

Cloud Computing Software: Selection, Requirements, and Workflow



- Software layer around cloud now decides cost, risk, and delivery speed



- Three governing questions: defensible selection, evaluable requirements, embedded workflow



- Category map: CMPs, FinOps, IaC/GitOps, observability, Kubernetes, CSPM/CNAPP, IDPs



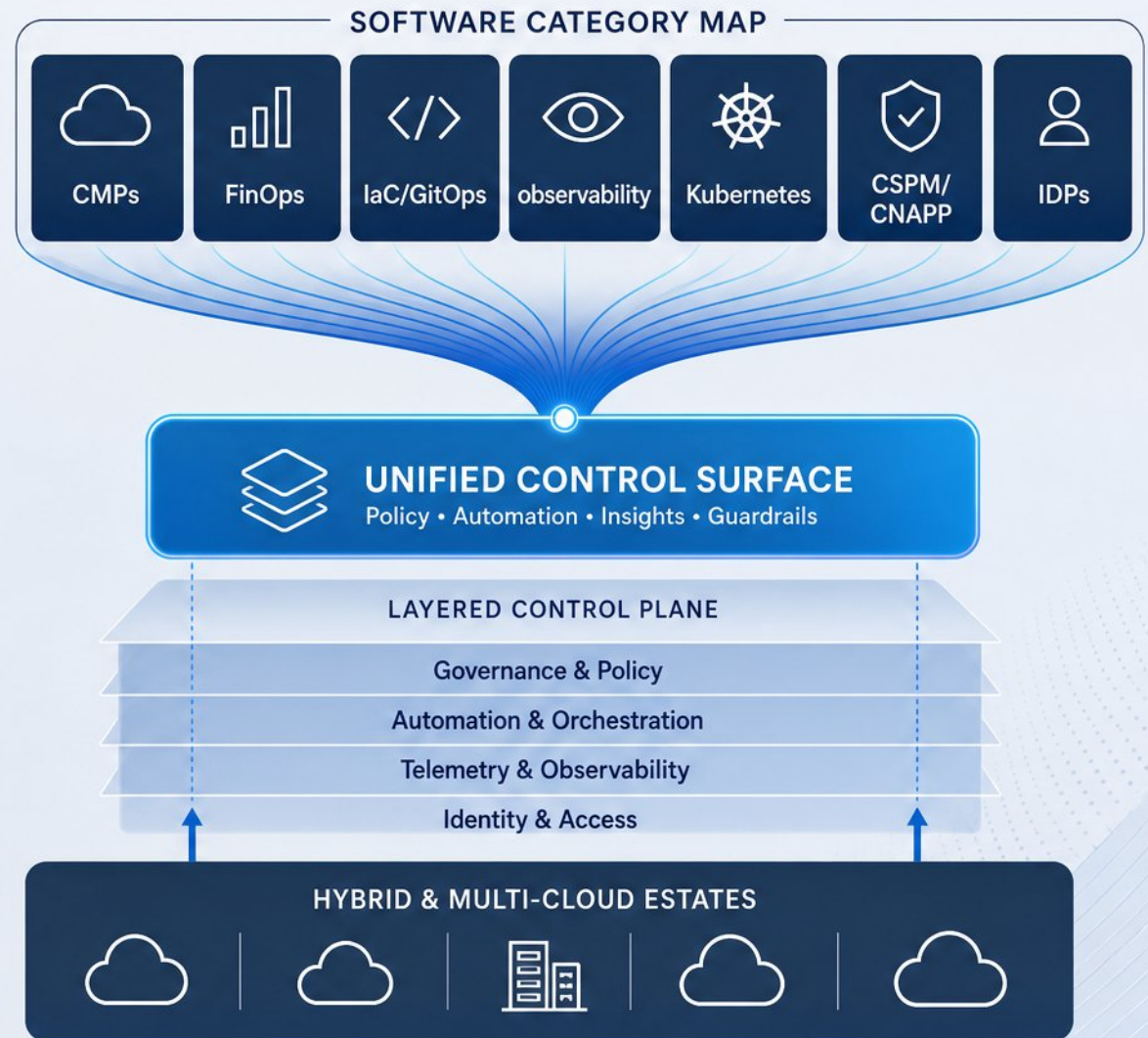
- Late-2026 context: governance toward \$18.7B by 2032; category boundaries converging



- Workflow dimension: day-1 provisioning, day-2 operations, cost governance, self-service

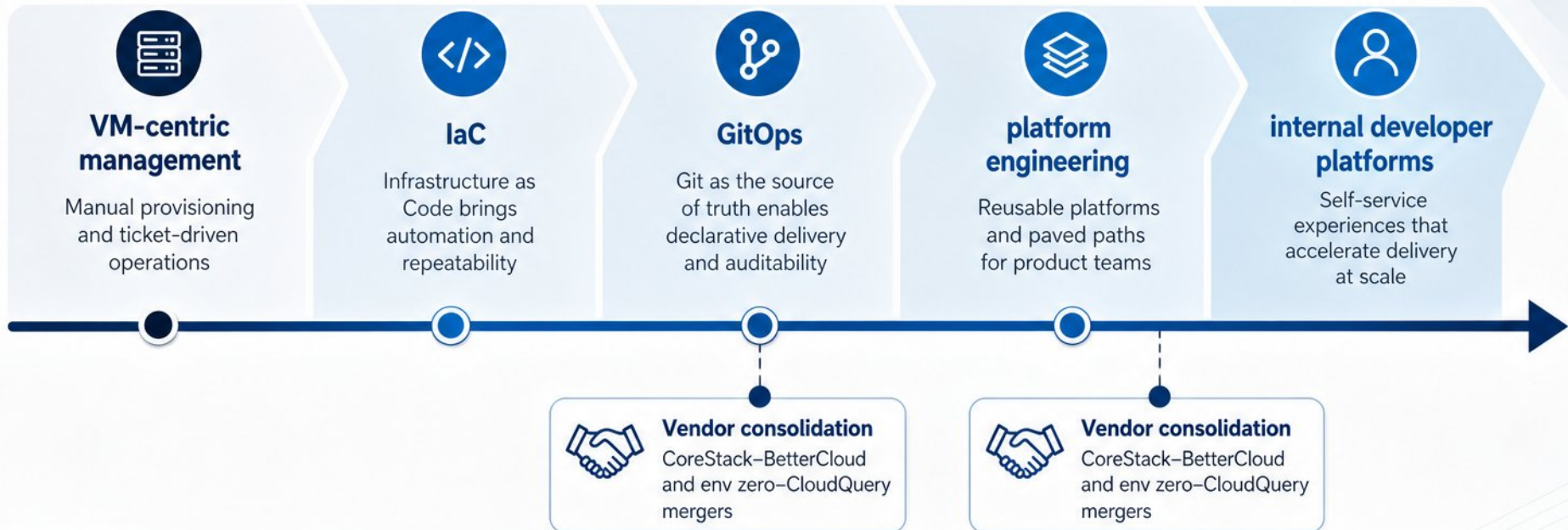


- Common failures: buying before requirements, over-weighting price, demo-anchored shortlists



Cloud Computing Software: Selection, Requirements, and Workflow

Background — How Cloud Tooling Reached Its Current Shape



- **Evolution:** VM-centric management → IaC → GitOps → platform engineering → internal developer platforms



- **Hybrid cloud dominates:** 73% of organizations, up 3 points year over year



- **Multi-cloud rises,** often from mergers and siloed apps — not deliberate strategy



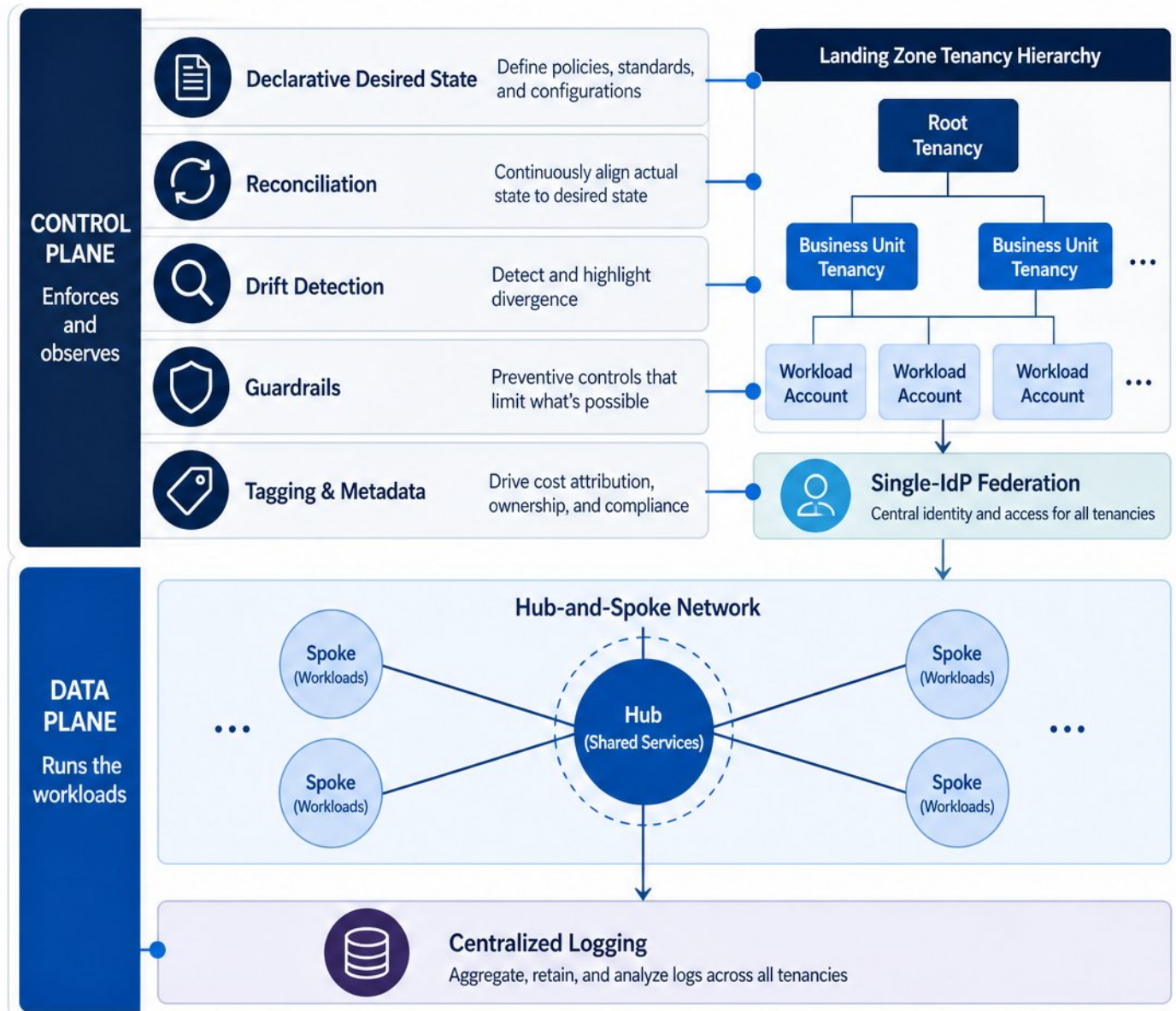
- **Vendor consolidation reshapes buying:** CoreStack-BetterCloud and env zero-CloudQuery mergers



- **FinOps is now first-class:** 63% have FinOps teams, 71% have cloud centers of excellence

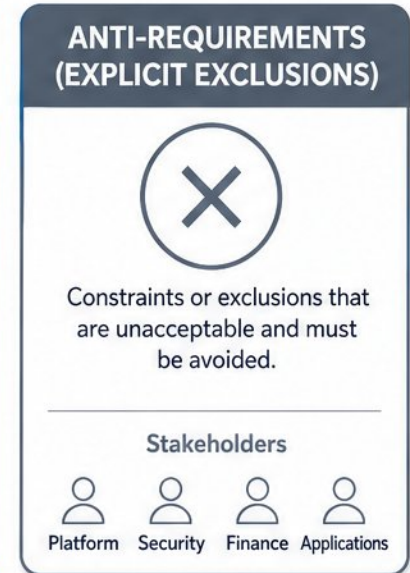
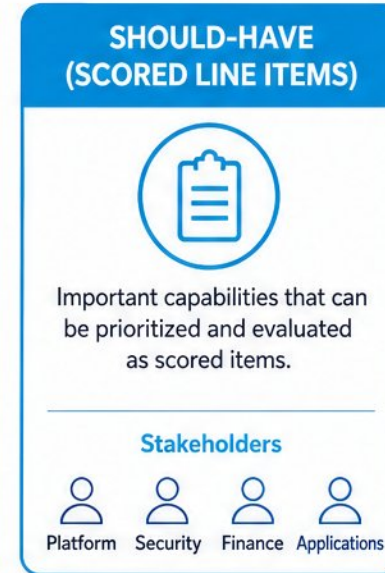
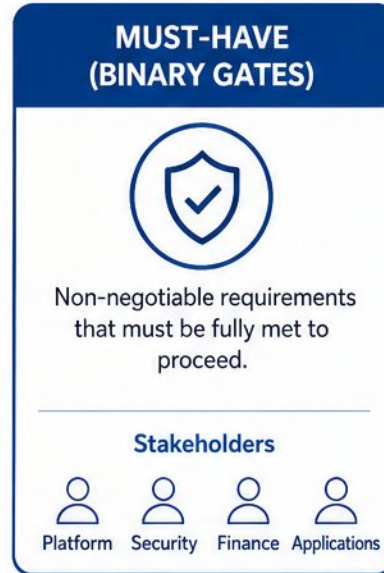
Core Concepts and Reference Architecture

- Control plane enforces and observes; data plane runs the workloads
- Declarative desired state, reconciliation, and drift detection catch divergence
- Preventive guardrails cap what is possible; detective controls catch the rest
- Tagging and metadata drive cost attribution, ownership, and compliance
- Landing zone: tenancy hierarchy, single-IdP federation, hub-and-spoke, central logging



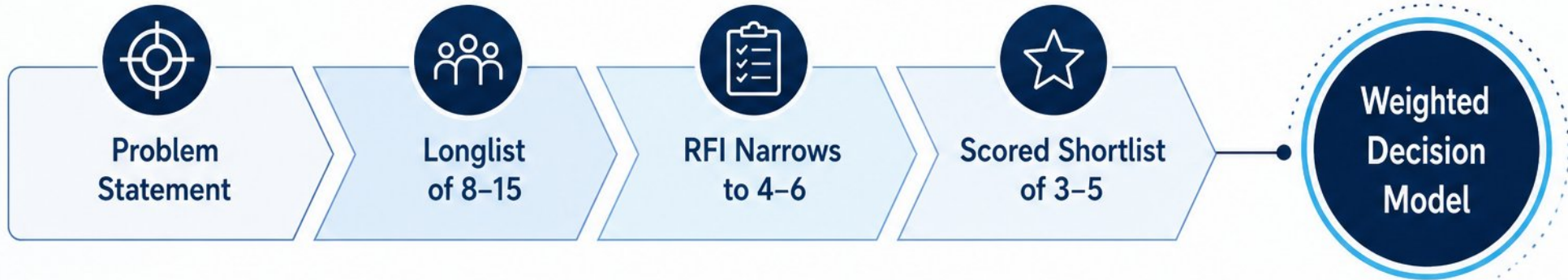
From Business Drivers to Evaluable Requirements

- ✓ Split requirements into must-have, should-have, and anti-requirements
- ✓ Must-haves are binary gates, never scored line items
- ✓ Elicit from platform, security, finance, and application teams
- ✓ Make asks measurable: per-resource allocation within X hours
- ✓ Test non-functionals: scale, latency, RBAC, tenancy, agent footprint
- ✓ Map residency to real data flows, backups, telemetry, subprocessors



WEIGHTED SCORECARD WITH EXPLICIT PASS/FAIL GATES				
Requirement Categories		 Must-Have (Binary Gates)	 Should-Have (Scored Line Items)	 Anti-Requirements (Explicit Exclusions)
 Platform		○	—	—
 Security		○	—	—
 Finance		○	—	—
 Applications		○	—	—

Selection Process: Longlists, RFPs, and Weighted Decision Models



- Phase the funnel: problem statement → longlist of 8-15 → RFI narrows to 4-6 → scored shortlist of 3-5

- Anchor on the problem statement: you are the problem expert, the vendor the solution expert

- Define the 1-5 scale with written anchors before seeing any vendor response

- Weight categories to 100%, lock weights, then run a $\pm 10\%$ sensitivity check

- Typical weighting: technical fit 30-40%, price and TCO 25-30%, vendor stability 15%, security 15%

- Enforce non-negotiables as pass/fail gates, not scored line items

- Score independently, resolve 2+ point deltas by structured discussion, log evidence everywhere

WEIGHTED SCORECARD FRAMEWORK				
Category	Weight (%)	1 – Poor (Anchor)	3 – Acceptable (Anchor)	5 – Excellent (Anchor)
Technical Fit	30–40%			
Price and TCO	25–30%			
Vendor Stability	15%			
Security	15%			
 NON-NEGOTIABLE PASS/FAIL GATES		Must meet all requirements to proceed. Fail any gate = do not advance.		
WEIGHTS SUM TO 100%				

Architecture and Integration Fit: What the Scorecard Must Test



- Versioned, documented APIs; data import/export without paid services



- Shift-left: IaC scan at PR, policy gate pre-merge, cost estimate on plan



- Confirm tenant isolation model and control-plane jurisdiction



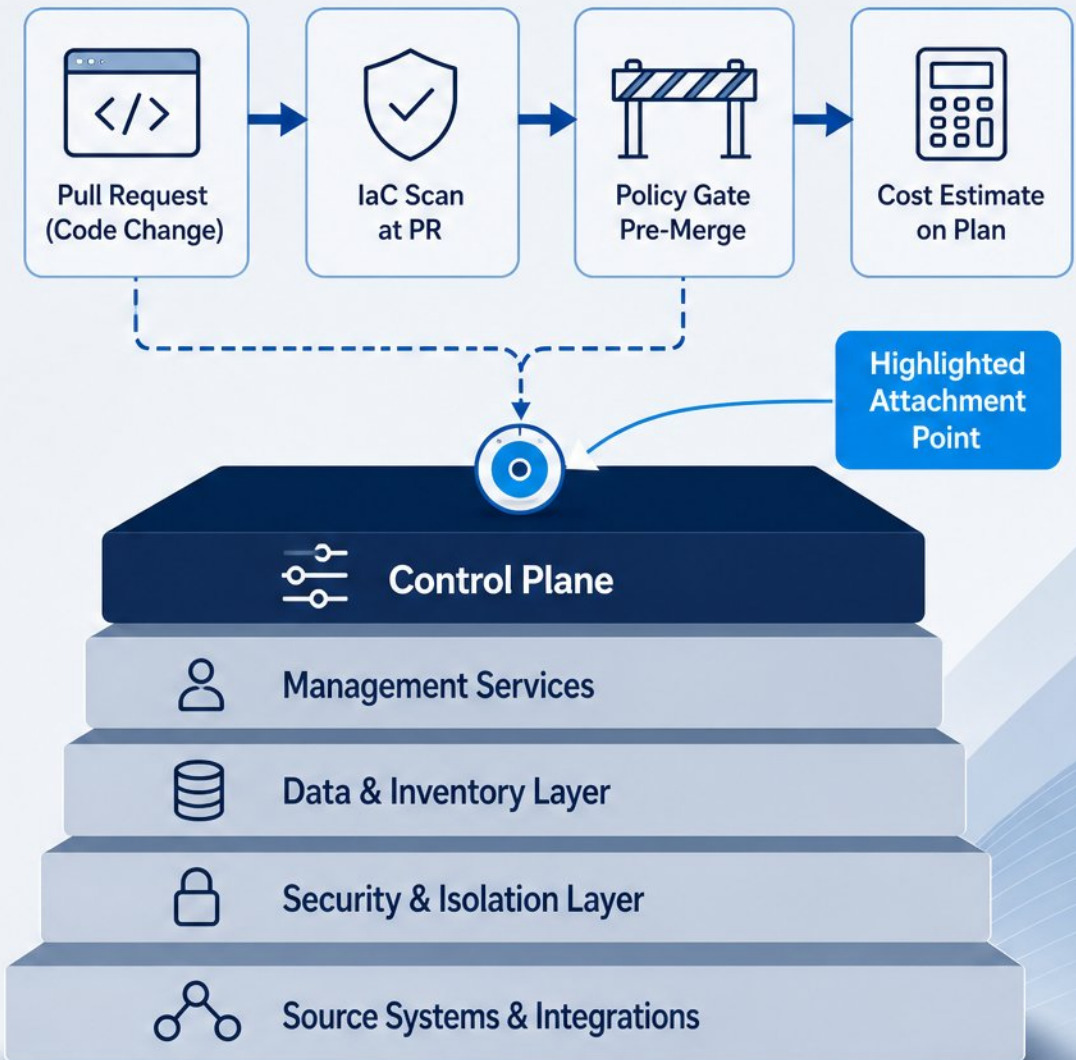
- Inventory completeness feeds every posture category: accounts, identities, SaaS



- Test scalability against real volumes, peak collection windows



- Edge-case handling: configurable in-house or vendor-dependent?



Evaluating Security, Compliance, and Sovereignty Fit

Security & Cloud Protection Layers

	CSPM	Cloud Security Posture Management — the floor.
	CWPP	Cloud Workload Protection Platform — workload defense layer.
	CIEM	Cloud Infrastructure Entitlement Management — identity and access layer.
	DSPM	Data Security Posture Management — data visibility and protection layer.
	CNAPP	Cloud-Native Application Protection Platform — bundles all the layers.

PREVENTIVE
POLICY



DETECTIVE
AUDIT

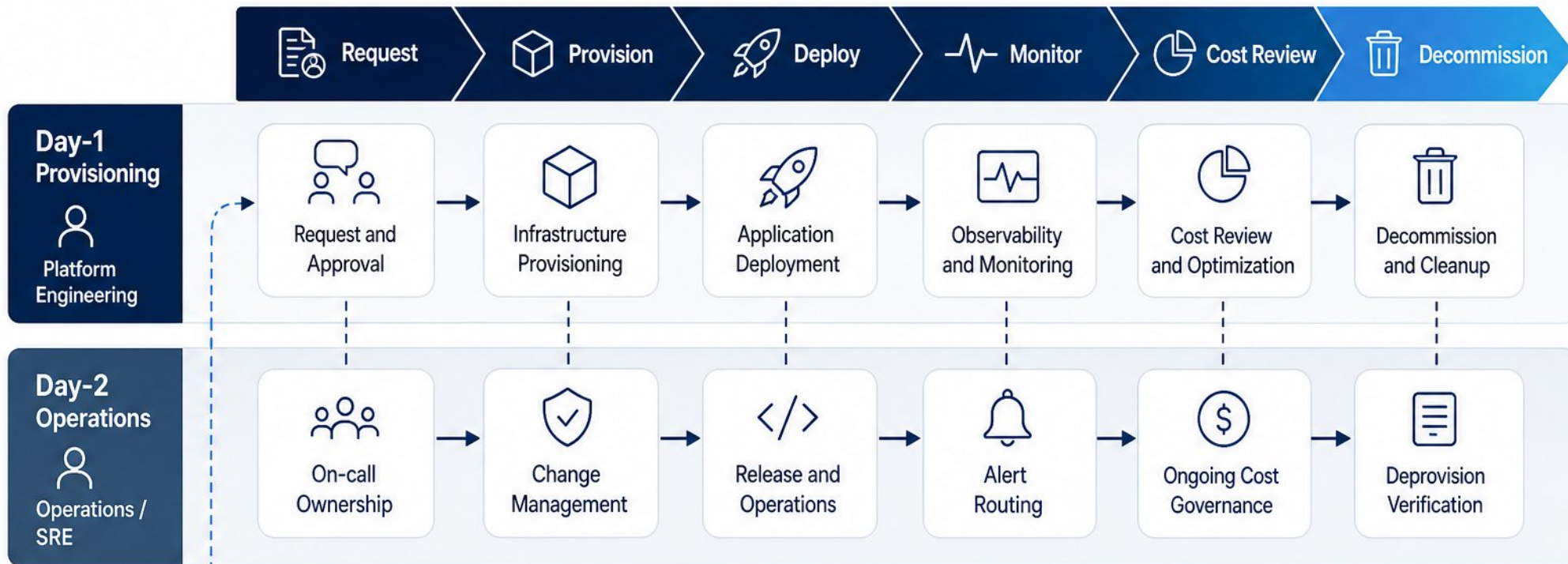
Data-Handling & Sovereignty Checklist

	Map data flows: telemetry, encryption, customer-managed keys, retention, deletion.	
	Review the tool itself: admin blast radius, SSO/SCIM, break-glass, support access.	
	Map data flows: telemetry, encryption, customer-managed keys, retention, deletion.	
	Sovereignty is a spectrum — match level to data classification, not marketing.	
	Contracts need re-evaluation triggers: ownership, jurisdiction, dependency changes.	

- CSPM is the floor; CWPP, CIEM, and DSPM are layers, CNAPP bundles them.
- Provider-native posture tools cover the baseline before you buy commercial.
- CNAPP is a market category, not a spec — evaluate against your weakest cloud.

- Review the tool itself: admin blast radius, SSO/SCIM, break-glass, support access.
- Map data flows: telemetry, encryption, customer-managed keys, retention, deletion.
- Sovereignty is a spectrum — match level to data classification, not marketing.
- Contracts need re-evaluation triggers: ownership, jurisdiction, dependency changes.

Workflow Integration: Day-1 Provisioning and Day-2 Operations



Golden Paths

Templates that ship repo, pipeline, and observability together

- **Map tool to stages:** request, provision, deploy, monitor, cost review, decommission
- **GitOps:** Git as source of truth; merge triggers automated plan and apply
- **CI gates:** lint, security scan, policy-as-code, cost estimate on pull request
- **Small blast radius:** one state file per environment per workload
- **Golden paths:** templates that ship repo, pipeline, and observability together
- **Day-2 readiness:** on-call ownership, alert routing, runbooks for drift failures

Cost, Chargeback, and FinOps Workflow

- Tag and attribute cost at provisioning, not after the invoice
- Wasted IaaS/PaaS spend rose to 29% in 2026, reversing five years of decline
- Showback by team, product, and environment — not one monthly bill
- Fewer than half use any discount program consistently; automate commitments
- Model AI and GPU spend separately: inference now outpaces training
- Track unit economics: cost per customer, transaction, and environment

FINOPS WORKFLOW



SHOWBACK VIEW



Visibility across teams, products, environments, and services

SPEND OPTIMIZATION FOCUS



AI & GPU SPEND

Model AI and GPU spend separately: inference now outpaces training



COMMITMENT COVERAGE

Fewer than half use any discount program consistently; automate commitments



Drive informed decisions and accountable spend

Commercial Evaluation: Licensing Models and Three-to-Five-Year TCO

PRICING MODELS



percent of spend



per-asset



per-user



per-account



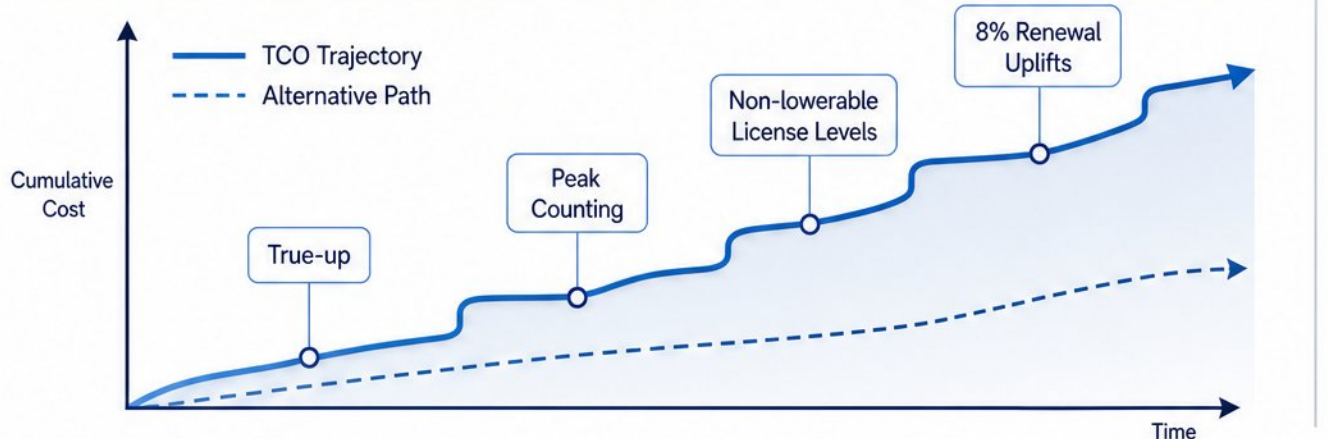
per-vCPU-hour



flat tiers



THREE-TO-FIVE-YEAR TCO (ILLUSTRATIVE CURVE)



- Pricing models: percent of spend, per-asset, per-user, per-account, per-vCPU-hour, flat tiers
- Percent-of-spend often meters GROSS list price before credits and discounts
- Ratchet mechanics: non-lowerable license levels, true-ups, peak counting, 8% renewal uplifts
- Build 3–5 year TCO: implementation, integration, training, admin overhead, switching cost
- Marketplace spend draws down AWS EDP, Azure MACC, GCP CUD — but consolidates dependency
- Negotiate: termination for convenience, uplift caps, true-down rights, data export, service credits

LICENSE LIFECYCLE



Assess Needs



Evaluate & Negotiate



Contract & Onboard

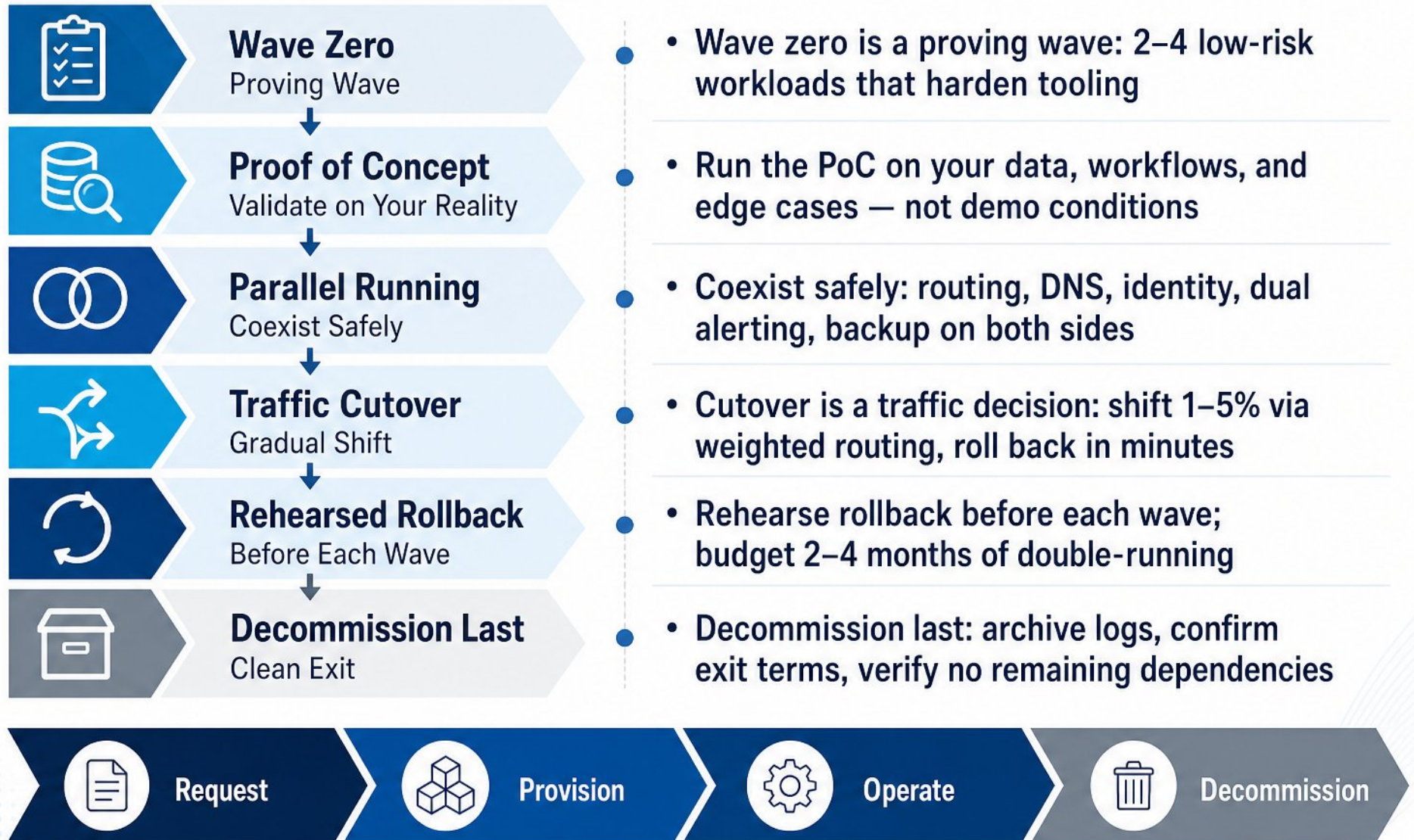


Operate & Optimize



Review & Renew
or Recompete

Pilots, Proofs of Concept, and Migration Without Breaking Delivery



Measuring Value, Governance, and Continuous Re-evaluation



- Define success metrics before go-live: provisioning time, spend allocated, drift incidents, MTTR



- Pair DORA delivery metrics with platform adoption metrics; DORA alone won't justify budget growth



- Translate engineering gains into financial terms: developer hours, incident cost, compliance speed



- Maintain a benefits register with baseline, mechanism, and attribution reviewed at 6, 12, 24 months



- Review quarterly by exception, timing each review 60–90 days before renewal notice



- Re-evaluate on triggers: scale thresholds, provider shifts, releases, pricing, ownership, restructuring

Quarterly Platform Review Scorecard

Delivery Metrics Paired with Benefits Register

DELIVERY METRICS		BENEFITS REGISTER		
		Baseline	Mechanism	Attribution
	Provisioning Time			
	Spend Allocated			
	Drift Incidents			
	Mean Time to Remediate (MTTR)			

PASS / FAIL GATES

	Value Realized and Attributable?		Risks Controlled and Compliant?		Future Fit and Still Relevant?
Proceed, Adjust, or Re-evaluate					

Putting It Together:

A 90-Day Action Plan and Course Artifacts



PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/17388d2d/public