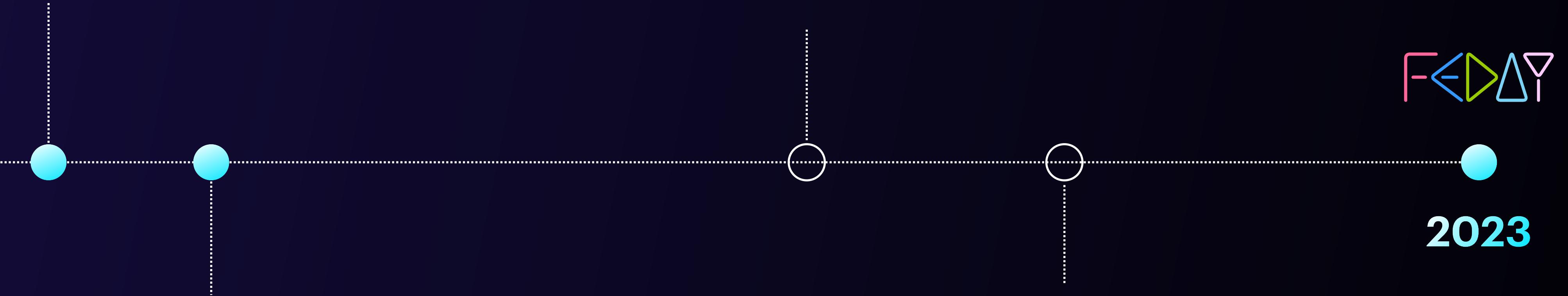




黃玄/@huxpro

「Progressive Web App, 复兴序章」  
QCon 上海 2016

「React Without Memo」  
React Conf 2021



「Upgrading to Progressive Web Apps」  
JSConf CN 上海 2017

「React 国情咨文 2022」  
第七届中国开源年会

黄玄/@huxpro



# My Worldview of 「大前端」

黃玄/@huxpro

@FEDAY Hangzhou Nov 2023



1. 未定义的前端
2. 拐点与成本理论
3. 再定义前端

黃玄/@huxpro

@FEDAY Hangzhou Nov 2023



# 1. 未定义的前端



**Back-end**

**Front-end**

**== Server?**

**Back-end**

**Front-end**

**== Server?**

**== Client?**

**Back-end**

**Front-end**

后端差不多就是服务端 🤔

== Client?

Back-end

Front-end

后端差不多就是服务端 🤖

Back-end

前端差不多就是客户端 😐

Front-end

**Back-end**

**Front-end**

**HTML, CSS, JS  
Browser**

HTML, CSS, JS  
Browser

Web 都是前端?

Back-end

Front-end

Wasm

PWA

HTML, CSS, JS  
Browser

Web 都是前端! 🙌

Back-end

Front-end

Flash      Wasm

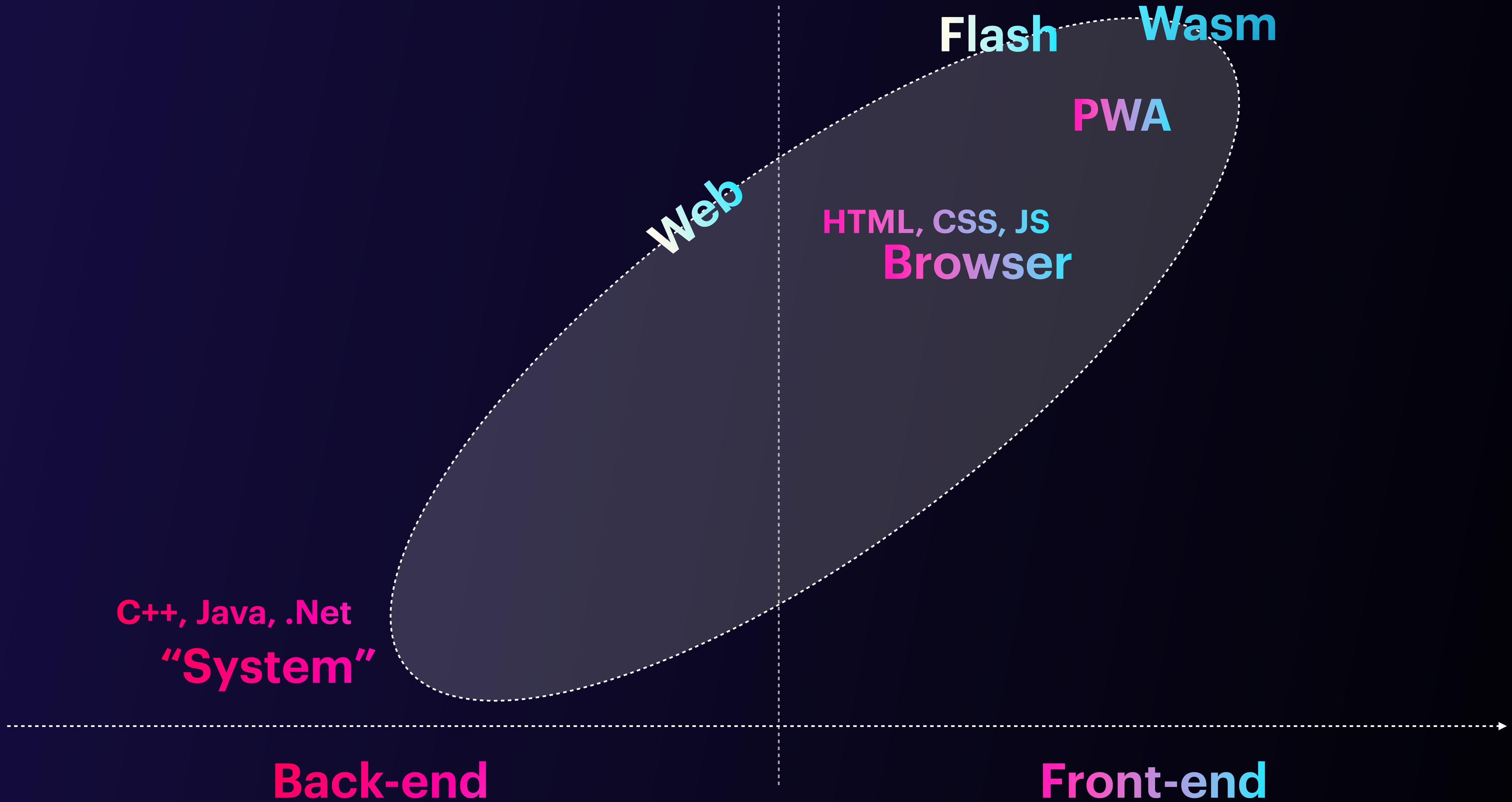
PWA

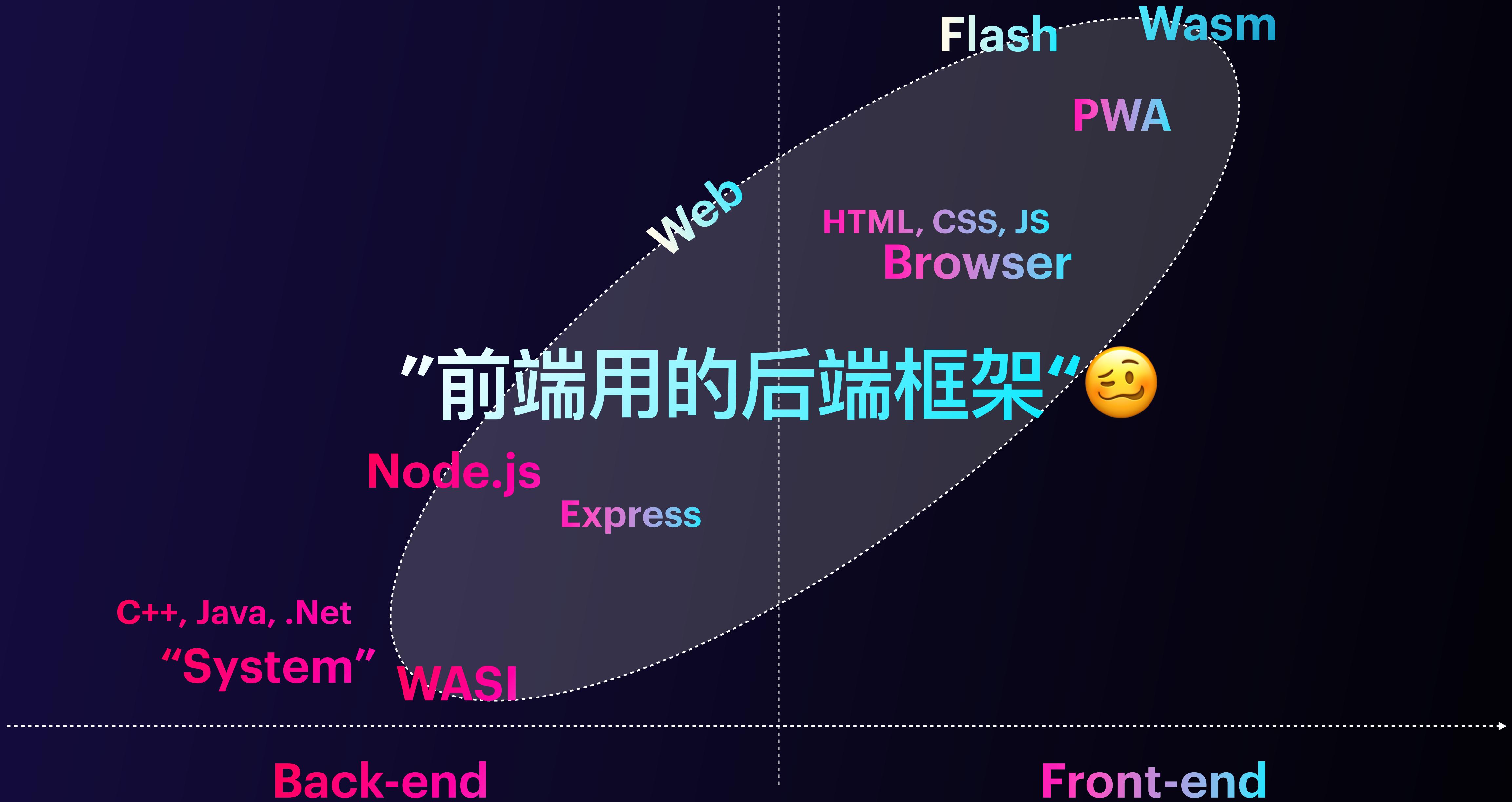
HTML, CSS, JS  
Browser

Web 都是前端! 🙌

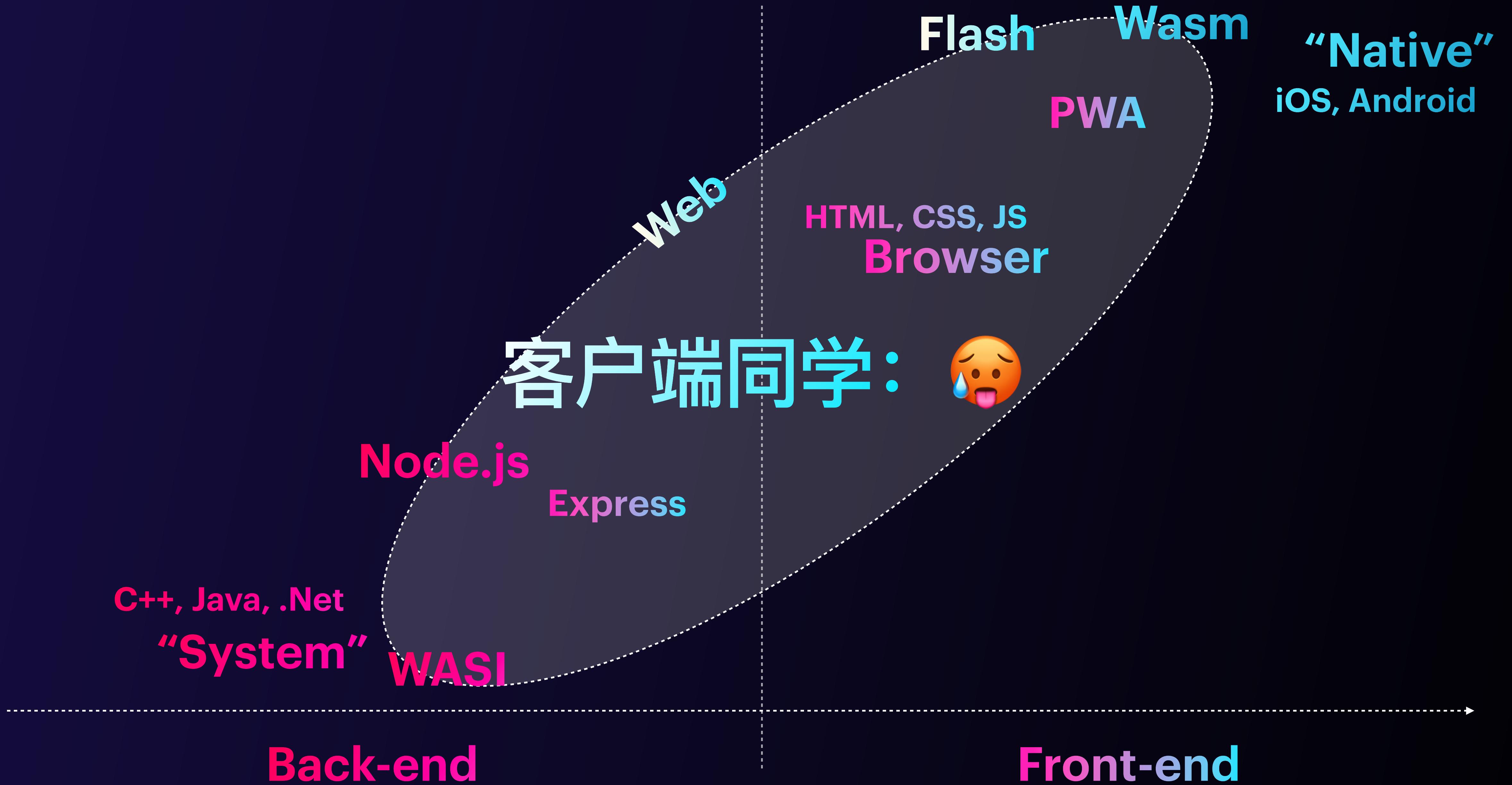
Back-end

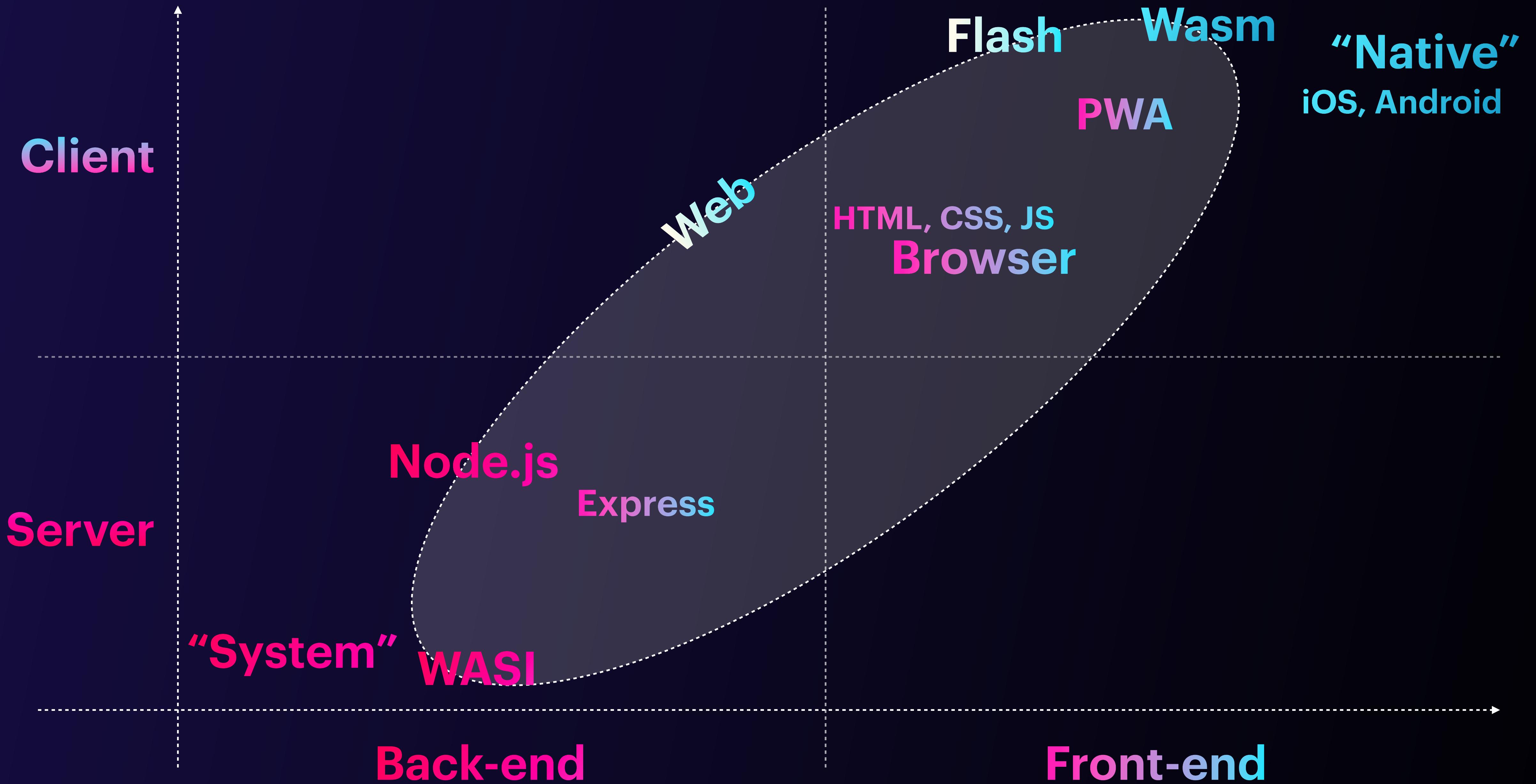
Front-end



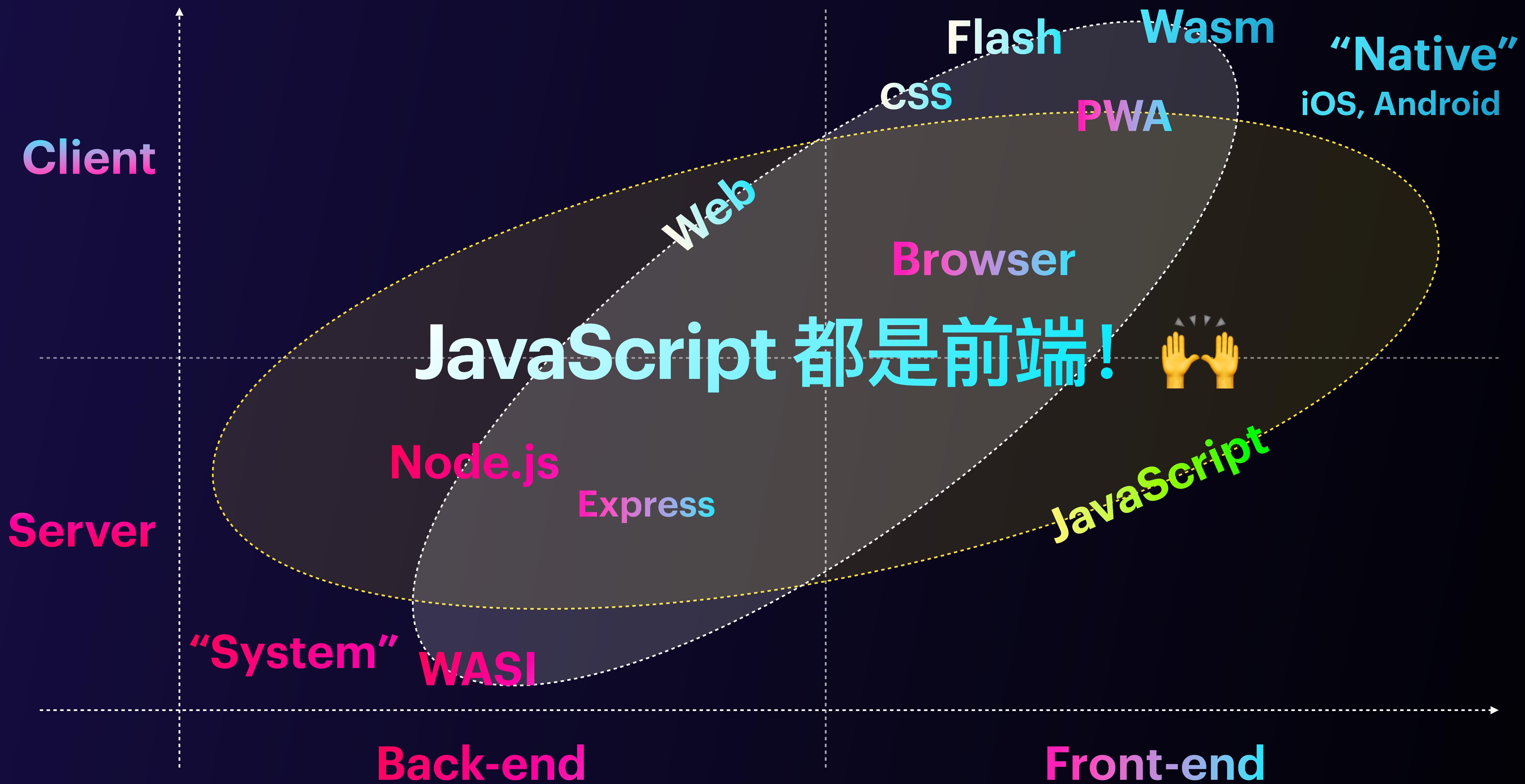


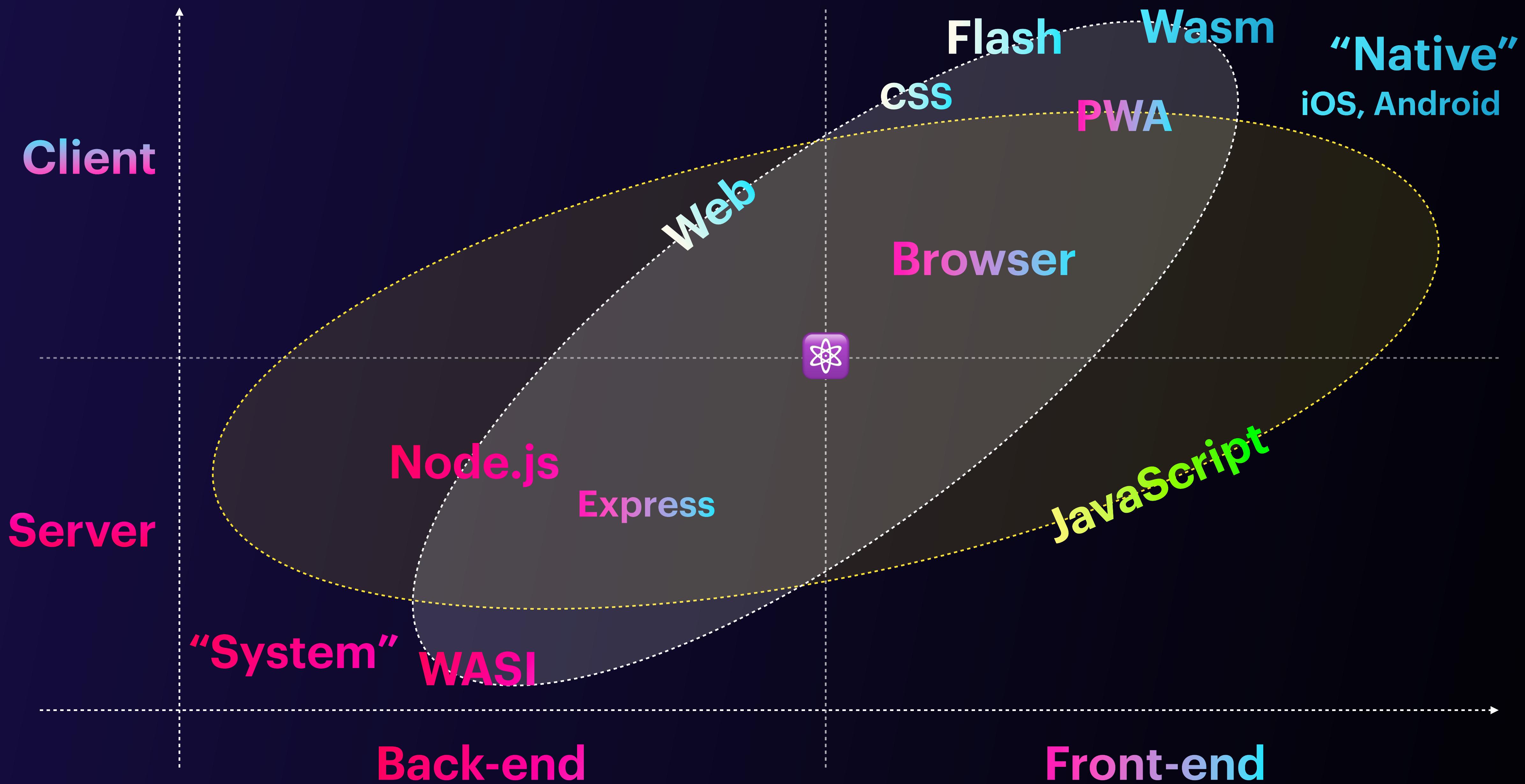


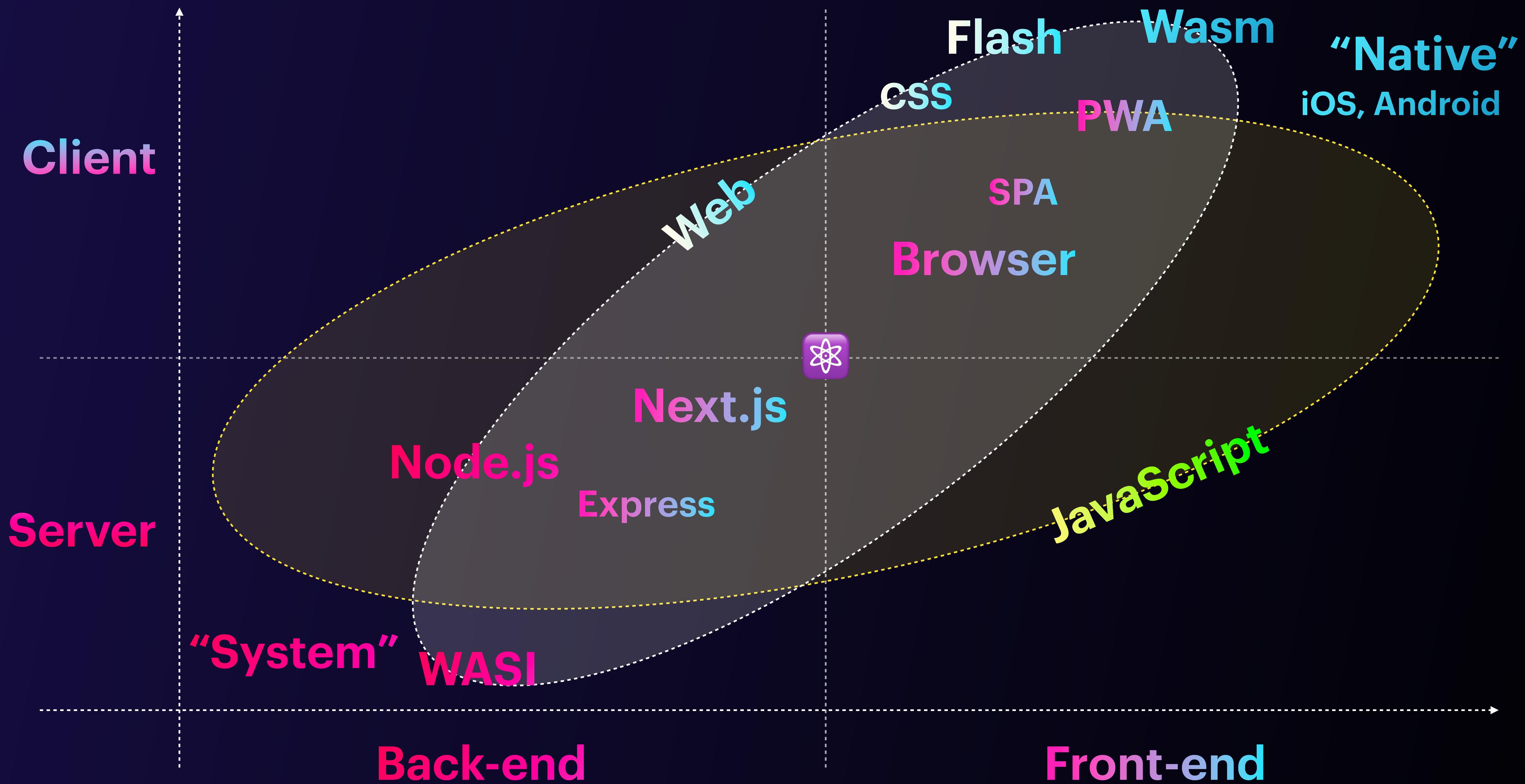


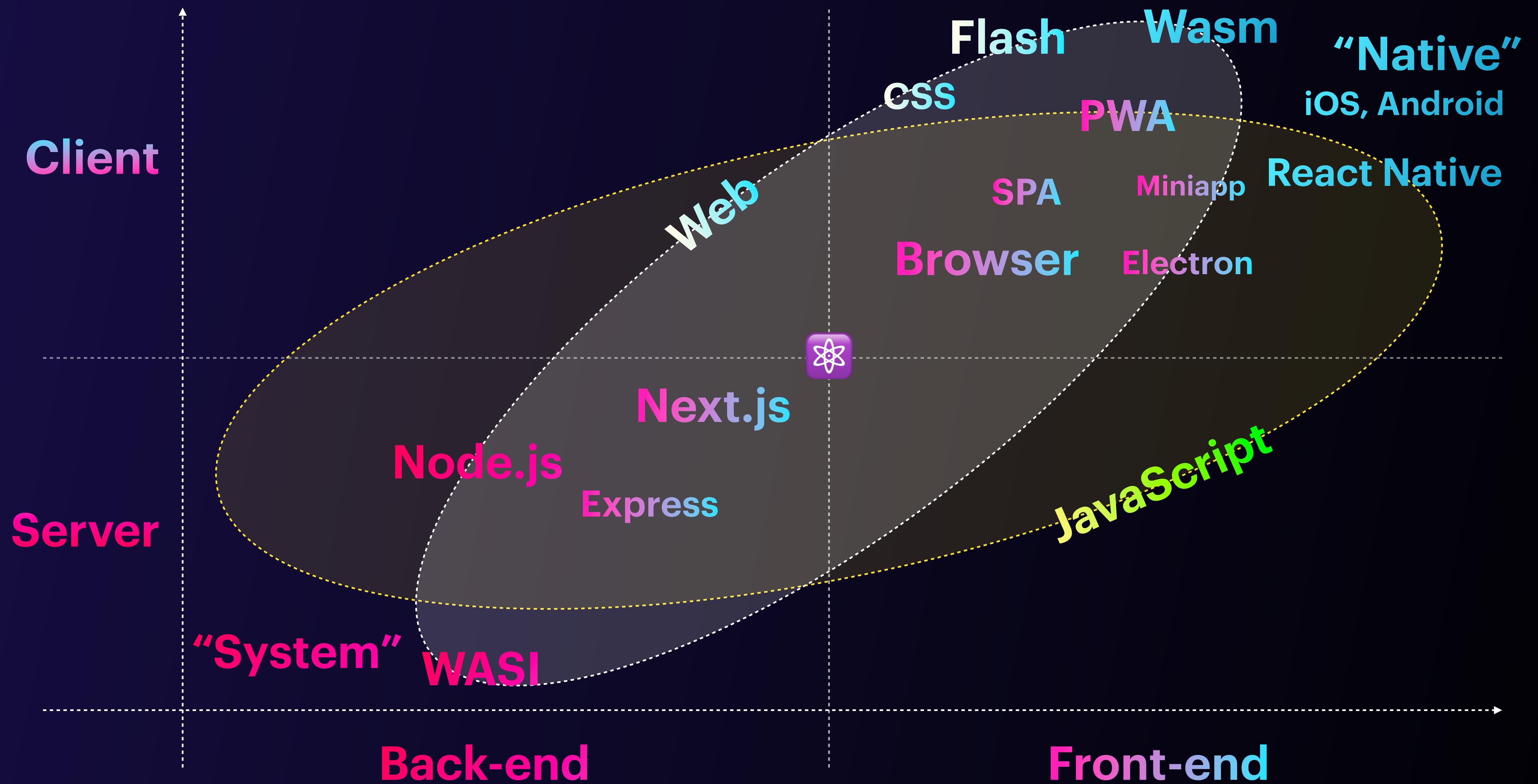


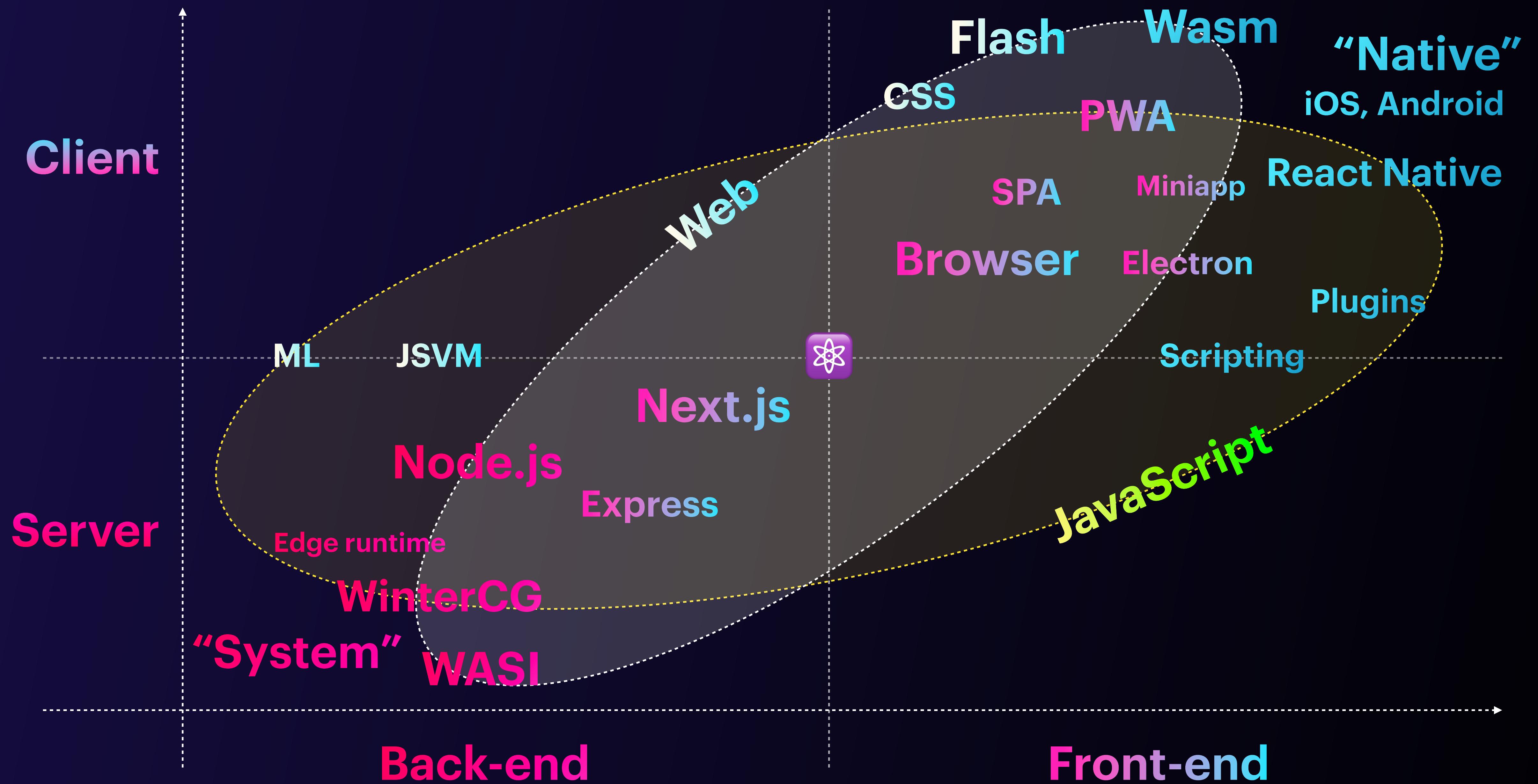












en.wikipedia.org

# WIKIPEDIA

## Frontend and backend

28 languages

Article Talk Tools

From Wikipedia, the free encyclopedia

In software engineering, the terms **frontend** and **backend** (sometimes written as **back end** or **back-end**) refer to the separation of concerns between the presentation layer (*frontend*), and the data access layer (*backend*) of a piece of software, or the physical infrastructure or hardware. In the client-server model, the **client** is usually considered the frontend and the **server** is usually considered the backend, even when some presentation work is actually done on the server itself.

### Introduction [edit]

In software architecture, there may be many layers between the hardware and end user. The *front* is an abstraction, simplifying the underlying component by providing a user-friendly interface, while the *back* usually handles data storage and business logic.

In telecommunication, the *front* can be considered a device or service, while the *back* is the infrastructure that supports provision of service.

A rule of thumb is that the client-side (or "frontend") is any component manipulated by the user. The server-side (or "backend") code usually resides on the **server**, often far removed physically from the user.

### Software definitions [edit]

This section is in list format but may read better as prose. You can help by converting this section, if appropriate. Editing help is available. (September 2020)

In content management systems, the terms *frontend* and *backend* may refer to the end-user facing views of the CMS and the administrative views,

chat.openai.com

## ChatGPT 4

You

Browse the given Wikipedia pages, does "back-end" equal to "server-side"? If it's not, give me some examples of non-Server-side (preferably client-side) backend?

- [https://en.wikipedia.org/wiki/Frontend\\_and\\_backend](https://en.wikipedia.org/wiki/Frontend_and_backend)
- [https://en.wikipedia.org/wiki/Server\\_\(computing\)](https://en.wikipedia.org/wiki/Server_(computing))

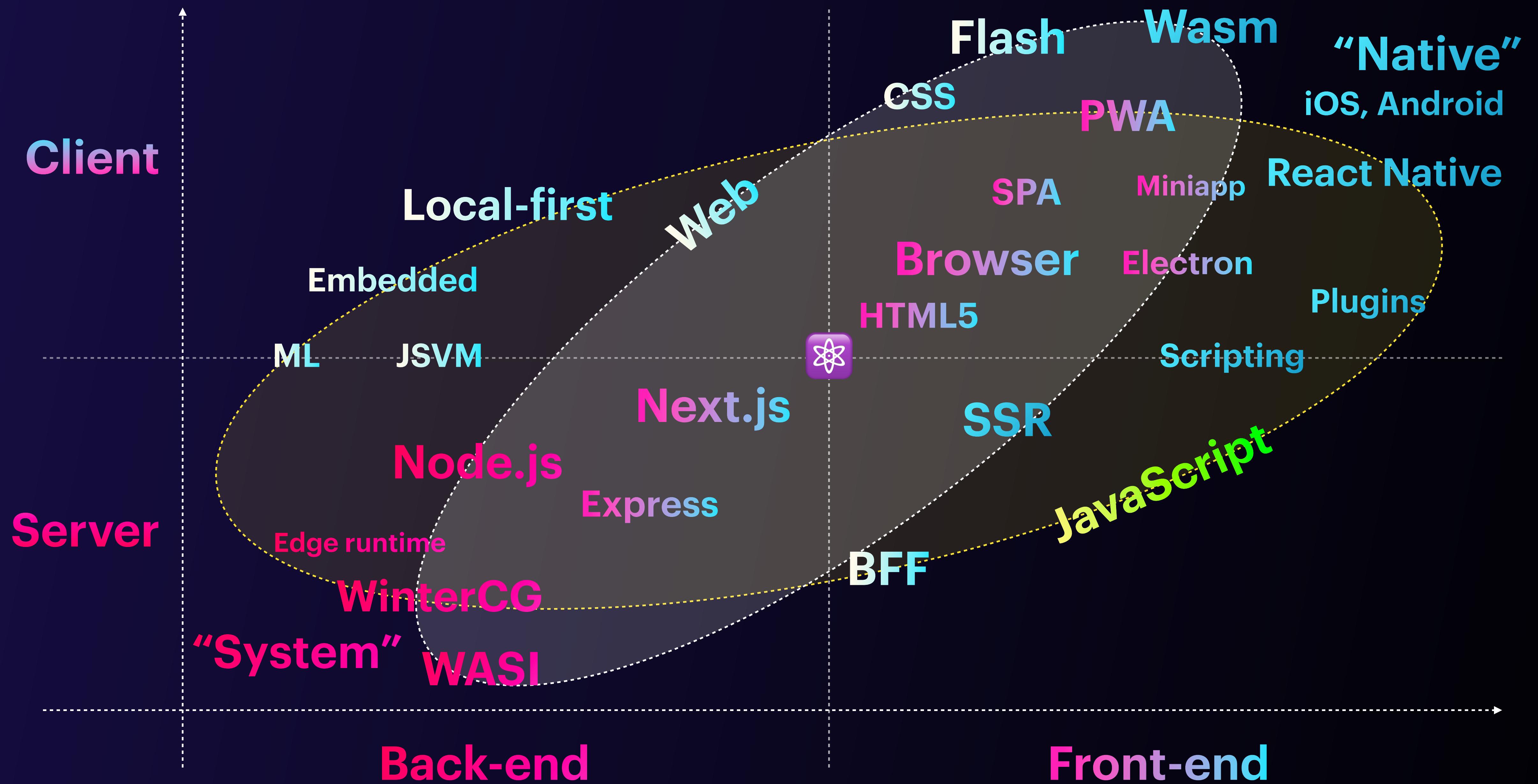
ChatGPT

In software engineering, "back-end" typically refers to the data access layer of a piece of software, or the physical infrastructure or hardware in the client-server model. The server, usually considered the back-end, is where the back-end code resides and handles data storage and business logic [1] [2] [3].

The term "server" in computing signifies either hardware or software (a computer program) that provides functionality for other programs or devices, known as "clients." This architecture, called the client-server model, allows servers to offer various functionalities, like sharing data or resources among multiple clients or performing computations for a client [4].

"Back-end" does not necessarily equate to "server-side" because not all back-end systems are servers in the traditional sense. Non-server-side back-end systems could include:

- **Local Databases:** These are used by client-side applications for storing data locally on a user's device. For instance, SQLite is a database often used as a local storage solution in mobile applications.
- **Background Processes:** In mobile applications, certain tasks may be performed in the background, such as data syncing or updating local content from a remote source.
- **Local File Systems:** Applications may also interact with the local file system on the client's device as a form of back-end storage and management, which doesn't involve a remote server.
- **Client-side Business Logic:** This can involve processing and calculations done on the client's machine, not on a remote server.





## 2. 拐点与成本理论

---

黃玄/@huxpro

Flash

Motion  
Game Graphics

Animation UI/UX



2012

黃玄/@huxpro

Flash

JavaScript

Game Motion  
Graphics

Animation

HTML5

UI/UX

Mobile

Web

PWA

FE Infra

Alibaba



饿了么

2012

2017

黃玄/@huxpro

# Flash

# Game Graphics

# Animation

# Motion Graphics

# HTML5

# UI/UX

 Alibaba

# JavaScript

# Mobile

# Web

# FE Infra



玄/@huxpro

**2012**

# 2017

# 2023

174



# PL&R

# ReasonML

# React Native

# React Core

兴趣使然  
前端？



2012

JavaScript

Mobile

Web

FE Infra

HTML5



2017

PL&R

ReasonML

PWA

JSVM (Hermes)

React Native

React Core



2023

黃玄/@huxpro

兴趣使然  
前端？



2012

前端工作者  
我是前端！



2017

PL&R

ReasonML

JSVM (Hermes)

React Native

React Core



2023

黃玄/@huxpro



职业规划

程序员修养

前端工程师

计算机科学与技术

关注者

248

被浏览

197,243

## 作为一个前端，看不懂@黄玄 的几乎每一个回答，只有我自己吗？

RT，不知道该怎么办，有点焦虑。大家都说看不懂，那@黄玄 他会独孤吗？不会取关他，没准哪天他突然开始疯狂输出一些我能看懂的东西呢！[显示全部](#)

已关注

[查看我的回答](#)[邀请回答](#)

好问题 31

3条评论

分享

...

他们也关注了该问题

[查看全部 42 个回答](#)

冯天然

让团队变得更好

互相关注

他早就不是前端了啊，你只是一个前端，看不懂不是很正常嘛？

-----稍微补充一下

我觉得绝大多数回答还是有个问题，那就是认为他是前端。

事实上他是一个还会在前端圈里游走的PL从业者，而非一个学习过PLT的前端从业者。

就像死月是个后端而非前端一样（倒不是故意拿死月举例子，不过我觉得死月也算是经常被误解为前端吧），黄玄也是个PL&编译器开发者而非前端（当然了过去曾经是）。

所以我觉得前端看不懂他的回答再正常不过，虽然我们经常说程序员自身是不设限的，但是你仍然会有自己的技术方向和岗位需求。现在的计算机科学本身也有非常多的细分领域。你并不需要搞清



### 关于作者



冯天然

让团队变得更好

玉伯、弘树.dev、岳逢乐也关注了他

兴趣使然

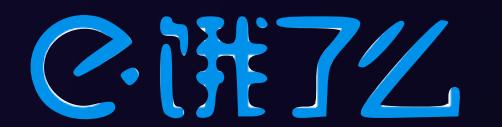
前端？

前端工作者

我是前端！

技术栈生产者

我是前端吗？



2012

2017

2023

黄玄/@huxpro



# 我的「大前端」世界观

黃玄/@huxpro

@FEDAY Hangzhou Nov 2023

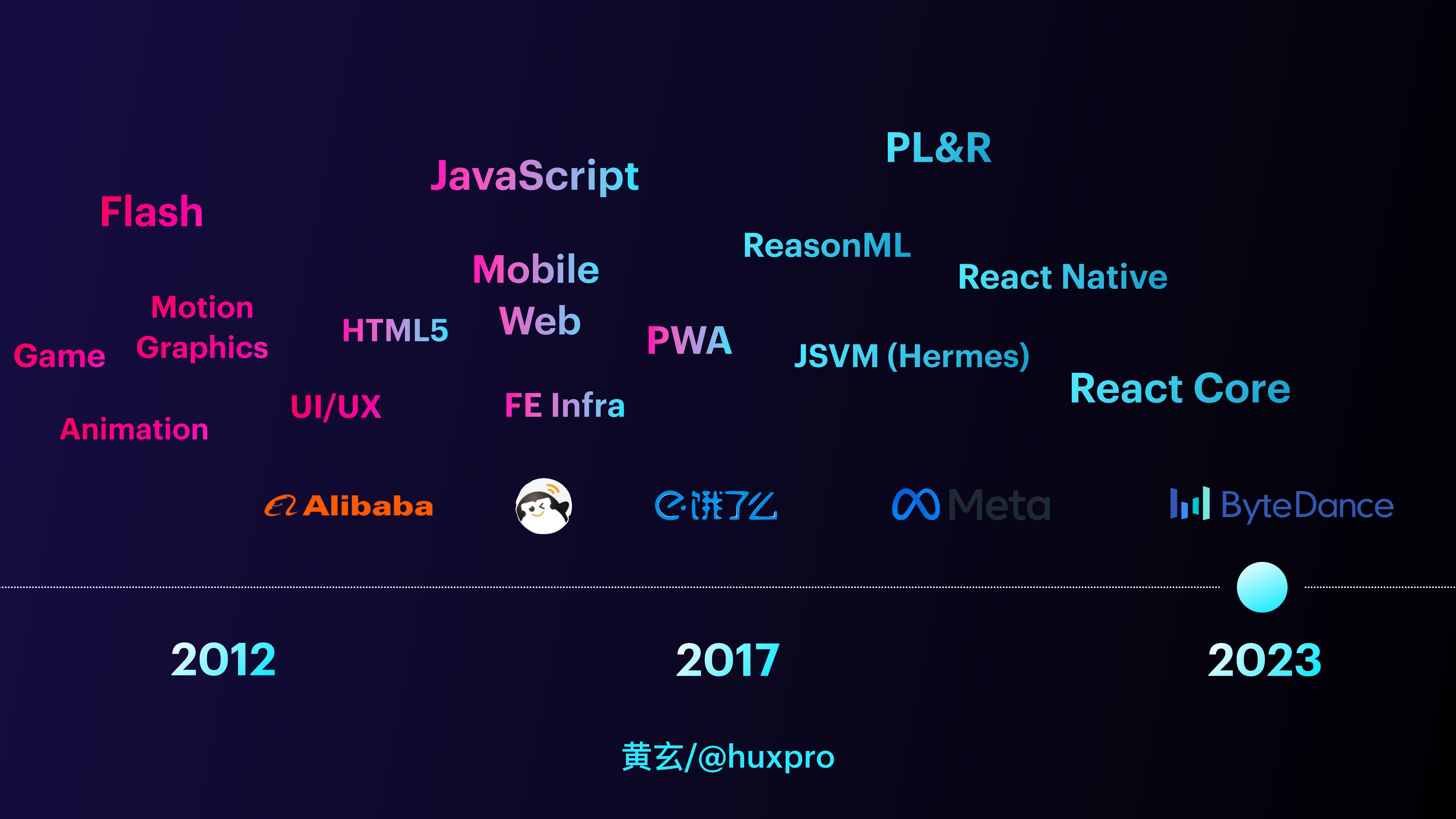


# 我的「前端」身份危机

Identity crisis

黃玄/@huxpro

@FEDAY Hangzhou Nov 2023



2012

2017

2023

黃玄/@huxpro

Flash

Mobile  
Web

React Native



---

2012

2017

2023

黃玄/@huxpro

Mosaic

Web

1993

iPhone

Mobile

2007

Flash

Mobile  
Web

2012

React Native

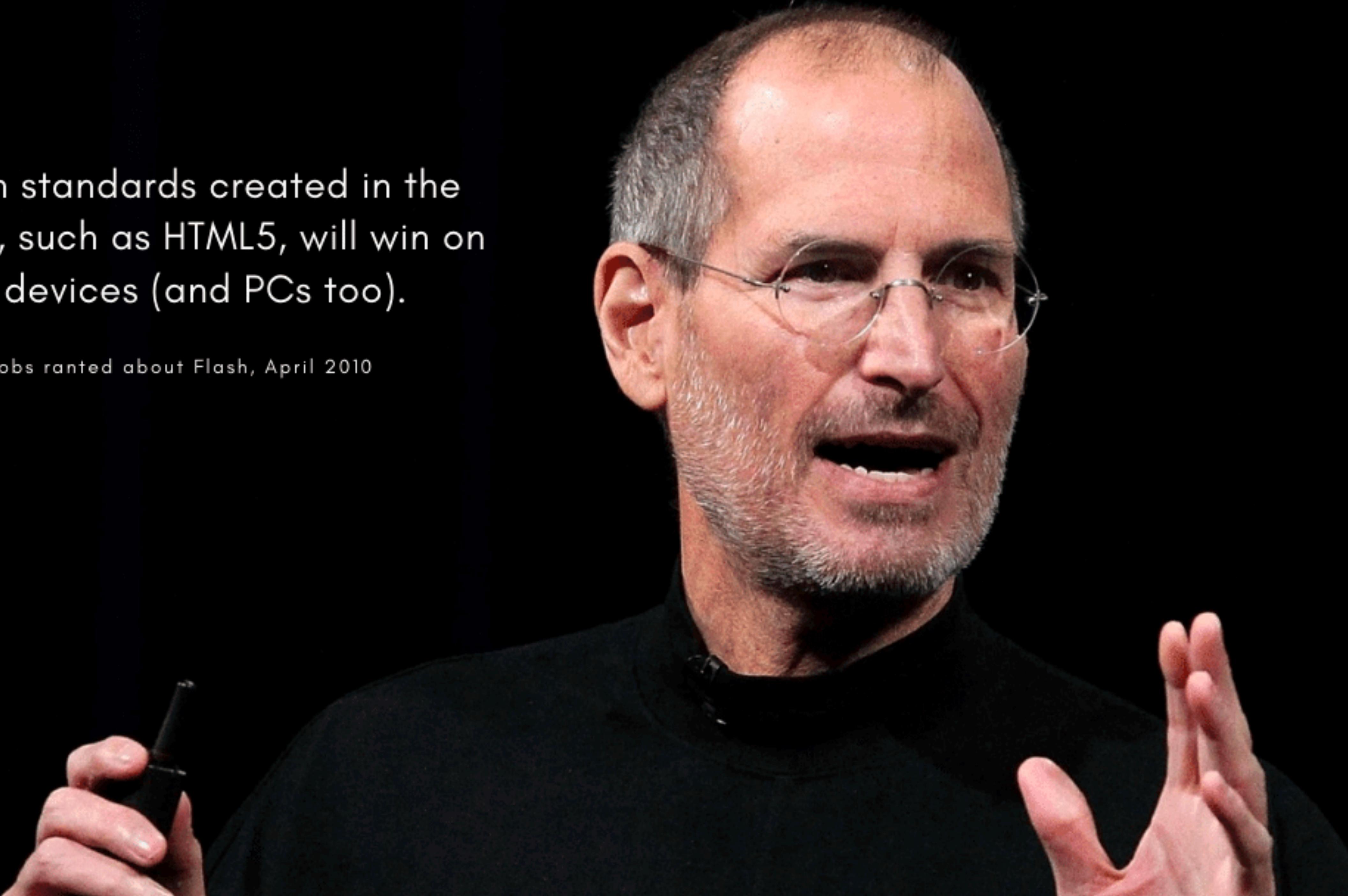
2017

2023



New open standards created in the mobile era, such as HTML5, will win on mobile devices (and PCs too).

Steve Jobs ranted about Flash, April 2010



“一个人的命运啊，  
当然要靠自我奋斗，  
但也要考虑历史的行程。”



Mobile first to AI first



Mosaic

Web

1993

iPhone

Mobile

2007

Google  
Assistant

AI

2012

2017

2023





Mosaic

Web

1993

iPhone

Mobile

2007

ChatGPT

AI

2012

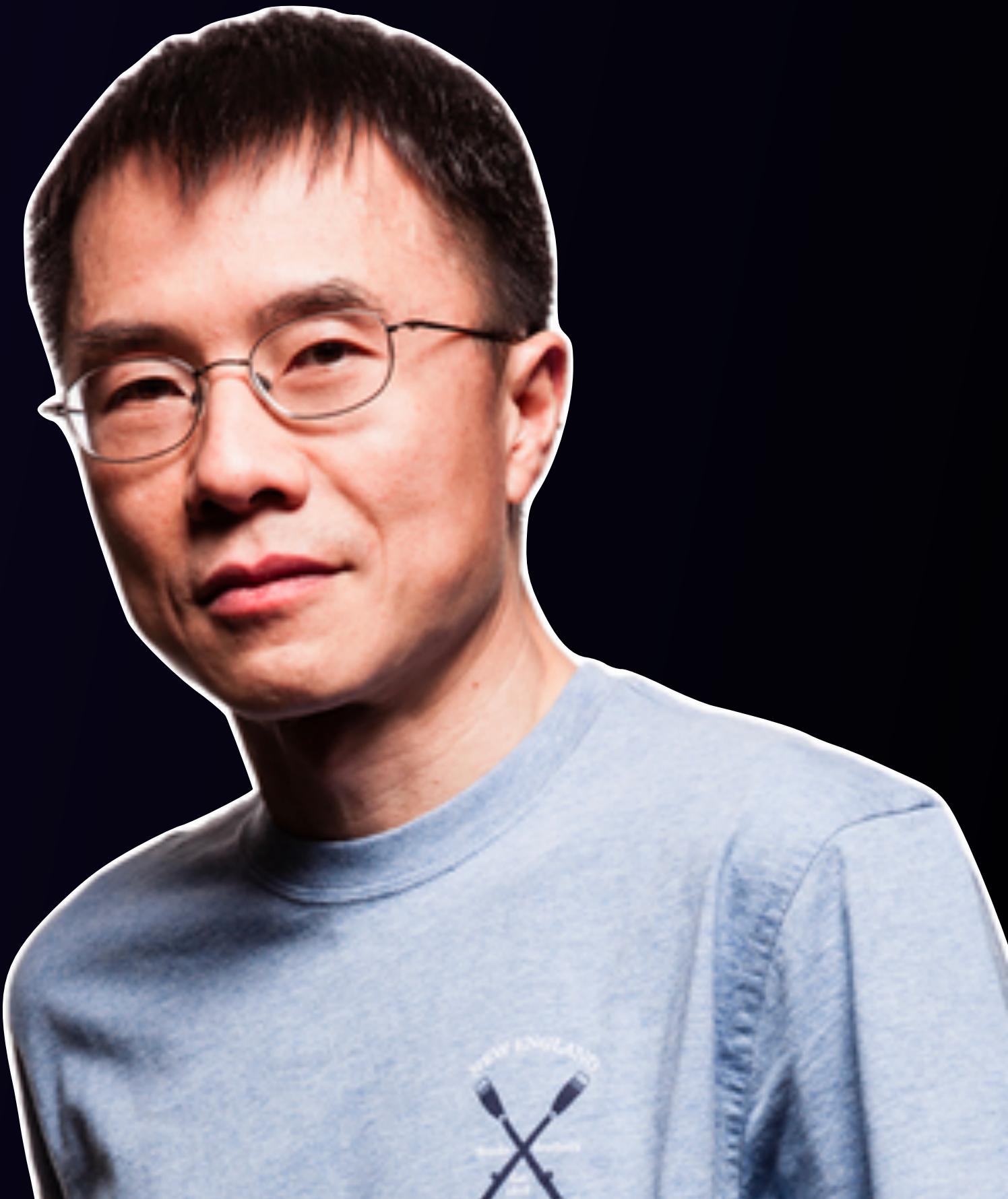
2017

2023



“社会性拐点的核心，  
是一项大型成本，  
从边际成本，  
变成固定成本。”

—— 陆奇《我的大模型世界观》



早在1995-1996年，通过PC互联网迎来一个拐点。那时我刚从CMU（卡内基梅隆大学）毕业。大量公司层出不穷，其中诞生了一家伟大公司叫Google。为什么会有这个拐点？为什么会有爆炸式增长？把这个观点讲清楚，就能把今天的拐点讲清楚。

**原因是，获取信息的边际成本开始变成固定成本。**

**一定要记住，任何改变社会、改变产业的，永远是结构性改变。这个结构性改变往往是一类大型成本，从边际成本变成固定成本。**

举个例子，我在CMU念书开车离开匹茨堡出去，一张地图3美元，获取信息很贵。今天我要地图，还是有价钱，但都变成固定价格。Google平均一年付10亿美元做一张地图，但每个用户要获得地图的信息，基本上代价是0。也就是说，获取信息成本变0的时候，它一定改变了所有产业。这就是过去20年发生的，今天基本是free information everywhere（免费的信息无处不在）。

Google为什么伟大？它把边际成本变成固定成本。Google固定成本很高，但它有个简单商业模式叫广告，它是世界上高盈利、改变世界的公司，这是拐点关键。

**今天2022-2023年的拐点是什么？**它不可阻挡、势不可挡，原因是什么？一模一样。模型的成本从边际走向固定，因为有件事叫大模型。

$O(N) \rightarrow O(1)$

0(Pointers) -> 0(1)

$O(\text{DOM Ops}) \rightarrow O(1)$

$O(\text{Platforms}) \rightarrow O(1)$



### 3. 再定义前端





现在，我给大家一个结构化思维框架。某种意义上你可以对号入座，知道我在这里，我怎么思考今天的机会点。



这张图是整个人类技术驱动的创业创新，所有事情的机会都在这张图上。



# “Omni-Frontend”



# Mobile

# AI



# Mobile

# AI



Flash

Web

Android

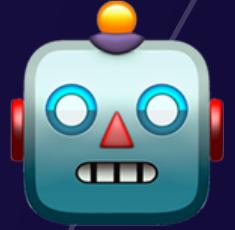
iOS

React



# Mobile

# AI



Flash

0(UIs) → 0(1) ?

Web

Android

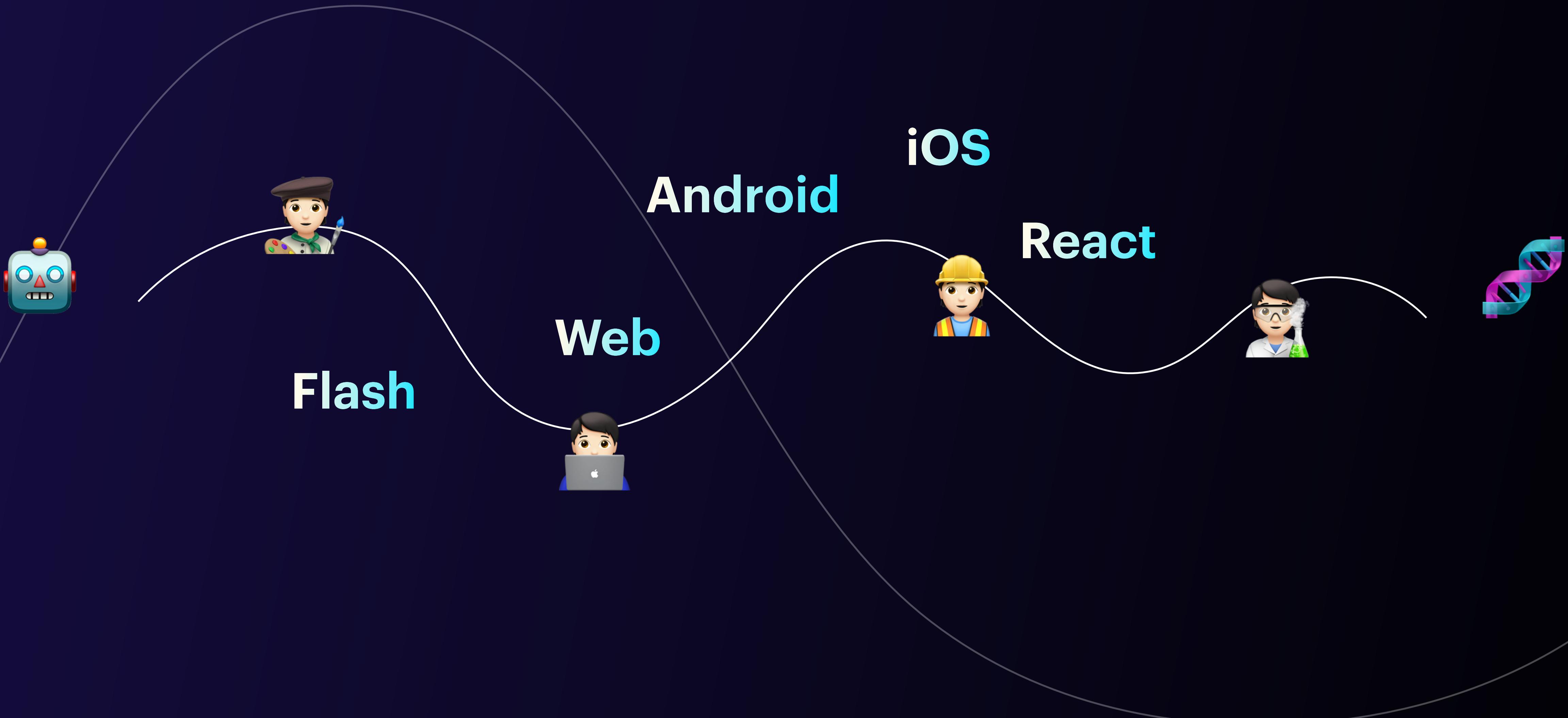
iOS

React



# Mobile

# AI



一个工种的价值，  
本质上即其人力（智）  
在对抗的边际成本的大小。

黃玄/@huxpro

0(Assembly Code) → 0(1)

技术的发展，  
必然朝着消解边际成本  
——即消解工种的方向。

黄玄/@huxpro

$O(\text{Browser Vendors}) \rightarrow O(1)$

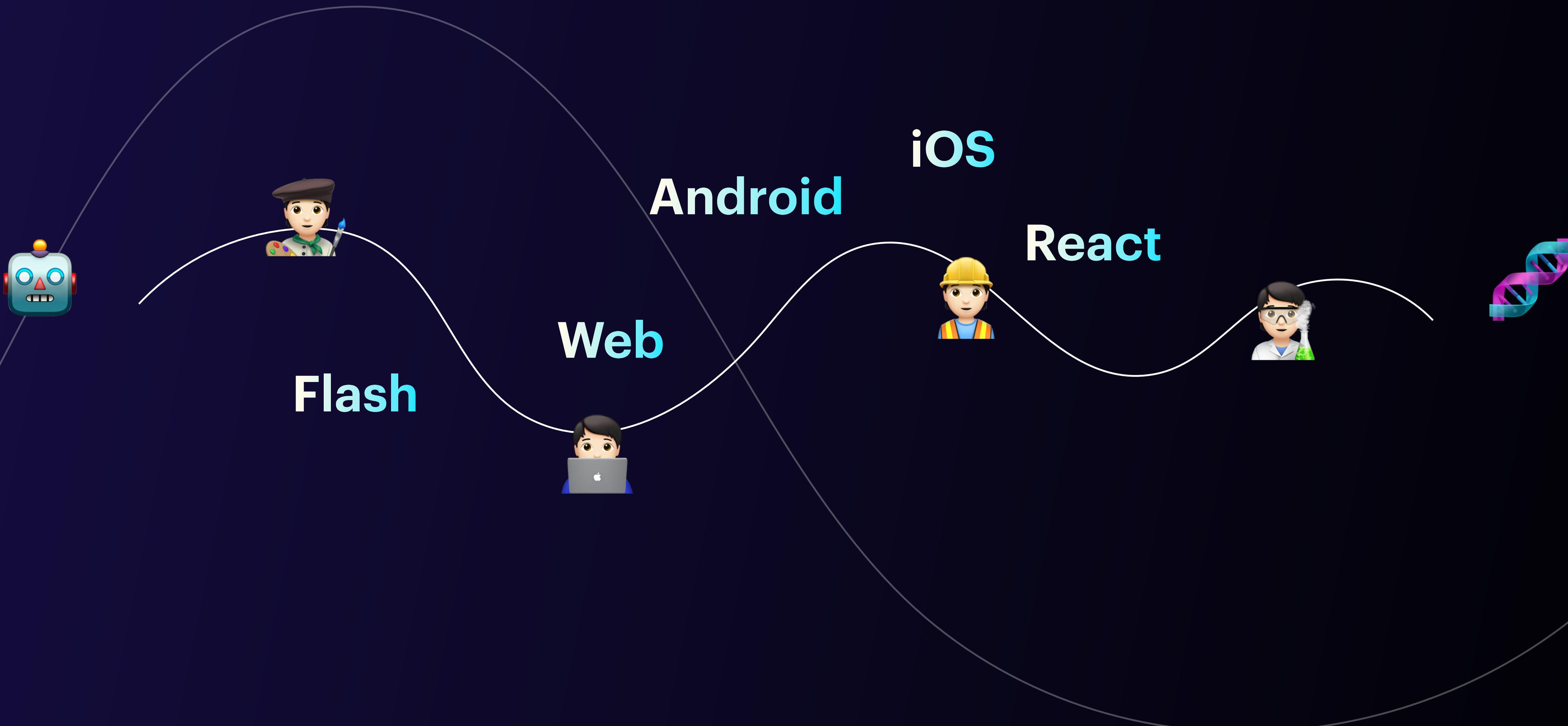
$O(\text{Servers}) \rightarrow O(1)$

这个过程带来的工种消解，  
首先往往发生在技术行业内，  
然后是全社会的重新配置。

黄玄/@huxpro

# Mobile

# AI



# mobile

# AI



Flash

Web



Android

iOS

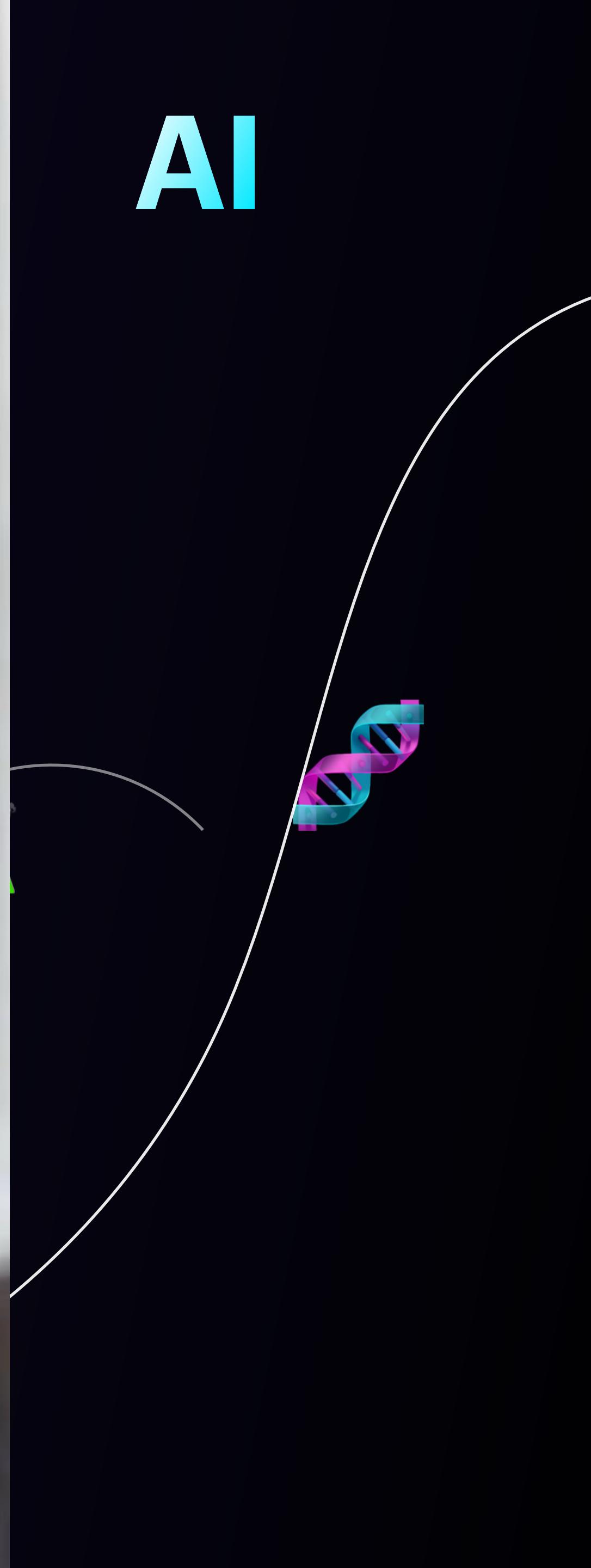
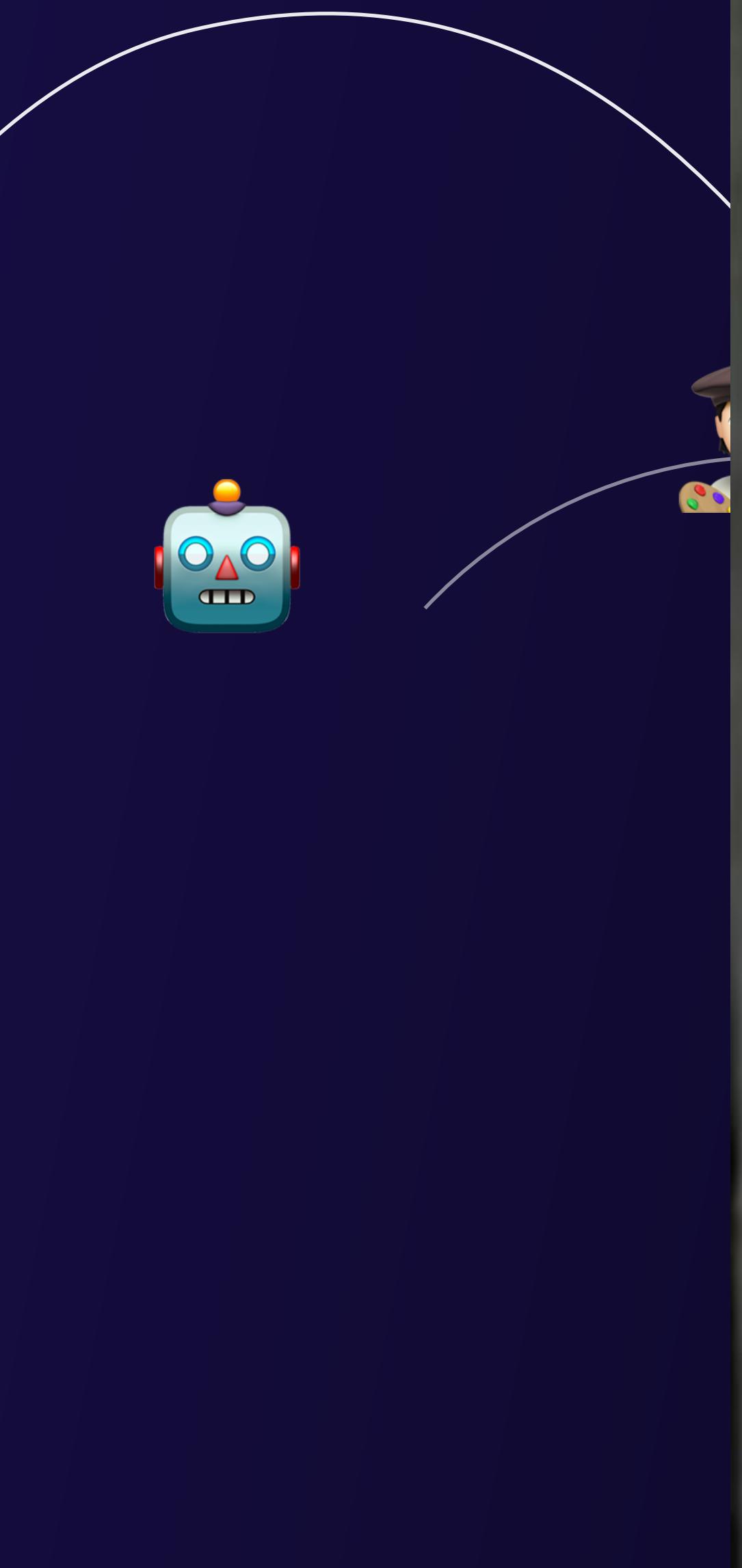


React



obile

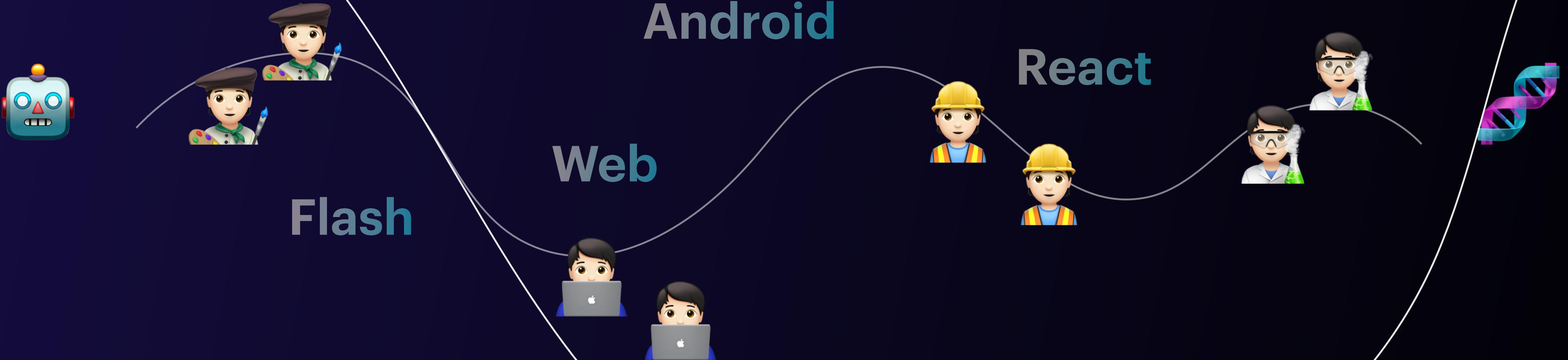
AI



小红书号：609979973

# mobile

# AI



obile

AI

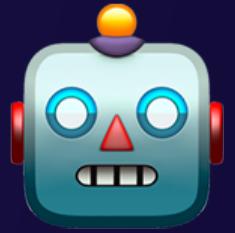
# Omni-Frontend



obile

# Omni-Frontend

AI



2D 展示信息足够好用  
3D/XR 创造新体验

obile

# Omni-Frontend

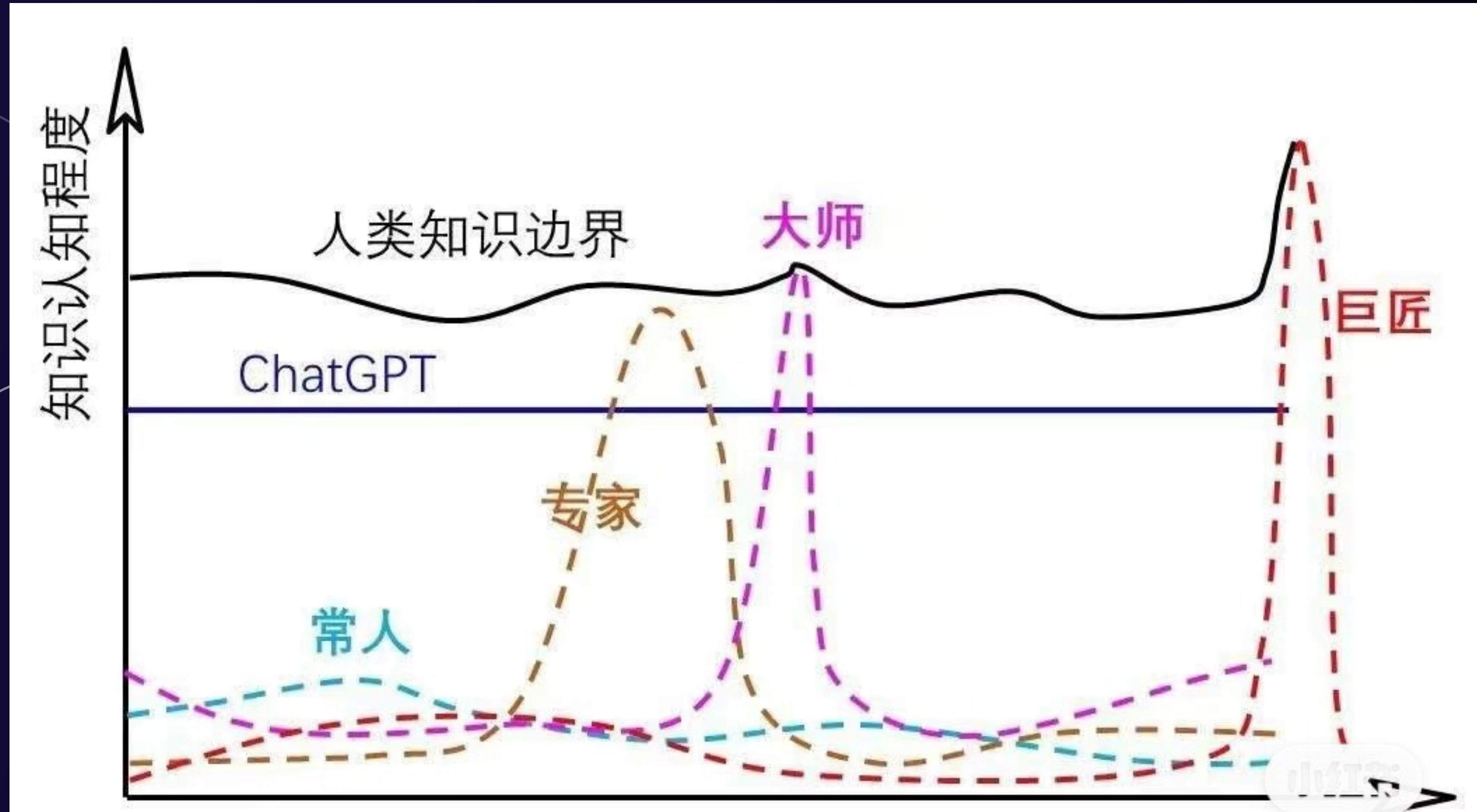
AI



obile

# Omni-Frontend

AI



知识领域维度 红书号：9334494259

知识领域维度

obile

# Omni-Frontend

AI



mobile

# Omni-Frontend

AI



**SDK Developer**  
**Infra Engineer**  
**Framework Engineer**

**App Developer**  
**Fullstack Engineer**  
**X-plat Engineer**

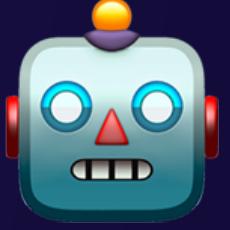
obile

# Omni-Frontend

AI



We are hiring!



**Web Infra, Xplat Infra, Virtual Machine, Rendering**



Shameless Plug

obile

# Omni-Frontend

AI



**System Engineer**



**Product Engineer**



obile

AI



**Scientist**



**Entrepreneur**





# Cause-Effect Relationship

Ray Dalio



# THE END

黃玄/@huxpro

@FEDAY Hangzhou Nov 2023