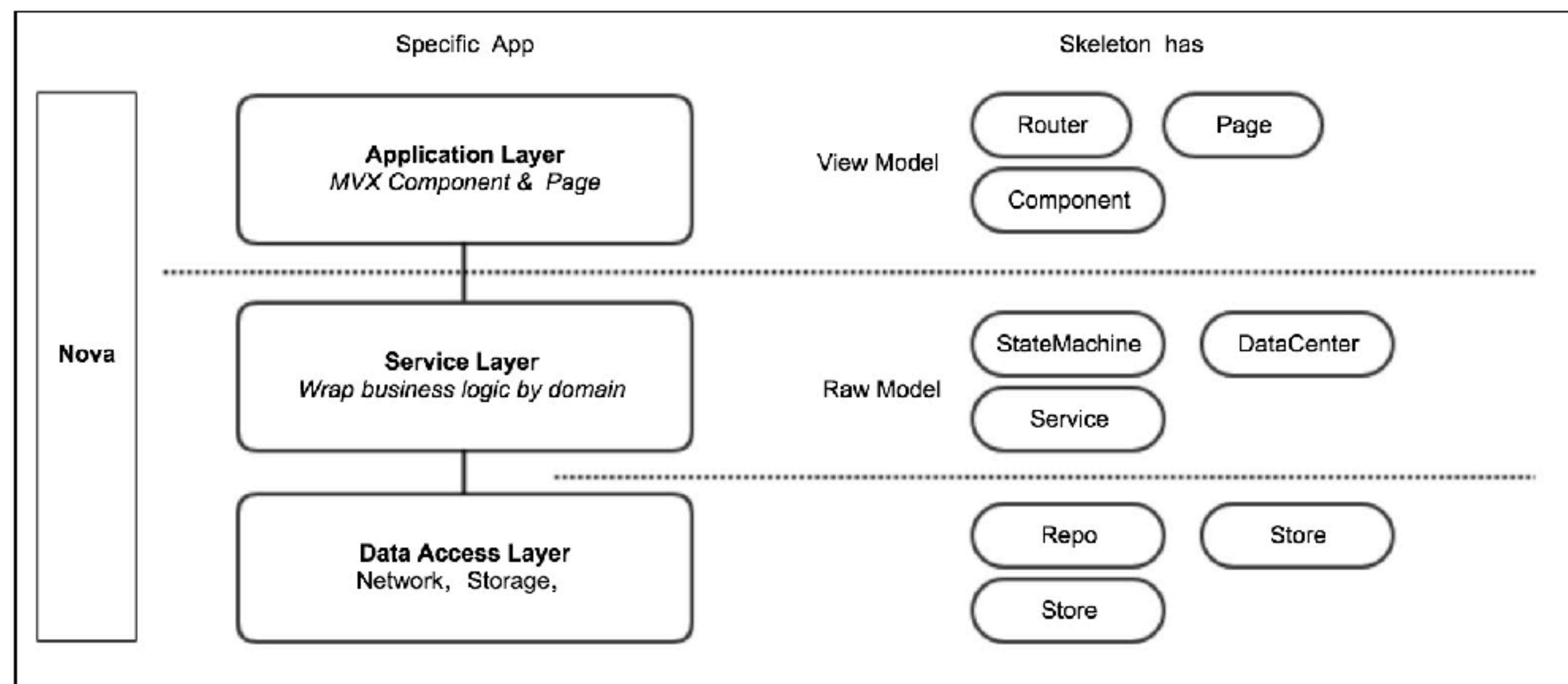
Skeleton 跨端

概念	职责	接口	Android	iOS
View	UI 展示能力	IView UIView	• onCreate() • onResume() • onPause() • onDestroy()	System Default
Presenter / ViewModel	UI 逻辑	IPresenter SKViewModel	• onCreate() • onResume() • onPause() • onDestroy()	
Service	业务逻辑封装			
Repo				
Component	管理View& View Model	IComponent SKComponent	• onCreate() • onResume() • onPause() • onDestroy()	- (void)viewDidLoad { } - (void)viewWillAppear { } - (void)viewDidAppear { } - (void)viewWillDisappear { } - (void)viewDidDisappear { }
ComponentManager	管理Components		- addComponent - removeComponent - getComponent	- (void)addComponent: - (BOOL)containsComponent: - (void)removeComponent:
Page	管理Components & 对接接口	BasePage/ MvPage SKPage	• onInitialize() • onCreate() • onResume() • onPause() • onDestroy() • onFinalize()	SKPageLifeCycleViewDidLoad = 0, // index 索引 SKPageLifeCycleViewDidLayoutSubviews, SKPageLifeCycleViewWillAppear, SKPageLifeCycleViewDidAppear, SKPageLifeCycleViewWillDisappear, SKPageLifeCycleViewDidDisappear
Router	管理Page的跳转	DiRouter	DiRouter.request().path("path").open()	



CHAPTER – 03

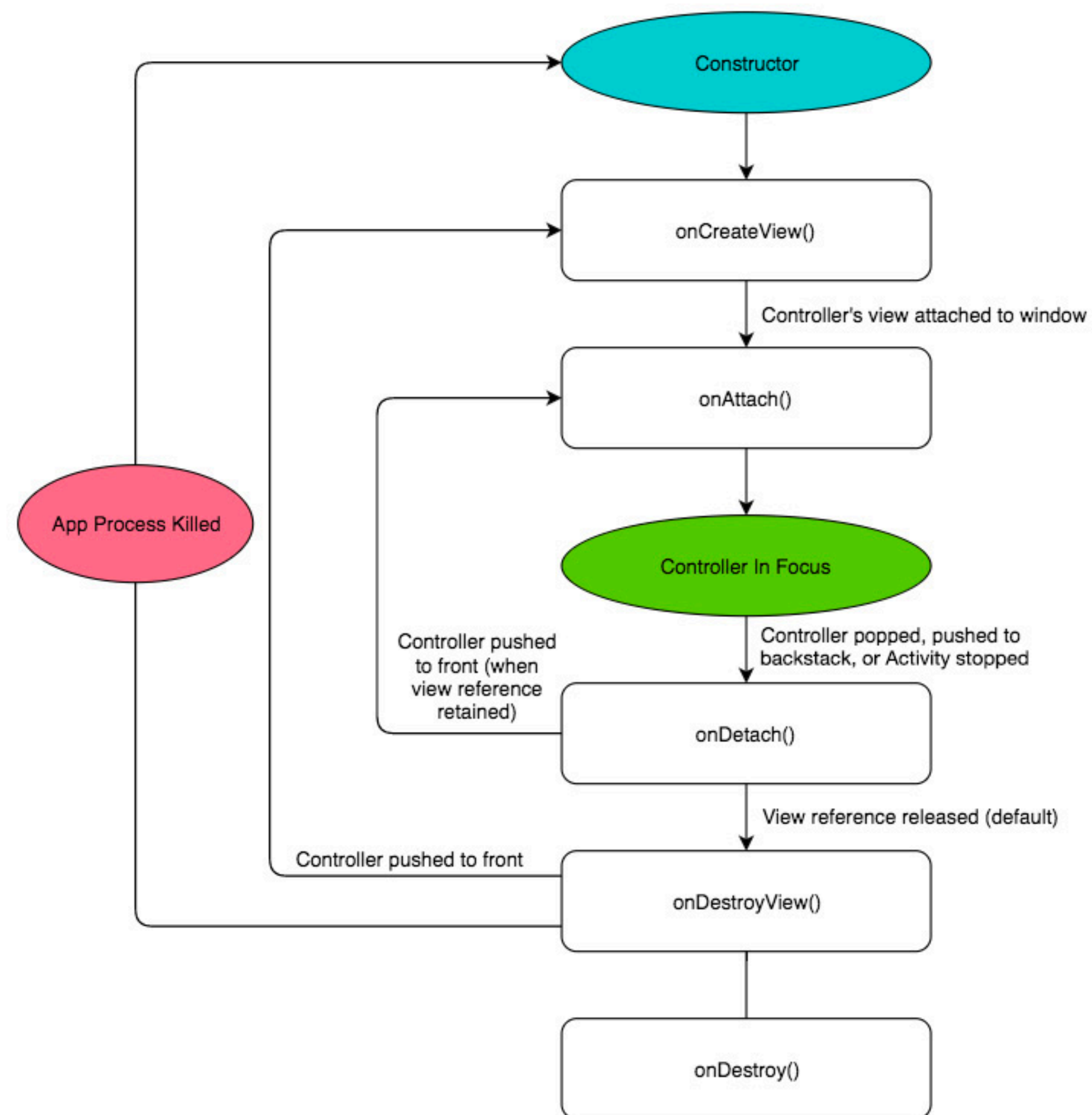
Skeleton – 分层





Conductor

<https://github.com/bluelinelabs/Conductor>





Dsl & Compiler

1. 使用注解的方式处理之前通过负载方法搞定的代码 — Java版本的实现稍显拙略 (Now)
2. 在Kotlin的Skeleton版本中提供更合理的实现 (Future)

Router

1. 实现页面之间的解耦 (Now)
2. 参考前端Router的实践, 彻底实现View的无状态化 (Future)



Skeleton Code

```
java
└─ com.didi.app.nova.skeleton
   └─ conductor
   └─ internal
   └─ mvvm
   └─ props
   └─ repo
   └─ title
      ├── BaseActivity
      ├── BaseComponent
      ├── BasePage
      ├── Component
      ├── ComponentGroup
      ├── INavigator
      ├── InnerPage
      ├── IPageLifecycle
      ├── IPresenter
      ├── IView
      ├── LifecycleObserver
      ├── LiveHandler
      ├── LiveHandlerOwner
      ├── MvComponent
      ├── MvModel
      ├── MvPage
      ├── MvView
      ├── Page
      ├── PageFactory
      ├── PageInstrument
      ├── PageInstrumentFactory
      ├── RestrictContext
      ├── Scope
      ├── ScopeContext
      ├── Service
      ├── ServiceRegistry
      ├── SimplePageLifecycleCallback
      ├── Skeleton
      ├── ViewDelegate
      └── ViewHolder
```

```
NVSkeleton
└─ Classes
   └─ Component
      ├── SKComponent.h
      ├── SKComponent.m
      └── SKComponentManger.h +9
         ├── SKComponentManger.m
         ├── SKViewModel.h
         ├── SKViewModel.m
         └─ Page +9
            ├── SKPage.h
            ├── SKPage.m
            └─ Responder
               ├── UIResponder+Skeleton.h
               ├── UIResponder+Skeleton.m
               └─ Router
                  ├── SKRequest.h
                  ├── SKRequest.m
                  ├── SKResponse.h
                  ├── SKResponse.m
                  ├── SKRouter.h
                  └── SKRouter.m
               └─ Service
                  ├── SKBizService.h
                  ├── SKBizService.m
                  ├── SKServiceManager.h
                  └── SKServiceManager.m
   └─ Resources
      └── NVSkeleton.h
```




Page

```
/**
 * Created by Alizon on 2018/26/02.
 */
@PageDsl(
    layout = R.layout.mvvm_page,
    components = {
        @Stub(component = MvComponent1.class, containerId = R.id.com1),
        @Stub(component = MvComponent2.class, containerId = R.id.com2),
    }
)
public class MvvmMainPage extends MvPage {

    private MvRv mModel;

    public MvvmMainPage(Bundle args) { super(args); }

    @Override
    protected void onCreate(View view) {
        super.onCreate(view);
        MvvmPageBinding binding = (MvvmPageBinding) getBinding();
        mModel = new MvRv();
        mModel.name = "MvVM";
        binding.setModel(mModel);
    }
}
```

Component

```
/**
 * Created by Alizon on 2018/08/02.
 */
@ComponentDsl(view = MvComView1.class,
    model = MvComModel1.class,
    layout = R.layout.mv_view1)
public class MvComponent1 extends MvComponent {

    @Override
    protected void onCreate() {
        super.onCreate();
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
    }
}
```

View

```
/**
 * Created by Alizon on 2018/08/02.
 */
class MvComView1 extends MvView<MvComModel1, MvView1Binding> {
    @Override
    protected void onCreate() {
        super.onCreate();
        // TODO 省略掉
        mViewModel.name = getClass().getSimpleName();
        mBinding.setModel(mViewModel);
    }
}
```

ViewModel

```
/**
 * Created by Alizon on 2018/08/02.
 */
public class MvComModel1 extends MvModel implements DslIModel {
    public String name;
    private int count;

    public void onClick() { getProps().put( k: "key", v: "" + (++count)); }

    @Override
    public void onPropsChanged(String key, PropsMap map) {
        super.onPropsChanged(key, map);
    }
}
```

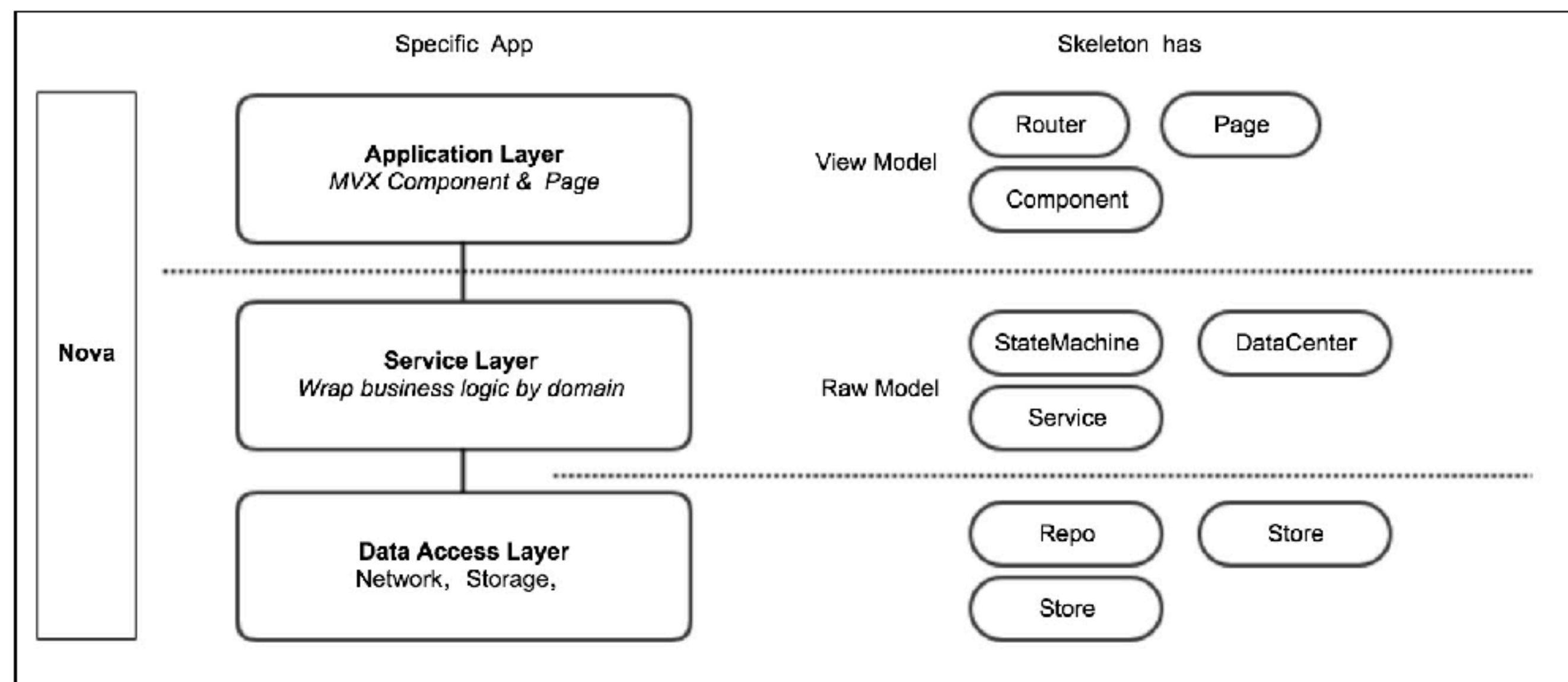



Demo



CHAPTER – 03

Thinking



Thinking (如何思考的)

1. **Ability** (DiDi Ability) — Nova Foundation
2. **Principle** — Nova Skeleton
3. **Utility** — Nova Support / Design / Assembly
4. **Engineering** — Nova Dev Support
5. **Business** — Specific Applications



CHAPTER – 05

Goto

方向	已经支持的功能	后续
Router	1. 页面间跳转 2. 跳转拦截	1. 打通跟H5& Hybird 技术方案的交互 2. 服务器端可配置
Component	1. MVP 2. MVVM a. MVVM的数据绑定 b. Page内组件间通信	1. Kotlin 语言的 Dsl 版本 2. DataBinding 对RecyclerView 的支持
Page	1. MVP 2. MVVM	2. Kotlin 语言的Dsl 版本
Repo	1. Repo a. Live Data b. autorelease	1. Repo 模板
Service		1. 统一的Service注册及发现 a. Service的生命周期管理
Build	1. Mono Repo Submoudle 的代码管理	1. 统一的cli 工具 a. 增加项目 b. 统一提交 c. 抓取代码 2. 脚本的统一
Design	无（立项阶段）	1. 统一的Widget 库 2. 为 Widget 提供 DataBinding 支持 3. Spinner 的富文本支持
Hybird	无（立项阶段）	1. Weex 作为一些弱交互界面的解决方案



CHAPTER – 06

Other

- 1 | 思维纠偏
关于业务开发工程师的一些误解
- 2 | 走出舒适区
走出任何领域的舒适区
- 3 | 思考本质
思考解决的本质问题
- 4 | 理解业务, 助力业务
Be a member of team
- 5 | 迁移
迁移自己的能力
- 6 | 机遇
Be a lucky man



Q&A