BIS Innovation Hub

Bank of England

Project Pyxtrial

Monitoring the backing of stablecoins

Contents

1. Executive summary 3
1.1. Key challenges 5
1.2. Key findings 6

2. Context for Project Pyxtrial 7

3. Stablecoins: an introduction 9
3.1. Stablecoins: an overview 10
3.2. A varied regulatory universe 11
3.3. Stablecoin risks 13

4. Project Pyxtrial 14
4.1. How Pyxtrial fits in the stablecoin universe 15
4.2. How does Pyxtrial compare to other solutions? 16

5. Design decisions, build and delivery 17
5.1. Design decisions 18
5.1.1. Liabilities 19
5.1.2. Assets 19
5.1.3. End users 19
5.2. Pyxtrial's data model and database 20
5.2.1. How Pyxtrial accounts for assets 21
5.2.2. Solving the liability challenge 24
5.3. The complications of on-chain and off-chain data 25

6. Outcomes 27
6.1. How Pyxtrial works in practice 28
6.2. Dashboard development 29
6.3. Insights and lessons learned 30
6.3.1. Five key lessons 30
6.4. Future considerations 31
6.5. What's next? 32

7. Conclusion 33
Acknowledgements 35
References 36

Appendices
Appendix A: Data volume estimates 37
Appendix B: Applying DTI to capture on-chain liabilities across blockchains 38
Appendix C: Stablecoin issuer report data generator 40
Appendix D: IT architecture 41
A-IV.1. IT architecture: From concept to final design 41
A-IV.2. Technical details 48
A-IV.3. Data flows 50
A-IV.3.1. Overview 50
A-IV.4. APIs and integrating data flows 51
A-IV.4.1. Deep dive: Assets data and the API 52

Endnotes 53

Glossary and abbreviations

API
Application programming interface

Backend
Part of software system or application that handles data processing, logic and database interactions

BoE
Bank of England

BIS
Bank for International Settlements

BIS CPMI
BIS Committee on Payments and Market Infrastructures

Cryptoasset
A digital asset (issued by the private sector) that depends primarily on cryptography and distributed ledger or similar technology

Data flow
Process description explaining the flow of data inside application/data warehouse

Data model
Data architecture that defines how data are structured, organised and stored, guiding the database schema and relationships between different data elements

DevOps
Software development methodology combining software development and IT operations, aiming to streamline the software delivery process

DLT
Distributed ledger technology. A means of saving information through a distributed ledger, ie a repeated digital copy of data available at multiple locations

DTI
Digital Token Identifier

EBA
European Banking Authority

FSI
Financial Stability Institute

Global stablecoin
A stablecoin with an existing or potential reach and use across multiple jurisdictions and which could become systemically important in and across one or many jurisdictions, including as a means of making payments and/or store of value

Integration
Process of combining applications or their components to work together seamlessly, enabling data-sharing, communication and interoperability across various technologies or platforms

ISIN
International Securities Identification Number

JSON
JavaScript Object Notation; a-language-independent data format

MiCA
Markets in Crypto-Assets Regulation (EU) 2023/1114 of the European Parliament and of the Council of 31 May 2023 on markets in crypto-assets, and amending Regulations (EU) No 1093/2010 and (EU) No 1095/2010 and Directives 2013/36/EU and (EU) 2019/1937

MMF
Money market fund

Node
Decentralised network application that maintains a copy of the DLT, facilitating the validation, storage and dissemination of transactions across a network

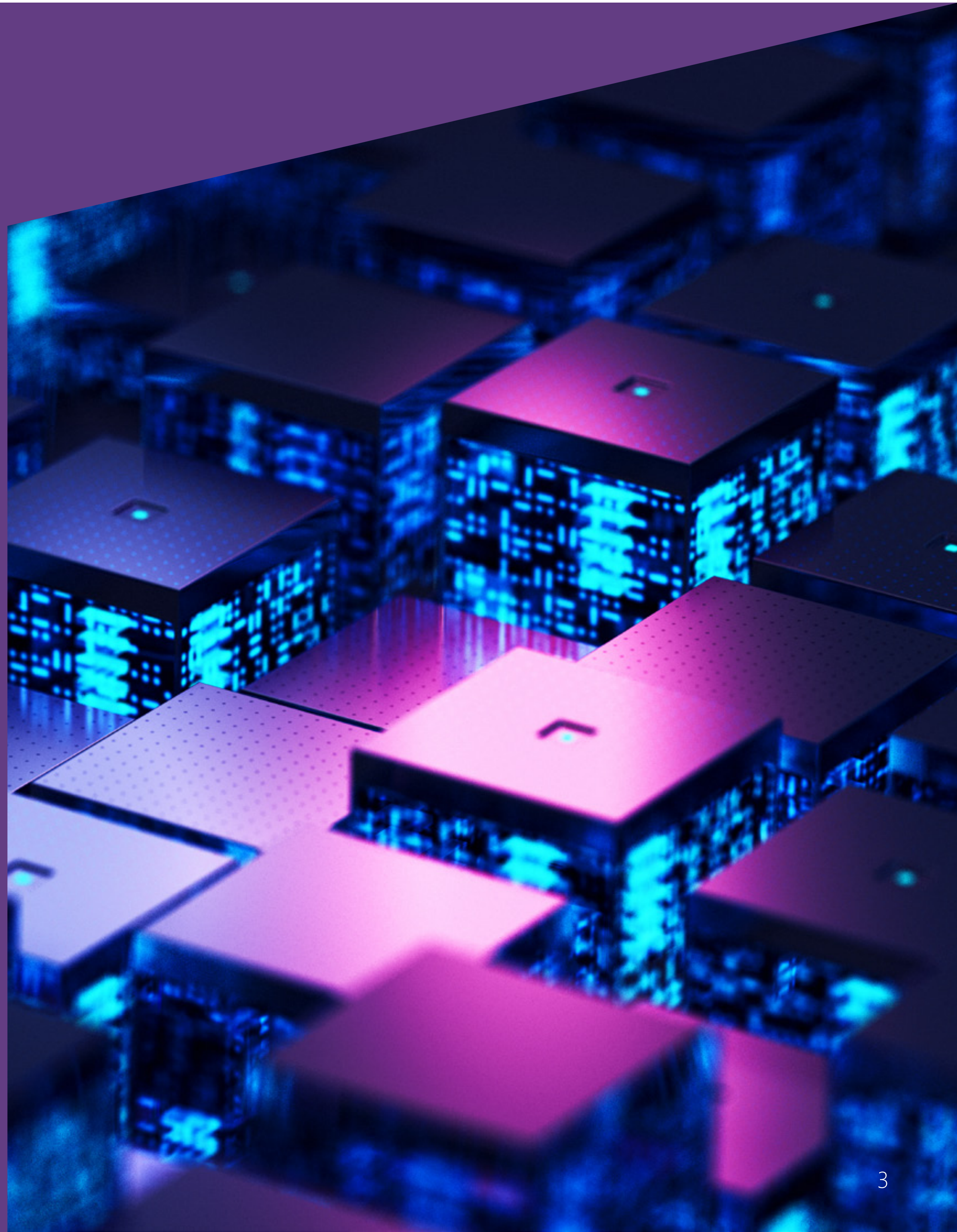
PoC
Proof of concept

REST API
Method for web services to communicate over HTTP, enabling access and manipulation of resources between software systems

Stablecoin
A cryptoasset that aims to maintain a stable value relative to a specified asset, or a pool or basket of assets

Web app
Application that runs on web browsers, allowing users to access and interact with services, data or functionality over the internet

1. Executive summary



Project Pyxtrial is a joint project between the Bank for International Settlements (BIS) Innovation Hub and the Bank of England. Pyxtrial explores how technology solutions can enable the monitoring of asset-backed stablecoins' balance sheets, providing insight into whether the backing assets exceed their liabilities at all times.

To support the supervision of issuers of asset-backed stablecoins, Project Pyxtrial has developed a prototype data analytics pipeline which includes data collection, storage and analysis. The technology is a first step towards a tool that could support supervisors and regulators[1] in proactively detecting issues in stablecoin backing and aid the development of policy frameworks based on integrated data.

"Project Pyxtrial explores how technology solutions can enable the monitoring of the balance sheets of asset-backed stablecoins."

Pyxtrial's development is motivated by several factors: (i) stablecoins are a growing and largely unregulated sector with a market capitalisation of around US$ 162 billion as of June 2024;[2] (ii) the relevant regulations are being developed; and (iii) there is a lack of appropriate supervisory technology (suptech) tools to improve the monitoring of risks that these instruments could create. Pyxtrial represents an important first step in creating a suitable suptech tool.

The value of asset-backed stablecoins is pegged to that of traditional financial assets. Should a mismatch occur between a stablecoin issuer's liabilities (the coins in circulation) and the assets backing that stablecoin, this could undermine confidence in the ability of the issuer to offer redemption at par and prompt a "run", ie a sudden loss of belief in the stablecoin's value.

The backing of stablecoins is critical in maintaining their stable value versus the underlying assets; this provides users with the confidence that they could get redemption in full. An asset-liability mismatch could create risks to financial stability if the stablecoin or its issuer is of systemic importance, or if the loss of confidence becomes contagious and affects the ability to make payments or the financial system more broadly.[3]

Pyxtrial's architecture has three elements: data model, database and APIs.

The data model is Pyxtrial's foundation, defining how data are structured, with the database reflecting data on liabilities and assets that have been retrieved from on-chain (public) and off-chain (non-public) sources, and then storing and processing those data. Pyxtrial makes the most recent data available to its users – the stablecoin issuer's regulator or supervisor – via a dashboard. The testing of Pyxtrial showed it has the capability to support the monitoring of asset-backed stablecoins' balance sheets, with five such stablecoins covered in the proof of concept (PoC) tool. For more information on the database, see Section 5.2.

The use of APIs as a mechanism for data transmission allows Pyxtrial to monitor stablecoin issuers' assets and liabilities without the need to directly connect to the blockchain network. Consequently, Pyxtrial could be adapted to also obtain data from permissioned private chains; however, this capability has not yet been tested. For detailed information on design and technical decisions, see Appendices I–IV.

It is important to note that Pyxtrial does not assess asset quality, the rigour of asset valuations or the quality of the data sources used. Hence there would be a key dependency on the rigour of valuations and the quality of data provided to Pyxtrial, and supervisors may need to play a role in setting out rules or guidance on data quality. Additionally, multiple valuations may exist for a particular asset, for example from different data providers. Functionality for asset valuation would need to be added to Pyxtrial and, should it rely on external data providers, would require them to be connected to the Pyxtrial database by some means, most likely via an API.

> "Pyxtrial showed it has the capability to support the monitoring of asset-backed stablecoins' balance sheets."

1.1. Key challenges

Developing the proof of concept required overcoming two key challenges:

i. Stablecoin regulations are largely a work in progress (for instance, the Bank of England is in the process of finalising its stablecoin regulations). This meant the team had to develop a working solution without having a final framework of rules governing stablecoin issuers. Therefore, the Pyxtrial solution had to have the flexibility to be used once the regulations are finalised and implemented, as well as adapted as regulations evolve. In particular, the ongoing development of regulations meant the data model had to either apply draft or pending stablecoin rules – such as the EU's Markets in Crypto-Assets Regulation (MiCA), which will be phased in by the end of 2024 – to determine how to value the declared underlying assets, or apply standards from other parts of the financial sector to use as a proxy (such as those that apply to valuing money market fund (MMF) assets).

ii. Accurately assessing a stablecoin issuer's assets which back its liabilities. The quality of reporting on such assets (also often referred to as "reserves") varies across stablecoin issuers as there is no standardised or regulatory obligation to do so. These self-declarations vary in granularity and frequency, typically being issued monthly or quarterly depending on the issuer.

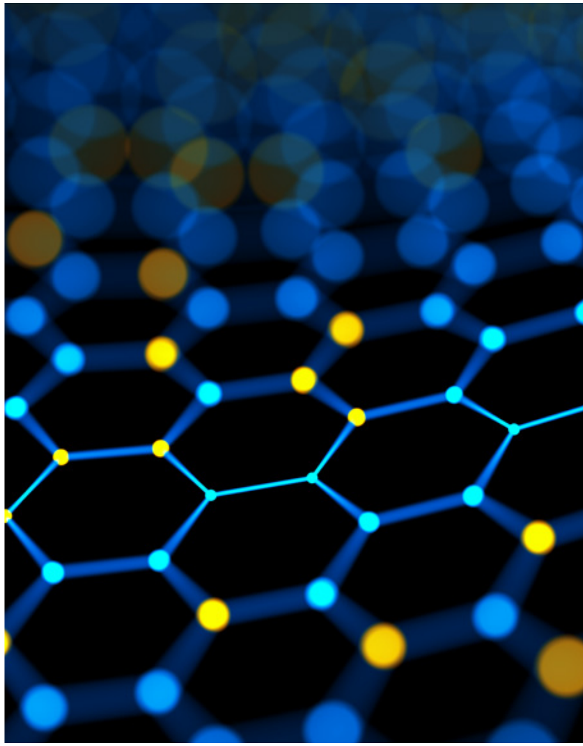
An algorithm was used to match the reporting frequency of the assets to the real-time availability of the liabilities-side data. This algorithm analysed the declared assets' opening and closing balances, then allocated the change in asset values suitably.

While changes in the value of assets can be tracked during the intervals between issuers' reports being published, it is not possible to track the composition of those assets, which might lead to an over-identification of shortfalls. Therefore, a working assumption of Pyxtrial is that stablecoin issuers would, at some future date, be required to provide such data in a specified format and at a fixed frequency.

1.2. Key findings

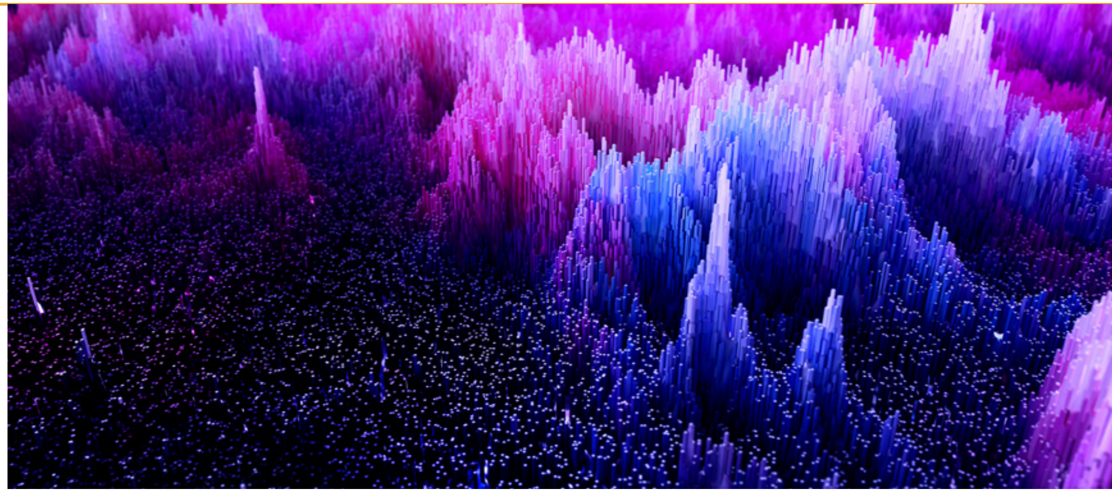
Pyxtrial's key findings are:

- **Pyxtrial provides a PoC monitoring tool that could support supervision of issuers of the selected digital assets.**[4] By collecting and processing both on-chain and off-chain data, Pyxtrial would enable the monitoring of digital asset balance sheets, thereby supporting the supervision of stablecoins.
- **Regulators must have the requisite in-house skills.** The operation and implementation of Pyxtrial will require a multidisciplinary team. Regulators will need to ensure that their supervisors possess the required skills for sourcing on-chain data.
- **Further work is required before Pyxtrial can be rolled out.** Pyxtrial is a reporting and monitoring platform. It anticipates supervisory needs by providing a template for future regulatory reporting tools that will be needed to monitor stablecoins. It is a PoC and not a ready-made product.
- **If Pyxtrial were to be implemented, it would facilitate risk monitoring.** Pyxtrial enables earlier identification of potential risks as supervisors have access to timelier stablecoin data in a standardised format. Additionally, Pyxtrial could bring value to cross-border supervisory cooperation as it enables host regulators to assess the backing of an overseas-issued stablecoin.
- **Flexibility is a fundamental feature (and requirement).** Pyxtrial was designed to be adaptable. Its technical components (the APIs, data model, data storage solution and dashboard) are modular so regulators and central banks can switch them out or build upon them. In this way, regulators can integrate Pyxtrial into existing systems, enhance it and adapt it to their current and future needs. Additionally, supervisors have the ability to adjust Pyxtrial to account for a higher frequency of attestation reporting in order to meet regulatory needs for more frequent reporting by stablecoin issuers, or to implement APIs, allowing them to pull data from firms' systems when needed.



2. Context for Project Pyxtrial

Project Pyxtrial is a joint experiment between the BIS Innovation Hub and the Bank of England. It explores how technology solutions can enable the monitoring of the balance sheets of asset-backed stablecoins.



In recent years, the total market capitalisation of asset-backed stablecoins has rapidly increased, nearly tripling in size from around US$ 56 billion in early 2021 to US$ 162 billion in June 2024;[5] consequently, regulators are closely monitoring developments.

At present, stablecoins are most often used as a bridge between crypto and fiat currencies, for settling transactions (eg buying or selling cryptoassets, making cross-border payments) or to store value in cryptoasset markets. While there are proposals for wider stablecoin use in everyday payments (eg USD Circle (USDC) or PayPal's PYUSD stablecoin), currently such use is limited in the United Kingdom. Nevertheless, stablecoins pegged to a single currency have the potential to be used at greater scale as a means of payment and thus become a systemic element of payments in the UK economy (BoE (2023)).

While still relatively small, stablecoins have become a growing part of the global financial system, with major financial services firms providing services for stablecoin settlement or even issuing their own stablecoins. Notably, in April 2024, PayPal announced that users of its Xoom service in the US could use its PYUSD stablecoin for cross-border money transfers.[6] Also in April, Ripple announced that it planned to launch a USD-denominated, asset-backed stablecoin later in the year.[7] For its part, Visa in 2023 expanded its stablecoin settlement capabilities using USDC on the Solana and Ethereum blockchains.[8]

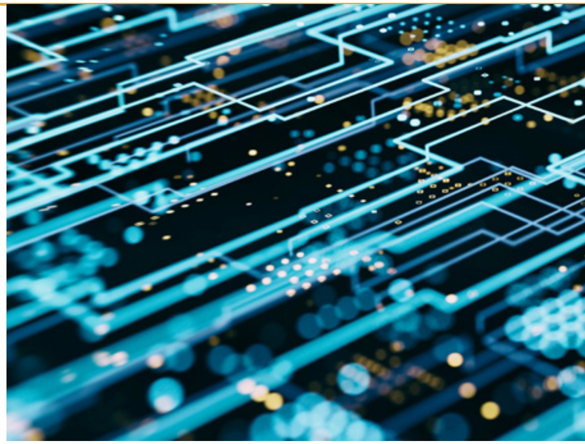
Should the use of stablecoins increase, regulators will increasingly require a system that checks that the assets backing individual stablecoins exceed their liabilities, which is what Pyxtrial focused on. That is crucial because protections for stablecoin holders, where they exist, are founded on issuers having a defined, ringfenced set of backing assets. Consequently, regulators need to know that a stablecoin's liabilities are fully backed at all times. In turn, this means those backing assets are held in a safeguarded account and are always at least equal to the amount of stablecoins that have been issued to the blockchain ledger.

Should stablecoins become more widely used for payments, there is a possibility that one or more could eventually become systemically important. That makes it prudent to prepare the ground for measuring those backing assets and checking that they exceed the issuer's liabilities, for systemic and non-systemic stablecoins alike. Pyxtrial has been designed to offer regulators and supervisors such a solution.

> "Regulators need to know that a stablecoin's liabilities are fully backed at all times."

3. Stablecoins: an introduction

Stablecoins are a form of digital asset that purports to maintain a stable value relative to a fiat currency by holding assets (which may be of variable value) as backing.



3.1. Stablecoins: an overview

Although the term "stablecoin" is commonly used by market participants and authorities, there is no universally agreed legal or regulatory definition for it.[9] Notably, use of the term should not imply that a stablecoin's value is in fact stable;[10] history shows that this is sometimes not the case.[11]

Stablecoins are underpinned by distributed ledger technology (or blockchain), a technology that allows a network of participants to set up and keep agreement on a shared set of evolving facts without needing to fully trust each other.

Stablecoins may be issued on either permissionless or permissioned blockchains. The first form describes an open network that uses advanced algorithms and on which anyone can validate data and transactions. Permissioned blockchains, on the other hand, are a closed network and consequently do not require algorithms that are as complex because there is less need to secure consensus between participants to overcome trust issues.

While the majority of stablecoins currently in issuance are on permissionless blockchains, stablecoins that mainstream banks would issue would likely be held on permissioned blockchains. As Pyxtrial is not part of the blockchain network but uses universal connectors (APIs), it can in principle work with both types.

Stablecoins may use various approaches to maintaining parity with their peg. A distinction can be made between the following four types of stablecoin based on whether they claim to hold a pool of reserve assets to back their value (ie whether they are collateralised or not) and, if so, the type of reserve asset:[12]

- **Fiat-backed stablecoins:** Stablecoins that claim to be backed by assets denominated in a fiat currency. They are also known as asset-backed stablecoins. Examples include USDT and USDC.[13]
- **Crypto-backed stablecoins:** Stablecoins that claim to be backed by other cryptoassets. Examples include Dai and Frax.
- **Commodity-backed stablecoins:** Stablecoins that claim to be backed by commodities. Examples are PAX Gold and Tether Gold.
- **Unbacked stablecoins:** Stablecoins that do not claim to be backed by any reserves, but rather seek to maintain a stable value through, for instance, algorithms or protocols. They are also known as algorithmic stablecoins. Examples include TerraClassicUSD and sUSD.

Pyxtrial's focus is on fiat-backed/asset-backed stablecoins which, should they become systemically important, might pose a financial stability risk.[14]

> "Pyxtrial is not part of the blockchain network but uses universal connectors (APIs), so it could work with many data sources."

3.2. A varied regulatory universe

In the wake of the sudden collapses of prominent stablecoins in 2022 and 2023,[15] there has been increased regulatory focus, with a range of work taking place at the international level:

- In July 2022, BIS's Committee on Payments and Market Infrastructure (BIS CPMI) and the International Organization of Securities Commissions (IOSCO) published joint guidance on stablecoin arrangements. This document applies the principle of "same risk, same regulation" to stablecoins and confirms that the Principles for Financial Market Infrastructures[16] apply to systemically important stablecoin arrangements that transfer stablecoins.[17]
- In December 2023, the Basel Committee on Banking Supervision (BCBS), the primary global standard setter for the prudential regulation of banks, released a consultation paper on banks' exposure to stablecoins.[18] That followed the issuance of its prudential standard on banks' exposure to cryptoassets – which include stablecoins – a year earlier;[19] its implementation is scheduled for January 1, 2026.[20]
- The Financial Stability Board (FSB) has an ongoing project examining the global requirements that should apply to stablecoins. That follows the July 2023 publication of its high-level recommendations on how systemically important stablecoins (which it refers to as global stablecoins) should be regulated and supervised.[21]

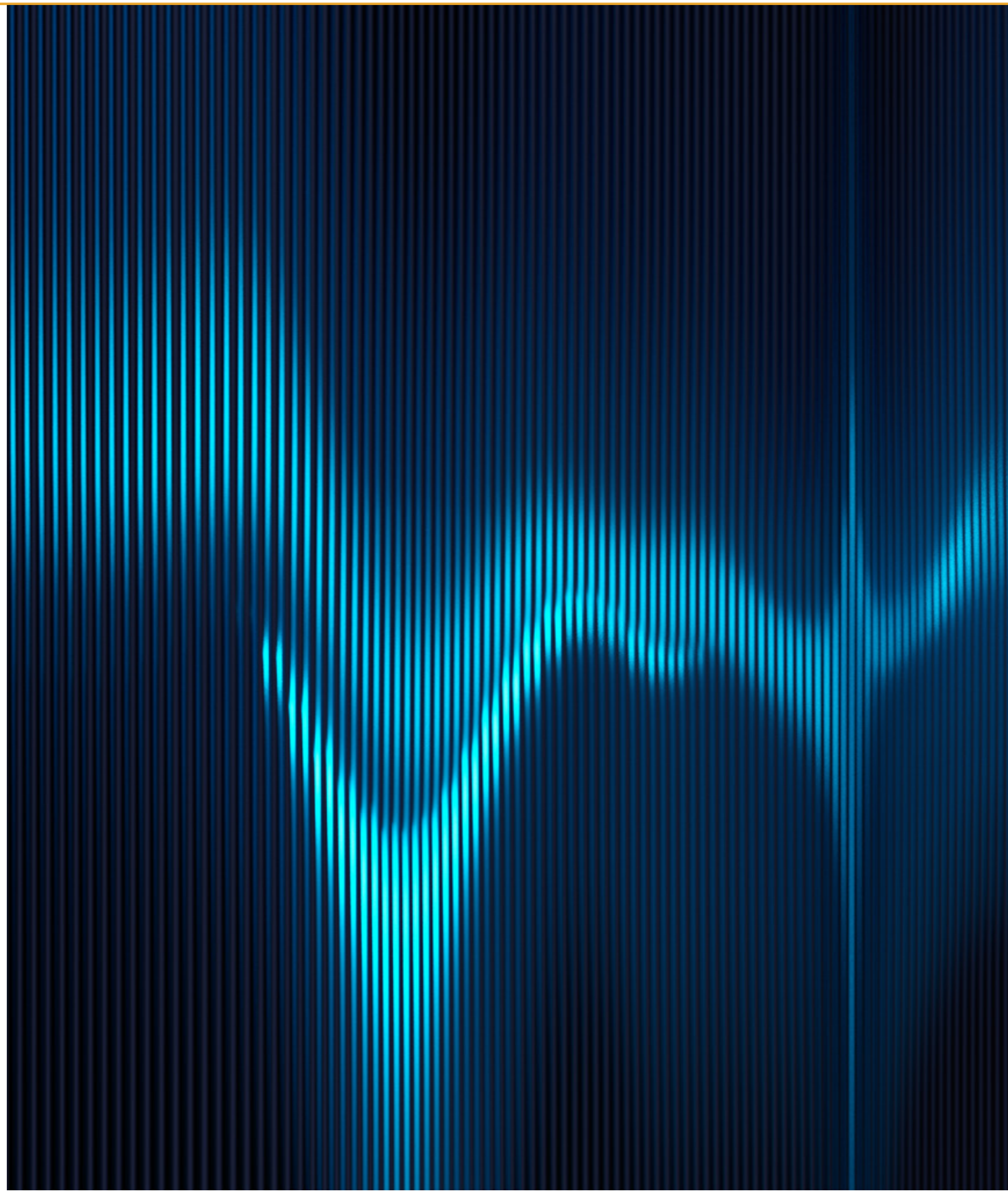
At the same time, many individual jurisdictions – for example the United Kingdom,[22] the European Union[23] and the United States[24, 25] – are developing or implementing stablecoin regulations.[26]

Stablecoin regulatory approaches in major jurisdictions have similar key requirements for stablecoin issuers. Most follow two types of authorisation regime, permitting stablecoin issuance by commercial banks and specific non-bank financial institutions and/or by a new type of designated entity in possession of a crypto-specific licence.[27] It is widely required that issuers maintain reserves equivalent to the value of stablecoins in circulation in segregated accounts.

In general, regulations stipulate rules on prudential, governance and risk management, with disclosure requirements concerning anti-money laundering and counter-terrorism financing (AML/CTF), as well as consumer protection provisions relating to information provided to stablecoin holders.

However, there are notable differences in the terminology used to define stablecoins, the regulatory treatment of reserves, and their segregation and custody, as well as stablecoin holders' rights of claim and the treatment of redemption fees. As such, there is a risk that differences in regulatory frameworks may lead to inconsistencies in overseeing stablecoins across different jurisdictions (Crisanto et al (2024)).

This complexity illustrates some of the challenges that the development of Pyxtrial faced. As a prospective suptech tool, Pyxtrial must anticipate the changing nature of the stablecoins market. Although existing regulatory requirements were known for certain jurisdictions (for instance, the United Kingdom and the European Union) and although there was an indication as to where draft regulations in others are headed, informed guesses about the regulatory direction had to be made.



3.3. Stablecoin risks

A stablecoin classed as systemic carries two main risks, both of which stem from the concept of the singleness of money. This term means all forms of money should have the same value, be generally accepted as a means of payment and be interchangeable without loss of value with all other forms of money used in the economy.[28]

The first risk is that a loss of confidence among holders that they would be able to redeem the stablecoin at par value could see its market value drop sharply. Such a run might be sparked by uncertainty around the assets that are meant to back the stablecoin, eg whether the assets exist in the first place or whether the value of those backing assets is undermined due to financial risks specific to them (Ahmed et al (2024)).[29]

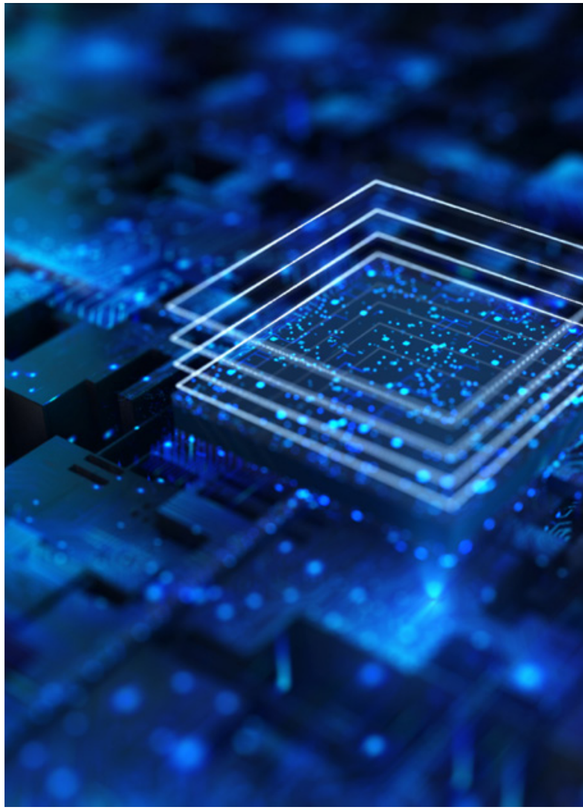
A run on the stablecoin could lead to rising pressure on the ability of the issuer to maintain the stated value of the stablecoin. This happened when USDT and USDC temporarily depegged from the US dollar in May 2022 and March 2023, respectively.[30, 31]

With stablecoins, compromises of the singleness of money could come in different forms: frictions, such as delays or costs; disruptions to the ability to make payments; an inability to exchange them for other forms of money; and a lack of confidence in the ability of the issuer to fully meet redemption requests. Ultimately, a lack of confidence could cause a run on the stablecoin as large numbers of coin holders seek to redeem their holdings. In turn, the issuer's capacity to redeem stablecoins might be overwhelmed, which could trigger its failure. Were this to happen to a systemic payment system that uses stablecoins, the ability of users to make payments could be disrupted, with a loss of confidence in money and payments, and in the financial system more broadly, as consequences.

That broader impact constitutes the second major risk.[32]

A suptech tool that is able to monitor the assets and liabilities of stablecoins, some of which may become systemic in the future, in near real-time could mitigate both risks: first, providing clear warnings to supervisors by flagging potential problems with assets; and second, monitoring issuers' liabilities to track sudden increases in demand.

> "A suptech tool that is able to monitor the assets and liabilities of stablecoins in near real-time could mitigate risks."



4. Project Pyxtrial

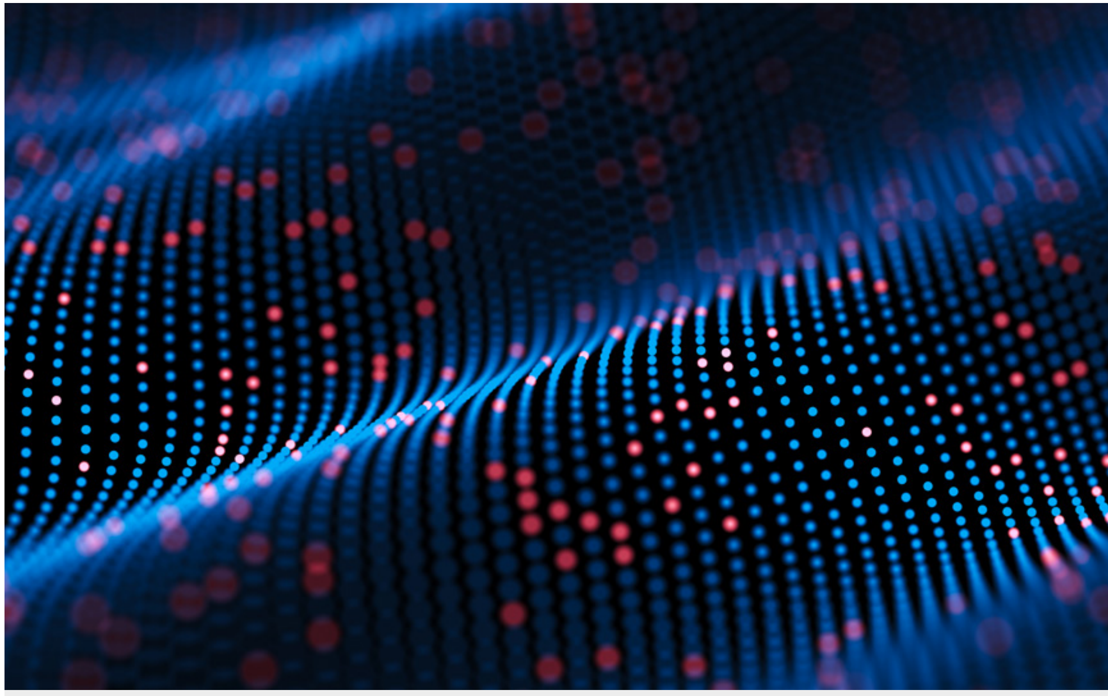
The Pyxtrial suptech tool would provide supervisors with an automated way to see each stablecoin's liabilities-side and assets-side data and flag any inconsistencies between its backing assets and liabilities.

4.1. How Pyxtrial fits in the stablecoin universe

The fundamental challenge of monitoring stablecoins concerns the assets that back their liabilities and, linked to this, how their composition changes over time. Pyxtrial delivers a PoC tool that can overcome those concerns and that allows near real-time monitoring of how a stablecoin issuer's assets and liabilities are changing.

To achieve this, Pyxtrial is designed to pull on-chain data about liabilities hourly from the various blockchains, or platforms, on which stablecoins are issued and then either receive or pull data on a matching frequency basis from stablecoin issuers about their backing assets. In this way, supervisors could get a near real-time snapshot of every stablecoin issuer's balance sheet.

Currently, Pyxtrial has a core limitation: all assets-side data are self-disclosed by stablecoin issuers and are provided infrequently (often monthly or quarterly). Additionally, the classification of assets, rather than adhering to a common standard, is often self-defined and at a high level rather than being sufficiently granular. These limitations mean some stablecoin issuers could window-dress their balance sheets, and this will remain a shortcoming as long as self-declared publicly available data remain the only source.



4.2. How does Pyxtrial compare to other solutions?

Pyxtrial is a tool for supervisors taking a prudential approach to the management of risk associated with systemically important stablecoins.

Specifically, it delivers a microprudential solution – one that assesses individual stablecoin issuers – rather than a macroprudential solution, which would examine the entire stablecoin market. There are few microprudential solutions available today because the market is too small for vendors that build technology solutions (Prenio et al (2024)).

"Pyxtrial is a microprudential solution."

Pyxtrial differentiates itself by applying a prudential policy perspective to stablecoins to gauge their impact on financial stability. Even though it is currently not possible to source these data more frequently than stablecoin issuers provide it (typically monthly or quarterly), stablecoin issuers may in the future update their assets-side data more frequently and in a format that allows for comparability and proper assessment.

5. Design decisions, build and delivery

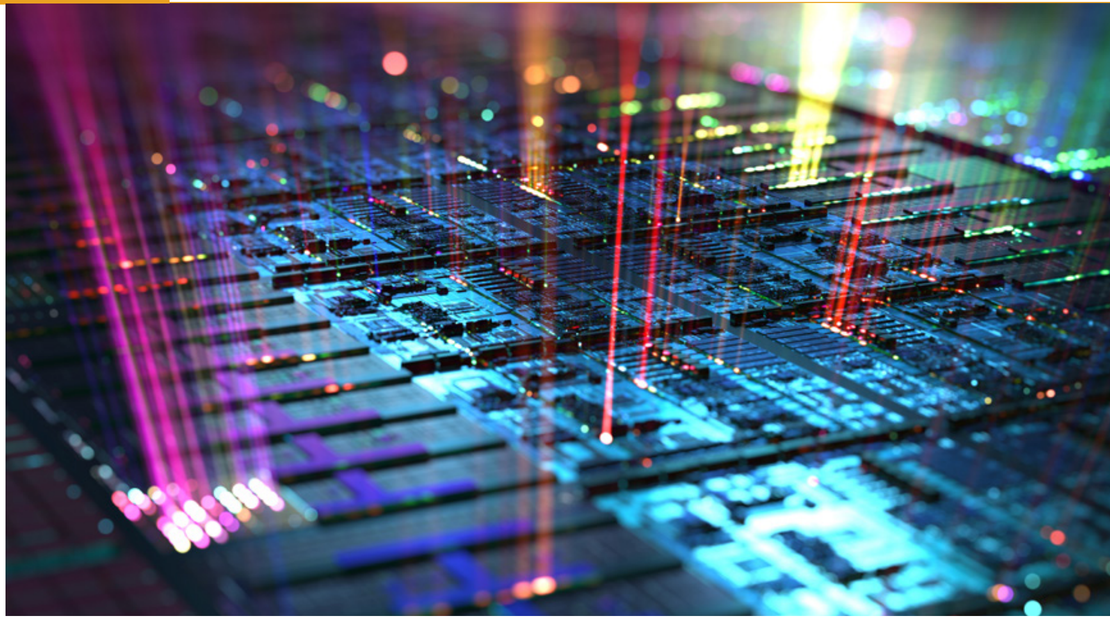
Building Pyxtrial required making several assumptions, while maintaining flexibility in the final shape of the tool.

5.1. Design decisions

Several crucial assumptions underpin Pyxtrial.

First is that the user of the tool would have the enforcement authority to request the data about stablecoin assets and liabilities at the requisite intervals. This might include a requirement to submit data to supervisors or publish to the general public at a regular frequency, or to integrate APIs to allow supervisors to "pull" data when they have reason to do so. Pyxtrial works with regulated issuers and is not currently designed for non-regulated issuers. Relying on stablecoin issuers to provide timely, accurate data to regulators or supervisors on a voluntary basis is unlikely to be a workable solution. The application would not receive sufficient data for its supervisory purposes if it relied only on publicly available data, as that is inadequate.

Second, the team developed a placeholder data model to build the reporting system. This was necessary since there is no regulatory finality covering Pyxtrial's scope. Typically, technical engineers would have an outline of the data requirements (the supervisory reporting schema or template) that would inform the creation of a data model to underpin the reporting system. With no established regulations, however, the placeholder data model is informed instead by published examples of stablecoin reporting templates (for example, by the FSB)[33] and MMF supervisory requirements.[34] The latter are viewed as being like stablecoins in many respects (Anadu et al (2023)), which makes those supervisory templates relevant.



The third assumption is that users of Pyxtrial are not interested in transaction-level data. Capturing every payment would generate huge volumes of granular data that are not needed for the purpose of monitoring backing. Pyxtrial's approach has the advantage of greatly simplifying its design and implementation as it minimises the number of data pipelines needed and makes it easier to maintain.

A fourth assumption relates to data on the assets-side. While liabilities-side data are on-chain and are taken from blockchains on an hourly basis, the data on the assets-side are part-simulated. The starting point for assets-side data is the information on assets that the stablecoin issuer publishes. However, different issuers publish at different frequencies (biweekly, monthly or quarterly) which do not match the hourly data gathering used on the liabilities side.

Consequently, the team used extrapolation algorithms that take these publicly available assets-side data and, by incorporating random noise, generate them on an hourly basis to match the liabilities-side data. In this way, the team generated data of a sufficient quality and frequency that could flow through the pipelines and prove that the concept works.

The fifth key assumption relates to the number of stablecoins Project Pyxtrial assesses and the number of blockchains from which it gathers liabilities-side data. As a PoC, Pyxtrial did not select all stablecoins or all blockchains, nor were any stablecoin issuers or infrastructure providers invited to work with the project. Rather, it uses five of the largest stablecoins – USD Tether (USDT), USD Circle (USDC), Binance USD (BUSD), Pax Dollar (USDP) and TrueUSD (TUSD) – and tracks their liabilities data on five blockchains: Ethereum, Binance Smart Chain (BSC), Avalanche, TRON and Solana. Because stablecoins are implemented as distinct tokens on each blockchain, and because Pyxtrial did not assess all five stablecoins on each of the five blockchains, this meant Pyxtrial analysed 13 representations of these five stablecoins across the five public blockchain networks. However, this approach is scalable to other blockchains and stablecoins.

Lastly, given that part of Pyxtrial's approach is flexibility, the system is designed to gather assets-side data in two ways: it can wait to receive that information, or it can connect proactively and retrieve it directly from the stablecoin issuer. This experimental feature represents a shift from current regulatory reporting approaches where the firms send data to supervisors, and could see supervisors extract information on demand at any time.

5.1.1. Liabilities

Pyxtrial performs regular on-chain data collection. Every hour, the blockchain scanner's API automatically serves data about the circulating supply of tokens – the five stablecoins across five blockchains (see Table 1 in A-IV.4). There is no International Securities Identification Number (ISIN) that identifies specific stablecoin tokens on different blockchains, which complicates on-chain retrieval of the liabilities. Pyxtrial resolves this by using Digital Token Identifier (DTI) codes (see Appendix B). In this automation, which is driven by a sub-scheduler within the Python application, the connectors pull data from API endpoints from those public on-chain scanners.

Importantly, the integration layer can also pull liabilities data from the stablecoin issuers' systems. While this complicates the design, it is useful as it allows supervisors to compare and verify on-chain liabilities against the liabilities that stablecoin issuers self-declare. In this way, a supervisor can see whether an issuer is issuing a number of liabilities onto the blockchain that differs from the amount reported to the regulator or announced to the public.

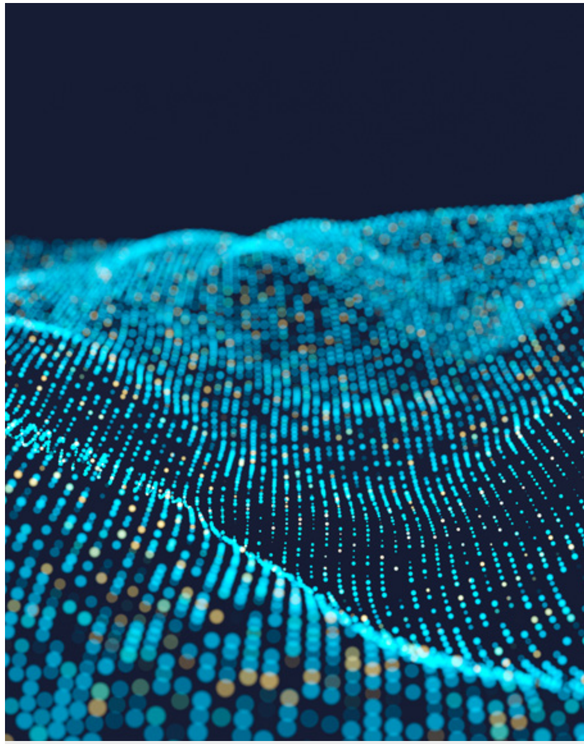
5.1.2. Assets

Pyxtrial's API is sent off-chain data on assets via the Pyxtrial REST API endpoint. This endpoint receives stablecoin issuers' transparency reports in a structured format that is defined by Pyxtrial.

Additionally, a separate connector was implemented that, instead of being sent generated data, can pull assets data from the stablecoin issuer. While not deployed in a production environment, it demonstrates that usage of such an approach is feasible.

5.1.3. End users

Pyxtrial's API endpoint provides the most up-to-date view of the collected data. The API fetches the most recent data from every entity in the core layer and prepares the target view model for the data consumer. In this way, supervisors can see the assets and liabilities for each stablecoin.



5.2. Pyxtrial's data model and database

While the APIs represent the inflows and outflows of data in the Pyxtrial PoC, the data model is its foundation.

Development of the template utilised existing work developed by the New York State Department of Financial Services (NYDFS) for its stablecoin supervision. The team augmented this to develop Pyxtrial's data model with inputs from international supervisors and experts.

The final data model used in Pyxtrial seeks to reflect the different attributes related to the assets and liabilities of stablecoin issuers' balance sheets, and in that way identify any mismatches. It mirrors the real-life assets and liabilities as follows:

- It breaks down **assets** into two levels of aggregation: asset classes and sub-asset classes. For example, cash is one asset class, and the data model has separate subclasses covering cash on hand, cash at the central bank and deposits of reverse repos (Pyxtrial has the flexibility to add further asset and sub-asset classes as needed).
- Then the sub-assets were extended with specific attributes. For debt securities, that required understanding the valuation method and the debt maturity.
- On the **liability** side, it seeks to reflect the current token supply. As noted earlier, it does this by retrieving that information from different platforms by using block explorers. The DTI taxonomy was used to reflect the intricacy of tokens tied to the underlying infrastructure and link them under the stablecoin level of aggregation, including reference information related to the issuer, unit of account and fungibility mechanism.

Other considerations were also factored in, including the features of those platforms, whether access to each platform is permissionless or not, where the stablecoin issuer is authorised, where it operates or who provides custody for the stablecoin issuer's liabilities.

> "The final data model used in Pyxtrial seeks to reflect the different attributes related to the assets and liabilities of stablecoin issuers' balance sheets."

Next, the sub-asset classes' attributes were mapped with regulatory sources to ensure that valuations follow established regulatory procedure where possible. In this, the following bank-related financial reporting standards and guidance notes were incorporated:

- EBA reporting framework 3.0, template code: F0101, as outlined in the Commission Implementing Regulation (EU) 2021/451 of 17 December 2020, extending the financial reporting standards to stablecoin issuers and ensuring alignment with best regulatory practices;
- Commission Implementing Regulation (EU) 2018/708 of 17 April 2018, which provides technical standards for MMF managers[35] (given the similarities in the nature and governance of MMFs and stablecoins, the annex of this regulation serves as a relevant reference);
- high-level recommendations for the regulation, supervision and oversight of global stablecoin arrangements issued by the Financial Stability Board (FSB) on 17 July 2023,[36] offering relevant guidance tailored specifically to stablecoins; and
- guidance on the issuance of US dollar-backed stablecoins, published by the New York State Department of Financial Services (NYDFS) on 8 June 2022,[37] providing further insights and recommendations pertinent to stablecoin issuance.

Pyxtrial's approach adheres to established standards and best practices to enhance regulatory compliance and oversight within the stablecoin ecosystem.

5.2.1. How Pyxtrial accounts for assets

Table 1 shows how Pyxtrial accounts for assets; in this case, how the data model factors in the opening transaction date of a reverse repo asset listed in a stablecoin issuer's report, to which it applies the regulatory provisions of the EBA's reporting framework 3.0.

Table 1: The data model's treatment of the transaction date for a reverse repo asset

Entity name	Attribute name	Description	Regulatory source
assetRrp	txDate	The date when the repo transaction begins	• EBA reporting framework 3.0, template code: F0101; Commission Implementing Regulation (EU) 2021/451 of 17 December 2020 laying down implementing technical standards for the application of Regulation (EU) No 575/2013 of the European Parliament and of the Council with regard to supervisory reporting of institutions and repealing Implementing Regulation (EU) No 680/2014.

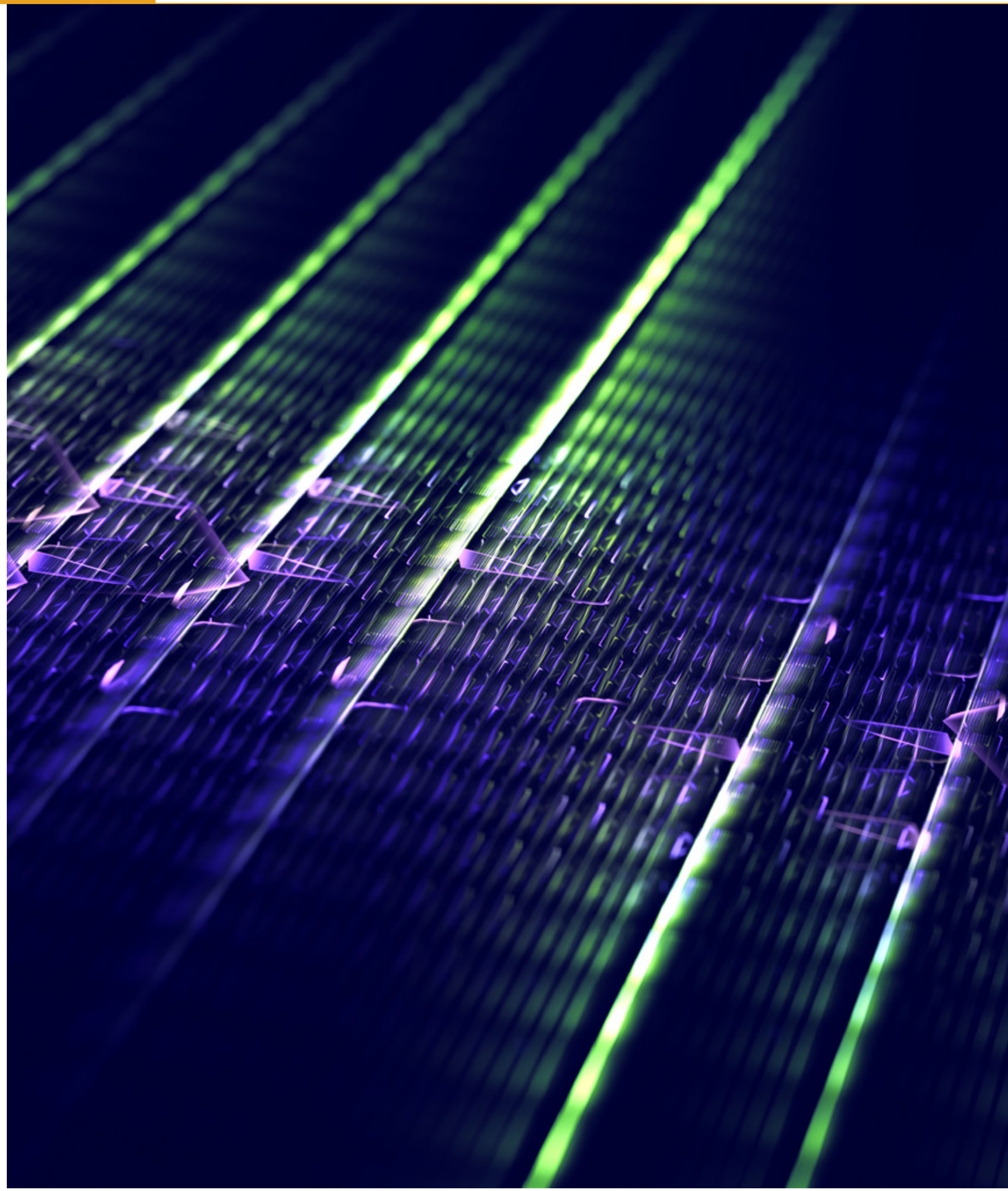
This approach is replicated across the data sources, including for every sub-asset class, every liability-related attribute and for the key players themselves. In short, every class has specific attributes, which are linked to the relevant regulatory source (or sources, where more than one might apply).

In practice, this operates in the following way. When using valuations as an example, Pyxtrial's data model incorporates key regulatory provisions, including from the EU's MiCA regulation, and is also consistent with the standards set out in BCBS's prudential treatment for banks' exposure to cryptoassets.[38] The latter sets out standards for banks' treatment of certain cryptoassets on their balance sheets.[39]

> "Pyxtrial's data model incorporates key regulatory provisions."

Another asset-related aspect considered by the data model is an assessment of the asset's characteristics: assets should be high-quality and liquid, and should carry minimal risk as this would mean a stablecoin issuer could convert them rapidly in the event of an asset-liability mismatch. To this end, Pyxtrial incorporates regulatory tags that rank assets by quality and liquidity. Cash deposits or sovereign debt, for example, are ranked as more liquid and of higher quality than debt issued by a private institution. This functionality enables the monitoring of compliance with the requirements on the composition of assets as well as their value.

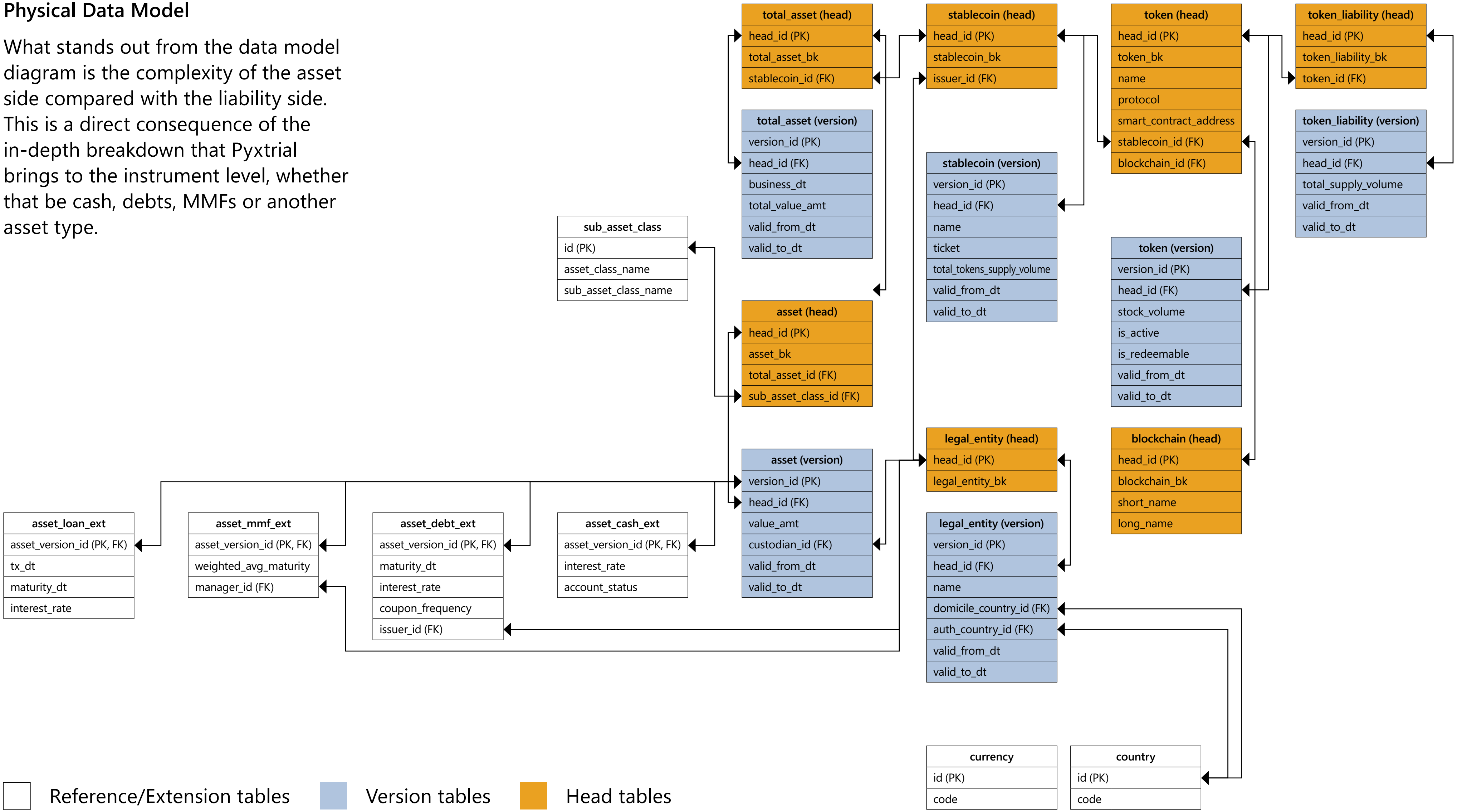
All this feeds into the data model (Graph 1), which lists the main business entities, reference data and extension tables (these provide extension attributes for specific asset classes like cash), as well as the relationships between them.



Graph 1: Pyxtrial's core data model

Physical Data Model

What stands out from the data model diagram is the complexity of the asset side compared with the liability side. This is a direct consequence of the in-depth breakdown that Pyxtrial brings to the instrument level, whether that be cash, debts, MMFs or another asset type.



5.2.2. Solving the liability challenge

As an established taxonomy for liabilities does not exist, one option considered was to use traditional regulatory resources for non-stablecoin liabilities and extend these. Another was to base the approach on documentation from the FSB or articles tailored to stablecoins. Both approaches were combined.

A separate liabilities-related challenge involved the difficulty of identifying each stablecoin on different blockchains. Take USDT: although USDT does have an ISIN, its 12-digit code is not used on Ethereum or Solana or other blockchains to identify it. This means that although USDT on Ethereum has the same value as USDT on Solana, and although ISIN does recognise USDT, applying the ISIN will not capture USDT's liabilities across the different platforms.

To resolve this, Pyxtrial uses the DTI methodology in accordance with ISO 24165 and the recommendation endorsed for implementation within the ESMA DLT Pilot Regime[40] (see Appendix B for further details).

5.3. The complications of on-chain and off-chain data

Finalising Pyxtrial required overcoming challenges associated with on-chain data (which are publicly available) and off-chain data (which are not publicly available).

The first challenge came at the start of the project, when it was assumed that the two sides of the balance sheet would separate into on-chain data (liabilities) and off-chain data (assets). However, a significant amount of liabilities-related data are also off-chain, eg flags indicating that the issuer currently supports the token (is it active?) and its fungibility (is it convertible?).

Given that these data are likely to remain predominantly off-chain, this means a fully on-chain supervisory process is unlikely. Whilst supervisors already work extensively with off-chain data, some might need to improve their on-chain sourcing capabilities.

A second challenge regarding on-chain and off-chain data concerns their frequency. It is likely that supervisors will find that on-chain liability data can be pulled more often than the stablecoin issuer sends off-chain asset data. The result is that reporting frequencies will differ, which will give rise to discrepancies. One solution is to use thresholds to introduce flexibility; these would ignore differences that fall below a certain risk level and flag those that exceed it. A stablecoin issuer whose discrepancy breached the regulatory risk level might, for instance, need to submit their asset declaration more regularly.

Box A: Pyxtrial and Project Atlas

As a way of bringing greater flexibility, the team tested how Pyxtrial could obtain data from sources owned or provided by public authorities. This would provide an alternative to Pyxtrial's current use of commercial data providers.

One such public provider is Project Atlas,[41] which is being developed by the BIS Innovation Hub.

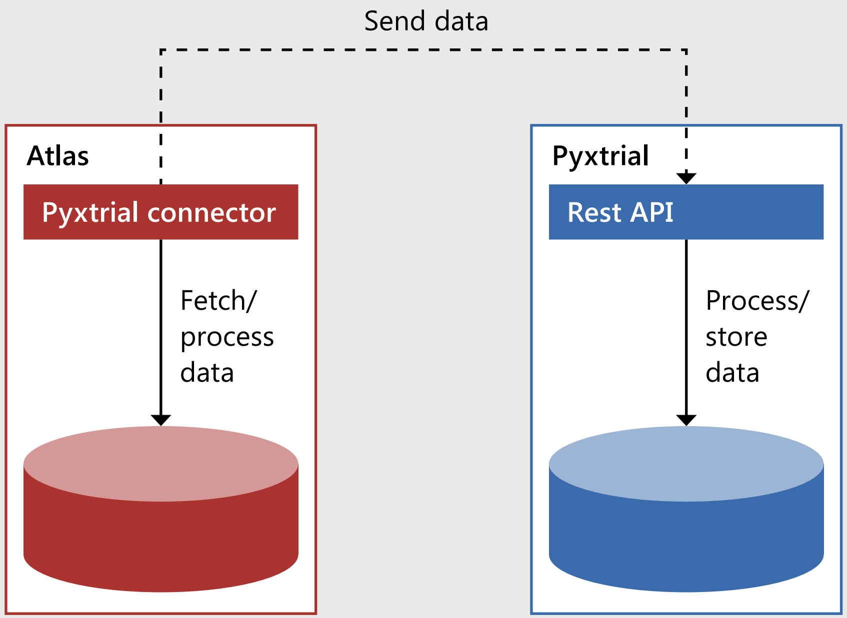
Atlas is a data platform that provides cryptoasset data that are tailored to the needs of central banks. Where Pyxtrial and Atlas differ is that Atlas gathers transaction-level data, which requires a big data infrastructure, while Pyxtrial is a regulatory reporting tool that transmits data to supervisors.

Pyxtrial and Atlas were tested to see whether they could connect and exchange information between their platforms (Graph A1).

This involved defining a Pyxtrial API to receive information about Bitcoin from Atlas. Atlas extracted information from its internal database, processed and aggregated those data, and successfully sent it to Pyxtrial via the API using a scheduler.

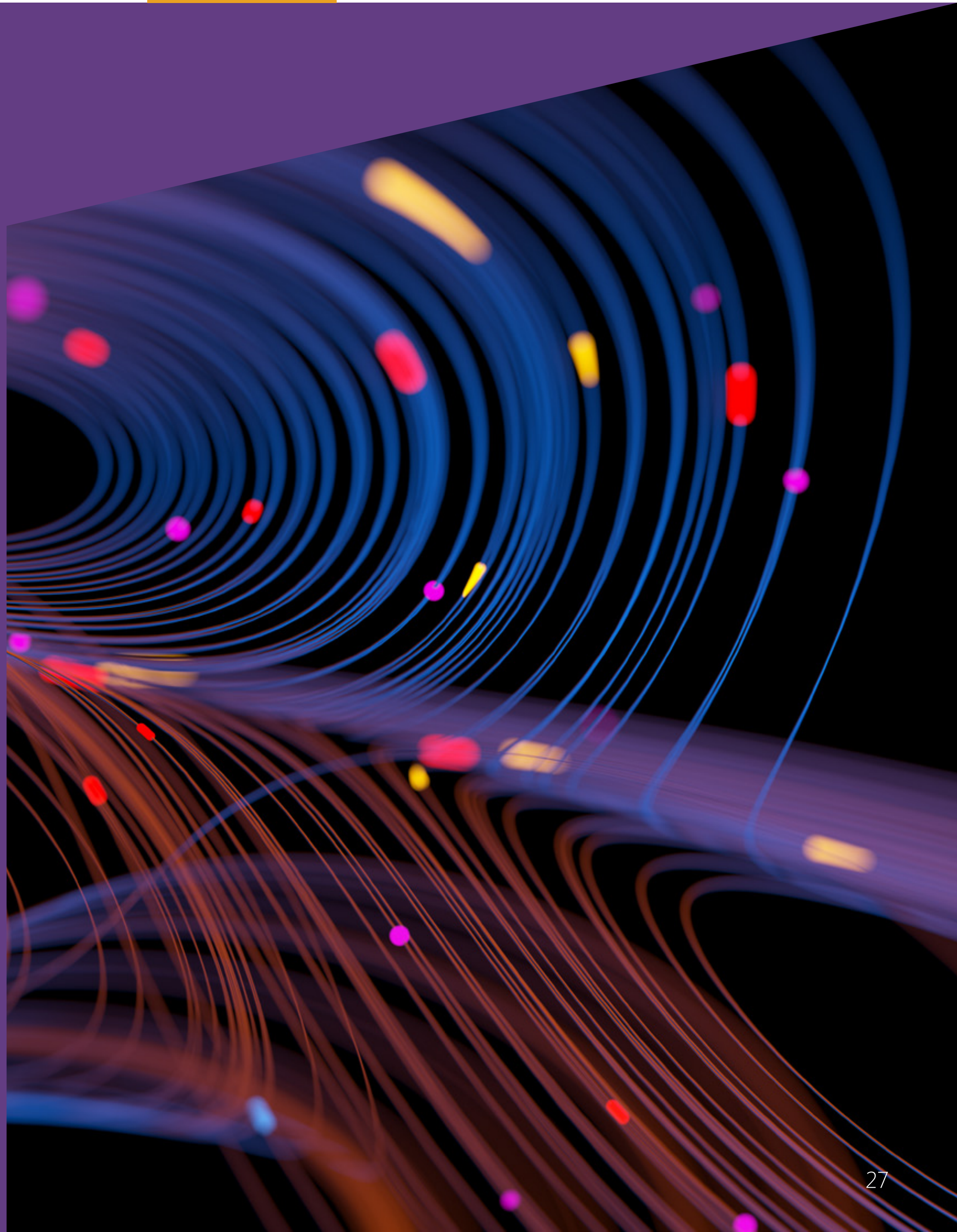
Because Atlas's capability could be expanded to include other data points, including stablecoins, the platform could, if required, provide scheduled and aggregated stablecoin data to Pyxtrial.

Graph A1: Schema of the data connection between Pyxtrial and Atlas



6. Outcomes

Pyxtrial shows how supervision of stablecoins, or of similar digital assets, could work in the future, and what would be required to implement it.



6.1. How Pyxtrial works in practice

Pyxtrial is designed to be positioned between the settlement ledger and the issuer, to retrieve the necessary information both from the blockchain and from the issuer (either actively or passively), and to generate an oversight report that shows the risks that a stablecoin could pose to the financial system.

This process starts when the integration layer connects to blockchain scanners (such as etherscan.io) to pull liabilities-side data from their API endpoints. At the same time, a generator provides the assets-side data, whose time frame is currently simulated.

The data are transformed and enriched in the processing layer, which also manages the database transactions, while the persistence layer provides low-level functionality that allows interactions with the database such as the storage and retrieval of data.

Collected data are stored in the PostgreSQL database, as are the raw JSON data that Pyxtrial has fetched, with the raw data processed and stored in target tables in the database.

Finally, the collected data are served to consumers via the second API endpoint.

The ability to pull data on demand gives further flexibility to supervisors should they need that information more frequently. Additionally, supervisors can use both push and pull approaches on the liability side, which lets them compare the publicly visible liabilities and those that the stablecoin issuer has sent to the regulator.

6.2. Dashboard development

Pyxtrial's user interface provides an option for supervisors to interact with the collected data by giving them the ability to monitor their quality and offering an overview of stablecoin balance sheets.

It can also show detailed information about assets, liabilities and their mismatches. Additionally, it can provide the latest data, offer the ability to focus on a specific time period, and illustrate relative changes over time.

While Pyxtrial's dashboard is fully functional, it is also replaceable. This allows regulators, many of which use commercial products to build dashboards for internal solutions, to switch out the supplied version for one that better meets their needs.

As with the rest of Pyxtrial's architecture, its modular approach means regulators can customise the existing dashboard to best suit their requirements.

6.3. Insights and lessons learned

Pyxtrial's key contribution is that it supports understanding of the risks underpinning stablecoins.

As regulations evolve, supervisors will have the option to request a regular report from stablecoin issuers about their assets and liabilities, as well as related information such as issuance, fungibility and redemption mechanisms. In this way, supervisors can understand how an issuer's balance sheet is built and the risks it might contain. Additionally, they could – depending on the use case – employ customised analysis based on their specific needs.

Importantly, Pyxtrial also shows that that when supervisors and technologists work together, they can overcome the challenges that technology can put in the way of creating solutions.

Finally, the PoC has proven that supervisory technology can be built in parallel with evolving regulatory requirements, and that such an approach can help regulators to determine which elements a solution needs to be effective.

6.3.1. Five key lessons

Of the numerous lessons that Project Pyxtrial generated, the following five stand out.

The balance sheets of asset-backed stablecoins can be supervised

Even though digital assets constitute a relatively new field, some of them – like stablecoins – can be supervised by monitoring their balance sheets. Pyxtrial does this by monitoring the assets and liabilities of the issuers behind asset-backed stablecoins through a platform that collects on-chain and off-chain data. Additionally, this process can be organised so that supervisors can have near real-time monitoring. One caveat is that, although the data points gathered by Pyxtrial were accurate for the PoC, it is possible that additional data would be needed in future to cover issuers' entire balance sheets.

Implementation and operational requirements

Successfully deploying Pyxtrial requires that regulators use a multidisciplinary team to implement and operate it. Supervisory skills fit for stablecoin supervision will have to encompass knowledge of cryptoassets and information technology, in particular data engineering and how data are transmitted.

Anticipating supervisory needs

While Pyxtrial anticipates supervisory needs, it is not a ready-made product. Instead, it is a reporting platform that provides a template for future regulatory reporting tools that are essential for supervisors.

Benefits for supervisors

Adopting Pyxtrial can support supervisors in receiving more frequent and fully automated reports. This enhances the efficiency and responsiveness of the monitoring process, which helps supervisors to respond faster to potential risks.

Modularity and reusability

Pyxtrial is designed so that its technical components (the APIs, integration layer, data model, data storage solution and dashboard) are modular, reusable and can be repurposed. It is intended to be adapted to specific regulatory needs and integrated into existing systems, while having the flexibility to be enhanced in future.

> "Supervisory technology can be built in parallel with evolving regulatory requirements."

6.4. Future considerations

A key challenge in monitoring the balance sheets of stablecoin issuers is the significant disconnect between the availability of liabilities-side data, which is in effect in real-time, and that of the assets-side data, where stablecoin issuers' reports might be published only monthly or even quarterly.

For a stablecoin deemed to be systemic (or the equivalent, depending on the jurisdiction), that is unsustainable from a regulatory point of view. This is why a mechanism would need to be in place to ensure stablecoin issuers supply data about their backing assets at a far higher frequency and in a more consistent manner than is currently the case.

Whether such a mechanism would be sufficient for Pyxtrial to function would depend on the regulations themselves, including the required frequency and the granularity of assets' reporting. Pyxtrial is not a replacement for regulation, so regulators would need to make sure that the necessary data were collected and that they could be reported at the required frequency. That might well vary for different assets.

Additionally, asset valuation is not straightforward and often requires a certain amount of manual effort, which is more complex for high-frequency reporting situations. As asset and liability valuation was not part of the scope of Pyxtrial, implementing this in practice is an aspect that is worth further discussion and, if external data sources were used, would likely require them to connect via an API to Pyxtrial's database to transmit data. Pyxtrial also does not assess asset quality, so it will retain a dependency on the data quality of stablecoin issuers or third-party providers.

Given that stablecoin liabilities are spread over different platforms, tracking the total liabilities requires pulling on-chain information from every blockchain that the stablecoin is on. Therefore, it would be useful if stablecoin issuers segregated their assets according to the liabilities those assets collateralise, and if they did so by platform. Moreover, given that the amount of stablecoins minted does not always match the amount in circulation, regulators would need to define how they want to treat both types of liability. Pyxtrial's flexibility means it can connect to specified wallets and that the precise requested fields can be adjusted.

The sharing of data is also worth considering, because some data might have an individual institutional data (IID) tag, which would make them confidential. As a result, users of Pyxtrial might need specific legal arrangements in place to share those data, including doing so in a manner that avoids complications when it comes to the growing need for international cooperation.

Finally, if stablecoins become more widely used, the subject of cross-border issuance and cross-border monitoring will become pressing, which could further add to the value of using Pyxtrial. Indeed, cross-border issuance of stablecoins is already under way: Circle, which issues USDC, also issues a euro-denominated coin called EUROC in the European Union[42] while Paxos has issued a USD-denominated stablecoin in Singapore (though it is subject to the Monetary Authority of Singapore's stablecoin regulations).[43] In situations like this, Pyxtrial could prove vital, with a stablecoin's home jurisdiction providing data to regulators in those jurisdictions where it is being used, thus allowing cross-border monitoring of the issuer's balance sheet.

> "Pyxtrial is not a replacement for regulation, so regulators would need to make sure that the necessary data were collected and that they could be reported at the required frequency."

Pyxtrial is not a replacement for regulation, so regulators would need to make sure that the necessary data were collected and that they could be reported at the required frequency.

6.5. What’s next?

Pyxtrial has shown that a rapid approach to developing a suptech solution that focuses on digital assets can succeed.

With a functional PoC, regulators and other interested parties can begin testing it in their environment. And, because Pyxtrial was designed as a modular, customisable solution that can be adjusted as business models and regulations change, they can modify, expand and amend it as they see fit to meet their purposes.

A test-and-use approach would provide valuable information on both the user experience and the utility and value that Pyxtrial offers. It could also show what Pyxtrial enables supervisors to achieve in practice and highlight any additional functionalities and refinements.

“Pyxtrial was designed as a modular, customisable solution that can be adjusted as business models and regulations change.”

7. Conclusion

Project Pyxtrial has shown it can provide supervisors with near real-time data about stablecoins' liabilities and their backing assets. It also has the potential to monitor other tokenised products that are backed by real-world assets.

Project Pyxtrial had two key objectives:

- to consider from first principles a technology-first approach to the supervision of digital assets that might come within regulatory scope, as such a solution would alleviate supervisory pain points around the irregular, historical and time-bound way in which data are received; and
- specific to stablecoins, which are at the interface between traditional finance and cryptoassets, to test the hypothesis of how to start supervising them and to see whether this approach could provide a useful roadmap for learning how to supervise this new type of asset class.

The development has shown how a rapidly developed technological solution can support regulators by tracking liabilities and assets in near real-time. In this way, supervisors would be able to monitor whether stablecoins are always fully backed.

Pyxtrial has also provided insights into the efforts needed to build a backend solution that can enable supervisors to engage with on-chain and off-chain data in a way that helps them meet their remit.

While Pyxtrial was developed specifically for stablecoins, it might prove useful for other digital assets that are backed by real-world assets – as has begun to happen with, for example, US Treasuries or tokenised securities[44] – as its approach mirrors what would be needed: ensuring that the on-chain liabilities of a tokenised security are matched by its off-chain assets. On-chain data for stablecoins, therefore, might well include assets as well as liabilities.

While Pyxtrial has proven successful as a PoC, implementing it as a solution will bring its own challenges.

One reason is that regulators and supervisory authorities around the world have different technology setups. For this reason, Pyxtrial used a modular, customisable approach. Another is because implementing and operating Pyxtrial as a working solution, like any other business tool, requires the right business change talent in place to communicate to, engage with and support the staff who will use it. Like any other PoC, Pyxtrial needs to be further tested and refined before it can be deployed. This process is under way.

Ultimately, the goal is to inspire supervisors and other authorities to consider how they can customise Pyxtrial and extend its use cases to encompass other digital assets, as well as to motivate new approaches to supervision.

> "Pyxtrial has shown how a rapidly developed technological solution can support regulators."

Acknowledgements

Project sponsors

Francesca Hopwood Road, London Centre Head, BIS Innovation Hub

Matthew Osborne, Senior Manager, Innovation and Payments Policy Team, Bank of England

Irina Mnohoghitnei, Head of Fintech Hub, Bank of England

Bank for International Settlements

Project Lead/Adviser: Eryk Walczak

IT: Rudolf Biczok, Darko Micic, Michael Stutz

Legal Adviser: Simona Lambrinoc

Technical Adviser: Micah Smith

Reviewers: Maha El Dimachki, Andras Valko, Cecilia Skingsley, Denise Garcia Ocampo (Financial Stability Institute)

Bank of England

Board Members: Chiranjit Chakraborty, Jennifer Enwezor, Amy Lee, Will Parry

Data Scientists: Matthew Leong, Lucia Resca, Hien-Anh Tran

Reviewers: Sasha Mills, Tom Mutton, Elina Rage

Institutions involved in the workshop

Bank of England,
Bank of Israel,
European Central Bank,
European Banking Authority,
Hong Kong Monetary Authority,
New York State Department of Financial Services,
Saudi Central Bank

References

Ahmed, R, I Aldasoro and C Duley (2024): "Public information and stablecoin runs", *BIS Working Papers*, no 1164, January.

Anadu, K, P Azar, M Cipriani, T Eisenbach, C Huang, M Landoni, G La Spada, M Macchiavelli, A Malfroy-Camine and J Wang (2023): "Runs and flights to safety: Are stablecoins the new money market funds?", *Federal Reserve Bank of New York Staff Report*, no 1073, September.

Bank for International Settlements (BIS) (2024): Recommendations for the regulation, supervision and oversight of global stablecoin arrangements – executive summary, February.

Bank of England (BoE) (2021): "New forms of digital money", *Discussion Paper*, June.

——— (2023): "Regulatory regime for systemic payment systems using stablecoins and related service providers", *Discussion Paper*, November.

Basel Committee on Banking Supervision (BCBS) (2022): Prudential treatment of cryptoasset exposures, December.

——— (2023): "Basel Committee consults on targeted adjustments to tighten its standard on banks' exposures to cryptoassets", December.

——— (2024): "Governors and Heads of Supervision reiterate commitment to Basel III implementation and provide update on cryptoasset standard", May.

BIS Committee on Payments and Market Infrastructures (BIS CPMI) and International Organization of Securities Commissions (IOSCO) (2022): Application of the Principles for Financial Market Infrastructures to stablecoin arrangements.

Cornelli, G, S Doerr, J Frost and L Gambacorta (2023): "Crypto shocks and retail losses", *BIS Bulletin*, no 69, February.

Crisanto, J, J Ehrentraud and D Garcia Ocampo (2024): "Stablecoins: regulatory responses to their promise of stability", *FSI Insights on policy implementation*, no 57, April.

European Securities and Markets Authority (ESMA) (2023): Markets in Crypto-Assets Regulation (MiCA).

European Union (2018): Commission Implementing Regulation (EU) 2018/708 of 17 April 2018 laying down implementing technical standards with regard to the template to be used by managers of money market funds when reporting to competent authorities as stipulated by Article 37 of Regulation (EU) 2017/1131 of the European Parliament and of the Council.

Financial Stability Board (FSB) (2023): High-level recommendations for the regulation, supervision and oversight of global stablecoin arrangements – Final report, July.

Kosse, A, M Glowka, I Mattei and T Rice (2023): "Will the real stablecoin please stand up?", *BIS CPMI Papers*, no 141, November.

New York State Department of Financial Services (NYDFS) (2022): "Virtual currency guidance: Guidance on the issuance of US dollar-backed stablecoins", June.

Prenio, J, A Pustelnikov and J Yeo (2024): "Building a more diverse suptech ecosystem: findings from surveys of financial authorities and suptech vendors", *FSI Briefs*, no 23, May.

Uhlig, H (2022): "A luna-tic stablecoin crash", *NBER Working Paper*, no 30256, July.

US Securities and Exchange Commission (SEC) (2023): Money market fund reforms; Form PF reporting requirements for large liquidity fund advisers; Technical amendments to Form N CSR and Form N-1A.

Appendix A: Data volume estimates

As part of Pyxtrial's architecture design process, the team defined and analysed the expected data volumes, data formats and data collection rates. This section briefly describes the approach that was used.

The Pyxtrial PoC relies on two primary source types: for liabilities, the public blockchain scanners' APIs; for assets, the simulated stablecoin issuer reports.

The following table lists the general metrics of the public blockchain scanner sources for gathering the token liabilities data.

Number of stablecoins	5	Defined by initial scope (USDT, USDC, BUSD, TUSD, USDP)
Number of blockchains	5	Defined by initial scope (ETH, BNB, TRX, AVAX, SOL)
Max number of tokens	25	Stablecoins are implemented as tokens on blockchains. Theoretically: tokens=stablecoins×blockchains
Max size of fetched data for one token (bytes)	104 bytes	This size derived from initial redundant liabilities data structure mapped onto PostgreSQL table and types
Collection rate	Hourly, 25 calls/hour	Data is pulled each hour for every token in scope to determine the token liability data
Max size of fetched data for all tokens daily (KB)	62.4 KB	data volume daily=tokens×token data size×24
Max size of fetched data for all tokens yearly (MB)	22.8 MB	data volume yearly=data volume daily×365

The actual number of tokens in scope is around half the theoretical one: 13 tokens.

The following table lists the general metrics of the simulated stablecoin issuer reports for gathering the stablecoin assets-related data.

Number of stablecoins	5	Defined by initial scope (USDT, USDC, BUSD, TUSD, USDP)
Max size of assets data for one stablecoin (KB)	2 KB	This size derived from initially proposed redundant assets data structure mapped onto PostgreSQL tables and types
Collection rate	Hourly, 5 calls/hour	Every hour receiving of assets-related data is expected for every stablecoin in scope
Max size of fetched data for all coins daily (KB)	245.8 KB	data volume daily=coins×assets data size×24
Max size of fetched data for all coins yearly (MB)	89.7 MB	data volume yearly=data volume daily×365

Stablecoin issuers typically publish transparency reports biweekly, monthly or even quarterly. This means the estimated annual volume of assets-related report data provides solid thresholds as well as the annual volume of token liabilities data. The total annual increment is expected to be around 120 MB, which means big data capabilities are not required to support implementation of the solution.

Appendix B: Applying DTI to capture on-chain liabilities across blockchains

Pyxtrial uses the Digital Token Identifier (DTI) methodology in accordance with ISO 24165 and the recommendation endorsed for implementation within the ESMA DLT Pilot Regime.[45] Using DTI offers the following benefits:

- **Transparency:** It identifies the digital token, its location on a public or private blockchain, and its link to underlying assets, if relevant.
- **Interoperability:** It is a standardised method that distinguishes between different ledgers and tokens.
- **Uniqueness:** DTI is based on the digital token's unique and verifiable origins on the distributed ledger data structure.
- **Consistency:** It can distinguish original, legitimate tokens from newly created digital tokens following fork events.
- **Scalability:** A unified token identifier method supports scalability and resilience to the currently fragmented cryptoasset industry.

This approach allowed Pyxtrial to overcome the limitations of using ISINs to represent stablecoin tokens, as the example of USDT shows:

- **Stablecoin:** USDT
- **Financial Instrument Short Name (FISN):** USD TETHER/USDT
- **International Securities Identification Number (ISIN):** XTL09Q657BK6
- **Digital Token Identifiers (DTIs):**

i. **Functionally Fungible Group (FFG):** L09Q657BK

ii. **Tokens within FFG:** 75T0GP5WJ, 2QWSBDMNC, NJ7X4BCTD, PJ1BC8W44, C9N6ZVN7S, Z7ZX774BJ, FZP9CBTG1, PDX13MN94, BHQ70PZ11, 6QBMW6DQZ, JGJZPPF7Z, NSTDLD1N5, TZFM5MHWR, 29DD3R2SH

DTIs provide a more precise and comprehensive representation of USDT tokens because every token within the FFG is a USDT token. In this way, Pyxtrial provides the granularity necessary to accurately identify USDT's liabilities across different platforms. Table B.1 shows how this works for each of the five stablecoins in scope.

Table B.1: How Pyxtrial uses DTIs for accurate on-chain identification of stablecoins

ISIN	FISN	CFI	FFG	Linked DTI(s)	DTI Long Name	DTI Short Name	DTI Status	Type
XTL09Q657BK6	USD TETHER/ USDT	TMXXXX	L09Q657BK	(L09Q657BK); 75T0GP5WJ; 2QWSBDMNC; NJ7X4BCTD; PJ1BC8W44; C9N6ZVN7S; Z7ZX774BJ; FZP9CBTG1; PDX13MN94; BHQ70PZ11; 6QBMW6DQZ; JGJZPPF7Z; NSTDLD1N5; TZFM5MHWR; 29DD3R2SH	USD Tether	USDT	Validated	Referential Instrument
XTTJWK5QTRK6	USD COIN/ USDC	TMXXXX	TJWK5QTRK	(TJWK5QTRK); CK9PW1MFH; M8G59S1ZR; 5R3XFPZSM; DWBBXD2F1; T6TQ8GN7L; XTS66R22W; 45J6XJX7H; P7H2X1B9H	USD Coin	USDC	Validated	Referential Instrument
XTBSF74H7GR2	BINANCE USD/BUSD	TMXXXX	BSF74H7GR	(BSF74H7GR); MJ4JJGJ97; DDWNJQZZH; HRBN31W17; FDBG7XVCM; BDFVNSJD2; 41W3DC2R4; T8HCTG9W6	Binance USD	BUSD	Validated	Referential Instrument
XTQ00DC0PHH4	TRUEUSD/ TUSD	TMXXXX	Q00DC0PHH	(Q00DC0PHH); 4GNLB78VC; JVVVH6K29; DLFDPGMTP; N19HT013T; 393P69V6L	TrueUSD	TUSD	Validated	Referential Instrument
XT8G4JTC9D06	PAX DOLLAR/ USDP	TMXXXX	8G4JTC9D0	(8G4JTC9D0); RF4V6BXGW; QLRMVKR7K	Pax Dollar	USDP	Validated	Referential Instrument

Finally, in anticipation of future supervisory needs, Pyxtrial's developers created the data model to be flexible enough to increase the level of complexity of the incorporated attributes. In this way, it will be able to comply with new regulatory needs, thus meeting another key requirement of the project.

Appendix C: Stablecoin issuer report data generator

Currently, stablecoin issuers provide transparency reports that are mostly in the form of PDF files. These reports are published at different frequencies, monthly on average. Every stablecoin issuer follows a specific report structure and granularity. Some report assets and liabilities amounts at the aggregated and sub-aggregated values level, while others also provide amounts breakdowns with respect to specific sub-asset classes or even specific financial instruments (eg US Treasuries assets for USD Coin reports).

To support Pyxtrial PoC development, the stablecoin issuer report data generator was introduced to generate plausible transparency report data based on real historical reports on the stablecoins in scope. The generator is implemented as a separate application that exposes an API endpoint for pulling stablecoin report data on demand.

This application serves end-to-end testing, benchmarking and development purposes exclusively. It is not intended to be a part of the production environment.

The following table contains paths to the API endpoint for development and user acceptance testing (UAT) environments (refer to **/docs** endpoints for detailed API documentation):

DEV	pyxcoinapi-dev.bisih.local/assets/{stablecoin}
DEV	pyxcoinapi-dev.bisih.local/docs
UAT	pyxcoinapi-uat.bisih.local/assets/{stablecoin}
UAT	pyxcoinapi-uat.bisih.local/docs

The generator serves report data in a special format defined as part of the Pyxtrial PoC. The format contains aggregated total assets amounts as well as assets breakdown structures with respect to multiple sub-asset classes (eg cash, debt, loans, MMFs).

The total assets value amount, as well as the sub-asset classes allocation weights, are modelled based on the collected transparency report data for a historical period of a year (where applicable) for every stablecoin in scope. The generator draws the total assets values from normal distribution with expectation equal to the total assets value mean of a particular stablecoin report's data and considerably small value of standard deviation to introduce fluctuations.

The generator also supports the generation of token liabilities data that are served within the same format in a separate tokens section. While the allocation weights of liabilities value amount per token are plausible, the liabilities value amount itself is not accurate and based on the total assets value.

Appendix D: IT architecture

A-IV.1. IT architecture: From concept to final design

Pyxtrial's goal was to develop a reporting analysis tool that can match the on-chain liabilities (the tokens that have been issued) with the assets that stablecoin issuers declare underpin those liabilities, and then report those to the end users (Graph D.1).

This required that the prototype had the following functions:

- a dedicated API endpoint that collects the stablecoin issuer's transparency report data on its assets (for the purposes of the prototype, these data are generated; however, once applied in a real-world setting, this could be a regulatory requirement, which is a core hypothesis of the Pyxtrial project) and can also collect the liabilities data from the stablecoin issuer itself, which allows the user to compare that information with the on-chain liabilities data;
- connection to blockchain scanners' public APIs to collect liabilities data from on-chain sources (the blockchain network scanners);
- the ability to store the raw data, then process them and load them into a core data layer (the target tables); and
- a second API endpoint that allows access to these data to consumers for further analysis and presentation.

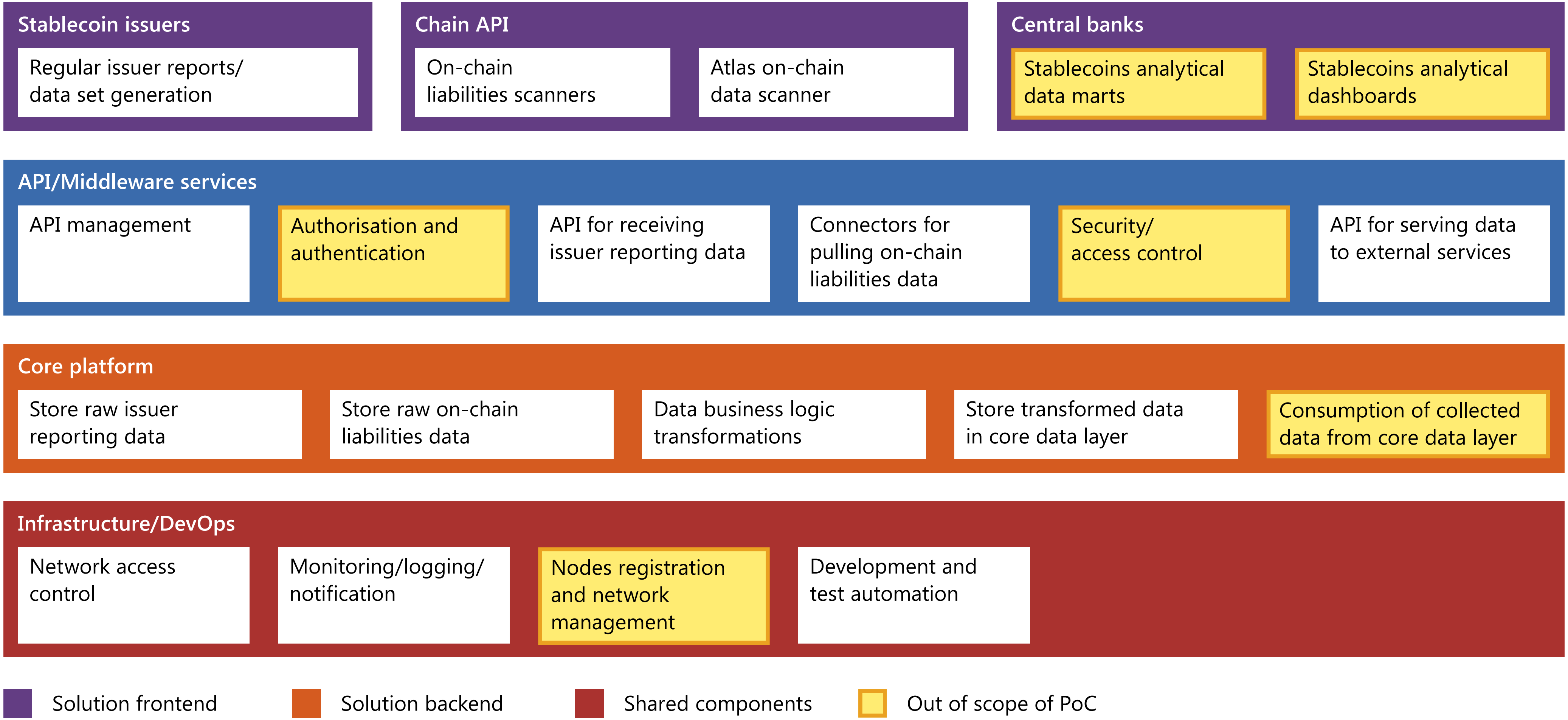
An alternative solution (Graph D.2) would use a big data approach, ie one suitable for processing very large data volumes. This would collect the assets and liabilities data using the same conceptual approach, but with big data (streaming) technologies.

This approach was not selected for the Pyxtrial PoC as it comes with its own challenges, in particular the need to orchestrate its multiple moving parts. These include Kafka solutions for the integration layer, Apache Ozone for the storage solution, Apache Spark for processing, and Delta Lake for the data lakehouse capabilities' enablement.

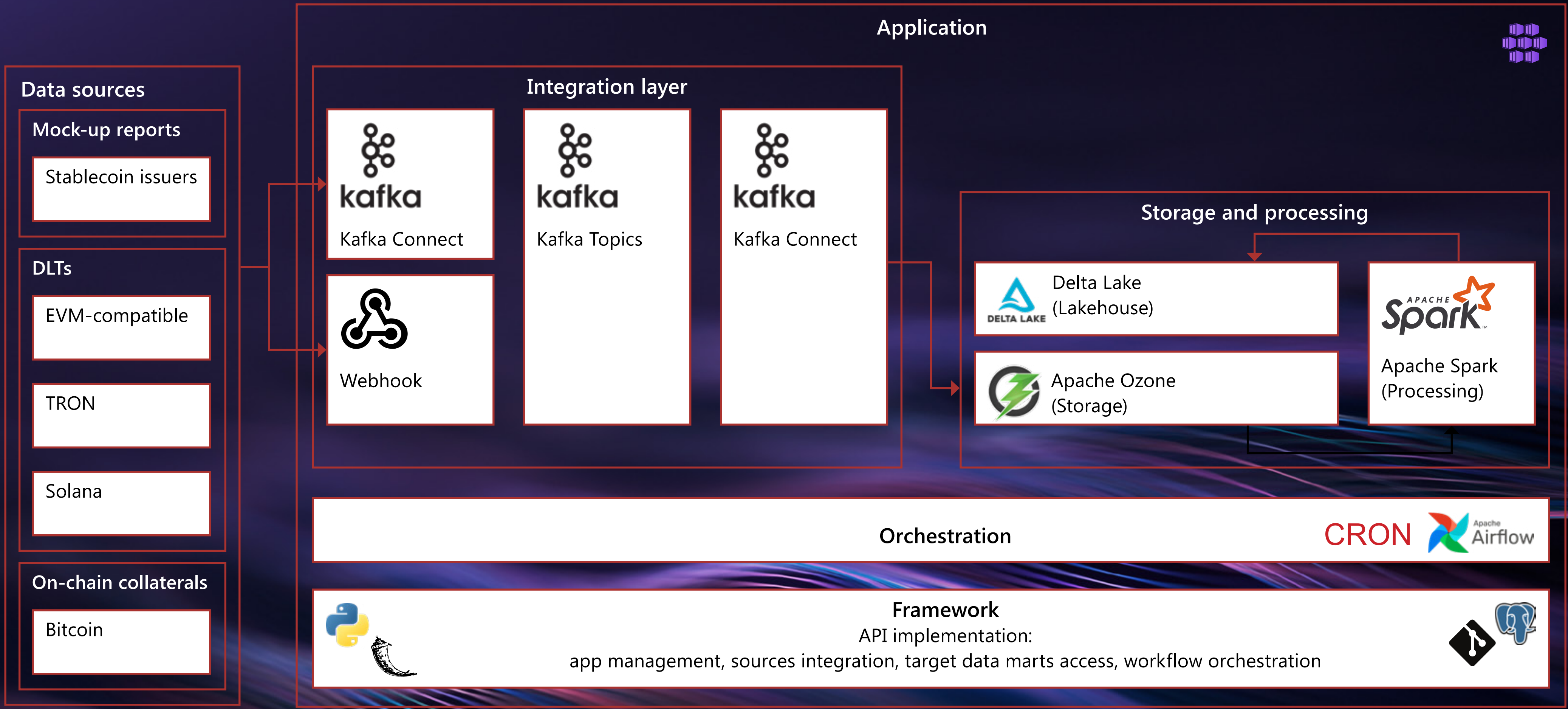
An added complication is that these components are large and would constitute a stack that would require significant maintenance and expertise. That said, Pyxtrial could be adapted to employ a big data approach.

An important reason for the PoC approach that was chosen stemmed from the data volumes: after analysing the expected data volumes and data retrieval rates, it became clear that the estimated total data increment would not exceed 120 MB annually, which removed the need for a big data stack (Appendix A outlines the data volume estimates).

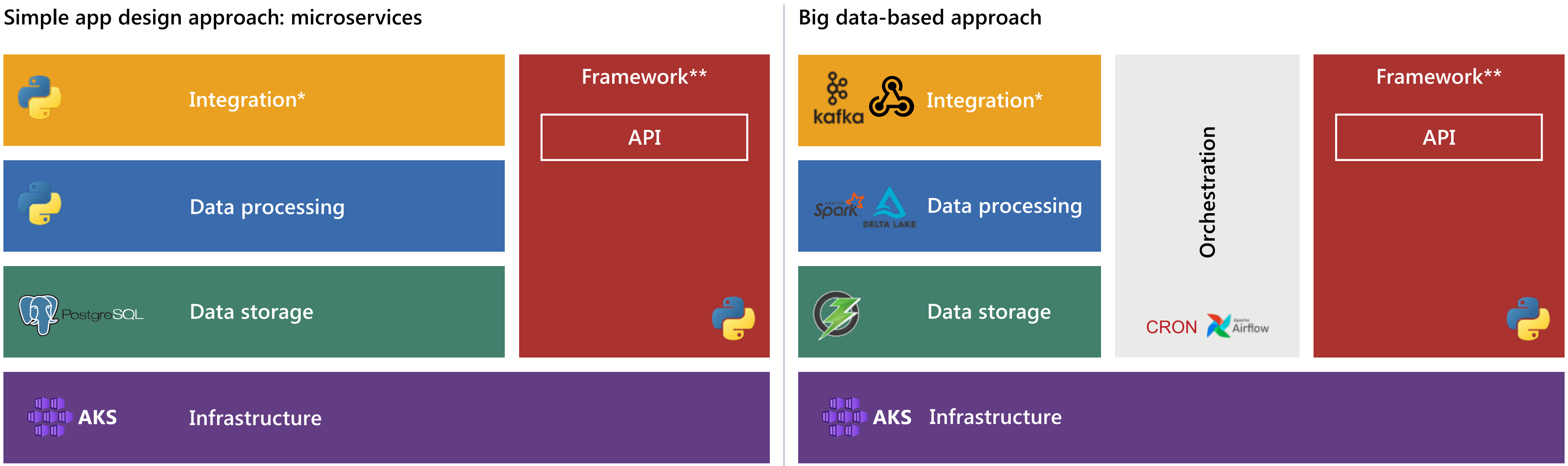
Graph D.1: Pyxtrial's functional architecture



Graph D.2: Alternative big data outline



Graph D.3: Simplified approach (left-hand panel) vs big data approach (right-hand panel)



With that determined, the functional layers of the two approaches were compared (Graph D.3). Factoring in the data requirements and considering that Pyxtrial is a PoC tool, not a final product, the simplified approach that uses a Python app integrated with a PostgreSQL relational database management system (RDBMS) and runs on Azure Kubernetes Service (AKS) was chosen.

Notes:

* Integration is the collection of assets and liabilities data.

** Framework is either a single Python app or a set of services written in Python.

Graph D.4 maps the functional architecture of this simplified solution, along with Pyxtrial's key use cases:

- Pyxtrial collects the assets and liabilities data via an API endpoint and integration layer (parts 1–4).
- It stores those raw data (part 5).
- It then processes and loads those data (parts 6–7) into the core data layer (the target tables in part 8).
- Finally, a dedicated API endpoint (part 9) provides the current view of those data to the regulators or supervisors who use it.

This streamlined solution eliminated the need for complex applications, instead using a backend app-based design. This means that Pyxtrial:

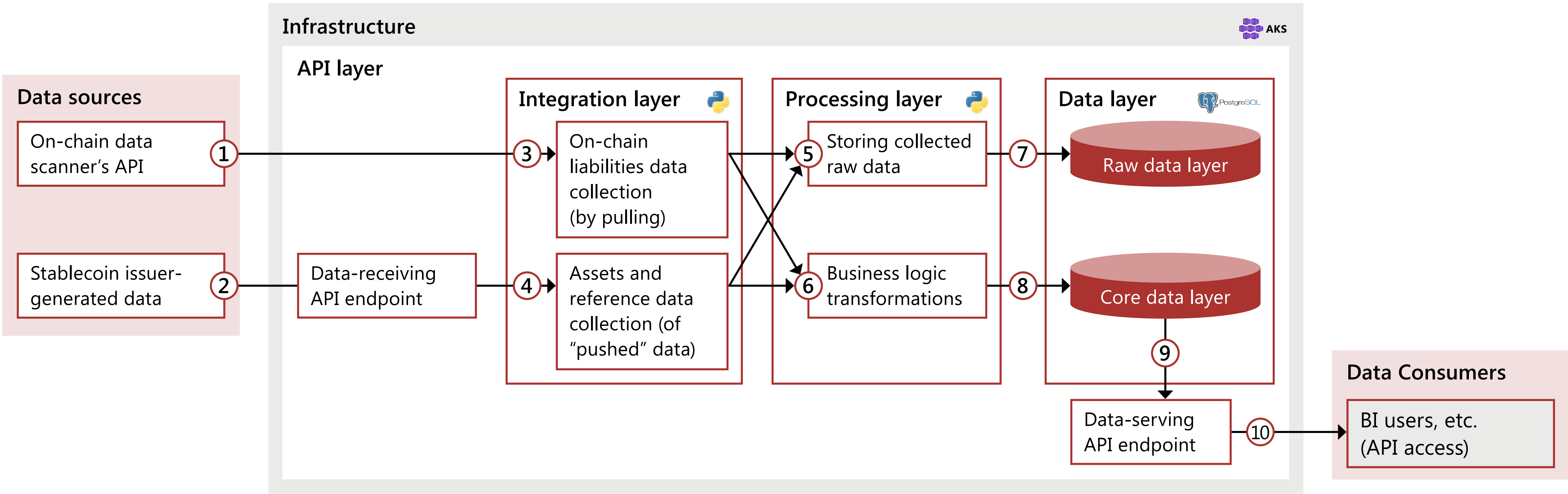
- has an application-centric approach that uses Python as the primary tool for data processing and integration, providing a leaner, more focused solution that could be tailored to specific prototype requirements;
- uses the open source PostgreSQL database for data storage, with components driven by AKS;
- does not require a data lake (though the solution is sufficiently flexible that this can be added later); and
- is sufficiently flexible to allow for greater customisation when managing the application's resources.

This approach makes Pyxtrial easier to deploy and removes complexity when it comes to investigating incidents and debugging, while also bringing independent service scalability and fault tolerance. It also offers development flexibility, with each service able to be modified or maintained independently. In other words, it meets the requirements that the Pyxtrial PoC be extensible, flexible, customisable and scalable.

This also brought the ability to adapt code rapidly to evolving requirements and changes in data sources, capabilities that were especially valuable during the PoC phase when iterations and adjustments were frequent. Lastly, it removed the need for powerful tools like Kafka and Spark, which might introduce unnecessary development costs and complexities.

With the architecture defined, the next step was to map these functional layers to the technical layers and to the application structure, with the technical layers using separate packages and following a modular design to make the application and its components loosely coupled, flexible and customisable.

Graph D.4: Pyxtrial's proposed data workflow mapped



Box D.A: Pyxtrial's data dictionary and the need to preserve history

While some stablecoin issuers provide aggregated reports, others are more detailed and break down their assets into sub-asset classes. This required the team to apply some sort of standardisation.

After considering various regulatory aspects and analysing the business of stablecoin issuance, the team identified the substantive fields that are retrievable in practice, as well as those that are not reported or that require additional calculations or data enrichment.

The above informed the build of a functional data dictionary – the foundation for the design of the data model – that specifies the collected attributes and maps them to different sub-asset classes.

It was at this point that relational data modelling was chosen. This was a fundamental decision as the database and its functionality are at the heart of Pyxtrial and have a significant impact on the persistence layer, the processing layer and how data should be retrieved for further analysis. This approach also incorporated flexibility for future unspecified needs.

Another consideration was how to preserve the history of changes within the primary tables in the database. Pyxtrial's database, for instance, needs to store a new version of USDT's report when the stablecoin issuer releases it and Pyxtrial must also preserve the old data.

Under the master data versioning (MDV) approach, every entity within the data model that needs to maintain a history is represented by two tables: a head table (which preserves static attributes along with a master key) and a version table (which retains all historical changes and all volatile attributes). The approach chosen for the version tables was Slowly Changing Dimension (SCD) Type 2, which implements the history of changes. This makes management of changes more flexible, while the extension tables provide extra content for versions of the asset entity with respect to specific sub-asset classes.

The benefit of the MDV approach is that it overcomes the possibility of the ripple effect, which is a complication that arises from an alternative approach that uses a single table for every entity, with that table updated each time. That alternative approach often requires updates across multiple tables – the ripple effect – which brings its own challenges. The MDV approach avoids this.

A-IV.2. Technical details

Pyxtrial's design meant it needed separate functions to ingest, process and store the data collected. To that end, the team selected the components in the following table.

Integration	Python app/service
Data processing	Python app/service
Data storage	PostgreSQL

These components work together to gather the liabilities-side data. They then transform and adapt those data, store them in a database and connect to the consumption layer, providing data in a format that is relevant to the end users who want to access them.

Specifically, Pyxtrial achieves its functionality through a multi-layered architecture that includes:

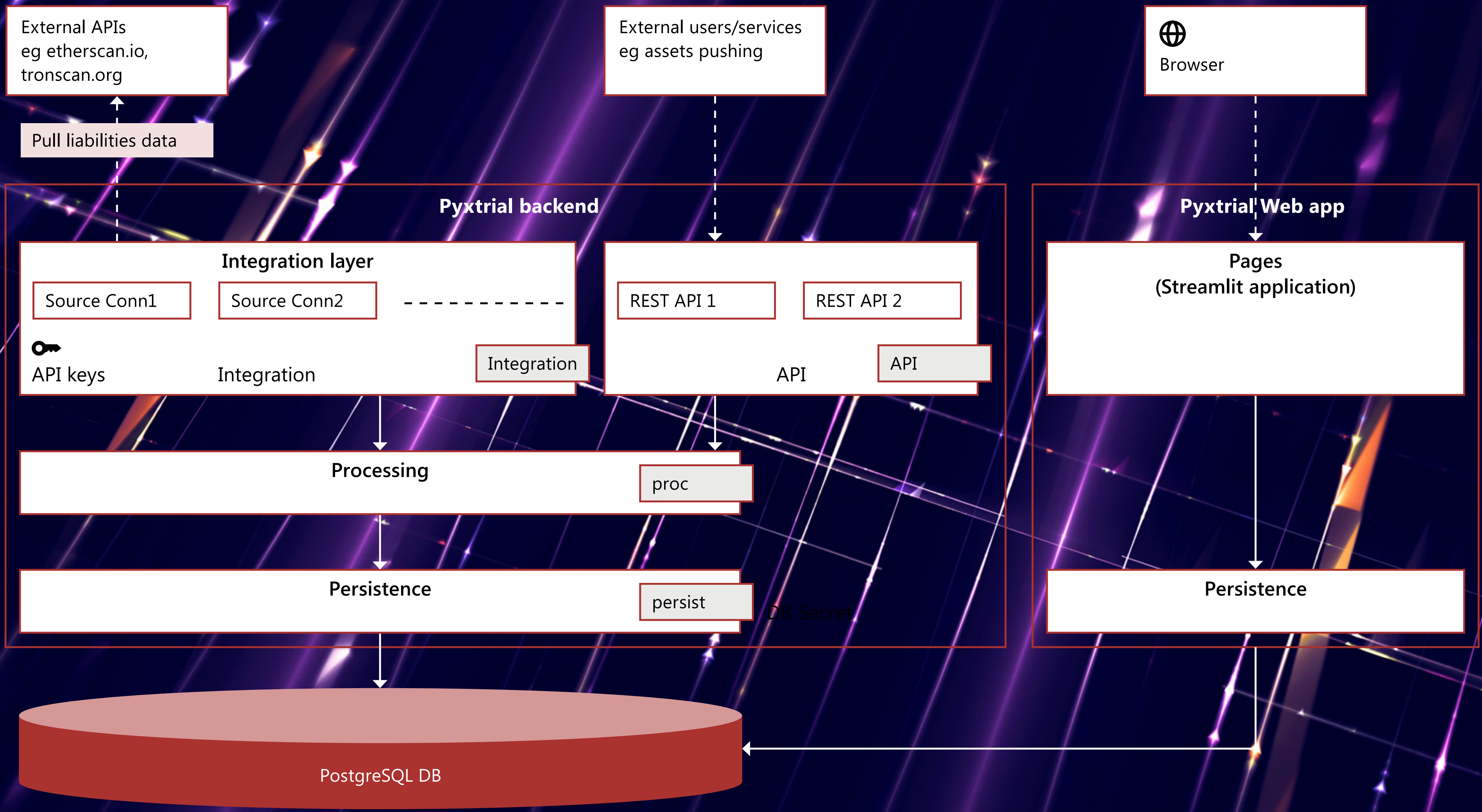
- the **API** layer, which receives generated issuer report data (off-chain);
- the **integration layer**, which connects to blockchain scanners' APIs and gathers stablecoin issuers' on-chain liabilities, for example via etherscan.io or tronscan.org;
- the **processing layer**, which transforms and enriches the data and manages database transactions; and
- the **persistence layer**, which enables low-level functionality for interaction with the database for application components. The persistence layer is based on the SQLAlchemy framework, whose benefits include a convenient way of defining the objects in the database and a migration tool that supports the evolution of the database.

This layered approach ensures the separation of concerns, in addition to enabling potential scalability and flexibility, and allows modular development and easier maintenance of the overall system. Graph D.5 shows the high-level technical design.

Python, which underpins the integration and data processing layers, was used as the main implementation language because it offers a rich ecosystem of frameworks and libraries and because it brings simplicity and ease of use, which are crucial in terms of rapid development and testing of the prototype. Python's flexibility also meant the developers could adapt code quickly to meet evolving requirements and changes in data sources.

PostgreSQL serves as the RDBMS, providing a structured way to store received data. Using PostgreSQL as the data storage layer derived from the decision to take a simplified approach for the PoC while leaving open the possibility for a data lake at a later stage.

Graph D.5: High-level technical design for Project Pyxtrial



A-IV.3. Data flows

A-IV.3.1. Overview

There are three notable aspects about Pyxtrial's data flows (Figure A-IV-F):

- **on-chain data collection**, which covers stablecoin liabilities;
- **off-chain data collection**, which covers stablecoin assets and could be expanded to additional liabilities data; and
- **retrieval of the current view** of collected data.

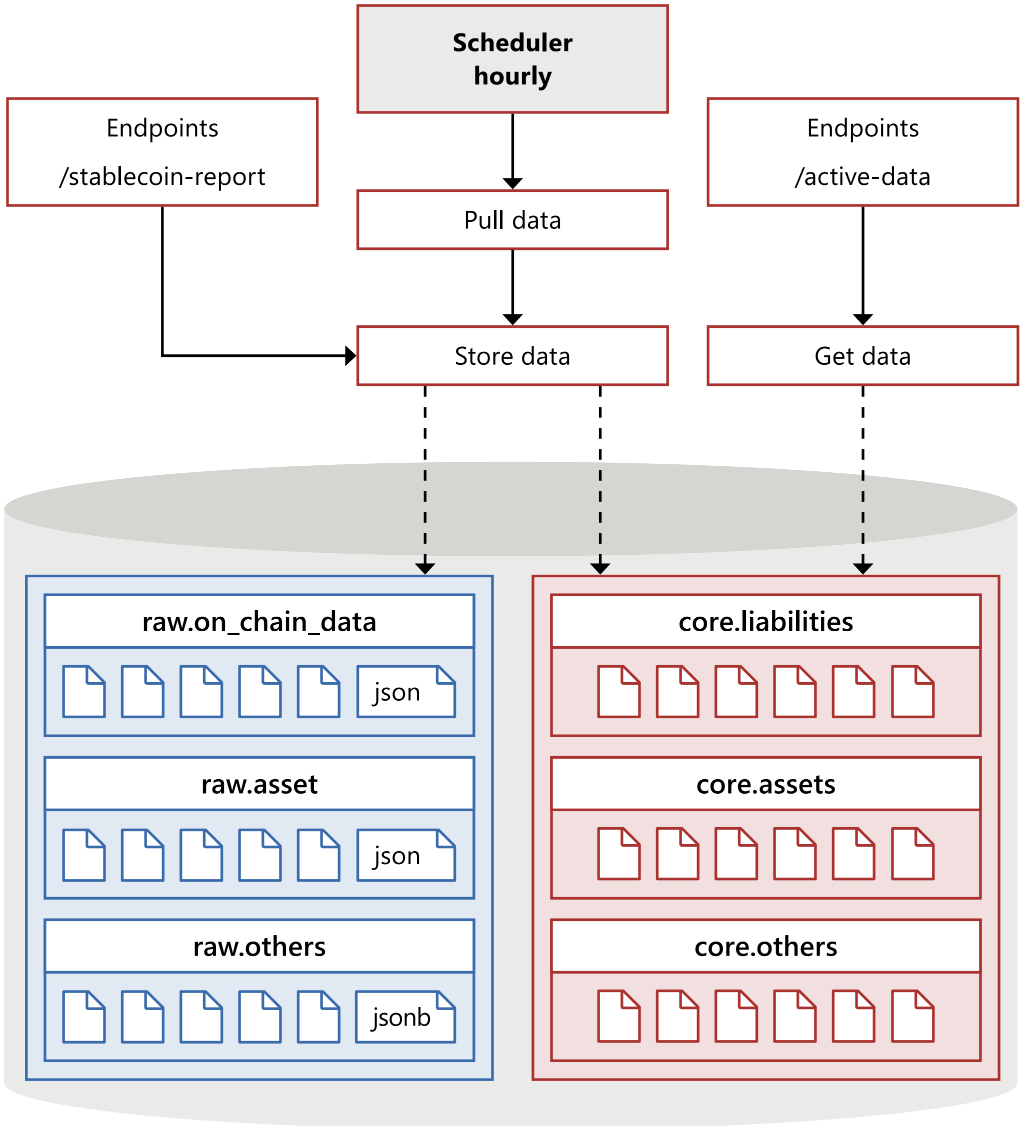
On-chain data collection happens on an hourly basis, with the integration layer fetching data on the circulating supply volumes of the five stablecoins from the five public blockchains using the blockchain scanners' APIs.

Off-chain data collection is enabled by the Pyxtrial REST API endpoint (flagged on Graph D.6 as **/stablecoin-report**). This endpoint receives the transparency reports from the stablecoin issuers in a structured format that is defined by Pyxtrial.

Both these data flows see the raw data payloads stored in raw schema tables. After this, they are stored in target tables of core schema.

Lastly, Pyxtrial provides an API endpoint (flagged on Graph D.6 as **/active-data**) to serve the current view of the collected data. It fetches the most recently updated or collected data from every entity of the core layer and prepares the target view model that the end user can access.

Graph D.6: Diagram of Pyxtrial's data flows



Box D.B: Security considerations

Security is a crucial aspect of Project Pyxtrial, which relies fully on BIS Innovation Hub's cloud environment and security controls configurations.

Pyxtrial does not implement any extra custom security features, since a security pillar was not part of the initial scope. However, it does use the Azure Key Vault cloud service to fetch and keep confidential information like API access keys, database access and credentials for DevOps tools.

Given this, and as Pyxtrial might be deployed or adapted for different environments, implementers might wish to incorporate the following non-exhaustive list of additional security features:

- integration with other secrets management solutions (either cloud-native or hosted on-premises);
- authentication and authorisation of calls to Pyxtrial API endpoints;
- signature verification of stablecoin issuer report data payloads (which would require establishing a process of signature creation and management for each data sender); and
- whitelisting of Pyxtrial API endpoint callers.

A-IV.4. APIs and integrating data flows

The Pyxtrial PoC works with two primary source types: an API that provides on-chain data from public blockchain scanners, and simulated reports from stablecoin issuers.

For the first type, Pyxtrial pulls token liability data from the scanners' APIs. Table D.1 shows the scanners that fall within the project's scope, one for each of the five blockchains.

Table D.1: How Pyxtrial sources the liabilities-side data

Scanner name	API base path	API type
Etherscan	api.etherscan.io/api	REST
Binance Smart Chain Explorer	api.bscscan.com/api	REST
Avalanche Snowtrace	api.routescan.io/v2/network/mainnet/evm/43114/etherscan/api	REST
TRONSCAN	apilist.tronscanapi.com/api	REST
Solana Mainnet Beta cluster	api.mainnet-beta.solana.com	JSON-RPC

Table D.2 shows Pyxtrial's two API endpoints.

Table D.2: Pyxtrial's API endpoints for assets and consumer access

Pyxtrial API endpoint	HTTP method	Description
/stablecoin-report	POST	Receives stablecoin issuer transparency report data generated by the special generator (refer to Appendix C)
/active-data	GET	Serves current view of most recently updated and collected data to consumers

Pyxtrial's design sees one API endpoint receive the data about each stablecoin, with fields for assets and liabilities. This API is programmed to connect to the blockchains and to the systems of stablecoin issuers, though for the purposes of the PoC, it sources these assets-side data from the simulated stablecoin issuer reports. The second API endpoint serves the collated data to the end user.

A-IV.4.1. Deep dive: Assets data and the API

As noted, Pyxtrial's generator is based on genuine data from stablecoin issuers' periodic reports (which are typically monthly or quarterly) and simulates hourly movements in the underlying assets to match the frequency with which it collects liabilities-side data.

The generator is a mock service that serves generated issuer data. This is not part of the primary application, but is needed as a placeholder.

Currently, each stablecoin issuer determines its own report structure and data granularity (though a working assumption is that regulators would require that these data be standardised). Consequently, some stablecoin issuers report the values of their assets and liabilities at aggregated and sub-aggregated levels, while others provide breakdowns with respect to specific sub-asset classes or even specific financial instruments (USDC, for example, lists US Treasury assets).

Pyxtrial's generator therefore provides plausible transparency report data based on actual reports from the stablecoin issuers in the PoC scope. The generator is implemented as a separate application that feeds stablecoin report data to the API endpoint on demand (this application also serves end-to-end testing, benchmarking and development purposes, but is not intended to be a part of the production environment).

Table D.3 shows the private paths to the API endpoint for the development and UAT environments (detailed API documentation can be found in the **/docs** endpoint).

Table D.3: Pyxtrial's private paths to the API endpoint for development and UAT

DEV	pyxcoinapi-dev.bisih.local/assets/{stablecoin}
DEV	pyxcoinapi-dev.bisih.local/docs
UAT	pyxcoinapi-uat.bisih.local/assets/{stablecoin}
UAT	pyxcoinapi-uat.bisih.local/docs

Pyxtrial's generator serves report data in a special format defined as part of the PoC. The format contains aggregated total assets amounts as well as breakdowns with respect to multiple sub-asset classes, for example cash, debt, loans and MMFs (see Section 5.2).

Next, the generator models the total assets value amount and the sub-asset classes allocation weights based on the collected transparency report data for a historical period of one year for each stablecoin that is in scope. This plausibly distributes the total assets value between specific asset classes. Additionally, the generator supports the generation of token liabilities data, served in the same format in a separate tokens section.

Finally, Pyxtrial is able to compare the liabilities-related data from on-chain sources against the liabilities data reported by stablecoin issuers. However, successfully comparing all on-chain versus off-chain liabilities is a step beyond the PoC, as the integration layer would need to gather information from every on-chain source for each stablecoin; currently it does so from just five.

Endnotes

1 In this report, we use the terms "regulators" and "supervisors" interchangeably to refer to the end user responsible for monitoring systemically important stablecoins. This overlap in usage is because some jurisdictions combine these functions while others keep them separate. Consequently, the end user of Project Pyxtrial would depend on how jurisdictions approach the supervision of such stablecoins.

2 See defillama.com/stablecoins.

3 Because Pyxtrial is a PoC and does not delve into policy recommendations, the project did not consider types of asset-liability mismatch and does not define the term. When analysing data, users would set their own parameters that would meet their own needs.

4 The five stablecoins are: USD Tether (USDT), USD Circle (USDC), Binance USD (BUSD), Pax Dollar (USDP) and TrueUSD (TUSD). Their liabilities data were tracked on five blockchains: Ethereum, Binance Smart Chain (BSC), Avalanche, TRON and Solana.

5 See defillama.com/stablecoins.

6 See H Miller, "PayPal enables PYUSD stablecoin for Xoom cross-border money transfers", Bloomberg, 4 April 2024, www.bloomberg.com/news/articles/2024-04-04/paypal-s-pyusd-stablecoin-now-available-for-xoom-cross-border-money-transfers.

7 See Ripple, "Ripple to issue USD-backed stablecoin bringing more utility and liquidity to XRP Ledger", 4 April 2024, ripple.com/ripple-press/ripple-to-issue-usd-backed-stablecoin-bringing-more-utility-and-liquidity-to-xrp-ledger/.

8 See Visa, "Visa expands stablecoin settlement capabilities to merchant acquirers," 5 September 2023, usa.visa.com/about-visa/newsroom/press-releases.releaseId.19881.html.

9 See, for example, Crisanto et al (2024).

10 See, for example, BIS (2024).

11 See Kosse et al (2023) for information on the use of stablecoins.

12 Ibid.

13 Note that fiat-backed, commodity-backed and crypto-backed stablecoins are sometimes also defined as collateralised stablecoins, with the first two referred to as "off-chain collateralised" stablecoins and the latter "on-chain collateralised" stablecoins.

14 In this report, "stablecoin" refers to stablecoins with this type of backing. In theory, algorithmic stablecoins might also be backed by traditional assets, but for simplicity they are excluded from this definition.

15 See Uhlig (2022), Cornelli et al (2023)

16 See Committee on Payment and Settlement Systems and International Organization of Securities Commissions, *Principles for Financial Market Infrastructures*, April 2012, www.bis.org/cpmi/info_pfmi.htm.

17 See BIS CPMI and IOSCO (2022).

18 See BCBS (2023).

19 See BCBS (2022).

20 See BCBS (2024).

21 See FSB (2023).

22 See BoE (2023).

23 See ESMA (2023).

24 See NYDFS (2022).

25 See US Department of the Treasury, "President's Working Group on Financial Markets releases report and recommendations on stablecoins", 1 November 2021.

26 See D Garcia Ocampo, N Branzoli and L Cusmano, "Crypto, tokens and DeFi: navigating the regulatory landscape", *FSI Insights on policy implementation*, no 49, May 2023.

27 See Crisanto et al (2024).

28 See BoE (2023).

29 This contrasts with, say, bank deposits or tokenised deposits, which tend to be fractionally backed (by loans, investments and other assets, as well as liquid assets); asset-backed stablecoins, on the other hand, depend for their value on the 1:1 backing with liquid assets, which is why maintaining or exceeding this 1:1 backing at all times is critical.

30 See R Browne, "World's biggest stablecoin regains dollar peg after $3 billion in withdrawals", *CNBC*, 13 May 2022, www.cnbc.com/2022/05/13/tether-usdt-stablecoin-regains-peg-after-3-billion-in-withdrawals.html.

31 See C Ventricelli, "USDC's depeg laid bare the risks traditional finance poses to stablecoins", *CoinDesk*, 2 April 2023, www.coindesk.com/consensus-magazine/2023/04/02/usdcs-depeg-laid-bare-the-risks-traditional-finance-poses-to-stablecoins/.

32 See BoE (2021).

33 See FSB (2023).

34 See SEC (2023).

35 See European Union (2018).

36 See FSB (2023).

37 See NYDFS (2022).

38 See BCBS (2022).

39 Specifically, this is the BCBS Prudential Treatment Classification Condition 1.5 Redemption Risk Test and Regulator Supervision, sub-criteria 1: Evaluation of the value and composition of reserve assets.

40 See dtif.org/.

41 See Circle, "Circle is first global stablecoin issuer to comply with MiCA, EU's landmark crypto law", 1 July 2024, www.circle.com/en/pressroom/circle-is-first-global-stablecoin-issuer-to-comply-with-mica-eus-landmark-crypto-law.

42 See BIS Project Atlas www.bis.org/publ/othp76.htm

43 See T Mitchelhill, "Paxos to issue USD stablecoin in Singapore, wins initial approval", *Cointelegraph*, 16 November 2023, cointelegraph.com/news/paxos-singapore-issue-usd-stablecoin-after-license-approval.

44 See B Lindrea, "BlackRock's BUIDL becomes the world's largest tokenized treasury fund", *Cointelegraph*, 1 May 2024, cointelegraph.com/news/blackrocks-buidl-becomes-worlds-largest-tokenized-treasury-fund.

45 See dtif.org/.

BIS Innovation Hub

Bank of England

Project Pyxtrial
Monitoring the backing
of stablecoins

bis.org