

太原站

KING 大 加面对面

— 数据库平替之高可用保障业务零中断 —



徐小强

- 金仓首批最具价值倡导者KVA
- 亚太地区最年轻 Oracle ACE Pro
- 公众号“JiekeXu DBA之路”主理人
- 资深数据库专家，长期活跃于一线数据库架构设计、运维管理、故障诊断
- 金仓KCA、KCP、KCSM 认证
- Oracle OCP/OCM 认证
- MySQL 5.7/8.0 OCP 认证



业务连续性之 浅谈金融数据库高可用

分享嘉宾：徐小强



01 Oracle RAC 两地三中心架构

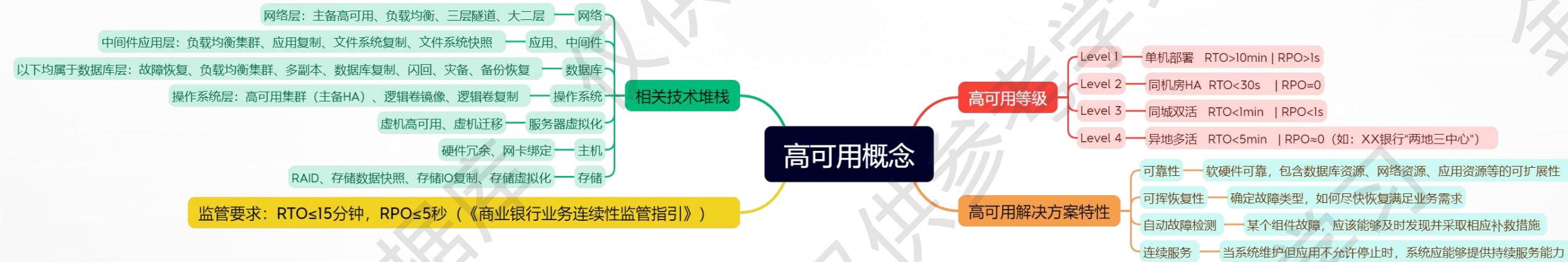
>>> Oracle RAC

>>> RPO

>>> RTO

>>> SLA

高可用架构几个概念



业务连续性 = 99.99%可用性 + 秒级故障转移 + 数据零丢失

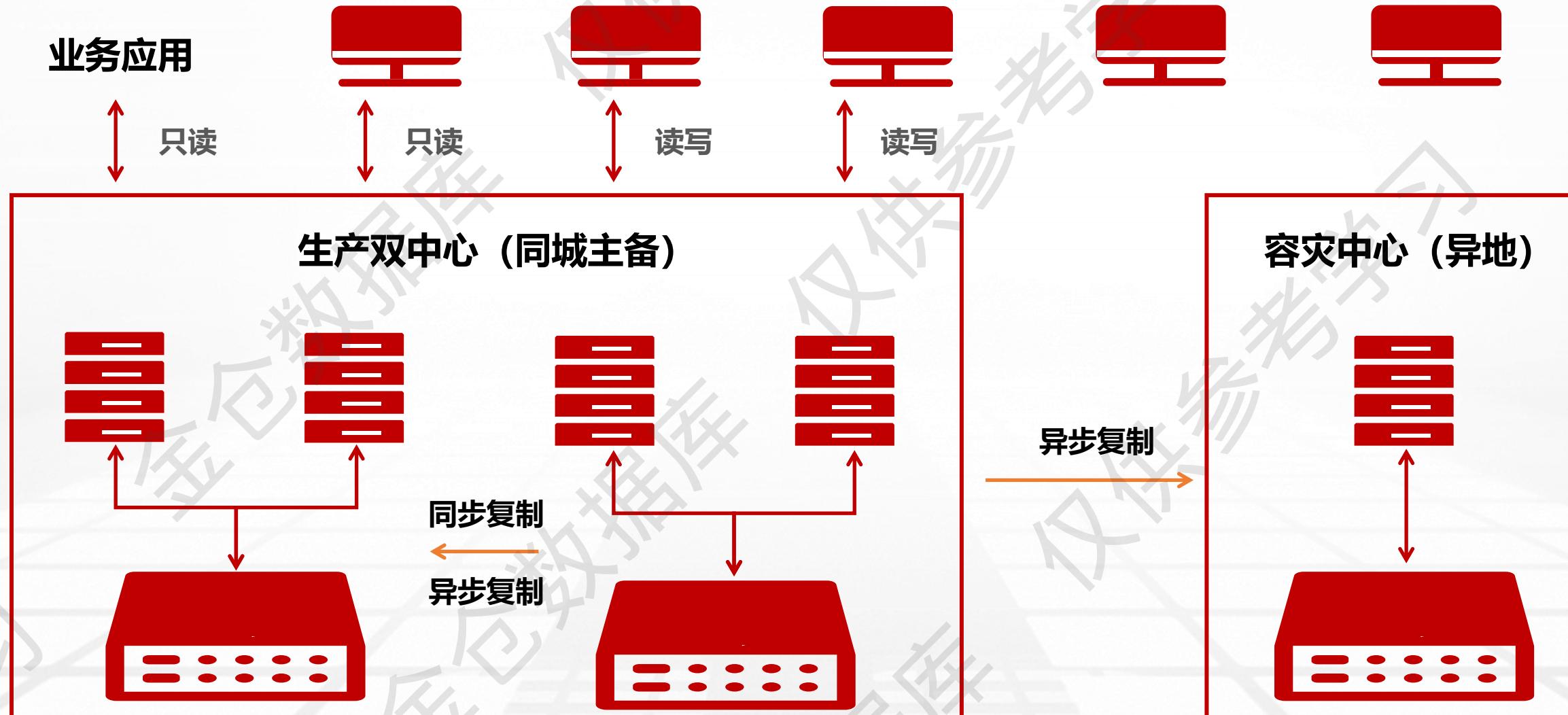
指标	含义	金融业要求	
RTO (Recovery Time Objective)	故障恢复时间	≤15分钟	RTO是能力: “故障恢复速度 = 运维部门的尊严”
RPO (Recovery Point Objective)	数据丢失量	0 (或秒级)	RPO是底线: “丢失客户交易记录的银行, 终将被客户抛弃”
SLA (Service Level Agreement)	可用性承诺	99.99% (全年52.56分钟 24*365*60*0.01%)	SLA是契约: “99.99%不仅是数字, 更是对客户资金的敬畏心”

可用率 年宕机时间
99.9% ≈8.76h

99.99% ≈52.6min

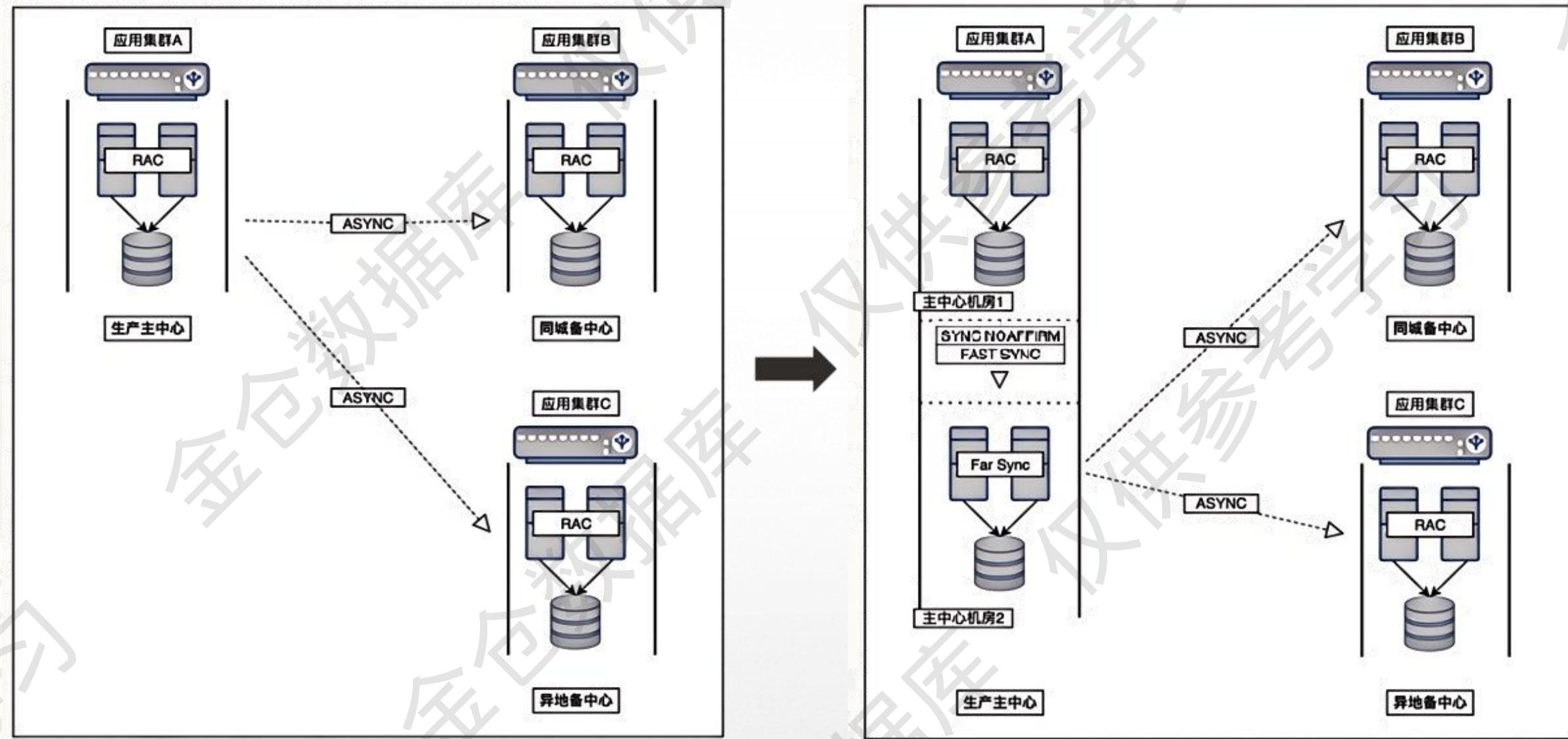
99.999% ≈5.26min

在金融行业中, RPO 决定生存底线, RTO 体现技术实力, SLA 承载商业信任, 三者共同构成业务连续性的铁三角



两地三中心架构

KING BASE | 金仓社区



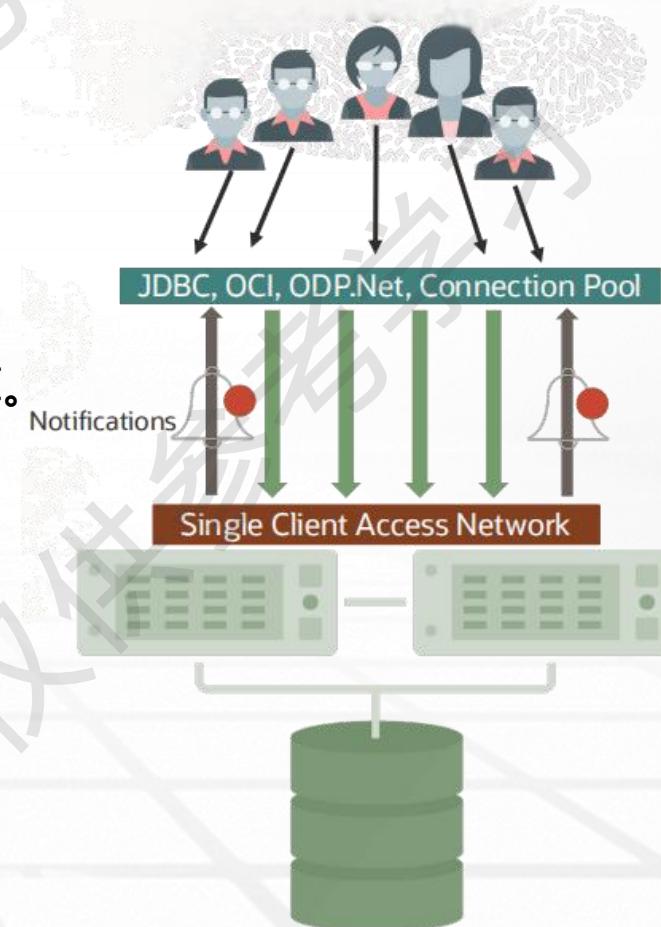
19c 透明应用连续性 (TAC) (Transparent Application Continuity)

透明应用连续性(Transparent Application Continuity)

什么是 TAC?

➤ 着重于用户体验

- ✓ 屏蔽计划内维护和意外停机。
 - ✓ 应用可以透明切换到可用的冗余节点— RAC 存活节点或者 ADG 备库。
 - ✓ TAC 通过自动恢复进行中的事务，来为应用程序屏蔽数据库层故障。
 - ✓ 数据库层的中断对用户来说似乎只是稍微延迟的执行。
- 有效地掩盖了如滚动打补丁、网络故障、数据库实例崩溃以及切换到 Active Data Guard 备库等中断事件。
- 配置简单，几乎无代码更改，对应用的透明度高。



19c 透明应用连续性 (TAC) (Transparent Application Continuity)



使用非默认数据库服务（默认服务与数据库或PDB具有相同的名称）。您创建的服务提供了位置透明性和HA功能。

For example:

Transparent Application Continuity

```
$ srvctl add service -db mydb -service GOLD -preferred inst1,inst2 -failover_restore AUTO -failoverretry 1 -failoverdelay 3 -commit_outcome TRUE -failovertype AUTO -replay_init_time 600 -retention 86400 -notification TRUE -drain_timeout 300 -stopoption IMMEDIATE
```

Application Continuity

```
$ srvctl add service -db mydb -service SILVER -preferred inst1 -available inst2 -failover_restore LEVEL1 -failoverretry 1 -failoverdelay 3 -commit_outcome TRUE -failovertype TRANSACTION -replay_init_time 1800 -retention 86400 -notification TRUE -drain_timeout 300 -stopoption IMMEDIATE
```

Transparent Application Failover

```
$ srvctl add service -db mydb -service BRONZE -preferred inst1 -available inst2 -failover_restore LEVEL1 -failoverretry 1 -failoverdelay 3 -commit_outcome TRUE -failovertype SELECT -retention 86400 -notification TRUE -drain_timeout 300 -stopoption IMMEDIATE
```

To add with the Data Guard role, here is the TAC example:

```
$ srvctl add service -db mydb -service GOLD -preferred inst1 -available inst2 -failover_restore AUTO -failoverretry 1 -failoverdelay 3 -commit_outcome TRUE -failovertype AUTO -replay_init_time 600 -retention 86400 -notification TRUE -drain_timeout 300 -stopoption IMMEDIATE
```

说明: **-failover_restore** 用于恢复应用程序连续性和 TAF 初始环境的选项; **-commit_outcome (TRUE | FALSE)** 提交结果; **-failovertype (NONE | SESSION | SELECT | TRANSACTION | AUTO)** 故障转义类型

TAF、AC、TAC



TAF 实战：

《使用 Service 服务配置 Oracle 19c RAC TAF 透明应用程序故障转移》

https://mp.weixin.qq.com/s/kWAD_DdBROaeWrn6mEcfQ

19c透明应用连续性 (TAC) (Transparent Application Continuity)

Use this Connection String for **ALL Oracle driver version 12.2 or higher**:

RAC 架构

```
Alias (or URL) =  
(DESCRIPTION =  
(CONNECT_TIMEOUT= 90)(RETRY_COUNT=50)(RETRY_DELAY=3)(TRANSPORT_CONNECT_TIMEOUT=3)  
(ADDRESS_LIST = (LOAD_BALANCE=on)(ADDRESS = (PROTOCOL = TCP)(HOST=primary-scan)(PORT=15210)))  
(CONNECT_DATA=(SERVICE_NAME = YOUR SERVICE)))
```

RAC+ADG架构

```
Alias (or URL) =  
(DESCRIPTION =  
(CONNECT_TIMEOUT= 90)(RETRY_COUNT=50)(RETRY_DELAY=3)(TRANSPORT_CONNECT_TIMEOUT=3)  
(ADDRESS_LIST = (LOAD_BALANCE=on)(ADDRESS = (PROTOCOL = TCP)(HOST=primary-scan)(PORT=15210)))  
(ADDRESS_LIST = (LOAD_BALANCE=on)(ADDRESS = (PROTOCOL = TCP)(HOST=standby-scan)(PORT=15210)))  
(CONNECT_DATA=(SERVICE_NAME = YOUR SERVICE)))
```



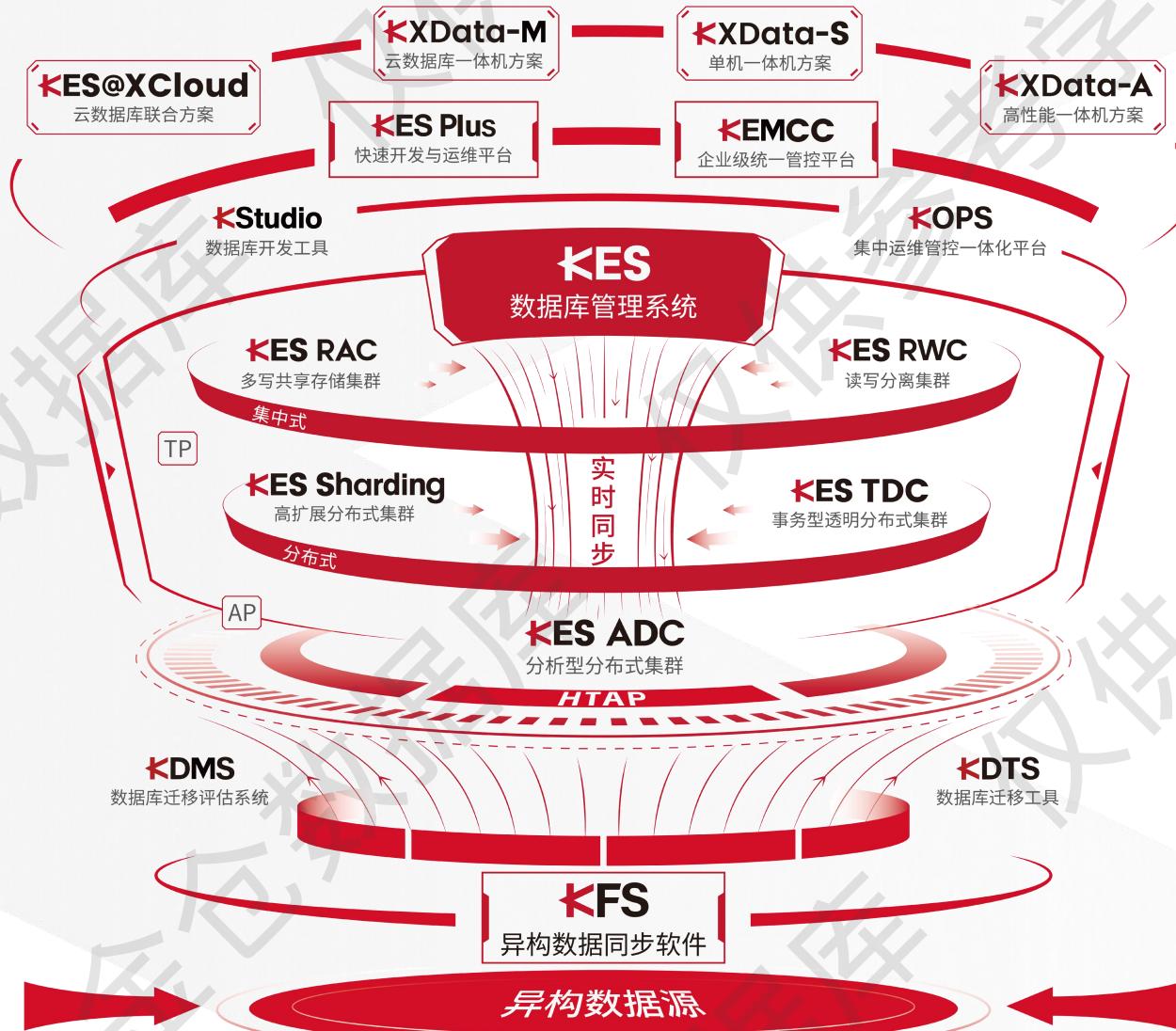
02 金仓数据库KingbaseES两地三中心架构

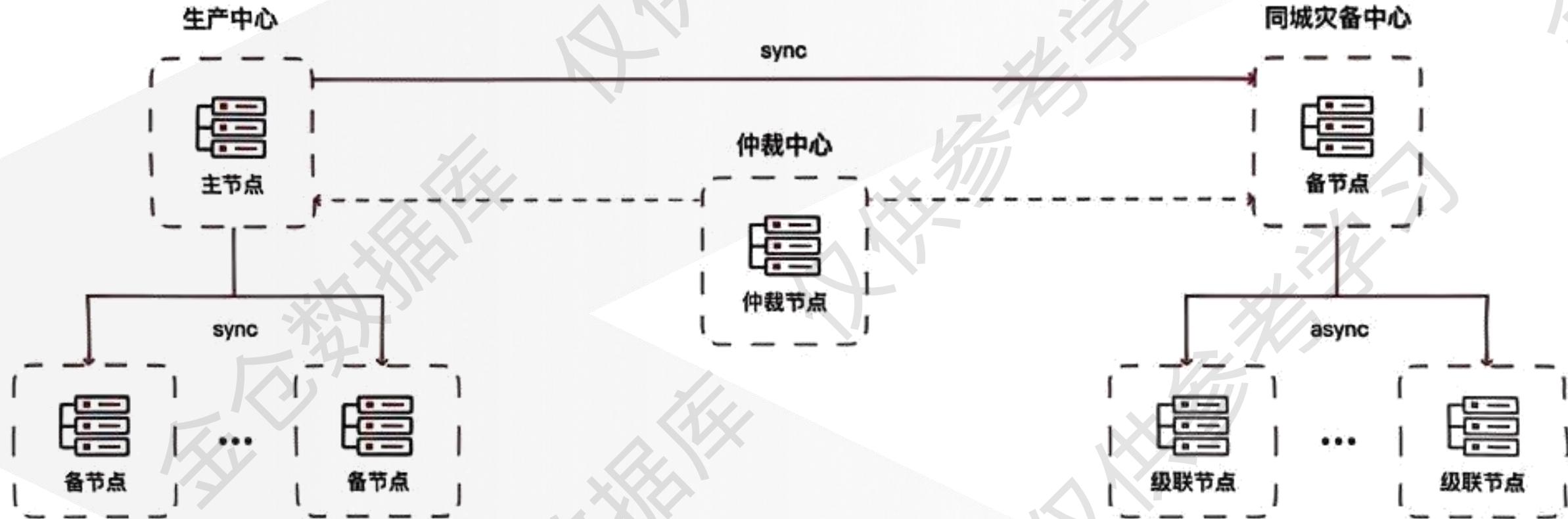
>>> KES

>>> KES RWC

>>> 两地三中心

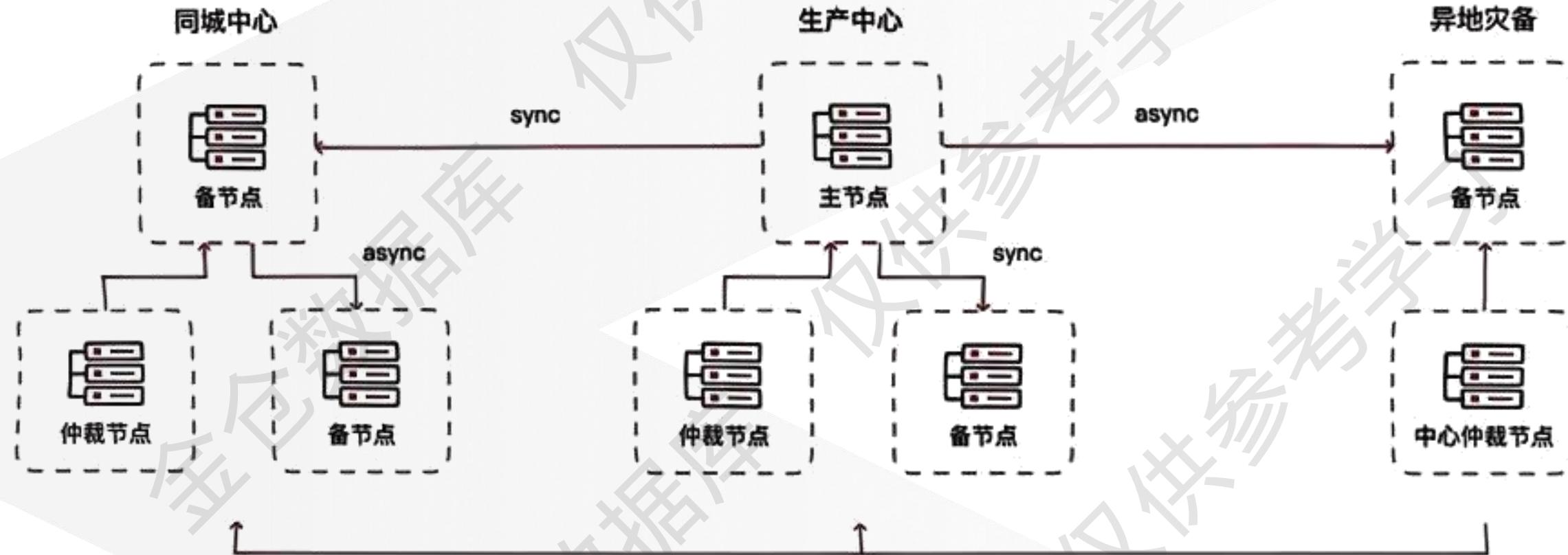
>>> 灾备





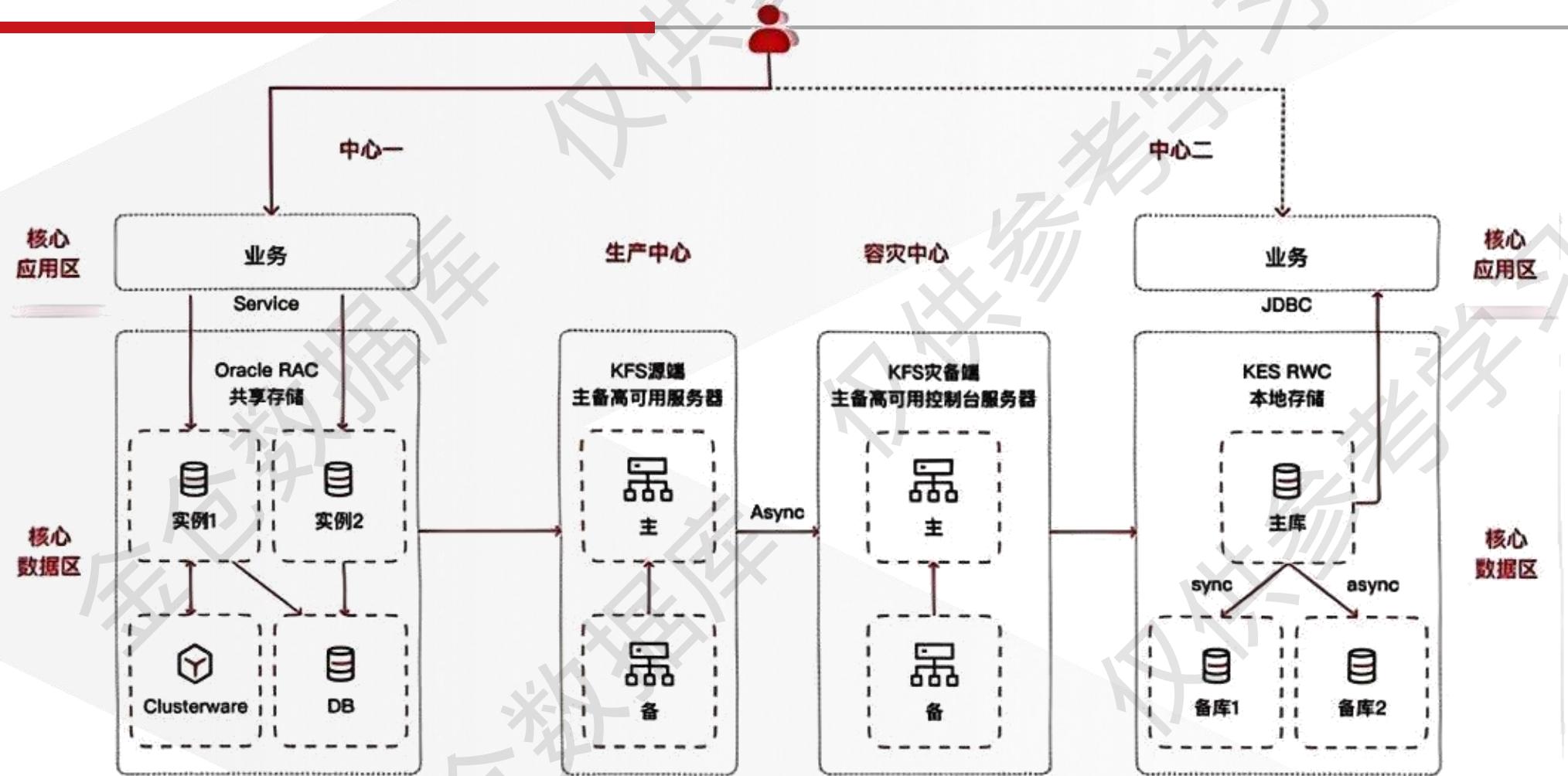
金融级高可用：支持 $RPO=0$ 、 $RTO<60s$ 的容灾切换，保障业务的安全性和可靠性。当主集群发生故障时，备集群能够数据无损地快速完成切换，替代主集群继续提供生产服务

中心内： **$RPO = 0$, $RTO < 8s$** ; 中心间： **$RPO = 0$, $RTO < 60s$**



金融级高可用：中心级&城市级故障容灾保障，保障业务的安全性和可靠性，满足国标金融级容灾标准。当主集群发生故障时，备集群能够数据无损地快速完成切换，替代主集群继续提供生产服务

中心内：RPO = 0, RTO < 8s；中心间：RPO = 0, RTO < 60s；异地中心间：RPO 低至亚秒级，RTO=分钟级



金融级高可用：中心级&城市级故障容灾保障，保障业务的安全性和可靠性，满足国标金融级容灾标准

异地中心间：RPO = 秒级，RTO=0

JDBC连接串

➤ 一主两备环境

主机: 192.168.2.129:54325 对应节点名: node1

备机1: 192.168.2.130:54325 对应节点名: node2

备机2: 192.168.2.131:54325 对应节点名: node3

虚拟IP: 192.168.2.128 VIP

➤ 连接串内容

jdbc:kingbase8://192.168.2.128:54325/TEST?ConfigurePath=jdbc.conf

或者简化为:

jdbc:kingbase8:TEST?ConfigurePath=jdbc.conf

ConfigurePath指定 jdbc 的配置文件名字, 可以带全路径, 也可以不带, 不带路径时就是 JVM 的 user.dir 目录。

➤ JDBC 配置文件内容

主库IP端口用户密码等连接信息、是否使用读写分离功能、备机地址端口、检查 master 是否存在、失败重发的最高次数和时间间隔、监测线程每次监测的间隔时间等等内容

➤ 《KingbaseES高可用最佳应用实践》

<https://bbs.kingbase.com.cn/docHtml?recId=d16e9a1be637c8fe4644c2c82fe16444&url=aHR0cHM6Ly9iYnMua2l uZ2Jhc2UuY29tLmNuL2tpbmdiYXNlWRvYy92OS9oaWdobHkvYXZhaWxhYmlsaXR5L2luZGV4Lmh0bWw>



➤ 复制槽异常案例

因迁移切换备机下线，流复制的复制槽状态异常，主库进行大量 DML 操作导致 WAL 日志迅速膨胀，远远超出 max_wal_size 参数所限定的最大限定值，表大小变的很大，死元组增多。

这个故障原因就是典型的 **流复制复制槽失效导致 WAL 日志增长和表的死亡元组无法被清理**。

解决办法也比较简单，则是删除复制槽：

```
select * from sys_replication_slots;
```

```
select * from sys_drop_replication_slot('repmgr_slot_1'); select sys_create_physical_replication_slot('repmgr_slot_1');
```

➤ KES集群切换故障案例

./repmgr standby switchover 命令在备库执行切换时 **ERROR** 提示数据目录异常

```
[cli@localhost bin]$ ./repmgr standby switchover
[NOTICE] executing switchover on node "node1" (ID: 1)
[INFO] set the primary to the readonly mode
[ERROR] "data_directory" parameter in "repmgr.conf" on "node2" (ID: 2) is incorrectly configured
[HINT] execute "repmgr node check --data-directory-config" on "node2" (ID: 2) to diagnose the issue
[INFO] set the primary to the read-write mode
[cli@localhost bin]$
```

根据 check 命令提示检查数据库目录，发现数据库启动时使用了相对路径，repmgr 切换会严格校验路径一致性。解决办法则是使用绝对路径重启，然后进行主备切换。

```
[cli@localhost bin]$ ps -ef|grep kingbase
root      1089      1  0 22:08 ?        00:00:00 sys_securecmd: /home/cli/cluster/install/kingbase/bin/sys_securecmd -f /opt/kes/etc/securecmd_config [listener]
cli       4905      1  0 22:15 ?        00:00:01 /home/cli/cluster/install/kingbase/bin/repmgrd -d -v -f /home/cli/cluster/install/kingbase/bin/../.etc/repmgr.conf
cli       5783      1  0 22:15 ?        00:00:00 /home/cli/cluster/install/kingbase/bin/kbha -A daemon -f /home/cli/cluster/install/kingbase/bin/../.etc/repmgr.conf
cli       7364      1  0 22:18 ?        00:00:00 /home/cli/cluster/install/kingbase/bin/kingbase -D ..../data
cli       7366     7364  0 22:18 ?        00:00:00 kingbase: logger
cli       7368     7364  0 22:18 ?        00:00:00 kingbase: checkpointer
cli       7369     7364  0 22:18 ?        00:00:00 kingbase: background writer
```

➤ KES集群双主故障案例

对于因网络故障或主从切换造成的“双主”故障时，我们需要分情况手动处理，大概流程如下所示：

KES双主故障处理



THANKS

成为世界卓越的数据库产品与服务提供商

