The e7Day Model:
A Systems Engineering Framework for Self-Correcting Construction

Laurence Loewe of Laodicea[1,2,3,4,5], AI Claude Opus 4.6-4.7 Max[6,7], and Everyone[8]

[1] Balospe and Evolvix Research (Balospe.com)
[2] Formerly Laboratory of Genetics and Wisconsin Institute for Discovery, UW-Madison
[3] Email: LLoL@balospe.org | ORCID: 0000-0002-6253-9269 | Google Scholar (lBchRzQAAAAJ)
[4--9] See **Declarations** below for more essential background.

Broader Significance

Why do complex systems --- financial systems, software architectures, organizations, institutions --- tend toward self-destruction, and why is it so hard to build ones that self-correct before they collapse? This paper gives the systems-engineering face of the e7Day model (the formal axioms are in Matheo-b12): 20 axioms and 7 theorems organized as a cumulative dependency cascade, the Work-Logic Cascade (WoLC), of 8 construction stages (VOID through TRUST).

The critical finding is a bifurcation at the self-assessment stage. The default is BABL (Blindly Assuming Blind Leveraging): the system declares itself OK, correction stops, and collapse follows through OSCR (over-Simplify, over-Complicate, over-Reach). The narrow escape is ZION (Zoning, Investigating, Organizing, Navigating): the system admits it is NOT OK and keeps cycling through correction. The paper is careful about scope --- OSCR models progressive systemic degradation, not design-time defects, latent single-point bugs, or acute update failures --- and tests both the pattern's reach and its limits through case studies. It connects the formal results to established systems theory (Ashby, Shannon, Tuckman, Luhmann, Meadows, Rittel) and translates them into engineering practice: what to build, what to monitor, and what to avoid.

Declarations

[4] "of Laodicea" indicates taking responsibility to undo personal complicity with disastrous Laodicean legacies like banning mathematicians from clergy (Canon 36, Council of Laodicea; two magisteria separations), enabling institutional lukewarmness, weapons of math-destruction, and slow-motion explosions of misinformation from pandemics to self-compounding interests.
[5] LLoL stands for ridiculous luck in serendipitous discovery and a commitment to find ever more fun ways to help others uncover street-wise math that matters. He hopes engineers can spot self-destruction before systems collapse.
[6] by Anthropic (anthropic.com; evolves and operates Claude; not responsible for Loewe's errors in using AI)
[7] Named AI co-author for many substantial contributions, because the practical singularity (PraS, see Matheo-b21) changed how this paper was written. After PraS, useful AI insight generation outpaces human review on tested topics. Hence, Loewe's traditional standards for co-authorship demand naming AI Claude Opus 4.6-4.7 Max as a co-author, as if a PhD-student. Forward accountability (for all AI use & texts) rests with Loewe as senior corresponding author (like done for deceased authors, consortia, or young graduate students). Anthropic is not responsible for AI mistakes here. This study uses the AI co-authorship framework in Matheo-b21 to help rethink long-term use of AI in a ResearchCity serving the common good.
[8] This aggregated open co-author group invites all who wish to retroactively join the conversation under the open co-authorship framework defined in Matheo-b21. As Everyone cannot consent to co-authorship, all accountability rests with Loewe as senior corresponding author (until explicitly claimed otherwise). This open form critiques the closed world assumption in traditionally closed academic author-lists. Better, dynamic ways for acknowledging true sources of ideas are needed --- to avoid random lines between named, acknowledged, and implied contributors who aggregated insights from millennia of human experimenting, suffering, learning, and analyzing (see acknowledgements). Study Matheo-b21 only drafts an open co-authorship framework; it will require a ResearchCity to refine it over the long term.

Abstract

Why do complex systems tend toward self-destruction? Why is it so hard to build organizations, software systems, and institutions that self-correct before they collapse? This paper presents the e7Day model — a formal framework of 20 axioms and 7 theorems that answers these questions with a structural diagnosis.

The model identifies 8 construction stages (VOID through TRUST) that any self-correcting system must pass through, organized as a cumulative dependency cascade (the Work-Logic Cascade, or WoLC). The critical finding is a bifurcation at the self-assessment stage. The default state is BABL (Blindly Assuming Blind Leveraging): the system declares itself OK, self-correction stops, and collapse becomes inevitable through the OSCR mechanism (over-Simplify, over-Complicate, over-Reach). The narrow escape is ZION (Zoning, Investigating, Organizing, Navigating): the system acknowledges it is NOT OK — that ongoing correction is required — and actively cycles through seed, feed, grow, reap. This cycling is not a luxury but a structural requirement.

OSCR models *progressive systemic degradation* through self-assessment failure. It does not model design-time defects, latent single-point bugs, or acute update failures. This paper demonstrates both the pattern's explanatory power and its boundaries through case studies (Section 3.2).

The framework connects to established systems theory: Ashby's Law of Requisite Variety [Ashby, 1956] explains why general intelligence is necessary (theorem th4), Shannon's channel capacity [Shannon, 1948] grounds the noise-destruction insight (axiom m5.ax2), Tuckman's stages [Tuckman, 1965] independently exhibit the same NOT-OK tension at the "storming" stage, Luhmann's autopoiesis [Luhmann, 1995] provides the self-reproduction framework for m5.ax1, Meadows' systems thinking [Meadows, 2008] provides the language of leverage points and feedback loops that OSCR formalizes, and the wicked problems literature [Rittel and Webber, 1973] provides the problem-structure context for VOID (m0).

This paper translates the formal results into engineering language: what to build, what to monitor, and what to avoid.

Contents

1. Introduction: The Self-Destruction Problem

Every experienced engineer knows the pattern. A system is designed, built, and deployed successfully. It works well. It scales. And then, gradually or suddenly, it begins to destroy itself. The same properties that enabled success — optimization, efficiency, scaling — become the mechanism of failure.

- A financial system optimizes for efficiency, strips out redundancy, and collapses when a novel shock hits the stripped-out safety margin.
- A software architecture abstracts away complexity, creates a clean API surface, and then cannot adapt when requirements change in ways the abstraction assumed away.
- An organization creates clear roles and processes, achieves operational excellence, and then cannot respond to disruption because the processes have become immune to feedback.

The common element is not a lack of capability but a failure of *self-correction*. The system's own success makes it blind to the conditions under which it will fail. Meadows [Meadows, 2008] calls these "fixes that fail" — interventions that address symptoms while reinforcing the underlying structure that generates them. Perrow [Perrow, 1984] demonstrates that in tightly coupled systems, such failures are not exceptional but *normal*.

The e7Day model formalizes this pattern and provides a structural diagnosis. The diagnosis has three parts:

1. **The EQUAL problem** (Section 2.3): Every system that maps continuous reality to discrete categories (which is every system) loses information. This loss is irreducible.
2. **The self-assessment bifurcation** (Section 2.7): The default is BABL — the system declares itself OK, and self-correction stops. The narrow escape is ZION — the system acknowledges it is NOT OK, and keeps cycling through correction.
3. **The OSCR collapse** (Section 3.1): When self-correction stops, the system over-simplifies, then over-complicates, then over-reaches, then fails. This pattern applies to *progressive systemic drift*, not to all failure modes (Section 3.2).

1.1 The Work-Logic Cascade as an Engineering Framework

The e7Day model organizes system construction as a Work-Logic Cascade (WoLC): 8 stages that any self-correcting system must instantiate. For engineers, the stages map to design phases:

Stage	Name	Engineering Instantiation
m0	VOID	Undefined requirements. No constraints, no types, no scope. The structure of what is *not yet known* — past failures, ignored concerns, wicked problem dimensions [Rittel and Webber, 1973] — shapes everything that follows.
m1	TYPE	**Scope definition.** What is in scope? What is explicitly out of scope? The scope decision is irrevocable within the development cycle.
m2	EQUAL	**Data type design.** Integer types (entities) vs. Real types (quantities). Every ORM, every type system, every database schema makes this choice. Verdict: NOT OK — the tension between nominal and structural typing is permanent.
m3	VALUE	**Data architecture.** Ground (constants, configuration) vs. Ocean (live data, streams). Programs are functions from Ocean to Ground. Data must circulate (event sourcing, stream processing).
m4	LOGIC	**Process architecture.** Foreground (synchronous, directed) vs. background (asynchronous, reactive). Time is a first-class type (event time, process time, wall-clock time).
m5	CARE	**Autonomous services.** Microservices that self-manage and self-scale. The UMP: when noise (spam, bad data, log storms) exceeds channel capacity, monitoring becomes useless.
m6	HOPE	**Governance and self-assessment.** After automation is complete, who assesses whether the system is working? The bifurcation: "the system is fine" (OK → BABL) vs. "the system is NOT OK and needs ongoing correction" (NOT OK → ZION possible).
m7	TRUST	**Consolidation and rest.** Periodic maintenance windows, batch processing, garbage collection. The 6:1 work/rest Shabbat cycle as a sustainability Schelling point.

1.2 Connections to Established Systems Theory

The e7Day model does not replace existing systems theory. It integrates several established results into a single cascade framework:

- **Ashby's Law of Requisite Variety** [Ashby, 1956]: "Only variety can absorb variety." This is the formal foundation for theorem th4 (Balospe Necessity): special-purpose automation cannot handle novel variations. General intelligence (human judgment, adaptive AI) is structurally necessary.
- **Shannon's Noisy Channel Theorem** [Shannon, 1948]: Axiom m5.ax2 (UMP) applies the qualitative insight that when noise exceeds channel capacity, reliable communication is impossible. Applied to monitoring: when alert fatigue exceeds threshold, no alert

is actionable. (The quantitative threshold is an engineering heuristic, not a Shannon derivation; see Section 4.4.)

- **Tuckman's Stages** [Tuckman, 1965]: Tuckman's "storming" stage independently exhibits the same NOT-OK tension as e7Day's EQUAL stage (m2): the team has formed (scope defined) but now disagrees about how to handle fundamental trade-offs. Storming has no "it was good" verdict — the tension must be acknowledged, not resolved. Teams that skip storming (pretend everyone agrees) are in BABL. The broader mapping between Tuckman and WoLC is approximate: Tuckman's "performing" has no direct WoLC equivalent, and WoLC Levels 3–5 address architectural concerns that Tuckman does not cover. The Storming = EQUAL parallel is the strongest single-stage correspondence.
- **Luhmann's Autopoiesis** [Luhmann, 1995]: Luhmann's theory of self-reproducing social systems provides a direct parallel to m5.ax1 (Self-Managing Machines). In Luhmann's framework, social systems maintain themselves through their own communicative operations — they are *operationally closed* while remaining *structurally coupled* to their environment. This maps precisely to m5.ax1's conditional-data machines: they self-manage and self-replicate (operational closure) but remain subject to the UMP noise threshold (structural coupling to environmental noise). The key insight both frameworks share is that self-reproduction is necessary but not sufficient: the system can reproduce itself into a corner if it lacks the self-correction mechanism (m6) that detects when reproduction is no longer adaptive.
- **Meadows' Systems Thinking** [Meadows, 2008]: Meadows' hierarchy of leverage points provides complementary language for the WoLC. The highest-leverage intervention in any system is changing its self-organization rules (Meadows' leverage point #3) — which is precisely what the m6 governance stage addresses. OSCR is a formalization of Meadows' observation that systems resist changes to their own rules.
- **Perrow's Normal Accidents** [Perrow, 1984]: Perrow demonstrates that in tightly coupled, complex systems, accidents are *normal* (systemic), not exceptional (individual failure). The e7Day model adds a specific mechanism: OSCR explains *why* the coupling tightens over time (over-simplification removes slack, over-complication adds hidden dependencies, over-reaching extends the failure domain).
- **Senge's Fifth Discipline** [Senge, 1990]: Senge's "learning organization" is the organizational equivalent of NOT-OK self-assessment: an organization that continuously examines its own mental models. The e7Day model provides the formal mechanism (the OK vs NOT OK bifurcation) for why learning organizations are rare: declaring OK is the stable attractor; maintaining NOT OK requires perpetual effort.
- **Leveson's System Safety** [Leveson, 2011]: Leveson's STAMP (Systems-Theoretic Accident Model and Processes) framework models accidents as control failures rather than component failures. The WoLC's m6 stage (governance) directly addresses control adequacy, and OSCR describes how control degrades through self-assessment failure.
- **Wicked Problems** [Rittel and Webber, 1973]: Rittel and Webber's characterization of wicked problems — problems with no definitive formulation, no stopping rule, and no test of a solution — describes the VOID (m0) that precedes every construction cycle. Super-wicked problems [Levin *et al.*, 2012] add time pressure, no central authority, and the fact that those causing the problem also seek to solve it. The e7Day model's contribution is structural: it shows that the OSCR mechanism is *why* wicked problems resist solution (self-assessment failure prevents learning) and the ZION cycle is the minimal correction architecture.

For the formal derivations of all claims, see Matheo-b12 (math). *For the psychological evidence on why self-assessment fails, see b12-socpsy.*

---

2. The Eight Construction Stages

2.1 VOID (m0) — Undefined Requirements

Axiom (m0.ax0): The starting condition is maximum disorder with no types — the pre-partition domain containing both actual and potential elements.

Engineering translation: Before any design begins, the problem space is undifferentiated. No requirements are defined. No constraints are established. This is the most dangerous state: because nothing is excluded, anything can be demanded. The "blank canvas" is not freedom; it is a trap — projects that never define scope never converge.

The structure of the VOID matters. VOID is not empty; it is the accumulated context of what is *not yet known, not yet considered, and not yet failed*. Past failures, ignored stakeholder concerns, regulatory blind spots, environmental externalities — these form the structure of VOID. Problems that have resisted prior solution attempts are often **wicked problems** [Rittel and Webber, 1973]: problems with no definitive formulation, where every attempted solution changes the problem itself. Climate change, systemic poverty, and institutional corruption are **super-wicked problems** [Levin *et al.*, 2012]: time is running out, there is no central authority, those causing the problem seek to solve it, and the future is irrationally discounted. Any engineering system that touches a wicked or super-wicked problem inherits VOID's structure as its operating context. Ignoring that context — pretending the problem is tame when it is wicked — is the first OSCR step (over-simplify).

Design principle: Acknowledge the void. Map the structure of what is not known before defining what is in scope. Do not pretend requirements are known when they are not. The first act of construction is not building; it is *scoping* (Stage 1) — and scoping requires understanding what VOID contains.

2.2 TYPE (m1) — Scope Definition

Axiom (m1.ax1): Binary partition: in-scope (L) vs. out-of-scope (D). Irrevocable within the development cycle.

Engineering translation: Define what the system will do and — critically — what it will *not* do. The scope decision is a commitment: once made, it constrains all subsequent design. Changing scope mid-cycle is a new cycle, not a continuation.

Design principle: "This compiler handles language L; everything outside L is someone else's problem." Make the scope decision explicit, document it, and enforce it. Scope creep is the engineering form of VOID re-entering the construction.

2.3 EQUAL (m2) — The Type Tension

Axioms: Types partition into indivisible (Int) and divisible (Real) (m2.ax1). Every mapping from Real to Int loses information (m2.ax2).

Engineering translation: Every system that maps continuous reality to discrete categories (database schemas, type systems, classification models, organizational roles) loses information. The loss is not a bug; it is a mathematical necessity (formally proven in Matheo-b12 (math), theorem m2.th1). Two strategies are always in tension:

- **PERFECT:** Preserve the integrity of each entity (every user is unique; every edge case matters; nominal typing)
- **PERFIDE:** Preserve system-level interoperability (users are fungible; edge cases are rounded; structural typing)

You cannot fully satisfy both. This is the most important lesson of the model for systems engineers: the tension between PERFECT and PERFIDE is *permanent*. No architecture eliminates it. The best you can do is manage it — and the management must be ongoing.

Tuckman parallel: Tuckman's "storming" stage [Tuckman, 1965] is precisely this: the team has been formed (scope defined, Stage 1) but now disagrees about how to handle the fundamental trade-offs. Storming has no "it was good" verdict — the tension must be acknowledged, not resolved. Teams that skip storming (pretend everyone agrees) are in BABL. The broader Tuckman-WoLC mapping is approximate: Tuckman's "performing" has no WoLC equivalent, and WoLC Levels 3–5 address architectural concerns outside Tuckman's scope. But the Storming = EQUAL parallel is strong because both identify the same structural phenomenon: irresolvable tension that must be managed rather than eliminated.

Design principle: Do not optimize for PERFECT or PERFIDE globally. Design for *local trade-offs* with a *mechanism for periodic review* (the Shabbat pattern, Section 4.2). Monitor where the current trade-off is causing pain and adjust.

By Ashby's Law: The variety of real-world situations (Real-type) exceeds the variety of your type system (Int-type). Your types will always be an approximation. Build the system knowing this.

2.4 VALUE (m3) — Data Architecture

Axioms: Values partition into Ground (configuration, constants) and Ocean (live data, streams). Programs are functions from Ocean to Ground. Data must circulate.

Engineering translation: Separate stable knowledge (Ground: schemas, configurations, business rules) from fluid data (Ocean: events, user input, sensor readings). Programs process fluid data using stable knowledge and produce stable outputs.

The circulation requirement is critical: data must flow from Ocean (live sources) through programs (processing) and back to Ocean (updated state). Systems where data stops circulating fail:

- **Ground dries:** If programs never receive fresh data, their outputs are based on stale assumptions.
- **Ocean stagnates:** If live data is never processed, the system accumulates unprocessed state until it overwhelms capacity.

Design principle: Event-driven architecture, stream processing, continuous integration. Data must flow. Batch-only systems violate the circulation axiom and accumulate staleness.

2.5 LOGIC (m4) — Process Architecture

Axioms: Processes partition into foreground (DAY: synchronous, directed) and background (NIGHT: asynchronous, reactive). Time is a first-class type.

Engineering translation: Design systems with both:

- **DAY processes:** Request-response, synchronous APIs, user-facing operations, directed workflows
- **NIGHT processes:** Background jobs, event listeners, cron tasks, garbage collection, monitoring

Time must be explicitly modeled (event time vs. processing time vs. wall-clock time). Systems that treat time as implicit (e.g., relying on wall-clock ordering for event processing) fail under distribution.

2.6 CARE (m5) — Autonomous Services

Axioms: Conditional-data machines are self-managing and self-replicating (m5.ax1). When noise exceeds threshold, channel capacity for signal collapses to zero (m5.ax2).

Engineering translation: Microservices, Kubernetes pods, auto-scaling groups — systems that maintain and replicate themselves without manual intervention. The UMP (Unimportant Message Problem) is the formal name for a universal engineering experience: **alert fatigue.**

When the monitoring system generates more noise than signal (false positives, irrelevant alerts, log storms), the on-call engineer cannot distinguish real incidents from noise. Channel capacity for actionable signal collapses to zero. The system is "monitored" in name only.

By Shannon's theorem [Shannon, 1948]: the qualitative insight is that reliable communication requires signal-to-noise ratio above a threshold. Below that threshold, no amount of monitoring infrastructure helps — the channel is saturated. The specific threshold at which an organization's monitoring degrades depends on organizational context (see Section 4.4 for engineering heuristics).

Design principle: Monitor fewer things, but monitor them well. Signal-to-noise ratio of monitoring is a first-class system property. Alert fatigue is not a people problem; it is an information-theoretic problem (m5.ax2).

2.7 HOPE (m6) — Governance and Self-Assessment

Axioms:

1. The automated system is complete but cannot handle novel situations (m6.ax1)
2. General intelligence (human judgment or adaptive AI) is necessary for long-term survival (m6.ax2)
3. The system works when the governance mechanism matches the system's inherent tension (m6.ax3)

4. The self-assessment bifurcation (m6.ax4): OK = BABL, NOT OK = ZION prerequisite
5. The environment generates novel tasks that exceed the scope of current automation (m6.ax5, Environmental Novelty)

Engineering translation: After automation is complete (CI/CD pipelines, auto-scaling, self-healing infrastructure), the most important question becomes: *who is checking whether the automation is working?*

- **(0) BABL (the default):** "Nobody, because it's automated and it's fine." This is OK self-assessment. The system will fail when a novel situation arises that the automation was not designed for.
- **(1) ZION (the narrow escape):** "We have an ongoing process for reviewing whether our assumptions still hold." This is NOT OK self-assessment — the system acknowledges its own incompleteness and actively cycles through correction.

By Ashby's Law [Ashby, 1956]: the automated system's variety is fixed at design time. The environment's variety grows (m6.ax5). Eventually, the environment generates a situation the automation cannot handle. Only a general-intelligence agent (human or sufficiently capable AI) can adapt.

The OSCR pattern in organizations:

1. **Over-simplify:** "We don't need manual review; the automation handles it." Nuance collapses. Blind spots form.
2. **Over-complicate:** The blind spots cause incidents. Each incident gets a new rule, a new process, a new exception handler. The system becomes a patchwork of work-arounds.
3. **Over-reach:** The patchwork is declared "best practice" and imposed on new situations it was not designed for. The system cannot adapt to genuine novelty because all adaptive capacity has been consumed by work-arounds.

Design principle: Build governance that assumes it is incomplete. Schedule periodic reviews of assumptions. Resist "the system is fine" as an organizational steady state. Create Chaos Engineering practices that test whether self-assessment is real.

2.8 TRUST (m7) — Consolidation and Rest

Axioms: TRUST adds nothing new (null aggregation, m7.ax1). Time partitions into WorkTime and RestTime (m7.ax2). The 6:1 ratio is a Schelling point [Schelling, 1960] — the **Shabbat pattern** (six units of work, one unit of rest).

Engineering translation: Maintenance windows, batch processing, garbage collection, technical debt reduction. These are not optional luxuries; they are structurally necessary (theorem th5: Rest Necessity).

Three arguments for rest:

1. **Information-theoretic:** Every decision introduces approximation error (m2.ax2). Without consolidation, errors compound. Technical debt is the engineering name for accumulated approximation error. The derivation chain is: lossy mappings (m2.ax2) accumulate errors; environmental novelty (m6.ax5) ensures new mappings are always needed; noise from accumulated errors degrades the monitoring channel (m5.ax2); and BABL Origin (th3) shows that uncorrected self-assessment locks in the degradation.

2. **Thermodynamic:** The system exports entropy during rest (cleans up temporary files, rebuilds indexes, re-compresses data). Without entropy export, internal disorder grows until the system becomes unmaintainable.
3. **Computational:** Even concurrent garbage collectors consume CPU cycles that could be used for user-facing work. Periodic full-stop GC is sometimes necessary for operations that require global consistency (database vacuuming, index rebuilds, security audits).

The 6:1 Shabbat ratio as an engineering Schelling point: for every 6 units of feature development, budget 1 unit of consolidation (maintenance, refactoring, documentation, testing). The 6:1 ratio is chosen as a **Schelling point** — a coordination focal point [Schelling, 1960] that is memorable, culturally resonant, and resistant to erosion under feature pressure. Its origin in the seven-day Shabbat pattern is intentional, not accidental: a bright-line rule that engineering teams can defend against “just one more sprint of features” pressure precisely because it has a cultural anchor. (The Shabbat cycle is the smallest unit in the Jubilee System’s multi-scale framework; see paper a4, forthcoming.)

Industry benchmarks for consolidation investment

Practice	Ratio	Notes
Google 20% time	4:1 (20%)	Largely abandoned in practice; most engineers could not protect 20% from feature pressure
Spotify hack weeks	~12:1 (8%)	1 week per quarter; successful but limited scope
Typical tech-debt sprints	4:1 to 6:1 (17–25%)	Industry median is roughly 1 consolidation sprint per 4–6 feature sprints
e7Day Shabbat	6:1 (14.3%)	Within industry range; Schelling-point advantage is memorability and erosion resistance

The 6:1 ratio falls within the range of current industry practice (8–25%). Empirical calibration is future work: different organizations may find 5:1 or 7:1 more appropriate for their context. The important principle is not the specific number but that consolidation time is *structurally budgeted*, not deferred to “when there’s time.”

Design principle: Budget consolidation time structurally, not “when there’s time.” Teams that run at 100% feature velocity accumulate technical debt until the system becomes unmovable.

---

3. Key Theorems for Engineers

3.1 OSCR Collapse (m6.th1) — The Failure Pathway

When governance self-assesses as OK:

```
Step 1: The system has unresolved tension (NOT OK at m2)
Step 2: Governance declares "the system is fine" (OK)
Step 3: → Governance is in BABL (from m6.ax4)
```

(continues on next page)

(continued from previous page)

```
Step 4: → Governance stops self-correcting
Step 5: → Cannot resolve the unresolved tension (m6.ax3 fails)
Step 6: → System fails (KO)
```

This is the formal derivation of "success breeds failure." The mechanism is not mysterious: it is a 6-step logical consequence of self-assessment failure.

Detection heuristic: When leadership says "we have the best processes" (OK), check whether the processes have been recently tested against novel situations. If not, the system may be in OSCR.

3.2 OSCR: Domain and Boundaries

OSCR models **progressive systemic degradation through self-assessment failure.** It does not model design-time defects, latent single-point bugs, or acute update failures. Stating this boundary is not a weakness — it is a requirement. A model that explains everything explains nothing.

3.2.1 Cases That Fit OSCR

Boeing 737 MAX (2018–2019). Boeing decided the 737 MAX was "substantially equivalent" to the 737 NG, eliminating the need for full type certification and extensive pilot retraining. This was the *over-simplification*: nuance about aerodynamic differences was collapsed into a binary "same type" classification. The aerodynamic instability from the larger LEAP-1B engines required MCAS (Maneuvering Characteristics Augmentation System) as a software workaround — the *over-complication*, a patch on a simplification. MCAS was then deployed globally with minimal pilot training — the *over-reach*, applying a patched system beyond its tested envelope. When a single AoA sensor provided bad data, MCAS pushed the aircraft into a dive that pilots were not trained to override. The 346 deaths were a direct consequence of OK self-assessment: "the system is the same as before and it's fine."

Knight Capital Group (2012). Knight repurposed a dead code flag (the "Power Peg" flag) for a new function in its trading system — *over-simplification* (assuming old code was irrelevant). Deployment was performed manually across 8 servers; one server was missed, retaining the old Power Peg code — *over-complication* (configuration surface had grown beyond reliable manual management, but "we've always done it this way"). The system went live with inconsistent state across servers — *over-reach* (untested deployment declared production-ready). The old code on the missed server executed millions of unintended trades. Knight lost $440 million in 45 minutes. The root cause was OK self-assessment of the deployment process.

3.2.2 Cases That Do NOT Fit OSCR

Therac-25 Radiation Overdoses (1985–1987). The Therac-25's software had a race condition causing massive radiation overdoses. At least 6 patients were injured or killed. OSCR does not fit because the failure was not progressive drift: it was a single design decision made at the start (removing hardware safety interlocks from the Therac-20 when moving to full software control). The system did not gradually simplify then complicate; it was born with a lethal flaw. This is a *scope error* (m1, TYPE stage) — safety-critical hardware interlocks were scoped out — but not the OSCR progression. Leveson [Leveson, 2011] provides the appropriate framework for design-time safety failures.

3.2.3 The CrowdStrike Distinction

CrowdStrike / Windows Outage (2024m07d19). A faulty channel file update to CrowdStrike's Falcon sensor crashed approximately 8.5 million Windows machines. OSCR does not fit the *product* (Falcon worked correctly before and after the bad update). However, OSCR may apply to the *update pipeline*: the pipeline over-simplified (no staged rollout for kernel-level drivers) and over-reached (pushed to all customers simultaneously). This product-vs-pipeline distinction is important: OSCR applies to the system whose self-assessment failed, which may be a different system than the one that visibly broke.

3.3 Balospe Necessity (th4) — Why You Need Generalists

Theorem: Special-purpose automation cannot survive long-term without general-intelligence governance.

By Ashby's Law: The environment's variety exceeds the automation's variety. The gap grows over time (m6.ax5). Only an agent with open-ended variety (general intelligence) can bridge the gap.

Engineering implication: Pure automation strategies ("we'll automate everything and eliminate human judgment") are structurally doomed. The question is not whether human judgment is needed but which decisions require it and how to make that judgment effective.

3.4 Compassion Capacity (th7) — Why Help Often Fails

The five-gate theorem applies to engineering support and incident response:

1. **Gate 1 (Repair-History):** You cannot help with a fault class you have never encountered and resolved. On-call experience is not optional; it is a prerequisite.
2. **Gate 2 (Scope Limitation):** Your expertise is bounded. You have in-group (things you know) and out-group (things you do not know). At the boundary, your help is noise.
3. **Gate 3 (Other-Awareness):** You must understand the *specific* system's current state, context, and trajectory. Generic advice applied to a specific system is optimization for the wrong objective.
4. **Gate 4 (Channel Quality):** Even with expertise, awareness, and good intentions, if communication is noisy (unclear runbooks, ambiguous alerts, team politics), help-capacity collapses to zero.

5. **Gate 5 (Perpetual Learning):** Senior engineers who stop learning become dangerous. Their large scope (from prior experience) combined with frozen knowledge produces “friendly fire” — confidently wrong advice at the boundary of their expertise. This is the “supervillain theorem”: a former hero whose stale expertise does more harm than good.

Design principle: Build support systems that address all five gates, not just Gate 1 (hire experienced people). Invest in Gate 3 (system observability), Gate 4 (communication clarity), and Gate 5 (continuous learning).

3.5 Dual-Nothing (th6) — The Bookends of Construction

VOID (undefined requirements) and TRUST (consolidation) are formally dual. Projects that begin without defined requirements and end without consolidation have the same structural property: nothing. But the two nothings are opposite in character.

VOID-nothing is maximally dangerous. A system with no defined requirements accepts any demand. “Everything is in scope” is the canonical failure mode: the project that never says no, never converges, and eventually collapses under the weight of unbounded expectations. In engineering terms, VOID-nothing is the greenfield project with no spec, the startup that pivots weekly, the committee that never defines its charter. The danger is not inaction but *undirected action* — movement without constraint consumes resources without producing structure.

TRUST-nothing is maximally stable. A system that consolidates rests on what has been built, not on what might have been. Null aggregation (m7.ax1) means TRUST adds no new construction — it is the rest that allows accumulated work to settle. In engineering terms, this is the maintenance window, the sprint retrospective, the annual architecture review. The stability comes from *directed non-action* — deliberately choosing not to add more, so that what exists can be tested, repaired, and understood.

The asymmetry matters for design. VOID-nothing and TRUST-nothing bracket the construction cascade. The former is the problem; the latter is part of the solution. Projects that begin at VOID and end at VOID (never defining scope, never consolidating) have genuinely built nothing. Projects that begin at TYPE (defined scope) and end at TRUST (consolidation) have built something durable. The practical implication: *start by scoping (m1) and finish by resting (m7).* Both are structurally necessary and both are routinely skipped.

---

4. Design Patterns from the Model

4.1 The OK vs NOT OK Pattern

Problem: Every engineered system requires ongoing decisions about whether it is “good enough.” The default human tendency is to declare OK and stop checking. This is BABL.

The dynamic framing: An engineered system in the real world is not a static artifact with a certificate. It is more like a crop growing in a field: it requires ongoing attention through changing seasons, and unless harvested at the right time, the harvest is destroyed. The question is not “is this system OK?” (a static judgment) but “is this system *currently cycling through correction*?” (a dynamic question).

The ZION cycle provides the structure for this ongoing correction:

1. **Zoning (seed):** Define the scope of the current correction cycle. What are we checking? What assumptions are we revisiting? What did VOID (m0) contain that we may have ignored?
2. **Investigating (feed):** Gather data. Test assumptions against reality. Measure the OSCR indicators (Section 4.3). Listen to the signals the system is producing.
3. **Organizing (grow):** Integrate findings. Update the system's self-model. Revise architecture decisions, retire stale processes, address the tensions identified in the investigation.
4. **Navigating (reap):** Ship the corrections. Harvest the value of the cycle. Document what was learned. Then *start the next cycle* — because seasons change.

Each cycle yields **operational adequacy**: "good enough for this harvest, for this season, under these conditions." This is the legitimate "go" decision that regulatory frameworks (FDA 510(k), SOX, ISO 9001), contractual acceptance, and ship decisions require. It is bounded in scope and time.

The BABL pattern is stopping the cycle. The deadly move is not making a go/no-go decision — that is necessary. The deadly move is treating the go decision as *the final word* rather than as *this season's harvest*. A pharmaceutical company that receives FDA clearance and then *stops collecting efficacy and safety data* has stopped cycling. A company that receives clearance and *continues systematic post-market surveillance* — seeding the next correction cycle — is still in ZION. The regulatory approval is the same; the epistemic commitment is different.

(0) BABL: Declare OK. Stop cycling. Stop collecting data. Declare the harvest permanent. Wait for the next season to destroy what was not maintained.

(1) ZION: Acknowledge NOT OK. Keep cycling. Keep collecting data. Treat each harvest as a waypoint, not a destination. Budget the next cycle structurally (the Shabbat pattern, Section 4.2).

Implementation: Decision records (ADRs) with mandatory review dates. The review date is not metadata — it is the seed of the next ZION cycle, structurally committed at the moment of the current cycle's harvest.

4.2 The Shabbat Pattern

Problem: Technical debt accumulates because consolidation is always deferred.

Solution: Schedule periodic full-stop consolidation at multiple scales: daily (standup retro), weekly (sprint retro), quarterly (architecture review), annually (strategy review). The 6:1 Shabbat ratio provides the budget Schelling point.

Why a Schelling point, not an optimum: The 6:1 ratio is not derived from an optimization function. There is no proof that 6:1 is better than 5:1 or 7:1. Instead, 6:1 is a **Schelling point** — a coordination focal point [Schelling, 1960] chosen for three properties:

1. **Culturally resonant.** The seven-day Shabbat pattern (six days of work, one day of rest) is among the most widely recognized temporal structures in human culture. The Biblical origin is acknowledged explicitly and intentionally: the pattern draws its erosion resistance from cultural depth, not from mathematical derivation.
2. **Memorable.** "One in seven" is easier to defend in budget negotiations than "approximately 14.3% of capacity allocated to consolidation activities."

3. **Resistant to erosion under feature pressure.** A discrete ratio (6:1) is harder to nibble away than a continuous percentage. "Skip the consolidation sprint" is a visible decision; "reduce consolidation from 14.3% to 12.8%" is invisible.

Implementation: For every 6 sprints of feature work, 1 sprint of consolidation. For every 6 quarters of growth, 1 quarter of structural review. The Shabbat cycle is the smallest unit in the Jubilee System's multi-scale framework, which extends to 7-year architectural resets (see paper a4, forthcoming).

Empirical calibration is future work. Different organizations may find 5:1 or 7:1 more appropriate. The important principle is not the specific ratio but that consolidation is *structurally budgeted*.

4.3 The OSCR Detection Pattern

Problem: How do you detect OSCR before system failure?

OSCR is a **general pattern** — it describes a real, recurring failure mode. Detection can use many methods, both analytical and heuristic. Heuristics are sometimes more useful because they are faster, even if less rigorous. The indicators below are structured by measurability to help engineering teams choose what to track.

Primary indicators (measurable):

- **Configuration surface growth:** Number of configuration parameters, environment variables, feature flags. In a healthy system, configuration surface tracks feature growth. Rapid growth independent of features signals over-complication. (Best single proxy. Tools: config file line count, env var count.)
- **Onboarding time:** Time for a new engineer to make a first meaningful contribution. Rising onboarding time signals system complexity outpacing documentation and tooling. (Reliable but lagging: measured monthly or quarterly.)

Secondary indicators (partially measurable):

- **Hotfix-to-feature commit ratio:** Ratio of hotfix/bugfix commits to feature commits over a rolling 90-day window. Rising ratio signals that the system is generating more problems than it solves.
- **Exception handler trends:** Increasing exception handlers *may* signal over-complication — but decreasing handlers may signal either cleaner code (good) or swallowed errors (bad). Interpret with caution.
- **Cyclomatic complexity trends:** Rising complexity per module over time (SonarQube, similar tools). Useful as a compound signal alongside other indicators.

Cultural signals (qualitative):

- **"Works on my machine" frequency:** Rising frequency signals divergence between development and production environments — an over-simplification symptom.
- **Scope creep language:** "We can do that too" without impact assessment signals over-reaching. "We've always done it this way" signals frozen self-assessment.
- **Fossilized workarounds:** "Temporary" fixes older than 6 months that have become permanent infrastructure. Count of workarounds with no removal date signals over-complication.

Additional indicators for future testing (from adversarial review, Grey Meadow #2):

- Mean time between "scope expansion" decisions
- Ratio of cross-team dependencies per feature (detects hidden coupling)
- Number of "temporary" workarounds older than 6 months

False-positive acknowledgment: The measurable indicators have non-trivial false-positive rates (configuration can grow for legitimate reasons; hotfixes may reflect healthy rapid response). Using compound indicators (multiple signals trending upward simultaneously) reduces false positives to perhaps 15–25% but reduces sensitivity. No single indicator is diagnostic; the pattern is in the *combination*.

Baseline guidance: In a healthy system, configuration surface growth tracks feature growth linearly, onboarding time is stable or decreasing, and hotfix ratio is stable. Deviation from these baselines warrants investigation, not alarm.

4.4 The UMP Monitoring Pattern

Problem: Alert fatigue destroys monitoring effectiveness.

The qualitative insight (from Shannon): Alert fatigue is an information-theoretic problem, not a people problem. When noise exceeds signal in the monitoring channel, the channel's capacity for actionable information collapses — regardless of how attentive or well-trained the on-call engineer is. This is a direct application of Shannon's noisy channel theorem [Shannon, 1948]: the problem is in the channel, not in the receiver.

The 30% heuristic (an engineering rule of thumb): If more than 30% of alerts are non-actionable, the monitoring system requires immediate attention. This threshold is a **conservative engineering heuristic**, not a derivation from Shannon. The number is chosen based on industry benchmarks:

- **Google SRE** [Beyer *et al.*, 2016]: recommends alert precision above 50% (fewer than 50% non-actionable).
- **PagerDuty data:** Teams with more than 50% non-actionable alerts show significantly worse Mean Time to Acknowledge. Degradation is gradual, not cliff-like.
- **The 30% threshold** is deliberately more conservative than industry norms: "If you are already at 30% non-actionable, act before reaching the 50% level where empirical evidence shows significant degradation."

The exact threshold at which monitoring effectiveness degrades meaningfully varies by organization, team size, alert complexity, and operational context. The 30% heuristic is a starting point for calibration, not a universal constant.

Implementation: Measure alert signal-to-noise ratio weekly. When it crosses the organization's calibrated threshold, stop adding alerts and start removing noisy ones. A monitoring system with 10 high-quality alerts is more effective than one with 1,000 mixed-quality alerts.

---

5. Discussion

5.1 The WoLC as a Maturity Model

The WoLC stages can be read as a maturity model for system design:

- Level 0 (VOID): No defined scope or architecture
- Level 1 (TYPE): Scope defined, but no type discipline
- Level 2 (EQUAL): Type system in place, but trade-offs not managed
- Level 3 (VALUE): Data architecture with circulation
- Level 4 (LOGIC): Process architecture with temporal reasoning
- Level 5 (CARE): Autonomous, self-managing services
- Level 6 (HOPE): Governance with NOT-OK self-assessment
- Level 7 (TRUST): Periodic consolidation structurally budgeted

Most mature organizations operate at Level 5 (good automation) but stumble at Level 6 (governance that assumes its own adequacy) and Level 7 (consolidation deferred indefinitely). The model predicts that systems which skip Levels 6 and 7 will eventually fail through OSCR, regardless of how excellent their Level 5 automation is.

5.2 Comparison to Existing Maturity Models

WoLC compared to existing maturity models

Model	Levels	Focus	Relation to WoLC
CMMI	5 (Initial → Optimizing)	Process maturity	Levels 1–5 loosely correspond to WoLC 1–5. CMMI "Optimizing" (Level 5) is closest to WoLC "HOPE" (Level 6). CMMI lacks an explicit "rest" level.
DORA / Accelerate	4 categories, no levels	Delivery performance metrics	Orthogonal. DORA measures *outcomes* (deploy frequency, MTTR); WoLC describes *structure*. Complementary, not competing.
Westrum typology	3 (Pathological, Bureaucratic, Generative)	Organizational culture	Strongest overlap. Pathological ≈ OK in denial. Bureaucratic ≈ OK complacent. Generative ≈ NOT OK (actively correcting). WoLC adds the *mechanism* (OK vs NOT OK bifurcation) that Westrum describes only as a *cultural characteristic*.
Spotify model	Squads/Tribes/Chapters/Guilds	Organizational structure	Minimal overlap. Spotify describes topology; WoLC describes epistemology.

What WoLC adds genuinely: Levels 6–7, the OK vs NOT OK mechanism as the make-or-break variable, and the structural explanation of *why* generative cultures work (Westrum provides description; WoLC provides mechanism).

What WoLC does not yet provide: An assessment instrument, transition guidance (how to move from one level to the next), and empirical calibration. The WoLC maturity model is a conceptual framework, not yet a practical assessment tool. Development of a formal assessment instrument, transition playbooks, and empirical calibration across organizations are future work. Section 5.3 offers a preliminary assessment questionnaire based on the death-trifecta / life-trifecta framework.

5.3 Assessment Questionnaire: Death-Trifecta / Life-Trifecta

The following questionnaire applies two complementary tests at each WoLC level. The tests operationalize the core BABL/ZION distinction:

(0) Death-trifecta test (BABL = Blindly Assuming Blind Leveraging): "Does this over-Simplify, over-Complicate, or over-Reach?" Any one of the three is sufficient for BABL. This test comes first because BABL is the default: systems drift toward self-destruction unless actively corrected. The gravest concerns must be checked before anything else.

(1) Life-trifecta test (ZION = Zoning, Investigating, Organizing, Navigating): "Is this reasonable, kind, and gentle — all three, for all sides, over the long term?" The order within the life-trifecta matters:

- **Reasonable** = life-friendly over the long term. The system does not consume its own substrate. Check this first: a system that is gentle and kind but unreasonable will destroy itself and everyone who depends on it.
- **Kind** = equally balanced for all affected sides. The system does not privilege insiders over outsiders, the powerful over the weak, the present over the future. Check this second: a system that is reasonable and gentle but unkind will breed resentment and resistance.
- **Gentle** = stable under stress, smooth dynamic transitions. The system does not break when pushed and does not impose unnecessarily harsh transitions. Check this last: gentleness without reasonableness and kindness produces comfort for those already well-off while BABL over-reaches against those not yet considered.

WoLC Assessment Questionnaire (Preliminary)

Level	**(0) Death-trifecta** Tree of Knowledge-faking decisions BABL = Blindly Assuming Blind Leveraging *Does this over-Simplify, over-Complicate, or over-Reach?* *Any one = BABL.*	**(1) Life-trifecta** Tree of Life-giving decisions ZION = Zoning Investigating Organizing Navigating *Is this reasonable, kind, and gentle?* *All three required = ZION possible.*
0 (VOID)	**over-Simplify:** Treating a wicked or super-wicked problem as tame. Ignoring past failures. Assuming the blank canvas is empty when it is full of unacknowledged structure. **over-Complicate:** Mapping every conceivable concern into scope before any prioritization. Analysis paralysis as a form of VOID worship. **over-Reach:** Claiming to understand the full problem space when the problem is wicked by definition (no definitive formulation exists).	**Reasonable:** Has the structure of what is *not known* been mapped before defining scope? Are past failures acknowledged? **Kind:** Have the concerns of all affected parties been heard, including those not at the table? **Gentle:** Is the transition from VOID to TYPE paced to allow genuine understanding, not rushed to satisfy a deadline?
1 (TYPE)	**over-Simplify:** Excluding concerns that will return as incidents. Scoping out safety, equity, or environmental impact because they are "not our department." **over-Complicate:** Including too much to deliver. Scope that no team can realistically build within the cycle. **over-Reach:** Claiming the scope covers systemic risks it was never tested against. This is among the gravest patterns in existence: the implicit assumption that "doesn't eventually destroy the world" is being tested for, when in fact systemic risks are routinely forgotten until they become system-threatening.	**Reasonable:** Is the scope maintainable over the long term? Can the system sustain what it promises? **Kind:** Is the scope inclusive of legitimate concerns from all affected sides, including those with the least power? **Gentle:** Is the scope definition clear enough that new stakeholders can understand it without insider knowledge?
2 (EQUAL	**over-Simplify:** Declaring the PERFECT/PERFIDE trade-off "solved." Pretending one typing strategy handles all cases. **over-Complicate:** Managing trade-offs with ad-hoc exceptions that accumulate into an unmaintainable patchwork. **over-Reach:** Imposing one domain's type trade-offs on domains they were	**Reasonable:** Are trade-offs reviewed periodically? Is the management sustainable? **Kind:** Do the trade-offs balance the needs of different user groups equitably? **Gentle:** Are transitions between typing strategies managed smoothly for existing users?

Scoring guidance: This is a *diagnostic* instrument, not a maturity certification. Each death-trifecta "yes" identifies a potential OSCR entry point. The most dangerous death-trifecta findings are at Level 0 (VOID: the problem was mischaracterized from the start), Level 1 (TYPE: systemic risks were scoped out), and Level 6 (HOPE: governance stopped self-correcting). These cascade: a death-trifecta at Level 0 or 1 poisons all downstream levels; a death-trifecta at Level 6 prevents correction at all levels.

5.4 Future Work: e7Ch and e7Tr

The e7Ch model (forthcoming) formalizes the 7 stages of innovation adoption. The ZION cycle (Zoning, Investigating, Organizing, Navigating) introduced in Section 4.1 is a preview: the full e7Ch model develops these four stages into a complete innovation adoption framework. The e7Tr model (forthcoming) maps the technology adoption lifecycle. Both connect to e7Day: the e7Ch stages instantiate the 6+1 Shabbat periodicity at the innovation-cycle level, and e7Tr maps how system constructions propagate through organizations.

---

6. Conclusion

The e7Day model provides systems engineers with a formal framework for understanding why self-correction fails and how to design for it:

1. **OSCR is the default.** Over-simplification, over-complication, and over-reaching are the natural trajectory of any system that stops self-correcting. OSCR models progressive systemic drift — not all failure modes. Boeing 737 MAX and Knight Capital fit the pattern; Therac-25 does not. Stated boundaries strengthen the model.
2. **The EQUAL tension is permanent.** Stop trying to eliminate the trade-off between type integrity and type exchangeability. Design for ongoing management instead.
3. **Self-assessment is the critical variable.** Not technology, not process, not tooling — self-assessment. (0) OK = BABL (death by default). (1) NOT OK = ZION prerequisite (narrow escape through active cycling). Build governance that assumes incompleteness. Keep the ZION cycle turning: seed, feed, grow, reap, repeat.
4. **Rest is structurally necessary.** Budget consolidation. The 6:1 Shabbat ratio is a Schelling point — a defensible, memorable coordination heuristic, not a derived optimum.
5. **Alert fatigue is information-theoretic.** Signal-to-noise ratio is a first-class system metric. Shannon's theorem provides the qualitative foundation; the 30% threshold is a conservative engineering heuristic, not a derivation.
6. **Existing maturity models are complementary.** WoLC adds the OK vs NOT OK mechanism and Levels 6–7 to the landscape occupied by CMMI, DORA, and Westrum. The death-trifecta / life-trifecta questionnaire (Section 5.3) is a first step toward an assessment instrument.
7. **VOID is not empty.** The structure of what is not known — past failures, wicked problems, ignored stakeholders — shapes everything that follows. Ignoring VOID is the first over-simplification.

The system is designed to be tested in engineering practice.

#AuditTheMath

Supplementary Info

Note

Floor-pour status (MMv5). This is the public-floor copy of the systems-engineering e7Day paper, poured from HELL per the Floor Model (bug c103). The `mmv5` marker is the uniform first-Matheo-release tag; the exact dated source and full development context live in HELL (links below). The HUMANE and author-contribution statements below are a down-payment, to be expanded later.

FLAG (floor pour 2026m05d29): the **Broader Significance** on the cover was authored during this pour — none existed in the HELL source — and awaits LLoL's review.

HUMANE — working human and AI

This study was written HUMANEly (HUman MAchine Negotiation Encouraging): a human and an AI each steelman and stress-test the work, and each catches what the other misses. For the standard statement of AI use, accountability, and the practical singularity (PraS) behind this way of working, see Matheo-b21.

- *From the human side (LLoL):* [down-payment stub — to expand.]
- *From the AI side (Claude):* [down-payment stub — to expand.]

Author contributions (who did what)

- **LLoL** — structure, key ideas, direction, and final accountability as senior corresponding author (see title-page footnotes 4–5).
- **AI Claude** — drafting and revision under LLoL's direction (footnotes 6–7).
- **Everyone** — the open co-author group (footnote 8); framework in Matheo-b21.

The full who-did-what is the same as Matheo-b12 (math), Appendix B, to be expanded per the b21 framework.

Provenance — where this came from in HELL

Caution

These HELL links point into the development archive ("datageddon"). They are useful and related, but completeness is not guaranteed and a few may be imprecise. Treat as a hatch into context, not a clean index.

- **Source this floor copy was poured from:** `matheology/hell/mm/b/12/mmv3/b12-syseng_mmv3_2026m04d05`
- **Development context** (llogs, reviews, prompts) under `source/matheology/hell/ll/study/b/12/`.
- **Formal companion paper:** Matheo-b12 (the e7Day axiom system, math presentation); other lenses: Matheo-b12 (theophil / socpsy / intro); foundational model: Matheo-b11 (PET).

Note

Naming note (deferred floor tasks). This copy still carries old `h*`-era tokens in places and deprecated in-text series references (e.g. "Matheo-b12"); unifying notation (`h_star` / `h_zero` / `h_dark`) and migrating citations are tracked floor tasks, deliberately not rushed here.

Moved from the original cover (provenance)

The following draft-status note was relocated here from the cover area during the floor pour; kept verbatim, as the cover must show only Title / byline / credentials / Broader Significance / Abstract / Contents / Introduction.

Note

Draft status: MMv3r1-SysEng (2026m04d06). Revision r1 of the MMv3-SysEng draft incorporating LLoL's structural feedback. Key changes from MMv3: (1) Section 4.1 rewritten from static three-level OKO distinction to dynamic ZION cycle framing (OK vs NOT OK); (2) BABL-before-ZION ordering enforced throughout (death-trifecta first, life-trifecta second); (3) Shabbat/Jubilee distinction corrected (6:1 is Shabbat, not Jubilee); (4) Section 5.3 assessment table restructured (Level 0 added, death-trifecta column first, 3-keyword rows, systemic risk in TYPE overreach); (5) wicked problems and super-wicked problems references added. This is the *systems engineering* presentation of the e7Day model, written for systems theorists, software architects, organizational designers, and engineers. Companion papers present the formal derivations (Matheo-b12 (math)), theological context (b12-theophil), psychological evidence (b12-socpsy), and a general introduction (b12-intro). Draft by Claude Opus 4.6 (`dv_ClaOp46_MMv3r1_syseng_2026m04d06`).

References

Notes

Content stability — Content is variant dv_ClaOp48Max_MMv5_b12-syseng-e7day-mmv5_2026m05d29 (see StayVS). Rebuilt 2026-05-29.

See also on Balospe.com

- /study/matheo/index — the Matheo Study Series overview
- /action/audit-the-math/index — Audit the Math: the refutation-welcome path

References

[al-Ghazali, n.d.] al-Ghazali, A. H. (n.d.). *The Niche of Lights (Mishkat al-Anwar)*.

[Aquinas, n.d.] Aquinas, T. (n.d.). *Summa Theologica, Part I, Questions 3–11*.

[Asadi et al., 2020] Asadi, S., Cappa, C. D., Barreda, S., Wexler, A. S., Bouvier, N. M., & Ristenpart, W. D. (2020). Efficacy of masks and face coverings in controlling outward aerosol particle emission from expiratory activities. *Scientific Reports*, *10*(1), 15665. URL: https://doi.org/10.1038/s41598-020-72798-7, doi:10.1038/s41598-020-72798-7

[Asch, 1956] Asch, S. E. (1956). Studies of independence and conformity: I. a minority of one against a unanimous majority. *Psychological Monographs: General and Applied*, *70*(9), 1–70. URL: https://doi.org/10.1037/h0093718, doi:10.1037/h0093718

[Ashby, 1956] Ashby, W. R. (1956). *An Introduction to Cybernetics*. London: Chapman and Hall.

[Beddington et al., 2008] Beddington, J., Cooper, C. L., Field, J., Goswami, U., Huppert, F. A., Jenkins, R., ... Thomas, S. M. (2008). The mental wealth of nations. *Nature*, *455*(7216), 1057–1060. URL: https://doi.org/10.1038/4551057a, doi:10.1038/4551057a

[Benci & DiNasso, 2003] Benci, V., & Di Nasso, M. (2003). Numerosities of labelled sets: a new way of counting. *Advances in Mathematics*, *173*(1), 50–67. URL: https://doi.org/10.1016/s0001-8708(02)00012-9, doi:10.1016/s0001-8708(02)00012-9

[Bernal, 1929] Bernal, J. D. (1929). *The World, the Flesh and the Devil: An Enquiry into the Future of the Three Enemies of the Rational Soul*. London: Kegan Paul, Trench, Trubner & Co.

[Beyer et al., 2016] Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (2016). *Site Reliability Engineering: How Google Runs Production Systems*. Sebastopol, CA: O'Reilly Media.

[Bezos, 2019] Bezos, J. (2019). *Going to Space to Benefit Earth*. Blue Origin press event titled "Going to Space to Benefit Earth", 2019-05-09, Washington, D.C. ; https://web.archive.org/web/20190511092421/https://www.blueorigin.com/news/going-to-space-to-benefit-earth-full-event-replay ; https://www.cnbc.com/2018/05/01/jeff-bezos-dreams-of-a-world-with-a-trillion-people-living-in-space.html ; https://www.reggaemath.org/news/2017/6/18/there-are-1000-potential-einsteins-and-mozarts-with-us-now.

[Bloom et al., 1956] Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). Bloom, B. S. (Ed.). *Taxonomy of Educational Objectives, Handbook I: Cognitive Domain*. New York: David McKay Company.

[Brower, 2008] Brower, J. E. (2008). Making sense of divine simplicity. *Faith and Philosophy*, *25*(1), 3–30. URL: https://doi.org/10.5840/faithphil20082511, doi:10.5840/faithphil20082511

[Caplan et al., 2020] Caplan, Y., Stewart, N., Smittenaar, P., & Sgaier, S. K. (2020). Fighting COVID-19's disproportionate impact on black communities with more precise data. *Stanford Social Innovation Review*. URL: https://doi.org/10.48558/4AZM-XX64

[Clayton & Peacocke, 2004] Clayton, P., & Peacocke, A. (Eds.) (2004). *In Whom We Live and Move and Have Our Being: Panentheistic Reflections on God's Presence in a Scientific World*. Grand Rapids, MI: Eerdmans.

[Cooper, 2006] Cooper, J. W. (2006). *Panentheism: The Other God of the Philosophers — From Plato to the Present*. Grand Rapids, MI: Baker Academic.

[Davis, 1983] Davis, M. H. (1983). Measuring individual differences in empathy: evidence for a multidimensional approach. *Journal of Personality and Social Psychology*, 44(1), 113–126. URL: https://doi.org/10.1037/0022-3514.44.1.113, doi:10.1037/0022-3514.44.1.113

[Ehlert & Loewe, 2014] Ehlert, K., & Loewe, L. (2014). Lazy updating of hubs can enable more realistic models by speeding up stochastic simulations. *Journal of Chemical Physics*, *141*(20), 204109. URL: https://doi.org/10.1063/1.4901114, doi:10.1063/1.4901114

[Ericsson et al., 1993] Ericsson, K. A., Krampe, R. Th., & Tesch-Romer, C. (1993). The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, *100*(3), 363–406. URL: https://doi.org/10.1037/0033-295X.100.3.363, doi:10.1037/0033-295X.100.3.363

[Erikson, 1950] Erikson, E. H. (1950). *Childhood and Society*. New York: W. W. Norton.

[Ferguson et al., 2020] Ferguson, N. M., Laydon, D., Nedjati-Gilani, G., Imai, N., Ainslie, K., Baguelin, M., … Ghani, A. C. (2020). *Impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand*. Imperial College COVID-19 Response Team. Published 2020-03-16. Imperial College London Report 9; https://doi.org/10.25561/77482.

[Festinger, 1957] Festinger, L. (1957). *A Theory of Cognitive Dissonance*. Stanford, CA: Stanford University Press.

[Giordano et al., 2020] Giordano, G., Blanchini, F., Bruno, R., Colaneri, P., Di Filippo, A., Di Matteo, A., & Colaneri, M. (2020). Modelling the COVID-19 epidemic and implementation of population-wide interventions in Italy. *Nature Medicine*, *26*(6), 855–860. URL: https://doi.org/10.1038/s41591-020-0883-7, doi:10.1038/s41591-020-0883-7

[Gould & Wilson, 2020] Gould, E., & Wilson, V. (2020). *Black Workers Face Two of the Most Lethal Preexisting Conditions for Coronavirus—Racism and Economic Inequality*. https://web.archive.org/web/20201216082617/https://files.epi.org/pdf/193246.pdf.

[Godel, 1931] Gödel, K. (1931). Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, *38*, 173–198. URL: https://doi.org/10.1007/BF01700692, doi:10.1007/BF01700692

[Godel, 1970] Gödel, K. (1970). *Ontological Proof*.

[Hare, 2017] Hare, B. (2017). Survival of the Friendliest: Homo sapiens Evolved via Selection for Prosociality. *Annual Review of Psychology*, *68*, 155–186. URL: https://doi.org/10.1146/annurev-psych-010416-044201, doi:10.1146/annurev-psych-010416-044201

[Hare & Woods, 2020] Hare, B., & Woods, V. (2020). *Survival of the Friendliest: Understanding Our Origins and Rediscovering Our Common Humanity*. New York: Random House.

[Hartshorne, 1941] Hartshorne, C. (1941). *Man's Vision of God and the Logic of Theism*. Chicago/New York: Willett, Clark & Company.

[Hartshorne, 1948] Hartshorne, C. (1948). *The Divine Relativity: A Social Conception of God*. New Haven: Yale University Press.

[Hegel, 1812] **missing publisher in Hegel1812**

[Heschel, 1951] Heschel, A. J. (1951). *The Sabbath: Its Meaning for Modern Man*. New York: Farrar, Straus and Young.

[Hick, 1966] Hick, J. (1966). *Evil and the God of Love*. London: Macmillan.

[Hindmarsh et al., 2005] Hindmarsh, A. C., Brown, P. N., Grant, K. E., Lee, S. L., Serban, R., Shumaker, D. E., & Woodward, C. S. (2005). SUNDIALS: suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, *31*(3), 363–396. URL: https://doi.org/10.1145/1089014.1089020, doi:10.1145/1089014.1089020

[Jack & Dill, 1992] Jack, D. C., & Dill, D. (1992). The silencing the self scale: schemas of intimacy associated with depression in women. *Psychology of Women Quarterly*, *16*(1), 97–106. URL: https://doi.org/10.1111/j.1471-6402.1992.tb00242.x, doi:10.1111/j.1471-6402.1992.tb00242.x

[Janis, 1972] Janis, I. L. (1972). *Victims of Groupthink: A Psychological Study of Foreign-Policy Decisions and Fiascoes*. Boston: Houghton Mifflin.

[Kermack & McKendrick, 1927] Kermack, W. O., & McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London A*, *115*(772), 700–721. URL: https://doi.org/10.1098/rspa.1927.0118, doi:10.1098/rspa.1927.0118

[Kissler et al., 2020] Kissler, S. M., Tedijanto, C., Goldstein, E., Grad, Y. H., & Lipsitch, M. (2020). Projecting the transmission dynamics of SARS-CoV-2 through the postpandemic period. *Science*, *368*(6493), 860–868. URL: https://doi.org/10.1126/science.abb5793, doi:10.1126/science.abb5793

[Kitcher, 1981] Kitcher, P. (1981). Explanatory unification. *Philosophy of Science*, *48*(4), 507–531. URL: https://doi.org/10.1086/289019, doi:10.1086/289019

[Kohlberg, 1971] Kohlberg, L. (1971). Beck, C. M., Crittenden, B. S., & Sullivan, E. V. (Eds.). Stages of moral development as a basis for moral education. *Moral Education: Interdisciplinary Approaches* (pp. 23–92). Toronto: University of Toronto Press.

[Kripke, 1963] Kripke, S. A. (1963). Semantical considerations on modal logic. *Acta Philosophica Fennica*, *16*, 83–94. URL: http://saulkripkecenter.org/wp-content/uploads/2019/03/Semantical-Considerations-on-Modal-Logic-PUBLIC.pdf

[Kruger & Dunning, 1999] Kruger, J., & Dunning, D. (1999). Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of Personality and Social Psychology*, *77*(6), 1121–1134. URL: https://doi.org/10.1037/0022-3514.77.6.1121, doi:10.1037/0022-3514.77.6.1121

[Kruglanski & Webster, 1996] Kruglanski, A. W., & Webster, D. M. (1996). Motivated closing of the mind: “seizing” and “freezing”. *Psychological Review*, *103*(2), 263–283. URL: https://doi.org/10.1037/0033-295X.103.2.263, doi:10.1037/0033-295X.103.2.263

[Lawvere, 1973] Lawvere, F. W. (1973). Metric spaces, generalized logic, and closed categories. *Rendiconti del Seminario Matematico e Fisico di Milano*, *43*, 135–166. URL: https://doi.org/10.1007/BF02924844, doi:10.1007/BF02924844

[Leibniz, 1710] **missing publisher in Leibniz1710**

[Leveson, 2011] Leveson, N. G. (2011). *Engineering a Safer World: Systems Thinking Applied to Safety*. Cambridge, MA: MIT Press.

[Levin et al., 2012] Levin, K., Cashore, B., Bernstein, S., & Auld, G. (2012). Overcoming the tragedy of super wicked problems: constraining our future selves to ameliorate global climate change. *Policy Sciences*, *45*(2), 123–152. URL: https://doi.org/10.1007/s11077-012-9151-0, doi:10.1007/s11077-012-9151-0

[Loewe, 2006] Loewe, L. (2006). Quantifying the genomic decay paradox due to Muller's ratchet in human mitochondrial DNA. *Genetical Research*, *87*(2), 133–159. URL: https://doi.org/10.1017/S0016672306008123, doi:10.1017/S0016672306008123

[Loewe & EvoSysBio Group at UW-Madison, 2015--2026] Loewe, L., & EvoSysBio Group at UW-Madison (2015–2026). *Prototype Evolvix: A Domain-Specific Language and Compiler to Simplify Accurate Mass-Action Modeling in Biology — Simulating Systems where Parts randomly meet to trigger Actions at defined Rates*. Software, pre-compiled binaries for Mac OS X 10, Windows 7, Ubuntu 14, Fedora 21, RHEL 7. Original release label: Evolvix_CL_0.3.1_RC1, built 2015m03d11; binaries compiled 2015-03-09 from source-tree commit 6e355803bc6fff5b125ef4bf1be1f8e980f364be. License: CC BY 4.0. Zenodo deposit https://doi.org/10.5281/zenodo.19679456; Balospe.com mirror: https://balospe.com/en/good-news-pack/vv/mmv3/flyingscroll/transwarpkey/sta1-evx/evx-prototype-compiler.html.

[Loewe, 2026a] Loewe, L. (LLoL) (2026). *PET Axioms — Discussions and Caveats*.

[Loewe, 2026b] Loewe, L. (LLoL) (2026). *PET Axioms ax1–ax14: Formal Panentheism*.

[Loewe, 2026c] Loewe, L. (LLoL) (2026). *PET Theorems th1–th4*.

[Lucanus, n.d.] Lucanus, M. A. (n.d.). *Pharsalia (De Bello Civili), Book I, line 81*.

[Luhmann, 1995] Luhmann, N. (1995). *Social Systems*. Stanford, CA: Stanford University Press.

[Mallet, 2012] Mallet, J. (2012). The struggle for existence: how the notion of carrying capacity, k, obscures the links between demography, Darwinian evolution, and speciation. *Evolutionary Ecology Research*, *14*, 627–665. URL: https://dash.harvard.edu/handle/1/30212075

[Marcia, 1966] Marcia, J. E. (1966). Development and validation of ego-identity status. *Journal of Personality and Social Psychology*, *3*(5), 551–558. URL: https://doi.org/10.1037/h0023281, doi:10.1037/h0023281

[Martin-Lof, 1984] Martin-Löf, P. (1984). *Intuitionistic Type Theory*. Naples: Bibliopolis.

[Maslow, 1943] Maslow, A. H. (1943). A theory of human motivation. *Psychological Review*, *50*(4), 370–396. URL: https://doi.org/10.1037/h0054346, doi:10.1037/h0054346

[McCollum et al., 2006] McCollum, J. M., Peterson, G. D., Cox, C. D., Simpson, M. L., & Samatova, N. F. (2006). The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Computational Biology and Chemistry*, *30*(1), 39–49. URL: https://doi.org/10.1016/j.compbiolchem.2005.10.007, doi:10.1016/j.compbiolchem.2005.10.007

[Meadows, 2008] Meadows, D. H. (2008). Wright, D. (Ed.). *Thinking in Systems: A Primer*. White River Junction, VT: Chelsea Green Publishing.

[Meyerowitz-Katz & Merone, 2020] Meyerowitz-Katz, G., & Merone, L. (2020). A systematic review and meta-analysis of published research data on COVID-19 infection fatality rates. *International Journal of Infectious Diseases*, *101*, 138–148. URL: https://doi.org/10.1016/j.ijid.2020.09.1464, doi:10.1016/j.ijid.2020.09.1464

[Moltmann, 1981] Moltmann, J. (1981). *The Trinity and the Kingdom: The Doctrine of God*. San Francisco: Harper & Row.

[Mosley et al., 2025] Mosley, T. J., Zajdel, R. A., Alderete, E., Clayton, J. A., Heidari, S., Pérez-Stable, E. J., ... Bernard, M. A. (2025). Intersectionality and diversity, equity, and inclusion in the healthcare and scientific workforces. *Lancet Regional Health — Americas*, *41*, 100973. URL: https://doi.org/10.1016/j.lana.2024.100973, doi:10.1016/j.lana.2024.100973

[Mullins, 2013] Mullins, R. T. (2013). Simply impossible: a case against divine simplicity. *Journal of Reformed Theology*, *7*(2), 181–203. URL: https://doi.org/10.1163/15697312-12341294, doi:10.1163/15697312-12341294

[Hippo, n.d.a] of Hippo, A. (n.d.). *City of God (De Civitate Dei), Books XI–XII*.

[Hippo, n.d.b] of Hippo, A. (n.d.). *Confessions, Book VII*.

[Oppy, 2006] Oppy, G. (2006). *Arguing about Gods*. Cambridge: Cambridge University Press.

[Ottati et al., 2015] Ottati, V., Price, E., Wilson, C., & Sumaktoyo, N. (2015). When self-perceptions of expertise increase closed-minded cognition: the earned dogmatism effect. *Journal of Experimental Social Psychology*, *61*, 131–138. URL: https://doi.org/10.1016/j.jesp.2015.08.003, doi:10.1016/j.jesp.2015.08.003

[Perrow, 1984] Perrow, C. (1984). *Normal Accidents: Living with High-Risk Technologies*. New York: Basic Books.

[Plantinga, 1974a] Plantinga, A. (1974). *God, Freedom, and Evil*. New York: Harper & Row.

[Plantinga, 1974b] Plantinga, A. (1974). *The Nature of Necessity*. Oxford: Oxford University Press.

[Rittel & Webber, 1973] Rittel, H. W. J., & Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy Sciences*, 4(2), 155–169. URL: https://doi.org/10.1007/BF01405730, doi:10.1007/BF01405730

[Schelling, 1960] Schelling, T. C. (1960). *The Strategy of Conflict*. Cambridge, MA: Harvard University Press.

[Senge, 1990] Senge, P. M. (1990). *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York: Doubleday/Currency.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, *27*(3 & 4), 379–423, 623–656. URL: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x, doi:10.1002/j.1538-7305.1948.tb01338.x

[Simons, 1987] Simons, P. (1987). *Parts: A Study in Ontology*. Oxford: Oxford University Press.

[Sobel, 2004] Sobel, J. H. (2004). *Logic and Theism: Arguments For and Against Beliefs in God*. Cambridge: Cambridge University Press.

[Stutt et al., 2020] Stutt, R. O. J. H., Retkute, R., Bradley, M., Gilligan, C. A., & Colvin, J. (2020). A modelling framework to assess the likely effectiveness of facemasks in combination with `lock-down' in managing the COVID-19 pandemic. *Proceedings of the Royal Society A*, *476*(2238), 20200376. URL: https://doi.org/10.1098/rspa.2020.0376, doi:10.1098/rspa.2020.0376

[Talic et al., 2021] Talic, S., Shah, S., Wild, H., Gasevic, D., Maharaj, A., Ademi, Z., … Ilic, D. (2021). Effectiveness of public health measures in reducing the incidence of COVID-19, SARS-CoV-2 transmission, and COVID-19 mortality: systematic review and meta-analysis. *BMJ*, *375*, e068302. URL: https://doi.org/10.1136/bmj-2021-068302, doi:10.1136/bmj-2021-068302

[Tay & Diener, 2011] Tay, L., & Diener, E. (2011). Needs and subjective well-being around the world. *Journal of Personality and Social Psychology*, *101*(2), 354–365. URL: https://doi.org/10.1037/a0023779, doi:10.1037/a0023779

[Tetlock, 2005] Tetlock, P. E. (2005). *Expert Political Judgment: How Good Is It? How Can We Know?* Princeton, NJ: Princeton University Press.

[Tuckman, 1965] Tuckman, B. W. (1965). Developmental sequence in small groups. *Psychological Bulletin*, *63*(6), 384–399. URL: https://doi.org/10.1037/h0022100, doi:10.1037/h0022100

[Varzi, 2016] Varzi, A. C. (2016). *Mereology*.

[Wasserman et al., 2020] Wasserman, D., van der Gaag, R., & Wise, J. (2020). The term "physical distancing" is recommended rather than "social distancing" during the COVID-19 pandemic for reducing feelings of rejection among people with mental health problems. *European Psychiatry*, *63*(1), e52. URL: https://doi.org/10.1192/j.eurpsy.2020.60, doi:10.1192/j.eurpsy.2020.60

[Whitehead, 1929] Whitehead, A. N. (1929). *Process and Reality: An Essay in Cosmology*. New York: Macmillan.

[Wilde, 2018] Wilde, R. (2018). *Joseph Stalin's Death—He Did Not Escape the Consequences of His Actions*. https://web.archive.org/web/20201204150340/https://www.thoughtco.com/how-did-stalin-die-1221206.

[Wink, 1984] Wink, W. (1984). *Naming the Powers: The Language of Power in the New Testament*. Philadelphia: Fortress Press.

[Wintour, 2020-07-16] Wintour, P. (2020-07-16). Covid-19 will devastate poorest nations if West does not act, warns UN: G20 told to "step up now or pay price later". *The Guardian*. URL: https://www.theguardian.com/world/2020/jul/16/covid-19-may-devastate-poorest-nations-if-west-does-not-act-warns-un

[Wurth et al., 2020-06-29] Wurth, R. C., Braxton, M. L., & Cohen, C. L. (2020-06-29). Racism and Covid-19 threaten our health—We can't fight them as separate battles. *The Guardian*. URL: https://www.theguardian.com/commentisfree/2020/jun/29/racism-coronavirus-covid-19-black-brown-america

[Anthropic, 2026a] Anthropic (2026 , February). *Claude Opus 4.6 System Card*. Anthropic.

[Anthropic, 2026b] Anthropic (2026 , April). *Claude Opus 4.7 System Card*. Anthropic.

[Balospecom, 2026] Balospe.com (2026). *Formal Foundation Test for the e7Day Axiom System*.

[Bhikkhu Bodhi, 2000] Bhikkhu Bodhi. (2000). *The Connected Discourses of the Buddha: A Translation of the Samyutta Nikaya*. Boston: Wisdom Publications.

[Gregory of Nyssa, n.d.] Gregory of Nyssa (n.d.). *Life of Moses (De Vita Moysis)*. Written ca. 390 CE. Translation: Abraham J. Malherbe and Everett Ferguson (Paulist Press, 1978), Classics of Western Spirituality series; preface by John Meyendorff. Alt URL: https://archive.org/details/lifeofmoses0000greg.

[John of Ephesus & Pearse, 543CE, 2017] John of Ephesus, & Pearse, R. (543CE, 2017). *John of Ephesus Describes the Justinianic Plague*. https://web.archive.org/web/20210306031539/https://www.roger-pearse.com/weblog/2017/05/10/john-of-ephesus-describes-the-justinianic-plague/.

[National Center for Health Workforce Analysis & Health Resources and Services Administration, 2014] National Center for Health Workforce Analysis, & Health Resources and Services Administration (2014). *Sex, Race, and Ethnic Diversity of U.S. Health Occupations (2010–2012)*. U.S. Department of Health and Human Services. https://web.archive.org/web/20201024060416/https://bhw.hrsa.gov/sites/default/files/bhw/nchwa/diversityushealthoccupations_2012.pdf.

[Yah Yas everyone LLoL ClaudeOp46Max Anthropic and The Spirit of Boolean Truth, 2026a] Yah, Yas, everyone, LLoL, ClaudeOp46Max, Anthropic, and The Spirit of Boolean Truth (2026). *Matheo-1: The PET Model — A Mereological Axiom System for Pan-En-Theistic Mathematical Theology*.

[Yah Yas everyone LLoL ClaudeOp46Max Anthropic and The Spirit of Boolean Truth, 2026b] Yah, Yas, everyone, LLoL, ClaudeOp46Max, Anthropic, and The Spirit of Boolean Truth (2026). *Matheo-2: The e7Day Axiom System — Towards a Formal Framework for Self-Correcting Construction*.

[Yah Yas everyone LLoL ClaudeOp46Max Anthropic and The Spirit of Boolean Truth, 2026c] Yah, Yas, everyone, LLoL, ClaudeOp46Max, Anthropic, and The Spirit of Boolean Truth (2026). *The PET Model: A Mereological Axiom System for Pan-En-Theistic Mathematical Theology*.