



# (12) 发明专利

(10) 授权公告号 CN 105431823 B

(45) 授权公告日 2021.06.29

(21) 申请号 201480043542.5

J·哈平

(22) 申请日 2014.05.30

(74) 专利代理机构 上海专利商标事务所有限公司 31100

(65) 同一申请的已公布的文献号  
申请公布号 CN 105431823 A

代理人 蔡悦

(43) 申请公布日 2016.03.23

(51) Int.Cl.

(30) 优先权数据

G06F 8/60 (2018.01)

61/829,647 2013.05.31 US

G06F 8/76 (2018.01)

61/830,508 2013.06.03 US

G06F 9/445 (2018.01)

(85) PCT国际申请进入国家阶段日  
2016.02.01

(56) 对比文件

US 2006/0233114 A1, 2006.10.19

US 2005/0289538 A1, 2005.12.29

(86) PCT国际申请的申请数据  
PCT/IB2014/002067 2014.05.30

CN 102360308 A, 2012.02.22

(87) PCT国际申请的公布数据  
W02015/019183 EN 2015.02.12

Vordel.Next-Generation API Delivery Platform.《http://www.cloudcomputingintelligence.com/component/phocadownload/category/1-latest-white-papers?download=7:api-delivery-platform》.2012,

(73) 专利权人 沃德尔有限公司  
地址 爱尔兰都柏林

审查员 葛运滨

(72) 发明人 M·科林斯 M·威姆斯  
D·麦肯纳 P·爱德华兹

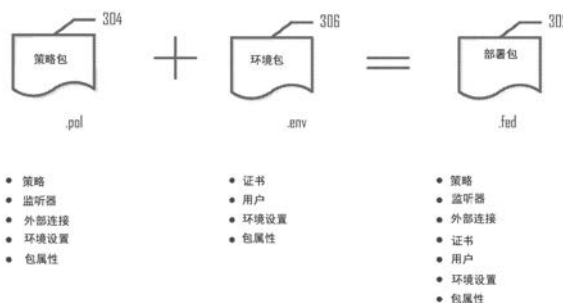
权利要求书4页 说明书11页 附图6页

## (54) 发明名称

管理应用编程接口 (API) 服务器配置在软件生命周期部署中的一系列部署

## (57) 摘要

提供了用于管理应用编程接口 (API) 服务器配置的一系列部署的示例方法、系统、和技术。示例方法包括定义API服务器的第一配置。该第一配置包括部署包,其至少将所定义的第一配置的策略、监听器以及外部连接组件与特别针对其开发环境部署中的API服务器的操作的环境设置编码在一起。方法还包括准备API服务器的第二配置。第二配置包括 (i) 从第一配置中推导出的策略包以及 (ii) 特别针对API服务器的测试环境部署的可分开的环境包。方法进一步包括准备API服务器的第三配置。该第三配置包括 (i) 推导出的策略包以及 (ii) 特别针对API服务器的产品环境部署的可分开的环境包。



1. 一种用于管理应用编程接口 (API) 服务器配置的一系列部署的方法,所述方法包括:  
定义所述API服务器的第一配置,所述第一配置被实现为部署包的计算机可读介质编码,所述部署包至少将所定义的第一配置的策略组件、监听器组件以及外部连接组件与对应于所述API服务器在所述API服务器的开发环境部署中的操作的环境设置编码在一起;  
准备所述API服务器的第二配置,所述第二配置被实现为以下的计算机可读介质编码 (i) 从所述第一配置中推导出的策略包以及 (ii) 对应于所述API服务器的测试环境部署的可分开的环境包;以及  
准备所述API服务器的第三配置,所述第三配置被实现为以下的计算机可读介质编码 (i) 基于所述第一配置的所述推导出的策略包以及 (ii) 对应于所述API服务器的产品环境部署的可分开的环境包,  
其中所述开发环境部署的环境设置和对应于所述API服务器的测试和产品环境部署的环境包相对于它们相应的部署环境规定不同的外部资源定位符、加密证书以及用户认证参数。
2. 如权利要求1所述的方法,其特征在于,进一步包括:  
不改变地将所述API服务器的策略组件、监听器组件以及外部连接组件从API服务器的开发环境部署推进到后续的测试和产品环境部署;以及  
通过创建针对后续的测试和产品环境部署的环境包来创建针对后续的测试和产品环境部署的API服务器的实例。
3. 如权利要求1所述的方法,其特征在于,进一步包括:  
通过创建针对相应的产品环境部署的环境包来创建针对相应的产品环境部署的API服务器的多个实例。
4. 如权利要求3所述的方法,其特征在于,  
针对相应的产品环境部署而创建的多个API服务器实例中的至少一些被相关联作为负载均衡或失效备援组的成员。
5. 如权利要求2所述的方法,其特征在于,  
所述API服务器的经推进的策略组件、监听器组件以及外部连接组件被实现在API服务器的后续测试和产品环境部署中作为基于所述第一配置的所述推导出的策略包的与环境无关的版本的一部分。
6. 如权利要求1所述的方法,其特征在于,  
所述第一配置能在所述开发环境中执行。
7. 如权利要求1所述的方法,其特征在于,进一步包括:  
基于所述API服务器在测试和产品环境中的一个或多个中的执行,更新所述开发环境部署包并且基于所述第一配置重新推导针对与所述API服务器的测试和产品环境部署中的一者或两者的相应环境包有联系的重部署的策略包。
8. 如权利要求1所述的方法,其特征在于,所述环境包中对应于所述测试环境部署的环境设置替代对应于所述开发环境部署的环境设置。
9. 如权利要求8所述的方法,其特征在于,所述环境包中对应于所述产品环境部署的环境设置替代对应于所述测试环境部署的环境设置。
10. 如权利要求1所述的方法,其特征在于,开发和测试环境被部署在同一域中。

11. 如权利要求1所述的方法,其特征在于,所述策略组件包括至少定义所述API服务器的部分操作的策略规则定义,并且其中所述策略规则组件中的至少一些包括实施安全性、依从性、以及操作策略的环境设置。

12. 如权利要求1所述的方法,其特征在于,所述监听器组件创建与所述API服务器的客户端有关的协议交互,并且其中所述监听器组件包括接受使用与所述策略组件兼容的协议的请求的环境设置。

13. 如权利要求12所述的方法,其特征在于,所述协议包括HTTP或JMS。

14. 如权利要求1所述的方法,其特征在于,所述外部连接组件创建到一节点的连接,所述节点存储了影响所述API服务器的操作的数据,并且其中所述外部连接组件包括环境设置,该环境设置被查询以认证所述API服务器的客户端。

15. 如权利要求14所述的方法,其特征在于,所述节点包括数据库或认证存储库。

16. 如权利要求1所述的方法,其特征在于,响应于模拟的API业务和外部配置查找,所述第一配置在所述开发环境中执行。

17. 如权利要求1所述的方法,其特征在于,响应于更新针对所述测试环境的环境设置,所述第二配置在所述测试环境中执行。

18. 如权利要求1所述的方法,其特征在于,响应于更新针对所述产品环境的环境设置,所述第三配置在所述产品环境中执行。

19. 如权利要求1所述的方法,其特征在于,进一步包括:

接收因环境而异的输入;以及

基于接收到的因环境而异的输入来更新对应于所述API服务器的测试环境部署的可分开的环境包。

20. 如权利要求19所述的方法,其特征在于,进一步包括:

接收因环境而异的第二输入;以及

基于接收到的因环境而异的第二输入来更新对应于所述API服务器的产品环境部署的可分开的环境包。

21. 如权利要求2所述的方法,其特征在于,所述推进包括将包括所述API服务器的策略组件、监听器组件和外部连接组件的配置文件传输到目标环境。

22. 如权利要求21所述的方法,其特征在于,所述传输包括使用文件传输协议来传输所述配置文件。

23. 如权利要求2所述的方法,其特征在于,所述推进包括将包括所述API服务器的策略组件、监听器组件和外部连接组件的配置文件加载到存储库,以及从所述存储库检索所述配置文件,其中所述配置文件能在开发和测试环境中执行。

24. 如权利要求2所述的方法,其特征在于,所述推进包括将包括所述API服务器的策略组件、监听器组件和外部连接组件的第一配置加载到存储库,从所述存储库检索所述第一配置,从检索到的第一配置中提取所述环境设置,以及将提取到的环境设置编码成对应于所述API服务器的测试环境部署的环境包。

25. 如权利要求24所述的方法,其特征在于,所述推进包括从所述存储库检索所述第一配置,从检索到的第一配置中提取所述环境设置,以及将提取到的环境设置编码成对应于所述API服务器的产品环境部署的环境包。

26. 如权利要求1所述的方法,其特征在于,进一步包括:

接收用户对于是对应于所述开发环境部署中的API服务器的操作的环境设置的策略组件、监听器组件以及外部连接组件的选择;以及

接收每一个用户选择的输入值,其中所述输入值确保所述第一配置保持能部署在所述开发环境部署中,

其中准备所述API服务器的第二配置包括将包括所述用户选择和输入值的策略包导出到盘。

27. 如权利要求26所述的方法,其特征在于,进一步包括:

获得所述加密证书以及所述用户认证参数,其中准备所述API服务器的第二配置包括将包括所获得的加密证书以及用户认证参数的可分开的环境包导出到盘。

28. 如权利要求26所述的方法,其特征在于,进一步包括:

基于输入值来更新所述API服务器的第一配置;

获得从所述API服务器的经更新的第一配置中推导出的经更新的策略包;

更新所述API服务器的第二配置,其中更新所述第二配置包括将对应于所述API服务器的测试环境部署的可分开的环境包与经更新的策略包合并;

接收针对经更新的策略包的新的环境设置的第二输入值;以及

使用接收到的第二输入值来更新所述新的环境设置的输入值。

29. 如权利要求28所述的方法,其特征在于,更新所述API服务器的第二配置包括添加证书、密钥、用户、或用户组。

30. 如权利要求28所述的方法,其特征在于,更新所述API服务器的第二配置包括移除证书、密钥、用户、或用户组。

31. 如权利要求28所述的方法,其特征在于,进一步包括:

更新所述API服务器的第三配置,其中更新所述第三配置包括将对应于所述API服务器的产品环境部署的可分开的环境包与经更新的策略包合并;

接收针对经更新的策略包的新的环境设置的第三输入值;以及

使用接收到的第三输入值来更新所述新的环境设置的输入值。

32. 如权利要求1所述的方法,其特征在于,所述策略包的方案包括:访问控制规则、内容过滤、威胁保护、配额管理、请求的完整性和可信性、转换、审计、或路由规则。

33. 一种计算机可读存储介质,所述计算机可读存储介质包括存储于其上的计算机程序,所述计算机程序包括:

能在第一计算系统的处理器上执行的指令,所述指令用于提供图形策略工作室,其中策略开发者用户可执行(i)如权利要求1中所述的方法的定义步骤,以及(ii)对所述第一配置的虚拟化实例的执行以及(iii)对部署包的计算机可读介质编码的导出;

能在第二计算系统的处理器上执行的指令,所述指令用于提供配置工作室,其中部署用户可执行(i)如权利要求1中所述的方法的与所述API服务器的第二和第三配置中的一者或两者有关的准备步骤。

34. 一种用于管理应用编程接口(API)的一系列部署的系统,所述系统包括:

定义API服务器的第一配置的第一部署引擎,其中所述第一配置包括部署包,所述部署包至少将所定义的第一配置的策略组件、监听器组件以及外部连接组件与对应于所述API

服务器在所述API服务器的开发环境部署中的操作的环境设置编码在一起；

准备所述API服务器的第二配置的第二部署引擎,所述第二配置包括(i)从所述第一配置中推导出的策略包以及(ii)对应于所述API服务器的测试环境部署的可分开的环境包；以及

准备所述API服务器的第三配置的第三部署引擎,其中所述第三配置包括(i)基于所述第一配置的所述推导出的策略包以及(ii)对应于API服务器的产品环境部署的可分开的环境包,

其中针对开发环境部署的环境设置以及对应于所述API服务器的测试和产品环境部署的环境包指定相对于它们相应的部署环境的不同的外部资源定位符、加密证书和用户认证参数。

35. 如权利要求34所述的系统,其特征在于,进一步包括:

推进引擎,所述推进引擎不改变地将所述API服务器的策略组件、监听器组件以及外部连接组件从API服务器的开发环境部署推进到后续的测试和产品环境部署,以及通过创建针对后续的测试和产品环境部署的环境包来创建针对后续的测试和产品环境部署的API服务器的实例。

36. 如权利要求34所述的系统,其特征在于,所述API服务器的经推进的策略组件、监听器组件以及外部连接组件被实现在API服务器的后续的测试和产品环境部署中作为基于所述第一配置的所述推导出的策略包的与环境无关的版本的一部分。

## 管理应用编程接口 (API) 服务器配置在软件生命周期部署中的一系列部署

[0001] 相关申请的交叉引用

[0002] 本申请要求编号为61/830508、题为“Environmentalization Technique for Promotion of Application Programming Interface (API) Server in Lifecycle Succession of Deployments” (用于在部署的生命周期演替中促进应用编程接口 (API) 服务器的环境化技术)、2013年6月3日提交的美国临时申请以及编号为61/829647、“Environmentalization Technique for Promotion of Application Programming Interface (API) Server in Lifecycle Succession of Deployments” (用于在部署的生命周期演替中促进应用编程接口 (API) 服务器的环境化技术)、2013年5月31日提交的美国临时申请的优先权,上述申请各自被通过引用合并与此。

### 背景技术

#### 技术领域

[0003] 本发明一般地涉及分布式计算系统中的接口,尤其涉及用于中介、转换、守护和/或管理应用编程接口 (API) 的网关和/或代理型工具。

[0004] 相关技术

[0005] 应用编程接口 (API) 已被长期用来规定软件组件如何与彼此交互。通过定义API,程序员或系统架构师可规定可由一个组件用来与另一组件交互的调用、方法调用和/或消息,这通常是在不展示API“背后”的软件、数据和协作的内部构件的情况下进行的。

[0006] 随着每天部署的web和移动应用的数量和种类的增加,并且考虑到针对企业应用的多样的生态系统的基于云的存储、应用以及服务平台的持续增长的重要性,对于这些应用和平台用来协作的接口 (API) 的管理是一项不断扩大的挑战。展示和/或使用API的应用和服务的纯粹组合式的激增产生了显著的生命周期管理方面的挑战,而对于安全性、可用性以及缩放性要求的需求需要针对API部署的综合性方法。

[0007] 一种由Vordel (现在是Axway, Inc. (Axway公司) 的一个部门) 推行的方法是部署API服务器运行时环境,其中多样的且不断进化的API集以及在这些API中穿行的信息事务可被部署、守护、监视、经受控制和监管等等。依赖于诸如Axway API服务器之类的产品,企业能够有效地部署和管理移动客户端应用用来与云中的企业数据和服务交互的API或者业务伙伴用来交换数据或执行事务的API。

[0008] 随着API服务器技术的普及性和市场渗透率的增长,需要经改善的方法来管理API部署的生命周期方面。在一些情况下,期望在API的开发、测试及产品部署的容易性和速度方面的改进。

[0009] 概述和描述

[0010] 已经发现API服务器配置的一系列部署可使用软件工具来管理,该软件工具允许使用预定义的、用户/企业定义的、和/或可扩展的构造块(例如,策略、过滤器、监听器和/或

外部连接构造块)集来对API服务器的初始定义(例如,在开发环境内),可针对这些构造块将其特定参数、对象、谓词等标识为特定于部署环境。以此方式,如此定义的API服务器配置的计算机可读介质编码可被分解为因环境而异的部分以及是(或可被视为)与环境无关的部分。当将一给定API服务器配置从开发环境部署推进至测试或产品环境部署时,该开发环境配置的计算机可读介质编码被“环境化”,以便将策略、过滤器、监听器和/或外部连接实例的环境中立方面与因环境而异的方面分开。一旦分开(例如,分成了环境中立策略包和环境包),API服务器配置可通过专门化环境包同时维护对于策略包的配置控制而被推进到部署的生命周期演替中的后续环境。

[0011] 在一些情况下,典型的企业级客户可能具有若干个环境,其中API服务器配置可能从开发移动到产品。例如,这可包括完全分开的开发、测试、以及产品域。包可根据策略开发者所开发的策略来被部署在这些环境中。策略开发者准备用于推进到上游环境(例如,测试和产品环境)的配置。这可包括决定哪些设置是因环境而异的,并且可假设具有策略开发和配置工具(例如Policy Studio)方面的专业知识。每个环境可由不同人员来处理。例如,策略开发者可负责开发环境,API服务器管理员可负责测试环境,而API服务器操作员可负责产品环境。API服务器管理员和API服务器操作员可不具有关于策略开发和配置工具方面的专业知识。因此,当配置被推进到上游时,API服务器管理员和API服务器操作员可能不能够在其相应环境中充分实施各策略。

[0012] 另外,为了标识哪些设置是因环境而异的以及哪些是与环境无关的,API服务器管理员或API服务器操作员可能需要导航服务器配置来标识什么需要被改变。这可能是耗时的,并且从上游用户的角度来看,这增加了API服务器的复杂性。

[0013] 在根据本方面的一些实施例中,一种管理API服务器配置的一系列部署的方法包括定义所述API服务器的第一配置,准备所述API服务器的第二配置,以及准备所述API服务器的第三配置。所述第一配置被实现为部署包的计算机可读介质编码,所述部署包至少将所定义的第一配置的策略、监听器以及外部连接组件与特别针对所述API服务器在所述API服务器的开发环境部署中的操作的环境设置编码在一起。所述第二配置被实现为以下的计算机可读介质编码(i)从所述第一配置中推导出的策略包以及(ii)特别针对所述API服务器的测试环境部署的可分开的环境包。所述第三配置被实现为以下的计算机可读介质编码(i)推导出的策略包以及(ii)特别针对所述API服务器的产品环境部署的可分开的环境包。所述开发环境部署的环境设置和特别针对所述API服务器的测试和产品环境部署的环境包相对于它们相应的部署环境规定不同的外部资源定位符、加密证书以及用户认证参数。

[0014] 在一些情况或实施例中,方法进一步包括(基本不改变地)将所述API服务器的策略、监听器以及外部连接组件从API服务器的开发环境部署推进到后续的测试和产品环境部署,以及通过专门化特别针对后续的测试和产品环境部署的环境包来专门化针对后续的测试和产品环境部署的API服务器的实例。在一些情况或实施例中,方法进一步包括通过专门化特别针对相应的产品环境部署的环境包来专门化针对相应的产品环境部署的API服务器的多个实例。针对相应的产品环境部署而专门化的多个API服务器实例中的至少一些被相关联作为负载平衡或故障切换组的成员。所述API服务器的经推进的策略、监听器以及外部连接组件被实现在API服务器的后续的测试和产品环境部署中作为推导出的策略包的与环境无关的版本的一部分。所述第一配置可执行在所述开发环境中。

[0015] 在一些情况或实施例中,方法进一步包括基于所述API服务器在测试和产品环境中的一个或多个中的执行,更新所述开发环境部署包并且重新推导针对与所述API服务器的测试和产品环境部署中的一者或两者的相应环境包有联系的重部署的策略包。环境包中特别针对测试环境部署的环境设置替代了特别针对开发环境部署的环境设置。此外,环境包中特别针对产品环境部署的环境设置替代了特别针对测试环境部署的环境设置。环境包中特别针对产品环境部署的环境设置替代了特别针对测试环境部署的环境设置。

[0016] 在一些情况或实施例中,开发和测试环境被部署在同一域中。此外,所述策略组件包括至少定义所述API服务器的部分操作的策略规则定义,并且其中所述策略规则组件中的至少一些包括实施安全性、依从性、以及操作策略的环境设置。所述监听器组件创建与所述API服务器的客户端有关的协议交互,并且其中所述监听器组件包括接受使用与所述策略组件兼容的协议的请求的环境设置。协议包括HTTP和/或JMS。外部连接组件创建到节点的连接,该节点存储影响API服务器的操作的数据。外部连接组件包括被查询以认证API服务器的客户端的环境设置。节点是数据库或认证存储库。

[0017] 在一些情况或实施例中,响应于模拟的API业务和外部配置查找,所述第一配置在所述开发环境中执行。另外,响应于更新针对所述测试环境的环境设置,所述第二配置在所述测试环境中执行。另外,响应于更新针对所述产品环境的环境设置,所述第三配置在所述产品环境中执行。

[0018] 在一些情况或实施例中,方法进一步包括接收因环境而异的输入并且基于所接收的因环境而异的输入来更新特别针对API服务器的测试环境部署的可分开环境包。方法还可包括接收因环境而异的输入并且基于所接收的因环境而异的第二输入来更新特别针对API服务器的产品环境部署的可分开环境包。

[0019] 在一些情况或实施例中,基本不改变地将所述API服务器的策略、监听器以及外部连接组件从API服务器的开发环境部署推进到后续的测试和产品环境部署包括将包括所述API服务器的策略、监听器和外部连接组件的配置文件传输到目标环境。传输到目标环境包括使用文件传输协议来传输所述配置文件。

[0020] 在一些情况或实施例中,基本不改变地将所述API服务器的策略、监听器以及外部连接组件从API服务器的开发环境部署推进到后续的测试和产品环境部署包括将包括所述API服务器的策略、监听器和外部连接组件的配置文件加载到存储库,以及从所述存储库检索所述配置文件。配置文件可在开发和测试环境中执行。

[0021] 在一些情况或实施例中,基本不改变地将所述API服务器的策略、监听器以及外部连接组件从API服务器的开发环境部署推进到后续的测试和产品环境部署包括将包括所述API服务器的策略、监听器和外部连接组件的第一配置加载到存储库,从所述存储库检索所述第一配置,从检索到的第一配置中提取所述环境设置,以及将提取到的环境设置编码成特别针对所述API服务器的测试环境部署的环境包。所述推进包括从所述存储库检索所述第一配置,从检索到的第一配置中提取所述环境设置,以及将提取到的环境设置编码成特别针对所述API服务器的产品环境部署的环境包。在一些情况或实施例中,方法进一步包括接收用户对于是特别针对所述开发环境部署中的API服务器的操作的环境设置的策略、监听器以及外部连接组件的选择以及接收每一个用户选择的输入值。所述输入值确保所述第一配置保持可部署在所述开发环境部署中,并且准备所述API服务器的第二配置包括将包

括所述用户选择和输入值的策略包导出到盘。方法还可包括获得所述加密证书以及所述用户认证参数。准备所述API服务器的第二配置包括将包括所获得的加密证书以及用户认证参数的可分开的环境包导出到盘。

[0022] 在一些情况或实施例中,方法进一步包括基于输入值来更新所述API服务器的第一配置,获得从所述API服务器的经更新的第一配置中推导出的经更新的策略包,以及更新所述API服务器的第二配置。更新所述第二配置包括将特别针对所述API服务器的测试环境部署的可分开的环境包与经更新的策略包合并。方法还可包括接收针对经更新的策略包的新的环境设置的第二输入值,以及使用接收到的第二输入值来更新所述新的环境设置的输入值。更新所述API服务器的第二配置包括添加包括证书、密钥、用户、以及用户组的组中的至少一者。更新所述API服务器的第二配置包括移除包括证书、密钥、用户、以及用户组的组中的至少一者。在一些情况或实施例中,所述策略包的方案包括包含以下各项的组中的至少一者:访问控制规则、内容过滤、威胁保护、配额管理、请求的完整性和可信性、转换、审计、以及路由规则。

[0023] 在一些情况或实施例中,物理计算机系统被提供用于执行本公开中所述的方法。在一个实施例中,编码在一个或多个介质中的计算机程序产品包括可在第一计算系统的处理器上执行的指令,所述指令用于提供图形策略工作室,其中策略开发者用户可执行(i)如权利要求1中所述的方法的定义步骤,以及(ii)对所述第一配置的虚拟化实例的执行以及(iii)对部署包的计算机可读介质编码的导出。计算机程序产品还包括可在第二计算系统的处理器上执行的指令,所述指令用于提供配置工作室,其中部署用户可执行(i)准备API服务器的第二配置,所述第二配置被实现为以下的计算机可读介质编码(a)从所述第一配置中推导出的策略包以及(b)特别针对所述API服务器的测试环境部署的可分开的环境包;或者(ii)准备API服务器的第三配置,所述第三配置被实现为以下的计算机可读介质编码(a)推导出的策略包以及(b)特别针对API服务器的产品环境部署的可分开的环境包。所述开发环境部署的环境设置和特别针对所述API服务器的测试和产品环境部署的环境包相对于它们相应的部署环境规定不同的外部资源定位符、加密证书以及用户认证参数。

[0024] 在根据本方面的一些实施例中,用于管理应用编程接口(API)的一系列部署的系统包括定义所述API服务器的第一配置的第一部署引擎。所述第一配置包括部署包,所述部署包至少将所定义的第一配置的策略、监听器以及外部连接组件与特别针对所述API服务器在所述API服务器的开发环境部署中的操作的环境设置编码在一起。系统还包括准备API服务器的第二配置的第二部署引擎。所述第二配置包括(i)从所述第一配置中推导出的策略包以及(ii)特别针对所述API服务器的测试环境部署的可分开的环境包。系统还包括准备API服务器的第三配置的第二部署引擎。所述第三配置包括(i)推导出的策略包以及(ii)特别针对API服务器的产品环境部署的可分开的环境包。针对开发环境部署的环境设置以及特别针对所述API服务器的测试和产品环境部署的环境包指定相对于它们相应的部署环境的不同的外部资源定位符、加密证书和用户认证参数。

[0025] 在一些情况或实施例中,系统进一步包括推进引擎,所述推进引擎基本不改变地将所述API服务器的策略、监听器以及外部连接组件从API服务器的开发环境部署推进到后续的测试和产品环境部署,以及通过专门化特别针对后续的测试和产品环境部署的环境包来专门化针对后续的测试和产品环境部署的API服务器的实例。所述API服务器的经推进的

策略、监听器以及外部连接组件被实现在API服务器的后续测试和产品环境部署中作为推导出的策略包的与环境无关的版本的一部分。

[0026] 虽然本发明的各创新方面在以下的描述和权利要求书中被具体化,但是阅读了本公开以及以下参考文件中公开的各细节的本领域的技术人员将理解本发明的各示例性设计选择和具体的实现:

- [0027] • Axway API服务器,概念指南(版本7.2.0)(附件A),
- [0028] • Axway API服务器,部署和推进指南(版本7.2.0)(附件B),
- [0029] • Axway API管理方案包,API管理指南(版本7.2.0)(附件C),以及
- [0030] • Axway API服务器开发者指南(版本7.2.0)(附件D),以及
- [0031] • 环境化示例(附件E),
- [0032] 以上的每一个被附连与此并被明确纳入到本公开中。

### 附图说明

[0033] 通过参照附图,能更好地理解本发明,而且使本发明的许多目标、特征和优点对本领域技术人员是显然的。

[0034] 图1解说了根据一实施例的正被推进到测试域并进一步推进到产品域的开发域中的两个API组。

[0035] 图2是根据一实施例的用于定义服务器配置的图形配置工具的截屏。

[0036] 图3解说了根据一实施例的包括策略包和环境包的部署包。

[0037] 图4是根据一实施例的用于标记服务器配置的各字段的图形配置工具的截屏。

[0038] 图5是根据一实施例的用于管理API服务器配置的一系列部署的方法的简化流程图。

[0039] 图6是适用于本公开的一个或多个实施例的计算机系统的框图。

[0040] 在不同附图中,相同的附图标记的使用指示相似或完全相同的项。

### 具体实施方式

[0041] 本公开提供了用于部署和推进API服务器配置以供管理API服务器配置的一系列部署的技术。

[0042] 用户可具有若干个环境,服务器配置将通过这些环境从开发移动到产品。这可包括完全分开的开发、测试、以及产品域。推进指的是将服务器配置从一个环境移动到另一环境以及配置因环境而异的值使得该配置能够被部署到特定环境中的动作。

[0043] 部署可指的是部署针对本地域中的API服务器组的配置。域可以是用于管理各个API服务器组的一个单独的管理实体。在一个示例中,产品环境被实现为一个单独的产品域,仅产品操作员工能够访问该产品域。

[0044] 图1解说了根据一实施例的正被推进到测试域112并进一步推进到产品域114的开发域110A和110B中的两个API组。

[0045] 在图1中,API组106在开发域110A中,而API组108在开发域110B中。每个环境可被实现为一个单独的API服务器域。在一个示例中,开发者工作在他们自己的开发环境中,并且随后将他们的API服务器配置推进到集中测试团队,集中测试团队在单个测试环境中执

进行测试。当测试完成时,测试团队将API服务器配置推进到产品操作团队,以供部署在产品环境中。开发、测试、以及产品操作团队能够访问他们各自的环境。虽然本公开描述了三个不同环境(开发、测试、以及产品),少于或多于三个的环境也落在本公开的范围之内。此外,一实施例可具有两个或更多个的任一特定环境。例如,一实施例可包括开发环境和两个测试环境。

[0046] API服务器可以是灵活的并且与任何体系结构一起工作。在一个示例中,API服务器不授权具体的环境对域的配置。开发和测试团队可工作在同一环境中,具有对于所有API服务器配置的共同访问。在这种情况下,针对开发和测试API服务器组可能存在单个域。服务器配置被部署到一个API服务器组,并且每个域包括适当的API服务器组以运行该配置。

[0047] 服务器配置可在各环境之间推进。在一个示例中,策略开发者准备用于推进到上游环境(例如,测试和产品环境)的配置。这可包括决定哪些设置是因环境而异的,并且假设具有策略开发和配置工具(例如Policy Studio)方面的专业知识。策略开发者可通过使用例如配置工具来定义API服务器的配置。服务器配置可包括不同配置类型,诸如策略、监听器、外部连接、证书、用户、环境设置、以及包属性组件。应当理解,其它配置类型也落在本公开的范围之内。

[0048] 上游用户(例如,API服务器管理员)接受策略开发者所准备的配置,创建因环境而异的配置,并且部署该配置。上游用户可使用工具(诸如Configuration Studio),该工具不假设上游用户具有策略开发和配置方面的专业知识。

[0049] 图2是根据一实施例的用于定义服务器配置的图形配置工具的截屏。虽然配置工具被描述为图形用户界面,但是其他用户界面(例如,外壳)也落在本公开的范围之内。

[0050] 在图2中,用于定义服务器配置的图形配置工具可使得策略开发者能够虚拟化API和开发策略。策略可实施安全性、依从性、和/或操作要求。图形配置工具可包括各种特征,诸如用于方便开发和维护的流程图样式可视化、使得策略开发者能够将过滤器(处理规则)拖动到策略画布上并配置它们的图形拖放用户界面、以及用于构建强大策略的广泛的过滤器库。

[0051] 在一个实施例中,服务器配置被实现为开发包的计算机可读介质编码,其至少将所定义的配置的策略、监听器以及外部连接组件与其开发环境部署中的特定于API服务器的操作的环境设置编码在一起。

[0052] 图3解说了根据一实施例的包括策略包304和环境包306的开发包302。

[0053] 在图3中,环境包306包括证书和用户以及环境设置和包属性。应当理解,环境包306可包括更少或更多的设置和/或属性。环境包306具有.env后缀。

[0054] 策略包304包括策略、监听器和外部连接组件以及环境设置和包属性。应当理解,策略包304可包括更少或更多的设置和/或属性。策略包304具有.pol后缀。策略组件可包括至少定义API服务器的部分操作的策略规则定义。另外,策略规则组件中的至少一些可包括实施安全性、依从性、以及操作策略的环境设置。监听器组件可创建关于API服务器的客户端的协议交互。另外,监听器组件可包括环境设置,环境设置接受使用与策略组件兼容的协议的请求。在一个示例中,协议包括HTTP和/或JMS。外部连接组件可创建到节点的连接,该节点存储影响API服务器的操作的数据。另外,外部连接组件可包括被查询以认证API服务器的客户端的环境设置。在一个示例中,节点是数据库或认证存储库。

[0055] 图形配置工具可接收用户对于是特别针对开发环境部署中API服务器的操作的环境设置的策略、监听器以及外部连接组件的选择。图形配置工具可接收每一个用户选择的输入值。输入值可确保针对开发部署的配置在开发环境部署中保持可部署。

[0056] 策略包304可包括由API服务器所虚拟化的API的细节。细节可包括源消息的协议(例如,HTTP、HTTPS、JMS、FTP等)、访问控制规则(例如,认证和授权)、内容过滤、威胁保护、配额管理、请求的完整性和可信性、转换、审计、以及路由规则。

[0057] 环境包306可包括环境设置和与环境无关的设置。图形配置工具可指导策略开发者分解环境设置和/或与环境无关的设置并且相应地接受来自策略开发者的输入。例如,图形配置工具可包括针对策略开发者这样做的提醒。

[0058] 策略开发者可将字段标记为环境设置。开发环境部署包括由策略开发者标记为环境设置的任何字段。被环境化的字段被标记为环境设置。被标记的字段可包括连接细节、策略流决策(例如如果在调用完整审计策略的产品中),用于SSL的证书、完整性检查和消息级加密、将请求路由到何处、以及在何处认证和/或授权请求和/或用户。

[0059] 服务器配置中的一个或多个字段可被标记为环境设置和/或与环境无关的设置。在一个实施例中,字段或者针对环境化来标记,或者不针对环境化来标记。在一个示例中,图形配置工具使得策略开发者能够将服务器配置中的一个或多个字段标记为专用于开发环境部署的环境设置。在这一示例中,被标记为环境设置的针对字段的指针被放到策略包304中,并且该字段的因开发环境而异的值被放到环境包306中。字段的值通常不流向上游环境,但是可能在其它示例中流向上游环境。测试和产品环境可能具有作为输入的策略包304(例如,在第一循环推进期间),并且具有来自策略包304中的用于确定需要哪些字段的信息。

[0060] 在另一示例中,图形配置工具可使得策略开发者能够将服务器配置中的一个或多个字段标记为不是专用于开发环境部署的环境设置。在一个实施例中,如果策略开发者正移除之前被标记为环境设置的字段,则对于该之前被标记为环境设置的字段的指针被从策略包304中移除并且任何相关联的因环境而异的值被从环境包306中移除。

[0061] 图4是根据一实施例的用于标记服务器配置的各字段的图形配置工具的截屏。在图4中,反病毒扫描类型和干净类型被环境化。

[0062] 在接收到合适的输入值之后,图形配置工具可创建策略包304和环境包306。在一个实施例中,服务器配置被存储在文件中(例如,XML文件)。当策略开发者将服务器配置的某一字段标记为环境设置或者与环境无关的设置时,该文件被相应地操作。

[0063] 服务器配置可被推送到API服务器并且在开发环境中执行。策略开发者可继续开发和改变开发环境中的API。

[0064] 当策略开发者满意开发环境的配置时,策略开发者可决定将服务器配置从开发环境推进到另一环境(例如,测试或产品环境)。策略开发者可将该配置移交给上游用户(例如,API服务器管理员或API服务器操作员)。取代让上游用户来寻找特别针对开发环境部署中的API服务器的操作的环境设置,上游用户可使用使得上游用户能够容易地找到这些设置并提供这些设置的值的上游配置工具。这可为上游用户节省时间。

[0065] 在一个实施例中,API服务器的针对测试环境部署的配置被准备。API服务器的针对测试环境部署的配置可被实现为以下的计算机可读介质编码(i)从开发环境的配置中推

导出的策略包以及 (ii) 特别针对API服务器的测试环境部署的可分开的环境包。API服务器的经推进的策略、监听器以及外部连接组件可被实现在其后续部署中 (例如, 后续的测试和产品环境部署), 作为推导出的策略包的与环境无关的版本的一部分。在一个实施例中, 策略包变成不可由上游用户更改的只读包。这可确保策略包不改变并且就像是上游一样被推进。

[0066] 上游配置工具可协助将服务器配置从开发环境推进到测试环境。在一个示例中, 配置工具生成环境包306, 而另一应用 (例如, 基于Web的应用) 将人工产物部署到API服务器。在另一示例中, 策略包可被导入到上游配置工具中, 该工具将上游环境的配置推送到API服务器供执行。

[0067] 服务器配置的推进可包括基本不改变地将API服务器的策略、监听器以及外部连接组件从开发环境部署推进到其后续的测试和产品环境部署。这可为上游用户提供优点, 因为对于上游用户可不需要更改策略包中与环境无关的设置的字段或者筛选这些设置来寻找因环境而异的设置以改变这些设置。与环境无关的字段可以是在各环境之间静态的并且被保留不被改变。为了指导用户输入执行策略包和可分开的环境包所需的环境设置, 上游配置工具可突出显示环境设置, 使得用户知晓执行上游环境中的上游部署包需要什么信息。

[0068] 测试环境的上游用户可输入测试环境的被突出显示的环境设置的值。上游配置工具可接收因环境而异的输入并且基于所接收的因环境而异的输入来更新特别针对API服务器的测试环境部署的可分开环境包。上游配置工具可生成测试环境的分开的环境包。环境包中特别针对测试环境部署的环境设置替代了特别针对开发环境部署的环境设置。在一个实施例中, 环境包306的“.env”文件中的值不从一个环境移动到另一个环境, 以便被替换。例如, 在第一周期推进期间, “.env”文件中的值可以是空白的, 并且管理员 (例如, 测试或产品管理员) 将值填入该空白以生成可部署的产品。

[0069] 测试环境的配置可在测试环境中执行, 例如响应于开发环境的开发环境设置被更新为测试环境设置并被推送给API服务器。

[0070] 服务器配置可从测试环境推进到产品环境。在一个示例中, API服务器的针对产品环境的配置被准备。在一个示例中, 产品环境的配置被实现为以下的计算机可读介质编码 (i) 推导出的策略包以及 (ii) 特别针对API服务器的产品环境部署的可分开环境包。

[0071] 在一个示例中, 产品环境的上游用户可使用上游配置工具来输入产品环境的环境设置的值。上游配置工具可突出显示这些环境设置以使其对于上游用户来说更容易被标识。上游配置工具接收因环境而异的输入并且基于所接收的因环境而异的输入来更新特别针对API服务器的产品环境部署的可分开环境包。上游配置工具生成产品环境的分开的环境包。环境包中特别针对产品环境部署的环境设置替代了特别针对测试环境部署的环境设置。产品环境的服务器配置可在产品环境中执行, 例如响应于测试环境的测试环境设置被更新为产品环境设置并被推送给API服务器。

[0072] 服务器配置可被导出到上游配置工具, 例如通过将包括API服务器的策略、监听器和外部连接组件的配置文件传输到目标环境。配置文件可使用文件传输协议 (例如, FTP (文件传输协议)) 来传输。

[0073] 各不同环境可以或可以不经由网络连接。在一个示例中, 测试环境可不具有用于

经由网络获得部署包的网络连通。如果上游环境不具有网络连通。在另一示例中,推进服务器配置包括将包括API服务器的策略、监听器和外部连接组件的配置文件加载到存储库,以及从存储库检索配置文件。配置文件可在开发和测试环境中执行。

[0074] 准备API服务器的测试环境的配置可包括将包括用户选择和输入值的策略包导出到盘。另外,准备API服务器的测试环境的配置可包括将包括加密证书和用户认证参数的可分开环境包导出到盘。

[0075] 在另一示例中,推进服务器配置包括将包括API服务器的策略、监听器和外部连接组件的开发环境的配置加载到存储库,从存储库检索开发环境的配置,从检索到的配置中提取开发环境的环境设置,以及将提取到的环境设置编码成特别针对API服务器的测试环境部署的环境包。开发环境的配置可从存储库检索,并且环境设置可从检索到的配置中提取。提取到的环境设置可被编码成特别针对API服务器的产品环境部署的环境包。

[0076] 此外,开发环境部署的环境设置和特别针对API服务器的测试和产品环境部署的环境包可相对于它们相应的部署环境规定不同的外部资源定位符、加密证书以及用户认证参数。加密证书以及用户认证参数可从合适的包中获得。

[0077] 此外,专门化特别针对API服务器的环境包可包括专门化针对后续的测试和产品环境部署的API服务器的实例。专门化特别针对API服务器的环境包还可包括专门化针对相应的产品环境部署的API服务器的多个实例。针对相应的产品环境部署而专门化的多个API服务器实例中的至少一些可被相关联作为负载平衡或故障切换组的成员。

[0078] 基于API服务器在测试和产品环境中的一个或多个中的执行,开发环境部署包可被更新并且针对与API服务器的测试和产品环境部署中的一者或两者的相应环境包有联系的重部署的策略包可被重新推导。

[0079] 当API服务器的服务器配置已经被推进时,用户可决定改变例如策略包中的字段。为此,用户可使用配置工具来改变开发环境中的设置。为了推进开发环境的经更新的配置,上游用户可使用上游配置工具,上游配置工具接受来自用户的对于上游环境的环境设置的输入。响应于用户的输入,配置工具可基于输入值更新针对开发环境的API服务器的配置。

[0080] 上游配置工具获得从针对开发环境的API服务器的经更新的配置中推导出的经更新的策略包并合并特别针对上游环境的经更新的策略包。在一个示例中,上游环境是测试环境,并且服务器配置从开发环境推进到测试环境。上游配置工具可通过例如将特别针对API服务器的测试环境配置的可分开的环境包与经更新的策略包合并来更新针对测试环境的API服务器的配置。上游配置工具可提示用户输入针对经更新的策略包的环境设置的输入值并且环境设置可使用用户的输入值来更新。

[0081] 在另一示例中,上游环境是产品环境,并且服务器配置从测试环境推进到产品环境。上游配置工具可通过例如将特别针对API服务器的产品环境配置的可分开的环境包与经更新的策略包合并来更新针对产品环境的API服务器的配置。上游配置工具可提示用户输入针对经更新的策略包的环境设置的输入值并且环境设置可使用用户的输入值来更新。

[0082] 更新针对测试环境的API服务器的配置可包括添加证书、密钥、用户、和/或用户组。更新针对测试环境的API服务器的配置可包括移除证书、密钥、用户、和/或用户组。

[0083] 开发、测试、和产品环境可以相同或不同。在一个示例中,产品环境部署被部署在物理硬件中,而开发环境部署和/或测试环境部署在虚拟环境中执行。

[0084] 图5是根据一实施例的用于管理应用编程接口 (API) 服务器配置的一系列部署的方法500的简化流程图。方法500并不意图是限制性的,而是可被使用在其它应用中。

[0085] 方法500包括步骤510—530。在步骤510中,API服务器的第一配置被定义,其中第一配置被实现为部署包的计算机可读介质编码,所述部署包至少将所定义的第一配置的策略、监听器以及外部连接组件与特定针对所述API服务器在所述API服务器的开发环境部署中的操作的环境设置编码在一起。在步骤520中,API服务器的第二配置被准备,其中第二配置被实现为以下的计算机可读介质编码(i)从第一配置中推导出的策略包以及(ii)特别关于API服务器的测试环境部署的可分开的环境包。在步骤530中,API服务器的第三配置被准备,其中第三配置被实现为以下的计算机可读介质编码:(i)推导出的策略包以及(ii)特别针对API服务器的产品环境部署的可分开的环境包,并且其中针对开发环境部署的环境设置以及特别针对API服务器的测试和产品环境部署的环境包指定相对于它们相应的部署环境的不同外部资源定位符、加密证书和用户认证参数。

[0086] 还可理解,附加的方法步骤可在以上讨论的步骤510—530之前、期间、或之后执行。还可理解,此处描述的方法500的步骤中的一个或多个可根据需要被忽略、组合、或以不同顺序来执行。

[0087] 图6是适用于本公开的一个或多个实施例的计算机系统600的框图。在各个实现中,配置可在计算机系统600的一个或多个处理器上执行。提供其中执行该配置的环境的API服务器可包括计算机系统(例如,计算机系统600)的一个或多个处理器。计算机系统可另外分别从由以下各项组成的组中选择一个或多个存储设备:软盘、柔性盘、硬盘、磁带、任何其它磁性介质、CD-ROM、任何其它光学介质、RAM、PROM、EPROM、FLASH-EEPROM、任何其它存储器芯片或磁带盒、和/或处理器或计算机被适配成读取的任何其它介质。该一个或多个存储设备可包括存储的信息,该信息可对于使用计算机网络(未示出)耦合到客户端或服务器的一个或多个计算设备和/或计算机程序(例如客户端)可用。计算机网络可以是任何类型的网络,包括能够将系统中的计算设备和/或计算机程序互联的LAN、WAN、内联网、互联网、云、和/或以上网络的任何组合。

[0088] 计算机系统600包括总线602或其它用于在计算机系统600的各个组件之间交流信息数据、信号、以及信息的其它通信机制。组件包括处理用户动作(诸如选择小键盘/键盘中的键、选择一个或多个按钮或链接等)的输入/输出(I/O)组件604,并且发送相应信号给总线602。I/O组件604还可包括诸如显示器611之类的输出组件,以及诸如光标控制器613(诸如键盘、小键盘、鼠标等)之类的输入控制器。还可包括可选的音频输入/输出组件605用于通过将音频信号转换成信息信号来允许用户使用语音来输入信息。音频I/O组件605可允许用户收听音频。收发机或网络接口606经由到网络的通信链路618在计算机系统600和其它设备之间发送和接收信号。在一个实施例中,传送是无线的,当然其它传送介质和方法也是适合的。处理器612(其可以是微控制器、数字信号处理器(DSP)、或其它处理组件)处理这种信号,诸如用于显示在计算机系统600上或者经由通信链路618传送到其它设备。处理器612还可控制将信息(诸如cookies或IP地址)传送给其它设备。

[0089] 计算机系统600的组件还包括系统存储器组件614(例如RAM)、静态存储组件616(例如ROM)、和/或盘驱动器617。计算机系统600通过处理器和其它组件通过执行包含在系统存储器组件614中的一个或多个指令序列来执行特定操作。逻辑可被编码在计算机可读

介质中,计算机可读介质可指参与向处理器612提供指令以供执行的任何介质。这样的介质可采取许多形式,包括但不限于非易失性介质、易失性介质和传输介质。在各个实现中,非易失性介质包括光盘或磁盘、或固态驱动器,易失性介质包括动态存储器(诸如系统存储器组件614),而传输介质包括同轴线缆、铜线、以及光纤,包括包含总线602的线缆。在一个实施例中,逻辑被编码在非瞬态计算机可读介质中。在一个示例中,传输介质也可以采用声波或光波的形式,诸如那些在无线电波、光、和红外数据通信期间生成的波。

[0090] 计算机可读介质的一些常见形式包括例如软盘、柔性盘、硬盘、磁带、任何其它磁性介质、CD-ROM、任何其它光学介质、穿孔卡片、纸带、具有孔图案的任何其它物理介质、RAM、PROM、EPROM、FLASH-EEPROM、任何其它存储器芯片或磁带盒、或者计算机被适配成从中读取内容的任何其它介质。

[0091] 在本公开的各个实施例中,用于实施本公开的指令序列的执行可由计算机系统600执行。在本公开的各个其它实施例中,通过通信链路618耦合到网络(例如,诸如LAN、WLAN、PTSN、和/或各种其它有限或无线网络,包括电信、移动、以及蜂窝电话网络)的多个计算机系统600可执行用于彼此协作实施本公开的指令序列。

[0092] 在合适的情况下,本公开所提供的各个实施例可使用硬件、软件或硬件和软件的组合来实现。同样在合适的情况下,此处提出的各种硬件组件和/或软件组件可被组合成包括软件、硬件、和/或两者的合成组件,而不背离本公开的精神。在合适的情况下,此处提出的各种硬件组件和/或软件组件可被分割成包括软件、硬件、和/或两者的子组件,而不背离本公开的精神。另外,在合适的情况下,设想了软件组件可实现为硬件组件,反之亦然。

[0093] 根据本公开的应用软件可被存储在一个或多个计算机可读介质上。还设想了此处所标识的应用软件可使用(联网的和/或其它方式的)一个或多个通用或专用计算机和/或计算机系统来实现。在合适的情况下,此处描述的各步骤的顺序可被改变、组合成合成步骤、和/或分割成子步骤以提供此处所描述的特征。

[0094] 前述公开不旨在将本公开限制为所公开的精确的使用形式或特定使用领域。因此,设想了本公开的各种替代实施例和/或修改,无论此处是明示地描述还是暗示,根据本公开都是可能的。可以以不背离本公开的范围的形式和细节来作出改动。因此,本公开仅受权利要求书的限制。

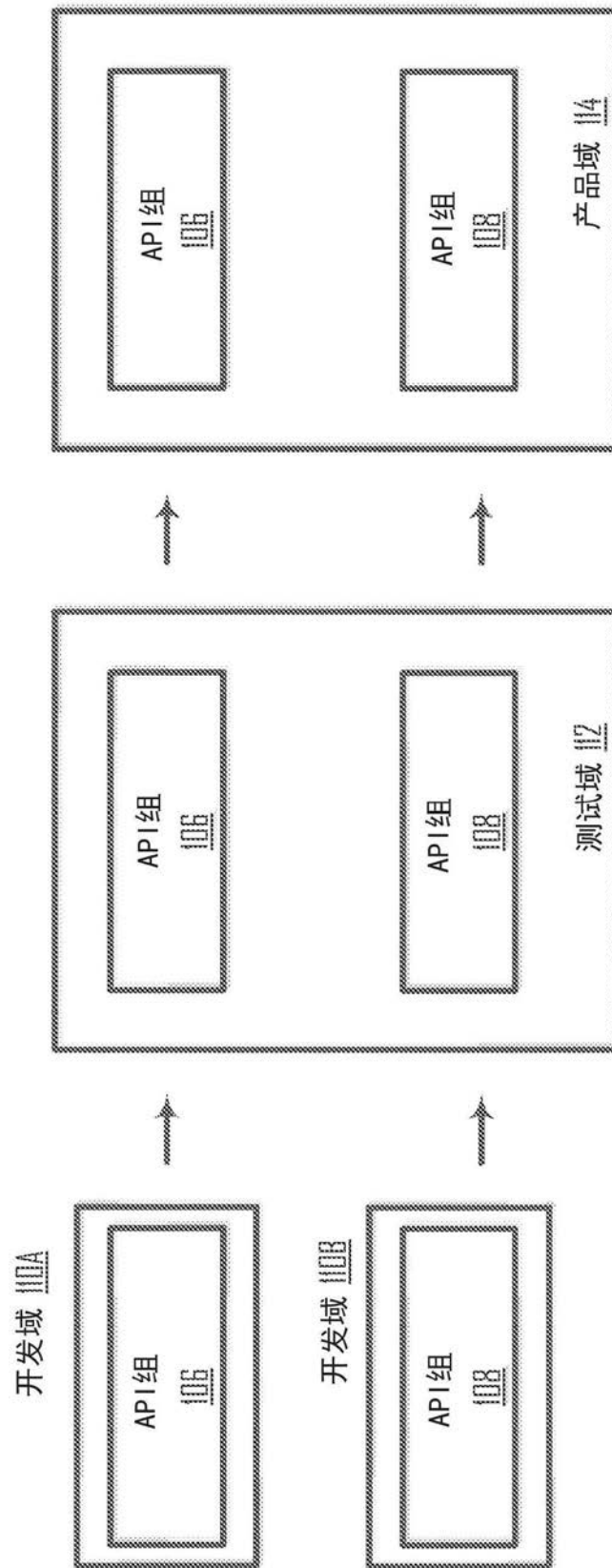


图1

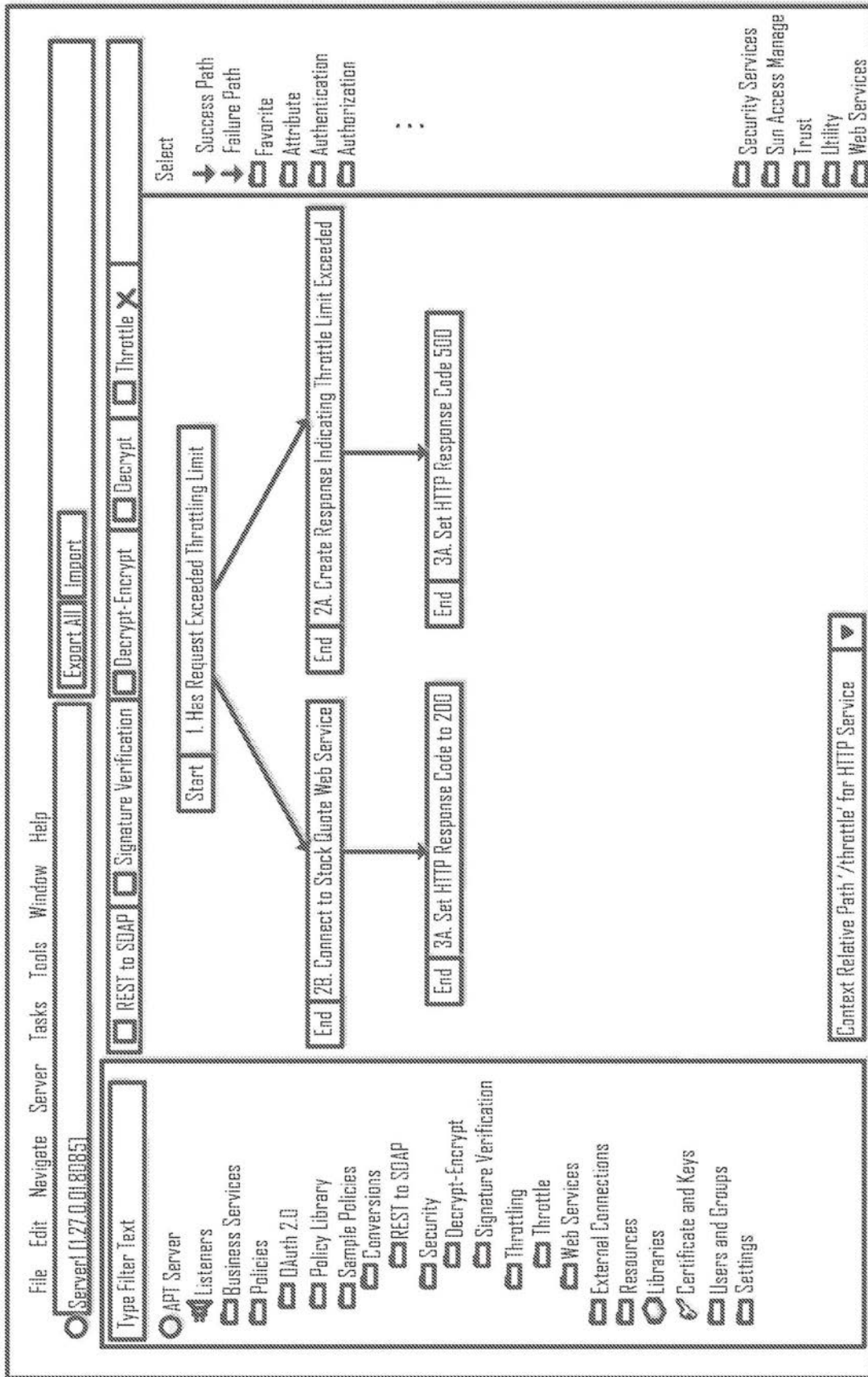


图2

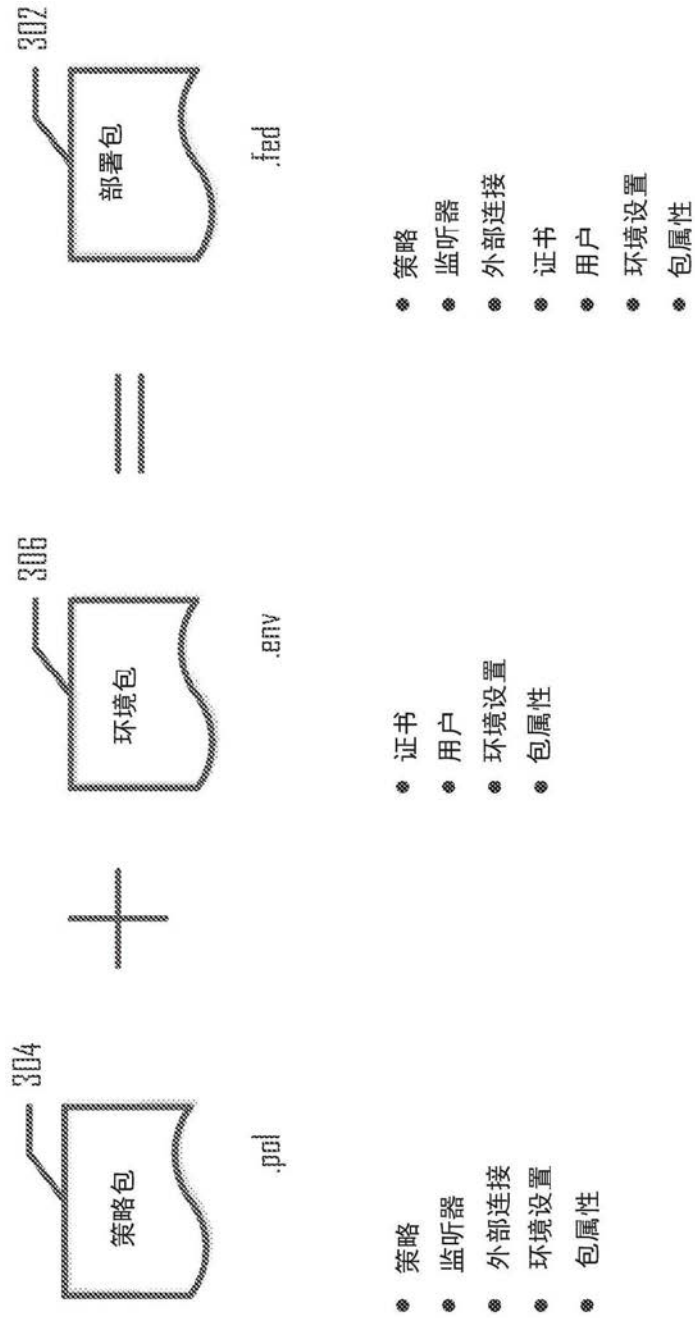


图3

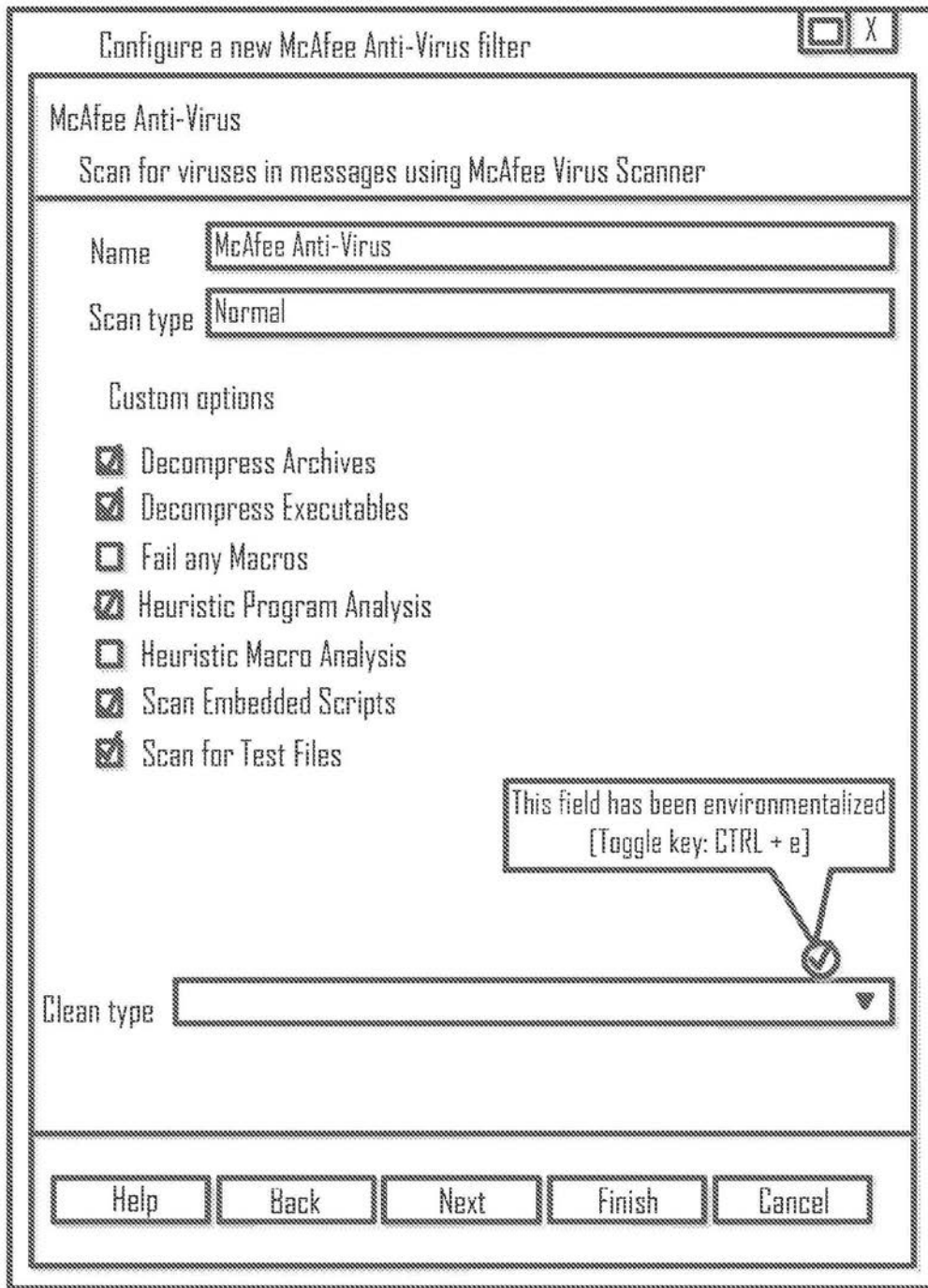


图4

500 ↘

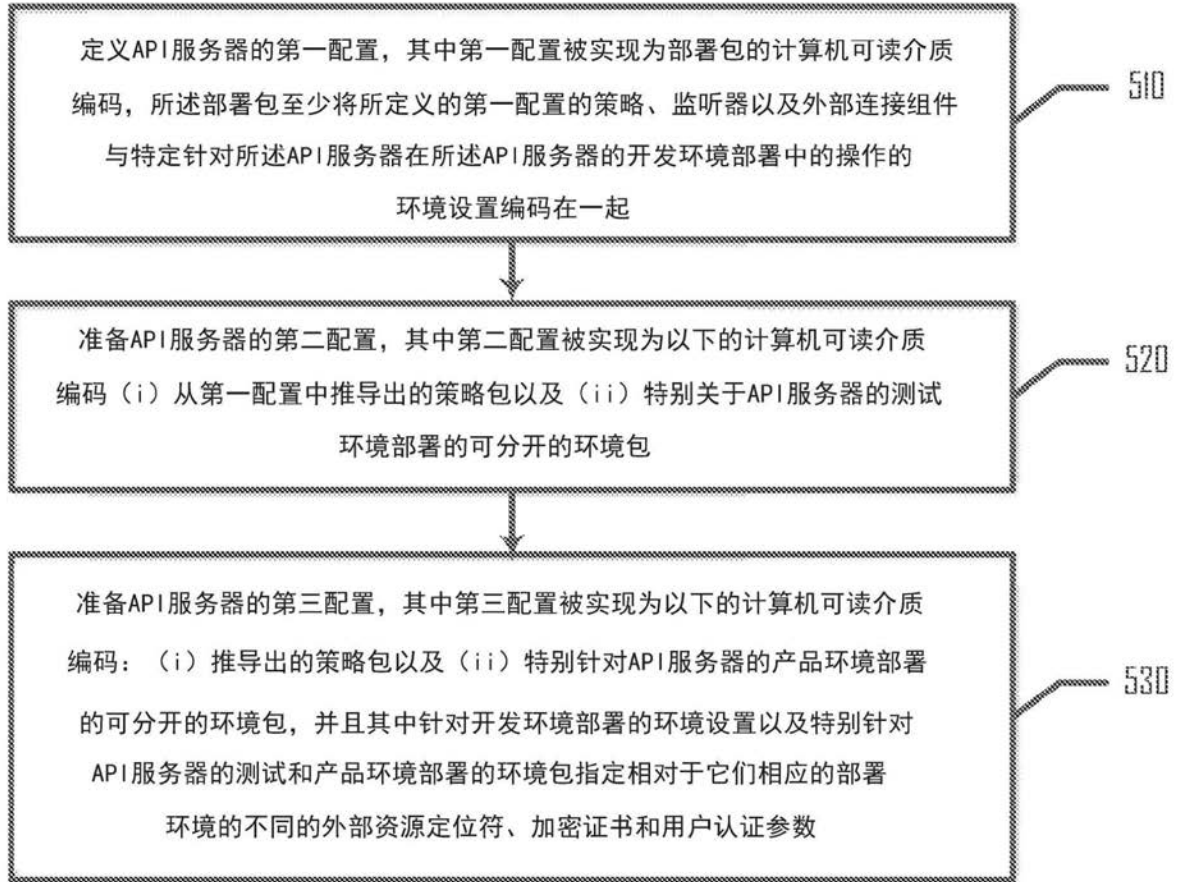


图5

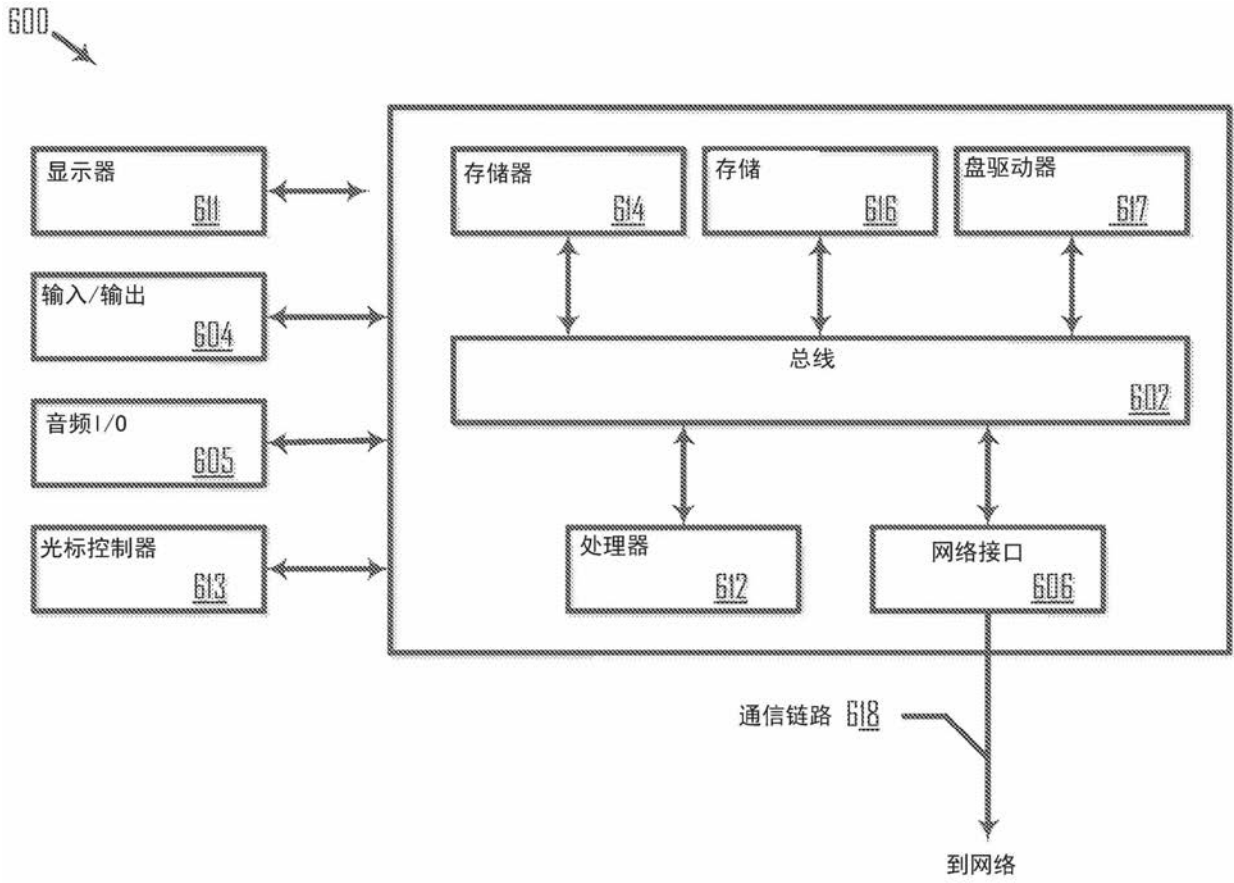


图6