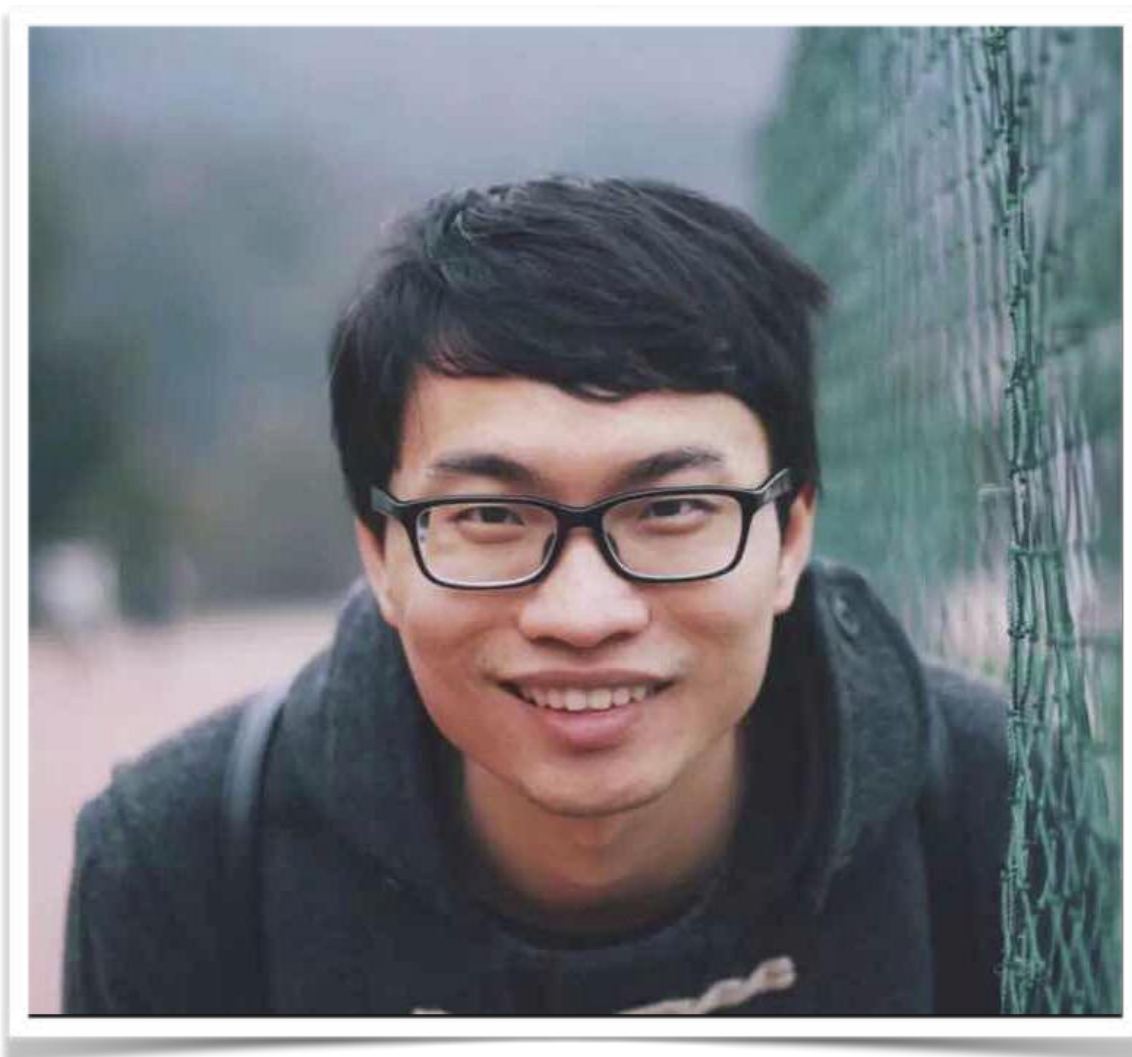


使用 React 构建 Flutter 应用

探索新一代渲染技术 Kraken



元彦

<https://github.com/yuanyan>

 淘系前端团队

- 2011 – 2013年

- WebQQ
- QQ
- 手机 QQ

- 2014年 – 现在

- 淘宝交易线产品
- 淘宝中后台产品
- 淘系无线架构、中后台架构



天猫未来店

TMALL
FUTURE

数字化门店智能
Intelligent solution for you

入口

ENTRANCE



天猫未来店 体验升级

9月19日 - 9月20日

09:00上午 - 17:30



仅支持
支付宝结算



须持有
淘宝/天猫账号

WELCOME WELCOME





GCanvas JS API



GCanvas C++ Engine

OpenGL ES / Vulkan

GPU



Kreken 原型技术原理

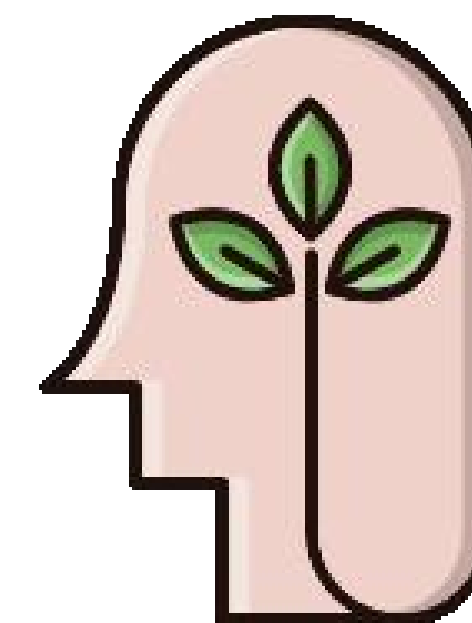
Kraken 原型版本**优劣**



简单直接

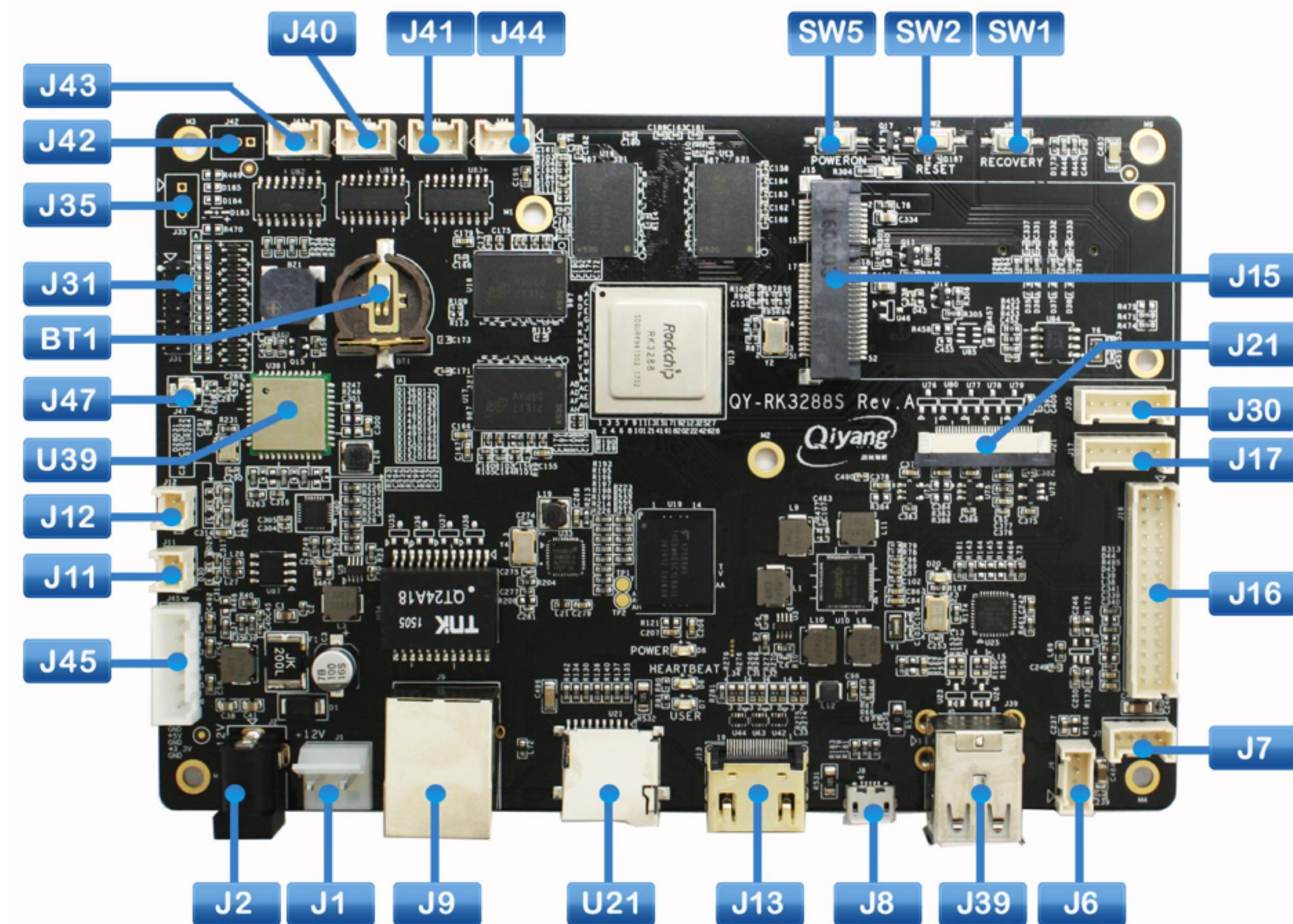
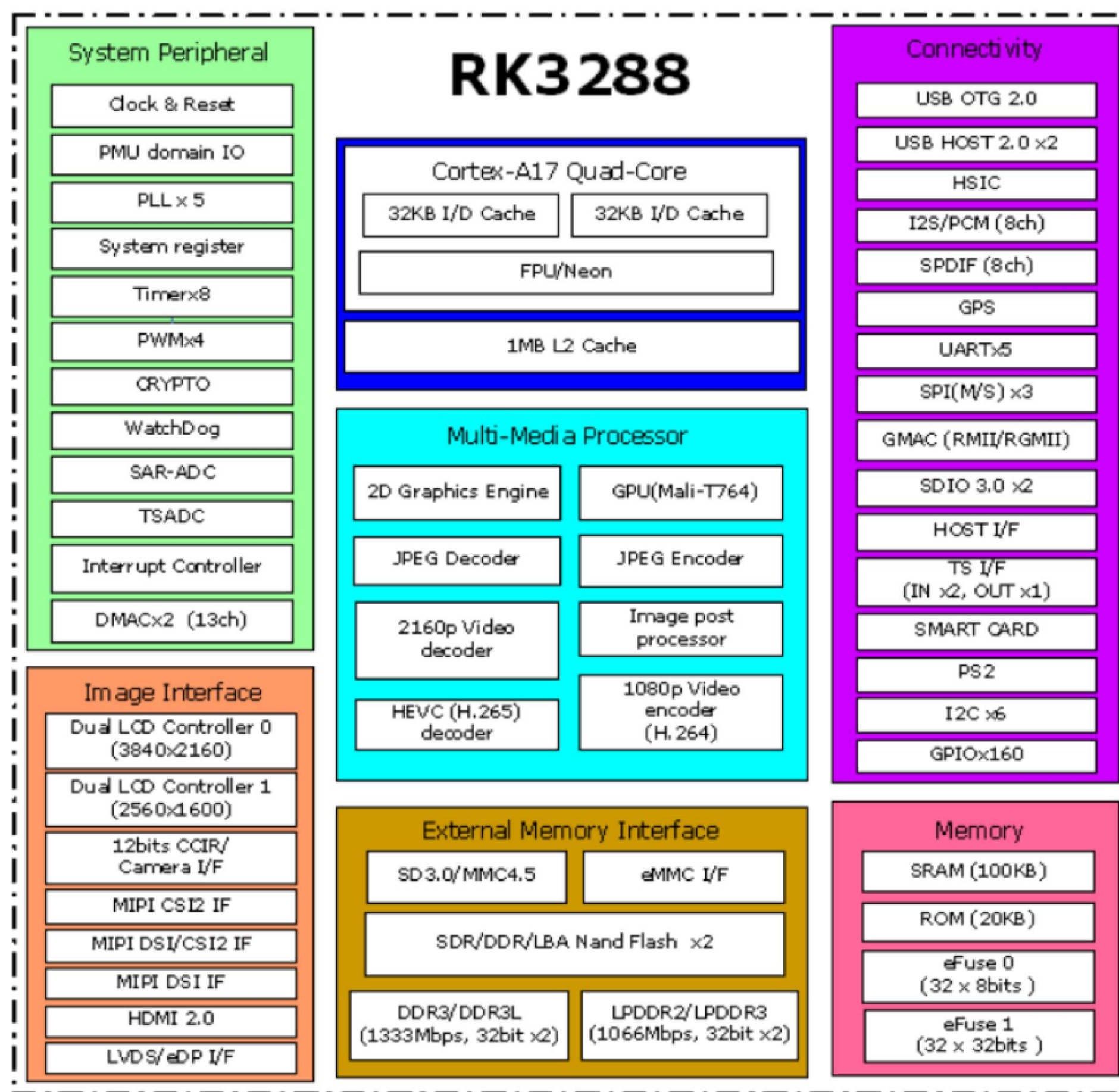


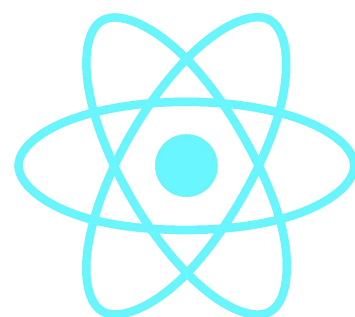
研发效率低



适用场景有限

嵌入式主板





Java / Kotlin

React.js

Rax.js / Vue.js

Dart

JS

Layout XML

VDOM

VDOM

Widget Tree

DOM Tree

Native View Tree (OEM)

RenderObject Tree

Layout Tree

Layer Tree

Layer Tree

Layer Tree

Skia

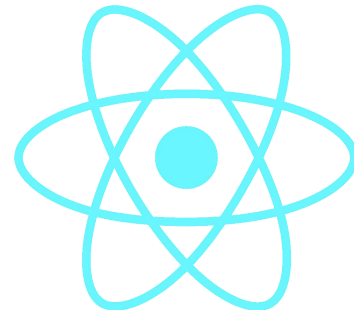
OpenGL ES / Vulkan

GPU

渲染技术探索



(14)



React Native



(10)



(9)



(13)

Dart

JavaScript

JavaScript





JavaScript

JIT (Develop) / AOT (Deploy)

JIT

JIT

JIT

WORE    
(Write Once Run Everywhere)

LORE  
(Learn One Run Everywhere)

WORE   

WORE     

Good Performance   

Good Performance  

Good Performance   

It' s OK  

Good Ecosystem    

Good Ecosystem    

It' s OK   

Amazing Ecosystem     

Built-in Features   

Built-in Features  

Built-in Features 

Built-in Features  

为什么 ❤️ Flutter



Web ?



Flutter

Flutter 是 Web 之外的另一个真跨端渲染技术

前端视角里的 Flutter

通过 platform channels 调用平台能力

支持 Hot Reload

比肩原生的性能

调试工具完善



不支持 Flex Layout 布局

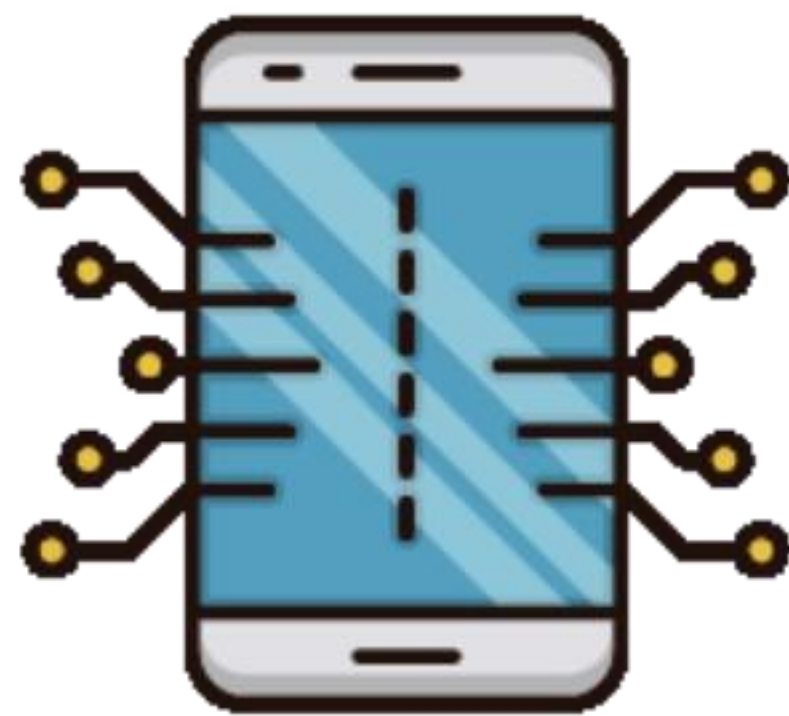
不支持类 JSX，可读性差

不支持动态下发 Bundle

脱离前端 JS 生态



使用 Flutter 的刚性诉求



具有动态性



可连接前端生态

Dart 生产模式

开发 Develop

JIT

Dart VM

部署 Deploy

AOT

Dart AOT Runtime

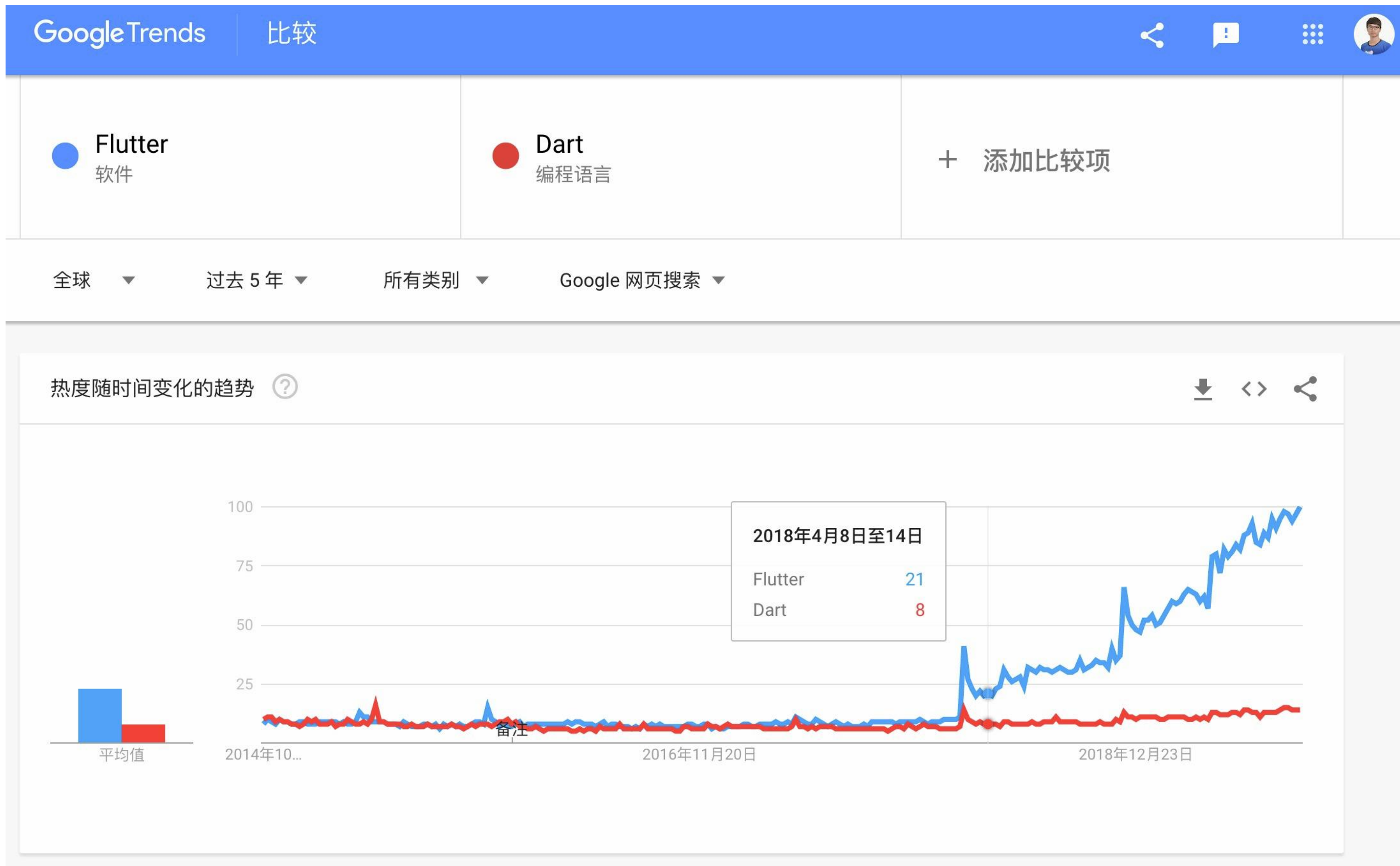


Jeff Atwood

Co-founder of StackOverflow

“

Any application that can be written in
JavaScript, will eventually be written
in JavaScript.



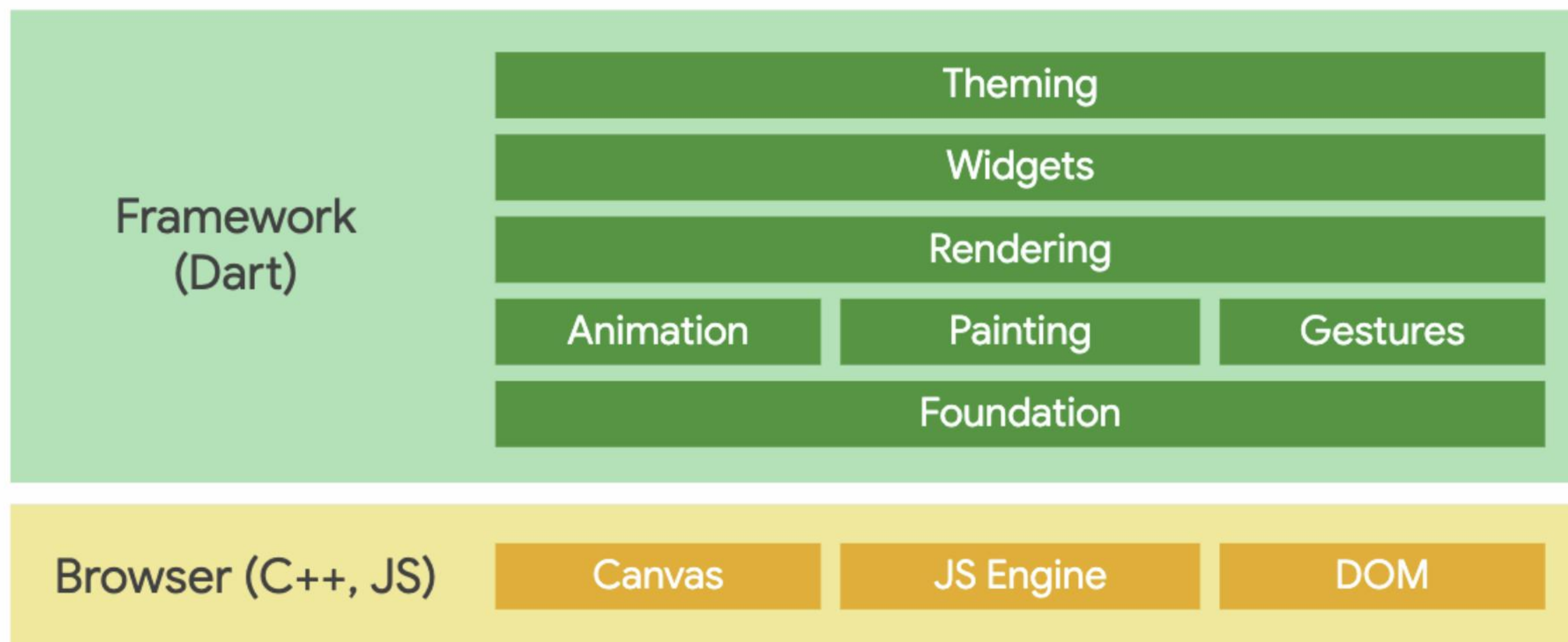
Flutter 增长迅速，但 Dart 关注平平



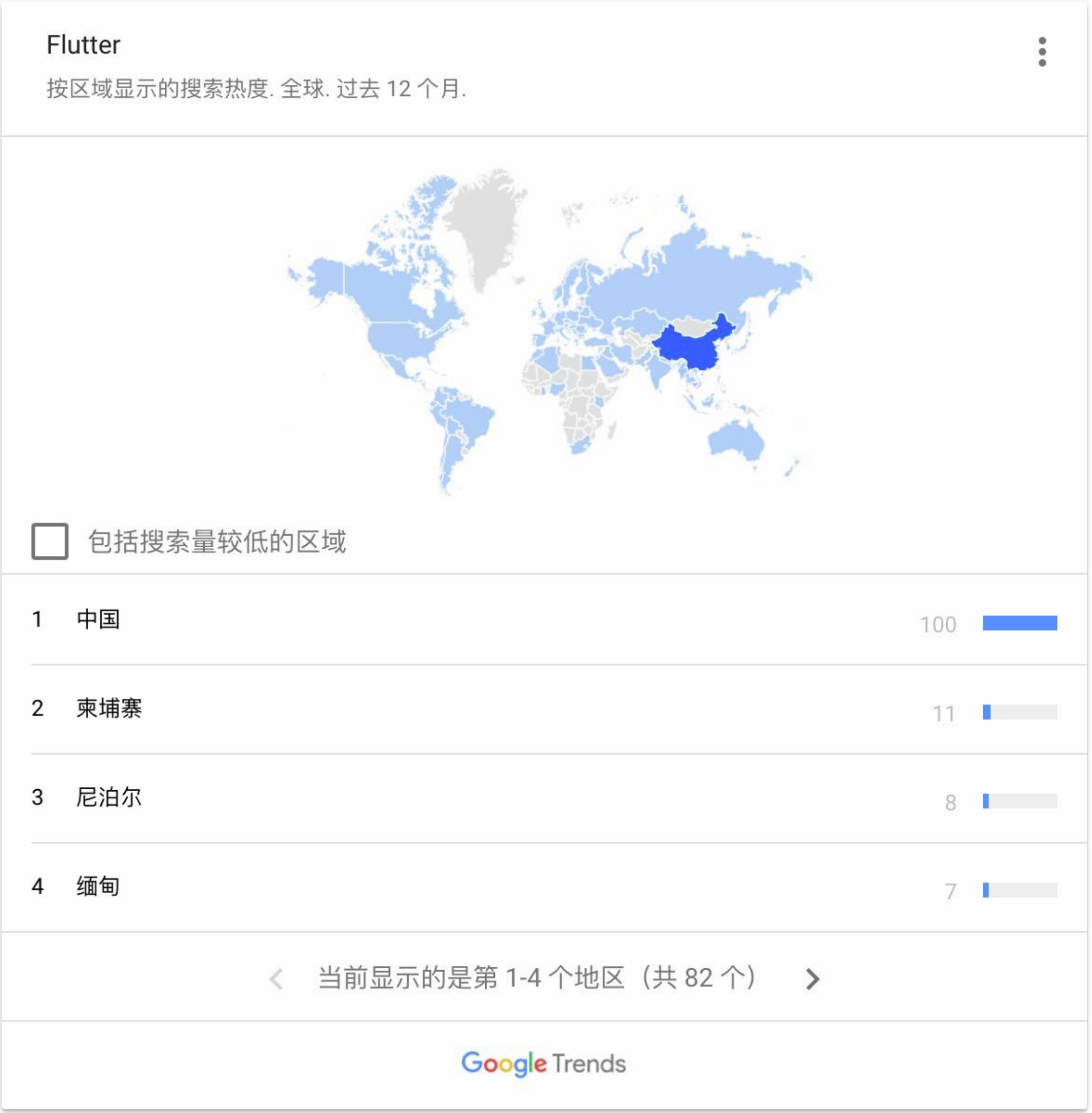
Kraken



关于 Flutter for Web



并不是完全鸡肋，**适合其只适合非 C 端业务场景**



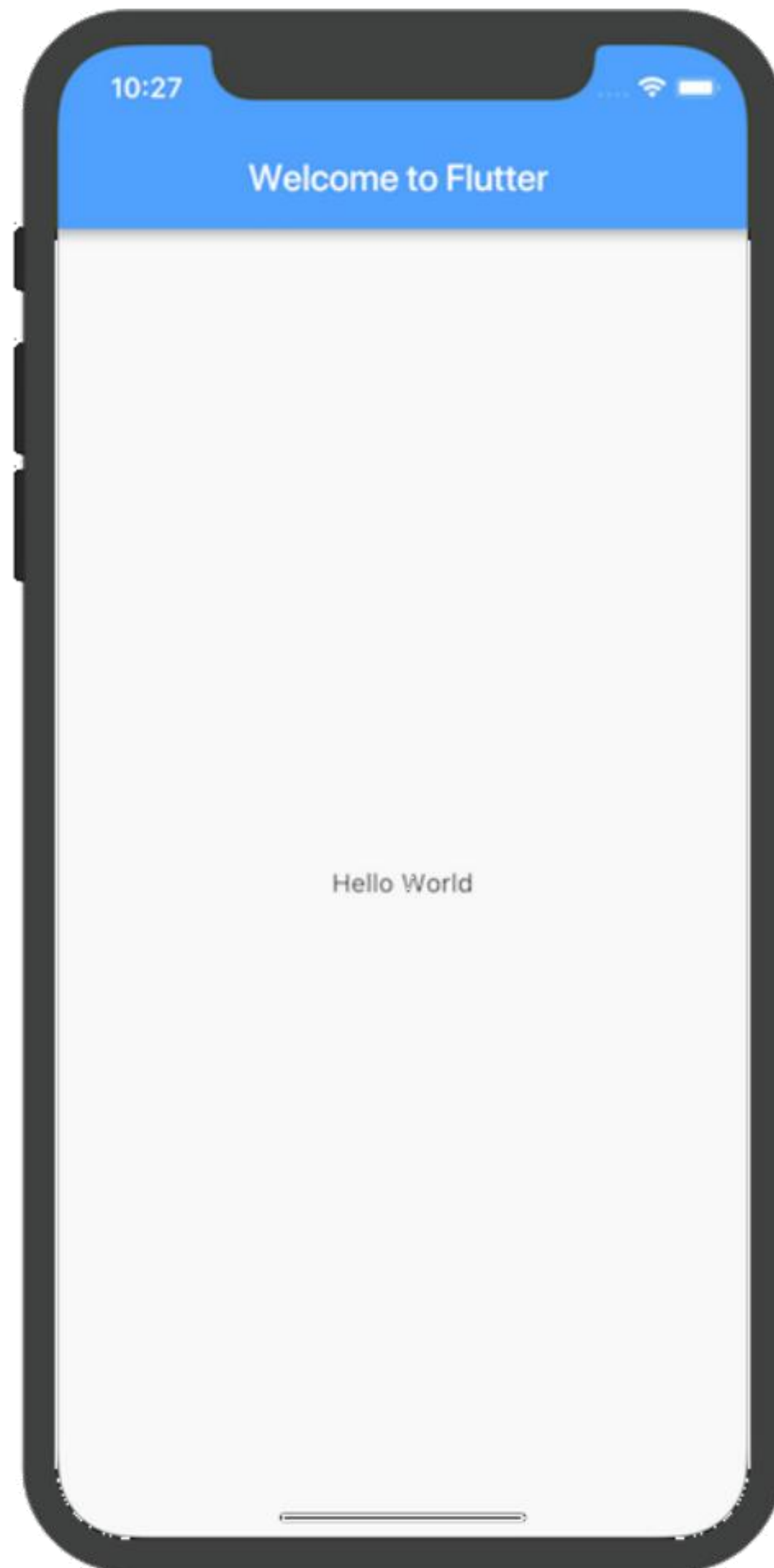
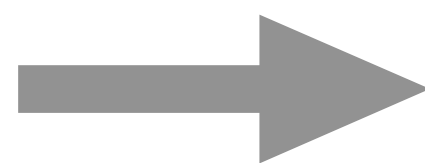
中国可能是全球最关注 Flutter 的国家



```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Welcome to Flutter',
      home: Scaffold(
        appBar: AppBar(
          title: Text('Welcome to Flutter'),
        ),
        body: Center(
          child: Text('Hello World'),
        ),
      ),
    );
  }
}
```





```
import { createElement } from 'react';  
import { render } from 'react-dom';  
  
const App = () => {  
  return (  
    <div>  
      <span>Hello World!</span>  
    </div>  
  );  
};  
  
render(<App />, document.body);
```

我们的目标：能直接运行吗？



Demo

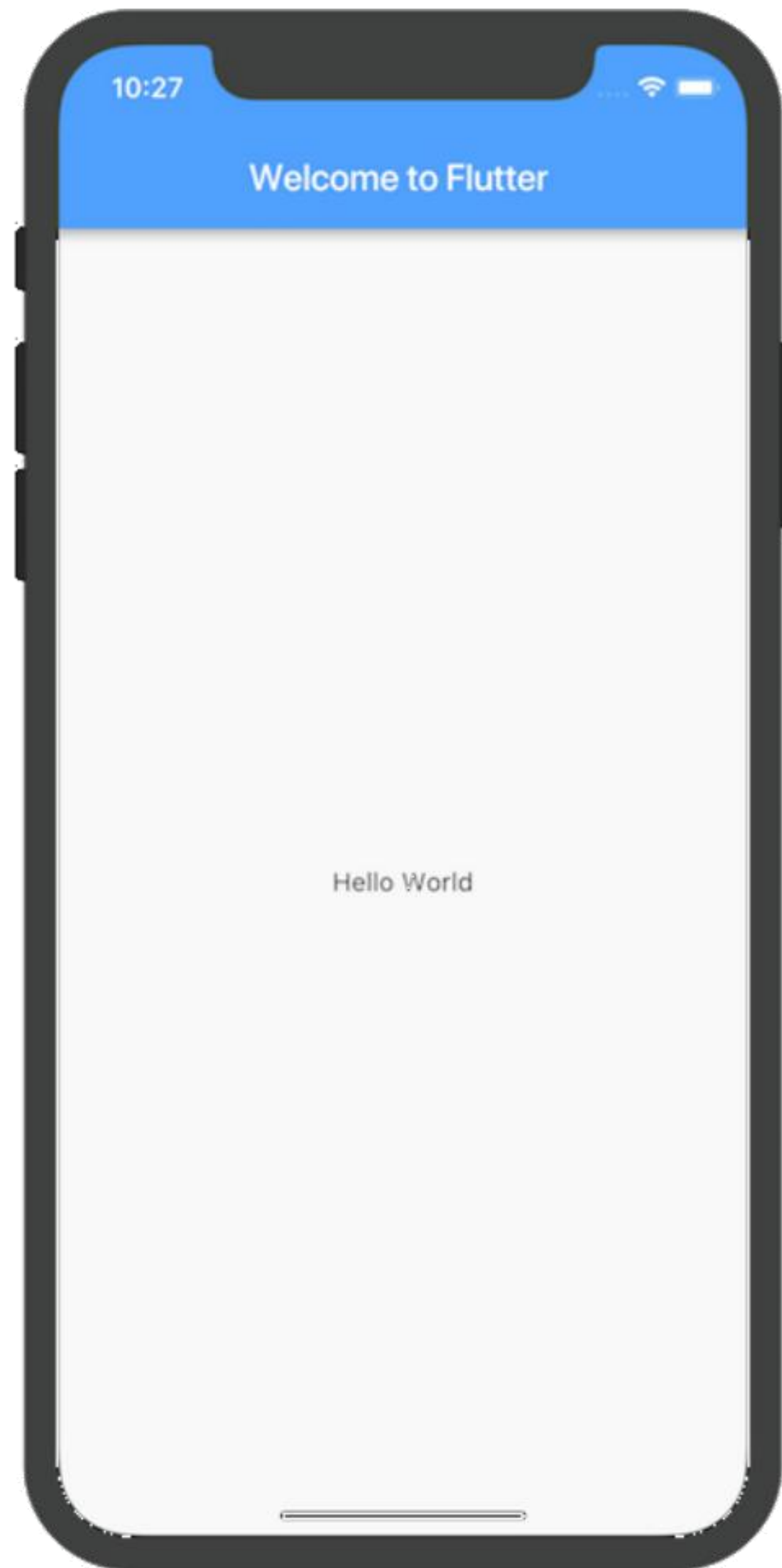
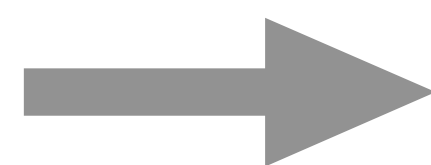




```
import { createElement, render } from 'rax';
import View from 'rax-view';
import Text from 'rax-text';
import DriverUniversal from 'driver-universal';

const App = () => {
  return (
    <View>
      <Text>Hello World!</Text>
    </View>
  );
};

render(<App />, null, { driver: DriverUniversal });
```





Yes

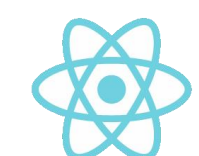
Redux
Mobx
Rxjs
Lodash
Underscore
Moment
graphql
Immutablejs
Ramda
Validator.js
...

Npm
Yarn
babel
webpack
Rollup
eslint
jest
mocha
...

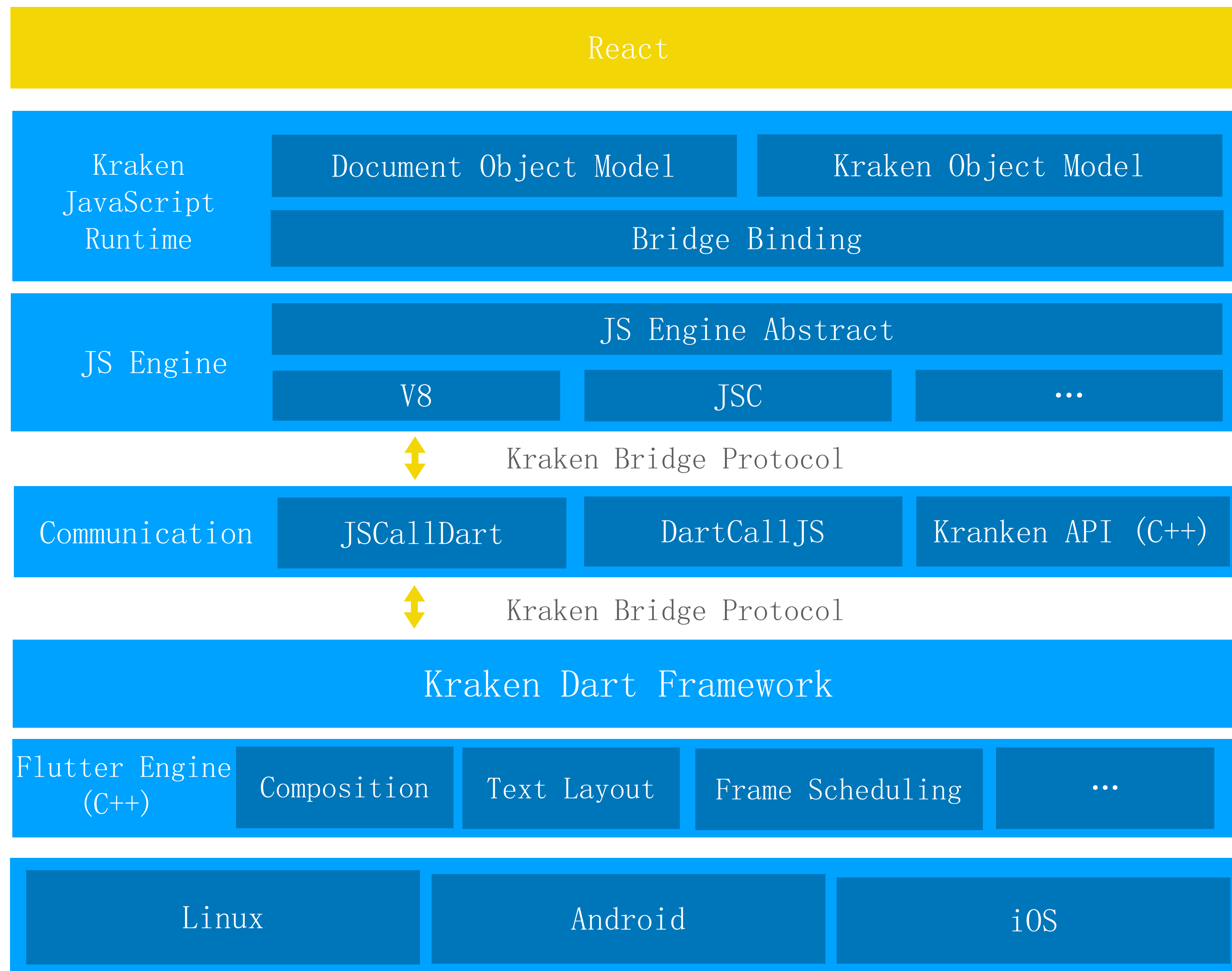
Less
Sass
Stylus
TypeScript
...

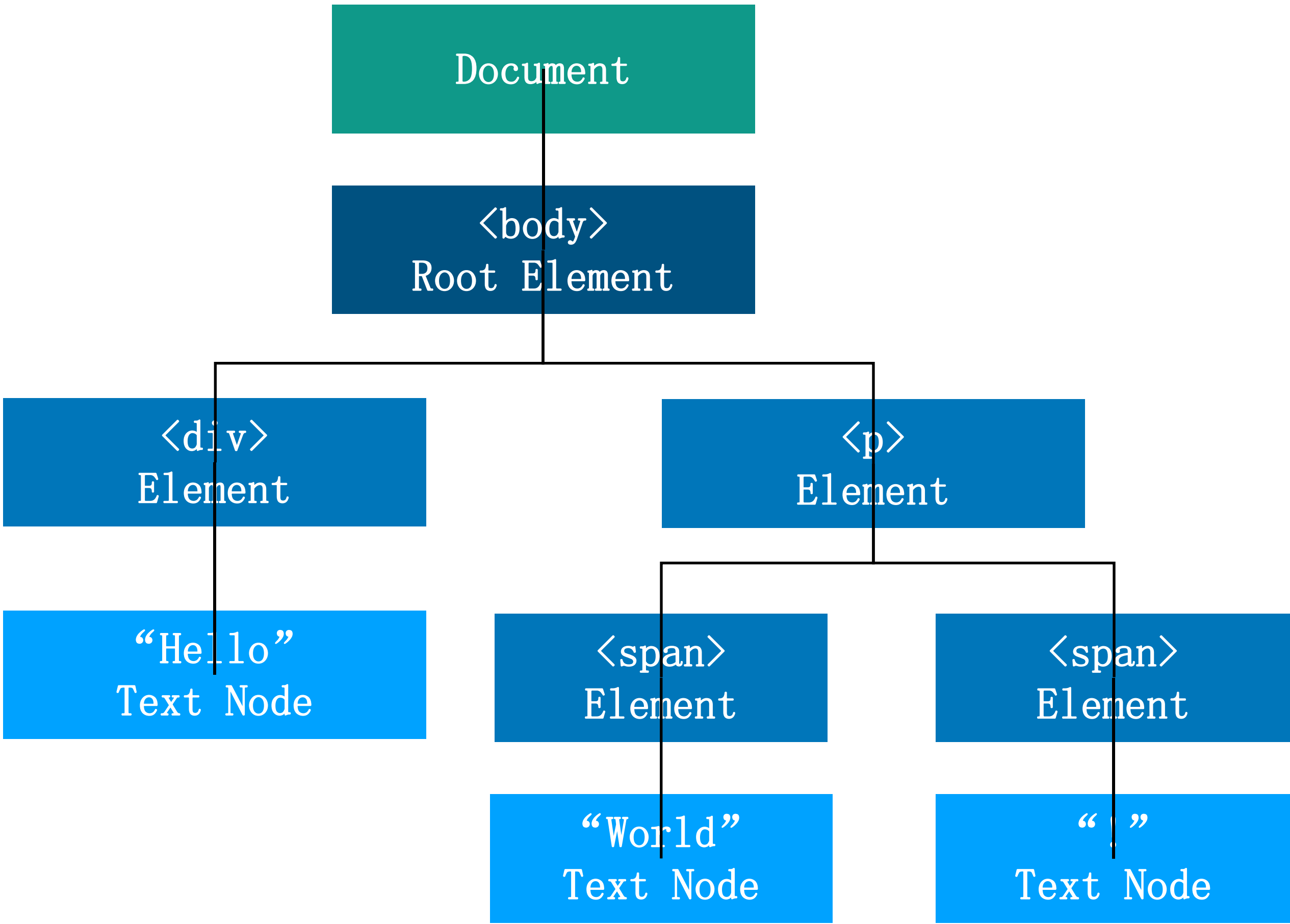


背后原理



React Frontend





← `const body = document.body;`

← `const div = document.createElement('div');
body.appendChild(div);`

← `const text = document.createTextNode('hello');
div.appendChild(text);`

Kraken Bridge Protocol

JS



`createElement('div')`



`body.appendChild(div)`



`JSON.stringify`



Kraken Bridge



`jsonDecode`

`{ id: 0 , type: ' DIV' }`

`['insertAdjacentNode' , [-1, beforeend, { id: 0 , type: ' D`

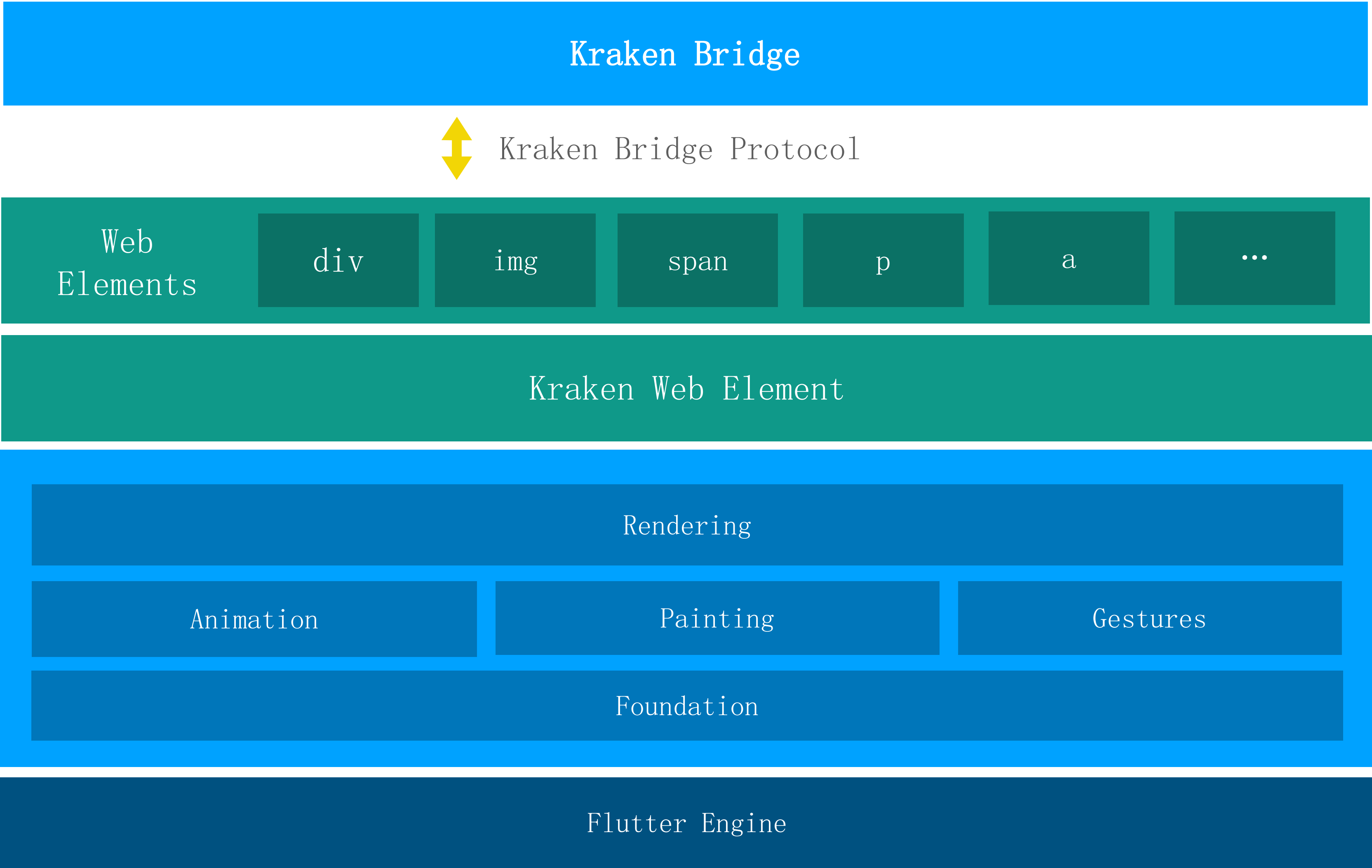
`“ ... ”`

`__kraken_js_to_dart__(“ ... ”)`

`['insertAdjacentNode' , [-1, beforeend, { id: 0 , type: ' D`

Flutter Backend Kraken Dart Framework

Kraken Dart
Framework



Kraken JavaScript Runtime

Kraken Object Model

Timer: setTimeout/clearTimeout

Interval: setInterval/clearInterval

Screen: { height, width, ...}

Location: { href, host, search, ...}

Window: { open, close, ... }

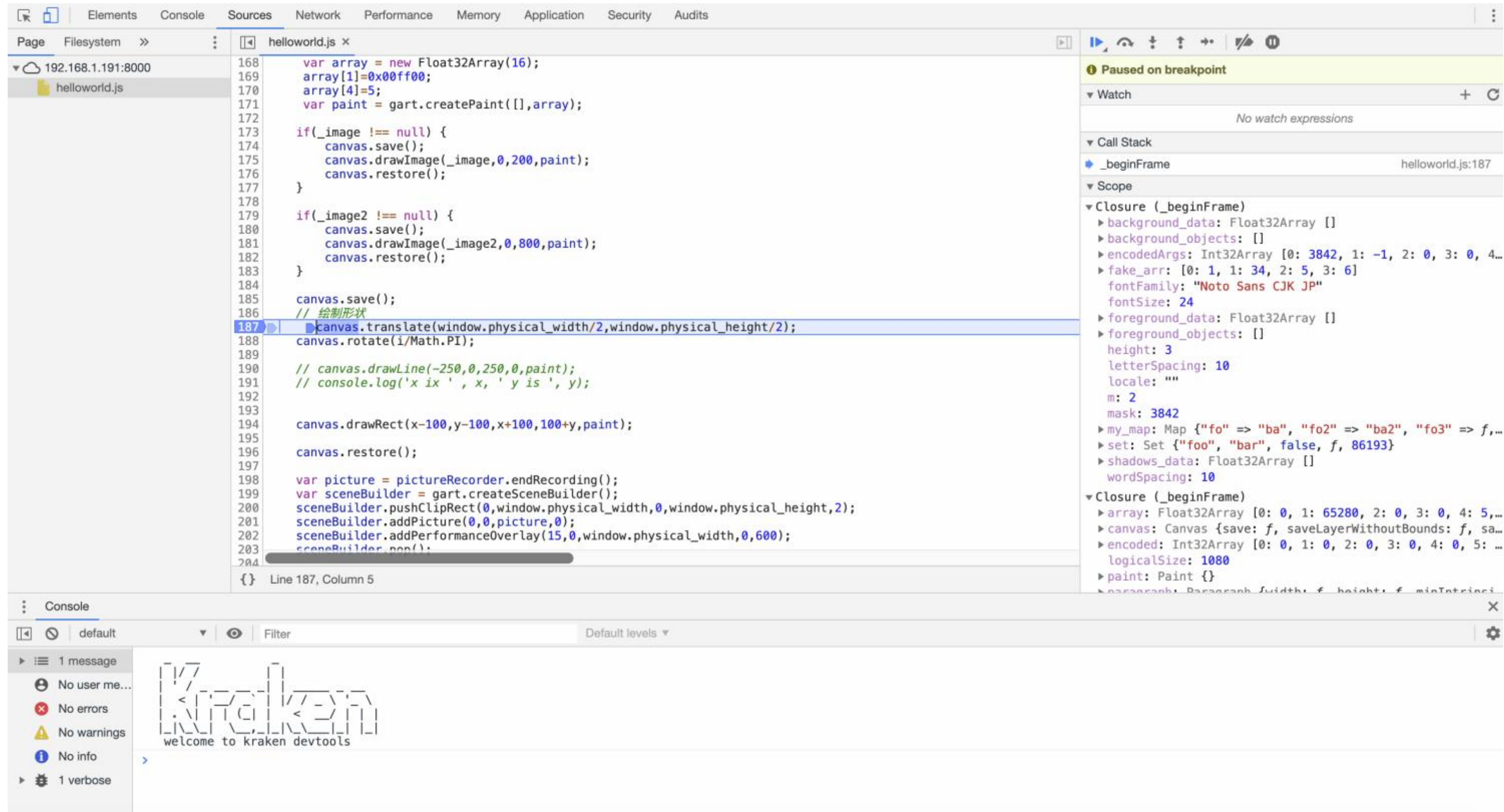
fetch

WebSocket

...

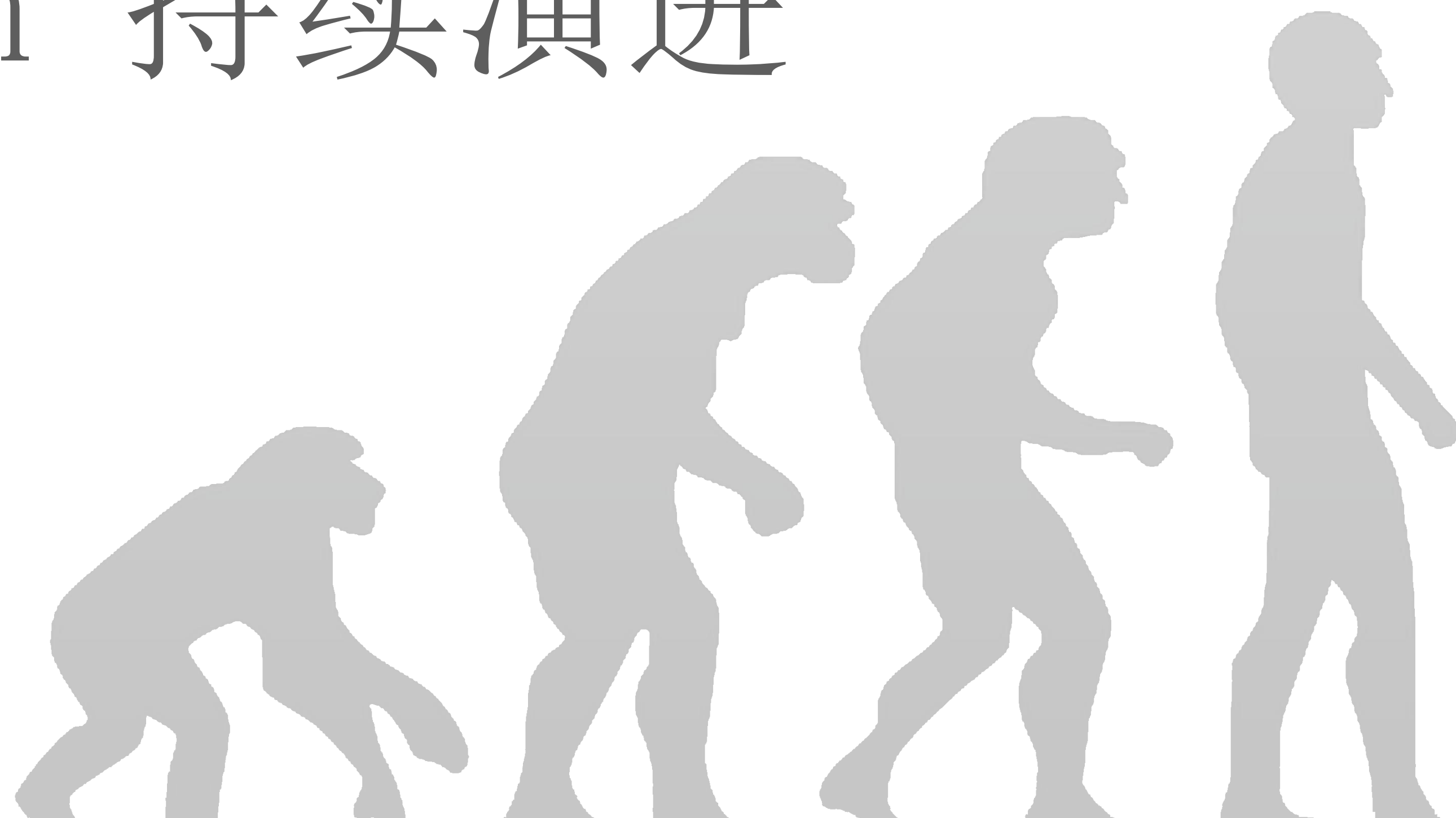
W3

Chrome Developer Tools

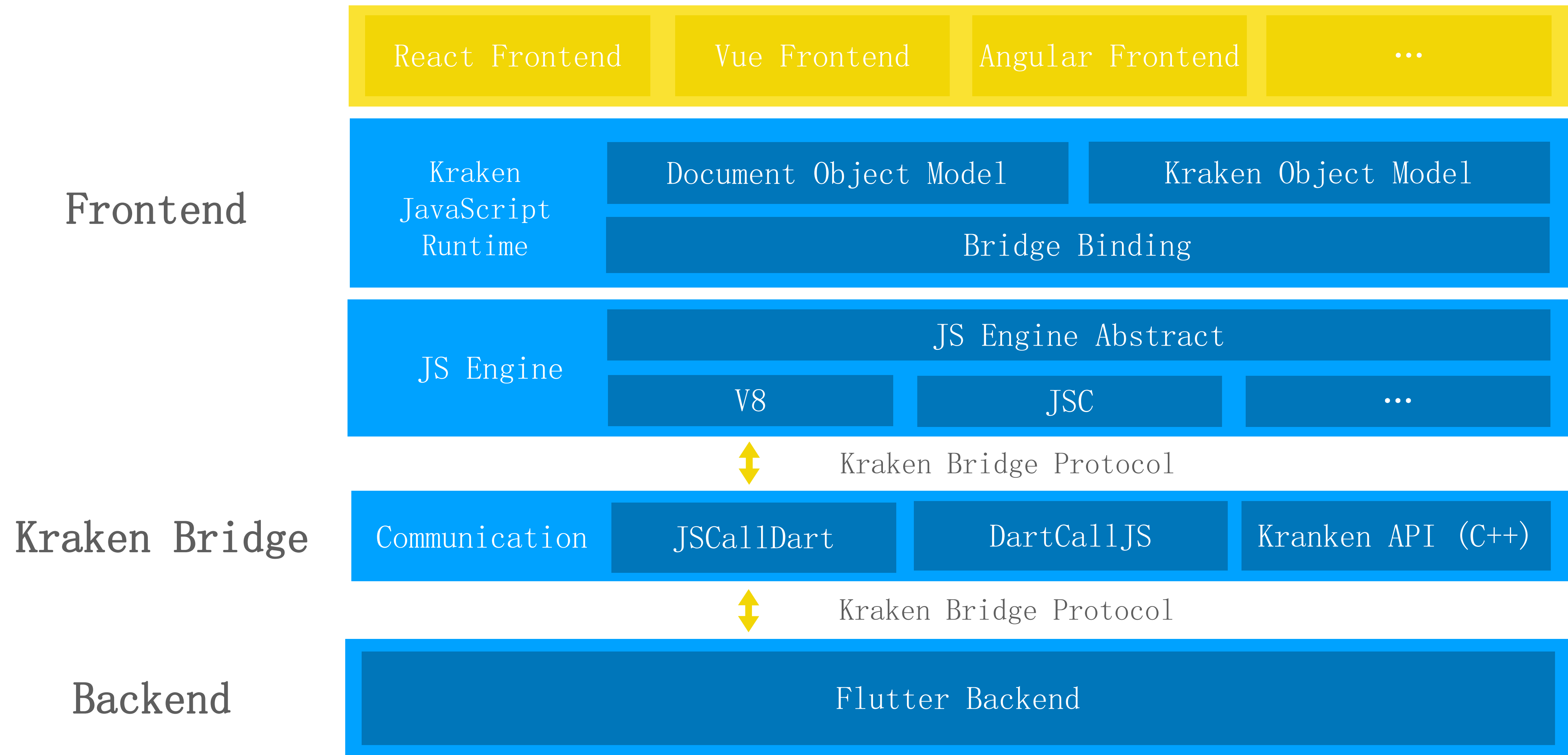




Kraken 持续演进

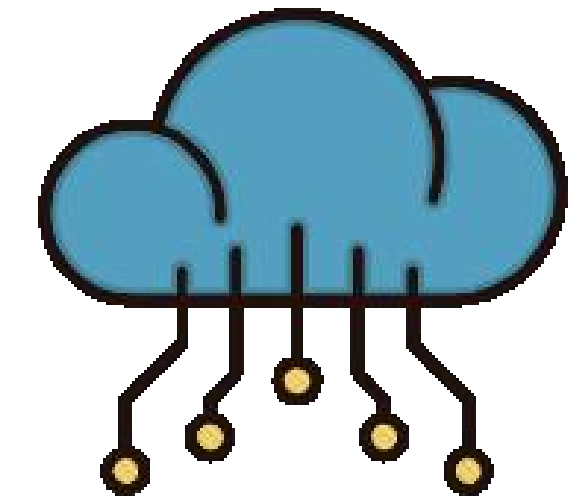


不仅仅是 React





Kraken with
Cloud



为什么上云？



云游戏



Microsoft xCloud



Google Stadia

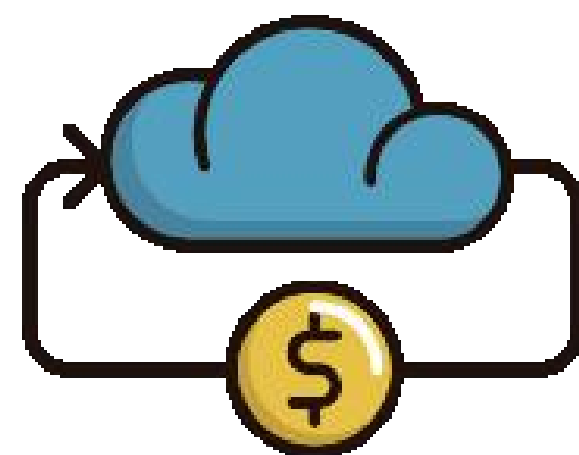


PlayStation Now

云化体验



免下载即开即玩



无需购买高性能设备

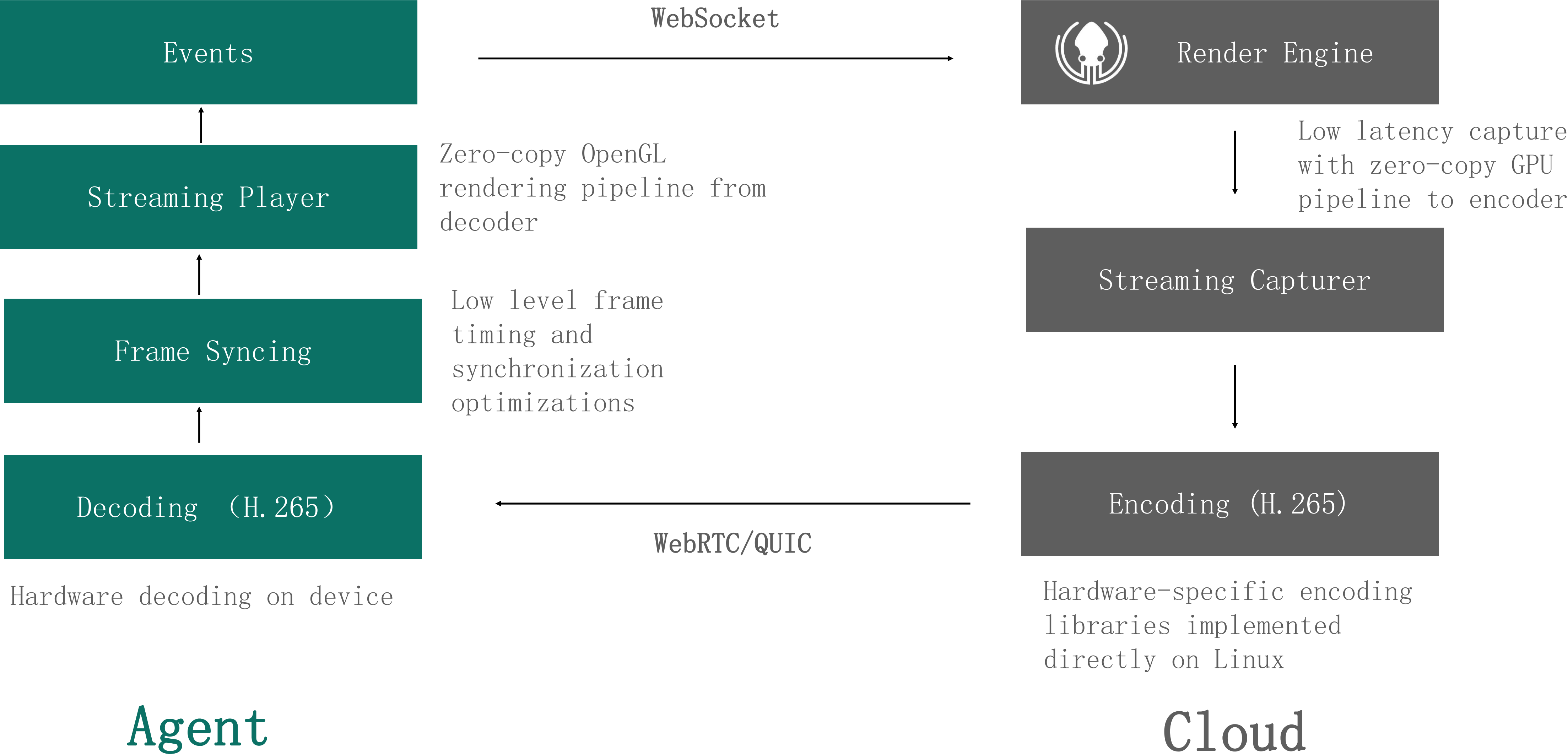


云计算共享

“

所有能云化的应用最终都将云化

Kraken 云渲染



“

云 + 端一体化是渲染技术的新趋势