Rerun: Entwickeln mit Ethereum

Vor rund drei Jahren haben wir das open source Projekt „Ethereum Development Environment and Network“, kurz: **EDEN**, ins Leben gerufen und in unserem Techblog [hier](https://www.maibornwolff.de/blog/eden-ethereum-development-environment-network) vorgestellt. EDEN soll das Testen und die Entwicklung im Bereich Ethereum Blockchain durch aufbereitete Test-Umgebungen, Ansammlung nützlicher Frameworks und Tools erleichtern.

So bietet das Projekt einen guten Einstieg in die Blockchain-Entwicklung und ermöglicht eine schnelle und effiziente Erprobung neuer Prototypen und Ideen auf Ethereum.

Mit dem Aufkommen neuer Konzepte und dem raschen Wandel der Technologien ist das Projekt inklusive Test-Umgebungen und Tools in die Jahre gekommen. Da der Bedarf an Blockchain-basierten Test-Umgebungen nicht abnimmt, haben wir uns es als Ziel gesetzt, das Projekt neu aufzubereiten.

Dieser Artikel stellt die wieder aufbereiteten und neuen Bestandteile von EDEN vor. EDEN’s Test-Umgebungen sind dockerisiert und mit zahlreichen Tools, verschiedenen Netzwerkmonitoren und Blockchain-Explorern ausgestattet. Das EDEN Projekt beinhaltet zwei verschiedene Arten von Setups: private Blockchain Netzwerke, in welchen die Docker-Umgebung mehrere Nodes mit jeweils unterschiedlichen Rollen startet, und zentrale Blockchain Test-Umgebungen, die vor allem für das Rapid Deploying, Testing und Analyzing genutzt werden können.

* Private­­­­­­­ Proof-of-Authority, kurz: **PoA**, Geth Blockchain, inklusive Bootnode, Fullnode und Miners
* Private Proof-of-Work, kurz: **PoW**, Geth Blockchain, inklusive Bootnode, Miner und skalierbare Anzahl an Fullnodes
* Private Proof-of-Authority Parity Blockchain, inklusive mehreren Authorities und Members.
* Private InterPlanetary File System, kurz: **IPFS**, Netzwerk

Jedes der privaten Blockchain Netzwerke ist unter der aktuellen Version von Geth, Parity und IPFS lauffähig. **Vorsicht:** Parity-Ethereum wurde als *Deprecated* erklärt, doch bleibt vorerst aus Veranschaulichungsgründen im EDEN Projekt enthalten. Neu hinzugekommen sind das PoA Geth Blockchain- und das IPFS Netzwerk Setup. PoA Geth ergänzt das deprecated PoA Parity Setup und ist zusätzlich mit zwei Ethereum Swarm Bee Nodes für Decentralized Storage ausgestattet. Das private IPFS Netzwerk kann unabhängig von einem privaten Blockchain Netzwerk genutzt werden und bietet eine Alternative zum Ethereum Swarm Bee als seperate Test-Umgebung für einen Decentralized Storage Provider. Um einen schönen visuellen Überblick über den Zustand der Test-Blockchains zu erhalten, haben wir in die Docker-Umgebungen Blockchain Explorer und Netzwerk Monitore, wie z.B. Alethio’s Lite-Explorer, Blockscout Explorer oder EthStats Monitor, integriert.

* Development Toolbox, inklusive Truffle, Ganache CLI als Test-RPC, Oyente, Smartcheck und Mythril als Smart Contract Analyzer und weiteren nützlichen Tools
* Single SKALE Node Test-Umgebung
* Open-Ethereum Test-Blockchain

Letztes Mal haben wir bereits die EDEN Toolbox für einfaches Deployment und Testing von Smart Contracts via Truffle Framework und Ganache CLI vorgestellt. Für die Analyze von Smart Contracts mit Solidity Version <= 0.4.21 war die Toolbox mit Oyente Python Scripts ausgestattet. Damit die Toolbox auch für Solidity Versionen > 0.4.21 nutzbar ist, haben wir Mythril und Smartcheck als Analyzer hinzugefügt. Zusätzlich wurde die Toolbox mit weiteren praktischen Tools angereichert, wie z.B. solhint, solidity-docgen und solgraph. Die Test-Umgebungen für SKALE und Open-Ethereum haben wir neu hinzugefügt und stellen entsprechend Test-Blockchains bzw -RPCs mit Instant-Mine/-Seal Mechanismen dar. Sie dienen zur Entwicklung und zum Testen von Ethereum dApps für das SKALE Netzwerk oder auf der Basis des Open-Ethereum Clients und zugehörigen Consensus Mechanismen. Beide Test-Umgebungen bieten visuelle Unterstützung durch Blockchain Explorer an.

Um den Einstieg in die Entwicklung in dem Bereich rundum Ethereum Blockchain so einfach wie möglich zu gestalten, haben wir alle Docker Setups mit Makefiles ausgestattet, sodass mit einem *make* Command eine Test-Umgebung gestartet werden kann, ohne weitere Konfigurationen vorzunehmen. Alle Custom Docker Images der verschiedenen Setups stehen zum Download via GitHub Container Registry bereit. EDEN soll auch in den kommenden Jahren um viele weitere nützliche Tools und Test-Umgebungen erweitert werden, dabei ist eure Hilfe gefragt und Ideen immer willkommen.