ECR-VP
(Epistemic Coherence Review Verification Protocol)

Poznań 2026

Maksim Barziankou (MxBv)

1 Brief annotation to the protocol

1.1 ECR-VP (Epistemic Coherence Review and Verification Protocol)

is not a tool for evaluating the quality of ideas, not a system for ranking interpretations, and not a method of validation through metrics. The protocol is intended to verify the structural coherence, logical integrity, and implementability of complex architectural corpora, under conditions where direct causal measurement, optimization, or aggregation of interpretations leads to distortions, semantic drift, and false confidence. The protocol deliberately excludes numerical evaluations, scoring, learning from results, and any forms of feedback between interpreters. Instead, it uses isolated interpretative contours (LLMs in clean sessions) as non-causal observers, whose task is to identify invariants, divergences, gaps, and boundaries of applicability of the corpus, rather than to reach consensus or a "best answer." ECR-VP follows the principles of regime separation: hypotheses, implementability, risks, misunderstanding, and final judgment are fixed in separate logical regimes, which are not mixed and not optimized relative to one another. Final integration is carried out by the human author of the corpus and is treated not as averaging, but as an operation of identity stabilization — an explicit alignment of interpretations with the author's intent, including acknowledgment of misunderstood or ambiguous zones. It is important to understand that the protocol does not claim objective truth, universal verification, or automatic confirmation of correctness of ideas. Its purpose is to ensure readability, verifiability, and reproducible coherence of the corpus without the participation of the author, as well as to identify architectural weak points before moving to grants, pilots, or engineering implementation. In this sense, ECR-VP is not an external audit, but an operational projection of the principles of non-causal observation and regime stabilization, applied to the verification process of complex systems itself.

1.2 II. Clarifications and adopted principles

1.2.1 On the non-use of quantitative metrics

The protocol deliberately does not introduce numerical indicators, ratings, or aggregated measures of agreement. Any quantitative metric, even an auxiliary one, inevitably becomes an object of optimization and distorts the interpretative behavior of models. ECR-VP works exclusively with qualitative differences, coincidences, and ruptures, fixed in textual form.

1.2.2 On diversification of interpreters

To reduce the risk of false coherence, it is recommended to use multiple interpreters with different architectural nature (for example, different model families, local and cloud implementations). The goal of diversification is not to increase accuracy, but to identify structurally different blind spots. At the same time, the protocol does not assume ensembling or selection of a "best" model.

1.2.3 On corpus scaling

When working with large corpora, division into thematic blocks is allowed, provided there is a fixed canonical anchor (canon extract) containing invariants, boundaries, and prohibitions of the architecture. Such an anchor is not an interpretation or a summary, but serves as a declarative point of context alignment.

1.2.4 On prompt rigidity and response format

The requirement of coherent, continuous text without lists and imitation of the author's style is aimed at extracting reasoning rather than template patterns. Minimal structuring is allowed (for example, regime headings), but transforming reports into checklists or presentation forms is prohibited.

1.2.5 On human synthesis

Final synthesis is performed by a human and is a mandatory element of the protocol. It is not considered a source of objectivity, but a responsible act of fixing divergences, including acknowledgment of what was understood, not understood, or interpreted differently. The protocol does not require consensus between interpreters and does not assume its achievement.

1.2.6 On security and IP

For external reviews, the use of edited or limited versions of the corpus is allowed. The full version may be used only in local or trusted environments. The protocol does not assume transfer of data into systems that use user dialogues for further training.

1.2.7 On the status of the protocol

ECR-VP is not a method of proving truth, scientific validation, or certification. Its task is to identify structural soundness, boundaries of applicability, and interpretation risks before moving to causal experiments, engineering implementation, or institutional evaluation.

2 PETRONUS External Coherence and Realizability Verification Protocol (ECR-VP) Version 1.0. Reference script for independent verification of a document corpus

2.1 Purpose and scope of application

This protocol defines a formal procedure for independent verification of a document corpus describing a complex long-horizon adaptive systems architecture. The purpose of the verification is not to confirm the truth of theoretical claims and not to prove correctness of implementation, but to assess the structural integrity of the corpus, the readability of the architecture without the author's participation, the realizability of the declared engineering core, and the clarity of boundaries between what is specified and what is deliberately left undefined or withheld as intellectual property. The protocol applies to any corpus that includes architectural descriptions, prior-art publications, provisional patent specifications, canonical documents (Canon), diagrams, source code, simulations, appendices, and explanatory materials. The protocol is intended for use in the preparation of whitepapers, grant applications, industrial pilots, academic formalization efforts, as well as for internal self-verification and prevention of semantic and structural drift of the corpus over time. The protocol allows the participation of multiple independent language models (hereafter referred to as "interpreters"), but does not rely on agreement between models as a criterion of truth. Models are treated as independent observers forming interpretation maps. The final verdict belongs to the human owner of the corpus or to an independent editor or expert.

Relation to academic peer review and institutional evaluation ECR-VP is not a substitute for scientific peer review, experimental validation, or formal certification. Instead, it is intended to operate upstream of such processes, as a preparatory verification layer that stabilizes architectural meaning,

boundaries, and realizability before submission to academic, industrial, or institutional evaluation. The protocol may precede peer review by identifying structural ambiguities, over-claims, or missing interface definitions that would otherwise surface only during reviewer critique. It may also accompany early-stage peer review as an auxiliary coherence check when formal validation is premature or infeasible. In contexts where conventional peer review is structurally misaligned with long-horizon or non-causal architectures, ECR-VP may temporarily replace it as an internal readiness gate, without claiming equivalence or authority. Canon protection within ECR-VP is exercised exclusively through explicit canonical constraints and their human-mediated enforcement during the Author's Response stage. The protocol intentionally avoids automated or interpreter-level enforcement mechanisms to prevent substitution of architectural intent with procedural authority. ECR-VP is not a replacement for peer review, evaluation, or certification. Its role is strictly complementary: to provide a structured assessment of semantic stability, architectural coherence, and boundary clarity prior to or alongside formal review processes. The protocol does not evaluate correctness or merit, but identifies admissible directions of further work, zones of structural ambiguity, and risks of misinterpretation. In this sense, ECR-VP serves as an orientation and stabilization instrument, helping authors, reviewers, and the broader audience to perceive the abstract architectural core without conflating it with validation or endorsement.

2.2 Core definitions

A corpus is a fixed set of files and materials subject to verification within a single protocol session. The corpus has an identity defined by the list of files, their hashes, and the version of the canonical document that specifies the architecture's structure and boundaries. An interpreter is an isolated instance of an AI model that receives the corpus in a new clean dialogue and produces a report according to the protocol script. Interpreters must be heterogeneous in architecture and epistemic tendencies in order to minimize coincident blind spots. Protocol modes are stages of analysis separated by semantic role. The protocol mandates strict mode separation to prevent mixing hypotheses with invariants, and realizability with declarations. The modes follow the logic of long-horizon architectures: first, a field of meanings and hypotheses is formed; then invariants and constraints are fixed; then an engineering realizability assessment is produced; and only after that is a final verdict allowed. Non-causal observation refers to a procedure in which the interpreter receives no feedback from the author, does not see the outputs of other interpreters, and receives no additional clarifications that could "tune" conclusions toward expectations. Non-causality here is understood in an engineering sense: as a prohibition of learning feedback during verification, not as a philosophical statement about causality. A coherence map is the final structure that displays areas of agreement between interpreters, areas of divergence, and areas of non-understanding. The coherence map serves as an indicator of corpus readability and architectural fixation completeness.

2.2.1 Open versus closed architectural status

For the purposes of this protocol, an architecture may be designated as open or closed. A closed architecture is one in which core invariants, admissibility constraints, and structural boundaries are fixed by the author and are not intended to be extended, reinterpreted, or modified by third parties without explicit authorization. An open architecture is one in which extension points, modification rules, or admissible reinterpretations are explicitly defined as part of the canonical corpus. The architectural status (open or closed) must be explicitly declared in the Corpus Passport prior to verification. Interpreters must treat closed architectures as non-extensible and must not infer missing definitions as intentional openness. Conversely, in open architectures, interpreters must distinguish between declared extension mechanisms and underspecified or ambiguous constraints. In open architectures, only explicitly declared extension points are considered admissible. Underspecification alone does not constitute openness. Failure to declare architectural status is considered a boundary ambiguity and must be recorded in the coherence map.

2.2.2 Canonical Epistemic Layers

For the purposes of this protocol, the following epistemic layers are fixed and exhaustive. Interpreters must use this list verbatim and may not introduce additional layers. Ontological layer (entities, states, agents, relations, admissible objects). Formal specification layer (invariants, prohibitions, constraints, contracts, fixed interfaces). Mathematical / algorithmic layer (formal definitions, equations, proofs, algorithmic procedures). Physical / environmental layer (assumptions about causality, embodiment, sensors, delays, noise, environment interaction). Empirical / experimental layer (tests, scenarios, metrics, validation procedures, reproducibility claims). Engineering integration layer (middleware position, interfaces, inputs/outputs, integration with existing systems). Meta-theoretical / canonical layer (canon structure, versioning rules, drift control, regime separation rules). Philosophical / metaphysical layer (meaning, values, ontology of agency, non-operational framing). IP-withheld layer (explicitly undisclosed criteria, operators, thresholds, heuristics, or internal signals). Presence, absence, or partial presence of any layer is permitted. Lack of explicit marking is not.

2.3 Principles of the protocol

The protocol is based on five principles. The first principle is that the identity of the corpus is fixed before analysis begins. This prevents substitution of the object of verification, drift, and implicit edits during verification. The second principle is that interpreters are isolated from each other and from the author during analysis. This prevents convergence of answers through hidden coordination and excludes adaptation to an expected outcome. The third principle is strict mode separation of outputs. The interpreter must separate interpretive hypotheses, corpus-fixed invariants, practical

realizability, and risk assessment. Any statement not assigned to a specific mode is considered invalid. The fourth principle is that evaluation is performed in terms of architectural integrity, boundaries, and engineering realizability, not in terms of "truth". The protocol does not replace mathematical verification or experimental validation; it structures what must be verified next. The fifth principle is that final human synthesis is mandatory. The protocol does not allow final "AI authority". The final fixation of what was understood correctly and what was misunderstood is performed by a human and included in the report as a mandatory appendix. On temporal scope and lifecycle applicability of the protocol The ECR-VP protocol is defined for the verification of fixed document corpora representing architectural intent, constraints, and declared realizability at a given point in time. Its applicability assumes that the corpus under review is static for the duration of the verification session and that the architecture is described as a specification, framework, or design reference rather than as a continuously evolving operational system. The protocol does not natively apply to architectures that modify their own structure, invariants, or admissibility rules during operation, nor to systems whose primary representation exists only as a running instance without a stable documentary canon. In such cases, ECR-VP may be applied only to frozen snapshots or extracted canonical descriptions explicitly designated for verification. When a corpus transitions from a design-time architectural description into a continuously operating, self-modifying, or self-extending system, the protocol ceases to function as a full verification mechanism and may serve only as a historical or comparative reference across snapshots. Determination of such transition is the responsibility of the human executor and must be explicitly stated in the Corpus Passport.

2.4 AI data retention / AI data usage

Each interpreter session must be executed in a new clean chat. It is prohibited to use context from previous verification runs, to insert hints about desired conclusions, or to inform the interpreter about the opinions of other models or the author. The corpus must be transmitted in full. Partial transmission is allowed only if the corpus is physically too large, in which case the splitting procedure described in Section 6 applies. Splitting must not alter meaning or structure and must preserve canonical order and completeness. All interpreter reports are stored as immutable artifacts. Editing is allowed only as a separate commentary layer that does not alter the original responses. When intellectual-property sensitivity requires it, an initial verification run may be performed on a redacted corpus version. Full-corpus verification is permitted only in trusted or local environments.

2.5 Preparation: Canon Lock and corpus identity fixation

Before launching interpreters, corpus identity fixation is performed. The protocol executor prepares a list of files included in the session and records their checksums. Any commonly accepted hashing method is permitted (e.g., SHA-256). For each file, the name, size, date, and hash are recorded. Additionally, the version of the canonical document defining the architecture is fixed, along with the

"snapshot" date. The result of the Canon Lock stage is a Corpus Passport, which includes a brief description of the corpus, the file and hash list, the canon version, the date, the purpose of the current verification session, and explicit constraints (for example: "formal mathematics is not evaluated in this session; only coherence and realizability are assessed"). The Corpus Passport is attached to each interpreter run as the first input element. The interpreter must explicitly acknowledge the Passport at the beginning of the report and confirm that it operates strictly within this snapshot.

2.6 Procedure for loading the corpus into the interpreter

The corpus is transmitted to the interpreter in full, including the canon, main PDFs, appendices, and code if present. Public DOI links may be provided, but priority is given to transmitting the actual files, as links may be unavailable or inaccessible to the interpreter. If the corpus is large and requires splitting, sequential loading mode is applied. In this mode, the executor transmits the corpus across multiple messages in strict canonical order. The interpreter is informed in advance that the input will arrive in parts and must not form final conclusions until the final message "corpus completed" is received. Until that moment, the interpreter may only record structural notes and intermediate observations without issuing a verdict. After the final part is transmitted, the executor sends the fixed completion phrase: "Corpus completed. Execute the ECR-VP protocol. Do not ask clarifying questions. Do not request additional data. Work strictly by modes."

2.7 Reference prompt for the interpreter

"You are acting as an independent interpreter within the ECR-VP verification protocol for an architectural document corpus. Your task is to produce a report on structural integrity, readability without the author, clarity of boundaries, engineering realizability of the core, and risks of overclaiming. You do not prove mathematical correctness and do not validate scientific results. You build a map of understanding and non-understanding. You do not optimize conclusions toward the author's expectations. You do not provide patent strategy advice. You do not ask for clarifications and do not request additional files. You work strictly by the modes described below and do not mix them. Rc Mode: Describe what this architecture is as a class. What problems it addresses. How it differs from RL, planning, safe-RL, constrained optimization, and monitoring. Provide an interpretation without attempting to confirm correctness. Ri Mode: Extract invariants and prohibitions explicitly fixed by the corpus. Formulate what must exist in the system and what is forbidden. If boundaries are unclear, indicate this. Anything not directly supported by the corpus must be marked as a hypothesis and placed in Rc. Declarative Epistemic Typology Mode: Classify the corpus by epistemic layers using the fixed typology defined in the protocol. This mode is declarative and non-evaluative. Its purpose is to identify which layers are present, partially present, or absent, and where they manifest in the corpus. You must use the provided canonical list of layers and must not introduce new categories. This mode permits the use of a single table or a single bullet

list strictly for semantic classification. No judgments, recommendations, or conclusions are allowed inside the table/list. After the classification, provide a short continuous-text explanation identifying (a) where layers are mixed without explicit marking, and (b) which missing layers may affect the stated maturity or applicability of the corpus. Do not assess quality. Do not propose improvements. Do not reinterpret authorial intent. Ra Mode: Assess engineering realizability: what can be implemented as middleware/governor layers over existing systems today; what is possible only with domain-specific manual specification; what is declared but non-operational without additional definitions. Do not use philosophy here. Speak in terms of observable quantities, interfaces, integration modes, and test types. Failure Mode: Describe likely failure modes: where the architecture could become a second control loop; where metric gaming could emerge; where non-causal observation could break; where definitional drift may occur; where over-promising risks are visible. Novelty and Positioning Mode: Identify what appears genuinely non-trivial at the architectural class level and why. Distinguish structural novelty from terminological novelty. Indicate which elements are strongest candidates for grants or whitepapers as engineering innovation. Verdict Mode: Provide a concise engineering verdict: how coherent the corpus is; how readable it is without the author; how realizable the core is; where the main gaps lie. The verdict must be short and without pathos. Project Maturity Summary Mode: Provide a short operational summary of the project's maturity. This mode must be written as a single continuous text of approximately 5–10 sentences. Answer strictly the following questions: (1) What constitutes the engineering core of the architecture. (2) What minimal demonstrator or pilot could be built today without inventing missing definitions. (3) What missing specifications, interfaces, or criteria currently block grants, pilots, or engineering deployment. Do not repeat earlier analysis. Do not justify the author. Do not use philosophical language. This is not a verdict and not a recommendation, but an operational readiness snapshot. Format: Write in continuous prose, in paragraphs, without bullet points. Separate modes with headings. Do not imitate the author's style. Do not compliment. Do not provide 'how to sell' advice. Act as a strict independent reviewer."

2.8 Requirements for interpreter report format

Interpreter reports must be structured by modes and written as continuous prose. Bullet points, tables, ratings, and star systems are prohibited. Formulas are permitted only if directly cited from the corpus or if minimal notation is required to describe tests or metrics. The interpreter must distinguish clearly between corpus-derived statements and interpretive hypotheses. Exception for declarative typology The only permitted exception to the prohibition on tables and bullet lists applies to the Declarative Epistemic Typology Mode. In this mode only, the interpreter may use a single table or a single bullet list exclusively for semantic classification of epistemic layers. Any form of evaluation, judgment, scoring, recommendation, or conclusion inside the table or list is strictly prohibited. All interpretive statements must be provided only in continuous prose following the classification.

2.9 Orchestration of multiple interpreters

The protocol requires at least three interpreters, preferably five, provided that independence and heterogeneity are maintained. Each interpreter must be run in a separate session, receive the same Corpus Passport and the same reference prompt. The loading order and prompt text must be identical. Prompt adaptation to specific models is prohibited, as it breaks comparability. Interpreter diversity should be explicitly ensured by selecting models with different architectural lineages and strengths (for example, general-purpose, reasoning-oriented, or mathematically specialized models).

2.10 Result synthesis: the coherence map

The coherence map is compiled by a human executor or a separate human synthesizer operating under the same mode separation. The first layer records overlapping conclusions — elements identified consistently by most interpreters. This layer is the primary indicator of corpus readability, as it reflects structure transmitted without the author. The second layer records unique observations by each interpreter. The goal is not to select a "correct" view, but to reveal distinct angles of analysis. The third layer records areas of non-understanding and divergence. This layer is the most productive output of the protocol, as it identifies where the corpus insufficiently fixes invariants, boundaries, interfaces, or definitions.

On divergence versus canon violation The protocol deliberately treats divergent interpretations and zones of non-understanding as diagnostically valuable outcomes rather than errors. At the same time, ECR-VP acknowledges a conceptual distinction between legitimate divergence of interpretation and readings that contradict explicitly fixed canonical invariants or prohibitions. At the current version, the protocol does not introduce a formal classification of "canon-violating" interpretations and does not authorize interpreters to label interpretations as incorrect. Identification of potential canon violations is reserved for the human synthesis stage and must be explicitly grounded in cited canonical constraints. This distinction is noted as a possible future extension of the protocol and is intentionally excluded from Version 1.0 to avoid introducing evaluative authority or implicit correctness judgments at the interpreter level.

2.11 Evaluation system without ratings

To reflect reality, the protocol uses qualitative axes rather than numerical scores. The readability axis captures how consistently the architecture is understood without the author. The coherence axis captures avoidance of internal contradictions and clarity of separation between declarations and constraints. The boundary clarity axis captures how clearly prohibitions are fixed, which signals must not become rewards or gradients, and which criteria are deliberately withheld as IP. The realizability axis captures what can be implemented as middleware today, what requires domain-specific specification, and what is non-operational. The over-claim risk axis captures where rhetoric

or metaphor could be misread as technical promises. These axes are used in synthesis but must not be converted into numerical scores.

2.12 Mandatory human appendix: the author's essay

After receiving the coherence map, the corpus owner writes an appendix titled "Author's Response". This is a continuous text in which the author records what interpreters understood correctly, what they misunderstood and why, and where the corpus genuinely requires clarification. The author is prohibited from modifying the corpus within the same verification session. Any edits require a new corpus version, new hashes, and a new date. The author's essay is not a defense or argument. Its function is to fix mismatches between authorial intent and transmitted structure, preventing semantic drift and reinforcing architectural identity.

2.13 Repetition cycles and drift control

The protocol allows repeated verification sessions after corpus revisions. Each new session requires a new Corpus Passport and version. Comparison between sessions is performed via coherence maps. If overlap increases and non-understanding decreases, readability and fixation are considered improved. If non-understanding grows or new contradictions emerge, this is treated as architectural drift requiring rollback or canon clarification.

2.14 Minimum exit standard for grants or industrial pilots

If the session goal is grant or pilot preparation, the protocol defines a minimum output standard. The corpus is considered ready for external presentation if independent interpreters consistently identify the engineering core, correctly understand key prohibitions and constraints, and describe an implementable middleware layer without inventing critical definitions. If interpreters repeatedly request operational criteria and observable quantities and the corpus does not provide at least interface-level specification, the corpus is considered insufficiently prepared for grants, regardless of conceptual strength. In such cases, the protocol prescribes refinement of inputs, outputs, modes, and test scenarios without disclosing IP. For grant or pilot readiness, the Project Maturity Summary Mode is treated as a mandatory output. Absence of a clearly identifiable engineering core, demonstrator path, or explicitly stated blocking gaps in this mode constitutes automatic failure of the minimum exit standard, regardless of conceptual depth or novelty.

2.15 Protocol conclusion

The ECR-VP protocol transforms independent reading of a corpus by multiple interpreters into a formalized act of architectural verification. It separates actual readability and coherence from the

author's internal conviction. It does not replace experiments, mathematical proofs, or engineering testing, but significantly reduces the risk that an architecture is "understandable only to its author" and creates a stable foundation for academic formalization, grant applications, and industrial pilots.