

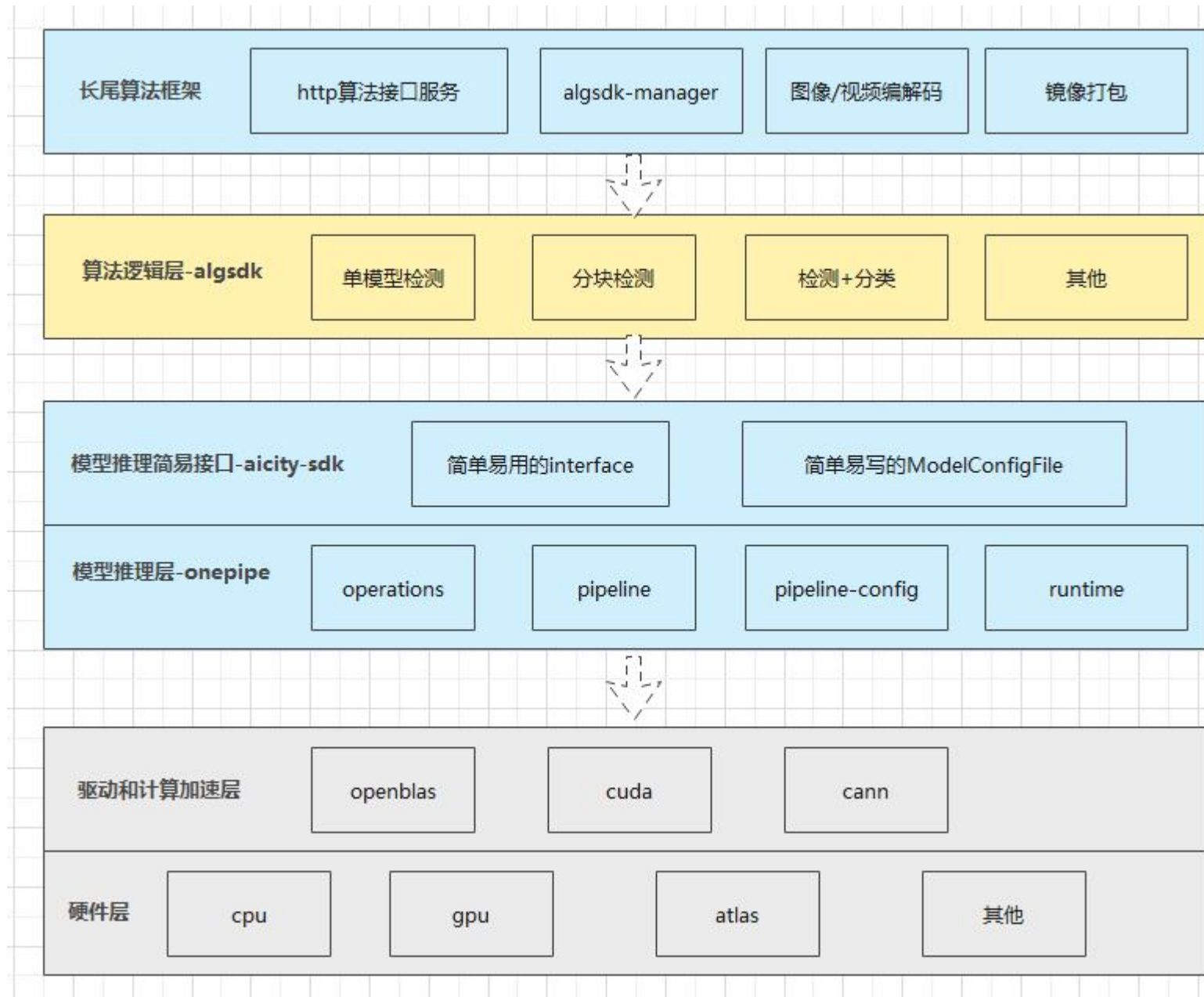
算法sdk+服务框架

高增辉-2021/10/15

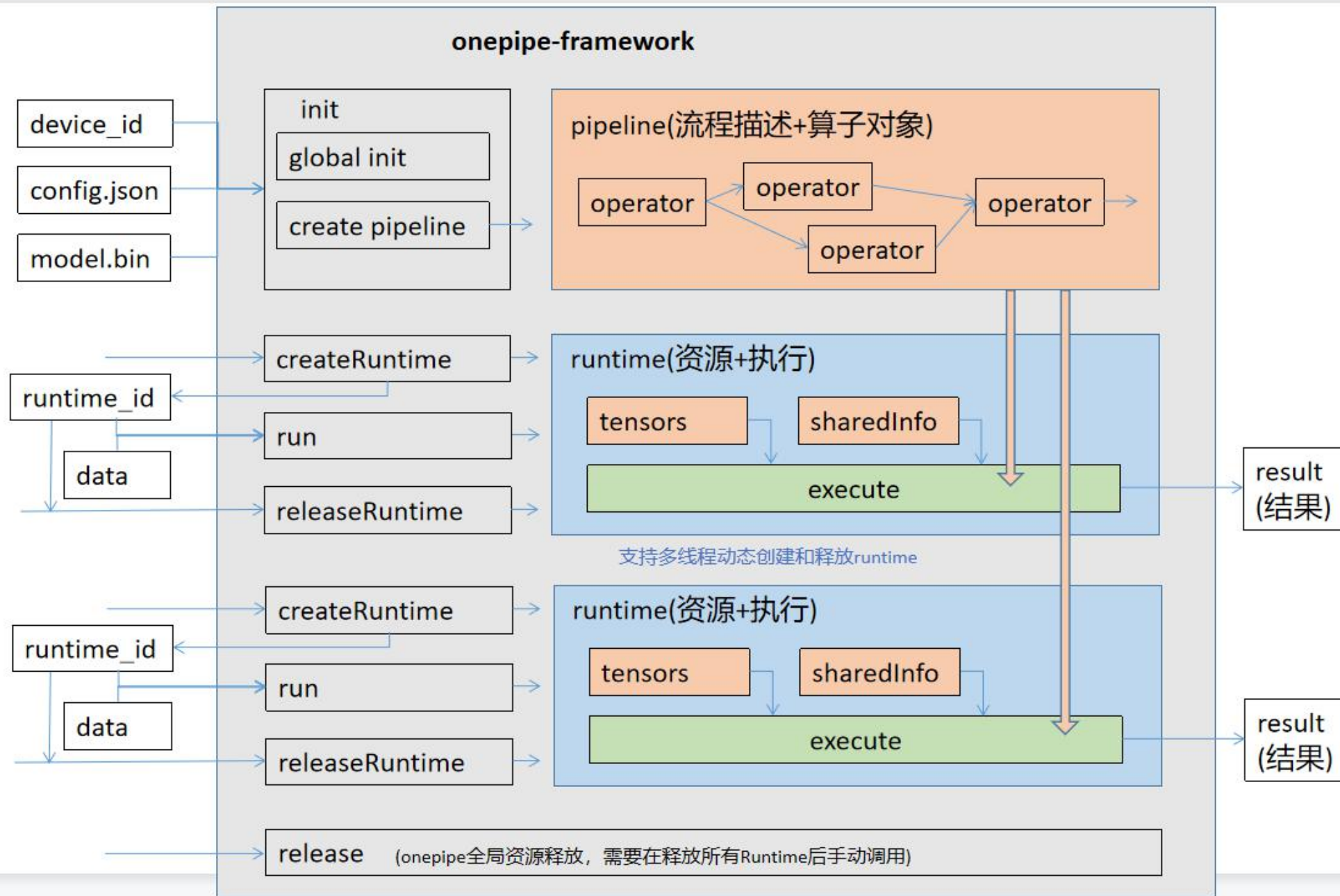
目录

- 整体架构
- onepipe
- aicitysdk
 - 接口使用
 - 模型配置文件编写
- algsdk
- LongTailFramework
- 新模型发布流程
- 一个例子

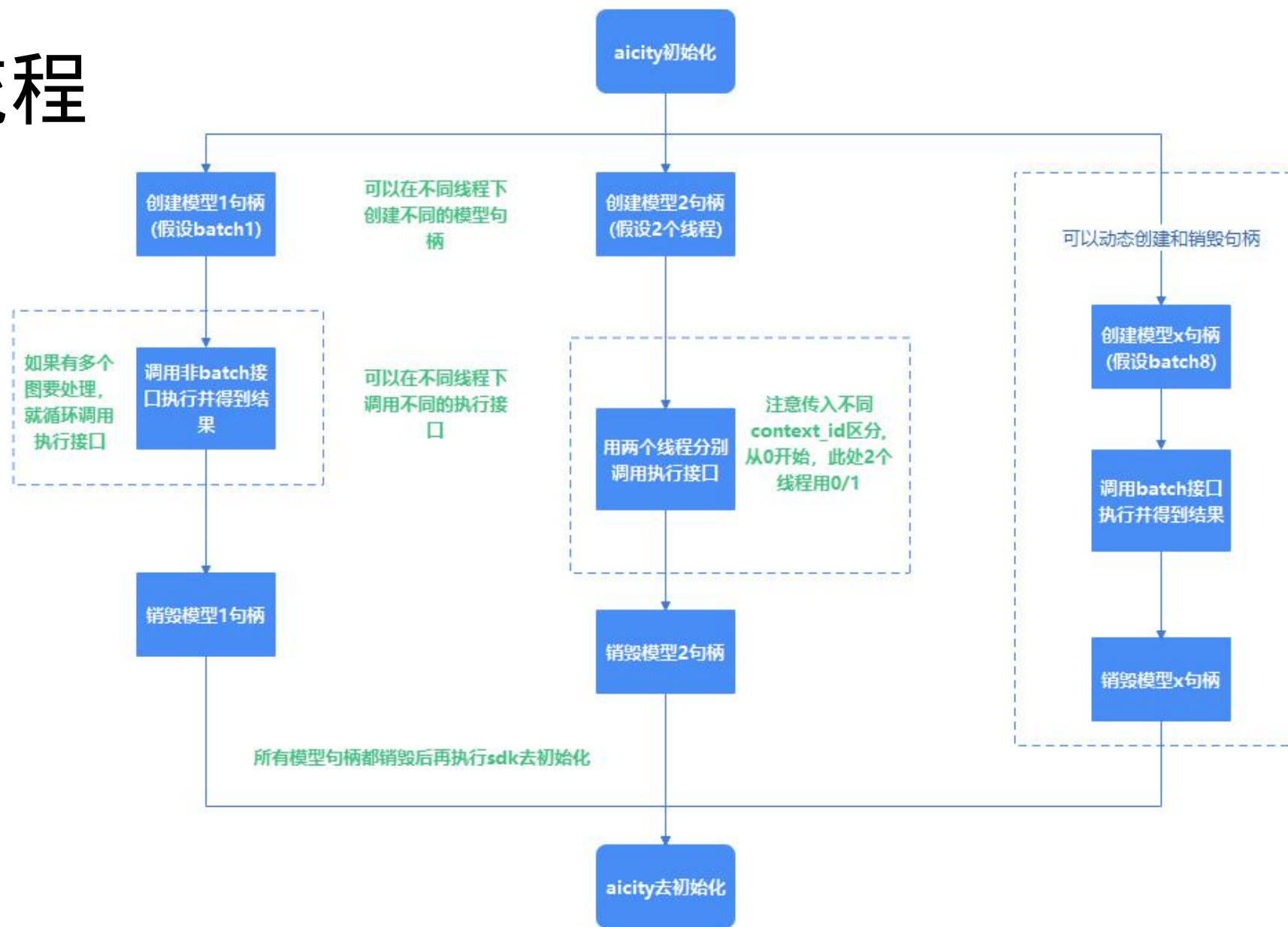
整体架构



onепipe



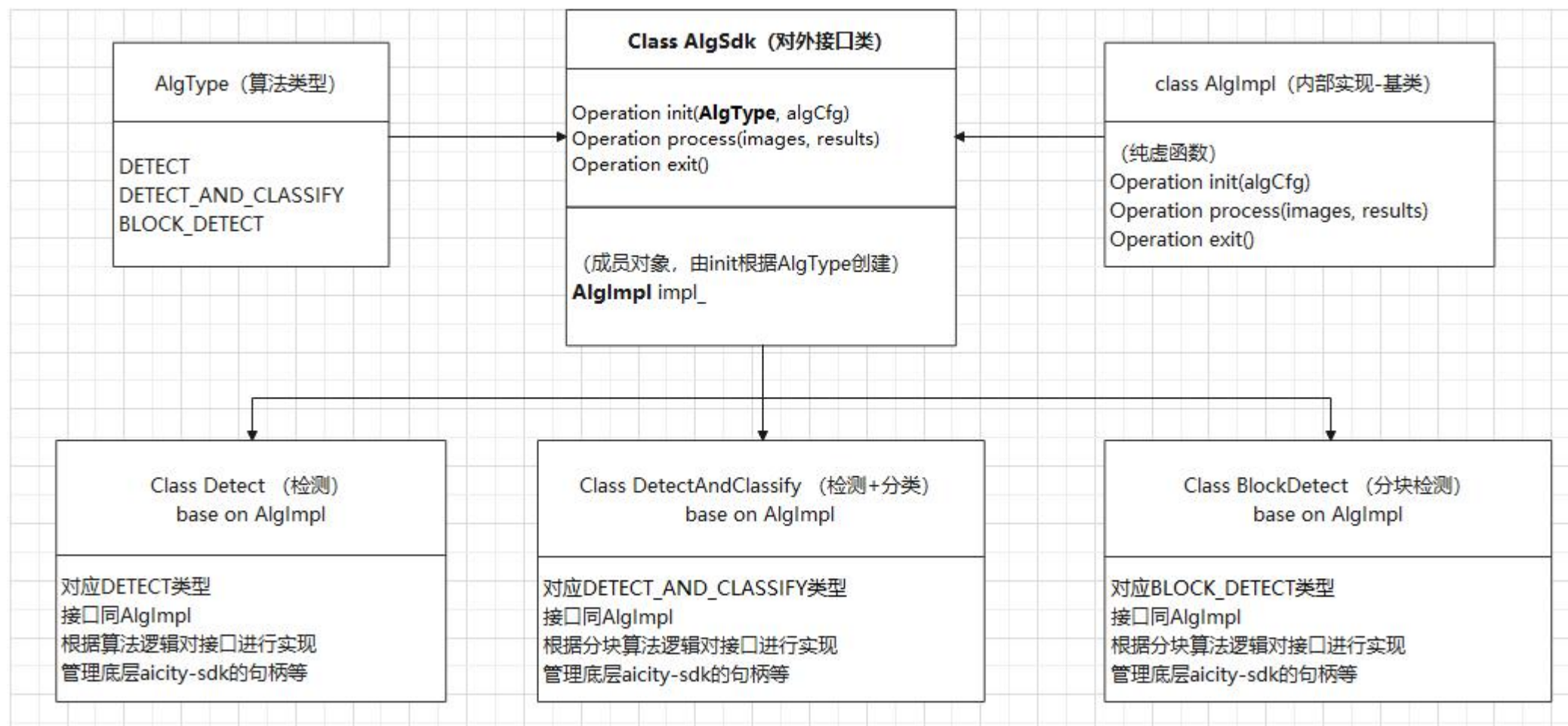
aicity-sdk 接口调用流程



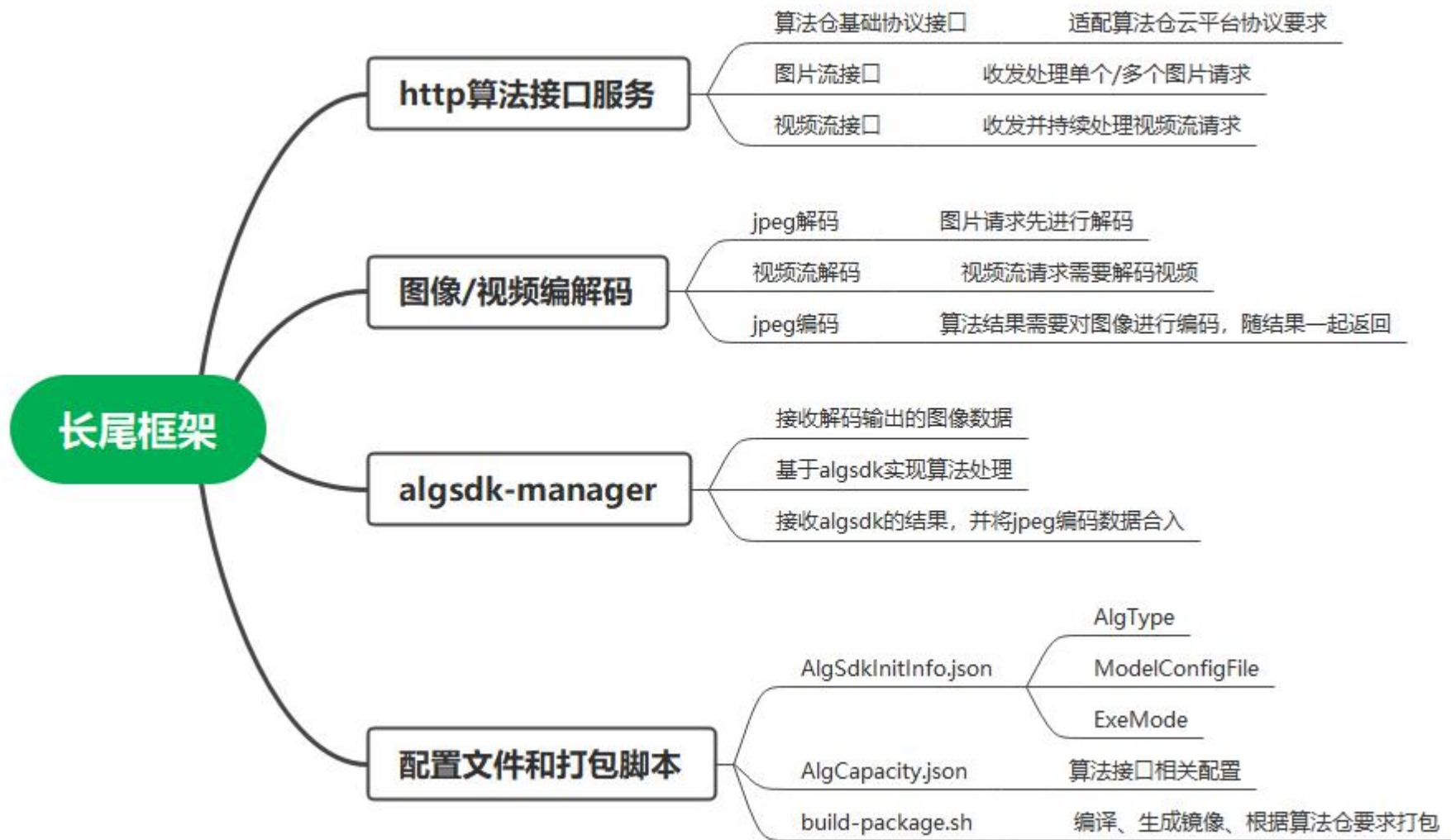
aicity-sdk 配置文件编写

```
{
  "model_file": "kitchen-uniform-det-acl-b1-0.0.1.bin",
  "task_type": "DETECT",
  "model_format": "ACL",
  "class_names": ["kitchen_hat", "non_kitchen_hat", "kitchen_uniform", "non_kitchen_uniform"],
  "class_num": 4,
  "batch_size": 1,
  "channel_num": 3,
  "image_height": 640,
  "image_width": 640,
  "preprocess": {
    "operator": "PreResize",
    "padding_type": "LEFT_TOP"
  },
  "model_infer": {
    "operator": "InferAcl",
    "output_num": 3
  },
  "postprocess": {
    "operator": "PostDetectResultYolov5",
    "conf_thresh": 0.3,
    "nms_thresh": 0.35,
    "anchors": [15, 18, 26, 31, 39, 47, 41, 93, 63, 74, 63, 142, 108, 133, 93, 209, 138, 247]
  }
}
```

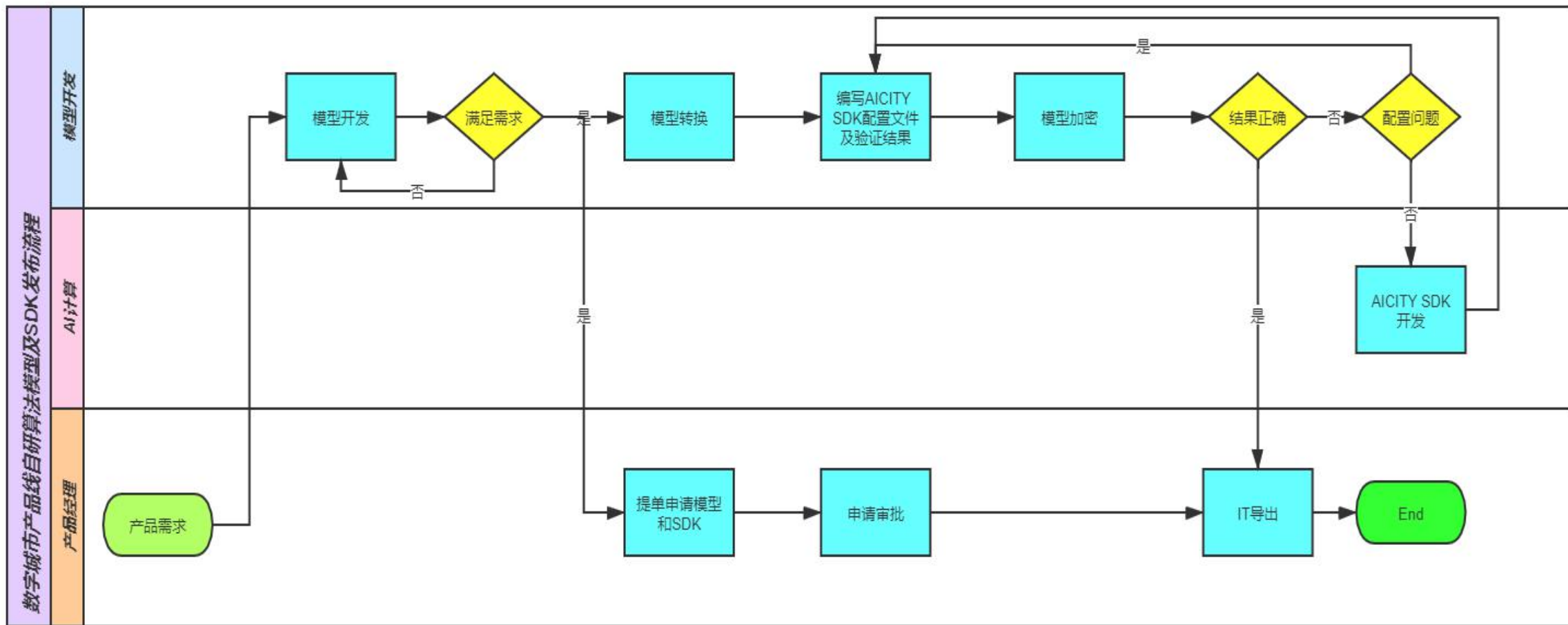
algsdk



LongTailFramework



新模型发布流程



一个例子——检测一只狗

1. 准备算法模型

dog_batch1_0.0.1.om

2. 模型打包

dog-det-acl-b1-0.0.1.bin.plan

3. 模型加密

dog-det-acl-b1-0.0.1.bin

实际发布模型需要在内网对模型加密

4. 编写模型配置文件

dog-det-acl-b1-0.0.1.json

5. 编写algsdk配置文件

AlgSdkCfg-dog-det.json

6. 编译测试algsdk

修改test + build-r.sh + run-detect.sh

长尾算法框架修改模型目前涉及的改动

```
docker/
├── algImage
│   └── algorithm
│       └── package.json 【1. image:修改镜像名, 要跟[2]中的docker build的镜像名对应一致】
├── make
│   └── servable-infer
│       ├── build.sh 【2. 修改镜像名, 避免多种模型镜像冲突】
│       └── Dockerfile
└── setup.sh 【3. 修改镜像名, 同[2], 和docker save的包名, 和最后tar的包名】

servable/3rdparty/
├── include
│   └── algsdk 【4. 如有改动更新algsdk头文件】
├── lib
│   └── algsdk 【5. 如有改动更新algsdk库】
└── servable/model/ 【6. 替换新的模型和配置文件】
    ├── xxx.bin
    └── xxx.json

servable/config/
├── AlgBaseInfo.json
├── AlgCapacity.json 【7. CapacityInfo: Uri, 根据[8]中的算法名修改】
└── AlgSdkInitInfo.json 【8. ModelConfigFile等, 对应6中的更新】

servable/src/
├── servableMain.h 【9. g_cityManagementAlgTypePool和g_algIndex, 如果是现有算法, 直接在g_algIndex配置对应的编号, 如果要新增算法, 跟算法仓确认具体算法名后添加到g_cityManagementAlgTypePool, 并在g_algIndex指定对应编号】
```

一个例子——检测一只狗

编译和打包服务框架

```
cd 长尾框架工程根目录  
sudo bash build-package.sh
```

测试服务框架镜像

```
docker run --rm -it -e ASCEND_VISIBLE_DEVICES=0 -p 25040:25040 -e  
HEATBEAT_CALLBACK_IP=localhost -e HEATBEAT_CALLBACK_PORT=25041  
-e HEATBEAT_CALLBACK_URL=/debug/test_heatbeat -e  
HEATBEAT_CALLBACK_TYPE=1 -e HEATBEAT_CALLBACK_RATE=5 -e  
REGISTER_CALLBACK_IP=localhost -e REGISTER_CALLBACK_PORT=25041  
-e REGISTER_CALLBACK_URL=/debug/test_register -e  
REGISTER_CALLBACK_TYPE=1  
intellif.io/algorithm/intellif_citymanagement_dog_armv8_kunpeng_atlas30  
0_cann5.0.1:v0.1
```

```
post http://192.168.13.94:25040/alg/cv/image/cityManagement/dogDetect/1.0 带body
```

其他

开发服务器

192.168.13.94 arm+atlas服务器

模型打包工具

13.94:/home/intellif/algsdk-dev/tools/model_pack_tool

模型加密工具（外网的仅用于测试）

13.94:/home/intellif/algsdk-dev/tools/model_encrypt_tool

谢谢！