***Disclaimer***

The present document has been produced and approved by the <long ISGname> (<short ISGname>) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.  
It does not necessarily represent the views of the entire ETSI membership.

ETSI GR PDL 010 V0.0.4 (2021-02)

**Group REPORT**

PDL Operations in Offline Mode

Release #

<

Reference

DGR/PDL-010\_PDL\_Offline\_Mode

Keywords

PDL, ledger, Smart contracts, Accountability

***ETSI***

650 Route des Lucioles

F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C

Association à but non lucratif enregistrée à la

Sous-préfecture de Grasse (06) N° 7803/88

***Important notice***

The present document can be downloaded from:  
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:  
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

***Copyright Notification***

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI yyyy.

All rights reserved.

**DECT**TM, **PLUGTESTS**TM, **UMTS**TM and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.  
**3GPP**TM and **LTE**TM are trademarks of ETSI registered for the benefit of its Members and  
of the 3GPP Organizational Partners.  
**oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and  
of the oneM2M Partners.  
**GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Contents

Intellectual Property Rights 5

Foreword 4

Modal verbs terminology 4

Executive summary 4

Introduction 4

1 Scope 5

2 References 5

2.1 Normative references 5

2.2 Informative references 5

3 Definition of terms, symbols and abbreviations 5

3.1 Terms 5

3.2 Symbols 6

3.3 Abbreviations 6

4 Introduction and Prior Art 6

4.1 Introduction To Used Terminologies 6

4.2 General Prior Art, Including Public Ledgers 6

4.3 Specific PDL Offline Solutions 6

4.4 Related Prior Art Enabling Offline PDL 7

5 Storage models 7

5.1 Offline storage models 7

5.2 Online storage models 7

6 Protocols and Architectures 7

6.1 Offline Mechanisms & Enablers 7

6.2 Offline PDL Protocols 7

6.3 Offline PDL Architectures 7

6.4 Offline PDL Reference Design 7

7 Threats and tradeoffs in Offline storages 7

7.1 Security threats 7

7.2 Potential ways to protect 7

7.3 Tradeoffs 7

8 Conclusions and Recommendations 8

Annex A: Title of annex 9

Annex B: Title of annex 10

B.1 First clause of the annex 10

B.1.1 First subdivided clause of the annex 10

Annex: Bibliography 11

Annex : Change History 12

History 13

# Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

# Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group Permissioned Distributed Ledger (PDL).

# Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

# Executive summary

One paragraph each on:

* general role/importance of ledgers / PDL
* problem statement we are trying to address
* reasons why nodes might be offline
* contribution we make; each subsequent high level section in one paragraph
* summary on high level findings and recommendation

# Introduction

Several paragraphs each on:

* general role/importance of ledgers / PDL
* current work done by ETSI PDL and wide ecosystem
* problem statement we are trying to address
* expanded view on why nodes might be offline
* rational of our work and resulting structure of document

# Scope

The present document describes the current challenges related to data storage and ledger operations when the PDL nodes are offline (duty cycled or truly offline); the methodologies and techniques that can be applied to Smart Contracts to operate when the nodes are offline and develop secure interim storage and negotiation algorithms that ensure the integrity of the data feed to the PDLs.

# References

## 2.1 Normative references

Normative references are not applicable in the present document.

## 2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non‑specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GR PDL001 V1.1.1/(2020-03): "Permissioned Distributed Ledger (PDL); Landscape of Standards and Technologies".

[i.2] European Commission CORDIS: "CORDIS: Projects and Results".

NOTE 1: Available at https://cordis.europa.eu/projects/en

[i.3] ETSI Research and Standards Website /(2020-06-15): "Research and Standards".

NOTE 1: Available at https://www.etsi.org/research

[i.4]

# Definition of terms, symbols and abbreviations

## Terms

For the purposes of the present document, the [following] terms [given in ... and the following] apply:

**Real-Time PDL**

**Synchronous PDL**

**Asyncrhonous PDL**

**Offline PDL**

## Symbols

For the purposes of the present document, the [following] symbols [given in ... and the following] apply:

## Abbreviations

For the purposes of the present document, the [following] abbreviations [given in ... and the following] apply:

AI Artificial Intelligence

DLT Distributed Ledger Technology

ETSI European Telecommunications Standards Institute

EU European Union

H2020 Horizon 2020

ICT Information and Communication Technology

IoT Internet of Things

IP Intellectual Property

PDL Permissioned Distributed Ledger

# Introduction

## PDL Overview

*Some general intro to PDL*

*Need of architectural distribution of storages*

Graphical user interface, application

Description automatically generated

## Reasons For Offline

Discuss reasons why parts of the chain go offline.

1 – Unavailability/Connectivity of Base stations – it is possible that some or all of the base stations (which maintain the ledger) are offline.(Eusebio commented that it can be satellite or wired as well we need to broaden the idea.). Remote Location – if an IoT such as a tractor is at a remote location where no connectivity PDLs might not be available.

3 – Attack – if some attacker brings down all or majority of the validating nodes – Chongong – is there a way to identify the vulnerable nodes (vulnerable to attacks), it may be possible to bring down few nodes for sometime and run safety/security check periodically before bringing them back. Eusebio – Another aspect of possible attack, circulation of power of validity - It is also possible through (Trusted Execution Enviornment ) TEE, chaincode/smart contract might have some authority to change the role of node.

4 - Congested network (Bandwidth problem ) – if a network is congested that is many devices simultaneously ask for resources, the channel becomes really congested and none of the device possibly end up getting any resources. This can be a problem with PDL clients which require timely data to be transferred to the nodes and they are using the same channel. Even if they are online, they might not be reachable by client nodes – Tooba

5 - Maintenance – Those devices are vulnerable and become offline there should be a mechanism to find if its down because of maintenance or by attack. Also the methods to protect a node when they are under maintenance (Eusebio).

6 - Duty Cycle – can be a reason some of the nodes are offline (Chonggong).

7 - Power Supply – may be some natural/non-natural disaster or lack of resources (Eusebio).

Solution(Eusebio):

TEE should be installed at the client’s end to maintain client/chain data integrity – Eusebio will contribute as section or a paragraph.

## Offline Challenges

Discuss offline challenges from a very high level point of view

1. *Data Integrity – Data that the node need to have in order to be accepted back in the DLT network, i.e. Authentication data, cryptographical keys, connectivity data, etc. The data related with the content of the DL itself is out of scope here.*
2. *Software Integrity – Before being accepted back in the DLT network, some kind of check list should be passed with the software/program running on it.*
3. *Chain merging (Forks) - should be dealt by consensus mechanism. May be details are out of scope of this document.*
4. *Side-Chain (Eusebio will contribute)*

*5) Stale transactions – may be a part of PDL type/technology and may not be specific to offline storages problem.*

*6) Reconciliation time – when some nodes come back online after certain time, it may take some time for them to catch up with the ledger. It may be possible that by the time they reconcile service is interrupted again and they go offline again. This is specific to the moving nodes (client nodes). - may be it a part of the PDL type.*

## High-Level Reference Architecture

## Introduction to Architecture

Verbal description to the architecture, the nodes, and other important issues.

## Type of Nodes

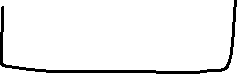
Discuss different types of nodes, discuss their roles, and how they interact;

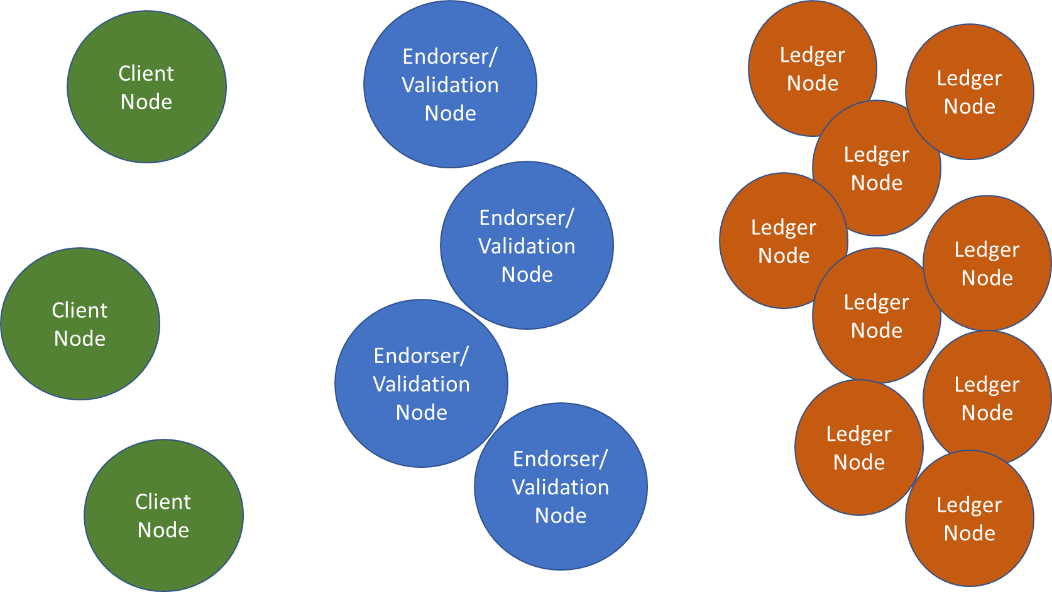
I think it is more towards hyperledger fabric architecture, we can come upto something novel.

(Chonggong ) Some additional nodes – Management/governing nodes which looks after the management of a PDL network including offline status.

Nice to have a layered model

Needs more explanation on nodes and layers – intercommunication between them. Some are nodes are there but not part of the ledger.



**

* Functionalities of the nodes (writer/readers)
* Which layer this node sits?
* Oraange (Infrastructure Layer), Blue(Operational Layer), Green(several possibilities – flexible – could be oracle or moving). Ismael would make some contributions.

## Offline Scenarios

* highlight that temporal causality is important here (in contrast to always-on ledgers); give examples of ledger ways of implementations/ philosophies
* consider both individual nodes going offline; or them going offline in a cluster (both new set of problems but also a new set of solutions)

Normal Situation:

Typically situation depends on applications, however for an IoT device it is expected that not all of the client/IoT nodes will be online all the time. They may send pings and proof-of-presence time to time.

Unexpected Situations:

Depdening on a usecase some natural disaster or any attack might bring down the devices, they may not sending data during that period(Brigitta will contribute here).

Scenarios:

* Scenario 0: all nodes online
* Scenario 1: all clients online all endorser node online some ledgers offline
* Scenario 2: all clients online some endorser offline all ledgers online
* Scenario 3: some clients offline all endorses online all ledgers online (this one important for IoT)
* Scenario 4: all clients online some endorser offline some ledgers offline
* Scenario 5: some clients offline all endorser online some ledgers offline
* Scenario 6: some clients offline some endorser offline all ledgers online
* Scenario 7: some clients offline some endorser offline some ledgers offline

## Postioning of the Endorsing node

If there are some minimal requirements for the nodes which are permemnant that is to say, certain number of nodes may always stay online. This minimal requirement can be decided by the governance of the PDLs.

## Positioning of Client Nodes

Moving devices such as connected cars may act as lightweight nodes, this means they don’t take part in consensus but still write to the ledger (to double check.)

## Recommendations for the Client nodes

*This can include:*

*How they will reconcile with the ledger*

*Should they keep the full ledger or light-weight*

**~~Q: validity of parts of a managed ledger going offline?~~**

## Temporal Characteristics

* Causality
* Gracefulness

# Technical Issues Arising From Offline Mode

## Introduction

…

## Offline Client Node

* **CONCERNED ABOUT:** 
  + Secure the offline data
  + Enable smart contract operations
  + Enable reconciliation with the main chain
* Also consider proxy nodes for smart contract scenario (scenario where node may not want to advertise it’s conditions of meeting a smart contract trigger; for instance hiding from competitors --- TBC how this is practically done)
* Make sure that offline generated content cannot be compromisesd:
  + Secure vault to temporarily store data / blocks
  + Consider using time-limited assymetric key derived from the moment the PDL was last accessible
  + Consider building a local PDL through Kubernetes clusters (tbc)
  + Ramp up local PDL which could be time limited eg deleted after a certain time to save memory (Tbc)
* Think on how to reconcile content on main PDL:
  + Constrained devices result in nodes that do not maintain the ledger, i.e. replicate the ledger. These nodes do, however, communicate with the ledger through an API; hence, offline services integrated into the API might provide better speed and scalability options.
    - The API should have the necessary facilities/functions for creating the time-limited asymmetric key.
    - The API would then also be responsible for ensuring that the data to be uploaded is in the right format, and has the right credentials.
  + Insert into PDLwhen online again
  + Post-insert and re-run consensus (hindsight consensus)
  + Post-insert but run “zipper” consensus principle
* The main aim of having a set of “orderers” in consensus for permissioned ledgers is to increase the rate of finality which is predicated on reducing the frequency of forks, or removing the possibility of forks. Hindsight consensus would in-effect create forks in the ledger and ultimately disrupt the deterministic execution of transactions.
* Hindsight consensus would also increase the energy consumed in consensus but not to the magnitude required to re-validate the previously accepted blocks, from the block height of the hindsight consensus.
* For nodes to create an offline ledger (side chain), the nodes would need to be able to create blocks while offline. If a consensus model such as the one seen in Hyperledger is assumed, each node in an offline state would assume the roles of “endorser, orderer, and validator”, to successfully append blocks to the offline ledger.

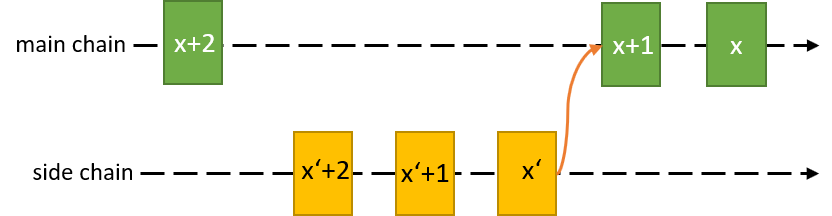


Figure 1 The view of the internal ledger of a node that has previoulsy been offline. The yellow blocks indicate the entries of the node to the ledger while disconnected from the p2p network.

## Offline Validation/Endorsement Node

## *General Considerations*

* **CONCERNED ABOUT:** 
  + Graceful disappearance eg due to network congestion (assign proxy node?)
  + Sudden disappearance, thus disrupting ongoing session (MPTCP?)
  + Disappearance of a critical mass of nodes, thus violating quora conditions

Consensus nodes are the type of nodes which partipate in the offline consensus, that is to say, that when the some of the nodes are offline, they can still run consensus among themselves. These nodes can be called as “backup nodes”

There are several situations, when the backup nodes that is the nodes those are running offline consensus, take over and run their own consensus which depends on the service providers and governance of the PDL. Some of them can be as follows:

1. When the signal quality reached below a threshold
2. When some number of base stations are offline (possibly due to malicious activity)

## *Transaction Validation*

## *Data Storage*

## *Smart Contracts*

## Offline Ledger Node

* **CONCERNED ABOUT:** 
  + Unless reaching critical amount of nodes, this is dealt with by the consensus protocol

Normally dealt with through consensus protocol

* Also explore proxy node assignment when gradual decrease in connection can be quantified
  + How can link deterioration trigger a proxy request? (Should the offline node ID be transferred to the proxy node? Important from a smart contract; should there be a specific trigger to a specific other consensus node; or should be a general request which then chooses a random consensus node; so we need to check also the “unproxy” mechanisms to avoid malicious attacks.
  + Consensus nodes in PDL are typically fixed per player (typically 1 consensus node per organisation; if there are more then either fair balance or a specific reason for an org to have more); Consider internal transfer of concensus node rights (deterministic vs random; what happens if there is no replacement within the organisation; etc)
* How do we account for false positives. For example, a temporary interference in signal of 3-5 seconds might cause the consensus node to request a proxy, but then might require the voting rights back immediately after.
* Another example might be a high retransmission rate which occurs during network peak periods. This might create a scenario where a proxy is necessary for he duration of the peak period.

## Offline Smart Contracts

*We need to see if this warrants a separate section of whether this shouldn’t be absorbed in above sections.*

* Consider reverse problem in that smart contract needs to be downloaded locally
* A scenario such as where a cluster of devices depend on the smart contract for the finality of certain operations but have lost connection to the ledger network. This loss of connection might be due to the more resource-heavy node going offline.
* A successful offline smart contract system would require
  + A trusted environment where the virtual machine (for example EVM) required to run the smart contract is hosted.
  + The bytecode for the specific smart contract must be located on the node and a hash proof of the validity of the smart contract, i.e. the smart contract on the node is the same as that on the ledger.
  + The resulting output from the execution of the offline smart contract must be synchronised to with the state of the online ledger. This synchronisation cannot happen individually to avoid a replay attack/scenario. Therefore, the synchronisation might involve a pointer to the block containing said transactions or be a singular entry of the transactions between two nodes, on the online contract.
* The obvious challenge would be reconciling the offline states with the online states. For example, the current price bid for oil online vs the submitted price bid for oil offline.
* How to create secure and trusted environment locally; how to execute locally; and how to inform back the execution results/confirmation (see Section 4.2).

## Monitoring Capabilities

* Duty cycling
* Graceful deterioration (proxy nodes)
* Sudden disappearance
* Monitoring/probing infrastructure (cloud, Kubernetes, independent PDL)
* Monitoring nodes integrity whilst they are offline? Other methods then PDL – can be traditional ones.

# Proposed Solutions

Proposed solutions

## Monitoring Capabilities

## Validator and Ledger Nodes Architecutral and security considerations and Requirements(can be a chapter)

# Offline Architecture Proposition

Glues together prior PDL work, as well as the proposed solutions in the context offline operations.

+++++ FROM HERE ONNWARD NOT TO BE DISCUSSED DURING OFFICIAL PDL CALL ++++

SEGMENT FROM TOOBA (15 Dec 2020)

# Offline data storages

## Introduction:

PDLs are distributed data storages, which means that every node in the network keeps a copy of the ledger; when an update comes all the nodes synchorize their local ledgers accordingly. The problem comes when some of the nodes are offline and they want to update the ledger; if a ledger is unreachable to nodes updating the ledger state, correct information cannot be available to the ledger. This is particularly very important when a transaction is dependent on other transactions. For example, if transaction A is dependent on transaction B, and the node sending sending transaction B is offline, and some other node X and sends transaction C and changes the state of transaction A. A mechanism to storage the data is required, to maintain the data integrity. Two major scenarios when a ledger or some of the nodes of ledgers are offline are discussed below:

1. One of the node is offline: This is a simplest scenario, when one of the node is offline. In this case, the node may have to store the data at some iterim storage to prove the integrity of the data.
2. Some of the nodes are offline: If two or more of the nodes are offline, they can create their own side-chain and fork to the main ledger when they go back online

## Challenges:

The major problem here is that, the state of PDLs changes very quickly and by the time offline nodes come back online, the main ledger might be on a completely different state.

### Data Integrity:

One of the major problems is the data integrity, PDLs achieve integrity through ledger replication to all of the node. When a node is offline when the data is stored at some iterim/middle storage, the main problem is the storage features this particular storage uses. Other mechanism of storage such as offline node establish a side-chain, can be adopted and discussed in clause 5.

### Chain merging

One possible solution, is that all the offline nodes establish a side-chain, and when they come back online, update the ledger. Since, many nodes have saved the data, and through side chain mechanism established conesus among themselves, can prove integrity of the data to the main legder. However, there are two main problems:

1. The main chain state can be moved forward and the old transactions (that is transactions in offline mode) are no longer applicable
2. If there are number of side-chains merging to the main chain/ledger number of factors should be considered such as latency of the merging(TBD).

### Stale transactions

As discussed in the earlier clause, the transactions in offline mode can become invalid after certain time; particularly when the target state is updated by some other transaction (from an online node). This can cause of monetary losses particularly; for example, if A and B are bidding for item I, and A bids higher then B but goes offline before the bid is received by the ledger, then item I may be sold to B.

When a PDL is offline, one or many nodes cannot access the main ledger, this means, records cannot be updated to the main ledger. The problem is when the offline nodes come back online, they need to prove the integrity of the data to nodes of the main ledger. Some of the methods to prove the integrity of offline ledgers is as follows:

1. Maintain a side-chain : The offline nodes can establish connection between themselves and start a side-chain, and when the main ledger comes back online they can fork their branch to the main ledger. In this strategy there are two problems, 1) there should be a minimum number of nodes that establish a side chain. Second, when number of nodes are offline and many side chains are created, forking many of them to the main chain can be a problem.
2. Cryptographic Means: In this strategy, the nodes can secure the data with pre-agreed keys(TBD)

# Introduction and Prior Art

\* discuss structure of this section

The necessity for the classification of an offline mode stems from the need to preserve and validate data that was generated when a previously online node loses connection with the rest of the p2p network. While offline full nodes are able to perform operations based on the most up-to-date version of the local ledger, the rest of the Blockchain network does not benefit from the result of the operations. These results must be cryptographically secure and timestamped in a manner that ensures the successful appending of the results to the ledger, when connection is resumed. Hence, a secure model that promotes offline security of data and the protocol for arriving at successful consensus on the back-dataed data is necessary.

From a security perspective, an offline mode might be desireable as it limits the exposure of the node to attacks i.e. less opportunities for remote attacks and thefts.

## Introduction To Used Terminologies

**Real-Time PDL**

A PDL node is considered to be in a “real-time” stater when two conditions are met:

1. The node does not have to initiate a “pull” operation i.e. the pushed blocks from other nodes sufficiently synchronise the node to the global state.
2. All nodes in the node list are “active” or “alive”. This provides the nodes with the confidence that the local ledger is up-to-date and that its pushed transactions are disseminated in real time.

Other metrics may be used such as the number of blocks needed the synchronise the node to the global state must not exceed a set threshold, as this may indicate that the node has been considerably offline. An example of such capping or threshold could be the number of blocks or block headers received in the last round from peers are 10 or less. This value can be substituted for a more adaptive figure such as 1-third of the max downloadable block/header count per message.

**Synchronous PDL**

**Asyncrhonous PDL**

A node can be considered to be in an asynchronous state when its synchronisation is periodic or where synchronisations are a function of a "pull" request. Traditionally, a node in an "active" state indicates that the node is available for push notifications. This node may still sign “alive” messages to other p2p nodes in the network. However, an additional parameter could be set that identifies the synchronisation preference of active nodes. This parameter would determine the synchronous or asynchronous state of a PDL node.

**Offline PDL**

Simply put, an offline node is a node that has no active connection to any synchronous or real-time PDL node. This node might be offline due to being duty-cycled of an inconsistent internet connection. This node might still carryout PDL related operations outside of the network either based on data from its most recent view of the ledger or self-generated data. Regardless, this data must be preserved in a state that it can be validated and appended upon connection.

## General Prior Art, Including Public Ledgers

Hardware-based Implementations

A cybersecurity company in Israel, GK8, successfully deployed an offline blockchain node. Similar to the offline node proposed by Cryptofuse, this node is fully offline and requires a tailor-made hardware to ensure certain security considerations. Their models are built around the node being “fully offline” i.e. the node will never have an active connection to the p2p network. These nodes through a combination of processes successfully pubish to the ledger network, however, all input or updates from the p2p network are confronted by several firewalls before being appended to the local ledger.

## Specific PDL Offline Solutions

## Related Prior Art Enabling Offline PDL

# Storage models

## Offline storage models

## Online storage models

# Protocols and Architectures

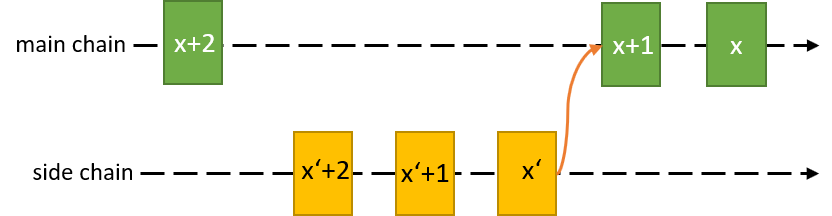
## Offline Mechanisms & Enablers

Everything about underlying tech enablers

## Offline PDL Protocols

Protcols resulting from above

A protocol similar to the GHOST protocol adopted in Ethereum might serve useful for appending back-dated transactions into the ledger. When the node is offline, all PDL operations and storage should be regarded as a side-chain operation. While these operations remain questionable to the rest of the network, the credentials of the side chain to the local node is valid. Hence, the transactions and storage on the side chain are valid, pending when there is a peg that either reference the data stored or facilitates the cryptographic operations necessary for appending the data to the chain. A peg refers to the protocol involved in achieving consensus on the data stored on the local side chain. The figure below demonstates the local storage of offline transactions and the continual synchronisation of the global ledger, while maintaining the local ledger. The protocol has to ensure the global chain will not be broken due to the presence of the offline, side-chain.



## Offline PDL Architectures

Architectures resulting from above

While offline data may be pushed onto the ledger, it should never be used to enhance or ensure any security protocols.

## Offline PDL Reference Design

TBC but would be good to extract from above a single reference design; that could help with future TS documentation.

# Threats and tradeoffs in Offline storages

## Security threats

## Potential methods to protect

## Tradeoffs

# Conclusions

Annex A:  
Title of annex

Annex B:  
Title of annex

# B.1 First clause of the annex

## B.1.1 First subdivided clause of the annex

Annex:  
Bibliography

Annex :  
Change History

| Date | Version | Information about changes |
| --- | --- | --- |
| <Month year> | <#> | <Changes made are listed in this cell> |
|  |  |  |
|  |  |  |
|  |  |  |

# History

|  |  |  |
| --- | --- | --- |
| **Document history** | | |
| v0.0.1 | 12 Nov 2020 | Creation of document, TOC and high-level content |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

*Latest changes made on 2019-09-10*