

可 Web 编程的NativeUI 设计与实现

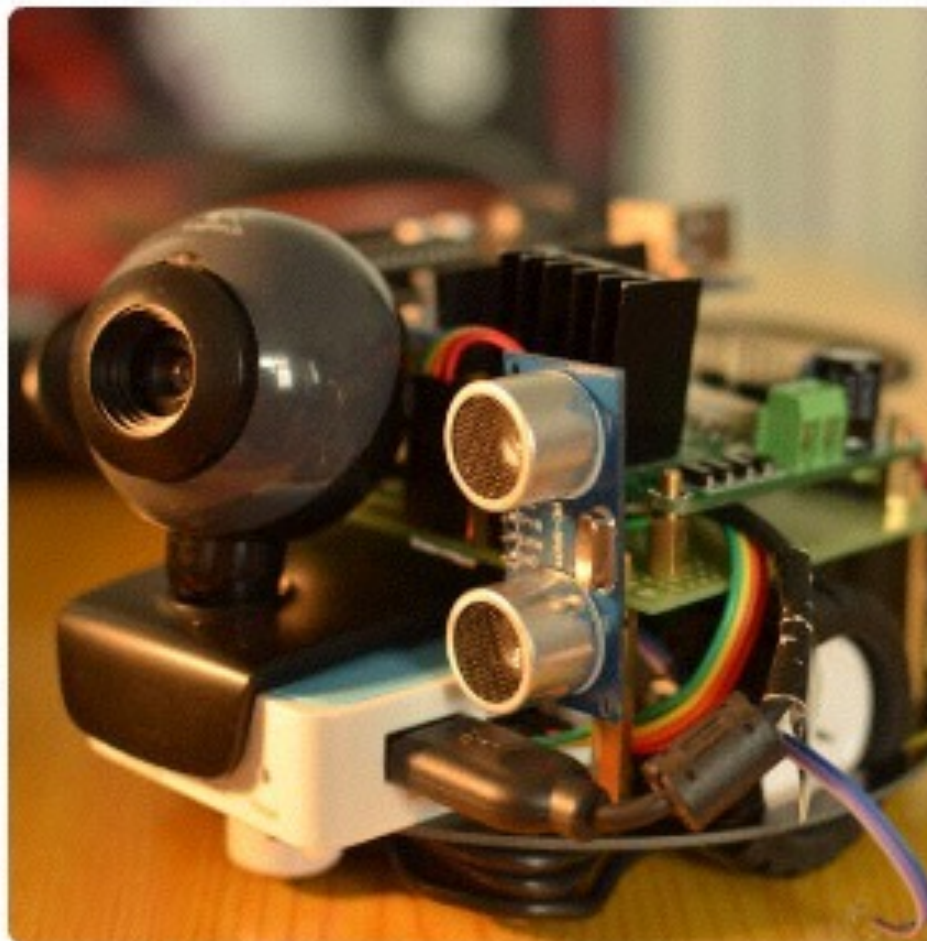
——张袁炜

About Me

- @张袁炜
- 直达号、网址导航、百度音乐
- 前端、Node.js、Android
- Arduino、RaspberryPI
- <https://github.com/zhangyuanwei>



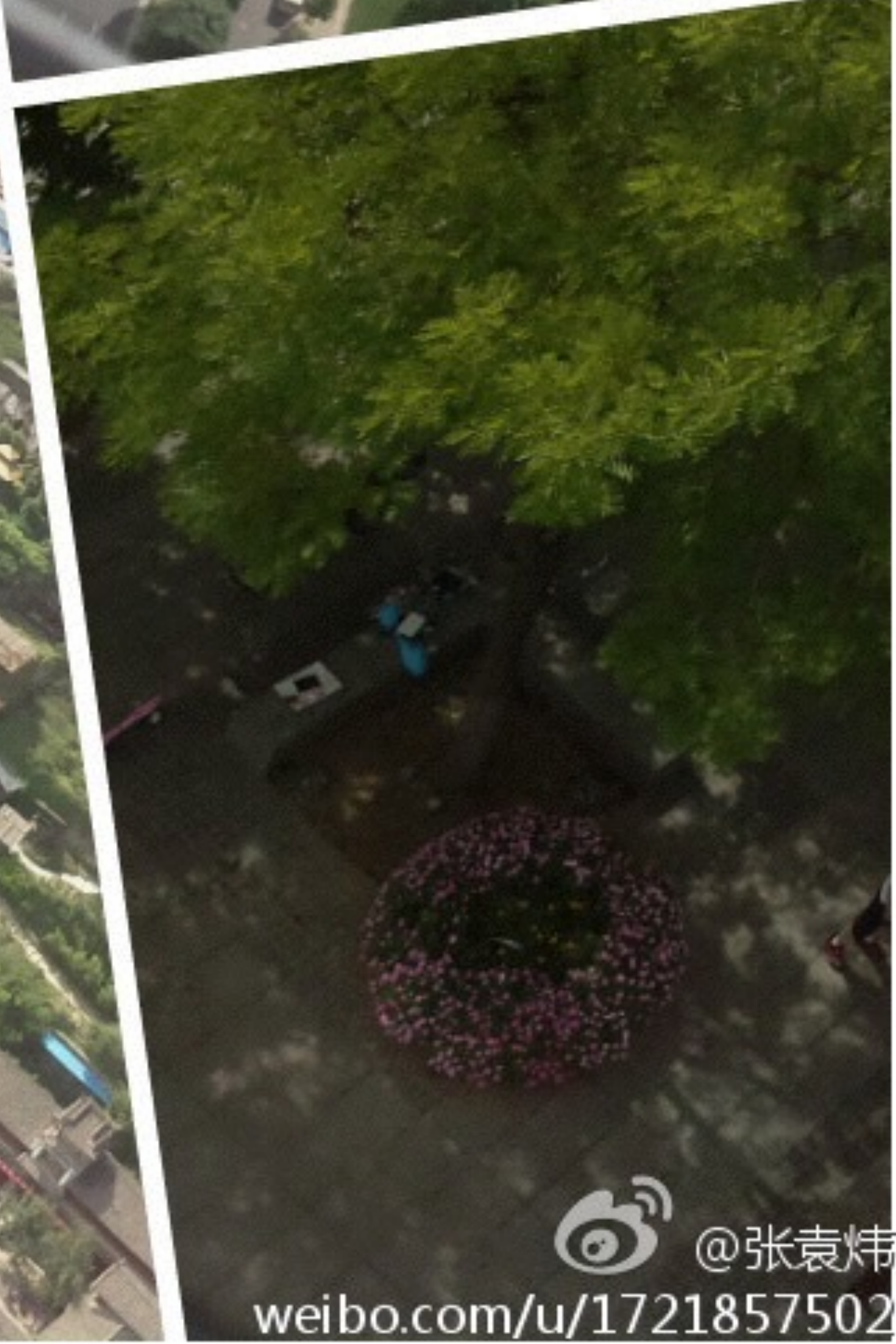
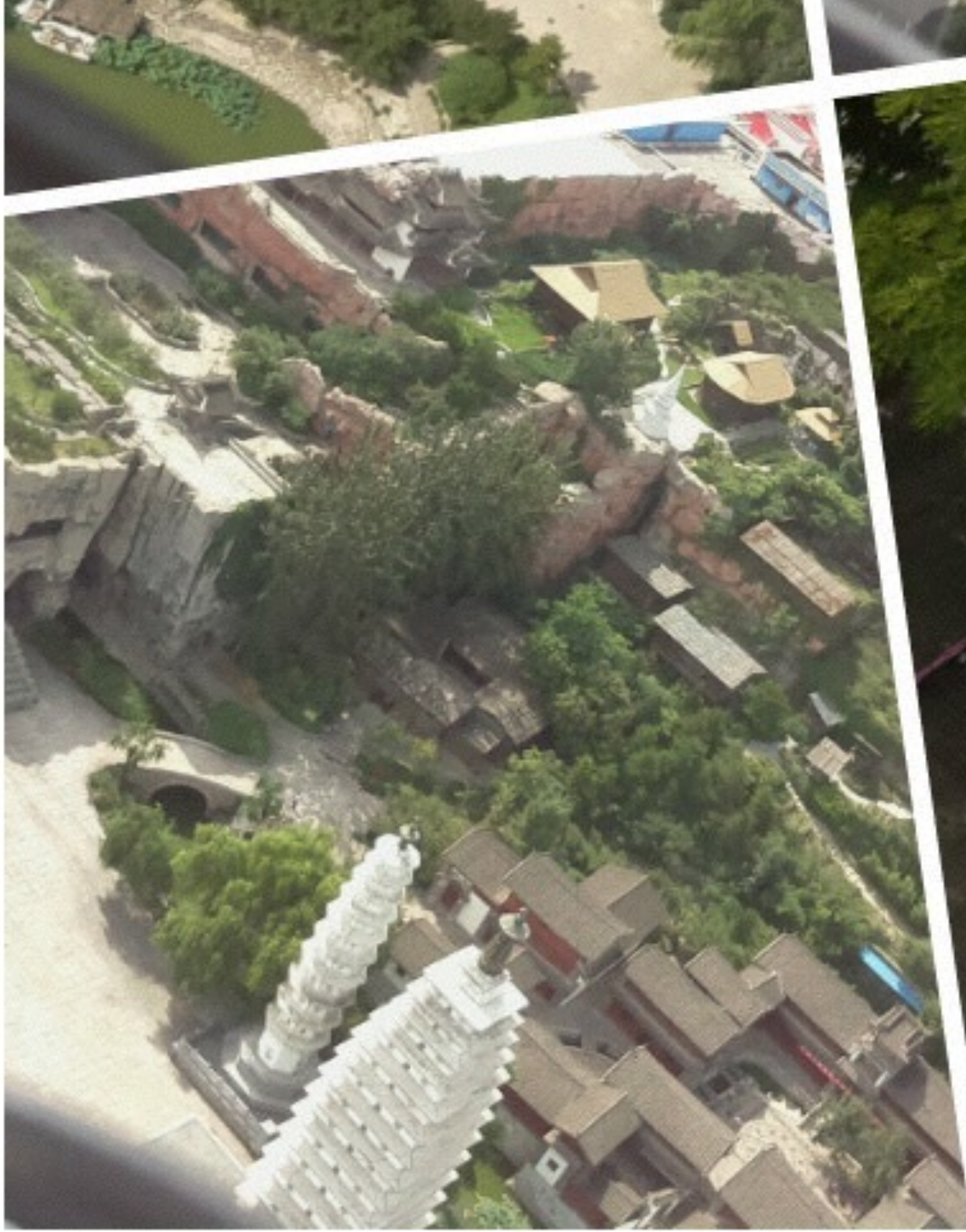
About Me



@张袁伟
weibo.com/u/1721857502

About Me

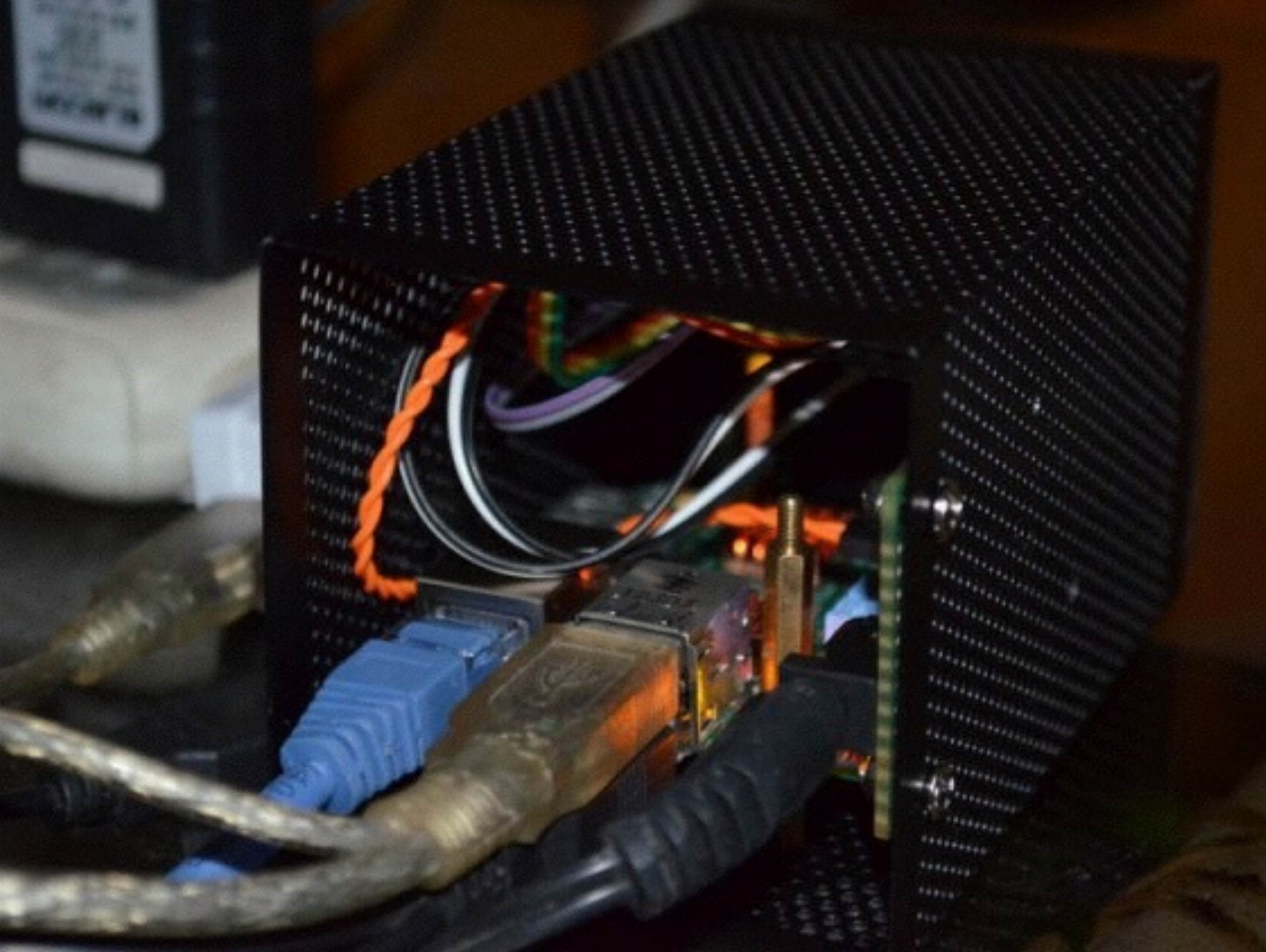




@张袁炜

weibo.com/u/1721857502

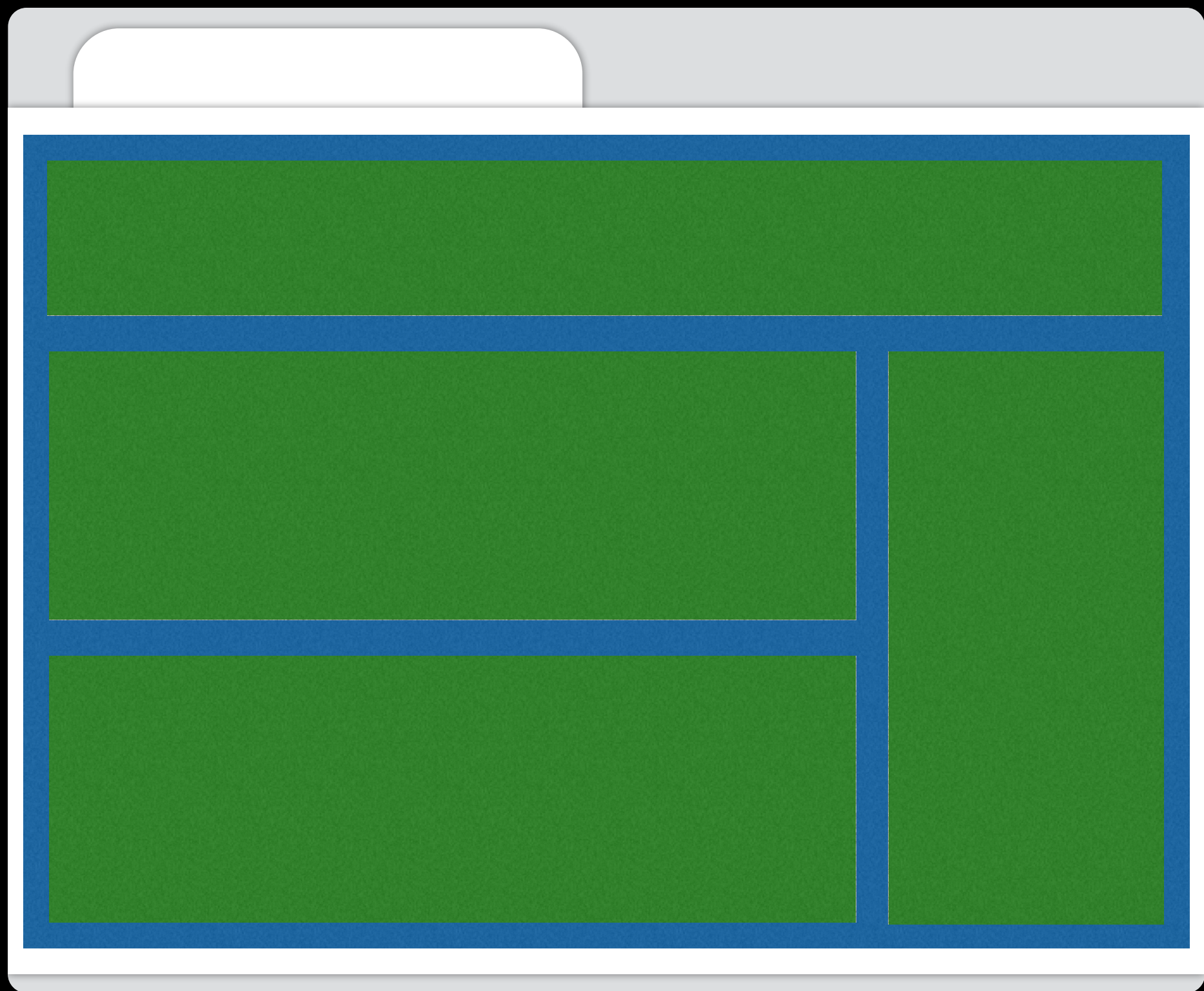
TIME: 2012.11.08 00:56:04
ETH0: 192.168.1.2
CPU0: 700MHz 9%
MEM: 280/483MB 78%
TEMP: 41.698 °C



Outline

- 一个 Hybrid App 的进化史
- Native UI 实现原理
- 架构设计
 - UI构建
 - 布局系统
 - Web 样式支持
 - 事件机制
- 性能优化
- API 设计

一个App的进化



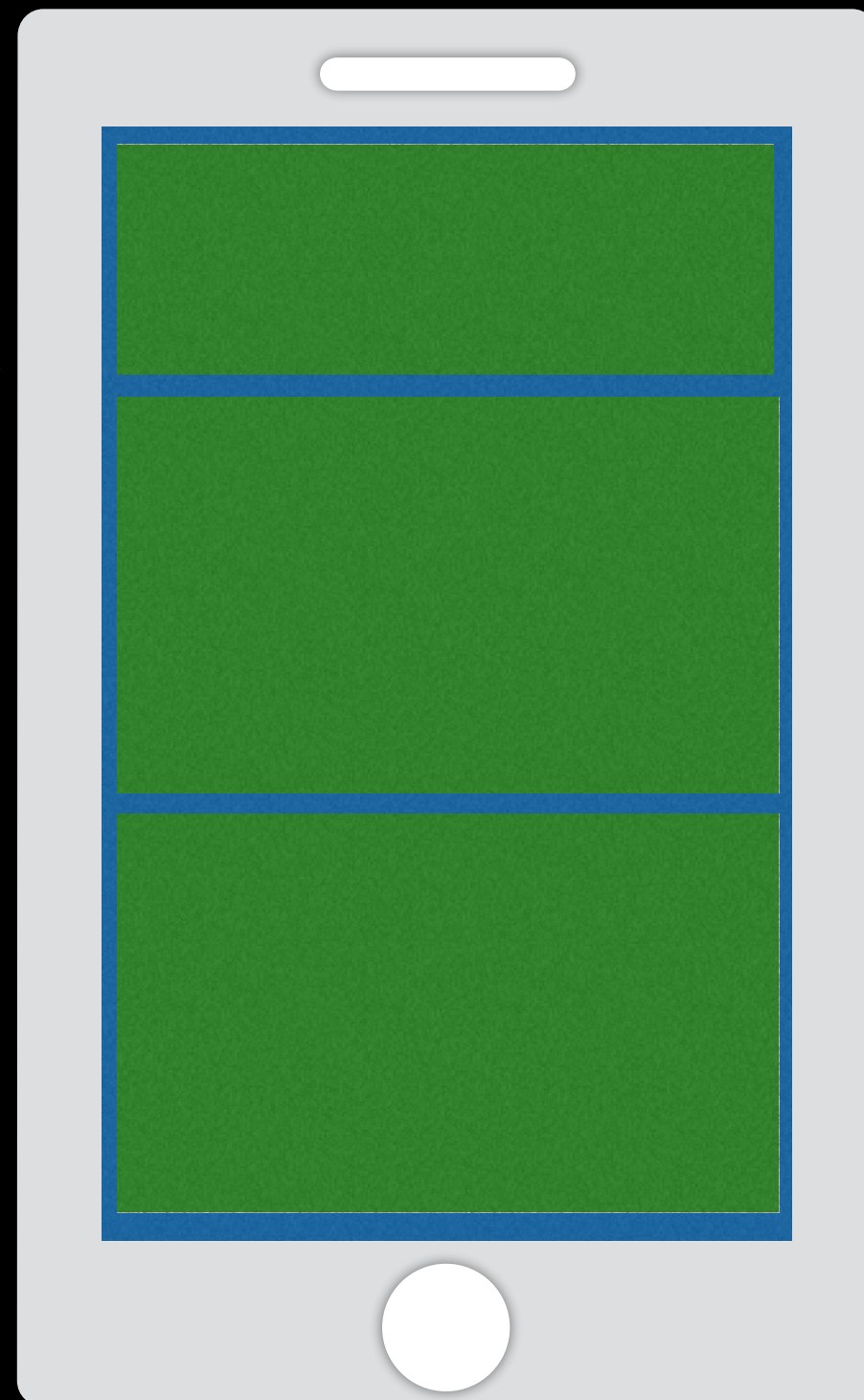
一个App的进化

地理位置

分享

摄像头

支付



一个App的进化

地理位置

分享

体验

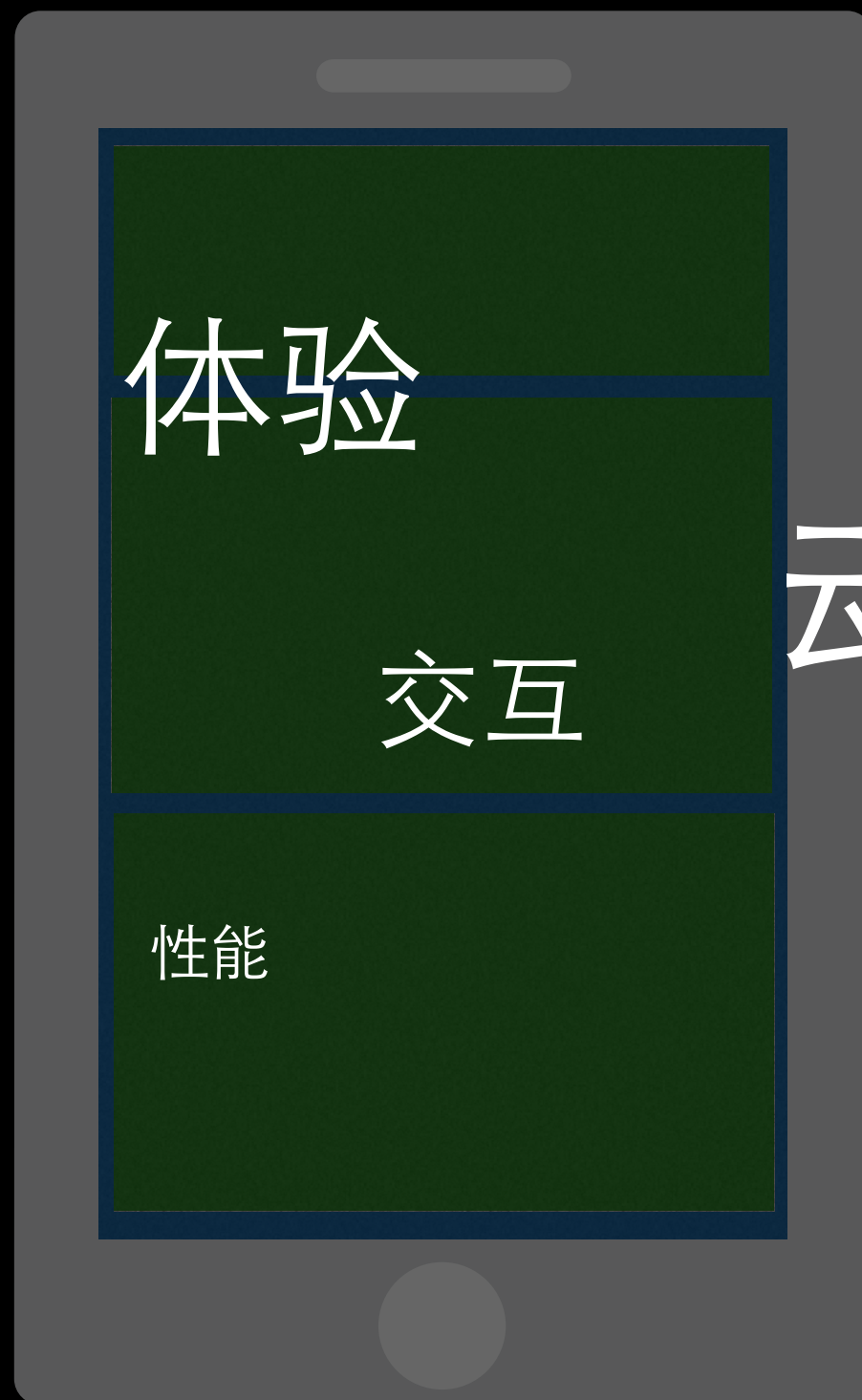
交互

动画

性能

摄像头

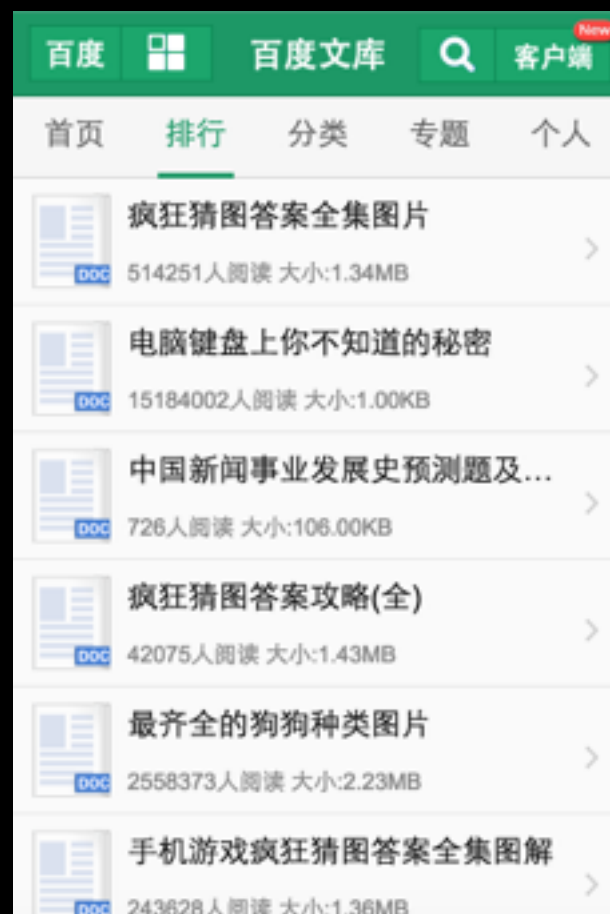
支付



一个App的进化



一个App的进化



一个App的进化

CSS3 硬件加速?

改 WebKit?

单页面应用?

手势跟随?



一个App的进化



一个App的进化



WebView



WebView



WebView

background page

```
// 系统原生事件，
// 新activity创建的时候触发webview_register
// activity销毁的时候触发webview_unregister
// var _events = ['webview_register',
var bridge = window.lc_bridge;
var webViews = {};
var status = {
  ACTIVE: 1,
  PAUSED: 2,
  STOPPED: 3,
  KILLED: 0
};
// 监听activity webView创建事件
document.addEventListener('webview_reg
  if (!(event.origin in webViews)) {
    webViews[event.origin] = {
      origin: event.origin,
      status: status.ACTIVE,
      data: event.data
    };
  }
});
```


代码重构

内存泄露

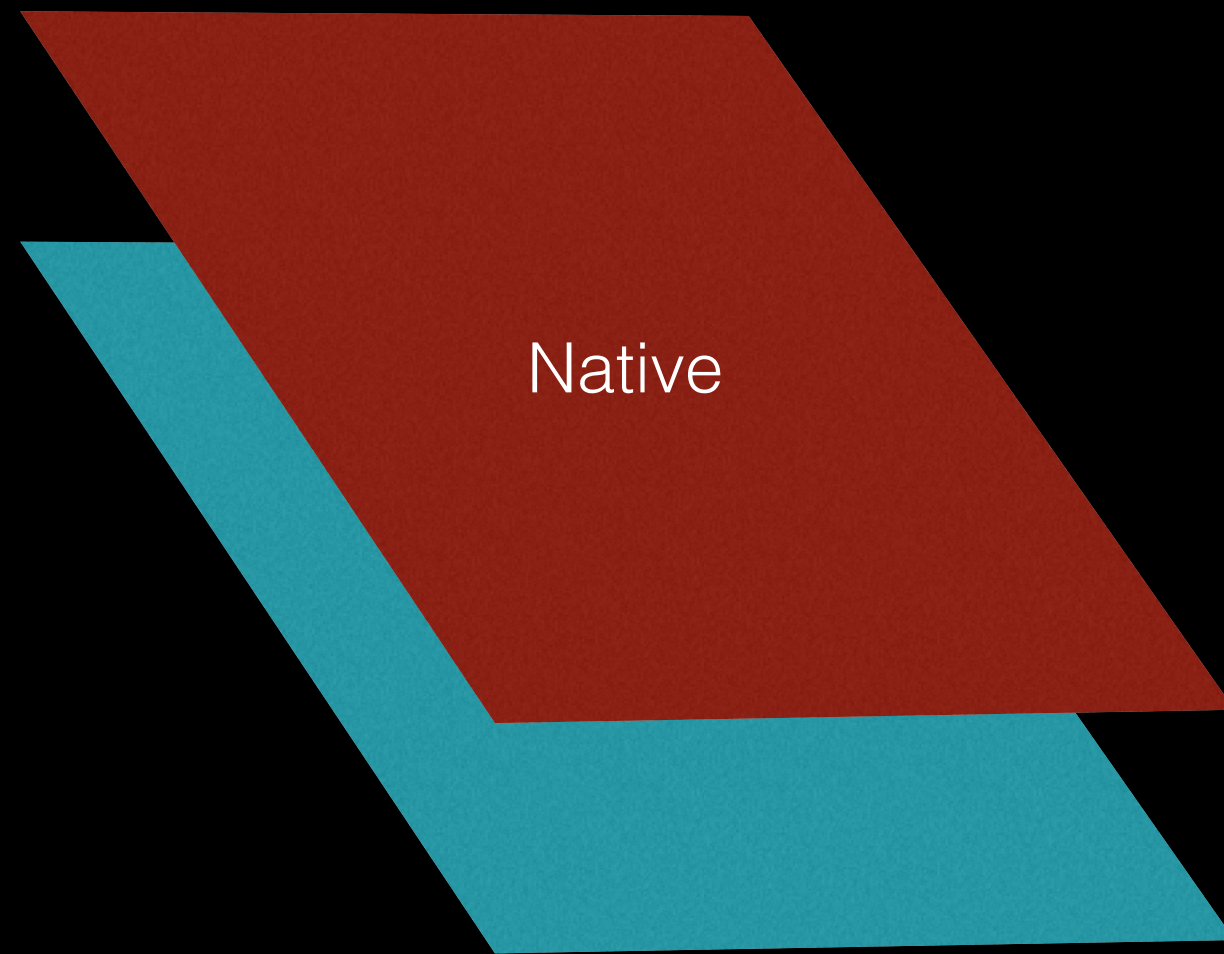
多页面通讯



一个App的进化

- 性能好（媲美Native）
- 定制性强（媲美Web）
- 学习成本低（最好为零）

页面增强

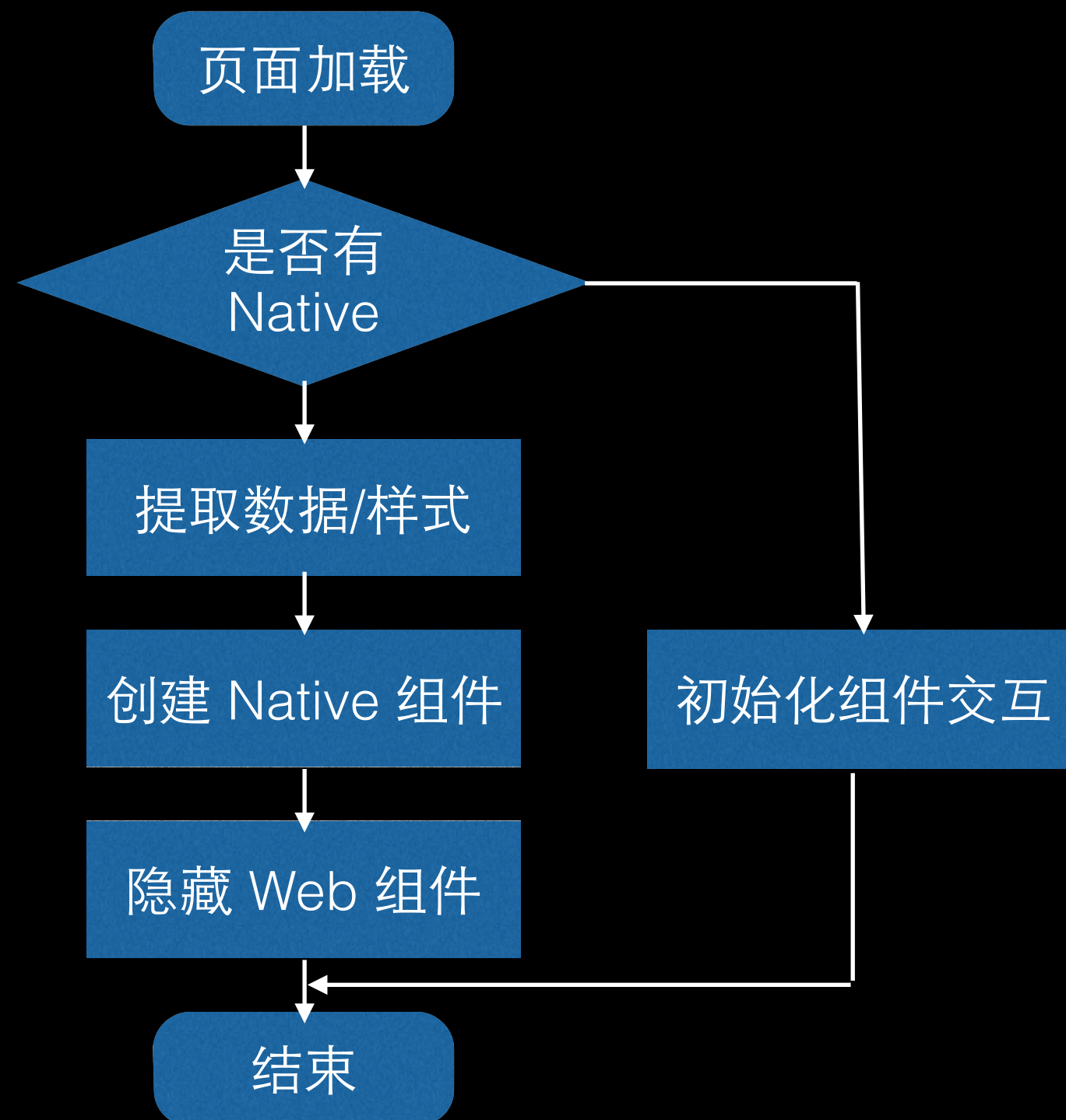


- 底层为 WebView
- 覆盖透明 Native 组件层
- Native 组件替换 Web 组件

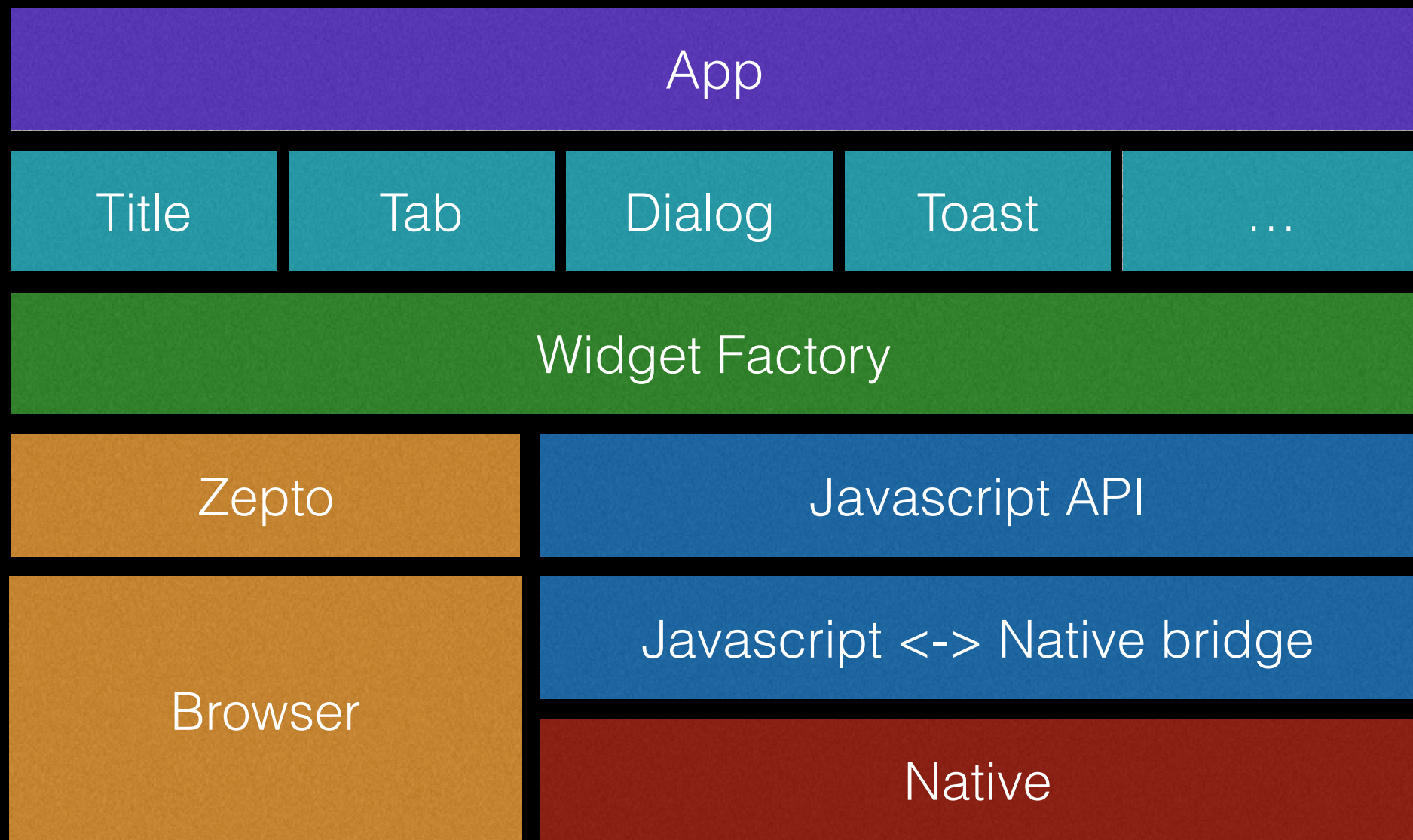
页面增强



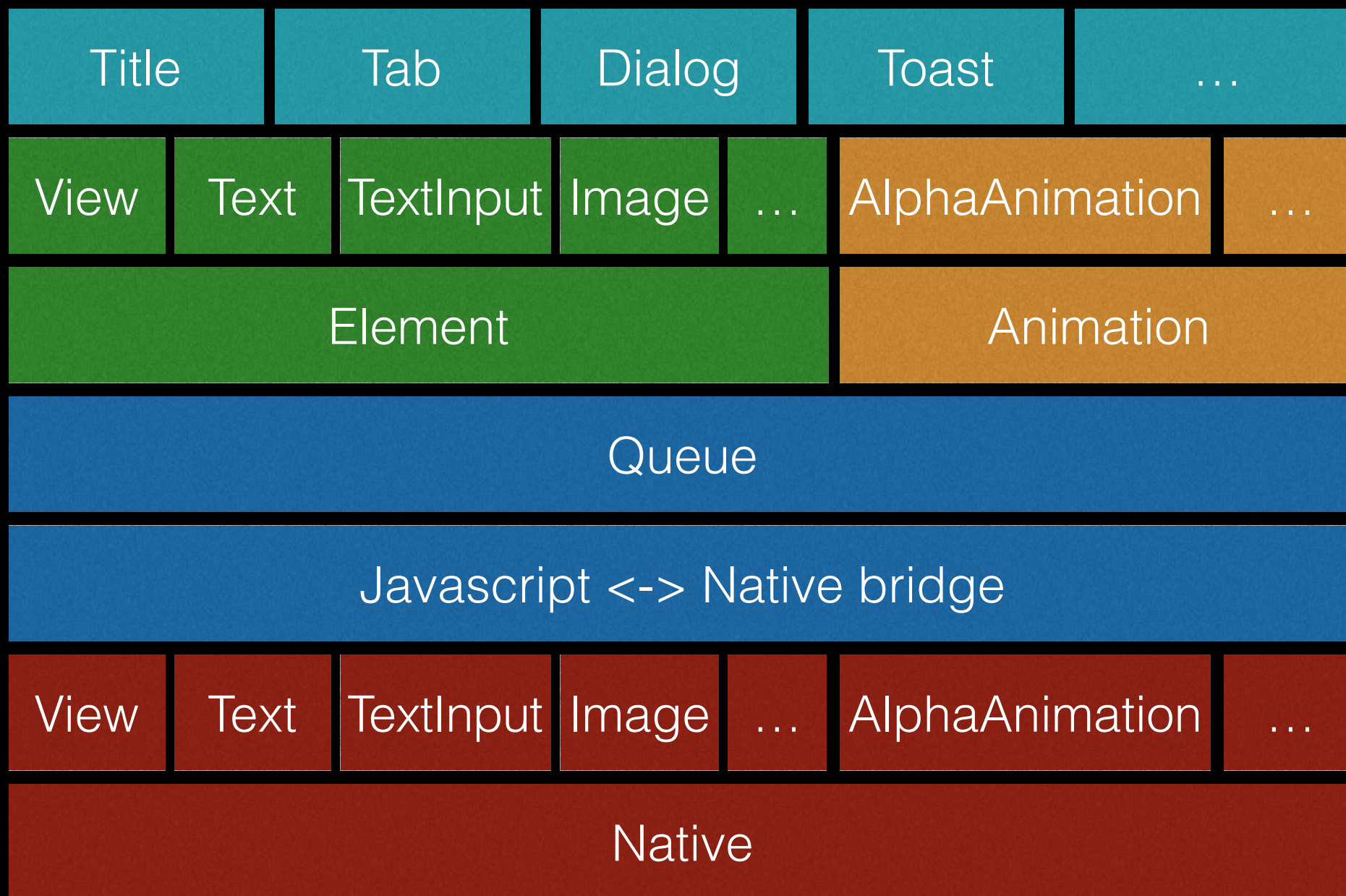
页面增强



Blend UI 系统架构



Blend UI 系统架构



UI 构建

提示信息	
提示内容	
确定	取消

UI 构建

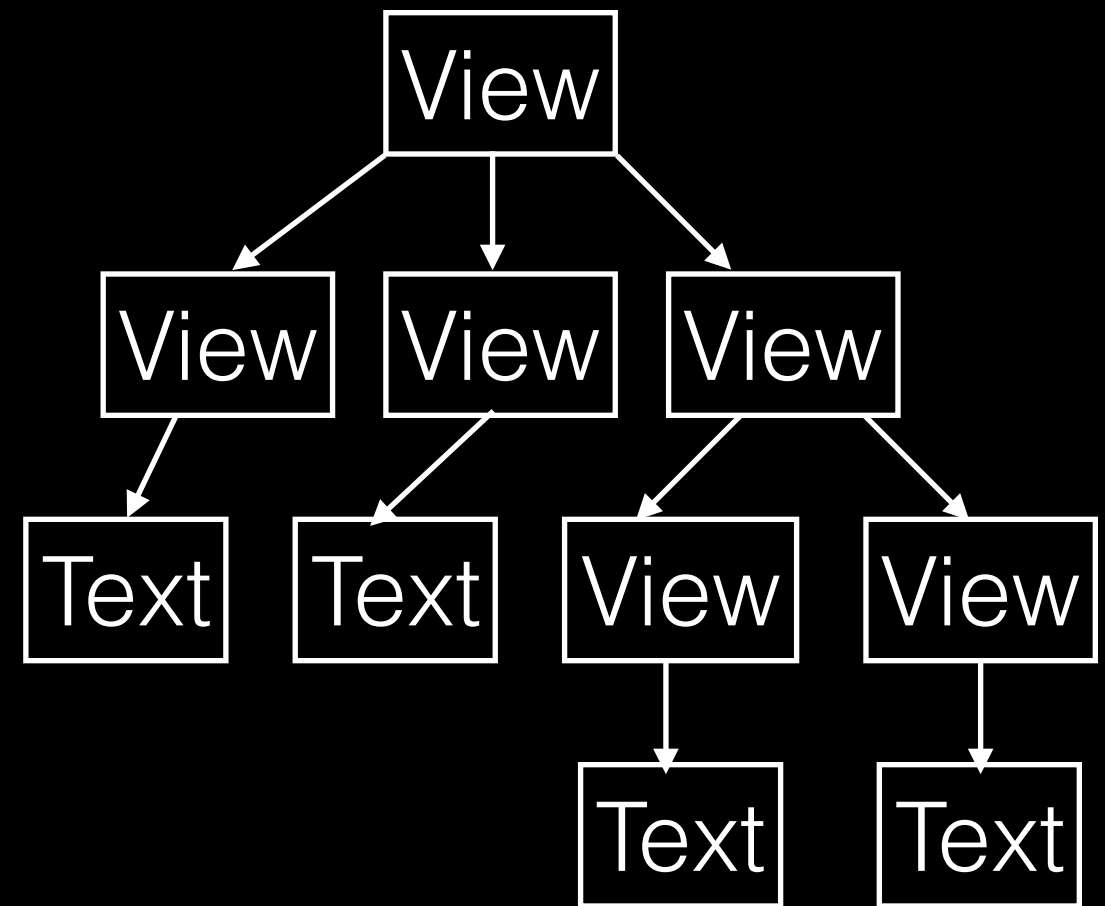
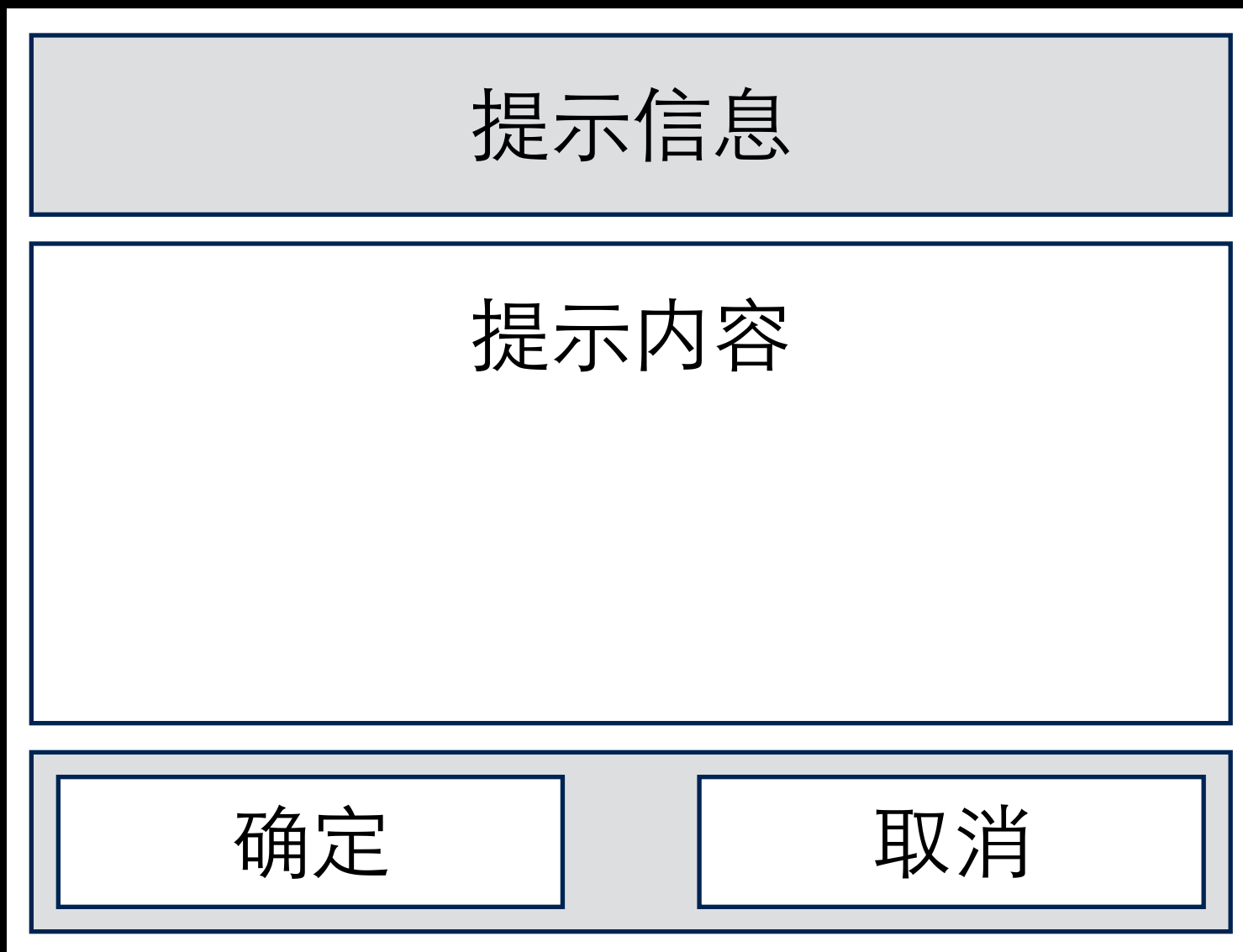
提示信息

提示内容

确定

取消

UI 构建



UI 构建

`createView("tag":string, "type":string, "config":object)`

根据给定的 config 配置，创建一个类型为 type 的基础 GUI 组件，并指定 tag。

`addView("parentTag":string, "childTag":string, "index":number)`

添加 childTag 所指定的 View 到 parentTag 所指定 View 的 index 位置。

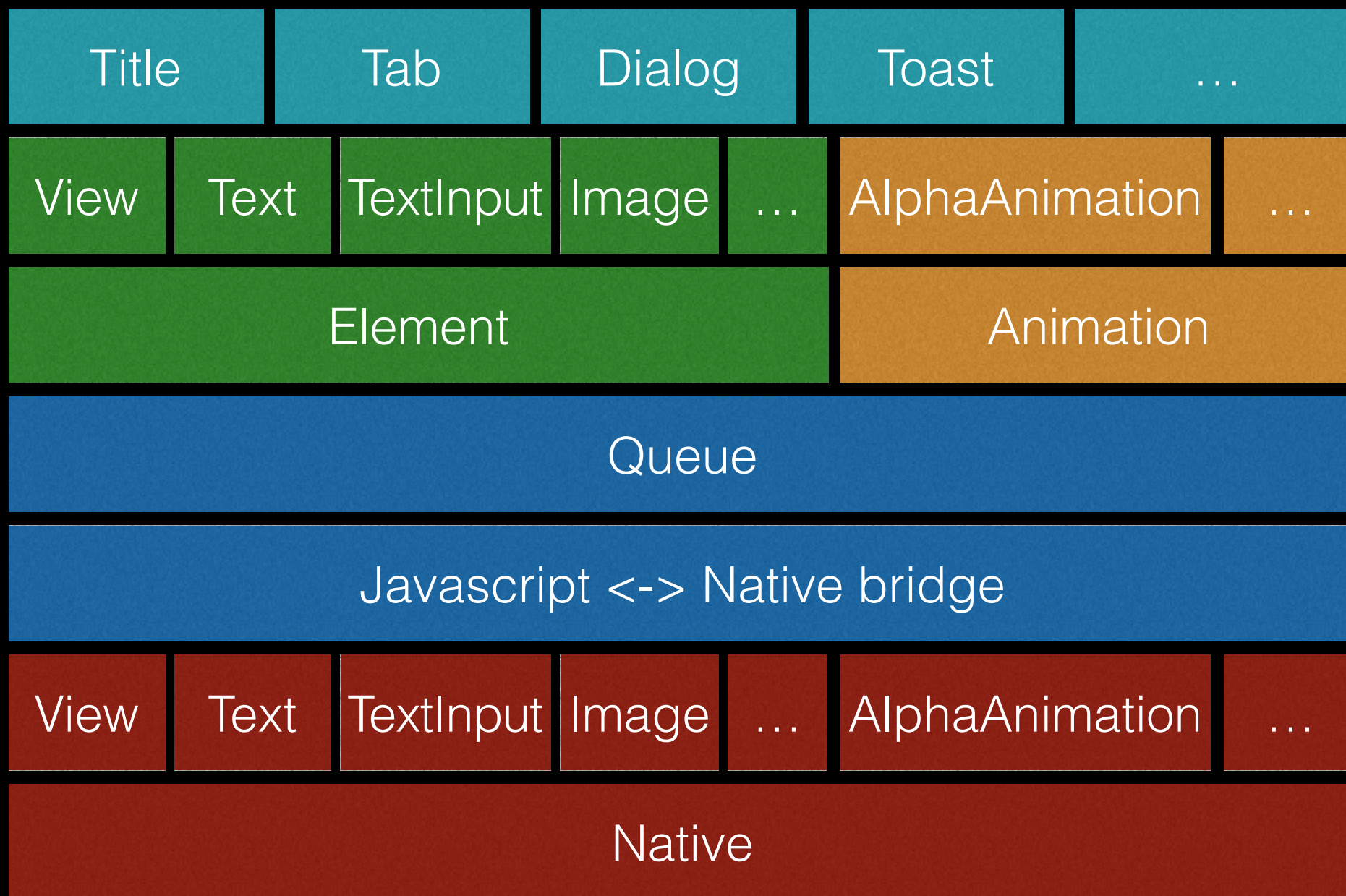
`removeView("parentTag":string, "index": number)`

从 parentTag 所指定 View 删除 index 位置的子 View。

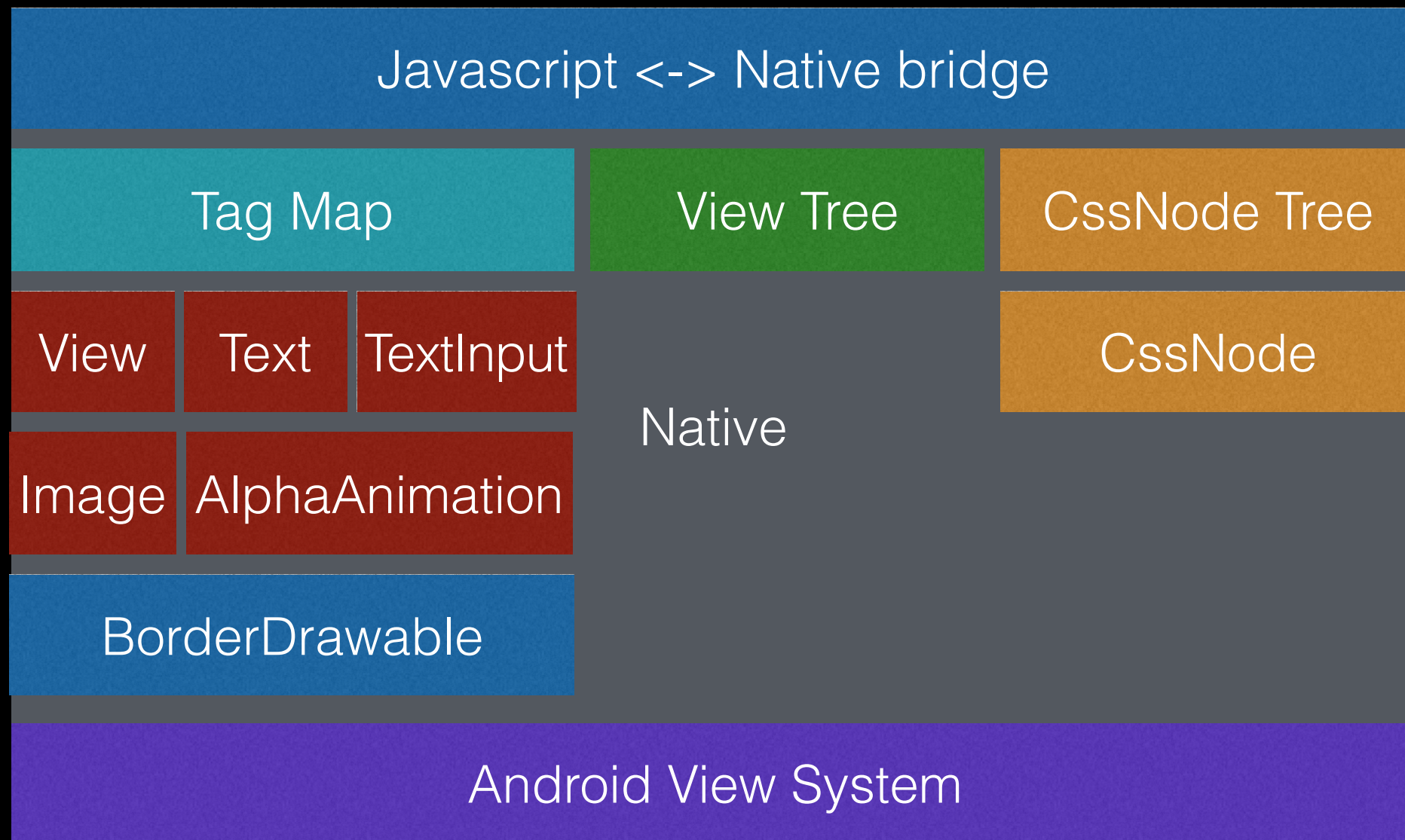
`updateView("tag":string, "type":string, "config":object)`

更新 tag 所指定的 View。

Blend UI 系统架构



Blend UI 系统架构



CSS Layout

CssNode

```
public void addChildAt(CSSNode child, int i)
public void removeChildAt(int i)

public void setFlexDirection(CSSFlexDirection direction)
public void setJustifyContent(CSSJustify justifyContent)
public void setAlignItems(CSSAlign alignItems)
public void setAlignSelf(CSSAlign alignSelf)
public void setPositionType(CSSPositionType positionType)
public void setWrap(CSSWrap flexWrap)
public void setFlex(float flex)
public void setMargin(int spacingType, float margin)
public void setPadding(int spacingType, float padding)
public void setBorder(int spacingType, float border)
public void setPositionTop(float positionTop)
public void setPositionBottom(float positionBottom)
public void setPositionLeft(float positionLeft)
public void setPositionRight(float positionRight)
public void setStyleWidth(float width)
public void setStyleHeight(float height)
```

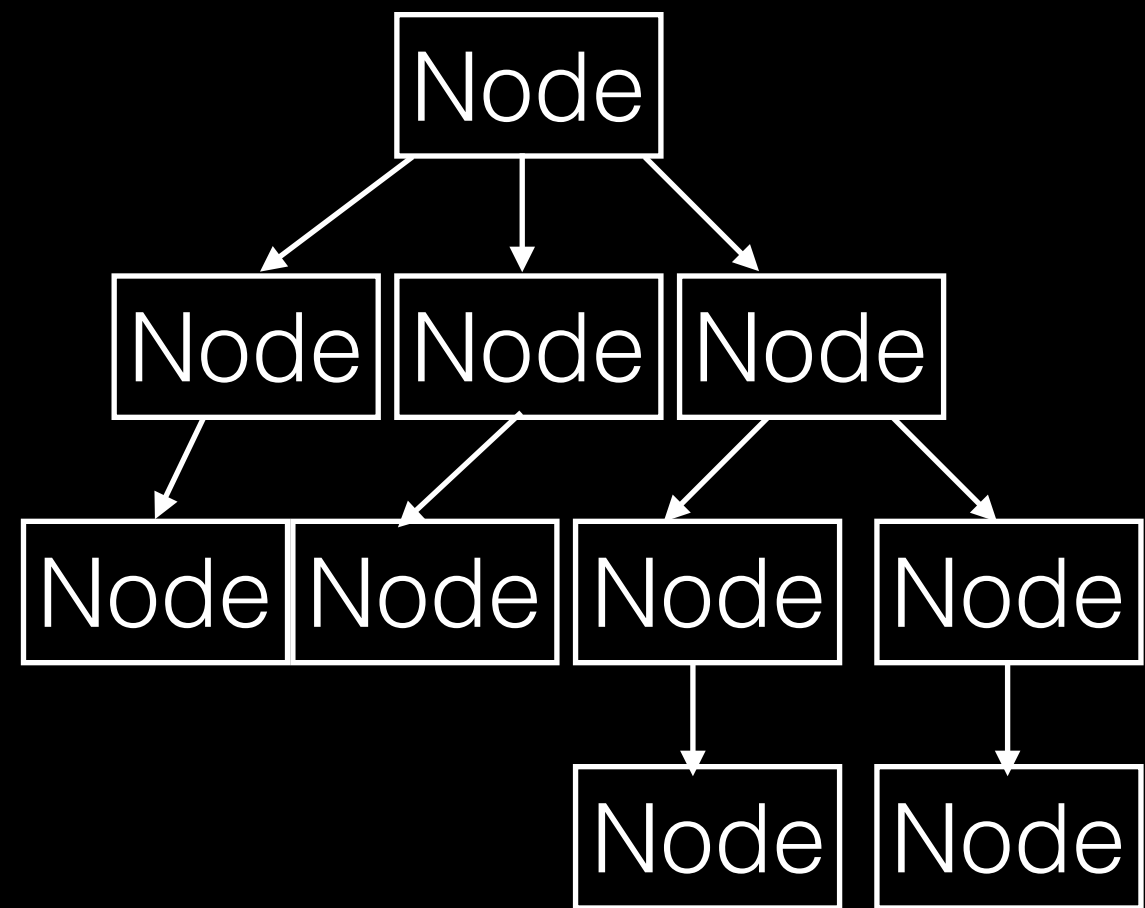
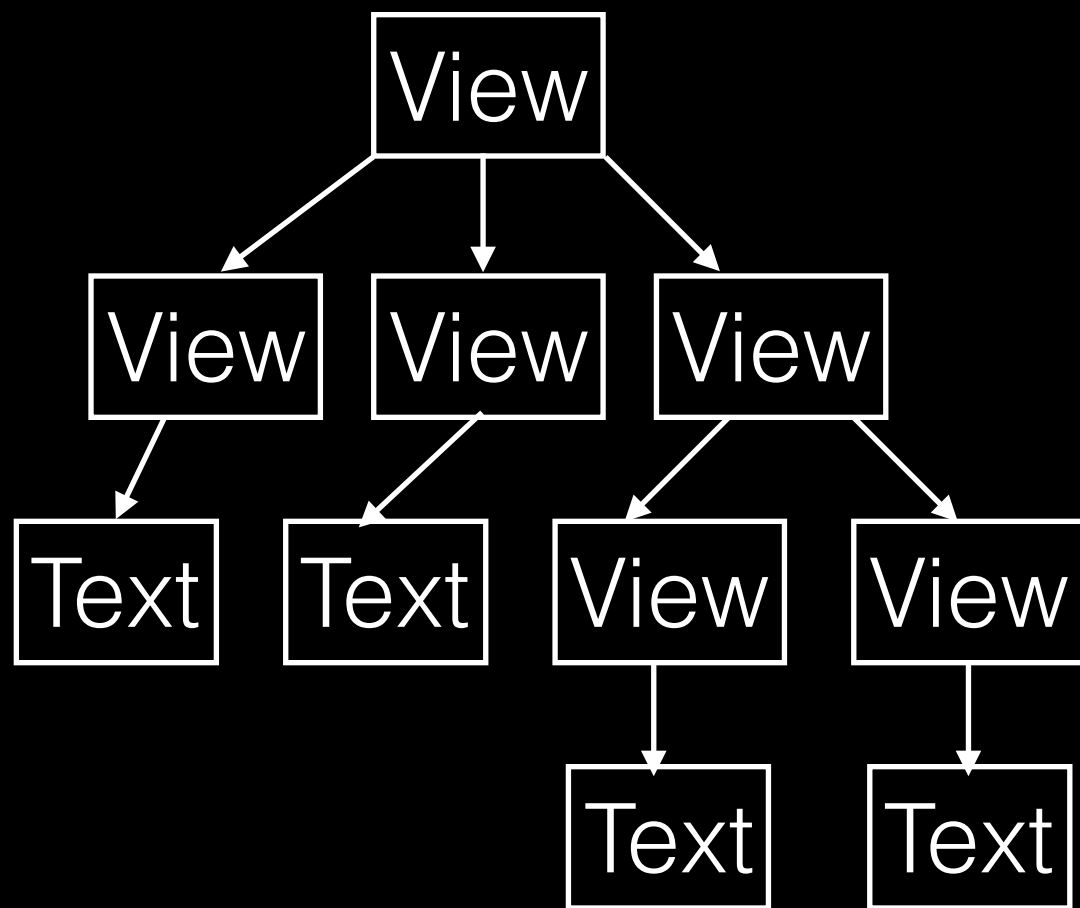
CSS Layout

CssNode

```
public void calculateLayout()  
public boolean hasNewLayout()  
public void markLayoutSeen()  
  
public float getLayoutX()  
public float getLayoutY()  
public float getLayoutWidth()  
public float getLayoutHeight()
```

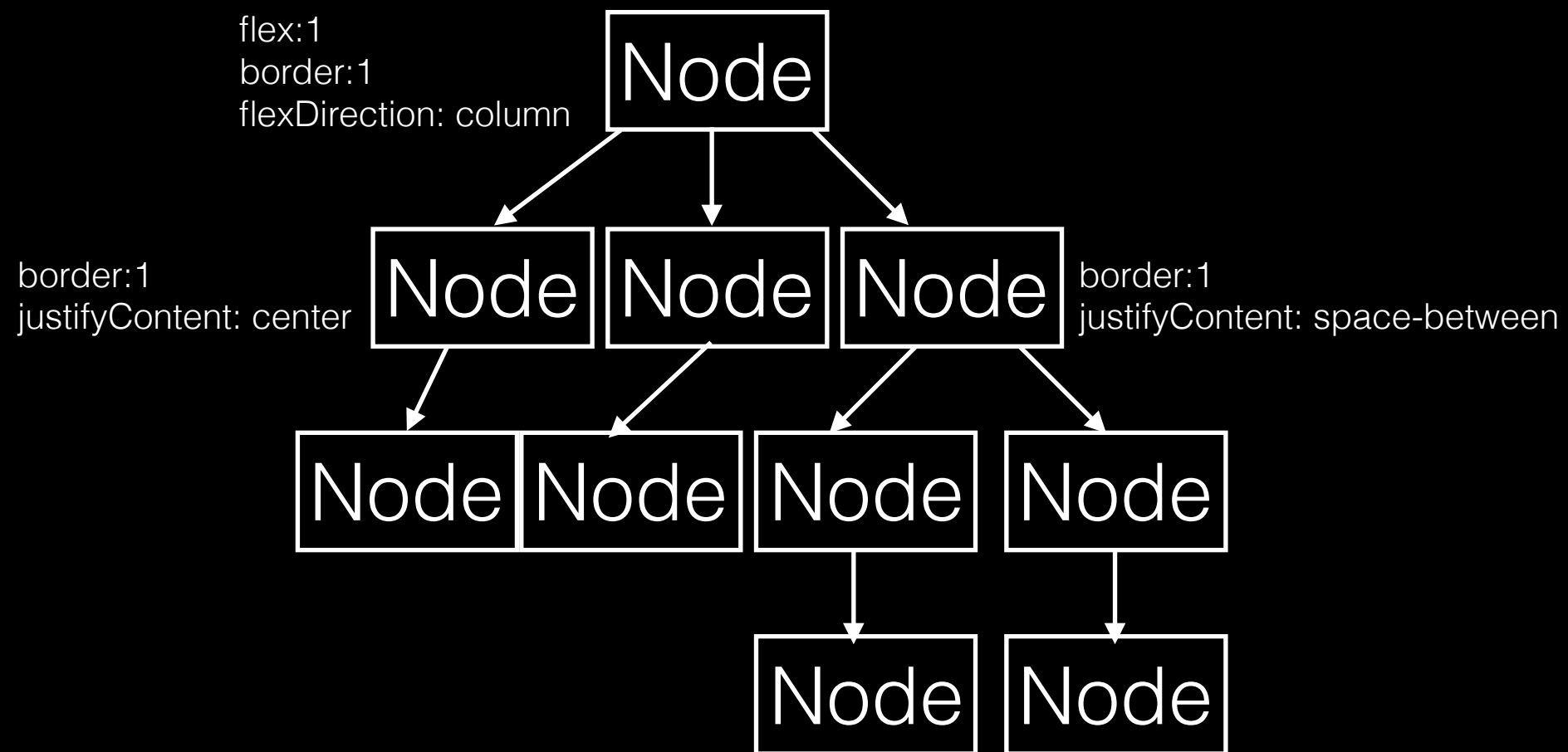

CSS Layout

CssNode



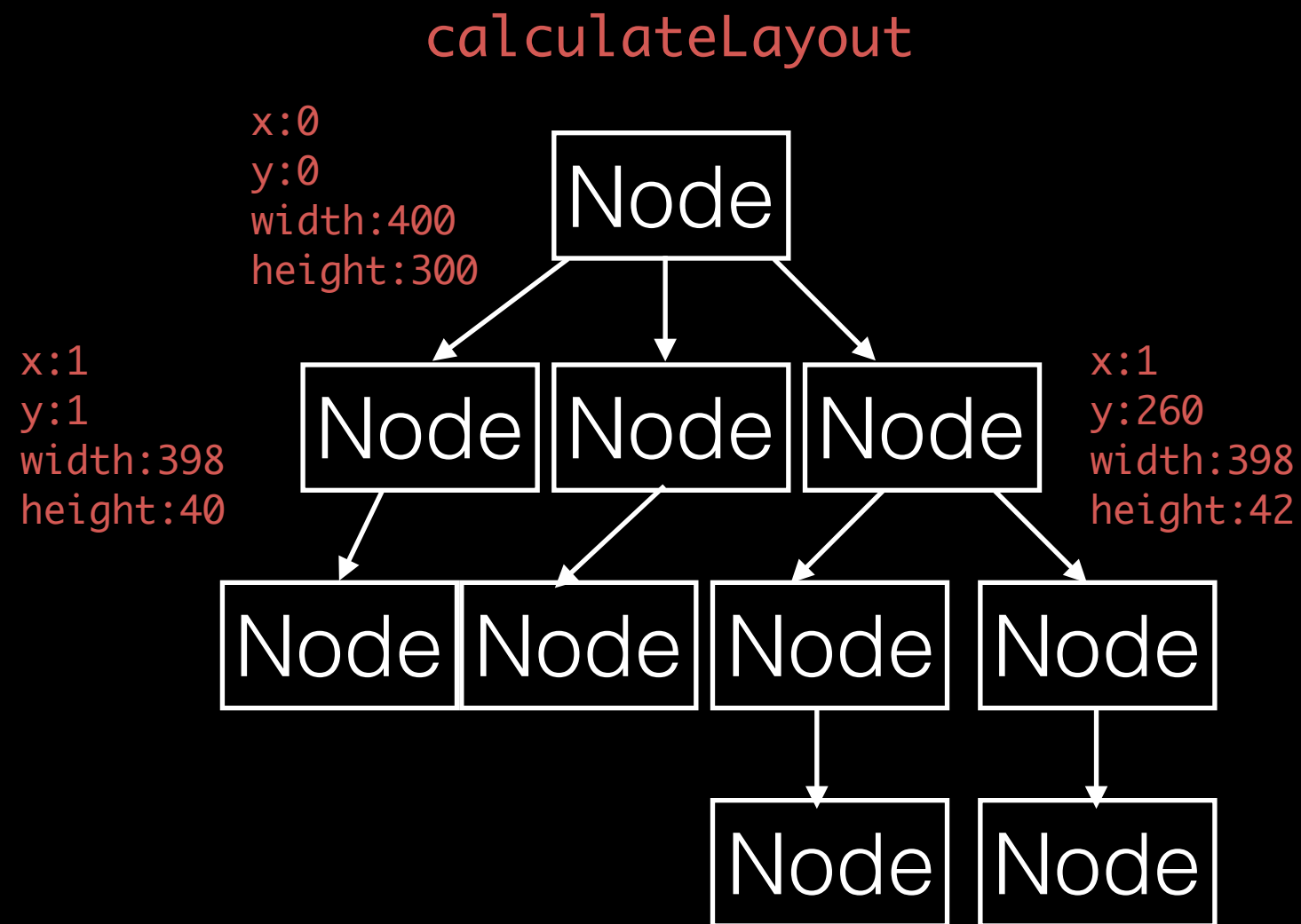
CSS Layout

CssNode



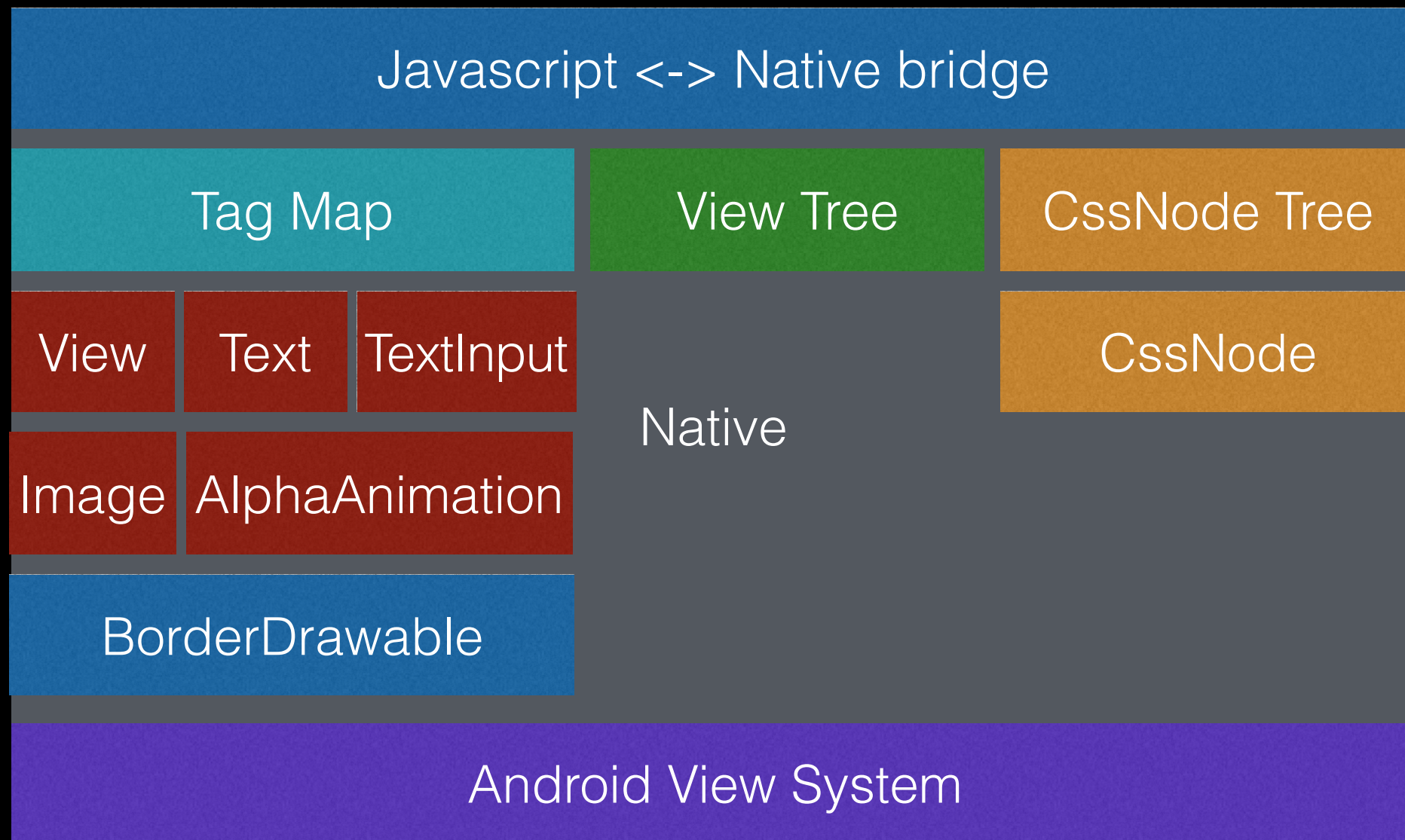
CSS Layout

CssNode



<https://github.com/facebook/css-layout>

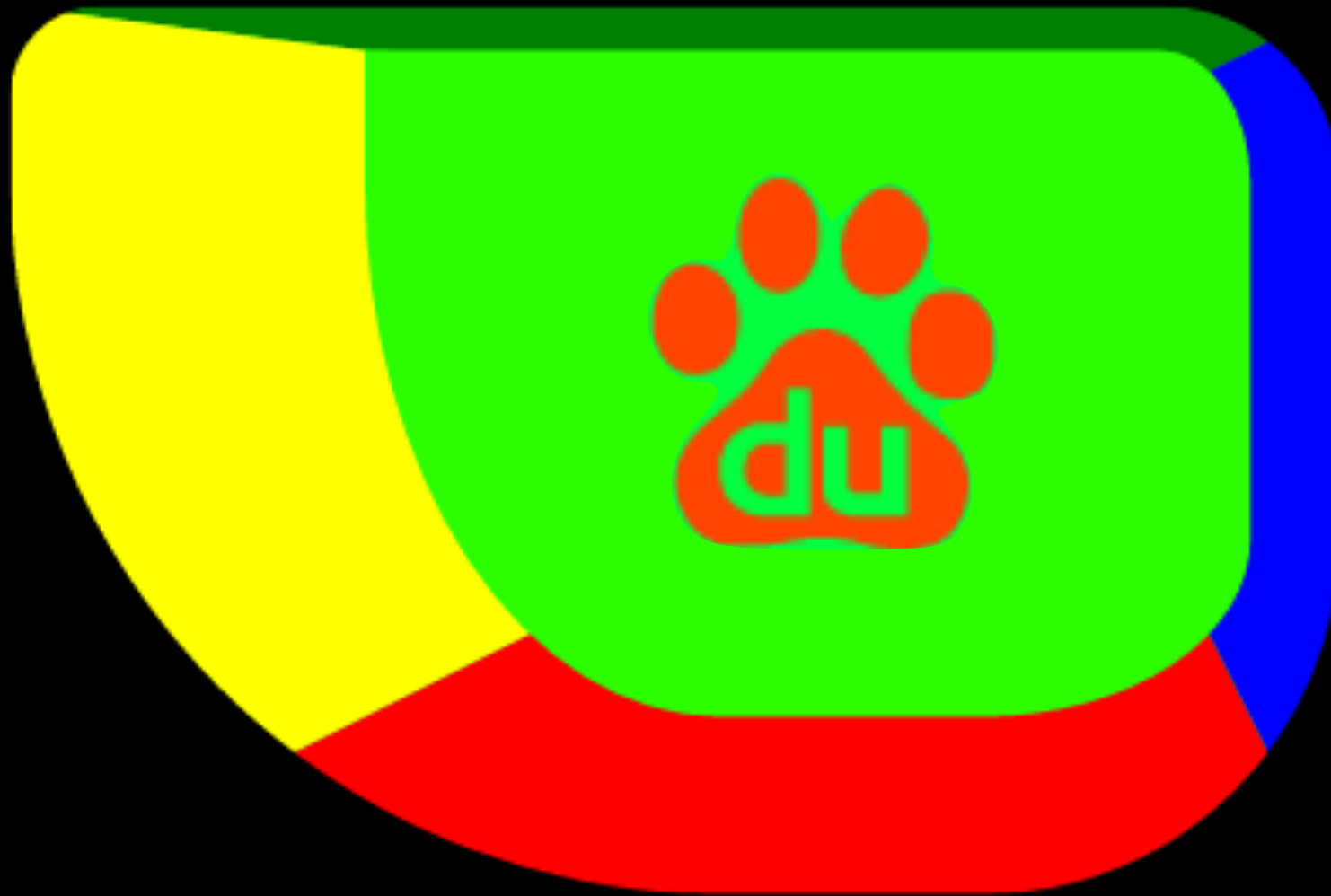
Blend UI 系统架构



Border Drawable

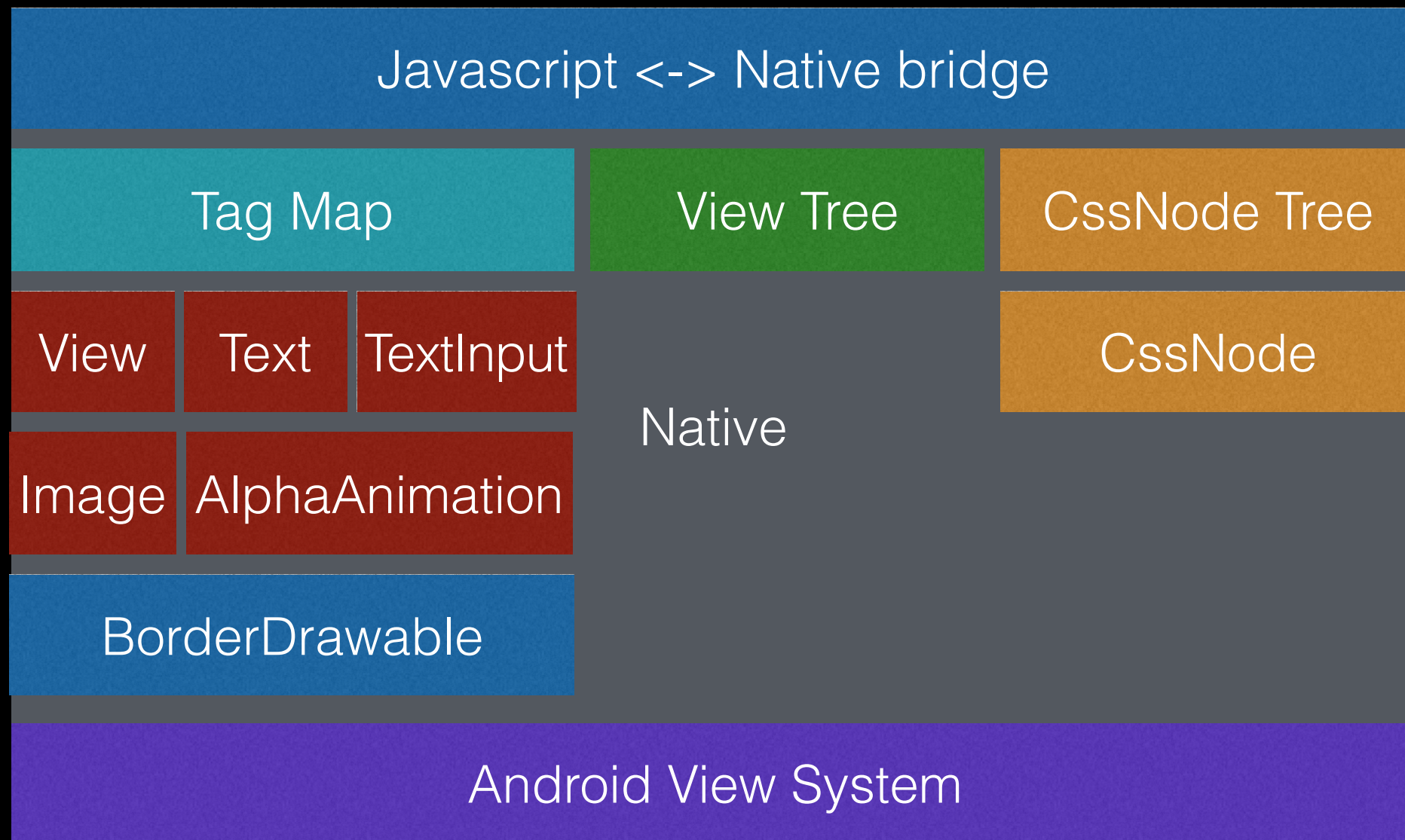


Border Drawable

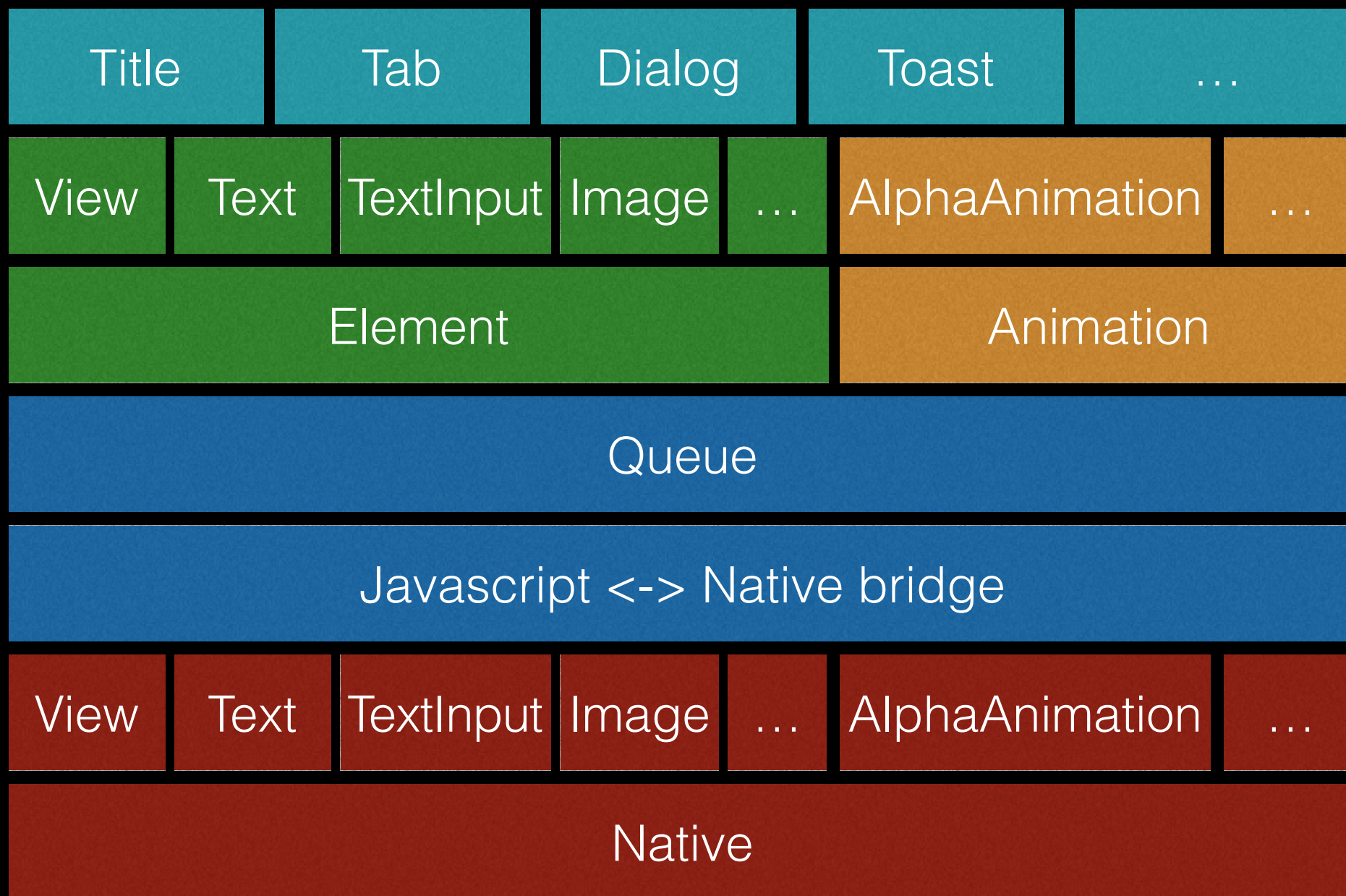


<https://github.com/zhangyuanwei/BorderDrawable>

Blend UI 系统架构



Blend UI 系统架构



事件机制

Javascript <-> Native bridge

```
webView.loadURL("javascript:XXXX");

(function(){
    var event = document.createEvent("Event");

    // 事件固定名称为 "boost" ,方便 Js 统一监听
    event.initEvent("boost", false, false);

    // 事件数据
    event.data = someData;

    // 事件源的 Tag
    event.origin = "originTag";

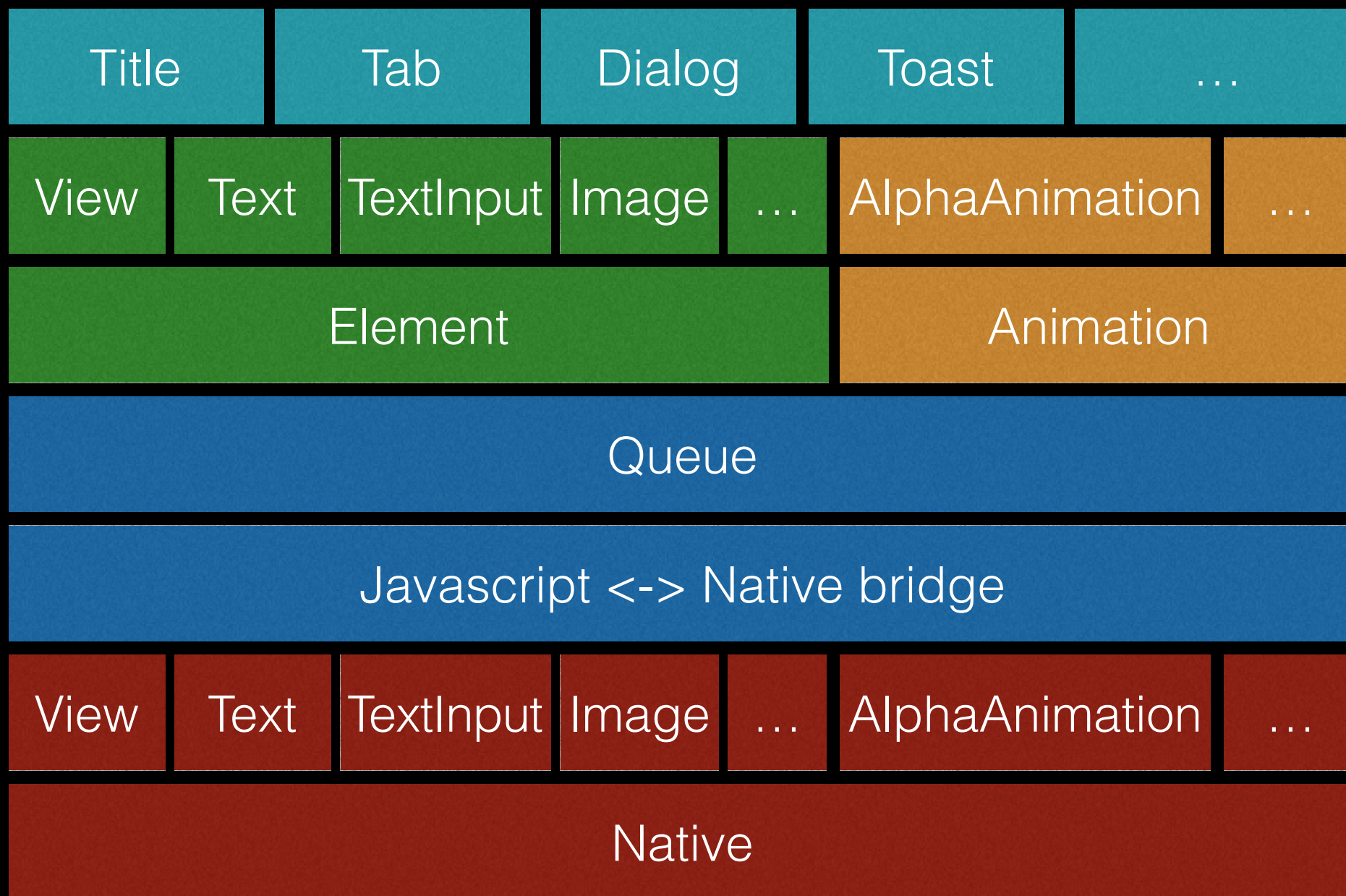
    // 具体的事件类型
    event.boostEventType = "touchend";
    document.dispatchEvent(event);
})();
```


事件机制

Element

```
document.addEventListener("boost", function (e) {  
  
    var target = getElementByTag(e.origin);  
    if (!target) {  
        return;  
    }  
    assert(target instanceof Element, target);  
  
    var event = {  
        type: e.boostEventType,  
        target: target,  
        data: e.data,  
    };  
  
    target.dispatchEvent(event);  
}, false);
```

Blend UI 系统架构



性能优化

Queue

```
var view = boost.createElement("VIEW");

view.style.position = "absolute";
view.style.top = 100;
view.style.left = 100;
view.style.width = 200;
view.style.height = 200;
view.style.backgroundColor = "#ff6600";

var text = boost.createElement("TEXT");

text.style.color = "#000000";
text.style.fontSize = 20;
text.value = "hello world";

view.appendChild(text);
boost.rootElement.appendChild(view);
```

性能优化

Queue

```
var view = boost.createElement("VIEW"); // createView

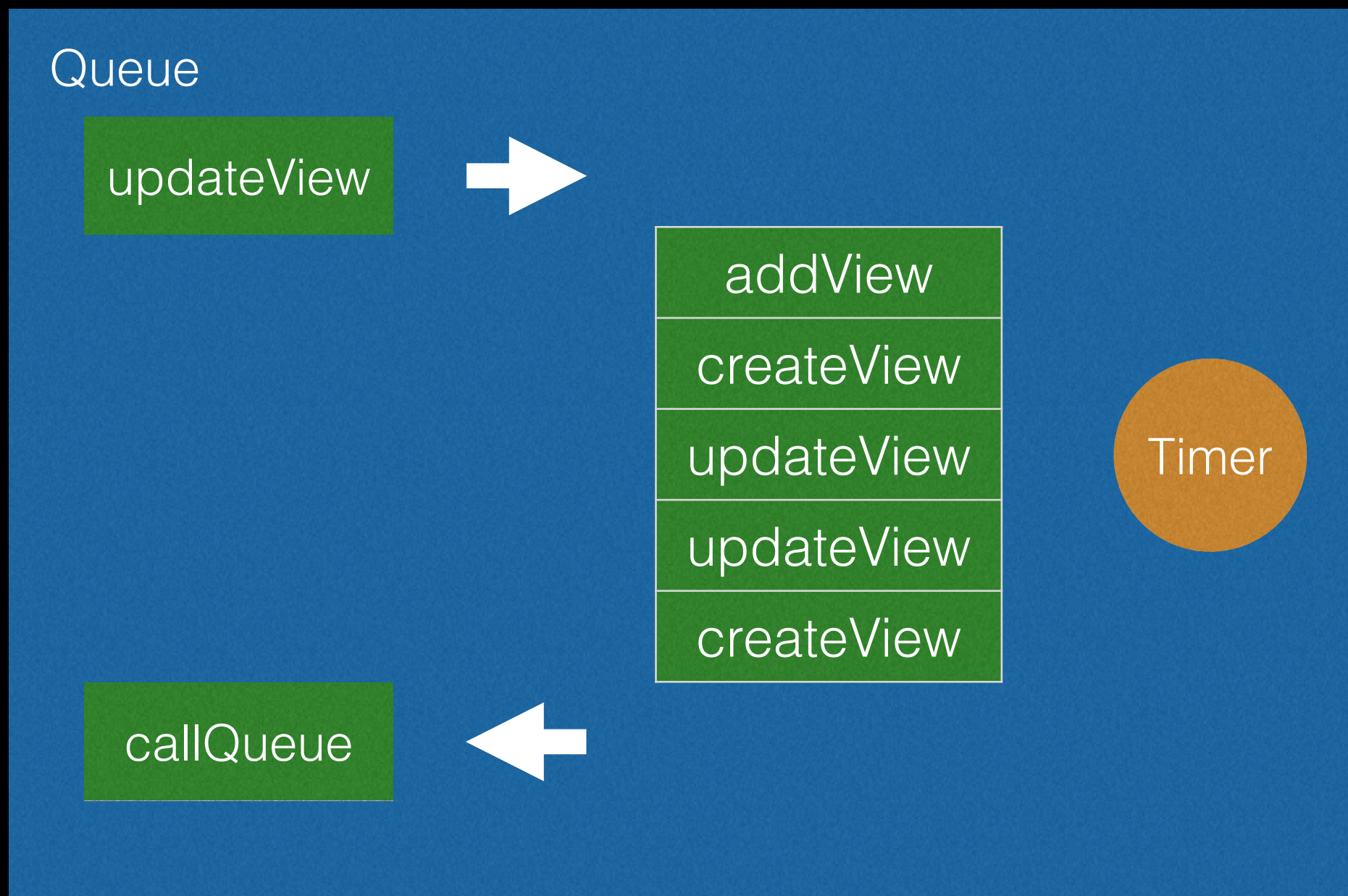
view.style.position = "absolute";      // updateView
view.style.top = 100;                   // updateView
view.style.left = 100;                  // updateView
view.style.width = 200;                 // updateView
view.style.height = 200;                // updateView
view.style.backgroundColor = "#ff6600"; // updateView

var text = boost.createElement("TEXT"); // createView

text.style.color = "#000000";           // updateView
text.style.fontSize = 20;                // updateView
text.value = "hello world";              // updateView

view.appendChild(text);                  // addView
boost.rootElement.appendChild(view);     // addView
```


性能优化

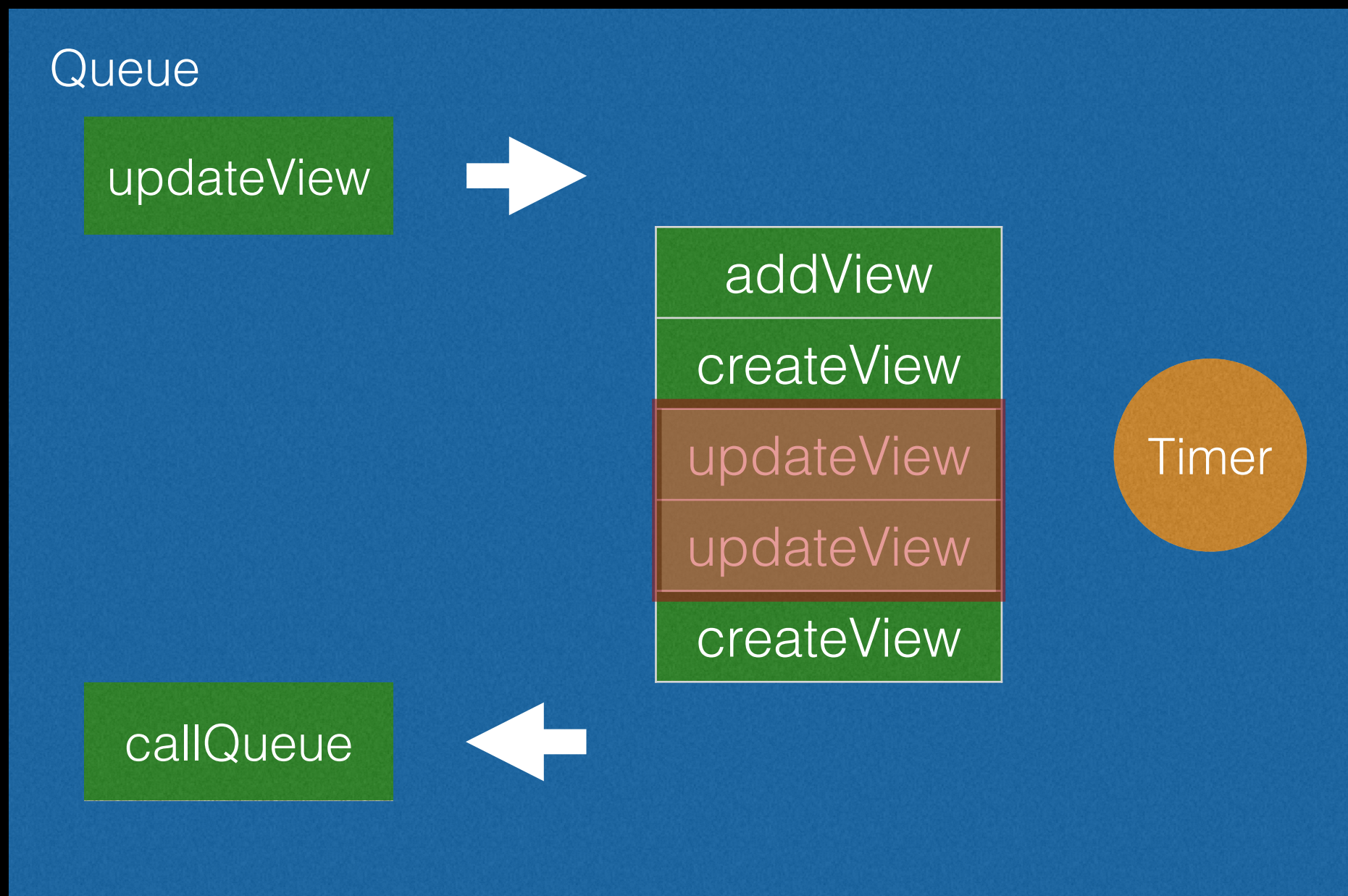


性能优化

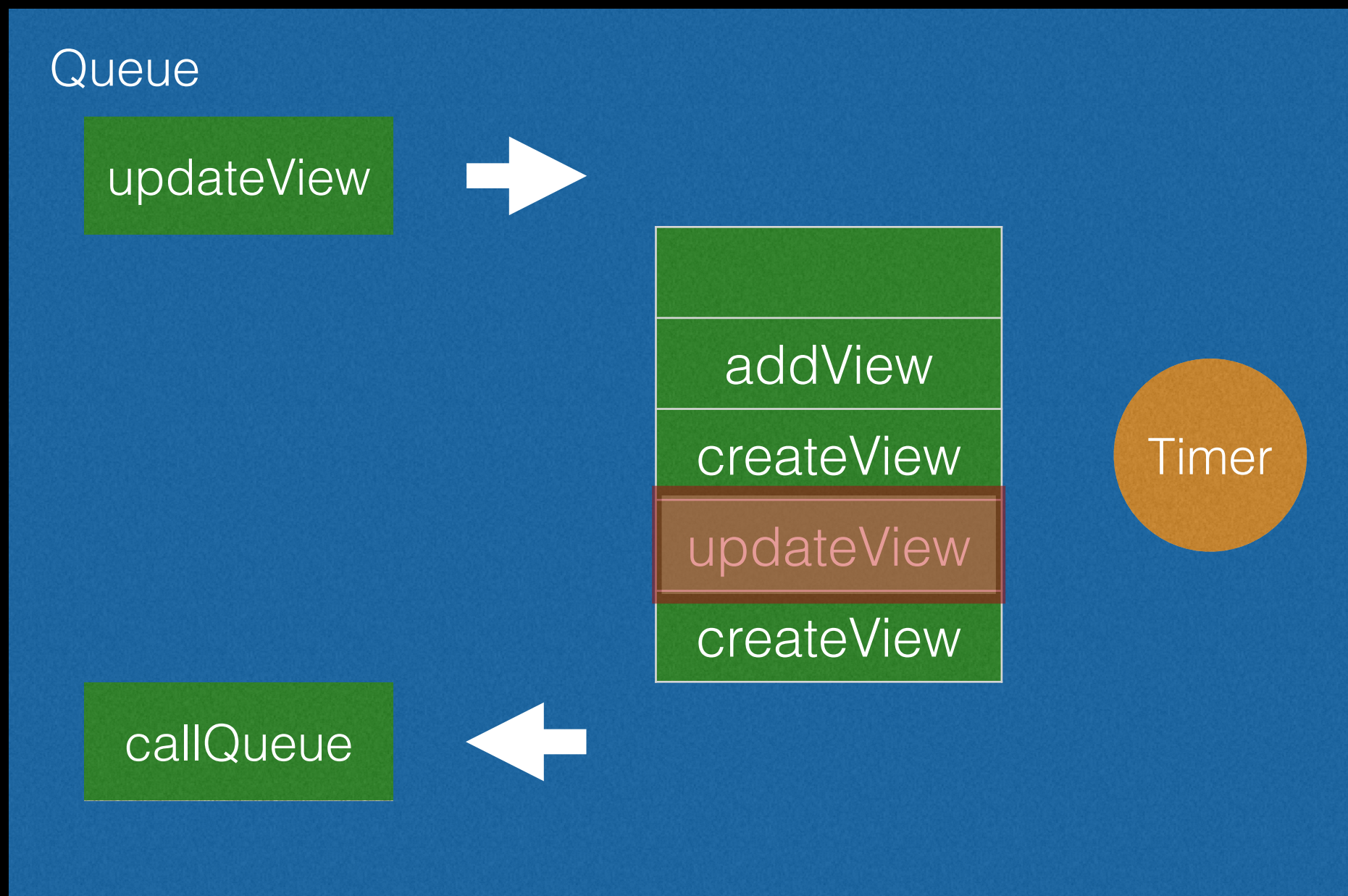
Queue

```
function queue(callback) {  
  var list = [];  
  var running = false;  
  
  function flush() {  
    running = false;  
    callback(list);  
    list = [];  
  }  
  
  return function (cmd) {  
    list.push(cmd);  
    if (!running) {  
      setTimeout(flush, 0);  
      running = true;  
    }  
  };  
}
```

性能优化



性能优化



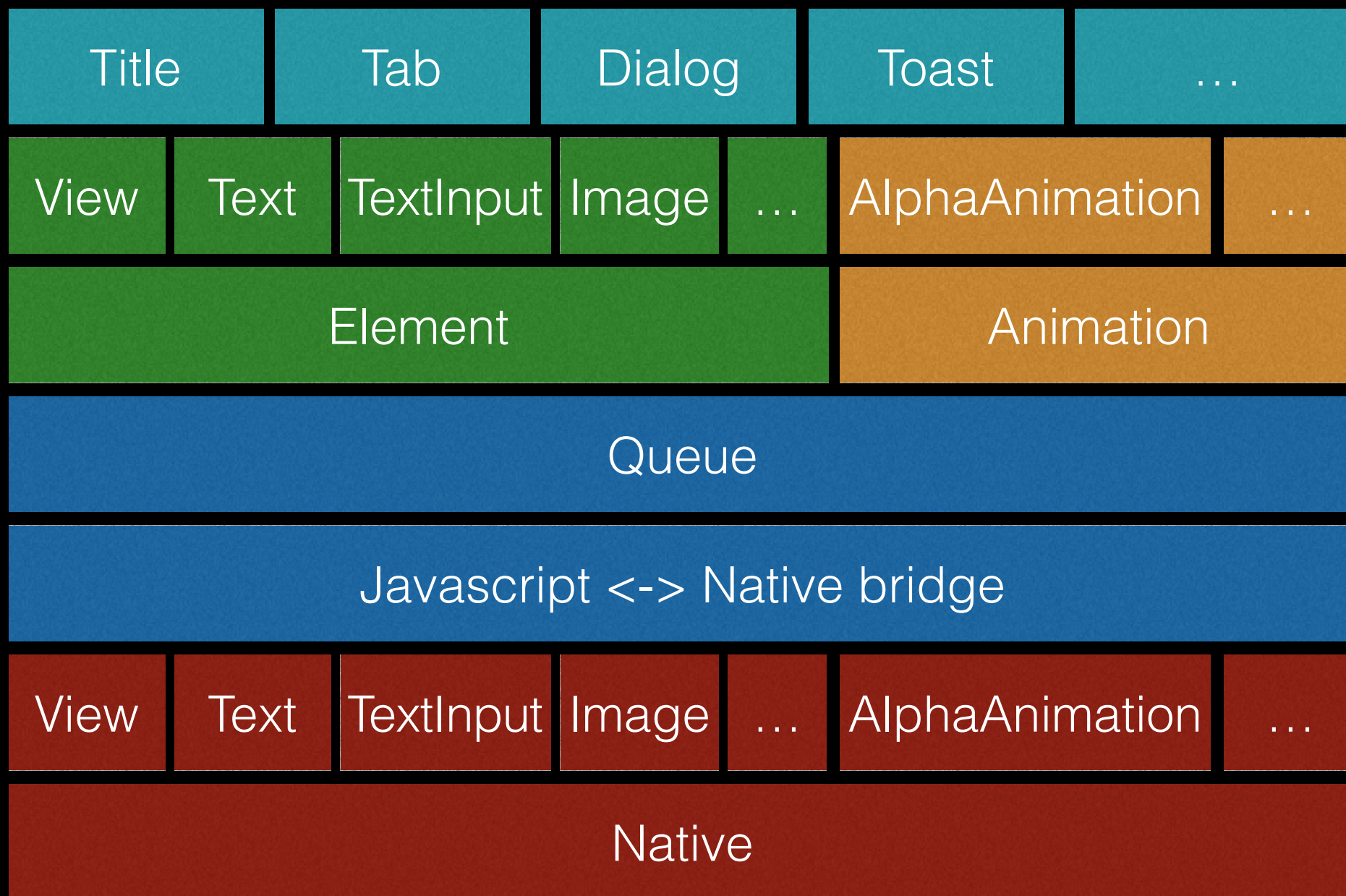
性能优化

Queue

```
window.lc_bridge.callQueue("commands": jsonString);
```

```
[  
  // 第一个命令  
  {  
    "tag": string, // 需要调用函数的目标 Tag  
    "method": string, // 函数名  
    "args": [ // 参数列表  
      object, // 参数 1  
      object, // 参数 2  
      ...  
    ],  
  },  
  
  // 第二个命令  
  { ... },  
  
  // ...  
]
```

Blend UI 系统架构



API 设计

```
var view = boost.createElement("VIEW");

view.style.position = "absolute";
view.style.top = 100;
view.style.left = 100;
view.style.width = 200;
view.style.height = 200;
view.style.backgroundColor = "#ff6600";

boost.rootElement.appendChild(view);

var text = boost.createElement("TEXT");

text.style.color = "#000000";
text.style.fontSize = 20;
text.value = "hello world";

view.appendChild(text);
```

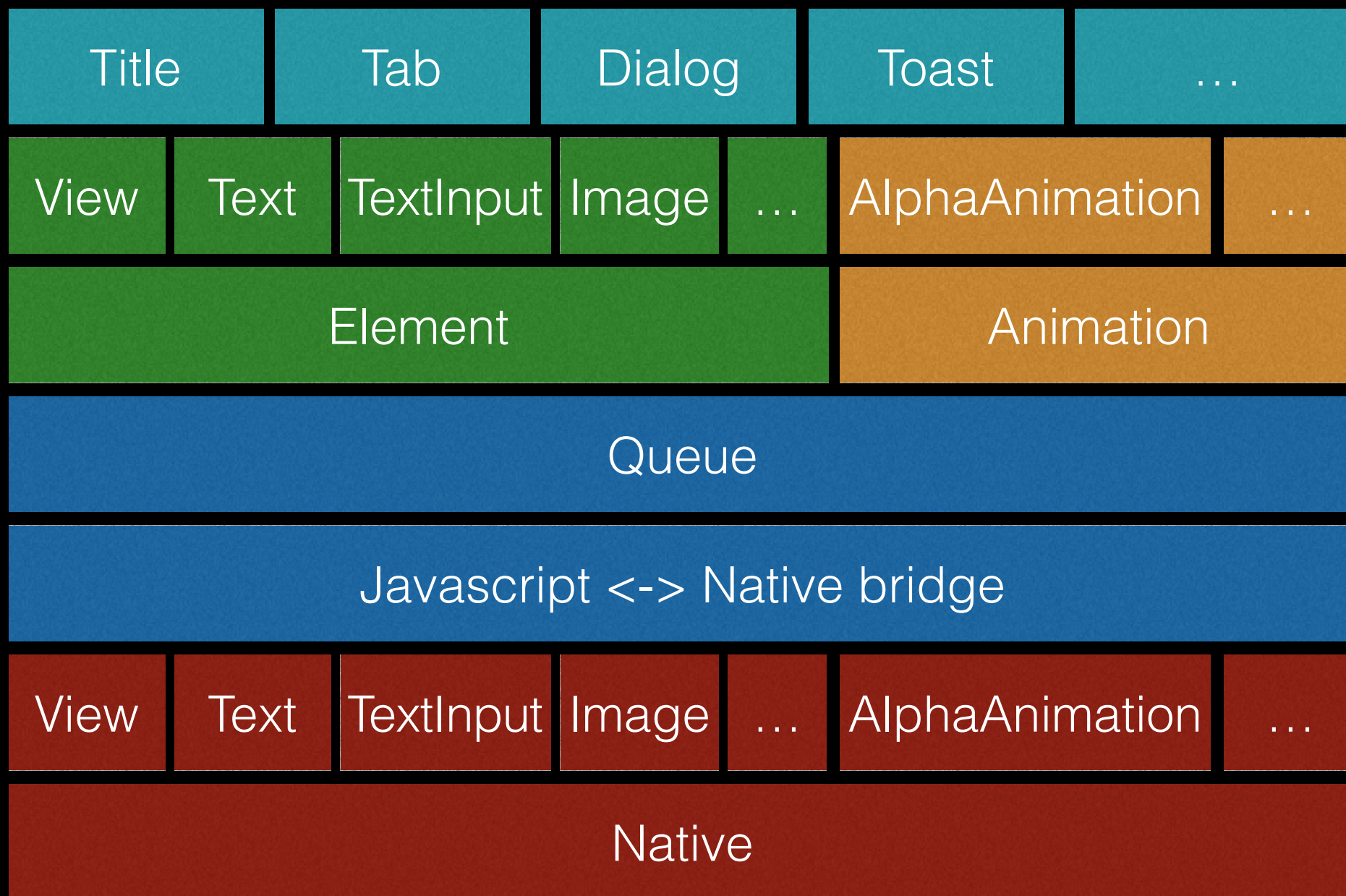
API 设计

```
view.addEventListener("touchend",  
  function (event) {  
    text.value = "doun't touch me!";  
    view.style.top = Math.random() * 400;  
    view.style.left = Math.random() * 400;  
  },  
  false);
```

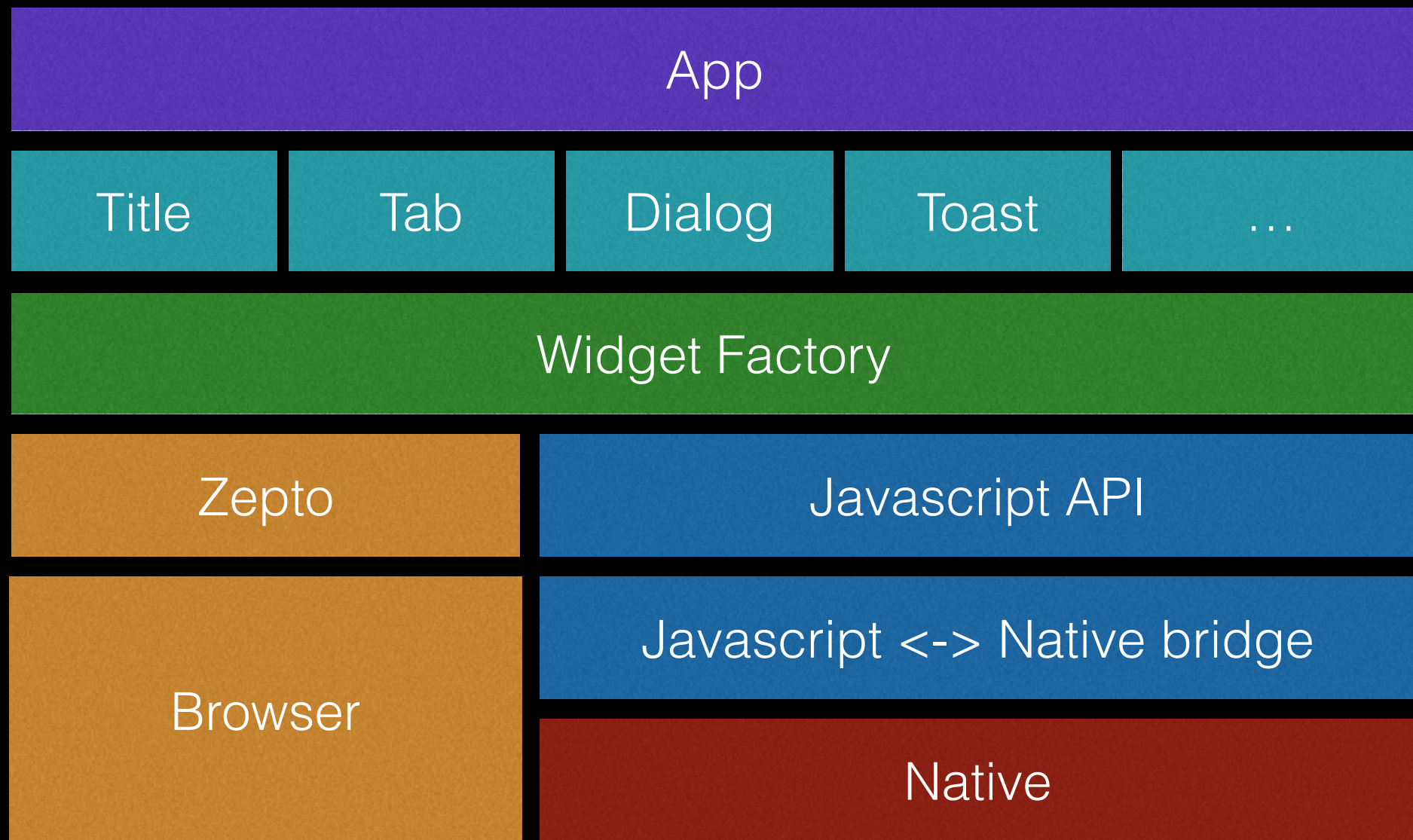

API 设计

```
<?xml version='1.0' encoding='UTF-8' ?>
<cart>
  <style>
    .cart-describe {
      flex-direction : row; /* 横向排列 */
      align-items: center; /* 居中 */
      padding : 10;
      background-color:green;
    }
  </style>
  <View class="wrapper">
    <!-- 内容需要用 ScrollView 包装起来, 不然超出屏幕了没法滚动 -->
    <ScrollView style="flex:1">
      <!-- ScrollView 里面不能直接放多个子节点, 需要用一个 View 包起来-->
      <View class="cart-wrapper">
        <!-- 购物车标题 -->
        <View class="cart-title">
          <Text>我是一个购物车</Text>
        </View>
        <!-- 购物车主体 -->
        <View class="cart-content">
          <!-- 一个购物车条目 -->
          <View class="cart-item">
            <View class="cart-del">
```

Blend UI 系统架构



Blend UI 系统架构





<https://github.com/zhangyuanwei>

Thanks