

Split-Domain Cognition — A Principle, Not a Pattern

Why SDC extends beyond AI tooling, and what the practice looks like when it does.

Canonical position essay. Version 1.0, April 2026.

Preface

About this document. This is the published position of Split-Domain Cognition. Its argument is short: SDC was crystallised as the name for an AI-pipeline pattern, but the principle is older than the implementation and outlasts any particular software. The essay distinguishes the principle from the pattern, sketches what the practice looks like once the distinction is made, and explains why the term has scope beyond AI tooling. It is the entry point for readers who already know what SDC does inside a Koher tool and want to know what it is when removed from that context.

Where this sits in the corpus. The canon home is splitdomaincognition.org. This essay is also rendered at [/principle/](#). The long-form articulation is in *Split-Domain Cognition* (the source document); the operational methodology is in *Process*; the protocol for deriving a domain-specific variant is at [/derivation-protocol-v1/](#); the catalogue of declared variants at [/variants/](#). The condensed in-brief reading at [/sdc-in-brief/](#) is the shortest entry into the same material.

Authority and version. Canonical, v1.0. April 2026. The website is the source of record. If this PDF and the website disagree, follow the website.

Use. Openly citable. See [Governance](#) for the one-person canon and how variants are admitted.

Author. Prayas Abhinav.

Introduction

A principle and a pattern are different objects. A pattern is a recurring shape in implementations — useful for builders, transferable across instances of the same kind. A principle is a commitment about what a practice is *for* — useful for any practitioner deciding whether a given act fits or breaks the practice. The two get conflated when the principle is first encountered in a single, vivid implementation and the implementation is mistaken for the thing itself.

This essay is about that mistake, and about what becomes visible once it is named. SDC's first vivid instance is a three-stage AI pipeline. The pipeline is real, load-bearing, and worth describing on its own terms. But it is not the principle. The principle holds wherever language work and judgement work are at risk of collapsing — in classrooms, in clinics, in courtrooms, in curatorial practice, in research, in policy. The pages that follow trace the line that separates the two.

The position

Split-Domain Cognition (SDC) was crystallised as the name for a pipeline architecture in AI tools: AI handles language in Stage 1, deterministic code handles judgement in Stage 2, AI narrates the judgement in Stage 3. That is the instance. The instance is real and load-bearing — it is how Koher builds its tools, and it is how the canonical claim ("language and judgement are two different domains") becomes operational in software.

But SDC is not only a pipeline pattern for AI tools. It is a principle about cognition, applicable wherever language work and judgement work are at risk of collapsing into the same call. The principle says: when two different kinds of thinking are being done in sequence or in parallel, build the work so the kinds stay distinct. Let the language domain be language; let the judgement domain be judgement; let each be answerable to the criteria proper to itself.

Koher has practised SDC across many domains for longer than the term has existed. Teaching, curation, public art, veganism and animal rights, writing, and the shape of the practice itself are each applications of the same principle. The AI tools are one instance.

They are the most precise instance, because software forces precision. But they are not the definition.

Why naming this matters

A term that crystallises in one medium tends to calcify there. *Split-Domain Cognition* was named during an AI-tool season, and its first definition — "a pipeline architecture for AI tools" — is true of the AI-tool instance. Left alone, that definition will quietly suggest that SDC is what Koher does when Koher is making AI tools, and that everything else Koher does is a separate practice.

That suggestion is false. SDC is the move that recurs across every medium Koher works in. Naming it as a principle rather than a pattern protects two things at once: the specificity of the AI instance (which deserves its technical definition) and the generality of the underlying move (which is older than the AI tools and will outlast any particular implementation).

The AI-tool instance, briefly

The AI pipeline is the most compact expression of SDC. It has three stages. Stage 1 is language work: a language model reads unstructured input (a concept statement, a transcript, a sketch description, a menu, a design brief) and produces structured signals — classifications, embeddings, tags, extracted features. Stage 2 is judgement work: deterministic code applies thresholds, priorities, and rules to those signals and produces a verdict — a score, a ranking, a diagnosis, a decision about what to surface. Stage 3 is language work again: a language model narrates the verdict in readable prose that a human user can absorb.

The three stages are held apart on principle. Language models are good at pattern recognition across language but unreliable at issuing verdicts that must be consistent, auditable, and reproducible. Code is good at verdicts but cannot generate fluent narration without overwhelming the user. Each stage does what it is good at. None borrows the criteria of another.

What is protected by the separation is precisely what is lost when it collapses: auditability, consistency, and trust. A pipeline that conflates the stages — asking a

language model to "just decide whether this is good" — cannot explain its decisions, cannot be checked against a rubric, and cannot be adjusted without retraining. A split-domain pipeline can be opened up, layer by layer, and examined. The architecture is legible because the domains are unlike and have been kept unlike.

The principle beyond the pipeline

The same move shows up whenever cognition is at risk of collapsing across unlike domains. Below are the applications already operative in Koher's practice. None of them was named SDC at the time. All of them are SDC.

Design studio pedagogy

A design studio is a daily exercise in cognition. Students produce work (language work: sketches, concepts, iterations, articulations). Faculty and peers respond (judgement work: rubric-bound evaluation, threshold-bound feedback, priority-bound critique). When these domains collapse — when students self-evaluate while generating, or when faculty generate alternatives while assessing — the studio becomes anxious and the learning stalls. Students cannot generate freely because they are simultaneously judging; faculty cannot judge fairly because they are simultaneously making.

The split-domain move in a studio is structural. The rubric is authored deterministically, outside the moment of critique. The language of student work is allowed to be generative and messy. Critique is the narration of judgement that has already happened in the faculty member's head against an explicit rubric. This is how *studioMeetingCompanion* was conceived: transcripts are Stage 1 language, faculty-authored rubrics are Stage 2 judgement, the returned narration is Stage 3. But the tool is an expression of the pedagogical move, not its origin. Good design studios have always worked this way. Koher's tool makes the move legible; the move predates the tool.

Curation

Curation is the most direct non-AI instance of SDC. *Growing Down* (2009–2010, rooftop concerts with Tanvi Srivastava) separated the language of performance (what musicians did on the night) from the judgement of venue (what was being curated by placing the

concerts on rooftops at all). *Of Games* at Khoj separated generative practice (what artists made during the residency) from curatorial judgement (what was framed, shown, and contextualised). *Museum of Vestigial Desire* (2012–2021) separated the generative making of fragments from the curatorial refusal to assemble them into a finished object; the refusal was the Stage 2 judgement, held deterministically against the form's own rules.

A curator who collapses these becomes a tastemaker performing authority — generating and judging in the same breath, with nothing left to check the judgement against. A curator who holds them apart can show without valorising, frame without endorsing, and build arguments the audience can re-enter. The two practices produce different publics. Koher's preference is clear.

Public art

The Murmur Engine (FICA submission, 11 April 2026) is SDC in public space. The project's central move is that a machine placed in a public site will refuse to render verdicts, instead emitting its hesitations, retrievals, and thresholds as ambient surface. Members of the public see the machine's language outputs (Stage 1), its internal verdicts made visible (Stage 2), and its public narration (Stage 3) — and they see, crucially, that the three layers are not the same thing. The work stages the architecture. The failure the project critiques — machines dressed up to issue confident verdicts while being probabilistic underneath — is exactly the failure SDC in AI tools is built to prevent. The gallery and the pipeline make the same argument.

Veganism and animal rights

The politics of animal ethics is a domain where SDC cuts. The language of *welfare* is probabilistic, managerial, contextual — it is the language of how animals are treated, measured in welfare-indicator units, balanced against other considerations. The language of *rights* is deterministic, rule-bound, absolute — it is the claim that certain treatments are categorically wrong regardless of context. The two are not two degrees of the same thing. They are two kinds of thing.

The conflation is the whole problem. "Humane meat" is welfare language smuggled into rights territory — a probabilistic adjective pretending to do deterministic work. Koher's position (and Prayas's lived practice) refuses the conflation. *animalRightsLens* builds the

refusal into software: AI describes a menu or a dish (language), code applies a rights-based rule (judgement), AI narrates the verdict (language). The tool is faithful to the ethics because it is structurally split-domain. The ethics are not a layer on the tool; the tool is an expression of the ethics.

Writing and philosophy

Museum of Vestigial Desire and the essays that followed it are SDC in prose. The work separates descriptive language (what is there, what is being traced, what remains) from evaluative judgement (what this means, what should be done). The essay form of MoVD was built so the two could not collapse: fragments held traces without rendering verdicts about them, and the reader had to do the judgement work in their own reading. This is why the project generated material for nine years and did not finish. Finishing would have required collapsing the domains, which the form refused.

The split between description and judgement is not new to philosophy. Hume named it; Moore formalised it; the descriptive–normative distinction is foundational. SDC is not a claim to have discovered this. It is a claim that the distinction is architecturally enforceable, in software and in practice, and that enforcement protects what conflation erases.

The shape of the practice itself

Koher as a whole is an SDC move at the scale of a life. The Level 1 / Level 2 position (`positions-source/level-1-living-practice.md`) names it explicitly: the lived practice (Level 1) is generative, ongoing, answerable to the practitioner's own attention; the shared practice (Level 2) is evaluative, answerable to reception and outcomes. Conflating the two produces the failure mode named in that position — a practice that borrows its sense of reality from Level 2 outputs.

The canonical claim therefore operates at three nested scales. Inside a pipeline, SDC keeps language and judgement unlike. Inside a medium (a studio, a curation, an essay, a public artwork), SDC keeps generation and evaluation unlike. Inside a life, SDC keeps the lived and the shared unlike. The same architecture recurses. Koher is that recursion, at whatever scale is being looked at.

The generalised failure mode

When SDC is absent, the failure mode is the same across domains. Language work borrows the authority of judgement work without earning it; judgement work borrows the fluency of language work without disciplining it. A language model issues "just decide" verdicts that cannot be audited. A faculty member generates alternatives inside the critique and students cannot track what is being graded. A curator endorses while showing and the audience loses the ability to re-enter the argument. A welfare regime dresses itself as an ethics and the rights question disappears. A writer smuggles evaluation into description and the reader is moved without being allowed to disagree. A practitioner measures the practice by Level 2 outputs and the Level 1 work atrophies.

The symptoms differ. The mechanism is the same — a collapse of unlike domains into a single call. SDC is the architectural refusal of that collapse, wherever the collapse threatens.

What this does not mean

It does not mean all thinking must be split-domain. Some work is legitimately unified — an improvisation, a single intuition, a moment of making that is not yet asking to be judged. SDC is a principle for when language and judgement are being asked to operate together. When only one is in play, the principle does not apply.

It does not mean code is always better than AI at judgement. The claim is not that code is superior to language models. The claim is that language models are good at language and code is good at judgement, and that each should do what it is good at. In domains without language models, the same principle becomes: let the generative faculty be generative; let the judging faculty judge; do not ask either to do the other's work.

It does not collapse the AI-tool instance into the general principle. The AI pipeline remains SDC's most precise expression, and its technical definition ("a pipeline architecture for AI tools" with three named stages) is unchanged. The generalisation is upwards, not sideways. The principle is larger than the pipeline, but the pipeline is still the pipeline.

It does not make Koher a philosophy of everything. SDC is narrow. It names one move about one kind of failure. It does not claim to describe how all human thought should

work, how all institutions should be organised, or how all domains should be governed. It names a principle that matters when cognition is at risk of collapsing across unlike domains, and it says nothing about domains where that risk is not present.

Implications for scope and decision-making

Several consequences follow.

Evaluating a new Koher project. The first question is not "does this involve AI?" but "does this involve a risk of collapse between language and judgement?" If yes, SDC applies and Koher's architectural commitment engages. If no, Koher may still be the right home for the project — but the principle's specific work is elsewhere.

Evaluating an external tool, institution, or claim. The question is the same. Where is the collapse between language work and judgement work? What is being asked to produce a verdict while being probabilistic underneath? What is being asked to describe while being evaluative underneath? Naming the collapse is half the critique.

Explaining Koher to someone outside AI. The AI pipeline is a good illustration, not a sufficient description. A fuller description includes the studio example, the curatorial example, the MoVD example, the Level 1 / Level 2 example. For many audiences these are more accessible than the pipeline, and they show that SDC is not a technical trick but a practice.

Grant and fellowship framing. Funders in non-AI domains (art, philosophy, humanities, public art, ethics) can legitimately be shown SDC as the underlying principle of their proposal, without pretending the proposal is an AI project. The Murmur Engine is an SDC public artwork. A philosophy fellowship could be an SDC writing project. An art commission could be an SDC curatorial project. The principle travels; the AI pipeline stays where it belongs.

Closing

Split-Domain Cognition is the name Koher has given to a move it has made for twenty-five years across many media. The AI pipeline is the move's most precise expression. The studio, the curation, the public artwork, the ethics, the essay, and the shape of the

practice itself are the move's earlier, ongoing, and distributed expressions. The term names them all.

The principle is the same at every scale: when two kinds of thinking meet, build the work so the kinds stay distinct. Let the language be language. Let the judgement be judgement. Let the lived be lived and the shared be shared. Keep unlike domains unlike — and trust that the domains being unlike is not a limitation but the source of whatever power the architecture has.

Origin: Prayas Abhinav's multi-domain practice (2001–), crystallised as a term on 14 April 2026 (see `term-chosen.md` for the naming decision). This position extends the term's scope from its AI-pipeline instance to its general principle. Indexed in `../core-docs.md`, `../position-statement.md`, `../CLAUDE.md`, and `../MEMORY.md`. Paired with `../positions-source/level-1-living-practice.md` — the same claim at the scale of the practice itself. Relocated from `positions-source/sdc-principle-not-pattern.md` into `split-domain-cognition/` on 16 April 2026.