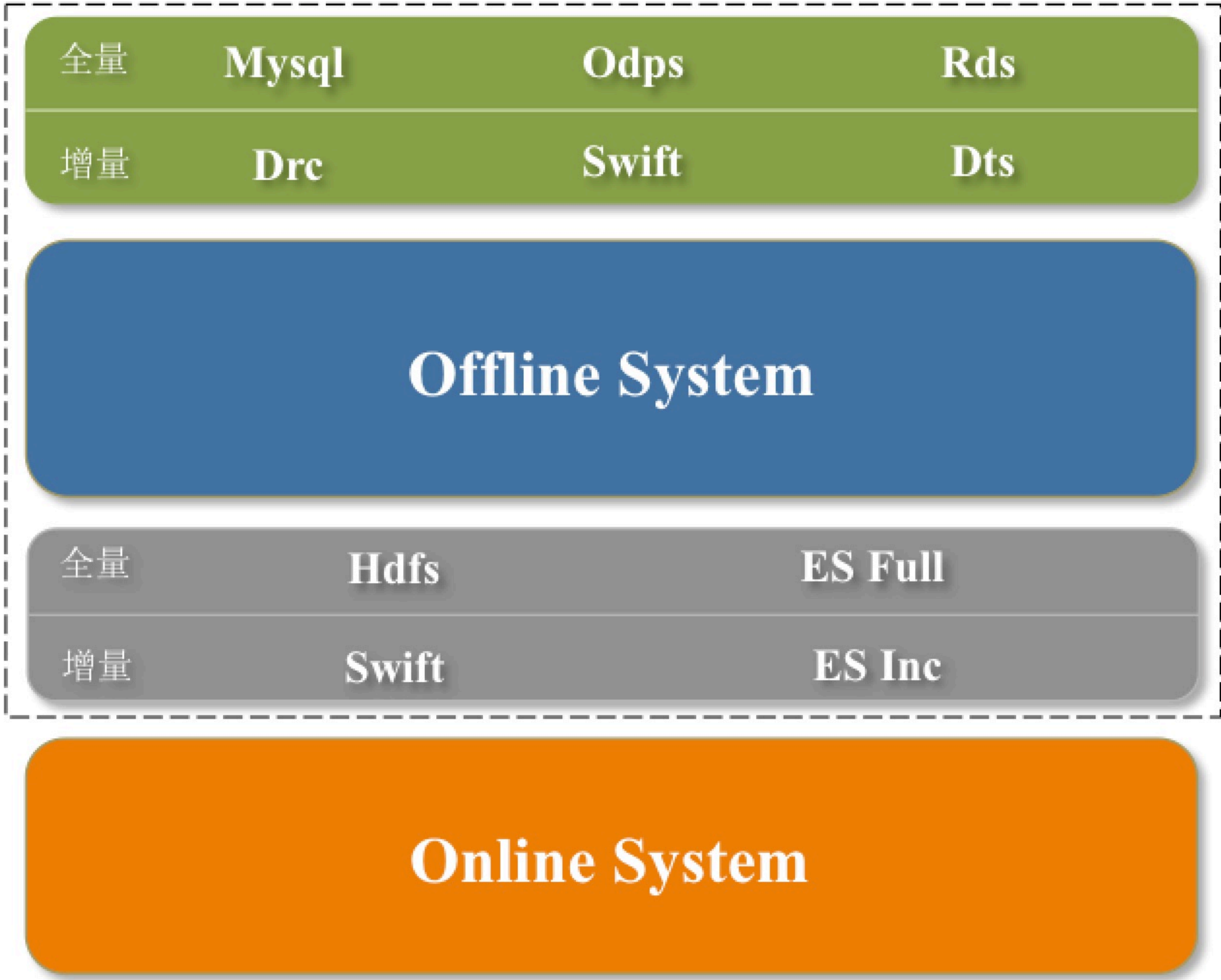


FY21 实习阶段性Review

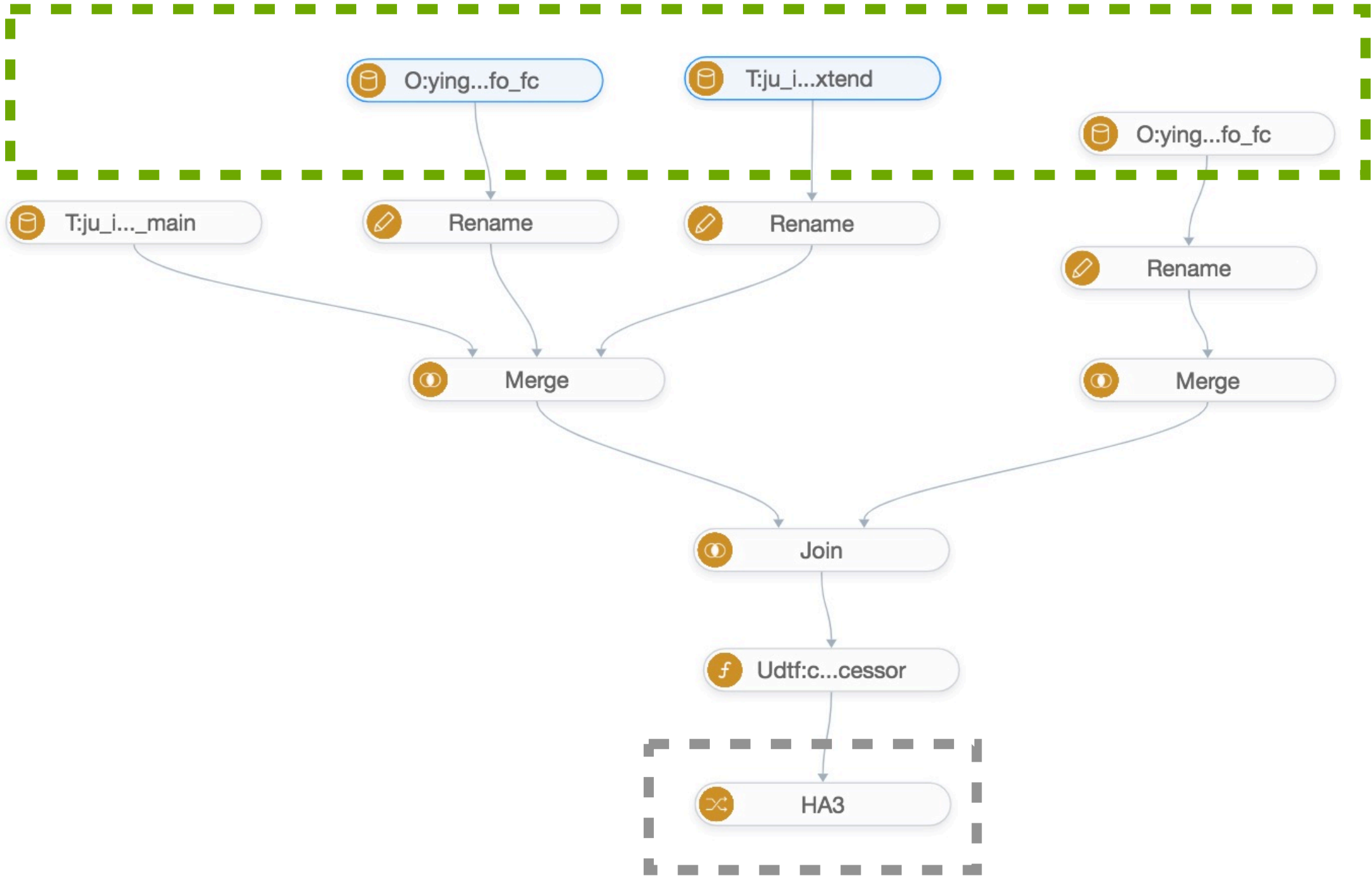
王沁馨

2020.7.19

搜索离线 - SARO团队



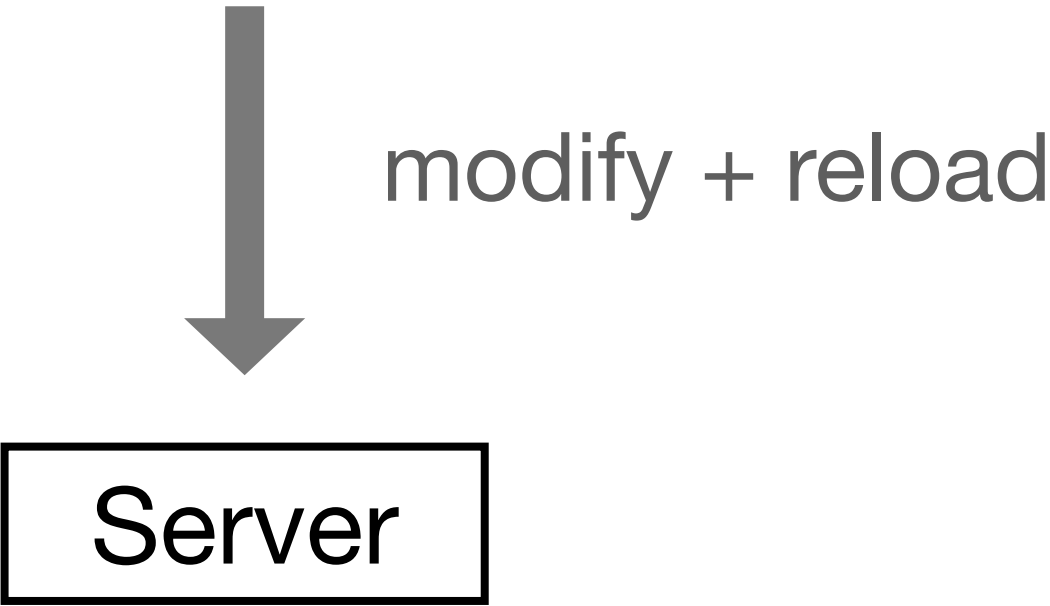
Business Graph



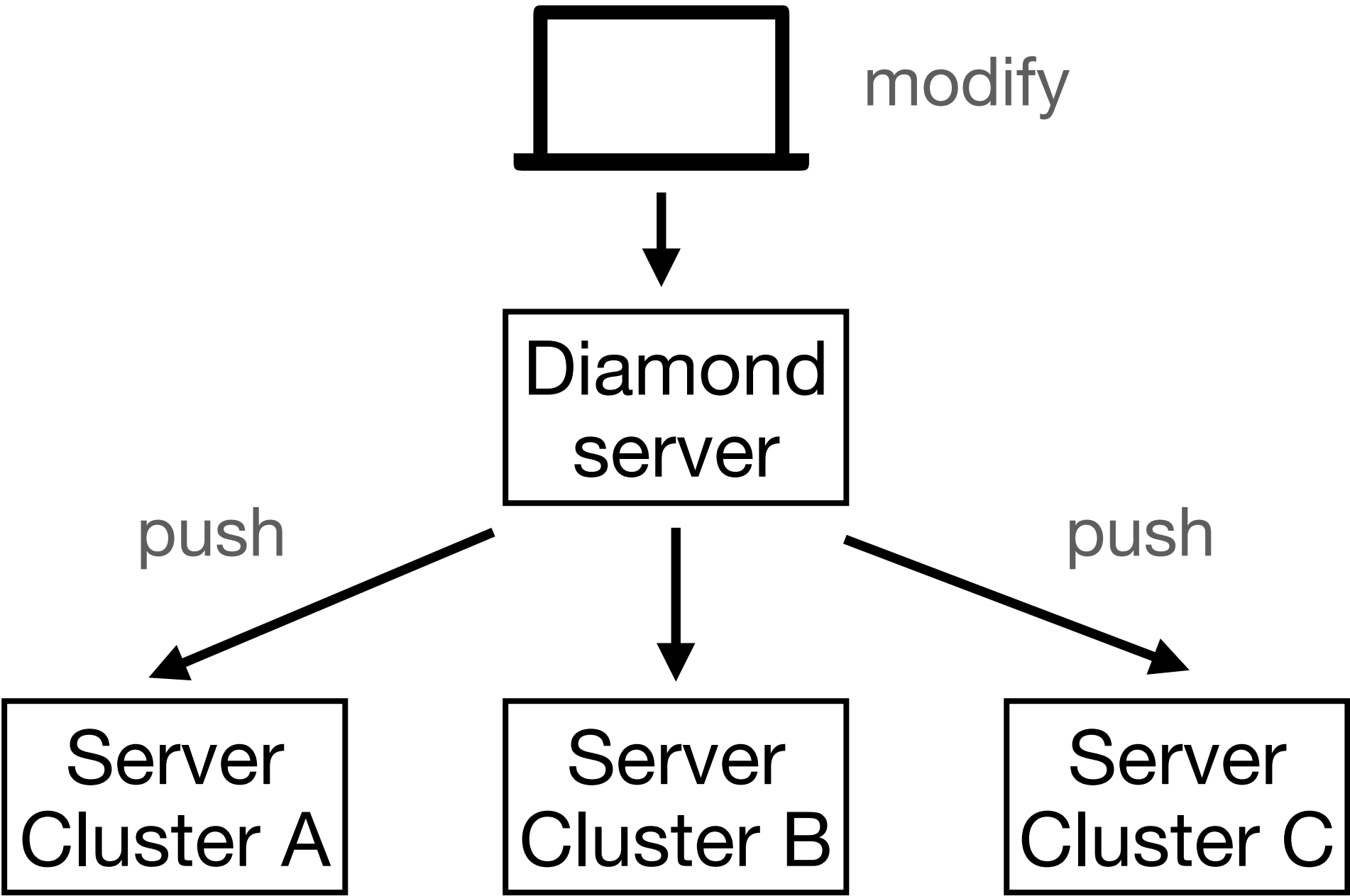
配置中心与Diamond

config.properties

Key1 = config1
Key2 = config2
...



Diamond



Diamond Config重构

diamond配置类别

- BahamutConfig
- BahamutEnvConfig
- ExternalSourceConfig



Global

Global
white/black
list

App customized
Job resource

App customized
Runtime Param

```
{
  "appVersionOssPath": "oss://appVersionOssPath",
  "bizVersionOssPath": "http://bahamutzmf.oss-cn-hangzhou-
zmf.aliyuncs.com",
  "bizVersion": "ctrace/pkg/bahamut-
biz-5b1b1ed4-226090-20200701.tar.gz",
  "flowVersion": "{ \"publish\": \"Publish\", \"build\": \"Build\", \"UDPDeduction\":
\\\"UDPDeduction\\\", \"ResourceRecycling\": \"ResourceRecycling\" }",
  "obsoleteAppListWithSwiftZkAddress": [List],
  "disableBlinkAutoScaleApps": [List],
  "personaliseResourceConfig": {
    "3499090057": {
      "IncJoin": "directMemory:64"
    },
    "10000000098": {
      "IncDynamicSync": "directMemory:256"
    },
    ....}
  "personaliseRuntimeParamConfig": {
    "10000002682": {
      "IncDailyDrive": {
        "runtimeParam": {
          "incDriveStartPoint": "BEFORE_SYNC"
        }
      },
      "IncJoin": {
        "runtimeParam": {
          "windowEnabled": true,
          "windowInSecond": 1
        }
      },
      .....}
  }
```

Data ID:	bahamutConfig
Group ID:	bahamut
归属应用:	bahamut-service
所属环境:	线上中心
beta IP:	
md5:	0ab3f10d85a516ed995370e8c56f4711
配置内容:	<div><pre>{ "appVersionOssPath": "oss://appVersionOssPath", "bizVersionOssPath": "http://bahamutzmf.oss-cn-hangzhou-zmf.aliyuncs.com", "bizVersion": "ctrace/pkg/bahamut-biz-5b1b1ed4-226090-20200701.tar.gz", "bizConfigVersion": "config-562d4470-qinxinwang-20200716.tar.gz", "flowVersion": "{ \"publish\": \"Publish\", \"build\": \"Build\", \"UDPDeduction\": \"UDPDeduction\", \"ResourceRecycling\": \"ResourceRecycling\" }", "obsoleteAppListWithSwiftZkAddress": [3410446179, 3813292638, 10000000730, 10000000556, 10000000584, 10000000642], "disableBlinkAutoScaleApps": [10000000011] }</pre></div>

Diamond Config重构

问题

1. BahamutConfig配置项冗长，包含全局配置和app层面的个性化配置
2. diamond中更改配置项不便，出错会导致整个配置文件不可用
3. 配置历史版本管理不便

需求

方便、快速的推送

历史版本管理&回滚

增删、修改字段方便

配置强校验

与biz多版本兼容

Diamond Config重构

配置方案

- 使用与diamond类似的配置中心；
- 将配置保存在数据库中；
- 用git进行文件管理 api ✕
 biz ✓

优势	劣势
有完善的管理方案、发布流程	部署、维护需要成本 不能完全满足个性化需求
以服务形式从前端交互，更改简单 热更新	实现比较复杂 版本管理不易实现
强大的版本管理； 易用，学习成本低	动态配置无法热更新 需要自己搭建推送流程

Diamond Config重构

Prove of concept

- 1. 服务器上如何获取从gitlab获取配置包？
- 2. 配置包如何自动化上传？
- 3. 如何指定配置的版本？

build-deploy-config

29秒

(4/4) 任务4-build-and-deploy

已完成

运行日志



Diamond Config重构

Prove of concept

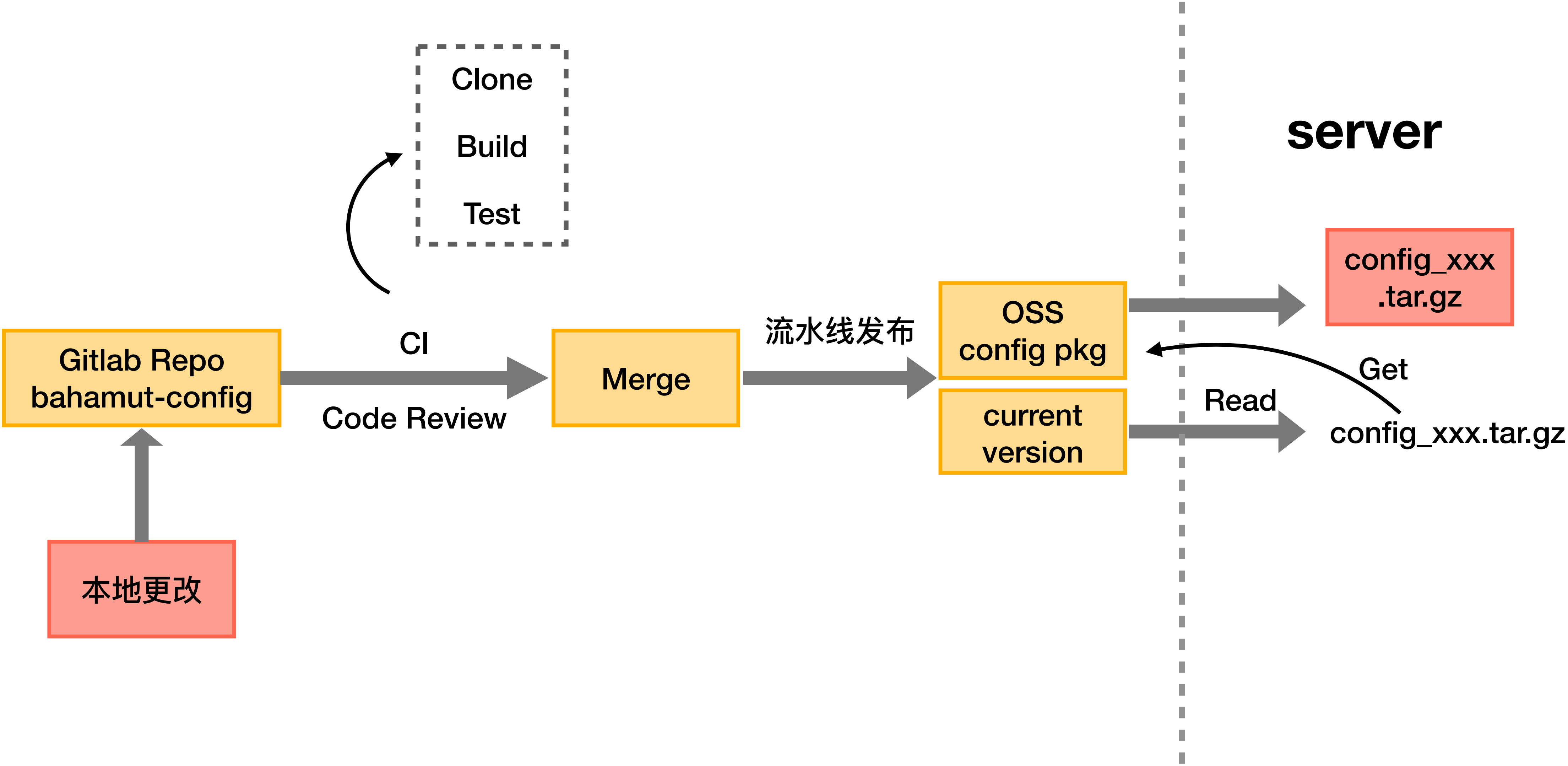
4. config是与代码放在同一个repo里还是分开存放?
5. 如何对config的内容进行校验?

Configuration as code

Configuration as code (CAC) is managing configuration resources in your source repository. You treat your application config resources as versioned artifacts.

Diamond Config重构

发布流程



Diamond Config重构

基于文件的config组织形式

按模块层次划分

bahamut-config.git

```
config/  
  global_config.json  
  env/  
    na61.json  
    ...  
  app/  
    100000231.json  
    ...
```

```
{  
  "INC_JOIN": {  
    "RuntimeParam": {...},  
    "BlinkJobOption": {...},  
    "ResourceConfig": {...}  
  },  
  "FULL_SYNC": {  
    "RuntimeParam": {...},  
    "BlinkJobOption": {...},  
    "ResourceConfig": {...}  
  },  
  ...  
}
```

读取

Class BahamutAppConfig

Class RuntimeParam

Class BlinkJobOption

Class ResourceConfig

Diamond Config重构

config调用

固定接口：

`configProvider.getGlobalConfig()`

`.getEnvConfig(env)`

`.getAppConfig(app)`

`.getJobConfig(app, job)`

可扩展：

`configProvider.get(Pojo.class)`

e.g.

`configProvider.get(RuntimeJobConfig.class)`

—> return RuntimeJobConfig rjc



Return POJOs

Diamond Config重构

config校验

开发者增加配置字段时，需要很方便地设置相应的校验规则。

- Use annotation:

```
@Max(value=120)
```

```
@Min(value=0)
```

```
Int windowInSeconds;
```

```
@Pattern(regexp = "[a-zA-Z_]+\\:[0-9]+")
```

```
@Numeric
```

```
...
```

遍历每个文件，检查文件格式和内容合法性

Config repo中只存放配置文件，把校验逻辑代码放在代码repo中

对比

	Diamond	基于文件的Config
存储	在diamond中用单个文件存储所有app配置	以文件形式分别存储每个app的配置
读取	提供group id, data id, 从diamond接口中读取	下载config包, 用provider接口从文件中读取
修改	在diamond中修改, 不便于编辑、查找	在ide中修改, 有语法检查和高亮
发布	通过diamond发布	通过流水线上传OSS, 加入合法性检查

迁移

- 先上线pipeline，再逐步替换使用点
- 迁移前进行Consistency test
- config文件中逐渐增加字段，diamond中逐渐删除

谢谢！