迈向 Future Fish Farm 的创建

Pavlos Papadopoulos*、William J Buchanan*、Sarwar Sayeed*、Nikolaos Pitropakis* †
Blockpass ID Lab，爱丁堡龙比亚大学

arXiv:2301.01618v1 [cs.CY] 2 Jan 2023

摘要—**目标：鱼场是养殖鱼类供人食用的区域。鱼场环境有助于在受控环境中照料和管理海产品。过去几十年来，海产品带来的蛋白质热量摄入显著增加。与此同时，鱼类养殖业也蕴含着促进经济发展的重要机遇。确定鱼类疾病、监测水生生物以及检查水元素是否失衡，是需要精确观察的关键因素，以判定采集数据的准确性。同样，随着水产养殖迅速发展，该领域不断采用新技术来提高效率。然而，现有方法往往未能提供高效的鱼类养殖方式。**

方法：本研究摒弃了传统方法，通过采用分布式账本技术开辟了新的方向，以进行精准分析。我们分析了鱼类养殖的当前最新进展，并提出了一种鱼场生态系统，其架构基于 Hyperledger Fabric 私有许可型分布式账本技术，采用隐私保护设计。

结果：所提出的方法能够准确、安全地存储从生态系统中多个传感器检索到的数据，使相关实体能够根据采集数据做出决策。

结论：本研究通过概念验证展现了 Future Fish Farm 的效率和可用性。

关键词：区块链、Hyperledger Fabric、鱼场、安全性、隐私、信任

一、引言

水产养殖是一种养殖方式，与农业采用类似的方法，但养殖对象是鱼类等水生生物，而非植物 [1]。养鱼不仅有助于缩小海产品供需缺口，还能提供一种获取环保蛋白质的途径。此外，与其他蛋白质来源相比，它对消费者而言也是一种高效的选择。水产养殖可采用粗放式或集约化生产方式 [2]。粗放式水产养殖对养殖生物所处环境的监测可能很少，而集约化水产养殖则依赖高度受控的环境，可能需要监测温度、溶解氧和饲料等多项指标，并将其维持在特定的理想水平。

鱼场是一种以水为基础的农业形式，属于水产养殖的一个分支。为了满足鱼类作为蛋白质来源的需求增长，鱼类养殖正在迅速发展 [3]。全球约 62.5% 的养殖鱼类来自河流、湖泊和鱼场；鱼场的核心功能则可能包括鱼类养殖和孵化。鱼场可以利用淡水、海水、咸水或半咸水进行运营。有多种因素需要水产养殖需要在养鱼时加以确保。食物是一种必不可少的物质，它能提供能量，确保鱼类正常生长 [2]。同样，随着鱼类不断演化，鱼场生态系统中的饲料需求也在持续变化。然而，传统鱼场无法持续稳定地供应食物，因而导致大量鱼类死亡。此外，水也是鱼场中的关键要素，是主要鱼类物种生存所必需的条件。然而，为了确保鱼类最大限度地生长，水质变量未必总能维持在适当水平。为应对这些挑战，从鱼场生态系统中的多个不同传感器收集准确数据至关重要。

总体而言，对海产品的需求持续增长，海产品消费在过去五十年中翻了一番 [4]。此外，全球约有15%的蛋白质热量摄入与海产品有关。

海产品产业还可以支持农村地区的经济发展。例如，在苏格兰，苏格兰政府已将水产养殖定义为经济发展的关键领域 [5]。这包括鱼类养殖相关领域，尤其是苏格兰北部和西部地区。其主要目标是通过世界领先的科学与研究，支持健康且可持续发展的苏格兰水产养殖业 [5]。

虽然许多鱼场具备本地数据采集能力，但通常不支持共享所采集的数据。此外，由于建立远程通信渠道的成本高昂，鱼场位置偏远也使数据采集变得困难。因此，通过卫星通信将卫星采集数据直接传输至云环境，可以为本地化数据采集带来诸多益处。然而，这类通信的隐私保护颇具挑战，也经常受到质疑 [6], [7]。此外，所收集数据的安全性和隐私保护仍是一项持续存在的挑战，只有通过从根本上安全的数字技术和方法才能得到保障 [8]。

区块链的迅速普及改变了水产养殖的运营方式，解决了许多棘手难题；与此同时，它还能够以不可篡改的方式存储可信数据，并加快相关任务的整体处理速度。因此，我们的工作概述了如何构建私有许可型区块链基础设施，以收集鱼场环境中多个传感器的数据。尽管许多鱼场会在本地收集数据，但收集到的数据往往缺乏共享机制，同时还面临多方面的安全性和隐私问题 [9]。由于鱼场通常地处偏远，设置远程

通信渠道需要投入成本，因此数据收集往往较为困难。我们的工作深入研究了最先进的方法，并最终提出了一种由区块链加速、适用于苏格兰的现代化互联鱼场系统。

鱼类养殖与现代方法

Wang 等人 [10], 在监测并分析中国的鱼类养殖后得出结论：综合鱼类养殖日益复杂，需要科学界给予更多关注。Choi 等人[11]指出，渔业产品需求不断增长，加之捕捞业面临已发现的局限，水产养殖业可通过提供鱼类种群来应对这一问题。Sangirova 等人[12]也认为，鱼类养殖能够在维持多种商品鱼供应的同时降低鱼类成本。预计到 2030年，水产养殖将占总产量的 60%，捕捞占 40% [13]。

维持鱼场内鱼类健康的关键因素与所提供的水环境质量直接相关 [14], ，具体如下：

- 浊度。浊度用于测量流体的混浊程度，单位为散射浊度单位（NTU）。如果水中悬浮物浓度较高，水体就会显得浑浊。藻类大量繁殖可能导致这一问题，进而危害鱼类健康，例如引发车轮虫病。浊度过高还会影响鱼卵和鱼苗的正常生长 [15]，并导致中毒。

- *pH*值。不同种类的鱼偏好不同的 pH 条件。9–14 之间的 pH 值可能损伤鱼类的细胞膜，而较低的 pH 值会导致沉积物中的岩石物质将金属释放到水中（从而增加浊度）。
- 温度。大多数淡水鱼都是冷血动物，会从周围环境中吸收热量。因此，温度会影响它们的新陈代谢，而温度的快速变化可能会抑制其生长，并使鱼类产生应激 [16]。
- 溶解氧（*DO*）。多项研究表明，水中的溶解氧含量会显著影响鱼类的健康状况 [17] [18]。DO 的测量单位为毫克/升。

2015年，Chen 等人 [19]定义了一种用于鱼类养殖的自动化环境，该环境由多种不同的传感器组成，例如温度传感器、溶解氧传感器、pH传感器和水位传感器，用于监测水槽中的鱼。他们还使用超声波测定水槽中的水位。在输出方面，主要的执行器包括：i）RGB灯光调节系统，通过驱动不同颜色和不同强度的灯光来控制光照输出；ii）加热器，将水加热到所需温度；iii）充氧器，当溶解氧值低于给定值时向水槽中补充氧气；iv）投喂器，在指定时间向鱼投喂；v）电源，为传感器基础设施供电，并在出现供电问题时发挥故障保护作用。Kim 等人 [20] 利用多种传感器和执行器实现了鱼场基础设施。在他们的系统中，他们创建了一个私有网络，其中的传感器连接到一台 Oxyguard 设备和一块 Arduino。

Ullah 等人 [21]开发了一种优化水泵控制的方法，从而在高效节能的同时维持所需的水位。该方法与水泵流量和水槽水位有关，在控制回路中使用消息队列遥测传输（MQTT），并应用卡尔曼滤波器来消除传感器误差。Taniguchi 等人 [22]还使用超声波监测鱼类活动，而 Angani 等人 [23] 则在鳗鱼养殖场中采用人工智能（AI），并结合物联网基础设施和 MQTT。Lee 等人[24]提出了一种优化循环水处理过程控制的方法。

Quek [25] 指出海上鱼类养殖场需要提升电源的韧性，并提出采用基于物联网的直流（DC）纳米电网，该电网使用光伏板。Arafat等人 [14] 定义了一个与物联网相关的鱼场数据集，重点监测水质。该数据集包含9,623条数据记录，涵盖两个不同水位的温度、pH 值和浊度数据。

Yang等人[26]概述了深度学习的应用方法，包括活鱼识别、物种分类、行为分析、投喂决策、尺寸或生物量估算以及水质预测。在鱼类识别方面，Yang等人指出使用 Fish4Knowledge（F4K） [27]–[29] 和克罗地亚鱼类数据集[30]。另一个常用数据集是LifeCLEF 2015（LCF-15）[31] ，该数据集从F4K中提取，包含93段水下视频，涵盖15种鱼类。它包含类别标签和20,000张样本图像。鱼类识别最常用的两种机器学习方法是卷积神经网络（CNN）和区域卷积神经网络（R-CNN） [32],；其中，CNN的准确率分别比支持向量机和Softmax高15%和10%。Meng等人 [33]使用从Google获取的鱼类图像训练CNN，而 Naddaf等人[34] 使用遥控潜水器（ROV）拍摄的视频记录。Salman等人 [35] 使用TensorFlow，通过F4K和 LCF-15数据集训练CNN [31]。

卷积神经网络方法的一个问题是，它们需要通过监督学习进行训练，而且生成的模型质量取决于训练集的质量。因此，Zhao [30] 改进后的深度卷积生成对抗网络方法采用了半监督深度学习（DL）模型。为克服难以获取训练数据的问题，Mahmood 等人 [36] 使用合成数据和目标检测器方法，创建了 You Only Look Once（YOLO） v3 方法。

鱼类物种超过 33,000 种 [26],[37], ，它们的大小、形状和颜色各不相同。不幸的是，环境变化可能多种多样，从而影响分类。深度学习模型通常会尝试学习这些变化并进行补偿。CNN 方法也最常用于此。Siddiqui等人 [38] 使用卷积神经网络，取得了 94.3% 的成功率；Salman等人 [27] 则取得了超过 90% 的准确率，并将卷积神经网络与支持向量机、K近邻等其他方法进行了比较，

SRC、PCA-SVM、PCA-KNN、CNNSVM 和 CNN-KNN，并应用于 LifeCLEF14 [39]和 LifeCLEF15 [31] 数据集。除了视觉方法外，声音也被用于识别鱼类物种。例如，Ibrahim等人 [40] 使用卷积神经网络和长短期记忆网络（LSTM）模型，取得了约 90% 的准确率。

除了鱼类识别和分类之外，鱼类的照料通常还需要监测其行为，尤其是为了支持捕捞和投喂决策 [41]。因此，深度学习已基于时间序列分析以及识别视觉模式的能力得到应用。CNN [26], [42], [43] 和循环神经网络（RNN）方法已被采用，因为它们有助于检测局部行为 [44]。这包括检测鱼群的交叉、重叠和阻挡。

鱼场有效规划的一个关键要素，是所管理鱼群的数量、总量、大小和重量 [26]。这通常通过长度、宽度、重量和面积特征进行估算。然而，由于环境条件的影响（例如光照强度和水体能见度的变化），监测可能颇具挑战，因此有必要应用使用 CNN [45], 、R-CNN [46] 和生成对抗网络（GAN） [45]的方法。

养殖和生产效率中的一个重要因素，是给予鱼类的投喂水平，而这可能是鱼类养殖环境中成本最高的要素之一。与投喂相关的因素很多，包括生理、营养、环境和饲养管理因素 [47]。M˚aløy 等人 [26], [48] 使用时空流结合三维卷积神经网络（3D-CNN）和 RNN 来识别投喂和非投喂行为。

如前所述，水质是鱼类养殖环境中的关键因素，而溶解氧是其中最重要的因素之一。遗憾的是，氧气供应及其对水质的影响可能存在滞后。深度学习方法通过使用卷积神经网络/LSTM [49],RNN [50] 和深度置信网络（DBN） [51]来解决这一问题并建立预测模型。Cordova-Rozas 等人 [13] 的云端监测系统重点关注水质。他们的系统监测了秘鲁一座水族馆中的鱼类品种 $3m \times 1m \times 2m$ 。Siva Kumar 等人 [52]也在其面向智慧水产养殖的云端系统中关注水质，并使用 Blynk私有云集成框架 [53]实时采集数据，监测温度、pH 值、溶解氧和氨。

Tawfeeq 等人 [54] 还通过将鱼场接入配备 ESP8266 的 Wi-Fi网络以及 Things Speak 云数据库，为 Omar 的鱼场实现了云端基础设施，以 [55] 采集温度、水位、pH 值和浊度。Dzulqornain 等人 [56] 提出了一种基于“If This Then That”（IFTTT）模型并结合云集成的水产养殖方案。智能传感器包括池塘区域内的溶解氧、氢离子浓度、水温和水位，区域面积为 $4m \times 5m$。

相关工作

Hang 等人 [4] 定义了一个利用区块链实现信任的安全鱼场平台。根据其方案，使用智能合约自动收集数据，并使用 Hyperledger Fabric 创建原型系统。该系统包括鱼场合约和策略合约。鱼场合约所收集的数据包括：异常值过滤、水位、温度和氧气含量，随后利用这些数据控制水泵。对于策略合约，涉及的实体包括养鱼户、鱼场所有者、多个设备、网络访问策略和业务访问策略。关于信任，每个实体——包括养鱼户、鱼场所有者和每台设备——都拥有公钥和私钥。这些密钥用于识别每个实体的身份，并由证书颁发机构签发。如果实体的私钥遭到泄露，则会发出吊销请求。交易要素包括：采集水位、预测水位、能耗、控制水泵、鱼场用户管理、鱼场传感器管理、执行器管理、预测水位历史记录、能耗历史记录和水泵历史记录。

据我们所知，我们的工作首次介绍了按照隐私保护设计原则开发的鱼场。与现有文献中的解决方案相比，该方案允许特定的参与组织根据其身份凭证查询敏感存储数据，同时阻止其他未经验证的参与者访问。具体而言，通过启用隐私保护功能，我们使鱼场能够存储与其业务连续性策略相关的敏感数据，同时消除传感器提供商泄露数据的风险。我们不应忽视这样一个事实：第三方提供商因安全控制措施不力，导致许多不同组织遭受经济损失 [57]。值得注意的是，相关文献指出，在内部攻击场景中，遭到入侵的传感器提供商可能会窃取其提供给鱼场的传感器所收集的敏感数据。这些收集到的数据之后可能会被出售给出价最高者，从而进一步增加幕后恶意方的利润。在我们的解决方案中，此类攻击无法得逞。

我们可以将本研究的主要贡献总结如下：

- 我们提出了一种新颖的分布式鱼场方案，这是首个在保持其一致性和鲁棒性的同时引入隐私保护设计特性的方案。
- 我们利用 Hyperledger Fabric 的私有数据集合功能实现了所提出的解决方案，从而构建了一个安全且具备隐私保护设计的智能鱼类养殖场。
- 我们基于已知文献确立了评估标准，并据此对智能鱼类养殖场的性能和鲁棒性进行了实证评估。

本文的组织结构如下：第二节详细介绍所提出实现方案所采用的方法和架构。此外，本文着重介绍 HyperledgerFabric 的整体功能和策略，以此说明许可型区块链技术。第三节

首先介绍所提出的 Future Fish Farm 的具体实现，其次全面呈现结果，并通过实验评估相关指标，以证明其效率和安全性。最后，第四节将讨论并得出结论，同时提出未来工作的若干方向。

II. 方法

分布式账本技术与 *Hyperledger Fabric*

分布式账本技术（DLT）是指一种在多个不同地点保持同步的数据库 [58]。其去中心化特性消除了验证或认证交易所需的中介。区块链是近年来最重要的创新之一，通过改变传统的中心化方式带来了巨大的进步。它已成为技术领域的变革者，目前正应用于几乎所有行业。区块链包括公有和私有选项，验证者和最终用户可根据其加入的平台获得访问权限。由于我们提出的 Future Fish Farm 架构采用私有许可型区块链实现，因此我们重点强调此类方法的核心功能。

Hyperledger Fabric 是一个由 Linux基金会支持的项目。其旨在构建私有许可型区块链架构，可应用于多组织协作方式，其中每个组织彼此相连。与其他方法相比，Hyperledger Fabric 的若干关键特性使其独具特色且更加稳健：

- 隐私：Hyperledger Fabric 要求通道内的所有节点都必须通过成员服务提供者（MSP）进行身份识别。这一过程被称为“私有”成员资格；与比特币等公有区块链不同，只有经过授权的成员才可加入 Hyperledger Fabric网络。对于许多关注数据隐私的企业和养殖场而言，Hyperledger Fabric 是一个卓越的选择。此外，Hyperledger Fabric 还可根据需求为架构提供灵活的设计方案；因此，权限要求也可以灵活设置，并根据具体需求进行调整。

- 通道：Hyperledger Fabric 具备一项独特功能，可将区块链账本划分为相互独立的通道，从而允许对等节点生成一组独立的交易，并将其与网络的其他部分隔离开来。当架构由多个域组成，且敏感数据需要与网络中的其他实体隔离时，这种方式十分高效。

- 可扩展性：可扩展性是 Hyperleder Fabric 的另一项显著特性，尤其是在创建大规模架构时。无论节点数量多少，参与节点都可以快速扩展，同时系统仍能以最少的资源处理大量数据。这在区块链基础设施仅由少量节点构建且规模根据需求调整时非常有用。

- 模块化：模块化是 Hyperledger Fabric 区别于其他区块链平台的另一项优势。Hyperledger Fabric 的设计允许在不同阶段添加和实现独立组件。此外，许多组件都是可选的；因此，在有需要时，可以将其完全移除或在后续阶段启用。这使相关域能够决定哪些部分需要在何时实现。Hyperledger Fabric 所包含的一些模块化或“即插即用”组件包括共识、账本存储、对应用程序接口的特定访问权限以及链码集成。

在 Hyperledger Fabric 中，根据既定策略，网络中的大多数节点都必须验证所有交易 [59]。交易验证的整个过程分为几个阶段，通常称为共识。链码的验证、提交和批准通过共识机制完成。达成共识的过程可确保区块链按照既定策略运行。因此，相关节点必须确保交易顺序可靠，并参与交易验证。

Hyperledger Fabric 中完整的共识过程可能包含 3 个阶段：背书、排序和验证。策略决定背书要求，并要求背书对等节点予以确认。排序节点设定待提交交易的顺序，而验证阶段则检查其正确性。就排序而言，在 Hyperledger Fabric 中，部分节点被指定为排序节点，以确保所有对等节点拥有相同的最新账本。某种意义上，排序节点能确保数据一致性得到维护，从而保护区块链的完整性。被专门指定为排序节点的对等节点（或节点）确保通道中的所有对等节点都拥有相同的最新账本。因此，排序对等节点能够确保数据一致性并保护账本的完整性。交易获得背书后，排序节点还会构建区块并将记录写入其中。排序对等节点协同工作时统称为排序服务，它会将新区块发送给通道中的每个对等节点，以更新各自的账本。排序服务是一个模块化组件。需要注意的是，在 Fabric 网络中，实现排序服务有多种方法。最后，每个对等节点都会验证按顺序排列的交易。由于交易按顺序排列，对等节点可以检查后续交易是否因先前的交易而被拒绝。此类验证可防止双重支付或数据不一致的情况。

在 Hyperledger Fabric 中，可以根据参与组织定义策略。背书策略规定，通道上的对等节点集合可以通过执行链码并为结果背书，参与交易验证过程。尽管背书策略无法确保正确的链码部署在正确的对等节点上，但另一种机制——“背书”并安装

链码包会执行此类检查。背书策略示例包括：i）通道中的所有对等节点都可以为一笔交易背书；ii）通道中的大多数对等节点都可以参与背书；iii）在某个通道中，对等节点必须为一笔交易背书。

架构

所提出解决方案的架构如图1所示。其技术架构基于一个计算测试平台，该平台采用 Ubuntu 20.04 LTS 操作系统，配备第8代 i7 CPU（6核，3.20GHz）、32GB RAM 和 1TB 固态硬盘。我们选择的分布式账本技术是 Hyperledger Fabric 私有许可型区块链框架，其交易处理速度比其他公有区块链更快 [60]。此外，由于 Hyperledger Fabric 的共识机制具有灵活性，因此可以根据用例的实施情况调整该技术的规格；因此，这一基础设施还可以扩展到其他类似用例，以便采用数据采集工具 [61]。

我们实施方案的拓扑结构和具体技术细节 [61] 如下：

1) 每个传感器提供方都作为 Hyperledger Fabric 对等节点，拥有对区块链账本的存储访问权限。2) 每个鱼场也都是 Hyperledger Fabric 对等节点，但仅拥有对区块链账本的查看权限。此外，还通过私有数据集合功能 [62], [63] 进一步配置区块链账本的查看权限，以区分各参与者的查看权限。此功能类似于其他计算系统中的访问控制策略。3) 我们基础设施中的对等节点，即 *developers.sensorsprovider.org*、*support.sensorsprovider.org*、*admin.fishfarm.org*和 *user.fishfarm.org*，保存区块链账本、依据既定策略定义的私有数据集合，并记录任何数据篡改。对等节点使用的状态数据库是 CouchDB 实例。4) 每个对等节点的身份由 X.509 证书标识，成员服务提供者（MSP）实体会验证该证书的有效性。5) 对等节点组可以构成 Hyperledger Fabric 组织。在我们的架构中，组织负责依据既定策略接受或拒绝每笔区块链交易。在技术实验中，指定了两个组织，即*sensorsproviders.org* 和 *fishfarm.org*。6) 排序服务在我们的案例中采用崩溃容错 RAFT 服务，它会创建新的区块链区块，并依据既定策略将其广播给所有参与的对等节点。因此，每笔存储交易均由三个排序节点处理，以避免单排序节点基础设施面临的潜在单点故障。需要注意的是，排序节点的数量可以任意设定；我们特意选择三个仅用于实验。排序节点的数量与其他 Hyperledger Fabric组件的数量并无关联。

排序节点的数量与其他 Hyperledger Fabric组件的数量并无关联。

7）我们解决方案中的智能合约，即 Hyperledger Fabric 中的链码，正在参与组织的所有对等节点和排序服务中进行审批和安装。链码使用 Go 编程语言编写。

8) 我们根据基础设施的数据字段生成并使用了合成数据。

9) 基础设施的策略在区块链初始化期间指定，但也可以进一步更新，以纳入新的区块链规则。在更新场景中，新策略需要获得若干参与组织和排序服务的批准，这与链码更新类似。

III. 结果

概念验证与访问控制策略

本小节定义了所开发的概念验证（PoC）以及系统详细的访问控制策略。该概念验证涉及使用 Minifabric 框架[1]搭建一个基于 Hyperledger Fabric 2.3.0 版的许可型区块链。

有两个不同的私有数据集合，分别是 *collectionFishFarm* 和*collectionFishFarmPrivateDetails*。我们假设*sensorsprovider.org* 组织是监测鱼场的传感器提供方，而*fishfarm.org* 组织则是安装这些传感器的鱼场。不过，传感器提供方通常需要访问传感器以进行维护，因此可以访问其传感器收集的数据。在所展示的测试平台中，鱼场可以利用基础设施，仅向传感器提供方披露必要信息，而不暴露任何已收集的敏感细节。这些敏感细节可能包括特定鱼场指标的信息，鱼场可以进一步利用这些信息、将其商业化、申请政府资助等。

因此，总共有 14 个数据字段，进一步划分为双方组织均可访问的数据字段，例如风速、降雨量、气压、温度、波高和水流；以及只有*fishfarm.org*组织可访问的私有数据字段，例如荧光溶解有机物（荧光溶解有机物）、盐度、*pH* 值、浊度、藻类、氧化还原电位（氧化还原电位）和硝酸盐。最后还有一个额外的数据字段，即名称，它是连接这两个数据集合的键 。正如拓扑结构中所述，这些数据字段中使用的数据均为合成生成。在真实场景中，相应的数据字段可以通过谨慎布设在鱼场中的物理传感器采集。

此概念验证（PoC）涉及的每个节点都以 Docker 容器的形式开发，并在加入通道之前由网络管理员授权。管理员向鱼场生态系统中的对等节点颁发数字证书。成员服务提供者（MSP）负责定义身份验证、

[1]Minifabric 框架：https://github.com/hyperledger-labs/minifabric

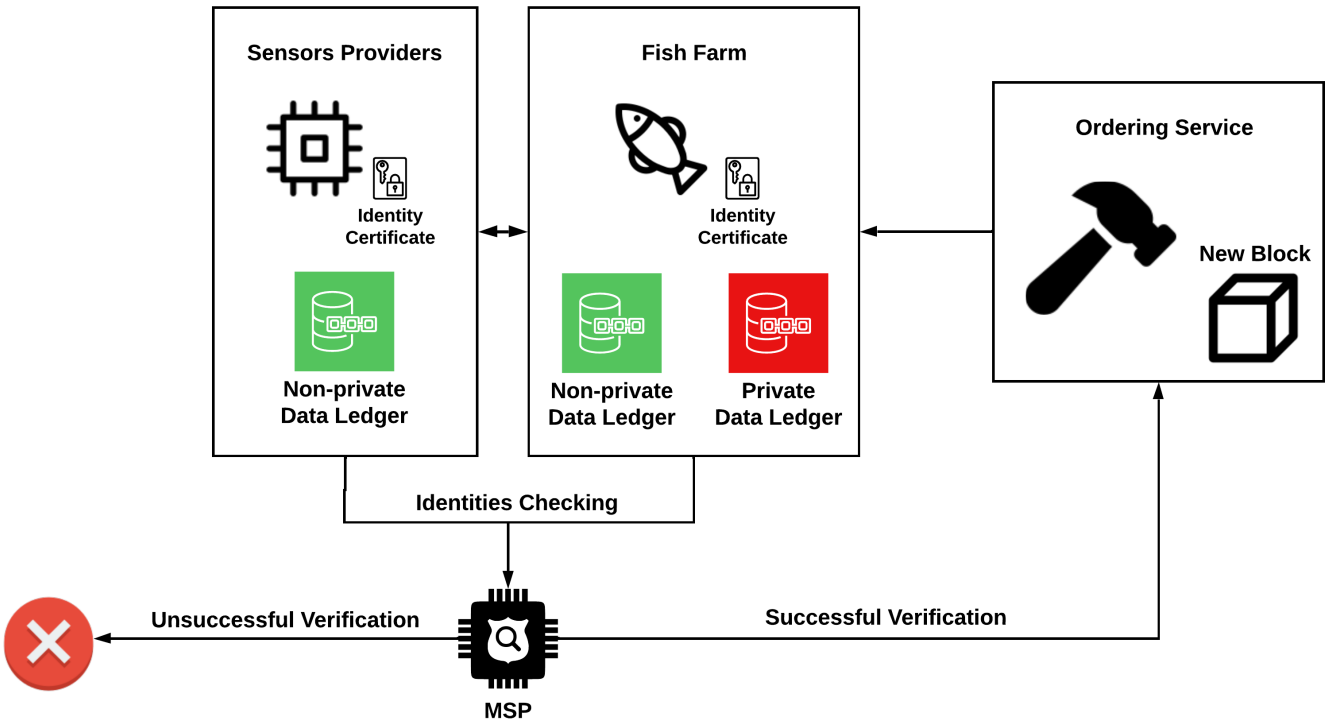


图 1：鱼场架构概览。两个参与组织对该系统拥有不同级别的访问权限。传感器提供商组织只能访问非私有数据账本，而鱼场组织还可以访问私有数据账本。未经授权的参与者将被拒绝访问。

身份认证以及获准访问网络 [64] 的规则。MSP 利用证书颁发机构（CA），该机构负责创建和撤销身份凭证。同样，网络中的每个实体都会获得 X.509 证书。Hyperledger Fabric 的模块化基础设施允许使用外部证书颁发机构。

评估

在本小节中，可以从性能和安全性方面了解系统的结果和实验评估。正如文献中所观察到的 [62], [63], ，与使用不同技术开发的其他类似系统相比，Hyperledger Fabric 的效率非常高。然而，尽管 Minifabric 框架的加入有助于概念验证的开发工作，但系统性能有所下降。执行一次写事务大约需要 7.3 秒（附录 A），而执行一次读事务大约需要 6.9 秒（附录 B）。因此，如 Hyperledger Explorer 所示，概念验证的吞吐量介于每分钟 7 至 8 笔交易之间，如图 2 所示。这一计算开销是由于执行*minifab* 脚本（Minifabric 框架的一部分）所导致的，该脚本负责管理 Hyperledger Fabric 环境。然而，在生产环境中开发基础设施时，如果采用传统的 Hyperledger Fabric 实践，则可以避免这一开销。尽管如此，当概念验证扩展到 100,000 条存储记录时，系统性能仍保持不变，这一结果符合预期的预期，也是所提出解决方案优于其他技术的证明（附录 B ）。此外，还使用 Python 的 *Matplotlib* 通过绘图对系统性能进行了实验评估和可视化。系统存储 100,000 条记录时的 CPU 性能如图 3 所示。该图显示，参与的每个对等节点在区块链系统中存储每条记录时，CPU 使用率都会持续波动。然而，在大多数情况下，这些波动都处于 0–20% 范围内（如命令行界面中以紫色显示的部分所示），并且经过一段时间后，所有参与对等节点的 CPU 使用率都会出现峰值。因此，据推测，这些 CPU 使用率峰值是由于在我们无法控制的情况下使用 Minifabric 框架，以及潜在的硬件限制和其他环境影响所造成的。

读取任意数量记录时的 CPU 性能，以及任何读写交易的 RAM 使用量都可以忽略不计（CPU 性能 ¡ 5%，RAM 使用量 ¡ 1%），因此未绘制在图中。由于 PREHEALTH [62]和 PRESERVE DNS [63]，使用的是“原版”Hyperledger Fabric，而非 Future Fish Farm 所基于的 Minifabric 框架，因此无法与文献中采用私有数据收集功能的其他工作进行公平的性能比较。如前所述，在本研究中，读事务约需 6.9 秒，而在 PREHEALTH 和 PRESERVE DNS 中约需 0.83 秒。

关于这两个数据集合，正如

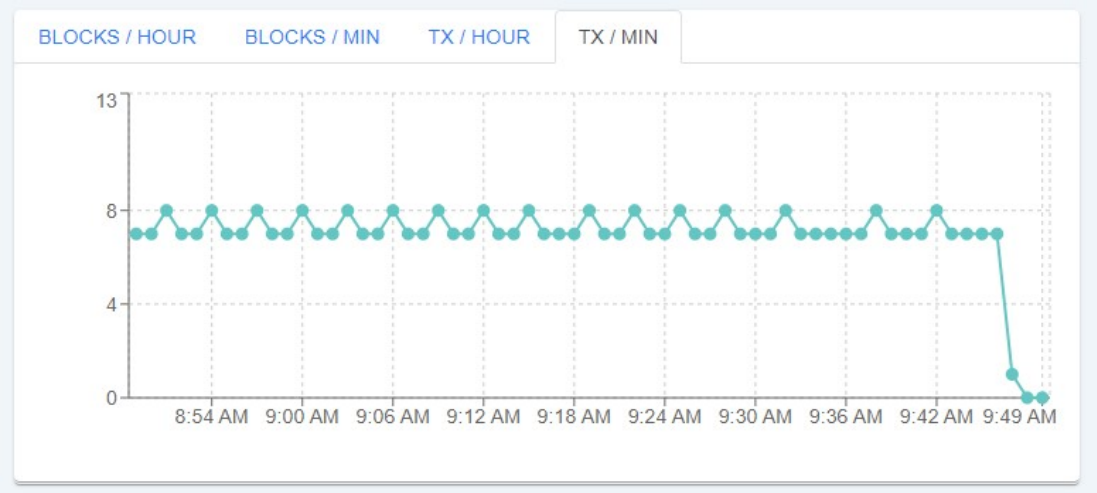


图 2：概念验证的吞吐量（每分钟交易数）。

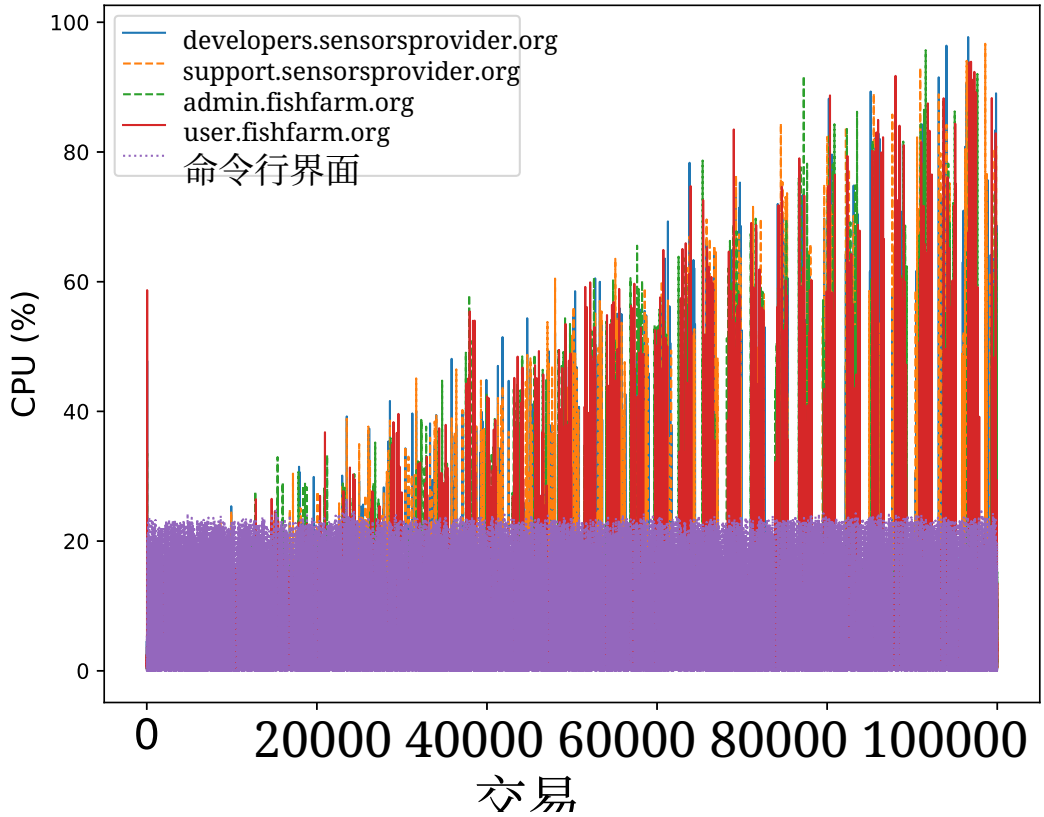


图 3：将 100,000 笔交易写入账本所需的 CPU 使用率。

前文所述，只有*fishfarm.org*组织的对等节点有权访问私有数据集合，即*collectionFishFarmPrivateDetails*；而来自两个参与方的所有对等节点都可以访问*collectionFishFarm*（附录 C）。由于这些其他方的身份凭证未包含在指定策略中（附录 C），因此它们无法访问存储在这两个数据集合中的数据。

IV. 讨论

鱼类养殖是动物食品生产中发展最快的通道。全球消费的鱼类中，有一半是在人工环境中生产的。本文提出了 Future Fish Farm 的概念，旨在展示如何通过对采集数据进行智能观测，做出知情决策。Future Fish Farm 的架构采用复杂方法，要求采集数据的准确性和可靠性、决策模型以及各类智能系统之间的关联均能正常运行。

我们的解决方案在测试平台中实现，清晰展示了新型架构所提出的功能，从而打造出效果更佳、性能效率更高的 Future Fish Farm。值得强调的是，这是首个使鱼场能够在不冒敏感数据可能暴露给遭到入侵或带有恶意的传感器提供商之风险的情况下采集敏感数据的解决方案。在恶意数据经纪人横行的世界里，恶意传感器提供商可能会窃取关键鱼场数据，然后将其出售给其他第三方，甚至篡改数据，诱使鱼场做出不必要的采购决策。这不仅会影响特定鱼场，还会波及整个供应链，给各类组织的业务连续性带来问题。不过，我们的解决方案能够阻断这类攻击向量，并保障安全性与隐私。

在未来的工作中，我们计划通过在更复杂、模拟真实世界用例的场景中增加更多功能和参与者，进一步开发 Future Fish Farm 基础设施。此外，利用人工智能技术从存储的数据中获取更多洞见，也是确定 Future Fish Farm 在真实环境中可用性的一项重要后续工作。

声明

作者贡献

对研究的构思和设计作出了重要贡献，并进行了数据分析和结果阐释：Papadopoulos P.、Buchanan W.、Sayeed S.、Pitropakis N.

进行了数据采集，并提供行政、技术和材料支持：Papadopoulos P.

数据和材料的可获取性

不适用。

财务支持与赞助

无。

利益冲突

所有作者均声明不存在利益冲突。

伦理批准与参与同意

不适用。

发表同意

不适用。

版权

© 作者 2022。

参考文献

[1] 全球海鲜联盟，《什么是水产养殖？我们为什么需要它？》https://www.globalseafood.org/blog/what-is-aquaculture-why-do-we-need-it/，2019年，在线；访问日期：2022年5月19日。[2] Peter J. Allen，《水产养殖：挑战与前景》，https://www.nature.com/scitable/knowledge/library/aquaculture-challenges-and-promise-23690921/#TB inline?height 300 =400 = 2011, [19 &width & inlineId trOutLine，在线；访问日期：2022]年[3] H. Kim，《ChirpStack，开源LoRaWAN(R)网络服务器协议栈》，2021年。 [在线]。可访问：https://sentientmedia.org/fish-farmer [4] L. Hang、I. Ullah和D.-H. Kim，《基于区块链的安全鱼场平台：保障农业数据完整性》，《农业中的计算机与电子学》，第170卷，第105251页，2020年。[5] S. Govern ment，《水产养殖》，2021年。 [在线]。可访问：https: // www.gov.scot/policies/aquaculture/ [6] A. T. McKenna、A. C. Gaudion和J. L. Evans，《卫星与智能设备的作用：数据惊喜以及安全性、隐私和监管挑战》，《宾夕法尼亚州法律评论》，第123卷，第591页，2018年。[7] M. Stoyanova、Y. Nikoloudakis、S. Panagiotakis、E. Pallis和E. K. Markakis，《物联网取证综述：挑战、方法与开放性问题》，《IEEE通信调查与教程》，第22卷，第2期，第1191–1221页，2020年。[8] M. Amiri-Zarandi、R. A. Dara、E. Duncan和E. D. Fraser，《智慧农业中的大数据隐私：综述》，《可持续发展》，第14卷，第15期，第9120页，2022年。[9] A. Andreas、C. X. Mavromoustakis、G. Mastorakis、D.-T. Do、J. M. Batalla、E. Pallis和E. K. Markakis，《面向物联网设备安全优化的方法：机密医疗数据交换》，《多媒体工具与应用》，第80卷，31 449，2021年20，31 435，无页码。[10] H.-X. Wang，《中国综合鱼类养殖的现状与前景》，载于《综合鱼类养殖》。泰勒与弗朗西斯，2020年，第45–54页。[11] J. Choi、J. Lee、Y. Kim和Y. Shin，《未来智能鱼场平台建设方案研究》，《KIPS计算机与通信系统汇刊》，第9卷，第7期，第157–164页，2020年。[12] U. Sangirova、Z. Khafizova、I. Yunusov、B. Rakhmankulova和U. Kholiyorov，《发展网箱养鱼的益处》，载于《E3S会议网络》，第217卷。EDP Sciences，2020年，第09006页。[13] M. Cordova-Rozas、J. Aucapuri-Lecarnaque和P. Shiguihara-Ju´arez，《一种使用物联网的水产养殖云监测系统》，载于《2019年IEEE科学与人文国际研究会议（SHIRCON）》。IEEE，2019年，第1–4页。[14] A. I. Arafat、T. Akter、M. F. Ahammed、M. Y. Ali和A.-A. Nahid，《一个基于物联网的鱼场监测与通知系统数据集》，《数据简报》，第33卷，第106457页，2020年。[15] A. H. Auld和J. Schubel，《悬浮泥沙对鱼卵和幼体的影响：实验室评估》，《河口与海岸海洋科学》，第6卷，第2期，第153–164页，1978年。[16] A.-F. El-Sayed、A. El-Ghobashy和M. Al-Amoudi，《池塘深度和水温对尼罗罗非鱼（Oreochromis niloticus (L.)）生长、死亡率和身体组成的影响》，《水产养殖研究》，第27卷，第9期，第681–687页，1996年。[17] D. Solstorm、T. Oldham、F. Solstorm、P. Klebert、L. H. Stien、T. V˚ agseth和F. Oppedal，《商业海上网箱中的溶解氧变化使养殖大西洋鲑暴露于限制生长的条件下》，《水产养殖》，第486卷，第122–129页，2018年。[18] S. E. Null、N. R. Mouzon和L. R. Elmore，《溶解氧、溪流水温和鱼类栖息地对环境用水采购的响应》，《环境管理杂志》，第197卷，第559–570页，2017年。[19] J.-H. Chen、W.-T. Sung和G.-Y. Lin，《鱼场水产养殖环境自动监测系统》，载于《2015年IEEE系统、人类与控制论国际会议》。IEEE，2015年，第1161–1166页。[20] Y. Kim、N. Lee、B. Kim和K. S hin，《使用移动应用实现基于物联网的鱼场控制》，载于《2018年计算机、消费电子与控制国际研讨会（IS3C）》。IEEE，2018年，第189–192页。[21] I. Ullah和D. Kim，《智能鱼场中高效能耗水泵控制的优化方案》，《流程》，第6卷，第6期，第65页，2018年。[22] Y. Taniguchi，《鱼场监测声学数据通信的评估》，《DNCOCO 2015论文集》，第195–199页，2015年。

[23] A. Angani、S. M. Oh、E. S. Kim和K. J. Shin，《人工智能鳗鱼养殖场的实现（第2部分）：使用MQTT的基于物联网的流量控制》，载于《2019年IEEE建筑、施工、环境与水利国际会议（ICACEH）》。IEEE，2019年，第97–100页。[24] J. Lee、A. Angani、T. Thalluri和K. jae Shin，《智能鱼场水处理过程控制的实现》，载于《2020年电子、信息与通信国际会议（ICEIC）》。IEEE，2020年，第1–5页。[25] Y. Quek，《海上鱼类养殖场中基于物联网的直流纳米电网的实施》，载于《2020年IEEE第十区会议（TENCON）》。IEEE，2020年，第1147–1152页。[26] X. Yang、S. Zhang、J. Liu、Q. Gao、S. Dong和C. Zhou，《智能鱼类养殖的深度学习：应用、机遇与挑战》，《水产养殖评论》，第13卷，第1期，第66–90页，2021年。[27] A. Salman、A. Jalal、F. Shafait、A. Mian、M. Shortis、J. Seager和E. Harvey，《基于深度学习的非受限水下环境鱼类物种分类》，《湖沼学与海洋学：方法》，第14卷，第9期，第570–585页，2016年。[28] X. Sun、J. Shi、L. Liu、J. Dong、C. Plant、X. Wang和H. Zhou，《将深度知识迁移用于低质量水下视频中的目标识别》，《神经计算》，第275卷，第897–908页，2018年。[29] R. B. Fisher、Y.-H. Chen-Burger、D. Giordano、L. Hardman、F.-P. Lin等，《Fish4Knowledge：收集和分析大规模珊瑚礁鱼类视频数据》。施普林格，2016年，第104卷。[30] J. Zhao、Y. Li、F. Zhang、S. Zhu、Y. Liu、H. Lu和Z. Ye，《利用改进的深度卷积生成对抗网络进行基于半监督学习的水产养殖活鱼识别》，《美国农业与生物工程师学会会刊》，第61卷，第2期，第699–710页，2018年。[31] A. Joly、H. Go¨eau、H. Glotin、C. Spampinato、P. Bonnet、W.-P. Vellinga、R. Planqu´e、A. Rauber、S. Palazzo、B. Fisher等，《LifeCLEF 2015：生物物种多媒体识别挑战》，载于《欧洲语言跨语言评测论坛国际会议》。施普林格，2015年，第462–483页。[32] A. B. Labao和P. C. Naval Jr，《具有链接集成组件的级联深度网络系统：用于野外水下鱼类检测》，《生态信息学》，第52卷，第103–121页，2019年。[33] L. Meng、T. Hirayama和S. Oyanagi，《基于深度学习的配备全景相机的水下无人机自动鱼类识别》，《IEEE开放获取》，第6卷，第17 880–17 886页，2018年。[34] M. Naddaf-Sh、H. Myler、H. Zargarzadeh等，《用于AUV/遥控水下航行器的辅助式实时红狮子鱼检测系统的设计与实现》，《复杂性》，第2018卷，2018年。[35] A. Salman、S. A. Siddiqui、F. Shafait、A. Mian、M. R. Shortis、K. Khurshid、A. Ulges和U. Schwanecke，《通过基于深度神经网络的混合运动学习系统自动检测水下视频中的鱼类》，《ICES海洋科学杂志》，第77卷，第4期，第1295–1307页，2020年。[36] A. Mahmood、M. Bennamoun、S. An、F. Sohel、F. Boussaid、R. Hovey和G. Kendrick，《使用合成数据自动检测西部岩龙虾》，《ICES海洋科学杂志》，第77卷，第4期，第1308–1317页，2020年。[37] T. Oosting、B. Star、J. H. Barrett、M. Wellenreuther、P. A. Ritchie和N. J. Rawlence，《在基因组学时代释放古代鱼类DNA的潜力》，《进化应用》，第12卷，1522，2019年8，1513，无页码。[38] S. A. Siddiqui、A. Salman、M. I. Malik、F. Shafait、A. Mian、M. R. Shortis和E. S. Harvey，《水下视频中的鱼类物种自动分类：利用预训练深度神经网络模型弥补标注数据有限的问题》，《ICES海洋科学杂志》，第75卷，第1期，第374–389页，2018年。[39] A. Joly、H. Go¨eau、H. Glotin、C. Spampinato、P. Bonnet、W.-P. Vellinga、R. Planque、A. Rauber、R. Fisher和H. M¨uller，《LifeCLEF 2014：生物物种多媒体识别挑战》，载于《欧洲语言跨语言评测论坛国际会议》。施普林格，2014年，第229–249页。[40] A. K. Ibrahim、H. Zhuang、L. M. Ch´erubin、M. T. Sch¨arer-Umpierre和N. Erdol，《使用深度神经网络根据声音自动分类石斑鱼物种》，《美国声学学会杂志》，第144卷，第3期，第EL196–EL202页，2018年。[41] V. M. Papadakis、I. E. Papadakis、F. Lamprianidou、A. Glaropoulos和M. Kentouri，《用于分析鱼类行为的计算机视觉系统与方法》，《水产养殖工程》，第46卷，59，2012年53页–。[42] J. Li、C. Xu、L. Jiang、Y. Xiao、L. Deng和Z. Han，《基于深度学习的海参行为轨迹检测与分析》，《IEEE开放获取》，第8卷，第18 832–18 840页，2019年。[43] F. Romero-Ferrero, M. G. Bergomi, R. C. Hinz, F. J. Heras, and G. G. de Polavieja，《Idtracker. ai：追踪小型或大型无标记动物群体中的所有个体》，《自然·方法》，第16卷，第2期，第179–182页，2019年。[44] J. Zhao, W. Bao, F. Zhang, S. Zhu, Y. Liu, H. Lu, M. Shen, and Z. Ye，《基于改进运动影响图和循环神经网络的集约化水产养殖鱼群局部异常行为监测》，《水产养殖》，第493卷，第165–175页，2018年。[45] D. Levy, Y. Belfer, E. Osherov, E. Bigal, A. P. Scheinin, H. Nativ, D. Tchernov, and T. Treibitz，《利用有限数据自动分析海洋视频》，载于《IEEE计算机视觉与模式识别会议研讨会论文集》，2018年，第1385–1393页。[46] A. ´Alvarez-Ellacur´ıa, M. Palmer, I. A. Catal´an, and J.-L. Lisani，《基于图像和深度学习的商业渔获上岸鱼体尺寸无监督估算》，《ICES海洋科学杂志》，第77卷，第4期，第1330–1339页，2020年。[47] M. Sun, S. G. Hassan, and D. Li，《水产养殖摄食量估算模型：综述》，《农业中的计算机与电子学》，第127卷，第425–438页，2016年。[48] H. M˚aløy, A. Aamodt, and E. Misimi，《利用水下视频识别水产养殖中鲑鱼摄食动作的时空循环网络》，《农业中的计算机与电子学》，第167卷，第105087页，2019年。[49] X. Ta and Y. Wei，《基于卷积神经网络的循环水养殖系统溶解氧预测方法研究》，《农业中的计算机与电子学》，第145卷，第302–310页，2018年。[50] Y. Liu, Q. Zhang, L. Song, and Y. Chen，《用于准确预测短期和长期溶解氧的基于注意力机制的循环神经网络》，《农业中的计算机与电子学》，第165卷，第104964页，2019年。[51] Q. Lin, W. Yang, C. Zheng, K. Lu, Z. Zheng, J. Wang, J. Zhu et al.，《基于深度学习的集约化虾塘水质预测方法》，《印度渔业杂志》，第65卷，第4期，第75–80页，2018年。[52] S. Sivakumar and V. Ramya，《一种利用云计算监测鱼塘水质的直观远程监测框架》，载于《IOP会议系列：材料科学与工程》，第1085卷，第1期。IOP Publishing，2021年，第012037页。[53] Blynk，《Blynk物联网平台》，2021年。[在线]。地址：https://blynk.io/[54] A. Tawfeeq, H. A. S. Al Wahaibi, and K. Vijayalakshmi，《基于物联网并结合云分析的水产养殖系统》，《国际应用工程研究杂志》，第14卷，第22期，第4136–4142页，2019年。[55] T. Speak，《物联网分析 - ThingSpeak》，2021年。[在线]。地址：https://thingspeak.com/[56] M. I. Dzulqornain, M. U. H. Al Rasyid, and S. Sukaridhoto，《基于IFTTT模型和云集成的智能水产养殖系统设计与开发》，载于《MATEC网络会议》，第164卷。EDP Sciences，2018年，第01030页。[57] T. Dubendorfer, A. Wagner, and B. Plattner，《大规模互联网攻击的经济损失模型》，载于《第13届IEEE协作企业基础设施使能技术国际研讨会》。IEEE，2004年，第223–228页。[58] M. P. Network，《区块链与分布式账本技术的区别》，2021年。[在线]。地址：https://sentientmedia.org/fish-farmer[59] C. Gorenflo, S. Lee, L. Golab, and S. Keshav，《Fastfabric：将Hyperledger Fabric扩展至每秒20 000笔交易》，《国际网络管理杂志》，第30卷，第5期，第e2099页，2020年。[60] A. Baliga, N. Solanki, S. Verekar, A. Pednekar, P. Kamat, and S. Chatterjee，《Hyperledger Fabric的性能表征》，载于《2018年加密谷区块链技术会议（CVCBT）》。IEEE，2018年，第65–74页。[61] P. Papadopoulos, N. Pitropakis, and W. J. Buchanan，《去中心化隐私：一种分布式账本方法》，载于《智能材料、技术与器件手册：工业4.0的应用》。施普林格，2022年，第1–26页。[62] C. Stamatellis, P. Papadopoulos, N. Pitropakis, S. Katsikas, and W. J. Buchanan，《一种基于Hyperledger Fabric的隐私保护医疗框架》，《传感器》，第20卷，第22期，第6587页，2020年。[63] P. Papadopoulos, N. Pitropakis, W. J. Buchanan, O. Lo, and S. Katsikas，《隐私保护被动DNS》，《计算机》，第9卷，第3期，第64页，2020年。[64]Moses Sam Paul，《Hyperledger Fabric组件——技术背景》，https://medium.com/swlh/hyperledger-chapter-6-hyperledger-fabric-components-technical-context-2018，[在线；访问日期：2021]年 8 月 12 日。

附录 A 写事务 写事务如图 4 所示。

附录 B
读事务

第 1 条记录的读事务见图 5。第 100,000 条记录的读事务见图 6。

附录 C 私有数据集合

对私有数据集合的成功读事务见图 7。对私有数据集合的不成功读事务见图 8。

```
Using spec file: /mnt/g/Projects/fish-farm/spec.yaml
Minifab Execution Context:
    FABRIC_RELEASE=2.3.0
    CHANNEL_NAME=mychannel
    PEER_DATABASE_TYPE=couchdb
    CHAINCODE_LANGUAGE=go
    CHAINCODE_NAME=fishfarm
    CHAINCODE_VERSION=1.0
    CHAINCODE_INIT_REQUIRED=true
    CHAINCODE_PARAMETERS="initMetrics"
    CHAINCODE_PRIVATE=true
    CHAINCODE_POLICY=
    TRANSIENT_DATA={"metrics":"eyJuYW1lIjoibWV0cmljczEwNjg1Iiwid2luZHNwZWVkIjoiMC41bXBoIiwicmFpbmZhbGwiOiIwLjVtbSIsImFpcnByZXNzdXJlIjoiNSBiYXJzIiwidGVtcGVyYXR1cmUiOiIyMCBDZWxzaXVzIERlZ3JlZXMiLCAid2F2ZWhlaWdodCI6IjFtIiwgIndhdGVyY3VycmVudCI6IjVtL3MiLCAiZmRvbSI6IjI1IiwgInNhbGluaXR5IjoiNWcvTCIsICJwaCI6IjciLCAidHVyYmlkaXR5IjoiMiIsICJhbGdhZSI6IjUiLCAib3JwIjoiMTBWIiwgIm5pdHJhdGVzIjoiMjAwcHBtIn0K"}
    BLOCK_NUMBER=newest
    EXPOSE_ENDPOINTS=false
    CURRENT_ORG=sensorsprovider.org
    HOST_ADDRESSES=172.30.65.83
    WORKING_DIRECTORY: /mnt/g/Projects/fish-farm
.......
# Preparing for the following operations: ********************
  verify options, cc invoke
.......
# Running operation: ****************************************
  verify options
.
# Running operation: ****************************************
  cc invoke
..............
# Chaincode invocation results ******************************
  ['\x1b[34m2022-05-13 17:01:43.923 UTC [chaincodeCmd] chaincodeInvokeOrQuery -> INFO 001\x1b[0m Chaincode invoke successful. result: status:200 ']

# STATS *****************************************************
minifab: ok=34  failed=0
```

图4：写事务

```
Using spec file: /mnt/g/Projects/fish-farm/spec.yaml
Minifab Execution Context:
    FABRIC_RELEASE=2.3.0
    CHANNEL_NAME=mychannel
    PEER_DATABASE_TYPE=couchdb
    CHAINCODE_LANGUAGE=go
    CHAINCODE_NAME=fishfarm
    CHAINCODE_VERSION=1.0
    CHAINCODE_INIT_REQUIRED=true
    CHAINCODE_PARAMETERS="readMetrics","metrics1"
    CHAINCODE_PRIVATE=true
    CHAINCODE_POLICY=
    TRANSIENT_DATA=
    BLOCK_NUMBER=newest
    EXPOSE_ENDPOINTS=false
    CURRENT_ORG=sensorsprovider.org
    HOST_ADDRESSES=172.30.65.83
    WORKING_DIRECTORY: /mnt/g/Projects/fish-farm
.......
# Preparing for the following operations: ********************
  verify options, cc query
.......
# Running operation: ****************************************
  verify options
.
# Running operation: ****************************************
  cc query
......
# Chaincode query results ***********************************
  ['{"airpressure":"1 bars","docType":"metrics","name":"metrics1","rainfall":"0.1mm","temperature":"20 Celsius Degrees","watercurrent":"5m/s","waveheight":"1m","windspeed":"0.1mph"}']

# STATS *****************************************************
minifab: ok=26  failed=0
```

图5：读事务的第一条记录

```
Using spec file: /mnt/g/Projects/fish-farm/spec.yaml
Minifab Execution Context:
    FABRIC_RELEASE=2.3.0
    CHANNEL_NAME=mychannel
    PEER_DATABASE_TYPE=couchdb
    CHAINCODE_LANGUAGE=go
    CHAINCODE_NAME=fishfarm
    CHAINCODE_VERSION=1.0
    CHAINCODE_INIT_REQUIRED=true
    CHAINCODE_PARAMETERS="readMetrics","metrics100000"
    CHAINCODE_PRIVATE=true
    CHAINCODE_POLICY=
    TRANSIENT_DATA=
    BLOCK_NUMBER=newest
    EXPOSE_ENDPOINTS=false
    CURRENT_ORG=sensorsprovider.org
    HOST_ADDRESSES=172.30.65.83
    WORKING_DIRECTORY: /mnt/g/Projects/fish-farm
.......
# Preparing for the following operations: ********************
  verify options, cc query
.......
# Running operation: ****************************************
  verify options
.
# Running operation: ****************************************
  cc query
......
# Chaincode query results ***********************************
  ['{"airpressure":"10 bars","docType":"metrics","name":"metrics100000","rainfall":"3.5mm","temperature":"22 Celsius Degrees","watercurrent":"10m/s","waveheight"
:"3m","windspeed":"3.5mph"}']

# STATS *****************************************************
minifab: ok=26  failed=0
```

图 6：读取第 100,000 条记录的读事务

```
Using spec file: /mnt/g/Projects/fish-farm/spec.yaml
Minifab Execution Context:
    FABRIC_RELEASE=2.3.0
    CHANNEL_NAME=mychannel
    PEER_DATABASE_TYPE=couchdb
    CHAINCODE_LANGUAGE=go
    CHAINCODE_NAME=fishfarm
    CHAINCODE_VERSION=1.0
    CHAINCODE_INIT_REQUIRED=true
    CHAINCODE_PARAMETERS="readMetricsPrivateDetails","metrics100000"
    CHAINCODE_PRIVATE=true
    CHAINCODE_POLICY=
    TRANSIENT_DATA=
    BLOCK_NUMBER=newest
    EXPOSE_ENDPOINTS=false
    CURRENT_ORG=fishfarm.org
    HOST_ADDRESSES=172.30.65.83
    WORKING_DIRECTORY: /mnt/g/Projects/fish-farm
.......
# Preparing for the following operations: ********************
  verify options, cc query
.......
# Running operation: ****************************************
  verify options
.
# Running operation: ****************************************
  cc query
......
# Chaincode query results ***********************************
  ['{"algae":"7","docType":"metricsPrivateDetails","fdom":"30","name":"metrics100000","nitrates":"300ppm","orp":"20V","ph":"8","salinity":"10g/L","turbidity":"5"
}']

# STATS *****************************************************
minifab: ok=26  failed=0
```

图 7：成功读取私有数据集合的读事务

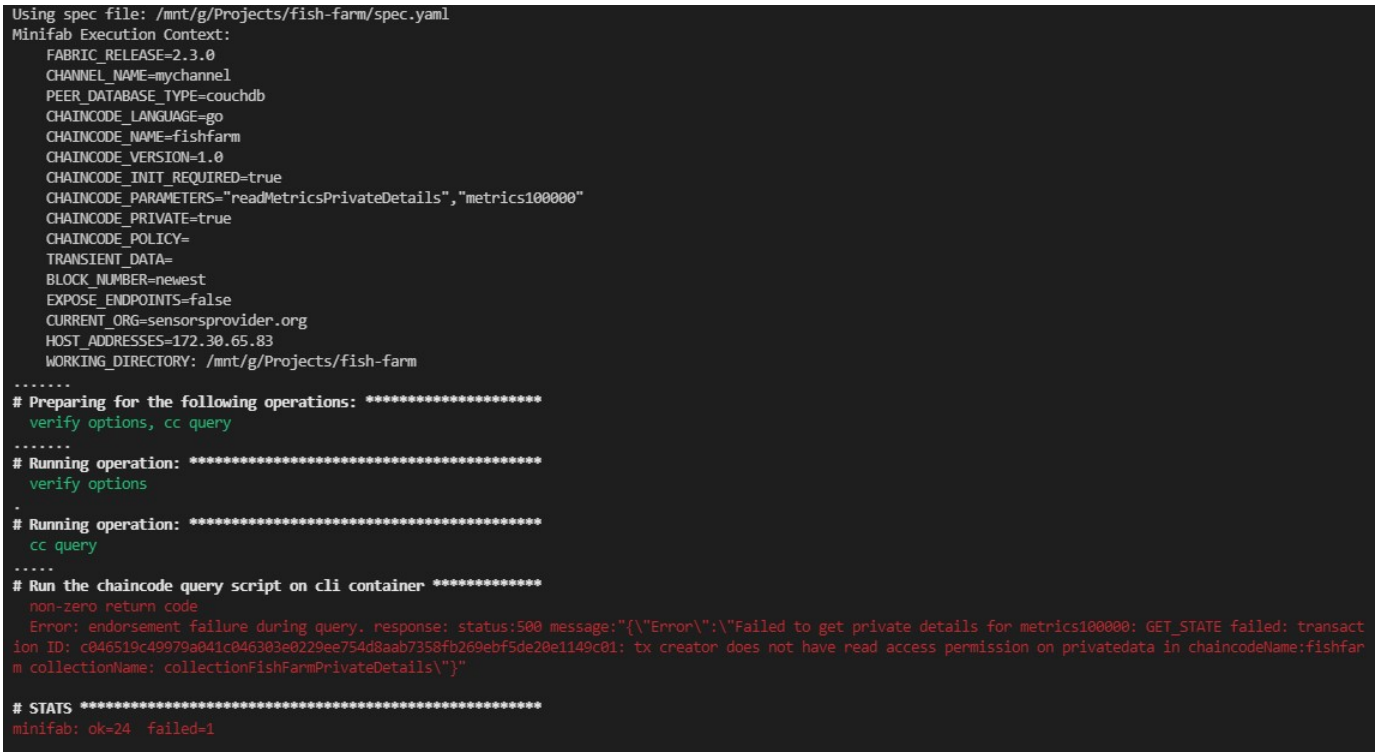


图 8：对私有数据集合的读事务失败，错误为权限被拒绝 or