



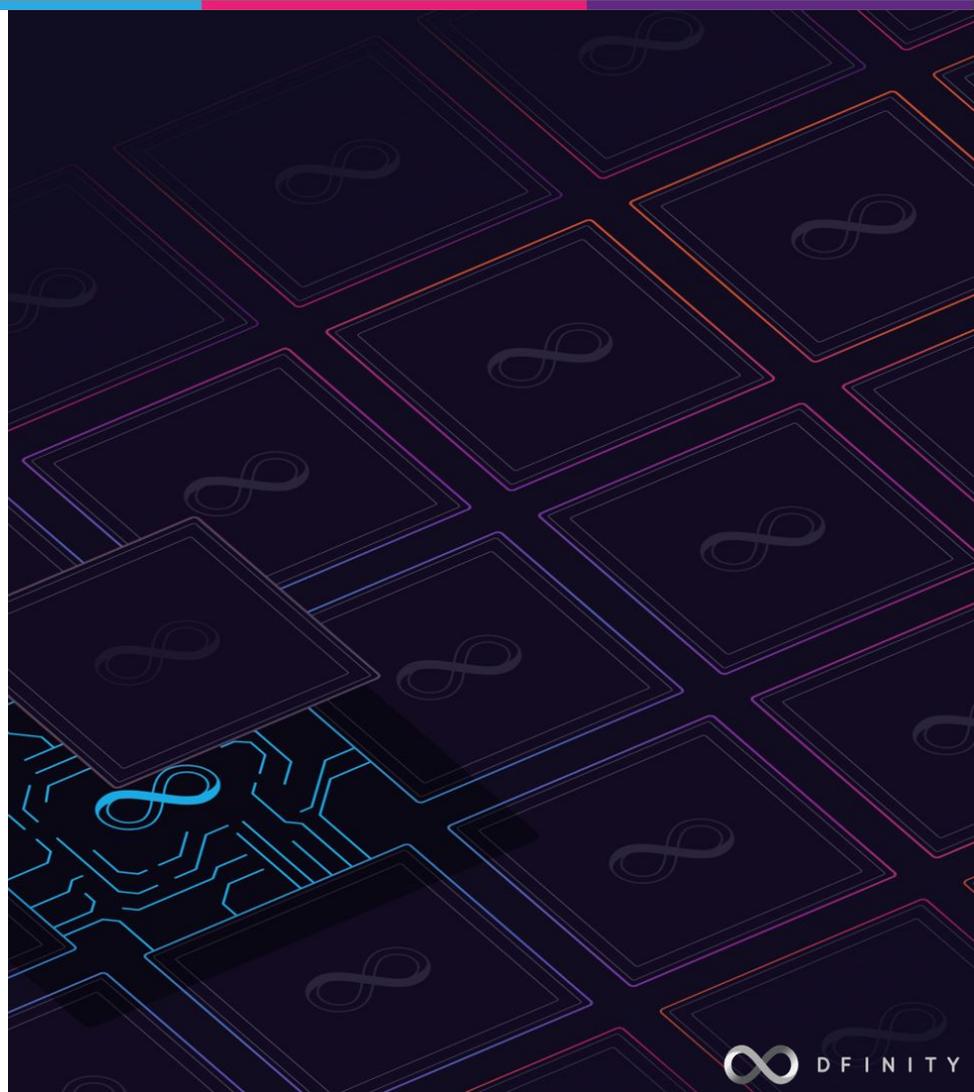
# ICP区块链开发入门课程

5. 用 Javascript 做前端

主讲: Paul Liu - DFINITY 工程师

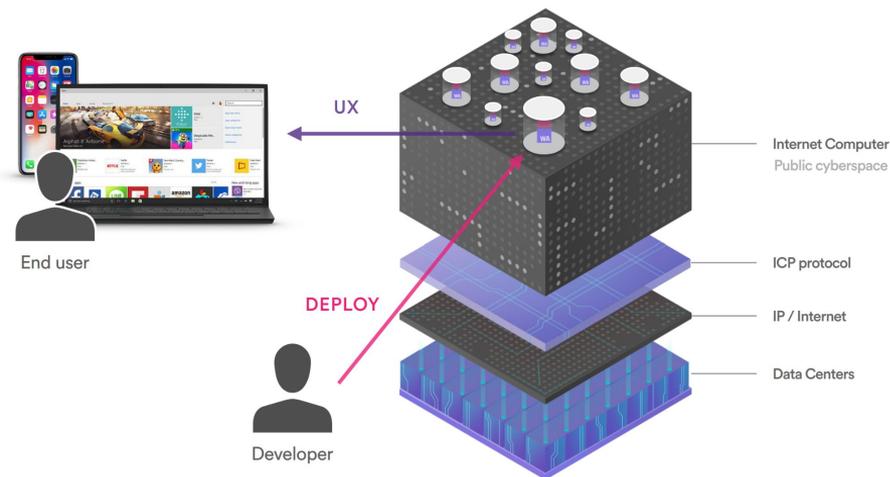
# 课程大纲

1. 使用 SDK 搭建一个简易网站
2. Motoko 语言简介
3. Canister 智能合约
4. 用 Motoko 做后端
5. 用 Javascript 做前端



# 前端逻辑

- 界面 (UI), 与用户交互 (UX)
- 输入用户提供的数据
- 调用业务后端
- 聚合多种后端业务
- 验证用户身份
- 保留一定的数据, 方便下次使用





## 在 SDK 中配置 Javascript 前端

# Agent JS

<https://github.com/dfinity/agent-js/tree/main/packages>

包括以下程序库（可以直接从 npm 安装）

- @dfinity/agent
- @dfinity/identity
- @dfinity/candid
- @dfinity/principal
- @dfinity/authentication
- @dfinity/auth-client

开发者文档（包含如何进行底层消息发送、轮询等）：

<https://erxue-5aaaa-aaaab-qaagq-cai.raw.ic0.app>

# 创建后端 canister 的代理对象

直接引入：

```
import { hello } from "../../declarations/hello";
```

手动创建：

```
import { Actor, HttpAgent } from "@dfinity/agent";
import { idlFactory as hello_idl, canisterId as hello_id } from "../../declarations/hello";

let agent = new HttpAgent();
// agent.fetchRootKey(); // 仅在本地图试时需要从后端获取验证所需的公钥
let hello = Actor.createActor(hello_idl, { agent: agent, canisterId: hello_id });

// 异步调用后端方法
hello.greet("Boy").then((txt) => console.log(txt));
```

# 数据格式的对应关系

Motoko	Candid	Javascript
Nat, Int	nat, int	0, -1, 1, 1234567890n
Nat8, Int16, ...	nat8, int16, ...	0, 255, ...
Float	float64	1.34e13
Bool	bool	true, false
Text, Blob	text, blob	String, Uint8Array
[Nat]	[nat]	[0, 1, 1234567890n]
{ a = "AB"; b = false }	record { a = "AB"; b = false }	{ "a": "AB", "b": false }
#Ok(123)	variant { Ok = 123 }	{ "Ok": 123 }
?123, null	opt 123, null	[123], []

# 远程调用中可能发生的错误

一个对用户友好的前端必须能够正确处理错误并采取适当措施

- 网络连接失败
- 发送的消息不正确
- 远程调用返回错误
  - 方法执行时发生错误
  - 系统返回的错误
    - 过期时间设定有误
    - 访问的 canister 不存在
    - 处理消息的子网节点发生故障
  - 消息没有被处理（超时设定为 5 分钟）
- 轮询调用执行结果时的暂时错误等

# Query Call 状态不一致的情况

- Query call 不经过共识协议



- 执行 query call 的可能是任意一个子网节点



- 执行时所使用的 canister 状态并不一定是最新的

## 解决方案:

- 前端缓存所需的状态, 并检查 query call 返回的结果
- 使用 certified var / reply
- 改用 update call



动手做一个 microblog 前端

# 课程作业 (完善 microblog 前端)

1. 显示消息的发布者名字
  - a. 给 Message 增加 author 字段
  - b. 增加 set\_name 和 get\_name 公共方法
2. 增加以下新 UI
  - a. 显示目前 follow 的作者列表
  - b. 在 timeline 中显示消息作者和时间
3. 关注(follow)其它学员的 canister
  - a. 关注三个以上作者
  - b. 点击作者名字, 会显示对方发布的消息列表

