WHITE PAPER / VERSION 1.0 / NH APPLIED DOCTOR

NH Applied Doctor

Software Compatibility Diagnosis, Repair and Qualification

Find the real fault. Avoid the wrong fix. Prove the result.
Supported by the NH Applied Doctor Development Lab

TECHNICAL ENQUIRIES | https://nhapplied.co.uk • nigel@nhapplied.co.uk
Permanent identifier | https://doi.org/10.5281/zenodo.24310226

Abstract

The costliest compatibility problem is often not the failed program, but the time spent pursuing the wrong explanation. NH Applied Doctor helps engineering teams identify credible failure mechanisms, eliminate unsupported assumptions and verify that a targeted remedy actually works. Its initial demonstrated work concerns GPU computing, including CUDA-to-AMD application ports, HIP/ROCm runtime failures and selected dependency problems. The NH Applied Doctor Development Lab develops and qualifies new approaches to these investigations.

This paper is for engineering managers, developers and prospective pilot customers. It presents **verified local results and explicitly bounded conclusions**, not a claim of universal repair. Implementation details, diagnostic decision rules, conversion methods and unpublished Lab processes are intentionally withheld. Wider cross-platform applications are an engineering direction, not capabilities already established.

VALIDATION PLATFORM	SPECIFICATION
Test computer	BOSGAME BeyondMax mini PC / Strix Halo
Processor and GPU	AMD Ryzen AI MAX+ 395 / Radeon 8060S (gfx1151, RDNA 3.5)
Memory and operating system	128 GB unified memory / Windows 11
Qualification environments	Windows ROCm HIP SDK 7.2; separate rocCV Linux work: Ubuntu 24.04 on WSL2 with ROCm 7.2.1
AT A GLANCE	**STATUS**
Customer outcome	Find the actual fault mechanism before committing to a speculative repair
Locally solved engineering cases	Verified CUDA application ports and targeted GPU runtime repair
Integrated source regression	347 tests executed; 344 passed, 3 skipped
Availability	Scoped engineering assessments; private product candidate, not unrestricted release

Nigel Hutchinson / NH Applied • 17 November 2026 • Version 1.0 • DOI: 10.5281/zenodo.24310226

01 / CUSTOMER VALUE

The expensive mistake is fixing the wrong problem

Why pay NH Applied instead of assigning another engineer? Because a plausible diagnosis can consume substantial engineering time without addressing the cause. A crash might arise from source code, runtime loading, kernel-image compatibility or asynchronous error reporting. NH Applied Doctor offers a focused, evidence-led investigation to **challenge the assumption, isolate the mechanism and test the remedy**. We complement an organisation's engineers rather than replace their knowledge of the product.

Where an evidence-led investigation changes the decision

CUSTOMER PROBLEM	WHAT NH APPLIED CONTRIBUTES
Teams are pursuing an uncertain cause	Test competing explanations and identify the evidence that would justify or reject another repair attempt.
CUDA application needs to run on AMD	Assess the conversion route, isolate blockers and validate bounded HIP/OpenCL outcomes where feasible.
Software crashes after a stack change	Trace environment, runtime and dependency identity; perform a controlled before/after test of a local correction.
A patch appears to work but risk remains	Compare observable outputs with a reference, challenge false positives and document precisely what passed.

What the customer actually receives

A suitable engagement begins with **one named application, one failing environment and an agreed acceptance test**. Deliverables may include an evidence-backed fault assessment, rejected and surviving explanations, a scoped repair candidate where feasible, before/after results, and a qualification report showing **what is fixed, what remains uncertain and what to do next**. If the evidence does not justify a repair, an honest diagnosis is still a useful result.

Doctor is currently best supplied through a **scoped engineering assessment or pilot**, not an unrestricted automated software licence. An engagement is agreed only after technical suitability, confidentiality, source permissions, relevant third-party licences and a measurable test scope have been reviewed. No saving in engineer-hours or time to resolution is guaranteed without customer-side measurement.

Designed to work alongside your engineers

Your engineers retain the application knowledge and deployment decisions. NH Applied contributes a dedicated compatibility investigation, experience from prior GPU failures and a separate check on whether the proposed explanation survives testing. The initial focus is GPU developers, AI application maintainers and teams migrating CUDA-oriented workloads to AMD; expansion beyond GPU computing still requires qualification.

Pilot enquiry: provide a reproducible symptom, target hardware/software stack, access and licensing constraints, and the result you need demonstrated. NH Applied can then establish whether a bounded technical engagement is suitable.

02 / ENGINEERING CONTEXT

Compatibility is more than translation

Moving software between hardware, runtime and operating-system environments is rarely reducible to changing a programming-language API. Successful compilation may conceal a missing device code object, a mismatched library, an incorrect memory assumption or a runtime path that differs from the one exercised in a development test.

These distinctions matter particularly in GPU computing. A visible error can be reported **after** the operation that caused it; a performance regression can masquerade as a missing kernel; and a correct result from one GPU architecture does not establish portability to another.

The engineering gap

- **Diagnosis before modification.** Establish the failing environment, component identities, observed error and the scope of evidence.
- **Compatibility route selection.** Distinguish source conversion from configuration, runtime provenance, missing functionality and application-specific failure modes.
- **Controlled repair.** Limit changes to an understood problem class; retain a clear distinction between a candidate and an accepted remedy.
- **Meaningful verification.** Separate static analysis, compilation, controlled GPU execution, independent acceptance and reporter-environment confirmation.

Initial focus: AMD GPU software

The first specialisation is CUDA-to-AMD compatibility and adjacent GPU runtime engineering. The primary qualification machine and its hardware are identified on page 1. Windows application and product tests used the AMD HIP SDK 7.2 environment; the separate rocCV Linux experiments used Ubuntu 24.04 under WSL2 with ROCm 7.2.1. Results are specific to those tested environments and do not establish compatibility across other hardware or ROCm versions.

AMD ROCm GPU specifications: https://rocm.docs.amd.com/en/latest/reference/gpu-specs.html

The broader identity, **software compatibility engineering**, reflects the kinds of problems already encountered: asynchronous error attribution, compiled-kernel compatibility, host/runtime dependencies, memory behaviour and source-level repair quality. This is the product direction—not a representation that all these problems can already be solved automatically.

03 / PRODUCT SCOPE

What NH Applied Doctor does

Doctor provides focused engineering functions rather than an unrestricted “convert any application” promise. In its present private development form, it can inspect selected source and runtime configurations, identify certain known compatibility risks, propose bounded technical routes, perform narrowly qualified transformations and produce verification evidence.

CAPABILITY AREA	CURRENTLY DEMONSTRATED SCOPE
Compatibility inspection	Selected source-level HIP-launch checks, runtime-component provenance and application-intake diagnostics.
Conversion	Bounded CUDA-oriented examples and approved HIP or OpenCL paths; unsupported patterns are explicitly excluded.
GPU verification	Selected real GPU kernel tests and numerical comparisons against independent reference results.
Packaging	Windows CLI/GUI private portable candidate, extracted and independently smoke-tested on Strix.
Evidence handling	Versioned observations, separate qualification receipts, failure preservation and explicit unverified status.

Non-goals and constraints

Doctor is **not** currently a universal CUDA-to-ROCm compiler, a guarantee that any unsupported GPU will work, or an automatic fixer for arbitrary production applications. Where its input falls outside a qualified scope, the appropriate result may be “not supported”, “review required” or “inconclusive”.

A source inspection is not an executed GPU proof. A local successful launch does not prove a reporter’s failure was resolved. A working private build is not equivalent to a commercially distributable release.

Why the name matters

“NH Applied Doctor” identifies the engineering product under the NH Applied name. “Doctor” is used in the sense of fault diagnosis and repair of **software systems**, not healthcare or clinical decision-making. Product branding is intentionally distinct from its initial CUDA-to-AMD project history.

04 / RESEARCH & QUALIFICATION

The NH Applied Doctor Development Lab

The value of the Lab is not the number of candidate patches, but the quality of the evidence used to accept or reject each explanation. It examines competing diagnoses, challenges plausible but weak assumptions and tests prospective improvements before they become Doctor capabilities. The aim is to avoid committing customers or maintainers to a repair path merely because it sounds technically convincing. A Lab observation does not automatically become a product feature or a confirmed field fix.

A high-level evidence lifecycle

STAGE	WHAT CAN LEGITIMATELY BE CONCLUDED
Observed external report	A specific symptom and environment have been reported; causation is not yet established.
Analysis and hypothesis	A technically plausible explanation or route exists; it still requires a discriminating test.
Controlled local test	A defined change performs as expected under stated local conditions.
Independent acceptance	A separate acceptance check supports the bounded claim and records its limitations.
External reproduction	The affected environment confirms behaviour; this enables a materially stronger conclusion.

Not every investigation reaches every stage. **Ruling out an incorrect cause is a valid engineering outcome**: it can redirect the next test before further development effort is spent. An inconclusive result is labelled as such, rather than presented as a successful patch. The proprietary selection and qualification mechanisms remain undisclosed here.

Why negative controls matter

Several investigated failures involve error-reporting reliability. A successful output alone may be misleading if a rejected kernel launch is never surfaced to the caller. Controlled negative cases can show whether a system correctly detects failure; independent verification then checks that the evidence corresponds to the intended execution. The product preserves failed and superseded findings rather than replacing them with a favourable summary.

“Independent” here denotes a separately performed acceptance/evaluation step within the project; it must not be read as certification by an external laboratory.

05 / PROVEN ON REAL HARDWARE

Engineering problems we have already solved locally

The first evidence for NH Applied Doctor is not limited to open GitHub requests. **Real workloads have been converted, built or repaired and run on the AMD Radeon 8060S (gfx1151).** These are bounded local successes, not universal compatibility guarantees or third-party confirmations.

PROJECT / PROBLEM	LOCALLY VERIFIED RESULT
CUDA voxelizer (real public application)	An isolated HIP build ran on the Radeon 8060S. 6/6 mesh/grid configurations produced byte-identical CPU and GPU outputs across 14 recorded executions.
CUDA GPU image filtering	Gaussian blur, Sobel edge detection and sharpening converted for HIP; all 9/9 observed GPU cases met numerical CPU-reference tolerances.
Windows HIP runtime crash (GGML/SOLVE_TRI)	A controlled 16-run DLL-origin A/B experiment identified a reproducible local crash: 8/8 without matching app-local HIP DLL crashed; 8/8 with it passed 24/24 GPU tests.
HIPIFY #2811 early warning gap	Doctor was changed to detect known unsupported CUDA headers before recommending conversion. The V16 integrated product passed 209/209 source regressions.
rocCV #200 error propagation	Controlled invalid-launch A/B demonstrated an immediate HIP error check; broad 43-site candidate compiled and Flip positive output passed 64/64 pixels.

What the application work demonstrates

CUDA voxelizer: Doctor identified a suitable source-conversion route and helped structure qualification; completing the port still required targeted manual engineering. The result is a working application on the tested Windows AMD environment, not a claim that Doctor automatically ports arbitrary CUDA software.

GPU image filter: The original CUDA source did not require manual source repair in the qualified route. The HIP executable passed nine small image-processing configurations, with maximum observed absolute error within the predefined tolerance. That is a second independent application category, not just another repeat of the voxelizer test.

GGML runtime repair: The local negative/positive comparison directly tied the Windows failure to HIP runtime origin. The original upstream issue's reported `hipErrorInvalidImage` scenario was **not** reproduced, so the local crash is classified as fixed while the upstream issue remains open.

What remains to be proved

These findings establish **local, evidence-backed technical capability**. They do not yet establish success on a customer's machine, unrestricted conversion coverage, completion of every upstream reporter case or commercial redistribution of the private Doctor package.

CUDA voxelizer upstream project: https://github.com/Forceflow/cuda_voxelizer

Second public CUDA image-processing project: https://github.com/keya-05/GPU-Image-Filter-CUDA-Accelerated-Gaussian-Blur-Sobel-Edge-Detection-and-Sharpening

06 / PRODUCT QUALIFICATION

What has been verified

Doctor V28 — integrated source qualification

On 17 November 2026, a copied V28 source integration was qualified on the local Strix Halo host: **347 tests executed, 344 passed and 3 skipped**. The tests covered a mixture of source inspection, route selection, negative cases, selected GPU operations and application-specific fixtures. The source acceptance was recorded separately from the subsequent portable-package check.

The three skips and fixture-specific qualification scope prevent treating this figure as a universal GPU compatibility success rate. The number is a software regression result, not a count of successful external migrations.

Private portable package — separate acceptance

The original V28 private Windows portable archive was independently extracted and checked using a shorter destination path. The extracted CLI version, command help, GUI smoke/navigation behaviour and pinned real-world HIP-source diagnostic were accepted. **1,059 archive entries were checked** (1,058 files and a manifest). The previously accepted archive was not rebuilt or modified for this check.

The portable-package acceptance did not run a complete end-user GPU application, demonstrate portability beyond the named qualification computer, or approve public commercial redistribution. These remain separate gates.

Demonstrated GPU work

The preceding page documents two complete application workloads and one bounded Windows runtime crash repaired on Strix, alongside diagnostic fixes. The 347-test source suite is **additional product regression evidence**, not a substitute for those real executable outcomes. Other targeted kernel and attention tests are separately scoped by their acceptance records.

EVIDENCE	OUTCOME / LIMIT
V28 integrated source	347 tests, 344 pass, 3 skip; separate acceptance on Strix.
V28 private portable	CLI/GUI extraction checks pass; not public-release or cross-machine clearance.
Locally solved field-style cases	Two public application ports and one identified Windows runtime crash, all under bounded Strix acceptance.

Source-of-record: V28 independent source acceptance (17 Nov 2026) and independent private-package acceptance (17 Nov 2026); evidence is retained by NH Applied for qualified review.

07 / EXTERNAL CASE STUDIES

GPU failures: local proof vs. field proof

rocCV #200 — kernel-launch error reporting

An upstream rocCV report identified GPU launch failures that could return control without useful error propagation. In a controlled local test on Radeon 8060S/gfx1151, the original Flip path returned despite an invalid launch configuration, while the candidate raised the HIP launch error immediately. A broader patch candidate added checks at **43 inventoried launch sites**, compiled across **21 operator source files**, and the Flip positive regression produced **64/64 expected pixels**.

Limit: the other 42 sites were compilation-tested, not individually runtime-qualified. The reporter's gfx1036 / invalid-image environment has not been reproduced, and the upstream issue remains open. The candidate patch and qualification notes are publicly inspectable.

rocCV #200 and public candidate evidence: https://github.com/ROCm/rocCV/issues/200

Candidate patch and qualification notes: https://github.com/nigelhutch/nh-applied-doctor-lab/tree/main/patches/rocCV-200

ComfyUI #17082 — attention-path investigation

A reported gfx1151 failure on a newer Linux ROCm/PyTorch combination motivated an opt-in, case-specific attention fallback investigation. The Lab recorded **13/13 isolated routing tests**, **9/9 real GPU mathematical-attention checks**, and **9/9 isolated SDPA-versus-math comparisons** on its own different Windows ROCm stack.

Limit: the reporter's original model and ROCm 10.1 environment have not been run locally. Passing numerical tests on another stack is not verification of the reporter's crash or of a production fix.

ComfyUI #17082: https://github.com/Comfy-Org/ComfyUI/issues/17082

ComfyUI toolbox #19 — architecture identity

The reporter identifies a Radeon 8060S, but an application log reports **gfx1100**, conflicting with the hardware's documented **gfx1151** target. The present toolbox source requests a gfx1151 PyTorch package; the older failing container has not been identified at an exact image digest. The Lab requested same-container runtime and device-architecture evidence instead of proposing an unverified driver change.

Toolbox issue #19: https://github.com/kyuz0/amd-strix-halo-comfyui-toolboxes/issues/19

08 / ADDITIONAL INVESTIGATIONS

Performance and memory behaviour

SCALE #89 — allocation performance

A public report describes comparatively slow repeated 64 KiB allocations on gfx1151 with an older SCALE release. SCALE's 1.7.5 notes describe improvements for a different allocation regime, but the published change does not establish an exact fix for the reporter's case. The Lab requested a matched version A/B test, retaining allocation sizes, repetition count, hardware and native HIP comparator.

Limit: NH Applied has not independently measured SCALE 1.7.5 in that environment. This is an evidence-driven comparison request, not a claim about the quality or behaviour of proprietary third-party software.

SCALE #89: https://github.com/spectral-compute/scale-validation/issues/89

SCALE release notes: https://docs.scale-lang.com/stable/manual/changelogs/whats-new/

comfy-kitchen #188 — identifying the actual bottleneck

An apparent quantisation-backend disparity was initially suspected of causing lengthy model loads. A maintainer's reported test showed that the alleged backend path was **not dispatched during the relevant load sequence**. The Lab therefore redirected the investigation toward memory staging and DynamicVRAM behaviour, using related native fault reports as leads rather than treating the original explanation as proven.

A narrow source-behaviour test of a loader-flag candidate passed **16/16 isolated stubbed cases**, but that test did not import ComfyUI or PyTorch or run a GPU workload. The proposed change could affect memory residency and ownership, so it has **not** been promoted as a runtime fix.

comfy-kitchen #188: https://github.com/Comfy-Org/comfy-kitchen/issues/188

What these cases demonstrate

The commercial value is reducing wrong turns, not accumulating patches. In comfy-kitchen #188, the initial backend explanation was undermined by a maintainer's measured load-path evidence; the sensible next action became investigation of memory and runtime behaviour, not a dequantization patch. In other cases the correct response has been a controlled runtime change, a real source port or an explicit decision that the evidence is insufficient. These are different technical outcomes from the same diagnosis-first discipline; no quantified customer time saving has yet been measured.

09 / MATURITY & COMMERCIAL PATH

From verified local work to customer value

Deployment readiness

MILESTONE	POSITION AS OF 17 NOVEMBER 2026
Internal technical qualification	Established for bounded source and portable-package cases on Strix.
Customer pilot	Not yet evidenced; no independent paying-customer deployment claimed.
Upstream issue closure	Five public cases under investigation; none claimed as independently resolved.
Redistribution/licensing	Third-party notices and Windows runtime component rights require completion before public distribution.

Commercial pilot programme

The immediate commercial offer is a **bounded diagnosis-and-qualification engagement**. The customer supplies the failing application, environment and success criterion; NH Applied helps determine whether the symptom is a conversion problem, environment mismatch, execution fault or another class of incompatibility. A subsequent repair phase is justified only by the evidence and accepted scope. Deliverables are a usable diagnosis, a verifiable repair where feasible, a clear evidence record and explicit unresolved risks—not a predetermined patch count.

Development priorities

- Secure reporter-matched reproduction and at least two independent confirmations of useful results.
- Recruit two technically suitable pilot users and document the initial problem, outcome, customer-side reproduction and engineering effort. Avoid invented time-saved or ROI estimates.
- Complete independent package portability, dependency provenance and redistributable licensing review.
- Extend to new compatibility domains only after representative problems are verified, rather than asserting general capability from architecture alone.

For potential customers

NH Applied invites enquiries from engineering teams facing an expensive or unresolved software-compatibility problem. For a scoped assessment, describe the failing application, observable symptom, hardware/software environment, relevant source or binary permissions, and the outcome you need independently demonstrated. Website: https://nhapplied.co.uk | Contact: nigel@nhapplied.co.uk.

Publication scope: This white paper reports bounded engineering evidence and the NH Applied service proposition. It does not disclose proprietary Doctor source code, conversion algorithms, diagnostic decision rules or unpublished Lab methods, and does not license the software or guarantee success on untested systems.

© 2026 Nigel Hutchinson / NH Applied. Document licence: CC BY-NC-ND 4.0 (https://creativecommons.org/licenses/by-nc-nd/4.0/); no proprietary software or trade secrets licensed. Public evidence: https://github.com/nigelhutch/nh-applied-doctor-lab. Brand names belong to their respective owners; no third-party endorsement is implied.