Rohu-Paplet

Fast Finality and Continuous Asynchronous Recovery

Naman Bajpai

November 3, 2026

Abstract

A timeout is not an eraser: votes already cast can still become a certificate, even after recovery has moved on. Rohu-Paplet makes fast finality and continuous asynchronous recovery agree on what survives that awkward moment. Independent producer chains supply retrievable transaction data; accepted DAG cuts define deterministic ordered deltas; each certified parent opens one successor domain. River drives fast finality, while a continuously running Ocean DAG orders recovery records. Recovery permanently closes the affected domain, preserves any child that could still acquire a fast certificate, and certifies exactly one successor. It appends no recovery tail that could collide with descendants already making progress.

For a fixed committee of $n = 5f + 1$ identities with at most f total faulty or unavailable identities, we give quorum-preservation and certified-prefix arguments, a native DAG ordering construction, and conditional progress and service bounds. An eligible proposal with complete evidence can be ratified in two message delays; fresh data must first satisfy availability and cut dependencies. A stalled parent needs one effective ordered resolution, a count of coordination stages rather than a bound on elapsed time or total work. The guarantees assume durable signing state, complete evidence, fair retrieval, and the stated ordered-service and coin assumptions.

The paper connects availability, ordering, finality, crash recovery, and blockchain delivery in one consensus design. A signed parent-local reference kernel and Ocean-model adapter pass 32 tests covering late certificates, proof aliases, concurrent descendants, and interrupted signing journals. Earlier native components and separately measured transport and extension experiments inform the implementation discussion; full native integration and end-to-end network evaluation remain open.

1 Introduction

Byzantine consensus must connect three statements that are easy to confuse: some data exists, validators agree on its position, and an application can consume the agreed result. A certificate for a block digest establishes the first statement only under a custody contract. Agreement on selected DAG anchors establishes the second only with a deterministic interpretation of their complete ancestry. A fast certificate establishes irreversible authority only if subsequent recovery preserves it, including when nobody participating in recovery has yet assembled that certificate.

Rohu-Paplet addresses this composition problem. Producers disseminate transaction blocks on independent chains. Validators certify compatible data paths and continuously construct a control DAG. An accepted source cut supplies evidence for a deterministic ordered delta. River, the fast path, asks a designated leader to propose an exact child of an already-certified parent. Ocean, an asynchronous ordering service, continuously orders control records and resolves stalled child opportunities. Both routes authorize the same kind of semantic transition and therefore feed one application order.

1.1 The late-certificate problem

An unsuccessful-looking proposal need not be unable to finalize. Messages already signed may be delayed, and a Byzantine collector may reveal them only after recovery makes another choice. Recovery therefore needs evidence about what *could* already finalize, not merely a report of what its participants have learned. The evidence must also survive the loss of a certificate's first recipient: a signed promise to possess a complete artifact must actually imply retention and eventual retrieval.

There is a second difficulty once certified descendants are allowed to run. Suppose a late certificate proves $p \to x$, and another observer has already certified $x \to y$. Recovery of p must authorize exactly x. It cannot preserve x and append an additional recovery transition after it, because that would create a competing successor of x. Nor can it distinguish two proofs for x as different parents: otherwise honest validators could sign conflicting children in what they mistakenly regard as separate domains.

The current construction consequently places exclusion at an exact semantic parent. Its successor domain has one fast attempt and can close permanently. Closure carries the signer's earlier choices and complete retained protection. An ordered resolution selects one immediate child, preserving possible late finality. The certified child enables the next domain. No reopening of the old parent is needed, and an ancestor's closure does not close descendant domains.

1.2 Contributions and scope

The paper makes four concrete contributions within the stated model:

1. A complete data-to-certificate interface: compatible producer paths, accepted cuts, exact semantic children, replaceable proof artifacts, and deterministic ordered deltas.
2. A parent-local preservation construction, with quorum arguments showing that closure and ordered resolution preserve possible late FAST and receipt finality, and an induction yielding one certified prefix across both routes.
3. A connection to native asynchronous DAG ordering, including broadcast, coin, complete-cone, and first-resolution obligations, together with explicit progress premises and structural recovery accounting.
4. An implementation and falsification record that separates the signed reference construction, its ordering adapter, earlier native components, and independent transport/extension experiments.

The established algorithmic comparison is with the archived era baseline, the earlier era-based design: one parent resolution replaces the barrier, executed checkpoint intent, and Heal ownership cycle. General fast/fallback composition and DAG interpretation have substantial prior work [10, 11, 12]. The present arguments do not establish optimal complexity or a universal performance advantage over those systems.

This is a consensus paper. The application supplies deterministic validity and consumes certified ordered deltas. Section 11.3 explains the blockchain interface, replay, and state-synchronization obligations. Virtual machine internals and application-specific execution architectures do not enter the protocol or its performance claims.

1.3 How to read the claims

Safety is conditional on authenticated identities, retained evidence, durable signing restrictions, deterministic validity, and a common ordered transcript. Almost-sure progress additionally uses eventual service and an ideal-coin contract. Numerical delay bounds require stronger service envelopes. Finite tests corroborate selected transitions and adversarial sched-

ules; they do not quantify over all executions. Section 11.4 maps these distinctions to the exact artifacts and proof obligations.

Sections 2–5 define the model and data path. Sections 6–8 specify the two finality routes and their ordering substrate. Sections 9–10 connect them through safety, progress, and cost arguments. Implementation and evaluation then identify what has been realized and what remains an integration obligation.

2 Model, assumptions, and guarantees

2.1 Participants and configuration

Fix $f \geq 1$ and a committee N of $n = 5f+1$ authenticated identities. At most f identities in total may deviate, disappear, or lose their safety state over the execution. This is a total-fault budget, including later failures, rather than f Byzantine identities plus a separate crash allowance. Quorums count distinct configured identities, not messages. A restarted process is the same identity and must restore its earlier restrictions before signing again.

Producers maintain signed transaction-block chains; a producer may also be a validator. The configured ordered producer list determines extraction order. Validators retain data, sign protocol evidence, and author control vertices. Observers verify finality and consume the resulting order without acquiring a vote. Clients submit transaction data to producers; client admission itself is not a consensus certificate.

The configuration commits to committee and producer order, public keys, protocol and encoding versions, genesis, application semantics, resource limits, and the Ocean instance. All signed objects and hash domains bind this configuration. The paper considers fixed-committee operation. A certificate under a different configuration is not usable merely because both committees have the same size. Membership changes require an additional transition rule and are discussed as an integration boundary in Section 11.3.

2.2 Communication, validity, and persistence

Honest-to-honest messages are eventually delivered and processed. Messages may be reordered and delayed arbitrarily. Executions are locally finite and every record is finite. Safety does not require a known delay bound. When a timed statement uses δ, its additional synchrony, processing, and readiness conditions are given explicitly. Signatures are unforgeable and hashes collision resistant; exact encodings and signature kinds are part of the verifier.

Assumption 1 (Durable evidence and common validity)**.** An honest signer records its signing choices, retained complete evidence, outbox, and closure tombstones before releasing the corresponding signature. Restart restores them without rollback. Honest evidence holders retain and serve their promised bodies and dependencies. Complete child validity is a common deterministic predicate. Missing required evidence yields Pending; it does not yield a different locally valid child. The baseline retains history and does not provide bounded-storage garbage collection or rollback detection.

The durability condition is a publication barrier, not a requirement that every implementation use a particular database. If a storage callback fails, an output justified only by the unacknowledged transition cannot escape. A signer whose state is lost must remain unavailable until safely recovered. Starting a new empty journal under the old key would violate the model.

Assumption 2 (Continuous ordered recovery service)**.** Ocean supplies portable authenticated evidence of a prefix-compatible ordered control transcript. Readers process the same complete records in the same order. Continually advertised eligible honest records are eventually ordered almost surely. Control progress does not depend on application execution, resolving another parent, or a successful fast proposal.

Section 8 supplies the native DAG construction and states its broadcast, ancestry, fair-inclusion, seal, and ideal-coin dependencies. The ordered transcript and its complete interpretation are separate from an observer's currently executed application head.

2.3 What consensus guarantees

Certified-prefix safety.
All valid finality proofs descending from the configured genesis describe prefix-compatible semantic histories and deterministic transaction order.

Certified progress.
Under the progress premises, every certified parent admitted by an honest participant eventually has a final child known to all honest observers, almost surely.

Qualified data inclusion.
Continually advertised, DA-certified data is eventually included under the retention, fair ancestry, and source-cut rules. This does not promise inclusion of an arbitrary transaction that no honest producer admits or certifies.

Application refinement.
If consumers start from the same state and apply the certified deltas with the same deterministic transition function, they obtain the same state at equal positions. Eventual execution additionally requires payload retrieval and application service; a finality proof alone does not assert that execution has already completed.

Notation	Meaning
p, x	Certified semantic parent and a proposed immediate child
d_p	Permanent successor-signing domain derived from $\mathsf{id}(p)$
F_p, L_p	Parent's certified producer frontier and ordered-log commitment
c_x, r_x, D_x	Child's accepted source cut, cut round, and ordered block delta
$U_p(x)$	Complete $3f+1$-PREPARE protection certificate
S_p	Exactly $4f+1$ complete closed statuses for d_p
Q, W	Ocean parent/seal quorum $4f+1$ and wave length

Table 1: The semantic object, its evidence, and the control clock are distinct.

3 System overview and a worked execution

The system has a data plane, a continuously advancing control plane, and one semantic finality interface. Data certificates establish compatible retrievable producer paths. Accepted control vertices expose finite source cones. A proposal uses a complete cone to construct an exact child of a certified parent. River can finalize that child directly; otherwise closed statuses constrain an Ocean resolution. Application delivery follows the certified parent chain, regardless of which route supplied a certificate.

3.1 From a transaction to a certified child

Consider $f = 1$, six validators, and a common certified parent p. A producer places transaction t in a signed block b. Four distinct DA votes certify the compatible path through b; at least three of those voters are honest complete-data holders. An accepted control cone advertises this certificate. The designated leader derives a frontier extending F_p, sweeps its newly included blocks, and proposes child x with t at a deterministic position. The source-cut round is strictly newer than p's, and all prerequisite evidence is available to the proposal verifier.

Open validators persist a choice for x and multicast PREPARE. Five distinct PREPAREs form FAST, allowing an observer to certify x. Alternatively, a complete four-PREPARE certificate $U_p(x)$ can be retained and acknowledged by three distinct receipt signers, producing FIN. A receipt is evidence of possession of the complete U, not another name for an isolated prepare vote. After verifying either proof, an observer enables the domain for children of x. The application can consume D_x once its payloads and predecessors are available. It need not wait for a recovery record for p.

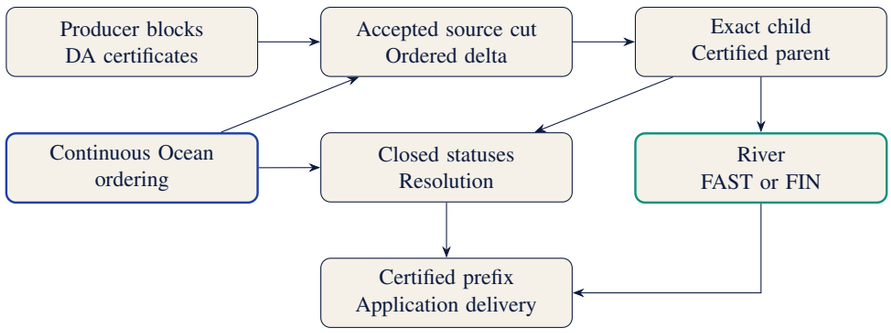


Figure 1: Dependencies of finality. Ocean supplies accepted cuts and orders recovery records continuously. Only a valid finality route authorizes a child.

3.2 The same execution with a late certificate

Suppose five PREPARE signatures for x exist but their assembled FAST is withheld. A timeout closes d_p at an honest signer. Closure disseminates, and five complete statuses are collected. Among the FAST signers and status authors are at least three honest identities whose earlier PREPARE for x must appear in their status. The selection rule therefore preserves x even though nobody collecting statuses has seen FAST. For a hidden FIN, at least one honest holder of its complete U occurs in any five-status set, supplying the corresponding preservation evidence.

A validator advertises a complete resolution record. Ocean's first valid resolution for p certifies x. If another observer already learned FAST and certified a descendant $x \to y$, that resolution is another proof for x; it has no authority to create an alternative successor of x. When no child is forced by the statuses, the record instead supplies a complete valid fallback child. Its delta can be empty only if its source cut and semantic height still advance under the normal validity rule.

3.3 Four distinctions used throughout

Data availability is distinct from transaction ordering. An accepted DAG vertex is distinct from an ordered recovery decision. A preservation status set is distinct from agreement on a recovery choice. Finally, a semantic child is distinct from its proof bytes. These distinctions explain why a system that merely has a working fast path, DAG, and persistence layer still needs a composition argument.

4 Data availability and deterministic extraction

4.1 Producer chains and the custody promise

A producer block has an authenticated header identifying the configuration, producer, increasing sequence number, exact predecessor, payload commitment, and committed transaction positions and identities. Each producer starts at a configured genesis. An honest DA voter durably chooses one append-only branch per producer and retains the complete payload path needed to justify every vote. It may not vote for conflicting branches at the same or different heights. The certificate for a block contains $3f+1$ distinct DA votes with matching bindings. This is a full-holder availability contract.

Lemma 3 (Certified data paths)**.** *DA-certified paths of one producer are compatible. Every certified block and the ancestry promised by its votes have at least $2f+1$ honest complete-data holders and are eventually retrievable under Assumption 1.*

Proof. Two sets of $3f+1$ voters intersect in at least $2(3f+1)-(5f+1) = f+1$ identities. At least one is honest and cannot vote on conflicting branches. A single certificate leaves at least $3f+1-f = 2f+1$ honest voters, each having retained the complete promised path. Their service and eventual delivery supply retrieval. The proof uses the branch restriction across heights, not merely a prohibition on voting twice at one height. □

A Byzantine producer may withhold a new block or strand an uncertified branch. Neither prevents the control DAG from advancing without that advertisement. Honest data holders continue serving earlier certified blocks. The baseline does not terminate their promise after a local timeout or after observing a new head; replacing it by checkpoint custody needs a separate retention rule.

4.2 Frontiers, cuts, and their certificates

A frontier F gives one certified position for each configured producer. A coordinate can remain at its parent's position or advance along a compatible DA-certified path. Lemma 3 makes valid positions on a producer chain comparable. Their greatest position is therefore well-defined.

A source cut c identifies an accepted Ocean vertex and its complete finite strong-and-weak ancestry, with authenticated bodies and accepted-slot proofs. The cut's physical round $r(c)$ is the root vertex's round. Source cuts are available before their vertices need be selected as Ocean ordering anchors. Acceptance proves the cut's contents; it does not by itself ratify a proposed application transition. Section 8 defines accepted vertices and the distinction between their strong and weak edges.

Starting at F_p, extract every valid compatible DA-certified coordinate advertised in the complete cut and take the coordinatewise maximum. The full contribution is mandatory. An invalid complete data advertisement contributes nothing under the common parser and verifier; a missing required opening is Pending and cannot be mistaken for an empty contribution. The source root, configuration, prior frontier, and derived vector are semantic commitments; their enclosing certificate signer subsets are replaceable evidence.

We call the resulting portable evidence a FrontierCert, or FC. It is a *candidate* until an exact child carrying its vector and source cut is ratified. Two valid candidates may be compatible coordinatewise but induce different transaction interleavings. The ratified parent chain selects the actual order. In this revision, an Ocean anchor alone does not create a new application FC: an effective parent resolution must certify an exact child.

4.3 Optional complete vector evidence

The conservative v4 evidence interface also permits a complete proposal-relative vector pool to supplement the mandatory cut. Such a pool contains $4f+1$ distinct signed records binding one configuration, source proposal, predecessor frontier, and vector schema. Each record supplies its body and authenticated compatible paths. On a proposed path, a position supported by at least $3f+1$ records is eligible; beyond a fully supported proposed tip, selection requires $4f+1$ support. A $2f+1$ extension-support result is carry evidence for a later proposal, not authority to execute that extension now. The base coordinate is the exact parent coordinate or a certified descendant, and every newly selected coordinate still requires DA evidence.

The child can take the coordinatewise maximum of the cut and eligible vector contributions. It commits one exact result before PREPAREs are signed. Different complete pools are permitted to justify different candidates; agreement comes from ratifying one child. This optional evidence form must not be confused with votes that individually change the child after its proposal. The latter can produce variable-length certified prefixes and needs the additional composition interface

discussed in Section 12.3. The signed parent-local kernel abstracts both cut and vector validation through its validity callback.

4.4 Sweep and duplicate handling

For each producer, enumerate blocks strictly after its parent coordinate and through its new coordinate. Sort this union lexicographically by producer sequence, configured producer index, then block digest. This is Sweep. Transaction order within a block is its committed order. Use the first occurrence of each canonical transaction identity in the delivered prefix, with duplicate handling included in the configured deterministic consumer. All observers use the same identities and predecessor history; a transaction already delivered cannot become a new effect on replay.

The child's ordered block delta and new log commitment bind this result to the exact predecessor. A receiver computes the whole delta before appending it. It cannot sweep only the currently visible part of a cone. Missing payloads may delay transaction decoding and execution, but cannot change the already certified order of block identifiers. Application validity that requires those payloads must await them before the corresponding validity decision.

4.5 The inclusion obligation

Every child names a cut with round strictly greater than its parent's cut round and includes its entire contribution. Honest Ocean authors include their own previous accepted vertex as a strong parent. They also serve an arrival-ordered FIFO of older accepted vertices absent from their own cone, using a positive bounded number of weak references per new vertex whenever that queue is nonempty. Arrival means complete accepted admission. Sorting the selected references canonically must not allow newer arrivals to overtake an older queued vertex.

Lemma 4 (Permanent advertisement inclusion)**.** *An accepted honest advertisement is in the complete cone of every sufficiently late valid Ocean vertex, under eventual broadcast and fair FIFO authoring.*

Proof. Broadcast eventually supplies the advertisement to every honest author. Local finiteness leaves only finitely many earlier admissions ahead of it. Continuing positive FIFO service therefore puts it into each author's cone, and the mandatory own-parent edge retains it permanently. Choose a round after this has happened for all honest authors. Every later $4f+1$ parent quorum contains an honest author's vertex and hence the advertisement in its complete ancestry. Induction extends the property to all subsequent valid vertices. □

The lemma requires an honest advertisement of the certified data. It makes no fairness statement about a client request never admitted or about a producer that never obtains its DA certificate. Its role is to prevent infinitely many otherwise successful children from omitting already eligible data forever.

5 Semantic objects and durable validator state

5.1 An exact child and its identity

The configured semantic genesis has height and source-cut round zero, initial producer frontiers, and the initial ordered-log commitment. Its authority comes from the agreed configuration rather than a collected runtime certificate. It is distinct from Ocean's round-zero seed vertices. Child height one therefore uses configured leader zero under the baseline rotation.

A child of certified parent p is the semantic tuple

$$x = (\mathit{config}, h_p + 1, \mathsf{id}(p), F_p, F_x, c_x, r_x, L_p, D_x, L_x).$$

Here F denotes the certified frontier, c_x the complete accepted source-cut commitment, $r_x > r_p$ its physical round, D_x the canonical ordered block delta, and L_x the extended log commitment. The configured encoding fixes how vectors and deltas are committed and opened. The child identity is

$$\mathsf{id}(x) = H(\texttt{semantic-child}, x),$$
$$d_p = H(\mathit{config}, \texttt{child}, \mathsf{id}(p)).$$

Proposal signatures, signer subsets, delivery routes, and Ocean proof representations do not occur in either semantic identity. Frontier and cut commitments identify their semantic contents, not enclosing DA or broadcast certificate bytes. The domain has no era number or retry counter.

Complete child validity. The verifier checks configuration and canonical encoding; exact predecessor and height; a complete finality proof for p; a complete accepted cut at a strictly later round; compatible certified frontier contributions; mandatory inclusion of the entire cut; and recomputed delta and log commitments. Optional vector evidence must satisfy Section 4. An empty delta is legal only when these same checks succeed. Renaming an arbitrary empty value is not a recovery opportunity.

Fast proposals additionally require the designated leader's signature. An Ocean-selected fallback needs no leader proposal signature because its authorization is the ordered resolution. Both routes still check the identical semantic child predicate. A finality proof for the parent is required before voting for a child; speculation on an uncertified parent is outside this rule.

5.2 Evidence types and quorums

PREPARE and receipt signatures bind $(\mathit{config}, d_p, \mathsf{id}(x), \mathit{kind})$. A receipt acknowledges possession of a complete valid $U_p(x)$ and carries that protection. It does not split aggregation by requiring every signer to have retained the same U signer subset. A FIN verifier checks the common child binding, distinct receipt authors, and complete valid prepare protection. An honest receipt issuer retains its own complete supporting artifact for later status and retrieval.

A closed status binds d_p and contains its author's own PREPARE, if any, and every complete U protection retained before closure, including protection for a receipt. A signer may have prepared a different child from the one in its retained U; the status must disclose both. Hiding that distinction would invalidate receipt preservation.

5.3 State and the publication barrier

Each validator maintains a durable map indexed by d_p. An entry contains the chosen PREPARE or reserved signing intent, retained complete protection, released receipt, immutable closed status if any, and corresponding outbox. Learned finality is recorded separately; learning a child does not erase its parent's entry. A monotone timer-backoff floor also survives parent changes. Ocean maintains its accepted DAG, seals and coin authority, emitted transcript, and first-effective-resolution map. These are distinct from the application delivery cursor.

The common mutation discipline is:

1. Validate the input and complete dependencies against the configured instance and semantic parent. Stop with Pending if required evidence is missing.
2. Reserve the permitted choice or closure in durable state. Once reserved, recovery cannot choose a conflicting signature even if no signature was sent.
3. Produce and durably retain the authorized signed output and outbox. Publish only after the persistence contract acknowledges the transition.
4. After restart, validate identity and state consistency, restore the restrictions, and retransmit the retained outbox as needed.

This gives a publication contract for implementations; it does not assume that the current in-memory reference journal is a physical disk implementation.

Object	Distinct authors	Meaning
DA certificate	$3f+1$	Complete compatible producer-path custody
$U_p(x)$	$3f+1$ PREPAREs	Unique prepare protection for an exact child
$\mathsf{FAST}_p(x)$	$4f+1$ PREPAREs	Immediate-child finality
$\mathsf{FIN}_p(x)$	$2f+1$ receipts	Finality backed by complete retained $U_p(x)$
S_p	$4f+1$ closed statuses	Permanent closure evidence for one parent
Closure trigger	$f+1$ closed statuses	Amplify closure at a local signer
STORE certificate	$4f+1$	Complete-object and dependency retention

Table 2: Evidence counts are typed: a STORE quorum is not a finality quorum, and a receipt is not an isolated PREPARE.

5.4 Validity, absence, and proof dependencies

All verifiers distinguish Valid, Invalid, and Pending. A known wrong digest, invalid signature, conflicting binding, or forward Ocean authority is Invalid. A promised but not yet fetched earlier opening is Pending. Only complete deterministic invalidity permits a record to be skipped.

Proof dependencies are finite and acyclic. Equivalent proofs establish one semantic identity, but an individual proof artifact can still be unusable at a particular Ocean position. For example, replacing a FAST proof by an Ocean proof from a future position cannot justify an earlier resolution. Section 7 specifies this recursive precedence rule and the additional firstness evidence needed for portable recovery finality.

6 River: fast finality

The leader at child height $h \geq 1$ is configured deterministically as $(h-1) \bmod n$ in the baseline rotation. The exact certified parent enables one attempt. A timer, alternate certificate encoding, or later arrival of evidence cannot change the leader or create another attempt for that parent.

6.1 Proposal, prepare, and possession receipt

The leader proposes an exact valid child and its complete parent, cut, frontier, and availability evidence. Each open validator verifies the designated-leader signature and complete child predicate. If it has not chosen another child in d_p, it persists its choice and multicasts one PREPARE. Duplicate messages can be answered from the outbox; they cannot authorize a new choice.

A complete certificate $U_p(x)$ contains $3f+1$ matching PREPAREs. A validator may retain it even if its own PREPARE named another child. If its domain remains open and its receipt restriction permits it, it persists this complete protection and multicasts a possession receipt. The prepare intersection proof in Section 9 guarantees uniqueness of the protected child. Receipts certify retention of that evidence; they do not assert that the receipt signer originally proposed or prepared the same value.

Fast-path transition rules. For an admitted parent p, the protocol executes the following guarded actions:

1. **On PROPOSE**(p,x)**.** Verify leader and complete child validity. If d_p is open and its chosen PREPARE is empty, durably choose x and multicast PREPARE$(d_p, \mathsf{id}(x))$. If the stored choice is already x, only retransmit; if it differs, do not sign.
2. **On complete** $U_p(x)$**.** Verify distinct authors, signatures, child bindings, and validity. Retain the complete artifact. If open and no receipt has been released, durably record the receipt and multicast it.
3. **On FAST or FIN.** Verify its typed quorum and dependencies. Record the exact certified child, gossip its proof, and enable d_x immediately. This action is allowed even if d_p has closed.
4. **After local closure.** Create no PREPARE or receipt in d_p. Continue serving its cached status, retained evidence, and learned finality.

6.2 Two ways to learn the same finality

An observer finalizes x on $4f+1$ matching PREPAREs, forming $\mathsf{FAST}_p(x)$, or $2f+1$ matching complete-U receipts, forming $\mathsf{FIN}_p(x)$. Both prove an exact immediate child of p. The observer need not wait for Ocean to process a resolution for p before proposing or voting for a child of x.

For $n = 5f+1$, two complete U quorums intersect in $f+1$ identities, including an honest signer. Since an honest signer prepares only one child, distinct valid Us cannot protect conflicting children. FAST contains such protection; FIN explicitly acknowledges it. Fewer receipt signatures do not mean weaker dependency checking: their value is that enough honest holders retain the complete artifact for any later closure set to encounter one of them.

6.3 When the fast path does not finish

The fast path has no internal view-change rule. If the parent remains unresolved, a finite local timer can close its domain and invoke the procedure in Section 7. A closure is a promise about future signing, not proof that the old child never finalized. An honest proposal can be preempted by an early timeout; safety survives because recovery preserves its possible finality, while fast-service claims must explicitly require signers to remain open for the stated interval.

7 Parent-local closure and finality transfer

Recovery answers a precise question: which immediate child may be certified after some participants have stopped signing for this parent? The answer must preserve evidence that can become visible later. Closure prevents additional honest old-domain signatures, complete statuses constrain the permissible child, and Ocean chooses one common resolution. These three functions are separate; a status set alone is not an agreement decision.

7.1 Permanent closure

A local finite timeout for an unresolved child closes only d_p. Receipt of $f+1$ valid closed statuses for d_p also causes closure and status multicast. Closing atomically records both the prohibition on future old signatures and the complete status before publishing it. A signer continues serving valid old-parent requests after advancing, returning its cached status and available finality evidence. Closing an ancestor never closes a child. Learning a late valid finality proof is permitted after closure; creating a new old signature is not.

7.2 Protection extracted from a status set

Given exactly $4f+1$ distinct complete statuses S_p, verify all signatures, parent bindings and embedded evidence. Define protect(S_p):

1. If a complete $U_p(x)$ is present, select its child x.
2. Otherwise, if at least $2f+1$ statuses contain their authors' own PREPAREs for x, select x.
3. Otherwise return $\perp$, meaning no child is forced by this set.

Two conflicting complete U certificates are impossible under the fault contract. Two distinct children cannot each have $2f+1$ own reports in a set of $4f+1$. A verifier rejects inconsistent evidence rather than arbitrarily choosing among it. A status set is preservation evidence, not an agreement certificate: different valid sets can return different unfinalized children.

7.3 The resolution record and its authority

Any validator may form $R = (p, \pi_p, S_p, z)$, where π_p proves the exact parent and z is a complete valid fallback child derived from a strictly newer accepted source cut. This fallback

Time in the eligible case	Message	Observer's new knowledge
s	PROPOSE with complete evidence	Leader fixes the exact child
$s+\delta$	Validators multicast PREPARE	Validity and durable choice are checked
$s+2\delta$	$4f+1$ PREPAREs arrive	FAST can be assembled at each included observer
$s+2\delta$	Complete U holders issue receipts	Receipt route has its causal prerequisite
$s+3\delta$	$2f+1$ receipts arrive	A freshly constructed FIN can be assembled

Figure 2: A message-delay ledger with prerequisites ready, open signers, and processing/persistence included in the service envelope or charged separately. The smaller receipt quorum does not remove the extra causal leg.

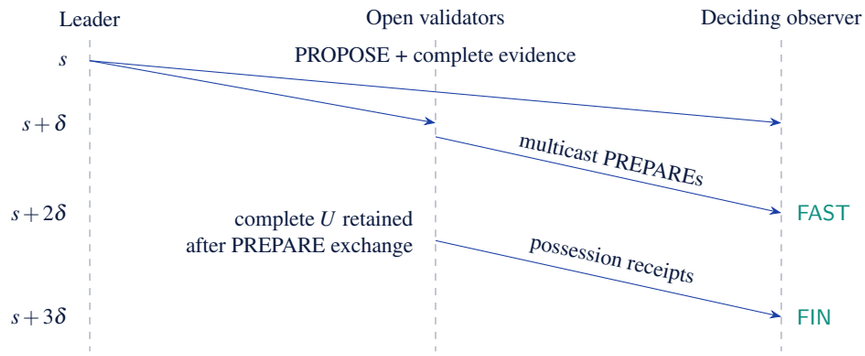


Figure 3: Fast-path message sequence. The validator lane represents the honest quorum and its internal multicast exchange. Each included observer has the complete proposal evidence; collector-only dissemination has a different observer latency. The same readiness and service premises as Figure 2 apply.

need not have the designated fast leader's signature. The common selection function is

$$\text{child}(R) = \begin{cases} \text{protect}(S_p), & \text{protect}(S_p) \neq \perp, \\ z, & \text{otherwise.} \end{cases}$$

Use the normal form $z = \text{protect}(S_p)$ when protection is present; the protected old child already has a valid cut strictly newer than p. Thus a resolution need not carry an unused additional fallback. In the unprotected case any complete valid fallback child is eligible. All dependencies, including the selected old child, must be retrievable under the storage contract. An external object reference is eligible only with $4f+1$ distinct STORE signatures committing to its type, canonical length, content hash, chunk Merkle root and exact acyclic dependency list. A STORE signer first retains the complete body and validates its dependencies' retention certificates. The resulting $3f+1$ honest holders supply eventual retrieval. An arbitrary unbacked hash is structurally invalid, not an instruction to wait forever. Retrieval and complete-object certification are charged to recovery.

Ocean readers maintain a first-resolution map for semantic parents. In the complete ordered transcript, the first valid resolution for p certifies exactly child(R). Its certificate binds that record's position and the verified earlier prefix establishing that it is first. A portable proof that a later record merely appears in an anchor is insufficient. Later resolutions for p cannot authorize another child. Every Ocean authority reachable through a resolution's parent-proof chain must precede the resolution's own transcript position. A complete forward or cyclic Ocean dependency is invalid. Substituting a future Ocean proof for an otherwise equivalent FAST parent proof must not make an earlier record wait forever. Apply this precedence check recursively at every embedded Ocean certificate's own position, including through intervening FAST or FIN proof wrappers. A known false position/digest membership is invalid; only genuinely unavailable earlier evidence is Pending. Check a visible reference's position and instance bounds before waiting for its opening. An explicitly forward Ocean dependency is invalid immediately; retrieval must not turn that structural contradiction into an indefinite wait.

Validity and firstness depend on complete evidence and transcript order, never on the reader's current application head or certificate-arrival timing. An earlier record with an unavailable opening keeps the interpreter Pending; it cannot be skipped in favor of a later opened record. Structurally invalid complete records are deterministic no-ops. Every anchor advances the control transcript; only effective parent resolutions advance application authority.

7.4 Installing certified children

The selected old x retains its exact identity, even if Ocean supplies an additional proof for it. Install certified ancestors before descendants and append a semantic transition at most once. A resolution for x may appear in Ocean before a resolution for its ancestor p when it carries a FAST proof of x. A later resolution of p must certify that same x; it neither rewinds execution nor appends an extra transition after x. An honest observer gossips every newly verified finality proof and serves its complete dependencies, independently of application execution and parent closure.

There is no mode switch, global signing stop, checkpoint attestation or Heal. After a leaderless fallback z is certified, d_z is the next opportunity. The old domain d_p stays closed forever. Storage compaction must preserve an equivalent permanent exclusion; this draft does not supply such compaction.

7.5 Deterministic interpretation

The interpreter maintains a durable transcript cursor and a map $\mathsf{first}[d_p]$ initially empty for each parent. It processes each ordered record through the following steps:

1. Verify the ordered position and authenticated record opening. Retrieve certified dependencies. If an earlier required opening is unavailable, return Pending without moving the cursor past it.
2. Check the configuration, parent proof, recursively decreasing Ocean authority positions, exactly $4f+1$ distinct complete statuses, and complete semantic validity of the selected child. Reject a structurally invalid reference or a complete contradiction as a deterministic no-op.
3. If the valid record's parent already has an effective first resolution, advance the cursor without changing its child. Otherwise durably record this position and selected child as that parent's first resolution.
4. Retain the verified preceding transcript and record evidence needed to justify firstness, publish the resulting finality authority, and install any now-complete semantic ancestors before the child. Advance the durable cursor.

A later Pending record does not erase certificates produced from an earlier complete prefix. Conversely, a local fast head is not permission to skip an old-parent record before checking it. Application delivery can be idempotent while record interpretation remains complete and ordered.

7.6 Why preservation still needs agreement

For $f = 1$, a five-status set can contain three own PREPAREs for x, two for another child, and no complete four-PREPARE

U. Another valid five-status set can expose a different unfinished distribution. The protocol does not require these sets to select the same child unless an admissible fast final forces it. Every eligible set must preserve such a final; choosing among the remaining possibilities is Ocean's job. The signed reference tests exercise split-support cases and competing records rather than assuming a common local selection.

Firstness is also an explicit proof cost. A membership proof for record i cannot establish that no earlier valid record resolves the same parent. The baseline permits replay of the authenticated complete preceding transcript; a succinct replacement would need to prove equivalent interpretation. The current reference fixtures do not implement a portable native proof of this property. Section 11 records that boundary.

8 Native Ocean ordering service

Ocean is the continuously running ordering substrate. It accepts authenticated control vertices, exposes complete source cuts to child construction, and orders recovery advertisements. Its ordering output is a control transcript; the parent-resolution interpreter in Section 7 supplies the application authority. This separation prevents a generic anchor from becoming an independent writer of application successors.

Strong edges connect a vertex to the preceding round and supply the wave arguments below. Weak edges bring older accepted vertices into complete emitted cones; they do not replace the strong-parent quorum. Acceptance requires the entire certified strong-and-weak parent closure. Honest authoring follows the own-parent and positive FIFO weak-reference rules of Section 4.5. The resulting complete cuts can advance even with empty transaction contributions.

8.1 Uniform reliable broadcast

Each slot is (configuration, round, author). The sender signs its exact payload. An honest validator ECHOs at most one authenticated valid payload and READYs at most one digest, persisting both choices before release. Every ECHO holder retains and serves the complete payload. The total-f rules are:

1. Send READY after $E = 3f+1$ distinct ECHOs of one digest, or after $f+1$ distinct READYs of it. READY relay need not wait for payload retrieval.
2. Deliver by the *fast route* on the exact valid sender payload and $F = 4f+1$ distinct ECHOs of its digest.
3. Deliver by the *slow route* on that payload and $2f+1$ distinct READYs.

A portable delivery proof explicitly distinguishes the two routes and verifies the corresponding signature kind, count, configuration, slot and payload. Ocean additionally checks the complete accepted parent closure. A receiver does not treat a generic signature count or a bare payload hash as delivery.

Lemma 5 (Uniform two-route broadcast)**.** *At most one payload has a valid delivery proof in a slot. An honest sender's valid payload eventually delivers to every honest participant. If either kind of delivery proof can form, all honest participants eventually deliver that same payload, even if its first recipient fails immediately without forwarding its certificate.*

Proof. Let H contain the identities that never deviate or disappear in this execution; $|N \setminus H| \le f$, including every later failure. Any two ECHO seed sets intersect in at least $2E - n = f+1$ identities, including a member of H. Its one-choice ECHO prevents different seed values. An H-member's first READY for a value originates in a seed: amplification by $f+1$ earlier READYs requires an earlier H-member, so a finite causal descent reaches an ECHO seed. Consequently all READYs from H agree with the sole seed value.

A fast proof and an ECHO seed intersect in $F+E-n = 2f+1 > f$ identities; two fast proofs intersect in $2F-n = 3f+1 > f$. A slow proof has at least $f+1$ members of H and hence implies the unique seeded READY value. These facts exclude conflicts between every pair of delivery routes.

For fast totality, its F ECHOs include at least $F - f = 3f+1 = E$ members of H. Their independently multicast ECHOs eventually reach every member of H, which therefore seeds READY. They also retain the exact body. This argument requires neither the fast recipient nor any faulty certificate signer to cooperate again. For slow totality, at least $f+1$ members of H already sent READY. Their multicasts amplify READY at every member of H. The originating seed has at least $E - f = 2f+1$ retained honest payload holders, so all can retrieve the body. In either case, the $4f+1$ or more members of H supply more than the required $2f+1$ READYs, and slow delivery eventually follows. For an honest sender, all members of H receive its valid body and ECHO, so fast delivery eventually holds as well. Durable choices and retention preserve each premise on restart. □

Corollary 6 (Two-delay accepted-cut clock)**.** *If all honest authors send valid vertices at time s and multicast ECHOs on receipt, and the complete certified strong-and-weak parent closure of each SEND is already at every honest receiver or is pushed with that SEND within one delay, every honest vertex of that round is accepted at every honest receiver by $s+2\delta$. Byzantine silence does not change this bound.*

Proof. SEND arrives by $s+\delta$ and at least $4f+1$ honest ECHOs by $s+2\delta$. These form fast delivery proofs. Every receiver can validate and install the complete certified parent closure by the first-leg deadline. Thus there is no remaining parent admission dependency. Local possession by the author alone would not imply this bound. In particular, all honest authors then have the $4f+1$ prior-round parents needed to begin the next round. □

Assumption 7 (Coin and reveal history)**.** For each wave, all honest participants eventually recover the same elected author. Every contribution generated by an uncorrupted signer follows the pre-coin seal rule. For analysis, reveal a wave's ideal author at the first such contribution, after its durable seal is fixed. Order these events by execution order, with a fixed tie-break for simultaneous events; the order need not follow wave numbers. This is an online event, independent of whether that signer is corrupted later.

Let $\mathscr{H}_j^-$ be the pre-reveal history at event j, including its wave identity, its fixed seal, all prior revealed coin values, and the intervening protocol events and adversarial choices. These histories extend the preceding post-reveal histories. For every $a \in N$, the fresh ideal author C_j satisfies $\Pr(C_j = a \mid \mathscr{H}_j^-) = 1/n$. In particular this conditioning includes previous waves' successful or failed core hits. A contribution is not withheld from this analytical reveal merely because some observer cannot yet reconstruct the signature. A computational realization must justify an adaptive, finite-horizon replacement for this joint ideal-coin contract.

8.2 DAG and coin boundary

The Ocean instance begins with one deterministic round-zero seed vertex per configured author. These seeds are bound by the agreed configuration, require no broadcast, and are never emitted as control records. Round-one vertices select their strong parents from these seeds. Network admission rejects a purported replacement round-zero vertex. The last closed wave is initially zero and the anchor and emitted-record sequences are empty.

An accepted vertex has $\mathsf{Q} = 4f+1$ or more distinct-author exact strong parents from the preceding round, optional weak parents from earlier rounds, and complete parent closure. The accepted slot is globally unique. Honest authors produce vertices after enough parents arrive, including empty control bodies when needed. Neither unresolved coins nor transaction execution block later control rounds. Weak parents precede the immediately previous round; references to that previous round belong to the strong-parent set. Wave one starts at physical round one, and subsequent waves partition consecutive groups of W rounds.

Set $k = \lceil \log_2 \mathsf{Q} \rceil$ and wave length $W = k+2$. A wave's candidate round is its first round. Before producing an honest coin share, its author durably seals Q accepted final-round

endpoints and their complete ancestry. The coin domain binds configuration, Ocean genesis instance, and wave. Different shares may have different seals; a seal digest is not part of the common signed coin message. It is the local durable prerequisite to signing that message.

Lemma 8 (Pre-coin common core)**.** *Every valid seal contains at least* Q *first-round candidate vertices strongly reached by all its endpoints.*

Proof. Pair the endpoints. Every pair has a common exact parent because $2\mathsf{Q} > n$ and slots are globally unique. Replace each pair by such a parent; an unpaired singleton is replaced by any parent. Each representative is an ancestor of the endpoints it represents. Repeat for k predecessor layers. There is then one second-round representative reached by every endpoint. Its Q first-round parents are the required core. All objects are in the immutable finite seal before that protocol-following coin contribution is released. □

The coin elects an author. A direct witness for its first-round candidate consists of Q accepted final-round vertices strongly reaching that exact candidate. By broadcast totality, a sealed core candidate's witness becomes available to every honest observer even when the candidate author is faulty.

8.3 Backward commitment and closed-cone emission

Maintain a last closed wave and an ordered anchor sequence. To close a later directly supported wave, first recover every earlier unresolved coin through that wave. Begin with its elected candidate as current. Scan waves backward to the last closed wave. Select an earlier elected vertex exactly when it lies in current's complete strong cone, and make each selected vertex the new current. Emit this selected chain in forward order. For each anchor, emit its complete strong-and-weak cone minus already emitted IDs, sorted by (round, author, digest). Persist the whole state update atomically. Missing evidence is Pending; absence becomes an exclusion only inside the specified complete strong cone.

Theorem 9 (Ocean prefix agreement)**.** *Honest Ocean readers produce prefix-compatible anchor sequences and identical control deltas, including under delayed direct witnesses.*

Proof. Let L_a have a direct witness of Q final-round supporters. Every next-round vertex's Q parents intersect that witness in an exact accepted vertex. Thus every such vertex reaches L_a, and by induction so does every later-wave candidate. A backward scan from any later direct root must select L_a: until wave a, each current selected candidate has a later wave and reaches it. After meeting L_a, the scan has the same current vertex, complete cone, and earlier coin values as a scan rooted there. It reproduces its earlier sequence. Direct roots are ordered by wave, so induction gives compatible prefixes for all direct and indirectly selected anchors. Complete deterministic cone deltas then agree. A direct witness arriving after its wave was closed cannot contradict an exclusion, since propagation would have forced its candidate into that scan. □

Theorem 10 (Ocean progress)**.** *Under the communication and ideal-coin assumptions, Ocean commits infinitely many anchors almost surely. In the reveal order of Assumption 7, an eligible wave succeeds with conditional probability at least* Q/n*; the expected number of eligible reveals to a successful core hit is at most* n/Q*. These are wave-count bounds, not bounds on elapsed time to output.*

Proof. The $4f+1$ honest participants suffice for every parent quorum, broadcast, and seal, so rounds advance by induction without faulty cooperation. At reveal j, choose a deterministic Q-member common core K_j from its fixed seal. The set K_j is measurable in $\mathscr{H}_j^-$. Thus the event $E_j = \{C_j \in K_j\}$ satisfies $\Pr(E_j \mid \mathscr{H}_j^-) \geq \mathsf{Q}/n$ by Assumption 7. All previous E_i are measurable in that history. Iterated conditioning gives, for any finite continuation of t eligible reveals,

$$\Pr\left(\bigcap_{j=1}^{t} E_j^c\right) \leq (1-\mathsf{Q}/n)^t.$$

The same argument applies after any fixed finite prefix. It uses conditional freshness in the growing reveal history; marginally uniform but correlated wave outputs would not suffice. Summing the tail gives expected waiting count at most n/Q, and infinitely many core hits occur almost surely. Each hit has a direct witness that eventually delivers everywhere. Every finite set of earlier numerical-wave coins needed by its backward scan eventually completes, even if the reveal events occurred out of numerical order. Infinitely many distinct successful waves therefore give infinitely many committed anchors. FIFO weak-reference inclusion additionally orders each honest control record eventually. □

For a computational coin, distinguish two possible guarantees. A per-reveal conditional deviation of at most ε gives success probability at least $\mathsf{Q}/n - \varepsilon$ within its justified horizon. Alternatively, if the joint finite-horizon execution differs from the ideal experiment by event-probability at most ε, the no-hit bound is $(1-\mathsf{Q}/n)^t + \varepsilon$ over that horizon; an unconditional infinite-execution almost-sure claim does not follow from it. Threshold-BLS share validity, domain separation, and unique reconstruction alone do not prove this adaptive pseudorandomness condition. The implementation must also justify output conversion and its bias bound. With f adversarial shares, the first released honest share may already enable adversarial reconstruction, which is why its seal must precede that release. This clarification asserts no attack on the threshold signature. Ideal elected-author fixtures do not establish the computational coin contract.

A shorter total-f wave. Four rounds admit a separate counting argument. Any Q third-round vertices have at least Q^2 edges into at most n second-round vertices. Since

$$(4f+1)^2 - (5f+1)f = 11f^2 + 7f + 1 > 0,$$

some second-round vertex has more than $f = n - \mathsf{Q}$ third-round supporters. Every fourth-round parent quorum intersects those supporters; its first-round parents form a common core. The executable baseline uses the uniform logarithmic wave. This shorter variant must be implemented and checked separately before its latency is reported.

8.4 From ordered cones to recovery service

Within an emitted vertex, the configured canonical body encoding determines the order of its advertised records. Thus the anchor order, deterministic cone emission, and body ordering jointly define the record transcript consumed by the first-resolution interpreter. Readers retain both the authenticated ordering evidence and the complete openings needed to interpret that transcript.

Theorem 9 establishes a common order, while Theorem 10 and fair weak-reference inclusion establish eventual ordering of continually advertised honest records. Assumption 1 supplies their bodies and dependencies. A firstness proof additionally verifies the preceding record interpretation; an accepted-slot proof or inclusion under one anchor alone does not suffice. These are the concrete dependencies of Assumption 2. Ideal coin callbacks or an external ordered-record fixture implement only selected parts of this service contract.

9 Safety and composition across parents

The argument composes the data and evidence contracts with the two finality routes. Its dependencies are Lemma 3, complete child validity, durable per-parent signing, and Ocean's authenticated common transcript. The first two lemmas concern one parent; induction then connects all parents.

Lemma 11 (Unique prepare protection)**.** *For one* d_p*, all valid complete* U *certificates name the same child. Consequently FAST and FIN certificates for* d_p *cannot conflict.*

Proof. Two sets of $3f+1$ PREPAREs intersect in at least $f+1$ identities. An honest intersection member never chooses two children in d_p. FAST contains a complete U, and FIN

explicitly retains one. The argument is about semantic parent domains, irrespective of how their parent certificates are encoded. □

Lemma 12 (Preservation of possible late finality)**.** *If $FAST_p(x)$ or $FIN_p(x)$ can be assembled from an admissible execution, every valid complete closure set for d_p selects x.*

Proof. For FAST, its $4f+1$ signers and the $4f+1$ status authors share at least $2f+1$ honest identities. Each such signature must precede that author's closure, since later signing is forbidden. Its own PREPARE for x therefore appears in its status. This forces the second selection branch unless a complete U is present; uniqueness forces any such U to name x too.

For FIN, its $2f+1$ receipt signers include at least $f+1$ honest complete-U holders. At most f identities lie outside the status set, so one honest holder reports inside it. That receipt precedes closure, and its status exposes the retained complete $U_p(x)$. The first branch forces x. Neither argument assumes anyone assembled or received the final certificate before closure or before the Ocean decision. □

Theorem 13 (One certified child per parent)**.** *Under the stated signing, storage and ordering contracts, all valid finality certificates for one semantic parent certify the same immediate child.*

Proof. Fast/fast uniqueness follows above. Every effective Ocean resolution preserves any possible fast final by Lemma 12. Ocean/Ocean uniqueness follows from one common complete transcript and the first-resolution rule. Multiple status sets are not assumed to choose the same unfinalized value. □

Theorem 14 (Certified-prefix composition)**.** *Starting from one configured genesis, all certified semantic histories are prefix compatible across arbitrarily many fast and recovered children.*

Proof. Induct on height. Each valid child has a complete finality proof of its exact parent. By induction the parent at that height is unique. The one-child theorem gives the same next semantic transition for every certificate route. Because certificate representations are excluded from identities, the inductive successor domain is also unique. Exact deterministic frontier and delta verification then produces the same ordered block prefix. Complete payload retrieval and deterministic execution refine it to application state. □

Theorem 15 (Ancestor-closure independence)**.** *Let a valid certificate establish $p \to x$. Processing a closure or effective resolution for d_p does not revoke any signing permission in d_x or a certified descendant domain, nor change any such descendant's semantic interpretation. A delayed resolution of p cannot create another successor of x.*

Proof. Closure mutates only the durable entry indexed by d_p. A resolution of p must certify exactly x by the one-child theorem, and its effects end there. Domain d_x depends on the semantic identity of x, not the old domain's open/closed flag or the representation of its certificate. Descendant validation uses the same exact parent chain in either processing order. Induction extends this argument along the certified chain; application replay is idempotent. No protocol rule permits the ancestor resolution to append a second transition after x. □

This is independence of signing permissions and semantic outcomes. It is not a claim that shared CPU, storage, or network queues cannot delay fast messages. At an unresolved current head, a strict successor still needs its certified parent; the theorem does not remove that causal dependency.

9.1 From the certified prefix to application state

Corollary 16 (Deterministic application refinement)**.** *Let two honest consumers have the same configured genesis state and deterministic application transition function. When both have retrieved and consumed the same certified semantic prefix using the prescribed Sweep and duplicate rules, their ordered transaction sequence and resulting state agree.*

Proof. Theorem 14 gives the same parents and exact children at each position. For each step, canonical child validity fixes the frontier and ordered block delta. Authenticated payloads fix transaction positions and identities. Deterministic duplicate handling and the common application function therefore yield the same next state from equal preceding states. Induction proves the claim. Missing data can postpone a step but cannot authorize a different step. □

This is a conditional refinement argument, not an execution attestation embedded in FAST or FIN. A client seeking evidence that a state root was executed needs the application's additional proof or attestation contract.

9.2 Proof dependencies and what they rule out

These are symbolic arguments over the stated contracts. The signed tests exercise selected counterexamples and invariants, but no universal mechanized refinement of the current native runtime is claimed. The earlier formal artifacts concern earlier transition systems; Appendix D preserves their role without transferring their guarantees to this revision.

10 Progress, latency, and resource costs

Theorem 17 (Asynchronous progress)**.** *Assume finite honest timers eventually fire when a known certified parent lacks a learned final child; complete child construction, retention and execution eventually finish; and Assumption 2 holds. Every certified parent admitted by an honest participant eventually has a final child known to all honest observers, almost surely.*

Proof. If an honest participant learns a fast final, dissemination supplies the certified child. Otherwise an honest timeout supplies a closed status. All honest participants eventually learn the parent and either its child or time out themselves. They supply $4f+1$ statuses without faulty cooperation, also honoring old-parent requests after advancing. Each honest proposer can construct a valid resolution: retain the selected complete old child when forced, using it also as the normal-form fallback; otherwise obtain a later accepted cut and derive a valid fallback. Even an empty delta advances its certified height and cut. The complete honest record eventually orders; an earlier effective resolution can only have already certified a child of the same parent. Prefix agreement and eventual retrieval disseminate that result. Applying the argument after each finite prefix gives unbounded certified progress almost surely. □

Honest authors permanently include their prior accepted vertex and serve a fair arrival-order FIFO of older weak references. Thus every continually advertised certified data block belongs to all sufficiently late valid source cones. Every certified child strictly increases its source-cut round and includes the entire cut contribution. Unbounded certified progress therefore includes every such block eventually, even when some earlier fallbacks had empty deltas. This relies on the full-holder availability contract; the local source-only repair experiment is not its implementation.

Good-case fast service. An honest leader's fully evidenced proposal is ratified by $4f+1$ multicast PREPAREs within 2δ if its first leg delivers complete prerequisites, one-way delay is at most δ, and honest signers remain open for that proposal. Consecutive eligible honest leaders can overlap accepted-cut production to sustain 2δ decision intervals after warmup when each turn has a fresh strictly later cut and complete evidence ready within its first-leg service bound, and signers remain open. Fresh data may still require 2δ for DA, 2δ for an accepted advertised cut and 2δ for ratification. The redesign removes recovery ownership stages; it does not remove those fresh-data dependencies.

Boundary	Required fact	Failure excluded
Data to frontier	Compatible certified paths; complete cut	Conflicting coordinates or extraction from a partial cache
Frontier to child	Canonical identities and common validity	Proof aliases creating multiple signing domains
Fast to closure	Earlier own votes and retained complete U	A hidden FAST or FIN contradicting a recovery choice
Closure to Ocean	First valid record in a common complete prefix	Different local status pools choosing different outcomes
Parent to descendant	Exact-child authority and persistent exclusion	Reopened old parents or a recovery tail after a certified child
Order to execution	Complete payloads and deterministic replay	Equal anchors with different emitted transaction order

Table 3: The safety argument's interfaces. Each row is necessary even when the adjacent component is correct in isolation.

Timeouts are a real interference condition. Background Ocean progress or quick recovery links are not evidence that an honest proposal cannot still arrive. An aggressive timeout can close a parent before its delayed honest proposal obtains FAST. No unconditional honest-leader noninterference result is claimed. Timers start only after the exact parent is admitted, are finite, and use a persisted monotone backoff floor carried across parent changes; an empty fallback must not reset that floor. A useful fast-service theorem needs bounds on parent-learning skew, proposal/evidence readiness, processing and timer expiry. Logical ancestor independence alone supplies none of these numerical bounds.

One-resolution service bound. Measure from the first honest admission of p. For a failed opportunity define T_p to include parent-learning skew, the timer wait and collection of a complete honest closure set, E_p as construction, data retrieval and retention service, O_p as advertisement-to-ordered-resolution service, and I_p as certificate installation. With supplied bounds on these services,

$$R_p \le T_p + E_p + O_p + I_p.$$

There is one new Ocean selection stage. The era baseline instead orders a barrier, a checkpoint intent and a Heal, and additionally waits for executed checkpoint attestations and readmission. Those stages are absent from this algorithm, not assumed to have zero cost. Preparing a large record or an accepted source cut still costs service in E_p.

Corollary 18 (Structural disruption accounting)**.** *For B distinct parent domains whose stalled child opportunities invoke recovery, at most B effective first-resolution records and B immediate recovered children are needed. Those closures revoke no signing rights in other certified-parent domains. Under per-opportunity service envelopes, a conservative additional-service charge is $\sum_{p=1}^{B} R_p$.*

This counts effective decisions, not all competing submitted records, network messages or CPU work. The same faulty identity can cause unboundedly many failed opportunities, so B is not bounded by f. Transaction throughput also needs a nonempty-workload inclusion bound and cannot be inferred from the number of certified empty children.

For native Ocean let $Q = 4f+1$ and $W = \lceil \log_2 Q \rceil + 2$. If an external service envelope bounds common-frontier catchup and advertisement by A, permanent ancestry inclusion by d accepted rounds, each complete round by ρ, and the required coins, earlier transcript and interpretation by K, a conservative m-fresh-wave budget is

$$O_p(m) = A + [d + 1 + (m+1)W]\rho + K.$$

Catchup must pass already-built candidates, endpoints and seals before trials are counted; eligibility is fixed before each ideal coin reveal. Then the no-core-hit probability is at most $(f/n)^m$. A union bound over B failed opportunities gives at most $B(f/n)^m$, plus service-envelope and computational-coin errors. Eventual delivery alone provides no wall-clock values for these parameters.

10.1 Data inclusion and execution endpoints

Lemma 4 puts each continually advertised certified block in every sufficiently late source cut. By Theorem 17, certified children continue; their integer source-cut rounds strictly increase. Eventually a child uses a cut containing that block. Mandatory cut extraction includes it unless its producer path has already been included. This proves the qualified data-inclusion claim under the complete construction's contracts.

The final step to executed application output requires successful retrieval, finite deterministic application work, and fair delivery. A client submission, a DA certificate, an accepted cut, learned FAST/FIN, an Ocean resolution, and an executed transaction are distinct timestamps. In particular, counting empty certified children proves neither positive transaction goodput nor bounded client latency.

10.2 Message, verification, and retention costs

The structural recovery bound does not bound all communication. To expose the remaining costs, let b_x be the size of a complete child proposal including its required evidence, σ the size of a signature, and e_p the size of a complete resolution with its dependencies. These sizes are parameters: retained history and proof ancestry need not remain bounded.

With direct unicast fanout, sending one proposal costs $O(nb_x)$ bytes. One multicast PREPARE per validator produces $O(n^2)$ transmissions of bounded vote metadata and signatures; receipts can add another such exchange. A complete U carries $O(n)$ signatures. A receipt sent with its complete U can therefore make naive receipt dissemination cost $O(n^3\sigma)$ bytes, apart from proposal bodies. Caching and content addressing can avoid repeatedly moving an identical artifact, but do not remove its retrieval and validation obligations.

A status set has $O(n)$ authors and may repeat a complete $O(n)$-signature U inside each status. Its naive signature material can thus be $O(n^2\sigma)$, in addition to child and ancestry evidence. Deduplicating identical bodies does not permit omitting the association between a receipt issuer and the complete protection it retained. The cost of advertising, STORE-certifying, retrieving, and validating a resolution must be charged in terms of e_p.

Each native reliable-broadcast slot uses all-to-all ECHO and potentially READY traffic, giving $O(n^2)$ messages for its basic exchanges. With n authors per round this is $O(n^3)$ control messages per round before retransmissions, dependency fetches, and competing records. Parent certificates and nested evidence can dominate the bytes even when message counts are fixed. These are direct-construction costs, not an optimality lower bound and not amortized payload-throughput claims.

Persistent state includes producer payloads, accepted DAG ancestry, seals, outboxes, proofs, and a closure entry for each old parent. Verification and portable firstness replay can grow with this retained history. This manuscript does not supply a bounded-storage checkpoint replacement theorem. Finite component limits in an executable artifact are capacity limits, not a proof that a long-running deployment can discard the corresponding evidence.

11 Implementation, restart, and blockchain integration

11.1 Three distinct implementation boundaries

The artifact repository [1] contains several protocol generations. The parent-local source baseline used here is commit `2812f22`; the manuscript expansion changes exposition, not

those executable protocol files. The distinction between a current reference model and an earlier native engine is material to every implementation claim.

Signed parent-local kernel. `spec/experiments/parent_local.py` verifies actual Ed25519 leader proposals, PREPAREs, receipts, and closed statuses. It models permanent per-parent restrictions, complete retained prepare protection, exact semantic identity, finality proofs, and deterministic resolution interpretation. Its child-validity callback abstracts source cuts, frontiers, DA, and application semantics. Its journal provides atomic acknowledged memory replacement and restoration, not disk fsync or rollback protection.

Ocean-model adapter. `spec/experiments/parent_local_ocean.py` connects parent-local recovery advertisements to the accepted-DAG and closed-cone ordering in `spec/v4/ocean.py`. Its tests exercise ordering-dependent selection, missing earlier records, and descendants whose resolution precedes an ancestor's resolution. Broadcast and coin verification callbacks, opening availability, complete STORE certification, and portable native firstness remain explicit boundaries. The separate kernel CLI uses a trusted ordered-record fixture; it must not be mistaken for this adapter or a distributed deployment.

Native components. The Rust modules in `rohu-paplet/src/weather_v4/` implement the earlier era-based revision: typed evidence, River, reliable broadcast, Ocean, threshold-BLS coin primitives and signing gates, restore validation, and snapshot storage. `NativeControl` connects durable RBC authoring, accepted-round construction, typed proof retrieval, and gated coin progress in a finite event-driven stream. It does not supply a complete deployed River/DA/STORE/recovery scheduler. Its existing Weather/Heal ownership is not reinterpreted as parent-local closure. The signed parent map, current resolution verifier, and validator-wide checkpoint orchestration need native integration.

11.2 Crash recovery as a protocol obligation

Restoring a signer means restoring its promises. The parent map must retain chosen PREPAREs, complete U protection, receipts, immutable closed statuses, outboxes, and timer-backoff state. The data layer must retain certified payloads and its selected producer branches. Ocean needs accepted exact vertices, complete ancestry, fixed seals, coin authority, emitted transcript, and cursor. The interpreter needs first-resolution authority or enough authenticated history to recompute it. The consumer needs its exact delivery position and either application state or sufficient data for deterministic replay.

The earlier native Unix `SnapshotStore` stages an identity-bound, checksummed snapshot, syncs the temporary file, atomically renames it, and syncs the containing directory before acknowledging success. It holds an advisory lock for its lifetime and halts on ambiguous I/O failure. Restore validates the installed snapshot rather than treating an unfinished temporary file as a checkpoint. These are concrete mechanisms for the storage contract, conditional on filesystem behavior. A checksum and a generation number in the same replaceable directory do not detect malicious rollback.

The publication argument is straightforward but indispensable. Before durable acknowledgement, no output depending solely on the staged state may escape. After acknowledgement, a restart must recover that state or a later one, even if its output was never sent. Retaining an unused vote reservation is safe; forgetting an already published choice is not. A parent-local integration must apply this argument to one coherent validator state, rather than combine individually correct stores whose checkpoints can disagree.

The native control profile has finite capacity: 16,384 RBC slots, a 256 MiB complete snapshot ceiling, and a 256-item body queue in the recorded revision. Capacity exhaustion is an explicit error. It is not a liveness-preserving garbage collector for arbitrarily long executions. Long-term key/share custody, network authoring and retrieval scheduling, atomic external effects, and safe checkpoint replacement remain separate obligations.

11.3 The blockchain-facing contract

A blockchain adapter supplies a trusted configuration and genesis, deterministic validity rules, transaction-data admission, and application execution. Consensus returns exact certified children with ordered block deltas and their verifiable parent chain. This gives a linear interface even though the control structure is a DAG. A consumer never chooses its own order from the latest visible DAG or from local proof-arrival order.

Interface event	Required interpretation
Submit data	Admission request; not evidence of DA or finality
Learn final child	Verify exact parent, child validity, and FAST/FIN or first-resolution authority; record the certified semantic identity
Deliver ordered delta	Install predecessors first; preserve Sweep order; record delivery position durably and make duplicate delivery idempotent
Execute	Retrieve required payloads and apply the configured deterministic transition; distinguish this completion from learned finality
Restore or synchronize	Verify configuration and proof history; recover signing promises and application cursor before resuming corresponding effects

Table 4: Consensus-to-application contract. No interface event implicitly asserts the completion of a later one.

A semantic child height counts transitions, while an application delivery position can count blocks or transactions inside their deltas. An empty child advances the former without creating application work. A chain adapter must define that mapping and its genesis offset explicitly. The historical Commonware/ETO integration used a committed-position stream to bridge a DAG to a linear chain. It is useful interface context, not evidence that the new parent-local certificate types have been deployed in that adapter.

A late observer can replay a complete certified history from genesis under the baseline retention contract. An optimized state-sync checkpoint must bind the semantic identity, frontier, log commitment, application cursor, and retained proof obligations, and must have an independently justified trust or certificate rule. The older Heal-based state-transfer and reconfiguration mechanisms do not automatically provide this rule for the current fixed-committee construction. In particular, importing an executed state snapshot does not grant permission to forget the signer's closed old domains.

11.4 Claim-to-artifact ledger

Table 5 maps the main claims to their evidence and assumptions.

12 Evaluation and falsification

The evaluation asks whether the implemented reference transitions respect the interfaces used by the proofs, and what the separate dissemination experiments say about transport cost. It does not report an integrated native parent-local WAN benchmark. We distinguish the September 8 recorded results, their September 10 reference rerun, earlier protocol deployments, and experiments still required to measure the complete construction.

12.1 Parent-local kernel and ordering adapter

The combined signed kernel/adapter suite contains 32 tests: 21 kernel tests and 11 Ocean-adapter tests. The September 8 record reports 7.293 seconds; the September 10 manuscript validation reran the same suite successfully in 7.107 seconds with Python 3.12.3 and cryptography 41.0.7. These durations

Claim	Argument or evidence	Boundary
Compatible available data	Lemma 3; native DA component evidence	Full-holder retention and service are required; the parent-local callback abstracts this layer
One child and one prefix	Lemmas 11–12; Theorems 13–14; signed kernel tests	Symbolic proof under stated contracts, not a universal mechanized runtime proof
Local ancestor closure	Theorem 15; concurrent-descendant tests	Signing-rights independence, not isolation of shared resources or timers
Native Ocean order	Theorem 9; Ocean reader and adapter tests	Complete accepted ancestry and authenticated ordering evidence
Almost-sure progress	Theorems 10 and 17	Ideal conditional coin, fair service, complete evidence; not measured latency
Data inclusion	Lemma 4; increasing mandatory cuts	Honest advertisement and full-holder availability; abstract in current kernel
Durable signing	Publication contract; interrupted memory-journal tests; earlier native store	Parent-local network-wide disk recovery is not implemented
Transport tradeoff	Recorded byte counts, modeled links, TCP/fsync experiments	Separate component; no consensus-finality or deployment benchmark
Application state agreement	Corollary 16	Common deterministic execution and retrieval; no execution attestation in FAST/FIN

Table 5: Claim-to-artifact ledger and the boundary of each supporting result.

are test-run times, not consensus latency. The executable sources are unchanged from the identified baseline.

The tests include 28,591 finite quorum-intersection checks at $f = 1, 2$, actual signed late FAST and FIN assembly, proof aliases, competing resolutions, permanent closure, and interrupted journal updates. Adapter cases exercise the actual Ocean model's accepted-DAG and closed-cone order while retaining the explicit broadcast/coin and opening-availability boundaries.

The CLI uses a trusted ordered-record fixture, whereas the adapter suite uses Ocean-model ordering. The fixed-leader cases deliberately override rotation to stress repeated recovery. The eleven-validator runs use an $f = 2$ quorum profile but only one actual silent identity; they are not measurements under two concurrent faults. The thirty restorations check signer journals against replayed authority, not complete networked-validator checkpoints.

12.2 Falsifying schedules and regression oracles

The ordered-log oracle must compare exact semantic child identity and ordered delta at each position, not only the latest height or selected anchor. A delivery-level test should additionally compare frontier/log commitments and an order-sensitive digest of distinct transaction contents. A later matching suffix cannot repair a conflicting prefix already published append-only.

This requirement follows from a recorded failure in the earlier full-system evaluation, discussed in Appendix D. Agreement on anchors survived while a lagging observer emitted them in a different order. The current construction's complete-cut and exact-child rules address the corresponding interface obligation; the old incident is not a claim that the same defect exists in this reference kernel.

12.3 Separate transport and extension evidence

The local transport experiment compares direct dissemination with a coded one-relay construction while retaining complete native DA custody. At six validators and 64 KiB application payloads, measured protocol producer bytes fall from 1,171,762 to 472,437, a 59.7% reduction; aggregate bytes increase from 1,173,770 to 1,418,305, or 20.8%. This is an upload/aggregate-traffic tradeoff, not a universal speedup. In the sequential local TCP harness the median whole-experiment time rises from 63.96 to 94.38 ms with memory-acknowledged snapshots, and from 106.38 to 135.30 ms with atomic fsync snapshots.

The deterministic link model can favor coding when serialization dominates, but loses for some small payloads and fast links because of the additional hop. Its endpoint is a DA certificate at a specified validator, not transaction finality. Repair is served by the original source; the experiment does not implement surviving-holder retrieval after source loss or uniform broadcast. Appendix C preserves the full environment, byte tables, fault cases, timing definitions, and limitations.

The separate signed extension model explored 5,554 bounded worlds and 1,112,888 fast/recovery prefix checks without a guarded-prefix violation. Naive lane merging produced incompatible order in 388 worlds, and complete statuses still permitted competing unfinished recovery continuations in 2,430 worlds. These are counterexamples to deliberately weakened interfaces, not attacks on published protocols. The model has fixed proposals, two lanes, genesis bases, and restricted Byzantine alternatives.

Variable-prefix voting and exact-child finality have different output types. Two compatible transaction prefixes of different lengths cannot simply become two immediate children of the same parent. Integrating proposal-relative extensions therefore needs a canonical prefix/anchor refinement rule and a matching preservation theorem. No current execution connects that model, coded delivery, and native Ocean recovery into one deployed protocol.

12.4 A complete-system evaluation contract

The unmeasured target is an execution through actual dissemination, DA certification, accepted-cut formation, exact child construction, FAST/FIN or native ordered resolution, application delivery, and restart with the same prefix. The evaluation must replace the corresponding trusted fixtures or name them as remaining assumptions. Merely increasing the number of CLI heights does not close those interfaces.

Measure admission-to-DA, cut readiness, proposal-to-learned-finality, external observer proof delivery, and executed application completion separately. Report unfinished requests as well as completed ones. Compare fully prepared proposals with fresh submitted data; inject silent/equivocating leaders, hidden certificates, slow retrieval, missing earlier records, and crashes around publication. A sustained-recovery run must measure transaction inclusion and backlog as well as empty-child progress.

For network evaluation, pin the source and binary, committee and actual fault budget, topology, payload/workload, cryptographic mode, persistence, service limits, and load. Report latency distributions, goodput, queue growth, retained bytes, and CPU/network use across repeated runs. Use distinct payloads and a per-position consistency oracle. Baseline comparisons need matched finality endpoints and compatible fault assumptions. These are stated experimental obligations; the tables above do not substitute for their results.

13 Related work and positioning

13.1 DAG dissemination and ordering

DAG-Rider separates reliable dissemination of a structured DAG from local ordering of its vertices [11]. Bullshark uses DAG interpretation to combine asynchronous ordering with a favorable synchronous path [12]. Rohu-Paplet's native Ocean likewise distinguishes accepted data from its ordered interpretation. Its contribution here is the specified interface between

Validators	Schedule	Children	FAST	Resolved	Restores
6	Rotation; one silent identity	18	15	3	5
11	Rotation; one silent identity	22	20	2	10
6	Fixed silent leader, synthetic	12	0	12	5
11	Fixed silent leader, synthetic	12	0	12	10

Table 6: Recorded signed CLI schedules from September 8. All four commands also completed successfully during the September 10 manuscript validation.

Adversarial or weakened schedule	Rule exercised
Reopen the same parent after closure	Unsafe reset admits a signed fork; permanent exclusion prevents a second old attempt
Encode two proofs for the same parent	Semantic identity shares one lock; certificate bytes cannot create separate domains
Withhold FAST or FIN until after closure	Own-PREPARE reports and retained complete U preserve the hidden child
Resolve an ancestor after a fast descendant	Recovery certifies the original child without appending a conflicting tail
Deliver a later resolution before an earlier opening	Missing evidence remains Pending; local arrival order cannot choose the winner
Insert a forward Ocean proof or known wrong membership	Complete contradiction is Invalid, preventing permanent poisoning of the transcript
Fail persistence before signing or release	Reserved choices survive; unacknowledged new outputs cannot escape
Close early during a delayed honest proposal	Locality alone does not imply unconditional fast-path noninterference

Table 7: Selected regression families. A passing finite schedule corroborates its stated boundary and does not establish all-schedule correctness.

that transcript and exact parent-local finality, with complete evidence and first-resolution authority. The presence of a DAG, a fast route, and a recovery route is not itself a novelty claim.

Optimistic reliable-broadcast designs provide further context for fast and slow delivery routes [8]. This paper gives its own threshold and totality argument for the stated total-f profile. It does not import another broadcast protocol's guarantees while changing its custody or fault assumptions.

13.2 Preserving fast decisions across recovery

Abstract explicitly composes abortable state-machine replication instances through commit, abort, and initialization histories [10]. Earlier commits must survive in the successor's admissible history. Consequently a general preservation interface is established prior art; the relevant object here is the exact immediate-child interface and its interaction with already certified descendants.

Ipotane connects optimistic certification to successor-indexed asynchronous agreement, with a pessimistic outcome ending the optimistic epoch [14]. Icarus uses parallel certified chains and an author/epoch alignment procedure: participants exchange their latest certificates and agree the previous path's final height before switching [9]. Both are relevant to the distinction between preserving an earlier result and agreeing on what comes next.

Ambulance replaces timeout-based racing with protocol-driven races and distinguishes evidence excluding leader commitment from stronger evidence excluding its lock [15]. Its recovery must still ensure a common choice among surviving possibilities. The analogous distinction in Rohu-Paplet is between a closure set's preservation guarantee and Ocean's first-resolution agreement. Rohu-Paplet still uses finite timers to close a parent, so the locality theorem must not be presented as Ambulance's slowdown-detection claim.

13.3 Low-delay finality and independent work

FinWhale combines fast-path evidence with slow-path DAG decisions under its partially synchronous fault model [4]. Fast Tendermint obtains good-case two-step decisions with a larger committee while retaining its leader-rotation structure [5]. Orcaella studies mixed Byzantine and crash-fault contracts and different finality paths [6]. A comparison must align the observer endpoint, preexisting evidence, network assumptions, and unavailable-identity budget; the phrase "two delays" alone does not align them.

Cadence separates slots into independent consensus instances and orchestrates their overlap [7]. Rohu-Paplet's certified-child chain has a different dependency: the next domain requires the exact certified parent. Ancestor closure does not revoke a descendant domain once it exists, but this does not make unresolved predecessor and successor decisions independent.

13.4 Dissemination and proposal-relative extensions

Alpenglow separates voting/finalization from Rotor's coded dissemination mechanism [13]. Our local coded-relay experiment is inspired by that transport direction, while retaining equal-weight identities and full-holder native DA custody. It is not an implementation of Rotor's complete deployment model, and its local timings are not Alpenglow finality measurements.

Multimmit allows proposal-relative per-chain votes to support data beyond the leader's original proposal [3]. Its dissemination-to-order endpoint differs from River's already-evidenced proposal endpoint. Our optional complete vector evidence fixes one exact child before River votes; the separate variable-prefix experiment does not yet compose with exact-child recovery. This interface difference must be resolved before transplanting a latency or inclusion result between the constructions.

13.5 Fast-path resilience and the closest DAG designs

Outside DAGs, the resilience of a two-step fast path is settled. Kuznetsov, Tonkikh and Zhang (KTZ) prove that no f-resilient t-two-step consensus protocol exists for $3f+2t-2$ processes. They reach $3f+2t-1$ by having the new leader wait for $n-f$ votes not including the equivocator's [16]. Abraham et al. show that two-round partially synchronous Byzantine broadcast is possible if and only if $n \geq 5f-1$ [17]. Banyan matches $n \geq 3f+2p-1$ without a view-change protocol [23]. Kudzu runs at $3f+2p+1$ and names the $3f+2p-1$ adaptation as future work [24]. All four assume partial synchrony. Appendix B borrows KTZ's exclusion step unchanged. Its selector-class inequality recovers the published endpoints and adds only the one-shot point at $n=5f$.

Several protocols read a fast path off an uncertified DAG. Mysticeti-FPC does so at $3f+1$ under partial synchrony, and unlocks equivocated objects only at epoch end [20]. BB-Core builds on an uncertified DAG at $5f+1$ and commits in two message delays under partial synchrony. Its variant BB-Core-Async uses a threshold common coin, three-round waves, and $4f+1$ and $2f+1$ certificates with a propagation lemma and an edge-counting core [21]. The $5f+1$ uncertified Ocean study summarized in Appendix B.3 repeats that structure, and we claim no novelty for it. FinWhale reaches $n=3f+2p-1$ on Mysticeti's DAG. After a detected equivocation, a validator waits for $n-f$ blocks not including the Byzantine leader's [4]. Hydrozoan makes a vote an edge, so it needs no new message types or view changes. Optimal-Hydrozoan reaches $3f+c+2p-1$ by treating an equivo-

cating leader as a detected Byzantine validator. Hydrozoan's proofs are checked in Lean 4 [22]. Alpenglow resolves each slot with no global view change, using author-blind counts at the $5f+1$ point [13]. FinWhale, Hydrozoan and Alpenglow assume partial synchrony.

Hydrozoan is concurrent and independent work. Its first version appeared on September 22, 2026, after the first private draft of this paper. With no crash budget and $p=f$, its thresholds at $5f+1$ equal River's, threshold for threshold: q_{fast} is FAST $(4f+1)$, q_{cert} is U $(3f+1)$, q_{direct} is the receipt quorum $(2f+1)$, q is the status set $(4f+1)$, and q_{weak} is the protection threshold $(2f+1)$. Its graded indirect rule checks for a certificate first, as protect does. The difference is that our votes are signed messages, while Hydrozoan's are DAG edges.

Asynchronous DAG ordering supplies the rest. Cordial Miners approves a block that is observed without an equivocating block, stops pointing to miners caught equivocating, and uses five-round waves under asynchrony [18]. Mahi-Mahi runs five-round waves on an uncertified DAG at $3f+1$ under asynchrony, with coin shares in blocks and no fast path [19]. The $5f-1$ Ocean of Appendix B is Cordial Miners' asynchronous rule at a different n. Carrying coin shares in vertices follows DAG-Rider [11].

Positioning. Rohu-Paplet is a per-parent River beside an asynchronous DAG that only orders recovery outcomes. FinWhale and Hydrozoan put the fast path and its recovery inside the DAG, under partial synchrony. River instead runs a leader per parent, decoupled from DAG rounds, with its own signed PREPAREs, receipts, statuses and resolution records. Ocean orders only the resolutions. Mysticeti-FPC unlocks equivocated objects per epoch; here each stalled parent is resolved once Ocean orders a valid resolution. The per-slot setting itself is not new: Banyan, Alpenglow and Hydrozoan also recover per slot.

The gap we claim. We looked for a consensus protocol with a 2δ fast path at $n=5f-1$, or generally at $3f+2p-1$ with $p\geq 1$, that stays live under full asynchrony. We found none. Every such protocol we found assumes partial synchrony for liveness: KTZ, Abraham et al., Banyan, FinWhale, Optimal-Hydrozoan and the Kudzu sketch. These are the near misses:

- Kursawe's and Bosco's protocols, as KTZ describe them, are asynchronous, but fast only with no failures or with unanimous inputs [16].
- BB-Core-Async is asynchronous at $5f+1$, with three-round waves [21].
- Ambulance is asynchronous at $3f+1$, with three message delays [15].
- The asynchronous reliable broadcast of Abraham et al. takes two rounds at $3f+1$. It is broadcast, which may never commit when the broadcaster is Byzantine [17].
- DAG-Rider, Bullshark, Cordial Miners and Mahi-Mahi are asynchronous, without a 2δ fast path [11, 12, 18, 19].

Some of the partially synchronous designs, Hydrozoan for one, keep safety under asynchrony, so the gap concerns liveness only. This is a list of what we found, not a proof that none exists. The asynchronous liveness argument of the main text covers $n=5f+1$ only. Appendix B closes the gap on paper: it gives hand-proved and script-checked lemmas for recovery, the Ocean, storage and availability at $n=5f-1$. None of them is mechanized or peer reviewed.

13.6 The comparison established by this revision

The concrete structural comparison is against the archived era baseline of Appendix D. That design uses an ordered barrier, checkpoint intent, executed checkpoint attestations, and Heal/readmission. The parent-local construction replaces those ownership stages with one effective resolution for one exact parent, preserving an already-certified child without a new recovery tail. This comparison establishes what stages the algorithm removes. It does not establish lower total latency under all service conditions, optimal communication, or a general advantage over the external protocols above. Those questions require matched implementations and a separating theorem or measurement.

14 Limitations and conclusion

The construction specifies a fixed-committee consensus path from certified producer data to one verifiable ordered prefix. Its main preservation result depends on exact semantic identity, complete retained prepare evidence, permanent parent-local closure, and agreement on the first effective resolution. Native Ocean supplies a concrete ordering construction under its broadcast, ancestry, fairness, and coin assumptions. Application state agreement follows only with the stated deterministic execution and retrieval contract.

Several boundaries remain substantive. The reference kernel abstracts data validity and uses memory-journal persistence; its CLI and adapter expose different modeled ordering boundaries. The native Rust components implement the earlier era revision and require parent-local integration. A portable native firstness proof, deployed retention/retrieval scheduler, validator-wide restart, bounded-storage replacement, and committee reconfiguration are not established by the current artifact. The ideal-coin progress argument needs a computational realization with the stated conditional-history guarantee. Variable-prefix voting requires a separate refinement interface.

These limitations delimit the results rather than changing their endpoints. The paper establishes, under explicit contracts, how possible late finality can survive one-parent recovery while certified descendants retain their signing rights. The existing signed schedules test the critical boundaries, and the transport measurements expose a real upload tradeoff. A complete native deployment and matched end-to-end measurements are the next evidence needed to turn this specified composition into a demonstrated system-performance claim.

A A scoped necessity result for receipt preservation

The honest-intersection inequality has a matching indistinguishability witness for a specific information interface. This is not a universal cost lower bound for consensus composition. Fix one certified predecessor p, prepare threshold $U=3f+1$, and $Q=4f+1$ complete retired statuses. A *one-shot status-only* selector must choose a successor from those statuses, without another old-side exchange. Returning p or waiting remains safe but does not meet this task.

Theorem 19 (Receipt threshold for this interface)**.** *In the stated interface, replacing the FIN threshold $2f+1$ by $2f$ permits identical complete recovery evidence in executions with incompatible hidden finals. No one-shot status-only selector can safely choose a successor in both. The $2f+1$ threshold, full-U retention and truthful retirement statuses suffice for the receipt-preservation component of the parent-local kernel.*

Proof. Partition the committee into Byzantine B of size f, honest sets A,C each of size $f+1$, honest D of size $f-1$, and honest T of size f. The reporters are $S=B\cup A\cup C\cup D$, of size $4f+1$. A Byzantine designated leader signs two distinct valid successors x,y of p in both executions. In both, A prepares x, C prepares y, and D prepares neither. In execution E_x, T prepares x, so $B\cup A\cup T$ supplies U_x and $B\cup T$ supplies $2f$ complete-U receipts. In execution E_y, T prepares y and $B\cup C\cup T$ supplies U_y and the analogous receipts. No honest identity equivocates within either execution.

Delay cross-group PREPAREs and every full U, receipt and T vote to S until after its finite recovery decision. Honest reporters freeze and report their identical own choices without U protection; Byzantine reporters sign the same empty hiding statuses. All complete signed status bytes agree in E_x,E_y, while the hidden weakened finals conflict. Releasing delayed messages afterward satisfies eventual delivery. A selector seeing only these bytes cannot choose a successor compatible with both; coupling any independent new-side coins

gives the same conclusion for randomized selection with zero-error safety.

The semantic pair is realizable with valid FCs: from a genesis frontier, one accepted round-one cut advertises certified producer block A_1 and another advertises certified producer block B_1. Their closed cuts, certified paths and distinct first-block deltas are individually valid; no conflicting DA branches or differing extraction from one identical cut are required.

At $2f+1$ receipts, at least $f+1$ honest holders exist and only f identities lie outside S. An honest holder inside S must have retained U before its freeze, since later receipts are forbidden. Its truthful complete status exposes that U. Lemma 12 then applies, including late assembly. □

The same attack applies to smaller receipt thresholds by taking subsets. This result concerns the specified receipt/retention interface. It does not establish lower bounds on rounds, bytes, or storage for other interfaces. Pre-distributed complete certificates, other voting locks, or another exchange change the interface. Moreover, preserving old finality does not ensure agreement between two new recovery choices; the first-resolution Ocean rule supplies that agreement here. Actual-signature witnesses at $f = 1,2,3,4$ accompany the symbolic construction in `spec/v4/test_transfer_bound.py`. The signed witness tests the shared one-parent receipt interface; it does not by itself validate the redesigned protocol or establish novelty.

B Optimal resilience and an uncertified Ocean

The main text fixes $n = 5f+1$. This appendix asks how small the committee can be while recovery still preserves every possible late fast final, and whether Ocean can follow it there. Both mechanisms we use are borrowed. The resilience $n = 5f-1$ and the step that excludes an equivocating leader are due to Kuznetsov, Tonkikh and Zhang (KTZ) [16]. The Ocean commit rule below is the asynchronous rule of Cordial Miners [18]. The new material is narrower: a three-class analysis of the recovery selector inside our interface, and the composition of parent-local recovery with that commit rule at $n = 5f-1$. Nothing here changes the $5f+1$ design of the main text. Section B.6 states what is proved, what is only checked, and what is open.

KTZ prove that no f-resilient t-two-step consensus protocol exists for $3f+2t-2$ processes, in their partially synchronous model [16]. With $t = f$ this makes $5f-1$ optimal, in that model, for a fast path that tolerates f faults. We did not re-verify that proof. Our own lower bounds hold only inside the recovery interface below. They are not bounds on every protocol.

B.1 The recovery interface

Fix one parent p, a committee of n identities and a total fault budget f. The designated leader $l = (h-1) \bmod n$ is also a voter. Write q_F, q_U, q_R and q_S for the FAST, U, receipt and status thresholds. $\mathsf{FAST}(v)$ is q_F PREPAREs for v, $U(v)$ is q_U PREPAREs, and $\mathsf{FIN}(v)$ is $U(v)$ plus q_R receipts from holders of a complete $U(v)$. A PREPARE is valid only together with the leader-signed proposal it answers. Valid PREPAREs for two children therefore prove l faulty. Statuses are as in Section 7: fixed at closure, and carrying the author's own PREPARE evidence and every complete U it retained.

A *selector* reads a set S of q_S closed statuses from distinct authors, once, with no other old-side exchange. It outputs a child or $\perp$. It is *correct* if it outputs v whenever $\mathsf{FAST}(v)$ or $\mathsf{FIN}(v)$ can still be assembled in some admissible execution that produced S. Late PREPAREs and receipts from open outsiders and faulty identities count. A *witness* is a pair of admissible executions with byte-identical S in which different children are possibly final. A correct selector exists exactly when no witness exists. Fast-path safety needs $2q_F$, $2q_U$ and q_F+q_U all at least $n+f+1$. Liveness needs every threshold at most $n-f$.

We compare three selector classes.

Author-blind.
A function of the multiset of status contents. The rule of Section 7 is in this class.

One-shot.
Any function of the bytes of S. In particular, it may read the leader's identity from the leader-signed proposal inside each PREPARE.

Adaptive.
The collector waits until it holds q_S statuses from authors that the set does not prove faulty. Only the leader can be proven faulty in this way. The set therefore holds q_S+1 statuses when it contains the leader's status and exposes the leader.

Write $o = n-q_S$ and O for the identities outside S, and let B be the fault set of an execution. Let $r(v)$ count the statuses in S whose own PREPARE names v, and $r'(v)$ the same count without l's status. The bracket $[l \notin S]$ is 1 if $l \notin S$ and 0 otherwise.

Lemma 20 (Counting). *Let S carry no complete U, and let q PREPAREs for v be formable in an execution consistent with S. Then S holds at least $q-o-|B\cap S|$ honest reports of v. If S names two children, then $l \in B$ and $r'(v) \geq q+q_S-n-f+[l \notin S]$. If $\mathsf{FIN}(v)$ forms, every receipt signer lies in $O \cup (B\cap S)$, so $q_R \leq o+|B\cap S|$.*

Proof. An honest quorum member inside S signed before its closure, so its status reports v. At most o members lie outside S, and $|B\cap S|$ are faulty insiders. If l is proven faulty and lies outside S, at most $f-1$ faulty identities are inside. An honest insider retained no U, because S carries none, so it signed no receipt. □

Theorem 21 (Selector-class inequality). *In this interface, a selector of a given class can preserve every possible late FAST if and only if*

$$2q_F+q_S \;\geq\; 2n+2f+c,$$

where $c=+1$ for author-blind, $c=0$ for one-shot and $c=-1$ for adaptive selectors. The liveness caps $q_F, q_S \leq n-f$ then require $n \geq 5f+c$: $5f+1$, $5f$ and $5f-1$ respectively.

Proof. For one-shot selectors, Theorem 22 gives a witness below the line and Theorem 23 gives a selector on it. For author-blind selectors, Theorem 24 gives a witness up to $c=0$, and the paragraph after it gives a selector at $c=+1$. Theorem 25 treats the adaptive class. The committee bound follows from $3(n-f) \geq 2n+2f+c$. □

Theorem 22 (One-shot impossibility). *Let $1 \leq q_S, q_F \leq n-1$ and $2q_F+q_S \leq 2n+2f-1$. Then two admissible executions produce byte-identical status sets, with $\mathsf{FAST}(x)$ possible in one and $\mathsf{FAST}(y)$ in the other. Under the liveness caps this covers every choice of thresholds when $n \leq 5f-1$.*

Proof sketch. Assume $q_S \geq 2f-1$ and let $a = \max(f-1, q_F+q_S-n-f)$. The hypothesis gives $2a+1 \leq q_S$. Let S consist of l, whose status carries no PREPARE, sets X and Y of size a, and honest identities with no PREPARE. The leader signs both x and y. X prepares x, Y prepares y, and every member of S closes before any U reaches it. In E_x the faulty set is $\{l\} \cup Y'$, with $Y' \subseteq Y$ and $|Y'| = f-1$. The members of Y' also sign PREPARE(x), and after S closes the outsiders prepare x. Then $X \cup Y' \cup \{l\} \cup O$ has $a+f+o \geq q_F$ members. E_y is the mirror image, with faulty set $\{l\} \cup X'$. Each honest identity signs one PREPARE. In E_x the members of Y' sign the same status bytes that they sign honestly in E_y. Delayed messages are released after the decision. If $q_S \leq 2f-2$, take $a = \lfloor (q_S-1)/2 \rfloor$ and $Y' = Y$. Then the faulty set has $a+1 \leq f$ members and $2a+1+o \geq n-1 \geq q_F$. □

Every honest local view is the same in E_x and E_y. Richer statuses, for example ones listing every PREPARE seen, therefore do not help.

Theorem 23 (One-shot sufficiency). *Assume fast-path safety, $2q_F+q_S \geq 2n+2f$, and either $q_R \geq o+f+1$ or $2q_U+q_S \geq$*

$2n+2f$. *Let $q = q_F$ in the first case and $q = \min(q_F, q_U)$ otherwise. The following selector preserves every possible FAST and FIN.*

1. *If a status carries a complete $U(w)$ or a final certificate, select w.*
2. *Otherwise, if S names two children, select the child v with $r'(v) \geq q + q_S - n - f + [l \notin S]$, or $\perp$ if there is none.*
3. *Otherwise, if S names one child w, select w.*
4. *Otherwise return $\perp$.*

Proof sketch. If S carries $U(w)$ or a final certificate, every execution that produces S contains $U(w)$. Fast-path safety then leaves only w possibly final. Otherwise, in the first case a hidden FIN would need $q_R \leq o + f$ by Lemma 20. In the second case $\mathsf{FIN}(v)$ contains $U(v)$. Either way, every possibly final v has at least q PREPAREs, and Lemma 20 puts it at the step-2 threshold. That threshold is at least $q_S/2 > 0$. Two children at it would need $2q + q_S \leq 2n + 2f - 1$, or $2n + 2f - 2$ when $l \notin S$. Both contradict the hypothesis. A possibly final child that no status in S reports would need $q \leq o + f$, which gives $2q + q_S \leq 2n + 2f - q_S$. □

At $n = 5f$ the inequality forces $q_F = q_S = 4f$. With $q_U = 3f + 1$ and $q_R = 2f + 1 = o + f + 1$, step 2 needs $2f$ non-leader reports when the leader is in S, and $2f + 1$ when it is outside. A variant that selects a lone child only with $r(w) \geq q + q_S - n - f$ always picks a child with an honest preparer, who checked its validity. The argument needs the leader's proposal signature inside each PREPARE. If a PREPARE named only $\mathsf{id}(x)$, a faulty non-leader could report a child that the leader never signed, the equivocation proof would fail, and the bound would return to $5f + 1$.

Theorem 24 (Author-blind selectors)**.** *If $2q_F + q_S \leq 2n + 2f$, there are two status sets with the same multiset of contents, with $\mathsf{FAST}(x)$ possible given one and $\mathsf{FAST}(y)$ possible given the other. Under the liveness caps this covers every $n \leq 5f$.*

Proof sketch. Below equality, Theorem 22 applies. At equality the liveness cap gives $q_S \geq 4f$. Let $c' = \max(f, q_F + q_S - n - f)$, so $2c' \leq q_S$. Each set holds c' reports of x, c' reports of y, and statuses with no PREPARE. In S_1 the leader is one of the y-reporters. Take the faulty set to be l and $f - 1$ non-leader y-reporters. Then $\mathsf{FAST}(x)$ has $c' + f + o \geq q_F$ members. S_2 is the mirror image, with the leader among the x-reporters. □

Sufficiency at $c = +1$ follows from Lemma 20 without the leader correction. When $2q_F + q_S \geq 2n + 2f + 1$, at most one child reaches $q_F + q_S - n - f$ reports. At the design point of the main text this threshold is $2f + 1$, which is the rule of Section 7.

Theorem 25 (Adaptive collector)**.** *An adaptive collector can preserve every possible late FAST if and only if $2q_F + q_S \geq 2n + 2f - 1$. The collector is live.*

Proof sketch. Liveness: once l is proven faulty, the other $n - 1$ identities include at least $n - f$ honest ones. Sufficiency: apply Lemma 20 to the collected set. When l is inside, the set has $q_S + 1$ statuses and $o - 1$ outsiders, which supplies the one status the one-shot bound lacks. Necessity: put l outside S, so the collector does not extend. Take $X, Y \subseteq S$ of size $a = \max(f - 1, q_F + q_S - n - f + 1)$, so $2a \leq q_S$. With $Y' \subseteq Y$ and $|Y'| = f - 1$, the set $X \cup Y' \cup O$ has $a + (f - 1) + o \geq q_F$ members, and the mirror image gives $\mathsf{FAST}(y)$. For small q_S, use $a = \lfloor q_S/2 \rfloor$ and $Y' = Y$. □

At $n = 5f - 2$ every live choice therefore has a witness. This matches KTZ Theorem 4.5 [16]. The adaptive step is the one in KTZ's view change, which waits for $n - f$ votes not including the equivocator's [16].

Remark 26 (The FIN receipt condition). FIN adds the condition $2q_U + q_S \geq 2n + 2f + c$, or $q_R \geq n - q_S + f + 1$. For one-shot selectors ($c = 0$) it is proved in both directions. The witness reuses the construction of Theorem 22 with $a = \max(f - 1, q_U + q_S - n - f)$. It takes receipts only from faulty identities and outsiders, so no honest status in S exposes $U(x)$. At $n = 5f$ with $q_S = 4f$, this forces $q_R \geq 2f + 1$ whenever $q_U \leq 4f - 1$. It is Appendix A moved to $5f$. For the adaptive class the condition was only computed, for $f \leq 5$, and it has a third option: $q_R \geq o + f + 1$, or $q_R = o + f$ with $q_U \geq o + f + 1$, or $2q_U + q_S \geq 2n + 2f - 1$.

What is already known. Both endpoints are published. At $c = +1$ the inequality gives the committee of Hydrozoan's Section IV-B, and at $c = -1$ that of FinWhale's Lemma 4 [22, 4]. KTZ and Abraham et al. reach $5f - 1$ outside DAGs [16, 17]. The two-validator gap between the endpoints is stated in that work too. We found no treatment of the middle point $c = 0$: a one-shot, fixed-size selector that reads the leader's signature and does not wait, at $n = 5f$. FinWhale and Hydrozoan both wait. The middle point is dominated by $c = -1$, which is also live and tolerates one more fault. Its only value is a fixed-size recovery set. Theorem 24 is a class lower bound inside our interface. Its informal version is the remark of Abraham et al. on a tie between two disjoint sets of $2f$ at $n = 5f$ [17]. Since our bounds hold only inside the interface, we do not call $5f$ optimal.

B.2 A parameter set at $n = 5f - 1$

The adaptive class reaches $n = 5f - 1$. Table 8 lists the thresholds beside those of the main text. Fast-path safety needs $n + f + 1 = 6f$, and $2 \cdot \mathsf{FAST} = 8f - 2$, $2U = 6f$ and $\mathsf{FAST} + U = 7f - 1$ all meet it. Two U quorums share $2 \cdot 3f - (5f - 1) = f + 1$ identities, so the argument of Lemma 11 carries over. A set of $f + 1$ closed statuses still includes an honest one, so closure amplification is unchanged. The message-delay ledger of Figure 2 is unchanged as well. With f faulty identities, however, FAST at 2δ needs every honest validator open, since $\mathsf{FAST} = 4f - 1 = n - f$.

Count	$n = 5f + 1$	$n = 5f - 1$
FAST PREPAREs	$4f + 1$	$4f - 1$
U PREPAREs	$3f + 1$	$3f$
FIN receipts	$2f + 1$	$\geq 2f$; checked at $2f, 2f + 1$
Statuses	$4f + 1$, author-blind	$4f - 1$ unexposed
Protection	$2f + 1$ own reports	$2f$ non-leader reports
Ocean parents Q	$4f + 1$	$4f - 1$
DA votes	$3f + 1$	$3f$
STORE signatures	$4f + 1$	$4f - 1$

Table 8: Recovery and ordering thresholds at the two committee sizes. The $5f + 1$ column is the design of the main text.

Adaptive protection. At $n = 5f - 1$ the selector protect$^-$ acts as follows.

1. Collect $4f - 1$ complete closed statuses from distinct authors. If they contain valid PREPAREs for two children, the leader is proven faulty. If the leader's own status is among them, collect one more status from a non-leader.
2. If a status carries a complete $U(w)$ or a final certificate, select w.
3. Otherwise, if the leader is proven faulty, ignore its status. Select the child with at least $2f$ own PREPAREs among the $4f - 1$ non-leader statuses, or $\perp$ if there is none.
4. Otherwise, if exactly one child w is reported, select w.
5. Otherwise return $\perp$.

A valid resolution record carries exactly $4f - 1$ statuses from authors that the set does not prove faulty, and at most one further status, from the exposed leader. Step 3 has at most one candidate, because two children with $2f$ reports each would need $4f$ non-leader statuses. The threshold is Lemma 20 at $q = 4f - 1$ for both placements of the leader: $(4f - 1) + 4f - (5f - 1) - f = 2f$ with the leader inside, and $(4f - 1) + (4f - 1) - (5f - 1) - f + 1 = 2f$ with it outside. This matches KTZ's selection step.

The resolver may choose its status set. Step 4 can select a child that only faulty identities hold. At $f = 1$, a faulty leader may report its own PREPARE for a child it sent to nobody. The same status bytes arise when one faulty non-leader hid its PREPARE and that child is still possibly final, so every preserving rule must select it. No threshold on step 4 avoids this at $f = 1$, or at any f with $2f$ receipts. An honest resolver therefore keeps every closed status it receives and may build its record from any status set the interpreter accepts. It submits a record only if it holds the selected child's complete evidence, or if the set selects $\perp$ and it uses its own fallback. Every valid set preserves every possibly final child, so this choice is safe. Record validity stays a function of the record's bytes, so firstness is unchanged. The set of honest statuses selects $\perp$ or a child that an honest validator holds, so some honest resolver eventually submits a valid record. The interpreter of Section 7, the fallback child and the parent proof are unchanged.

B.3 An Ocean on an uncertified DAG at $\mathsf{Q} = 4f - 1$

The native Ocean of Section 8 relies on reliable broadcast for slot uniqueness and totality. A separate study replaced it with an uncertified DAG at $5f + 1$. There, each author pushes each vertex once to the other $n - 1$ identities. A receiver installs a vertex after its parents and fetches missing parents from the sender. Correct identities retain and serve what they installed, and statuses, resolution records and coin shares ride in vertex bodies. That study counts approvers directly, with $\mathsf{Q} = 4f + 1$ and a four-round wave. Its thresholds, propagation lemmas and edge-counting core repeat BB-Core-Async [21]. Its approval predicate and its exclusion of equivocators are those of Cordial Miners [18]. Its approval-count scan holds only at $n = 5f + 1$ and fails at $5f$ and $5f - 1$ for every f checked. At $5f - 1$ a later cone can hold only $2f - 1$ honest approvers of a version L beside $2f$ approvers of a rival L', so no count of approvers separates them.

We therefore keep that substrate and use Cordial Miners' asynchronous certificate rule [18]. A vertex has at least $\mathsf{Q} = n - f = 4f - 1$ strong parents from distinct authors at the previous round, including its author's own previous vertex. Its identity is its digest, so a faulty author can sign two vertices for one slot. Two such signed vertices are equivocation evidence. After holding evidence against an author, a correct author stops referencing it, as in Cordial Miners. This rule affects cost only.

Definition 27 (Certificate commit rule)**.** Wave w covers rounds $5w - 4$ to $5w$. Offsets count from round $5w - 4$.

1. *Offset 0.* The leader slot is $(5w - 4, \mathrm{coin}(w))$.
2. *Offset 3.* A vertex v *approves* a version L of that slot if L is in v's strong cone and no other version of L's slot is in v's full cone. Coin shares for wave w ride only on offset-3 vertices.
3. *Offset 4.* A vertex is a *certificate* for L if at least t_c distinct authors among its strong parents approve L.
4. *Direct commit.* A reader commits L directly when at least t_d distinct authors in its view hold a certificate for L.
5. *Backward scan.* From a new anchor, for each earlier unclosed wave j, select the version of j's leader slot that has a certificate in the current anchor's strong cone, if one exists, and make it current.

Emission follows Section 8, keyed by digest, so both versions of an equivocated slot may be emitted. The defaults are $t_c = t_d = \mathsf{Q}$. The proofs need only $t_c \geq \lceil (n + f + 1)/2 \rceil$, which is $3f$ at $5f - 1$, and $t_d \geq 2f + 1$.

Approval is Cordial Miners' Definition 18. A certificate is their ratifying block, with a supermajority of more than $(n + f)/2$ approvers; that is the minimum t_c. Cordial Miners runs this rule at $n = 3f + 1$ with five-round asynchronous waves, and Mahi-Mahi uses the same wave shape at $3f + 1$ [18, 19]. We run it at $n = 5f - 1$ with $\mathsf{Q} = n - f$. The lemma numbers L1–L12 follow the research note.

Lemma 28 (L1, certificate uniqueness)**.** *At most one version of a slot has a certificate.*

Proof. Certificates for $L \neq L'$ share at least $2t_c - n \geq f + 1$ approver authors, so one of them is honest. An honest author has one offset-3 vertex, and it cannot approve both versions. The bound is tight at $n = 4$, where $2\mathsf{Q} - n = 2 = f + 1$. □

Lemma 29 (L2, propagation)**.** *If L in wave w commits directly, every vertex above round $5w$ has a certificate for L in its strong cone. With $t_d = \mathsf{Q}$, it has at least $n - 3f$ of them.*

Proof. A vertex at round $5w + 1$ misses at most $n - \mathsf{Q} = f$ authors at round $5w$. At least $t_d - f \geq f + 1$ certificate holders are honest, each with a unique vertex, so one is among its parents. Higher vertices reach such a vertex. □

Lemma 30 (L3, one direct commit per wave)**.** *Two direct commits in one wave choose the same version.*

Proof. Each direct commit needs a certificate for its version, so Lemma 28 applies. □

Lemma 31 (L4, prefix agreement)**.** *Honest readers produce prefix-compatible anchor sequences.*

Proof sketch. Let w be one reader's highest direct wave and $w' \geq w$ another's. The second reader's scan reaches wave w from an anchor at round at least $5w + 1$. By Lemmas 28 and 29 it finds the unique certificate and selects L_w. Cones are fixed, so the scan then repeats the first reader's scan exactly. □

Lemma 32 (L5, common transcript and first resolution)**.** *Prefix-compatible anchor sequences give prefix-compatible record transcripts. Every honest reader finds the same first valid resolution for each parent.*

Proof. Emission and resolution validity are deterministic functions of complete cones. The first-resolution rule of Section 7 reads only the transcript. □

Lemma 33 (L6, common core)**.** *Let a seal be the strong parents of an honest offset-3 vertex. It determines at least $n - 2f$ honest offset-0 versions that every offset-3 vertex approves and every offset-4 vertex certifies.*

Proof. The seal has at least $\mathsf{Q} - f = n - 2f$ honest members. Counting edges to honest offset-1 vertices gives a hub y with at least $f + 1$ honest supporters in the seal whenever $(n - 2f)^2 > f(n - f)$. At $5f - 1$ the difference is $5f^2 - 5f + 1 > 0$. Every offset-3 vertex misses at most f authors at offset 2, so it reaches a supporter and hence y. The honest offset-0 parents of y are unique versions, so every offset-3 vertex approves them. An offset-4 vertex has $\mathsf{Q} \geq t_c$ such parents, so it certifies them. □

Lemma 34 (L6b, coin timing)**.** *Each honest author's core is fixed when its coin share is created, so before $f + 1$ shares exist. Nothing after the reveal shrinks the core.*

Proof. An honest share exists only inside its author's offset-3 vertex. That vertex's parents already exist, and y and its core depend only on existing vertices. The adversary needs one honest share beyond its own f, so $f + 1$ shares exist only after some honest author has shared. Its core is already fixed. The $f + 1$ honest supporters of y already exist. Any later offset-3 vertex, including a faulty one signed after the reveal, has Q distinct-author parents and so includes one of them. □

The counting fails at $n = 3f + 1$ for $f \geq 2$, and seals with no hub occur there. The proof does not cover $3f + 1$.

Lemma 35 (L7, per-wave success)**.** *Under Assumption 7, with the seal of Lemma 34 and an unpredictable unique threshold coin, a wave succeeds with conditional probability at least $(n - 2f)/n = (3f - 1)/(5f - 1)$. The bound is tight for this rule.*

Proof sketch. The lower bound follows from Lemmas 33 and 34. For tightness, before the reveal every vertex at offsets 1 to 3 skips the same f honest authors. This is legal, since $\mathsf{Q} = n - f$ parents remain. Those f honest slots and the f faulty slots then fail. □

Lemma 36 (L8, four-round variant)**.** *With a four-round wave, a wave succeeds with probability at least m_h/n, where $m_h \geq \lceil ((n-2f)^2 - f(n-f))/(n-3f) \rceil$.*

The proof counts edges from the seal directly to the honest offset-0 vertices, as in Lemma 33. At $5f-1$ this bound is 0.25, 0.44 and 0.50 for $f = 1,2,3$, so the four-round wave needs fewer expected rounds than the five-round wave only for $f \geq 3$.

Lemma 37 (L9, honest rounds advance)**.** *No honest author waits for a faulty one to complete a round.*

Proof. There are $n - f = \mathsf{Q}$ honest authors, so their vertices fill every parent quorum. □

Lemma 38 (L10, collection terminates)**.** *Step 1 of* protect$^-$ *completes at $n = 5f-1$.*

Proof. Exposure proves the leader faulty, which leaves at most $f-1$ other faults. There are then $(n-1)-(f-1) = 4f-1$ honest non-leaders. With $f-1$ silent faults, the collector waits for every honest status. □

Remark 39 (L11, selector check). At $n = 5f-1$ with $q_F = q_S = 4f-1$, $q_U \in [3f, 4f-1]$ and $q_R \in \{2f, 2f+1\}$, two independent checkers find that protect$^-$ preserves every late FAST and FIN for $f = 1,2,3$. Both report a failure at $q_R = 2f-1$ for $f = 2,3$. The FAST half holds for every f by Theorem 25. The FIN half is computational only.

Theorem 40 (L12, one child per parent at $5f-1$)**.** *Assume the property of Remark 39 and the signing and storage contracts at the parameters of Table 8. With the Ocean of Definition 27, all valid finality certificates for one semantic parent certify the same immediate child.*

Proof. FAST and FIN cannot conflict, because two U quorums share an honest identity. Every effective resolution preserves a possible fast final by assumption. Two effective resolutions agree by Lemma 32. The argument is that of Theorem 13. □

B.4 Costs

The numbers below come from the research scripts. They measure models, not a deployment.

- *Waves.* A wave has five rounds, not four. The worst-case per-wave success is $(3f-1)/(5f-1)$, so expected rounds between commits are at most 10, 9 and 8.75 for $f = 1,2,3$, tending to $25/3 \approx 8.33$. The four-round approval-count wave at $5f+1$ needs 6.0 to 6.66, but its per-wave bound $(3f+1)/(5f+1)$ is only sketched. A good-case commit lands about 5δ after the leader slot, against 4δ for the four-round wave.
- *Messages and bytes.* Each round sends $n(n-1)$ vertex pushes, plus 4–9% for fetches. Over $n = 4$ to 24 the log-log slopes are 2.11 for messages and 2.90 for bytes. At $n = 24$ a round is 598 messages and 492 KB. Sustained equivocation raises the slopes to 2.38 and 3.47 through fetches. Excluding equivocators returns them to 2.09 and 2.90.
- *Vertex size.* A vertex is 83 B plus 36 B per reference, with at least Q references, and a 48 B coin share once per wave. That is $\Theta(n)$ per vertex and $\Theta(n^3)$ bytes per round.
- *Recovery.* A status is 260 B, and a resolution is $(q_S+1) \cdot 260 + 132$ B. Statuses are copied, so one recovery event costs $\Theta(n^3)$ bytes. Faulty authors can attach a resolution to every version, a mean of 10.5 MB per round at $n = 24$. A per-author body cap is needed.
- *Against the broadcast Ocean.* From the $5f+1$ study, with no faults and coin traffic excluded, an uncertified round sends $n(n-1)$ messages. Section 8's broadcast sends $n(n-1)(2n+1)$ with proposer-only relay, $3n^2(n-1)$ eagerly and $4n^2(n-1)$ in the native schedule. At $n = 21$ that is 420 against 18,060, 26,460 and 35,280. With the same vertex encoding, bytes fall 3.6–4.4× against proposer-only relay and 8.6–24.4× against eager relay at $n = 6$ to 21. In native wire encoding they fall 25–77× at $n = 6$ to 16. This saving is against our own broadcast Ocean only. Any uncertified DAG sends $n(n-1)$ messages per round.

B.5 Storage and availability at $n = 5f-1$

The data path of Section 4 carries over with smaller thresholds. Each value below is at most $n-f$, and each formula gives the main text's value at $n = 5f+1$.

- A DA certificate needs $\lceil (n+f+1)/2 \rceil = 3f$ votes. Two certificates share $2 \cdot 3f - (5f-1) = f+1$ voters, so Lemma data holds, and $2f$ honest voters keep each certified path. Keeping $3f+1$ would not be live at $f = 1$.
- STORE needs $4f-1$ signatures, which leaves $3f-1$ honest holders.
- The vector pool keeps $n-f$ records. Its eligibility rank becomes $n-2f = 3f-1$ and its carry becomes $n-3f = 2f-1$. With today's $3f+1$ and $2f+1$, a position that every honest record holds can lose eligibility.
- Without reliable broadcast, a STORE signer treats an Ocean vertex as retained once it has accepted that vertex with its complete ancestry. A source cut becomes a root vertex plus its hash-linked ancestry, and it may hold two versions of one slot.
- Honest authors weak-reference at most one version per slot.

The two-route broadcast of Section 8 cannot exist at $n = 5f-1$: seed uniqueness needs $E \geq 3f$, fast totality needs $F \geq E+f$, and liveness needs $F \leq 4f-1$. A signed run of the repository's broadcast model at $f = 1$ shows one honest node delivering while the others never do. The Ocean of Definition 27 needs no broadcast, so the claim is not affected. Fresh data then reaches FAST in 4δ to 5δ instead of 6δ, because an uncertified vertex is accepted after one hop.

B.6 Status of these results

Proved by hand, for every f: Theorems 21 to 25, for FAST in all three classes; the one-shot FIN condition; Lemmas 28 to 38, including the lower bound and the tightness of Lemma 35; and Theorem 40, conditional on Remark 39 and the storage contract.

Checked by script. The selector searches are exhaustive over safe and live threshold tuples for $f = 1$ to 5. For $f \leq 3$ they cover every q_S. For $f = 4,5$ they fix $q_S = n-f$, which is the most favourable case for the selector. The closed-form survivor predicates match the computed sets in all 49 (f, n, class) cells, and the closed-form rules were rechecked status set by status set with no failures (475,860 orbits at $n = 25$). Ed25519 witnesses cover $f = 1$ to 4. The FIN condition for the adaptive class is computational only, for $f \leq 5$. For the Ocean, a fuzzer ran 3000 adversarial DAGs per rule and committee for $f = 1,2,3$ with no violations. Mutants one below each threshold fork, but the fuzzer catches them reliably only at $f = 1$, rarely at $f = 2$ and never at $f = 3$. Beyond $f = 1$, tightness rests on the hand proofs and one hand-built counterexample at $f = 2$. The coin timing check sampled 900 pre-reveal states for each $n \in \{4,5,6,9,10,11,14,15,16\}$. Remark 39 covers $f \leq 3$ only. A network simulation over $n = 4$ to 24 saw no forks, no stuck honest node and no recovery split, and all 1410 collectors finished. The scripts are in `docs/v4/research/lb/`, `compose/` and `n2/`.

The storage and availability counts of Section B.5 are proved by hand and checked by script for $f \leq 6$, with brute force over fault placements. A cut fuzzer and the recovery checker cover $f \leq 3$ and $f \leq 4$. These scripts are in `docs/v4/research/storeda/`.

Open or conjectured. The main text is not rewritten at $n = 5f-1$. This appendix restates what changes. The inclusion delay bound under the one-version weak-reference rule is conjectured. The resolver rule and the STORE and cut changes are hand proofs, and no simulation runs the $5f-1$ recovery path end to end. The native runtime still uses the $5f+1$ constants. The simulated recovery model has no U or FIN fields, and River votes do not ride in vertices. Mahi-Mahi's higher per-wave bound at $3f+1$ is not reconciled with the adversary of Lemma 35. A reveal at offset 4, closure-based certificates and pipelined leaders were not evaluated. Referencing statuses through the resolver's cone would make a resolution $O(n)$ bytes, but needs its own validity proof. That the combination is new is conjectured; we discuss the search

in Section 13. Nothing in this appendix is mechanized, and none of it has been peer reviewed.

C Local transport and extension-transfer experiments

The September 8 run tests two isolated research components, not an implementation that integrates Alpenglow, Multimmit and Rohu-Paplet. The hardware was an AMD Ryzen 9 5900X with 24 logical CPUs and approximately 32 GB RAM, running Linux 7.0.0-31, Rust 1.93.0, Python 3.12.3 and cryptography 41.0.7. The Rust transport was release built. Sixteen cases with three fresh-process repetitions at seed 1 produced 48 invocations and 96 direct/coded outcomes. Ninety obtained native DA certificates; all six deliberate permanently-withholding outcomes remained incomplete. Seven transport tests and seven signed extension-model tests passed.

C.1 Actual bytes and modeled network service

The prototype uses Reed–Solomon coding, a signed Merkle commitment, authenticated fragments, reconstruction and re-encoding checks. Before a DA vote, the native component validates and retains the complete block and checks snapshot restoration. There is one original producer, equal-weight identities and fixed relay assignment; this is not Rotor's stake sampling. The code has $k = n - f$, while certificates remain native full-holder DA certificates.

Table 9 counts actual protocol frames, including authentication metadata and votes. Both transports use the same JSON-serialized native block, which expands the application payload. Producer and source-relay traffic share one physical uplink. No production compact-wire efficiency is assumed.

At 64 KiB, producer upload falls 59.7% while total traffic rises 20.8%. At 1 KiB the aggregate penalty is 52.1%. With eleven validators and a 64 KiB payload, producer bytes fall from 2,342,896 to 527,706 (77.5%), while aggregate bytes rise from 2,347,359 to 2,904,659 (23.7%). The eleven-validator matrix case is honest; it does not test two concurrent faults.

The deterministic network model serializes each physical node's uplink and adds 10 ms one-way propagation, with unlimited ingress. It excludes actual coding, verification, persistence and driver durations. Its endpoint is a verified DA certificate at validator 1, not ordered execution or consensus finality. Repeated projections are identical, not independent network samples.

At 64 KiB and 100 Mbps, the six-validator projection is 76.30 to 54.06 ms; all correct holders reconstruct by 103.74 to 47.79 ms. With eleven validators, the certificate projection is 132.51 to 59.60 ms. The extra hop loses when serialization savings are small, as the small-block fast-link cases show.

C.2 Measured local execution and faults

Fifteen invocations also use six live TCP node threads in one process. A sequential driver sends framed commands and nodes execute the actual relay, validation, reconstruction and native DA handlers. There is no wall-clock network shaping. Socket bytes include control envelopes and replies and are kept separate from protocol-frame counts.

Coded relay is slower in this local harness. The durations include all scheduled work, first-use costs and driver overhead, rather than time only until a certificate forms. The host is not isolated (initial one-minute load average 7.80); three repetitions do not justify tail-latency claims. Operation timers use wall-clock Instant, not CPU counters. Persistence is an overlapping subset of native DA/restore time and cannot be added twice.

The fsync cases perform atomic temporary-file writes, file and directory fsync, readback and 36 complete-holder snapshot restorations before releasing votes. Both modes retain 1,409,494 aggregate snapshot bytes. These are restoration checks, not injected process crashes or proof of rollback detection.

The matched six-validator fault cases use 64 KiB, 100 Mbps and 10 ms delay: Faulty identities contribute no DA votes in either mode. The corrupted coded fragment causes four rejected deliveries. The withholding case needs repair after a 228.40 ms deadline; coded delivery is slightly slower. The first three fault cases also pass through live TCP. Repair is authenticated but served only by the original source. The experiment does not implement surviving-holder retrieval after source loss, fragment-only custody, or uniform reliable broadcast.

C.3 A separate signed extension model

The model uses real Ed25519 votes and statuses, six validators, two producer lanes, genesis bases, one fixed proposal per family and at most one extension per lane. Honest reporters have already voted. Byzantine alternatives range over declared canonical choices, not all syntactically accepted vectors.

Three counterexamples guide the design. First, sweeping independently finalized lane tips while skipping unresolved earlier positions gives incompatible orders in 388 worlds, for example $[B_1]$ versus $[A_1, B_1]$. Stopping at the first unsettled gap avoids those tested conflicts. Second, identical authenticated statuses saying only "I voted" conceal old certificates with incompatible late outputs $[A_1]$ and $[B_1]$; retaining complete signed contributions distinguishes the executions. Third, complete evidence still admits competing unfinalized recovery continuations in 2,430 worlds. Preserving possible old finality does not make different new choices agree.

A fresh-data schedule admits A_1 after the proposal, orders it without an unavailable B_1, preserves it through retirement and late certificate assembly, and rejects five old-attempt signing replays. Mixed honest no-votes, general Byzantine vectors, recursive bases, multiple views, split-lock progress and repeated-fault service bounds are outside that model. These counterexamples target deliberately weakened candidate interfaces, not the published Alpenglow or Multimmit protocols. This model is also separate from the new exact-child kernel of Section 6.

C.4 Reproduction

From the repository root:

```
python3 bench/local_composition_lab.py \
  --include-loopback --include-fsync --repeats 3
python3 -m unittest \
  spec.experiments.test_extension_transfer -v
cargo test --locked \
  --manifest-path rohu-paplet/Cargo.toml \
  --example coded_relay_lab
```

The raw directory is `bench/results/local-composition-20260908-224511/`. It contains the machine/configuration manifest, source and binary fingerprints, per-process commands and outputs, transport JSONL, complete model witnesses and completion assertions. Source, executed binary and recorded Commonware dependency state remained unchanged during the run. The detailed result record is `docs/v4/LOCAL_COMPOSITION_RESULTS_2026-09-08.md`.

D Design history and inherited obligations

This appendix records the earlier work relevant to the complete consensus account. It does not specify an alternate current mode. The archived manuscripts, formal artifacts, and raw evidence remain available in the repository [1, 2].

D.1 From two readers to exact parent domains

The earlier weather-aware Rohu-Paplet draft and the two-plane manuscript that followed it [2] developed a weather-dependent reading of a control DAG: River for favorable conditions, Monsoon for a collectively derived boundary, Ocean for asynchronous commitment, and Heal for re-entry. Their visibility-gap and re-entry arguments explain why the absence of a known fast result does not authorize an independent recovery result. The later era-based v4 manuscript, the archived era baseline of this revision, added complete-artifact receipts, native broadcast and Ocean, and an explicit barrier–intent–Heal ownership sequence.

The current parent-local design retains producer-chain availability, River prepare/receipt protection, native Ocean

Payload	Block bytes	Producer bytes Direct	Producer bytes Coded	Aggregate bytes Direct	Aggregate bytes Coded
1 KiB	4,469	22,859	12,279	24,874	37,834
64 KiB	234,250	1,171,762	472,437	1,173,770	1,418,305
1 MiB	3,737,394	18,687,483	7,478,518	18,689,520	22,436,575

Table 9: Honest six-validator protocol traffic; all entries are bytes.

Payload	Uplink	Direct	Coded
1 KiB	10 Mbps	31.54	36.47
1 KiB	1 Gbps	20.12	30.06
64 KiB	10 Mbps	583.01	322.84
64 KiB	1 Gbps	25.63	32.27
1 MiB	10 Mbps	8,990.56	4,806.74
1 MiB	1 Gbps	109.71	71.88

Table 10: Six-validator network-only DA projections in milliseconds, excluding computation.

ordering, and deterministic semantic transitions. It replaces global era ownership and re-entry with one permanently closable domain per certified parent. A certified fallback child starts the next opportunity without reopening the predecessor. The former Monsoon/Heal rules therefore belong to the history and native-component description, not the current algorithm.

D.2 The data/control composition obligations

The earlier full paper organized its composition into four obligations. Their questions remain useful, but the answers must be checked against this revision:

T1: Independent control progress.
Missing transaction data must not stop production of otherwise valid control vertices. Here, empty control bodies and the accepted-parent clock supply that separation; parent closure does not wait for application execution.

T2: Certified frontier authority.
Compatible local data views do not alone determine one executable order. Here, complete frontier evidence justifies candidates and exact-child finality ratifies one frontier transition. Bare roots without their required openings are insufficient.

T3: Deterministic extraction.
Observers must agree on the complete ordered contribution, including under delayed ancestry. Here, native Ocean emits complete closed cones and semantic children bind deterministic Sweep deltas.

T4: One log and checkpoint sequence.
Application delivery must follow certified order with consistent durable cursors. Here, exact-child induction gives the semantic prefix; a native checkpoint/state-transfer realization remains an implementation obligation.

The earlier Ivy/mypyvy and calibration artifacts concern their particular transition systems, including older clock, switch, and extraction rules. Their gate counts cannot be combined with current signed tests into a universal proof of the parent-local construction. A future formal refinement must map its states, proof representations, Pending semantics, and publication actions explicitly to the new runtime.

D.3 What the earlier deployment taught

The earlier full manuscript reports eleven validators across five cloud regions on four continents, using inter-region links and a threshold-BLS coin. It demonstrates its own weather behavior and also records a failure exposed by a stronger consistency oracle [2]. A lagging observer emitted the same agreed anchors in a different order. Append-only publication made that ordering persistent; agreement on head position had not revealed it.

The reported failure occurred at the interface between arriving ancestry and the canonical-log emit path. Differential comparison of incremental versus full computation on a fixed complete DAG did not exercise that delivery schedule. The resulting methodological requirement is to vary both graph contents and arrival order, use distinct payloads, and compare exact output at each position. A matching later suffix does not repair earlier disagreement.

Those experiments are historical observations of an earlier protocol and implementation. Their commit rates, fast-path fractions, and fault profiles are not parent-local v4 performance results. The current evaluation retains their strongest lesson—test the emitted order and the interface that makes it durable—without transferring their benchmark numbers.

D.4 Which integration work survives

Commonware compatibility, linear delivery adapters, state synchronization, runtime reconfiguration, incremental data structures, and verification batching remain useful engineering material. Each optimization must preserve the exact certified sequence under adversarial arrival and restart schedules. In particular, an earlier Heal-based committee transition cannot be reused after Heal is removed without a new authorization rule. A faster serializer must not make partial evidence look complete, and a checkpoint must not erase permanent old-parent exclusions.

E Protocol objects and transition summary

This appendix collects the semantic schemas and verifier obligations dispersed through the main text. All objects use the configured canonical encoding. Signatures authenticate the type tag and complete signed body, including its configuration and instance/domain bindings. Ordered lists, sets of distinct authors, and dependency lists have canonical representations. A digest is checked against its opening; the digest alone is not an availability proof. Replaceable proof attachments are checked separately when they are not part of the signed body. In particular, a receipt's signed child binding does not include a particular supporting U signer subset.

E.1 Object schemas

Table 14 collects the authenticated fields and supporting evidence.

E.2 Common child-verification procedure

For input $(p, x, \pi_p, \mathscr{E})$, where $\mathscr{E}$ contains cut, frontier, and data evidence, perform these steps in dependency order:

1. Check canonical structure, configuration, and hashes. Verify π_p through finite, acyclic dependencies to the configured genesis; apply the Ocean-position precedence checks when verifying an ordered record.
2. Require x's predecessor to be $\mathsf{id}(p)$ and its height to be $h_p + 1$. Its prior frontier and log commitment must equal those in p.
3. Verify the accepted cut and its complete parent closure, with $r_x > r_p$. Verify all required DA and compatible-path evidence before treating a coordinate as certified.
4. Derive the mandatory cut contribution and any admitted complete vector contribution. Require the proposed new frontier to equal the resulting coordinatewise maximum with F_p.
5. Recompute Sweep, the ordered block delta, and the new log commitment. Require equality with the exact child's commitments and apply the configured common application-validity predicate.
6. Return Valid only when every required check is complete. Return Invalid on a complete contradiction and Pending on a missing required dependency. For a fast proposal, additionally check the designated leader's signature; for a fallback, require authorization through the resolution rules instead.

64 KiB TCP execution	Direct median (range), ms	Coded median (range), ms
Memory-acknowledged snapshots	63.96 (61.59–64.45)	94.38 (94.36–96.06)
Atomic fsync snapshots	106.38 (98.32–117.94)	135.30 (131.45–143.45)

Table 11: Whole-experiment wall time, three fresh-process repetitions.

Fault	Direct	Coded
One silent validator/relay	76.30	54.06
One relay corrupts a fragment	76.30	54.06
Producer initially withholds, then repairs	277.21	281.13
Producer never answers repair	Incomplete	Incomplete

Table 12: Matched fault cases: modeled DA latency in milliseconds.

Bounded coverage	Count
Worlds	5,554
Five/six-vote fast pools	38,878
Five-reporter recovery pools	158,984
Fast/fast comparisons, including Byzantine alternatives	596,295
Fast/recovery prefix checks	1,112,888
Guarded prefix-extraction violations	0

Table 13: Bounded coverage of the separate signed extension model.

An implementation may cache validated dependencies by their bound semantic identity. A cache hit must not substitute a different configuration, conceal an unavailable firstness prefix, or bypass a proof's position restriction.

E.3 Per-parent state transitions

The durable choice state is open until closure. Choosing a PREPARE does not close it; retaining a complete U can permit a receipt while it remains open. Closing records the status and forbids all new PREPARE/receipt signatures in that domain. This closed state is permanent. Learning FAST, FIN, or a first Ocean resolution is a separate transition that records the final child and enables its domain even if the old domain is already closed.

Therefore the implementation must not represent the protocol as a single global mode bit. It has a map of domain-local restrictions and an independently learned certified chain. A delayed old-parent request is answered from its retained entry. A delayed old-parent resolution is validated against the ordered transcript and installed idempotently. Neither operation resets an entry or changes the identity of a previously certified transition.

F Artifact guide and reproduction

The repository is `https://github.com/etofdn/SingularityBFT`. The executable baseline is commit `2812f22`. The manuscript sources in `paper4/` are assembled by `main.tex`; the older compact `parent_local_protocol.tex` is retained as the September 8 presentation and is not included in this expanded assembly.

F.1 Build and current reference checks

Build the manuscript from `paper4/`:

```
latexmk -pdf -interaction=nonstopmode \
  -halt-on-error main.tex
```

From repository root, run the current signed kernel and Ocean-adapter suite:

```
python3 -m unittest \
  spec.experiments.test_parent_local \
  spec.experiments.test_parent_local_ocean -v
```

The September 10 check passed all 32 tests. Python 3.12.3 and cryptography 41.0.7 were used, matching the recorded environment. These checks verify selected reference behaviors under their modeled boundaries.

The four signed CLI schedules in Table 6 are:

```
python3 -m spec.experiments.parent_local \
  --f 1 --heights 18 --schedule round-robin
python3 -m spec.experiments.parent_local \
  --f 2 --heights 22 --schedule round-robin
python3 -m spec.experiments.parent_local \
  --f 1 --heights 12 --schedule all-silent
python3 -m spec.experiments.parent_local \
  --f 2 --heights 12 --schedule all-silent
```

The option `all-silent` names the synthetic fixed-silent-leader schedule; it does not mean all validators stop processing. These runs use trusted ordered records, not a networked asynchronous ordering deployment.

F.2 Recorded evidence and component suites

Table 15 locates the recorded evidence and component contracts.

Appendix C gives the transport and extension reproduction commands and environment. Their September 8 measurements are retained as recorded component results. The September 10 validation reran the current parent-local suite and four CLI schedules. The earlier native and formal suites remain separate artifacts with their own reproduction instructions.

For any replication, retain the command, exit status, environment, source and binary fingerprints, raw outputs, workload, and unresolved requests. For a negative schedule, record the mutated rule and resulting conflicting proofs or outputs. A successful command without its model boundary is insufficient evidence for a broader theorem or deployment claim.

References

[1] Naman Bajpai and contributors. *Rohu-Paplet protocol and executable artifacts*. Source baseline `2812f22`, September 8, 2026. https://github.com/etofdn/SingularityBFT/tree/2812f22.

[2] Naman Bajpai. *Singularity BFT: Adaptive Two-Plane BFT*. Earlier full manuscript and evaluation under the protocol's former name, retained in `paper2/` at the source baseline. https://github.com/etofdn/SingularityBFT/blob/2812f22/paper2/main.pdf.

[3] Andrew Lewis-Pye and Patrick O'Grady. *Multimmit: Extending Blocks for Faster Finality*. Version 4, August 28, 2026. https://arxiv.org/abs/2607.21021v4.

[4] Razya Ladelsky and Roy Friedman. *FinWhale: An Optimally Resilient Two-Round Terminating DAG Protocol*. Version 2, August 18, 2026. https://arxiv.org/abs/2606.26292v2.

[5] Preston Vander Vos and Daniel Cason. *Fast Tendermint: Speeding Up a Foundational Consensus Protocol*. Version 1, August 13, 2026. https://arxiv.org/abs/2608.13434v1.

[6] Lefteris Kokoris Kogias and Alberto Sonnino. *Orcaella: Hybrid Fault Tolerance with Client-Selectable Finality Latency*. Version 2, August 9, 2026. https://arxiv.org/abs/2607.04789v2.

[7] Kushal Babel, Fatima Elsheimy, Lioba Heimbach, Mohammad Mussadiq Jalalzai, Tobias Klenze, Jovan Komatovic, Jason Milionis, Mike Setrin, and Victor Shoup. *Cadence: Extreme Pipelining with Multiple Concurrent Proposers*. Version 2, July 7, 2026. https://arxiv.org/abs/2607.02275v2.

[8] Nibesh Shrestha, Qianyu Yu, Aniket Kate, Giuliano Losa, Kartik Nayak, and Xuechao Wang. *Optimistic, Signature-Free Reliable Broadcast and Its Applications*. Version 4, May 24, 2026. https://arxiv.org/abs/2505.02761v4.

[9] Xiaohai Dai, Yiming Yu, Sisi Duan, Rui Hao, Jiang Xiao, and Hai Jin. *Icarus: Achieving Performant Asynchronous BFT with Only Optimistic Paths*. NDSS, 2026. https://www.ndss-symposium.org/wp-content/uploads/2026-f60-paper.pdf.

[10] Pierre-Louis Aublin, Rachid Guerraoui, Nikola Knežević, Vivien Quéma, and Marko Vukolić. *The Next 700 BFT Protocols*. Extended author manuscript, 2014, Sections 3.3–3.4. https://www.iijlab.net/en/members/pierrelouis/tocs-700.pdf.

Object	Authenticated contents and evidence
Producer block	Configuration, producer, sequence, predecessor digest, payload/transaction commitment, producer signature; opening supplies the ordered payload and authenticated path
DA vote/certificate	Typed vote for the exact producer block; certificate contains $3f+1$ distinct matching votes and complete compatible-path evidence
Control SEND	Ocean configuration/instance, round, author, exact strong and weak parent references, canonical body, sender signature; complete parent closure accompanies or is retrieved for acceptance
ECHO / READY	Distinct signature types binding the same slot and payload digest; ECHO is backed by full payload retention; a delivery proof identifies its fast or slow route explicitly
Accepted cut	Exact accepted root vertex and complete certified strong-and-weak cone; its identity commits semantic contents independently of delivery-proof signer subsets
Frontier evidence	Exact parent frontier, cut identity and its complete contribution, resulting vector, DA/path evidence, and any complete optional proposal-relative vector pool
Child proposal	Exact semantic tuple from Section 5, complete validity evidence, exact parent's finality proof, and designated-leader signature
PREPARE	Signature binding configuration, parent domain, exact child identity, and PREPARE kind
Receipt	Signature acknowledging complete valid U protection of that same child; supporting complete evidence is retained and supplied for verification
Closed status	Parent domain, author's earlier own PREPARE if any, and complete retained U protections; the immutable signed body records permanent closure
STORE	Object type, canonical length, content hash, chunk Merkle root, exact acyclic dependency list; $4f+1$ distinct signatures certify full-object retention
Resolution	Exact parent, parent finality proof, exactly $4f+1$ complete closed statuses, and complete selected/fallback child under normal-form selection
Ocean finality	Ordered record position and digest, resolution opening, and authenticated preceding transcript establishing first valid resolution; embedded Ocean ancestry must cite earlier positions recursively
Coin contribution	Configured coin signature domain including Ocean genesis instance and wave; release is gated on a durable complete local seal, whose digest is not part of the common coin message

Table 14: Protocol object schemas. Every signature also binds its configured type and domain.

Location	Contents and scope
`docs/v4/PARENT_LOCAL_RESULTS_2026-09-08.md`	Recorded current-reference tests, exact schedule counts, assumptions, and commands
`bench/results/parent-local-20260908-232242/`	Manifest, source hashes, test log, schedule outputs, and completion assertions
`spec/v4/README.md`	Earlier native-model RBC, River, Ocean, and adapter boundaries; distinct from the new parent-local kernel
`docs/v4/EVIDENCE.md`	Native component coverage and revision boundaries
`docs/v4/NATIVE_CONTROL.md`	Finite durable control-driver contract
`docs/v4/RECOVERY.md`	Snapshot publication, retention, and restore contract
`bench/results/local-composition-20260908-224511/`	Transport frames, modeled service, TCP/fsync runs, and signed extension witnesses
`spec/ivy/README.md`	Earlier formal models, calibration, and proof lineage

Table 15: Artifact locations relative to the repository root and their scope.

[11] Idit Keidar, Eleftherios Kokoris-Kogias, Oded Naor, and Alexander Spiegelman. *All You Need is DAG*. 2021. https://arxiv.org/abs/2102.08325.

[12] Alexander Spiegelman, Neil Giridharan, Alberto Sonnino, and Lefteris Kokoris-Kogias. *Bullshark: DAG BFT Protocols Made Practical*. Version 3, 2022. https://arxiv.org/abs/2201.05677v3.

[13] Quentin Kniep, Jakub Sliwinski, and Roger Wattenhofer. *Solana Alpenglow Consensus: Increased Bandwidth, Reduced Latency*. Anza whitepaper, version 1.2, July 29, 2026. https://www.anza.xyz/blog/alpenglow-a-new-consensus-for-solana.

[14] Xiaohai Dai, Chaozheng Ding, Hai Jin, Julian Loss, and Ling Ren. *Ipotane: Balancing the Good and Bad Cases of Asynchronous BFT*. NDSS, 2026. https://www.ndss-symposium.org/wp-content/uploads/2026-s3-paper.pdf.

[15] Neil Giridharan, Shubham Mishra, Lorenzo Alvisi, Natacha Crooks, Benjamin Marsh, Hein Meling, Kartik Nayak, and Grzegorz Prusak. *Ambulance: saving BFT through racing*. USENIX OSDI, 2026. https://www.usenix.org/system/files/osdi26-giridharan.pdf.

[16] Petr Kuznetsov, Andrei Tonkikh, and Yan X Zhang. *Revisiting Optimal Resilience of Fast Byzantine Consensus*. PODC, 2021. Extended version 3, July 27, 2021. https://arxiv.org/abs/2102.12825v3.

[17] Ittai Abraham, Kartik Nayak, Ling Ren, and Zhuolun Xiang. *Good-case Latency of Byzantine Broadcast: A Complete Categorization*. PODC, 2021. Version 3, March 10, 2022. https://arxiv.org/abs/2102.07240v3.

[18] Idit Keidar, Oded Naor, Ouri Poupko, and Ehud Shapiro. *Cordial Miners: Fast and Efficient Consensus for Every Eventuality*. DISC, 2023. Version 6, September 22, 2023. https://arxiv.org/abs/2205.09174v6.

[19] Philipp Jovanovic, Lefteris Kokoris-Kogias, Bryan Kumara, Alberto Sonnino, Pasindu Tennage, and Igor Zablotchi. *Mahi-Mahi: Low-Latency Asynchronous BFT DAG-Based Consensus*. Version 2, October 17, 2024. https://arxiv.org/abs/2410.08670v2.

[20] Kushal Babel, Andrey Chursin, George Danezis, Anastasios Kichidis, Lefteris Kokoris-Kogias, Arun Koshy, Alberto Sonnino, and Mingwei Tian. *Mysticeti: Reaching the Latency Limits with Uncertified DAGs*. NDSS, 2025. Version 6, November 24, 2025. https://arxiv.org/abs/2310.14821v6.

[21] Preston Vander Vos, Alberto Sonnino, Giorgos Tsimos, Philipp Jovanovic, and Lefteris Kokoris-Kogias. *BlueBottle: Fast and Robust Blockchains through Subsystem Specialization*. Version 2, July 6, 2026. BB-Core-Async is Appendix G. https://arxiv.org/abs/2511.15361v2.

[22] Qianyu Yu, Lefteris Kokoris-Kogias, and Alberto Sonnino. *Hydrozoan: Latency-Adaptive DAG Consensus under Mixed Byzantine and Crash Faults*. Version 1, September 22, 2026. https://arxiv.org/abs/2609.25918v1.

[23] Yann Vonlanthen, Jakub Sliwinski, Massimo Albarello, and Roger Wattenhofer. *Banyan: Fast Rotating Leader BFT*. Middleware, 2024. Version 3, December 11, 2024. https://arxiv.org/abs/2312.05869v3.

[24] Victor Shoup, Jakub Sliwinski, and Yann Vonlanthen. *Kudzu: Fast and Simple High-Throughput BFT*. Version 2, September 29, 2025. https://arxiv.org/abs/2505.08771v2.