

# Apache Doris 在地平线 实时数仓 0-1 建设应用实践

宋凯

地平线 云服务部 AI 基础平台资深研发专家

# 目录

1. 个人介绍
2. 背景介绍
3. 架构演进
4. 应用实战
5. 未来展望



## 个人介绍



宋凯 | 地平线AI基础平台研发工程师

---

- **百度** 系统部&基础架构部 负责时序存储数据库&离线计算框架等研发
- **腾讯** pcg大数据平台部，负责OLAP融合引擎优化&性能负载中心建设等
- **地平线** 云服务部-AI基础服务平台，负责实时数仓，平台数据链路治理，多模态检索等



# 背景介绍--公司&产品形态

公司简介 -- 地平线



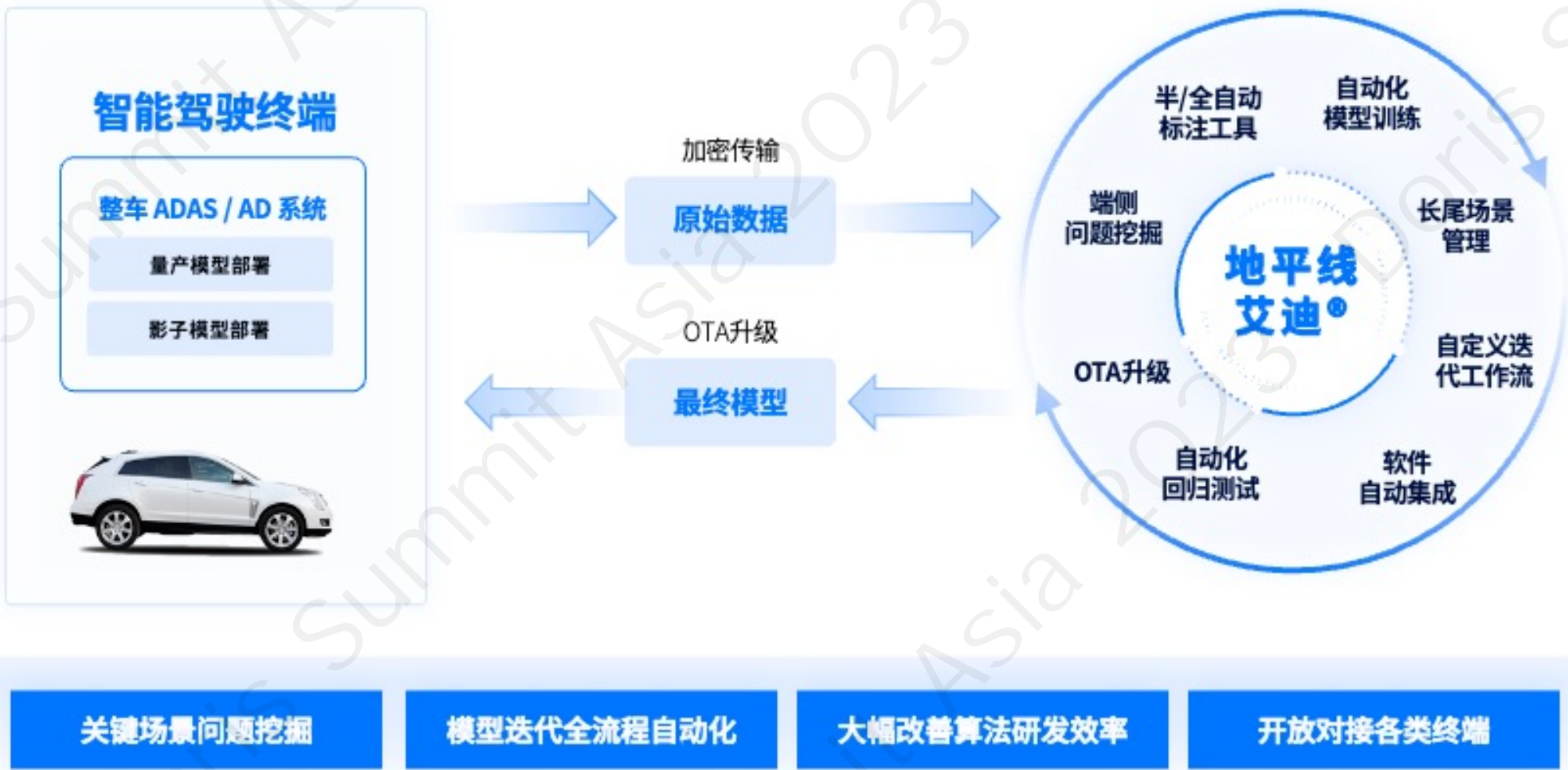
Horizon Robotics

地平线机器人是中国第一，世界第三实现车载芯片量产的深度神经网络芯片公司，**90%**的中国自主品牌整车厂是我们的客户，集成芯片，AI,自动驾驶三大热门赛道。截止到2023年4月底，芯片的累计出货量大于**300W+**

一站式数据闭环模型迭代 -- 艾迪平台

边缘端

云端



# 背景介绍--基础数据流



每年产生的元数据记录约：**100亿+**

平台处理部分

包含感知数据包

- 1 采集
- 自采
  - 影子模式

mcap  
pack

2 上传

3 加载

对象存储

数据处理

抽帧

解析

模型打标

业务系统

标注平台

回灌平台

仿真评测

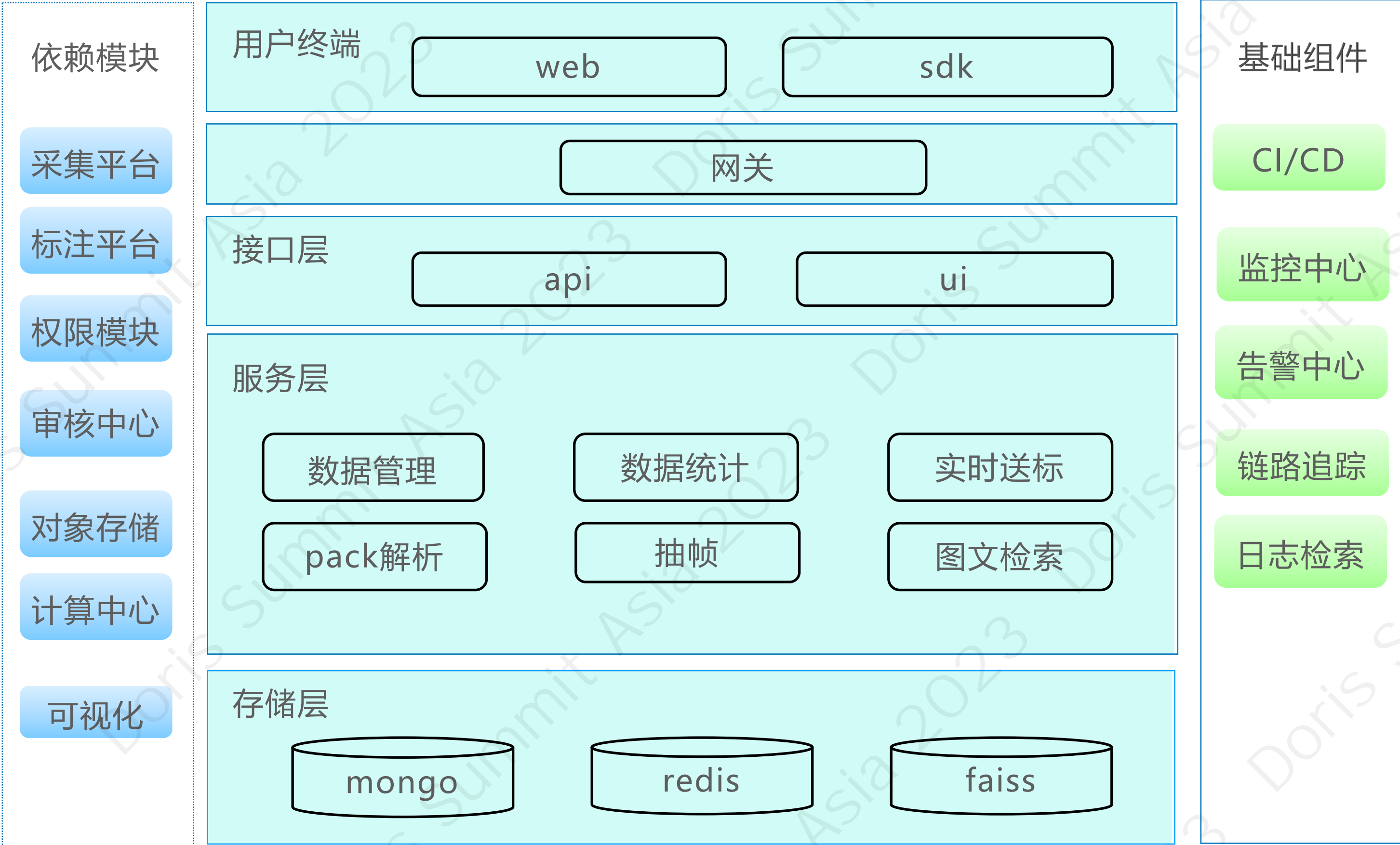
5 业务交互

数据库



# 架构演进 --之前技术架构&问题

2021年的技术架构图



该架构的瓶颈问题

1 随机字段检索出现超时

2 数据统计不灵活且部分大表无法统计

3 超大行记录表检索较慢

数据不断累积  
(单表1亿)

存在的问题

架构演进 -- 选型过程

需求核心点

- 1 点查性能优越，mongo上的部分点查能切换到存储引擎
- 2 支持复杂查询，例如多表关联join
- 3 海量数据场景下随机多维查询能力优秀
- 4 实时性能优越，方便对接常用的BI组件
- 5 方便修正少量数据
- 6 集群运维压力小，不要投太多人力维护

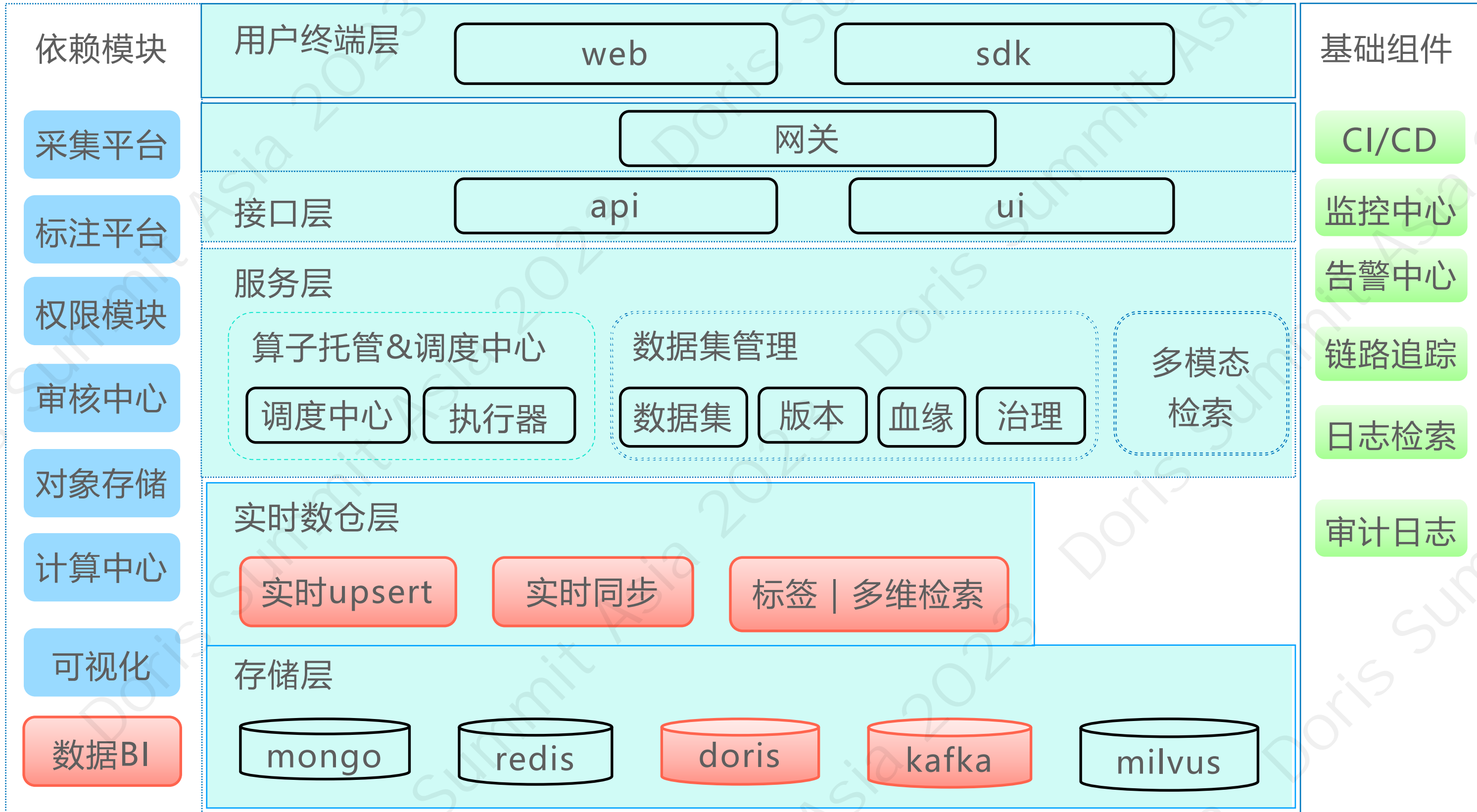
选型对比项

对比项	clickhouse	doris	impala	trino(prest)	spark-sql	hive	kylin	druid
压秒级响应	Y(单表)	Y	Y	N	N	N	Y	Y
高并发	N	Y	Y(结合kudu)	N	N	N	Y	Y
SQL友好度	Y(不支持开窗函数,其他都支持)	Y	Y	Y	Y	Y	Y	N
离线查询	Y	Y	Y	Y	Y	Y	Y	Y
实时查询 (数据实时性)	Y(最好)	Y	N(如果需要实时可以结合kudu)	N	N	N	N	Y
去重能力	Y (session级别)	Y (千万级别支持)	Y	Y ⊖	Y	Y	Y	N
明细查询	Y	Y	Y	Y	Y	Y	N	N
多表JOIN	Y (大表join性能差)	Y	Y	Y	Y	Y	Y	N
模型修改 (动态修改表字段)	Y	Y	Y	Y	Y	Y	N	N
ODBC/JDBC	Y	Y(mysql协议)	Y	Y	Y	Y	Y	Y
数据更新	Y (合并)	Y (合并)	N	N	N	N	N	N
数据湖	N(支持中)	Y	Y	Y	Y	Y	N	N
集群运维友好性	N	Y	Y	Y	N	Y	Y	N

# 架构演进 --当前技术架构&效果

当前架构

达到的效果



集群规模 3FE + 10BE

写入

- 入库延迟: 5s
- 存储空间: 原来1/3
- 单表吞吐: 3.2w/s

检索-随机多维

- 数据量: 20亿行, 100列+
- 检索时效: 约3s

检索--点查 (Rollup)

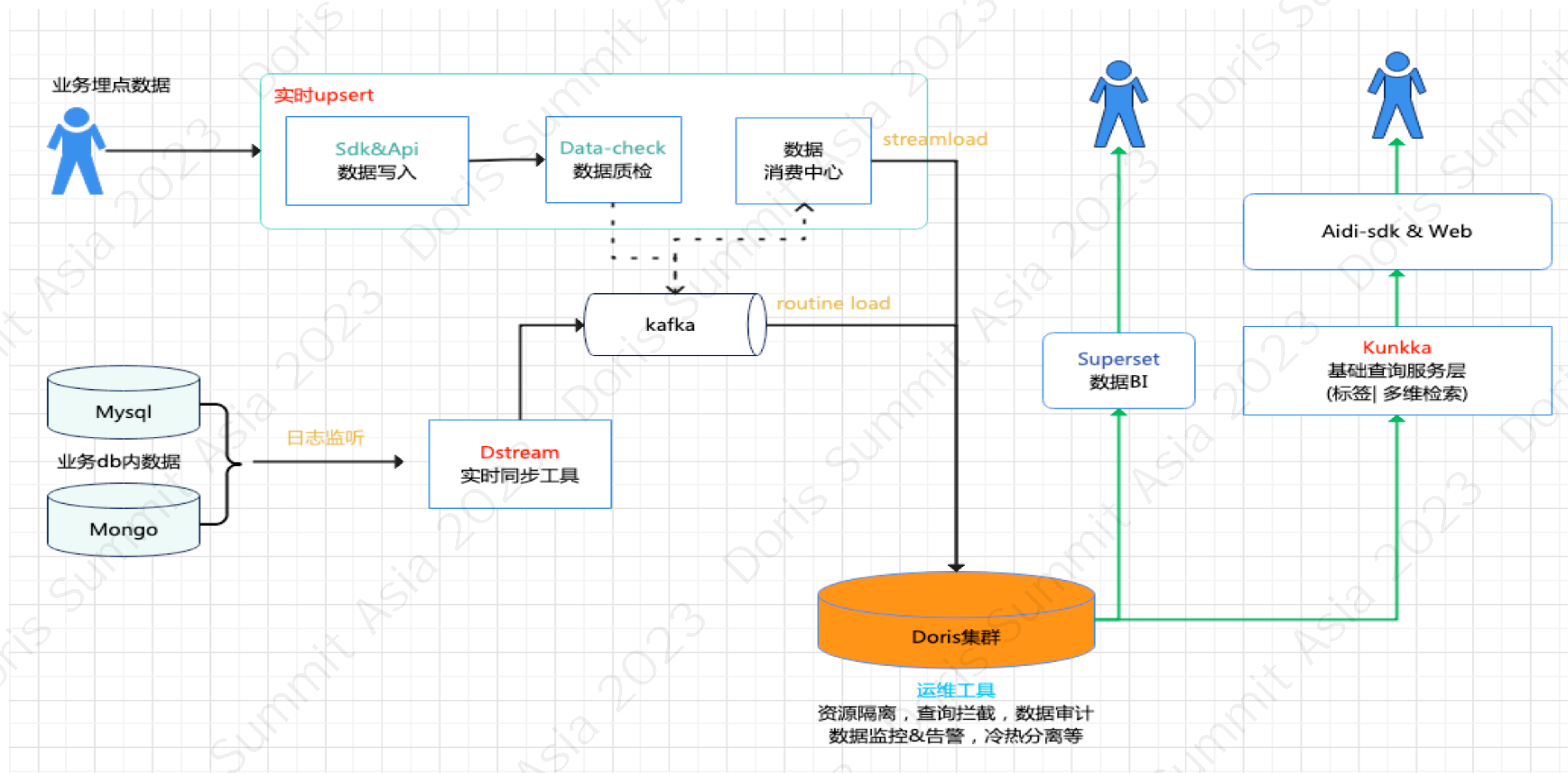
- 数据量: 60亿+
- 并发: 300
- 时效: 80 ~ 200ms

检索--标签

- 数据量: 100亿
- 标签个数: 5000
- 时效: 300ms [同时检索 < 10 标签]

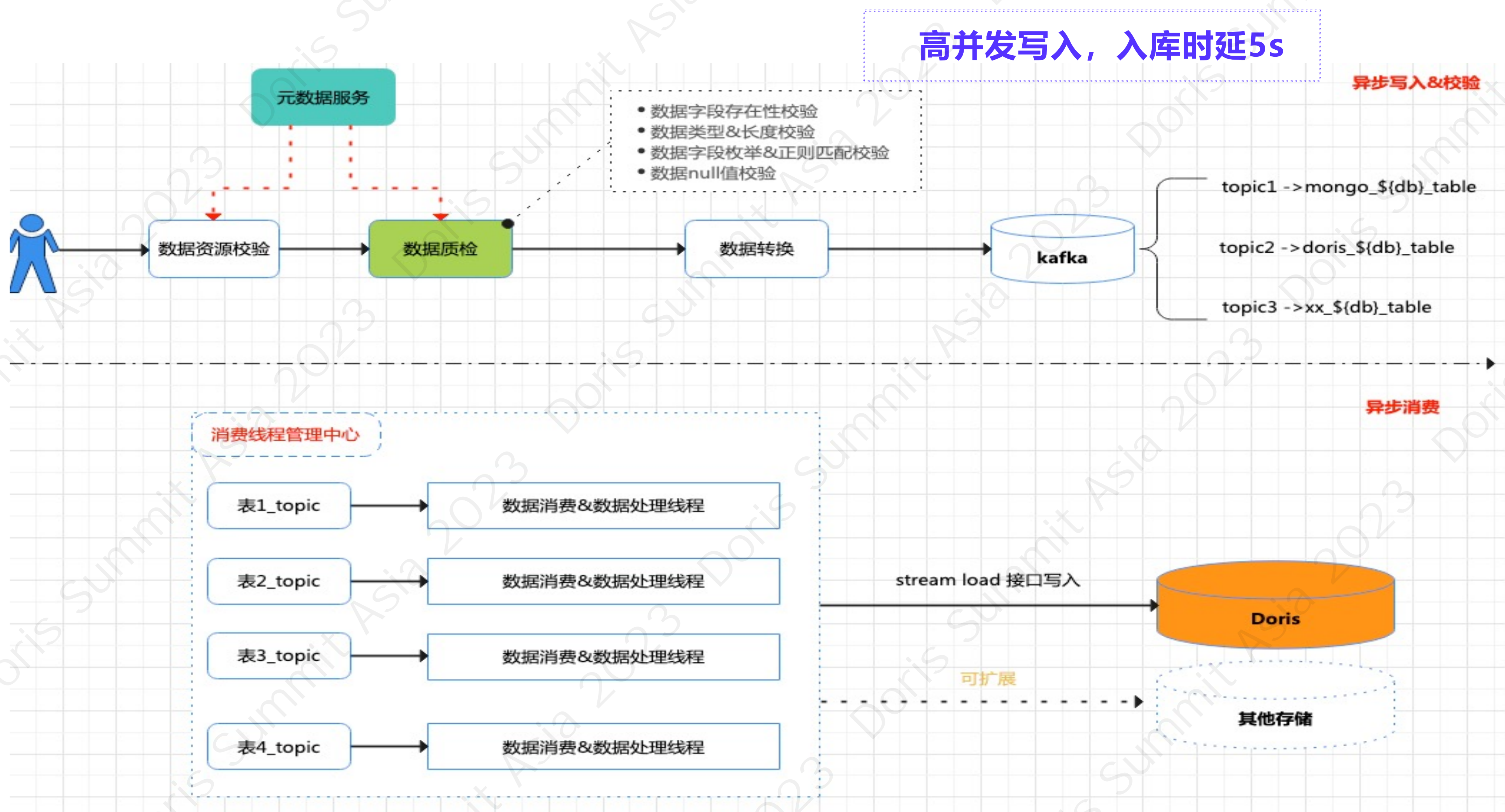


## 应用实战 -- 实时数仓数据流架构



# 应用实战 --实时Upsert

高并发写入，入库时延5s



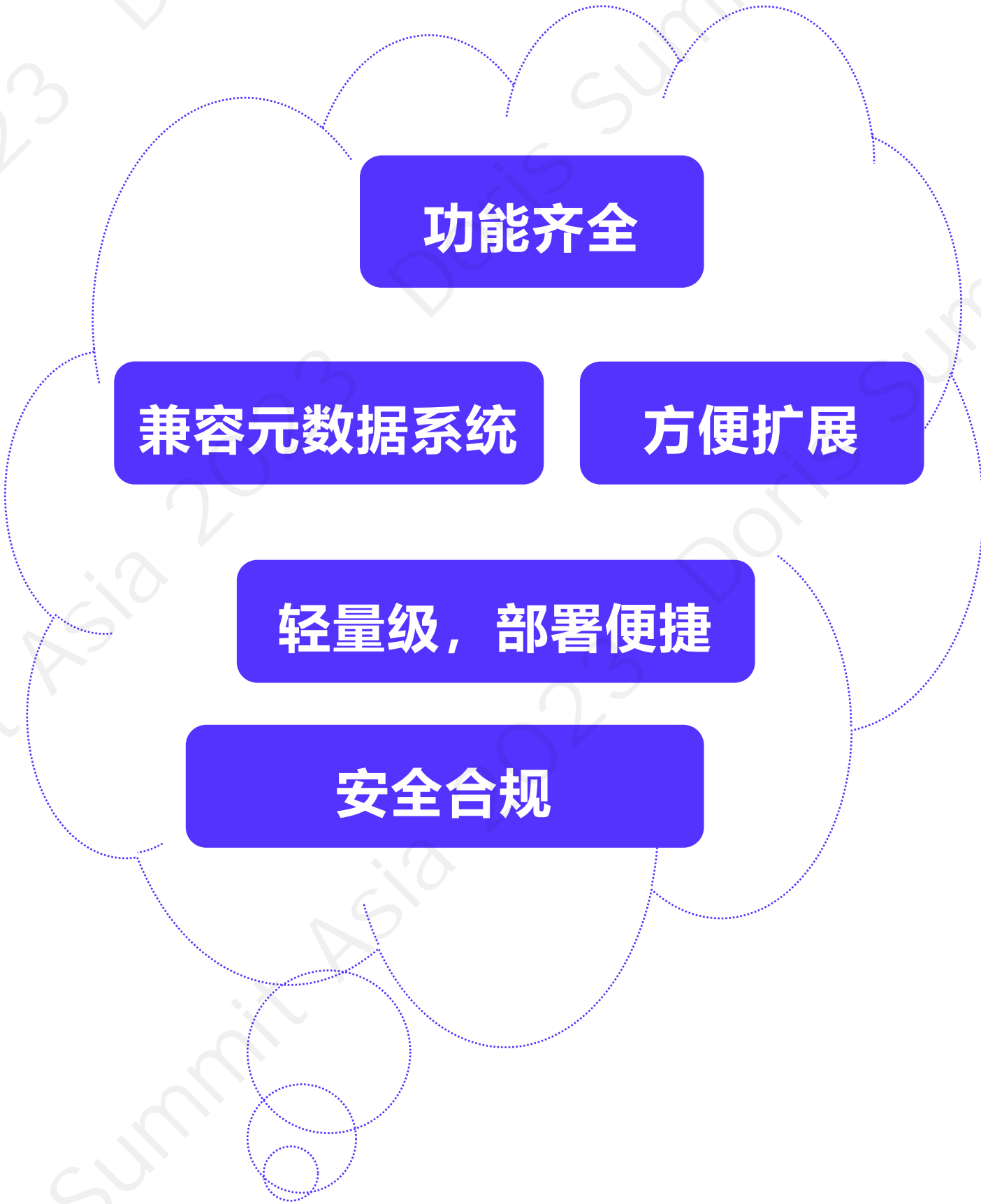


# 应用实战 --数据同步Dstream --研发背景

## 业界的方案

	Flink CDC	Debezium	DataX	Canal	Sqoop	kettle	Oracle Goldengate
CDC 机制	日志	日志	查询	日志	查询	查询	日志
增量同步	✓	✓	✗	✓	✓	✗	✓
断点续传	✓	✓	✗	✓	✗	✗	✓
全量同步	✓	✓	✓	✗	✓	✓	✓
全量+增量	✓	✓	✗	✗	✓	✗	✓
架构	分布式	单机	单机	单机	分布式	分布式	分布式
Transformation	★★★★★	★★	★★	★★	★★	★	★
生态	★★★★★	★★★	★★★★	★★★★	★★	★★	★★★★

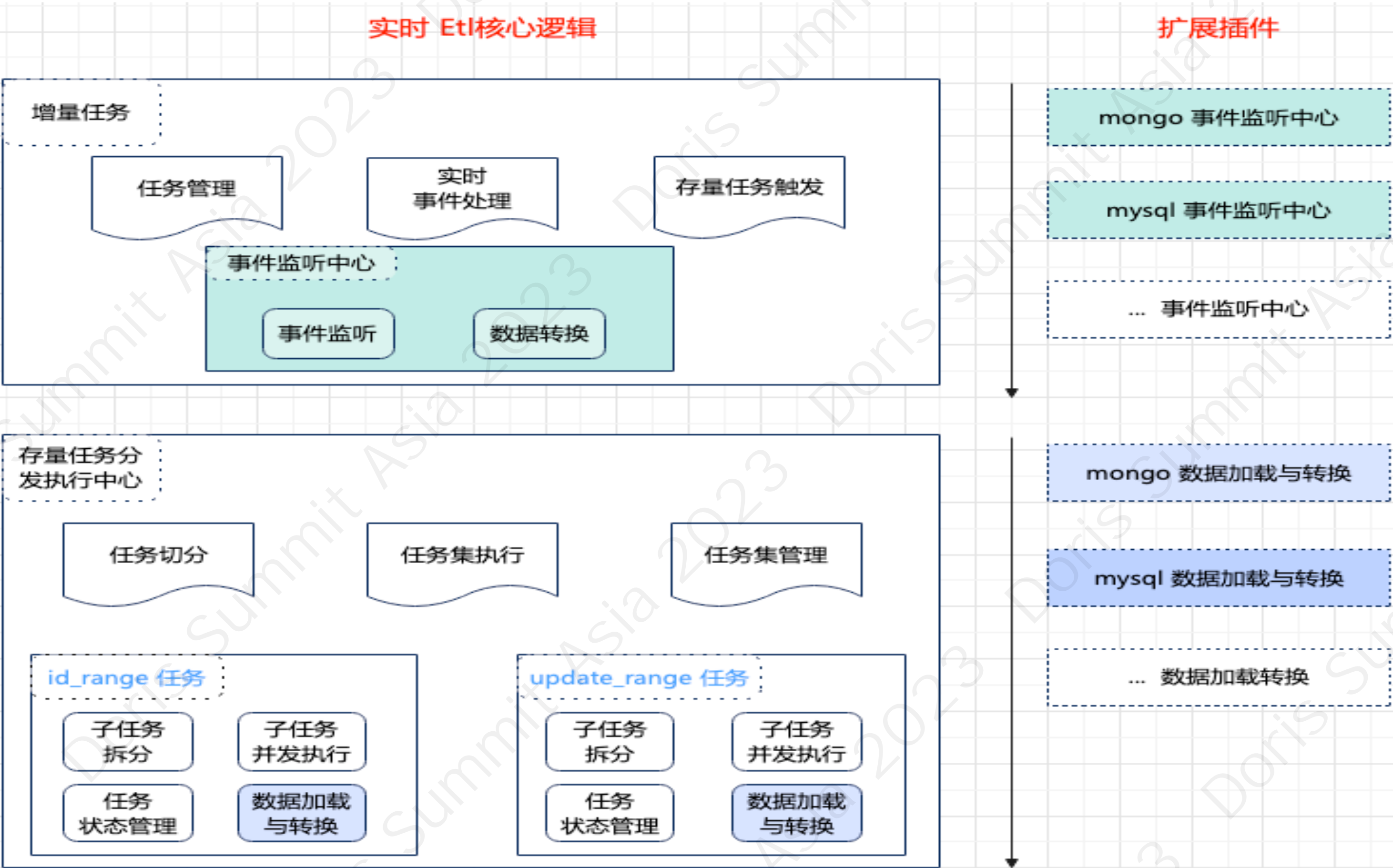
## 我们的需求



# 应用实战 --数据同步Dstream

## 组件架构

## 特点



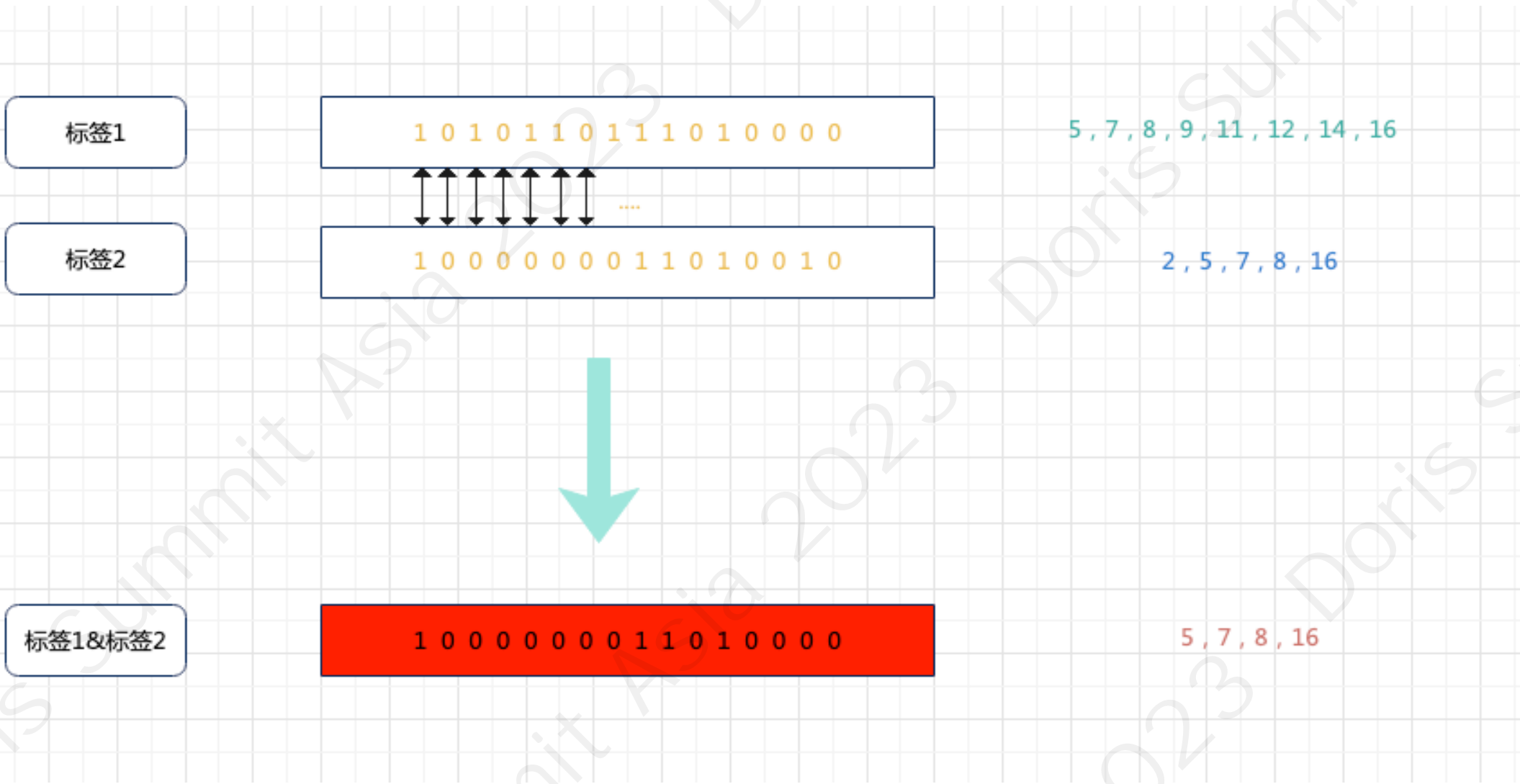
- 1 轻巧灵活，无需基于大数据计算引擎
- 2 高稳定和可用性
- 3 支持分布式部署
- 4 特殊功能支持【json字段指定层级拆解】
- 5 高性能

核心指标	性能
写入吞吐量	单表10w/s
同步时延	> 5s
支持表数量	无限制
业务库支持	基于行存数据库



# 应用实战 -- 海量标签检索背景

## 基本原理



### bitmap计算的特点

- 1 快速的运算
- 2 低存储开销
- 3 高并发性能
- 4 丰富检索支持-AND, OR, NOT

## Doris 原生支持

- SQL 函数
    - 数组函数
    - 日期函数
    - 地理位置函数
    - 字符串函数
    - Struct Functions
    - Combinators
    - 聚合函数
    - Bitmap 函数
      - TO\_BITMAP
      - BITMAP\_HASH
      - BITMAP\_FROM\_STRING
      - BITMAP\_TO\_STRING
      - BITMAP\_TO\_ARRAY
      - BITMAP\_FROM\_ARRAY
      - BITMAP\_EMPTY
      - BITMAP\_OR
      - BITMAP\_AND
      - BITMAP\_UNION
- 函数支持相当丰富

## 数据场景

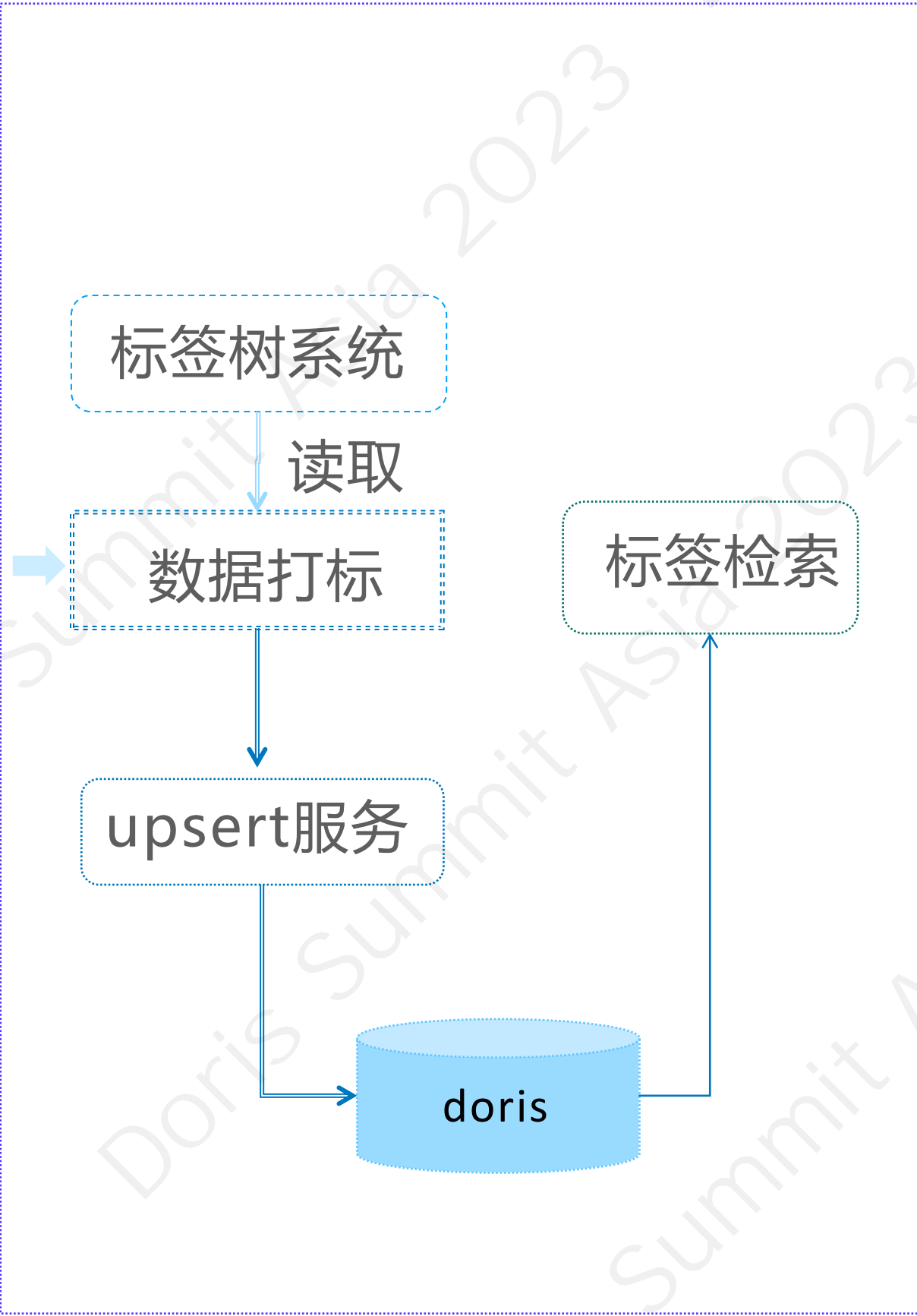
### 规模

实体frame和虚拟切片基础数据在100亿级  
标签个数约5000，会随着业务扩展



# 应用实战 -- 海量标签检索设计

## 数据流架构



## 表设计 & 检索

```
CREATE TABLE `tag_index` (  
  `tag` varchar(255) NOT NULL COMMENT '标签值',  
  `resource` varchar(32) NOT NULL COMMENT '标签关联的资源',  
  `dt` date NOT NULL COMMENT '打标日期, 例如2022-04-30',  
  `seeds` int(11) NOT NULL COMMENT '打散因子, 按需调整',  
  `vid` bitmap BITMAP_UNION NULL COMMENT 'uid聚合bitmap值',  
  INDEX resource_b (`resource`) USING BITMAP COMMENT 'resource的bitmap索引',  
  INDEX dt_b (`dt`) USING BITMAP COMMENT 'dt的bitmap索引'  
) ENGINE=OLAP  
AGGREGATE KEY(`tag`, `resource`, `dt`, `seeds`)  
COMMENT 'OLAP'  
DISTRIBUTED BY HASH(`dt`) BUCKETS 8  
PROPERTIES (  
  "replication_allocation" = "tag.location.default: 3",  
  "in_memory" = "false",  
  "storage_format" = "V2",  
  "disable_auto_compaction" = "false"  
);  
  
SELECT  
  bitmap_count (  
    bitmap_intersect ( vid ) )  
FROM  
  (  
    SELECT  
      tag,  
      bitmap_union ( vid ) vid  
    FROM  
      tag_index  
    WHERE  
      tag IN (  
        '64f9a63405d06b57082f2123',  
        '64f07b6515704979848a9b12',  
        '64fcdf42b44b3a69e0db9562',  
        '64fcb17604846e86e5f3a121',  
        '64fcd069692f0e79fce607dd',  
        '64fd5c61692f0e79fce8d3e2',  
        '64fd5c61692f0e79fce8d414',  
        '64fd5c61692f0e79fce8d43e'  
      )  
      AND resource = "image"  
      AND dt > 20221201  
    GROUP BY  
      tag  
  ) a;
```

## 效果

存储空间占用: 约130GB

性能:

检索标签数	检索耗时(近似-统计值)
1	90ms
5	280ms
8	380ms
10	670ms
15	900ms
20	1.2s



# 应用实战 -- 总体受益

线上集群规模 3FE + 10BE

## 实时Upsert

- 入库延迟: 5s
- 单表吞吐: 3.2w(记录)/s
- 单表写入并发度: 1w+ [受限服务端连接数]

## 实时同步

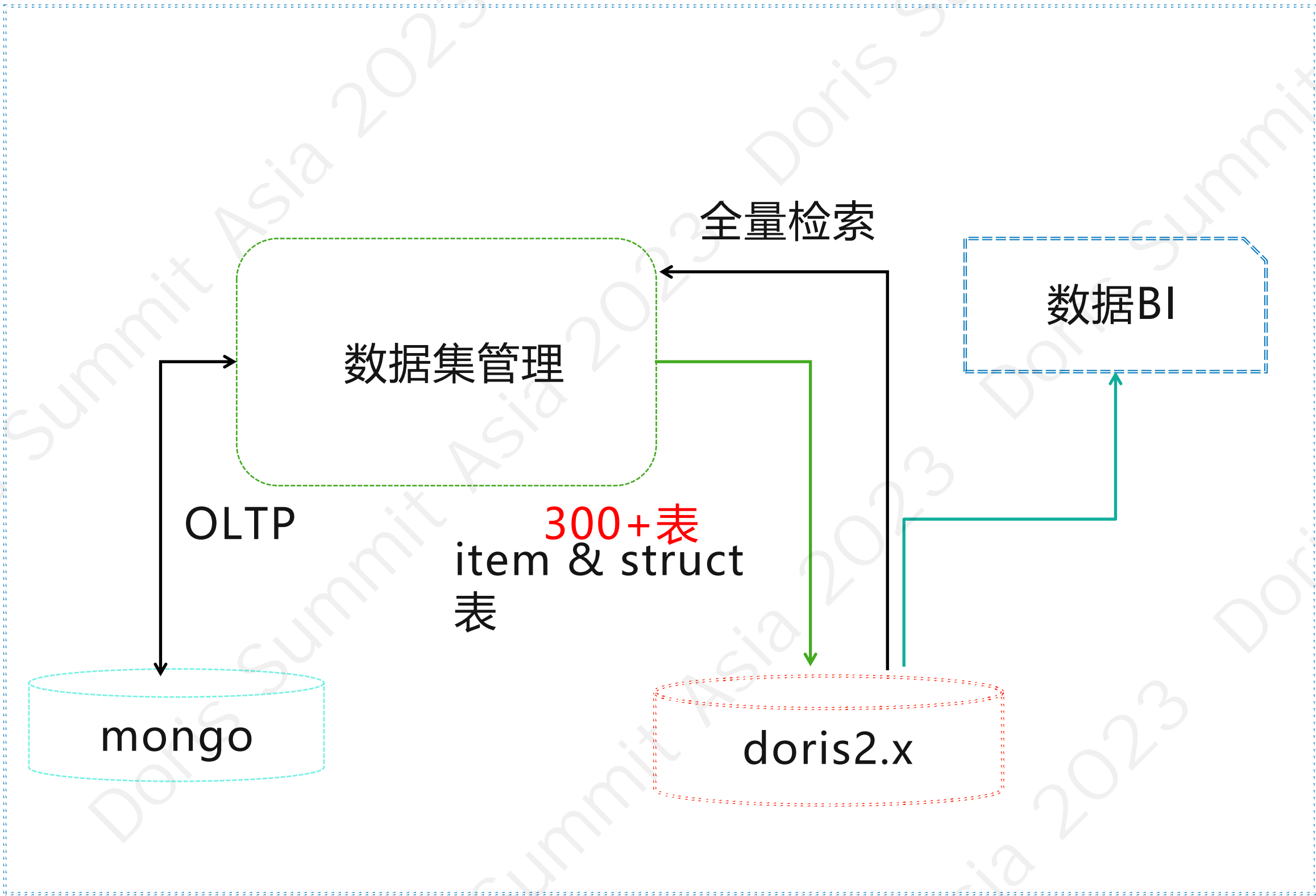
- 增量入库延迟: 5s+
- 存量单表同步速度: 10w/s+ 【受限源集群最大读取IO量】
- 同步表数量: 不限

## 标签检索

- 标签数据量: 100亿+
- 标签个数: 5000
- 检索时效: 约300ms 【同时检索标签数约8个】

# 未来展望

## 架构



## 跟进项

- 1 doris2.x 有限维度点查性能和稳定性测试【替换掉mongo的点查场景】
- 2 标签检索，分页方案优化【bitmap函数功能扩展】
- 3 doris2.x 冷热存储方案测试，降低部分大表存储成本
- 4 将湖仓方案，引入到生产环境
- 5 基于doris2.x 全文检索场景落地





获取更多社区动态与最佳实践

### Apache Doris 官方平台:

- Apache Doris 官网: [doris.apache.org](https://doris.apache.org)
- Apache Doris GitHub: [github.com/apache/doris/](https://github.com/apache/doris/)

### 获取更多峰会资料:

- Doris Summit 峰会官网: [doris-summit.org.cn](https://doris-summit.org.cn)
- Doris Summit 峰会回放: <https://space.bilibili.com/1196172099/channel/collectiondetail?sid=1824324>