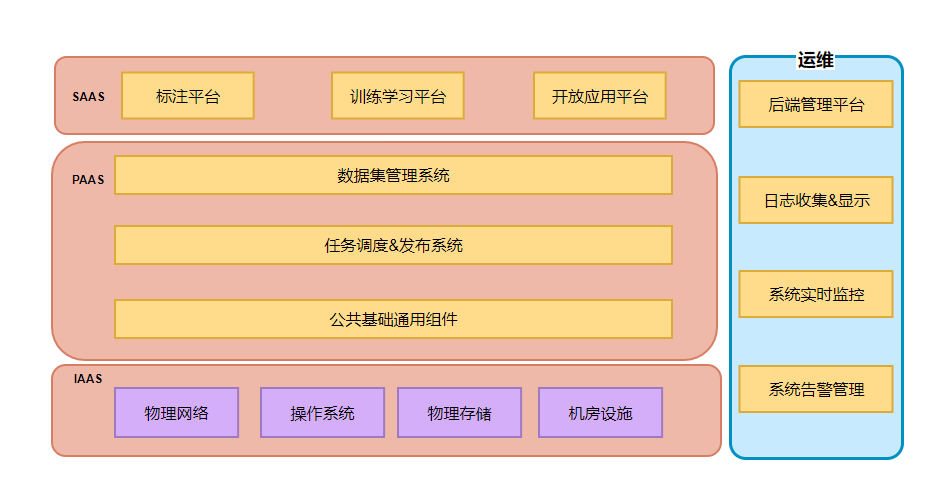
## Overview

需求定义： 参照产品设计文档

## 平台交付部署参考

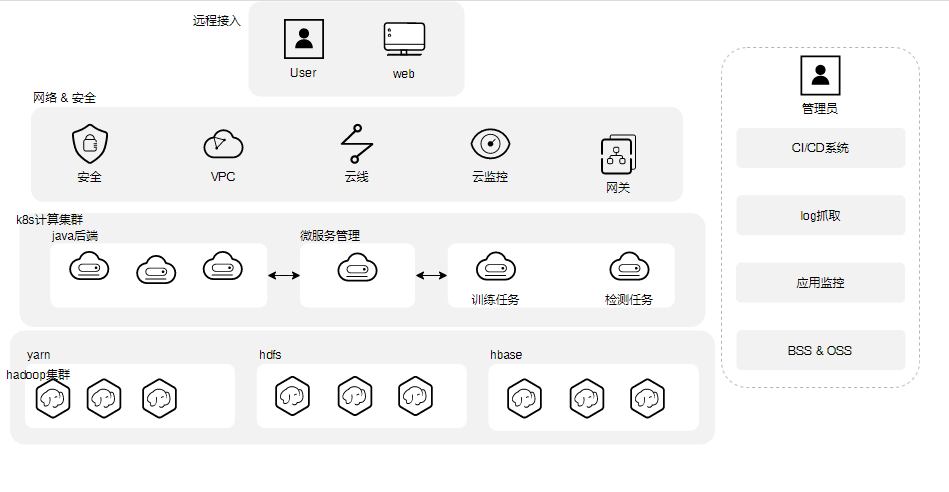
### 用户视图及简要说明



* SAAS层提供SAAS级别的云服务，用户通过SAAS的页面来操作平台
* PAAS的其他组件和SAAS层需要本司(深兰北京)交付
* IAAS 层资源需要用户方完整提供

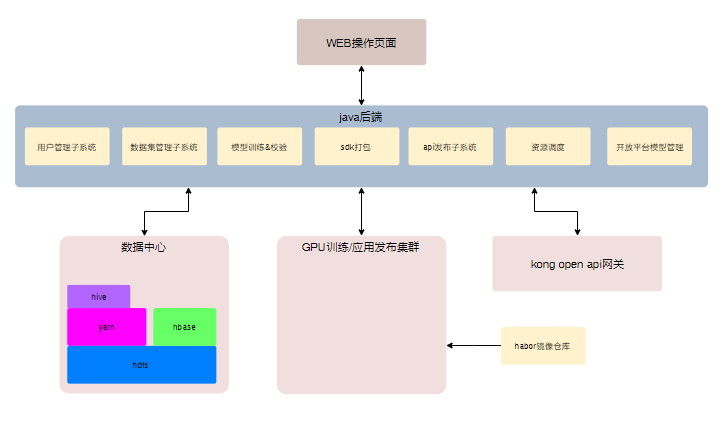
### 用户环境对接参考

* 物理视图



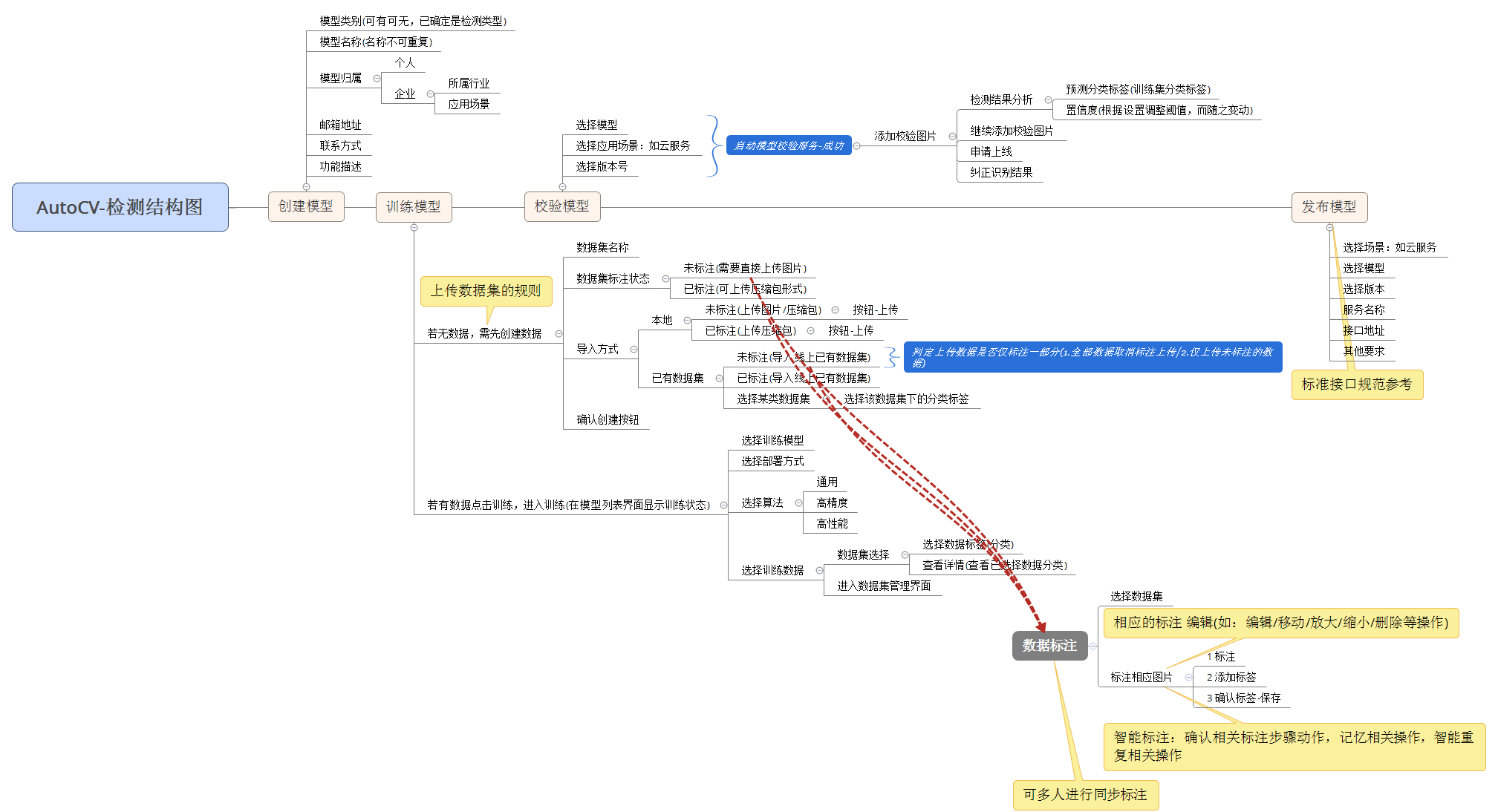
## 子系统功能定义

* 系统架构略图

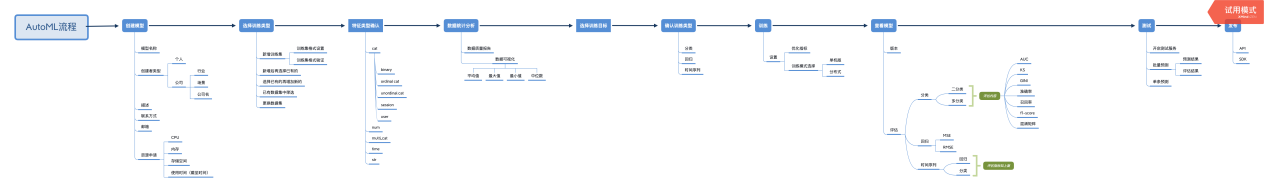


### WEB操作页面流程

* autocv的用户操作视图示例

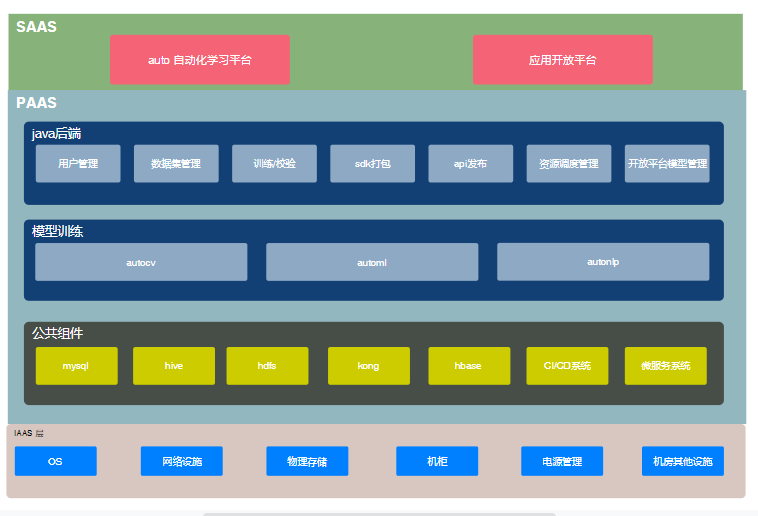


* automl的用户操作视图示例



### JAVA后端平台分解

#### 组件略图



#### 组件描述

##### Java后端

平台系统的控制核心，负责用户管理，数据集管理，训练任务管理，api发布，模型管理等功能。

##### 模型训练

目前适配有主流的cv，ml，nlp三个方向的训练任务

##### 公共组件

基础的开源组件

## 授权

### 标注平台

待定。

### 训练平台

待定。

### AI开放平台

一般意义按照用户调用接口次数收费，具体参照产品文档。

## 部署安装

以下内容是目前已经交付给用户的一份安装示例

管理口 | label | 备注

--------- | ----------- | -------

`10.16.101.38` | autotrain=true | `用于模型训练任务`

`10.16.101.39` | autotrain=true | `用于模型训练任务`

`10.16.101.48` | autotrain=true | `用于模型训练任务`

`10.16.101.49` | autotrain=true | `用于模型训练任务`

`10.16.101.58` | gpushare=true | `用于AI开放应用服务`

`10.16.101.59` | gpushare=true | `用于AI开放应用服务`

### kubernetes集群安装部署

#### 1） docker 安装

#### **2） nvidia docker2 安装，以及nvidia runtime配置**

#### **3） apt 源配置**

        deb http://mirrors.aliyun.com/ubuntu/ xenial main restricted

        deb http://mirrors.aliyun.com/ubuntu/ xenial-updates main restricted

        deb http://mirrors.aliyun.com/ubuntu/ xenial universe

        deb http://mirrors.aliyun.com/ubuntu/ xenial-updates universe

        deb http://mirrors.aliyun.com/ubuntu/ xenial multiverse

        deb http://mirrors.aliyun.com/ubuntu/ xenial-updates multiverse

        deb http://mirrors.aliyun.com/ubuntu/ xenial-backports main restricted universe  multiverse

        deb https://mirrors.aliyun.com/kubernetes/apt kubernetes-xenial main

#### **4）Install kubelet kubeadm kubectl组件**

        apt-get update &&  \

        apt-get install -y kubelet kubeadm kubectl --allow-unauthenticated

#### **5） kubeadm add master**

        kubeadm init --pod-network-cidr=10.244.0.0/16 --apiserver-advertise-address=10.16.32.49

        kubeadm token create

        kubeadm token list

    获取ca证书sha256编码hash值

    openssl x509 -pubkey -in /etc/kubernetes/pki/ca.crt | openssl rsa -pubin -outform der 2>/dev/null | openssl dgst -sha256 -hex | sed 's/^.\* //'

#### **6） 加入worker 节点**

    kubeadm join 192.168.0.200:6443 --token rw4enn.mvk547juq7qi2b5f --discovery-token-ca-cert-hash sha256:ba260d5191213382a806a9a7d92c9e6bb09061847c7914b1ac584d0c69471579

#### **7） 选择网络插件**

     参照： https://kubernetes.io/docs/concepts/cluster-administration/addons/

### 调度器定制化部署

#### 1）nvidia默认调度器部署

        kubectl apply -f nvidia-device-plugin.yml

    note： 需要提前再需要调度的nvidia机器上打上标签：

        kubectl label nodes 10.2.2.123 gpushare=false

#### 2）gpu共享调度器部署

    note： 需要提前再需要调度的nvidia机器上打上标签：

        kubectl label nodes 10.2.2.123 gpushare=true

        1. 替换 scheduler-policy-config.json kube-scheduler.yaml 两个文件

        2. Install ali scheduler ： https://github.com/AliyunContainerService/gpushare-scheduler-extender/blob/master/docs/install.md

        3. Install ali device plugin： https://github.com/AliyunContainerService/gpushare-scheduler-extender/blob/master/docs/install.md

### Hadoop数据存储

\* 安装脚本： http://203.195.212.206/hadoop/hadoop\_install.sh

#### 存储hdfs

    启动namenode： ${HADOOP\_HOME}/sbin/hadoop-daemon.sh start namenode

    启动datanode   ${HADOOP\_HOME}/sbin/hadoop-daemon.sh start datanode

#### 存储hbase

    一键启动脚本  ${HBASE\_HOME}/bin/start-hbase.sh

### 对外API网关

#### 1. download rpm package

    https://docs.konghq.com/install/centos/

#### 2. Install rpm package

```shell

 $  sudo yum install epel-release

 $  sudo yum install kong-2.0.3.\*.noarch.rpm --nogpgcheck

```

#### 3. create your dabase for your kong

```shell

    CREATE USER kong; CREATE DATABASE kong OWNER kong;

    $ kong migrations bootstrap [-c /path/to/kong.conf]

```

#### 4. start your kong gateway

```shell

    kong start [-c /path/to/kong.conf]

```

### 后端服务&前端页面程序安装

参照git地址提供的安装脚本： http://10.16.32.59:30001/root/auto-deploy.git