

Phronesis

*An overview of a work in progress — a stance, four tools, eleven
norms and a chain that turns them into code.*

AUTHOR Kaspar Brönnimann

DATE 14 August 2026

VERSION v1

FOREWORD

Why this paper

In a training session, a tool shows a project manager a formulation — smooth, plausible, apparently verified. She adopts it. A year later it turns out that the formulation contains a citation that does not exist.

Neither the tool nor the project manager lied. Both did their best. Nevertheless the damage is real.

This paper orders a work that seeks to avoid precisely such cases — not through better prompting, but through a different fundamental stance: the responsible statement remains reserved. Tools may calculate and interpret; they must not judge. What follows is the order out of which this stance becomes software.

Six levels carry the work, one chain turns them into code, one workshop produces them all. Whoever wants to know how this order came about and what it requires will find it on the following pages.

C O N T E N T S

— Foreword — Why this paper

I Preface — What is brought together here

II The Stance — Phronesis

III The Four Tools

IV The Instantiations

V The Workshop — Cross-beam: The Workbench

VI The Eleven Norms of the Suite

VII The Chain — How Norms Become Code

VIII The Substructure — Capabilities, Skills, Code

IX The Flank — Science and Public Presence

X The Current State

XI Closing — The One Figure of Thought

Preface — What is brought together here

This paper is not a collection. It is a map.

This paper is not a collection. It is a map.

This paper orders a work that, across four tools, their concrete instantiations, eleven norms and two scientific publications, sustains the same stance. It is not an anthology, but a map. Whoever has the map can place every single artefact — and see why none of them stands alone.

What lies here has emerged in many places, on many occasions and in different languages — German, English, code, document, diagram, conversation. An outsider might get the impression that someone is working on many unconnected things. The impression deceives. There is only one matter; the multiplicity is its exposition.

This one matter is called Phronesis: practical wisdom, the ability to weigh, to judge and to take responsibility in the concrete individual case. It is a stance, not a tool. But it carries a family of tools that makes this stance applicable in everyday practice — and it carries a workshop that not only builds these tools but grounds them verifiably.

The whole has six levels and one flank alongside them:

- Level 1 — Stance. Phronesis as manifesto and guiding principle.
- Level 2 — Tools. Aletheia, Mnemosyne, Methodos, Kairon.
- Level 3 — Instantiations. Concrete products: ALETHEIA Radar, pseudonymisation layer, kennora, HERMES PIA, KAIRON Dashboard, DigiTwin.
- Level 4 — Norms. Eleven regulatory frameworks that lie above all instantiations.
- Level 5 — Chain. The agent development chain that turns norms into code.

- Level 6 — Substructure. Capabilities, skills, code — the shared technical foundation.
- Flank — Science and public presence. Two scientific publications and one quiet repository.

The Stance — Phronesis

Level 1. What cannot be software, because it must first be borne.

Phronesis is an open family of tools for deliberate decision-making in a world shaped by artificial intelligence. But the first and most important sentence is: it is not software. It is a stance served by three conditions: it must be spoken, it must be shared, and it must find its form in tools — otherwise it remains opinion.

The reference is Aristotle. In the *Nicomachean Ethics* (Book VI) he distinguishes three modes of knowing: *episteme* — knowledge of what is necessarily so —, *techne* — the ability to produce a workpiece according to a rule — and *phronesis* — practical wisdom, doing the right thing in the concrete case. Only *phronesis* knows the individual case, the context, the person who takes responsibility. Only it is inescapably bound to a human being who acts and can bear what follows from that action.

This is more than a philological footnote. It is the fundamental distinction on which the whole construction hangs. For artificial intelligence is at its best *episteme* (it knows what is the case in many instances) and *techne* (it can do what many often do). Whether an artificial system could one day also judge *phronetically* is an open question — perhaps even one that can never be conclusively answered. This paper does not settle that question. It settles a different one, independent of it: in a human institution, a human must take responsibility for what the institution does. The right and the duty to judge, in a matter with consequences for others, are bound to persons who can be held accountable — not to systems whose accountability remains unclarified. This question of attribution stands even if the ontological question were one day to be decided otherwise.

Phronesis: practical wisdom remains human.

The manifesto formulates this stance in ten convictions. They are not treated here as a catalogue, but summarised in a brief address — the way they belong together.

We think that the greatest weakness of human action is not malice, but unawareness. Where something happens without anyone noticing it, action loses its moral place. We think that transparency is more important than perfection; whoever promises perfection must hide errors, whoever promises transparency must show them. We think that AI may advise, but responsibility must remain with the human — socially, politically, morally. We think that uncertainty is a field, not a formulation: it wants to be named, not veiled by rhetoric.

We think that the quality of judgement is more decisive for the future than technical intelligence alone — because even correct recommendations turn wrong when no one is left to examine them. We think that culture and meaning-making are primordially human: they are the ground from which judgement can arise at all. And we think, finally, that the observed and the presumed must be strictly distinguished.

The observed and the presumed are two different things.

From this stance follow the four tools. They are its open exposition — open, because they are not closed software, but thinking roles that take on different shapes in different applications. And they follow from the stance, not the other way round: only once it is settled what a tool should serve can it meaningfully be asked how it is to be built. A tool that does not know its stance leads its users astray.

That is why this chapter comes first. Everything described in the following chapters — the four tools, the concrete instantiations, the workbench, the eleven norms, the chain, the substructure — is a consequence of what is said here. Nothing of it stands for itself; everything belongs together; and the hyphen between all the parts is the one stance that cannot be delegated.

The Four Tools

Level 2. The open exposition of the stance — not software, but thinking roles.

The four tools are the open exposition of the stance. They are not themselves software, but thinking roles: Aletheia (perceiving), Mnemosyne (remembering), Methodos (guiding methodically), Kairon (deciding). Every concrete application — radar, pseudonymisation layer, knowledge graph, project assistant, decision dashboard — is an instantiation of one of these four, often with traits of several. The four tools give it the way of thinking; the instantiation gives it its shape.

TOOL	FUNDAMENTAL OPERATION	TIPPING POINT
<i>Aletheia · Gold</i>	Perceiving change	where making visible turns into judging
<i>Mnemosyne · Blue</i>	Memory; separating statement from person	where remembering turns into surrendering
<i>Methodos · Green</i>	From conversation to structured outcome	where form comes to rule over the human
<i>Kairon · Terracotta</i>	Decision: Kairos and Aion together	where an outcome stands without reasons

Aletheia · Gold

Aletheia — perceiving change

Aletheia is the philosophy of Phronesis for perception. The Greek word means the unconcealing, the coming forth out of hiddenness. Aletheia makes developments visible,

classifies them and proposes an assessment — but submits the judgement to the human for confirmation.

Making visible is not yet judging. To bring something to light is not to have already evaluated it.

Six convictions carry Aletheia: first, the strict separation of making visible and judging; second, the duty of source (what remains without a source is rumour); third, the distinction by degree of certainty (confirmed event, early signal, mere announcement); fourth, attention as selection, not completeness; fifth, testing the new against the past; sixth, the orientation of unconcealing towards an inside — towards memory, form and decision. Perception that mistakes itself for a judgement is nothing but a feed.

Mnemosyne · Blue

Mnemosyne — memory that protects

Mnemosyne — in Greek remembrance the mother of the Muses — stands for a memory that preserves without surrendering. Her fundamental operation is the separation of statement and person: the substantive content remains usable, the identity remains protected.

A statement and the person it concerns are two different things.

Four convictions: knowledge can be shared without exposing people. A protective layer must itself be visible — no one trusts a black box that promises trust. In case of doubt, protection takes priority: an overlooked name weighs more heavily than one replacement too many. And finally: remembering and protecting belong together; a memory that does not protect is not a memory, but a betrayal.

Methodos · Green

Methodos — from conversation to structured outcome

Methodos leads from conversation to outcome, without forgetting the way. It asks, listens, orders — and lets a structured version emerge in which what was said remains recognisable.

Knowledge arises in conversation, not in the empty form.

A method is the crystallised judgement of many.

Structure is a means, not an end.

Four convictions: knowledge arises in conversation, not in empty fields. A proven method is not a bureaucratic corset, but the collected practical knowledge of those who had to solve the same task before us. Thinking along means noticing what is missing — completeness is not achieved through mandatory fields, but through good questions. And form must serve the human, not the other way round.

Kairon · Terracotta

Kairon — decision that remains observed

Kairon carries two Greek words: Kairos, the right moment, and Aion, the duration. Both belong to a decision: it falls in a moment, but it acts through time. Without Kairos there is no decision; without Aion there is no learning.

Decisions need not only outcomes, but reasons.

Every good decision must remain observable.

Dissent can be a sign of maturity.

Three convictions: decisions need reasons, not just outcomes — a yes or no without a context of options, assumptions, risks, responsibilities and expected effects is not a decision but a note. Good decisions must remain observable, because only in reality does it become clear whether assumptions held. And dissent is not a disturbance, but often a sign that a matter has been understood in its depth.

The Instantiations

Level 3. Thinking roles become products that stand in the world.

At this level the four tools become concrete products. Every instantiation is an instantiation of a tool — often with traits of several, since no real product can be cleanly reduced to a single thinking role. A project interview is predominantly Methodos, but it makes decisions; a knowledge graph is predominantly Methodos, but it remembers; a pseudonymisation layer is predominantly Mnemosyne, but it also perceives what is to be protected.

INSTANTIATION	TOOL	MATURITY	BRIEF CHARACTERISATION
<i>ALETHEIA Radar</i>	Aletheia	under construction	AI technology radar for public authorities, multi-tenant
<i>Pseudonymisation layer v0.0.1</i>	Mnemosyne	first version, operational	local on 127.0.0.1, between application and provider, 243,402 surnames + 66,916 first names (BFS)
<i>kennora</i>	Methodos	under construction	listener for knowledge from conversations, knowledge graph as tree and network
<i>HERMES PLA v0.35.0</i>	Methodos + Kairon	in use	HERMES-2022 initialisation order in interview form, formal and substantive review separated
<i>KAIRON Dashboard (Feature-V2, Commit</i>	Kairon	feature-mature, in testing	ten-state lifecycle, 75 regression tests, 4,507 lines of application code

369db5a)

<i>DigiTwin (Process Simulator)</i>	related · not part of family	Hackathon eJustice.ch prize	related figure of thought, not core of the tool family
-------------------------------------	------------------------------	-----------------------------	--

ALETHEIA Radar — the verified outward gaze

The ALETHEIA Radar is the consistent instantiation of Aletheia for public authorities. It gathers and orders observations on AI developments, distinguishes between confirmed events, early signals and mere announcements, and attaches its source to every entry. It is multi-tenant: each authority sees its own radar, with its own selection of attention, its own history, its own review stamps. It does not judge, but proposes a classification; the judgement remains with the human. The maturity is under construction — the structure stands, the practice of observation is growing.

Pseudonymisation layer v0.0.1 — the quiet upstream step

The pseudonymisation layer is a small service placed between application and language model provider. It replaces names with stable pseudonyms before content leaves the machine — and reverses the substitution again after the answer. The provider sees the substantive statement; the persons remain with themselves. The name lists come from the Swiss Federal Statistical Office (BFS): 243,402 surnames and 66,916 first names. The layer runs locally on 127.0.0.1 and is thus verifiable for whoever uses it: it is not a black box, but a visible service. Maturity: first version, operational.

kennora — the listener for knowledge

kennora listens to a conversation and lays out a knowledge graph. The graph has two views: a tree, showing hierarchies and affiliations, and a network, taking up connections of every kind. Besides the usual edges (is a, belongs to, follows from) there is the special connection rhymes-with. It marks semantic neighbourhoods that arise from analogies and images — not from formal similarity, but from the heard sound of a connection. Maturity: under construction.

HERMES PIA v0.35.0 — the interview with review

HERMES PIA accompanies the initialisation phase of a HERMES-2022 project. The user narrates in everyday language what he intends; the software shapes an initialisation order from it. Two reviews then run separately: a formal review in code, without a language model — it counts whether all mandatory entries are there, whether placeholders have been left standing, whether chapters contradict one another. And a substantive review with a language model — it reads the order from the perspective of the client and proposes a release recommendation together with conditions. The user signs in with his professional e-mail address; each organisational unit sees only its own projects. Numbers: 104 tests, 2,054 lines of substantive code, 42 public functions. Maturity: in use.

KAIRON Dashboard — the ten-state lifecycle

The KAIRON Dashboard is the most consistent instantiation of the Kairos-Aion axis. Every decision runs through a ten-state lifecycle that lets the decision not end with release, but pass into observation: Draft → In Review → Simulated → Risk Reviewed → Governance Reviewed → Approved → Observed → Reassessment Needed → Reassessing → Archived. The state Observed is the important one: here the decision enters time and is tested for whether its assumptions hold. Reassessment Needed and Reassessing close the circle. Numbers: 75 regression tests, 4,507 lines of application code. Technically: Flask, SQLAlchemy, Jinja, Alembic migrations. Branch Feature-V2, Commit 369db5a. Maturity: feature-mature, in testing.

DigiTwin — the related neighbour

DigiTwin is a process simulator that won a prize at the Hackathon eJustice.ch. It does not belong to the core family of Phronesis tools, because it is not a tool in the sense of the four thinking roles. But it is related thinking — above all with respect to the Kairos-Aion axis: simulation tests assumptions before a decision is taken, and thus flanks the Kairon mode from the other side. Where Kairon observes effect in retrospect, DigiTwin tests it in advance.

The Workshop — the Workbench

Cross-beam. Not a tool of the family, but that which produces tools.

The workbench has a special role in the whole. It is itself not a tool of the Phronesis family — it is neither Aletheia nor Mnemosyne, neither Methodos nor Kairon. It is instead the workshop in which the tools come into being. And it does not merely produce them, but grounds them verifiably. That is its actual achievement: it delivers not code alone, but the dossier that shows why one may trust this code.

Four stages, seven products

A tool comes into being in four stages: requirements picture, architecture, specification, implementation. Each stage has its producers, its reviewers and its gates. At the end, seven products lie ready: requirements picture, architecture, specification, test cases, technical specification, commissioning, and — as the actual heart of the matter — the review and decision protocol.

The review and decision protocol is not an ornament. It is the product that holds the others together. It shows who reviewed which statement, which models were involved, where they agreed, where they contradicted one another, which findings remained open and what has not been remedied despite release. It is the document in which the workbench does justice to its own norm.

Producers and reviewers, with different models

At every stage a producer-reviewer pair works. Both are different instances; more importantly, they are different models. In the example VOR-0580b0fa — the quantum mechanics tutor on which the workbench in its version 1.0 is documented — the producer was xai/grok-4.3 and the reviewer openai/gpt-4.1. Two different training philosophies, two different RLHF regimes, two different hallucination profiles. Different models

guarantee no independent errors; but they at least avoid the trivial self-review of the same model instance and make model dependence visible.

The core calculates, the model interprets, the human judges and takes responsibility

The core calculates, the model interprets, the human judges and takes responsibility.

That is the architectural fundamental decision of the workbench, and it is the craft twin of RA-01. Where determinism is required — pytest return values, start-up probe, whether the service responds, whether test cases pass, whether identifiers are complete — no language model may decide. Where interpretation is required — whether a chapter substantively matches the requirements, whether architecture fulfils its purpose — no rule-based code may judge. The core counts and measures; the model reads and evaluates; and where a responsible judgement is to be made from that, the human steps in. All three remain in their role.

Convergence is not correctness

Convergence is not correctness.

The second load-bearing sentence of the workbench. When a reviewer falls silent, it can mean two things: the producer got it right — or the reviewer gave up. The workbench does not confuse the two. That is why it records when a review ends by agreement, and when it ends by exhaustion. The second possibility is the more frequent and the more dangerous; it is not blanked out.

A release version is never finding-free. What stands here has been decided — not remedied.

And that is why a workbench release can never claim to be free of errors. It only claims that the findings still present are decided: seen, named, located, provided with a handling. What could be done has been remedied; what remains has been known.

Core mode and human mode

The workbench knows two decision modes. Core mode holds precisely when reviewer and producer agree and the finding reads: "applies." All other cases — every contradiction, every "does not apply," every "condition" — fall into human mode. The machine may only confirm. All negating, restricting and conditioning remains reserved for the human.

An agent is only as good as its acceptance criterion.

That is the most consistent implementation of the condition that in the research paper is called consequentiality: the human position determines the outcome, unless a documented act replaces it. In the workbench this condition is not philosophical but operational: it is the difference between a green and a red line in the protocol.

What follows

The workbench is thus the operational answer to the fundamental question of the stance: how can practical wisdom remain with a human when machines write along, review along, propose along? The answer is: through the machine knowing its role, through two different machines reviewing each other, through the core counting instead of judging, through every negation returning to a human.

The Eleven Norms of the Suite

Level 4. The common regulatory framework above all instantiations.

The eleven norms form the common regulatory framework of the suite. They give the workbench its measure, the tools their form and the instantiations their verifiability. They are the reason why two tools that differ in substance — HERMES PIA and KAIRON — nonetheless remain recognisable as parts of the same family: their identifiers, their states, their review regimes, their language follow the same norms.

NO.	NORM	PURPOSE
1	Manifestos	stance, purpose, limits
2	Appearance	brand & logo styleguide, logo family
3	Operation	UX guideline, user guidance
4	Reference architecture v0.4	13 principles RA-01 to RA-13, 10 prohibitions NR-01 to NR-10
5	Capability catalogue	AI call, persistence, workflow, STT, TTS, renderer, export
6	Contract form	skill-manifest.yaml
7	Skill norm	SK-01 to SK-17 (Skill norm v0.2; the previous version v0.1 ended at SK-15), with rule + reason + review
8	Test concept	deterministic and model-supported reviews

9	Development guidelines	CLAUDE.md per repository
10	Role model	HERMES specification: business role ↔ method ↔ reviewer ↔ governance
11	Substantive quality	quality model with D and F criteria

The thirteen principles of the reference architecture v0.4

The reference architecture is the core of the norm suite. Its thirteen principles describe how a tool of the family must be built if it is to carry the stance. They are not recommendations, but commitments: whoever builds according to them can have it verified that he has built according to them.

RA-01 The core calculates, the model interprets, the human judges and takes responsibility.

RA-02 No authoritative facts from the model.

RA-03 Every core object carries its governance mixin.

RA-04 Draft and confirmation are separate fields.

RA-05 Release is a gate; a change invalidates it.

RA-06 Changes are transitions, not breaks.

RA-07 Configuration before programming.

RA-08 Tenant separation is structure, not filter.

RA-09 No direct model contact.

RA-10 Every result carries its provenance.

RA-11 Uncertainty is a field, not a formulation.

RA-12 Controlled vocabulary.

RA-13 A tool produces evidence, not only results.

Note: The version of RA-01 given here is a refinement over the reference architecture v0.4 (there: "The core decides, the model proposes, the human takes responsibility."). The new version follows the tripartite formula of Chapter V and will be brought over with the next version of the reference architecture. Until then, RA-01 in this version is a proposal for adjustment, not the current norm.

Not all thirteen principles are anchored in the same way. Ten rest directly on a manifesto sentence. Two — RA-07 "Configuration before programming" and RA-12 "Controlled vocabulary" — are technical rules without direct value anchoring, but legitimised by proven product practice. One — RA-08 "Tenant separation is structure, not filter" — is indirectly anchored; its load-bearing manifesto sentence PH-11 on the handling of what has been entrusted is deferred until the operationalisation of "expressly intended for sharing" has been clarified. This differentiation is not concealed but declared: only what is grounded holds; only what is declared can be examined.

The ten prohibitions

NR-01 A tool must not: pass a judgement or take on a responsibility that belongs to a human.

NR-02 A tool must not produce a citation, a number or an item of evidence that does not come from a source.

NR-03 A tool must not address a model directly.

NR-04 A tool must not cross a tenant separation.

NR-05 A tool must not retain a released state while the content changes.

NR-06 A tool must not extend a controlled vocabulary of its own accord.

NR-07 A tool must not emit a result without provenance.

NR-08 A tool must not secure a guarantee solely through instruction to a model.

NR-09 A tool must not change the meaning of stored values without recording the transition.

NR-10 A tool must not subsequently edit an issued piece of evidence.

Workbench prohibitions (WB-01 to WB-10)

These prohibitions are derived from workbench practice and supplement the reference architecture; they carry their own prefix in order to preclude confusion with the NR identifiers of the reference architecture.

WB-01 No silent narrowing.

WB-02 No confirmation routines without real choice.

WB-03 No model selection without protocol.

WB-04 No confusion of convergence and correctness.

WB-05 No language-model judgements in deterministic counts.

WB-06 No mixing of producer and reviewer in the same model instance.

WB-07 No release without openly declared residual findings.

WB-08 No statement without provenance identifier in workbench outputs.

WB-09 No renaming of decided findings into remedied findings.

WB-10 No delivery of identities without verifiability of pseudonymisation.

Why these eleven norms together

The eleven norms do not by chance cover different levels. From stance (manifestos) through appearance and operation, through architecture with its principles and prohibitions, capabilities, contract form and skill norm, all the way to test, development guidelines, role model and quality model — they trace the path that a tool takes before it goes out into the world.

The Chain — how norms become code

Level 5. The agent development chain v0.7 — four stages, five gates.

The chain is the operative version of the workbench. Where the workbench is the workshop in which a thing is actually manufactured, the chain is the procedure that makes this manufacture describable. It has four stages and at every transition a gate that only a human may open: requirements picture → architecture → specification → implementation → evidence.

The four stages and their gates

At the beginning stands the requirements picture. It emerges in conversation — not in an empty form, but in a guided narration from which the requirements are drawn out. At the end of this stage stands the first human gate: the client reads what is to be done, sees the still open points and closes them or confirms them as deliberately open. Only then is the requirements picture released.

Then follows the architecture. It answers the question of how the tool must be built in order to fulfil the requirements picture. Here a producer-reviewer pair works on different models. At the end stands a conformity statement against the reference architecture: which of the thirteen principles are fully met, which partially, which remain open. Only when a human has examined this picture and decided on the open findings is the architecture released.

Then follows the specification. It transfers the architecture into a description so precise that code can be made from it — core objects, interfaces, states, transitions, test cases. Here too a producer-reviewer pair, here too a gate that only a human opens.

Finally the implementation. Here the code arises, and with it arise the products that make the code verifiable: test cases, technical specification, commissioning. At the end stands the evidence: the review and decision protocol that makes the entire chain readable backwards.

Conformity statement using HERMES PIA as an example

In the example of HERMES PIA — reviewed against the reference architecture in its version v0.2 — two of the principles are fully met, nine partially, two open. The most critical finding was RA-09 (no direct model contact): via the browser-native Web Speech API there was a path along which audio data could reach a provider without an upstream pseudonymisation. It was named, decided — and closed in the subsequent version. It is precisely the case for which the chain is made: not so that nothing goes wrong, but so that what could go wrong is named and remedied before release is declared.

Why the chain looks the way it does

The chain protects at three points at once. It protects the stance, because at every gate a human decides — not a release button, not a confirmation routine. It protects the tools, because their products remain verifiable against the norms. And it protects the users, because they do not take on the tool in a non-transparent state: they receive it with its own history.

The Substructure — Capabilities, Skills, Code

Level 6. The technical foundation that all tools share.

At this level lies the substructure: the technical building blocks that all tools of the family share. They are the reason why a new tool does not start from zero — and the reason why an error in a shared building block can be remedied at one place and all tools nonetheless recover from it.

Capabilities — the shared building blocks

Capabilities are building blocks used by several instantiations — and therefore explained, built and reviewed exactly once. They are not reinvented in every tool. The family currently comprises eight capabilities:

- AI call — with upstream pseudonymisation; no tool calls a provider directly and unprotected (RA-09).
- Persistence — the regulated storage of core objects together with governance mixin (RA-03).
- Workflow — the control of transitions between states; the basic machinery behind the ten-state lifecycles.
- Speech recognition — speech to text; guided through the pseudonymisation layer, where personal names may fall.
- Speech synthesis — text to speech; for output to the user.
- Formula renderer — mathematical notation, e.g. for teaching material such as the quantum mechanics tutor.
- Diagram renderer — diagrams and schemas; deterministically rendered, not model-generated.

- Document export — output in Word and PDF form; native export, so that the results remain editable.

Skills — the contract form

Skills are the smallest functional units. They follow a contract form so that a skill not only functions but describes its function: a frontmatter file SKILL.md with the fields name, description, scope, applies_to — and a skill-manifest.yaml as the actual contract form. Both together make a skill verifiable, exchangeable and reusable.

Code — the technical decisions

The code of the tools follows deliberately unpretentious decisions. In KAIRON: Flask as web framework, SQLAlchemy as ORM, Jinja as template engine, Alembic for migrations. That is the classical combination of the Python ecosystem — not the newest, but the most proven. It is chosen because its behaviours are known: a team that knows it knows where things go wrong where.

For reviews, pytest serves as the deterministic review instance. Results are green or red; they are not debatable. That is the core mode that the workbench demands: the core calculates.

For skill integration, MCP, the Model Context Protocol, serves as the interface. It decouples the tools from individual model providers and thereby makes the model change a matter of configuration, not of rupture. That is RA-07 (configuration before programming) in its technical form.

Taken together, the substructure yields nothing spectacular — and that is intentional. It is built so as to step back behind the tools. Whoever carries the stance and knows the norms need not be surprised at this level any more.

The Flank — Science and Public Presence

What the work measures itself against outwardly and where it becomes visible.

Alongside the six levels lies the flank: that which examines the work outwardly and where it shows itself. It consists of two scientific publications and one quiet repository.

Research paper — Where Judgement Is Reserved

The research paper with the working title Where Judgement Is Reserved is the scientific core of the flank. It follows the methodology of Design Science Research according to Peffers and introduces three terms that make the stance scientifically verifiable: structural reservation — the structural reservation of occasions for judgement as a system property, not an interaction feature; judgement occasion rate (JOR) — a time series that measures what share of the occasions requiring judgement actually returns to a human; and silent narrowing test — a review criterion that asks whether the set of reserved cases can be quietly narrowed without this narrowing itself being a documented, reasoned decision.

The paper positions the stance against Green (2022), rests empirically on Bainbridge (1983), Shen & Tamkin (2026), Parasuraman & Manzey (2010) and Buçinca et al. (2021), and finds in FAA-SAFO-17007 (2017) the historical precedent of a supervisory-order-imposed structural reservation.

Position paper — the twelve pages for everyone

The position paper is the unifying reading of the five manifestos — Phronesis, Aletheia, Mnemosyne, Methodos, Kairon — in a single twelve-page text. It is directed at interlocutors who know none of the originating texts. Where this overview paper is a map, the position paper is a panorama: the same ground, different perspective.

kaspair.ch — the quiet repository

kaspair.ch is the author's presence. It is deliberately quiet: no marketing, no newsletters, no analytics in the foreground. It is a place where the texts lie — for all who wish to read them, without a relationship being initiated. Whoever finds it finds only the work.

The flank is thus neither advertising nor sales. It is the place where the work in its present shape is to be examined — through peer review, through readership, through readers who read it into their own everyday lives.

The Current State

What is finished and what is open. An honest balance.

The state on 14 August 2026 can be read in two blocks: what is finished and what is open.

The honesty of this balance belongs to the stance itself — what I conceal, I cannot improve.

What is finished

- Reference architecture v0.4. The thirteen principles and the ten prohibitions lie consolidated.
- Position paper. Twelve pages, ready for publication; brings together the five manifestos.
- Research paper v0.6. Working version with complete narrative arc.
- HERMES PIA v0.35.0. In use; 104 tests, 2,054 lines of substantive code, 42 public functions.
- KAIRON Feature-V2. With 75 regression tests and 4,507 lines of application code; ten-state lifecycle closed.
- Workbench version 1.0. Four stages, seven products, producer-reviewer pairs on different models — documented on the example VOR-0580b0fa.

What is open

- Q4 of the research agenda. The question of whether reserved occasions for judgement preserve capacity or merely generate work is declared open in the paper. It is the actual empirical axis of a follow-up project.
- Two limits from the Contribution Boundary v0.5. First, the selection risk: organisations may prefer the less effective designs because they feel better — Buçınca shows the negative correlation between effectiveness and preference. Second, the

intervention-generated inequalities: the reserved occasions for judgement do not help all groups equally. Both limits are to be named in the paper, not hidden.

- Extension of HERMES PIA from single-tenant to multi-tenant. Currently the structure is prepared but not technically completed (RA-08 as "structure, not filter" demands more than a single line).
- Extension of the tool family. HERMES PIA covers only parts of initialisation. The phases concept, realisation, introduction, closure and implementation stand as open candidates for further instantiations.

The two blocks show where the work stands: with a foundation that carries, and with questions larger than the next release. None of this shall be tidied away quickly. What is open remains open — and will be answered at precisely the place where it arises.

CHAPTER XI

Closing — the one figure of thought

Back to the beginning: the open circle within the ring.

Four levels of resolution carry this work, and they all say the same thing.

PHILOSOPHICALLY

Practical wisdom remains human.

ARCHITECTURALLY

*The core calculates, the model interprets, the human judges and takes
responsibility.*

OPERATIONALLY

*Everything that means negation, restriction or condition falls into human
mode.*

METROLOGICALLY

Structural reservation, judgement occasion rate, silent narrowing test.

These are not four ideas. They are four levels of the same.

At the beginning stood the question about the one figure of thought that holds everything together. At the end stands the answer, in the shape of a simple sign: the open circle within the ring. The whole remains closed — the six levels interlock, the workbench connects them, the flank examines them —; and in the middle a circle remains open: the human who judges and takes responsibility.

Everything described in this paper is an exposition of this one sign. The stance is the ring. The four tools are its four expositions as thinking roles. The instantiations are their forms

in the world. The workbench is the hand that makes them verifiable. The eleven norms are the regulatory framework that lets the ring find its form. The chain is the way. The substructure is the earth on which it runs. And the flank is the place at which it must show that the sign is worth something.

What may be missing and what may not, what should be added and what cleared away, can be read off this figure of thought. The open circle in the middle is the safeguard. It must not close — neither through convenience nor through efficiency nor through the temptation to make a tool so good that it no longer needs judgement. If this circle closes, the entire sign is altered.

Five signs carry the same basic figure: the ring as the whole, the open circle as the human. The Phronesis sign shows them in their pure form. The four tool signs each add precisely one gesture from which the meaning of the tool is derived.

Phronesis

THE STANCE

The whole remains closed, the human remains open — the basic figure from which the four tools are derived.

Aletheia

PERCEPTION

The core rises like a sun above the horizon — what was hidden steps forth.

Mnemosyne

MEMORY

The core rests in preserving layers — knowledge remains without exposing the person.

Methodos

METHOD

The path rises in steps to the core — from what is said becomes what the method requires.

Kairon

DECISION

The pointer finds the signal at the edge — the right moment within the whole of time.

“Practical wisdom remains human.”