



CQRS实践

类目管理的开发

谷首道



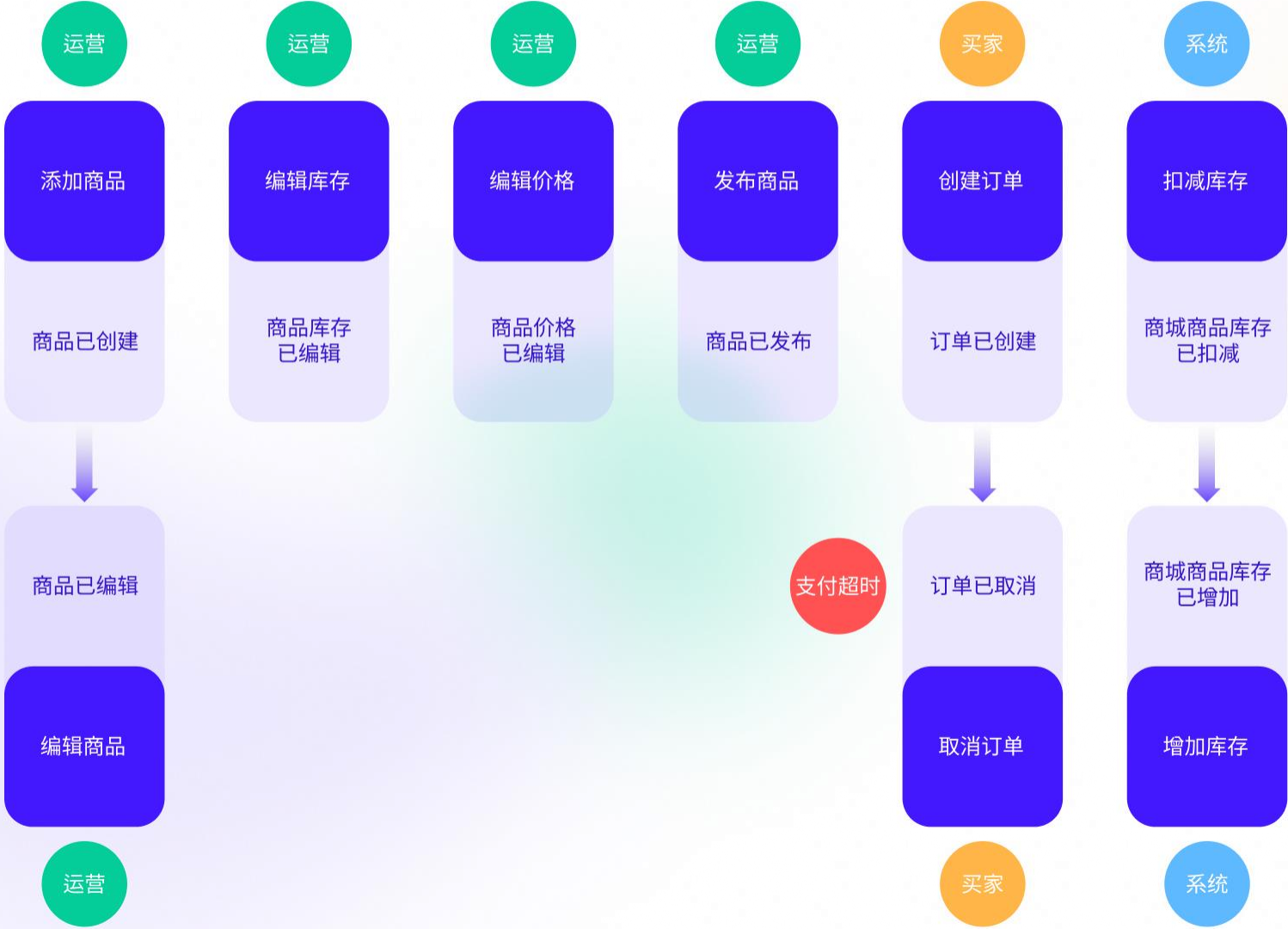


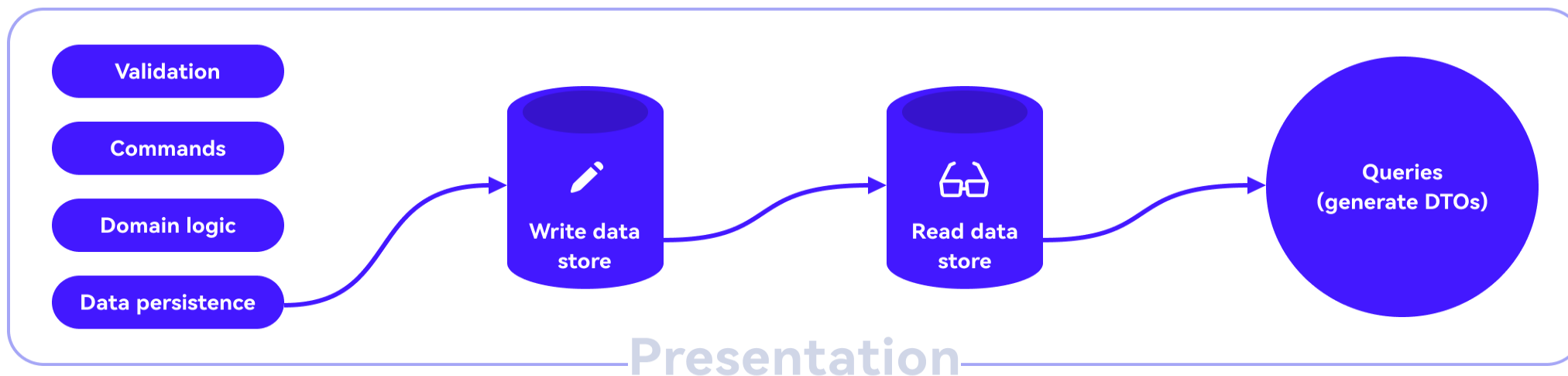
1 事件风暴回顾 - 事件流

2 CQRS

3 Event Bus

●●●● 角色/用户/系统
—— 命令
—— 事件





优势

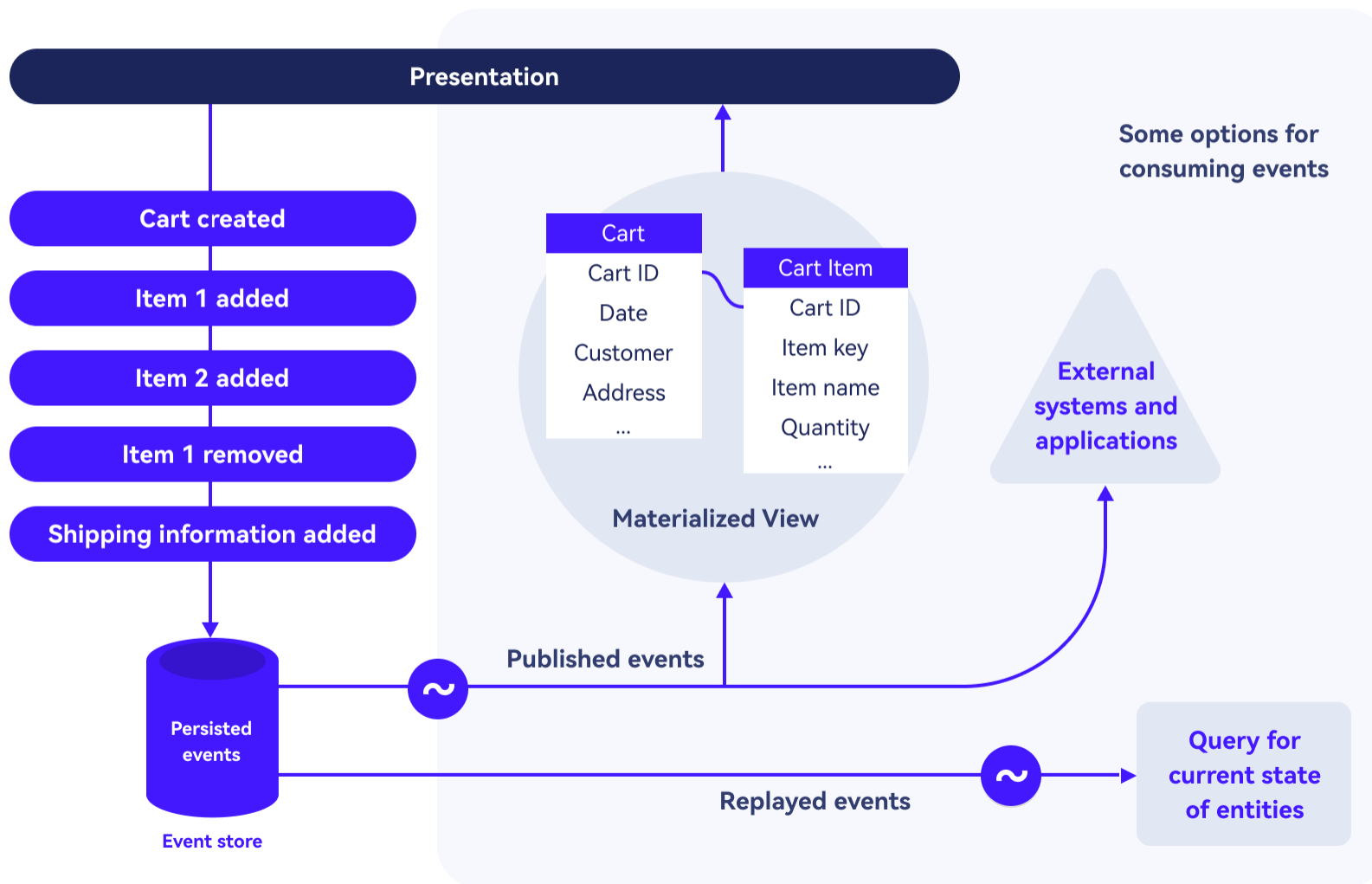
- ✓ 独立缩放
- ✓ 根据读写分别优化数据结构
- ✓ 更轻松确保仅正确的域实体对数据执行写入
- ✓ 关注点分离，读关注读取性能，写关注复杂业务逻辑
- ✓ 物化视图的读模型可有效减少复杂的表链接

什么时候用

- ✓ 源于访问频次、数据量或者是数据模型数量等因素导致查询和更新的需求差距逐渐变大
- ✓ 最大限度地提高性能
- ✓ 控制细粒度的缩放
- ✓ 控制不同应用的安全级别
- ✓ 降低多开发者协作冲突，事件的粒度可以足够细

注意

- ✓ 命令应基于任务，而不是数据
- ✓ 命令可入队，使用异步而非同步
- ✓ 查询不包含修改动作和领域知识
- ✓ 事件溯源支持重播，比如用于通知读模型



优势

事件不可变

事件有助于领域专家了解过程

事件只增不改，不需要考虑并发冲突

事件存储有助于监控数据变化

事件重播可轻易的对系统状态进行还原，调试，测试

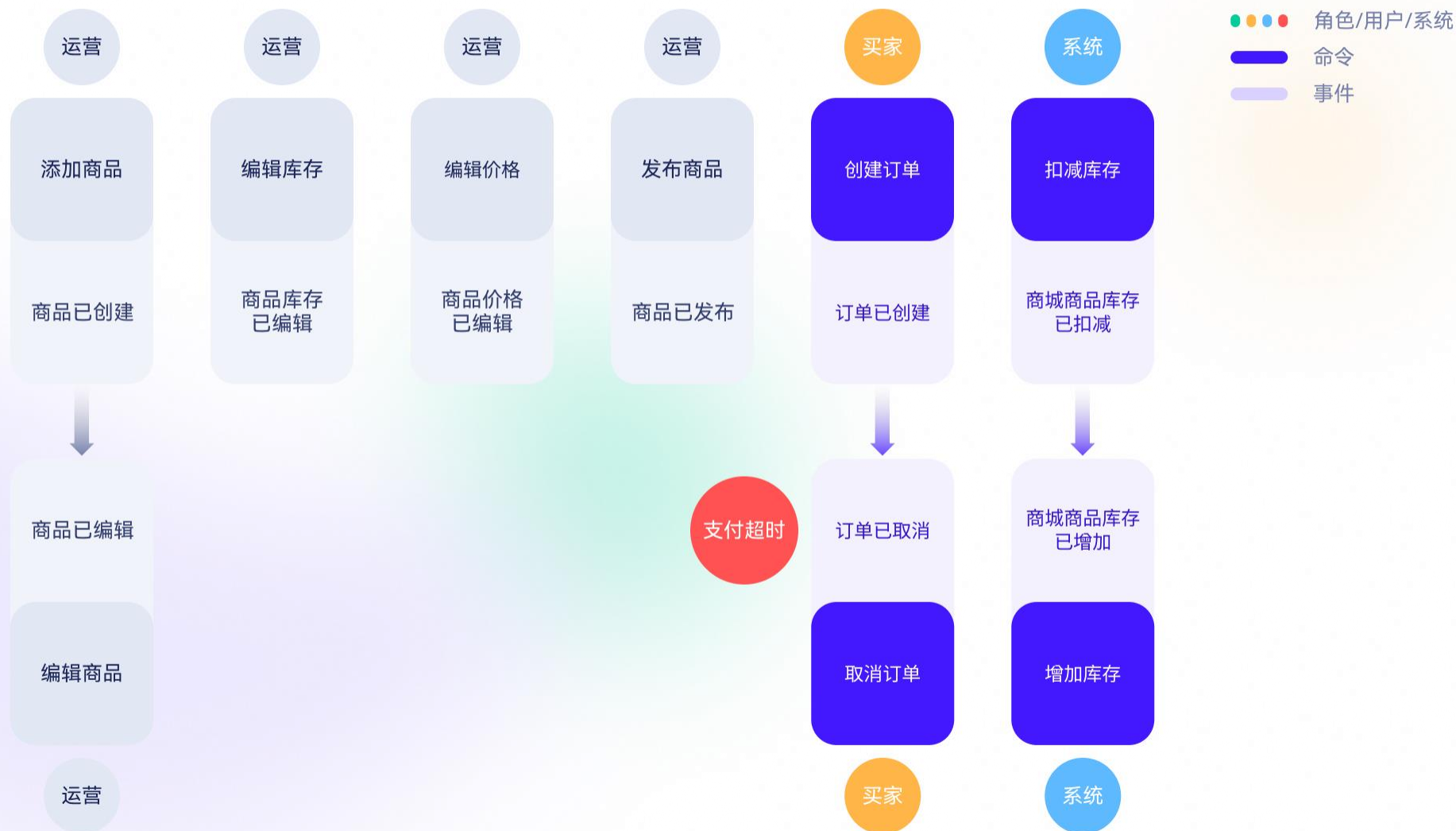
事件可被不同的应用多次，异步消费

注意

事件驱动是异步的，读模型会有一定程度的延迟

修改事件的唯一方式只能通过补偿事件

事件顺序至关重要，事件存储有责任保证事件顺序完整性



事件驱动的小知识点

事件 != 命令

事件不会告诉订阅者如何做

事件只通知某个事件发生了

事件是不可变的

事件被执行时，可能会产生新的事件并形成事件流

事件标准化

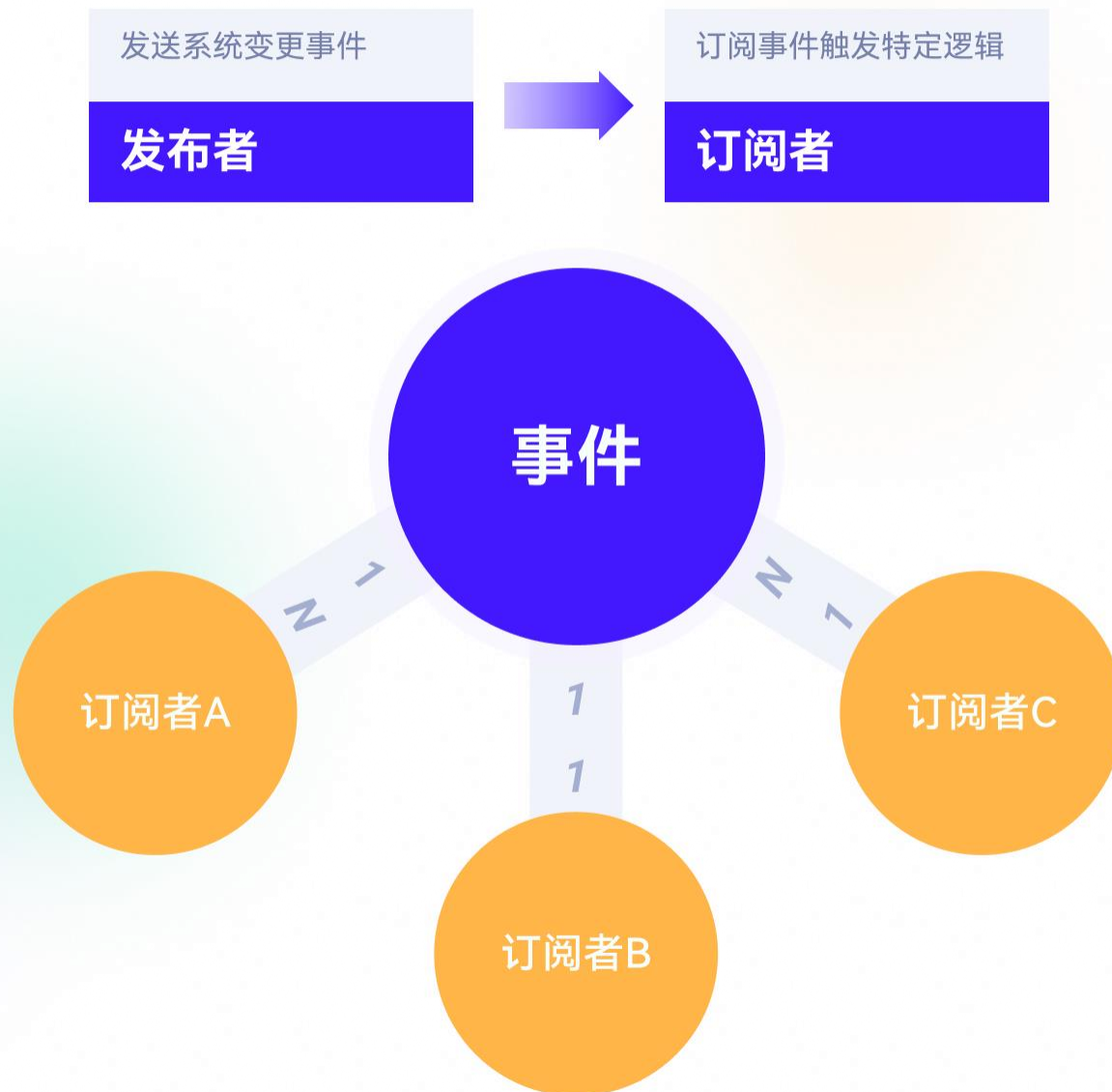
支持过滤/转换/查询等

CloudEvents

包含了事件发生的上下文和相关数据，事件代表了已发生的事实，不包含任何目的地相关信息。消息能够传达事件内容，从而将事件数据从源头传输到指定的目的地

发送的消息符合规范，那么它就是一个有效的 CloudEvent

支持的云厂商包括微软，谷歌，阿里，甲骨文等



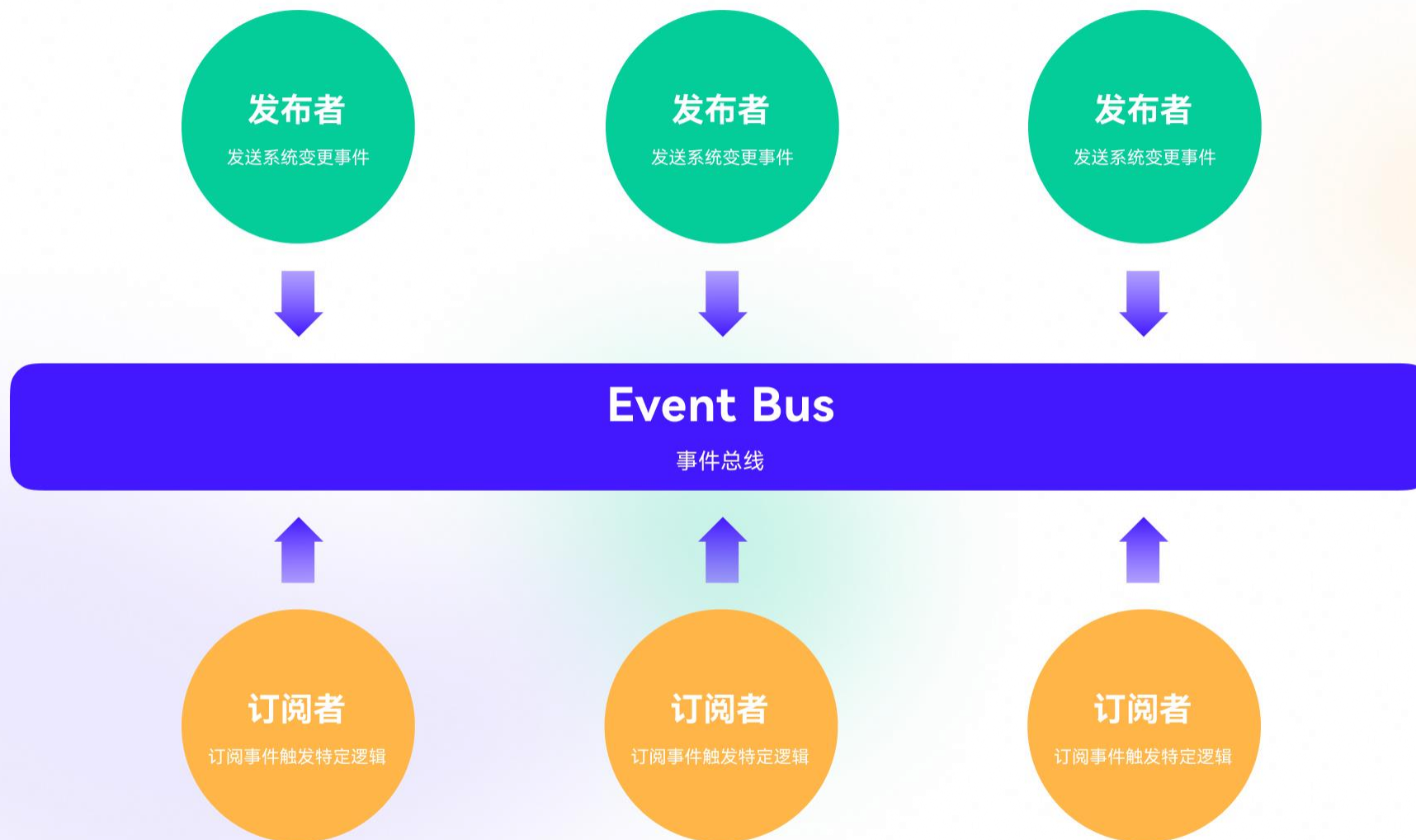
微服务解耦
跨集群通信

兼顾遗留
系统对接

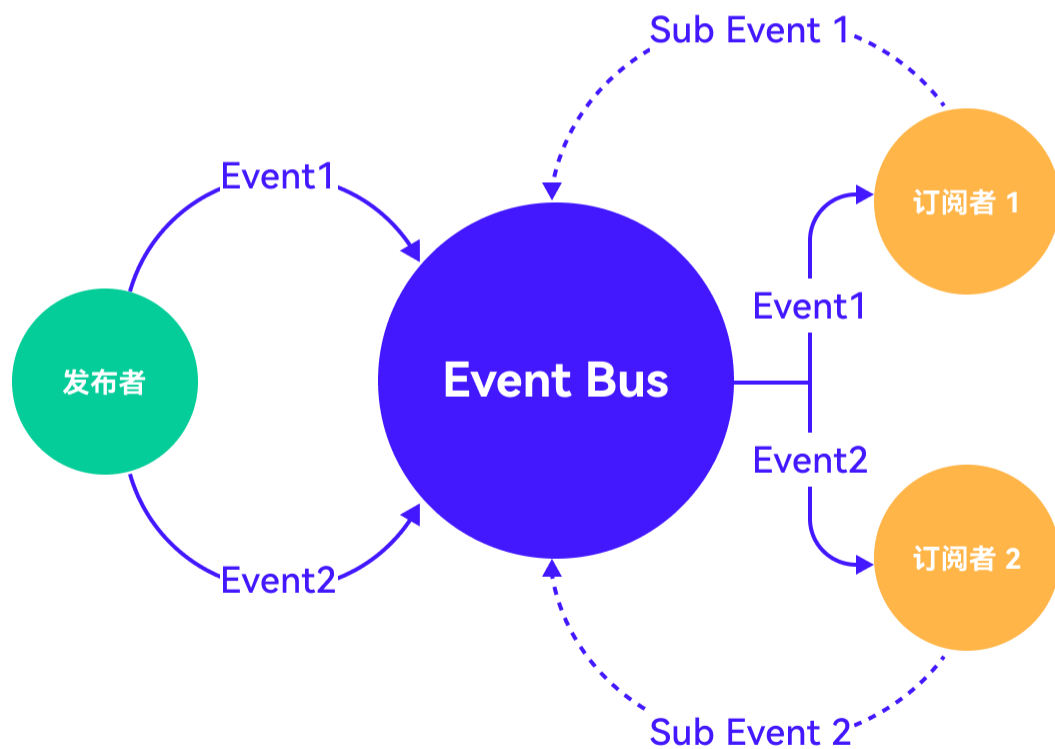
游走在
云与非云中

部分 AOP
类场景

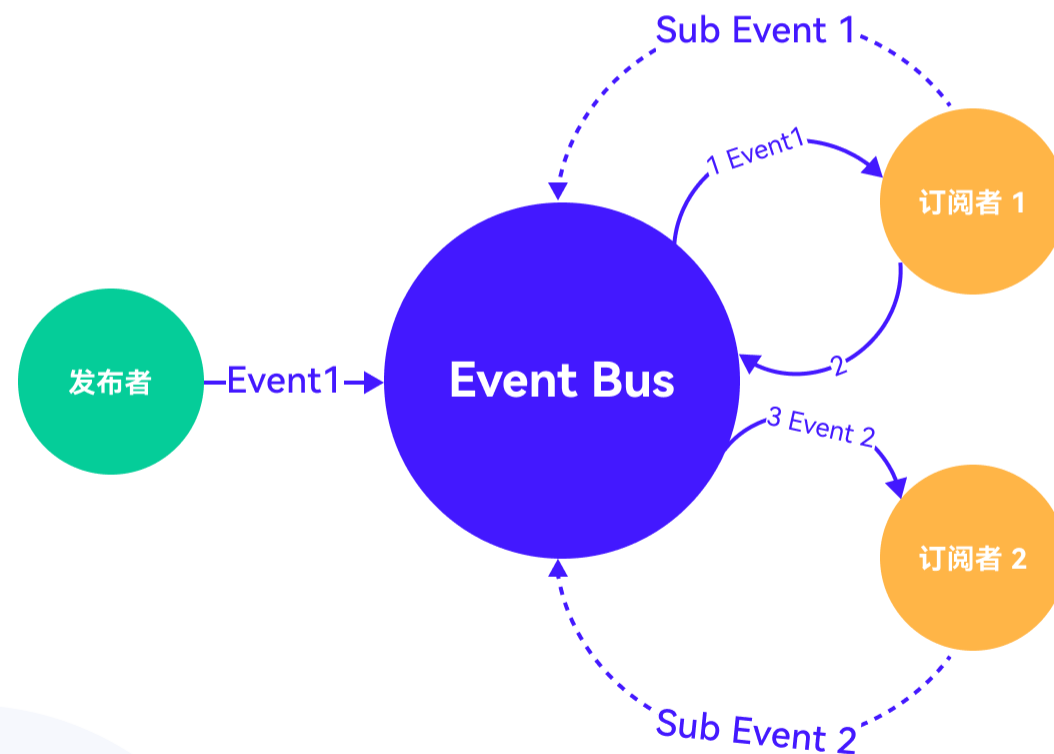
流计算



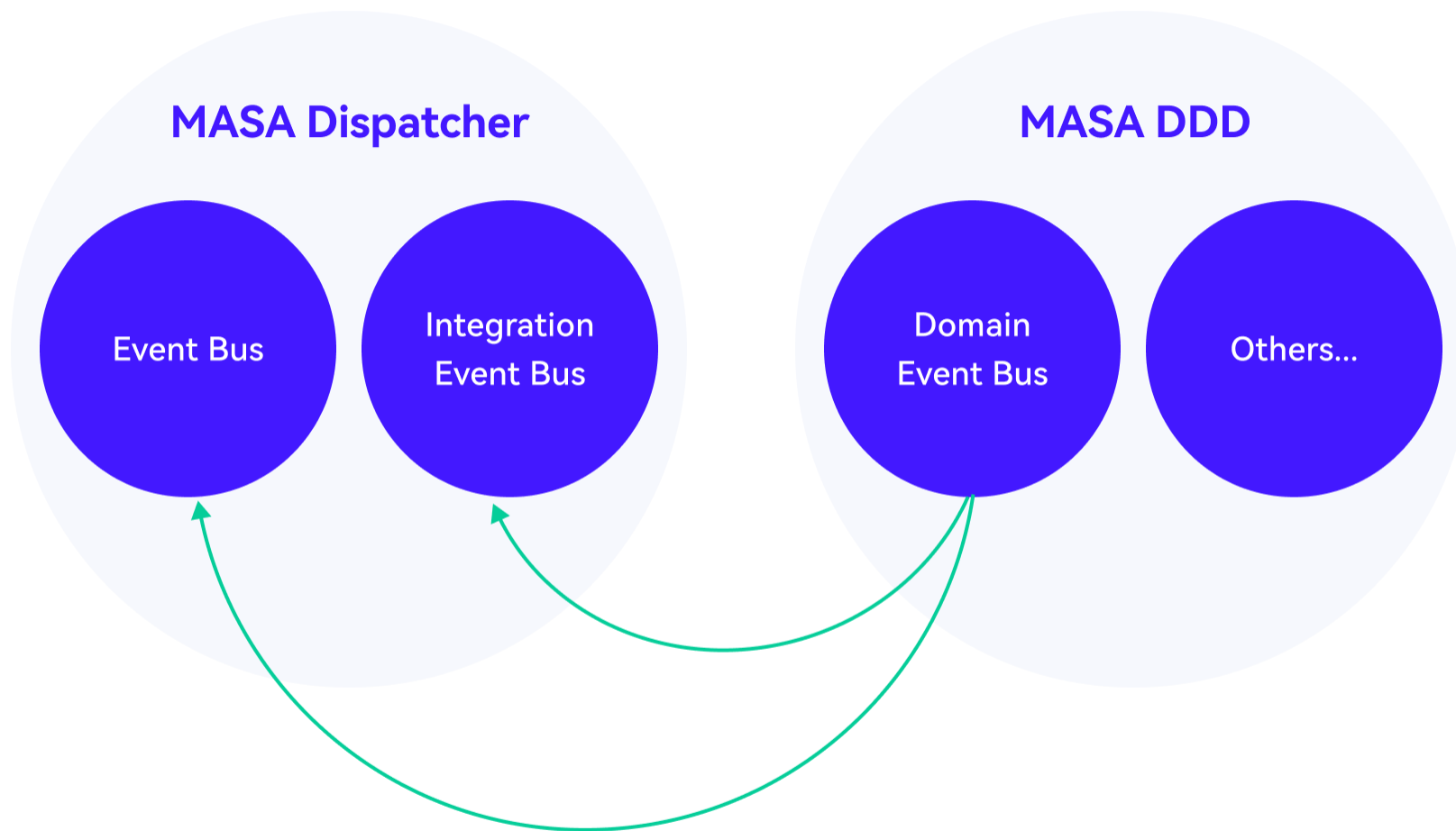
并行执行



顺序执行



动态执行?



进程内事件总线

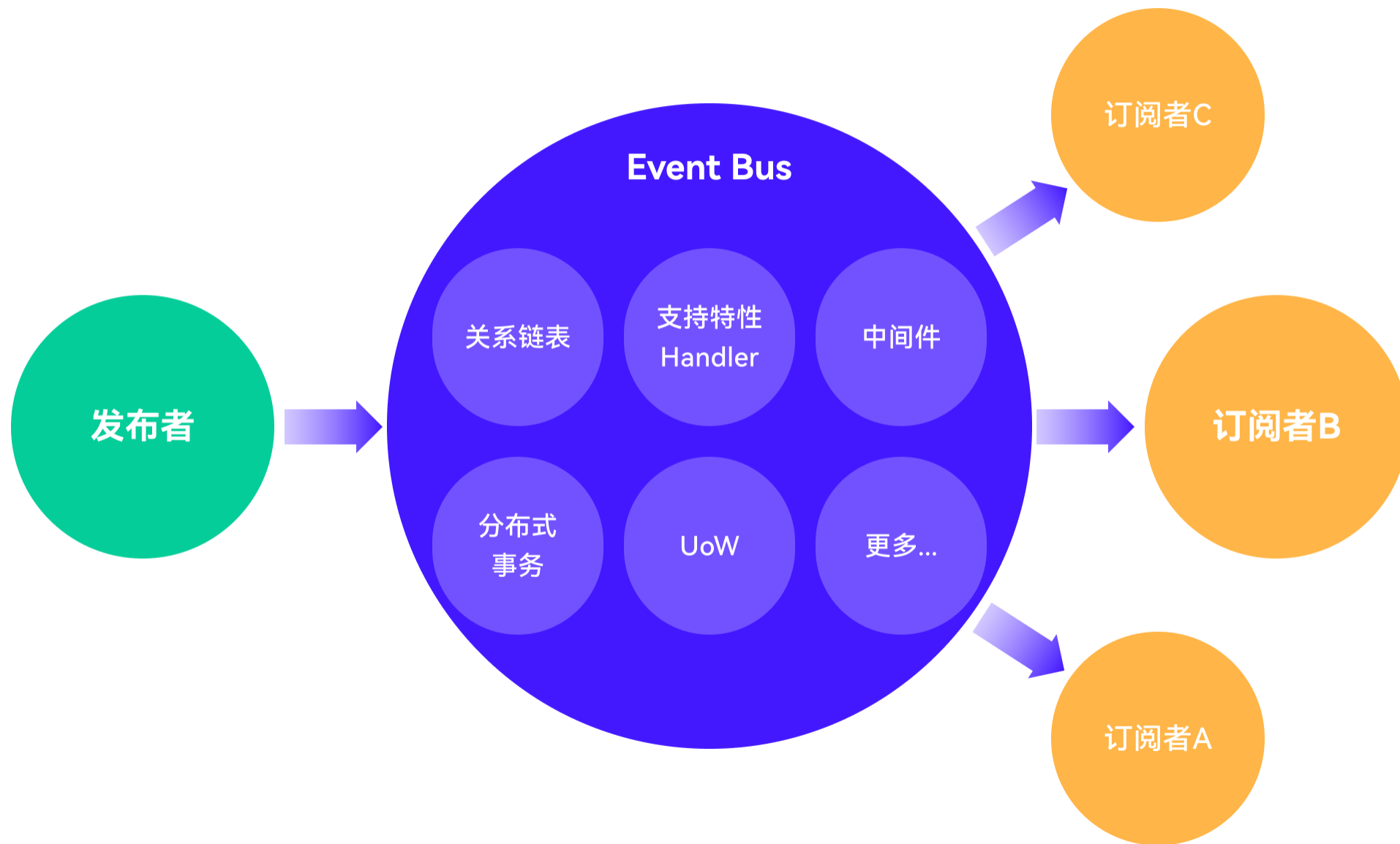
集成事件总线（跨进程）

领域事件总线（自动协调进程内事件和继承事件调度）

Event Bus

Integration Event Bus

Domain Event Bus



第一课

MASA Framework的设计理念

- 1.什么是现代应用开发？
- 2.MASA Framework的诞生
- 3.为什么选择MASA Framework？

第二课

如何做产品设计

DDD实践、事件风暴、C4模型、架构设计、前后分离

第三课

解决方案搭建

技术栈清单、项目准备、初始化、MASA Blazor介绍、用户登录

第四课

开发类目管理

CQRS、RESTful、记录事件（Dispatcher Middleware）

第五课

开发商品管理

接口缓存、双写、Fluent Validation、审计、软删除、映射；图片上传存储；Dapr Service Invocation

第六课

开发首页、商品详情页

高并发解决方案、缓存设计

第七课

开发订单主线流程（一）购物车、下单

最终一致性（本地消息、补偿、重试、幂等）、下单异常（Saga）、通知推送（模板、SignalR）

第八课

开发订单主线流程（二）秒杀、支付

高并发（Actor）、最终一致性

第九课

开发订单主线流程（三）订单查询

AutoComplete用法、Elasticsearch操作技巧

第十课

多语言实现

前后端方案、MASA DCC进阶

第十一课

隔离性实现

租户隔离、环境隔离

第十二课

全局异常处理

I18n、错误码、用户友好异常

第十三课

可观测性配置

OpenTelemetry、.Net Diagnostics、Dapr Observability

第十四课

测试

Mock、TDD

第十五课

回顾和总结

- 1.进阶课程安排预告
- 2.如何做开源贡献

谢谢观看

Github项目地址 欢迎



关注

<https://github.com/masastack/MASA.Framework>

官网

<https://www.masastack.com/>

MASA 技术团队

微信号:MasaStackTechOps

