TL;DR

Introducing a.DI (Aave Delivery Infrastructure) v1.1, improving scalability, security and permissions granularity on the system currently used for all cross-chain communications of the Aave DAO.

Context. a.DI until now

Since November 1st 2023 the Aave Delivery Infrastructure (a.DI) is the system powering in production all the cross-chain communication needs for the Aave DAO, mainly its multi-chain governance (Aave Governance v3).

But like any other Aave project, its activation only marked the beginning of the road, and since end of 2023, multiple improvements have been made:

- Added coverage for new underlying bridge providers:
- Native Scroll bridge.
- Wormhole compatibility.
- (and others in progress).
- Native Scroll bridge.
- Wormhole compatibility.
- (and others in progress).
- Maintenance upgrade of CCIP, LayerZero and Hyperlane adapter to their most up-to-date versions.
- Minor optimisations of the a.DI core contracts, related with gas granularity and meta-data.
- Released a.DI Monitoring, an interface to get detailed visibility on all messages going through a.DI.

In parallel, we have been preparing 2 other updates to be batched in a single upgrade: a.DI v1.1.

The new v1.1

v1 - The bottleneck of underlying bridges

a.DI v1 has one constraint, that even if not major, limits the scalability of the system. Whenever a new underlying bridge provider is added to the set of a communication path, each message on that path will be passed through all the bridges, no matter if the consensus required on the recipient side is lower.

For example, currently all messages Ethereum → Polygon are sent via CCIP, LayerZero, Hyperlane, and the Polygon native bridge, even if the consensus required on the Polygon side is 3-of-4. This means that even if consensus was reached already before the slowest bridge delivers the message (Polygon native in this case), the cost incurred is of the whole set of 4 bridges, not only 3.

The main consequence of the previous is that adding new providers on existing paths was not really an option whenever others have shown reliability, as each new provider would add extra cost for each message passed, even if the consensus requirements would not change: the cost of 3-of-4 would be lower than 3-of-5 and so on.

This is problematic, because the core idea of a.DI is security by consensus, so if properly handled, the most diverse set of bridges, the better.

v1.1 features - a.DI Shuffling

The main feature of a.DI v1.1 solving the previous bridges' bottleneck is the new Shuffling mechanism.

a.DI Shuffling introduces 2 new concepts in the forwarding side of a.DI:

- Bandwidth

. The total number of underlying bridges whitelisted on a.DI for an specific communication path.

For example, in a path with 4 whitelisted underlying bridges, 4 is its bandwidth.

- Optimal bandwidth

. A parameter regulating how many underlying bridges will be used to send a message, always lower or equal than bandwidth, and loosely correlated with the threshold on the recipient side of a.DI.

For example, in a path with 4 whitelisted underlying bridges, and a threshold on the recipient side of 3, the optimal bandwidth will be by general rule also 3.

With those concepts defined, the flow of a.DI with Shuffling is simple: whenever a message needs to be sent, the system chooses pseudo-randomly a sub-set of all the available bridges for that specific path (e.g. Ethereum → Polygon) with optimal bandwidth size, and only forwards the message via those bridges.

Cross-chain forwarder with Shuffling for bridge selection

By doing this:

- Cost is reduced on all those configurations with optimal bandwidth < bandwidth

.

- Any bottleneck to add new underlying bridge providers is removed, as increasing bandwidth doesn't really have any effect on the optimal bandwidth of the path.
- Security-wise, the pseudo-random nature of the shuffling mechanism improves the system, in the line of other rotation-based systems like validators-selection in PoS blockchains.

v1.1 - Granular control

On v1.1, Shuffling is an "optimistic" mechanism: we assume that is very safe to configure the optimal bandwidth of a communication path equal to the threshold required on the recipient side, as historically we have observed that pretty much all messages passed via any bridge of a.DI have been delivered on destination without any problem.

However, Shuffling highlights the hypothetical need of the retry mechanisms included on a.DI. If the optimal bandwidth is configured as 3, the threshold on recipient side is 3 and for any reason one of the bridges is temporarily not operational, a.DI allows to retry the message via another bridge not selected before, to fulfil the recipient threshold.

On v1, access control of the retry mechanism and emergency solving was monolithic, with only 1 entity (denominated a.DI Guardian) being able to do both, no matter if the higher-order one (solve emergency) would be dependent on a trigger by the Chainlink Emergency Oracle.

On v1.1, a.DI's access control is now extracted to a GranularGuardianAccessControl

smart contract, and the roles are the following:

- DEFAULT_ADMIN_ROLE.

The "super-admin" of the system, with permissions to grant/remove roles. To be configured on the Aave Governance smart contracts.

- RETRY_ROLE

. Entity capable of calling retryEnvelope()

and retryTransaction()

, for example when a message was sent via one bridge temporarily not operative, and to fulfil recipient threshold it should be sent via another. To be configured to a technical entity contributing to Aave, currently BGD.

- SOLVE_EMERGENCY_ROLE

. Entity able to call the solveEmergency()

whenever the Chainlink Emergency Oracle is triggered by the Aave Governance. To be configured to the Aave Guardian.

Next steps

Similar as with any other major upgrade on production systems, the steps required are the following:

- Finish security procedures. Currently the security service provider of the community (Certora) is reviewing the changes, and we will study engaging an additional one.

- Create an ARFC Snapshot for the pre-approval by Aave governance.
- If the ARFC Snapshot is positive, proceed to the on-chain AIP, executing all the necessary smart contracts changes/configurations for the full activation of a.DI v1.1.