

Tencent 腾讯

腾讯云

腾讯云商业银行核心系统 分布式转型白皮书



2023年5月

| 版权声明 |

©2018-2023 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 |

2023年5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）



前言

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》，金融科技作为金融创新的技术驱动，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。

商业银行在业务系统建设上，一直是电子化、数字化最为成熟且全面的行业之一。银行核心系统作为IT建设最为关键的应用系统，支撑着整个银行的商业运转，其业务系统的高开发、高稳定和存储数据的安全性等一直都是银行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，随着银行核心IT从业者和各类厂商的不断实践和经验积累，国产化的分布式银行核心系统已经成为未来发展趋势。

腾讯作为一家互联网技术公司，多年以来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，包括像神州信息、长亮科技、中电金信等优秀的系统服务商，大家一起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大家对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，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，同时也结合了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



1

PART

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 分布式是银行核心转型的必然路径 / 03
- 1.3 业内建设模式分析 / 05

2

PART

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 / 11

3

PART

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 / 15
- 3.2 客户如何选择 / 16

PART 4

腾讯云分布式核心—标准微服务架构路线

- 4.1 腾讯云银行核心标准微服务架构 / 18
- 4.2 腾讯云银行核心标准微服务架构的设计要点 / 19
- 4.3 落地实践案例—某城商行核心系统 / 29

PART 5

腾讯云分布式核心—单元化架构路线

- 5.1 什么是单元化架构 / 31
- 5.2 单元化的价值 / 32
- 5.3 单元化的落地挑战 / 33
- 5.4 腾讯云分布式核心—单元化架构解决方案 / 35
- 5.5 落地实践案例—某国有大行核心系统 / 45

PART 6

典型银行分布式核心系统应用架构与实践

- 6.1 腾讯云与神州信息的深化合作 / 48
- 6.2 腾讯云与长亮科技的深化合作 / 50
- 6.3 腾讯云与中电金信的深化合作 / 52
- 6.4 落地实践案例—多地城商行核心系统升级 / 55

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，也被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，再将我们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

在上一章，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教育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：

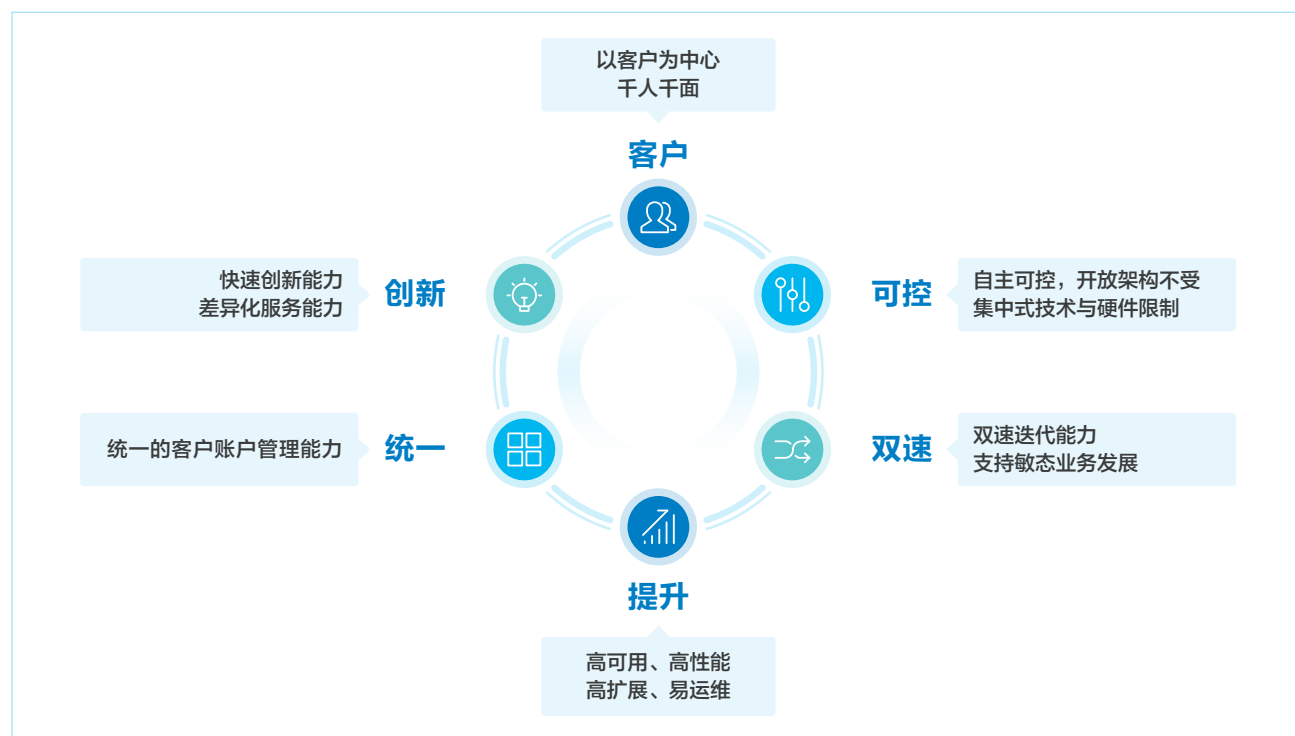
集中式架构的最大问题是相关技术体系掌握在少数一两个国外厂商手中，整个技术体系是封闭不开放的，这是目前行业共识必须要进行的改变。

集中式架构对硬件依赖较高，边际扩容成本高效益低，并非可持续发展的架构。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：



新一代核心业务系统的主要特点从业务层面首先需要实现以客户为中心、千人千面的设计理念，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，通过千人千面的客户标签为银行提供全方位的决策信息。

其次是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，实现产品创新的快速参数化配置。差异化的定价功能，可以为银行提供便利，使其弯道超越，将成为未来银行竞争的重要策略。

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。

在技术层面则率先需要具备业务连续性和稳定性。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。

其次是自主可控性。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地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。

接着是双速迭代能力。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。

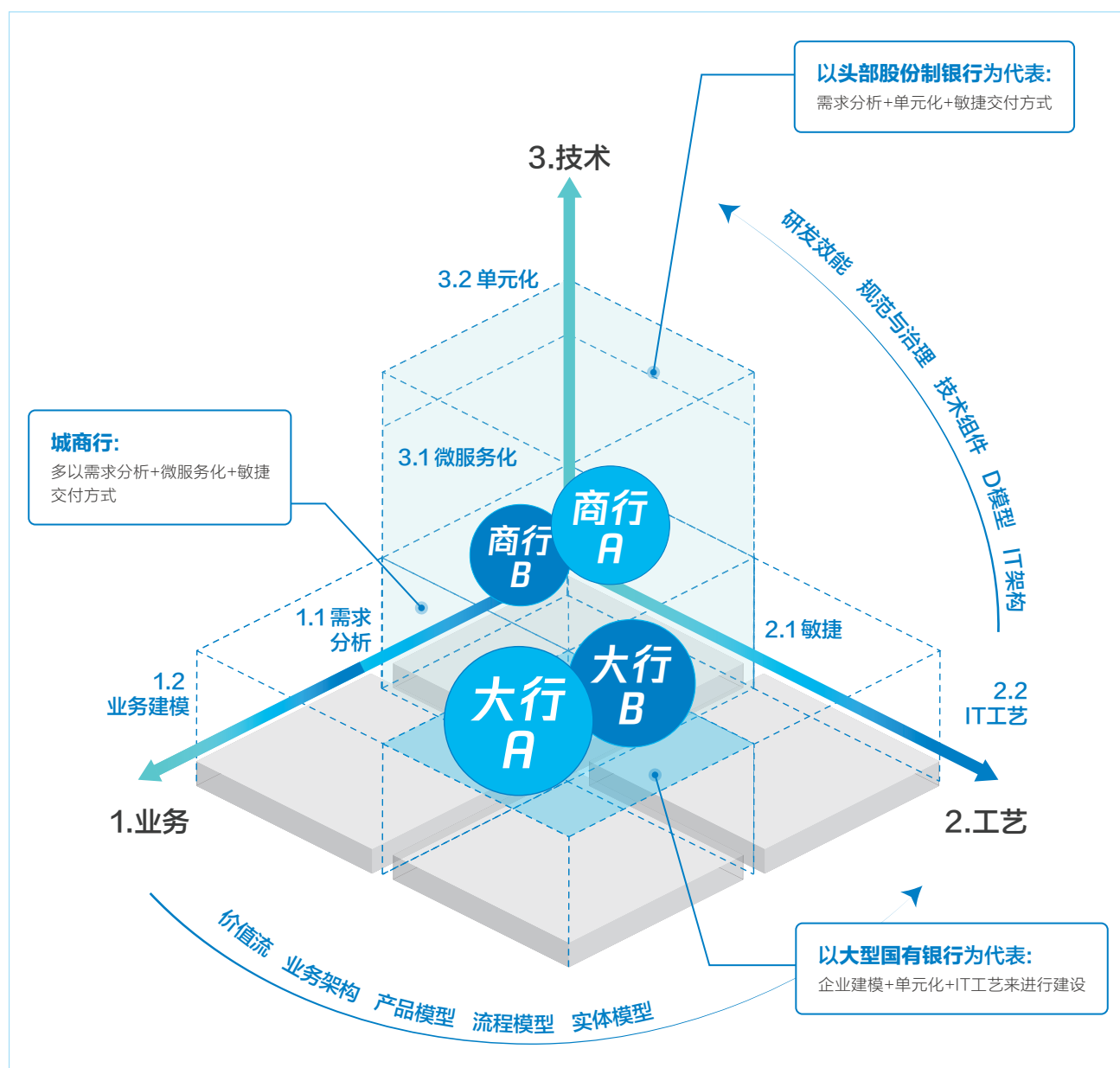
核心系统的运维是一个长期工程，必须具备良好的可观测和易维护特性。各交易链路端到端可追溯，可以进行统一的日志管理与检索分析，同时要保证核心系统的运维简便高效。

这些技术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。



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。



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。

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。





银行业分布式核心六大设计要点

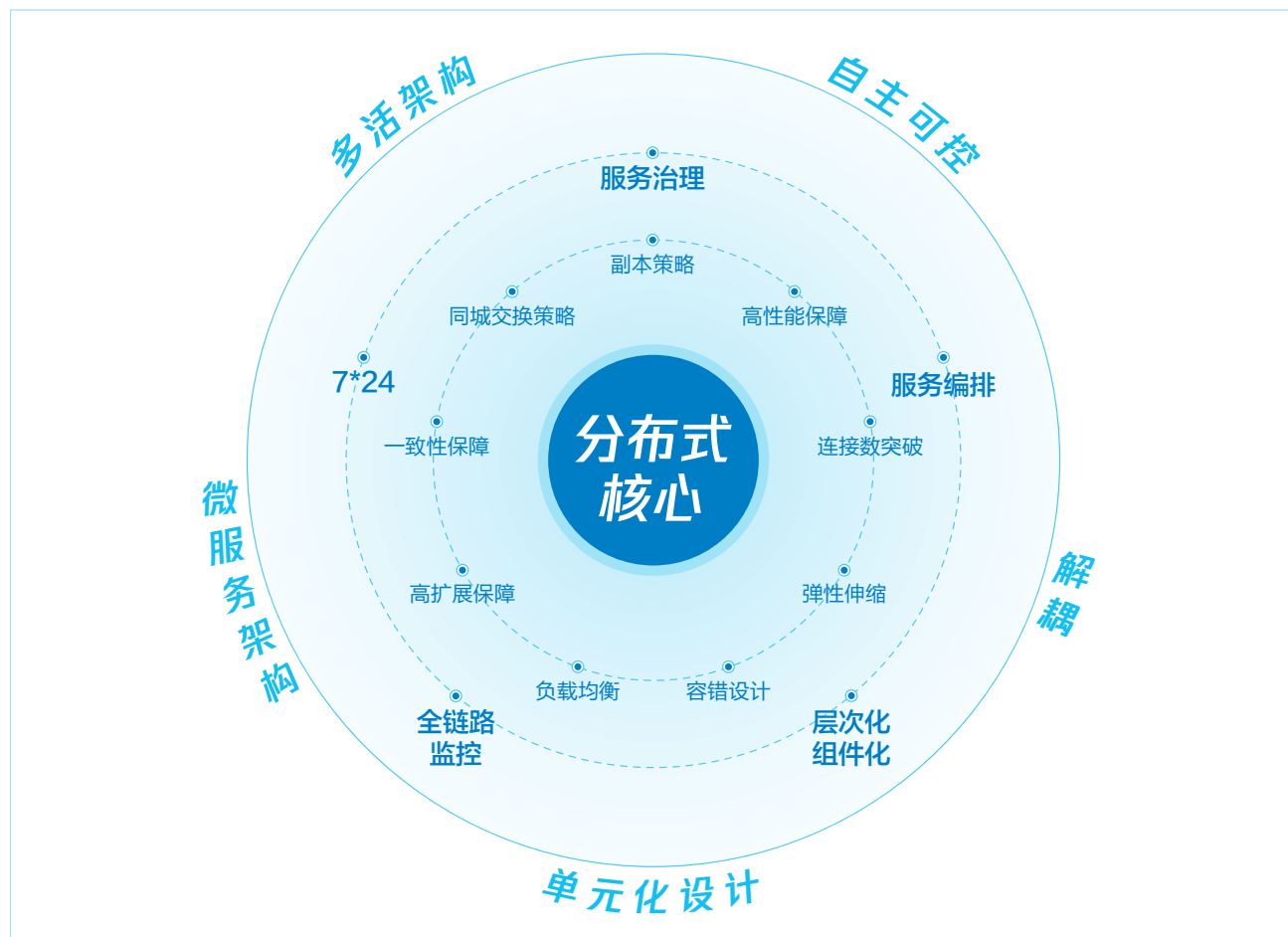
腾讯结合众多银行分布式核心实践，综合提炼出百余项公共技术能力，并联合专家多轮推演论证、梳理沉淀，形成新一代银行分布式核心整体架构。

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，以及单元化和标准微服务两种架构。



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，需考虑和应对各类困难与挑战。

接入层	服务层	数据层
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。	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。	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

在腾讯成立之初，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，在技术架构上都面临着**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和海量互联网业务，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，也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”

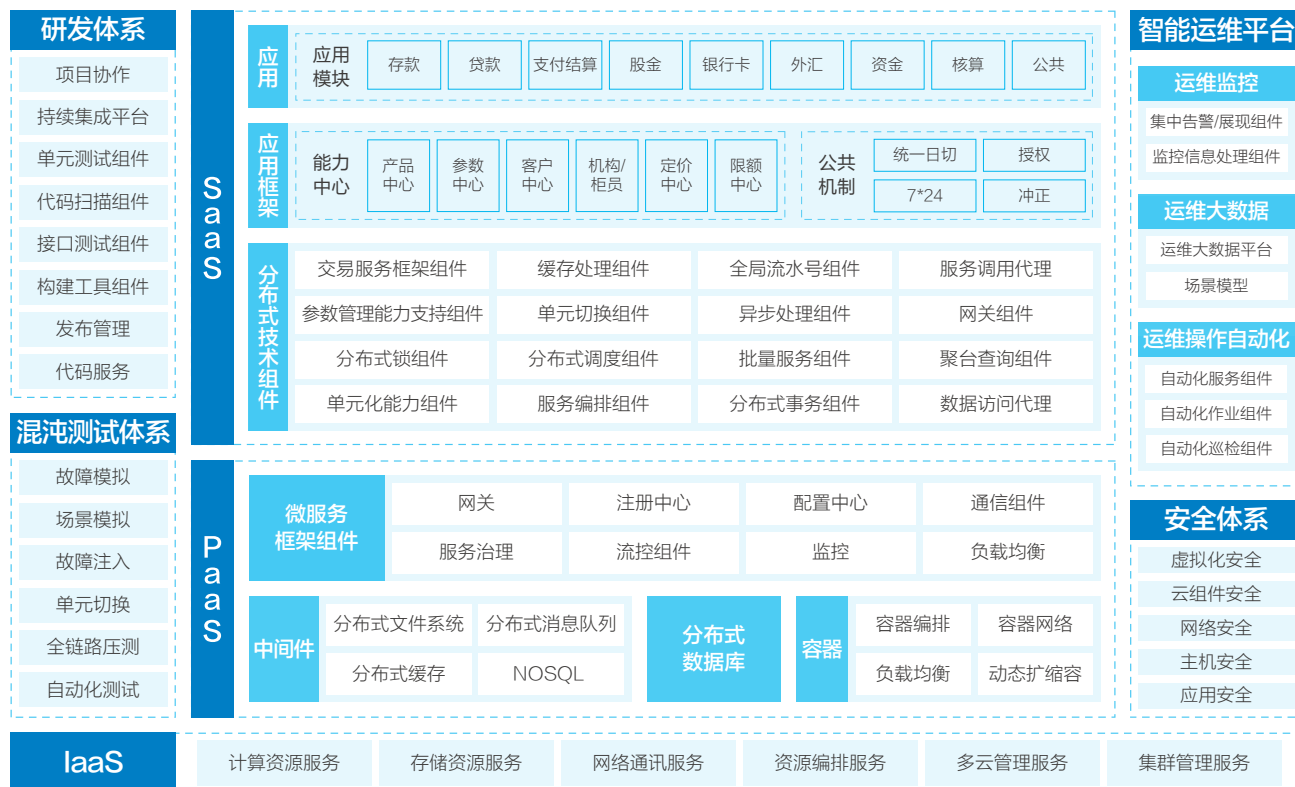
1 SET模型 <ul style="list-style-type: none">• SET化部署• 业务部署标准化，统一规则• 容量按SET动态伸缩	2 全网调度 <ul style="list-style-type: none">• 业务全网分布• 业务自协调自适应，弱化故障影响• 灵活控制业务切换、启用，灵活调度	3 柔性可用 <ul style="list-style-type: none">• 基于资源和策略对业务分层分级• 优先保证核心业务服务质量
4 灰度发布 <ul style="list-style-type: none">• 通过灰度小范围进行版本发布• 充分验证功能复合预期后，扩大发布范围，完成变更	5 过载保护 <ul style="list-style-type: none">• 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 <ul style="list-style-type: none">• 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 <ul style="list-style-type: none">• 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专家多轮经验推演与论证，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。



如上图所示，整体架构可分为七大部分内容。底层是IaaS云平台，中间是PaaS云原生平台，上层是SaaS核心应用领域，左侧为研发效能体系和混沌测试体系，右侧为智能运维体系和安全体系。上图七块内容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。及所需的能力中心和7*24等公共机制。分布式技术组件用于封装分布式底层技术，负责对上层应用模块屏蔽产品差异与技术复杂度，同时连接着下层PaaS平台的各项能力。其中，一部分与云厂商的产品存在一定重叠，如：单元化能力组件、分布式事务、聚合查询组件、服务调用代理等。在实际落地过程中需要明确双方边界，进行有效的整合与联动形成一体化的、稳定的技术平台底座。

IaaS层主要包括了云平台的计算、存储、网络、资源编排、多云管理和安全能力。在PaaS层以云原生容器平台、分布式数据库、分布式消息队列、分布式缓存、微服务平台等中间件为主。

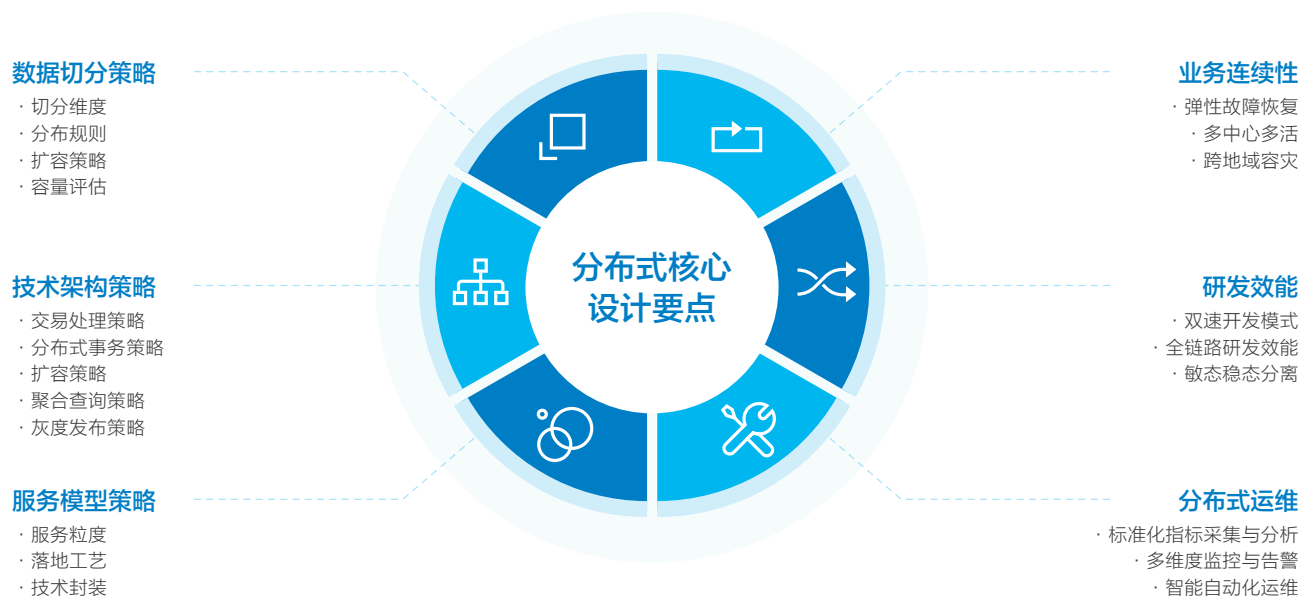
研发效能体系中包括了传统的Devops体系，同时还集成了项目管理和质量管控体系。以需求为抓手，贯穿设计、开发、测试、安全、发布、生产全生命周期管理，每个环节形成有效的反馈机制，确保在每一轮迭代都能有效量化生产数据，实现精益管理。

从传统集中式架构到分布式架构演进过程中，“面向弹性容错”的设计思想贯穿始终。混沌测试体系就是针对分布式架构下高故障率的基础设施和被放大的网络不稳定等因素进行提前暴露，来验证应用系统整体的容错和自恢复能力。

面向分布式的运维体系，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”。大规模集群中，硬件故障和网络抖动时有发生，自动化的运维能力成为标配。同时结合大数据算法实现故障预防是下一阶段的发展趋势。

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，及单元化和标准微服务两种架构路线。



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

在扩容策略方面，业界有两种常见的做法。一种是采用自研的扩容工具，另一种是通过数据库产品的能力来实现扩容。具体取决于技术架构和产品选型，如果数据库采用了多个独立的集中式数据库，那么就更适合采用自研的扩容工具。而如果采用了一个分布式数据库实例，并且内部划分了多个不同的数据分片，那么就可以通过数据库自身的能力来实现扩容。因此，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

决定了数据存放到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 >>

服务颗粒度

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

银行机构越来越重视核心系统的高可用性。同城双中心双活架构是最常见的设计，即在同一城市的两个机房同时接入并响应业务请求，一旦某一机房发生故障，另一个机房将接手处理业务请求。大型的银行机构规划采用异地多活的架构，即在异地多个机房中同时接入并响应业务请求。然而，与互联网业务不同，银行核心业务的异地协同是具有挑战性的，例如服务目录在异地间同步的时效性、跨异地的分布式事务协调等问题。

异地容灾

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。在分布式系统中，银行通过定期的灾备切换演练，或将允许的只读交易发送到灾备机房运行，又或者将可控的灰度单元放至灾备机房中运行，以确保和验证灾备体系的日常可用。同时，容灾机房也需要满足一定距离等要求。

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可自治的特性。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、发布三板斧，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。



腾讯云分布式核心两种技术架构路线介绍

结合腾讯参与的多个新核心落地案例，综合多家客户的调研访谈，总结归纳为“三维六阶模型”。

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，让微服务中无属性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，那可能就要回到从前ESB的时代，通过企业总线进行服务间的互通。不采用单元化架构，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，其只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上。

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

第四、第五章就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，由于两种路线在研发效能和分布式运维上差异不大，相关内容便不重复赘述。



PART

4

腾讯云分布式核心—标准微服务架构路线

本章将从核心应用厂商的视角，介绍业界常见的银行核心系统微服务架构；并综合第二章的六大设计要点，总结提炼标准微服务架构的设计要点。



4.1 腾讯云银行核心标准微服务架构

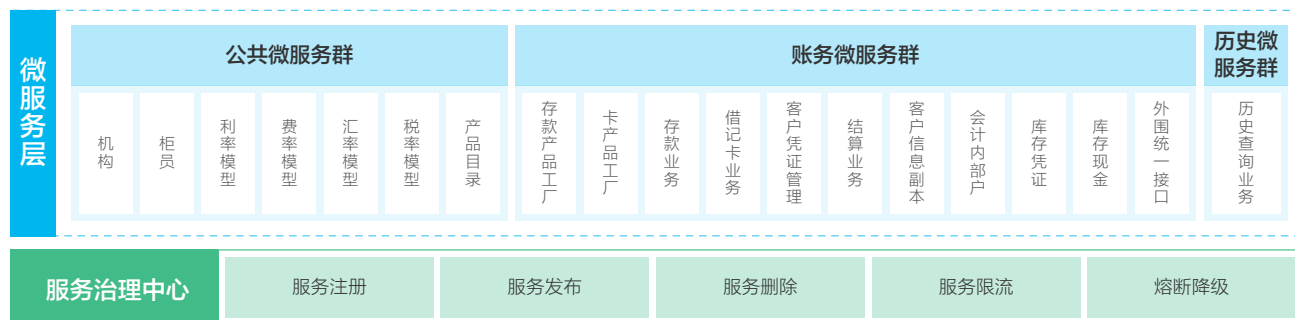
本节主要从核心应用厂商的视角，介绍业界较为常见的核心系统微服务架构。

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有不同，结合行内核心系统现状后也会有所调整。

📁 账务微服务群	👤 公共微服务群	🕒 历史微服务群
主要包含银行核心业务所涉及的账务服务模块，如：存款、贷款、卡业务、支付结算、客户凭证、库存现金与凭证管理等。	主要包含支持核心业务所需要的公共服务模块，如：机构、柜员、利率、费率模块等，以及像产品中心、客户中心、定价中心等能力中心服务。	主要包含历史查询业务所需的相关服务。



4.2 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章节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。

4.2.1 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核心业务特征来确定。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。

❏ 典型问题：以客户号作为数据分片维度是不是唯一方法？

答：并不是，只是在核心业务场景中单客户的交易场景居多，未避免产生分布式事务所以大多以客户号作为数据分片的维度。当然也有按地域和机构的维度来切分数据的，这种切分方式一定程度上解决了核心场景中非客户维度的查询问题，但会导致潜在的数据分布不均以及访问不均的风险。

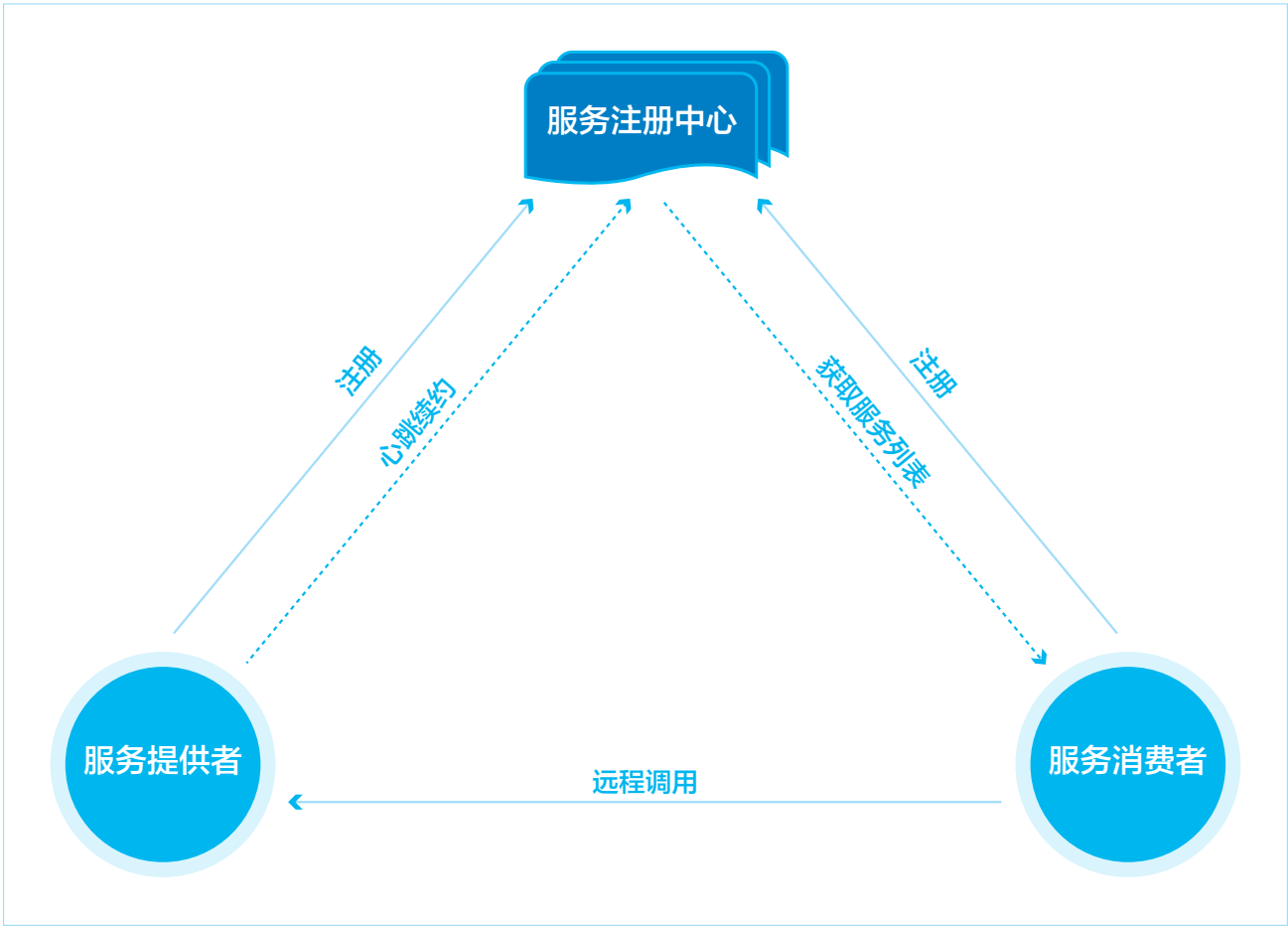
设计切分策略时需要同步考虑未来的扩容策略，针对微服务架构路线，可直接采用分布式数据库自身的扩容能力实现数据层的扩容。无论用哪种方案原则都是在扩容时要尽可能的减少数据迁移的规模。在银行核心场景中，我们建议充分评估未来业务增长量，结合单个物理分片能承载的业务量基线，评估能满足未来5年的规模容量，尽可能避免数据迁移场景的产生。



4.2.2 技术架构标准 >>

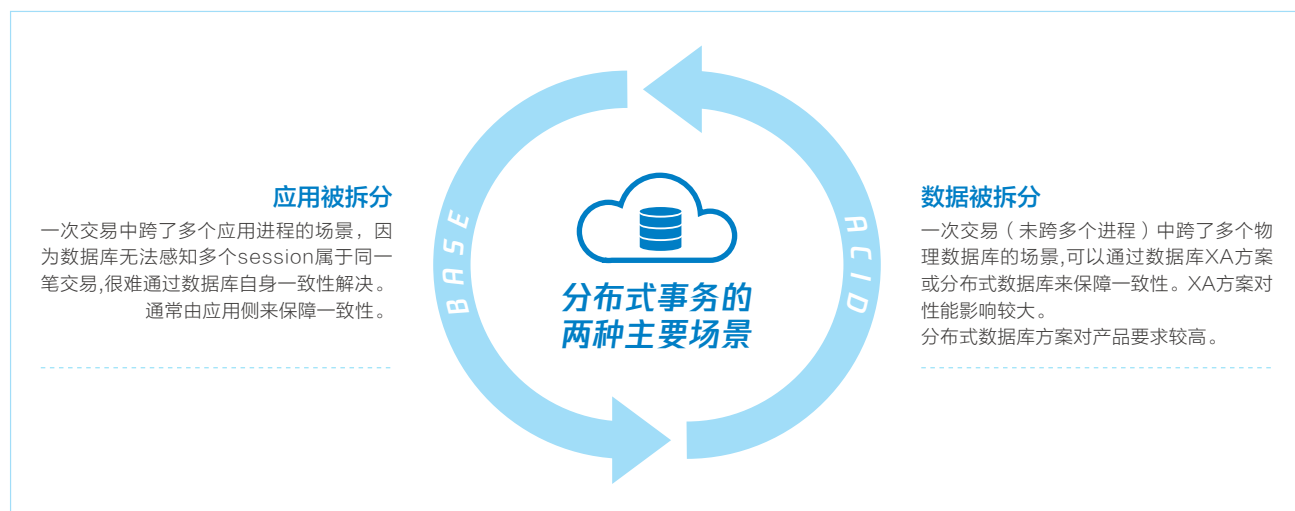
交易处理策略

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。负载均衡算法需要支持主流的几种策略，包括：随机、轮询、最小连接数等。



分布式事务

分布式事务是集中式系统微服务化之后不可避免的问题，在金融领域显得尤为重要。然而目前为止在微服务体系中尚未统一分布式事务的处理标准。要解决分布式事务，首先要明确产生分布式事务的原因，一种是由于应用被拆分后导致的分布式事务，从前由应用系统与数据库的一次会话被拆分为多次会话；另一种是由于数据被拆分后导致的分布式事务，从前由单一数据库保障的事务被拆分到多个不同数据库中。后者可以采用分布式数据库自带的分布式事务机制保障，而前者则需要由应用侧保障。应用侧分布式事务的处理模型主要有：SAGA、TCC、AT、事务性消息等。



综合整体架构来说，分布式事务最好的处理方式是“恰好不用解决”。即通过合理的应用规划和数据切分来有效地避免分布式事务的产生。比如用客户为维度切分数据就可以很好的减少分布式事务的产生，因为银行核心的交易场景多以单客户交易为主，将一个客户的所有数据都划分到一个库中，则多数场景便只要通过数据库本地事务即可保证。另外，所有的分布式事务处理机制都会带来不同程度的损耗，无论是在应用侧实现还是数据库侧实现。

典型问题：针对银行核心场景中推荐用哪种事务处理模型？

答：传统银行天然具备正反交易的设计思路，所以业界落地案例中较多采用SAGA模型。由于TCC对应用设计有一定侵入性且实现难度和成本相对较高，仅在一些关键场景中使用。同时外围一般业务系统可考虑采用AT自动冲正模型，而事务性消息常被使用在一些特定的业务场景，比如积分登记。实际落地过程中，在数据库性能和容量允许的情况下，也会考虑将相关微服务组件对应的数据库进行合并，来减少因为数据拆分而产生的分布式事务，应用层也会考虑将关联的服务组件合并成一个微服务组件进行发布，以减少因为应用拆分而产生的分布式事务。

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。

服务编排

通常我们把核心系统的服务分为两层，第一层为敏态服务，第二层为稳态服务。敏态服务即组合服务层，稳态服务即原子服务层。通过服务编排组件将多个原子服务进行可视化编排实现组合服务。比如“转账”服务就是在组合服务层，通过服务编排能力组装了“取款”和“存款”两支原子服务。其将转账的报文进行拆解映射成取款和存款两个原子交易的报文，并按照配置规则分别对齐发起调用，将返回的两支报文再映射组合成转账的返回报文对外返回结果。该组件与场景绑定较深，通常由核心厂商的技术平台提供。

4.2.3 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，但如何保持一个合理的微服务颗粒度一直是业界讨论的焦点，因为服务颗粒度过粗则无法体现微服务的敏捷和自治，但服务颗粒度过细则又会加大分布式事务出现概率和运维复杂度。在标准微服务架构中，基本以 ISV 核心系统提供的服务领域为基准，结合行内业务现状与规划进行微调。服务颗粒度会以应用内置的服务颗粒度为准。

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。



4.2.4 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：

设计左移

分布式架构下，基础设施本身具有不稳定性，需要从依赖基础架构的设计思维向应用健壮性导向的设计进行转变。

服务治理

充分利用分布式框架的优势，从服务治理的角度提升系统整体的可靠性，甚至具备抵御恶意攻击、流量激增等传统容灾与高可用设计无法涉及的领域。

主动监测

充分集中式向分布式的转换，导致系统节点数量呈几何级的增长，传统事后补救方式已不能满足业务需求，主动监测、故障隔离、服务自愈等能力需着重进行考虑。

自动处置

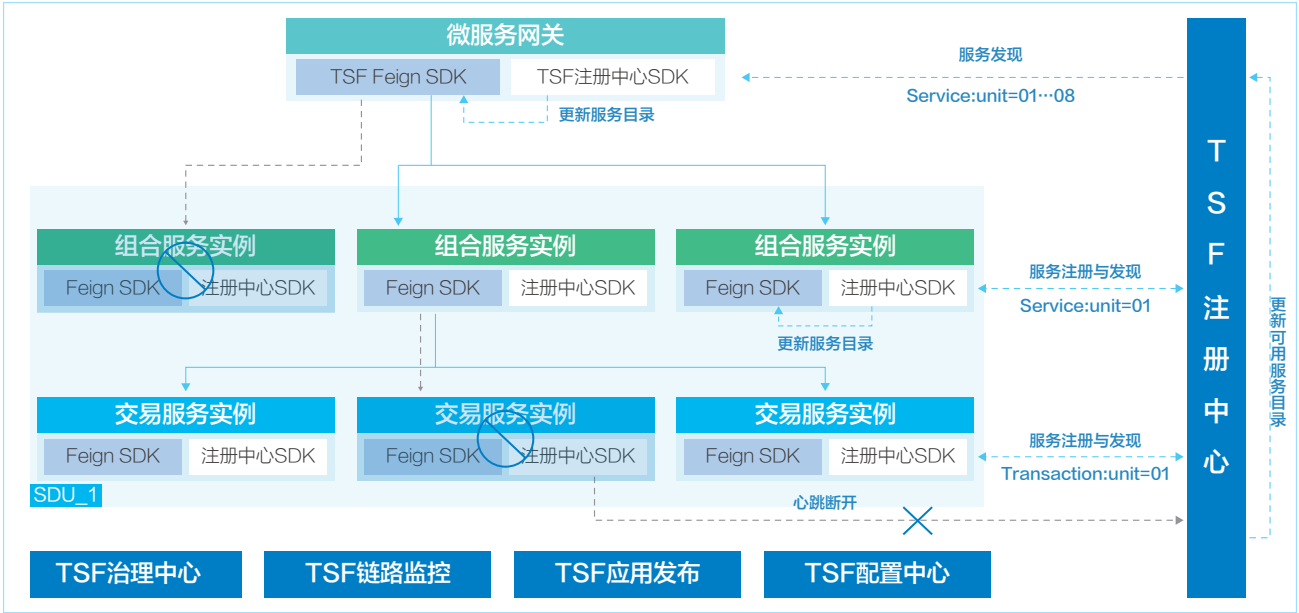
结合自动化平台，实现分布式环境下，海量节点的容灾切换，以满足RTO、RPO目标。

部署解耦

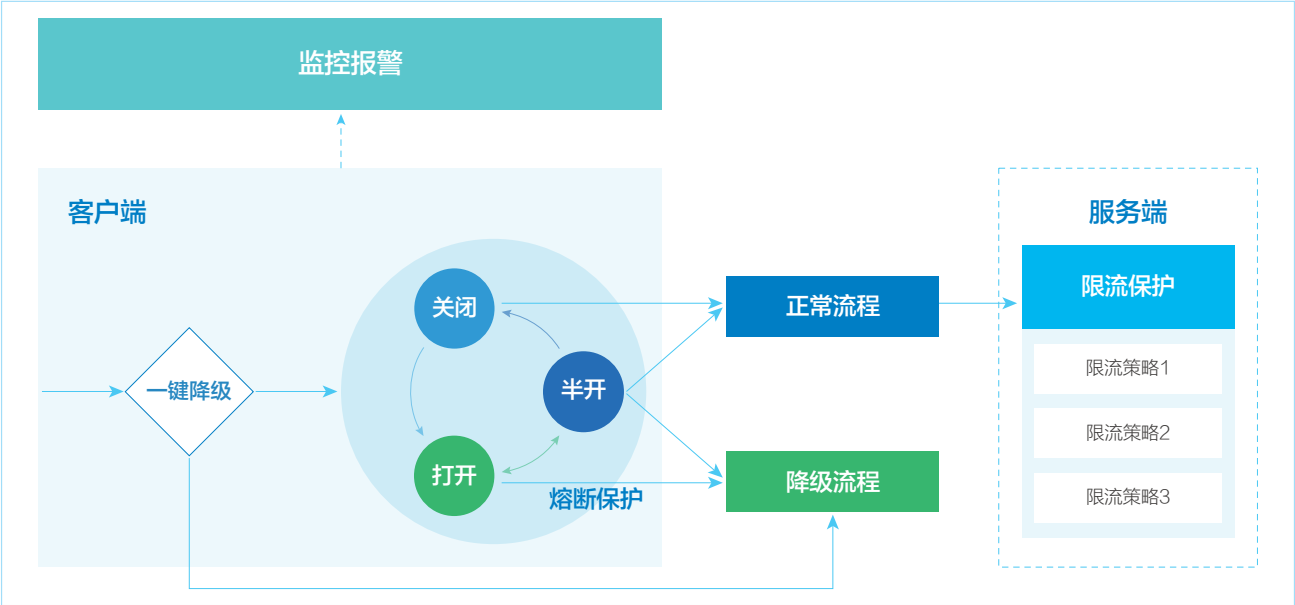
部署架构与基础环境基于反亲和模式解耦，规避底层设备与故障对业务造成的影响。

业务连续性包括数据中心内的连续性、同城及跨城市的连续性，下面将一一介绍。

实例级故障自动隔离



通过统一的服务注册与发现体系是目前在分布式架构中实现实例级高可用的最常见方法。各应用实例通过在启动时将自身注册到注册中心，并从注册中心订阅其他服务来实现服务间的点对点通讯。在这个过程中，注册中心维护了整个系统的所有服务信息，并负责下发给前来订阅的服务实例。当有实例宕机时，其与注册中心的心跳断开，注册中心会更新该服务的健康状态，更新服务目录并下发到各个订阅方。订阅方再次发起调用时便可以在本地过滤掉不健康的实例后再进行负载调用，以此实现故障实例的快速隔离。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、限流的措施：



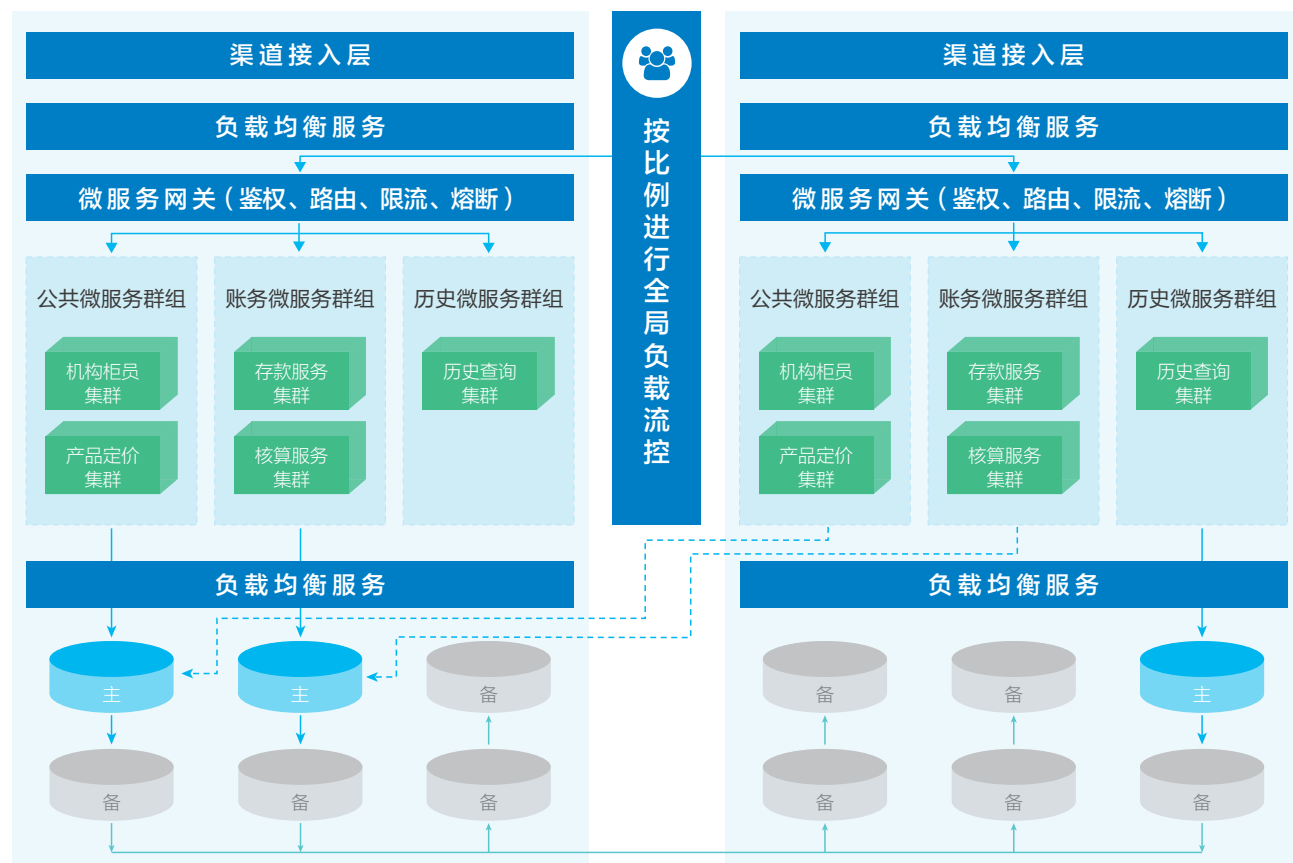
在业务规划时，提前识别强弱依赖业务，流量激增时，通过平台一键限流让弱依赖业务直接进入降级，不发生网络调用，节省资源消耗，保证强依赖业务。客户端需要具备“一键降级”功能，即交易维度的黑名单，在黑名单中的交易直接进入“降级流程”。

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，应用只需连接本单元的数据库代理进行访问。此方案中不会像单元化架构中把数据分片，再把不同的逻辑分片散列到各个数据中心。所以对于一类业务的数据主本都在主中心部署，应用在两个中心中分别部署。

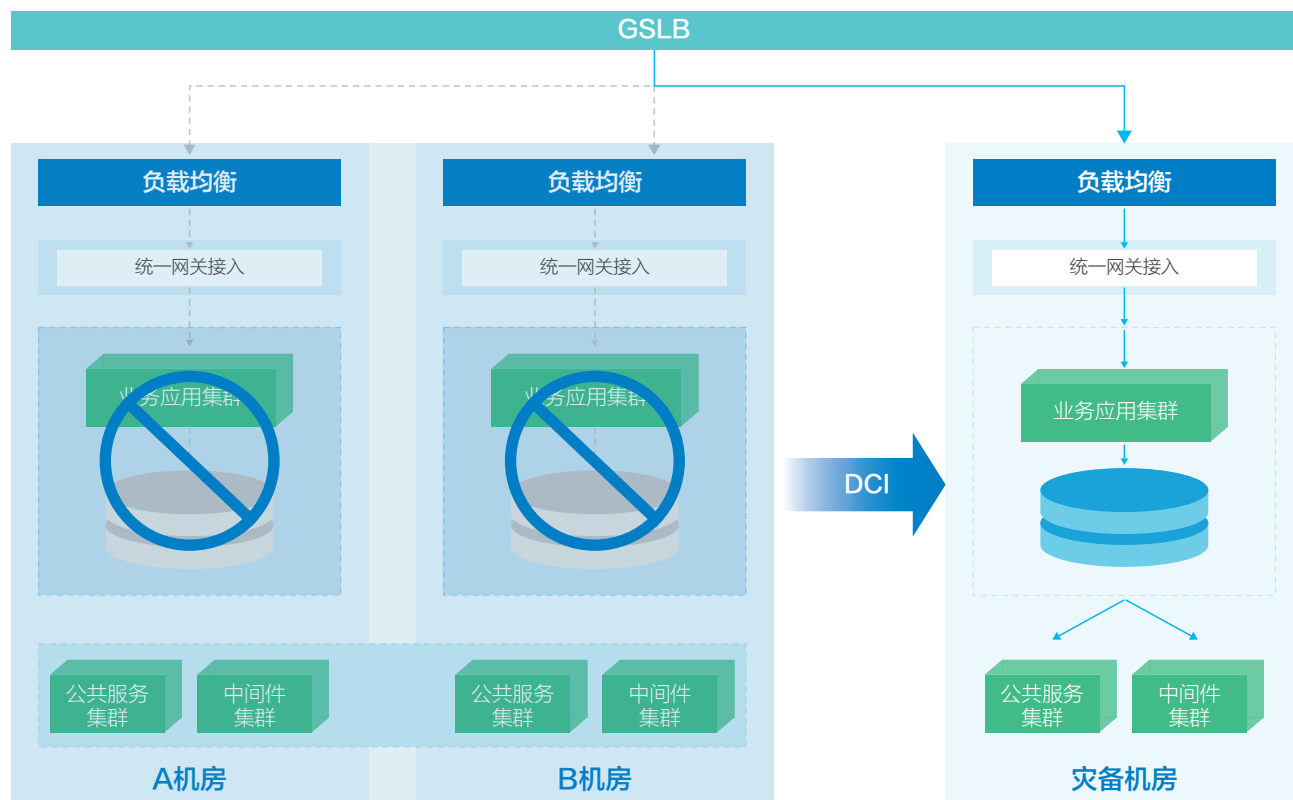


如图，流量通过顶层接入，经过 DNS 到中心内负载均衡服务再到微服务网关，经过鉴权、限流等一些列检测后进到微服务业务集群，处理时两个中心的应用都连接本机房的数据库代理进行访问数据，但在数据库代理这一层会根据路由算法计算客户所在的数据主备位置，并发起转发，这个过程中同城机房的请求将出现横向流量访问主中心的数据主本，这个过程应用是不可见的，因为分布式数据库对上屏蔽了底层的数据库架构和数据路由。以此实现同城两个数据中心同时处理业务请求的双活架构。

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

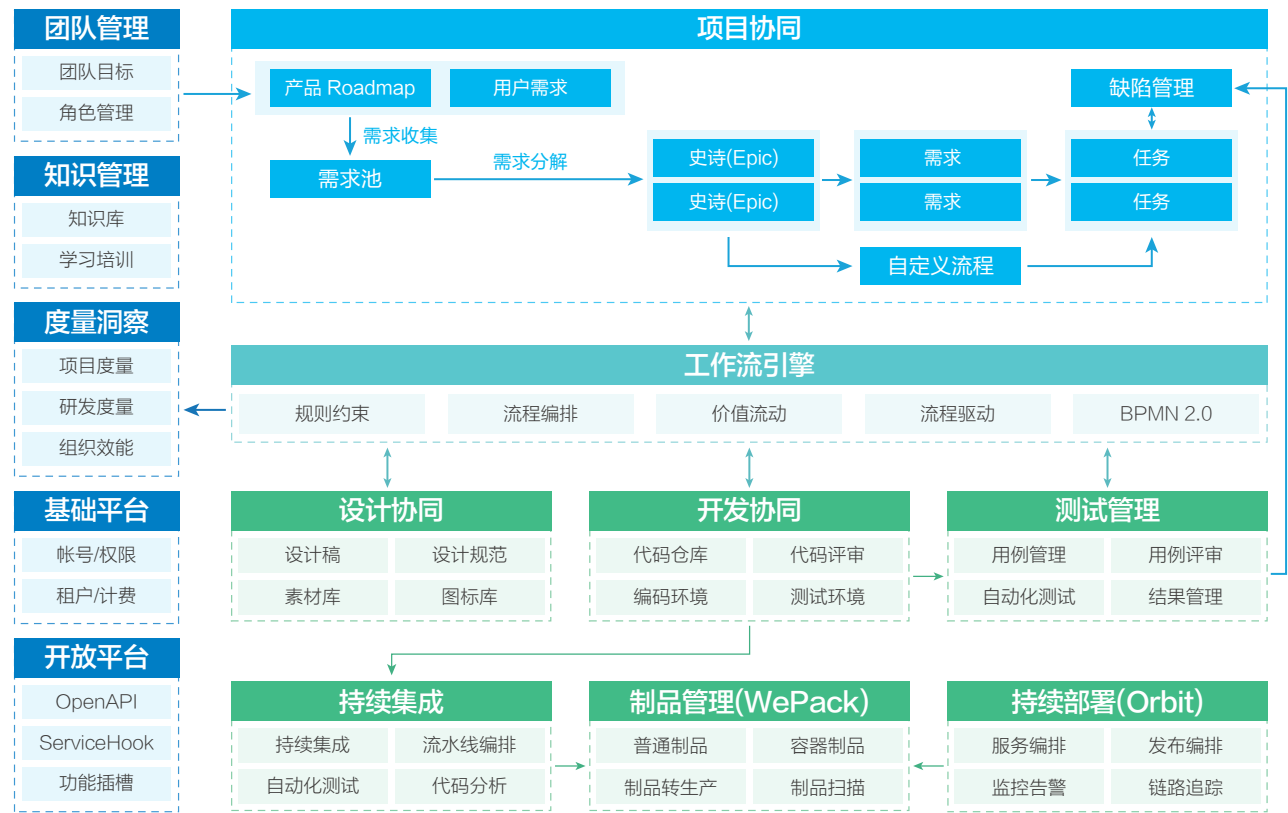
在分布式应用系统中,应用的无状态属性使得扩容和迁移都变得方便许多,我们只需要考虑有状态服务,站在核心系统视角主要是数据库。数据库的 RTO 和 RPO 是整个核心做异地切换的主要关键指标。在大型复杂的银行核心系统中,会涉及到多个业务数据库,且每个数据库下又会有多个数据分片,由于异地间的数据传输是异步模式,其切换过程比同城切换要复杂许多,会涉及网络状况的判断,数据补偿与校验等。



当数据库切换完成后,容灾中心的应用集群连到切换后的数据库,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,且与客户实际运维环境有关,此处仅做概要性说明。

4.2.5 研发效能体系

随着越来越多的银行客户启动新一代分布式核心系统的建设，与之配套的研发效能体系逐渐受到关注，在长期大型复杂项目的实施交付过程中，日常需求、设计、研发、测试、运维等形态将会十分复杂且难以管理。在传统的企业组织架构中，根据职责来划分各个部门，普遍存在筒仓效应。团队间以流程化方式进行协作，但也存在着较高的沟通壁垒，在资源协调的过程中，也会出现若干问题。

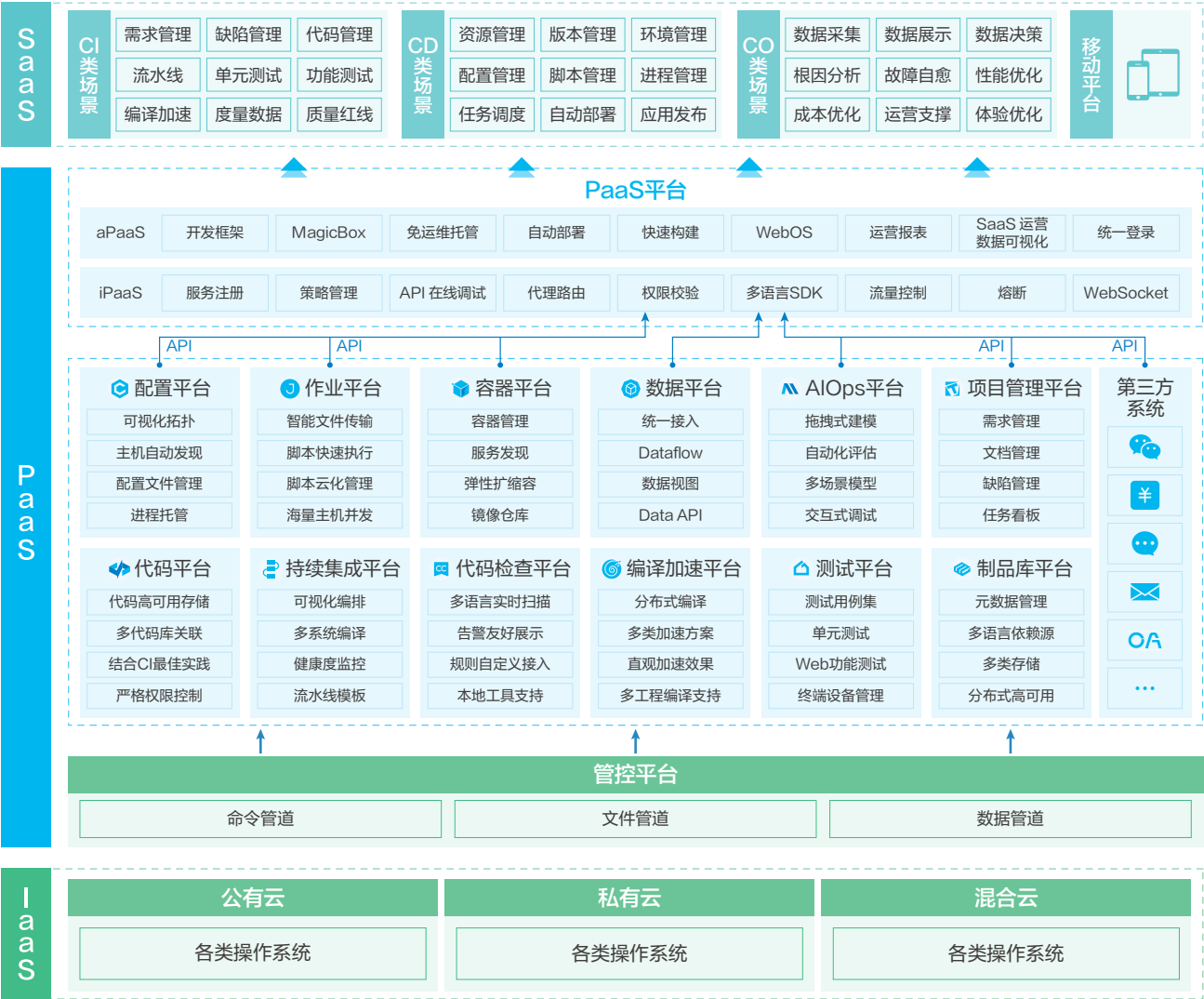


业务需求的快速变化对核心系统的高质量快速交付提出更高的要求，研发效能平台就是为了提高软件研发效率，快速应对变化，持续交付价值的一系列理念和实践。利用平台打通研发过程中的工具链孤岛及写作壁垒，覆盖敏捷开发全生命周期，自动扫描代码缺陷和漏洞，持续提升代码质量。流水线无缝衔接制品管理，随时拉出稳定制品进行发布。结合项目工作流，来满足核心及外围业务场景的协作需求。提供数据看板，支持对代码、项目进度、人员工作量等不同维度的度量与可视化，为团队管理者提供决策依据，调整项目计划和合理安排人力资源。以快速反馈机制来迭代优化整个生产过程，降低各环节的损耗，最终实现精益化生产。

4.2.6 分布式运维体系

为了能够支持各类不同的应用场景以及提供不同级别的高可用性、高性能、可扩展性、一致性等，分布式系统通常具有极高的复杂度。在复杂实际生产环境中运行时，分布式系统往往伴随着各种无法预料的突发故障，导致系统服务响应延时、数据计算出错、不一致/丢失，甚至服务崩溃等问题，从而带来无法估量的损失和灾难。另外，随着微服务化、云原生、敏捷开发的快速普及，在快速迭代业务需求即时变化的同时，也面临着分布式系统复杂度急剧上升，故障频发，业务连续性难以保证等问题，这使得分布式系统的稳定性受到了前所未有的挑战。因此，分布式系统的稳定性建设至关重要，即探索如何提高分布式系统在异常情况下的稳定性，避免因为各种故障带来的风险，进而为用户提供高稳定、高品质服务，成为分布式系统工程实施过程中的重点关注内容，针对分布式架构下的运维体系是分布式系统平稳运行的保障基础。

分布式运维体系的建立是保障分布式核心运行的基石，其会围绕核心系统运行的要求来做整体设计，所以在核心系统规划的中后期，就应该启动运维平台的设计规划。



运维体系设计的核心是依托 PaaS 平台的原子化服务能力，根据系统运行的不同要求进行服务迭代来支撑整体运维的要求，其中 PaaS 服务的基础支撑服务包括：配置平台、作业平台、容器平台、测试平台、自动化调度平台等，基于这些平台的能力，通过服务的方式对外提供服务支持，从而构建出相应的运维 SaaS 服务如：监控管理、事件管理、报警管理、故障处理、数据决策等能力。分布式运维体系也可以把 PaaS 平台提供服务支持设计成分布式运维中台架构，能够更好的针对客户自身的运维体系来发挥分布式运维各个组件之间的协同。

分布式运维体系的建立也是围绕几大能力主题来设计的，包括：运维监控、运维配置管理、运维数据湖、运维智能分析、运维自动化、运维流程管理、多云管理以及运营统一管理。每个能力模块都是运维体系中的一个重要环节，根据自身用户的特点，还可以引入应用发布，或者在核心系统运维的角度可以命名单元发布，不同的客户在定义自己运维体系过程中，还会把应急处理的模块以及相应预案的策略作为分布式运维体系中的一个重要环节，当系统出现问题可以通过此模块来统一处理流量、恢复系统等操作，通常这些能力也可以是技术中台会覆盖的范围，所以功能具体落地在哪个系统需要结合自身情况来设计。

4.3 落地实践案例—某城商行核心系统升级

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，客户数据切片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

分布式核心投产效果显著，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。系统运行正常，各项指标表现稳定。支撑了新业务场景下需求变更的快速响应，在国产化战略上迈出结实一步。



腾讯云分布式核心—单元化架构路线

2015年微众银行核心系统上线，单元化架构也在银行领域初露锋芒；随着银行技术架构逐步迈入分布式领域，单元化架构也逐渐进入大家的视野。

本章将结合腾讯实践经验，剖析单元化架构路线价值、设计要点、落地案例，解析标准微服务和单元化架构之间不同应用场景和价值。



5.1 什么是单元化架构

单元化通过把一部分计算资源和一部分数据资源进行逻辑上的绑定，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。

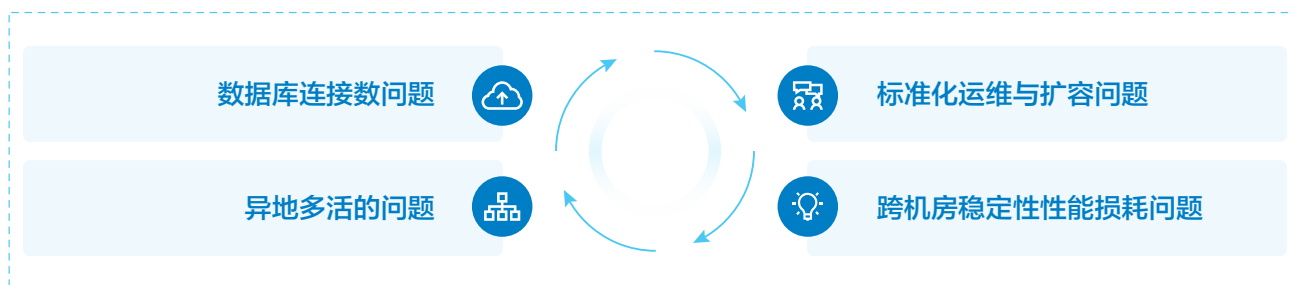
在腾讯“海量服务之道”提出的SET部署模型，就是腾讯在早期处理海量互联网业务的一种“单元化实践”。在2015年支持微众银行建设核心业务系统的过程中，创新性地提出按DCN（数据中心单元）的设计理念，按照业务和用户维度进行水平拆分，将银行核心数据库账户体系、业务交易处理打包在一个DCN单元中（含服务器/数据库），在前端通过业务全局路由实现业务转发，这也是比较早期的银行核心单元化架构实践案例。

❏ 典型问题：单元化架构与标准微服务架构有哪些区别？

答：标准微服务架构默认不对跨中心流量进行治理，其对双活两个机房之间的距离和延时有一定要求，假设两个异地机房，没有单元化的微服务架构中，应用在访问数据库时会存在异地间的数据访问，因为计算层的负载是无状态的，而数据的主本是固定在某一机房的，应用跨异地访问数据时，其稳定性和时延是无法保证核心系统正常运行的，也就是微服务架构在不做改进的情况下较难适应未来异地多活的场景。这是标准微服务架构和单元化架构之间较大的区别之一。单元化架构因其学习成本和实施成本较高，比较适合体量相对较大或有异地多活规划的银行，需要结合具体银行客户在账户数、预算、自主研发能力、业务连续性要求、异地多活规划等方面来综合考虑。单元化架构是银行核心分布式转型的一种重要方向，但不是唯一的技术方向。两者的差异及如何选择可以参考第三章。

5.2 单元化的价值

在解决大型分布式系统面临的如下挑战时，单元化架构能更好的体现其核心价值。



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。通过单元来控制数据库的连接数，并以单元的扩容来实现分布式系统的整体扩容。

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，规范针对不同类型单元的运维方案，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。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。

5.3 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起初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，在新核心内部统一用客户号作为单一客户识别要素。那么一个客户的识别要素有多少，除了卡号、账号、手机号、身份证等，可能还有折号，这些要素汇总起来的总量巨大。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，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KV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。

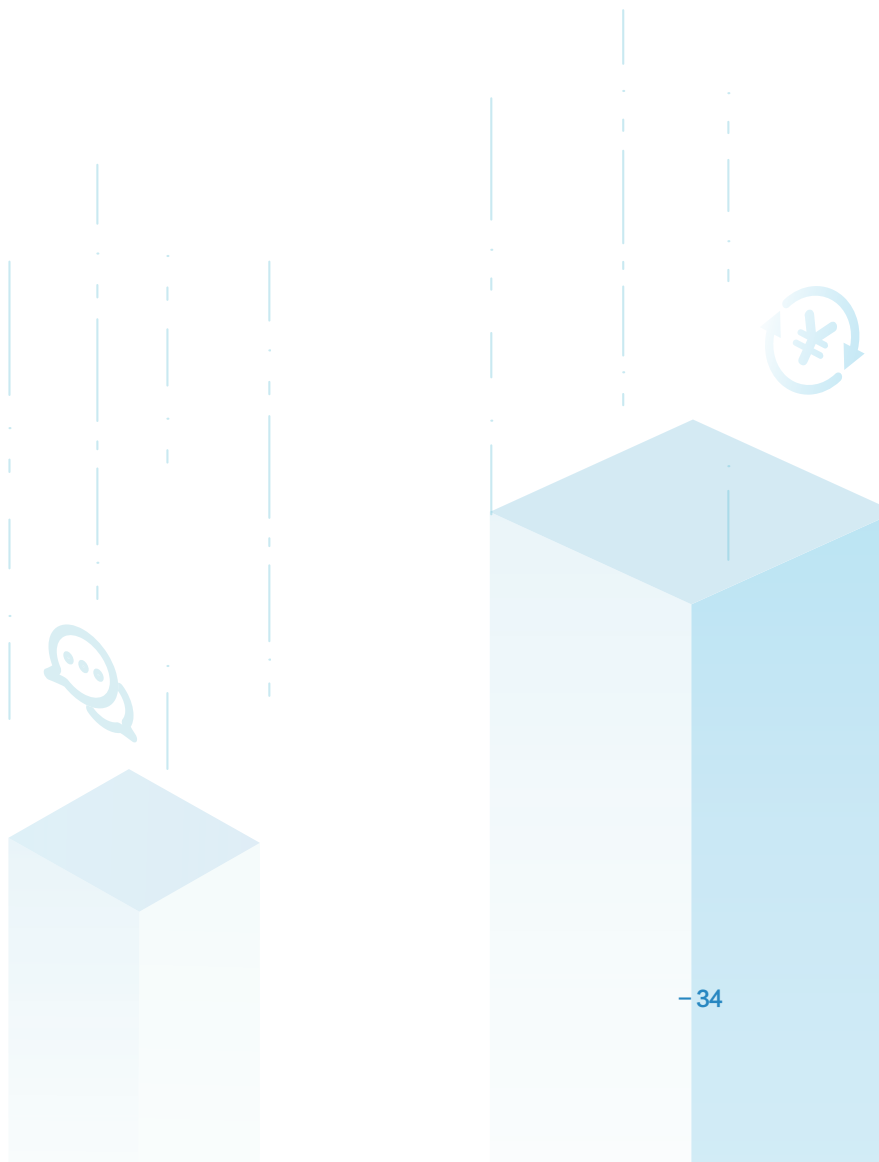
保障业务连续性

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，比如更新失败转异步、寻址失败转随机分发，但要求数据层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。

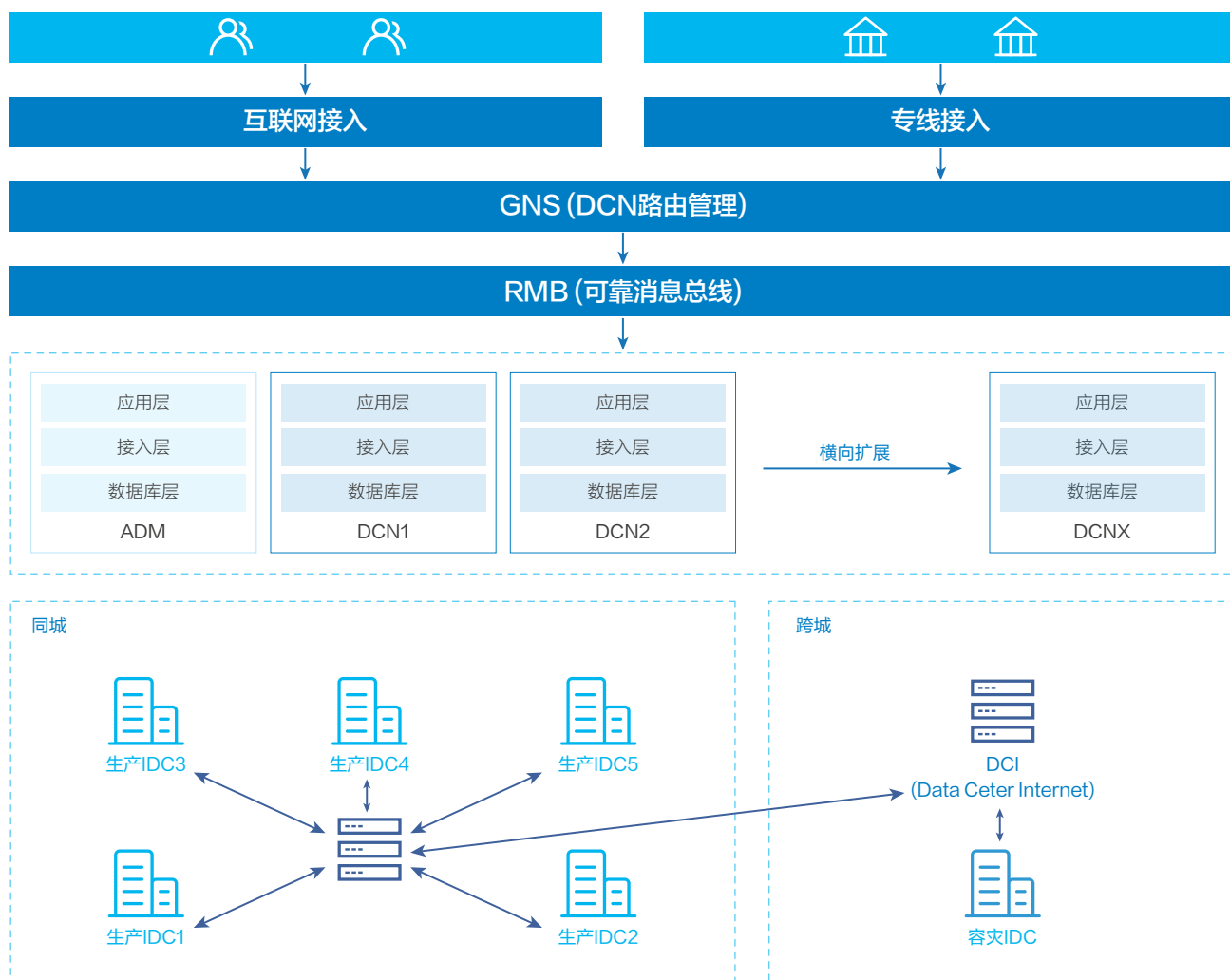
单元化的可观测性

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。



5.4 腾讯云分布式核心—单元化架构解决方案

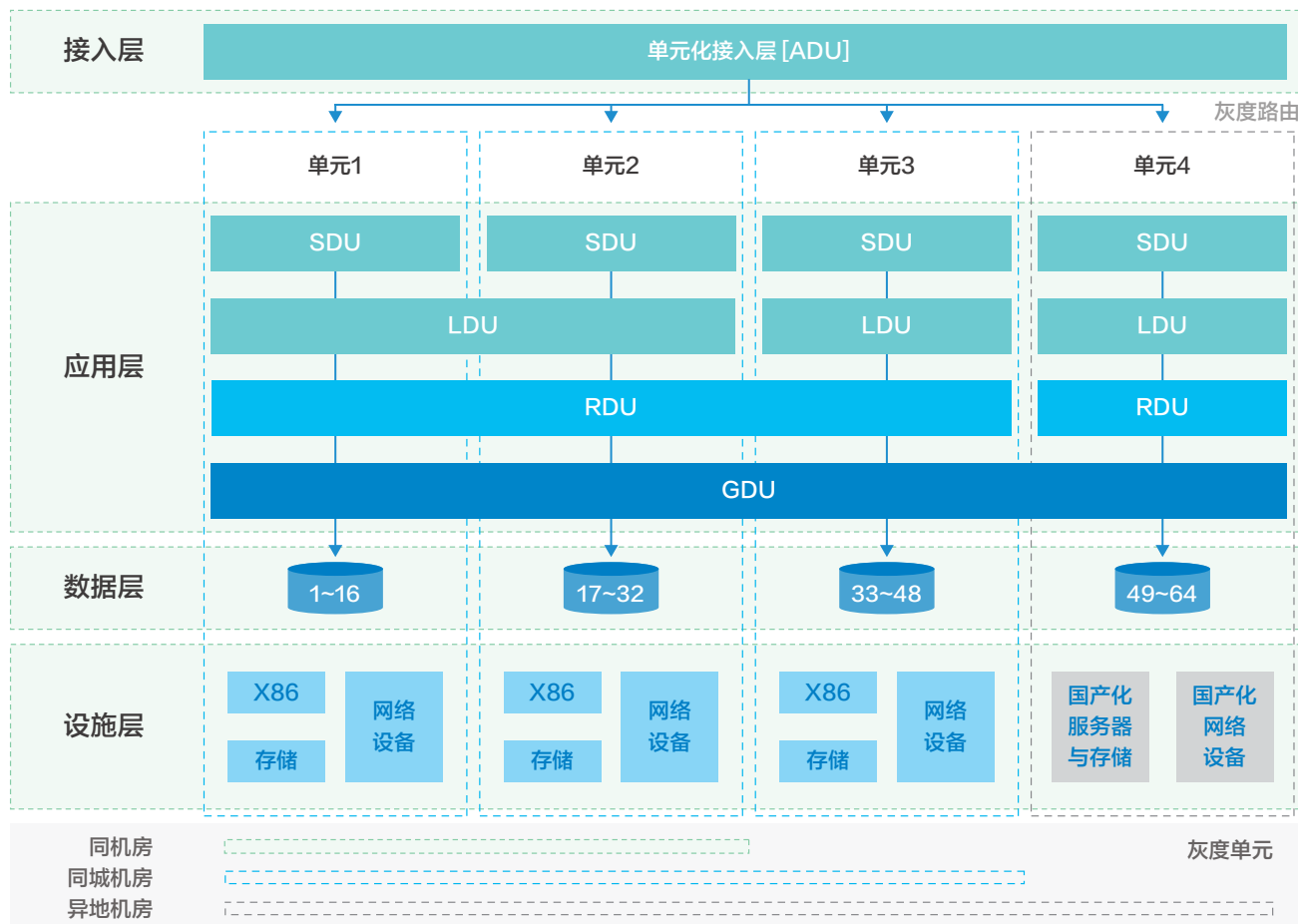
随着银行技术架构逐步迈入分布式领域，单元化架构也逐渐进入大家的视野。从2015年微众银行核心系统上线开始，单元化架构在银行领域初露锋芒。



其通过设计标准化的处理单元DCN，将多个处理单元组合形成一个整体的系统。每个处理单元都是标准的，可单独治理的闭环结构。可以理解为一个单元就是一个小分行，包含了全部的核心业务能力，只是处理的客户不同。这样便形成了大规模集群下标准化的、内部相互隔离的、安全可控的架构体系。

5.4.1 腾讯云单元化架构（TUA）

腾讯通过自身参与微众银行、国有大行、头部商业银行核心系统规划建设积累的宝贵经验，归纳提炼形成符合银行核心系统特点的腾讯云单元化架构TUA(Tencent 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。数据默认被分为64份，被均匀放置在若干个单元内。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。

典型问题：单元的拆分维度必须是客户维度吗？数据是否必须拆成64份？

答：单元的拆分维度和数据切分维度保持一致，按客户拆分是当下落地案例中的最佳实践，但同样可以考虑按机构或地域拆分，参考“4.2.1数据切分策略”。64只是方案默认的逻辑分片数量，可以按照业务体量调整。逻辑分片数量决定了未来最大能扩展的数据分片数量，64即代表未来最大支持扩展到64个物理分库。

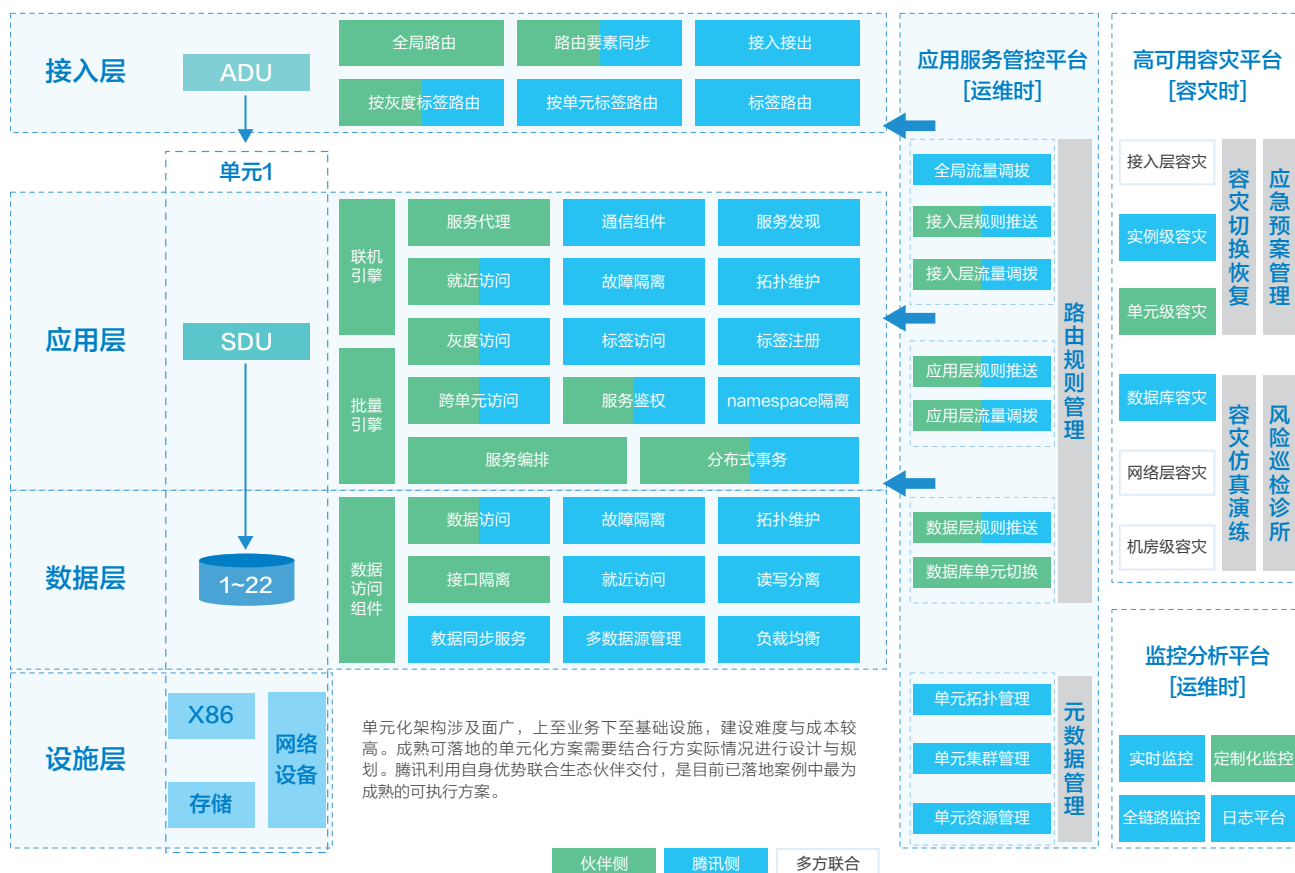
ADU	联机网关	外呼网关	文件网关	全局路由
SDU	存款中心	贷款中心	客户中心	合约中心
	联机引擎	调用代理	服务编排	分布式事务
	批量引擎	批量控制	全局序号	中间件代理
	微服务框架	数据访问	指标采集	参数代理
RDU/LDU	产品中心	机构中心	核算中心	参数中心
	注册配置中心	事务协调器	服务治理中心	PaaS服务
GDU	全局注册中心	全局配置中心	全局事务中心	全局参数中心

服务组件SDK组件

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)负责接入接出能力、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共享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。

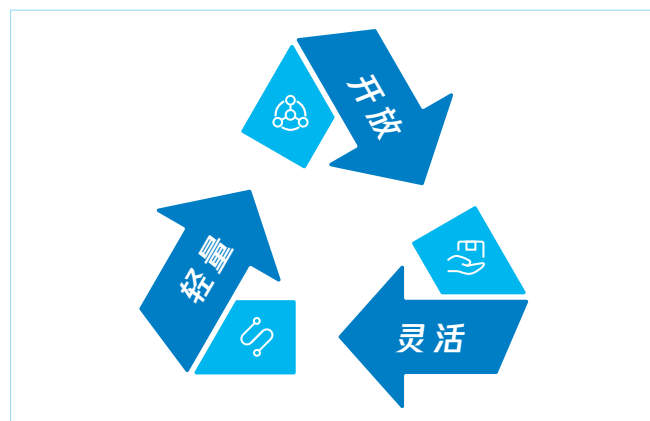


5.4.2 腾讯云单元化落地实践 >>



完整成体系的单元化架构一定是包含了各个层面的综合设计，其中就有ISV伙伴的应用架构、行内的技术标准容灾体系等。所以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。

腾讯单元化架构的设计与交付理念如下：



开放性

底层设计透明化，提升自主可控能力；沉淀行业通用功能，强化底层稳定性；功能聚合与组件化设计，增强解耦与组装能力。

轻量化

联合ISV提供端到端单元集成方案；提供集成标准，简化集成难度，提升质量；提供集成目录；量化投入产出比。

灵活交付

可结合行方现状与规划提供可演进的单元化架构方案；针对通用性需求，腾讯可提供拆箱扩展服务；联合ISV提供可集成的单元管控模块。

在腾讯参与交付的众多核心案例中，此方案已与多家主流ISV厂商联合适配落地。

5.4.3 腾讯云银行核心—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 64 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。

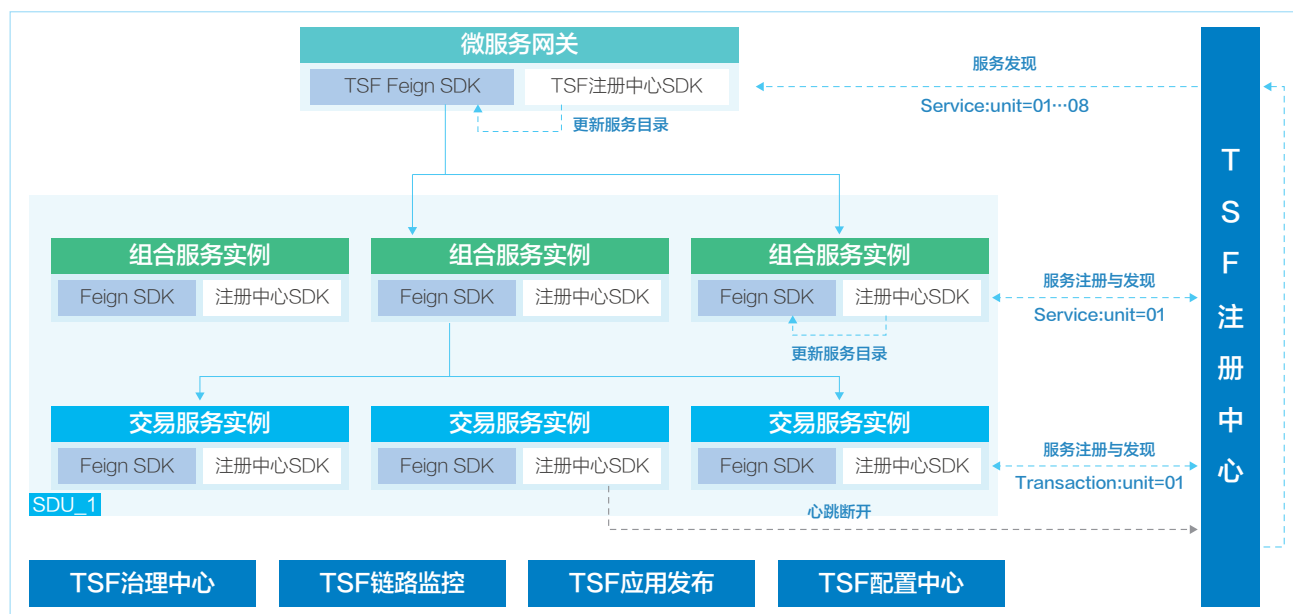
❏ 典型问题：为什么单元化架构下需要设计逻辑分片而微服务架构不需要？

答：这样设计的原因是单元化架构中要求计算和数据绑定，这就需要网关到应用层的路由不能再是随机负载，它需要和数据库网关到物理分片的路由算法保持一致才行，否则可能会出现网关路由到某个单元的应用，但该单元的应用却在访问其他单元的物理分片，这样就失去了单元化的意义。

技术架构标准

交易处理策略

单元化架构下交易处理策略以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。

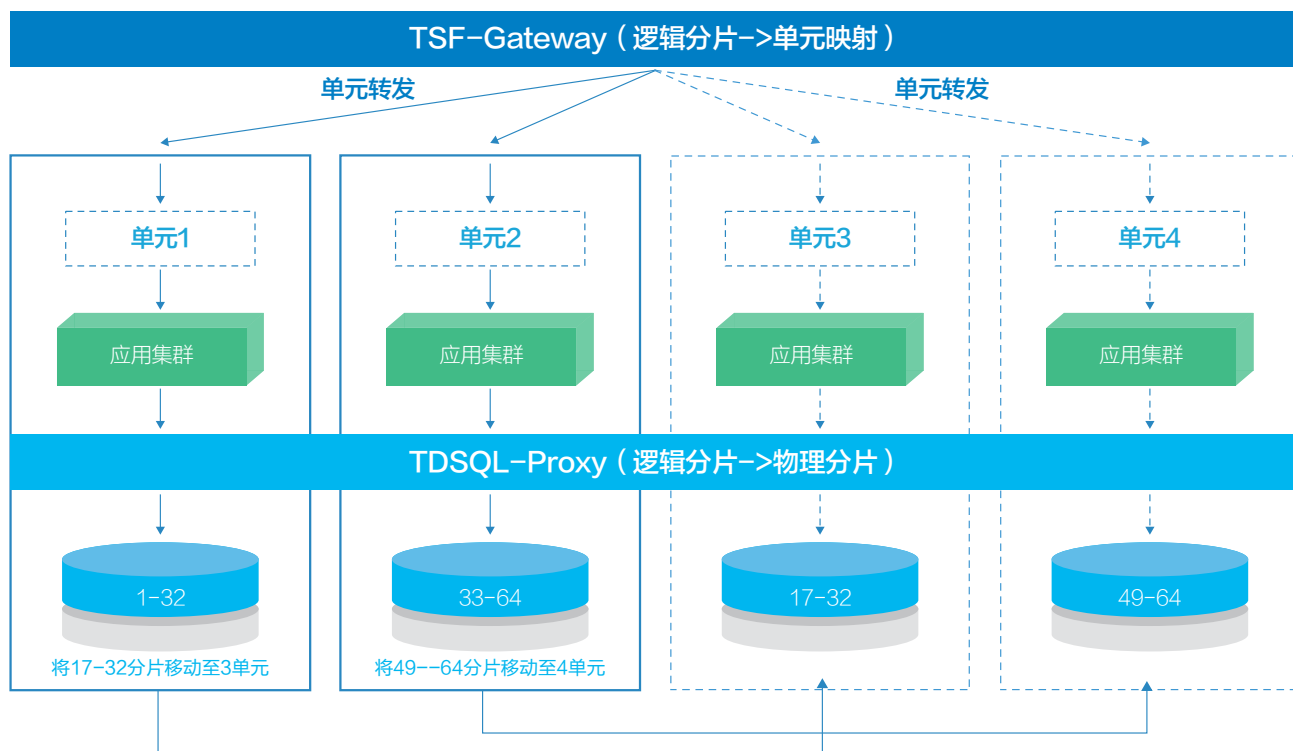
分布式事务策略

分布式事务处理策略与标准微服务架构保持一致，请参考4.2.2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。先扩容新的单元标记，再增加新的计算资源和物理分片，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，最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，这样扩容的时机是可以提前规划的，这种标准化正是银行系统治理工作所需要的。

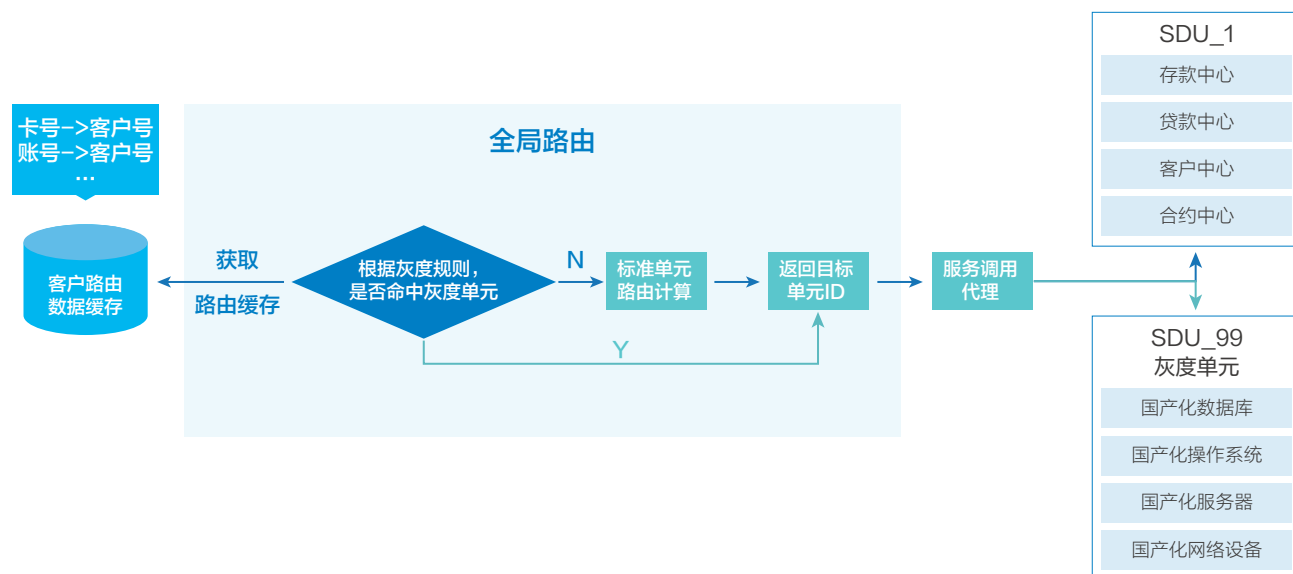
举例说明：假设以客户维度切分全量数据，每个逻辑分片中存放着1万客户数据。现在有两个单元，每个单元中各一个物理分片，每个物理分片中有32个逻辑分片，也就是一个单元中有32万客户数据。其中1号逻辑分片原本是在1单元的1号物理分片中，由于识别到其为热点分片，现要将其迁移至一个新的3号单元的3号物理分片。那么首先要将1号逻辑分片中的所有数据迁移至3号物理分片，再修改1号逻辑分片到3号物理分片的映射关系以及1号逻辑分片到3号单元的映射关系。当更新生效后，1号逻辑分片中的客户再发起交易时，网关就会将其路由到3号单元，其中的应用会从3号物理分片中获取数据进行处理。通过这种方式解决扩容问题。下图展示了从2个单元扩容到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灰度，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64个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充分做好设计。



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。

典型问题：灰度能力一定要依靠单元化实现吗？它和全局路由的关系是怎样的？

答：没有单元化架构也能实现灰度，只是实现方式不同。因为单元化架构与生俱来的隔离特性使其在灰度发布和运行上有着天然的优势。无论用哪种方式实现都会涉及对流量的控制和目标的寻址，其实也就是灰度机制中的路由能力，它可以算是全局路由的一个补充：全局路由处理标准单元的路由，而灰度机制处理的是灰度单元的路由。

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2019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

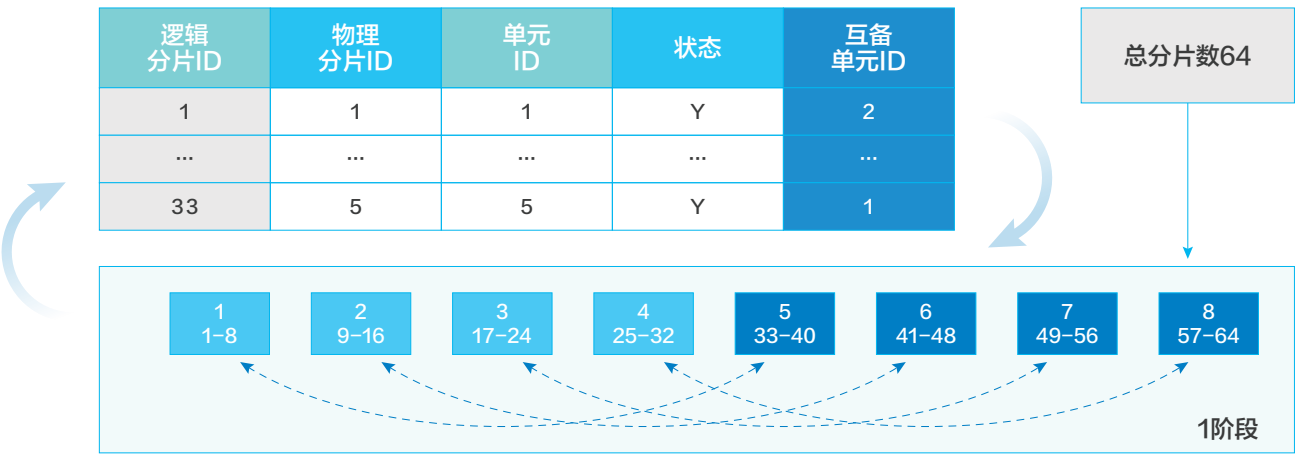
对于采用业务建模的各家大行，其对应的IT实施工艺则很好地适配了业务建模的产出，能通过识别->定义->分解->细化等工序把业务组件转化到微服务组件，其中包括细化后的构件，一个构件作为一个最小的开发任务，可用于工作量分配和质量跟踪。通过组装构件形成可对外提供最小业务能力的基础服务，通过可视化的服务编排能力，快速组装基础服务形成组合服务。实现稳态和敏态分离的双速模式，适应未来的敏态业务和快速发布。

分布式技术组件

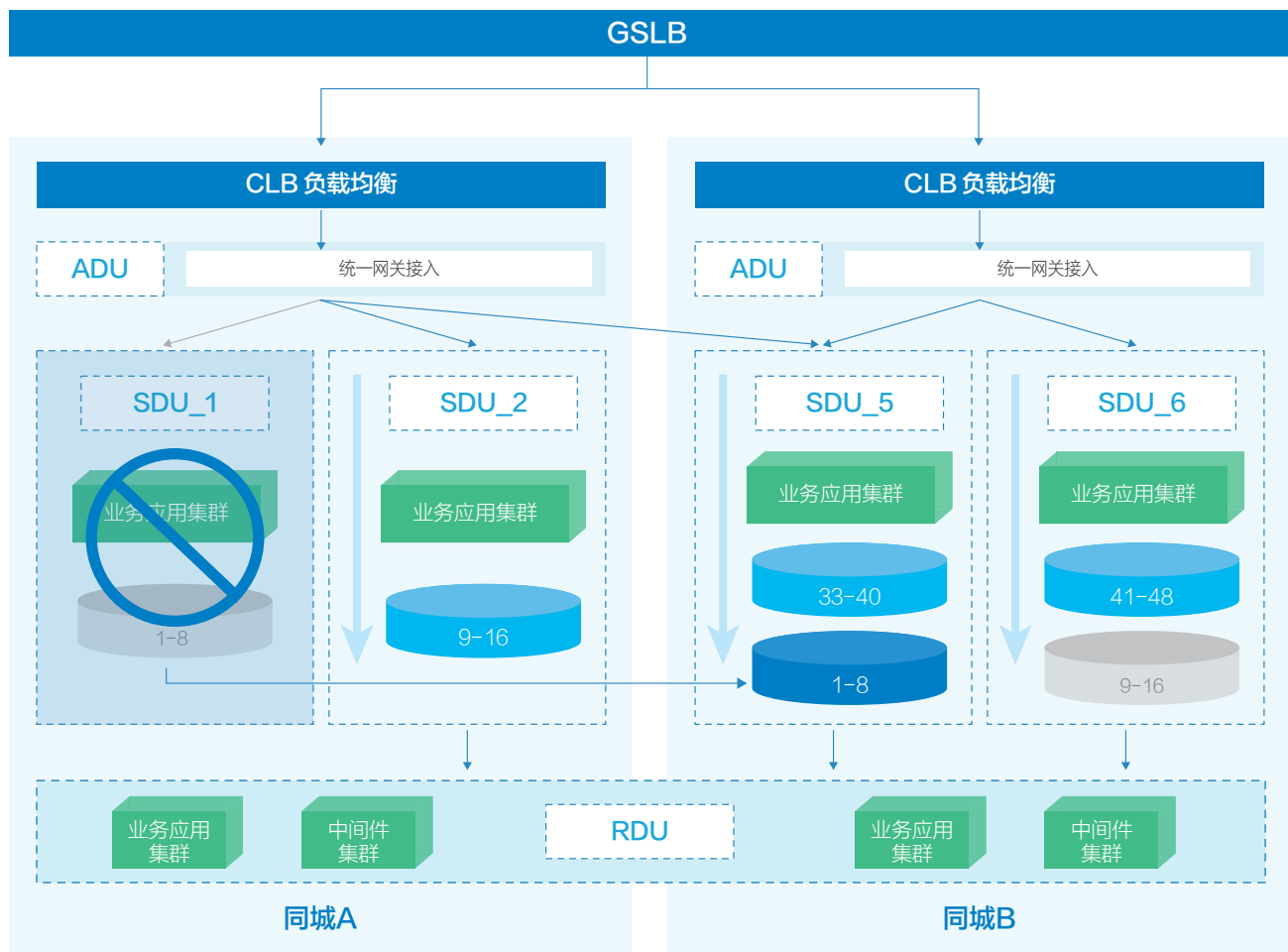
用于对应用层屏蔽分布式底层技术细节的组件合集，以核心应用厂商的软件包自带的组件为主。但这一层在业界并未有统一标准，各家厂商的实现方式与功能范围和边界有所不同。常见的组件包括了联机交易和批处理引擎、分布式任务调度组件、分布式流水号组件、分布式锁组件、数据访问组件、服务编排组件等。个别组件与PaaS平台存在重叠，如：微服务框架、分布式事务组件等。在规划阶段需要明确双方职责边界，在设计阶段确认组件间的集成机制。

业务连续性

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



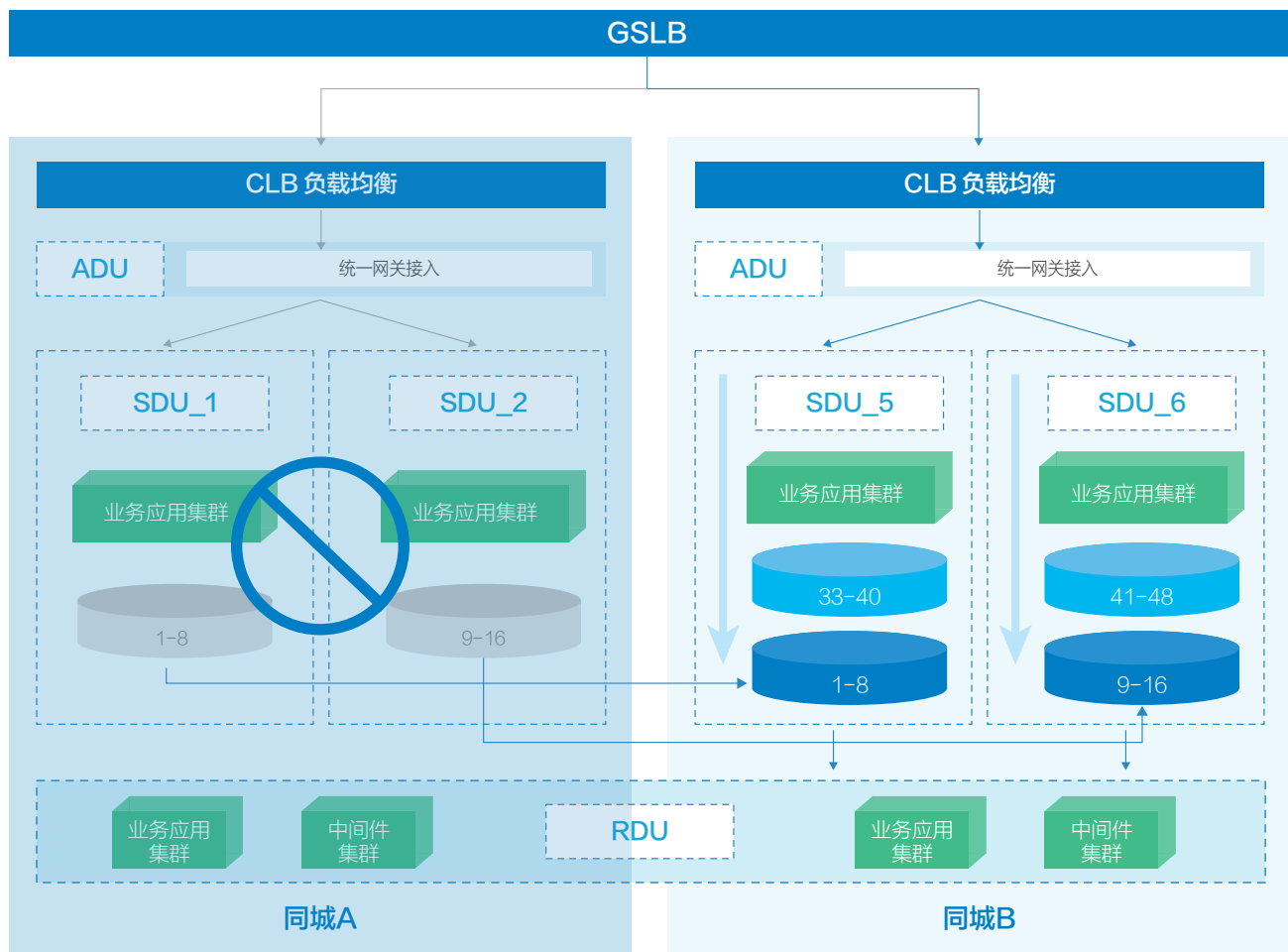
通过设计“互备单元表”来建立两个单元间的互备关系。当其中一方出现故障时，通过更新状态，来指向其互备单元ID从而进行路由调整。实现单元间互备要求单元的数据副本需要放置在对方单元中，当发生切换时，数据库会将对方单元下的副本调整为主本。



如图，请求通过全局负载服务进入到两个机房，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，停用单元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。

典型问题：单元的业务连续性有没有更简单的实现方式？

答：有，采用单元冷备设计，即每个单元都配备一个独立的备份单元做为冷备。这种模式的设计较为简单，但资源利用率很低，业界较少采用。其次是采用腾讯云提供的TSF微服务框架来实现单元化能力，其能够独立实现单元化，也能联动TDSQL数据库实现单元互备能力，进行一键单元容灾与切换。



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：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、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。

5.5 落地实践案例—某国有大行核心系统

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500个数据库实例。在业务投产之前，客户进行了严苛的业务旁路验证。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的快速定位。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。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）旁路验证，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500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，切换过程可以在分钟级完成。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，规定时间内500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。

应用监控（全链路跟踪系统）

分布式架构体系下业务应用进行了微服务拆分、单元化部署，微服务众多且存在复杂的调用关系，由此业务一旦出现故障，如何快速定位和隔离故障变成一个突出问题。全链路跟踪系统可以实现从业务的发起端如：ATM、柜员、手机APP等渠道，再经过各类业务系统的微服务处理，最终调用核心系统相关服务实现查询或者转账等操作逻辑，整个业务链路额可以通过分布式运维模块精准的记录、分析和展现出来，并通过交易在各微服务上的响应时间和成功率来直观的反应出具体的业务问题模块以及相关组件。通过引入新一代消息中间件，升级技术中台、提升PaaS平台多项关键能力，有效提升分布式金融级交易一致性、高并发业务稳定性、应用服务安全性、跨站点容灾高可用性。

“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1200个，在行方多个数据中心投产的TDSQL数据库物理节点超过600个。

分布式PaaS平台和单元化技术平台的建设代表着全行科技的转型，技术平台将作为全行分布式技术底座面向全行推广，全面提升数字化转型进程。



典型银行分布式核心系统应用架构与实践

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。



6.1 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Memcache、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。

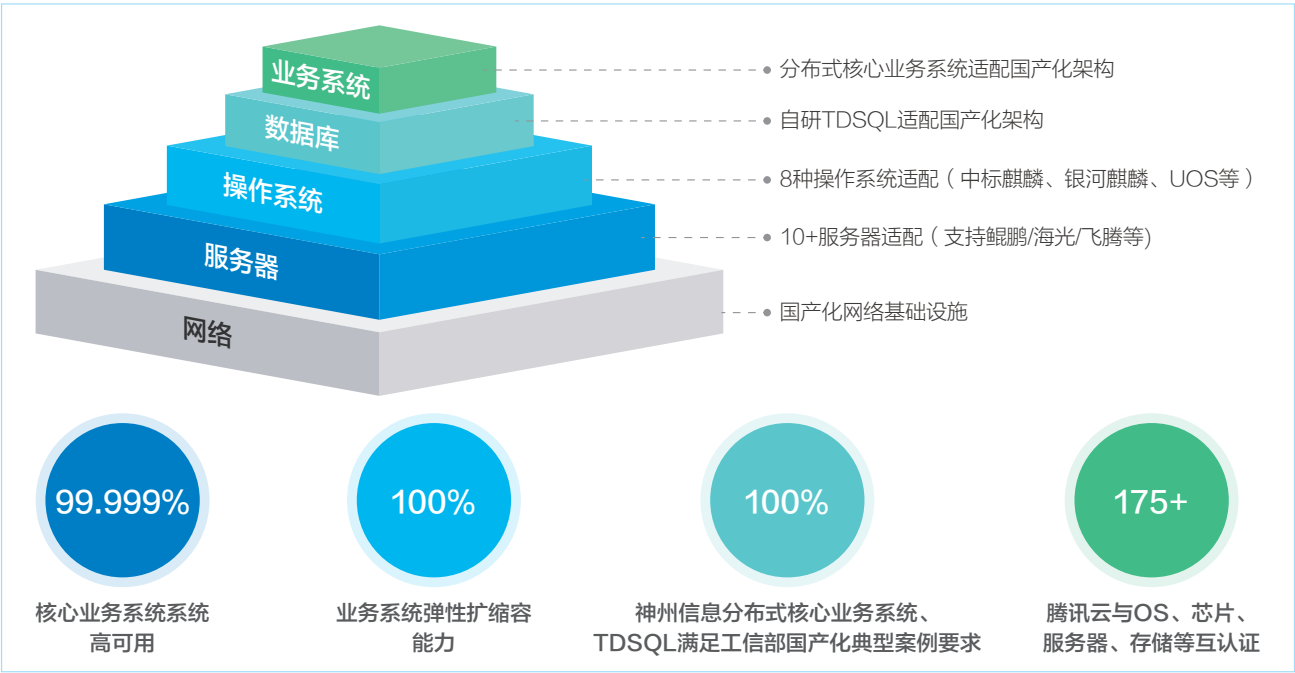


在联合方案基础上，神州信息与腾讯云持续针对双方核心产品最新版本进行深度适合作。如下是神州信息 Sm@rtEnsemble16 的应用架构。

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		
	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。

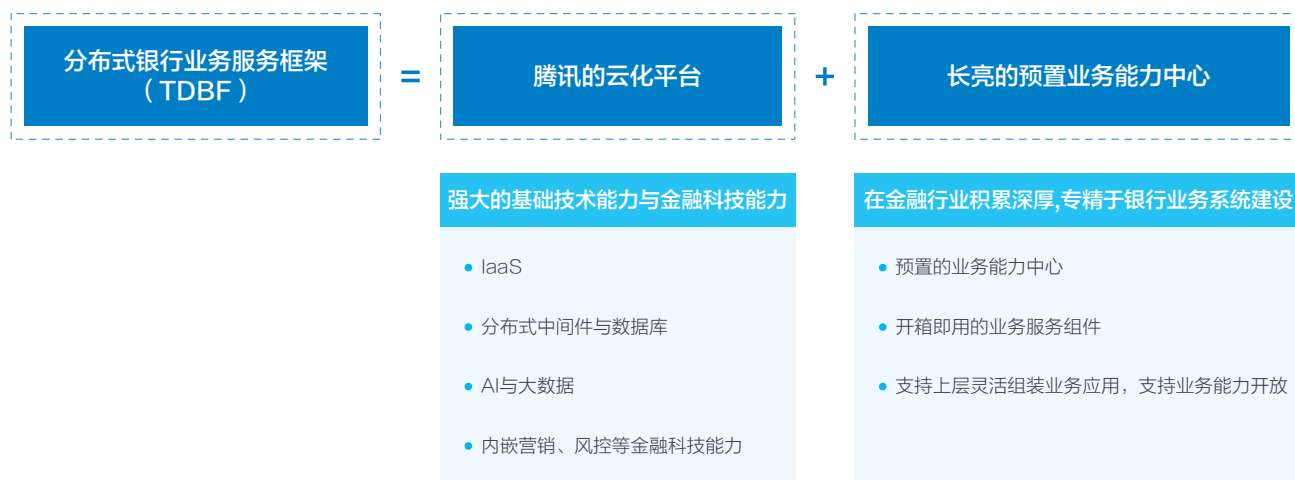
针对双方产品最新版本，腾讯云与神州信息联合开展了基于云原生、微服务架构、分布式数据库 TDSQL 的全栈国产化新一代核心系统的投产级测试，对核心业务系统的各项功能和技术性能进行了深度验证，经过适配和调优，双方产品可靠性、性能指标满足金融机构自主可控要求。



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。

6.2 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。



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 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扩展性，能够获得良好的处理性能值，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。



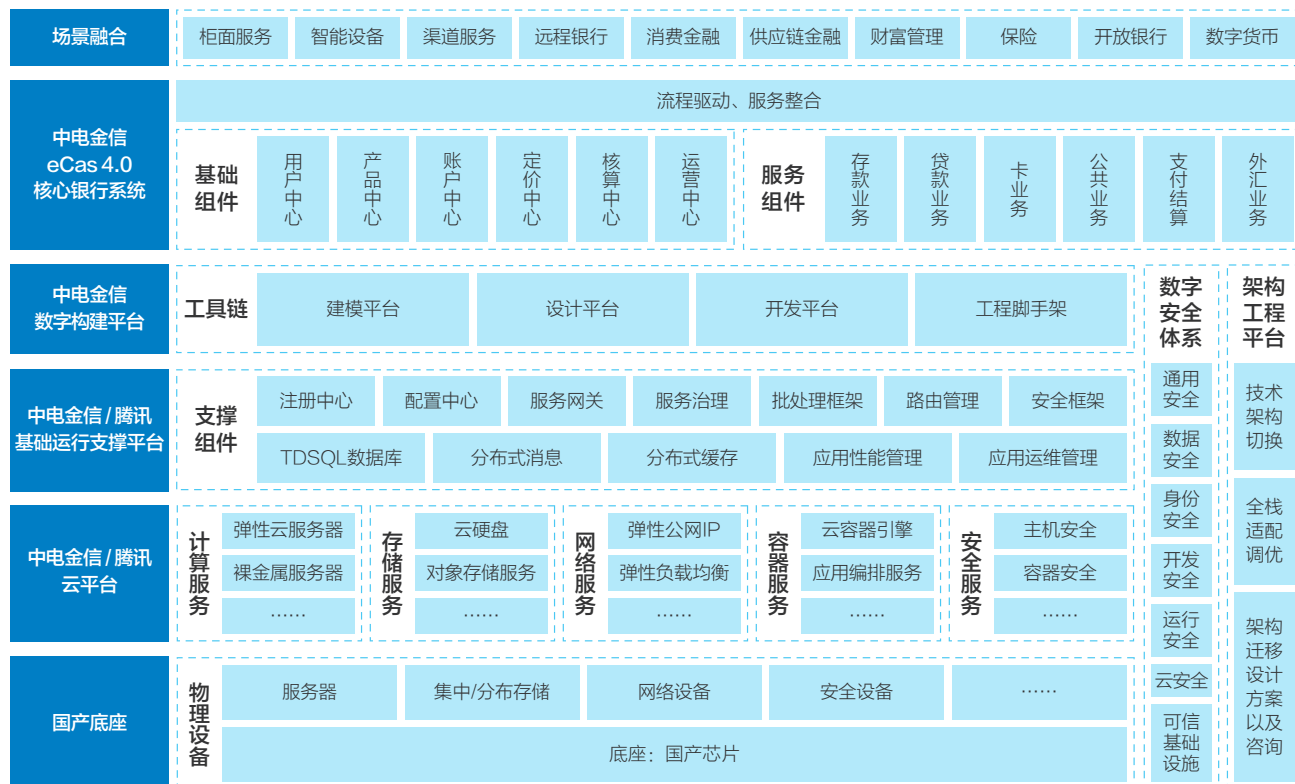
腾讯云与长亮科技双方持续打磨联合方案，加速赋能银行数字化转型，合作范围现已囊括了 TDSQL、TCE、TDMQ、TSF、Coding、TCS 在内的腾讯云全体系的 PaaS 产品。同时联合方案也相继在平安银行、福建海峡银行、昆山农商行、云南红塔银行为代表的多个银行核心系统相继成功投产。

未来随着双方合作的深入，将为市场带来更高效、更安全、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。



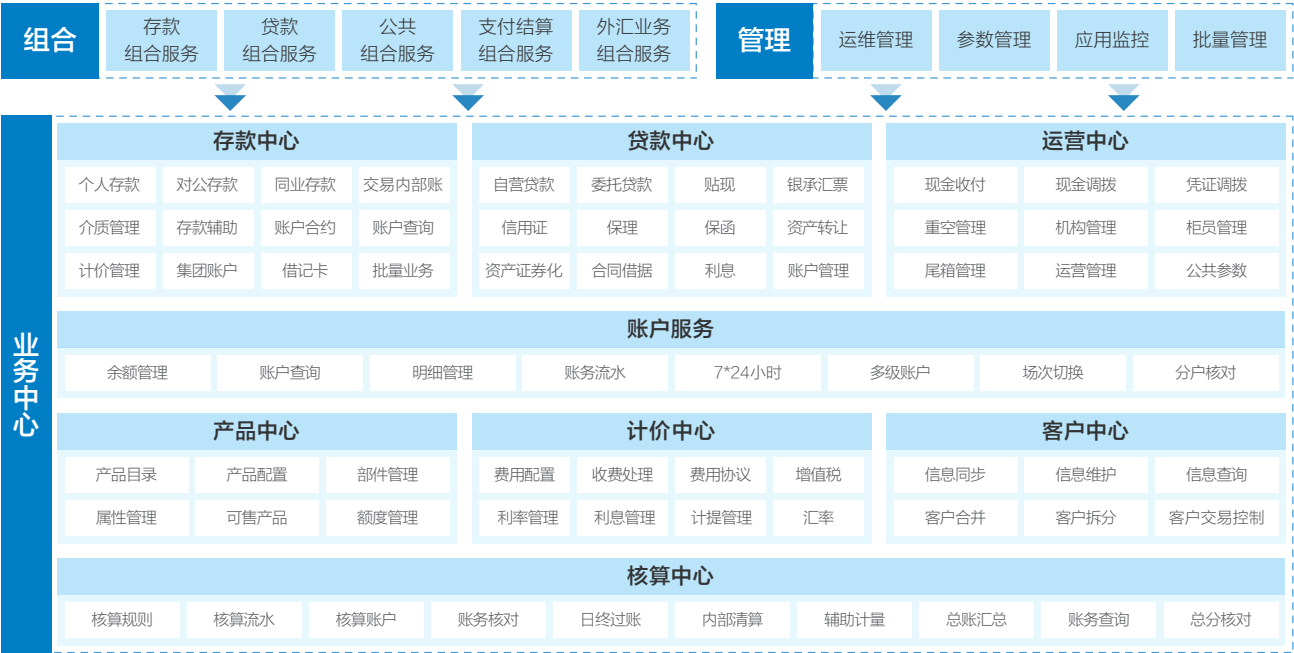
6.3 腾讯云与中电金信深化合作

腾讯云与中电金信于2022年展开全面合作，双方基于中电金信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。腾讯云主要聚焦支撑组件层和计算服务层，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。以下是中电金信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。



6.4 腾讯云与生态伙伴合作案例实践

6.4.1 某华北城市商业银行 >>

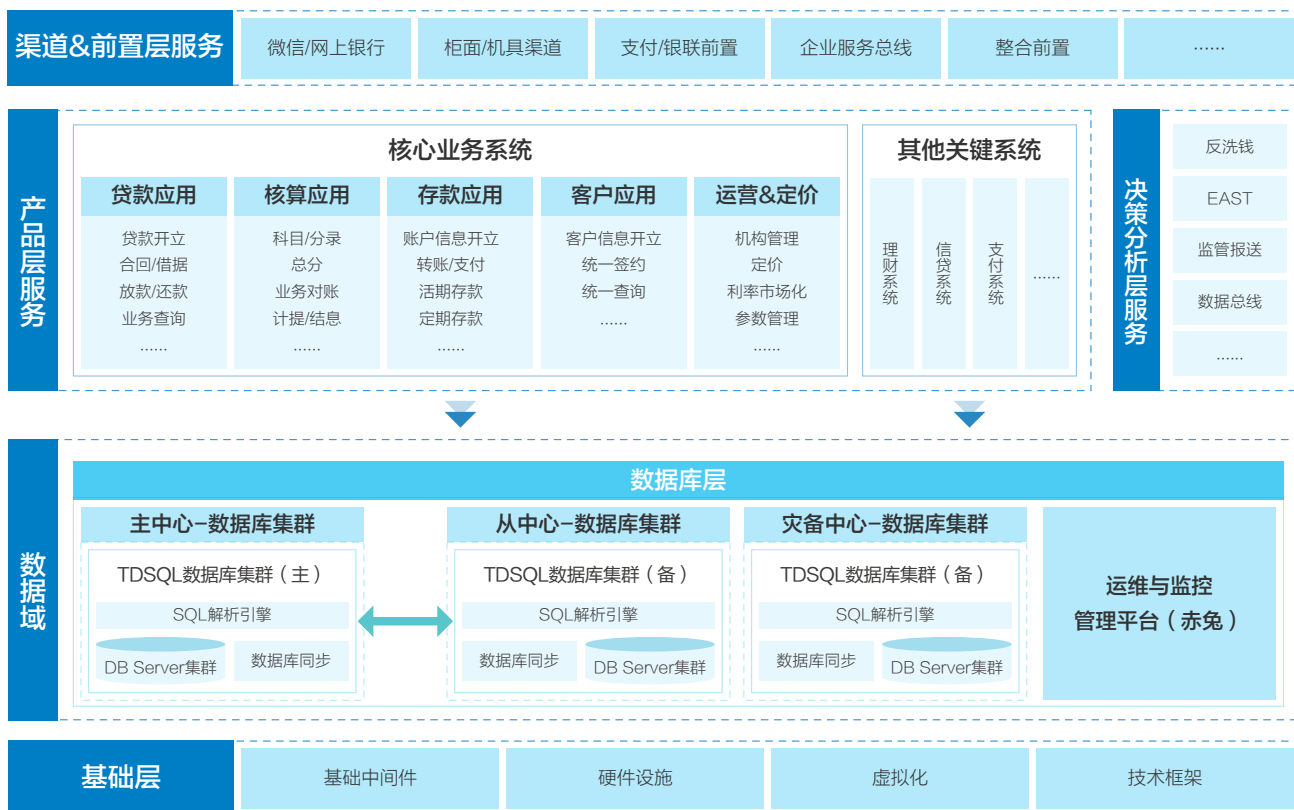
案例背景

结合“十四五”软件和信息技术服务业发展规划以及金融发展战略，某城商行以“创新驱动”、“重点突破协同推进”、“安全可控”等基本原则做为未来产业发展的指导原则；此外，随国际金融局势和疫情变化，金融银行业发展即要满足科技创新要求，同时又要在保持稳定基础上走出自主、安全可控的路线。在这两个背景下，某城商行以国产分布式数据库系统为切入，于2021年启动了新核心建设。主要建设目标如下：

- 技术架构升级-分布式、高可用、灵活扩展、云化、虚拟化；
- 关键技术自主可控；可视化开发、运维、监控；
- 国产软硬件适配；
- 分布式环境场景创新与融合；
- 金融业务服务能力提升；
- 持续迭代建设积累实践经验；

建设方案

整体项目既要完成自主安全可控以及技术经验、人才储备积累的目标，同时要保证银行业、监管对于信息安全、金融服务能力、金融稳定的要求以及国家信息化建设要求，因此采用“小步快跑”迭代思路逐级开展国产化分布式数据库的建设，采用标准微服务架构路线。整体建设方案如下图所示：



整体项目分为三个阶段进行实施，主要依托神州信息 Sm@rtEnsemble16 核心系统和腾讯云 TDSQL 分布式数据库。通过核心系统的分阶段升级，逐步实现银行 IT 系统架构的不断优化，满足行业金融科技不断发展的趋势，实现银行核心系统微服务、分布式的技术特点满足互联网高并发场景业务创新的需要，实现银行统一产品工厂、统一定价工厂、交易与核算分离等业务特点满足银行快速进行金融产品创新的需求。

项目采用腾讯云 TDSQL 国产分布式数据库作为技术底座，将核心系统数据使用 hash 方式平均分布在集群物理服务器节点上，依托横向线性扩展和多副本读写技术，实现了高并发处理和高可用容灾保障。高可用容灾设计为两地三中心架构，同城主从双中心部署一套集群，通过集群内部的同步机制保证跨中心级别高可用能力；异地灾备中心计划部署一套容灾集群；两地集群间通过异步方式进行复制。通过行内已有 HDFS 服务进行备份数据存储，保证数据安全性。

建设成果

该城商行新一代核心系统于2022年7月正式投产上线，是首批次在传统核心业务领域使用国产数据库的城商行，也是河北省首家在传统核心业务场景下使用国产数据库的城商行。为金融行业信息技术创新探索与实践肩负了应有的社会责任与担当。

本次核心系统国产化升级同时伴随着外围63个系统模块的更新换代，是该行有史以来规模最大、范围最广、难度最高的一次系统升级。为该行建设全面提升了数字化金融应用能力和服务水平，在运营基础、服务能力、产品工具、风险防控等多领域提供了更为强劲的支撑。同时通过核心系统国产化建设，打破国外技术壁垒，逐步实现自主可控，实现可持续的健康发展。

该行核心系统首阶段国产化数据库的成功也为核心系统后续几个阶段顺利实施积累了大量的经验：

- 在传统数据库海量数据向腾讯云TDSQL的无损迁移方面。借助腾讯云数据迁移工具DBbridge，新核心系统可以轻松实现异构数据库之间数据的迁移和同步，将传统数据库数据平滑、安全地迁移到TDSQL上，为银行业务后续的扩展提供基础。
- 使用腾讯云TDSQL结合不同业务场景，从存量数据量、未来数据增长量、性能指标、硬件资源使用率等维度，将数据库表按照不同表单进行数据分片设计，可以有效提升了新核心系统的库表管理能力，为新一代核心系统实现了客户体验的全面优化。是该行实现专业化、特色化、差异化的发展路径的坚实后盾。

案例价值

该项目是神州信息与腾讯云数据库联合方案的首个国产化传统核心落地案例，对后续城商行传统核心系统国产化、微服务化、分布式改造具有非常积极的借鉴意义。

6.4.2 某东南城市商业银行 >>

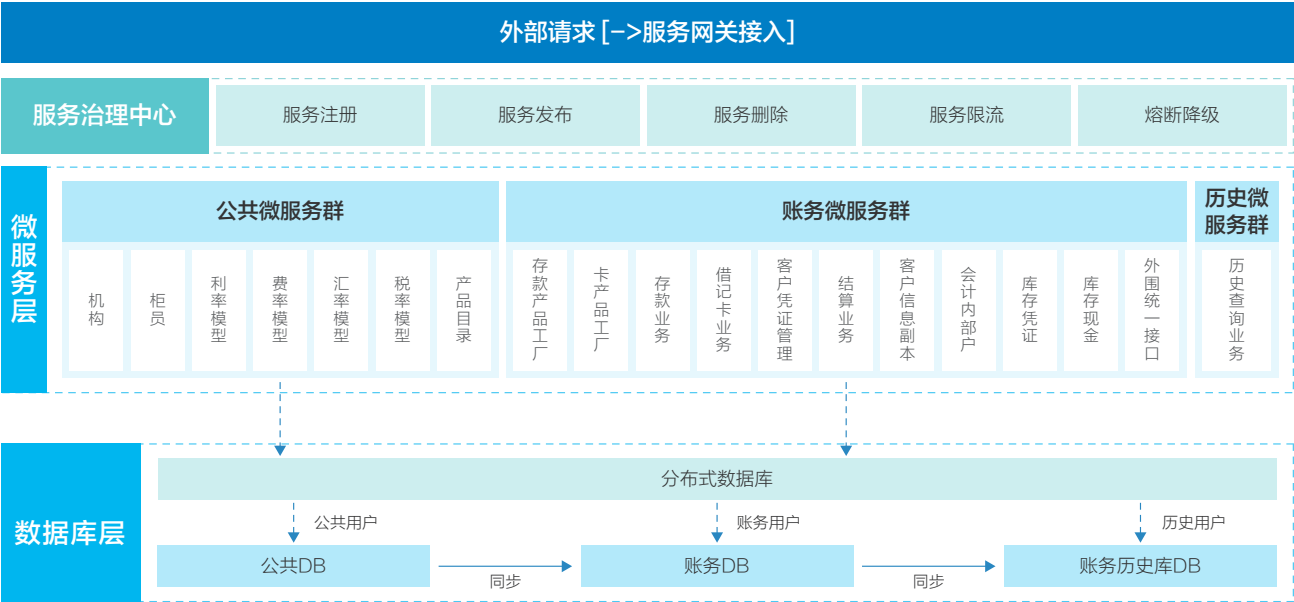
案例背景

“十四五”规划指出，要打造数字经济新优势，加速金融行业的数字化转型已成为大势所趋。对于银行业自身，移动互联网、大数据、分布式等大量新技术的引入，使传统的银行核心系统架构已经无法适应新时代的要求。某城商行为实现跻身全国城商行第一梯队的业务发展目标，必须对作为“内部发动机”的核心业务系统进行升级优化，构建在新技术条件下适应未来发展的信息系统架构体系，实现业务运营能力与技术支撑能力的双重提升，提高该行在省内银行业的综合竞争力，为后续全面开展数字化转型奠定了坚实基础。

新一代核心业务系统建设时间紧、任务重，而科技队伍和科技投入规模相对有限，所以该行在技术方案的选择上，尤其关注成熟性、适用性、灵活性和经济性。腾讯云与长亮科技联合设计的核心业务系统标准微服务架构，基于长亮V8核心能力，无缝衔接国产分布式数据库TDSQL，并融入微服务、读写分离、多源同步等技术，在保证金融级数据全局一致性的基础上，降低了系统的复杂性，能有效解决传统集中式核心的并发瓶颈，提升核心系统的高可用性和动态扩容能力，完全符合某城商行业务和技术发展需要。

项目技术方案

长亮科技与腾讯云共同支撑了该行新一代核心系统建设项目，采用标准化架构,项目整体架构图如下所示：



建设方案整体情况

- 长亮科技依托其“微服务+分布式”架构的新一代分布式银行核心系统，全面梳理核心系统业务流程，建立微服务治理中心、公共微服务群、账务微服务群和历史微服务群，强化业务操作风险控制，提升客户体验，亮点有：采用微服务架构，具备横向扩展、负载均衡、服务治理的能力，满足系统性能灵活扩展需要；构建全行统一的产品工厂、定价工厂模型，打造通过模型参数定制开发新产品的能力，高度实现业务规则化、产品参数化，支持业务产品快速创新。
- 腾讯云提供国产数据库TDSQL的成熟方案，为客户构建自主可控的国产金融级分布式数据库服务平台，采用两地三中心部署架构，同城TDSQL集群采用跨中心高可用容灾双活部署架构，异地灾备中心部署1套TDSQL集群，主生产集群和异地灾备集群之间采用通过TDSQL DCN异步复制的方式，亮点有：实现中心级别灾难快速自动恢复，且数据零丢失；仅有核心公共库、核心联机库、核心历史库3个数据库，运维管理成本低；提供数据库实时同步工具，满足核心业务系统与周边系统（如数据仓库、大数据平台等）的数据实时同步和数据交换的需求。

项目成果

- 增强业务连续性保障能力。通过业务系统同城双活部署，TDSQL数据库“两地三中心”部署，实现数据库同城RPO=0和RTO≈30s。
- 系统性能明显提升。新核心性能远超原核心系统，可支持每日亿级交易量，日常跑批由1小时降到约10分钟，季度结息也由3、4个小时降至30分钟，24分钟可完成300万笔社保代发，真正帮助该银行实现降本增效、风险可控。
- 有效降低IT成本。随着腾讯云分布式数据库 TDSQL 的引入，降低了该银行在数据库相关软硬件成本的投入，在同等的计算和存储资源需求下，预计每年可节约800万元，并将持续发挥降本增效的作用。
- 助力数据价值释放。通过TDSQL多源同步技术，轻松实现大数据平台数据准实时更新，构建实时营销、实时风控能力，有力支撑业务发展。
- 提升数据库管理效率。利用TDSQL数据库智能管家，统一管理TDSQL数据库，将大量数据库运维工作自动化，辅助DBA完成数据库性能优化、问题定位、根因分析等数据库运维管理工作，有效保障了数据库服务的安全、稳定及高效运行。

案例价值

历时14个月，依托长亮科技“微服务+分布式”架构的新一代分布式银行核心系统和腾讯云企业级分布式数据库TDSQL，该城商行快速完成了新一代核心业务系统的投产上线。在本项目中，腾讯云和长亮科技联合推出的标准微服务架构方案成熟性、适用性、灵活性和经济性表现突出，对中小型城商行、农商行有较大参考意义。

6.4.3 某西南城市商业银行 >>

案例背景

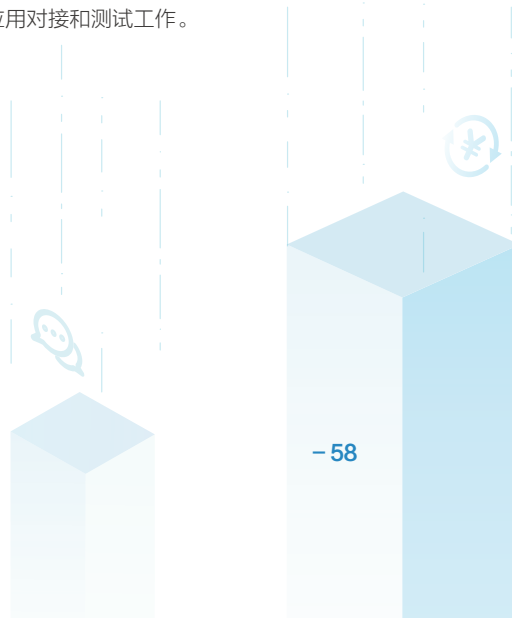
“企业上云”已然成为银行业寻求数字化转型升级的基本路径，我国有近九成金融机构已经或正计划应用云计算技术。围绕“产业银行+科技银行”双轮驱动战略发展思路，某城商行于2021年初启动新一代核心系统项目群建设项目，新建系统有核心业务系统、总帐系统、柜面系统、远程视频银行系统。项目整体目标如下：

品 架构	业务	安全	数据
采用分布式核心架构，新建金融云、支撑系统云服务化、弹性、容器化。	满足业务连续(RPO/R-TO)监管要求，支持业务双活体系。	满足商业银行数据中心监管指引，满足银行业信息系统灾难恢复管理规范。	支持数据本地高可靠，支持数据同城多副本，支持数据异地容灾。

项目技术方案

本项目采用标准微服务架构路线，项目的应用层选用了长亮科技核心系统，云平台选用了腾讯云TCE云平台底座（含微服务框架TSF、消息中间件TDMQ等众多PaaS服务）。具体而言：

- 全面梳理目前核心系统业务流程，进一步整合交易功能，形成全局交易流水，大幅度优化系统功能布局，提升交易功能的集中度，最大程度降低操作人员学习的成本，强化业务操作风险控制，提高业务办理效率，提升客户体验。
- 构建全行统一的产品工厂、定价工厂模型，打造通过模型参数定制开发新产品的能力，支持产品快速创新、业务创新。
- 借鉴集中式架构的优点，充分结合分布式架构的优势，采用了一种更为精简的分布式架构，即“云化部署+全功能对等应用+原生分布式数据库”。核心系统应用层面只保留了联机和批量两类微服务，而且服务之间的调用不涉及分布式事务。
- 联合腾讯云TSF构建了全行统一的微服务开发框架，并将腾讯云TCE云平台中的PaaS服务作为技术开发标准化组件，为全行技术架构的标准化奠定了基础。在统一、开放的微服务框架下，新核心系统业务只需要少量适配就可接入，并能实现模块化开发和灵活发布。
- 长亮核心对接腾讯TSF微服务平台，整体改造成本非常小，核心全量近1000多支交易，1周完成应用对接和测试工作。



项目整体逻辑架构图如下：



项目成果

新核心建设历时14个月，通过本次新核心建设，该行具备了服务亿级客户、每天亿级交易量的“双亿”能力，支撑平均每分钟处理约50万笔联机交易、交易平均响应时间不高于300毫秒。此外还实现如下建设成果：

- 本项目新建应用系统核心系统完成paas层上云，支持独立部署、腾讯云平台的弹性扩容功能良好，支持数据标准、参数体系和数据模型贯标。
- 业务侧协助行方实现产品组件化、参数化，大幅度提升产品化配置能力，更好地支持银行业务创新；灵活实现多维度、多层次的定价支持；新建更强健的账户结构，灵活支持虚实结合、本外币一体化的账户体系。
- 在生产环境以最优成本建立小型化国产化专区，快速启动支持业务全栈国产化试点。
- 联合腾讯云TCE打造统一云管理平台，支撑业务实现全功能对等部署，确保业务请求可以发送到任一数据中心部署的核心应用中，实现全集业务双活状态运行

案例价值

该项目基于腾讯专有云平台TCE、微服务框架TSF和长亮分布式核心系统，3个月内完成了核心及其他系统生产环境上云，是分布式银行核心上云的典型案例。

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的几年，与生态伙伴共同服务了上千家的银行客户，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、微服务平台、PaaS中间件等一系列成熟产品的成功实施经验，我们沉淀了丰富的银行核心系统设计的方法论和建设经验。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，帮客户实现自主可控国产化

涉及产品包括：TCE金融专有云、TCS金融原生云、TSF微服务平台、TDSQL金融分布式数据库、TDMQ分布式消息队列、CRedis分布式缓存、蓝鲸自动化运维平台和Coding研发效能平台等。全面覆盖了银行分布式核心系统所需的技术底座，并已在国有大行、股份制、城商行、农商行等数十家银行核心系统建设中成功交付，成熟稳定。

先进可靠的产品力，给客户核心建设提供强有力的支撑

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；Tencent 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获得金融科技发展奖。今年3月30日，权威机构国际事务处理性能委员会（TPC）官网披露，腾讯云数据库TDSQL以每分钟8.14亿笔交易的性能，打破世界记录，实现tpmC和性价比双榜世界第一。

丰富的生态合作体系，给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。

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。

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，期望与合作伙伴及银行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！





获取更多信息

如需了解更多行业解决方案的更多信息

请扫描二维码关注官方微信获取, 随时随地掌握我们的最新服务



联系人

马明月 | 邮箱: mingyuema@tencent.com

张帆 | 邮箱: joffyzhang@tencent.com