

SAP 区块链

SAP 作为区块链社区的重要参与者,您是否好奇它究竟提供了哪些区块链服务?本章将从 SAP 区块链成功案例出发,然后从技术角度逐一、详细介绍 SAP 云平台上提供的区块链服务和 SAP HANA 区块链服务。希望能帮助您更好地了解如何使用 SAP 的区块链服务。

3.1 基于区块链的新商业模式

通过前两章的介绍,我们了解到区块链和分布式记账技术利用了一系列技术手段对数据进行了上链“封装”,形成了可信、共识防篡改的技术体系,从而在应用层面,为存证和确权、交易和交换、溯源、资产金融化等方面提供了技术支持。企业面对具体的业务应用挑战时,往往可以通过两种途径来解决,如图 3-1 所示。一种选择是由组织主导,借助管理和运营的中央数据库实现;另一种选择是通过区块链方法建立共享所有权,借治理和运营的共享分布式账本实现。例如,在面对药品追溯验证的需求时,欧盟存在一个中央系统可以直接将 SAP ATTP 等系统连接到“国家药品验证系统”,但是如美国,没有中央系统,就可以尝试用区块链来解决这个问题。

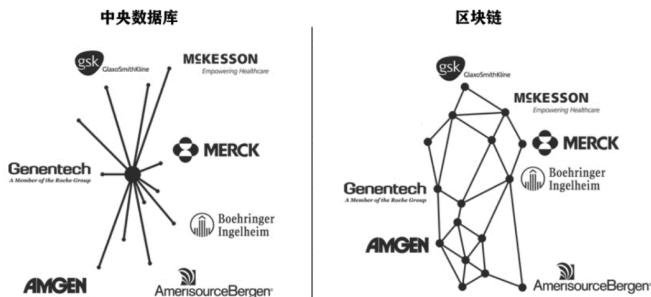


图 3-1 药品验证传统模式与区块链模式

3.1.1 SAP 区块链联盟项目

SAP 积极致力于企业级区块链场景的探索和实践,并且运用区块链业务网络对客户提供不同场景的业务支持。SAP 于 2017 年加入由 Linux 基金会领导的超级账本项目,并在中国可信区块链联盟被选举为副理事长单位,同时成立了 SAP 区块链联盟,该联盟致力于运用区块链技术探索行业特定的挑战,现在联盟里有超过 40 位客户,其中包括高科技公司英特尔、联想等。同时,SAP 区块链合作创新项目整合了 SAP 客户的行业价值链,并且有超过 100 位客户已经参与到区块链概念验证和产品验证中。与 SAP 区块链合作创新项目的部分客户如图 3-2 所示。



图 3-2 与 SAP 区块链合作创新项目的部分客户

在不同行业、不同业务线上,当企业的共享数据需要治理时,区块链就会发挥作用,为效率提升带来机遇。SAP 在溯源、供应链、公共部门信息管理、运输、金融等方向通过突出展示有效的解决方案来证明企业区块链所带来的商业价值。

区块链的应用可以分为如下两种模式。

(1) 与现有公链集成。

模式一是与现有公链集成,直接基于去中心化的区块链技术,实现价值传递和交易等应用,如数字货币。瑞波(Ripple)是一家专注于国际支付转账的实时支付结算服务商,受区块链启发的瑞波交易协议(Ripple transaction protocol, RTXP)创建于 2012 年,是一份旨在促进和发展全球最快速最廉价货币交易系统的国际协议。目前全球大多数国家的大多数银行已使用 SWIFT 系统。SWIFT 又称为“环球同业银行金融电讯协会”,是国际银

行同业间的国际合作组织,成立于 1973 年。相对于 SWIFT,在当前的国际支付场景中,Ripple 更加快捷,可以做到在几分钟内就完成支付结算。加拿大最大的金融机构 ATB Financial 已经与 SAP、Ripple 和德国 ReiseBank AG 合作,将第一笔真正的国际区块链款项从加拿大汇往德国。

但模式一场景的创新潜力有限,因为它是一个纯粹的集成主题,没有太多空间用于新的业务应用程序。预计区块链在各行业的应用,将以模式二为主,即多方协作场景。

(2) 多方协作。

模式二即是多方协作的“区块链+”模式,将传统的场景和区块链底层协议相结合,以便提高效率,降低成本。

什么是多方协作呢?指的是许多在组织单元中拥有同等权利的参与者在业务流程中一起工作,例如:

- 一组银行合作进行国际支付转移;
- 一群来自食品行业的人员合作追踪食品;
- 一群制造商共同建造一架飞机;
- 一组保险公司在担保方面进行合作;
- 非政府组织、一些国家代表和公民致力于“安全捐赠”。

3.1.2 SAP 区块链行业应用案例

SAP 作为世界最大的商业软件提供商,一直致力于探索区块链在各个行业的商业应用价值,为客户提供最佳技术解决方案。目前,SAP 已与政府、食品供应链、运输、制药等领域的 60 多家企业进行合作,共同探索区块链技术的应用。下面介绍 SAP 是如何利用区块链技术解决各行业中多方协作的痛点的。

1. 公共事务

(1) 行业现状和痛点。

通常政府职能部门工作人员每天需要处理大量的文档,审核、盖章然后等待申请者领取纸质文档。申请者多数情况下还需要将类似或者完全一样的信息提交到其他的政府部门,进而完成所有审批流程。不难看出在当前情况下,政府机构之间信息共享还没能完全实现,申请者不得不去不同的机构完成类似甚至相同的事情,与此同时,政府工作人员也需要花费较长的时间反复验证申请者的信息。设想如果申请者信息能够被有效、统一、安全地管理,这将极大缩短审核流程,申请者再无须去不同的机构反复证明自己的身份信息,这将从根本上颠覆目前的工作流程。而区块链技术有很大的潜力解决此类复杂流程问题。

(2) 案例阐述。

在意大利, Agenzia per l'Italia Digitale (AGID) 定义了所有办事机构必须遵循一定的公民办事流程。博尔扎诺 (Bolzano) 是意大利东北部的城市, 是重要的工商业中心。博尔扎诺的基础建设办公室遵循着 AGID 定义的协议来管理公民办事流程。在博尔扎诺术语中, 他们使用所谓的“分册管理业务”, 分册是一种容器, 其中业务流程和附加文档通过数字化方式保存。数字化存储的分册为区块链的实施提供了可能性。博尔扎诺有 21 个办事处, 目前主要使用 IDP (Identity Provider) 系统作为用户身份验证, 所有的办事处都连接到 IDP 系统。

(3) SAP 解决方案。

SAP 区块链团队携手当地机构就建设单位的补助金问题一同创建了区块链的解决方案。该方案力图创建新的组织模型和办事流程, 公务员可以通过移动应用端, 远程轻松管理申请者物理文件的转移。

在概念验证阶段, SAP 团队在 SAP Cloud Platform (SAP CP) 一个账户中创建了 3 个相互连接的 Multichain 节点和 3 个 Java 应用程序。每个 Java 应用程序都连接到其中一个区块链节点。然后为每一个办事处创建一个区块链账户, 这些账户用于创建区块链交易。接下来通过一个简单的 REST 服务为每个办事处账户分配名称, 这些名称显示在用户界面中, 如图 3-3 所示, 例如创建者: Office for Construction。

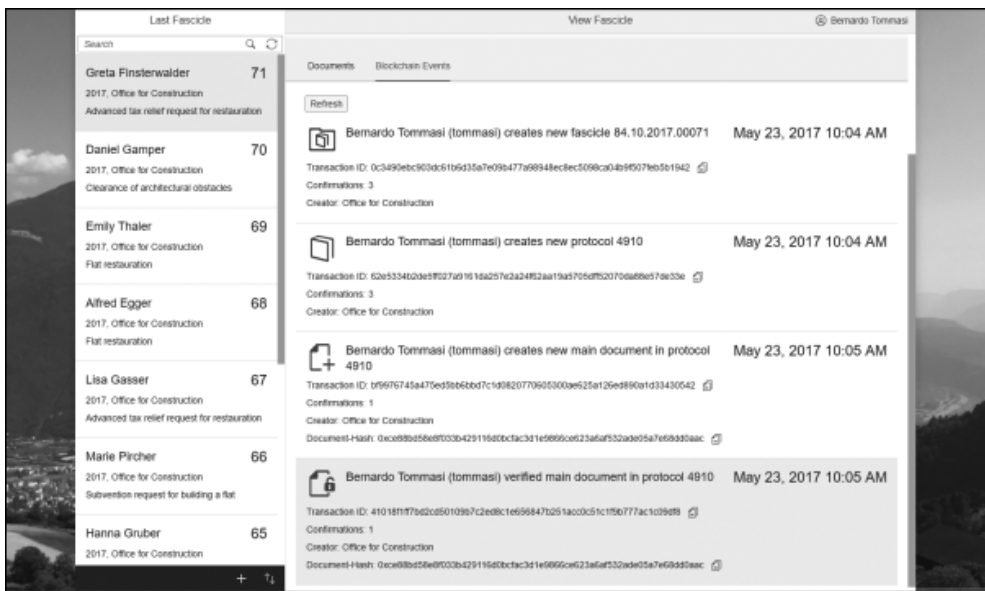


图 3-3 显示办事处账户名称的用户界面

SAP 团队通过区块链存储所谓的分册的所有交易。区块链记录了交易的时间戳,保证了信息的不可篡改性。我们将分册 id 作为键值,数据作为流值存储,如图 3-4 所示。



图 3-4 存储在区块链的原始数据格式

通过该方案,新分册的创建、物理文档的扫描、物理和数字文档的验证以及员工的数字签名全部合并为一步,如图 3-5 所示。整合好的分册信息可以方便地通过 multichain 实现信息共享。

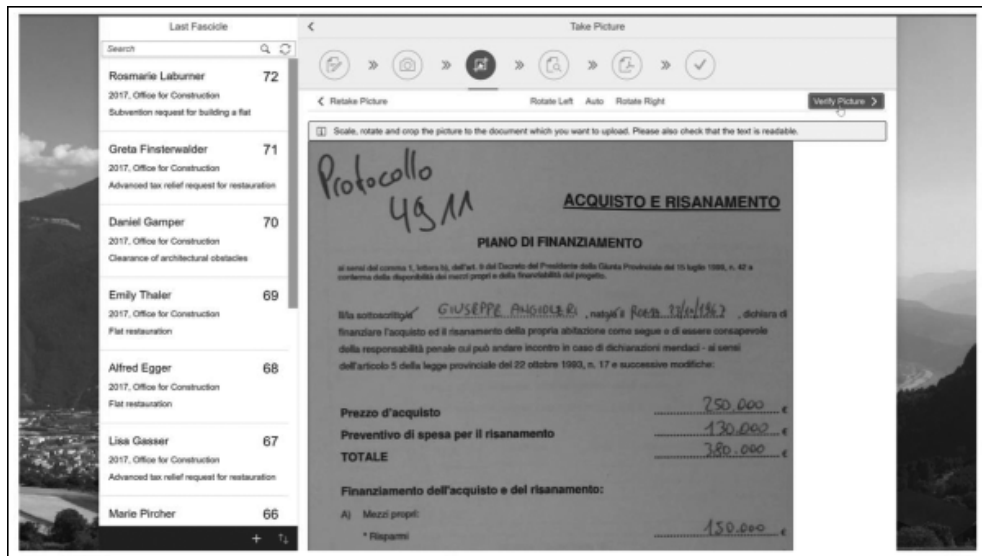


图 3-5 存储在区块链的扫描件

(4) 商业价值。

该方案成功帮助博尔扎诺市政机构重新定义了办事流程,整合了原来相对零散、不统一的分册处理机制。这极大程度上去除原先重复性的工作,申请人无须反复去不同的市政机构递交材料,公务员也可以方便地在移动端处理文档,同时在区块链保证的前提下有效共享给其他机构。更重要的是,该方案也推开了区块链在政府机构应用的大门,证明了区块链的应用价值,相信区块链还会在相同领域扮演更加重要的角色。

2. 食品供应链

(1) 行业现状与痛点。

食品供应链有着极其复杂的流程,参与方直接或间接地影响着食品的生产和运输及最终品质。在食品的生产、流通、销售过程中,各参与方分别记录产品信息,并且只在必要的时候跟有商业利益关系的合作伙伴共享这些信息。随着食品在供应链中的流转,完整的溯源信息并没有被记录下来。这就导致了以下问题:首先,缺乏时效数据降低了供应链参与方对市场预判的准确度;其次,消费者只能通过最后环节,即零售商贴的标签了解产品,无法真实完整地获知食品的产地和质量信息;最后,在需要对产品进行召回时,由于信息的缺失,供应商不得不将各方数据进行汇总比较,这无疑是低效的。

(2) 案例阐述。

① 从藤枝到餐桌——蓝莓的验证之旅。

Naturipe 是著名的浆果品牌,其历史可以追溯到 1917 年。旗下产品包括蓝莓、黑莓、树莓、草莓和蔓越莓等。Naturipe 在全球拥有 700 家中小型农场,其中蓝莓种植历史悠久,很多都是多代传承种植,种植地区包括智利、美国、阿根廷和墨西哥等,南北美洲两地恰好形成季节互补,实现全年无间断供应。

为了追溯鲜果产品,统一产品品质,加快鲜果出口,针对食品供应链行业面临的种种挑战,Naturipe 与 SAP 合作,通过实施区块链技术,建立水果溯源体系,记录蓝莓从种植到零售的生命周期。

农场收集和分享农作物的生产信息,例如产地、是否有机、生长周期等,数字化作物生长周期如图 3-6 所示。

分销商将货运信息、发送者、接受者和批次信息等上链,如图 3-7 所示。

在最后环节,即零售商收到产品之后,将自己的信息也记录在区块链上,如图 3-8 所示。

当发生产品召回时,分销商和零售商可以快速找到被影响到的商品批次,及时从市场撤回问题产品,并通过记录在区块链中的信息来甄别问题源头。同时,客户也可以通过手机 App 等终端了解产品的生命周期。



图 3-6 数字化作物生长周期



图 3-7 采集,物流信息上链



图 3-8 验证蓝莓之旅



② 从海洋到餐桌——讲述每一条金枪鱼的故事。

总部位于圣地亚哥的 Bumble Bee Foods(大黄蜂食品)是北美最大的海鲜罐头公司,提供全系列罐装和袋装金枪鱼、鲑鱼、沙丁鱼和特种蛋白产品。Bumble Bee Foods 长期以来一直是海鲜产品追踪的行业领导者,为了进一步提高产品供应链对消费者和客户的透明度,履行对国际海产品可持续性基金会的承诺,确保原料是公平采购的新鲜金枪鱼。Bumble Bee Foods 和 SAP 合作,继续创新和改进海鲜的可追溯性。

Bumble Bee Foods 正在使用 SAP 云平台区块链服务追踪黄鳍金枪鱼从印度尼西亚洋到餐桌的旅程。

借助 SAP 的区块链技术,通过将金枪鱼捕捞、运输、检验、产品包装等环节信息上链。消费者和客户可以通过使用智能手机扫描产品包装上的二维码,轻松访问 Anova[®] 黄鳍金枪鱼罐头的完整来源和历史。区块链技术通过代码的快速转换,提供了有关鱼市之旅的即时信息,包括捕获物的大小、捕获点和捕获它的社区,以及验证真实性、新鲜性、安全性等。

区块链技术允许 Bumble Bee Foods 等公司存储数据并创建防篡改的供应链历史记录,每个参与者都可以共享和查看这些历史记录。对于 Bumble Bee Foods 来说,区块链是在各方之间共享数据的最安全的方式,因为它是不可破坏和可验证的。Bumble Bee Foods 是第一家将 SAP 区块链技术成功纳入生产的食物公司。

(3) 商业价值。

据世界卫生组织统计,每年全球有 40 多万人死于受污染的食物,全球每年有约 10% 的人口因为食品安全问题罹患疾病,通过区块链提升食品供应链的效率,提供对于食品数据的存储和追溯能力,可以准确及时地发现和控制在食源性疾病的传播蔓延。同时,食品企业通过区块链技术实现创新的同时,支持消费者溯源并加强他们对品牌的信心。

3. 运输行业

(1) 行业现状与痛点。

在当前的国际航运管理中,有许多缔约方扮演着不同的角色。如卖方(托运人)和买方(收货人),它们的银行(出口、进口、代理银行)、货运代理、承运人和政府部门等(海关、税务、安全、港口等)。这些缔约方都使用着各自独立的数字系统。其中一些系统之间通过 B2B(Business to Business)接口进行数据交换,但是由于没有单一的真实来源。为了“证明事实真相”,仍需通过快递服务(DHL 快递、联邦快递等)交换并装运带有印章和签名的纸质原始表格。文件的处理和运输占据了物流总成本很大一部分。

提货单是一种纸质单据,由海运公司签发。在国际贸易中,它被用作货物的确权证据,经承运人确认已收到货物,并在目的地将货物交给单据持有人。这是货物所有权的证明。提货单原件持有人是货物的所有人,是唯一有权从卸货港提货的人。当船在途中(如商品交易)时,货物所有权可能会发生变化,由于货物无法移交,提货单却是可以移交的。

所以各缔约方都需要查看提货单原件,以确认或批准贸易、出口、进口、融资等。

这个过程主要存在以下痛点。

① 多方处理的数据或文档没有单一的真实来源→各方系统之间 B2B 接口的开发成本高昂。

② 因纸质文件的运输导致的时间成本,在文件运输过程中如果集装箱无法被按时提取,在卸货港产生的滞留成本。

③ 去数字化和再数字化:将数字化单据打印成纸质单据→寄送纸质单据→将纸质单据扫描成 PDF 文件。

④ 基于电子邮件的审查周期、审批流程冗长。

⑤ E2E 流程中相关文件真实性无法保证。

⑥ 遭遇假冒提货单/欺诈,导致货物被盗。

⑦ 纸质提货单丢失风险。

(2) 案例阐述。

通过区块链技术,SAP 引入了“电子提货单”,不同业务伙伴间的货品传递实现了真正的安全、互信、无纸化。当货物到港的时候,电子提货单的所有者能直接从货运商变更为下一阶段的所有者。

(3) 技术架构。

其技术架构如图 3-9 所示。

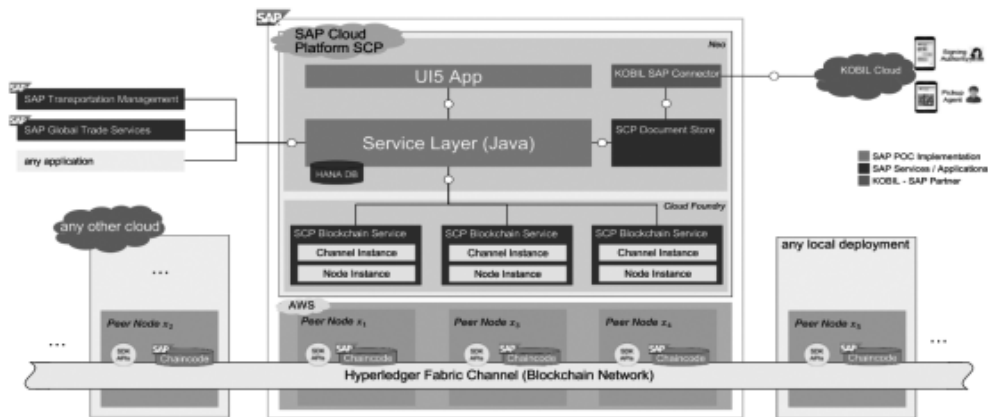


图 3-9 技术架构

(4) 商业价值。

将航运各方聚集在区块链网络中,可以使每一方以不同的方式和程度受益。在信息

数字化的过程中提高效率与透明度,从而节省实际成本,例如纸张运输成本,或在卸货港的滞期费/滞留费。

由于区块链技术非常昂贵,产生的价值和成本节约应以一种有待商定的方式共享。在一个完美的世界场景中,世界上的公司组成一个联盟(联合),以便拥有一个管理委员会来决定成员资格、费用、规则、技术、合作伙伴等事宜。

未来,这些各方将实施接口或所谓的后端适配器,以便将记录系统与区块链网络连接起来。最终用户将无法识别数据是否来自区块链等。SAP 将需要为相关后端提供区块链适配器,以提供获得区块链价值的先决条件。

产生的价值如下。

- ① 提高最终用户的效率。
- ② 无须人工处理纸质文件(接收、分类、扫描、发送等)。
- ③ 不需要重复输入数据。
- ④ 提高信息透明度/可见性。
- ⑤ 确保 E2E 流程中所有文件版本的单一真实来源。
- ⑥ 与各方系统之间 B2B 接口的开发成本相比,区块链只需要一个与区块链系统集成的接口——供应链可见性。例如,3 个合作伙伴可能使用不同的系统追踪事件,则对于同一事件,可能有 3 种不同的结果。

4. 制药行业

1) 行业现状与痛点

为了提高药品利用率,在美国每年有价值近 70 亿美元的退回药品被转售,然而由于全球范围内假冒药品存在严重问题(据路透社称,全球假药市场价值每年可能高达 2000 亿美元),如何对退回药品进行鉴别是一个不容忽视的问题。

为了在提高药品利用率的同时防止假冒药品被退回,《美国药品供应链安全法》(DSCSA)增加了一系列条例来保证药品行业的供应链安全。

(1) 产品序列化:从 2017 年 11 月 27 日开始,药品制造商应在每个包装上贴上或印上产品标识;

(2) 退回药品验证:从 2019 年 11 月 27 日开始,药品制造商收到退回的产品后,在打算进一步分销该产品之前,必须对产品进行验证。验证内容包括每个包装上的标准数字标识符、批号和有效期等。

在欧洲也有类似的法律,但欧盟采用了集中数据库,制造商将数据上传到监管机构的集中数据库。在美国,由于数据缺乏集中性,将需要在所有不同的信息化系统之间进行大量的集成。

2) 案例阐述

为了降低系统间集成的复杂性,打造以行业为联盟的可信任共享信息平台,SAP 与美国医药行业一起,利用区块链概念和技术来搭建确保数据安全和完整性的 SAP 生命科学信息协作中心(information collaboration hub for life sciences)。将药品供应链的参与方纳入联盟链中,实现对药品的溯源。在药品生产过程中,各药品制造商必须将产品代码、批次、到期日期和序列号等信息上链。以便供应链下游以及客户可以轻松验证药品真伪,实现药品供应链上下游对共享数据的治理,如图 3-10 所示。

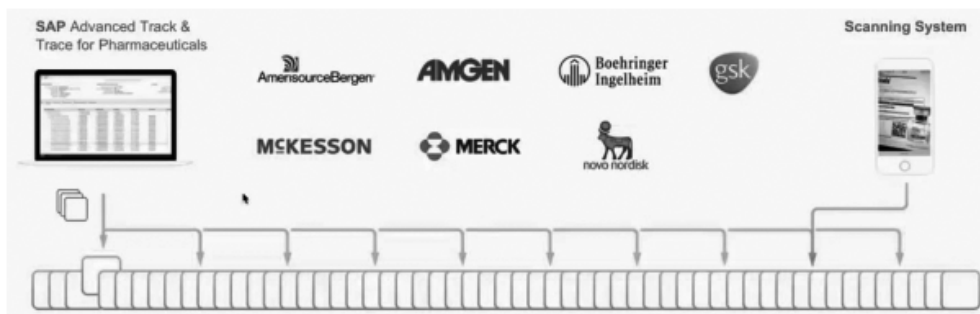


图 3-10 使用区块链实现药品供应链上下游对共享数据的治理

结合移动互联网,该区块链解决方案支持行业对共享数据不可篡改的需求,避免了许多复杂的集成,为药品制造商和批发商提供了可扩展和安全的数据平台,以满足即将到来的验证监管要求,如图 3-11 所示。

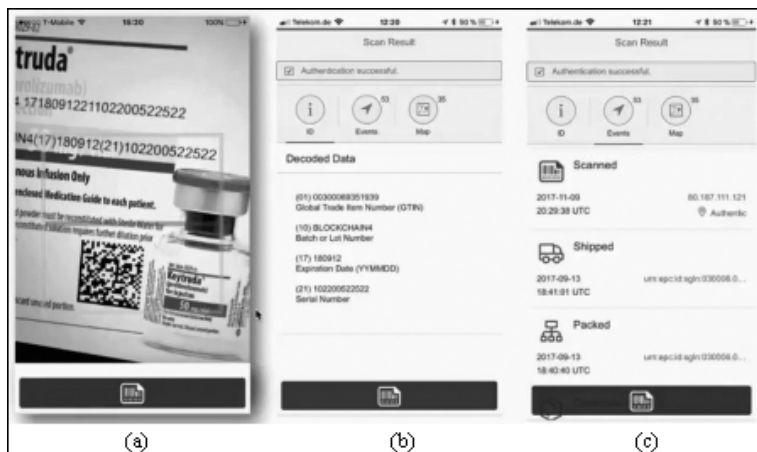


图 3-11 药品验证

3) 商业价值

该方案帮助美国制药行业建立了可信任的药物共享信息平台,实现了低成本的药物溯源与鉴真。在提高药品利用率的同时,杜绝了假药流入市场,保障了消费者的生命安全与医药行业的可持续发展。

在了解 SAP 区块链各行业的应用后,本书接下来会深入阐述 SAP 云平台区块链服务。

3.2 SAP 云平台区块链服务

SAP 云平台在 SAP 倡导的数字化平台概念中起了重要作用。SAP 云平台是一种平台,即服务(PaaS)产品,可为云应用程序提供开发和运行环境。SAP 云平台基于 SAP HANA 内存数据库技术,使用开源和开放标准,允许独立软件供应商、初创企业和开发人员创建基于 HANA 的云应用程序。SAP 致力于为客户打造一个业务技术平台,客户只需专注于业务,无须关注底层技术模块搭建,如图 3-12 所示。

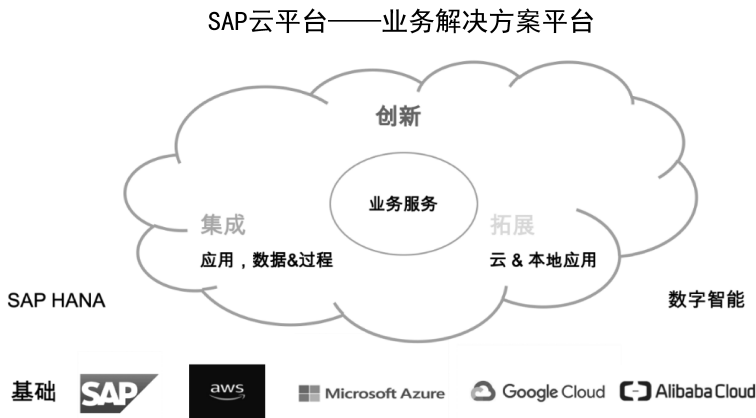


图 3-12 SAP 云平台概述

SAP 云平台通过使用前沿技术,如区块链、机器学习和高级分析功能帮助客户扩展本地或基于云的 ERP 应用程序;建立和部署新的企业业务云和移动应用程序;集成和连接不同数据源的企业应用程序;将企业应用程序和数据连接到物联网。

SAP 云平台促进了 SAP S/4HANA 财务与 SAP Ariba 采购和 SAP SuccessFactors 人力资源等云应用程序的集成。不仅如此,它还可以将这些应用程序与非 SAP 系统的数据源集成在一起,包括社交媒体站点和其他供应商的企业应用程序。总结来说,SAP 云平台能够帮助企业数字化转型得更敏捷、更高效、更彻底。

通过 SAP 云平台,可以很方便地扩展现有的本地和云解决方案,最大化利用之前的投资。既可以在 SAP 云平台上部署自己的应用,也可以借助 SAP 应用中心现有的应用快速开展自己的业务。截至 2019 年,SAP 已经在 SAP 云平台上部署了 50+ 的企业级应用,SAP 应用中心也有超过 1700+ 合作伙伴提供的现成应用可以直接被下载使用。SAP 云平台作为 SAP 客户数量增长最快的产品,已经有超过 14000 个客户。这里也有 1600+ 经验丰富的合作伙伴随时待命,支持您对 SAP 云平台的使用,如图 3-13 所示。

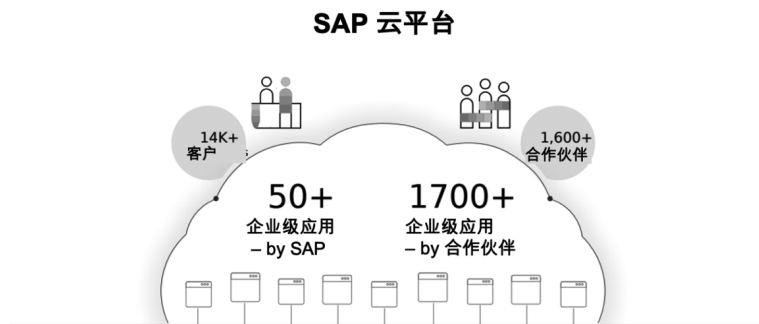


图 3-13 SAP 云平台现状

当打开 SAP 云平台 Cockpit,在区域标签页内能发现 SAP 云平台提供两套运行环境,多环境和 Neo 环境。多环境的基础设施由公有云基础设施提供商提供,SAP 仅负责维护平台层。而 Neo 环境的底层基础设施是由 SAP 自己提供。多环境的运行环境包含了 Cloud Foundry application runtime,这是基于 Cloud Foundry 基金会的开源应用平台构建而成。如果需要开发基于微服务架构的应用,SAP 推荐选择多环境运行环境。因为在多环境下能使用多种编程模型,例如 Java、nodejs、SAP HANA extended application services 和 advanced model (SAP HANA XSA)。Neo 环境则更偏向于 UI5、Java 和 HANA Extended Service 开发。

SAP 云平台数据中心大多位于美国和欧洲。2019 年底,SAP 云平台成功落地阿里云。这对于中国的 SAP 客户、合作伙伴都是振奋人心的消息,今后可以更方便地在 SAP 云平台上开展业务。

通过 SAP 云平台可以尝试将最新技术集成到 SAP 传统产品中。SAP 云平台提供了 100+ 个服务,如区块链、机器学习、物联网服务等。就区块链服务而言,SAP 已经集成了 Hyperledger Fabric、Multichain 以及 Quorum 开源区块链技术并且提供区块链应用启用服务,如图 3-14 所示。

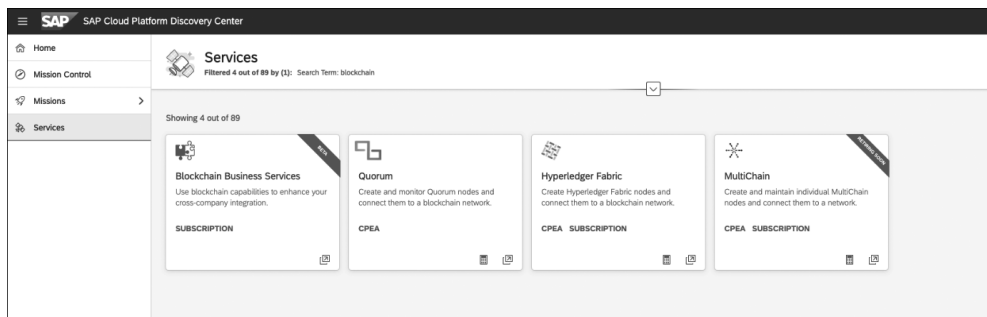


图 3-14 SAP 区块链服务

想要实现一个区块链系统,除了参照开发者文档自己开发之外,也可以使用提供区块链服务的云平台直接搭建,这可以帮助我们在项目初期节约经费和时间。SAP 云平台提供了两种主流的区块链服务:Hyperledger Fabric 和 Multichain。

3.2.1 Hyperledger Fabric 区块链平台服务

Hyperledger Fabric 是一种开源的企业级许可分布式分类账技术平台,专为在企业环境中使用而设计。它是一个基于模块化架构,提供分布式账本解决方案的平台,是 Linux 基金会在 2015 年创立的超级账本(Hyperledger)项目下的区块链项目之一。Hyperledger Fabric 支持智能合约,利用容器技术装载一些链码(chaincode)实现业务逻辑。和其他区块链不同的是,它是一个私有的、有许可的区块链系统。在有许可的区块链系统中,数据的隐私性和安全性可以得到保护,也可以实现网络成员管理。

1. 技术架构

1) 分布式账本

在第 1 章中我们已经了解到,分布式账本是区块链网络的核心,它记录了网络上发生的所有交易,网络中每个成员都有一份账本。在 Hyperledger Fabric 区块链网络中,这个账本系统有两个组成部分:世界状态(world state)和区块链(blockchain)。

世界状态描述的是某一时刻的账本状态,一般是以键值对组合的形式保存在世界状态数据库中,并且可以对其进行增、删、改、查的操作。世界状态中记录了当前每个业务对象(如车、发票、个人资产等)的真实属性以及版本号信息。应用程序对世界状态的改变会体现在这些对象的属性值和版本号上。

当 Hyperledger Fabric 网络节点接收到交易请求信号之后,会先根据共识机制验证该请求,随后就会被网络成员节点打包并以区块链的形式存储在数据库中。区块链是由多个区块利用哈希算法生成的区块头前后相连构成的,而区块则是由多条交易记录组

成的。

2) 成员节点

Hyperledger Fabric 网络是由各种各样的节点构成的,每个节点都有独特的功能和角色。

成员节点是区块链参与者的基本节点,一共有两种。最常见的成员节点是提交节点,每个节点都可以是提交节点,其主要功能就是维护区块链账本和提交交易请求。还有一种成员节点叫作背书节点,是负责执行智能合约并验证交易请求合法性的节点,同时也负责对交易请求进行签名背书。由此可以看出 Hyperledger Fabric 区块链的许可链属性,即并不是所有满足基本交易条件的交易请求都会改变账本状态,而是需要根据背书策略经过背书节点的许可才能继续。

3) 共识

除了上面两种成员节点之外,网络中还有一种必不可少的排序节点。排序节点就是用来对交易记录进行排序的节点,进而防止出现“双花”问题。为了对交易的合理合法性进行验证,必须将它们按照发生的顺序记录在账本上,即使它们可能是网络中不同的参与者生成的。保持账本中交易记录在区块链网络中同步的过程称为共识。共识机制确保了账本仅在交易获得适当的参与者批准后才会被更新,并且在账本被更新时以相同的顺序记录下相同的交易。

比特币区块链是通过“挖矿”(即工作量证明共识算法)来验证交易的,但这种算法会消耗巨大的计算资源,而且每个网络成员节点都可以参与共识过程,以及对交易的排序和打包过程。当节点由于某些人为或者不确定的原因对交易的顺序产生歧义时,还是有可能出现账本分叉的情况,即网络节点的账本区块链存在多个版本。

Hyperledger Fabric 则提供了专门负责交易排序的节点(orderer)和可配置的背书策略。网络中的多个排序节点提供交易排序服务,然后再发送给背书节点和提交节点。这套系统使得网络管理者可以自行选择一种最能代表参与者之间关系的共识机制。排序节点上可以使用的共识机制包括 Raft(官方推荐,v1.4.1 加入)、SOLO(v2.0 已弃用)和 Kafka(v2.0 已弃用)。

4) 通道

在 Hyperledger Fabric 这样的分布式系统中通常会有多个组织共同参与,即使操作的业务对象相同,在与不同的网络成员交易时的业务逻辑也不尽相同,因此 Hyperledger Fabric 提供了通道(channels)的功能,可以让一个组织参与到不同的区块链网络中。这个功能对于在网络上存在竞争关系的成员非常重要,它们可能并不想要将自己的一部分敏感信息(如价格)告诉竞争对手。

每个通道都可以拥有一份完全独立的账本,并且在创建通道之后,只有通道内的成员

才可以保存所在通道中的账本。图 3-15 是一个简单的多通道网络案例,图中三个组织各有一组成员节点,每个节点上都记录着与组织业务相关的链码和账单,并且组织 B 可以同时两个通道中处理不同的业务。

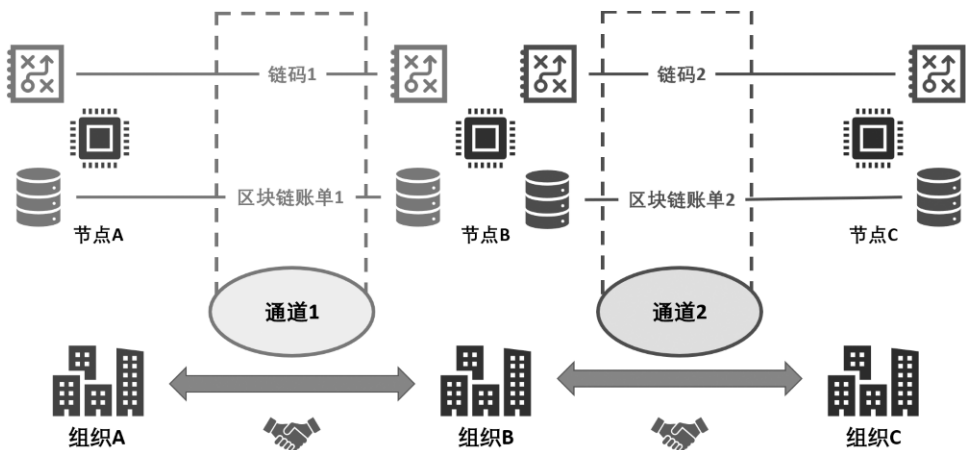


图 3-15 Hyperledger Fabric 分布式账本

5) 链码

智能合约是用来定义和管理区块链账本所记录的资产的。在 Hyperledger Fabric 中,智能合约就是以链码的形式来开发和管理的。除此之外,链码可以用来表示 Hyperledger Fabric 的底层代码,但实际上只有区块链网络的管理者才需要从链码角度来进行维护管理和部署,其他区块链成员和应用程序开发者都可以直接从智能合约角度来理解链码。

2. 使用演示

下面主要介绍在 SAP 云平台上 Hyperledger Fabric 的使用。截至目前,SAP 云平台上所支持的 Hyperledger Fabric 服务版本为 v1.4.4,预期将于 2021 年 3 月 21 日从 SAP 云平台下线,届时将不再支持新的订购和续订。更多详情请咨询官方账号团队。

在 SAP 云平台上获取区块链服务之前,首先需要确保完成以下准备工作。

- (1) 开启全局账户并可以使用 Cloud Foundry 环境。
- (2) 购买 Hyperledger Fabric 服务以及各种服务计划的使用额度。
- (3) 将所需的额度分配到子账户。
- (4) 创建至少两个空间用于部署 Hyperledger Fabric 服务。

具体操作步骤介绍如下。



1) 创建服务实例。

在空间面板中,选择 Service Marketplace,并在右侧的服务列表中选择 Hyperledger Fabric 服务,如图 3-16 所示。

单击进入服务详情页面,并在“创建实例”窗口创建新实例,如图 3-17 所示。

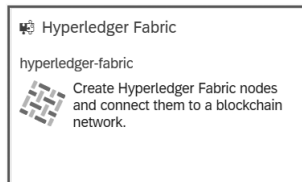


图 3-16 Hyperledger Fabric 服务

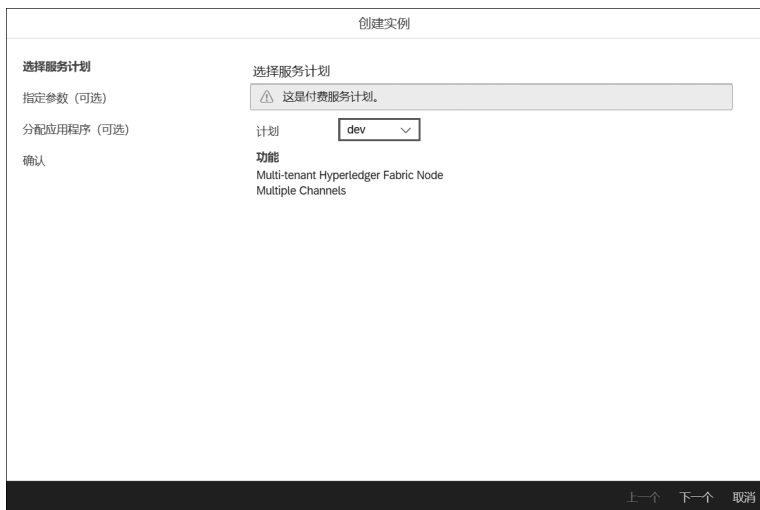


图 3-17 创建 Hyperledger Fabric 实例

新建实例时需要选择服务计划,基本的服务计划有 dev 和 channel。创建 dev 实例会建立一个 Hyperledger Fabric 基本的节点网络,包括一个 Orderer 节点,4 个 Peer 节点和账本数据库;channel 实例可以管理和监控 channel 内的交易记录,API 调用记录和节点状态,还可以管理链码。

由此可见,不同的服务计划对区块链的控制程度是不同的,但因为一个空间下的所有成员可以管理空间内的实例,所以需要将 dev 实例和 channel 实例分别创建在两个不同的空间下,方便对区块链进行角色分配和权限控制。

这里我们先创建一个 dev 实例,在完成下一个步骤之后再创建 channel 实例。

2) 建立实例间的联系

dev 实例和 channel 实例不能独立产生效果,它们之间必须建立联系。在 Hyperledger Fabric 中是通过 Service Key 来连接 channel 实例和 dev 实例的。Service Key 中可以维护 channel 实例对 dev 实例的操作权限。

在 Hyperledger Fabric 中要创建一个 Service Key,需要先在 dev 实例中创建其所属的 channel,然后再为这个 channel 创建 Service Key,如图 3-18 所示。注意,在上一步骤中“dev 实例”里创建的 channel 和选择服务计划时创建的“channel 实例”并不是同一个概念。

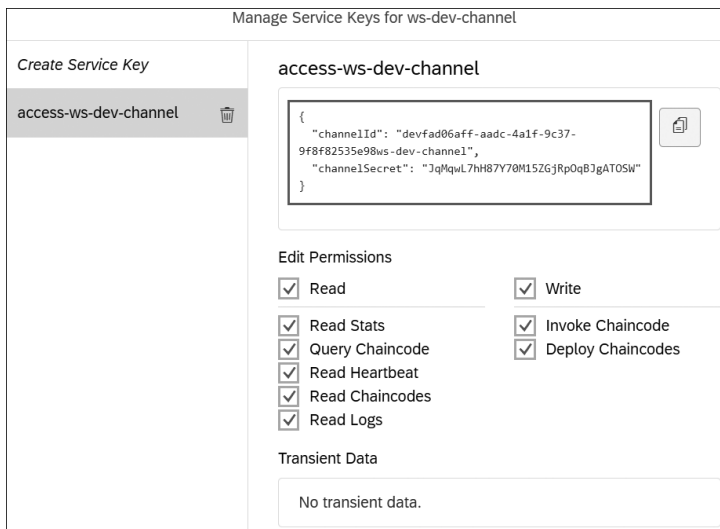


图 3-18 创建 Service Key

创建完 Service Key 之后,就可以去第二个空间,按照之前的步骤创建一个 channel 实例,同时指定这个 channel 实例的参数,即其 Service Key。至此,我们已经成功连接了两个实例,并可以通过 channel 实例看到 dev 实例网络的状态。

3) 链码的开发

在 SAP 云平台中,一份可部署的链码包结构如图 3-19 所示。

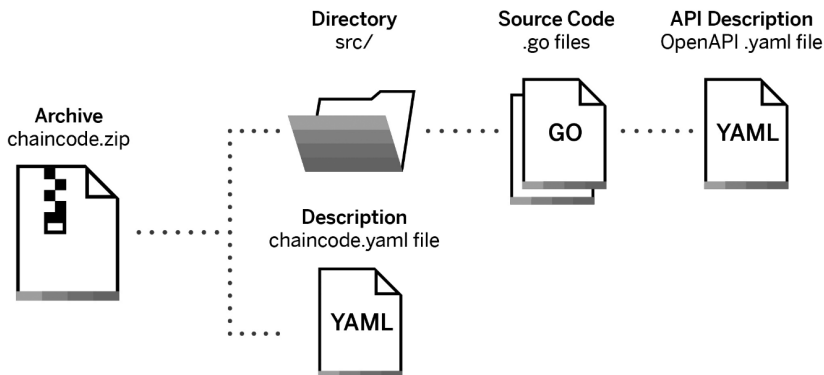


图 3-19 链码包结构图

接下来用一个简化版的 Hello World 链码来了解链码的具体内容。

首先定义一个包含了两个简单函数(read 和 write)的 go 语言程序文件 hello_world.go 作为智能合约。此外,链码中还必须完成 Init 和 Invoke 函数。Init 函数是在 channel 上初次部署链码时运行的初始化代码。Invoke 函数是每次调用智能合约的必经之路,其参数就是所需调用的函数名。

```
type HelloWorld struct {
}

func main() {
    shim.Start(new(HelloWorld))
}

func (cc *HelloWorld) Init(stub shim.ChaincodeStubInterface) peer.Response {
    return shim.Success(nil)
}

func (cc *HelloWorld) Invoke(stub shim.ChaincodeStubInterface) peer.Response {
    function, args := stub.GetFunctionAndParameters()
    switch function {
        case "read":    return read(stub, args)
        case "write":   return write(stub, args)
        default:        return shim.Error("Valid functions are 'read|write'!")
    }
}

func read(stub shim.ChaincodeStubInterface, args []string) peer.Response {
    if value, err := stub.GetState(args[0]); err == nil && value != nil {
        return shim.Success(value)
    }
    return shim.Error("Not Found")
}

func write(stub shim.ChaincodeStubInterface, args []string) peer.Response {
    if err := stub.PutState(args[0], []byte(args[1])); err != nil {
        return shim.Error(err.Error())
    }
    return shim.Success(nil)
}
```

然后定义一份 OpenAPI 接口描述文档 hello_world.yaml,包含智能合约的函数名称和输入参数。

```
paths:
  /{id}:
    get:
```

```
operationId: read
parameters:
  -name: id
    in: path
    type: string
    required: true
responses:
  200:
    description: OK
  500:
    description: Failed
post:
  operationId: write
  parameters:
    -name: id
      in: path
      type: string
      required: true
    -name: text
      in: formData
      type: string
  responses:
    200:
      description: OK
    500:
      description: Failed
```

最后创建整个链码的描述文件 chaincode.yaml,用来管理链码生命周期。

```
Id:      com-sap-icn-blockchain-example-helloWorld
Version: 13
```

4) 部署和测试链码

接下来就可以在 channel 实例面板中的 Instantiate Chaincode 页面中安装并部署链码包,如图 3-20 所示。

部署链码的同时可以设置 Endorsement Policy 背书策略,用来控制发生在链上的交易需要经过哪些节点的认证才可以被记录在账本中。图 3-20 中的 Any 指任意一个所选组织(organization)中的节点认证过后即可记录, Majority 指需要超过 50%的节点认证之后才可以记录,而 All 指必须经过全部节点的认证。此外,还可以使用 Hyperledger Fabric 提供的简单的代码实现更复杂的背书策略。使用 n 和 out_of 关键词规定交易被记