2.2.2.3.2.3 Distributed Applications

return to Ancillary Data

One of the major extensions to the original Satoshi Nakamoto [1] [27] papers on peer-to-peer electronic cash systems is the use of oracles to access external data and Smart Contract to enforce business rules on ledger transactions. For example, an account holder wants to transfer Bitcoins from one account to another, but there is a “reserve” placed on the account to cover potential debts from option trading. This “reserve” information would not be part of the Ledger, but would be ancillary data. The enforcement of the “reserve” is done by a distributed application called a smart contract.

> *A distributed application is software that is executed or runs on multiple computers within a network. These applications interact in order to achieve a specific goal or task. Traditional applications relied on a single system to run them. Even in the client-server model, the application software had to run on either the Client, or the server that the client was accessing. However, distributed applications run on both simultaneously.*
>
> *With distributed applications, if a node that is running a particular application goes down, another node can resume the task.* [1)]

A distributed application (DApp) is software that is executed or runs on multiple computers within a network simultaneously. The software is deterministic meaning that given the same inputs, they all produce the same outputs. One of the main benefits of DApps is they are extremely durable and hardened against a single point of failure. Therefore, to be distributed, deterministic and durable, any services that the DApp must interact with must also be distributed, deterministic, and durable. This makes Distributed Application (ĐApp or DApp) interaction with traditional client/server cloud services difficult. Naturally, cloud services applications such as Software as a Service (SaaS) and Data as a Service (DaaS) have some redundancy and reliability built into them; however, they cannot achieve the same level of robustness as a DApp. Therefore, SaaS and DaaS need to expose their functionality using a proxy DApp, which by its nature has latency. This latency could adversely affect the DApp’s deterministic and durability nature.

Another important aspect of a DApp is that it should be runtime environment agnostic: allowing as many DIDO nodes into the DIDO network as possible. This can be achieved using Virtual Machine (VM) such as a Java Virtual Machine (JVM) or interpretive engines such as those available with ECMAScript. There are some platform specific virtual machines available such as the Common Language Runtime (CLR) which are not standardized and do not run with the deterministic rigor on non-Windows platforms. Consequently, the list of languages allowable in DApps is limited to those that have a standardized runtime environment (i.e., VM or Engine) that runs on multiple platforms.

1)

Techopedia, “Techopedia Definitions,” 1 December 2017. https://www.techopedia.com/definition/23971/distributed-application.