说　　明　　书

DATA PROCESSING METHOD AND PROGRAM MODIFICATION METHOD FOR MIDDLE PLATFORM SYSTEM, SYSTEM, AND STORAGE MEDIUM

一种中台系统的数据处理以及程序修改的方法、系统以及存储介质

技术领域

本说明书涉及数据处理领域，尤其涉及一种中台系统的数据处理以及程序修改的方法、系统以及存储介质。

背景技术

随着信息技术的迅猛发展，为了实现数字资源的共享和复用，中台系统应运而生。中台系统能够为多个不同的租户提供统一的技术平台，从而避免技术的重复开发。随着经济发展，租户的规模和技术复杂度逐渐增加，相关技术中的中台系统架构不清晰，功能交叉，导致一个租户的程序变更会引发其他租户的程序变更，从而引发系统故障。

背景技术部分的内容仅仅是发明人个人所知晓的信息，并不代表上述信息在本公开申请日之前已经进入公共领域，也不代表其可以成为本公开的现有技术。

发明内容

本说明书提供一种更高效的中台系统的数据处理以及程序修改的方法、系统以及存储介质。

第一方面，本说明书提供一种中台系统的数据处理方法，所述中台系统用于向多个租户提供目标服务，所述目标服务包括通用引擎层和个性化功能层，所述通用引擎层被配置为向所述多个租户提供所述目标服务，所述个性化功能层包括与所述多个租户对应

的多个计算模块，一个计算模块被配置为向对应的租户提供所述目标服务，所述计算模块由对应的租户进行配置，其中，所述通用引擎层和所述个性化功能层之间的调用通过接口层实现，所述方法包括：接收目标租户对所述目标服务的目标请求，所述多个租户包括所述目标租户；基于所述目标请求调用所述通用引擎层和所述个性化功能层中与所述目标租户对应的目标计算模块执行所述目标请求，得到目标结果，所述多个计算模块包括所述目标计算模块；以及输出所述目标结果。

第二方面，本说明书提供一种中台系统，所述中台系统用于向多个租户提供目标服务，所述目标服务包括通用引擎层和个性化功能层，所述通用引擎层被配置为向所述多个租户提供所述目标服务，所述个性化功能层包括与所述多个租户对应的多个计算模块，一个计算模块被配置为向对应的租户提供所述目标服务，所述计算模块由对应的租户进行配置，其中，所述通用引擎层和所述个性化功能层之间的调用通过接口层实现，所述中台系统包括：至少一个存储介质，存储有至少一个指令集，用于为多个租户提供目标服务；以及至少一个处理器，同所述至少一个存储介质通信连接，其中，所述中台系统运行时所述至少一个处理器读取所述至少一个指令集，并根据所述至少一个指令集的指示：接收目标租户对所述目标服务的目标请求，所述多个租户包括所述目标租户；基于所述目标请求调用所述通用引擎层和所述个性化功能层中与所述目标租户对应的目标计算模块执行所述目标请求，得到目标结果，所述多个计算模块包括所述目标计算模块；以及输出所述目标结果。

第三方面，本说明书还提供一种中台系统的程序修改方法，所述中台系统用于向多个租户提供目标服务，所述目标服务包括通用引擎层和个性化功能层，所述通用引擎层被配置为向所述多个租户提供所述目标服务，所述个性化功能层包括与所述多个租户对应的多个计算模块，一个计算模块被配置为向对应的租户提供所述目标服务，所述计算模块由对应的租户进行配置，其中，所述通用引擎层和所述个性化功能层之间的调用通过接口层实现，所述方法包括：接收目标租户对所述目标服务的程序的修改请求，所述

多个租户包括所述目标租户；以及基于所述修改请求对所述个性化功能层中与所述目标租户对应的目标计算模块的代码进行修改。

第四方面，本说明书还提供一种中台系统，所述中台系统用于向多个租户提供目标服务，所述目标服务包括通用引擎层和个性化功能层，所述通用引擎层被配置为向所述多个租户提供所述目标服务，所述个性化功能层包括与所述多个租户对应的多个计算模块，一个计算模块被配置为向对应的租户提供所述目标服务，所述计算模块由对应的租户进行配置，其中，所述通用引擎层和所述个性化功能层之间的调用通过接口层实现，所述中台系统包括：至少一个存储介质，存储有至少一个指令集，用于进行程序修改；以及至少一个处理器，同所述至少一个存储介质通信连接，其中，所述中台系统运行时所述至少一个处理器读取所述至少一个指令集，并根据所述至少一个指令集的指示：接收目标租户对所述目标服务的程序的修改请求，所述多个租户包括所述目标租户；以及基于所述修改请求对所述个性化功能层中与所述目标租户对应的目标计算模块的代码进行修改。

第五方面，本说明书还提供一种计算机可读非易失性存储介质，其中，所述计算机可读非易失性存储介质中存储有至少一个指令集，所述至少一个指令集被至少一个处理器执行时，实现如第一方面所述的中台系统的数据处理方法。

第六方面，本说明书还提供一种计算机可读非易失性存储介质，其中，所述计算机可读非易失性存储介质中存储有至少一个指令集，所述至少一个指令集被至少一个处理器执行时，实现如第三方面所述的中台系统的程序修改方法。

由以上技术方案可知，在本说明书提供的中台系统的数据处理以及程序修改的方法、系统以及存储介质中，通用引擎层和个性化功能层共同为目标租户提供目标服务。其中，通用引擎层为所有租户提供目标服务的通用功能，而个性化功能层中的计算模块为对应的租户提供个性化定制功能。不同的租户各自对应有计算模块。本说明书提供的方法、系统以及存储介质中的中台系统中，通用引擎层与个性化功能层之间通过接口层实现调

用，从而实现通用引擎层与个性化功能层的代码隔离。而个性化功能层中的计算模块与租户之间存在一一对应的关系。通用引擎层在调用个性化功能层时会根据目标租户的目标请求调用与目标租户对应的目标计算模块，从而实现了不同租户间的个性化功能层的计算模块之间的隔离。因此，本说明书提供的方法、系统以及存储介质不仅能够实现通用引擎层与个性化功能层的代码隔离，还能够实现个性化功能层中不同租户间的计算模块的代码隔离。因此，当一个租户维护或更新其对应的计算模块的代码时，不会导致通用引擎层的代码变更，也不会导致其他租户的计算模块的代码变更，从而不会引起中台系统的故障，提高了系统的高可用性、维护性和研发效率。

本说明书提供的中台系统的数据处理以及程序修改的方法、系统以及存储介质的其他功能将在以下说明中部分列出。本说明书提供的中台系统的数据处理以及程序修改的方法、系统以及存储介质的创造性方面可以通过实践或使用下面详细示例中所述的方法、装置和组合得到充分解释。

附图说明

为了更清楚地说明本说明书实施例中的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本说明书的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

图 1 示出了根据本说明书的实施例提供的一种中台系统的应用场景示意图；

图 2 示出了根据本说明书的实施例提供的一种计算设备的硬件结构图；

图 3 示出了根据本说明书的实施例提供的一种中台系统的系统架构图；

图 4 示出了根据本说明书的实施例提供的一种中台系统的个性化功能层的架构图；

图 5 示出了根据本说明书的实施例提供的一种中台系统的配置管理后台的界面示意图；

图 6 示出了根据本说明书的实施例提供的一种中台系统的数据处理的方法流程图；

图 7 示出了根据本说明书的实施例提供的一种中台系统的执行目标请求的方法流程图；以及

图 8 示出了根据本说明书的实施例提供的一种中台系统的程序修改的方法流程图。

具体实施方式

以下描述提供了本说明书的特定应用场景和要求，目的是使本领域技术人员能够制造和使用本说明书中的内容。对于本领域技术人员来说，对所公开的实施例的各种局部修改是显而易见的，并且在不脱离本说明书的精神和范围的情况下，可以将这里定义的一般原理应用于其他实施例和应用。因此，本说明书不限于所示的实施例，而是与权利要求一致的最宽范围。

这里使用的术语仅用于描述特定示例实施例的目的，而不是限制性的。比如，除非上下文另有明确说明，这里所使用的，单数形式“一”，“一个”和“该”也可以包括复数形式。当在本说明书中使用时，术语“包括”、“包含”和/或“含有”意思是指所关联的整数，步骤、操作、元素和/或组件存在，但不排除一个或多个其他特征、整数、步骤、操作、元素、组件和/或组的存在或在该系统/方法中可以添加其他特征、整数、步骤、操作、元素、组件和/或组。

考虑到以下描述，本说明书的这些特征和其他特征、以及结构的相关元件的操作和功能、以及部件的组合和制造的经济性可以得到明显提高。参考附图，所有这些形成本说明书的一部分。然而，应该清楚地理解，附图仅用于说明和描述的目的，并不旨在限制本说明书的范围。还应理解，附图未按比例绘制。

本说明书中使用的流程图示出了根据本说明书中的一些实施例的系统实现的操作。应该清楚地理解，流程图的操作可以不按顺序实现。相反，操作可以以反转顺序或同时实现。此外，可以向流程图添加一个或多个其他操作。可以从流程图中移除一个或多个

操作。

为了方便描述，首先对本说明书后文中将会出现的术语进行解释。

中台系统：也可以叫做多租户系统，通过共享和复用资源以降低成本。中台系统是在前台和后台之间演化而来的产物，旨在解决前台项目和后台技术之间的矛盾，实现前台项目和后台技术的高效协同。中台系统的核心在于将后台系统中通用的模块或服务抽象出来，形成一个独立的服务层，为前台提供快速响应和高效支持。这样，当中台系统发生变化时，前台的应用可以快速适应，而不需要对每个前台应用进行大量的修改。中台系统是前台作战单元和后台资源部门之间的组织模块，整合了企业的核心能力和资源以支持前台事务的快速发展和创新，实现后台资源的最大化利用和优化管理。

通用引擎层：指中台系统中可以跨租户复用的技术组件或服务，为不同的租户提供通用的功能支持，这些功能可能包括但不限于用户身份验证、数据存储、消息传递、支付处理等。实现不同功能或服务的中台系统所提供的通用引擎层的功能也可能是不同的。

个性化功能层：指在中台系统中能够根据不同租户需求进行定制化开发的服务或组件。个性化功能层是租户独有的功能，不同租户对应的个性化功能层是不同的。

计算模块：指个性化功能层中根据不同租户需求定制的、可运作的计算单元，不同的租户对应不同的计算模块，以实现个性化功能定制。计算模块可以是软件层面的算法、数据处理流程。

接口层：指定义系统内部各层或组件之间以及系统与外部应用之间的交互协议。

租户：指在中台系统中的客户或组织。

目标服务：指中台系统中为了满足特定目标或需求而设计的服务。

隔离：包括物理隔离和逻辑隔离。物理隔离指的是采用物理方法将内部设备、网络与外部公共网络进行隔离；逻辑隔离指的是在计算机系统或网络中通过软件或配置实现的一种隔离机制，它通过逻辑上分离不同的资源、服务或用户,以确保它们之间的相互影

响最小化。

SPI 拓展：SPI 全称为 Service Provider Interface，是一种服务发现机制；它用来给第三方实现或扩展的 API，它可以用于实现框架扩展或组件替换；SPI 机制本质是将接口实现类的全限定名配置在文件中，并由服务加载器读取配置文件，加载文件中的实现类，这样运行时可以动态的为接口替换实现类。

应用程序编程接口（Application Programming Interface，简称：API），是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。

树状路由：按照树的结构进行条件路由。

动态脚本：groovy 提供的动态脚本编译，以实现不同的功能。

下面对本说明书的应用场景进行介绍。

本说明书提供的中台系统的数据处理以及程序修改的方法、系统以及存储介质可以应用于中台系统的软件架构中。中台系统可以应用在多种场景下，比如中台系统可以是风控场景中提供风控服务的风控系统，也可以是支付场景下提供支付服务的支付系统，还可以是数据管理场景下提供数据管理服务的管理系统等。

以中台系统是风控系统为例。中台系统可以为不同的租户提供风控服务。中台系统为租户提供的风控服务既可以包括通用的风控计算模型，也可以包括个性化定制模型。通用的风控计算模型可以为所有租户使用，以实现资源共享和复用，降低成本。而个性化定制模型则可以根据不同租户的需求进行个性化定制，以满足不同租户的使用。比如，银行或者金融机构作为租户，可以利用中台系统提供的风控服务来评估和管理贷款、信用卡等金融产品的信用风险。每个机构可能有特定的风险评估模型和合规要求，中台系统可以为每个租户定制化这些服务。电商平台作为租户，需要对用户交易行为进行风险控制，防止欺诈和滥用行为。此时中台系统可以提供实时监测和分析服务，帮助平台识

别异常交易模式，并采取相应措施。保险公司作为租户，需要利用中台系统进行保险欺诈检测。通过集成保险索赔数据和市场数据，中台可以帮助保险公司评估索赔的真实性，并减少欺诈损失。

以中台系统是支付系统为例。中台系统可以为不同的租户提供支付服务。中台系统为租户提供的支付服务既可以包括通用的支付计算模型，也可以包括个性化定制模型。通用的支付计算模型可以为所有租户使用，以实现资源共享和复用，降低成本。而个性化定制模型则可以根据不同租户的需求进行个性化定制，以满足不同租户的使用。比如，在线零售商作为租户，需要一个稳定且安全的支付系统来处理顾客的支付事务。中台系统可以提供定制化的支付网关服务，支持多种支付方式，并确保交易的安全性。移动应用开发商作为租户，可能需要集成内购或订阅服务的支付功能。中台系统可以为这些开发商提供灵活的支付解决方案，包括不同地区的支付偏好和法规遵从。在线教育或医疗咨询公司作为租户，需要处理用户的预付款、周期性付款等。中台支付系统可以为这些企业提供定制化的账单管理和资金流转服务。

需要说明的是，上述场景只是本说明书提供的多个使用场景中的几个，本说明书提供的中台系统的数据处理以及程序修改的方法、系统以及存储介质不仅可以应用于风控和支付场景，还可以应用于需要中台系统的所有场景，比如，供应商管理场景、营销管理场景等等。本领域技术人员应当明白，当本说明书提供的中台系统的数据处理以及程序修改的方法、系统以及存储介质应用于其他使用场景时，其实现方式及技术效果是类似的。

图 1 示出了根据本说明书的实施例提供的一种中台系统的应用场景示意图。如图 1 所示，该应用场景 100 可以包括租户 110、客户端 120、网络 130 以及中台系统 140。

中台系统 140 可以是执行目标服务的设备或者服务器。当中台系统 140 是风控系统时，目标服务可以是风控服务。当中台系统 140 是支付系统时，目标服务可以是支付服务。中台系统 140 可以存储有执行本说明书描述的数据处理的方法的数据或指令，并可

以执行或用于执行所述数据和/或指令。在一些实施例中，中台系统 140 可以存储有执行本说明书描述的程序修改的方法的数据或指令，并可以执行或用于执行所述数据和/或指令。在一些实施例中，中台系统 140 可以包括具有数据信息处理功能的硬件设备和驱动该硬件设备工作所需必要的程序。

租户 110 可以为中台系统 140 的客户，即需要使用中台系统 140 提供的目标服务的用户。中台系统 140 可以为多个租户 110 提供目标服务。租户 110 可以是企业，也可以是自然人。在一些实施例中，租户 110 可以是电子商务平台、移动支付服务提供商、银行或其他金融服务机构等等。

客户端 120 可以为响应于租户 110 和中台系统 140 的中间设备。客户端 120 可以是租户 110 与中台系统 140 进行数据交互的设备。客户端 120 可以与中台系统 140 通信连接。中台系统 140 客户端 120 不仅是用户接触和操作系统的第一界面，也是展示中台数据处理结果、提供用户操作入口的直接平台。在一些实施例中，客户端可以包括移动设备、平板电脑、 笔记本电脑、机动车辆的内置设备或类似内容，或其任意组合。在一些实施例中，移动设备可包括智能家居设备、智能移动设备或类似设备，或其任意组合。在一些实施例中，所述智能家居装置可包括智能电视、台式电脑等，或任意组合。在一些实施例中，所述智能移动设备可包括智能手机、个人数字辅助等，或其任意组合。在一些实施例中，客户端 120 可以包括具有数据信息处理功能的硬件设备和驱动该硬件设备工作所需必要的程序。在一些实施例中，客户端 120 可以是具有定位技术的设备，用于定位客户端 120 的位置。中台系统 140 可以与多个客户端 120 通信连接，并接收多个客户端 120 发送的数据。

网络 130 可以促进信息和/或数据的交换。如图 1 所示，客户端 120 以及中台系统 140 可以同网络 130 连接，并且通过网络 130 互相传输信息和/或数据。例如，中台系统 140 可以通过网络 130 向客户端 120 采集所想要的数据。在一些实施例中，网络 130 可以是任何类型的有线或无线网络，也可以是其组合。比如，网络 130 可以包括电缆网络，

有线网络、光纤网络、电信通信网络、内联网、互联网、局域网(LAN)、广域网(WAN)、无线局域网(WLAN)、大都市市区网(MAN)、广域网(WAN)、公用电话交换网(PSTN)、蓝牙网络、ZigBee 网络、近场通信(NFC)网络或类似网络。在一些实施例中，网络 130 可以包括一个或多个网络接入点。例如，网络 130 可以包括有线或无线网络接入点，如基站和/或互联网交换点，通过该接入点，客户端 120 以及中台系统 140 的一个或多个组件可以连接到网络 130 以交换数据和/或信息。

应该理解，图 1 中的客户端 120、网络 130 以及中台系统 140 的数目仅仅是示意性的。根据实现需要，可以具有任意数目的客户端 120、网络 130 以及中台系统 140。

需要说明的是，本说明书获取到的用户数据均经过用户授权，不涉及用户隐私。

图 2 示出了根据本说明书的实施例提供的一种计算设备 200 的硬件结构图。计算设备 200 可以作为图 1 中的中台系统 140 的执行设备，执行本说明书描述的中台系统的数据处理和/或程序修改的方法。

如图 2 所示，计算设备 200 可以包括至少一个存储介质 230 和至少一个处理器 220。在一些实施例中，计算设备 200 还可以包括通信端口 250 和内部通信总线 210。计算设备 200 还可以包括 I/O 组件 260。

内部通信总线 210 可以连接不同的系统组件。例如，内部通信总线 210 可以连接存储介质 230、处理器 220、通信端口 250 和 I/O 组件 260 等。

I/O 组件 260 支持计算设备 200 和其他组件之间的输入/输出。

通信端口 250 用于计算设备 200 同外界的数据通信。比如，通信端口 250 可以用于计算设备 200 同网络 140 之间的数据通信。通信端口 250 可以是有线通信端口也可以是无线通信端口。

存储介质 230 可以包括数据存储装置。所述数据存储装置可以是非暂时性存储介质，也可以是暂时性存储介质。比如，所述数据存储装置可以包括磁盘 232、只读存储介质

（ROM）234 或随机存取存储介质（RAM）236 中的一种或多种。存储介质 230 还包括存储在所述数据存储装置中的至少一个指令集。所述指令集可以包括计算机程序代码，所述计算机程序代码可以包括程序、例程、对象、组件、数据结构、过程、模块等等。

至少一个处理器 220 可以同至少一个存储介质 230 通信连接。当计算设备 200 运行时，至少一个处理器 220 读取所述至少一个指令集，并且根据所述至少一个指令集的指示，执行本说明书提供的中台系统的数据处理和/或程序修改的方法。处理器 220 可以执行中台系统的数据处理和/或程序修改的方法包含的步骤。处理器 220 可以是一个或多个处理器的形式，在一些实施例中，处理器 220 可以包括一个或多个硬件处理器，例如微控制器，微处理器，精简指令集计算机（RISC），专用集成电路（ASIC），特定于应用的指令集处理器（ASIP），中心处理单元（CPU），图形处理单元（GPU），物理处理单元（PPU），微控制器单元，数字信号处理器（DSP），现场可编程门阵列（FPGA），高级 RISC 机器（ARM），可编程逻辑器件（PLD），能够执行一个或多个功能的任何电路或处理器等，或其任何组合。

仅仅为了说明问题，附图中计算设备 200 仅示出了一个处理器 220。然而，应当注意，本说明书中计算设备 200 还可以包括多个处理器，因此，本说明书中披露的操作和/或方法步骤可以由一个处理器执行，也可以由多个处理器联合执行。例如，如果在本说明书中描述计算设备 200 的处理器 220 执行步骤 A 和步骤 B，则应该理解，步骤 A 和步骤 B 也可以由两个不同处理器 220 联合或分开执行（例如，第一处理器执行步骤 A，第二处理器执行步骤 B，或者第一和第二处理器共同执行步骤 A 和 B）。

图 3 示出了根据本说明书的实施例提供的一种中台系统 140 的系统架构图。如图 3 所示，中台系统 140 用于向多个租户 110 提供目标服务。目标服务包括通用引擎层 141 和个性化功能层 143。通用引擎层 141 被配置为向多个租户 110 提供目标服务。个性化功能层 143 包括与所述多个租户 110 对应的多个计算模块。一个计算模块被配置为向对应的租户 110 提供所述目标服务的个性化功能。计算模块由对应的租户 110 进行配置。

通用引擎层 141 和个性化功能层 143 之间的调用通过接口层 145 实现。

通用引擎层141可以针对所有租户110，为所有租户110提供目标服务的通用功能。所谓的通用功能可以理解为不同租户 110 在实现目标服务过程中所使用的相同的技术、模型或数据，等等。中台系统 140 通过设置通用引擎层 141 实现为不同租户 110 提供目标服务过程中的技术共享和复用，从而在提高计算效率的同时解决计算资源。当中台系统 140 是风控系统时，通用引擎层 141 提供的通用功能可以是属性转换、引擎分析（多个风险域）、决策合并、后置任务，等等。属性转换模块能将原始数据转换为适合风险评估的格式。引擎分析负责使用规则引擎、模型和其他分析工具来评估交易或用户行为的风险等级。决策合并模块负责整合多个风控策略或模型的输出，形成一个统一的风控决策,可以为确定最终的风险评分或决策，如批准、拒绝或进一步审查。后置任务模块负责处理风控决策后的活动，如记录风控事件、更新用户风险档案、通知相关系统采取行动等。通用引擎层 141 负责通用功能的串联和编排。图 3 中的通用引擎层 141 提供的通用功能只是示例性说明，不同的中台系统 140 可以提供不同的通用功能，本说明书对此不做限定。

个性化功能层 143 用于向不同的租户 110 提供个性化的定制功能。在使用目标服务时，不同的租户 110 可能存在不同的需求。比如，当中台系统 140 是风控系统时，不同的租户 110 可能面对不同的合规需求、风险类型、风险等级，等等。因此，为了提高风控计算的准确性，面对不同租户 110 的不同需求，中台系统 140 设置了个性化功能层 140 以满足不同租户 110 的需求。为了实现不同租户 110 的个性化功能的配置，个性化功能层 143 由对应的租户 110 进行配置。租户 110 配置的属于自己的个性化功能层为计算模块。为了方便描述，图 3 中示出了 2 个计算模块，分别标记为计算模块 143a 和计算模块 143b。其中，计算模块 143a 与租户 110a 对应，为租户 110a 提供目标服务的个性化功能，并由租户 110a 维护和配置。计算模块 143b 与租户 110b 对应，为租户 110b 提供目标服务的个性化功能，并由租户 110b 维护和配置。一个计算模块中可以根据不同的

个性化定制需求，具有不同的子计算模块。子计算模块可以根据租户 110 自身的需求进行维护、配置、更改、甚至是删减。为了方便描述，我们将子计算模块定义为扩展点。

其中，通用引擎层 141 和个性化功能层 143 相互配合以实现完整的目标服务。通用引擎层 141 可以为目标服务的实现提供基本的实现框架，个性化功能层 143 则为通用引擎层 141 的计算提供个性化的数据以及实现方式。即一个租户 110 在使用目标服务时，目标服务的完整功能需要通用引擎层 141 和个性化功能层 143 中与租户 110 对应的计算模块共同参与计算完成。而在完成目标服务过程中，通用引擎层 141 需要调用个性化功能层 143，调用过程通过接口层 145 实现。接口层 145 可以是应用程序编辑接口（API 接口）。API 是应用程序在以编程方式与其他应用程序交互时需要遵循的一组规则。API 代表应用程序编程接口。在 API 上下文中，“应用程序”一词指的是具有独特功能的任何软件。接口可以被认为是两个应用程序之间的服务契约。该合约定义了两者如何使用请求和响应相互通信。在一些实施例中，接口层 145 可以是 SPI 拓展接口。SPI 扩展接口可以将接口实现类的全限定名配置在文件中，并由服务加载器读取配置文件，加载文件中的实现类，这样运行时可以动态的为接口替换实现类。也就是说，SPI 扩展接口可以将计算模块的全限定名配置在文件中。当通用引擎层 141 调用计算模块时，可以根据发起目标服务的目标租户对应的计算模块的全限定名调用对应的计算模块来实现目标服务。也就是说，SPI 扩展接口在调用计算模块时，是根据计算模块对应的全限定名来调用的。不同的计算模块具有不同的全限定名。如前所述，租户 110 与计算模块相对应。即租户 110 与计算模块的全限定名相对应。中台系统 140 中可以预先存储有租户 110 与计算模块的全限定名的映射关系。中台系统 140 可以根据目标租户，确定目标租户对应的计算模块的全限定名，进而调用对应的计算模块的代码进行应用的实现。当计算模块的代码进行变更时，只要计算模块的全限定名和租户 110 的映射关系不变更，就不会出现调用错误的情况。并且，SPI 扩展接口可以实现对计算模块的动态替换和扩展。当新增计算模块时，只需将新增的扩展模块的全限定名与对应的租户的映射关系保存即可快

速实现计算模块的扩展。

接口层 145 可以连接通用引擎层 141 和个性化功能层 143，从而实现通用引擎层 141 的通用功能和个性化功能层 143 的个性化功能的隔离。通常通用功能的变更是低频的，而个性化功能变更是高频的。因此，中台系统 140 将通用引擎层 141 和个性化功能层 143 隔离后，租户 110 在修改由自己配置的个性化功能层 143 的计算模块的代码时，不会修改通用引擎层 141 的代码，也就不会有误修改通用引擎层 141 导致的系统故障的问题。

同时，个性化功能层 143 中的计算模块与租户 110 之间存在一一对应的关系，从而实现不同租户 110 的计算模块的隔离，即不同租户的个性化功能也是隔离的。具体的隔离方式可以是计算模块的全限定名与租户 110 之间的租户标识一一对应，且不同租户 110 的租户标识是唯一的。不同的租户 110 具有不同的租户标识。租户标识可以用于标识租户 110 的身份。租户标识可以是基于特定的编码规则生成的字符串。在一些实施例中，租户标识与租户 110 所使用或绑定的客户端 120 的设备标识有关。比如，租户标识可以是基于客户端 120 的设备标识生成的。再比如，租户标识可以与客户端 120 的设备标识相关联。在通用引擎层 141 对个性化功能层 143 进行调用时，可以根据租户 110 的租户标识调用于其对应的计算模块。这样的设置可以使得租户 110 间的计算模块相互隔离。一个租户 110 的计算模块的变更，不会导致其他租户 110 的计算模块的变更，因此不会导致目标服务的故障。

图 4 示出了根据本说明书的实施例提供的一种中台系统 140 的个性化功能层 143 的架构图。如前所述，接口层 145 为 SPI 扩展接口。个性化功能层 143 中的计算模块中的各个扩展点的代码可以通过软件包的形式进行管理。在租户配置对应的计算模块时，其配置的扩展点的代码的软件包可以加载到模块拓展包仓储 147 中进行存储。当中台系统 140 启动时，多个租户 110 各自对应的扩展点的代码的软件包可以加载至模块拓展包仓储 147 中。如前所述，多个租户 110 与多个计算模块相对应。具体地，多个租户 110 与

多个计算模块一一对应。多个租户 110 与多个计算模块的对应关系可以通过租户 110 的柱标识与计算模块的全限定名的映射关系实现。中台系统 140 中可以预先存储有租户 110 的租户标识与计算模块的全限定名的映射关系。租户 110 的租户标识与计算模块的全限定名的映射关系的存储方式可以包括多种，比如列表形式、键值对形式，等等。在一些实施例中，所述计算模块与对应的租户标识通过键值对的方式存储。即计算模块的全限定与名租户 110 的租户标识的映射关系通过键值对（key-value）的方式进行存储。比如，租户 110a 的租户标识=》计算模块 143a 的全限定名。租户 110b 的租户标识=》计算模块 143b 的全限定名。键值对的方式简单易用，具有高性能、可扩展性和灵活性等优点。当数据量较大时，键值对的方式在查找和检索时更快速、更高效。

在一些实施例中，中台系统 140 中还可以存储有计算模块的全限定名与存储路径的映射关系。具体地，计算模块的全限定名与存储路径的映射关系可以存储在扩展点路由器 148 中。计算模块的全限定名与存储路径的映射关系的存储方式可以包括多种，比如列表形式、键值对形式，等等。中台系统 140 可以根据租户 110 的租户标识，确定对应的计算模块的全限定名，进而确定对应的计算模块的存储路径，从而在模块拓展包仓储 147 中对应的存储路径中找到计算模块的代码的软件包以实现应用的功能。

如前所述，一个计算模块中可以包括多个子模块（也称为拓展点）。不同的子模块对应不同的个性化场景。即一个计算模块可以包括多个场景下的子模块，以满足不同场景的个性化需求。而针对不同场景，需求又是不一样的，功能实现也就会不一样。为了应对不同场景的个性化功能，一个场景还可以进一步划分为更深层次的子场景。中台系统 140 可以通过一种树状路由+动态脚本的实现方式来实现对场景的划分。所述多个场景下的子模块根据对应的租户预设的分类规则进行分类。中台系统 140 中可以预设标准的分类规则，不同的租户也可以自定义分类规则。不同的租户可以设置不同的分类规则。分类规则是租户预先配置好的。

树状路由即按照租户自定义的条件进行路由，当树的分支路径都满足后，最终会路

由到一个叶子结点上。考虑到需要将同一个租户的目标服务在不同场景之间变更的相互隔离，因此将树状路由分为标准路由和自定义路由。

标准路由：它的顶层结点为目标服务中的通用引擎层 141 的通用功能（比如，属性转换、引擎分析、决策合并、后置任务，等），子结点为该通用功能下不同的场景。

自定义路由：按照租户 110 需要设置路由条件。

租户的管理员或用户可以在配置管理后台对树状路由进行管理。

子模块的代码通过脚本实现，可以使用包括且不限于 Python、JavaScript、Java 等编程语言实现脚本。关于子模块的脚本，租户的管理员或者用户可以在配置管理后台进行变更修改，可以在配置管理后台中动态地添加、删除、修改或发布子模块的脚本。配置管理后台允许管理员通过可视化的界面对子模块的脚本进行管理。一旦脚本在后台被更新，中台系统 140 可以实时加载新的脚本，实现功能的即时变更。此外，系统还可以会记录详细的日志信息，并进行监测，以便于问题排查和性能优化。

图 5 示出了根据本说明书的实施例提供的一种中台系统的配置管理后台的界面图。图 5 只是示例性地提供了一种配置管理后台的界面图。本领域技术人员应当明白，其他配置管理后台的界面图也在本说明书的保护范围内。如图 5 所示，例如，对于租户 110a 中的属性转换计算模块来说，风控识别可以分为注册时的风险识别、登录时的风险识别、支付时的风险识别以及退款时的风险识别，等等。此时，注册、登录、支付、退款可以作为风险识别的多个场景，而注册时的风险识别、登录时的风险识别、支付时的风险识别以及退款时的风险识别则可以作为属性转换计算模块下的多个子模块。而在每个场景下的子模块又可以进一步进行划分。比如在支付场景中，又可以进一步根据国家进行划分，比如，A 国、B 国、C 国；确定好国家之后，又可以进一步根据用户状态进行划分，比如新用户和老用户；在确定用户状态之后，则可以确定最终场景下对应的计算代码。当然，租户 110 也可以配置其他的场景划分规则，比如还可以根据账龄、卡龄和/或支付方式等进行子场景的划分。

综上可知，租户 110 对场景的划分可以通过树状路由的方式实现，通过不同的层级进行划分，从而将一个计算模块的代码根据不同的场景，划分为多个场景或者多个子场景下的代码。同一场景可以同时根据不同的分类规则进行划分。而场景划分的层级的数量则可以根据不同租户 110 的需求自行进行配置。

需要说明的是，根据场景的树状路由，在树状路由的叶节点（即划分的最后一个层级）上的子模块的代码通过脚本实现。树状路由的叶节点关联了子模块的代码的 groovy 脚本。叶结点关联的脚本，其可在配置管理后台进行变更修改，通过在后台的增、删、改，实现动态、灵活的功能的个性化实现。

综上可知，在中台系统 140 中，对同一个计算模块的不同场景进行隔离，通过树状路由，使得同一个计算模块的不同场景个性化功能变更隔离；同时依托 groovy 脚本动态更新技术，实现个性化功能的动态拓展。通过对不同场景进行隔离，使得不同场景下的代码变更相互独立，一个场景的变更不会影响其他场景的变更，使得代码的实现更简单、高效。同时，通过动态脚本的方式来实现脚本动态编译、动态加载技术，实现个性化功能的动态修改，大大提高了代码的研发效率和修改效率。同时，通过树状路由和动态脚本，将个性化功能搬迁到页面上，以能够可视化地呈现出来，后续代码梳理时，梳理成本可大大降低。

图 6 示出了根据本说明书的实施例提供的一种中台系统的数据处理的方法 P300 的流程图。如前，计算设备 200 可以执行本说明书的中台系统的数据处理的方法 P300。具体地，计算设备 200 中的处理器 220 可以读取存储在其本地存储介质中的指令集，然后根据指令集的规定，执行本说明书的中台系统的数据处理的方法 P300。如图 6 所示，以风控系统为例，所述方法 P300 可以包括 S310-S330：

S310：目标租户通过客户端 120 向中台系统 140 发送对目标服务的目标请求，中台系统 140 统接收目标租户对所述目标服务的目标请求。

所述多个租户 110 包括所述目标租户。即目标租户可以是前述的中台系统 140 的多个租户 110 中的一个。比如，目标租户是正在通过中台系统 140 进行风控识别的租户。客户端 120 可以是目标租户所使用的与中台系统 140 进行交互的终端设备。所述目标请求可以是目标租户请求中台系统 140 进行风控计算的请求。在一些实施例中，所述目标请求携带有上述目标租户的目标租户标识。

S330：中台系统 140 调用通用引擎层 141 和个性化功能层 143 中与目标租户对应的目标计算模块执行所述目标请求，得到目标结果。

目标计算模块可以是多个租户 110 对应的多个计算模块中与目标租户对应的计算模块。多个计算模块包括目标计算模块。在步骤 S330 中，中台系统 140 可以通过目标租户的目标租户标识确定目标计算模块的全限定名，从而调用通用引擎层 141 和目标计算模块执行所述目标请求。

S350：中台系统 140 输出所述目标结果。

目标结果可以是中台系统 140 根据目标请求执行目标服务所得到的结果。比如，当中台系统 140 是风控系统时，目标结果可以是风控识别的结果。中台系统 140 可以将所述目标结果输出至目标租户的客户端 120。客户端 120 可以将目标结果展示给目标租户。

图 7 示出了根据本说明书的实施例提供的一种中台系统 140 的执行目标请求的方法流程图。图 7 对应步骤 S330。如图 7 所示，步骤 S330 可以包括：

S331：基于目标租户标识识别出目标计算模块；

S333：基于目标请求调用通用引擎层 141 执行所述目标服务的通用功能；

S335：通过接口层 145，调用所述目标计算模块执行所述目标服务中与所述目标租户对应的功能；以及

S337：得到所述目标结果。

在一些实施例中，中台系统 140 还可以包括识别层 149。识别层 149 用于接收目标请求，并基于目标请求中的目标租户标识识别出对应的目标计算模块。步骤 S331 是指

中台系统 140 的识别层 149 可以根据租户 110 的租户标识与计算模块的全限定名的映射关系，基于目标请求中携带的目标租户标识，识别出与目标租户对应的目标计算模块的全限定名。中台系统 140 可以将目标计算模块的全限定名添加至目标服务的调用链路中。在执行目标服务的过程中，通用引擎层 141 每一次对个性化功能层 145 的调用都会携带目标计算模块的全限定名。这样的设置，可以使得目标计算模块的全限定名伴随整个目标服务的执行过程，从而确保每一次都能准确调用。

在一些实施例中，中台系统 140 还可以根据计算模块的全限定名与存储路径的映射关系，识别出目标计算模块的存储路径，以实现对目标计算模块的准确调用，确保不同租户的个性化定制功能的实现不出现交叉，从而确保目标服务的准确执行。

如前所述，计算模块可以根据场景划分为多个子模块。而场景也可以进一步划分为子场景，从而将子模块进一步划分。此时，所述目标计算模块可以包括多个场景下的子模块。目标请求中还可以包括目标场景对应的目标场景标识。目标场景可以是多个场景中的一个，也可以是多个场景所包含的子场景中的一个，还可以是树状路由中的叶节点的场景中的一个。在一些实施例中，目标场景标识可以包括树状路由中整个链路中的每个节点的场景标识。

此时，步骤 S335 可以包括：通过所述接口层，调用所述目标计算模块中与所述目标场景标识对应的子模块执行所述目标服务中与所述目标租户的所述目标场景对应的功能。此时，所述目标计算模块中与所述目标场景标识对应的子模块可以是树状路由中符合条件的叶节点对应的子模块。具体地，中台系统 140 可以将目标场景标识与树状路由中的节点的标识进行匹配，从而确定目标场景标识对应的叶节点，从而调用叶节点对应的子模块执行对应的功能。

综上所述，本说明书提供的中台系统 140 中的通用引擎层 141 和个性化功能层 143 之间通过接口层 145 实现隔离，从而使得通用引擎层 141 的代码和个性化功能层 143 的代码实现隔离。这样的设置使得个性化功能层 143 的代码变更时，不会引起通用引擎层

141 的代码变更，避免中台系统 140 的故障，从而在保证了中台系统 140 的通用引擎层 141 的高复用性的基础上，实现可不同租户 110 的个性化定制，同时确保了中台系统 140 的稳定性。同时，个性化功能层 143 中的不同租户 110 之间通过租户标识与计算模块代码的全限定名的映射关系，实现了不同租户 110 的代码的隔离。接口层 145 在调用计算模块时，根据租户标识调用对应的计算模块的代码，从而避免代码调用错误，确保中台系统 140 的准确性。同时，由于不同租户 110 的代码隔离，使得一个租户的代码变更，不会导致其他租户的代码变更，进一步确保了中台系统 140 的稳定性，降低故障率。并且，一个租户 110 的计算模块可以根据不同的需求通过树状路由的方式进行精细的场景划分，并在树状路由的叶节点通过动态脚本的方式进行代码的编译。这样的设置使得一个计算模块划分为多个场景下的子模块，在进行个性化功能的变更时，能够根据树状路由快速定位至变更的叶节点，同时动态脚本的编译方式可以实时实现代码变更，大大提高了研发效率。

图 8 示出了根据本说明书的实施例提供的一种中台系统的程序修改的方法 P400 的流程图。如前，计算设备 200 可以执行本说明书的中台系统的程序修改的方法 P400。具体地，计算设备 200 中的处理器 220 可以读取存储在其本地存储介质中的指令集，然后根据指令集的规定，执行本说明书的中台系统的程序修改的方法 P400。如图 8 所示，方法 P400 包括：

S410：接收目标租户对目标服务的程序的修改请求；

S430：基于所述修改请求对所述个性化功能层 146 中与所述目标租户对应的目标计算模块的代码进行修改。

在步骤 S410 中，与中台系统的数据处理的方法 P300 类似，不同的是，在方法 P300 中，中台系统 140 接收的是目标服务的目标请求，而在方法 P400 中，中台系统 140 接收的是目标服务的程序的修改请求。所述修改请求中与目标请求类似携带有所述目标租户的目标租户标识。所述修改请求中还携带有修改后的脚本代码。

在步骤 S430 中，中台系统 140 基于修改请求对所述个性化功能层 143 中与所述目标租户对应的目标计算模块的代码进行修改。具体地，步骤 S430 可以包括：

S431：基于目标租户标识识别出所述目标计算模块。在此处，与方法 P300 中的 S331 类似。中台系统 140 的识别层 149 用于接收修改请求，并基于修改请求中的目标租户标识识别出对应的目标计算模块。步骤 S431 是指中台系统 140 的识别层 149 可以根据租户 110 的租户标识与计算模块的全限定名的映射关系，基于修改请求中携带的目标租户标识，识别出与目标租户对应的目标计算模块的全限定名。在一些实施例中，中台系统 140 还可以根据计算模块的全限定名与存储路径的映射关系，识别出目标计算模块的存储路径，以实现对目标计算模块的准确定位，确保对目标计算模块的代码修改准确，不会对其他租户的计算模块的代码进行修改。

S433：对目标计算模块的代码进行修改。如前所述，目标计算模块包括多个场景下的子模块。子模块的代码通过脚本实现，可以使用包括且不限于 Python、JavaScript、Java 等编程语言实现脚本。关于子模块的脚本，管理员或者用户可以在配置管理后台进行变更修改，可以在配置管理后台中动态地添加、删除或修改子模块的脚本。配置管理后台允许管理员通过可视化的界面对子模块的脚本进行管理。一旦脚本在后台被更新，中台系统 140 可以实时加载新的脚本，实现功能的即时变更。此外，系统还可以会记录详细的日志信息，并进行监测，以便于问题排查和性能优化。

在一些实施例中，修改请求中还可以携带有待修改场景标识。待修改场景标识可以是管理员通过可视化的界面选定的需要修改的叶节点的子场景对应的标识，也可以是叶节点的子场景所在的链路中的所有节点的场景标识的集合。中台系统 140 可以根据待修改场景标识，确定目标计算模块中需要进行修改的子模块。比如，中台系统 140 可以根据树状路由，对待修改场景标识对应的叶节点进行定位，从而确定叶节点上的脚本，并对脚本进行修改。

通过方法 P300 和方法 P400，中台系统 140 不仅可以实现数字资源的共享和复用，

而且当一个租户维护或更新其计算模块的代码时，不会影响通用引擎层的代码，也不会导致其他租户的计算模块发生变化，避免了中台系统 140 的故障，从而提高了系统的高可用性、可维护性和研发效率。

本说明书另一方面提供一种计算机可读非暂时性存储介质，存储有至少一个指令集，用来进行中台系统的数据处理和/或程序修改。当所述至少一个指令集被处理器执行时，所述至少一个指令集指导所述处理器实施本说明书所述中台系统的数据处理的方法 P300 和/或程序修改的方法 P400 的步骤。在一些可能的实施方式中，本说明书的各个方面还可以实现为一种程序产品的形式，其包括程序代码。当所述程序产品在计算设备 200 上运行时，所述程序代码用于使计算设备 200 执行本说明书描述的中台系统的数据处理的方法 P300 和/或程序修改的方法 P400 的步骤。用于实现上述方法的程序产品可以采用便携式紧凑盘只读存储器（CD-ROM）包括程序代码，并可以在计算设备 200 上运行。然而，本说明书的程序产品不限于此，在本说明书中，可读存储介质可以是任何包含或存储程序的有形介质，该程序可以被指令执行系统使用或者与其结合使用。所述程序产品可以采用一个或多个可读介质的任意组合。可读介质可以是可读信号介质或者可读存储介质。可读存储介质例如可以为但不限于电、磁、光、电磁、红外线、或半导体的系统、装置或器件，或者任意以上的组合。可读存储介质的更具体的例子包括：便携式盘、硬盘、随机存取存储器（RAM）、只读存储器（ROM）、可擦式可编程只读存储器（EPROM 或闪存）、光纤、便携式紧凑盘只读存储器（CD-ROM）、光存储器件、磁存储器件、或者上述的任意合适的组合。所述计算机可读存储介质可以包括在基带中作为载波一部分传播的数据信号，其中承载了可读程序代码。这种传播的数据信号可以采用多种形式，包括但不限于电磁信号、光信号或上述的任意合适的组合。可读存储介质还可以是可读存储介质以外的任何可读介质，该可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。可读存储介质上包含的程序代码可以用任何适当的介质传输，包括但不限于无线、有线、光缆、RF 等等，或者上述的任意合

适的组合。可以以一种或多种程序设计语言的任意组合来编写用于执行本说明书操作的程序代码，所述程序设计语言包括面向对象的程序设计语言—诸如 Java、C++等，还包括常规的过程式程序设计语言—诸如“C”语言或类似的程序设计语言。程序代码可以完全地在计算设备 200 上执行、部分地在计算设备 200 上执行、作为一个独立的软件包执行、部分在计算设备 200 上部分在远程计算设备上执行、或者完全在远程计算设备上执行。

上述对本说明书特定实施例进行了描述。其他实施例在所附权利要求书的范围内。在一些情况下，在权利要求书中记载的动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外，在附图中描绘的过程不一定要求示出特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中，多任务处理和并行处理也是可以的或者是可能有利的。

综上所述，在阅读本详细公开内容之后，本领域技术人员可以明白，前述详细公开内容可以仅以示例的方式呈现，并且可以不是限制性的。尽管这里没有明确说明，本领域技术人员可以理解本说明书需求囊括对实施例的各种合理改变，改进和修改。这些改变，改进和修改旨在由本说明书提出，并且在本说明书的示例性实施例的精神和范围内。

此外，本说明书中的某些术语已被用于描述本说明书的实施例。例如，“一个实施例”，“实施例”和/或“一些实施例”意味着结合该实施例描述的特定特征，结构或特性可以包括在本说明书的至少一个实施例中。因此，可以强调并且应当理解，在本说明书的各个部分中对“实施例”或“一个实施例”或“替代实施例”的两个或更多个引用不一定都指代相同的实施例。此外，特定特征，结构或特性可以在本说明书的一个或多个实施例中适当地组合。

应当理解，在本说明书的实施例的前述描述中，为了帮助理解一个特征，出于简化本说明书的目的，本说明书将各种特征组合在单个实施例、附图或其描述中。然而，这并不是说这些特征的组合是必须的，本领域技术人员在阅读本说明书的时候完全有可能

将其中一部分设备标注出来作为单独的实施例来理解。也就是说，本说明书中的实施例也可以理解为多个次级实施例的整合。而每个次级实施例的内容在少于单个前述公开实施例的所有特征的时候也是成立的。

本文引用的每个专利，专利申请，专利申请的出版物和其他材料，例如文章、书籍、说明书、出版物、文件、物品等，除了与本文件不一致或相冲突的、或者对权利要求的最宽范围具有限制性影响的文件，均可以通过引用结合于此，并用于现在或以后与本文件相关联的所有目的。此外，在任何材料对相关术语的描述、定义和/或使用与本文件对相关术语的描述、定义和/或使用之间存在任何不一致或冲突时，以本文件中的术语为准。

最后，应理解，本文公开的申请的实施方案是对本说明书的实施方案的原理的说明。其他修改后的实施例也在本说明书的范围内。因此，本说明书披露的实施例仅仅作为示例而非限制。本领域技术人员可以根据本说明书中的实施例采取替代配置来实现本说明书中的申请。因此，本说明书的实施例不限于申请中被精确地描述过的实施例。

权　利　要　求　书

1. 一种中台系统的数据处理方法，所述中台系统用于向多个租户提供目标服务，所述目标服务包括通用引擎层和个性化功能层，所述通用引擎层被配置为向所述多个租户提供所述目标服务，所述个性化功能层包括与所述多个租户对应的多个计算模块，一个计算模块被配置为向对应的租户提供所述目标服务，所述计算模块由对应的租户进行配置，其中，所述通用引擎层和所述个性化功能层之间的调用通过接口层实现，所述方法包括：

接收目标租户对所述目标服务的目标请求，所述多个租户包括所述目标租户；

基于所述目标请求调用所述通用引擎层和所述个性化功能层中与所述目标租户对应的目标计算模块执行所述目标请求，得到目标结果，所述多个计算模块包括所述目标计算模块；以及

输出所述目标结果。

2. 根据权利要求 1 所述的方法，其中，所述接口层包括 SPI 扩展接口。

3. 根据权利要求 1 所述的方法，其中，所述目标请求中携带有所述目标租户的目标租户标识，所述基于所述目标请求调用通用引擎层和个性化功能层中与所述目标租户对应的目标计算模块执行所述目标请求，得到目标结果，包括：

基于所述目标租户标识识别出所述目标计算模块；

基于所述目标请求调用所述通用引擎层执行所述目标服务的通用功能；

通过所述接口层，调用所述目标计算模块执行所述目标服务中与所述目标租户对应的功能；以及

得到所述目标结果。

4. 根据权利要求 3 所述的方法，其中，所述目标计算模块包括多个场景下的子模块，所述目标请求中还包括目标场景对应的目标场景标识，所述多个场景包括所述目标场景，所述通过所述接口层，调用所述目标计算模块执行所述目标服务中与所述目标租户对应的功能，包括：

通过所述接口层，调用所述目标计算模块中与所述目标场景标识对应的子模块执行所述目标服务中与所述目标租户的所述目标场景对应的功能。

5. 根据权利要求 4 所述的方法，其中，所述多个场景下的子模块根据所述目标租户预设的分类规则进行分类。

6. 根据权利要求 5 所述的方法，其中，所述子模块的代码通过脚本实现。

7. 根据权利要求 1 所述的方法，其中，所述计算模块与对应的租户标识通过键值对的方式存储。

8. 一种中台系统，所述中台系统用于向多个租户提供目标服务，所述目标服务包括通用引擎层和个性化功能层，所述通用引擎层被配置为向所述多个租户提供所述目标服务，所述个性化功能层包括与所述多个租户对应的多个计算模块，一个计算模块被配置为向对应的租户提供所述目标服务，所述计算模块由对应的租户进行配置，其中，所述通用引擎层和所述个性化功能层之间的调用通过接口层实现，所述中台系统包括：

至少一个存储介质，存储有至少一个指令集，用于为多个租户提供目标服务；以及

至少一个处理器，同所述至少一个存储介质通信连接，其中，所述中台系统运行时所述至少一个处理器读取所述至少一个指令集，并根据所述至少一个指令集的指示：

接收目标租户对所述目标服务的目标请求，所述多个租户包括所述目标租户；

基于所述目标请求调用所述通用引擎层和所述个性化功能层中与所述目标租户对应的目标计算模块执行所述目标请求，得到目标结果，所述多个计算模块包括所述目标计算模块；以及

输出所述目标结果。

9. 根据权利要求 8 所述的系统，其中，所述接口层包括 SPI 扩展接口。

10. 根据权利要求 8 所述的系统，其中，所述目标请求中携带有所述目标租户的目标租户标识，为了基于所述目标请求调用通用引擎层和个性化功能层中与所述目标租户对应的目标计算模块执行所述目标请求，得到目标结果，所述至少一个处理器：

基于所述目标租户标识识别出所述目标计算模块；

基于所述目标请求调用所述通用引擎层执行所述目标服务的通用功能；

通过所述接口层，调用所述目标计算模块执行所述目标服务中与所述目标租户对应的功能；以及

得到所述目标结果。

11. 根据权利要求 10 所述的系统，其中，所述目标计算模块包括多个场景下的子模块，所述目标请求中还包括目标场景对应的目标场景标识，所述多个场景包括所述目标场景，为了通过所述接口层，调用所述存储路径下的所述目标计算模块执行所述目标服务中与所述目标租户对应的功能，所述至少一个处理器：

通过所述接口层，调用所述存储路径下的所述目标计算模块中与所述目标场景标识对应的子模块执行所述目标服务中与所述目标租户的所述目标场景对应的功能。

12. 根据权利要求 11 所述的系统，其中，所述多个场景下的子模块根据所述目标租户预设的分类规则进行分类。

13. 根据权利要求 12 所述的系统，其中，所述子模块的代码通过脚本实现。

14. 根据权利要求 8 所述的系统，其中，所述计算模块与对应的租户标识通过键值对的方式存储。

15. 一种中台系统的程序修改方法，所述中台系统用于向多个租户提供目标服务，所述目标服务包括通用引擎层和个性化功能层，所述通用引擎层被配置为向所述多个租户提供所述目标服务，所述个性化功能层包括与所述多个租户对应的多个计算模块，一个计算模块被配置为向对应的租户提供所述目标服务，所述计算模块由对应的租户进行配置，其中，所述通用引擎层和所述个性化功能层之间的调用通过接口层实现，所述方法包括：

接收目标租户对所述目标服务的程序的修改请求，所述多个租户包括所述目标租户；以及

基于所述修改请求对所述个性化功能层中与所述目标租户对应的目标计算模块的代码进行修改。

16. 根据权利要求 15 所述的方法，其中，所述修改请求中携带有所述目标租户的目标租户标识，所述基于所述修改请求对所述个性化功能层中与所述目标租户对应的目标计算模块的代码进行修改，包括：

基于所述目标租户标识识别出所述目标计算模块；以及

对所述目标计算模块的代码进行修改。

17. 一种中台系统，所述中台系统用于向多个租户提供目标服务，所述目标服务包括通用引擎层和个性化功能层，所述通用引擎层被配置为向所述多个租户提供所述目标

服务，所述个性化功能层包括与所述多个租户对应的多个计算模块，一个计算模块被配置为向对应的租户提供所述目标服务，所述计算模块由对应的租户进行配置，其中，所述通用引擎层和所述个性化功能层之间的调用通过接口层实现，所述中台系统包括：

至少一个存储介质，存储有至少一个指令集，用于进行程序修改；以及

至少一个处理器，同所述至少一个存储介质通信连接，其中，所述中台系统运行时所述至少一个处理器读取所述至少一个指令集，并根据所述至少一个指令集的指示：

接收目标租户对所述目标服务的程序的修改请求，所述多个租户包括所述目标租户；以及

基于所述修改请求对所述个性化功能层中与所述目标租户对应的目标计算模块的代码进行修改。

18. 根据权利要求 17 所述的系统，其中，所述修改请求中携带有所述目标租户的目标租户标识，为了基于所述修改请求对所述个性化功能层中与所述目标租户对应的目标计算模块的代码进行修改，所述至少一个处理器：

基于所述目标租户标识识别出所述目标计算模块；以及

对所述目标计算模块的代码进行修改。

19. 一种计算机可读非易失性存储介质，其中，所述计算机可读非易失性存储介质中存储有至少一个指令集，所述至少一个指令集在被至少一个处理器执行时，实现如权利要求 1-8 中任一项所述的数据处理方法。

20. 一种计算机可读非易失性存储介质，其中，所述计算机可读非易失性存储介质中存储有至少一个指令集，所述至少一个指令集在被至少一个处理器执行时，实现如权利要求 15-16 中任一项所述的程序修改方法。