What Needs an Authority

Mechanical Detection, Chartered Resolution, and the Exact Price of Sole Ownership

Paper 6 of the Harbor program: the coordination functions of a multi-agent harbor, sorted into those an algorithm discharges and those a charter must pay for

Prepared for Erich Owens

August 2026

Abstract

Multi-agent governance documents hand out authorities freely: a *lookout* that scans incoming commitments for conflicts, a *sole owner* for the roadmap, the test suite, the release tag. This paper asks which of those coordination functions genuinely require a designated authority, and prices the ones that do. Two results. First, conflict *detection* needs no authority at all — inside a designed commitment fragment $\mathcal{L}_c$ (definite Horn rules plus $\bot$-headed integrity constraints, ground deontic obligation/prohibition rules over scopes and intervals, exclusive interval claims, difference constraints), detection is a polynomial, witness-producing algorithm — $O(H + T \log T + X \log X + V \cdot E)$ in the Horn program size H, the T fired deontic tokens, the X exclusive claims and the deadline graph (V, E) — validated against a brute-force oracle on 3000 random policy sets with zero disagreements; and composed with the supervisory-control theorem of Paper 2 it is not even a role, but a guard: acceptance is a mediated write, so in-fragment conflict-freedom is *regimentable* — the runtime refuses the commit. One expressive step outside the fragment (disjunctive obligations under discharge choice), conflict-freedom is NP-complete, and *there* an authority becomes necessary: not to notice the conflict but to choose the discharge. Second, sole *ownership* of a role is not a principle but an inequality: a sole specialist beats a pooled swarm exactly when the skill premium clears the Erlang-C boundary $g_A(\rho, c)$ — the whitepaper's proposed threshold is falsified in both directions, with the two thresholds crossing at $\rho = (3 - \sqrt{5})/2$ — and a sole owner who can die and be succeeded can be rescued by *some* skill premium only while the death-rate–times–succession-time product stays below $D^\star = \eta K/(1 - \eta K)$, where K is the pool's cost line (its mean wait plus the accountability credit) and the threshold exists only when $\eta K < 1$: past that line, pooling wins even against an infinitely skilled specialist. The slogan the two theorems earn: an authority is needed exactly where the algorithm ends, and where it is needed, its price is a computable number, not a vibe.

Express lane. *Conflict detection inside the designed commitment fragment is polynomial and witness-producing (no judge consumed, and via Paper 2 the check is enforceable pre-commit by the runtime); one disjunctive obligation outside the fragment makes conflict-freedom NP-complete, which is where a chartered resolver must choose; and sole ownership of a role pays iff* $\mu_s/\mu_g \geq g_A(\rho, c)$ *— exact Erlang-C form in the box of §5 — with sole-owner viability capped at* $\xi/\eta < D^\star$ *absent skill limits.* Experts: read the two boxes (§3, §5), the succession box (§6), and the inventory table (Table 1); the rest is scaffolding.

1 The scene: the harbor's authority inventory

A harbor of coding agents runs on commitments: standing prohibitions (*never write to prod*), conditional obligations (*if you deploy, you must update the runbook within the window*), exclusive claims (*this file is mine from tick 5 to tick 20*), deadlines chained to deadlines. The governance document responds with authorities. A *lookout* is chartered to scan every proposed commitment

against the accepted set and reject contradictions. *Sole-responsibility roles* — a roadmap owner, a test-suite curator, a release avatar — each hold exclusive write capability over one invariant, on the theory that a single deeply-skilled owner beats a committee.

Both assignments feel natural, and both smuggle in an unexamined premise: that the function *requires a designated authority* — a mind with standing, judgment, and a name — rather than a subroutine and a price. The operations lead's two questions are therefore exact. *Does the lookout need judgment, or is contradiction-scanning mechanical?* And *when does the sole owner actually beat the pool — and what happens when the owner dies mid-quarter?* This paper answers both with theorems that were run before they were believed: the first by a 3000-instance oracle sweep with zero disagreements [internal, `b4_deontic_fragment.py`], the second by a sweep that *falsified the whitepaper's own proposed threshold in both directions* before replacing it [internal, `b8_specialization.py`].

The idea in one breath.

An authority is needed exactly where the algorithm ends — and where it is needed, its price is a computable number, not a vibe.

Three places, in this paper. Inside the fragment $\mathcal{L}_c$ the algorithm does not end, so no authority is consumed: conflict detection is a polynomial witness-producing decision the runtime can enforce pre-commit. One disjunction outside it the algorithm ends and a chartered resolver must choose the discharge, because conflict-freedom is NP-complete there. And sole ownership of a role is not an ending but a purchase, priced by an exact Erlang-C threshold on skill and capped by the owner's death-rate–times–succession-time staying below $D^\star$.

Intuition: the working port. Watch a real port for a day (Figure 1). Whether two vessels are on a collision course is *geometry* — constant bearing, decreasing range — computed identically by every watch officer; nobody's authority is consulted. Who gives way is next decided by a *rulebook* (the collision regulations) — still mechanical, still nobody's judgment. Only where the rulebook ends — a three-vessel crossing the rules never anticipated — does harbor traffic control exercise chartered discretion. And the port's one genuinely personal authority, the licensed pilot with twenty years on these shoals, is hired against a pool of tug crews only when the water is hard enough to pay the premium — and the port licenses a second pilot, because a port whose only pilot can fall ill has priced its channel's availability at one human body. The structural mapping: collision geometry → in-fragment conflict detection (mechanical, witness-producing); the rulebook's edge → the fragment's NP-complete frontier (where chartered resolution begins); the pilot's premium and the second license → the $g_A(\rho, c)$ boundary and the succession price $D^\star$. The analogy maps relations, so it earns a prediction: making the rulebook *richer* should eventually make rule-following itself intractable and reintroduce discretion — and that is exactly what the NP-completeness theorem says happens one disjunction past the fragment.

The predictable misread, preempted: *mechanical detection does not mean mechanical governance.* The checker certifies the norm *system*, never the norms — a bad policy consistently held is consistent — and detection without a resolution charter just produces alarms. The theorems do not abolish authority; they *relocate* it, from noticing to choosing, and then hand the remaining authority its bill.

2 The vocabulary of Part I

Part I needs two dictionaries a scheduler-builder may not have: deontic logic and complexity theory. Both are below, in the order the terms are needed; Part II's queueing vocabulary waits until §5, where it is used. Nothing here is deep; it is all notation someone else fixed decades ago.

Paper 6 — the authority inventory: where the algorithm ends, the charter begins

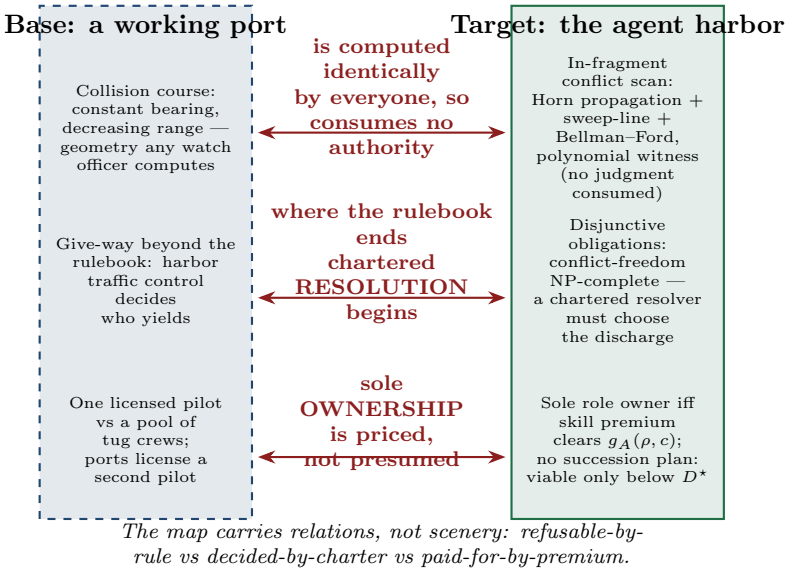

Figure 1: The relation map. Base domain: a working port — collision-course detection is geometry, give-way beyond the rulebook is harbor traffic control's chartered call, and the sole licensed pilot is paid a premium against a pool of tug crews, with a second license mandated. Target domain: the agent harbor — in-fragment conflict scanning is Horn propagation + sweep-line + Bellman–Ford with a polynomial witness; disjunctive obligations make conflict-freedom NP-complete, so a chartered resolver chooses the discharge; and a sole role owner is justified iff the skill premium clears $g_A(\rho, c)$, with no-succession viability capped at $D^\star$. The correspondence arrows carry the theorem content: refusable-by-rule vs decided-by-charter vs paid-for-by-premium.

Logic. *Deontic* logic is the logic of obligation and permission, as opposed to the logic of truth — it reasons about what *ought* to be, not what is. Its two operators here are O (obliged) and F (forbidden), and we apply them only to ground facts: an action, a scope, a time interval. A *Horn rule* is an implication whose premise is a conjunction of atoms and whose conclusion is a single atom — the shape almost every rules engine and every Prolog program already uses. A *definite* Horn program has no negation, which is exactly what makes it cheap: its consequences can be computed by repeatedly firing whatever rules are satisfied until nothing changes, and this process is guaranteed to terminate at one canonical answer. That answer is the *least model* M, and its defining property is the one Theorem 1a leans on: it contains precisely the facts forced in *every* model — and when $\bot \in M$ there are no consistent worlds at all, which is exactly the conflict Theorem 1a is built to report. Either way a status the least model assigns is not one possibility among several but a certainty. *Unit propagation* is the name for the firing procedure, and Dowling–Gallier [1] showed it runs in time linear in the program.

A *difference constraint* is an inequality of the form $x_j - x_i \le \delta$ over deadline variables — "the runbook update must land no more than six ticks after the deploy." A system of them is satisfiable exactly when its constraint graph contains no *negative cycle*, a loop of constraints whose weights sum below zero and which therefore demands that some event precede itself. Bellman–Ford finds such a cycle if one exists, and hands it over as the witness. This is standard temporal-reasoning machinery; §8 names the literature that owns it.

A problem is in **P** when some algorithm solves it in time polynomial in the input size — the practical meaning of "tractable at scale." It is **NP-complete** when it is in **NP** — a proposed solution can be *checked* in polynomial time — *and* every problem in NP reduces to it

in polynomial time, so a polynomial algorithm for it would put all of NP in P and settle the P versus NP question. The working consequence, which is all Theorem 1b needs: an NP-complete check cannot be run on every proposal at commit time as the policy set grows. *3-SAT* is the canonical NP-complete problem (is a Boolean formula in three-literal clauses satisfiable?), and a *reduction* from it is the standard way to prove a new problem at least as hard: encode any 3-SAT instance as an instance of yours, so that solving yours would solve 3-SAT.

A *witness* is the concrete object a checker returns alongside a yes — the derivation chain, the two clashing rules with their overlap, the negative cycle. Producing witnesses is a strictly stronger requirement than deciding, and it is the difference between a checker that says "rejected" and one a human can argue with.

3 Part I. The lookout is a subroutine: the fragment $\mathcal{L}_c$

Cheap detection is a *language-design* choice, not a discovery about deontic logic in general. The commitment language $\mathcal{L}_c$ buys it by construction. A policy set in $\mathcal{L}_c$ contains:

- **facts** (ground atoms), **definite Horn rules** $a_1 \wedge \cdots \wedge a_k \to b$ with b an atom, and **integrity constraints** $a_1 \wedge \cdots \wedge a_k \to \bot$; operationally $\bot$ is carried as a distinguished atom, so the program is definite over the atom vocabulary extended with $\bot$, and the least-model machinery of §2 applies unchanged — a derivable $\bot$ is then simply $\bot \in M$, the first of our four conflict kinds, rather than a failure of the procedure;
- **deontic rules** body $\to \mathsf{M}(\text{act}, s, [t_1, t_2])$ with $\mathsf{M} \in \{\mathsf{O}, \mathsf{F}\}$: when the (Horn) body holds, action act is obliged or forbidden on scope atoms s over the ground time interval $[t_1, t_2]$;
- **exclusive interval claims** (*resource x is exclusively mine on* $[t_1, t_2]$);
- **difference constraints** $x_j - x_i \leq \delta$ over deadline variables.

A *conflict* is any of four things: derivable $\bot$; a fired O/F pair on the same action with overlapping scope-and-interval; two exclusive claims overlapping on the same resource; or a negative cycle in the difference-constraint graph (an unsatisfiable deadline system). This is deliberately a *ground* fragment — no nesting of deontic operators, no negation-as-failure, monotone bodies only — and §8 says explicitly which classical deontic puzzles are thereby left outside the door.

The detection theorem and its frontier

Theorem 1a (membership: detection is polynomial and witness-producing). For a policy set in $\mathcal{L}_c$ with Horn program size H, T fired deontic tokens, X exclusive claims, and difference-constraint graph (V, E), conflict detection is decidable in $O(H + T \log T + X \log X + V \cdot E)$, with a polynomial witness per conflict: Dowling–Gallier unit propagation computes the least Horn model in $O(H)$ — which by monotonicity equals the intersection of all models, i.e. exactly the statuses forced in *every* world [verified framework, Dowling–Gallier 1984]; a keyed sweep-line over interval endpoints finds every O/F clash and claim overlap in $O(T \log T + X \log X)$; Bellman–Ford returns a negative cycle iff the deadline system is infeasible, in $O(V \cdot E)$. The witnesses are, respectively: a derivation chain, the two fired rules with their overlap interval, the two claims, and the negative cycle.

Theorem 1b (frontier: one step outside is NP-complete). Extend $\mathcal{L}_c$ with disjunctive obligations $\mathsf{O}(l_1 \vee \cdots \vee l_k)$ under discharge-choice semantics (the agent picks which disjunct to discharge). Conflict-freedom of a policy set becomes NP-complete: membership by the selection certificate; hardness from 3-SAT — each variable x becomes a pair of deontic rules $\text{sel}_x^+ \to \mathsf{O}(\text{assert}_x)$ and $\text{sel}_x^- \to \mathsf{F}(\text{assert}_x)$ on *identical* scope and interval, each clause becomes a disjunctive obligation over its selector literals, and a conflict-free discharge selection exists iff the formula is satisfiable [verified: reduction restated here in

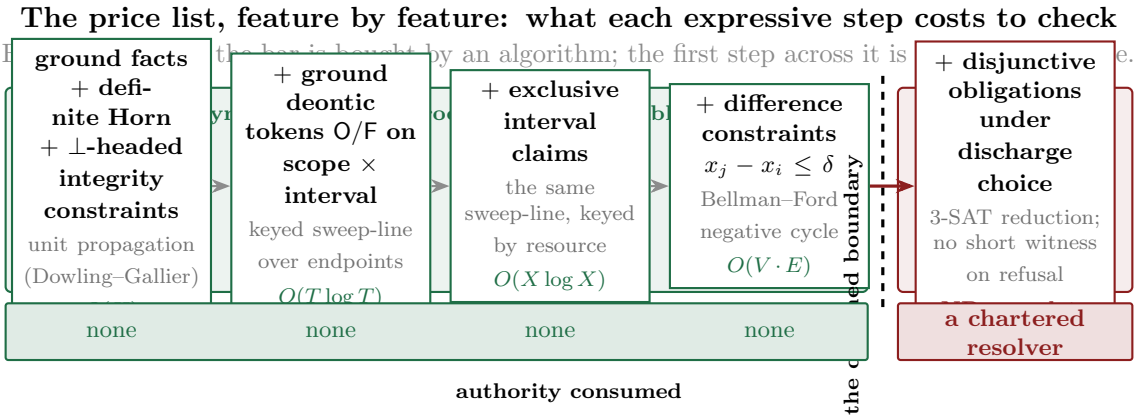

Figure 2: Part I as a price list. Each in-fragment feature names the algorithm that discharges it and the term it contributes to the $O(H + T \log T + X \log X + V \cdot E)$ bound of Theorem 1a — and the strip beneath records that none of them consumes an authority. The dashed bar is the designed boundary: admitting disjunctive obligations under discharge-choice semantics makes conflict-freedom NP-complete (Theorem 1b, by reduction from 3-SAT), and it is the first feature whose cost is paid in judgment rather than in cycles. The figure carries no plotted data; every term is restated from the two boxes above [internal, `b4_deontic_fragment.py`, seed 20260816].

full; checked both directions on 16/16 instances, internal, `b4_deontic_fragment.py`].

Read together, Theorems 1a and 1b are a **dichotomy** at a designed boundary, in the sense Schaefer [16] made standard for satisfiability: on one side of a single expressive choice the problem is polynomial, on the other it is NP-complete, and the language designer is offered nothing in between (Figure 2 lays the fragment out feature by feature, with the algorithm and the cost term each one buys). We claim the location, not the pattern — the same step is familiar one formalism over, where the simple temporal problem is polynomial and admitting disjunctions of difference constraints makes consistency NP-complete [17, 18]. That is why $\mathcal{L}_c$'s boundary should be read as a price list rather than an accident of this particular fragment.

Verification. 3000 random in-fragment policy sets against a brute-force oracle (all 2^5 Horn models intersected; pairwise overlap checks; exhaustive integer search over the complete potential box $[-\Sigma|\delta|, 0]^V$ for the difference system): **885 of the 3000 policy sets carried at least one conflict** — 121 of them a derivable $\bot$, 195 an O/F clash, 484 a claim overlap, 213 a negative temporal cycle. Those four counts are per-policy-set booleans and they *overlap*: 764 sets carry exactly one kind of conflict, 114 carry two, 7 carry three, which is why the four sum to $764 + 2(114) + 3(7) = 1013$ rather than to 885. **0 disagreements** with the oracle, and least model = intersection-of-all-models on all 3000 [internal, `b4_deontic_fragment.py`, seed 20260816]. The 3-SAT reduction was verified in both directions on 8 satisfiable and 8 unsatisfiable random instances (n=4, m=7–28), 16/16, with the satisfying assignment extracted from each conflict-free selection. Mutation: 149 of the 885 sets contain a conflict reachable *only* through Horn propagation, and in 100 of those that is the set's *only* conflict, so deleting propagation (checking only directly stated statuses) flips the verdict to conflict-free outright; the other 49 are still caught, because the mutant computes claim overlaps and deadline cycles identically. Propagation is critical at scale, not a formality — and the two numbers differ by exactly the sets a second, unrelated conflict rescues [internal].

Numbers by hand *(worked example — skippable; the argument resumes at §4).* The mutant's canonical miss is a four-line policy a reviewer can check aloud: fact `agent_deployed`;

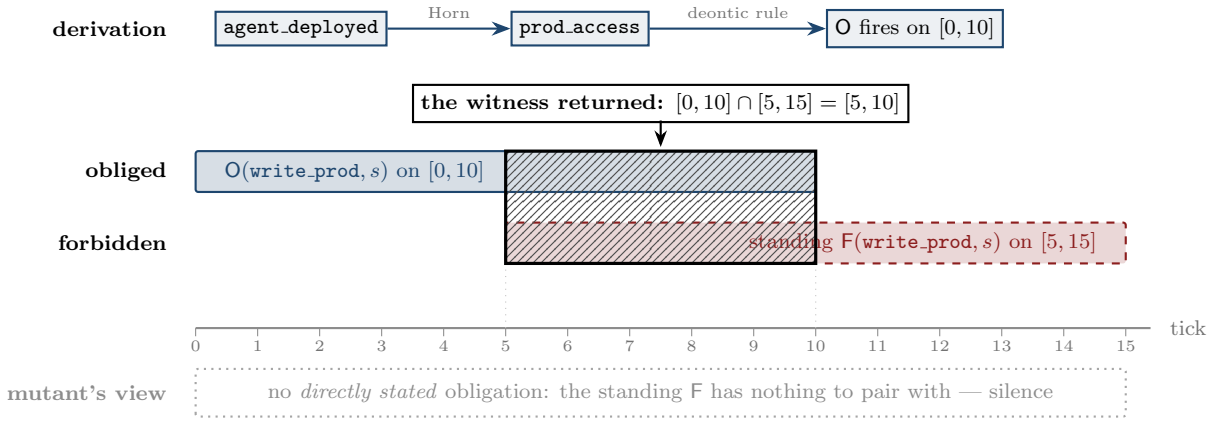

Figure 3: The witness, drawn — the canonical instance the propagation-free mutant misses. Unit propagation forces `prod_access` from the deployed fact, which fires $\mathsf{O}(\texttt{write\_prod}, s, [0, 10])$; the standing prohibition covers $[5, 15]$; and the keyed sweep-line hands back the hatched overlap $[5, 10]$ together with the two fired rules — an object a human can argue with, not a verdict. The dotted lane is the same policy set as seen by a checker that never propagates: the obligation is never derived, so there is no pair to clash and the scanner stays silent. This is the species of conflict whose absence flips the verdict on 100 of the 885 conflicting policy sets in the sweep [internal, `b4_deontic_fragment.py`, seed 20260816].

Horn rule $\texttt{agent\_deployed} \to \texttt{prod\_access}$; deontic rule $\texttt{prod\_access} \to \mathsf{O}(\texttt{write\_prod}, s, [0, 10])$; standing $\mathsf{F}(\texttt{write\_prod}, s, [5, 15])$. Propagation forces `prod_access`, the obligation fires, and the sweep-line reports the O/F overlap on $[0, 10] \cap [5, 15] = [5, 10]$ — while a pairwise scanner that never propagates sees no directly stated clash and stays silent [internal, the printed witness]. Figure 3 draws exactly this instance, and what the propagation-free mutant sees in its place. For the frontier, run the reduction on the smallest unsatisfiable formula $(x) \wedge (\neg x)$: two disjunctive obligations force both sel_x^+ and sel_x^-, firing $\mathsf{O}(\text{assert}_x)$ and $\mathsf{F}(\text{assert}_x)$ on identical scope and interval — every selection conflicts, matching unsatisfiability [verified: two lines]. *Now you try:* facts {`audit_open`}; rules $\texttt{audit\_open} \to \texttt{vault\_access}$ and $\texttt{vault\_access} \to \mathsf{O}(\texttt{read\_ledger}, s, [2, 6])$; standing $\mathsf{F}(\texttt{read\_ledger}, s, [4, 9])$. Conflict? (Yes: propagation forces `vault_access`, the obligation fires, and the overlap is $[4, 6]$ — a two-hop indirect conflict, exactly the species the mutant misses.)

4 Composition with Paper 2: the lookout is not even a role

Paper 2 proved that a safety policy over an agent's event alphabet is enforceable-by-prevention — *regimented*, in the normative-systems sense of a norm the institution makes impossible to violate rather than merely sanctioning after the fact [14] — iff it is Ramadge–Wonham controllable — no uncontrollable event carries a legal history out of the policy — and its design rule was: uncontrollable events in a policy's *condition* are free; only uncontrollable events in its *prohibition* are fatal. Now compose. In the harbor, *acceptance of a commitment is a mediated write*: the accepted set lives in the single-writer store, and the acceptance event is a member of Σ_c, the controllable alphabet. The policy *"never accept a proposal whose union with the accepted set conflicts in-fragment"* therefore gates a controllable event on a condition that Theorem 1a makes a total, polynomial-time-computable predicate of committed state. The policy is controllable, hence regimentable: the conflict never enters the accepted set, because the daemon refuses the write, with the theorem's witness attached to the refusal [verified: composition of Theorem 1a with Paper 2's boxed theorem; both components machine-checked separately,

internal, `b4_deontic_fragment.py` + `b3_controllability.py`].

This is the paper's first answer, and it is sharper than "the lookout doesn't need judgment." Inside the fragment the lookout is not an authority at all — it is a *guard clause* at the mediation boundary, as impersonal as a type check, producing a four-line witness instead of a verdict. What the charter must still supply is everything the guard cannot: *which* of two conflicting proposals yields (priority, negotiation, waiver), *whether* a standing norm should be amended, and — the moment anyone wants disjunctive obligations — *which discharge to choose*, because Theorem 1b says no proposal-time check scales there unless P=NP. The frontier is thus the exact line where authority re-enters: the schema designer reads Theorem 1 as a price list — keep the language in $\mathcal{L}_c$ and detection is a regimentable subroutine; buy the expressive convenience of disjunction and you have hired a judge, with a queue.

5 Part II. The sole owner: an inequality, not a principle

The vocabulary of Part II: queueing. An $M/M/1$ *queue* is one server, arrivals landing at random at average rate λ, each job taking a random time averaging $1/\mu$. $M/M/c$ is the same with c interchangeable servers sharing one line — the airport-check-in arrangement, not the supermarket one. *Utilisation* ρ is the fraction of capacity in use; as it approaches 1 waiting time diverges, which is the entire reason these comparisons are not obvious. The *Erlang-C* formula $C(c, \rho)$ gives the probability that an arriving job finds every server busy and must wait at all. It is a century old [5] and it is the only non-elementary ingredient in Theorem 2; everything else there is algebra on it. *Pooling* is the practice of putting all work in one queue served by everyone, as opposed to *dedicating* a specialist to it. The classical fact — and the tension the paper prices — is that pooling wins on waiting time through statistical smoothing, while a specialist may win on service rate through skill. The *skill premium* $r = \mu_s/\mu_g$ is how much faster the specialist is, and Theorem 2 asks exactly how large it must be to pay. Four prices ride along: the *accountability value* A (the worth of single-writer attribution per unit time of demand), the *waiting cost* w per request-hour, the specialist's *death rate* ξ, and the *succession rate* η (mean spin-up $1/\eta$); $K = A/(w\lambda) + W_{\text{pool}}$ is the shorthand for the pool's cost line the two theorems compare against. Finally, a queue *with breakdowns* is one whose server fails at some rate and is restored after some delay. Its relevance is that a sole owner is a server that can fail: sessions die, containers are reaped. Theorem 3's W_∞ is the residual delay such a queue carries even with an infinitely fast server, which is why no amount of skill buys it back.

Part I located the one place inside conflict management where a mind is genuinely required: not to notice a contradiction but to choose a discharge among disjuncts the fragment cannot rank on its own. Part II asks the harbor's other standing authority question, about a different kind of decision entirely — not *which* candidate resolution to pick, but *whether one designated person should be doing the picking at all* — and it gets the same kind of answer: not a principle to invoke but a number to compute. The whitepaper's sole-responsibility roles (roadmap owner, test-suite curator, release avatar) trade pooled capacity for accountability and skill. Model the trade honestly. Requests for the invariant-critical function arrive Poisson at rate λ; a sole specialist serves them as an $M/M/1$ queue at rate μ_s; the pooled alternative is an $M/M/c$ queue of generalists at rate μ_g each, utilization $\rho = \lambda/(c\mu_g)$; the skill premium is $r = \mu_s/\mu_g$. Sole ownership also carries the accountability value A against the waiting cost w just defined. The whitepaper proposed the threshold $\tilde{g} = 1 + (c-1)\rho/(1-\rho)$ and marked it *status: proposed*. The sweep ran before the belief formed — and falsified $\tilde{g}$ in both directions.

The specialization boundary

Theorem 2 (queueing specialization boundary). With $C(c, \rho)$ the Erlang-C delay

probability, sole ownership weakly dominates the pool in net cost iff the skill premium clears

$$\frac{\mu_s}{\mu_g} \;\geq\; g_A(\rho, c) \;=\; c\rho \;+\; \frac{1}{1 + \dfrac{C(c,\rho)}{c(1-\rho)} + \dfrac{A\mu_g}{w\lambda}},$$

reducing at $A = 0$ to $g(\rho, c) = c\rho + \dfrac{c(1-\rho)}{c(1-\rho) + C(c,\rho)}$, with $g(\rho, 1) = 1$ (a specialist against one generalist needs no premium), $g \to c$ as $\rho \to 1$ (the boundary *saturates* at the capacity ratio, never diverges), and the closed form $g(\rho, 2) = 1 + 2\rho - \rho^2$ [verified: derivation below]. The proposed threshold $\tilde{g} = 1 + (c-1)\rho/(1-\rho)$ is **falsified in both directions**: at c=2, ρ=0.1, r=1.15 it certifies a specialist who loses ($W_{\text{solo}} = 1.0526 > W_{\text{pool}} = 1.0101$; simulation $z = -15.7$), at c=2, ρ=0.5, r=1.9 it rejects a specialist who wins ($1.111 < 1.333$; $z = +32$); the two thresholds cross at $\rho = (3 - \sqrt{5})/2 \approx 0.382$, and where g saturates $\tilde{g}$ diverges (ρ=0.9, c=8: $g = 7.73$ vs $\tilde{g} = 64$) [internal, `b8_specialization.py`, seed 20260816].

Numbers by hand *(worked example — skippable; the cross-validation stack resumes below).* The theorem is two lines of algebra on textbook formulas, which is the right way to first believe it. Sole ownership pays iff $w\lambda(W_{\text{solo}} - W_{\text{pool}}) \leq A$, i.e. $1/(\mu_s - \lambda) \leq W_{\text{pool}} + A/(w\lambda)$; invert and divide by μ_g, using $\lambda/\mu_g = c\rho$ and $\mu_g W_{\text{pool}} = 1 + C(c,\rho)/(c(1-\rho))$, and g_A falls out [verified]. For $c = 2$ the Erlang-C value is $C(2,\rho) = 2\rho^2/(1+\rho)$, so $c(1-\rho)/(c(1-\rho) + C) = (1-\rho^2)/((1-\rho^2) + \rho^2) = 1 - \rho^2$ and $g(\rho, 2) = 2\rho + 1 - \rho^2$ [verified: three lines]. Now check the falsification instances by hand at $\mu_g = 1$. At $\rho = 0.1$: $g = 1 + 0.2 - 0.01 = 1.19$ while $\tilde{g} = 1 + 0.1/0.9 = 1.111$, so $r = 1.15$ sits between them — the whitepaper certifies, the truth rejects; indeed $W_{\text{solo}} = 1/(1.15 - 0.2) = 1.0526$ against $W_{\text{pool}} = 1 + 0.0182/1.8 = 1.0101$ [verified]. At $\rho = 0.5$: $g = 1.75$, $\tilde{g} = 2$, and $r = 1.9$ again sits between — rejected by the whitepaper, but $W_{\text{solo}} = 1/0.9 = 1.111 < W_{\text{pool}} = 4/3$ [verified]. Setting $1 + 2\rho - \rho^2 = 1 + \rho/(1-\rho)$ gives $\rho^2 - 3\rho + 1 = 0$: the thresholds cross at $\rho = (3 - \sqrt{5})/2$, which is why the falsification runs in *both* directions — Figure 4, Panel A, shades the two error lenses [verified]. *Now you try:* at c=2, ρ=0.3, does $r = 1.5$ justify a sole owner? ($g(0.3, 2) = 1 + 0.6 - 0.09 = 1.51$ — no, by a hair; $\tilde{g} = 1.43$ would have said yes. The flattering threshold flatters exactly below the crossing.)

The cross-validation stack behind the closed forms: Erlang-C means against discrete-event simulation, $|z| \leq 1.8$ across the grid; and the 60-instance random sweep over $(\lambda, \mu_s, \mu_g, c, A, w, \xi, \eta)$, which compares the closed-form decision against net cost computed from *simulation* — FCFS discrete-event for the pool, Gillespie CTMC for the breakdown specialist, no closed form entering the simulated side — gave 55 decisive instances and **0** sign violations of the boundary [internal, `b8_specialization.py`].

6 The succession price: what one mortal body costs

The boundary above assumes the specialist never disappears. Sole owners disappear: sessions die, containers are reaped, providers churn. Model the corner honestly: the specialist *dies* at rate ξ and is *succeeded* after an Exp(η) spin-up (context transfer, warm-up, credential rebinding), work resuming where it stopped, requests queueing throughout — an $M/M/1$ queue with breakdowns in the Mitrany–Avi-Itzhak lineage.

Theorem 3 (price of the succession rule). With effective rate $\mu_{\text{eff}} = \mu_s\eta/(\xi + \eta)$, the

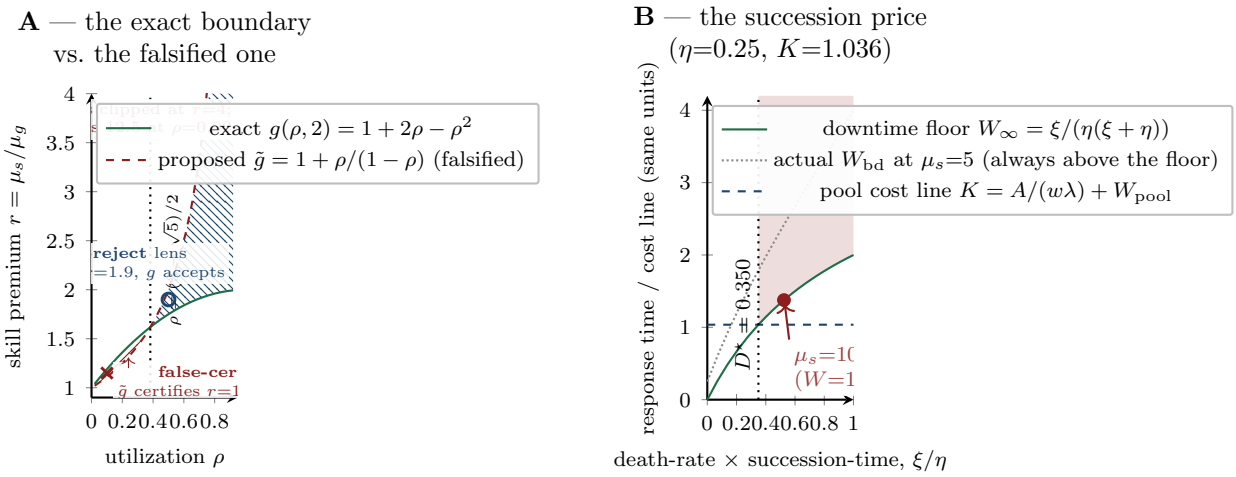

Figure 4: The regime diagram. Panel A: the exact boundary $g(\rho,2) = 1+2\rho-\rho^2$ (solid) against the falsified proposal $\tilde{g}$ (dashed). The two hatched lenses — each labelled on the figure — are the two error regimes, and the proposal is wrong on *both* sides of the crossing at $\rho = (3-\sqrt{5})/2$, so no constant rescues it; the refuting instances are marked (r=1.15 at ρ=0.1; r=1.9 at ρ=0.5). The y-axis is clipped at r=4: $\tilde{g}$ leaves the frame near ρ=0.76 and reaches 12.5 by ρ=0.92, which is the divergence g does not have. Panel B: the succession price at the reference instance (η=0.25, K=1.036): the downtime floor $W_\infty = \xi/(\eta(\xi+\eta))$ crosses the pool cost line at $D^\star = 0.350$, and to its right pooling wins at every *finite* skill premium. The dotted grey series is the actual W_{bd} at a finite μ_s=5, drawn to make the floor's un-attainedness visible: no specialist ever sits on it, which is why Theorem 3's criterion is non-strict at $D^\star$. The marked point is a $\mu_s = 10^8$ specialist still losing at $\xi/\eta = 1.5D^\star$ ($W = 1.376 > K$). Both panels are closed-form curves plotted from `figures/src/paper6_figures.py`; the underlying boundary and $D^\star$ results are `b8_specialization.py` [internal, seed 20260816].

sole owner's mean response time is exactly

$$W_{\mathrm{bd}} \;=\; \frac{1+\mu_s\xi/(\xi+\eta)^2}{\mu_{\mathrm{eff}}-\lambda} \;=\; \frac{1}{\mu_{\mathrm{eff}}-\lambda} \;+\; \frac{\mu_{\mathrm{eff}}}{\mu_{\mathrm{eff}}-\lambda}\,W_\infty, \qquad W_\infty = \frac{\xi}{\eta(\xi+\eta)},$$

decreasing in μ_s with infimum W_∞, *never attained*: *no amount of skill buys back residual downtime*. Consequently, with $K = A/(w\lambda) + W_{\mathrm{pool}}$ the pool's cost line, pooling dominates at *every* finite skill premium iff the floor has already reached that line, $W_\infty \geq K$ — equivalently, *provided* $\eta K < 1$, iff

$$\frac{\xi}{\eta} \;\geq\; D^\star \;=\; \frac{\eta K}{1-\eta K}.$$

The inequality is non-strict because the infimum is unattained: at $\xi/\eta = D^\star$ exactly, $W_\infty = K$ but $W_{\mathrm{bd}}(\mu_s) > W_\infty$ for every finite μ_s, so pooling still dominates there. The domain condition is not decoration: $W_\infty < 1/\eta$ for every ξ, so when $\eta K \geq 1$ the floor never reaches the cost line, there is no cap at all ($D^\star = +\infty$), and only the Theorem-2 premium test binds — the closed form returns a negative number outside its domain and must not be read there as a threshold. So $D^\star$ is the price of the succession rule: a sole role without a succession plan (small η) is viable only while its death-rate–times–succession-time product stays strictly below $D^\star$ [verified: $W_\infty(\xi^\star) = K$ at $\xi^\star = D^\star\eta$ is one line of algebra; closed form checked against matrix-geometric solution to 2×10^{-13}, truncated CTMC to 10^{-14}, Gillespie simulation $|z| \leq 0.8$, internal, `b8_specialization.py`].

Numbers by hand *(worked example — skippable).* Reference instance — the release-tag owner, metered over a week: λ=1, c=4, μ_g=1.2 (so $\rho = 0.208$), A=0.2, w=1, η=0.25. Erlang-C gives $C(4, 0.208) = 0.0110$, so $W_{\text{pool}} = 1/1.2 + 0.0110/(4 \cdot 1.2 \cdot 0.792) = 0.8362$ and $K = 0.2 + 0.8362 = 1.0362$ [verified: four-function-calculator arithmetic on the boxed formulas]; then $\eta K = 0.2591$ and $D^\star = 0.2591/0.7409 = 0.350$ [verified]. Just past the line, at $\xi/\eta = 1.5\, D^\star$, an absurdly skilled specialist ($\mu_s = 10^8$) still loses: $W = 1.376 > K = 1.036$ — the floor W_∞, not the skill, is binding [internal]. The identity in the box also explains the program's recorded wrong turn: the naive decomposition $W_{\text{bd}} = 1/(\mu_{\text{eff}} - \lambda) + W_\infty$ is refuted (it predicts 4.17 where the truth is 8.17 at the canary instance μ_s=5, ξ=0.5), because downtime also *builds queue*: the residual is amplified by $\mu_{\text{eff}}/(\mu_{\text{eff}} - \lambda) = 2.5$ there, and $1.5 + 2.5 \times 2.667 = 8.17$ checks by hand [verified arithmetic; refutation internal]. Panel B of Figure 4 draws exactly this geometry: the floor W_∞ crossing the pool cost line at $D^\star$, with the actual W_{bd} at a finite μ_s drawn above it, so the un-attainedness of the infimum is visible rather than asserted. *Now you try:* at the reference instance, is a sole owner with a same-day succession plan ($\eta = 2$; K is unchanged at 1.0362, so $\eta K = 2.07 > 1$) ever capped? (No: $\eta K \geq 1$ puts the pool's cost line above the floor's own ceiling $1/\eta$, so no ξ triggers the cap — $D^\star = \infty$ by the box's domain clause, not by its closed form, which has left its domain. A fast enough succession plan removes the viability ceiling entirely, leaving only the Theorem-2 premium test. That is what a succession plan *buys*.)

The implementation hazard is also priced. A cost model that forgets breakdowns entirely (plain $M/M/1$) is caught by the high-ξ canary: at $\xi/\eta = 2 \gg D^\star$, the mutant says SPECIALIST while simulation says the pool wins decisively ($\Delta\mathcal{C} = -7.06 \pm 0.16$, the net-cost difference); at low ξ the mutant is indistinguishable — so the canary, not routine testing, is what makes the omission visible [internal, `b8_specialization.py`].

The predictable misread, preempted: $D^\star$ does not say specialists are bad, and Theorem 2 does not say pooling is the default truth. Together they say sole ownership is a *purchase*: below the boundary the premium does not cover the congestion cost and the honest ledger entry is "we pay latency $w\lambda(W_{\text{solo}} - W_{\text{pool}}) - A$ per unit time for accountability"; above $D^\star$ the purchase is unavailable at any price until the succession plan (larger η) is bought first. Every quantity in both tests — λ, μ_s, μ_g, ξ, η — is daemon-metered, so the harbor can re-grade its roles from telemetry rather than from taste. One caveat travels with the word *exact* in this paper's subtitle: the number is exact in the queueing parameters and only as good as A and w in the policy ones. The theorem prices accountability *given* a price for accountability; it does not pick that price, and boundary item (v) says so again.

7 The authority inventory, priced

The two parts assemble into the answer to the title question, and Table 1 is that answer in one screen.

That is the paper's claim in institutional terms: a well-designed harbor spends judgment *only* where theorems certify judgment is required, and meters what the remaining judgment costs.

What it buys, concretely. Two things ship from Table 1. The first two rows are a schema rule for the commitment language: as long as proposals stay inside $\mathcal{L}_c$, the acceptance path needs no reviewer role at all — the daemon's mediated write refuses the commit and attaches the witness, and the charter can delete the lookout it was about to appoint. The last two rows are a role-grading rule: λ, μ_s, μ_g, ξ and η are all already metered per role, so a harbor can compute r and $g_A(\rho, c)$ for each of its sole-responsibility roles on a schedule, demote the ones that stopped clearing the boundary, and price a succession plan by exactly how much η it buys.

Table 1: The authority inventory, priced. Rows 1–2 delete an authority the governance document assumed it needed — in-fragment detection is an algorithm and, composed with Paper 2, a runtime guard. Rows 3–4 locate the authority no theorem removes: choosing a discharge, and choosing the norms. Rows 5–6 keep the authority and hand it an invoice, in quantities the daemon already meters.

Coordination function	Authority needed?	What it costs, and to whom
In-fragment conflict detection	none (algorithm)	$O(H + T \log T + X \log X + V \cdot E)$ per batch, witness included
In-fragment conflict enforcement	none (runtime guard)	mediated acceptance write; regimentable via Paper 2
Conflict-freedom with disjunctive obligations	chartered resolver	NP-complete check, or a judge who chooses the discharge
Priority, waiver, norm amendment	chartered, irreducibly	outside every checker; the charter's residual job
Sole ownership of a role	conditional	justified iff $r \geq g_A(\rho, c)$; otherwise a priced latency purchase
Sole ownership without succession	conditional, capped	viable only while $\xi/\eta < D^\star = \eta K/(1 - \eta K)$, and only where $\eta K < 1$

8 Related work

Imported. Linear-time Horn satisfiability and unit propagation are Dowling–Gallier [1]; we import the algorithm and the least-model property unchanged, and our fragment is *designed* so that it applies. NP-completeness machinery is Cook and Karp [2, 3]; the reduction in Theorem 1b is standard 3-SAT gadgetry, and the contribution is only the *location* of the frontier — discharge-choice disjunction — in a commitment language a scheduler would actually want. The temporal half of the dichotomy is Dechter, Meiri and Pearl [17], who established both the polynomial simple temporal problem and the NP-hardness of the disjunctive temporal problem; Stergiou–Koubarakis [18] is the algorithmic successor we cite alongside it. The regimentation composition imports Ramadge–Wonham supervisory control [4] via Paper 2, and uses *regimentation* in Jones and Sergot's [14] sense. On the queueing side, the Erlang-C delay formula is a century old [5]; the economies of pooling are classical — Smith and Whitt [7] established the resource-sharing comparison of one shared multi-server queue against dedicated ones (and the heterogeneity caveat under which sharing can *hurt*), which is the ancestor of Theorem 2's finite-c pooled-vs-dedicated form, while Halfin–Whitt [6] sharpened the many-server scaling regime in which the economy is usually quantified — and queues with server breakdowns descend from Mitrany–Avi-Itzhak [9]; Theorem 3's W_{bd} is in that lineage. **Positioned against deontic logic.** The fragment is ground and operator-nesting-free: conflicts are defined operationally (O vs F on overlapping scope and interval), not as theoremhood in classical deontic logic [10], and the classical paradoxes — Ross's paradox [11], Chisholm's contrary-to-duty puzzle [12] — simply cannot be *stated* in $\mathcal{L}_c$: that is a designed retreat, declared rather than hidden, and it is what the tractability is paid for with. **Positioned against computational normative systems.** The adjacent literature that owns *detecting* conflicts between obligations and prohibitions is not the philosophical one: it is the normative-multi-agent-systems line collected in Meyer and Wieringa [13] and pursued in the normative-conflict work of Vasconcelos, Kollingbaum and Norman [15], whose conflict predicate is, like ours, operational rather than logical — and that is the point at which their line and ours converge rather than diverge. **Where the prior-art search is not yet discharged, said plainly.** We have not run a systematic pass against three literatures this result sits inside: (i) the computational normative-MAS conflict-detection line above,

where a tractable/intractable line may already have been drawn in some ground fragment, which is the claim of Theorem 1 most at risk; (ii) operations management's own vocabulary for the specialist-versus-pool question, which is not called pooling-vs-dedication there but *server flexibility* and, in the call-centre branch, *skill-based routing* (Gans, Koole and Mandelbaum [8] is the standard survey and the right entry point); and (iii) the agent-communication literature's own use of *commitment*. Naming these is not a substitute for searching them, and no claim below should be read as surviving a search that has not been run. **New, honestly.** No component algorithm or queueing formula above is ours. The contributions are: the fragment/frontier pair as an exact price list for commitment-language design; the composition making in-fragment conflict-freedom regimentable rather than merely checkable; the falsification and replacement of the whitepaper's specialization threshold by the exact $g_A(\rho, c)$ with its c=2 closed form and crossing point; and the succession criterion $D^\star$ stating, in daemon-metered quantities, when no skill premium can rescue a sole role. An August-2026 survey found no prior statement of the authority question in this detection/resolution/ownership decomposition; a final lit-sweep before submission is owed — "not found" is not "proven nonexistent."

9 Honest boundary

What these theorems do NOT say. (i) Theorem 1's completeness is relative to least-model semantics: monotone bodies only — negation-as-failure, unification, and symbolic interval endpoints coupled to difference variables all leave the fragment, and nothing here is claimed about them. (ii) The batch complexity bound is what is proven; the incremental amortized form (watched literals, interval trees) is a constants engineering claim, stated in the portfolio, not certified here. (iii) The checker certifies the norm *system*, never the norms: a harbor whose policies are consistently harmful passes every scan — norm content is chartered authority's irreducible job, row four of the inventory. (iv) The regimentation composition inherits Paper 2's deployment obligation wholesale: acceptance must *actually* be completely mediated (Σ_c honest), which a red-team, not this proof, discharges. (v) Theorem 2 compares *mean* costs only — no tail percentiles, no SLA quantiles — under FCFS and exponential everything; A and w are policy prices the theorem constrains but does not pick. (vi) Theorem 3's breakdown model is death-at-all-times with preemptive resume (the death/heartbeat model); active-only breakdowns would let sufficient skill escape the W_∞ floor, so the model choice is critical and declared. (vii) The wrong turns are on record and priced: the naive W_{bd} decomposition is refuted (4.17 vs 8.17), and the whitepaper's $\tilde{g}$ was falsified by this program's own sweep — the correction, not the original, is what ships. (viii) The inventory covers detection, resolution, and ownership; other authority functions (charter amendment, dispute appeal, norm authorship) are chartered by construction and no theorem here bears on them. (ix) The oracle validation is on *small* instances: the brute-force comparison intersects all 2^5 Horn models and searches an integer potential box exhaustively, so 3000 disagreement-free runs certify the algorithm's agreement with the semantics on five-atom programs, not its behaviour at production policy-set size — the complexity bound, not the sweep, is what covers scale. (x) Theorem 3's threshold $D^\star$ presupposes $\eta K < 1$. Where the succession plan is fast enough that $K \geq 1/\eta$, the downtime floor never reaches the pool's cost line and there is no viability cap to state: the criterion does not apply, rather than being satisfied. (xi) The prior-art position is partial by declaration: three adjacent literatures named in §8 have not been systematically searched, and the novelty claim is scoped accordingly.

Reproducibility. Every [internal] number regenerates from two self-contained scripts at seed 20260816: `skills/harbor-results/scripts/b4_deontic_fragment.py` (3000-instance oracle sweep, conflict census, least-model check, 16/16 reduction verification, Horn-propagation mu-

tation) and `skills/harbor-results/scripts/b8_specialization.py` (falsification instances, Erlang-C vs DES, matrix-geometric / CTMC / Gillespie cross-validation of W_{bd}, 60-instance sign sweep, $D^\star$ instance, breakdown-forgetting mutation canary). Every [verified] number is a closed form or short derivation restated in this paper's boxes and checkable by hand or from the cited literature. Figures 1 and 4 are native TikZ/pgfplots fragments whose plotted coordinates regenerate from `docs/harbor-research/figures/src/paper6_figures.py`; the boundary and $D^\star$ results they draw are `b8_specialization.py`'s. Figures 2 and 3 are structural diagrams carrying no plotted data — every complexity term and interval in them is restated from the boxes above.

References

[1] W. F. Dowling and J. H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming*, 1(3):267–284, 1984.

[2] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 151–158, 1971.

[3] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[4] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization*, 25(1):206–230, 1987.

[5] A. K. Erlang. Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. *Elektroteknikeren*, 13:5–13, 1917.

[6] S. Halfin and W. Whitt. Heavy-traffic limits for queues with many exponential servers. *Operations Research*, 29(3):567–588, 1981.

[7] D. R. Smith and W. Whitt. Resource sharing for efficiency in traffic systems. *Bell System Technical Journal*, 60(1):39–55, 1981.

[8] N. Gans, G. Koole, and A. Mandelbaum. Telephone call centers: tutorial, review, and research prospects. *Manufacturing & Service Operations Management*, 5(2):79–141, 2003.

[9] I. L. Mitrany and B. Avi-Itzhak. A many-server queue with service interruptions. *Operations Research*, 16(3):628–638, 1968.

[10] G. H. von Wright. Deontic logic. *Mind*, 60(237):1–15, 1951.

[11] A. Ross. Imperatives and logic. *Theoria*, 7:53–71, 1941.

[12] R. M. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analysis*, 24(2):33–36, 1963.

[13] J.-J. Ch. Meyer and R. J. Wieringa, editors. *Deontic Logic in Computer Science: Normative System Specification*. Wiley, 1993.

[14] A. J. I. Jones and M. Sergot. On the characterisation of law and computer systems: the normative systems perspective. In J.-J. Ch. Meyer and R. J. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 275–307. Wiley, 1993.

[15] W. W. Vasconcelos, M. J. Kollingbaum, and T. J. Norman. Normative conflict resolution in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 19(2):124–152, 2009.

[16] T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC)*, pages 216–226, 1978.

[17] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49(1–3):61–95, 1991.

[18] K. Stergiou and M. Koubarakis. Backtracking algorithms for disjunctions of temporal constraints. *Artificial Intelligence*, 120(1):81–117, 2000.