

2024 金融科技趋势 研究报告

出品：
恒生电子股份有限公司

联合发布：
中信证券 中金公司 国泰君安证券 兴业证券 安信证券 中信建投证券 西南证券

序

Preface
序言一

程永

中金公司 首席信息官

很荣幸接到恒生的邀请，为《2024 金融科技趋势研究报告》作序。近年来，金融科技的发展日新月异，以大模型为代表的人工智能技术以及大数据、云计算、低代码等新技术日渐兴起，同时随着信息技术应用创新进程的推进，对数据库等技术的自主创新以及信息技术应用创新工程的推广提出了新的要求。种种变化正在为金融机构的数字化转型带来新的机遇与挑战，也将为金融 IT 带来新的发展范式。

在这种背景下，构建数字化转型基座，加强金融科技能力建设，以领先科技能力赋能数字化转型非常重要，既要加快前沿技术应用落地，拓展应用场景，促进新技术规模化应用，也要推动信息技术应用创新工程推广成熟，加快规模化部署。与行业的金融科技能力建设相契合，《2024金融科技趋势研究报告》用三个篇章深入分析了大模型等新技术、信息技术应用创新以及架构等方面的趋势，并且每个趋势都给出了典型场景阐述及IT规划建议。我相信，这份报告将为金融机构的数字化转型提供宝贵的参考价值。

数字化转型是提升金融机构自身竞争力、实现高质量发展的必由之路。未来，期待在行业的共同努力下，金融科技将会带来更多的创新和变革，推动行业的健康发展，进而为实体经济的发展注入新的活力。

Preface

序言二

序



恒生电子 首席科学家

2023年，生成式人工智能引领数智应用变革，信息技术应用创新驱动业务系统焕新。我们看到重大的技术变量正在推动金融机构与金融科技企业寻找破局增长的“第二曲线”。而在信息技术爆发式革新的今天，趋势的研判与洞察更加具有挑战性。恒生的金融科技趋势报告已经来到了第三个年头，去年报告提及了安全可控、数据智能、性能体验、效率协作，同时也高度重视行业的标准化工作。阅读今年的报告，以下两点值得关注：

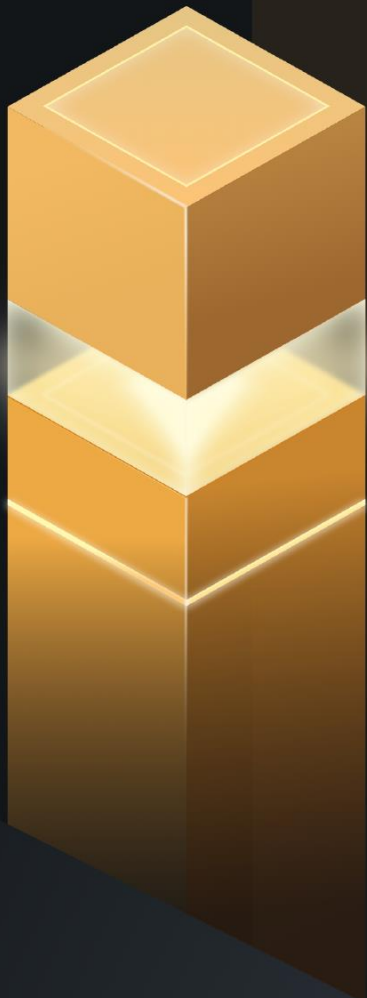
重视网络与信息安全建设，护航技术创新。今年6月中证协、中基协、中期协正式发布本行业《网络和信息安全三年提升计划(2023-2025)》，聚焦科技投入、IT治理、系统架构规划、研发测试管理等多个方面提供了工作指导。在今年的报告中，金融系统架构设计独立成章，Scale-up+Scale-out双轮驱动证券核心业务系统。针对研发测试管理，本报告给出了关于AI赋能质量工程以及业务连续性风险管理的深度洞见。

技术预判贴近业务的发展与变化。数字化浪潮下，金融业务向“以客户为中心”策略转变，呈现机构化、专业化、差异化、国际化的发展特点，对金融产品服务的个性化、敏捷迭代、快速开发、开放架构提出新的要求。从细分业务来看，证券公司将更侧重在财富管理、资产管理业务中寻找数智转型的突破口。在今年的报告中，我们可以看到金融行业大模型赋能投研、投顾、客服等场景；AI编码助手助力研发提效；事件驱动与服务的联合编织满足业务沟通结构多样化、的需求等深度思考。

数字化转型与信息技术应用创新纵深推进，《2024金融科技趋势研究报告》遵循监管指导、立足资本市场业务发展逻辑，不断拓宽“数智金融”想象的边界。恒生也将与金融机构共同探索，在未来实现创新驱动的科技成果转化，推进金融行业生产力的新征程。

目录

Table of contents



01

数智新应用

02

安全新态势

03

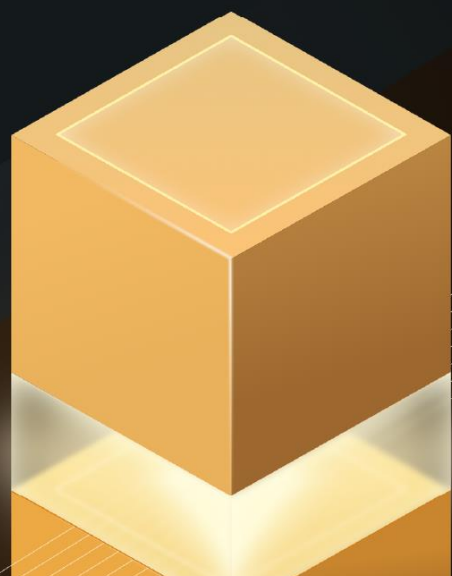
架构新未来

数智新应用

金融行业正处于智化转型和信息技术创新的双期叠加阶段，中央金融工作会议提出以加快建设金融强国为目标，以金融高质量发展助力强国建设和民族的复兴伟业。伴随着大模型的出现，创新技术和金融应用场景相互交织，成为资本市场发展的核心驱动力之一。

行业在推动大语言模型应用落地的过程中，不仅需要大模型本身的建设，更需要构建连接大模型和行业应用的“中控”基础设施，为投研、投顾、客服、风控等场景带来崭新的想象空间。与此同时，低代码平台与代码辅助工具的融合，全链路赋能研发流程，将领衔软件工程3.0时代。大模型在质量工程领域的初步落地，也为其带来了更智能和更高效的解决方案，AI正深刻影响着金融行业的数智进程，推动行业的功能创新、性能创新与应用创新。大模型带来了新的视野，信息技术应用创新则开始全面加速，数据技术的融合创新将与业务实现“双向奔赴”，推进行业走向科技自强。

万物生长，适者生存。面向2030，我们将坚守技术创新，携手行业共同预见金融科技的数智化图景。





趋势一：大模型对场景普遍渗透，以“中控”为核心的生态范式初步定型

趋势简介

近期，国内外出现了许多大模型并形成了百模大战的现状。研究指出，金融行业头部机构会在未来 1 年内在相对成熟的场景中尝试引入大模型，以及生成式 AI 能力。大语言模型时代的到来，为投研、投顾、风控、研发提效等场景带来了新的想象空间，行业在关注大模型本身建设的同时，作为连接大模型与应用的“中控”成为了行业未来大模型生态的核心。

趋势阐述

国内外大模型的涌现，呈百模大战的态势，例如：OpenAI 的 ChatGPT、谷歌的 PaLM、Meta 的 LLaMA、清华的智普、华为云的盘古、百度的文心、阿里的千问、科大讯飞的星火、商汤的日日新等。预计 2024 年，已针对人工智能有领先性的探索且硬件配备较完善的金融机构，会尝试在一些业务场景中引入大模型以及生成式 AI 能力。此外，RPA 作为扩展 AI 落地的“最后一公里”，将助力大模型扩展其应用边界，且通过易于使用、易于管理的部署来帮助 AI 加速转型，增强认知决策能力以处理复杂的长链条业务，降低运维成本来提升应用价值。未来，金融行业在推动大语言模型的应用落地过程中，将会遇到生成式深度合成类应用行业的强监管以及数据安全境内保护的强监督。

专业化的面向资本市场的金融大模型：金融领域具有复杂的金融知识和特殊的投资决策需求。为更好地满足行业特性，研发金融领域垂直大模型势在必行，典型代表有彭博的 BloombergGPT 和恒生的 LightGPT。这类模型能够触达金融



市场动态，学习和掌握投资组合优化、风险评估、金融法律法规等相关领域的知识。

与大模型本身建设相比，大模型配套基础设施建设同样重要。通俗来讲，有了新工具，新工具的使用指南，也十分重要。构建连接大模型和行业应用的“中控”基础设施，对于像金融这样受到强监管的垂直细分行业在内容合规、业务适当性、部署模式和数据流向等数据安全方面都是至关重要的。

“中控”基础设施，由三大部分组成：共性大模型插件、大模型工具链（任务编排和指令流水执行）、行业资源图谱。

金融大模型插件：减少幻觉——连接文档、数据库、API、元数据

金融领域存在大量数据需要连接，如行情、上市公司信息、基金基础信息、金融机构内部知识库文档等，因此需要大量共享插件能力，例如文档的解析处理能力、自然语言转 SQL 能力（NL2SQL）、智能 API 调用能力等。这些能力以插件的方式存在，构成对大模型能力的重要补充。

更加智能的工具链：规划和执行——智能体（Agent）

与大模型配套的转接工具链，是大模型“中控”的基础部分之一。它负责处理输入输出、数据转换和模型部署，特别是任务编排和指令流水执行等任务。幸运的是，目前已经有很多这样的开源组件被证明是有效的，可供选择，但是现阶段基本是靠手工拆分任务和规划。

为了更加智能化地处理、执行未见过的复杂任务，AI Agent 是时下热门的一个方向。在 OpenAI 应用研究负责人 Lilian Weng 撰写的一篇万字长文中，她提出 Agent = 大模型 + 记忆 + 规划技能 + 工具使用，以「输入一段指令自动实现复杂任务拆分和函数调用」的场景为例来构建基础 Agent 流程，并侧重讲



解如何通过「基础模型选择」、「Prompt 设计」等来成功构建「任务拆分」和「函数调用」模块。例如问“某个基金经理怎么样”，通过大语言模型拆分任务，对基金经理基础信息（姓名、学历等）、任职情况、历史收益等因素进行拆分，并完成相应的 Prompt 设计，最后调用相关基金经理卡片的 API 得到最后的结果，从而完成相对复杂的任务。

更专业的行业资源图谱

与大语言模型相配套的基础设施还包括应用侧的资源组织。从金融等垂直行业的需求来看，由于语料原因，单纯依赖大模型本身可能在短时间内没法快速满足金融业务场景的深度需求，需要在运行过程中结合丰富的应用侧专业化数据资源和计算资源，才能为面向特定应用场景的人工智能赋能。比如近几年新出现的向量数据库和图数据库，在应用侧资源组织中发挥着重要作用。

向量数据库主要应用于大模型训练、推理和知识库补充等场景，在接入层、计算层和存储层等方面已实现了全面的语义化。向量数据库可以用于大模型预训练数据的分类、去重和清洗等任务。在大模型学习阶段，向量数据库接收多模态数据进行向量化表示，让大模型在训练时能够更高效地调用和处理数据。在大模型训练方面，向量数据库的应用也非常广泛，此外大模型在调用 API 过程中，向量数据库通过对 API 进行检索，也可以确保 API 调用后执行得到相应的实时数据。

因此，数据库、图数据库对于组织企业和行业内部的数据要素资源、应用接口资源、流程角色资源和业务逻辑资源至关重要。我们可以将其打造成“中控”核心引擎，实现与大模型的统一对接，通过工具链将大模型解析的用户意图与企业和行业资源进行编排，按需集成和通联。这一环节也是未来制定大型语言模型行业标准的核心环节，具有重要的战略价值。因为在垂直领域的场景中，以大模型为



中心的插件联盟模式并不适用，而以中控为中心的企业和行业人工智能能力中心模式，将更具领先优势。

典型场景

大模型正深刻影响着金融行业的智能化水平和数字化程度，开启了金融行业智能化转型升级新的大门，从场景层面来看，大模型与金融行业有很多结合点，包括但不限于以下几点：

1) 投顾

大语言模型可用于辅助投资顾问工作，针对投资者生成个性化的投资建议，供投资顾问参考，极大地提高了投顾工作效率。通过分析客户的投资偏好、风险承受能力以及市场动态，大语言模型能够为客户初步生成符合其需求的投资策略。此外，大语言模型还可以用于辅助资产配置、组合优化、基金筛选等投资管理工作，提高投资效率和收益。

2) 客服

金融机构可以利用大语言模型搭建智能客户服务系统，针对客户咨询提供 24 小时不间断的在线服务和有效的解决方案。智能客服可以处理包括账户查询、交易操作等各种类型的问题，人工客服的工作量将大量减少。同时，智能客服还可以实时学习和优化，不断增进对客户的洞察理解，提升服务品质，这种进化能力超越很多金融从业者。

3) 舆情风控

大语言模型可以赋能金融机构监测舆情，及时挖掘潜在风险。大语言模型在海量数据基础上进行预训练，通过对互联网上的新闻、社交媒体等公开信息进行实



时分析，可以挖掘出可能对金融市场产生影响的重要事件和趋势，更好地理解 and 响应客户需求。

4) 运营

金融机构可以利用大语言模型对内部运营数据进行深入分析，发现业务流程中的痛点和优化点。大语言模型可以辅助金融机构挖掘客户需求、优化产品组合、提高市场推广效果等，进一步提升运营效率和客户满意度。

5) 投研

在投资研究领域，大语言模型可以辅助分析师快速获取和处理大量信息。例如，对公司财报、行业研究报告等文本数据进行自动摘要、关键指标提取，提高研究效率。同时，大语言模型还可以通过对海量历史数据的挖掘，发现投资机会和风险信号，辅助投资决策。

6) 监管合规

随着注册制的推进，金融市场对投行底稿的审核要求越来越高。大语言模型可以帮助投行和监管部门更高效地审核 IPO、并购重组等项目的底稿资料，通过自然语言处理技术，分析底稿中的文字、数据和图表，快速发现潜在的问题和风险。

信息披露审核是金融监管的一个重要环节，主要关注上市公司、证券公司等金融机构的公开信息披露是否真实、准确和完整。大语言模型可以辅助监管部门自动化地审核金融机构披露的信息，识别出不符合规定的内容，提高审核效率和准确性。同时，大语言模型还能辅助金融机构进行自查，确保信息披露的合规性。

7) 软件研发提效

大模型技术的引入将对信息技术每一位从业者产生深远影响，从软件开发者到产品经理，再到用户支持团队、运维团队等，都将在生产力的提升和工作方式



的变革中受益。

📦 机构 IT 规划建议

1) 理场景

确定应用场景。首先, 金融机构需要明确希望在哪些方面应用大模型, 如投顾、投研、客服、运营等。然后, 针对不同的场景, 制定合适的模型应用计划。

2) 储算力

提前规划。鉴于 GPU 采购周期较长, 建议机构提前半年规划和储备适用于各种场景的 GPU 资源。这样可以确保在实际需要的时候, 能够及时获得足够的计算资源。

3) 选模型

确保机构选择的大模型在金融领域有良好的表现, 可以通过与业内专家合作、调研相关文献等方式进行评估。避免盲目选择通用模型, 而是根据实际需求选择专为金融领域设计的模型。

4) 建队伍

外部分享与学习。组织团队成员参加行业内的分享会、研讨会等, 以便了解金融领域的最新发展趋势和技术进展, 从而更好地指导大模型的应用。投资于培养团队成员的技能, 使他们能够有效地操作、管理和优化大模型。这可能包括深度学习、自然语言处理等领域的培训。

其他建议:

1) 积极参与行业标准建设：生成式人工智能目前仍处于技术炒作的巅峰期，其落地应用仍未出现大规模的典型案例，因此需要积极参与行业对大模型数据标准、应用标准、架构标准等一系列标准建设工作。

2) 推动行业级大模型平台的建设：机构在算力、云资源、模型本身储备有限，行业级大模型的建设将有助于解决机构间重复建设、数据安全、应用合规以及生态连接等一系列问题。

3) 谨慎预训练，投入产出平衡：考虑预训练成本。大模型的预训练过程可能需要大量的计算资源和时间，而且投入产出比可能不总是很高。在决定是否进行预训练时，需要综合考虑成本与预期收益。

4) 法律法规遵循：大模型可能面临模型风险等监管问题。金融机构在构建大模型时，要确保模型遵循监管机构的相关要求，遵循国家和地区相关的法律法规，例如参考数据安全法、网络安全法、个人信息保护法等，制定并执行相应的用户隐私保护政策。



趋势二：低代码融合 Copilot 等代码辅助工具，走向研发提效新纪元

趋势简介

作为 AI 增强开发的重要组成部分，以 Copilot 为代表的 AI 编码助手对低码行业产生了较大的影响，并引起广泛关注。低代码与 Copilot 将互相融合，全链路赋能到研发流程，包括编码前需求、设计的质量提升，到编码工作转向半人工的效率提升，以及编码后的测试、部署、运维智能化、自动化变革，低代码+Copilot 将会领衔软件工程 3.0 时代，给研发提效带来新的生机。

趋势阐述

2022 年末，OpenAI 更新版本至 GPT-3.5-turbo，GitHub Copilot 代码辅助工具的代码生成和智能助手能力大幅提升。AI 编码助手服务于研发环节提效，对低码行业产生了冲击，并引起广泛关注。

1. 代码辅助工具(如 Copilot)持续升级的影响

GitHub Copilot 基于 OpenAI 版本的持续升级对低代码行业可能产生冲击性的积极影响。它有助于提高开发效率、降低学习成本、提供定制化选项，但同时也需要开发人员注意代码质量和安全性的保障。这样的升级和进步有助于推动低代码行业的发展，为开发者提供更好的开发工具和体验。具体为以下几个方面：

1) 持续增强开发效率：随着 GitHub Copilot 持续升级，它的代码生成能力和智能提示能力将不断改进和优化。这将使低代码开发人员能够更快速地生成高质量的代码、快速原型验证和迭代。这进一步提高了低代码平台的开发效率和灵活性。



2) 持续降低学习成本：低代码平台旨在让非开发人员无需深入的编程知识也能快速构建应用程序。随着 GitHub Copilot 的不断升级，它可以为低代码开发人员提供更多自动化支持和智能建议，减轻对编程语言和框架的依赖。低代码行业的学习门槛将进一步降低，更多公民开发者可以与专业开发者协作。

3) 提供更多定制化选项：随着 GitHub Copilot 的版本迭代，可能会引入更多的定制化选项，允许低代码开发人员根据自己的需求和偏好进行调整。这可以满足不同项目和行业的特定需求，提供更灵活、可定制和个性化的低代码开发体验。

4) 辅助缓解代码质量和安全性问题：尽管 GitHub Copilot 可以辅助代码的生成，但质量和安全性仍然是必须关注的问题。低代码开发人员需要仔细审查生成的代码，确保其符合最佳实践和项目要求，避免潜在的代码漏洞和安全问题。伴随 GitHub Copilot 的不断升级，上下文空间扩大，允许在生成代码的同时，生成覆盖特定要求的测试用例代码，减少开发人员在这方面的工作量。

此外，该趋势相应的消极影响也存在，建议行业同步关注：

1) 降低对开发技能的追求：GitHub Copilot 的不断升级和自动化代码生成能力的提升持续降低了对开发技能的需求，但这可能导致一些开发人员过度依赖自动化工具，缺乏对底层代码和编程概念的深入理解，这可能会对他们的职业发展和解决复杂问题的能力产生负面影响。

2) 代码质量和安全问题：尽管 GitHub Copilot 可以生成代码，但生成的代码可能存在潜在的漏洞、性能问题或不符合最佳实践。低代码开发人员仍然需要审查和测试由 GitHub Copilot 生成的代码，以确保其质量和安全性。

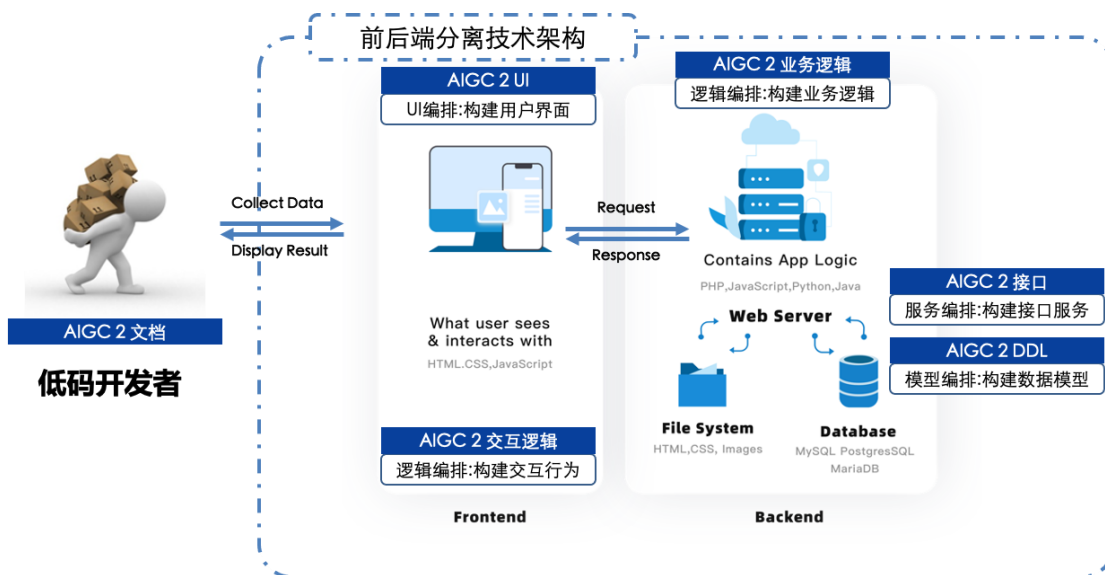
3) 依赖于第三方平台：GitHub Copilot 是基于 OpenAI 模型的 AI 助手，依

赖于第三方平台的支持和持续更新。如果该平台停止支持或没有及时更新，使用 GitHub Copilot 的开发人员可能会面临困扰，影响他们的开发流程和项目进展。

4) 限制创造性和创新：虽然 GitHub Copilot 可以加速开发过程，但过度依赖代码自动生成可能削弱了开发人员的创造性和创新能力。创造性的编码解决方案和创新的思维往往需要开发人员自己思考和实现，而不是完全依赖于自动化工具。

5) 数据出境风险：GitHub Copilot 服务位于境外，金融行业有更为严格的数据安全要求，使用 GitHub Copilot 则必然涉及数据出境，需要在具备数据合规出境的前提下才能适用。金融机构使用境外大模型必须考虑到生成内容合规性。

2. 低代码平台用户的工作场景



传统情况下，前端会调用后端接口进行数据处理和逻辑编排。在低码平台中，可以使用 UI 编排来实现低代码界面搭建，使用逻辑编排来构建页面间的交互行为和业务逻辑，而接口服务的构建则可以通过服务编排来实现。图示五个组成部分，是低码开发者在完成业务开发过程中的核心工作。由于业务的复杂度，此部分功能在可视化编排过程中，免不了需要使用脚本、代码块进行辅助完成。



为了屏蔽低码平台内部前后端技术体系之间的差异，构建属于低码平台自身的领域概念，业内绝大多数的低码平台，无论是解释执行还是出码执行，都会构建起一套完善的 DSL 体系，在设计环节代替源码完整表达上述各环节中包含的业务逻辑。DSL 作为一个语言体系，能够很好地消减业务差异和复杂度，并确保技术可行性和表达完整性。

综上，DSL 是低码平台业务开发的产出物，也是表达完整业务语义的载体。DSL 本质上并不能降低业务表达的复杂度。在一些较复杂的业务场景中，仍然对低码开发者有比较高的技术能力要求。也正是因为 DSL 的存在，利用代码辅助工具来辅助完成这部分复杂度较高的工作成为了可能。一旦分解任务、构建 Prompt、将生成特定的 DSL 设定为目标，所有这些低码平台中的复杂工作都可以利用代码辅助工具来辅助实现。

3. 代码辅助工具切入低代码平台开发场景分析

DSL 通常是为特定领域或问题域而设计的，具备与该领域相关的特定语法、语义和功能，与已经经过通用语言训练的代码辅助工具不同。如果希望代码辅助工具在未经训练的基础上协助生成特定的 DSL 代码，技术上需要把完整的 DSL 定义提供出来作为前提，此过程必须包含包括语法规则、关键概念、操作符、函数或其他与 DSL 定义密切相关的信息。

现有市面上大部分的辅助工具或类 ChatGPT 工具已经处于 GPT-3.5 模型级别。GPT-3.5 模型的最大输入长度限制是 2048 个令牌。一个令牌可以是一个字符，也可以是一个子词（subword，例如一个词的一部分）。对于 DSL 定义，字符的数量通常会转化为较多的令牌数量，这对于未经专门训练的辅助工具参与低码开发场景形成了比较大的限制。



一方面,行业可以将低码平台中,主要的业务工作划分为以下五个大的组成部分: UI 编排来实现低代码界面搭建,前端逻辑编排来实现客户端交互行为的构建,后端逻辑编排来实现业务逻辑的构建,服务编排来实现接口服务的构建,模型编排完成数据持久化层的对接和构建工作。每个组成部分具备相对独立的领域,形成低码平台 DSL 的一个子集。在做具体工作的时候,只传入必要的 DSL 子集定义,可以有效减少 DSL 定义对输入令牌的消耗,留出更多输入空间给到业务逻辑表达。

另一方面,OpenAI 版本也在持续升级。GPT-4 预训练阶段的上下文长度已经从 GPT-3.5-Turbo 的 2K 提高到 8K。后续还将在预训练后的 8K 基础上进行微调,形成 32K 的令牌长度版本。每个令牌的字节数也将当前版本的 4 字节扩大为 600 字节。最大输入长度限制的放开,将为动态提供 DSL 的完整定义创造条件。也规避了有限输入长度中 DSL 定义对业务表达空间的侵占。

综上,DSL 的存在对代码辅助工具切入到低代码平台开发的具体场景创造了条件。开发人员利用代码辅助工具快速生成应用页面或逻辑,再利用低代码平台进行微调、补充和确认。以此规避代码辅助工具与生俱来的安全风险,降低低码平台使用的门槛,并大大提高业务开发的效率。

典型场景

低代码平台与代码辅助工具完成融合之后,UI 编排、前端逻辑编排、后端逻辑编排、服务编排、模型构建等以 DSL 来进行设计和表达的任务,都可以通过代码辅助工具辅助实现。开发人员的主要工作,聚焦于使用低代码平台设计器对辅助生成的 DSL 进行确认、补充和微调。详细的融合点场景如下:



1) 自动生成表单和界面：通过整合 GPT 模型，低代码平台可以自动生成基于用户输入和需求的**表单和界面**。开发人员只需提供必要的信息，GPT 可以帮助生成相应的表单字段、布局和交互逻辑，减少开发人员手动设计和编写的工作。

2) 代码自动生成：利用 GitHub Copilot 的能力，低代码平台可以提供更强大的代码自动生成功能。例如，在低代码平台的可视化界面中，开发人员可以通过拖拽和配置来定义**业务逻辑**，然后通过集成 GitHub Copilot 来自动生成与逻辑相关的代码片段，例如数据操作、网络请求等。

3) 智能代码补全和提示：通过整合 GitHub Copilot 的**智能代码提示**，低代码平台可以提供更智能、个性化的代码建议。根据用户的输入和上下文，Copilot 可以自动补全代码、提供方法、类名等建议，以加速低代码平台上的开发过程。

4) 自动化流程生成：GitHub Copilot 可以帮助低代码平台生成自动化的**工作流程**代码。根据开发人员的定义和配置，Copilot 可以自动生成处理流程、状态转换、事件触发等代码，这样就可以轻松地实现复杂的业务流程。

5) 定制化代码生成模板：在低代码平台中，可以为开发人员提供定制化的**代码生成模板**。通过结合 GitHub Copilot 的能力，开发人员可以在模板中插入变量或条件，使得生成的代码更加灵活适应不同的需求。

机构 IT 规划建议

在低代码平台与 AIGC 持续融合的大背景下，对机构 IT 规划的下一步建议如下：

1) 深化 DSL 体系的理解和应用：进一步深化 DSL 技术体系的理解和开发，根据金融行业的特点和需求，构建垂直业务领域的 DSL，并将 DSL 应用于低代码



平台的业务开发过程中，以降低业务表述的复杂度、提高开发效率、确保业务语义表达的准确性。

2) 加强自动化代码生成与优化：在低代码平台中，加强与代码辅助工具的整合，探索更多领域和场景下的自动化代码生成和优化。通过自动生成代码段、模块或整个功能模块，以及优化生成的代码质量和逻辑性，进一步提高低码开发速度和质量。

3) 强化数据治理和安全保障：随着低代码平台中 AIGC 的应用程度逐渐深入，要加强数据治理和安全保障。面向 AIGC 生成数据的不确定性，要确保数据的质量、完整性和安全性，满足内容合规性要求。加强数据隐私保护和安全防护措施，同时建立健全的数据管理和治理机制，确保数据的可靠性和可追溯性。

4) 持续学习和创新：持续关注低代码和 AIGC 技术的发展和 innovation，积极参与行业活动和社区交流，与同行分享经验和案例。通过不断学习和创新，探索更多的应用场景和最佳实践，为结构 IT 部门带来更多的业务价值和竞争优势。

低码平台利用其 DSL 特性承接和阐述业务场景，将更加充分地发挥 AIGC 智能化生成能力，使其脱离基础技术栈的约束，展现出更加强大的创造性。伴随低代码平台与代码辅助工具持续融合，未来的业务创新将更加技术无关化、平台无关化。



趋势三：AI 加速测试智能化进程，开启质量工程新征程

趋势简介

随着大模型技术在垂直行业的应用探索，其在代码检测分析、测试用例生成和补全、自动化测试脚本生成（RPA）、质量评价和风险管理、线上故障检测和预防等领域逐步落地，AI 不仅提高了这些场景中测试的效率和质量，也为质量工程带来了更智能和更高效的解决方案。

趋势阐述

AI 技术的应用场景，核心在于发挥各类大模型的特征能力，如自然语言处理、科学计算、多模态转换、语音识别、计算机视觉、逻辑推理等。通过一些大语言模型的能力，可以自动分析测试需求、生成测试用例、执行测试和分析测试结果，在测试过程中重要的自动化测试、代码检测、问题分析定位环节中减少人工干预，使测试过程更智能化更高效。

随着大模型（典型如：LLaMA2，GLM2）的开源，垂直行业的应用深入发展，以目前最具典型代表代码检测分析、自动化脚本、模糊测试为例，各技术提供商已进入到深度集成的阶段，也会彻底改进原来通过规则匹配的方式带来的高误报、高转换维护成本的问题，也会促使该部分的应用实现真正的智能化。

典型场景

1) 测试用例生成和补全：AI 可以根据特定行业知识（如金融行业）、应用程序的规范、历史测试数据和用户行为等信息，自动生成具有高覆盖率和效果的测试用例和测试数据，提高测试效率和质量。并且根据测试覆盖数据的搜集，自动



调整或生成新的用例，并从原有的案例库中根据变动部分选取、推荐更加匹配的用例，从而提高测试覆盖率。这类场景会在单测、接口测试、模糊测试等领域率先落地。再进一步可以利用 AI 大模型的多模态输入、总结、推理特性，输入系统的业务流程图、架构图等需求、设计文档，对测试方案和用例进行多个方面的优化补全。

2) 测试代码和脚本生成：AI 可以助力更智能、更高效的自动化脚本开发。通过使用 AI 技术，模型可以根据用例描述，自动识别应用程序的界面和操作流程，并生成相应的自动化脚本。这将加快自动化脚本的开发速度，并提高脚本的稳定性和可维护性。AI 还可以通过大量既有代码、单测代码的模型数据的训练，结合语法、结构和规范的要求，针对性地生成和完善单元测试代码。帮助开发团队提升单测能力和覆盖面，提高代码质量。结合多种数据源和技术手段，如图像识别、语音识别等，AI 还可以实现多模态融合，极大提升测试用例的覆盖率，更好地应对各种测试场景和需求。

3) 代码检测分析：利用 AI 技术对代码进行静态分析和检测，通过分析代码的语法、结构和规范，检测发现潜在的代码缺陷、安全漏洞和性能问题，并给出改进建议，帮助开发团队提高代码质量、降低排错成本。

4) 质量评价、风险管理：AI 在质量工程中的应用还体现在质量管理过程，包括质量数据分析和风险预防。模型可以分析测试结果、代码质量、线上监控和其他相关数据，提供准确和全面的质量评价指标和风险预测。这将帮助质量保证(QA)团队评估和监控整个研发过程的质量，并提前发现潜在的质量风险。再进一步，可以结合一些业界通用的质量维度和评估画像模型，以模型图谱的形式直观地呈现质量风险和薄弱部位。



5) 线上故障检测和预防：AI 可以通过线上运维监控系统监控应用程序的运行情况和各类指标状态，识别异常行为的发生模式，结合原有系统的架构设计、部署设计预测潜在的故障，有助于提前采取措施，防止系统故障。将这类异常发生场景总结为可靠性测试案例，可以提升系统的可靠性，反过来也可以通过累计的故障库去强化模型的故障识别能力。

总体而言，AI 在质量工程领域的应用场景丰富，但目前仍处于探索阶段，还需要进一步的研究和实践来完善和提升应用效果。

机构 IT 规划建议

机构将 AI 应用于的质量工程领域建议循序渐进，区分上述多种场景，在某一场景下得到良好应用后，再探索应用其它场景，以确保成功地落地这些技术并实现预期的效果：提升产品质量、降低运行风险。

1) 测试代码和脚本生成的场景应用，比较适合作为先期落地的场景。一方面业界已经有现成的案例和开源工具，近些年也涌现出一些不错的商业化产品。另一方面生成的脚本和代码会经过人工的审核才会应用于测试实践，对生产环境基本无风险。在这个场景中，测试效率会获得比较明显的提升，提升大家对 AI 应用场景落地的信心。

2) 测试用例生成和补全适合作为第二阶段。AI 在内容生成方面具有优势，用例补全的场景容易落地，但部分情况下，大模型的逻辑推理能力稍显不足，用例编写本身有一定的逻辑性要求，AI 技术生成的用例的覆盖面是否能够满足质量覆盖要求还存疑。AI 结合现有的一些数据结构理论，在测试数据生成方面具有明显的效率优势，也会比较容易落地。



3) 质量评价和风险管理、代码检测分析、线上故障检测和预防适合作为第三阶段的场景落地。一方面机构本身有一定的历史数据和代码积累，有一定的落地基础；另外一方面使用 AI 进行优化提升的效果，还需要一段时间的实践验证，特别是对代码检测发现的问题和识别出来的风险点，需要经过充分的测试和验证。这种场景的落地时间较长，且对机构的日常运营会带来一定的影响，需要谨慎评估。

以上 3 个阶段的具体步骤和时间都可能因机构的规模、资源、技术要求和目标而有所不同。需要根据实际情况进行定制，并逐步推进。同时在整个过程中需要充分考虑合规性和数据安全问题。



趋势四：数据技术的融合创新与业务实现“双向奔赴”

趋势简介

对数据的离线和在线处理方式未来会呈现一种融合的趋势。这种趋势将促使数据技术需要能够同时处理在线的事务处理和离线的分析查询，实现一种混合负载的模式。未来，数据可能不再被严格限制于离线的大数据平台，而是被拆分后用于各种数据应用，以数据编织的形式提供数据服务能力。

趋势阐述

数字化业务时代，金融机构数据技术的建设重点正逐渐与业务创新、战略目标深度融合。数据领域的技术融合创新的态势主要由以下技术趋势驱动，将同时赋能在线事务处理和离线分析查询两大场景：

数据编织（Data Fabric）技术走向成熟。通过对数据的虚拟化和服务化，数据编织可以更好地管理、集成和处理不同数据源的数据，组织成实时、综合的数据视图，快速解决多源、多类型数据的协同问题，方便企业可以更高效、更直接地理解和利用这些数据。

流批数据协同。在整合和协调处理实时数据流以及存储的批量数据时，可以通过采用统一的数据处理引擎、编程模型和数据管理层来实现，以帮助用户在处理流式和批量数据时能够更加一致和高效。

业务数据服务化。传统的数据平台设计复杂，包含大量中间件和技术栈，需要高昂的成本和复杂的运维。企业迫切需要以更灵活、快速的方式完成数据处理，实现数据快速流动，价值实时变现，这其中需要业务系统一同转变，一方面做好交易管理，一方面做好数据服务。



HTAP 能力进一步加强。通过在 OLTP 分布式数据库上发展的 HTAP 关键技术，实现一套引擎同时支撑业务系统运行和分析决策场景，避免在传统架构中，在线与离线数据库之间大量的数据交互，为混合负载场景的应用系统开发提供了便利性，大幅提升面向复杂查询场景的处理能力。

典型场景

1) 实时风控：基于流数据以及批数据的数据技术应用，是实时风控指标数据计算、分析和应对的关键组成部分。要满足实时风控指标计算的要求，通常会采用流式计算引擎、实时数据库、复杂事件处理（CEP）等先进技术。这些技术有助于高效地处理实时数据流和批数据调用，同时确保在低延迟和高可靠性的条件下进行风险计算和决策。

2) 业务流程协同：应用间、系统间的业务协同必然导致数据的协同。应用开放数据服务可以将业务系统中的数据通过开放的接口和标准进行共享和访问。这有助于在不同系统之间更加灵活地访问和调用数据，实现数据的快速流动。

3) 交易分析一体化：业务系统既要支持交易数据的产生和管理，也要支持数据在本地的快速查询、分析。满足用户实时分析、实时交易的业务操作连贯性，同时确保数据分析对业务交易 0 干扰。

4) 去中台化：数据中台大而全，但未必适合所有的企业，数据中台通常是一个更加封闭的数据平台，架构重，其目标是在一个中心化的环境中整合和管理数据。数据编织更注重开放性，通常采用开放标准和建立生态系统，以便更好地支持不同数据源和工具的集成。

📦 机构 IT 规划建议

1) 选型维度: 信创是大趋势, 根据业务系统/生产应用/资源投入的不同等级, 可以为性能设置不同的基准线。

2) 实施步骤: 小步快走、灰度推进。在关键业务系统中按子系统或者模块逐步做验证和技术切换。

3) 建设规范: 从应用建设、技术开发和数据处理、数据管理等多个角度, 要充分考虑国家规范、行业规范、企业规范、自定义规范。

4) 运维管理: 硬件监控、系统监控、数据监控、服务监控, 要逐步智能化、自动化, 从事中、事后逐步转为事前主动发现, 主动告警, 预防问题的出现。

安全新态势

技术自立是国家发展的强盛之基，技术创新是国家现代化建设的核心力量。与此同时，科技安全和业务发展的平衡，对信创时代新格局至关重要。核心技术作为国之重器，其安全性问题不容小觑。

金融行业信息技术应用创新背景下，业务连续性风险管理将成为金融科技安全的新内容，行业迫切需要加快软件供应链安全治理工作的落地实施。随着信创进入深水区，先进替代，即推动新技术和新架构的应用完成信创替代，将会成为核心内涵与要求。在这个过程中，内存技术将发挥关键作用。

自主可控不是安全的充分条件而是必要条件，站在新的历史起点上，构筑数字中国背景下的信息安全，是全行业聚各方众力，为业务保驾护航的长远命题。





趋势一：业务连续性风险管理将成为金融科技安全新内容

趋势简介

2023 年 6 月中证协、中基协、中期协正式发布行业《网络和信息安全三年提升计划(2023-2025)》，推动证券/基金/期货公司加强网络与信息系统安全稳定运行和保障体系和能力建设，防范化解系统性风险，对系统运行保障和全生命周期的安全治理提出了新的要求。行业迫切需要加快软件供应链安全治理工作的落地实施，金融机构信息技术和软件资产的独立自主、安全合规成为了行业中的重中之重。

趋势阐述

作为国家关键基础设施，金融机构的网络与信息系统安全稳定运行是金融安全乃至国家网络安全战略中至关重要的一环。随着数字化转型的深入，金融系统的软件资产及其供应链的复杂度也在不断增加，面临着更加复杂的安全威胁和风险，迫切需要加快软件供应链安全治理，防范化解系统性风险。

行业近年来也在不断出台相关规范和指导意见，逐步完善软件供应链安全管理工作。2021 年 7 月国务院公布《关键信息基础设施安全保护条例》，强调了保障关键信息基础设施的供应链安全，才能维护国家安全。2022 年 1 月中国人民银行印发《金融科技发展规划(2022-2025 年)》，提出了切实保障供应链稳定可靠，构建稳健高效的关键核心技术金融应用供应体系。2023 年 6 月中证协、中基协、中期协分别发布本行业《网络和信息安全三年提升计划(2023-2025)》，均强调了证券、期货、基金管理公司需要加强软件供应链管理，防范软件供应链风险。同时近年来，金融业积极开展商用密码应用工作，通过与密码产业联合创新，形成



了完整且安全可控的金融业商用密码产品服务供给体系，这类关键技术应用也防范了隐性风险和不确定性风险。

全链路软件供应链安全治理是实现关键核心技术和基础设施自主可控、安全合规的必要手段。传统软件供应链治理更多聚焦在开源治理方向，仅关注开源软件的安全风险，治理重点则在软件开发阶段的风险发现和移除，对于自研软件资产、商采软件资产在引入、集成、运行等阶段缺乏有效治理措施和手段。只有将软件资产进行全面梳理，构建全维度的软件资产台账，从引入到运行对整个软件生命周期进行综合的风险治理，实现精准实时的风险感知、风险追溯和风险预警，才能为金融机构提供全方位的供应链安全保障，为金融机构技术发展战略、信息安全战略、信创战略的落地保驾护航。

全链路的软件供应链安全治理可以帮助金融 IT 厂商和金融机构更好地管理和控制软件的风险，提高软件的可信度和安全性。通过引入软件物料清单（SBOM）对软件资产进行全面管理，而 SBOM 可以借助软件成分分析（SCA）类工具进行自动生成。借助全链路的视图和细化的安全分析，可以实现对整个软件生命周期的全面管理和控制，从而降低潜在的安全风险和威胁。

全链路的软件供应链安全治理应重点识别软件资产中所涉及关键数据保护所采用的密码算法，确保密码技术应用的合规性、正确性及有效性。

随着全链路安全治理的标准和实践的逐步发展，金融行业将实现建立适合自身的安全治理方案，并有效应对软件供应链安全带来的挑战。

典型场景

1) 软件供应链管理场景



软件供应管理职责,是从源头解决存在风险的软件被引入的情况。针对自身引入情况,可以对引入的软件进行全面风险评估,提前识别风险,并制定准用条件,规避风险软件被引入。

针对商业采购引入的情况,可要求软件供应商提供软件物料清单(SBOM)和供应商风险评估结论,进而及时和全面了解商业采购软件资产的物料组成及风险情况,并纳入统一管理。供应商风险评估应重点进行技术风险评估,如:技术可控、安全合规等情况。

2) 软件资产管理场景

依托全链路的软件供应链安全治理体系,软件资产管理和技术管理人员,可以从全视角掌握企业内部使用的二方、三方软件/组件及其加解密算法等关键技术的情况,为企业内部的技术规划、架构选型、研发管理提供数据支撑。

3) 软件研发使用场景

在软件研发使用时,针对软件中使用到的二方、三方软件/组件及其加解密算法等关键技术,可以通过将 SCA 等工具接入 CI/CD 流水线进行自动化风险扫描和卡点,杜绝有风险的软件被集成发布。

4) 软件风险管理场景

在软件资产被披露出安全漏洞后,安全人员、研发人员、合规人员等角色可以快速地依托全链路的软件供应链安全治理体系,识别出风险软件的影响面,并针对性的高效处置。对于软件供应商来说,可以进一步掌握客户影响情况,及时做到风险预警和应急处置预案。对于金融机构来说可以有效实现安全治理的目标。

📦 机构 IT 规划建议



机构要实现全链路的软件供应链安全治理的快速落地，首先要明确目标和原则，并依据这些目标原则制定具体的实施方案。在实施过程中，以下四个关键要素需要得到重点关注：

1) 覆盖全流程：全链路的软件供应链安全治理应覆盖从需求收集、设计、研发、测试、发布、部署到运维的整个软件研发流程。每个环节都应当纳入安全治理的范围，以确保整个软件供应链的安全性和可靠性。

2) 全面治理：在全链路的软件供应链安全治理中，不仅要关注软件本身的安全性，还需要考虑软件成分的安全性。同时，还需要考虑软件供应商或者开源贡献团队是否可信，以及软件成分和所涉及采用的关键技术（如加解密算法）是否满足合规要求。

3) 工具链选择：实施全链路的软件供应链安全治理需要借助一系列的工具，可以选择成熟的商业工具，也可以考虑开源工具。在工具选择时，应考虑工具是否能满足全流程覆盖、全面治理、安全可控等需求。

4) 建立安全管理制度：全链路的软件供应链安全治理应与企业的研发流程相融合，并建立一套完善的风险处理流程及应急响应机制，明确各方的职责与义务。这有助于在遇到安全问题时，迅速采取有效的应对措施，降低潜在的风险和损失。

5) 人员的培训和管理：全链路的软件供应链安全治理不仅需要技术的支持，还需要人员具备足够的安全意识和技能。因此，定期开展培训是非常必要的，有助于提高员工的安全意识和技能水平，确保软件供应链的安全性和可靠性。



趋势二：先进替代是信息技术应用创新的核心内涵和要求

趋势简介

信息技术应用创新（简称信创）已进入深水区，金融行业信创需要实现先进替代，金融机构以信创作为切入点，推动新技术和新架构的应用进而完成信创替代，是信创的核心内涵和要求。在这个过程中，行业需要通过软件补全硬件不足、业务逻辑与存储分离、架构重构等诸多方式来解决信创过程中的性能、体验等问题。其中最关键的技术是内存技术。

趋势阐述

2023 年 7 月 27 日，习近平在四川考察时强调：以科技创新开辟发展新领域新赛道、塑造发展新动能新优势，是大势所趋，也是高质量发展的迫切要求，必须依靠创新特别是科技创新实现动力变革和动能转换。因此，信创不是简单的国产替代，而是瞄准下一代信息技术与数字经济的创新发展，实现先进替代。

尽管长远的目标是实现更先进的替代，但现阶段不管是基础硬件还是软件，从芯片、服务器到操作系统、数据库，信创产品与非信创产品之间还存在明显的差距，特别是在性能和可靠性方面。在这个基础上，怎么实现整体的更快、更稳，需要在软件的技术和架构上创新。

众所周知，可靠性是软件系统最关键的要求，而可靠性取决于软件系统依赖所有组件可靠性的乘积，这些依赖的组件可能包括从芯片、操作系统到网络、数据库以及内存、硬盘等所有的设备和基础软件，基本规律是依赖的组件越多，可靠性越差。当基础的组件可靠性小于 1 时，最终软件的可靠性将远小于 1。因此，提高可靠性的关键路径是减少组件依赖，而内存技术就成为关键。引入内存技术，



减少了对硬盘和数据库的依赖，也减少了对网络的依赖，减少了依赖的组件后，在单机层面的可靠性将有效提升，分布式架构也将进一步通过集群和分片提升整体的可靠性。

信创基础组件在性能方面与非信创也存在明显的差距，特别是传统的金融系统软件基于关系型数据库进行搭建，而关系型数据库部署在高性能 PC 服务器、小型机、中型机甚至大型机等服务器上，充分发挥服务器的稳定、高性能优势。在信创环境下，怎么保持性能不落后甚至领先是迫切的要求。那么内存技术、低时延平台成为了金融客户的首选。使用内存技术可以显著加快数据的读写速度，提高系统的响应能力和性能。并且进一步融合 CXL 技术以及其他加速器如 GPU 和 FPGA，形成一种强大的高性能数据处理平台，实现更高的数据处理速度和更大规模的数据分析能力。通过使用更大容量的内存和更高效的内存管理技术，可以加快数据的读写速度，提高系统的响应能力和性能。

内存技术对我们的系统开发提出了更高的的要求。一方面，我们必须确保业务系统的开发者从繁琐的技术细节中脱离出来，使其专注在业务领域，以达到快速的迭代、业务系统的健壮性。在这个层面，就需要有一套懂业务且和业务高度贴近的高可用的平台。在降低业务开发难度的同时，平台还需要数据的正确性、一致性，确保数据不丢失。低时延开发平台(包括：分布式一致性协议、高吞吐低延迟的通信机制、快速高效的内存资源管理：可以高效进行数据增删改查的内存数据组织形式等)成为开发券商核心业务系统的基石。另外一方面，即便有平台的助力，全内存的业务系统开发对开发者的技能要求也比 DB 模式更高，开发也更容易出错，那么这时候就必须有相应的工具来进一步降低业务开发难度，减少业务系统出错的可能性。在这个层面，一套专业的、适用于金融领域的开发工具，



就显得尤为重要。借助开发工具强大的表达能力，使用尽可能少的、简单明了的逻辑，来生成复杂的业务逻辑，同时也使得问题更趋向于收敛。

此外，先进替代还包括优化软件技术以解决硬件局限性的其他方面。例如，通过软件优化和算法改进，可以提高硬件的效率和利用率，从而实现更好的性能。当然，随着基础设置的完善，性能和可靠性方面信创与非信创拉平甚至赶超后，整体系统的性能和可靠性才能实现全面的先进替代。

典型场景

内存交易从专业投资者逐渐向普通散户普及。金融交易系统往往面临大量的并发交易请求，内存数据库的高并发处理能力可以满足这一需求，确保稳定的系统性能和高吞吐量的交易处理能力。另外，交易系统对于低延迟的处理速度要求极高，内存数据库能够提供快速的读写和实时的数据处理能力，帮助金融机构实现快速交易执行。

内存风控从事前风控向事中、事后风控延展。内存数据库能够支持实时风险管理，通过快速访问和处理大量的交易数据，帮助金融机构实时监测市场风险、评估交易风险，提高风险控制的能力。

内存清算从小数据量（证券）向大数据量（TA、估值）场景铺开。在内存清算中，实时计算非常重要，可以及时处理和结算交易数据，以降低风险和提供即时的决策支持。内存数据库具备出色的并发处理能力，能够同时处理多个交易请求，确保即时清算。在高峰时段，内存数据库能够有效地应对并发请求，保持良好的性能表现。

内存数据分析在策略场景逐渐深入。内存数据库能够快速回放和分析历史交

易数据，帮助金融机构进行模拟交易、回测策略和优化交易算法，提高交易决策的准确性和效率。

📦 机构 IT 规划建议

1) 投资内存技术：金融机构应积极投资先进的内存技术，培养和储备内存技术先进人才，掌握内存技术相关的开发、测试、运维相关的技术，探索内存技术使用场景。

2) 关注内存技术的开发：内存技术的可靠性是生命线。而减少内存开发环节的缺陷是可靠性的有效保障。需要重点关注内存技术开发相关的工具，并建立内存技术的使用规范，进一步通过开发工具封装内存技术的使用规范，屏蔽风险和简化开发，提高内存技术使用的可靠性和提高内存技术的开发效率。

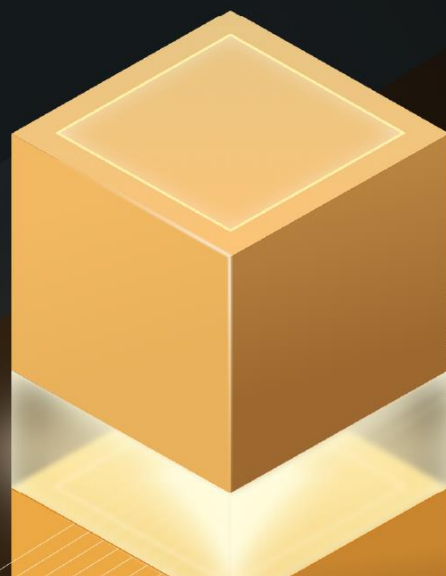
3) 关注双活（多活）架构：传统的内存有易失性的特点，在内存技术使用的同时，需要通过系统架构实现高可用性，需要关注并研究基于内存技术的双活（多活）技术。在可靠性要求最高的场景下，选择最合适的双活（多活）架构。

4) 关注内存技术的发展：关注和研究内存技术的发展，比如内存技术的持久化技术、内存压缩技术、内存加速技术以及非易失性内存技术等，积极的推进内存技术的发展和 innovation。

架构新未来

今年6月，中证协、中基协、中期协正式发布本行业《网络和信息安全三年提升计划(2023-2025)》，鼓励机构积极探索架构转型和技术创新，提升系统的健壮性和扩展性。随着金融业务模式的创新和发展，业务系统的复杂性和多样性也随之增长，IT系统架构成为构建证券科技新生态新未来的基础。

双期叠加对机构业务沟通结构提出了新的要求，包括多样化、快速变化和时效性等，通过新系统沟通结构来构建具有解耦、可扩展和可插拔特性的谈业务系统成为关键，事件/消息驱动加服务的联合编织模式将逐渐成为未来架构的核心趋势。同时，金融业务系统敏态稳态并存，核心交易系统作为证券业务系统的重中之重，如何在满足不断增长的交易量和用户需求的同时保持高可靠性和可扩展性成为关键命题。Scale-up和Scale-out双轮驱动策略将灵活地提高核心业务系统的性能和承载能力。此外，愈加复杂的软件架构和IT资源对IT治理提出了更高、更全面的能力要求，构建统一的运维平台则是企业IT治理核心关键。





趋势一：事件 / 消息驱动 + 服务的联合编织成为未来架构的核心趋势

趋势简介

随着数字化时代的到来,企业和组织面临着越来越大的挑战,对业务沟通结构也提出了新的要求。增强的业务信息共享需求包括多样化、快速变化和时效性等,传统的应用程序或服务间的数据沟通结构难以满足。因此,寻找新的系统沟通结构来构建具有解耦、可扩展和可插拔特性的企业数字化系统成为关键。在这样的背景下,事件 / 消息驱动+微服务的联合编织模式崭露头角,并逐渐成为未来架构的核心趋势。

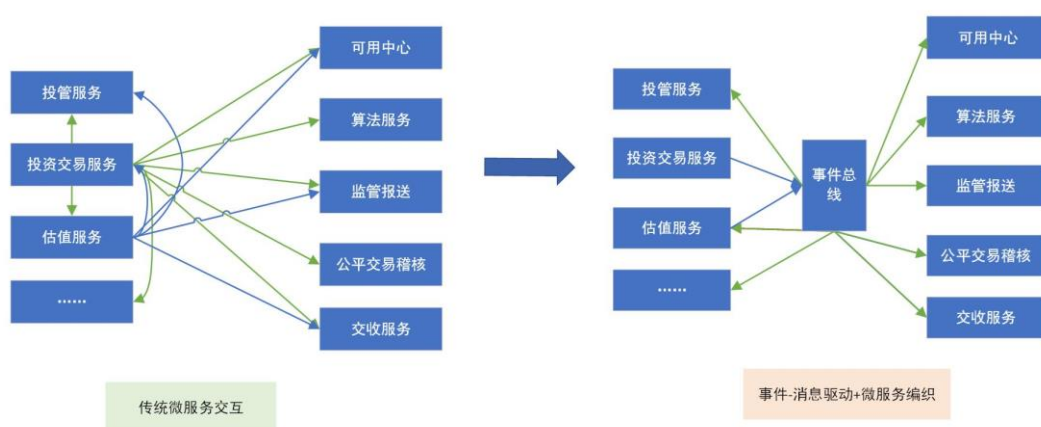
趋势阐述

事件驱动将重点放在了业务事件上。它通过捕获和处理公司运转过程中的事件来完成特定的企业内部的数据交换和流转。业务事件可以是各种类型,比如市场数据(如公司行为、财务报告等)、用户操作、传感器数据、状态改变等。触发事件后,系统中相关的服务会根据定义的规则进行处理,实现异步和实时的交互。这种架构风格可以提供更好的松耦合性、可扩展性和弹性,同时也支持消息传递、事件驱动和数据的发布-订阅模式。

微服务架构是指将应用程序拆分为一组小型、自治的服务单元,每个微服务都定向完成特定的业务功能,并通过轻量级的通信机制进行交互。通过业务域的新划分和涉及,解耦了原来单体系统中交织在一起的各个领域,微服务架构使得每个业务域的服务能够独立开发、测试、部署和扩展,带来了更优的灵活性、可伸缩性和可维护性,使应用程序能够更好地适应不断变化的需求。

但是单纯的微服务 RPC 交互方式，无法满足企业数字化过程中更多系统间交互。一般意义的发布订阅机制，更多是系统/服务间数据交互的一种技术手段。

基于以上背景和现状，事件/消息驱动架构与微服务结合起来，能够提供更加强大、灵活和可扩展的系统架构。它们相互补充，使得系统能够更好地应对复杂性、实时性和规模需求的挑战。目前，许多系统已经开始采用这种架构，落地了成功的应用案例，并且在未来将继续得到广泛应用和发展。



典型场景

1) 实时市场数据处理：资产管理业务需要实时监控市场资讯，以便及时调整投资组合来应对可能发生的风险或获取更高收益。事件/消息驱动架构可以通过订阅市场数据源，并将数据推送给关心该类信息的业务服务（比如研究、风险分析、内部评级等），实现实时数据处理和决策。

2) 交易执行和处理：资产管理业务需要执行和处理各种交易，如买入、卖出、调仓等。事件/消息驱动架构可以由相应的交易执行系统对各类交易事件的发布，驱动各事件关联方，通过消息发布订阅机制，将交易事件信息传递给交易分析、公平交易稽核、事中合规风控、估值等系统，实现交易事件信息的分发执行和处理。



3) 风险监控和预警：资产管理机构在产品成立时，往往需要将产品相关的信息传递给多个关联服务，如投资组合管理、投资交易、合规风险、估值、TA、CRM 等。事件 / 消息驱动架构可以通过运营系统发布产品成立事件、各关联服务订阅产品成立事件，从而实现产品成立信息的实时共享。后续如有新的服务需要获得产品成立事件，自主加入产品成立事件的消息主题订阅列表即可。

机构 IT 规划建议

传统的单体应用程序和集中式架构难以适应不断变化的市场需求、规模扩展的要求和快速交付的期望。因此，采用事件 / 消息驱动架构和微服务架构的事件编织成为必要的选择。

增强可组合性：事件编织使得服务能够以一种更灵活和可组合的方式进行组织和协作。这样，可以通过重新组合和构建流程，快速地构建新的业务流程或应用。

提升复杂业务的可控性：事件编织对于处理复杂的业务流程非常有帮助。通过定义和管理服务之间的交互和顺序，可以确保复杂流程的正确执行，减少错误和故障。

支持异步和实时处理：事件编织能够处理和控制异步和事件驱动的任务、消息和事件。这使得系统能够快速响应和处理实时的业务需求，提高系统的性能和用户体验。

提高可维护性和扩展性：通过事件编织，系统的不同部分变得松耦合，易于单独开发、测试和维护。由此，在需要增加、替换或修改服务时，可以更容易地进行调整和扩展，不会影响整个系统的稳定性。



实时性和响应性：事件编织架构实现实时事件处理和响应，使得系统能够更快地响应用户操作、外部事件或环境变化。这对于需要快速响应的业务场景非常重要。

由于不同机构业务规模、技术和运维力量等都各有差异，需要各机构根据自身的需要，合理地选择微服务的部署形式。恒生公司基于 JRES 平台的新一代产品，均支持展业机构按需部署：设计上能按业务域做到各微服务解耦，部署上能按需合并多个微服务到一个进程；也能按照容器化方式实现各微服务独立部署、弹性扩展；根据业务量在资源占用上可大可小、硬件要求可轻可重。



趋势二：Scale-up+Scale-out 双轮驱动证券公司核心业务系统

趋势简介

金融业务系统中存在敏态和稳态两种状态，稳态变动少，对稳定性的要求更高，敏态则相反。在核心系统中，Scale-up 即为用软件或硬件方式提升性能，无论是用硬件的手段还是软件的手段；非核心系统中，Scale-out 即为云原生能够容错、试错、快速灰度上线的一些措施，助力开发态、运行态中应用的快速上线。

“Scale-up”指的是通过增加单个服务器或计算机的处理能力来扩展系统。这可以通过升级服务器硬件、增加内存、增加处理器核心数等方式来实现。Scale-up 的优势是可以提供更高的性能和处理能力，但成本较高，而且有一定的物理限制。

“Scale-out”指的是通过增加服务器节点数来扩展系统。这可以通过在现有系统中增加更多的服务器节点来实现，每个节点负责处理一部分工作负载。Scale-out 的优势是可以在较低的成本下提高系统的承载能力，并且具有较好的横向扩展性。

通过同时采用 Scale-up 和 Scale-out 的双轮驱动策略，证券公司可以在需要的时候灵活地提高核心业务系统的性能和承载能力。这可以确保系统能够满足不断增长的交易量和用户需求，同时保持高可靠性和可扩展性。

趋势阐述

随着证券市场的不断发展和数字化转型，交易量和交易频率不断增加。投资者对实时交易和迅速执行的需求也越来越高。为了应对这些挑战，证券公司需要提升核心业务系统的性能和承载能力，以满足市场需求。另外，计算机硬件和软件技术的不断进步为系统的扩展提供了更多选择和可能性。在硬件方面，处理器的性能不断提高，内存和存储容量也不断增加，这使得单个服务器的处理能力大幅提升。在软件方面，



分布式计算、云计算和虚拟化等技术的出现，为系统的可扩展性和弹性带来了更多的选择。其次，作为金融服务机构，证券公司的核心业务系统必须具备高可靠性和容错性。如果系统出现故障或负载过高，可能导致交易延迟、数据丢失或业务中断，从而对公司的声誉和客户信任造成严重影响。通过采用"Scale-up + Scale-out 双轮驱动"策略，可以提高系统的可靠性和容错性，确保系统在高负载和故障情况下仍能正常运行。

分布式技术允许将系统的工作负载分散到多个服务器节点上并行处理，从而提高系统的性能和可扩展性；内存技术（如高速缓存、内存数据库等）在提高系统性能和响应速度方面起着关键作用。分布式技术和内存技术对于证券公司核心业务系统的性能和扩展能力至关重要。它们可以提高系统的并发处理能力、响应速度和可靠性，满足日益增长的交易量和用户需求。随着技术的不断发展，分布式技术和内存技术将继续演进并提供更多创新，为证券公司的核心业务系统提供更高的效能和竞争优势。

典型场景

1) Scale-out 场景

整个集中交易系统可以分为两类场景，一类是以交易、两融、期权为主体的通道业务场景，另外一类是以认证、账户、资金、查询、清算为主体的运营场景。运营场景中，重点是提高运营系统的迭代速度，逐步实现公司级的运营底座。

Scale-out 即云原生的核心目标是提升系统的迭代速度、实现业务 7×24 小时等，所以更侧重于关注运营系统场景。云原生系统不仅仅是要引入云原生相关的技术，更要实现微服务之间的松耦合（解耦）。只有在解耦的基础上，才能实现微服务的独立迭代升级，提高迭代速度，也能降低微服务之间运行的影响，更好地实现业务 7×24 小

时。再进一步,通过解耦还能实现系统的组件化支持,进一步提高不同部位的竞争力。

微服务的难点是微服务划分粒度、接口的制定以及接口的管控等。微服务划分粒度应该以康威定律为指导,将开发团队与业务系统对应起来,一般每个开发团队人数不超过 10 个人。接口是微服务的边界,需要保持一定的稳定性。接口的制定要以系统之间松耦合为目标,系统之间的接口数量应尽量少,并且减少频繁变更。接口制定需要首先规定好每个微服务的核心职责,围绕核心职责提供相应的接口,同时关注接口的抽象。同时,接口变更需要严格的管控,需要有专门的接口管控工具或者平台,需要做好接口的内外部审核。接口的变化需要做好向下兼容。

2) Scale-up 场景

全内存柜台:通过内存技术,将所有交易相关的数据存储在内存中,而不是传统的基于磁盘的存储系统。这样,交易数据可以以更快的速度读取和写入,大大提高了系统的响应速度。投资者下单或修改订单时,系统可以立即处理并执行,减少了订单处理和交易执行的延迟时间。全内存交易技术还使得券商可以更好地处理大规模并发请求。传统基于磁盘存储的系统可能无法及时处理大量并发的交易请求,导致延迟和瓶颈。使用全内存交易技术后,系统可以快速处理众多交易请求,满足高频交易的需求。

📦 机构 IT 规划建议

1) 做好企业架构:证券公司应该以公司战略为目标,规划企业级别的业务架构和企业级别的技术架构,逐步实现企业架构的落地。企业架构的规划和建设应该具有全局的视角,在建设上根据轻重缓急规划好建设路径。

集中交易系统的建设应该纳入企业架构的整体规划中,而不是单独作为一个独立



的系统存在。集中交易沉淀了大部分的客户资源与员工资源，应该将集中交易运营相关的部分规划为公司级系统，成为公司级的运营系统，实现整个公司运营体系的统一。

2) Scale-out: 在企业业务架构和技术架构的基础上，选择敏捷要求比较高的模块进行 Scale-out 试点。同时积累 Scale-out 即云原生相关的技术和能力，逐步建立云原生开发与运维体系。

3) Scale-up: 使用内存技术可以显著提升稳态业务的稳定性和性能要求。使用内存数据库作为交易数据的存储和管理系统。内存数据库具有高速读写和查询能力，可以实现实时数据更新和查询，大大提高交易数据的响应速度和吞吐量。



趋势三：标准化、智能化加速运维数智化进程

趋势简介

企业现有 IT 系统面临着来自不同系统和基础设施的异构性和复杂性,对 IT 治理带来了巨大挑战。为确保 IT 资源和系统能够有效地支持业务目标,构建统一的运维平台是企业 IT 治理核心关键。通过提供统一的监控、管理和运维工具,协调各个环节之间的工作,保证系统持续稳定运行,及时解决故障,迅速减少潜在风险。

趋势阐述

金融业务模式持续创新和发展,业务系统的复杂性和多样性也随之增长。同时,面对越来越严格的监管要求,如何保证业务连续性对运维形成巨大的挑战。面对大量异构系统,如何能够实时监控和评估业务系统的运行状态,如何能够在业务出现异常时快速发现与定位、及时止损与解决;面对不断变化的业务和更新的技术,如何引入新的技术和能力保证运维质量;这些问题都缺乏有效的工具和方法论支撑。大部分客户在不同运维问题上都使用专用运维工具,相互割裂无法串联,运维过程中在多个工具中不停切换,运维数据逐渐沉淀在各个工具中形成数据孤岛,具备较高价值但暂时没有得到合理的利用。同时面对有限的资源和预算,如何进一步优化资产管理,减少不必要的开支,提升资产使用效率,也是对运维提出的新课题。

这些需求驱动运维系统从数据标准化与数据治理开始,不断地集成、融合更多的技术与组件,构建起一个连贯、高效的智能化平台,通过数据治理、数据建模、关联分析、数据挖掘、智能分析、处置串联等方式,实时综合呈现业务运行状



态，自动分析判断系统异常、实现自主决策处理，并伴随业务的发展不断扩展和更新。运维技术不断发展，有以下主要趋势：

1) **运维对象关联建模**：构建数据管理平台，将各种运维对象（比如服务器、网络设备、应用程序等）的数据进行关联和建模，以便更好地理解它们之间的关系、相互影响以及运行状态。这通常需要对 IT 资产对象全局架构、分层分类，并借助 CMDB 配置管理数据、动态链路请求关系、生产消费者关联等数据快速构建关系模型。

2) **数据标准化与治理**：随着运维的不断发展，各种运维工具采集的数据越来越多，但是各工具采集的数据具有各自独立的格式，没有统一标准。未来，智能运维将更深入地融合与分析多种数据来源，如日志、指标、事件、配置等，建立统一的数据格式标准，并在各种数据间建立融合基础元数据（数据标注与注释），将各类数据标准采集、存储并有效串联，实现对各类数据全面、持续的收集与分析。

3) **异常检测和自动诊断**：通过对历史数据的分析和对实时数据的监控，对系统的工作状态进行决策树诊断、AI 智能模型分析，自动检测潜在的故障和异常。对于已经发生的故障，借助机器学习和知识图谱等技术实现自动诊断，并给出相应的解决建议，甚至可以自动完成故障修复操作。

4) **智能预测决策与处置**：利用深度学习和时间序列分析等技术对系统性能指标、应用行为、资源使用等方面进行预测，为运维团队提前发现潜在问题提供支持。同时，结合业务场景和运维策略，自动为运维团队提供合理的决策建议与自动处置。未来通过智能生成技术，自动组建原子操作并编排生成新操作流程自动恢复故障。



5) **自我学习和训练**：为运维系统赋予自我学习和优化的能力，根据系统实际运行状况和运维团队的反馈，不断完善运维知识和策略。这将有助于提高系统的预测和分析能力，同时减少人工干预和维护成本。

随着大语言模型与 AIGC 技术的不断成熟，可基于大语言模型构建运维领域专有模型，并将海量的历史运维数据通过格式化、清洗与标注，注入到分析模型，通过不断训练与调优以得到最优的运维分析模型，从而形成有效的可落地路径实现上述功能。

典型场景

随着企业数字化转型的不断深入，云原生微服务化已经逐渐进入成熟阶段，微服务业务的部署与运维成为重点需要解决的难题，业务实施从 Day0 到 Day2 均需要有平台化、智能化的运维工具支撑。

首先是大量服务的资源规划、部署与调度。由于服务数量急剧膨胀，传统的人工分配与调度已经不能胜任，需要平台能够依照统一标准快速地部署与升级业务系统，并依照服务的运维负载情况和总体计算资源池的使用情况，智能决策服务运行资源、合理调度，并指导部署服务动态的扩缩容以满足业务的连续性与容量要求。同时也需要提前预测业务峰值以及资源容量，以便尽早发现资源瓶颈并早做规划与准备，应对业务洪峰。

在业务运行期，为保障业务连续性，需要快速发现基础资源、中间件以及业务系统问题，通过平台对多维观测数据进行实时采集、清洗与预警，并通过数据标记与数据串联，实现告警收敛，快速发现问题根源。基于大语言模型的分析算法，对运维平台内所有相关性数据聚合分析与诊断，发现问题根因，基于根因给出解



决方案指导用户恢复系统，并在长期人工辅助介入后，生产智能处置流程自动处理问题并恢复业务运行。

📦 机构 IT 规划建议

随着运维平台化、智能化的不断发展，外部机构需要转换运维建设思路，打破部门、业务产品的隔离墙，站在公司级视角统一思考，构建公司级的统一运维 PaaS 平台。部门、业务产品等按照 PaaS 平台租户的概念申请使用，统一管理，从而构建全局视野的物理资源管理视图、业务运维管理视图、业务运营管理视图；同时可以有效地制定企业级运维标准与数据治理标准，以统一的运维方案运维不同厂商提供的业务系统，同时又能从全局的维度规划与优化资源成本，掌握业务的运行状态，做出有效的 IT 决策。

企业级的基础设施管理平台，将所有计算资源统一管理与池化，以标准化的规格与接口对外提供服务，供企业内部组织使用。通过动态分配与调度策略提升资源的使用率与可用性。同时也为业务提供统一标准的物理资源供给，所有接入业务系统需满足接入标准；并在平台准确高效地维护运维对象关系模型。

企业级业务运维标准建设，基于企业运维 PaaS 平台，构建统一的运维标准，所有厂商业务产品接入需要满足统一的接入标准方可接入上线。对于自带运维解决方案的产品与厂商，其运维系统需要开放从资源规划到线上运维全生命周期的所有运维接口，通过运维能力插件化的方式接入到统一运维 PaaS 平台中以后，方可通过自带运维产品统一运维。

运维数据采集与数据治理标准化，对各业务系统运维过程中产生的运维数据，需要统一格式或经过清洗转换后接入到企业级运维 PaaS 平台中，以支撑统一的



运维分析与决策。

构建智能运维平台，建立智能运维平台，整合先进的人工智能和大数据技术，以提高运维效率和可靠性。集成大语言模型和自动化工具，以处理自然语言文本、日志数据和监控数据，实现自动化故障诊断和解决方案建议。持续的学习与演化，从根本上减轻运维成本与负担，并帮助企业管理者做出经营、管理、成本决策。

持续改进和反馈，智能运维是一个不断演进的领域，机构需要建立持续改进的文化。定期评估智能运维系统的表现，收集用户反馈，根据实际情况调整策略和工具。



版权声明

本白皮书版权属于恒生电子股份有限公司所有并受法律保护，未经恒生电子股份有限公司书面许可，任何单位或个人不得转载、摘编或利用其它方式对本白皮书进行商业使用。如有违反，恒生电子股份有限公司将依法追究相关法律责任。同时，恒生电子股份有限公司确保不存在任何侵犯他人知识产权的情形，如有侵权纠纷，恒生电子股份有限公司负责处理与承担，与联合发布单位无关。