INTERNATIONAL MONETARY FUND

ASAP: A Conceptual Model for Digital Asset Platforms

Victor Budau and Herve Tourpe

WP/24/19

2024
FEB

WORKING PAPER

 WP/24/19

IMF Working Paper
IT Department

ASAP: A Conceptual Model for Digital Asset Platforms
Prepared by Victor Budau and Herve Tourpe*

Authorized for distribution by Pierre Passin
February 2024

ABSTRACT: This working paper inaugurates the "Technology Fundamentals for Digital Finance" series, concentrating on the technical aspects of financial Digital Assets. The series aims to facilitate the use of a clear terminology in a nascent platform-oriented paradigm of financial infrastructures, by laying the groundwork for technical discussions on digital asset standards. The paper introduces a conceptual model named ASAP (Access, Service, Asset, Platform) for Digital Asset Platforms (DAP), leveraging insights from IT industry practices and experiments by central banks. The ASAP model is illustrated through examples and use cases of tokenized assets, to demonstrate the possible usage and merits of modeling Digital Asset Platforms with four layers. Just as the utilization of a seven-layer model (often refered to as TCP/IP) has been fundamental to the interoperability of the internet, it is anticipated that the four-layer ASAP model for Digital Asset Platforms will similarly promote cross-platform interoperability, including across various jurisdictions, paving the way for a more cohesive digital asset ecosystem .

RECOMMENDED CITATION: Budau, V., Tourpe, H. “ASAP: A conceptual model for Digital Asset Platforms”, IMF Working Paper No 24/19, International Monetary Fund, Washington, D.C. 2024.

JEL Classification Numbers:	G12
Keywords:	digital asset, platform, conceptual model, service, tokenization, tokenized asset, interoperability, technical standards, platform-enabled finance, fintech, CBDC, unbundling, open banking
Authors’ E-Mail Address:	vbudau@imf.org, htourpe@imf.org

* This working paper is written in collaboration with Alan Lim and Chin Sin Ong from the Monetary Authority of Singapore, whose participation from the inception, along the chapter IV-C contribution, is highly appreciated. The authors would also like to thank for extremely helpful comments to: Itai Agur, Joshua Aslett, Emmanuel Crown, Federico Grinberg, Tayo Koonprasert, Xavier Lavayssiere, Raunak Mittal, Tao Sun, Wee Kee Toh, Nicolas Zhang, as well as to the World Bank Group (WBG) Technology & Innovation Lab.

WORKING PAPERS

ASAP: A conceptual model for Digital Asset Platforms

Prepared by Victor Budau and Herve Tourpe*

* This working paper is written in collaboration with Alan Lim and Chin Sin Ong from the Monetary Authority of Singapore, whose participation from the inception, along the chapter IV-C contribution, is highly appreciated.

Contents

Glossary

API	Application Programming Interface
ASAP	Access Service Asset Platform
AMM	Automated Market Maker
CBDC	Central Bank Digital Currency
DAP	Digital Asset Platform
DeFi	Decentralized Finance
DLT	Distributed Ledger Technology
DPI	Digital Public Infrastructure
DvP	Delivery versus Payment
ERC	Ethereum Request for Comment
EVM	Ethereum Virtual Machine
FPS	Fast Payment System
FSB	Financial Stability Board
FX	Foreign Exchange
HTLC	Hashed Time-Locked Contract
IAM	Identity and Access Management
ISO	International Organization for Standardization
IT	Information Technology
KYC	Know Your Customer
OSI	Open System Interconnection
PBM	Purpose Bound Money
PSP	Payment Service Provider
PvP	Payment versus Payment
QR Code	Quick Response Code
RTGS	Real Time Gross Settlement
TCP/IP	Transmission Control Protocol/ Internet Protocol
ZKP	Zero Knowledge Proof

Executive Summary

This working paper inaugurates the "Technology Fundamentals for Digital Finance" series, focusing on the technical aspects of financial Digital Assets. This series aims to support the technical collaboration of various stakeholders for addressing known technical challenges of the financial sector or improving its current shortcomings. Among such challenges, tokenized assets, digital platforms, interoperability, and risk frameworks thereof, are of primary interest for this initiative.

This introductory paper introduces a conceptual model for Digital Asset Platforms (DAPs), which aims to facilitate the understanding and foster collaboration of technologists on key digital concepts. We begin by presenting the growing use of digital platforms in financial infrastructures for managing tokenized assets. The paper analyzes the current limitations in interoperability across platforms, primarily due to the absence of consensus on architecture and standards for DAPs. Drawing from successful IT industry practices and experiments by central banks and international institutions, we propose a conceptual framework for DAPs, dubbed the ASAP model. This model, structured into four functional layers – Access, Service, Asset, Platform – is described and applied to real-world scenarios.

Our goal is to establish a set of clear terms and concepts for comprehending DAPs, thereby laying a groundwork for future discussions on digital asset standards, including central bank digital currencies (CBDC). Subsequent publications will explore market requirements in the financial infrastructure landscape to promote interoperability and secure digital finance operations. While this framework identifies new platform model risks, an in-depth exploration of these, including regulatory and legal aspects, is reserved for future publications.

I. Problem Statement and Motivation

The efficiency and stability of financial systems are hotly debated topics, intrinsically linked to the evolving nature of financial assets in the digital era. Innovations like tokenization and programmability[1] of assets have spurred discussions about finance's future. Efforts to understand and shape this trajectory are underway, aiming to mitigate the spillover effects from emerging financial infrastructure paradigms, like digital asset platforms. While there are numerous theoretical and practical approaches to address current system issues, they often fall short in addressing the inherent heterogeneity of these systems – a result of the diverse architectures, design patterns, implementations, technologies, and protocols accumulated over time, along with varied governance and regulatory frameworks.

Current platforms address many issues, and introduce new ones. Proposals aiming to simplify market architecture through integrated platforms are noteworthy. Yet, their scope is inherently limited when compared to the complexity of international financial systems and their multifaceted requirements. These proposals, while addressing certain issues, can inadvertently introduce new complexities by adding components to an already dense landscape. Thus, they form “islands of harmony” within a “sea of diversity” of jurisdictions, technologies, and rules. However, these efficiency-providing islands must still connect and integrate with other, sometimes competing, islands or “continental-scale” legacy systems that constitute the majority of the current global system.

Interoperability at scale[2] across platforms requires the establishment of common approaches and standards, inspired by the success of the internet interoperability. Efficient corridors for cross-border payments and remittances are often enabled by point-to-point integration of payment systems. But this approach is limited, lacking scalability beyond pre-established routes and use cases. The technology sector offers numerous examples where standardization efforts, beyond the financial domain, resolved point-to-point communication challenges across “islands” of information, but failed to address the broader issue of global data transmission. The Internet community resolved this issue with the introduction of OSI[3] and TCP/IP[4] protocols in the 1980s that revolutionized internet connectivity, this paper seeks to address similar challenges in the financial sector. Current banking and payment infrastructures, while utilizing these protocols, have developed in a more siloed manner due to the complexities of handling monetary value. As the Financial Stability Board (FSB) 2020 report highlights, overcoming these payment frictions is both a motivation and a challenge. This paper seeks to answer the question, “What is the TCP/IP-equivalent for digital assets?”

Technical standards are essential for market-level interoperability. The market is exploring various scalable solutions for interoperable cross-border payments and financial services, including access enablers,[5]

[1] Programmability (of a platform) represents its ability to define and customize functions or services through computer code.

[2] Interoperability is strictly referring here to the technical capability of a system to function with, or utilize, the components of another system. We acknowledge the importance of other crucial dimensions like legal or regulatory considerations. The number of point-to-point links for bilaterally interconnecting systems grows factorially with the number of those systems.

[3] OSI – Open System Interconnection model is an ISO reference model for the coordination of standards development for the purpose of system interconnection (ISO - International Organization for Standardization 1994)

[4] TCP/IP, also known as Internet protocol suite, is a IETF technical standard framework for organizing the communication protocols used in the Internet according to functional criteria (IETF, Internet Engineering Task Force 2022)

[5] See (BIS Innovation Hub 2023) for description of access enabler concept, type of technical intermediary (such as in project Sela).

bridging patterns,[6] and multilateral platforms.[7] These aim to serve as hubs supporting multiple currencies, assets, and services. As this exploration progresses, it becomes evident that a structured approach towards interoperability is necessary. This entails identifying a technical base – fundamental functions and their implementations – essential for any high-level design. Interoperability mechanisms must be designed impartially, independent of the specific market-level schemes and the different types of infrastructures they encompass. Such a neutral approach in platform design, with clearly defined functions and their interrelations, is vital for seamless cross-platform operations.

A conceptual model for digital platforms is therefore essential to promote interoperability at scale. This paper introduces a straightforward, inclusive conceptual model to foster a common understanding and collaborative dialogue on digital asset platforms. The model extends beyond technological aspects, shedding light on regulatory issues such as risk management and the financial stability impact of various platform elements. Additionally, the model serves as a tool for identifying key areas where public authorities can effectively and responsibly encourage interoperability among both established and emerging platforms. While in-depth exploration of these topics will be featured in subsequent publications of the "Technology Fundamentals for Digital Finance" series, this initial paper lays the groundwork, providing essential insights and tools for future discussions.

II. Digital Asset Platforms for Finance

This chapter introduces the concept of a *Digital Asset Platform* (DAP) as a foundational element of an emerging sector of the financial infrastructure landscape, known as *platform-enabled finance*. Just as the *platform economy* has evolved through the adoption of digital platform business models in commerce, financial activities have similarly become more complex due to increased digitization, necessitating systematic digitalization and automation of their core functions.

Digital platforms introduce a *vertical* organization of functions and roles within the system, contrasting with the traditional *horizontal* chaining of siloed intermediaries in financial transactions. Since the functions involved in such transactions are executed by various organizations using different systems, the double-entry bookkeeping traditionally used in interconnecting architectures generates numerous information flows and activities for reconciliation purposes. The DAP paradigm offers a unique communication and settlement environment where numerous functions can be securely integrated vertically. DAPs also enable organizations to share a common understanding of the platform's ledger state without the need for explicit reconciliation messages.[8]

Digital Asset Platforms represent the combination of research initiatives that leverage tokenization and programmability, potentially taking the financial sector to new levels in terms of services, participant inclusion, competition, resilience, and integration with other economic activities. As with any innovative approach, this

[6] See (BIS Innovation Hub 2022) for example of bridging mechanisms between payment systems (such as project mBridge).

[7] See (Adrian, et al. 2022) for example of multilateral platform (such as X-C platform) or (BIS Innovation Hub, Banque de France, Monetary Authority of Singapore, Swiss National Bank 2023) for the Mariana project.

[8] Distributed ledgers (as those built with DLTs – Distributed Ledger Technologies) make use of message exchanges between the various parts (nodes) of the system to ensure their copies of the ledger remain consistent. If this synchronization (also called consensus) resembles the reconciliation processes in the financial markets, it is however done at a very low technical level, for any type of information contained on the ledger, and remains transparent to the financial services that execute on the platform and thus benefit from this integrity assurance.

paradigm also introduces new risks in finance, that need to be understood and managed. While peripherally touched upon in this document, they need be explored in future papers.

A. Tokenization, an Enabler for Digital Asset Platforms (DAPs)

The way assets are represented (specifically, their data model) significantly influences how they are managed within a system. Traditional asset representation often necessitates substantial trust in the operator managing their support, namely, the entity controlling the underlying infrastructure. In contrast, representing assets as digital tokens allows for a fundamental separation between the management of functions underpinning an asset and the rest of the supporting platform. Such segregation profoundly impacts both the governance of an asset and the architecture of DAPs that facilitate tokenized assets[9].

Segregation of Asset and Platform

Tokenization of financial assets introduces a modular approach to the architecture of financial systems. It encapsulates information and rules within an asset's implementation, safeguarding the integrity of its state and behavior. This cryptographically enforced data structure[10] provides guarantees for the valid conditions of a token’s state change, reducing the need for trust in the platform's operator. For instance, the transfer of a tokenized asset occurs only after the underlying infrastructure receives a signed instruction from the asset's holder, preventing any interference from the platform operator.[11]

Architecturally, tokenization creates a distinct separation in the management of the *asset* and *its hosting platform.* The same cryptographic methods also provide independent control and governance over the tokenized assets and any services, such as smart contracts,[12] deployed on the platform. These features lay the foundation for an unbundling of systems managing financial assets.

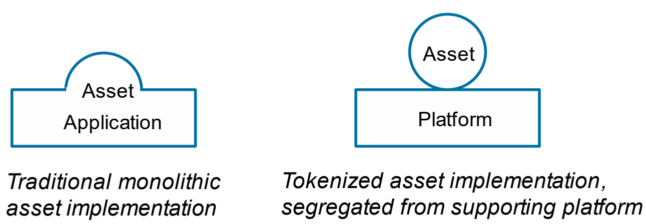


Figure 1: Monolithic vs. tokenized implementations of assets

Traditional Asset Systems

[9] The concepts of *digital asset* and *tokenized asset* are used interchangeably in this paper, as tokenization is the fundamental characteristic of the implementation of financial digital assets in DAPs.

[10] Usually, a token’s ‘enveloping’ structure make use of asymmetric cryptographic primitives, which allow its identifier to be mathematically linked to a unique secret, called private key. Knowledge of such secret (usually stored in a wallet) can be leveraged for the authentication of the token’s rightful owner.

[11] The platform’s operator can still control the access to the platform (by delaying, reordering, or censoring the execution of transactions) but cannot validly alter the ledger for a state change that is not instructed by assets’ holders. An operator’s control of the access to a platform’s resources can be alleviated through the decentralization of the platform’s operation.

[12] Smart contracts are inalterable computer programs (applications) executed on programmable platforms that change the state of a ledger according to precise inputs and rules. They can implement financial services, assets or any arbitrary logic, making them a common choice for implementing any sort of tokens.

In traditional systems, such as core banking systems, the distinction between Assets and Platforms is not as pronounced as in tokenization. These systems typically combine both asset functions (like value representation and exchange) and platform capabilities (like data storage and communication) in a unified, monolithic structure. This approach aligns with the simpler representation of traditional assets as database entries and the critical role of trusted tiers in operating these systems (Figure 1).[13] Since these systems are often centered around simple assets like bank clients' deposits, a distinct internal implementation of asset and platform functions is not a primary concern. The key is adhering to standard communication protocols through APIs (Application Programming Interfaces), which shield the internal workings from external systems (such as payment systems). In this setup, the bank serves dual roles: as the issuer of the asset and the operator of its underlying infrastructure, essential for maintaining the integrity of the traditional asset.

Tokenized Asset Systems

Tokenization, by segregating the implementation of an asset's functions, allows for a clear separation between the Asset and the Platform regarding management, control, risk, stakeholders, governance, and economics. The asset becomes an independent entity that can be designed separately from the platform. Trust assumptions and non-functional attributes, like security or integrity, are not solely dependent on the platform and its operator. The issuer of an asset is not necessarily the operator of the platform, which is instead focused on providing capabilities for other similarly issued assets. This specialization can lead to greater efficiencies and economies of scale, as seen in multi-currency or multi-asset platforms, enhanced distribution facilities (such as issuing an asset across multiple platforms, like the widespread multi-platform issuance of stablecoins), and the creation of ecosystems conducive to innovation (Figure 2).

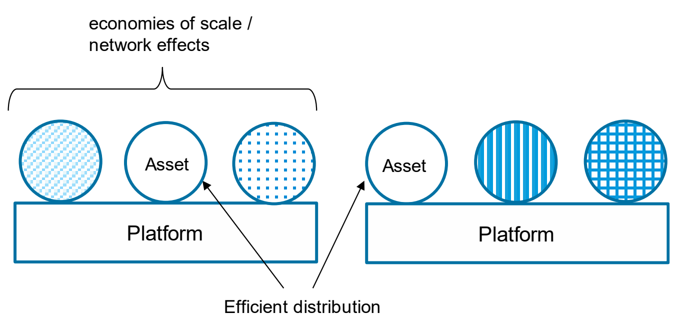


Figure 2: Benefits of leveraging Asset-Platform separation through tokenization

Issuance of different assets (pictured above as different shades) can be done at low marginal costs by reusing a platform designed to support multiple assets. Their co-existence on the same platform is prone to direct interaction and the creation of network effects. Additionally, the distribution of an asset across several platforms may be facilitated through the porting, when possible, of a token implementation over several infrastructures.

The shared capabilities of platforms across multiple assets foster an integrated environment conducive to interoperability. Particularly, a common shared state on a single platform creates a tightly knit ecosystem, unlike the more loosely connected components operating on independent systems and interacting through asynchronous messages. This shared state can be harnessed to enable powerful features like the atomicity of

[13] The integrity of the traditional assets, as records in the ledger, is warranted by the database operator, hence the necessary regulation in the financial sector that applies to these tiers, such as the banks.

complex transactions across different assets, without additional trust in third parties or the operational risks associated with external execution.

These benefits, alongside other desirable attributes such as programmability and composability, position DAPs as a significant trend in the financial infrastructure landscape. They hold great promise as tools for devising market-level interoperability mechanisms.

B. Platform-Enabled Finance

The widespread adoption of public blockchain networks has democratized token issuance by reducing costs and access barriers, fostering a trend where asset segregation from the platform is increasingly common. These networks encourage the development of diverse use cases for tokens, contributing to the proliferation of DAPs. However, the expansion of DAPs and their ecosystems, attempting to mirror the success of e-commerce counterparts, faces constraints due to trust assumptions about governance and security, and the challenges of meeting regulatory requirements.

As the digital platform ecosystem expands, there's a risk of market fragmentation at various levels. This includes liquidity fragmentation when assets are distributed over multiple platforms, service provisioning issues arising from different organizations governing services even on the same platform, and complexities in participation and communication due to market players assuming multiple roles and operating within siloed infrastructures. Indeed, the race to innovation and the time-to-market incentives are *centrifugal* forces that disperse assets, capacities, and participants, further compounded by sovereignty, geopolitical and regulatory considerations[14] that inherently split the global market. However, existing DeFi[15] ecosystems and on-going experiments with digital assets also show that *centripetal* forces – the concentration of market functions on trusted platforms – can yield significant improvements, particularly in cross-border contexts.

The result of these antagonistic forces is an emerging, heterogeneous ecosystem of DAPs. The concept of DAP is highly inclusive in this paper, encompassing platforms of various sizes, technologies, governance models, financial services offerings, and business models. The ecosystem they create is referred to as *platform-enabled finance*, by analogy to the *platform economy*. While the comparison to e-commerce marketplaces is not perfect, it underscores the shift in parts of the financial infrastructure towards structured digital platforms. Here, specialized operators host other market participants who issue assets and develop services. Platform operators provide the necessary infrastructure services, enabling ecosystem participants to focus on their unique value propositions. This trend towards "platformization" in e-commerce and digital product management is indicative of the transformative potential of platform-enabled finance, should it scale effectively.

The international monetary system (IMS) and the broader financial market are evolving into increasingly diverse and complex landscapes. This evolution spans a continuum from traditional, mono-currency

[14] One of these would be, for example, the PFMI set of standards for financial market infrastructures that promotes the distribution of functions on various platforms for implementing strong risk management and achieving global resilience and financial stability.

[15] DeFi, Decentralized Finance, are mostly permissionless and largely unregulated ecosystems today operating on public blockchain networks and proposing financial services accessible to any participant.

applications operated by single banks to private or consortium-led service-focused platforms,[16] extending to new permissionless platforms supporting multiple assets and services.[17]

C. Risks and challenges of platform-enabled finance

While the architectural composability[18] of a system introduces flexibility, it simultaneously ushers in new risks. These risks may arise from the addition of new functions, different implementations of existing functions, or the division of governance responsibilities.

In platform-enabled setups, the singular responsibility traditionally held by a monolithic system's operator is replaced by a more complex governance structure involving new roles and market players. As a result, risk assessments become multifaceted, encompassing multiple components. For example, in DAPs, the issuer of a currency may not be the same entity operating its ledger or holding users' assets. This separation of roles diversifies the risk associated with each function, ultimately affecting the overall risk exposure for asset holders. The risk profile of holding an asset also varies depending on the ledger it's recorded on, influenced by factors such as resilience and access policy.

Beyond the challenge of managing multiple independent risks on these platforms, assessing the risk components can be complex. For instance, stablecoins are often backed by other assets, but the risk assessment of their redemption processes is intricate. Similarly, services implemented via smart contracts may carry operational risks that require in-depth code inspections by external parties or empirical analysis over time to evaluate.

The redefinition of risk models[19] in this context makes the operation, oversight, and supervision of DAPs a complex endeavor. It necessitates an understanding and assessment of various risks, potentially requiring new risk frameworks and data sources. Clear taxonomies are essential to ensure the accuracy and comprehensiveness of information. Moreover, standard formats for accessing and using this information become critical. Regulation must evolve to address these new and accumulating risks, such as automated contagion or excessive leverage facilitated or amplified by the modular nature of financial services on these platforms.

Both the interoperability and risk management of DAPs underscore the need for precise concept definitions and their detailed and comprehensive descriptions. This document proposes several approaches, offering guidance and clarity in defining objects within a DAP, standardizing them, and managing them efficiently, both within a single platform and across multiple platforms.

[16] Such broad category includes, for example, Banque de France's DL3S securities settlement platform, Goldman Sachs' tokenization platform GS DAP, WeChat and Alipay universal networks, or the future announced 'everything app' X.com

[17] This category mainly includes public programmable blockchains (such as Ethereum, Tezos, Algorand, etc.) and their composite scalability modules (such as Layer 2 applications).

[18] Composability is a system design principle that promotes the assembly of well-defined and reusable parts to form a larger system.

[19] This note acknowledges the need to rethink some of the risk models currently used in the industry, but does not expand on their identification, definition, nor possible changes. The risk constituent is used to support the introduction of an architectural framework that, along with other tools, could help with its redefinition.

III. ASAP: a Conceptual Model for Digital Asset Platforms

The proposed ASAP conceptual model (Access-Service-Asset-Platform) promotes technical cooperation on interoperability and on the management of risks inherent to the platform paradigm. It organizes the essential components of digital asset platforms into four distinct layers, facilitating a clear conceptual understanding and streamlined implementation efforts. This model acts as a functional - and to some extent, also governance - blueprint for DAPs, enabling the identification of commonalities among ecosystems for their proper design, and determining where efforts should be committed to improve interoperability. Moreover, it aids in comprehending and addressing new risks introduced by DAPs, when combined with complementary models such as asset descriptions, which will be described in subsequent papers of this series.

A. The Layered Stack of the ASAP Model

The ASAP model, akin to the OSI model which serves as the foundational layers for internet protocols, is structured into four interconnected layers: Access, Service, Asset, and Platform.

These layers encompass high-level functional components necessary for the interoperability of DAPs, each layer fulfilling separate objectives. The stacked layout highlights that functions in each layer are being utilized, or consumed, by other functions in the same layer or from above in higher layers.[20]

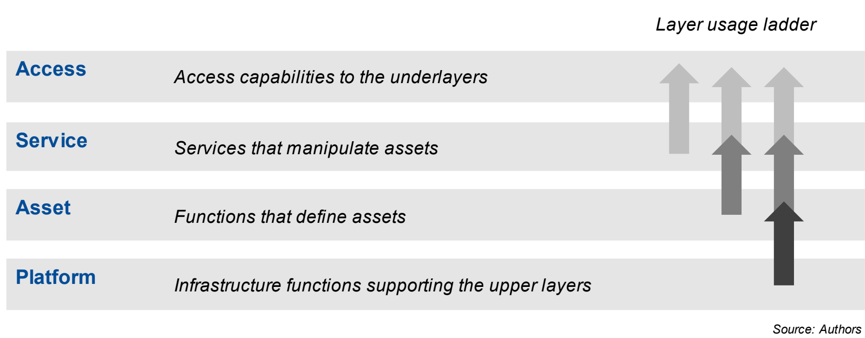


Figure 3: ASAP model layers and their serving relationships

The ASAP model aligns with other existing models, such as those by Schär (2020), Banco Central do Brasil (2022), and the Monetary Authority of Singapore (2021). These models either expand and rename layers, focus on decentralized platforms, or offer frameworks for governing digital currencies. They also inform the ASAP model, designed to be inclusive of all platform types, irrespective of their technology, governance, or the types

[20] The similarity with the ISO's OSI stack or the Internet/TCP-IP stack modeling approach is not fortuitous, as it has proved very useful in decomposing and abstracting the interaction of complex systems for communication purposes. Distinctively however, in these models one layer is strictly providing services to the layer above itself, whereas in our model a function is used by any other function in the same layer or in any layer above.

of assets they handle. Such an implementation-agnostic approach of the ASAP model is crucial for achieving comprehensive interoperability across various platforms.

In the following sections, we provide a concise description of each layer of the ASAP model, highlighting key characteristics and functions of each. The layers are discussed in a bottom-up sequence, reflecting how concepts in the upper layers build upon and utilize those in the layers beneath.

B. Platform Layer

The Platform layer includes runtime capabilities supporting all the other layers' functions. In certain platform models, the Platform layer capabilities are primarily used to implement the transfer of financial assets of the above layer, hence the occasional reference to it as the "Settlement layer". The same goes for specific models dedicated to Decentralized Finance (DeFi), where the natively issued assets can indeed settle on the platform, as they are not digital representations of assets issued on other platforms.

Platform layer

- Execution
- Storage/Ledger
- Communication
- Consensus
- Identification
- Authentication
- Authorization

Source: Authors

Figure 4: Infrastructure functions of the Platform Layer (examples)

Its implementation is arguably the most technically sophisticated part of any DAP and can range in practice from a simple backend[21] of a core banking application, to a DLT (Distributed Ledger Technology) flat network of several nodes, to a composite system of several layers[22] implemented with Web 3[23] technical stacks. Regardless of its structural complexity or the various degrees of performance, service level agreements, trust, or resilience it may achieve, this lower layer caters to the needs of the upper layers with foundational functions (Figure 4), such as the following:

- ***Execution* of software code for higher layers components**

  A platform's ability to execute code is a critical function. Often referred to as an execution engine[24] this capability runs code, such as scripts or smart contracts, that implements services built in the upper layers. Irrespective of its technical implementation[25] or distribution over one or several nodes or layers,[26] the

[21] *Backend* is a software engineering term designating the core tier of a web application that implements its business services, as opposed to the *frontend* tier that presents those services for user consumption (and whose functions belong to the Access layer in our model).

[22] Such as the model N2 of the Network structure explored by project Guardian (Monethary Authority of Singapore, BIS 2023)

[23] Web3 is a blockchain-based technology stack of protocols that enable the development of decentralized applications.

[24] An analogy for the Execution capability in the context of a computer could be its CPU (microprocessor) that executes instructions.

[25] The execution can be achieved using general programming language runtime libraries (e.g., Java, Python, Rust, Golang), restricted versions of these (for deterministic guarantees), dedicated Virtual Machines tailored for various platforms' needs, or a combination of them.

[26] Execution capability can sometimes be implemented over very complex platform architectures, stretching *horizontally* with several nodes (for trust considerations) or *vertically* with layers (mostly for scaling its operation, such a Layer 2 patterns). The ASAP model abstraction considers the capability, and remains agnostic to its implementation details.

execution environment is required to deterministically and atomically[27] process sequences of code instructions. One of the most popular execution environments is the *Ethereum Virtual Machine (EVM)* that updates the global state of a system built on Ethereum or Ethereum-like technologies, by executing specific programs called smart contracts which implement logic from the upper layers.

- ***Storage* of data structures for platform-hosted objects (e.g., assets, services)**

  This foundational function is essential for securely storing the representation of elementary data, assets, and services on the platform, as well as the representation of the transactions submitted to the platform.[28] It works in conjunction with the *Execution* capability, which reads the state of the system from *Storage*, computes its new state according to users' valid transactions, and saves it back into *Storage*.

  An illustrative application of this capability is the implementation of a financial ledger, which records the data that makes up the financial assets deployed within the platform. Traditional platforms implement the ledger as a collection of database entries, but blockchain technologies and DLTs usually use more complex and cryptographically linked data structures that improve platforms' integrity.[29]

- ***Communication* between relevant platform functions**

  This function facilitates the information exchange within the platform, ensuring seamless interaction between the different functions that require collaboration. For instance, a foreign exchange (FX) service might trigger the payment of an asset, necessitating seamless communication. This process combines both memory sharing within a node and message transmission between nodes, creating a virtual shared space across the platform. Such a setup ensures a high degree of integration of assets and services on the same platform. This distinctive intra-platform integration is crucial for the composability of user-added services on the platform, contributing significantly to the growth of the platform's network effects.[30]

  In the context of distributed platforms, the communication function also plays a key role in disseminating user transactions from the entry node to all execution nodes. It aids in the platform's resilience by transferring parts of the ledger to nodes that are recovering from crashes. Furthermore, this communication capability is integral to the platform's consensus mechanism, as discussed later.

- ***Consensus* between parts of a distributed system**

  Consensus refers to a platform's capacity to uphold a unified state across different nodes. It is usually implemented as a multi-party voting process that executes periodically between several nodes called "validators".[31] Consensus is needed in any distributed system for the synchronization of its nodes, and its characteristics need to be adapted to the trust assumptions vis-a-vis the platform's operators. In permissioned platforms, which are restricted to a select group of participants with known identities (often verified through Know Your Customer, or KYC, processes), consensus mechanisms can be more efficient, as these platforms leverage the trusted status of their operators to streamline decision-making. In contrast,

[27] The atomic execution of an instruction means that it is either completed entirely – transitioning the system into a new valid state - or is rolled back to the system's previous state. Nevertheless, the transaction atomicity at this level does not guarantee the atomicity of higher-level concepts like business transactions or processes (e.g., DvP), which need to be explicitly programmed in the upper layer logic (for example in a smart contract).

[28] An analogy for the Storage capability in the context of a computer could be its memory.

[29] A popular class of such data structures is the DAG (Directed Acyclic Graph) for implementing append-only storages, where new transactions add on to the previous ones.

[30] A platform's network effects also depend on many other factors, including its trust model, public/private participation model at each of the layers, access policies, ease of use or develop, etc.

[31] Validators are (a subset of) privileged nodes that control the validity of a distributed platform's state, through engaging in a communication-intensive and weighted- (over some criteria) vote process.

permissionless networks use algorithms designed to function in highly adversarial environments, potentially including anonymous malicious participants.

The choice of consensus mechanism can significantly affect a platform's performance attributes, such as throughput and latency, and is critical for security. Effective management of the ledger's integrity by the consensus mechanism is key to mitigating risks like double-spending of an asset. However, consensus mechanisms can also introduce new risks, particularly in decentralized platforms, due to susceptibility to novel types of attack vectors.[32]

From a risk management standpoint, understanding a platform's consensus mechanism is crucial. It not only has security implications but also affects how transactions are finalized, whether in a deterministic or stochastic manner. In terms of interoperability, platforms should be designed to be agnostic regarding their respective consensus mechanisms. The ASAP model considers these mechanisms as internal functionalities confined within each platform.

- ***Identification* of platform resources**

  This functionality is crucial for a platform's ability to access its internal resources, like accounts, assets, or services. In modern platforms, these resources are typically identified through unique addresses within a platform-wide public-key cryptographic scheme. Each address is associated with a secret piece of information, known as a private key, used for authentication purposes. The use of cryptographically verifiable identifiers is central to the tokenization of platform resources. It provides the means for authenticating assets and valid transactions that interact with them, an essential aspect of tokenization (discussed in the following *Authentication* capability).

  The concept of *identity* in digital platforms often extends beyond the simple identification within platform boundaries. For interoperability purposes, it often also involves mapping the local, and sometimes proprietary, identifiers of digital concepts on the platform (like a participant's address) to publicly recognized identities of the real-world entities they represent (such as a government-issued ID). Modern management of this identity capability is usually handled by specialized systems. Ideally, these are implemented as Digital Public Infrastructures (DPI), which can be utilized by various financial and non-financial platforms for robust identity management.[33]

- ***Authentication* of transactions**

  This capability enables a platform to securely validate legitimate transactions submitted by participants through the verification of their integrity prior to execution. Several authentication protocols with varying levels of security are available, but most of them rely on variations of the electronic signature. An electronic signature accompanying a transaction can guarantee – by the sole virtue of cryptography/mathematics – that a received instruction’s content is as intended by its rightful sender (which, in this case, is the owner of the private key used for its signature).

  When transaction confidentiality is a requirement, other more complex authentication mechanisms may be used, such as Zero-Knowledge Proofs (ZKPs), that present the added capability of allowing a platform to

[32] Common attacks on the consensus process include “51 percent attack” or “sybil attack”, which attempt to control a distributed network by using overwhelming computing power or creating multiple fake identities, respectively. (Bains 2022)

[33] Traditional infrastructures for identity are the Public Key Infrastructures (PKI) that make use of electronic certificates to provision Certification Authority-sanctioned real-world identities. Government-led Digital Public Infrastructures (DPI) may also provide Digital Identity services, with more user-friendly credentials than certificates, thus facilitating the mass enrolment of the population to new payment platforms. For instance, Aadhaar, the digital identity public infrastructure in India, provides a nationally unique 12-digit identifier to every person in India, irrespective of their, potentially different, identifiers in other systems.

securely change its state without executing the transaction itself. In this case, the platform only verifies the result of a concealed transaction - executed by an exogen system - by the means of an accompanying cryptographic proof of computation's integrity.

- ***Authorization* of transactions**

  This optional function determines a transaction's access rights to resources in the Platform layer. While the authentication process verifies the identity of a participant, authorization decides what actions a participant's transaction is permitted to execute. Most platforms leverage this capability to regulate user access in line with either regulatory requirements or specific platform criteria. Typically, private or consortium-governed platforms restrict access at the Platform layer by allowing only transactions from authorized users, based on identity credentials verified by the *Identification* capability, described above. In contrast, platforms that do not enforce authorization checks at this layer are usually termed "permissionless". They allow open access to any individual or agent to the resources of this layer.

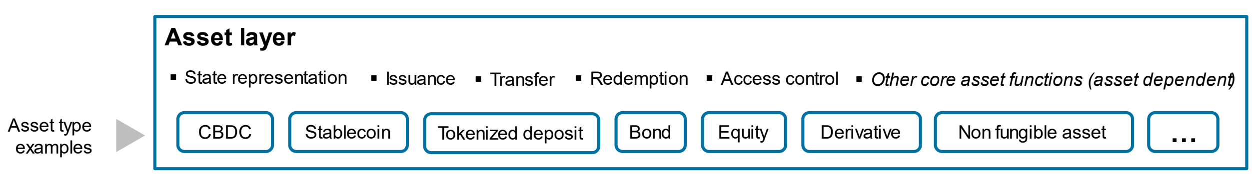
  However, the presence of authorization checks at the Platform layer does not exclude the possibility of similar checks at higher layers. Assets and services often embed programmatic authorization checks to differentiate access among participants or to establish distinct roles. For instance, a tokenized financial asset might limit issuance privileges exclusively to transactions from its authenticated issuer(s).

C. Asset Layer

The Asset layer encompasses core functions that purely define a financial asset (Figure 5). Different asset types require distinct sets of functions (e.g., a bond does not have a strike price like a derivative), but some constitutive attributes – like the representation of units of value, the issuance or transfer functions – may be common across most asset definitions.

Digital platforms have the capacity to incorporate one or multiple types of assets within the same platform. Although assets are recorded within the underlying platform storage capability, each asset possesses logically distinct data and, in some cases, code structures. This arrangement enables assets to be managed independently on the same platform while maintaining the ability to interact seamlessly with each other and with the services situated in the above Service layer.

The implementation of a dedicated Asset layer within a financial platform, separated – through tokenization – from the foundational Platform layer, is a distinctive characteristic of a DAP.



Source: Authors

Figure 5: Asset layer's functions and asset type mapping (examples)

If DAPs make this clear distinction between the Platform and Asset layers, the development of functions above the Platform layer still struggles to map complex digital financial instruments to Asset and Services layers distinctively, from both functional and governance standpoints. This often remains a challenge as both layers are essentially implemented with code that frequently exists within the same structure, such a smart contract.[34] These blended implementations often result from software engineering decisions, but they can cause real difficulties in the analysis and classification of the implemented concepts, particularly when it comes to regulatory considerations.

In the case of a currency asset, aside from the representation of holdings,[35] this layer typically covers functionalities like money issuance, transfer, or redemption. Fundamental governance rules, which define roles, privileges, and their access control logic,[36] are integral to this layer's representation of money. These rules are essential for a comprehensive and correct definition of an asset. For example, an access rule would ensure that a regular asset holder role *is able* to transfer their holdings but may not be *entitled to* issuance privilege.

However, additional functions, such as restrictions[37] or escrow locks,[38] are considered optional for this type of asset. They do not represent core attributes of the money asset itself. Therefore, they fall under the purview of the Service layer rather than being intrinsic to the Asset layer.

This separation also facilitates the research on the right balance to apply between core asset functions and add-on features, when implementing a tokenized asset. For example, implementing a CBDC asset with functions that may impact its circulation perimeter or universal use (such as, conditions for holding or spending it) would risk denaturing its own definition. Preventing the money asset from being “programmable” is essential and requires understanding the fine line between the Asset and Service layers in the implementation of a CBDC system. The Asset and Service domains can be separately managed in terms of access and operational governance, allowing, for instance, a central bank to design, create and manage a CBDC and the private sector to provide services that use this asset, without ever risking impacting the core nature of money.

D. Service Layer

The Service layer covers functions that handle or utilize the financial assets deployed on the platform, as described in the previous section. Within a DAP, these functions, or their combinations, typically facilitate the implementation of financial services. Similar to other resources in a DAP, the governance of these services[39]

[34] Many digital financial assets are being enriched with new behavioral functions implemented in the same smart contract as the asset itself. For example, an investment DAO (investment fund governed by a Decentralized Autonomous Organization that makes investment decisions through member voting) would issue a liability asset (think fund parts) to account for the value that participants entrust to it. This liability asset - that is also used for the fund’s governance (think shareholders vote) – is sometimes implemented in the same smart contract as the core operational investment functions the fund provides, leading to difficulties in understanding the nature of such complex on-chain technical compounds.

[35] Holdings of currency assets are typically implemented through internally mapping local identifiers of holders (addresses or accounts) to balances representing units of that currency.

[36] On DAPs, an asset’s own access control function is not to be confused with the authorization function of the Platform layer, as they regulate access to different types of resources and can be provisioned separately. As operations on tokenized assets are executed by the Execution capability, a successful transaction on such asset would need privileges to access both Asset and Platform level resources, if applicable.

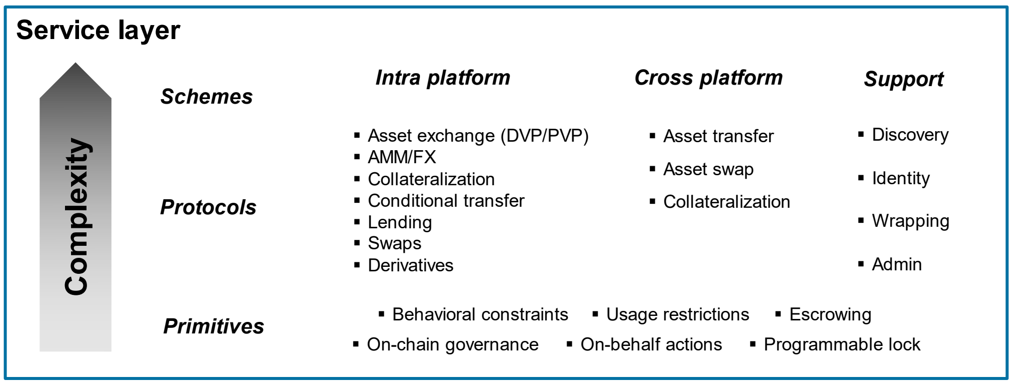
[37] Examples of restriction functions that overpass the strict definition of money: superior limit of a transaction amount or account balance, a list of beneficiary addresses that are not allowed to receive transfers, etc.

[38] An escrow lock is a programmed function that releases the asset(s) it has received in custody only upon the satisfaction of predefined conditions or the occurrence of certain events.

[39] The governance is meant here as exercising, and parameterizing through access control, the various roles a service implements: administrator, transactor, liquidity provider, etc.

can be designed for independence, separate from other services or the assets they utilize. DAPs not only diverge from a monolithic system model in terms of independently governed layers, but also within these layers themselves. The layers are not monolithic either, as the various assets and services within them can be managed separately by different organizations.

The scope and complexity of such services are only constrained by the limits of the *Execution* capability that execute them or the intricacies of the financial instruments themselves. As this layer can be very complex and moldable, we will only provide a broad characterization and classification of its elements, refraining from definitive definitions or taxonomies.



Source: Authors

Figure 6: Service types in the Service Layer, clustered by affinities (examples)

The lower end of the complexity spectrum, as illustrated by the vertical axis in Figure 6, encompasses basic yet fundamental services, characterized by specific rules or mechanisms. These services are often termed 'Primitives' due to their straightforward yet modular nature. For instance, a remuneration service represents a fundamental function, primarily involving the modification of balances based on time-related events and pre-agreed rules. Given its prevalence in many instruments and services, this primitive function can be parametrized, and its implementation modularized, to meet efficiency and other architectural considerations.

While programmability offers substantial potential to a platform, a disorganized approach to developing this Service layer can have adverse effects on its maintenance, evolution, and the interoperability across its services. From both platform management and ecosystem development standpoints, there is considerable value in crafting these services as modular components. The public Ethereum network serves as a prime example. As the largest platform for decentralized finance (DeFi), Ethereum has successfully pioneered this composability approach. Its operational success is largely attributed to strict adherence to best practices in service design and programming.

These essential functional building blocks, often compared to *Lego* elements, can thus be reused by more complex structures that offer standalone financial services, often called *Protocols.*[40] Protocols come off as services designed to cover well-defined and complete use cases like asset exchange, foreign exchange (FX), lending, collateralization, etc. If carefully designed and implemented, they can also be reused by other protocols as well, through the same composition mechanism. For example, a Delivery vs. Payment (DvP) protocol can use an existing conditional payment service for an atomic asset exchange, and provide its own

[40] The protocol appellation, especially used in the DeFi space, narrows however the semantics of an already overloaded term.

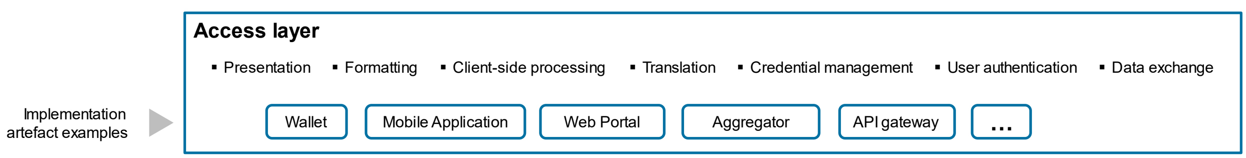
parameters, such as addresses, asset types, or amounts. Occasionally, collections of services oriented towards a specific functional domain can coalesce into *Schemes*, particularly when orchestrating services across multiple platforms.[41]

Considering the usage perspective, shown horizontally in Figure 6, most services are predominantly confined within the platform (that is, *Intra platform*), and capable of fulfilling their intended purpose only using the resources of a single platform. However, many business processes that need to span across multiple platforms are facilitated by the *Cross-platform* type of services. These services would implement one or more local functions on each platform as needed to execute the global service or protocol.[42] The standardization of these cross-platform services emerges as a very promising area for exploration, to achieve interoperability across platforms.

Lastly, several *Support* services from this layer do not directly implement vertical financial logic. Instead, they focus on specific utility functions beneficial for any service within the platform. These functions might include activities like token wrapping,[43] identity-related processing, external information discovery, or functional administrative tasks, and are often implemented as reusable libraries. They can also represent local stubs of larger support processes needed by platform ecosystems, especially when integrating to other non-financial infrastructures. Examples include the resolution of real-world identities of participants based on their local addresses, or the “injection” of external information into the platform, also known as *oracle* services.

E. Access Layer

The Access layer contains functions and interfaces that enable clients such as users, applications, and other market components, to engage with the underlying Service, Asset, and Platform infrastructure layers.



Source: Authors

Figure 7: Access Layer functions and application types (examples)

[41] Frequently found in the payment industry, schemes also involve extensive rules, procedures and arrangements that guarantee behavioral consistency across platforms.

[42] For example, a cross platform asset transfer protocol would entail the secure coordination of two local phases: the neutralization of an asset’s value on the originating platform (through locking or burning) and the release into circulation, or the minting, of the same asset’s value into the destination platform.

[43] Token wrapping is a process that transform a token into an enhanced version of itself, generally to add new properties or restrictions. Technically, it may require the issuance of a new token bearing these characteristics in lieu of the original token that remains locked, until the un-wrapping phase that destroys the new token and automatically releases the original one.

This layer, illustrated in Figure 7, encompasses functions that allow data presentation, translation, and formatting,[44] client-side processing,[45] credential management,[46] or data exchange.[47] Also, user authentication can be implemented at this layer in order to leverage existing capabilities outside the platform, such as IAM (identity and access management) systems.[48] The Access layer can become quite substantial as it covers the necessary capabilities for interacting with the resources exposed by the platform to all stakeholders within their ecosystems. Due to the diverse and distinct requirements that can be easily tailored to individual needs, this upper layer is constructed through a significant and disparate collection of software components and tools, including wallets, web API gateways, portals, client applications, access protocols such as QR codes, or aggregators. Most of the capabilities of this layers are user-oriented and their role is to connect to any type of backend system or platform, transform the information and present it in a user-friendly way.

Many front-end tiers within existing systems can be seamlessly integrated with DAPs, significantly enhancing interoperability between diverse systems. End solutions that deliberately refrain from incorporating lower-layer functions can be 'plugged into' DAPs and other financial infrastructures with different application architectures, such as the monolithic backend applications used by traditional banks, with minimal modifications. This 'vertical interoperability' between the Access layer of DAPs and the various underlying market infrastructures could be greatly enhanced by adopting shared standards in their respective foundational layers.[49]

F. Focus on Assets and Services

The ASAP conceptual model abstracts distinct layers by following specific interests of the financial community, such as the interoperability of systems, their governability or risk analysis. Variations of this model’s layers already exist in the financial sector, and markets featuring specialized actors and products have emerged over the years. We identify nonetheless the two middle layers as an understudied and prolific space for the DAP exploration, where the financial sector members can create a meaningful impact of responsible innovation.

As outlined in section B above, the Platform layer has a very technical nature and is currently undergoing an intense race for innovation, driven by open communities[50] and software vendors.[51] This has led to a wide variety of architectural and technological choices, challenging the attempts to achieve compatibility at this level. Standardization at this layer is complex, and although some de-facto convergence might emerge over time, software makers lack sufficient incentives to engage in such initiatives given the current stage of market's maturity.

[44] Data presentation, translation, and formatting are functions that ensure the proper transformation of the information from an internal format of a system to another format that is adapted to a specific type of consumption (e.g., a human user or a different machine system).

[45] Client-side processing refers to a software engineering pattern of having functions execute on the client device (frontend, instead on the server/backend) that is sometimes desirable for security, performance, scalability, or user experience purposes.

[46] Credential management is a security practice – usually implemented as a function of a wallet – that protects login credentials, certifications, passwords, usernames, keys, power of attorney, etc.

[47] Typical data exchange function at this level would be implemented through the use, by some client application, of API (Application Programing Interfaces) gateways to access different services on various platforms.

[48] For example, web portals may implement a single user authentication to several platforms for user experience improvement considerations.

[49] For example, the vertical interoperability of wallets within a platform can be fostered by adopting standard interfaces in the implementation of assets and services, thereby allowing a broad range of uses for the same wallet on that platform.

[50] Examples of open communities: Ethereum community, Hyperledger Foundation, Tezos community, etc.

[51] Examples of software vendors for platform infrastructures: R3, Consensys, Digital Assets, Knox, etc.

On the opposite end of the ASAP model, the Access layer is a bustling arena in numerous aspects: it incorporates functions that directly cater to extensive user bases, benefits from mature and scalable web technologies, and has garnered attention from numerous actors, including various Payment Service Providers, and participants in the open banking arena, aiming to capitalize on the already built financial market infrastructures. Many privately created, and sometimes regulatory and enforced standards are being created at this layer, particularly catering to the overall user experience and/or trying to bridge siloed systems underneath. For instance, seamless payment data exchange is now largely facilitated through standard QR code formats.[52] Nonetheless, this dominant top layer interoperability approach stands to gain significantly from leveraging standards at Service and Asset layer levels. For instance, the ERC20[53] standard in the Ethereum-based ecosystems has enabled wallet compatibility across assets and platforms, resulting in a thriving wallet market that provides users with a multitude of compatible options for storing and utilizing their tokens.

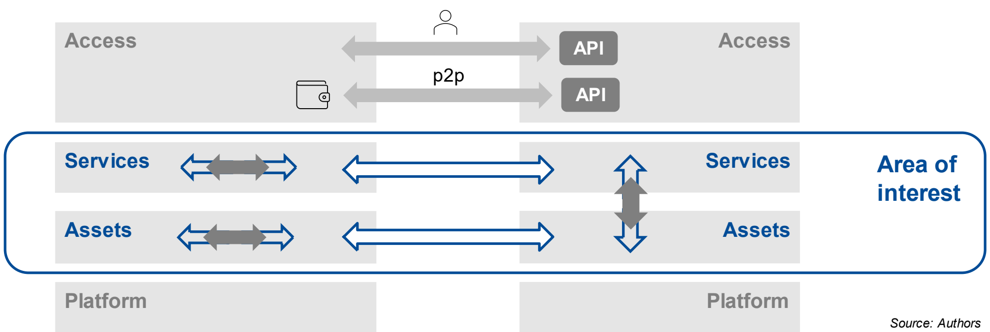


Figure 8: Area of specific interest for extending interoperability and enhancing risk management in DAPs

The Asset and Service layers (Figure 8) are particularly interesting for the financial sector because of the regulated nature of most financial assets and services they concentrate. Under the guidance of the public institutions like central banks, sectorial supervisors or various international bodies, this sector has a long history of cooperation for implementing multi-party financial services in efficient ways. These two layers offer an avenue where the financial sector can harness its potential for coordination to attain interoperability. Just like the ERC20 standard allowed unprecedented composability of assets and services on the platforms that support it, a standardization effort in these layers may bring efficiency in channeling the resources of the financial sector in this regard and effectiveness in delivering the important externalities brought by the interoperability.

The design decisions in the Asset and Service layers are crucial as they could significantly influence the structure of several markets, historically evolved to meet the sector’s challenges. Their position as foundational elements within the financial market infrastructure underscores the need for advanced frameworks in risk assessment. Addressing interoperability and risk management in these layers is of paramount importance for platform-enabled finance. These critical topics will be further explored in subsequent notes of this series.

[52] An example of interoperability at this level is the Interoperable QR code specification (Centre for Digital Public Infrastructure 2023)
[53] ERC20: token standard (Ethereum Foundation 2015)

IV. Applications of the ASAP Model

To illustrate the usefulness and effectiveness of the proposal, this chapter proposes several use cases that test the validity and applicability of the ASAP model for various purposes.

A. Representation of key DAP concepts

One benefit of the ASAP model lays in its capacity to help clarify and understand complex concepts within digital finance, such as digital assets, programmability, and tokenization, as well as their interrelated dynamics.

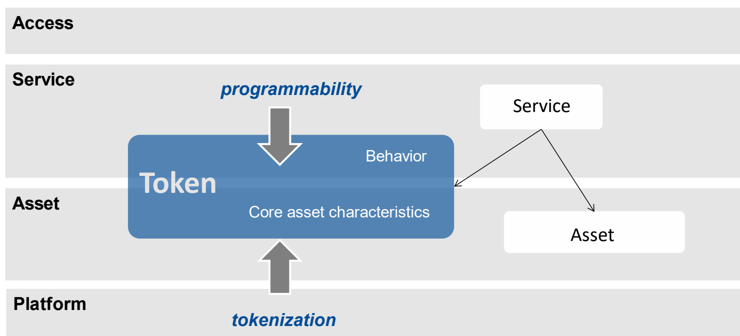


Figure 9: Tokenized asset concept and related notions

The *token*, seen as the container of a *digital asset,* represents an applicative construct that implements functions from both Asset and Service layers of a platform (Figure 9). The Asset layer provides the core characteristics inherent to any object managed by the DAP (III.C), while the Service layer implements the services that guarantee the intended behavior of the asset (III.D).

Digital assets are the result of two powerful capabilities of modern platforms: tokenization and programmability.

Tokenization creates the distinctive "envelope" of the digital asset, which ensures that the governance of its content is independent of the platform environment. Only organizations holding the relevant private keys can interact with the token, each according to their role (like issuer or holder). In particular, it insulates the asset from the interference of the platform's operator, shielding it from the Platform layer and positioning it into an Asset layer with distinct state and representation of core asset characteristics, as well as its own governance.

Programmability allows for the precise definition of an asset's behavior through programming code, situating it within the Service layer. This means the digital asset straddles both the Asset and Service layers. Embedding the asset's behavioral rules alongside its core characteristics within the token's cryptographically secured boundaries ensures consistent state management over time, executed as intended by its creator.

In the case of the money asset, such design is referred to as *programmable money*. However, caution should be exercised in using this term due to the specific nature of money as a universally accepted asset.[54] If the behavioral functions were separate from the tokenized representation of the asset, and implemented instead as an independent Service interacting with an Asset (as shown in Figure 9, each in their respective layer)[55], maintaining consistency between the asset state and its intended behavior would necessitate the unified governance of both the asset and the service implementations. Whilst achieving, in this case, similar results as the inclusion of both state and behavior into a token, this separation could nonetheless be driven by software engineering needs, such as code maintainability or change management.

Yet, there are scenarios where it might be preferable to have services governed and operated separately. For traditional money assets, a design model where the payment function is implemented as a distinct service, under different governance and possibly on a separate execution platform, is known as *programmable payment*. In the context of Central Bank Digital Currencies (CBDC), this model emphasizes the role of the private sector, where Payment Service Providers (PSPs) offer payment services for central bank-issued public money.

B. Charting evolving market models in banking

The financial sector is currently experiencing a substantial shift towards more open designs and distribution of services, as illustrated in Figure 10. This trend is particularly pronounced in retail money markets, where the traditional monolithic banking model is being challenged. In this model, institutions offer fully integrated solutions based on customer deposits, encompassing custody, money services, and access. However, this approach is facing pressure from both market dynamics and regulatory bodies advocating for increased competition and innovation within the industry.

The rise of *Open Banking*[56] has paved the way for the emergence of new players, predominantly from the Fintech sector. These participants provide customer-centric services within the Access layer, such as account aggregation and payment initiation. It is important to note that these new entrants are essentially only altering the presentation of assets and services,[57] while the underlying nature of these components remains unchanged.

Nevertheless, the banking sector also capitalizes on this architectural openness through the implementation of common services that banks can collectively offer while leveraging their existing API-enabled platforms. A notable example of such a service is the Fast Payment Services (FPS) that usually allows a sector-wide and commonly governed entity to collect and real-time process clients’ payment transactions. These services typically enable a sector-wide, collectively governed entity to gather and process client payment transactions in

[54] Functions that change a currency’s core characteristics (by restricting, for instance, its employment to specific purposes or use cases) may not qualify for being programmed into this type of digital asset.

[55] Figure 9 illustrates another perfectly valid interaction in DAPs, where a “pure” service (implementing solely logic) uses a tokenized asset too.

[56] Such as EU’s DSP2 directive (European Parliament and the Council 2015) or UK’s Open Banking Standard (Open Banking Limited 2018)

[57] New propositions for the presentation of banking services are made possible by having banks implement Application Programming Interfaces (APIs) for their core banking systems, opening them for “consumption” by external applications.

real time. This entity then interacts with the banks' core system APIs to update their ledgers and sometimes also directs their settlement in the Real-Time Gross Settlement (RTGS) systems.

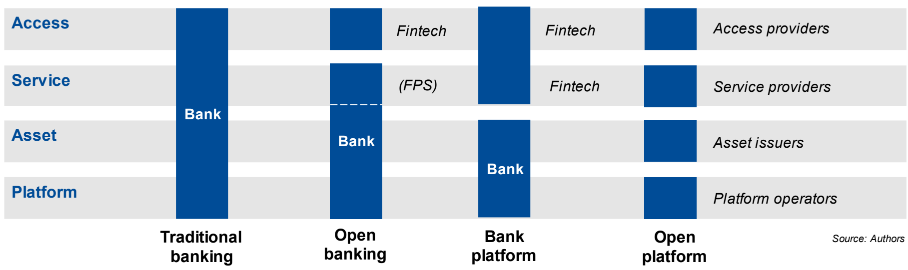


Figure 10: Banking market trend of openness

In the increasingly platform-oriented financial landscape, several banks are exploring proprietary platforms where they exercise full control over the Platform layer. Their aim is to capitalize on the assets issued within the Asset layer. As operators of these *bank platforms*, they often invite non-bank financial entities, typically Fintech companies, to develop services that leverage the bank's tokenized currency.

However, this approach encounters certain limitations and challenges. One issue is the restriction to a single currency on the platform, limiting use cases to the domestic domain. Additionally, governance-related obstacles might discourage actors from joining platforms operated by competitors.[58] Trust in a platform governance is a key factor in fostering successful ecosystems. Consequently, platforms operated by reputable public entities, such as central banks, may have an advantage. Several central banks are experimenting with this model, creating platforms for issuing wholesale Central Bank Digital Currencies (CBDCs). Private sector players are encouraged to innovate by building services atop these platforms, potentially including the issuance of various financial assets like securities.

Expanding the scope to allow more issuers at the Asset level could lead to more *open platforms.* Here, operators can leverage their trust status to foster ecosystems where participants can issue assets, create services, and develop client-facing applications. This open model of issuing multiple tokenized currencies opens doors to innovative services that address existing limitations in cross-border transactions, like remittance settlement and foreign exchange.[59]

This approach mirrors the model of permissionless platforms[60] like the public Ethereum network, which allows unrestricted access across all layers and value chain segments. While fostering innovation, this openness can also be a conduit for financial misconduct and consumer abuse. As financial regulations evolve to address these challenges, open platforms may need to implement mechanisms ensuring trust, security, and stability for sustainable growth in a regulated environment.

[58] This phenomenon is visible, for example, in trade finance where many platforms fail to build *coopetition* (cooperative competition) among business competitors.

[59] Adopters of the multi-currency models are for example the X-C platform (Adrian, et al. 2022) or Mariana (BIS Innovation Hub, Banque de France, Monetary Authority of Singapore, Swiss National Bank 2023)

[60] Permissionless platforms (also known as public blockchain networks) represent a subset of the Open platforms category that maximize decentralization of their ecosystems, in all their layers.

This discussion underscores the emerging trend of openness in the financial ecosystem and highlights the complex challenge of asset data representation and provisioning in platform-oriented models. In traditional banking systems, limited and often implicit information about financial assets, such as currencies, is inadequate in this new context. This is especially true when different functions of an asset can be defined and handled in distinct ways by various stakeholders, all with differing underlying trust assumptions. In a forthcoming paper, we intend to address this subject and propose an extended informational model, geared towards offering explicit and verifiable information about digital assets, for both risk management and interoperability considerations.

C. Defining standardization areas for Purpose Bound Money (PBM)

The concept of *Purpose Bound Money (PBM)*[61] has been introduced by the Monetary Authority of Singapore (MAS) as a design framework for imbuing digital assets with programmable behavior. This is achieved by harnessing both the versatile applicability of the *programmable payments* to various money forms, and the encapsulation and transferability of *programmable money*. In this respect, the PBM type of asset cuts across both the Service and Asset layers, acting as a digital asset that augments the behavior of an initial asset.

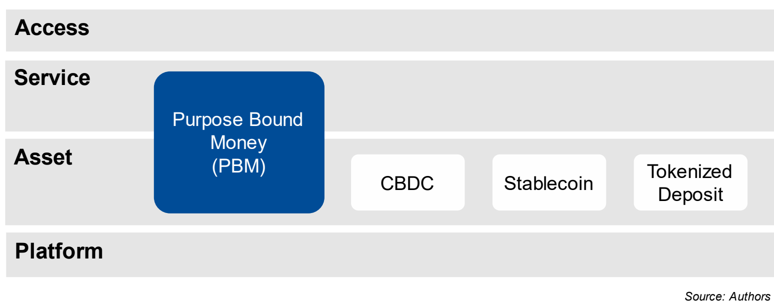


Figure 11: PBM positioning vis-a-vis various types of digital money.

The originality of this digital asset lies in its ability to function – at an abstract level – as both money and bearer instrument. Take the example of vouchers: PBM can encapsulate underlying digital money with specific logic (binding it to a purpose), thereby enforcing usage restrictions throughout the payment chain without intermediaries. These restrictions, which would normally be incompatible with the universal acceptance of money, are maintained until all conditions for “unwrapping” are met. Once these conditions are satisfied, the underlying digital money is released to a recipient, who can then use it according to its original logic and conditions, unaffected by the prior restrictions.

The PBM protocol emerges as a particularly promising solution for creating interoperable systems. It facilitates the reuse of the same software components or their standardized implementations across multiple scenarios, such as commercial vouchers or financial grants. The potential design approach for the PBM protocol can be effectively illustrated using the ASAP model, which provides a clear framework for standardization.

[61] See detailed description of the MAS-introduced concept of PBM (Monetary Authority of Singapore 2023)

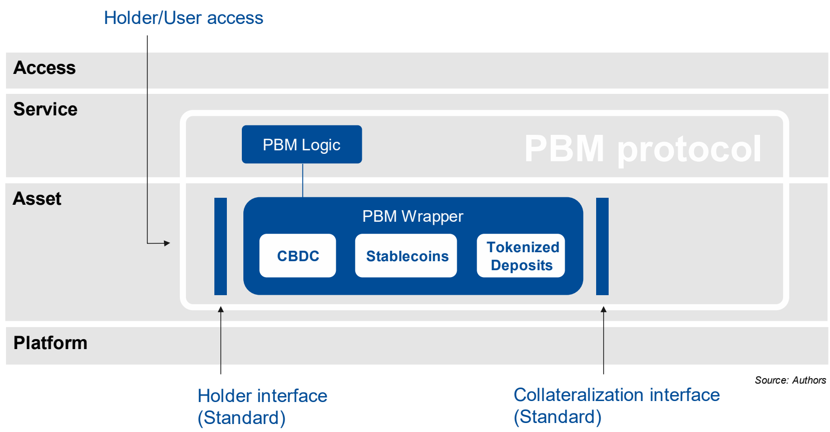


Figure 12: PBM protocol components and its key public interfaces

At the core of PBM protocol lie a wrapper, where the programmable rules are defined, and the digital money, which backs the PBM. To attach rules, digital money is placed in a wrapper as a collateral for a secondary token (PBM token) issuance. The PBM token essentially “wraps” the digital money and places it in a wrapped state. When the PBM token is used, it operates under the constraints of its own rules. Once the PBM token serves its intended purposes, it unwraps and releases the underlying digital money, allowing the holder to utilize the digital money according to its native properties.

While interactions between various PBM modules remain internal to the scheme, two crucial public interfaces establish a connection between the PBM and a platform ecosystem. Firstly, the *Holder Interface* enables PBM holders to interact with their tokens and necessitates implementation within wallets. Secondly, the *Collateralization Interface* allows for the definition of rules and formats for establishing collateralization relationships between a Wrapped token and its backing assets, potentially of diverse natures.

These two interfaces represent focal points for standardization efforts due to the immediate externalities of such initiative. Standardizing the Holder Interface facilitates the creation of interoperable wallets for PBM users, while standardizing the Collateralization Interface effectively and efficiently enhances liquidity accessibility for PBM issuers. The current MAS implementation leverages several Ethereum standards and the trust assumptions in a single platform, thus limiting collateralization to currencies available on the same platform as the PBM. However, a cross-platform standard for the PBM, assorted with the necessary trust assumptions, would allow to “tap” into vast arrays of assets issued on multiple platforms. Given the considerable scale of the user, wallet, PBM, and digital currency markets, establishing and utilizing standards for the PBM protocol is indispensable for its widespread adoption by both issuers and users.

D. Exploration of service interoperability between platforms

The use of the ASAP conceptual model for understanding interoperability between platforms will be explored more deeply in a future paper of this series. An example of the applicability of the ASAP model to this purpose is the illustration of a technical interoperability solution for the settlement of cross-border payments by means of a widened access to central bank liabilities.

A prominent early experiment in the CBDC space exploring the feasibility of this design was the Jasper-Ubin experiment led by Bank of Canada and the MAS. Its design paper (MAS 2019) explains in detail a technical approach of an asset exchange protocol using *Hashed Time-Locked Contracts* (HTLC).[62] It also describes the cross-platform workflow that participants need to follow to implement an atomic Payment versus Payment (PvP) transaction across the platforms. In this setup, payments on each platform are either simultaneously executed or collectively rolled back, ensuring transaction finality and reducing counterparty risk.

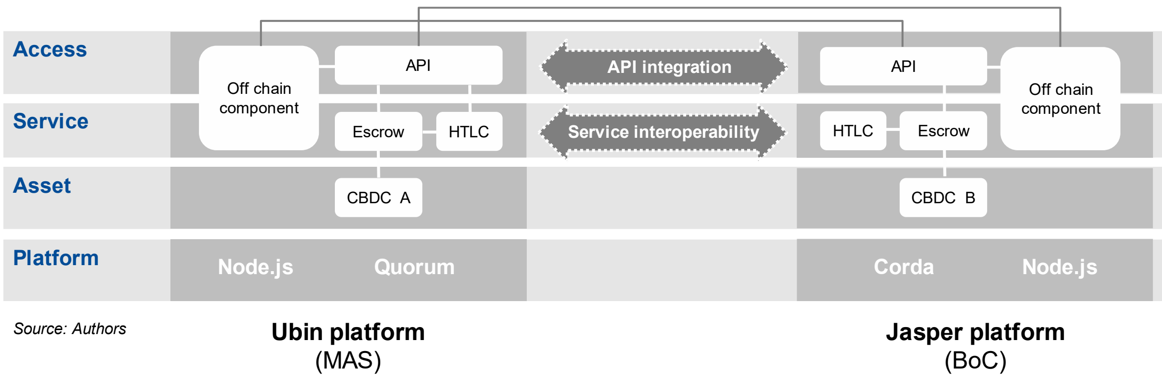


Figure 13: Asset exchange protocol between two CBDC platforms in the Jasper-Ubin project.

In this experiment, each central bank's platform (Figure 13) uses a DLT infrastructure at the Platform layer – Corda for the Bank of Canada, Quorum for MAS – that allows the issuing of the tokenized version of their money, i.e., a CBDC in Asset layer. It also supports the necessary services that implement the cross-border PvP, essentially in the form of an Escrow service and an HTLC. The elements of a platform that use DLT for infrastructure are called *on-chain*, in reference to the chained structure of a blockchain ledger.[63]

The platforms used in the Jasper-Ubin experiment also feature *off-chain* components that do not execute on the DLT infrastructure but on traditional execution environments (Node.js in this case). These off-chain components are not only necessary for the user to interact with the platform resources,[64] but also to connect the on-chain part of the platform with external entities,[65] such as other platforms. Therefore, these off-chain components play a critical role in implementing the technical data exchange between platforms.

The interoperability between these platforms takes place at both the Access and Service layers – generically represented with dashed wide arrows in Figure 13.[66] Basic technical interoperability at the **Access** layer is achieved through API integration, where participants' wallets from one platform can exchange data with another, adhering to each platform's unique API specifications and rules. While the API specifications differ to

[62] In short, the HTLC protocol implements a cross-platform interoperability for atomically exchanging assets without the use of trusted third parties. These intermediaries, that traditionally warrant the atomicity of the transactions, are replaced by smart contract-based escrows that synchronize the rightful releases of previously locked assets.

[63] This appellation still stands even in cases when some DLTs, like Corda, are not modeling their ledger with chained blocks.

[64] The user presentation component is typically implemented as a client application with a User Interface at the Access layer

[65] To preserve the execution determinism of DLT/blockchain based platforms, all the nodes can only process data that is already part of the shared ledger, preventing them from seeking external information that may not be consistent across nodes.

[66] The Asset layer does not share interoperable characteristics in this specific setup (currency token implementations are actually very different on the two platforms), however, other use cases may require interoperability at this layer too.

accommodate diverse platform technologies, some level of syntactic and semantic sharedness would be beneficial for the integration of multiple platforms and efficient wallet development.

The **Service** layer interoperability is enabled by adopting a common protocol for asset exchange between two platforms. Banque de France (Banque de France 2023) showed, in its experimentation series, how various local configurations of the HTLC protocol, customized for each cross-platform case study, can be used to implement multiple asset exchanges. A successful interoperability does not hinge on the convergence of the local protocol implementations,[67] but on agreeing to use the same specification and parameters, such as the secret format, the cryptographic protocol used for hashing the shared secret, the timeframe of asset escrowing on both platforms, or the protocolar steps that participants should follow.

Despite these protocols largely following the same principles, minor configurational differences can still impede the interoperability of these related protocol implementations. Achieving interoperability at scale necessitates sector-wide coordination to standardize these service specifications – including potential variants – to implement and operate compatible versions across multiple platforms.

V. Final Remarks and Next Steps

This paper aims to highlight the need for coordinated technical efforts among various stakeholders in both traditional and emerging financial infrastructure ecosystems. The future of market architecture design is likely to be as complex and varied as the current international monetary system. Alongside ongoing policy and business-level research aimed at simplifying financial processes, there's a growing necessity to address technical challenges emerging from the introduction of new and intricate systems. Key issues such as interoperability across different ecosystems and the challenges of risk assessment in fragmented ecosystems with limited informational models are crucial areas that require attention.

As this work can only be cooperative, the paper proposes the ASAP conceptual model to identify, create or share definitions of new concepts that can improve the traditional approaches for money, financial assets, platforms and their respective capabilities or governance. This work is inspired by the success of models such as TCP/IP in promoting the interoperability of the Internet. In that sense, ASAP attempts to provide the conceptual model necessary for answering the question “What is the TCP/IP of digital assets”. The financial sector can count on its long experience of collaborating to comply with regulation and improving its operations, and strong public national and international organizations can be leveraged to move from theory to reality.

Moving forward, our next steps involve developing an extended informational model at the Asset layer. This model will aim to ensure consistent management of digital assets across platforms, addressing both interoperability and risk management. Additionally, we see significant promise in creating a service description framework at the Service layer. This framework could align disparate service implementations and protocols, offering substantial potential for standardizing practices across the financial sector.

[67] In fact, the local implementation of the protocol (Escrow, HTLC and the orchestration thereof) varies between platforms, according to their inner capabilities and, sometimes, trust assumptions.

References

Adrian, Tobias, Federico Grinberg, Tommaso Mancini Griffoli, Robert M. Townsend, and Nicolas Zhang. 2022. "A Multi-Currency Exchange and Contracting Platform." *IMF Working Papers.* November 4. https://www.imf.org/en/Publications/WP/Issues/2022/11/04/A-Multi-Currency-Exchange-and-Contracting-Platform-525445.

Bains, Parma. 2022. "Blockchain Consensus Mechanisms: A Primer for Supervisors." *IMF Fintech notes*, 4.

Banco central do Brasil. 2022. "Relatorio de Economia Bancaria." 172. https://www.bcb.gov.br/content/publicacoes/relatorioeconomiabancaria/reb2022p.pdf.

Banque de France. 2023. *Wholesale cental bank digital currency experiments with the Banque de France.* https://www.banque-france.fr/system/files/2023-08/Banque_de_France_stabilite_financiere_rapport_mnbc_2023.pdf

BIS Innovation Hub. 2022. "Project mBridge: Connecting economies through CBDC." October. https://www.bis.org/publ/othp59.htm.

BIS Innovation Hub. 2023. *Project Sela: an accessible and secure retail CBDC ecosystem.* September. https://www.bis.org/publ/othp74.htm.

BIS Innovation Hub, Banque de France, Monetary Authority of Singapore, Swiss National Bank. 2023. "Project Mariana: Cross-border exchange of wholesale CBDCs using automated market-makers." June. https://www.bis.org/about/bisih/topics/cbdc/mariana.htm.

Centre for Digital Public Infrastructure. 2023. "Interoperable QR Code." July. https://docs.cdpi.dev/technical-notes/digital-payment-networks/interoperable-qr-code#attributions.

Codruta Boar, Stijn Claessens, Anneke Kosse, Ross Leckow and Tara Rice. 2021. "Interoperability between payment systems across borders." *BIS Bulletin.*

Ethereum Foundation. 2015. "ERC-20: Token Standard." *Ethereum Improvement Proposals.* https://eips.ethereum.org/EIPS/eip-20.

European Parliament and the Council. 2015. "Payment services directive 2." November. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32015L2366.

Financial Stability Board (FSB). 2020. *Enhancing Cross-Border Payments: Stage 3 Roadmap.* https://www.fsb.org/wp-content/uploads/P131020-1.pdf.

IETF, Internet Engineering Task Force. 2022. "Transmission Control Protocol." https://datatracker.ietf.org/doc/html/rfc9293.

ISO - International Organization for Standardization. 1994. "Open Systems Interconnection - Basic Reference Model ISO/IEC 7498-1:1994." https://www.iso.org/standard/20269.html.

MAS, Bank of Canada. 2019. "Jasper–Ubin Design Paper: Enabling Cross-Border High Value Transfer Using Distributed Ledger Technologies."

Monetary Authority of Singapore. 2021. "Multi-CBDCs: Designing a digital currency stack for governability." *by Toh Wee Kee.* https://opennodes.com/2021-04-22-02-40-28-multi-cbdcs-designing-a-digital-currency-stack-for-governability.

Monetary Authority of Singapore. 2023. "Purpose Bound Money (PBM) Technical Whitepaper." https://www.mas.gov.sg/publications/monographs-or-information-paper/2023/purpose-bound-money-whitepaper.

Monethary Authority of Singapore, BIS. 2023. "Project Guardian | Enabling Open and Interoperable Networks." 17. https://www.mas.gov.sg/-/media/mas-media-library/development/fintech/project-guardian/project-guardian-open-interoperable-network.pdf.

Open Banking Limited. 2018. "Open Banking Specification." https://standards.openbanking.org.uk/.

Schär, Fabian. 2020. "Decentralized Finance: On Blockchain- and Smart Contract-based Financial Markets." *FRB of St. Louis Review* 155-156.

PUBLICATIONS

ASAP: A conceptual model for Digital Asset Platforms

Working Paper No. WP/2024/019