The Blockchain Built for Finance

Liquidity Availability

A New Framework
for Efficient Liquidity Management

Injective Research

Introduction

Innovation across blockchain and on-chain finance has progressed considerably throughout the last decade. The cornerstone of functionality and growth for technology within the industry has always been liquidity, persisting to this day. Traditional approaches to liquidity management, often reliant on isolated pools for individual decentralized applications (dApps), present significant inefficiencies and barriers to innovation. This paper introduces a novel concept: Liquidity Availability.

The objective of this paper is to contextualize and explore the concept of Liquidity Availability, establish its significance in the broader crypto ecosystem, and discuss the technical mechanisms required to effectively apply it. Additionally, the paper will examine its potential to address current liquidity challenges in Web3.

What is Liquidity Availability?

Liquidity Availability refers to the ability to meet the liquidity requirements needed to ensure the successful execution of a transaction at any given moment given a particular set of bounded constraints.

At the application level, Liquidity Availability is primarily driven by **Application-Specific Liquidity**, which is liquidity confined within a particular dApp and cannot be repurposed for other uses. This constraint occurs in two key ways:

1. **Network-Level Isolation:** The liquidity within a dApp cannot be leveraged by other applications within the same network to power processes aside from its specific use case (e.g., asset pair swaps), leading to fragmentation across the ecosystem.
2. **Intra-Application Constraints:** Even within a single dApp, liquidity is often locked into specific pools for singular functions, such as a liquidity pool for ETH/BTC swaps. This means that the application cannot repurpose or leverage this liquidity for other use cases within its own system, which limits its flexibility and decreases capital efficiency.

Applications must attract asset deposits from liquidity providers (LPs) into designated repositories, typically smart contracts, each of which facilitate only specific transactions. Common examples include liquidity pools for asset swaps, lending protocols for overcollateralized loans, insurance pools for protocol backstops, staking pools for liquid staking derivatives (LSDs), etc. This in and of itself is a form of liquidity fragmentation, which is compounded in multichain applications, where liquidity is also spread across multiple networks.

Therefore, Total Value Locked (TVL), often used as a measure of a dApp's liquidity, does not accurately reflect Liquidity Availability. TVL aggregates the total liquidity deposited within an application, which may be significantly fragmented across various use cases and therefore not fully accessible for broader transaction needs. However, within the current system, it can serve as a rough estimate for application-application comparisons in certain situations.

Currently, at the network level, Liquidity Availability is still constrained within individual applications. This means that a network cannot leverage the total liquidity housed across its ecosystem of dApps to improve overall Liquidity Availability. As such, application-specific liquidity remains the primary driver.

Due to this, interoperability solutions have been the industry's primary focus for enhancing Liquidity Availability. Various techniques have been explored, such as cross-chain liquidity aggregators, unified liquidity pools, atomic swaps, liquidity routing pools, and tokenized liquidity bridges. While these efforts have led to some marginal improvements in Liquidity Availability, liquidity still remains increasingly siloed within applications.

Why Does it Matter?

Liquidity isolation and fragmentation across networks brings inherent disadvantages, such as stunted growth and capital inefficiency. For example, Ethereum, with a TVL of ~$163 billion, sees Uniswap V2 holding ~$1.8 billion of that TVL, which itself is spread across over 400,000 pools. The inability for an Ethereum application to leverage Uniswap V2's capital, as well as Uniswap's own inability to freely leverage its own liquidity, overtly demonstrates the importance of Liquidity Availability. Crypto's failure to effectively solve this has created an unsustainable environment, where accumulating capital is highly competitive and net growth requires a continual inflow of new capital on-chain.

The needs prioritized within such an environment run directly counter to the needs of pushing forward bleeding-edge technology, where the ability to launch fast and iterate are innovation's greatest catalysts. This results in a player versus player (PvP) ecosystem where score is kept in capital and resources are forced to be allocated accordingly. As the market becomes more competitive and crowded, fighting for capital becomes an increasingly large cost center for projects. The risk of testing new products or approaches also often grows exponentially with time, as the need to retain application-specific liquidity is paramount. Together, these harmful symptoms of the current system have a decelerating effect on progress as a whole.

In terms of launching quickly, the dependency on application-specific liquidity is realized as a significant barrier to entry. This industry-wide issue has already shown ripple effects, as applications reaching a critical mass of liquidity have grown more quickly than new entrants. This trend leads to a less competitive and more consolidated marketplace, introducing the detrimental effects of monopolization commonly seen within inefficient traditional markets. As long as application-specific liquidity remains inherent within the current system, this trend will persist and likely worsen over time.

Taking stock of these headwinds, a common thread emerges. Poor Liquidity Availability has led to an embedding of application-specific liquidity within the DNA of decentralized systems. As a result, the negative effects of this paradigm have permeated throughout the industry. Addressing a systemic issue necessitates a focus on its root cause, which in this case, places Liquidity Availability as the central concept requiring attention.

Liquidity Availability in TradFi

Liquidity Availability is a critical concept not only in blockchain and DeFi, but also in traditional finance (TradFi). The degree of Liquidity Availability in TradFi stands in stark contrast to that observed within the crypto ecosystem. Understanding these differences is crucial for grasping the importance of Liquidity Availability and recognizing potential enhancements that blockchain-based solutions may offer to the broader economic system.

The traditional financial system includes banks and nonbank lenders, insurers, securities markets, and investment funds. It also includes clearing counterparties, payment providers, central banks, as well as financial regulators and supervisors. These institutions provide a framework to conduct economic transactions, monetary policy, and channel savings into investment, thus supporting economic growth. Each of these components play a distinct role in facilitating Liquidity Availability across the system, collectively ensuring the provision of liquidity required for the successful execution of a transaction at any given moment.

In TradFi, Liquidity Availability is enhanced through well-established mechanisms. Key measures of Liquidity Availability include the volume of liquid assets, speed and efficiency of transaction settlements, and accessibility of credit and financial services. The following systems play pivotal roles in maintaining and enhancing Liquidity Availability:

Credit Systems in TradFi are foundational to liquidity. They provide the ability to borrow funds or assets with the agreement to repay later, thus ensuring that liquidity needs are met promptly. Credit cards, for instance, offer revolving lines of credit that facilitate continuous liquidity for both consumers and businesses. Moreover, the credit system enables borrowers to shift future consumption into the present while allowing lenders to defer current consumption to the future, a dynamic not yet fully realized in DeFi. This enhances the volume of liquid assets available in the economy and increases the accessibility of credit and financial services, ensuring that liquidity is readily available to meet the demands of various transactions.

Insurance Mechanisms further enhance Liquidity Availability by pooling risks and providing financial protection against losses. This allows individuals and businesses to manage economic fluctuations more effectively, maintaining liquidity even in adverse conditions. Insurance spreads risk, stabilizing the volume of liquid assets across different sectors of the economy and time periods, ensuring that funds are readily available when needed, thereby improving transaction speed and efficiency.

Refinancing practices, which involve revising and replacing existing credit agreements, allow borrowers to take advantage of favorable financial conditions, thus improving liquidity management. This infrastructure enables continuous liquidity optimization, ensuring that borrowers can meet their liquidity requirements effectively. Refinancing provides Liquidity Availability to whoever is holding the debt by providing opportunities to lower costs or extend payment terms, thus increasing the accessibility of credit and enhancing the volume of liquid assets in the financial system.

Clearing Houses play a crucial role by acting as intermediaries between buyers and sellers of financial instruments. They settle trades, collect margins, and mitigate counterparty risk, ensuring that transactions are executed efficiently and securely. For example, the Depository Trust & Clearing Corporation (DTCC) in the United States handles the majority of securities transactions, providing a stable and reliable infrastructure for Liquidity Availability. Clearing houses diversify risk (rather than having liquidity sit within a particular DeFi protocol), significantly improving the speed and efficiency of transactions and ensuring that liquid assets are promptly available.

Interbank Lending markets, such as the federal funds market in the United States, facilitate the distribution of liquidity within the banking system. Benchmarks like the Secured Overnight Financing Rate (SOFR) influence borrowing and lending costs across the financial system, ensuring that liquidity is efficiently allocated where it is most needed. This infrastructure prevents liquidity from being siloed, increases the volume of liquid assets available across the banking system, and enhances the accessibility of credit.

Escrow Services enhance Liquidity Availability by holding assets or funds on behalf of transaction parties until contractual obligations are met. This ensures that transactions are completed smoothly, reducing the risk of liquidity shortfalls and enhancing overall market stability. Similar to application-specific liquidity, escrow establishes trust by proving that funds are available, thereby enhancing transaction speed. Escrow also supports contract enforceability, helping parties gain trust and ensuring that funds are securely held until all conditions are satisfied. However, while escrow has its place in the TradFi system, it is not suitable for every transaction type due to its inherent limitations.

Centralized Dealers and Market Makers, such as large investment banks, continuously quote buy and sell prices for financial instruments, ensuring that there is always a counterparty for transactions. This practice reduces capital inefficiency and removes liquidity silos, providing greater utility to existing liquidity. By maintaining inventories of various assets, these market participants enhance the volume of liquid assets and ensure that transactions can occur swiftly and efficiently.

By examining these mechanisms, it becomes clear that infrastructure in TradFi is designed to meet liquidity requirements at any given moment, ensuring a robust and efficient financial system. While the methods to achieve comparable Liquidity Availability in a decentralized environment may differ, the reasons for doing so strongly parallel traditional systems. By drawing upon concepts from established systems, the crypto ecosystem can more effectively address the challenges born out of application-specific liquidity. This can ultimately foster a more efficient and robust financial framework, enhancing the overall viability and growth potential of blockchain and DeFi technologies

Applying the Liquidity Availability Concept

It may be possible to address current limitations by using the concept of Liquidity Availability to think outside of the box. Rather than viewing it as an application problem, where application-specific liquidity is the limiting factor, Liquidity Availability should be thought of as a network problem. From this perspective, ensuring the successful execution of a transaction is abstracted from where it originates, and as such, so is the location of the liquidity.

In TradFi, numerous established systems ensure that liquidity is always available where it is needed, preventing stunted growth and capital inefficiency. Drawing inspiration from these mechanisms, a unique approach can be applied within the crypto ecosystem.

In the practical sense, a network would need to be able to mobilize the liquidity hosted across its network, making it readily accessible for the agnostic servicing of transactions. As a result, applications within the network would essentially be decoupled from their liquidity bases, potentially eliminating the dependency on application-specific liquidity. In this system, Liquidity Availability would be measured at the network-level and would be driven by the aggregate growth of its ecosystem. Importantly, this would change the limiting factor from being externally dependent (application-specific liquidity), to internally dependent (network performance capability), which could be controlled and incrementally improved.

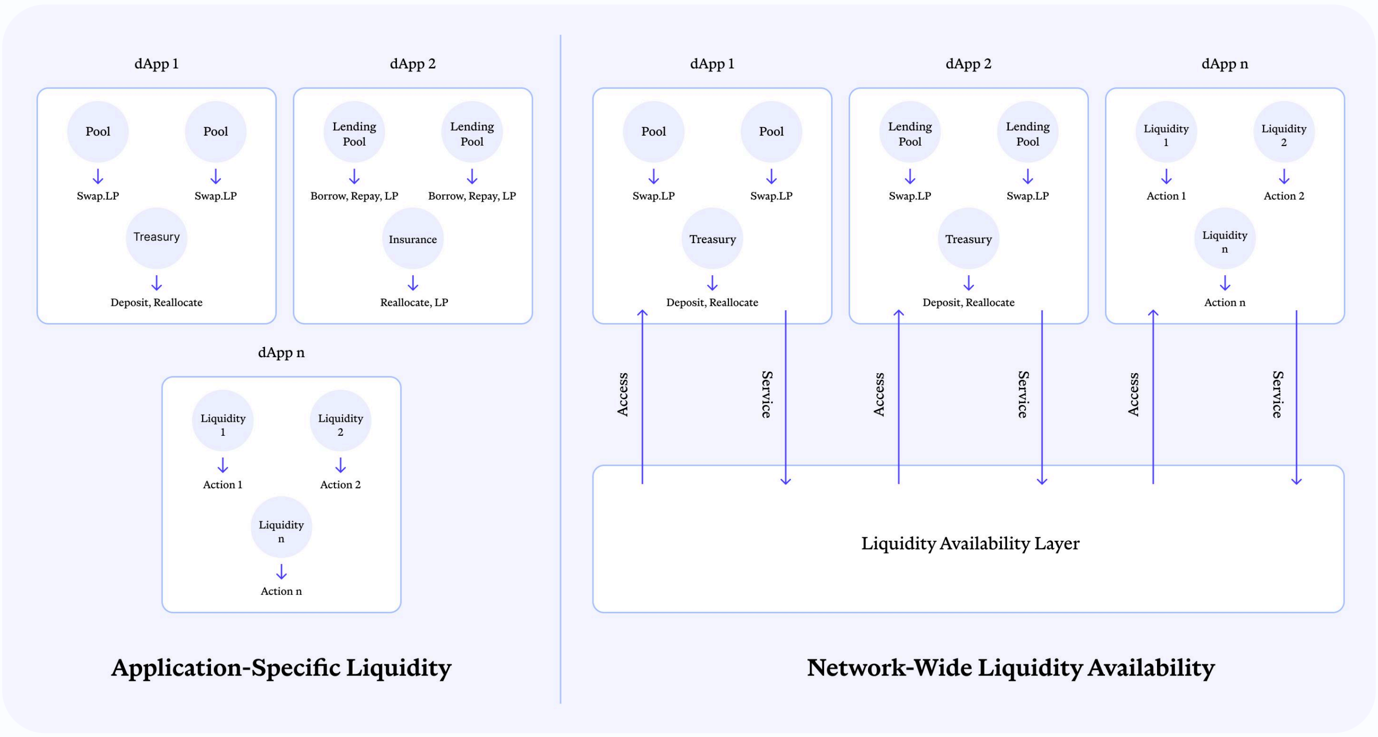

Figure 1: Application-Specific Liquidity vs. Network-Wide Liquidity Availability

On the left side, the diagram illustrates how liquidity is confined within individual dApps, creating isolated liquidity (application-specific liquidity). On the right side, the diagram demonstrates how the Liquidity Availability framework enables network-wide liquidity, allowing for a more efficient and dynamic liquidity system.

Moreover, addressing Liquidity Availability at the network level can significantly lower barriers to entry for new projects. Currently, the need for substantial application-specific liquidity poses a major challenge for startups, delaying their launch and limiting their ability to iterate quickly. By ensuring that liquidity is accessible network-wide, new applications can launch faster and innovate more freely, without the immediate pressure of securing large liquidity pools. This creates a more competitive and diverse ecosystem, where innovation thrives and smaller players have a fair chance to succeed.

In essence, applying the concept of Liquidity Availability at the network level transforms liquidity management from a competitive, isolated struggle, into a collaborative, integrated process. This paradigm shift could unlock new possibilities for innovation and growth in the crypto space, making it more adaptable and resilient to the evolving needs of users and applications alike. By leveraging the lessons from traditional finance, the crypto ecosystem can develop more sophisticated and efficient liquidity management solutions, ensuring sustained growth and technological advancement. This, in turn, supports a faster-paced environment for launching new projects, tokenized assets, and on-chain markets, further enhancing the vitality and dynamism of the decentralized solutions.

Critical Mechanisms for Implementation

On the surface, the idea of mobilizing liquidity between dApps across a network may sound somewhat straightforward. However, there are a number of issues this presents that necessitate innovation in order to implement an efficient end-to-end solution.

For starters, the issue of LPs is glaring. Providing liquidity to a particular application or pool is often permissionless and generally for a specific purpose. Participants do not envision a scenario where their deposited liquidity is taken and used for another purpose, let alone on an entirely different application. Furthermore, they benefit from the permissionless nature of these protocols. What if they want their liquidity back, but it’s being used elsewhere? Moreover, why would an app want to surrender its hard-earned liquidity to service what may be a competitor? Not to mention, how would the assets actually move between applications in a cost-effective manner?

Economic Incentives & Risk Mitigation

First and foremost, an economic incentive would have to be established such that the opportunity cost of not participating in boosting network Liquidity Availability would outweigh that of opting-in. Specifically, capital efficiency amongst participating dApps would be enhanced, as well as potential revenue generation. Applications that participate would benefit from increased liquidity, leading to improved user experiences, greater transaction volumes, and potentially higher protocol revenue via a flywheel effect.

However, these benefits must be carefully balanced with assurances that the interests of LPs are protected. This requires developing mechanisms to ensure that withdrawals can be serviced promptly to avoid events similar to a bank run. Addressing this aspect could involve developing a liquidity reserve strategy, or implementing time-based withdrawal schedules that allow for the staged reclamation of liquidity. Such systems would have to be carefully engineered to protect against financial manipulation and other vectors for exploitation in order for this to be secure and effective, though.

Once economic incentives were aligned, the behavioral challenges relative to applications and users would largely be ironed out. This would pave the way for the technical implementation.

Just-in-Time (JIT) Actions

Ostensibly, three core functionalities would need to be built for such a system to be possible: Just-in-Time (JIT) Actions, Liquidity Proving, and a Solver & Routing Layer.

Just-in-Time (JIT) Actions are actions that respond to specific changes in the chain state when certain trigger conditions are met. These actions are designed to optimize the allocation of various resources by dynamically reallocating them as needed.

- **Trigger Mechanisms:** JIT Actions would be initiated based on pre-defined triggers, such as sudden spikes in resource demand, changes in user activity, fluctuations in network conditions, or inputs from a coordination layer. These triggers would be chain messages that live within the module state and would be stateful.
- **Asynchronous Interfaces for Integration:** These interfaces enable conditional engagement with the JIT Actions system, allowing dApps to dynamically contribute or retrieve liquidity based on preset conditions, thereby optimizing network-wide liquidity distribution.
  - **Async Deployment Interface:** Allows dApps to contribute liquidity to the Liquidity Availability system when certain conditions are met (e.g., idle liquidity within the application). This lets dApps make liquidity available without interrupting their core operations, supporting a network-wide availability model.
  - **Async Retrieval Interface:** Enables dApps to request liquidity on demand in response to predefined conditions like spikes in user demand. By interacting asynchronously, dApps gain access to network-wide liquidity without being restricted to their own reserves, ensuring operational flexibility.

- **Instruction:** When the trigger conditions for the particular action are met, a sequence of instructions would be triggered. These instructions could be chained in a way that the fundamental building blocks of these instructions are JIT Actions by themselves.
- **Smart Contract Automation:** The execution of JIT Actions would be governed by smart contracts that would automatically adjust resource allocations in response to trigger conditions. The instruction triggers would be monitored by the smart contracts, which would first simulate whether the instruction remains valid given a particular chain of JIT Actions.
- **Multi-Resource Allocation:** While liquidity would be a primary focus, JIT Actions could extend to other critical resources within a blockchain network. For example, JIT Actions could allocate computing power or storage resources between two parties.

Liquidity Proving

In addition to JIT Actions, **Liquidity Proving** would be essential for ensuring that liquidity is available and can be mobilized as required across the network.

- **Liquidity Proving Mechanisms:** Liquidity Proving could involve a mechanism where dApps provide verifiable proofs that they possess the liquidity required to meet obligations and JIT Action servicing.
- **Incentivized Participation:** While Liquidity Proving would need to be a requirement for participation in the overall system, the computational overhead to do so could end up being a deterrent. Therefore, dApps may need to be incentivized to participate in the process. This could be done through token incentives, subsidies, or reduced fees. However, it may be sufficient to present data on the net benefits of opting into the overall system. This could include verifiable measurements of liquidity depth gain, capability enhancement, revenue growth, and/or capital efficiency comparisons.
- **Fail-Safes and Backstops:** To prevent systemic risks, fail-safes and backstop mechanisms would need to be in place. For example, in the event of a sudden liquidity shortfall, the network could deploy emergency reserves or temporarily restrict certain activities until liquidity is restored.

Solver & Routing Layer

While the concept of JIT Actions addresses real-time liquidity provisioning and Liquidity Proving ensures sufficient liquidity for execution, a critical intermediary is needed to efficiently route and allocate liquidity across applications or chains. This intermediary, a **Solver & Routing Layer**, would act as the decision-making engine within the Liquidity Availability framework, responsible for optimizing and dynamically routing liquidity based on real-time network conditions.

- **Optimization & Decision Engine:** The solver would continuously evaluate liquidity pathways, optimizing for factors such as transaction cost, speed of liquidity movement, capital efficiency, and network conditions (e.g., congestion, gas fees). This would guide its decisions on the most efficient routes for shipping liquidity across applications or chains, ensuring liquidity is allocated in real-time to meet specific user needs and maximize efficiency.
- **Constraint Handler:** Beyond optimization, the solver would account for constraints such as Liquidity Availability, gas costs, withdrawal times, and risk factors. By adhering to pre-defined parameters (e.g., maintaining reserves or avoiding over-leveraged pools), the solver would ensure liquidity is allocated safely and efficiently.
- **Dynamic Behavior:** The solution would be designed to dynamically adjust its behavior based on real-time conditions. For example, if a sudden increase in liquidity demand occurred in one area of the network, the system would re-route liquidity accordingly, ensuring that the network remained balanced and responsive to idiosyncratic changes.
- **Abstract Layer & JIT Interaction:** The Solver & Routing Layer would act as an abstraction layer sitting between users (applications) and the JIT system, managing liquidity sourcing and provisioning. It would signal when and where liquidity is required, triggering JIT Actions like borrowing, moving liquidity, or reallocating resources. While users would interact with the system at a high level, the Solver & Routing Layer would handle real-time decision-making based on network conditions, ensuring that JIT Actions use optimal pathways and resources to efficiently meet liquidity demands.

Initially, the Solver & Routing Layer would focus on simple, direct routing between liquidity sources. However, as the network grows and matures, the solver would evolve to support more complex procedural routes. These could involve multi-hop liquidity transfers across several applications or even multiple chains, dynamically responding to shifting liquidity needs and maximizing capital efficiency.

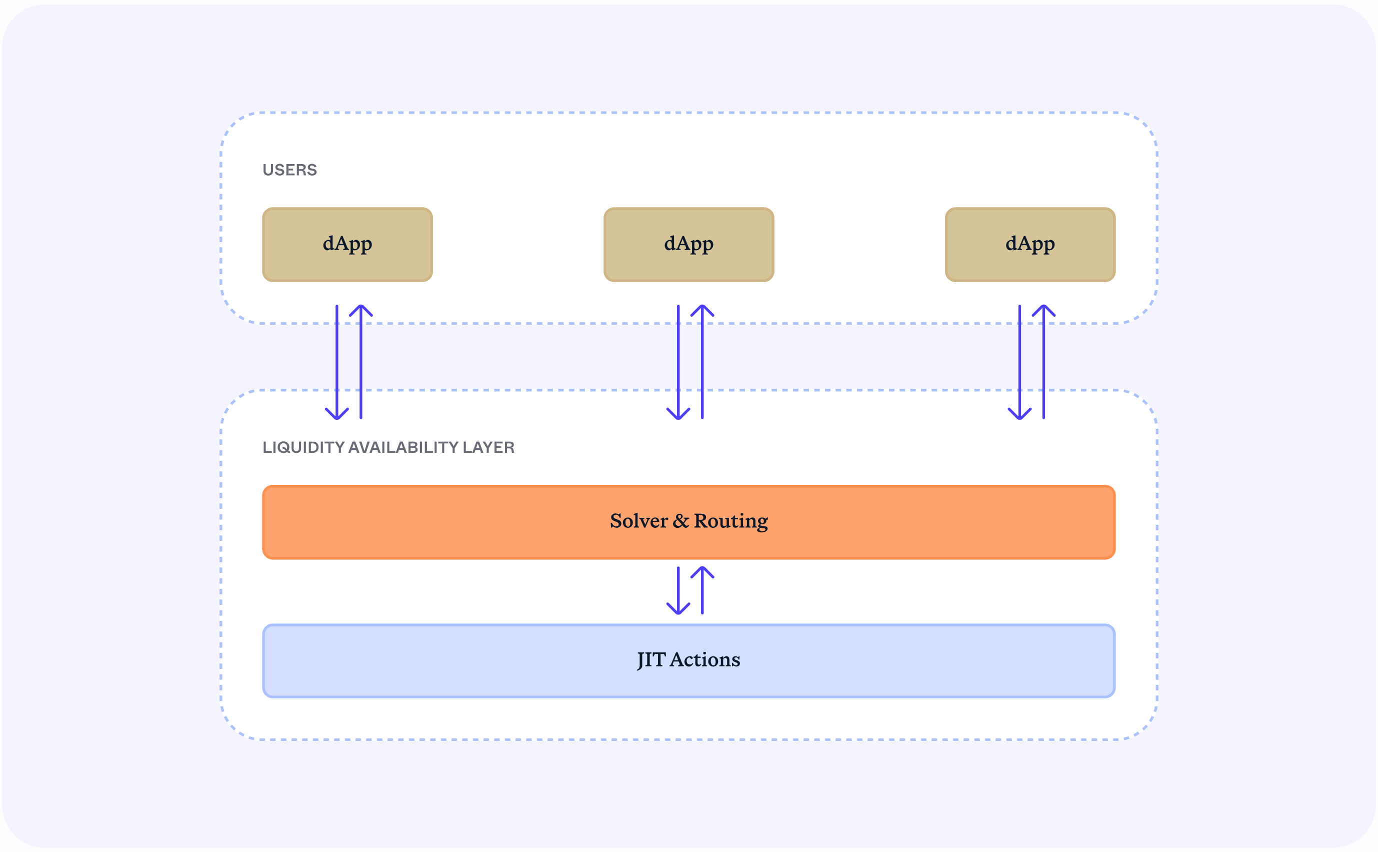

Figure 2: Liquidity Availability Layer Architecture

In a mature state, the end-to-end system would exist such that JIT is the signaling layer, with the Solver & Routing Layer sitting between the users and the JIT layer—allowing for the comprehensive Liquidity Availability Layer to instantly ship liquidity to supported applications and chains.

It is important to note that the Solver & Routing Layer described within the Liquidity Availability framework is distinctly different from existing solutions in the industry, most of which aim to solve liquidity fragmentation top-down. Current cross-chain liquidity routing solutions focus on aggregating the liquidity within individual applications spread across multiple networks. In contrast, the Liquidity Availability framework seeks to eliminate the concept of application-specific liquidity altogether. Part of its tech stack would employ a Solver & Routing Layer to optimize and streamline the sourcing and servicing of liquidity needs through JIT Actions across applications and chains, turning liquidity into a flexible, network-wide asset.

Foundational Building Blocks for Network Liquidity

Delivering on JIT Actions, Liquidity Proving, and the Solver & Routing Layer would not just be beneficial; it could provide the foundational building blocks necessary for evolving Liquidity Availability from an isolated, application-specific issue to a comprehensive network-wide challenge. By establishing these fundamental capabilities, the network could be better equipped to address liquidity needs dynamically and holistically, enabling a more resilient and fluid on-chain environment.

In addition, these foundational components lay the groundwork for further innovation. Developers would have access to a modular framework to build upon, allowing them to create new and diverse solutions that address more complex problems, across numerous industries. With these systems in place, the network would not only provide immediate, massive capability enhancements, but also support significant future growth.

A Note on Individual Participation

While this paper positions the "users" of the Liquidity Availability framework as applications, it is ultimately users that inject liquidity into networks, whether they be institutions or individuals. Under the Liquidity Availability architecture, users inherently participate if the applications to which they supply liquidity opt to utilize the resource. Additionally, users may wish to deploy liquidity directly into the Liquidity Availability layer, bypassing specific applications entirely.

To facilitate a somewhat predictable behavior, it would be prescient to ensure the capability is available. Implementing this would be rather straightforward, and could manifest as a dApp on the network. However, the "dApp" itself would have a singular, one-sided purpose—servicing the Liquidity Availability Layer. Herein, permissionless lending vaults could serve as effective vehicles. Vaults would accept a broad range of asset deposits, transmit liquidity proofs to the network, and dynamically deploy and manage liquidity, similar to other integrated dApps. By maintaining a dedicated purpose, such a servicer could potentially achieve greater capital efficiency, enabling liquidity providers to earn competitive, fee-generated yields.

The vault strategy is just one possibility for enabling this capability. At this stage of on-chain finance development, there is a broad range of existing solutions that could effectively fulfill such a role. Regardless of the specific implementation, it is reasonable to assume that such venues would self-populate, further enhancing the effectiveness and adoption of the Liquidity Availability framework.

Conclusion

The concept of Liquidity Availability offers a critical perspective on addressing the inefficiencies and challenges of current liquidity management across crypto and the emerging world of on-chain finance. This paper posits that shifting from application-specific liquidity to a network-wide Liquidity Availability framework could unlock significant growth potential, lower barriers to entry for new projects, and foster a more innovative and resilient ecosystem.

The proposed mechanisms of JIT Actions, Liquidity Proving, and a Solver & Routing Layer, while theoretical, are essential building blocks for achieving this transition. These mechanisms could redefine liquidity management from a fragmented, competitive approach into a cohesive, integrated strategy. If implemented effectively, a comprehensive Liquidity Availability framework would transform liquidity from a static, application-specific resource into a flexible, network-wide asset, allowing applications to scale without being constrained by liquidity silos.

However, the ideas explored here are just the beginning. Further research and experimentation will validate these concepts, explore practical applications, and refine the strategies for implementation. Drawing inspiration from traditional financial systems while pushing beyond their limitations provides a unique opportunity for advancing the entire on-chain economy forward.

In summary, framing Liquidity Availability as a network-wide challenge rather than an application-specific issue holds significant promise for the future of decentralized solutions. Embracing this shift could lead to more effective solutions to current limitations, paving the way for sustained growth, technological advancement, and a more equitable and dynamic financial ecosystem.

Ultimately, this approach could catalyze the next wave of innovation, driving the blockchain and on-chain finance toward a future where liquidity is no longer a limiting factor but a powerful enabler of progress.

The Blockchain Built for Finance

Liquidity Availability

Injective Research