区块链怎么搭建（教会你构建区块链金融应用详解） 无论你是一个初创公司的创始人，还是一个计划推出金融服务应用的企业领导人，可能你都需要知道如何将区块链用于金融服务。

行业观察家强调了区块链在金融服务市场的稳定增长。Markets and Markets的一份报告指出，金融科技区块链市场将从2017年的2.3亿美元增长到2023年的62.82亿美元。该报告预计，在2018-2023年期间，年复合增长率为75.9%。

在当前的经济趋势发展下，你一定想知道如何构建区块链金融服务应用，请继续阅读本文，我将回答这个问题。

你可以使用哪种区块链：公共与私有

你需要决定使用哪种区块链。这取决于你的商业模式。考虑以下场景：

1. 你的金融产品或服务仅限于注册客户

你是否想将产品或服务限制为在你业注册的客户？你可能已经按照相关的 KYC（了解你的客户）流程来注册客户。你可能会选择只有此类客户才能访问新金融服务应用程序。

如果是这样，那么你需要在这种情况下使用许可的区块链。企业区块链符合你的要求。它们允许你实施访问控制，并且只允许受信任方查看敏感信息。

你可以使用以下任何企业区块链框架/平台：

Hyperledger Fabric：这是来自 Hyperledger Consortium 的开源企业区块链框架。你需要找一个区块链网络主机，比如 AWS、IBM 等。

R3 Corda：这是来自 R3 的企业区块链平台。

ConsenSys Quorum：摩根大通是最初开发 Quorum 的公司。 ConsenSys 收购了它，现在该公司提供 Quorum 作为企业区块链平台。

2. 产品或服务可以向任何人提供

这种情况甚至包括匿名或假名用户。任何人都可以查看交易，并且没有访问控制。如果你想启动一个去中心化的金融服务应用程序，需要一个公共区块链。

许多企业家使用以太坊区块链平台构建了DeFi（去中心化金融）应用程序。以太坊为此提供了以下内容：

EVM（以太坊虚拟机）：这是一台去中心化的计算机。开发者可以使用这个计算引擎来构建 DApps（去中心化应用程序）。

Solidity：它是一种用于编码以太坊智能合约的语言。智能合约是带有“If-Then-Else”语句的开源代码。它们自主执行，并根据预定义的条件转移加密资产。智能合约是不可变的。他们的执行结果是不可逆的。

以太坊区块链开发工具：以太坊生态系统提供IDE（集成开发环境）、测试网（测试网络）、API、部署工具等。我将在稍后讨论它们。

注 1：我在这里提供了一个DApp的快速概述，如下所示：

DApps是网络应用程序。

你可以使用任何标准的前端开发技术来开发他们的前端。

DApp的后端必须由智能合约组成。

DApp运行在像以太坊这样的去中心化区块链网络上。

DApp将数据存储在去中心化的区块链上。他们必须遵守既定的加密标准才能做到这一点。

DApp需要一个加密令牌。你需要使用已建立的加密标准来创建它。

DApp是开源程序。

没有用户社区的共识，你无法更改DApp。

没有一个用户可以控制大多数加密代币。

注意 2：在本指南中，我假设你将在以太坊上构建一个DeFi应用程序。

如何在以太坊网络上构建区块链金融服务应用程序

你需要采取以下步骤：

1.成立软件开发团队并以动态的方式加入

你需要在以太坊区块链开发团队中担任以下角色：

项目经理（PM）、建筑师、业务分析师、用户界面设计师、网络开发人员、以太坊区块链开发者、测试人员。

注 3：我假设你将提供一个网络应用程序。如果你计划提供移动应用程序，则需要聘请移动开发人员。

我推荐以下内容：

聘请在领导区块链开发项目方面具有丰富经验的 PM。

寻找能够将可扩展性措施纳入技术解决方案的架构师。你需要一个计划，以便在时机成熟时扩展你的以太坊区块链应用程序。

聘请对以太坊 DeFi 生态系统有深入了解的业务分析师。

聘请 JavaScript 开发人员来开发 Web 应用程序的前端。许多 Web 开发人员都知道 JavaScript，并且有丰富的框架和库生态系统。

为什么我建议你以错列的方式加入团队？原因如下：

你需要足够的时间来定义区块链开发项目的范围。

计划工作也需要时间。

你应该首先加入 PM、架构师和一些业务分析师。

他们需要定义项目的范围并收集需求。

这个团队还需要计划项目。

在此期间，开发人员将没有足够的工作。区块链开发人员的成本可能很高，因此，当你工作量达到一定程度时，需要雇佣他们。

在进入编码阶段之前加入其他角色。你可以在规划阶段加入开发和测试负责人。

2. 定义项目范围并规划项目

向你的团队指定你的目标。指出你想要构建的 DeFi 类型以及它应该能够做什么。

以下是几种 DeFi 类型：

跨境支付应用程序：这些应用程序促进全球支付交易。

稳定币：这些是加密代币，旨在维持稳定的价格。稳定币的价格通常与美元等现实世界资产挂钩。

去中心化借贷应用程序：这些 DeFi
应用程序让借款人无需任何第三方干预即可直接从贷方借款。

DEX（去中心化交易所）：这些是没有集中管理员的加密交易所。

DeFi 的种类还有很多。

业务分析师需要与业务利益相关者交谈并收集需求。 PM
和架构师需要计划项目，他们需要关注以下内容：

开发工具；

可扩展性解决方案；

项目任务和时间表；

资源加载；

开发、审查、测试；

其他项目管理方面，如沟通、风险管理等。

3. 安装所需的开发和测试工具

现在你可以整合开发团队的其他成员了。该团队需要安装和配置以下工具。

3a. MetaMask

MetaMask是一个加密货币钱包。它还提供了一种简单的方式来连接到基于区块链的应用程序。你可以用它来连接到以太坊区块链网络。

MetaMask可作为浏览器扩展和移动应用程序。你的开发团队需要下载浏览器扩展。

你可以在你的MetaMask钱包中购买和存储以太币（ETH）。为了测试你的以太坊区块链应用，你需要假的以太币。你可以在MetaMask上得到它们。

3b. Ganache

Ganache是一个区块链客户端。你可以用它来开发Ethereum DApp，它是更大的Truffle工具套件的一部分。程序员可以用它来运行测试和执行命令。

你可以使用Ganache UI。它是一个桌面应用程序。你也可以使用Ganache CLI（命令行界面），它早期被称为TestRPC。阅读Ganache文档来使用它。

3c. Web3.js

Web3.js是一个Ethereum JavaScript API。它也是一个重要库的集合。以太坊区块链开发人员可以使用Web3.js与本地或远程以太坊节点进行交互。通过使用Web3.js文档来安装和配置它。

3d. Truffle套件

Truffle套件是一个用于Ethereum区块链开发的工具集合。它包括一个开发环境和测试框架。你可以使用Truffle部署Ethereum智能合约，而且有大量的Truffle文档。

注4：你需要使用更多的工具来开发Ethereum DApp。然而，你不需要安装它们。这些工具如下。

Remix IDE（集成开发环境）：你需要使用Remix IDE来编码Solidity智能合约。它是一个DApp，因此，你不需要安装它。

Ropsten：这是一个用于测试以太坊智能合约的testnet（测试网络）。你需要使用假的以太币来测试Ropsten上的智能合约。

4. 配置工具

你现在需要配置这些工具。

注5：我建议你阅读两篇详细的文章，说明如何配置这些以太坊区块链开发工具。Alex Miller写了第一篇（https://hackernoon.com/getting-started-as-an-ethereum-web-developer-9a2a4ab47baf）。这是一篇关于以太坊DApp开发的指南。我把它称为 "参考文章1"。Merunas Grincalaitis写了第二篇（https://medium.com/ethereum-developers/ultimate-guide-to-convert-a-web-app-to-a-decentralized-app-dapp-f6112a079509），这是一份将网络应用转换为DApps的指南。我们把它称为 "参考文章2"。

做好以下工作：

4a. 配置MetaMask加密钱包和浏览器扩展

你需要采取以下步骤。

创建一个账户，并确保你的密钥。

以 "开发者模式 "连接到Ethereum区块链。

建立与Ropsten testnet的连接。

使用MetaMask指南和 "参考文章2 "获取更多信息。

4b. 配置Ganache、Web3.js和Truffle

采取以下步骤来配置Ganache、Web3.js和Truffle。

阅读Ganache的GitHub文档（https://github.com/trufflesuite/ganache-ui）来配置它。打开一个Ganache实例。

按照Web3.js的 "入门 "指南（https://web3js.readthedocs.io/en/v1.5.2/getting-started.html）来配置它。

阅读Truffle文档（https://trufflesuite.com/docs/truffle/overview）来配置它。

查看 "参考文章2 "（https://medium.com/ethereum-developers/ultimate-guide-to-convert-a-web-app-to-a-decentralized-app-dapp-

f6112a079509）以获得更多指导。

5. 设计和开发DApp的前端

在安装和配置好工具后，你可以开始编码了。首先设计和开发DApp的前端。做好以下工作。

遵循既定的用户界面（UI）设计指南。

使用JavaScript、HTML和CSS来开发用户界面的前端。或者也可以使用前端网络框架，如Angular或React.js。

6. 编码智能合约

现在是对智能合约进行编码的时候了，智能合约是后端的构建块。采取以下步骤。

6a. 回顾相关的智能合约以获得有用的想法

以太坊DApp开发项目是开源的。软件工程师可以审查其他项目的智能合约以获得想法，这是一个优势。

你也可以审查相关项目的智能合约。选择与你的项目相似的DApp项目，例如：

如果你打算建立一个稳定币，就审查Dai稳定币（https://github.com/makerdao/dss）的智能合约。

如果你想一个创建DeFi协议，研究Uniswap DeFi协议的智能合约（https://github.com/Uniswap）。

如果你计划建立一个去中心化的借贷平台，请查看Aave协议的智能合约（https://github.com/aave/protocol-v2）。

6b. 编码智能合约

使用Remix IDE来编码智能合约。我建议你保持代码简单。我这样说是出于以下原因。

成本：一个具有复杂逻辑的智能合约需要更多的计算。如果你的智能合约有复杂的处理逻辑，你会支付更多的 "汽油费"。通过保持逻辑简单，你可以控制你的成本。

维护：调试具有复杂处理逻辑的智能合约更难。你可以通过保持逻辑简单来提高你的智能合约的可维护性。

好好记录你的智能合约，使用Truffle来管理它们。

7. 实施结构化的智能合约审计流程

你需要对你的智能合约进行彻底的审计。记住，你不能在部署智能合约后修改它，这使得测试和代码审查变得加倍重要。

然而，测试不可能找出所有的bug。这使得智能合约的代码审查非常重要。如果你实施一个结构化的代码审查过程，你可以更早发现错误。

在任何项目中，找到有经验的代码审查员都是很难的。以太坊DApp开发项目涉及利基技能，因此，找到专业的审查员可能更难。

有足够的预算。计划好聘请独立智能合约审计师的准备时间。

有经验的智能合约审计师应该关注以下内容：

7a. 进行结构化的智能合约审计

智能合约审计不应该是临时性的。独立审查员应该遵循一个系统的方法。他们需要做到以下几点：

审核人员应该坚持获得源代码的锁定版本。

审查员必须彻底了解项目。

他们需要系统地审查项目文件。

专业的合同审查员应该进行初步的代码审查。

随后应该进行静态代码分析。

审核人员应进行代码质量分析。

审核人员应该寻找已知的智能合约漏洞。

他们应该分析代码是否能提供正确的功能。

审核人员应确定代码优化机会。

智能合约审核人员应跟踪所有问题，并系统地报告它们。

你需要确保关闭智能合约审计中发现的任何问题。

7b. 识别已知的智能合约漏洞

审查人员应留意已知的智能合约漏洞。以下是突出的智能合约漏洞的例子：

重入性：这也被称为
"递归调用漏洞"。它发生在外部智能合约调用可以对调用合约进行新的调用。

访问控制：访问控制问题会影响到所有种类的软件系统。当智能合约实现允许特权角色采取单边行动的功能时，就会出现这种问题。例如，一个有特权的用户可以从合同中提取资金。

算法错误：程序员可能在智能合约中编码无符号整数。这可能会导致 "整数溢出 "和 "整数下溢 "等错误。

智能合约可能不检查低级别调用的返回值。Solidity
允许低级别的函数调用。在执行过程中，这种调用可能会返回一个布尔值
"false"。不对此进行检查的智能合约将继续运行。这可能会产生漏洞。

"拒绝服务"（DoS）：黑客可以审查智能合约，并找到发起DoS攻击的漏洞。以 "for "循环为例。黑客可以给这样的智能合约输入太大的数字。以太坊网络与气体有关的限制将阻止这样的合同迭代这么多次。因此，智能合约将停止运作。黑客可以发动不同类型的DoS攻击，永久阻止智能合约的运作。

编码不正确的函数来生成随机数。黑客可以利用Solidity中的函数和变量来创建随机数。他们可以预测将由智能合约创建的随机数。然后黑客可以操纵DApp。

前端运行：智能合约可能会发布敏感信息。黑客可能会复制它，并提交一个具有较高费用的交易。这将优先考虑交易，黑客将得到操纵系统的机会。

8. 测试智能合约

你现在需要测试你的智能合约。在一开始应创建一个全面的测试计划。测试计划应该有足够的测试案例来覆盖智能合约的所有路径和分支。你需要对测试计划进行结构化审查。

随后，做以下工作：

使用MetaMask购买假的以太币。为此，请访问MetaMask上的测试以太坊龙头。

你已经将MetaMask连接到Ropsten。检查连接是否仍然开放。

导航到Remix IDE。使用相关菜单选项，将你的智能合约部署到Ropsten。

使用MetaMask确认此操作。

按照你的测试计划，测试智能合约。

9. 部署智能合约

现在你已经测试了智能合约，你需要部署它们。做好以下工作：

从Binance等加密货币交易所购买真正的以太币。把它们储存在你的MetaMask钱包里。你需要真实的以太币来在以太坊主网上部署智能合约。

检查Ganache实例，确保它仍在运行。

导航到Truffle。导航到你组织智能合约的Truffle目录。

使用 "truffle deploy "命令将你的智能合约部署到Ethereum主网上。

注意部署后的智能合约地址。

10. 将你的智能合约连接到网络应用程序的前端

你已经对你的网络应用程序的前端进行了编码，现在你需要将智能合约连接到它。请采取以下步骤：

在你选择的IDE中打开Web应用项目。

创建一个空文件，并将其命名为

"web3.min.js"。你将需要这个文件在Web3.js的帮助下将智能合约连接到前端。

访问GitHub上的Web3.js ChainSafe资源库。复制 "web3 min代码"。

将整个代码粘贴到你在web应用项目中创建的web3.min.js文件中。

保存web3.min.js文件。

把它导入到你的Web应用项目的主HTML文件中。

11. 初始化你的智能合约

你现在需要初始化你已经部署的智能合约。这涉及到以下步骤：

11a. 获取合同ABI（应用二进制接口）

以太坊使用称为ABI的数据编码方案来与智能合约进行通信。做好以下工作，以获得ABI：

导航到Remix IDE的 "编译
"选项卡。导航到ABI部分，在一个文本文件中记下ABI信息。

你需要对ABI信息进行清理。它是JSON格式的，而且有空格。使用JSON minify来删除空格。为此，将ABI信息粘贴在JSON minify面板上。随后，点击 "JSON minify "按钮。

复制输出的数据。

把它粘贴到你的代码中的 "const contractABI "变量中。

11b. 获取合同地址

导航到Remix IDE的 "Run "标签。获取合同地址，并将其粘贴到代码中一个名为 "const contractAddress "的变量中。

11c. 更新网络应用的不同功能以使用智能合约

你已经编码了你的智能合约以满足你的DApp中的不同功能。你用JavaScript编码了前端，现在你需要用相应的智能合约更新这些功能。

做好以下工作：

为每个智能合约创建一个合约实例。

用相应的 "const contract ABI "和 "const contractAddress "变量更新它。

打开一个函数的JavaScript文件。

用该函数的合同实例代码更新它。

对所有的函数重复这一步骤。

总结

我已经介绍了用于构建金融服务应用程序的不同种类的区块链网络，你可以根据需要选择最合适的一种，还一并介绍了如何使用以太坊区块链平台构建一个去中心化的金融服务应用程序，以及如何去构建你自己的基于区块链的金融服务应用程序。祝你的项目取得好成绩。