**分布式人工智能服务服务平台技术方案**

### 1、目前现状以及方案

目前人工智能服务框架基于算法组redis或restfull接口，结构分散，部署需要每个机器部署sdk，调用需要基于众多sdk服务器来编程，不好维护，且容易出各种问题。

从开发、功能角度讲，sdk接口功能少，接口太过于单一，无法实现灵活实时画框，等内容丰富的各种应用场景。目前技术栈过多，前端使用nodejs，后端Java，算法使用python，新上一个人工智能需求等于重做。

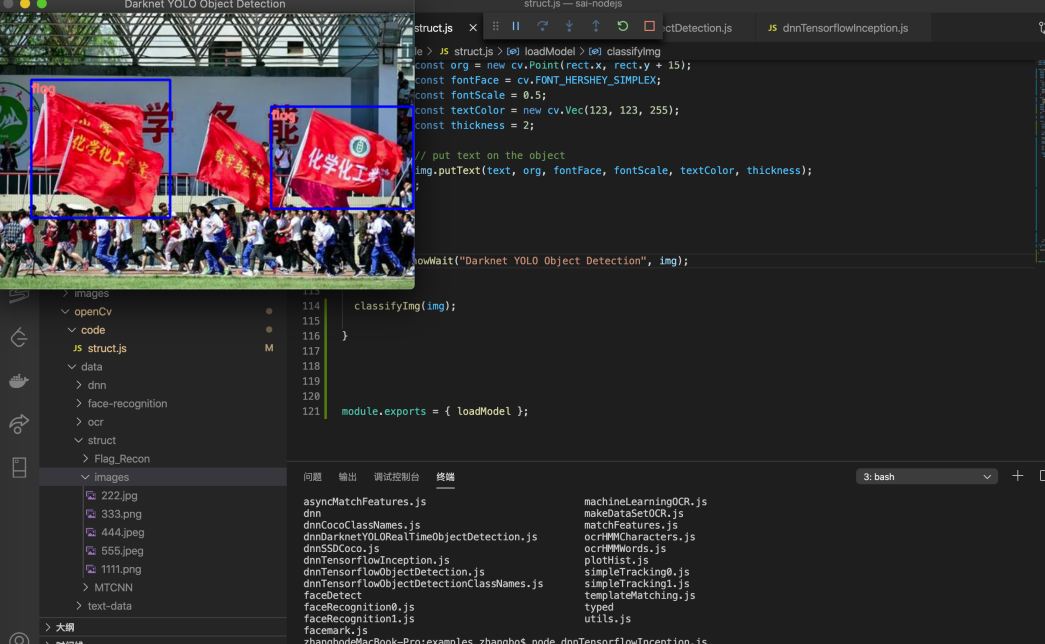
由此，建议使用nodejs+sofa-rpc+zookeeper+opencv技术栈。代码统一使用nodejs开发。

### **2、openCV使用**

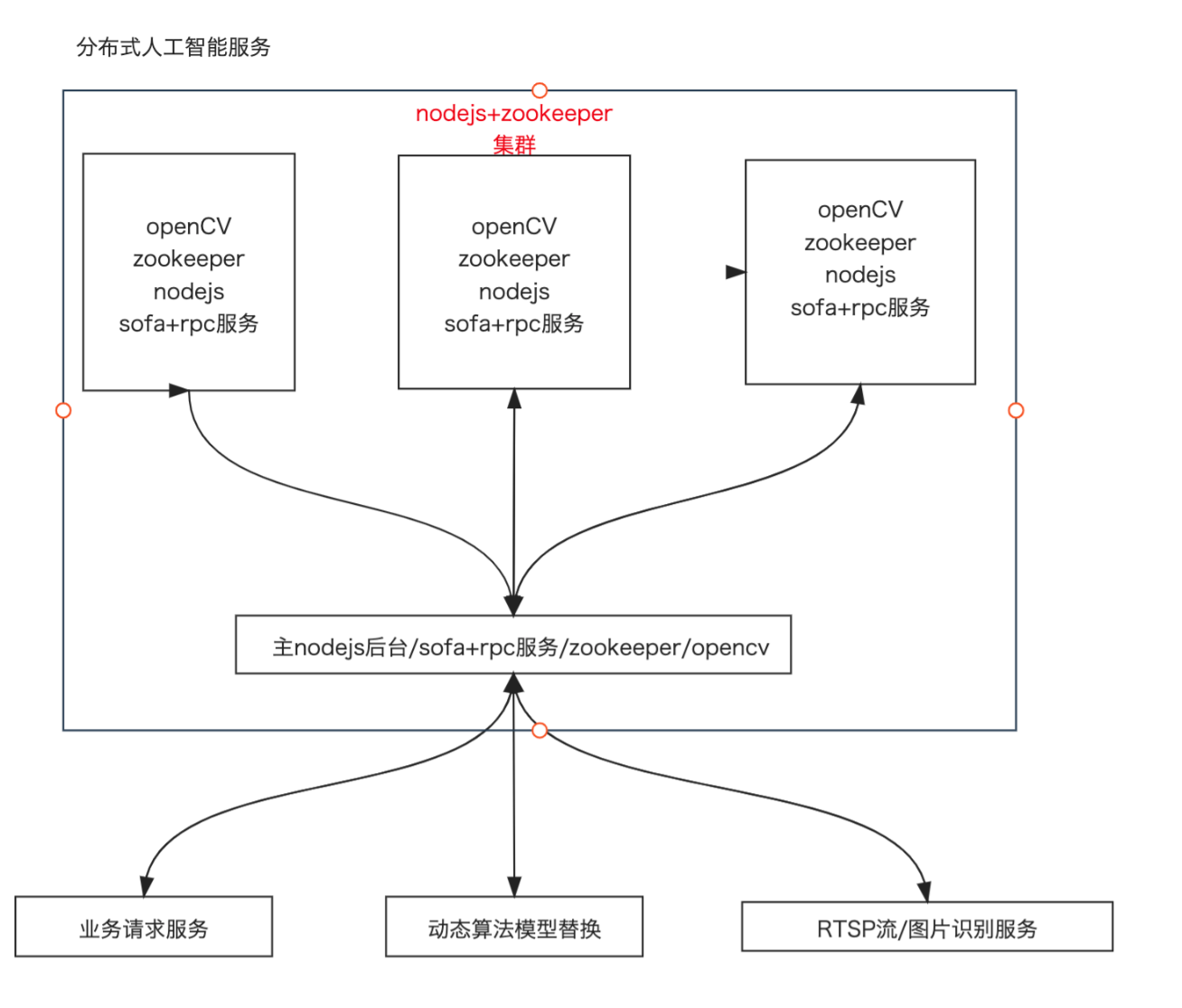
openCV是开源计算机视觉和机器学习软件库，算法团队也可基本基于此库训练、推理相关模型。 相关应用场景，用户使用的业务都是基于推理模型。

由于应用端所编写的目前应用都是基于推理模型，而nodejs与opencv交互性与python相差无几，使用其编写推理代码编写成本一致。所以可直接使用算法团队训练出来的模型推理代码。目前nodejs调用opencv并使用模型方法已经打通可用。

下面截图为nodejs代码调用阳博，对旗帜识别的训练模型的结果：



1. **分布式方案规划**



分布式方案应用主流分布式方案，使用zookeeper做到负载均衡，服务器通信使用阿里sofa-rpc方案，实现任务分解。

优势：

1. 前后端以及算法调用都由nodejs开发，技术栈统一。
2. 容易维护，代码统一部署，无需多个环节。
3. 整合算法团队代码，打通技术壁垒。
4. 根据后期持续项目使用，可迭代积累，算法、模型等资源。
5. 新项目需要人工智能服务开发代价小。
6. python业务逻辑不利于产品化、工程化。

劣势：

1. nodejs调用opencv，不如python广泛，资料比较少。
2. nodejs使用第三方库，非opencv官方库，安装且同 opencv编译较为麻烦。

开发规划：

1. nodejs+opencv框架搭建

作用：基于框架给算法团队编写推理代码指定适当位置。由此编写的推理模型代码可动态整合入系统。

1. sofa-rpc+zookeer 服务端，客户端代码编写

应用主流分布式方案，设计方便调用方式，以及扩展。

1. 业务端代码编写使用。

对于过去业务场景，人脸结构化，前端现有页面较多可不用开发，只做后台部分。可以纳入现有所有结构化人脸。各种功能需求需要规划。

根据具体使用，迭代优化系统结构。