区块链开发怎么做（最好用的区块链开发框架分享） 智能合约开发框架让工程师可以部署和测试他们的智能合约，简化他们的开发工作。我花了很多天时间来了解 ConsenSys 和 Ethereum 网站上推荐的每一个框架，想看看哪些框架是 2021 年最好用的。你可以在 GitHub 上看到我对这些平台的实验。

不仅如此，我还对区块链黑客马拉松参与者做了几百次采访，并查阅了几百个黑客马拉松提交，想要找出哪些工具是性能最好、开发体验最出色的。我自己也使用这些框架做了很多工作，想为 2021 年做好准备。我为大家做好了这期 2021 年优秀框架评选，这样你就不用陷入选择困难症了。

本文采用的标准很简单。评选的框架需要：

部署到一个本地区块链

进行快速有效的测试

使用 Chainlink 将一个智能合约部署到KovanTestnet

第三个标准很重要，因为它让我得以测试框架如何与其他软件包交互和集成测试交互。从 2021 年开始，搭配oracles和 OpenZeppelin 等第三方工具已经成了一项基本需求，所以最后一项标准就非常重要了。

我们将简要介绍其他一些 2021 年你需要了解的重要工具，但本文不会涉及初学者工具，例如 ETH 钱包、以太坊客户端或文本编辑器。

视频简介：https://youtu.be/motYF5daaOc

Top 4 智能合约开发框架

总的来说，我将向所有人推荐三个智能合约开发框架。每种框架的风味都稍有不同，选择它们的理由也略有差异。它们分别是：

Hardhat（JavaScript）

Truffle（JavaScript）

Brownie（Python）

我们来深入研究一下这几个框架，谈谈你为什么要使用它们，为什么它们排在了名单的前列，以及其他值得注意的地方。

Hardhat

Hardhat 是目前最好的框架之一，有着同类竞品中最快的测试、最好的教程和最简单的集成。老实说，喜欢 JS 框架的人们迟早都应该试一试 Hardhat。它真的很容易上手，测试飞快而且入门非常简单。他们的 Discord 也一直在非常迅速地回答各种问题，因此如果你遇到了问题，总是很容易寻求帮助。他们使用 Waffle 和 Ethers.js 进行测试，这可以说是更好用的 JavaScript 智能合约框架，因为它们相比 web3.js 做了一些非常不错的改进。他们还可以与 OpenZeppelin 的可升级智能合约插件直接集成，这是一个很大的加分点。

这个项目给人一种很棒的感觉：它很干净。它能执行你想要的操作。它真的很快。这个项目正在不断改进，显然开发团队正在努力让智能合约开发人员的生活更加轻松。

小结

ETH 基金会资助的项目，以前是 Builder

技术：Javascript、Web3.js 和 Ethers.js 插件，OpenZeppelin 可升级合约插件，Etherscan 插件，区块链分叉

区块链：Hardhat 运行时环境/本地，测试网，主网

有测试：Waffle

维护：非常活跃

支持：活跃

开源

Truffle

Truffle 几年来一直是开发人员的首选框架，这是有充分理由的。它是一个强大的框架，为其他跟随者树立了标准。你会很容易找到一大堆项目都在使用这个平台，因此示例查找起来很容易。Truffle 也可以很容易地与它的姊妹工具Drizzle 集成在一起。特别是 Ganache，它是工程师运行本地区块链的最流行方法之一。如果你需要更多工具，你可以花钱购买升级版的 Truffle 团队帐户，获得智能合约的持续集成、可视化部署和监视功能。他们还可以与 OpenZeppelin 的可升级智能合约插件直接集成，这是一个很大的优势。他们显然是一群有才华的工程师，都想要为智能合约创造一个更美好的世界。

Truffle 测试的运行速度不如 Hardhat 那样快，并且由于前者的用户数量众多，支持获取起来可能会很困难。自从 Truffle 被 ConsenSys 收购以来，我很期待看到他们将如何改善这个项目。他们的文档质量似乎开始下降，读起来可能会磕磕绊绊，但是如果你用谷歌搜索一下自己遇到的错误，很可能会找到曾经遇到这个错误并已解决它的例子。我发现改善项目的最佳方法之一就是在他们的 GitHub 上留一个问题。无论如何，让生态系统不断壮大是我们开源社区的职责！

因为几乎所有人都很熟悉它，因此获得同行支持通常会很容易。我真的希望看到开发团队在这个项目上获得更多支持，毕竟他们有这么多的用户。我希望他们能看到本文并能努力改善项目文档，让它能继续作为测试和部署智能合约的首选平台之一。

小结

使用最广泛的平台；最近被 ConsenSys 收购（2020 年 11 月）

技术：Javascript，Web3.js，OpenZeppelin 可升级合约插件，Etherscan 插件，区块链分叉

区块链：Ganache/本地，测试网，主网

有测试

维护：非常活跃

支持：活跃

具有付费升级功能的开源项目

Brownie

Brownie 很大程度上受到了 Truffle 的启发，前者基本上是后者的 Python 等效版本，并增加了一些体验改进。这个项目起源于大多数 Python 工程师都同样拥有的困惑：他们不喜欢 JavaScript。由于它是用 Python 编写的，因此避免了使用 JavaScript 带来的许多麻烦，并为大家带来了一个简单优秀的框架来处理和部署智能合约。

我自己就很喜欢 Python，因此对这个项目的未来感到非常兴奋，随着采用率的提高，越来越多的项目正在使用 Python 和这个框架来部署他们的智能合约。我已经将 Brownie 用作了我的入门框架，用下来感觉非常不错。我还同一些用它做更多事情的开发人员做过交流，他们真的非常优秀。Yearn.finance 似乎也已经用上了这个框架。

如果你来自 Python 开发世界，或者想要一个非常简单、干净的框架，那么这一款就是为你而生的，毫无疑问。

还有一个框架！我知道我只提到了三个，但我认为下面的框架也应该在这里占据一席之地。

小结

最好的 Python 框架，由 Yearn.finance 等 DeFi 项目使用。

技术：Python，Web3.py，区块链分叉，Etherscan 插件，Vyper 支持

区块链：Ganache/本地，测试网，主网

有测试

维护：非常活跃

支持：活跃

开源

Embark

我很惊讶能排名前三以外的框架能在 GitHub 上有这么多星（3.1k）。这款工具花样很多。我花了一些工夫才搞清楚它到底是怎么回事，但这是因为 Embark 旨在成为一个完整的 dApp 框架。这是一个全栈的区块链平台。在他们 Gitter 的一些帮助下（以及在他们 GitHub 上创建的一个问题；请记住，开源社区责任！），我得以将自己的 Chainlink 合约部署到 Kovan 网络。它带有一个 UI，这样你就可以在 GUI 中与区块链和合约交互。看来它有一段学习曲线，我没有花足够的时间来克服障碍，但能看出它的潜力很大。这就是为什么我想把它放在这里的原因所在，因为我觉得自己还有很多东西没能完全消化。

我希望看到有更多人能尝试这个框架并意识到它的能力。由于我在该项目上花费的时间有限，所以我觉得自己可能无法做到公正评价。我确实认为，将前端与后端解耦仍然是最佳实践，但是如果你需要开始一个具有良好前端的项目并且不关心解耦需求，那么你绝对应该尝试这个项目。

这个项目很酷，如果有人喜欢 Hardhat 和 Truffle，但又想集成一个全栈解决方案，我会首先推荐它。

小结

用于前端开发的，具有大量功能的 JavaScript 框架

技术：JavaScript，Web3.js，代理合约支持

区块链：Ganache/本地，测试网，主网

有测试

维护：轻度活跃

支持：活跃

开源

我们暂停一下

继续介绍上述框架的替代选项之前，我想暂停一下，先来介绍一些 2021 年重要的非框架工具。

OpenZeppelin 和升级插件（https://openzeppelin.com/contracts/）

Etherscan/BlockExplorer 验证器（https://etherscan.io/）

create-eth-app（https://github.com/paulrberg/create-eth-app）

OpenZeppelin 和升级插件

OpenZeppelin 是一个平台，可为智能合约工程师创建有用的工具。他们有一些默认选项，例如 ERC20 和 ERC721 工具，你可以在其中导入他们的合约并在几秒钟内建立一个令牌。看看用 OpenZeppelin 制作一个 ERC20 有多简单：

pragmasolidity^0.6.0;import“@openzeppelin/contracts/token/ERC20/ERC20.sol”;contractGLDTokenisERC20{constructor(uint256initialSupply)publicERC20(“Gold”,“GLD”){_mint(msg.sender,initialSupply);}}复制代码

重新发明轮子没有意义。OpenZeppelin 是一个开源项目，主要目的就是省去大家重新造轮子的功夫。OpenZeppelin 的另

一个重要特性是它的可升级合约功能。部署一个实际项目意味着你必须部署一个代理合约，这个代理合约在你要进行任何更改时都需要更新。这又是一个体验出色且干净的项目。老实说，对每一个项目来说，无论你使用的是什么平台，迟早都会用上 OpenZeppelin 创建的内容。

Etherscan/Block Explorer 验证器

Etherscan 是最受欢迎的区块浏览器。当人们想要签出你的合约以及你已部署的内容时，除非他们具有 ABI
和合约地址，否则就无法签出。找到合约地址通常很容易，但是要获得 ABI 却很困难。你肯定希望人们能轻松地查看你的智能合约并与之互动，因此你需要随时对其进行验证，以便其他人可以做到这一点。大多数框架都有某种验证插件，你可以在其中部署智能合约，然后立即在 Etherscan
上对其进行验证。一定要在你的应用程序中使用它。

create-eth-app

拥有一大堆后端智能合约诚然很不错，但是如果没有人可以使用你的
dApp，那这些合约将毫无价值。create-eth-app 是一个平台，允许你使用
ReactJS 为你的智能合约增加一个前端。它使用了许多最新的 React
技术（例如状态 Hooks），并让开发人员可以很容易地构建真正强大的前端。

2021 年的一些出色的初学者工具

Scaffold- ETH（https://github.com/austintgriffith/scaffold-eth）

eth.build（https://eth.build/）

Remix（https://remix.ethereum.org/）

OneClick dApp（https://oneclickdapp.com/）

我不会在这里介绍它们的详细情况，但如果你是这个领域的新手，用它们就对了！

其他框架选项

剩下的这些框架也都很不错，并且可能更适合你的需求。它们中的一些确实对智能

合约框架中应该放什么内容有一些非常酷的想法，我希望前三名选手能够注意到其中一些很棒的理念。这些工具可能需要更多的探索时间。

Waffle（JS）

Dapp.Tools（Haskell/CLI）

SBT（Scala/CLI）

Epirus（Java）

Waffle

Waffle本身可以用作部署框架，只不过你需要编写更多自定义脚本。虽说拿它和其他框架进行对比有点愚蠢，但这里还是把它列为智能合约框架。Waffle
可以与你喜欢的任何框架搭配使用，并且当前是搭配 Hardhat 的默认工具。

如果你希望框架可以提供所有不错的功能，那我不建议单独使用它。但如果你想使用更原始、更简单、可自定义的框架，那么选它就不会出错。

小结

轻量级 JavaScript 框架，专注于测试

技术：JavaScript，Ethers.js

区块链：你所经营的一切

有测试

维护：非常活跃

支持：活跃

开源

Dapp.Tools

MakerDAO 使用了Dapp.Tools，它也惊艳了我一把。它是用 Haskell 编写的，但支持许多我们喜欢和常用的插件，例如 OpenZeppelin。它看起来像一个极简主义的框架，但功能强大到足以做到你想做的事情。我可以找到关于使用这个平台部署合约的帮助，另外它的简洁也给人留下了深刻印象。如果你正在寻找一个对命令行友好的极简主义框架，那么我建议你也看一下这一款。

小结

MakerDAO 使用的 Haskell 简单框架

技术：Haskell/CLI

区块链：本地/Ganache，测试网，主网

有测试

维护：活跃

支持：活跃

开源

Epirus

虽然我说过它是“类 Truffle”框架，但Epirus的风格也不是完全像 Truffle，只是在商业层面比较相似。他们有一个开源项目，但还有一个付费的升级工具，供想要用它做更多事情的用户选择。他们使用 Java 的 Web3 实现，而且看起来仍处于项目开发的早期阶段。我很高兴看到他们提供了对 Android 的支持，因此我很想知道这个项目的进展情况。由于 Java 智能合约平台不多，因此它可以吸引更多的 Java 工程师进入这一行业。

我无法用它部署到 Kovan，因为它不受支持，但看起来他们已经支持Rinkeby了。无论如何，我很高兴看到这个项目能顺利推进。

小结

类似于 Truffle 的 Java 实现

技术：Java

区块链：Ganache/本地，测试网（不是 Kovan）

有测试

维护：活跃

支持：不是很活跃

具有付费升级功能的开源项目

SBT

这是我玩过的一个很有趣的项目，SBT的特色是几乎完全从命令行或使用 Scala 与智能合约交互。我的 Scala
经验几乎为零，因此没有足够的背景知识来评估这个平台。如果你了解 Scala
并想要踏入以太坊世界，这款工具可能真的是很酷的选项。

小结

同时满足命令行与 Scala 的开发需求。适用于智能合约爱好者和工程师。

技术：Scala

区块链：本地/Ganache，测试网，主网

有测试

维护：轻度活跃

支持：不活跃

开源

过时的框架

你经常会见到这些框架的身影。我敢肯定它们问世时都是很棒的工具，但是到了今天，它们不再获得支持或处于非活跃状态。下面的这些框架不值得你浪费精力去了解和尝试。

OpenZeppelinCLI/SDK（JS）

Etherlime（JS）

Parasol（JS）

Cobra（PY）

OpenZeppelin SDK

现在，我明白为什么这个项目停止为此工作了——作为一个团队，他们希望专注于可升级智能合约这一棘手的问题——但说实话这挺遗憾的。这个项目相当干净。我可以部署我的 Chainlinked 智能合约，并能比几乎其他所有框架更快地读取 ETH 的价格。它还有一些非常不错的 CLI 选项，就像 Epirus 和 Hardhat 一样。它还内置了可升级的选项，其他平台很少有这种设计，所以这是很大的优点。

我不建议使用这个框架，因为他们不会再支持它了。但这是我使用智能合约框架获得的最佳开发体验之一。我希望前三名选手能够从中汲取一些令这个项目与众不同的灵感。我上面说过，每个人都应该使用 OpenZeppelin
工具是有原因的。他们就是很好用。

如果他们不停止这个项目，那么他们的 SDK 很容易成为前四名的框架。

小结

OpenZeppelin 弃用的智能合约框架。他们弃用它是为了将重点放在升级插件上。不管怎样，它真的是很干净的 JS 框架。

技术：JavaScript，Web3.js 和 Ethers.js 插件，OpenZeppelin 可升级合约插件

区块链：本地，测试网，主网

有测试

维护：不活跃

支持：不活跃

开源

Etherlime

Etherlime 似乎是另一个不受青睐且未积极开展工作的 JS 框架。它与 Infura 的一项过时特性紧密耦合，因此我们无法在 Kovan 上部署智能合约。它活跃的时候看来是一个非常强大的项目，而且似乎仍然可以在本地开发中使用，但是我不建议用它替代前三个选项。

小结

不活跃的 JS 框架

技术：JavaScript，Ethers.js

区块链：本地，测试网，主网

有测试

维护：不活跃

支持：不活跃

开源

Parasol

另一个不活跃的 JS 框架（超过两年没有一项提交）。它有一些很酷的功能，例如 parasol interact 和一些集成，而且显然曾有一些很聪明的人在从事这项工作，但

是我认为没有理由用它代替其他任何 JS 框架了。

小结

不活跃的 JS 框架

技术：JavaScript，Web3.js

区块链：本地/Ganache，测试网，主网

有测试

维护：不活跃

支持：不活跃

开源

如果你觉得我错过了什么框架或在哪里弄错了，或者你有其他想法，请随时发表评论或联系我。本文的想法是让智能合约工程师能够尽快高效地入门。掌握所有这些工具将使我们能够更快地实现去中心化的世界。希望这对你能有所帮助！