CONSENSYS

CONSENSYS GUIDES

Developer Tools

A guide to the available tools, components, patterns, and platforms for developing applications on Ethereum.

CONTENTS

Introduction

A guide to the available tools, components, patterns, and platforms for developing applications on Ethereum.

The creation of this list was spurred by product managers at ConsenSys who saw a need for better sharing of tools, development patterns, and components amongst both new and experienced blockchain developers. If you're ready to delve into the Ethereum wormhole, get scrolling!

This resource is meant to be focused on developer tools, but the list also includes other Ethereum ecosystem resources, such as dapps, wallets, and other services.

ConsenSys does not guarantee that any or all of the following tools are free of errors, bugs, or potential vulnerabilities. However, most of these tools are used by thousands of blockchain developers and are considered some of the best in the business. Feel free to scroll through the guide and find the tools that you need or want to try. Each tool links to the website or documentation so that you can learn more!

NEW DEVELOPERS

If you're saying to yourself "help me, I'm a new developer and this list is paralyzing me with fear". First, hello friend, take a deep breath and don't worry.
Second, start by reading the sections, and test out some of the foundational Ethereum infrastructure tools such as Truffle, Infura, MetaMask, IPFS, and read through some of the educational resources. If you're from the enterprise world, then check out Kaleido, Microsoft, and PegaSys to see which tools may be best for your organization.

Educational Resources

EthHub – Comprehensive crowdsourced overview of Ethereum, its history, governance, future plans, and development resources.

Kauri –Learn to build on Ethereum by reading all the latest articles, tutorials, documentation and best practices.

ConsenSys Academy – Academy provides an end-to-end Ethereum developer course that is self-paced and open year-round.

Mastering Ethereum – Mastering Ethereum is a book for developers, offering a guide to the operation and use of the Ethereum, Ethereum Classic, RootStock (RSK) and other compatible EVM-based open blockchains.

Smart Contracts

SMART CONTRACT LANGUAGES

If you're developing on Ethereum, you're going to need to know how to write smart contracts. Luckily there are not too many different languages that you need to know. Solidity the main smart contract language is easier to learn if you already possess a good understanding of Javascript.

Solidity – Ethereum smart contracting language

Bamboo – A morphing smart contract language

Vyper – New experimental pythonic programming language

Flint – New language under development with security features including asset types, state transition, and safe integers

SMART CONTRACT FRAMEWORKS AND TOOLS

Truffle – Most popular smart contract development, testing, and deployment framework. The Truffle suite includes Truffle, Ganache, and Drizzle. Deep dive on Truffle here

Embark – Framework for DApp development

Waffle – Framework for advanced smart contract development and testing,

> *Telling a programmer there's already a library to do X is like telling a songwriter there's already a song about love."*
>
> **Pete Cordell**

small, flexible, fast (based on ethers.js)

Dapp – Framework for DApp development, the successor to DApple

Etherlime – ethers.js based framework for Dapp deployment

Parasol – Agile smart contract development environment with testing, INFURA deployment, automatic contract documentation and more. It features a flexible and unopinionated design with unlimited customizability

0xcert – JavaScript framework for building decentralized applications

OpenZeppelin SDK – OpenZeppelin SDK: A suite of tools to help you develop, compile, upgrade, deploy and interact with smart contracts.

sbt-ethereum – A tab-completey, text-based console for smart-contract interaction and development, including wallet and ABI management, ENS support, and advanced Scala integration.

Brownie – Brownie is a Python framework for deploying, testing and interacting with Ethereum smart contracts.

Cobra – A fast, flexible and simple development environment framework for Ethereum smart contract, testing and deployment on Ethereum virtual machine(EVM).

SMART CONTRACT LIBRARIES

You've probably used programming libraries before, and these are no different. A smart contract library is the reusable piece of code for a smart contract that is deployed once and shared many times. Below are the most used smart contract libraries.

Zeppelin – Contains tested reusable smart contracts like SafeMath and ZeppelinOS library for smart contract upgradeability

cryptofin–solidity – A collection of Solidity libraries for building secure and gas–efficient smart contracts on Ethereum.

Modular Libraries – A group of packages built for use on g the Ethereum Virtual Machine

DateTime Library – A gas–efficient Solidity date and time library

Aragon – DAO protocol. Contains aragonOS smart contract framework with a focus on upgradeability and governance

ARC – an operating system for DAOs and the base layer of the DAO stack.

0x – DEX protocol

Token Libraries with Proofs – Contains correctness proofs of token contracts wrt. given specifications and high–level properties

Provable API – Provides contracts for using the Provable service, allowing for off-chain actions, data–fetching, and computation

SMART CONTRACT MONITORING

Whether you're building a personal application or an application for the masses you'll need to be able to monitor your smart contract. Use the following monitoring tools inspect and analyze how your smart contracts are performing or get notifications to adjust gas fees or when certain events occur. Some of these tools even provide visualizations so you can see exactly how your smart contract is functioning.

Alethio – Actively monitor any external account, dapp or smart contract.

amberdata.io – Provides live monitoring, insights and anomaly detection, token metrics, smart contract audits, graph visualization and blockchain search.

Neufund – Smart Contract Watch – A tool to monitor a number of smart contracts and transactions

Scout – A live data feed of the activities and event logs of your smart contracts on Ethereum

Tenderly – A platform that gives users reliable smart contract monitoring and alerting in the form of a web dashboard without requiring users to host or maintain infrastructure

Chainlyt – Explore smart contracts with decoded transaction data, see how the contract is used and search transactions with specific function calls

BlockScout – A tool for inspecting and analyzing EVM based blockchains. The only full featured blockchain explorer for Ethereum networks.

Testing Tools

Test our your solidity code and smart contracts using the following tools! Testing tools can execute tests automatically while freeing up developer time and system resources.

Truffle Teams – Zero-Config continuous integration for truffle projects.

Solidity code coverage – Solidity code coverage tool

Solidity coverage - Alternative code coverage for Solidity smart-contracts

Solidity function profiler - Solidity contract function profiler

Sol-profiler – Alternative and updated Solidity smart contract profiler

Espresso – Speedy, parallelized, hot-reloading solidity test framework

Eth tester – Tool suite for testing Ethereum applications

Cliquebait – Simplifies integration and accepting testing of smart contract applications with docker instances that closely resembles a real blockchain network

Hevm – The hevm project is an implementation of the Ethereum virtual machine (EVM) made specifically for unit testing and debugging smart contracts

Ethereum graph debugger – Solidity graphical debugger

Tenderly CLI – Speed up your development with human readable stack traces

Solhint – Solidity linter that provides security, style guide and best practice rules for smart contract validation

Ethlint – Linter to identify and fix style & security issues in Solidity, formerly Solium

Decode – npm package which parses tx's submitted to a local testrpc node to make them more readable and easier to understand

truffle-assertions – An npm package with additional assertions and utilities used in testing Solidity smart contracts with truffle. Most importantly, it adds the ability to assert whether specific events have (not) been emitted.

"Security is a state of mind."

NSA Security Manual

Psol – Solidity lexical preprocessor with mustache.js-style syntax, macros, conditional compilation, and automatic remote dependency inclusion.

solpp – Solidity preprocessor and flattener with a comprehensive directive and expression language, high precision math, and many useful helper functions.

Decode and Publish – Decode and publish raw ethereum tx. Similar to https://live.blockcypher.com/btc-testnet/decodetx/

Doppelgänger – a library for mocking smart contract dependencies during unit testing.

rocketh – A simple lib to test ethereum smart contract that allow to use whatever web3 lib and test runner you choose.

pytest-cobra – PyTest plugin for testing smart contracts for Ethereum blockchain.

Security Tools

Ok, so you've finally built your dapp or smart contract. But how do you know it was set up correctly and is safe from hackers? The security tools below will help ensure that your code is safe and follows all Ethereum development best practices.

MythX – Security verification platform and tools ecosystem for Ethereum developers

Mythril Classic – Open–source EVM bytecode security analysis tool

Oyente – Alternative static smart contract security analysis

Securify – Security scanner for Ethereum smart contracts

SmartCheck – Static smart contract security analyzer

Porosity – Decompiler and Security Analysis tool for Blockchain-based Ethereum Smart–Contracts

Ethersplay – EVM disassembler

Evmdis – Alternative EVM disassembler

Hydra – Framework for cryptoeconomic contract security, decentralized security bounties

Solgraph – Visualise Solidity control flow for smart contract security analysis

Manticore – Symbolic execution tool on Smart Contracts and Binaries

Slither – A Solidity static analysis framework

Adelaide – The SECBIT static analysis extension to Solidity compiler

solc-verify – A modular verifier for Solidity smart contracts

Solidity security blog – Comprehensive list of known attack vectors and common anti-patterns

Awesome Buggy ERC20 Tokens – A Collection of Vulnerabilities in ERC20 Smart Contracts With Tokens Affected

Free Smart Contract Security Audit – Free smart contract security audits from Callisto Network

Ethereum Clients

An Ethereum client refers to any node that is able to parse and verify the blockchain, its smart contracts, and everything in between. An Ethereum client also provides interfaces to create transactions and mine blocks which is the key for any Ethereum transaction. Below are the most popular Ethereum clients.

Infura – A managed service providing Ethereum client standards-compliantt APIs

Hyperledger Besu – Java client by PegaSys

Geth – Go client

Parity – Rust client

Aleth – C++ client

Nethermind – .NET Core client

Pyethapp – Python client using pyethereum

Trinity – Python client using py-evm

Ethereumjs – JS client using ethereumjs-vm

Ethereumj --Java client by the Ethereum Foundation

"Good programming is 99% sweat and 1% coffee."

anonymous

Harmony –Java client by EtherCamp

Seth – Seth is an Ethereum client tool—like a "MetaMask for the command line"

Mustekala – Ethereum Light Client project of Metamask.

Exthereum – Elixir client

EWF Parity – Energy Web Foundation client for the Tobalaba test network

Quorum – A permissioned implementation of Ethereum supporting data privacy by JP Morgan

Mana – Ethereum full node implementation written in Elixir.

IDE's/Editors

IDE stands for Integrated Development Environment. IDEs and Editors are what you need to write and test software. They are software suites that consolidate basic tools that are required to start writing on Ethereum. Below are the most popular IDEs and Editors.

Remix – Web IDE with built-in static analysis, test blockchain VM.

Superblocks Lab – Web IDE. Built in browser blockchain VM, Metamask integration (one-click deployments to Testnet/Mainnet), transaction logger and live code your WebApp among many other features.

Atom – Atom editor with Atom Solidity Linter, Etheratom, autocomplete-solidity, and language-solidity packages

Pragma – Very simple web IDE for solidity, and auto-generated interfaces for smart contracts.

Vim solidity – Vim syntax file for solidity

Visual Studio Code – Visual Studio Code extension that adds support for Solidity

Intellij Solidity Plugin – Open–source plug-in for JetBrains IntelliJ Idea IDE (free/commercial) with syntax highlighting, formatting, code completion etc.

YAKINDU Solidity Tools – Eclipse-based IDE. Features context-sensitive code completion and help, code navigation, syntax coloring, build in compiler, quick fixes and templates.

Eth Fiddle – IDE developed by The Loom Network that allows you to write, compile and debug your smart contract. Easy to share and find code snippets.

Interfaces

FRONT-END INTERFACES

If you want to start developing dapps, you'll need front-end development skills. Below are the most popular front-end interfaces that will help you turn your dapp from an idea to a live Ethereum mainnet application.

Web3.js – Javascript Web3

Eth.js – Javascript Web3 alternative

Ethers.js – Javascript Web3 alternative, useful utilities and wallet features

light.js A high–level reactive JS library optimized for light clients.

Web3Wrapper – Typescript Web3 alternative

Ethereumjs – A collection of utility functions for Ethereum like ethereumjs-util and ethereumjs-tx

flex-contract and flex–ether – Modern, zero-configuration, high–level libraries for interacting with smart contracts and making transactions.

ez-ens – Simple, zero-configuration Ethereum Name Service address resolver.

web3x – A TypeScript port of web3.js. Benefits includes tiny builds and full type safety, including when interacting with contracts.

Nethereum – Cross-platform Ethereum development framework

Drizzle – Redux library to connect a frontend to a blockchain

> *Bad programmers worry about the code. Good programmers worry about data structures and their relationships."*
>
> **Linus Torvalds**

Tasit SDK – A JavaScript SDK for making native mobile Ethereum dapps using React Native

Subproviders – Several useful sub providers to use in conjunction with Web3-provider-engine (including a LedgerSubprovider for adding Ledger hardware wallet support to your dApp)

web3-react – React framework for building single-page Ethereum dApps

Vortex – A Dapp-ready Redux Store. Smart and Dynamic background data refresh thanks to WebSockets. Works with Truffle and Embark.

Strictly Typed – Javascript alternatives

elm-ethereum

purescript-web3

ChainAbstractionLayer – Communicate with different blockchains (including Ethereum) using a single interface.

Delphereum – a Delphi interface to the Ethereum blockchain that allows for the development of native dApps for Windows, macOS, iOS, and Android.

BACK-END INTERFACES

If you want to graduate from just building dapps, you'll need to start learning and using the backend interfaces listed below. If you're interested in doing backend/protocol work on Ethereum, you should have significant experience with Go, Rust, Java, .NET, Ruby, or Python. Explore some of the most frequently used backend interfaces below.

Web3.py – Python Web3

Web3.php – PHP Web3

Ethereum-php – PHP Web3

Web3j – Java Web3

Nethereum – .Net Web3

Ethereum.rb – Ruby Web3

Web3.hs – Haskell Web3

KEthereum – Kotlin Web3

Pyethereum – The Python core library of the Ethereum project

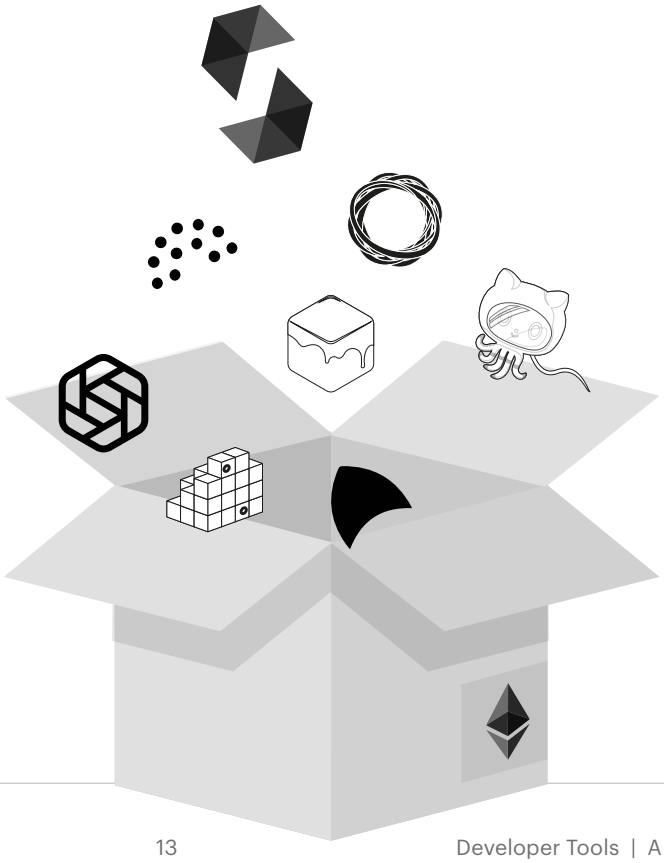
Eventeum – A bridge between Ethereum smart contract events and backend microservices, written in Java by Kauri

Ethereumex – Elixir JSON–RPC client for the Ethereum blockchain

EthContract – A set of helper methods to help query ETH smart contracts in Elixir

Ethereum Contract Service – A MESG Service to interact with any Ethereum contract based on its address and ABI.

Ethereum Service – A MESG Service to interact with events from Ethereum and interact with it.



Ethereum ABI (Application Binary Interface) tools

An ABI is an interface between two program modules, one of which is often at the level of machine code. The ABI interface is the primary method for encoding/decoding data into/out of the machine code. For instance, you'll use an ABI to encode Solidity contract calls for the EVM (Ethereum Virtual Machine) and, vice versa, how to read the data out of transactions.

ABI decoder – library for decoding data params and events from Ethereum transactions

ABI-gen – Generate Typescript contract wrappers from contract ABI's.

Ethereum ABI UI – Auto-generate UI form field definitions and associated validators from an Ethereum contract ABI

headlong – type-safe Contract ABI and Recursive Length Prefix library in Java

One Click dApp – Instantly create a dApp at a unique URL using the ABI.

Ethereum Contract Service – A MESG Service to interact with any Ethereum contract based on its address and ABI.

Messaging

Communication protocols for dapps and nodes to communicate with each other. Healthy communication is important in life and in your dapp!

Whisper – Communication protocol for DApps to communicate with each other, a native base layer service of the Ethereum web3 stack

DEVp2p Wire Protocol – Peer-to-peer communications between nodes running Ethereum/Whisper

Pydevp2p – Python implementation of the RLPx network layer

Storage

Ethereum allows you to save variables or data in permanent storage. The storage platforms below are where all of the smart contract data lives. IPFS is the most commonly used storage system on Ethereum. Explore the platforms below to learn more about how storage on Ethereum works.

IPFS – decentralized storage and file referencing

OrbitDB – decentralized database on top of IPFS

Swarm – Distributed storage platform and content distribution service, a native base layer service of the Ethereum web3 stack

Infura – A managed IPFS API Gateway and pinning service

3Box – A suite of APIs to easily build distributed consumer applications (profile, messaging, storage).

Testnets

PUBLIC TESTNETS

Public Testnets on Ethereum offer a way for developers to test what they build without putting their creations on the main Ethereum network. Developers are able to obtain as much ETH as you want on testnets because testnet ETH doesn't carry any monetary value. Below are the most used testnets to start testing on and the links for where you can request testnet ETH.

Ropsten – A proof-of-work blockchain that most closely resembles Ethereum and allows you to easily mine faux–Ether.

Görli – Proof-of-authority cross-client testnet, synching Parity Ethereum, Geth, Nethermind, Pantheon, and EthereumJS. This testnet is a community–based project, completely open-source.

Kovan – A proof-of-authority blockchain started by the Parity team. Test ether must be requested.

Rinkeby – A proof-of-authority blockchain started by the Geth team. Test ether must be requested.

TESTNET FAUCETS

Use these faucets to obtain testnet ether.

Rinkeby faucet

Kovan faucet

Ropsten faucet

Goerli faucet

LOCAL TESTNETS

Similar to Public Testnets, Local Testnets are a place for you to test your software without pushing it public. Unlike Public Testnets, the Local Testnet software will only run on your computer/node and other users won't be able to see it or interact with it.

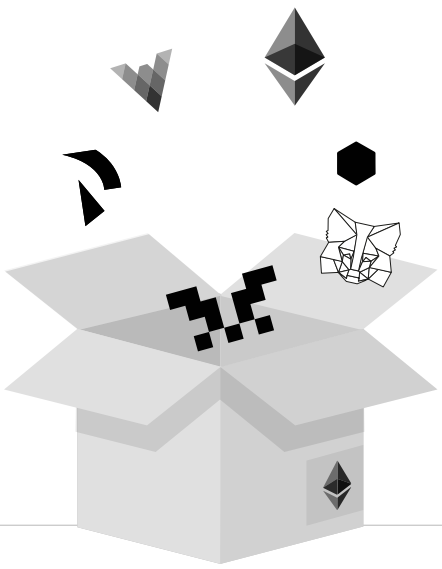
Local Raiden – Run a local Raiden network in docker containers for demo and testing purposes

Ganache – App for test Ethereum blockchain with visual UI and logs

Ganache CLI – Fast Ethereum RPC client for testing and development. The command-line version of Ganache, your personal blockchain for Ethereum development.

Local Raiden – Run a local Raiden network in docker containers for demo and testing purposes

Local Ethereum Network – Out-of-the-box deployment scripts for private PoW networks



Block Explorers

A block explorer is an online blockchain browser that displays the contents of Ethereum blocks, gas costs, transactions, account balances, and more!

EthStats – Ethereum block explorer and analytics by Alethio

Etherscan – Ethereum block explorer

Ethplorer – Ethereum block explorer

Enterprise Tools

If you want to get a job as a developer or bring blockchain implementations to your company, you may need to become proficient with some enterprise blockchains and tools. Below are some great enterprise tools that make setting up a blockchain quick and easy.

Ethereum on Google Cloud – Build Ethereum network based on Proof of Work

Ethereum on Azure – Deployment, and governance of consortium Ethereum PoA networks

Kaleido – Use Kaleido for spinning up a consortium blockchain network. Great for PoCs and testing

Pegasys Hyperledger Besu Private Network – Run a private network of Hyperledger Besu nodes in a Docker container

PegaSys Orion – Component for performing private transactions

PegaSys Artemis – Java implementation of the Ethereum 2.0 Beacon Chain

*"Talk is cheap.
Show me the code."*

Linus Torvalds

Prebuilt UI Components

Utilize some of these prebuilt UI components to expedite your development time and create applications that are easy and fun to use. These components will save you time and make your dapp better!

aragonUI – A React library including Dapp components

components.bounties.network – A React library including Dapp components

lorikeet.design – A React library including Dapp components

ui.decentraland.org – A React library including Dapp components

dapparatus – Reusable React Dapp components

Metamask ui – Metamask React Components

DappHybrid – A cross-platform hybrid hosting mechanism for web-based decentralized applications

Nethereum.UI.Desktop – Cross-platform desktop wallet sample

eth-button – Minimalist donation button

Rimble Design System – Adaptable components and design standards for decentralized applications.

Developer Guides for 2nd Layer Infrastructure

Layer 2 infrastructure is important for various applications and the widespread adoption of crypto assets and Ethereum. Developing and using layer 2 infrastructure such as payment channels, plasma, and side-chains is important for creating more complex smart contracts and applications.

PAYMENT/STATE CHANNELS

Ethereum Payment Channel – Ethereum Payment Channel in 50 lines of code

µRaiden Documentation – Guides and Samples for µRaiden Sender/Receiver Use Cases

PLASMA

Learn Plasma – Website as Node application that was started at the 2018 IC3-Ethereum Crypto Boot Camp at Cornell University, covering all Plasma variants (MVP/Cash/Debit)

Plasma MVP – OmiseGO's research implementation of Minimal Viable Plasma

Plasma MVP Golang – Golang implementation and extension of the Minimum Viable Plasma specification

Plasma Cash – Simple Plasma Cash implementation

Plasma Exit – Automatically watch and challenge or exit from Omisego Plasma Network when needed.

Plasma OmiseGo Watcher – Interact with Plasma OmiseGo network and notifies for any byzantine events.

SIDE-CHAINS

POA Network – POA products are for people who believe that the Ethereum protocol must be cheap, fast, and scalable. POA, xDai, BlockScout, TokenBridge, POSDAO, NiftyWallet.

POA Bridge

POA Bridge UI

POA Bridge Contracts

Loom Network – is a Layer 2 scaling solution for Ethereum that is live in production.

Privacy / Confidentiality

Privacy is imperative for blockchain networks, especially when examining the best ways to product user or customer data and identity. Zero-knowledge proof technology, specifically, zkSNARKs are one of the most utilized privacy technology within Ethereum.

ZoKrates - A toolbox for zkSNARKS on Ethereum

The AZTEC Protocol - Confidential transactions on the Ethereum network, implementation is live on the Ethereum main-net

Proxy Re-encryption (PRE) NuCypher Network - A proxy re-encryption network to empower data privacy in decentralized systems pyUmbral - Threshold proxy re-encryption cryptographic library

Fully Homomorphic Encryption (FHE) NuFHE - GPU accelerated FHE library

Conclusion

While not an exhaustive list, this document has pretty much every tool you'll likely need to develop dapps, smart contracts, or build on Ethereum. So hold onto this list and check back when you need a tool or want to experiment with something new. There's always more to learn! So what are you waiting for, check out some tools, read the documentation, and get started building. And after you've built something, apply for ConsenSys Grant, join a hackathon, or participate in the greater Ethereum community.

CONSENSYS

Bad programmers worry about the code. Good programmers worry about data structures and their relationships.”

Linus Torvalds