

腾讯云商业银行核心系统 分布式转型白皮书

2025 版



| 版权声明 |

©2018-2025 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分文本。

| 商标声明 |

 腾讯云 及其他腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。
本文档涉及的第三方主体的商标，依法由权利人所有。

| 服务声明 |

2025年6月

本文档中的信息，腾讯云不作明示、默示的保证。本文档基于现状编写，在本文档中的信息和意见，均可能会改变，恕不另行通知。
本文件未授予您任何腾讯产品的任何知识产权的法律权利。

主导单位



联合单位
(排名不分先后)



序言

从今年中共中央、国务院印发的《数字中国建设整体布局规划》，再回顾到去年由央行印发的《金融科技发展规划》，金融科技作为技术驱动的金融创新，已经逐步成为金融供给侧结构性改革和增强金融服务实体经济能力的重要引擎。数字技术的快速迭代和演进，也为金融机构的数字化转型注入了充沛的活力，使得金融科技逐步迈入了高质量发展的新阶段。

商业银行的业务系统建设一直是电子化、数字化最为成熟且全面的行业之一。银行核心系统一直作为IT建设最为关键应用支撑着整个银行的商业运转，其业务系统的高性能、稳定性和安全防范以及系统数据的产生、传输、存储一直是银行IT管理者日思夜想的关键问题。在过往几十年的金融IT建设历史过程中，高度依赖海外公司在关键技术平台的能力；在最近的几年，以云计算平台、分布式数据库、操作系统、中间件等为代表的国产化技术软件逐步走进大众视野，随着各类厂商与银行核心团队紧密的配合，已经为金融行业国产化技术创新打造了坚实的基础。

腾讯作为一家互联网技术公司，一直在数字技术的探索和创新不断深耕，致力于通过科技与创新，创造用户、商业与社会价值；从刚刚发布的腾讯新一轮财报中可以看出，代表数实经济的金融科技和企业服务板块已然是腾讯集团最大的营收来源，过去5年腾讯在研发投入的开支上累计超过2000亿元。

我很荣幸作为推动金融科技发展的一员，能够参与到中国数字金融的建设中。从2015年上线的国内第一家民营银行—微众银行开始，我们就看到了数字技术能够为金融科技发展带来了更多元的可能。我们用云平台、分布式技术与微众银行共同建设了国内第一家以分布式云架构为核心的银行系统，这也为我们后续在商业银行核心系统的分布式转型提供了坚实的基础和宝贵的实践经验。

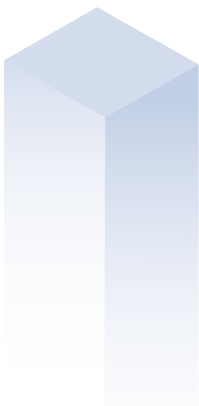
在过去的几年，我们先后为国有大行、股份制银行、城商行和省农信联社等二十余家银行机构提供了全面分布式转型的咨询和落地实践。这里不仅仅是腾讯作为技术平台的提供方，还有像神州信息、长亮科技、中电金信等优秀的系统服务商共同努力和攻坚的结果。从18年开始，我们与这些优秀的ISV厂商开始逐步建立合作，在这过程中我看到了作为推进中国数字金融科技建设的使命感和责任心。几个月前，我向优秀的同仁们发起邀请，希望能够将我们过去几年沉淀的合作案例和经验做系统性的梳理，以为目前正在规划和研究的金融机构提供一些参考。

这份白皮书是我和团队在过往的银行系统分布式转型项目中，总结的一套符合目前商业银行IT发展现状的解决思路，也同样结合了与核心系统服务商的先进实践经验。非常感谢参与本书编写的合作伙伴和团队成员，我希望能够通过本书，为金融机构的IT从业者提供一些新的思路和实践参考，“仅以厚劲之心，尽绵薄之力”。

我也同样相信随着吾辈不断破解发展瓶颈和难题，推动中国金融科技全面迈入积厚成势的新阶段将势不可挡！

胡利明 腾讯云副总裁

2023年4月 深圳



PART 1

银行业核心系统行业分析

- 1.1 银行核心业务系统面临多重挑战 / 02
- 1.2 分布式转型是银行核心升级的必然路径 / 03
- 1.3 业内建设模式分析 / 05

PART 2

银行业分布式核心六大设计要点

- 2.1 银行业分布式核心建设的困难与挑战 / 08
- 2.2 腾讯海量服务之道在金融业的发展 / 09
- 2.3 腾讯云银行分布式核心系统整体架构 / 10
- 2.4 腾讯云银行分布式核心方案六大设计要点 / 12

PART 3

腾讯云分布式核心两种技术架构路线介绍

- 3.1 微服务与单元化架构的差异 / 16
- 3.2 客户如何选择 / 17
- 3.3 腾讯云分布式核心—标准微服务架构路线 / 18
- 3.4 腾讯云分布式核心—单元化架构路线 / 26

PART 4

腾讯云分布式核心系统技术底座

- 4.1 APaaS分布式技术组件 / 45
- 4.2 IPaaS系统集成服务平台 / 46
- 4.3 GPaaS通用技术组件服务 / 47
- 4.4 IaaS基础设施服务 / 48

PART 5

腾讯云分布式核心与伙伴联合方案

- 5.1 腾讯云与神州信息的深化合作 / 50
- 5.2 腾讯云与长亮科技的深化合作 / 52
- 5.3 腾讯云与中电金信深化合作 / 54

PART 6

落地实际案例

- 6.1 某国有大行核心系统单元化案例 / 58
- 6.2 某国有大行核心系统案例 / 60
- 6.3 某西南省级城商行全面云原生案例 / 62
- 6.4 某城商行核心系统升级案例 / 64

银行业核心系统行业分析

在当下“不断变革、持续创新”的时代里，变革与创新对银行业信息科技的影响是不可忽视的。

· 随着互联网以及移动支付的飞速发展，银行业务持续飞速发展和变革，在过去十几年中，主要经历以下三大转型：

一、**网点转型**：从传统物理网点向“线上云网点”转型，大量必须到现场网点柜台办理的业务，可以通过远程视频银行实现流程闭环。

二、**渠道转型**：从传统ATM、STM等用户自助渠道，向线上手机银行、微信银行、社交银行等客户主动自主渠道不断拓展延伸。

三、**开放银行**：金融服务以开放的形态嵌入到用户衣、食、住、用、行的多样场景之中，每个人生活的方方面面，都无处不金融。

· 信息科技变革的内外驱动力共同推进转型：过去的信息科技体系以集中式架构为主，而数字经济时代更加强调核心技术体系的开放和自主掌握，整个技术体系存在非常强烈的转型升级诉求，这是内在驱动力；随着互联网金融时代的到来，银行业务的渠道多样性和营销多样性，也不断要求信息科技更加敏捷、弹性、智能和安全，这是外在驱动力。正是内外的共同驱动，使得当下的银行业信息科技，正处于加速变更和创新转型的关键期。

· 在银行信息科技建设工作中，核心业务系统承载“存、贷、汇、银行卡、清结算、核算”等核心交易，也普遍被认为是信息科技建设工作的核心。基于银行客户的调研经验，核心业务系统是否成功转型升级，也成为衡量信息科技变革成功的关键标志，这是行业的普遍共识。

但是问题也随之而来：

银行核心业务系统如何进行转型？

技术体系建设的趋势和发展方向是什么？

什么样的路径才是适应银行自身情况的最优路径？

.....

以上都是核心系统从业者一直在思考探索的问题。

腾讯金融云团队基于多年在消费互联网、产业互联网以及金融核心系统领域的实践和沉淀，联合核心领域的生态合作伙伴，共同整理推出本书，提供一些解答建议，希望为银行业核心系统的转型升级添砖加瓦。

第一章，我们将从“银行业核心系统行业分析”入手，分析当前银行核心系统所面临的挑战和痛点。后续章节，再将对银行核心系统转型升级的方案建议层层剖析，为读者提供可落地的最佳实践参考。



1.1 银行核心业务系统面临多重挑战

近年来，科技领域快速进步、银行业务跨越式发展，在此环境下，银行传统核心系统面临诸多挑战：

难以适应激烈的市场竞争

传统国有大行和股份制商业银行纷纷实行客户下沉策略，加剧了同业竞争；互联网公司和其他非金融机构进入了支付结算、投融资领域，构成跨界竞争。在这些激烈的竞争环境下，大部分银行的老核心系统已难以适应，无法应对客户细分、实时风控等诸多诉求。

产品迭代创新速度慢

业务品种迅速增加，对核心系统的需求快速增多，以产品创新为例：针对新产品、新政策的调整，现有核心业务系统往往不能实现产品和参数的快速配置，导致新产品开发周期长，响应速度慢，不能快速满足业务发展需求。

差异化利率定价支持弱

大部分核心业务系统基于以往的监管环境与市场环境设计，随着利率市场化进程加快，核心系统已不能灵活实现利率差异化定价。新核心建设需完善利率定价机制，理顺内外部利率体系，提升自主定价能力。

技术体系封闭，自主掌握能力不足

技术架构方面，现有核心系统由于技术架构陈旧，多以闭源体系为主，以IOE体系的主机、数据库和存储来构建，也无法适应开源体系下开放性和标准化的生态场景，不能满足目前国产化和自主可控的要求。

应对海量高并发能力存在不足，边际扩容成本高，硬件供应存风险

随着互联网支付体系的发展，银行核心系统经常要应对线上活动的潮涌现象，针对一些特殊事件的营销活动，会突然产生非常高的交易并发，突破了传统核心的设计性能上限。传统集中式架构多以垂直扩容为主，边际扩容成本高收益少，且依赖硬件老化，供应存在风险。

功能模块紧耦合、迭代创新速度慢，多业态业务发展的瓶颈

传统核心模块与模块之前存在耦合关系，交易的参数化程度也不够，随着外围渠道的不断增多，交易的配置化和服务可编排的诉求越来越强烈。核心系统每一次版本更新，都需要进行多轮测试和多个环境的充分验证，牵一发而动全身，对外围系统的影响巨大，因而整个系统的迭代速度比较缓慢。

1.2

分布式转型是银行核心升级的必然路径

在上一节，我们看到了传统银行核心系统目前所面临的普遍性挑战和痛点，特别是目前时代迫切需要在关键基础设施与核心领域具备国产化的技术体系。而技术体系的发展成型和成熟并不是一朝一夕就能完成。所以未来几年，银行行业核心系统需要尽快开展转型升级已成为一种共识。

那么回到最初的问题，银行核心系统如何进行转型升级？最优的实现路径是什么？近年来，经过腾讯团队对业内银行核心系统的充分调研、腾讯内部金融业务的相关实践、以及腾讯金融云团队在金融领域持续探索，我们认为分布式转型是银行核心系统升级的必然路径，也是最优路径。

1.2.1 银行业务和技术需要从“集中式”往“分布式”转型 >>

集中式架构其实是一种非常优秀的架构，也支持了近30年来国内银行业进入到PC时代，实现数据大集中的发展历程。

从业务逻辑上来看，集中式架构是把应用开发过程中，业务逻辑集中起来，以实现业务上线为目标。并不太多的去考虑在架构设计上如何让各个业务板块松耦合，带来的问题也是业务逻辑太集中，导致“牵一发而动全身”，一个业务改动修改上线时间动不动就需要几个月，和典型互联网企业的按天来迭代，上线前还能动态进行修改的敏捷迭代相比，在竞争中就会处于不利的态势。

所以这些年来，从业务功能层面，银行核心系统其实已经在不断进行转型去集中化：

首先，是从“胖核心”到“瘦核心”转型。最早的传统核心系统非常庞大，包括银行涉及的各类业务处理。比较典型的像客户信息、票据业务、国际结算、托收、总账等业务系统都囊括在核心系统之中。二代支付兴起之后，这些业务系统逐渐从核心系统剥离，分别由专业的应用系统来承载。核心系统简化成了交易与核算分离的交易处理系统。

其次，交易“微服务化”，功能模块独立化。这个是目前银行核心系统在应用开发上的共识和重要趋势。随着互联网金融业务的兴起，移动端已经占据90%以上的对客交易，银行核心系统以传统的面向账户为中心，转向为面向客户为中心。银行核心系统需要适应于互联网业态下的产品敏捷更新，以及电商促销、秒杀活动带来的交易瞬间高并发及交易潮涌现象。集中式架构已经不能满足核心业务敏捷更新、系统快速扩容的需求，分布式微服务的架构应运而生。服务拆分为越来越小的业务单元，通过模块化、组件化的领域构建，沉淀为多个互相独立的能力中心，应对前台业务的快速变化及系统的动态扩容。

技术架构层面，集中式架构是技术上“把鸡蛋都放在一个篮子里”，同时这个篮子也能满足在稳定性、可用性等各方面的要求。银行核心业务系统从集中式往分布式发展的主要原因有：

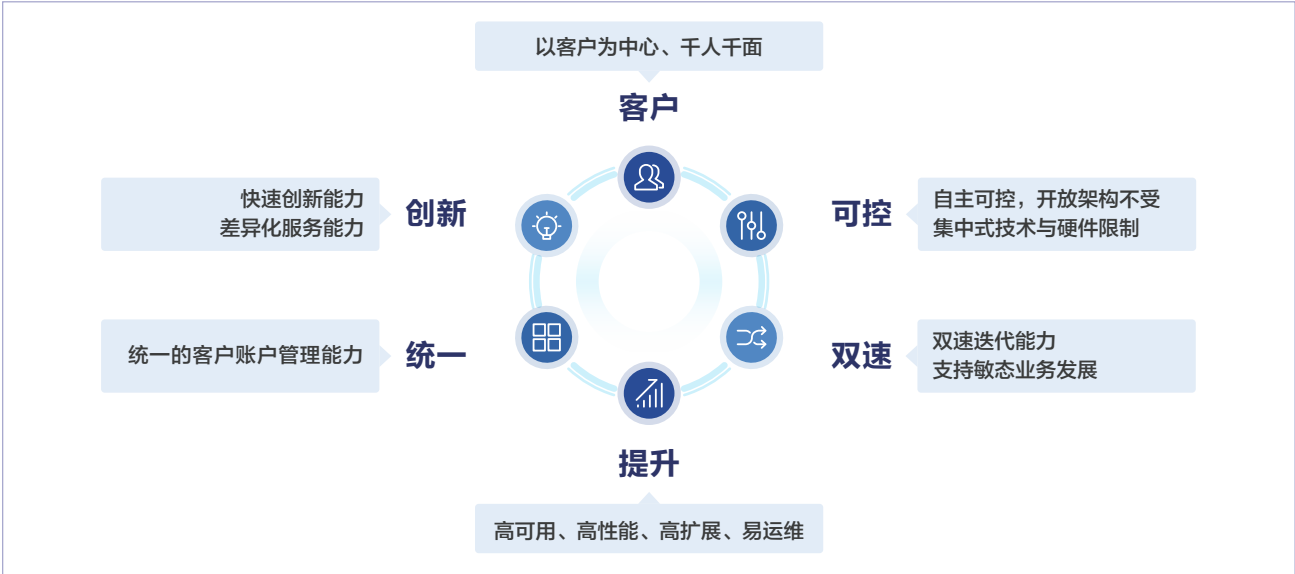
集中式架构的最大问题是相关技术体系掌握在少数一两个国外厂商手中，整个技术体系是封闭不开放的，这是目前行业共识必须要进行的改变。

集中式架构对硬件依赖较高，边际扩容成本高效益低，并非可持续的架构。且存在硬件老化停产所带来的风险，需要提前规划新一代核心的技术架构转型。

集中式架构的内部模块紧耦合、无法独立发布，导致整体迭代低效，无法很好适应未来敏态业务的场景，需要重新设计应用架构，使其支持松耦合、可组装的业务模块，并能支持各模块间自治与独立演进的技术体系。

1.2.2 银行新一代核心业务系统需要具备的特点 >>

基于上文分析和讨论，我们总结新一代核心业务系统所需的主要特点：



业务层面

- 以客户为中心、千人前面的设计理念，金融产品和服务以客户为中心开展，实现客户信息与金融产品紧密关联，可基于 AI 大模型算法整合行为数据（如交易频次、APP 停留时长）、外部场景数据（如电商消费、社交关系）等，生成千人千面的客户标签体系，支撑精准服务匹配，为银行提供全方位的决策信息。
- 快速创新和差异化服务能力，支持产品创新的端到端全流程设计与管理，通过 AI 大模型解析市场需求与监管政策，自动生成产品参数化配置模板（如利率浮动规则、风险对冲条款），缩短新产品上线周期。差异化定价，可基于 AI 大模型动态整合客户信用评分、市场流动性及竞争格局，支持实时利率博弈优化，助力银行构建“定价智能体”，实现弯道超车。
- 统一的客户账户管理能力。通过统一的客户及账户管理能力，实现客户信息管理及客户服务承载。通过集中的账户管理能力，做到全行一本账，实现全行财务会计精确归集与分析。

技术层面

- 三高一易，即高可用、高性能、高扩展、易运维。核心系统是银行的命脉，牵一发而动全身，系统必须稳定可靠且具备多重高可用保障机制。“面向弹性容错”的原则是设计分布式核心系统高可用架构的核心思想。借着是高并发与弹性扩展：核心系统交易需要满足互联网业态下的高并发要求，具备熔断限流机制。在特殊活动的潮涌现象下，可以进行弹性扩容，提高交易并发能力。运维方面，核心系统的运维是一个长期工程，必须具备良好的可观测和易维护特性。各交易链路，端到端可追溯，可以进行统一的日志管理与检索分析，同时要保证核心系统的运维简便高效。
- 自主可控性。新核心需要通过开放的、标准的技术体系实现自主可控的核心技术平台，不再受传统架构和硬件的约束，使外围系统能够快速的与核心系统进行集成和交互。能有效应对数据量变化快，业务多变的新业态场景。
- 双速迭代能力。以稳态和敏态业务为指导设计技术架构，通过稳态服务和可组合的敏态服务实现的动静分离部署，核心系统要能够快速的根据业务要求进行服务编排和服务重组实现快速交付。开发态则通过可视化编排和研发效能构建双速迭代的能力。

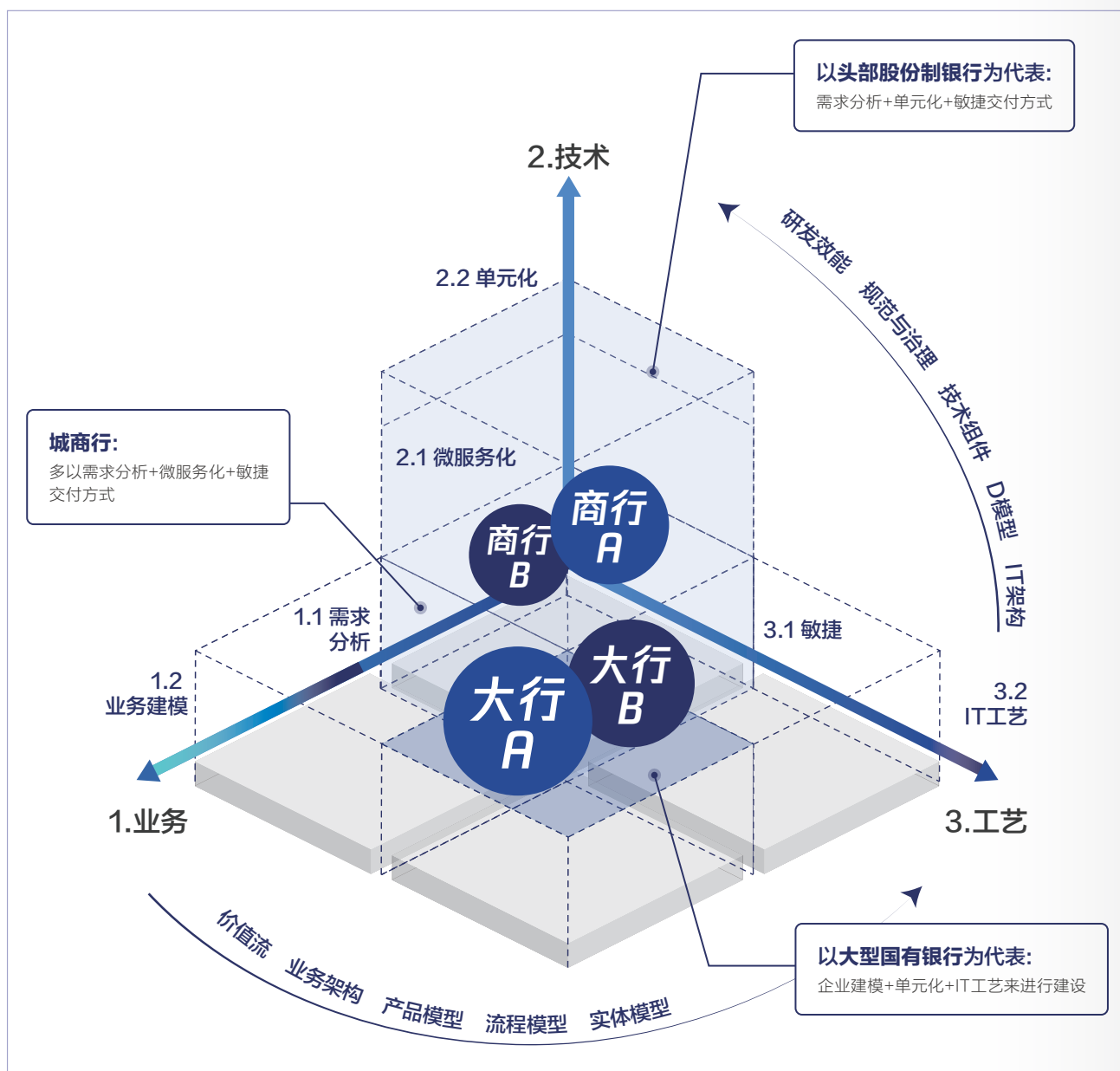
这些特点都指向了基于开放体系建设的分布式架构路线，因为具备了技术多元性、开放性、标准化等特点，生态丰富且易于集成；不依赖具体硬件，可通过横向扩容来扩展业务系统的吞吐量；同时，更适应松耦合的应用架构，有众多技术框架和工具可支持业务模块的独立自治。其中微服务又无疑是当下分布式架构建设中最为主流且成熟稳定的方案，是未来几年银行核心系统转型大浪潮中的必然选择。

1.3

业内建设模式分析

腾讯云根据多家案例分析得出当下银行新核心的建设模式主要分为两种：**技术平移和规划重建**。技术平移是科技部门内部能够闭环的模式，在保持功能基本不变的情况下，将应用迁移到分布式平台，或将应用适配到国产数据库或云平台上。规划重建则需要联合业务部门，结合全行业务发展战略，整合业务需求，重新设计建设新一代核心系统。

技术平移首先要确定平移目标，如平移到国产化的云、操作系统、数据库、中间件等等。其次是要确定平移的路径，比如优先使用产品、定价等模块做试点，再平移核算，形成完善的平移经验和工序后，最后是进行存贷模块的平移。规划重建整体上比技术平移更具挑战，腾讯联合ISV伙伴结合实际案例，归纳提炼**3维6阶模型**，用于指导新核心规划重建的模式定位。





业务维度

一阶段采用需求分析方法建立业务需求；二阶段采用业务建模方法建立企业级业务模型。



技术维度

一阶段建立微服务体系，实现应用自治，独立演进；二阶段建立单元化架构实现流量闭环与多活基础架构。



工艺维度

一阶段采用DDD+敏捷的交付模式；二阶段采用IT实施工艺承接企业建模产出与技术模型之间的衔接设计。

业务建模是最早由建行联合众多业内专家实践产生的需求分析与统一建模的方法论，随后在几家国有大行的新核心建设中逐步丰富延展。IT工艺是与之配套的新核心建设方法论，其能承接业务建模的产出，用业务模型指导微服务模型落地，用识别、定义、分解等标准工序拆解复杂架构，最终基于构件实现松耦合的组装和编排，在复杂项目设计与建设中发挥了重要作用。

单元化是一种将计算与数据绑定处理的架构设计思想，其有多种不同的落地形式。单元化与微服务架构并不冲突，是微服务架构能力的加强，也是未来实现异地多活的架构基础。

目前国有大行和头部股份制银行在新核心建设上多以采用业务建模、IT实施工艺和单元化架构的组合为主要的建设模式。这种模式下资金和人力投入都相对较高，对行内业务与科技人员要求也普遍较高。其具有模式和架构上的先进性，能起到引领同业科技和标杆的作用。科技力量较强的银行客户在新核心的设计建设过程中，会投入较多的自有人力、提升自研比重，以掌握关键方法、模型和技术来提升自主可控力。

除此之外，大部分的银行客户以需求分析、敏捷和微服务架构的组合作为其建设模式。这种模式下资金和人力投入相对可控，以把控业务方向和应用设计为主，建设工作主要以ISV厂商为主。方案的标准化程度高，建设风险小，同业案例多，是较为成熟可控的建设模式。

由于业务建模、IT工艺和单元化架构的学习成本及资源投入相对较高，各家厂商也正在积极推进将这些方法和架构沉淀，作为产品的一部分输出，以减轻客户在这块投入多、落地难的情况。整体上来看，建设模式与各家行自身的业务体量、科技规划等有着较强关联，只有符合自身场景、满足实际需求的建设模式才是有效的、可持续的选项。



PART

2

银行业分布式核心六大设计要点

腾讯结合众多银行分布式核心实践，综合提炼出百余项公共技术能力，并联合专家多轮推演论证、梳理沉淀，形成新一代银行分布式核心整体架构。

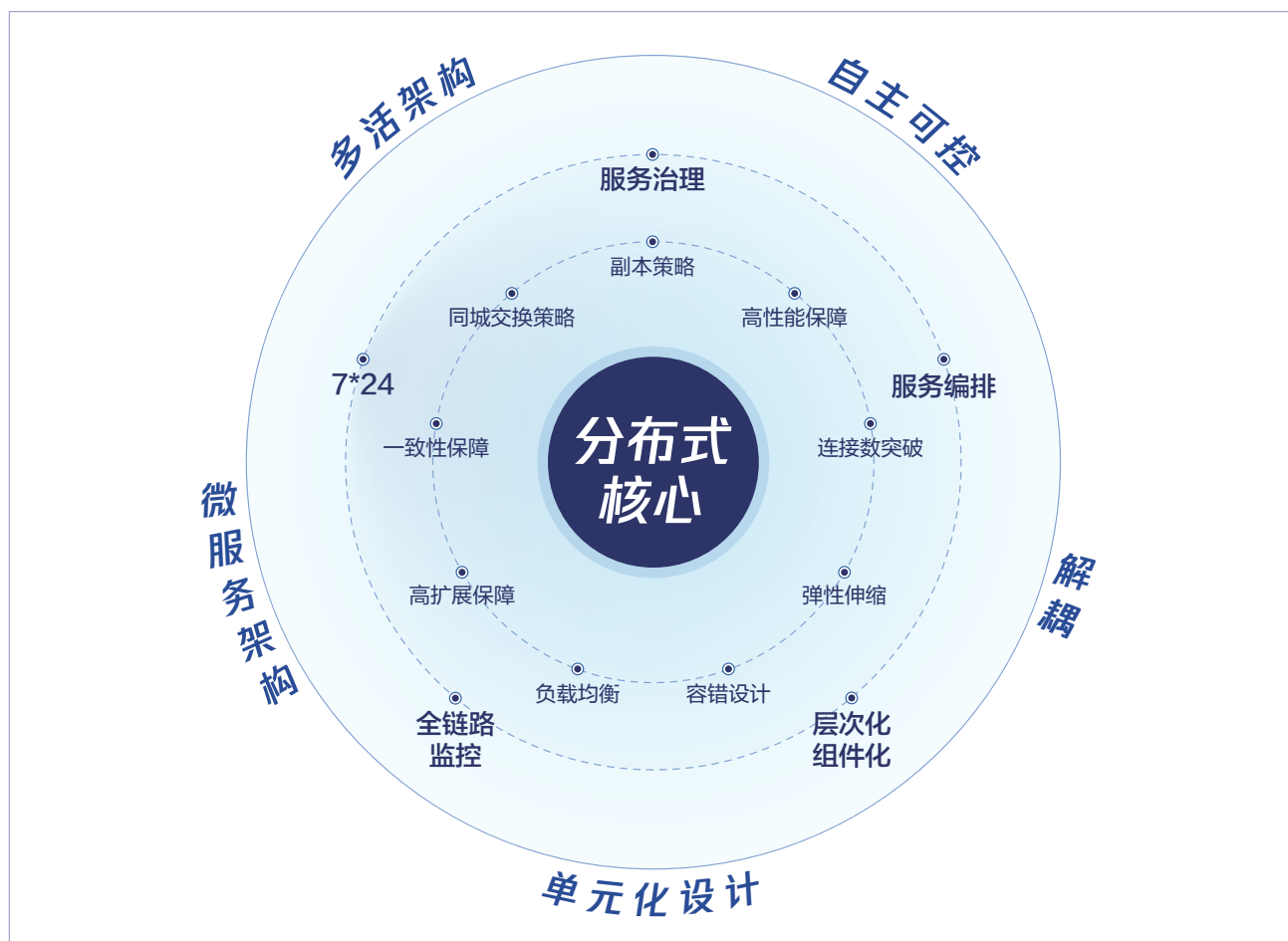
本章将重点介绍腾讯经过调研与实践，归纳提炼的六大分布式核心设计要点。



2.1

银行业分布式核心建设的困难与挑战

在互联网和开源技术广泛普及的今天，银行的核心系统逐步从封闭的技术生态，迈入开放的技术生态。在分布式架构下，丰富多元的技术组件及相对高故障率的基础设施对过往依赖单一技术栈和高度稳定主机平台上的银行核心架构带来空前的挑战。



分布式核心系统建设过程中，会面临各类困难与挑战。

接入层

需要考虑全行报文贯标、链路追踪、服务治理、流量染色及统一认证等相关能力。

服务层

需要考虑业务领域与微服务之间的设计定律，过度松散的模块会对系统多个方面带来的冲击。

数据层

需要考虑数据分片后的一致性问题，以及多活架构下的数据同步策略等等。

此外，还需要考虑海量服务下的自动化运维问题，大型交付团队中的研发和组织效能问题等。

核心系统建设本身具有投入大、影响面广、风险高等特点，再叠加上述分布式核心建设的新困难，对银行而言无疑是更大的挑战。但机会与挑战并存，新核心的建设通常被赋予诸如提升业务与技术能力、带动全行业务模式转型、推动科技创新、培养复合型人才等多项目标。只有通过科学的规划论证，严密的组织管理，多方的共同参与才有实现的可能。

2.2

腾讯海量服务之道在金融业的发展

从腾讯公司成立早期的时候，所推出的各个产品，如 PC 互联网时代的 QQ、QQ 空间、移动互联网时代的微信等应用，在技术架构上无一不都面临**如何应对互联网海量用户访问业务的压力和挑战**。从很早期，腾讯就在探索研究如何基于分布式架构来处理应对海量互联网业务，逐步形成了“腾讯海量服务之道”，成为腾讯内部技术人员的必修宝典，成为腾讯系很多互联网应用的架构开发设计指南。

支撑海量服务之道的“七大技术手段”

1 SET模型

- SET化部署
- 业务部署标准化，统一规则
- 容量按SET动态伸缩

2 全网调度

- 业务全网分布
- 业务自协调自适应，弱化故障影响
- 灵活控制业务切换、启用，灵活调度

3 柔性可用

- 基于资源和策略对业务分层分级
- 优先保证核心业务服务质量

4 灰度发布

- 通过灰度小范围进行版本发布
- 充分验证功能复合预期后，扩大发布范围，完成变更

5 过载保护

- 架构设计考虑处理突发大量请求时，系统能安全承载，避免雪崩
- “轻重分离、及早拒绝、量力而行、动态调节”

6 立体监控

- 对业务或系统进行完整监控
- 多维度了解业务运营信息

7 自动部署

- 日常运维操作提炼固化，用自动化工具实现
- 提高效率，减少故障

随着业务的发展，海量服务之道也逐步扩大到财付通、微信支付、微信红包等涉及金融交易场景的应用实践中。在**微众银行**的核心系统实施过程中，腾讯海量服务之道结合银行业务不断打磨，微众是国内首家完全去IOE，采用分布式核心系统，服务亿级用户规模的银行。

在银行分布式核心领域，微众银行首次提出并落地了按DCN（数据中心单元化）的设计理念，按照业务和用户维度进行水平拆分，将银行核心账户体系、核心客户体系、核心业务处理等打包在一个DCN（含服务器/数据库），业务流量经全局路由实现到DCN的转发，成为国内银行业核心单元化架构的首个案例。

2.3

腾讯云银行分布式核心系统整体架构

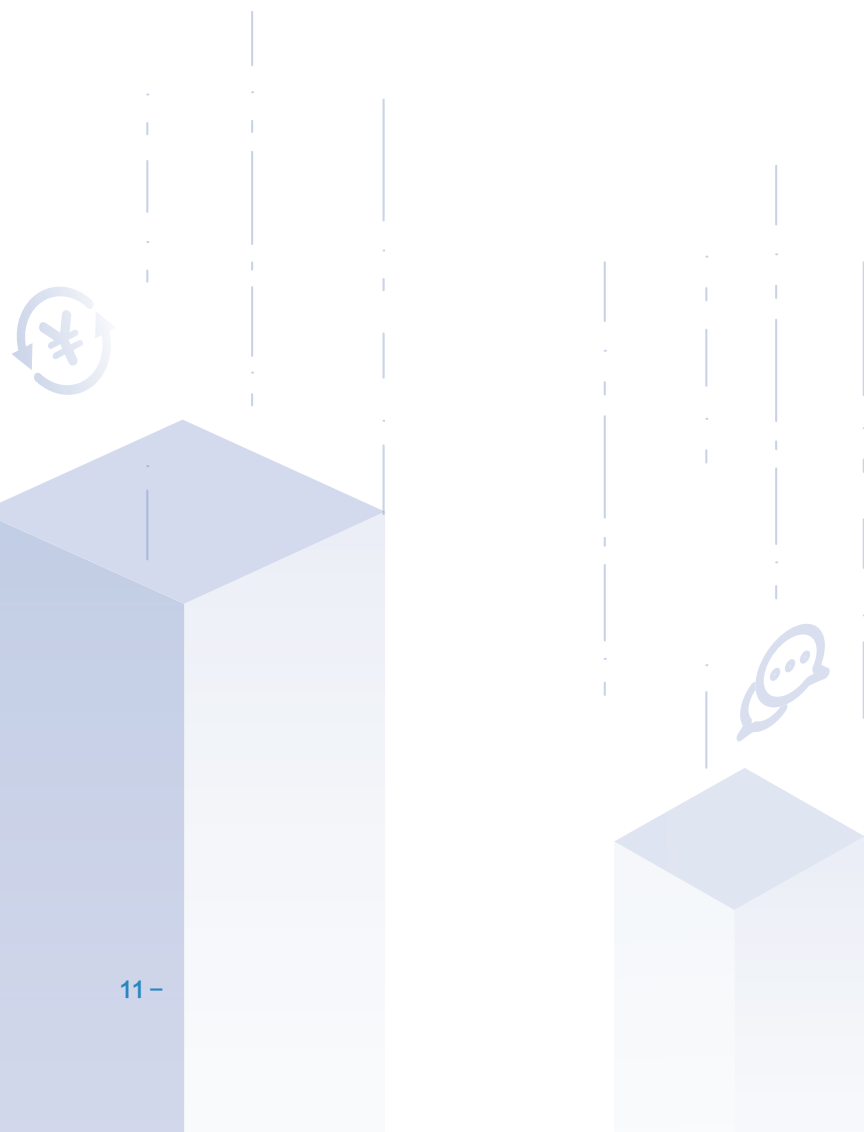
腾讯结合众多银行分布式核心实践，综合提炼出300余项公共技术能力，并联合ISV专家的经验进行多轮推演与论证，以高内聚低耦合为原则，通过拆摆并合等方法进行梳理沉淀，形成了新一代银行分布式核心系统的整体架构。



如图所示，自下而上分为 IaaS 云平台、PaaS 云原生平台、SaaS 核心应用层，左侧为研发效能体系和非功能测试体系，右侧为智能运维体系及安全体系，形成新核心所需的整体能力框架。

- SaaS 层包括了银行核心系统的应用域，如：存贷、支付结算、银行卡、核算及公共服务等。以及为支持上述领域所需的能力中心和7x24 等公共机制。
- PaaS 层呈现三级能力分化，其中 APaaS 聚焦通用业务场景封装，构建交易框架、参数管理等标准化模块并抽象云产品接口，通常由核心应用厂商提供，但部分组件与云厂商原生能力存在重合，在新核心系统建设时，需边界厘清，突破产品间的能力重叠与协议互通等实施难点，形成既解耦又协同的技术支撑体系。IPaaS 强化全链路集成治理，应对云上云下交错环境中的协议冗杂难题，通过异构系统协同形成一体化底座；GPaaS 作为通用技术服务，专注云原生中间件供给，为分布式架构提供容器平台、微服务、消息队列、缓存、数据库等基础支撑。
- IaaS 层则主要包括了云平台的计算、存储、网络、资源编排、多云管理等底座能力，以及容器安全、云组件安全、网络安全、主机安全、应用安全等基础安全能力。

- 研发效能系统,传统的研发工艺已经很难满足大规模复杂项目的协同管理。在研发效能体系中,不仅包括了传统的DevOps体系,同时还集成了 AI 项目管理和 AI 质量管控体系、AI 代码助手。并以需求为抓手,贯穿设计、开发、测试、安全、发布、投产全生命周期。每个环节形成有效的反馈机制,确保每一轮迭代都能有效量化生产数据,实现精益化管理。
- 非功能测试体系包括全链路压测、容灾切换测试、混沌测试,全链路压测模块聚焦性能瓶颈突破,通过模拟真实业务场景的全流程压力测试,持续优化核心系统的吞吐能力和响应效率;容灾切换测试模块着力高可用保障,通过多中心故障转移演练,确保系统在极端情况下的业务连续性和服务可靠性;混沌测试模块则针对分布式架构特性,主动注入基础故障与网络异常,验证弹性容错下的系统自愈能力。这种 " 面向弹性容错 " 设计可应对节点故障常态化、网络波动等挑战,提前暴露风险以提升系统稳定性与容灾韧性。
- 分布式下的运维体系,从传统的 " 快速发现 -> 快速定位 -> 快速修复 " 转变为 " 快速发现 -> 快速隔离 -> 快速恢复 "。在大规模集群中,标准服务器的硬件故障和网络抖动频发,优先需要保障业务连续性,自动化的运维能力是核心。同时可以考虑结合 AI 大模型实现故障预防,将运维工作从 " 被动应对 " 转变为 " 主动防御 "。



2.4

腾讯云银行分布式核心的六大设计要点

在技术架构设计方面，我们调研了多家银行的分布式核心系统架构，归纳提炼成六个主要设计要点。



2.4.1 数据切分策略 >>

首先要确定分布式下的数据切分策略，包括了切分维度、分布规则、扩容策略、容量评估四个重要方面。

切分维度

决定了某一维度的数据会存放在某一数据分片中，比如客户维度或省市维度。

扩容策略

业界有两种常见的做法。一种是采用自研的扩容工具，另一种是通过分布式数据库原生能力实现。具体取决于技术架构和产品选型：如果数据库采用集中式架构，则依赖于自研工具扩容；而如果采用分布式数据库架构，可借助数据库的原生能力扩容。故在确定扩容策略时，需要仔细考虑数据库的结构和特性，以便选择最合适的扩容方案。

分布规则

决定了数据存放各个分片时采用的规则，比如按范围段存放，或者按Hash散列存放。

容量评估

容量评估的目的是评估当前整体业务量，并预测未来两年到五年的增长量。在此基础上，结合数据库产品自身的容量参数，确定数据分片数量及分片配置规格。评估的原则是尽量做好冗余规划，以减少扩容场景的出现。

2.4.2 技术架构策略 >>

应用在分布式下的技术架构包括：交易处理策略、分布式事务处理策略、应用侧扩容策略、聚合查询策略、灰度发布策略。

交易处理策略

指的是一次交易从网关进入到内部应用集群时，应用集群之间服务调用的机制和故障处理策略。这包括点对点直达调用还是通过消息传递调用，以及相应的负载均衡策略。同时，还包括应用实例故障时的熔断与降级策略等。

应用侧扩容策略

针对分布式集群中的应用扩容升级，对于运行在云平台上的核心应用，可以通过实现无状态的应用实例来支持自动扩缩容；对于运行在传统 IDC 内的核心应用，可通过手工脚本方式实现应用实例的扩缩容。就银行核心场景而言，指定统一的、标准的扩容模式，对于银行核心系统的稳定运行非常重要。

灰度发布策略

灰度本质上是希望控制指定流量到指定的集群中处理，多用于验证新版本功能，或验证新的数据库、中间件及基础设施。对资源要能按照标签隔离开，流量也能按照标签进行路由，再集成管控面进行控制管理。

分布式事务策略

指的是采用分布式架构后，带来的事务一致性问题。由于分布式架构下的场景较为复杂，需要决策是由应用侧来保障事务，还是交由分布式数据库来保障。值得注意的是，任何一种事务处理机制都会存在损耗和场景限制。在指定分布式事务策略时，需要结合实际情况进行综合考虑和决策。

聚合查询策略

数据被切分之后，如何满足聚合查询的要求，需要定制相应的策略。当底层数据被分散到多个数据分片之后，原本在单一数据源上进行的聚合查询需要被下推到多个不同的数据分片上进行。最后再将这些分片的结果传回进行汇总、排序等计算。这个过程中对网络和聚合计算节点的要求非常高。

2.4.3 服务模型策略 >>

服务粒度

微服务的颗粒度是业内经常争议的话题。若颗粒度过大，将无法充分体现微服务的低耦合与自治性；若颗粒度太细，则可能对性能、稳定性、链路监控和运维产生不良影响，甚至导致大量分布式事务场景的出现。

技术组件

在分布式核心系统中，需具备一系列技术组件，用于隔离底层分布式技术复杂度，并对底层技术产品进行隔离。此层需要具备良好的南北向接口，帮助核心系统通过分层隔离来实现自主可控。

落地工艺

从应用设计到研发测试上线的整个生命周期，需要具备全方位的实施方法论。其中包括对需求产出的业务组件到微服务组件的映射，流程与实体等业务模型的识别、定义、分解等内容，应用架构的构件定义、组装、发布的端到端方法论。

2.4.4 业务连续性 >>

弹性故障恢复

“面向弹性容错”是分布式系统设计时常被采用的设计思想，即在面对高故障率和不稳定的基础设施时，业务连续性的保障需要从整体性视角考虑，应用系统为此需要考虑多种故障场景，进而做出故障自隔离、自动扩缩容等自恢复机制。

多中心多活

银行机构日益重视核心系统的高可用性。同城双活架构是主流设计，即同一城市的两个机房同时接入并处理业务请求，任一机房故障时，另一机房可秒级接管业务流量。而大型银行机构则规划采用异地多活架构，即在异地多个机房同时响应业务请求。但与互联网业务不同，银行核心业务的异地协同面临特殊挑战，如服务目录异地同步时效性、跨地域分布式事务协调等问题。

跨地域容灾

根据监管要求，银行的关键业务系统需要建设容灾体系以应多不可抗力灾害。实践中，银行通过定期灾备切换演练、将只读交易定向引流至灾备机房运行，或向灾备机房部署可控灰度流量等方式，持续验证容灾体系的有效性与日常可用性。同时，容灾机房还需符合物理距离要求等监管标准。

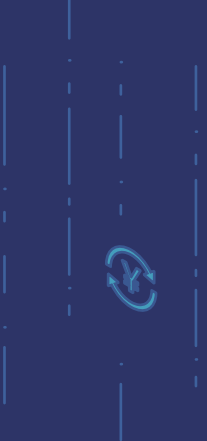
2.4.5 研发效能 >>

为配合新核心的建设，需要一套行之有效的研效体系支撑，才能发挥分布式系统敏捷、自治的特性。除了传统DevOps需要具备的集成、反馈等自动化能力之外，还需要以项目视角，从需求出发到设计，再到研发、测试、发布三个关键环节，打通各环节数据和管控面，形成一体化的持续跟踪、反馈、提升体系。除了工具层面，更应在文化、分享、度量、自动化、精益管理等多个维度进行对标和增强，实现动静结合、双速开发等全链路的效能提升。

2.4.6 分布式运维 >>

为了保证分布式核心系统的运维效率和连续性，需要建设专门的分布式运维平台，并在规划新核心时就一并启动规划。在一些实践中，出现过由于运维平台启动较晚，导致新核心相关设计出现反复的问题。此外，分布式系统的运维与传统主机的方式截然不同，需要从快速修复模式转变为快速隔离，系统需要自发现并隔离故障节点，以保证业务连续性。因此，标准化的指标采集与分析工作，多维度的监控与告警，CMDB和标准化运维动作都需要围绕分布式系统进行重新梳理和设计。最后，在规划时需要考虑大数据和AI故障智能分析与预测的能力，这将是未来分布式运维的发展趋势。

结合腾讯参与的多个新核心落地案例与多家客户的调研访谈，围绕着六大设计要点，归纳总结出两种主流技术架构路线：标准微服务架构和单元化技术架构。



PART

3

腾讯云分布式核心两种技术架构路线介绍

本章主要介绍“三维六阶模型”中技术维度的两个阶段：标准微服务架构和单元化架构之间的差异，以及这两种架构在六大设计要点上的不同架构决策。



3.1

微服务与单元化架构的差异

微服务架构和单元化架构并不冲突，单元化是微服务架构的能力增强。微服务架构更注重利用微服务的设计理念来解耦应用，以及利用微服务技术手段来重建应用间的互通和治理模式。而单元化架构更侧重整体性规划，更偏向顶层治理。它在微服务体系之上加强了标准化治理，包括标准化扩容、标准化路由，尤其对跨中心流量做了治理，让微服务中无状态的随机负载变得更加有序。微服务像是在微观层面把功能体系建立起来，单元化则更像是在宏观层面把架构体系完善起来，两者相辅相成，相互促进。

微服务架构通过服务注册和发现模型，重新定义了服务间的交互模型，解决了大规模集群下服务的自动发现和故障服务自隔离的能力，还包括集群中配置的推送、链路的监控、服务的治理等等。这些机制是微服务系统运作的基石，如果没有这些机制，那可能要回到ESB的时代，通过企业总线进行服务间的互通。没有单元化，微服务一样可以运行起来，因为底层的功能机制已建立，大部分能力都能在这套机制下运转起来。但以SpringCloud为标准的微服务体系也并非万能，和大多数开源框架一样，其仅针对微服务落地中的关键共性问题提供解决方案，在方案完整性和运维运营层面覆盖是不足的。在异地多活的支持和分布式事务等方面，需要提前在架构规划中设计好对应的解决方案。

单元化架构通过合理的数据切分，把计算与数据进行绑定处理，来控制业务流量单元内闭环，提供了异地多活架构的基础条件。同时提供了大规模系统的标准化管理、扩容、灰度等方案，其更像是一种架构设计思想，而非一种具体技术。这种架构思想也并非只能运用于微服务上，只要有底层原子能力的支撑，也是可以被运用于其他架构上的，只是微服务恰巧提供了这种底层原子能力。

所以，微服务和单元化之间并不是一个包含关系，而是一种加强关系：单元化是微服务架构在宏观架构层面的增强，而微服务是单元化架构在微观机制层面的支撑。

3.2 客户如何选择

标准微服务方案中，技术能力更多采用产品来实现，在技术路线选择上没有单元化方案丰富。但由于方案成熟度高，建设难度和成本都相对较低，建设过程中更多关注业务和应用架构的设计，比较适合体量一般的银行。

单元化方案中，技术能力多采用自研+产品标准能力实现，更关注自主可控，尽可能不去依赖产品自身能力，避免绑定。建设难度和成本都相对更高，更适合体量较大的银行，以及有多活架构规划或追求同业架构先进性的银行。

单元化是微服务架构的强化，两者在核心六大设计要点上存在一些差异，主要集中在数据切分、扩容与路由、交易处理机制、灰度实现机制等能力的实现方式。

目标方案	数据切分策略	技术架构标准	微服务实现策略	业务连续性	研发效能	分布式运维
标准微服务方案	客户号哈希	机房内随机负载 优先同机房调用	以应用软件包内置 领域为服务颗粒度	同城双活+异地灾备 两地三中心	持续集成与发布	标准化指标采集
	分布式分片架构	应用+数据库处理 分布式事务	应用厂商的交付 实施工艺	实例级故障自恢复	快速反馈与迭代	大数据分析告警
	纯数据库扩容	数据库处理聚合查询	应用软件包内置的 技术组件	中心级故障告警+ 人工决策切换	双速开发+自动测试	可视化+自动化
	业务/公共/历史	微服务平台 实现灰度发布		地域级故障告警+ 人工决策切换	过程可视化+ 精益管理	配置与流程管理
单元化方案	客户号自定义路由	单元内随机负载	以业务建模产出 为服务颗粒度	近期：两地三中心 远期：异地多中心	持续集成与发布	标准化指标采集 大数据分析告警
	独立分片架构	自定义就近调用	咨询公司的 IT实施工艺	实例级故障自恢复 单元级故障自恢复	快速反馈与迭代	可视化+自动化
	自定义扩容	应用处理分布式事务	以外购+自研的 形式形成技术组件	中心级故障告警+ 人工决策切换	双速开发+自动测试	配置与流程管理
	业务/公共/历史	全局路由+微服务平台 实现灰度单元		地域级故障告警+ 人工决策切换	过程可视化+ 精益管理	单元化运维管理

我们将分别对标准微服务架构和单元化架构路线的关键设计点进行介绍。

3.3

腾讯云分布式核心 —标准微服务架构路线

3.3.1 腾讯云银行核心标准微服务架构 >>

本节主要从核心应用厂商的视角，介绍业界较为常见的核心系统微服务架构。

微服务架构可以带来像应用解耦、独立自主等诸多优势，但正如上一章提到的，微服务并没有解决所有问题。如服务拆分后的服务编排问题、分布式事务问题，或者数据拆分后的聚合查询问题等。在银行核心如此严谨的领域，微服务的使用一定要综合考虑各种场景，充分听取多方意见，从中找到平衡点。如果一味从理论角度推进，往往得不偿失。

考虑到微服务颗粒过细所带来的整体复杂度，通常在设计时会按照标准的业务领域进行组件的设计和开发，但在发布时会将多个相关的组件进行合并发布。以此来减少组件间的调用链路，提供系统稳定性和性能，简化运维复杂度，包括减少分布式事务的产生。

举例：通常核心会被拆为账务、公共、历史3类微服务群，不同厂商的微服务领域划分会略有不同，结合行内核心系统现状后也会有所调整。



3.3.2 腾讯云银行核心标准微服务架构的设计要点 >>

本章将结合2.4章中提出的六大设计要点，总结标准微服务架构路线的设计要点。

3.3.2.1 数据切分策略

数据切分维度主要以客户号Hash和List两种方式为主。分别从分布式事务出现概率、数据分布均匀程度、访问压力的均匀程度、扩容的难易度等几个维度结合自身业务特性确定。针对数据分散之后带来的聚合查询问题，业界有不同的技术手段来处理，比如把相关数据同步到聚合库或搜索引擎，或者采用分布式数据库产品自带的AP引擎进行处理。

设计切分策略时需要同步考虑未来的扩容策略，针对微服务架构路线，可直接采用分布式数据库自身的扩容能力实现数据层的扩容。无论用哪种方案原则都是在扩容时要尽可能的减少数据迁移的规模。在银行核心场景中，我们建议充分评估未来业务增长量，结合单个物理分片能承载的业务量基线，评估能满足未来5年的规模容量，尽可能避免数据迁移场景的产生。

3.3.2.2 技术架构策略

交易处理策略

技术标准主要以SpringCloud微服务体系的标准为主，服务注册和发现模型是目前分布式系统最主流的模式交互模式。应用实例在启动时均会向注册中心提交自身信息，而注册中心会维护一个全量的服务目录，并下推到所有消费方。服务消费方通过微服务提供的负载均衡组件和服务调用组件实现对目标服务的寻址与点对点的调用能力。这个过程中，注册中心作为非关键交易路径提供服务，提升了该模型的整体稳定性。

分布式事务

分布式事务是微服务化架构在金融等领域的核心挑战，其根源在于业务拆分与数据分片引发的跨节点操作。业务拆分导致单一数据库会话分解为多服务调用，数据分片使单库事务分散至多个数据库。对于后者可采用分布式数据库内置事务机制，前者则需应用层通过SAGA、TCC、AT、事务性消息等模型保障。当前业界方案均存在性能或复杂度代价，需结合场景权衡选择。

分布式事务处理原则遵循“恰好不用解决”理念，优先通过业务设计规避分布式事务产生。典型如银行系统可按客户维度分库，单客户交易通过本地事务即可完成，显著降低跨库事务需求。

扩容策略

在微服务架构下，因为有注册中心的存在，新的应用实例在启动时会自动注册到注册中心，注册中心会更新服务目录并推送到服务消费方，随后消费方在新的服务目录中负载时就会将一部分流量随机到新的实例上，以此来实现应用集群的扩容。数据库的扩容则基本依托分布式数据库能力实现。

灰度发布

灰度发布基本依托于微服务平台内置的灰度能力，其底层原理是在应用实例上打上版本标记，通过微服务平台的标记路由能力来实现分流。标记路由也是单元化架构实现的原子能力之一。

3.3.2.3 微服务实现策略

服务颗粒度

微服务体系帮我们解决了分布式系统落地的主要难点，但服务颗粒度始终是设计关键：过粗弱化敏捷优势，过细则加剧分布式事务风险与运维压力。在标准微服务架构中，基本以ISV核心系统提供的服务领域及颗粒度为基准，结合行内业务现状与规划进行微调。

实施工艺

指新核心的实施交付方式，标准微服务架构中，以ISV核心系统的交付工艺为准，管理方面采用PMO项目群管理模式，以大瀑布+小敏捷融合的模式进行。

分布式技术组件

用于对应应用层屏蔽分布式底层技术细节的组件合集，以ISV核心系统内置的技术平台组件为基础，其中包含了服务编排、分布式锁、批量引擎等分布式技术组件。通常这层组件的设计需要保持良好的南北向接口，建立适配层隔离底层产品差异，来降低产品依赖性。

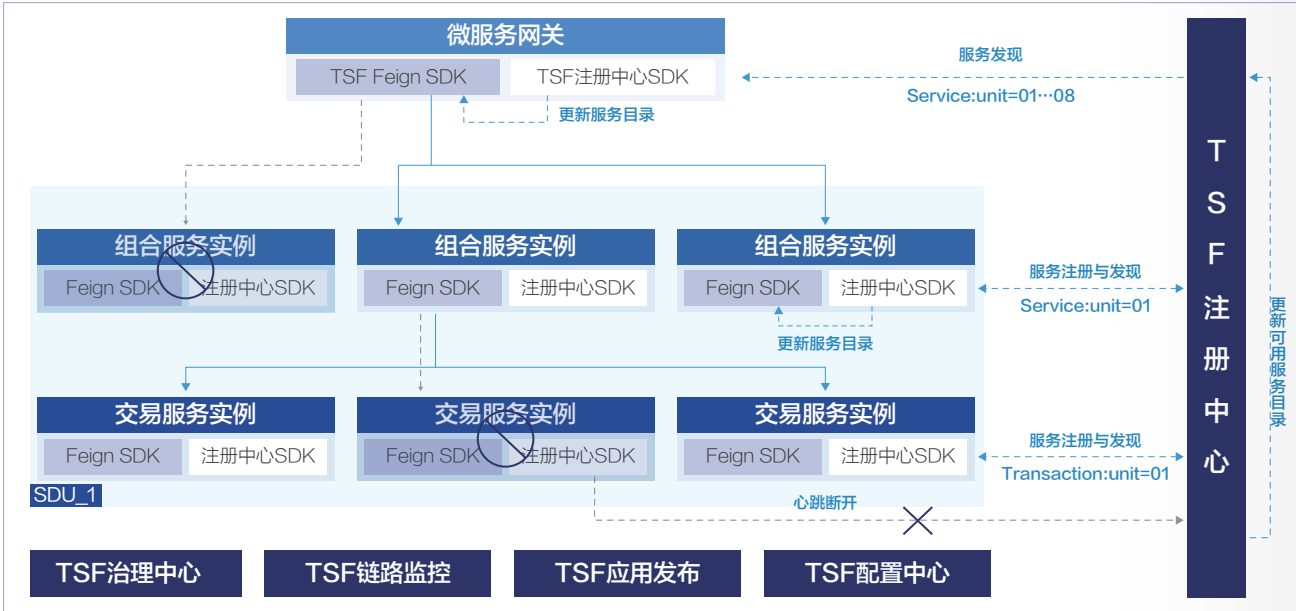
3.3.2.4 业务连续性

在分布式核心领域，业务连续性是个重要的考量，银行客户对于核心稳定性及故障后的自愈、快速恢复能力，有着较高的要求。腾讯从多个分布式核心落地案例中总结设计原则如下：



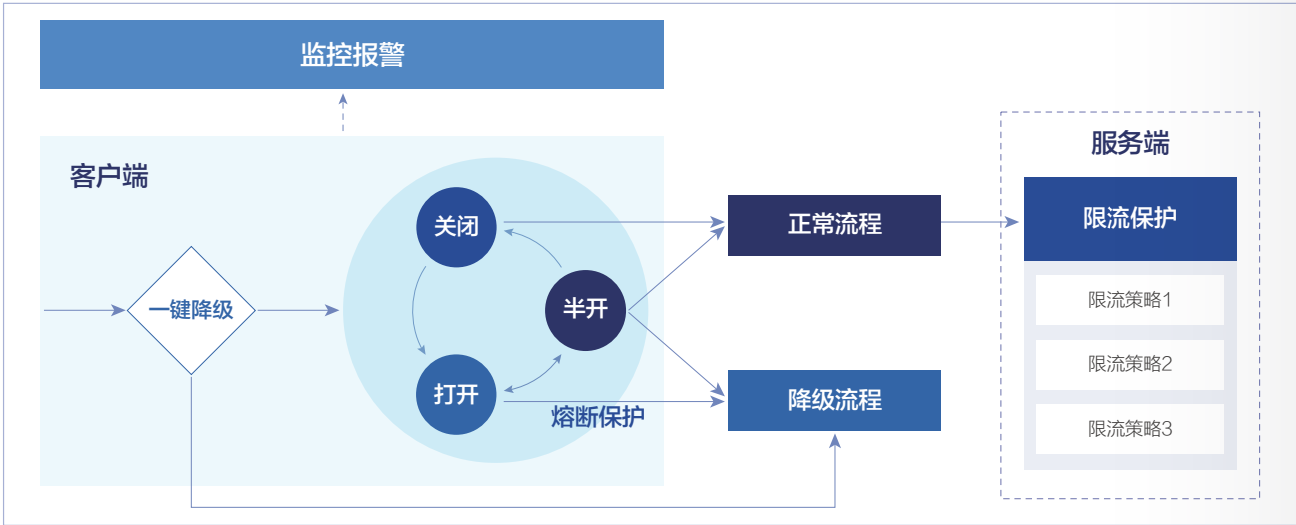
业务连续性包括数据中心内、同城及跨城市的连续性，下面将一一介绍。

数据中心内故障隔离



在分布式架构下，实例级故障自动隔离通过服务注册中心的高可用机制实现：实例启动时自动注册并实时同步服务目录；注册中心实时推送节点状态变更。当探测到节点心跳超时，系统秒级标记异常并触发服务列表全局刷新。调用方基于更新的服务列表自动剔除失效节点，配合负载均衡策略无缝完成故障隔离。

除此之外，服务治理体系也是业务连续性保障不可或缺的机制。其中就包括一系列的熔断、降级和限流的措施：



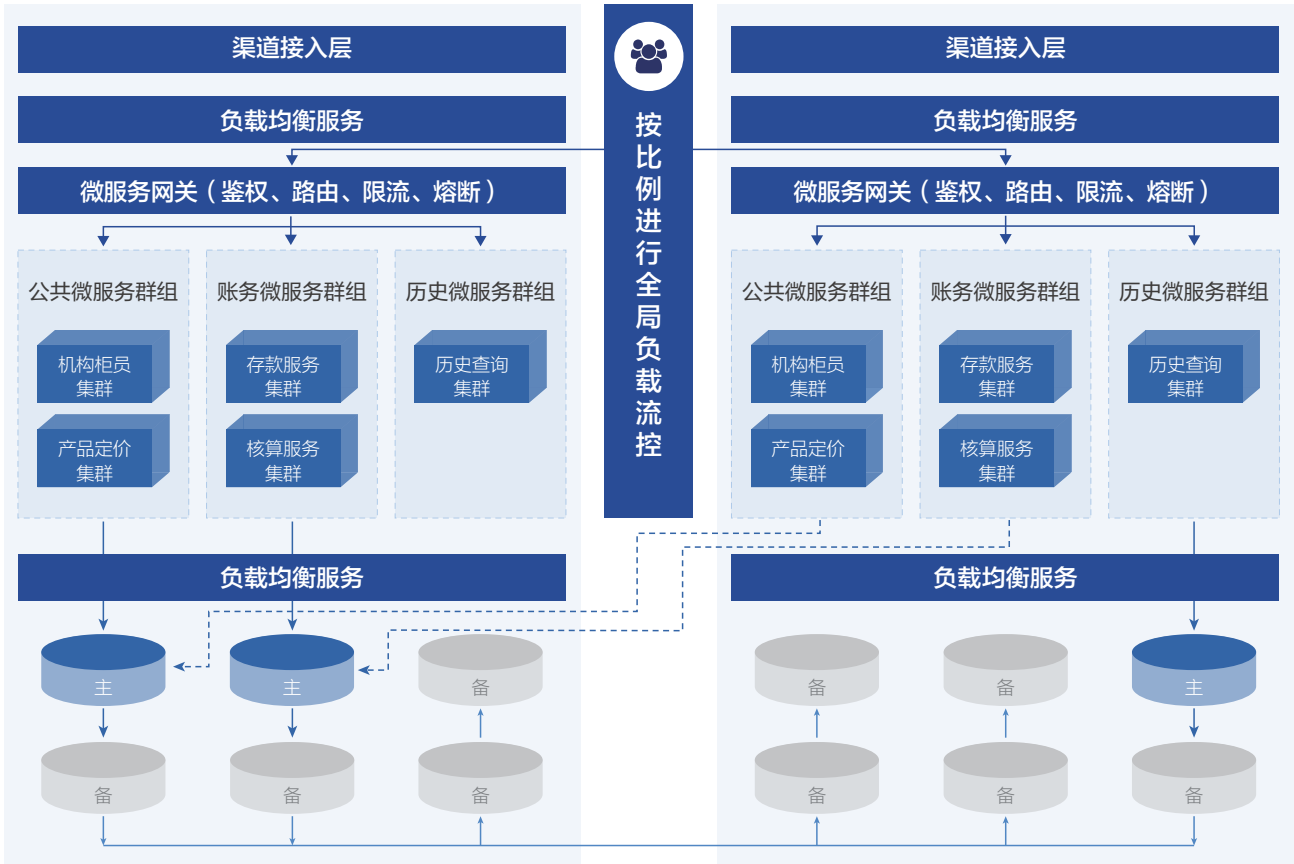
在业务规划时，提前识别强弱依赖业务，当流量激增时，通过平台一键限流让弱依赖业务直接进入降级，不发生网络调用，节省资源消耗，保证强依赖业务。客户端需要具备“一键降级”功能，即交易维度的黑名单，在黑名单中的交易直接进入“降级流程”。

客户端集成熔断组件后，可针对服务端的响应超时和异常率高的接口提供“半开”和“全开”的流量控制，被熔断的流量直接进入”降级流程“。

服务端则通过集成限流组件，提供针对 QPS 和线程数的限流控制，被限流后直接进入客户端“降级流程”。建议提供集群流控 + 单机流控兜底，可以发挥更好的限流效果。

同城高可用

在同城两个数据中心中，标准架构中数据库采用分布式实例，对上层应用屏蔽底层数据库结构，应用只需连接本机房的数据库代理进行访问。

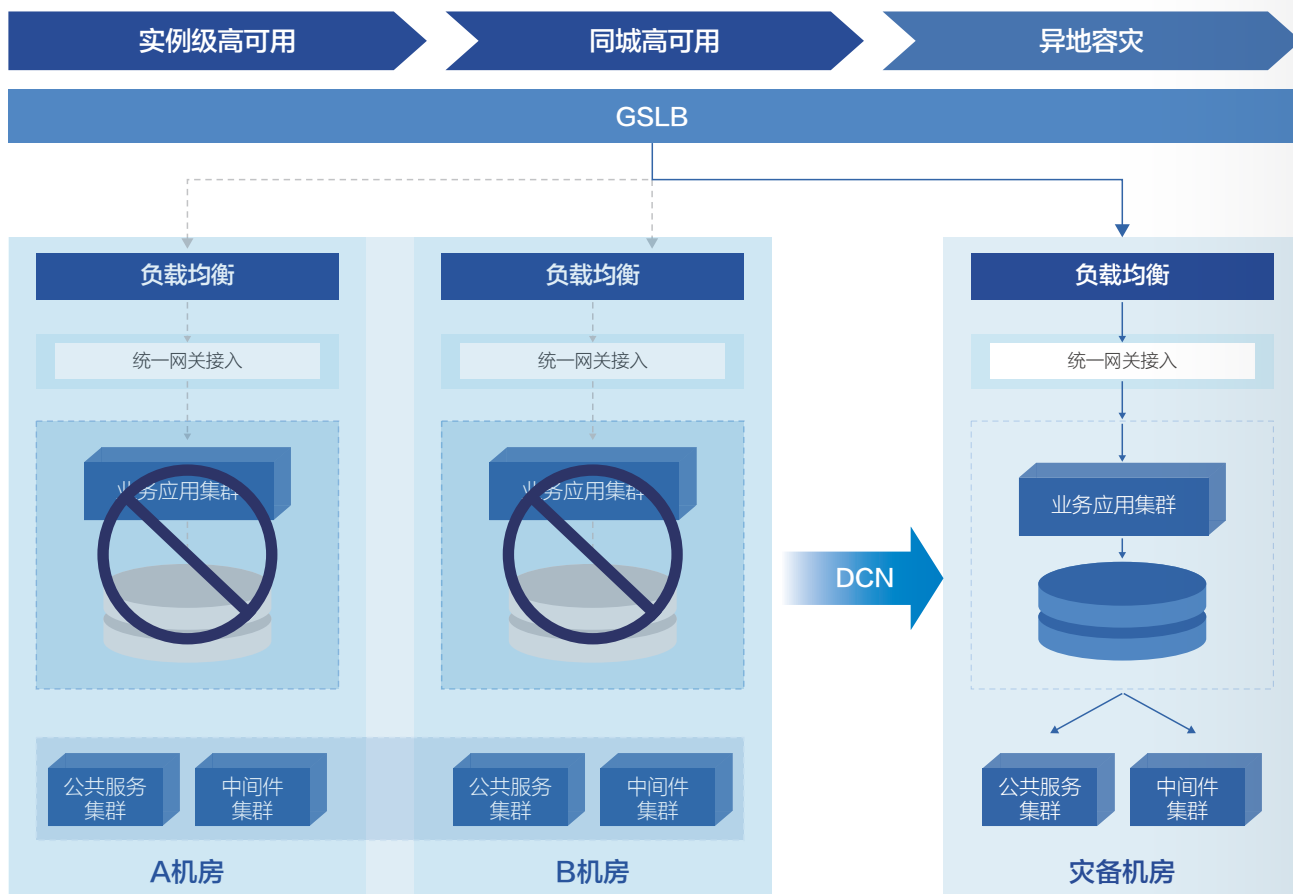


如图，流量通过顶层接入，经过 DNS 到中心内负载均衡服务再到微服务网关，经过鉴权、限流等一些列检测后进到微服务业务集群，处理时两个中心的应用都连接本机房的数据库代理进行访问数据，但在数据库代理这一层会根据路由算法计算客户所在的数据主备位置，并发起转发，这个过程同城机房的请求将出现横向流量访问数据主本，这个过程应用是不可见的，因为分布式数据库对上屏蔽了底层的数据库架构和数据路由。以此实现同城两个数据中心同时处理业务请求的双活架构。

当下基础设施在同城之间的数据传输和稳定性已经非常成熟，对于大多数系统组建这样的分布式架构是完全没有问题的。只有在两个机房间超过一定距离，或延时过高时才需要考虑横向流量的治理问题，那时单元化架构是一种好的解决方案。

异地容灾

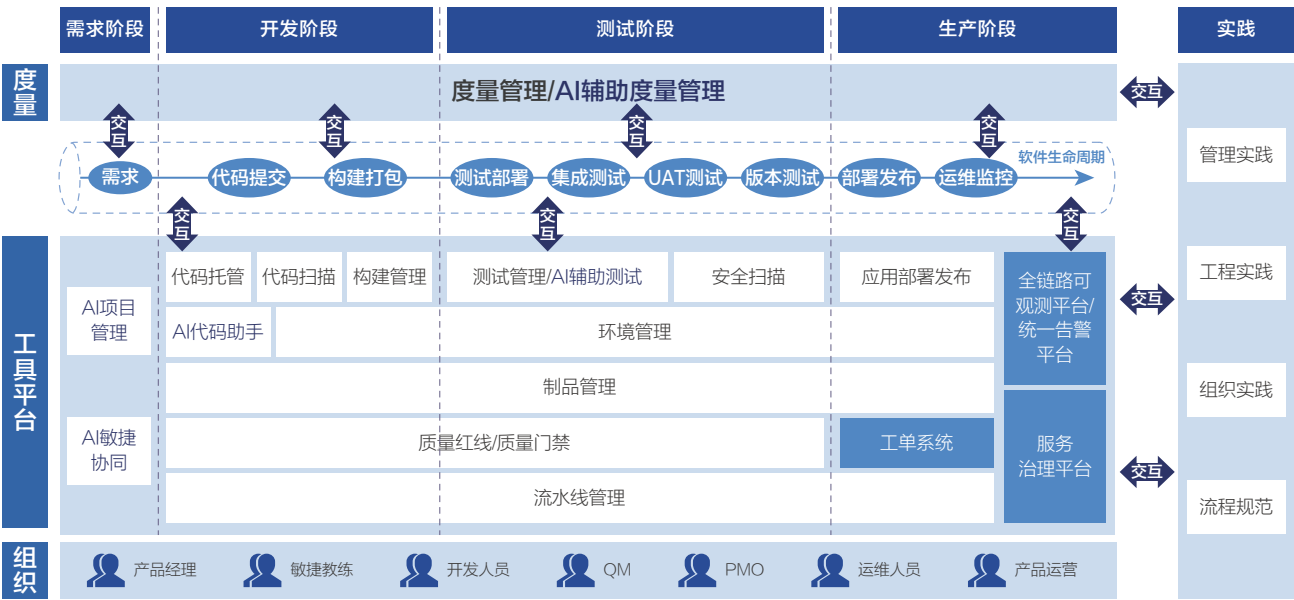
在分布式应用系统中，应用的无状态属性使得扩容和迁移都变得方便许多，我们只需要考虑有状态服务，站在核心系统视角主要是数据库。数据库的 RTO 和 RPO 是整个核心做异地切换的主要关键指标。在大型复杂的银行核心系统中，会涉及到多个业务数据库，且每个数据库下又会有多个数据分片，由于异地间的数据传输是异步模式，其切换过程比同城切换要复杂许多，会涉及网络状况的判断，数据补偿与校验等。



当数据库切换完成后，随后启动内部校验，确认无误后，在全局负载均衡服务（GSLB）中将流量指向异地的容灾中心，实现地域级故障的业务连续性保障能力。整体异地容灾切换的流程极为复杂，且与客户的实际运维环境有关，此处仅做概要性说明。

3.3.2.5 研发效能体系

在银行业加速推进新一代分布式核心系统建设的背景下，构建高效的研发效能体系成为破局关键。传统组织架构因职能划分形成的“筒仓效应”，导致跨部门协作存在流程冗长、资源协调低效等问题，难以应对业务快速迭代与系统高质量交付的双重挑战。研发效能平台需以全链路协同为核心，打破工具链孤岛，贯穿需求 - 开发 - 测试 - 部署全生命周期：通过自动化流水线实现代码扫描、制品管理和一键发布，结合敏捷工作流支撑核心与外围系统的高效协作；依托数据看板构建多维度度量体系（代码质量、项目进度、资源负载），驱动管理决策科学化；建立快速反馈机制持续优化流程瓶颈，降低各环节冗余损耗，最终形成“需求响应 - 价值交付 - 效能反馈”的闭环，实现复杂系统研发的精益化与可持续性。



此外，伴随AI大模型的快速发展，我们在现有DevOps工具链及技术平台基础上深度集成AI大模型技术，形成新一代DevOps平台，通过在项目管理、敏捷协同工具、代码助手、测试管理等工具与平台上融合AI大模型技术，有效提升了产品的需求、开发、测试的效率。

- AI 辅助需求分析，通过自动化处理与智能推荐机制降低人工失误，有效提升需求文档的产出效率与准确性；
- AI 日常工作进度跟进，通过实时追踪任务状态与自动预警机制，实现研发过程可视化与迭代效率提升；
- AI 代码助手，提供基于上下文感知的代码补全、支持智能技术问答与自动化代码评审能力；
- AI 辅助测试，基于需求文档自动生成测试场景及可执行的测试用例脚本；
- AI 度量管理，通过自动化生成数据洞察报告，为研发流程优化和效能提升提供决策支持。

3.3.2.6 分布式运维体系

在银行业分布式核心系统建设中，稳定性是保障金融级高可用服务的核心命题。分布式架构在支撑业务敏捷与弹性扩展的同时，面临三大核心挑战：**复杂故障定位难**（节点爆炸性增长导致根因追溯效率低下）、**数据强一致性要求高**（跨中心多活场景下的账务精准性保障）、**业务连续性压力大**（高频迭代与分布式技术叠加的运维风险）。为此，需构建与分布式架构深度适配的运维体系，聚焦以下能力建设：

全链路可观测能力

建立覆盖基础设施、中间件、业务组件的立体监控网络，实现交易链路追踪精度达毫秒级。

精细化配置管理

构建跨中心集群的统一配置中心，支持灰度发布、流量调度策略联动，结合资金业务时段特征设计分级发布机制，保障核心交易变更过程零中断。

标准化流程引擎

基于 ITIL 规范构建自动化事件管理平台，实现故障响应 SLA 数字化管控，打通研发、测试、运维全流程数据，提升跨团队协同效率。

工程化稳定性体系

将混沌工程纳入持续交付体系，通过模拟网络分区、节点宕机等极端场景，验证分布式事务、幂等控制等核心组件的容错能力。

建议在核心系统设计阶段同步规划运维中台，重点强化数据一致性、容灾切换验证等金融专属模块，通过运维数据与业务指标的多维关联分析，构建覆盖事前防御、事中处置、事后优化的全生命周期护盾，为账户体系安全、账务一致性等核心诉求提供底层保障，筑牢银行业务连续运行的数字化根基。

3.4

腾讯云分布式核心 —单元化架构路线

3.4.1 什么是单元化架构 >>

所谓单元即把一部分计算资源和一部分数据资源进行逻辑上绑定，形成一个标准化的处理单元。我们将一个大型系统拆解成多个标准的处理单元，每个处理单元具备完整的业务能力，但只处理全量数据中的一部分，简单理解一个单元就相当于一个小分行。

在腾讯“海量服务之道”提出的 SET 模型部署，就是腾讯在早期处理海量互联网业务的一种“单元化实践”。在 2015 年支持微众银行建设核心业务系统的过程中，创造性的提出按 DCN（数据中心单元）的设计理念，按照业务和用户维度进行水平拆分，将银行核心数据库账户体系、业务交易处理打包在一个 DCN 单元中（含服务器 / 数据库），在前端通过业务全局路由实现业务转发，这也是比较早期的银行核心单元化架构实践案例。

3.4.2 单元化的价值 >>

单元化架构的核心价值主要用于解决在大型分布式系统中面临的如下挑战：



解决传统数据库连接数上限问题

应用的无状态化可以实现横向扩容，而每个扩容出来的实例都会对数据库产生一定的连接数，连接数消耗着数据库服务器的物理内存。随着业务量的增大，数据库连接数的上限将限制着整个集群的扩容规模。

早期在分布式数据库还没有成熟到能运用到银行核心时，单元化架构主要解决的问题就是单一数据库的连接数问题。其通过将一部分计算资源与一部分数据资源绑定的方式来解决该问题。通常情况下，数据资源体量稳定，其对应的计算资源也相对稳定。计算资源相对稳定，则其对数据库产生的连接数也相对稳定。

解决分布式运维及标准化扩容问题

在扩容方面，复杂应用系统集群扩容时大多没有标准可依，一般分布式系统都是通过监控阈值告警配合人工或自动方式扩容。但在银行核心领域，这种模式相对松散且不可控。以单元维度的标准化扩容对长效运维有很大的帮助，能做到架构上的整齐统一，运维动作标准化，也能通过一个单元的业务量来实现提前量的规划。在运维方面，通过提炼不同的单元特征，定制针对不同类型单元的标准化运维方案，标准化不同单元的运维操作。

跨机房稳定性和性能问题

在微服务集群中，应用是无状态的，这也意味着负载均衡会进行无差别分发。在同城双中心架构中，两个中心的应用都在一个注册中心上，那么网关在往服务层负载时，默认也就会有一半的流量被分发到对方中心的服务上，产生了跨中心的场景。而在应用访问数据库时又要回到主中心访问，由于应用到数据库的访问频次是会被放大，假如这次交易需要操作50次数据库，那就有50次跨中心的访问，这极大影响了系统的稳定性和性能。站在银行核心系统角度是要尽可能减少不确定性，即在设计层面需要减少这种场景的产生。因此，单元化的流量闭环特性可以较好地满足这个场景。

异地多活问题

异地多活是一个极富挑战的课题，也是目前最为前沿的架构之一。其对底层基础设施、数据库、中间件，以及我们的应用架构都提出了很高的要求。上面提到的稳定性和性能问题在异地的场景下会被无限放大，一个系统跨异地部署，且内部流量在异地间不受控的负载时，系统的性能和稳定性等指标都是不可控且无法预测的。因此，单元化架构也是实现异地多活架构的必要条件之一。

3.4.3 单元化落地挑战 >>

单元化架构的落地也存在着诸多挑战：



海量映射关系维护

与互联网银行不同，传统银行的应用系统在规划设计时并没有以客户为中心进行设计，这导致行内大量的系统，包括核心的交易接口中并没有客户号这个字段。基本都是以卡号、账号或身份三要素等信息为主，这就导致新核心建设时需要在入口层进行统一的业务要素转换，通常在新核心内部统一用客户号作为单一客户识别要素。然而，客户的关联识别要素（如卡号、账号、手机号、证件号、存折号等）数量庞大且不断累积。举个例子，假设一个拥有 1 亿客户的核心系统，每个客户有 10 个要素字段的话，那么这个映射关系的总数据量就要接近 10 亿。好在是相对简单的 K,V 存储，落地时可以采用缓存集群或单体数据库落地。基于缓存的性能表现较好但设计难度较大，需要考虑到客户开销户场景时要及时更新映射关系并同步到缓存，以及同步失败时的兜底方案。

实现流量打标与路由

流量打标是实现单元化的核心能力。首先，应用实例在启动时就需要带上单元标签到注册中心注册，其次负载均衡时需要优先过滤服务目录中的标签，再执行负载策略。在设计规划阶段，要设计好单元标签等相关字段，注册中心也需要支持元数据注册能力，再通过封装通信组件解析遍历约定好的标签来实现实例的过滤。

路由一致性

单元是将计算资源与数据资源进行逻辑绑定从而提供服务的，那么从网关到应用层到数据库存在着至少两次路由，这两次路由的结果必须保持一致，才能保证流量闭环在这个逻辑单元内部。也就是说，无论采用数据访问组件的方式还是采用分布式数据库自身的路由，都必须和网关层的路由计算结果保持一致。

保障业务连续性

全局路由一般发生在网关层，一旦出现任何问题都将影响后续整个链路。所有和路由相关的操作都需要考虑对应的兜底机制，比如失败转异步，或者寻址失败就随机分发，但要求数据层能具备路由修正的能力。

高可用方面除了要考虑实例级、机房级、地域级容灾之外，还需要增加对单元级高可用的设计。并将单元灾备关系及属性在系统上下文中保持传递，确保在任何路由环节都能在当前单元故障时找到容灾单元。

单元化的可观测性

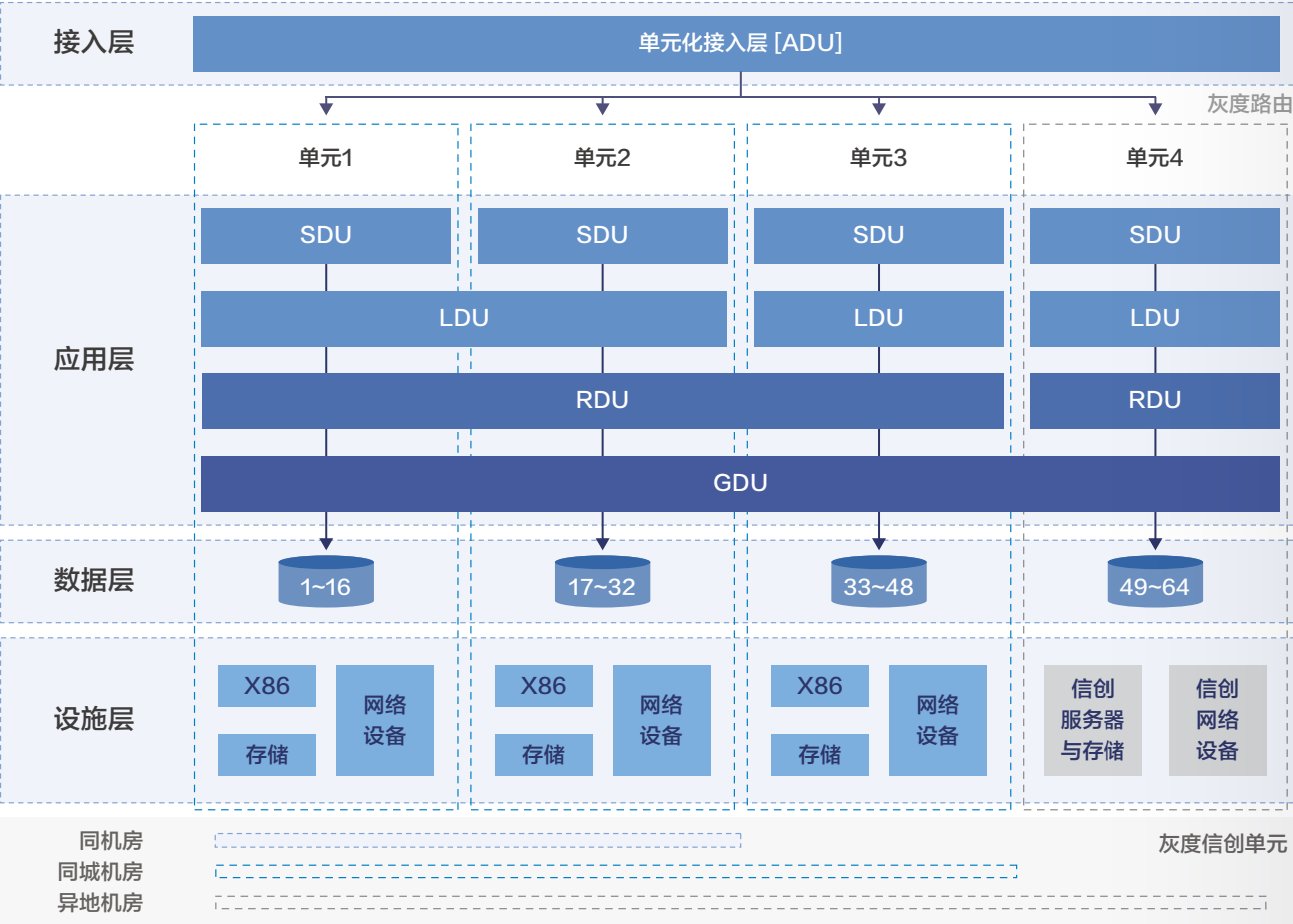
为了更好地维护这套机制，需要接入行内的统一监控、运维与告警体系。在传统监控中加入跨单元和跨机房调用的路由，通过采集相关埋点数据，对比分析后可可视化展现。以及系统内的全链路监控体系，将单元化的观测性接入全行链路体系中，实现一体化监控。

3.4.4 腾讯云分布式核心—单元化架构解决方案 >>

随着银行业分布式转型的深入应用，单元化架构作为代表先进架构的技术路径快速崛起。2015 年微众银行率先实现单元化架构在核心系统的成功落地，自此该架构凭借其弹性扩展和容灾能力优势，在金融领域实现规模化推广。

3.4.4.1 腾讯云单元化架构（TCUA）

腾讯通过自身参与的微众银行、国有大行和头部商业银行等各大核心系统设计中归纳提炼形成符合银行核心系统特点的单元化架构体系，TCUA(Tencent Cloud Unitization Architecture)。



腾讯单元化架构是一种针对银行分布式核心系统技术架构的体系化、分层次、分领域的架构思想。接入层负责接收流量，识别流量，转发流量。识别后的流量被转发至应用层对应单元处理，单元默认按照客户维度拆分，单客户交易实现单元内闭环处理。跨客户交易存在一定比例的跨单元协同处理。数据会切成若干份（如图所示切分为 64 份，可根据实际情况调整）均匀的放置在若干个单元内。此外，可通过灰度机制将指定标签的客户放置在灰度单元实现生产灰度运行或验证。

站在整体核心系统应用视角，根据应用服务特征归纳形成多种单元类型。将单元以业务处理顺序的维度进行分解，形成：接入单元 (ADU) 负责接入接入能力、标准处理单元 (SDU) 负责业务处理能力、本地单元 (LDU) 和同城单元 (RDU) 分别用于提供单 AZ 服务能力和同城共享服务能力、以及在异地多活架构中会涉及的全局类型服务则由全局单元 (GDU) 提供。

3.4.4.2 腾讯云单元化架构的设计要点

数据切分策略

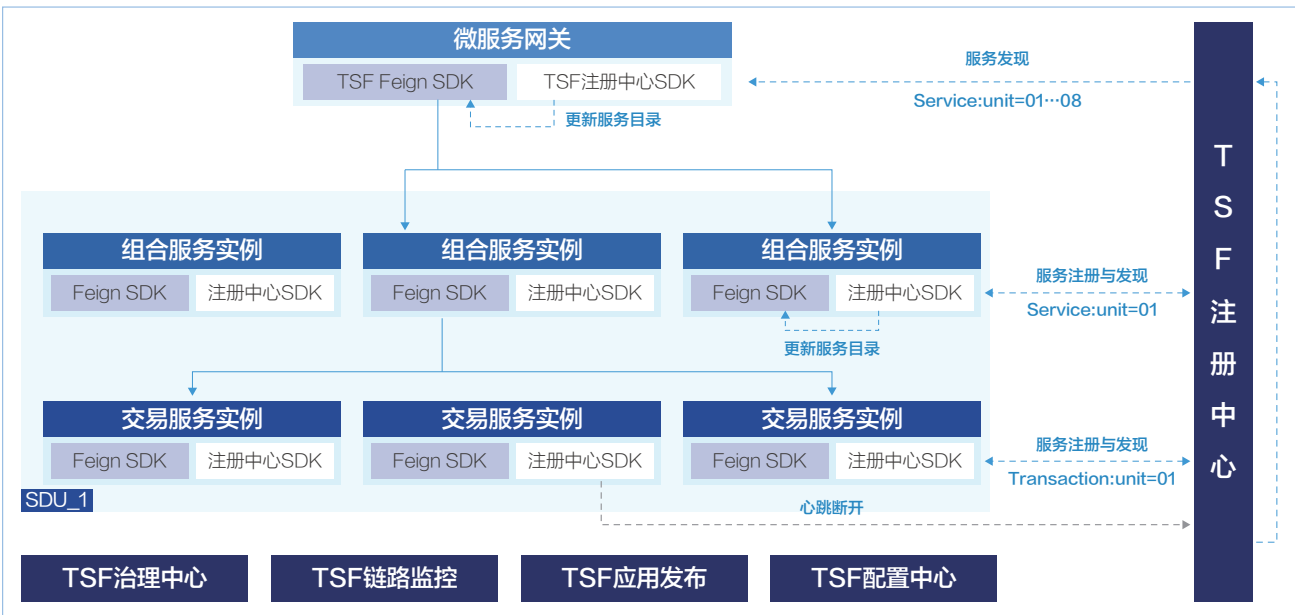
单元化架构下，数据分片的维度、分布规则都与微服务架构相似，主要以客户号 Hash 和 List 两种方式为主。同样需要考虑分布式事务出现概率、数据分布均匀程度、访问压力的均匀程度、扩容的难易度等几个方面。不同的是单元化架构多以应用侧主导数据分片策略及路由策略。这种方式能摆脱对分布式数据库或中间件产品的依赖，灵活可控是单元化架构的重要优势。

扩容策略上与标准微服务架构有所不同，单元化架构中会将全量数据分为若干份，每一份称为一个逻辑分片，然后建立逻辑分片到物理分片和单元的映射关系。扩容时，首先加入新的物理分片，再将指定逻辑分片迁移到新的物理分片中，最后更新逻辑分片到物理分片的映射关系。这个扩容的逻辑通常是由应用侧结合数据迁移工具来做定制化开发，也可以通过 TDSQL 的扩容功能实现。在核心场景中，原则上要尽可能减少数据层的扩容，其次是扩容时要尽可能减少数据迁移的规模。

技术架构标准

交易处理策略

单元化架构下交易处理策略同样以 SpringCloud 的注册发现体系为基础，但在调用与负载时需要遵循以单元维度和就近调用原则，即：所有路由均需要判断单元维度，识别单元内调用还是跨单元调用，对于相同服务的调用顺序为：优先单元内调用，其次本中心调用，最后同城中心调用。



单元化架构会要求所有应用在往注册中心注册时需要带上自身所在单元的标记信息，同时微服务之间在调用时需要先根据单元标记过滤目标服务的实例，确保在指定单元的服务实例之间进行负载均衡算法。这种根据单元标记进行路由和目标寻址的逻辑在标准微服务架构中是不存在的。

分布式事务策略

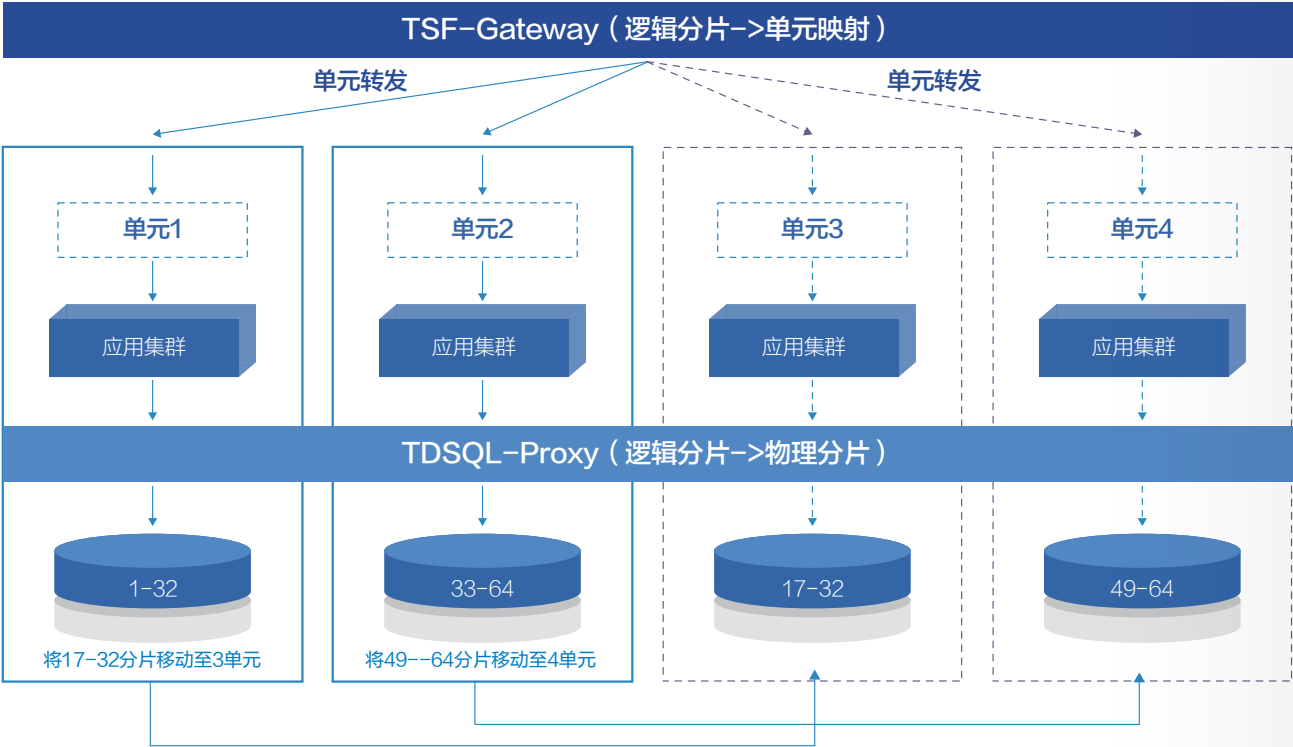
分布式事务处理策略与微服务架构保持一致，请参考 3.3.2.2 章节。

单元扩容策略

在分布式系统中，应用的无状态化让计算资源的扩容变得容易，但在银行核心场景中就显得不规范和不受控，而单元化架构则提供了一种分布式系统下计算和数据资源扩容的标准规范。在上一节中介绍了已经介绍了数据层的扩容逻辑，本节结合单元扩容进行整体扩容说明。先增加新的单元标记，再增加新的计算资源和数据库物理分片，其次执行上一节中提到的将指定逻辑分片迁移到新的物理分片，然

后修改这些逻辑分片到物理分片及单元的映射关系。在更新生效后，新的逻辑分片中的客户再次访问时会被路由到新的单元中进行业务处理。每个处理单元的业务量是可估算的，每次扩容的资源也是可以提前预判的，这样扩容的机制就可以提前规划，这种标准化正是银行系统治理工作所需。

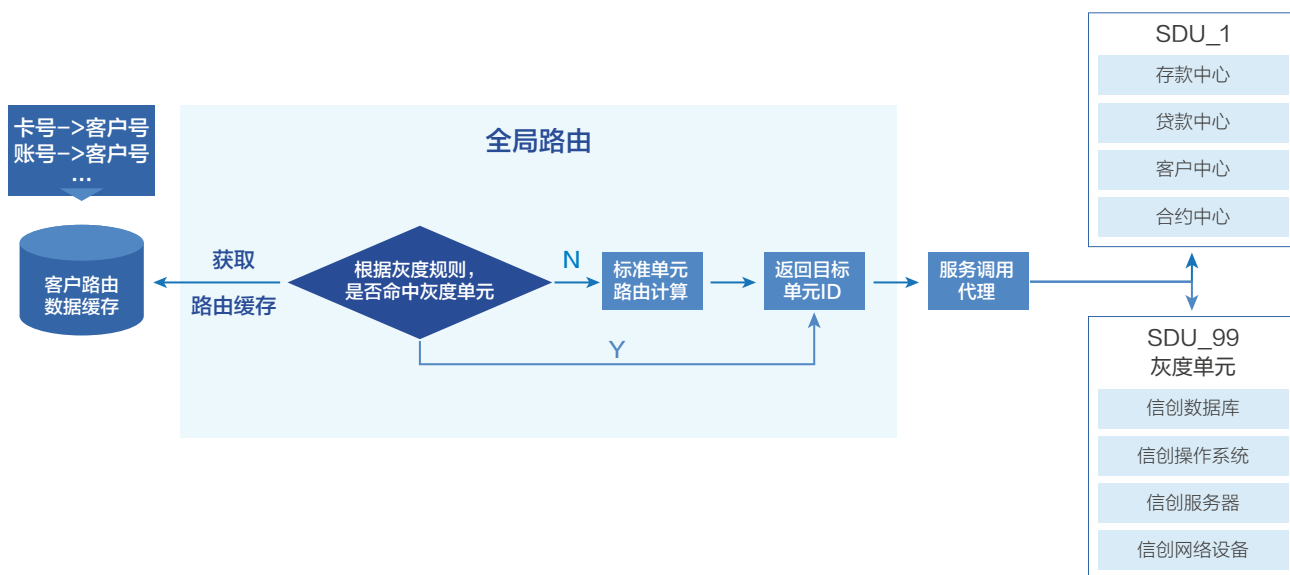
举例说明：假设以客户维度切分全量数据，当前有两个单元，每个单元中各一个物理分片，每个物理分片中有 32 个逻辑分片。其中 1 号逻辑分片原本在 1 单元的 1 号物理分片中，由于识别到其为热点分片，现要将其迁移至一个新的 3 号单元的 3 号物理分片。那么首先要将 1 号逻辑分片中的所有数据迁移至 3 号物理分片，再修改 1 号逻辑分片到 3 号物理分片的映射关系以及 1 号逻辑分片到 3 号单元的映射关系。当更新生效后，1 号逻辑分片中的客户再发起交易时，网关就会将其路由到 3 号单元，其中的应用会从 3 号物理分片中获取数据进行处理。通过这种方式解决扩容问题。下图展示了从 2 个单元扩容到 4 个单元的过程。



灰度机制

在标准微服务架构中，灰度发布主要依靠微服务框架能力实现。其通过应用之间版本隔离，流量能按照版本标记发送到指定的应用版本集群中，通过流量比例的控制，来实现生产流量中的一部分请求由灰度的应用集群处理。这节我们介绍下如何利用单元化的架构，进一步优化灰度发布的机制。

首先利用单元的隔离能力，为灰度目标单独设定 1-2 个灰度单元。其次，确定灰度维度有哪些。比如常见的有按指定客户号或者客户标签维度，例如把行员打上标签后加入灰度表，还有可以按照逻辑分片，把逻辑分片中的某几个分片调度到灰度单元等等。明确了灰度维度之后，进行逻辑设计。在网关进行单元路由计算之前，优先查询灰度表，如果当前客户的 ID、标签或分片 ID 中任一维度被命中则直接按照表中定义好的单元进行路由，跳过标准的单元路由计算步骤。逻辑上并不复杂，只是在配套的能力支撑上要做好充分设计。



通过单元标记进行灰度可以充分发挥单元化架构的隔离特性，不仅是应用版本，包括计算资源和数据库都会一并纳入灰度的范畴，是更具标准化和完整性的灰度方案。

微服务实现策略

服务颗粒度、实施工艺和分布式技术组件概念已经在微服务架构中介绍，此处就单元化架构与标准微服务架构的在微服务实现策略上的差异性进行介绍。

微服务实现策略主要包含以下几个维度：

服务颗粒度

在19年邮储银行启动新一代核心项目之前，在分布式核心领域业界尚没有明确的、清晰的方法论指导银行核心的微服务颗粒度。当时已经落地的几家分布式核心中，多以专家经验推导的方式为主。邮储银行采用业务建模方法，形成了统一的、标准化的五级建模，结合IT实施工艺来指导应用层微服务组件的颗粒度，随后逐步在整个行业受到关注。业务建模是一套优秀的企业建模方法论，但由于其实施成本相对较高，主要还是集中在国有大行和头部城商行之中使用。也有科技实力较强的商业银行结合自身技术特点，以改良的DDD设计模式作为分布式核心的应用设计方法。除此之外，更多的是以核心应用厂商提供的服务颗粒度为基准，结合各家行的自身业务与技术特点，有针对性地进行调整后形成的微服务颗粒度。

实施工艺

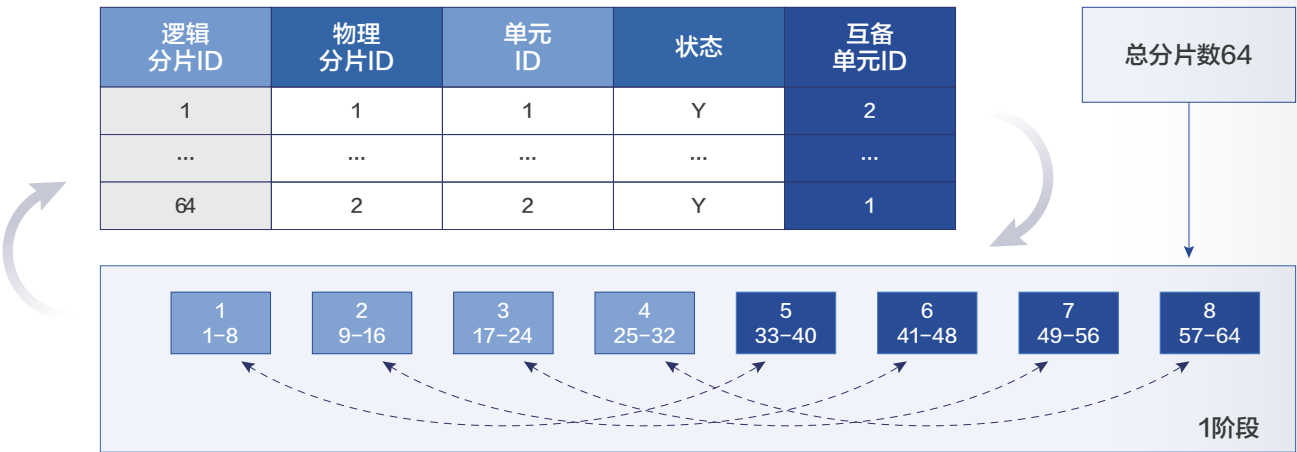
对于采用业务建模的各家大行，其对应的IT实施工艺则很好地适配了业务建模的产出，能通过识别->定义->分解->细化等工序把业务组件转化到微服务组件，其中包括细化后的构件，一个构件作为一个最小的开发任务，可用于工作量分配和质量跟踪。通过构件组装形成可对外提供最小业务能力的基础服务，通过可视化的服务编排能力，快速组装基础服务形成组合服务。实现稳态和敏态分离的双速模式，适应未来的敏态业务和快速发布。

分布式技术组件

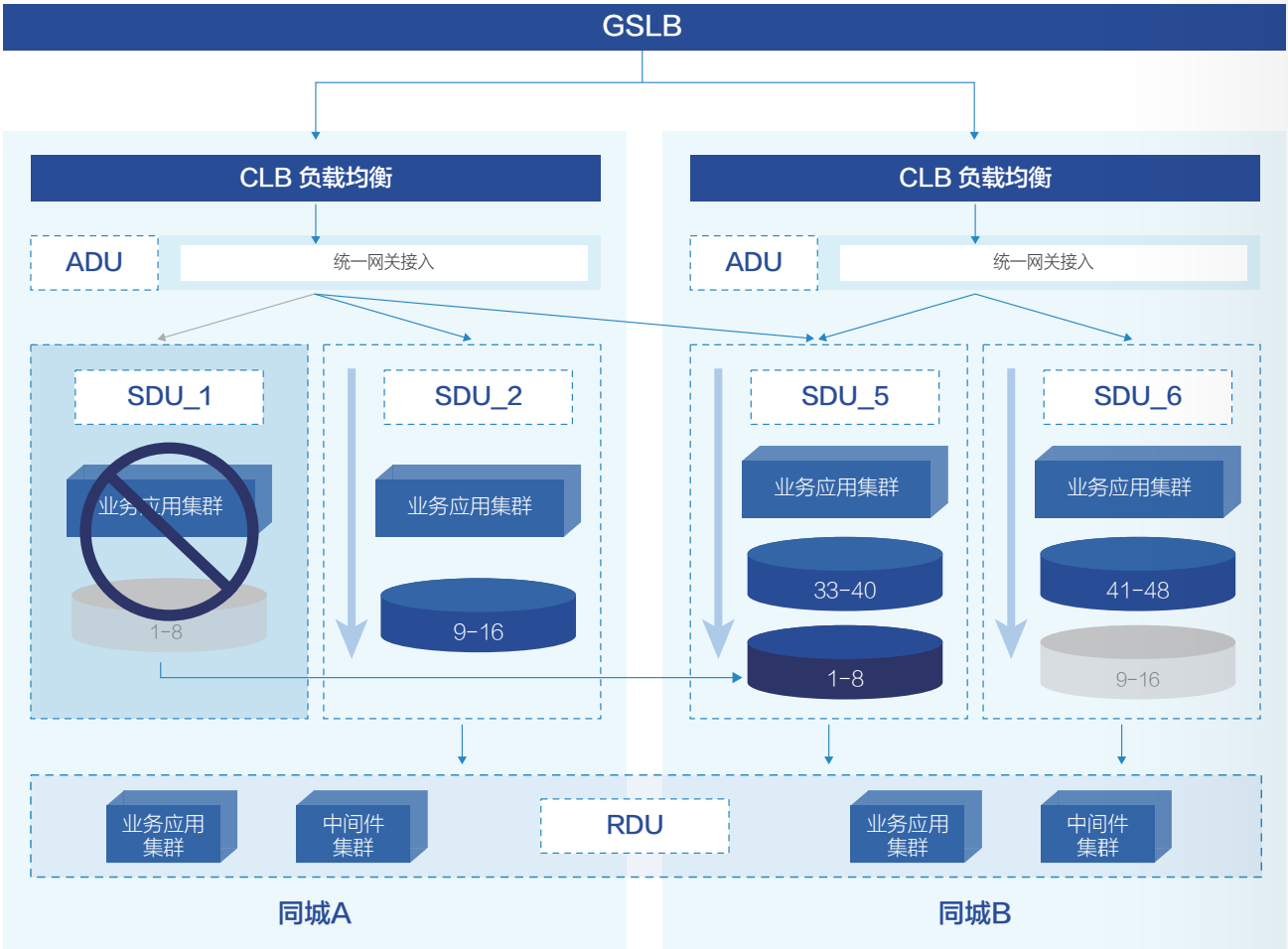
用于对应应用层屏蔽分布式底层技术细节的组件合集，以核心应用厂商的软件包自带的组件为主。但这一层在业界并未有统一标准，各家厂商的实现方式与功能范围和边界有所不同。常见的组件包括了联机交易和批处理引擎、分布式任务调度组件、分布式流水号组件、分布式锁组件、数据访问组件、服务编排组件等。个别组件与PaaS平台存在重叠，如：微服务框架、分布式事务组件等。在规划阶段需要明确双方职责边界，在设计阶段确认组件间的集成机制。

业务连续性

在标准微服务架构中已介绍数据中心内部、同城以及异地容灾的业务连续性。本章将重点介绍单元化架构下的单元级业务连续性设计。

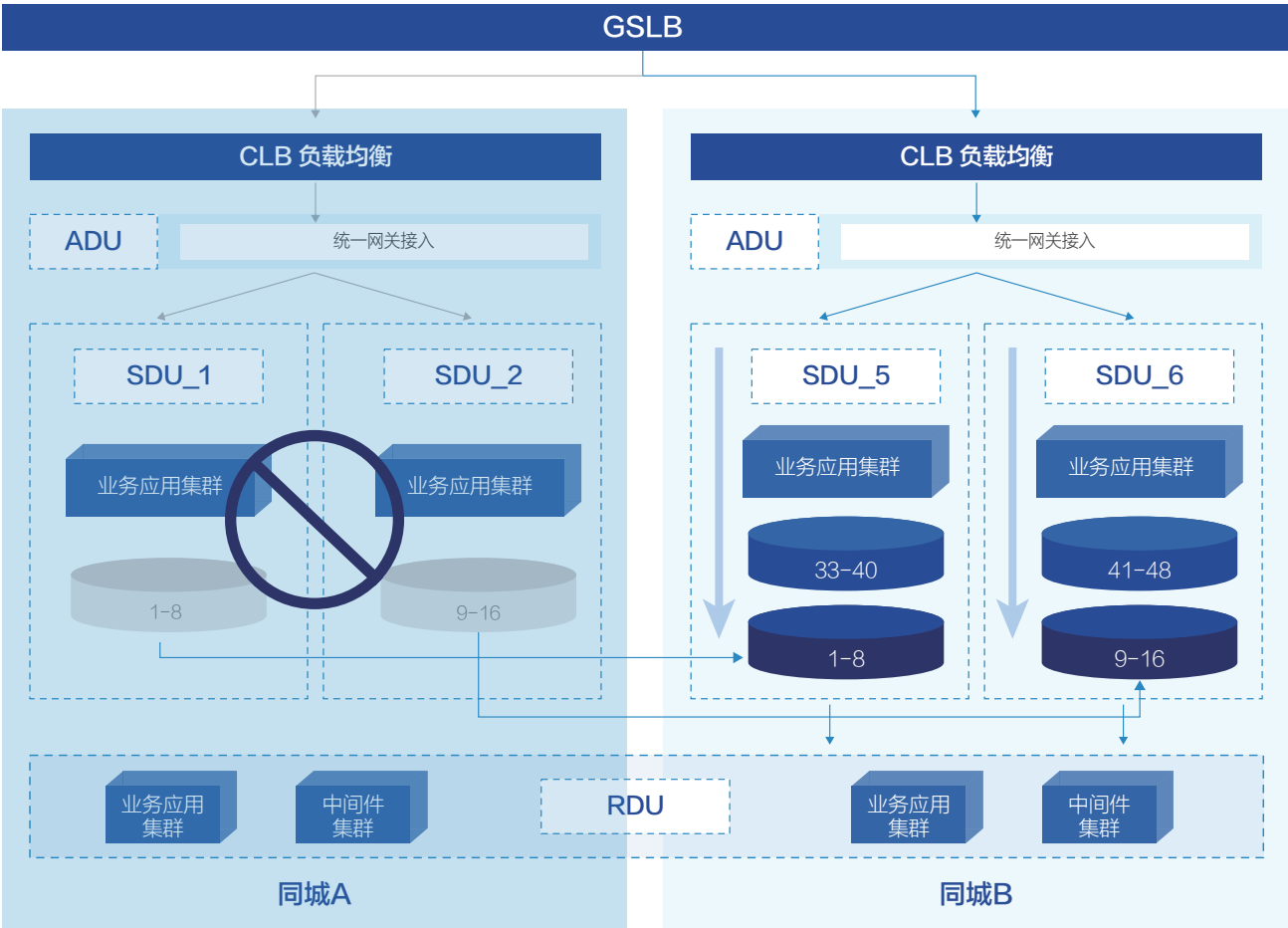


通过设计“互备单元表”来建立两个单元间的互备关系。当其中一方出现故障时,通过更新状态,来指向其互备单元 ID 从而进行路由调整。实现单元间互备要求单元的数据库副本需要放置在对方单元中,当发生切换时,数据库会将对方单元下的副本调整为主本。



如图，请求通过 GSLB 负载到同城两个机房，机房内部通过负载均衡服务转发到本中心的微服务网关集群，在网关这一层识别请求对应的单元和路由信息，转发到指定单元处理。当单元 1 发生故障时，切断单元 1 的流量，并获取单元 1 对应的互备单元信息（单元 5），等待数据库主备切换完成，更新全局路由将流量转发至单元 5。此时单元 5 除了承载自身业务流量还会承载单元 1 的业务流量。单元切换的指标绝大部分情况下取决于数据库的 RPO 和 RTO。回切过程类似，先关停单元 1 的流量，执行主备切换，让主库回到原单元 1 下，调整网关层路由回到单元 1。

单元化架构在同城高可用设计上与标准微服务架构有所不同，微服务架构因为没有单元概念，在故障发生时是直接将整个数据库切到同城，再切换 GSLB 的流量。而单元化架构则是以单元维度进行切换，虽然最终效果是一样的，但是内部运作机制不同，切换控制的颗粒度也不同。



对于高可用架构，最基本的要求便是应用需要实现无状态、能支持被调度。故障发生时，数据库会先切换到同城 B，其次更新 ADU 中路由映射关系：如图所示，将 1-8 分片的客户转入单元 5，将 9-16 分片的客户转入单元 6，最后在域名解析层将流量转移到同城 B。此时请求都被转移到同城 B，在 ADU 这一层再细分请求到单元 5 和 6，这两个单元承载了自身业务的同时，还承载了单元 1 和 2 的业务请求。至此实现了同城接管故障机房的高可用机制。

其它

在研发效能体系上，单元化架构中需要考虑按单元化的应用部署即支持按某个单元进行应用部署，其他能力与微服务架构保持一致，请参考 3.3.2.5 章节。

在分布式运维体系上，单元化架构中需要考虑如何保障整个单元化的运行，从单元发布、流量控制、单元切换，以及按单元视角的资源监控与汇总、跨单元的链路监控、跨单元流量监控等，包括整体的运维动作，需要形成一个完整的分布式运维体系。其他方面，单元化架构与微服务架构基本相同，便不在此赘述。

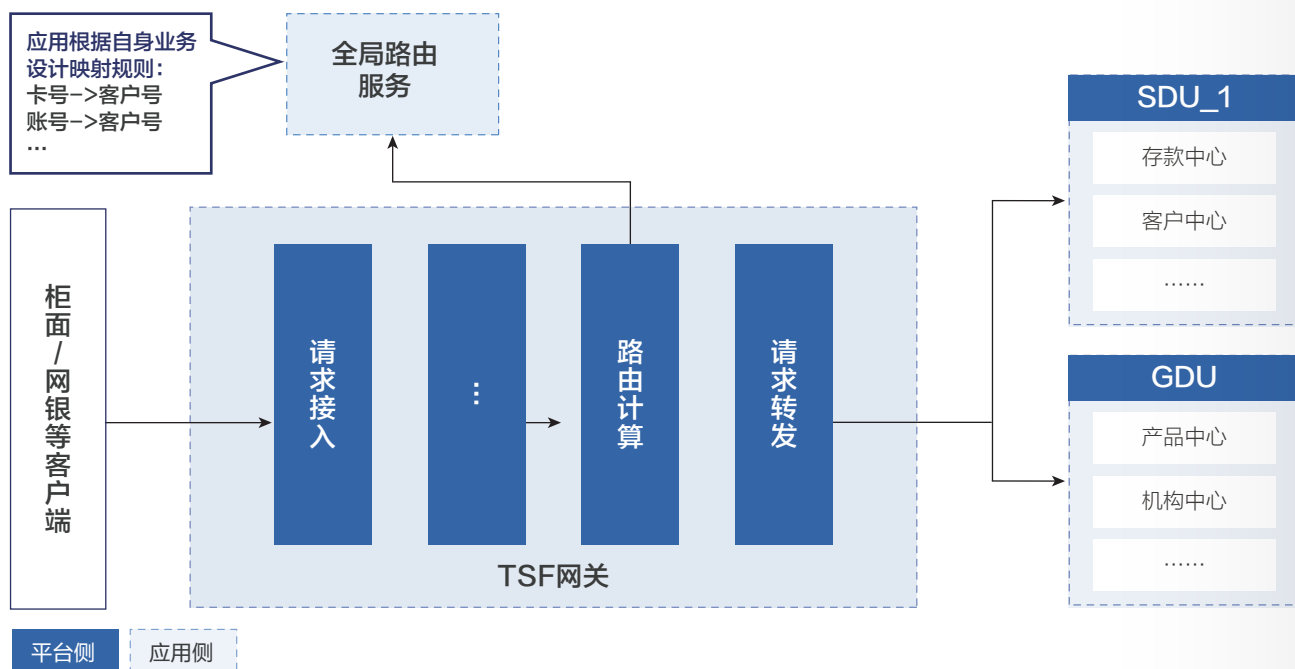
3.4.4.3 腾讯云单元化架构的落地实践

本节聚焦以下几个维度介绍腾讯在单元化架构领域的落地实践经验：1）应用系统集成，解析腾讯单元化架构与应用生态的深度适配方案；2）应用单元化改造，分析一个非单元化架构应用（比如标准的微服务架构应用）如何改造成一个单元化架构应用；3）数据库架构，提炼分布式数据库与单元化架构的融合范式；4）部署架构，归纳总结单元化架构的几种部署架构。通过体系化方法论输出，帮助金融机构更好的快速落地单元化架构。

应用系统集成

完整的企业级单元化架构是一项涵盖多维度能力的系统性工程，需融合 ISV 生态伙伴的应用架构设计、符合金融行业标准的容灾体系以及自主可控的技术中台能力。在单元化架构落地的过程中，需要结合实际情况进行改进、集成、甚至自研一部分组件来实现。其中腾讯侧结合自身产品优势，提供稳定的原子服务能力、由 ISV 伙伴集成到分布式核心技术平台中，该平台用于屏蔽底层分布式技术复杂度及隔离产品，封装成单一、易用的接口提供到上层应用。

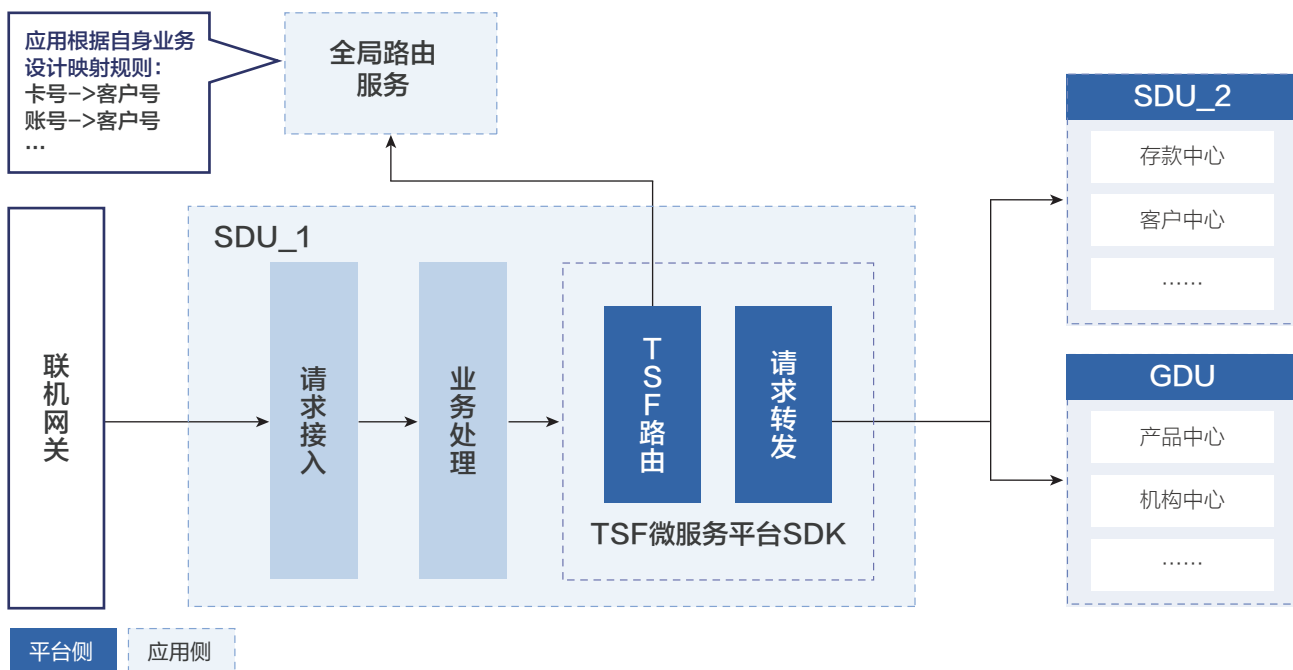
接入层集成及单元化路由



接入层主要用于将客户端访问流量根据全局统一的单元化路由规则分发到目标单元中，实现流量动态调拨。主要设计要点如下：

- 全局路由服务，即路由映射关系的维护系统，包括路由要素的新增、修改、查询、同步。因涉及客户数据信息且不同的应用映射规则不一致，全局路由服务由 ISV 厂商提供，按照 TSF 路由映射关系转换的接口标准提供查询服务。
- TSF 网关集成腾讯单元化路由 SDK，并且根据单元化管理平台 TCUA 下方的数据进行路由的计算。

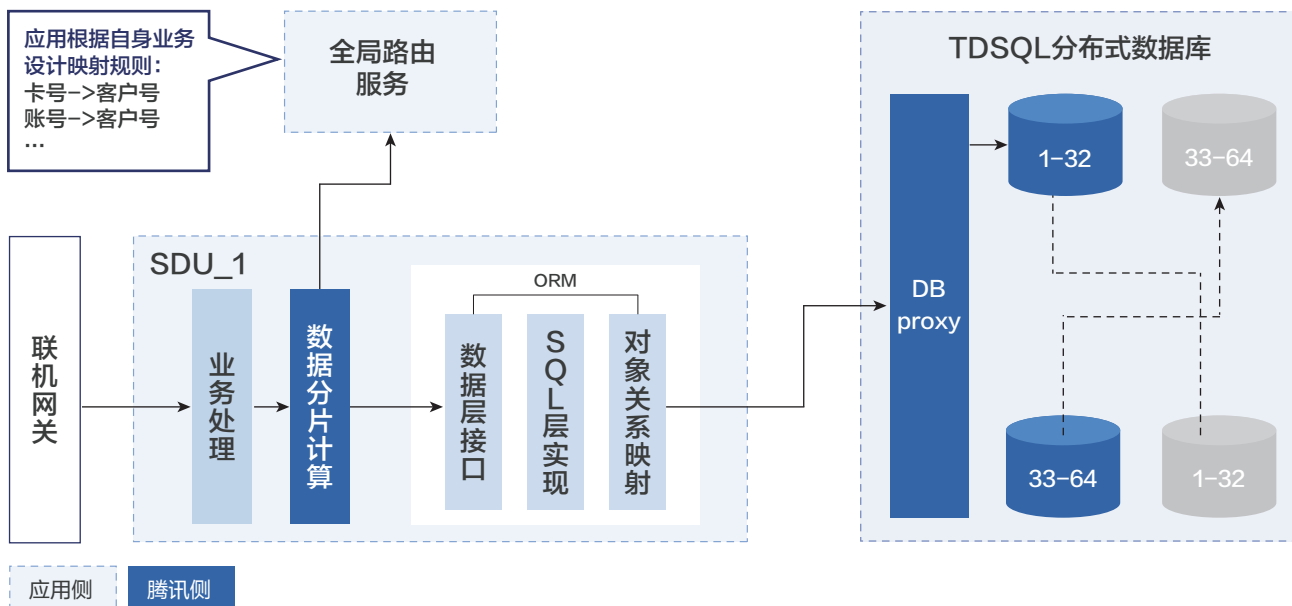
应用层集成



应用层集成包括微服务平台及消息队列集成，实现应用层的流量调拨。

- 应用需集成腾讯 TSF-SDK，服务的远程调用通过腾讯 TSF-SDK 进行，SDK 组件完成路由寻址，发起远程调用，应用不感知目标服务是否跨单元、跨 AZ、跨地域，由 SDK 组件自闭环。
- 当应用侧有异步通讯的需求，可集成腾讯侧的 TDMQ-rocketmq，实现消息的单元化路由功能，TDMQ-rocketmq 对腾讯单元化架构 TCUA 进行了适配，具备单元化路由能力。

数据层集成



单元化架构数据层集成，需注意应用层的路由策略和数据层的路由策略适配问题，具体策略，不同的数据库集成方式不一样，具体可查看下面数据架构维度章节内容。同时，还需考虑数据库表分片键问题。

- 针对应用系统设计时带有统一的数据分片键的数据库表，应用开发的 SQL 语句需带上分片键，此时分布式数据库可完成数据的路由。
- 针对应用系统设计时不带有统一的数据分片键的表，数据层的路由则需应用侧的 ORM 框架拦截并改写 SQL 语句，增加 hint 操作。

此外，在单元化落地过程中还应考虑灰度单元的实现，应用如果传递灰度标识，灰度名单配置等技术专项。

应用单元化改造

在落地单元化架构核心时，为了更好的发挥单元化架构的价值，通常会将一些非单元化架构的系统改造成单元化架构从而形成端到端的单元化架构，实现业务整个链条都闭环中一个单元内。本节分析探讨如何将一个非单元化的应用改造成单元化架构的应用。

应用单元化架构可行性评估

单元化架构虽然具备很多优点，但并不是所有应用系统都适合单元化架构，需要根据应用系统的特点进行评估，具体评估方法主要有以下几点：

- 业务系统分析：分析系统是否在面向客户业务场景的关键路径上，原则上只有处在面向客户交易路径上的关键应用系统才需要考虑单元化架构，因为通常面向客户类的应用系统对高可用和性能要求较高。管理分析类的系统对高可用和性能要求较低，无需考虑单元化改造。
- 数据架构分析：分析系统的数据架构，判断系统的数据库表是否可以按照按统一的维度进行切分数据。数据切分是单元化的基础，如果不能进行数据切分，则无法进行单元化改造。
- 交易接口分析：确定数据切分维度后，还需评估请求方能否改造以提供路由要素，并建立其到分片键的映射。否则，即使业务系统是关键路径且支持数据切分，也无法实现单元化改造，因为接入层无法据此路由交易到正确分片。

应用单元化改造方案

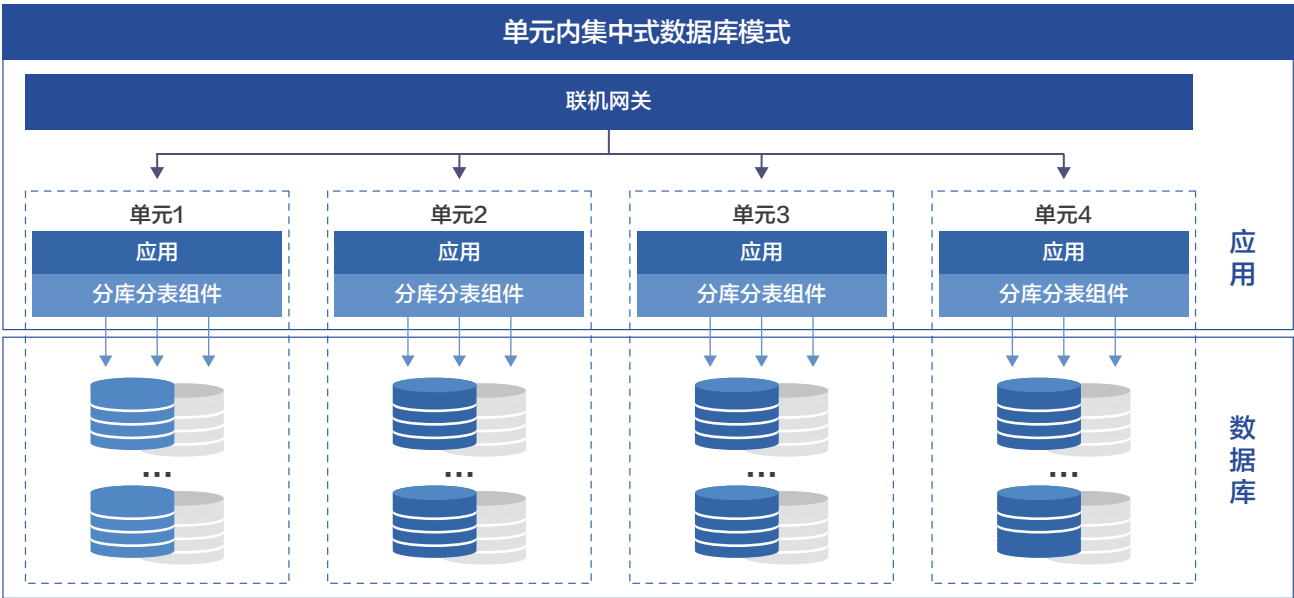
当应用系统确定采用单元化架构后，可参考如下步骤进行单元化的改造：

- 系统拆分：对应用系统进行业务模块拆分，将面向客户交易的模块规划到标准业务单元 SDU，参数类、后台管理类、非客户类业务模块规划到非标准业务单元中。
- 数据库表调整：按规划切分为全局应用库表和单元化应用库表，原则上需将单元化应用库表需加上分片键，且相关应用 SQL 语句也需加上分片键，确保分布式数据库按分片键路由数据。如无法添加，可参考前述单元化架构数据层集成方案。
- 路由要素设计：分析应用交易场景，识别可映射为核心客户号的要素，并在全局路由系统中建立映射关系。对可能产生新路由要素的交易（如开户），需在交易结束后将账户与客户号同步至全局路由系统。
- 业务功能调整：针对联机交易，将可能涉及到跨单元交易的服务进行拆分。以转账为例，当转入 / 转出客户分属不同单元时会出现跨单元的服务调用，因此需拆分为独立的转入、转出服务。针对批量及日终，单元化后批量交易及日终涉及按照单元进行数据拆分与合并，需要进行改造，并且要做好单元内批量执行的状态监控。

数据库架构选型

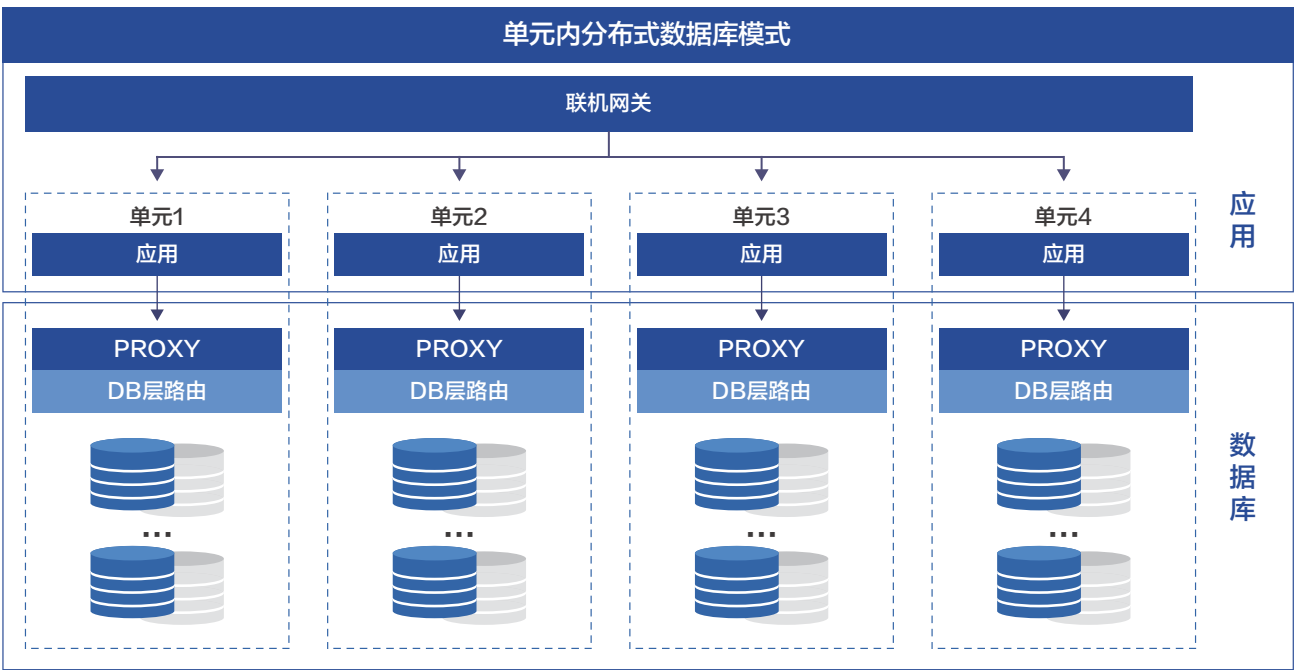
我们通过对多家银行核心单元化架构的落地总结，在单元化架构下分布式数据库集成模式有如下三种：

单元内集中式数据库



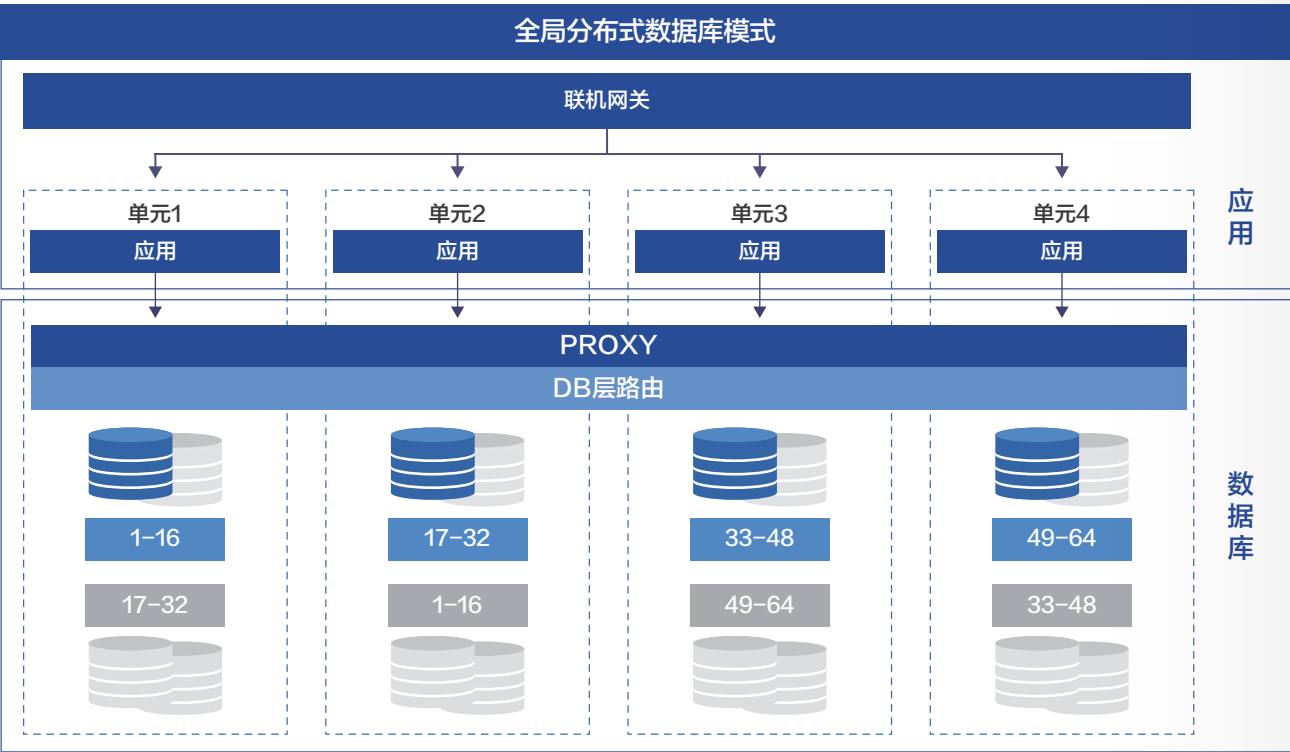
如图，每个单元内采用多个集中式数据库实例进行数据分片，每个单元通过分库分表组件访问多个数据分片，该方案的重点是分库分表组件的路由算法和接入层、应用层的路由算法保持一致，适合研发能力较强的客户。

单元内分布式数据库



如图，每个单元内部署一个分布式数据库分布式实例，该种架构下需注意应用层路由和数据层路由范围不一致，应用层的路由范围是全局的，交易在应用层可路由到所有单元，数据层路由范围是局部的，交易进入到某个单元化内，再由该单元内的分布式数据库实现数据的路由分发。

全局分布式数据库



如图，全局部署一个分布式集群，每个单元负责分布式数据库部分分片的业务处理，此架构下，应用层的路由策略和分布式数据库的路由策略需保持一致。

如下表是从架构的灵活性、国产化支持的粒度、应用开发设计的侵入性、非功能表现等多个维度进行对比和分析：

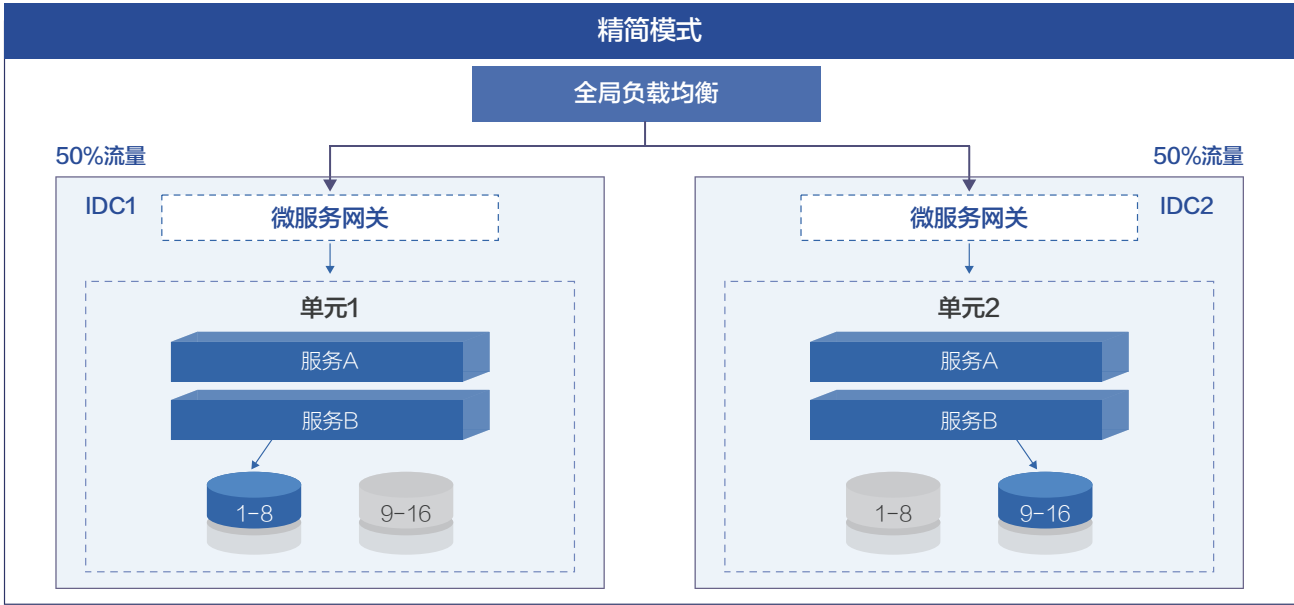
对比维度	应用分库分表模式	单元内数据库分布式模式	全局分布式数据库模式
架构灵活性	架构与路由灵活性高。可支持单元级灰度，包含数据和应用。	架构与单元间路由由灵活性较高，单元内分库路由由分布式数据库实现。可支持单元级灰度，包含数据和应用，但灰度颗粒度会比应用分库分表模式大。	路由策略需与分布式数据库路由策略一致。灰度策略主要是单元应用层灰度+数据打标。
国产化支持	可满足更灵活的国产化要求，支持单元级国产化部署。	可满足更灵活的国产化要求，支持单元级国产化。	部署在同一个分布式集群的单元可整体国产化，或灾备国产化。
汇聚支持	需应用汇聚或引入汇聚库集群	需应用汇聚或引入汇聚库集群	对于简单汇聚场景可直接基于分布式数据库实现
应用设计入侵	1.应用需考虑交易是否存在跨单元场景，指导应用服务拆分，批量交易需先按照单元拆分。 2.需要在应用层引入分库分表组件，对应用有侵入。需应用实现单元内分库路由。	1.应用需考虑交易是否存在跨单元场景，指导应用服务拆分，批量交易需先按照单元拆分。 2.单元内应用适配需考虑分布式数据库最佳实践。	1.应用需考虑交易是否存在跨单元场景，指导应用服务拆分，批量交易需先按照单元拆分。 2.应用适配需考虑分布式数据库最佳实践。
多活支持	支持同城多活，未来更利于扩展为多地多活。	支持同城多活，未来更利于扩展为多地多活。	支持同城多活
可扩展性	扩容实现难度大，包括新增单元和单元内扩容。扩展时需要应用考虑数据迁移，对客户的服务影响时间更长。	单元内扩容可依托分布式数据库实现，单元间扩容复杂度同应用分库分表模式	单元内和单元间扩容均可依托分布式数据库实现。扩展时最方便，由数据库完成扩容和数据重分布。
可维护性	运维复杂度非常高，涉及的数据库实例多，日常运维工作较复杂，每个单元需要配置不同的数据库链接。	运维复杂度较高，每个单元需要配置不同的数据库链接，多单元运维操作复杂度高。	运维复杂度较低，只需要运维一个分布式数据库实例，应用的连接地址也统一。
高可用表现	数据库故障仅影响一个单元或一个单元的一个分库。	数据库故障影响一个单元或一个单元的所有分库。	数据库故障影响所有单元。

从实践来看，业内上述 3 类单元化架构下分布式数据库集成模式均有较多的实践应用，银行可基于自身核心关注的维度进行评估选择。例如如果新核心主要关注灾难影响半径、多地多活、或单元路由和灰度实现灵活性，可以更多考虑“应用分库分表模式”或“单元内数据库分布式模式”；如果新核心主要关注应用的侵入以及日程运维复杂度，可以更多考虑“全局分布式数据库模式”。

部署架构介绍

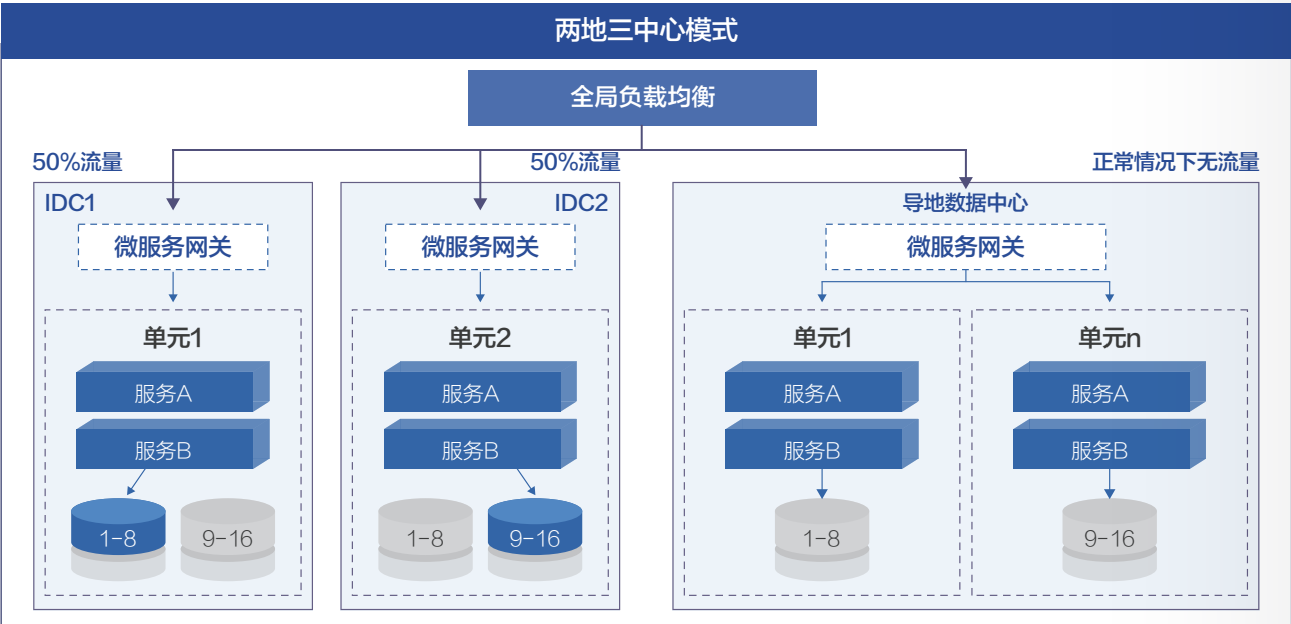
单元化架构的部署模式，我们通过调研和腾讯自身帮助客户落地的案例分析总结，有如下三种方式：

同城双活模式



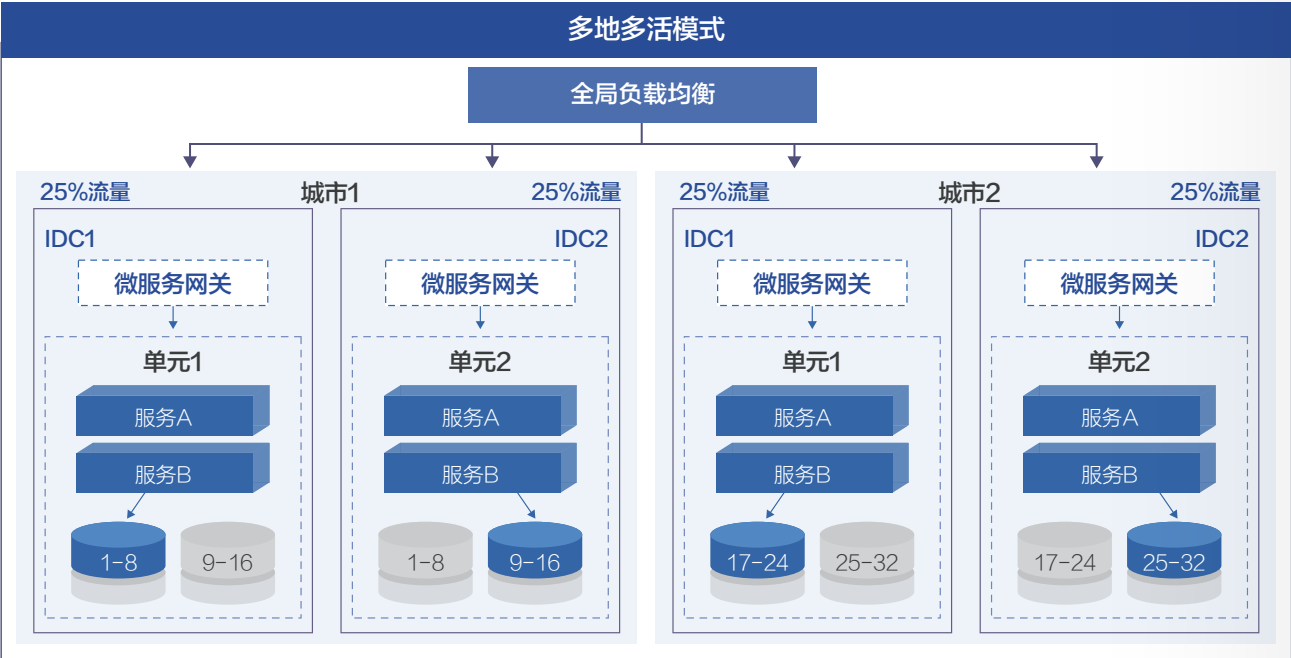
同城双活模式，业务部署在同城两个数据中心，最少可以只有 2 个单元，一个数据中心一个单元，彼此互备。该模式下，成本最低：仅需 2 个同城数据中心，硬件投入小，架构简单：无异地组件，流量均分策略易实现，缺点是容灾能力弱：无法应对城市级灾难（如地震、大规模断电）。适合小型金融机构，或者当前只有同城两个数据中心的客户。

两地三中心经典模式



两地三中心模式，在监管的要求下，当前大部分金融客户都有规划两地三中心的架构，也是单元化架构的经典模式，其原理是，主城两个数据中心对等部署若干个业务单元，并且相互备份，同时每个单元在异地也都部署一份，作为灾备单元，这样形成同城双活，异地灾备的架构，符合监管要求。此外该种架构还可以继续延伸，可以将前面章节提到的灰度单元部署在异地，这样异地机房平时也有流量进入，将异地的资源利用起来。该架构是大多数金融机构的平衡选择，满足监管容灾要求且成本合理。

多地多活模式



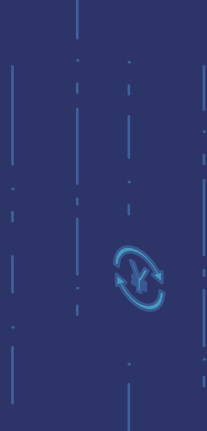
多地多活模式下，业务单元被分部在不同城市的不同数据中心，且每个地域都有正常的流量进入，该模式下拥有极致的高可用，任何单点/城市故障自动切换，容灾全覆盖，同时防数据中心级、城市级灾难，但是模式建设成本高，需建设多地数据中心及专线网络，技术复杂度陡增：需全局负载均衡、全局注册中心，运维成本也高。

如下表是几个关键维度的对比：

对比维度	同城双活模式	两地三中心模式	多地多活模式
容灾能力	低，仅限城市内，无异地容灾。	高，有异地宅备，但是多数情况下处于冷备状态。	高，可做到城市级容灾。
建设成本	低，大部分机构都有同城双机房。	中，两地三中心是监管部门的要求，大部分机构都有建设。	高，数据中心的建设成本高。
资源利用率	高，资源都处于使用中。	低，异地灾备机房通常处于冷备状态，即使灰度单元部署到异地，流量也较少。	高，资源都处于使用中。
技术门槛	相对较低，单元化技术架构经过多年的发展及大量客户的使用已经很成熟。	相对较低，单元化技术架构经过多年的发展及大量客户的使用已经很成熟。	相对较高，需考虑全局注册中心，全局负载均衡，全局路由需异地多活，同时需考虑异地数据一致性方案。
运维复杂度	低，只需运维一个城市的数据中心。	中，需多维护一个异地灾备机房。	高，全建设跨城市的运维中心。

综述，对于大部分银行机构采用两地三中心的架构即可，该架构是大多数银行机构的平衡选择，满足监管容灾要求且成本合理、建设难度适中。对于有超大规模集群或追求极致高可用架构的银行机构可以选择多地多活的架构。





PART

4

腾讯云分布式核心系统技术底座

在介绍完分布式核心的两种技术架构路线后，本章聚焦分布式核心系统的技术底座，阐述分布式核心系统的技术底座能力要求。





如 2.3 章节所述,腾讯云新一代分布式银行核心系统基于云计算体系构建,并整合了行业领先的专业核心应用厂商提供的分布式技术组件,形成了分层解耦的架构。整体以 IaaS 基础设施层和 PaaS 平台层作为坚实的技术底座,支撑核心应用的研发与稳定运行。其中, PaaS 层按能力范围进一步划分为 APaaS、IPaaS 和 GPaaS。

4.1

APaaS分布式技术组件

APaaS 聚焦通用分布式技术封装，为 SaaS 应用提供统一的 API 服务，因与应用耦合的比较紧密，因此大部分核心能力组件（如交易服务框架、批量服务框架、参数管理、分布式锁、数据访问组件、全局流水号等）通常由核心应用厂商提供。其设计要求如下：

开发框架

屏蔽底层分布式技术细节，目标通过封装一系列的技术组件，让应用研发团队只需要关注银行核心系统本身的业务而无需关注分布式技术细节，比如分布式事务、分布式锁、全局流水号、服务编排等。

兼容解耦

需要具备高度的兼容性和灵活性，能与下层的分布式中间件及微服务套件，数据库产品、云平台解耦，底层产品的切换应对上层应用无感。

双架构支持

需适配标准的微服务架构和单元化架构两种技术路线，具备高可用、高性能、高可靠性，适合大型分布式应用，足以支撑高并发与海量数据业务。

高效开发

提供脚手架工程（代码生成、配置向导等），内置一些金融场景的典型案例，降低开发者的学习成本，使应用开发者可快速上手银行核心应用开发。

APaaS 通过技术抽象与标准化能力，实现分布式复杂度下沉，支撑银行核心系统高效、灵活演进。





4.2 IPaaS系统集成服务平台

银行核心系统作为数字化转型的关键枢纽，新一代分布式核心系统的建设将深度重构全行 IT 架构生态。由于核心系统与支付清算、信贷管理、渠道服务等上百个周边系统存在高频交互，新核心的建设必然触发关联系统的协议适配与接口重构。在此背景下，IPaaS 平台的建设具有战略必要性：其通过构建统一的集成治理平台，有效应对云环境与传统架构并存的混合技术栈难题。在这一领域通常有两个重要的平台，系统服务集成平台与文件传输平台，系统服务集成平台主要完成周边系统与银行核心系统联机交易部分的集成，文件传输平台主要完成各个系统间的文件交互，这两个平台通常也由该领域的专业供应商提供。

系统服务集成平台（新一代ESB），核心理念是采用边车思想，去中心化的架构，实现核心系统与周边系统集成。需具备如下能力：

- 需适配云厂商的 GPaaS 服务，形成与银行核心系统统一的技术架构。
- 需具备报文格式转换能力，包括定长、变长、8583 报文等，支持不同格式报文之间转换的能力。
- 需支持多种通讯协议，包括 socket，http、grpc、rpc 等协议。
- 需具备简单的微服务接口编排功能，通过该能力来屏蔽新核心服务接口变化带来的差异，目标是周边系统无须改动即可与新核心系统集成，降低周边系统的改造成本。

系统集成服务平台通过标准化集成能力，降低周边系统对接成本，提升核心与外围系统协同效率。

文件传输平台，定位于银行各个系统之间的文件交换。其核心能力要求如下：

- 构建统一的文件传输平台，实现银行各系统之间高效的文件传输需求；
- 实现高可靠可横向扩展的全行节点管理；
- 支持文件传输加密、文件完整性校验、文件分片传输、断点续传和数据压缩处理；
- 支持各类文件格式的传输和文件处理能力（格式转换、合并、拆分、转码等）；
- 支持文件传输过程的实时监控、文件审计；
- 实现与云厂商的对象存储集成，支持文件流传输。

文件传输平台通过标准化、高安全的文件传输能力，实现全行系统间文件高效协同与全链路可控。





4.3 GPaaS通用技术组件服务

GPaaS 作为通用技术服务，专注稳定可靠的中间件供给，为分布式核心系统提供数据库、微服务套件、消息队列、缓存、容器平台等基础支撑服务。腾讯云在这一领域拥有成熟稳定的产品，包括分布式数据库 TDSQL、微服务平台 TSF、分布式消息队列 TDMQ、分布式缓存 CRedis、容器服务平台 TKE 等支撑新一代分布式核心系统的稳定运行。

分布式数据库TDSQL

TDSQL 分布式数据库需具备良好的跨机房高可用容灾能力，是构建银行核心的金融级数据库底座，TDSQL 分布式数据库为众多金融机构提供了稳定、可靠、安全的数据服务，助力其关键核心业务系统的国产数据库替换，其核心优势及能力主要体现在：

- 数据库集群支持同城双活高可用架构，通过多副本实现跨机房数据强同步，机房级故障 RPO=0，支撑上层银行核心业务极致的可用性。
- 数据库集群可按需基于数据备副本构建只读平面，支撑应用双活数据读写分离架构的落地。
- 数据库集群采用多租户架构，可以按需为银行新核心微服务提供数据库租户实例。不同的实例主库可以按需放置到指定数据中心，增强业务跨中心双活架构。
- 提供跨机房灾备部署架构，支撑业务同城灾备和异地灾备的高可用部署需求。数据库集群的灾备同步采用异步复制，通过管控平台提供数据库集群的灾备演练和灾备故障切换支持。

在金融领域，腾讯云数据库 TDSQL 稳居中国市场份额第一，已覆盖超过 1000 家金融机构。其中，国内前 10 大银行中就有 7 家用腾讯云 TDSQL，并助力 40 多家金融机构的核心系统实现自主创新升级。

微服务平台与通用分布式中间件

腾讯提供 TSF 微服务平台、TDMQ 分布式消息队列及 CRedis 分布式缓存等微服务套件与分布式中间件。其中 TSF 全面兼容 Spring Cloud、Ggpc、Dubbo、service mesh 开源生态，是腾讯云分布式核心系统交易处理的核心框架；TDMQ（含 rocketmq、rabbitmq、Kafka、pulsar 等产品）为核心系统提供异步通信消息服务；CRedis 则提供分布式缓存能力，三者均支持腾讯云分布式核心系统的两种架构路线。

腾讯云微服务平台与通用分布式中间件源于公有云，完全兼容开源生态，将公有云的成熟产品能力复刻至私有化部署环境，服务过多家金融机构的核心系统建设，具备安全、稳定、高可用、扩展性强、开放兼容等特点。

云原生容器平台

银行核心系统通过分布式架构转型已初步解决了传统集中式架构的性能瓶颈与单点风险问题，实现了核心系统从稳态到敏稳双态的跨越式发展。然而，随着数字金融场景的爆发式增长，传统分布式架构在业务敏捷性、资源效能等方面逐渐显现出新的挑战，云原生技术基于 Kubernetes 的声明式 API 与 GitOps 工具链可以提升核心系统的敏捷性，此外，Kubernetes 弹性伸缩能力，可有效提升资源效能。

腾讯云容器服务平台 TKE 基于原生 Kubernetes，提供高性能、可扩展的容器管理服务，技术栈与腾讯自研业务同源，支持千万核级集群，助力银行核心系统云原生演进。其核心能力如下：

- 异构融合与国产化适配，支持 "利旧 + 纳新" 的混合资源管理能力，支持 X86/ARM 架构混合部署，提供跨芯片平台的统一资源调度；通过全栈国产化技术适配，实现从芯片、操作系统到中间件的全链路国产化替代，保障系统平滑迁移。
- 安全合规，符合金融监管政策，提供容器网络安全，镜像安全，容器运行时安全等全方位云原生安全服务。
- 高性能，银行核心系统对性能要求极高，平台具备良好的性能以满足金融业务的需要。
- 高可用，银行业务对业务连续性有较高要求，要求平台支持多活部署，支持自动化容灾。

- 智能运维，完整的运维能力，从资源管理到故障现、监报告警、日志收集与检索、全链路追踪的全流程能力，并支持自动化运维。
- FinOps 支持，FinOps 通过成本洞察、优化、运营联动，构建 "可观测 - 可执行 - 可量化" 云成本闭环，平衡性能与财务责任，推动跨部门协同，建立成本可控、效率可溯的云治理体系。

TKE通过金融级云原生能力，为银行核心系统提供安全可靠、高效敏捷的容器化支撑，加速分布式架构向云原生进化。

4.4 IaaS基础设施服务

IaaS 基础设施服务为分布式核心系统提供计算、存储、网络、安全等基础设施服务，是分布式核心安全、稳定运行的基石。

腾讯云专有云企业版 TCE（Tencent Cloud Enterprise）是腾讯面向企业数字化转型需求推出的专有云平台，通过“同源同构”设计，将公有云的成熟产品能力复刻至私有化部署环境，深度融合了云计算、分布式架构与金融级安全能力，为金融机构提供从安全、稳定、可靠的基础设施服务。该平台基于腾讯服务海量互联网业务的技术积淀，采用自主研发的云原生技术栈，支持 x86、ARM 多芯异构架构，兼容国产化软硬件（如自研芯片、操作系统），实现从底层芯片到上层应用的全链路自主可控，具备金融监管合规、业务高可用、资源弹性扩展等核心特性，已通过数十项行业权威认证，成为银行新一代分布式核心建设的首选技术基座。

在技术架构层面，TCE 构建了 "双引擎驱动" 的混合云管理体系，既支持私有化独立部署保障数据主权，又能无缝对接腾讯公有云形成混合云架构，实现跨云资源统一调度。平台通过智能虚拟化技术实现计算、存储、网络资源的全栈池化，单集群可扩展至数万节点规模，支持百万级容器并发调度，结合分布式服务治理框架，使得核心交易系统响应延迟降低至毫秒级，系统可靠性达到金融级标准。特别在容灾能力方面，TCE 支持 "同城多活" 及 "多地多活" 的容灾机制，支持分钟级故障切换与数据零丢失，可有效支撑核心系统的平稳运行。

针对金融行业特殊需求，TCE 打造了行业领先的技术中台能力矩阵。通过整合包括分布式数据库 TDSQL、微服务平台 TSF、分布式消息队列 TDMQ、分布式缓存 CRedis、容器服务平台 TKE，具备全栈分布式与云原生的国产化技术产品体系。平台还内置智能运维大脑，配合可视化运维大屏，可系统故障恢复时间从小时级缩短至分钟级。TCE 支撑多家银行、保险、证券等金融机构核心系统的稳定运行。

目前 TCE 平台已形成覆盖开发规范、部署模板、运维 SOP 的完整交付体系，为金融客户提供全生命周期的云服务保障，持续推动金融行业数字化基础设施升级。

PART



腾讯云分布式核心与伙伴联合方案

腾讯云从成立伊始就保持着高度开放的生态策略，我们始终坚持技术架构的开放性，在金融行业国产化的攻坚战中，我们与神州信息、长亮科技、中电金信等头部核心应用厂商形成紧密的合作阵型，一同为银行客户提供灵活多元、体系先进、深度融合、落地经验丰富的高质量银行核心系统分布式转型方案。

5.1

腾讯云与神州信息的深化合作

腾讯云与神州信息在2021年6月共同推出了“金融分布式核心”联合解决方案。联合方案由腾讯云金融级专有云平台TCE作为技术底座，数据则由腾讯云分布式数据库TDSQL承载；在此基础上，神州信息Sm@rtEnsemble分布式微服务核心业务系统充分与云平台和分布式数据库深度融合，在保证核心系统稳定、安全、高效、易扩展等优势外，满足动态扩容、分布式多活、多中心容灾、海量数据并发处理等业务诉求。

双方的联合方案通过“微服务 + 分布式 + 云”模式构建可弹性伸缩的核心业务系统，可以为客户提供：

- 核心业务系统微服务化设计支持各业务模块的独立设计与敏捷交付，按需配置资源，实现系统的按需灵活部署
- 通过核心平台分布式组件实现业务的分布式处理，使系统具备海量数据的处理能力
- 具备应用级分布式事务保障方案，确保分布式环境下的数据一致性
- TDSQL实现数据库层的跨中心分布式处理，提供高效、高可靠的数据处理及存储服务，支持数据分片、单元化等多种部署模式
- PaaS层组件Redis、Kafka提供了分布式环境下的查询加速与高效协同能力
- 腾讯专有云提供弹性计算服务，支持按需扩容，做到核心业务系统的敏捷部署与动态伸缩。

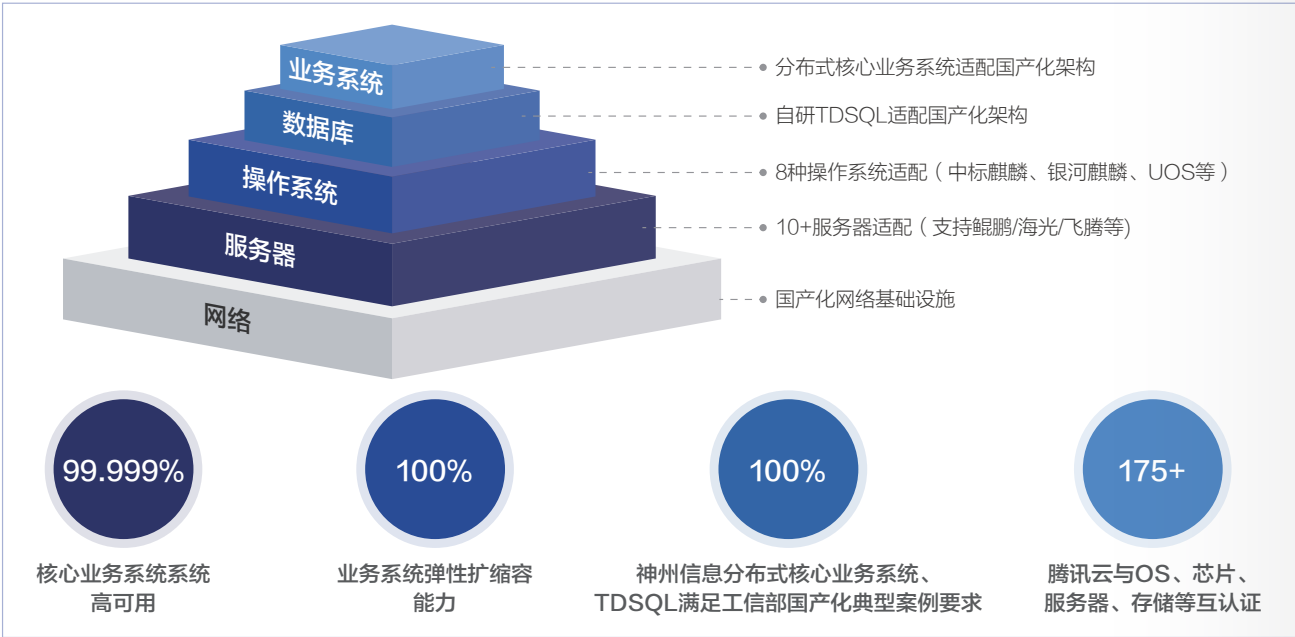


在联合方案基础上，针对双方持续演进的核心产品最新版本，神州信息与腾讯云也在持续地进行深度的适合作。如下是神州信息 Sm@rtEnsemble16 的应用架构。

业务服务	客户服务		存款服务			贷款服务		支付服务		参数平台
	客户识别	客户信息管理	活期业务	定期业务	保证金	个人贷款	网上贷款	参数管理	对账差错	参数定义
	客户账户管理	客户签约管理	借记卡	现金池	大额存款	公司贷款	融资贷款	交易管理	清算管理	界面定义
	客户关系管理	客户风险管理	结构性存款	账户限制	司法查控	票据贴现	联合贷款	报文管理	头寸管控	变更管理
基础能力	运营服务		定价中心		产品中心		核算中心		流程管理	
	机构管理	现金管理	利率计算	费率计算	产品定义	产品管理	会计引擎	业务总账	参数发布	
	柜员管理	辅助交易	汇率计算	税率计算	基础产品	可售产品	核算处理	损益结转	参数对比	
			差异化价格		组合产品		价税分离	科目余额		
技术平台	凭证管理		个性化价格		产品引擎		清算处理	外币重估		
							对账处理	月结年结		
技术平台	注册中心		服务管控		服务调用		批处理调度		配置管理	
			分布式缓存		多数据源访问		平台监控		平台运维	

神州信息核心业务系统采用面向服务的 SOA 架构体系，通过分层实现数据、产品、交易和展示的有效分离，实现业务数据和业务逻辑的分离，系统模块间松耦合，功能分布合理；对外提供标准化的服务接口，服务接口由各个服务组件装配而成，最大化的实现了服务组件的组装与重用，可对接前置、服务总线或者与外围系统直连，以适应 IT 系统、服务、产品、流程的变化。

针对双方产品最新版本，腾讯云与神州信息联合开展了基于云原生、微服务架构、分布式数据库 TDSQL 的全栈国产化新一代核心系统的投产级测试，对核心业务系统的各项功能和技术性能进行了深度验证，经过适配和调优，双方产品可靠性、性能指标满足金融机构自主可控要求。



目前双方联合解决方案已在包括秦皇岛银行、浦发硅谷银行在内的多个商业银行落地。未来，腾讯云与神州信息将围绕产品、生态及服务等多维度持续深化合作，为银行业提供更高品质的产品和服务，为技术创新和行业发展持续贡献力量。

5.2

腾讯云与长亮科技深化合作

腾讯云与长亮科技在银行传统核心领域的合作始于 2018 年的张家港农商行核心系统，这是首个传统银行核心业务系统全部采用国产数据库的标杆项目。在此基础上，双方于 2019 年腾讯全球数字生态大会智慧金融专场上联合发布了新一代分布式金融业务服务框架 TDBF。框架以腾讯云的云化技术支撑平台为底座，结合长亮科技预置业务能力，在满足金融系统全面自主可控的基础上，集中解决金融行业转型的“慢、难、窄”问题，进一步革新系统建设模式，助力金融机构快速实现数字化转型。

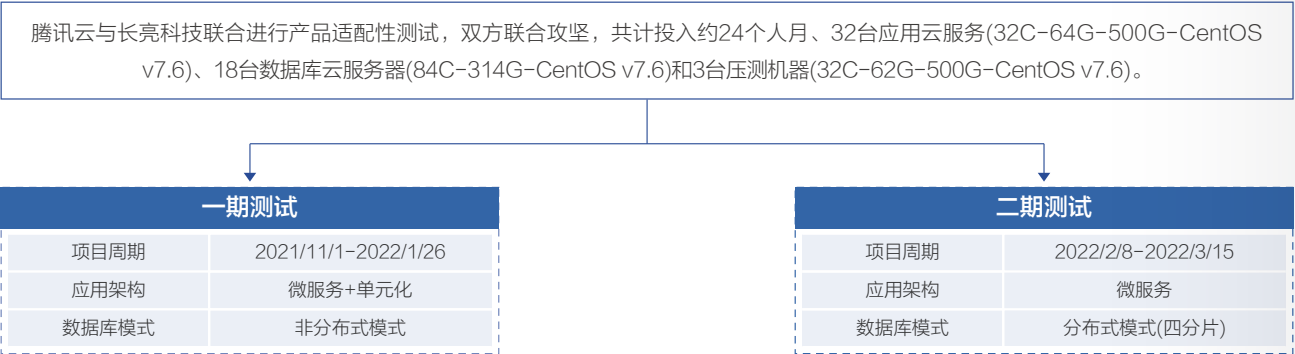


近年来，腾讯云与长亮科技针对双方产品最新版本持续推进高质量的深度适配与协同。如下是长亮科技分布式核心最新的 V8 版本应用架构图。

领域级 业务应用层	负债中心		资产中心		借记卡中心		信用卡中心	
	负债业务	组合产品	贷款业务	抵债资产	借记卡业务	借记卡前置	进件审批	发卡用卡
	中间业务	综合账户	抵质押物	银承汇票	借记卡发卡	交易对账	交易反欺诈	授权处理
	外汇买卖	同业存放	票据贴现	表外业务	电子现金	卡组清算	贷后管理	账务处理
复用 能力中心层	用户中心		产品中心		权益中心			
	客户信息		产品目录		红包			
	客户标签		产品模型		卡券			
	名单管理		产品发布		积分			
	计价中心		核算中心		参数中心			
	利率		科目管理		参数权限			
	费率		核算引擎		参数流程			
	汇率		清算引擎		公共参数			
	限额中心		合约中心		运营中心			
	账户限额		产品合约		机构柜员			
	产品限额		定价合约		现金			
	II III 类户限额		服务合约		凭证			

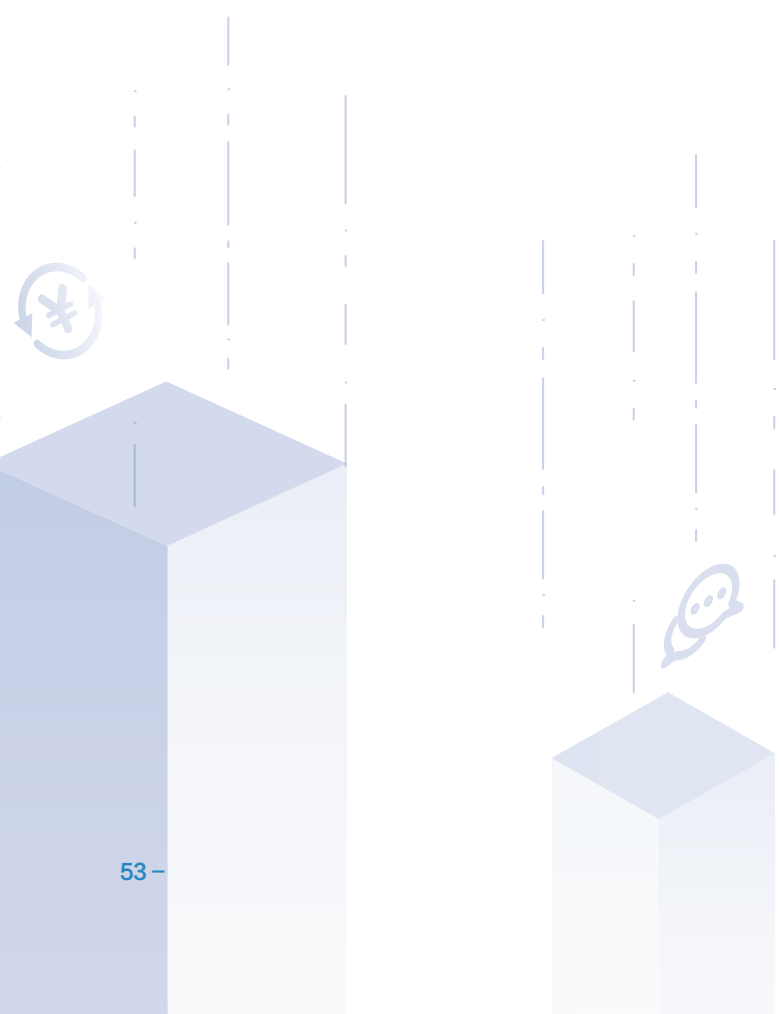
长亮分布式核心业务系统是一套基于分布式技术平台的以客户为中心，以产品为业务框架，以账户为业务实体，以合约为业务纽带，以事件为业务驱动，并拥有灵活多变的账户体系与计价体系的、完全自主可控且功能全面、稳定高效、高可用、高扩展性的核心业务系统。

针对双方产品最新版本，2021 年腾讯云与长亮科技完成了长亮”新一代 V8.0 微服务 + 单元化分布式核心系统”与腾讯 TDSQL MySQL 版产品投产级适配性测试。在金融测试场景中，验证负债、资产、运营、银行卡、核算 5 大类业务，涉及 1000+ 个交易及批量业务，达到了良好的处理能力和效率以及 100% 的正确性。并且通过底层服务器和数据库的可扩展，能够获得良好处理性能值，以及线性增长能力，为全面保障银行核心业务系统数据安全提供有力的支撑。



腾讯云与长亮科技双方持续打磨联合方案，加速赋能银行数字化转型，合作范围现已囊括了包括 TDSQL、TCE、TDMQ、TSF、Coding、TCS 在内的腾讯云全体系的 PaaS 产品。同时联合方案也相继在平安银行、福建海峡银行、昆山农商行、云南红塔银行为代表的多个银行核心系统相继成功投产。

未来随着双方合作的深入，将市场带来更高效、更安全的完整生态链的分布式核心系统解决方案，为金融客户深入开展数字化转型持续注入强劲动力。



5.3

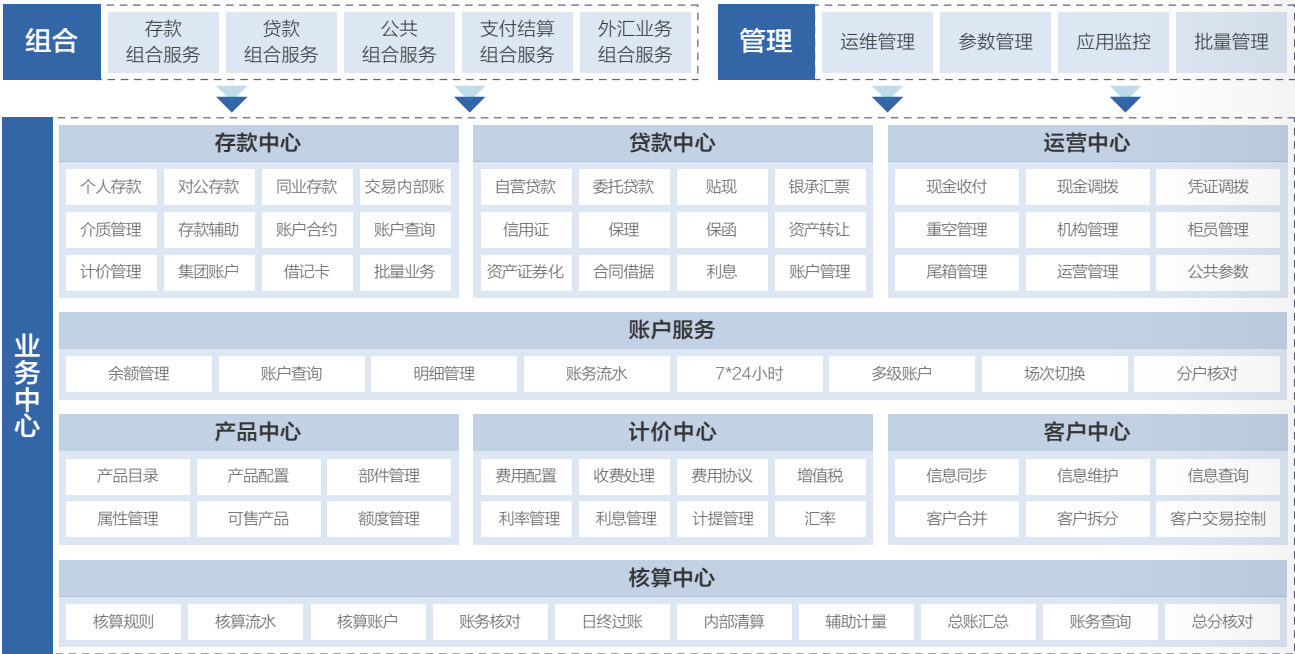
腾讯云与中电金信深化合作

腾讯云与中电金信于 2022 年展开全面合作，双方基于 eCAS 分布式核心系统 + 源启 + 腾讯金融云产品打造联合解决方案。联合方案支持云原生和单元化，面向银行类金融机构实现全栈国产化落地，具备高性能、高可用、高安全和可动态弹性扩展能力，为金融用户提供关键的核心业务处理能力。



联合方案中腾讯云和中电源启底座进行了深度适配，并在此基础上通过协同，能够为金融机构提供具有多样性的、适合实际技术需要的金融数字化基础设施的组合方案，为金融机构的核心应用提供全面支撑。腾讯云主要聚焦在计算服务层和支撑组件层，提供 TDSQL 数据库、TCS 云原生平台、TSF 微服务框架、TDMQ 等众多 PaaS 组件以及 TCE 专有云平台。中电金信源启金融级数字底座主要包含基础运行支撑平台、数字构建平台及数据资产平台，是基于全栈技术的具有通用性、完备性、广泛适配性的开放式分布式技术平台。

联合方案基于的中电金信核心系统版本为 eCas4.0。如下是中电金信 eCas4.0 的系统应用架构。



中电金信分布式核心系统遵循了“以客户为中心，以产品为主线，以组织和定价管理为支撑，以财务核算体系为基础，以数据资产价值为驱动”的设计理念；采用了“应用分层、业务分域”的方法来构建一体化、产品化、流程化、规范化的核心应用架构；在工艺方面采用了在企业架构指导下，以业务建模为引导、以一体化工具链为支撑的端到端工程方法，实现了由“业务决定架构，由业务能力决定微服务设置”的业务技术一体化的架构风格，确保金融机构能力要求向系统设计的无缝衔接、传导落地，从而支持银行用户企业级的敏捷。

腾讯云与中电金信联合方案具有如下价值：

业务理念方面

- 以客户为中心的一体化账户体系，从客户体验出发进行产品创新，支持全面营销，具备产品及客户服务流程的标准化与差异化支撑等能力，提供超个性化服务；
- 提供业务合规性和数据、业务等资源的智能配置，优化业务流程，降低运营成本，提升运营能力，实现降本增效；
- 采用中台化架构，沉淀稳定的核心业务支撑能力，助力金融机构实现业务敏捷。

技术架构方面

- 以云原生为根本属性，采用分布式架构、容器技术、微服务等技术，具备单元化方案，实现系统弹性伸缩，具备高扩展性和负载均衡，实现技术敏捷；
- 深度适配 TDSQL 分布式数据库，提供金融数据强一致保障与高性能吞吐支撑。
- 基于源启金融级数字底座，结合腾讯云一站式、智能化、高可用、多样化兼容及便捷扩展性能力，深度的国产化适配，满足高效能、高可靠、高一致性要求，为银行系统带来平滑上云的一体化整体解决方案。

敏捷交付方面

- 提供开放的 API，运用 DevOps 的思想，提供端到端的一体化工具链，实现低代码的开发，能够快速开发和交付，实现运营敏捷。



腾讯云与中电金信核心下移联合解决方案已完成投产级适配，并正在多个银行和省级农信进行推广落地。

未来腾讯云和中电金信在银行核心领域展开更深入的合作，通过双方产品层面充分的技术融合，共同打造更加灵活、高效、安全的全栈国产化银行核心系统解决方案，共同推动银行核心领域的数字化转型与业务价值实现。



PART

6

落地实践案例

本篇章聚焦分布式核心转型的最佳实践，甄选具有行业里程碑意义的标杆案例进行剖析，从技术架构路线上看，涵盖单元化架构、标准微服务架构两种技术架构路线的案例，此外，从核心上云角度看，选取了核心云原生的案例，为不同发展阶段的银行提供可落地的核心分布式转型路线参考。



6.1

某国有大行核心系统单元化案例

银行的核心业务系统主要包括借记卡，信用卡，客户信息管理系统，存款等主要业务系统。国有大型商业银行由于超大的客户量级和超高的稳定性要求，一直是选择主机来支撑其核心系统运行。由于客户量级的不断增长和业务创新带来的技术变革要求，国有大行的新的技术路线选择了基于分布式的单元化架构来做整体系统架构重构。

2022年某国有大行开启了分批次的核心系统下移业务投产工作。基于腾讯云多款PaaS产品，腾讯云与客户一起联手打造了具备组件化、平台化、服务化特性的分布式技术中台等一系列新技术平台。此次核心系统改造采用腾讯云企业级分布式数据库 TDSQL 承载新核心业务和交易数据，实现关键数据库领域的自主可控；采用腾讯云企业级缓存产品 CRedis 作为高速缓存系统，CRedis 具有弹性扩展、超高性能等优势；在高可靠消息系统上，采用腾讯云分布式消息中间件 TDMQ 作为支持，支持系统解耦、异步通信、削峰填谷的应用场景。在架构设计上，基于微服务设计理念，依托腾讯云微服务平台 TSF，实现了多地多中心的业务高可用和多活的架构，且分布式核心支持业务应用根据流量负载情况及时便捷地进行单元化流量切换。

整体设计思路

该行在单元化架构设计考虑上也遵从了单元化的六大设计要点。单元架构下的服务调用与负载，遵循以单元维度和就近调用的设计原则。业务应用拆分为微服务，基于腾讯云 TSF 实现微服务的治理，服务间的事务通过引入分布式事务组件来实现。业务连续性的设计上充分利用了业务连续性的全面设计原则来保证业务的整体服务质量。为了提高研发效能，该行充分利用主流的开发平台，并且在组织架构上做了适配和调整，确保研发的高效性。分布式运维体系的整体引入确保整个分布式系统运行过程中故障的快速定位、快速隔离，实现了对业务连续性的优先保障。

基于单元化设计架构，客户结合业务路由策略和数据分片策略，通过单元的水平扩容，实现业务吞吐量的线性增长。在同城业务单元模块切换时，可以实现计划内一键切换和计划外自动切换。该行客户信息系统客户规模达到数亿级别，使用了超过 1400 个数据库实例。在业务投产之前，客户进行了严格的业务仿真验证。其中在高可用专题下，客户进行了模拟单 AZ 不同层级的支撑组件出现问题、同城数据中心级别的网络故障、异地数据中心容灾等多种极限场景的高可用测试。测试结果显示整个腾讯云 PaaS 平台完全满足客户的非功能设计要求，可实现跨多数据中心业务多活。在新架构下，客户的业务应用可在多地同步发布，真正地实现了去中心化的全网分布式部署，从而改变了传统部署模式下客户灾备中心资源不能充分利用、业务模块紧耦合等一系列问题。

分布式核心的业务发布系统和运维系统根据整体架构的分布式特点也做了全面重构。新的业务发布系统打通了资源管理平台和 DevOps 平台（基于腾讯蓝鲸平台构建）。业务发布过程中的配置管理，版本管理，路由策略管理，业务切换等已经集成到客户的技术平台。新架构对运维系统监控的颗粒度和关联性分析能力提出了更高的要求。新的运维系统在业务层面实现了全链路跟踪，在系统层面实现了全系统拓扑结构关联模型建立和展示。新的运维系统在复杂的分布式架构体系下保证了客户可以实现故障快速定位。结合运维知识库里的应急预案，客户可以采取故障快速处置、隔离等措施，保证业务的连续性。运维系统设计的理念和落地从传统运维的思路已经迈入智能化运维阶段。其中基于腾讯 TBDS 大数据套件建设的运维大数据平台通过不同场景下的模型设计和运维数据支撑，已经可以提前感知系统风险并给出相应处理建议。

分布式核心整体平台设计要点

分布式技术中台

为应用服务端系统提供全生命周期支持。在架构设计方面，支撑单元化、微服务架构，成为企业级架构最重要的能力载体；在开发方面，实现一站式低代码开发模式，大幅提高开发效率；在系统运行过程，提供了事务一致性保障、消息可靠传输等核心机制；在运维方面，平台提供了熔断、限流、切换等丰富的治理手段。分布式技术中台是基于腾讯 TSF 和 TDMQ 构建的。TSF 除了支撑以上所提到的单元化、微服务治理、统一配置管理、应用监控、路由策略等基本能力之外，还实现了和蓝鲸 DevOps 业务发布的整体端到端打通。通过统一

云管理平台基于资源拓扑模版通过自动化驱动实现业务单元整体发布。通过 TSF 实现微服务的本地服务注册，同时在本地注册中心的基础上，构建了全局注册中心来实现业务服务跨地域的注册和发现。

分布式数据库平台

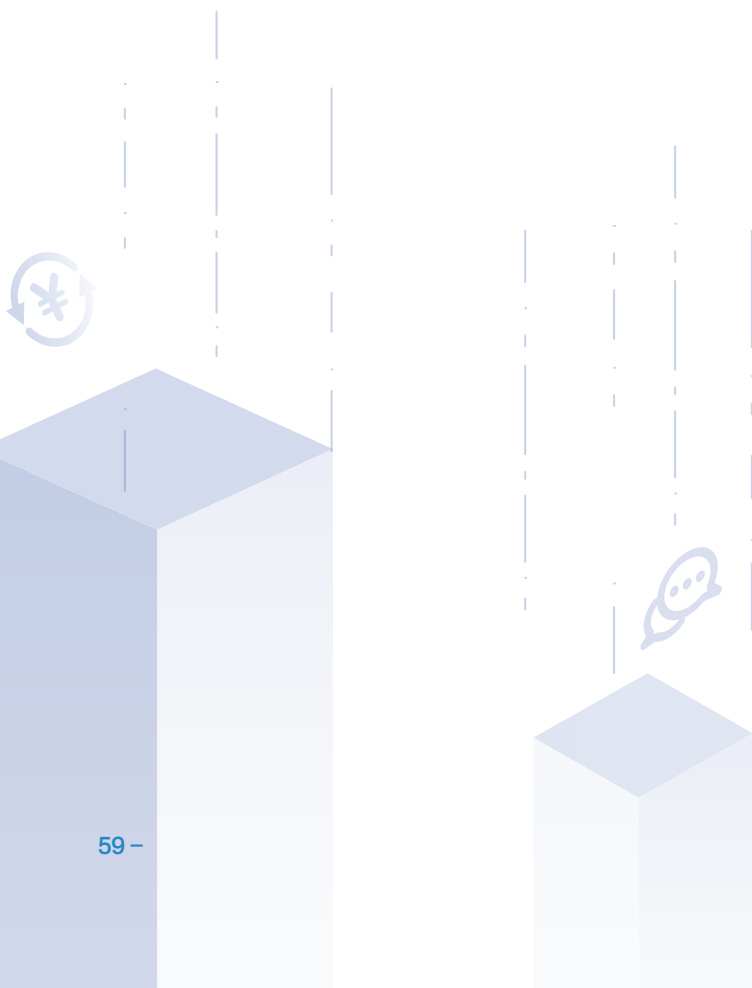
分布式数据库平台可以说是行内最重要的平台，行内原有支持核心系统的 Oracle，DB2 等国外数据库会随着主机业务下移被逐步替换。根据行内针对不同业务特点和数据规模，TDSQL 分布式架构和集中式架构都有大规模部署和使用，行方在业务投产前做了多轮带载（负载达到客户实际业务峰值的 2 倍）仿真验证，对 TDSQL 在极端情况下的高可用能力做了全面测试。测试结果显示在 1400 个数据库实例的规模条件下，进行计划内和计划外的数据库异地切换，切换过程可以再 3 分钟内完成。在模拟单数据中心故障，同城另一个数据中心接管所有业务的同城孤岛演练中，TDSQL 实现了 RPO=0，且在 3 分钟内 1400 个数据库实例全部完成切换。流量回切后，全部业务的平均响应时间以及成功率都能恢复到切换前的状态。TDSQL 的性能和高可用能力得到充分验证。

分布式运维平台

分布式运维平台的建设也是行内采用新技术挑战比较大的一个系统平台，分布式运维建设对比传统运维在思路和方法上会有一些比较大的区别。客户也在 ITIL 的建设思路上去逐渐适应分布式系统。比如故障出现后，传统思路是要快速定位故障并第一时间解决问题。而分布式运维平台处理问题的思路是根据故障类型的不同，采用直接隔离或者业务降级等方式进行处置，确保整体业务的连续性。分布式运维平台结合多云管理平台可以实现资源统一管理（IaaS 和 PaaS），以及租户管理，统一鉴权，资源编排 & 自动化驱动。依托 CMDB 的拓扑结构可以实现业务模块与资源模块相结合，实现全网资源透明化。

“分布式核心” 整体平台投产成果

行方分布式核心系统已经进行多个批次业务投产，投产的业务包括：客户信息、支付、借记卡、信用卡、存款核心等最主要的银行核心业务系统。目前行内投产的数据库实例数量超过 1400 个，在行方多个数据中心投产的 TDSQL 数据库物理节点超过 4000 个。



6.2

某国有大行核心系统案例

银行业作为关乎国计民生的重要行业，其核心系统是业务稳健发展和产品创新迭代的基础支撑。传统的集中式系统架构因缺乏快速有效的横向扩容能力、单体规模过于庞大，难以有效应对互联网类业务的瞬时高并发、大数据量、快速迭代等发展需要。

在此背景下，该国有大行于 2021 年启动了核心系统技术架构转型，系统架构由基于大型机与封闭式中间件的传统架构，转向基于云平台与分布式数据库的新架构，以大幅度提升了横向扩缩容能力和容灾能力，并针对当前新型业务瞬时高并发、大数据量、快速迭代等特点进行功能升级，高效支持产品创新及市场发展。应用采用微服务技术架构，数据库选用腾讯云分布式数据库 TDSQL，实现从应用到数据的线性可扩展及多活高可用。

建设方案

分布式核心系统技术上从集中式架构转向分布式架构，以“一体化云平台 + 分布式数据库”为支撑，采用分布式部署，涉及单元化设计、分布式事务实现、核心内外关联系统解耦和客户数据迁移。这些技术架构的变化对底层分布式数据库的技术架构设计提出更高的要求，整体方案关键建设内容如下：

数据拆分与分布设计

分布式核心系统采用分区多活架构设计，数据组织方式采用以客户为主的单元化多分片式，通过分布式数据库 TDSQL 支撑上层微服务应用单元化能力。

在单元数据分布上，分布式核心系统根据数据量、交易量，以客户为中心、兼顾机构维度确定数据分片规则，按水平维度进行数据表拆分。数据首先按客户号 / 机构号归属到不同的逻辑分片，多个逻辑分片多对一归属到不同的物理分片，多个物理分片多对一归属到不同的单元。业务流量先由 DNS 域名解析找到数据中心，在数据中心内部通过全局路由由中心路由访问相关业务单元，并根据客户号 / 机构号找到物理分片数据。

数据存储层采用 TDSQL Groupshard 大分片设计，由一套 TDSQL 分布式集群承载业务和单元化数据，对应用完全透明，大幅降低后期数据维护成本，同时从数据库层配合微服务弹性，实现核心业务更高效的线性按需扩展。

分布式事务实现设计

分布式核心系统采用单元化数据拆分后，分布式事务的实现设计主要包括如下几个方面：

对于因业务单元化而在应用层进行数据拆分导致的分布式事务（例如跨单元数据访问），由应用层分布式事务组件利用 SAGA 和 TCC 分布式事务框架保障业务层微服务间分布式事务一致性。

对于分布式数据库内部数据拆分产生的分布式事务由 TDSQL 原生的分布式事务能力保障数据操作的事务 ACID。

对于所有数据操作都落在单分片的事务操作，TDSQL 将优化事务路径，采用快速一阶段事务完成提交。

高可用与业务连续性设计

核心系统的业务连续性采用两地三中心架构设计。

单元层面设计“互备单元表”与 TDSQL 相结合来建立两个单元间的互备关系。当其中一方出现故障时，通过更新状态，来指向其互备单元 ID 从而进行路由调整，同时互备单元的数据库副本需要放置在对方单元中，当发生切换时，数据库会将对方单元下的副本调整为主本。

数据库层面，同城 TDSQL 集群采用两地三中心三集群，形成在线、近线、离线的多层次数据分布规划。每个中心各 2 个数据副本，集群内副本间基于 TDSQL MAR 数据副本强同步技术，确保副本和机房故障数据零丢失 (RPO=0)，全面提升数据应用和数据安全服务能力。

全链路智能运维方案设计

为了充分支撑分布式核心系统“监、控、管、配、析”一体化运营能力,保障系统生产运行的稳定性,TDSQL 结合 DBBrain 智能运维套件,针对分布式核心系统提供了数据库秒级监控和智能化运维,可以针对 SQL 运行的全流程耗时进行分析优化,实现数据库全链路分析。

TDSQL 内核针对监控进行了深度优化,在获取完整的数据库执行语句详情和数据库全局快照的同时,对数据库性能影响微乎其微。在此基础上,DBBrain 进一步打通数据全链路,提供 7*24 小时数据库异常发现、诊断分析服务,同时通过四层分析聚合,实现了业务多层、多维度的性能分析诊断,正反向追踪数据链路,与透视链路,有效保障分布式核心数据库服务的安全、稳定及高效运行。

项目建设成果

2023 年 4 月,该大行信用卡分布式核心系统成功投产,标志着该行迈入分布式核心系统大规模应用的新阶段。2024 年 11 月实现主机应用全部关停,全面开启分布式核心新时代。主机交易流量清零、账务清零、数据消费清零、主机应用全面关停。

TDSQL 全程助力该大行新一代分布式核心系统建设,具体的亮点成果如下:

- 一是该行新一代分布式核心是国有大行中首家通过使用数据库原生分布式能力支撑新核心系统建设的案例,其分布式实例规模业界领先,核心投产使用 TDSQL 大分片模式。
- 二是高扩展性,采用通用物理服务器,支持系统横向扩展,可以满足大容量和突发峰值等场景的需求,为亿级客户的经营与管理提供了安全稳定的数据服务支撑。
- 三是高容灾等级,保障业务连续性。在数据分布上,分布式核心系统根据数据量、交易量,以客户为中心、兼顾机构维度确定数据分片规则,按水平维度进行数据表拆分,采用“一主五副本”方式进行部署,并在数据全生命周期管理中形成在线、近线、离线的多层次数据分布规划,全面提升数据应用和数据安全服务能力,夯实以客户为中心的多级数据服务基石。
- 四是数据库易运维性,实现了数据库秒级监控和智能化运维,可以针对 SQL 运行的全流程耗时进行分析优化,实现数据库全链路分析。



6.3

某西南省级城商行全面云原生案例

项目背景

作为西部某省首家省级法人城市商业银行，该行于 2020 年 11 月通过合并重组正式成立，总部设于该省省会城市。该行以“打造全国一流城市商业银行”为战略目标，致力于建设成为一家管理规范、经营稳健、品牌卓越，在全国具有较强影响力和竞争力的优质上市银行，切实发挥服务地方经济社会发展的金融主力军作用。

鉴于重组设立初期实际情况，原 IT 系统采用“桥接”方案难以有效支撑业务快速发展需求。为此，亟需通过实施新一代信息系统工程，推进全行级技术架构升级，重点解决原有科技体系在系统扩展性、业务连续性保障能力等方面存在的瓶颈问题，全面提升科技赋能水平。

建设方案

本项目紧密围绕该行数字化转型战略部署，以保障金融级应用系统的高扩展性、高可用性以及关键技术自主可控为核心战略目标，通过构建“全景单元化架构+全栈信创云原生平台”双轮驱动技术体系，全面构建兼具稳健运行能力与敏捷响应特性的新一代数字化技术底座。具体方案关键设计点如下：

全栈信创

本项目严格遵循全栈信创实施原则，自底层硬件至云平台及上层应用均采用信创产品。底层硬件层全面部署信创设备，涵盖服务器、网络交换设备、路由设备及密码机等核心组件；云平台基于腾讯专有云 TCE 构建，包含基础设施即服务（IaaS）层采用一云多芯架构，平台即服务（PaaS）层集成容器服务 TKE、分布式数据库 TDSQL、微服务治理 TSF、单元化架构平台 TCUA、消息队列 TDMQ 及分布式缓存 CRedis 等核心模块；安全体系构建全栈安全产品矩阵，覆盖主机安全、网络安全、容器安全及安全运营中心等关键领域；上层软件即服务（SaaS）应用系统全部采购信创供应商提供的合规软件解决方案，形成完整的信创技术生态闭环。

全景单元化架构

本项目与腾讯联合研发基于单元化架构平台 TCUA 的系统架构方案，采用“两地四中心”部署架构，通过基于客户维度的数据分片策略及“单元内闭环”的架构设计，实现应用系统解耦及交易响应时效的显著提升，有效降低跨中心数据交互流量，灾难恢复能力达到近六级标准（RTO 接近分钟级，RPO=0），系统整体可扩展性与高可用性得到显著增强，全面保障业务连续性运行。

云原生技术体系

本项目依托腾讯 TKE 容器平台与微服务治理平台 TSF，构建具备可观测性、弹性扩展能力及服务治理能力的云原生技术体系。截至该行新一代系统成功投产，该体系已全面推广应用至核心系统群、信贷系统群、支付系统群、渠道系统群等近百套业务系统，实现云原生架构的规模化落地，显著提升系统运行效能与技术适配能力。

全链路灰度

本项目全链路灰度能力依托 TSF 微服务平台构建，具备降低版本发布风险、提升新功能验证效率的双重作用。该能力基于流量染色与灰度路由双策略机制，支持按照业务场景、地域等多维度实现精细化流量调度，并达到“小流量验证—快速回滚”的闭环管理流程，显著提升版本回退效率，有效降低资源错配风险。

DevSecOps实践

DevSecOps 实践作为数字化转型过程中构建内生安全体系的关键路径，是在现有 DevOps 平台基础能力之上，通过贯彻“安全左移”理念，系统性重构研发流程，实现安全能力在从需求设计至上线运维的全生命周期中的深度嵌入与协同融合，全面提升软件交付的安全性及合规水平。

全链路可观测

在银行业务持续优化、交易链路日趋复杂的背景下，特别是在单元化架构、微服务与容器化技术广泛应用的环境下，一旦发生系统异常，传统监控手段往往难以迅速定位故障根源，显著增加了运维复杂度。本项目通过构建一体化可观测平台，依托全局流水号关联机制、近实时数据采集能力及流式分析技术，打造端到端的全链路可观测体系，实现对交易链路状态的全面可视与精准掌控。

项目建设成果

2025 年 5 月，该行新一代信息科技工程建设任务圆满完成，并实现主体批系统顺利投产。该项目历时近 3 年建设周期，全面完成了涵盖核心业务、支付结算、信贷管理及渠道服务等百余套关键业务系统的架构升级与重构工作，整体建设成效显著。本项目主要成果如下：

- 系统性能方面，通过构建分布式微服务架构体系，采用容器化部署、动态资源调度技术以及基于单元化架构的横向扩展能力，结合 CRedis 缓存中间件与分库分表数据库架构，在新一代信息系统中，主要系统特别是核心系统的并发处理能力实现显著提升，较原有水平提升 28 倍。
- 研发效率方面，通过构建 DevSecOps 全生命周期管理平台，集成需求分析、自动化测试、安全漏洞扫描等功能模块，形成完整的辅助工具链。依托标准化 CI/CD 流水线，实现从开发、测试到生产的全流程自动化交付，整体研发效率提升 1.2 倍。
- 批处理能力方面，通过对批量引擎进行重构，采用单元化架构设计、分布式任务调度框架及弹性计算资源池，实现批处理任务的分片执行与并行调度，有效提升批量作业处理效率，核心日终批处理效率提升 4 倍。
- 容灾能力方面，通过建设同城双活与异地灾备相结合的单元化架构体系，系统容灾能力实现显著增强，RTO（恢复时间目标）已缩短至接近分钟级水平，大幅提升了业务连续性保障能力。

在充分吸收和融合上述科技成果的基础上，本项目系统提炼出一套具有行业示范意义和推广价值的方法论体系，并已获得相关权威机构的认可：

- 《全栈信创云原生平台：提升供应链安全水平》获得金融电子化杂志社第十五届金融科技应用创新奖。
- 《金融级单元化架构设计及应用研究报告》获得北京金融科技产业联盟 2024 年度优秀课题称号。

项目价值

通过本项目的实施，该行不仅实现了全行技术架构的整体优化与升级，有效破解了原有科技体系在系统扩展性及业务连续性保障能力等方面存在的瓶颈问题，还系统梳理并沉淀形成了一套涵盖标准技术方案、规范化实施流程及技术管理规范完整方法论，构建了一整套“可借鉴、可复制、可推广”的云原生单元化架构落地实施方案，为中小银行数字化转型提供了更多的实践参考。

6.4

某城商行核心系统升级案例

银行核心业务系统承载着存、贷、汇、银行卡、结算、客户开户、客户统一视图等主要业务和产品的交易处理。随着金融创新节奏的加快，互联网场景与金融的进一步融合，以及行业对安全可控诉求的进一步提升，传统核心的瓶颈逐渐显现。某城商银行在新一代核心系统建设中，充分考虑业务发展的中长期需求，选择“标准微服务架构 + 分布式数据库”的组合，支撑业务安全、高效、稳定运行。该系统采用腾讯云国产金融级分布式数据库 TDSQL，支撑公共服务、客户中心、账户体系、存款业务、贷款业务、借记卡业务、支付结算、现金管理、凭证管理、批量等业务模块，能够满足某城商银行未来 10 年客户及账户数量的发展需求。

该银行于 2020 年实现了分布式核心系统的投产，新核心系统整体处理能力可达到万级 TPS，可支持每日亿级交易量，高频帐户类交易平均响应时间比原有系统缩短 2 倍，日终批量时间缩短至分钟级，季度结息时间比原系统缩短 1.5 倍，性能远超原核心系统，在全国同类型银行中处于领先地位。

整体设计思路

某城商银行新建分布式核心系统，采用标准微服务架构核心系统无缝衔接国产分布式数据库 TDSQL，并融入微服务、读写分离、多源同步等的技术，实现在保证金融级数据全局一致性的基础上，把大系统拆分成小型微服务，以降低系统的复杂性，消除耦合，并有效解决了传统集中式核心并发量瓶颈，提升了核心系统的高可用性和动态扩容能力，同时大幅降低系统建设、升级、运维的风险和成本，实现了安全可控。

新核心系统有三个微服务集群：公共服务微服务集群、账务微服务集群和历史微服务集群。每个微服务集群由一系列功能职责单一、高度聚合的服务组成，可支持灵活部署。这三个微服务集群运行在分布式 TDSQL 集群中。在架构部署细节上，某城商银行的架构是“两地三中心”部署，分布式数据库采用一主三备，中心间数据强同步，实现中心级别灾难快速自动恢复，且数据零丢失。

项目技术方案基于标准微服务架构，客户数据切分策略采用客户号哈希、分布式分片架构，通过纯数据库的水平扩容，实现业务吞吐量的线性增长。机房内随机负载、优先同机房调用，分布式应用结合分布式数据库方式处理分布式事务，分布式数据库处理聚合查询性能优异。同城双活、异地灾备两地三中心架构保证业务连续性，实现实例级故障自恢复，中心级 / 地域级故障告警后可自动化决策切换。

分布式核心整体平台设计要点

本项目构建了分布式技术中台，专注于为分布式核心系统提供全面的支持，包括从架构设计到开发、系统运行和运维的全生命周期支持。中台支持标准化微服务架构，实现了一站式低代码开发模式、事务一致性保证、可靠消息传输、以及多种运维治理手段，成为企业级架构中最重要的能力载体，保障系统的高可用性和可扩展性。

同时，为了满足某城商银行对数据一致性、高性能、易扩展等方面的要求，腾讯云 TDSQL 数据库与新核心系统以及分布式技术中台之间做了深度适配和优化，充分结合系统架构和应用程序设计为某城商银行核心业务数据提供强一致性保障，并通过数据库多源同步等措施提高系统的实时交易性能，能够为某城商银行未来业务发展中的海量数据高并发场景提供强有力保障；其多中心多活架构特性，也能够支撑该城商银行构建完善的数据容灾能力，满足金融数据高标准容灾需求。

“分布式核心”整体平台投产成果

分布式核心投产共分多个批次，陆续投产业务包括 公共服务、客户中心、账户体系、存款业务、贷款业务、借记卡业务、支付结算、现金管理、凭证管理、批量等业务模块等主要核心业务下移到分布式架构体系上。目前投产的数据库实例数目已经超过 150 个，TDSQL 数据库物理节点超过 60 个，分布在行内多地多个数据中心。

写在最后

作为分布式技术转型和云基础设施的领先实践者，腾讯金融云团队在过去几年，与生态伙伴共同服务了上千家银行客户，包括国有大行、股份制、农信、城商行等。基于这些丰富实践，结合腾讯在国产云平台、数据库、云原生 PaaS 等一系列成熟产品的成功实践经验，沉淀了丰富的银行核心系统架构设计方法论。我们总结了如下几点优势：

具备全栈分布式与云原生的国产化技术产品体系，帮助客户实现自主可控国产化

包括：金融专有云 TCE、云原生 PaaS 平台 TCS、微服务平台 TSF、分布式数据库 TDSQL、分布式消息队列 TDMQ、分布式缓存 CRedis、研发效能平台 Coding、敏捷协同项目管理平台 TAPD 等。全面覆盖了银行云原生分布式核心系统所需的技术底座，并已助力国有大行，股份制，城商行，农商行等数十家银行核心系统成功投产，稳定运行。

可靠先进的产品力，给客户核心建设提供强力的支撑

经过十多年大量内部业务的打磨和丰富的项目实战经验，腾讯云产品矩阵已进入行业领先水平。

腾讯专有云 TCE 凭借和公有云一致的用户体验、成熟丰富的全栈云产品体系、多地多活架构支持多种高可用容灾方案、通过标准 API 广泛兼容三方应用、一云多芯兼容主流硬件等优势，在中国金融私有云平台软件跻身市场第一；腾讯 TCS 云原生 PaaS 平台在国产化优秀案例、金融场景容器集群性能、云原生中间件等产品均达到了业内领先水平；

腾讯云国产数据库 TDSQL，过去的几年里，在金融行业核心下移的场景中，帮助 20 余家金融机构完成关键核心业务系统的国产数据库替换；2021 年作为唯一数据库品牌入选《科创中国》先导技术榜，近 3 年时间多次入选 Forrester 领导者象限和 Gartner 魔力象限，多次助力金融机构获得中国人民银行金融科技发展奖。参考权威机构国际事务处理性能委员会 (TPC) 官网，腾讯云数据库 TDSQL 继 2023 年以每分钟处理 8.14 亿笔交易打破 TPC-C 世界纪录之后，2024 年再次斩获佳绩，以高出第二名 3 倍以上的得分登顶 TPC-DS 榜首。

技术开放与产品组件松耦合，给客户技术架构提供灵活的选择

腾讯是一家开放创新的公司，产品充分参考业界主流的技术架构，遵从开放的设计标准和协议，一方面可以容易衔接银行客户现有的技术架构，做更好的集成，另外一方面，也更容易被客户和合作伙伴掌握，便于后续的运维和产品迭代；另外，腾讯的产品在设计孵化之初，就充分考虑了产品的开放解耦，我们的IaaS、PaaS、数据库等各自解耦，相互独立，能适配不同平台底座交付。这些在各大行、中小行都得到了充分的验证，取得了良好的客户口碑。

丰富的生态合作体系，为客户提供可靠持久的服务

腾讯云的解决方案体系充分参考了多家头部核心应用厂商的专业方案，无论是在技术架构上的先进性还是开放性上，都已经得到市场的验证和客户的高度认可。腾讯云未来将继续聚焦打造自身的核心产品能力，在应用方案设计与技术平台的集成上充分考量生态伙伴的能力与特点，最终形成符合客户诉求的整体落地方案，为核心系统分布式转型的平稳落地保驾护航。

兼顾落地性的解决方案，给客户更科学可行的参考

腾讯云分布式核心解决方案由多位一线架构师和产品专家结合实际参与的核心系统建设项目提炼沉淀而成。方案紧紧贴合银行核心系统典型业务场景，融合关键设计要点，同时结合腾讯云实际参与的多家已上线的案例经验，给出切实可行的方案建议，为银行核心系统的规划与设计提供参考。

广泛的成功案例及专业的服务能力，为客户成功保驾护航

从 2015 年服务微众银行开始，腾讯云在银行核心系统领域已经积累了二十多家成功案例的经验，并基于此，培养了一支集咨询、规划、交付、运维于一体的专业团队，由众多银行核心领域和分布式技术领域的专家组成。团队具备坚实的技术积累和先进的架构设计能力，国内专家团队资源覆盖全国各省的重点城市，可以提供端到端的解决方案与技术服务支撑，保障银行核心业务系统的顺利投产。

核心系统分布式转型是当前时代洪流中银行业务和科技的共同诉求，我们期望与合作伙伴及银行各位核心领域的专家一起，并肩携手，砥砺前行，赢取数字化转型的胜利！

