数据一致性解决方案

姜宁

willem.jiang@gmail.com

关于我


中国航信
TravelSky


redhat


HUAWEI


ActiveMQ


Apache CXF


Apache
Camel


ServiceMix™

让云原生开发更简单

Github：https://github.com/ServiceComb

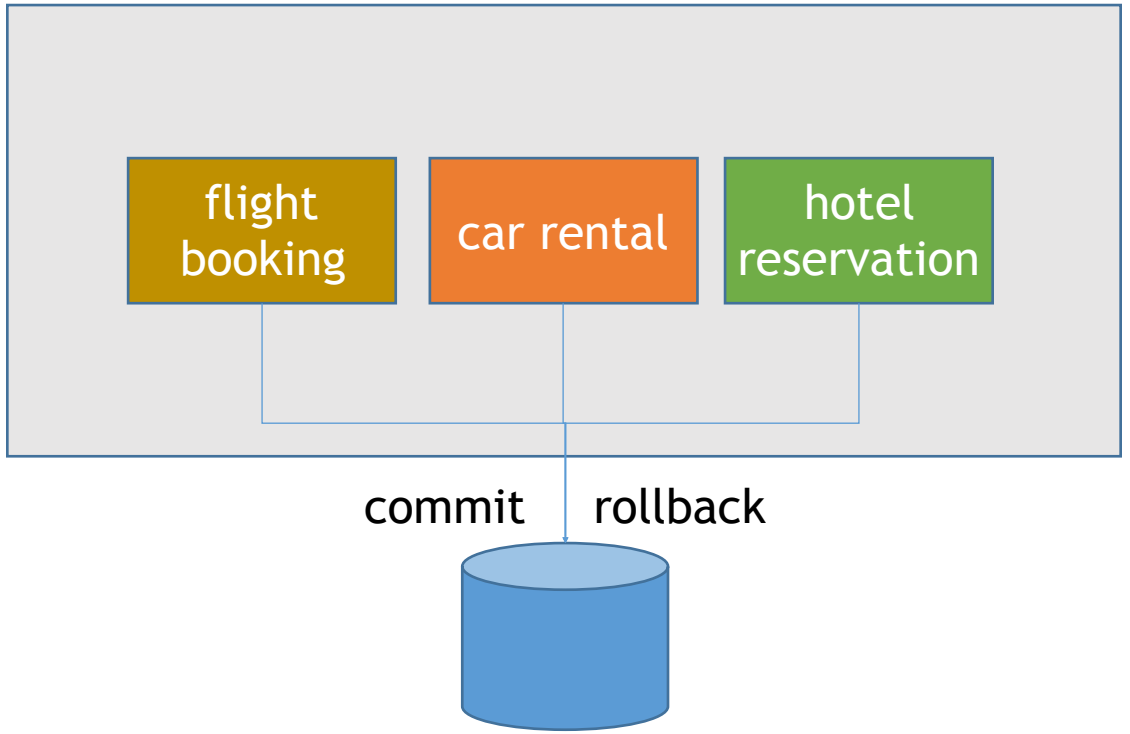
官网：https://www.servicecomb.io

大纲

- 数据一致性的起因
- 数据一致解决方案
- Saga简介
- ServiceComb Saga 实现演示

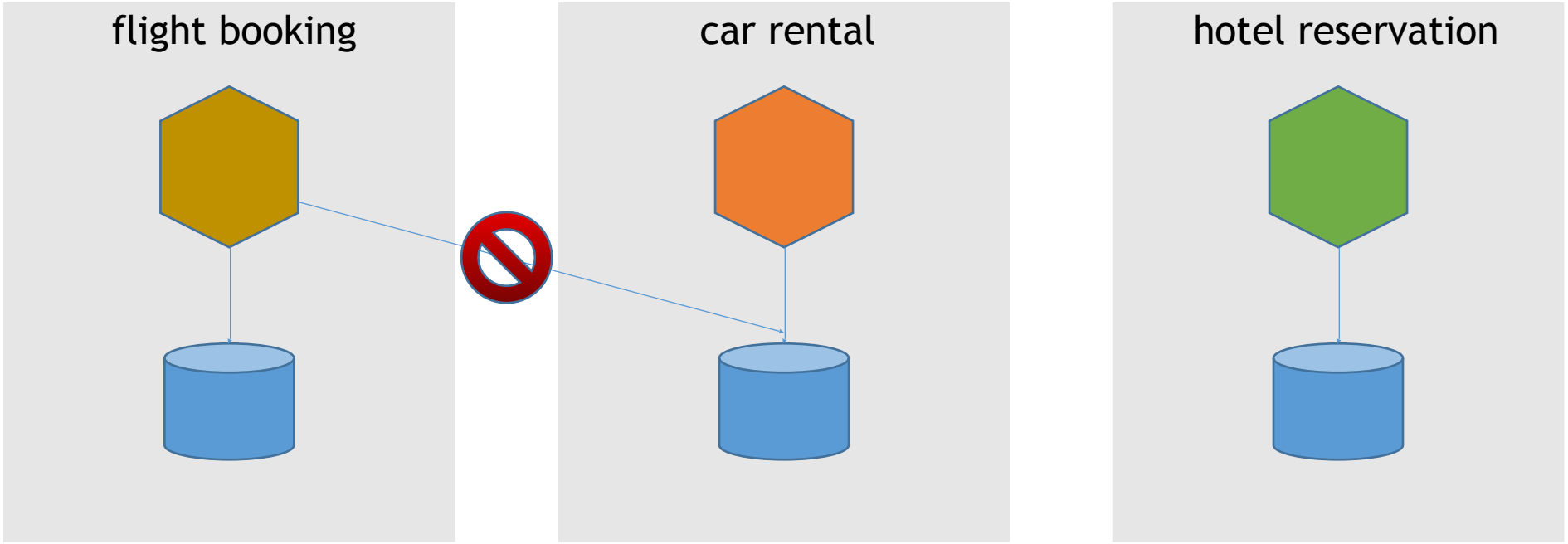
单体应用

- 单体应用由于所有模块(A/B/C)使用同一个数据库，数据一致性通过数据库transaction保证



微服务场景

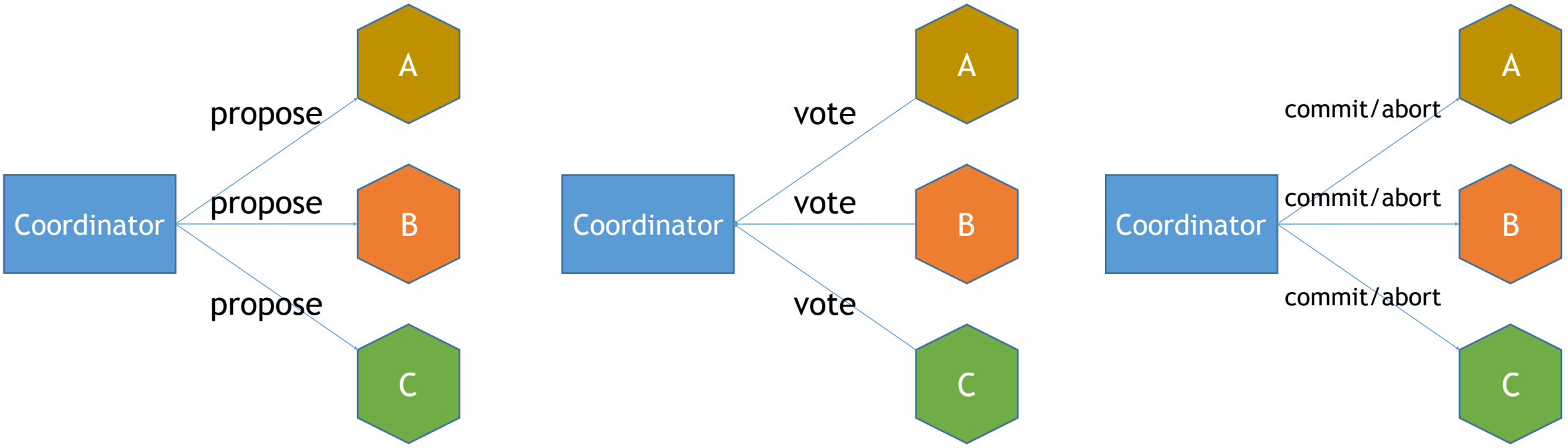
- 每个微服务的数据库为私有，而且数据库选型也可能不同，导致数据一致性无法通过数据库保证



解决方案

- 两阶段提交 (2PC)
- 事件驱动 (Event Driven)
- Try/Confirm/Cancel (TCC)
- Saga

两阶段提交(2PC)

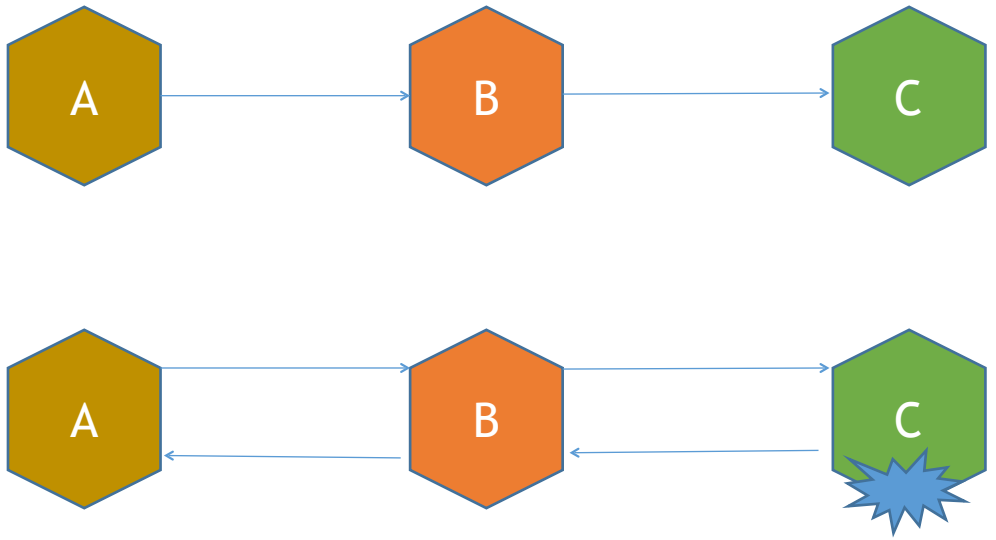

Coordinator
propose
propose
propose
A
B
C
Coordinator
vote
vote
vote
A
B
C
Coordinator
commit/abort
commit/abort
commit/abort
A
B
C

两阶段提交 (2PC)

- 优点：
  - 数据一致性好
- 缺点：
  - 阻塞式，锁定资源
  - 复杂度高，难以扩展 $O(n^2)$
  - 如果commit阶段coordinator挂了，事务将处于悬而未决状态

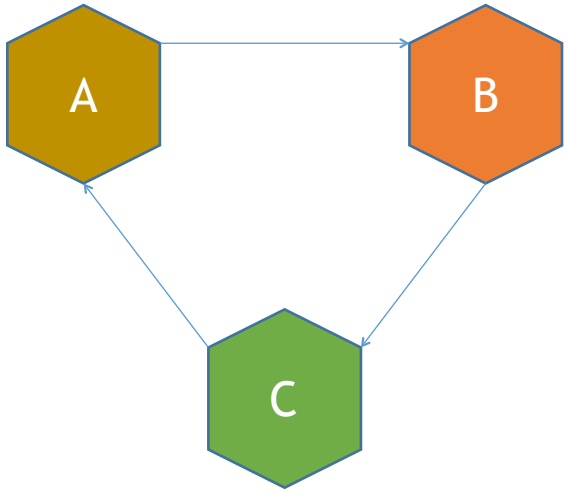
事件驱动方式

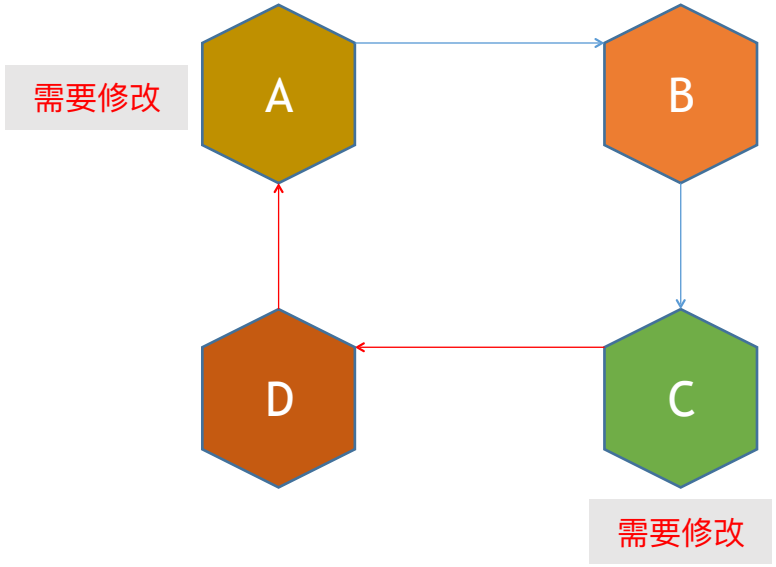
- 建立消息队列基础上，保证消息不丢不重。
- 通过消息将本地事务串接起来。



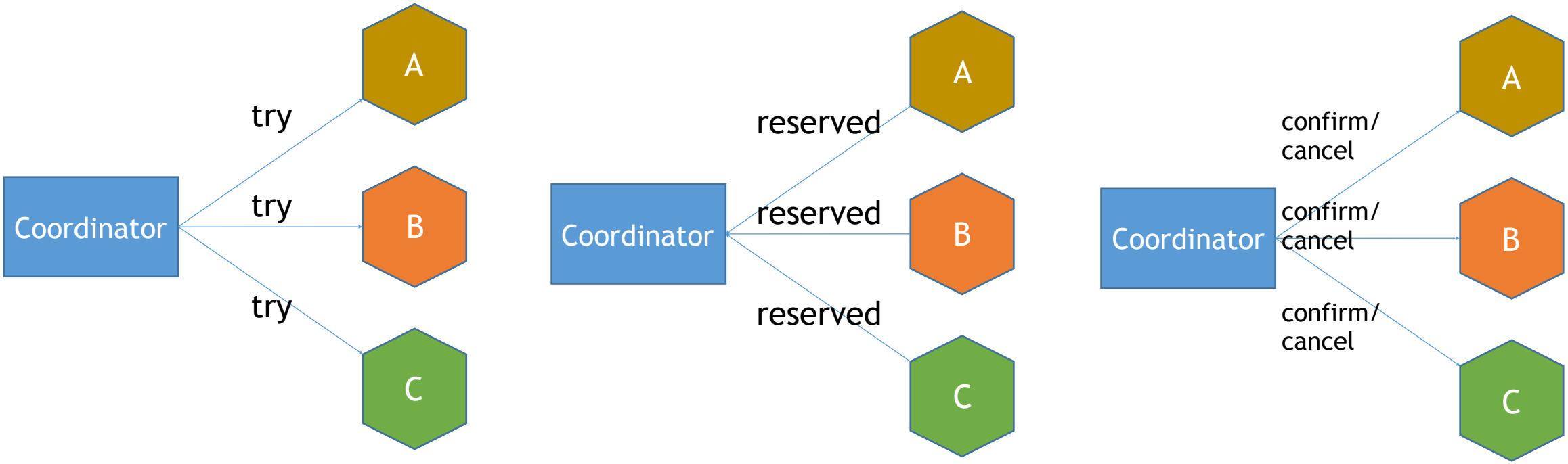
事件驱动

- 优点：
  - 服务自组织，去中心化
- 缺点：
  - 服务互相耦合，新增服务导致服务间事件流转变化
  - 服务越多，服务间事件流转越难可视化，难以定位问题





TCC (Try/Confirm/Cancel)

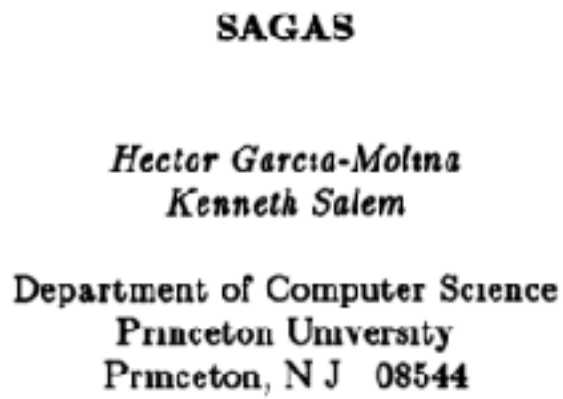


TCC (Try/Confirm/Cancel)

- 优点：
  - 如果在try阶段失败，服务可恢复至原状态
  - 服务间无耦合，新增服务不需要修改原有服务
- 缺点：
  - 在confirm阶段失败，对已完成事务的补偿不一定能恢复服务至原状态
  - 额外的try流程，服务需要提供额外try接口，实现额外reserved状态
  - 所有阶段需要等待参与方执行完毕才能完成
  - 中心化，coordinator需要容错能力

Saga简介

- 1987年Hector & Kenneth 发表论文 Sagas
- Saga = Long Live Transaction (LLT)
- LLT = T1 + T2 + T3 + ... + Tn
- 每个本地事务Tx 有对应的补偿 Cx
- 已知使用saga的厂商： Microsoft/Twitter/Uber



SAGAS

Hector Garcia-Molina
Kenneth Salem

Department of Computer Science
Princeton University
Princeton, N J 08544

T1 T2 T3 ... Tn
C1 C2 C3 ... Cn

T1 T2 T3 ... Tn

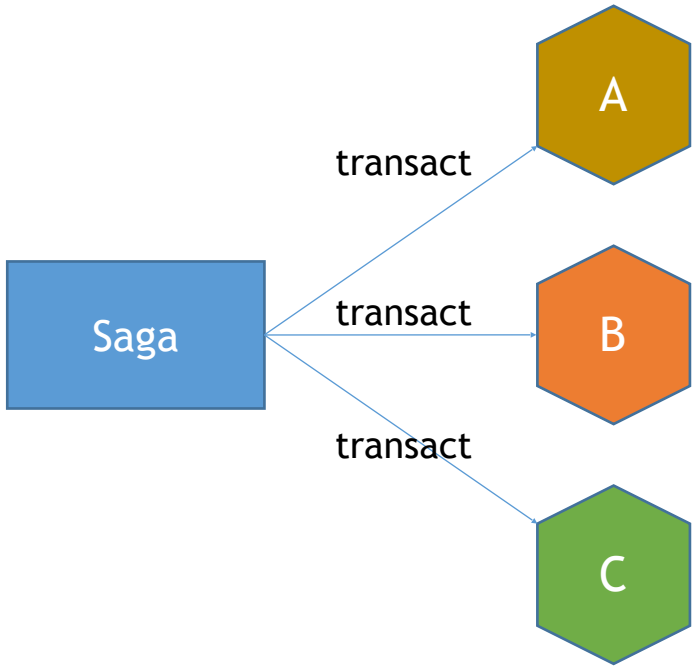
正常情况

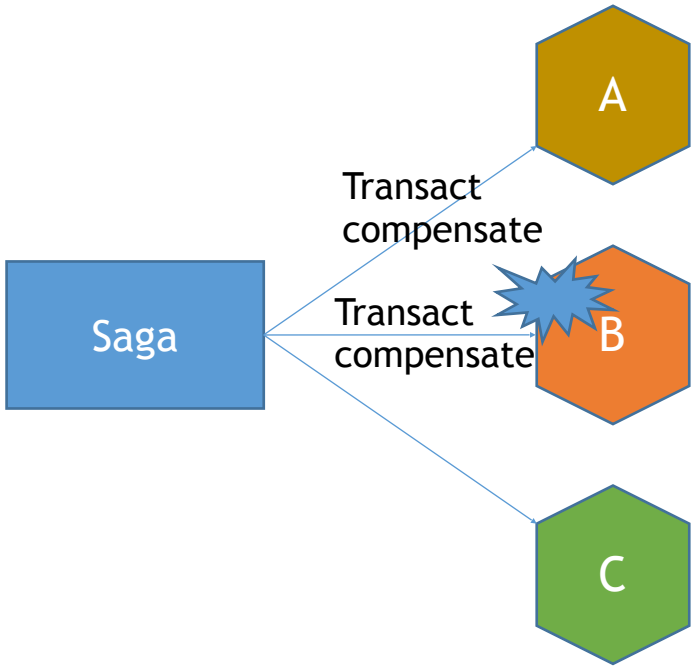
T1 T2 T3 C3 C2 C1

异常情况

https://www.cs.cornell.edu/andru/cs711/2002fa/reading/sagas.pdf

分布式Saga


A
transact
Saga
transact
B
transact
C


A
Transact
compensate
Saga
Transact
compensate
B
C

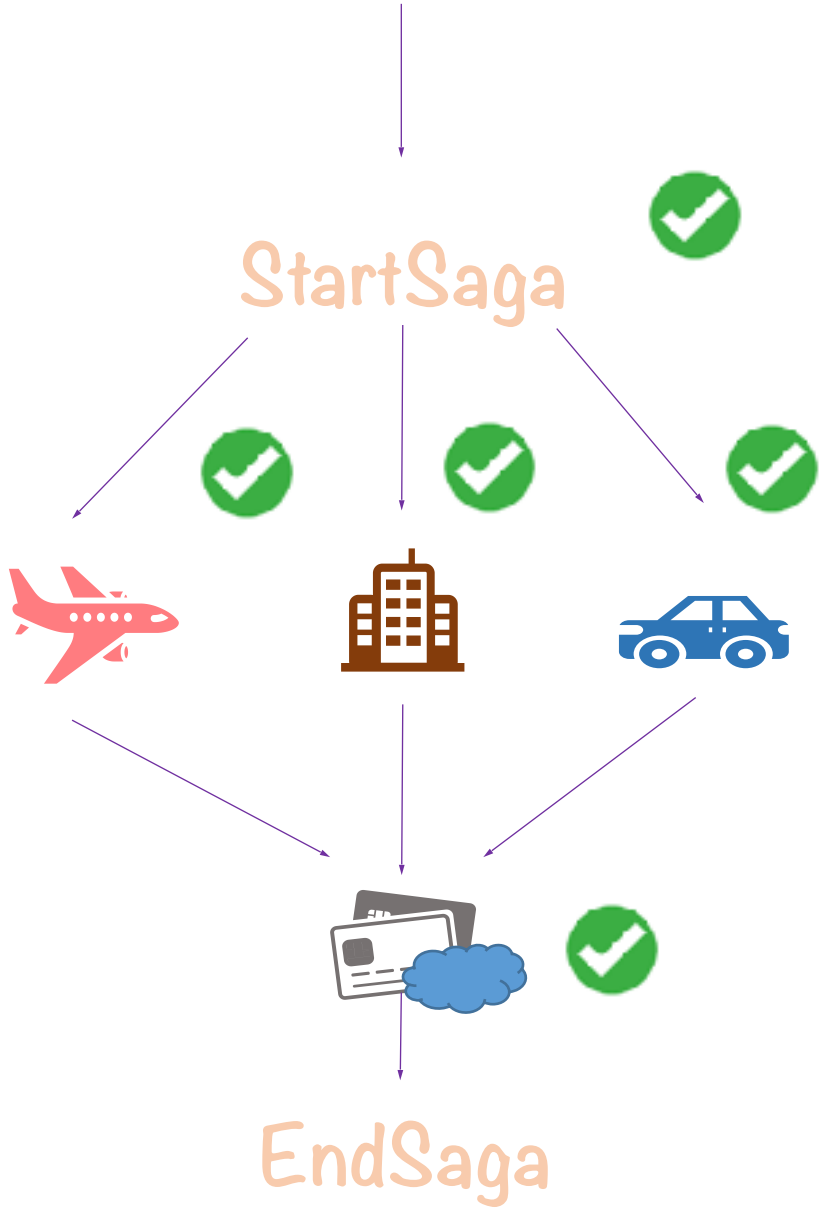
分布式Saga

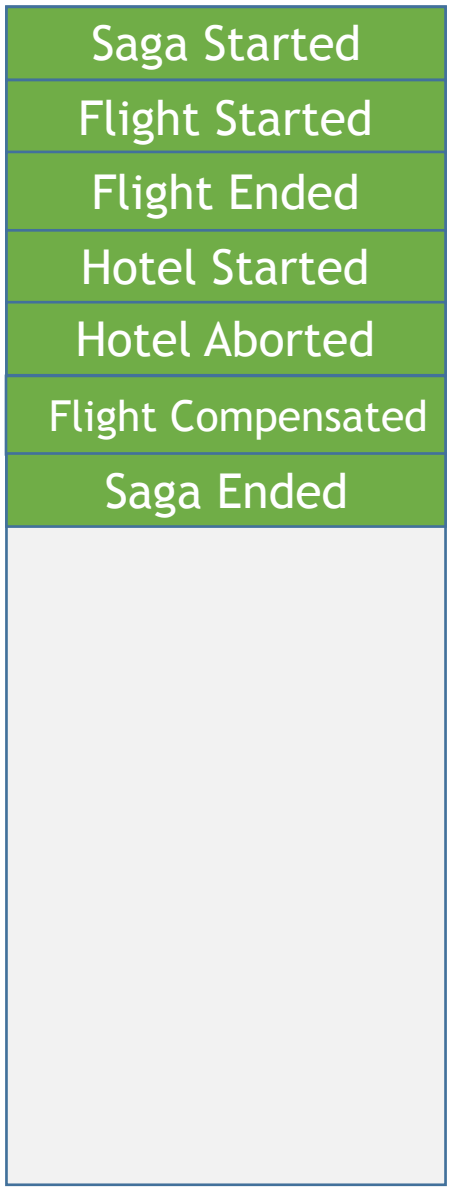
- 优点：
  - 服务间无耦合，新增服务不需要修改原有服务
  - 不需要服务提供额外接口 (通常服务有类似commit/abort接口)
  - 服务流程可基于Saga可视化，容易定位问题
- 缺点:
  - 对事务的补偿不一定能恢复服务至原状态
  - 中心化，Saga需要容错能力

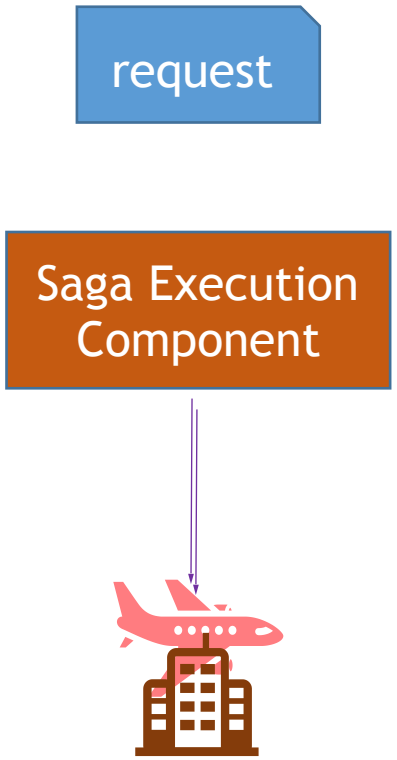
Saga Started
Flight Started
Flight Ended
Hotel Started
Hotel Ended
Car Started
Car Ended
Payment Started
Payment Ended
Saga Ended

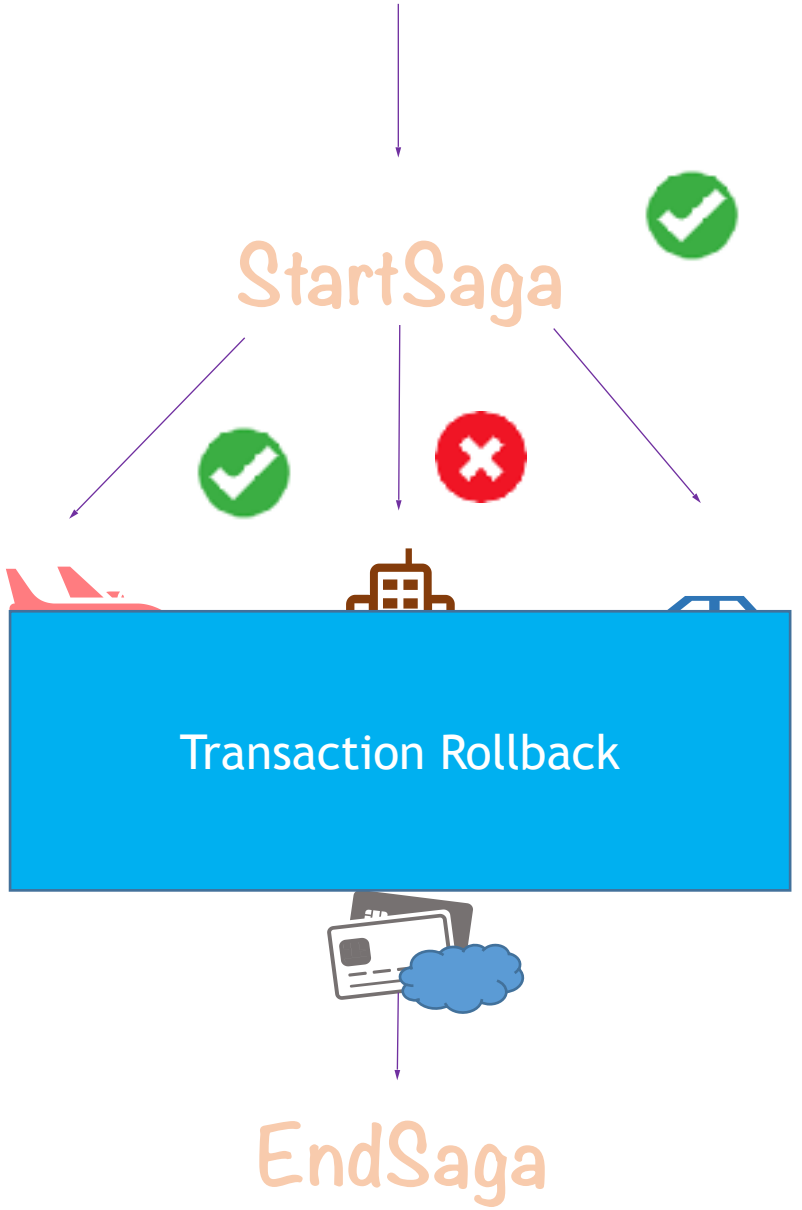
request

Saga Execution Component



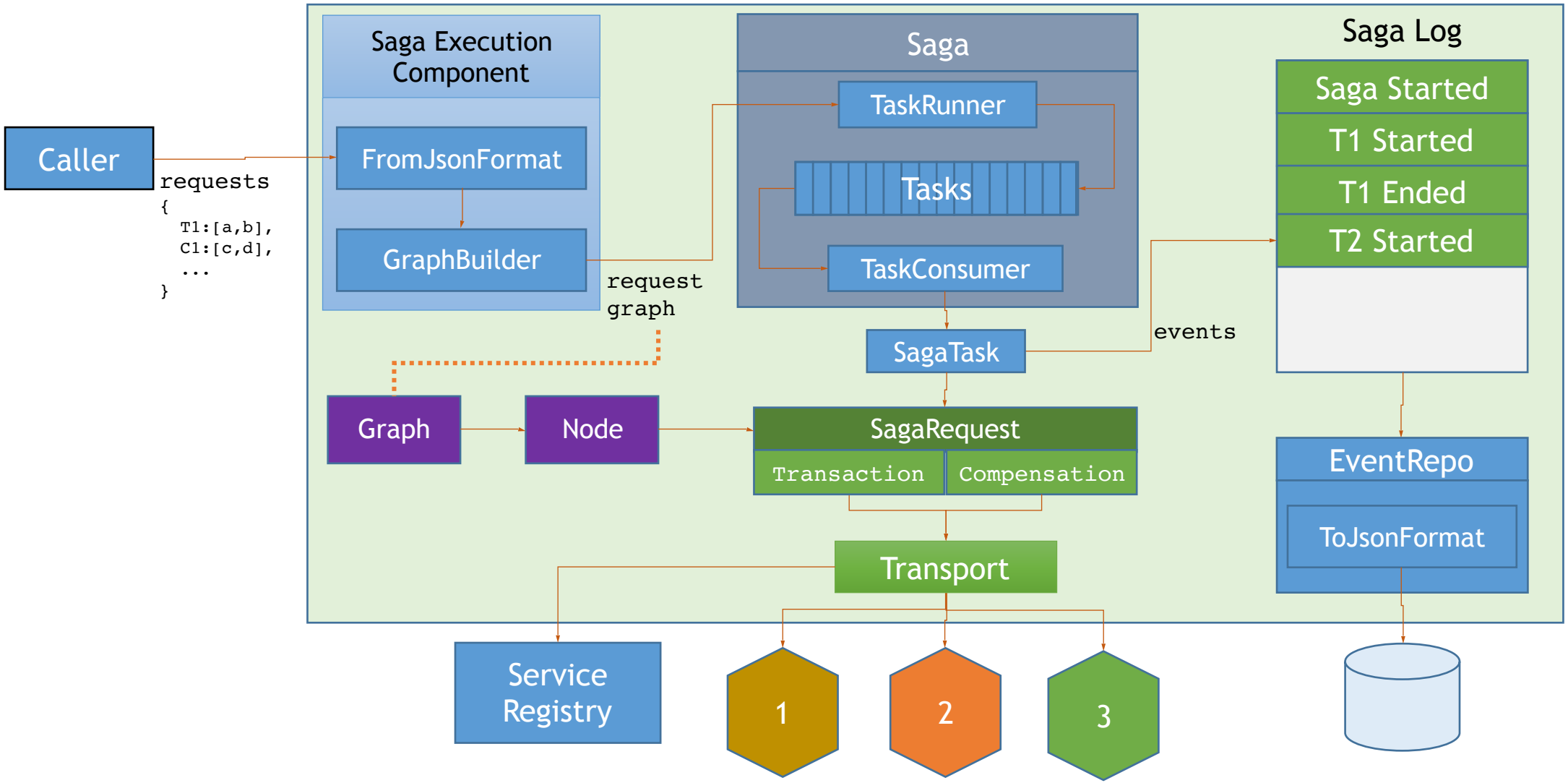

Saga Started
Flight Started
Flight Ended
Hotel Started
Hotel Aborted
Flight Compensated
Saga Ended


request
Saga Execution Component


StartSaga
Transaction Rollback
EndSaga

Demo Time

系统架构


Caller
requests
{
T1:[a,b],
C1:[c,d],
...
}
Saga Execution Component
FromJsonFormat
GraphBuilder
request graph
Saga
TaskRunner
Tasks
TaskConsumer
SagaTask
events
Saga Log
Saga Started
T1 Started
T1 Ended
T2 Started
Graph
Node
SagaRequest
Transaction
Compensation
EventRepo
ToJsonFormat
Transport
Service Registry
1
2
3

让云原生开发更简单

Github：https://github.com/ServiceComb

官网：https://www.servicecomb.io