

# Technology Stack Nomenclature

---

# Technology Stack Nomenclature

Digital Innovation is, by definition, a rapidly evolving sector. These guidelines are expected to be updated to keep abreast with technology, regulatory and operational developments.

Document Version: 30 October 2018

## Contents

1. Introduction.....	3
2. Networking, Infrastructure and Physical Layer .....	4
3. DLT Implementation Layer .....	5
4. DLT Application Layer.....	6
5. Off-DLT Application Layer .....	7
6. Variations and Hybrids.....	8
7. Boundaries, Dependencies and Interfaces .....	9
References .....	10

## 1. Introduction

A DLT platform, from an abstract view, provides a method for distributed nodes which:

- i) do not inherently trust each other
- ii) cannot necessarily communicate directly with all other nodes, but indirectly through other nodes

to provide for the execution of system logic without a central authority.

Blockchain was the first implementation of a DLT, however, since it was first researched, other methods to implement a DLT have been proposed. Therefore, we will not refer to any specific implementation and neither to how implementation specific algorithms (e.g. using Proof-of-Work, Proof-of-Stake, etc.), but use a generic description of what a DLT is to encapsulate different implementations.

The following is an abstract Technology Stack, which is being depicted and used by the Authority to describe an ITA:

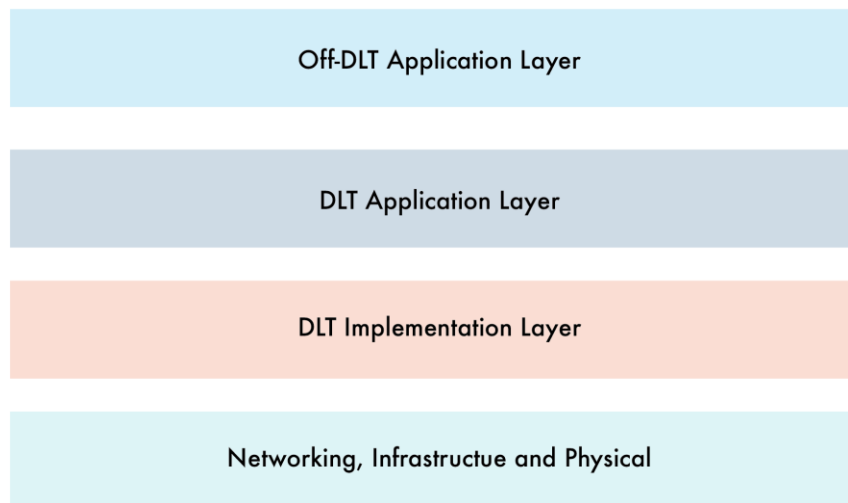


Figure 1: Technology Stack

## **2. Networking, Infrastructure and Physical Layer**

This layer consists of all hardware, software and networking protocols which are not part of the DLT implementation, yet on which the DLT implementation runs. For example, this may include TCP/IP protocols, hardware network interfaces, routers and other networking infrastructure. Whilst a DLT implementation applicant may not necessarily have access to and/or have influence on the design and implementation of this layer, the applicant should be well informed of the various components used in this layer and the implications of how such components and any changes in this layer may affect the DLT platform.

### **3. DLT Implementation Layer**

The DLT implementation layer is that system logic, typically encoded within a DLT node software application that enables distributed nodes to interact and work together in a decentralised manner to provide the trust guarantees proposed. This may include a number of components included but not limited to network access and node discovery, validation, conflict resolution, incentivisation, an application execution environment, governance mechanisms, a means of consensus and any other code or logic that is required to implement and provide the DLT platform.

## 4. DLT Application Layer

Many DLT implementations allow for application logic to be executed on top of the platform, ensuring decentralised code execution. This application logic is separate to and executes on top of the *DLT Implementation Layer*. Such application logic may or may not require knowledge of the lower level *DLT Implementation Layer* (e.g. such code does not have visibility of the fact that it is being executed on a number of DLT nodes in a decentralised manner). This layer could allow for applications to be loaded by users who may not necessarily be developers of or run a participating node within the *DLT Implementation Layer* on which it is meant to execute. The application logic that this layer encapsulates is often termed *Smart Contracts* (e.g. Solidity Smart Contracts which execute on Ethereum), *Scripts* (e.g. Bitcoin Script) and *Chaincode* (e.g. Hyperledger Chaincode). For the purpose of disambiguation, irrespective of commonly used terms, the *DLT Application Layer* is defined by that code which executes on top of the *DLT Implementation Layer* and which allows for trusted execution of applications (which are not necessarily part of the DLT Implementation).

## 5. Off-DLT Application Layer

The DLT trusted environment which the *DLT Implementation Layer* provides is limited to the *DLT Application Layer* executing on top of the DLT platform. As soon as any interaction is required from outside the *DLT Application Layer*, the trust features that the DLT platform provide do not continue to hold. The *Off-DLT Application Layer* encapsulates all logic that executes outside of the DLT platform. For example, web pages that are served from outside of the DLT, which provide users with services to interact with the DLT platform and/or DLT Application, are considered to be Off-DLT Application logic.



## **6. Variations and Hybrids**

Since the introduction of Cryptocurrencies and Blockchain systems, many different variations of DLT implementations have been proposed. It is for this reason that the stack above is a generic one, so as to encompass different DLT platform variations and hybrids - irrespective of whether they are public, private, permissioned or comprised of parts residing on separate DLTs; use Proof-of-Work, Proof-of-Stake, or any other consensus mechanism; whether some guarantees are provided in a hybrid manner (e.g. zero-knowledge proofs); or any other variation thereof.

## **7. Boundaries, Dependencies and Interfaces**

When describing the Innovative Technology Arrangement (ITA) it is important to define:

- i) the boundaries of the ITA, i.e. the parts of the ITA that fall within a specific layer or across different technology stack layers
- ii) any dependencies which the ITA relies on and their implications
- iii) the interfaces which each layer will expose for use by other layers or users (whereby users could also be other computational systems)

## References

- Distributed Ledger Technology Systems – A Conceptual Framework, Centre for Alternative Finance, Judge Business School, University of Cambridge