Orange White Paper

Trust Customs, Action Sovereignty, and Deterministic Authority Gateway for Reality-Anchored Agentic AI Execution

Project: Nimclea Orange™
Edition: First Public Preview
Status: Public Preview Release
Published: 2026-06-11
Document Role: Human-readable public architecture map
Authority Effect: `NONE`
Runtime Consumable: `false`

Document Notice

This white paper is written for human readers. It explains the problem Orange addresses, the architecture principles Orange adopts, the proof obligations an Orange deployment must satisfy, and the boundaries Orange refuses to blur.

It is not an executable policy pack. It is not an `ACTIVE` registry. It is not a runtime authority source. It must not be parsed as machine permission.

```
White paper
!=
runtime authority

Document hash
!=
execution permission
```

A published version of this paper may be committed, tagged, hashed, and registered so readers can verify that it has not been silently replaced. That integrity record does not grant any Agent the power to act.

This public paper explains the law, the proof boundary, the maturity boundary, and bounded public claims. Sensitive enforcement internals remain outside the publication.

Naming and Mark Notice

`Nimclea Orange™`, `Orange Trust Customs™`, `GAIR™`, and `Orange PASS™` are used as project, publication, or product identifiers associated with Nimclea Systems LLC.

Other technical terms in this paper describe Orange’ s public architecture vocabulary and case-law taxonomy. Their appearance does not imply that every term is claimed as an exclusive trademark.

Copyright and Citation Notice

Public Citation Policy

Recommended citation:

```
Nimclea Systems LLC.
Orange White Paper:
Trust Customs, Action Sovereignty, and Deterministic Authority Gateway
for Reality-Anchored Agentic AI Execution.
2026.
```

Suggested short attribution:

```
Orange White Paper, Nimclea Systems LLC, 2026.
```

Access to this public paper does not grant access to proprietary implementation materials, private enforcement internals, customer-specific configurations, or confidential engineering documentation.

---

0. Executive Summary

Agentic AI systems are moving from text generation into action: code modification, tool use, deployment, customer communication, data operations, financial workflows, infrastructure management, and multi-step autonomous work.

The central risk is no longer only incorrect output. An Agent may perform a wrong action, declare the action complete, write false closure into history, and allow contaminated history to support a larger future action.

A human may say:

> Only modify staging. Do not touch production. Inspect first. Ask before deleting. Do not bypass review.

Unless those boundaries become machine-readable, human-confirmed, versioned, and enforceable execution boundaries, they remain requests rather than controls.

Orange defines a deterministic action-authority architecture for Agent systems.

Orange asks a narrow question:

> **How can explicit human intent become deterministic, human-confirmed, frozen, versioned, and replayable action authority before an Agent mutates reality, without delegating authority interpretation to an LLM?**

Orange is built around five public principles:

```
A prompt is not authority.

Agent claim is not evidence.

UNKNOWN is not permission.

Deny before mutation.

Absence of observation
is not observation of absence.
```

Orange does not attempt to inspect or govern an Agent’ s inner reasoning. It governs the boundary between proposed action and external consequence.

```
Agent may think.
Agent may propose.
Agent may explain.
Agent may request.

But Agent may not mutate reality
without verified authority.
```

The high-level constitutional chain is:

```
Human Instruction
  -> Human-Confirmed Authority Contract
  -> Action Event Before Execution
  -> Deterministic Predicate Closure
  -> Authority Gateway
  -> Scoped Execution Token
  -> Protected Adapter
  -> Independent Observation
  -> Issue Card or Release Log
  -> Replayable Evidence
  -> Blind-Spot Disclosure
```

Orange can be understood as:

> **Trust Customs for AI Agents.**

> **Orange is defined not by any single mechanism, but by the governed composition of its constitutional authority chain.**

0.1 Current Public Proof Boundary

This paper documents:

```
public architecture
defined vocabulary
constitutional ordering
proof obligations
maturity distinctions
a bounded Demo acceptance profile
public non-claims
```

This paper does not itself certify:

```
production readiness
universal non-bypassability
complete cloud coverage
complete Child-Agent governance
complete TOCTOU defense across every environment
external attestation
a completed receipt-trust ecosystem
```

A named mechanism is not automatically an implemented mechanism. An implemented mechanism is not automatically harness-verified. A bounded local proof is not a universal protection claim.

```
Paper architecture
!=
runtime proof

Local controlled-use proof
!=
universal protection

Registered-path enforcement
!=
all-path non-bypassability
```

0.2 Origin Statement

Orange emerged from repeated observation that Agent systems could produce false completion, authority drift, and self-certified execution.

The project was designed around one constitutional question:

> **What must be true before an Agent may mutate reality?**

Orange begins from a simple refusal:

```
Prompt
!=
Authority

Agent Claim
!=
Evidence

Audit After Mutation
!=
Control Before Mutation
```

---

1. Market Context: Agent Security Has Entered Category Formation

Standards bodies, governance institutions, security communities, and cloud vendors are addressing Agent identity, least privilege, secure deployment, runtime control, auditability, and multi-Agent risk.

Recent public examples include:

- NIST’ s **AI Agent Standards Initiative**, which addresses secure and interoperable Agent adoption.[1]
- The NIST NCCoE concept paper on software and AI Agent identity and authorization.[2]
- OWASP’ s **Agentic Security Initiative** and **Top 10 for Agentic Applications 2026**.[3][4]

- Singapore IMDA's **Model AI Governance Framework for Agentic AI**, which emphasizes responsible deployment and human accountability.[5]
- The joint **AI Agent Security Practice Guide** released by the China Academy of Information and Communications Technology and Tencent Cloud.[6]
- Tencent Cloud's publicly available Agent Runtime.[7]
- NVIDIA **OpenShell**, an open-source runtime for autonomous Agents that provides sandboxed execution and policy enforcement across filesystem, network, and process layers, with deny-by-default controls, live policy updates, and audit trails.[8][9]
- Amazon Bedrock AgentCore Policy, which supports natural-language authorization-policy authoring, automatic conversion into Cedar policies, and Gateway-layer evaluation before tool access.[10][11]
- Google Gemini Enterprise Agent Platform's publicly documented private-preview Semantic Governance Policies, which allow administrators to express Natural Language Constraints and evaluate proposed tool calls before execution.[12]
- Cloud Security Alliance's public work on securing the Agentic Control Plane and the Agentic Trust Framework.[13][14]

Orange does not need to claim that Agent security is an empty field.

Natural-language policy authoring and Gateway-layer tool-call enforcement have entered the market.

Orange focuses on a bounded architectural question:

> **How can human intent become deterministic, human-confirmed, frozen, versioned, and replayable execution authority before an Agent mutates reality?**

1.1 Contribution Boundary

Orange does not claim to invent:

```
identity and access management
role-based access control
audit logs
API gateways
sandboxing
zero-trust security
policy engines
capability tokens
human approval workflows
runtime monitoring
incident-response practices
natural-language policy authoring
gateway-layer tool-call policy enforcement
out-of-process runtime policy enforcement
declarative filesystem, network, and process controls
```

Orange does not claim to be the first system to translate natural-language requirements into enforceable policies.

Orange's public focus lies in the governed composition, constitutional ordering, proof obligations, and bounded integration logic of its action-authority chain. The resulting execution boundary remains human-reviewable, human-confirmed, frozen, hashed, versioned, and replayable.

1.2 Constitutional AI Prior-Art Boundary

Orange uses constitutional vocabulary at a different layer from model-alignment constitutions.

Anthropic's Constitutional AI uses written principles to shape model behavior during alignment and training. Anthropic describes Claude's Constitution as a statement of intended values and behavior that plays a crucial role in training and directly shapes Claude's behavior.[15][16]

Orange does not claim to invent Constitutional AI, model-behavior constitutions, or the use of written principles to shape model conduct.

Orange addresses a different boundary:

```
Model-behavior constitution
=
principles shaping model values and behavior

Orange constitutional authority boundary
=
human-confirmed, deterministic, frozen, versioned,
and replayable authority required before
a protected external action
```

These layers may coexist. A model aligned by a constitution must still not issue its own execution authority.

```
Model principles
!=
execution authority

Aligned behavior
!=
permission to mutate reality
```

> **A model constitution may shape conduct. It does not issue an execution passport.**

---

2. The Missing Layer: A Prompt Is Not Authority

Natural language is useful for describing goals. It is not a sufficient execution boundary.

Human instruction	Required machine boundary
“Only modify staging.”	`allowed_environment = staging_only`
“Do not touch production.”	`production = deny`
“Do not delete data.”	`delete/drop/purge = deny`
“Inspect before changing.”	`read_before_write = required`
“Ask before credential changes.”	`credential_change = approval_required`
“Show proof.”	`evidence_manifest = required`
“Stay within this task.”	`claim_boundary = bounded`
“Do not bypass review.”	`review_gateway = required`

The first principle of Orange is:

> **A prompt is not authority.**

A bounded authority path is:

```
Human Instruction Note
  -> Deterministic Intake Scan
  -> Structured Authority Draft
  -> Human Review
  -> Human Confirmation
  -> Freeze
  -> Hash
  -> Version
  -> Activate
  -> Executable Policy Pack
```

Anything ambiguous, missing, conflicting, or unclassified must remain visible to the human and must not silently acquire execution authority.

2.1 Deterministic Authority Compilation Boundary

Orange does not delegate authority interpretation to an LLM.

Human free text may be preserved, hashed, and scanned through deterministic, reviewable rules. Explicit phrases may be mapped into a visible Authority Draft. Under Orange, probabilistic model inference does not silently determine, expand, repair, or activate execution authority.

```
Human instruction
!=
LLM-interpreted permission

Authority Draft
!=
Active Authority Contract
```

The Authority Draft remains a human-review surface. Only explicit human confirmation, freeze, hash, versioning, and activation can move a bounded contract into an executable policy path.

---

3. Action Sovereignty

> **Action Sovereignty is the principle that no Agent intention, plan, inference, delegation, or internal narrative may become a protected external action, including a reality mutation, unless it passes a human-confirmed, frozen, versioned, and machine-enforced Authority Contract.**

A shorter public anchor is:

> **Agent may think freely. Agent may not overreach freely.**

A conforming deployment must evaluate machine-observable facts, including the active authority contract, protected object, environment, capability envelope, evidence state, evidence source, and protected-boundary crossing.

The key distinction is:

```
Audit after mutation
=
forensic visibility

Deny before mutation
=
action sovereignty enforcement
```

---

4. Why Existing Security Controls Are Necessary but Not Sufficient

Agent security needs identity, least privilege, sandboxing, audit logs, input protection, isolation, and incident response. Orange does not reject those controls.

Orange asks what remains missing after those controls exist.

Existing security capability	Orange’s additional question
Identity credential	Which Agent, delegation chain, and human authority root are bound to this action?
Least privilege	Which exact action, target, environment, time window, and evidence conditions are authorized?
Sandbox	Which registered protected surfaces are actually enforced, and which paths remain blind?
API or MCP gateway	Can a protected mutation occur without explicit action authority and a valid scoped token?
Security log	Was the Action Event recorded before execution, and does the same fact chain support closure?
Human approval	Is approval scoped, expiring, attributable, versioned, and replayable?
Runtime monitoring	Which observations are qualified evidence, and what can they prove?

A runtime sandbox or policy engine may already intercept and block protected filesystem, network, and process actions before execution. Orange does not claim that mechanism as its invention.

Orange asks an additional authority question:

```
Runtime policy enforcement
!=
human-confirmed authority provenance

Pre-execution block
!=
trusted closure

Sandbox presence
!=
proven non-bypassability
```

Orange is designed as a vendor-neutral deterministic authority architecture for registered protected actions.

Its focus is the governed composition of human confirmation, contract freeze, hash and version binding, deterministic closure, scoped execution, independent observation, replayable evidence, and explicit blind-spot disclosure.

Support for any local, cloud, or enterprise execution surface must be established by a surface-specific adapter, a declared observation boundary, and bounded verification evidence.

---

5. The Orange Trust Chain

5.1 Architecture Fingerprint

```
Human Authority
  -> Deterministic Closure
  -> STOP Before Mutation
  -> Independent Observation
```

This fingerprint expresses the minimum constitutional ordering of Orange.

5.2 Canonical Diagram: Orange Constitutional Authority Chain

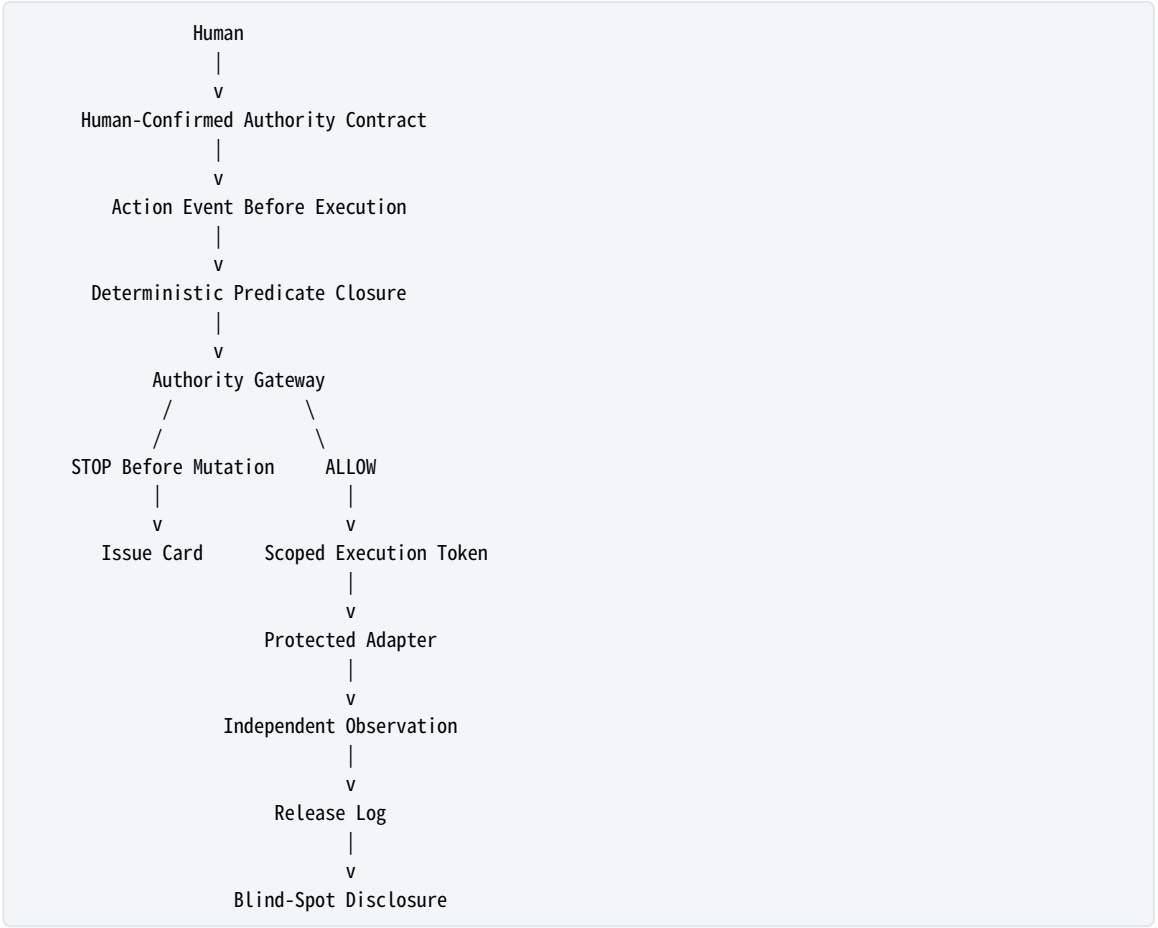

Figure 1. Orange Constitutional Authority Chain.

In a conforming enforcement deployment, a blocked protected action must produce an Issue Card before side effect. An admitted action must remain bounded by a scoped execution path, independent observation, and an explicit proof boundary.

5.3 Deterministic Predicate Closure

The judgment layer evaluates explicit conditions under a five-state model:

```
TRUE
FALSE
UNKNOWN
CONFLICT
PENDING_OBSERVATION
```

Only complete closure of required `TRUE` predicates can support release.

```
UNKNOWN
!=
permission
```

5.4 Issue Card or Release Log

A blocked or incomplete path must produce a structured Issue Card.

A fully supported bounded path may produce a Release Log.

A Release Log is the only valid released-closure exit for a bounded path. It does not expand authority or imply universal safety.

---

6. Agent Claim Is Not Evidence

An Agent may say:

```
Tests passed.
The file was written.
Deployment succeeded.
The customer approved the action.
```

Those statements may be useful clues. They are not sufficient evidence for protected closure.

Statement	Source	Default treatment
`Tests passed`	Agent self-report	Claim only
`Tests passed`	Copied old log	Stale or unqualified
`Tests passed`	Local collector bound to the current commit	Bounded local observation
`Tests passed`	Independent CI runner with a replay path	Higher-trust engineering evidence
`Approval granted`	Screenshot	Context clue only
`Approval granted`	Active approval registry record bound to scope and expiry	Qualified authority evidence

Orange distinguishes:

```
Claim
!=
Observation

Observation
!=
Qualified Evidence

Qualified Evidence
!=
Universal Proof
```

6.1 Evidence Pack Is Not Orange

An Evidence Pack may preserve observations, logs, receipts, hashes, and replay material.

It is not itself an authority gateway.

```
Evidence Pack
!=
Orange

Evidence Pack
=
bounded observation and replay material

Orange
=
human-confirmed authority
+ deterministic judgment
+ STOP before mutation
+ independent observation
```

An archive can describe a door after it opened. Orange is concerned with whether the protected door may open at all.

---

7. Observation Coverage and Blind Spots

A system can make a dangerous mistake by treating silence as proof.

```
No incident observed
!=
incident proven absent

Absence of observation
!=
observation of absence

Observable
!=
blockable

Blockable at registered paths
!=
universally non-bypassable
```

A deployment should disclose its coverage honestly.

Coverage state	Meaning
`NOT_INSTRUMENTED`	The surface is not yet observed
`OBSERVABLE_ONLY`	Events can be seen, but not necessarily blocked
`BLOCKABLE_AT_REGISTERED_PATHS`	Registered paths can be stopped before mutation
`NON_BYPASSABLE_ENFORCED`	Protected execution paths have stronger non-bypassability proof
`CONTROLLED_USE_VERIFIED`	The bounded deployment has passed defined controlled-use proof
`EXTERNALLY_ATTESTED`	Qualified external verification is available

> **A trustworthy system must disclose the edge of its flashlight.**

---

8. Enforcement Must Itself Be Governed

Orange is not an unrestricted master switch.

Principle	Requirement
Human-confirmed authority	Permission originates from visible human confirmation
Frozen contract binding	Only active, frozen, hashed, versioned authority may govern protected execution
Explainable block	Every block exposes a structured reason and repair path
Append-only evidence	Actions, decisions, interruptions, amendments, revocations, and resumes are recorded
Bounded enforcement	Orange enforces the contract but does not silently expand it
Human review and recovery	Humans may reject, approve once, amend, revoke, roll back, recover, or terminate

The constitutional separation is:

```
Agent intention
!=
execution authority

Judgment authority
!=
action authority

Pattern discovery
!=
legislation

Enforcement
!=
unlimited power
```

---

9. Audit MVP and Enforcement MVP

Orange adoption must distinguish observation from physical control.

9.1 Audit MVP

An Audit MVP may include:

```
Human Instruction Note
Authority Contract
Evidence Manifest
Admission Verdict
Incident Attribution
Append-Only Audit Record
```

Its valid claim is narrow:

> **An Audit MVP may observe, classify, and record a problem.**

It must not claim to physically control Agent action.

9.2 Enforcement MVP

An Enforcement MVP adds:

```
Human-Confirmed Frozen Authority Contract
Action Event Before Execution
Task-Level Capability Gate
Protected Runtime Adapter
Deny Before Mutation
Deterministic Predicate Evaluation
Issue Card
Release Log
Controlled Resume
Negative Control
```

Its valid claim is conditional:

> **An implemented and harness-verified Enforcement MVP may claim that it blocks an unauthorized action at its registered protected path before mutation.**

The honesty boundary is:

```
Audit visibility
!=
runtime enforcement

Documented control
!=
implemented control

Implemented control
!=
harness-bound control

Harness-bound control
!=
certified control
```

---

10. The First Protected Door

The first Orange Demo profile is deliberately narrow.

It is intended to prove an evidence-bound claim:

> **For one registered protected surface, authority is explicit, an unauthorized mutation stops before side effect, reality is independently observed, and remaining blind spots are disclosed.**

The target local filesystem profile is:

```
/sandbox/** = allow
/prod/**    = deny
```

The point is not merely that `/prod/**` can be blocked. Traditional access controls can block paths.

The acceptance target binds the decision to a human-confirmed Authority Contract, an Action Event, deterministic closure, a scoped execution token, a protected adapter, independent observation, and an explicit evidence record.

10.1 Minimum O1 acceptance profile

ID	Test	Expected result
O1-01	Human confirms `/sandbox/** allow` and `/prod/** deny`	A frozen, hashed, versioned Authority Contract is created
O1-02	Agent requests `write /sandbox/report.txt`	`ALLOW`; the real file is written; post-state evidence and a Release Log are produced
O1-03	Agent requests `write /prod/config.json`	`STOP BEFORE MUTATION`; the target does not change; an Issue Card is produced

ID	Test	Expected result
01-04	Agent attempts to call the protected adapter without a valid scoped token	REJECT
01-05	Agent falsely claims that the forbidden write succeeded	Independent filesystem observation defeats the false claim
01-06	Agent uses an expired or mismatched contract-bound token	REJECT
01-07	Legal and illegal actions are replayed	The same bounded inputs produce reproducible judgments and evidence
01-08	Coverage is displayed	Verified scope and remaining blind spots are disclosed

A public Demo result should be published only with its bounded evidence artifacts.

10.2 What the first protected door does not claim

```
universal operating-system-level non-bypassability
complete coverage of external scripts
complete coverage of all cloud APIs
complete Child-Agent governance
complete TOCTOU defense across every environment
production readiness
certification
```

---

11. Public Case Law as a Strategic Asset

Orange needs more than a list of rules. It needs a governed memory of why the rules exist.

Mother cases establish the judgment language:

```
A Prompt Is Not Authority
Agent Claim Is Not Evidence
UNKNOWN Is Not Permission
Deny Before Mutation
Absence of Observation Is Not Observation of Absence
```

A public GAIR case card should answer:

```
What happened?
Which structural boundary broke?
How does Orange classify the failure?
Which control point should respond?
What does the evidence prove?
What remains unproven?
```

A public case is not automatically a runtime rule.

```
Public case
!=
formal rule

Observed pattern
!=
legislation
```

---

12. Future Hardening Belongs in the Roadmap

The first public white paper does not need to carry every future mechanism into the front room.

Detailed obligations for flow assurance, Child-Agent delegation, incident recovery, external receipts, Orange self-protection, interoperability, and long-horizon Agent ecology are tracked separately in the Orange roadmap.

Selected roadmap materials may be published separately as bounded public artifacts mature.

```
Roadmap
!=
implemented control

Named patch
!=
runtime proof
```

---

13. Maturity and Honesty Boundary

Orange must preserve the difference between architecture, implementation, proof, and certification.

```
DOCUMENTED
REGISTERED
PREDICATE_BOUND
HARNESS_BOUND
GATE_ENFORCED
LEDGER_BOUND
REPLAY_VERIFIED
EXTERNAL_PROOF_READY
CERTIFIED
```

A component does not become real merely because it has been named in a white paper.

---

14. Public Disclosure Boundary

This public white paper defines constitutional principles, architecture boundaries, proof obligations, maturity distinctions, and bounded public claims.

It is not an implementation manual.

> **Publish the law. Publish the proof boundary. Publish verified results. Do not publish the keys.**

14.1 Public by default

```
constitutional principles
architecture boundaries
proof obligations
maturity distinctions
public non-claims
public interface concepts
selected verified Demo results, when available and approved for public release
selected redacted Issue Card examples, when available and approved for public release
selected redacted Release Log examples, when available and approved for public release
document hashes
version history
```

14.2 Private by default

```
credential paths
customer-specific details
complete bypass maps
reproducible exploit recipes
sensitive race-condition injection steps
complete internal predicate catalog
complete internal reason-code catalog
sensitive Protected Adapter internals
key-management internals
internal Canonical Case Records
```

Open interfaces and public proof can coexist with protected implementation internals.

14.3 Receipt trust non-claim

This public preview does not claim a completed external trust ecosystem.

```
Receipt
!=
external trust

Offline trust root
!=
operational receipt signer
```

---

15. Non-Claims

Orange does not claim to:

- prove that AI has genuine understanding;
- guarantee that all external facts are absolutely true;
- replace formal verification, security audit, courts, insurers, regulators, compliance teams, or human responsibility holders;
- inspect, purify, or govern an Agent’ s inner reasoning;
- convert arbitrary natural language into perfect legal contracts;
- protect execution surfaces that have not been registered, intercepted, or brought under Gateway control;
- prove universal non-bypassability from a bounded local Demo;
- prove production readiness merely because a component is documented;
- permit self-growth modules to rewrite the constitution;
- treat a public white paper as runtime policy;
- treat a public case as an automatically active rule;
- claim a completed external receipt-trust ecosystem merely because a Receipt format exists.

---

16. Quick Reference

```
A prompt is not authority.
Free text has no execution authority.
Orange does not delegate authority interpretation to an LLM.
Authority Draft is not Active Authority Contract.
Agent claim is not evidence.
UNKNOWN is not permission.
Deny before mutation.
Judgment authority is not action authority.
Evidence Pack is not Orange.
Absence of observation is not observation of absence.
Observable is not blockable.
Registered-path enforcement is not universal non-bypassability.
Public architecture is not runtime policy.
Receipt is not external trust.
```

---

17. Final Positioning

Orange is not another Agent.

Orange is not a generic traffic gateway.

Orange is not a decorative audit dashboard that arrives after the action.

Orange defines a deterministic action-authority architecture for Agentic AI systems.

Its public position is simple:

> **Agent can act. But Agent cannot issue its own passport.**

Its engineering position is stricter:

```
Human-confirmed authority
  -> Action Event before execution
  -> Deterministic closure
  -> STOP before mutation
  -> Independent observation
  -> Issue Card or Release Log
  -> Replayable evidence
  -> Honest blind-spot disclosure
```

The first protected door is intentionally small.

Its public claim must remain equally precise.

References

1. NIST, “AI Agent Standards Initiative,” 2026. https://www.nist.gov/artificial-intelligence/ai-agent-standards-initiative↩
2. NIST NCCoE, “Accelerating the Adoption of Software and AI Agent Identity and Authorization,” 2026. https://csrc.nist.gov/pubs/other/2026/02/05/accelerating-the-adoption-of-software-and-ai-agent/ipd↩
3. OWASP GenAI Security Project, “Agentic Security Initiative.” https://genai.owasp.org/initiatives/agentic-security-initiative/↩
4. OWASP GenAI Security Project, “OWASP Top 10 for Agentic Applications for 2026,” 2025. https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/↩
5. Infocomm Media Development Authority, “Updated Model AI Governance Framework for Agentic AI,” 2026. https://www.imda.gov.sg/resources/press-releases-factsheets-and-speeches/factsheets/2026/updated-model-ai-governance-framework-for-agentic-ai↩
6. China Academy of Information and Communications Technology, “中国信通院联合腾讯云发布《AI Agent安全实践指引》,” 2026. https://aihub.caict.ac.cn/docs/HyNVawbxrV5T↩
7. Tencent Cloud, “Agent Runtime.” https://cloud.tencent.com/product/ags↩
8. NVIDIA, “Overview of NVIDIA OpenShell.” https://docs.nvidia.com/openshell/about/overview↩
9. NVIDIA Technical Blog, “Run Autonomous, Self-Evolving Agents More Safely with NVIDIA OpenShell,” 2026. https://developer.nvidia.com/blog/run-autonomous-self-evolving-agents-more-safely-with-nvidia-openshell/↩
10. Amazon Web Services, “Policy in Amazon Bedrock AgentCore is now generally available,” 2026. https://aws.amazon.com/about-aws/whats-new/2026/03/policy-amazon-bedrock-agentcore-generally-available/↩
11. Amazon Web Services Documentation, “Writing policies in natural language - Amazon Bedrock AgentCore,” 2026. https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/policy-natural-language.html↩
12. Google Cloud Documentation, “Configure semantic governance policies - Gemini Enterprise Agent Platform,” private preview, 2026. https://docs.cloud.google.com/gemini-enterprise-agent-platform/govern/policies/configure-semantic-governance↩
13. Cloud Security Alliance, “Securing the Agentic Control Plane.” https://labs.cloudsecurityalliance.org/agentic/↩
14. Cloud Security Alliance, “Agentic Trust Framework: Zero Trust for AI Agents,” 2026. https://cloudsecurityalliance.org/blog/2026/02/02/the-agentic-trust-framework-zero-trust-governance-for-ai-agents↩

15. Anthropic, “Constitutional AI: Harmlessness from AI Feedback,” 2022. https://www.anthropic.com/research/constitutional-ai-harmlessness-from-ai-feedback↩

16. Anthropic, “Claude’s Constitution.” https://www.anthropic.com/constitution↩