



(12) 发明专利申请

(10) 申请公布号 CN 111950866 A

(43) 申请公布日 2020. 11. 17

(21) 申请号 202010723048.0

(22) 申请日 2020.07.24

(71) 申请人 合肥森亿智能科技有限公司  
地址 230088 安徽省合肥市高新区彩虹路  
222号创新国际B座1703室

(72) 发明人 杨文龙 沈邗 罗大地

(74) 专利代理机构 上海光华专利事务所(普通  
合伙) 31219

代理人 李治东

(51) Int. Cl.

G06Q 10/06 (2012.01)

G06Q 10/10 (2012.01)

G06Q 50/04 (2012.01)

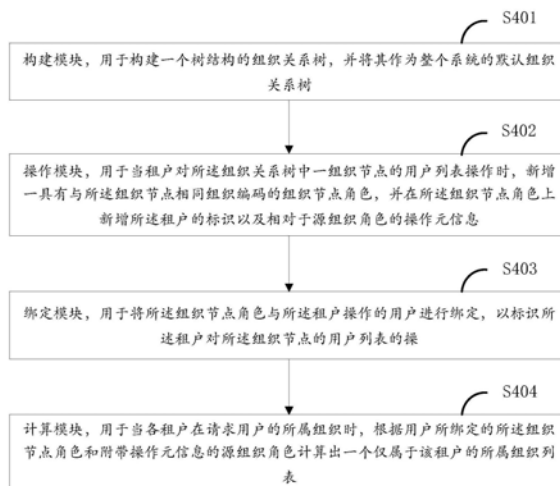
权利要求书1页 说明书7页 附图4页

(54) 发明名称

基于角色的多租户组织结构管理系统、方法、设备和介质

(57) 摘要

本申请提供了一种基于角色的多租户组织结构管理系统、方法、设备和介质,通过构建一个树结构的组织关系树;当租户对组织关系树中一组织节点的用户列表操作时,新增一具有与所述组织节点相同组织编码的组织节点角色,并在所述组织节点角色上新增所述租户的标识以及相对于源组织角色的操作元信息;将所述组织节点角色与所述租户操作的用户进行绑定,以标识所述租户对所述组织节点的用户列表的操作。本申请针对在云服务场景下,多租户对同一组织结构树进行操作并需要互相隔离的情况,采用了一种同时兼容角色模型和传统域模型的新型组织结构管理系统,有效的在保证了系统复杂度低的同时,实现了多租户共用同一组织结构树同时根据自身业务进行微调的需求。



1. 一种基于角色的多租户组织结构管理系统,其特征在于,所述系统包括:  
构建模块,用于构建一个树结构的组织关系树,并将其作为整个系统的默认组织关系树;  
操作模块,用于当租户对所述组织关系树中一组织节点的用户列表操作时,新增一具有与所述组织节点相同组织编码的组织节点角色,并在所述组织节点角色上新增所述租户的标识以及相对于源组织角色的操作元信息;  
绑定模块,用于将所述组织节点角色与所述租户操作的用户进行绑定,以标识所述租户对所述组织节点的用户列表的操作;  
计算模块,用于当各租户在请求用户的所属组织时,根据用户所绑定的所述组织节点角色和附带操作元信息的源组织角色计算出一个仅属于该租户的所属组织列表。
2. 根据权利要求1所述的系统,其特征在于,所述组织关系树中每个组织节点包含:组织名称、组织编码、父组织节点编码、及组织节点所包含的用户。
3. 根据权利要求2所述的系统,其特征在于,各所述组织节点作为实际存储对象,同时每个组织节点会映射一个角色编码与组织编码相同的角色。
4. 根据权利要求2所述的系统,其特征在于,组织关系节点与用户的包含关系同时映射为角色和用户的绑定关系。
5. 根据权利要求1所述的系统,其特征在于,所述组织结构树为基于微软域模型并支持多租户的组织结构管理系统;所述组织结构管理系统以角色作为组织结构树的具体实现模型,以兼容云计算的场景。
6. 根据权利要求1所述的系统,其特征在于,多个租户对同一组织结构树进行操作并需要互相隔离。
7. 一种基于角色的多租户组织结构管理方法,其特征在于,应用于如权利要求1-6中任意一项所述的基于角色系统的多租户组织结构管理系统,所述方法包括:  
构建一个树结构的组织关系树,并将其作为整个系统的默认组织关系树;  
当租户对所述组织关系树中一组织节点的用户列表操作时,新增一具有与所述组织节点相同组织编码的组织节点角色,并在所述组织节点角色上新增所述租户的标识以及相对于源组织角色的操作元信息;  
将所述组织节点角色与所述租户操作的用户进行绑定,以标识所述租户对所述组织节点的用户列表的操作;  
当各租户在请求用户的所属组织时,根据用户所绑定的所述组织节点角色和附带操作元信息的源组织角色计算出一个仅属于该租户的所属组织列表。
8. 一种计算机设备,其特征在于,所述设备包括:存储器、及处理器;所述存储器用于存储计算机指令;所述处理器运行计算机指令实现如权利要求7所述的方法。
9. 一种计算机可读存储介质,其特征在于,存储有计算机指令,所述计算机指令被运行时执行如权利要求7述的方法。

## 基于角色的多租户组织结构管理系统、方法、设备和介质

### 技术领域

[0001] 本发明涉及计算机系统管理技术领域,特别是涉及一种基于角色的多租户组织结构管理系统、方法、设备和介质。

### 背景技术

[0002] 组织结构的的管理一直是各种管理系统的重要功能。当前也已经有很多进行组织结构管理的模型和算法。例如微软的域模型结构,有着几十年的历史,为广大企业和机构使用。然而,在现代互联网和云计算场合,现有的组织机构管理模型和算法略显不足。尤其是作为一个中心组织结构管理系统,无法处理多租户之间对于同一组织结构的微调。目前通过建立多个组织结构树可以解决这个问题,但是会增加同步上的困难并增加整个系统的复杂性,使其变得难以维护。

### 发明内容

[0003] 鉴于以上所述现有技术的缺点,本申请的目的在于提供一种基于角色的多租户组织结构管理系统、方法、设备和介质,以解决现有技术中存在的至少一个问题。

[0004] 为实现上述目的及其他相关目的,本申请提供一种基于角色的多租户组织结构管理系统,所述系统包括:构建模块,用于构建一个树结构的组织关系树,并将其作为整个系统的默认组织关系树;操作模块,用于当租户对所述组织关系树中一组织节点的用户列表操作时,新增一具有与所述组织节点相同组织编码的组织节点角色,并在所述组织节点角色上新增所述租户的标识以及相对于源组织角色的操作元信息;绑定模块,用于将所述组织节点角色与所述租户操作的用户进行绑定,以标识所述租户对所述组织节点的用户列表的操作;计算模块,用于当各租户在请求用户的所属组织时,根据用户所绑定的所述组织节点角色和附带操作元信息的源组织角色计算出一个仅属于该租户的所属组织列表。

[0005] 于本申请的一实施例中,所述组织关系树中每个组织节点包含:组织名称、组织编码、父组织节点编码、及组织节点所包含的用户。

[0006] 于本申请的一实施例中,各所述组织节点作为实际存储对象,同时每个组织节点会映射一个角色编码与组织编码相同的角色。

[0007] 于本申请的一实施例中,组织关系节点与用户的包含关系同时映射为角色和用户的绑定关系。

[0008] 于本申请的一实施例中,所述组织结构树为基于微软域模型并支持多租户的组织结构管理系统;所述组织结构管理系统以角色作为组织结构树的具体实现模型,以兼容云计算的场景。

[0009] 于本申请的一实施例中,多个租户对同一组织结构树进行操作并需要互相隔离。

[0010] 为实现上述目的及其他相关目的,本申请提供一种基于角色的多租户组织结构管理方法,应用于如上所述的基于角色系统的多租户组织结构管理系统,所述方法包括:构建一个树结构的组织关系树,并将其作为整个系统的默认组织关系树;当租户对所述组织关

系树中一组织节点的用户列表操作时,新增一具有与所述组织节点相同组织编码的组织节点角色,并在所述组织节点角色上新增所述租户的标识以及相对于源组织角色的操作元信息;将所述组织节点角色与所述租户操作的用户进行绑定,以标识所述租户对所述组织节点的用户列表的操作;当各租户在请求用户的所属组织时,根据用户所绑定的所述组织节点角色和附带操作元信息的源组织角色计算出一个仅属于该租户的所属组织列表。

[0011] 为实现上述目的及其他相关目的,本申请提供一种计算机设备,所述设备包括:存储器、及处理器;所述存储器用于存储计算机指令;所述处理器运行计算机指令实现如上所述的方法。

[0012] 为实现上述目的及其他相关目的,本申请提供一种计算机可读存储介质,存储有计算机指令,所述计算机指令被运行时执行如上所述的方法。

[0013] 综上所述,本申请提供一种基于角色的多租户组织结构管理系统、方法、设备和介质,通过构建一个树结构的组织关系树,并将其作为整个系统的默认组织关系树;当租户对所述组织关系树中一组织节点的用户列表操作时,新增一具有与所述组织节点相同组织编码的组织节点角色,并在所述组织节点角色上新增所述租户的标识以及相对于源组织角色的操作元信息;将所述组织节点角色与所述租户操作的用户进行绑定,以标识所述租户对所述组织节点的用户列表的操作;当各租户在请求用户的所属组织时,根据用户所绑定的所述组织节点角色和附带操作元信息的源组织角色计算出一个仅属于该租户的所属组织列表。

[0014] 具有以下有益效果:

[0015] 本申请针对在云服务场景下,多租户对同一组织结构树进行操作并需要互相隔离的情况,采用了一种同时兼容角色模型和传统域模型的新型组织结构管理系统,有效的在保证系统复杂度低的同时,实现了多租户共用同一组织结构树同时根据自身业务进行微调的需求。

## 附图说明

[0016] 图1显示为本申请于一实施例中基于角色的多租户组织结构管理系统的模块示意图。

[0017] 图2-3显示为本申请于一实施例中基于角色的多租户组织结构管理系统的场景应用示意图。

[0018] 图4显示为本申请于一实施例中基于角色的多租户组织结构管理方法的流程示意图。

[0019] 图5显示为本申请于一实施例中计算机设备的结构示意图。

## 具体实施方式

[0020] 以下通过特定的具体实例说明本申请的实施方式,本领域技术人员可由本说明书所揭露的内容轻易地了解本申请的其他优点与功效。本申请还可以通过另外不同的具体实施方式加以实施或应用,本说明书中的各项细节也可以基于不同观点与应用,在没有背离本申请的精神下进行各种修饰或改变。需说明的是,在不冲突的情况下,以下实施例及实施例中的特征可以相互组合。

[0021] 需要说明的是,以下实施例中所提供的图示仅以示意方式说明本申请的基本构想,虽然图式中仅显示与本申请中有关的组件而非按照实际实施时的组件数目、形状及尺寸绘制,但其实际实施时各组件的型态、数量及比例可为一种随意的改变,且其组件布局型态也可能更为复杂。

[0022] 为了解决上述现有技术的问题,本申请发明了一种基于微软域模型并支持多租户的组织结构管理系统。同时为了更好的兼容云计算的场景,本发明使用角色作为组织结构树的具体实现模型。

[0023] 如图1所示,展示为本申请于一实施例中的基于角色的多租户组织结构管理系统的模块示意图。如图所示,所述系统100包括:

[0024] 构建模块101,用于构建一个树结构的组织关系树,并将其作为整个系统的默认组织关系树;

[0025] 于本实施例中,构建一颗树结构的组织关系树,并将其作为整个系统的默认组织关系树。

[0026] 其中,每个节点包含了以下信息:1)组织名称;2)组织编码;3)父组织节点编码;4)该组织节点所包含的用户。同时,作为组织节点的实际存储对象,每个节点会同时映射一个角色编码与组织编码相同的角色;组织关系节点与用户的包含关系同时映射为角色和用户的绑定关系。

[0027] 于本实施例中,所述组织结构树为基于微软域模型并支持多租户的组织结构管理系统;所述组织结构管理系统以角色作为组织结构树的具体实现模型,以兼容云计算的场景。

[0028] Windows域是计算机网络的一种形式,其中所有用户帐户,计算机,打印机和其他安全主体都在位于称为域控制器的一个或多个中央计算机集群上的中央数据库中注册。身份验证在域控制器上进行。在域中使用计算机的每个人都会收到一个唯一的用户帐户,然后可以为该帐户分配对该域内资源的访问权限。从Windows Server 2003开始,Active Directory是负责维护该中央数据库的Windows组件。Windows域的概念与工作组的概念形成对比,在该工作组中,每台计算机都维护自己的安全主体数据库。域:域是一种管理边界,用于一组计算机共享共用的安全数据库,域实际上就是一组服务器和工作站的集合。

[0029] 域英文叫DOMAIN——域(Domain)是Windows网络中独立运行的单位,域之间相互访问则需要建立信任关系(即Trust Relation)。信任关系是连接在域与域之间的桥梁。当一个域与其他域建立了信任关系后,2个域之间不但可以按需要相互进行管理,还可以跨网分配文件和打印机等设备资源,使不同的域之间实现网络资源的共享与管理,以及相互通信和数据传输。

[0030] 域既是Windows网络操作系统的逻辑组织单元,也是Internet的逻辑组织单元,在Windows网络操作系统中,域是安全边界。域管理员只能管理域的内部,除非其他的域显式地赋予他管理权限,他才能够访问或者管理其他的域;每个域都有自己的安全策略,以及它与其他域的安全信任关系。

[0031] 于本实施例中,多个租户对同一组织结构树进行操作并需要互相隔离。

[0032] 一般来说,多租户,解决的就是让多个客户“共享”统一使用一套程序界面,且保证不同客户之间的数据各自独立。它是一种架构,也可以在同一个服务器上运行的多个程序

实例,为多个客户(租户,通常指的是企业级客户)提供服务。形象点的说,“租户”和我们合租一套房子是类同的意思,张三和李四同租一套房子,各住一个卧室,互相不干扰,客厅、厨房为公共区域,大家可以一起公用一些生活设施。租户的意思,从字面的理解,也就是租用房东的房子住,不具有产权,只在有限的范围内拥有使用权,且各个租户只能在自己租住的房间贴个墙纸搞点小装饰,不能拆门拆墙搞装修。房东(平台方)不但有整套房子的大门,还可能收回出租的房间。这种架构也称之为SAAS(软件即服务),它能够支持不同租户之间数据的和配置的隔离,从而保证每个租户数据的安全和隐私,以及用户对界面、业务逻辑、和数据结构的个性化需求,而平台方,不但掌握全局的业务,可以调整业务流程,还掌握全部的数据。也就是,在多租户模式下,完全可以实现同一个平台下不同租户的业务单据不同,流程不同,而又同时从属于同一个平台,对平台而言,租户只是平台的一个账户,依托于平台开展业务。形成的局面就是,平台级用户管理整个平台的数据和用户,租户级别的用户只能管理该租户下的用户和业务数据,平台和租户,租户和用户之间是一种1对多和多对1的三层体系结构。

[0033] 操作模块102,用于当租户对所述组织关系树中一组织节点的用户列表操作时,新增一具有与所述组织节点相同组织编码的组织节点角色,并在所述组织节点角色上新增所述租户的标识以及相对于源组织角色的操作元信息;

[0034] 绑定模块103,用于将所述组织节点角色与所述租户操作的用户进行绑定,以标识所述租户对所述组织节点的用户列表的操作;

[0035] 计算模块104,用于当各租户在请求用户的所属组织时,根据用户所绑定的所述组织节点角色和附带操作元信息的源组织角色计算出一个仅属于该租户的所属组织列表。

[0036] 于一或多个实施例中,当某个租户对某个特定的组织节点的用户列表进行添加用户或者移除用户操作时,会新增一个具有相同组织编码的组织节点角色,并在该角色上新增租户的标识以及相对于源组织角色的操作元信息:如添加/移除。接着将该角色与租户操作的用户进行绑定,来标识租户对组织节点用户列表的操作。最后,各租户在请求用户的所属组织时,将根据用户所绑定的组织节点角色和附带操作元信息的角色计算出一个仅属于该租户的所属组织列表。

[0037] 举例来说,如图2所示,在数据库中有一默认组织-医院,该医院有中医科,在中医科下附属有中医门诊,鼻炎门诊两个附属科室。张三同时属于中医门诊和鼻炎门诊科室,李四只属于鼻炎门诊科室。

[0038] 那么对于默认组织结构树,它拥有三个节点,分别是:中医科、中医门诊、鼻炎门诊。其中中医门诊和鼻炎门诊的父节点字段为中医科。同时,这三个节点与三个同名角色互相映射。

[0039] 张三与中医门诊和鼻炎门诊的组织角色进行绑定,李四与鼻炎门诊的组织角色进行绑定。

[0040] 接下来,应用A作为租户,将张三移出中医门诊,将李四加入中医门诊。此时,创建两个组织角色,分别为“+中医门诊/应用A”,并将其操作元数据标记为添加以及“-中医门诊/应用A”,并将其操作原数据标记为移除。接着,将“+中医门诊/应用A”绑定至李四,将“-中医门诊/应用A”绑定至张三。

[0041] 在应用A请求张三的所属科室时,由于张三与“-中医门诊/应用A”、“中医门诊”、

“鼻炎门诊”三个角色绑定,在计算后张三对于应用A来说只有“鼻炎门诊”的组织角色。而对于应用B来说,标记了应用A的组织角色对其不可见,所以不会参与到最终的合成。因此应用B在请求张三的所属科室时,将得到“中医门诊”和“鼻炎门诊”的结果,如图3所示。

[0042] 综上所述,本申请针对在云服务场景下,多租户对同一组织结构树进行操作并需要互相隔离的情况,采用了一种同时兼容角色模型和传统域模型的新型组织结构管理系统,有效的在保证了系统复杂度低的同时,实现了多租户共用同一组织结构树同时根据自身业务进行微调的需求。

[0043] 需要说明的是,上述装系统各模块/单元之间的信息交互、执行过程等内容,由于与本申请所述方法实施例基于同一构思,其带来的技术效果与本申请方法实施例相同,具体内容可参见本申请前述所示的方法实施例中的叙述,此处不再赘述。

[0044] 还需要说明的是,应理解以上系统的各个模块的划分仅仅是一种逻辑功能的划分,实际实现时可以全部或部分集成到一个物理实体上,也可以物理上分开。且这些单元可以全部以软件通过处理元件调用的形式实现;也可以全部以硬件的形式实现;还可以部分模块通过处理元件调用软件的形式实现,部分模块通过硬件的形式实现。例如,计算模块104可以为单独设立的处理元件,也可以集成在上述系统的某一个芯片中实现,此外,也可以以程序代码的形式存储于上述系统的存储器中,由上述系统的某一个处理元件调用并执行以上计算模块104的功能。其它模块的实现与之类似。此外这些模块全部或部分可以集成在一起,也可以独立实现。这里所述的处理元件可以是一种集成电路,具有信号的处理能力。在实现过程中,上述方法的各步骤或以上各个模块可以通过处理器元件中的硬件的集成逻辑电路或者软件形式的指令完成。

[0045] 例如,以上这些模块可以是配置成实施以上方法的一个或多个集成电路,例如:一个或多个特定集成电路(Application Specific Integrated Circuit,简称ASIC),或,一个或多个微处理器(digital signal processor,简称DSP),或,一个或者多个现场可编程门阵列(Field Programmable Gate Array,简称FPGA)等。再如,当以上某个模块通过处理元件调度程序代码的形式实现时,该处理元件可以是通用处理器,例如中央处理器(Central Processing Unit,简称CPU)或其它可以调用程序代码的处理器。再如,这些模块可以集成在一起,以片上系统(system-on-a-chip,简称SOC)的形式实现。

[0046] 如图4所示,展示为本申请于一实施例中的基于角色的多租户组织结构管理方法的流程示意图。其中,所述方法应用于如图1所述的基于角色系统的多租户组织结构管理系统,如图所示,所述方法包括:

[0047] 步骤S401:构建一个树结构的组织关系树,并将其作为整个系统的默认组织关系树;

[0048] 步骤S402:当租户对所述组织关系树中一组织节点的用户列表操作时,新增一具有与所述组织节点相同组织编码的组织节点角色,并在所述组织节点角色上新增所述租户的标识以及相对于源组织角色的操作元信息;

[0049] 步骤S403:将所述组织节点角色与所述租户操作的用户进行绑定,以标识所述租户对所述组织节点的用户列表的操作;

[0050] 步骤S404:当各租户在请求用户的所属组织时,根据用户所绑定的所述组织节点角色和附带操作元信息的源组织角色计算出一个仅属于该租户的所属组织列表。

[0051] 于本实施例中,构建一颗树结构的组织关系树,并将其作为整个系统的默认组织关系树。

[0052] 其中,每个节点包含了以下信息:1)组织名称;2)组织编码;3)父组织节点编码;4)该组织节点所包含的用户。同时,作为组织节点的实际存储对象,每个节点会同时映射一个角色编码与组织编码相同的角色;组织关系节点与用户的包含关系同时映射为角色和用户的绑定关系。

[0053] 于本实施例中,所述组织结构树为基于微软域模型并支持多租户的组织结构管理系统;所述组织结构管理系统以角色作为组织结构树的具体实现模型,以兼容云计算的场景。

[0054] 于一或多个实施例中,当某个租户对某个特定的组织节点的用户列表进行添加用户或者移除用户操作时,会新增一个具有相同组织编码的组织节点角色,并在该角色上新增租户的标识以及相对于源组织角色的操作元信息:如添加/移除。接着将该角色与租户操作的用户进行绑定,来标识租户对组织节点用户列表的操作。最后,各租户在请求用户的所属组织时,将根据用户所绑定的组织节点角色和附带操作元信息的角色计算出一个仅属于该租户的所属组织列表。

[0055] 综上所述,本申请针对在云服务场景下,多租户对同一组织结构树进行操作并需要互相隔离的情况,采用了一种同时兼容角色模型和传统域模型的新型组织结构管理系统,有效的在保证了系统复杂度低的同时,实现了多租户共用同一组织结构树同时根据自身业务进行微调的需求。

[0056] 如图5所示,展示为本申请于一实施例中的计算机设备的结构示意图。如图所示,所述计算机设备500包括:存储器501、及处理器502;所述存储器501用于存储计算机指令;所述处理器502运行计算机指令实现如图4所述的方法。

[0057] 在一些实施例中,所述计算机设备500中的所述存储器501的数量均可以是一或多个,所述处理器502的数量均可以是一或多个,而图5中均以一个为例。

[0058] 于本申请一实施例中,所述计算机设备500中的处理器502会按照如图4所述的步骤,将一个或多个以应用程序的进程对应的指令加载到存储器501中,并由处理器502来运行存储在存储器501中的应用程序,从而实现如图1所述的方法。

[0059] 所述存储器501可以包括随机存取存储器(Random Access Memory,简称RAM),也可以包括非易失性存储器(non-volatile memory),例如至少一个磁盘存储器。所述存储器501存储有操作系统和操作指令、可执行模块或者数据结构,或者它们的子集,或者它们的扩展集,其中,操作指令可包括各种操作指令,用于实现各种操作。操作系统可包括各种系统程序,用于实现各种基础业务以及处理基于硬件的任务。

[0060] 所述处理器502可以是通用处理器,包括中央处理器(Central Processing Unit,简称CPU)、网络处理器(Network Processor,简称NP)等;还可以是数字信号处理器(Digital Signal Processing,简称DSP)、专用集成电路(Application Specific Integrated Circuit,简称ASIC)、现场可编程门阵列(Field-Programmable Gate Array,简称FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。

[0061] 在一些具体的应用中,所述计算机设备500的各个组件通过总线系统耦合在一起,其中总线系统除包括数据总线之外,还可以包括电源总线、控制总线和状态信号总线等。但

是为了清除说明起见,在图5中将各种总线都成为总线系统。

[0062] 综上所述,本申请提供了一种基于角色的多租户组织结构管理系统、方法、设备和介质,通过构建一个树结构的组织关系树,并将其作为整个系统的默认组织关系树;当租户对所述组织关系树中一组织节点的用户列表操作时,新增一具有与所述组织节点相同组织编码的组织节点角色,并在所述组织节点角色上新增所述租户的标识以及相对于源组织角色的操作元信息;将所述组织节点角色与所述租户操作的用户进行绑定,以标识所述租户对所述组织节点的用户列表的操作;当各租户在请求用户的所属组织时,根据用户所绑定的所述组织节点角色和附带操作元信息的源组织角色计算出一个仅属于该租户的所属组织列表。

[0063] 本申请有效克服了现有技术中的种种缺点而具高度产业利用价值。

[0064] 上述实施例仅例示性说明本申请的原理及其功效,而非用于限制本发明。任何熟悉此技术的人士皆可在不违背本申请的精神及范畴下,对上述实施例进行修饰或改变。因此,举凡所属技术领域中包含通常知识者在未脱离本发明所揭示的精神与技术思想下所完成的一切等效修饰或改变,仍应由本申请的权利要求所涵盖。

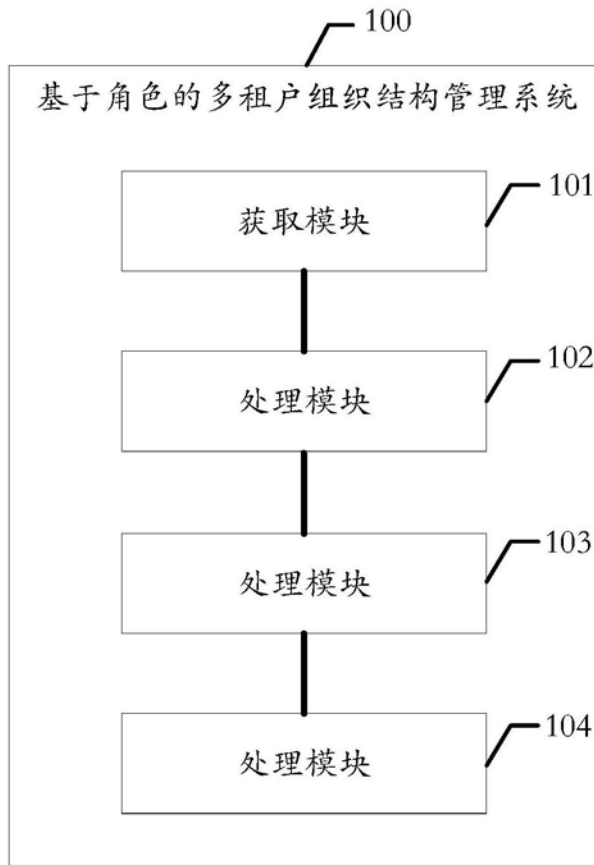


图1

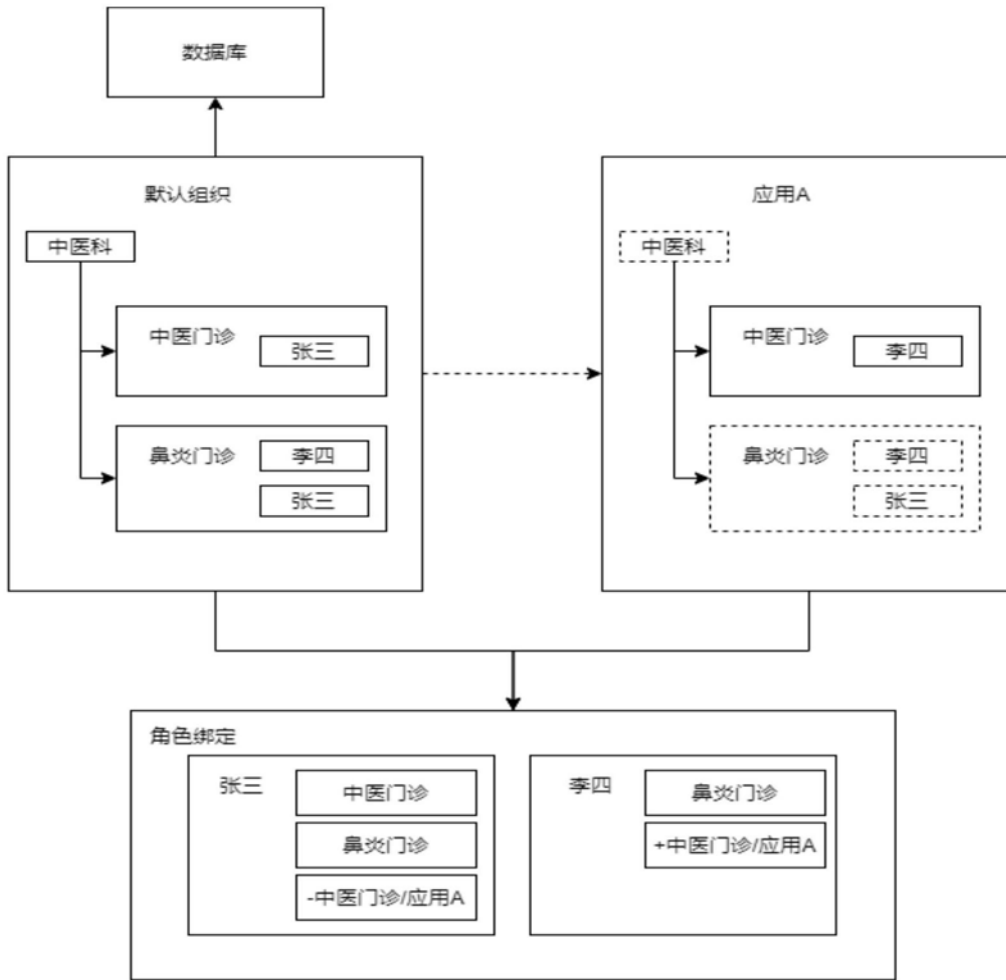


图2

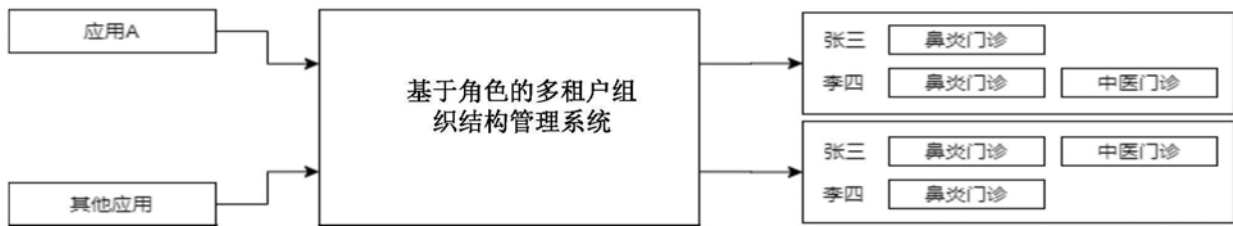


图3

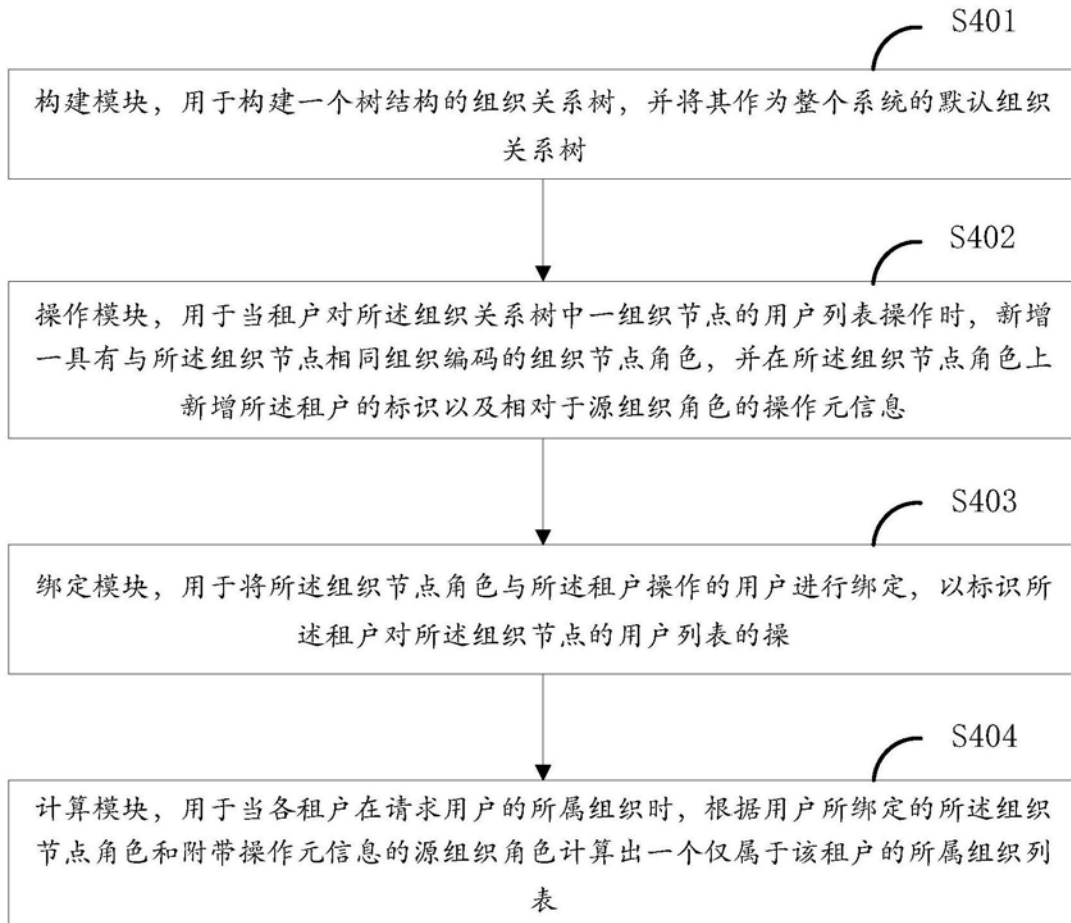


图4

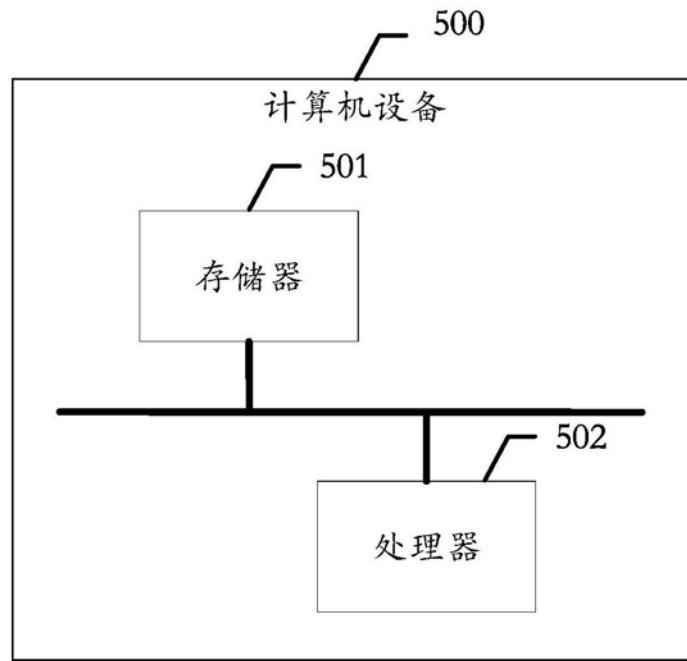


图5