构建多公有云系统部署实践

上海曙安数据服务有限公司

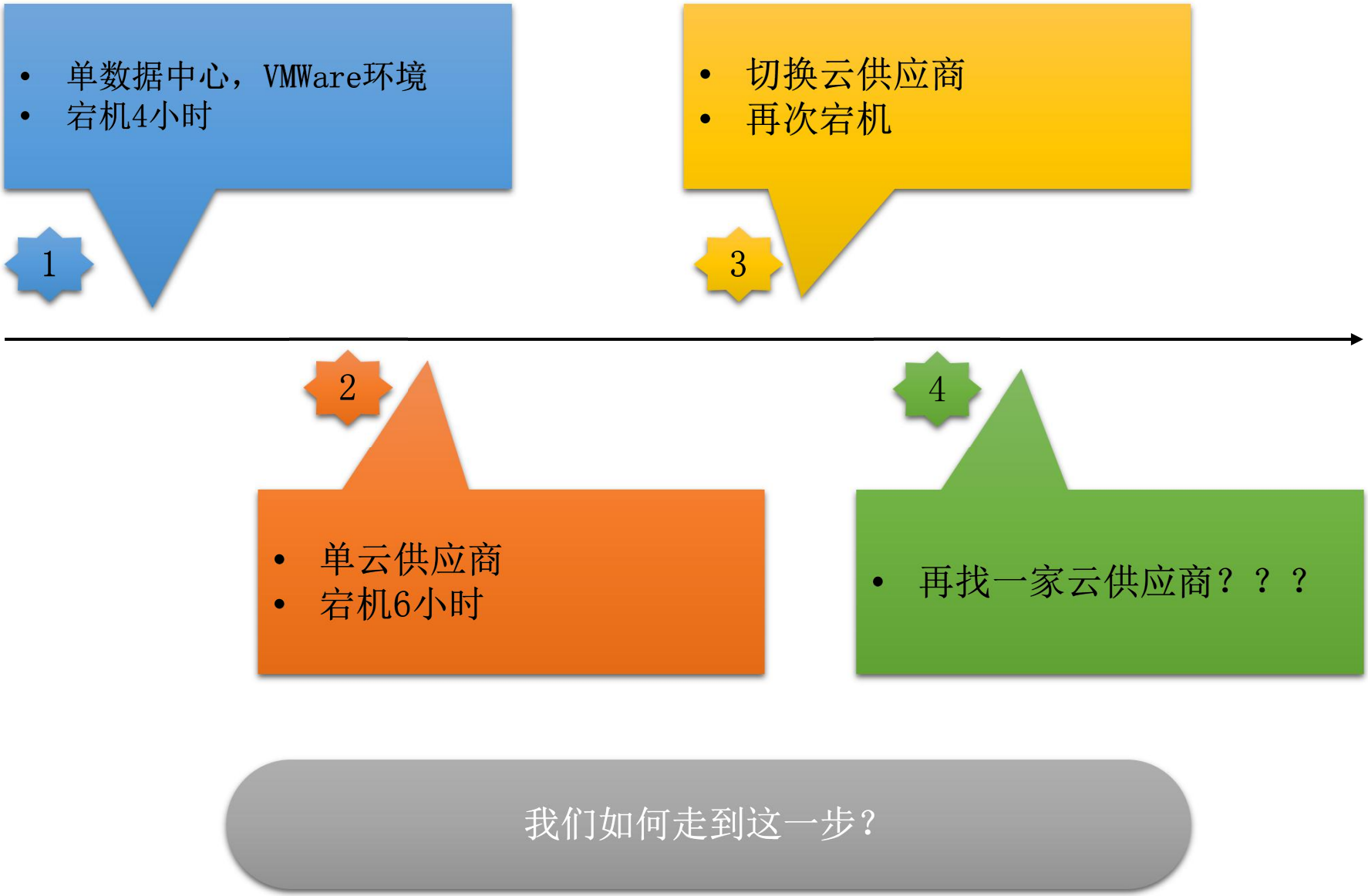
叶向宇

关于我

- 过去：
  - 联想集团服务器事业部
  - 微软中国技术中心
  - 惠普中国技术中心
  - 微软云计算事业部
- 现在：
  - 上海曙安数据服务有限公司
  - VC3多云管理平台架构师

今天的话题

- 我们的目标是什么？
- 实现业务目标过程中遇到了什么问题？
- 我们是如何思考的？
- 我们是如何实践的？
- 我们下一步的计划是什么？


• 单数据中心，VMWare环境
• 宕机4小时
1
• 切换云供应商
• 再次宕机
3
2
• 单云供应商
• 宕机6小时
4
• 再找一家云供应商？？？
我们如何走到这一步？

从宕机中学到的几件事（1）

SLA <99.95% → $$

- 供应商SLA不是保证不宕机，而是索赔的依据

从宕机中学到的几件事（2）

- 小范围宕机几乎不可避免

从宕机中学到的几件事（3）

Region	Status	30 Day Availability	*1 block = 1 mins*	Outages	Downtime
ams2	↑	86.7087%		1	95.7 hours
de-germany	↑	93.8392%		2	44.36 hours
us-virginia	↑	95.6044%		1	31.65 hours
perth	↑	98.5208%		4	10.65 hours
paris	↑	99.7083%		1	2.1 hours
frankfurt	↑	99.8611%		2	60 mins
us-nevada	↓	99.877%		2	53.15 mins
ZRH	↑	99.882%		1	50.98 mins
UC1	↑	99.8954%		2	45.2 mins
sweden-south	↑	99.9126%		4	37.75 mins

https://cloudharmony.com/status-of-compute

- 大范围宕机发生可能性依然存在

“不宕机是核心需求”


awendemo@awendemo-pc: ~/桌面/Server/solution
awendemo@awendemo-pc:~/桌面/Server/solution$ ./bin/application
佛祖保佑 永不宕机
心外无法 法外无心

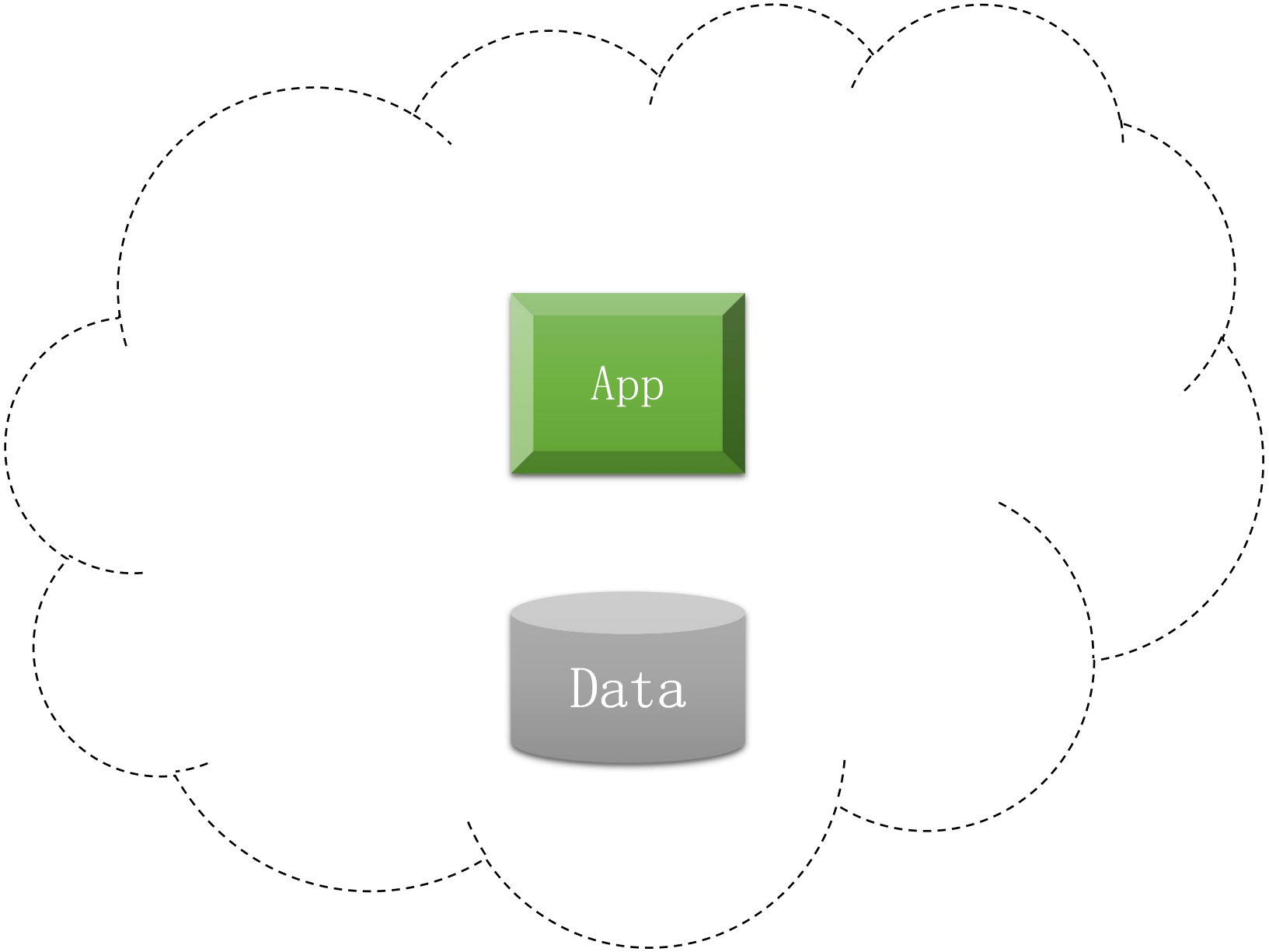

@聚美陈欧
weibo.com/leochenou

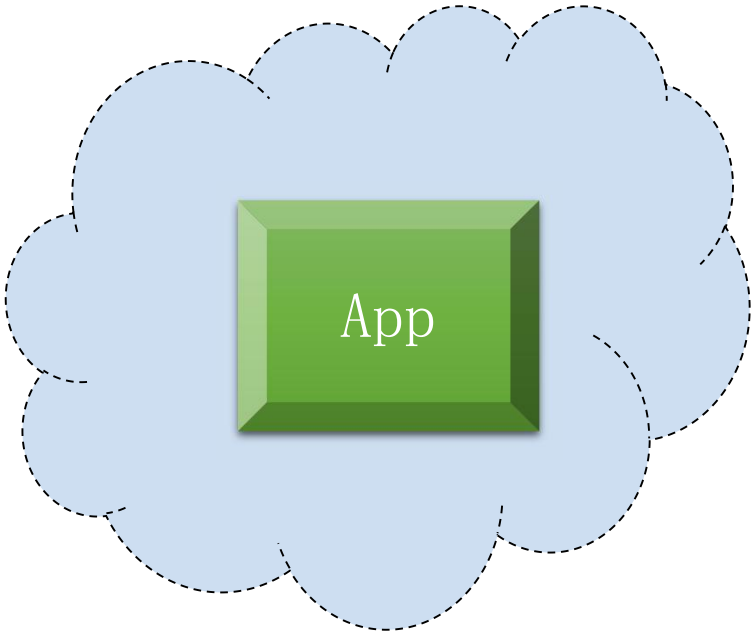
“如果一朵云宕机不可避免

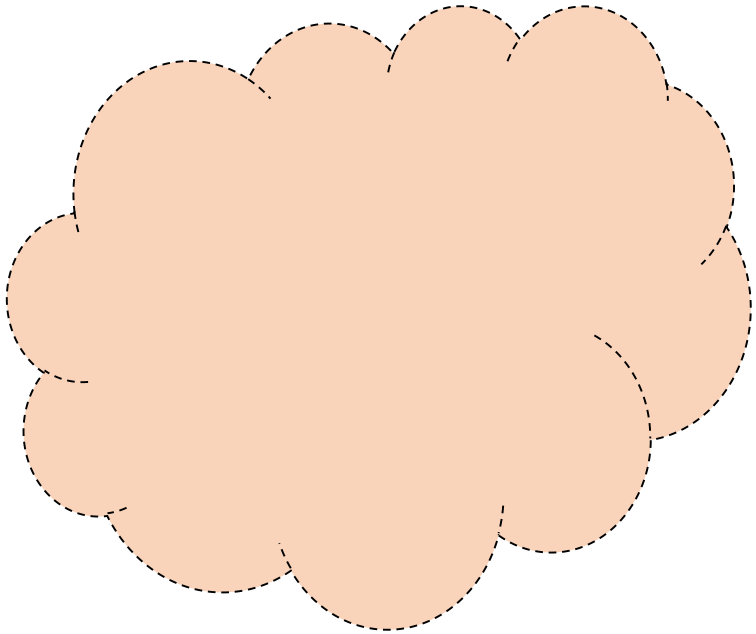
那就把应用部署到多个云上”

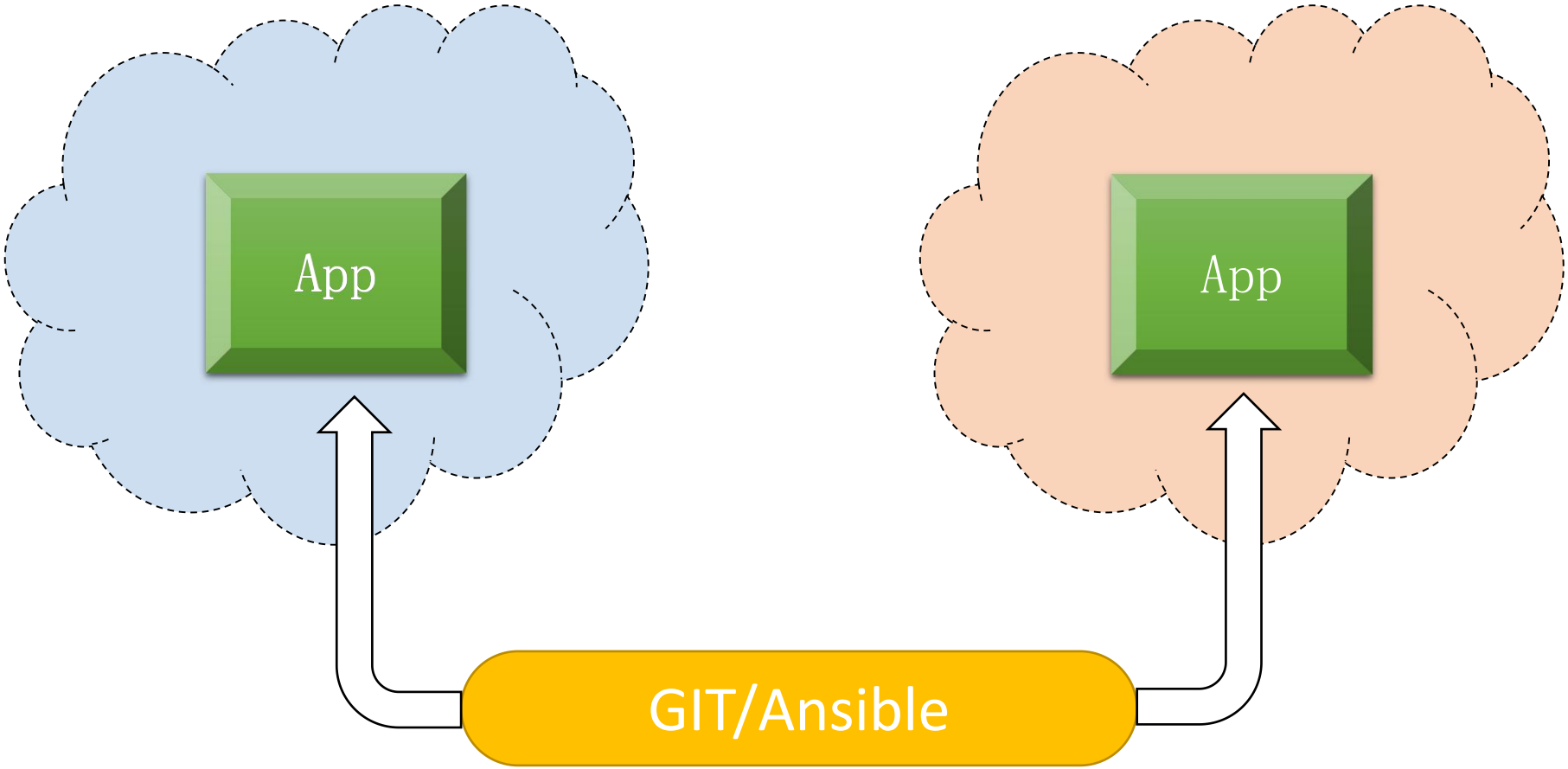
目标

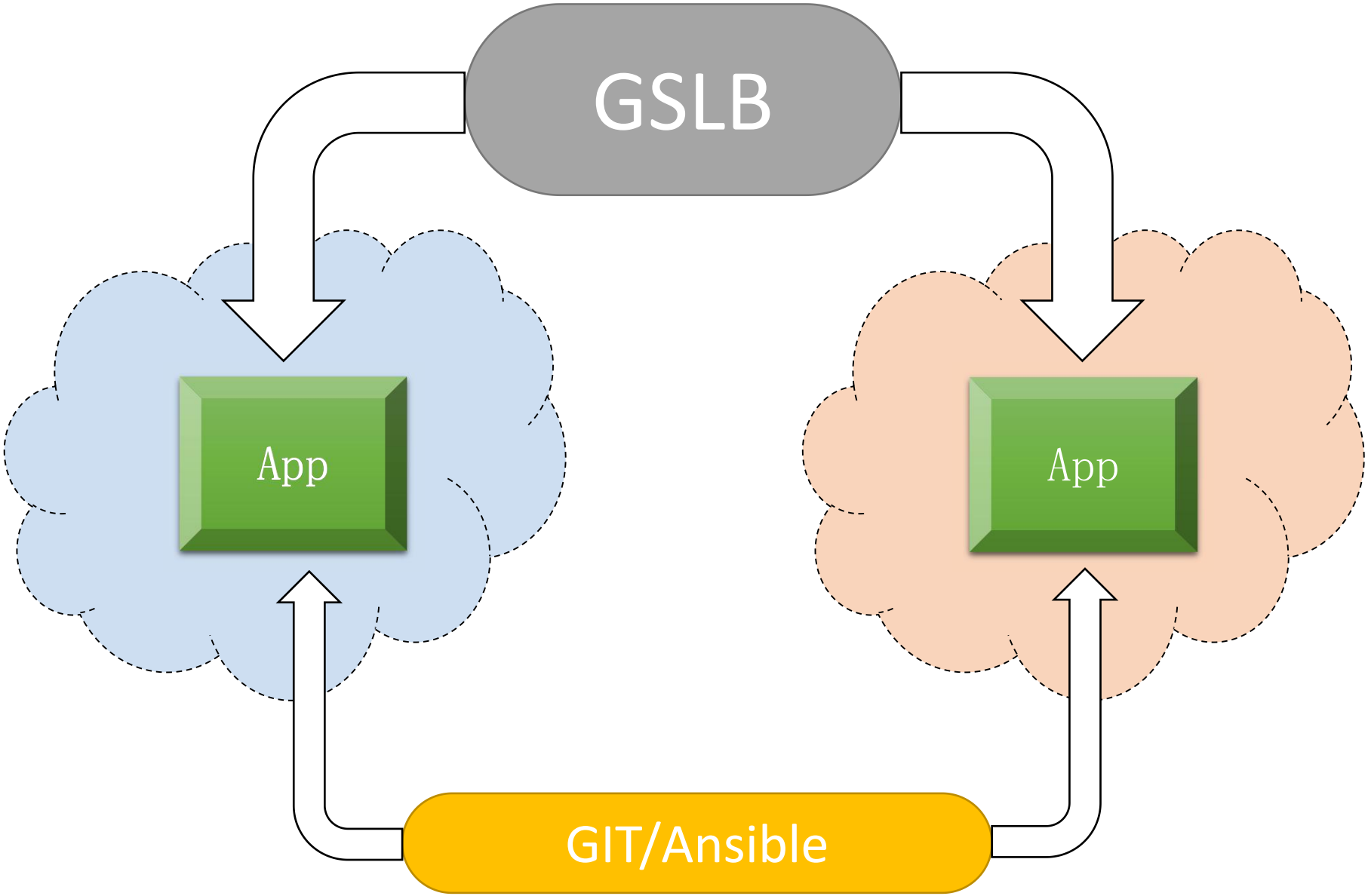
- 多数据中心多活
- 节省成本、可负担的解决方案
- 在灾难发生的过程中，如果无法达到完全可用，则至少应该保证部分可用：
  - 部分业务功能可用
  - 部分客户业务可用
  - 部分数据可用
- 尽量少的人工干预

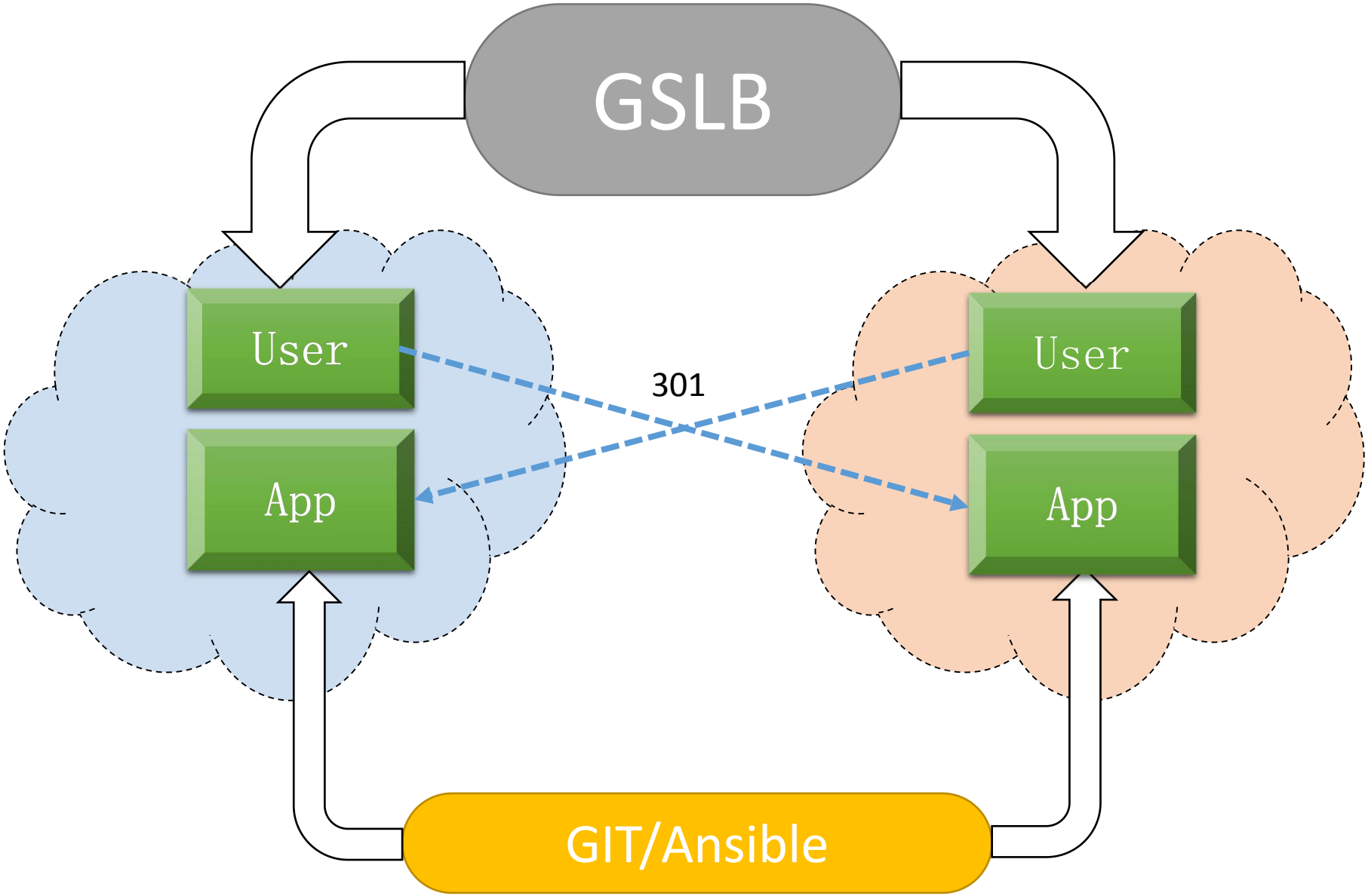

App
Data

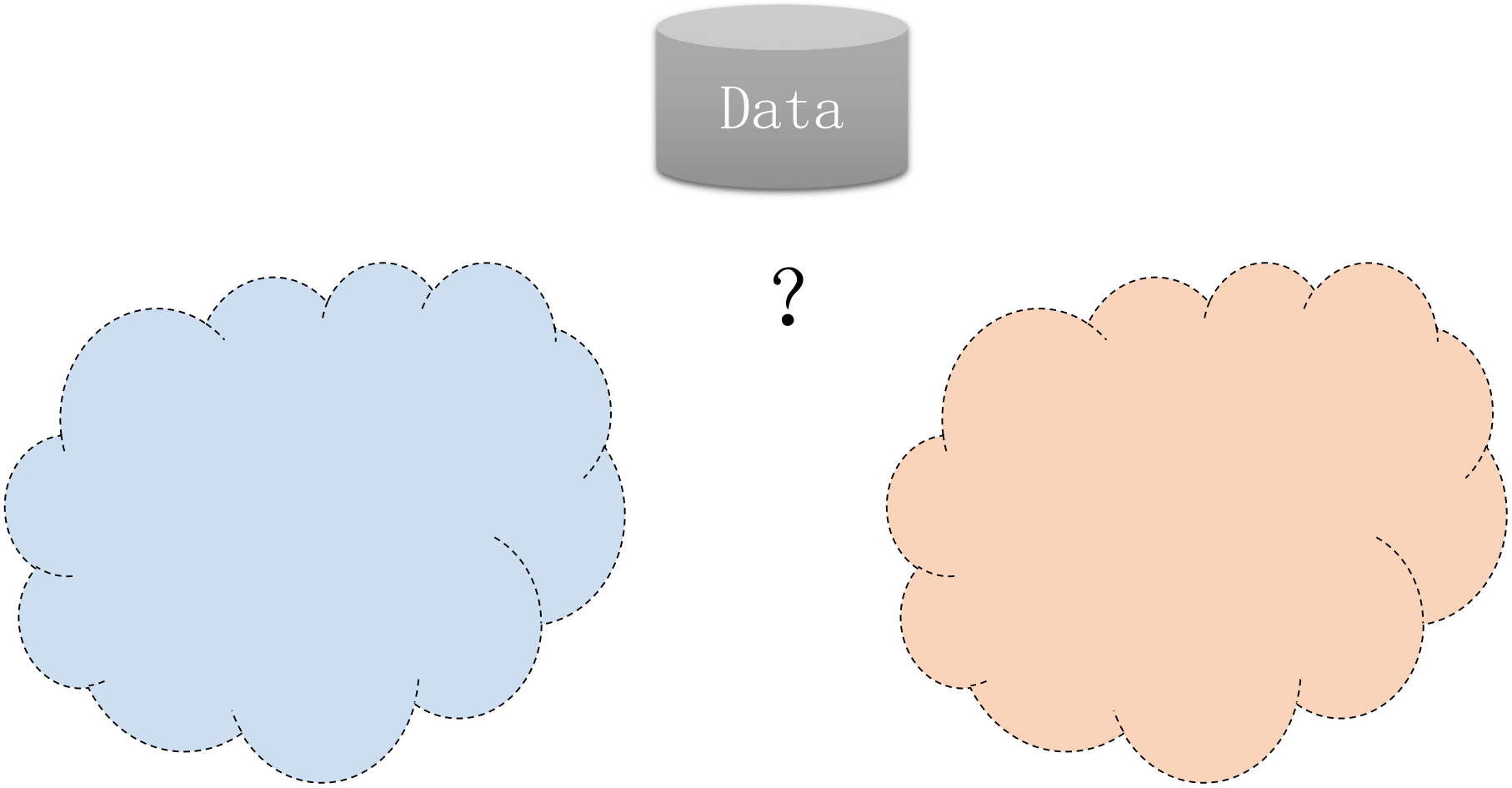

App




App
App
GIT/Ansible

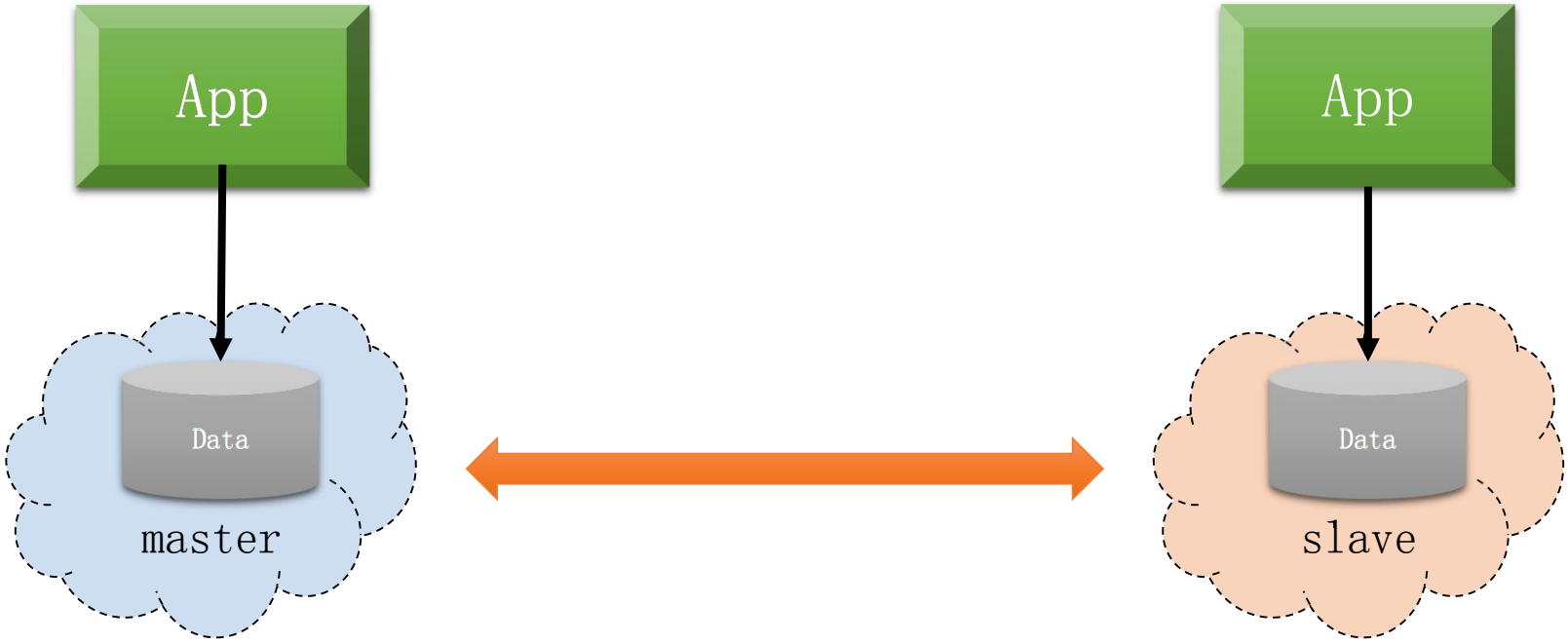

GSLB
App
App
GIT/Ansible


GSLB
User
App
301
User
App
GIT/Ansible

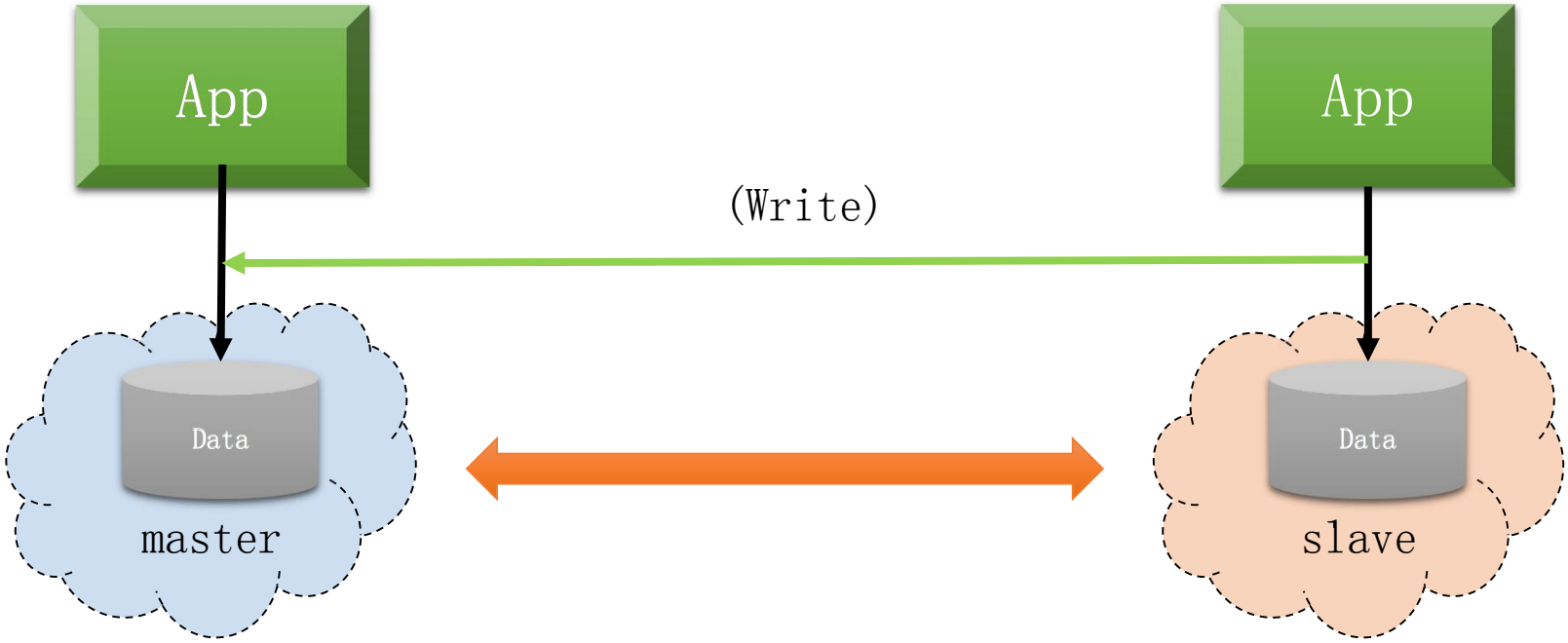

Data
?

Master / Slave

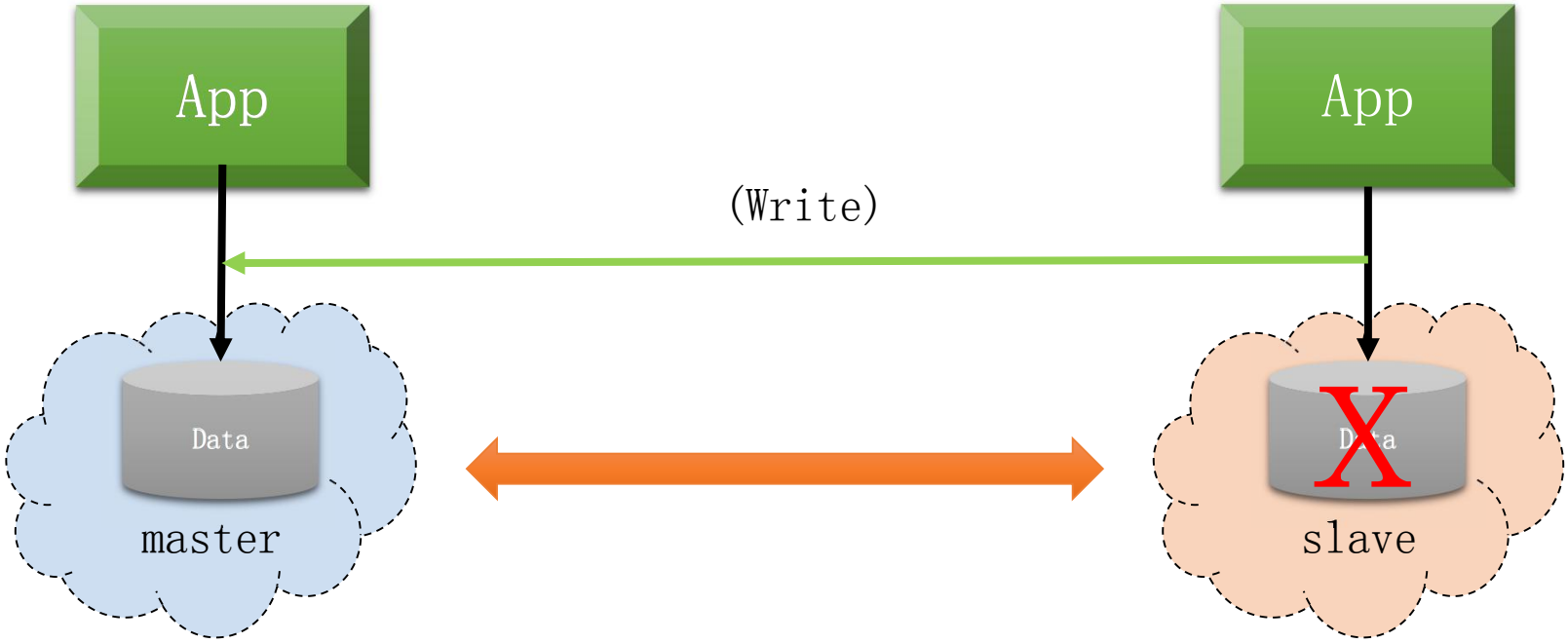
Master / Slave



Master / Slave

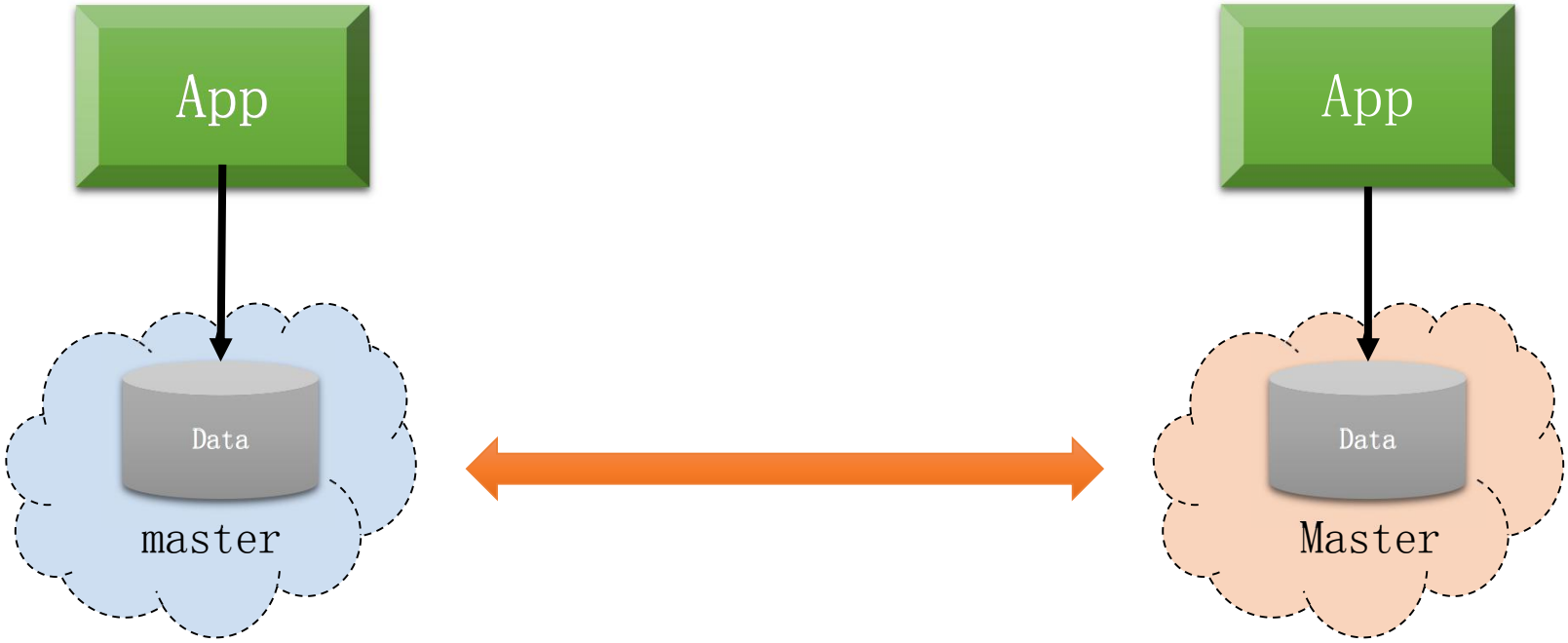


Master / Slave



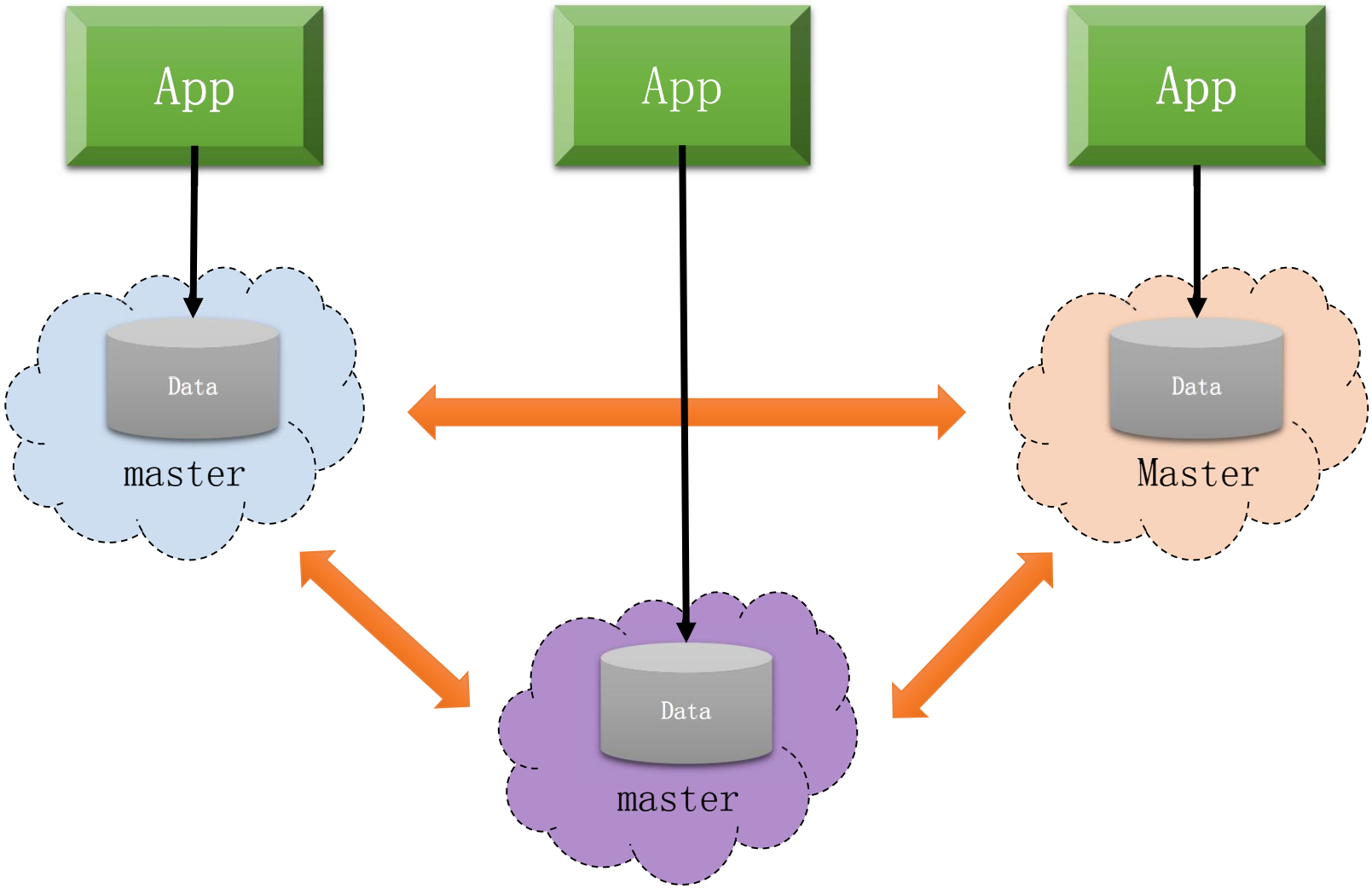
- 如果设计目标是随时保持2分数据拷贝，那么slave宕机的情况下，master应不能写入

Master / Master

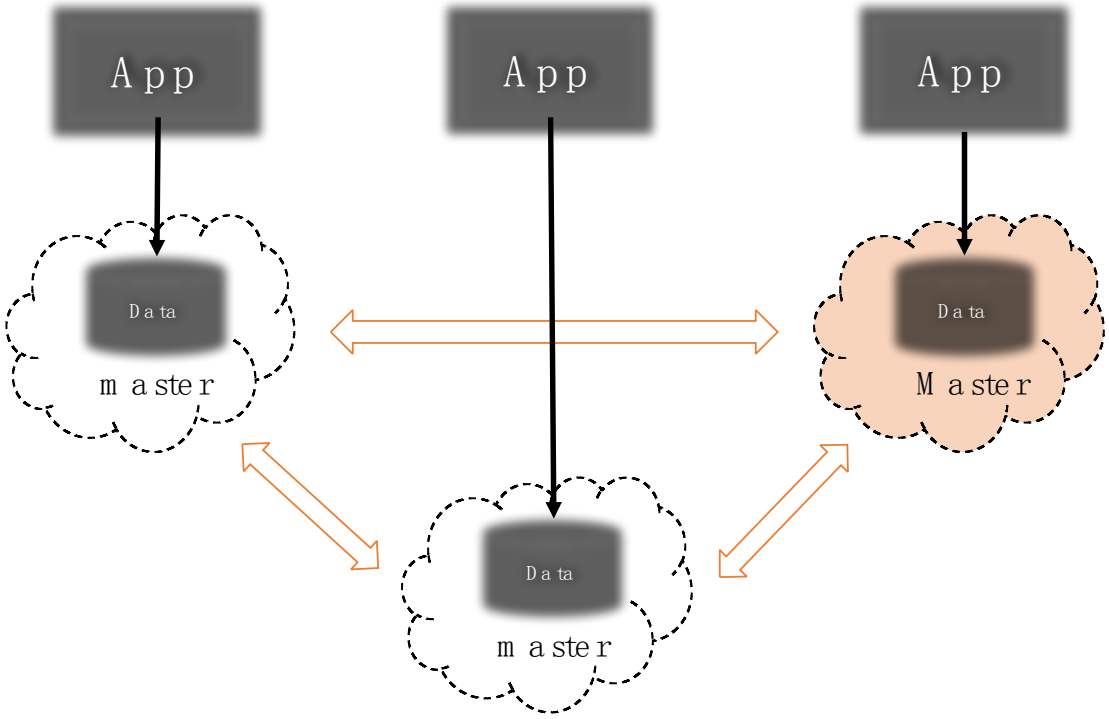


- 需要假定网络可靠（拜占庭将军问题）
- Master越多越慢，代价越高，不可扩展
- 适合单数据中心内部，可以用来解决**局部故障**
- 跨数据中心则可能需要**投入专线**

Master / Master / Master



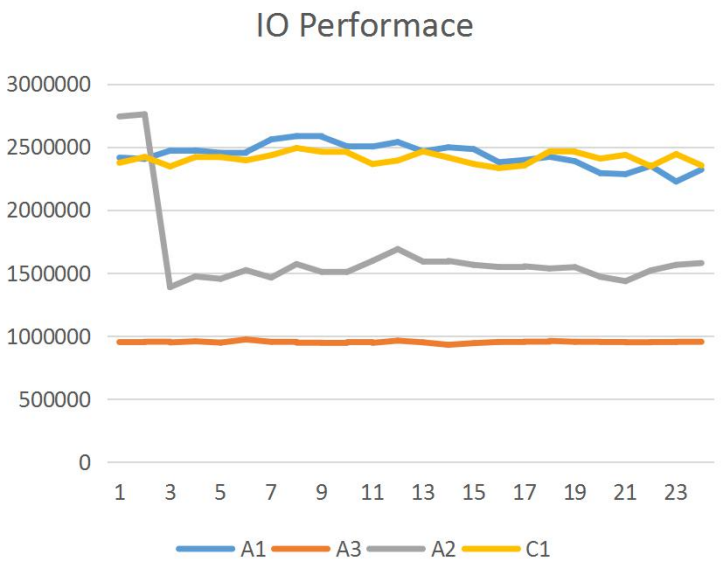
Master / Master / Master

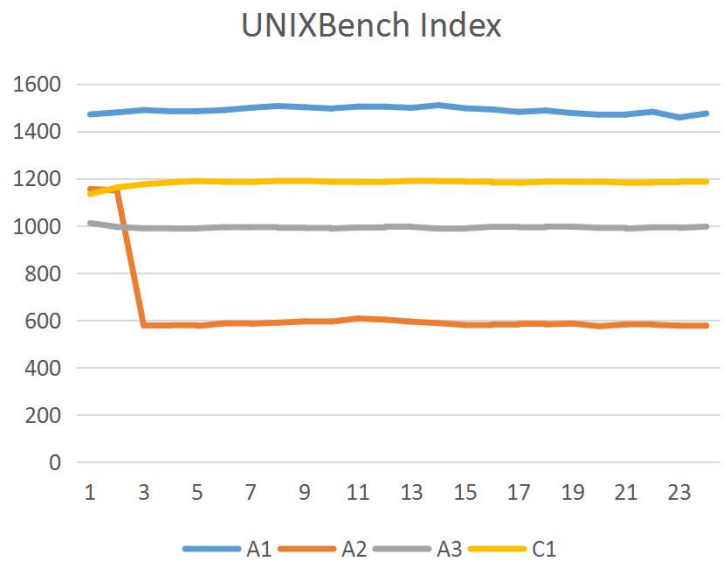


1. 允许任意一个站点宕机、断网的情况下保证依然有2个可用站点
2. 可以形成2:1的多数派解决数据不一致的问题
3. 超高的可用性

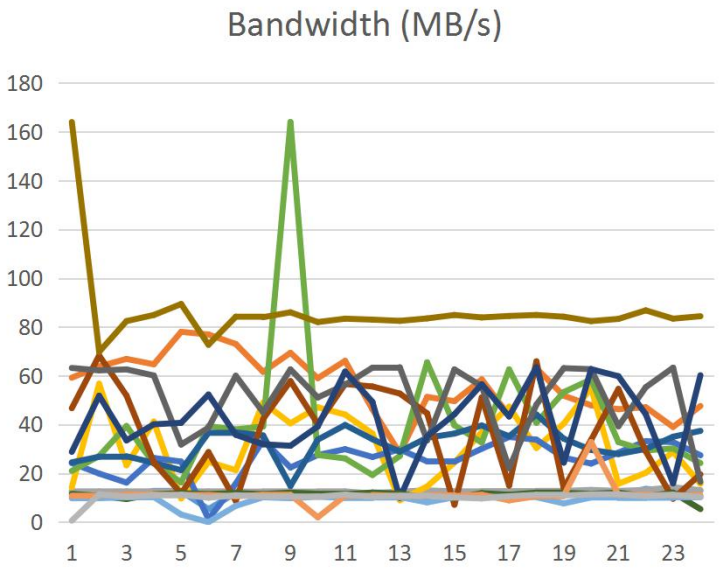
实战之：找到合适的云

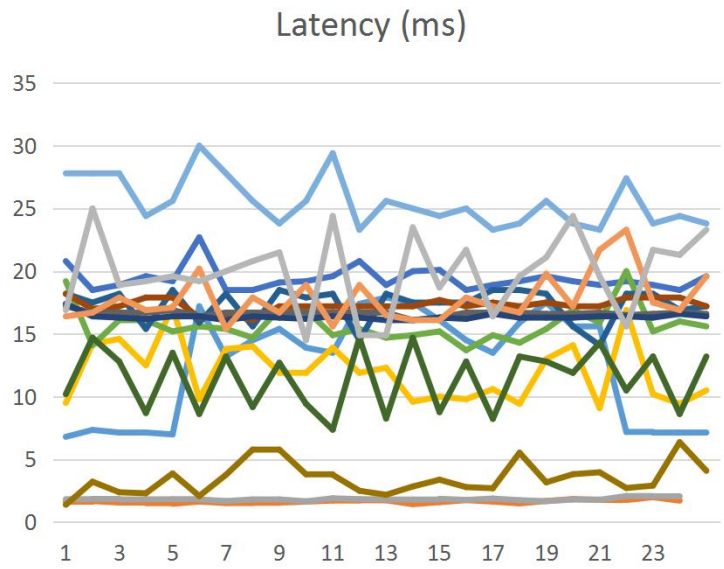
不同云的性能


IO Performace
3000000
2500000
2000000
1500000
1000000
500000
0
1 3 5 7 9 11 13 15 17 19 21 23
A1 A3 A2 C1

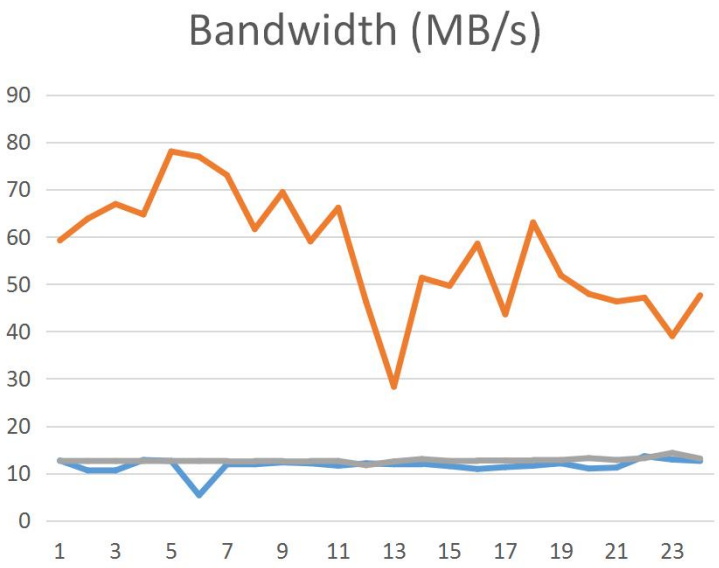

UNIXBench Index
1600
1400
1200
1000
800
600
400
200
0
1 3 5 7 9 11 13 15 17 19 21 23
A1 A2 A3 C1

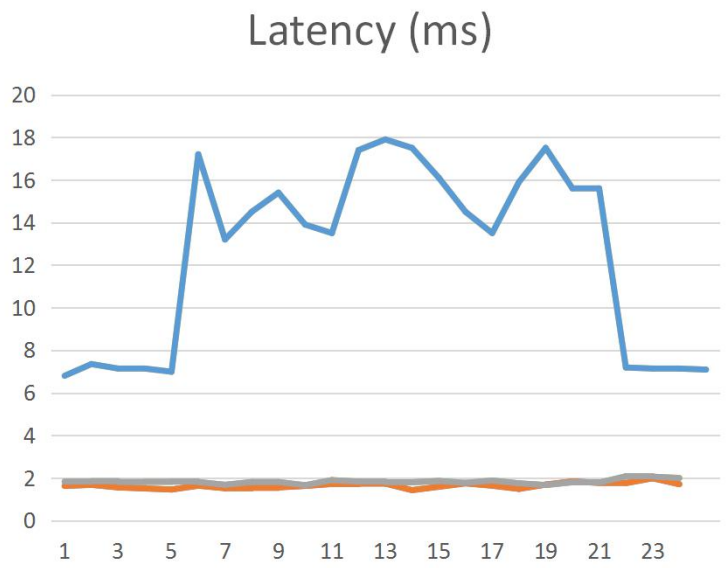
云间的网络


Bandwidth (MB/s)
180
160
140
120
100
80
60
40
20
0
1
3
5
7
9
11
13
15
17
19
21
23

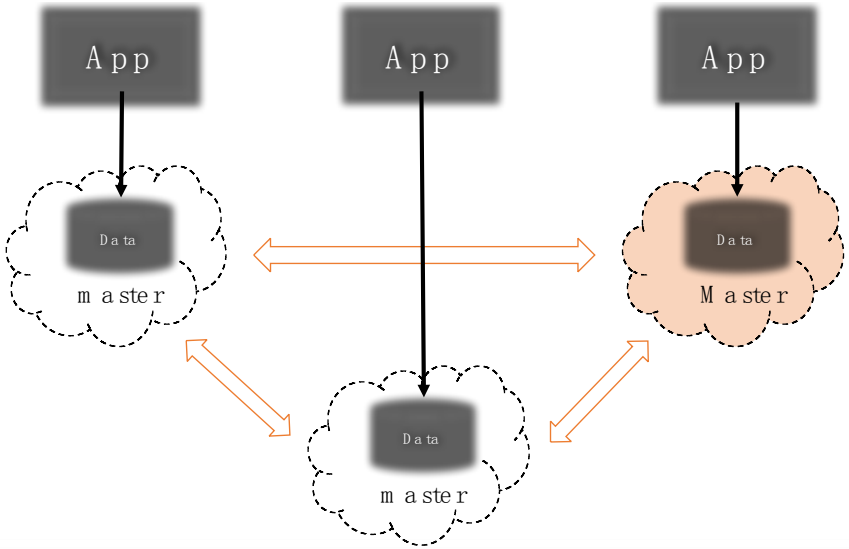

Latency (ms)
35
30
25
20
15
10
5
0
1
3
5
7
9
11
13
15
17
19
21
23

最后的选择


Bandwidth (MB/s)
90
80
70
60
50
40
30
20
10
0
1
3
5
7
9
11
13
15
17
19
21
23


Latency (ms)
20
18
16
14
12
10
8
6
4
2
0
1
3
5
7
9
11
13
15
17
19
21
23

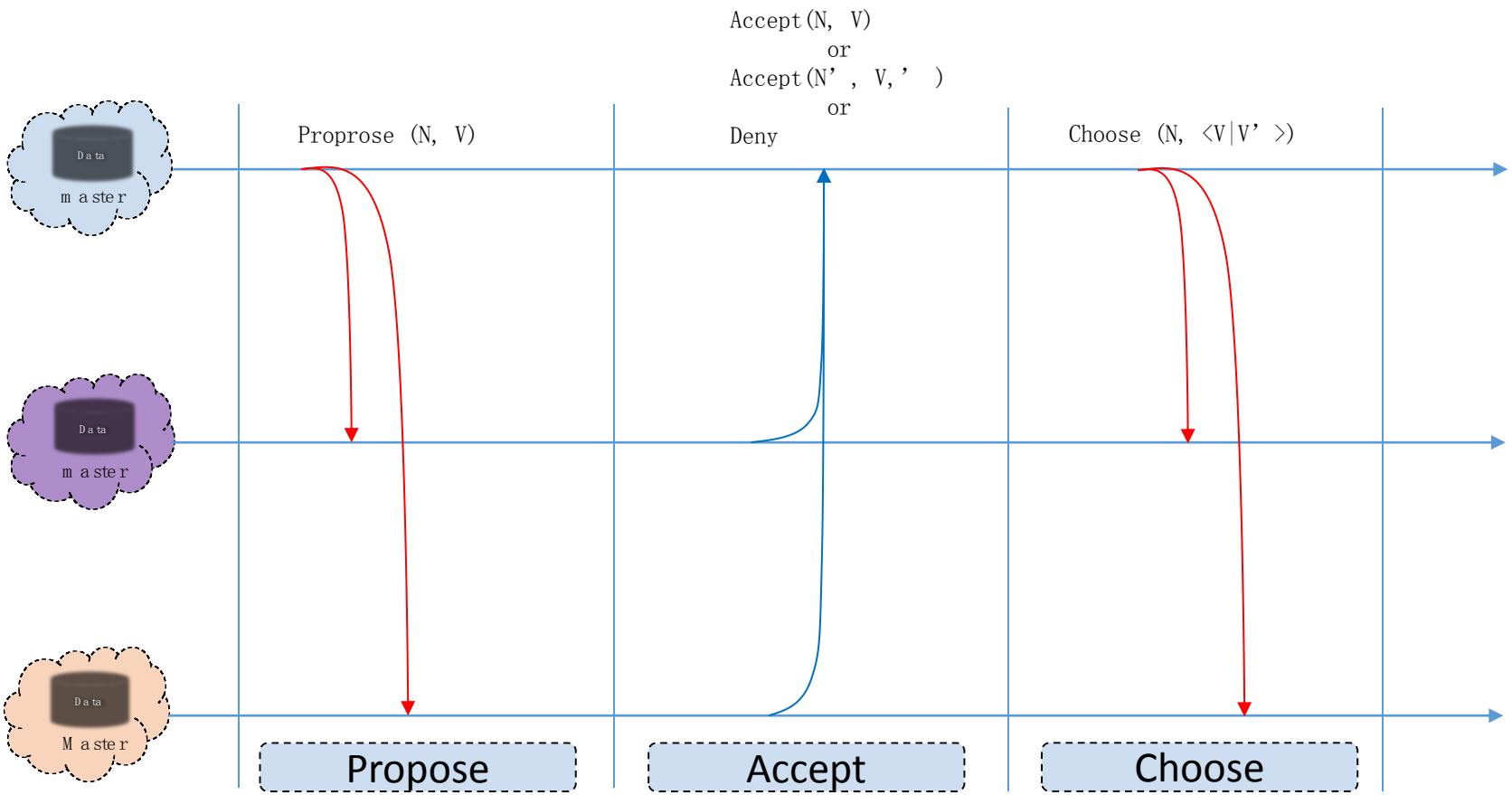
Master / Master / Master



- 公网传输数据（专线成本过高）
- 允许1个云计算数据中心宕机
- 允许网络传输不稳定
- 允许时钟不同步
- 当数据差异发生时可以做到多数票

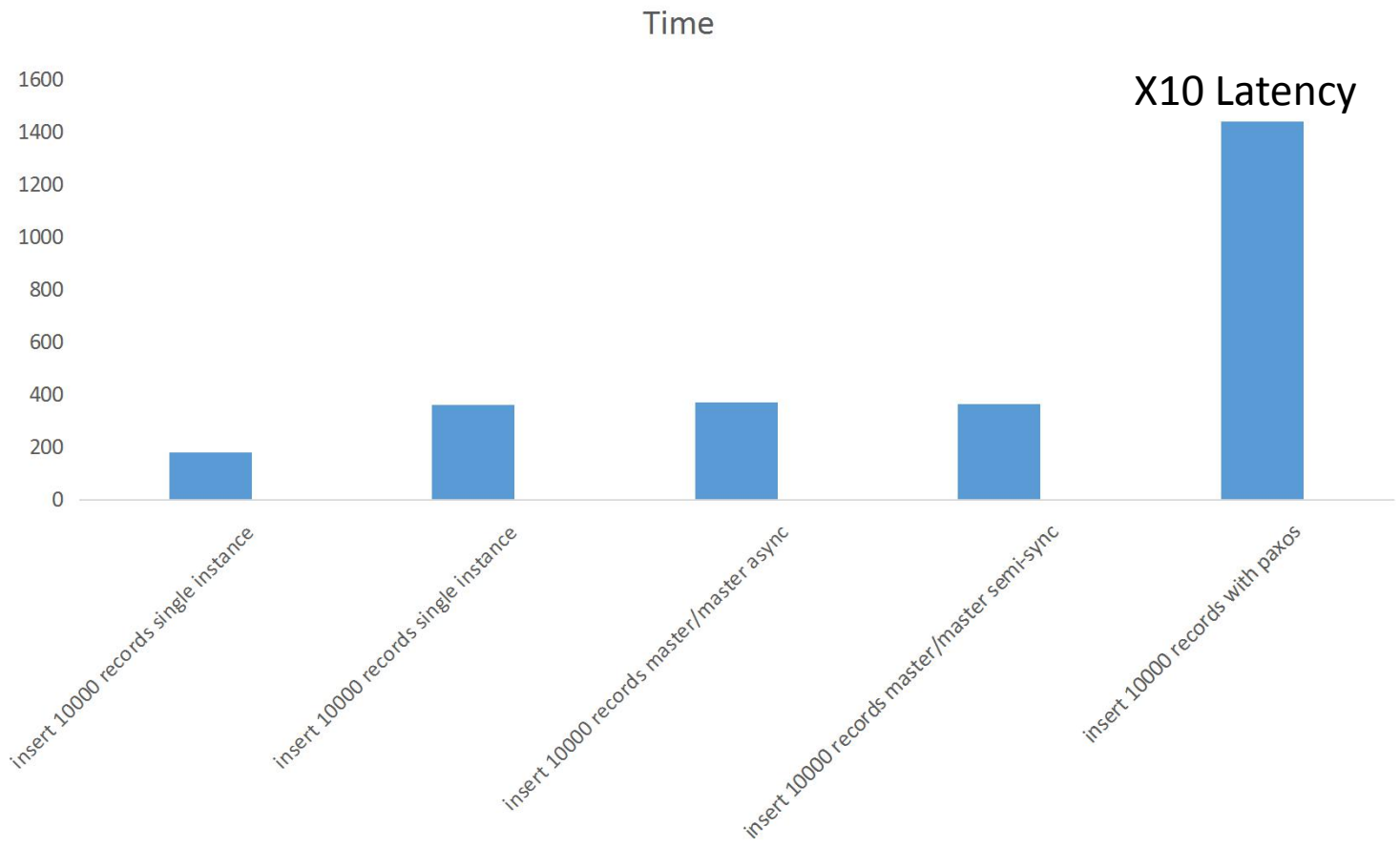
PAXOS

Paxos PG数据库



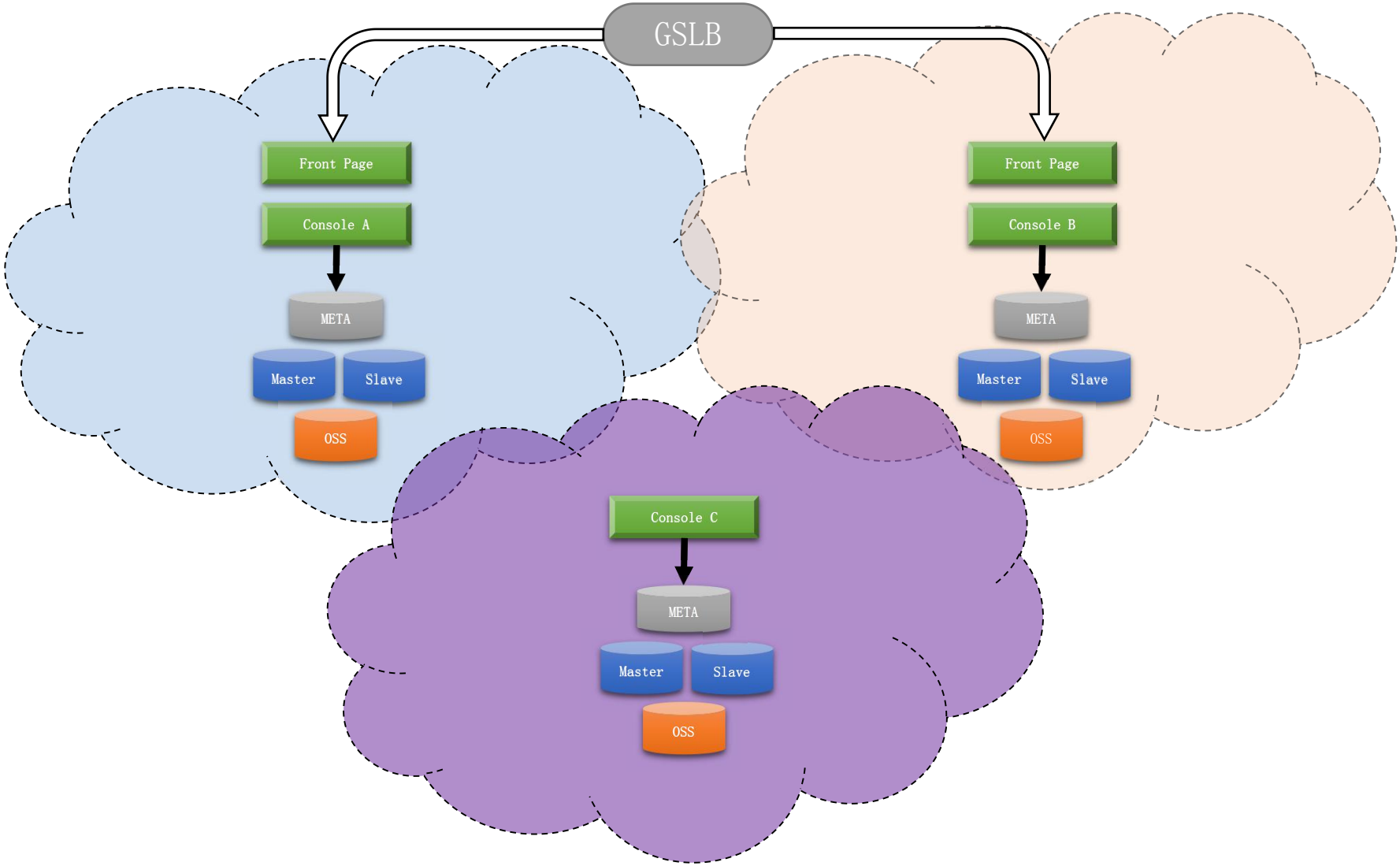
https://github.com/citusdata/pg_paxos
https://www.youtube.com/watch?v=WX4gjowx45E

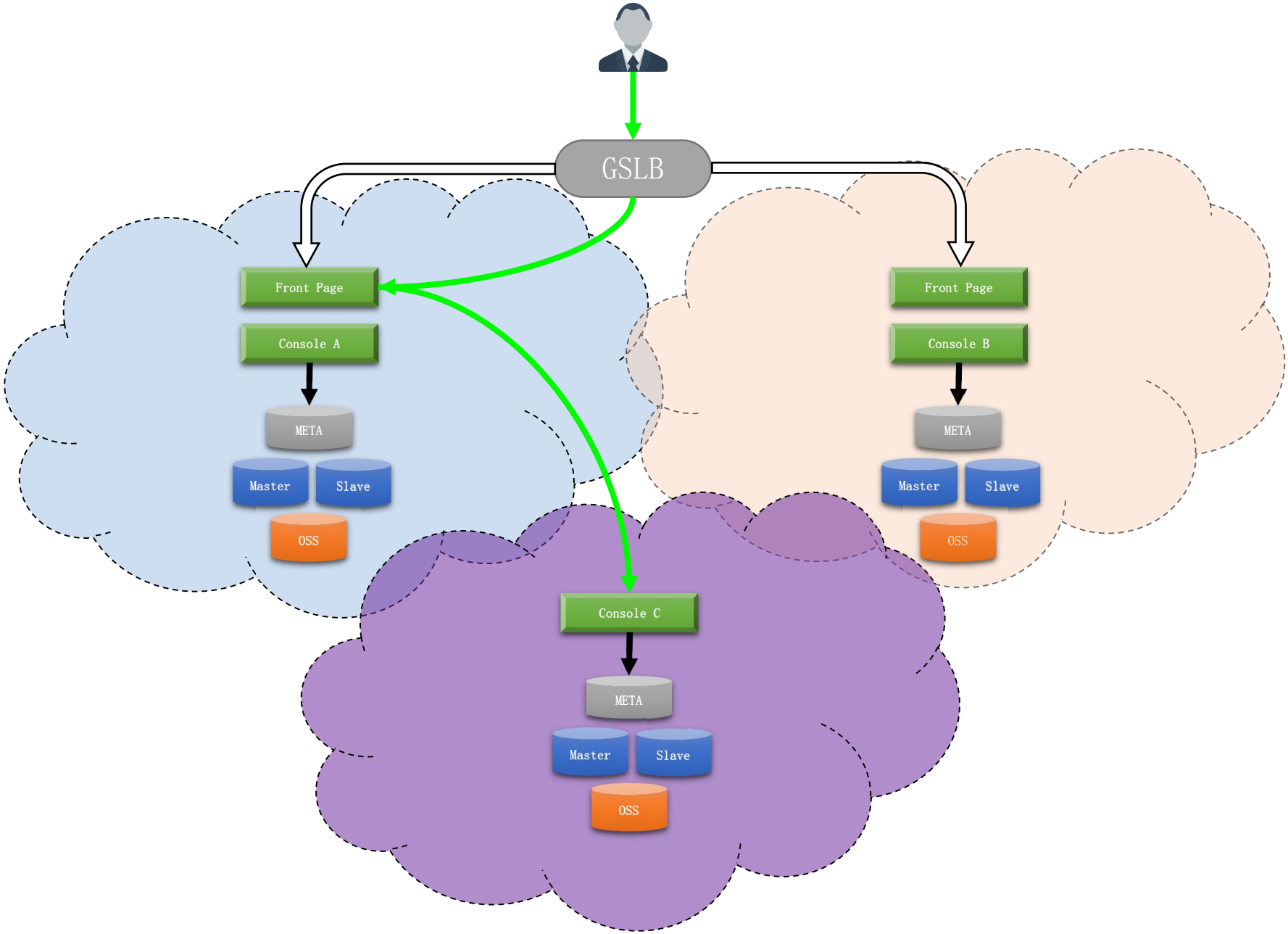
Paxos PG 性能问题

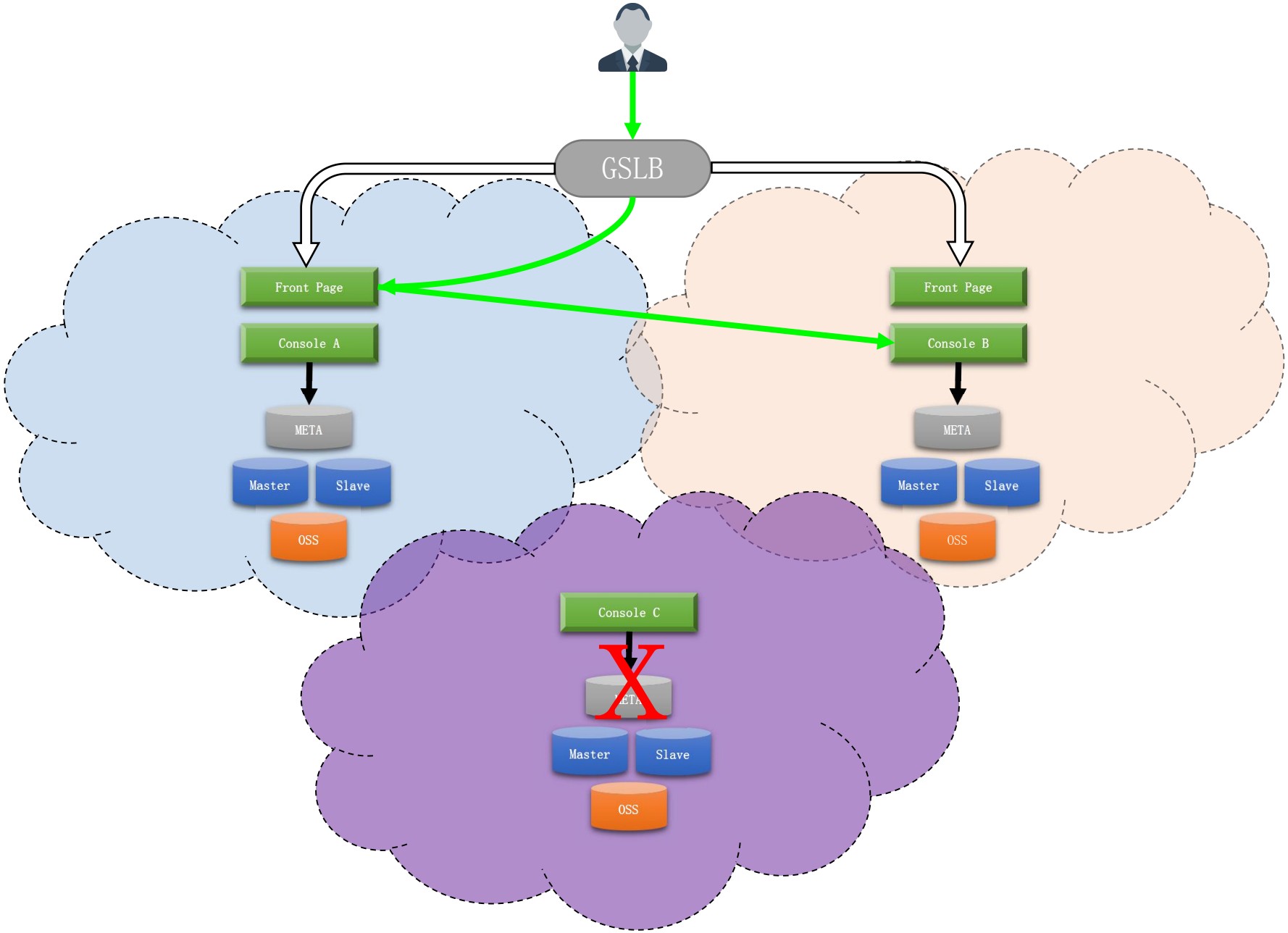


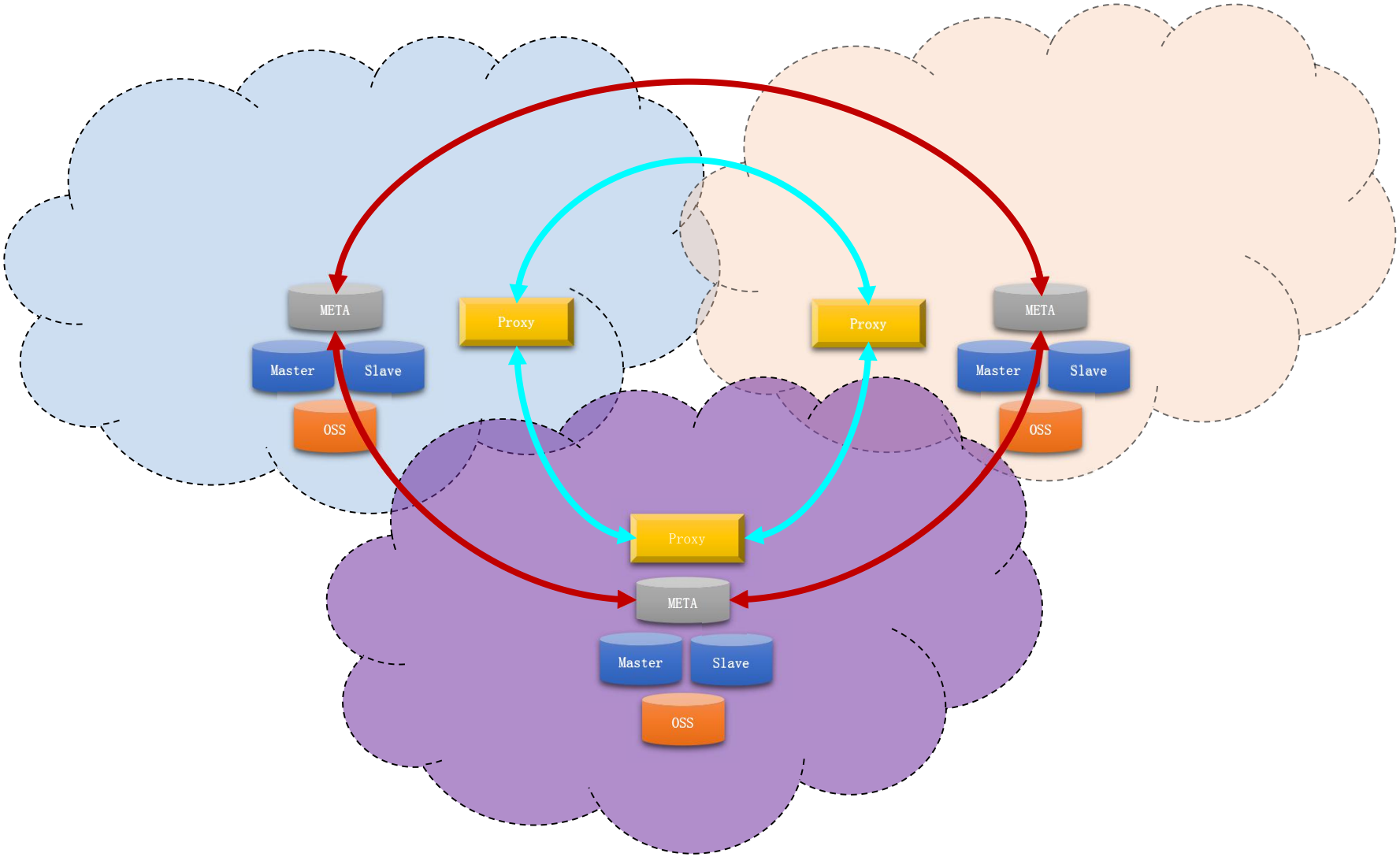
数据的分类处理

数据分类	常见操作	存储选择
元数据 （用户信息、权限、记账）	Create Read Update Delete	Paxos PostgreSQL
资产数据 （云上资源资产信息）	Create Read Update Delete	双实例数据库 ＋ 定期批量复制到从站 ＋ 资源实际状态定期更新
操作数据 （云上资源的操作日志）	Create Append Read	OSS和异步复制


GSLB
Front Page
Console A
META
Master
Slave
OSS
Front Page
Console B
META
Master
Slave
OSS
Console C
META
Master
Slave
OSS


GSLB
Front Page
Console A
META
Master
Slave
OSS
Front Page
Console B
META
Master
Slave
OSS
Console C
META
Master
Slave
OSS


GSLB
Front Page
Console A
META
Master
Slave
OSS
Front Page
Console B
META
Master
Slave
OSS
Console C
X
Master
Slave
OSS


META
Master
Slave
OSS
Proxy
Proxy
META
Master
Slave
OSS
Proxy
META
Master
Slave
OSS

尚需解决的问题

- paxos_pg 事务性的问题
- paxos的算法的数据代理？？？
- 异地数据中心部署的可能性
- 10～100万量级云资产管理

总结

- 故障理所当然发生
- 打破Dev|Ops的边界，双方共同构建可用性
- 理解业务目标，按需架构设计
- 学会妥协

VC³云管家
跨平台、智能化管理运维
让所有云轻松用、易管理

Make Cloud Work for You

WWW.VC3MARKET.COM