naprzód jest wdrożenie przez Microsoft w chmurze Azure Blockchain as a Service (BaaS). Aplikacje rozproszone (DApp) oparte na Ethereum są projektowane i publikowane przy wykorzystaniu innych elementów komputera światowego, takich jak Mist.

Podstawy Ethereum

Była połowa 2013 roku, kiedy przeważająca część społeczności skupionej wokół Bitcoina wdała się w romans z ideą rozszerzenia jego zastosowań, tak by stał się czymś więcej niż tylko walutą. Wkrótce nastąpił napływ nowych pomysłów, nad którymi debatowano na internetowych forach. Powszechne przykłady to rejestracja domen, ubezpieczanie aktywów, głosowanie czy internet rzeczy (IoT). Kiedy szum osłabł, poważniejsze analizy pokazały, że budowa protokołu Bitcoin znacząco ogranicza możliwość oparcia na nim aplikacji.

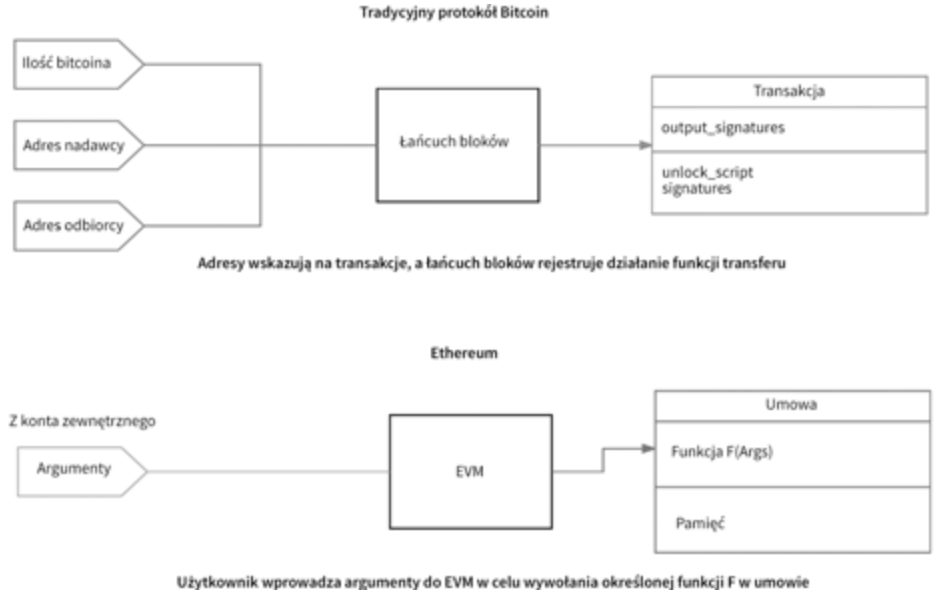
Kluczowym punktem debaty było pytanie, czy w ramach blockchaina powinien być dozwolony pełny język skryptowy, czy też aplikacje powinno się tworzyć za pomocą logiki spoza blockchaina. Debatę wywołały dwa kluczowe zagadnienia:

- Język skryptowy i OPCODES protokołu Bitcoin zostały zaprojektowane jako bardzo ograniczone funkcjonalnie.
- Protokół sam w sobie nie był wystarczająco ogólny, a kryptowaluty, takie jak Namecoin i inne, wyspecjalizowały się w konkretnym zadaniu. Główne pytanie, jakie wówczas stawiano, brzmiało: Jak sprawić, by protokół był na tyle ogólny, że będzie cechował się kompatybilnością w przód względem zastosowań, o których obecnie nic nie wiemy?

Ostatecznie wyłoniły się dwie szkoły reprezentujące odmienne poglądy na język skryptowy. Dokument Nakamoto będący przykładem podejścia klasycznego proponował, by język skryptowy był bardzo ograniczony funkcjonalnie. Pozwoliłoby to uniknąć problemów z bezpieczeństwem, jakie niosłaby za sobą obecność w blockchainie kodu wykonywalnego. W pewnym sensie kod wykonywalny blockchaina jest ograniczony do garstki podstawowych operacji aktualizujących stany rozproszone. Drugą szkołę reprezentował Buterin, który pojmował łańcuch bloków jako coś więcej niż tylko rejestr. Stworzył on wizję blockchaina jako platformy obliczeniowej, która może wykonywać dobrze zdefiniowane funkcje za pomocą umów i argumentów. Konstrukcja EVM pozwala na całkowite odizolowanie kodu wykonywalnego i bezpieczne działanie opartych na niej aplikacji. Zacznijmy od zasad budowy Ethereum i stojącej za nim głównej idei.

GŁÓWNA IDEA. W przypadku Ethereum zamiast platformy wspierającej konkretne zastosowania tworzy się platformę wspierającą natywny język programowania, który dzięki swej rozszerzalności pozwala na wdrażanie w jej ramach logiki biznesowej.

Wkrótce wrócimy do rozważenia implikacji tej zasady. Tymczasem pomówmy o innej własności Ethereum, jaką jest konsensus. Koncepcję konsensusu omawialiśmy w poprzednich rozdziałach: w przypadku kryptowalut opartych na dowodzie pracy (PoW) sieć wynagradza górników, którzy rozwiązali kryptograficzną zagadkę, zatwierdzili transakcje i wydobyli nowe bloki. Ethereum zamierza wprowadzić inny algorytm konsensusu, zwany dowodem stawki (PoS). W algorytmie tym podmiot zatwierdzający czy twórca następnego bloku zostaje wybrany w pseudolosowy sposób na podstawie stawki, jaką w sieci ma konto. Im wyższą stawkę ma się w sieci, tym większa szansa na otrzymanie tej roli. Następnie podmiot zatwierdzający zaczyna wykuwanie kolejnego bloku i otrzymuje wynagrodzenie od sieci. W tym przypadku podmiot taki naprawdę wykuwa blok (jak kowal) zamiast go wydobywać, ponieważ w PoS koncept wydobywania opartego na sprzęcie został zastąpiony wirtualną stawką. Do pewnego stopnia przesłanką przechodzenia na PoS są wysokie wymagania energetyczne algorytmów PoW, które stały się powszechną bolączką. Peercoin był pierwszą kryptowalutą, która użyła PoS, ale bardziej znaczące implementacje dokonane w ostatnim czasie dotyczą ShadowCash, Nxt i Qora. Główne różnice między Bitcoinem a Ethereum jako protokołami zostały wyszczególnione na ilustracji 4.1.



Ilustracja 4.1. Porównanie Bitcoina i Ethereum jako platform obliczeniowych