ALSP Commercial Ontology and Engagement Economics

Engineering Requirements v0.1

Status: Draft for architectural review
Date: November 3, 2026
Working filename: ALSP_COMMERCIAL_ONTOLOGY_SPEC.md
Source foundation: *ALSP Commercial Models, Pricing Ontology, and Engagement Economics* (research report supplied in conversation; cutoff November 3, 2026)
Target implementation: Python 3.11+, Pydantic, deterministic calculation library, optional MCP interface
Scope: Commercial design, engagement evaluation, scenario comparison, and evidence-backed advisory reasoning. **Not** a general ledger, invoicing system, or automated contracting platform.

> **Evidence convention. [R]** means a concept documented or explicitly proposed in the supplied research report; it does not imply independently reverified external sources. **[D]** means a new design decision proposed in this specification. **[O]** means an open question requiring stakeholder validation. Illustrative values are not market benchmarks. The research report's source register remains the evidence source; individual claims must be linked to that register during implementation.

1. Purpose and intended decisions

The system SHALL represent the structure and economics of Alternative Legal Services Provider (ALSP) engagements in a way that is compositional, explainable, time-aware, and testable. It SHALL answer questions such as:

1. What exactly is being purchased and billed, and which contractual commitments modify it?
2. What is the expected provider contribution and client effective cost under alternative demand, complexity, staffing, and automation assumptions?
3. Which party bears volume, capacity, performance, scope, and technology-cost risks?
4. What changes if a fixed monthly service is replaced with a base-plus-overage or capacity-plus-consumption arrangement?
5. Which conclusions are contract facts, calculated results, assumptions, evidence-backed generalizations, or recommendations?

Success criterion [D]: A user can load a validated engagement definition, run reproducible monthly financial scenarios, compare at least three feasible commercial arrangements, and receive traceable explanations of economic differences without an LLM performing financial arithmetic.

2. Product boundaries

2.1 In scope for v1

- Typed representation of engagements, parties, services, priced units, fee components, rate schedules, commitments, entitlements, capacity, delivery topology, SLAs, adjustments, risks, and evidence.
- Monthly contractual-fee and cost simulation with annual and term-level aggregation.
- Time-indexed contract amendments and effective dates.
- Fixed, hourly/resource, output/unit, capacity, and base-plus-overage configurations; extensible support for other billing bases.
- Tiered and complexity-weighted rates; minimum spend; included allowance; overages; credits and collars where defined.
- Dedicated and pooled delivery costs, transition costs, technology costs, QA/rework, and constrained capacity.
- Deterministic calculations and scenario comparison; warnings for incomplete or inconsistent inputs.
- Provenance and a human-review boundary for recommendations.

2.2 Explicitly out of scope for v1

- Issuing invoices, receiving payments, journal entries, tax accounting, or ASC 606 / IFRS 15 accounting automation.
- Automated interpretation of signed contracts as legally authoritative facts.
- Legal advice or automatic determination that contingent/fee-sharing arrangements are lawful.
- Optimizing across an entire enterprise workforce scheduling problem.
- Predicting client churn or asserting market-standard margin targets without internal empirical data.
- Automatically negotiating or approving commercial terms.

2.3 Architecture principles

P-01 [D] Compositionality. An engagement may contain multiple fee components and modifiers. ManagedService is a delivery/obligation configuration, not a unique billing primitive.

P-02 [D] Typed semantics. Monetary values, percentages, periods, work units, resource units, and capacity units must be explicit. Do not mix document, contract, matter, or weighted_unit.

P-03 [D] Deterministic economics. LLMs may propose and explain arrangements; code computes prices, costs, and margins.

P-04 [D] Temporal integrity. Historical periods must use the contract version and assumptions effective for that period.

P-05 [D] Provenance. Source claims, observed client facts, internal policies, forecasts, and illustrative assumptions remain separate.

P-06 [D] Dual-sided evaluation. Assess provider economics and client consumption/value together. Unused commitment must not be labeled automatically as success or failure.

P-07 [D] Explicit uncertainty. Unknowns must not silently become zeroes, defaults, or fabricated market averages.

3. Domain model

3.1 Core aggregate boundaries

Aggregate / entity	Meaning	Core fields (illustrative)	Key relationships
Engagement	Evaluated client-provider relationship	id, name, currency, start_date, end_date, status	has parties, services, arrangement versions, scenarios
ContractParty	Buyer, provider, subcontractor	id, role, jurisdiction, regulated_status	participates in engagement; bears risk
ServiceOffering	Defined operational obligation	id, service_category, scope, unit_definition_id	has SLA, delivery model, priced units
WorkUnitDefinition	Unambiguous measurable output	code, name, counting_rule, acceptance_rule, exclusions	referenced by volume and output fee
PricingArrangementVersion	Effective-dated commercial configuration	id, effective_from, effective_to, supersedes_id	contains fee components, commitments, adjustments
FeeComponent	One independently computable charge	id, billing_basis, calculation_method, rate_schedule_id	prices service/entitlement
RateSchedule	Rates and tiers	currency, tiers, tier_method, effective_dates	applies to fee component
CommercialCommitment	Minimum spend, volume, reserved capacity, etc.	kind, amount, period, settlement_rule	constrains fee/capacity
Entitlement	What base/commitment purchases	unit, allowance, carryover_policy, expiry	consumed by recorded activity
AdjustmentMechanism	Overage, true-up, indexation, collar,	type, trigger, formula, priority	modifies consideration

Aggregate / entity	Meaning	Core fields (illustrative)	Key relationships
	credit		
DeliveryPlan	How work is produced	topology, resource_assignments, capacity_plan	drives cost and service capability
ResourceRole	Costed staffing role	role_code, loaded_cost_rate, availability	used in delivery plan
CostComponent	Direct or allocated cost	category, behavior, driver, allocation_policy	feeds cost engine
ServiceLevel	Measurable service obligation	metric, target, window, remedy_id	drives quality/credits
RiskAllocation	Identified exposure and bearer	risk_type, bearer, limit, mitigation	attached to arrangement
Scenario	Time-indexed assumption set	id, baseline_id, inputs, probability_label	evaluates arrangement
EvidenceClaim	Atomic proposition	claim_id, text, classification, scope	supported by evidence records
EvidenceRecord	Source and applicability	source_id, url, publication_date, retrieved_date, source_type	supports claim
CalculationResult	Reproducible output	engine_version, input_hash, periods, metrics, warnings	produced from scenario
PricingRecommendation	Advisory comparison, not contract fact	candidate_id, rationale, tradeoffs, assumptions	references results, evidence, approval

[R] The research proposes 34 candidate entity classes and 20 relationships. This table is a deliberately smaller v1 aggregate model **[D]**, not a claim that the remaining classes are unimportant. Concepts such as PerformanceObligation, ClientValueMetric, VolumeBand, AutomationComponent, and CommercialConstraint may initially be typed subobjects and later become independent graph entities.

3.2 Normalized billing bases

Use the report's ten bases as a controlled vocabulary **[R]**:

TIME, RESOURCE, CAPACITY, OUTPUT, USAGE, FIXED, RECURRING, MILESTONE, COST, OUTCOME.

[D] Important distinction: RECURRING describes an entitlement/cadence and can overlap with other priced objects. Preserve it as a source vocabulary value for compatibility, but model recurrence as an orthogonal charge_cadence property in the executable engine. Likewise, MILESTONE may be a payment trigger rather than the priced object. Do not force these categories into a mutually exclusive ontology hierarchy until semantics are reconciled.

3.3 Relationship requirements

- Engagement **has** one or more ServiceOffering objects.
- Engagement **has** one or more effective-dated PricingArrangementVersion objects; at most one version may be authoritative for the same service and time interval unless explicit component-level coexistence is defined.
- PricingArrangementVersion **contains** one or more FeeComponent objects.
- FeeComponent **references** exactly one primary calculation method and the required unit/rate definitions.
- Commitment **constrains** one or more fee components or entitlements, with an explicit aggregation boundary.
- AdjustmentMechanism **modifies** a named base amount or fee component in an explicit evaluation order.
- DeliveryPlan **drives** cost and capacity, not contractual revenue by default.
- Scenario **provides** assumptions for one or more comparable arrangements.
- Recommendation **references** calculation results and assumptions, never replaces them.
- EvidenceClaim **references** zero or more external sources and must indicate when it is inference or proposed design rather than externally observed fact.

4. Contract and fee semantics

4.1 Fee component contract

Each fee component MUST declare: id, service_id, billing_basis, charge_cadence, currency, calculation_method, quantity_source, unit, effective_period, and any required rate_schedule or fixed_amount. Missing required inputs must yield validation errors.

Initial supported methods **[D]**:

Method	Formula	Notes
Fixed recurring	fee_t = fixed_amount_t	Subject to effective dates and proration policy
Time	fee_t = sum(hours_role,t × rate_role,t)	Hours must be approved/eligible
Dedicated resource	fee_t =	Must define fractional FTE

Method	Formula	Notes
	sum(committed_FTE_role,t × rate_role,t)	and partial periods
Unit output	fee_t = sum(accepted_units_type,t × unit_rate_type,t)	Counting/acceptance rules required
Included allowance + overage	fee_t = base_t + overage_rate × max(eligible_units_t - allowance_t, 0)	Allowance and carryover ledger scoped to period
Cost-plus	fee_t = eligible_cost_t × (1 + markup)	Explicit cost eligibility policy required; post-v1 candidate
Outcome adjustment	fee_t = base_t + incentive(outcome_t)	Requires baseline, attribution, cap, approval; post-v1 candidate

4.2 Rate tiers

RateSchedule.tier_method MUST distinguish:

- MARGINAL: each tier prices only units within that tier.
- ALL_UNITS: the achieved tier sets the rate for all eligible units.
- GRADUATED_BLOCK: fixed price for defined blocks, if introduced later.

Tier boundaries MUST be nonoverlapping and unambiguous. Tier calculations MUST be tested at boundaries (threshold - 1, threshold, threshold + 1).

4.3 Commitments and entitlement ledger

A minimum-spend commitment is not equivalent to an included-unit allowance. The engine SHALL model separately:

- minimum_payable_amount and its settlement period;
- included_quantity and qualifying unit definition;
- consumed_quantity and remaining_quantity;
- rollover_quantity, expiration, and carryover ceiling;
- true_up and whether the minimum applies before or after discounts/credits;
- credit_application_order and prohibition on double counting.

A ledger entry MUST include period, unit, origin, consumption, expiry, and contract version. **[O]** Treatment of unused prepaid entitlements for financial-statement accounting is explicitly excluded from v1.

4.4 Adjustments and precedence

An arrangement MUST declare an adjustment order rather than relying on implicit business logic. Proposed default **[D]**, subject to contract override:

1. Compute base and metered charges.
2. Apply tier/complexity rate schedule.
3. Apply minimum commitment and qualifying allowances.
4. Compute overage and true-up.
5. Apply contractual discounts/rebates.
6. Apply performance credits/bonuses, subject to caps.
7. Apply currency conversion and taxes only in future modules.

This ordering is **not a universal commercial rule**. If the contract is ambiguous or terms conflict, return requires_contract_interpretation rather than guessing.

4.5 Amendments

An amendment MUST specify affected terms, effective date, precedence, and whether retroactive recalculation is authorized. Re-running a historical scenario SHALL preserve the original contract version unless an explicit amended-run flag is provided. Results MUST store an input/configuration fingerprint.

5. Economics engine

5.1 Time model

The base calculation grain SHALL be calendar month **[D]**, with aggregation to quarter, year, and contract term. Each month SHALL carry:

- contracted fee and earned contractual consideration (not automatically GAAP revenue);
- estimated billed amount and cash collections only if provided by an optional cash schedule;
- actual or forecast demand by unit and complexity;
- staffed and available capacity, productive hours, overflow and backlog;
- direct labor, pooled labor allocation, technology, third-party, QA/rework, management, transition and other engagement costs;
- gross profit, contribution profit, margin, and cumulative contribution;
- client paid amount, consumed units, and effective cost per accepted normalized unit.

The system MUST label contractual fee calculations as such; it MUST NOT call them recognized_revenue absent an explicit accounting adapter.

5.2 Cost classification

Cost components MUST declare:

cost_category, fixed_variable_or_step, dedicated_or_shared, cost_driver, cost_rate, allocation_method, effective_period, avoidability_class, source_type.

Required cost categories: dedicated labor, pooled labor, delivery management, QA/rework, AI inference, platform operations, third-party services, client-specific implementation, reusable engineering allocation, transition, and engagement-specific overhead.

[D] Define metrics using configured policies:

gross_profit = contractual_fee - direct_delivery_cost

contribution_profit = contractual_fee - direct_delivery_cost - engagement_avoidable_cost

gross_margin = gross_profit / contractual_fee

contribution_margin = contribution_profit / contractual_fee

Zero-fee periods MUST return an undefined/null margin with an explanation, not divide by zero. Corporate overhead allocations may be shown as a separate full-cost view; they MUST NOT be silently merged into contribution.

5.3 Capacity and operational production

Capacity MUST distinguish theoretical hours, planned available hours, safe service capacity, and delivered throughput. Demand above safe capacity SHALL either incur explicit overflow cost, create backlog/SLA risk, or fail scenario feasibility validation; it cannot be served at an assumed unlimited linear marginal cost.

For pooled resources, allocation policy MUST be declared and shared portfolio capacity MUST NOT be double-booked. v1 may model pooled capacity with a simplified aggregate capacity pool, but it must expose that simplification.

For AI-assisted services, cost drivers SHOULD include document complexity, inference volume, human validation time, exception rate, rework, and engineering amortization. **[O]** Correlated stochastic production drivers are a later-stage enhancement; initial v1 uses explicit scenario inputs and sensitivity bands.

5.4 Financial outputs

Required metrics: contractual revenue/fees, direct delivery cost, gross profit/margin, engagement-avoidable cost, contribution profit/margin, cost per accepted unit, effective client price per accepted unit, reserved-capacity utilization, productive-hours utilization, included-allowance consumption, commitment shortfall, overage revenue, cumulative contribution, transition payback month, and break-even volume when mathematically defined.

The engine SHALL separately label **provider-side financial outcomes** and **client-side consumption/value indicators**. It SHALL not infer retention probability from unused commitment.

6. Scenarios, comparison, and advisory behavior

6.1 Scenario definition

A scenario is an immutable snapshot of arrangement version, cost policy, demand profile, complexity mix, staffing/capacity plan, technology assumptions, and simulation horizon. Every assumption SHALL have value, unit, time_period, and origin (OBSERVED, CONTRACTED, ESTIMATED, ILLUSTRATIVE, INTERNAL_POLICY).

v1 SHALL support low/base/high scenarios and deterministic one-way sensitivity analysis. Probability distributions and Monte Carlo are deferred until calibrated data is available.

6.2 Candidate comparison

Given one service and demand profile, the engine SHOULD compare at least:

- fixed recurring managed service;
- dedicated resource/capacity model;
- base fee plus output-based overage.

Each comparison MUST show fee, cost, contribution, client effective unit cost, capacity utilization, risk-bearing changes, and assumptions that differ. A model can be rejected for infeasibility without being labeled commercially inferior.

6.3 Advisory guardrails

An AI layer MAY propose alternatives, explain tradeoffs, or identify missing facts. It MUST NOT:

- invent market rate cards, target margins, renewal probabilities, or legal permissions;
- treat provider case studies as independent market benchmarks;
- change financial inputs without showing the change;
- state that one structure is optimal absent a declared objective function and constraints;
- bypass legal/commercial approval requirements.

Recommendations SHALL cite calculation IDs, assumptions, constraints, and evidence claims. Human sign-off is required before adopting a commercial recommendation.

7. Evidence and governance

7.1 Evidence classes

Preserve research classifications **[R]**:

- A_DIRECT: directly documented claim;
- B_CORROBORATED: independently supported generalized claim;
- C_INFERENCE: analytical interpretation;
- D_PROPOSED: proposed mechanism not established as industry practice.

Evidence strength is independent of prevalence (ESTABLISHED_COMMON, DOCUMENTED_SPECIALIZED, EMERGING, UNVERIFIED). A claim may be direct but specialized.

7.2 Provenance schema

Each claim SHALL store: claim ID, atomic statement, classification, source references, applicability (provider, jurisdiction, service, date_range), date verified, reviewer, and caveats. Each source SHALL store publication date if known and date_unknown=true otherwise. Source URLs from the report MUST be independently checked before being marked verified; citation tokens in the pasted report are not durable identifiers.

7.3 Approval policy

The system SHALL support warnings and approval requirements for: contribution below internal policy floor, undefined pricing unit, missing commitment terms, ambiguous tier logic, unbounded fixed-fee volume, contingent fee or regulated legal-work structures, material uncapped SLA exposure, and missing evidence for asserted market benchmarks. Thresholds are configurable internal policy **[O]**, not embedded industry facts.

8. Data contracts and implementation shape

8.1 Proposed Python modules [D]

```
alsp_commercial/
  domain/          # Pydantic schemas, enums, units, contract versions
  pricing/         # fee components, tiers, commitments, adjustments
  economics/       # costs, capacity, period aggregation, margins
  scenarios/       # immutable scenario inputs, sensitivity, comparisons
  evidence/        # claims, sources, applicability, provenance
  validation/      # schema, semantic, policy, compatibility checks
  advisory/        # candidate generation and explanation contracts
  interfaces/      # CLI and optional MCP wrappers
  fixtures/        # report-derived illustrative engagement cases
  tests/           # unit, property, boundary, regression tests
```

8.2 Example engagement input [D, illustrative]

```
{
  "engagement_id": "EX-MANAGED-001",
  "currency": "USD",
  "start_date": "2027-01-01",
  "end_date": "2027-12-31",
  "service": {
    "id": "contract_review",
    "unit": "accepted_contract_review",
    "counting_rule": "One completed and accepted review of one eligible contract"
  },
```

```
 "arrangement": {
  "version": "1",
  "effective_from": "2027-01-01",
  "components": [
   {"id": "base", "method": "fixed_recurring", "amount": "100000", "cadence":
"MONTHLY"},
   {"id": "overage", "method": "unit_overage", "rate": "120", "unit":
"accepted_contract_review"}
  ],
  "entitlement": {"included_quantity": 1000, "period": "MONTHLY", "rollover": "NONE"}
 },
 "scenario": {
  "id": "base_case",
  "monthly_accepted_units": 1200,
  "origin": "ILLUSTRATIVE"
 }
}
```


Expected fee: $100,000 + 200 × $120 = **$124,000** for that month. This example deliberately omits costs and therefore cannot yield a margin. The final schema must make omitted optional sections explicit and reject incomplete requests for metrics requiring those sections.

8.3 Interface contracts

Initial CLI/API operations **[D]**:

- validate_engagement(input) -> validation_report
- calculate_periods(engagement, scenario, cost_policy) -> calculation_result
- compare_arrangements(engagement, candidates, scenario) -> comparison_result
- explain_fee(calculation_id, period) -> line_item_trace
- check_commercial_constraints(engagement, policy) -> warnings_and_approvals
- list_applicable_evidence(claim_or_concept) -> provenance_records

MCP may expose these as read-only tools initially. Persistence, writeback, and approval actions are deferred.

9. Verification and acceptance tests

The test suite MUST include:

Test ID	Requirement	Expected behavior
T-001	Base plus overage	1,200 units, 1,000 included, $100k base, $120 overage → $124k
T-002	Minimum with zero	Contractual minimum

Test ID	Requirement	Expected behavior
	consumption	charged, zero effective unit cost denominator handled as undefined
T-003	Marginal versus all-units tiers	Different, correct totals at same threshold-crossing quantity
T-004	No unit mismatch	document volume cannot silently price contract rate
T-005	Contract amendment	Prior months retain prior terms unless retroactivity authorized
T-006	Rollover ledger	Credits are carried, consumed, and expired exactly once
T-007	No duplicate minimum	Minimum applied at declared aggregation level only
T-008	Zero fee	Margin null/undefined, never infinity or division error
T-009	Dedicated FTE low demand	Contract fee and dedicated labor cost remain fixed absent contractual/cost change
T-010	Capacity breach	Overflow cost/backlog/feasibility warning required
T-011	Evidence provenance	Inference is not serialized as externally verified fact
T-012	Reproducibility	Same inputs and engine version produce identical outputs/hash
T-013	Scope ambiguity	Missing unit counting rule blocks output pricing approval
T-014	Margin versus markup	$75 cost, $100 price → 25% margin and 33.33% markup
T-015	Annual vs monthly	Annual total equals sum of period-level contractual fees and costs
T-016	Report casebook	Reproduce six illustrative

Test ID	Requirement	Expected behavior
		case results under documented assumptions

Property tests SHOULD verify nonnegative accepted units, nonoverlapping effective dates, rate-tier continuity, and that a nonnegative overage rate never decreases the pre-credit fee when eligible consumption increases (holding other terms fixed).

10. Delivery milestones

Milestone	Deliverable	Exit criterion
M0 — Domain contract	Reviewed ontology v0.2, glossary, schema, unit rules	Stakeholders approve semantics for first three pricing configurations
M1 — Deterministic engine	Python package, validators, monthly calculations, tests	Cases A–F and boundary tests pass; outputs reproducible
M2 — Scenario comparison	Candidate comparisons, sensitivities, capacity warnings	Three arrangements evaluated against same demand profile
M3 — Evidence and advisory	Claim provenance, approval flags, MCP read-only interface	Explanations trace to formulas and evidence; no LLM arithmetic
M4 — Pilot validation	Anonymized real engagement(s), reconciliation report	Internal commercial/finance reviewers accept calculations and caveats

Do not advance from M0 to M1 before resolving the billing-basis and entitlement semantics. Do not claim M4 accuracy based only on illustrative casebook tests.

11. Open decisions for review

1. **[O] Commercial ownership:** Who owns the margin policy, exceptions, and approval thresholds?
2. **[O] Financial conventions:** What is included in gross margin, contribution, and shared delivery allocation?
3. **[O] Time and currency:** Is monthly granularity sufficient? Are multicurrency contracts in scope for pilot?
4. **[O] First engagement:** Which anonymized fixed-monthly managed service can serve as the pilot?
5. **[O] Volume semantics:** What counts as an accepted review/contract/matter? Who adjudicates complexity bands?
6. **[O] Capacity policy:** What productive-hour and safe-capacity assumptions are approved?
7. **[O] Contract versioning:** How are amendments and retroactive adjustments represented operationally?
8. **[O] Evidence review:** Who verifies source links and internal commercial practices?

9. **[O] Recommendation objective:** Optimize contribution, client value, stability, or a constrained multi-objective score?
10. **[O] Persistence:** Begin with versioned JSON fixtures or a relational database?

12. Decisions proposed for v0.2 review

- Adopt a **compositional PricingArrangement** instead of a flat enumeration of 34 pricing models. **[R/D]**
- Use **Pydantic domain models and pure Python financial functions** before adding a graph database. **[D]**
- Implement **monthly time-indexed simulation** before Monte Carlo optimization. **[D]**
- Keep **client value indicators** distinct from provider financial measures. **[R/D]**
- Require **contract facts, assumptions, evidence, and recommendations** to be explicitly labeled. **[R/D]**
- Use the research's **six illustrative financial cases as regression fixtures**, not as market benchmarks. **[R/D]**
- Expose **read-only MCP operations only after deterministic validation**. **[D]**

Appendix A. Research-to-implementation traceability

Research domain	Initial implementation location	Status
Ten billing bases and normalized pricing taxonomy	domain, pricing	M0 semantic normalization required
58 commercial levers	pricing, validation	Subset M1; remainder catalogued for expansion
35 financial metrics	economics	Priority subset M1; accounting metrics deferred
34 candidate entity classes	domain, evidence	Collapsed to core aggregates for v1
20 proposed semantic relationships	domain	Reconciled against typed schema in M0
Six illustrative cases	fixtures, tests	M1 regression baseline
Evidence register E001–E040	evidence	URLs/claims require verification before trusted use
Regulatory constraints	validation	Warning/escalation only; no automated legal determination
AI cost and automation benefit	economics, scenarios	Cost categories M1; sophisticated inference M3+
Adjacent-sector mechanisms	evidence, advisory	Design candidates, not universal ALSP practice

Appendix B. Explicit non-claims

This specification does **not** assert that any particular pricing structure is universally optimal, that a fixed margin target is industry standard, that unused commitments predict churn, that an ALSP can legally use every modeled fee type, or that the illustrative scenario prices reflect actual market rates. Such claims require evidence, internal policy, or legal review beyond the supplied report.