

Doris 4.0

构建下一代Agent数据基座

SelectDB 朱伟



目录

Agent现状、趋势与难点

Agent对Data Infra的四大要求

Doris 4.0 for Agent全解析

数据分析迈向AI时代

Busniess Intelligence

Data is becoming ever more important
and an indispensable asset

1990s

Modern Data Warehouses

More flexible architecture, capability of
instant analysis and pay-as-you-go
costs

2022

New Mission in the AI Era

AI era is coming...

Now

Agent开枝散叶

智能问数

业务人员用自然语言提问，Agent实时转化为SQL并返回结果，人人都能秒级做数据分析。

.....

智能销售

销售Agent融合CRM、订单、互动数据，实时生成客户画像与跟进策略，提升转化率。

.....

智能客服

客服Agent实时查询订单与用户画像，秒级解决客户问题，减少人工介入。

研发助手

研发Agent整合文档、代码库、实验数据，为开发人员提供秒级知识检索与方案推荐。

自然语言
实时分析

实时响应
上下文增强

多源联邦
客户洞察

多模态
知识检索

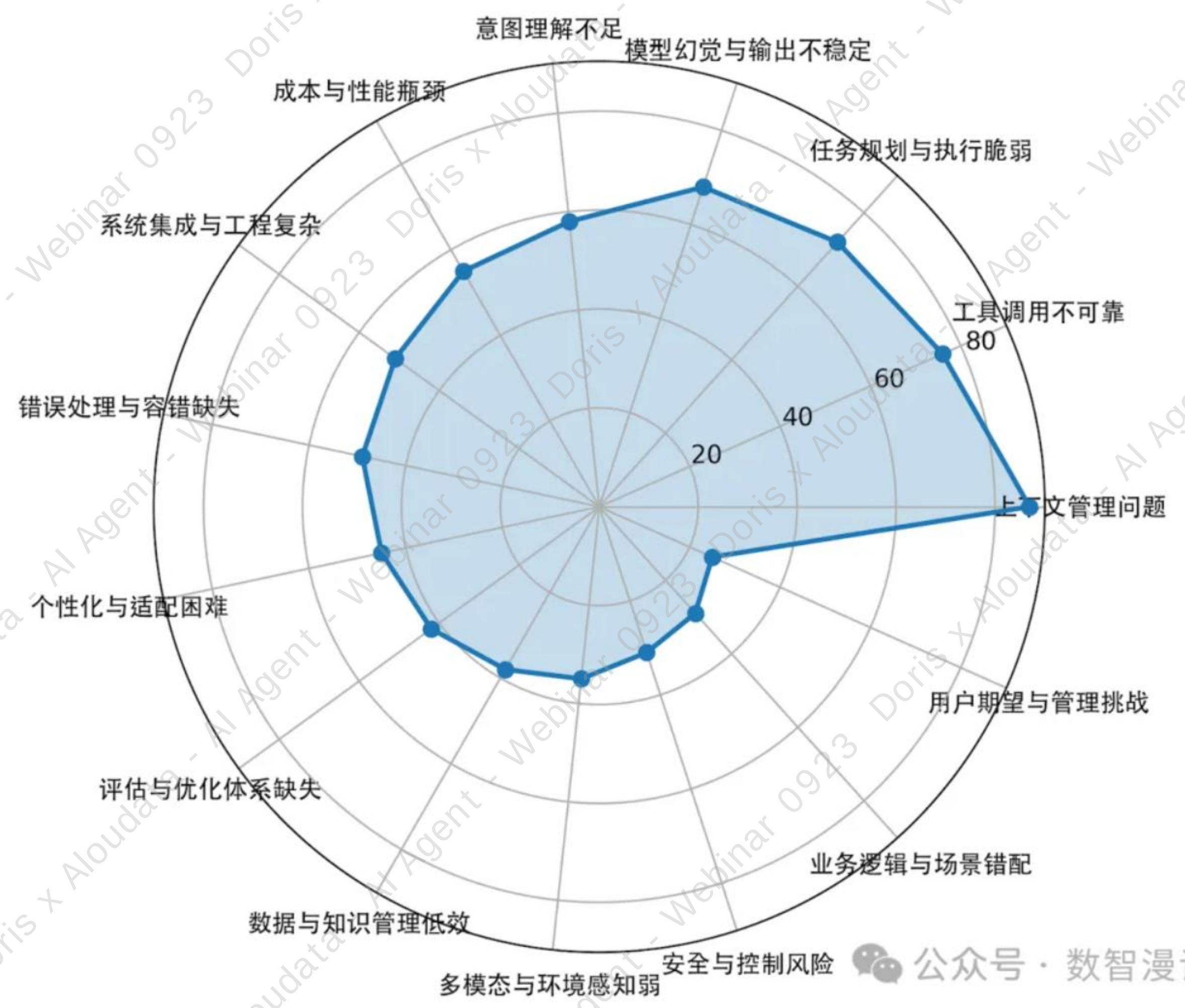
Agent落地难点

AI

- 模型输出不稳定
- 工具调用不可靠
- 成本与性能瓶颈
- ...

Data

- 上下文缺失
- 知识陈旧
- 跨系统数据管理
- 跨模态数据管理



目录

Agent现状、趋势与难点

Agent对Data Infra的四大要求

Doris 4.0 for AI Agent全解析

Agent对Data Infra的四大要求



实时



多源



多模



可观测

实时

1 并发性能指数级挑战

并发量增长趋势

传统BI: 10-100 QPS
智能客服: 1,000-10,000+ QPS
推荐系统: 数万+ QPS
需要MPP架构支撑

电商平台Agent集群

客服Agent + 推荐Agent + 库存Agent
价格Agent + 营销Agent + 风控Agent
数百个Agent同时访问数据库
传统架构无法支撑

技术挑战

- 单点瓶颈问题
- 资源争用冲突
- 查询队列拥堵
- 需要分布式无主架构

2 实时性要求新高度

响应时间要求

传统BI: 分钟-小时级
Agent应用: 秒级-毫秒级
实时决策: 亚秒级响应
任何延迟都影响体验

金融风控场景

实时交易行为分析 + 历史信用记录
当前账户状态 + 欺诈风险评估
毫秒级风险判断, 零容忍延迟
延迟即风险

技术要求

- 秒级数据写入延迟
- 高并发主键等值查询
- 实时流处理能力
- 需要行列混存架构

3 智能查询优化挑战

查询模式变化

传统: 预定义查询路径
Agent: 动态生成查询
高度不可预测性
需要智能优化

复杂查询特征

多表关联分析 + 复杂查询依赖图
多维度聚合计算 + 实时指标计算
探索性数据发现 + 上下文关联
传统优化器失效

解决方案

- CBO智能优化器
- 多种索引策略
- 向量化执行引擎
- 适应动态查询模式

数据规模爆炸式增长

传统BI: MB-GB级数据量 → Agent应用: GB-TB级实时处理 → 大型企业: PB级日增数据

关键数据: 单次Agent请求触发数十到数百个查询 | 中等企业需支持数千并发Agent | 每个Agent每秒发起数次到十几次查询

核心挑战: 传统数据库架构在Agent时代面临性能、实时性、智能化三重挑战, 急需新一代数据平台

多源

企业数据高度分散，并非全部集中化

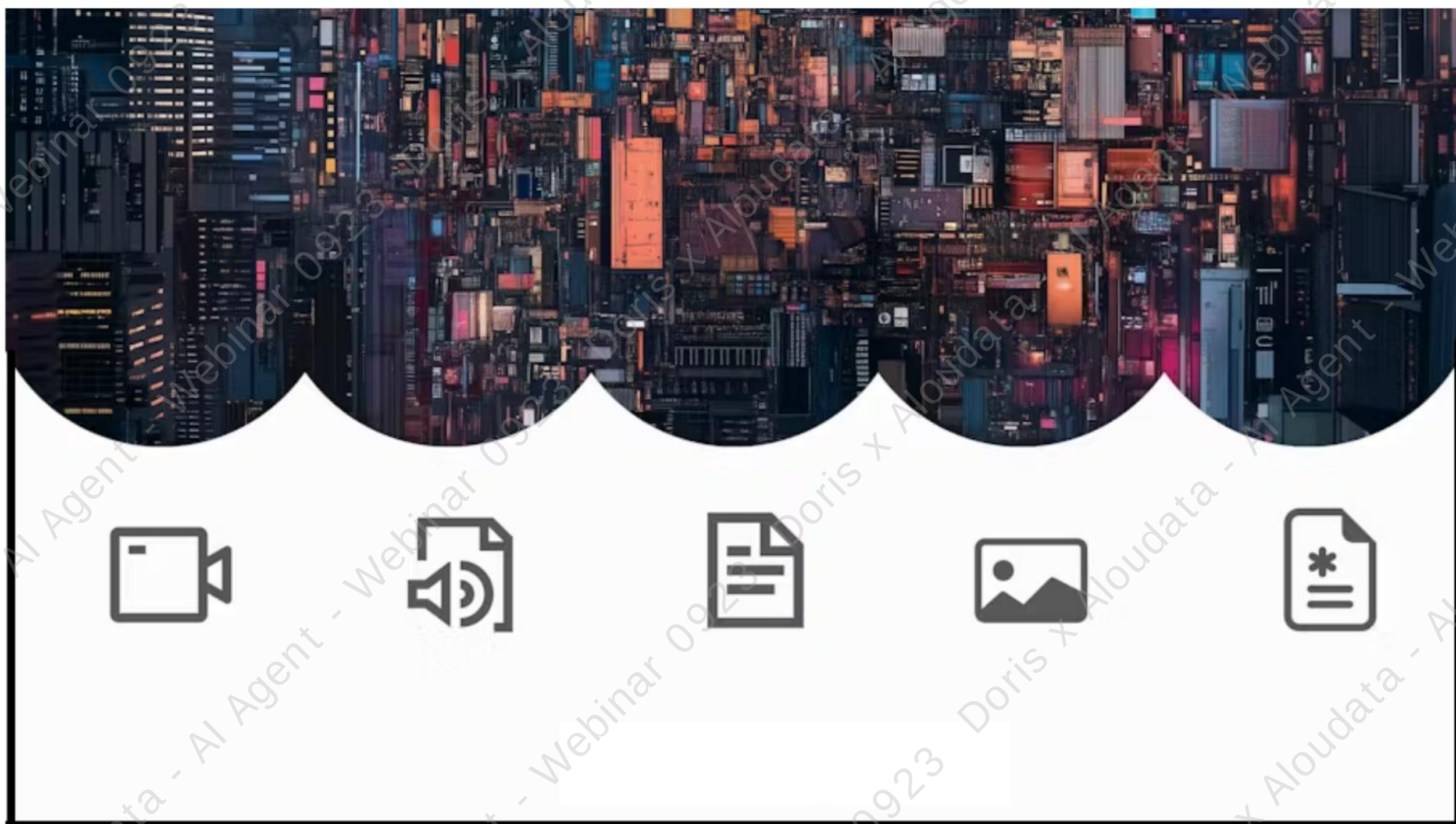
- 数据存储在不同系统:
 - 业务数据库 (MySQL、Postgres)
 - 数据湖 (Iceberg、Hudi)
 - 流日志 (Kafka)
 - SaaS/第三方API (CRM、ERP)

Agent要处理“跨域任务”，天然需要联邦能力

- 一个销售Agent要给出“客户下季度的续费风险预测”:
 - 要查**CRM**里的合同信息
 - 要查**客服系统**里的工单
 - 要查**产品日志**里的使用数据

湖仓系统	数据库	文件格式	元数据服务	数据存储
• Hive • Iceberg • Hudi • Paimon • Delta Lake • Kudu • Bigquery • Max Compute • Lakesoul • ...	• MySQL • PostgreSQL • Oracle • SQL Server • IBM Db2 • ClickHouse • SAP HANA • OceanBase • Elasticsearch • ...	• Parquet • ORC • Text • Json • SequenceFile • RCFile • ...	• Hive Metastore • AWS Glue • Unity Catalog • Aliyun DLF • Iceberg Rest Catalog • Filesystem • ...	• HDFS 及兼容存储 • AWS S3 及兼容存储 • Azure Storage • ...

多模



企业知识本身是多模态的

- 结构化数据：订单、用户、交易、日志表格。
- 半结构化数据：JSON配置、系统日志、埋点数据。
- 非结构化数据：文档、合同、客服对话、邮件、评论、研发文档。
- 感知数据：图片、音频、视频、传感器信号。

Agent需要跨模态推理

智能销售助手可能需要：

- 结构化：**CRM**（合同金额、到期日）、财务数据（**MRR**）
- 时序/半结构化：产品使用日志（活跃用户、**API** 调用量）
- 文本（非结构化）：邮件线程、通话转录、支持工单
- 文档（向量）：内部成功案例、过往报价单、白皮书
- 外部信号（可选）：行业新闻、竞争对手信息（网页抓取）

可观测



1. 模型输出不稳定，容易产生“幻觉”或错误

可观测系统记录每次推理的输入、模型版本、上下文、证据来源，可以追溯为什么产生不同结果，并对比输出差异。

2. 成本控制 — 避免不必要的资源浪费

通过可观测记录每次调用及命中率，优化缓存、重用历史计算结果，降低资源消耗。

3. 业务效果评估 — 量化Agent价值

可观测系统可以收集指标：成功率（问题是否解决）、平均响应时间、人工干预频率等，用量化数据持续迭代模型与策略。

4. 调优与问题定位 — 快速定位异常原因

追踪每次推理涉及的数据源、模型版本、向量索引版本，记录中间推理结果（embedding、topK检索结果），快速回溯问题并修复

目录

AI Agent现状、趋势与难点

Agent对Data Infra的四大要求

Doris 4.0 for Agent全解析

Apache Doris: Open-Source Real-Time Analytics Database



2013 Project Creation

2017 Open Source

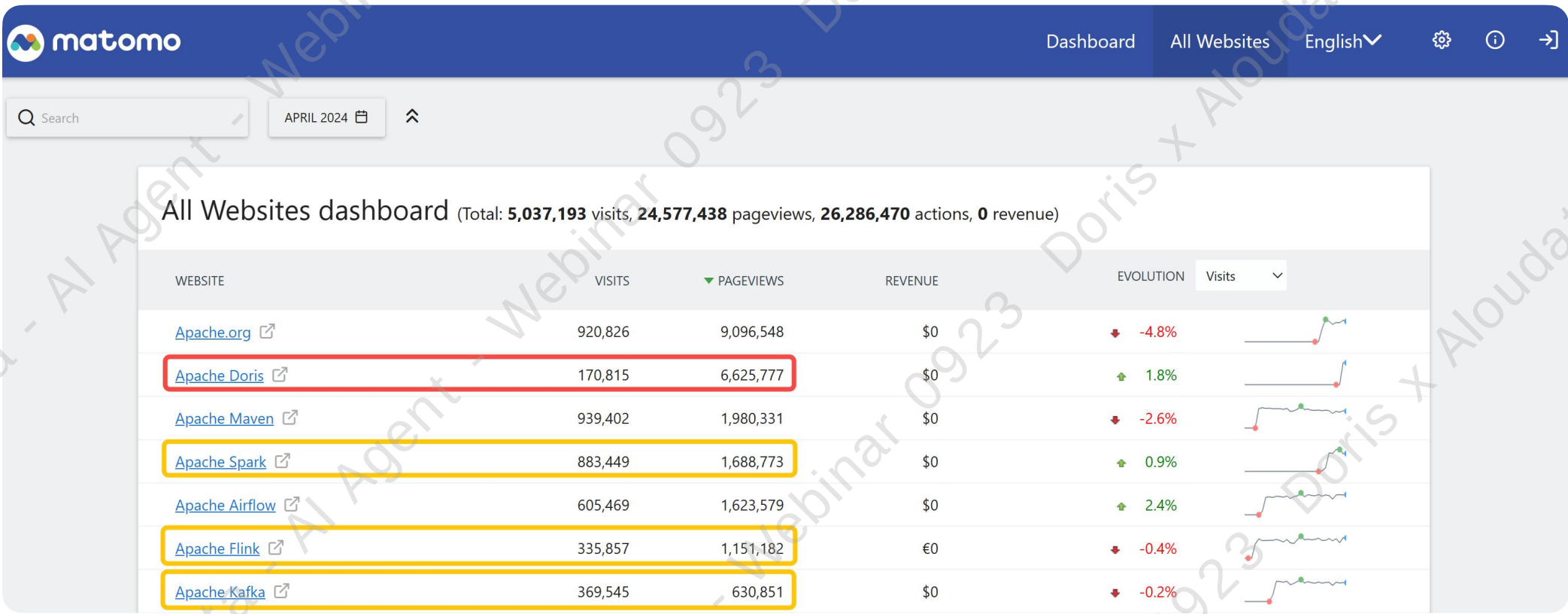
2022 ASF Top Project

13k+ GitHub Stars

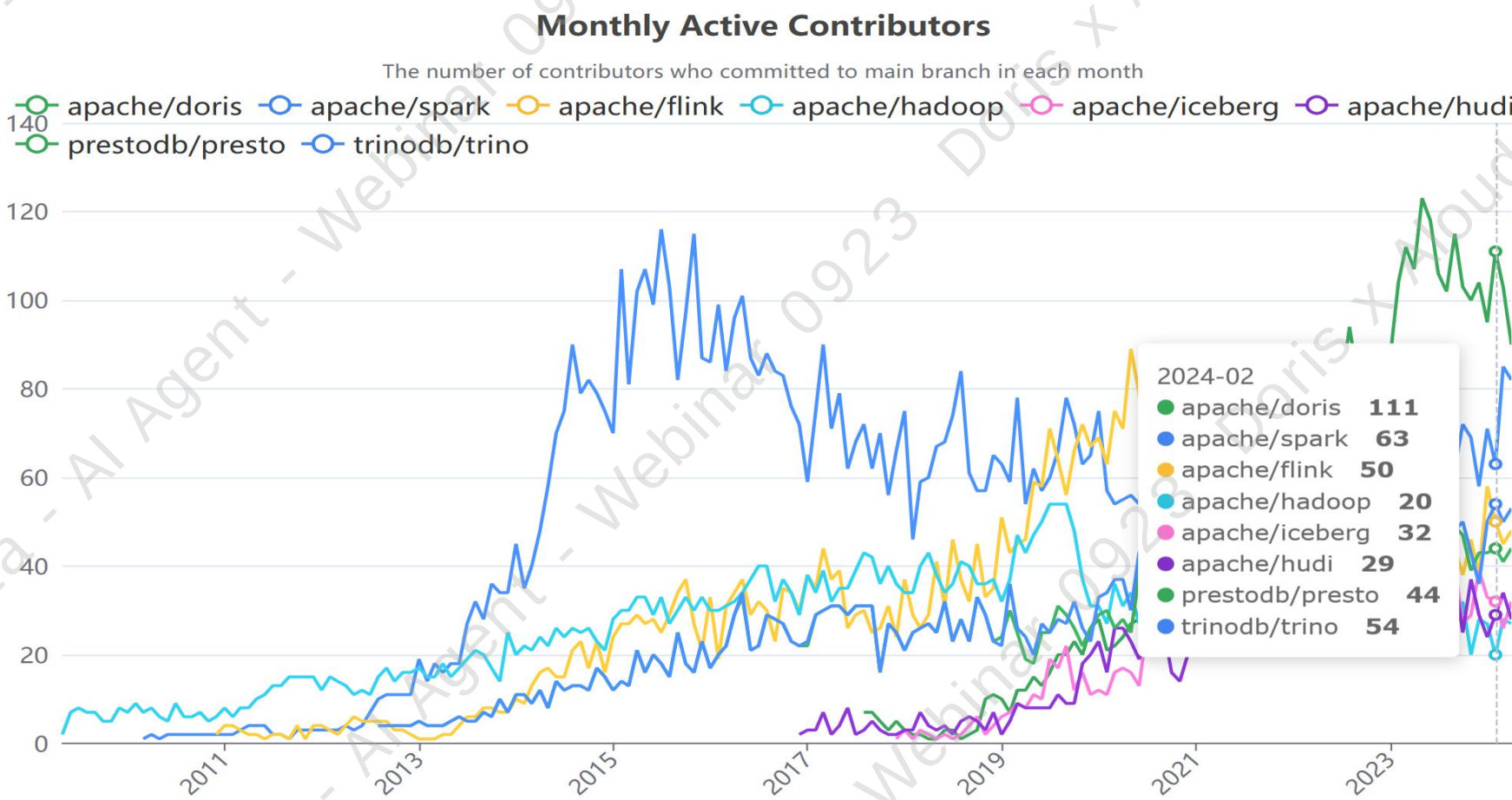
640+ Contributors

5000+ Enterprises

#1 Daily Page Views of Apache Project Websites



#1 Monthly Active Contributors



Doris 4.0 for Agent

实时

极致性能

- 分区裁剪
- Data Trait 能力释放

多源

湖仓融合

- 分区增量物化视图
- Snapshot增量物化视图
- 查询系统表
- Batch Incremental Read
- Branch/Tag管理
- Time Travel

多模

向量检索

- 向量检索
- LLM Function

可观测

灵活类型

- Variant 数万子列支持
- Variant Schema Template
- Inverted Index V3
- 分词器

极致性能

RTABench (Real-Time Analytics Benchmark) 最新榜单公布

Apache Doris 以绝对领先的成绩登顶，结果表明，Apache Doris 在实时分析场景下性能是 ClickHouse 的 6 倍，PostgreSQL 的 30 倍，MongoDB 的 100 倍。

RTABench 官网地址：
<https://rtabench.com>

RTABench 是一套专为实时分析 (Real-Time Analytics) 场景设计的开源数据库性能基准测试工具 (Benchmark)。它基于 Clickbench 框架开发，通过引入更贴合真实业务的数据模型与查询集，有效弥补了传统基准测试工具在“实时性测试”维度的不足，为实时分析类数据库的性能评估提供了更具参考价值标准。



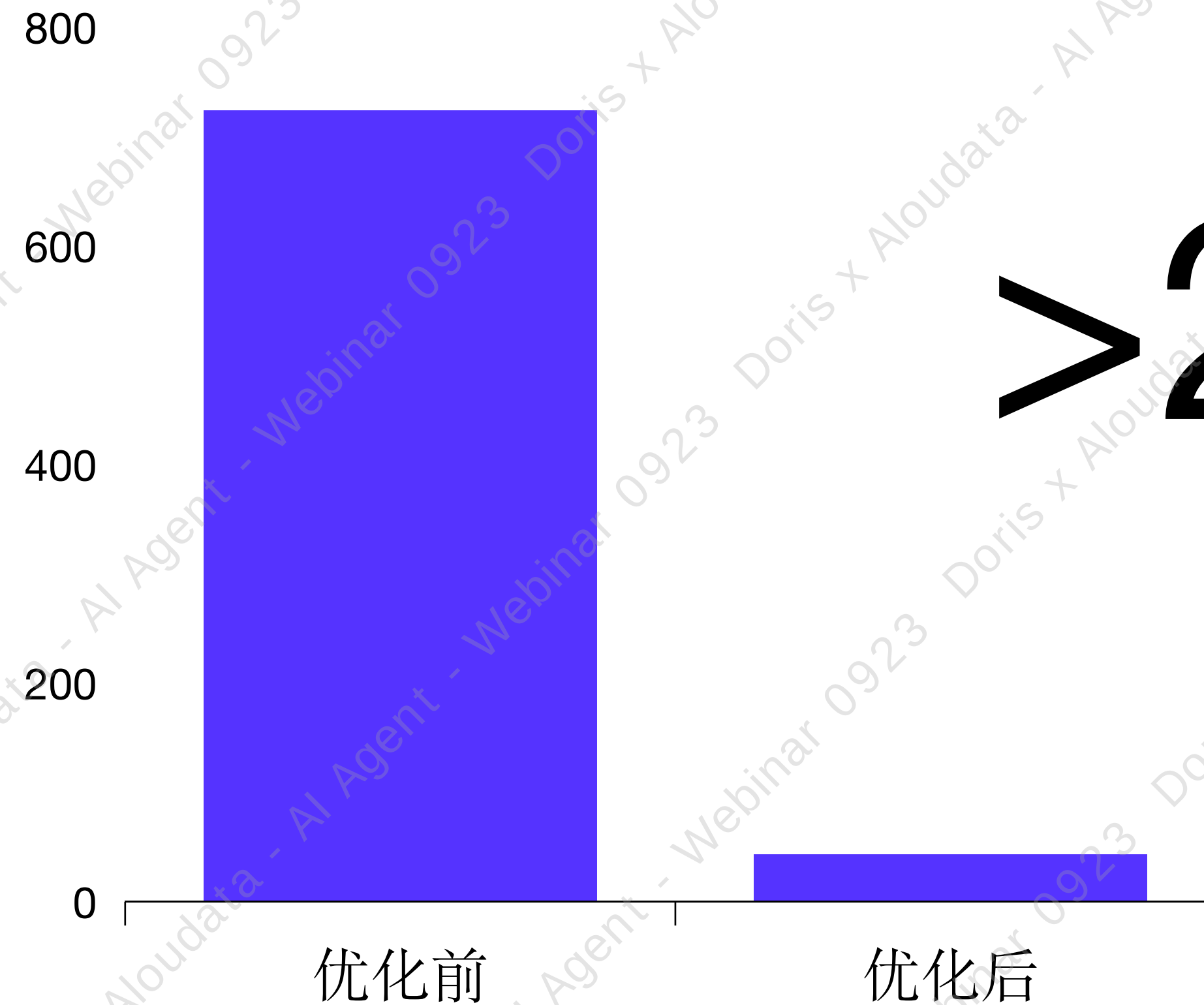
分区裁剪性能优化

性能优化点

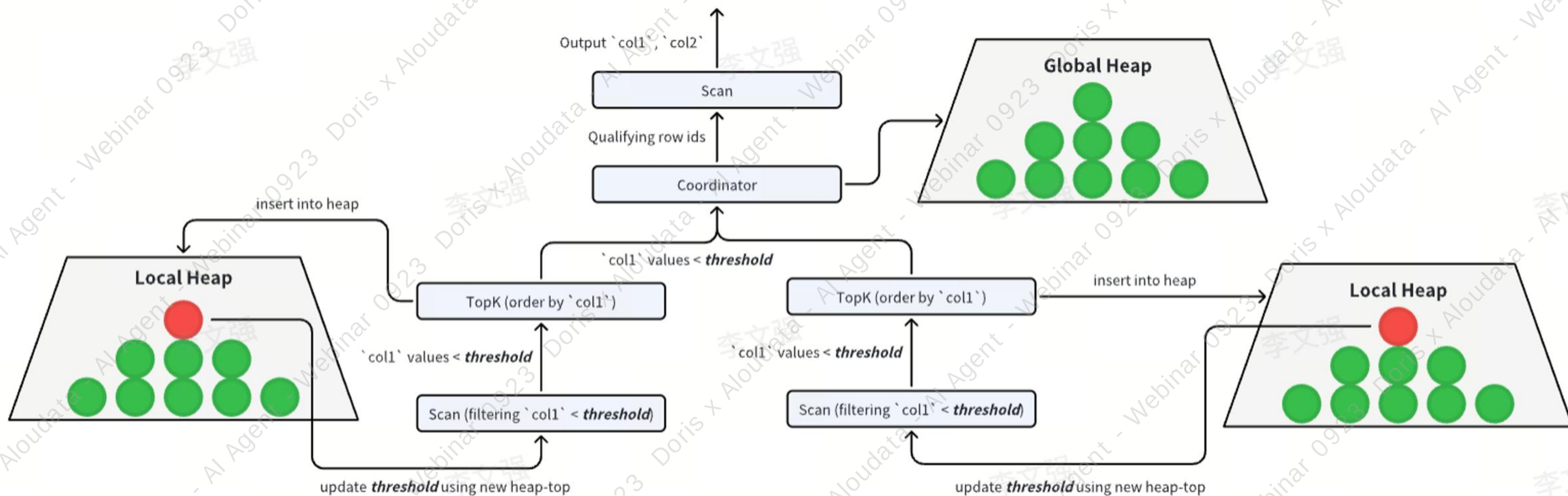
针对十万级分区场景，为了提升分区裁剪的性能，DORIS 在分区裁剪上使用了二分查找，加速分区过滤。同时，全链路细致优化了分区裁剪的性能。

适用范围提升

- 支持单调时间函数表达式
- `DATE_TRUNC(p1, month) > '2020-02-01'`

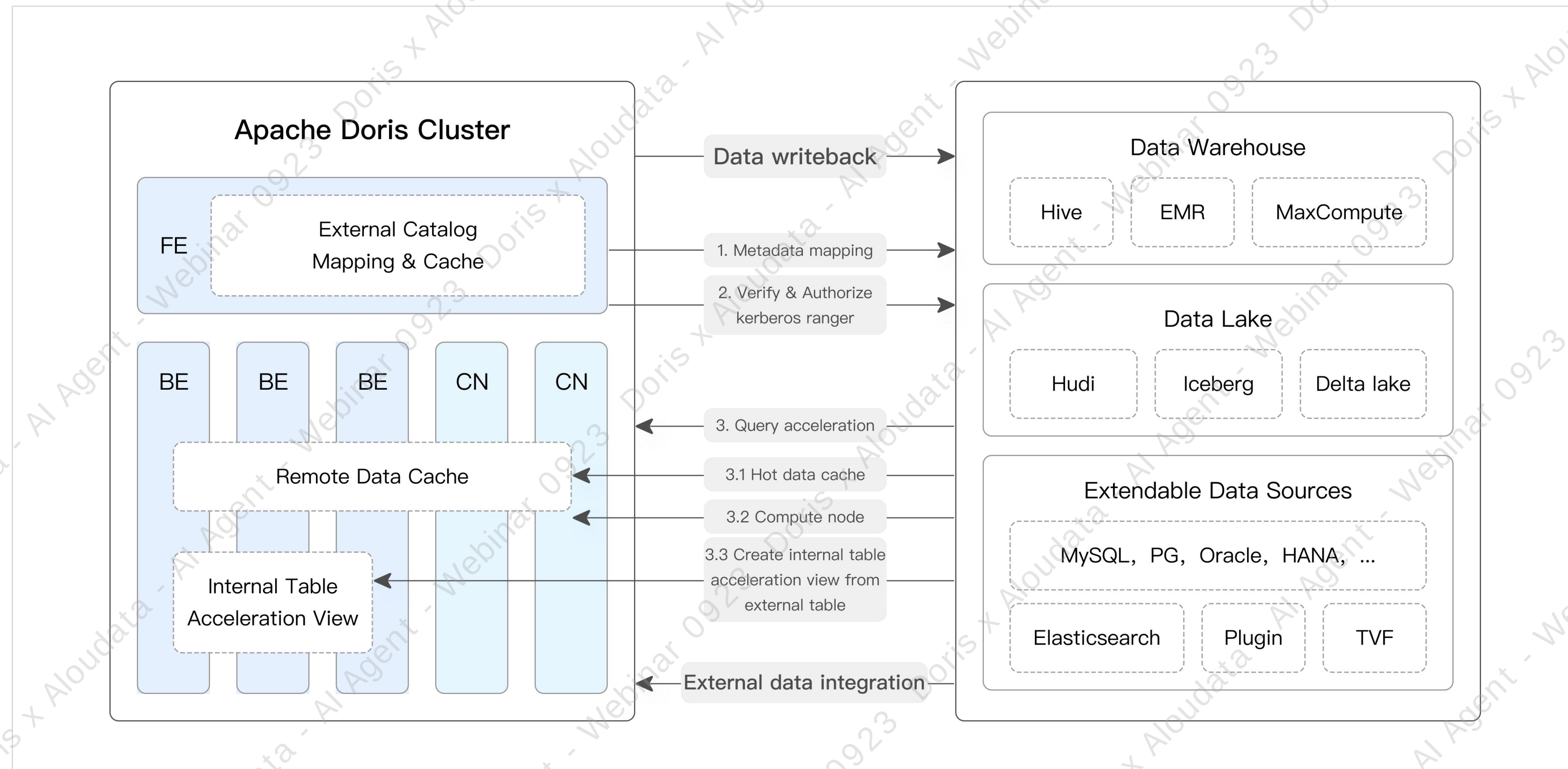


TopK性能优化



SELECT 'col1', 'col2' FROM 'tbl' ORDER BY 'col1' LIMIT K

湖仓融合



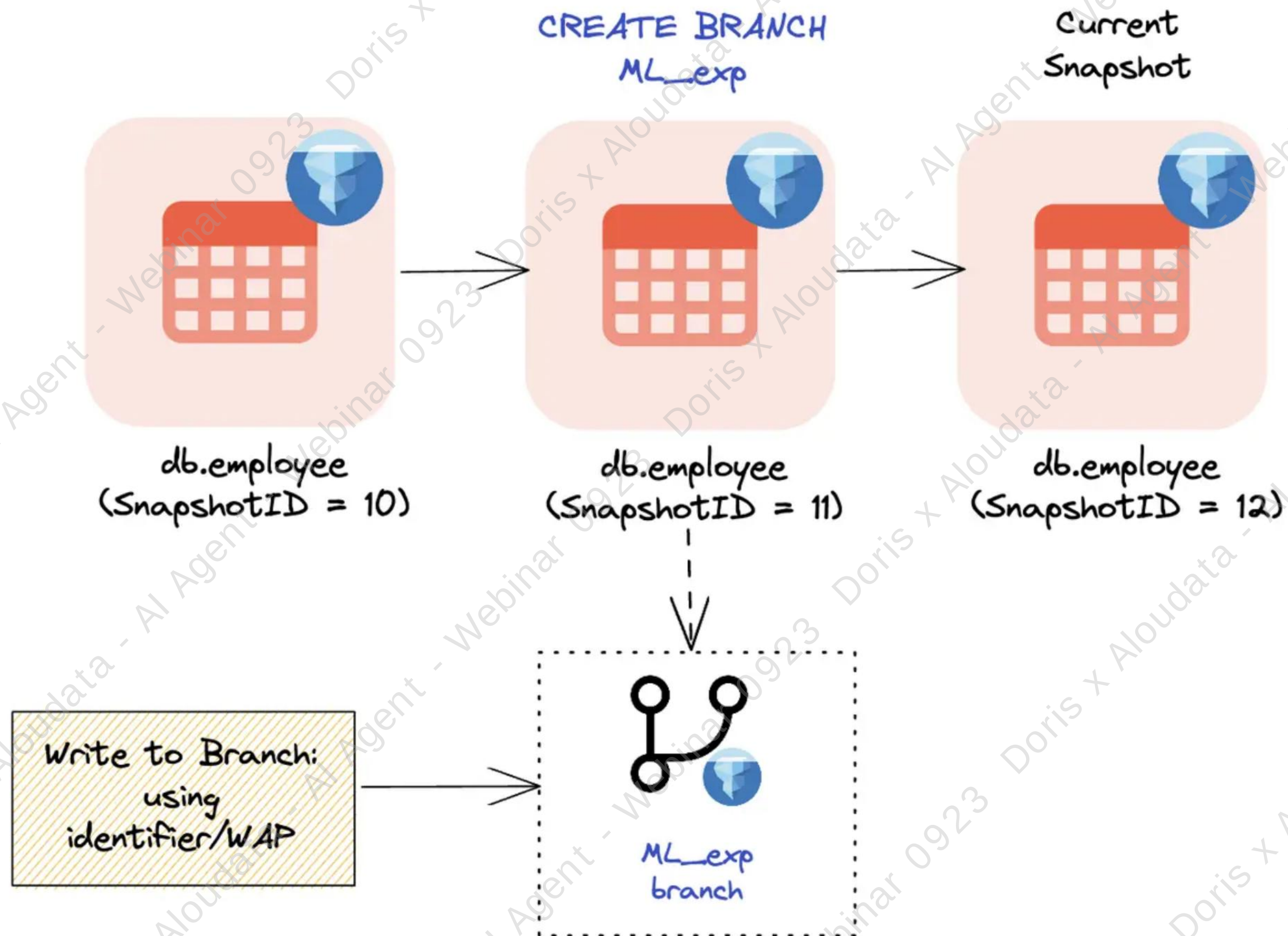
Open Data Lakehouse

Lakehouse is designed for AI era

Why Choose Doris

- 元数据映射、cache 和 自动刷新利用 Doris 高效的分析引擎加速
- 热数据 Cache 到本地
- 支持弹性计算节点，实现计算弹性加速
- 外表处理结果可写入内表，形成加速视图
- 湖仓查询加速性能是 Presto/Trino 的 3-5 倍

Icerberg / Paimon 第一公民



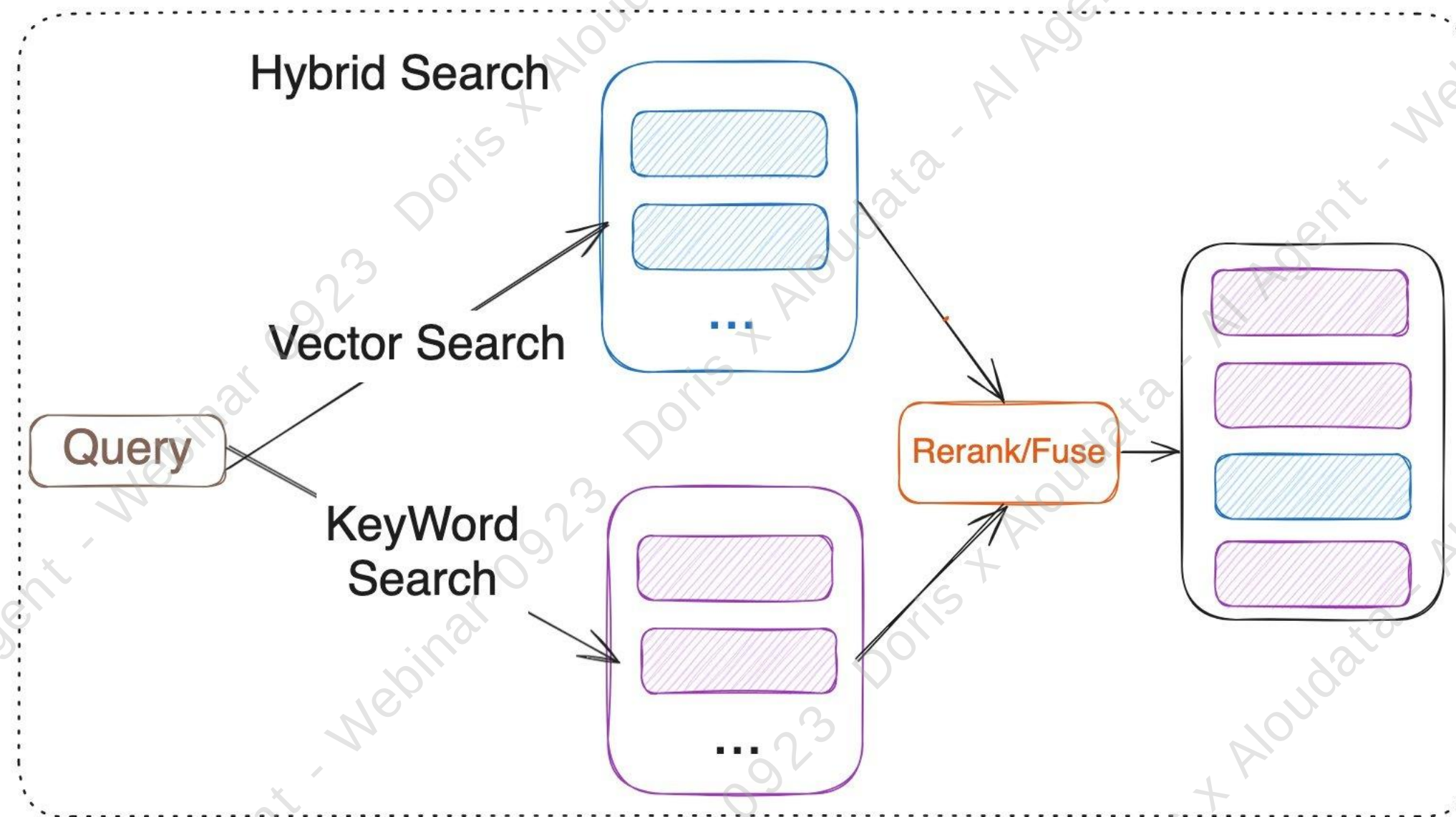
Icerberg

- 完整支持 Iceberg 的 Branch / Tag 能力 (tbl@tag)
- 丰富的 Iceberg 系统表支持 (tbl\$files)
- Iceberg 视图查询
- 通过 ALTER 语句修改 Iceberg 表结构

Paimon

- 支持 Paimon 的 Branch / Tag 读取
- 丰富的 Paimon 系统表支持
- 支持 Paimon 增量数据查询

Hybrid Search



```
SELECT
  id,
  name,
  description,
  price,
  1 - SQRT(SUM(POWER(embedding - [0.1, 0.2, ..., 0.128], 2))) AS
  similarity_score
FROM
  products
WHERE
  (name LIKE '%手机%' OR description LIKE '%手机%')
GROUP BY
  id, name, description, price
ORDER BY
  similarity_score DESC
LIMIT 10;
```

Why Vector-Capable General-Purpose Databases are Better for Enterprise GenAI ?

- Lack of Hybrid Query Capabilities
- Limited Integration with Structured Data
- Operational Complexity and Increased Costs

LLM Functions

文本理解与分析

AI_CLASSIFY

智能文本分类

- 多标签分类支持
- 自然语言分类标准
- 语义理解分类
- 动态分类规则

AI_SENTIMENT

情感倾向分析

- 复杂情感识别
- 情感强度评估
- 多维度情感分析
- 上下文情感理解

AI_EXTRACT

信息结构化提取

- 关键信息识别
- 实体关系提取
- 结构化输出
- 语义上下文理解

AI_SIMILARITY

语义相似度计算

- 深度语义理解
- 跨语言相似度
- 内容去重匹配
- 智能推荐支持

AI_GENERATE

智能内容生成

- 创意文案生成
- 建议方案生成
- 风格化写作
- 上下文感知生成

AI_TRANSLATE

多语言智能翻译

- 上下文感知翻译
- 专业术语处理
- 文化背景适配
- 多语言数据分析

AI_SUMMARIZE

智能文本摘要

- 关键要点提取
- 结构化摘要
- 可控摘要长度
- 多文档汇总

AI_FIXGRAMMAR

语法纠错优化

- 语法错误修正
- 表达优化建议
- 可读性提升
- 风格统一化

AI_MARK

敏感信息脱敏

- 隐私信息识别
- 智能脱敏处理
- 合规性保障
- 数据安全防护

AI_FILTER

智能内容过滤

- 语义匹配过滤
- 复杂条件筛选
- 内容质量控制
- 智能内容审核

典型应用场景组合

智能客服系统

CLASSIFY + SENTIMENT + EXTRACT + GENERATE

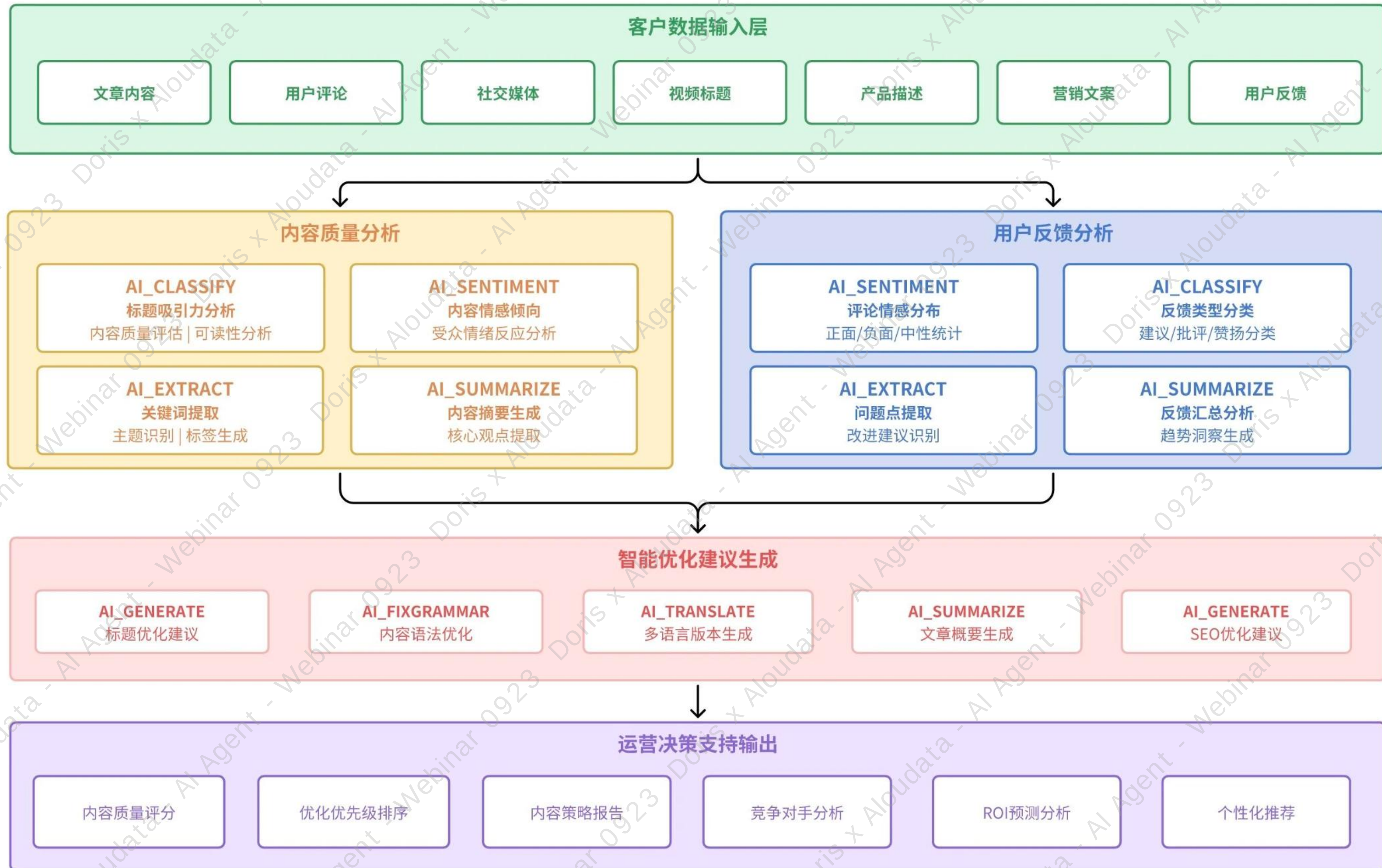
内容运营分析

SENTIMENT + SUMMARIZE + GENERATE + CLASSIFY

金融风控决策

EXTRACT + CLASSIFY + MASK + GENERATE

LLM Functions - 场景



LLM Functions – 示例

```
INSERT INTO claims VALUES
```

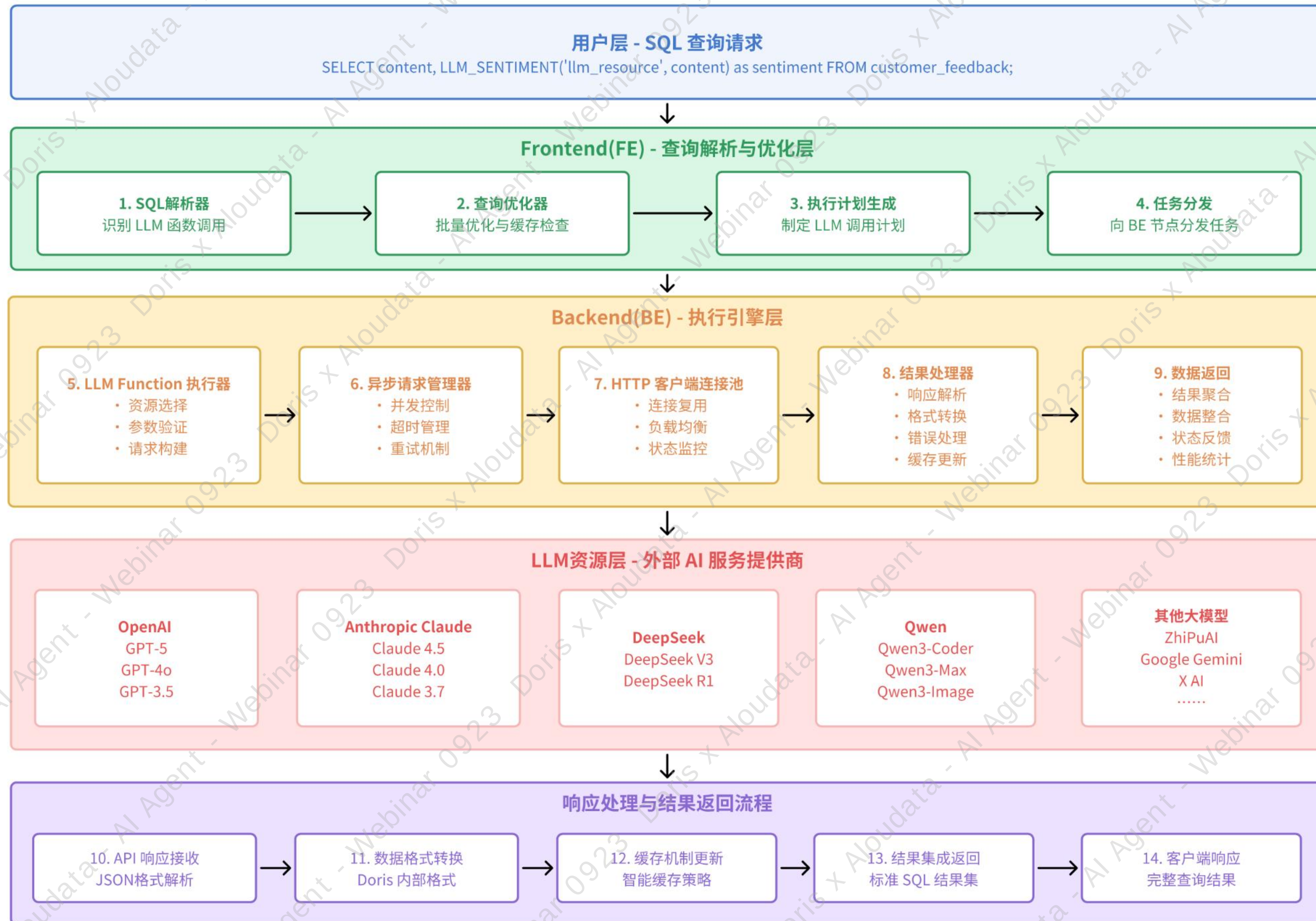
```
(1, 101, '2025-08-18', '昨天下午三点左右，我在东三环辅路开车时，与前车发生了追尾。'),  
(2, 102, '2025-08-18', '上周五在公司楼下，我不小心扭伤了脚踝，需要理赔医疗费用。'),  
(3, 103, '2025-08-18', '8月17日夜里家中管道破裂，导致部分家具被水浸泡。'),  
(4, 104, '2025-08-18', '晚上喝酒后开车回家与其他车辆发生了碰撞。'),  
(5, 105, '2025-08-18', '早上8点，在去上班的路上，发现车辆停放时被刮擦。');
```

```
SELECT
```

```
  c.claim_id,  
  c.incident_description,  
  llm_classify(c.incident_description, ['交通事故', '人身意外', '财产损失', '其他']) AS incident_category  
FROM claims AS c  
JOIN policies AS p ON c.policy_id = p.policy_id  
WHERE  
  p.policy_type = '车险' AND LLM_FILTER(CONCAT('下列情形是否支持保险赔偿: ',  
c.incident_description));
```

可以利用 **LLM_CLASSIFY** 函数对事件性质进行智能分类，并通过 **LLM_FILTER** 函数对事件进行有效性校验，以筛选出符合理赔标准的有效事件。

LLM Functions – 机制



可观测

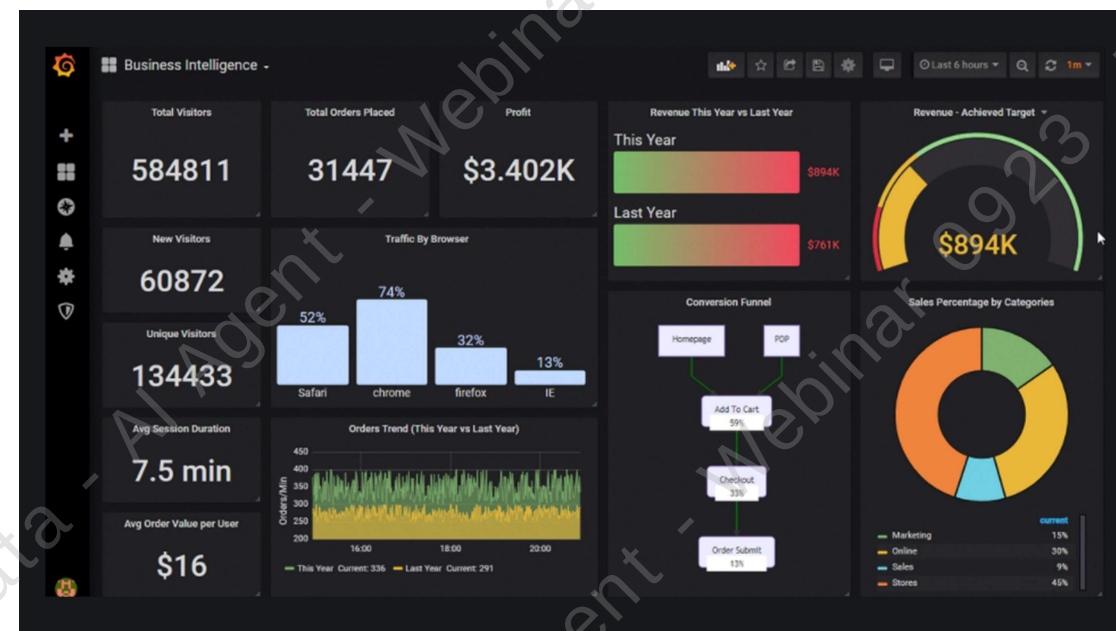
From Cloud Native + Microservice to LLM + AI Agent

History of Observability

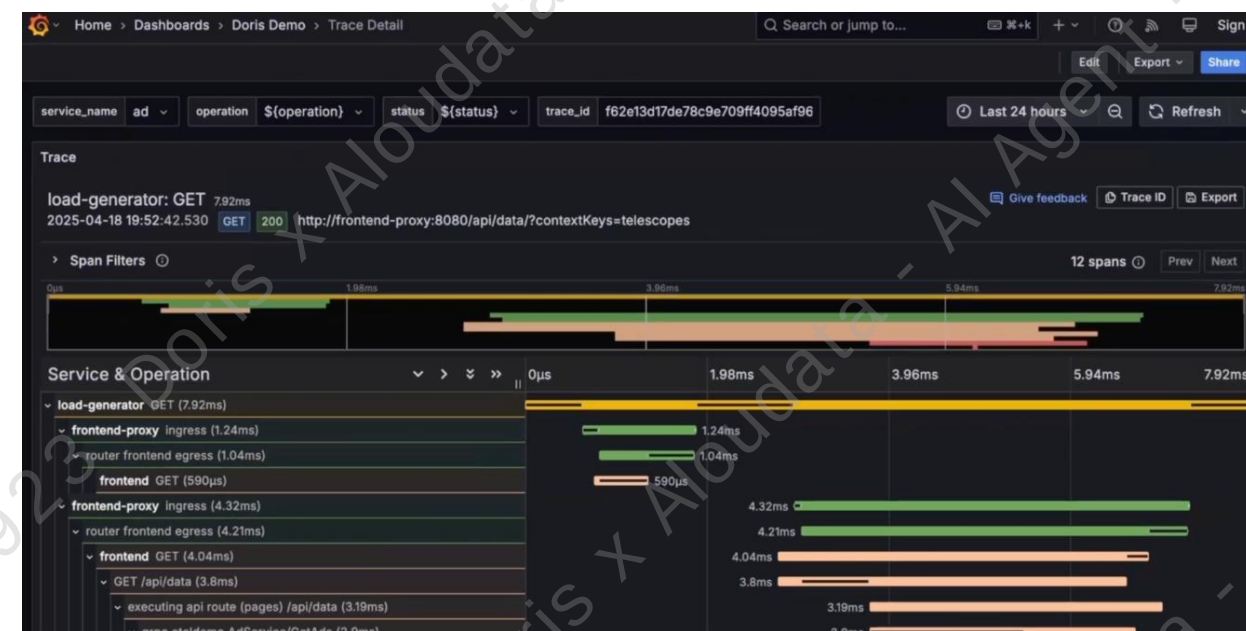
2013: The concept of observability began to gain traction as companies like Twitter started to adopt it to manage their increasingly complex distributed systems.

2020: Observability became a hot topic in DevOps, with more companies recognizing its importance in managing complex IT architectures

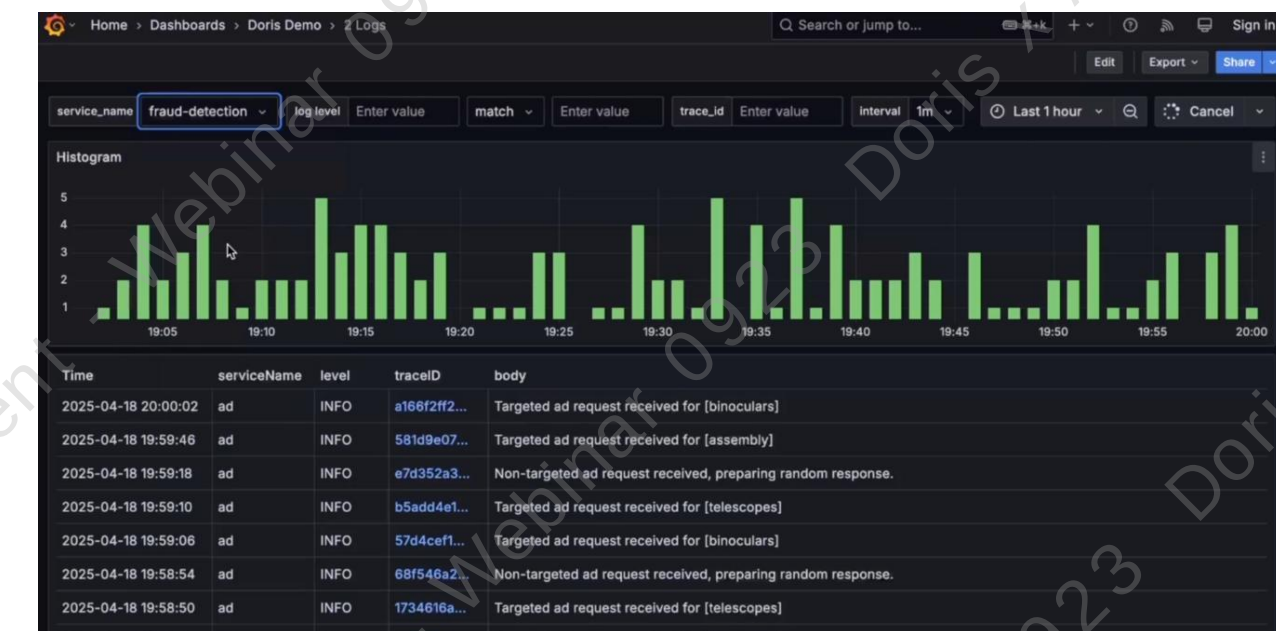
Metrics



Traces



Logs



Why Choose Doris

- 10x Cost Effective Compared to Elasticsearch
- Flexible Semi-Structured Data Variant Type
- Open Integration for ELK, OpenTelemetry, Grafana and more

可观测性——Variant 新能力

Schema Less

- 通过稀疏列能力，严控元数据和索引的空间占用，将支持的子列数量提升至**万列**级别。
- 通过 Schema 模板能力，能够更精细的控制子列的行为，让数据导入和查询更有的放矢。可以**单独为子列设置索引**行为

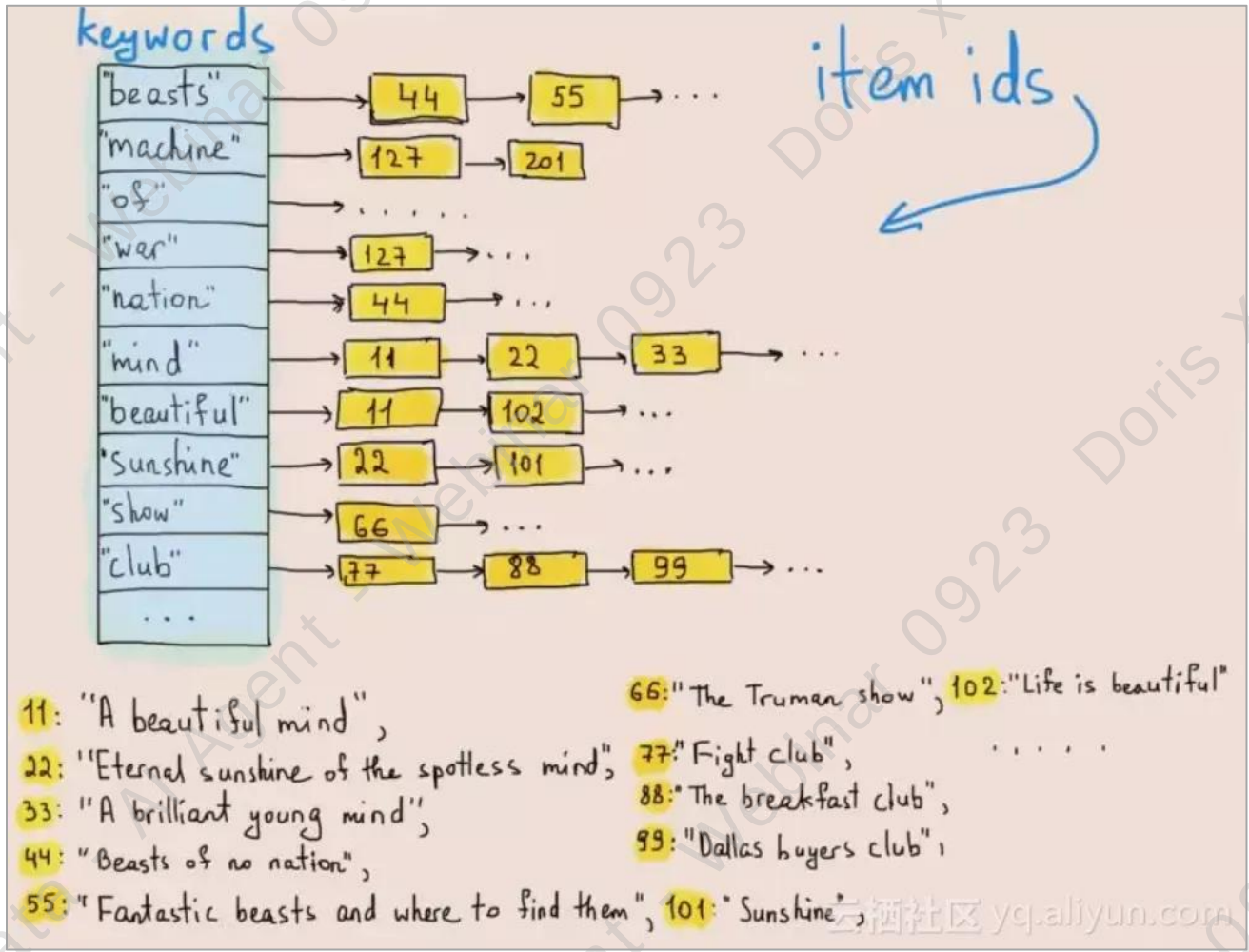
```
CREATE TABLE IF NOT EXISTS ${table_name} (  
  k bigint,  
  v variant<'content' : STRING>  
)  
DUPLICATE KEY(`k`)  
DISTRIBUTED BY RANDOM BUCKETS 5  
properties("replication_num" = "1");  
  
// insert into 写入或是stream load  
insert into ${table_name} values(1, '{"a" : 1}');  
  
// 查询需要带上cast，存储层尽可能消除cast加速查询  
select cast(v:a as int) from ${table_name} where cast(v:a as int) > 1
```

	存储空间	第一次查询	第二次查询	第三次查询
预定义静态列	12.618 GB	233.79 s	86.02 s	83.03 s
Variant 类型	12.718 GB	248.66 s	94.82 s	92.29 s
JSON 类型	35.711 GB	大部分超时	789.24 s	743.69 s

可观测性——倒排索引新分词器

倒排索引

- 倒排索引实现多维度快速检索
- 支持文本分词，实现全文检索，加速日志场景
- IK、ICU、Basic和自定义分词



```
mysql> CREATE INDEX idx_request ON httplogs(request) USING INVERTED
Query OK, 0 rows affected (0.03 sec)

▪ fulltext search query

-- search for request contains word 'login'
SELECT * FROM httplogs WHERE request MATCH 'login';

-- search for request contains word 'login' or 'error'
SELECT * FROM httplogs WHERE request MATCH 'login error';

-- search for request contains word 'login' and 'error'
SELECT * FROM httplogs WHERE request MATCH_ALL 'login error';

▪ equal query

SELECT * FROM httplogs WHERE status = 404;

▪ range query

SELECT COUNT() FROM httplogs WHERE size > 1024;
```

- 倒排索引实际业务场景效果

某手机厂商用户行为分析场景，使用倒排索引后，并发查询性能提升 5 - 90 倍

Query	V1.1.5 50 并发	V1.2.4 50 并发	V2.0 倒排索引 50 并发	V1.1.5 100并发	V1.2.4 100 并发	V2.0 倒排索引 100 并发	V1.1.5 单并发	V1.2.4 单并发	V2.0 倒排索引 单并发
SQL1 平均执行时间 (秒)	0.151	0.069	0.026	0.130	0.140	0.023	0.04	0.04	0.02
SQL2 平均执行时间 (秒)	0.699	0.274	0.022	1.178	0.397	0.048	0.07	0.06	0.03
SQL3 平均执行时间 (秒)	0.322	0.125	0.016	0.653	0.284	0.028	0.06	0.03	0.02
SQL4 平均执行时间 (秒)	2.446	0.406	0.036	4.601	0.972	0.055	0.09	0.08	0.06

Thanks !

