Chain Interoperability: R3 Viewpoint

Nigel King & Ross Nicoll

September 9, 2016

With the potential existence of multiple independent blockchains / distributed ledgers in finance there is a need to understand what businesses may require chains to interact and under what circumstances, and how such interoperability might be supported. The paper identifies some high-level financial use-cases and then explores some of the current methods in the blockchain world for interoperability that might prove useful for the mainstream financial industry. It also looks at some other issues around the use of chains, such as security and governance models, that may be impacted by increasing interoperation between chains.

The paper notes that three key approaches exist today for achieving interoperability between cryptographically secured chains:

- Notary schemes, where one or more trusted parties validate events on another chain, as part of the verification of a local chain. This approach simplifies consensus mechanisms at the cost of necessitating trusted parties.
- Sidechains/relays, where the chains validate events on external chains as part of validating local events.
- Hash-locking (or atomic cross-chain trades), where events on different chains are unlocked by a single secret value.

The paper focuses on methods for inter-chain operation, rather than integration of chains with the external world. We note that the use-cases for interoperation that we envisage can equally apply between conventional systems and chains thus creating a potential need for a different class or style of interoperation.

Notary Schemes

The technologically simplest approach to consensus between chains is to have entities which validate external chain(s) and sign events on the local chain to confirm the matching external event occurred. Examples of this approach include the Interledger protocol and Blockstream's "Liquid" sidechain platform.

Notary Schemes are receiving a lot of attention in the permissioned ledger world and are probably the principal contender for providing flexible consensus without costly proof of work or complex proof of stake mechanisms. However, the concept does bring with it the danger of adding third parties into the ledger ecosystem that are both powerful (in that they can decide which transactions to allow/disallow) and scale-limiting.

Sidechains / Relays

The paper argues that the concept of sidechains is no longer a useful one, being too generic and implying a subservience between chains. Instead it is argued that the concept of different chains validating events on each other in the course of executing their own business is more helpful, as exemplified in the relay method.

Relays establish an alternative to having external events verified by a notary, where the normal verification process of the chain itself includes validation of events on an external chain. This requires that fully validating nodes also validate transactions on the external chain, which can be a significant additional effort which is partially mitigated through inclusion and validation of just the blockheaders from the external chain to simplify the validation (as in Bitcoin 'light client verification'). Of course simplified validation comes at the cost of increased security implications.

Examples may include issuing an asset on the local chain equal to an amount destroyed on a remote chain, to simulate movement of the asset between the two chains. This requires that the receiving chain verifies the destruction of the asset, in order for the issuance transaction to be valid.

The paper concludes that relays are quite a versatile method and can handle atomic swaps, asset portability (an example for asset portability is included) and a range of other use cases. Given the reliance on blockheaders for light verification we believe that further work to evaluate this approach from a security and practicality perspective is needed.

Hash-locking

Hash-locking is a form of atomic cross-ledger protocol which, rather than validate events on the other chain, seeks only to ensure that events of interest completed. This can be useful for co-ordinated trading of assets on separate chains (e.g. DVP transactions). This approach cannot move assets between chains as the sum of total of assets in issue on either side is unchanged following the transaction.

In this approach one party chooses a secret value and shares its hash with the other party. The other party then submits their transaction which is locked to a combination of the recipient's signature and the hash of the secret value.

Once the parties have submitted their transactions, the first party provides the secret value within a certain time to claim their assets, exposing the value to the other party for them to complete, also within a certain time. The first party will only reveal when they see the other transactions are there to be unlocked.

Time-out limits (nSeconds) are used in this scenario to ensure completion or roll-back at an atomic level, but the fact that there is a defined sequence to the order of events does allow for the possibility of gaming the system.

The other implication of this approach is that the locking process has to be built into each ledger and each side of the deal needs visibility into both ledgers.

Hash locks are not useful for read-only type interoperation either and so appear to us to have limited application for financial use cases.

Security Implications and Recovery from Failure

The paper notes that chain interoperability can have significant implications for both security, and error recovery.

In the case of pegged sidechains, there are complex questions around behaviour in case of the external chain forking; if a transaction was validated based on an external event occurring, and the external event no longer occurs, theoretically the local chain must also be forked before the now invalid transaction. If multiple chains are sidechained to each other, this could theoretically trigger further forks of other chains.

Related to this, depending on how the validation is performed there is a risk of invalid events being considered valid by a dependent chain, and from a financial system perspective this type of risk must be negligible or zero.

In the case of hash-locking, care must be taken that a hostile actor cannot convince others to commit significant funds to a transaction which cannot be refunded for an extended period of time. Such "financial denial of service" could be used as part of an attempt to interfere with normal operations of the entity, or its ability to meet existing commitments.

The paper covers a number of other scenarios, which only heighten the importance of a thorough security assessment of each of these nascent approaches to interoperability.

Lifecycle Events

The paper goes on to discuss some of the numerous economic and geopolitical events that will impact the lifecycle of interoperable chains, noting that new chains may come into being, existing ones be shut down, their *raison d'etre* cease to exist, chains be subject to forks (as with existing Ethereum), links between them forbidden, or they become subject to other sanctions or censorship.

Governance models for chains and the technical links between them will need to be able to gracefully handle such events, and the complexity of that task is magnified the more interoperation links that exist.

The paper goes on to argue that it may be difficult, in practice, to physically prevent chains from interoperating, or even detect that the interoperation is occurring.

In summary the paper introduces a number of approaches for handling interoperability, each with their own strengths and weaknesses. It is a cogent starting point, however there is significant further work to be done to evaluate feasibility of these solutions in the real world, as well as to ensure that they perform as anticipated. Security implications are a particular concern that requires very careful consideration before solutions could be deployed in a large scale environment.