

众安国际寿险基于 Apache Doris 的实时数仓实践与高可靠探索

沈健

众安国际 寿险资深研发工程师



目录

01 背景介绍

02 架构演进

03 实践与探索

04 总结与未来展望

众安国际 寿险 (ZA Insure) 业务介绍

ZA Insure 是香港保险业监管局授权于香港运营的全数字化保险公司

公司在香港提供在线保险服务，主要为香港居民提供 人寿保，癌症保，自愿医疗，储蓄保等众多保险产品
是目前在香港运营的保费收入第一的虚拟保险公司。

ZA Insure
香港第一虚拟保险

總保費收入最高，是香港人最信任的虚拟保险

產品 保單索償 我的保單 幫助 登錄/註冊

真正數碼化 人人買得起

ZA Insure 技术背景

ZA Insure 依托于公司内部保险基线开发，部署在混合云上，有独立的技术团队维护全套的业务技术流程。

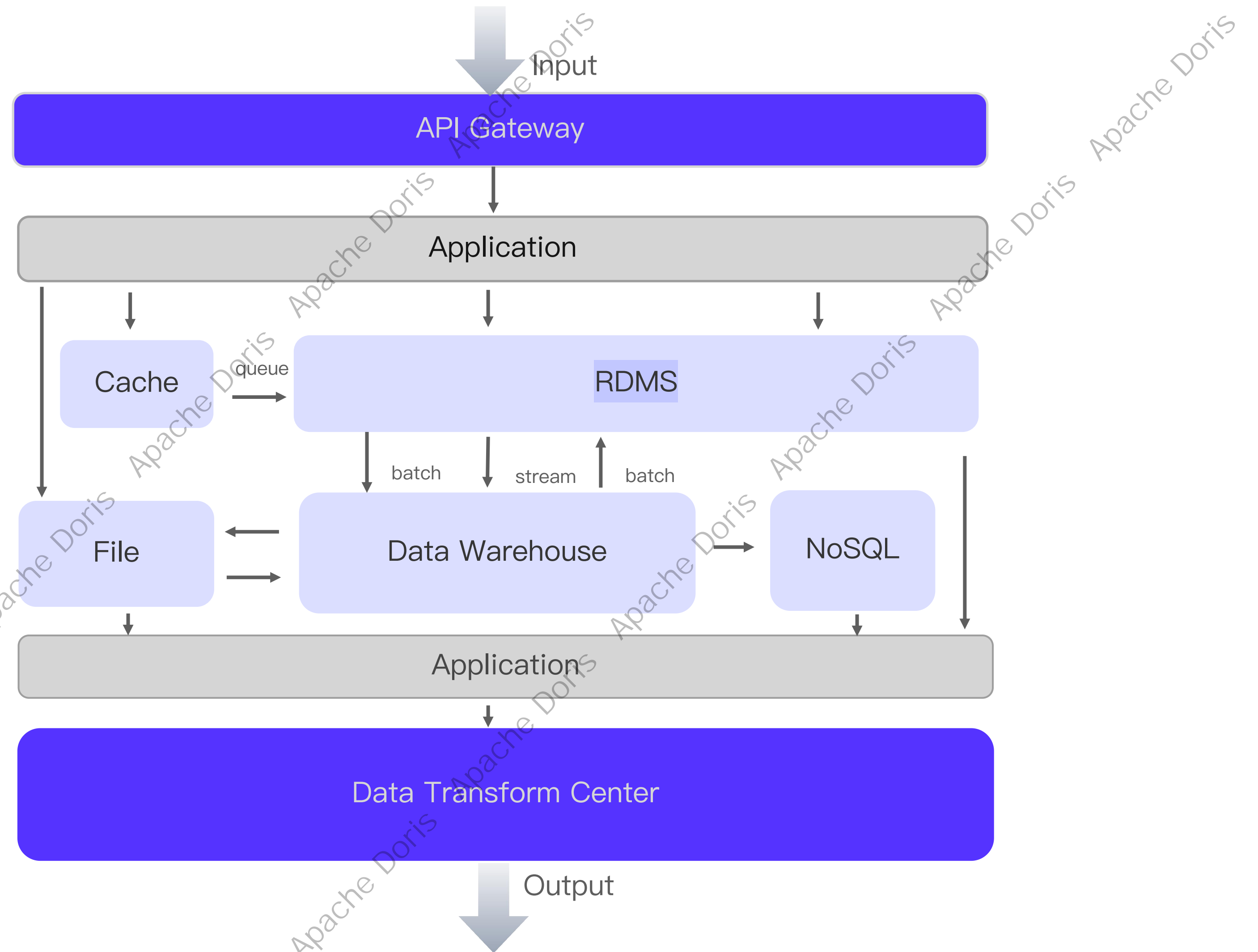
金融公司本质上是数据公司。

系统架构整体的设计方法论是：**业务的运行就是数据的流动。**

ZA Insure 技术背景

数据的流动与分布

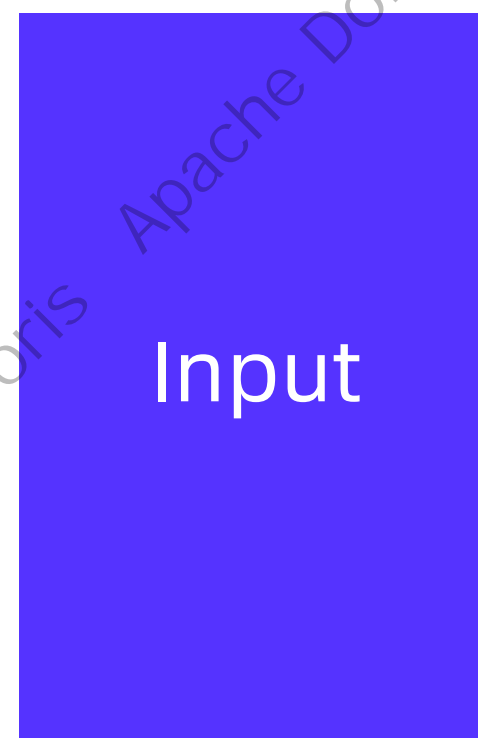
- 降低微服务架构下数据流动的成本
- 数据的标准化管理
- 构建完整的数据模型



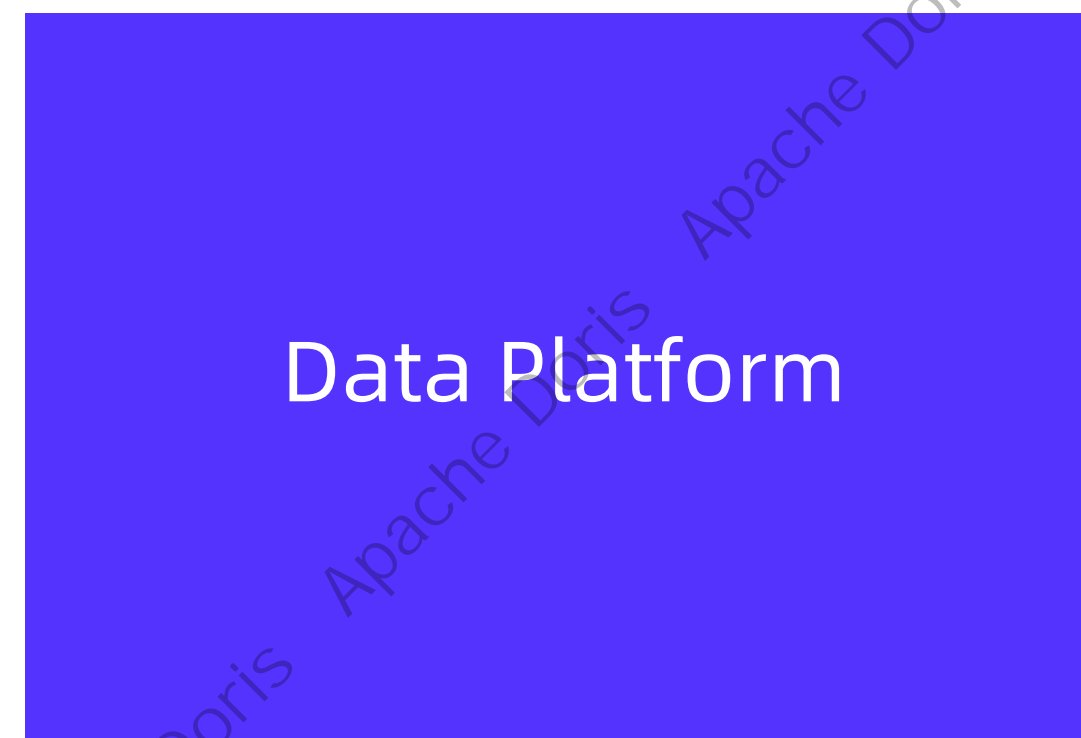
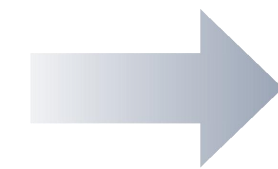
ZA Insure 技术背景

数据异构的应用开发

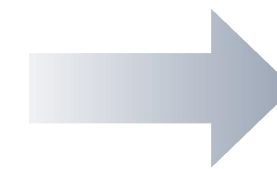
1. 应用层做 '薄'
2. 微服务只做数据的基础处理
3. 业务流程异步化



1. 输入校验
2. 数据预处理
3. 数据落地



1. Join Any Resource
2. Business As SQL



1. 消费计算结果
2. 执行外部操作(发短信、邮件、支付等)

目录

01 背景介绍

02 架构演进

03 实践与探索

04 总结与未来展望

ZA Insure 基于 CDH 的数仓架构

- **架构复杂**

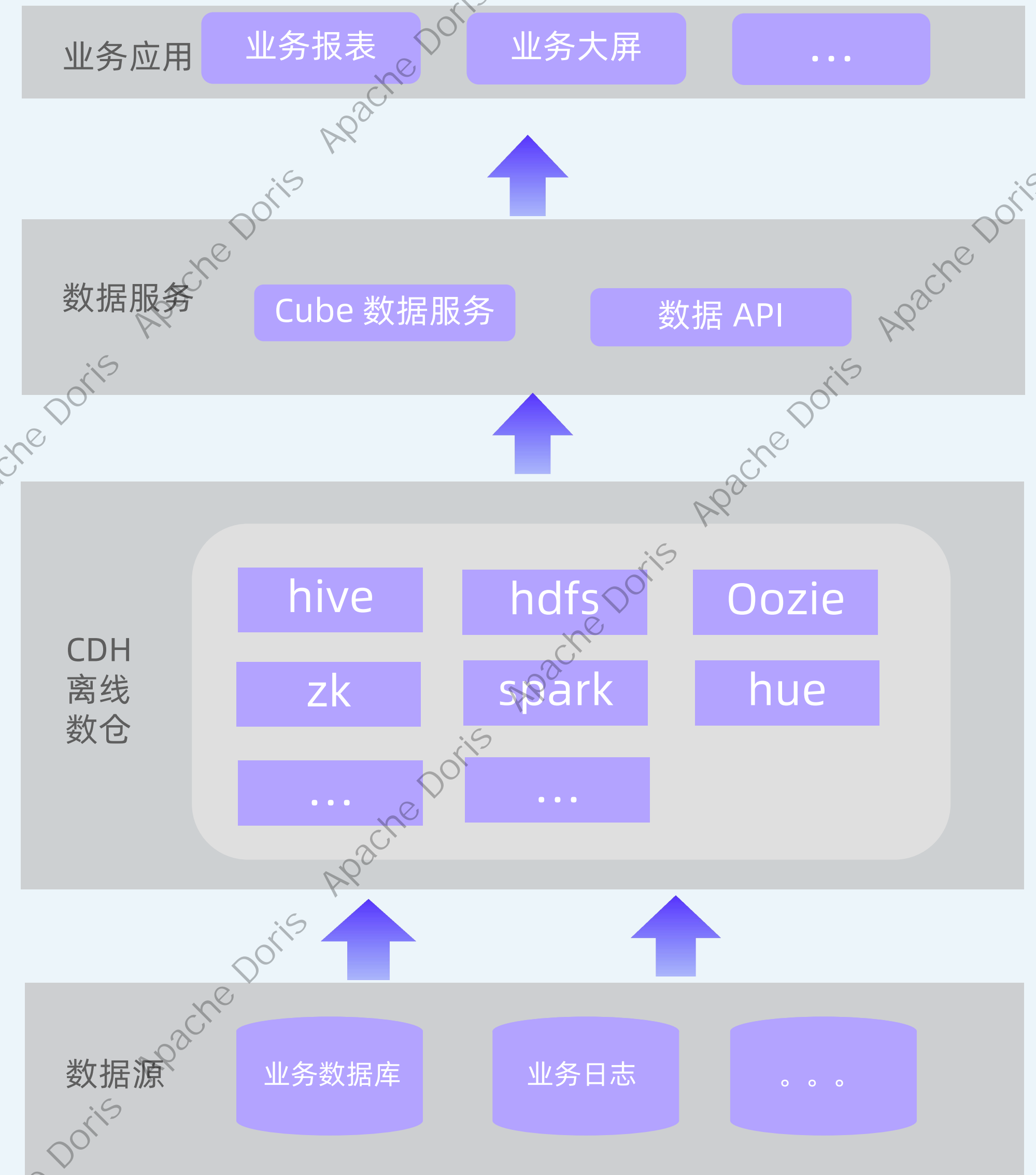
- 系统组件复杂，维护困难

- **资源频繁报警**

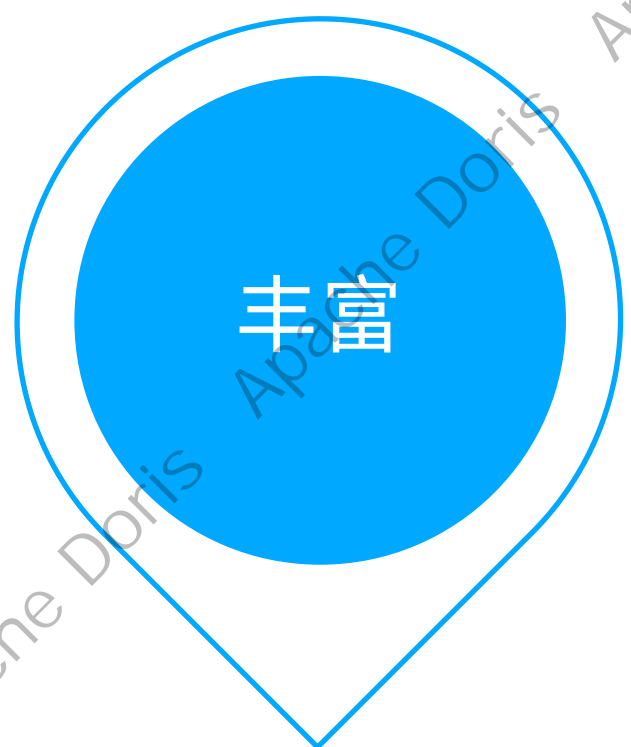
- 月底报表集中生成时资源紧张，需要人工干预

- **场景单一，效率低**

- 只能提供离线数据场景，查询效率和数据处理能力有限

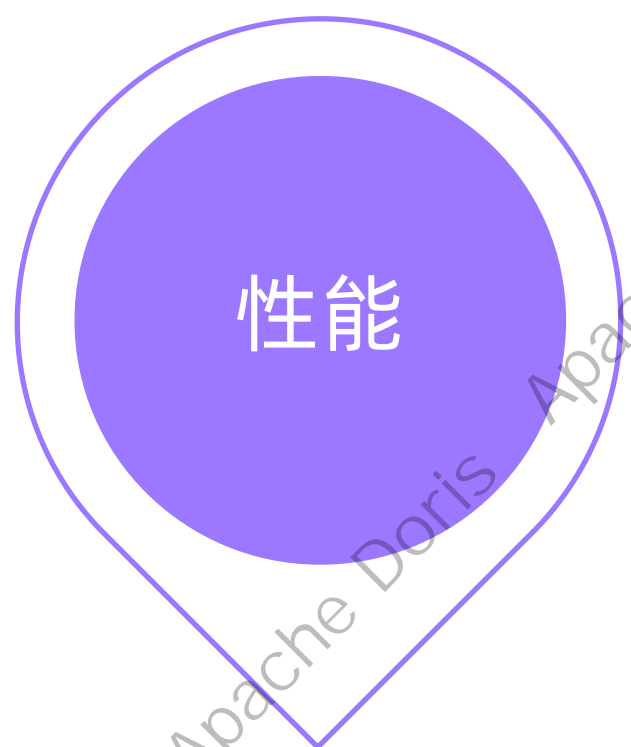


数仓架构改造的目标



功能丰富

- 覆盖目前的业务场景
- 提供实时数据处理功能



性能提升

- 相同资源的情况下，查询性能和数据处理性能高
- 降低存储成本



系统稳定

- 系统稳定，完善的监控

技术选型因素

迁移成本低

兼容 Hive SQL, 兼容 JAVA UDF
等

功能完善

丰富的数据导入导出方式

架构简洁

组件少, 运维简单, 并且监控配套完善

性能优越

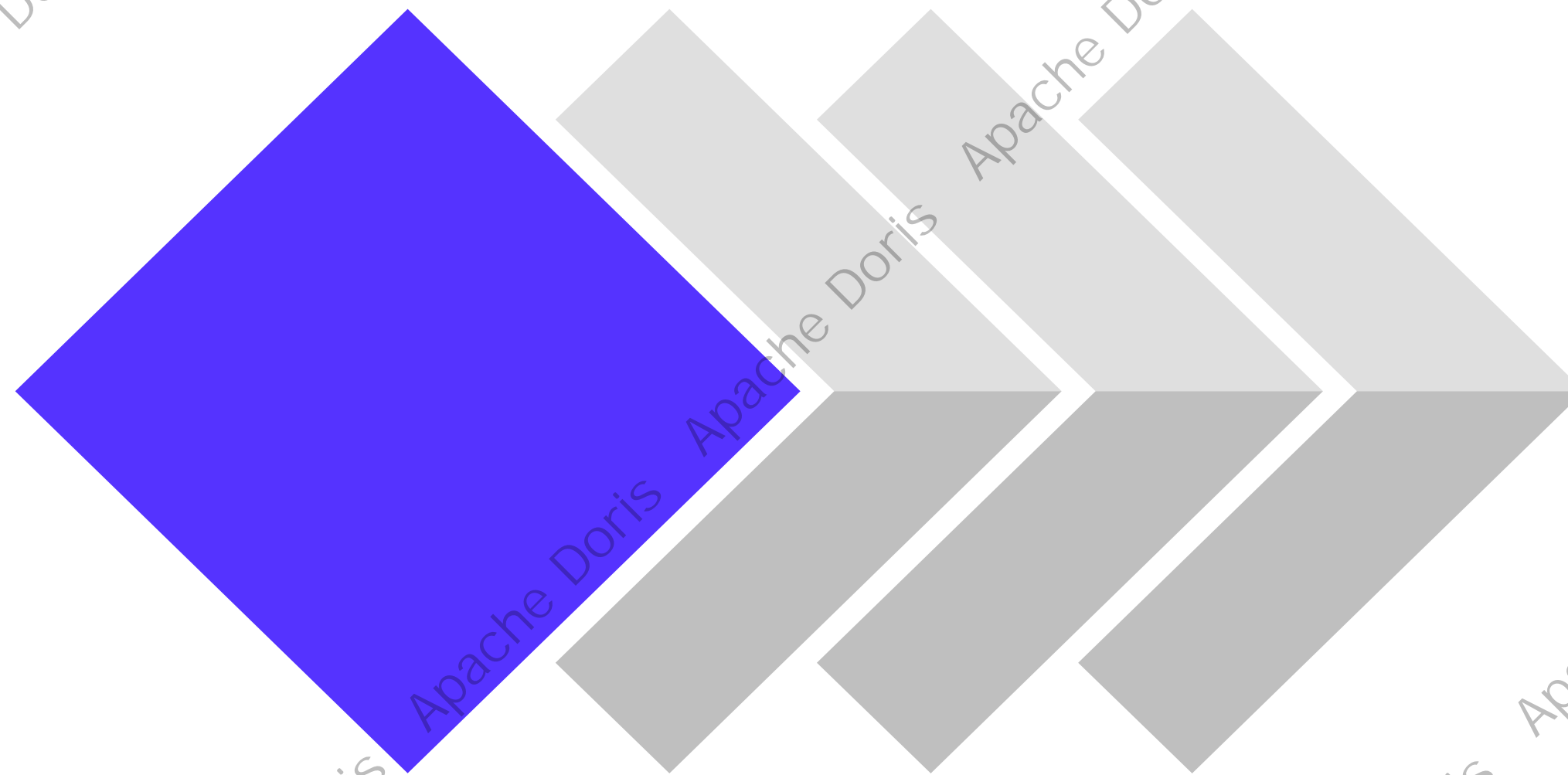
点查, Join 查询速度快

学习成本低

官方文档详细, 支持 SQL,
有助于团队共建

社区活跃

业界使用广泛, 能够有一定的技术支持



基于 Doris 的新数据仓库

• 简化架构

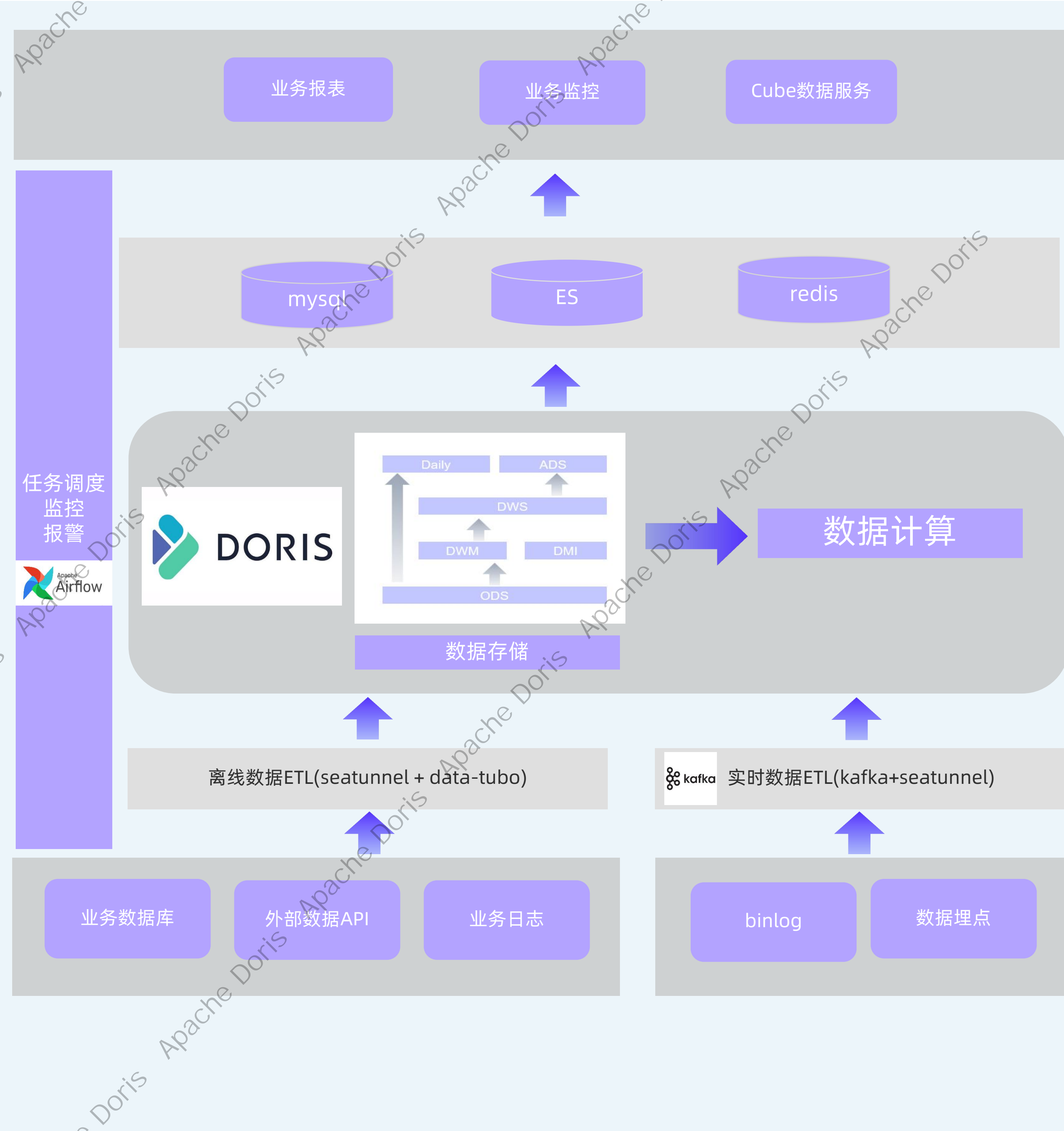
- 架构简洁清晰，能够独立维护组件不需要考虑版本兼容问题。数据存储和计算组件统一，并且便于横行扩展，数据压缩比例高等。

• 覆盖离线数仓功能

- 支持原有的离线数仓功能，又能支持实时数据业务。

• 提升查询效率

- 查询效率和数据处理能力明显提升



目录

01 背景介绍

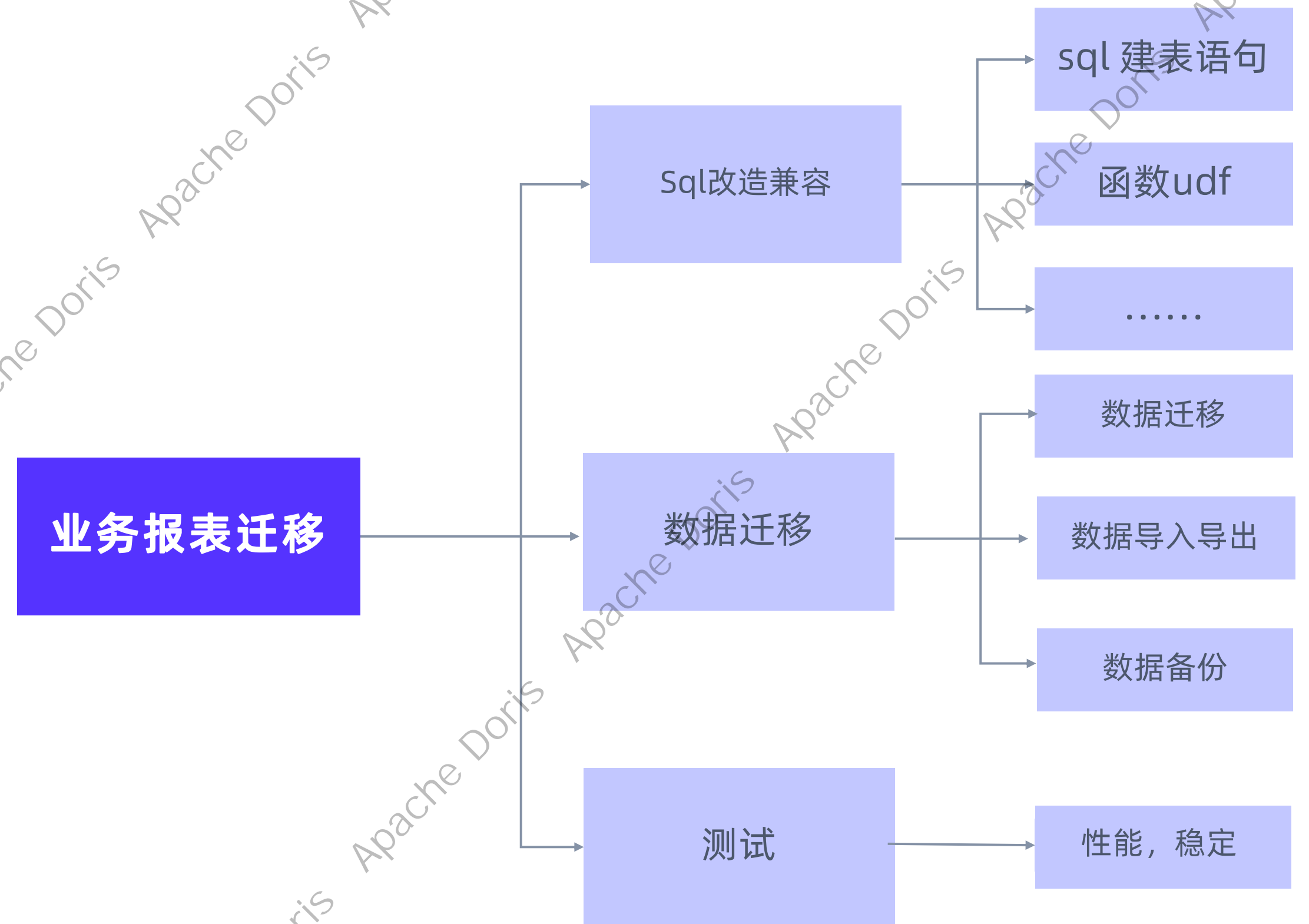
02 架构演进

03 实践与探索

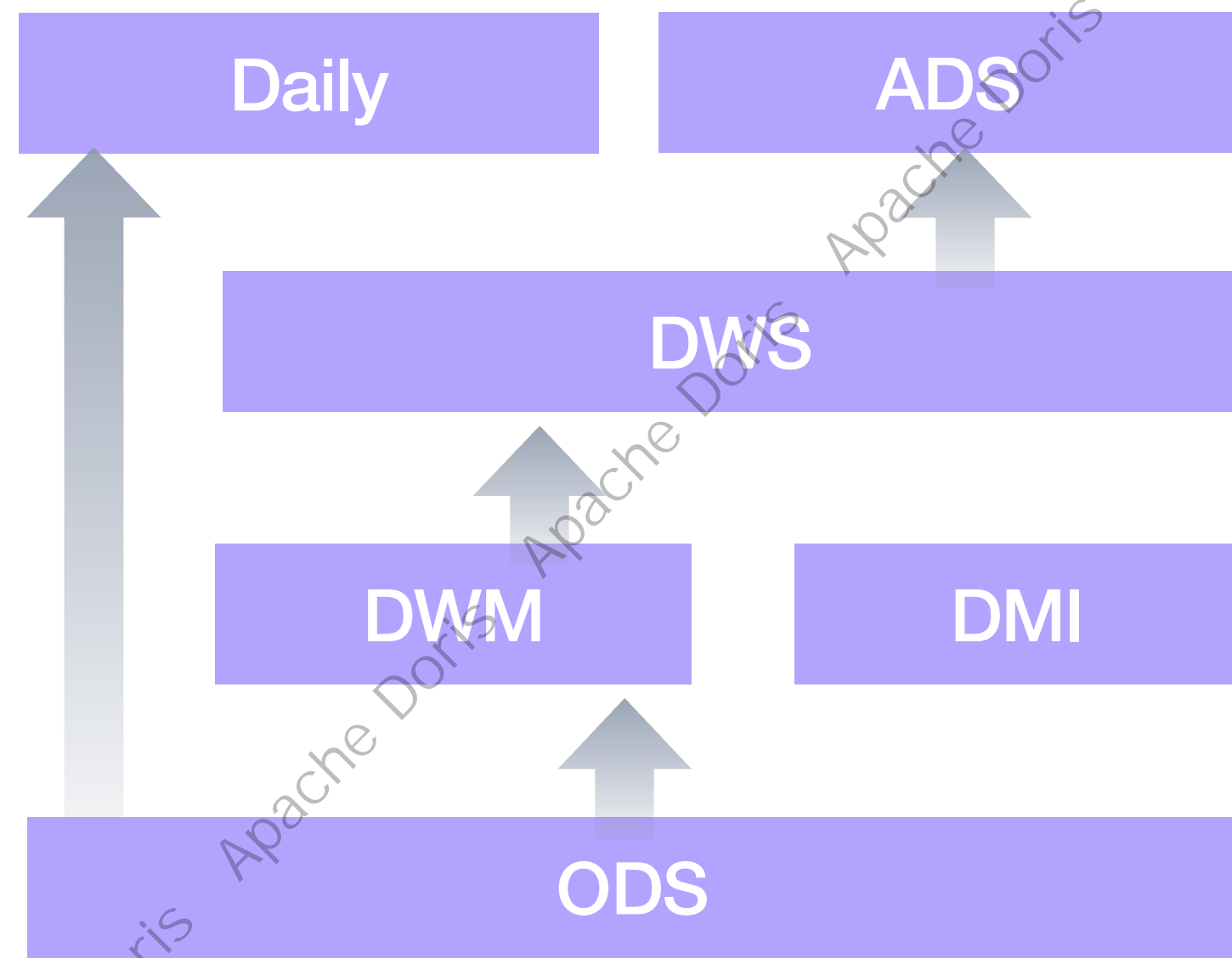
04 总结与未来展望

3.1 业务报表迁移

CDH 离线数仓主要功能是提供业务报表，
Doris 数仓体系完全可以覆盖相关功能：
在 Doris 数仓体系下替换原先**数据存储和数据聚合功能**



3.1.1 业务报表迁移-数据分层



数仓分层

相比较与传统的数仓分层 基于寿险自身的业务，单独创建 Daily 数据快照层

Daily层 每天对重点业务数据快照进行备份，备份的数据存储在Doris 中。

因此在数据存储以及数据同步方面遇到了一些问题。

3.1.2 数据分区问题

分区的问题：不合理设置了分区分桶策略

手动分区 初始化会一次性创建多个分区，手动分桶数值不合理会导致数据分布分散。

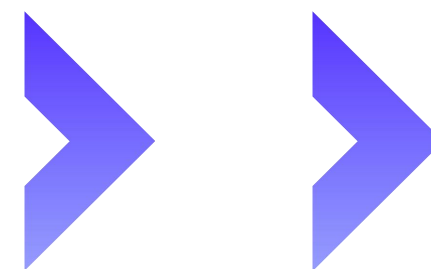
tablet 数量过多，增加FE 管理元数据的压力，FE 频繁full GC 从而影响到其他task 执行。

优化方案：

手动分区---> 动态分区 ---> 分区创建生命周期交由系统规则管理，并且不需要一次性创建

手动分桶---> 自动分桶 ---> 分桶数量由Doris 推算，保证 单个tablet 1-10G, 防止过度创建

```
CREATE TABLE test_daily.customer_address (
  `id` bigint COMMENT "",
  `customer_id` bigint COMMENT '客户id',
  `building` varchar(765) COMMENT '建筑及楼层',
  `street` varchar(765) COMMENT '街道',
  `district` int COMMENT '区',
  `address` varchar(1536) COMMENT '地址',
  `zip_code` varchar(24) COMMENT '邮编',
  `is_deleted` char(4) COMMENT '是否删除',
  `pt` date null
)
PARTITION BY RANGE(`pt`)
(
  FROM ("2000-01-01") TO ("2099-01-01")
  INTERVAL 1 YEAR
) --- 手动分区
DISTRIBUTED BY HASH(id)
BUCKETS 10
----固定分桶
```



```
CREATE TABLE test_daily.customer_address (
  `id` bigint COMMENT "",
  `customer_id` bigint COMMENT '客户id',
  `building` varchar(765) COMMENT '建筑及楼层',
  `street` varchar(765) COMMENT '街道',
  `district` int COMMENT '区',
  `address` varchar(1536) COMMENT '地址',
  `zip_code` varchar(24) COMMENT '邮编',
  `is_deleted` char(4) COMMENT '是否删除',
  `pt` date null
)
PARTITION BY RANGE (pt)()
DISTRIBUTED BY HASH(id)
BUCKETS AUTO
---自动分桶
PROPERTIES
(
  "dynamic_partition.enable" = "true",
  "dynamic_partition.time_unit" = "YEAR",
  "dynamic_partition.end" = "1",
  "dynamic_partition.prefix" = "p",
  "dynamic_partition.start_day_of_month" = "1"
); ---动态分区
```

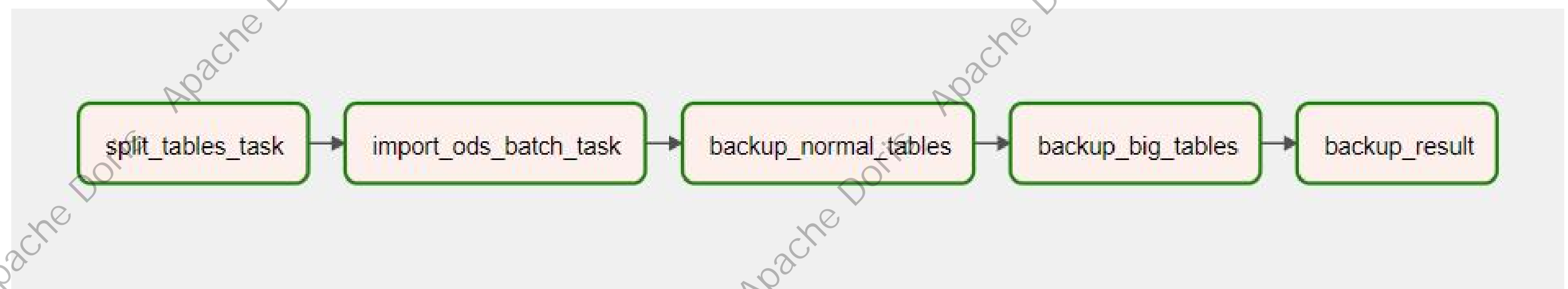
3.1.3 Daily 快照数据备份问题

Daily 数据快照遇到的问题：

数据量大的表同步会占用大量资源，导致系统资源占用，以及释放不及时，其他任务执行失败。

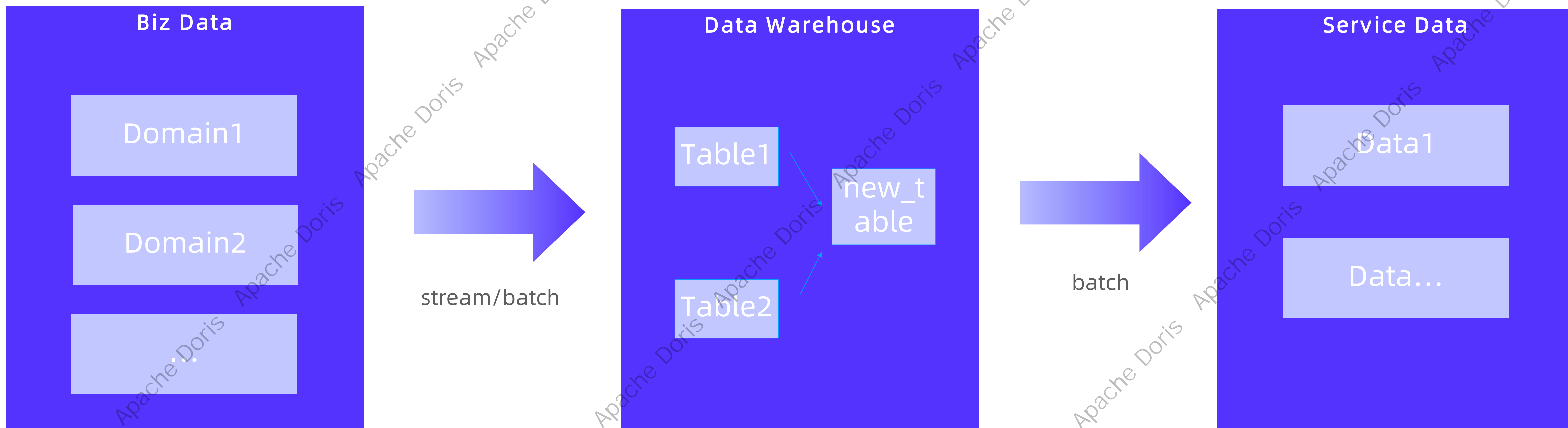
- 改造前
原先需要 ODS 备份的表 全量备份
- 改造后
全量
区分 大数据量表 和小数据量表
对于 小数据表 Doris 内部备份
对于 大数据表 全量抽取备份

场景	方式	说明
ODS->Daily	Insert into select * from ods	小于10w行的表走数仓内部层级备份
Mysql->Daily	Stream_load API	大于10w的表走stream load api性能更好



数据快照 workflow

3.2 数据服务业务流程



原始业务数据

在数仓处理数据

新业务数据

3.2.1 业务重构--数据异构的应用开发

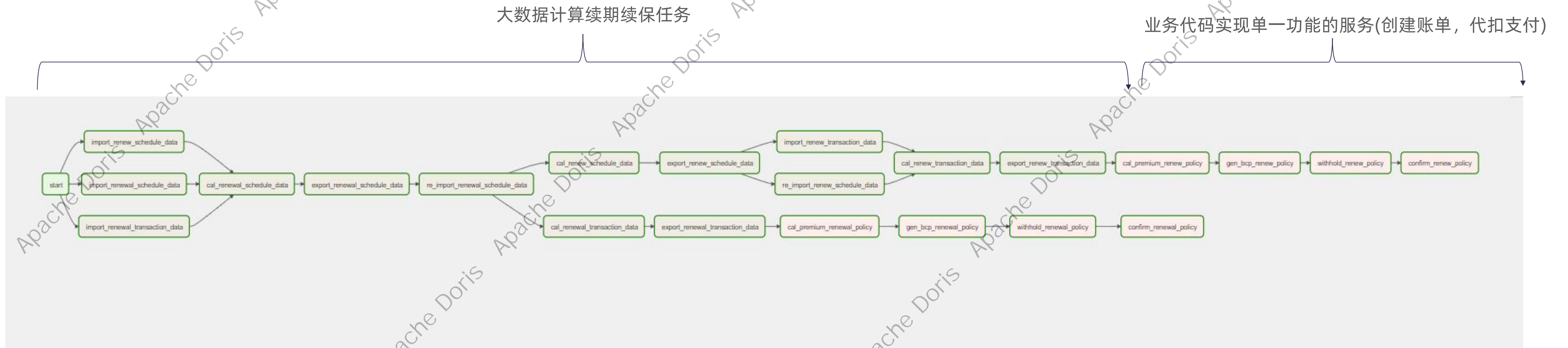
- 批处理任务优化

将批处理任务交由大数据处理，减少代码依赖

- 续期续保业务改造

通过数据异构方式，在数仓中完成数据计算，简化业务代码

- 改造效果



3.2.2 数据同步方案



DataTube

1. 通过CDC机制基于Doris stream load实现实时数据同步
2. 统一外部数据多样性，把各种外部数据导入Doris数仓
3. 寿险项目只做实时数据流同步

1. 自研数据同步工具，基于Mysql Stream和Doris的Stream load实现高性能数据导入，相比其他ETL工具性能提高80%，资源占用降低60%
2. 支持多种数据源，兼容SeaTunnel数据同步配置
3. 寿险项目主要做批数据的全量/增量导入导出

3.2.3 分布式ID

方案	优点	缺点
数仓自增 ID	<ul style="list-style-type: none">数仓原生支持，性能更好，不需外部依赖	<ul style="list-style-type: none">2.1 版本后支持，老版本升级后用复杂业务场景 ID 幂等不好处理
数据库自增 ID	<ul style="list-style-type: none">ID 生成后置，简化处理逻辑	<ul style="list-style-type: none">复杂业务场景 ID 幂等不好处理过分依赖导出业务表
雪花算法分布式 ID	<ul style="list-style-type: none">原始支持分布式ID 生成性能复杂场景业务处理友好	<ul style="list-style-type: none">需要额外 UDF 实现

雪花算法UDF实现

使用过程中遇到的问题：

主键重复

分析原因：

Doris的Java UDF 支持向量优化，在加载 UDF 过程中会并发加载多个实例，导致普通Java UDF 实现的雪花算法 nextId 方法锁功能失效，从而产生重复 ID 的问题。

解决方案：

修改 FE 的 java-extensions 中的 java-udf.UdfExecutor 的 evaluate 方法，在多实例 invoke 方法前判断雪花算法请求，直接调用雪花算法函数返回生成 ID，避免多实例并发问题。

```
/**
 * Evaluates the UDF with 'args' as the input to the UDF.
 */
no usages
public Object[] evaluate(int numRows, Object[] column) throws UdfRuntimeException {
    if (SnowflakeIdWorker.isIdFunction(fn.getName().getFunctionName())) {
        Object[] result = (Object[]) Array.newInstance(Long.class, numRows);
        for (int i = 0; i < numRows; ++i) {
            result[i] = SnowflakeIdWorker.getId();
        }
        return result;
    }
    try {
        Object[] result = (Object[]) Array.newInstance(method.getReturnType(), numRows);
        Object[][] inputs = (Object[][]) column;
        Object[] parameters = new Object[inputs.length];
        for (int i = 0; i < numRows; ++i) {
            for (int j = 0; j < column.length; ++j) {
                parameters[j] = inputs[j][i];
            }
            result[i] = methodAccess.invoke(udf, evaluateIndex, parameters);
        }
        return result;
    } catch (Exception e) {
        LOG.info("evaluate(int numRows, Object[] column) Exception: " + e.toString());
        throw new UdfRuntimeException("UDF failed to evaluate", e);
    }
}
```

3.3 其他应用场景

实时业务监控

- **业务指标监控大盘**
通过Doris 数仓 聚合实时指标 实现对 客群用户和异常事件的实时监控。
- **保全等业务场景流程监控**
标准化保全业务流程，配置业务报警规则，通过Doris 计算出异常数据，导出到业务中，发送业务报警，及时通知运营人员处理。

数据迁移

- **数据迁移**
通过Doris 实时数仓完成数据迁移，无需停机。
- **系统重构**
重构系统，通过Doris 优化表结构，实现数据迁移。

目录

01 背景介绍

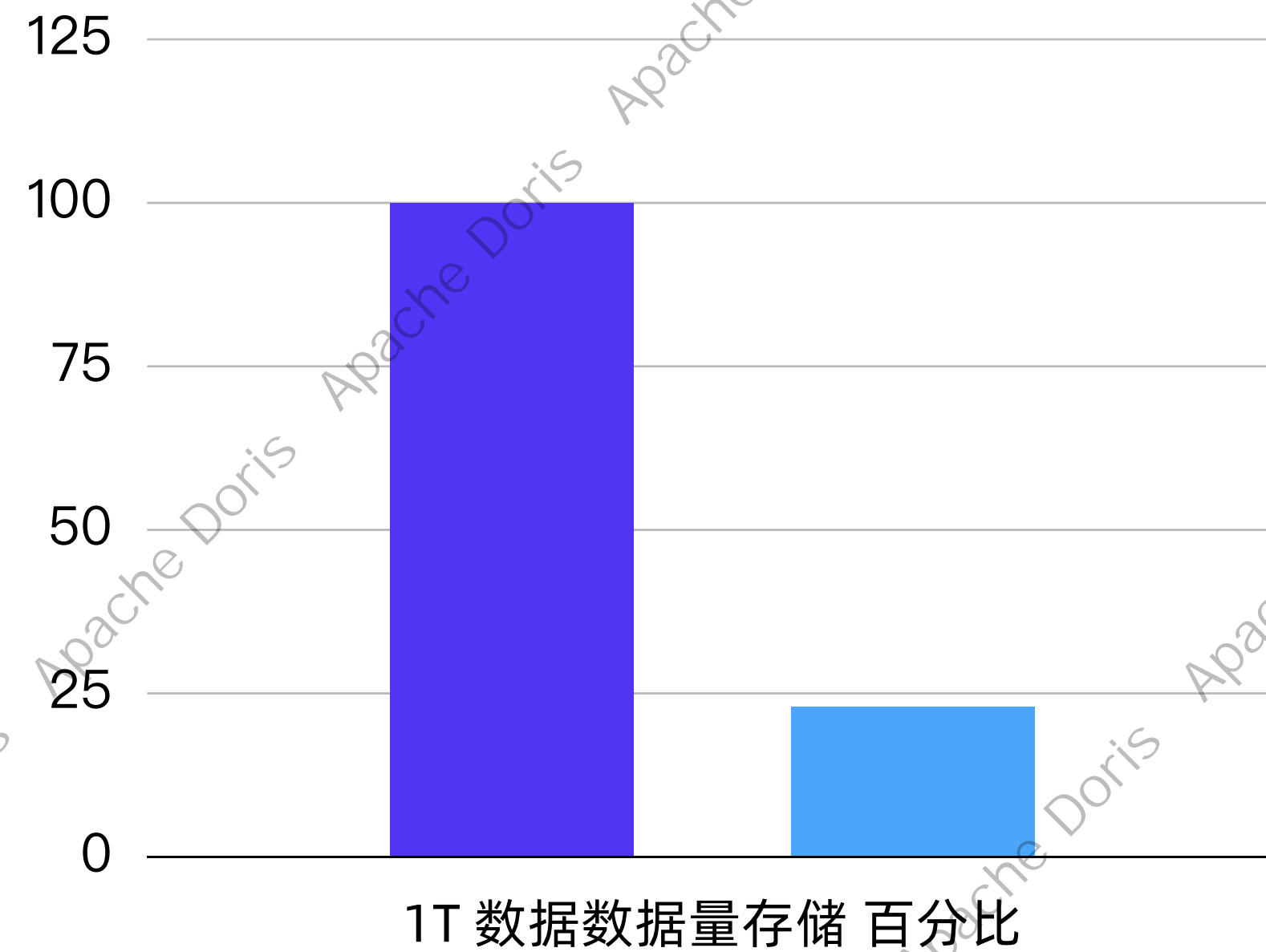
02 架构演进

03 实践与探索

04 总结与未来展望

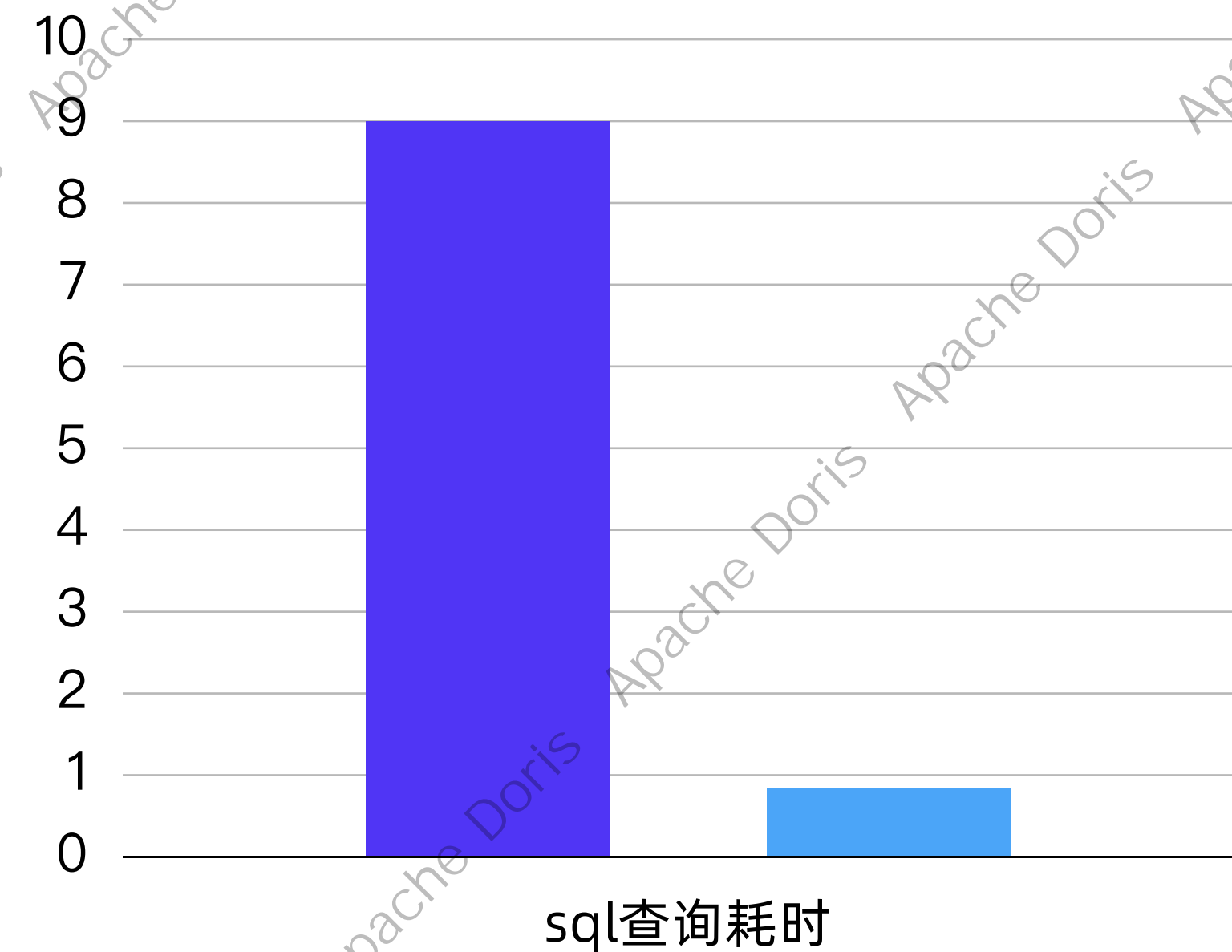
4.1 改造收益

数据存储压缩效率高
相同数据量存储占比减少 80% 以上



■ CDH数仓 ■ Doris 数仓

数据查询效率提升明显
等效环境下 SQL 执行查询平均耗时减少 90%



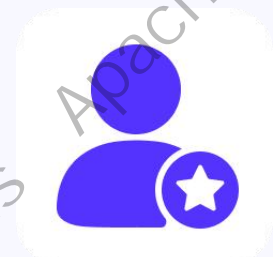
■ CDH数仓 ■ Doris 数仓

改造收益



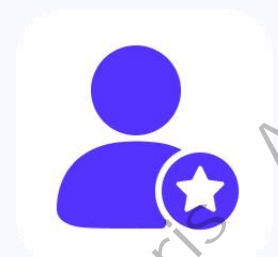
业务报表迁移完成

- 上百张业务报表迁移至新实时数仓平台，稳定运营半年



业务新解决方案

- 将续期续保等业务计算迁移至数仓平台进行业务重构



数据异构迁移功能

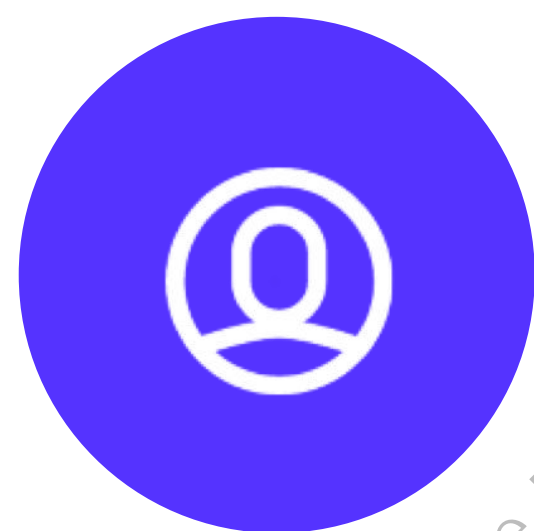
- 10+ 应用已完成不停机数据异构迁移



实时业务监控

- 提高业务运营响应速率，达成数字化运营目标。

技术展望



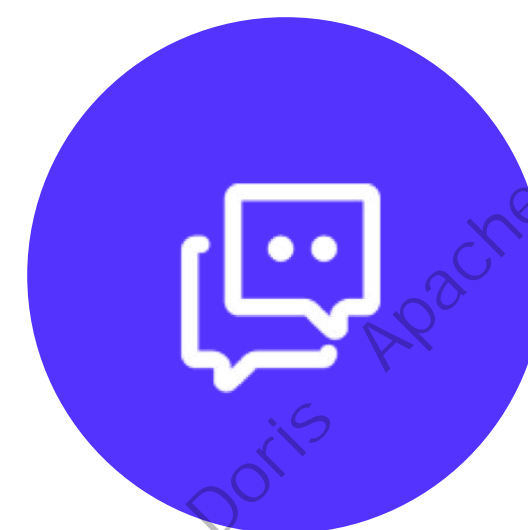
实时业务探索

探索更多结合 Doris 的 OLAP 实时业务场景



数据中台体系完善

包括数据质量和数据资产管理，构建完善的数据中台体系。
提升系统稳定性



构建湖仓一体

结合ES 集群，构建数据共享层

Thank You!

