



MongoDB

高级应用场景及技术特点

锦木信息
唐峰

哪些场景适合MongoDB?

RDBMS, MongoDB?

NoSQL, MongoDB?

核心特点

Document Model

最佳的数据管理方式



Run Anywhere

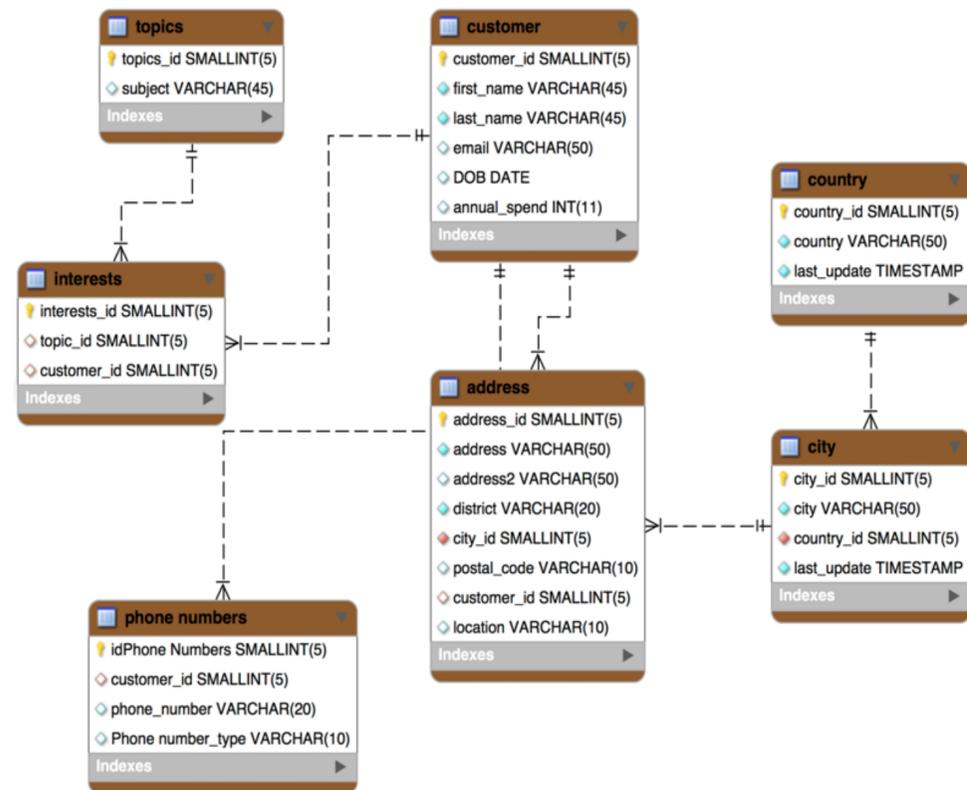
自由得运行于任意平台

Distributed System

智能得将数据放在需要的地方

文档模型

模型易用



```
{
  _id : 123456 ,
  first_name : "Mark" ,
  last_name : "Smith",
  city : "San Francisco" ,
  phones: [
    {
      number : "1-212-777-1212",
      dnc : true,
      type : "home"
    },
    {
      number : "1-212-777-1213",
      type : "cell"
    }
  ]
}
```

{ JSON }

举例

金融行业资产管理案例

1. 基础客户信息

① 客户名称 ② 客户地址 ③ 组织机构代码 ④ 营业执照号 ⑤ 社会统一信用代码

2. 客户行业信息

矿业行业：



- ① 煤炭生产许可证
- ② 矿井储量 (万吨)
- ③ 矿井位置 (经纬度)
- ④ 百万吨死亡率(%)

地产行业：



- ① 房地产资质证书
- ② 年开工面积 (平米)
- ③ 年销售面积(平米)

电力行业：



- ① 装机容量 (兆瓦)
- ② 年发电量 (兆瓦)

关系模型

传统关系型设计模式

传统方案1：宽表

客户ID	名称	地址	组织机构代码	营业执照	矿井储量	开工面积	发电量
100001	多多的电力	XXXX	XXXX	XXXX				XXXX
100002	大大的地产	XXXX	XXXX	XXXX		XXXX		

传统方案2：主从表

客户ID	名称	地址
100001	深深的煤业	XXXX



客户ID	METRIC_NAME	METRIC_VALUE
100001	生产许可证	mtsc017
100001	矿井储量	2000000
100001	矿井位置	138.2031 -124.4904
100002	百万吨死亡率	2%

文档模型

文档类型设计模式

{ JSON }

文档1

```
{
  name: "深深的矿业",
  orgId: "70107890X",
  industryInfo: {
    type: "采矿"
    coalCertificate: "mtsc017",
    location: [ 38.2031, -120.4904 ],
    mineReserves: 2000000,
    deathsByMT: 0.02
  }
}
```

文档2

```
{
  name: "大大的地产",
  orgId: "565425429",
  industryInfo: {
    type: "地产",
    estateCertificate: "fdczz0000001",
    areaPerYear : 897600.00,
    salesAreaPerYear : 300000.00
  }
}
```

储量小于2000万吨:

```
db.collection.find( { industryInfo. mineReserves : { $lt: 2000000 } } )
```

事务支持

4.0版本支持多文档事务

```
with client.start_session() as s:  
    s.start_transaction()  
    try:  
        collection.insert_one(doc1, session=s)  
        collection.insert_one(doc2, session=s)  
        s.commit_transaction()  
    except Exception:  
        s.abort_transaction()
```

和关系数据库的事务操作一样

- 语法基本一样
- 任何应用都可以使用
- 一个表或多个表中的多个文档都适用

ACID 保证

- 要么全成功，要么全失败

版本要求

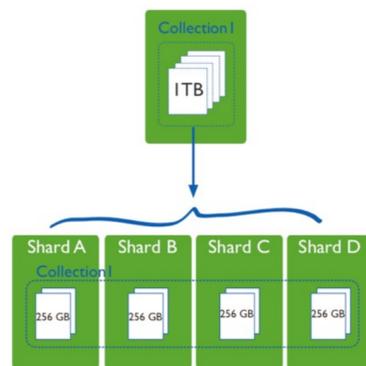
- MongoDB 4.0, 支持复制集，已经发布
- MongoDB 4.2: 支持分片，预计19年发布

文档模型

性能强大、功能丰富

性能强大

文档模型、自动分片
PB级数据库支持



功能丰富

地理位置查询



图查询

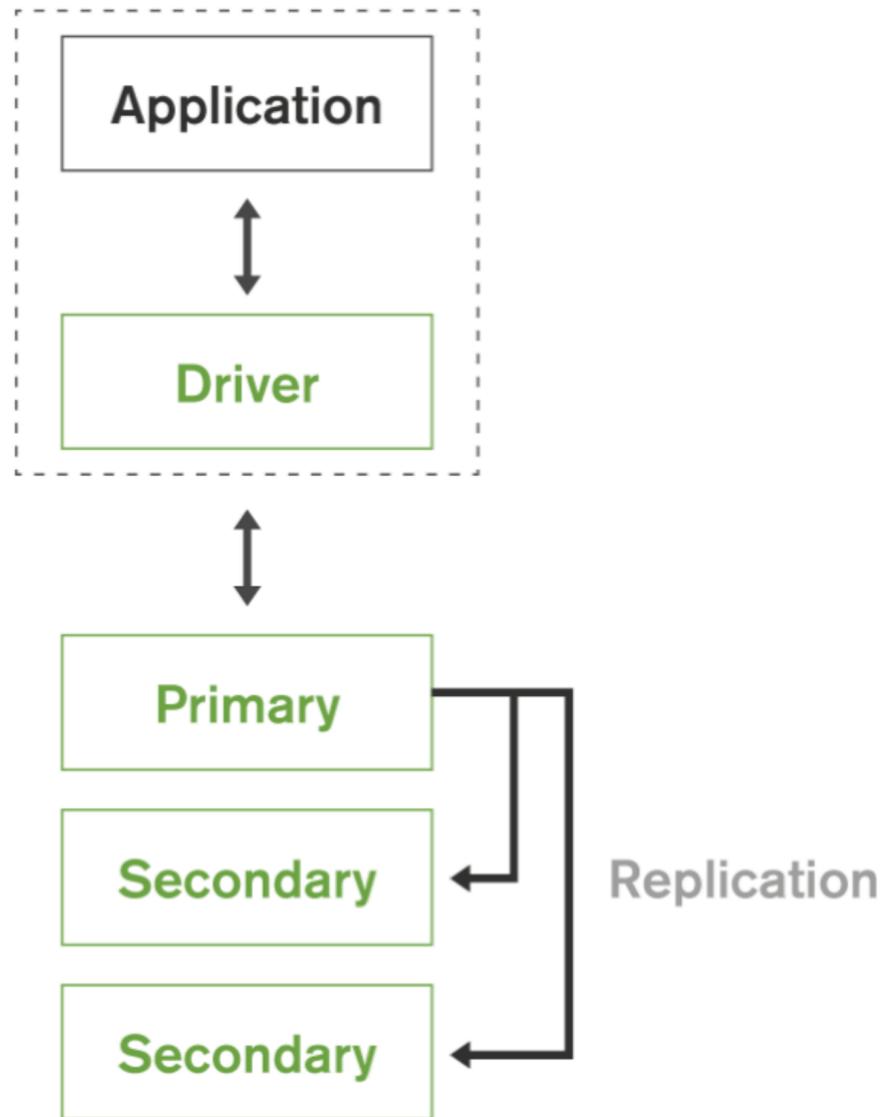


列表查询



分布式扩展

复制集 & 分片



复制集支持2-50个节点

自动恢复

关键能力：

- 高可用
- 容灾
- 维护升级

任务隔离：联机与分析作业隔离

支持不同存储引擎（WireTiger、In-memory）

分布式扩展

复制集 & 分片



分片策略：范围、哈希、标签

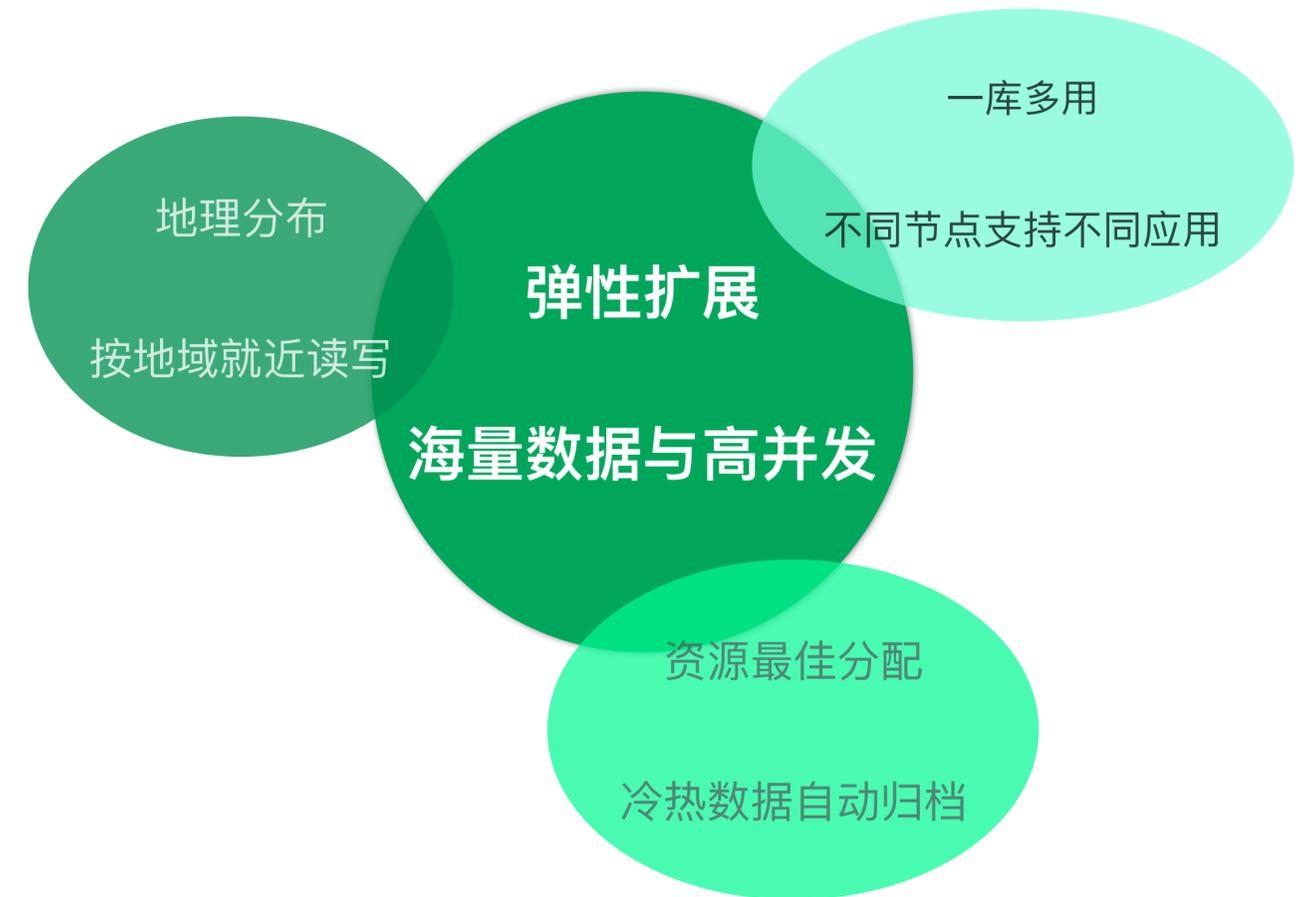
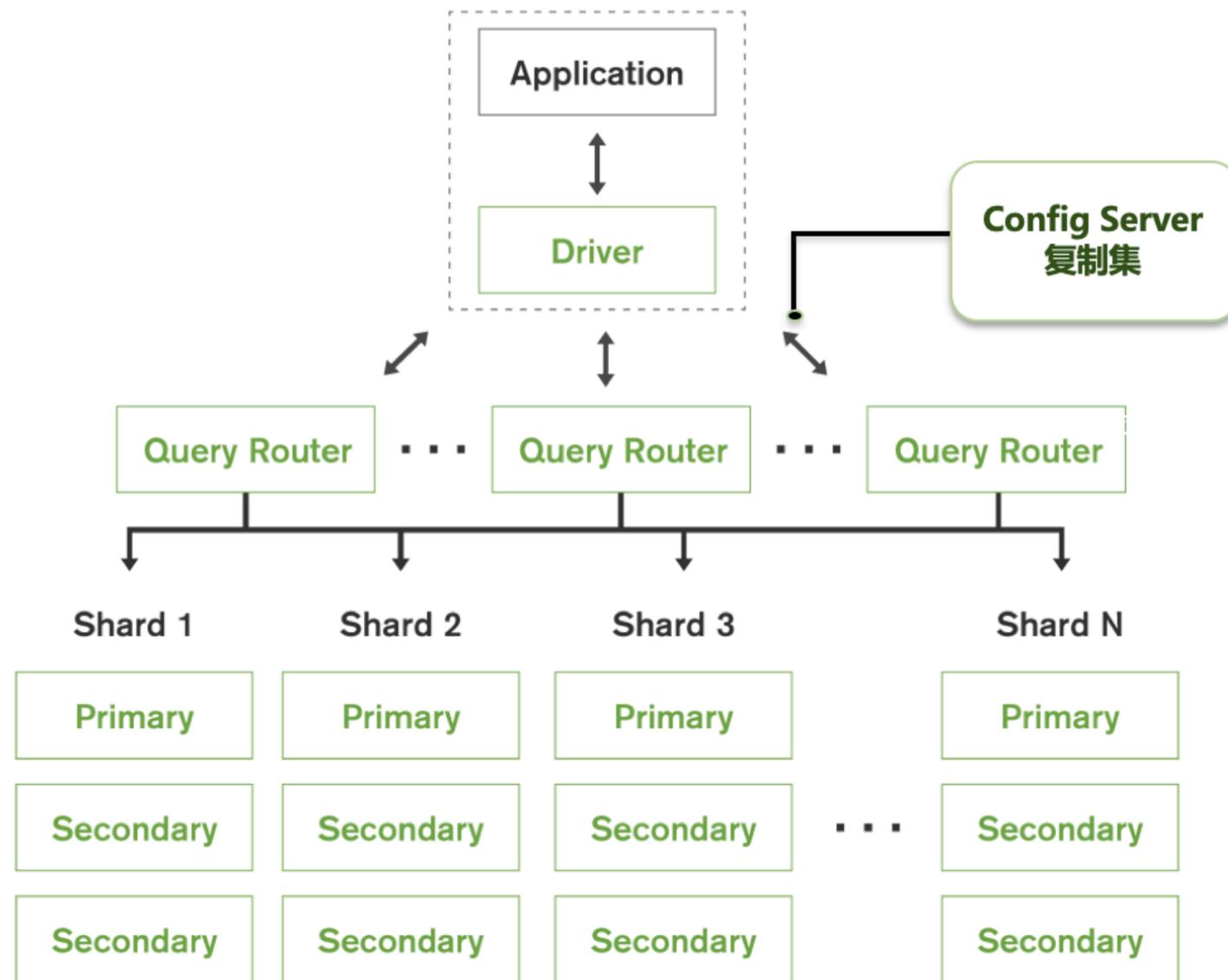
弹性扩展与收缩

数据自动均衡

应用完全透明

分布式扩展

复制集 & 分片



任意部署

本地、云端、移动端

桌面



本地服务器



私有云



混合云



公有云



数据库云服务



从桌面到云中均可部署运行



核心特点



最佳的数据管理方式

- 文档模型易用
- 模型变更零代价
- 性能表现优异
- 查询功能丰富

智能的将数据放在需要的地方

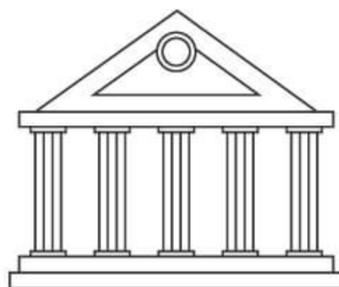
- 高可用复制集
- 分片弹性扩展
- 一库多用
- 可插拔引擎
- 就近数据读写

自由的运行于任意平台

- 服务器
- 云 (Atlas)
- 容器
- 移动端

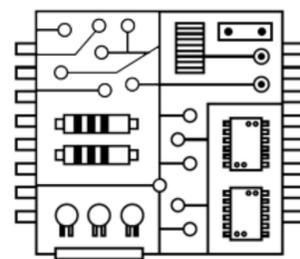
统一视图

Single View



金融

分析用户资产、信用、盈利、投资数据，控制风险，提供增值业务。



互联网

根据用户行为与浏览内容，实时推荐感兴趣的商品、内容。



零售

全渠道用户信息清洗合并，生成标签信息，个性化活动推送与服务。

统一视图

Single View

功能要求	MongoDB
数据模型 <ul style="list-style-type: none">- 数据来源于多个系统，数据格式差异大- 随着业务发展，需要动态调整数据模式	数据模型 <ul style="list-style-type: none">- 文档模型支持丰富的数据结构（Attribute Pattern）- 灵活模式无变更成本
查询 <ul style="list-style-type: none">- 查询条件、类型多变- 数据算法分析&快速可视化	查询 <ul style="list-style-type: none">- 丰富的查询、索引、聚合- 数据可视化：MongoDB Charts & BI Connector- AI: Spark connector、R & Python Drivers
性能与安全 <ul style="list-style-type: none">- 新数据源持续集成，数据量增长迅速- 敏感数据安全与保护	性能与安全 <ul style="list-style-type: none">- 自动扩展分片架构- 鉴权、权限、加密、审计- 基于Zone的分区

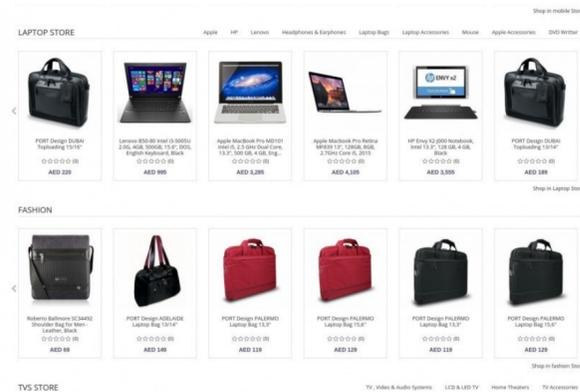
物联网 & 时序数据

IoT & Time-series Data

功能要求	MongoDB
数据模型 <ul style="list-style-type: none">- 时序类数据，大并发写入- 数据结构随着软/硬件版本的更新发生变化	数据模型 <ul style="list-style-type: none">- 文档模型支持丰富的数据结构 (Bucket Pattern)- 灵活模式无变更成本
扩展与性能 <ul style="list-style-type: none">- 数据量大，随设备全球分布- 持续在线- 读写实时性要求高	扩展性 <ul style="list-style-type: none">- 分布式架构auto-sharding- rang hash zone的分片策略
计算分析 <ul style="list-style-type: none">- 实时写入与实时分析- 大数据量计算分析	计算分析 <ul style="list-style-type: none">- 一库多用- 丰富的查询、索引、聚合- 数据可视化: MongoDB Charts & BI Connector- AI: Spark connector、R & Python Drivers

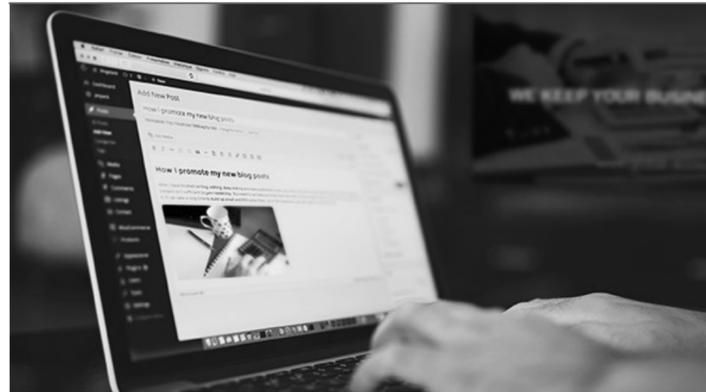
产品目录 & 内容管理

Product Catalog & Content Management



产品目录

SKUs、F/X标的，设备等。属性和metadata频繁更新。



内容管理

组织管理任何形式的数据：文本、视频、音频、图片等。



移动应用

移动/现代应用快速开发、快速上线、快速迭代。

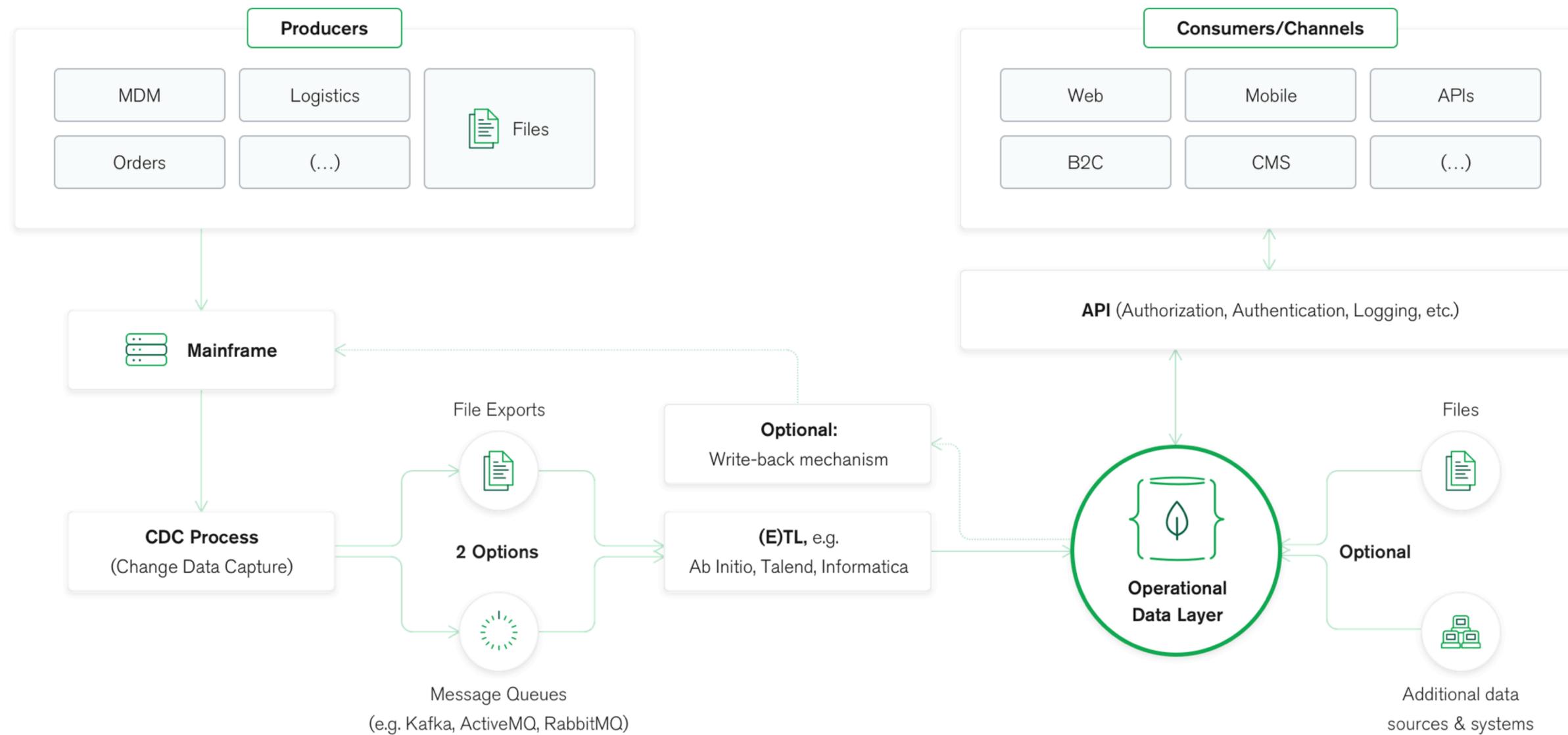
产品目录 & 内容管理

Product Catalog & Content Management

功能要求	MongoDB
数据模型 <ul style="list-style-type: none">- 对象有复杂多变的属性，元数据- 应用快速迭代，频繁更新数据模型	数据模型 <ul style="list-style-type: none">- 文档模型支持丰富的数据结构- 灵活模式无变更成本，应用版本快速迭代上线- GridFS支持存储二进制数据
用户体验 <ul style="list-style-type: none">- 用户查询请求条件组合多变- 促销、秒杀等活动期间数据库性能不能降低- 7*24在线	用户体验 <ul style="list-style-type: none">- 丰富的查询、索引、聚合- 分布式架构auto-sharding，用户就近访问数据- 一库多用：复制集同时满足在线交易与分析型需求
实时分析 <ul style="list-style-type: none">- 用户个性化推荐- 实时统计销售指标与内容消费情况	实时分析 <ul style="list-style-type: none">- 聚合Aggregation Pipeline- AI: Spark connector、R & Python Drivers- 数据可视化：MongoDB Charts & BI Connector

主机下移 & 数据中台

Mainframe Offload



正在使用

用户	案例情况
 <p>中國銀行 BANK OF CHINA</p>	<p>用户交易</p> <ul style="list-style-type: none">- 整合全行分散交易数据，实现客户层面一本帐- 收支实时汇总交易金额，秒级响应- 千亿级别数据实时处理
 <p>国泰君安证券 GUOTAI JUNAN SECURITIES</p>	<p>中台系统&策略量化</p> <ul style="list-style-type: none">- 用户全局数据、共享数据、标签、字典，处理三端一微30000/s的并发请求- 产品资讯信息，模拟大盘数据- 行情数据存放与分析，量化交易
 <p>中國東方航空 CHINA EASTERN</p>	<p>旅客数据</p> <ul style="list-style-type: none">- 旅客多数据源的快速集成（航信、行程、行李、投诉、贵宾厅、网站、App等）- Spark和MongoDB集成，大幅提升数据分析合处理速度- 旅客敏感数据集中到MongoDB中，成为唯一用户数据来源

正在使用

用户	案例情况
	<p>机票缓存</p> <ul style="list-style-type: none">- 机票价格缓存在MongoDB内存引擎中，快速响应票价请求- 利用TTL索引，自动删除过期票价- 分片架构应弹性对压力增长
	<p>网盘</p> <ul style="list-style-type: none">- 存放网盘文件的metadata元数据- 3PB数据量- 分片架构应弹性对压力增长
	<p>智能捷创平台</p> <ul style="list-style-type: none">- 保证普华永道全球所有用户能够协同工作- 支持跨区读写文档、附件；读写高性能，低延迟- 基于地理位置的分片集群- 低代码、低花费、快速开发

谢谢