和平统一

微服务场景下的数据一致性解决方案

殷湘
华为PaaS微服务架构师
开源能力中心

大纲

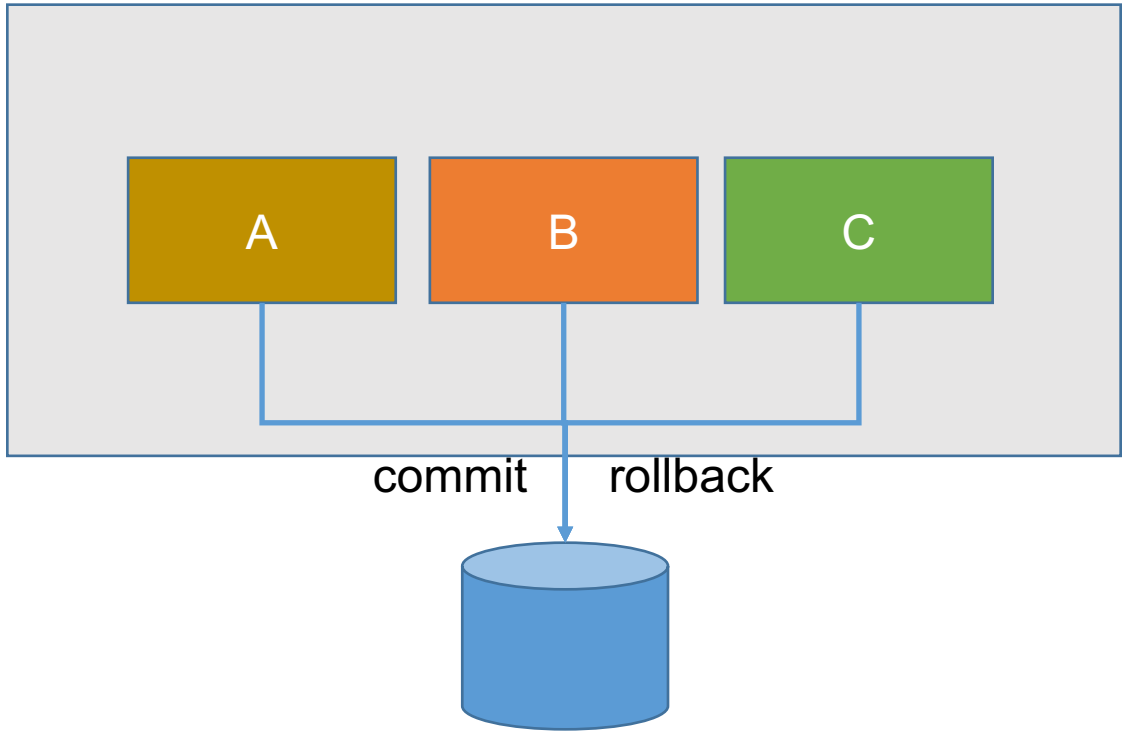
- 离 数据一致性的起因
- 合 数据一致性的解决方案
- 断 方案选择建议

离

数据一致性的起因

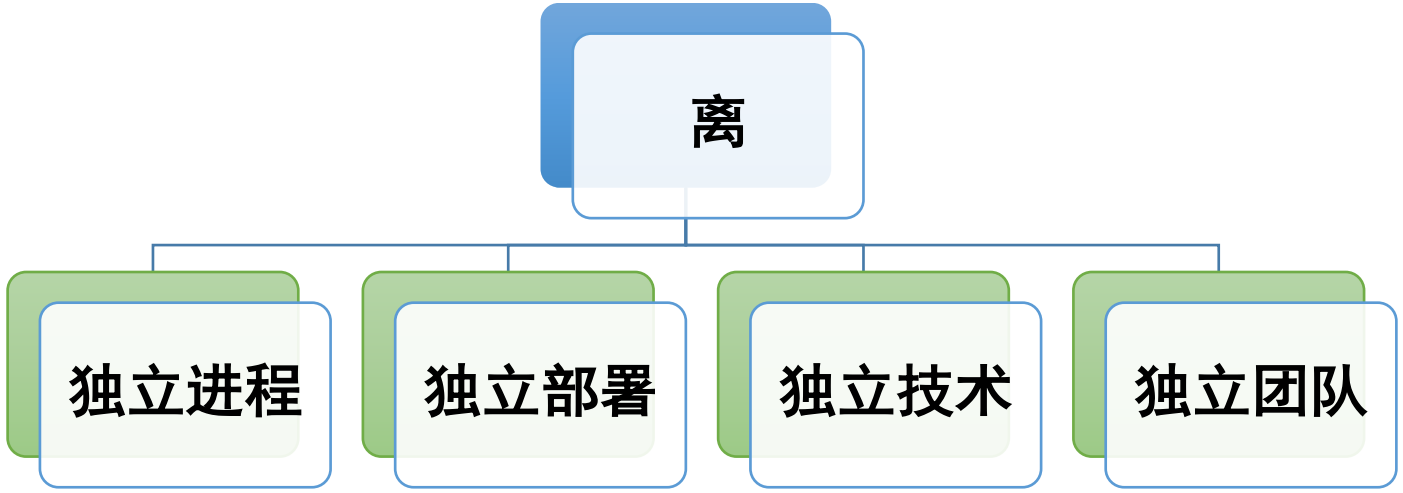
单体应用

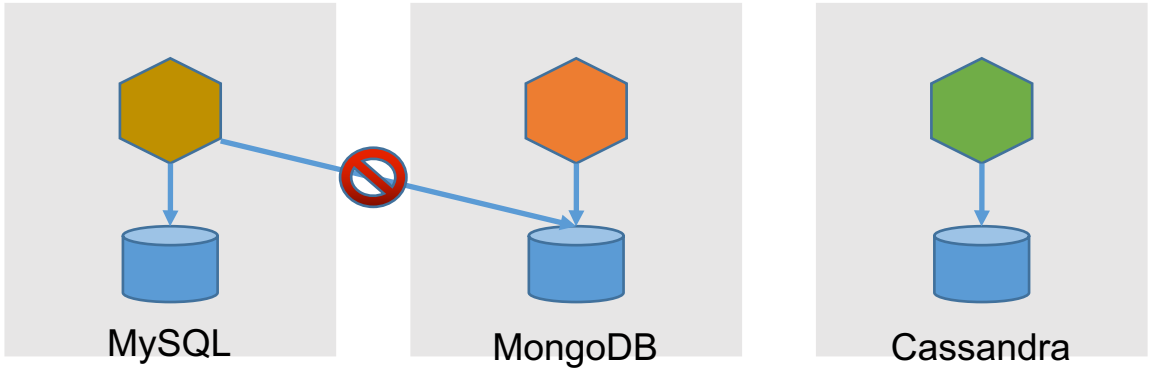
- 单体应用由于所有模块(A/B/C)使用同一个数据库
- 数据一致性通过数据库事务保证



微服务场景

数据一致性无法完全通过数据库保证





合

数据一致性的解决方案

Saga

- 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

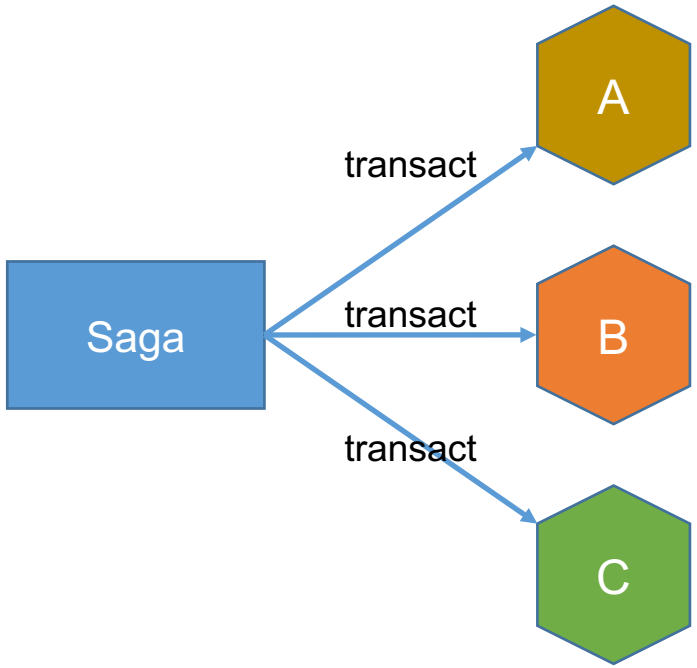
正常情况

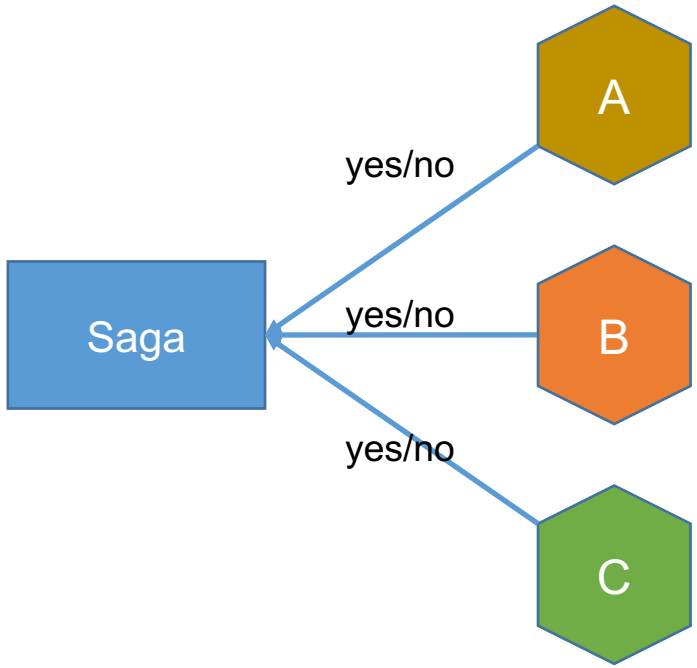
T1 T2 T3 C2 C1

异常情况

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

Saga – 最终一致


A
transact
Saga
transact
B
transact
C


A
yes/no
Saga
yes/no
B
yes/no
C

Saga - 222

- 2种恢复策略
  - 向前恢复
  - 向后恢复

2

- 2个特点
  - 和平
  - 统一

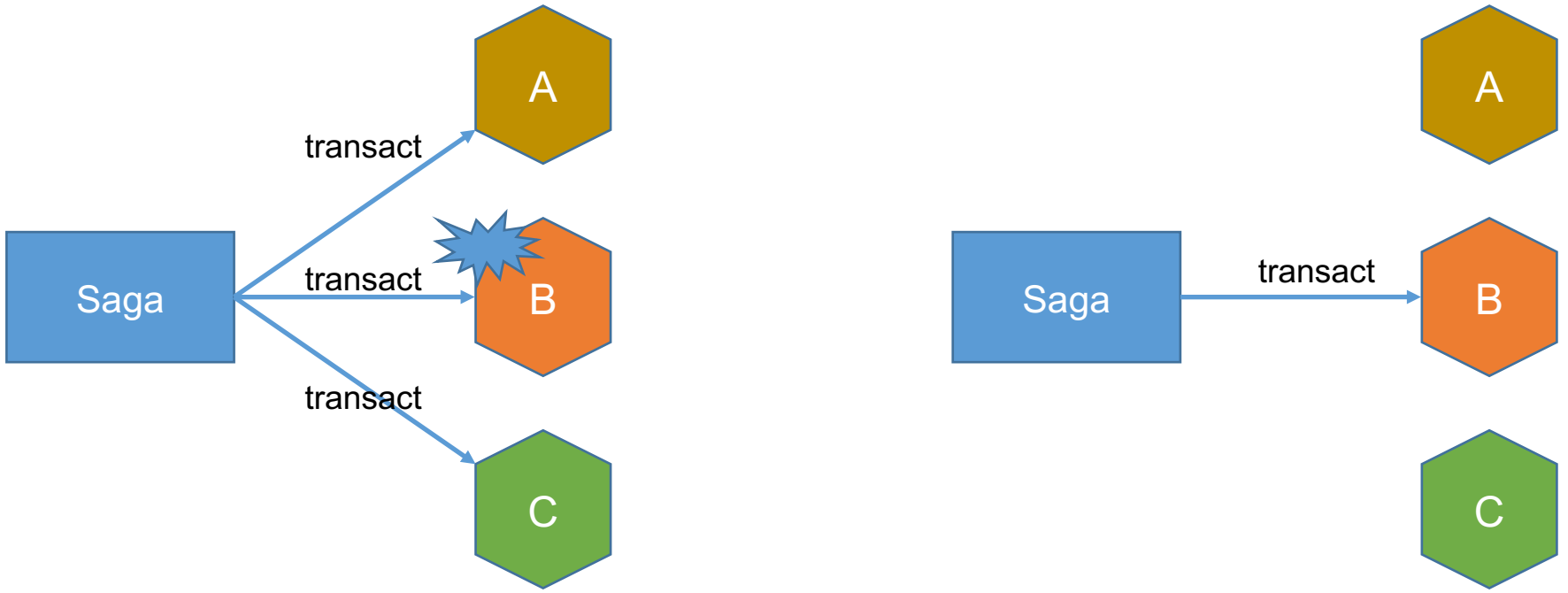
2

- 2种运行模式
  - 图遍历
  - Akka Actor

2

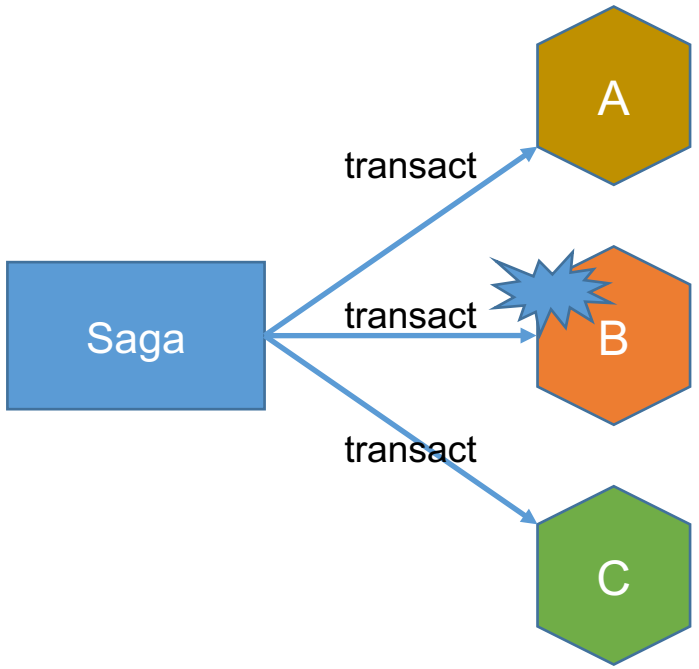
恢复策略 - 向前恢复

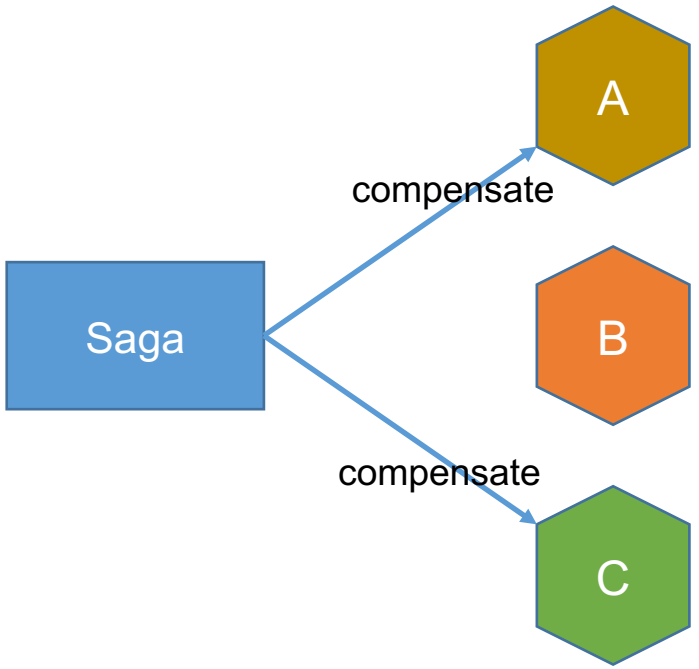
- 重试N次直到成功或采取回退措施 (人工干预)



恢复策略 -向后恢复

- 补偿


A
transact
Saga
transact
B
transact
C


A
compensate
Saga
B
compensate
C

和平统一

和平：低侵入

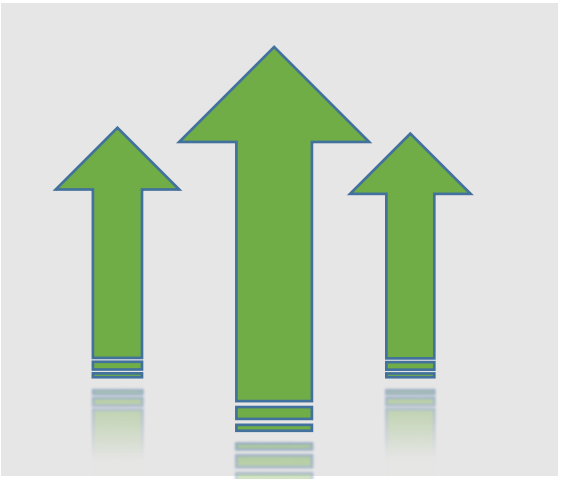
减少业务代码集成 / 运维难度

剥离业务与数据一致性复杂度

统一：集中式

让运维监控更加简单

可视化事务、调用链

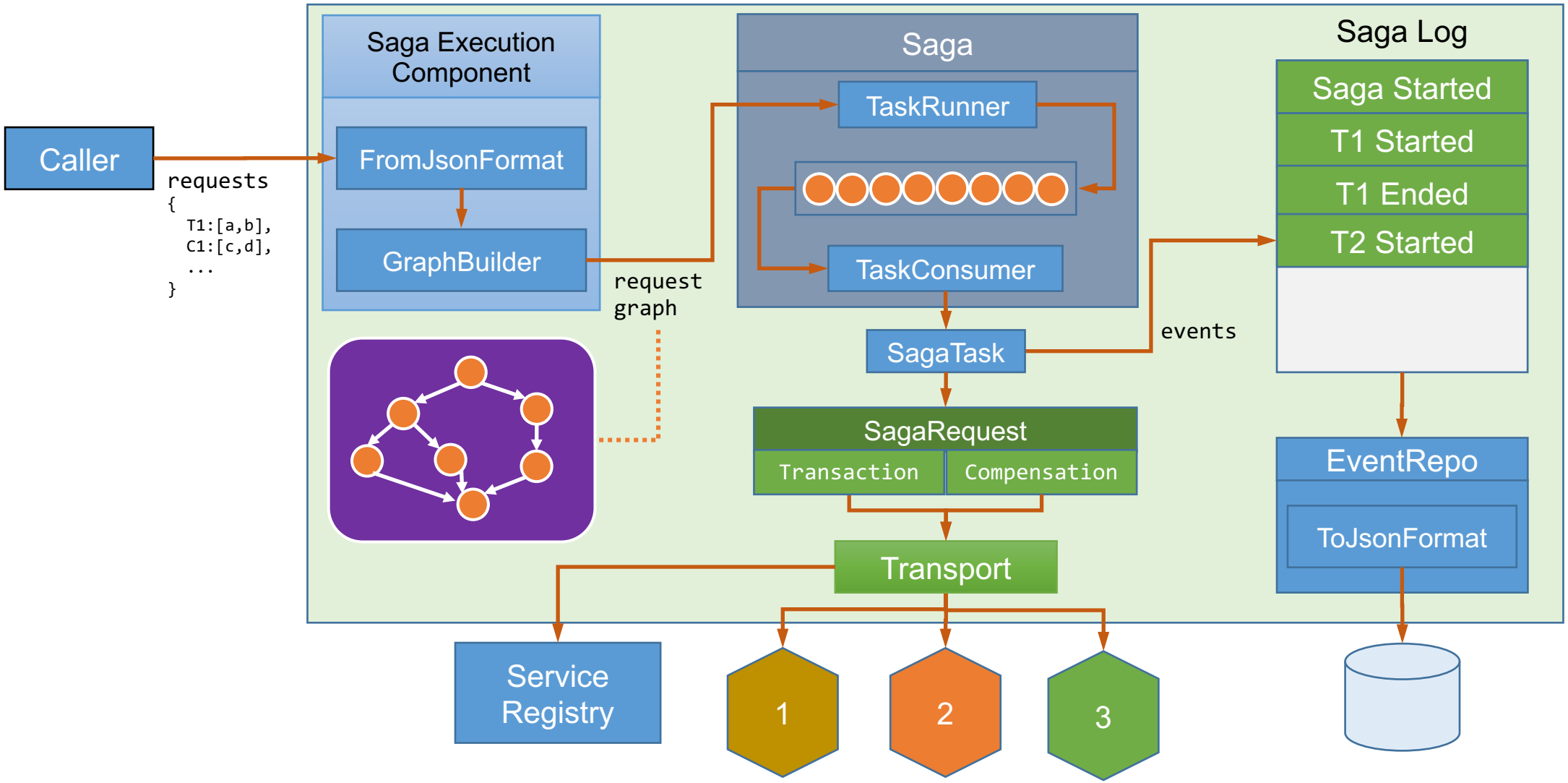


高可用

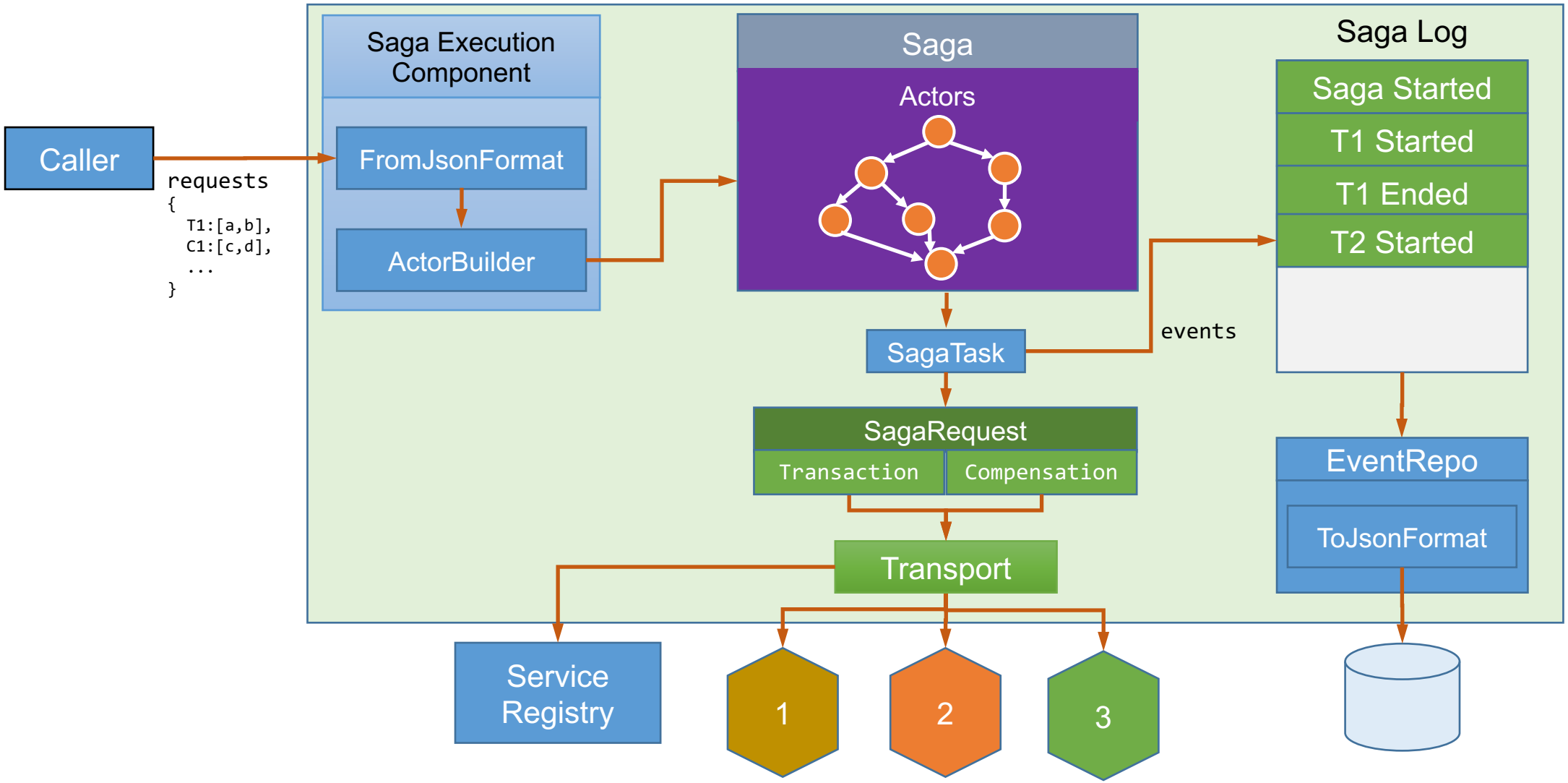
无状态、可集群、可分片

Event Sourcing架构

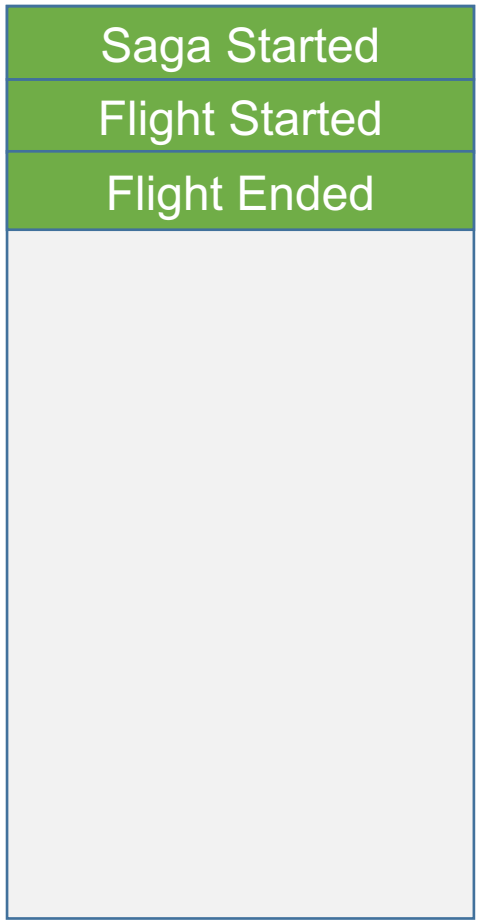
系统架构 – 基于图形

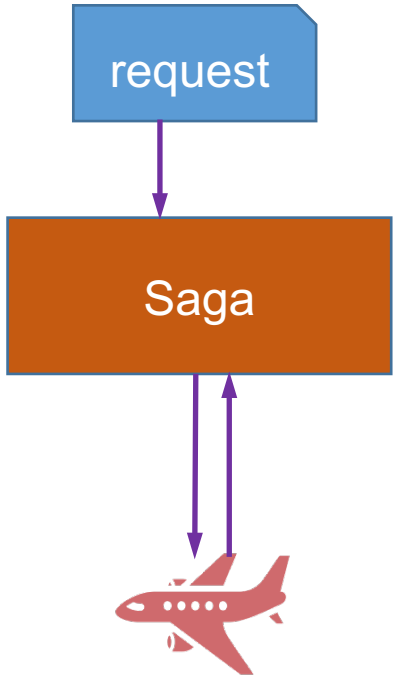

Caller
requests
{
T1:[a,b],
C1:[c,d],
...
}
Saga Execution Component
FromJsonFormat
GraphBuilder
request graph
Saga
TaskRunner
TaskConsumer
SagaTask
events
SagaRequest
Transaction
Compensation
Transport
Saga Log
Saga Started
T1 Started
T1 Ended
T2 Started
EventRepo
ToJsonFormat
Service Registry
1
2
3

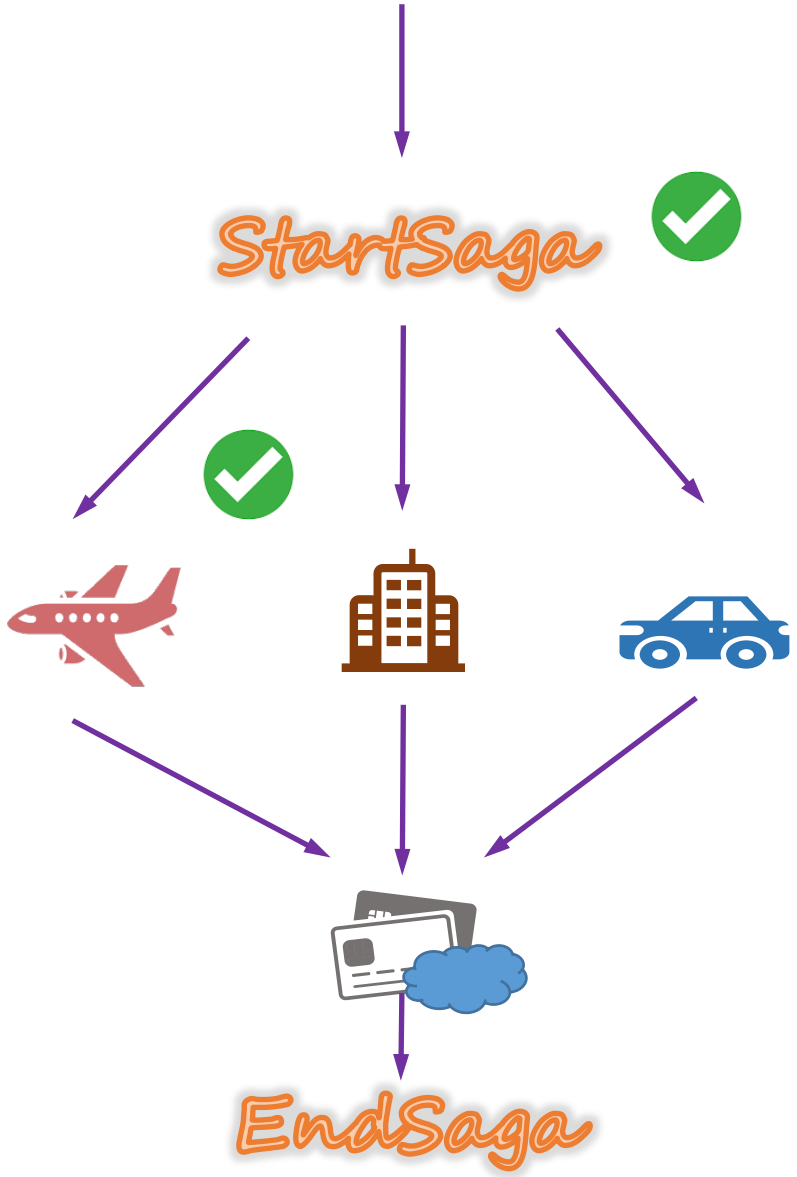
系统架构 – 基于Akka actor模型



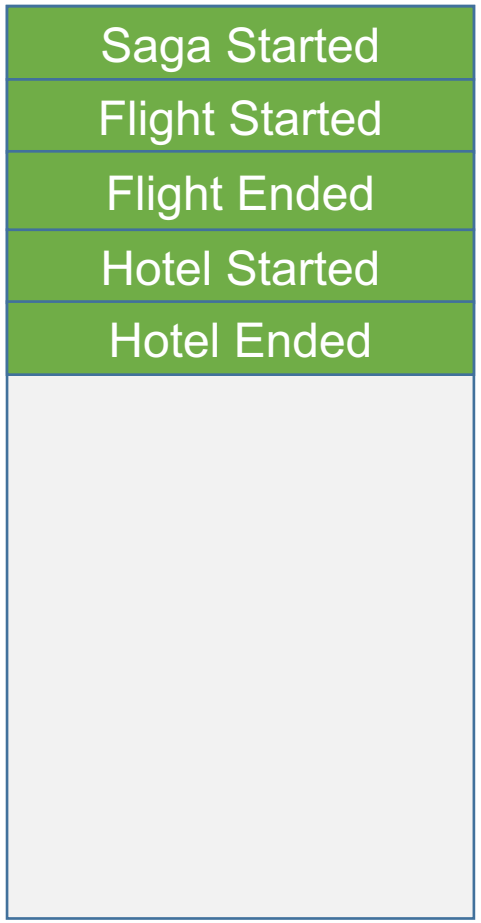
案例 – 正常情况

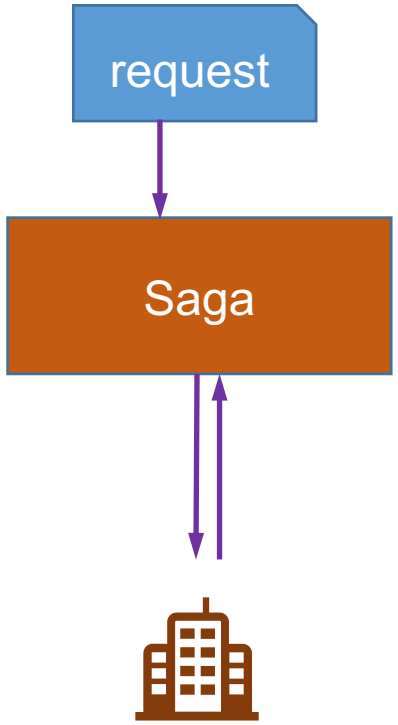

Saga Started
Flight Started
Flight Ended

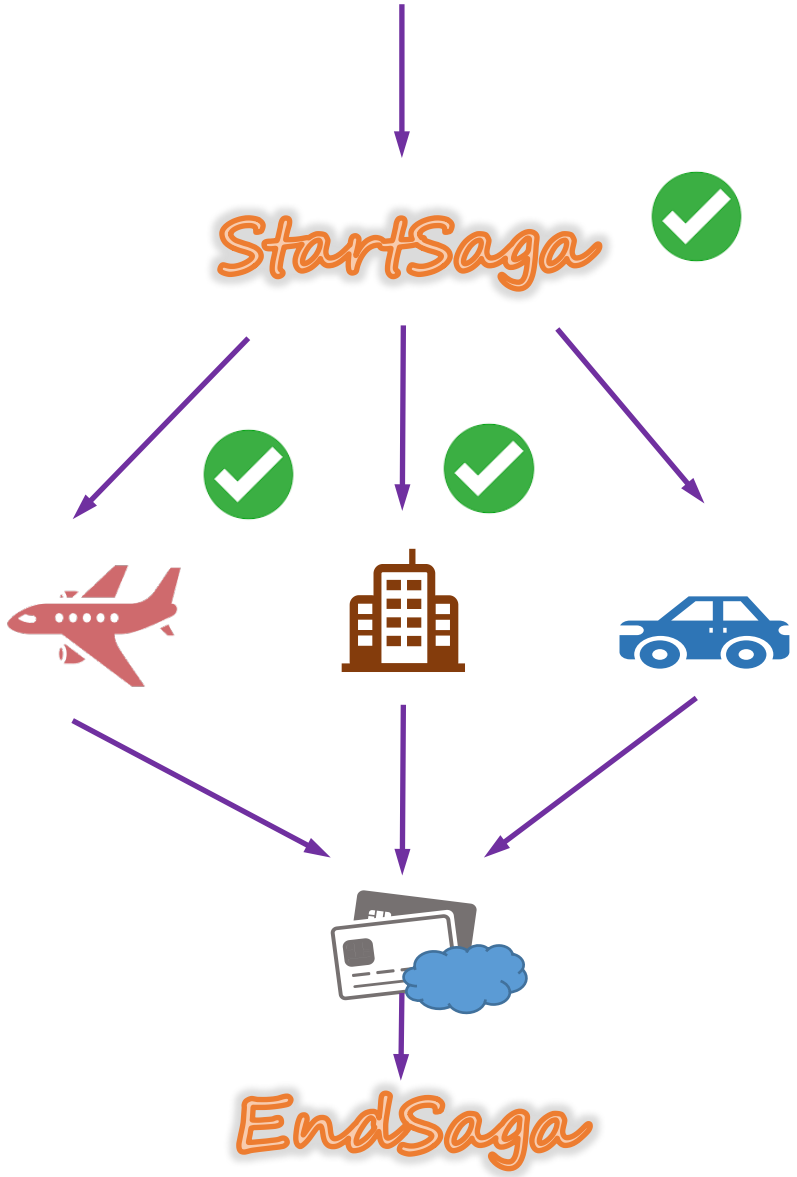

request
Saga


StartSaga
EndSaga

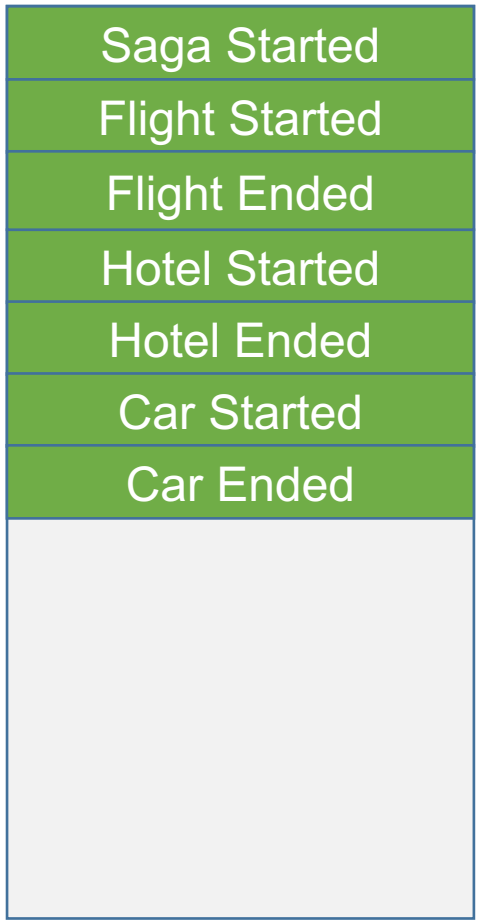
案例 – 正常情况

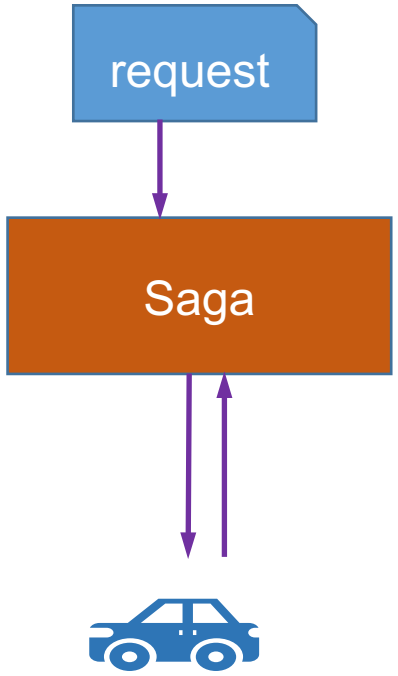

Saga Started
Flight Started
Flight Ended
Hotel Started
Hotel Ended

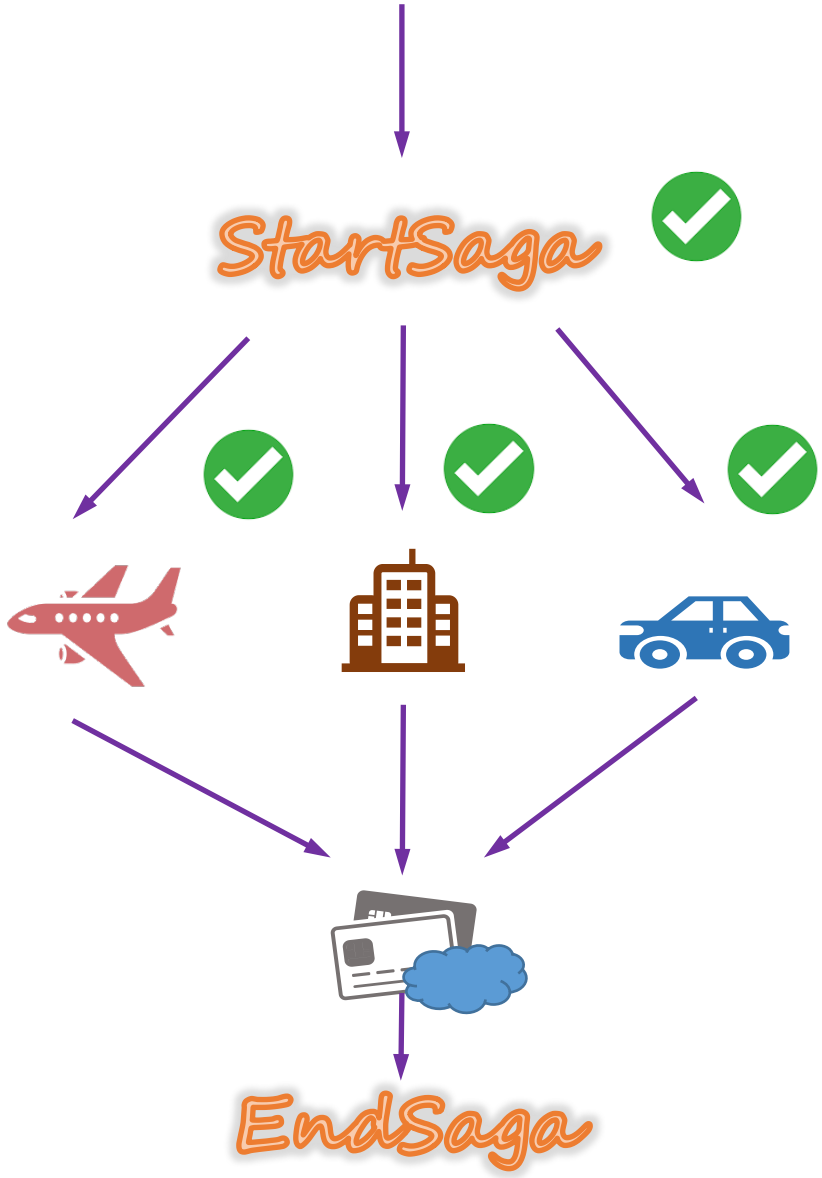

request
Saga


StartSaga
EndSaga

案例 – 正常情况

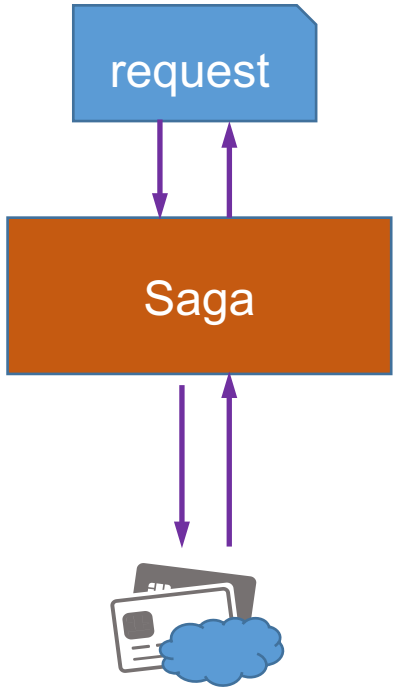

Saga Started
Flight Started
Flight Ended
Hotel Started
Hotel Ended
Car Started
Car Ended

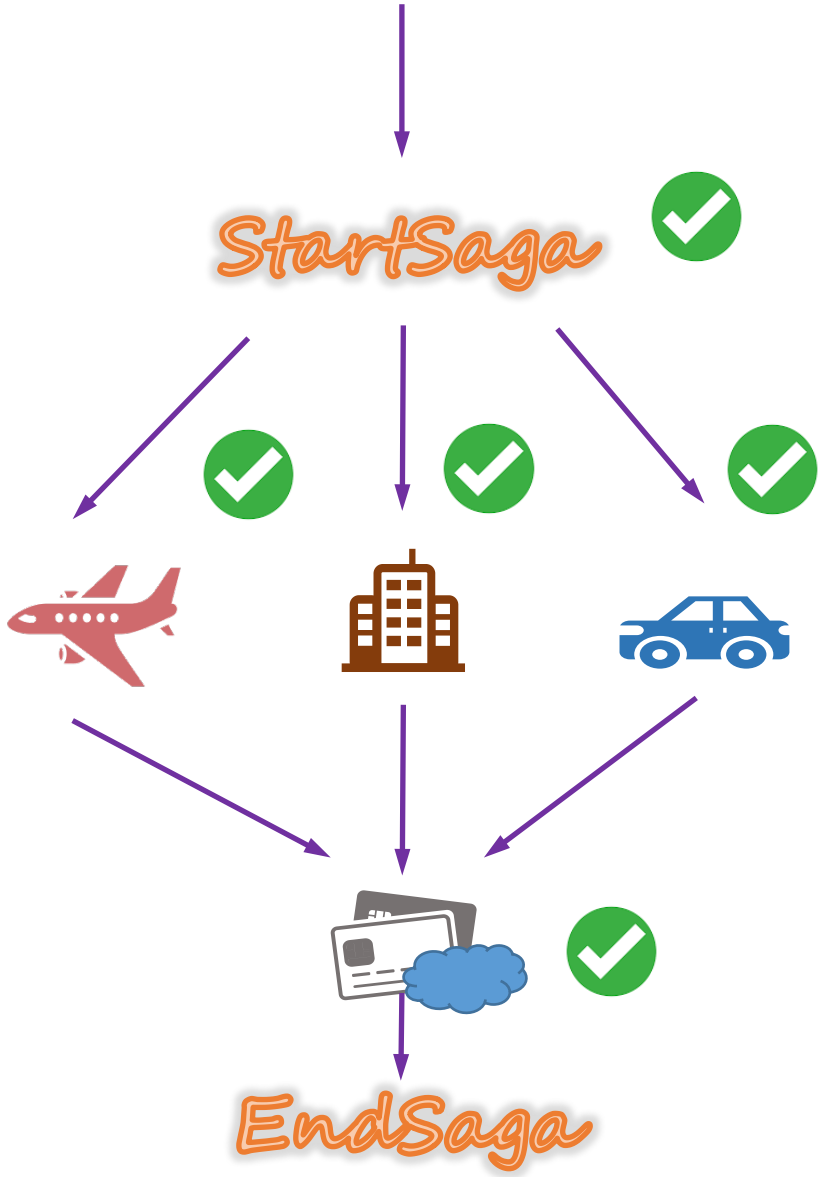

request
Saga


StartSaga
EndSaga

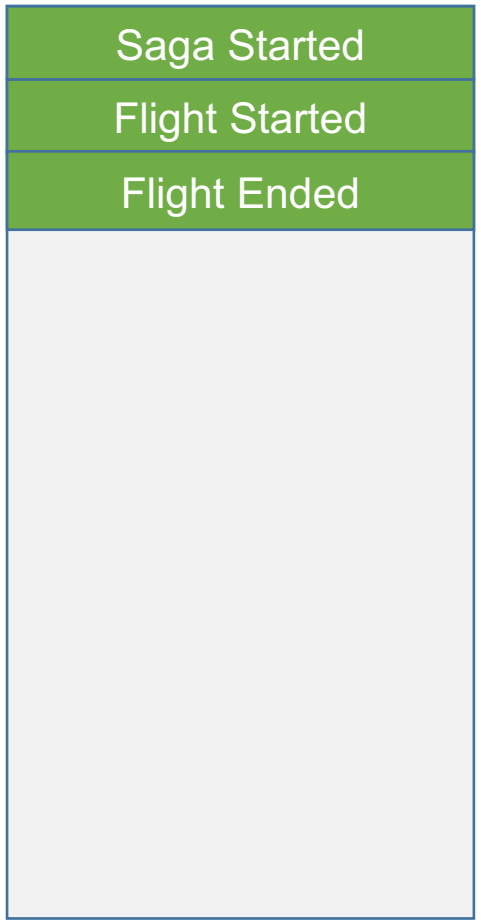
案例 – 正常情况

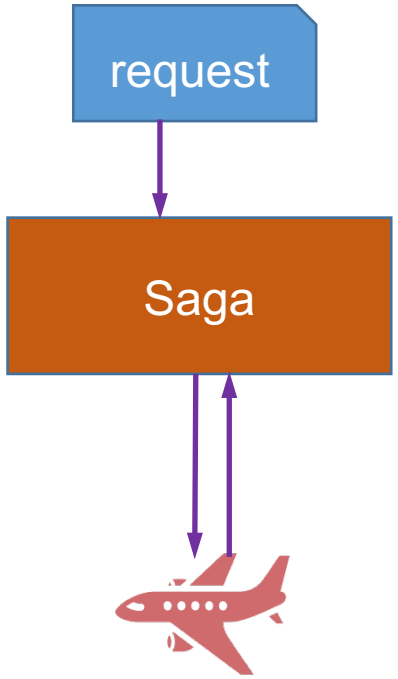
Saga Started
Flight Started
Flight Ended
Hotel Started
Hotel Ended
Car Started
Car Ended
Payment Started
Payment Ended
Saga Ended

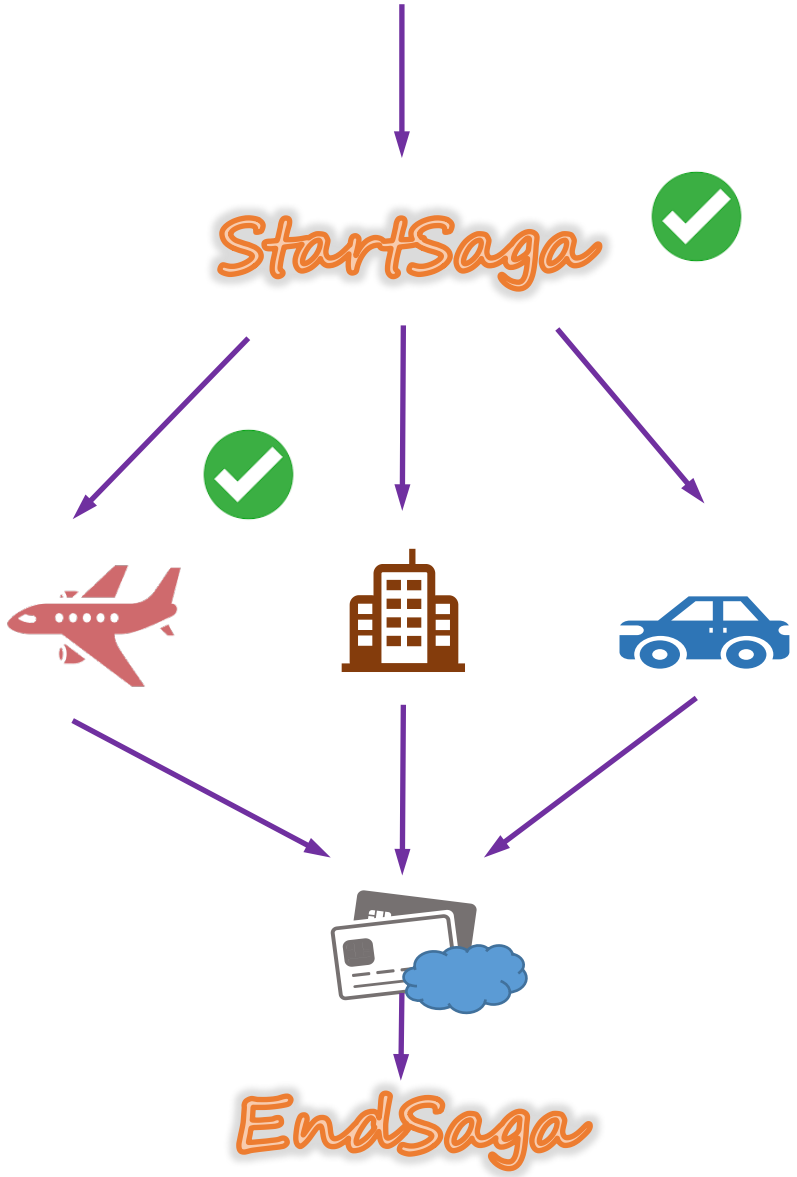




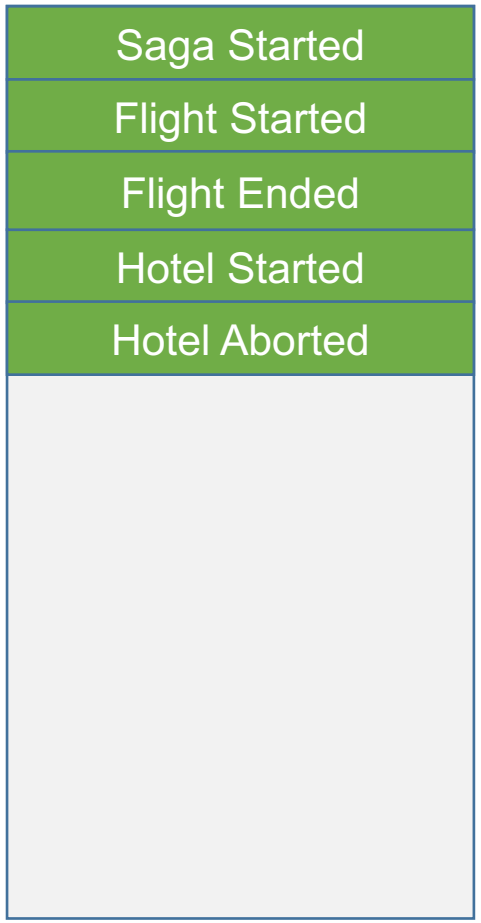
案例 – 异常情况

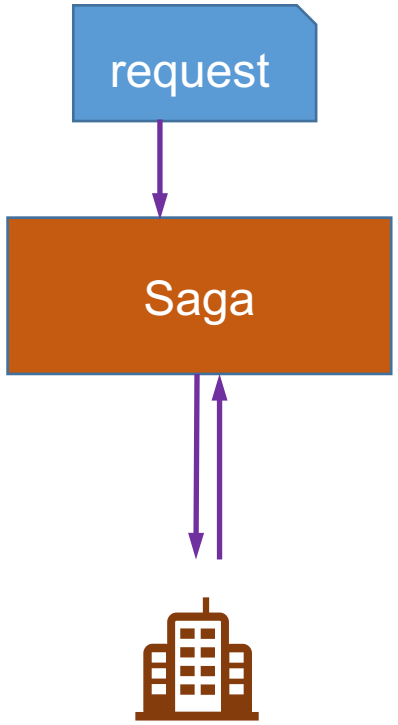

Saga Started
Flight Started
Flight Ended

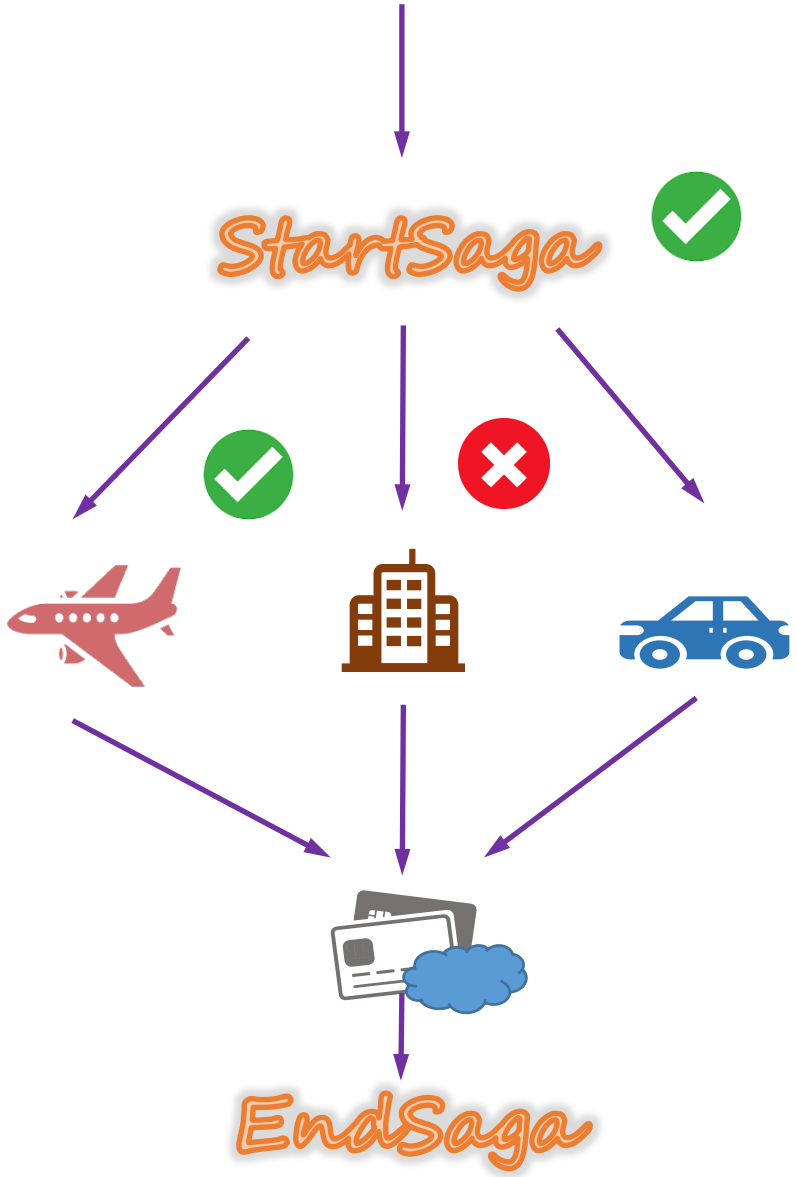

request
Saga


StartSaga
EndSaga

案例 – 异常情况

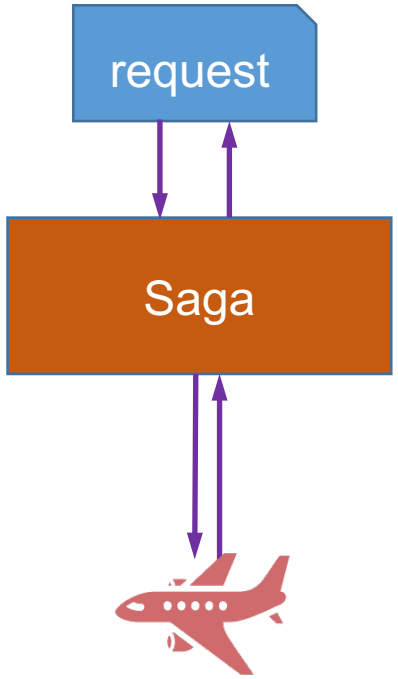

Saga Started
Flight Started
Flight Ended
Hotel Started
Hotel Aborted

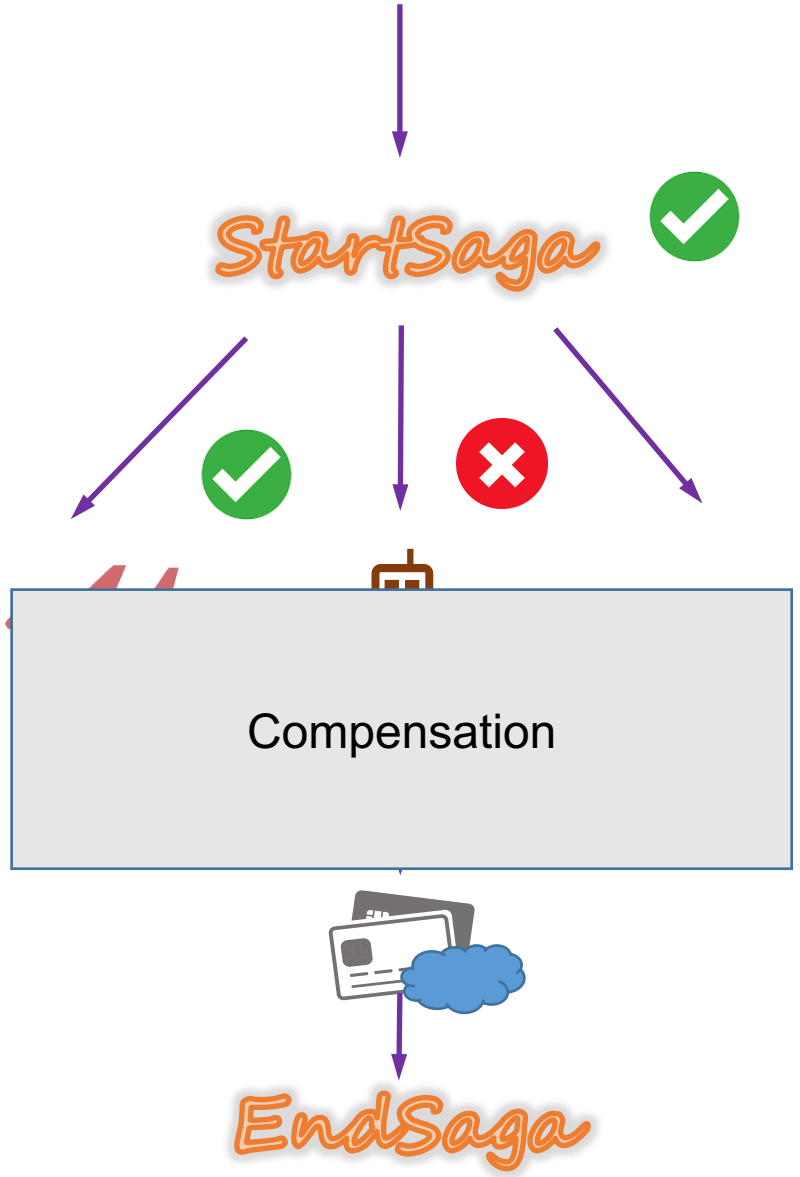

request
Saga


StartSaga
EndSaga

案例 – 异常情况

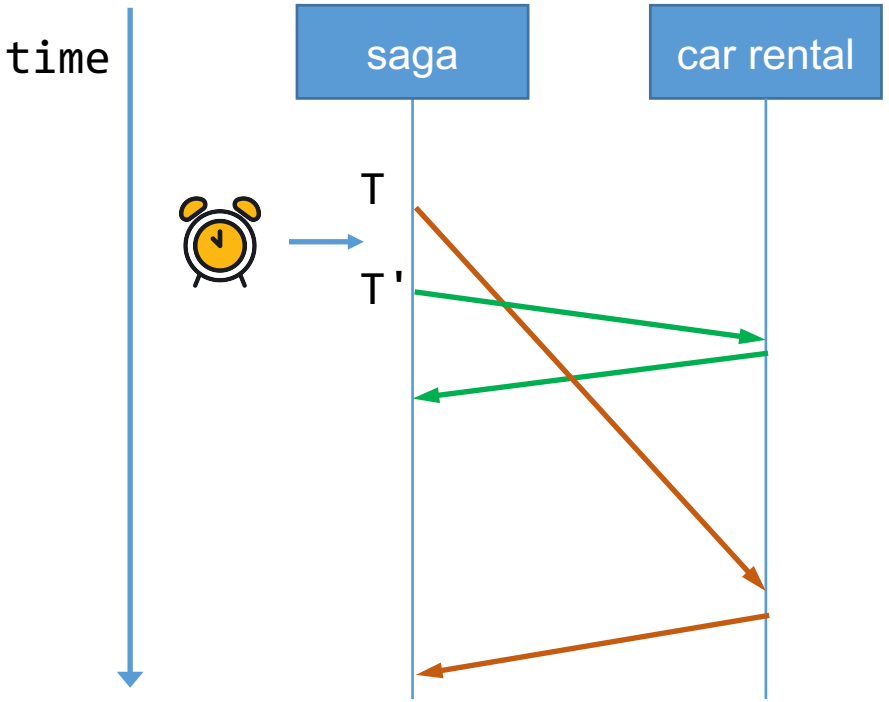
Saga Started
Flight Started
Flight Ended
Hotel Started
Hotel Aborted
Flight Compensated
Saga Ended





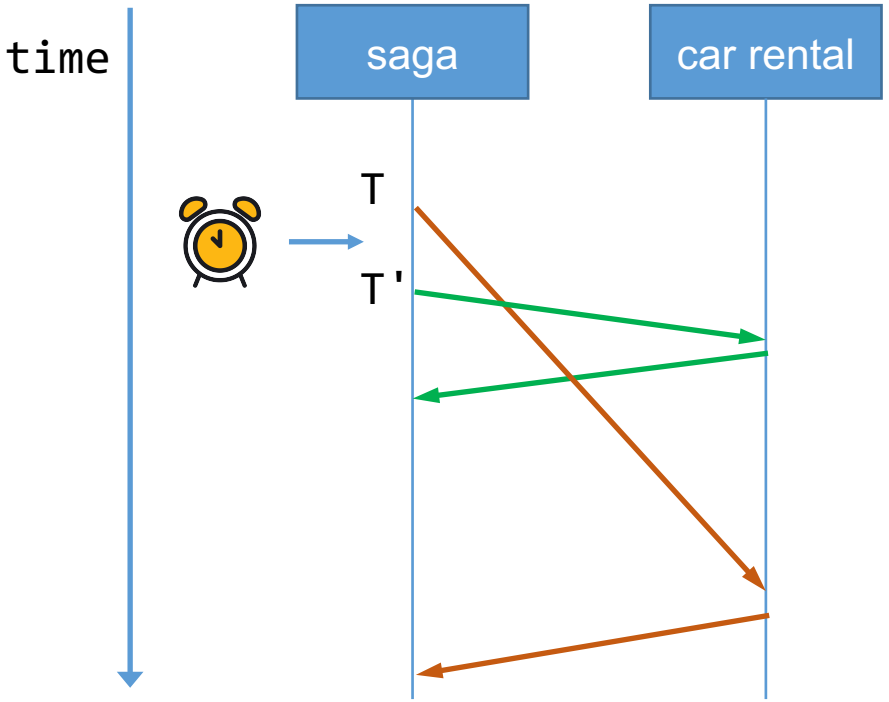
要求

- 冪等 T = T T … T

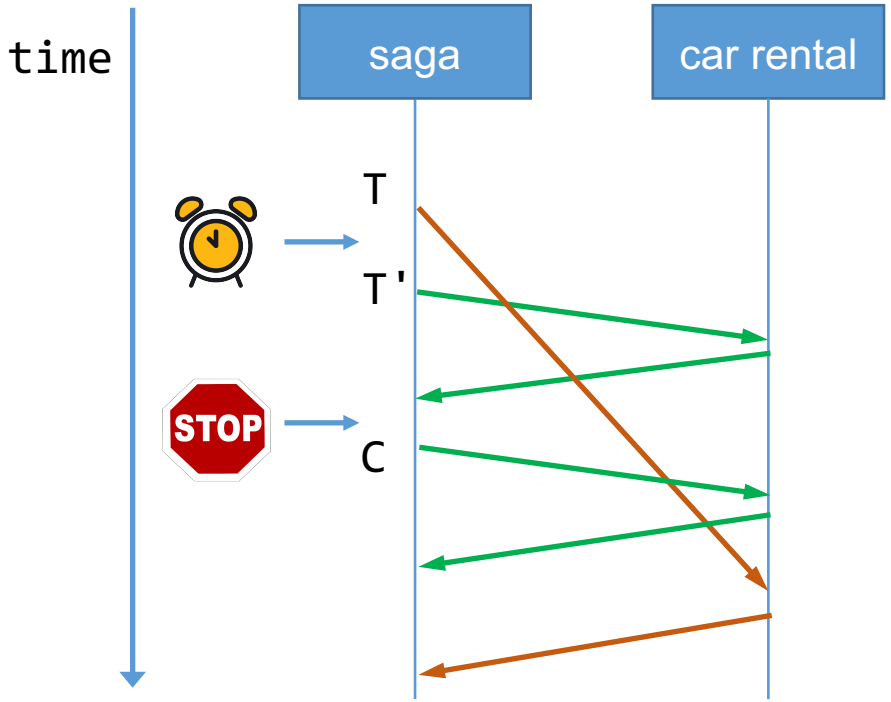


要求

- 幂等 T = T T … T



- 可交换 T C = T C T



Do NOT delete transaction records!

断

一致性方案的选择建议

一致性方案的选择建议

内刚

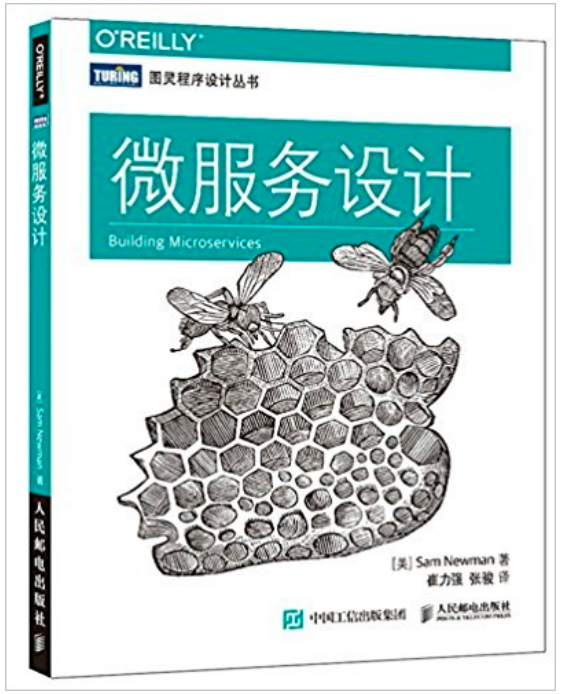
- 微服务内：聚合通过数据库事务保证强一致

外柔

- 微服务间：最终一致

微服务架构与领域驱动设计

- **if** our **service boundaries align to the bounded contexts** in our domain, and our microservices represent those bounded contexts, we are off to an excellent start in ensuring that our **microservices are loosely coupled and strongly cohesive**.



领域驱动设计是微服务系统架构的最佳指南

微服务：限界上下文

聚合与数据一致性

聚合边界：强一致边界

RULE: MODEL TRUE INVARIANTS IN CONSISTENCY BOUNDARIES

- *A properly designed **Aggregate** is one that can be modified in any way required by the business with its invariants **completely consistent within a single transaction***

聚合内：强一致

RULE: USE EVENTUAL CONSISTENCY OUTSIDE THE BOUNDARY

- ***Any rule** that **spans AGGREGATES** will **not** be expected **to be up-to-date at all times**.*

跨聚合：最终一致

一致性方案选择建议

- 微服务：限界上下文
- 聚合边界：强一致边界
- 限界上下文 -> 1 .. N 聚合

- **内刚** 微服务内：聚合通过数据库事务保证强一致
- **外柔** 微服务间：最终一致

如果需要分布式强一致，先考虑**设计**是否合理而非追求最新**技术**
合理的设计能大大减少技术复杂度和商业成本

总结

- 起因：离
- 方案：Saga 222
- 选择建议：内刚 外柔

未来的开发计划

- 更易使用的数据一致性方案
- 集成调用链追踪 (Zipkin)，定位性能瓶颈
- 可视化事务拓扑，定位异常最多的服务
- 集成熔断功能 (Hystrix)
- 实现基于消息队列的通信模式
- ……

- https://issues.apache.org/jira/projects/SC

谢谢

http://servicecomb.io

https://github.com/ServiceComb