重新思考数据库架构设计：

面向微服务和分布式应用程序开发

公益讲座11：00准时开始，请大家先浏览云技术微信公众号技术文章。资料会在各群同步发布，已入群客户请勿重复入群！

20-19

数据库和云讲座群

甲骨文云技术公众号

重新思考数据库架构设计：
面向微服务和分布式应用程序开发

许光

Dec 2022

Oracle SE Hub Database Management

Copyright © 2022, Oracle and/or its affiliates

现代应用开发

- 应用开发范式处于快速转型的状态
- 目标是，加速应用开发并为用户提供更丰富的体验

部署应用 新方法

服务构建块

容器

K8S

云

构建应用 新方法

微服务

事件

数据网格

接口驱动

低代码

新的数据模型和 数据 分析

机器学习

文档

图

文本搜索

空间处理

区块链

Copyright © 2022, Oracle and/or its affiliates

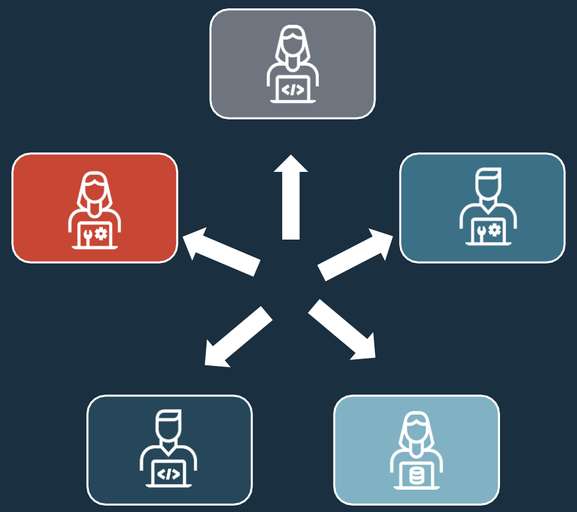
现代应用开发

- 简化应用程序的过程，可能会无意中使**数据库体系结构复杂化**
- 使用Oracle调整其数据库架构，以简化使用现代分布式应用开发范例构建的应用

Copyright © 2022, Oracle and/or its affiliates

应用和应用开发团队的变化

- 现代应用开发转型超越了技术变革
- 改变技术团队的组织方式
- 应用程序、开发团队和数据独立性的更广泛宏观趋势的一部分



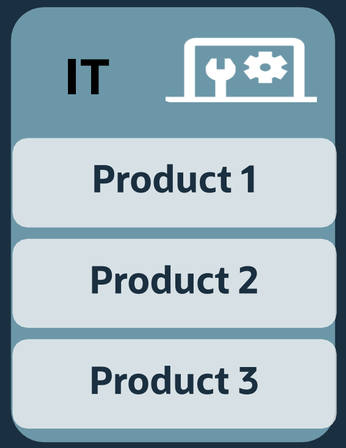
Copyright © 2022, Oracle and/or its affiliates

I.T集中式与分布式

- 集中型和分散型组织优缺点

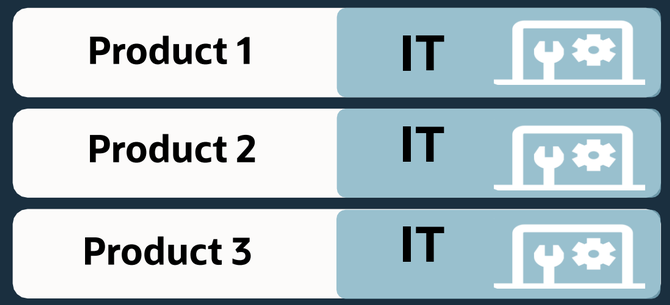
集中式 IT 优势

效率、一致性、降低风险



分布式 IT 优势

专注、结果导向、敏捷



IT组织传统上倾向于团队和技术的集中化

Copyright © 2022, Oracle and/or its affiliates

针对集中式的 Oracle 数据技术

- Oracle以其领先的数据技术而闻名，支持集中式架构/组织

整合和混合工作负载

Multitenant Database, Exadata, OLTP and analytics in one

容错可用性

RAC, Data Guard, Flashback, Online Data Reorg, Online Patching, MAA

高安全性

Data Safe, Advanced Security, Encryption, Roles, Privileges …

透明缩放

RAC, Exadata, Parallel SQL, Native Sharding

全面的数据模型支持

JSON, text, graph, spatial, blockchain

强大的分析

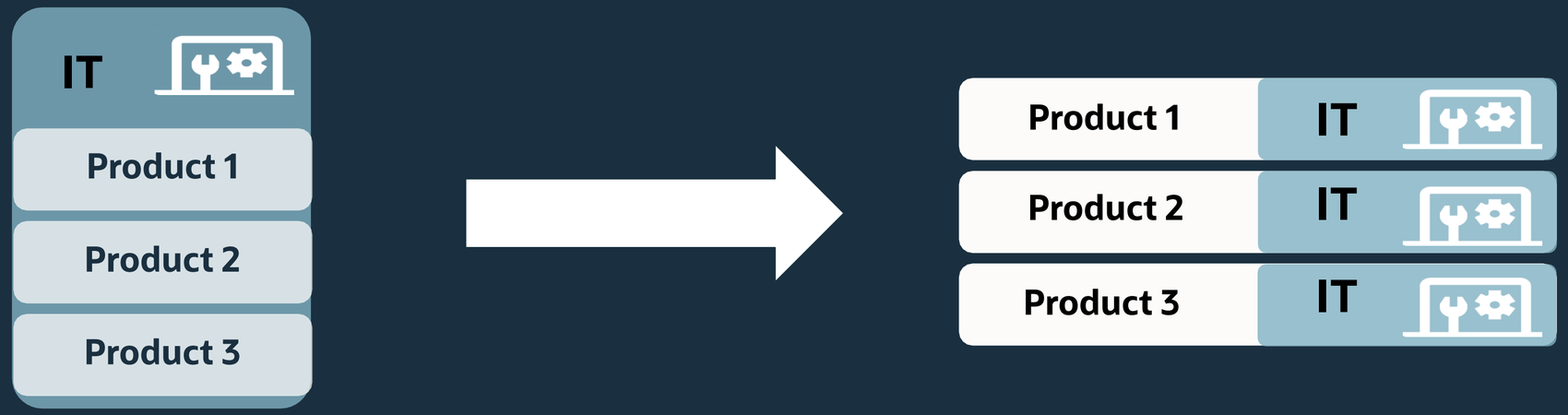
In-memory analytics, machine learning, data partitioning

这些技术通常部署在集中式架构上，但对于分布式也是关键

Copyright © 2022, Oracle and/or its affiliates

IT趋势正在走向分布式

- 最近，IT一直在转向更分散型组织模式
- 目标：更小、更敏捷、更专注的团队，拥有端到端的解决方案
- 现代应用开发技术使这种分布式方案成为可能



Copyright © 2022, Oracle and/or its affiliates

分散的技术组织 - DevOps

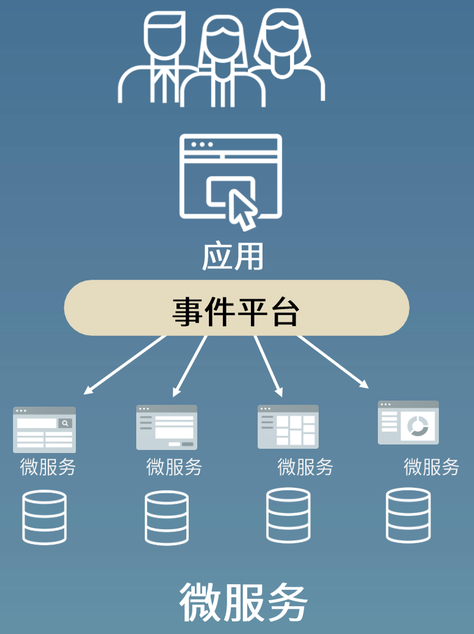
- 应用开发简化技术，使职能 IT 团队能够合并到 DevOps 中，并按产品进行拆分

Copyright © 2022, Oracle and/or its affiliates

分散型技术组织 - 微服务

- API 和事件驱动应用开发技术使团队能够进一步分散，专注于特定产品或服务，而不是完整的应用

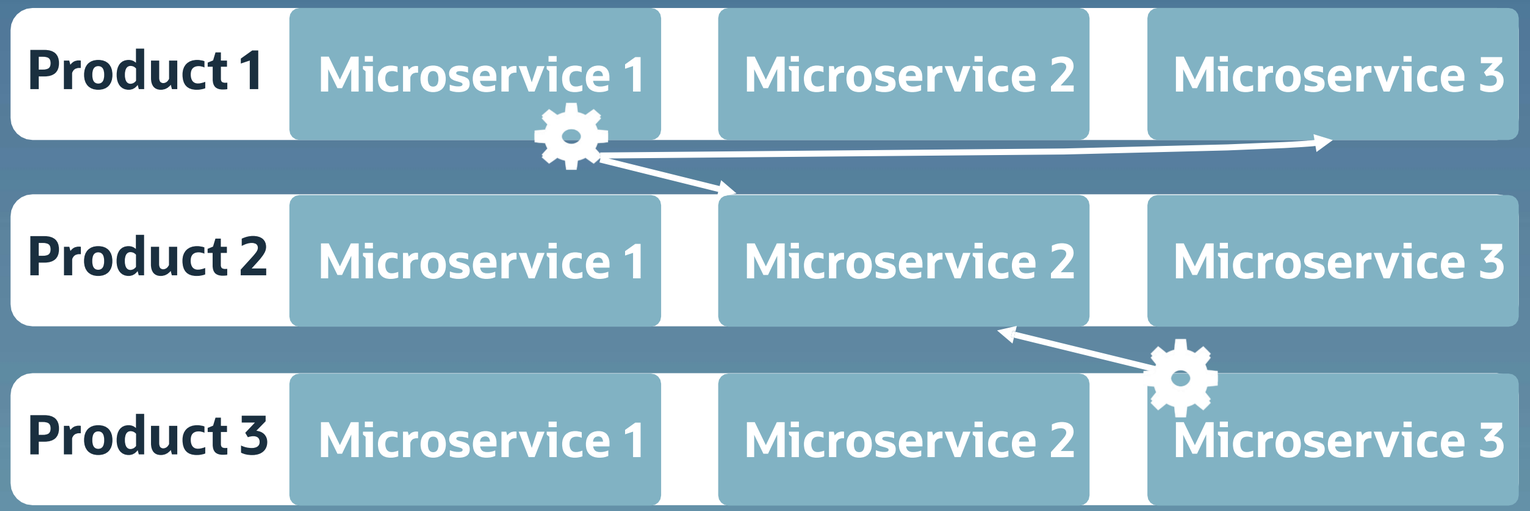
Product 1	Microservice 1	Microservice 2	Microservice 3
Product 2	Microservice 1	Microservice 2	Microservice 3
Product 3	Microservice 1	Microservice 2	Microservice 3



Copyright © 2022, Oracle and/or its affiliates

分散型技术组织 - 数据网格

- 数据网格架构也促进分散/分布
- 在数据网格中，领域数据“作为产品”发布给使用者
- 数据格式和发布归各个领域专家所有，而不是中心平台技术专家



Data Mesh

Copyright © 2022, Oracle and/or its affiliates

非集中式 - 数据

- 随着应用和应用开发团队变得更加分散，数据也变得更加分散
  - 会使数据库体系结构复杂化

Copyright © 2022, Oracle and/or its affiliates

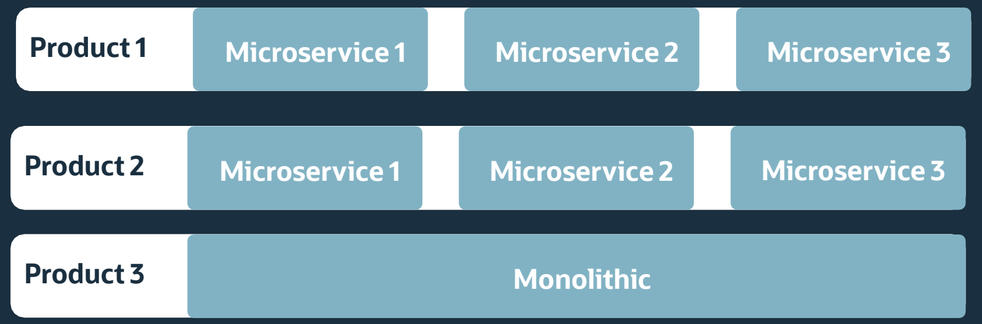
集中式和非集中式的更好选择

- **Oracle战略：为集中式和非集中式架构提供更好的数据技术**

提高集中式敏捷性

减少效率低下和分布式的风险

- 使IT领导者能够根据其业务需求选择替代方案或两者混用
  - 不基于底层数据技术的局限性

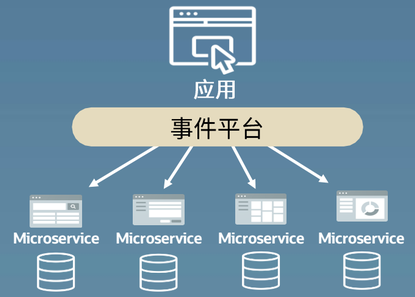


Copyright © 2022, Oracle and/or its affiliates

本次分享的重点 – 非集中式

- 我们将介绍我们的非集中式数据库架构，促进三个关键领域：
  - DevOps
  - 微服务
  - 数据网格
- 数据架构的挑战性

DevOps



微服务

数据网格

Copyright © 2022, Oracle and/or its affiliates

Part 1

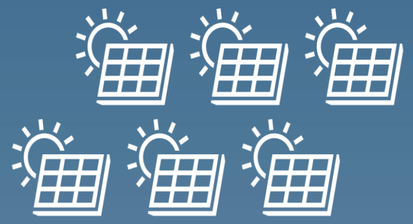
Oracle 如何简化分布式部署

Copyright © 2022, Oracle and/or its affiliates

分布式由简单性实现

分布式应用需要易于部署和操作的技术

发电厂的复杂性要求集中发电



太阳能电池板的简单性使发电分散

Copyright © 2022, Oracle and/or its affiliates

数据技术即服务

- 利用模块化，构建技术创建易于使用的API驱动服务实现分布式
- Oracle 的分布式数据库架构使分布式应用开发，从根本上简化部署和使用

数据技术即服务 (DataTech as Service)

数据库内部容器

Docker for Database

Kubernetes for Database

Database as Cloud Service

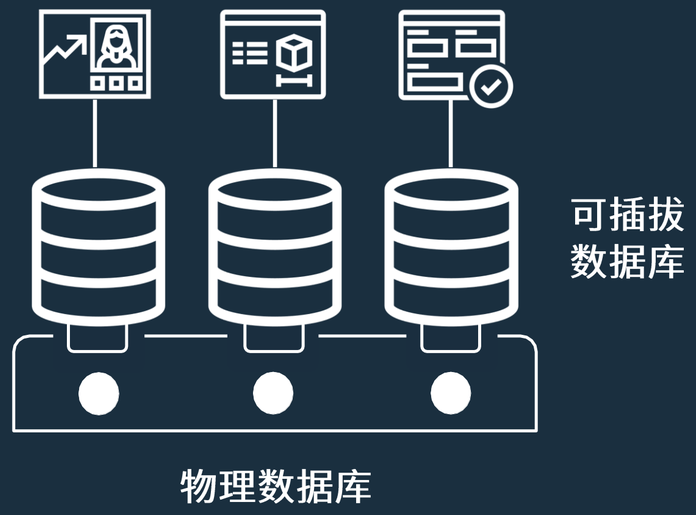
Autonomous Database

Copyright © 2022, Oracle and/or its affiliates

数据库内部容器

- 容器通过支持逻辑系统(而不是物理系统)的快速轻量级部署来简化部署

- Oracle 使用轻量级可插拔容器数据库实现容器的概念
  - 避免为每个微服务部署物理数据库
  - 许多可插拔数据库可以部署在单个物理数据库之上，以简化操作
  - 可插拔数据库可以在物理数据库之间轻松克隆和移动



Unlimited Pluggable Databases Now Available in Database 21c XE

Copyright © 2022, Oracle and/or its affiliates

Docker 容器

- 可轻松地使用 Docker 或 Podman 容器简化物理数据库的部署
- Oracle 为单实例、RAC 和分片部署提供预构建的容器映像
  - https://container-registry.oracle.com
- Oracle在GitHub上以发布用于创建这些映像的Docker文件
  - 用户可以使用 Docker 文件，为特定数据库版本和修补程序构建自定义映像

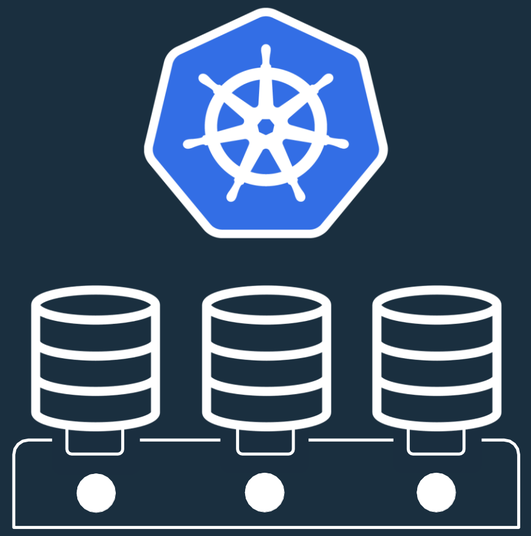
podman

More details in Wei Hu's session: Kubernetes and Microservices with the Oracle Database

Copyright © 2022, Oracle and/or its affiliates

Kubernetes

- Oracle 可用 Kubernetes 简化数据库部署编排
- Oracle Kubernetes Operators 编排可插拔容器数据库
  - 创建、删除、克隆、缩放、修补等
- Oracle 在 GitHub 上提供描述数据库结构的 Helm 图表
  - 支持分片数据库等复杂配置
- Kubernetes 工具和仪表板可以监控跟踪完整的应用程序



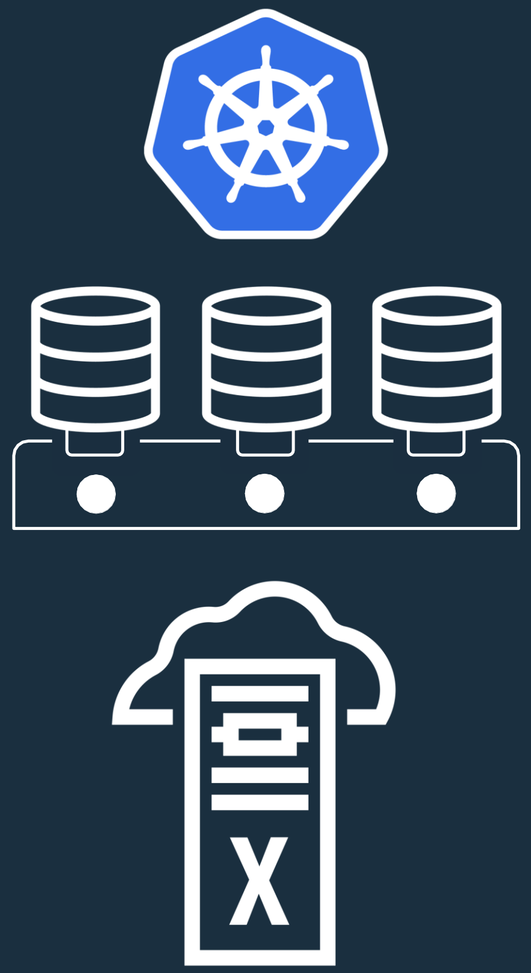
podman

More details in Wei Hu's session: Kubernetes and Microservices with the Oracle Database

Copyright © 2022, Oracle and/or its affiliates

Exadata 云服务

- 可以使用 Exadata 数据库服务，实现完整的数据库即服务（Database as a Service）模型
  - 运行在世界上更好的数据库平台 – Exadata
- 从硬件开始，自动化和优化所有数据库基础架构
- 提供用户控制的数据库自动化、弹性伸缩和按需计费
- 部署在公有云或使用Cloud@Customer部署在客户数据中心
  - 在多云配置中轻松使用，如Azure应用程序使用 Oracle 数据库服务



More details in my session: Oracle Exadata: Architecture and Internals Deep Dive

Copyright © 2022, Oracle and/or its affiliates

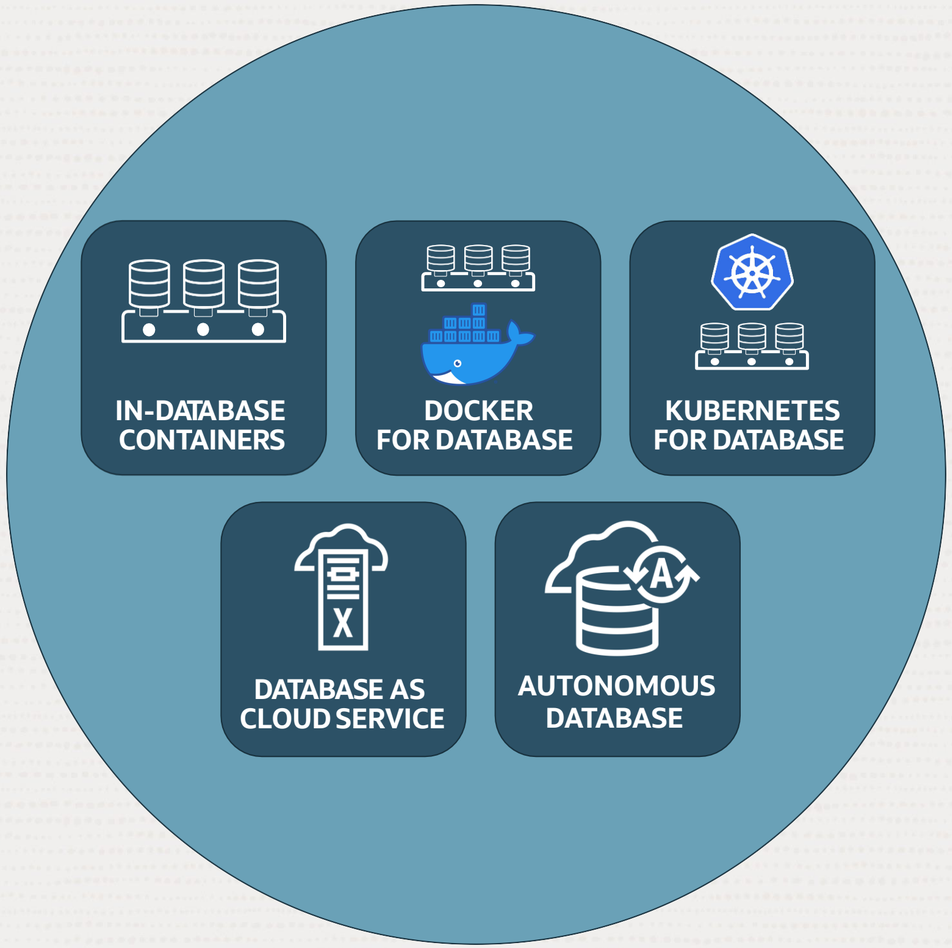
自治数据库

- 使用 Oracle 自治数据库提供的自动驾驶、自我保护和自我修复能力实现极致的简单性
- Oracle 接管所有通用数据库管理
  - 允许 DevOps 专注于应用的数据建模、安全性和性能
- 通过Cloud@Customer部署在公有云或数据中心
- 对于以 JSON 为中心的工作负载，请使用自治 JSON 数据库
  - 具有比MongoDB更好的JSON功能
  - 与MongoDB API兼容

More details in Yasin Baskan session: Oracle Autonomous Database: Simplify Your Database Management

Copyright © 2022, Oracle and/or its affiliates

重点



Oracle 分布式数据库架构使

微服务应用易于部署和操作

Copyright © 2022, Oracle and/or its affiliates

Part 2

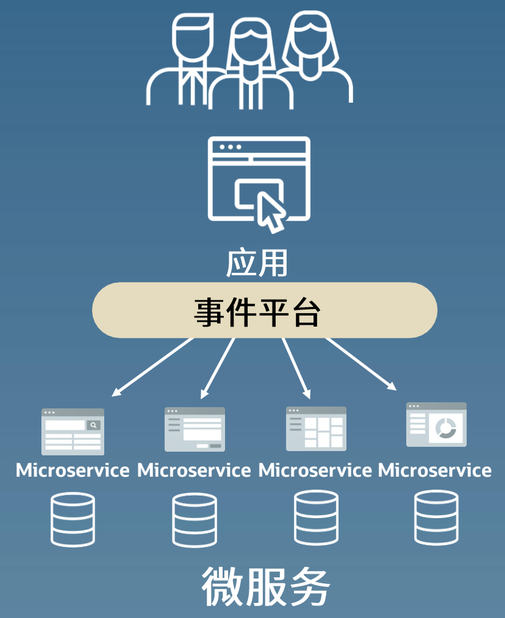
Oracle 如何使用异步事件简化分布式应用的架构

Copyright © 2022, Oracle and/or its affiliates

分布式应用程序如何实现

- 分布式应用通常使用异步事件，实现跨微服务操作
  - 例如，发布“处理此新订单”事件可能会触发多个微服务或操作

- 事件由Kafka 等事件平台发布



Copyright © 2022, Oracle and/or its affiliates

分布式应用程序问题

- 事件处理程序通常由开发人员手动编码 - 编码事件
  - 事件处理程序实现操作
  - 更改数据库状态
  - 产生其他事件

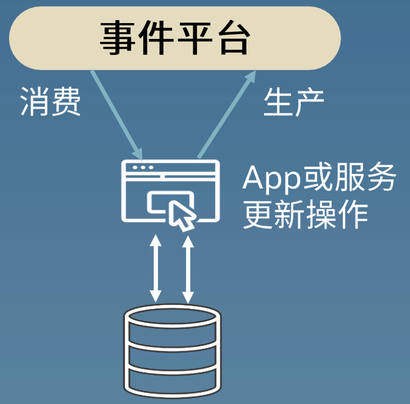
编码事件

- 事件处理在微服务之间保持数据一致性很困难，因为必须要处理以下问题
  - 进程和网络故障、并发更改、备份还原以及灾难故障转移
  - 随着时间的推移，任何数据不一致都会导致更多问题

Copyright © 2022, Oracle and/or its affiliates

事件系统到数据库的一致性

- 即使忽略跨服务一致性，维护有状态事件平台和单个有状态数据库之间的一致性也很困难
- 事件平台要求开发人员将一致性、故障恢复手动编码到每个事件处理程序中
- 对于简单的发布或使用情况，开发人员必须手动跟踪事件使用的状态并将其存储在数据库中
- 对于涉及多个事件的更复杂情况，保持一致性需要编写大量容易出错的代码

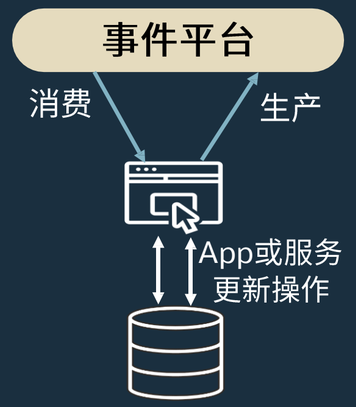


Copyright © 2022, Oracle and/or its affiliates

ACID 和事务的好处

- 原子事务和ACID的好处很好理解，但经常被遗忘

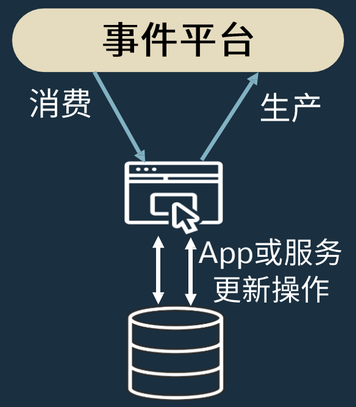
- NoSQL数据库最初声称事务是不必要的，且有害的
  - 然而，NoSQL数据库的用户通过经验了解到，应用程序中的编码数据一致性容易出错且耗时



Copyright © 2022, Oracle and/or its affiliates

事务和Kafka

- 较新版本的 Kafka 支持事务 - 但只支持事件平台内部的事务
- 对于不支持事务的异构服务之间的事件，Kafka是一个很好的解决方案
- 对于 Oracle 数据库内或跨 Oracle 数据库的事件，使用数据库原生事务事件作为事件平台更可靠，也更简单



Copyright © 2022, Oracle and/or its affiliates

Oracle数据库支持事务性事件

- Oracle 事务事件队列 (TxEventQ) 是内置于 Oracle 数据库中的免费事件平台，20 多年来一直在不断的增强
  - 早期更有限的版本称为AQ
- 如今，TxEventQ 已被数以万计的企业使用

Copyright © 2022, Oracle and/or its affiliates

Oracle数据库支持事务性事件

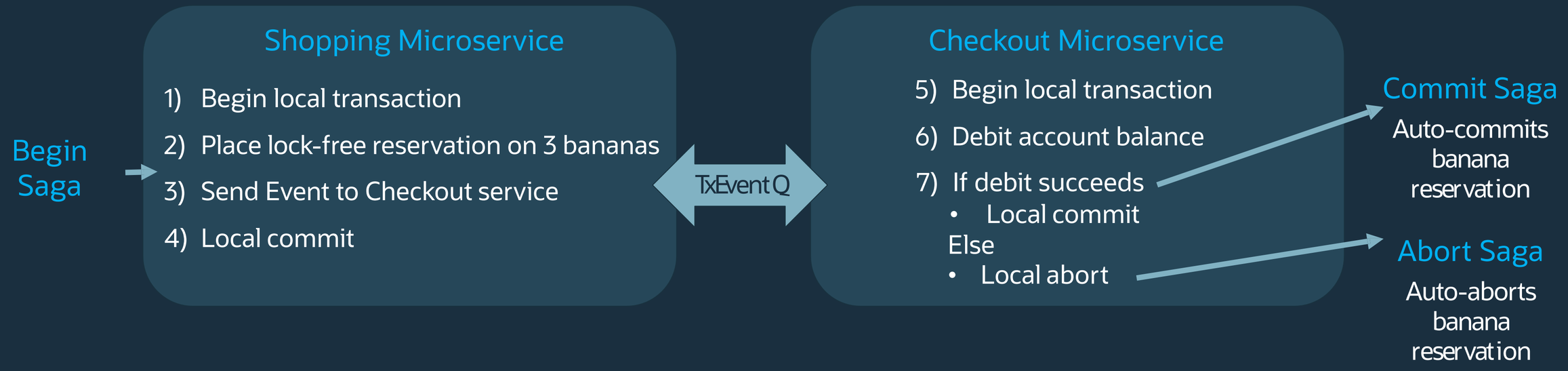
- TxEventQ 使分布式应用程序的开发更简单
  - 事件的发布/消费与数据更改以事务方式的集成
  - 处理复杂情况，例如更改数据库状态并生成更多事件的事件
  - 提供仅一次的语义 - 没有两阶段提交和幂等性难题
  - 事件和数据的安全性、备份、恢复等是联系在一起的
- 横向扩展架构，每秒可处理超过一百万个事件
- TxEventQ 无需两阶段提交，即可跨数据库实现事件的可靠传播
- 融合连接器可以在 Kafka 和 TxEventQ 之间传播异构事件

More details in session: How Oracle Database Transactional Events Interoperates with Apache Kafka

Copyright © 2022, Oracle and/or its affiliates

长时间运行的分布式事务

- 处理跨微服务工作流尤其复杂，如果之后的更改失败，则必须回退较早提交的更改
- 带有 Oracle Database 23c 无锁预订和 Sagas 的 TxEventQ，将自动处理其中许多困难场景
- 如在网上商城



More details in session: Event Driven Applications with a Global Event Mesh

Copyright © 2022, Oracle and/or its affiliates

数据变更事件

- 事件分为两类：由应用程序编码的事件和数据更改事件

编码事件

- 数据事件是状态变化，例如价格更改：
  - 通知相关微服务重要数据的更改
  - 使其相关微服务保持同步
  - 允许创建数据网格事件，供 OLTP 或分析系统使用

数据事件

Copyright © 2022, Oracle and/or its affiliates

使用 Kafka 传播跨服务状态更改

- 应用程序可以使用 Kafka 发布和消费数据事件
  - 但需要生产者以手动编码的方式捕获状态变更
  - 由消费者手动编码方式设置状态变化
- 复杂的手动编码难以保持数据一致性
  - 繁琐、脆弱的且难以并行化
- 无法直接使用 SQL 快速轻松地实现临时更改和批处理
  - 必须对所有状态更改进行编码，以便由代码事件捕获和传播
  - 敏捷性和生产力的巨大损失

编码事件

数据事件

Copyright © 2022, Oracle and/or its affiliates

原生数据事件

- Oracle GoldenGate 无需手动捕获和应用数据事件，而是自动捕获
  - 从所有主流的数据库、应用和云中捕获数据事件
  - （可选）通过流处理转换、丰富和分析数据事件
  - 将丰富的数据事件发布到 Kafka 等事件平台
  - 或者，直接将数据事件应用于数百种目标类型
- 整个过程是自动化、并行和事务性的，以确保高性能和完整的数据一致性
- GoldenGate是开发数据网格的理想底层基础架构

GoldenGate is the Market and Technology Leader for Data Events and Replication
Trusted By 1000’s Of Enterprises, Including 80% Of The Global Fortune 100

Copyright © 2022, Oracle and/or its affiliates

重点 - 异步事件

- 区分2种类型的事件 - 编码事件和数据事件
- 对于编码事件，使用 Oracle 数据库内部和数据库之间的事务性事件队列，将受益于事务一致性 (ACID)
- Kafka 适合用于跨异构数据库和微服务的应用逻辑
- 当构建Oracle 数据库和异构系统时，可用 Kafka 连接器在 Kafka 和 TxEventQ 之间传输事件
- 对于数据事件，可以使用GoldenGate实时的发布和消费数据事件
  - 微服务或应用之间
  - 公共云之间（启用多云）
  - 操作系统和分析系统之间
  - 数据网格

编码事件

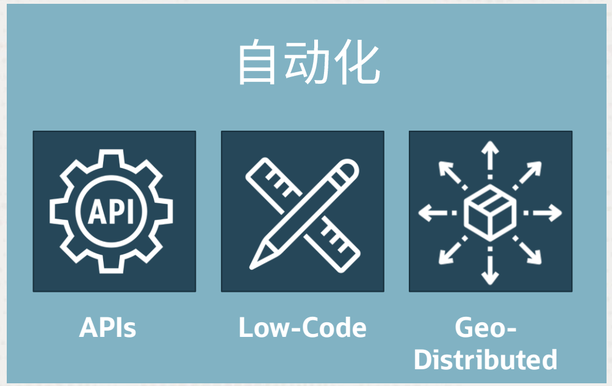
数据事件

数据网格

Copyright © 2022, Oracle and/or its affiliates

Part 3

Oracle 如何简化分布式应用的创建



Copyright © 2022, Oracle and/or its affiliates

加速分布式应用开发

- 通常有几种技术用于加速分布式应用程序的开发

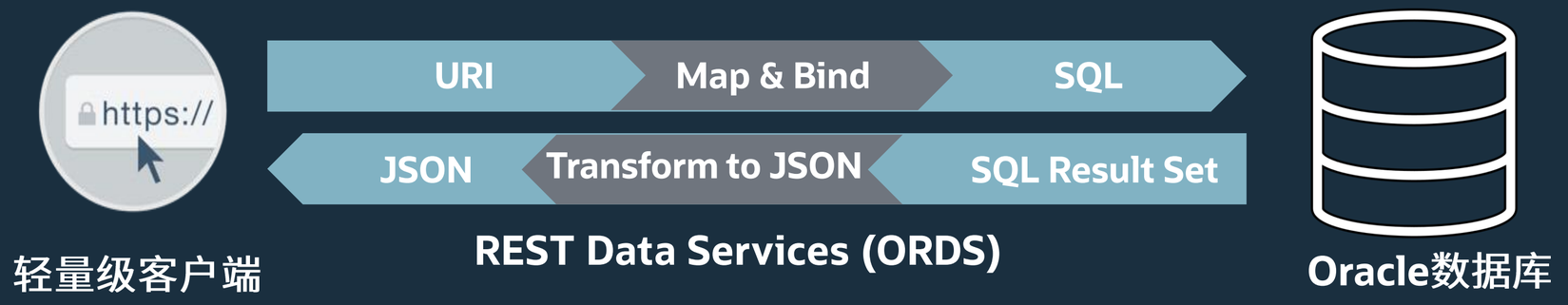
用于通信的 REST API

低代码开发

分片部署，实现可扩展性和数据主权

Copyright © 2022, Oracle and/or its affiliates

API 驱动架构 – 数据库原生 REST



- Oracle Rest Data Services （ORDS） 可在任何 SQL 语句或存储过程之上自动创建 REST API
  - 免费支持Open API 、OAuth、分页、过滤、LOB、连接池等功能
  - 将应用 API 与数据 API 统一起来，使分布式应用更易于开发
  - 实现数据即服务

Copyright © 2022, Oracle and/or its affiliates

使用低代码 APEX 加速应用开发

- Oracle 的免费 APEX 低代码 IDE， 显著简化并加速了分布式应用开发
  - 借助 APEX，可以自动生成企业应用，而不是以编码的方式开发
- APEX 统一应用层和数据层
  - 通过消除编码和管理中间层来提高敏捷性
  - 不再需要应用层状态或连接管理，也不再需要将应用类型映射到数据库类型
- APEX 应用程序使用 REST 数据服务，与其他分布式应用程序进行通信
- 低代码的简单性使精通技术的领域专家能够创建自己的应用程序，从而进一步分散应用程序开发

Copyright © 2022, Oracle and/or its affiliates

地理分布式数据 – 数据库原生分片

- Oracle 原生分片可自动实现数据的分散化
  - 地理数据分布或大规模
- 将集中式数据库分解为由较小的分布式数据库组成的集群
- 分片可以位于不同的国家，以满足数据主权
- 分片是完全隔离的 - 提供数据和用户的线性可扩展性
  - 基于分片键路由 SQL 或跨分片运行 SQL
  - 提供分片的在线添加和重组
- 保有分片的所有优势以及成熟 SQL 数据库的所有优势

Used Today by Many of the Largest Internet Providers and Telecoms in the World

Copyright © 2022, Oracle and/or its affiliates

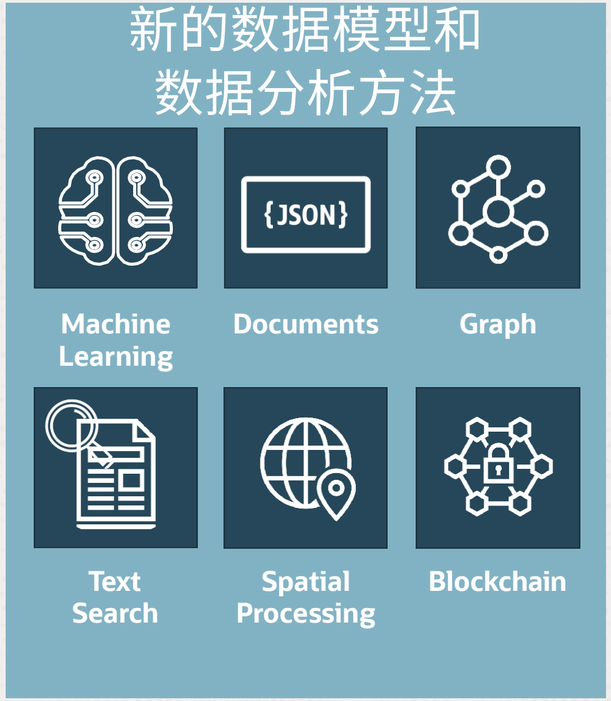
重点

甲骨文的分布式数据库架构

帮助加速分布式应用程序的创建

Copyright © 2022, Oracle and/or its affiliates

Part 4
Oracle 如何在分布式应用中轻松使用现代数据模型和负载



Copyright © 2022, Oracle and/or its affiliates

分布式应用支持现代数据模型和工作负载

- 分布式应用程序使用新一代数据技术，为用户提供更具吸引力的体验，并为企业提供更多价值

新的数据类型

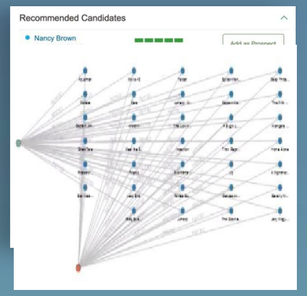
Relational | {JSON} Document | Spatial | Text

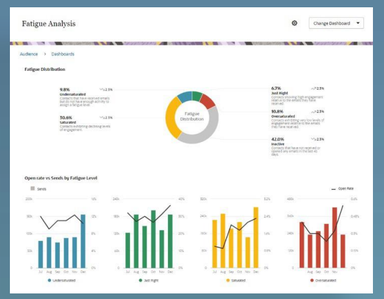
新的分析类型

Data Warehouse | Machine Learning | Graph | Lake House

新的负载类型

Micro Services | Geo-Distributed | DynamoDB IoT | Blockchain





Copyright © 2022, Oracle and/or its affiliates

分布式应用支持现代数据模型和工作负载

- 构建分布式应用程序的一种方法是，为每个应用需求使用专用数据库

新的数据类型	新的分析类型	新的负载类型
PostgreSQL Relational	Snowflake Warehouse	kafka Micro Services
MongoDB Documents	Spark MACHINE LEARNING Machine Learning	cassandra Geo-Distributed
PostGIS Spatial	Neo4j Graph	DynamoDB IoT
Elastic Search	Elastic Search	AWS QLDB Blockchain

- 每个专用数据库都专注一种特定应用需求

Copyright © 2022, Oracle and/or its affiliates

分布式应用支持现代数据模型和工作负载

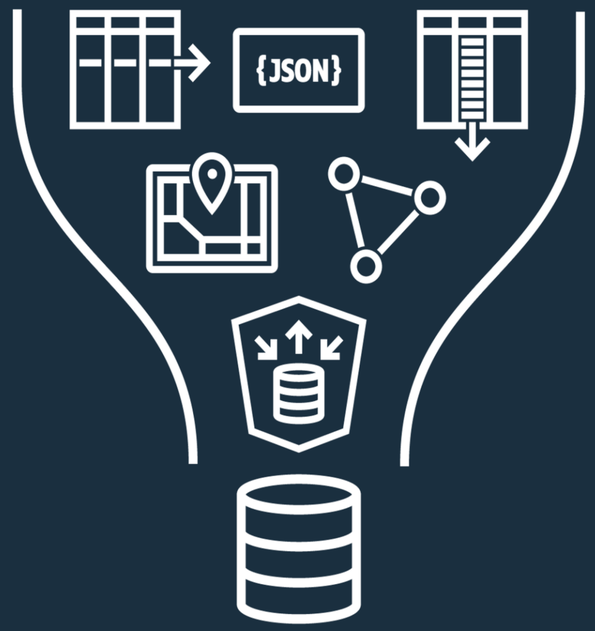
- 本质上，会创建异构的应用程序体系结构

新的数据类型	新的分析类型	新的负载类型
PostgreSQL Relational	Snowflake Warehouse	kafka Micro Services
MongoDB Documents	Spark MACHINE LEARNING Machine Learning	cassandra Geo-Distributed
PostGIS Spatial	Neo4j Graph	DynamoDB IoT
Elastic Search	Elastic Search	AWS QLDB Blockchain

Copyright © 2022, Oracle and/or its affiliates

使用现代数据模型和工作负载

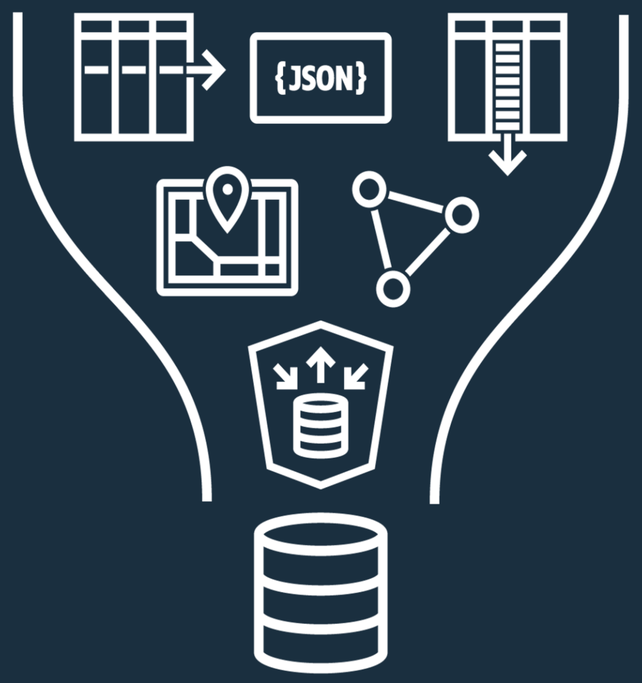
- Oracle 是一个融合数据库，可为所有数据模型和负载提供更好的支持



- 关系数据
- JSON 和 XML 文档
- 机器学习
- 图分析
- 文本搜索
- 空间操作
- 区块链

Copyright © 2022, Oracle and/or its affiliates

使用现代数据模型和工作负载

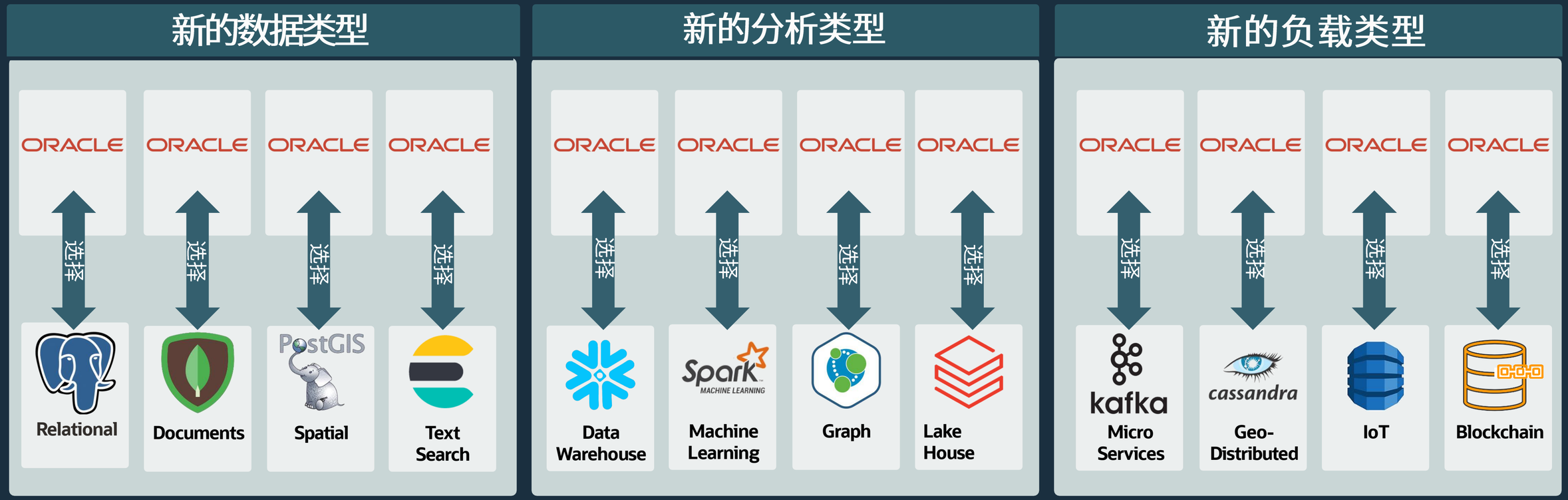


- 所有数据类型和工作负载都是内置的
  - 只需调用 SQL 即可使用
- 相同的 SQL、REST API 和事务跨所有数据类型和工作负载运行
- 显著简化应用开发和运维，使小型分散式团队能够使用现代数据类型和工作负载

Copyright © 2022, Oracle and/or its affiliates

Oracle 融合数据库为客户提供选择

- 融合并不意味着数据必须在一个整体数据库中
- 融合数据库可为每个应用程序提供数据架构



Copyright © 2022, Oracle and/or its affiliates

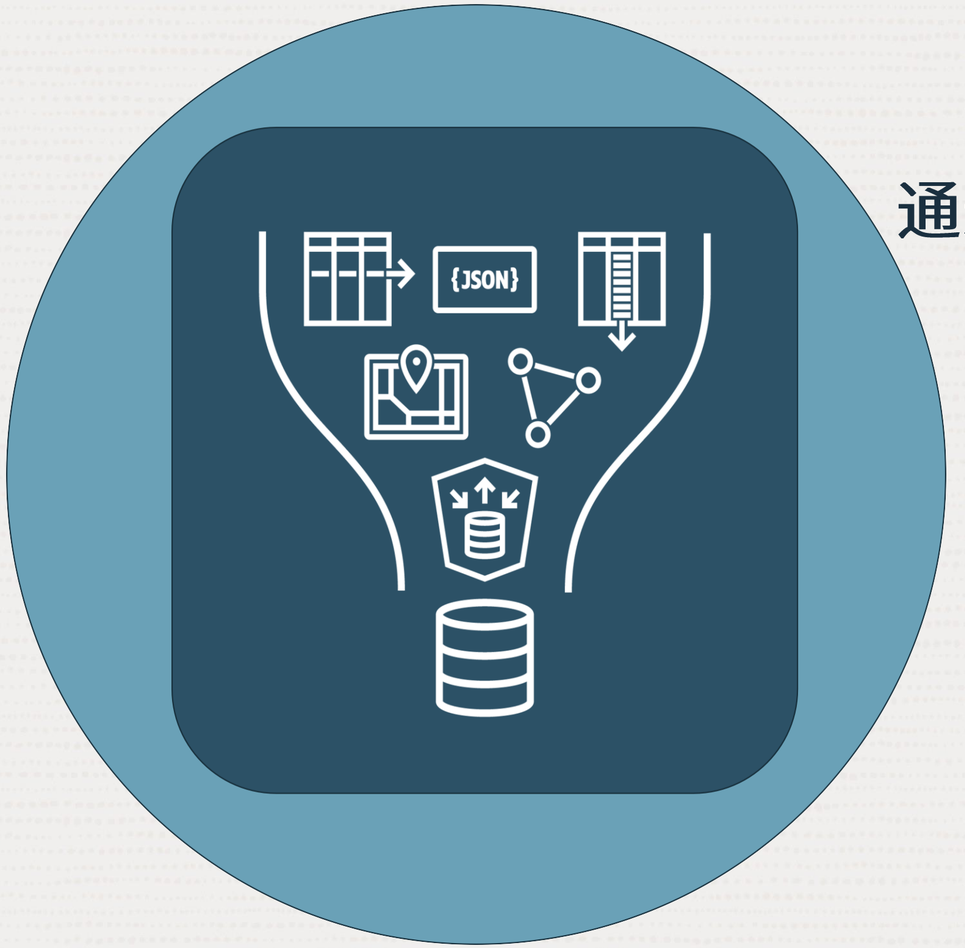
Oracle 融合数据库为客户提供选择

- 可以选择何时更好地将数据合并到更少的数据库中
- 不会因为数据库的限制，而被迫进行完全异构或分布式设计

新的数据类型	新的分析类型	新的负载类型
ORACLE: Relational, Documents, Spatial	ORACLE: Data Warehouse, Machine Learning, Graph, Lake House	kafka: Micro Services
Elastic Search		ORACLE: Geo-Distributed
		ORACLE: IoT, Blockchain

Copyright © 2022, Oracle and/or its affiliates

重点



通过消除单一用途数据存储带来的复杂性

Oracle 融合数据库显著简化

分布式应用开发

Copyright © 2022, Oracle and/or its affiliates

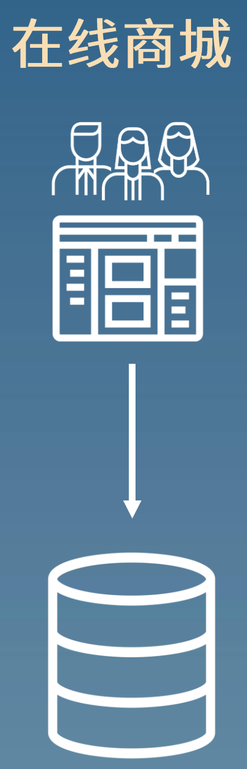
Part 5

分布式应用程序的简化示例

Copyright © 2022, Oracle and/or its affiliates

在线商城示例

- 传统上，应用程序的所有的方面都将集中在一起
- 单一应用
- 单一数据库



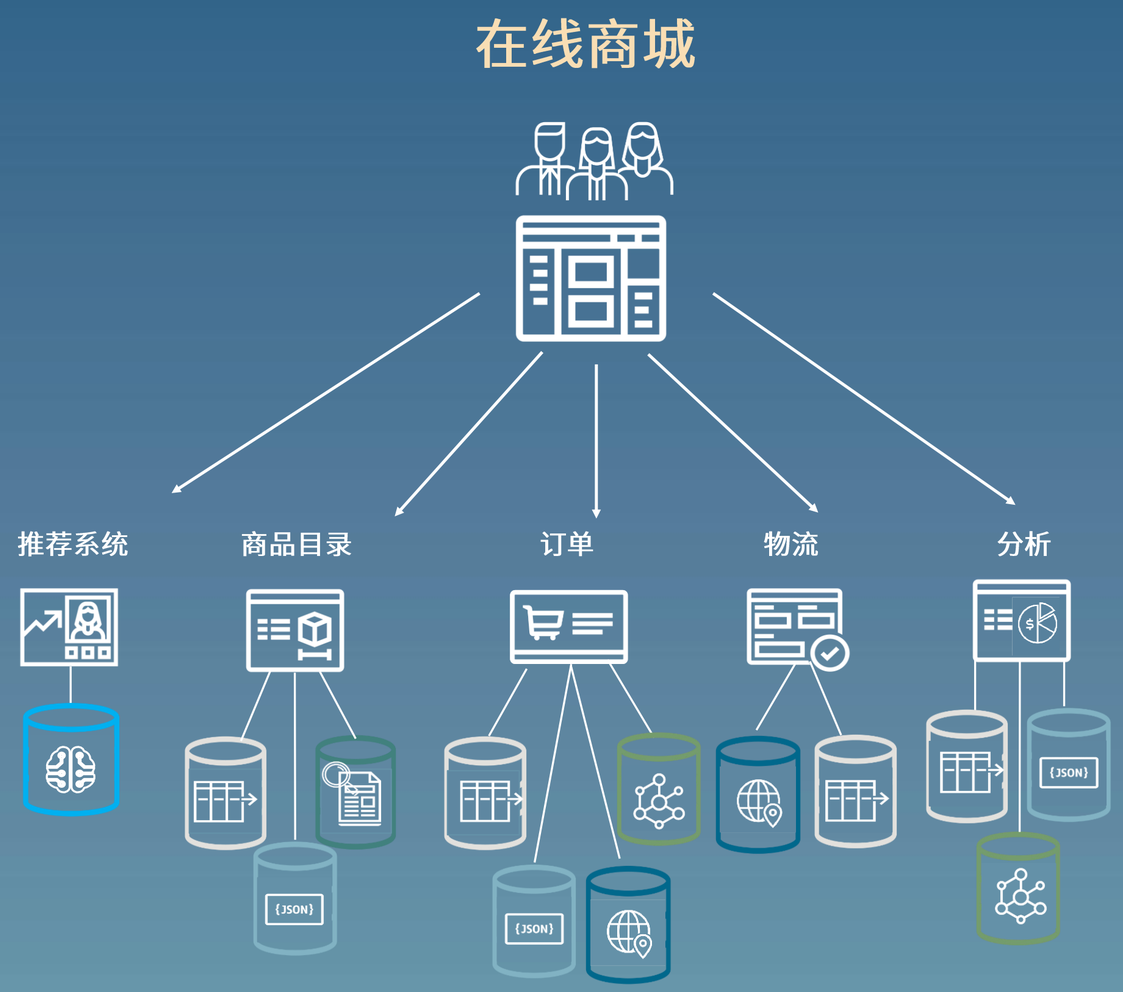
Copyright © 2022, Oracle and/or its affiliates

在线商城 – 分布式微服务架构

- 在线商城的微服务体系结构看起来非常不同
- 应用架构和数据库架构为分布式

- 应用分解为 5 个微服务
  - 商品推荐、商品目录、订单、物流、分析

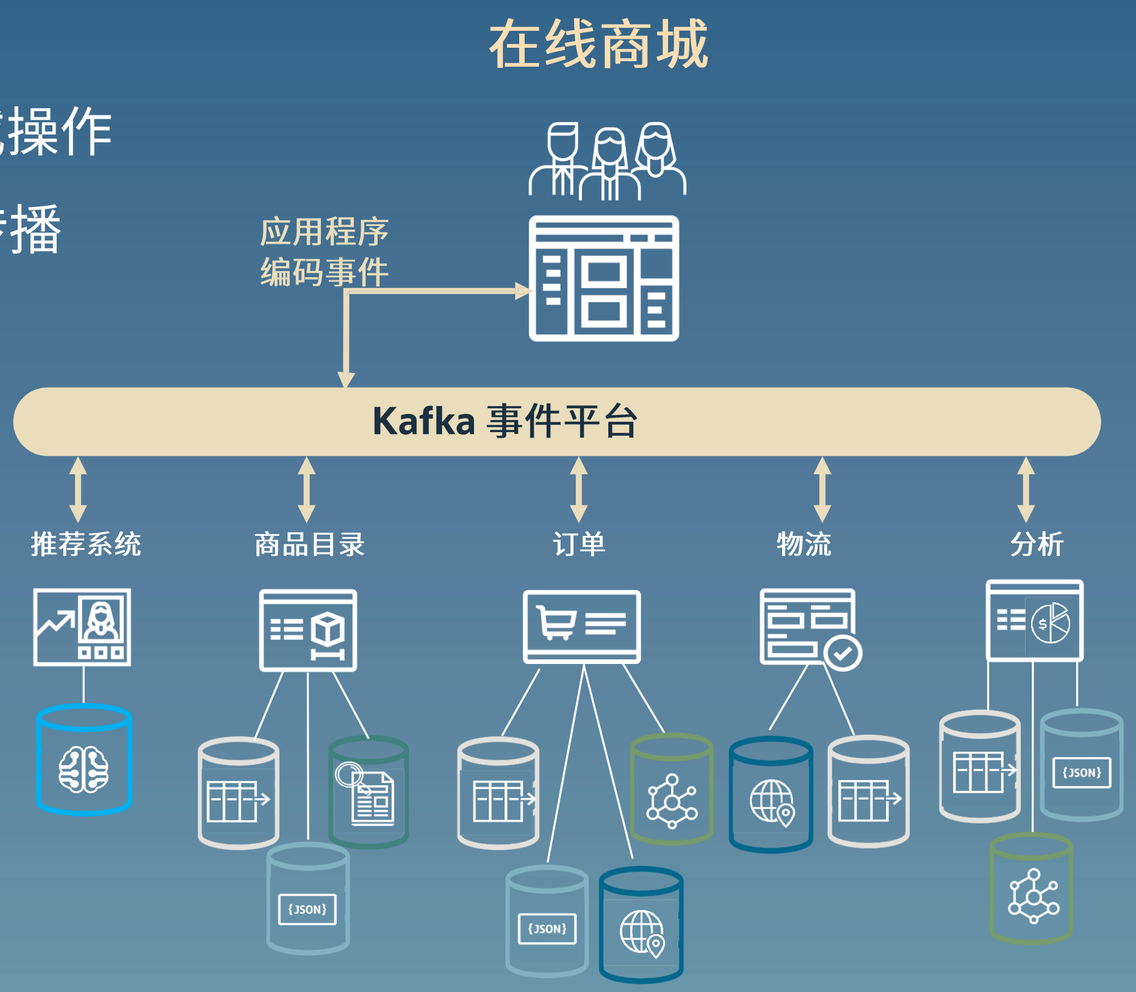
- 每个微服务使用多个单一用途数据库
  - 每个对应一个数据模型或一个工作负载类型



Copyright © 2022, Oracle and/or its affiliates

应用操作编码事件

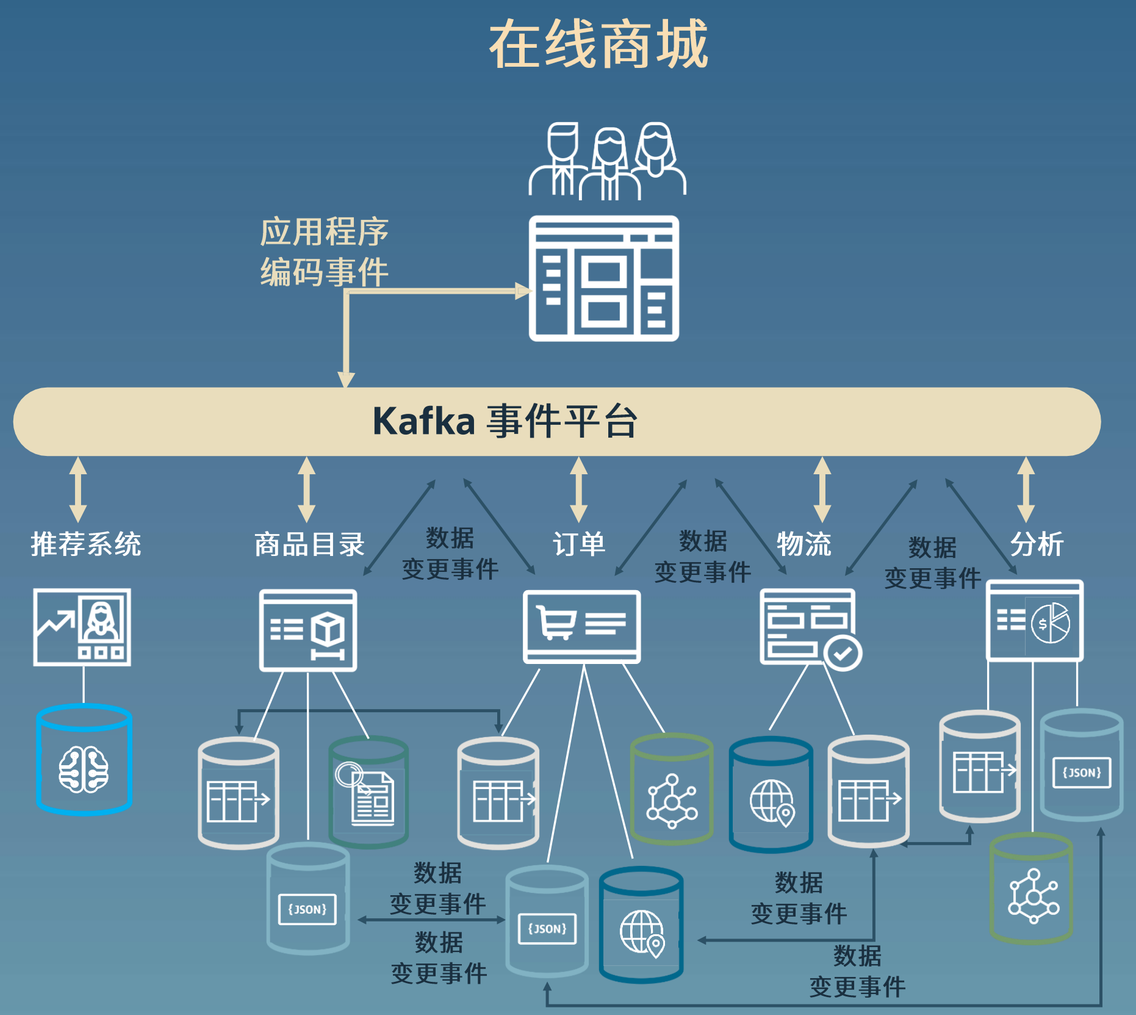
- 异步编码事件用于实现跨微服务的应用逻辑或操作
- 编码事件使用 Kafka 等事件平台在应用级别传播



Copyright © 2022, Oracle and/or its affiliates

数据更改编码事件

- 编码事件还被用于微服务之间传播数据变更



Copyright © 2022, Oracle and/or its affiliates

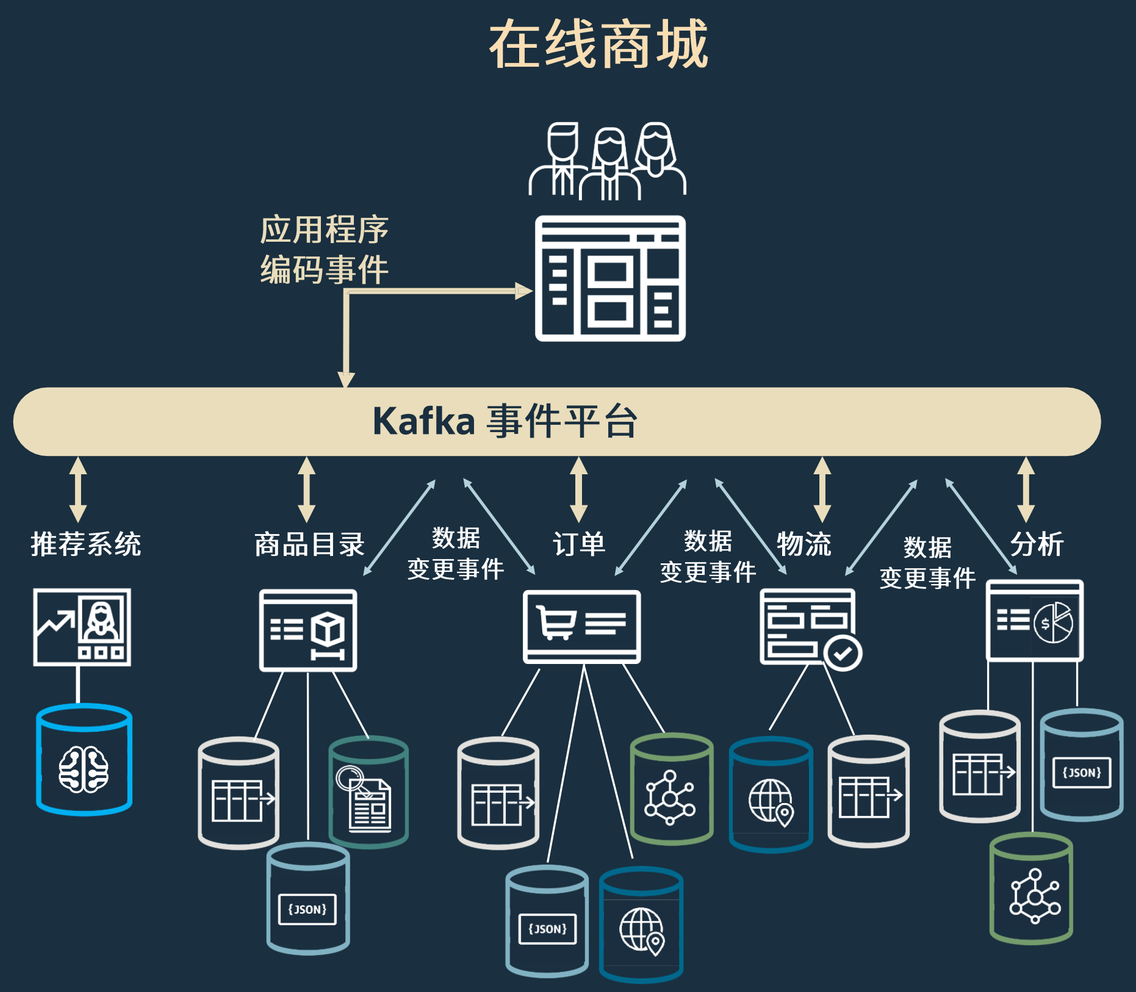
Oracle 如何简化微服务架构?

Copyright © 2022, Oracle and/or its affiliates

在微服务中融合多个数据库

步骤 1

- 将微服务中的多个单一用途数据库替换为单个融合数据库

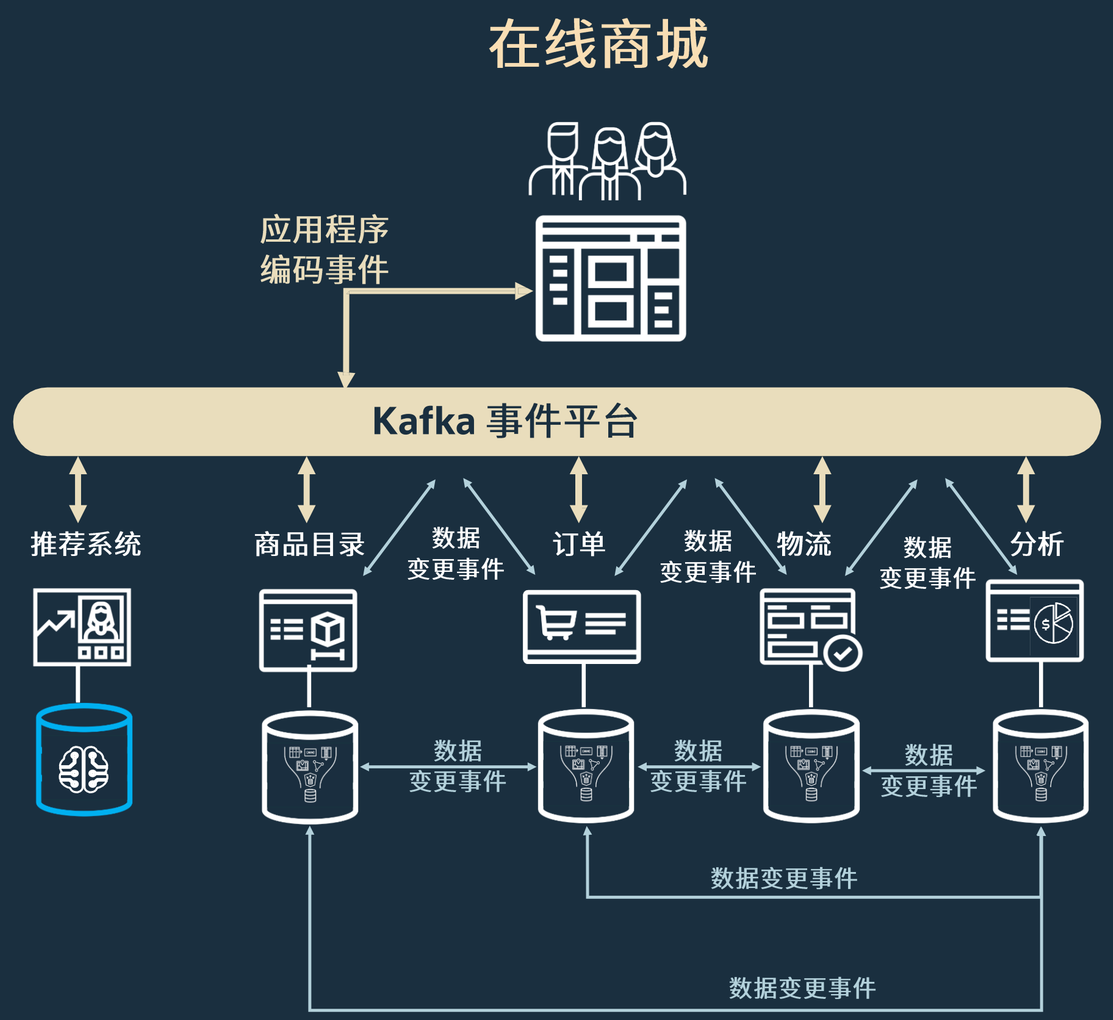


Copyright © 2022, Oracle and/or its affiliates

在微服务中融合多个数据库

步骤 1

- 将微服务中的多个单一用途数据库替换为单个融合数据库
- 对微服务进行调优、保护、扩展和故障排除要容易得多
  - 如果使用同一种数据存储方式

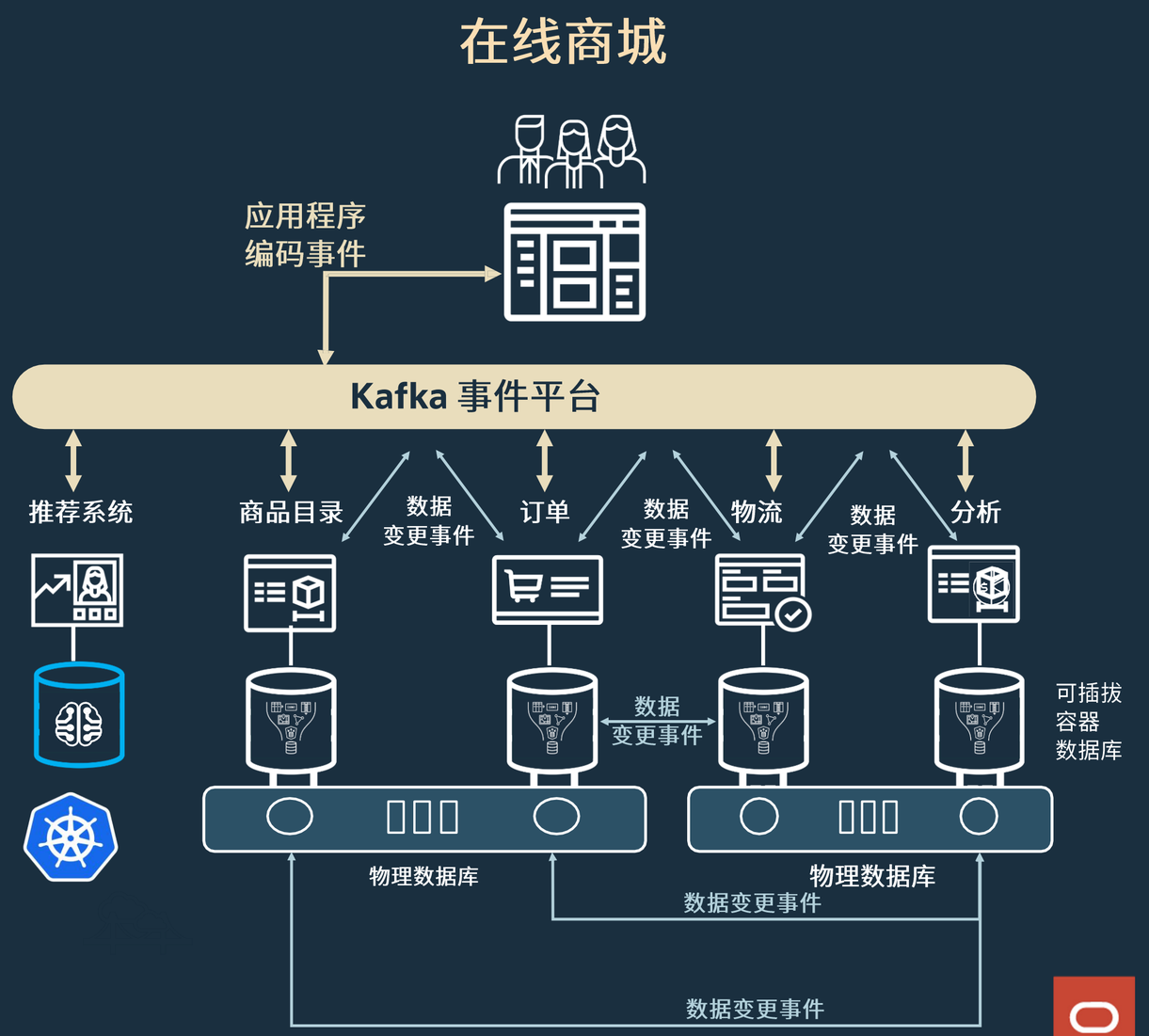


Copyright © 2022, Oracle and/or its affiliates

容器化微服务数据库

步骤 2

- 使用可插拔数据库，将融合数据库容器化为更少的物理数据库
  - 将多个数据库作为一个整体管理
- 使用 Kubernetes 操作编排可插拔数据库



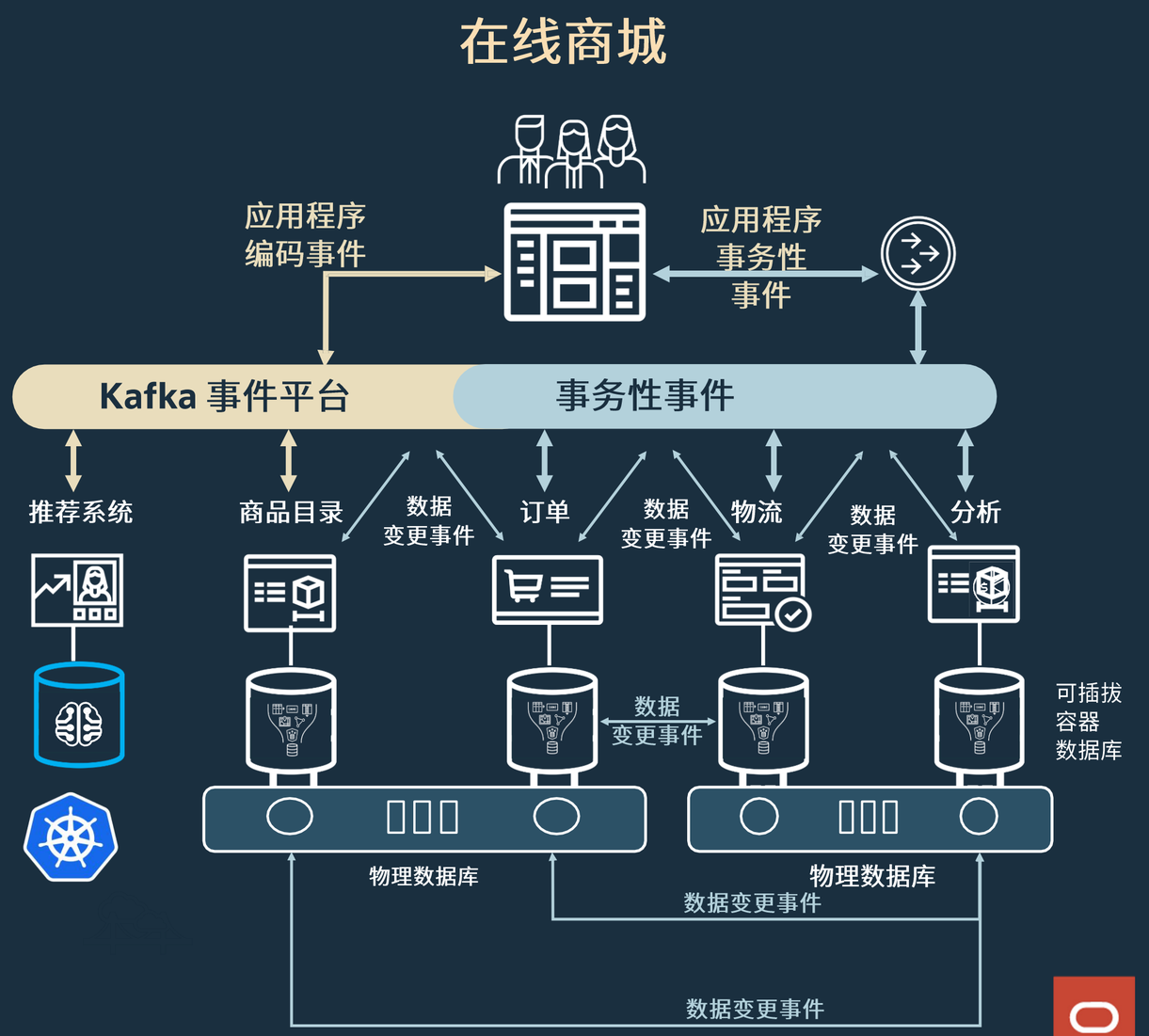
Copyright © 2022, Oracle and/or its affiliates

使用事务性事件队列

步骤 3

- 尽可能使用事务性事件队列实现编码事件
  - 事件和数据库更改在一个原子事务中执行

- 继续将 Kafka 用于异构事件

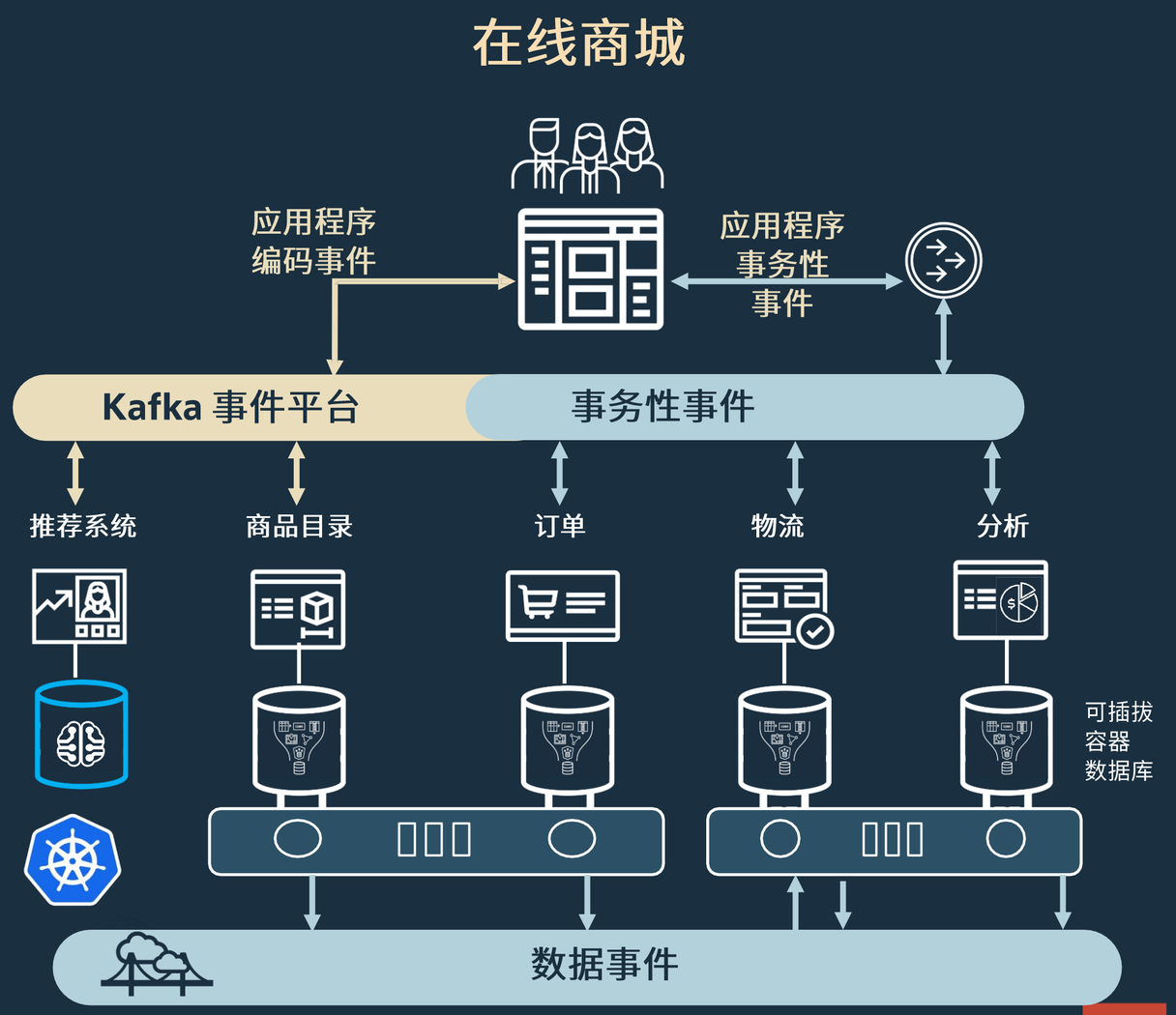


Copyright © 2022, Oracle and/or its affiliates

使用 GoldenGate

步骤 4

- 使用 Oracle GoldenGate 在微服务、应用或云之间以事务方式发布和使用数据事件
  - 无需手动编码捕获和应用状态更改
- 应用可以同时具有编码事件和数据事件
- GoldenGate 可以向Kafka发布数据事件

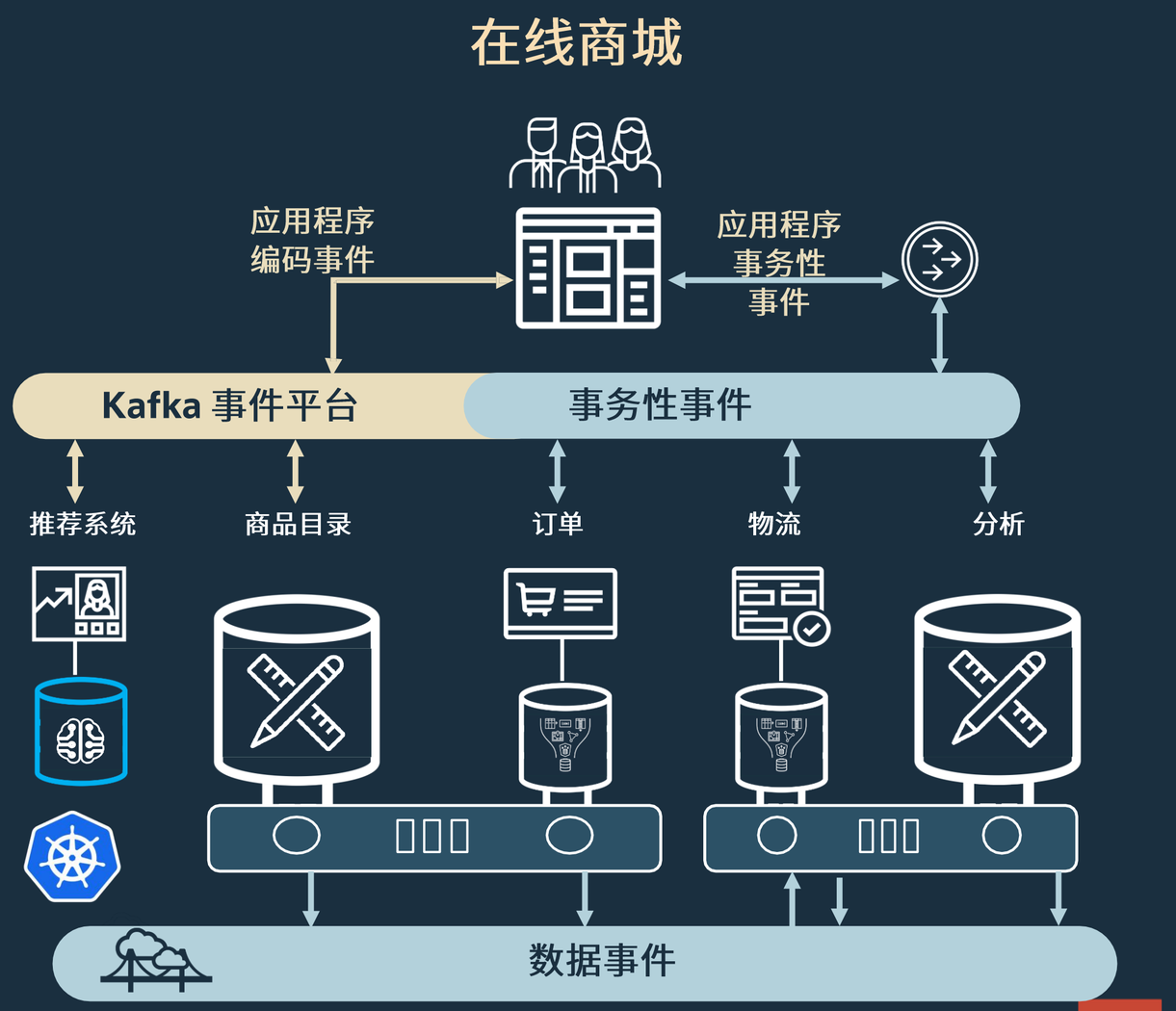


Copyright © 2022, Oracle and/or its affiliates

使用低代码 APEX 生成应用

步骤 5

- 使用低代码 APEX 生成许多企业应用或微服务，而不是手动编码
  - 还消除了单独的应用层和数据层的复杂性

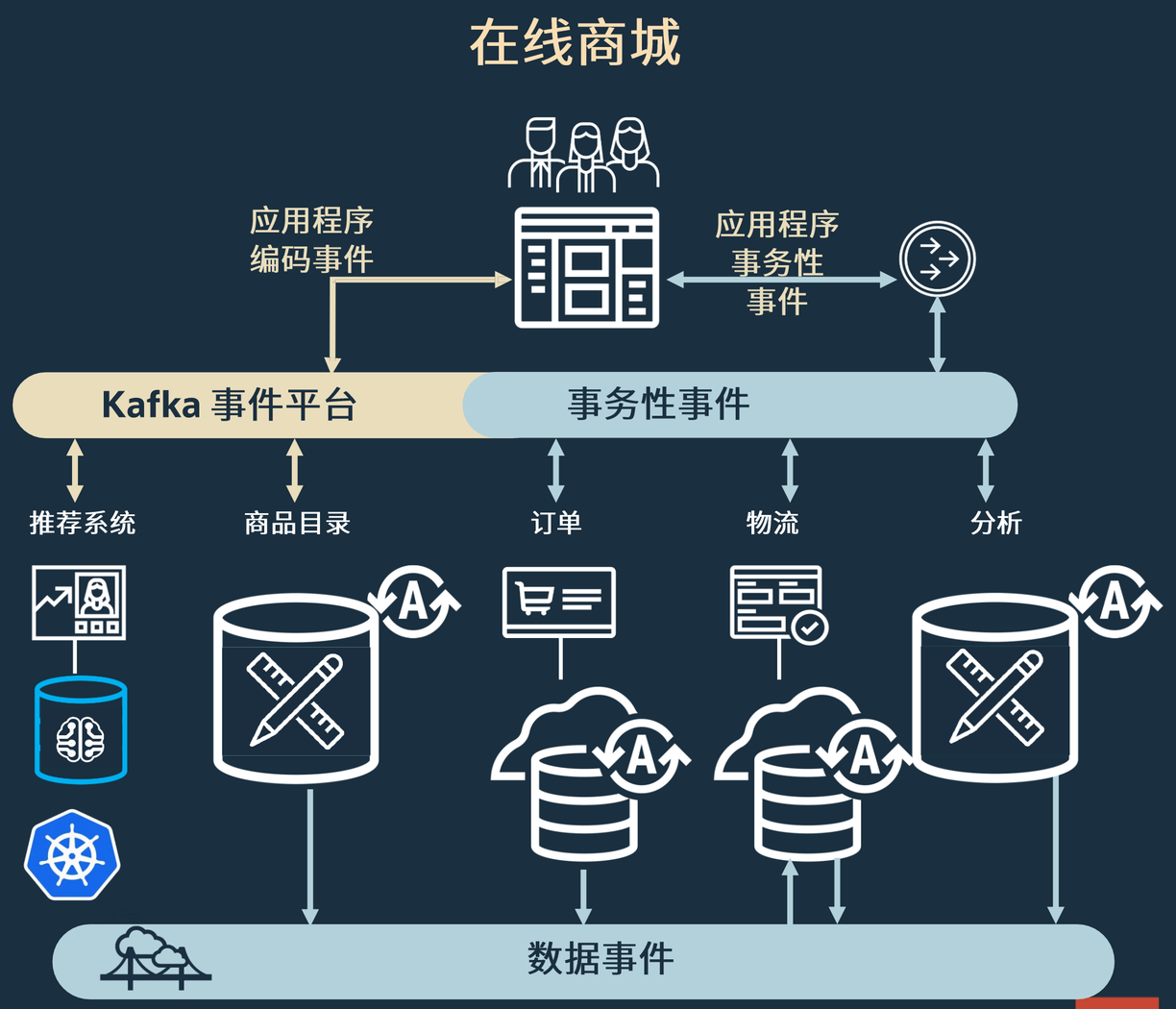


Copyright © 2022, Oracle and/or its affiliates

使用云技术部署和管理数据库

最后步骤

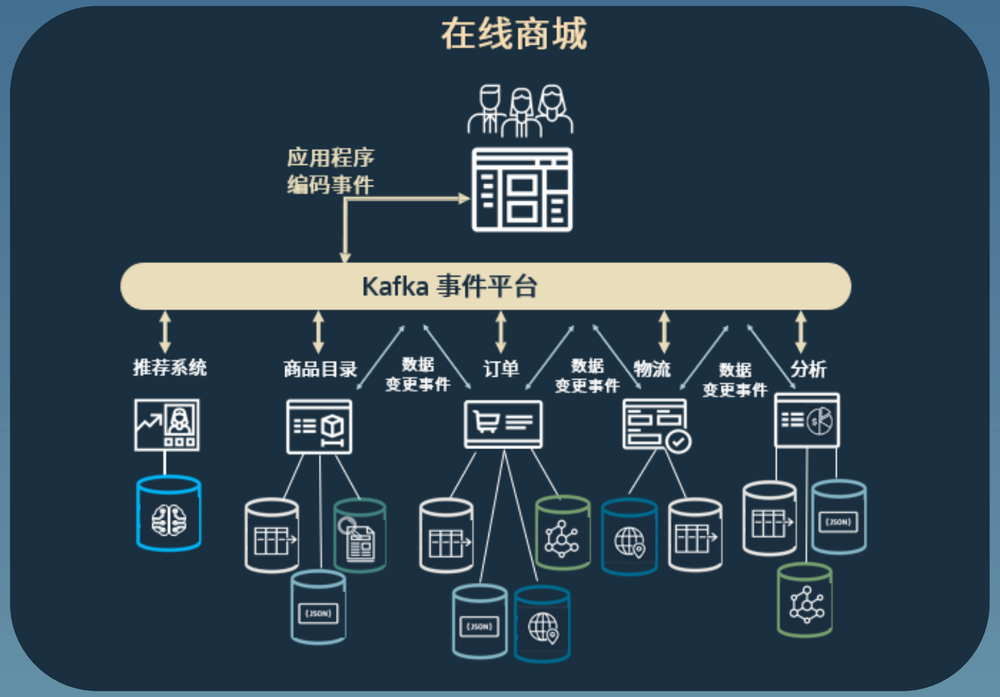
- 使用 Oracle Exadata 云服务自动化和优化整个数据库基础设施
  - 在公有云中或本地使用 Cloud@Customer
- 使用自治数据库使您的数据库自我管理、自我修复和自我保护

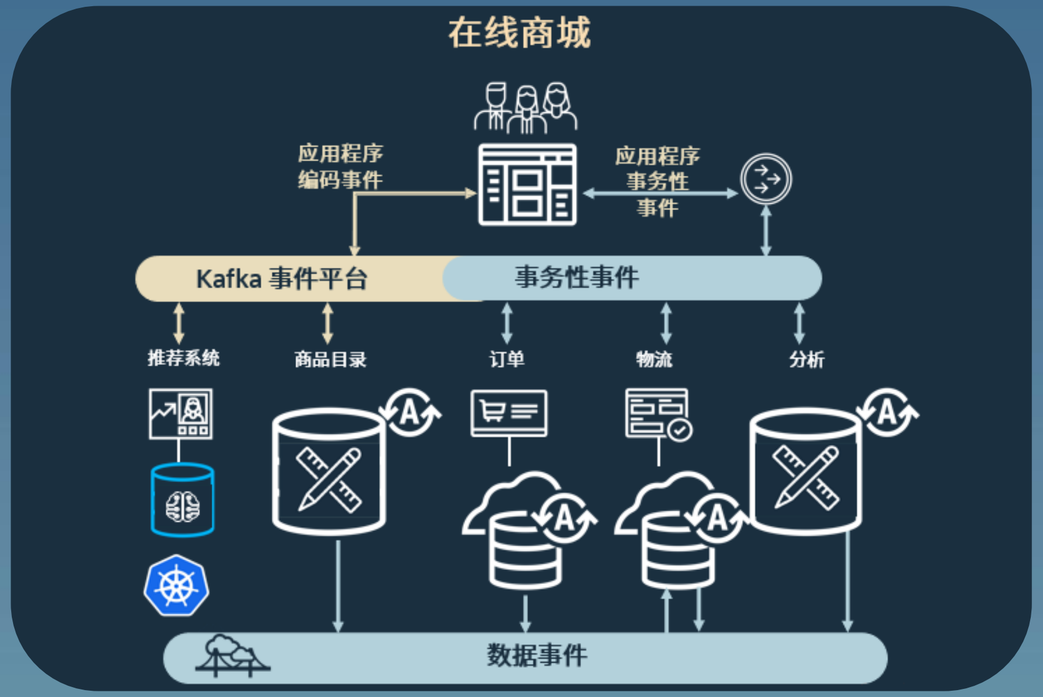


Copyright © 2022, Oracle and/or its affiliates

简化的在线商城

- 重新思考数据库架构以采用现代应用程序开发可以大大简化和更强大的应用程序





更多信息：**https://developer.oracle.com/microservices/**

Copyright © 2022, Oracle and/or its affiliates

关键信息

- 现代范例简化和加速应用开发
  - 促进应用和开发团队的微服务化
  - 然而，分布式应用程序开发使数据库架构复杂化

- Oracle的策略
  - 重新思考数据库架构，降低应用开发的复杂性和风险

Copyright © 2022, Oracle and/or its affiliates

ORACLE 甲骨文

Oracle数据库23c JSON关系二元性

Oracle Cloud World系列

陈文斌

- 资深解决方案工程师
- 成功迁移100余套系统上云
- 10年架构及研发经验，完成通信、金融、物流、仓库等领域系统的开发及上线

内容简介

- Oracle Database 23c Beta 支持 JSON关系二元性，让数据同时拥有关系型数据和JSON数据的形态,让开发效率加倍，让应用性能更好，让应用架构更简单
- Oracle Database 23c，代号为“App Simple”，专注于简化应用程序和开发

直播时间 : 12月23日 11:00 - 12:00

扫描二维码注册并安装手机Zoom进入直播

Zoom ID: 976 6962 5763 密码: 98039717

数据库和云讲座群

20-19

甲骨文云技术公众号

技术专家1V1深入交流