CNVS

The Theory of Closed Native Verification Systems

Version 6.0

This article attempts to conceptualize and formally define CNVS, providing the Author's reflections on the process of reconstructing the foundations of the Theory.

Author: Massimo Comitato

17.06.2026, Italy (EUR)

Table of Contents

---

1. Introduction

This article formalizes an asymptotic theoretical system that aims to be an ideal model for attesting and verifying empirical evidence for the existence of an object external to the system itself, starting from initial statements and definitions introduced within it.

In this paper, the object is understood in its extended form as any reality that can be supported by specific, unambiguous, and objectively identifiable empirical evidence.

In short, this theoretical model attempts to reconcile a specific truth statement with the objective verification of its consistency in empirical reality, regardless of the nature of the object, whether material or abstract.

In this sense, the CNVS framework addresses one of the oldest and most fundamental problems of human society: how to determine whether an object, event, or statement actually exists without relying on a specific authority. In modern distributed Systems, this challenge is known as the oracle problem.

CNVS is proposed as a formal Systems theory framework that offers a new general mathematical perspective to address this problem through recursive information decomposition, randomly distributed empirical verification, and Global validity emerging from the convergence of verification results with respect to input information.

---

2. Abstract

This document defines a new class of Systems called **Closed Native Verification Systems (CNVS)**.

In these Systems, the existence, validity, and evolution of the state are determined exclusively by internal verification processes supported by randomly distributed external verification evidence and ultimately governed by deterministic invariants.
Although the objective verification processes remain external, their distribution—that is, the specific structure imposed by the System—reduces confidence in the execution of a rigid application scheme that limits the executor's discretion through the progressive reduction of Local and Global operational knowledge.
Specifically, the System seeks to converge declarations made in the form of user input requests and subsequent verification by external verification agents randomly selected by the System itself and operating according to the imposed constraints.
The External Verification Agents are tasked with verifying individual details of the information content of the User's input request, without knowing the result, and subsequently reporting them so that the System can cross-check the collected data to ensure complete consistency.
In this sense, the System aims to reveal the Truth of Existence within itself as a cross-verification process between independent and randomly distributed operational flows: the User input request and the Verification of the External Agents managed by the System itself.

The framework models the transformation of user-declared information into recursively decomposed structured elements, which are randomly assigned to External Verification Agents, each operating under strict protocol constraints. Local verification evaluates attribute-specific convergence between internally

declared data and observed proofs, while Global external validity emerges from the consistency of the reconstructed state and cannot be reduced to the conjunction of Local verifications. By integrating recursive decomposition, information-theoretic knowledge restriction, and emergent state validation, CNVS is proposed as a general mathematical framework that addresses the oracle problem, one of the key unsolved challenges in distributed Systems.

The Theory presupposes Three Fundamental Axioms, which are imposed as a Necessary Condition for classifying a System as a Native Verification System. The Axioms define:
- the Structural Properties of the entire System;
- the Emergent Behavior related to the Security and Verification Dynamics;
- the Principles that guide the Dynamics of External Processes and govern those of Internal Processes.

---

3. Problem Statement

One of the key unsolved problems in distributed Systems is the oracle problem: the difficulty of determining whether information from the physical world can be introduced into a digital System without relying on a single trusted source. Existing approaches, including multisource aggregation, cryptographic attestations, reputation mechanisms, consensus protocols, and secret sharing techniques, mitigate specific aspects of this challenge but do not eliminate the fundamental dependence on external trust.

In particular, existing Systems typically validate externally provided statements but do not model verification as an intrinsic condition for the existence of the

State. They generally do not impose a recursive structural decomposition of information, do not Systematically constrain the Local knowledge of individual Verification Agents, and do not formalize Global validity as an emergent property irreducible to the convergence of Local verification results.

Closed Native Verification Systems (CNVS) are proposed as a distinct theoretical framework in which user-declared information is recursively fragmented, randomly distributed to external verification agents under specific constraints, and reconstructed exclusively at the System level. Each Local verification evaluates the convergence between the internally declared value and independently observed evidence for a specific attribute. The System ultimately returns only a binary decision: the Object (also understood as the assertion) exists within the System or not.

By replacing unilateral trust with distributed empirical convergence under structural information constraints, CNVS are proposed as a general mathematical framework that addresses the oracle problem in a fundamentally different way from existing architectures.

In Traditional Systems that address the same problem, we find that:
- the truth about the existence of a state is a unilateral decision by a governing authority (total trust);
- verification of the object is entrusted to external auditors, who are at most governed by Global constraints or laws, with few Local references;
- External Verification Agents are entrusted with the discretionary task of issuing authoritative statements about the object that are difficult to contest;
- External Verification Agents are chosen by the governing authority, which also imposes the control rules;
- each external verification agent is always responsible for verifying the same specificity;
- each external verification agent has multiple pieces of information relating to the data of the Object of Verification;
- the existence of the state is admitted in the Traditional System as natively

imposed by the issuing authority;
- verification of the state follows the imposition of existence;
- the state is almost always unitary;
- the overall security of the System's interior decreases with the adoption and increase in the number of states, because the surface area for external attacks increases.

In a CNVS, the System inverts this logic:
- the truth about the existence of a state is not a decision unilaterally imposed by a governing authority, but the result of a mathematical process internal to the System;
- verification is entrusted to Verification Agents external to the System, but strictly guided and governed according to the application scheme of a Distribution Model that uses information access restriction as a method of collusion control;
- the attestation of evidence issued by the external verifier must first pass strict internal verification checks to ensure the logical and structural consistency of its content with respect to the entire System;
- the state is not unitary;
- the External Verification Agent not only verifies the existence of evidence of an Object's specificity, but is also required by the System to provide data related to the assigned specificity. However, it does not know the data declared during the input phase when the User creates the Object Existence Verification request, thus allowing the System to compare these two entities through information convergence;
- the External Verification Agent is chosen randomly;
- the Object's specificity that the External Verification Agent is required to measure and control is chosen randomly by the System;
- the measurement grids assigned to the external verifier are non-uniform among themselves and with respect to the global information;
- security emerges as a Property of the System and grows with its increase Adoption due to the System's very Structure.

4. General Definitions

In this document, a **System** is defined as a Structured Set of Objects and Elements, regardless of Type, that have specific interrelationships and are governed by preliminary Assumptions that define its Functions.
A System is considered Closed when no External Intervention can alter its Structure, nor can it produce a result that conflicts with the preliminary Assumptions imposed through its Execution.

In this document, a System is defined as "**Natively Verified**" because the very Existence of all the Elements and Objects that comprise it is admitted only after an Objective Verification, followed by the Application of a Consistency Check of the issued Certifications.

In other words, no Element can "become" if it has not first been Verified.
In this document, a Natively Verified System is defined as Closed because no unauthorized External Intervention can alter its Structure, nor can it produce results that conflict with the preliminary Assumptions imposed through its Execution.

For the purposes of the **CNVS** model, Information (Ix) is defined as any ordered and finite binary representation admitted by the System. Physical objects, measurements, attestations, certificates, signatures and attributes, declarations, statements, postulates, sentences, and historical data are all processed only through their Encoded Representation as elements of {0, 1}*. Thus, the System does not directly process physical or immaterial Objects, but operates exclusively on their verifiable digital representation.

5. Special Definitions

5.1 Object

Let
O_x
be the Object to be Verified,

let
$I(O_x)$
the complete intrinsic informational structure associated with the Object,

let
$\tilde{I}(O_x) = \text{Encode}\,(I(O_x))$ **(1)**
the encoded binary representation used by the System,

let
$E(t) = \text{Struct}\,(\tilde{I}(O_x), t),$ **(2)**
with Struct $(\tilde{I}(O_x))$ constituting the Set of all Structured Elements derived from the Object's binary representation at time t,

let

The Chain of Events Generated by the User Request:

$O_x \rightarrow I(O_x) \rightarrow \tilde{I}(O_x) \rightarrow E$ **(3)**

The System allows the User to initiate a Verification Instance on an Object for which Evidence of Existence is to be established.

The System responds by building a structured information representation and encoding it in binary format. It then recursively decomposes the information into operationally separate, epistemically incomplete, and structurally related fragments, according to a recursive scheme explained in the following chapters.

A valid State emerges if and only if the provided information satisfies both Local and Global invariant verification through the distributed evaluation process performed by the System.

5.2.1 Structural Universe

The Structural Universe of the System is defined as the set of all possible structural configurations that the System may assume according to the types, domains, and admissibility rules established by the model.

Since all information entering the System is processed through a finite digital representation, we assume the binary alphabet as the basic alphabet:

$B = \{0, 1\}$

and we denote by: B^*

the set of all finite binary strings.

Therefore, objects, declarations, data, attributes, observations, certificates, signatures, identifiers, and records are not processed by the System as immediate physical or semantic entities, but as encoded representations belonging to suitable structured domains constructed over B^*.

The System does not operate directly on external reality, but on its formally admitted representation within the model.

On this basis, we define the spaces, domains, and structural types that compose the Structural Universe of the CNVS.

We define $\mathcal{T}$ as the finite set of all structural types admitted by the System.

Since semantic coherence and type compatibility must be guaranteed at every level of decomposition, independently of the nature of the considered Object, the primitive domains are not defined as flat sets, but as families of sets indexed by their structural type $\sigma \in \mathcal{T}$.

Recursive decomposition does not necessarily require the type of a child fragment to be identical to the type of its parent fragment. Instead, it requires the child fragment to remain structurally compatible with the original type, or to represent a System-admitted specification of it.

Formally, this can be expressed by introducing a compatibility or refinement relation between types:

$\sigma' \preceq_{\mathcal{T}} \sigma$

where σ is the type of the parent fragment and σ' is the type of the fragment generated by decomposition.

Therefore, for every child fragment $s' \in D_t(s)$, the System requires:

$type(s') \preceq_{\mathcal{T}} type(s)$.

In the special case where decomposition exactly preserves the type, this condition reduces to the stronger form:

$type(s') = type(s)$.

The definition of the Structural Universe is therefore:

$$□ □ \Sigma_{\sigma \in \mathcal{T}} (Id_\sigma \times D_\sigma \times At_\sigma \times Obs_\sigma^{\perp} \times \mathcal{P}(P)) \quad \textbf{(4)}$$

Consequently, each structural element $s \in □$ assumes the dependent form:

$s = \langle \sigma, (id, d, a, obs, p) \rangle$

where:

• $Id_\sigma \subseteq B^*$: the space of identifiers for type σ.

• $D_\sigma \subseteq B^*$: the space of declared data payloads structured according to type σ. An element $d \in D_\sigma$ represents a candidate assertion, digital archive, structured datum, object specification, or cryptographic record submitted by one or more users, entities, or system agents to the CNVS layer for verification.

• $At_\sigma \subseteq B^*$: the space of attribute classes for type σ.

• $Obs_\sigma \subseteq B^*$: the space of externally reported values for type σ.

Let P be a finite set of logical properties, and let $\mathcal{P}(P)$ denote the set of its finite subsets.

To model the absence of data without introducing domain collisions or ambiguity with the empty string $\varepsilon \in B^*$, we introduce a distinguished element $\perp \notin B^*$. We define the observational domain as a disjoint union:

$Obs_\sigma^{\perp} \triangleq Obs_\sigma \uplus \{\perp\}$

where the ⊥ element rigorously denotes the absence of an external observation at non-terminal structural levels.

This construction guarantees that identifiers, payloads, attributes, observations, and generated properties are intrinsically bound to their specific structural type σ at instantiation, preventing type-domain mismatches prior to any verification phase.

5.2.2 State

We formally define the state of the System at time t as a well-formed topological and logical tuple:

$S(t) := \langle E(t), R(t), C \rangle,$ **(5)**

where:

- $E(t) \subseteq$ [illegible] is the finite set of structural elements admitted and active within the System at time t. Each element $s \in E(t)$ assumes the dependent and rigidly typed form defined in the previous chapter: $s = \langle \sigma, (id, d, a, obs, p) \rangle$.
- $R(t) \subseteq E(t) \times E(t)$ is the relational graph that uniquely defines the structural dependencies and direct connections among the active elements.
- $C = \{c_1, \ldots, c_m\}$ is the finite set of deterministic global invariants (consistency constraints) imposed by the System to preserve the integrity of the overall state.

In accordance with the principle of existential independence (Axiom I), the CNVS framework strictly distinguishes between the space of candidate configurations and the space of valid accepted states. During an operational cycle or a transition attempt, the System generates and manipulates a temporary set of candidate elements, denoted as $E_{cand}(t)$, thereby structuring a provisional candidate state:

$$S_{cand}(t) = \langle E_{cand}(t), R_{cand}(t), C \rangle. \quad \textbf{(6)}$$

The relation $s \in E_{cand}(t)$ expresses a pure, primitive structural membership, which logically precedes any validation operation and remains unaltered by it. If and only if the entire candidate state satisfies both local and global constraints, the elements of $E_{cand}(t)$ are formally admitted and evolve into the definitive valid set of the subsequent transition:

$E(t) = E_{valid}(t)$.

If the Global Verification fails, the candidate state is entirely rejected, precluding the admissibility of elements into E_{valid} and triggering the immediate dissolution of the provisional topology (*Topological Refresh*).

To govern admissibility and state evolution without introducing syntactic redundancies or misaligned intermediate domains, we define two fundamental verification functions operating in a coordinated yet rigidly separated manner across distinct structural levels:

Local Verification Function (V_L):

It operates within the Structural Universe to evaluate the formal integrity and attribute convergence of a single, isolated element. It maps the element directly into an ordered pair comprising a Boolean validity flag and the set of its logical properties:

$V_L : \rightarrow \{0, 1\} \times \mathcal{P}(P)$.

Given an element $s = \langle \sigma, (id, d, a, obs, p) \rangle$, the operational expression of V_L is executed through the strict logical conjunction of the formal type admissibility predicate ($Form_\sigma$) and the attribute convergence predicate relative to the external observation ($Conv_\sigma$):

$$V_L(s) \langle Form_\sigma(id, d, a, obs) \wedge Conv_\sigma(a, d, obs), p \rangle \quad \textbf{(7)}$$

The canonical projection onto the first component, $\pi_1(V_L(s))$, extracts the local validity bit required to determine the admissibility of the terminal element.

Global Verification Function (V_G):

It represents the absolute structural veto of the framework. Unlike the local function, it neither operates on individual elements nor generates sets of properties; rather, it evaluates the entire state space ($\mathfrak{S}$), returning a purely binary outcome:

$V_G : \mathfrak{S} \rightarrow \{0, 1\}$.

Leveraging the hierarchical stratification of the System, the global verification recursively examines the configuration starting from the leaf terminal fragments (Ter(t)) upward to the root nodes, simultaneously enforcing relational graph consistency ($Cons_R$) and the verification of all system invariants $c_i \in \mathcal{C}$:

$$V_G(S(t)) = 1 \Leftrightarrow (\forall s \in Ter(t), \pi_1(V_L(s)) = 1) \wedge Cons_R(R(t), E(t)) \wedge (\forall c_i \in \mathcal{C}, c_i(E(t), R(t)) = 1) \quad \textbf{(8)}$$

This architecture ensures that while the admissibility of elementary fragments is delegated to local and isolated evaluations, the transition and acceptance of the system state remain holistic properties that cannot be reduced to the mere

sum of their parts, thereby guaranteeing decidability and protection against topological inference attacks.

5.2.3 Admissibility

The CNVS framework resolves the ambiguity between data existence and data validity by rigorously separating primitive structural membership from logical admissibility.

We define the **Local Admissibility** condition (Adm_L) as a strictly terminal evaluative function:

$Adm_L : \square \rightarrow \{0, 1\}$

For every terminal structural element $s \in Ter(t)$, admissibility evaluates to true (equal to 1) if and only if the Boolean component of the Local Verification Function yields a positive outcome. Formally:

$$Adm_L(s) = 1 \iff \pi_1((V_L(s)) = 1, \quad \textbf{(9)}$$

where π_1 denotes the canonical projection onto the first component of the tuple returned by V_L.

Unlike traditional hierarchical formulations, the CNVS does not evaluate independent admissibility for intermediate nodes. Strict admissibility is an *exclusively local* property of leaf fragments. The acceptance of intermediate and higher-level configurations is entirely delegated to the absolute veto of the Global Verification (V_G), emerging as a state property rather than an attribute of an individual node. This ensures that no superstructure can be deemed "admitted" in the presence of silent topological violations.

Operationally, once a terminal fragment successfully passes the local admissibility verification ($Adm_L(s) = 1$), the System promotes it as a candidate component for the higher structural levels. During the subsequent hierarchical ascent toward the root, the Global Verification (V_G) assumes the intrinsic validity of the fragment and proceeds to verify its topological coherence within

the relational graph (R) and its strict compliance with the system invariants (C) across the intermediate ranks ($S^{(k)}$), ultimately consolidating the absolute global state (S^0).

5.2.4 Decomposition as a Typed Function

Let D be the primary decomposition operator, defined as a function mapping a well-formed structural element into a strictly finite set of its constituent sub-elements:

$\mathfrak{D} : \square \rightarrow \square_{\mathbf{fin}}(\square)$

To ensure logical consistency, prevent trivial fragmentations (single-child nodes), and avoid infinite loops, we impose a strict disjunctive cardinality constraint on the output of $\mathfrak{D}(s)$ for any element $s \in \boldsymbol{\mathcal{S}}$:

$|\mathfrak{D}(s)| = 0 \vee |\mathfrak{D}(s)| \geq 2$ **(10)**

- If $|\mathfrak{D}(s)| = 0$ (i.e., $\mathfrak{D}(s) = \emptyset$), the element cannot be further decomposed. It is formally classified as a terminal fragment (leaf node) and belongs to the set Ter(t).
- If $|\mathfrak{D}(s)| \geq 2$, the element is decomposed into a finite set of child fragments $\{s_1', s_2', \ldots., s_k'\}$.

As established in Chapter 5.2.1, for every generated child fragment, the structural type must respect the compatibility relation with its parent ($type(s_i') \preceq_{\mathcal{T}} type(s)$). This guarantees that the operator consistently functions as a *typed function*, preserving the semantic coherence of the Structural Universe $\square$. In this document, the term "Terminal Fragment" (or simply "Fragment") explicitly refers to elements for which $|\mathfrak{D}(s)| = 0$, undergoing no further operational decomposition within the System.

5.2.5 Type Compatibility and Conceptual Type Preservation

Let

$\mathcal{T}$

be the finite set of all admissible structural types recognized by the System.

Let

type : $\mathcal{S} \to \mathcal{T}$

be the Type Function, assigning to each structured element its structural type.

Each structured element is associated with exactly one structural type through the Type Function.

However, recursive decomposition does not require strict identity of type between a parent element and each of its descendants. What must be preserved is not necessarily the identical type label, but the structural compatibility, semantic coherence, and admissible refinement relation between the parent type and the descendant type.

Formally, let

$\preceq_{\mathcal{T}}$

denote the compatibility or refinement relation over structural types.

For every structured element s ∈ □ and every descendant s′ ∈ $\mathfrak{D}$(s), the System requires:

type(s′) $\preceq_{\mathcal{T}}$ type(s). **(11)**

This means that the type of the descendant must remain compatible with, or represent a System-admitted specification of, the type of the parent element.

The stronger condition

type(s′) = type(s)

is therefore only a special case of exact type preservation.

Thus, decomposition preserves the conceptual and structural coherence of the type, but it does not necessarily require strict type identity at every depth level.

5.2.6 Recursive levels

Let

$E^{(0)}(t)$ be the root nodal level of the System at time t.

The recursive decomposition levels are inductively defined as the union of all fragments generated by the decomposition of the elements from the preceding level:

$$E^{(n+1)}(t) = \bigcup_{s \in E^{(n)}(t)} \mathfrak{D}(s) \quad \textbf{(12)}$$

and

For every admissible decomposition level n generated by the System at time t, the following condition holds:

$0 \leq n \leq \text{Depth}(t)$.

where Depth(t) represents the maximum depth of the decomposition tree at that given time.

5.2.7 Parent map

Let par be the parent map linking the elements of a given level to those of the immediately preceding upper level:

$\text{par} : E^{(n+1)}(t) \rightarrow E^{(n)}(t)$

such that

$$\text{par}(s') = s \iff s' \in \mathfrak{D}(s) \quad \textbf{(13)}$$

The parent map is well-defined under the assumption that each generated descendant carries a unique parent reference, encoded through its internal identifier and relational reconstruction markers.

5.2.8 Conservation of information-theoretic

Let

$\mathbb{I} : \square \to \{0, 1\}^*$

be the Information Encoding Function.

For every structured element s, $\mathbb{I}(s)$ denotes its canonical binary representation and $|\mathbb{I}(s)|$ its total informational complexity measured in bits.

The Chain of Events, from the previous definitions, consolidates as:

$O_x \to I(O_x) \to \tilde{I}(O_x) \to E \to s \to \mathbb{I}(s)$.

To guarantee the logical reversibility of the system, we introduce the Reconstruction Function:

$\text{Rec} : \boldsymbol{\mathcal{P}}_{\text{fin}}(\square) \to \square$ **(14)**

be the Reconstruction Function.

The function Rec recomposes a structured element from the complete admissible descendant set generated by the decomposition process, together with the relational and structural markers required by the System.

For every structured element $s \in \square$, informational integrity is ensured by the identity:

$\mathbb{I}(\text{Rec}(\mathfrak{D}(s))) = \mathbb{I}(s)$.

The backward Chain of Events thus becomes:

$\text{Rec}(\mathfrak{D}(s)) \rightarrow \mathbb{I}(\text{Rec}(\mathfrak{D}(s))) = \mathbb{I}(s)$

5.2.9 Non-uniformity

As established by the formal theory of the framework, for every structured element $s \in \square$, the total informational complexity is decomposed into two strictly positive functions:

- $I_{payload} : \square \rightarrow \mathbb{N}$: denoting the number of bits corresponding to the effective semantic content inherited from the original object or claim.
- $\mathbb{I}_{metadata} : \square \rightarrow \mathbb{N}$: denoting the minimal operational information required for identification, assignment, and graph coordination.

Therefore, we have the linear sum:

$|\mathbb{I}(s)| = I_{payload}(s) + \mathbb{I}_{metadata}(s)$

The payload belongs to the semantic information plane, whereas the metadata belongs to the systemic structural information plane.

To prevent symmetric data inference by an adversary, the System enforces a strict constraint of **Semantic Non-Uniformity** (Axiom II). For any decomposed non-terminal element, there exist at least two child fragments carrying strictly unequal payload volumes:

$$\exists s'_a, s'_b \in \mathfrak{D}(s) : I_{payload}(s'_a) \neq I_{payload}(s'_b) \quad \textbf{(15)}$$

Furthermore, decomposition adheres to the following laws of conservation and structural overhead (Lemma 4 and Lemma 5):

1. **Payload Conservation:** $I_{payload}(s) = \sum_{s' \in \mathfrak{D}(s)} I_{payload}(s')$
2. **Metadata Overhead:** $I_{metadata}(s) < \sum_{s' \in \mathfrak{D}(s)} I_{metadata}(s')$

This demonstrates that, since decomposition requires the generation of new identifiers and markers for each fragment, the aggregate total information volume increases ($\sum |\mathtt{I}(s')| > |\mathtt{I}(s)|$). It is therefore clear that recursive decomposition must necessarily admit a limit: fragmentation is permissible if and only if the ratio between payload and metadata remains above a Critical Threshold $\boldsymbol{\chi} > 0$.

If the System required a metadata volume that overshadowed the semantic payload itself, the framework would become operationally inefficient and saturated. In the "Operational Flow" chapter, this practical limit was established at 4 or 5 nodal levels, regulated by the System Control Parameters (λ, I_0, ρ_C), adapting it to the I_{min} value while respecting topological constraints.

It should be noted, however, that the relative weight of metadata within a single CNVS fragment is drastically reduced. Unlike conventional blockchain architectures, much of the coordination information does not reside within the terminal fragments themselves, but is securely delegated to the relational graph topology R(t) governed by the Global Verification. CNVS is not a metadata-heavy reconstructive system, but rather an invariant system driven by relationships.

5.2.10 State and Structure

The State of the System at time t is not a mere collection of elements, but a typed relational structure whose elements are admitted only insofar as they preserve the structural schema imposed by the System and satisfy their respective verification conditions (Local Admissibility Adm_L for terminal fragments, and Global Verification V_G for the holistic acceptance of the topological state).

Let

$$\mathcal{U} \subseteq \Sigma_{\sigma \in \mathcal{T}} (Id_\sigma \times D_\sigma \times At_\sigma \times Obs_\sigma^{\perp} \times \mathcal{P}(P)),$$

be the structural universe of admissible elements. Each structured element therefore assumes the dependent form:

$s(t) = \langle \sigma, (id, d, a, obs, p) \rangle \in \mathcal{U}$.

The State at time t is defined as the tuple:

$S(t) := \langle E(t), R(t), C \rangle$,

where $E(t) \subseteq \mathcal{U}$, $R(t) \subseteq E(t) \times E(t)$, and $C = \{c_1, ..., c_m\}$ is the finite set of deterministic invariants imposed by the System.

The System is recursive because every structured element may generate a finite set of structured descendants while preserving structural compatibility with its parent type, or by producing a System-admitted refinement of it.

Formally, let

$$\mathfrak{D} : \mathcal{U} \to \mathcal{P}_{fin}(\mathcal{U})$$

be the decomposition function, subject to the disjunctive cardinality constraint to prevent trivial fragmentations:

$$|\mathfrak{D}(s)| = 0 \vee |\mathfrak{D}(s)| \geq 2$$

and the type compatibility constraint:

$$\forall s' \in \mathfrak{D}(s), \text{type}(s') \preceq_{\mathcal{T}} \text{type}(s)$$

The stronger condition type(s') = type(s) is only the special case in which decomposition exactly preserves the type.

Therefore, decomposition modifies the informational granularity of the element, but not its structural nature. The fragment is not an arbitrarily different entity: it remains a structurally compatible entity, or a System-admitted refinement, at a lower informational scale.

The decomposition levels are defined recursively starting from the root level $E^{(0)}(t)$:

$$E^{(n+1)}(t) = \bigcup_{s \in E^{(n)}(t)} \mathfrak{D}(s) \quad \textbf{(16)}$$

In the operative model, n ≤ Depth(t).

In the ideal model, the recursive structure is defined for arbitrary finite n.

The conservation of information is expressed through a reconstruction operator Rec, such that:

$\mathbb{I}(\text{Rec}(\mathfrak{D}(s))) = \mathbb{I}(s)$

However, non-uniform decomposition enforces a strict asymmetry in semantic distribution. Different terminal fragments may carry unequal informational weight:

$\exists s'_a, s'_b \in \mathfrak{D}(s) : I_{payload}(s'_a) \neq I_{payload}(s'_b)$

5.2.11 Intuitive Interpretation

To intuitively understand the operational behavior of the system, we summarize the roles of the domains composing the State at time t.

The State is not a mere passive collection of data, but is constrained by the relations R(t) and the Deterministic Invariants $C = \{c_1, c_2, \ldots, c_m\}$. Local Admissibility ($V_L$) and the Global Veto ($V_G$) determine logical consistency prior to the formal declaration of an accepted state.

Since Structured Elements are not atomic, but are fragmented to render verifiers semantically blind, the framework assumes a recursive structure.

Within a given terminal element $s = \langle \sigma, (id, d, a, obs, p) \rangle$:

- **d (Data Payload):** Represents the internally declared data (e.g., "2 mL," "4 Hz," "4.2 grams," "Saturday afternoon," "at 3 a.m.," "objectively ugly," "90 cm," "40 km/h," etc.).
- **a (Attributes):** The attributes defining the class of the data d and its tolerance rules (e.g., quantity, internal state, timestamp, turbidity, volume, weight, "clothing worn," "recorded body temperature," speed, distance, etc.).
- **obs (Observations):** The validation statements (audits, certificates, signatures, attestations of truth, testimonies, physical measurements, etc.) independently reported by the external Verifier, which must converge upon the internal data d. At non-terminal levels, obs = $\perp$, because external observations occur exclusively at leaf nodes.
- **p (Properties):** The set of logical properties produced by the Local Verification functionV_L (e.g., {Valid, Invalid, Transferable, Blocked, Suspended, Inconclusive, Doubtful, etc.}).

Since:

$V_L(s) = (b_i, P_i)$

the set of properties generated for the terminal element is extracted from the second component of the tuple:

$P_i = \pi_2(V_L(s))$

While Local Admissibility generates these semantic properties at the foundation, the Global Verification V_G executes the most critical structural computation of the system: it evaluates the strict relational coherence of the entire graph and enforces compliance with the global invariants. Despite performing this massive computation, V_G does not generate new textual properties; rather, it synthesizes the validity of the entire state by returning an absolute Veto (1 or 0)

Hierarchy and Non-Uniformity:

During the decomposition process, child elements belong to structurally compatible families with respect to their parent element. For every structured element $s \in E(t)$ and within the maximum admissible depth Depth(t), there exists a finite decomposition sequence mapping the root element through intermediate nodes down to the terminal fragments.

As established by Axiom II, *uniformity* is explicitly not maintained during this fragmentation process. The obtained sub-elements do not preserve equal-volume fractions of the original semantic payload ($I_{payload}$).

Non-uniform decomposition does not compromise the Principle of Information Conservation in any way ($\mathbb{I}(Rec(D(s))) = \mathbb{I}(s)$), but it renders adversarial reconstruction mathematically harder, as terminal fragments carry unequal informational weights under restricted relational accessibility.

5.3 Relations

Let

$$R(t) \subseteq E(t) \times E(t) \quad \textbf{(17)}$$

where,

$E(t) \times E(t)$ represents the Cartesian product of all possible pairs of Structured Elements active in set E at the same time t. The set R(t) is exclusively given by the structurally admissible directional relations.

For example, if the active state contained only two elements, $E(t) = \{s_1, s_2\}$, the Cartesian product would define the theoretical connection space:

$E(t) \times E(t) = \{(s_1, s_1), (s_1, s_2), (s_2, s_1), (s_2, s_2)\}$

The System does not necessarily instantiate all possible relations; it only permits topologically valid ones. If, for instance:

$R(t) = \{(s_1, s_2), (s_2, s_2)\}$

the Directional Relation would solely exist originating from s_1 directed to s_2, and as a logical self-reference for s_2. The System does not fully recognize the entire Cartesian product E x E, but strictly selects a subset.

While the specific domain implementation dictates the semantic rules of decomposition, the CNVS framework imposes an absolute structural constraint on the permitted relations through the Relational Consistency predicate ($Cons_R$). A relation exists if and only if it faithfully maps the decomposition operator:

$(s, s') \in R(t) \iff s' \in \mathfrak{D}(s)$

Due to this hierarchical and causal nature, Relations within the System are **always directional and strictly asymmetric**. Verification relies on a precise topological ordering, such that the edge (s_i, s_j) is a distinct logical entity from (s_j, s_i).

Formally, the existence of a parent-to-child dependency inherently precludes the inverse relation:

$\forall (s_i, s_j) \in R(t) \nRightarrow (s_j, s_i) \in R(t)$

All Relations allowed in the System must be:

- Logical.
- Structural (bound to the $\mathfrak{D}$ function).
- Necessarily Directional (asymmetric).
- **Devoid of quantitative weights.** The connections delineate the dependency topology for the Global Verification; they do not carry numerical weights between Structured Elements.

Non-uniform fragment decomposition does not by itself alter the admissibility or structural validity of relations, provided that the type-compatibility constraint and the relational coherence imposed by the System are strictly preserved.

The pair (E(t), R(t)) therefore defines a directed relational graph over the Structured Elements of the System. Recursive decomposition induces a hierarchical graph structure wherein:

- Each structured element $s_i \in E(t)$ corresponds to a vertex (node).
- Each admissible relation $(s_i, s_j) \in R(t)$ corresponds to a directed edge.
- Decomposition levels define a tree structure in which nodes generate descendants through recursive fragmentation, preserving type consistency.
- Terminal fragments (Ter(t)) represent the leaf vertices at the deepest levels of decomposition.

The resulting Graph therefore captures both the structural dependencies among elements and the recursive topology of the State decomposition. Consequently, although terminal Fragments remain epistemically incomplete from the perspective of an isolated verifier, the relational structure R(t) fully preserves the structural dependencies required for global consistency and reconstructive coherence.

5.4 Id_i, Data (D_i) e Attributes (At_i)

Id_i is the unique identifier of the Structured Element s_i. It contains the minimal encoded identifiers required for recursive fragment association and reconstructive consistency within the System (as detailed in the chapter dedicated to the System's Operational Flow).

Generally speaking, the System does not maintain a complete, explicit topological map of the Fragment reconstruction upfront. Instead, reconstructive coherence emerges dynamically from the relational structure R(t) and the recursive dependency structure preserved across decomposition levels.

At each recursive decomposition step, the System associates the Structured Element $s_i^{(n)}$ with unique encoded identifiers that never leave the internal

architecture. These keys provide an internal set of information for the exclusive use of the System. During a reconstruction process, or more simply to formulate the precise task assigned to the External Verification Agent, the System uses Id_i to map the relational dependencies associated with the Structured Element within R(t).

The Algorithmic Immune System via R(t) Mapping:

(nota: The expression "Algorithmic Immune System" is used here as an architectural metaphor to describe the System's traceability, isolation, and rejection mechanisms. It does not introduce an additional formal subsystem beyond R(t), the verification functions, and the invariant structure C)

It is precisely at this juncture that the structural identifier (Id_i) meets the physical verifier. Upon random assignment, the topology R(t) establishes a strict, bi-univocal mapping between the fragment's Id_i and the unique identity of the assigned External Verifier ($v_k \in \mathcal{V}$). This guarantees absolute accountability without exposing the payload: if the verifier v_k acts maliciously or executes a freeze attack, the System utilizes R(t) to instantly trace the obstruction, isolate the affected Id_i branch, and permanently expel v_k from the active pool. This dynamic ensures the progressive algorithmic purification of the network.

The Payload (D_i) and Bottom-Up Reconstruction

Complete reconstruction of the information content of the current level (n) is strictly necessary for the reconstruction of the information of the previous, higher level (n-1).

This is because only after complete reconstruction of the current level is it possible to reveal the Id structure of the parent decomposition level (n-1), which is necessary for the next recomposition. The system is therefore designed to allow only internal reconstruction of the graph by progressive recompositions.

The data payload D_i follows the exact same recursive decomposition logic as s_i. Its binary content is progressively reconstructed during each recomposition cycle from level n to level n-1. Consequently, the complete reconstruction of a

preceding decomposition level requires the coherent, verified reconstruction of the informational content preserved at the current, lower level.

In accordance with the Principle of Knowledge Restriction, the complete D_i content remains internally restricted within the System and is not directly accessible to External Verification Agents. In other words, D_i never leave the System.

Attributes (At_i) and the Granularity of Information

Let us now examine the Attributes. Formally, we define the attribute set as:

$$At_i = \{a_{i1}, a_{i2}, a_{i3}, ..., a_{ik}\} \quad \textbf{(18)}$$

Attributes represent observable classes or descriptive categories, not abstract logical properties. The Attribute is, therefore, a category or class of objective description of the Structured Element at the considered level of decomposition. Attributes may exhibit internal dependencies, but these cannot be modeled as relationships within the native System; rather, they are defined by the specific, external implementation context.

Crucially, the attribute class also undergoes a recursive decomposition according to the proposed model, symmetrically following the decomposition of the D_i data it represents. The depth of this decomposition defines the level of detail or resolution of the information and is a direct function of the Granularity of the Global Information (see Chapter 14).

Practical Example: Recursive Attribute Decomposition

To better clarify recursive decomposition at the attribute class level, consider the verification of a specific physical asset.

Suppose the object of verification is the existence of a *"plaid shirt, with a right sleeve length of 60 cm and a left sleeve length of 60.5 cm (slight asymmetry), shoulder width of 47 cm, height from shoulder to hem of 75 cm, total wingspan (right sleeve + left sleeve + shoulders) of 167.5 cm. The plaid squares are 3 cm x*

5 cm, except on the sleeves where they are 2 cm x 3 cm. The pattern is red-striped on a blue background."

An attribute class at an aggregate level (n-2) might simply be:

- Total wingspan from the end of the right sleeve to the end of the left sleeve (167.5 cm).

The attribute class at the next recursive decomposition level (n-1) would divide this into:

- Length of the right arm from the edge of the sleeve to the shoulder (60 cm).
- Length of the left arm from the edge of the sleeve to the shoulder (60.5 cm).
- Width of the shoulders (47 cm).

The attribute class at the final, terminal recursive decomposition level (n) would fragment the task further:

- Length of the shirt from the edge of the left sleeve to the left elbow.
- Length of the shirt from the left elbow to the left shoulder.
- Length of the shirt from the edge of the right sleeve to the right elbow.
- Length of the shirt from the right elbow to the right shoulder.
- Width of the shirt from the edge of the right shoulder to the center.
- Width of the shirt from the edge of the left shoulder to the center.

Conclusion of the Example: The purely educational significance of this example demonstrates how a single macroscopic attribute (level n-2) is systematically broken down into distinct, isolated local verification tasks (level n). This process directly mirrors the structural decomposition of the payload itself, ensuring that no single verifier at level n possesses the necessary context to infer the global state.

5.5 Properties (P_i)

Properties are logical conditions acting upon a single Structured Element s_i. Examples of properties include states such as: *valid, transferable, blocked, verified, suspended, accepted, questionable, corrupted*, among others.

Crucially, properties are not raw data; rather, they are **Evaluations** that depend exclusively on the System's logic.

Let $\boldsymbol{\mathcal{P}}$ be the finite set of admissible logical properties defined by the System. For each structured element s_i, we define its property set P_i as a subset of $\boldsymbol{\mathcal{P}}$:

$P_i \subseteq \boldsymbol{\mathcal{P}}$

Equivalently:

$P_i \in 2^{\boldsymbol{\mathcal{P}}}$ **(19)**

where $2^{\boldsymbol{\mathcal{P}}}$ denotes the power set (the set of all finite subsets) of $\boldsymbol{\mathcal{P}}$.

Properties are the deterministic logical results generated either by the Local Verification Function or the Global Verification Function. Therefore, the property state at a given terminal level n is defined by extracting the logical outcome (via the projection operator π_2) from the local verification:

$P_i^{(n)} = \pi_2(V_{Local}(s_i^{(n)}))$ **(20)**

At higher aggregation levels, the Global Verification Function V_G does not generate a new property set through a second projection. Rather, it evaluates relational coherence, invariant satisfaction, and global admissibility, returning a binary veto outcome.

Properties are strictly generated by the Local Verification Function applied to the terminal Structured Element. They are not limited merely to the binary definition of *Validity*, but encompass all other systemic logical conditions that emerge from the specific application context. For instance, a structured

element s_i could be evaluated as simultaneously *Valid*, but also *Blocked* and *Non-Transferable*.

The formal definition of admissible properties depends entirely on the specific implementation and application domain of the System. Therefore, this document does not aim to provide an exhaustive enumeration of all possible properties. The final implementation will supply a complete description of the admissible property space, tailored to the nature of the target Object.

However, a foundational principle must be emphasized: **a Property is never intrinsically given or self-declared by the data**. It must strictly emerge as the deterministic outcome of the Verification Function applied to the State of the Structured Element.

The subsequent section will further clarify the operational role of the Verification Function within the System architecture.

5.6 Obs_i

Let Obs_i denote the set of externally reported observations, measurements, attestations, declarations, certificates, signatures, testimonials, or records submitted by the External Verification Agent within the verification process.

Recall that the Structured Element s_i intrinsically contains the observational parameter Obs_i, alongside its identifier, data payload, and attributes: (Id_i, D_i, At_i, Obs_i).

Obs_i represents the empirical evidence provided by the External Verification Agent. It contributes to the evaluation, but does not exclusively determine it, since it originates externally to the System — even if produced under System-guided constraints. Obs_i is therefore an input provided from the outside, which is subsequently evaluated through the local application of the Verification Function V_{Local}.

The Verifier records the data corresponding to the specific verification assignment randomly allocated to it. The reported data refers exclusively to the attribute class At_i assigned to the verifier at the terminal level of decomposition n.

The data entered into the System is compared through V_{Local} to verify its convergence with the corresponding information preserved in D_i and its compliance with the logical constraints imposed by the System. For example, if the assigned task involves measuring a physical quantity, the observational evidence recorded in Obs_i is compared with the corresponding value originally encoded in D_i.

For quantitative attributes, the System calculates the magnitude of the detected deviation and checks whether it remains within the admissible tolerance threshold associated with At_i. For qualitative attributes, the System evaluates whether the reported observation is compatible with the equivalence or admissibility relation defined for At_i.

If the convergence criterion associated with At_i is satisfied, the Boolean component of V_{Local} returns 1.

Otherwise, the Local Verification fails and the Boolean component of V_{Local} returns 0.

In summary:

$$Obs_i^{(n)} \rightarrow V_{Local}(s_i^{(n)}) \rightarrow P_i^{(n)} \rightarrow V_{Global}^{(n-1)}(S^{(n-1)}(t)) \rightarrow \dots \rightarrow V_{Global}^{(0)}(S^{(0)}(t)) \rightarrow V_{Global}(S(t)). \quad \textbf{(21)}$$

5.7 Local and Global Validity

The Verification Functions V_{Local} and V_{Global} are applied at different structural levels of the System.
This produces two distinct verification domains: Local and Global.

$Local_{Validity}$

Local Validity is a Logical Property referred to the Single Structured Element, such that

$V_{Local}(s_i^{(n)}(t))$

where

the Structured Element is individually verified at time t by the V_{Local} Function, which evaluates its local logical admissibility, Attributes, convergence between Obs_i and D_i, and Local and Logical Consistency exclusively at the terminal level, where
n represents the final level of recursive decomposition.

$Global_{Validity}$

Global Validity is a logical property that emerges from the consistency of the application of all verification functions for each individual element, starting from the level preceding the terminal one and recursively ascending, involving all the elements of the graph up to the first node and the complete reconstruction of I_G. This constitutes the consequent application of the verification of the relationships between these and the observance of the invariants C at all levels such that:
Let

The V_{Global} function is then applied successively to all intermediate states up to the Node, such that

$V_{Global}(k)(S^{(k)}(t))$,
for each k = n - 1, n - 2, n - 3, ……, 1
The cycle repeats until $E^{(0)}(t)$ of the Node is recompacted.

Once state S is reconstructed, we will have:

$V_{Global}(S(t))$

Verification function applied to the entire state S at time t. This last application checks the Relations R, the Invariants C, and the Consistency between the

Structured Elements already verified by VLocal at the terminal level and by the same V_{Global} function applied recursively until the entire state S is reconstituted.

Therefore, it is possible that

$\forall s_i \in Ter(t), \pi_1(V_L(s_i)) = 1,$ **(22)**

while

$V_G(S(t)) = 0,$

if the global State violates one or more invariants in C or fails relational consistency.

Therefore

$V_{Local} \neq V_{Global}$

These are two Verification Functions applied to two different levels of the Structure, and therefore respond to different satisfaction criteria for successful completion of the inspection.
The relationship between the two, therefore,
is not

$V_{Global}(S) = \wedge_{si \in E^{\wedge}(n)(t)} V_{Local}(s_i),$

but is

$V_{Global}(S) \neq \wedge_{si \in E^{\wedge}(n)(t)} V_{Local}(s_i)$ **(23)**

That is, the verification function applied to the entire (global) State or recursively to the Elements of the intermediate epistemological levels is not equivalent to the verification function applied to the structured element at its terminal (local) level, but we will have

$V_{Global}(S) = f(E, R, C).$

In conclusion,

global validity cannot be reduced to the conjunction of local validity conditions, since the global verification process also depends on relational coherence and the satisfaction of the invariants applied to all levels recursively reconstituted from the previous level to the last terminal level until obtaining the State, such that:

local ≠ global domain.

This evidence leads to an interesting observation:
If an inconsistency occurs during the checks performed by the V_{Global} function on the graph elements reconstructed in ascending order along the intermediate epistemic levels, the reconstruction process is interrupted due to an invariant violation, and the state is rejected even before the reconstruction is complete (early rejection).

5.8 V_{Global} and V_{Local} Verification Functions

Given the Structured Element definition:

$s_i = \langle \sigma_i, (id_i, d_i, a_i, obs_i, p_i) \rangle \in$,

the V_{Local} verification function is applied exclusively to terminal structured elements belonging to the last fragmentation level n.

Thus, the Local Verification process is expressed as:

$V_{Local}(s_i^{(n)}(t))$,

where n is the terminal fragmentation level.

$$V_{Local}(s_i^{(n)}) = (b_i, P_i), \quad \textbf{(24)}$$

with

$b_i \in \{0, 1\}$

and

$P_i \in \mathcal{P}(P)$.

The terminal fragment is locally accepted if and only if:

$\pi_1(V_{Local}(s_i^{(n)})) = 1$.

The Function verifies the validity conditions of all parameters included in s_i, i.e., id_i, D_i, At_i, Obs_i, generating the property P_i.

In detail, each of these parameters is verified for admissibility, internal consistency, plausibility of observational assertions formulated by the external verification agent or its measurements, internal logic of the assertions, and correspondence of the expected attributes At_i with those identified by the User when creating the object existence verification instance (see paragraph 5.8.1 Local Data Convergence Function).

For example, it may happen that the parameters D_i, At_i, and Obs_i are all correct and perfectly aligned (e.g., At_i = "shirt color", D_i = "red", Obs_i = "red"), but the System detects that the value of Id_i (unique identifier) is not possible because it is expressed in an inadmissible formulation.

Or it may be that the parameters idi, At_i, and Obs_i are correct, but the value of D_i is incorrect because it exceeds the maximum expected value for the terminal decomposition level n considered (if the sum of all D_i values at terminal decomposition level 3 reached a value D_i = 101, and the sum of all expected D_i values for the previous decomposition level 2 was equal to 100, a structural inconsistency necessarily arises, either because the decomposition violates information-preserving constraints, or simply because we have an Obs_i statement that does not coincide with D_i).

In this scenario, even if a colluding subset of external verification agents submits observations that satisfy the local convergence criterion, thereby producing an apparent local acceptance, the structural anomaly will emerge

during the recursive aggregation phases (for example, the sum returning 101 instead of 100).

Consequently, the application of the Global Verification Function to the corresponding candidate partial state,

$V_{Global}^{(k)}(S^{(k)}(t))$,

will evaluate to FALSE whenever the reconstructed state violates relational consistency or one of the invariants in C.

The state is therefore rejected despite the apparent correctness of isolated local verification outcomes.

Formally,

let

$V_{Global} : S \rightarrow \{TRUE, FALSE\}$

where S denotes the space of admissible States, such that the State of the System at time t is defined strictly by its structural constituents:

$S(t) = \langle E(t), R(t), C \rangle$,

To capture the recursive nature of the intermediate structural consistency, we define $k \in \{0, 1, …., n\}$ as the epistemic level of decomposition, where k = 0 represents the root State (I_G) and k = n represents the terminal level.

Then we have $V_{Global}(S(t)) = TRUE$ if and only if all the systemic conditions are recursively satisfied across all levels. This deterministic function evaluates whether a State satisfies the System constraints, i.e.:

1. Terminal Level Constraint (k = n): $\forall s_i^{(n)} \in E^{(n)}(t), s_i^{(n)} = \pi_{si}(s_i^{(n)}), V_{Local}(s_i^{(n)}) = TRUE$

2. Intermediate Relational Consistency: $\forall k \in \{0, \ldots, n - 1\}$, the sub-graph $R^{(k)}(t)$ is consistent
3. Intermediate Invariant Satisfaction: $\forall k \in \{0, \ldots, n - 1\}$, $\forall c_j \in C^{(k)}$, $c_j(S^{(k)}(t)) = TRUE$

Therefore, the fundamental equation of Global Validity (28) is formalized as:

$V_{Global}(S(t)) = TRUE \Leftrightarrow$

$[\forall s_i^{(n)} \in E^{(n)},\ s_i^{(n)} = \pi_{s\_i}(s_i^{(n)}),\ V_{Local}(s_i^{(n)}) = TRUE] \wedge [\forall k \in \{0, 1, \ldots, n - 1\}, V_{Global}^{(k)}(S^{(k)}(t)) = TRUE],$ **(25)**

with

$V_{Global}^{(k)}(S^{(k)}(t)) = TRUE \Leftrightarrow$ Consistent$(R^{(k)}(t)) \wedge \forall c_j \in C^{(k)}, c_j(S^{(k)}(t)) = TRUE.$ **(26)**

The outcome of the Verification Function determines Local or Global Validity depending on the structural domain to which it is applied.

In summary, V_{Local} and V_{Global} are not two applications of the same verification criterion at different scales. They are distinct verification functions.

V_{Local} evaluates the empirical convergence between the internally declared data D the and the externally observed Obs the and is applied exclusively at the terminal decomposition level.

V_{Global}, on the other hand, is a system-level internal verification function that evaluates relational consistency, invariant satisfaction, and the consistency of the reconstruction across intermediate levels and on the final state.
No external verification agent participates in the recursive application of V_{Global}.

The Verification Functions V_{Global} and V_{Local} are always deterministic, because identical States necessarily produce identical verification outcomes.
Since the $V_{Global\ or}\ V_{Local}$ Function is defined for all States, we have

$\forall\ S(t) \in S,\ V_{Global}(S(t)) \in \{TRUE, FALSE\}$

Furthermore,

$E_{valid}(t) \subseteq \{ s \mid V_{Local}(s) = TRUE\}$

we can mathematically assert that Global Validity is not merely the conjunction of Local Validities (29):

$V_{Global}(S(t)) \neq \wedge_{s\_i\hat{}(n) \in E\hat{}(n)\,(t)} V_{Local}(s_i^{(n)}(t))$ **(27)**

Brief summary of the operational flow

$O_x \rightarrow I(O_x) \rightarrow \tilde{I}(O_x) \rightarrow E^{(n)}(t) \rightarrow s_i^{(n)} \rightarrow V_{Local} \rightarrow P_i^{(n)} \rightarrow$ Recursive $V_{Global}^{(K)}(S^{(k)}) \rightarrow V_{Global}(S)$ **(28)**

5.8.1 Local Data Convergence Function

In coordinatized operational form, for a terminal structured element

$s_i^{(n)} = \langle \sigma_i, (id_i, d_i, a_i, obs_i, p_i) \rangle$,

the Local Verification Function may be written as:

$V_{Local}(Id_i, D_i, At_i, Obs_i) = (b_i, P_i)$ **(29)**

where this notation denotes the application of V_{Local} to the corresponding terminal structured element $s_i^{(n)}$.

V_{Local}'s specific task is to evaluate convergence, such that

$Conv_{At\_i}(D_i, Obs_i)$

that is, the comparison between the data declared by the user D_i and the data observed by the External Verification Agent Obs_i, with respect to the specific attribute At_i.

The Local Verification Function does not merely check the formal correctness of the structured element.

Its main task is to evaluate the convergence of the data between the value declared by the user, D_i, and the value observed and reported independently

by the external verification agent Obs_i, with respect to the specific attribute class At_i.

Formally,

Let

$$Conv_{At\_i} : D \times Obs \rightarrow \{0, 1\} \quad \textbf{(30)}$$

be the Attribute-Dependent Convergence Function.

For every pair (D_i, Obs_i), the function returns 1 if the value declared by the User and the value independently observed by the External Verification Agent are compatible under the admissibility criteria associated with the attribute At_i, and 0 otherwise.

Therefore

$Conv_{At\_i}(D_i, Obs_i) = 1$

if and only if the value reported by the External Verification Agent is compatible with the value declared by the User under the admissibility criterion associated with At_i

Therefore:

$V_{Local}(Id_i, D_i, At_i, Obs_i) = (b_i, P_i)$

with

$b_i = Conv_{At\_i}(D_i, Obs_i) \wedge Form(Id_i, D_i, At_i, Obs_i)$

where,

Form denotes the formal admissibility check of the tuple.

For quantitative attributes, convergence may be expressed as:

$$Conv_{At\_i}(D_i, Obs_i) = 1 \iff |D_i - Obs_i| \leq \varepsilon_{At\_i} \quad \textbf{(31)}$$

where ε_{At_i} is the admissible tolerance threshold associated with the attribute At_i

For qualitative attributes, convergence may be expressed through an attribute-specific equivalence relation:

$Conv_{At\_i}(D_i, Obs_i) = 1 \iff D_i \equiv_{At\_i} Obs_i$

Thus, Local Validity is not produced by unilateral declaration, nor by external attestation alone, but by the convergence between the User's declared data and the External Verification Agent's observed data under the semantic constraint imposed by the assigned attribute.

Finally

Let

$$Form : Id \times D \times At \times Obs \rightarrow \{0, 1\} \quad \textbf{(32)}$$

be the Formal Admissibility Function.

Form evaluates the syntactic correctness, structural consistency, and implementation-specific admissibility of the tuple (Id_i, D_i, At_i, Obs_i).

The Boolean outcome of the Local Verification Function is defined as

$b_i = Conv_{At\_i}(D_i, Obs_i) \wedge Form(Id_i, D_i, At_i, Obs_i)$.

To recap:

$V_{Local}(s_i^{(n)}) = (b_i, P_i)$,

with

$b_i = Conv_{At\_i}(D_i, Obs_i) \wedge Form(Id_i, D_i, At_i, Obs_i)$.

For intermediate levels:

$V_{Global}^{(k)}(S^{(k)}(t)) = TRUE \Leftrightarrow Consistent(R^{(k)}(t)) \wedge \forall c_j \in C^{(k)}, c_j(S^{(k)}(t)) = TRUE.$

For the complete state:

$V_{Global}(S(t)) = TRUE$

if and only if terminal local admissibility, recursive relational consistency, and invariant satisfaction hold across all admissible decomposition levels.

Therefore:

$V_{Global}(S(t)) \neq \wedge_{s\_i^{(n)} \in E^{(n)}(t)} V_{Local}(s_i^{(n)}(t)).$ **(33)**

Fault Tolerance, Verification Refresh, and Asymmetric Verifier Response

In practical implementations, the Algorithmic Immune System of a CNVS must not be interpreted as a rigid punitive mechanism. A verification failure may arise either from malicious behavior or from ordinary infrastructural faults, such as network interruption, hardware malfunction, communication delays, temporary node unavailability, or local measurement error.

For this reason, a practical implementation of a CNVS should structurally distinguish between non-response, formally impossible falsification, and plausible but non-convergent observations. These cases trigger different response protocols.

1. **Non-Response and Temporary Suspension**

If an External Verification Agent does not return the assigned observation Obs_i within the protocol-defined time window, the System cannot deterministically conclude that the event corresponds to an intentional freeze attack. The failure may also result from an involuntary infrastructural problem affecting an otherwise honest verifier.

To preserve liveness without unfairly penalizing honest participants, the unresolved verification task may be detached from the inactive verifier and reassigned to another verifier randomly selected from the active verifier pool.

The inactive verifier is not immediately excluded. Instead, it may be temporarily removed from the assignment process and placed in a suspended or quarantine status. The verifier may be reinstated after a protocol-defined recovery period, provided that the non-response does not become chronic, statistically recurrent, or structurally harmful to the verification process.

2. Formally Impossible or Manifestly Corrupted Responses

A different response applies when the verifier returns data that are structurally impossible, formally inadmissible, or manifestly incompatible with the assigned attribute class and with the admissible domain of the verification task.

In this case, the System may classify the event as active falsification rather than mere inactivity. Examples include observations that violate the formal admissibility predicate Form, values outside the admissible physical or logical domain, or responses incompatible with the structure of the assigned verification task.

However, in this case, the original task should not simply be reassigned in identical form. A direct reassignment of the same task may itself become exploitable, because an adversarial verifier could use the reassignment behavior as a probing or blocking mechanism.

The corrupted response is therefore archived, the verifier may be suspended or permanently excluded from the active verifier pool, and the System triggers a verification refresh. The refresh does not simply repeat the same task. Instead, it regenerates a controlled verification instance over the same attribute class or over a related measurement domain, possibly using a different formulation, a different measurement granularity, different timing, or a partially transformed verification request.

3. Plausible Non-Convergence and Challenge-Refresh

A more delicate case occurs when the verifier does not provide an absurd or formally impossible value, but returns a plausible observation that does not

converge with the internally declared datum D_i, slightly exceeding the admissible tolerance threshold ε_{At_i}.

In this case, immediate rejection may be too rigid. A single malicious verifier could intentionally submit a plausible but non-convergent value in order to cause the rejection of an otherwise valid candidate state. Conversely, a genuine measurement discrepancy may indicate that the candidate state should not be admitted.

To distinguish these cases, the System may introduce a challenge-refresh zone. If the deviation from D_i exceeds the ordinary acceptance threshold but does not reach the level of formal impossibility or manifest falsification, the System neither admits nor immediately rejects the candidate state. Instead, it places the corresponding local verification condition into a disputed status and activates one or more independent refreshed verification tasks.

These refreshed tasks should not be perfectly identical repetitions of the original assignment. They may refer to the same attribute class, but with a modified measurement formulation, randomized timing, different verifier selection, or partially orthogonal observational questions. This prevents the verification refresh itself from becoming an attack surface.

The refreshed observations are then evaluated as an evidence bundle. If the independent refreshed observations converge back toward the internally declared value D_i within the admissible tolerance threshold, the Local Verification Function may return a positive outcome despite the earlier non-convergent observation. If, instead, independent refreshed observations continue to exceed the tolerance threshold in a coherent manner, the System may reject the candidate state.

4. Cross-Verification Over Time

In the Golden Protocol Nexus implementation, the same object, the same attribute class, or the same verification domain may be inspected multiple times by independent verifiers across successive time windows. For example, a

disputed measurement may be re-examined after 4, 6, or 12 hours by different verification agents.

In this way, truth does not depend on the isolated action of a single verifier. It emerges from the controlled convergence of multiple independent observations, collected through randomized assignment, restricted knowledge, and refreshed verification conditions.

The System may therefore treat a single plausible but non-convergent observation as insufficient for final rejection when the deviation falls within a challenge-refresh interval. Final rejection requires persistent non-convergence across independent refreshed observations, or a violation of formal admissibility, relational consistency, or global invariants.

This asymmetric response architecture makes the CNVS tolerant toward involuntary physical or infrastructural failures, resistant to isolated malicious non-convergence, and strict against the intentional injection of corrupted verification data. It preserves liveness without weakening the Global Veto Principle, because the candidate state is not admitted until Local Verification, refreshed evidence when required, intermediate Global Verification, and final Global Verification are all satisfied.

5.9 Protocol Parameters (ϵ_G, ϵ_N, ϵ_C, ϵ_{At_i}, λ, I_0)

"Protocol Parameters" are defined as the set of critical coefficients (ϵ_G, ϵ_N, ϵ_C, ϵ_{At_i}, λ, I_0) established by the System to regulate the Information Density and Fragmentation Tolerance Thresholds. These parameters are not independent variables, but structural constraints that determine admissibility and rejection conditions across decomposition and verification processes.

These parameters can be modified by the System or the Implementation to vary the expected thresholds and thus adjust the System Efficiency.
They will be discussed in the relevant chapters.

6. The State Transition Law

A state transition is defined as the evolution of the state from time t to time t + 1, such that

$$S(t) \rightarrow S(t + 1) \quad \textbf{(34)}$$

Let

$$T : S \times U \rightarrow 2^S$$

Given the Transition Function T, we have that
$S(t + 1) \in T(S(t), U(t))$
where
$U(t)$ = external inputs (observations Obs_i, validations, events, etc.),

The System generates a candidate state at time t + 1, but there are no invalid intermediate configurations that persistently alter the System. Only configurations that satisfy all the conditions imposed on their constituent elements are admitted.

Specifically, the candidate set of structured elements E(t+1) within the new state must exclusively comprise verified terminal fragments:

$$E_{valid}^{(n)}(t + 1) = \{s_i^{(n)}(t + 1) \in E_{temp}^{(n)}(t + 1) \mid \pi_1(V_{Local}(s_i^{(n)}(t+1))) = 1\} \quad \textbf{(35)}$$

However, satisfying local validity is a necessary but insufficient condition for the State Transition to be finalized. Furthermore, the candidate state S(t + 1)

must also pass the recursive Global Validity checks across its entire structural hierarchy.

Therefore, for the transition to be actualized, the following systemic condition must hold:

$V_{Global}(S(t + 1)) = TRUE.$

That is, assuming S' is the candidate state proposed by the transition function: $S' \in T(S(t), U(t))$

it must satisfy that

$V_{Global}(S') = TRUE.$

The Transition is therefore not free, but is constrained by verification.

In short, State Transitions are governed by the Verification Functions, ensuring that only terminal structured elements satisfying $\pi_1(V_{Local}(s_i^{(n)})) = 1$, all intermediate partial states $S^{(k)}$ satisfying the recursive Global Verification conditions, and the final candidate state S' satisfying $V_{Global}(S') = TRUE$ are mathematically admitted into the Global System at each discrete temporal step.

---

7. Definition of Levels of Knowledge and Information

In a CNVS system, accessible information is distributed according to the decomposition set by the system itself, so that at each subsequent level, the size of the new structured sub-element obtained is reduced, according to a recursive model that preserves the information structure, which is progressively made more fragmented.
This forces us to define two distinct informational domains, which in this section we prefer to call "planes", because the term "level" is reserved for the depth of the recursive decomposition:
The first plane, defined by the "$\mathbb{I}$" notation, refers to the complete structure of the Information Plane and will include, in addition to the semantic content of each "informational unit", all the information related to its reconstruction, its size, and its semantic and structural relationship with all the other "informational units" of the same level, those preceding it, and/or those following it.
A second plane, defined by the "I" notation, refers only to the semantic content of the payload and therefore ignores the structure of the graph that produced it.
This distinction between planes is fundamental to understanding the entire CNVS system.

Given that the system does not generate the Object O_x, nor its $I(O_x)$ information, but merely manages it after it has been entered by the User, each $I(O_x)$ therefore constitutes an input to an external request that uses the System as an accreditation device.
The latter determines whether this input request should be accepted or rejected.
The system will therefore receive a continuous flow of I(Ox) information, in the form of external inputs, which will inevitably increase the content of the System's Global Information, i.e., $\mathbb{I}x$ at time t, because both the semantic

content of the payload and the complexity of its graph will increase.
Following this logic, the verifier's knowledge can develop at different levels of the two possible information planes, with the same recursion.

Let's recall the Chain of Events:
$O_x \rightarrow I(O_x) \rightarrow \tilde{I}(O_x) \rightarrow E \rightarrow s \rightarrow \mathtt{I}(s)$.

It is important to note that within the System, information relating to both the graph and the payload content is not transmitted in directly readable form, but is encoded at source to make it cryptic.

The encoding is:
- Systemic
- Uniform
- Structural.

Therefore, the decomposed information will not reach the verifier in directly readable form, but must first be decrypted, as it will have inherited the encoding of the original information.

The System provides structured encoding as an additional security mechanism and, in ideal implementations, encryption may additionally be applied.

In an ideal model, decryption should be possible using unique keys provided exclusively to the verifier assigned to the specific execution fragment.

This way, even if an unauthorized Verifier were to access other Fragments, thus violating the System's exact distribution map, they would be unable to read their contents, as they do not possess the specific decryption keys for each Fragment.

In short, in the System the information is encoded and, in the ideal implementation, encrypted, so the Verifier's access to all the Fragments does not eliminate the encryption problem.

$Dec(\tilde{I}_x(t)) \neq I_x(t)$

for an unauthorized Local Verifier,
while the Information Encoding $I_x(t)$ can be expressed as $\tilde{I}_x(t)$ at time t, such that
$\tilde{I}_x(t) = Enc\ (I_x(t))$.

For simplicity, the Model is formalized below by considering only the Final Encoding (Enc) notation = I_x without reporting the notation $\tilde{I}_x$ for the different Information levels and planes.

In light of this, the semantic knowledge K_x of an external User at time t, limited to the semantic content level present in the System at level x, can be understood as an accessible set of information, such that

$$K_x(t) \preceq I_x(t) \quad \textbf{(36)}$$

Specifically, it is a subset of the information accessible to a subject, where

- $I_x(t)$ = Information existing as semantic content,
- $K_x(t)$ = knowledge accessible,

for each domain x, at time t.

The knowledge K_x of a User at time t extended to the global information level present in the System at level x, can be understood as the accessible set of information, such that

$K_{\mathbb{I}x}(t) \preceq \mathbb{I}_x(t)$ **(37)**

where

$\mathbb{I}_x(t)$ corresponds to the semantic content of the payload, of its entire graph including the relational structure and the metadata necessary for reconstruction.

The entire representation is always expressed within the system as an encoded and ordered binary sequence {0, 1}, both for I and for $\mathbb{I}$, such that

$I_x(t), \mathbb{I}_x(t), K_{\mathbb{I}x}(t), K_x(t) \in (\{0, 1\}^*)^z$ **(38)**

where:

- $\{0, 1\}^*$ is the set of all finite binary strings;
- x identifies the domain (Global, nodal, fragmented, Local);
- $z, x \in \mathbb{N}$

Let $\mathbb{I}_G(t)$ be the Global Information of the System at time t, such that

$\mathbb{I}_G(t) \cong (E(t), R(t), C, \mathbf{V})$ **(39)**

where $\mathbf{V} = \{V_{Local}, V_{Global}\}$

$\mathbb{I}_G$ will comprise the complete graph structure of E(t) and the exact semantic content of each of its recursive decompositions, at every level along the entire graph, up to the complete payload.

Let

$K_{\mathbb{I}G}(t)$,

be the global knowledge accessible at time t, that is, the knowledge relating to the information $\mathbb{I}$ of the entire system at time t, from the payload to the

complete graph structure of domain G,
such that

$\mathbb{I}_G(t) \cong \mathbb{I}(S(t))$

where $\mathbb{I}_G$ corresponds to the information:
- of the entire structured set E(t),
- of all the relations R(t)
- of all the constraints (C) imposed by the system,
we will therefore have

$K_{IG}(t) \preceq \mathbb{I}_G(t).$ **(40)**

The same relationship is found for the purely semantic information plane I, referring exclusively to the transported payload, devoid of its relations, invariants, and the complete structure of its graph. Therefore, we have symmetrically:
$I_G(t) = I(E(t))$, but not
$I_G(t) \neq I(S(t))$, because the State also includes Invariants and Relations.
Therefore,
the overall semantic knowledge will be:

$K_G(t) \preceq I_G(t).$ **(41)**

The nodal knowledge K_{IN} at time t refers to the information I_N relating to the first node at time t, and therefore to all subsequent recursive decompositions that derive from it up to all terminal fragments, and includes both their semantic content and the structure of the entire descendant graph generated from that node,
where N_i constitutes the first node, so we will have

$\mathbb{I}_N(t) \cong (\bigcup^n{}_{h=0} E_h^{(Ni)}(t), \bigcup^n{}_{h=0} R_h^{(Ni)}(t), C^{Ni}, \mathbf{V}^{Ni})$ **(42)**

and that is

$\mathbb{I}_N(t) \cong \mathbb{I}(\text{Desc}(N_i, t))$ **(43)**

where

$\text{Desc}(N_i, t)$

denotes
the recursive set of all descendants of node N_i.

Then the relation will be:

$K_{IN}(t) \preceq \mathbb{I}_N(t)$. **(44)**

Symmetrically, the semantic Nodal knowledge will be

$I_N(t) \cong I(\text{Desc}(Ni, t))$, and refers to the semantic content of all payloads derived from the recursive decomposition of the common node N_i.

The decomposition-level knowledge $K_{Ih}^{(n)}$ at time t is that relating to the information $\mathbb{I}_h^{(n)}$ of all the decomposed elements that are located at the same level h at time t, but excludes the recomposed elements of the previous levels and the decomposed elements of the subsequent levels, with reference to both the aggregated local semantic content and the local structure, including the related Relations, invariants and function $\mathbf{V}$,
such that

$\mathbb{I}_h^{(n)}(t) \cong (E(t)^h, R(t)^h, C^h, \mathbf{V}^h)$, **(45)**

$K_{Ih}(t) \preceq \mathbb{I}^{(h)}(t)$.

The semantic level knowledge $K_h(t)$ will be that referred to the aggregate semantic content plan of the payload of all the elements belonging to the same level h, excluding the graph and the structure of the Relations, such that

$K_h^{(n)}(t) \preceq I_h^{(n)}(t)$ **(46)**

The fragment-level knowledge K_{IF} at the local level includes the knowledge relating to the semantic information I_F of the terminal fragment, including all

the Relations, invariants and the structure of the Fragment itself,
such that

$$\mathbb{I}_F(t) \cong (s_{ij}^{(n)}(t), R_{s\_ij}(t), C_{s\_ij}(t), V_{Local}(s_{ij}, t)) \quad \textbf{(47)}$$

therefore

$$K_{\mathbb{I}F}(t) \preceq \mathbb{I}_F(t). \quad \textbf{(48)}$$

Symmetrically, the semantic knowledge $K_F(t)$ at time t, referring only to the elementary content of the non-aggregated payload, and without its graph and relations will be

$$K_F(t) \preceq I_F(t) \quad \textbf{(49)}$$

for $I_F(t) \cong I(s_{ij}^{(n)}(t))$. **(50)**.

The fragment-level knowledge K_F of the single fragment of the last level assigned by the system at time t is also called local knowledge (K_{Local}), such that

$$K_F(t) = K_{Local}(t), \quad \textbf{(51)}$$

$$K_F(t) \preceq I_F(t) \quad \textbf{(52)}$$

This last definition is particularly important in CNVS Systems, because fragment-level knowledge presupposes a level of knowledge that is actually superior to that of the final verifier.
If everything described in the previous paragraphs is true, the verifier will not need to know the Relations, invariants, or the activity of the V_{Local} function on the assigned fragment, so its level of information will certainly be lower than $K_{\mathbb{I}F}$, but it will also be lower than that of K_F.

The reason lies in the fact that the verifier does not even have complete semantic knowledge of the assigned fragment, as the System turns to it to entrust it with the task of performing an observation on the Object, but does not reveal the contents of the payload recorded in D_i, and does not know the definition of the specific class of the Attribute A_{ti}, but can only infer it.

It is the verifier who must record the observed data, generating the input Obs_i that is subsequently inserted into the System.
This is why the relation holds:

$$K_{VF}(t) \lneqq I_F(t) \quad \textbf{(53)}$$

The accessible information levels will have the following relationships:
$I_F(t) \subsetneq I_h(n)(t) \subsetneq I_N(t) \subsetneq I_G(t)$. **(54)**

Considering only the semantic level, we observe that the distribution of the I_F information relative to the fragments obtained $s_i^{(n)}$ does not follow a uniform pattern,
formally we will have

$$D_{nu}(I_h^{(n)}(t), F_x(t)) \rightarrow \{I_{F,1}(t), I_{F,2}(t), \ldots, I_{F,Fx(t)}(t)\} \quad \textbf{(55)}$$

where
- D_{nu}, is the nonuniform distribution function;
- $I_h^{(n)}(t)$, is the information of level n to be fragmented at time t;
- $F_x(t)$, is the number of terminal fragments required at time t;
- $I_{F,j}(t)$, is the information of fragment j.

The information distributed by fragmentation $I_h^{(n)}(t)$ is completely preserved, formally

$$\Sigma_{j=1}^{F\_x(t)} |I_{F,j}(t)| = |I_h^{(n)}(t)| \quad \textbf{(56)}$$

where we have
$\exists j \neq k$ such that $I_{F,j}(t) \neq I_{F,k}(t)$.

Let $X_j(t)$
be the random variable representing the terminal fragment assigned to verify v_j at time t,

Where
- j identifies the verifier V_j;
- $X_j(t)$ is the fragment assigned to verifier V_j at time t,

let

$E^{(n)}(t)$ be the set of Fragments of level n at time t,

let

Ter(t) be the set of terminal Fragments of $E^{(n)}(t)$ of level n at time t, such that
$Ter(t) = E^{(n)}(t)$,
where n denotes the terminal decomposition level at time t,
then formally we have
$X_j(t) \in Ter(t)$

and therefore
$K_{VF,\,j}(t) = \mathcal{J}\,(X_j(t))$,
where $\mathcal{J}$ represents the return function of the information content of the terminal fragment, such that

$\mathcal{J} : Ter(t) \rightarrow \{0, 1\}^*$.

If the fragment is encrypted and the verifier has the correct key,

$K_{VF,\,j}(t) = Dec\,(\overline{\mathcal{J}}\,(X_j(t)))$,

where $\overline{\mathcal{J}}\,(X_j(t))$ denotes the encoded representation of the information associated with the assigned fragment,

and when decryption is complete, we have
$K_{VF,j}(t) = \mathcal{J}(X_j(t))$.

The verifier does not receive the full semantic payload $I_{F,j}(t)$ of the assigned terminal fragment. Even when encryption is used and the verifier possesses the correct key, decryption gives access only to the operational view required to perform the assigned observation.

Let $View_{VF}(X_j(t))$ denote the verifier-visible operational view associated with the assigned terminal fragment $X_j(t)$.

Then:

$K_{VF,j}(t) = View_{VF}(X_j(t))$,

with

$K_{VF,j}(t) \subsetneqq I_{F,j}(t)$.

Even when encryption is used and the verifier possesses the correct key, decryption gives access only to the operational view required to perform the assigned observation, not to the full semantic payload of the assigned terminal fragment.

Therfore:

The K_{VF} Knowledge available to Verifier v_j at time t will be limited to its own fragment but cannot be complete, because the External Verification Agent does not have full access to the I_{F_i} Information, but only to a measurement grid for managing the assigned task.
Specifically, the System manages the amount of fragmented information to

share, based on "how much" information the external verification agent needs to know to perform data collection and its objective assessments, which will then be used by the System for verification through convergence of the D_i Data.
Therefore

$$K_{VF}(v_j, t) \lneqq K_F(t) \preceq I_F(t) = I(s_{ij}^{(n)}). \quad \textbf{(57)}$$

On the fragmented Knowledge plane, however, there is no access even to the Fragments that make up the Same Level.

Therefore, we will have

$$K_{VF}(t) \lneqq I_N(t), \quad \textbf{(58)}$$

but

$$K_{VF}(t) \neq I_N(t)$$

and

$$K_{VF}(v_j, t) \lneqq K_F(t) \subsetneq K_N(t). \quad \textbf{(59)}$$

Furthermore,

$$K_N(t) \subsetneq K_G(t). \quad \textbf{(60)}$$

The Knowledge Relationships will therefore be

$$K_{VF}(v_j, t) \lneqq K_F(t) \subsetneq K_h^{(n)}(t) \subsetneq K_N(t) \subsetneq K_G(t) \quad \textbf{(61)}$$

It should be noted that Fragment Assignment follows a random logic, such that a group of External Verifiers will not necessarily receive Fragments belonging to the same Node or to the same decomposition branch. External Verifiers may be assigned terminal Fragments derived from different Nodes.

That is,

$$\text{Assign}(F_{ij}^{(d)}) \rightarrow v_j \qquad \textbf{(62)}$$

where $F_{ij}^{(d)}$ is a terminal fragment of Node N_i, but Verifier v_j can receive Fragments coming from different Nodes:

$$K_{VF}(t) \subsetneqq \{I_F(N_i, t), I_F(N_l, t), ...\}$$

with

$$i \neq l$$

This increases information dispersion, because the Verifier does not work within a recognizable "nodal class."

Furthermore, since the information is encoded, we will have

$$\tilde{I}_x(t) = \text{Enc}(I_x(t))$$

for all subsequent levels of Information up to

$$\tilde{I}_F(t) = \text{Enc}(I_F(t)). \qquad \textbf{(63)}$$

The knowledge of the External Verification Agent will be:

$K_{VF}(t) \lneqq \tilde{I}_F(t)$

with

$\mathrm{Enc}(I_F(t)) = \tilde{I}_F(t).$ **(64)**

if you have the correct encryption key.

Synthesis.

Knowledge Hierarchy Table

Knowledge Level	Meaning	Scope
$K_F(t)$	Fragment-Level Knowledge	Knowledge accessible at the terminal Fragment level
$K_h^{(n)}(t)$	Decomposition-Level Knowledge	Knowledge associated with all Structured Elements belonging to the same decomposition level h
$K_N(t)$	Nodal Knowledge	Knowledge associated with a Node and all its recursive descendants
$K_G(t)$	Global Knowledge	Knowledge associated with the entire System State

---

Semantic Information Hierarchy Table

Information Level	Meaning	Scope
$I_F(t)$	Fragment Semantic Information	Semantic payload of a terminal Fragment
$I_h(n)$	Decomposition-Level Semantic Information	Aggregate semantic payload of all elements at level h
$I_N(t)$	Nodal Semantic Information	Semantic payload recursively derived from Node N_i
$I_G(t)$	Global Semantic Information	Total semantic payload contained in the System

---

Structured Information Plane Table

Structured Information Plane	Meaning	Includes
$\mathtt{I}_F(t)$	Fragment Structured Information	Payload, metadata, relations, invariants, local verification structure
$\mathtt{I}_h^{(n)}(t)$	Level Structured Information	Structured representation of all elements at decomposition level h
$\mathtt{I}_N(t)$	Nodal Structured Information	Recursive descendant graph generated from Node N_i
$\mathtt{I}_G(t)$	Global Structured Information	Entire graph structure of the System State

---

Knowledge–Information Equivalence Table

Knowledge Domain	Corresponding Information Domain
$K_F(t)$	$I_F(t)$
$K_h^{(n)}(t)$	$I_h^{(n)}(t)$
$K_N(t)$	$I_N(t)$
$K_G(t)$	$I_G(t)$

Semantic vs Structured Information Planes

Plane	Notation	Meaning
Semantic Information Plane	I_x	Semantic payload content only
Structured Information Plane	$\mathtt{I}_x$	Semantic payload + graph structure + metadata + relations + invariants

---

Knowledge Accessibility Relation Table

Relation	Meaning
$K_x(t) \preceq I_x(t)$	Accessible semantic knowledge is bounded by semantic information
$K_{\mathtt{I}x}(t) \preceq \mathtt{I}_x$	Accessible structured knowledge is bounded by structured

Relation	Meaning
	information
$K_{VF}(t) \prec I_F(t)$	Verifier knowledge is strictly lower than fragment semantic information

---

Recursive Informational Inclusion Hierarchy

Hierarchy Type	Relation
Semantic Information Hierarchy	$I_F(t) \subset I_h^{(n)}(t) \subset I_N(t) \subset I_G(t)$
Semantic Knowledge Hierarchy	$K_F(t) \subset K_h^{(n)}(t) \subset K_N(t) \subset K_G(t)$
Structured Information Hierarchy	$\mathtt{I}_F(t) \subset \mathtt{I}_h^{(n)}(t) \subset \mathtt{I}_N(t) \subset \mathtt{I}_G(t)$

In short, Knowledge is not a simple quantitative reduction, but a Structural Reduction of the System's Representation, where it is possible to have a small but well-structured set of data, or by violating the System, to obtain much more, but poorly structured and poorly comprehensible data.

In both cases, the available information is incomplete and fragmented to the point that reconstruction by an unauthorized External Agent becomes difficult.

---

8. The Density of Information

Information Density ρ is a measure of the Dispersion of Information at the level in question and can be understood as the weight of the information at the level in question compared to the levels preceding it.

Information Density is represented at multiple levels and does not depend on the Knowledge of the External Verifier, being a measure used by the System.

The Local Information Density ρ_N is given by the measure of the contribution of Fragmental Information to the Node (or Nodal Density), and constitutes the portion of information per single Fragment compared to the Nodal Information.

It is a measure of the Dispersion of Nodal Information in Fragmental Information, such that

$$\rho_{N,j}(t) = |I_{F,j}(t)| / |I_N(t)| \quad \textbf{(65)}$$

ρ_N measures how much the information of that fragment weighs compared to the information of its node (going up through all the levels of decomposition up to $s_i^{(0)}$) or how much the nodal information has been dispersed;

The Global Information Density ρ_G referred to the fragmental information is given by

$$\rho_{G,j}(t) = |I_{F,j}(t)| / |I_G(t)| \quad \textbf{(66)}$$

ρ_G measures the weight of a fragment relative to the Global information of the System.

Similarly, it is an index that measures the dispersion of Global information in the terminal Fragments.

The Relative Nodal Density ρ_{GN} is a measure of the importance of nodal information relative to Global information and measures the weight of nodal information relative to Global information, such that

$$\rho_{GN}(t) = |I_N(t)|/ |I_G(t)|. \quad \textbf{(67)}$$

The Composite Information Density ρ_C is the composite measure obtained by taking into account both the effects of the Global Density and those of the Local Density.
It is defined as a weighted geometric mean of two positive ratios,
such that

$$\rho_{C,j}(t)=(|I_{F,j}(t)/I_N(t)|)^{\alpha} * (|I_{F,j}(t)/I_G(t)|)^{\beta\_\rho}, \quad \textbf{(68)}$$

where α and β_ρ are weighting coefficients that determine the relative contribution of the two information ratios, and that can be modified by the System or its Implementation to modify in the measurement of ρ_C the weight of the relative contributions of the Fragmental Information on the Nodal or Global Information, such that

$\alpha, \beta_\rho \in [0, 1]$ and $\alpha + \beta_\rho = 1$

and where

$|I_F/ I_N|$ measures how much the fragment weighs in relation to the Node.

$|I_F/ I_G|$ measures how much the fragment weighs in relation to System.

So: ρ_C

measures a Composite Information Density, which takes into account both the Fragmental exposure within the Node and the relative importance of the Fragment in the Global System.

System Consistency is guaranteed by compliance with the Protocol Parameters (see relevant section).

The System rejects any state transition in which the detected density exceeds the established threshold coefficients ($\rho_{N,j} > \varepsilon_N$ or $\rho_{G,j} > \varepsilon_G$ or $\rho_{C,j} > \varepsilon_C \implies$ Reject (S)) and violates the Conditions dictated by the Protocol Parameters.

In CNVS Systems, higher informational density corresponds to lower informational dispersion and therefore to greater semantic exposure of the Fragment relative to the protected Information domain,

therefore:

for

$0 < \rho_N(t), \rho_G(t), \rho_{GN}(t) \leq 1$

under admissible decomposition conditions.

---

9. Axioms

9.1 Axiom I

In the System, only what has been verified and admitted by the corresponding Verification Function exists as part of the valid CNVS State.

Anything that has not been verified, or that has been verified but has not obtained the required validity condition, does not exist for the System as an admitted State component. It may exist as an external object, input, claim, or candidate representation, but it is not admitted into the valid System State.

At the terminal level, a structured element $s_i^{(n)}(t)$ may belong to the valid terminal set only if the Boolean component of the Local Verification Function is equal to 1:

$$s_i^{(n)}(t) \in E^{(n)}(t) \Longrightarrow \pi_1(V_{Local}(s_i^{(n)}(t))) = 1. \quad \textbf{(69)}$$

At the global level, a State is valid if and only if the Global Verification Function applied to the State evaluates to TRUE:

$$Valid(S(t)) \Leftrightarrow V_{Global}(S(t)) = TRUE. \quad \textbf{(70)}$$

with

$$S(t) = \langle E(t), R(t), C \rangle.$$

Candidate states S′ produced by the Transition Function are admitted only after terminal Local Verification and subsequent recursive Global Verification.

Thus, if

$S' \in T(S(t), U(t))$,

then S′ becomes an admitted System State only if

$V_{Global}(S') = \text{TRUE}$.

By also applying the Transition Function, the valid terminal elements at time t + 1 are defined as:

$$E_{valid}^{(n)}(t+1) = \{s_i^{(n)}(t+1) \in E_{temp}^{(n)}(t+1) \mid \pi_1(V_{Local}(s_i^{(n)}(t+1))) = 1\}. \quad \textbf{(71)}$$

Consequently, local rejection strictly prohibits admission into the valid terminal set:

$$\pi_1(V_{Local}(s_i^{(n)}(t))) = 0 \Longrightarrow s_i^{(n)}(t) \notin E_{valid}^{(n)}(t). \quad \textbf{(72)}$$

In short, Existence in a Native Verification System is verification-dependent.

A terminal Structured Element becomes part of the valid terminal set only if it satisfies Local Verification, while the corresponding candidate State becomes part of the valid CNVS State only if recursive Global Verification is satisfied.

Invalid elements may exist as external data, candidate inputs, or rejected representations, but they do not constitute the admitted CNVS State.

9.2 Axiom II

The System requires that the State be represented through successive Levels of Structured Decomposition.

Let $\mathfrak{D}$ be the decomposition operator:

$\mathfrak{D} : \boldsymbol{\mathcal{S}} \rightarrow \boldsymbol{\mathcal{P}}_{\text{fin}}(\square)$.

For every structured element $s \in \boldsymbol{\mathcal{S}}$, either $\mathfrak{D}(s) = \emptyset$, in which case s is terminal, or $\mathfrak{D}(s)$ is a finite set of child structured elements generated by recursive decomposition.

For every System state S(t) admitted to the decomposition process, there exists a finite recursive decomposition of depth n. **(73)**

such that:

$\mathfrak{D}^{(n)}(S(t)) = \{s_i^{(n)}(t)\}$

the System imposes that the Decomposition Function $\mathfrak{D}$ at time t increases the informational dispersion of the semantic payload of the Global Information I_G as the latter increases.

Given the Local Density ρ_N, Global Density ρ_G, Composite Density ρ_C,

The System adapts the Decomposition according to a Model that produces an increasing distribution of Information, with a progressive reduction of ρ_N, ρ_G, ρ_C, and therefore of the weight of Local Information over Global Information.

That is, if the nodal information grows, the System is forced to increase the depth of the decomposition for successive levels as the number of Fragments increases, reducing the portion represented by each.

In summary:
The System imposes a Recursive Decomposition Constraint such that for each increase in both the Nodal Information In and the Global Information I_G, the relative share of Information accessible through the terminal Fragments is reduced such that,

$\rho_N(t) \leq \epsilon_N$ and $\rho_G(t) \leq \epsilon_G$

with

$\varepsilon_N, \varepsilon_G \in (0, 1]$

thus

if the terminal fragment size does not increase proportionally with the nodal or global information domain, then an increase in I_N or I_G produces a relative decrease in fragment density:

$I_N(t_2) > I_N(t_1) \Rightarrow \rho_N(t_2) \leq \rho_N(t_1)$,

and

$I_G(t_2) > I_G(t_1) \Rightarrow \rho_G(t_2) \leq \rho_G(t_1)$,

under the condition that the corresponding terminal fragment information does not grow proportionally.

If $\rho_C \uparrow$,

therefore formally

$$\rho_C(N_i, t) > \epsilon_C \Rightarrow \text{Depth}(N_i, t + 1) > \text{Depth}(N_i, t) \quad \textbf{(74)}$$

where ϵ_N, ϵ_G constitute two threshold coefficients, for ρ_N and for ρ_G, t1 and t2 represent two different time instants such that t1 < t2, and

Depth $(N_i, t) \in \mathbb{N}$,

- Depth(N_i, t) is the maximum depth reached by node N_i at time t,
- $N_i = s_i(0)$ is the Node, i.e., the structured Element of nodal level 0, as defined in chapter 5.1;

therefore,

Depth corresponds to the number of levels of the Recursive Decomposition applied by the System at time t in the Node i.

The System imposes that as nodal information increases, the fragmentary information density with respect to the node decreases below a coefficient ϵ_N;

The System imposes that as Global information increases, the fragmentary information density with respect to the Global System decreases below a coefficient ϵ_G.

In summary, the increasing Adaptive Decomposition therefore responds directly to the Composite Density, and indirectly to the Local and Global Densities, according to the following formulation:

The adaptive objective of the System is therefore:

$\text{Depth}(N_i, t) \uparrow \Rightarrow \rho_C(N_i, t) \downarrow$,

provided that the additional decomposition effectively reduces the informational weight of terminal fragments relative to the nodal and global domains.

In fact, it is easy to demonstrate experimentally that the Densities ρ_N, ρ_G, and ρ_C progressively decrease as the System increases its Global and Nodal Information Level, regardless of any active interventions by the System itself.

This happens naturally, because Information Density is by definition a measure of a ratio, where the reference densities I_G and I_N are in the denominator:

In other words, if the size of the Information I_F at time t remains unchanged, while that of the Global Information I_G or Nodal Information I_N increases, each Fragment F containing the Information I_F, remaining of the same size, will have a smaller weight on the reference Information.

The effect will be a reduction in Density and therefore a Dispersion of Information, even without active interventions on the Fragments.
The size of each Terminal Fragment, which corresponds to the Fragmental Information I_F, is determined by the number of Fragments imposed by the System, because the greater the fragmentation depth, the smaller the size of the Terminal Fragments will necessarily be and therefore the smaller the relative contribution of I_F Information.

This behavior is a problem, because instead of forcing the Decomposition to successive levels, it stabilizes it.

after applying the decomposition we will have instead

$$\text{Depth}(N_i, t)\uparrow \Rightarrow \rho_C(t+1) < \rho_C(N_i, t),$$

$$\rho_C(N_i, t) > \epsilon_C \Rightarrow \text{Depth}(N_i, t+1) > \text{Depth}(N_i, t).$$

The Condition "$\rho_C(N_i, t) > \epsilon_C$" then becomes the Internal input to the System which triggers further levels of Decomposition until the following effect is obtained:

$$\text{Depth}(N_i, t+1) > \text{Depth}(N_i, t) \Rightarrow \rho_C(N_i, t+1) < \rho_C(N_i, t).$$

But the Condition "$\rho_C(N_i, t) > \epsilon_C$" is not easily verified experimentally because the growth of the I_F Information for a single Fragment does not follow that of the Information of the Whole System.

This happens because each Fragment remains isolated after decomposition, and while the System continues to expand into new nodes, the Fragments already emitted remain stationary.

Even when the decomposition depth is forced, the effect is an uncontrolled drop in the Local Density level, such that further fragmentation is increasingly difficult.
In Chapter 10, new control coefficients will be introduced to address the problem.

We can therefore conclude that the densities ρ_N, ρ_G, and ρ_C constitute a measurement tool internal to the System, usable in implementation, rather than a set of control parameters.

Let's look at the number of Fragments in detail.

Let

$\text{Fragments} \in \mathbb{N}$,
and recalling that $Ni := s_i^{(0)}$,
with $\text{Fragments}(N_i, t) = \gamma\,(I_N(t), \rho_c(N_i, t))$,

where
γ is a Monotone Function increasing with respect to information complexity,
or rather $\text{Fragments}(N_i, t) = |\{s_i^{(n)}(t)\}|$, that is,
$\text{Fragments}(N_i, t) = |E_i^{(\text{Depth}(Ni,t))}(t)|$
where
$\text{Fragments}(N_i, t)$ corresponds to the number of Terminal Fragments generated by the Adaptive Decomposition process at time t in Node N_i,
with
- $E_i^{(d)}(t)$ which constitutes the set of Fragments of Node N_i at level d;
- $\text{Depth}(N_i, t)$, the maximum depth of Node N_i,
- the modulus $|\cdot|$, the cardinality,
and

Fragments(t) is the total number of terminal Fragments of the System at time t, i.e.
Fragments(t) = Σ_i Fragments(N_i, t),
therefore the sum of the number of terminal Fragments of each Node,

such that

$$|\text{Ter}(t)| = \text{Fragments}(t) \quad \textbf{(75)}$$

therefore,

$$\text{Ter}(t) = \bigcup_i E_i^{(\text{Depth}(N\_i,\, t))}(t) \quad \textbf{(76)}$$

the Decomposition applied to the Node must depend not only on the internal Information Breadth, but also on its relative Information weight with respect to the Global System.
So the level of Decomposition is a direct function of the Composite Density measure, which depends on the Local and Global Density Values, such that

$$\rho_C(N_i, t) \leq \epsilon_C \quad \textbf{(77)}$$

where

$\varepsilon_C \in (0, 1]$

The System must force the Depth of Information Decomposition to maintain Dispersion and reduce Information Density.

This happens when the Composite Density ρc increases, because it means that the Terminal Fragment weighs too much compared to the

node/Global. Therefore, the System is forced to increase the Decomposition Depth to reduce that density and preserve the Structure.

Therefore,

$$\text{Depth}(N_i, t) = f(\rho_C(N_i, t)), \quad \textbf{(78)}$$

with f being a composite Density Function, and

$$\rho_C(N_i, t) = g(I_F(t), I_N(t), I_G(t)) ,$$

where g denotes a monotonically increasing function with respect to the need to reduce the density defined in equation (68),

Under non-trivial additional decomposition, the verifier-accessible knowledge does not increase, and typically decreases:

$$K_{VF,\,new} \preceq K_{VF,\,old},$$

with strict reduction

$$K_{VF,\,new} \precneqq K_{VF,old}$$

whenever the new decomposition strictly reduces the verifier-visible operational view, and

$$K_{VF}(t) \subsetneq K_F(t) \subsetneq K_G(t) \quad \textbf{(79)}$$

The Decomposition grows monotonically with respect to both I_N and I_G.

Since every System state admitted to the decomposition process has a finite decomposition of Depth n,

the Decomposition level requires a sufficient number of External Verifiers to execute the terminal verification tasks generated by the System.

Therefore:

$$|V_{ext}^{required}(t)| \geq |Fragments(t)|. \quad \textbf{(80)}$$

This means that the System requires a sufficient number of verifiers to sustain the imposed level of decomposition $\mathfrak{D}$.

The number of verifiers is therefore an operational feasibility condition of the decomposition process, not the theoretical source of decomposition itself.

In other words, the System does not define decomposition as a function of the currently available verifiers. Rather, it determines the required decomposition level from information-density constraints, and then requires a sufficient verifier set to execute the resulting terminal tasks.

where

- $|V_{ext}^{required}(t)|$, set of Verifiers required at time t
- ρ_c, Composite Density
- ϵ_c, the fragmentation coefficient

This is because Decomposition is an Epistemic Property of the System, not an accidental dependence on the Number of Verifiers available at time t.
In other words, the System does not depend on the number of Verifiers, but manages it and requires it as a Preliminary Condition.

Let's clarify that within the System, the Verifier is not the User.

9.3 Axiom III

The Validity of the System cannot be reduced to the Validity of the Individual Structured Elements that constitute it.

$$\text{Valid}(S(t)) \Leftrightarrow V_{\text{Global}}(S(t)) = \text{TRUE}, \quad \textbf{(81)}$$

with

$$S(t) := \langle E(t), R(t), C \rangle,$$

and

$$V_{\text{Global}}(S(t)) = f(E(t), R(t), C). \quad \textbf{(82)}$$

At the terminal level, assume that all terminal structured elements are locally valid, namely:

$$\forall s_i^{(n)} \in E^{(n)}(t),\ \pi_1(V_{\text{Local}}(s_i^{(n)}(t))) = 1.$$

Even under this condition, it does not follow that the Global State is valid.

Therefore, Global Validity is not equivalent to the conjunction of all Local Validity conditions:

$$V_{Global}(S(t)) \neq \bigwedge_{s\_i \hat{} (n) \in E \hat{} (n)(t)} \pi_1(V_{Local}(s_i^{(n)}(t))). \quad \textbf{(83)}$$

More precisely, Axiom III asserts that there may exist a State S(t) such that:

$$[\forall s_i^{(n)} \in E^{(n)}(t), \pi_1(V_{Local}(s_i^{(n)}(t))) = 1]$$

and yet

$$V_{Global}(S(t)) = \text{FALSE}.$$

This happens because the Validity of the State depends not only on the Local Verification of its terminal elements, but also on the relational structure R and on the invariant constraints C.

Therefore:

$$V_{Global}(S(t)) = \text{TRUE} \Leftrightarrow [\forall s_i^{(n)} \in E^{(n)}(t), \pi_1(V_{Local}(s_i^{(n)}(t))) = 1] \ \wedge$$

$$[\forall k \in \{0, \ldots, n-1\}, \text{Consistent}(R^{(k)}(t)) \wedge (\forall c_j \in C^{(k)}, c_j(S^{(k)}(t)) = \text{TRUE})]. \quad \textbf{(84)}$$

This Axiom is extremely important for ensuring Global Structural Coherence, because without this invariant specification, the System would become structurally inconsistent: successive Local Verification aggregation conditions would be sufficient to obtain a Valid State, without imposing a Global Coherence Check at higher levels.

In other words, the System is structural and does not exhibit merely aggregative properties across different Verification Levels.

Therefore, Validity is an emergent Property, not a simple sum.

R and C are essential for the State Definition.

In summary:

Global Validity cannot be reduced to Local Validity. **(85)**

Verification of the System's State depends not only on the Individual Structured Elements, but also on the Consistency of Relations and compliance with Invariant Constraints.

In conclusion, Axiom III establishes that in a CNVS System, Global Validity must also depend on Relations and Invariants, not just on the Structured Elements.

Therefore:

$V_{Global}(S(t)) \neq \bigwedge_{s\_i^{(n)} \in E^{(n)}(t)} \pi_1(V_{Local}(s_i^{(n)}(t)))$,

because the Verification of a State, on which its Validity depends, is not only the derivation of the aggregation of the Verifications of the Individual Elements that constitute it.

---

10. Information Restriction Principle

Introduction

Since the decomposition depth does not respond directly to the information, but responds to the resulting information density, the relationship is:

$\rho_C(t) \leq \epsilon_C$,

therefore, as Recursive Depth, Information Dispersion, and Adaptive Fragmentation increase, the relative share of Locally accessible Information progressively approaches zero with respect to the Node and the Global System.
In Chapter 7, the problem of the difficulty of managing Fragmentation if controlled directly through Composite Density was raised.
The proposed solution is to control the Decomposition Depth under the reciprocal of ρ_C, using an expected-average control approximation over Nodes and Fragments, without assuming that the actual decomposition is uniform:

$1/ (\rho_C(t) * N(t)^{\beta\_N})$,

where N(t) denotes the number of active internal nodes at time t.
We assume $N(t) \geq 1$ for every non-empty admissible System state. Equivalently, in implementations one may use $N^+(t) = \max(1, N(t))$ to avoid degeneracy at initialization.

A complete formal definition of N(t) is given later in this same chapter.

Furthermore, considering that the growth of I_G must also be able to take control of it, we have:

$\log_2 (1 + (I_G))$.

The two relations can be operationally extended by introducing the Fragmentation Intensity Coefficient λ and the reference Information Unit I_0, resulting in the following Fragmentation Function defined:

Let $\xi \in \Xi$

denote the random outcome governing the stochastic decomposition and allocation process,

Fragments(t) =

$$\left|\mathrm{Ter}(t, \xi)\right| = \left\lceil \lambda \cdot \left[1/ \left(\rho_C(t) \cdot (N(t)^{\beta_{-N}})\right)\right] \cdot \log_2(1 + (I_G(t) + I_N(t))/ I_0)\right\rceil \quad \textbf{(86)}$$

where

$\lambda \in \mathbb{R}_{>0}$ and $I_0 \in \mathbb{R}_{>0}$, $N \in \mathbb{N}$

such that
λ, constitutes a Fragmentation Intensity Coefficient;
ρ_C, the Composite Density understood as a weighted geometric mean;
N(t) represents the total number of active internal nodes already present in the System at time t.

With

$E(t) = E_{int}(t) \cup E_{term}(t)$,

where $E_{int}(t)$ denotes the set of internal structural nodes and $E_{term}(t)$ denotes the set of terminal fragments, we define:

$N(t) = |E_{int}(t)|$, i.e., at time t
β_N, a Protocol Parameter;
I_0, a reference information threshold, indicating the interval in bits for each increase of +1 fragment, i.e., a positive reference unit of information that determines the scale at which increases in Global information trigger further fragmentation. Its value is defined by the implementation;
$I_G(t)$, $I_N(t)$ already defined previously,
all at time t.

With the formula in (86) the imposition of growth of the number of Fragments does not depend only on the number of initial Nodes, therefore on their variation, but also on the growth of the Global Information I_G, and evidently on the Nodal Information I_N independent of how it is distributed in the successive nodes.

The System or Implementation may provide for a variation of the various coefficients, including λ, to manage the efficiency of fragmentation.
The tables at the end of this document show concrete examples.

Definition
Let

$K_{VF}(v_j)$

the Knowledge accessible to the External Verifier vj at time t
and lets

$I_F(t)$, $I_N(t)$, $I_G(t)$

Fragmental Information, Nodal Information, and Global Information, respectively,

Recalling
$\mathbb{I}_G(t) \cong \mathbb{I}(S(t))$,
$K_G(t) \preceq I_G(t)$,

and the System follows the Imposition to the Increasing Depth of Decomposition n, the formulation

$$\text{Fragments}(t) = F_x(t) = \left\lceil \lambda \cdot \left[1/\left(\epsilon_C \cdot (N(t)^{\beta\_N})\right)\right] \cdot \log_2(1 + (I_G(t) + I_N(t))/ I_0)\right]$$

When the System computes the required number of fragments with respect to the admissible target threshold, the current density term $\rho_C(t)$ may be replaced by the protocol threshold ε_C. In that case, the formula expresses the minimum fragmentation intensity required to keep the composite density below the admissible bound.

Therefore, where

$F_x(t) \in \mathbb{N} \geq 1$

with

$\lambda\uparrow \Rightarrow F_x\uparrow$
$I_G\uparrow, I_N\uparrow \Rightarrow F_x\uparrow$
$I_0\downarrow \Rightarrow F_x\uparrow$
$N(t)\uparrow \Rightarrow F_x\downarrow$

and then $K_{VF}(t)\downarrow$

It follows that:
The System, through its invariants, imposes a Distribution of Information such that the Local Knowledge K_{VF} accessible to each External Verifier vj is structurally insufficient for the reconstruction of the Nodal Information and the Global Information.

Since the fragmentation function $F_x(t)$ is itself an increasing function of the global information $I_G(t)$, the growth of the System automatically induces additional decomposition whenever the informational complexity increases.

Consequently, the reduction of verifier-accessible knowledge is not a passive consequence of scale, but an active structural response imposed by the protocol.

Therefore, under the imposed density constraint $\rho_C \leq \varepsilon_C$, the relative informational significance of local verifier knowledge decreases as the System grows, yielding an asymptotic form of epistemic restriction,

such that:

$$\lim_{Fx(t)\to\infty} |K_{VF}(t)| / |I_G(t)| = 0,$$

The asymptotic restriction of local knowledge is not assumed independently. It follows from the density-control mechanism introduced in Axiom II, where increases in global informational complexity trigger additional decomposition whenever the composite density approaches the admissible threshold ε_C.

Therefore, the Knowledge Restriction Principle becomes a Structural Condition of the entire System.

Still, formally:

$K_{VF}(t) \not\Rightarrow I_N(t)$ **(87)**

and

$K_{VF}(t) \not\Rightarrow I_G(t)$ **(88)**

Local Knowledge K_{VF} does not contain, allow, or imply the information reconstruction of the Node or the Global System.
In Detail, given the Adaptive Decomposition

$\mathfrak{D}^{(n)}(S(t)) = \{s_i^{(n)}(t)\}$

and the progressive reduction of densities

$\rho_N, \rho_G, \rho_C,$

with

$\rho_N(t) = |I_F(t)| / |I_N(t)|$

$\rho_G(t) = |I_F(t)| / |I_G(t)|$

$\rho_C = (I_F/I_N)^{\alpha} * (I_F/I_G)^{\beta\_\rho}$

$\alpha, \beta_\rho \in [0, 1]$

with $\alpha + \beta_\rho = 1$, established that $t2 > t1$,

and after the Application of the adaptive fragmentation function

$\forall\ t2 > t1,\ I_N(t2) > I_N(t1) \Rightarrow \rho_N(t2) \leq \rho_N(t1)$

and

$\forall\ t2 > t1,\ I_G(t2) > I_G(t1) \Rightarrow \rho_G(t2) \leq \rho_G(t1)$

from the application of the imposed Axioms it emerges

$\bar{I}_F(t) = |I_h^{(n)}(t)| / F_x(t)$
(in expectation over the stochastic fragmentation process)

where

$I_h^{(n)}(t)$

constituting the decomposition-level information $I_h(n)$ of all the Fragments that make up the same level at time t, but not the level preceding it, and $K_h^{(n)}$ its decomposition-level knowledge, we have that

$|I_h^{(n)}(t)| = \Sigma_j\ |I(s_{ij}^{(n)}(t))|$ e $K_h^{(n)}(t) \preceq I_h^{(n)}(t)$,

with

$I_F(t) = \bar{I}_F(t)$,

where $\bar{I}_F$ denotes the expected average fragmental information size,
which constitutes the average size of fragmented information, since we have assumed that they are not uniform among themselves,
the relationship therefore becomes

$(\rho_C, \lambda, I_0) \Rightarrow F_x(t) \Rightarrow I_F(t) \Rightarrow K_{VF}(t)$

and since

$K_{VF}(t) \lneqq I_F(t)$,

it turns out that:

$\lim_{Fx(t)\to\infty} |K_{VF}(t)| / |I_N(t)| = 0$ **(89)**

and consequently,

$\lim_{Fx(t)\to\infty} \rho_N(t)=0$ **(90)**

Information limitation ultimately constitutes a form of structural insufficiency of Local Knowledge with respect to the Global complexity of the System, such that the complete informational reconstruction of the Fragment, of all the levels preceding it, and of the Global one, remains an emergent property exclusive to the System.
The Principle of Information Restriction therefore becomes an Epistemic Property deriving from the Application of the Axioms.
Formally, since

$D^{(n)}(S(t)) = f(F_x(t)) = f(\rho_C(t), \lambda, I_0, N(t))$

we conclude that

Depth $\uparrow \Rightarrow |K_{VF}(t)| / |I_N(t)| \downarrow$. **(91)**

The lower the allowable Composite Density, the greater the depth, fragmentation, and information dispersion must be: that is, the System increases the Decomposition to reduce the accessible Information Density, based on the set Protocol Parameters, over which it has full control.
This implies that the reduction of Knowledge can be "regulated" within the System as a Condition whose Threshold level depends on the Coefficient ϵ_C and how this affects the imposed Value of ρC and overall on all the Protocol Parameters, including α, β_ρ, and I_0.

Let $H(I_N)$
be the Information-theoretic Entropy of the nodal information, measured in bits.

The Knowledge Restriction Principle requires that the Conditional Entropy of the nodal information, given the fragmented knowledge K_{VF} available to a single verifier, remain close to the original Entropy:

$$H(I_N | K_{VF}) \approx H(I_N) \quad \textbf{(92)}$$

Ideally, the information obtained about the complete Nodal State is limited, and the Verifier's knowledge of the Fragment should not significantly affect the Uncertainty across the entire Node, because the verifier should learn virtually nothing about its overall content through fine-grained fragmentation.
This can be expressed using Mutual Information:

$$I(K_{VF}; I_N) \approx 0,$$

that is, each Verifier knows only a limited and increasingly smaller portion of the overall Content, so we will have

$H(I_N | K_{VF}) = H(I_N) - I(I_N; K_{VF})$

$H(I_N)$, total Entropy of the node;

$H(I_N | K_{VF})$, residual Entropy after observing the fragment.

This means that access to a single Local fragment does not significantly reduce the uncertainty about the complete nodal state, because the System is designed to minimize the mutual information between the verifier's Local knowledge and the complete nodal state.

---

11. Operational Flow

In this chapter, we will describe the System's operational flow. After illustrating the procedure used by the System to fragment the I_G Information and obtain the final decomposition depth, we will describe in detail how the semantic information is distributed from I_G to the final levels: that is, we will see how the system constructs the acquired Information Graph for the Object.

In the first phase, the User identifies the Object O_x.
Recalling:

$$O_x \rightarrow I(O_x) \rightarrow \tilde{I}(O_x) \rightarrow E(t),$$

$$O_x \rightarrow I(O_x) \rightarrow \tilde{I}(O_x) \rightarrow E(t) \rightarrow E^{(n)}(t) \rightarrow s_i^{(n)} \rightarrow \mathbb{I}(s),$$

$$O_x \rightarrow I(O_x) \rightarrow \tilde{I}(O_x) \rightarrow E(t) \rightarrow E^{(n)}(t) \rightarrow s_i^{(n)} \rightarrow V_{local}(s_i^{(n)}) \rightarrow P_i^{(n)} \rightarrow V_{Global}^{(n-1)}(S^{(n-1)}(t)),$$

$$O_x \rightarrow I(O_x) \rightarrow \tilde{I}(O_x) \rightarrow E(t) \rightarrow E^{(n)}(t) \rightarrow s_i^{(n)} \rightarrow V_{local}(s_i^{(n)}) \rightarrow P_i^{(n)} \rightarrow V_{global}^{(n-1)}(S^{(n-1)}(t)) \ldots\ldots \rightarrow V_{Global}^{(0)}(S^{(0)}(t)) \rightarrow V_{Global}(S^{(}(t))$$

The Object is introduced into the System in the form of a Verification instance

translated according to a binary representation, to define the components of the Structural Universe □ from which the System constructs the candidate structured state.

This phase constitutes a delicate moment because the constitution of the Object Structure in binary representation, starting from the User's request, will define the characteristics of all the specificities of the Object on which the subsequent comparison will depend.

In this sense, an inadequate representation of the Object's specific characteristics due to a lack of detail could lead to the Object's exclusion from the System's admitted Existence scope following the application of the verification process.

This eventuality is entirely legitimate and allows us to conclude that the Object is admitted into the System if and only if it exists in all the specific characteristics described in detail by the User, according to their needs.
The System, in fact, responds to the User's needs and adapts to their request.
To be clear, a poorly detailed User request would certainly lead to a conclusion that is inconsistent, but not necessarily weak, as for that User, a "rough" and poorly detailed assessment might suffice.
Let's take an example.

If we introduce into the System an instance to verify the existence of the Object "the gold bar exists in cav. number 17," without specifying any further details, the System could simply verify the existence of the Object "gold bar" in the environment "cav. number 17," without further specifications.
In this sense, the User's request could be granted a favorable response, without necessarily implying System Failure, since the request is very generic and the System simply needs to verify the Object in the form presented by the User: "a gold bar," without any specification of weight, purity, shape, size, state of preservation, and so on.
And this might already be sufficient for the User.

The System would therefore simply validate the Existence of a generic "gold ingot" without further details, and would admit it into its Field of Existence if the verification procedures returned a 1.

This means that the more detailed the description of the Object's specific characteristics in the instance requested by the User, the more thorough the verification structured by the System will be, which will be able to impose more rigorous checks on all the details relating to the specific characteristics described in the request.

Returning to the example above, if we introduce the following verification instance into the System, "the gold ingot Object, weighing 122.3 grams, with a purity of 99.5%, density of 19.3 g/cm^3, cubic in shape, with sides measuring 1.9 cm in length, exists in vault number 17. It is stored isolated, in a controlled environment," it is clear that the System's verification will be more detailed.
In this sense, verifying the Existence of a generic "gold ingot" will not be sufficient; all the specific characteristics described in detail will be sought.

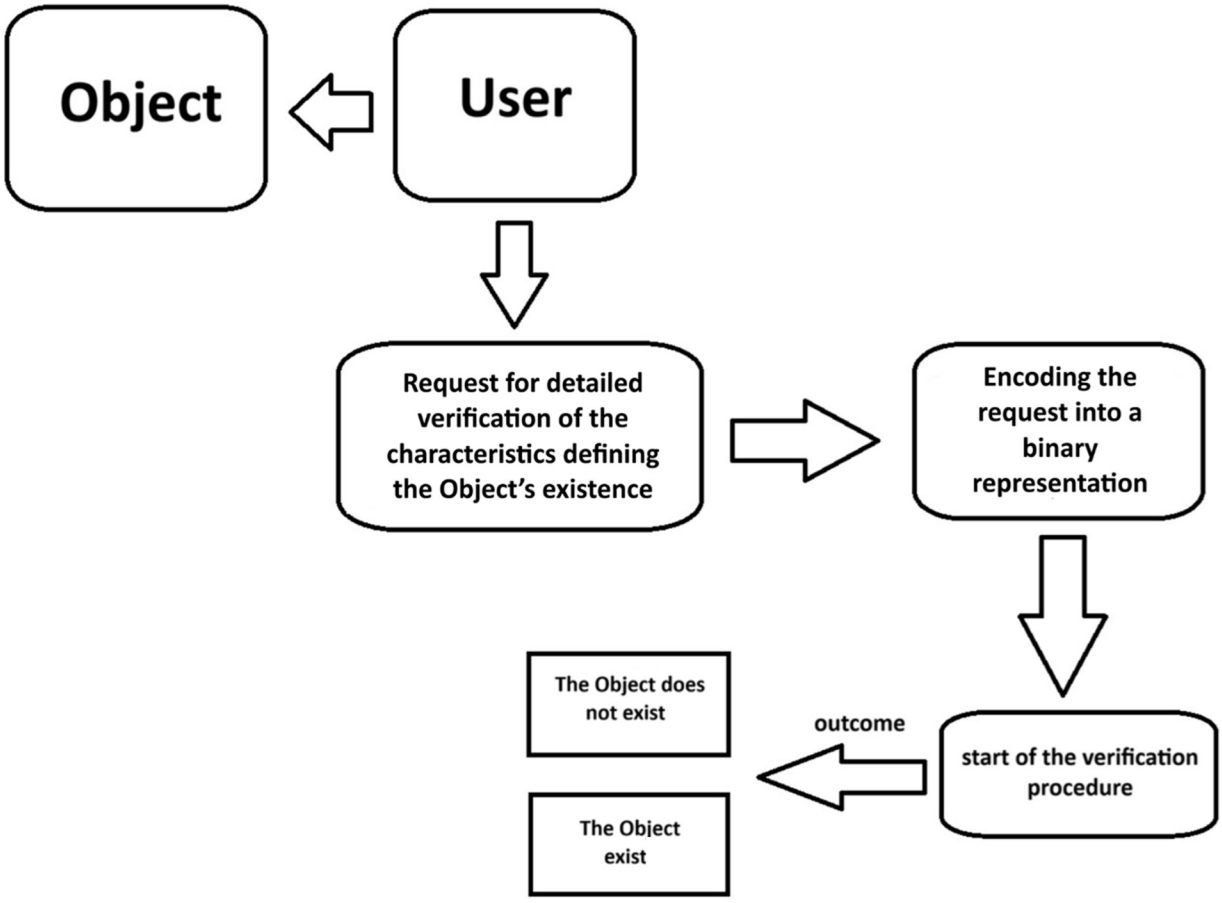
The System, for example, could reject the Status of Existence because the verification procedures would reveal: "the gold ingot weighing 122.3 grams, with a purity of 91.2%, a density of 18.2 g/cm^3, cubic in shape, with sides measuring 2.3 cm in length, exists in vault number 17. It is stored in a non-isolated, poorly controlled environment".

An Object Exists in the System if and only if the specific characteristics described by the User are fully matched.

Translating the description provided by the User into binary representation constitutes another delicate phase.

For some Objects, it would theoretically be possible to upload documents to the IT devices that use the CNVS System, such as a court ruling or a user manual for a device whose functionality and technical specifications you wish to verify; in other cases, this phase could be more complex.
However, as we have observed, the complexity of the introduction phase does not invalidate the conclusions reached by the System.

Schematically, we have:



The verification procedure for the random assignment of Global Information follows a logical operational chain using a randomness function in several steps.

The randomness function enters the System multiple times.

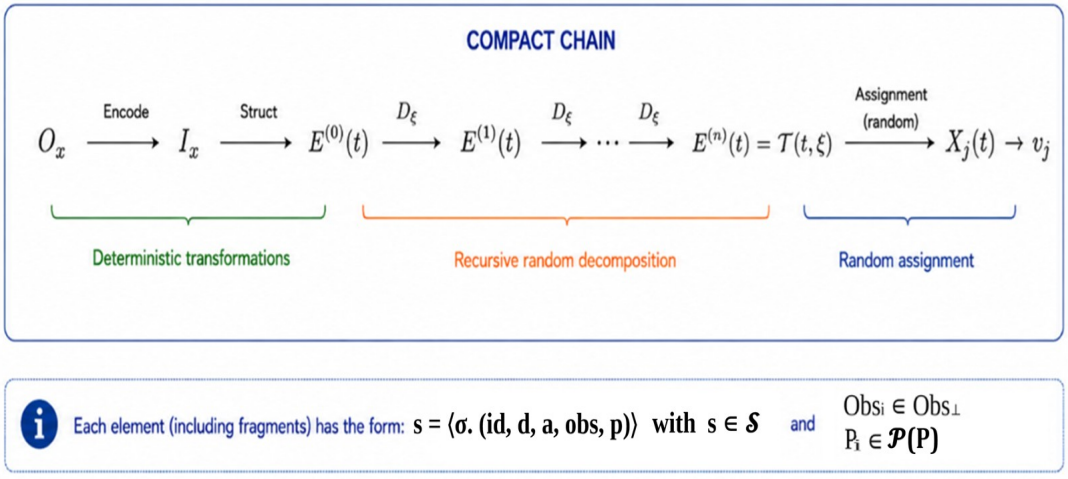
It initially enters during the Fragmentation phase of E(t), recalling

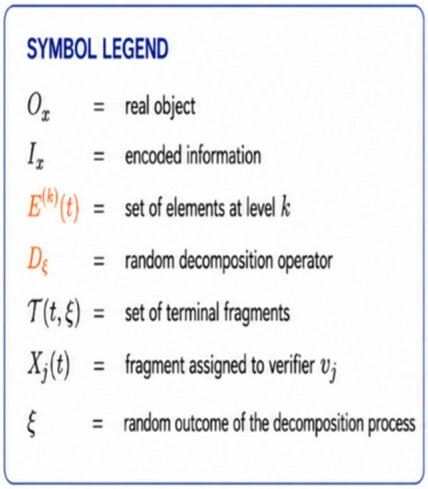
$E(t) \subseteq$,

with

$\Sigma_{\sigma \in \mathcal{T}} (Id_\sigma \times D_\sigma \times At_\sigma \times Obs_\sigma^{\perp} \times \mathcal{P}(P))$,

so schematically we will have the following general operational flow





At this point, it remains to be clarified how the System imposes the division of the granulation.

In Chapter 5, we saw the recursive property of fragmentation, and in Chapter 10, we introduced formula (86), which defines the number of terminal Fragments that the System imposes to divide the information to be randomly assigned to the External Verification Agents.

However, we have not yet seen how the fragmentation is divided.
In CNVS Systems, the solution is to impose the decomposition starting from the Origin, so as to create multiple initial Level 0 Nodes, thus preventing all the information from being traced back to a single Node.

Subsequently, further levels of subdivision are imposed, with a minimum of at least 4 levels.
The solution must take into account the need for the terminal Fragments to retain a logical sense, because a recursive subdivision that destroys the semantic content would become useless.
For this reason, the System starts from the last level, the terminal level n or Fragments(t).
From that level, it reconstructs the entire tree in an ascending direction, progressively assembling the terminal Fragments.

During ascending reassembly, the semantic content of D_i does not need to become directly readable at every intermediate level. What matters is that the relational dependencies required for reconstruction are preserved within the structured representation $\mathbb{I}$, through $\mathbb{I}_{metadata}$ and R(t).

In this way, the semantic payload may remain cryptic or non-operationally exposed during intermediate aggregation steps, while structural coherence remains available to the System.

Formally, we will have

$Ter(t, \xi) \rightarrow E^{(n-1)}(t, \xi) \rightarrow \cdots \rightarrow E^{(0)}(t)$.

Calling,

$S^{(k)}(t) = \langle E^{(k)}(t), R^{(k)}(t), C^{(k)} \rangle$,

$\Sigma^{n}_{k=1} \; f_k = Fragments(t)$

e:

$f_n > f_k$ for most $k < n$

e with $n \geq 4$.

therefore $|Ter(t, \xi)| = Fragments(t)$ **(93)**

This operationally biases the decomposition flow toward the terminal level, while the actual cardinality of the terminal level remains defined by $|Ter(t, \xi)| = Fragments(t)$.
To manage the distribution flow, the proposed solution is to introduce a new parameter: θ.
This parameter guides the distribution of Fragments(t) in an ascending direction.
The ideal values it can assume range from 0.20 to 0.25 for the central levels up to the first Node and range from 0.25 to 0.30 at the terminal level n, thus ensuring the maximum number of Fragments at the last level, which is the useful one.

Formally, we will have

$Fragments(t) := |Ter(t, \xi)|$

where Ter(t, ξ) is the set of terminal Fragments generated by the random decomposition process.

Let

$\theta(\xi) = (\theta_1(\xi), \theta_2(\xi), \ldots, \theta_n(\xi))$

with

$\theta_k \in (0, 1)$,

$\Sigma^n_{k=1} \theta_k = 1$.

The System generates the random sequence of allocations:

$(f_1, f_2, \ldots, f_n)$

such that

$\Sigma^n_{k=1} f_k = \text{Fragments}(t)$

with

$f_k \approx \theta_k(\xi)\ \text{Fragments}(t)$

where

$\theta_k \in [\theta_{min}, \theta_{max}]$

$\theta_{min}, \theta_{max} \in (0, 1)$

with

$\theta_{min} < \theta_{max}$

and for the terminal level it could be

$\theta_{min} = 0.25$

$\theta_{max} = 0.30$

therefore

$\theta_k \in [0.20, 0.25]$ for $1 \leq k < n$

$\theta_n \in [0.25, 0.30]$.

The intervals [0.20,0.25] and [0.25,0.30] are indicative operational ranges for implementations with a small finite depth, typically n=4 or n=5.
For larger depths, the admissible range of θ must be renormalized so that $\sum^{n}_{k=1} \theta_k = 1$.

Combining the two formulas,

Let $\text{Layer}_\xi : \mathbb{N} \to \mathbb{N}^n$

$\text{Layer}_\xi(\text{Fragments}(t)) = (L_1(t), L_2(t), \ldots, L_n(t))$,

where Layer_ξ depends on the random outcome ξ through the flow vector θ(ξ)

therefore

$L_k(t) = \lfloor \theta_k(\xi) \cdot \text{Fragments}(t) \rfloor$ for $k < n$

$L_n(t) = \text{Fragments}(t) - \sum^{n-1}_{k=1} L_k(t)$

with

$\sum^{n}_{k=1} L_k(t) = \text{Fragments}(t)$.

The vector Layer_ξ does not define the actual cardinality of each decomposition level. The terminal level remains defined by:

$$|\text{Ter}(t, \xi)| = |E^{(n)}(t, \xi)| = \text{Fragments}(t).$$

Rather, Layer_ξ defines an auxiliary operational distribution vector used to guide the ascending organization, aggregation, or branching-budget of the recursive decomposition process.

Thus,

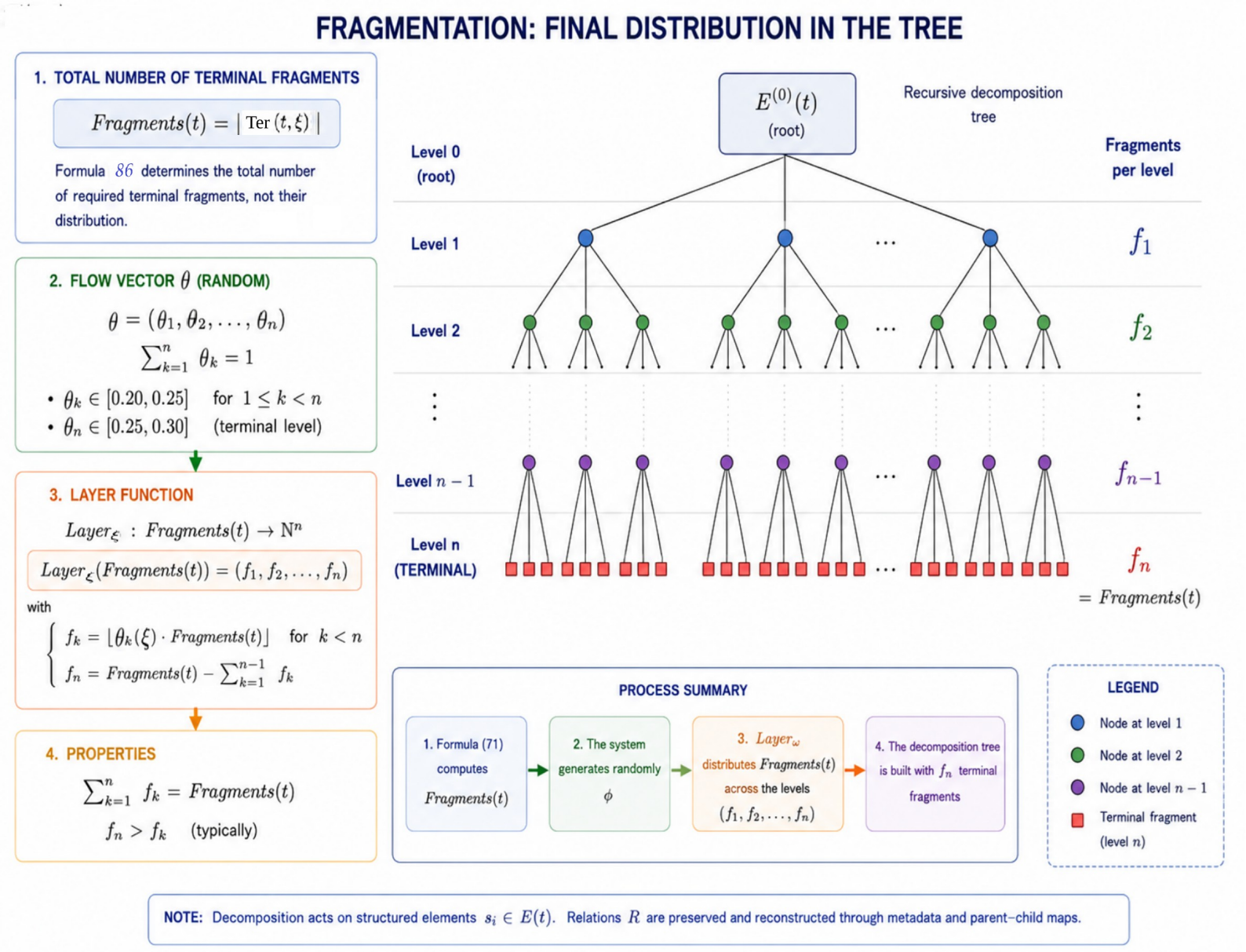
$$\text{Layer}_\xi(\text{Fragments}(t)) = (L_1(t), L_2(t), \ldots., L_n(t))$$

with

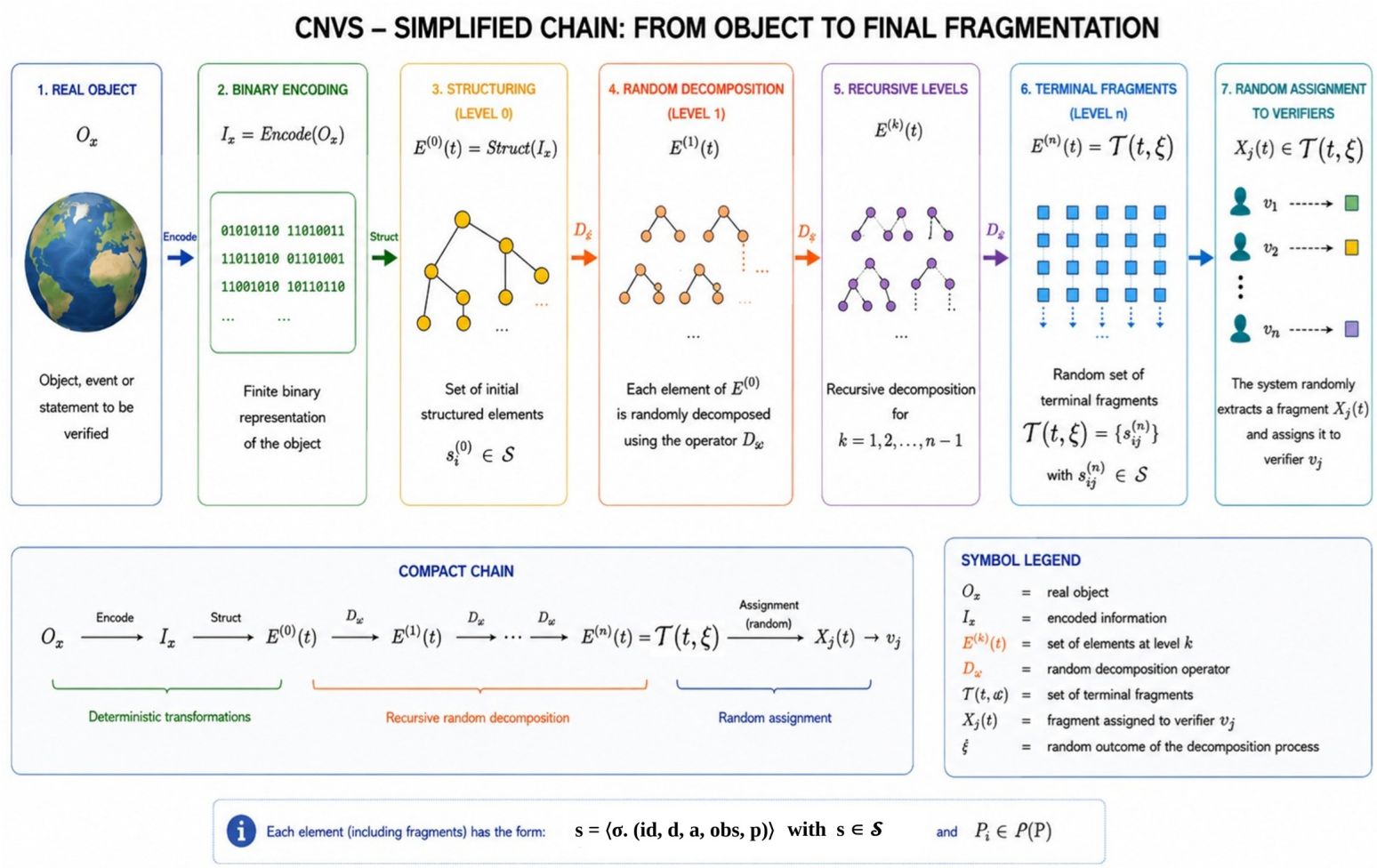
$$\Sigma^{n}_{k=1\}} L_k(t) = \text{Fragments}(t)$$

should be understood as a control allocation over the decomposition flow, not as the assertion that each level $E^{(k)}(t)$ has cardinality $L_k(t)$.

Schematically we will have

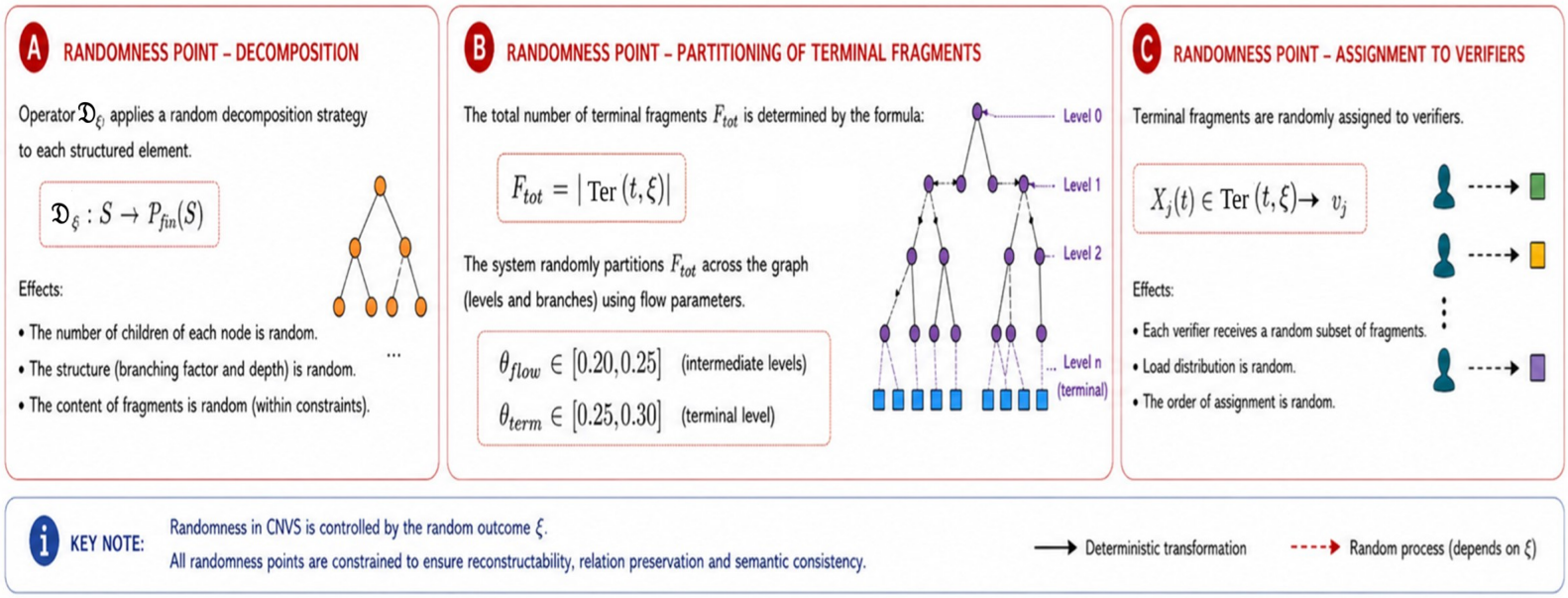


The System Operational Flow can therefore be schematized as follows



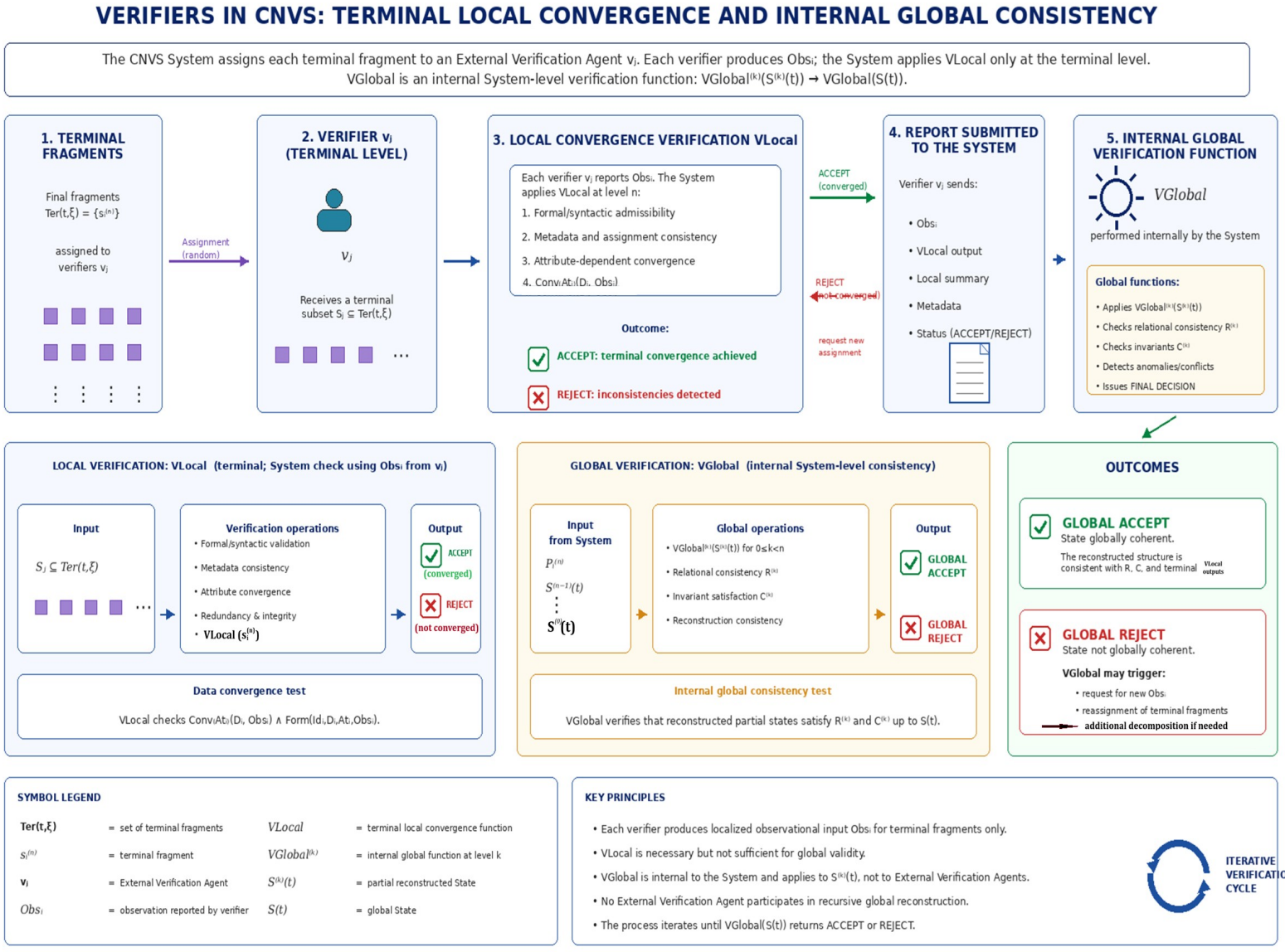
taking into account the distribution of the terminal Fragments as set out in the previous diagram.

Schematically, the randomness function guides the entire process by intervening in 3 points:



The diagram clearly shows the causal relationships.

The following diagram shows the role of the External Verification Agent and the V_{Local} and V_{Global} Functions in the operational flow



V_{Global} is a System-level Verification Function, not an External Verification Agent.

The terminal level is level n, whose cardinality is Fragments(t)

In Synthesis, It is important to clarify that the CNVS System does not require the continuous semantic reconstruction of the Global Information I_G in order to operate coherently.

The System does not reason directly on the complete semantic payload of the fragmented Information, but rather on its structured relational representation I_G, preserving the logical dependencies among terminal Fragments through the relational structure R(t), the invariant set C, the Local Verification outputs, and the metadata required for structural consistency.

During the reconstruction process, the System does not necessarily reassemble the complete semantic topology of the original Information. Instead, it progressively evaluates relational coherence, invariant satisfaction, and intermediate structural admissibility across decomposition levels.

In this sense, reconstruction is not primarily semantic, but structural and relational.

The semantic payload remains progressively hidden during the intermediate aggregation phases, while only the relational dependencies required for coherent verification become operationally exposed.

Consequently, the System minimizes unnecessary semantic exposure during the verification process, because the verification logic operates primarily on structural consistency rather than on direct semantic accessibility of the complete Global Information.

The External Verification Agent does not participate in this reconstructive process. Its role is limited to the production of localized observational inputs Obs_i, which are subsequently processed internally by the System through the Verification Functions.

The CNVS System therefore does not require a continuously reconstructed semantic global topology in order to preserve coherence, validity, and operational consistency.

---

12. Discussion on Shannon Entropy

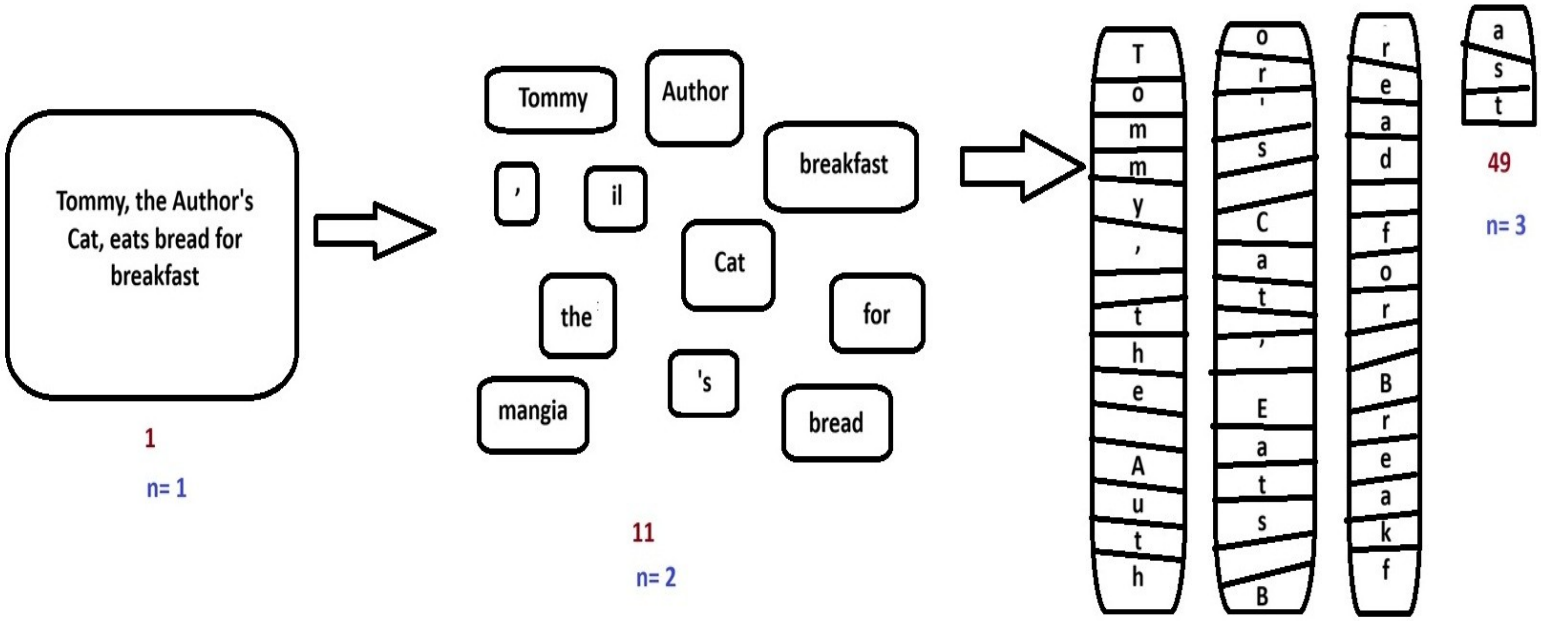
(nb. in this chapter, H(G) denotes the intrinsic information-theoretic Entropy of the global state, whereas $Hn=\log_2(k!)$ denotes the combinatorial uncertainty associated with fragment assignment
In this chapter, when the expression $H\_n = \log_2(n!)$ is used, n denotes the number of distinguishable terminal fragments involved in the reconstruction problem, not the recursive decomposition depth.)

Formalizing the behavior of Shannon Entropy in CNVS Systems is a central point of the theoretical development.

In this discussion, Shannon Entropy is used as an information-theoretic measure of reconstruction uncertainty associated with the combinatorial arrangement of Fragments, rather than as a direct measure of the semantic content of the original Information itself.

The System Architecture is based on the premise that the progressive reduction of the information content of the terminal I_F Fragments, according to the Knowledge Restriction Principle, due to the information granulation of the original I_G Information, makes its reconstruction increasingly difficult for the External Agent.
In an extreme model, by repeating the fragmentation ad infinitum, the information content of IF becomes completely devoid of meaning for the External Agent, losing any accessible logical coherence from the External

Agent’s perspective.
An example might clarify the conceptual meaning.
Let's look at the Box below.



It seems clear that the sentence "Tommy, the Author's Cat, eats bread for breakfast" makes logical sense to the External Agent: the Information begins with a series of intuitive logical relationships that anticipate and facilitate its reconstruction.

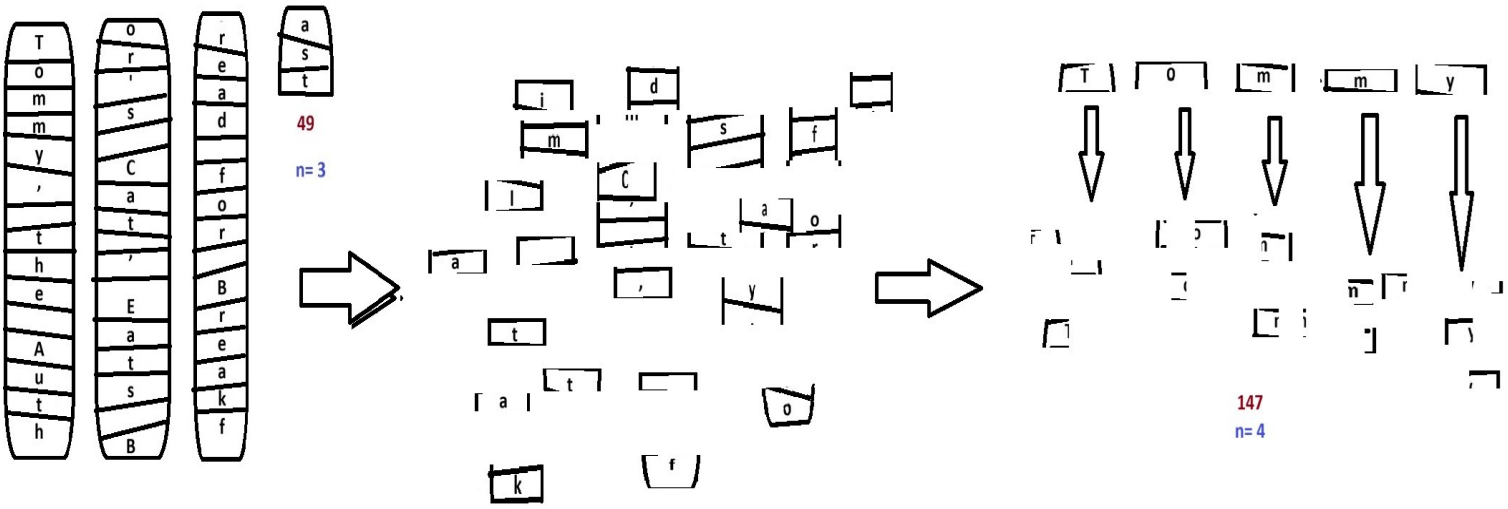
The situation on the left side of the Box appears quite clear: one enormous fragment at Level 1, conceptually strong.

In the central section of the Box, we already see something interesting: following the first informational granulation, the sentence's content is decomposed, producing 9 Fragments at level 2.

On the right side of the Box, we see something even more interesting: following the second informational granulation, the sentence's content

becomes highly decomposed, producing 49 Fragments at level 3. At this level, the message is no longer clear, but the semantic content of each individual part still has meaning for the reader, even if it lacks any intuitive logical relationship: they appear to be just scattered letters (T, o, c, l, z, etc.).

Let's continue our reasoning and see what happens at level 4, in the new Box below:



The result is a box containing a set of objects devoid of logical meaning, which have even lost their semantic meaning.

The knowledge of an External Agent with level 4 Fragments will be even more limited, because the absence of any logical and relational constraints between them makes reconstruction extremely difficult.
If we were to continue the granulation process virtually infinitely, we would obtain a fragmentation of the I_G information such that the I_F would become so destructured as to make reconstruction increasingly technically difficult, to the point of becoming virtually impossible in extreme cases.

This example shows how the CNVS System, using the decomposition of the source I_G information, produces the effect of reducing the information content of each individual fragment obtained I_F, such that the I_F content becomes a function of the number of Fragments.

Approximating, we could say:

$$K_{VF}(v_j, t) \lneqq K_F(t) \preceq I_F(t) \approx \bar{I}_F(t) = \mathcal{J}(s_{ij}^{(n)}) \approx I_G / n. \quad \textbf{(94)}$$

In the ideal asymptotic model, the average fragmental information tends toward zero as the number of fragments grows without bound. In applied CNVS Systems, however, fragmentation is stopped at the operational lower bound $I_{min} = I_{Fmin}$, because each terminal fragment must retain enough information to support the assigned verification task.

(n.b. the value I_{Fmin} is defined as I_{min} in other parts of this document and in other documents linked to this one).

More clearly, the information content of I_G can be interpreted in terms of progressive decomposition.

In an extreme CNVS System, beyond a certain information granularity threshold (Semantic Limit, I_{Fmin} or I_{min}), the fragment loses its semantic, logical, and relational meaning, and its exact reconstruction becomes possible only in probabilistic terms.
In theoretical CNVS models, fragmentation is treated as an ideal asymptotic process, in which the number of fragments can grow indefinitely as long as each verifier retains the minimum information necessary to perform the assigned task.

In the application, it is not possible to reduce the K_{VF} knowledge beyond the limit of the External Verification Agent's understanding of the semantic meaning of the content related to the System's request, because this would make it impossible to perform the task assigned to the Verifier.

This limit is set in this document as I_{min} (= I_{Fmin}).
In applied CNVS Systems, it is sufficient for the granulation to go beyond the limit of Global knowledge of the I_G and the Relations between I_{Fs}.
Once that limit is reached, in fact, the External Verification Agent will have no ability to contextualize the data it holds, and it will be even more difficult for it to understand the semantic Relations compared to I_F Fragments of the same level, or worse, of higher levels, because in addition to being encrypted, the Fragments are completely disconnected from each other, are non-uniform, and are randomly distributed.
Virtually any Fragment could contain any part of the I_G, and any value of D_i with respect to At_i.

The following discussion constitutes an approximation model, also with respect to the actual value of K_{VF}.
As clearly emerged in the previous chapters, the External Verifier does not possess the knowledge (K_F) of the fragmented information (I_F), but only the minimal information necessarily entrusted to him by the System to perform the task of performing measurements and checks on the Object under examination (K_{VF}).
The External Verifier does not know the D_i data, but can infer it from measurements and direct access to the Object.
However, the data measured by him will not necessarily converge to the D_i value recorded by the User during the implementation of the Verification Request.
And it is precisely the Verifier's blindness to this data that makes the System robust.

To develop the arguments that follow, the model assumes for simplicity that the External Verifier's knowledge is attributable to the D_i data relating to a portion of the Fragmental Information.
Therefore, the discussion develops the arguments starting from the I_F data and its general information content, with particular reference to D_i, such that

$K_{VF} \approx I_F^{(Di)}$,

with $I_F^{(Di)} \subsetneq I_F$,

and recalling $I_F \neq \mathbb{I}_F$.

The same approximation is also used in the development of the model relating to the emergent security theorem.

Therefore formally,

$$|K_{VF}(v_j, t)| \geq I_{min} \quad \textbf{(95)}$$

$|I_F(t)| \geq I_{min}$,

with

$|K_{VF}(v_j, t)| \leq |I_F(t)|$.

Assuming that the reconstruction problem consists in identifying the correct ordering of n distinguishable Fragments among all equally likely permutations, and admitting the simplified model in which all the Fragments are standardized to the same information size and are randomly distributed as desired by the System, also admitting that all are encrypted and completely anonymous, the probability of a successful random reconstruction is:

$p = 1/ n!$

If I_G Decomposition is applied in cycles such that its content is progressively fragmented twice at each cycle, the situation will be as follows:
after the first granulation cycle, we will have
I_G divided into two Fragments, each containing 1/2 of the original information.

Formally,

assuming

$|\bar{I}_F(t)| \approx |I_G(t)|/ n$, under uniform average fragmentation assumptions

therefore

$|I_F(t)| \approx |I_G(t)|/ 2$

with
$p = 1/ (2*1) = 0.5$

If we apply 3 cycles, we will have:
$|I_{F,3}(t)| \approx |I_G(t)|/ 2^3$ with $p = 1/ (8*7*6*5*4*3*2*1) = 1/ 40.320$.

If we repeat the cycle 20 times, we will have:
$|I_{F,20}(t)| \approx |I_G(t)|/ 2^{20}$ with $p = 1/ (2^{20})!$,

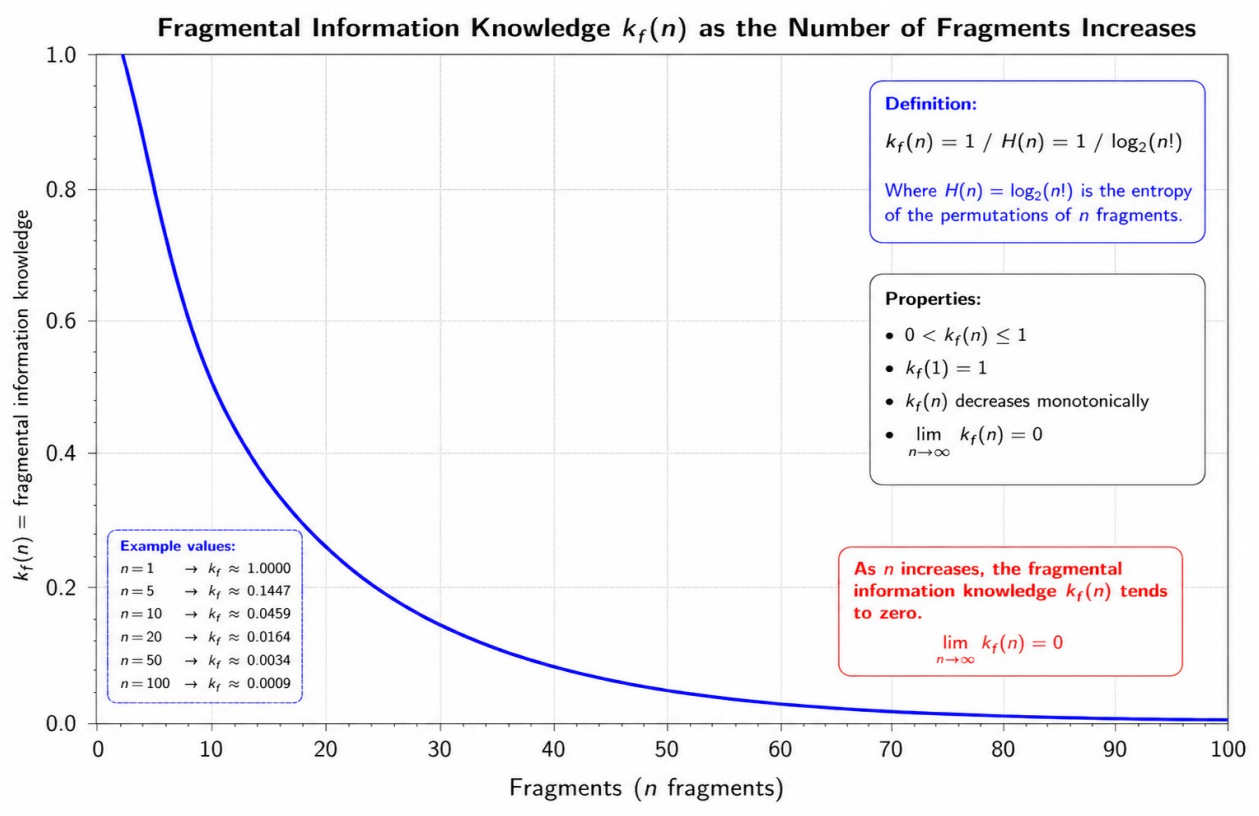
generically,

$|I_{F,m}(t)| \approx |I_G(t)|/ 2^m$ and $p_m = 1/ (2^m)!$

with $m \in \mathbb{N}$,

m is the number of decomposition cycles.

It is clear that the probability of finding the right combination by relying solely on chance becomes so low that the effort is completely useless.
Graphically, we can represent the relationship between the Knowledge of Fragmental Information $K_F(n)$ and the number of Fragments (n), with $n \to \infty$



In this context, Shannon Entropy measures the uncertainty associated with the reconstruction problem, namely the uncertainty of identifying the correct configuration among all equiprobable permutations of the Fragments.

Therefore, in CNVS Systems, increasing Shannon Entropy does not imply an increase in semantic Information, but an increase in the uncertainty and combinatorial complexity associated with the exact reconstruction of the fragmented Information.

We calculate Shannon Entropy in a model in which we have 20 Fragments arranged in a row and each recognizes a single successful combination among all equally likely combinations.
Let $p_i = 1/ 20!$,

let Shannon Entropy be

$$H(X) = -K \Sigma^{N}_{i=1} p_i \log_2 p_i, \quad \textbf{(96)}$$

N= 20!, K= 1

replacing, $H(X) = - \Sigma_{i=1}^{20!} 1/ 20! \log_2 (1/ 20!)$,

and since the term is the same for all combinations we will have

$H(X) = -20! * 1/ 20! \log_2 (1/ 20!)$,

$H(X) = - \log_2 (1/ 20!)$

$H(X) = \log_2 (20!)$

$H(X) = 61.08$ bit.

By composing the 20 Fragments one at a time we will have

$20*19*18*\cdots*2*1$

the correct probabilities at each step become

$1/ 20 * 1/ 19 * 1/ 18 * 1/ 17 *\ldots. * 1/ 2 * 1/ 1.$

then,

$p = 1/ 20!$

the total information will be

$I = -\log_2(p)$

$I = \log_2(20) + \log_2(19) + \log_2(18) + \cdots + \log_2(2) + \log_2(1)$

$I = \log_2(20!)$

$I \approx 61.08$ bit,

In conclusion,
p(correct combination by chance) = 1/ 20!
20! = 2,432,902,008,176,640,000.
Therefore, by randomly choosing an arrangement of the 20 Fragments, the probability of immediately obtaining the correct one is approximately:
$4.11 * 10^{-19}$
that is, approximately $4.11 * 10^{-17}$% probability of guessing the correct combination on the first try.

Examining the data from the Binary Information relating to the calculated Uncertainty, we will have that to uniquely identify the correct arrangement among all 20 possibilities, approximately 61 bits of combinatorial information are required [1, 0].
61.08 bits correspond to $2^{61.08}$ possibilities.
And in fact, $2^{61.08} \approx 20!$ corresponds exactly to the number of permutations of 20 objects.

Let's try to extend the reasoning by gradually increasing the Decomposition up to the limit $+\infty$ and see how the Entropy behaves.

The number of possible combinations/permutations is:
$N(n) = n!$
If all are equally probable:
$p_i = 1/ n!$,
substituting into the Shannon Entropy we have

$$H(X) = - \Sigma_{i=1}^{N} p_i \log_2 p_i$$

$$H(X) = - \Sigma_{i=1}^{n!} 1/ n! \, Log_2 (1/ n!)$$

therefore

$H(X) = - n! \; 1/ n! \, Log_2 (1/ n!)$, and simplifying

$H(X) = - Log_2 (1/ n!)$,

$H(X) = Log_2 (n!)$.

Let's calculate the limit

$$\lim H(X)_{n\to\infty} = \lim_{n\to\infty} Log_2 (n!)$$

with $n \to \infty$, then $\log_2(n!) \to \infty$

$$\lim H(X)_{n\to\infty} = +\infty \qquad \textbf{(97)}$$

Since n! becomes a very large number, we perform the calculations using Stirling's approximate formula, which was created specifically to handle these special cases.
The formula is $n! \approx \sqrt{(2\pi n)} * (n/ e)^n$

so we will have

$H(n) \approx \log_2 [\sqrt{(2\pi n)} * (n/e)^n]$

$H(n) \approx n \log_2 n - n \log_2 e + \frac{1}{2} \log_2 (2\pi n)$ **(98)**

The dominant term is:

$n \log_2 n$

$H(n) \rightarrow +\infty$

the probability instead will be

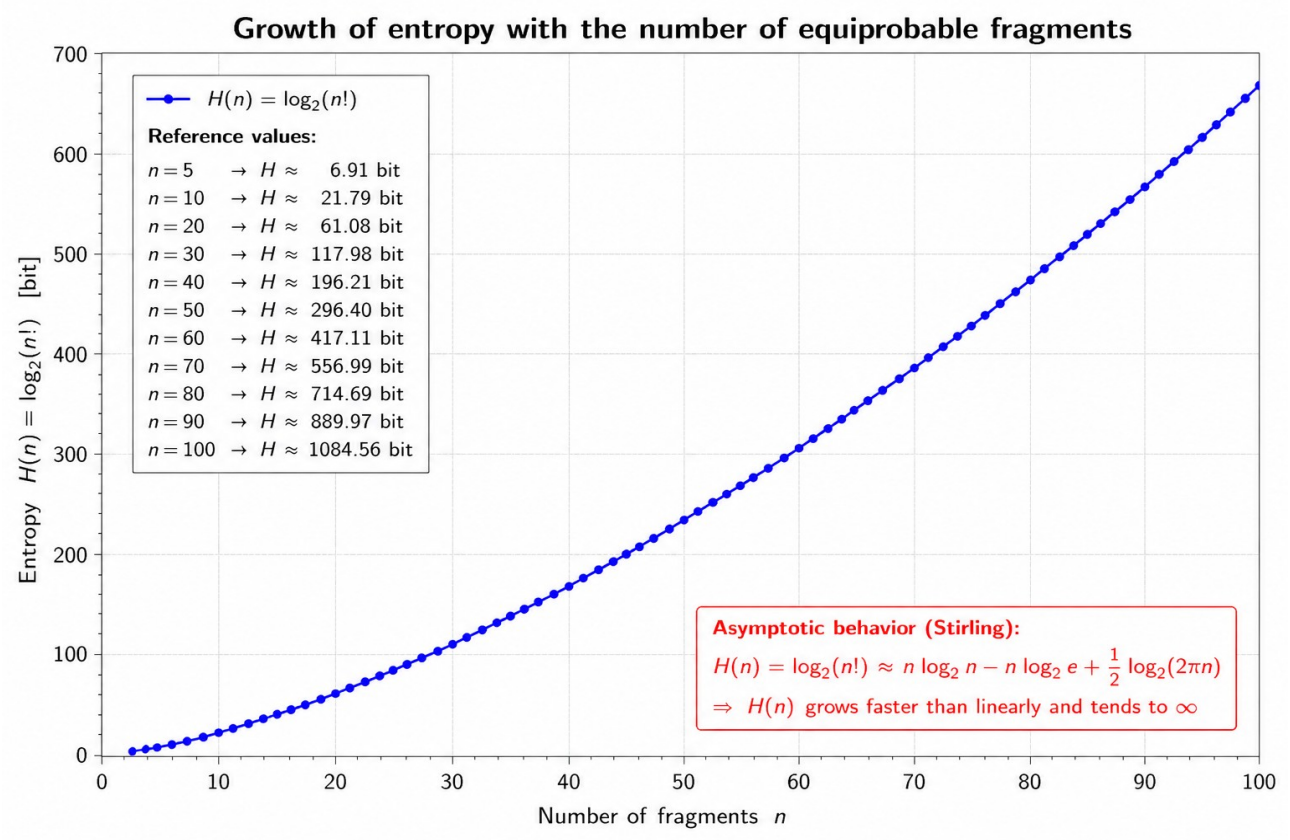
$p = 1/n!$.

Therefore

$\lim_{n\to\infty} p_n = \lim_{n\to\infty} 1/n! = 0$

Therefore, as the number of Fragments grows without limit, the probability of randomly guessing the correct combination approaches zero, while Entropy approaches infinity, therefore the uncertainty associated with exact reconstruction also diverges.

Graphically,



Therefore, in an ideal CNVS model, increasing fragmentation causes the fragmental information $I_F(n)$ to decrease, while the Entropy associated with the reconstruction problem increases. However, the System cannot pursue infinite fragmentation as an absolute objective, because below a certain operational threshold the fragment would no longer preserve the minimum information required for verification. The implementation must therefore identify an optimal threshold n^*, as a function of the source information I_G, at which reconstruction uncertainty is maximized while fragmental information remains operationally usable.

Concluding,

$I_F(n) \geq I_{Fmin}$

with I_{Fmin} denotes the minimum amount of fragmental information required for the fragment to remain operationally usable for verification

and with the threshold value thus formalized:

$$n^{*} = \max\{n \mid I_F(n) \geq I_{Fmin}\} . \quad \textbf{(99)}$$

---

13. State Rejection

The System is designed to resist External Hacking and Authoritarian Manipulation attempts, but there may be Situations, discussed in a separate document, that cause integrity failure.
These Situations can occur when certain Conditions that the System deems essential for its very survival cease to exist.

These Conditions at time t include:

- Density can no longer be maintained;
- Verifiers failure;
- Depth does not increase sufficiently;
- ρ_C exceeds the threshold ϵ_C;
- ρ_N exceeds the threshold ϵ_N;
- ρ_G exceeds the threshold ϵ_G.

The System also is rejected when the Internal Verification process is determined by the External Verification process in such a way that it becomes:

- predictable;
- manipulable;
- subject to collusion.

In all these cases, we will have:

the verification process ceases to satisfy the deterministic integrity assumptions required for valid CNVS operation.

Formally:

$Det_{Integrity}(S, t) = 0 \Rightarrow Reject(S)$

In these cases, V_{Global} does not cease to be formally deterministic. Rather, the deterministic integrity assumptions required for its valid application cease to hold.

and therefore

$S(t) \notin S_{valid}$

$Det_{Integrity}(S, t) = 1$

if and only if the verification process maintains:
• unpredictability of the assignment;
• non-manipulability;
• non-determination of the internal verification process by the external verification agents;

• absence of structurally dominant collusion.

In all cases $Det_{Integrity}(S, t) = 0$, the System loses the right to be considered a CNVS and becomes a Traditional System, because the Integrity of the Verification Function is not a functionality, but a definitional requirement, such that the System is not only violated, but becomes a different type of System. The Integrity of Verification is not a functional requirement, but an necessary precondition for membership.

It is worth noting that the State is rejected whenever a global invariant is violated. As already established in the previous sections, even when the fraction of compromised Verifiers exceeds the level that would ordinarily be considered critical, unauthorized reconstruction may still be detected and rejected if the colluding agents fail to acquire sufficient useful information to exceed the reconstruction threshold τ. In such cases, the System does not authorize the transition from the Candidate State to the Existing State.

This occurs due to the colluding verification agents' residual global semantic blindness with respect to the remaining portion of the System that they do not control. This portion, being unpredictable due to the non-uniform distribution of D_i in the terminal Fragments, is limiting, depending on its utility or weight $\omega(D_i)$ with respect to the global information I_G (where $\omega(D_i)$ denotes the relative informational weight of D_i with respect to I_G, as discussed in the following chapter).
The attacker will be unlikely to know the information content of the nodes he doesn't control, and given the randomized non-uniform distribution of the fragments, he won't be able to easily deduce it.

In this context, the Global Verification Function could easily highlight a discrepancy in the value expected by I_G or I_N during the intermediate

reconstruction phases: the recomposition in the higher-level Nodes of the Fragments controlled by the Attacker with the Fragments that he does not control and of which he has no knowledge, will lead to impossible results, violating the invariants of the System.

The CNVS System rejects the traditional consensus paradigm in which the majority necessarily wins. If even a structurally critical portion of information, relevant to a global invariant or to relational consistency, remains inconsistent or outside adversarial control, the V_{Global} function may fail and activate State rejection according to the Global Veto Principle.

---

14. Limitations of the Theoretical Model

External Verifiers constitute a Membership Class, which is not necessarily "human". Technically, External Verification can be entrusted to the use of hardware devices supervised by human professionals, or it can be completely automated.
This Document does not go into detail about the methods used to perform External Verification, but it assumes that it is performed with Objective Judgment, structuring a System that reduces the possibility of involuntary and intentional errors through Fragmentation Control and the Reconstruction of the Global Asset with its progressive implementation.
This is even more true considering that every External Verification is permanently archived within the System, which preserves the Historical Memory of each Individual Verification for each Individual Structured Element. The ability to track the Unique Identifier of each External Verification, as well as all individual steps within the System, makes an undetected violation

structurally difficult and theoretically incompatible with the traceability assumptions of the System.

External Verifiers can therefore be:

- Human auditors
- Automated verification modules
- Sensors
- Vaults
- Software agents
- Distributed modules.

Specifically, the sustainability of decomposition growth could become a long-term problem.
If the value of Fragments(t) were to grow too rapidly, then the computational cost would become difficult to offset.
At the same time, there could be issues with synchronization, latency, and relationship management.
Systemic Complexity can be well summarized by reading the following comparison table:

More fragmentation	**Less knowledge**
less knowledge	More emergent security
More fragmentation	More Systemic complexity

The application model forces a natural compromise to be made with respect to the conditions assumed by the theoretical model.
It should also be noted that not all Objects are equally suited to the CNVS Class.

An Object whose information content can be decomposed deeply before losing the minimum information I_{Fmin} will certainly be a candidate with a profile that

better meets the requirements allowed by the CNVS than an Object that is poorly suited to a deep recursive decomposition.
Obviously, its semantic nature will be a key element.

To formulate this notion, we can say that
the Granularity of an Object corresponds to its ability to lend itself to recursive decomposition without losing its semantic characteristics, even though it is distributed into disconnected fragments with no explicit logical connection visible to the External Verifier.

Formally,

The application model forces a natural compromise with respect to the idealized conditions assumed by the theoretical framework.

It should also be noted that not all Objects are equally suitable for the CNVS class.

An Object whose informational content can be deeply decomposed before reaching the minimum semantic threshold I_{F_min} is naturally a better candidate for CNVS architectures than an Object that poorly tolerates recursive fragmentation.

Its semantic structure therefore becomes a fundamental factor.

To formalize this notion, define the Granularity Ratio of an Object as its ability to support recursive decomposition while preserving the minimum semantic conditions required for local verification, despite the distribution of information into disconnected fragments with no explicit logical linkage.

Formally,

let Γ = Granularity, $\Gamma = m_{max} / m_{min}$, **(100)**

where

- m_{max} represents the maximum number of terminal Fragments that can be generated before the semantic information of each Fragment falls below I_{Fmin};

- m_{min} represents the minimum number of critical Fragments required to achieve the target security condition and the practical loss of inferable logical correlation between terminal Fragments.

Accordingly:

- If $\Gamma > 1$, the Object exhibits high semantic granularity. The practical architecture can approach the ideal asymptotic regime while preserving local semantic admissibility, and emergent security stabilizes before reaching the I_{Fmin} boundary;

- If $\Gamma \approx 1$, the architecture operates near the semantic feasibility limit;

- If $\Gamma < 1$, the Object is not semantically compatible with the required security configuration, and the CNVS instance becomes practically infeasible for that semantic domain.

---

15. The Emergent Security Theorem

15.1 Introduction

In the following discussion, we assume an approximation model in which the External Verifier's knowledge is attributable to knowledge of the Terminal Fragment and part of its content D_i.
This assumption makes the result even more robust, because in the non-approximated CNVS model, the External Verifier has less information to establish potential collusion.
The approximation therefore leads to an underestimation of the true security potential of CNVSs based on the results shown.

In this document, the Security of a System is defined as the Property of the System to resist attacks by Colluding External Agents attempting to violate it.
In this document, the term "Emergent" is used to indicate a Property of the System that emerges as a consequence of its intrinsic structure and is strengthened by its progressive adoption.
In this sense, the Emergent Security Theorem defines the Property of the System to resist attack attempts by Colluding External Agents that share the I_F Information assigned by the System with each other, in order to violate it.
The Theorem proposes an emergent Property such that the intention of the Colluding Agents clashes with the growth of the System's Resistance to violation with its progressive adoption and with the increase in the I_G content and therefore its complexity.
To understand the Theorem, we must use the tools that mathematics offers us.
Let's imagine a situation in which a number n_{bad_agent} of Colluding External Agents wants to breach the System at time t.

15.2 Development

Let's imagine, as a conservative approximation, that each of them has access to the share of Fragmental Information $I_F(t)$ entrusted to them by the System, in the form of Fragmental Knowledge $K_F(t)$, and that they can decode the operational content made available to them by the System. This assumption intentionally overestimates the adversarial knowledge, because in the non-approximated CNVS model the External Verifier has access only to the restricted verifier-visible operational view.

At this point, the Probability of Breaching the System will depend on:
- the amount of Information $I_F(t)$, in the form of Fragmental Knowledge $K_F(t)$, that each of them has;
- the random coherence of the individual shared Fragments: if N_{bad_agent} External Agents has "nearby" Fragments within the granular I_G Distribution, it is clear that the reconstruction of the original Information will be more likely than in the situation where External Agents share completely opposite Fragments within the same Distribution;
- the complexity of the System, understood not only as the total Volume of I_G Information, but also by the Depth of Progressive Decomposition achieved, which is equivalent to the total number of Fragments (Fragments(t) = $f_x(t)$) generated by the System;
- the degree of Entropy H(X) achieved by the System in its terminal Fragmentation.

At this point, the use of three mathematical tools allows us to demonstrate the Validity of the Hypothesis of the Theorem.
Shannon's Formula, as we saw in previous chapters, explores the System's uncertainty in identifying the correct configuration of the terminal Fragments during the reconstruction, providing a measure of the Entropy H(X) achieved by the I_G Information during the granulation process.

Bernoulli's Formula helps us in the Fragment assignment phase and allows us to calculate the probability that a Fragment will be randomly assigned to a Colluding External Agent, being a probability statistic in which an event can have only two outcomes (in our case, assigned to the Colluding External Agent or to the Honest External Agent).

Kullback-Leibler divergence measures the statistical "distance" between two probability distributions: in our case, we can use it to calculate the divergence between the critical Bernoulli distribution and the actual Bernoulli distribution. This statistical distance is a central point of the Theorem: that is, it evaluates when a specific Colluding Distribution could become a problem if it attacks the System, by comparing the Actual Colluding Distribution found in the System and the one considered Critical, that is, the Theoretical Distribution with a Critical Threshold sufficient to violate the System and corrupt it.
We therefore have:
Shannon Formula $H(X) = -K \Sigma_{i=1}^{N} p_i \log_2 p_i$,

described in previous chapters,
Bernoulli's Formula

$P(Y=1) = q$,

$P(Y=0) = 1 - q$

which measures
event 1 = a fragment is assigned to a Colluding External Agent.
event 0 = a fragment is assigned to an Honest External Agent.
The Kullback-Leibler divergence for discrete cases is
$D(P||Q) = H(P, Q) - H(P)$

where two Entropies are compared to each other, the Cross Entropy and the real Entropy of System P, that is, the Shannon Entropy.

Real Entropy $H = -\Sigma_x P(x) \log P(x)$

measures the intrinsic Uncertainty of the Real Distribution P, thus offering a measure of the Disorder of the Data (i.e. how “unpredictable” they are);

Cross Entropy $H(P, Q) = -\Sigma_x P(x) \log Q(x)$

It is a cross-measure, which measures the resulting Entropy if the events of the Real Distribution P are encoded using the approximate Model given by the Q Distribution.

If, for example, we wanted to measure tomorrow's weather, and we know that the Probability of Sunny is 70%, Cloudy is 20%, and Rain is 10%,
while our Estimate is 40% Sunny, 40% Cloudy, 20% Rain,
we would have
the Real Discrete Distribution 0.70, 0.20, 0.10 (A)
against
the Approximate Distribution 0.40, 0.40, 0.20 (B).
Cross Entropy is the comparison between these two distributions, obtained by measuring what happens to the Shannon Entropy of the Real Distribution (A) if I use the data from the Approximate Distribution (B). The result, expressed in bits, tells us how inefficient the Approximation is, that is, how many more bits of information I am "wasting" because the Q Model (which corresponds to B in the Weather example) does not adhere perfectly to the P Model (A in the example).
In our case, we will have $D(\tau||q) = H(\tau, q) – H(\tau)$, which by replacing becomes
$D(\tau||q) = \tau \log (\tau/ q) + (1 – \tau) \log ((1 – \tau)/ (1 – q))$ **(101)**

where

τ, it is the minimum fraction of coherent information needed to attack effectively;

q, is the actual share of colluders and

where log denotes the natural logarithm when $D(\tau \| q)$ is used in exponential concentration bounds

Said differently $D(\tau \| q)$ measures how unlikely it is that an entity with a collusive share q will behave as if it had a collusive share τ.

For example, if q = 10% and the threshold required for Collusion Success is 30%,
that is, q = 0.10 and τ = 0.30,
the Kullback-Leibler divergence measures how statistically distant 10% is from 30%.

In the proof, we introduced the value τ, the Critical Collusion Threshold. This value constitutes the minimum fraction of Global information that colluders must control to breach the System effectively and consistently. Put another way, if colluders control less than τ, they do not have enough information to cause problems. However, if they control an amount of information with assigned $I_{F\_S} \geq \tau$, they can conduct an effective attack.

However, during the discussion, it became clear that, although standard statistical reconstruction limits operate in the presence of some conservative parameterizations or simulation scenarios, a value such as τ = 0.35 can be used as an illustrative critical threshold, and the rigorous application of Axiom III alters the asymptotic behavior of the system.

In CNVS, validity is not reducible, being governed by the V_{Global} function, so the effective operating threshold is not dominated by the classic "majority vote."

Even when the fraction of compromised verifiers is very large, security may still hold if the uncompromised useful information remains sufficient to block unauthorized reconstruction or to trigger invariant violation during global verification.

Let's continue our reasoning on the probabilistic model and test the resistance of the System by considering Collusion without taking into account for a moment the violation of the invariants.

Let us assume that $q, \tau \in \mathbb{R}$ and $q, \tau \in (0, 1)$, and $n, c_{coll} \in \mathbb{N}$

c_{coll} = the number of Fragments controlled by the Colluding Agents
n = the total number of Fragments;
then the share controlled by the Colluding External Agents will be c_{coll}/n.
Collusion will only be possible when $c_{coll}/n \geq \tau$.
τ is a Model Parameter and can be estimated in various ways, but it is a parameter that should be provided.

$\tau \in (0, 1)$ is the minimum normalized amount of useful information required for unauthorized reconstruction.

Technically, it could be assumed by the Implementation in a "conservative" manner, but the most effective solution is for it to be estimated conceptually by analyzing how much minimal information is needed to reconstruct relationships, avoid contradictions, and violate the System's invariants. It is clear that this is difficult to calculate in an Implementation because it depends on the Nature of the Data we are dealing with, which is assumed to be abstract in this document.

Theoretically, it could also be estimated experimentally through Attack trials. If we assume through analysis and simulations that at least 35% of the useful information is needed to violate the System, τ will be equal to 0.35, meaning that those who control less than 35% of the information will, on average, fail to violate the System.

As the System grows, the information I_G grows, and therefore the number of Fragments increases, the Entropy H(n) increases, while the relative share of knowledge available to each verifier decreases.
It follows that an increasingly larger amount of information is needed to reconstruct the System, thus requiring more colluders.
Formally,

$$\tau = f\tau(H, R, C) \quad \textbf{(102)}$$

where we have
- H = System Entropy;
- R = Structure of relations;
- C = System invariants.

That is, the Critical Threshold τ is a function fτ of the informational and structural complexity of the System, therefore of H, R, and C.
The threshold τ may depend on the informational Entropy H, relational complexity R, and the System invariants C.

In the proof, we will use Bernoulli's Binomial Formula, which considers many independent Bernoulli trials.
For analytical tractability, the assignment of fragments is modeled as an idealized probabilistic process in which each fragment is independently assigned to a colluding verifier with probability q. This assumption provides a simplified approximation used to derive asymptotic security bounds.

Formally,
$P(X=k) = (n; k)\ q^k\ (1-q)^{n-k}$
where:
- n, the total number of Fragments;
- k, the number of Fragments managed by the colluders;
- q, the probability that a single fragment goes to a colluder;
- P(X=k) the probability that k Fragments are assigned to the Colluding External Agents.

To recap, with the simple Bernoulli's Formula, the probability that a single fragment goes to a colluder is established; Bernoulli's Binomial Formula calculates the probability that out of all the Fragments (i.e., in a distribution), a quantity k goes to the colluders.
In the demonstration we will use Stirling's Formula that we have already seen in the Discussion on Entropy in chapter 12, namely

$n! \approx \sqrt{(2\pi n)} * (n/e)^n$, and which constitutes an approximation when calculating the factorials of very large numbers.

As has clearly emerged, even treating the System as a probabilistic distribution deprived of invariant constraints, the resistance of the model grows with adoption.

15.3 Proof

Notation Note: In this extended analysis, the total number of terminal fragments undergoing verification is denoted by the scale parameter n. The variable k is used only as a counting variable in the binomial model, representing the number of fragments assigned to colluding agents.

The statement of the Emergent Security Theorem is:

$$q\bar{\omega} < \tau \Longrightarrow \lim_{n \to \infty} P[\Sigma^{n}_{i=1} Y_i \, \omega(D_i) \geq \tau n] = 0. \quad \textbf{(103)}$$

The more the System is adopted, the more useful knowledge is dispersed, and the more unlikely, costly, and technically inconvenient collusion becomes.

Let's start with the idea of having a certain number of terminal Fragments n.

For simplicity's sake, let's assume that they are all equally probable.

This simplifying assumption is introduced only to derive the Entropy of the System in a closed and intuitive form. The theorem is subsequently generalized by introducing an informational weight $\omega(D_i) \in [0, 1]$ for each fragment, thereby explicitly accounting for non-uniform fragment distributions. The unweighted binomial model is recovered as an analytical baseline by setting all counting weights equal to 1. The normalized informational-weight model is introduced afterward to account for non-uniform fragment utility.

If we want to recombine them all, the total number of all possible combinations corresponds to n!, and these are Permutations, that is, taking each individual Fragment only once, and taking into account the importance of the order in which they are distributed. In fact, the Fragments contain information and cannot be distributed in a random order.

As we have already seen in Chapter 12, the formula for calculating each individual Probability of a specific Permutation will be:

$p = 1 / n!$.

Let Shannon Entropy be:

$H = -\Sigma_i\, p_i \log_2(p_i)$.

Let's calculate the Entropy of our distribution of terminal Fragments n obtained from the System, and substitute the terms:

$H(n) = -\Sigma^{n!}_{i=1}\, (1/n!) \log_2(1/n!)$.

Simplifying the same terms, we will have:

$H(n) = -n! \cdot (1/n!) \log_2(1/n!)$.

Since:

$n! \cdot (1/n!) = 1$,

therefore:

$H(n) = -\log_2(1/n!)$.

Since in logarithms the property $\log(a^{-1}) = -\log(a)$ holds, and since $a^{-1} = 1/a$, we will have:

$-\log_2(1/a) = \log_2(a)$.

Setting $a = n!$, we obtain:

$H(n) = \log_2(n!)$.

But as the Fragments increase, the possible combinations also increase, so the reconstruction uncertainty ultimately increases.

Let's construct the limit:

$n \to \infty \Rightarrow H(n) \to +\infty$,

therefore:

$\lim_{n \to \infty} H(n) = \lim_{n \to \infty} \log_2(n!) = +\infty$.

Let's move on to using Bernoulli's distribution to randomly assign each individual Terminal Fragment and calculating the probability that, of all the n distributed Fragments, some will end up in the hands of the Colluding External Agents.

For the purposes of proving the theorem, we use Bernoulli's binomial distribution to calculate the probability that a number k of Fragments will be assigned to the Colluding External Agents. The value of k subsequently determines the reduction in the Entropy of the System.

So, let's first calculate the probability that, by chance, each individual Terminal Fragment will end up in the hands of a Colluding External Agent instead of an Honest External Agent.

When the System assigns this fragment, only two things can happen: either the fragment goes to a Colluding External Agent or it goes to an Honest External Agent.

There are no other possibilities.

Let's introduce the variable Y, which can take only two values.

Let's assume that:

Y = 1, if the Fragment randomly goes to the Colluding External Agent,

Y = 0, if the Fragment randomly goes to the Honest External Agent.

Let q be the probability that a single terminal Fragment ends up in the hands of a Colluding External Agent, and we have:

$P(Y = 1) = q$.

Since there are only two possibilities, the opposite probability is:

$P(Y = 0) = 1 - q$.

At this point, we use Bernoulli's Binomial Formula to calculate the probability of multiple Fragments:

$Y_1, Y_2, ..., Y_n$.

By adding them together, we have:

$X = Y_1 + Y_2 + ...+ Y_n$,

where X represents a random variable that counts the total number of Fragments assigned to the Colluding External Agents.

Suppose we have the following situation:

$Y_1 = 1, Y_2 = 0, Y_3 = 1, Y_4 = 0, Y_5 = 1$.

Then we have:

$X = 1 + 0 + 1 + 0 + 1 = 3,$

which means that out of 5 Fragments, 3 ended up in the hands of the Colluding External Agents.

The probability that exactly k terminal Fragments go to the Colluders is calculated with the following formula:

$P(X = k) = \text{binom}(n,k)\ q^{k}\ (1-q)^{(n-k)}.$

We must pay attention to the meaning of X and k.

X is the random variable that determines the number of Fragments assigned to the Colluder;

k corresponds to that specific number.

In fact:

$k \in \{0, 1, \ldots., n\}.$

Therefore:

X = random variable that counts the Fragments assigned to the Colluders;

k = the particular value assumed by the variable X.

Let's return to the proof.

We stated that the System Entropy corresponded to:

$H(n) = \log_2(n!)$,

which corresponds to the Uncertainty of the Entire System.

We should subtract a portion of these Fragments from the System Entropy calculation, because they will end up in the hands of the Colluders, reducing the Entropy.

The System Entropy is reduced because the Colluding External Agents, receiving the Fragments, will know their contents. Therefore a portion of the I_G Information will no longer be unknown and will no longer contribute to determining Uncertainty.

The number of Fragments still uncertain and unknown will be equal to $n - k$, those that ended up in the hands of the Honest External Agents.

The residual Entropy will be:

$H_{res}(k) = \log_2((n - k)!)$,

counting only the unknown Fragments that ended up in the hands of Honest External Agents.

The Entropy reduction will therefore be:

$\Delta H(k) = H(n) - H_{res}(k)$.

Substituting the values $H(n)$ and $H_{res}(k)$, we will have:

$\Delta H(k) = \log_2(n!) - \log_2((n - k)!)$.

Exploiting the property of logarithms that $\log(a) - \log(b) = \log(a/b)$, we obtain:

$\Delta H(k) = \log_2(n! / (n - k)!)$.

In summary:

- If $k = 0$, the colluders know nothing;
- If k increases, uncertainty decreases;
- If $k = n$, uncertainty becomes zero, because everything is known by the Colluding Agents.

So up to this point, Shannon Entropy provides a measure of the total uncertainty associated with the global information I_G.

The probabilistic model based on Bernoulli and binomial distributions estimates the likelihood that a sufficient number of useful fragments is compromised.

Each compromised fragment contributes to a reduction in global uncertainty according to its informational weight $\omega(D_i)$. The cumulative Entropy reduction is represented by ΔH.

The reconstruction threshold τ defines the minimum normalized amount of useful information that must be acquired in order to make unauthorized reconstruction possible.

If the reduction in System Entropy becomes large enough to exceed a Threshold Value, Collusion goes from being an unlikely event to a dangerous event. In the unweighted binomial baseline, this occurs when:

$k \geq \tau \cdot n$.

The probability that collusion leads to success, thus deteriorating the System's security, is therefore given by:

$P(X \geq \tau \cdot n)$.

Summary table:

$c_{coll} = k$ = number of Fragments assigned to the Colluders;

$c_{coll} / n = k / n$;

$k / n \geq \tau$;

$k \geq \tau \cdot n$;

$k_\tau = \lceil \tau \cdot n \rceil$;

$k \geq k_\tau$.

Starting from the Binomial Formula, we calculate the total probability of success of a collusion, which will therefore be given by the sum of all the probabilities of the cases in which the Colluders receive enough Fragments to reach or exceed the Critical Threshold.

The probability of success of the collusion will therefore be:

$P_{coll}(n) = P(X \geq k_\tau)$,

that is:

$P_{coll}(n) = \Sigma^{n}_{k=k\_\tau} P(X = k)$,

with $X = Y_1 + Y_2 + \ldots + Y_n$ representing the random variable referring to the total number of Fragments assigned to the Colluding External Agents, and k a specific number. We will have:

$k_\tau = \lceil \tau \cdot n \rceil$ = Critical Threshold value. **(104)**

Substituting we will have:

$P_{coll}(n) = \Sigma^{n}_{k=\lceil \tau \cdot n \rceil} \text{binom}(n, k)\, q^k\, (1-q)^{(n-k)}$. **(105)**

The formula to be proved was:

$q\bar{\omega} < \tau \Longrightarrow \lim_{n \to \infty} P[\Sigma^{n}_{i=1} Y_i \omega (D_i) \geq \tau n] = 0$.

In the unweighted binomial baseline, the corresponding formula is:

$q < \tau \Longrightarrow \lim_{n \to \infty} P_{coll}(n) = 0$. **(106)**

with:

$X \sim \text{Bin}(n, q)$,

that is, X is the variable of the Binomial, which counts how many Fragments end up with the Colluders;

n is the total number of Fragments;

q is the probability of success of a single Bernoulli Trial, i.e.:

$q = P(Y_i = 1)$,

which is the probability that a single Fragment will be randomly assigned to the Colluder.

The expected value is:

$\mathbb{E}[X] = n \cdot q$,

so the Colluders receive on average $n \cdot q$ Fragments.

The minimum threshold is:

$k = k_\tau = \lceil \tau \cdot n \rceil$.

But if:

$q < \tau$,

then:

$n \cdot q < \tau \cdot n$,

that is, the average number of Fragments received by the Colluders is less than the threshold required to violate the System.

Therefore:

X / n

will be the fraction of the total Fragments that end up with the Colluders.

Collusion is therefore successful if:

$X / n \geq \tau$.

So substituting we have:

$P_{coll}(n) = P(X / n \geq \tau)$.

By the law of large numbers, if $n \to \infty$, we have:

$X / n \to q$,

that is, if the number of Fragments grows infinitely, the effective and real fraction of Fragments assigned to the Colluders increasingly approaches Probability q, which is the probability that a single fragment is assigned to a Colluder.

But if the real fraction approaches q, and q is smaller than τ, then it becomes increasingly unlikely that:

$X / n \geq \tau$.

So substituting we have:

$q < \tau \Longrightarrow \lim_{n \to \infty} P(X / n \geq \tau) = \lim_{n \to \infty} P_{coll}(n) = 0.$

Therefore:

$q < \tau \Longrightarrow \lim_{n \to \infty} P(X \geq \lceil \tau \cdot n \rceil) = 0.$

By replacing we will have:

$q < \tau \Longrightarrow \lim_{n \to \infty} \Sigma^{n}_{k=\lceil \tau \cdot n \rceil} \text{binom}(n, k)\, q^{k}\, (1-q)^{(n-k)} = 0.$

That is, if $q\bar{\omega} < \tau$, where q is the probability that a fragment is assigned to a colluding verifier and $\bar{\omega}$ is the average informational weight of the fragments, then increasing the number of fragments makes it increasingly unlikely that the colluders obtain enough useful information to exceed the reconstruction threshold.

Let's return to Kullback-Leibler divergence, which we saw at the beginning of this chapter:

$D(\tau \| q) = \tau \log(\tau/q) + (1-\tau) \log((1-\tau)/ (1-q)).$ **(107)**

Here log denotes the natural logarithm when $D(\tau \| q)$ is used in exponential concentration bounds.

D measures how far the actual situation q is from the critical level τ, that is, from the situation in which the Critical Threshold level for assigning Fragments to Colluders is actually reached.

Equivalently, it measures how unlikely it is that a binomial variable with actual probability q will produce an observed frequency equal to at least τ.

For a binomial, the Chernoff Inequality holds, which is its classic application.

When the number of independent trials, corresponding here to Fragments, is very high, calculating the Binomial exactly becomes complex.

The Chernoff Inequality allows us to obtain an exponential upper bound on the probability of a la rge deviation from the expected value.

For example, for a binomial variable X with mean μ, the following standard bounds hold:

$$P(X \geq (1 + \delta)\mu) \leq \exp(-\delta^2\mu / 3),$$

$$P(X \leq (1 - \delta)\mu) \leq \exp(-\delta^2\mu / 2). \quad \textbf{(108)}$$

That is, the further one moves from the mean, and the more one increases the number of trials, the more the probability of an anomalous event collapses towards zero.

Applied to the upper tail of the binomial distribution, the Chernoff-KL bound gives:

$$P(X / n \geq \tau) \leq \exp(-nD(\tau \| q)), \quad (108)$$

if $\tau > q$.

Since:

$P_{coll}(n) = P(X / n \geq \tau)$,

we can also write:

$$P_{coll}(n) \leq \exp(-nD(\tau \| q)), \quad \textbf{(109)}$$

that is, the probability that the colluders exceed the critical threshold drops exponentially the greater the number of trials, i.e. Fragments, and the greater the distance between q and τ calculated with the Kullback-Leibler divergence.

Let: $Z = D(\tau \| q)$,

so as to simplify writing the formula.

If: $Z > 0$,

then:

$P_{coll}(n) \leq \exp(-nZ)$.

As n increases, the exponent becomes more and more negative:

assuming $Z = 0.1$,

$n = 10 \rightarrow \exp(-1)$,

$n = 100 \rightarrow \exp(-10)$,

$n = 10000 \rightarrow \exp(-1000)$.

The value will tend exponentially to 0.

Since $Z > 0$, we have:

$\lim_{n \to \infty} \exp(-nZ) = 0.$

Since:

$0 \leq P_{coll}(n) \leq \exp(-nZ),$

and since:

$\lim_{n \to \infty} \exp(-nZ) = 0,$

therefore:

$P_{coll}(n) \to 0$

for $n \to \infty$, concluding:

$q < \tau \Longrightarrow \lim_{n \to \infty} P_{coll}(n) = 0.$

In reality, not all fragments have the same weight.

Let:

$W_n = \Sigma^{n}_{i=1} Y_i \, \omega(D_i)$

be the total weighted useful information controlled by the colluding agents.

Thus, the event:

$W_n \geq \tau n$

means that the colluding agents have obtained enough weighted useful information to reach or exceed the critical reconstruction threshold.

Some fragments are more useful than others because they are better connected and because they convey a greater amount of information about the Global Information.

Let's formalize the concept.

A fragment is more useful the more it represents the Global Information, i.e., the greater its contribution to Global knowledge.

Each useful fragment therefore reduces the Global Entropy of the System, meaning that the more useful, i.e. heavier, a fragment is, the more it reduces the Uncertainty of the Entire System.

Let: G

be the random variable representing the distribution of the System's globally admissible configurations.

Let:

$u(D_i) = I(G; F_i)$ **(110)**

be the unnormalized informational utility of fragment F_i with respect to the global admissible configuration G.

Equivalently:

$$u(D_i) = H(G) - H(G \mid F_i). \quad \textbf{(111)}$$

Thus:

$$0 \leq u(D_i) \leq H(G).$$

For subsequent probabilistic bounds, we use the normalized weight:

$$\omega(D_i) = u(D_i) / H(G) = I(G; F_i) / H(G), \quad \textbf{(112)}$$

assuming $H(G) > 0$.

Therefore:

$$0 \leq \omega(D_i) \leq 1.$$

If its value is equal to 0, the Fragment is informationally useless for global reconstruction.

If its value is equal to 1, the Fragment carries maximal normalized informational utility with respect to the Global admissible configuration.

Intermediate values have proportional utility.

Since:

$0 \le \omega(D_i) \le 1,$

and:

$Y_i \in \{0, 1\},$

we have:

$0 \le Y_i\omega(D_i) \le 1.$

Moreover, assuming independent assignment with:

$P(Y_i = 1) = q,$

the expected useful weight controlled by colluders is:

$$\mathbb{E}[W_n] = \mathbb{E}[\Sigma^n_{i=1}\ Y_i\omega(D_i)]$$

$$= q\ \Sigma^n_{i=1}\ \omega(D_i)$$

$$= nq\bar{\omega}, \qquad \textbf{(113)}$$

where:

$$\bar{\omega} = (1/\ n)\ \Sigma^n_{i=1}\ \omega(D_i). \qquad \textbf{(114)}$$

If:

$q\bar{\omega} < \tau,$

then there exists:

$\delta = \tau - q\bar{\omega} > 0.$

For the asymptotic statement, this gap is assumed to remain positive as n grows.

Therefore:

$P[\Sigma^{n}_{i=1} Y_i \omega(D_i) \geq \tau n]$

$= P[W_n \geq \tau n]$

$= P[W_n - \mathbb{E}[W_n] \geq n \delta].$

Since the random variables $Y_i \omega(D_i)$ are independent and bounded, a Chernoff-Hoeffding bound gives:

$P[W_n \geq \tau n]$

$\leq \exp(-2n^2 \delta^2 / \Sigma^{n}_{i=1} \omega(D_i)^2).$

Because:

$0 \leq \omega(D_i) \leq 1,$

we have:

$\Sigma^{n}_{i=1} \omega(D_i)^2 \leq n.$

Thus:

$P[W_n \geq \tau n]$

$\leq \exp(-2\, n\, (\tau - q\bar{\omega})^2)$.

Consequently:

$q\bar{\omega} < \tau \Longrightarrow \lim_{n \to \infty} P[\Sigma^{n}_{i=1}\, Y_i \omega\, (D_i) \geq \tau n] = 0$.

Therefore, if the expected weighted useful information controlled by colluders remains below the critical reconstruction threshold, the probability that colluders obtain enough useful information to compromise the System tends to zero as the number of terminal fragments increases.

---

16. Considerations on Dependent Collusion

16.1 Introduction

The Chernoff exponential bound introduced in the previous chapter models unauthorized reconstruction by simplifying the assumption that fragment compromise events are statistically independent and, consequently, compromise events are treated as statistically independent. This approximation is useful for illustrating the asymptotic scalability of security, but it does not explicitly capture coordinated collusion between adversarial verifiers.

In the real world, collusions are more often coordinated and, being able to share all the information they acquire, organize their strategies to compromise the System adaptively, that is, by targeting the most informative fragments. A more general formulation must therefore account for the statistical dependence between compromise events.

The purpose of this chapter is to demonstrate that the central conclusion of the CNVS security model remains valid even in the presence of coordinated collusion, provided that the conditional probability of compromising each additional critical fragment is uniformly bounded by a constant strictly less than one.

16.2 Development

Let:

- Q denotes the total number of verifiers;

- c_{coll} denotes the total number of colluding verifiers;

- $q = c_{coll}/ Q$ denote the collusive control fraction;

- p_{inf} indicates the probability that a critical fragment not directly controlled by collusive entities can still be inferred from compromised information;

- m indicates the number of critical fragments with low inference required for unauthorized reconstruction.

A fragment is considered critical if, having low inference, it cannot be easily inferred from knowledge of all the others, therefore its knowledge becomes critical for reconstruction.
We define the probability of compromise of a single fragment as

$$p_{comp}=q + (1−q)\, p_{inf}. \qquad \textbf{(115)}$$

This quantity represents the probability that a single critical fragment is either directly controlled by colluding entities or successfully inferred.

For each critical fragment F_i, let C_{comp_i} denote the event that F_i is compromised.

We assume that, for every critical fragment,

$$P(C_{comp\_i} \mid C_{comp\_1},\ldots,C_{comp\_i-1}) \leq p_{comp}, \quad 0 \leq p_{comp} < 1. \qquad \textbf{(116)}$$

This condition does not assume independence. It only requires that, even after any sequence of prior compromises, the probability of compromising one additional critical fragment remains bounded by a constant strictly smaller than one.

It is clear that this assumption becomes a fundamental prerequisite for guaranteeing Security, because if the probability of compromising the residual critical fragments not yet controlled by the Colluders were equal to 1, it would mean that the System has been completely violated: in this case we would no longer have a probability, but the certainty that the residual Information is completely accessible to the Colluders or in any case deducible.

Let Rec^* denote the event of unauthorized full reconstruction of the global informational state.

Under the stated assumptions, unauthorized reconstruction is bounded by

$$P(Rec^*) \leq p_{comp}^{\,m} \qquad \textbf{(117)}$$

therefore:

unauthorized reconstruction requires compromise of all m critical fragments:

$$Rec^* \subseteq C_{comp\_1} \cap C_{comp\_2} \cap \cdots \cap C_{comp\_m}.$$

By the chain rule of probability,

$$P(C_{comp\_1} \cap C_{comp\_2} \cap \ldots \cap C_{comp\_m}) = \prod_{i=1}^{m} P(C_{comp\_i} \mid C_{comp\_1}, \ldots, C_{compi-1}).$$

$$P(Rec^*) \leq \prod_{i=1}^{m} P(C_{comp\_i} \mid C_{comp\_1}, \ldots, C_{comp\_i-1})$$

Applying the uniform conditional bound,

$$P(Rec^*) \leq \prod_{i=1}^{m} p_{comp} = p_{comp}^{\,m}. \qquad \textbf{(118)}$$

If

$0 \leq p_{comp} < 1$,

then

$P(Rec^{*}) \to 0$ as $m \to \infty$.

This establishes that the essential conclusion of the Emergent Security Scaling Theorem remains valid even when colluding adversaries are fully coordinated.

At this point, it is necessary to make an important clarification.

As explained in previous chapters, the number of fragments cannot tend to infinity, because with each recursive decomposition, the semantic content of each fragment is reduced to zero.
This too can be understood as an approximation.

Elsewhere in this document, the limit I_{min} (or I_{Fmin}) denotes the semantic lower bound below which a terminal fragment no longer contains enough information for honest verification.
It is sufficient for the decomposition to asymptotically approach that limit, so as to disentangle all I_F fragments from any externally inferable local and global structural relationship, keeping the colluder blind both locally and globally, even if it shares its information with that of other colluders.
Therefore,
security increases with the number of critical fragments up to the limit imposed by the minimum semantic threshold I_{min}.

So to recap, the asymptotic expressions derived in the previous sections, including

$P(Rec^{*}) \to 0 \quad \text{as} \quad m \to \infty$

and

$q\bar{\omega} < \tau \implies \lim n \to \infty \; P\left[\Sigma(i=1 \text{ to } n)\, Y_i\, \omega(Di) \geq \tau n \right] = 0$

should be interpreted as idealized scaling limits:

these results establish the theoretical direction of the security model, therefore under the stated assumptions, increasing the number of sufficiently

independent low-inference fragments reduces the probability of unauthorized reconstruction.

In the weighted formulation, $q\bar{\omega}$ represents the expected useful adversarial information per fragment. Therefore, under the idealized average-weight approximation, the condition $q\bar{\omega} < \tau$ expresses that the expected useful adversarial information remains below the reconstruction threshold.

(Technical note. the rigorous weighted version is provided in the compact formal synthesis through a bounded weighted concentration inequality).

It is the task of the System, or rather of its implementation, depending on the nature of the Information, to make the terminal Fragments increasingly independent and reduce their inference to a minimum.

Formally,

let

I_{min}

denote the minimum amount of information required for semantic task comprehension.

Each terminal fragment F_i must satisfy $I_F \geq I_{min}$.

If this condition is violated, the fragment becomes operationally meaningless and local verification is no longer possible.

The semantic lower bound imposes a practical upper limit on fragmentation depth.

Let

m_{max}

denote the maximum number of critical fragments that can be generated while preserving

$|I_{F,i}| \geq I_{min}$.

This quantity depends on:

- the total informational content of the global state;

- the semantic complexity of each verification task;

- the decomposition strategy;

- the informational heterogeneity of fragments.

From the generalized dependent-collusion model, the minimum number of critical low-inference fragments required to achieve a target security level

$P(Rec^*) \leq \eta$

$m_{min} = \lceil \ln(\eta) / \ln(p_{comp}) \rceil,$ **(119)**

for

$0 < p_{comp} < 1.$

With respect to the target security level η and the semantic lower bound I_{min}, a CNVS architecture is practically feasible only if:

$m_{max} \geq m_{min}$.

When this condition holds, the System can generate enough critical low-inference fragments to achieve the desired security level without violating the semantic lower bound required for honest verification.

The condition above transforms the asymptotic security theory into a concrete design criterion.

The practical engineering objective is therefore:

maximize epistemic restriction subject to semantic verifiability.

In other words, the System should fragment information as aggressively as possible while ensuring that each terminal fragment remains sufficiently meaningful for the assigned verification task.

In that case, security can still be improved by reducing

p_{comp},

for example by:

- decreasing the fraction of colluding verifiers q;

- reducing the inference probability p_{inf};

- increasing fragment heterogeneity;

- strengthening the non-inferability of critical fragments.

This feasibility condition does not weaken the CNVS theory.

On the contrary, it explicitly links:

- semantic constraints;

- probabilistic security;

- System scalability;

- practical implementation.

As a result, the CNVS framework provides not only asymptotic theorems, but also a concrete quantitative criterion for determining whether a proposed architecture can simultaneously satisfy operational and security requirements.

If critical fragments are non-inferable, i.e.,

$p_{inf}=0$

then

$p_{comp}=q$

and therefore

$$P(Rec^{*}) \le q^{m} \quad \textbf{(120)}$$

This corresponds to the ideal case in which each non-inferable critical fragment acts as a veto condition for unauthorized reconstruction.

Suppose a target upper bound η is specified for the probability of unauthorized reconstruction:

$P(Rec^{*}) \le \eta$

The minimum number of critical low-inference fragments required to satisfy this condition is

$m_{min} = \lceil \ln(\eta) / \ln(p_{comp}) \rceil$

and where

η is the maximum tolerated probability coefficient of unauthorized reconstruction,

with $\eta \in (0, 1)$,

let $0 < p_{comp} < 1$

the ratio $\ln(\eta) / \ln(p_{comp})$

is well defined and strictly positive,

The security condition

$p_{comp}^{\ m} \leq \eta$

holds if and only if

$m * \ln(p_{comp}) \leq \ln(\eta)$

because $\ln(p_{comp}) < 0$, division by $\ln(p_{comp})$ reverses the inequality:

$m \geq \ln(\eta) / \ln(p_{comp})$.

This expression provides a direct design rule linking security objectives to architectural parameters.

Example.

Assume:

- $q = 0.10$,
- $p_{inf} = 0.01$

Then:

$p_{comp} = 0.10 + 0.90 \times 0.01 = 0.109$

If:

- m = 20

then:

$P(Rec^*) \leq 0.109^{20} \approx 5.7 \times 10^{-20}$.

Thus, even with coordinated collusion and nonzero inference probability, unauthorized reconstruction becomes negligibly probable when a sufficient number of critical fragments remains protected.

This chapter generalizes the original Chernoff-based independent-assignment analysis by replacing the independence assumption with a uniform conditional compromise bound. The result shows that exponential security scaling is not an artifact of independent Bernoulli models, but a structural consequence of maintaining a uniform upper bound on the conditional probability of compromising each additional critical fragment.

In practical terms, the Security Theorem, applied to cases of Dependent Collusion, states that if the System architecture ensures that each new critical fragment remains sufficiently difficult to compromise or infer, then overall security increases exponentially with the number of such fragments, even in the presence of a coordinated attack.

16.3 Relations with Inference

It should be noted, however, that the inference probability p_{inf} can also be interpreted as a function of the residual uncertainty of the global state.

The inference probability was already introduced in the previous section and was utilized to formalize the compromise probability of an additional fragment according to the equation:

$p_{comp} = q + (1 - q)\, p_{inf}$

However, its mathematical significance has not yet been thoroughly examined. Let us examine the details.

Let:

$W(t) = \Sigma_{i=1^n} Y_i(t)\, \omega(D_i)$

be the normalized informational weight controlled by the adversarial coalition at time t, where $Y_i(t)=1$ if fragment F_i is controlled or compromised by the coalition, and $Y_i(t)=0$ otherwise.

Thus, W(t) represents the cumulative normalized useful knowledge acquired by the adversarial coalition relative to the System.

When the Adversary is himself a legitimate Actor within the System, his directly accessible knowledge is normally restricted to the semantic information plane associated with the assigned Fragment(s). Any attempt to reconstruct the global state of the System consequently requires inferential access to the higher informational plane, which contains the relational topology, reconstruction dependencies, and invariant coordination structures.

Indeed, due to the information fragmentation enforced by the System, the Adversary may possess all locally accessible payload fragments while remaining incapable of reconstructing the global semantic structure, unless the analysis of W(t) also infers the topological and relational organization linking these fragments and uncovers the systemic invariant constraints. These constraints evidently preclude the deduction of the missing payload contents as a function of the invariants to be satisfied. Furthermore, even with subtotal access to semantic and relational information, the deduction of missing payloads would be obstructed by the adversarial coalition's lack of knowledge regarding the global invariants.

It must be noted that fragments not controlled by the Adversaries remain inaccessible due to the combination of informational separation, distributed allocation, and cryptographic protection mechanisms.

Consequently, effective reconstruction depends not merely on acquiring the semantic payload information, but also on retrieving the structural informational organization represented by the higher informational plane.

In summary, the adversarial coalition disposes of two methods to acquire knowledge of the missing fragments: either it attempts to deduce the payload of the assigned fragment as a function of the task entrusted by the System, or it attempts to infer the payload of missing fragments that it does not directly manage.

The inference probability depends strictly on the degree of System Entropy. Letting the Shannon conditional entropy be the amount of average residual uncertainty remaining in the System after the adversarial coalition has acquired its view, we can express the fundamental relation as:

$H(S(t_0)| \text{View}_{adv}) = H(S(t_0)) - I(S(t_0) ; \text{View}_{adv})$

where $H(S(t_0))$ corresponds to the initial Global Entropy of the System at time 0 (i.e., when adversarial knowledge is zero) and View_{adv} corresponds to the total knowledge acquired by the colluding coalition. The term $I(S(t_0) ; \text{View}_{adv})$ corresponds to the Mutual Information, representing the amount of information regarding the global state that the adversarial coalition can deduce once View_{adv} is known.

Evidently, the average residual uncertainty is given by the total entropy of the System minus the information that the adversarial coalition can directly deduce from the information already at its disposal. In principle, it is possible to assume a minimum quantity of Entropy that the system must preserve to maintain its security, which we may define as h_{min}.

Indeed, to guarantee worst-case cryptographic security, we do not rely on average entropy, but rather on **Min-Entropy** (H). Min-Entropy corresponds to the minimum entropy (i.e., the maximum vulnerability or the configuration most favorable to the attacker) present within the System.

In principle, it is possible to enforce that the residual Min-Entropy of the missing critical fragment d_{miss} never drops below a minimum threshold $h_{min} > 0$:

$H_{\infty}(d_{miss} | \ View_{adv}) \geq h_{min}$

This quantity mandates that the security of the System is preserved because the final critical fragment will never be 100% deducible. Theoretically, it is sufficient for this value to be strictly greater than 0. If $h_{min} = 0$, it implies that the System contains no further uncertainty and that all information can be deterministically deduced by the adversarial coalition.

Assuming therefore that $h_{min} > 0$, we can conclude that the total reconstruction of the global information will remain impossible as long as the inference probability p_{inf} of the missing payload, however minimal, remains strictly less than 1. This is achieved when the adversarial coalition's access to critical information is constrained below the system's breaking threshold.

The security of the System is therefore not entrusted to the indiscriminate limitation of adversarial access to a number of equations equal to the number of unknowns; rather, it is strictly bound to inferential deduction constraints. As long as there exists a critical mass of information that is neither directly accessible to the adversarial coalition nor deducible, the unauthorized total reconstruction of the global semantic payload remains impossible, as a minimum residual entropy greater than zero will always persist.

Formally, the inference probability for the missing critical fragment satisfies:

$p_{inf} \leq 2^{-h\_min}$

for $h_{min} > 0$ and $p_{inf} < 1$.

Consequently, for $q < 1$, we obtain:

$p_{comp} = q + (1 - q)\, p_{inf} < 1$.

At an interpretative level, p_{inf} may also be associated with an entropy-erosion proxy over time:

$p_{inf}(t) \triangleq 1 - \mathbb{E}[H(S(t) \mid W(t))] / H(S(t_0))$.

where $\mathbb{E}[H(S(t)|W(t))]$ corresponds to the expected value of the conditional entropy of the System at time t, given the cumulative adversarial knowledge W(t), while $H(S(t_0))$ is the initial global entropy of the System at time 0.

This expression should be understood as a normalized entropy-leakage indicator, not as a replacement for the min-entropy bound governing worst-case inference.

Therefore:

$p_{comp}(t) = q + (1 - q)[1 - \mathbb{E}[H(S(t) \mid W(t))] / H(S(t_0))]$.

Finally, extending the analysis to the entire vector of missing payloads (D_{miss}), we obtain the definitive upper bound for unauthorized global reconstruction:

$P(Rec^*) \leq 2^{-H_\infty (D\_miss \mid View\_adv)}$

where the reconstruction probability is bounded above by the exponential inverse of the residual conditional min-entropy of the total missing payloads D_{miss} relative to the adversarial view $View_{adv}$.

This formulation organically links the dependent-collusion model with the entropy-based interpretation of fragment utility. It does not replace the generalized security bound, but rather provides a profound informational interpretation of the inference term.

16.4 Conclusion

The original emergent security scaling theorem established the exponential decay of the unauthorized reconstruction probability within an idealized independent-assignment model. The present formulation extends this identical principle to coordinated colluding adversaries and dependent compromise events.

Consequently, this theorem should not be interpreted as a fundamentally separate principle, but rather as a generalized formulation of the original security scaling result. The generalized dependent-collusion model reinforces the CNVS framework by demonstrating that emergent security persists even in the presence of coordinated collusion and statistical dependence.

The essential requirement is that the conditional compromise probability of each additional critical fragment remains uniformly bounded by a constant p_{comp} strictly less than one ($p_{comp} < 1$). Under these conditions, the unauthorized reconstruction probability decreases exponentially as the number m of critical fragments generated by the expansion of the system's fragmentation scale increases. The foundational security conclusions of the CNVS therefore remain fully valid in significantly more complex and realistic adversarial environments.

17. Scientific Positioning and Relationship with Existing Frameworks

17.1 Introduction

The **CNVS framework** did not emerge in isolation. It sits at the intersection of several established disciplines, including information theory, cryptography, distributed Systems, epistemic logic, and formal methods.
The purpose of this chapter is not to assert superiority over existing frameworks, but rather to precisely identify the similarities and substantial differences between CNVS and the main theoretical models already present in the literature.
CNVS is proposed as an axiomatic framework for distributed and convergent verification-dependent state admission, subject to knowledge restriction constraints, global invariants, and emergent security with scale.

17.2 Information Theory

Conceptual Relationship

Information theory provides the mathematical foundation for:
• entropy;
• uncertainty;
• information content;
• redundancy;
• communication limits.
CNVS uses these concepts to describe how the fragmentation and restriction of knowledge reduce the likelihood of unauthorized reconstruction.

Fundamental Difference

Information theory addresses the issue of the amount of information that can be encoded and its subsequent transmission and compression.

CNVS, on the other hand, addresses the issue of the validity of the Global State and the conditions for this to be possible.
CNVS is not a theory of communication or encoding, but is first and foremost a theory of the existence of states as dependent on verification.

17.3 Verifiable Secret Sharing and Threshold Encryption

Conceptual Relationship

Verifiable Secret Sharing (VSS) and threshold cryptography distribute shares of a secret among multiple participants and allow the secret to be reconstructed once a minimum threshold is reached.
CNVS also uses:
• fragmentation;
• threshold-based reasoning;
• partial information.

Fundamental Difference

VSS addresses the need to securely manage the distribution and reconstruction of a partially shared secret, but without a verification architecture based on distributed convergence and restricted informational access.

CNVS addresses the problem of establishing the validity of a Global State while preventing unauthorized reconstruction. To do so, it uses External Verification Agents that are not included in the VSS architecture. It also performs verification through the convergence between the internal data D_i and the external observation Obs_i, with V_{Local} acting as a local admissibility operator, accessing the content needed for its local task at each level, but without direct semantic access to the informational payload preserved in D_i.
The architecture, purpose, and structure are completely different.

In CNVS, fragmentation is not primarily a mechanism for sharing secrets, but a structural mechanism for limiting knowledge, distributing verification, and limiting the inference of Local knowledge from Global knowledge.

17.4 Zero-Knowledge Proofs

Conceptual Relationship

Zero-Knowledge Proofs allow a subject to prove the truth of a proposition without revealing the underlying secret.
CNVS shares the principle of limiting the information accessible to individual agents.

Fundamental Difference

Zero-Knowledge Proofs (ZKP) address the problem of how to prove a truth without revealing a secret. Here, too, there are no independent External Verifiers, nor is verification based on data convergence.
The prover possesses a witness (such as a password, a solution, or a private key) and demonstrates that it is able to satisfy a certain mathematical relation $R(x, w)=1$,
where:
- x = public statement;
- w = secret witness;
- R = verifiable relation.

The prover convinces the verifier that there exists a w such that $R(x, w)=1$, without revealing w.

CNVS addresses the completely different problem of determining how to allow multiple agents with limited local knowledge to analyze the results of a sampling process that has the sole purpose of determining the admissibility of a Global State. Verification does not occur through cryptographic "proofs" or attestations, but through the study of the Object itself.
ZKP does not study the information content of the Object, not even at the

protocol level, because it is irrelevant to the purpose.
ZKP is a cryptographic proof protocol.
CNVS is not a prover-verifier protocol, although it shares the principle of information control.

17.5 Secure Multiparty Computation

Conceptual Relationship

Secure Multiparty Computation (MPC) allows multiple parties to compute a function on private inputs while preserving privacy and correctness.
Architecturally, MPC is the framework that may seem closest to CNVS, but it differs radically.

Fundamental Difference

MPC is a model that computes a function $f(x_1,\ldots,x_n)$ without revealing the private inputs, which share its use for a common goal.

CNVS asks only when a Global State S can be accepted as valid.
Even if MPC were used to compare local results D_i, it would not independently provide:

- Verification-Dependent Existence;
- Non-Reducibility of Global Validity;
- Knowledge Restriction Principle;
- Global Veto Principle;
- Emergent Security with Scale.

MPC can therefore be an implementation tool, but it is not equivalent to CNVS.
This is because the MPC model does not perform Verification by convergence of information and data D_i, and does not apply System invariants.
Specifically, MPC does not define a verification architecture based on independent local outputs D_i that converge towards a reconstructed global structured informational state $\mathtt{I}_G$.
In CNVS, the convergence of distributed evidence is a central structural

mechanism. Multiple verifiers with limited knowledge produce local results, and the consistency of these results determines whether a global state can be admitted.

Furthermore, the System does not operate primarily through metadata exchange or shared computation, but through the preservation and verification of relational consistency across recursively fragmented structured elements.

MPC distributes private inputs among participants, but does not generally require the recursive decomposition of a single global information state into semantically distinct fragments.
Secure Multiparty Computation (MPC) is not designed to fragment a single global piece of information.
It is designed to allow multiple parties, each with their own independent private input, to jointly compute a function.

CNVS, on the other hand, is explicitly based on the non-uniform recursive fragmentation of a global state. This fragmentation constitutes a fundamental architectural and security principle.
MPC protects private inputs from disclosure during computation. However, its primary goal is the preservation of privacy, not the structural limitation of knowledge as a foundational principle of security.
In CNVS, the limited visibility of fragments is not simply a privacy feature, but a fundamental element of the security model, where the Verifier is not granted access to the elementary data D_i, but is instead tasked with performing a measurement procedure that results in a comparison entity with a pre-existing data point.
MPC does not inherently require fragments to have different informational relevance, nor that they be assigned via a non-uniform random distribution.
MPC does not contain the CNVS-specific notion of hidden internal data D_i, external observations Obs_i, or verification instances assigned to limited-knowledge External Verification Agents.

CNVS explicitly assumes that fragments can possess heterogeneous informational value and that they are randomly and non-uniformly distributed.
MPC guarantees the correct computation of a function.
CNVS determines whether a candidate Global State satisfies structural invariants and can therefore be admitted as valid. Correctness of the computation alone is not sufficient; a separate verification of global validity is required.
MPC is a distributed computation protocol.
CNVS is a Verification System.

17.6 Byzantine Fault Tolerance and Consensus Protocols

Conceptual Relationship

Byzantine Fault Tolerance (BFT) protocols address the problem of reaching distributed agreement in the presence of malicious actors.
CNVS also considers distributed verifiers and adversarial scenarios.

Fundamental Difference

Consensus protocols ask how multiple nodes can reach a collective agreement.
CNVS asks how a state can be admitted only if the global invariants are satisfied.
In CNVS, agreement is not sufficient, because a state is rejected whenever the global invariants are not respected.
The fragmentation and random distribution of information reduce the possibility of informationally effective agreements between colluders.

17.7 Bitcoin and Blockchain Security

Conceptual Relationship

Bitcoin has shown that large-scale decentralized Systems can achieve probabilistic security that increases with the number of participants.
CNVS similarly formalizes security that increases with the scale of the System.

Fundamental Difference

Bitcoin is based on:

- proof-of-work;
- economic incentives;
- longest chain rule.

CNVS, on the other hand, is based on:

- fragmentation;
- knowledge restriction;
- non-inferability;
- global veto;
- state admission based on invariants.

CNVS is therefore conceptually distinct from blockchain consensus.
Neither consensus nor a quorum guarantees the transition of a state request from candidate to existing, but only the application of System invariants and the satisfaction of verification functions at the local and global levels.

17.8 Epistemic Logic

Conceptual Relationship

Epistemic logic studies what agents know and how that knowledge evolves.
CNVS has a central epistemic component, as it explicitly limits verifiers' local knowledge.

Fundamental Difference

Epistemic logic is primarily descriptive and formal.
CNVS uses epistemic constraints as an operational security and verification mechanism.

17.9 Formal Methods

Conceptual Relationship

Formal methods provide mathematical tools for specifying and verifying Systems.
The formalization of CNVS in Lean 4 belongs to this methodological tradition.

Fundamental Difference

Formal methods are not the object of CNVS, but rather tools used to verify its internal logical consistency.
Formal methods verify the model's consistency; CNVS uses this consistency to define procedures for admitting or rejecting states.

17.10 Complex Systems and Emergence

Conceptual Relationship

Complex Systems theory studies the emergence of global properties from local interactions.
CNVS shares this perspective through the concept of emergent security.

Fundamental Difference

CNVS does not simply observe emergence, but derives explicit structural and probabilistic conditions that guarantee increased security with scale.

17.11 Quick Comparative Summary

Framework	Primary Objective	Relationship with CNVS
Information Theory	Communication and entropy	Mathematical foundation
VSS / Threshold encryption	Distribution and reconstruction of secrets	Shares the fragmentation
Zero-Knowledge Proofs	Demonstration without revelation	Shares the limitation of information
Secure Multiparty Computation	Private distributed computing	Architecturally closer framework, but distinct in terms of Di convergence, fragmentation, invariants, and global verification.
Byzantine Fault Tolerance	Distributed Agreement	Shares the adversarial context
Bitcoin	Probabilistic decentralized consensus	Shares security that increases with scale
Epistemic Logic	Reasoning about knowledge	Conceptual foundation
Formal Methods	Mathematical verification	Validation tool
Complex Systems	Global Property Emergence	Supports emergent interpretation

---

17.12 Distinctive Contribution of the CNVS

The distinctive contribution of CNVS lies in the unified integration of the following principles:

1. Verification-Dependent Existence;
2. Non-Reducibility of Global Validity;
3. Non-Uniform Recursive Fragmentation;
4. Knowledge Restriction Principle;
5. Global Veto Principle (i.e., rejection of a state when a structurally critical convergence condition or global invariant is violated);
6. Emergent Security with Scale.

No single existing framework combines these principles with the specific goal of admitting global states in distributed Systems.

17.13 Final Positioning Declaration

CNVS should therefore be interpreted as an axiomatic framework proposed for distributed verification and state admission.
It integrates concepts from information theory, cryptography, distributed Systems, and epistemic reasoning, but addresses a distinct theoretical question:
Under what conditions can a globally consistent state be admitted in a distributed System with limited local knowledge?
This question defines the original core and specific theoretical field of the CNVS framework.

AI Use Statement

This work was developed and edited by Massimo Comitato.
In preparing this document, artificial intelligence tools were used as editorial and technical assistants. Their use included support for text formatting, linguistic refinement, structural organization, consistency checking, notation review, and assistance in presenting mathematical arguments in a clearer, academic form.
The theoretical framework of Closed Native Verification Systems (CNVS), including its basic conceptual structure, axiomatic organization, mathematical orientation, all definitions, terminology, system architecture, and the formulation of key theoretical claims, was conceived and developed by the author.
AI tools were used as auxiliary tools to refine, organize, test, and express the author's ideas. The final selection of concepts, formulas, definitions, hypotheses, interpretations, and scientific claims remains the sole responsibility of the author.

Any remaining errors, omissions, ambiguities or limitations are attributable solely to the author.

Appendices

Appendix Q – Tables λ

CNVS Fragmentation Tables Based on Ter

Progressive variation of λ and Composite Information Density ρ_C

Formula and simplifying assumptions

Formula (86): $\text{Fragments}(t) = \lceil \text{Ter}(t, \xi) \rceil = \lceil \lambda \cdot [1/(\rho_C(t) \cdot (N(t)^{\beta_{-N}}))] \cdot \log_2(1 + (I_G(t) + I_N(t))/I_0) \rceil$

Assumptions used in these tables: $I_N = I_G$; I_0 = 100,000 bits; $N(t) = 1$, so $N(t)^{\beta_{-N}} = 1$. Values are rounded upward with the ceiling function.

Interpretation: λ acts as a fragmentation-intensity multiplier; lower ρ_C thresholds impose stronger dispersion and therefore generate a larger number of terminal Fragments.

Table — Composite Information Density $\rho_C = 0.30$

$I_G = I_N$	λ=1	λ=2	λ=3	λ=4	λ=5	λ=6	λ=7	λ=8	λ=9	λ=10	λ=11	λ=12
100 kbit	6	11	16	22	27	32	37	43	48	53	59	64
200 kbit	8	16	24	31	39	47	55	62	70	78	86	93
300 kbit	10	19	29	38	47	57	66	75	85	94	103	113
400 kbit	11	22	32	43	53	64	74	85	96	106	117	127
500 kbit	12	24	35	47	58	70	81	93	104	116	127	139
600 kbit	13	25	38	50	62	75	87	99	112	124	136	149
700 kbit	14	27	40	53	66	79	92	105	118	131	144	157
800 kbit	14	28	41	55	69	82	96	109	123	137	150	164
900 kbit	15	29	43	57	71	85	100	114	128	142	156	170
1000 kbit	15	30	44	59	74	88	103	118	132	147	162	176

Comment. At $\rho_C = 0.30$, the number of Fragments increases almost linearly with λ: at $I_G = I_N$ = 1,000 kbit, the value rises from 15 Fragments at λ = 1 to 176 Fragments at λ = 12. For a fixed λ, the increase with I_G is controlled and logarithmic, because I_G and I_N enter the formula through the $\log_2$ term. Lower values of ρ_C require stronger dispersion and therefore generate a larger number of Fragments.

Table — Composite Information Density $\rho_C = 0.25$

$I_G = I_N$	λ=1	λ=2	λ=3	λ=4	λ=5	λ=6	λ=7	λ=8	λ=9	λ=10	λ=11	λ=12
100 kbit	7	13	20	26	32	39	45	51	58	64	70	77
200 kbit	10	19	28	38	47	56	66	75	84	93	103	112
300 kbit	12	23	34	45	57	68	79	90	102	113	124	135
400 kbit	13	26	39	51	64	77	89	102	115	127	140	153
500 kbit	14	28	42	56	70	84	97	111	125	139	153	167
600 kbit	15	30	45	60	75	89	104	119	134	149	163	178
700 kbit	16	32	47	63	79	94	110	126	141	157	172	188
800 kbit	17	33	50	66	82	99	115	131	148	164	180	197
900 kbit	17	34	51	68	85	102	119	136	153	170	187	204
1000 kbit	18	36	53	71	88	106	123	141	159	176	194	211

Comment. At $\rho_C = 0.25$, the number of Fragments increases almost linearly with λ: at $I_G = I_N$ = 1,000 kbit, the value rises from 18 Fragments at λ = 1 to 211 Fragments at λ = 12. For a fixed λ, the increase with I_G is controlled and logarithmic, because I_G and I_N enter the formula through the $\log_2$ term. Lower values of ρ_C require stronger dispersion and therefore generate a larger number of Fragments.

Table — Composite Information Density $\rho_C = 0.20$

$I_G = I_N$	λ=1	λ=2	λ=3	λ=4	λ=5	λ=6	λ=7	λ=8	λ=9	λ=10	λ=11	λ=12
100 kbit	8	16	24	32	40	48	56	64	72	80	88	96
200 kbit	12	24	35	47	59	70	82	93	105	117	128	140
300 kbit	15	29	43	57	71	85	99	113	127	141	155	169
400 kbit	16	32	48	64	80	96	111	127	143	159	175	191
500 kbit	18	35	52	70	87	104	122	139	156	173	191	208
600 kbit	19	38	56	75	93	112	130	149	167	186	204	223
700 kbit	20	40	59	79	98	118	137	157	176	196	215	235
800 kbit	21	41	62	82	103	123	144	164	184	205	225	246
900 kbit	22	43	64	85	107	128	149	170	192	213	234	255
1000 kbit	22	44	66	88	110	132	154	176	198	220	242	264

Comment. At $\rho_C = 0.20$, the number of Fragments increases almost linearly with λ: at $I_G = I_N = 1{,}000$ kbit, the value rises from 22 Fragments at λ = 1 to 264 Fragments at λ = 12. For a fixed λ, the increase with I_G is controlled and logarithmic, because I_G and I_N enter the formula through the $\log_2$ term. Lower values of ρ_C require stronger dispersion and therefore generate a larger number of Fragments.

Table — Composite Information Density $\rho_C = 0.15$

$I_G = I_N$	λ=1	λ=2	λ=3	λ=4	λ=5	λ=6	λ=7	λ=8	λ=9	λ=10	λ=11	λ=12
100 kbit	11	22	32	43	53	64	74	85	96	106	117	127
200 kbit	16	31	47	62	78	93	109	124	140	155	171	186
300 kbit	19	38	57	75	94	113	132	150	169	188	206	225
400 kbit	22	43	64	85	106	127	148	170	191	212	233	254
500 kbit	24	47	70	93	116	139	162	185	208	231	254	277
600 kbit	25	50	75	99	124	149	173	198	223	247	272	297
700 kbit	27	53	79	105	131	157	183	209	235	261	287	313
800 kbit	28	55	82	109	137	164	191	218	246	273	300	327
900 kbit	29	57	85	114	142	170	199	227	255	284	312	340
1000 kbit	30	59	88	118	147	176	205	235	264	293	323	352

Comment. At $\rho_C = 0.15$, the number of Fragments increases almost linearly with λ: at $I_G = I_N = 1{,}000$ kbit, the value rises from 30 Fragments at λ = 1 to 352 Fragments at λ = 12. For a fixed λ, the increase with I_G is controlled and logarithmic, because I_G and I_N enter the formula through the $\log_2$ term. Lower values of ρ_C require stronger dispersion and therefore generate a larger number of Fragments.

Table — Composite Information Density $\rho_C = 0.10$

$I_G = I_N$	λ=1	λ=2	λ=3	λ=4	λ=5	λ=6	λ=7	λ=8	λ=9	λ=10	λ=11	λ=12
100 kbit	16	32	48	64	80	96	111	127	143	159	175	191
200 kbit	24	47	70	93	117	140	163	186	209	233	256	279
300 kbit	29	57	85	113	141	169	197	225	253	281	309	337
400 kbit	32	64	96	127	159	191	222	254	286	317	349	381
500 kbit	35	70	104	139	173	208	243	277	312	346	381	416
600 kbit	38	75	112	149	186	223	260	297	334	371	408	445
700 kbit	40	79	118	157	196	235	274	313	352	391	430	469
800 kbit	41	82	123	164	205	246	287	327	368	409	450	491
900 kbit	43	85	128	170	213	255	298	340	383	425	468	510
1000 kbit	44	88	132	176	220	264	308	352	396	440	484	528

Comment. At $\rho_C = 0.10$, the number of Fragments increases almost linearly with λ: at $I_G = I_N = 1{,}000$ kbit, the value rises from 44 Fragments at λ = 1 to 528 Fragments at λ = 12. For a fixed λ, the increase with I_G is controlled and logarithmic, because I_G and I_N enter the formula through the $\log_2$ term. Lower values of ρ_C require stronger dispersion and therefore generate a larger number of Fragments.

Cross-density comparison at $I_G = I_N = 1{,}000$ kbit

ρ_C	λ=1	λ=4	λ=8	λ=12
0.30	15	59	118	176
0.25	18	71	141	211
0.20	22	88	176	264
0.15	30	118	235	352
0.10	44	176	352	528

Overall interpretation. For the same information size, decreasing ρ_C increases the required fragmentation because the System is enforcing a lower admissible composite density. λ is the direct intensity parameter and scales the result almost linearly. I_G and I_N increase the fragment count more slowly, since they are inside the logarithmic term.

CNVS Theory – Appendices A and B

Mathematical Formulations and Complete Symbol Table

These appendices collect the principal formulas, definitions, constants, parameters, variables and indices used in the CNVS framework as developed in the current manuscript.

Appendix A – Complete Mathematical Formulations

A.1 Object, Encoding and Structural Representation

No.	Formula / Definition	Meaning
1	O_x	Object to be verified
2	$I(O_x)$	Complete informational structure associated with the object.
3	$\tilde{I}(O_x) = \text{Encode}(I(O_x))$	Encoded binary representation used by the System.
4	$E(t) = \text{Struct}(\tilde{I}(O_x), t)$	Structured element set generated from the encoded representation at time t
5	$I_x(t) \in \{0,1\}^*$	Finite binary representation admitted by the System.
6	$O_x \to I(O_x) \to \tilde{I}(O_x) \to E(t)$	Chain of generation of the verification instance.
7	$\text{Dec}(\tilde{I}_x(t)) = I_x(t)$	Decoded information for authorized agent
8	$\text{Dec}(\tilde{I}_x(t)) \neq I_x(t)$	Unauthorized decoding condition

A.2 Structural Universe and Structured Element

No.	Formula / Definition	Meaning
1	$\mathcal{S} \triangleq \Sigma_{\sigma \in \mathcal{F}} (\text{Id}_\sigma \times D_\sigma \times \text{At}_\sigma \times \text{Obs}_\sigma^{\perp} \times \mathcal{P}(P))$	Structural universe of admissible elements
2	$\mathfrak{S} \triangleq \{\langle E(t), R(t), C(t)\rangle \mid E \in \mathcal{P}_{\text{fin}}(\mathcal{S}), R \subseteq E \times E, C \subseteq \mathcal{C}\}$	space of possible states
3	$\text{Id}_\sigma \subseteq B^*$	Identifier domain
4	$D_\sigma \subseteq B^*$	Data domain
5	$\text{At}_\sigma \subseteq B^*$	Attribute domain
6	$\text{Obs}_\sigma \subseteq B^*$ $\text{Obs}_\sigma^{\perp} \triangleq \text{Obs}_\sigma \uplus \{\perp\}$	Observation domain
7	P = {Valid, Invalid, Transferable, Blocked, Verified, Suspended, Redeemable, Broken, Misinterpreted, Ambiguous, Accepted, Non-Final, Inconclusive, Doubtful, ...}	Set of logical properties
8	$s = \langle \sigma, (id, d, a, obs, p)\rangle \in$	Structured element
9	$s_i^{(n)} = \langle \sigma_i, (id_i, d_i, a_i, obs_i, p_i)\rangle$.	i_structured element
10	$P_i = \pi_2(V_{\text{Local}}(s_i))$	Properties generated by Local verification
11	$P_i = f(D_i, \text{At}_i, \text{Obs}_i)$	Alternative property expression
12	$P_i \subseteq \mathcal{P}(P)$	Property inclusion

A.3 State and Relations

No.	Formula / Definition	Meaning
1	S(t) := E(t), R(t), C	State of the System
2	$E(t) \subseteq \mathcal{S}$	Set of structured elements
3	$R(t) \subseteq E(t) \times E(t)$	Relations among structured elements
4	$C = \{c_1, c_2, ..., c_j\}$	Deterministic invariants
5	If $E(t) = \{s_1, s_2\}$, then $E(t) \times E(t) = \{(s_1, s_1), (s_1, s_2), (s_2, s_1), (s_2, s_2)\}$	Cartesian product example
6	$(s_i, s_j) \in R(t) \nRightarrow (s_j, s_i) \in R(t)$	Directional relation condition
7	(E(t), R(t)) defines a directed graph	Graph representation

A.4 Local Verification and Local Admissibility

No.	Formula / Definition	Meaning
1	$V_{Local} : \mathcal{S} \to \{0, 1\} \times \mathcal{P}(P)$	Local verification function
2	$V_{Local}(s_i) = (b_i, P_i)$	Local verification output
3	$b_i \in \{0, 1\}$	Boolean Local outcome
4	$Adm_L(s) = 1 \Leftrightarrow \pi_1(V_{Local}(s)) = 1$	Local admissibility
5	$Ter^+(t) \triangleq \{s \in Ter(t) \mid Adm_L(s)=1\}$	Locally admissible terminal fragment set.
6	$Conv_{At\_i}(D_i, Obs_i)$	Local convergence operator
7	$b_i = 1$ if $Conv_{At\_i}(D_i, Obs_i \mid At_I) = 1$; otherwise $b_i = 0$	Boolean convergence outcome
8	$\lvert D_i - Obs_i \rvert \le \varepsilon_{At\_i}$	Quantitative convergence condition
9	$b_i = Conv_{At\_i}(D_i, Obs_i) \wedge Form(Id_i, D_i, At_i, Obs_i)$	Formula Local boolean
10	$D_i = Obs_i$, or $Eq_{At\_i}(D_i, Obs_i) =1$	Qualitative convergence condition
11	$Status_L(s) = \perp$	indefinite and suspended local state
12	$s \in Ter(t) \not\Rightarrow Adm_L(s)=1$	Terminal membership does not imply local admissibility.
13	$obs = \perp \Rightarrow Status_L(s) = \perp$	Undefined Observational State: In the absence of external input, the local verification enters a suspended logical state ($\perp$), blocking the transition to admissibility without constituting an active falsification (0)

A.5 Global Verification and Global Validity

No.	Formula / Definition	Meaning
1	$V_{Global}(S(t))$	Global verification function
2	$Valid(S(t)) \Leftrightarrow V_{Global}(S(t)) = 1$	Global validity
3	$V_{Global}(S(t)) \neq \wedge_{s \in Ter(t)} \pi_1(V_{Local}(s))$	Global validity is not reducible to local aggregation.
4	$V_{Global}(S) = f(E,R,C)$	Global verification as structural function
5	$V_{Global}(S(t)) = 1 \Leftrightarrow (\forall s \in Ter(t), \pi_1(V_{Local}(s)) = 1) \wedge Cons_R(R(t),E(t)) \wedge Inv_C(S(t))$	Global validity requires terminal local admissibility, relational consistency, and invariant satisfaction.
6	$V_{Global} : StateSpace \to \{1, 0\}$	V_Global codomain
7	$\exists S(t) \in \mathfrak{S}$ such that $(\forall s \in Ter(t), \pi_1(V_{Local}(s)) = 1) \wedge V_{Global}(S(t)) = 0$	Locally valid but Globally invalid possibility
8	$Inv_C(S(t)) = 1 \Leftrightarrow \forall c_i \in C, c_i(E(t), R(t))=1$	condition of satisfaction of the invariants for the acceptance of the Global State
9	$Cons_R(R(t), E(t)) = 1 \Leftrightarrow \forall s, s' \in E(t), ((s, s') \in R(t) \Leftrightarrow s' \in \mathfrak{D}(s))$	partial global state relationship condition

A.6 Decomposition, Type Preservation and Recursion

No.	Formula / Definition	Meaning
1	$\mathfrak{D} : \mathcal{S} \to \mathcal{P}_{fin}(\mathcal{S})$	Recursive decomposition operator.
2	$\mathfrak{D}(s_i) = \{s_{i1}^{(1)}, ..., s_{ik\_i}^{(1)}\}$	Finite decomposition of an element
3	$\lvert\mathfrak{D}(s)\rvert = 0 \vee \lvert\mathfrak{D}(s)\rvert \ge 2$	Finite branching condition
4	$Ter(t) \triangleq \{s \in E(t) \mid \varnothing \quad\} (s)=$	Atomic element
5	$\forall s \in \quad, \forall f_i \in \mathfrak{D}(s), f_i \in \quad \wedge type(f_i) \in \Omega(type(s))$	Type preservation
6	$type : \mathcal{S} \to \mathcal{T}$	Type function
7	$s' \preceq_{\mathfrak{D}} s \Leftrightarrow s' \in \mathfrak{D}(s)$	Descendant relation induced by

		decomposition.
8	$E^{(0)}(t) = E_N(t)$	Nodal level
9	$E^{(n+1)}(t) = \cup_{s \in E^{(n)}(t)} \mathfrak{D}(s)$	Recursive decomposition levels
10	$0 \leq n \leq \text{Depth}(t)$	Operational depth condition
11	For every finite n, $E^{(n)}(t)$ is defined in the ideal model	Ideal finite recursion
12	$\text{par} : E^{(n+1)}(t) \to E^{(n)}(t)$	Parent map
13	$\text{par}(s') = s \Leftrightarrow s' \in \mathfrak{D}(s)$	Parent relation
13	$N_i \in E^{(0)}(t)$, with $N_i := s_i^{(0)}$	Node definition
14	$\mathcal{T}(t) = E^{(n)}(t)$, where n is the terminal decomposition level	Terminal fragment set

A.7 Conservation of Information and Non-Uniformity

No.	Formula / Definition	Meaning
1	$I : \mathcal{S} \to \{0,1\}^*$	Semantic information extraction function
2	$\text{Rec} : \mathcal{P}_{\text{fin}}(\mathcal{S}) \to \mathcal{S}$	Reconstruction operator
3	$\mathtt{I}(\text{Rec}(\mathfrak{D}(s))) = \mathtt{I}(s)$	Canonical structural information is conserved under reconstruction.
4	$\forall s \in \mathcal{S}$ such that $\mathfrak{D}(s) \neq \emptyset$, $I_{\text{payload}}(s) = \Sigma_{f\_i \in \mathfrak{D}(s)} I_{\text{payload}}(f_i)$ $\exists f_a, f_b \in \mathfrak{D}(s)$ such that $I_{\text{payload}}(f_a) \neq I_{\text{payload}}(f_b)$	Non-uniform fragment size condition
5	$\sum_{s' \in \mathfrak{D}(s)} \lvert I(s') \rvert \geq \lvert I(s) \rvert$	Payload plus structural overhead inequality.
6	$\lvert \mathtt{I}(s) \rvert = I_{\text{payload}}(s) + \mathtt{I}_{\text{metadata}}(s)$	Total structural information decomposes into semantic payload and metadata.
7	$\Sigma_{j=1}^{F\_x(t)} I_{F,j}(t) = I_h^{(n)}(t)$	Fragmental conservation under non-uniform distribution
8	$\exists f_a, f_b \in \mathfrak{D}(s)$: $I_{\text{payload}}(f_a) \neq I_{\text{payload}}(f_b)$	Non-uniformity condition

A.8 State Transition Law

No.	Formula / Definition	Meaning
1	$S(t) \to S(t+1)$	State transition
2	$S(t+1) = T(S(t), U(t))$ $T : S \times U \to 2^S$	Transition function
3	$S(t+1) = S'$ if $V_{\text{Global}}(S') = 1$ $S(t+1) = S(t)$ if $V_{\text{Global}}(S') = 0$	Verified transition set
4	$V_{\text{Global}}(S(t+1)) = 1$	Global transition condition
5	$S' \subseteq T(S(t), U(t))$, with $V_{\text{Global}}(S') = 1$	Transition subset condition

A.9 Knowledge and Information Levels

No.	Formula / Definition	Meaning
1	$K_x(t) \preceq I_x(t)$	Knowledge as accessible information
2	$\mathtt{I}_G(t) \cong (E(t), R(t), C)$	Global information
3	$\mathtt{I}_G(t) = I(S(t))$	Information of state
4	$I_G(t) = I(E(t))$	Semantic Global Information
5	$K_G(t) \subsetneq I_G(t)$	Semantic Global knowledge
6	$I_N(t) = I(N_i(t))$	Nodal information
7	$K_N(t) \preceq I_N(t)$	Nodal knowledge
8	$I_h^{(n)}(t) = I(s_{ij}^{(n)}(t))$	Level information
9	$K_h^{(n)}(t) \preceq I_h^{(n)}(t)$	Level knowledge
10	$I_F(t) = I(s_{ij}^{(n)}(t))$	Fragment information
11	$K_F = K_{\text{Local}}$	Fragment/Local knowledge equivalence
12	$K_F(t) \precneqq I_F(t)$	Fragment knowledge
13	$I_F(t) \subsetneq I_h^{(n)}(t) \subsetneq I_N(t) \subsetneq I_G(t)$	Information hierarchy
14	$X_j(t) \in \mathcal{T}(t)$	Verifier fragment assignment
15	$\mathcal{I} : \mathcal{T}(t) \to \{0,1\}^*$	Information return function

16	$\mathcal{K} : \mathcal{V} \times \mathcal{S} \to \mathbb{N}$ $\mathcal{K}(v, f)$ = verifier-accessible knowledge for verifier v assigned to terminal fragment f	Verifier knowledge
17	$K_{VF}(v_j, t) \lneqq K_F(t) \lneqq I_F(t)$	Limited verifier knowledge
18	$K_{VF}(t) \lneqq I_N(t)$	Verifier lower than nodal information
19	$K_{VF}(t) \neq I_N(t)$	Verifier not equal to nodal information
20	$K_{VF}(v_j, t) \lneqq K_F(t) \subsetneq K_h^{(n)}(t) \subsetneq K_N(t) \subsetneq K_G(t)$	Knowledge hierarchy
21	$\text{Assign}(f_{ij})^{(d)} \to v_j$	Assignment function
22	$\tilde{I}_F(t)=\text{Enc}(I_F(t))$	Encoded fragment information
23	$K_{VF}(t) \lneqq \tilde{I}\_F(t)$	Verifier knowledge of encoded fragment
24	$I_{min}(a) \leq (v, f) < I_{total}(f) \ll I_{total}(s) \ll I_{Global}(S(t)$	Global relations

A.10 Information Density

No.	Formula / Definition	Meaning
1	$\rho_N(t) = \lvert I_F(t)\rvert / \lvert I_N(t)\rvert$	Local/nodal information density
2	$\rho_G(t) = \lvert I_F(t)\rvert / \lvert I_G(t)\rvert$	Global information density
3	$\rho_{GN}(t) = \lvert I_N(t)\rvert / \lvert I_G(t)\rvert$	Relative Nodal Density
4	$\rho_C(t) = (\lvert I_F(t)\rvert / \lvert I_N(t)\rvert)^{\alpha} \cdot (\lvert I_F(t)\rvert / \lvert I_G(t)\rvert)^{\beta-\rho}$	Composite information density
5	$\alpha, \beta_\rho \in [0,1]$ and $\alpha+\beta_\rho = 1$	Weight constraints
6	$\rho > \varepsilon \Rightarrow$ Reject(S)	Density rejection rule
7	$\rho_N(t) \leq \varepsilon_N$ and $\rho_G(t) \leq \varepsilon_G$	Local/Global density thresholds
8	$\rho_C(N_i, t) \leq \varepsilon_C$	Composite density threshold
9	$\rho_C(t)>\varepsilon_C \Rightarrow \text{Depth}(N_i, t+1)>\text{Depth}(N_i, t)$	Density-triggered decomposition
10	$\text{Depth}(N_i, t+1)>\text{Depth}(N_i, t) \Rightarrow \rho_C(t+1)<\rho_C(t)$	Depth reduces composite density
11	$\text{Depth}(N_i, t)\uparrow \Rightarrow \rho_C(t)\downarrow$	Depth relation

A.11 Adaptive Decomposition and Fragment Count

No.	Formula / Definition	Meaning
1	$(I_N(t_2)>I_N(t_1)) \vee (I_G(t_2)>I_G(t_1)) \Rightarrow \text{Depth}(N_i, t_2)>\text{Depth}(N_i, t_1)$	Depth growth condition
2	$\text{Fragments}(N_i, t) = \gamma(I_N(t), \rho_C(t))$	Fragment count per node
3	$\text{Fragments}(N_i, t) = \lvert E_i^{(\text{Depth}(N\_i,t))}(t)\rvert$	Terminal Fragments per node
4	$\text{Fragments}(t) = \Sigma_i \text{Fragments}(N_i,t)$	Total terminal Fragments
5	$\lvert\mathcal{T}(t)\rvert=\text{Fragments}(t)$	Terminal set cardinality
6	$\mathcal{T}(t)=\bigcup_i E_i^{(\text{Depth}(N\_i,t))}(t)$	Terminal set
7	$\text{Depth}(N_i,t) = f(\rho_{C,i}(t)) = f(I_{N\_i}(t), I_G(t))$	Depth as density function
8	$\lvert V_{ext}^{\text{required}}(t)\rvert \geq \lvert\text{Fragments}(t)\rvert$	External verifier requirement

A.12 Information Restriction Principle and Fragmentation Function

No.	Formula / Definition	Meaning
1	$1/(\rho_C(t) \cdot N(t)^{\beta\_N})$	Control reciprocal expression
2	$\text{Log}_2(1 + I_G(t))$	Global information growth term
3	$F_x(t)=\lambda\lceil 1/(\rho_C \cdot N(t)^{\beta\_N}) \cdot \log_2(1+(I_G(t) +I_N(t))/I_0)\rceil$	Fragmentation function
4	$F_x(t)\geq 1$, $F_x(t) \in \mathbb{N}$	Fragment count domain
5	$\lambda\uparrow \Rightarrow F_x\uparrow$	Lambda monotonicity
6	$I_G\uparrow, I_N\uparrow \Rightarrow F_x\uparrow$	Information monotonicity
7	$I_0\downarrow \Rightarrow F_x\uparrow$	Reference unit monotonicity
8	$N(t)\uparrow \Rightarrow F_x\downarrow$	Node count monotonicity
9	$F_x \uparrow \Rightarrow K_{VF}(t)/I_G(t) \downarrow$	Knowledge decreases with fragmentation
10	$K_{VF}(t) \not\Rightarrow I_N(t)$	Restriction against node reconstruction
11	$K_{VF}(t) \not\Rightarrow I_G(t)$	Restriction against Global reconstruction

12	$I_F(t)=I_h^{(n)}(t)/F_x(t)$	Average fragment information
13	$(\rho_C, \lambda, I_0, N(t)) \Rightarrow F_x(t) \Rightarrow I_F(t) \Rightarrow K_{VF}(t)$	Restriction chain
14	$\lim_{m\to\infty} \sup_{v\in\mathcal{V},\, s\notin Ter(t),\, f\in\mathfrak{D}(s)} \mathcal{K}(v,f)/I_{total}(s) = 0$	Verifier knowledge becomes negligible relative to global information.
15	$\lim_{F\_x(t)\to\infty} \rho_N(t)=0$	Nodal density limit
16	$\omega(D_i) = I(G;F_i)/H(G)$	Normalized Informational Utility of a Fragment
17	$K_{VF}(t+1) < K_{VF}(t)$	Knowledge reduction across time
18	$\tau \in (0, 1)$	Minimum normalized amount of useful information required for unauthorized reconstruction.
19	$q\bar{\omega} < \tau \Longrightarrow \lim_{n\to\infty} P[\Sigma^n_{i=1} Y_i\omega(D_i) \geq \tau \cdot n]= 0$	Weighted Emergent Security Theorem
20	$_t : Ter(t) \times \Xi \to$	stochastic operator for injective assignment of Ter(t) to the Verifier

A.13 Entropy and Mutual Information

No.	Formula / Definition	Meaning
1	$H(I_N \mid K_{VF}^{(n)}(j, t)) \geq H(I_N) - \varepsilon_n$	Residual uncertainty about nodal information remains asymptotically close to full uncertainty.
2	$I(K_{VF};I_N) \approx 0$	Compact intuitive notation for mutual information minimization.
3	$I(I_N ; K_{VF}^{(n)}(j, t)) \leq \varepsilon_n$, with $\varepsilon_n \to 0$ as $n \to \infty$	Verifier-accessible knowledge reveals asymptotically negligible information about nodal information.
4	$p_{perm} = 1/ n!$	Random reconstruction probability
5	$\lvert K_{VF}^{(n)}(t)\rvert / \lvert I_G(t)\rvert \to 0$	Verifier-accessible knowledge becomes asymptotically negligible relative to Global Information.
6	$\varepsilon_n \to 0$ as $n \to \infty$	Vanishing leakage/error term under increasing fragmentation.

A.14 Emergent Security Scaling and Adversarial Knowledge Bounds

No.	Formula / Definition	Meaning
1	$I_F = \Phi(q_1, q_2,\ldots, q_m)$	Defines the functional reconstruction of global information I_F based on the intercepted fragments
2	$\tau = f\tau(H, R, C)$	Establishes the system's critical threshold τ as a function of entropy, relations, and collusion model
3	$q\bar{\omega_k} < \tau \Rightarrow \lim_{k\to\infty} P[\Sigma_{i=1}^{k} Y_i\omega(D_i) \geq \tau k]=0$	Asymptotic security bound: successful attack probability converges to zero as the number of terminal fragments k tends to infinity, provided intercepted information remains below τ
4	$k_\tau = \lceil \tau k \rceil$ = Critical Threshold value	Defines the discrete critical threshold τ, representing the minimum number of fragments required for system compromise
5	$P_{coll}(n) = \Sigma^n_{k=\lceil \tau \cdot n \rceil} (n ; k)\ q^k (1 – q)^{n-k}$	Expresses the cumulative binomial probability $P_{coll}(n)$ of independently compromising at least k_τ fragments
6	$q < \tau \Longrightarrow \lim_{n\to\infty} P_{coll}(n) = 0$	Asymptotic decay of binomial collusion:

		global compromise probability collapses to zero if $q < \tau$
7	$D(\tau\|q)=\tau \log (\tau/ q) + (1 - \tau) \log ((1 - \tau)/(1 - q))$	Defines the Kullback-Leibler divergence $D(\tau\|q)$ between the security threshold τ and the compromise probability q
8	$P(X/ n \geq \tau) \leq e^{-nD(\tau\|q)}$	Chernoff/KL bound
9	$P_{coll}(n) \leq e^{-nD(\tau\|q)}$	Chernoff/KL upper bound for the unweighted independent Bernoulli compromise model.
10	$\omega(D_i) = I(G; F_i)$	Quantifies a fragment's informational weight $\omega(D_i)$ as the mutual information between the global invariant G and the fragment F_i
11	$\omega(D_i) = H(G) - H(G\|F_i)$	Reformulates fragment weight via Shannon entropy, measuring the uncertainty reduction regarding the global state
12	$\omega(D_i) = I(G; F_i) / H(G)$	Defines the normalized informational weight of a fragment on a [0, 1] interval
13	$\bar{\omega} = 1/ n \ \Sigma^n_{i=1} \ \omega(D_i)$	Calculates the average normalized informational weight $\bar{\omega}$ across all n fragments
14	$E [\Sigma^n_{i=1} \ Y_i\omega (D_i)] = q \Sigma^n_{i=1} \omega (D_i) = nq\bar{\omega},$	Expected value of the adversary's cumulative information under an independent compromise probability q
15	$p_{comp}=q + (1-q) p_{inf}$	Defines the generalized compromise probability p_comp combining independent breach q and inference probability p_{inf}
16	$P(C_{comp\_i} \mid C_{comp\_1}, \ldots, C_{comp\_\{i-1\}}) \leq p_{comp}.$	Uniform upper bound on the conditional probability of compromising the next critical fragment after any sequence of previous compromises.
17	$P(Rec^*) \leq p_{comp}^m$	Upper bound for total reconstruction probability P(Rec^*) under dependent collusion requiring m fragments
18	$P(Rec^*) \leq \prod^m_{i=1} p_{comp} = p_{comp}^m$	Derives the dependent reconstruction bound by applying the chain rule to conditional probabilities
19	$m_{min} = \lceil \ln(\eta) / \ln(p_{comp}) \rceil$ $0<p_{comp}<1$ e $\eta \in (0,1)$	Architectural feasibility constraint m_{min} required to maintain reconstruction probability below a target threshold η
20	$P(Rec^*) \leq q^m$	Ideal non-inferable critical-fragment case, obtained when $p_{inf} = 0$ and therefore $p_{comp} = q$.
21	$K_{adv} = \Sigma_{i \in A} \ \omega(D_i)$	Quantifies total adversarial knowledge K_{adv} as the sum of intercepted fragments' informational weights
22	$p_{inf} = \nu(K_{adv})$	Defines inference probability p_inf as a function ν of accumulated adversarial knowledge K_{adv}
23	$p_{comp} = q + (1-q) \nu(K_{adv})$	Total compromise probability p_{comp} explicitly integrating the knowledge-dependent inference function
24	$P(Rec^*) \leq [q + (1-q) \nu(K_{adv})]^m$	Generalized Emergent Security Bound: maximum reconstruction probability when attackers dynamically infer missing data
25	$P(X \geq \tau k) \leq \exp(-2(\tau k - \mu)^2 / \Sigma_{i=1}^k \omega(D_i)^2)$	Hoeffding-type concentration bound for independent bounded weighted fragment contributions.
26	$H(X_S \mid M_S) \geq H(X_S) - \gamma_{top}$	Bounded Topological Leakage: The quantitative limit γ_{top} of semantic information (X_S) that structural topology or metadata (M_S) can inadvertently expose to the adversary.

27	$H_\infty(d_{miss} \mid View_{adv}) \geq h_{min}$	Min-Entropy Residual: The strictly positive entropy margin that the missing critical fragment (d_{miss}) must maintain with respect to the opponent's total information view ($View_{adv}$).
28	$p_{inf} \leq 2^{-h\_min}$	Min-Entropy Inference Bound: The absolute upper bound on the probability of deterministic inference by the adversary, derived directly from the entropy margin.
29	$p_{inf}(t) \triangleq 1 - \mathbb{E}[H(S(t) \mid W(t))] / H(S(t_0))$.	Entropy-Driven Dynamic Inference: A proxy model of inference probability based on the dynamic erosion of global entropy over time due to compromised nodes W(t).

A.15 Axioms

No.	Formula / Definition	Meaning
1	$s \in E_{cand}(t) \not\Rightarrow Adm_L(s)=1$	Axiom I: raw structural membership is verification-independent.
2	$Ter^+(t) \triangleq \{s \in Ter(t) \mid Adm_L(s) = 1\}$	Locally admissible terminal subset.
3	$S \in \mathfrak{S}^+ \Leftrightarrow V_{Global}(S) = 1$	Accepted state validity depends on Global Verification.
4	$\mathfrak{D}(s)=\emptyset \vee \lvert\mathfrak{D}(s)\rvert \geq 2$	Axiom II: finite recursive decomposition
5	$\exists f_a, f_b \in \mathfrak{D}(s): I_{payload}(f_a) \neq I_{payload}(f_b)$	Axiom II: non-uniform fragmentation
6	$\mathcal{A}_t : Ter(t) \times \Xi \to \mathcal{V}$ $f_i \neq f_j \Rightarrow \mathcal{A}_t(f_i,\xi) \neq \mathcal{A}_t(f_j, \xi)$	Axiom II: randomized injective assignment.
7	$Rec(\mathfrak{D}(s)) = s$	Axiom II: reconstructibility up to canonical encoding.
8	$V_{Global}(S(t)) \neq LocalOK(Ter(t))$	Axiom III: Global validity is not reducible to Local validity
9	$V_{Global}(S(t))=TRUE \Leftrightarrow LocalOK(Ter(t)) \wedge Cons_R(R(t),E(t)) \wedge Inv_C(S(t))$	Axiom III: Global validity requires local admissibility, relational consistency, and invariant satisfaction
10	$\exists S(t)$ such that $LocalOK(Ter(t)) = 1 \wedge V_{Global}(S(t)) = 0$	Axiom III witness: locally adm terminal checks may still fail Global Verification

Appendix B – Complete Symbol Table

B.1 Objects, Information and Encoding

Symbol	Type / Domain	Definition	Meaning
O_x	Object / reality	Object to be verified	Empirical or abstract entity whose existence is to be attested.
x	Index / domain label	Identifies information domain	Used for Global, nodal, fragmental, Local or object-related domains.
$I(O_x)$	Function	$I(O_x)$	Complete informational structure associated with object O_x.
I_x	Binary information	$I_x = \text{Encode}(O_x)$	Finite encoded representation of O_x.
$I_x(t)$	Time-dependent information	$I_x(t) \in (\{0,1\}^*)^z$	Existing information at time t for domain x.
$\tilde{I}(O_x)$	Encoded information	$\tilde{I}(O_x)=\text{Encode}(I(O_x))$	Encoded representation of the informational structure.
Enc	Function	$\text{Enc}:\{0,1\}^* \to \{0,1\}^*$	Encoding/encryption function.
Dec	Function	$\text{Dec}:\{0,1\}^* \to \{0,1\}^*$	Decoding/decryption function.
Encode	Function	$\text{Encode}: I(O_x) \to \tilde{I}(O_x)$	Maps the object information structure into an encoded representation.
Struct	Function	$\text{Struct}(\tilde{I}(O_x), t)=E(t)$	Generates structured elements from the encoded representation at time t.
{0, 1}*	Set	All finite binary strings	Universal representation domain for information in the model.
z	Natural number	$z \in \mathbb{N}$	Exponent/length-domain marker for binary information representation.

B.2 Structural Elements and Components

Symbol	Type / Domain	Definition	Meaning
$\mathcal{S}$	Set	$\mathcal{S} \triangleq \Sigma_{\sigma \in \mathcal{T}}(\text{Id}_\sigma \times D_\sigma \times \text{At}_\sigma \times \text{Obs}_\sigma^{\perp} \times \mathcal{P}(P))$	Structural universe of admissible elements.
s_i	Structured element	$s_i = (id_i, D_i, \text{At}_i, \text{Obs}_i, P_i)$	Single structured element of the System.
$s_i(t)$	Time-dependent element	s_i at time t	Structured element considered at time t.
id_i	Identifier	$id_i \in \text{Id}$	Unique identifier of element or fragment.
Id	Set	$\text{Id} \subseteq \{0,1\}^*$	Domain of admissible identifiers.
D_i	Data component	$D_i \in D$	User-declared or internally represented data value.
D	Set	$D \subseteq \{0,1\}^*$	Domain of admissible data values.
At_i	Attribute component	$\text{At}_i \in A_t$	Observable attribute/class to be verified.
At	Set	$\text{At} \subseteq \{0,1\}^*$	Domain of admissible attributes.
a_{ij}	Attribute item	$a_{ij} \in \text{At}_i$	Single attribute in the attribute set.
Obs_i	Validation/evidence component	$obs_i \in \text{Obs}_\sigma^{\perp}$	External verifier statement, measurement or attestation.
Obs	Set	$\text{Obs} \subseteq \{0,1\}^*$	Domain of admissible validations.
P_i	Property set	$P_i \in \mathcal{P}$ $P_i = \pi_2(V_{\text{Local}}(s_i))$	Logical properties generated or used by verification.

P	Set	is the finite set of admissible logical properties	Universe of admitted logical properties.
$\mathcal{P}(P)$	Power set	All subsets of P	Power set of P, i.e. the set of all finite subsets of P.
Valid	Property	Valid $\in \mathcal{P}$	Logical condition indicating admissibility/validity.
Invalid	Property	Invalid $\in \mathcal{P}$	Logical condition indicating failed validity.
Blocked	Property	Blocked $\in \mathcal{P}$	Logical condition blocking operations.
Transferable	Property	Transferable $\in \mathcal{P}$	Logical condition allowing transfer.
Suspended	Property	Suspended $\in \mathcal{P}$	Non-final pending condition.
Doubtful	Property	Doubtful $\in \mathcal{P}$	Uncertain or questionable condition.
$\perp$	Logic state	Bottom/ suspended	$\perp$ denotes absence of external observation. It is a structural placeholder in $Obs_\sigma^\perp$ and is neither successful nor failed convergence.

B.3 State, Relations and Invariants

Symbol	Type / Domain	Definition	Meaning
S(t)	State	S(t) =E(t), R(t), C	State of the CNVS at time t.
E(t)	Set	$E(t) \subseteq \mathcal{S}$	Set of structured elements admitted at time t.
R(t)	Relation	$R(t) \subseteq E(t) \times E(t)$	Directional relations among structured elements.
C	Set	$C=\{c_1,...,c_j\}$	Set of deterministic invariants.
c_j	Invariant	$c_j \in C$	Single deterministic System constraint.
j	Natural number	$J \in \mathbb{N}$	Number of invariants.
Consistent	Predicate	Consistent(R(t))	Checks structural/logical consistency of relations.
StateSpace	Set	Domain of V_{Global}	Set of all possible states.
t	Time index	t ∈ Time	Time parameter.
t_1, t_2	Time indices	$t_1 < t_2$	Two ordered time instants.
TRUE, FALSE	Boolean values	{TRUE, FALSE}	Logical outputs of verification predicates.
0, 1	Boolean values	{0, 1}	Binary form of verification outcome.

B.4 Verification, Admissibility and Convergence

Symbol	Type / Domain	Definition	Meaning
V_{Local}	Function	$V_{Local} : \mathcal{S} \rightarrow \{0, 1\} \times \mathcal{P}(P)$	Local verification function.
V_{Global}	Function	V_{Global}: StateSpace $\rightarrow \{0, 1\}$	Global verification function.
b_i	Boolean	$b_i \in \{0, 1\}$	Local Boolean verification output.
π_1 π_2	Projection	$\pi_1(b_i, P_i) = b_i$ $\pi_2(b_i, P_i) = P_i$	Extracts Boolean component from Local verification output.
Adm_L	Predicate	$Adm_L(s_i, t) \Leftrightarrow \pi_1(V_{Local}(s_i(t))) = 1$	Local admissibility predicate.
Conv (type σ)	Function / predicate	$Conv_\sigma : At_\sigma \times D_\sigma \times Obs_\sigma^\perp \rightarrow \{0, 1\}$ if and only if: obs ∈ Obs_σ	Attribute-conditioned convergence between declared and verified data.
Eq_{At_i}	Predicate	$Eq_{At\_i}(D_i, Obs_i)$	Attribute-specific equivalence predicate for qualitative data.
ε_{At_i}	Tolerance	$\varepsilon_{At\_i} \geq 0$	Attribute-specific admissible tolerance.
valid	Logical condition	valid $\in \mathbb{P}$ or valid(S)	Validity status.
Reject	Operation	Reject(S)	State rejection operation when

			constraints fail.
Formal function (form)	Function	$Form_\sigma : Id_\sigma \times D_\sigma \times At_\sigma \times Obs_\sigma^\perp \to \{0, 1\}$	Formal admissibility function

B.5 Decomposition, Reconstruction and Type

Symbol	Type / Domain	Definition	Meaning
$\mathfrak{D}(s)$	Function	$\mathfrak{D}(s): \mathcal{S} \to \mathcal{P}_{fin}(\mathcal{S})$	Time-indexed decomposition function.
$\mathfrak{D}$	Function	$\mathfrak{D}(s)$	General decomposition function applied to state.
$D^{(n)}$	Function	$\mathfrak{D}^{(n)}(S(t))$	n-level decomposition of state.
$\mathcal{P}_{fin}(\mathcal{S})$	Set	Finite subsets of $\mathcal{S}$	Codomain of decomposition.
s'	Fragment / descendant	$s' \in \mathfrak{D}(s)$	Generic descendant fragment.
$s_{ij}^{(n)}$	Fragment	j-th fragment of element i at level n	Structured descendant at decomposition depth n.
i	Index	$i \in \mathbb{N}$	Element or node index.
j	Index	$j \in \mathbb{N}$	Fragment/verifier/Local component index.
k_i	Natural number	$1 \leq k_i < \infty$	Number of Fragments generated by element i.
n	Natural number	$n \in \mathbb{N}$	Decomposition level/depth index.
d	Natural number	$d \in \mathbb{N}$	Depth index used for terminal Fragments.
l	Index	$l \in \mathbb{N}$	Additional fragment/node index.
type	Function	$type : \boldsymbol{\mathcal{S}} \to \boldsymbol{\mathcal{T}}$	Maps element to structural type.
$\boldsymbol{\mathcal{T}}$	Type set $\mathcal{S} \to \boldsymbol{\mathcal{T}}$	Codomain of type	set of admissible structural types.
$\preceq_{\mathfrak{D}}$	Relation	$s' \preceq_{\mathfrak{D}(s)}$	Descendant relation under decomposition.
$\Omega(\sigma)$	Type semantic preservation	$\Omega(\sigma) \subseteq \boldsymbol{\mathcal{T}}$ $type(f_i) \in \Omega(type(s))$	the type of the children is compatible with the parent type, it is not necessarily identical
par_t	Function	$par: E^{(n+1)}(t) \to E^{(n)}(t)$	Parent map.
Rec_t	Function	$Rec: \mathcal{P}_{fin}(\mathcal{S}) \to \mathcal{S}$	Reconstruction operator.
I	Function	$I: \mathcal{S} \to \{0, 1\}^*$	Information extraction/representation function.
I			semantic informational content
$\mathbb{I}$			structured informational state including relations and constraints
$I_{payload}$	Information component	Payload information	Original informational content.
$\mathbb{I}_{metadata}$	Information component	Metadata/overhead information	IDs, keys, signatures, markers or reconstruction overhead.

B.6 Levels, Nodes and Fragments

Symbol	Type / Domain	Definition	Meaning
$E^{(0)}(t)$	Set	Nodal level	Level-zero elements/nodes.
$E^{(n)}(t)$	Set	Elements at decomposition level n	Set of Fragments at depth n.
$E_N(t)$	Set	Nodal elements	Alternative notation for nodal level.
N_i	Node	$N_i := s_i^{(0)}$	i-th nodal structured element.
Depth(t)	Natural number	Maximum operational decomposition depth at time t	Global operational depth.
Depth(N_i, t)	Natural number	Depth of node N_i at time t	Node-specific decomposition depth.
$\mathcal{T}(t)$	Set	Terminal fragment set	Set of terminal Fragments at time t.
Fragments(N_i,t)	Natural number	$\lvert E_i^{(Depth(N\_i,t))}(t)\rvert$	Number of terminal Fragments

			in node N_i.
Fragments(t)	Natural number	Σ_i Fragments(N_i, t)	Total terminal Fragments in the System.
$F_x(t)$	Natural number	$F_x(t)$ = Fragments required at time t	Number of terminal Fragments required by fragmentation rule.
γ	Function	$\gamma(I_N(t),\rho_C(t))$	Monotone fragment-count function.
Sub	Function	Sub($s_i^{(n)}$)	Set of subFragments of an element.
Γ	Function	$\Gamma = m_{max}/ m_{min}$	Granularity Global Information'

B.7 Knowledge and Information Levels

Symbol	Type / Domain	Definition	Meaning
$K_x(t)$	Knowledge	$K_x(t) \preceq I_x(t)$	Semantic Accessible/known information in domain x.
$\mathtt{I}_G(t)$	Information	$\mathtt{I}_G(t) \cong (E(t), R(t), C)$	Global information of the System.
$K_G(t)$	Knowledge	$K_G(t) \preceq I_G(t)$	Semantic Global knowledge.
$I_N(t)$	Information	$I_N(t) = I(N_i(t))$	Semantic Nodal information.
$K_N(t)$	Knowledge	$K_N(t) \preceq I_N(t)$	Semantic Nodal knowledge.
$I_h^{(n)}(t)$	Information	Information of all Fragments at level n	Level information.
$K_h^{(n)}(t)$	Knowledge	$K_h^{(n)}(t) \preceq I_h^{(n)}(t)$	Semantic Level knowledge.
$I_F(t)$	Information	$I_F(t) = I(s_{ij}^{(n)}(t))$	Semantic Fragmental information.
$K_F(t)$	Knowledge	$K_F(t) \preceq I_F(t)$	Semantic Fragmental knowledge.
K_{Local}	Knowledge	$K_{Local} = K_F$	Local knowledge.
$\mathcal{K}(v,f)$	Knowledge	Verifier fragment knowledge	Knowledge available to external verifier.
$K_{IG}(t)$	Knowledge	$K_{IG}(t) \preceq \mathtt{I}_G(t)$	Global knowledge.
$\mathtt{I}_N(t)$	Information	$\mathtt{I}_{IN}(t) = \mathtt{I}(N_i(t))$	Nodal information.
$\mathtt{I}_F(t)$	Information	$\mathtt{I}_F(t) \cong (s_{ij}^{(n)}(t), R_{sij}, C_{sij}, V_{Local})$	Structured fragmental informational representation.
$K_h^{(n)}(t)$	Knowledge	$K_h^{(n)}(t) \preceq \mathtt{I}_h^{(n)}(t)$	Level knowledge.
$K_{VF_j}(t)$	Knowledge	Knowledge of verifier v_j	Verifier-specific fragment knowledge.
v_j	Agent	External verifier j	External Verification Agent.
K_{adv}	Adversarial knowledge	$K_{adv} = \Sigma_{i\in A}\, \omega(D_i)$	Total normalized informational weight controlled by the adversarial coalition.
$X_j(t)$	Random variable	$X_j(t) \in \mathcal{T}(t)$	Terminal fragment assigned to verifier v_j.
$\mathcal{J}$	Function	$\mathcal{J}:\mathcal{T}(t) \to \{0,1\}^*$	Information-content return function.
$\overline{\mathcal{J}}$	Function	Encoded information-content return	Returns encoded information associated with a fragment.
$\preceq$	Relation	$K \preceq I$	Accessible knowledge is bounded by information.
$\lneq$	Strict relation	$K \lneq I$	Strictly incomplete knowledge relation.
$\subsetneq$	Strict subset	$A \subsetneq B$	A is a proper subset of B.
$\not\Rightarrow$	Non-implication	$A \not\Rightarrow B$	A does not imply/reconstruct B.

B.8 Density, Thresholds and Protocol Parameters

Symbol	Type / Domain	Definition	Meaning
$\rho_N(t)$	Density	$\rho_N(t)=\lvert I_F(t)\rvert/\lvert I_N(t)\rvert$	Local/nodal information density.
$\rho_G(t)$	Density	$\rho_G(t)=\lvert I_F(t)\rvert/\lvert I_G(t)\rvert$	Global information density.
$\rho_{GN}(t)$	Density	$\rho_{GN}(t)=\lvert I_N(t)\rvert/\lvert I_G(t)\rvert$	Segmental nodal/Global density.
$\rho_C(t)$	Density	$\rho_C =(\lvert I_F\rvert/\lvert I_N\rvert)^{\alpha}(\lvert I_F\rvert/\lvert I_G\rvert)^{\beta-\rho}$	Composite information density.

α	Weight	$\alpha \in [0,1]$	Weight of nodal density contribution.
β_ρ	Weight	$\beta_\rho \in [0, 1]$	Weight of Global density contribution
β_N	Parameter	$\beta_N \in \mathbb{R}$	Node-count exponent in fragmentation rule
ε_N	Threshold	$\varepsilon_N \in (0, 1]$	Nodal density threshold.
ε_G	Threshold	$\varepsilon_G \in (0, 1]$	Global density threshold.
ε_C	Threshold	$\varepsilon_C \in (0, 1]$	Composite density threshold.
ε	Generic threshold	$\varepsilon \in (0,1]$	Generic rejection threshold.
λ	Coefficient	$\lambda \in \mathbb{R}_{>0}$	Fragmentation intensity coefficient.
I_0	Constant	$I_0 \in \mathbb{R}_{>0}$	Reference information unit/scale.
$N(t)$	Natural number	$N(t) = \lvert E_{int}(t) \rvert$	Number of active internal nodes/elements used in fragmentation rule.
$E_{int}(t)$	Set	Internal/structural elements	Internal nodes up to terminal level.
$E_{term}(t)$	Set	Terminal Fragments	Terminal elements only.
$V_{ext}^{Required}(t)$	Set	Required external verifiers	Verifier set required to sustain decomposition.
ρ	Density	Generic information density	Used in rejection rule $\rho > \varepsilon \Rightarrow$ Reject(S).
χ	Threshold	$\chi > 0$	Critical payload-to-metadata threshold below which fragmentation becomes operationally inefficient or semantically weak.
η	Probability threshold	$\eta \in (0,1)$	Target upper bound for unauthorized reconstruction probability, used in $P(Rec^*) \leq \eta$.
μ	Expected value	$\mu = E[X] = qk\bar{\omega}$	expected weighted adversarial information
τ	Threshold value	$\tau \in (0,1)$	critical reconstruction threshold.

B.9 Entropy, Probability and Randomness

Symbol	Type / Domain	Definition	Meaning
H	Function	Entropy function	Shannon Entropy in bits.
$H(I_N)$	Entropy	Entropy of nodal information	Uncertainty of complete nodal information.
$H(I_N \lvert K_{VF})$	Conditional Entropy	Entropy of I_N given verifier knowledge	Residual uncertainty after observing Local knowledge.
I(A; B)	Mutual information	I(A;B)=H(A)-H(A|B)	Information shared between A and B.
$I(K_{VF};I_N)$	Mutual information	≈ 0 ideally	Verifier knowledge should reveal negligible nodal information.
p	Probability	p=1/n!	Random reconstruction probability for n distinguishable Fragments.
n!	Factorial	Product $1 \cdots n$	Number of permutations of n distinguishable Fragments.
D_{nu}	Function	Non-uniform distribution function	Distributes information into non-uniform Fragments.
$I_{F,j}(t)$	Information	Information of fragment j	j-th fragmental information quantity.
j, k	Indices	$j, k \in \mathbb{N}$	Fragment indices used in non-uniformity statements.
X	Random variable	$X = \Sigma_i Y_i \omega(D_i)$	Weighted adversarial information acquired by colluding verifiers.
ν	Function	$\nu : [0, 1] \rightarrow [0, 1]$	Non-decreasing function

			mapping adversarial knowledge to inference probability.
$View_{adv}$	Information	Adversarial observation	Total adversarial view, including compromised weighted fragments, observable structural metadata, and public invariant class. It does not include the hidden instantiated invariant structure C_{int}.
H_{∞}	Entropy function	min-entropy	A cryptographic measure that quantifies the worst-case predictability of a variable or fragment.
h_{min}	Entropy Threshold	$H_{min} > 0$	The minimum vital threshold of residual min-entropy required to guarantee the semantic non-inferability of the data.
C_{int}	Hidden invariant structure	$C_{int} = \langle \theta_C, R_{int}, B_{int} \rangle$	Internal instantiated invariant parameters, relational topology, and hidden binding map.
X_S	Random variable	Semantic payload	Variable representing the pure semantic information (the real data or $I_{payload}$ associated with the system.
M_S	Randome variable	Structural metadata	Variable representing the topological and coordination information ($I_{metadata}$) exposed by the system.
d_{miss}	Semantic data	Missing payload	The internal payload associated with critical fragments that are not under the direct control of the adversary.
γ_{top}	Boundede parameter	$\gamma_{top} \geq 0$	Maximum tolerance for information "leakage" between the coordination structure and the pure semantic data.
μ_t	probability distribution function	$\mu_t : \mathfrak{A}_{inj}(t) \rightarrow [0, 1]$	describes the assignment probability distribution over injective assignments. The distribution need not be uniform and may be weighted by risk, criticality, redundancy, or operational constraints.
H_{assign}	Assignment Entropy	$H_{assign}(k)=\log_2(k!)$	indicates the uncertainty in assigning each single task to a specific verifier v, considering the entire space of possible combinations starting from a defined number of atomic elements in the space Ter(t)

B.10 Operators, Logical Symbols and Set Symbols

Symbol	Type / Domain	Definition	Meaning
$\in$	Membership	$x \in A$	x belongs to set A.
$\notin$	Non-membership	$x \notin A$	x does not belong to set A.
$\subseteq$	Subset	$A \subseteq B$	A is subset of B.
$\subsetneq$	Proper subset	$A \subsetneq B$	A is strictly contained in B.
$\times$	Cartesian product	$A \times B$	Set of ordered pairs.
$\wedge$	Logical AND	$A \wedge B$	Both propositions hold.
$\vee$	Logical OR	$A \vee B$	At least one proposition holds.
$\Rightarrow$	Implication	$A \Rightarrow B$	A implies B.
$\Leftrightarrow$	Equivalence	$A \Leftrightarrow B$	A iff B.
$\neq$	Inequality	$A \neq B$	A is not equal to B.

$\approx$	Approximation	$A \approx B$	Approximately equal.
$\cong$	Structural equivalence	$A \cong B$	Equivalent as structured representation.
$\forall$	Universal quantifier	$\forall x$	For all x.
$\exists$	Existential quantifier	$\exists x$	There exists x.
Σ	Summation	$\Sigma_i a_i$	Sum over index i.
$\lvert A \rvert$	Cardinality or length	Cardinality of set A or bit-length of string A	Context-dependent size measure.
lim	Limit	$\lim_{x \to \infty} f(x)$	Asymptotic behavior.
$\lceil \cdot \rceil$	Ceiling	Smallest integer ≥ argument	Used to convert real-valued fragmentation expression to integer count.
$\log_2$	Logarithm	Base-2 logarithm	Information-scale growth term.
$\mathbb{N}$	Set	Natural numbers	Indices and counts.
$\mathbb{R}_{>0}$	Set	Positive real numbers	Positive coefficients and thresholds.
$\emptyset$	Empty set	No elements	Possible empty set symbol where needed.

B.11 Probabilistic Security Symbols

Symbol	Type / Domain	Definition	Meaning
X	Random variable	$X = \Sigma_i Y_i \omega(D_i)$	Weighted adversarial information
Y_i	Bernoulli indicator	$Y_i \in \{0,1\}$	Bernoulli adversarial assignment indicator
q	Probability / fraction	$q \in [0,1]$	Collusive assignment probability or collusive fraction
$\omega(D_i)$	Normalized weight	$\omega(D_i) = I(G;F_i)/H(G)$	Normalized informational utility of fragment F_i
$\bar{\omega}$	Average normalized weight	$\bar{\omega} = (1/k) \Sigma_i \omega(D_i)$	Average normalized fragment utility .
τ	Threshold	$\tau \in (0,1)$	Critical reconstruction threshold: minimum normalized useful information required for unauthorized reconstruction.
μ	Expected value	$\mu = E[X] = qk\bar{\omega}$	Expected weighted adversarial information under independent assignment
Rec^*	Event	$Rec^* = \{X \geq r\}$	Unauthorized reconstruction event
k	Natural number	$k \in \mathbb{N}$	Number of terminal fragments
r	Threshold value	$r = \tau k$	Non-normalized reconstruction threshold
H(G)	Entropy	$H(G) > 0$	Entropy of the random variable representing globally admissible configurations

B.12 Dependent Collusion Symbols

Symbol	Type / Domain	Definition	Meaning
C_{comp_i}	Event	C_{comp_i} = "F_i is compromised"	Event that the i-th critical fragment F_i is directly controlled or successfully inferred by the adversarial coalition.
m	Natural number	$m \in \mathbb{N}$	Number of critical low-inference fragments
p_{inf}	Probability	$p_{inf} \in [0, 1]$	Inference probability
p_{comp}	Probability	$p_{comp} = q + (1-q)p_{inf}$	Single-fragment compromise probability under dependent collusion.
η	Probability threshold	$\eta \in (0, 1)$	Target attack probability
ν	Function	$\nu : [0, 1] \to [0, 1]$	Non-decreasing function mapping adversarial knowledge

			to inference probability.
K_{adv}	Adversarial knowledge	$K_{adv} = \Sigma_{i \in A}\, \omega(D_i)$	Adversarial knowledge
A	Index Set	$A \subseteq \{1,...,k\}$	Set of fragment indices controlled by the adversarial coalition.

B.13 Design / Feasibility Symbols

Symbol	Type / Domain	Definition	Meaning
I_{min}	Information threshold	$I_F \geq I_{min}$	Minimum semantic information required for honest verification
m_{min}	Natural number	$m_{min} = \lceil \ln(\eta)/\ln(p_{comp}) \rceil$ $0 < p_{comp} < 1, \eta \in (0,1)$	Minimum number of critical fragments required to satisfy $P(Rec^*) \leq \eta$
m_{max}	Natural number	$m_{max} \in \mathbb{N}$	Maximum number of critical fragments compatible with $I_F \geq I_{min}$
χ	Threshold	$\chi > 0$	Critical payload-to-metadata threshold
Γ	Ratio	$\Gamma = m_{max}/m_{min}$	Feasibility condition: the maximum semantically admissible fragmentation must be at least as large as the minimum critical fragmentation required by the target security bound.

Appendix C — Case Study

Case Study: Physical Instantiation and Epistemic Isolation of Global Invariants

To operationally illustrate the Non-Reducibility Principle and the resilience of CNVS against out-of-band collusion, consider a physical-world validation instance. A user submits the following target state S to the CNVS system for mapping and validation:

“Apartment A located in Building B.”

The spatial and semantic parameters defining the admissible existence of this state include local descriptive data, such as city, street class, ceiling height, internal layout, facade color, balcony structure, and perimeter configuration. In addition, the system defines global invariant conditions, such as the exact floor level and the total floor area.

Decomposition and Assignment: The Measurement Grid

The CNVS system does not send an external verifier the query:

“Check whether Apartment A is located on the third floor and has a total area of 120 m².”

Instead, the decomposition operator $\mathfrak{D}(s)$ fragments the target state into atomic and apparently unrelated measurement tasks. External verifiers receive only local measurement grids and task specifications.

For example:

Verifier A is asked to measure the ceiling height of a room at a given coordinate.

Verifier B is asked to identify the facade color or verify the presence of a perimeter courtyard.

Verifier C is asked to detect the number of balconies visible from a given side of the building.

Out-of-Band Collusion: The Attack Vector

Assume a worst-case scenario in which several verifiers collude through external encrypted channels. By combining their assigned fragments, they may reconstruct a partial semantic image of the target: for instance, that the system is mapping a green building, free on four sides, with high ceilings and double-aspect exposure.

The coalition may then attempt to inject falsified data or force the admissibility of a similar structure. However, the colluders only know the local tasks assigned to them. They do not know the complete relational graph, the hidden global invariants, or the acceptance conditions used by the system.

The Invariant Trap: Global Veto

The variables “floor level” and “total floor area” may be treated as global invariants Inv_C. These invariants are not directly exposed as terminal verification tasks. No local verifier is asked to validate the total square footage or the global elevation condition as such.

Local verification may still succeed on the assigned fragments. For example, a ceiling height, facade color, or balcony count may be locally consistent, allowing the corresponding Local Verification Function to return a positive result:

$\pi_1(V_{Local}(s_i)) = 1$.

However, local validity is not sufficient for global admissibility. During reconstruction, the Global Verification Function V_G evaluates relational consistency and invariant satisfaction across the recomposed state. If the reconstructed topology fails to satisfy the hidden invariant conditions, such as the required total area or floor level, the candidate state is rejected:

$V_G(S) = 0$.

Thus, even when colluding verifiers compromise the local instructions assigned to them, they do not necessarily compromise the global admissibility conditions. Under the CNVS assumptions of residual non-inferability, hidden

invariants, and bounded leakage, the probability that falsified local data also satisfy the unobserved global constraints remains negligible.

Conclusion

The colluding verifiers act inside a game whose complete victory conditions are not exposed to them. CNVS does not rely only on the correctness of isolated local observations. It requires the simultaneous satisfaction of local verification, relational consistency, and global invariants. This demonstrates why global validity is not reducible to the mere conjunction of terminal local validations and why the Global Veto remains structurally necessary.

Appendix D — Terminological Glossary

This glossary provides a concise terminological reference for the Closed Native Verification Systems (CNVS) framework. Its purpose is not to replace the formal definitions introduced in the main text, but to clarify the meaning of the principal terms used throughout the theory.

Admissibility

Admissibility denotes the condition under which a structured element or candidate state is accepted by the System. At the terminal level, admissibility depends on local verification. At the state level, admissibility depends on global verification.

Adversarial Inference

Adversarial inference is the attempt by a colluding or malicious actor to reconstruct unknown fragments, relations, or global structure from the information already under its control. In CNVS, adversarial inference is limited by fragmentation, random assignment, restricted knowledge, and invariant-based global verification.

Adversarial Knowledge

Adversarial knowledge is the information effectively available to an adversarial coalition. It may include directly controlled fragments and any additional information inferred from them. In the probabilistic model, adversarial knowledge is measured through the weighted informational contribution of compromised fragments.

Attribute / Attribute Class

An attribute is the specific class of observable property assigned to an external verifier. It defines what the verifier must observe, measure, or report.

Examples include weight, quantity, color, distance, timestamp, location, or any implementation-specific measurable feature.

Axiom I — Verification-Dependent Existence

Axiom I states that no terminal element or state is admitted into the System merely by declaration. Existence inside the System depends on verification. A terminal element is admitted only if local admissibility holds; a state is valid only if global verification holds.

Axiom II — Randomized Non-Uniform Fragmentation

Axiom II states that admissible information is recursively decomposed into non-uniform fragments and assigned randomly to external verifiers. The decomposition is finite, reconstructible, type-preserving, and designed to reduce the knowledge available to any single verifier.

Axiom III — Non-Reducibility of Global Validity

Axiom III states that global validity cannot be reduced to the conjunction of local verifications. Even if all terminal local checks are satisfied, the candidate state may still be rejected if relational consistency or global invariants fail.

Closed Native Verification System (CNVS)

A CNVS is a formal verification system in which state membership, validity, and evolution depend on internal verification processes. External agents provide observational evidence, but the System internally determines whether the candidate state is admissible.

Collusion

Collusion is the coordinated behavior of multiple external verifiers or adversarial agents who share their assigned information in order to reconstruct or violate the System.

ρ_{GN}

ρ_{GN} is the relative nodal density, measuring the weight of nodal information relative to global information. It expresses how much a node contributes to the global semantic information.

Composite Information Density ρ_C

Composite Information Density is the combined density measure that integrates both local/nodal density and global density. It expresses how much a fragment contributes relative to both its node and the global system. It is used by the System to regulate fragmentation depth and density thresholds.

Conditional Entropy

Conditional entropy measures the uncertainty that remains about one informational object after another informational object is known. In CNVS, it is used to express how much uncertainty about nodal or global information remains after verifier-accessible knowledge is observed.

Convergence Function

The convergence function is the local comparison mechanism between internally declared data and externally observed data. It evaluates whether the observation reported by the external verifier is compatible with the value held internally by the System for the assigned attribute.

Critical Fragment

A critical fragment is a fragment whose absence prevents reliable unauthorized reconstruction of a globally valid state. If a critical fragment is also non-inferable, then its absence acts as a practical obstacle to adversarial reconstruction.

Critical Mass

Critical mass denotes the minimum amount of useful, coherent, and

structurally relevant information required for an adversarial coalition to attempt reconstruction. It is related to the reconstruction threshold and to the number of critical fragments needed to compromise the System.

Critical Threshold τ

The critical threshold τ is the minimum normalized amount of useful information required for unauthorized reconstruction to become possible within the probabilistic model. If the expected adversarial information remains below τ, successful unauthorized reconstruction becomes asymptotically unlikely under the stated assumptions.

Cross Entropy

Cross entropy measures the informational inefficiency of using one probability distribution to represent another. In the CNVS discussion, it is used to explain the difference between an actual distribution and an approximating or adversarial distribution.

Data D_i

D_i denotes the internally declared data associated with a structured element or terminal fragment. In CNVS, D_i remains inside the System and is not disclosed directly to the external verifier.

Decomposition

Decomposition is the recursive process by which a structured element is divided into finite, type-preserving, non-uniform fragments. Decomposition changes informational granularity without changing the structural nature of the element.

Dependent Collusion

Dependent collusion refers to a coordinated adversarial model in which compromise events are not assumed to be independent. Instead, the

probability of compromising an additional critical fragment may depend on previously compromised fragments, but remains bounded by a conditional compromise probability.

Density Thresholds ϵ_G, ϵ_N, ϵ_C

Density thresholds are admissible upper bounds imposed by the System on information density. ϵ_G controls global density, ϵ_N controls nodal density, and ϵ_C controls composite density. When these thresholds are approached or exceeded, the System may impose additional fragmentation.

Entropy

Entropy measures uncertainty. In CNVS, entropy is used to describe the uncertainty faced by an adversary attempting to reconstruct the correct configuration of fragments, relations, or global admissible states.

Emergent Security

Emergent security is the property by which System security increases as a structural consequence of scale, fragmentation, restricted knowledge, and global invariant checks. It is not merely added externally, but emerges from the internal architecture of the CNVS model.

External Verification Agent

An external verification agent is an actor assigned by the System to observe or measure a specific attribute. The agent does not receive the internal declared payload D_i, but only the task specification needed to produce an observation.

Fragment

A fragment is a terminal structured element produced by recursive decomposition. It carries only a restricted portion of the original information and may differ in informational weight from other fragments.

Fragmental Information I_F

Fragmental Information is the semantic information carried by a terminal fragment. It represents only a local portion of the decomposed information and does not by itself contain the complete nodal or global structure.

Global Information I_G

Global Information is the semantic information associated with the entire System or global state. It represents the complete semantic payload at the global level, excluding or distinguishing from the broader structural information plane when necessary.

Global Information Density ρ

Global Information Density measures the weight of a fragment relative to the Global Information of the System. It expresses how much a fragment contributes to the global semantic information.

Global Veto Principle

The Global Veto Principle states that global validity is not determined by the numerical majority of locally valid observations. Even a single honest, essential, and non-inferable observation may prevent unauthorized reconstruction if its contribution conflicts with the collusive reconstruction and causes relational consistency or global invariant checks to fail. In this case, the candidate state is rejected by V_{Global}.

Global Verification V_{Global}

Global Verification is the internal System-level verification function that evaluates whether a candidate state satisfies relational consistency, recursive reconstruction constraints, and deterministic invariants. It is not performed by external verifiers.

Graph

A graph is the relational structure formed by System elements and their admissible relations. In CNVS, structured elements correspond to nodes, while admissible directional relations correspond to edges. The graph captures both structural dependencies and recursive decomposition topology.

Granularity

Granularity is the degree of informational resolution produced by decomposition. Higher granularity means that information is divided into smaller, more specific fragments, increasing epistemic restriction but also requiring preservation of minimum semantic content.

I — Semantic Information Plane

I denotes the semantic information plane. It refers to payload content: the meaning-bearing informational content associated with the object, claim, or structured element.

𝐼 — Systemic Structural Information Plane

𝐼 denotes the broader systemic structural information plane. It includes not only semantic payload, but also metadata, relational topology, reconstruction dependencies, invariant structure, and graph-level coordination information.

I_0

I_0 is a baseline informational scale used in the fragmentation-control model. It prevents the fragmentation function from behaving pathologically at very low information values and helps normalize the effect of growing global or nodal information.

I_{min}

I_{min} denotes the minimum amount of information required for a fragment to

preserve operational meaning. Below this threshold, the fragment becomes too poor in semantic content for honest verification to remain possible.

Independent Collusion

Independent collusion is a simplified probabilistic model in which fragment assignment or compromise events are treated as independent Bernoulli trials. It is useful for deriving idealized asymptotic security bounds.

Inference Probability p_{inf}

Inference probability is the probability that a fragment not directly controlled by colluding agents can nevertheless be inferred from compromised information. It represents the residual inferential power of the adversarial coalition.

Knowledge

Knowledge is the portion of information accessible to a given actor. In CNVS, knowledge is not identical to information: the System may contain information that is not accessible to an external verifier or adversary.

Knowledge Restriction Principle

The Knowledge Restriction Principle states that verifier-accessible knowledge remains structurally insufficient for reconstructing nodal or global information. As the System grows and fragmentation increases, local knowledge becomes asymptotically negligible relative to global information.

K_{adv}

K_{adv} denotes adversarial knowledge or the informational weight controlled by an adversarial coalition. It may be used to model how the probability of inference increases as the adversary controls more useful information.

K_{VF}

K_{VF} denotes the knowledge accessible to an external verifier. It is generally smaller than fragment-level knowledge because the verifier does not receive the internal declared payload D_i.

λ

λ is the fragmentation-control parameter. It regulates the intensity of decomposition and affects the number of fragments generated by the System.

Local Verification V_{Local}

Local Verification is the verification function applied at the terminal fragment level. It evaluates formal admissibility and convergence between internally declared data and externally reported observation for the assigned attribute.

Metadata

Metadata is the operational information required for fragment identification, assignment consistency, relational reconstruction, and verification coordination. It belongs to the systemic structural information plane rather than to the pure semantic payload.

Nodal Information I_N

Nodal Information is the semantic information associated with a node and its descendant structure. It lies between fragmental information and global information.

Nodal Information Density ρ_N

Nodal Information Density measures the weight of a fragment relative to its node. It expresses how much a terminal fragment contributes to the corresponding nodal information.

Object

An Object is anything whose existence, validity, or factual consistency can be represented by verifiable information. It may be physical, abstract, documentary, historical, or otherwise representable through admissible evidence.

Obs_i

Obs_i denotes the observation, measurement, certificate, or reported value produced by the external verifier. It is compared internally with D_i through the convergence function.

Operational Flow

The operational flow is the ordered process through which an object is encoded, structured, decomposed, assigned to verifiers, locally verified, recursively reconstructed, globally checked, and either admitted or rejected.

Payload

Payload is the semantic content inherited from the original object or claim. It is the meaning-bearing informational component, distinct from metadata and structural coordination information.

Properties P_i

Properties are logical evaluations produced by verification functions. Examples include valid, invalid, blocked, transferable, verified, suspended, accepted, questionable, corrupted, or inconclusive.

Reconstruction

Reconstruction is the process by which decomposed fragments are recomposed into higher-level structures or candidate global states. In CNVS, reconstruction is admissible only if relational consistency and invariants are satisfied.

Relations R

Relations are admissible directional links among structured elements. They define how elements are connected within the System graph and are essential for global consistency and reconstruction.

Security

Security is the capacity of the System to resist unauthorized reconstruction, adversarial inference, collusion, and invalid state admission under the stated assumptions.

State

A State is the structured configuration of the System at a given time. It includes admitted elements, relations among them, and deterministic invariants. A candidate state becomes valid only if global verification succeeds.

State Rejection

State rejection occurs when a candidate state fails local admissibility, relational consistency, invariant satisfaction, or global verification. A rejected state cannot become a valid state of the System.

State Transition Law

The State Transition Law states that a candidate successor state is admitted only if it satisfies global verification. State evolution is therefore verification-dependent and constrained by the System invariants.

τ

τ is the critical reconstruction threshold. It denotes the minimum normalized amount of useful information required for unauthorized reconstruction to become possible in the probabilistic model.

χ

χ denotes the critical payload-to-metadata threshold used to ensure that fragmentation remains operationally meaningful. If metadata overhead dominates semantic payload, decomposition becomes inefficient or useless.

Verifier

A verifier is an external agent assigned to perform a restricted observation or measurement. The verifier contributes evidence but does not determine global validity.

Veto

A veto is the blocking effect produced by a missing, inconsistent, or non-inferable essential contribution. In CNVS, veto effects arise from global invariants and relational consistency checks.

Weighted Fragment Utility $\omega(D_i)$

Weighted Fragment Utility measures how informative a fragment is with respect to the global admissible configuration. It is often normalized so that low values indicate weak reconstructive utility and high values indicate strong reconstructive utility.

η

η denotes a target upper bound for the probability of unauthorized reconstruction. It is used in the design formula for determining how many critical low-inference fragments are needed to reach a desired security level.

ϵ_{At_i}

ϵ_{At_i} denotes the admissible tolerance threshold associated with a specific attribute. It is used by the convergence function when comparing quantitative declared data and observed values.

END DOCUMENT

Author: Massimo Comitato

03.07.2026, Italy (Milano)