

Fractal Ontological Architectures

Autonomy, Relations, and Recursive Composition

Cédric Abunar

9 October 2026 · v1.2.1

Contents

Fractal Ontological Architectures	2
Foundational Manifesto — Autonomy and Relations	2
Part I — Structural Extension Under Locality Assumptions	4
Part II — Architectural Choices and Their Conditions	6
Part III — Comparison with Distributed Design Patterns	8
Part IV — Failure Questions and Conditional Results	10
Part V — Making the Model Understandable	11
Part VI — States, Contexts, and Local Transitions	13
Part VII — Responsibility, Ethics, and Scope	19
Appendix A - Formal Axioms and Invariants of Fractal Ontological Architectures	20
Appendix B — Alternative Designs and Model Boundaries	23
Appendix C - Boundary Entities (Normative)	24
Appendix D - Interpretation Constraints (Normative)	27
Appendix E — Influence and Participation (Normative)	29
Appendix F — Emergence Is Not Authority (Normative)	30
Appendix G — Enigma: Conditions of Containing Coherence	32
Appendix H — Sphaera: A Situated State and Dimension Profile	33
Glossary - Fractal Ontological Architectures (FOA)	37
Notation and Formal Conventions - Fractal Ontological Architectures (FOA)	41

Fractal Ontological Architectures

Autonomy, Relations, and Recursive Composition

Cédric Abunar · Independent researcher · Québec, Canada

Version: v1.2.1 · Revised research edition · 9 October 2026

Publication: <https://cedricabunar.com/en/reading/foa/>

Status: Research corpus; not peer reviewed. Part VI develops identity dimensions, context-relative state pairs, partial local transitions, freshness, and incarnation continuity. Appendix H presents a conceptual profile inspired by Sphaera, without an implementation audit or source-tree provenance. This edition revises the public presentation and verification metadata. The scoped mathematical results and diplomatic framing remain in place.

Scope: The axioms define a model. Results about bounded local roles and finite recursive extension depend on their hypotheses. Usefulness, physical scalability, safety, and implementation conformance require separate evidence.

The term FOA here means **Fractal Ontological Architectures**. In the SERIS paper, FOA means **Field of Attention Operator**. The terms name different concepts.

License: CC BY-ND 4.0. Earlier sealed releases retain their versions and contents.

Foundational Manifesto — Autonomy and Relations

Fractal Ontological Architectures (FOA)

FOA explores a class of architectures in which autonomous entities compose an ensemble through local, explicitly described relations. The same rules apply at each level of composition.

This manifesto introduces the research intention. The axioms in Appendix A define the model; practical behavior must be examined separately.

1. Intent

The research asks how presence, absence, and local action can coexist without a higher-level entity prescribing the outcomes of another.

Architecture choices answer different needs. Execution guarantees, coordinated recovery, and shared state can be valuable when a product requires them. FOA studies a different set of assumptions and the uses for which they may be appropriate.

2. Core Principle

Within the FOA model, no entity may impose an outcome on another.

A higher level may describe what it observes. The observed entity retains responsibility for its local behavior. This is a definition of the architectural class, rather than a judgment about the people who design other systems.

3. Composition

Each entity has a local identity and a declared scope of observation. A Genesis observes its immediate neighborhood without issuing commands. Adding a level repeats these rules instead of introducing a new decision authority.

4. Presence and Silence

A received observation does not establish an obligation to act. Silence is an admissible state. It does not establish availability, failure, or intention beyond what was actually observed.

A product that needs a timely response must account for that need explicitly when assessing whether this model is suitable.

5. Responsibility

Local responsibility identifies the behavior and decisions owned by an entity. Human, organizational, and legal responsibilities still need to be assigned explicitly; the model does not discharge them.

6. Scope of the Research

FOA does not promise delivery, execution, convergence, operational safety, or an absence of harm. Its structural constraints exclude certain forms of imposed coordination from the model. That exclusion is not an empirical guarantee about every implementation.

Physical resources, network behavior, persistence, security, and usability remain questions to examine.

7. Emergence

Entities may interact, develop conventions, and form patterns. A pattern becoming common or persistent does not, within the model, give it authority over its participants.

How to distinguish local participation from indirect influence remains an important question for implementation and interpretation.

8. Research Direction

The aim is to describe the conditions under which autonomous parts can form a coherent ensemble, then examine the cost, usefulness, and limits of that composition.

Autonomy in the parts, coherence in the whole.

Part I — Structural Extension Under Locality Assumptions

Abstract

We define a recursive model of autonomous cells and observational Genesis entities. Under bounded local work and fixed branching assumptions, a finite structure can be extended to greater depth without introducing an entity that must inspect the entire ensemble. This is a result about model composition. It does not establish infinite physical capacity, constant total cost, or operational reliability.

1. Research Question

Which responsibilities grow when an ensemble gains a level? FOA explores a model in which higher-level observation remains local and does not acquire authority over the observed entities. The question concerns the structure of dependencies; resource budgets and product requirements must be evaluated alongside it.

2. Definitions

A **cell** has local state, a local capacity for action, and a declared neighborhood. A **Genesis** is a cell whose role is to describe the presence and locally observable coherence of its immediate children. It has no authority to order their execution.

A **level** is a finite grouping of cells or substructures with an associated Genesis. A cell has no transitive view of descendants beyond the declared adjacency relation.

A **Fractal Ontological Architecture** repeats these rules at each level. It excludes system-wide command, mandatory synchronization, and an authoritative global view from the modeled relations.

3. Explicit Hypotheses

H1 — Bounded local work. The number of adjacent relations, the size of each observation, and the work of one local evaluation are bounded independently of the total number of cells. Bounded input size alone does not establish a runtime bound; termination and a bound on evaluation work are also assumed.

H2 — Non-imposition. A cell may act locally using its observations, but cannot compel another cell to act, recover, or converge. The absence of an external command does not imply the absence of local computation or automation.

H3 — Observational composition. Inter-level observations describe an immediate neighborhood. They do not convey command authority or aggregate a recursively complete global state.

H4 — Bounded descriptive exchange. Observations may be partial, differential, or absent. The model assumes bounded local exchange; it does not infer bounded traffic from topology alone.

These hypotheses concern the model. Shared power, hosts, transport, storage, and administrative authority are external dependencies that can still affect many cells simultaneously.

4. Growth Model

For a regular tree with branching factor $k \geq 2$ and depth n , the number of leaves is

$$N_{\text{leaf}}(n) = k^n.$$

The total number of vertices, including observational vertices, is

$$N_{\text{total}}(n) = \sum_{j=0}^n k^j = \frac{k^{n+1} - 1}{k - 1}.$$

These counts describe a finite regular construction. They do not describe an arbitrary tree using only its average branching factor. Total resources grow with the number of vertices even if local evaluation remains bounded.

5. Local Results

Lemma 1 — Bounded Work Per Evaluation

Under H1, one local evaluation has cost $O(1)$ with respect to total size N , with the local budget held fixed. This says nothing about the number of evaluations per second, total communication, or the complexity of unrestricted domain reasoning.

Lemma 2 — No Required Global Inspection

Under H1 and H3, no modeled entity must inspect all N vertices to carry out its local role. This excludes a required global inspection step from the model. It does not exclude congestion or bottlenecks in a shared implementation substrate.

Lemma 3 — No Imposed Recovery Chain

Under H2 and H3, an absence observation does not itself order recovery in another entity. Consequently there is no mandatory recovery-command chain in the modeled relations. Shared-resource failures, loss of useful outputs, and correlated local reactions remain possible.

6. Proposition — Finite Recursive Extension

For every finite depth n , a regular FOA structure satisfying H1–H4 can be extended to depth $n + 1$, provided new cells and their local resources can be supplied and the same hypotheses continue to hold.

Proof sketch. Start with k disjoint copies of a valid depth- n structure. Add a Genesis with one declared observational relation to each copy. Its degree is k , fixed independently of n . Its observations remain bounded and descriptive. The new composition introduces no system-wide command or mandatory global inspection. Thus the local hypotheses are preserved. Repeating the construction gives a structure for any chosen finite depth. $\square$

The proposition concerns an unbounded family of finite abstract structures. It is not a claim that one physical system can grow indefinitely or that an infinite implementation exists.

7. Information and Usefulness

Let I_n denote the distinctions available in a particular level's observations. If an added level merely transforms I_n without providing additional useful distinctions for a stated task, it may be informationally redundant.

This is a criterion to evaluate against a use case, rather than a universal information-theoretic saturation theorem. Utility, observation accuracy, physical feasibility, and structural extensibility are separate properties.

8. Discussion

Locality can avoid requiring every cell to participate in a global decision. In exchange, the model gives up guarantees that some products need. A useful implementation must specify its traffic rates, persistence needs, shared dependencies, security boundaries, and handling of missing information.

9. Conclusion

The recursive construction preserves bounded local roles under its stated assumptions. Questions of deployability, recovery, and useful scale remain open to measurement. A counterexample should identify the relevant assumption, relation, or implementation dependency so the disagreement can be examined.

10. Transition

Part II considers the needs that motivate different architectural choices. Those needs help explain when FOA's assumptions may be useful and when a different model is appropriate.

Part II — Architectural Choices and Their Conditions

11. Scope

This part examines choices about coordination, state, causality, and recovery. It does not attribute architectural differences to a limitation of human reasoning. Existing systems often embody legitimate requirements for delivery, accountability, compatibility, and continuity.

12. Starting with the Need

A design can require a response, an ordered history, a common decision, or a corrective procedure. These requirements introduce dependencies worth describing. The question is whether their cost and behavior fit the need, rather than whether the people who chose them were mistaken.

13. Orchestration

An orchestrator can make ordering and ownership explicit. If one component must individually examine n participants during an evaluation, that evaluation involves at least n participant examinations. Whether it saturates depends on arrival rates, batching, parallelism, capacity, and the required service level.

FOA excludes an orchestrator with authority over other cells from its modeled inter-level roles. This is a change of assumptions, not a general proof that orchestration is ineffective.

14. Shared State

Shared or agreed state can support transactions and a consistent interpretation of a decision. Maintaining it has a protocol-dependent cost. FOA uses local observations and does not require a single authoritative global state. Applications that need agreement must account for that requirement explicitly.

15. Causal Execution Chains

For k independent steps, all required for success, with failure probabilities p_i ,

$$P(\text{failure}) = 1 - \prod_{i=1}^k (1 - p_i).$$

If $p_i = p$ with fixed $0 < p < 1$, the probability approaches one as k grows. This conclusion requires the independence and fixed-probability assumptions. It does not hold for every possible sequence of p_i , correlated failures, redundancy, or recovery policies.

FOA studies relations without mandatory cross-entity execution. A locally valid structure can still be unable to provide a useful result when a required observation is missing.

16. Recovery and Control

Monitoring, restart procedures, and feedback control can provide valuable continuity. Their stability depends on the controlled dynamics, delays, gains, noise, and available resources. Removing an imperative relation does not by itself prove stability or safety.

Within FOA, a Genesis describes observations and does not prescribe recovery. A cell's own local reaction and the operation of its substrate require separate examination.

17. Comparison of Requirements

Requirement	A possible coordinated design	FOA model choice
Response	An explicit service obligation	Response remains optional
Agreement	A declared consensus or transaction scope	Local observations suffice for modeled existence
Recovery	An assigned operator or controller	No recovery command from Genesis
Silence	Interpreted under a service contract	Admissible without a compulsory reaction

The choice depends on the product and its context.

18. Tradeoffs

Locality reduces the information required by each modeled role. It also limits what that role can know or guarantee. There is no general impossibility theorem here covering all combinations of coordination, consistency, and scale.

19. Examining Objections

Questions about usefulness, resource costs, missing observations, security, and recovery are valid research questions. A discussion is more productive when the requirement and the relevant hypothesis are both explicit.

20. Conclusion

FOA offers a particular way to frame autonomy and composition. Its value must be demonstrated for specific needs alongside the constraints faced by the people who would design, operate, and use it.

Part III — Comparison with Distributed Design Patterns

21. Purpose

The comparisons below concern selected design patterns. Each family contains many variants. Its name alone does not determine compatibility with FOA, performance, authority, or failure behavior.

22. Reference Patterns

We consider service interfaces, actors, managed functions, event streams, and desired-state controllers. Each provides useful mechanisms for particular requirements.

23. Services and Interfaces

An API can make a capability and its expected responses clear. Some service compositions rely on synchronous calls, shared transactions, or a mandatory response; others use independent stores and asynchronous exchange. A response obligation between modeled cells would differ from FOA's optional-action semantics. The presence of an API alone is insufficient to decide compatibility.

24. Actors

Actors provide a way to describe concurrent entities communicating by messages. Supervision and delivery behavior depend on the particular language or runtime. A supervisor with authority to restart another modeled entity would depart from FOA's non-imposition axiom. Message passing and local state can still be useful implementation mechanisms.

25. Managed Functions

A managed-function platform commonly owns placement, invocation, and infrastructure lifecycle. An evaluation of FOA must say whether those mechanisms belong to the modeled entity relations

or to an explicitly separated substrate. Externalizing infrastructure does not eliminate its shared dependencies or responsibilities.

26. Events and Streams

Streams can preserve an ordered record and support recovery or replay. FOA observations describe a local view and do not require universal agreement on a history. A stream can carry descriptive information; compatibility depends on whether its semantics introduce mandatory processing, an authoritative view, or cross-level convergence.

27. Desired-State Controllers

A controller may compare an observed state with a declared target and take corrective action. That responsibility can be useful and explicit. A Genesis in the FOA model has a different role: it describes its neighborhood without imposing the target state of another entity.

28. Comparison Table

Pattern	Capability it may provide	Question for FOA interpretation
Service interface	An explicit interaction contract	Does reception compel another entity to act?
Actor runtime	Concurrent local behavior	Does supervision introduce modeled authority?
Managed function	Infrastructure-managed execution	Where does lifecycle authority belong?
Event stream	Ordered history and replay	Does processing require shared convergence?
Desired-state controller	Automated reconciliation	Is another entity's outcome prescribed?

29. Research Question

The central question is how entities can remain locally understandable when reception, silence, or disappearance does not carry a compulsory reaction. Answering it requires evaluating the actual relations and the needs of the application.

30. Conclusion

FOA's constraints describe a research model rather than a ranking of distributed architectures. A comparison should name the particular semantics, guarantees, and substrate assumptions being compared.

Part IV — Failure Questions and Conditional Results

31. Scope

This section examines situations that can affect distributed systems. It does not establish that every architecture outside FOA must fail, nor that a FOA implementation is immune to cascading failures.

32. Questions to Examine

Coordination load, chains of dependencies, agreement protocols, feedback dynamics, and interpretation of observations all deserve attention. Their behavior depends on explicit workloads and implementation conditions.

33. Coordination Load

If a coordinator must inspect n states, the number of inspections grows with n . With a fixed inspection rate $r > 0$, a sequential full scan takes at least n/r units of time. This is a conditional work bound. It is not a latency theorem independent of hardware, workload, or parallelism.

FOA's bounded local roles avoid requiring this full scan at one modeled entity. A shared transport or host can still saturate.

34. Dependent Steps

Under independent per-step failures with fixed probability p , the probability that all k required steps succeed is

$$P(\text{success}) = (1 - p)^k.$$

This simple model highlights a cost of mandatory dependency chains. Real systems may introduce retries, redundancy, correlated risks, or different success criteria, which require different models.

35. Agreement and Convergence

A protocol that informs every participant of a new value has work associated with those participants. The message, time, and resource costs depend on the network model and protocol. Removing required global agreement changes the capability offered; it does not establish that agreement protocols inevitably collapse.

36. Feedback Dynamics

Feedback stability must be derived from a specified dynamical model. There is no general instability test of the form $g d \varepsilon > 1$ applicable to arbitrary gains, delays, and measurement noise. That heuristic from earlier editions is withdrawn.

A FOA observer does not order corrective action. Local policies and substrate controls may still form feedback loops and must be evaluated.

37. Semantic Interpretation

Explicit definitions can make interpretation easier to discuss and test. They do not prevent implementation mistakes, version mismatches, missing context, or disagreements. Presence, silence, and observation age must retain their declared meanings.

38. Scope of Exclusion

Excluding a command edge from a model excludes the failures specific to that command edge. It does not exclude shared-resource faults, security failures, loss of useful observations, or other channels of interaction. The general inevitability and immunity claims of earlier editions are withdrawn.

39. Evaluation Direction

A prototype should identify its shared dependencies, traffic limits, interpretation rules, and useful behavior when observations are incomplete. Tests should cover correlated faults as well as individual disappearances.

40. Conclusion

The model makes particular dependency choices explicit. Operational reliability and safety remain properties to establish for each implementation and use case.

Part V — Making the Model Understandable

41. Purpose

This part considers how to explain FOA to readers with different responsibilities. It makes no empirical claim about universal cognitive limitations or the motives of people who question the model.

42. Begin with Context

A reader may reasonably expect a service to respond, a procedure to recover an interrupted task, or a record to support an audit. Those expectations often arise from the work that the system must support. They deserve to be understood before a different model is proposed.

43. Presence and Action

Explain what counts as observed presence and what information a local action needs. An entity being present does not, within FOA, imply that it must perform an action for another entity.

44. Observation and Intervention

Identify the observer, its view, and its permitted effects. Explain who operates the underlying infrastructure and where any recovery responsibility belongs.

45. Histories and Local Views

Sequence diagrams can explain an example interaction. Invariants and state descriptions explain which conditions apply beyond that example. These tools can complement one another.

46. Local Knowledge

A local observation has a scope and an age. It may be incomplete or stale. Coexistence of local views does not prove factual agreement, global correctness, or a complete view of the ensemble.

47. Silence

Silence can have several causes. Its admissibility means that the model does not automatically order a corrective action. A product's requirements for timeliness and missing information remain explicit questions.

48. Scale

Explain a small, bounded example before extending its construction. Show which local responsibilities stay fixed and which total costs grow. Large examples need measured resource and communication budgets.

49. Questions and Disagreement

An objection may identify an unclear definition, an unsupported claim, a missing requirement, or an implementation risk. These are useful contributions to the research. A formal model is strengthened by making such questions examinable.

50. Reading Guide

Question	Material to examine
What exists?	Cell, logical identity, and incarnation
What is observed?	Neighborhood, presence, age, and silence
What may act?	Local conditions and non-imposition
What does composition add?	A bounded observational level
What remains to establish?	Usefulness, resources, safety, and implementation behavior

51. Teaching Direction

Use a concrete example, explicit hypotheses, a comparison with a familiar requirement, and a small experiment. Preserve the reasons for existing choices so that the reader can judge the proposed change.

52. Conclusion

Understandability is part of the work. The model should provide enough context for people to question its assumptions and assess its usefulness.

53. Research Status

The recursive extension argument is conditional. The comparisons concern selected semantics. The failure discussion identifies questions rather than a general impossibility theorem. The remainder of the corpus specifies constraints and open directions for implementation and interpretation.

Part VI — States, Contexts, and Local Transitions

54. Purpose and Status

FOA needs to distinguish an entity's existence, its observations, and its ability to act. A cell can continue to exist while its operational conditions are unmet. A new context can require renewed observation without requiring a new incarnation.

This part develops a state framework for candidate implementations. The distinctions are part of this edition's model; particular state names, guards, and storage mechanisms belong to an explicitly declared implementation profile. Appendix H illustrates these questions with a conceptual profile inspired by Sphaera. Neither the framework nor that example establishes implementation conformance or operational safety.

55. Identity, Dimensions, and Composition

Three questions require separate answers:

Question	Model distinction	What it does not establish
What kind of cell is this?	Logical type, <i>geminae</i>	A live or reachable occurrence
Where is this responsibility situated?	Situated identity, when a profile defines one	Its current operational state
Which occurrence is executing?	Unique incarnation, <i>meta</i>	Permanent continuity of that occurrence

A profile may describe dimensions such as capability, role, technology, origin, application, environment, or execution substrate. These are coordinates used to distinguish identities and contexts. They are not automatically independent variables: admissible combinations may form a constrained subset of a Cartesian product.

The composition level n answers a different question: where an entity sits in the recursive structure of Part I. Adding an identity coordinate does not add a recursive level. Nor does the number of coordinates establish geometric, physical, or cognitive dimension.

An execution configuration may admit several distinct incarnations over time. A type or a configuration therefore does not uniquely identify a living occurrence. A birth profile must define admissible configurations and allocation of fresh incarnation identifiers; it must also distinguish a birth that cannot occur from the termination of an occurrence that existed.

56. Local Ownership Invariants

Lifecycle ownership. A modeled entity does not order another cell's birth, restart, or death. Shared substrates can still create correlated failures or provide infrastructure controls. Those mechanisms and their owners must be described separately; naming them a substrate does not prove compliance.

State ownership. Each cell owns its execution state and decisions. Dependencies are explicit. A store or memory entity with an explicit owner is distinct from implicitly shared execution state; its relations still need examination under the axioms.

Non-imposition. A received observation is data for a local evaluation, not an instruction whose execution the sender can require. Information can have causal effects. The excluded relation is authority over another modeled entity's outcome, including indirect compulsion through its environment (Appendix E).

Observation scope. Genesis describes its immediate observations and has no command authority. Neither a descriptive label nor a locally computed state proves the factual accuracy of the underlying observation.

Silence. Lack of a response does not establish remote failure or death. A cell may nevertheless decide that its own information is insufficient for a particular action.

57. Birth Facts, Local State, and Durable Records

An implementation profile should separate at least these responsibilities:

Category	Owner and lifetime	Examples
Birth facts	Fixed for one incarnation	Type, incarnation identifier, declared contract
Mutable observations and dispositions	Owned by that incarnation	Received projections, pending confirmations, local degradation causes
Primary runtime facts	Maintained locally	Initialization, current context, existence of an action context
Derived states	Computed from declared inputs	World-relative state, operational state, public readiness flags
Durable records	Explicit owner and retention policy	Structural contracts, retained knowledge, provenance

Immutability for an incarnation does not mean that a record cannot be stored. Conversely, storage does not turn a record into live evidence. A profile must name what survives an observation refresh, a context change, the death of an incarnation, or the replacement of a storage process.

Keeping primary facts separate from derived flags helps expose the inputs of a decision. It does not by itself guarantee atomic updates, concurrency safety, or correct observations.

58. Lifecycle and Reversible State

Birth begins an occurrence identified by a fresh meta; termination ends that occurrence. Announcing presence, observing, acting, waiting, degrading, recovering, and remaining silent are possible activ-

ities or local transitions during its life. They are not a mandatory linear sequence ending in silence and then death.

For a living incarnation i , write a context-relative state as

$$q_i = (\kappa_i, o_i, a_i),$$

where κ_i identifies the current observation context, o_i describes the cell's relation to that context, and a_i describes its locally determined ability to act. The profile must define these domains and an admissible pair set

$$\Omega_i(\kappa_i) \subseteq O_i \times A_i.$$

The two state axes need not be interchangeable or independent. A cell may be active relative to its observed world and still be operationally blocked. Not every pair in the Cartesian product need be allowed.

A change in q_i is not, by definition, a birth or death. The occurrence identifier remains i unless an explicit lifecycle boundary ends that occurrence. A new context identifier need not be a new meta.

Termination does not permit reuse of the same occurrence identifier for a later birth. Retained records may establish filiation between occurrences; they do not make the previous occurrence living again.

59. Descriptive Observation and Freshness

A useful local observation record identifies its subject, source, declared scope, context, and the basis on which it is retained. A profile must specify how ordering, receipt time, source time, missing fields, and freshness are interpreted. Receipt time alone does not prove when the described event occurred.

A local freshness predicate may be written

$$\text{fresh}_i(r, \kappa, t) \in \{\text{true}, \text{false}, \text{unknown}\}.$$

Its clock assumptions and admissible evidence must be explicit. A timeout may make the receiver's own action conditions unsatisfied: "I no longer have sufficiently recent information." It does not establish: "the other cell has died." The choice of a threshold is a profile policy, not a universal FOA constant.

Changing contexts may invalidate earlier observations for current action while leaving local history intact. A retained contract can describe a type even when no incarnation of that type is currently observed. Observed presence, structural membership, successful response, and permission to act locally are separate claims.

Genesis follows the same distinction between its own occurrence and its observation activity. Observing, aggregating a bounded local view, emitting a snapshot, and remaining silent can recur during its life. These activities confer no authority over the observed cells.

60. Inter-Level Communication and Local Evaluation

The inter-level constraint applies to $L_n \leftrightarrow L_{n\pm 1}$. An inter-level emission describes a scoped observation; it may not demand a response, prescribe a state transition in its receiver, or enforce convergence. Snapshots and diffs describe what was observed, including uncertainty and partial visibility.

A receiving cell may evaluate that information under its own policy. A locally chosen transition after reception does not by itself give Genesis authority. Whether a mechanism actually imposes an outcome must be assessed from its behavior and environmental effects, not from the name of the message.

Relation	Meaning within this model
Descriptive snapshot or diff	A scoped input the receiver may evaluate
Fire-and-forget observation	No required reception or reaction
Required cross-level reply or acknowledgement	Departs from the inter-level constraint
Repeated descriptive observation	Admissible if bounded locally; no compulsory reconciliation
Cross-level restart, correction, or convergence order	Departs from non-imposition

A transport fan-out is not, by its name alone, a global view. Its recipients, information scope, and obligations must be examined against locality. Likewise, local waiting or renewed observation is distinct from a sender retrying until it obtains a required outcome in another entity.

Within a level, dialogs, direct addressing, fan-out, and memory relations may be used when their terms are explicit and participation remains optional. They may not acquire authority over another cell or propagate compulsory outcomes across levels (Appendices D and E).

61. Local Projection and Partial Transitions

Let x_i denote the locally owned facts and observations of incarnation i , including its current context, and ρ_i its declared profile. On an admissible local domain D_i , a state projection has the form

$$\Pi_i(x_i, \kappa_i; \rho_i) = (o_i, a_i) \in \Omega_i(\kappa_i).$$

This is a consistency requirement on the profile. Defining two separate projection functions does not prove that their outputs always form an admissible pair. The profile must restrict the input domain, enforce the compatibility condition, or explicitly report a projection failure. Such a failure is not evidence of another entity's death.

Local evolution may be described by a partial transition function

$$\delta_i : D_i \times U_i \rightharpoonup D_i,$$

where U_i contains locally interpreted inputs: observations, application events, freshness evaluations, or a locally selected action. Its domain and guards must be specified. An input is not intrinsically a command; the meaning of the relation determines whether it imposes an outcome.

When a guard is unmet, the profile must say whether the cell waits, remains silent, records an unsatisfied condition, or terminates locally. This edition does not prescribe one universal transition table. Lifecycle-ending inputs must be distinguished from transitions within the same occurrence.

Conditional preservation statement. If the initial local facts belong to D_i , every defined nonterminal transition preserves membership in D_i , and Π_i maps D_i into Ω_i , then every finite sequence of those transitions yields an admissible pair. The proof is induction on the sequence length. These are hypotheses to establish for a profile, not a proof that any implementation already satisfies them. The statement establishes neither eventual action nor factual truth.

The cell owns its evaluation of its own action conditions. This ownership rule does not mean that it authors every value it receives or controls the substrate on which it runs.

62. Waiting, Degradation, Recovery, and Death

The following situations should remain distinguishable:

Situation	Local interpretation	Inference not justified by that fact alone
No adequate current observation	Awaiting usable context	Remote death
A structural provider is absent from the current projection	A local prerequisite is unmet	Provider malfunction or a duty to respond
A local execution condition is impaired	Locally reported degradation	System-wide failure
A missing condition becomes satisfied	Possible local recovery	A new incarnation or guaranteed execution
Presence is withdrawn from an observation view	Change in that view	Termination of the process previously observed
This occurrence terminates	Its lifecycle ends	Destruction of its type, contract, or retained records

One cell's recovery does not clear another cell's recorded causes. Even within a cell, removing one local cause need not remove others. A profile should identify ownership of each degradation cause and the evidence for clearing it.

Observation can remain partial or absent without an architectural error. Consequences for a product or a person awaiting a result still require explicit design and responsibility.

63. Persistence and Continuity

Continuity has several scopes: stable type, situated responsibility, one living incarnation, retained local history, durable knowledge, and a later occurrence linked by provenance. They are not equivalent.

A context restart may leave one incarnation and its mutable dispositions intact while replacing its observation basis. A later incarnation may read retained material under an explicit memory relation, but must identify itself as a new occurrence. Storage restoration does not prove that earlier live observations are current or that missed events have been recovered.

A composed ensemble can persist through changes in its members. Describing that pattern does not create a higher-level owner of their decisions, establish one uninterrupted incarnation, or prove a physical field. Appendix H develops this conceptual distinction without claiming implementation evidence.

64. Observability and Local Conditions

State publication describes the emitter's scoped evaluation. A view of structural readiness is not automatically a view of local action readiness, and local readiness is not a promise of a response. Projections must expose which question they answer.

Observability can inform local evaluation and human decisions. It does not grant Genesis corrective authority. A dashboard or alert must distinguish description, inference, and the separately authorized action of an operator. Local feedback and substrate controls require their own dynamical and operational assessment.

Local validation of an input, state pair, or action guard is compatible with this framework. A validation mechanism that orders another modeled entity's behavior is a different relation. The same distinction applies to local invariants and guarantees.

65. Boundedness and Evidence

The finite-extension proposition in Part I still requires H1–H4. A richer state model adds no proof that those hypotheses hold. Profiles must account for the cost of dimensions, projections, guard evaluation, observation rates, queues, histories, durable records, and shared resources.

For each state or transition, a useful evaluation identifies its owner, admissible inputs, context scope, freshness policy, guard, persistence boundary, and observable consequence. Static source reading, checked mathematical conditions, executed tests, and runtime observations provide different kinds of evidence and should be reported separately.

66. Questions for a Candidate Implementation

- Can a cell remain present while locally unable to act?
- Which transitions preserve the same incarnation, and which end it?
- Can stale information suspend local action without declaring remote death?
- Are the primary facts, derived states, and compatibility checks consistent?
- Which records survive a context change or an incarnation's termination?
- Do any observation or resource mechanisms impose another cell's outcome?
- Are local work and retained history bounded under the stated workload?

67. Contribution and Limits

FOA can describe autonomous dynamics without reducing existence to operational readiness or imposing one linear lifecycle. Identity, context, observation, action, and continuity become distinct responsibilities with explicit local transition conditions.

These distinctions make implementation questions more precise. They establish no universal state vocabulary, complete conformance result, safety guarantee, intelligence claim, or physical scalability result. Appendix H presents a Sphaera-inspired conceptual profile and identifies its open verification obligations.

Part VII — Responsibility, Ethics, and Scope

69. Purpose

This part describes the ethical intention associated with non-imposition. It distinguishes that intention from a demonstrated property of an implementation or a general account of human institutions.

70. Non-Imposition

Within the FOA model, a cell cannot prescribe another cell's outcome. A Genesis describes its immediate observations and holds no command authority. This rule defines the studied class.

71. Explicit Non-Goals

The model does not promise delivery, execution, correctness, fairness, optimization, convergence, or operational safety. A product requiring these properties must specify how they are provided and whether its relations remain within the model's scope.

72. Guarantees and Their Scope

An imposed outcome between modeled cells would conflict with non-imposition. A local mathematical invariant or a human service commitment is not automatically such an outcome. Its owner, scope, and enforcement mechanism need to be named.

Guarantees can protect people and support accountable services. This corpus does not make a general ethical judgment against those purposes.

73. Local Responsibility

The model assigns local behavior to its owner. A cell is not a substitute for the human or organization responsible for design, authorization, operation, and consequences. Those responsibilities must remain identifiable.

74. Safety and Harm

FOA's constraints do not prove the absence of harm, misuse, cascading resource failures, or unsafe decisions. A safety assessment needs a particular implementation, environment, and use case.

75. Authority

The intended observational relations do not grant command authority to Genesis. Whether administrative infrastructure, resource ownership, or indirect influence can introduce authority in practice is a question to examine. An architectural definition alone does not eliminate power in a social setting.

76. Human Governance

Human governance and organizational obligations must be explicit. A model that omits an internal policy-enforcement role does not remove the need for accountability, informed decisions, and appropriate oversight around its use.

77. Intelligence

Autonomous composition does not establish intelligence. Learning, interpretation, and reasoning require their own definitions and evaluation. The model studies conditions of coexistence and local action.

78. Extension and Resources

An abstract family of finite structures may be recursively extended under the assumptions of Part I. Physical resources and the cost of operation still grow and can limit a real system.

79. Silence and Disappearance

Silence and disappearance are admissible observations within the model. Their consequences for a product, a person waiting for a result, or an operator require separate attention.

80. Questions for Use

Who owns an action? What is known about its effects? What remains uncertain? Who can authorize an intervention? How can someone revisit a decision? These questions help connect the model to the people using and operating it.

81. Research Intention

FOA explores the conditions under which autonomous entities can compose an ensemble while keeping relations and responsibilities explicit. Its ethical intention is to make imposed authority examinable. Its technical value and practical limits remain matters for evidence and discussion.

Appendix A - Formal Axioms and Invariants of Fractal Ontological Architectures

A.1 Purpose of This Appendix

This appendix records the core axioms. Conditional results also rely on the hypotheses stated in the relevant parts of the paper.

Its purpose is to:

- prevent hidden premises,
- block interpretative drift,
- make the model's constraints available for examination.

Any proposal that violates **one** axiom is **not FOA**, regardless of intent.

A.2 Core Axioms

Axiom A1 - Ontological Autonomy

Every cell exists independently.

- No shared memory
- No shared execution state
- No implicit dependency

Scope: Cell **Forbidden implications:** compulsory shared lifecycle or implicitly shared execution state. Physical shared dependencies and correlated failures are not excluded by this axiom alone.

Dependent sections: Parts I, IV, VI

Axiom A2 - Strict Locality

An entity may only observe or interact with:

- its immediate level,
- its immediate parent or children.

No transitive visibility is allowed.

Scope: Cell, Genesis **Forbidden implications:** global view, recursive aggregation **Dependent sections:** Parts I, III, VII

Inter-level interactions ($L_n \leftrightarrow L_{n\pm 1}$) are further constrained by §60, which explicitly restricts the communication patterns permitted across levels.

Axiom A3 - Non-Imposition

No entity may impose:

- execution,
- behavior,
- correction,
- convergence,

on any other entity.

Scope: All **Forbidden implications:** compulsory execution or corrective supervision of another modeled entity **Dependent sections:** Parts II, IV, VII

Axiom A4 - Observational Genesis

Genesis entities may only:

- observe existence,
- record presence or absence,
- emit non-authoritative snapshots.

Genesis entities may **never** command.

Scope: Genesis **Forbidden implications:** governance, enforcement **Dependent sections:** Parts I, VI, VII

Genesis entities may emit observational snapshots derived from their local view. Such emissions are informational only and must not introduce authority, obligation, compulsory coordination, or command authority. A receiver may evaluate a descriptive observation under its own local policy (§60).

A diff is not a correction. It is a description of change as observed, not a prescription of change to apply.

Axiom A5 - Silence Validity

Silence is a valid state.

- No response $\neq$ failure
- Absence $\neq$ error

Silence alone is not an architectural error or proof of remote failure. A local freshness or action guard may nevertheless become unsatisfied (§59); this is a condition of the receiver, not authority over the silent entity.

Scope: Communication **Forbidden implications:** mandatory replies, timeouts treated as proof of remote failure or death **Dependent sections:** Parts IV, V, VI

Axiom A6 - No Required Global Convergence

FOA does not seek:

- global truth,
- final state,
- consensus.

Local coherence is sufficient for modeled existence. Spontaneous convergence is permitted, as clarified in Appendix F.

Scope: Architecture **Forbidden implications:** compulsory global consistency or reconciliation imposed across modeled entities **Dependent sections:** Parts I, IV, VII

Axiom A7 - Responsibility Locality

Responsibility cannot be delegated upward.

- No higher level absorbs failure.
- No abstraction dissolves accountability.

Scope: Assignment of modeled responsibility; human and organizational responsibility remains explicit **Forbidden implications:** blame abstraction, safety delegation **Dependent sections:** Parts V, VII

A.3 Derived Invariants

Within the modeled relations, A1–A7 exclude an authoritative global observer and mandatory corrective command chains. Bounded local evaluation additionally requires H1–H4 from Part I.

The axioms do not establish an absence of substrate bottlenecks, correlated failures, harmful actions, or indirect authority in an implementation. Those are evaluation questions.

A.4 Axiom Closure Statement

Any system claiming FOA compatibility must satisfy **all axioms simultaneously**. Partial compliance is invalid.

Appendix B — Alternative Designs and Model Boundaries

B.1 Purpose

The examples below identify relations that would change the FOA model. An alternative may be useful for a different need. Departure from an axiom is a classification of the modeled relation, rather than a judgment about its designer.

B.2 Recovery Commands

If a Genesis orders the restart of a child, it acquires authority over that child's lifecycle. This departs from A3 and A4. A candidate implementation must also describe how its substrate lifecycle is operated.

B.3 Delivery Obligations

A delivery obligation imposed by one cell on another differs from A3 and A5. Local transport mechanisms must be distinguished from a semantic obligation to respond or act.

B.4 Actor-Based Implementations

An actor runtime may provide a useful message-passing mechanism. Compatibility depends on the selected semantics, including supervision and mandatory responses, rather than on the name of the computational model.

B.5 Fairness

Resource allocation has practical and ethical consequences. A global arbiter commanding modeled cells would depart from A2 and A3. Local allocation policies and substrate responsibilities need explicit evaluation; this corpus establishes no universal impossibility of fairness.

B.6 Misuse

FOA does not establish a defense against malicious behavior. An external safety or governance requirement must be addressed directly, with its scope and responsibilities stated.

B.7 Authority at Higher Levels

Adding command authority at a higher modeled level changes A3 and A4. Repeating the same non-imposition rule at every level is part of the definition.

B.8 Emergency Intervention

An emergency command is still a command in the model. An actual deployment may need emergency intervention; that need can determine the suitability and boundaries of the architecture.

B.9 Shared Memory

A shared store that silently becomes a common authority would conflict with local ownership. An explicitly represented Memoria entity can make a memory relation visible, while preserving optional interaction and a declared owner. Its persistence, absence, and resource dependencies still require a concrete design.

B.10 Comparison

Each alternative should be evaluated against its intended capability and the exact relation it introduces. A change in requirements may justify a change in model.

B.11 Closing

A useful discussion makes both the architectural assumption and the practical need visible. It leaves room to revise either in light of evidence.

Appendix C - Boundary Entities (Normative)

Scope of this edition. Normative statements below define the FOA model and its represented relations. They do not assert that an implementation, substrate, or human institution cannot introduce other effects. Conformance must be examined against actual behavior.

This appendix is normative. It defines boundary semantics and forbids interpretative drift.

C.0 Identity Primitives - **geminae** and **meta**

geminae - Logical Identity (Type)

geminae is the **logical identity** of a cell (its ontological type).

It defines:

- what the cell **is** (class of existence),
- which notitia set it belongs to,
- which providers are meaningful as prerequisites.

geminae is:

- stable for the life of the cell,
- not negotiable at runtime,
- not derived from observation.

geminae is ontology, not telemetry.

meta - Runtime Incarnation (Unique Instance)

meta is the **runtime incarnation** of a cell instance.

It identifies:

- this unique living occurrence,
- at this moment,
- in this runtime context.

meta is:

- unique,
- ephemeral,
- not reusable as a logical identity.

meta is a body. geminae is the kind.

C.1 Cell - The Living Unit

A cell is an autonomous living entity identified by:

- geminae: logical identity (type)
- meta: runtime incarnation (unique instance)

A cell:

- is born,
- exists,
- disappears,

without ever being piloted, commanded, or corrected by another entity.

C.2 notitia - Observed Presence

notitia is the **observed presence** of a living cell instance (meta) for a given geminae.

Properties:

- volatile (runtime-only)
- never persisted
- may be withdrawn or expire under a declared observation policy; withdrawal does not prove remote death
- treated as factual observation only

notitia is not:

- a guarantee of availability,
- an obligation to respond,
- a promise of cooperation.

notitia is presence as observed - nothing more.

C.3 praerequisitum - Logical Contract

praerequisitum is a **declared, structural contract**: the list of provider geminae required for a cell's coherence.

Properties:

- defined at cell startup
- persisted (structural memory)
- invariant during the life of the cell
- independent from runtime observation

praerequisitum does not guarantee:

- that providers are alive,
- that providers respond,
- that any interaction succeeds.

praerequisitum expresses what must exist for coherence, not what must happen.

C.4 gemilogica - Local World Projection

gemilogica is a **local, temporary projection** derived from observation:

- a snapshot of notitia perceived for the providers listed in praerequisitum
- computed by a declared observation mechanism (a CellDependencyGraph in the Sphaera profile)
- never persisted
- never authoritative
- never a source of truth

gemilogica is not:

- a registry,
- a reconciliation target,
- a convergence mechanism,
- a dependency resolver with authority.

A cell may evaluate this projection under its own local policy, including its own action guards or local ranking. The projection must not prescribe or enforce another entity's outcome (Part VI and Appendix E). Its source, scope, context, and freshness must remain explicit.

gemilogica is a projection, not a memory.

C.5 Closure Table - What Each Entity Can and Cannot Mean

Entity	Nature	Persisted	What it is	What it must never imply
geminae	ontology	no	logical type	runtime truth, presence
meta	incarnation	no	unique instance	type identity, reuse after termination
notitia	observation	no	observed presence	availability, obligation
praerequisitum	contract	yes	structural prerequisites	cooperation, response
gemilogica	projection	no	local world view	truth, convergence

The persistence column concerns these entities as live semantic evidence. A stored type declaration, historical identifier, or observation log is a record, not a living instance or current projection (§§57–59).

C.6 Non-Negotiable Statements

1. *notitia* may vanish without fault.
 2. A *praerequisitum* contract may be valid and retained while its runtime presence conditions are unmet.
 3. *gemilogica* may be incomplete, stale, or empty without error.
 4. *geminae* never changes to “match reality”. Reality is observed, not imposed.
-

Appendix D - Interpretation Constraints (Normative)

Scope of this edition. Normative statements below define the FOA model and its represented relations. They do not assert that an implementation, substrate, or human institution cannot introduce other effects. Conformance must be examined against actual behavior.

This appendix is normative. It restricts interpretation, not implementation.

D.1 Scope

This appendix applies **only to interpretation of behavior** derived from Appendix C semantic entities.

It does not define new entities.

It does not introduce new semantics.

D.2 What Is Explicitly Allowed (Intra-Level)

Within a single level (L_n), the following are **semantically valid**:

- request/response dialogs
- one-to-one dispatch
- one-to-many broadcast
- fixed or shared subjects
- fan-out or fan-in patterns
- interaction with Memoria cells (shared memory of a *geminae*),
where memory is incarnated as an entity and accessed exclusively via Dialog

These forms are **structural shapes**, not contracts.

They imply:

- no obligation,
- no guarantee,
- no required reaction.

D.3 What Is Explicitly Forbidden (Semantic Drift)

The following interpretations are **invalid**, regardless of implementation:

- “a request must be answered”
- “absence indicates failure”
- “retry restores correctness”
- “broadcast implies coordination”
- “presence implies availability”

Any system relying on these assumptions is **not FOA**.

D.4 Silence Semantics (Non-Negotiable)

Silence is:

- valid,
- expected,
- non-exceptional.

Silence may result from:

- choice,
- overload,
- delay,
- disappearance,
- indifference.

None of these require correction.

D.5 Dependency Misinterpretations (Explicitly Blocked)

- *praerequisitum* does not compel a provider to act; its declared presence conditions may inform the receiver’s local operational state.
- *notitia* is **not** a health signal.
- *gemilogica* is **not** a registry.
- *geminae* names a logical type; it does not establish a live or available service.

Any design treating them as such is invalid.

D.6 Final Appendix Statement

FOA allows interaction.
FOA forbids obligation.
FOA tolerates absence.

If an interpretation introduces:

- a compulsory response,
- correction imposed on another entity,
- required global convergence,
- or enforcement of another entity’s outcome,

then that interpretation describes a different model or needs an explicit scope clarification.

Appendix E — Influence and Participation (Normative)

E.0 Scope

This appendix defines a constraint on relations between modeled entities. It does not claim that information has no causal effects, that an implementation can avoid all shared-resource influence, or that the constraint is a general law of human interaction.

E.1 Distinguish Information from Prescribing an Outcome

An entity may emit a descriptive observation. A receiving cell may use it in its own local evaluation. Such a use does not, by itself, grant the observer authority over the receiver.

The prohibited relation is one in which an entity prescribes, enforces or covertly selects another entity's outcome by controlling its environment, opportunities or interpretation. Conformance must be assessed on the actual mechanism; "descriptive" is not sufficient as a label.

E.2 Participation Constraint

A relation that intentionally shapes another cell's opportunity space must make its terms explicit to the participating cells. Participation must remain optional and cannot introduce command authority or mandatory execution.

An explicit dialog is not an exception to non-imposition. The inter-level descriptive constraint in Part VI still applies; dialog cannot authorize a cross-level command.

E.3 Projections

A projection, metric, heuristic or estimate describes its declared inputs and assumptions. It cannot carry an undisclosed directive to the receiver. A receiver may rank its own local candidates using observations, provided it does not acquire authority to impose outcomes on other cells.

E.4 Environmental Shaping

A modeled entity may not silently manipulate another cell's resource conditions to force a preferred result. Physical resource allocation, shared substrates and administrative decisions require explicit examination; this model does not demonstrate that their effects disappear.

E.5 Prediction

A local prediction may inform a cell's own decision. Prediction does not authorize preemptive control of another entity. Its uncertainty and observation scope should remain visible.

E.6 Reputation and Metrics

A score used locally is distinct from a mechanism that makes other cells' access or existence depend on that score. The latter relationship must be examined against participation and non-imposition. A metric does not establish correctness merely because it is numerical.

E.7 Optimization

Local optimization is compatible with autonomous evaluation. Optimization that prescribes another modeled entity's response or induces compulsory convergence departs from the model.

E.8 Information Effects and Authority

Receiving information can change a local decision. That fact alone does not establish authority. Conversely, absence of a command message does not prove absence of imposed outcomes: environmental mechanisms may also impose them.

E.9 Review Questions

- Who owns the decision whose opportunities are changed?
- Are the relation's terms visible to its participants?
- Can the receiver decline or remain silent?
- Does any mechanism force another cell's outcome?
- Do shared dependencies introduce effects outside the modeled relations?

E.10 Closure

The constraint preserves locally owned decisions and explicit relations. It is a model requirement to check, not a demonstrated immunity to every form of influence.

Appendix F — Emergence Is Not Authority (Normative)

F.0 Scope

This appendix concerns interpretation within FOA. It distinguishes a spontaneous pattern from a relation that imposes another entity's outcome. An implementation must establish which mechanism actually produced the observed pattern.

F.1 Emergence

A pattern is emergent, in this usage, when no modeled entity mandates the collective outcome. It may arise from locally owned preference, learning, adaptation or coincidence. A coordinated-looking result does not by itself prove a command relation.

F.2 Authority

Authority is the capacity to prescribe or enforce another modeled entity's outcome. Local optimization, a cell's own invariants and intentional local stability are not alone evidence of that relation. A distributed mechanism can nevertheless impose an outcome without having a single commanding node.

F.3 No Mandate from Prevalence

Convergence does not imply correctness. A prevalent practice does not create an obligation. A stable pattern does not establish factual truth. These are distinctions between kinds of evidence, not assertions that convergence or stability is undesirable.

F.4 Convergence without Imposition

Cells may align, imitate, diverge or form compatible groups. The model does not require them to converge. If a mechanism makes another cell's convergence compulsory, the relation departs from non-imposition.

F.5 Norms and Conventions

Cells may adopt conventions that make interactions understandable. A convention within this model remains optional to the participating cells and does not grant a right to command another entity.

F.6 Selection

A cell may select its own local action according to preference and declared compatibility. A relation that controls another cell's access, opportunities or execution must be reviewed under Appendix E. Calling that relation "selection" does not settle its status.

F.7 Stability and Evidence

Persistence of a pattern is evidence of persistence. Correctness, usefulness and suitability require their own criteria. A stable pattern can be useful without becoming authoritative.

F.8 Evolution

Long-term structural change may occur through local choices. Its existence does not by itself establish design intent or a global mandate. A deliberately chosen local update must be distinguished from externally imposed change.

F.9 Closure

FOA permits spontaneous patterns and locally owned decisions while excluding command authority between modeled entities. Those rules do not establish that every emergent pattern is useful, safe or conformant in every other respect.

F.10 Distinctions to Preserve

- Convergence and correctness require different evidence.
- Stability alone does not establish authority.
- A convention is distinct from an enforced obligation.
- Local selection is distinct from prescribing another cell's outcome.
- Emergence alone does not establish full model conformance.

F.11 Final Statement

The research intention is to preserve the conditions under which local decisions and relations can produce an ensemble. Conformance and the resulting behavior must both be examined; neither can be inferred from the word “emergence”.

Appendix G — Enigma: Conditions of Containing Coherence

G.0 Research Scope

This appendix proposes conditions for dynamics within a bounded domain. Its use of “life” is an architectural metaphor. It establishes no equivalence with biological life, intelligence, safety, or the full FOA model.

G.1 Dynamical Setting

Consider $\dot{x} = f(x, t)$ on a neighborhood of a domain Ω . Assume unique solutions exist, f is locally Lipschitz in x with suitable continuity in t , and the solution exists over the time interval being considered.

Let V be continuously differentiable and let $\Omega = \{x : V(x) \leq c\}$ be closed with a regular boundary: $\nabla V(x) \neq 0$ on $\partial\Omega$. These assumptions are needed for the boundary argument below; unrestricted discontinuous dynamics are not covered.

G.2 Containment Condition

An inward or tangent vector field satisfies

$$\nabla V(x) \cdot f(x, t) \leq 0 \quad \text{on } \partial\Omega.$$

Under the stated regularity and existence assumptions, the boundary condition is used to establish forward invariance over the solution’s existence interval. Global-in-time claims require global existence as well. This condition does not prescribe a unique internal trajectory.

G.3 Boundary Exchange

A proposed exchange model may constrain a boundary flux J by

$$|n \cdot J(x, t)| \leq \kappa \Phi(x, t), \quad 0 \leq \kappa \leq 1.$$

Here n is the outward normal and Φ is a nonnegative reference flux with compatible units. A nonzero lower permeability requires an additional assumption; it does not follow from this upper bound. A relation between this flux model and the state dynamics must be specified separately.

G.4 Internal Dynamics

Containment can coexist with more than one internal behavior. If sensitive dependence or chaotic dynamics is claimed, the flow, invariant measure, Lyapunov exponents, and relevant entropy notion must be defined and established for the example.

A positive Lyapunov exponent alone is not a definition of freedom or intelligence. These interpretations remain research questions.

G.5 Recovery and Meaning

Let R be a proposed local recovery operator and I a declared invariant. Requiring

$$I(R(x, t)) = I(x)$$

specifies preservation of that invariant when R is applied. It does not prove that recovery is feasible, safe, or available, and does not authorize one entity to apply R to another.

G.6 Relationship with FOA

A containing domain, a declared local invariant, and a descriptive exchange relation can help frame a FOA example. Compatibility requires examining who owns the state, which relations impose effects, and whether observation grants authority. Containment by itself proves no non-imposition property.

The intended direction is to preserve room for local dynamics while making their conditions understandable. A concrete example must demonstrate that those conditions can hold together.

G.7 Closing

Enigma asks how continuity and changing internal dynamics can coexist within stated boundaries. This is a proposal to investigate through defined examples, rather than a universal law establishing that a system is alive.

Appendix H — Sphaera: A Situated State and Dimension Profile

H.1 Scope of the Public Profile

This appendix presents a conceptual profile inspired by the Sphaera research project. It makes distinctions about states, lifecycle, identity dimensions, and continuity available within FOA. It does not document a particular source tree, installation, or running service, and it provides no implementation-conformance evidence.

The proposals below belong to this public research text. A candidate implementation must declare its own state vocabulary, admissible combinations, transition guards, observation policy, and resource assumptions. The profile does not establish that every FOA axiom or the bounded-work hypotheses of Part I hold in an implementation.

H.2 What the Comparison Adds

FOA distinction	Conceptual contribution	What remains to establish
Existence and action	Separate projections of contextual presence and ability to act	Compatibility of all reachable state pairs
Lifecycle and transitions	Blocking, recovery, and context renewal within one incarnation	Explicit birth and termination criteria
Type and occurrence	Type, situated responsibility, execution configuration, and incarnation	Identifier allocation and consistency
Local observation	Context-relative evidence with a declared freshness policy	Accuracy, time assumptions, and completeness
Continuity	Separation of birth facts, local dispositions, and retained records	Retention boundaries and restart semantics

These distinctions help make FOA's dynamics expressible. They do not establish a general law of autonomy, a complete memory theory, or a working intelligence.

H.3 Presence and Ability to Act

A contextual state answers what a cell can establish about its own presence in a locally interpreted context. An operational state answers whether its declared conditions for a particular action are met. Those questions need separate projections, even when they depend on some of the same facts.

For an incarnation i and context κ , let

$$\Omega_i(\kappa) \subseteq O_i \times A_i$$

be the admissible pairs defined by the profile in Part VI. The sets and their compatibility predicate are profile parameters, not a universal enumeration of states.

A possible pair describes a present cell whose action conditions are unmet. Such a cell need not be absent, dead, or newly born. Conversely, a declared readiness label does not establish that the information supporting it is current or accurate. A local state describes the scope of its own evaluation; it does not guarantee delivery, response quality, safety, or another cell's conduct.

Compatibility of two labels is only one obligation. The profile must also show that its projection maps admissible facts into $\Omega_i(\kappa)$ and that reachable transitions preserve those admissible facts. Declaring a matrix does not prove its enforcement.

H.4 Changes Within One Incarnation

Three conceptual situations illustrate why state transitions should not be reduced to a lifecycle sequence:

1. **A locally required observation disappears and returns.** A cell can suspend an action while continuing to exist, then resume when its own conditions are met. A profile may preserve the incarnation identifier throughout.
2. **An observation expires.** The cell can judge its own evidence insufficient for an action. This does not establish the observed entity's death or the observer's failure.
3. **The context changes.** Renewed observations may be required while a previously recorded local cause remains relevant. Context renewal alone neither allocates a new incarnation nor clears every local condition.

These situations are model examples, not reports of executed scenarios. The profile must specify which facts survive a context change, which confirmations need renewal, and when an action is permitted. Reversibility of a state change does not imply reversibility of termination.

H.5 Lifecycle and State Are Different Descriptions

A lifecycle describes whether an occurrence has been admitted, born, or terminated. A state projection describes its conditions at a particular observation point. An occurrence can pass through several local states during one lifetime, and similar state labels can recur without a new birth.

An initialization narrative therefore should not be treated as the complete state machine. Labels used for admission, construction, structural availability, and application readiness may answer different questions. A profile must give their meanings and the guards connecting them.

Likewise, silence, an expired observation, withdrawal of an observed presence, and termination are distinct events. No one of them proves all the others. The same caution applies to an ensemble that continues while its members change: continuity of the pattern is not continuity of every occurrence.

H.6 Type, Situated Identity, and Execution Dimensions

A conceptual type can be described by coordinates such as architectural role, kind of entity, purpose, and technology. Using subscript S for this appendix's illustrative sets,

$$G_S = P_S \times C_S \times R_S \times T_S.$$

These are descriptive coordinates, not four execution states. The product describes a possible vocabulary; a concrete profile may constrain which combinations are meaningful.

A situated responsibility can combine capability, type, and topological origin:

$$L_S \subseteq M_S \times G_S \times O_S.$$

The subset matters: origin, capability, and type need not vary independently. A profile must state its constraints rather than infer them from the number of coordinates.

Execution can be further situated by application, environment, local context, and substrate:

$$I_S \subseteq L_S \times K_S \times E_S \times X_S \times B_S.$$

Several configurations may refer to the same situated responsibility. Changing an environment does not by that fact alone rename the responsibility. Repeated births under an identical configuration still require distinct incarnation identifiers.

These coordinates answer where and how an entity is situated. Recursive depth in Part I answers where it sits in a composition. Neither the coordinate count nor the tuple notation establishes a physical geometry, a dimension theorem, or a cognitive capacity.

H.7 Local State and Continuity

A candidate profile should distinguish:

- **Birth facts:** identity and declarations fixed for one incarnation.
- **Local observations and dispositions:** mutable information owned by that incarnation.
- **Primary facts:** locally maintained inputs from which public states are projected.
- **Retained records:** contracts, histories, or other records whose owner and lifetime are independently declared.

An observation may survive a context change if the profile permits it. Its retention does not establish its continuing freshness. A contract can outlive an observer or provider without establishing either one's current presence. Durable memory likewise requires its own retention and access semantics; it is not an automatic property of an incarnation.

The same distinctions apply to composed presence. An ensemble can reform as its parts change, while a particular occurrence remains bounded by birth and termination. This is a conceptual account of continuity. It does not establish a physical field, an implemented memory mechanism, or command authority of the ensemble over its members.

H.8 Limits of the Contribution

The profile does not prescribe a deployment, admission protocol, storage arrangement, or universal state vocabulary. Those choices require a separately scoped implementation specification and evidence.

Local evaluation is not sufficient by itself to establish bounded local work. Observation sets, projection cost, event rates, queues, retained histories, and shared resources must be accounted for under H1–H4. A signed declaration establishes the identity of the signer and integrity of the signed bytes under the verification assumptions; it establishes neither factual truth nor universal compliance.

The contribution is narrower: distinguish lifecycle from reversible states, contextual presence from local action conditions, identity coordinates from composition depth, and occurrence continuity from retained records. Another implementation can examine these questions without adopting a complete Sphaera design.

H.9 Questions for Evaluation

Evidence for an implementation would need to address at least:

1. Whether all reachable fact combinations project to admissible state pairs.
2. Whether transition guards preserve the declared domain and account for undefined inputs.
3. Whether observation freshness is interpreted consistently under stated time assumptions.

4. Whether birth, termination, context renewal, and retained records have distinct identity and lifetime rules.
5. Whether local work and retained information remain within the resource hypotheses claimed.

These are open verification obligations. This public conceptual profile does not report their completion or substitute for reproducible implementation evidence.

Glossary - Fractal Ontological Architectures (FOA)

This glossary defines terms **as used in FOA**. Common meanings in other systems do not apply unless explicitly stated.

Architecture

A set of ontological rules defining what entities may exist and how they may relate. In FOA, architecture does **not** prescribe behavior or outcomes.

Authority

The capacity to impose outcomes, corrections, or constraints on another entity. FOA explicitly forbids authority at all levels.

Cell

The minimal ontological unit in FOA.

Properties:

- autonomous existence,
- independent lifecycle,
- no shared state,
- no obligation to act.

A cell may observe, act locally, or remain silent.

Coercion

A mechanism by which one modeled entity forces another entity's behavior. Local retries and guarantees require their own scope; their names alone do not establish coercion.

Control

The ability to influence or determine system behavior globally or transitively. FOA eliminates control as an architectural property.

Convergence

The process by which a system moves toward a single global state or truth. FOA excludes required global convergence imposed across modeled entities. Spontaneous convergence is permitted.

Failure

A specified loss of a capability or the termination of an occurrence. Silence, loss of observed presence, stale information, operational blocking, and confirmed termination are distinct facts; none is proof of the others. The model permits absence without an architectural error. Consequences for a product require their own assessment.

Fractal

A structure where the same ontological rules apply at every level of composition. In FOA, fractality applies to **existence**, not behavior.

Context

The declared scope in which observations and local state conditions are interpreted. A context identifier and an incarnation identifier answer different questions; renewing one need not replace the other.

Dimension

An explicitly named coordinate used to distinguish a type, situated identity, or execution configuration. Coordinates can be constrained and are distinct from recursive composition depth. Their count is not a physical dimension claim.

Incarnation

One executing occurrence identified by a fresh meta, from birth to termination. Reversible state transitions and context changes may occur during that lifetime. Retaining records does not revive a terminated occurrence.

Genesis

An observational entity associated with a level.

Genesis may:

- observe existence,
- record presence or absence,
- emit non-authoritative snapshots.

Genesis may **never** impose behavior.

Guarantee

A promise that an action, delivery, correctness, or outcome will occur. Guarantees that impose another modeled entity's outcome conflict with non-imposition. Local mathematical properties and human commitments require their own stated scope.

Locality

The constraint that an entity may only depend on its immediate level. FOA enforces strict locality.

Non-Imposition

The foundational law of FOA: no entity may impose outcomes on another.

Observation

The passive reception or recording of information without obligation or authority. Observation does not require an action. A receiver may use it in locally owned evaluation; doing so does not by itself grant authority to the observer.

Operational State

A cell's locally derived description of whether its declared action conditions are satisfied. It is distinct from observed existence, and readiness is not a promise of response or successful execution.

Ontology

The set of rules governing what exists and what relations are possible. FOA is defined ontologically, not operationally.

Outcome

Any externally imposed result or required state. FOA refuses to define or enforce outcomes.

Responsibility

The obligation borne by an entity for its own actions or failures. In FOA, responsibility is strictly local and non-transferable.

Scale

The number of entities or levels in the system. In FOA, scale has no semantic impact on behavior or authority.

Silence

The absence of response or activity. Silence is a valid and meaningful state in FOA.

Snapshot

A non-authoritative observation emitted by a Genesis. Snapshots describe existence, not correctness.

System

A collection of entities sharing ontological rules. In FOA, a system does not imply coordination or control.

Truth

A statement supported by local observations. The model does not require a global truth service; it does not establish that local observations are factually correct.

Validation

A check against explicit conditions. A cell may validate its own inputs, state compatibility, or action guards. A modeled observer may describe its scoped checks, but cannot use validation to prescribe another cell's outcome. A valid local projection does not prove factual truth, successful execution, or safety.

Violation

A departure from a stated axiom, including authority or coercion between modeled entities. Conformance must be assessed against the actual relations, not a mechanism's name.

Notation and Formal Conventions - Fractal Ontological Architectures (FOA)

This document defines the symbols, conventions, and formal language used throughout the FOA paper and appendices.

No symbol carries implicit operational meaning unless explicitly stated.

General Conventions

- Capitalized terms refer to **defined concepts** (see `glossary.md`).
 - Lowercase terms refer to informal or descriptive usage.
 - Logical statements are descriptive, not prescriptive.
-

Sets and Entities

Cell

- Denoted by: $c \in \mathcal{C}$
- A cell is an autonomous ontological entity.

No assumption is made about:

- behavior,
 - execution,
 - responsiveness.
-

Genesis

- Denoted by: $g \in \mathcal{G}$
- A Genesis is an observational entity associated with a level.

Genesis entities have no authority over cells.

Levels and Structure

- Levels are indexed by $n \in \mathbb{N}$
- The finite constructions studied in Part I use a finite set of entities at each level. Infinite structures require additional definitions and are not inferred from those constructions.

$$L_n = \{c_1, c_2, \dots\} \cup \{g\}$$

No entity in L_n has visibility into L_{n+k} for $k > 1$.

Fractal Composition

Let:

- S_0 be a single cell,
- S_1 be a sphere (set of cells + Genesis),
- S_n be a composite structure of level n .

Fractal composition is defined as:

$$S_{n+1} = \bigcup_{i=1}^k S_n^{(i)} \cup \{g_{n+1}\}$$

Where:

- k is the branching factor,
- g_{n+1} is the Genesis observing S_{n+1} .

No additional rules are introduced at higher levels.

Observation

Observation is denoted by:

$$\mathcal{O}(x)$$

Meaning:

- the existence or state of x is recorded,
- no action is implied,
- no obligation follows.

Observation does not command the observed entity. Recording may change the observer's local state; physical observation costs require separate accounting.

Silence

Silence is denoted implicitly by absence:

$$\neg \mathcal{O}(x)$$

Silence is not equivalent to failure, error, or invalidity.

Authority and Imposition

Authority is denoted abstractly as a relation:

$$x \rightarrow y$$

Where x can impose outcomes on y .

FOA explicitly forbids all such relations.

Formally:

$$\forall x, y \in \mathcal{E}, \quad x \nrightarrow y$$

Where $\mathcal{E}$ is the set of all entities.

Guarantees

Guarantees would imply:

$$\exists x, y : x \Rightarrow y$$

Meaning:

- the existence of x forces an outcome in y .

This implication is forbidden in FOA.

Responsibility

Responsibility is local:

$$\forall c \in \mathcal{C}, \quad \text{Resp}(c) = c$$

No responsibility propagation is allowed.

Failure

Failure is modeled as disappearance:

$$c \notin L_n$$

Here membership denotes an observation scope, not knowledge of global mortality. Disappearance from that scope does not prove termination. Local action conditions may become unsatisfied; the missing entity acquires no obligation (§§59–62).

Truth

Truth is local and observational:

$$\text{Truth}(x) = \mathcal{O}(x)$$

This is notation for a local observation claim, not a proof of factual truth. The model supplies no authoritative global truth service.

Non-Convergence

The model does not require a function:

$$f : \mathcal{S} \rightarrow T$$

that imposes convergence of all system states to a single terminal state T . This does not deny the existence of constant mathematical functions or spontaneous convergence.

Context-Relative State and Transitions

Part VI uses i for an incarnation, κ_i for its context, and $q_i = (\kappa_i, o_i, a_i)$ for context-relative and operational states. The allowed pairs belong to $\Omega_i(\kappa_i) \subseteq O_i \times A_i$. These profile-defined axes are distinct from composition level n .

D_i is the admissible locally owned fact domain, U_i the locally interpreted input domain, Π_i the state projection, and $\delta_i : D_i \times U_i \rightharpoonup D_i$ a partial transition function. A partial function need not be defined for every possible input; undefined cases require an explicit profile policy.

The freshness predicate $\text{fresh}_i(r, \kappa, t)$ has declared values true, false, or unknown. Its time and evidence assumptions belong to the profile.

Appendix H uses subscript S for Sphaera dimension sets; these symbols do not redefine the main paper's cell, Genesis, or level sets.

Formal Scope Statement

Notation states model constraints, local dynamics, and conditional results. It does not constitute an executable protocol or give one modeled entity command authority over another. Local causal dependence on information is permitted. Conformance, reachability, factual truth, and operational behavior require evidence with their own scope.

Bibliography and Influences - Fractal Ontological Architectures (FOA)

This bibliography lists conceptual, technical, and philosophical works that are **adjacent to FOA**, not predecessors or implementations of it.

FOA is not derived from any single work listed here. References are provided for orientation and comparison only.

Distributed Systems (Foundational)

- Lamport, L.
Time, Clocks, and the Ordering of Events in a Distributed System
Communications of the ACM, 1978.
- Brewer, E.
Towards Robust Distributed Systems (CAP Theorem)
PODC Keynote, 2000.
- Lynch, N.
Distributed Algorithms
Morgan Kaufmann, 1996.

These works formalize coordination, consistency, and failure - properties whose requirements and assumptions help situate the choices made in FOA.

Actor Model and Message Passing

- Hewitt, C., Bishop, P., Steiger, R.
A Universal Modular Actor Formalism for Artificial Intelligence
IJCAI, 1973.
- Agha, G.
Actors: A Model of Concurrent Computation in Distributed Systems
MIT Press, 1986.

Actor semantics and supervision vary by model and implementation. Part III examines selected relations rather than classifying every actor system.

Event-Driven and Reactive Systems

- Hohpe, G., Woolf, B.
Enterprise Integration Patterns
Addison-Wesley, 2003.
- Bonér, J., Farley, D., Kuhn, R., Thompson, M.
Reactive Manifesto
2014.

These works offer patterns for responsiveness and resilience. Their control and recovery mechanisms must be examined in a particular implementation.

Fault Tolerance and Consensus

- Fischer, M., Lynch, N., Paterson, M.
Impossibility of Distributed Consensus with One Faulty Process (FLP)
Journal of the ACM, 1985.
- Lamport, L.
Paxos Made Simple
ACM SIGACT News, 2001.

FOA avoids consensus entirely rather than attempting to optimize it.

Systems Theory and Complexity

- Simon, H.
The Architecture of Complexity
Proceedings of the American Philosophical Society, 1962.
- Perrow, C.
Normal Accidents
Princeton University Press, 1984.

These works provide context for studying composition and failure; they do not establish the conditional results of this corpus.

Philosophy and Ontology (Contextual)

- Deleuze, G.
Difference and Repetition
Columbia University Press, 1968.
- Simondon, G.
Individuation in Light of Notions of Form and Information
University of Minnesota Press, 2005.

Referenced for ontological framing only. FOA does not adopt their metaphysical positions.

Ethics, Power, and Control

- Foucault, M.
Discipline and Punish
Pantheon Books, 1975.
- Scott, J. C.
Seeing Like a State
Yale University Press, 1998.

These works analyze the relationship between visibility, control, and systemic domination - themes FOA addresses architecturally.

Scope Clarification

The references provide context for the research model. They do not establish a claim of universal novelty or operational superiority.