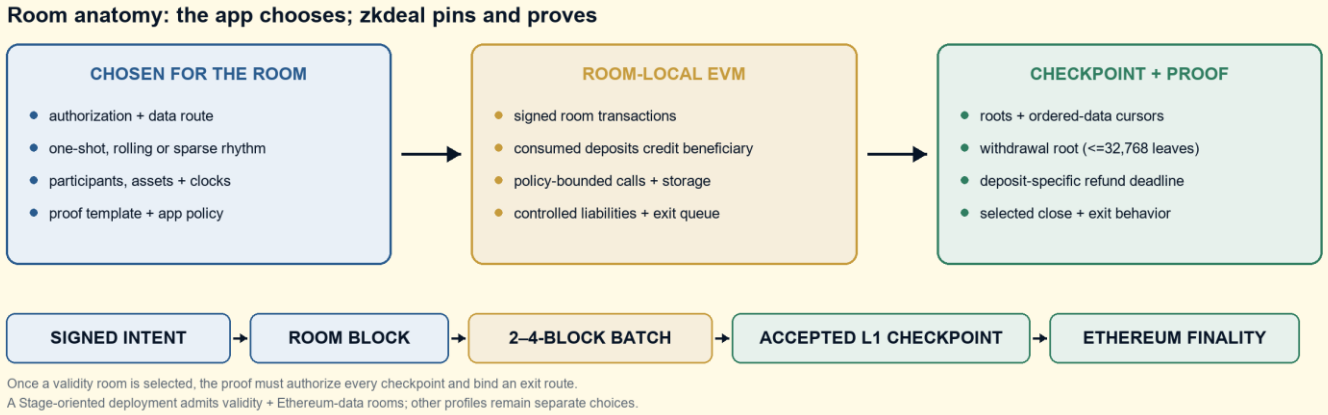
I CORE MODEL • TECHNOLOGY BRIEF

zkdeal: more than a transaction, less than a chain

A guide to zkdeal's room model, capital path and relationship to the proving stacks used by ZK-based Ethereum L2s

Thesis: Most L2s prove a continuing shared chain. zkdeal proves one bounded room. An application chooses an allowed room profile; zkdeal binds that profile at room opening and proves each accepted checkpoint against it. The difference is lifecycle and settlement design, not a new cryptographic primitive.



I.1 The five-stage time ladder and core vocabulary

Room: A room is an isolated EVM workflow with its own policy, assets, membership state and proof program. It has a deliberate close instead of running as one shared, always-on chain.

Settlement sequence: A signed intent enters a room block, and room blocks form a batch. The proof and journal turn that batch into an accepted L1 checkpoint. Ethereum finality makes the checkpoint canonical; until then, the room result remains provisional.

Term	Plain meaning	Term	Plain meaning
Intent	A participant-authorized room transaction or application commitment.	**Room block**	A local EVM block; it is not an Ethereum block.
Batch	A proved sequence of room blocks; current hosted progress uses 2–4 room blocks.	**Checkpoint**	A proof-backed journal accepted by the L1 RoomManager.
Journal	A compact commitment to roots, data, cursors, liabilities, withdrawal leaves and close—not a readable event log.	**Proof program**	The RISC Zero guest/STF that re-executes the batch under the pinned policy.
Forced inbox	An L1 request that must be included, revert, receive a proved rejection, or halt room progress after its deadline.	**RoomVault**	L1 escrow with balances and liabilities segregated per room and asset.
Exit binding	A required validity-only policy component that pins how proved state becomes withdrawal leaves.	**Terminal batch**	A recovery-only final proof on deployment rails that support recovery close; the Stage-oriented validity rail exits through ordinary proof-bearing batches.

I.2 Authorization, membership and checkpoint rhythm

Joint-consent rail: An application may choose unanimous approvers for a small jointly controlled room. Every active approver signs the same checkpoint journal, and participants keep the replay data. This is a deliberate consent-room option rather than the standard rollup Stage path.

Stage-oriented validity rail: An application may choose validity-only settlement with Ethereum calldata. Customers sign room transactions; the bound proof checks input and execution, while exact genesis data, deposits and forced inputs remain reconstructable from L1. The rollup-oriented deployment enforces this combination for every room it admits.

One-shot, rolling and sparse: A one-shot room closes at its first checkpoint; a rolling room checkpoints repeatedly; a sparse room waits longer. Sparse timing can amortize per-checkpoint submission overhead, but it also extends the rollback and operator-dependency window.

I.3 What the proof establishes—and what it does not

Enforced after the validity rail is selected	Chosen by the application or deployment
Validity-proof authorization, exact-genesis binding, deterministic execution, state continuity, Ethereum-data binding and liability conservation. L1-funded validity rooms also enforce beneficiary credit and a room-bound route from proved state to L1 claims.	The application selects admission clocks, ordering and clearing, visibility, participant capacity, assets, confirmations and exit allocation from the deployment's allowed templates. Governance and the outer settlement profile belong to the deployment.

I.4 Where the architecture is useful

Best fit: Choose a room when a high-value workflow has a clear beginning and end—for example, a deal, auction, game, fund episode or controlled settlement—and needs its own provable market rule and room-scoped L1 accounting.

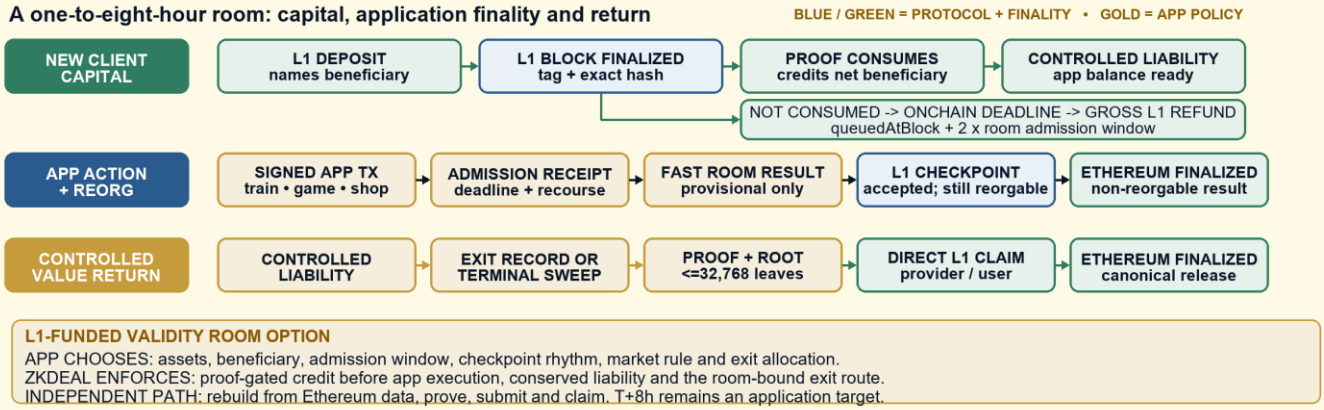
Complementary fit: A persistent L2 remains the natural home for always-on composability, shared liquidity and broad wallet, RPC and indexer support. zkdeal can handle a bounded episode that begins in, returns to or coordinates with that wider market.

Use a continuing chain when: the application needs unrestricted contracts, permissionless entry, live cross-application composability or sustained chain-wide throughput during the episode. The eight-room recursive limit applies to one aggregate receipt; it is not a concurrency limit.

II MARKET ROOMS IN PRACTICE

From a one-checkpoint clear to an eight-hour market

Why use a room? A room lets a market episode end: the application pins its inputs, rule and close. **Reading the figure:** The 1–8-hour window is an application target, not a proof deadline. The figure separates room choices from proof-enforced accounting and Ethereum finality. Depositors name the beneficiary; the application chooses clocks, cadence and exit allocation; and the validity proof makes consumed funding canonical for that beneficiary.



II.1 Worked target: a four-hour power auction

POWER DESK — TARGET PROFILE Six counterparties fund collateral and wait for finality. Each deposit names its bidder; the consuming proof credits the bidder's net amount before application execution and records the controlled liability. At L1 block N, the application closes bidding, runs uniform-price DvP and produces provider and user claims. Physical delivery starts only after Ethereum finalizes that checkpoint.

Capacity envelope: The application selects a power-of-two participant capacity from 128 to 32,768, constrained by the number of supported assets and the withdrawal-tree limit. Unanimous mode caps approvers at 256. On the open deployment rail, one recursive receipt may aggregate 1–8 rooms; the Stage-oriented calldata rail disables aggregation so each accepted root uses its room-pinned proof.

II.2 Three useful room patterns

Room pattern	What the application pins	Why it helps
Sealed RFQ / call auction one terminal block or 2–4-block burst	The application prefunds cash or inventory and pins the cutoff, signed price and size fields, nonce, expiry, clearing rule and exit-record behavior.	Once the room has a client's signed intent and the proof has credited finalized funds to that beneficiary, the client may disconnect. The pinned fallback states whether low volume causes a partial clear, deferral or no trade.
Protected market round repeated 2–4-block batches	The application pins commit and reveal deadlines, the eligible set, an exposure cap and the execution rule. A commit-reveal AMM shows how sealing the set can block reactive insertion.	Once the eligible set is sealed, a batch or uniform-price rule can remove arrival order from price formation. The application must separately encode any liquidity-provider lock, notice period or reserve floor.
Commercial session 1–8 hours	The application pins eligible commodity, freight, power or warehouse lots; escrow rules; L1-block deadlines; checkpoint cadence; exit allocation and terminal DvP close.	A validity-only checkpoint may publish claims before close; a terminal proof can sweep code-free residual accounts. Legal title and delivery still wait for finalized L1 facts and accepted attestations.

Sparse-mode mapping: Choose sparse checkpoint timing only when participants accept a longer provisional window. Between checkpoints, results can roll back. Recovery starts from the latest L1-finalized checkpoint. On the Stage-oriented validity rail, a replacement prover advances or closes the room through the ordinary proof-bearing path; a live all-services-off exercise remains qualification evidence.

II.3 How the room fails closed

Failure mode	Fail-closed room response
An L1 branch changes	The operator should treat the room opening, each deposit or import, and every checkpoint as provisional until Ethereum's finalized tag covers the exact L1 block hash; confirmation depth is not finality. A production host should verify the tag and hash through two providers and fail closed on disagreement. The operator should retain proof and calldata; a changed branch or deadline requires re-proving.
Liquidity vanishes before seal	A quote is not capital. The application requires finalized funding, and each deposit names its beneficiary. If the proof never consumes it, the depositor can reclaim the gross amount after that deposit's onchain deadline. Once consumed, the proof credits the beneficiary and can move controlled value to an L1 claim.
The client disconnects after seal	After the operator durably records the signed transaction, its receipt survives a disconnect. A receipt promises a SUCCEEDED, REVERTED or REJECTED outcome by an L1-block deadline and gives finite omission recourse; it does not by itself guarantee proof, finality, payment or notional coverage. Proof-enforced beneficiary credit supplies the payment link.
The operator or prover stalls	Admissible forced transactions are injected before seal. Each must execute, revert or receive a proof-derived rejection; once an unresolved request is overdue, room progress stops. For calldata rooms using the Ethereum state commitment, independent tooling rebuilds genesis, deposits, imports and batches from L1, then composes, proves, submits and claims. A recorded all-services-off exercise remains a deployment qualification gate.

II.4 Physical release is a separate finality gate

Finality state	Permitted external effect
Room result	Fast and provisional. The app may quote or simulate, but should keep positions capped and rollbackable.
L1 accepted	Proof-backed, but the accepting Ethereum block can still reorg. Payment, title and physical delivery remain pending.
Ethereum finalized	The provider may act when the finalized checkpoint proves both the application transaction and the proof-enforced beneficiary credit.

III MEV POSTURE FOR APPLICATION DEVELOPERS

Prove the market rule you actually choose

Base rule: Today, the engine orders transactions by when they reach the operator: FIFO by gossip arrival. That order is neither globally fair nor private. An application can instead pin its own ordering or clearing rule, and the accepted proof can check that the room followed it.

Where zkdeal differs: Applications on general-purpose ZK L2s can also build auctions, commit–reveal schemes and protected intents. zkdeal's distinction is packaging the selected rule, room-scoped accounting, recovery and explicit close into one isolated lifecycle.

III.1 Who can see an order before the seal

Submission pattern	Visible before seal	Protection and remaining exposure
Plaintext signed intent	The operator receives the price, size and direction in plaintext. In the shipping P2P mode, room members or the relay may also see them.	The proof checks the signature, signed limits and pinned execution policy. It does not prevent censorship before admission, a reactive order that the policy permits, or an external hedge on L1 or a CEX.
Commit–reveal	Before reveal, observers see the commitment and metadata, not the committed price and size. The participant later opens the commitment.	If the application seals the eligible set and order before any reveal, the proof can reject insertion based on revealed content. The application must still define cutoff, metadata and missing-reveal rules.
App-level ZK or threshold encryption	An application-level proof reveals only selected facts. Threshold encryption exposes ciphertext until the decryption keys are released.	zkdeal supports application-specific proofs. Threshold-encrypted room order flow is not a current base feature and introduces separate key-release, availability and disclosure assumptions.

CROSS-DOMAIN BOUNDARY A pinned batch rule can remove arrival-order games inside the room after the eligible set is sealed. It cannot stop an operator or participant who learns that set from trading on another chain or centralized venue before the clear. A shorter reveal-to-clear interval and an external-market policy can reduce this exposure, but cannot remove it.

III.2 How common MEV approaches differ

Approach	Primary job	What it does not solve alone
CoW-style fair combinatorial auction	Match signed constraints through competing solver solutions and pair-specific uniform clearing prices.	Encrypted submission, guaranteed fills, reserved capital or Ethereum finality.
Shutter threshold encryption	Hide order content until inclusion/order is committed and a Keyper threshold releases decryption material.	Price discovery, fills, liquidity or timely key release without additional assumptions.
FCFS sequencing	Order by receipt at one sequencer; simple and low latency.	Global arrival fairness, privacy or immunity from network-location races.
Timeboost-style priority	Auction a time advantage and redirect some ordering value to the chain.	MEV elimination or fair price formation; priority remains the product.
Private order flow / OFA	Keep orders out of the public mempool and optionally auction backrun rights.	Cryptographic secrecy from recipients, guaranteed inclusion, liquidity or settlement.
zkdeal room	The application pins the ordering, clearing, collateral, timeout and close rules; the proof checks them for one bounded episode.	Encryption, fair ordering and censorship resistance are not automatic. The application must encode the rule and provide the required infrastructure.

III.3 Cutoff, censorship economics and capital

Cutoff control: The application should tie the seal to L1 block N or another authenticated L1 signal. If the operator may choose any instant inside a wide window, that choice becomes a timing advantage.

Forced route: In validity-only mode, the application chooses a maximum admission window within protocol bounds, and each user chooses a deadline inside it. A signed room transaction can be queued on L1 without operator permission. The room cannot advance past an overdue request unless the proof shows it executed, reverted or was deterministically rejected. A live replacement-operator exercise remains qualification evidence.

Bond versus notional: An admission receipt makes omission challengeable against the service bond; it does not insure the trade value. The venue should size the penalty to its failure exposure and use the forced route as a separate path for progress.

Capital isolation: Each room keeps separate accounting, which limits the blast radius of a failure. The base contracts provide no cross-room netting. Market makers must preposition funds or add a reserve, credit, routing or rebalancing layer. Recursive aggregation shares proving and submission overhead, not balances.

III.4 Developer path and operating model

Self-operated path: Developers can run a room operator or node, prove room checkpoints and submit them to L1 permissionlessly. For calldata rooms using the Ethereum state commitment, the fallback rebuilds the opening state, policy, deposits, authenticated imports and accepted batches from L1, then composes, proves, submits and claims without hosted credentials. Imports require historical Ethereum state proofs; sparse-state reconstruction is not yet provided.

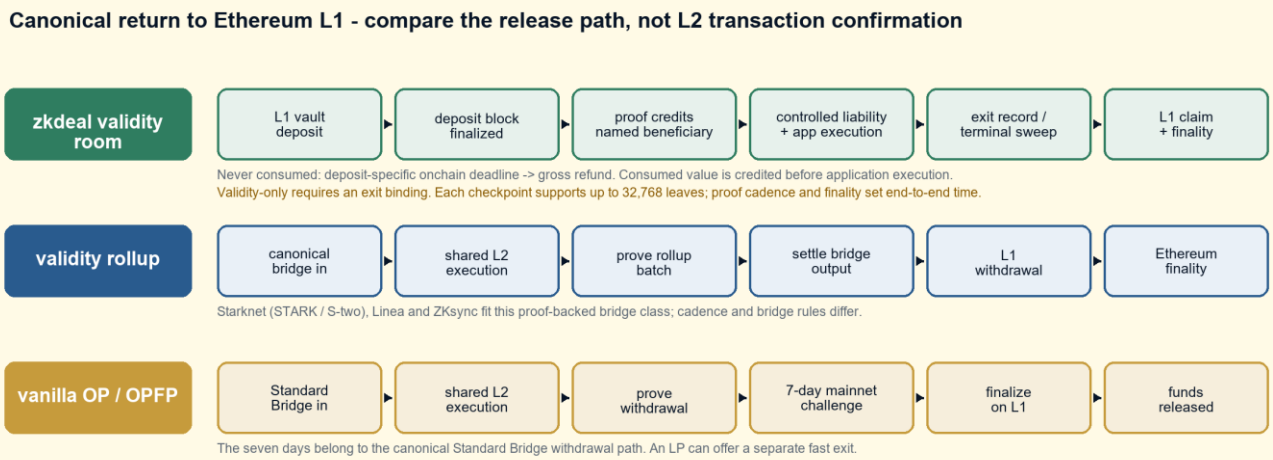
Hosted path: Hosted validity rooms inject admissible forced inputs and apply proof-bound deposit credits before the first application transaction. The host, execution engine and proof therefore start from the same credited state. The application still selects its admission clocks, service bond, omission penalty, checkpoint rhythm and market policy.

Signatures and policy: Customers sign EVM transaction envelopes with room-specific replay protection. Checkpoint approvals and admission promises use EIP-712 typed data. The application can pin permitted targets, selectors, call types, storage namespaces and resource limits.

IV CAPITAL MOVEMENT • CANONICAL EXIT PATH

Three ways capital returns to Ethereum L1

Comparison lens: A fast L2 confirmation and a fast canonical L1 release are different things. The lanes below compare the point at which Ethereum can release the asset without relying on a third-party liquidity provider.



ONE SELECTABLE L1-FUNDED VALIDITY PATH The application chooses the assets, beneficiary, admission window, checkpoint rhythm and exit allocation. Once this path is selected, the consuming proof must credit the beneficiary before application execution, preserve liabilities and follow the room-bound exit route. A checkpoint can carry up to 32,768 withdrawal leaves.

IV.1 Read the accounting states correctly

Term	Meaning
PENDING	Queued L1 escrow. If no proof consumes it, the depositor may reclaim it after that deposit's onchain deadline.
CONTROLLED	A proof consumed the deposit; the room's proved accounting now controls its allocation.
CLAIMABLE	An accepted validity-only proof moved value into a withdrawal root reserved for an L1 recipient.
PAID	The recipient claimed on L1. Ethereum finality then makes that release canonical.

IV.2 What each path means for liquidity

zkdeal L1-funded validity room: On the L1-funded validity path, assets stay in the RoomVault and remain segregated by room and asset. A deposit names its beneficiary and, until consumed, carries its own onchain refund deadline derived from the room's admission window. Consumption credits the beneficiary before application execution. The application's bound exit policy maps proved balances and exit records into L1 claims; code-free residual accounts can be swept at close.

Validity rollup—including Starknet: Starknet uses STARK-based proofs, with S-two in its prover stack; validity rollup describes its settlement and exit architecture. The canonical bridge moves value into a continuing shared L2, and a validity proof supports the output used by the bridge. There is no optimistic dispute window after proof acceptance; actual release still follows that rollup's proof cadence, bridge design and Ethereum finality policy.

Vanilla OP / OP fault proofs: Optimism documents typical Standard Bridge deposits from Ethereum in 1–3 minutes. The canonical return to Ethereum waits seven days on mainnet so an invalid withdrawal can be challenged. A fast bridge can advance liquidity sooner, but that is a separate liquidity-provider service with its own price and trust assumptions.

Exit path	Components of release time
zkdeal unconsumed refund	**deposit-block finality + deposit-specific onchain deadline + refund inclusion/finality**
zkdeal validity-only claim	**deposit finality + proof credit + exit record or terminal sweep + accepted root + claim + finality**
vanilla OP Standard Bridge	**output / withdrawal proof + 7-day challenge + L1 finalization**

IV.3 Speed claim and measurement boundary

Architectural advantage: The validity path uses proof acceptance rather than an optimistic challenge waiting period, while custody and accounting remain on L1.

Measurement boundary: Removing an optimistic challenge wait does not by itself guarantee a faster deposit or a seconds-level withdrawal. Teams should measure deposit finality, proof generation, checkpoint inclusion/finality and claim inclusion/finality separately.

IV.4 Application-facing liquidity claims

PROGRESSIVE RELEASE: A validity-only rolling room can publish withdrawal roots before close; closure is not the first opportunity to claim.

PENDING REFUND: A deposit never consumed by a proof can return after its deposit-specific onchain deadline, even while the room continues.

PROOF-GATED EXIT: That canonical claim path waits for proof acceptance and Ethereum finality rather than an optimistic challenge period.

PROVED CLOSE: A validity room may close through a proof-bearing batch. Its close sweep covers code-free accounts and up to 32,768 leaves; contract balances still use the room's exit records.

V GENERAL-PURPOSE L2 PROOF BASES

How zkdeal compares with shared-L2 proof stacks

V.1 Comparison scope

Comparison unit: This table compares proving and validation technology, not individual chains. Vanilla OP appears once as the optimistic benchmark. A proof family does not by itself determine data availability, sequencing, governance or exit rules.

Underlying technology	Used by L2BEAT-listed systems (examples)	Validation, data availability and canonical exit	Best fit and relation to zkdeal
zkdeal RISC Zero room STF + Groth16	Independent protocol • not yet L2BEAT-listed	Selectable room profiles. On the Stage-oriented rail: a validity proof gates every checkpoint, Ethereum calldata binds genesis and transitions, and the selected exit policy turns proved balances into L1 claims.	A bounded EVM workflow whose application policy, proof identity, clocks, assets and close are bound at room opening. It can begin in, return to or coordinate with a larger L2 ecosystem.
OP Stack + OPFP vanilla optimistic	OP Mainnet • Ink • Unichain	Interactive fault proofs; Ethereum data availability; Standard Bridge withdrawal waits seven days on mainnet.	Best for continuous public execution and shared-chain composability; liquidity depth is deployment-specific. zkdeal complements it when one finite episode selects proof-gated room settlement and room-scoped L1 accounting.
SP1 Hypercube / Turbo OP Succinct validity	Mantle • Katana; SP1 also participates in Taiko	In full-validity mode, SP1 proves each accepted OP/EVM state update; Ethereum data availability; proof-backed bridge exit. OP Succinct can also expose an owner-controlled optimistic fallback.	A strong fit for a full OP-compatible validity chain. zkdeal applies a smaller room lifecycle around one workflow.
SP1 ZK fault proofs OP Succinct Lite / 1R	Base • Morph • Facet • Ethscriptions • Celo	ZK resolves disputes; the happy path remains optimistic, so canonical exits retain challenge mechanics. Celo also uses external data availability.	A strong fit when OP compatibility is primary and ZK accelerates disputes. zkdeal requires validity at every accepted room checkpoint.
RISC Zero zkVM Kailua / multiproof	Taiko • BOB • Ronin; zkdeal uses RISC Zero too	General RISC-V execution proofs; a deployment may use full validity or ZK fault proofs; data availability and exit follow the host rollup.	The same prover family supports different products: zkdeal proves a certified room journal; these integrations prove a continuing chain.
Linea prover	Linea	EVM-specific validity proof; Ethereum data availability; persistent rollup bridge.	Linea serves public EVM composability. zkdeal adds task-scoped state, constrained calls/storage, L1 escrow and terminal close.
S-two	Starknet • Paradex	Cairo-focused STARK validity proofs; persistent rollup state and bridge; Ethereum settlement.	S-two now appears in Starknet and Paradex. It is a strong fit for Cairo-based ecosystems; zkdeal is EVM-room oriented.
Boojum / ZK Stack; legacy ZKsync Lite stack	ZKsync Era • Abstract • Sophon • Lens • Cronos zkEVM • zkCandy	Separate Matter Labs validity stacks: Boojum for ZK Stack and Era-era chains; legacy custom circuits for ZKsync Lite. Ethereum data for rollups or external data for validiums; exit follows the deployment.	A strong fit for launching a ZK Stack chain with shared tooling. zkdeal offers a bounded room with per-workflow policy and close.
OpenVM	Scroll	RISC-V zkVM-based EVM validity proof; Ethereum data availability; persistent rollup bridge.	Scroll serves shared EVM execution. zkdeal focuses on a room-chosen, proof-pinned workflow policy, room-scoped liabilities and a deliberate end state.
Airbender	ADI Chain	RISC-V validity proof; Ethereum data availability; continuing L2 state and bridge.	A strong fit for a policy-aligned general chain. zkdeal makes policy and decommissioning specific to each room.
zkProver	Silicon • Wirex Pay Chain	Validity proof with external committee data availability in these deployments.	A strong fit for continuing application chains using committee availability. zkdeal's current Stage-oriented deployment uses Ethereum calldata; blob modes remain options on the broader deployment rail.

DECISION GUIDE Choose a shared L2 for always-on public execution and ecosystem-wide composability. Choose or add zkdeal for a finite episode with isolated policy, app-defined clearing, room-scoped L1 accounting and explicit close. Depositors should name the correct beneficiaries, and the application should rehearse the standalone fallback path.

VI APPLICATION-SPECIFIC AND PRIVACY PROOF BASES

How zkdeal differs from specialized validity systems

VI.1 Specialized systems in the L2BEAT landscape

Scope: These application-specific exchange, payment and privacy systems are closer to zkdeal in product scope than generic chains, but differ in state lifetime, data availability, custody and privacy.

Underlying technology	Used by L2BEAT-listed systems (examples)	Validation, data availability and canonical exit	Best fit and relation to zkdeal
zkdeal RISC Zero room STF + Groth16	Independent protocol • not yet L2BEAT-listed	Selectable room profiles. On the Stage-oriented rail, proof-backed checkpoints bind Ethereum data, beneficiary credit, room liabilities and the application's exit policy.	Best for bespoke, bounded EVM episodes that choose their market policy, clocks, assets and close without creating a permanent venue.
Stone / StarkEx family	Legacy StarkEx-family deployments	Application-specific STARK validity proofs; data availability and exit depend on the deployment, including validium committee models.	A strong fit for specialized exchange lineage. zkdeal generalizes the bounded-application idea to policy-pinned EVM rooms with explicit room liabilities and close.
Lighter custom prover	Lighter Exchange	Application-specific validity rollup; Ethereum blobs/state deltas; exchange-oriented exit model.	Lighter is purpose-built for exchange execution. zkdeal targets bespoke multi-step EVM workflows beyond a single exchange design.
INTMAX	INTMAX Payments	Application proof stack with self-custodied data; payments and private-transfer focus.	INTMAX is purpose-built for network-wide private payments. zkdeal focuses on L1-custodied assets, EVM workflow policy and application-selected privacy.
Barretenberg / Aztec	Aztec	Privacy-first validity rollup with private/public state and the Aztec Virtual Machine; Ethereum settlement.	Aztec is the natural fit for general private smart contracts. zkdeal focuses on bounded EVM rooms and uses privacy only where the application adds it.

VI.2 Starknet, S-two, STARKs and validity rollups

PROOF SYSTEM VS SETTLEMENT MODEL Starknet is a validity rollup; S-two is the STARK proof system in its prover stack. ‘Validity rollup’ describes how Ethereum accepts state updates, while ‘STARK’ describes the proof family. StarkEx systems also use STARK proofs, but can be application-specific rollups or validiums.

VI.3 How to read this comparison

Technology family: A proof system explains how computation is verified. It does not by itself determine data availability, sequencing, custody, governance or exits.

zkdeal comparison unit: The comparison uses the selectable validity-only + Ethereum-data + L1-vault profile. It is not a description of every zkdeal room; unanimous and blob-oriented profiles have different assurance boundaries.

VI STAGE LENS • ROOM-PROFILE CHOICES

Choose a room profile; stage the deployment

VI.4 Room choice inside a deployment envelope

TWO LAYERS OF ASSURANCE The deployment chooses the outer settlement and governance envelope. Inside it, an application chooses an allowed proof template, policy, data route, assets, clocks and exit behavior. Those choices are bound when the room opens, and the ZK proof enforces them at every accepted checkpoint. L2BEAT evaluates the deployed rollup and canonical bridge, not each room setting; a room is therefore Stage-aligned, not independently staged.

VI.5 Options, enforcement and Stage effect

Choice	What zkdeal enforces after selection	Stage effect
Deployment rail validity + Ethereum data, or open/mixed	The Stage-oriented manager admits only validity-proof rooms whose canonical data is in Ethereum calldata. The open manager retains consent, blob and aggregate options.	Only the concrete deployed manager is assessed. Weaker options on the open rail do not inherit another deployment's Stage.
Room-specific ZK template, policy and proof identity	The application selects a registered template and policy. Program identity, verifier, calls, storage, assets, capacity and close behavior are bound to that room.	Supports a narrow, reviewable trust surface. A new template must join the deployment's governed and reproducible proof inventory.
Data + reconstruction calldata or blob; Ethereum or sparse state	The Stage-oriented rail requires calldata. Independent reconstruction now rebuilds Ethereum-state rooms from L1 genesis, deposits, imports and accepted batches.	Calldata + Ethereum state supplies the current Stage 0 path. A separately assessed blob rail and sparse-state reconstruction are not yet provided.
Funding + exits beneficiaries, assets and exit mapping	Validity rooms require an exit-bearing template. A consumed deposit credits its beneficiary before app execution; an unconsumed deposit becomes refundable after its onchain deadline; proved exits become L1 claims.	Provides room-level accounting and exit controls needed for Stages 0 and 1. Unusual asset accounting needs a reviewed template.
Admission + forcing window, bond, penalty and user deadline	The application chooses the room window and accountability terms within protocol bounds. Any user can queue an L1 request; an overdue request must execute, revert, receive a proved rejection or stop room progress.	Supplies the Stage 1 censorship path for validity rooms. It is not available to unanimous consent rooms.
Recovery + governance clocks room timing and deployment delay	The room chooses inactivity, admission and confirmation clocks; the deployment chooses the timelock and council model. Protocol floors and ceilings prevent unbounded room settings.	The complete exit must fit before a non-council upgrade. Maximal room clocks can approach 21 days, so a seven-day governance delay does not fit every room. The application chooses shorter clocks or the deployment chooses a longer delay; the deployer demonstrates the selected timing envelope.

VI.6 Current readiness

STAGE 0-ALIGNED RAIL
In place: Rollup identity, validity-gated roots, Ethereum calldata, bound genesis and L1-only reconstruction are provided for the Ethereum-state room profile.
To qualify: The deployer selects the room profile to assess, deploys its manager, runs that configuration end to end, accepts live checkpoints and demonstrates clean reconstruction from L1. Blob and sparse-state profiles are assessed separately when selected.

STAGE 1-ALIGNED RAIL
In place: Forced requests, proved rejection, permissionless proof submission and claims, deposit refunds, exit-bearing templates and critical-role handoff are provided for validity rooms.
To qualify: First qualify Stage 0. The application selects the room clocks, and the deployer selects either a no-council delay long enough for a complete exit or a qualifying council path. For that configuration, the deployer demonstrates the full exit with all services off and provides governance and council evidence.

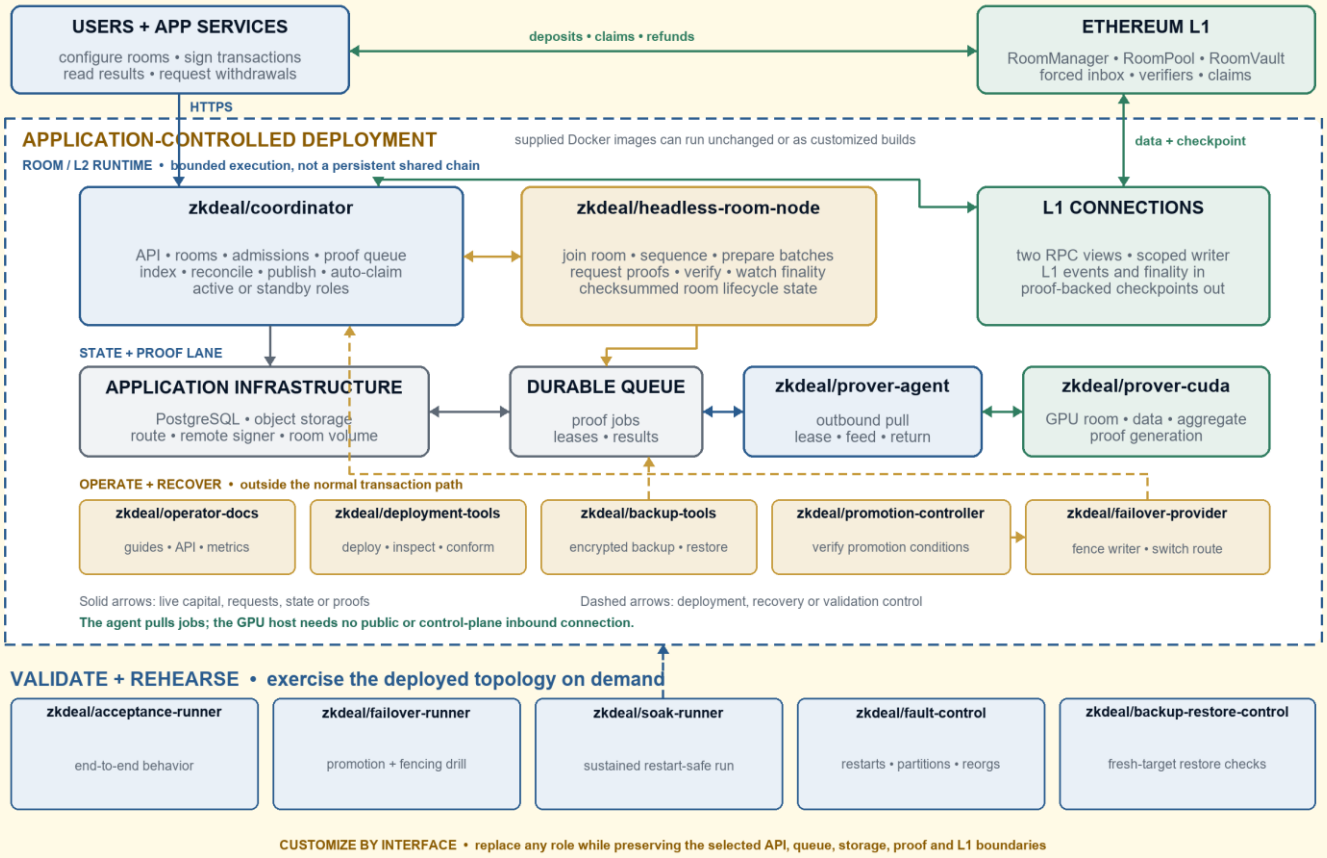
DEVELOPERS CHOOSE THE GUARD SET For a Stage 0-aligned deployment path, allow only rooms with validity-gated checkpoints, Ethereum calldata, bound genesis and L1 reconstruction. For Stage 1 alignment, also require forced requests with proved outcomes, permissionless proof submission and claims, deposit refunds, exit-bearing templates and compatible governance clocks. When those assurances are unnecessary, a developer can deliberately use a simpler room - for example, unanimous approval among known participants, participant-held data, or no canonical L1 asset exit. These options reduce operating and governance complexity. They remain proof-bound zkdeal rooms, but sit outside that Stage-aligned deployment path.

VII DEPLOYABLE DOCKER ARCHITECTURE • READY-TO-RUN IMAGES

Run the supplied Docker stack—or customize any role

zkdeal supplies separate Docker images for live room execution, operations and validation. Teams can run the default topology, co-locate roles for a smaller deployment, or substitute individual components while preserving the selected room and settlement boundaries.

VII.1 From an application request to Ethereum settlement



VII.2 Ready images by operating plane

RUN zkdeal/coordinator hosts the API and service roles; zkdeal/headless-room-node runs bounded room execution; zkdeal/prover-agent pulls proof work; zkdeal/prover-cuda generates room, data and aggregate proofs.

OPERATE + RECOVER zkdeal/operator-docs, zkdeal/deployment-tools and zkdeal/backup-tools support the deployment; zkdeal/promotion-controller and zkdeal/failover-provider verify a handoff, fence the old writer and switch the route.

VALIDATE + REHEARSE zkdeal/acceptance-runner, zkdeal/failover-runner, zkdeal/soak-runner, zkdeal/fault-control and zkdeal/backup-restore-control exercise normal operation, recovery, long runs, controlled faults and clean-target restore.

VII.3 Customize by interface, then revalidate the chosen profile

ONE STACK, DIFFERENT OPERATING SHAPES A small deployment can co-locate service roles. A production deployment can separate its public API, L1 publisher, active and standby coordinators, storage and GPU proving. Any supplied image may be configured, rebuilt or replaced. The replacement must keep the APIs, proof formats, queues, storage rules and L1 contract behavior required by that deployment's room profile; the validation images then exercise the customized stack.