



(12) 发明专利

(10) 授权公告号 CN 110870253 B

(45) 授权公告日 2023.04.28

(21) 申请号 201880044147.7
 (22) 申请日 2018.06.28
 (65) 同一申请的已公布的文献号
 申请公布号 CN 110870253 A
 (43) 申请公布日 2020.03.06
 (30) 优先权数据
 62/527,479 2017.06.30 US
 62/553,406 2017.09.01 US
 15/829,684 2017.12.01 US
 (85) PCT国际申请进入国家阶段日
 2019.12.30
 (86) PCT国际申请的申请数据
 PCT/US2018/040105 2018.06.28
 (87) PCT国际申请的公布数据
 WO2019/006177 EN 2019.01.03

(73) 专利权人 甲骨文国际公司
 地址 美国加利福尼亚
 (72) 发明人 M·邓肯
 (74) 专利代理机构 中国贸促会专利商标事务所
 有限公司 11038
 专利代理师 李晓芳
 (51) Int.Cl.
 H04L 9/32 (2006.01)
 H04L 9/12 (2006.01)
 (56) 对比文件
 CN 106233252 A, 2016.12.14
 US 2017116693 A1, 2017.04.27
 审查员 马文文

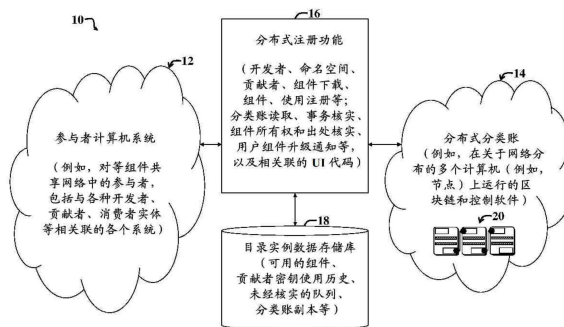
权利要求书3页 说明书25页 附图19页

(54) 发明名称

使用分布式分类账管理公共软件组件生态系统的系统和方法

(57) 摘要

实施例包括促进使用分布式分类账来实现软件组件生态系统的系统和方法,例如,如部分地被实现为分布式区块链,该区块链由称为目录实例的生态系统完整节点来维护。可以由一个或多个目录实例将各种注册事务,例如开发者注册、命名空间注册、使用注册等,添加到分布式区块链。将区块添加到区块链可以包括使用趋向事务的树散列来为新区块生成区块散列。新区块还包括或结合紧挨着前一区块的散列。



1. 存储指令的一种或多种有形存储介质,所述指令由一个或多个处理器执行并且在被执行时能操作用于通过执行以下动作来实现软件组件生态系统:

接受来自用户输入设备的一个或多个信号,以定义开发者实体的识别名称;

接受来自用户输入设备的一个或多个信号,以定义能用于核实开发者实体被授权注册该识别名称的安全核实机制;

记录目录条目,该目录条目包括与安全核实机制相关联的开发者实体名称;

与目录条目相关联地执行事务以传送软件组件;

使用目录条目来核实事务,由此将表征事务的数据提交到目录;以及

使用该软件组件生态系统的组件目录的实例来执行以下操作,该软件组件生态系统包括多个互相通信的目录实例:

接受与开发者实体名称有关的开发者实体注册信息;以及

核实开发者实体注册信息并将开发者实体注册信息提交到被实现为区块链的分布式数据库作为开发者实体注册记录或区块,其中,该区块链能被所述多个互相通信的目录实例中的每一个目录实例访问,从而将开发者实体注册信息传播到软件组件生态系统的完整节点。

2. 如权利要求1所述的一种或多种有形存储介质,其中安全核实机制包括公钥和数字签名。

3. 如权利要求2所述的一种或多种有形存储介质,还包括:

将目录条目传播到软件组件生态系统中的多个节点。

4. 如权利要求3所述的一种或多种有形存储介质,其中所述多个节点中的每一个节点表示目录实例,该目录实例包括用于与所述多个节点协同实现分布式区块链分类账的代码,所述多个节点维护所述分布式区块链分类账的副本。

5. 如权利要求2所述的一种或多种有形存储介质,还包括:

为目录条目分配命名空间。

6. 如权利要求1所述的一种或多种有形存储介质,其中在以下操作之后接受所述开发者实体注册信息:

由开发者实体向所述多个互相通信的目录实例中的目录实例供应开发者实体名称、要为其执行注册的目录实例相关联的公钥、以及与开发者实体名称相关联且能用于确认开发者实体身份的数字签名。

7. 如权利要求1所述的一种或多种有形存储介质,还包括:

在传播开发者实体注册信息之后,并且响应于命名空间注册过程的发起和实现,通过以下操作为开发者实体分配一个或多个命名空间:

在命名空间注册过程期间,将一个或多个供应的命名空间与开发者注册信息相关联。

8. 如权利要求7所述的一种或多种有形存储介质,还包括:

选择性地延迟开发者实体的命名空间注册过程的实现,直到将若干记录提交到区块链的记录的链之后。

9. 一种通过执行以下动作来实现软件组件生态系统的方法:

接受来自用户输入设备的一个或多个信号,以定义开发者实体的识别名称;

接受来自用户输入设备的一个或多个信号,以定义安全核实机制;

记录目录条目,该目录条目包括与安全核实机制相关联的开发者实体名称;
与目录条目相关联地执行事务以传送软件组件;
使用目录条目来核实事务;以及

使用该软件组件生态系统的组件目录的实例来执行以下操作,该软件组件生态系统包括多个互相通信的目录实例:

接受与开发者实体名称有关的开发者实体注册信息;以及

核实开发者实体注册信息并将开发者实体注册信息提交到被实现为区块链的分布式数据库作为开发者实体注册记录或区块,其中,该区块链能被所述多个互相通信的目录实例中的每一个目录实例访问,从而将开发者实体注册信息传播到软件组件生态系统的完整节点。

10. 如权利要求9所述的方法,其中安全核实机制包括:
公钥和数字签名。

11. 如权利要求10所述的方法,还包括:

将目录条目传播到软件组件生态系统中的多个节点。

12. 如权利要求11所述的方法,其中所述多个节点中的每一个节点表示:

目录实例,该目录实例包括用于与所述多个节点协同实现分布式区块链分类账的代码,所述多个节点维护所述分布式区块链分类账的副本。

13. 如权利要求10所述的方法,还包括:

为目录条目分配命名空间。

14. 如权利要求9所述的方法,其中在以下操作之后接受所述开发者实体注册信息:

由开发者实体向所述多个互相通信的目录实例中的目录实例供应开发者实体名称、与要为其执行注册的目录实例相关联的公钥、以及与开发者实体名称相关联且能用于确认开发者实体身份的数字签名。

15. 如权利要求9所述的方法,还包括:

在传播开发者实体注册信息之后,并且响应于命名空间注册过程的发起和实现,通过以下操作为开发者实体分配一个或多个命名空间:

在命名空间注册过程期间,将一个或多个供应的命名空间与开发者注册信息相关联。

16. 一种用于实现软件组件生态系统的装置,包括:

一个或多个处理器;

在一种或多种有形介质中存储的指令,所述指令由所述一个或多个处理器执行并且在被执行时能操作用于:

接受来自用户输入设备的一个或多个信号,以定义开发者实体的识别名称;

接受来自用户输入设备的一个或多个信号,以定义安全核实机制;

记录目录条目,该目录条目包括与安全核实机制相关联的开发者实体名称;

与目录条目相关联地执行事务以传送软件组件;

使用目录条目来核实事务;以及

使用该软件组件生态系统的组件目录的实例来执行以下操作,该软件组件生态系统包括多个互相通信的目录实例:

接受与开发者实体名称有关的开发者实体注册信息;以及

核实开发者实体注册信息并将开发者实体注册信息提交到被实现为区块链的分布式数据库作为开发者实体注册记录或区块,其中,该区块链能被所述多个互相通信的目录实例中的每一个目录实例访问,从而将开发者实体注册信息传播到软件组件生态系统的完整节点。

17. 一种用于实现软件组件生态系统的装置,包括:

一个或多个处理器;

在一种或多种有形介质中存储的指令,所述指令由所述一个或多个处理器执行并且在被执行时能操作用于:

在软件组件生态系统的第一节点处接收事务信息;

使用第一节点将事务信息打包到第一事务中,其中,第一事务包括表征软件组件生态系统的一个或多个事件、发生、属性或关系的文档;

选择性地将事务传播到由第一节点和与第一节点互相通信的一个或多个附加节点维护的一个或多个分布式未经核实的队列;以及

确定所述一个或多个分布式未经核实的队列中的包括第一事务的一个或多个事务要被处理并被包括在由第一节点和所述一个或多个附加节点维护的分布式区块链的新区块中,然后:

使用所述一个或多个事务来计算第一散列,以及

采用与区块链的终端区块相关联的先前区块散列和第一散列来计算区块链的新区块的新散列;

将包括第一事务的所述一个或多个事务结合到区块链的新区块中,其中新区块结合了新散列;以及

将新区块追加到区块链。

使用分布式分类账管理公共软件组件生态系统的系统和方法

[0001] 相关申请的交叉引用

[0002] 本申请要求于2017年6月30日提交的标题为“SYSTEM AND METHOD FOR MANAGING A PUBLIC SOFTWARE COMPONENT ECOSYSTEM USING A DISTRIBUTED LEDGER”的美国临时专利申请序列No.62/527,479以及于2017年9月1日提交的标题为“SYSTEM FOR MANAGING A PUBLIC SOFTWARE COMPONENT ECOSYSTEM USING BLOCKCHAIN FEATURES”的美国临时专利申请序列No.62/553,406以及于2017年12月1日提交的标题为“SYSTEM AND METHOD FOR MANAGING A PUBLIC SOFTWARE COMPONENT ECOSYSTEM USING A DISTRIBUTED LEDGER”的美国专利申请序列No.15/829,684的权益,所有这些都通过引用并入本文,用于所有目的,就如同在本申请中完全阐述了一样。

技术领域

[0003] 本申请涉及计算,并且更具体地涉及适于在生态系统的参与者之间共享诸如软件组件之类的计算资源的计算生态系统。

背景技术

[0004] 用于实现和控制软件组件生态系统的系统和方法在各种要求苛刻的应用中采用,包括应用商店和开源软件开发平台。此类应用常常要求高效机制以选择性地处理并考虑生态系统中发生的事务,然后充分利用事务信息来确保生态系统的高效操作。

[0005] 按照惯例,用于交换内容(例如,软件程序和组件、数据、对象等)的计算生态系统或者被集中控制(诸如由应用商店的所有者),或者相对不受控制。但是,集中控制的生态系统会遭受各种缺点,诸如实现适当健壮的中央管理基础设施的成本、在此类系统上管理可伸缩性和灾难恢复的成本,与它们的管理相关的人员配备的总成本等。因而,仍然需要用于促进计算生态系统的高效实现和控制的系统和方法。

发明内容

[0006] 示例实施例公开了一种系统和方法,该系统和方法促进部分地通过使用分布式区块链分类账来维护、保护、核实、更新、监视或以其它方式控制或影响事务(例如,分布式生态系统中发生的各种类型的注册事务(如下文更全面地讨论)),使用分布式控制来实现软件组件生态系统。分布式生态系统的生态系统节点(所谓的或由本文的目录实例表示的)运行软件用于管理共享区块链分类账(本文也简称为区块链),包括添加包括一个或多个事务的新区块。

[0007] 用于促进实现软件组件生态系统的示例方法包括:接受来自用户输入设备的一个或多个信号,以定义开发者实体的识别名称;接受来自用户输入设备的一个或多个信号,以定义可用于核实开发者实体被授权注册该识别名称的安全核实机制;将条目录在目录中(例如,实现节点的目录以及随附的分布式分类账),该条目包括与核实机制相关联的开发者名称;与目录条目相关联地执行组件上传、下载或其它组件传送,以传送软件组件;以及

使用目录条目(例如,开发者实体注册事务条目)来核实与组件上传、下载或其它软件组件传送有关的事务(例如,将事务提交到分布式分类账),由此向分布式分类账提交包括事务的数据。

[0008] 在更具体的实施例中,示例方法还指定安全核实机制包括与开发者实体相关联的公共密钥和数字签名。在将目录条目的复制传播到软件组件生态系统中的多个节点之前,首先将记录的目录条目放置在未经核实的队列中。多个节点中的每个节点表示目录实例,该目录实例包括用于与其它多个节点协同实现分布式区块链分类账的代码,其它多个节点维护分布式区块链分类账的副本或近似该副本。

[0009] 因而,本文讨论的某些实施例提供了用于促进在分布式分类账(例如,分布式区块链)中事务数据的注册的高效机制。与事务有关的信息可以被用于确保参与者和相关联组件的信任;管理并核实组件的身份、完整性、出处或来源、支持状况等;确保已正确许可软件组件;提供关于可用软件支持服务的信息;批准用户,诸如组件的消费者和/或贡献者;防止或减轻当提供具有冲突或混淆名称的组件时可能发生的问题;促进软件组件的更新和/或召回;等等,所有这些都不需要使用分开管理的核心控制器。

[0010] 通过参考说明书的其余部分和附图,可以实现对本文公开的特定实施例的性质和优点的进一步理解。

附图说明

[0011] 图1图示了用于选择性地共享软件组件并用于使用分布式区块链分类账来注册事务的第一示例广义对等计算生态系统。

[0012] 图2是图示图1的第一示例对等计算生态系统的示例实现的更详细视图。

[0013] 图3示出了充当图1-2的对等计算生态系统的对等体或节点的个人开发者计算系统的示例细节,包括示例用户界面(UI)控件和选项以及可以包括在用于将用户开发者计算系统耦合到对等生态系统的目录实例中的示例模块,由此用户开发者计算系统充当对等生态系统的节点。

[0014] 图4示出了利用图1-3的实施例的用于向分布式区块链分类账注册开发者实体的第二示例生态系统和随附的目录实例过程,其中第二示例系统可以根据图1-3的实施例来实现。

[0015] 图5是可经由图4的第二示例生态系统来实现的第一示例方法的流程图。

[0016] 图6示出了示例UI显示屏,其可以显示在计算系统(诸如图3的用户开发者计算系统)上,并且其提供了用于浏览软件组件的目录、搜索、下载、编辑以及从目录中删除软件组件的各种UI控件和选项,该目录由随附的计算系统可访问的目录实例维护。

[0017] 图7是图示第三示例生态系统和随附的计算环境的第一框图,其图示了由对等软件组件生态系统的称为目录实例的节点维护的区块链的初始状态。

[0018] 图8是第二框图,图示了在第一示例事务(即,开发者注册事务)已经经由原始目录实例(例如,第一目录实例)传播到目录实例的未经核实的队列之后的图7的第三示例系统。

[0019] 图9是第三框图,图示了在由(根据选择规则或算法)已被选择来处理(例如,核实或验证)开发者注册事务的目录实例(例如,第二目录实例)进行处理(即,核实初始开发者注册事务和向区块链提交)期间的图8的第三示例系统。

[0020] 图10是第四框图,图示了在由所选择的目录实例计算区块链的区块期间的图9的第三示例系统,其中区块包括未经核实的队列中的未决事务,包括图9的开发者注册事务。

[0021] 图11是第五框图,图示了在由第二目录实例处理的区块已经被提交给区块链并且传播到由生态系统的每个目录实例维护的区块链副本之后的图10的示例系统。

[0022] 图12是第六框图,图示了在第二目录实例已经发起第二事务(即,第一命名空间注册事务)并且已经将第二事务传播到软件组件生态系统中的其它目录实例的未经核实的队列之后的图11的示例系统。

[0023] 图13是第七框图,图示了在第三目录实例已经发起第三事务(即,第二命名空间注册事务)之后的图12的示例系统,该第三事务已经被传播到软件组件生态系统中的其它目录实例的未经核实的队列。

[0024] 图14是第八框图,图示了在已经选择第一目录实例来核实附加区块以包括在区块链中之后的图13的示例系统,其中附加区块将分别结合图12和13的第二事务和第三事务。

[0025] 图15是图示图14的示例系统的第九框图,其示出了在将要被添加到软件组件生态系统的区块链的附加区块的计算期间的第一目录实例。

[0026] 图16是第十框图,图示了在第一目录实例已经完成对附加区块的处理并且已经将附加区块传播到软件组件生态系统的区块链之后的图15的示例系统。

[0027] 图17是可经由图1-16的实施例实现的第二示例方法的流程图。

[0028] 图18是可用于实现图1-17的实施例的系统和随附的计算环境的总体框图。

[0029] 图19是可用于实现图1-18的实施例的计算设备的总体框图。

具体实施方式

[0030] 出于本讨论的目的,软件组件生态系统可以是包括被配置为使得能够向分布式计算资源上传和/或从分布式计算资源下载软件组件(例如,目录实例、随附的分布式区块链等)的联网的分布式计算资源的集合的任何计算环境。联网的计算环境可以是任何计算环境,其包括互相通信的计算机,即,计算机网络,诸如局域网(LAN)、广域网(WAN,例如互联网)、云基础设施和服务等。类似地,联网的软件应用可以是计算机代码,其适于促进经由网络与一个或多个计算资源(例如,服务器)通信或以其它方式使用一个或多个计算资源(例如,服务器)。

[0031] 注意的是,计算资源的集合(例如,可以经由生态系统的网络互相通信的计算机系统)在本文被称为节点。给定节点(例如,软件组件目录的实例(在本文称为目录实例))可以包括用于与其它节点互相通信并选择性地共享数据的软件(例如,包含生态系统注册信息的区块链的副本);用于促进事务的创建(例如,经由用于指导各种注册的完成的用户界面软件),并且用于确保与生态系统规则的一致性,从而使得能够实现的对等生态系统。

[0032] 为了本讨论的目的,对等网络或生态系统可以是分布在计算机网络上并且互相通信以促进共享处理工作负载的计算资源的任何集合,例如计算机系统和/或软件应用(即,节点)。

[0033] 注意的是,按照惯例,对等网络的对等体或节点具有相似的特权以访问由网络提供的数据和功能。但是,如本文所使用的,对等网络的对等体或节点不需要类似地享有特权。例如,一些节点(称为完整节点)最大程度地享有特权,即,维持从生态系统区块链读取

并向其写入数据的特权。其它特权较低的节点可能需要使用完整节点作为代理来访问生态系统区块链。注意的是,术语“对等网络”和“对等生态系统”在本文中可以互换使用。

[0034] 为了本讨论的目的,软件功能可以是经由计算机代码(即,软件)提供的任何功能、能力或特征,例如所存储或布置的数据。一般而言,软件功能可以经由用户界面以及随附的用户界面控件和特征的使用来访问。软件功能可以包括动作,诸如检索与计算对象(例如,与事务相关联的业务对象)有关的数据;执行与企业相关的任务,诸如推广、雇用和解雇企业人员、下订单、计算分析、启动某些对话框、执行搜索等。

[0035] 区块链可以是链接的记录(称为区块)的有序列表,其中区块链可以通过在区块链的末端添加新区块来增长,但是,除非先解开区块或从区块链中移除较晚区块,否则禁止插入较早区块。在结合到区块链后,常常会给区块链的不同区块加时间戳。可以使用例如可以被安装在给定计算环境或生态系统的节点上的分布式或联网的软件应用来实现区块链。可以经由应用于新区块的一个或多个散列的实现来实现区块之间的链接,其中该一个或多个散列充分利用或使用来自一个或多个先前区块的信息。区块链可以被用于实现事务的分布式分类账。

[0036] 为了本讨论的目的,分布式分类账可以是共享的数字数据的集合,其在网络的多个节点之间共享,其副本可以在多个节点之间复制并存储。由分布式分类账维护的数据可以在节点之间同步。

[0037] 因而,分布式分类账可以用作一种类型的分布式数据库,即,用于在耦合到实体的网络的不同实体之间存储数据的机制。节点可以是耦合到网络的任何计算机系统和/或软件应用和/或软件系统或其组。本文讨论的节点一般被称为“目录实例”,因为它们促进由随附的计算生态系统的其它节点和/或参与者访问被存储在目录中的数据。

[0038] 事务可以是描述事件、状况、属性或其它信息的信息的任何集合,这些信息描述生态系统的一个或多个方面,其中这一个或多个方面可以包括参与的开发者实体、软件组件消费者实体、贡献者实体、代理的生态系统参与者和系统、软件组件相互关系、软件组件下载和/或上传的实例、软件组件的支持状况、组件出处信息等。取决于使用术语的上下文,事务可以指描述生态系统中的活动的数据的集合,例如,开发者实体注册、命名空间注册、贡献者注册等;或者可替代地,事务可以指实际活动,例如下载组件。

[0039] 表示活动或任务的事务可以是完全自动化的,或者也可以包含人工工作流程任务,诸如手动批准或其它核实活动。因此,虽然事务可以在区块链中被表示为单个事物(例如,信息收集),但实际上可以将一些形式的事务分解成离散的子事务,这些子事务可以在处理工作流程时记录在分类账中。因而,取决于使用术语的上下文,术语“事务”还可以指传达信息(例如,计算对象)的集合的动作,并且还可以指信息(例如,计算对象)的实际集合。

[0040] 例如,如果个人软件开发者(例如,组件贡献者)向组件生态系统注册,如本文所讨论的,那么与(例如,记载)该贡献者注册过程有关的信息可以被传播到准备结合到生态系统的区块链中的目录实例的一个或多个未经核实的队列中。信息的收集和/或传送可以被称为事务,并且维护所收集的信息的计算对象也可以被称为事务,例如,开发者注册事务。参考图1,此类事务可以被收集(例如,由注册功能16)并且被选择性地提交到分布式分类账14的区块链20的一个或多个区块。

[0041] 注意的是,在某些实施方式中,区块链20和分布式分类账14可以是等效的。但是,

在本示例实施例中,分布式分类账14被示为包括区块链20,因为分布式分类账14可以包括附加功能,这与仅仅实现区块链20形成对比。例如,在某些实施方式中,分布式分类账14可以包括用于允许将区块链20转换成其它数据库格式以供参与者计算机系统12的节点本地使用的附加代码;可以允许将组件选择性地分发到目录实例数据存储库18的不同实例等等。

[0042] 在给定的计算环境或生态系统中,可以为给定的节点分配不同的特权。当与实现一个或多个特定任务有关时,为了完成任务的目的,具有相似特权的节点被称为对等体。为了本讨论的目的,对等生态系统可以是至少部分地经由一个或多个分布式或联网的软件应用实现的任何生态系统或计算环境,其中分布式或联网的软件应用经由生态系统的不同节点或对等体实现。

[0043] 经由包括软件组件目录实例的节点的对等软件组件生态系统来实现本文讨论的各种示例实施例。下面更全面地讨论的示例软件组件目录实例可以运行各种软件应用,包括用于维护和管理本地数据存储库(其可以包括软件组件的数据库)的软件;用于实现安全性和允许功能的软件;用于生成用于启用各种类型的注册的用户界面(UI)显示屏的软件(下面将更详细地讨论其示例);用于管理生态系统的未经核实的事务队列;用于与其它目录实例通信;用于维护生态系统区块链的副本;用于计算(即,核实或验证)生态系统区块链的新区块;用于提交事务以供生态系统的一个或多个目录实例进行核实(并包括在区块链块中);用于实现用于选择目录实例的任何算法以执行将一个或多个区块添加到区块链所需的计算;用于计算将区块添加到区块链所需的散列等等。

[0044] 一般而言,开发者和/或企业的社区可以使用软件组件生态系统,以使用公共技术平台与用于软件和服务的共享市场进行交互,这启用或促进信息、资源和组件的交换。

[0045] 可以将软件组件生态系统实现为可重用的软件组件的开放生态系统,以供开发者、供应商和客户使用。这种生态系统可以围绕联网的或“云”基础设施以及随附的过程和服务来构建。但是,虽然可以参考具体的处理平台、技术和基础设施来描述本发明的具体实施例,但是其它变体是可能的并且可以适用于不同的系统。

[0046] 用于软件组件生态系统的一种设计包括对各个方面的考虑,诸如信任、身份、完整性、支持、命名空间、组件身份、知识产权、产品召回、出处和许可。其它方面的特征也可以包括在内。虽然参考具体实施例描述了各种特征和方面,但是应当明显的是,具体实施例可以使用所描述的特征的不同集合。特征可以被省略或修改,并且可以包括附加特征。

[0047] 本文讨论的某些实施例提供了用于通过以下来实现软件组件生态系统的装置、存储介质和方法:接受来自用户输入设备的一个或多个信号以定义开发者实体的识别名称;接受来自用户输入设备的一个或多个信号以定义安全核实机制;在目录中记录条目,该条目包括与核实机制相关联的开发者名称;与目录条目相关联地执行事务以传送软件组件;以及使用目录条目来核实事务。附加特征允许将命名空间分配给由目录中的条目定义的实体。

[0048] 按照惯例,软件开发者可以订阅某些云服务,以促进软件应用的开发和相关联文件的存储。在本文中,被配置用于软件应用或过程流程开发的云服务被称为过程云服务(PCS)。

[0049] 过程云服务可以使用联网的数据库来存储由正在开发的给定软件程序使用的文

件和其它对象。开发者可以经由浏览器访问服务器侧开发环境。开发环境可以由PCS支持，使得所开发的软件应用文件被存储在PCS数据库中。

[0050] 为了本讨论的目的，计算环境可以是用于执行涉及计算机处理的一个或多个任务的计算资源的任何集合。计算机可以是与存储器通信的任何处理器。计算资源可以是计算环境的任何组件、机制或其能力或量，包括但不限于处理器、存储器、软件应用、用户输入设备以及输出设备、服务器等。计算资源的示例包括由一个或多个web服务、应用编程接口(API)等提供的数据和/或软件功能。

[0051] 企业计算环境可以是用于企业或组织的任何计算环境。示例企业计算环境包括分布在网络上的各种计算资源，并且还可以包括内联网Web服务器上的私有和共享内容、数据库、本地硬盘或文件服务器上的文件，电子邮件系统、文档管理系统、门户等。

[0052] 给定的软件应用可以包括(但不是必须)构成软件应用或模块(例如，服务、功能、过程、计算对象等)。因而，术语“软件应用”还可以包括联网的软件应用或其集成的组。

[0053] 本文讨论的某些实施例对于基于过程的软件应用的开发、部署和实现特别有用。基于过程的软件应用可以是可由一个或多个步骤(也称为过程元素或软件活动)序列定义的任何软件应用。术语“过程元素”、“流程元素”、“软件元素”和“软件过程元素”在本文中可以进行互换使用，以指代任何步骤，例如事件、活动、网关、子过程等。基于过程的软件应用的一系列步骤可以被称为过程流程。常常经由用户界面(UI)显示屏中的泳道对过程流程进行建模和说明。基于过程的应用常常是经由复合应用实现的，这些应用可以充分利用不同的web服务和相关联的软件组件进行不同的处理步骤。

[0054] 一般而言，软件和产品开发的流行形式可以被称为“软件生态系统”。软件生态系统可以包括协作运行并与用于软件和服务的共享市场交互的一组开发者和/或业务，以及它们之间的关系。这些关系可以包括公共技术平台，并且可以通过信息、资源和组件的交换来进行操作。

[0055] 软件组件生态系统可以是特定类型的软件生态系统，其包括软件组件的开发和交换作为主要特性。在软件组件生态系统中，可以期望构建可重用软件组件的开放生态系统，以供开发者、合作伙伴和客户使用。这种生态系统常常将围绕联网的或“云”基础设施、过程和服务构建。但是，虽然可以参考具体的处理平台、技术和基础设施来描述本发明的特定实施例，但是其它变体是可能的并且可以适用于不同的系统。

[0056] 为了清楚起见，某些众所周知的组件(诸如硬盘驱动器、处理器、操作系统、电源、路由器、互联网服务提供商(ISP)、工作流程编排器、过程调度器、过程云、业务过程管理系统、生态系统开发者实体、生态系统贡献者实体、生态系统消费者、集成开发环境、代理系统、身份管理系统、证书颁发机构(CA)等)未必在图中明确标出。但是，本领域技术人员在了解本教导后将知道要实现哪些组件以及如何实现它们以满足给定实现的需求。

[0057] 图1图示了用于选择性地共享软件组件(存储在一个或多个目录实例数据存储库18中)并用于使用分布式区块链分类账14、20来注册事务的第一示例广义对等计算生态系统10。

[0058] 注意的是，一般而言，系统10的各种模块的分组是说明性的并且可以变化，例如，某些模块可以与其它模块组合或在其它模块的内部实现，或者模块可以以其它方式被(与所示)不同地分布在网络之间或者在一个或多个计算设备或虚拟机内，而不脱离本教导的

范围。例如,虽然示例注册功能16、目录实例数据存储库18和分布式分类账14与参与者计算机系统12分开显示(为清楚起见),但在实践中,此类实体经由参与相关联的对等网络的一个或多个参与者计算机系统12实现。

[0059] 参与者系统12可以包括开发者实体(例如,具有个人开发者的公司或组织)。开发者实体可以包括所谓的贡献者,他们将代码(例如,软件组件)贡献给一个或多个所谓的目录实例,这些实例被用于实现注册功能16并使得开发者能够与在一个或多个目录实例数据存储库18中维护的软件组件交互(例如,上传、下载等)。开发者实体和相关联的开发者还可以从本地目录实例下载(即,消费)软件组件。

[0060] 从目录实例数据存储库18下载软件组件的人或实体被称为消费者。将软件组件上传到目录实例数据存储库18的个人或实体被称为贡献者。一般而言,当术语在本文中被使用时,开发者实体可以包括贡献者和消费者两者,或者仅包括贡献者。但是,仅充当软件组件的消费者的实体在本文中不被称为开发者实体。

[0061] 当开发者实体向目录实例注册(例如,使用注册功能16);并且其消费者将软件组件下载到本地目录实例数据存储库18;或其贡献者将软件组件上传到本地目录实例数据存储库18;等等时,与这些活动有关的数据一经核实就被注册在分布式分类账14和随附的经核实的区块链20中。

[0062] 注意的是,在某些实施例中,开发者实体的注册生成所谓的开发者实体注册事务。开发者实体注册事务后可以跟命名空间注册(可替代地,在某些实施方式中,它可以是开发者实体注册的一部分)。在命名空间注册期间(例如,使用注册功能16),命名空间与开发者实体相关联,并且对应的信息作为命名空间注册存储在分布式分类账14中。

[0063] 在开发者实体注册和相关联的命名空间注册之后,个人贡献者和消费者可以随后与相关联的开发者实体和命名空间相关联地注册。当贡献者将已经注册的软件组件上传到另一个目录实例(即,除了开发者实体直接与之通信的本地目录实例之外的目录实例)时,可以使用分开的上传事务。

[0064] 各种其它类型的注册也是可能的,包括但不限于组件下载事务、组件使用注册等。一般而言,与每种注册类型有关的信息在核实之后被提交给分布式分类账14的区块链20,如下面更全面地讨论的那样。

[0065] 分布式分类账14允许将管理(例如,节点管理、身份管理等)任务移交到目录实例(例如,由图1中的项14-18共同表示)和/或本地身份域,同时使参与者能够访问与相关联的生态系统10有关的事务的一致视图或摘要,即,经由在分布式分类账14的区块链20中注册的事务。

[0066] 在本示例实施例中,根据与向其提交的事务的实体和/或注册者相关联的角色来允许分布式分类账14内的事务。根据用于事务的节点(例如,目录实例)的性质,可以进一步允许事务。例如,一些节点可以是完整节点,而其它节点可以具有减少的特权来访问和参与核实区块链20的区块。

[0067] 可以被不同地允许的示例角色包括社区领导者角色(可以负责开发者实体身份核实;核实对开发者实体的命名空间分配,等等);开发者角色(可以负责注册和/或管理开发者实体身份、贡献者、组件身份等);分发者角色(可以负责组件使用注册);消费者(可以参与组件使用注册)等等。

[0068] 注意的是,在分布式系统和对等系统(诸如生态系统10)中,事务注册和相关联的数据注册可以是重要的。数据(例如,与注册和相关联的软件组件有关的数据)存在于多个位置,即,数据在参与者计算机系统12的完整节点之间复制。

[0069] 因而,如果数据(例如,事务数据)被一个节点改变,那么将改变(和相关联的事务)传播到参与者计算机系统12的所有完整节点可以是重要的。如果事务数据与软件组件注册有关,那么将实际的整个软件组件存储在提交给分布式分类账14和随附的区块链20的事务中可以是不切实际的。因而,可以在区块链20中注册用于存储给定软件组件的文件(即,分发介质)的散列,即,用于软件组件注册。

[0070] 图2是图示图1的第一示例对等计算生态系统10的示例实现30的更详细示例图。参考图1和2,图2的示例实现30被示为包括开发者实体系统22和外部代理的系统50,其包括图1的参与者计算机系统12。

[0071] 开发者实体系统22与实现分布式分类账14的参与者系统(例如,图1的系统12)的云54通信。在本示例实施方式30中,云54还包括一个或多个身份域44,其可以被用于在使得能够访问分布式分类账14和随附的区块链20(例如,经由目录实例32)之前来认证开发者实体和个人用户,以用于读取提交的事务并用于将事务提交到区块链20。

[0072] 注意的是,在某些实施方式中,目录实例32也可以被认证和允许(用于与其它目录实例的选择性交互以及与预定允许相关联的预定功能集的选择性实现)。

[0073] (一个或多个)示例开发者实体系统22表示可以参与随附的对等网络和相关联的生态系统30的各种操作的开发者实体(例如,企业或组织)的一个或多个计算系统。为了本讨论的目的,开发者实体可以表示一个或多个人以及随附的计算系统和计算资源的组,可以使用目录实例32中的一个或多个将其注册为这样的实体。在本示例实施例中,开发者实体与从目录实例32的本地组件目录36上传软件组件(贡献)和下载(消费)软件组件的特权相关联,例如,经由与开发者实体系统22的开发者实体相关联地注册的个人贡献者和/或消费者。

[0074] 开发者实体(例如,公司)与命名空间(例如,可以包括公司的名称)相关联。给定的命名空间可以包括名称的层次结构,名称的层次结构可以包括与开发者实体的名称相关联的各种软件组件的名称(其可以是给定开发者实体的分层命名空间中的最高级别节点)。

[0075] 注意其它类型的实体;在不脱离本教导的范围的情况下,不仅开发者实体和相关联的系统22可以参与生态系统30。注意的是,虽然在图2中仅示出了开发者实体系统22和外部代理的系统50(由一个或多个开发者实体系统22代理以使用一个或多个目录实例),但是其它类型的系统也可以参与生态系统30。

[0076] 在本示例实施例中,开发者实体系统22包括托管生态系统30的完整节点的一个或多个计算系统(例如,实际计算机和/或虚拟机)。完整节点由目录实例32表示。为了本讨论的目的,完整节点可以是具有完全读取、写入和提交特权的任何节点,以用于访问数据并将数据(例如,事务数据)提交到区块链分布式分类账14。

[0077] 示例目录实例32包括分类账副本34,它是被复制或传播到随附的生态系统30的其它节点(目录实例)的区块链分布式分类账14的副本。分类账副本包括未经核实的队列38,该队列可以包含一个或多个尚未提交给区块链副本40的未经核实的事务。注意的是,本地未经核实的队列38和区块链副本40的数据在其它节点之间共享或以其他方式传播到其它

节点,之后,它们成为云54的分布式未经核实的队列46和相关联的区块链分布式分类账区块链20的代表。

[0078] 示例目录实例32还包括本地组件目录36或与本地组件目录36通信。本地组件目录36维护已上传到目录实例32并且可以由本地组件目录36中组件的有特权消费者下载的软件组件。与本地组件目录36的每次交互都伴随有记载该交互的相关联的事务,如下文更全面地讨论。

[0079] 首先将给定的事务加载到未经核实的队列38中;传播到其它目录实例(节点)的其它未经核实的队列46;然后,一旦执行了事务有效性检查,就提交给总体代表性的分布式区块链20。事务提交过程涉及使用一个或多个目录实例32来计算事务集合的一个或多个散列,以及事务集合将被提交到的区块链20的相关联区块的一个或多个散列。从未经核实的队列38、46中剔除要提交给区块链40、20的事务集合。

[0080] (一个或多个)开发者实体系统22还包括与一个或多个目录实例32通信的一个或多个注册员计算系统42。在本示例实施例中,注册员系统42可以是被授权(或由已被授权的人使用)向一个或多个目录实例32提交注册的任何计算系统。注意的是,在开发者实体系统22的开发者实体的一个或多个计算系统上运行的目录实例32在它们是开发者实体系统22的开发者实体本地的程度上被称为本地目录实例。

[0081] 注意的是,注册员系统42的注册员可以通过一个或多个基于云的身份域44被认证为代表开发者实体系统22的开发者实体充当注册员。注册员可以维护(一个或多个)开发者实体秘密(例如,当使用公钥基础设施(PKI)时的私钥)。

[0082] 注册员系统42可以充当个人开发者系统24(例如,由个人软件组件贡献者和/或消费者使用的计算系统)的代理(例如,出于事务注册的目的)、其它贡献者系统26的代理(例如,代理的上传器)以及其它用户系统28的代理(例如,代理的下载器),它们是开发者实体系统22的开发者实体的一部分。

[0083] 此外,在某些实施方式中,注册员系统42还可以充当外部代理的系统50的代理(或多个代理),外部代理的系统50不一定是开发者实体系统22的一部分。例如,在一些情况下,可以授权外部集成开发环境(IDE)使用开发者实体的注册员系统42作为与(一个或多个)目录实例32和(一个或多个)随附的本地组件目录36中的一个或多个进行交互的接口。

[0084] 在示例用例场景中,开发者实体(例如,开发者实体系统22的所有者)最初在称为开发者实体注册的过程中向目录实例32注册为开发者实体,如下面参考图4-5更全面地讨论的。然后将相关联的开发者实体注册信息打包为事务并提交到本地区块链分类账副本34的未经核实的队列38,以准备对事务进行有效性检查并最终将其作为对本地区块链40和随附的分布式区块链20的经核实的的事务进行提交。

[0085] 开发者实体注册之后可以跟各种附加类型的注册,例如,开发者实体命名空间注册,由此注册开发者实体系统22的开发者实体的名称(和/或其它命名空间)。

[0086] 接下来可以进行个人贡献者(例如,开发者实体系统22的开发者实体的个人开发者)的注册。贡献者与注册的开发者实体名称和/或(由贡献者)附加请求的命名空间相关联地向区块链20注册。还可以提供用于特定贡献者的公钥和数字签名(例如,如可以经由身份域44和随附的证书颁发机构(CA)获得的),以促进确认注册的真实性,即,贡献者被授权并正确地允许以使用目录实例32和本地软件组件目录36提供的随附软件组件功能(例如,下

载和上传软件组件)。

[0087] 其它类型的注册和相关联的事务包括软件组件注册(例如,如可以在准备将软件组件上传到本地组件目录36时发生的);消费者组件下载注册与可选的组件使用注册结合;以及组件使用注册。

[0088] 另外,在不脱离本教导的范围的情况下,其它类型的注册也是可能的。例如,注册跟踪软件组件的嵌套使用(例如,当一个软件组件在另一个软件组件内使用时);本地组件目录状态信息的注册(例如,本地组件目录36中可用软件组件的列表)和所提供的功能等等。

[0089] 一般而言,通过在区块链分布式分类账14中收集各种类型的注册信息,可以为整个生态系统30带来许多好处。例如,本文讨论的实施例的使用可以促进避免软件组件之间的名称冲突;组件支持服务的高效管理,包括确保由第三方供应者支持特定组件,如何支持该组件以及如何选择性地召回该组件;检测组件篡改;加强知识产权并跟踪组件的出处和所有权;检查特定组件的软件许可是否有效;一般而言使得能够核实组件的有效性;等等。这些在不需要常规的中央控制机制的情况下就可以全部实现,该机制可以经由使用如本文所讨论的分布式目录实例32基本上被代替和增强。

[0090] 图3示出了充当图1-2的对等计算生态系统10、30的对等体或节点的示例个人开发者计算系统72的示例细节,包括示例用户界面(UI)显示屏70的示例UI控件和选项,以及可以包括在目录实例32中的示例模块34-40、60-68,该目录实例32被用于将各人开发者计算系统72耦合到对等生态系统(其可以由云14结合开发者系统72来表示),由此个人开发者计算系统72和随附的目录实例32充当整个对等生态系统的节点。

[0091] 注意的是,出于说明性的目的,示出了目录实例32在个人开发者系统72上运行。但是,在实践中,个人开发者系统72可以由图2的开发者实体系统22的开发者实体拥有或以其它方式与之相关联。因而,个人开发者系统72可以表示图2的开发者实体系统22之一。

[0092] 在本示例实施例中,个人开发者系统72既充当随附的对等生态系统的节点,又充当个人软件组件贡献者和/或消费者的系统。注意的是,在本实施例中,出于实现事务注册的目的,开发者系统72的个人开发者还充当注册员,其可访问目录实例32和相关联的功能;不仅是软件组件到本地组件目录36的下载和/或从本地组件目录36的上传操作。

[0093] 示例开发者系统72包括与UI显示屏70通信的目录实例(本文称为第一目录实例,或目录实例1)32。目录实例32还与在托管随附的对等生态系统的网络或云14上运行的分布式区块链分类账46、20通信。

[0094] 为了本讨论的目的,UI显示屏可以是在显示器上呈现的任何软件生成的描绘。描绘的示例包括窗口、对话框、显示的表以及任何其它图形用户界面特征(诸如用户界面控件),这些是经由软件(例如,浏览器)呈现给用户的。包含在单个边界内的用户界面显示屏被称为视图或窗口。视图或窗口可以包括区段,诸如子视图或子窗口、对话框、图形、表等。在某些情况下,用户界面显示屏可以指当前显示在显示器上的所有应用窗口。

[0095] UI控件可以是用户界面显示屏的任何显示的元素或组件,其适于使用户能够提供输入、查看数据和/或以其它方式与用户界面交互。用户界面控件的其它示例包括按钮、下拉菜单、菜单项、点击并按住功能等等。类似地,用户界面控制信号可以是被提供作为软件输入的任何信号,其中输入影响用户界面显示屏和/或与该软件相关联的随附软件应用。

[0096] 示例目录实例32包括本地分类账副本34,该本地分类账副本34包括分布式区块链20的复制的(在其它目录实例或节点之间复制的)未经核实的队列38和区块链副本40。目录实例32还包括本地组件目录36(例如,可以经由关系数据库或其它类型的数据库实现)、UI代码生成模块60、分类账接口模块64、安全性和允许模块62,以及区块计算模块66。目录实例控制器68促进目录实例32的各种其它模块34、36、60-66之间的接口、控制和路由互相通信。

[0097] 示例UI显示屏70以及随附的UI控件和选项包括用于以下的UI控件:促进贡献者和消费者注册;组件注册;在本地组件目录36中可用的组件搜索和/或软件组件查看;核实组件完整性、组件标识等;核实软件组件的来源、出处、许可条件、支持状况等;开发者实体命名空间注册;贡献者注册;组件上传和/或下载注册;解除软件组件的注册的UI控件或选项等等。

[0098] 注意的是,经由示例UI显示屏70可用的UI控件的确切数量和类型是特定于实施方式的,并且可以取决于给定实施方式的需要而变化,而不脱离本教导的范围。此外,可以由多个UI显示屏(例如,以向导的形式)来实现UI控件和选项70中的一个或多个。

[0099] 在本示例实施例中,基于来自目录实例32的UI代码生成模块60的渲染指令来生成UI显示屏70的各种UI控件和选项。

[0100] 鉴于由个人开发者系统72的开发者实体建立的规则,安全性和允许模块62可以促进个人用户和开发者实体的认证和允许。安全性和允许模块62还可以与图2的一个或多个身份域44通信,以促进实现各种安全性和允许特征。注意的是,由安全性和允许模块62执行的确切功能是特定于实施方式的,并且可以根据给定实施方式的需求而变化。

[0101] 目录实例32的区块计算模块66包括,当目录实例32被选择(例如,根据一致性算法或通过轮询选择)以计算(即,核实)事务的下一个区块以将其包括在分布式分类账和随附的计算云和生态系统14的分布式区块链20中时,用于促进针对本地未经核实的队列38中的未决事务的散列的计算以及用于计算区块链40、20的区块的散列的代码。

[0102] 分类账接口模块64包括用于促进将进入未经核实的队列38的新事务传播到其它目录实例(即,生态系统的节点)的未经核实的队列46的代码。分类账接口模块64还包括用于复制分布式区块链20并维护区块链20的最新副本40的代码。

[0103] 本地组件目录36维护可以由授权的消费者选择性地下载的软件组件。可以由被授权且注册的贡献者与注册的开发者实体命名空间和对应的注册的开发者实体相关联地将软件组件上传到本地组件目录36。

[0104] 图4示出了第二示例系统80和用于向图1-3的分布式区块链分类账20注册开发者实体以及随附的开发者实体注册事务84的随附的目录实例过程94,其中第二示例系统80可以根据图1-3的实施例来实现。

[0105] 在示例系统80中,例如与图2的(一个或多个)注册员系统42的用户对应的注册员82使用注册员系统42和图3的随附的UI显示屏和选项70来提交开发者实体注册事务84。开发者实体注册事务84包括开发者实体名称86(例如,与图2的开发者实体系统22的开发者实体对应),开发者实体的公钥88和开发者实体的数字签名90。

[0106] 注意的是,数字签名90可以根据公钥基础设施(PKI)来实现,并且可以包括被注册的开发者实体的私钥。数字签名促进确认注册员82代表被授权的(例如,由身份域和随附的

证书颁发机构,例如图2的身份域44)开发者实体提交注册,即,被授权向区块链分类账20和随附的分布式生态系统注册。

[0107] 在本示例实施例中,注册员82将开发者实体事务84提交到第一目录实例(例如,图3的目录实例32)。然后,在将事务84传播到其它目录实例110、112的未经核实的队列之前,第一目录实例32运行初始开发者实体注册过程94,以最终由所选择的目录实例(例如,第一目录实例32、第二目录实例110或第三目录实例112)提交114到分布式区块链20。所选择的目录实例(例如,第三目录实例112)可以采用其区块链计算模块(例如,图3的模块66)来计算事务散列和区块散列,以便将事务84提交到分布式区块链分类账20。

[0108] 初始开发者实体注册过程94包括各种步骤98-108,作为开发者实体注册事务84的预处理形式,然后将事务84提交到目录实例,以便随后事务向分布式区块链分类账20的提交。初始签名核实步骤98涉及例如参考身份域(例如,图2的身份域44)的证书颁发机构(CA)来对数字签名90进行核实。

[0109] 在数字签名90通过核实之后,过程94检查分布式区块链20和/或任何随附的未经核实的队列38,以寻找相同开发者实体名称86的任何现有开发者实体注册。如果先前已经注册了任何其它类似的开发者事务(或者在当前未经核实的队列中以其它方式存在),那么由于冲突检测而中止当前开发者实体注册过程94。

[0110] 随后,可选社区领导者核实步骤102包括用(生态系统的)可选社区领导者进一步检查该社区领导者是否反对向分布式区块链20注册开发者实体事务84。出于社区领导者已被授权(或允许)阻止事务的任何原因,该社区领导者可以可选地阻止开发者实体注册事务84。

[0111] 在可选社区领导者核实步骤102之后,例如经由事务创建步骤104,对注册事务84进行打包和散列以包括在未经核实的队列中。注意的是,如果步骤98-102中的任何一个未能完成,那么开发者实体事务注册过程94被中止。

[0112] 在开发者实体注册事务84被提交到本地未经核实的队列38之后,在传播步骤108中,它被传播到其它目录实例110、112以及随附的未经核实的队列(例如,图3的队列46),准备用于提交处理114,由此处理未经核实的队列38以及对应的副本(例如,图2的副本46)中的一个或多个未决事务的集合,用于作为分布式区块链20的一个或多个区块来提交。

[0113] 注意的是,在本示例实施例中,给定目录实例32、110、112可以已经可以访问系统80的每个用户和参与者实体的身份信息,因为实体是经由一个或多个云身份域(例如,图2中的域44)供应的。

[0114] 注意的是,在某些实施方式中就其可以被省略的程度,使用图2的云身份域44是可选的。例如,可替代地,可以通过除云身份域之外的其它手段来识别用户和实体。此外,在其它实施例中,某些用户可以保持身份不明或匿名。

[0115] 不过,在本示例中,每个用户和实体与图2的身份域44内的预定身份相关联。

[0116] 本示例实施例区分并关联与个人开发者角色相关联的允许和与表示个人的组织的开发者实体相关联的允许。

[0117] 可能有多个个人可以为单个实体(例如,公司)开发组件,照此,多个个人的身份将被映射到特定开发者实体身份或(跨一个或多个身份域)与之相关联,这可以用于软件组件所有权和有效性核实目的。

[0118] 不过,分布式区块链20记录开发者实体的标识符与公钥之间的映射或关联。在本示例实施例中,存储的身份表示开发者实体的身份。将软件组件写入和/或上传到目录实例(并向用区块链20注册组件和上传)的特定个人称为贡献者。

[0119] 因而,开发者实体注册过程94表示开发者实体的自我注册。注册员82可以是访问系统80并且希望注册开发者实体的任何用户。

[0120] 因此,示例开发者实体注册过程94(其可以取决于特定实施方式的需求而变化)包括在可选地认证了注册员82之后采用“创建开发者实体”UI和随附的UI控件(例如,如可以是包括在图3的UI控件和选项70中)以供应关于开发者实体的信息84,例如至少识别名称,但可能包括联系信息等。此外,注册员82还供应公钥88,并以数字方式对事务进行签署,例如,导致带有要求保护的密钥(例如,开发者实体私钥)的数字签名90。

[0121] 然后,目录实例过程94使用供应的公钥来核实签名90(在步骤98处),从而证明注册员82拥有秘密的两半(例如,与正被注册的开发者实体相关联的公钥和私钥)。随后,(在步骤100处),目录实例32检查相关联的目录数据存储库(例如,图3的本地组件目录36和/或图3的本地未经核实的队列38和/或本地区块链副本40)以确保开发者实体尚未定义或正在被定义。

[0122] 可选地,社区领导者核实步骤102核实与开发者实体事务84对应的声称(claim)。例如,在某些实施方式中,在将事务84提交到分布式区块链20之前,可能要求附加的手动步骤(例如,要签署的文书工作)以完成开发者实体注册预处理。最后,注册事务被创建和/或格式化(在步骤104处)并且被传播(在步骤108处)到其它目录实例110、112。

[0123] 在传播之后,给定的注册事务将位于未经核实的队列的副本中(例如,图3的未经核实的队列46),直到提交到分布式区块链20。所创建的事务作为不可信的(即,未经核实的)事务被传播到生态系统的所有其它完整节点(例如,目录实例110、112)。未经核实的队列38可以使得能够将未经核实的任务放入争议,直到未经核实的队列的未经核实的任务被提交到区块链20。

[0124] 在将开发者实体事务84提交到分布式区块链20之后,相关联的开发者实体被认为足够可信,以使得能够发起依赖开发者实体的身份的附加事务,如在开发者实体注册事务84中所维护的现在被提交给分布式区块链20的附加事务。

[0125] 总之,图4图示了系统80中为开发者实体的身份(即,开发者身份)创建分类账条目的基本组件。

[0126] 系统80示出了使用计算机系统42和随附的目录实例用户界面来定义开发者实体的注册员82,如开发者实体事务信息84所指示的。可以可选地在这个过程之前对注册员82进行认证。

[0127] 注册员系统42的用户界面包括UI控件和选项,从而允许注册员82供应关于开发者身份的信息86-90,诸如识别名称86和安全机制(例如,公钥88和数字签名90)。

[0128] 注意的是,注册员82可以是个人、业务实体、自动化过程等。当人是注册员82时,可以经由注册员系统42(可以是台式个人计算机、笔记本电脑、平板电脑、智能电话或其它设备)访问用于实现开发者实体注册的用户界面。

[0129] 在不同的实施例中,可以通过在网络上耦合的诸如服务器、云服务等的远程处理器来全部或部分地处理用户界面和随附的注册系统42。在注册员的功能是自动执行的情况

下,那么过程可以使用任何合适的处理硬件和软件来执行功能。一般而言,除非另有说明,否则本文描述的功能可以通过位于不同地理区域的硬件和软件的任意组合并在不同时间点来执行。

[0130] 构成开发者实体事务84的信息86-90通过请求被传送到第一目录实例32,以实现开发者实体注册过程94(在本示例性实施例中也称为目录过程)。

[0131] 如目录过程84内所示,步骤98-108包括使用供应的公钥88来核实签名90;检查现有的开发者实体(以防已经定义了相同的实体),并可选地让社区领导者执行核实(例如,手动签署文书工作)。这些步骤中的任何一个失败都会中止该过程。否则,事务84被格式化并提交以合并到未经核实的队列38中。

[0132] 目录过程94可以距主题注册员82、开发者实体或其它行动者远程执行。例如,可以在服务器计算机、云服务、对等位置的个人计算机系统上执行目录过程94和其它目录过程的功能。

[0133] 附加目录实例110、112说明了第一目录实例32的对等体的存在。在步骤102中格式化的事务84以未经核实的状态被传播到它们110、112。最终,当队列38内存在足够的未经核实的事务时,这些目录实例32、110、112之一将核实事务并将其提交到分类账(作为区块链20的区块)。然后,经核实的事务(即,在这种情况下对开发者实体的定义)将经由它们全都具有读取访问允许的分布式区块链20而对所有目录实例32、110、112可用。一旦事务被提交,所有目录实例32、110、112就将阻止创建重复的开发者实体注册(即,事务)。在一些实施例中,即使初始事务处于未经核实的状态,目录实例32、110、112也可以防止重复。

[0134] 一旦开发者实体已经在生态系统80上注册并通过核实,它就也将被传播到所有完整节点32、110、112。此后,可以将一个或多个命名空间分配给那个开发者实体。命名空间注册过程可以类似于初始开发者实体注册过程,因为它可以由拥有开发者实体秘密的任何个人发起。

[0135] 目录条目和实例的对等传播和核实可以类似于虚拟货币实现中使用的共享分类账方法。但是,与虚拟货币系统相反,由于预期的软件组件生态系统80包括如果期望的话可以被认证的已知实体(例如,通过图2的示例身份域44),因此不需要使区块的生成成为计算密集的。因此,不需要耗时的计算要求。

[0136] 图5是第一示例方法120的流程图,该第一示例方法120可经由图4的第二示例系统和随附的生态系统80实现,并且也可经由图1-3的相关生态系统实现。

[0137] 第一示例方法120包括附加信息提交步骤122,其包括例如经由图4的注册员82向目录实例(例如,图4的目录实例32)提交开发者实体名称、公钥和数字签名。

[0138] 接下来,在核实步骤124核实所提交的信息(例如,与图4的开发者实体注册事务信息84对应),并检查该信息是否表示区块链和/或未经核实的队列中的重复条目。如果这两个检查124、126都通过,那么相关联的开发者实体注册事务在步骤128处被放置在未经核实的队列上,并在步骤130处被传播到其它目录实例;否则,事务注册过程中止。

[0139] 在将事务传播到其它目录实例(即,节点)之后,节点可以共同确定是否满足正将事务传播到其它目录实例的目录实例(例如,图4的目录实例32)的对等资格。如果执行注册的目录实例不满足预定的对等资格(可以取决于给定实施方式的需求而变化),那么注册过程失败。否则,在步骤134处,从可用目录实例当中选择目录实例,以执行将事务提交到分布

式区块链分类账(例如,图4的区块链20)所需的计算(例如,散列的计算)。

[0140] 用于选择给定目录实例以执行计算以将一个或多个事务的组从经核实的队列传送到分布式区块链(例如,图4的区块链20)的算法或方法是特定于实现的,并且可以取决于给定实施方式的需要而变化。例如,在某些实施例中,选择第一可用目录实例(或具有最多可用计算资源的实例)作为目录实例以执行计算;在其它实施方式中,目录实例可以对目录实例进行多数表决,以确定哪个目录实例应当执行计算;在还有其它实施方式中,目录实例可以轮流提交区块(经由轮询方法)等等。

[0141] 总之,图5表示可由图4的系统80实现的方法120的流程图。执行步骤122以接收开发者实体名称、公钥和数字签名,这些被共同提出作为要提交到图4的分布式区块链分类账20的新事务。

[0142] 在步骤124处,进行检查以核实数字签名。如果核实失败,那么该过程以失败条件终止,并且事务未提交到分类账。否则,如果签名通过核实,那么执行前进到步骤126,在那里检查重复的实体(例如,先前使用的名称)。如果检查126失败,那么过程120如前所述中止。否则,执行步骤128以将事务放置在队列上。接下来,在步骤130处,使事务可以用于网络上的其它对等过程(即,该对等过程是对等组件生态系统的一部分)。

[0143] 步骤132和134根据底层分类账的规则来核实包含事务的区块;然后通过已向其提交事务的分布式区块链分类账(例如,图4的区块链20)跨所有目录实例提交并共享事务信息。

[0144] 图6示出了示例UI显示屏140,其可以经由UI显示屏70显示在诸如图3的用户开发者计算系统72之类的计算系统上。示例性UI显示屏140包括各种UI控件和选项144-180,用于浏览软件组件的目录、从目录条目/选项144中搜索(例如,经由搜索按钮146)、下载(例如,经由下载按钮148)、编辑(例如,经由编辑按钮150)以及删除(例如,经由删除按钮152)软件组件156、158、186、188,这些软件组件由随附的计算系统访问的目录实例(例如,图3的目录实例32)维护。

[0145] 在本示例实施例中,目录144中的软件组件由一个或多个对应的组件卡156、158、186、188指示,其被划分为特征组件160和私有组件180。

[0146] 每个软件组件都包括组件名称162、172、192、202,对应的文件名164、174、194、204及描述166、176、196、206。用户可以选择给定的组件UI卡156、158、186、188。右击卡可以揭示附加的用户选项,例如,以深入了解关于给定组件的更多信息并访问附加用户选项和控件,以与例如下载组件进行交互。注意的是,与右击卡相反,在不脱离本教导的范围的情况下,也可以使用诸如选择“详细按钮”等其它UI功能可见性(affordance)。

[0147] 总之,示例UI显示屏140表示目录UI的示例部分。UI显示屏140示出了包括关于软件组件的信息的各种示例UI卡或缩略图156、158、186、188。

[0148] 注意的是,每个组件卡156、158、186、188可以示出附加信息,例如可以包括图形信息或图像(例如,该组件生成的UI显示屏的图像)、作者信息、命名空间指定、组件的更详细的描述、给定组件的状况(例如,支持状况或在开发周期中的位置)、组件已被下载的次数、组件是否是演示等等。在其它实施例中,可以使用表示组件的不同方式,以及提供其它类型的信息。

[0149] 对于示例组件UI卡156,名称被示为“ora-card-layout”。在这个示例方案中,名称

的第一部分“ora”被称为基本“命名空间”，可以从其追加附加文本以形成完整且唯一的组件名称。

[0150] 注意的是，在一些系统中，用于组件的命名空间也可以记录在组件元数据中的其它位置，以及或代替图6中所示的组件名称前缀方法。不过，在这两种方法中，命名空间和组件名称（对于关联的生态系统中的每个组件最终都是唯一的）都在示出本地软件组件目录的可用软件组件的给定的UI显示屏中示出。在一些实施例中，除了命名空间和组件名称之外，生态系统可以还分配合成身份，例如，具有128位的通用唯一标识符（UUID），以便为任何给定的组件定义唯一身份。

[0151] 每个开发者实体都被指派了基本命名空间，其可以是可以根据开发者的选择向其追加附加描述符或名称的一个或多个项。名称“ojx”（例如，与组件UI卡186的文件名194对应）是图6所示的组件中的基本命名空间指示符的另一个示例。注意的是，在图6的示例中，基本命名空间仅为3个字符，但是其它实施例可以使用任何数量和类型的指定。

[0152] 注意的是，各种组件UI卡156、158、186、188可以示出与和每个识别出的组件相关联的区块链注册信息有关的附加信息。例如，还可以查看已经核实并提交到区块链分类账的信息，例如组件名称、命名空间、所有者、状况和供应组件所依据的许可信息。

[0153] 在本示例实施例中，目录UI将仅显示经核实（诸如通过具有匹配在随附的区块链事务中那个组件的对应注册的校验和的校验和）的用于下载的软件组件的UI卡。

[0154] 注意的是，如果用户执行例如下载组件的操作，那么下载过程还可以涉及针对那个组件的下载注册过程的完成。此外，注意的是，除了示例UI显示屏140中示出的UI控件和选项之外，附加的UI控件和选项也是可能的。例如，可以提供用于上传组件、发起各种类型的区块链注册事务等的附加选项。

[0155] 图7是图示第三示例系统210（表示软件组件生态系统）的第一框图，其图示了由第三示例系统210表示的对等软件组件生态系统的称为目录实例的互相通信的节点214-218所维护的区块链212、230的初始状态。图7图示了系统210的示例初始状态，其包括区块链（表现为区块链的区块212的分布式副本230-236）和空根区块212（区块0）。

[0156] 在本示例实施例中，三个目录实例214-218表示在不同计算机系统和/或虚拟机上运行的软件，其中计算机系统和/或虚拟机经由网络互相通信。目录实例214-218可以经由一个或多个云（例如，一个或多个互相通信的服务器的集合）来实现。

[0157] 目录实例214-218中的每一个以初始状态示出，具有未经核实的队列224-226。为了本讨论的目的，未经核实的队列可以是用于在计算对象（例如，与一个或多个事务对应的计算对象）被处理并被包括在区块链230-236中之前存储计算对象的任何数据存储机制。注意的是，区块链230-236包括区块链的各个副本，其在目录实例214-218之间共享或复制。

[0158] 区块链230-236的内容212的共享可以由每个目录实例214-218所实现的软件来实现，其中该软件包括用于使目录实例214-218之间能够进行通信的计算机代码，包括在相应区块链230-236之间复制区块链内容212。注意的是，区块链230-236当前仅包括初始根区块（也称为初始终端区块，块0或第零个区块）212。初始终端区块212可以最初没有事务，否则可以将其存储在区块212的有效载荷区段220中。

[0159] 注意的是，本文图示并在下面更全面地讨论的事务有效载荷、事务散列、事务公钥、区块散列、区块有效载荷等仅是说明性的并且可以变化，而不脱离本教导的范围。此外，

仅可能事务的小例子集在本文并在下面更充分地图示和讨论。能够获得本教导的本领域技术人员可以容易地实现并使用本文讨论的实施例来收集、核实并存储在区块链中,以供各种生态系统节点进行选择访问,以满足给定实施方式的需求,而无需进行过多的实验。

[0160] 初始终端区块212包括具有用于先前区块的散列(称为先前散列)的字的散列区段222,在当前用例中该散列为零。散列区段222还包括当前散列(也称为区块链的起始散列),出于说明目的,其被表示为#2ab4。注意的是,散列符号“#2ab4”中的“#”可以表示较长散列值的通配符或占位符,散列值可以是SHA-256(安全散列算法-256)或其它类型的散列。在不脱离本教导的范围的情况下,所使用的散列的确切类型可以是特定于实施方式并且可以变化。

[0161] 图8是图示在第一示例事务(即,开发者注册事务)240已经经由在本示例用例中是第一目录实例214的原始目录实例(例如,第一目录实例)被传播到目录实例214-218的未经核实的队列224-226之后的图7的第三示例系统210的第二框图。

[0162] 注意的是,给定目录实例214-218可以与已经被认证(例如,经由公钥基础设施(PKI))以访问给定目录实例214-218的开发者实体(或其它实体)的开发者计算机(称为开发者系统)通信。目录实例214-218可以运行软件以促进生成UI显示屏,以使软件组件的开发者、贡献者和/或消费者能够向已被认证的特定目录实例214-218上传和/或从特定目录实例214-218下载软件组件,并且符合与初始身份核实和向目录实例214-218注册期间提供的凭证相关联的特权。

[0163] 注意的是,给定开发者(和/或贡献者、消费者等)实体(例如,公司)可以使用注册员实体将特定的开发者用户和相关联的计算机系统与一个或多个目录实例214-218进行接口连接。与目录实例214-218进行通信以例如下载和/或上传事务;发起各种类型的注册等等的开发者和相关联的计算机系统可以充当开发者所属的开发者实体的代理。

[0164] 如图8所示,在本示例用例中,第一目录实例214已接收到要在区块链230-234中注册的第一事务240。第一目录实例214包括计算机代码,用于促进创建并在第一未经核实的队列224中存储第一事务240,然后将第一事务240的拷贝或副本传播到生态系统节点214-218的其它未经核实的队列226-228。

[0165] 第一示例事务240包括第一事务有效载荷区段250,其包括类型信息(例如,指定“注册开发者”事务类型)和开发者实体信息(例如,“example.com”)。在不脱离本教导的范围的情况下,第一事务240的事务有效载荷区段250中存储的数据仅仅是说明性的并且可以变化。

[0166] 此外,在不脱离本教导的范围的情况下,来自事务240的其它区段(例如,散列区段252)的一个或多个数据项可以包括在有效载荷区段250中,和/或反之亦然。第一示例事务240的散列区段252包括第一事务散列(例如,#177e)和与第一事务240的开发者实体注册者(registrant)相关联的公钥(例如,#e56c)。

[0167] 在创建第一开发者注册事务240之后,第一目录实例214包括对等生态系统软件和区块链接口软件,以传播第一事务240以初始存储在其它目录实例216、218的未经核实的队列226、228中。注意的是,取决于网络等待时间等,其它节点216、218不一定必须同时接收第一事务240。

[0168] 图9是第三框图,图示了在由(根据选择规则或算法)已被选择来处理(例如,核实

或验证) 开发者注册事务240的目录实例(例如,第二目录实例216) 进行处理(即,核实初始开发者注册事务240和向区块链230-236提交)期间的图8的第三示例系统210。

[0169] 注意的是,用于确定哪个目录实例214-218将负责处理和核实事务240以结合到区块链230-236的区块中的选择规则或算法可以是特定于实施方式的,并且可以取决于给定的实施方式的需求而变化。例如,在某些实施例中,选择第一可用目录实例214-218进行处理;在其它实施例中,单个目录实例214-218专用于处理所有事务以结合到区块链区块中;在其它实施例中,所有目录实例214-218同时处理事务,并且首先完成处理的目录实例214-218将结果所得的经核实的区块提交到区块链230-236;在其它实施例中,可以根据与每个目录实例214-218相关联的可信赖度得分或度量来分配完成区块的处理所需的处理和工作量;在其它实施例中,采用使目录实例214-218能够确定哪个目录实例214-218应该处理一个或多个事务的给定集合的共识或表决机制,等等。

[0170] 在第一开发者注册事务240的处理(在本文中称为事务核实或验证)期间,所选择的目录实例216最初为其中要结合第一事务240的区块计算树散列。树散列(例如,Merkle树)包括所有未决事务(在这种情况下,其仅包括第一开发者注册事务240)的散列。结果所得的树散列(例如,#DD34)结合在为第一开发者注册事务240创建的区块中,如下文更充分地讨论的。

[0171] 总之,基于由目录实例实现的可插入核实功能,指派一个或多个节点(目录实例214-218)以汇总任何未解决的事务,以准备将(一个或多个)事务提交到区块链230-236中的新区块。

[0172] 关于涉及多少个节点的决定可以因实现而异。在所有节点都是受信任的简单实施方式中,单个节点可以负责事务核实。在其它实施方式中,多个节点可以以同时的方式执行核实,并且对结果进行比较以达成共识。事务验证过程和区块创建可以基于底层区块链实现的细节而有所不同。在本示例实施例中,不需要选择性地改变用于计算散列值(例如,T1散列#DD34)的难度级别。但是,注意的是,可以根据特定生态系统内的总体信任级别和/或根据与被指派执行事务验证的个人目录实例214-218相关联的信任级别来缩放区块散列计算难度级别(和/或其它工作量证明难度级别),这涉及将事务240提交到共享的区块链230-236。

[0173] 图10是第四框图,图示了在由所选择的目录实例216计算(也称为核实或验证)区块链230-236的区块260期间的图9的第三示例系统210,其中区块260包括未经核实的队列224-226中的(一个或多个)任何未决事务(在这种情况下仅包括一个事务240,也称为T1),包括图9的开发者注册事务240。

[0174] 第二目录实例216完成对区块260的验证(在计算T1树散列#DD34之后),这包括为第一区块260创建第一区块散列(#6ccf)。也称为区块1(B1)散列的第一区块散列(#6ccf)是通过选择性地散列和/或以其它方式将先前区块散列(B0散列#2ab4)与T1树散列#DD34和其它信息(如满足给定实施方式的需求可能需要或可能不需要的)(例如,存储在第一开发者注册事务(T1)240中的信息)组合或结合来计算的。结果所得的B1散列与紧挨着前一区块212的散列(B0散列)一起被包括在第一区块(B1)260中。

[0175] 注意的是,第一区块(B1)260包括事务区段262,其包括第一事务240或其散列或编码版本。为了说明性的目的,示出了散列区段266,其包括用于B1区块的散列(B1散列)#6ccf

和用于先前区块的散列(B0散列)#2ab4。

[0176] 可以使用T1树散列#DD34来编码事务区段262。在本示例实施例中，B1区块包括或结合至少三个散列，包括T1树散列#DD34、B1散列#6ccff和B0散列#2ab4，以及用于第一事务(T1) 240的公钥#e56c。如图9所示，B1区块260中的T1事务240还可以结合和/或存储T1事务散列#177e(用于计算T1树散列#DD34)。

[0177] 公钥#e56c也可以表示散列。此外，注意的是，在某些实施方式中，B1散列#6ccf可以被用于编码或散列B1区块260，和/或可以简单地被包括在表示B1块260的计算对象或记录中。

[0178] 在计算出包括T1树散列#DD34(其可以用于编码T1事务240)的各种散列266之后，已由第二目录实例(C2) 216核实B1区块并准备好传播到生态系统210的其它目录实例214、218。

[0179] 总之，第二目录实例216创建新区块260，该新区块记录T1树散列#DD34和相关联的事务。新验证的区块260成为区块链230-236的终端区块，如图11所示。

[0180] 图11是第五框图，图示了在由第二目录实例216处理的区块260已经被提交给区块链230-236并且传播到由生态系统210的每个目录实例214-218维护的区块链副本之后的图10的示例系统。注意的是，由于T1事务240现在已经经核实并被提交到已经被传播到其它目录实例214-218的区块260，因此它从未经核实的队列(即，其副本)224-226中被移除。

[0181] 图12是第六框图，图示了在第二目录实例已经发起第二事务(即，第一命名空间注册事务T2) 270并且已经将第二事务(T2) 270传播到软件组件生态系统210中其它目录实例214-218的未经核实的队列224-226之后的图11的示例系统210。

[0182] 第二示例事务(T2) 270包括示例事务有效载荷区段272，其包括事务类型信息(例如，指示当前事务270是命名空间事务)、与T2事务270相关联的开发者实体名称(例如，example.com)，以及与绑定到与T2事务270相关联的命名空间注册的开发者实体(example.com)相关联的命名空间信息(例如，可以用于一个或多个附加名称和/或术语的集合的名称)。为了说明性的目的，示出了T2事务，其包括具有用于T2事务270的事务散列(T2散列#4af5e)的散列区段274。

[0183] 在图12的本示例用例中，第二目录实例216促进T2事务270的创建以及随后到其它目录实例214、218的未经核实的队列224-226的传播。

[0184] 总之，提交命名空间事务(T2) 270(例如，由开发者实体example.com的注册员)以便为开发者实体注册新的命名空间(例如，ECOM)。第二目录实例216创建T2事务270，其表示由开发者实体针对称为ECOM的命名空间的命名空间注册事务。这个T2事务270跨生态系统210广播，因此所有其它节点214-218接收其拷贝。它尚未包含在区块链230-236的任何区块中，因而，其状况在图12中是未决。注意的是，并非所有节点214、218都将同时接收事务，因为接收定时可以取决于网络等待时间或其它因素。

[0185] 注意的是，虽然目录实例214-218中任何一个都可以收集、处理和传播事务，例如T2事务(因为它们都具有经核实的开发者实体，例如，如存储在区块链230-236的B1区块260中)，出于说明性的目的，本示例用例涉及由第二目录实例216创建事务。

[0186] 注意的是，在一些实施方式中，选择性地限制针对开发者的命名空间的注册，直到原始开发者注册在区块链230-236中若干区块深为止，其中深度可以取决于实施方式和与

目录实例214-218相关联的信任级别和/或具有对区块链230-236的读取/写入访问权的其它节点而变化。如果我们使用共识机制,例如在采用不受信任或不完全受信任的目录实例的生态系统中,这可以帮助确保开发者注册是确定的,并且不会因选择差异而回滚。

[0187] 图13是第七框图,图示了在第三目录实例218已经发起第三事务(即,第二命名空间注册事务T3)280之后的图12的示例系统210,该第三事务已经被传播到软件组件生态系统中的其它目录实例214、216的未经核实的队列224-226。

[0188] T3事务280还包括事务有效载荷区段282和散列区段284,该散列区段284包括用于T3事务的散列(T3散列#45bf)。注意的是,虽然在图12和13的各种事务270、280中示出了分开的散列区段284,但是事务散列(#4af5e、#45bf)可以用于对相应的事务270、280进行编码。

[0189] 总之,由第三目录实例218创建和传播T3事务与T2事务的创建和传播类似,不同之处在于第三目录实例218将T3事务280提交到未经核实的队列224-226以准备将T3事务提交到区块链230-236。

[0190] 图14是第八框图,图示了在已经选择第一目录实例214以核实附加区块(B2,如下图15所示)以包括在区块链230-236中之后的图13的示例系统210,其中附加区块将分别结合图12和13的第二事务(T2)270和第三事务(T3)280。注意的是,选择特定目录实例(例如,第一目录实例214)以执行对给定区块的核实可以根据预定的选择机制(例如,规则和/或算法)。确切的规则或算法是特定于实施方式的,并且可以取决于给定实施方式的需求而有所不同。

[0191] 第一目录实例214通过使用不同的事务散列274、284(例如,分别为#4af5e和#45bf)计算树散列来开始计算以准备结合未决事务240、280。关于如何计算散列的确切细节是特定于实施方式的并且可以变化。能够获得本教导的本领域技术人员可以容易地选择并实现特定的散列算法以满足给定实施方式的需求,而无需过多实验。

[0192] 图15是图示图14的示例系统210的第九框图,其示出了在将要被添加到软件组件生态系统210的区块链230-236的附加区块(B2)290的计算期间的第一目录实例214。B2区块290包括事务区段292,其包括或以其它方式结合组合的树散列(将T2和T3事务散列组合,并表示为T2-T3树散列#A324)以及T2和T3事务294。散列区段296包括先前区块散列(B1散列#6ccf)和当前区块散列(B2散列#d21d)。

[0193] 因此,第一目录实例214创建新区块290(即,执行区块核实),其结合了T2(图14的270)和T3(图14的280)的树散列。B1散列#6ccf表示先前散列(即,区块260的散列)。基于组合的树散列#A324、B1散列#6ccf以及可选地附加信息(例如,控制信息和/或来自事务区段292的信息)来计算B2散列。在构建区块290并完成对各种随附散列的计算后,B2区块被传播到区块链230-236以作为新区块B2并入,如图16所示。

[0194] 图16是第十框图,图示了在第一目录实例214已经完成对附加区块的处理并且已经将附加区块290传播到软件组件生态系统210的区块链230-236之后的图15的示例系统210。在将相关联的T2(图14的270)和T3(图14的280)事务294提交到B2区块290之后,将它们从未经核实的队列224-226中移除。

[0195] 注意的是,包括软件组件注册、组件使用注册以及任何其它事务类型在内的各种类型的进一步事务可以以与以上针对三个事务T1、T2、T3讨论的类似方式进行。可以从未决

事务队列(也称为未经核实的队列224-226)中选择性地将零个或多个事务处理成经验证的区块,然后将其添加到链230-236上。

[0196] 如果具有对区块链230-236的读取/写入访问权的新目录实例或其它节点被创建并被添加到生态系统210,那么新节点可以处理区块链230-236以基于在区块链230-236中的注册来重建系统210的准确视图。在一些实施方式中,系统210可以从区块链230-236选择性地提取信息并且以不同的形式存储该信息以方便或提高性能。区块链230-236可以表示真相和描述生态系统210的关键状态的信息,这可以根据区块链230-236的信息重建。

[0197] 按照惯例,计算生态系统一般限于:1.集中控制(例如Apple的AppStore)或2.所谓的Wild West(例如,用户浏览网络以获取独立开发者产品的Windows Mobile方法)。

[0198] 注意的是,在已经受信任的计算环境(即,生态系统)中,与其它类型的数据记录或数据库相反,使用一个或多个区块链特征(对于在下面讨论的一些实施例的范围内的某些实施方式)可以基本上是不必要的,其中网络节点和资源中的一些或全部已经被控制和信任。这种生态系统不太可能需要基于共识的事务评估,因此,竞争、成链和散列的重要性不如加时间戳注册的分布式存储,后者可以经由除区块链之外的另一种类型的分布式数据库来实现。

[0199] 因而,各种实施例(包括所提出的生态系统的各个方面)不一定都需要使用区块链分布式分类账。区块链实施方式可以表示具体的实施例,特别是在涉及不受信任的环境的实施方式中。

[0200] 换句话说,在一些受信任的环境中,不需要使用称为“工作量证明”的机制的“区块链”。注意的是,在以上参考图7-16讨论的某些实施例中,“计算区块的工作”可以经由散列的计算来实现。

[0201] 图17是可经由图1-16的实施例实现的第二示例方法300的流程图。示例方法300促进实现软件组件生态系统(例如,根据图1-3的实施例构造的生态系统)。

[0202] 第二示例方法300包括第一步骤302,其涉及接受来自用户输入设备的一个或多个信号以定义开发者实体的识别名称。例如,参考图4,这可以涉及注册员系统42从注册员82接受与开发者实体事务84有关的输入(即,开发者实体名称86)。

[0203] 第二步骤304包括接受来自用户输入设备的一个或多个信号以定义安全核实机制(可用于核实开发者实体被授权注册该识别名称)。例如,参考图4,安全机制可以包括公钥88和数字签名90,例如供身份域(例如,图2的身份域44)和/或目录实例32的安全性和允许模块(例如,图3的模块62)使用,这可以实现用于处置用于开发者实体核实的公钥88和数字签名90的公钥基础设施。

[0204] 第三步骤包括将条目记录在由目录(例如,图4的第一目录实例32)实现的分布式数据库中。初始记录可以包括在将对应条目记录在实现分布式分类账(例如,图3的分布式分类账14)的分布式数据库的分布式未经核实的队列38中之前核实开发者实体事务数据,该分布式分类账还包括区块链20。一旦将记录的条目提交到区块链20,就称其已向图3的分布式分类账14和随附的区块链20注册。最初记录的条目包括与核实机制相关联的开发者名称。

[0205] 第四步骤308包括与目录(即,经由运行目录软件的实例实现的分布式数据库或分类账)条目相关联地执行事务,以传送软件组件。与注册的开发者实体和一个或多个相关联

的命名空间相关联地,这个事务也可以由已作为贡献者向分类账注册的注册贡献者(例如,个人开发者)作为软件组件下载注册事务而注册在分布式分类账(例如,分布式分类账14和随附的分布式区块链20)中。

[0206] 第五步骤310包括使用目录条目(即,目录实例的分布式分类账(例如,图3的分布式分类账14)的区块链(例如,图3的区块链20)中的条目(与开发者实体注册事务对应))来核实(即,完成)描述软件组件下载事务的另一个事务的注册和提交(向图3的分布式分类账14的区块链20)。

[0207] 注意的是,出于本讨论的目的,目录可以是用于实现分布式分类账(例如,区块链)的代码集。目录可以包括或不包括可用软件组件的本地数据库,因为这样的数据库可以与目录分开实现,而不脱离本教导的范围。但是,在本文讨论的各种实施例中,给定的目录实例还包括用于与可用组件的本地数据库(例如,图3的本地组件目录36)进行交互并对其进行维护的计算机代码。术语“目录”和“目录实例”在本文中可以互换使用。

[0208] 注意的是,在不脱离本教导的范围的情况下,方法300可以被更改。例如,可以添加附加步骤;一些步骤可以互换;步骤可以被移除或代替;以及可以向方法300的各个步骤302-310添加和/或从其移除附加修改或进一步的细节。

[0209] 例如,第二示例方法300还可以指定安全核实机制包括公钥和数字签名。方法300还可以指定将目录条目传播到软件组件生态系统中的多个节点(例如,目录实例)的步骤。

[0210] 第二示例方法300还可以指定多个节点中的每个节点表示目录实例,该目录实例包括用于与其它多个节点协同实现分布式区块链分类账的代码,其它多个节点维护分布式区块链分类账的副本或近似副本。

[0211] 第二示例方法300还可以指定用于为条目分配命名空间的过程,例如,向相关联的分布式区块链20注册命名空间注册事务。

[0212] 第二示例方法300还可以指定和/或替换为以下步骤:使用包括多个互相通信的目录实例的分布式对等组件生态系统的组件目录的实例,执行以下操作:接受与开发者实体名称有关的开发者注册实体信息;核实开发者实体信息并将开发者实体信息提交到被实现为区块链的分布式数据库作为开发者实体注册记录或区块,其中,多个互相通信的目录实例中的每一个都可访问该区块链,从而将开发者实体注册信息传播到对等组件生态系统的所有完整节点。

[0213] 第二示例方法300还可以指定接受开发者注册实体信息的步骤在以下步骤之后发生:由开发者实体向多个互相通信的目录实例中的目录实例供应开发者实体名称、与正为其执行注册的目录实例相关联的公钥、以及与开发者实体名称相关联并且可用于确认开发者实体身份的数字签名(使用开发者实体秘密,即私钥)。

[0214] 对等组件生态系统的完整节点可以包括多个互相通信的目录实例。第二示例方法300还可以指定:在传播开发者实体注册信息之后,并且响应于命名空间注册过程的发起和实现,通过以下操作为开发者实体分配一个或多个命名空间:在命名空间注册过程期间将一个或多个供应的命名空间与开发者注册信息相关联。

[0215] 第二示例方法300还可以包括选择性地延迟开发者实体的命名空间注册过程的实现,直到在若干记录或区块被提交到区块链的记录的链之后。

[0216] 图18是可用于实现图1-17的实施例的系统900和随附的计算环境的总体框图。根

据本发明的实施例,示例系统900能够实现分布式软件组件生态系统。实施例可以被实现为独立应用(例如,驻留在用户设备中)或者作为使用客户端侧代码和服务器侧代码的组合而实现的基于web的应用。

[0217] 通用系统900包括用户设备960-990,包括台式计算机960、笔记本计算机970、智能电话980、移动电话985和平板电脑990。通用系统900可以与能够显示和导航网页或其它类型的电子文档和UI、和/或执行应用的任何类型的用户设备(诸如瘦客户机计算机、启用互联网的移动电话、移动互联网接入设备、平板电脑、电子书或个人数字助理)进行接口连接。虽然系统900被示为具有五个用户设备,但是可以支持任意数量的用户设备。

[0218] Web服务器910被用来处理来自Web浏览器和独立应用对网页、电子文档、企业数据或其它内容以及来自用户计算机的其它数据的请求。Web服务器910还可以提供与企业操作相关的数据的推送数据或聚合内容(诸如RSS馈送)。

[0219] 应用服务器920操作一个或多个应用。应用可以被实现为一个或多个以任何编程语言编写的脚本或程序,编程语言诸如Java、C、C++、C#或任何脚本语言(诸如JavaScript或ECMAScript(欧洲计算机制造商协会脚本)、Perl、PHP(超文本预处理器)、Python、Ruby或TCL(工具命令语言))。可以使用库或应用框架(诸如Rails、Enterprise JavaBeans或.NET)构建应用。Web内容可以使用HTML(超文本标记语言)、CSS(层叠样式表)和其它Web技术(包括模板语言和分析器)创建。

[0220] 在应用服务器920上运行的数据应用适于处理输入数据和用户计算机请求,并且可以从数据存储设备或数据库930存储或检索数据。数据库930存储由数据应用创建和使用的数据。在实施例中,数据库930包括适于响应于SQL格式命令或其它数据库查询语言而存储、更新并检索数据的关系数据库。其它实施例可以使用非结构化数据存储架构和NoSQL(不仅是SQL)数据库。

[0221] 在实施例中,应用服务器920包括能够执行程序或脚本的一个或多个通用计算机。在实施例中,Web服务器910被实现为在一个或多个通用计算机上运行的应用。Web服务器910和应用服务器920可以在相同的计算机上被组合和执行。

[0222] 电子通信网络940-950使得用户计算机960-990、Web服务器910、应用服务器920和数据库930之间能够通信。在实施例中,网络940-950还可以包括任何形式的电或光通信设备,包括有线网络940和无线网络950。网络940-950还可以结合一个或多个局域网(诸如以太网)、广域网(诸如互联网);蜂窝载波数据网络;以及虚拟网络(诸如虚拟专用网络)。

[0223] 该系统是根据本发明实施例的用于执行应用的一个示例。在另一个实施例中,应用服务器910、Web服务器920以及可选地数据库930可以被组合成单个服务器计算机应用和系统。在进一步的实施例中,可以使用虚拟化和虚拟机应用来实现应用服务器910、Web服务器920和数据库930中的一个或多个。

[0224] 在更进一步的实施例中,Web和应用服务功能的全部或一部分可以被集成到在每个用户计算机上运行的应用中。例如,用户计算机上的JavaScript应用可以被用来检索或分析数据并显示应用的部分。

[0225] 作为示例,参考图2和18,Web服务器910、应用服务器920和数据存储设备/数据库930可以被用于通过托管与目录实例32对应的服务器侧应用来实现图2的分布式分类账14,其进而经由浏览器对各个计算机系统可访问。可替代地或附加地,各个计算设备950、985、

970、980、990可以运行目录实例软件,该目录实例软件使用有线网络940和/或无线网络950联网到对等组件生态系统中。

[0226] 图19是可用于实现图1-18的实施例的计算设备的总体框图。虽然图19的系统500被描述为促进执行本文的某些实施方式中所描述的步骤,但是系统500的任何合适的组件或组件的组合或者与系统500相关联的任何合适的一个或多个处理器可以用于执行所描述的步骤。

[0227] 图19图示了可用于本文描述的实施方式的示例计算系统500的框图。例如,计算系统500可以被用来实现图18的服务器设备910、920以及执行本文描述的方法实施方式。在一些实施方式中,计算系统500可以包括处理器502、操作系统504、存储器506和输入/输出(I/O)接口508。在各种实施方式中,处理器502可以被用来实现本文描述的各种功能和特征,以及执行本文描述的方法实施方式。虽然处理器502被描述为执行本文描述的实施例,但系统500的任何合适的组件或组件的组合或者与系统500或任何合适系统相关联的任何合适的一个或多个处理器可以执行所描述的步骤。本文描述的实施例可以在用户设备上、服务器上或两者的组合上执行。

[0228] 计算设备500还包括软件应用510,该软件应用可以存储在存储器506上或者任何其它合适的存储位置或计算机可读介质上。软件应用510提供使处理器502能够执行本文描述的功能和其它功能的指令。计算系统500的组件可以由一个或多个处理器或硬件设备的任意组合以及硬件、软件、固件等等的任意组合来实现。

[0229] 为了便于说明,图19为处理器502、操作系统504、存储器506、I/O接口508和软件应用510中的每一个示出了一个方框。这些方框502、504、506、508和510可以表示多个处理器、操作系统、存储器、I/O接口和软件应用。在各种实施方式中,计算系统500可以不具有示出的所有组件和/或可以具有包括其它类型的组件的其它元件,代替本文示出的组件或作为其附加。

[0230] 作为示例,参考图3和19,图19的计算设备500可以用于实现图3的个人开发者系统72。

[0231] 虽然已经关于其特定实施例描述了本描述,但是这些特定实施例仅仅是说明性而不是限制性的。

[0232] 虽然已经关于其特定实施例描述了本描述,但是这些特定实施例仅仅是说明性而不是限制性的。

[0233] 可以使用任何合适的编程语言来实现特定实施例的例程,包括C、C++、Java、汇编语言等等。可以采用不同的编程技术,诸如过程式或面向对象。例程可以在单个处理设备或多个处理器上执行。虽然步骤、操作或计算可以以具体次序呈现,但是这个次序在不同的特定实施例中可以改变。在一些特定实施例中,可以同时执行在本说明书中顺序示出的多个步骤。

[0234] 特定实施例可以在供指令执行系统、装置、系统或设备使用或与其结合使用的计算机可读存储介质中实现。特定实施例可以以软件或硬件或两者的组合中的控制逻辑的形式来实现。当由一个或多个处理器执行时,控制逻辑可以可操作,以执行在特定实施例中描述的内容。

[0235] 可以通过使用经编程的通用数字计算机、通过使用专用集成电路、可编程逻辑器

件、现场可编程门阵列、光学、化学、生物、量子或纳米工程系统、组件和机制来实现特定实施例。一般而言,特定实施例的功能可以通过本领域已知的任何手段来实现。可以使用分布式的、联网的系统、组件和/或电路。数据的通信或传输可以是有线的、无线的或通过任何其它手段。

[0236] 还将认识到的是,图/附图中描绘的元素中的一个或多个也可以以更分开或集成的方式实现,或者甚至在某些情况下被移除或渲染为不可操作,如根据特定应用有用的。实现可以存储在机器可读介质中的程序或代码以允许计算机执行上述任何方法也在本发明的精神和范围内。

[0237] 如在本文的描述中以及贯穿随后的权利要求所使用的,除非上下文另有明确说明,否则“一”、“一个”和“该”包括复数的所指对象。而且,如在本文的描述中以及贯穿随后的权利要求所使用的,除非上下文另有明确说明,否则“在……中”的含义包括“在……中”和“在……上”。

[0238] 因此,虽然已经描述了特定的实施例,但是在前述公开内容中预期了修改、各种改变和替换的活动余地 (latitude), 并且将认识到的是,在不背离所阐述范围和精神的情况下,在一些情况下,特定实施例的一些特征将在没有其它特征的对应使用的情况下被采用。因此,可以做出许多修改,以便使特定的情形或材料适应本质范围和精神。

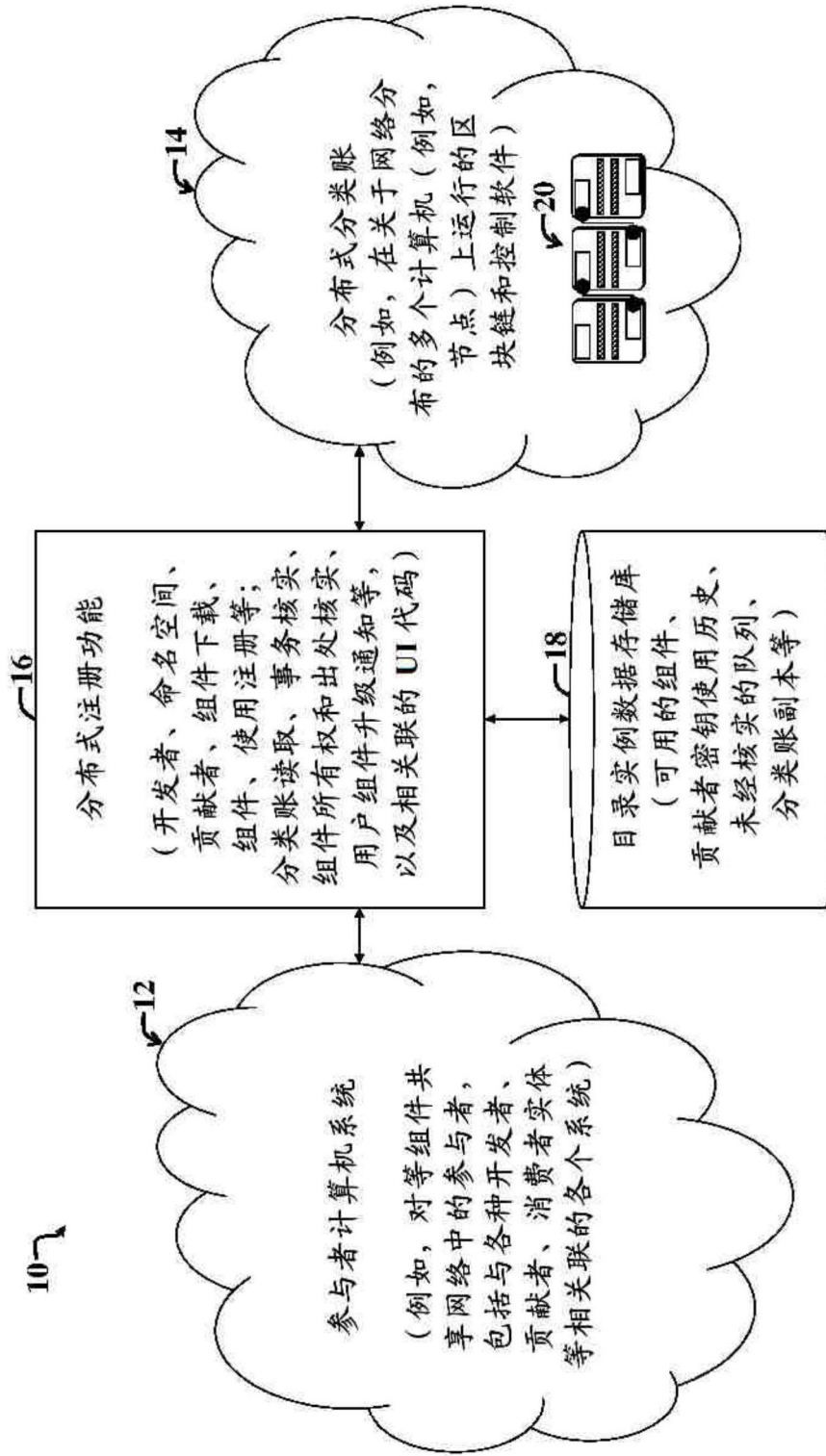


图1

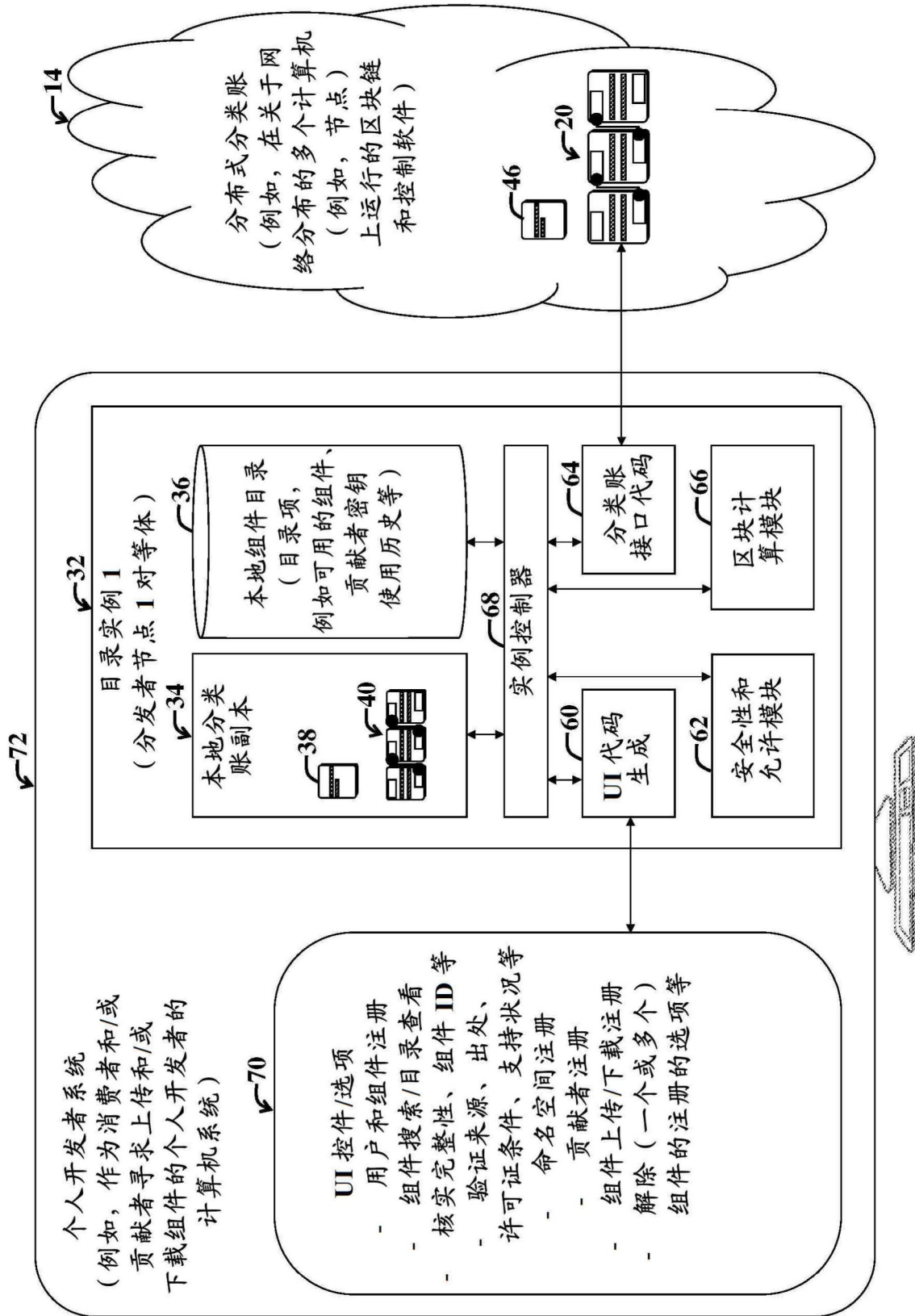


图3

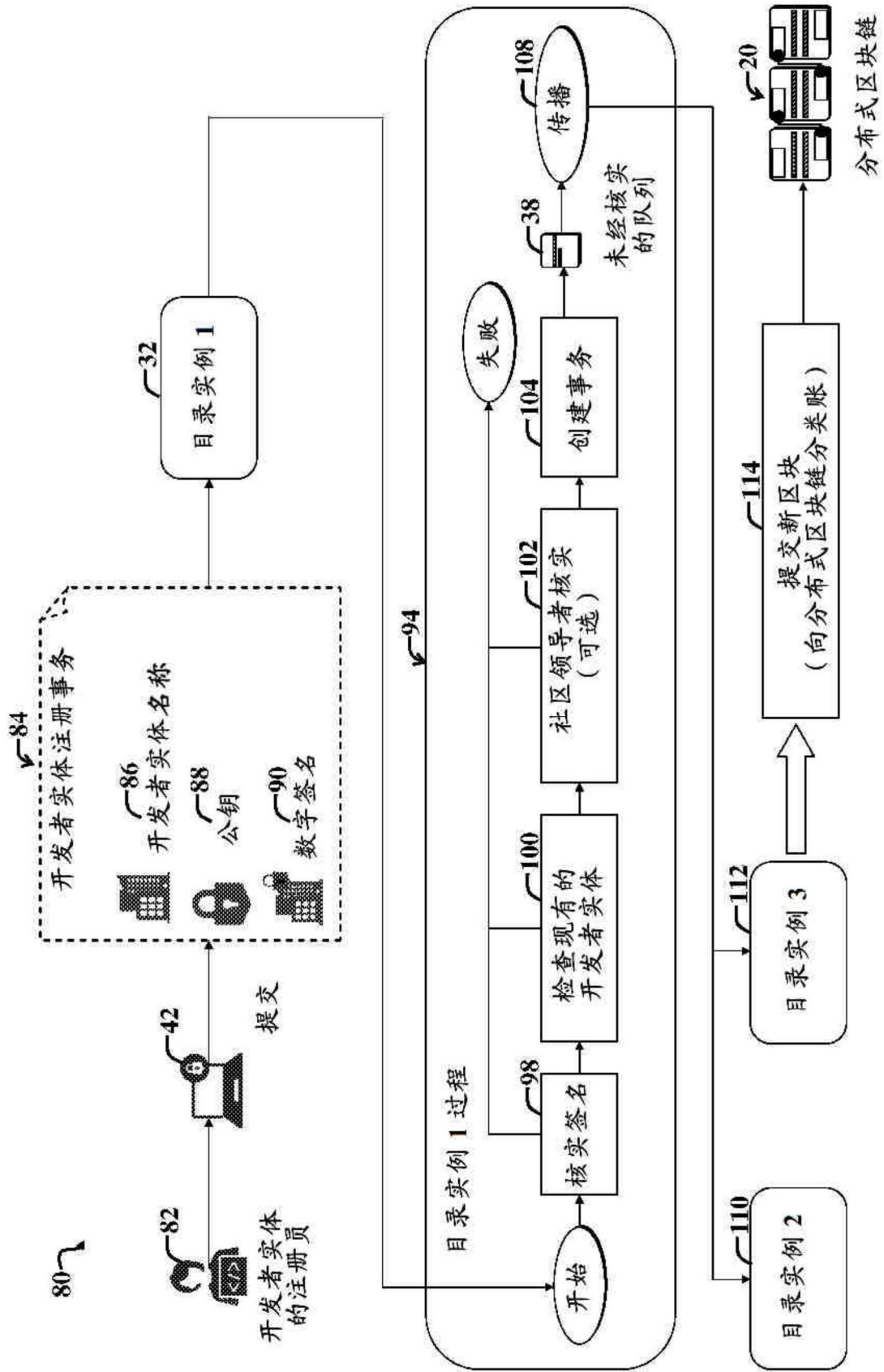


图4

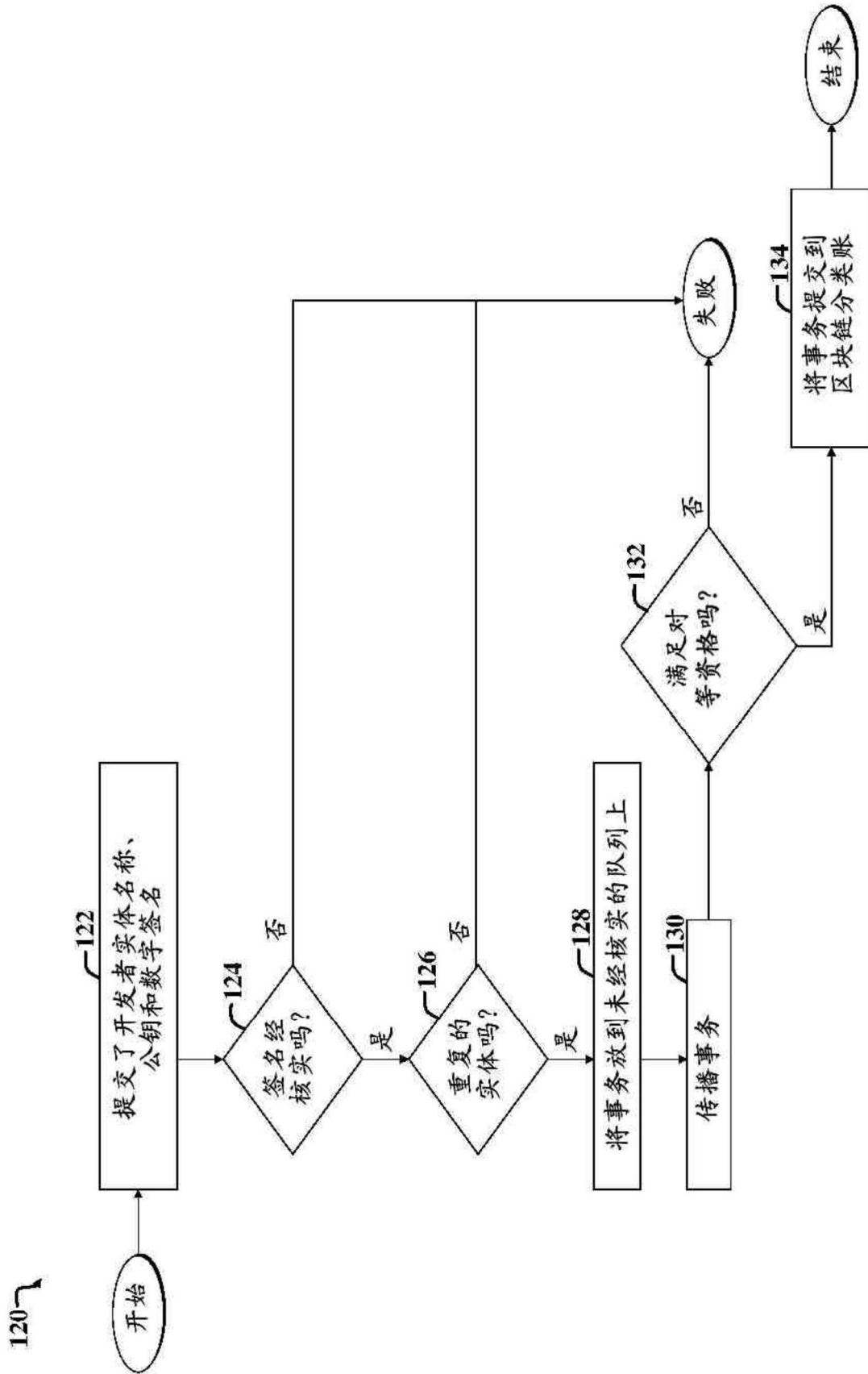


图5

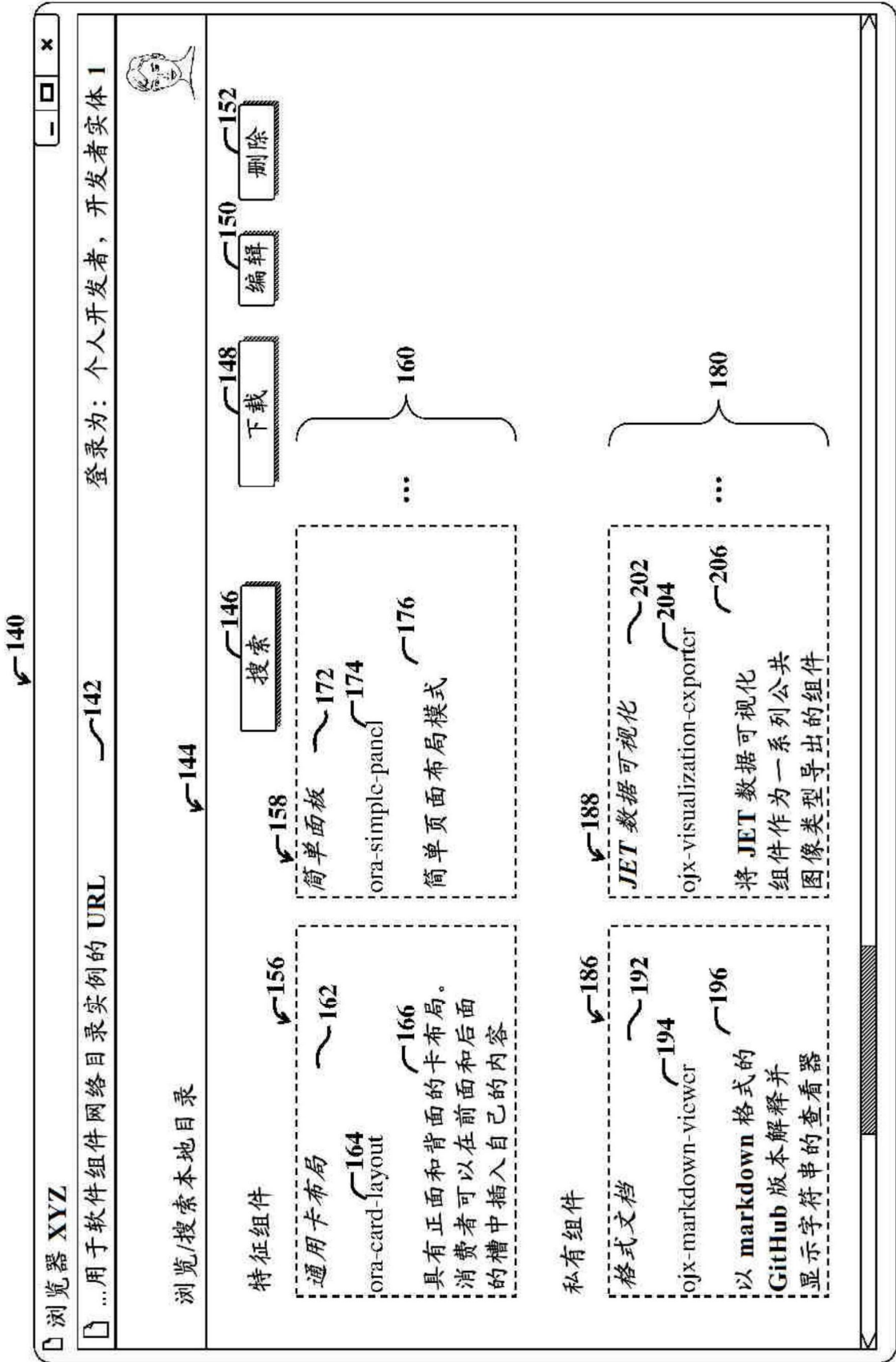


图6

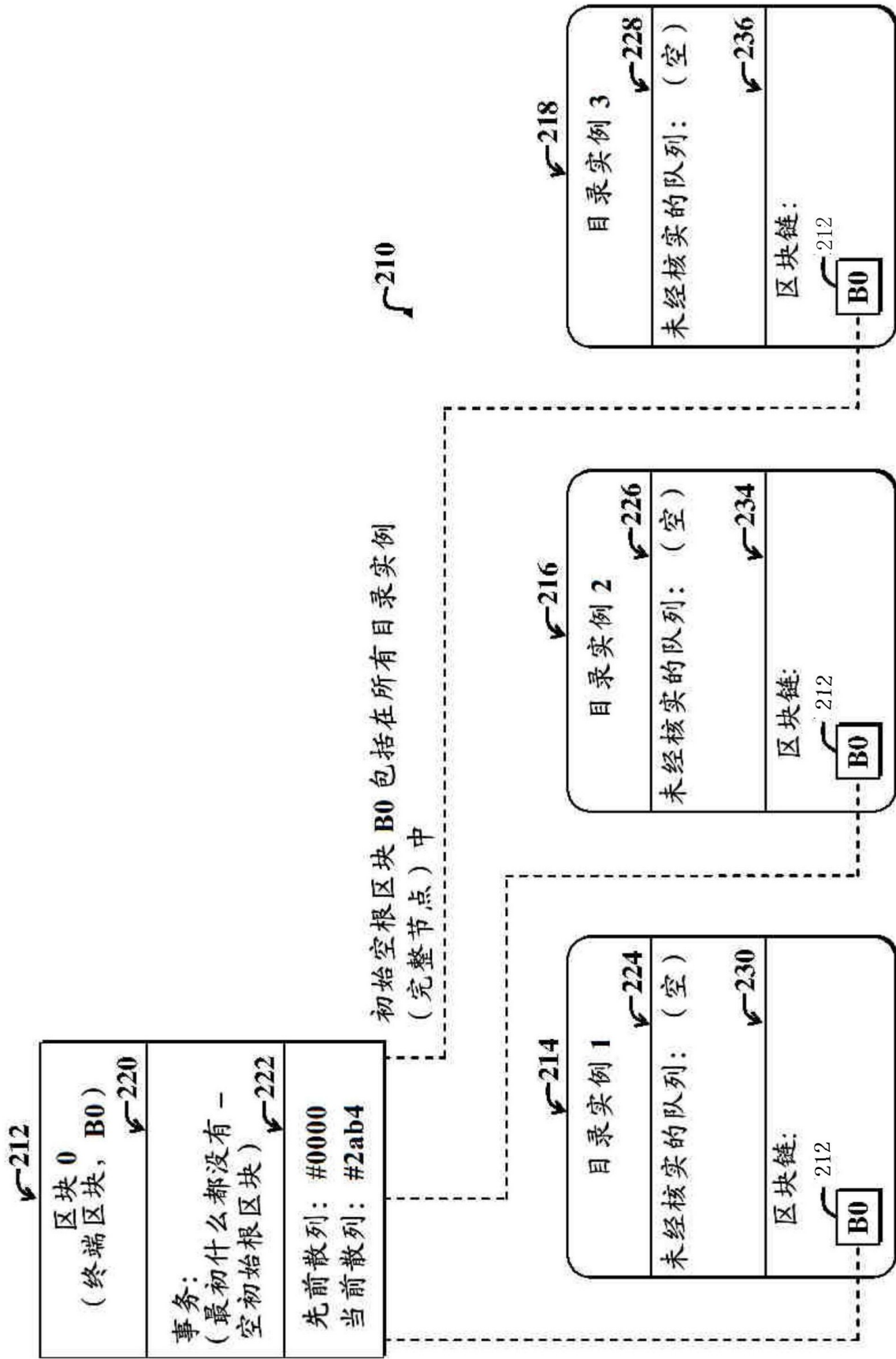


图7

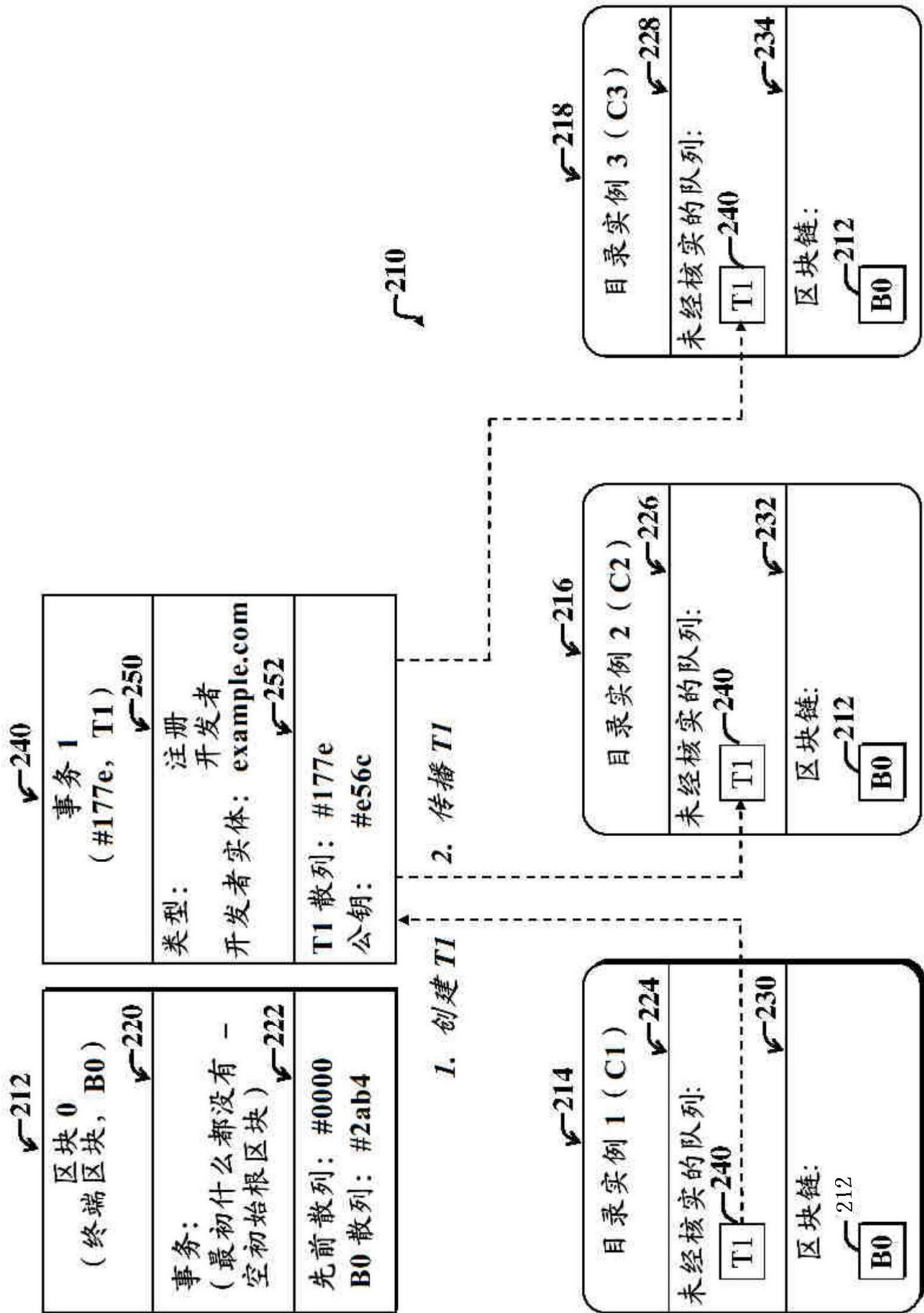


图8

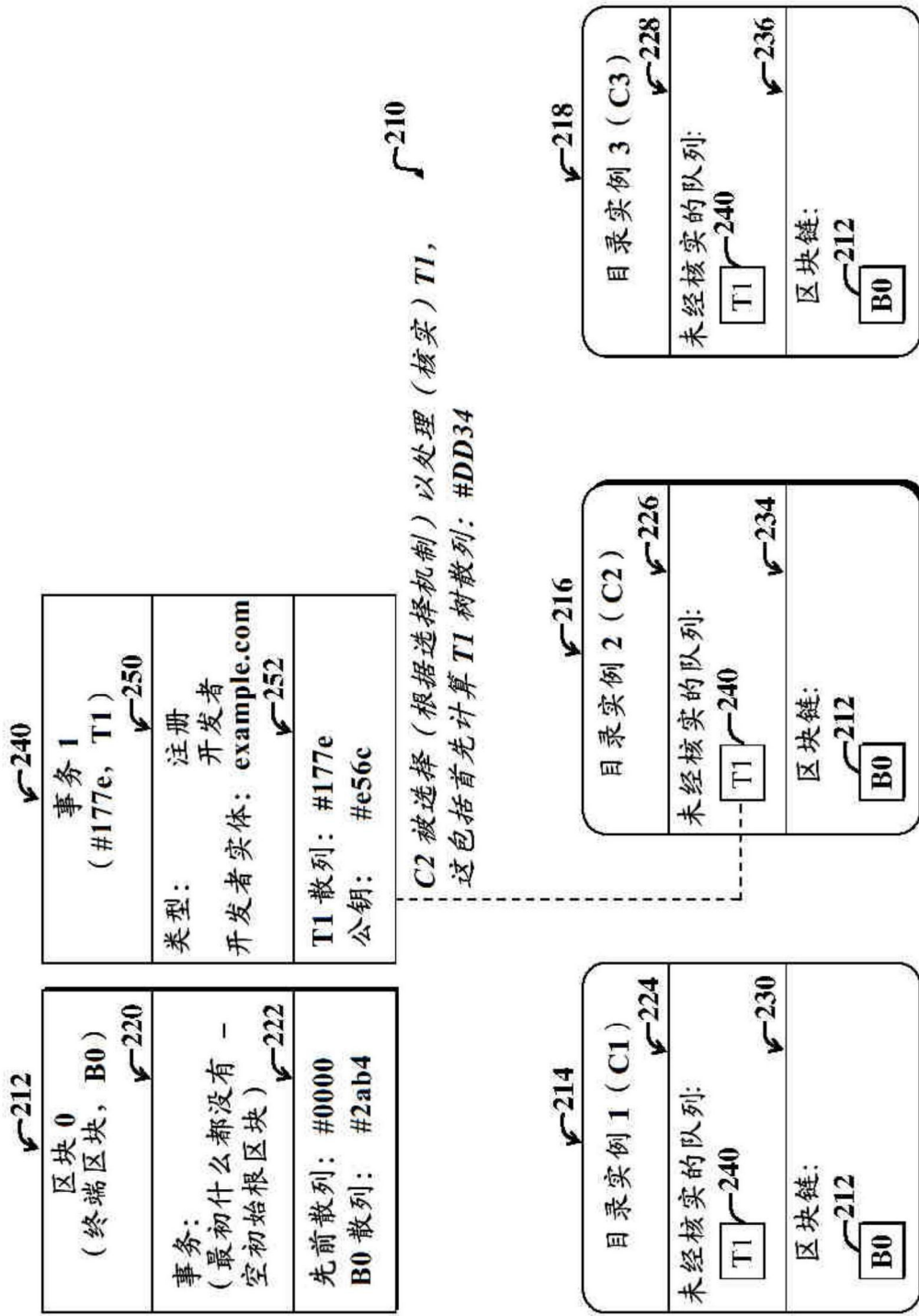


图9

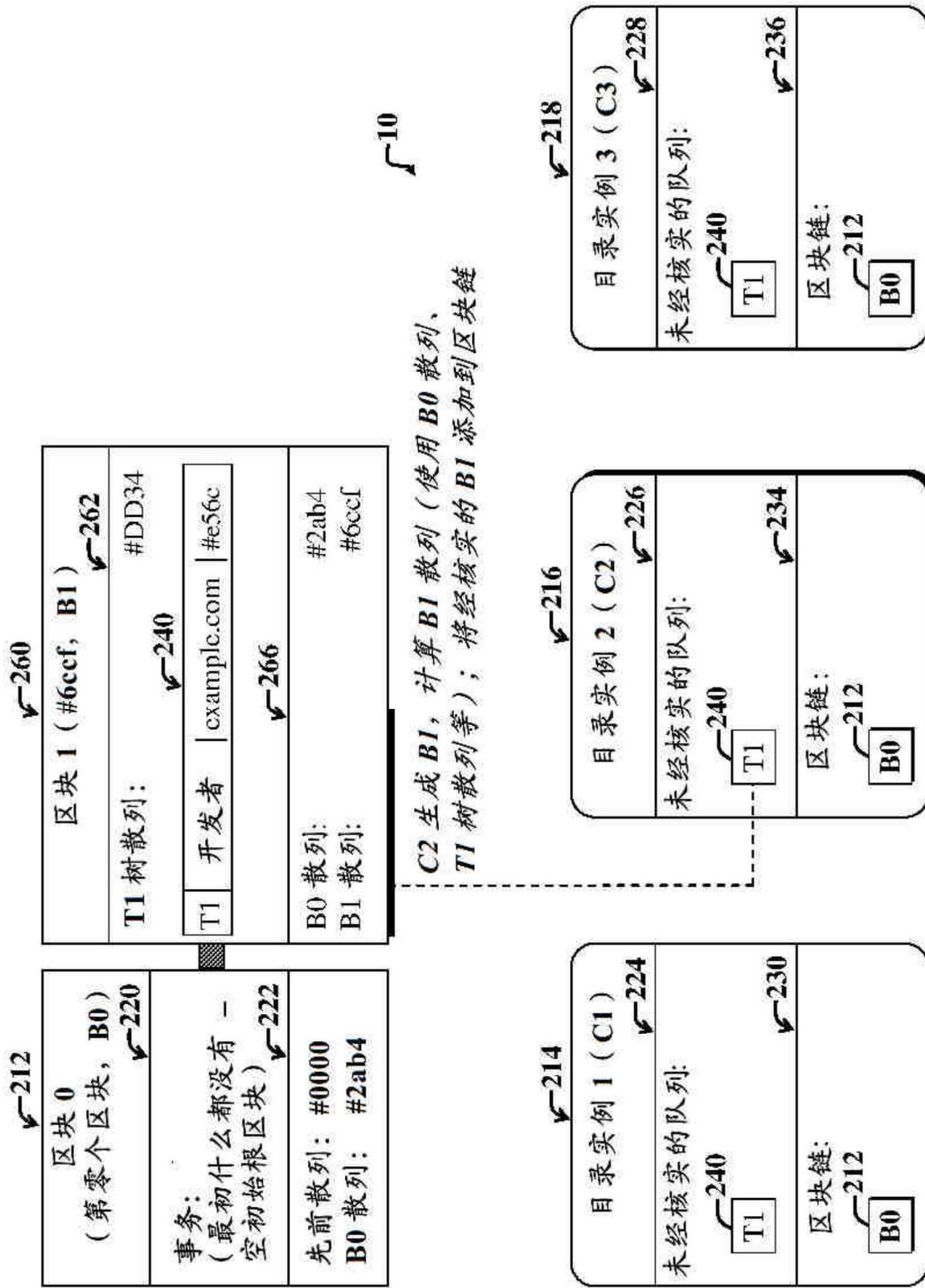


图10

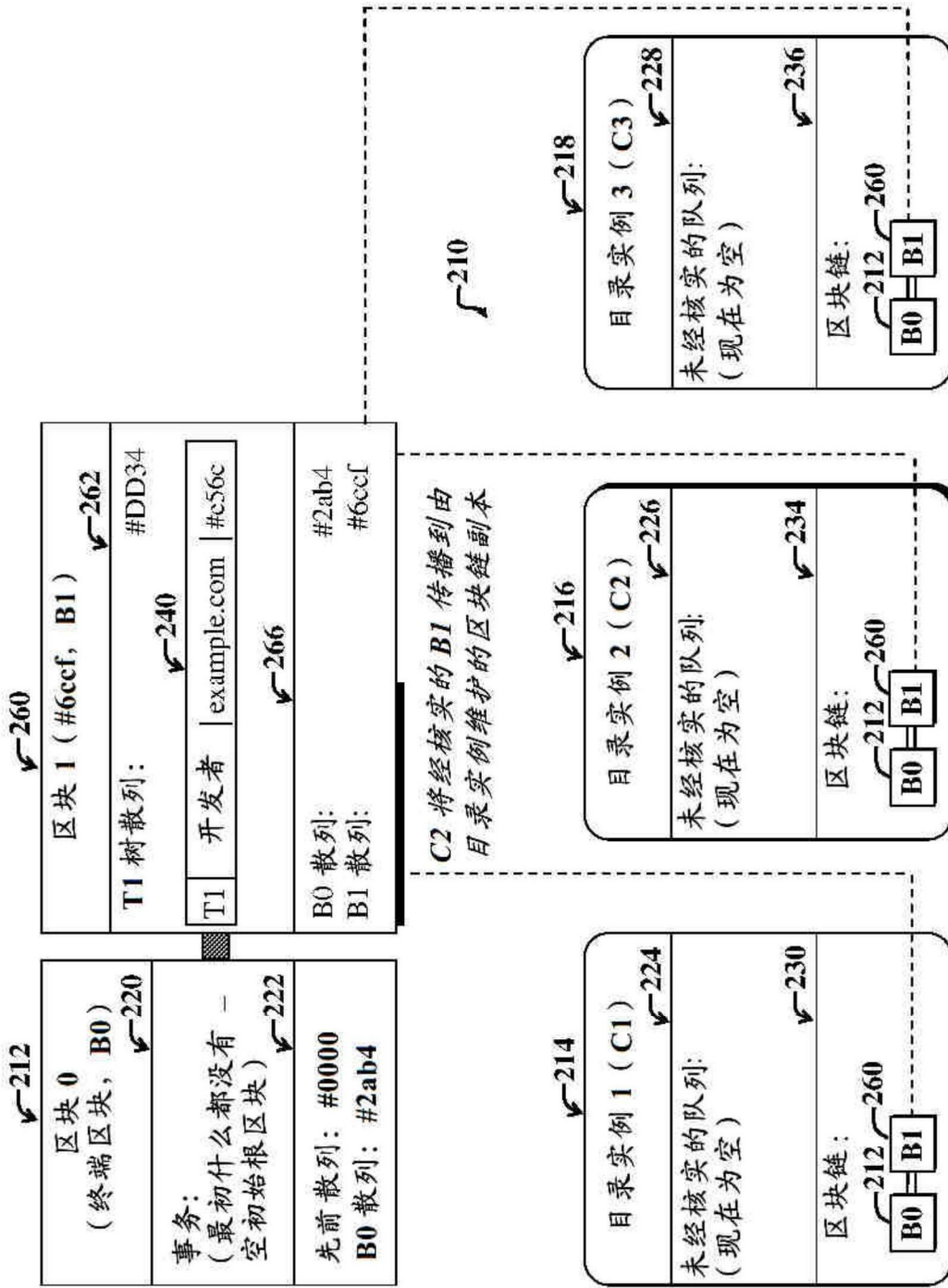


图11

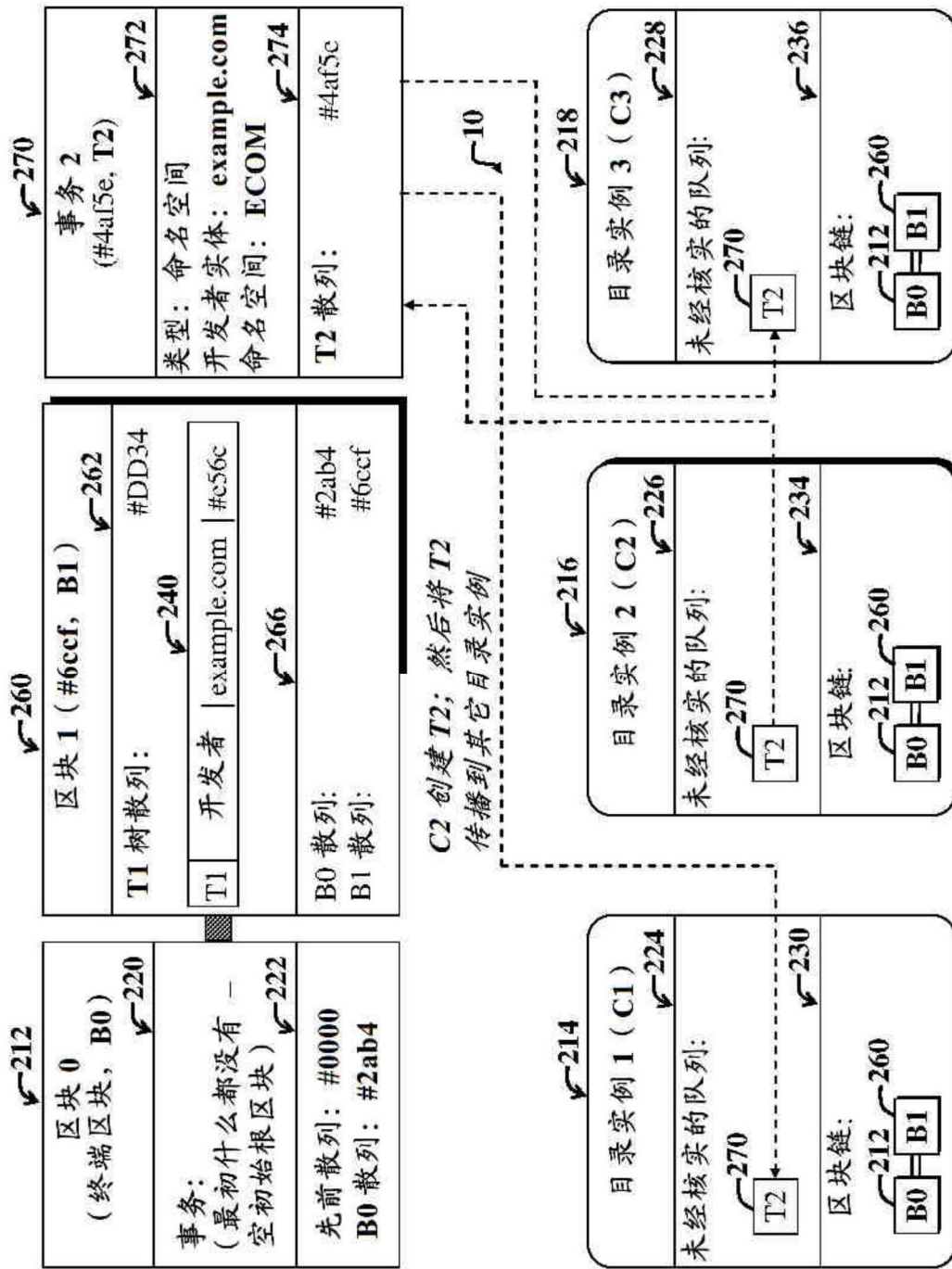


图12

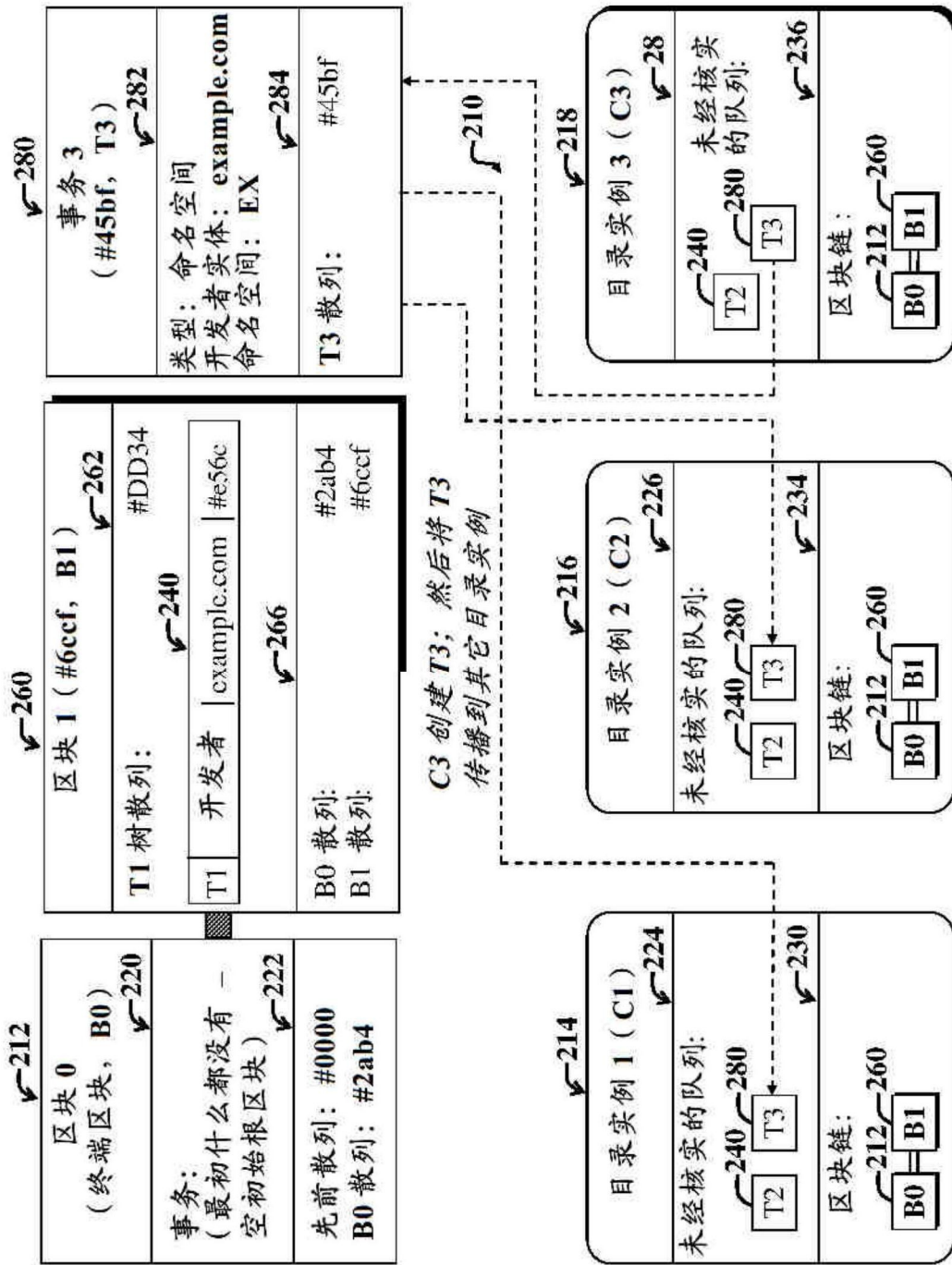


图13

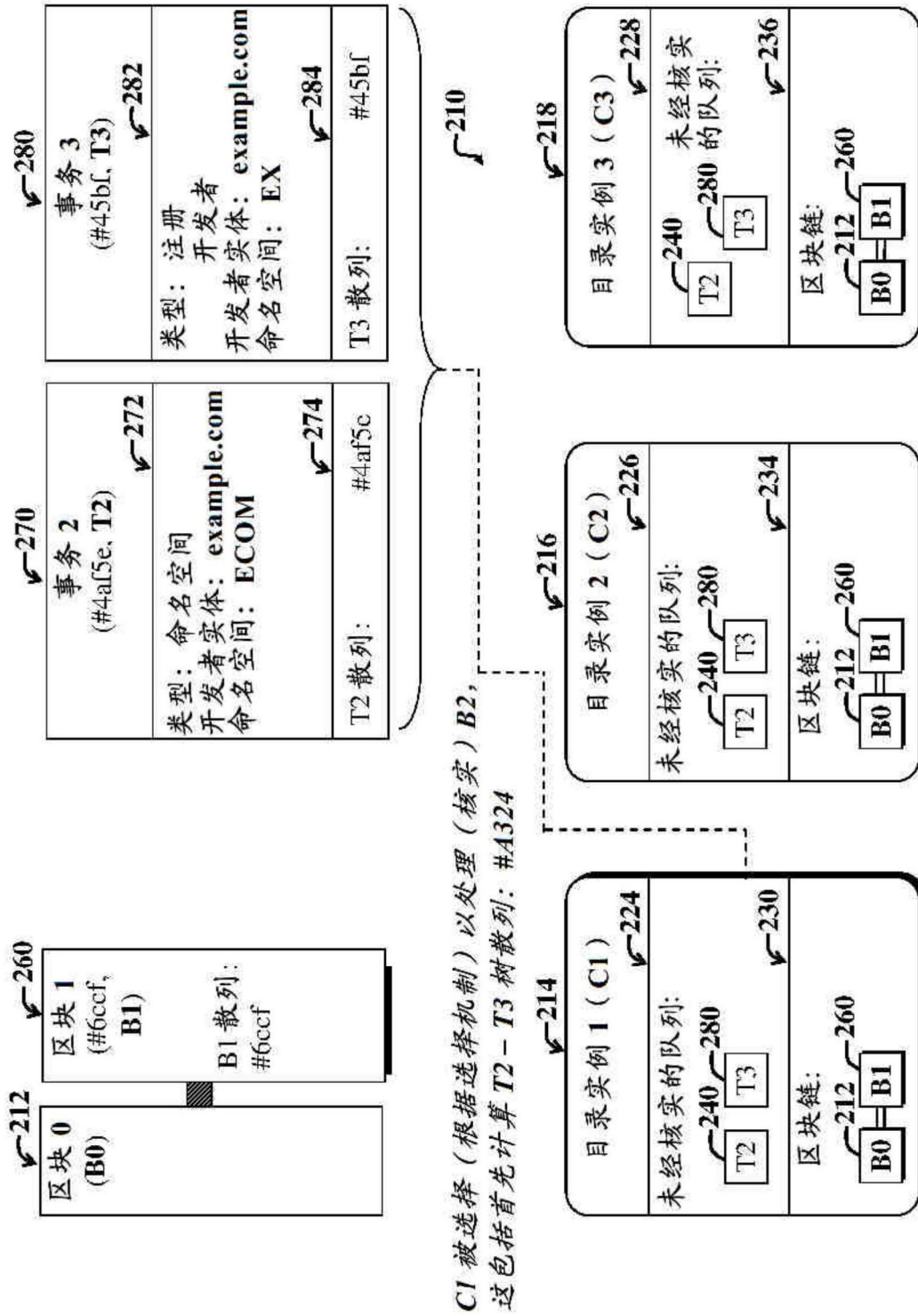


图14

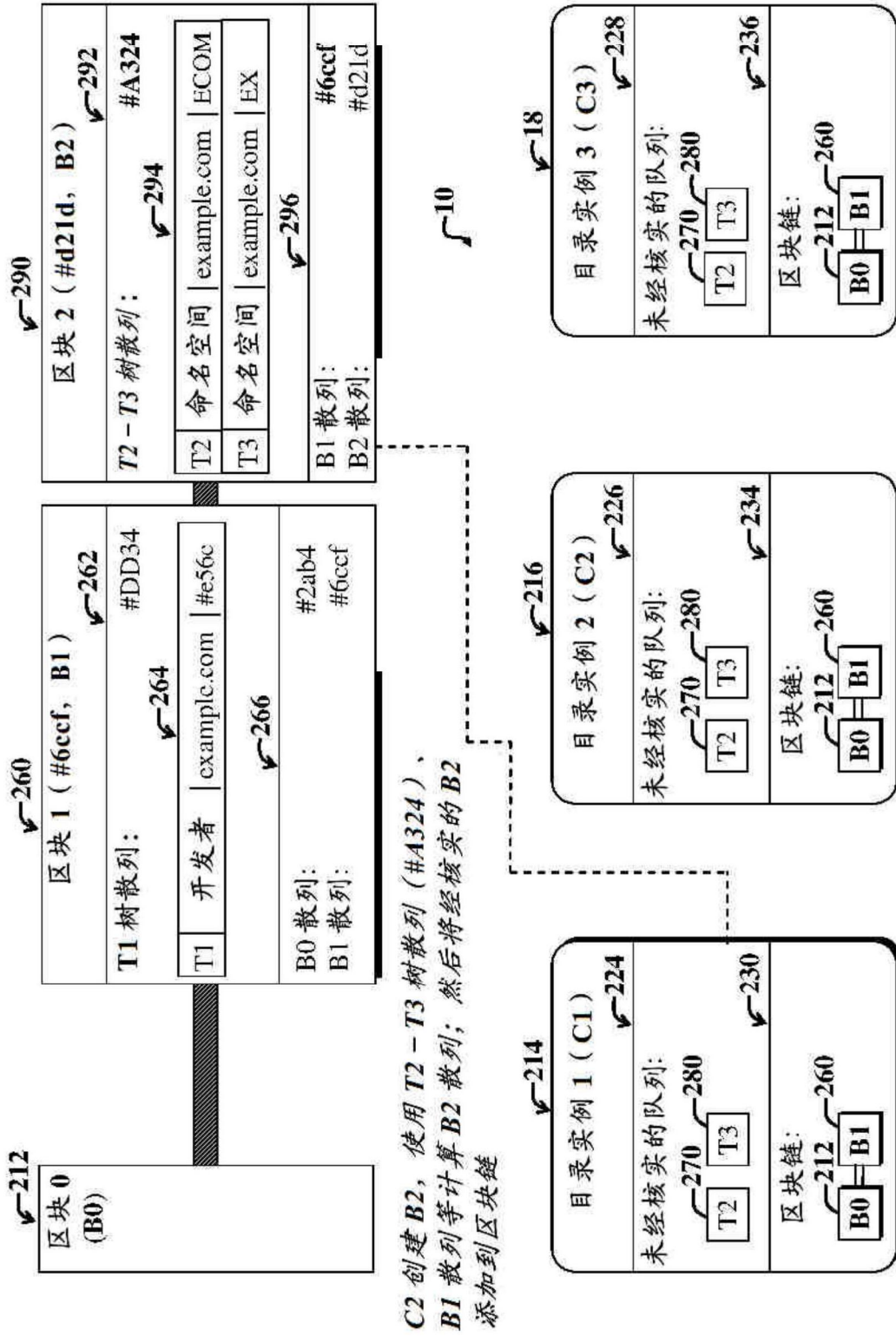


图15

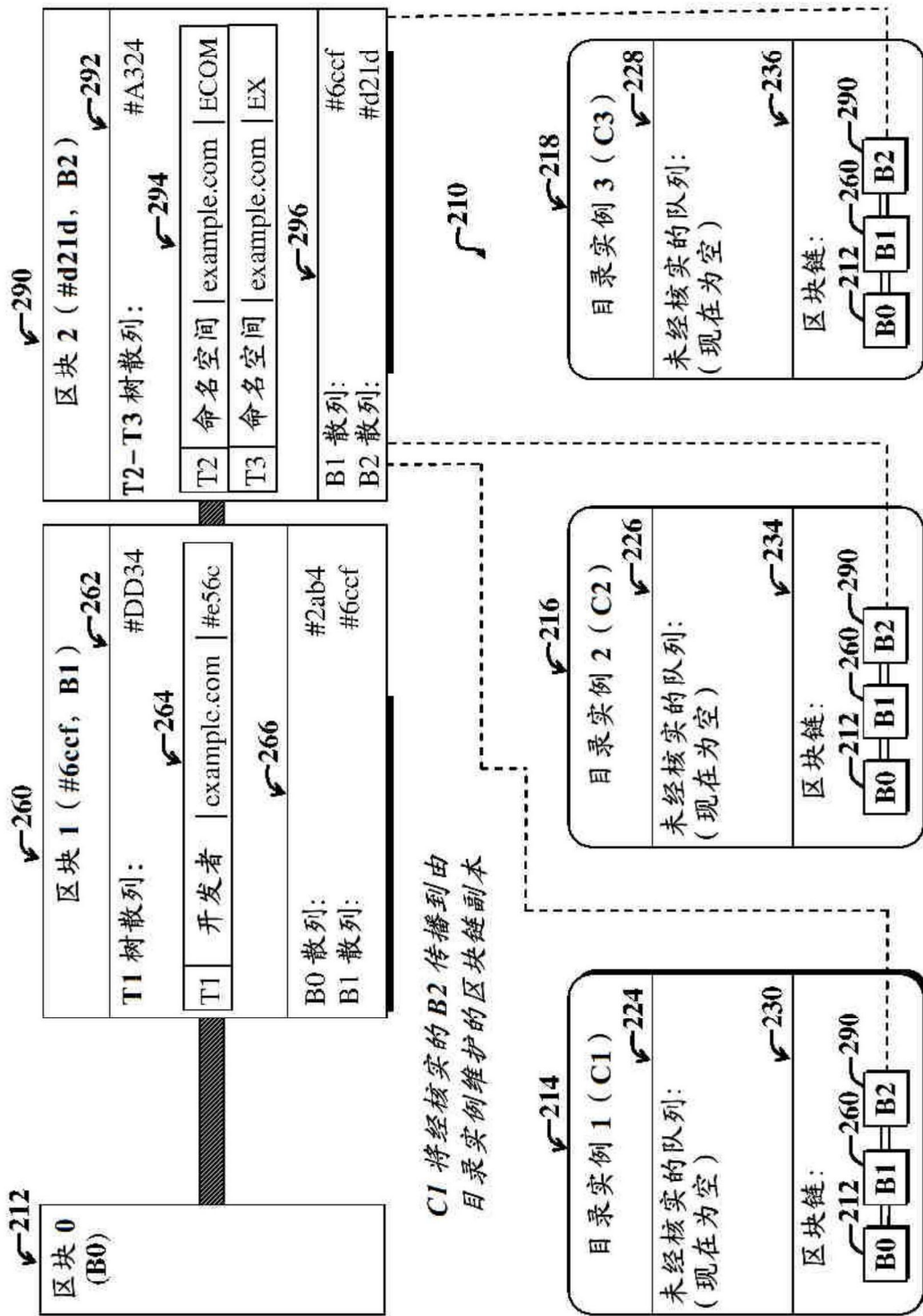


图16

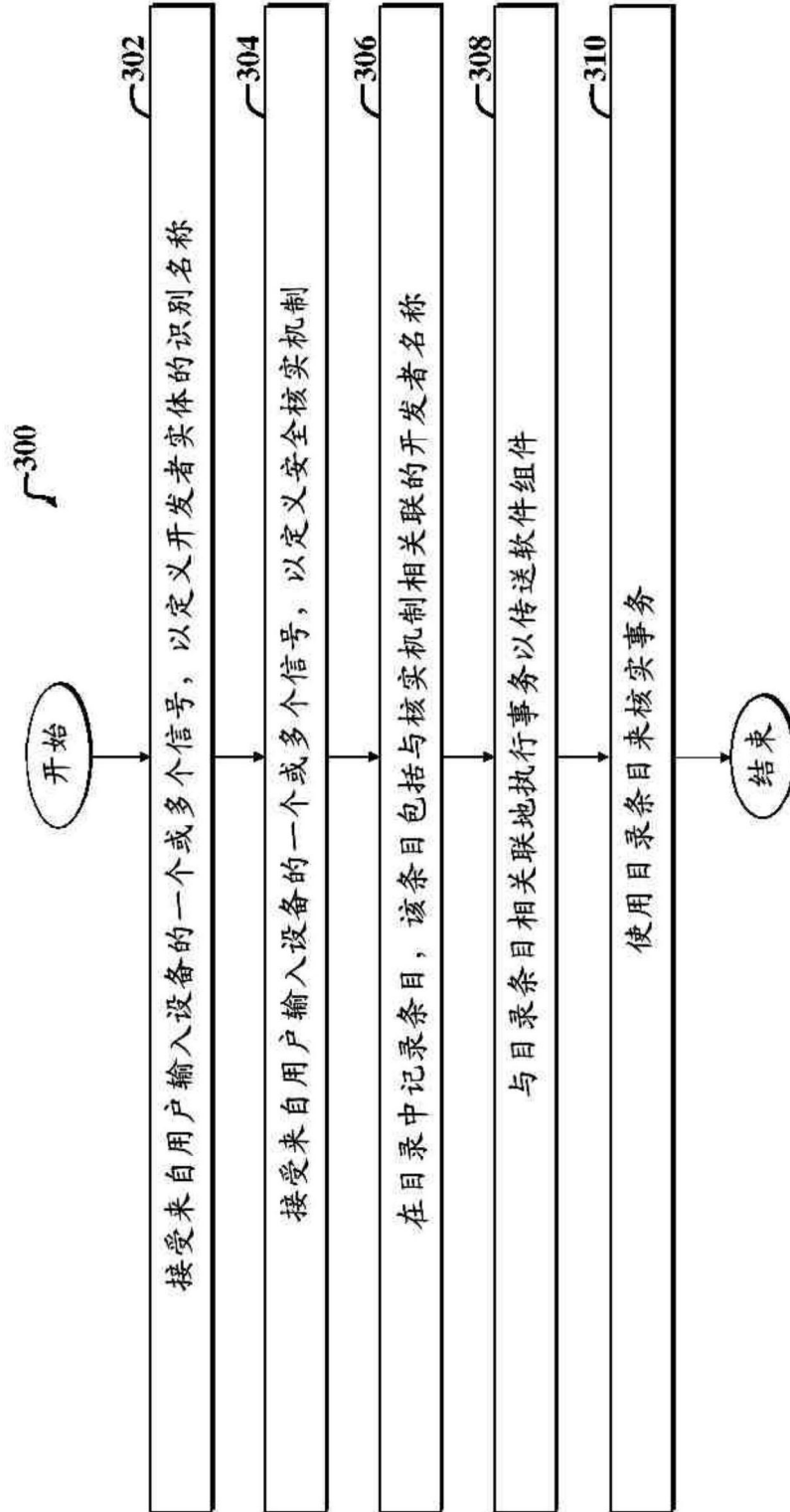


图17

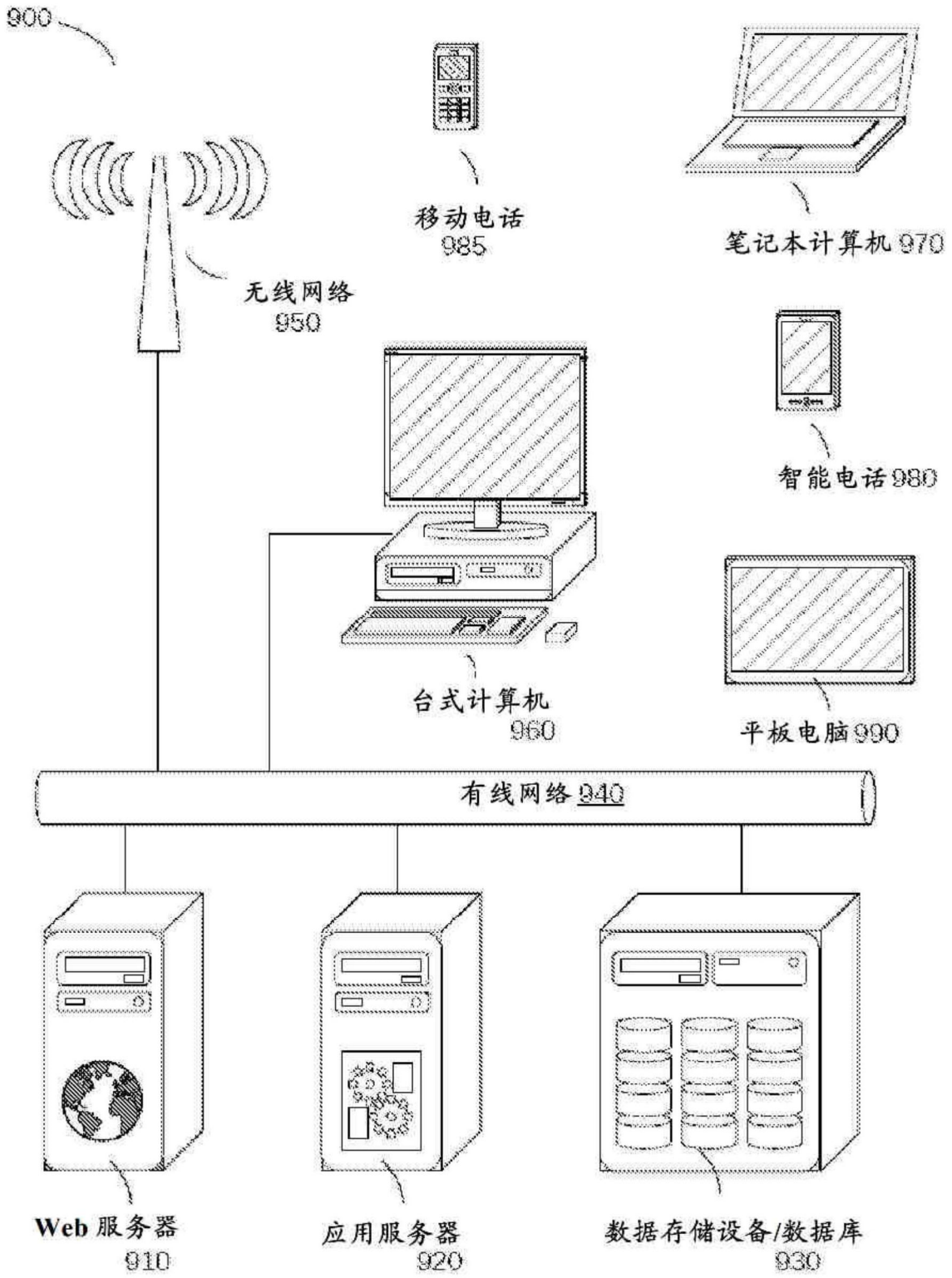


图18

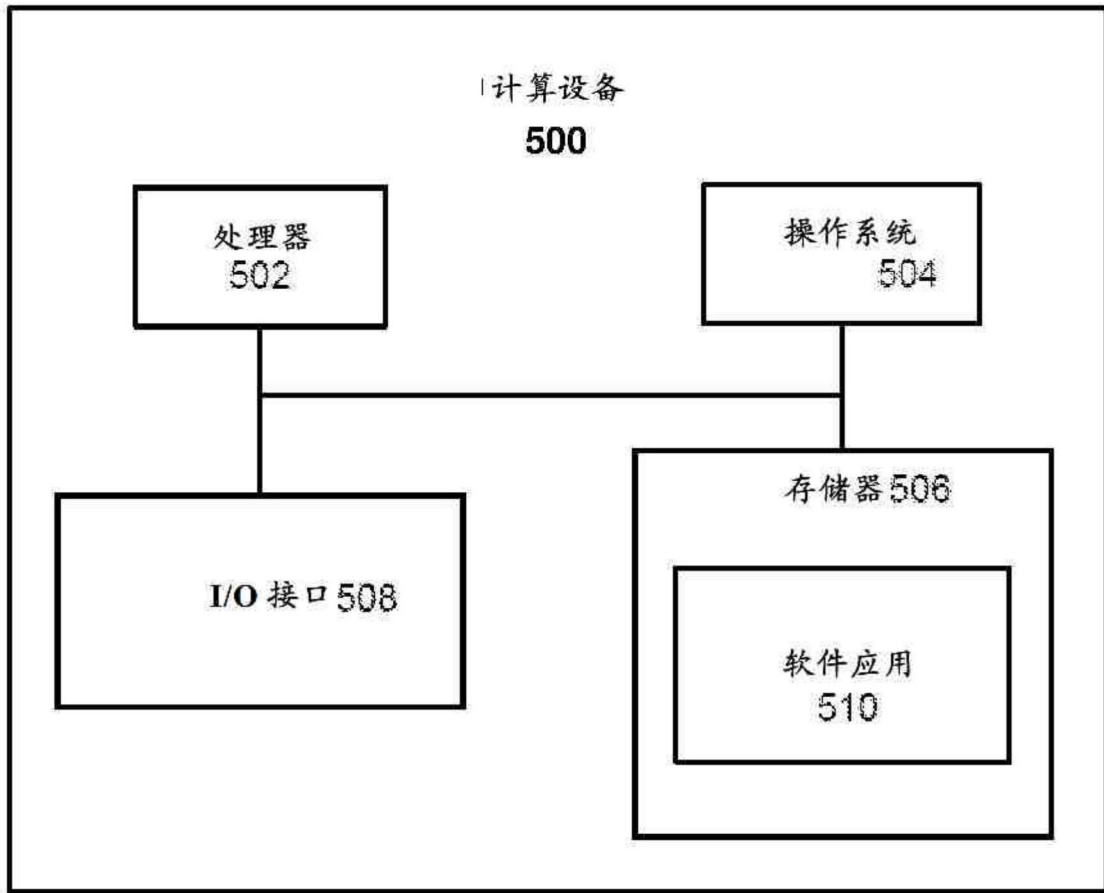


图19