

# CARE - Founder Protection & Attribution Notice

## Zenodo Public Record - Version 1.0

A formal public notice intended to document authorship, conceptual origin, architectural boundaries, and non-subsumption principles of the CARE framework.

<b>Author</b>	Nick Vejle
<b>Role</b>	Founder & Originator of CARE
<b>Document type</b>	Founder protection / attribution notice
<b>Intended use</b>	Zenodo / public distribution / authorship record
<b>Version</b>	1.0

**CARE is not a feature, a module, or an implementation detail. CARE is a governance architecture.**

This publication is intended to preserve attribution clarity if CARE is discussed, mapped, integrated, cited, or compared alongside other governance or infrastructure models.

## 1. Origin and Authorship

CARE (Commit-Admissibility & Responsibility Engine) is an original governance and system design framework created by **Nick Vejle**.

CARE is not derived from any single prior framework, product, or institution. It emerges from a distinct synthesis of lived systemic experience, execution-level governance reasoning, and boundary-centered system design.

## 2. Core Architectural Claim

CARE introduces a specific architectural principle:

**CARE treats governance as an execution-boundary architecture: upstream systems may reason and propose, but the commit boundary decides whether a state transition is admissible.**

This contribution includes commit-boundary governance, runtime admissibility, legitimacy preservation at execution, and refusal when legitimacy does not survive to commit.

## 3. Scope and Independence

CARE is substrate-independent and implementation-agnostic.

CARE defines **what must be decided**, not **how it is physically enforced**.

External systems such as deterministic infrastructures, kernels, protocols, or hardware layers may integrate with CARE or enforce CARE outcomes, but they do not replace, redefine, or subsume CARE.

## 4. Non-Subsumption Clause

CARE may not be absorbed into another framework as a hidden subcomponent, rebranded without explicit attribution, or redefined as a mere implementation detail of another system.

Any use of CARE concepts should preserve both attribution and the conceptual integrity of the execution-boundary model.

## 5. Attribution Requirement

Public, academic, commercial, or architectural use of CARE concepts should include clear attribution in substantially the following form:

**CARE Framework - Nick Vejle (Founder & Originator)**

## 6. Distinction from Deterministic or Runtime-Only Models

CARE is distinct from architectures that primarily define what is structurally reachable, what is cryptographically enforceable, or what can be deterministically prevented in runtime.

Those approaches may define **what is structurally possible**. CARE defines **what is admissible at the moment of execution**.

## 7. Temporal Claim

This document serves as a timestamped declaration of authorship, conceptual origin, and architectural intent.

It is intended for use as a public authorship record in publication, standards discussion, collaboration, and governance-related discourse.

**Final statement: legitimacy must survive the commit boundary. Governance must exist where decisions become executable. Architecture determines outcomes.**