V2 CONTRACTS · PINNED SOURCE INTERPRETATION

A source reference should lead to observable evidence

The V2 convention connects product intent to an owned artifact and its checks.

01 → 02 → 03 → 04

Intent

Feature and architecture pages state the behavior and owning boundary.

Ownership

The numbered backlog task owns the target artifact.

Contract

A machine-readable contract records the expected shape or invariant.

Verification

A checker and workflow evaluate the artifact; current results establish evidence.

> A traceable reference is the start of verification, not proof that the full player journey passed.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

One numbering scheme prevents ambiguous task references

Bare section numbers belong to the V2 backlog; other documents use explicit prefixes.

NOTATION	DESTINATION
§N or §N.M	V2_TODOS.md; the numbered backlog
deps§N	V2_DEPENDENCIES.md
features§"anchor"	V2_features.md; a named heading
arch§"anchor"	V2_ARCHITECTURE.md; a named heading

> Resolve the document first, then the task or named heading inside it.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

Documentation checks make claims easier to trace

Numbering, links, decisions and package names each have a distinct validation rule.

CHECK	WHAT THE PINNED VALIDATOR INSPECTS
Backlog structure	Exactly 138 distinct top-level section numbers
Files and anchors	Local Markdown targets and generated heading slugs
Decision language	A fixed set of unresolved-decision markers
Package names	Declared names, planned dependencies or explicit historical context

> Passing documentation validation establishes traceability within the rules it actually checks.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

Follow the exact subsection even after a green check

The pinned bare-reference validator accepts a known parent when a child reference is absent.

01 → 02 → 03 → 04

Read

A page names a specific backlog subsection.

Validate

The checker accepts its known top-level parent.

Inspect

Open the exact requested heading and verify its owner.

Resolve

Correct the reference or record the missing target.

> A checker’s implementation defines its evidence boundary; the intended convention can be stricter.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

A named path can describe work that still needs to exist

Target-artifact discipline keeps planned architecture separate from the current tree.

01 → 02 → 03 → 04

Name

A source identifies the intended module, package, path or feature.

Inspect

Check whether the artifact exists and what kind of artifact it is.

Own

The first backlog task referencing a target owns its creation and validation.

Qualify

Record implementation, integration and acceptance evidence separately.

> Do not infer an installed module or completed feature from its appearance in a glossary.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

Reconcile the glossary with the project and disk

The source records thirty glossary names and twenty-seven actual Source modules.

RECORD DIFFERENCE	NAMED ARTIFACTS
Absent targets	V2Peripherals, V2AntiCheat, V2DynamicMusic, V2FrameDataPublisher
Present as plugins	V2AICommentary and V2AdaptiveAI
Source modules missing from the glossary	V2, V2Modding and V2World
V2Editor has two meanings	A Source module; a separate inspector-resource path without a plugin descriptor

> Classify the differences explicitly; a larger vocabulary does not imply a larger installed module set.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

Repeatable simulation starts with a deliberate build policy

The target rules select floating-point settings before gameplay code is compiled.

BUILD SETTING	PINNED TARGET CONFIGURATION
Win64 compiler arguments	/fp:strict /fp:except-
Other target platforms	-fno-fast-math -ffp-contract=off
Project definitions	DETERMINISM=1; V2_DETERMINISM=1; V2_STRICT_FP=1
Build environment	bOverrideBuildEnvironment = true

> Compiler configuration supports determinism; the actual repeated simulation still needs verification.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

Compare the game, server and editor targets directly

They repeat the determinism helpers while retaining different build responsibilities.

TARGET	EXPLICIT DIFFERENCES AT THE SOURCE PIN
Game	Unity disabled; developer tools disabled; V2 composition root
Server	Unity and editor-only data disabled; dedicated-server definitions and plugin exclusions
Editor	Adds V2 and V2Tests; no explicit unity override in this target file

> The overview’s all-target unity claim is broader than the pinned target files support.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

A defined symbol is not evidence of an active code branch

V2_DETERMINISM and V2_STRICT_FP are reserved seams in the pinned Source tree.

01

Declared

All three targets add both symbols to ProjectDefinitions.

02

Consumed

No pinned Source .cpp or .h branches on those two symbols.

03

Demonstrated

Compiler, cook and simulation evidence must be inspected separately.

> Trace a definition to its consumer before treating it as runtime behavior.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

Read the determinism contract as a concrete review brief

Section 7.1 connects an owner, a versioned schema and the target settings to inspect.

CONTRACT FIELD	WHAT THE READER FOLLOWS
schema / task	v2.netcode.floatingPointDeterminism.v1 / V2_TODOS.7.1.floating-point-determinism
targets	V2.Target.cs; V2Editor.Target.cs; V2Server.Target.cs
compilerDefinitions	The three determinism definitions
targetRules / strictFloatingPoint	ProjectDefinitions, build-environment override and platform-specific arguments

> Identify the expected artifact and owner before evaluating whether the tree satisfies the contract.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

The same contract reaches cook policy and automation

The delivery fields describe required settings and the files that should enforce them.

CONTRACT FIELD	EXPECTED DELIVERY BOUNDARY
cookSettings	Deterministic cook order, shader-code order and a reference DDC
architecture	x86-64 CI host; Win64 BuildCookRun uses -architecture=x64
automation checker / workflow	Floating-point checker; .github/workflows/v2-build.yml
automation graph / wrappers	Build/Build.xml; cook-all.sh and cook-all.ps1

> A contract’s named automation is a path to evidence, and each path has its own implementation limits.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

Trace the reference cache through the selected cook route

Compare command arguments and inherited environment; these are not one uniform cook path.

CONFIGURATION BOUNDARY	REFERENCE-CACHE EVIDENCE IN THE PINNED SOURCE
Windows CI setup	Binds UE_SHARED_DDC/V2/DDC; publishes cache environment values
BuildGraph: explicit Cook	Arguments include -ddc=$(ReferenceDDC)
BuildGraph: BuildCookRun	Arguments include -cook, but no explicit DDC argument
Direct cook wrappers	No explicit DDC argument; effective cache depends on invocation context

> Confirm the actual cache and architecture used by the executed command, not only the intended contract.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

Read what the checker asserts before interpreting its pass

This gate compares contract values and source snippets; it does not run the simulation.

01

Contract values

Schema, task, definitions, target rules, compiler flags, cook policy and architecture

02

Cross-file snippets

Targets, configuration, graph, wrappers, workflow, companion checkers and backlog state

03

Separate evidence

Effective compiler/cook configuration, runtime repeatability and release acceptance

> A green source check supports its exact assertions and leaves stronger runtime claims to stronger evidence.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

Proto definitions become deterministic Unreal service descriptors

The pre-build command keeps generated metadata aligned with the discovered proto signatures.

01 → 02 → 03 → 04

Discover

Read and sort .proto files beneath libs/proto/src.

Describe

Collect package, service, method, type and streaming metadata.

Generate

Write V2GeneratedGrpcStubs.h and .cpp in V2Services/Generated.

Check

Compare expected text with both existing files using --check.

> Generated descriptor freshness and a working RPC transport are different contracts.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

A descriptor tells Unreal what a service is called

The generated structures expose signatures and lookup data within a deliberately limited parser.

DESCRIPTOR	FIELDS AND BEHAVIOR
Method	MethodName; RequestType; ResponseType; client/server streaming flags
Service	Qualified name, package, service name, proto path, methods and count
Lookup	Case-sensitive qualified-name search; nullptr when absent
Parser scope	Regex-based semicolon RPC signatures; no general protobuf compiler

> Review the generated interpretation as well as its freshness when proto syntax or service shape changes.

V2 CONTRACTS · PINNED SOURCE INTERPRETATION

Finish with a reproducible claim and its evidence boundary

Repository counts orient the review; dated execution results establish what actually passed.

01

Inventory

826 contract files, 1,155 named checkers and 27 Source modules at the pin

02

Trace

Source reference → owner → artifact → checker → chosen execution route

03

Conclude

Name the revision, target, environment, result and remaining limitations

> Carry the scope of the evidence into the claim, then follow the next mechanism in depth.