CBAN 1.0
Reference Architecture

March 2022

Disclaimer

Implementation or use of specific CBAN standards, specifications, or recommendations will be voluntary, and no Member shall be obliged to implement them by virtue of participation in CBAN. CBAN is a non-profit international organization facilitating the collaborative development and industry-wide adoption of services that will transform the settlement of traffic among global communications providers in an inclusive and interoperable ecosystem.

CBAN does not, expressly or otherwise, endorse or promote any specific products or services.

Contents

Intellectual Property Rights 6
List of Contributing Members 6
Compliance Levels 6
Executive summary 6
Introduction 8
1 Scope 9
2 References 9
3 Definition of terms, symbols and abbreviations 10
3.1 Terms 10
3.2 Symbols 16
3.3 Abbreviations 16
4 Introduction 17
5 CBAN Normative Reference Architecture 18
5.1 Definition of a Functional Block 18
5.2 Reference Architecture Overview 19
5.3 Development Guiding Principles 22
5.3.1 Platform development guiding principles 22
5.3.1.1 Platform Categories 22
5.3.1.2 Category Alpha Platform 23
5.3.1.2.1 Introduction 23
5.3.1.2.2 Category Alpha-1 Platform 24
5.3.1.2.3 Category Alpha-2 Platform 24
5.3.1.3 Category Bravo Platform 24
5.3.1.3.1 Introduction 24
5.3.1.3.2 Category Bravo-1 Platform 24
5.3.1.3.3 Category Bravo-2 Platform 24
5.3.1.4 Category “Charlie” Platform 25
5.3.1.4.1 Introduction 25
5.3.1.4.2 Category Charlie-1 Platform 25
5.3.1.4.3 Category Charlie-2 Platform 25
5.3.1.4.4 Category Charlie-3 Platform 26
5.3.1.4.5 Category Charlie-4 Platform 26
5.3.1.5 Category Delta Platform 26
5.3.1.5.1 Introduction 26
5.3.1.5.2 Category Delta-1 Platform 26
5.3.1.5.3 Category Delta-2 Platform 27
5.3.1.5.4 Category Delta-3 Platform 27
5.3.1.5.5 Category Delta-4 Platform 27
5.3.2 Application development guiding principles 27
5.3.3 Platform Services Dependency 27
5.3.4 Abstraction Layer Implementation 28
5.3.5 Implementation of Interoperability with existing/legacy systems 28
5.4 CBAN Platform Services 28
5.4.1 List of all Platform Services 28
5.4.2 CBAN Atomic Platform Services 34
5.4.2.1 Introduction to Atomic Platform Services 34
5.4.2.2 CBAN Namespace Platform Service 35
5.4.2.3 CBAN Identity Platform Service 36
5.4.2.4 CBAN Location Platform Service 36
5.4.2.5 CBAN Registration Platform Service 37
5.4.2.6 CBAN Discovery Platform Service 37

5.4.3 CBAN Composite Platform Services 37
5.4.3.1 List of all Composite platform Services 37
5.4.3.2 CBAN Messaging Platform Service 39
5.4.3.3 CBAN Policy Platform Service 39
5.4.3.4 CBAN Security Platform Service 40
5.4.3.4.1 Introduction to Security Platform Services 40
5.4.3.4.2 CBAN Authentication Platform Service 40
5.4.3.4.3 CBAN Authorization Platform Service 40
5.4.3.4.4 CBAN Cryptography Platform Service 41
5.4.3.4.5 CBAN Encryption Platform Service 41
5.4.3.4.6 CBAN Identity Management Platform Service 41
5.4.3.4.7 CBAN Key Management Platform Service 41
5.4.3.5 CBAN Logging Platform Service 41
5.4.3.6 CBAN Governance Platform Services 42
5.4.3.6.1 Introduction to Governance Platform Services 42
5.4.3.6.2 CBAN Implementation Agreements 42
5.4.3.6.2.2 Common Implementation Agreements – Common Rules 43
5.4.3.6.2.3 Common Implementation Agreements – Specific Rules 43
5.4.3.6.2.4 Specific Implementation Agreements 43
5.4.3.6.3 CBAN Governing Entity 43
5.4.3.6.3.1 Governing Entity types 43
5.4.3.6.3.2 Centralized governance 44
5.4.3.6.3.3 Decentralised governance 44
5.4.3.6.3.4 Automated governance 44
5.4.3.6.3.5 Other types of governance 44
5.4.3.6.4 Crating, Changing and Enforcing Governance IAs and rules 45
5.4.3.7 CBAN Composition Platform Service 45
5.4.3.8 CBAN Access Control Platform Service 45
5.4.3.9 CBAN Fault Tolerance Platform Service 46
5.4.3.10 CBAN Distribution Transparency Platform Service 46
5.4.3.12 CBAN Concurrency Platform Service 47
5.4.3.13 CBAN Storage related services 47
5.4.3.13.1 Types of Storage Platform Services 47
5.4.3.13.2 CBAN In Memory Storage Platform Service 47
5.4.3.13.3 CBAN File System Storage Platform Service 47
5.4.3.13.4 CBAN On-Chain Storage Platform Service 48
5.4.3.13.5 CBAN Off-Chain Storage Platform Service 48
5.4.3.13.6 CBAN Distributed Blockchain Storage Platform Service 49
5.4.3.14 CBAN Modelling related Platform Services 49
5.4.3.14.1 Introduction to Modelling 49
5.4.3.14.2 The CBAN Information Model 50
5.4.3.14.3 CBAN Data Models 50
5.4.3.14.4 CBAN Model Search 51
5.4.3.14.5 CBAN Model Stitching 52
5.4.3.15 CBAN Topology Platform Service 52
5.4.3.16 CBAN Event Processing Platform Service 52
5.4.3.17 CBAN Distributed Data Collection Platform Service 53
5.4.3.18 CBAN Distributed Secret Sharing Platform Service 54
5.4.3.19 Resource Management Platform Services 54
5.4.3.19.1 Introduction to Resource Management 54
5.4.3.19.2 Resource Discovery Platform Service 54
5.4.3.19.3 Resource Virtualization Platform Service 55
5.4.3.19.4 Resource Inventory Management Platform Service 55
5.4.3.19.4.1 Categories of Resource Inventory management 55
5.4.3.19.4.2 Node-specific resources 55
5.4.3.19.4.3 Platform resources 55
5.4.3.19.5 Resource Administration and Management Platform Service 56
5.4.3.19.6 Resource FCAPS 56
5.4.3.19.7 Resource Composition 56
5.4.3.20 CBAN Platform Service Management services 56
5.4.3.20.1 Introduction to Platform Service Management 56

5.4.3.20.2 Platform Service Discovery Platform Service....57
5.4.3.20.3 Platform Service Virtualization....57
5.4.3.20.4 Platform Service Inventory Management....57
5.4.3.20.5 Platform Service Administration and Management....57
5.4.3.20.6 Platform Service FCAPS....58
5.4.3.20.7 Platform Service Composition....58
5.4.3.21 CBAN Application Management Services....58
5.4.3.21.1 Introduction to Application Management....58
5.4.3.21.2 Application Composition....58
5.4.3.21.3 Application and Platform Service Orchestration....58
5.4.3.21.4 Orchestration Platform Service....59
5.4.3.21.5 Application Registration....59
5.4.3.22 CBAN Transaction Management Service....59
5.4.3.23 CBAN Data Model Gateway/Broker Platform Service....60
5.4.3.23.1 Introduction to presentation services....60
5.4.3.23.2 CBAN API Presentation Platform Service....61
5.4.3.23.2.1 Introduction to APIs....61
5.4.3.23.2.2 RESTful-APIs....61
5.4.2.21.1.2 Non-RESTful-APIs....61
5.4.3.24 CBAN Application Specific Services....61
5.5 CBAN Application Clients....62
5.5.1 Introduction to Application Clients....62
5.5.2 CBAN Computer Applications....62
5.5.3 CBAN Mobile Device Application....62
5.5.4 CBAN Cloud Applications....63
5.6 Summary....63

Annex (informative): Change History....64

History....65

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to CBAN. The information pertaining to these essential IPRs, if any, is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.
Pursuant to the CBAN IPR Policy, no investigation, including IPR searches, has been carried out by CBAN. No guarantee can be given as to the existence of other IPRs 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. CBAN claims no ownership of these except for any which are indicated as being the property of CBAN 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 CBAN of products, services or organizations associated with those trademarks.

List of Contributing Members

The following companies and individuals participated in the development of this document and have requested to be included in this list:

Shahar Steiff – PCCW Global LTD.

Dr. John Strassner - Futureway

Compliance Levels

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**MUST**", "**MUST NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 (RFC 2119[17], RFC 8174 [18]) of the IETF when, and only when, they appear in all capitals, as shown here. All key words must be in bold text.

Items that are **REQUIRED** (contain the words **MUST** or **MUST NOT**) are labeled as **[Rx]** for required. Items that are **RECOMMENDED** (contain the words **SHOULD** or **SHOULD NOT**) are labeled as **[Dx]** for desirable. Items that are **OPTIONAL** (contain the words **MAY** or **OPTIONAL**) are labelled as **[Ox]** for optional**.**

Executive summary

This Documentdefines a high level and abstract Normative Reference Architecture (NRA) for a Permissioned Distributed Ledger platform. This Documentalso describes the characteristics and behaviour of this platform, along with the services that it provides.
The following areas are discussed in detail:

1. CBAN Platform Services
2. Abstraction Layers
3. Interfaces and Interface Reference Points

The objectives of this document are to

- Maximize the choice of technology solutions available to entities using CBAN endorsed DLT platforms.

- Maximize CBAN endorsed DLT platforms' scalability in terms of the applications supported and the number of entities able to use them.

The CBAN NRA is described in terms of abstract foundational (required minimum) and functional components that support specific sets of functionalities and reference points that describe the standardized interactions between different parts of the platform and with the external entities or platform. This enables technology vendors and developers to focus on their respective areas of expertise and platform users to choose a best-of-breed team of vendors/developers for their specific requirements.

Introduction

This document defines a Normative Reference Architecture ("NRA") for a Distributed Ledger Technology ("DLT") platform. This document also describes the characteristics and behaviour of this platform, along with the services that it provides and exemplary solutions that can be built using it.

A NRA is a *template* for defining a solution to a particular problem domain (in this case, a DLT platform). It provides a set of common definitions of concepts, terminology, and common characteristics and behavior of the system, including a set of external Reference Points that standardize communication. This document uses a functional block architecture to define three key aspects of the CBAN settlement Platform:

- **Platform Services**, which are services and functionality provided by the CBAN settlement platform.
- **Abstraction layers**, which are Data Model Brokers allowing different and diverse applications on one side and different DLT chain types on the other side to interface with the CBAN settlement platform.
- **Modularity**, which allows evolution and adaptation of the platform to changing requirements.

The objectives of using the NRA are to:

- Maximize the choice of technology solutions available to entities using CBAN-endorsed technologies, Services, and applications.
- Maximize the CBAN settlement platforms' scalability in terms of the applications supported and the number of entities able to use them.

The CBAN NRA also provides standardized terminology to simplify the interaction between CBAN Platform Services and applications developed by technology vendors/developers.

1 Scope

This Documentdefines a NRA for the CBAN settlement platform. It also describes the characteristics and behaviour of this platform, along with the services that it provides and exemplary solutions that can be built using it.

The objectives of the present document are to:

- Maximize the choice of technology solutions available to entities using CBAN compliant DLT platforms.
- Maximize the CBAN platforms' scalability in terms of the applications supported and the number of entities able to use them.

In scope:

- Definition of an Architecture, Business and Operational guidelines, Development guidelines, Functionalities, Interfaces, Platform Services, Reference points.

Out of scope:

- Specific implementation details (e.g. Implementation of identity using a specific method). Such implementation details may be added at a later phase as separate documents or as corollaries/annexes to future releases of this document.

The approach taken in this document is to focus on defining *what* needs to happen, not *how* it is implemented.

2 References

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] Framework for Permissioned Public Blockchain Networks, Inter American Development Bank, 2021, “LACCHAIN FRAMEWORK FOR PERMISSIONED PUBLIC BLOCKCHAIN NETWORKS”.

NOTE: [i.1] Available on-line at https://publications.iadb.org/publications/english/document/LACChain-Framework-for-Permissioned-Public-Blockchain-Networks-From-Blockchain-Technology-to-Blockchain-Networks.pdf

[i.2] MEF Sonata IRP MEF 55 Lifecycle Service Orchestration, 55.0.1 October 2017: "MEF 55 - LSO Reference Architecture".

NOTE: [i.2] Available on-line at https://www.mef.net/resources/mef-55-1

[i.3] NIST Special Publication 800-162 (January 2014): "Guide to Attribute Based Access Control (ABAC) Definition and Considerations".

NOTE: [i.3] Available on-line at https://doi.org/10.6028/NIST.SP.800-162.

[i.4] Gamma, E., Helm, R. Johnson, R., Vlissides, J., "Design Patterns:_Elements of Reusable Object-Oriented Software", Addison-Wesley, Nov, 1994._ISBN 978-0201633610

[i.5] Riehle, D., "Composite Design Patterns", Proceedings of the 1997 Conference on Object-Oriented Programming Systems, Languages and Applications (OOPSLA '97), ACM Press, 1997, Page 218-228

[i.6] ABAC, Guide to Attribute Based Access Control (ABAC) Definition and Considerations, NIST Special Publication 800-162, January 2014 "https://doi.org/10.6028/NIST.SP.800-162"

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

Abstraction Layer: functionality that serves as an intermediator between subsystems that may be using different protocols, vocabulary, and methods that serves their respective purposes.

Access Control Policy: defines the privileges and permissions of a subject entity to perform operations on a set of target entities

addressable storage: content/data that can be accessed through a web link (URL)

API Broker: software that mediates between two systems with different Data Models implemented as APIs

NOTE: Also referred to as API Gateway

API Gateway: See API Broker.

application (software): program or group of programs designed to perform specific tasks for end users

application abstraction layer: APIs and interfaces, including API Brokers, enabling Applications to communicate with an ETSI-ISG-PDL Platform

Application Programming Interface (API): system of tools and resources in an operating system, enabling developers to create software applications

asynchronized data: data that does not require synchronization with other data

Attribute Based Access Control (ABAC): access control method where the subject requests for performing an operation on objects are granted/denied based on:

- Assigned attributes of the subject.
- Assigned attribute of the object.

- Environmental conditions.
- Set of policies.

NOTE: As defined by NIST [i.3].

blockchain: censorship and tamper-proof growing list of records, called blocks, that are linked using cryptography

NOTE: Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data.

business service: service that is delivered to business customers by business units

category alpha application: application that is developed and delivered to all users of said application by a single vendor/developer using a Category Alpha Platform developed by that same vendor/developer

NOTE Can only use a single DLT type prescribed by the developer.

category alpha platform: PDL platform that is designed, developed, delivered, and integrated to all users of said platform by a single vendor using a single DLT technology

NOTE: Broken down to sub-categories "Alpha-1" and "Alpha-2".

category bravo application: application that is developed and delivered to all users of said application by a single vendor/developer using a Category Bravo Platform developed by that same vendor/developer

NOTE: Can only use DLT types prescribed by the developer.

category bravo platform: PDL platform that is designed, developed, delivered, and integrated to all users of said platform by a single vendor, but can operate using two or more underlying DLT technologies

NOTE: Broken down to sub-categories "Bravo-1" and "Bravo-2".

category charlie application: application that is developed towards a specification of an Application so that any user of an application supporting such specifications can fully interoperate with other users of other applications built towards the same Application specifications

category charlie platform: PDL platform that can operate using two or more underlying DLT technologies and is designed and developed towards a specification of an application abstraction layer so that any Application that supports such an abstraction layer can interface with said platform

NOTE: Broken down to sub-categories "Charlie-1", "Charlie-2", "Charlie-3" and "Charlie-4".

category delta platform: Category Charlie platform that only supports a single DLT type

NOTE: Broken down to sub-categories "Delta-1", "Delta-2", "Delta-3" and "Delta-4".

Certificate Authority (CA): entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate

composite application: applications using the PDL platform that are made up of other applications that use the PDL platform

composition: act of creating a new object or a new functionality through combination of two or more existing objects or functionalities

concurrency: occurrence of and/or execution at the same time of different programmatic units

consumer: PDL Platform entity that consumes data produced by another entity

data model: Represents concepts of interest to an environment in a form that is dependent on data repository, data definition language, query language, implementation language, and/or protocol

NOTE: Data Models are derived from the Information Model.

data model broker: Software that mediates between two systems with different data models

NOTE: Also referred to as Data Model Gateway.

data model gateway: Same as Data Model Broker.

directly connected storage: storage that is local to the node and is either physically connected to the node or is external storage connected using a shared communication channel that is managed by the owner of that node

NOTE: Examples of physically connected storage: internal drive, external thunderbolt drive. Examples of external storage: NAS, Cloud.

Discretionary Access Control (DAC): access control policy where the owner of a resource/object defines the access control policy for the users

distributed addressable storage: addressable Storage that is distributed across multiple storage devices

Distributed Ledger Ttechnology (DLT): technology implementing a distributed ledger which is a consensus of replicated, shared, and synchronized digital data geographically spread across multiple sites, countries, or institutions

NOTE: Unlike with a distributed database, there is no central administrator.

DLT Abstraction Layer: APIs and interfaces, including API Brokers, enabling Platform services to communicate with ETSI-ISG-PDL endorsed PDL types

DLT Hardware Interface: point across which electrical, mechanical, and/or optical signals are conveyed from a sender to one or more receivers using one or more protocols

DLT Data Model Broker/Gateway: translates between data models allowing entities using different data models to communicate each using its own data model

NOTE: Details are for further study.

DLT Platform Atomic Service: PDL Platform Service that does not use any other PDL Platform Service to perform its functionality

NOTE: May use external applications or functions.

DLT Platform Composite Service: PDL Platform Service that uses one or more other PDL Platform Services to perform its functionality

DLT Platform Mandatory Service: PDL Platform Service that is mandated to be included in an ETSI-ISG-PDL compliant PDL platform

DLT Platform Optional Service: PDL Platform Service that does not need to be included in a PDL platform for it to be considered ETSI-ISG-PDL compliant

DLT Platform Service: services and functionality provided by the PDL platform that all applications may use

NOTE: Same as "Platform Service".

DLT Software Interface: point through which communication with a set of resources of a set of objects is performed

NOTE: Resources such as memory, CPU, Location, User roles or Smart Contracts.

Domain Name System (DNS): hierarchical and decentralized naming system for computers, services, or other resources connected to the Internet or a private network

NOTE: It associates various information with domain names assigned to each of the participating entities.

external data: data obtained from resources or systems external to the PDL platform

external IRP: IRP between a PDL platform and external entities

functional block: abstraction that defines the external structural representation of the capabilities and functionality of a component or module, and its relationships with other functional blocks

NOTE: Functionalities such as capabilities, behaviour, and relationships, as well as their inputs, outputs, and optionally, transfer functions. The internal structure of a functional block is **not** revealed.

functional capability: capabilities that a system has to manage resource in each functional area of operations

Governance: collection of rules and tools that control the behaviour and function of a PDL platform

implementation agreement: rules and agreements that describe how a Platform Service is implemented

information model: representation of concepts of interest to an environment in a form that is independent of data repository, data definition language, query language, implementation language, and protocol

insignificant event: event that does not affect any node other than the node where it occurred and does not affect the chain or consensus mechanism

Interface Reference Point: communication channels through which functional blocks communicate with each other

NOTE: IRPs are given names for reference purposes (e.g., "Debka").

Internal Data: Data that is generated by a node either through computation or through a directly connected sensor that feeds data to that node

Internal IRP: IRP between functional blocks internal to a PDL platform

Internet Corporation for Assigned Names and Numbers (ICANN): American multi-stakeholder group and non-profit organization responsible for coordinating the maintenance and procedures of several databases related to the namespaces and numerical spaces of the Internet, ensuring the network's stable and secure operation

Internet Engineering Task Force (IETF): open standards organization, which develops and promotes voluntary Internet standards, in particular the standards that comprise the Internet protocol suite

InterPlanetary File System (IPFS): protocol and peer-to-peer network for storing and sharing data in a distributed file system that uses content-addressing to uniquely identify each file in a global namespace connecting all computing devices

NOTE: A global namespace connecting all computing devices such as the public internet.

Loosely Coupled: functionality that has little or no dependency on other functionalities

Mandatory Access Control (MAC): access control policy defined by system administrators

Minimum Viable Product (MVP): is a version of a product with just enough features to satisfy early customers and provide feedback for future product development

Non-Addressable Storage: Content/Data that cannot be addressed and accessed by any other entity except for the entity that directly manages this data

orchestration: automated (and/or manual) configuration and management of systems and their Functional Blocks

NOTE: Orchestrated objects may be Resources, Platform Services, Applications. Orchestration emphasizes coordinated actions; one form of this coordination is service function chaining.

Platform Service: services and functionality provided by the PDL platform that all applications may use

NOTE: e.g., Governance, Identity, Storage.

policy: set of rules that is used to manage and control the changing and/or maintaining of the state of one or more managed objects, defined by the Governance

Policy Based Access Control: An Access Control method that uses Policies to determine the appropriate type of access control based on the needs of the PDL Platform

principal: highest authority or most important position in an organization, institution, group or system

producer: DLT Platform entity that generates data that other entities may consume

RAM Swap Space: portion of a computing device's hard drive that is used for virtual memory in the event that there is insufficient physical RAM installed on the device

Random Access Memory (RAM): hardware in a computing device where the operating system, application programs and data in current use are kept so they can be quickly reached by the device's processor

Reference Architecture (RA): *template* for defining a solution to a particular problem domain

Remote Procedure Call (RPC): in distributed computing, a remote procedure call is when a computer program causes a procedure to execute in a different address space, which is coded as if it were a normal procedure call, without the programmer explicitly coding the details for the remote interaction

Role Based Access Control (RBAC): access control approach based on the roles the user assumes in a system, rather than the user's identity

NOTE: As defined by NIST [i.6]

Service: instance of a technology product implemented using a CBAN compliant platform

NOTE: e.g., a communication circuit connection between two offices.

Significant Event: event that occurred on any node that may affect the behaviour of the node, the chain or the consensus mechanism

Software Reference Model: set of architectural patterns and other supporting artifacts that presents a set of unifying terminology, concepts, axioms, and functional blocks within a particular problem domain

Synchronized Data: data that requires sequencing and has dependency on timing or content of other data being collected

Tightly Coupled: functionality that has a high degree of dependency on other functionalities

Trusted Third Parties: in cryptography, a trusted third party is an entity which facilitates interactions between two parties who both trust the third party; the Third Party reviews all critical transaction communications between the parties, based on the ease of creating fraudulent digital content

Universal Resource Locator (URL): reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it

NOTE: A URL is a specific type of Uniform Resource Identifier, although many people use the two terms interchangeably.

Use Case: specific situation in which a product or service could potentially be used

Virtual Service: service that uses one or more virtual objects

NOTE: Objects such as Resources, Services.

3.2 Symbols

No symbols were used in this document.

3.3 Abbreviations

For the purposes of this document, the following abbreviations apply:

ABAC	Attribute Based Access Control
API	Application Programming Interface
CA	Certificate Authority
CPU	Central Processing Unit
DAC	Discretionary Access Control
DLT	Distributed Ledger Technology
DNS	Domain Name System
DSL	Domain Specific Language
ETSI	European Telecommunications Standards Institute
ETSI-ISG-PDL	ETSI Industry Specification Group for Permissioned Distributed Ledger
FCAPS	fault, configuration, accounting, performance, security
GDPR	General Data Protection Regulation
GPS	Global Positioning System
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IETF	Internet Engineering Task Force
IFPS	InterPlanetary File System
ICT-SP	Information and Communications Technology Service Provider
IP	Internet Protocol
IRP	Interface Reference Point
ISG	Industry Specification Group

ISO	International Organization for Standardization
LSO	Lifecycle Service Orchestration
MAC	Mandatory Access Control
MEF	MEF Forum, formerly known as Metro Ethernet Forum
NAS	Network Attached Storage
NRA	Normative Reference Architecture
PBAC	Policy Based Access Control
PC	Personal Computer
PDL	Permissioned Distributed Ledger
RA	Reference Architecture
RAM	Random Access Memory
RBAC	Role Based Access Control
SASE	Secure Access Service Edge
SDO	Standards Defining Organization
SFTP	Secure File Transfer Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TMS	Transaction Management Service
URL	Universal Resource Locator

4 Introduction

This Document defines a NRA for the CBAN settlement platform. This Document also describes the characteristics and behaviour of this platform, along with the services that it provides and exemplary solutions that can be built using it.

A NRA is a *template* for defining a solution to a particular problem domain (in this case, a DLT platform). It provides a set of common definitions of concepts, terminology, and common characteristics and behavior of the system, including a set of external Reference Points that standardize communication between the CBAN platform and external entities.. This Document uses a functional block architecture to define key aspects of a DLT Platform.

The objectives of using the Normative Reference Architecture are to:

- Maximize the choice of technology solutions available to entities using CBAN-endorsed technologies, Common Services, and applications.

- Maximize the CBAN settlement platforms' scalability in terms of the applications supported and the number of entities able to use them.

The CBAN NRA also provides standardized terminology to simplify the interaction between CBAN Platform Services and applications developed by technology vendors/developers.

The CBAN NRA depicted in Figure 1 below describes the abstract functional components that support specific sets of functionalities and reference points that describe the standardized interactions between different parts of the platform and users of that platform. This approach enables technology vendors and developers to focus on their respective areas of expertise and market leadership by providing solutions for one or more Normative Reference Architecture functional components and/or services. It also allows users to choose best of breed vendors and solutions for their specific environment and product portfolio.

The architecture aims to be independent of specific implementations to accommodate a wide range of technology solutions that comply with both the requirements of the supported applications and ensures adherence to critical architectural requirements such as interoperability, security, privacy etc.

The CBAN NRA comprises two categories of architectural components - those components mandated in all CBAN compliant platforms – ***CBAN Mandatory Platform Services***, and those components that are optional and may be included or excluded depending on the applications implemented on the DLT - ***CBAN Optional Platform Services***. This approach facilitates the introduction and support of new applications in a structured manner without changing the common, mandatory, parts. The NRA also supports the concept of a distributed lifecycle for applications, where different parties take different roles and responsibilities *e.g. Buyer versus Seller*. This expands the vendor-member space, by allowing vendors to focus on, and members to choose from, specific architectural components in the stacks and focus their offerings on the different DLTs, Platform Services, and applications.

Following the terminology and general architectural requirements, this document discusses the architectural components listed below. Covering Services and their operating model followed by describing the DLT Platform Services and their stacks in detail and concluding with presentation architectural components.

- Orchestration (Governance and process management and coordination in a complex environment. e.g. node and DLT management in an environment involving multiple competing parties or supply chains).
- Commercial related applications using DLT to create/trade value (e.g., Commercial settlement, Cryptocurrency, Asset/inventory tokenization and management).
- External and internal information exchange (e.g. Oracles, APIs, External data sources)
- Functional Capabilities (e.g., Security, Access Control)
- DLTs (common aspects of DLTs which can become agnostic – platform independence/interoperability).
- Off-chain Storage (another chain, local/cloud node that is not part of the DLT, DLT node but not sharing with other nodes of said DLT, trusted by a single node or trusted by all nodes based on governance etc.)
- Smart Contracts (commonalities, interoperability, DLT agnosticism)

5 CBAN Normative Reference Architecture

5.1 Definition of a Functional Block

This standard uses a functional block architecture to define a software reference architecture.

A Functional Block is an abstract concept that defines a “black box” structural representation of the functionality (i.e., capabilities, behavior, and relationships) of a component, module, or system. A software reference model is an abstract definition of a set of architectural patterns and other supporting artifacts that presents a set of unifying terminology, concepts, axioms, and functional

blocks within a particular problem domain. A set of functional blocks interact using a set of Internal and External IRPs that standardize communication, and collectively define the functionality provided independent of specific technologies, implementations, or other concrete details. A software reference architecture provides a template for defining interoperable solutions to a particular problem domain (e.g. an interoperable settlement platform) in accordance with applicable business rules, regulations, and other constraints.

Thus, a software reference architecture specifies the salient characteristics and behavior of a platform. This takes the form of a set of functions and services that can be used to build more complex and detailed functions and services.

5.2 Reference Architecture Overview

The CBAN NRA depicted in Figure1 is a software reference architecture for the CBAN settlement platform.

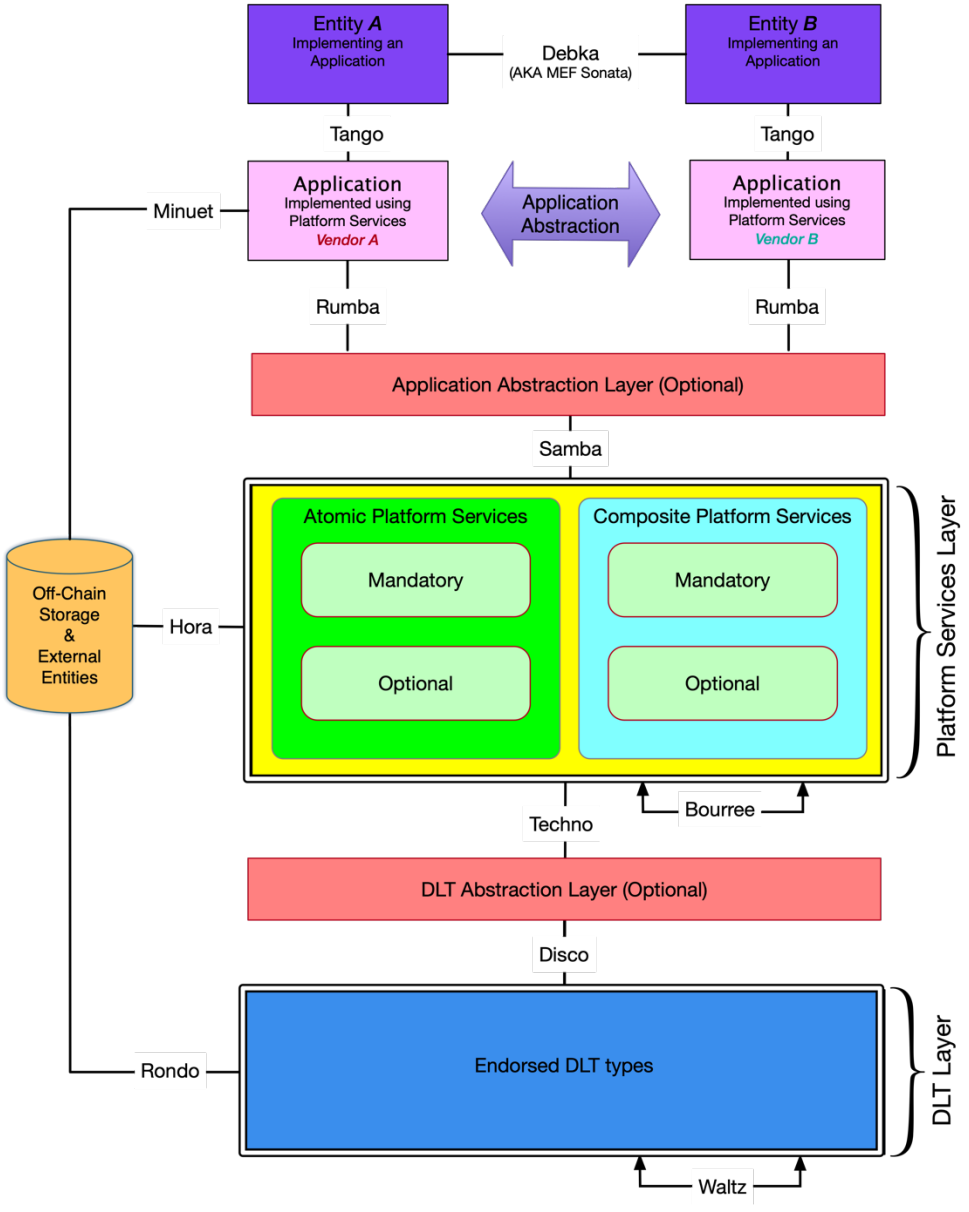

Figure 1. Main Components of the CBAN Normative Reference Architecture

The CBAN NRA is a modular architecture, and reuses individual Functional Blocks to compose new, more powerful, Functional Blocks. In addition, IRPs are defined between Functional Blocks. Accordingly:

1. **CBAN Applications**, which are applications using DLT technology.

2. **Application Abstraction Layer,** which are Data Model Brokers/Gateways enabling Applications that use different data models to communicate with the CBAN settlement platform. This layer is located between the "Samba" and "Rumba" IRPs and implemented through the Data-Model Broker Platform Service where necessary. This layer is optional and may be omitted in the "Category Alpha" and "Category Bravo" platform types as defined in Clause 5.3.1 below.

3. **CBAN Platform Services Layer**, which may support various types of applications. The Platform Services Layer can provide useful services for applications. As a result, an application could simply leverage services from the Application Service Layer, which will reduce the application's complexity, accelerate application development and deployment and increase interoperability. This is where the Application Abstraction Layer is essential, as it enables the continuing development of the DLT architecture to proceed independently of any specific requirements of interacting with external entities. For example, the Platform Services Layer could have Transaction Management Service to facilitate an application to easily create transactions without knowing details of a specific DLT type (i.e., a specific deployed DLT network); in essence, this Transaction Management Service can perform transaction transformation/adaptation between applications running on different DLT types to facilitate application operations in a complex environment. For abstraction purposes the Platform Services Layer is divided to sub-groups according to the matrix defined in Table herewith. Applications' access to services is independent of service classification and is subject to governance, identity and security considerations.

	Mandatory	**Optional**
Atomic	Mandatory Atomic CBAN Platform Services	Optional Atomic CBAN Platform Services
Composite	Mandatory Composite CBAN Platform Services	Optional Composite CBAN Platform Services

Table 1 - Service types

a. **CBAN Mandatory Platform Services** are services which the CBAN settlement platform must include.

b. **CBAN Optional Platform Services** are services the CBAN settlement platform may include. Such services should be included if required by the applications running on the platform.

c. **CBAN Atomic Platform Services**, which are services and functionality provided by the CBAN settlement platform that are independent of any other CBAN Platform Service and do not contain other CBAN Atomic or Composite Platform Services.. Such services may use external resources which are not a CBAN Platform Service (e.g. a Location service may use a GPS receiver, an Identity service may use a Certification Authority).

d. **CBAN Composite Platform Services**, which are services and functionality provided by the CBAN settlement platform that are made up of other Atomic and/or Composite CBAN Platform Services (e.g., a Security service includes an Identity service).

4. **DLT Abstraction,** which consists of a Data Model Broker/Gateway enabling Platform services to communicate with CBAN compliant DLT types regardless of the specific type of that underlying DLT. An additional functionality of such an abstraction layer is to allow interoperability between different DLT types, which may differ not only in data model structure but also on consensus mechanism and smart-contract functionality. Such an abstraction layer hides the differences between DLT types and provides a unified service-facing interface on the services side and a DLT specific interface on the DLT side. This layer is located between the “Techno” and the "Disco" IRPs and implemented through the Data-Model Broker Platform Service where applicable. This is an optional component of the architecture which can be omitted in the “Category Delta” platform type as defined in clause 5.3.1.5 below.

5. **DLT,** which is an implementation of a DLT using a specific DLT type.

6. **Interface Reference Points (IRPs),** which define communication channels through which the functional blocks defined above communicate with each other. The IRPs are given names for reference purposes (e.g., Debka, Tango, etc.).

A **DLT Data Model Broker/Gateway,** as discussed in greater detail in section 0) allows different clients (applications, external systems/entities) that use proprietary data models to interact and communicate with the CBAN Settlement platform using APIs or other communication methods. The Data-Model Broker/Gateway service together with the respective external IRPs (“Tango”, “Debka”, "Samba", "Rondo", "Hora" and "Minuet") are used to allow such communications.

[R1] The CBAN settlement platform MUST include all Mandatory Services.

[R2] The CBAN settlement platform MUST include all Optional Services required by applications using such a platform.

[O1] The CBAN settlement platform MAY include Application Specific Services.

An **Interface Reference Point** (**“IRP”**) is a logical point of interaction. The CBAN NRA defines two types of IRPs. An **External IRP** is an IRP that is used to communicate between a CBAN Platform Functional Block and an external system. An **Internal IRP** is used to communicate between two or more CBAN Platform Functional Blocks. This communication stays within the SBAN settlement platform and is not seen by external systems.

Based on Figure 1 the following IRPs are External: Rumba, Tango, Debka, Minuet, Hora, and Rondo.

Based on Figure 1 the following IRPs are Internal: Samba, Techno, Bouree, Waltz.

Note: The "Disco" IRP may be considered an Internal or an External IRP depending on the implementation. When the Application and DLT are implemented on the same node (physical or logical/virtual) it will be an Internal IRP. When it is implemented on different nodes it becomes an External IRP.

Note: The "Debka" IRP is equivalent to the "Sonata" IRP on the MEF-55 LSO Reference Architecture [i.2].

An **Interface** describes the public characteristics and behavior that specify a software contract for performing a service specific action that is implemented through an IRP. There may be multiple Interfaces implemented on an IRP. CBAN will define Software Interfaces and APIs, and optionally, hardware interfaces. An External IRP defines a *message channel*, which is a dedicated communications path connecting two endpoints that has specific associated semantics.

There are two types of CBAN interfaces:

- **Software Interface** defines a point through which communication with a set of resources (e.g., memory or CPU) of a set of objects is performed. This decouples the implementation of a software function from the rest of the system. It consists of tools, object methods, and other elements of a model and/or code. A commonly used Software Interface is an **Application Programming Interface** (API) which is a set of communication mechanisms through which a developer constructs a computer program. APIs simplify producing programs, since they abstract the underlying implementation and only expose the objects, and the characteristics and behavior of those objects that are needed. Other software interfaces may include protocols, DSL (Domain Specific Language) and more.

- **Hardware Interface** is a point across which electrical, mechanical, and/or optical signals are conveyed from a sender to one or more receivers using one or more protocols. A Hardware Interface decouples the hardware implementation from other Functional Blocks in a system. Examples may include a sensor (e.g. thermometer) connected by wire to a node, an Ethernet cable connected to a node, a fiber-channel connection between a node and directly-attached storage.

  [R3] The CBAN settlement platform MUST use External Reference Points to communicate to external systems.

Note: Platform Services can be either DLT-specific (availability of certain/all features mandates use of a specific DLT type) or DLT-independent (all features are available on all DLTs compliant with the CBAN Normative Reference Architecture).

[D1] CBAN Platform Services **SHOULD** be DLT-Independent.

[O2] CBAN Platform Services **MAY** be DLT-Specific.

5.3 Development Guiding Principles

5.3.1 Platform development guiding principles

5.3.1.1 Platform Categories

DLT platforms fall into four major categories as defined herewith. Some of those major categories can then be broken down to sub-categories. The CBAN settlement platform may start off in one

category and evolve with time into other categories. This document does not prescribe which category the CBAN settlement platform should belong to.

- Platforms that are designed, developed, delivered, and integrated to all users of said platform by a single vendor using a single DLT technology. Such platforms will be labeled as ***"Category Alpha Platforms"*** for the remainder of this document.

- Platforms that are designed, developed, delivered, and integrated to all users of said platform by a single vendor, but can operate using two or more underlying DLT technologies. Such platforms will be labeled ***"Category Bravo Platforms"*** for the remainder of this document.

- Platforms that can operate using two or more underlying DLT technologies and are designed and developed towards a specification of an Application abstraction layer so that any Application that supports such an abstraction layer can interface with said platform. Such platforms are labeled as ***"Category Charlie Platforms"*** for the remainder of this document.

- Platforms using a single DLT technology that are designed and developed towards a specification of an Application abstraction layer so that any Application that supports such an abstraction layer can interface with said platform. Such platforms are labeled as ***"Category Delta Platforms"*** for the remainder of this document.

5.3.1.2 Category Alpha Platform

5.3.1.2.1 Introduction

A Category "Alpha" platform is designed, developed, delivered, and integrated to all users of said platform by a single vendor using a single DLT technology.

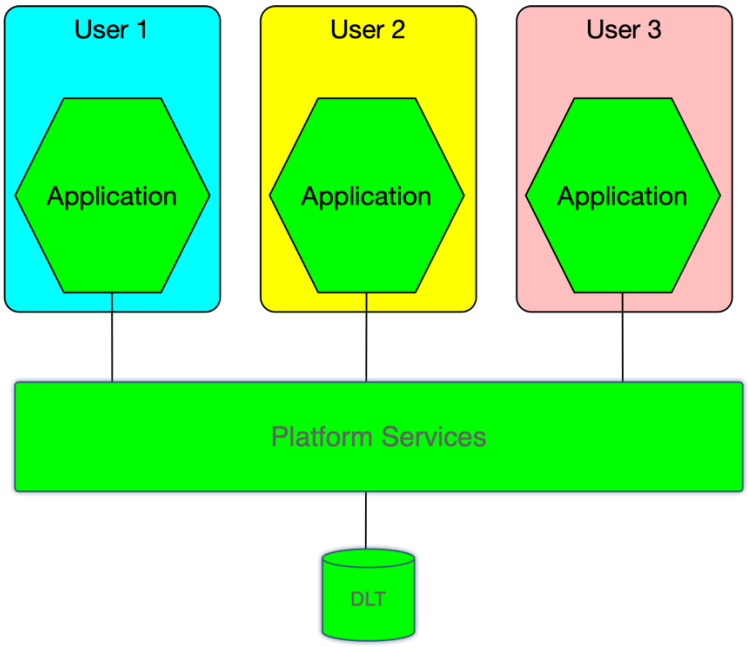

Figure 2 - A Category "Alpha" platform

The "Alpha" category is broken down into two options:

5.3.1.2.2 Category Alpha-1 Platform

The DLT and some or all the Platform Services are proprietary to the vendor.

5.3.1.2.3 Category Alpha-2 Platform

The DLT and all the Platform Services are open-sourced.

5.3.1.3 Category Bravo Platform

5.3.1.3.1 Introduction

A Category "Bravo" platform is designed, developed, delivered, and integrated to all users of said platform by a single vendor, but can operate using two or more underlying DLT technologies. A Category "Bravo" platform includes an abstraction layer between the DLT layer and the Platform Services layer that offers a unified northbound interface between the abstraction layer and the Platform Services layer, and a unique, per DLT type, interface between the abstraction layer and the specific DLT types. This abstraction layer is labeled as the ***"DLT Abstraction Layer"*** for the remainder of this document.

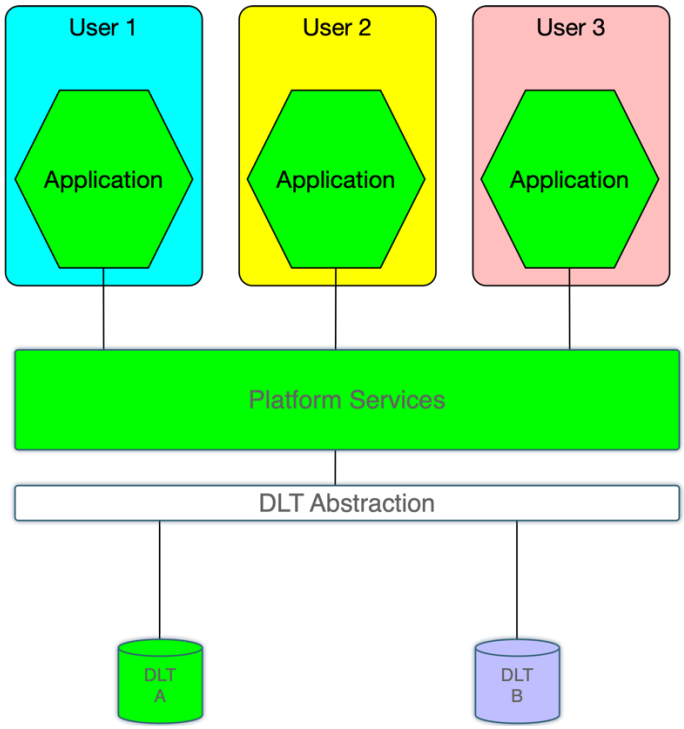

Figure 3 - Category "Bravo" platform

The "Bravo" category is broken down into two options:

5.3.1.3.2 Category Bravo-1 Platform

One or more of the underlying DLT types and some or all the Platform Services are proprietary to a vendor.

5.3.1.3.3 Category Bravo-2 Platform

The DLTs and all the Platform services are open-sourced.

5.3.1.4 Category “Charlie” Platform

5.3.1.4.1 Introduction

A Category “Charlie” platform is designed and developed towards a specification of an Application Abstraction Layer so that any application that supports such an abstraction layer can interface with the CBAN platform.
This abstraction layer is labeled as the ***“Application Abstraction Layer”*** for the reminder of this document. The Application Abstraction Layer implements a unified northbound interface between the abstraction layer and the applications using the platform, and a per-platform-specific-service interface between the abstraction layer and the underlying services implemented in the Platform Services layer.

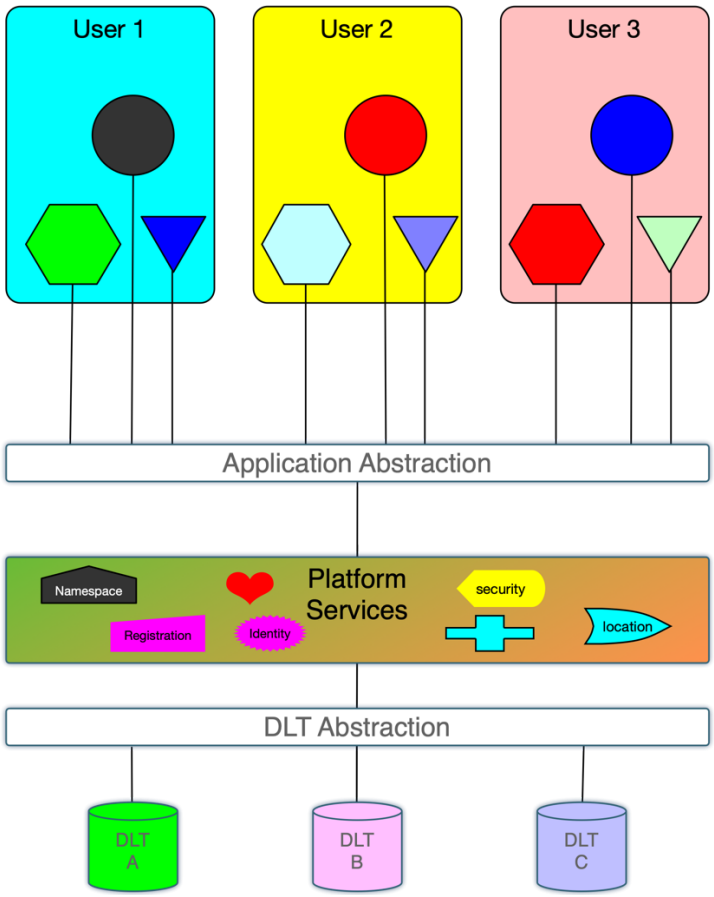

Figure 4 - Category “Charlie” platform

The “Charlie” category is broken down into four options:

5.3.1.4.2 Category Charlie-1 Platform

The platform is being developed and integrated by a single vendor who may integrate third party elements into the platform and may include proprietary elements in the platform.

5.3.1.4.3 Category Charlie-2 Platform

The platform is being developed and integrated by a single vendor who may integrate third party elements into the platform and all elements, including the third-party elements, are open-sourced.

5.3.1.4.4 Category Charlie-3 Platform

The platform consists of a collection of interoperable modules, each offering one or more of the platform services. Such modules may be developed by different vendors towards service specifications defined or endorsed by CBAN. Integration of such modules into an operational platform may be performed by any entity as long as the resulting platform complies with certification tests performed by CBAN or a certification entity endorsed by CBAN. Some or all of the modules may be proprietary.

5.3.1.4.5 Category Charlie-4 Platform

Like Category Charlie-3 Platform but all modules must be open-sourced.

5.3.1.5 Category Delta Platform

5.3.1.5.1 Introduction

A Category "Delta" platform is the same as Category Charlie with the exception that it uses a single DLT, thus eliminating the need for the DLT Abstraction layer.

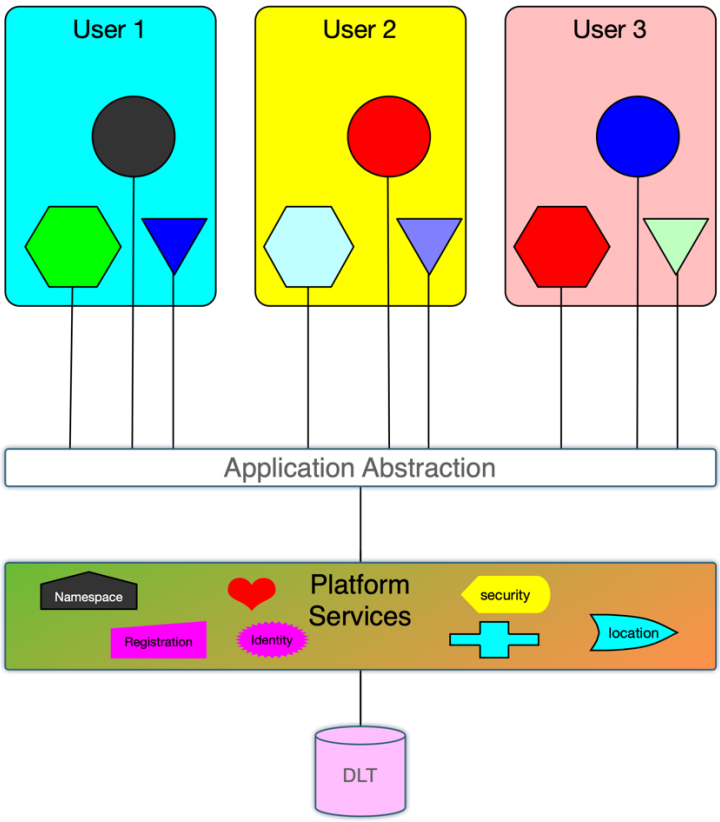

Figure 5 - Category "Delta" platform

The "Delta" category is broken down into four options:

5.3.1.5.2 Category Delta-1 Platform

The platform is being developed and integrated by a single vendor who may integrate third party elements into the platform and may include proprietary elements in the platform.

5.3.1.5.3 Category Delta-2 Platform

The platform is being developed and integrated by a single vendor who may integrate third party elements into the platform and all elements, including the third-party elements, are open-sourced.

5.3.1.5.4 Category Delta-3 Platform

The platform consists of a collection of interoperable modules, each offering one or more of the platform services. Such modules may be developed by different vendors towards service specifications defined or endorsed by CBAN. Integration of such modules into an operational platform may be performed by any entity as long as the resulting platform complies with certification tests performed by CBAN or a certification entity endorsed by CBAN. Some or all of the modules may be proprietary.

5.3.1.5.5 Category Delta-4 Platform

Like Category Delta-3 Platform but all modules have to be open-sourced.

5.3.2 Application development guiding principles

The guiding principles of Application development follow similar logic and categorization of platform development principles:

- Applications that are developed and delivered to all users of said application by a single vendor using a Category Alpha Platform developed by that same vendor and thus can only use a prescribed DLT type. Such applications will be labeled as ***"Category Alpha Applications"*** for the remainder of this document.
- Applications that are developed and delivered to all users of said application by a single vendor using a Category Bravo Platform developed by that same vendor. Such applications will be labeled as ***"Category Bravo Applications"*** for the remainder of this document. Category Bravo Applications are not limited to a prescribed DLT type and can be implemented using any DLT type supported by the Category Bravo Platform.
- Applications that are developed towards a specification of an Application so that any user of an application supporting such specifications can fully interoperate with other users of other applications built towards the same Application specifications. Such applications are labeled as ***"Category Charlie Applications"*** for the remainder of this document.

note: *Category Delta Applications,* is redundant to *Category Charlie Applications* as the Platform Services Layer hides the underlying DLT type hence there is no such category defined.

5.3.3 Platform Services Dependency

Due to the dependency of Composite Platform services on other Platform Services, when a Composite Platform Service is implemented in the CBAN settlement platform, and that Composite Platform Service is using an Optional Platform Service, that Optional Platform Service must be implemented on the CBAN settlement platform.

[R4] When a Composite Platform Service that is implemented in the CBAN Platform is dependent on or made up of an Optional Platform Service, said Optional Platform Service **MUST** be implemented in the CBAN Platform.

5.3.4 Abstraction Layer Implementation

An Abstraction Layer is an abstract structure that serves as an intermediator between subsystems that may be using different vocabulary and methods that serve their respective purposes. As discussed earlier, the CBAN settlement platform may include up to two abstraction layers: An *Application Abstraction Layer* and a *DLT Abstraction Layer*. The functionality of an abstraction layer is implemented by routing all ingress and egress communications traversing through the external IRPs to the Data-Model Broker/Gateway Platform Service. A typical implementation is described in Figure 6 herewith:

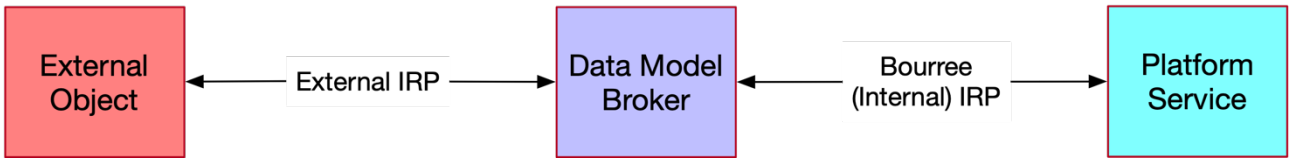

Figure 6 - Abstraction Layer Implementation

The "External Objects" depicted above may be any object external to the Platform Services Layer. E.g., an application, a DLT, external storage, an external service/platform.

5.3.5 Implementation of Interoperability with existing/legacy systems

Interoperability with legacy/existing systems is tailored on a per-application-per-system level and is not part of the CBAN settlement platform. Interoperability will be realized through the *Minuet* external IRP and needs to be developed collaboratively between the application developer and the development resources (internal/external) used by the respective carrier to develop their existing/legacy systems.

5.4 CBAN Platform Services

5.4.1 List of all Platform Services

As discussed in the previous section, the CBAN Platform Services is a set of modular Functional Blocks that are either CBAN Platform Services themselves (Atomic Services) or are used to create CBAN Platform Composite services. In order to maximize reusability, Composite services are built using composition and appropriate software patterns [i.4, i.5]. This enables improved components of a composition to be used without affecting other services.

Table 2 herewith lists all the CBAN Platform Services.

CBAN Platform Service name	**Mandatory (M) or Optional (O)**	**Atomic (A) or Composite (C)**	**Location in this document (section number)**	**Short description**
Namespace	M	A	5.4.2.2	Ensures that all of a given set of objects for a particular function have unique names.

Identity	M	A	5.4.2.3	Unambiguously identifies an instance of an entity from all other instances of this and other objects.
Location	O	A	5.4.2.4	Associates an object with a location.
Registration	O	A	5.4.2.5	List a managed object with authorities or registries.
Discovery	O	A	5.4.2.6	Discovery of services offered by the services layer and discovery of DLT networks.
Messaging	M	C	5.4.3.2	Enables communication between a group of entities.
Policy	O	C	5.4.3.3	Manage and control the changing and/or maintaining of the state of managed objects.
Security	M	C	5.4.3.4	A collection of services that assess, reduce, protect, and manage security risks.
Authentication	M	C	5.4.3.4.2	Verifies that a subject requesting to perform an operation on a target is who they say they are.
Authorization	O	C	5.4.3.4.3	Permitting or denying access to a target by a subject.
Cryptography	O	C	5.4.3.4.4	Managing protocols that prevent third parties from reading private communications.
Encryption	O	C	5.4.3.4.5	Encoding information using a key into an unintelligible form.

Identity Management	O	C	5.4.3.4.6	Access control based on the identity of an entity.
Key Management	O	C	5.4.3.4.7	Management of cryptographic keys.
Logging	O	C	5.4.3.5	Dynamic ingestion and collection of logs.
Governance	M	C	5.4.3.6	Rules and tools that control the behavior and function of a DLT.
Implementation Agreements	O	C	5.4.3.6.2	Rules and agreements that describe how CBAN Platform Services are implemented and control the behavior of a DLT platform.
Governing Entity	M	C	5.4.3.6.3	Defines the rules and implementation agreements. Ensures compliance. Resolves conflicts where needed.
Composition	O	C	5.4.3.7	Defines who can compose new services and how such new services are composed.
Access Control	M	C	5.4.3.8	Defines who can perform which operations on which set of *target* entities.
Fault Tolerance	O	C	5.4.3.9	Defines how to handle faulty instructions.
Distribution Transparency	O	C	5.4.3.10	Defines how to maintain transparency when distributing information to target entities.
Publish and Subscribe	O	C	5.4.3.11	Defines how entities publish services and subscribe to services.

Concurrency	O	C	5.4.3.12	Defines how entities handle concurrency.
Storage	M	C	5.4.3.13	A group of services related to Storage.
In Memory Storage	M	C	5.4.3.13.2	Data that is stored in the RAM of a computer running an application.
File System Storage	M	C	5.4.3.13.3	Storage on a directly connected storage device.
On-Chain Storage	M	C	5.4.3.13.4	Application data that is stored in blocks on all nodes using the chain.
Off-Chain storge	O	C	5.4.3.13.5	Information in a digital, machine-readable medium that is not stored on the main chain.
Distributed Blockchain Storage	M	C	5.4.3.13.6	Storage on a Distributed Blockchain ledger.
Modelling	M	C	5.4.3.14	A group of services related to Modeling.
Information Model	M	C	5.4.3.14.2	Presentation of concepts of interest to platform management environment in a *technology-neutral* form as *abstract* objects and relationships between objects.
Data Model	M	C	5.4.3.14.3	Representation of applicable concepts in a *technology-specific concrete* form.
Model Search	O	C	5.4.3.14.4	Enables search for specific or generic models within existing information and data models.

Model Stitching	O	C	5.4.3.14.5	Enables integrating multiple models or parts of models into a single model.
Topology	M	C	5.4.3.15	Allows a node to identify other nodes on the DLT and identify which nodes to communicate with when performing DLT related tasks.
Event Processing	M	C	5.4.3.16	Processes node-specific and platform-wide events as they occur.
Distributed Data Collection	O	C	5.4.3.17	Performs tasks related to collection of data.
Distributed Secret Sharing	O	C	5.4.3.18	Sharing of confidential data between nodes in a manner that maintains confidentiality of the data.
Resource Management	M	C	5.4.3.19	Defines how to administer and manage Resources.
Resource Discovery	O	C	5.4.3.19.2	Enables discovery of resources available to applications and nodes.
Resource Virtualization	O	C	5.4.3.19.3	Creating a virtual resource that mimics the behavior of a physical resource.
Resource Inventory Management	O	C	5.4.3.19.4	Management of node-specific and platform-wide resource inventory.
Resource Admin and Management	M	C	5.4.3.19.5	Administration and management of node-specific and platform-wide resources.

Resource FCAPS	O	C	5.4.3.19.6	Resource management tasks defined by the ISO model.
Resource Composition	O	C	5.4.3.19.7	Management of composite resources.
Platform Services Management	M	C	5.4.3.20	Defines how to administer and manage Platform Services.
Platform Service Discovery	M	C	5.4.3.20.2	Provides means to discover services available to applications and nodes.
Platform Service Virtualization	O	C	5.4.3.20.3	Creating a service using virtual resources.
Platform Service Inventory Management	O	C	5.4.3.20.4	Keeping track of inventory and serviceability of Platform services.
Platform Service Admin and Management	M	C	5.4.3.20.5	Administration and management of Platform Services through governance.
Platform Service FCAPS	O	C	5.4.3.20.6	Platform Service management tasks defined by the ISO model.
Platform Service Composition	O	C	5.4.3.20.7	Management of the composition of Composite Platform Services.
Application Management	M	C	5.4.3.21	Handles composition and orchestration of Applications.
Application Composition	M	C	5.4.3.21.2	Composing an Application from two or more managed objects.
Application and Service Orchestration	O	C	5.4.3.21.3	Orchestrating multiple managed objects so they are chained in the

				right sequence and their operation is synchronized.
Orchestration	O	C	5.4.3.21.4	Orchestration of objects so they are chained in the right sequence and topology resulting in new functionality.
Platform Exploration	O	C	5.4.3.21.5	Allows an application to indicate its requirements and explore whether the platform offers such service capabilities
Application Registration	O	C	5.4.3.21.6	Registers and lists all applications operated on a platform.
Transaction Management	O	C	5.4.3.22	Facilitates transaction related interactions between applications/services and underlying DLT networks.
Data Model Gateway/Broker	O	C	5.4.3.23	Defines tools that enable two systems with different data models to interact.
API Presentation	O	C	5.4.3.23.2	A specific Data Model Gateway/Broker implementation for environments that use APIs to exchange data between objects.
Application Specific Services	O	C	5.4.3.24	Serve a specific application or a group of applications but not required or used by other applications using the platform.

Table 1 - CBAN Platform Services

5.4.2 CBAN Atomic Platform Services

5.4.2.1 Introduction to Atomic Platform Services

The CBAN Atomic Platform Services are a set of DLT Platform Services that other CBAN Services may use, either directly or indirectly. Atomic Platform Services do not use any other DLT Platform Service but may use services external to the CBAN platform.

[R5] Atomic Platform Services **MUST NOT** use any other Platform Service to fulfill their functionality.

[O3] Atomic Platform Services **MAY** use services external to the CBAN Platform.

There are five (5) CBAN Atomic Platform Services, four (4) of which are also Mandatory Platform Services. They are shown in Figure 7 herewith.

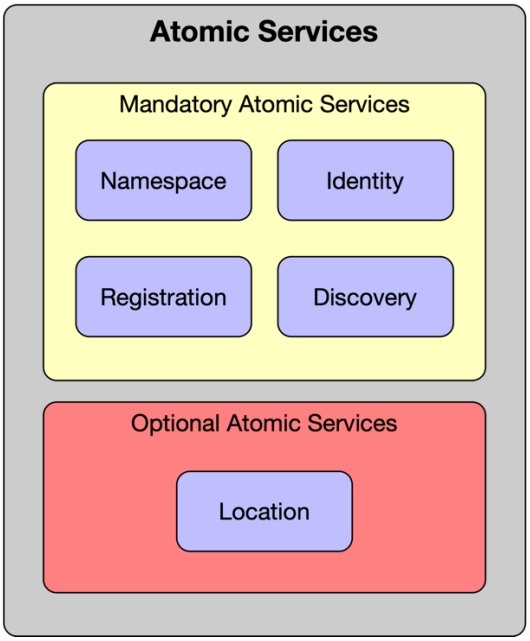

Figure 7 - CBAN Atomic Platform Services

5.4.2.2 CBAN Namespace Platform Service

The CBAN Namespace Platform Service ensures that all of a given set of objects for a particular function have unique names so that they can be easily identified. This enables multiple internal and external domains to communicate and interact with each other while avoiding name collisions between multiple identifiers that share the same name for a given object. Examples of internal domains are different administrative domains within an organization (e.g., engineering and sales), while examples of external domains include different partners (e.g., service and content providers) of an organization.

[R6] The CBAN Namespace Platform Service **MUST** provide a unique name for each managed object in its models that distinguishes each object instance from all other object instances (including multiple instances of the same object) that it contains.

Namespaces provide a scope for object names. Namespaces are typically structured as hierarchies to allow reuse of names in different contexts. Examples include file systems and DNS. A namespace is a *scoping container*. Examples include application container and messaging container services.

[D2] The CBAN Namespace Platform Service **SHOULD** support hierarchical names.

Namespaces may be simplified by using consistent prefixes for each namespace.

[D3] A name in the CBAN Namespace Platform Service **SHOULD** consist of a namespace identifier and a local (to that namespace) unique name.

5.4.2.3 CBAN Identity Platform Service

The Identity of an entity is a set of context-dependent digital identifiers that unambiguously identify an instance of that entity from all other instances of this and other objects. An identity may require multiple attributes to uniquely identify it (e.g., two products with the same name have other different attributes, such as different serial numbers).

[R7] A CBAN Identity **MUST** be constructed using one or more context-dependent digital identifiers that enable an object instance to be unambiguously identified.

A digital identifier is a secure object that is unique within a particular namespace. It is recommended that every digital identifier is assigned a namespace.

[D4] A CBAN digital identifier **SHOULD** be defined within a namespace to guarantee its uniqueness.

An entity may be used in different situations. Therefore, the same entity may be identified using a different set of digital identifiers for each situation. This enables the semantics of the use of an entity in each situation to be taken into account.

[O4] A CBAN Managed Object **MAY** have multiple context-dependent digital identifiers for establishing the Identity of that Managed Object in different situations in which it is used.

A CBAN Identity Service provides a single identity token per instance of an entity for all services so that this instance is identified unambiguously and in the same manner by all services..

[R8] A CBAN Identity Service **MUST** provide a single digital identity token per instance of an entity.

5.4.2.4 CBAN Location Platform Service

The location of an entity may or may not be relevant to the function of the CBAN Platform or a service, thus this Atomic Platform Service is optional. In applications and scenarios where location is of essence, it may affect factors such as network latency (and the resulting transaction speeds), governing laws and regulations, costs, access restrictions and more. There are multiple methods of defining locations. There are physical addresses (e.g., GPS longitude/latitude coordinates, street addresses, postal codes, building names), relative addresses (e.g., “50 meters east of the main gate”, and Virtual locations (e.g., IP address, Telephone number, MAC address). Certain location descriptors are more accurate than others (e.g., a postal code may relate to a whole street while GPS coordinates may define a location with an accuracy of a few meters). Methods of representing locations vary from country to country (e.g., coordinate systems, street name and number, postal codes).

[O5] An CBAN Managed Object **MAY** be associated with a set (one or more) of locations.

[R9] The location of a CBAN Managed Object associated with a location **MUST** be represented in a method understood in the respective geography where it is located.

[R10] The location of a CBAN Managed Object associated with a location **MUST** be defined using a location method compliant with the level of accuracy required by the respective application.

5.4.2.5 CBAN Registration Platform Service

Registration services provide means to list a CBAN Managed Object with local or international authorities or registries. Such registries allow reference to such Managed Objects for legal, commercial, and Operational purposes. Registration requirements vary with geography, though not all registries are linked to the geography in which they are used. Certain Managed Objects (e.g., a DLT serving a geographically diverse application) operate in multiple geographies and may require multiple registrations.

[O6] A CBAN Managed Object **MAY** be registered in one or more registries.

[R11] A registered CBAN Managed Object **MUST** be registered in accordance with the regulations and rules applicable in the geographies in which it operates.

5.4.2.6 CBAN Discovery Platform Service

Discovery services provide means to:

1. Discover CBAN Platform Services offered by the CBAN Platform Services layer; and/or
2. Discover a registered DLT object (e.g., entity, network).

For example, an application can discover the available CBAN Platform Services. In another example, a CBAN Platform Service can discover an underlying DLT network, which has been registered to the CBAN Platform Services layer.

[R12] The CBAN Services Layer **MUST** have a CBAN Discovery Service.

[R13] The CBAN Discovery Service **MUST** support discovery of CBAN Platform Services.

[R14] The CBAN Discovery Service **MUST** support discovery of DLT networks that have been registered to the CBAN Service Layer.

[R15] The CBAN Discovery Service **MUST** support discovery of DLT managed objects that have been registered to the CBAN Platform Services Layer.

5.4.3 CBAN Composite Platform Services

5.4.3.1 List of all Composite platform Services

The CBAN Composite Platform Services are a set of Functional Blocks that provide services that other Platform Services use, either directly or indirectly. They use one or more other Platform Services to fulfil their functionality. Composition allows building more complex architectural concepts and functions. There are a total of 53 Composite Platform Services, 16 of which are Mandatory. Services are grouped into sub-groups by their function for reference purposes but are

non-hierarchical. Any Platform Service or application may use any other Platform service. Some of the Composite Platforms are Mandatory and some are Optional. They are shown in Figure 8. Additional Platform Services may be added in the future where needed.

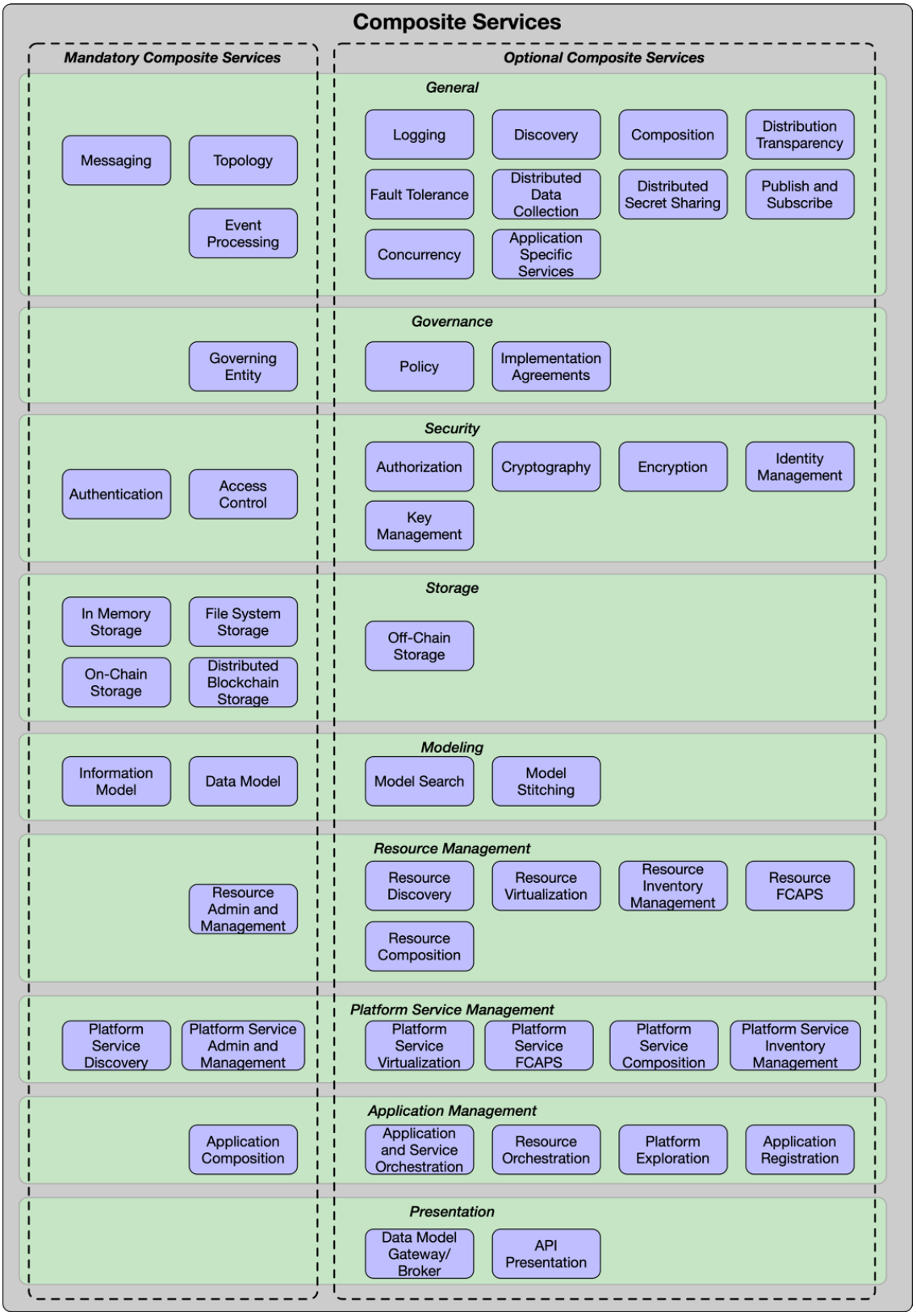

Figure 8 - CBAN Composite Services Overview

5.4.3.2 CBAN Messaging Platform Service

The CBAN Messaging Platform Service enables communication between a group of entities (e.g., DLT nodes, Application users, Platform Services). A message is a discrete unit of communication, sent by a *producer* and received by a *consumer*. There are two fundamentally different types of messaging:

- Synchronous communication, which is a *tightly coupled* solution to exchange information (e.g., opening a socket over a connection-oriented protocol such as TCP/IP and transmitting data through it).
- Asynchronous communication, which is a *loosely coupled* solution that minimizes producer and consumer dependencies.

Synchronous messaging is tightly coupled because of its main three dependencies: temporal (all components must be available at the same time), location (each component must know the address of each other component), and data structure (all components must agree on the data format and on the binary representation). Asynchronous messaging acts as an indirection layer among entities that want to communicate, removing the above three dependencies.

[R16] The CBAN Messaging Framework Service **MUST** support asynchronous communications.

[O7] The CBAN Messaging Framework Service **MAY** support synchronous communications.

There are two types of asynchronous communication models. A Message Broker is a centralized system that receives messages, determines the correct destination for each message, and sends the message to that destination. A Message Bus enables interacting entities to communicate using a set of shared interfaces.

[R17] The CBAN Messaging Framework Service **MUST** support a Message Bus.

[O8] The CBAN Messaging Framework Service **MAY** support a Message Broker.

5.4.3.3 CBAN Policy Platform Service

A Policy is a set of rules that is used to manage and control the changing and/or maintaining of the state of one or more managed objects. The CBAN Policy Platform Service is a collection of technologies that enable policies to be created, validated, read, updated, deleted, and managed. CBAN clients must use policies to interact with the CBAN platform.

[R18] CBAN-compliant client implementations **MUST** use policies to communicate and interact with the CBAN settlement platform.

Policies are used in two important ways. First, they enable a consistent and auditable delivery mechanism for requesting and receiving data, and performing commands, to be implemented. Second, policies provide a common communications mechanism for exchanging information and commands.

[R19] Components of a distributed implementation of the CBAN platform **MUST** use policies to exchange information and commands.

[D5] CBAN-compliant client implementations **SHOULD** use policies for requesting services of, and exchanging information with, the CBAN platform.

[R20] Policies **MUST** be defined, maintained, and enforced by the Governance.

5.4.3.4 CBAN Security Platform Service

5.4.3.4.1 Introduction to Security Platform Services

The CBAN Security Platform Service is a collection of security technologies that assess, reduce, protect, and manage security risks. These technologies are atomic in nature. This means that a category such as access management has to be included, since different types of access management solutions (e.g., MAC, DAC, ABAC, and RBAC) use different technologies, but all serve the same fundamental service. In contrast, solutions such as Zero Trust or SASE are NOT included as a CBAN Platform Service because both are constructed from other security related Platform Services.

The basic CBAN Security Services are shown in Figure 9.

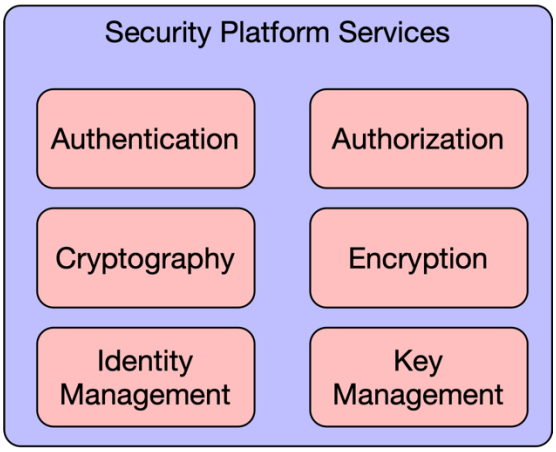

Figure 9 - Security Platform Services

5.4.3.4.2 CBAN Authentication Platform Service

Authentication is the process of verifying that a subject requesting to perform an operation on a target is who they claim to be. Policies may be used to dictate the set of verification criteria used. The CBAN Authentication Platform Service depends on the CBAN Namespace Platform Service and the CBAN Identity Platform Service.

[R21] An Authentication Platform Service **MUST** be implemented in the CBAN settlement platform.

[O9] Policies **MAY** be used to dictate the set of verification criteria used for authentication.

5.4.3.4.3 CBAN Authorization Platform Service

Authorization is the process that results in permitting or denying access to a target by a subject. Policies may be used to prescribe the criteria for the authorization decision. The CBAN Authorization Platform Service depends on the CBAN Namespace Platform Service and the CBAN Identity Platform Service.

[R22] An Authorization Platform Service **MUST** be implemented in the CBAN settlement Platform.

[O10] Policies **MAY** be used to prescribe the criteria for the authorization decision.

5.4.3.4.4 CBAN Cryptography Platform Service

Cryptography is the process of constructing and verifying protocols that prevent third parties from reading private communications. The CBAN Cryptography Platform Service depends on the CBAN Namespace Platform Service and the CBAN Identity Platform Service.

[R23] A Cryptography Platform Service **MUST** be implemented in the CBAN settlement Platform

5.4.3.4.5 CBAN Encryption Platform Service

Encryption is the process of encoding information using a key into an unintelligible form to protect sensitive information. The unintelligible form of the information must be decrypted using the key to recover the original information. The CBAN Authentication Platform Service depends on the CBAN Namespace Platform Service and the CBAN Identity Platform Service.

[R24] An Encryption Platform Service **MUST** be implemented in the CBAN settlement Platform.

5.4.3.4.6 CBAN Identity Management Platform Service

Identity Management defines access control based on the identity of an entity that initiates a particular set of operations on a target according to a set of criteria. The CBAN Identity Management Platform Service depends on the CBAN Namespace Platform Service and the CBAN Identity Platform Service.

[R25] An Identity-Management Platform Service **MUST** be implemented in the CBAN settlement platform.

Note: The *Identity Management Platform Service* and the *Identity Platform Service* are two distinct and different services. The *Identity Platform Service* defines how identities are being assigned, while the *Identity Management Platform Service* defines how access is managed based on an assigned identity.

5.4.3.4.7 CBAN Key Management Platform Service

Key management refers to management of cryptographic keys in a cryptosystem. This includes dealing with the generation, exchange, storage, use, destruction, and replacement of keys. It also includes cryptographic algorithm and protocol design. The CBAN Authentication Platform Service depends on the CBAN Namespace Platform Service and the CBAN Identity Platform Service.

[R26] A Key Management Platform Service **MUST** be implemented in the CBAN settlement platform.

5.4.3.5 CBAN Logging Platform Service

The CBAN Logging Service is a collection of technologies that enable different types of logs to be ingested and collected dynamically. The CBAN Logging Service may provide an optional

normalization service, which enables related logs generated by different sources using different technologies to be normalized into a single data model.

[R27] A Logging Platform Service **MUST** be implemented in the CBAN settlement platform.

[O11] When a Logging Platform Service is implemented it **MAY** also provide a normalization service.

5.4.3.6 CBAN Governance Platform Services

5.4.3.6.1 Introduction to Governance Platform Services

Governance Platform Services are a collection of rules and tools that control the behavior and function of the CBAN Settlement Platform. The implementation and enforcement of the rules is carried out using other Platform Services. The Governance Platform Services are depicted in Figure 10 herewith:

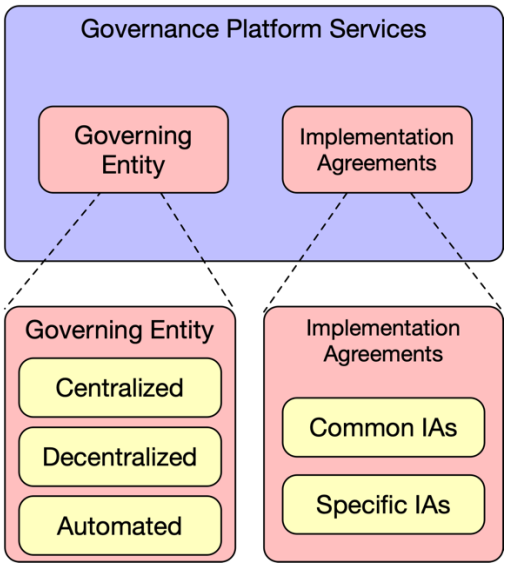

Figure 10 - Governance Platform Services

Governance is divided to two functions:

- **Implementation Agreements ("IAs")**: A collection of rules and agreements that describe how CBAN Services are implemented and control the behavior of the CBAN settlement platform. In a Category Charlie/Delta platform such agreements and rules are typically developed in a collaborative manner by the CBAN members. They would typically be prescribed by the developer in a Category Alpha and Category Bravo type platforms.

- **Governing Entity**: An entity that performs governance tasks by defining the rules and IAs, as well as ensuring compliance and resolving conflicts where needed. Governance also defines the methods by which the Governing Entity is established, its composition and the methods by which it defines/accepts rules/IAs and enforces compliance.

5.4.3.6.2 CBAN Implementation Agreements

Rules and agreements that describe how the CBAN Platform Services defined in this document are implemented. Such rules and agreements define the specific details and methods used to implement

the services. E.g., the choice of a specific DLT chain type or the acceptance criteria of entities to the CBAN platform.

Implementation Agreements are divided to three groups:

5.4.3.6.2.2 Common Implementation Agreements – Common Rules

Common IAs IAs are those that are used by all applications, users and entities involved with the CBAN settlement platform. As a result the rules stipulated in Common AIs are applicable to all. E.g., The CBAN settlement platform may require that all entities and applications use a specific Identity Service and specific security methods.

[R28] All applications, users and entities participating in the CBAN settlement platform **MUST** comply with all Common Implementation Agreements – Common Rules.

5.4.3.6.2.3 Common Implementation Agreements – Specific Rules

These types of IAs are common to all applications, users and entities participating in the CBAN settlement platform, but the specific rules are application or jurisdiction dependent. E.g., The CBAN settlement platform may require that all entities and applications use a specific Location Service, but different geographies may use different methods to define a location and different applications may require different granularity/accuracy of location information.

[R29] All applications, users and entities participating in the CBAN settlement platform **MUST** comply with all Common Implementation Agreements – Specific Rules.

[O12] The specific rules **MAY** vary depending on the specific user, application, and/or jurisdiction.

5.4.3.6.2.4 Specific Implementation Agreements

Those are IAs that are specific to one or more application or jurisdiction and only apply to entities subject to such jurisdiction and/or using such applications. E.g., all European entities are subject to GDPR thus any application operated by an entity subject to European jurisdiction must use an Implementation Agreement that complies with GDPR. Another example is a requirement that all applications involved with monetary transactions use a specific encryption method prescribed by the governance.

[D6] All applications, users and entities involved with the CBAN settlement platform **SHOULD** comply with all Specific Implementation Agreements.

[O13] Certain applications and entities **MAY** be exempt from compliance with Specific Implementation Agreements.

5.4.3.6.3 CBAN Governing Entity

5.4.3.6.3.1 Governing Entity types

The Governing Entity performs governance tasks by defining the rules and AIs, as well as ensuring compliance and resolving conflicts where needed. Governance also defines the methods by which the Governing Entity is established, its composition and the methods by which it defines/accepts rules/IAs and enforces compliance and the legally binding agreements that need to be signed by entities and individuals.

There are several types of Governing Entities:

5.4.3.6.3.2 Centralized governance

Centralized Governance is a scenario where the Governing Entity is chosen or agreed upon by the CBAN settlement platform participants.

[R30] The Governing Entity in a Centralized Governance scenario **MUST** be elected through consensus.

[D7] The Governing Entity in a Centralized Governance scenario **SHOULD** be an entity participating in the CBAN settlement platform.

[O14] The Governing Entity in a Centralized Governance scenario **MAY** be an external entity not participating in the CBAN Settlement Platform.

[O15] The Governing Entity in a Centralized Governance scenario **MAY** consist of more than a single entity.

[R31] When the Governing Entity in a Centralized Governance scenario consists of more than a single entity all its decisions **MUST** be reached through consensus between the entities the Governing Entity consists of.

5.4.3.6.3.3 Decentralised governance

Decentralized Governance is a scenario where the Governing Entity consists of all CBAN Settlement Platform participants or a group of representatives thereof. Governance tasks are performed using DLT consensus and policies.

[R32] The representatives participating in a Governing Entity in a Decentralized Governance scenario **MUST** be elected through consensus.

[R33] The Governing Entity in a Decentralized Governance scenario **MUST** use DLT consensus and Implementation Agreements to perform governance tasks.

[D8] The Governing Entities in a Decentralised Governance scenario **SHOULD** each be CBAN settlement platform participants.

5.4.3.6.3.4 Automated governance

Automated governance is a scenario where decisions, consensus and policy enforcement are taken by software (pre-programmed or Artificial Intelligence) on behalf of the CBAN settlement platform participants.

[R34] Changes to the governing software in an Automated governance scenario **MUST** be accepted through consensus.

5.4.3.6.3.5 Other types of governance

Additional governance structures may be formed in the future, and documented in a future version of the present document.

5.4.3.6.4 Crating, Changing and Enforcing Governance IAs and rules

Governance IAs and rules are created, maintained, changed and enforced by the Governing Entity using consensus. It is recommended that as many of the governance tasks as possible be handled automatically, but some tasks may require manual/human intervention.

[D9] Governance tasks **SHOULD** be performed automatically.

[O16] Governance tasks **MAY** be performed manually.

[R35] Any formal or official communication between DLT participants **MUST** be routed through, monitored, and recorded by the governance.

The above requirement can be fulfilled by storing all such communications on-chain in a manner readable by the governance.

5.4.3.7 CBAN Composition Platform Service

The CBAN Composition Platform Service defines which *subject* entities can compose new objects and how such new objects are composed from other objects.

The requirements related to composition of different object types are listed in the below sections:

- 5.4.3.19.7 Resource Composition
- 5.4.3.20.7 Platform Service Composition
- 5.4.3.21.2 Application Composition

5.4.3.8 CBAN Access Control Platform Service

CBAN Access Control Service defines which *subject* entities can perform which operations on which set of *target* entities according to a set of criteria.

There are four established Access Control Policies in use today: [i.3];

- MAC (Mandatory Access Control)
- DAC (Discretionary Access Control)
- RBAC (Role Based Access Control)
- ABAC (Attribute Based Access Control)

It is beyond the scope of this document to go into discussion of the differences between those policies.

A Policy is a set of rules that is used to manage and control the changing and/or maintaining of the state of one or more managed objects. An Access Control Policy defines the privileges and permissions of a subject entity to perform operations on a set of target entities. Policy Based Access Control (PBAC) defines the type of Policies required to implement the appropriate type of access control methodology according to the needs of the PDL Platform.

[R36] CBAN Access Control Services **MUST** use Access Control Policies to manage access control for the entities that it protects.

[O17] CBAN Access Control Services **MAY** use Policies to extend the functionality of standardized Access Control approaches.

5.4.3.9 CBAN Fault Tolerance Platform Service

The CBAN Fault Tolerance Service defines how a PDL Platform behaves in two scenarios:

- ***Faulty Instructions***: How *target* entities handle situations where faulty instructions are being given by *subject* entities according to a set of criteria. Faulty instructions may include, but are not limited to, violation of consensus, violation of security protocol, violation of policy. Faulty instructions may also result from incorrect or inaccurate data ingestion (e.g., a clock giving inaccurate readings, or a communications error due to congestion). It is highly recommended, though not mandated, that the CBAN platform establishes processes, automated or manual, to overcome such faults while retaining the platform's integrity.
- ***Faulty Components***: The ability of the CBAN settlement platform to continue operating correctly when one or more of its components fails.

[D10] Reasonable measures **SHOULD** be taken to allow the CBAN settlement platform to continue operations in presence of a level of faults that is below a pre-defined threshold.

[R37] Measures **MUST** be taken to notify entities using the CBAN settlement platform when faults exceed a pre-defined threshold.

5.4.3.10 CBAN Distribution Transparency Platform Service

CBAN Distribution Transparency Service defines how *subject* entities maintain transparency when distributing information to *target* entities. Transparency is defined by Policy and may vary depending on location, regulation, and application. E.g., A commercial agreement between two entities may include confidential commercial information that should not be visible to other parties; yet those other parties may need to be aware that a commercial agreement exists between said entities. Under such a scenario the Distribution Transparency service will have to ensure the details of the involved parties are visible to other parties, while the confidential parts of the agreement are encrypted and cannot be read.

[R38] The CBAN settlement platform **MUST** maintain Distribution Transparency in accordance with all applicable Policies.

5.4.3.11 CBAN Publish and Subscribe Platform Service

The CBAN Publish and Subscribe Platform Service defines how entities publish services and subscribe to Platform Services. This service is optional but will become a mandatory service in the event that entities need to publish services and/or subscribe to Platform services.

[D11] Entities publishing services and subscribing to Platform Services **SHOULD** use the Publish and Subscribe Platform Service.

[R39] When entities need to publish services and/or subscribe to CBAN Platform Services the CBAN settlement platform **MUST** make such service available.

5.4.3.12 CBAN Concurrency Platform Service

The CBAN Concurrency Platform Service defines how entities handle concurrency. Concurrency is the occurrence of different instances of events at the same time. Such events may or may not be dependent on each other. Concurrency may be allowed, banned or subject to certain restrictions depending on Policy and use case. An example for a banned concurrency would be for two entities adding a block to a DLT at the same time (which will create a fork). An example of an allowed concurrency may be collection of information from multiple sources at the same time and writing such information to a table in a certain order (e.g., alphabetical, ascending) as prescribed by Policy.

[R40] The CBAN settlement platform **MUST** implement a Concurrency Service based on Policy.

5.4.3.13 CBAN Storage related services

5.4.3.13.1 Types of Storage Platform Services

The CBAN Platform Storage related Platform Services provide different types of storage services for the CBAN settlement platform.

Note: The definition of "Directly Connected Storage" in the context of this section is that the storage is local to the node or is external storage that is managed by the owner of that node. It includes internal RAM, internal file system, external drive, NAS and Cloud storage services. It is not limited to storage that is physically connected to a node.

5.4.3.13.2 CBAN In Memory Storage Platform Service

In Memory Storage is any data that is stored in the RAM (or RAM swap space on a local disc) of the computer running an application. Such storage is typically used for confidential information.

There are two types of In-Memory Storage options:

- Volatile - the contents of such storage must not be recorded anywhere and must not survive a restart of the computer or application.

  [R41] Volatile In Memory Storage **MUST NOT** keep a copy of the RAM contents on a disc or any other sort of non-volatile memory.

  [R42] If a node uses RAM Swap Space for Volatile Storage the contents of such storage **MUST** be erased when the node or application restarts.

- Non-Volatile - the contents of such storage may be recorded on a local disc or non-volatile memory and may survive a restart of the application or computer.

  [O18] Non-Volatile Storage **MAY** survive a node/application restart.

5.4.3.13.3 CBAN File System Storage Platform Service

Any storage on a directly connected storage such as a local disc, an external drive, a NAS or cloud storage.

[R43] CBAN nodes **MUST** support directly connected storage.

[O19] Devices used to access or run a CBAN compliant application **MAY** support directly connected storage.

[R44] Access to NAS and Cloud storage **MUST** follow all security and access policies prescribed by the governance.

5.4.3.13.4 CBAN On-Chain Storage Platform Service

On-Chain storage is any application data that is stored in blocks on all nodes using the chain. Each block in a chain is numbered and is identical to the respective blocks on all nodes using the chain. Data is stored locally on the node or on an external storage managed by the owner of the node.

[R45] Each On-Chain block **MUST** have a unique number.

[R46] Each On-Chain block **MUST** be identical to all other blocks carrying that unique number on other nodes participating in the chain.

[D12] A node **SHOULD** store the On-Chain blocks on directly connected storage that is physically connected to the node.

[O20] A node **MAY** store On-Chain blocks on directly connected storage that is not physically connected to the node as long as it is managed by the owner of the node and follows all security and access policies prescribed by the governance.

[R47] On-Chain storage **MUST** be secured according to the security policy defined by the governance.

5.4.3.13.5 CBAN Off-Chain Storage Platform Service

Off-chain data storage is the storing of information in a digital, machine-readable medium that is not stored on the main chain. The main differentiator between On-Chain and Off-Chain storage is that Off-Chain storage is not loaded as a block to the main chain used by all nodes.

Off-chain storage is a key enabler to scale blockchain-based applications that are data-intensive and/or data sensitive. It is often used to store non-transactional data that is too large to be stored in the blockchain efficiently or requires the ability to be changed or deleted. Off-Chain data is typically only accessible by a subset of the nodes participating in a chain.

There are two types of off-chain storage:

1. Distributed Addressable Storage (e.g. IFPS), which is content that can be accessed through a link (URL). Such URL may be loaded into the main chain thus such off-chain storage is accessible by all nodes even though it is not loaded to the main chain. Addressable storage may be distributed (e.g., stored on a distributed ledger, with or without blockchain) which is then called “Distributed Addressable Storage”.
2. Non-Addressable Storage, which is content that cannot be addressed and accessed by any other entity except for the entity that directly manages this data.

[R48] Off-Chain data **MUST** be accessible to the node to which it is directly connected.

[R49] Addressable Storage Off-Chain data **MUST** be accessible by any other node meeting the access control policies defined by the owner of the node to which it is directly connected.

[O21] Addressable Storage Off-Chain data **MAY** be accessible by any other node meeting the access control policies defined by the governance.

[O22] Off-Chain data **MAY** be stored on a sidechain.

5.4.3.13.6 CBAN Distributed Blockchain Storage Platform Service

Preface: The concepts of Distribution, blocks and a chain are not necessarily interdependent. Data may be stored on a single location or may be distributed, regardless of the type of data (blockchain or other). On the other hand, blocks can be chained (using hashes or other methods) and stored locally on a non-distributed ledger.

The definitions in this section apply to the specific case of a distributed blockchain ledger, that is - scenarios where the blockchain is stored in a distributed ledger. Such scenarios also include provisions to ensure integrity of the data across the distributed nodes.

The Distributed Blockchain Storage Platform Service is inherent to the DLT. Each DLT (CBAN) node stores the exact same copy of the chain as all other nodes in a DLT. However – situations may arise where certain nodes add invalid blocks thus invalidating the entire chain. Those are considered temporary events and the governance and consensus mechanisms offer ways to identify such invalid blocks/chains and to take actions to eliminate the problem. The methods by which such events are treated, and such situations are resolved vary by DLT type, consensus mechanism and governance and are beyond the scope of this document.

[R50] Each node **MUST** store the exact same chain as all other nodes on a DLT.

[R51] When a node detects an anomaly or a discrepancy between the chain stored on it and the consensus-driven chain stored on other nodes it **MUST** flag its chain as "invalid" and replicate the entire chain, or the invalid parts of the chain, from a valid node holding a valid chain.

[R52] Upon replication of a valid chain from a valid node the node **MUST** recalculate the hashes to ensure validity of the new blocks and upon successful recalculation it may remove the "invalid" flag.

Due to the structure of the chain as linked blocks, the validation of an invalid chain can be achieved by only replacing the blocks that include and follow the invalid block and any subsequent block or blocks added to the chain afterwards. Thus, it is not required that the entire chain is replaced.

5.4.3.14 CBAN Modelling related Platform Services

5.4.3.14.1 Introduction to Modelling

The CBAN Modeling Platform Services define part of the common vocabulary and concepts for the CBAN settlement platform. Those are:

- Information Model
- Data Models

Modelling Platform Services also define services that are required for using a model as a common vocabulary:

- Model Search

- Model Stitching

The CBAN Modeling Platform Services consist of services that are fundamental for building more powerful CBAN compliant DLT Services as well as distributing the CBAN compliant platform. The CBAN Information Model and the CBAN Data Model Platform Services are mandatory.

[R53] The CBAN Information Model Platform Service **MUST** be implemented on the CBAN settlement platform.

[R54] The CBAN Data Model Platform Service **MUST** be implemented on the CBAN settlement platform.

5.4.3.14.2 The CBAN Information Model

An information model represents concepts of interest to the CBAN management environment in a *technology-neutral* form. An information model is a template, and is not meant to be instantiated. Rather, data models derived from it are instantiated.

[R55] The CBAN settlement platform **MUST** use a single information model to represent managed objects.

To accommodate future changes and applications the CBAN Information Model should be designed in an extensible manner so additional modules could be added to it to facilitate such applications.

[D13] The CBAN settlement platform **SHOULD** use a modular and extensible information model.

For example, an information model would define generic objects such as location, entities, ownership, device types, functionalities and other high-level concepts. This information model could then be used for an abstract description of applications from different ICT disciplines: IoT, Mobile, Connectivity, Compute etc.

[D14] The CBAN Information model **SHOULD** be specified as a standard in a formal document issued by CBAN.

5.4.3.14.3 CBAN Data Models

A CBAN data model represents applicable concepts in a CBAN compliant implementation in a *technology-specific concrete* form. A CBAN compliant implementation may require multiple data models that represent objects using different repositories, protocols, and data formats. Data Models represent Application specific, Lifecycle-step specific and Product specific implementations and are derived from the respective parts of the Information Model. Since all Managed Objects require at least one CBAN Software Interface through which they can be managed, and all Software Interface communications require a Data Model that defines the representation of data exchanged through such interface, each Managed object must have at least one Software Interface and at least one Data Model implemented through that interface.

[D15] Every Managed Object **SHOULD** be represented in at least one Data Model.

[O23] Every Managed Object **MAY** be represented in two or more Data Models.

[D16] CBAN implementations **SHOULD** associate the software interfaces provided by a particular IRP with a set of class methods of a Data Model.

NOTE: Class methods may be invoked directly (e.g., using code) or indirectly (e.g., using APIs or DSLs).

[O24] CBAN Data Models **MAY** be Application Specific.

Each Data Model is derived from the CBAN Information Model, which facilitates reconciling these different representations of the same concept into a single object. There are currently three common practices to structuring an Information Model:

- **Bottom-Up**: Construction of an Information Model from multiple Data Models through iterative, consensus based, manual processes.
- **Top-Down**: Design of a high-level, abstract, and modular, Information Model in a manner that allows adding up content and capabilities as required.
- **Iterative Development**: use of bottom-up and top-down methods iteratively to produce the information model.

These processes are typically performed through industry standard body meetings to construct an accepted Information Model. This process may be automated in the future through Model-Driven Engineering where an application (possibly embedded into the DLT) derives data models from the information model by crossing it with the specific object and functionality.

[R56] Data Models used in the CBAN settlement platform **MUST** be derived from the CBAN Information Model.

[O25] Data models used in the CBAN settlement platform **MAY** be derived automatically by the CBAN settlement platform.

[D17] Data models **SHOULD** be specified as a standard in a formal document issued by CBAN.

[O26] CBAN **MAY** adopt Data Models developed by other fora or SDOs.

[R57] Data Models Adopted from other fora or SDOs **SHALL** be compliant with the CBAN Information Model.

[R58] Data Models Adopted from other fora or SDOs **SHALL** be formally approved by CBAN.

5.4.3.14.4 CBAN Model Search

Model search is the functionality that allows a developer or an application to search for specific or generic models within existing information and data models. Such search functionality may assist a developer or an application in deciding what needs to be added to a model in order to support certain applications or application functionalities.

[R59] The CBAN model search functionality **MUST** have full visibility to the Information Model and all Data Models in use by the CBAN settlement platform.

[D18] The CBAN model search functionality **SHOULD** provide a human-readable response to queries based on keywords and text snippets.

[O27] The CBAN model search functionality **MAY** offer a GUI based search in a model tree representation.

[D19] The CBAN model search functionality **SHOULD** provide API access to queries made through external search engines.

[R60] API access by external search engines **MUST** be controlled by the governance.

5.4.3.14.5 CBAN Model Stitching

Model stitching is the functionality that enables integrating multiple models or parts of models into a single model. The resulting model must be duplicate-free so in the event that two models or parts of models each include the same objects or relations between objects – such duplicates are removed. In the event that the models or parts thereof include objects of the same name that offer different purposes or relations, the resulting model must separate those to unique objects with unique names and relations.

[R61] Stitched models **MUST NOT** include duplicate attributes.

[R62] Each attribute **MUST** be unique in a Stitched model.

5.4.3.15 CBAN Topology Platform Service

The Topology Platform Service allows a node to identify other nodes on the DLT and, depending on consensus mechanism, identify which nodes to communicate with when performing DLT related tasks such as consensus and block replication. The number of nodes with which a node should communicate depends on the consensus mechanism, total number of nodes, number of valid nodes and governance.

[R63] The Topology Service **MUST** publish the status of the node and the chain to all other nodes in the CBAN settlement platform.

[D20] The Topology Service **SHOULD** maintain the status of a sufficient number of nodes as required by the governance.

[O28] The number of nodes required to perform distributed DLT tasks **MAY** vary with time depending on the number of valid nodes at any given moment.

The discovery process through which the Topology Service identifies other nodes depends on the specific governance and DLT type and is out of scope of this section.

5.4.3.16 CBAN Event Processing Platform Service

The Event Processing Platform Service processes events as they occur.

Such events are broken to different categories, and are presented here from the perspective of a specific node in the CBAN settlement platform:

1. Events that occurred locally on a specific node and do not affect other nodes nor do they affect the chain or consensus mechanism. E.g., a user had logged in to the node or a backup of data was initiated. Such events are defined as *insignificant* for the proper function of the CBAN settlement platform.

2. Events that occurred locally on a specific node and may affect other nodes or the behavior of the chain, including the consensus mechanism. E.g., the storage device is reporting errors, CPU usage has reached a threshold, CPU is overheating, a block is validated, a block is invalid thus the node is flagged "invalid". Such events are defined as *significant* for the proper function of the CBAN settlement platform.

3. Events that occurred on other nodes and may affect the chain or the consensus mechanism. E.g., a block is validated, a block is invalid, a specific node is flagged as "invalid", a specific node is in jeopardy (storage errors, CPU overheat, etc.). Such events are also defined as *significant* for the proper function of the CBAN settlement platform.

[D21] The Event Processing Platform Service **SHOULD** collect platform wide events.

[R64] The Event Processing Platform Service **MUST** store the events On-Chain.

[R65] The Event Processing Platform Service **MUST** process all Significant Events.

[O29] The Event Processing Platform Service **MAY** process Insignificant Events.

[R66] The Event Processing Platform Service **MUST** notify the governance when Significant Events have caused a change to consensus operations.

[R67] CBAN settlement platform nodes **MUST** follow governance on behavior upon occurrence of Significant Events.

[O30] The Event Processing Platform **MAY** trigger actions independent of the governance, upon occurrence of certain Significant Events, based on smart contracts or prescribed lists of actions.

[R68] The Event Processing Platform Service **MUST** trigger actions when specific events occur based on a prescribed list of actions.

[O31] The Event Processing Platform Service **MAY** use Artificial Intelligence when triggering actions based on events.

5.4.3.17 CBAN Distributed Data Collection Platform Service

The CBAN Distributed Data Collection Platform Service performs tasks related to collection of data. Data collection is defined in the following matrix:

1. Internal / External data.
2. Synchronized / Asynchronized data.

Internal data is data that is generated by a node either through calculation or through a directly connected system (e.g., a voice switch generating CDRs) that feeds data to a specific node.

External data is data obtained from external resources or systems. E.g., ForEx rates obtained from a bank.

Synchronized data is data that needs to be stored in synchronization with other data and thus requires sequencing and has dependency on timing or content of other data being collected. The

methods used to ratify the integrity of synchronized data may vary depending on DLT and governance.

Asynchronized data is data that does not require synchronization and does not have dependency on other data and other data does not depend on it.

[R69] The governance **MUST** define the level of synchronization required for data.

[O32] The level of synchronization **MAY** vary depending on application and type of data.

[R70] The governance **MUST** define the method by which data is synchronized and the method of assuring synchronization meets such criteria.

[D22] The application and governance **SHOULD** define the type of data that needs to be collected and the duration it should be stored.

5.4.3.18 CBAN Distributed Secret Sharing Platform Service

Preface: Secret Sharing is sharing of confidential data between nodes in a manner that maintains confidentiality of the data. The method by which data remains confidential (e.g., encryption) is out of scope of this document.

[O33] A DLT used by the CBAN settlement platform **MAY** offer a Distributed Secret Sharing service.

[R71] When a Distributed Secret Sharing service is available it **MUST** meet the confidentiality requirements defined by the governance.

[D23] The data shared through Secret Sharing does **NOT HAVE** to be stored on the chain.

5.4.3.19 Resource Management Platform Services

5.4.3.19.1 Introduction to Resource Management

The CBAN Resource Management Platform Services consist of a set of platform services that enable resources to be discovered, administered, managed, inventoried, composed and virtualized.

5.4.3.19.2 Resource Discovery Platform Service

The Resource Discovery Platform Service provide means to discover resources available to CBAN Platform applications and nodes.

For example, an application can discover resources available through platform services. Such resources can be computation power, storage space, connectivity between certain locations, sensor or other equipment availability at certain locations.

[R72] CBAN Platform resources **SHOULD** be discoverable.

[R73] The CBAN Resource Discovery Platform Service **MUST** support discovery of CBAN Platform resources.

5.4.3.19.3 Resource Virtualization Platform Service

Resource Virtualization is the act of creating a virtual resource that mimics the behavior of a physical resource. A virtual resource is constructed through software based on one or more physical devices. Such device(s) can be used to create one or more virtual resources that can be used by services and applications.

[R74] The CBAN Platform **MUST** allow use of virtual resources.

[O34] A CBAN Platform Virtual Resource **MAY** be constructed using more than one physical resource.

[R75] A CBAN Platform Virtual Resource **MUST** offer functionality that complies with the specifications of such resource.

5.4.3.19.4 Resource Inventory Management Platform Service

5.4.3.19.4.1 Categories of Resource Inventory management

Resource Inventory Management is divided to two categories:

- Node Specific Resources
- Platform resources

[R76] The node management **MUST** keep track of all resources available on that node, both those available to all platform users and those that are node specific.

5.4.3.19.4.2 Node-specific resources

Node specific resources are only available for use of the node on which such resource is installed. E.g. directly connected storage that is not addressable.

[R77] Node-specific resources **MUST** be discoverable and usable only by applications, services, and users of the specific node on which the resource is installed.

5.4.3.19.4.3 Platform resources

Platform resources are available to all nodes, services, applications, and users regardless of the node(s) on which it is implemented or installed. E.g., an addressable IFPS storage or a network printer or a sensor.

[R78] The governance **MUST** keep track of inventory and serviceability of all Platform resources.

[D24] Platform resources **SHOULD** be available to all Services, Applications, and users on any node.

[O35] Platform resources **MAY** be only available to select Services, Applications, and users on specific nodes.

5.4.3.19.5 Resource Administration and Management Platform Service

Resource administration and management is an integral part of any platform. As described in previous sections, in a distributed platform resources may be associated with specific nodes, be accessible by specific nodes or be available and accessible by multiple, possibly all, nodes participating in the platform.

[R79] The CBAN Platform **MUST** provide means to manage and administer Platform Resources.

[O36] The CBAN Platform **MAY** provide means to manage and administer Node-specific resources.

[R80] The CBAN Platform Resource administration and management service **MUST** retain the exclusivity of Node-specific resources.

[R81] An CBAN node **MUST** provide means to manage and administer Node-specific Resources.

5.4.3.19.6 Resource FCAPS

FCAPS is an acronym for *fault, configuration, accounting, performance, security,* which are the management tasks defined by the ISO model.

[D25] The CBAN Platform **SHOULD** offer FCAPS in accordance with this document.

[R82] CBAN Platform resources **MUST** support FCAPS functionality.

5.4.3.19.7 Resource Composition

Resources may be composed from other resources. Such resources are referred to as Composite Resources. As such their management and administration should follow the hierarchy of resources so that usage of a composite resource will mark the resources it is composed of as being in use. The same applies to all FCAPS functions.

[O37] CBAN Platform resources **MAY** be composed from other resources.

[R83] Composite CBAN Platform resources **MUST** support FCAPS functionality.

5.4.3.20 CBAN Platform Service Management services

5.4.3.20.1 Introduction to Platform Service Management

The CBAN Platform Service Management services define how to discover, administer, and manage Platform Services for the CBAN platform. They consist of a set of platform services that enable services to be discovered, administered, managed, inventoried, and virtualized. FCAPS operations, as well as the ability to compose new Services from existing CBAN Platform Services, are also included.

5.4.3.20.2 Platform Service Discovery Platform Service

The Platform Service Discovery Platform Service provides means to discover Platform Services available to the CBAN Platform applications and nodes.

[R84] A CBAN Platform service **SHOULD** be discoverable.

[R85] The CBAN Service Discovery Platform Service **MUST** support discovery of CBAN Platform Services.

5.4.3.20.3 Platform Service Virtualization

In essence all services implemented through software can be considered as being virtual as there is no dedicated machine performing a service and it is done through code running on a processor and using resources of the node it is running on. For the purpose of this document Platform Service Virtualization is the act of creating a Platform Service using virtual resources. A Virtual Platform Service is constructed through software based on one or more virtual resources.

[R86] The CBAN Platform **MUST** allow use of Virtual Platform Services.

[O38] A CBAN Platform Virtual Platform Service **MAY** be constructed using one or more virtual resource.

[O39] A CBAN Platform Virtual Platform Service **MAY** be constructed using two or more other services where at least one of which is virtual.

Note: When a composite service is constructed of multiple objects (e.g., Resources, Services), none of which being virtual, it is considered a regular Composite Platform Service, not a Virtual Platform Service.

[R87] A CBAN Platform Virtual Platform Service **MUST** offer functionality that complies with the specifications of such service.

5.4.3.20.4 Platform Service Inventory Management

The CBAN settlement platform manages inventory of Platform Services through the governance that keeps track of inventory and serviceability of all Platform Services and manages availability of such Platform Services to applications and users.

[D26] The governance **MUST** keep track of inventory and serviceability of all Platform services.

[D27] Platform services **SHOULD** be available to all applications and users on any node.

[O40] Platform services **MAY** be only available to select applications and users on specific nodes.

5.4.3.20.5 Platform Service Administration and Management

Service administration and management is an integral part of any platform. In the CBAN settlement platform, management tasks are handled through governance.

[R88] The CBAN Platform **MUST** provide means to manage and administer Platform services.

[R89] All Platform Services implemented on the CBAN settlement platform **MUST** be authorized by the governance.

[R90] Users **MUST NOT** implement Platform Services without permission from the governance.

5.4.3.20.6 Platform Service FCAPS

FCAPS is an acronym for *fault, configuration, accounting, performance, security,* which are the management tasks defined by the ISO model.

[R91] CBAN Platform Services **MUST** support FCAPS functionality.

5.4.3.20.7 Platform Service Composition

Platform Services may be composed from other Platform Services. Such Platform Services are referred to as Composite Platform Services. As such their management and administration should follow the hierarchy of Platform Services so that usage of a Composite Platform Service will mark the resources used by the Platform Services it is composed of as being in use. The same applies to all FCAPS functions.

[O41] CBAN Platform services **MAY** be composed from other services.

[R92] Composite CBAN Platform services **MUST** support FCAPS functionality.

5.4.3.21 CBAN Application Management Services

5.4.3.21.1 Introduction to Application Management

The CBAN Application Management Services are a set of Platform Services that enable Platform Services to be composed and orchestrated into an application. In addition, resources that are used to support such Platform Services can also be orchestrated.

5.4.3.21.2 Application Composition

As described earlier in this document, service composition is the act of composing a Platform Service from two or more other Platform Services. When Applications are concerned – they too can be composed of other applications or re-use other applications. E.g., A Video Conference scheduling application can be composed of a video conferencing application and a calendar application combined such that the application user can tailor the timing of distribution. An application may also be composed of a mix of applications and services.

[R93] The CBAN platform **MUST** support Application and Service composition.

[R94] The Governance **MUST** provide Application, Service and Resource management services to composite applications.

5.4.3.21.3 Application and Platform Service Orchestration

When objects (Applications and Platform Services) are combined into a composite object there may be a need to orchestrate the construction and behaviour of the composite object. E.g., When an

application of which a composite object is constructed is using the output of another application as its input, the applications should be orchestrated such that the data traverses the applications in the right sequence.

[R95] The governance **MUST** ensure data traverses the applications and services of which a composite object is constructed in the appropriate sequence as designed by the developer of the composite object.

5.4.3.21.4 Orchestration Platform Service

Composite objects require Orchestration in order to become such. Orchestration is the act of chaining the objects in a manner that connects the respective ingress and egress interfaces of such objects in a topology and sequence that yields the required functionality. The Orchestration service provides the necessary management tools to chain the objects, publish them and manage their composite operation.

[R96] The Orchestration Platform Service **MUST** chain objects as defined and designed by the governance.

[R97] The Orchestration Service **MUST** apply all Resource, Service and Application management requirements as defined in the previous sections on the resulting composite object.

5.4.3.21.5 Application Registration

Application registration is a functionality that registers and lists all applications operated on the CBAN settlement platform.

[R98] The CBAN platform **MUST** maintain a list of all applications registered and operated on it.

5.4.3.22 CBAN Transaction Management Service

Transaction Management Service (TMS) facilitates transaction related interactions between applications/services and underlying DLT networks by providing the following functionalities:

1) Configure transaction-related policy rules for and/or to applications/services;
2) Receive and authenticate transaction-related requests (e.g. a request for creating a transaction) from applications/services;
3) Select an appropriate underlying DLT network for applications/services in scenarios where such a selection exists (e.g., Category “Charlie” platform);
4) Interact with underlying DLT networks and/or external storage on behalf of applications and services, to retrieve and send transaction-related requests and data to and from applications and services; This may include:
    - retrieve transaction-related data from external data sources for applications/services;
    - receive responses from underlying DLT networks and/or external storage; and
    - process and forward the responses to applications and services.

[R99] The CBAN Transaction Management Service **MUST** authenticate, process and manage incoming transaction operations from applications and services.

[R100] The CBAN Transaction Management Service **MUST** interact with designated underlying DLT networks and/or external storage to fulfil transaction operations on behalf of applications and services.

[R101] The CBAN Transaction Management Service **MUST** process responses for transaction operations as received from designated underlying DLT networks and/or external storage and forward them to applications and services.

[O42] The CBAN Transaction Management Service **MAY** configure transaction-related policy rules for applications and services.

[O43] In a platform that handles more than one DLT network, the CBAN Transaction Management Service **MAY** select an underlying DLT network to be used for transactional purposes by applications and services.

[O44] In a platform that handles or has access to external storage, the CBAN Transaction Management Service **MAY** handle transaction related activities using such external storage for applications and services.

5.4.3.23 CBAN Data Model Gateway/Broker Platform Service

5.4.3.23.1 Introduction to presentation services

In general, each product or functional block has its own Data Model, and possibly one or more interfaces through which it exchanges data with other entities. When different entities that use different Data Models need to exchange information a Data Model Broker, also called a Data Model Gateway, is required in order to ensure information is exchanged correctly. Such Broker/Gateway is software that mediates between two systems with different data models, yet enabling the two different systems to communicate transparently with each other. There are many benefits of using Data Model Brokers, including error reduction via software transmitting data instead of humans and business process automation through automated transfer of data between applications. Data Model Brokers also enable custom applications that integrate different application data.

The purpose of the Data Model Broker/Gateway is to:

- translate data communicated from an external system/entity into a normalized form that all CBAN platform Functional Blocks can understand, and
- translate recommendations and commands from the normalized form of an CBAN platform to a form that the external system/entity can understand, and
- manage authentication and authorization of the entities that want to communicate with the CBAN platform.

As discussed earlier in this document, APIs are the most common method of data exchange between entities and functional blocks, hence an implementation of a Data Model Broker/Gateway in an environment where APIs are in use will be in the form of an API Broker/Gateway.

The CBAN settlement platform API Broker ingests APIs through an appropriate Reference Point, analyses the API, and then routes the functionality of the ingested API to an appropriate CBAN Platform Functional Block. Similarly, APIs sent to external clients are sent to the API Broker, which routes the functionality of the API to the appropriate client.

Alternative information exchange methods exist and may be used between the CBAN platform and external entities. One such example is the use of "hot-folders" which are IFPS where data can be exchanged by uploading/downloading data files by multiple entities using a file transfer method such as SFTP. The use of such alternative data transfer methods may still require brokering between data models/formats. While "hot folders" may be easier and faster to implement than APIs they are typically less secure than APIs and brokering data models using such "hot folders" is more complex to implement than an API Broker/Gateway. The respective CBAN settlement platform parties may choose an implementation that meets their requirements.

[R102] The CBAN platform **MUST** exchange data with external entities and users through a Data Model Broker/Gateway.

5.4.3.23.2 CBAN API Presentation Platform Service

5.4.3.23.2.1 Introduction to APIs

An API is a set of communication protocols, code, and tools that enable one set of software components to interact with either a human or a different set of software components. APIs are critical for platform and ecosystem development. Effective API programs lay the foundations for digital transformation by enabling organizations to build a platform and develop an ecosystem.

5.4.3.23.2.2 RESTful-APIs

A REST API (also known as RESTful API) is an API that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for *representational state transfer*. The definition of REST and REST compliance is beyond the scope of this document.

5.4.2.21.1.2 Non-RESTful-APIs

APIs that do not conform to the REST architecture style are considered Non-RESTful.

[D28] The CBAN platform and platform-services **SHOULD** use RESTful-APIs.

[O45] CBAN APIs **MAY** be non-RESTful.

5.4.3.24 CBAN Application Specific Services

As their name implies, Application-Specific Services serve a specific application or a group of applications that require this specific service but is not required by all applications operated using the platform. The characteristics of an Application-Specific service are that:

1) It is only used by Applications and cannot be used by other Platform Services.
2) It can be implemented as part of an Application rather than as a Platform Service.

[R103] An Application Specific Service **MUST NOT** be used by other Platform Services.

[O46] An Application Specific Service **MAY** use other Platform Services.

[O47] An Application Specific Service **MAY** be implemented as part of an Application.

Due to their circumstantial nature the present document does not list such specific services. During the evolution of the CBAN settlement platform - some application developers may consider moving certain functionalities from the applications they are developing to the platform thus making them available to other applications to use.

5.5 CBAN Application Clients

5.5.1 Introduction to Application Clients

Application clients are the user front end that allows the user to interface with an application and perform application tasks such as initiating a transaction, performing a query, participating in a consensus vote, etc.
There may be multiple application clients to the same application, which differ in terms of the hardware and software implementation of the underlying device being used for the application client. Some of the more common types of application clients are listed herewith:

5.5.2 CBAN Computer Applications

Computer applications are running on a personal computer. There are multiple types of personal computers in common use today, notably the Microsoft Windows enabled PC (typically using an Intel or similar processor), the Apple Mac (which runs on both Intel and Apple processors), Linux, the Chrome devices and numerous others.

An application must be tailored to the specific hardware and operating system that personal computer is using and offer a uniform interface to the user regardless of that operating system.

[D29] A CBAN Computer Application **SHOULD** offer functionality in a manner agnostic to the underlying hardware and operating system of the computer it is implemented on.

[O48] A CBAN Computer Application **MAY** require specific minimum hardware and software configurations to function properly.

[O49] A computer running a CBAN Application **MAY** also be used as a network node.

5.5.3 CBAN Mobile Device Application

Mobile Applications run on mobile devices such as smartphones and tablets. Such mobile devices may run on various hardware types and vary in terms of operating system, screen size, computation power and internal architecture. While there is a challenge to maintain software compatibility with multiple mobile environments the benefit is the wide-spread adoption and availability of such devices which makes the application available to larger audiences. An additional benefit is the portability of mobile devices that makes a mobile device application available in locations where a personal computer cannot be operated or cannot be connected to the network.

Since mobile devices may be limited in resources, computation power and storage space, and would typically not have directly connected storage, they would not typically be used as network nodes. Furthermore, the application interface may lack some functionality that may only be available on other application types.

[R104] A CBAN Mobile Device Application **MUST** offer sufficient functionality as required by the DLT application to perform tasks required by its user.

[O50] A CBAN Mobile Device Application **MAY** offer reduced functionality compared to other Device application types.

5.5.4 CBAN Cloud Applications

Cloud Applications run on a network-based machine (typically a virtual machine) and are accessible through the Public Internet, through private networks or through Intranets, depending on the network environment implementation. Such applications would typically be accessed through an HTML GUI (e.g., Web browser) using HTTP or HTTPS or other mark-up languages as used by the respective developers. The GUI would typically offer access to most, or all, application features.
The GUI implementation may vary depending on the Web browser used and the operating system of the device used to run such web browser.

[D30] A CBAN Cloud application **SHOULD** be compatible with all commonly used Web Browsers on all commonly used operating systems as prescribed by the governance.

The definition of "commonly used" in the context of this requirement may vary with time, and is beyond the scope of this document, thus this document does not list the specifics and leaves such a decision to the governance and developers of each specific application or platform.

[R105] A CBAN Cloud Application **MUST** use a secure connection (e.g., HTTPS) to the web client.

[R106] A CBAN Cloud Application **MUST** be compatible with a list of web browsers and operating system environments defined by the governance for each such application.

5.6 Summary

The CBAN Normative Reference Architecture defined in this document offers an abstract architecture and an extensive list of Platform Services. To realize the CBAN settlement platform the specifics must be designed and defined through implementation agreements and through adoption of existing implementations of services that can be made compliant with the requirements set forth in this document. While being high level and abstract, this document is all encompassing in the sense that it includes services that may be required in a very large list of use cases and environments. It is designed as general guidelines to be followed when defining the specifics, yet keeps the door open for future expansion, without prescribing the specifics for one use-case or another.

Annex (informative): Change History

Date	Version	Information about changes
March 2022	1.0	Initial Release

History

Document history		
V.0.0.1	20211130	Completed editing of first draft
V.1.0	20220321	Completed final draft for board approval and ratification