ADIC Omnichain Commerce Intent Mempool and Just-in-Time Compiled Agreements at Zero Fees

Applications, Tap-to-Pay, Fiat Connectivity, and Atomic Cross-Ledger Settlement

Samuel Reid

August 23 2025

MSC 2020: 68M12, 68M14, 05C65, 55N31, 11S80, 94A60, 91A43

Abstract

This applications paper introduces the *ADIC Omnichain Commerce Intent Mempool* (OCIM) and *Just-in-Time Compiled Agreements* (JITCAs). OCIM is a feeless (refundable-deposit) broadcast layer on ADIC for *finalized intents* that may settle *on ADIC* or *externally* (e.g., BTC/ETH/SOL/fiat). It details the message types, state-machine semantics, proof-of-finality artifacts, and atomic settlement patterns (HTLC/adaptor signatures/light-client checks). We also formalize a contactless *tap-to-pay* consumer card outside of the credit system and show how ADIC's dual finality (structural k-core; topological/spectral) under multi-axis p-adic diversity enables reliable zero-value messaging and value-bearing transfers at Internet scale.

Design lineage. OCIM/JITCA build on ADIC's feeless consensus with refundable deposits, MRW tip selection, axis-diverse admissibility, and dual finality from the Yellow Paper and the Math paper.

Contents

1 Introduction and contributions

ADIC is a feeless, reputation-weighted ledger where each message attaches to $d+1$ parents across distinct p-adic neighborhoods, enforcing multi-axis diversity at attachment time; finality is certified by (F1) a diversified k-core and (F2) stabilization of weighted persistent homology.

This paper. We present:

1. OCIM: a finalized, feeless intent rail with a minimal JSON schema, semantics, and SLAs.
2. JITCAs: compilation of intents into settlement-native transactions on ADIC or external rails.
3. Atomic cross-ledger patterns: HTLC, adaptor signatures, and proof-of-finality light clients.
4. Tap-to-pay without credit: an NFC card that publishes intents and settles on best-available rails.
5. Fiat connectivity: mapping to ISO 20022 message fields, RTP/ACH/SEPA handshakes, and compliance guidance (hash-first, DA pointers).

2 Background and prerequisites

2.1 Protocol primitives (summary)

Admissibility. For radii $\rho = (\rho_1, \ldots, \rho_d)$ and threshold q, a candidate parent set $A(x) = \{a_0, \ldots, a_d\}$ is admissible if (C1) $\mathrm{v}_p(\varphi_j(x) - \varphi_j(a_k)) \geq \rho_j$ for all axes j, (C2) at least q *distinct* p-adic balls per axis are represented among the parents, and (C3) reputation floors hold.

Tip selection (MRW). A multi-axis Gibbs-type random walk biases towards ultrametric proximity, reputation, and non-conflict.

Finality. (F1) diversified k-core with reputation and depth thresholds; (F2) topological stabilization. The Math paper adds a *sheaf-spectral finality* (SSF) using a diversity-augmented Laplacian with a sustained eigen-gap.

Conflict drift. A supermartingale potential over conflict sets exhibits negative drift, implying uniqueness of the winning alternative with finite expected resolution time.

2.2 Zero-value and value-bearing messages

ADIC processes arbitrary messages (notes, attestations, DA pointers) without token transfer, and value-bearing transfers (UTXO/account operations). Deposits are escrowed and refunded on finality; only objective faults are slashed.

3 OCIM: the Omnichain Commerce Intent Mempool

3.1 Intent objects and states

Definition 3.1 (Intent)**.** An *intent* is a standard ADIC message whose payload encodes a commercial desire (buy/sell/pay/invoice/escrow/subscribe/redeem) but transfers no token value at publication. It contains an optional *DA pointer* to a contract template for compilation on a chosen settlement rail.

State machine. Nodes process intents with the usual ValidateAndAttach and Finalize transitions; on finality, the deposit is refunded.

3.2 Canonical JSON schema

```
{
  "type": "Intent",
  "id": "adic:intent:...base58",
  "nonce": 173, "expires": "2025-10-20T23:59:59Z",
  "party": {"pubkey": "...", "proofOfControl": "..."},
  "action": "buy|sell|pay|invoice|escrow|subscribe|redeem",
  "asset_in": "ADIC|ETH|SOL|BTC|USD|...|null",
  "amount_in": "0|<number>", // zero at post time (no transfer)
  "asset_out": "ADIC|ETH|SOL|BTC|USD|...",
  "min_out": "<number|string>",
  "target_rails": ["ADIC","ETH","SOL","BTC","ACH","SEPA","RTP"],
  "settlement_hints": {"gas": "...", "preferredRoute": "..."},
  "conflict_set": "optional-id",
  "DA_pointer": "ipfs://... or https://...#sha256=...",
  "axes": {"time":"t/tau","topic":"LSH(...)","region":"ASN","tier":"..."},
  "signature": "sig(Px, payload)"
}
```

3.3 OCIM semantics (informal)

- **Admission.** MRW selects parents; admissibility enforces axis diversity and reputation floors; escrow D.
- **Finality.** Either (F1) diversified k-core or (F2/SSF) stabilization; upon success: refund D.
- **Immutability.** Finalized intents serve as public commitments; settlement references the `id` and proof-of-finality artifact.

4 JITCAs: Just-in-Time Compiled Agreements

4.1 Compilation targets

A JITCA compiles a finalized intent into settlement-native transactions:

1. **ADIC-native:** a value-bearing ADIC message referencing the intent `id`.
2. **External DLT:** ETH/SOL/BTC transactions (or L2s), constructed from the DA template.
3. **Fiat rails:** ACH/SEPA/RTP instructions (ISO 20022 fields), emitted by a licensed PSP.

4.2 Preconditions and artifacts

Definition 4.1 (Proof-of-finality artifact)**.** A compact object containing (i) `id`, (ii) finality gate used (F1/F2/SSF), (iii) parameters $(k, q, R^*, D^*, \Delta, \varepsilon/\theta)$, (iv) a Merkle-style inclusion witness for the future cone subset, and (v) validator signatures. External verifiers accept this as a condition for release.

4.3 Compilation examples

ADIC-native transfer.

```
ValueTx {
  ref_intent: "adic:intent:...",
  outputs: [{"to":"adic:addr:...", "asset":"ADIC", "amount": 125.0}],
  proofs: {"finality": <artifact>}, "signature": "..."
}
```

ETH ERC-20 transfer (external).

```
// JITCA fills {to, value, data} for ERC-20 transferFrom/permit
tx = {
  "to": "<ERC20>",
  "data": "a9059cbb...<to><amount>", "value": "0x0",
  "precondition": {"adic_finality": <artifact>, "deadline": T+Delta}
}
```

BTC PSBT (external).

```
psbt: { inputs:[...], outputs:[{addr: "...", amount: ...}],
        precondition: {"adic_finality": <artifact>, "hashlock": H, "expiry": T+Delta} }
```

5 Atomic cross-ledger settlement patterns

5.1 Hashed timelocks (HTLC)

Use the intent to commit a hash H. External payouts succeed only upon presenting the preimage before deadline; ADIC publishes/reveals the preimage when finality holds, yielding atomicity without trusted bridges.

5.2 Adaptor signatures and DLC-style flows

The JITCA includes an adaptor signature fragment on the external rail that becomes valid iff an ADIC-finalized condition is satisfied (e.g., a signature share released when SSF threshold holds).

5.3 Proof-of-finality light client

External contracts accept a succinct proof (inclusion + finality predicate) and release funds upon verification; SSF's single-operator eigen-gap is amenable to incremental/light-client checks.

6 Tap-to-pay outside the credit system

6.1 Actors and keys

- **Card (NFC)**: secure element with ADIC key; signs intents; no credit line.
- **POS**: reads intent, adds axes (time/region/tier), submits to ADIC and fronts the refundable deposit if needed.
- **ADIC node**: MRW + admissibility; escrows D; finalizes; refunds D.
- **Settlement adapter**: chooses ADIC/native or external rail; compiles JITCA; executes HTLC/adaptor-sig as required.

6.2 Online flow

1. *Tap:* card emits 0-value intent (`action=pay`) to merchant.
2. *Finality:* POS displays "Approved—Ready to Settle" once F1/F2/SSF passes (refund D).
3. *Settle:* JITCA executes on ADIC or external rail atomically (HTLC/adaptor-sig).
4. *Receipt:* a 0-value `SettlementProof` message records the external tx hash on ADIC.

6.3 Offline (degraded) flow

Commit-only (hash of intent) at POS with later reveal on ADIC; settlement occurs after finality.

6.4 Security notes

Replay protection by nonces/expiries; lost-card revocation via an ADIC attestation; PII remains off-chain (hash+DA pointer); deposits are refundable and can be fronted by provider.

7 Fiat connectivity and compliance

7.1 Message mapping (indicative)

OCIM field	ISO 20022 / rail field (example)
`party.pubkey`	Debtor/Creditor acct reference (mapped via KYC registry attestation)
`amount_in/min_out`	`InstdAmt/EqvtAmt` (pain.001 / pacs.008)
`expires`	Requested execution time / cut-off
`DA_pointer`	`RmtInf` (structured remittance; hash as reference)
`SettlementProof`	End-to-end id + reference to external tx id

Regulatory posture. Keep PII off-chain; publish attestations (hashes) to ADIC; perform KYC/KYB off-ledger via accredited PSPs; use OCIM as a tamper-evident registry of intents, invoices, receipts.

8 Zero-value and value-bearing application patterns

8.1 Zero-value

Attestations, invoices, RFQs/intents, IoT beacons, governance posts: all publishable as 0-value intents with DA pointers and finalized receipts; no gas fees, deposits refunded.

8.2 Value-bearing

Micropayments, storage/bandwidth/compute markets (PoUW hooks), grants and treasury payouts linked to finalized intents; JITCAs choose the optimal rail.

9 SLA-style guarantees from ADIC theory

9.1 Convergence guarantees

Conflict energy exhibits negative drift; uniqueness of winner holds with finite expected resolution, enabling stable intent matching.

9.2 Finality checks at scale

Run F1 first (cheap diversified k-core with thresholds), then PH or SSF; SSF yields a single spectral gap predicate maintained by sparse updates.

10 Reference algorithms and schemas

10.1 Proof-of-finality artifact (canonical YAML)


```yaml
finality_artifact:
  intent_id: "adic:intent:..."
  gate: "F1|F2|SSF"
  params: {k: 20, q: 3, Rstar: "...", Dstar: 12, Delta: 5, eps_or_theta: "..."}
  inclusion: {root: "0x...", witness: ["0x..","0x..", "..."]}
  validators: [{pk:"...", sig:"..."}, ...]
  timestamp: "2025-08-24T..Z"
```


10.2 SettlementProof message (0-value)


```json
{
  "type":"SettlementProof",
  "ref_intent":"adic:intent:...",
  "rail":"ETH|SOL|BTC|ACH|SEPA|RTP|ADIC",
  "txid":"0x... or external-id",
  "status":"executed|reversed|disputed",
  "DA_pointer":"ipfs://...#sha256=...",
  "axes":{"time":"...", "topic":"...", "region":"...", "tier":"..."},
  "signature":"sig(Px, payload)"
}
```

10.3 OCIM processing (pseudocode)

```
process_intent(x):
  if !verify_signature(x): reject
  if !acyclic_per_axis(x): queue
  if !admissible_C1_C3(x): reject
  escrow_deposit(x.P, D)
  attach_simplex(x, A(x)) # MRW + diversity merge
  if finality_F1(x) or finality_F2_or_SSF(x):
     finalize(x); refund(x.P, D)
```

11 Security considerations

Spam/Sybil. Refundable deposits and ADIC-Rep with overlap/motif penalties, enforced axis diversity, and MRW mixing defend against sybil clustering and captive mempools.

Censorship. Parents must spread over q distinct balls per axis; SSF enforces axis consistency and diversity via a single spectral gap.

Reorg risk. Dual finality and negative drift reduce reversal probability; SettlementProof trails create auditability across rails.

12 Operational guidance and parameter presets

Adopt the v1 profile $(p, d) = (3, 3)$, $\rho = (2, 2, 1)$, $q = 3$, $k = 20$, $D^* = 12$, window $\Delta = 5$; PAC-style tuning can refine (ρ, q, k, θ) for workload SLAs.

13 Conclusion

OCIM and JITCAs convert ADIC's mathematical guarantees into practical commerce rails: intents finalize at zero net fee; agreements compile to optimal settlement domains; atomicity and receipts are first-class; a tap-to-pay card functions without credit. This extends ADIC from consensus and theory to a usable omnichain application layer.

A Appendix A: Tap-to-pay protocol roles and timelines

1. **T=0:** Card signs `Intent{pay}`; POS submits (fronts D if needed).
2. **T=O(1):** F1/F2/SSF success; POS prints receipt; D refunded.
3. **T $\leq$ expiry:** JITCA executes (ADIC-native or external via HTLC/adaptor-sig).
4. **Post:** `SettlementProof` anchors external txid on ADIC.

B Appendix B: Axis catalog (suggested)

- Time bucket: $b = \lfloor t/\tau \rfloor$ encoded base-p in $\mathbb{Q}_p$.
- Topic LSH: SimHash/LSH of payload/intention codebook.
- Region/ASN: network/geography codebook (privacy-preserving).
- Tier: service tier / QoS band.

C Appendix C: JITCA template snippets

ACH

```
pain.001: {
  EndToEndId: "<intent_id>",
  InstdAmt: "<amount>",
  RmtInf: { Ustrd: ["ADIC SettlementProof: <txid>"] }
}
```

D Acknowledgments

This paper builds directly on the ADIC White Paper, Yellow Paper, and Math paper, which define the protocol, proofs, and parameterization this applications layer relies upon.