Cross-Vendor State Observability: An Evidence-Bounded Case Study of Mobile and Automotive Systems

Independent systems research | Silicon Gremlin™

October 20, 2026

Abstract

State is not observed directly across mobile and automotive systems. It is reconstructed from application behavior, transformed media, lifecycle diagnostics, interface events, and vehicle records that expose different representations at different times. This study develops an evidence-traceable account of that problem using two independently assembled investigations and concrete image and restore experiments. The Apple record includes cross-conversation Messages photo-picker behavior, originally investigated as a possible privacy concern and reported still reproducible on October 17, 2026; path-dependent handling of extreme-geometry JPEGs; and restore-path observations in which two runs using a modified cached OS member failed at operation `0x6B4` with status `1109`, while a later run using the original cache completed with status `0`. An earlier successful restore followed a host-network change, but network conditions were not established as equivalent across the cache-comparison runs. Source excerpts document baseline 4:4:4 JPEG test generation and a separate DWT/DCT/SVD watermark-and-extraction workflow; saved Fourier-processing metadata extends the representation-fidelity inquiry. The vehicle investigation documents recurrent display behavior, an FCA app scene-creation watchdog termination, radio firmware information, and stored diagnostic codes. We formalize representation fidelity as an invariant evaluated across declared transformations and distinguish observed outcomes from causal attribution. The findings establish executable test methods, continued researcher-reported reproduction of the cross-conversation picker behavior, and differing restore outcomes associated with the documented cache conditions. They do not establish that the cache perturbation caused the restore failures or a shared Apple–Jeep failure chain. The contribution is an operational framework for separating tested state transitions, diagnostic observations, and open cross-system causal questions.

Keywords: system observability; state continuity; evidence provenance; diagnostic ambiguity; mobile systems; automotive infotainment; causal attribution

1 Introduction

A system can preserve one representation of state while another changes. That difference matters whenever a user-facing view, encoded artifact, process lifecycle, or diagnostic record is treated as a complete account of what occurred. The research problem is not whether these records exist; it is which transitions they expose, which invariants they preserve, and which causal conclusions they can support.

This paper combines executable image-generation and watermark methods, an Apple restore-path and cache-perturbation comparison, device-side incident records, and a separately assembled Jeep/Uconnect chronology. The experimental and observational records address a common question: *when state crosses*

a representation or lifecycle boundary, what remains observable and what can be demonstrated about the transition?

Three results anchor the analysis. First, the Messages cross-conversation photo-picker behavior remains reproducible as of October 17, 2026; the paper retains the original reproduction sequence and distinguishes it from the unresolved internal mechanism. Second, source excerpts establish how extreme-geometry baseline 4:4:4 JPEGs and a separate watermark-recovery probe were constructed. Third, the restore records show two failures in runs identified as using a modified cached member and a successful run after restoring the original cache. An earlier successful restore followed a host-network change, while the available records do not establish equivalent network configurations across the later comparison. The outcomes therefore document a condition-associated difference but do not isolate the cache perturbation as its cause. [W-1–W-6]

The automotive investigation supplies a different form of evidence: a documented display complaint, diagnostic descriptions, an FCA app-lifecycle termination transcribed from an iOS report, and partially aligned interface observations. These records warrant a cross-layer state model. They do not establish that an Apple-originated payload caused a Jeep condition. We examine the two investigations together for their observability structure, not by presuming a common defect. [P-1]

The contribution is therefore methodological and empirical: concrete test construction, a restore-path and cache-perturbation comparison with bounded causal interpretation, a current reproducibility statement, and a representation-fidelity model that keeps observed behavior, experimental manipulation, and untested propagation paths distinct.

2 Background and System Scope

2.1 Mobile-device observations

The Apple investigation concerns cross-conversation media selection, image handling across distinct application paths, and device lifecycle behavior. On October 17, 2026, we confirmed continued reproduction of the Messages cross-thread image-selection behavior under the previously reported conditions. The initial report spans multiple iOS test contexts; their individual observation dates are not fully resolved. State retention, cache reuse, and asynchronous execution remain candidate mechanisms rather than implementation findings.

2.2 Vehicle and interface observations

The Jeep investigation covers service interventions, repeated Uconnect/CarPlay display behavior, driver-assistance and electrical complaints, stored diagnostic codes, and a photographed radio firmware version. The observations span presentation and vehicle-diagnostic layers. “USB/CarPlay interface” identifies the connection under investigation; a specific data path from the phone to a vehicle module has not been established.

Domain	What the corpus describes	Architectural information not established
Apple devices	User-visible Messages/photo-picker and image-path observations; named device and software context in source records.	Internal state ownership, data lifetime, process boundaries, and the mechanism underlying any reported behavior.
Jeep/Stellantis	Vehicle alerts, service actions, a display complaint, separately reported driver-assistance/electrical behavior, and stored diagnostic-code descriptions.	Relevant module topology, message origins, code trigger conditions, event-time telemetry, and software/calibration behavior.
Cross-vendor interface	Researcher-reported presence of an Apple device in the USB/CarPlay context.	Payload contents, transfer of abnormal data, receiving component, subsequent processing, and any effect on vehicle behavior.

Table 1: Scope of the architecture statements: the middle column is descriptive; the right column marks gaps, not presumed implementation details.

3 Related Work

3.1 Distributed State and Event Ordering

The relationship between independently maintained system states has long been studied in distributed computing. Lamport's formulation of the happens-before relation establishes a partial ordering of events based on process-local ordering and communication relationships [R-1]. This provides a foundational distinction between temporal proximity and an established ordering relationship.

Chandy and Lamport subsequently developed a distributed snapshot algorithm for recording consistent global states during distributed execution [R-2]. Their approach relies on coordinated observations of process states and communication channels.

The present study addresses a different observational setting. The Apple and automotive records were assembled independently, without coordinated instrumentation, shared logical clocks, or a verified communication trace between the investigated components.

Consequently, the study does not reconstruct a consistent global state or derive a happens-before relationship between vendor-specific events. Instead, it characterizes the observable representations available within each record and identifies the additional evidence required before relationships between those representations can be established.

3.2 Observability and Representation Consistency

Traditional system observability concerns the ability to infer internal behavior from available measurements. In distributed software, diagnostic events, application states, and externally visible outputs may represent different aspects of an execution.

The present study applies this distinction to records that do not provide direct access to the underlying vendor implementations.

Its representation-fidelity analysis examines whether an artifact preserves specified properties across documented transformations and processing operations. Its state-transition model separately considers whether observable representations from different system layers satisfy an explicitly declared

correspondence.

These are distinct analytical questions. A representation may fail a fidelity test without establishing a cross-component state transition. Conversely, a discrepancy between observable layer states does not independently identify the transformation or mechanism responsible.

The contribution is therefore not a new general theory of distributed consistency or internal observability. It is an evidence-bounded application of representation and state-comparison concepts to independently retained experimental and diagnostic records.

3.3 Causal Attribution and Evidence Boundaries

Causal inference distinguishes observed associations from relationships established through an appropriate causal model or intervention. Pearl's treatment of causality formalizes this distinction and the role of interventions in evaluating causal claims [R-3].

This distinction is particularly important when observations originate from independently managed systems whose internal execution paths are unavailable.

In the present study, controlled image transformations and restore-cache interventions are distinguished from passively collected application, interface, and automotive diagnostic observations.

The restore comparison documents outcomes associated with different cached artifacts but does not establish that all other relevant conditions were equivalent. Similarly, the independently collected Apple and Jeep records do not establish a verified source-to-receiver transfer or a shared underlying defect.

The proposed analytical framework therefore separates observable discrepancies, experimental interventions, and causal hypotheses. It identifies what the retained evidence supports without treating temporal proximity, similar symptoms, or representation differences as sufficient proof of a common cause.

3.4 Position of the Present Study

The present work draws on established distinctions between event ordering, distributed state, observable representations, and causal attribution.

Its specific contribution is the application of these distinctions to a heterogeneous evidence corpus containing controlled image-processing experiments, restore-path interventions, application diagnostics, and automotive observations.

Unlike coordinated distributed-state reconstruction, the methodology explicitly accommodates incomplete instrumentation, uncertain temporal alignment, and unverified cross-component relationships.

The resulting framework provides a structured way to identify observed behavior, evaluate representation fidelity, preserve experimental provenance, and state the additional evidence necessary for stronger causal conclusions.

4 Research Methodology

4.1 Evidence selection and experimental controls

The dated evidence covers Apple device reports beginning August 8, 2025; Jeep service, display, and diagnostic records; and an experimental series extending through October 17, 2026. Each observation retains its original event date, collection context, artifact type, and test conditions. Subsequent analysis does not replace the primary date of an event.

Three evidence forms are used. First, source code and retained file metadata specify controlled inputs and transformations. Second, repeated firmware restoration trials compare an altered cached system member against the original vendor package. Third, diagnostic fields, service records, photographs, and media indexes document observed system behavior. The original FCA IPS fields were transcribed in an earlier technical examination and are reproduced here with attribution to that inspection, not as an independent extraction. Two CarPlay fault timestamps are preserved as provisional pending the original reports. See the evidence references and artifact register.

Missing event-time scan data, full vehicle logs, raw media, and synchronized phone/head-unit traces constrain specific causal inferences. They do not change the documented restore outcomes or the currently reported Messages reproduction.

4.2 Claim classification

Observations, interventions, diagnostics, vendor statements, and proposed mechanisms carry different evidentiary weight.

Observation
: A reported user-visible event or action documented in experimental, service, or vendor records and attributed to its original source.

Interpretation
: An explanation offered by a participant, including a state/caching/race characterization; it is not a mechanism finding.

Hypothesis
: A proposed relationship or propagation path that remains testable and unestablished.

Diagnostic artifact
: A code, printout, timestamp, or media-preservation record; its existence does not resolve origin, meaning, or cause.

Vendor statement
: A test result, classification, disposition, or service statement attributed to the vendor; it is evidence of the response, not automatically of the underlying mechanism.

Independent verification
: Independent reproduction or technical validation. No such cross-vendor reproduction is documented in the available corpus.

Unresolved question
: A proposition for which the current record is insufficient and the evidence needed to address it is stated.

Findings are distinguished from interpretations and hypotheses according to the evidence supporting each proposition. Documentary verification establishes source traceability; experimental reproduction establishes behavior under stated test conditions. Neither substitutes for the other. Claims lacking

sufficient support are not presented as findings.

4.3 Chronology and source attribution

Event times, submission dates, collection times, service dates, and publication dates are tracked separately. A recorded device fault is not automatically a vehicle fault; a stored diagnostic code does not establish the event that set it; and a vendor's classification is not a reproduction experiment. The analysis advances causal claims only where the intervention or trace discriminates between outcomes.

5 Case Study A: Apple Device Reports

5.1 Messages photo-selection report

Feedback Assistant report FB19482757, submitted August 8, 2025, documents a repeatable cross-conversation photo-selection sequence. After previewing an unsent image in Conversation A through "View Photos," the researcher captured an image in Conversation B. Reopening B's photo picker surfaced A's image despite an apparently empty compose area in B; A's draft remained present. The original report recorded 100% reproduction with the "View Photos" step and listed iOS 18.6 and iOS 26 test contexts. The same cross-conversation picker behavior remains reproducible on October 17, 2026. The apparent context crossover prompted an initial privacy concern; the record does not establish an unintended send or disclosure.

The reported reproduction shows an image associated with a different draft or conversation appearing in the picker presentation. This motivated the initial privacy concern without, by itself, demonstrating disclosure. Whether selection state, cache lifetime, or asynchronous execution causes the behavior is unknown; no unintended send or disclosure has been demonstrated. The report and its captured reproduction sequence establish the observed condition, not its internal implementation.

5.2 Separate PhotoPicker/JPEG report and other submissions

On August 28, 2025, a separate Apple Product Security submission documented deterministic rendering anomalies and application crashes associated with extreme-dimension JPEGs in image-selection and attachment workflows. The investigation identified two distinct reproduction cases. The G1 variant exhibited maroon-colored rendering corruption and triggered an application crash during image selection. The G2 variant exhibited green rendering corruption and triggered an application crash during the send operation. Both reproduction sequences were captured in a contemporaneous screen recording submitted with the original Product Security report, as confirmed by the researcher.

The observed behavior differed across image-processing operations. Direct viewing handled some of the same inputs differently from picker and attachment workflows. These observations demonstrate operation-dependent differences in rendering and processing outcomes, including incorrect rendering and application failure. The recording documents the observable reproduction sequences but does not independently identify the underlying decoder implementation or internal crash mechanism. The JPEG findings remain distinct from the separate Messages investigation and do not establish that the two reports share a common defect.

Apple subsequently stated that an issue associated with the JPEG submission had been addressed in iOS 26.5, iPadOS 26.5, and macOS Tahoe 26.5. The statement does not specify the remediated code

path. It is retained as disposition history, separate from experimental observation.

5.3 FCA Uconnect application watchdog

An FCA Uconnect iOS diagnostic dated September 15, 2026 records termination during foreground scene creation. The selected fields below were extracted in an earlier technical examination of `fca_uconnect-2026-09-15-233910(1).ips`; they have not been independently re-extracted for this publication. [P-1]

```
captureTime: 2026-09-15 23:39:09.4070 -0500
procLaunch: 2026-09-15 23:38:48.7082 -0500
procName: fca_uconnect
bundleID: com.fca.myconnect.nafta
osVersion: iPhone OS 26.6.2 (23G90)
exception: EXC_CRASH / SIGKILL
termination.namespace: FRONTBOARD
WatchdogEvent: scene-create
code: 0x8BADF00D
exhausted real (wall clock) time allowance of 19.35 seconds
Elapsed total CPU time (seconds): 9.690
Elapsed application CPU time (seconds): 0.114
Thermal Level: 0; Thermal State: nominal
```

The diagnostic identifies a FrontBoard scene-creation watchdog termination at the application/OS lifecycle boundary: `EXC_CRASH/SIGKILL`, `FRONTBOARD`, `scene-create`, and `0x8BADF00D`. Capture and launch timestamps are 20.6988 seconds apart; the watchdog independently reports a 19.35-second wall-clock allowance. These quantities measure different intervals. Total CPU time was 9.690 seconds, including 0.114 seconds attributed to application CPU time. The record establishes a lifecycle timeout, but not the stalled operation, a CarPlay transport defect, or a vehicle-side consequence. [P-1]

5.4 Finding boundary

The Messages observation, JPEG-path behavior, and FCA watchdog event are separate findings at different boundaries. None supplies a direct transfer trace connecting an Apple device to the Jeep display or diagnostic events.

6 Case Study B: Jeep/Stellantis Vehicle Record

6.1 Service and diagnostic chronology

The vehicle history begins with a powertrain alert at 379 miles, a separate P2172 service event, an air-bag alert at 1,466 miles, and subsequent service-light reports. In March–April 2025, a dealer recorded warning indications, a pending U0287-00 code, its characterization of an electric oil-pump fault, replacement of that component, a failed battery test and battery replacement, and a post-repair test drive with no issue noted. Those entries describe a bounded service visit and do not explain every earlier or later event.

Later records include a September 27, 2025 maintenance entry, the March 2026 launch of Recall 12D, and June 2–3, 2026 TCM service. In September 2026 the customer reported a driver-assistance/electrical concern and separately described a CarPlay display condition involving a green line. The customer's reported onset chronology after TCM replacement does not establish that the TCM work caused either complaint or that the two complaint classes share a fault domain.

6.2 Reed diagnostic epoch and post-release sequence

A September 16, 2026 diagnostic printout photograph records stored codes including U0422-00, described as "Implausible Data Received From Body Control Module," and U148D-00, described as "Implausible Data Received From Forward Facing Camera," alongside other module codes. A separately retained code table identifies ETM B1498-19 ("USB Hub – Overcurrent") and B1570-19 ("USB Communication – Overcurrent"). These are recorded descriptions, not a demonstrated Apple-originated message path. The stored descriptions do not identify code set-time, active status during the later drive, the triggering message, or a common cause. [J-3]

Reed reported that it could not reproduce the condition during its test window and reported successful testing with an alternate cord and alternate Apple phone. The precise devices, software, cable, port, route, duration, pairing state, and vehicle state are not established. Separately, the customer identified a cable substitution while retaining the same owner phone and reported a recurrence after release. Phone-video artifacts are indexed at 15:26 and 15:34 CDT. The eight-minute difference is between the displayed timestamps, not a measured dashcam recording duration. The original dashcam/GPS card was preserved without acquisition at the time of that record. The September 16 dealership recording and post-departure observations are documented in Appendix G, Figures 1 and 2, respectively.

The dealer non-reproduction and the later customer-reported recurrence are separate outcomes under incompletely documented conditions. Neither falsifies the other. The cable intervention remains a customer-identified artifact claim pending independent indexing or transcription and is not an isolated-variable experiment.

Selected October 4 dashcam frames preserving roadway context and elapsed recording time, and separate phone-recorded instrument-cluster states including an ALM warning, are documented in Appendix G, Figures 3 and 4, respectively.

6.3 Radio firmware and published service coverage

A retained head-unit version photograph records Android 8.1.0, the March 5, 2020 security patch level, kernel `4.4.138-perf+`, and build `FCA_R1L_ARM64_00.47.03.26.00 release-keys`. The build corresponds to radio version `R47.03.26`. These values describe the photographed system; they do not establish its original factory software or subsequent update history. The service order and version photograph are shown in Appendix G, Figure 5. [J-5]

FCA TSB 08-172-24 (July 31, 2024) specifies a USB update to `R47.03.26` for eligible 2022–2023 Compass vehicles with listed Uconnect 5 radio configurations and build-date restrictions. The photographed release matches the bulletin's software version; the subject vehicle is identified on the September 16 service order as a 2024 Compass and therefore is not in that bulletin's stated model-year population. [T-1]

FCA TSB 08-215-24 (October 2, 2024) addresses eligible 2024–2025 Compass configurations and describes a separate firmware-over-the-air update from `U31.42` to `U33.43`. Applicability depends on build date

and specified equipment codes. The two bulletins do not establish that their `R-` and `U`-series labels belong to the same radio variant or can be directly compared as versions. No matching vehicle configuration or repair history has been demonstrated for the latter bulletin. [T-2]

The useful discrepancy is documentary: a software release photographed in the 2024 vehicle is also documented in a bulletin addressing an earlier, restricted model-year population, while a separate 2024–2025 bulletin describes another update branch. The comparison warrants radio-variant identification and service-history review alongside the reproducible display behavior. It does not establish an improper installation or the origin of the display artifact.

6.4 Evidence limits

The Jeep record supports service chronology, specific recorded code descriptions, the bounds of Reed's reported test, and an indexed phone-video interval. It lacks the full scan/freeze-frame data, a complete test protocol, synchronized vehicle and phone logs, the original dashcam/GPS contents, and a demonstrated technical relationship between the display complaint, driver-assistance report, diagnostic codes, and USB/CarPlay activity.

7 Cross-Vendor Interface Analysis

7.1 Established context and proposed relationship

The Apple and Jeep investigations began independently. A cross-vendor hypothesis asks whether a specified phone-side state or input could traverse the USB/CarPlay interface and alter vehicle-side presentation or diagnostics. No recorded transfer, identified receiver, or controlled vehicle outcome currently establishes that path. In particular, "implausible data" in a vehicle DTC is a diagnostic description, not identification of its sender.

Apple-device observations
↓
Apple evidence limitations

Jeep observations and diagnostic artifacts
↓
Jeep evidence limitations

Shared reported USB/CarPlay context
↓
Falsifiable interface hypothesis
↓
Synchronized acquisition and controlled testing

This sequence is an investigation plan, not an asserted architecture or propagation chain. The proposed hypothesis is: under controlled and repeatable conditions, a specifically characterized Apple-device input or state could be shown to participate in a USB/CarPlay session and to precede a reproducible vehicle-side display or diagnostic event, with synchronized evidence identifying the interface and receiving subsystem. Such a transfer has not yet been captured or experimentally reproduced.

7.2 Falsifiability and alternative explanations

Testing must compare the hypothesis against alternatives rather than presume it: the reported mobile behavior may be independent of the Jeep behavior; the vehicle condition may arise within a vehicle-side subsystem; the display and driver-assistance/electrical reports may be unrelated; or session, cable, device, software, or environmental conditions may affect reproduction without an Apple-originated malformed payload. These are candidate models for discrimination, not findings about the actual events.

Evidence that could support the interface hypothesis would include repeatable controlled reproduction, preserved and hashed input artifacts, synchronized device/interface/vehicle logs, defined code-trigger conditions, and a traceable temporal path to the receiving module. Evidence that the relevant code predated the session, arose from a documented unrelated source, or persisted without the implicated device/session under controlled comparable conditions could weigh against the proposed path. A single non-reproduction under unknown conditions would not resolve it.

7.3 Fault chronology and state-transition model

Two CarPlay fault entries record 15:26:30.8268 and 15:26:37.3639, separated by 6.5371 seconds. Their original reports have not been inspected in this edition; calendar date, timezone, process, device, exception, and session metadata remain unresolved. The apparent match to the Jeep phone-video index at 15:26 CDT is therefore not a synchronized event correlation. The September 15 FCA app termination and September 16 vehicle service/media record also remain distinct events. The head-unit build and radio version are documented separately by the version photograph in Section 5.3. [P-1; J-5]

The recorded layers are application lifecycle (FCA watchdog), interface session (two provisional CarPlay entries), presentation (repeated display artifact), and vehicle diagnostics (USB-related and inter-module codes). Each exposes a different projection of runtime state. A plausible divergence mechanism is an asynchronous transition in which the application, session, and head unit update or retain state on different schedules. Establishing that mechanism in this vehicle requires synchronized logs and a trace to the receiving subsystem; no such cross-layer sequence has been demonstrated. [P-1]

7.4 Candidate mechanisms and discriminating observations

Five candidate mechanisms can be evaluated against the available observations without assigning them equal evidentiary weight. Missing or non-synchronized evidence constrains attribution but is not affirmative disproof. [P-1]

App lifecycle stall
: The FCA app fails to complete foreground scene creation before the watchdog deadline. The transcribed IPS fields establish the termination; the operation that stalled is unknown. A re-openable IPS, stacks, signposts, launch dependencies, and app-only repeat trials would discriminate waiting, initialization, and scheduling explanations. Current interpretation: high-confidence app-lifecycle termination, not a CarPlay transport finding.

Independent infotainment rendering
: A head-unit presentation-state defect could produce stale or abnormal display behavior without the phone. The Jeep display complaint is supporting context, but no disconnected-phone or alternate-source control establishes independence. Head-unit logs, source switching, display capture, and phone-disconnected reproduction are needed. This remains a live vehicle-side explanation.

CarPlay session-state mismatch

Session or pairing transitions could leave phone and head-unit state out of agreement. The secondary timestamp pair is a reconciliation lead only; date, timezone, process, session, and event alignment are unavailable. Original reports plus synchronized phone/head-unit capture across controlled connection and source-switch conditions would test it. It remains testable but underdetermined, not the preferred causal conclusion.

Vehicle module/network issue

A vehicle-side module or network condition could account for stored codes and driver-assistance/electrical reports. The reported U0422-00, U148D-00, B1498-19, and B1570-19 descriptions provide diagnostic context, but event-time status, trigger, source module, and linkage are unknown. Full scan/freeze-frame records, module topology, timestamps, and no-CarPlay controls would discriminate this candidate. It remains a live explanation.

Shared cross-vendor defect

A specific state or payload could cross the Apple/CarPlay/vehicle boundary, be accepted or misinterpreted by a receiving parser or state machine, and produce a reproducible vehicle effect. No transferred data, receiving state, synchronized path, or repeatable downstream effect is documented. A source-to-receiver trace and controlled reproduction with removal of the implicated transfer as a negative control are required. The candidate is retained as a hypothesis, but it cannot be preferred without synchronized evidence.

8 Observability and Evidence Limitations

8.1 State Continuity and Representation Fidelity

The Apple observations motivate a state-continuity explanation because they concern behavior across changes in conversational or media-handling context. In the Messages report, a selection associated with one conversation was reported as visible in another conversation's picker flow. In the separate PhotoPicker/JPEG report, the same described files reportedly behaved differently in direct viewing and picker or attachment paths. Both observations are compatible with a divergence between the state a user believes is current and the representation retained or presented by a particular application path. The mechanism-level interpretation is that independently maintained selection, draft, or media representations may persist or be transformed across a context transition without being reconciled at the moment they become visible. This explains why a state can appear continuous to the user while its representation or ownership differs between paths; it does not identify a particular internal store or prove that the two reports share an implementation.

The FCA event supplies an independent lifecycle observation: FrontBoard terminated the application during scene creation. A separate interface or presentation layer could retain state through this transition, but the IPS contains no direct measurement of head-unit state. The watchdog therefore establishes an application lifecycle failure, not a CarPlay transport failure. [P-1]

8.2 Cross-System Observability

Application lifecycle, CarPlay session entries, display output, and diagnostic codes observe different layers of the system. The FCA watchdog identifies a process termination; the CarPlay pair records provisional times; the Jeep evidence records visual behavior and U0422-00, U148D-00, B1498-19, and B1570-19 descriptions. These observations cannot be collapsed into a single event clock without source

metadata and synchronization. [P-1]

This layered view explains the significance of partial observability: a local transition need not be represented identically at every boundary. An application may terminate during scene creation; an interface session may record faults on its own timeline; a head unit may retain or present state; and a vehicle module may record a diagnostic condition without exposing the triggering message or its source. Consequently, the visible interface, the application lifecycle record, and vehicle codes may differ in fidelity and temporal resolution. Vendor-controlled diagnostics can reveal that a condition was recorded while withholding the event-time context needed to reconcile it with another layer. The Jeep code descriptions, including "implausible data," do not by themselves identify the originating interface or event. [P-1]

8.3 Explanatory Significance

Taken together, the observations motivate a state-divergence mechanism: when application, session, presentation, and diagnostic representations are managed separately, asynchronous transitions can leave one representation current while another remains retained, transformed, or differently exposed. At the user interface, that divergence can appear as state persisting across a context change or behavior varying by the path through which media is handled. At a system boundary, it provides a technically coherent way to reason about why an application event, interface fault record, vehicle display, and diagnostic code might not present a synchronized account of state. This model is explanatory because it connects the observed forms of behavior through state ownership and representation transitions, rather than treating each visible output as a transparent view of a single underlying state. [P-1]

This model explains how nonidentical representations can coexist without requiring a shared defect. The September 15 app watchdog is a documented lifecycle event; the September 16 vehicle observations form a separate service and media chronology; the CarPlay fault pair has unresolved date and session context. A cross-vendor propagation claim requires an identified transfer and downstream effect, not merely plausible state divergence. [P-1]

8.4 Evidence Boundaries

The FCA fields are reproduced from a prior extraction rather than re-extracted here; the original CarPlay fault reports remain unverified in this edition. Complete scan and freeze-frame data, synchronized phone/head-unit logs, and full source-video acquisition are required for precise event alignment and source attribution. [P-1]

9 Experimental Lineage: Decoder Stress, Watermark Recovery, and Representation Fidelity

The experimental investigation began with watermark insertion and recovery rather than extreme-geometry image generation. A repeated token was embedded into the M16 source image using a DWT/DCT/SVD watermarking procedure, with frequency-based majority voting applied to recovered token candidates. The resulting `M16-Cropped_wmtest.jpg` artifact became the initial input to the Linux mozjpeg/libjpeg processing pipeline. Subsequent transformations produced PPM intermediate representations organized into a letter-and-number variant ladder. These variants supported progressively more demanding image-decoding tests, including extreme-geometry JPEG generation. Following the

Linux transformations, `bug_test.py` was used to evaluate watermark embedding and extraction against the modified image.

9.1 Experimental artifacts and observables

The experimental materials include source code, a retained JPEG header, SHA-256 inventories, saved PixInsight transforms, and Fourier-processing methods. The scripts define input construction and attempted extraction; the retained artifacts identify format and configured operations. Together they specify testable representation changes. A device-specific processing outcome requires the corresponding run record. [W-1–W-5]

An October Bluetooth diagnostic records `GUARD_EXC_MTE_SYNC_FAULT` with `isSimulated: 1`. A separate accessory RTKitOS panic identifies audio-accessory metadata; later iPhone 17 Pro diagnostics contain camera-graph, JPEG, RTBuddy, and MTE-related worker references. These are distinct fault records with different execution contexts. Their common subsystem vocabulary supports comparison, but does not establish identical input lineage or propagation. The significance of `isSimulated: 1` remains unresolved. [W-6]

9.2 Extreme-geometry JPEGs as decoder-boundary inputs

The `make_extremes.sh` generator constructs images of dimensions 1×4096 and 4096×1 and encodes baseline 4:4:4 JPEGs with mozjpeg, using ImageMagick for fixed-geometry input preparation. [M-1–M-3] These dimensions hold pixel count constant while exchanging the dominant spatial axis and thereby provide a controlled way to examine path-sensitive decoding and presentation. The retained `G2_444_RGB_baseline_4096x1.jpg` header is consistent with a 4096-by-1 baseline JPEG with three components and 1-by-1 sampling; its reported SHA-256 is `57b87ebe9c0dae85bd9342dcc9bd6f64e5a1a94a0f16cba283930ca974ac20ab`. The header establishes the file's format characteristics, not which Apple framework processed it during any recorded event. [W-1, W-2]

Implementation excerpt: extreme-geometry JPEG generation. The generator fixes geometry with `-resize` and selects baseline 4:4:4 JPEG encoding using `-sample 1x1` and `-baseline`:

```
convert "$SRC" -resize 4096x1\! "$OUT/G2_4096x1.ppm"
"$CJPEG" -sample 1x1 -quality 75 -optimize -baseline \
  -outfile "$OUT/G2_4096x1_444.jpg" "$OUT/G2_4096x1.ppm"
```

The displayed generator commands identify `G2_4096x1_444.jpg` as their output. The separately retained artifact `G2_444_RGB_baseline_4096x1.jpg` has the JPEG characteristics and SHA-256 digest reported above. The retained artifact was subsequently configured as the input to `bug_test.py`. The displayed commands do not independently establish a byte-identical relationship between the generator output and the retained artifact. Accordingly, the filenames are preserved as distinct artifact identifiers rather than treated as interchangeable. [W-1–W-3]

9.3 Robust watermarking as a provenance instrument

This probe extends the watermark investigation to the retained G2 extreme-geometry representation. The displayed configuration identifies the input and output filenames. The reported procedure involved watermark embedding and extraction, but the excerpt above contains only artifact-path assignments and does not independently demonstrate the invoked operations or their results. No successful recovery result is inferred from those assignments alone. [W-3]

Implementation excerpt: watermark embedding and serialization. The implementation identifies the source image, output artifacts, embedding method, repeated payload, serialization formats, and subsequent JPEG recovery attempt:

```
INPUT_IMG = "M16-1.jpg"
OUT_JPG = "M16-Cropped_wmtest.jpg"
OUT_PNG = "M16-Cropped_wmtest.png"

METHOD = "dwtDctSvd"
BASE_TOKEN = "M16WM001"
REPEAT = 8
PAYLOAD = SEP.join([BASE_TOKEN] * REPEAT)

enc.set_watermark("bytes", PAYLOAD.encode("utf-8"))
marked = enc.encode(src, method=METHOD)

cv2.imwrite(OUT_JPG, marked,
            [int(cv2.IMWRITE_JPEG_QUALITY), JPEG_Q])
cv2.imwrite(OUT_PNG, marked)
decode_from_path(OUT_JPG, METHOD)
```

The configured input is `M16-1.jpg`, and the serialized JPEG output is `M16-Cropped_wmtest.jpg`. The latter was subsequently used as the initial input to the Linux image-processing workflow. The implementation attempts watermark recovery after JPEG serialization; successful recovery is not inferred from the source code alone.

Implementation excerpt: transformed-image watermark probe. Following the Linux image transformations, `bug_test.py` performed a subsequent watermark embedding and extraction experiment using the extreme-geometry JPEG `G2_444_RGB_baseline_4096x1.jpg`.

The script explicitly configured the following input and output artifacts:

```
INPUT_IMG = "./G2_444_RGB_baseline_4096x1.jpg"
OUTPUT_IMG = "./G2_444_RGB_baseline_4096x1_wm.jpg"
```

This probe extends the original watermark investigation to a transformed image representation. Unlike the initial `wm_robust.py` stage, which operated on `M16-1.jpg`, this subsequent implementation operated directly on the G2 extreme-geometry artifact.

The configured operations establish the intended embedding and extraction procedure but do not, by themselves, establish successful watermark recovery.

9.4 Later image processing, watermark-blend strength, and phase-domain operations

A 650-entry image-processing inventory preserves relative names, file sizes, timestamps, digests, and header bytes. PixInsight project metadata [M-5, M-6] separately records three Fourier and three inverse Fourier operations, magnitude/phase views, and PixelMath settings that combine the green channel with `Image12` using coefficients 0.01, 0.07, 0.0996, and 0.0000000966, with recorded clipping to $[0, 1]$. These are saved process settings, not a measured watermark-detection response curve. The settings alone do not determine the exact operand order, input normalization, `Image12` identity, or which output corresponds to each coefficient. [W-4]

The coefficients represent *watermark contribution strengths*, controlling the amount blended into the main image rather than modifying the Fourier transform itself. The blending was performed on RGB data, followed by PixInsight LRGBCombination. The recorded metadata documents the coefficient settings, while the processing sequence is established through the author's firsthand methodological account. [W-4]

Separate retained PixelMath instructions document synthetic SHO channel assembly and other astronomical processing operations. For example, the SHO instructions specify `G*0.7+B*0.3` as the construction for `Oiii`; the star-combination and luminance-sharpening notes retain distinct expressions. These operations establish concrete image-processing techniques used in the wider image workflow, *not* the missing `Image12` watermark-blend expression or a coefficient-to-output assignment. [W-7]

A Fourier methods document describes magnitude/phase inspection, spectral masking, inverse transforms, and channel synthesis. The preserved inventory also includes watermark-named and inverse-transform outputs, but it does not establish that every Fourier operation was applied to every watermarked image or measure marker recovery after those steps. [W-4, W-5]

The experimental record contains a connected progression from initial DWT/DCT/SVD watermark embedding through Linux image transformations and extreme-geometry JPEG testing, followed by additional watermark embedding and extraction attempts against a transformed artifact. The later PixInsight Fourier-processing and watermark-blending records extend the broader investigation of representation fidelity, but the available metadata does not establish their complete execution order or a one-to-one correspondence with every earlier image variant. These limitations concern the reconstruction of particular processing operations, not the documented relationship between the initial watermark artifact and the subsequent geometry-testing workflow. [W-1–W-7]

9.5 An explicit fidelity test model

Let $\mathcal{R}_k(x)$ denote an encoded or decoded *artifact representation* of source x, and let $F_{k\to\ell}$ be the declared transform from representation k to representation ℓ. Select feature measurements ϕ_k and ϕ_ℓ on the respective representation domains and an expected feature mapping g_F appropriate to that transform. Define the transformation discrepancy as

$$D_{F,\phi}(x) = d_\phi\big(\phi_\ell(F_{k\to\ell}(\mathcal{R}_k(x))),\, g_F\big(\phi_k(\mathcal{R}_k(x))\big)\big)\,. \tag{1}$$

Here d_ϕ is defined for the chosen feature and ϵ_ϕ is a declared acceptance threshold. *Feature preserved* means $D_{F,\phi}(x) \le \epsilon_\phi$; *measured departure* means $D_{F,\phi}(x) > \epsilon_\phi$ with both operands observed; missing input, output, calibration, or comparison rule makes the result indeterminate. This does not assume that every feature is mathematically invariant. For geometry, ϕ may be the ordered integer pair (w, h), g_F the expected decoded dimensions, and d_ϕ exact pair mismatch with threshold zero. For a marker, ϕ can be an extracted bit string compared under Hamming distance against a declared payload and bit-error

threshold. A phase statistic requires its own domain and circular or otherwise suitable distance. Each experiment must declare these choices; none supplies a universal cross-system tolerance. A measured departure describes the tested representation transform, not a vendor implementation fault by itself.

10 Restore-Path Investigation and Controlled Artifact Perturbation

The iPhone 15 Pro restoration investigation initially focused on repeated failures at operation `0x6B4` (`await_system_image_invert_retry`), where updater logs recorded CFNetwork timeouts, retry exhaustion, unavailable WKMS authorization tokens, and error `1109 / 06B4.0000`. Network observations identified an active NordLynx route and a connection associated with the public WKMS endpoint, motivating investigation of host connectivity as a potential contributing condition. Following a transition to direct Wi-Fi with NordLynx disabled, a subsequent restoration completed `0x6B4`, streamed ASR to completion, and returned status `0`. Earlier unsuccessful attempts with the VPN disabled prevent attributing this success exclusively to the route change.

The successful restoration established an observed execution path for subsequent experimental investigation. Its updater log recorded successful delivery of `SystemVolume_043-69111-662.dmg.aea.img4` during Customer Erase Install. This observation motivated a controlled artifact-perturbation experiment targeting the corresponding OS member of the selected `d83ap` Erase BuildIdentity.

The intervention modified a single bit at byte offset 4,294,967,296 within `043-69111-662.dmg.aea`, preserving the surrounding manifest and signature material. The modified member was introduced through the Apple Devices cache under the expected filename. The experiment compared restoration attempts associated with this modified cache against a subsequent restoration using the preserved original vendor cache member.

Two subsequent restoration attempts identified as using the modified cache, Runs 10 and 11, terminated with `1109 / 06B4.0000`. Both recorded WKMS-related timeout and authorization failures at operation `0x6B4`. Run 12, performed after restoring the preserved original cache member, completed ASR and returned status `0`.

Run	Start (UTC, Oct. 3)	Cache condition	Observed result
10	14:49:00.0552	Single-bit modified member	`0x6B4`; WKMS no token; `1109/06B4.0000`
11	15:04:01.0282	Same modified-cache condition	`0x6B4`; WKMS no token; `1109/06B4.0000`
12	15:32:06.0144	Original vendor cache	`0x6B4` complete; ASR 100%; status `0`

Table 2: Recorded restoration starts and outcomes on October 3, 2026. Runs 10 and 11 used the modified cached OS/SystemVolume member; Run 12 used the preserved original. The iPhone 15 Pro and selected restore target build were common to the reported comparison. The original updater logs provide the primary records for the timestamps, operation identifiers, and reported outcomes.

Comparison conditions and diagnostic checkpoints. The reported comparison used the same iPhone 15 Pro and restore target build. The selected `d83ap` Erase BuildIdentity identified the OS/SystemVolume member targeted by the cache intervention. The controlled change was the replacement of the original cached member with a modified version under the expected filename, followed by restoration of the preserved original member for Run 12.

The recorded runs completed the pre-`0x6B4` baseband and FDR verification checkpoints, including

`baseband_postseal` and `fdr_verify_sealed_manifest`. The eUICC parameter lines remained unchanged in the recorded failed trials.

The retained run descriptions identify the intervention condition, restore operation, error status, and completion outcome. The exact bit index within the modified byte, original and modified member digests, complete host and cable configuration, and verbatim primary log excerpts are not reproduced in this manuscript section. These details remain relevant to independent reproduction and should be included in an accompanying reproducibility deposit where supported by the original experimental records. [W-6]

Interpretation and causal limitations. The modified-cache trials exhibited repeated failure at the same restore operation, while the subsequent original-cache trial completed successfully. This constitutes an observed association between the recorded cache conditions and restoration outcomes.

However, the available comparison does not establish equivalent network configurations across all three runs. The earlier host-network investigation also demonstrated that VPN state alone did not determine restoration success. Consequently, the observed outcomes do not isolate the single-bit cache modification as the independent cause of WKMS authorization failure.

The investigation establishes two experimentally relevant observations: successful restoration following a host-network intervention and differing restoration outcomes associated with a subsequent SystemVolume cache perturbation.

Neither observation independently establishes Apple's internal WKMS authorization mechanism, proves that the modified bytes reached a device-side consumer, or demonstrates a causal relationship with the earlier JPEG experiments. [W-6]

11 Discussion

The relationship between the mobile and automotive investigations emerged from an observed similarity in rendering behavior. During controlled JPEG experiments, the researcher observed localized green visual corruption associated with an intentionally constructed image input. A subsequently observed green rendering artifact in the vehicle's CarPlay display resembled the earlier behavior, motivating a comparative investigation into how each system processed, transformed, and presented state across subsystem boundaries. This resemblance served as an investigative trigger rather than evidence of a shared defect or causal relationship. [W-1, W-2; J-1–J-4]

Subsequent automotive observations, including stored diagnostic trouble codes and USB overcurrent indications, provided additional reasons to examine the interaction between device communication, application execution, and vehicle-side presentation. These records expanded the scope of the investigation but did not establish that the earlier JPEG input caused the automotive behavior. Similar visual symptoms can arise from different underlying mechanisms, and the available evidence does not identify a common implementation fault. [P-1; J-1–J-4]

The investigations nevertheless share a substantive methodological relationship. In both environments, externally observable behavior was compared against available diagnostic records to identify where expected state transitions became difficult to reconstruct. The mobile investigation employed controlled image transformations, decoder stress testing, and restore-path interventions to examine representation fidelity and system behavior. The automotive investigation combined display observations, application diagnostics, vehicle records, and interface-level evidence to characterize failures whose underlying execution paths were not directly observable. [W-1–W-6; P-1; J-1–J-4]

The resulting cross-vendor comparison illustrates a broader challenge in systems analysis: an observable anomaly may establish that expected behavior has diverged without identifying the transformation, component, or boundary responsible. Diagnostic records can narrow the investigation, but their interpretation depends on the execution context, temporal ordering, and relationship to the observed artifact. Similarity between symptoms is therefore insufficient for causal attribution, while independently collected evidence can still support meaningful comparisons of how different systems expose—or obscure—their internal state. [W-1–W-6; P-1; J-1–J-4]

The central contribution of this comparison is not a claim that the mobile and automotive incidents share a common root cause. Rather, it demonstrates how an initially observed rendering similarity motivated a systematic investigation across distinct technical environments, using controlled interventions where possible and preserving uncertainty where direct causal evidence was unavailable. The findings support an evidence-bounded approach to cross-vendor state observability in which experimental lineage, representation changes, diagnostic provenance, and explicit causal limitations remain central to interpreting complex system behavior. [W-1–W-6; P-1; J-1–J-4]

12 Threats to Validity

12.1 Construct validity

"State continuity," "representation fidelity," and "interface participation" describe analytical properties evaluated through observable system behavior and experimental artifacts.

Representation fidelity is evaluated through controlled image transformations and observable processing outcomes, including correct rendering, incorrect rendering, and complete processing failure. These categories reflect the experimental methodology and documented observations rather than exclusively hypothetical classifications.

The August 28, 2025 JPEG investigation provides two distinct reproduction cases. The G1 variant exhibited maroon-colored rendering corruption and triggered an application crash during image selection. The G2 variant exhibited green rendering corruption and triggered an application crash during the send operation. Both reproduction sequences were captured in a contemporaneous screen recording submitted with the original Apple Product Security report, as confirmed by the researcher.

These observations establish distinct rendering and application failure conditions but do not independently identify the underlying decoder implementation or internal failure mechanism. The outcome categories characterize observable behavior rather than establish a particular internal cause.

State continuity and interface participation remain more indirectly observable where vendor-internal state and cross-component execution paths are unavailable. Accordingly, the study distinguishes experimentally observed representation outcomes from inferred state transitions and unverified cross-system propagation mechanisms. [W-1–W-3; P-1]

12.2 Internal validity

The mobile restore comparison and the cross-vendor investigation present distinct causal limitations. In the restore experiment, the cached SystemVolume member was deliberately modified, and subsequent trials recorded differing outcomes between modified-cache and original-cache conditions. However, equivalent host-network configurations were not established across the comparison runs, preventing

isolation of the cache modification as the cause of the observed restoration failures. [W-6]

Separately, no synchronized device, interface, and vehicle logs or controlled cross-vendor experiment establish a causal path between the Apple-side observations and the Jeep behavior. Device, cable, software, vehicle state, timing, environmental conditions, and test protocols remain incompletely specified. These limitations prevent attribution of the automotive observations to a particular Apple-originated input or shared implementation defect. [P-1; J-1–J-4]

12.3 External validity

The documented investigations cannot support prevalence estimates or generalization to other devices, vehicles, software releases, or users. Experimental observations and reported reproductions remain bounded by their recorded configurations and conditions. [W-1–W-6; P-1; J-1–J-4]

12.4 Reproducibility

Reported reproductions, controlled image experiments, restoration trials, and vehicle recurrence involve different test conditions and levels of experimental control. The available source code supports reconstruction of specified image-processing procedures, while complete run-level records and independent replication remain necessary to establish reproducibility across environments. A common cross-vendor reproduction protocol has not been established. [W-1–W-6; P-1; J-1–J-4]

12.5 Source provenance

Some observations are supported by retained transcriptions, photographs, source-code excerpts, or secondary records rather than independent re-examination of the original artifacts. These distinctions are preserved in the evidence register. Source-code inspection establishes configured experimental behavior but does not, without corresponding execution records, establish the outcome of a particular run. [P-1; W-1–W-6; J-1–J-5]

12.6 Researcher interpretation bias

The cross-vendor hypothesis originated from the observation of visually similar green rendering artifacts in the controlled JPEG investigation and the subsequent CarPlay display behavior. This resemblance motivated further investigation rather than establishing a shared technical mechanism. Later examination of vehicle diagnostic records, including USB overcurrent indications, increased the relevance of investigating the interface but did not independently establish a causal connection. [W-1, W-2; J-1–J-4]

Because the initial visual similarity influenced the direction of subsequent evidence collection, confirmation and selection bias remain potential limitations. Explicit separation of observations, hypotheses, and causal findings, together with consideration of independent vehicle-side explanations, mitigates but cannot eliminate these risks.

12.7 Vendor-side incompleteness

Vendor telemetry, complete diagnostic protocols, internal implementation details, and relevant module software behavior are not presently available for independent examination. These gaps constrain reconstruction of internal state transitions and identification of the components responsible for observed behavior.

12.8 Temporal and causal uncertainty

Report dates, service dates, displayed device times, collection dates, experimental execution times, and later reconstructions represent distinct temporal references. Clock normalization and synchronization remain incomplete across the investigations. Temporal proximity, repeated visual characteristics, and diagnostic co-occurrence are not treated as sufficient evidence of causation. [W-6; P-1; J-1–J-4]

13 Future Work

The state-divergence model is the present explanatory account; the following evidence-acquisition steps and experiments are designed to determine its applicability, operating conditions, and relationship to particular events. They are not prerequisites for articulating the model. All testing remains subject to appropriate owner authorization, counsel direction, and safe test design.

1. **FCA lifecycle reproduction:** Obtain and hash the original FCA Uconnect IPS report; confirm the extracted process, exception, termination, timestamp, and scene-creation fields. Capture stacks, signposts, and launch dependencies if available, then reproduce the scene-create timeout in an app-only test. This can identify the operation or dependency associated with the watchdog event and distinguish app-lifecycle conditions from a CarPlay transport explanation. [P-1]
2. **CarPlay session synchronization:** Obtain the original reports underlying the two fault timestamps and verify their date, timezone, process/device identifiers, exception fields, and session metadata before aligning the events. Synchronized capture across application lifecycle and session transitions can test whether those representations diverge in time, establish the conditions under which they do so, or falsify an asserted temporal relationship. [P-1]
3. **Presentation-state testing:** Acquire native Apple report exports, attachment manifests, original reproduction media, relevant device logs, exact OS/build details, and report-specific Apple review or remediation records. Reproduce the Messages and PhotoPicker/JPEG reports separately, tracking which selection or media representation is displayed across context and path changes. For the vehicle display, compare controlled source switching and phone-connected versus phone-disconnected conditions. These tests can support or distinguish presentation-state persistence from path-specific behavior without presuming a shared implementation.
4. **Jeep diagnostic reconstruction:** Preserve and forensically acquire the unread dashcam/GPS source and original phone media; document hashes, custody, time zones, clock sources, and synchronization uncertainty. Obtain full scan output, DTC status and freeze-frame data, module/software/calibration details, repair records, and Reed's exact test protocol, along with authoritative code descriptions. This can establish which subsystem generated each recorded condition and when, and distinguish a vehicle-local diagnostic event from a condition temporally associated with an interface session.
5. **Cross-vendor controlled testing:** If source evidence supports a safe test, fix vehicle, phone, OS, cable, port, pairing, route, and environmental conditions; vary specified factors independently; record synchronized device/interface/vehicle data; and include baseline and repeat trials. Predefine results

that would support state divergence across the interface, distinguish it from the existing competing explanations, identify operating conditions, or falsify a specific predicted relationship. Avoid testing that could create unsafe vehicle behavior.

6. **Source and publication review:** Seek independent technical review of provenance and the experimentally supported claims. Address external scholarly literature, disclosure review, and any required permissions or privacy review in the publication stage; these activities do not replace the explanatory model or alter the evidence status reported here.

14 Conclusion

This study establishes a practical distinction between *observing a representation*, *testing a transformation*, and *demonstrating a causal dependency*. The extreme-geometry JPEG generator and watermark script provide inspectable test construction. Saved Fourier operations and variable watermark-blend settings extend the inquiry into controlled image transformation; the coefficient-to-output mapping has not been reconstructed. The restore records show two runs identified as using a modified cached member failing at `0x6B4` with status `1109`, followed by a successful run using the original cache. Because network configurations are not established as equivalent across those runs, this comparison does not isolate the cache change as the cause. The Messages cross-conversation picker behavior remains reproducible on October 17, 2026. [W-1–W-6]

The Jeep case separately documents an FCA application watchdog, repeated display behavior, vehicle diagnostic codes, and a photographed radio release that can be compared with two NHTSA-hosted service bulletins. Those findings motivate an asynchronous, layered state model. No source-to-receiver path has yet linked an Apple experiment to a Jeep event. [P-1; J-5; T-1; T-2]

The central result is a defensible observability framework supported by source excerpts, documented restore-path observations, and bounded incident records. It permits affirmative conclusions where experiments discriminate outcomes and identifies the precise missing measurements where cross-system attribution remains open.

Evidence References

The following identifiers label the underlying records and experimental artifacts. Public service bulletins are linked directly; unpublished artifacts are identified by type, date, or digest for evidentiary retrieval.

A-1–A-6
: Apple Feedback Assistant and Product Security records, dated correspondence, technical observations, and report-specific dispositions, including FB19482757 and the August 28, 2025 JPEG submission.

J-1–J-4
: Jeep/Stellantis service and diagnostic chronology, September 16, 2026 diagnostic printout, dealer testing account, and September 17 post-release chronology.

X-1
: September 18, 2026 cross-vendor research hypothesis and associated evidence chronology.

P-1
: September 15, 2026 FCA Uconnect iOS diagnostic, fields extracted in a prior technical examination; October CarPlay timing entries retain provisional status.

W-1
: Retained image-processing source, checksums, and recorded processing metadata.

W-2
: JPEG stress-input source and retained 4096-by-1 baseline 4:4:4 JPEG metadata, including the recorded SHA-256 digest.

W-3 DWT/DCT/SVD watermark implementation, configured payload, serialization steps, and source digest.

W-4 Image-processing inventory and PixInsight Fourier/PixelMath operation history.

W-5 Fourier-processing workflow documentation.

W-6 Apple diagnostic event records and cached-firmware restore comparison (runs 10–12), including the iPhone 15 Pro/23G83 target and selected OS member `043-69111-662.dmg.aea`; run start times and failure/success checkpoints transcribed from the previously analyzed restore record. The modified byte offset is reported in Section 9; exact modified-member digest and bit index within that byte remain unlisted.

W-7 Retained original PixelMath expression notes for synthetic SHO channel assembly, star composition, and luminance sharpening. These are separate documented processing expressions, not evidence of the `Image12` watermark blend expression.

J-5 Retained FCA R1-Low version photograph; system build `FCA_R1L_ARM64_00.47.03.26.00`, radio release `R47.03.26`; 2024 Jeep Compass designation from September 16 service record.

T-1 FCA, TSB 08-172-24, July 31, 2024: radio USB update to R47.03.26 for specified 2022–2023 Compass vehicles.

T-2 FCA, TSB 08-215-24, October 2, 2024: U31.42 to U33.43 FOTA update for specified 2024–2025 Compass vehicles.

Methods and Software References

The following are standards and tool/method references, distinct from the case-study evidence keys. Tool versions and local package builds are not recoverable from the excerpts quoted here and are not inferred from current online documentation.

M-1 ITU-T, Recommendation T.81 (1992), *Digital compression and coding of continuous-tone still images: Requirements and guidelines* (equivalent to ISO/IEC 10918-1).

M-2 Mozilla, mozjpeg `cjpeg` command reference, including `-sample`, `-quality`, `-optimize`, and `-baseline` option semantics.

M-3 ImageMagick, Command-line processing and image-geometry documentation, including force-size `!` geometry.

M-4 `imwatermark` (invisible-watermark) API documentation, including the `set_watermark` encoder call and `dwtDctSvd` method. The method and API match the shown calls; the installed release is not established by the excerpt.

M-5 PixInsight, Reference Documentation source repository; see the PixelMath and image-processing reference materials.

M-6 PixInsight technical staff, LRGBCombination operation and channel-weight explanation, PixInsight Forum (2009). This explains the tool semantics; it does not authenticate the order of the author's historical process instances.

Scholarly References

R-1 Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7), 558–565. `https://doi.org/10.1145/359545.359563`

R-2 Chandy, K. M., and Lamport, L. (1985). Distributed snapshots: Determining global states of distributed systems. *ACM Transactions on Computer Systems*, 3(1), 63–75. `https://doi.org/10.`

1145/214451.214456

R-3 Pearl, J. (2009). *Causality: Models, Reasoning, and Inference*. Second edition. Cambridge University Press.

A Selected Chronology

Date	Matter / record	Bounded proposition
2025-08-08	Apple, FB19482757	Displayed Feedback Assistant report date; reported Messages/photo-selection scenario.
2025-08-28	Apple, separate PhotoPicker report	Product Security submission with researcher-reported extreme-dimension JPEG behavior.
2025-09-27	Jeep maintenance entry	Service-contract maintenance event at 10,417 miles; primary underlying work detail is unresolved.
2026-03-03	Jeep Recall 12D	Recall launch recorded; no causal relation to later behavior established.
2026-06-02–03	Jeep TCM service	Documented service chronology; not a cause finding for later complaints.
2026-07-01	Apple PhotoPicker disposition	Apple stated an issue associated with that report was addressed on specified 26.5 platforms.
2026-09-15	FCA Uconnect iOS diagnostic	FrontBoard scene-create watchdog recorded at 23:39:09 -0500 (fields retained from prior extraction). [P-1]
2026-09-15–16	Jeep display / Reed service epoch	Separate display and driver-assistance/electrical complaints; diagnostic printout and dealer test record.
2026-09-16	Jeep phone-video interval	Indexed artifacts at 15:26 and 15:34 CDT; eight minutes between displayed phone timestamps.
2026-09-17	Jeep evidence preservation	Dashcam/GPS source preserved; acquisition and synchronization pending.
2026-09-18	Cross-vendor hypothesis	Apple-originated input and vehicle response proposed for source-to-receiver testing.
2026-10-15	CarPlay timing note (record date)	Secondary record preserving two time-of-day entries 6.5371 seconds apart. The underlying events' calendar date, timezone, and session metadata remain unverified. [P-1]
2026-10-17	Experimental review	Source-code methods, retained artifacts, and controlled cache comparison consolidated. [W-1–W-6]

B Observation–Hypothesis Boundary

Record item	Evidence class	What remains unresolved
Messages image displayed in another conversation's picker	Researcher-reported observation; source-captured report	Independent reproduction, internal state owner, mechanism, actual send or disclosure.
Extreme-dimension JPEG behavior in picker paths	Researcher-reported observation and artifact inventory	Raw files, independent replication, exact remediated issue.
FCA Uconnect scene-create watchdog	Fields extracted from diagnostic	Raw IPS/hash, blocking operation, stack, CarPlay participation, and vehicle relationship. [P-1]
CarPlay fault time pair	Secondary note, provisional	Original reports, date, timezone, process, session, and alignment to Jeep event. [P-1]
U0422-00, U148D-00, B1498-19, B1570-19	Diagnostic descriptions in Jeep records	Set time, active status, source message, triggering event, relationship to drive/interface. [J-3]
Reed alternate phone/cable test	Vendor statement	Complete protocol and comparability to the owner-reported conditions.
Post-release interval	Indexed phone-video metadata and customer report	Dashcam contents, synchronized event timing, exact route and causal relation.
Apple-device-to-vehicle propagation	Researcher hypothesis	Any transferred abnormal data, receiving subsystem, repeatable effect, or causal link.

C Diagnostic Artifact Register

This register identifies the evidence behind each diagnostic observation and the status of its original record. A retained extraction is not labeled a new independent examination.

Record / provenance	Observed fields or report	Interpretive boundary
FCA Uconnect iOS diagnostic; September 15, 2026. [P-1]	September 15, 2026, 23:39:09.4070 -0500; `fca_uconnect`; `com.fca.myconnect.nafta`; iPhone OS 26.6.2 (23G90); `EXC_CRASH/SIGKILL`; `FRONTBOARD`; `scene-create`; `0x8BADF00D`.	OS-enforced app scene-creation watchdog; not, by itself, a CarPlay transport, USB, head-unit, or vehicle fault.
CarPlay fault entries; originals pending reinspection. [P-1]	15:26:30.8268 and 15:26:37.3639; difference 6.5371 seconds. Date, timezone, process, session, and device metadata not established.	Provisional timing lead; not synchronized with Jeep video or FCA IPS.
Jeep September 16 diagnostic printout and September 17 memo [J-3].	U0422-00: Implausible Data Received From Body Control Module; U148D-00: Implausible Data Received From Forward Facing Camera.	Descriptions do not establish offending data, set-time, source interface, or cause.
Jeep ETM codes in September 17 memo [J-3].	B1498-19: USB Hub – Overcurrent; B1570-19: USB Communication – Overcurrent.	USB-related diagnostic context; no demonstrated Apple-originated input or downstream vehicle effect.

Record / provenance	Observed fields or report	Interpretive boundary
Jeep phone-video index.	Displayed times 15:26 and 15:34 CDT, September 16, 2026; eight-minute difference.	Interval between displayed phone timestamps, not established dashcam duration.

C.1 FCA watchdog excerpt: selected normalized fields

Field	Reported value
`captureTime`	`2026-09-15 23:39:09.4070 -0500`
`procLaunch`	`2026-09-15 23:38:48.7082 -0500`
`procName`	`fca_uconnect`
`bundleID`	`com.fca.myconnect.nafta`
`osVersion`	`iPhone OS 26.6.2 (23G90)`
`exception`	`EXC_CRASH / SIGKILL`
`termination.namespace`	`FRONTBOARD`
`WatchdogEvent`	`scene-create`
`code`	`0x8BADF00D`
`real-time allowance`	`19.35 seconds`
`elapsed total CPU`	`9.690 seconds`
`application CPU`	`0.114 seconds`
`Thermal Level / State`	`0 / nominal`

The displayed capture and launch timestamps differ by 20.6988 seconds; this interval is distinct from the 19.35-second watchdog allowance. The low application CPU figure motivates investigating waiting, initialization dependencies, or scheduling without selecting one explanation. [P-1]

D State-Transition Explanatory Model

The model formalizes the paper's account of independently managed representations; it is not a reconstruction of vendor-internal architecture. Let $A(t)$, $S(t)$, $P(t)$, and $D(t)$ denote application lifecycle, interface/session, presentation, and diagnostic representations, respectively. Each is an analytical variable inferred from a different observable record, not an instrumented shared state variable.

Layer	Observable proxy	Illustrative transition / divergence
A – application lifecycle	FCA FrontBoard scene-create watchdog	Foreground scene creation may fail or terminate while another layer's presentation is managed separately.
S – interface/session	Provisional CarPlay fault timestamps	A session may transition on its own timeline; its relationship to A or P is not established by the notes.
P – presentation	Messages picker/media-path behavior; Jeep display report	A selected or rendered representation may persist, transform, or differ across a context or source change.
D – diagnostics	Jeep DTC descriptions	A module may record a condition without exposing trigger time, originating message, or interface relationship.

D.1 Representation transition schema

For layers $i \in \{A, S, P, \text{diag}\}$, distinguish the *time-varying layer state* $X_i(t)$ from the artifact representation $\mathcal{R}_k(x)$ defined in Section 8.5. Here, A, S, and P retain their previously defined layer meanings, while diag denotes the diagnostic layer.

The diagnostic-layer state is written $X_{\text{diag}}(t)$ to distinguish it from the transformation-discrepancy function $D_{F,\phi}(x)$.

Conceptual local evolution is represented by

$$X_i(t + \Delta t) = \mathcal{L}_i(X_i(t), u_i(t), e_i(t); \Delta t), \tag{2}$$

where t denotes the reference time and $\Delta t > 0$ denotes the elapsed interval between modeled states.

The variable $u_i(t)$ represents inputs or control actions affecting layer i, while $e_i(t)$ represents events or environmental influences during the transition.

The operator $\mathcal{L}_i$ describes conceptual evolution of layer state over the specified interval. It is not the artifact transformation $F_{k\to\ell}$ defined in Section 8.5.

Inputs and events are modeled as summaries of influences over $[t, t + \Delta t]$ rather than instantaneous measurements sufficient to reconstruct an internal vendor implementation.

Neither the complete internal state nor the local transition rule is directly measured in the present vendor records.

Observable representations. Let $O_i(t)$ and $O_j(s)$ denote observable proxies for layers i and j, respectively, at their recorded observation times.

These proxies may represent diagnostic events, application states, presentation outcomes, or other declared measurements.

Under a specified operating condition, M_{ij} maps an observation from layer i into the expected observation domain of layer j.

The comparison function d_{ij} measures discrepancy between commensurate observations after applying this mapping.

Clock normalization. Observations from different layers may originate from independently maintained clocks.

Let c_i and c_j denote declared clock-normalization functions mapping recorded timestamps from layers i and j into a common temporal reference.

Clock normalization must account for known offsets and time-zone conventions. The normalization procedure, clock uncertainty, and any assumed clock-drift bounds must be declared before comparison.

Let T_j denote the set of retained observation timestamps for layer j.

For an observation at reference time t, define the admissible comparison set as

$$T_{ij}(t) = \{s \in T_j : |c_j(s) - c_i(t)| \le \tau_{ij}\}, \tag{3}$$

where $\tau_{ij} \ge 0$ is the declared maximum permitted temporal separation after clock normalization.

The temporal tolerance must be justified by the intended comparison and the documented synchronization uncertainty. It is not inferred from the observed discrepancy values.

Temporal pairing. The operational reconciliation test uses nearest-time pairing rather than an existential best match across the entire admissible window.

For a nonempty admissible comparison set, define the selected timestamp by

$$s_{ij}^{*}(t) = \operatorname*{arg\,min}_{s \in T_{ij}(t)} \left| c_j(s) - c_i(t) \right| . \tag{4}$$

If multiple observations have equal minimum temporal separation, the earliest normalized timestamp is selected. Any remaining duplicate-timestamp ambiguity must be resolved by a predeclared deterministic selection rule.

The operational reconciliation discrepancy is then

$$d_{ij}^{*}(t) = d_{ij}\Big(O_j\big(s_{ij}^{*}(t)\big), M_{ij}\big(O_i(t)\big)\Big) . \tag{5}$$

Temporal selection occurs before evaluating the observation discrepancy.

Consequently, an observation elsewhere in the admissible window cannot replace the nearest eligible observation solely because it produces a smaller discrepancy.

No interpolation or synthetic observation is assumed.

Operational classification. Before applying the reconciliation test, the investigator must specify the observation proxies and their units or categories, the expected mapping M_{ij}, the comparison function d_{ij}, the clock-normalization procedure, the temporal tolerance τ_{ij}, the sampling coverage, and an acceptance threshold ϵ_{ij}.

A *match* occurs when $d_{ij}^{*}(t) \leq \epsilon_{ij}$ under a validated mapping and adequate observation coverage.

A *candidate divergence* occurs when $d_{ij}^{*}(t) > \epsilon_{ij}$ under the same declared conditions.

A result is *indeterminate* when a required observation, temporal alignment, mapping, distance function, threshold, or sampling requirement is absent or insufficient.

In particular, an empty admissible comparison set does not produce a numerical discrepancy.

Clock uncertainty that prevents a defensible temporal pairing also produces an indeterminate result.

Interpretation and causal limits. A candidate divergence establishes a failure of the declared expected correspondence between observable layer states, not the underlying cause of that failure.

Nearest-time pairing provides an operational comparison rule but does not independently establish that the paired observations represent the same underlying event.

Causal attribution across vendor boundaries additionally requires intervention or source-to-receiver tracing.

The present Apple and Jeep records do not establish the synchronized observations and validated cross-system mappings necessary for numerical reconciliation.

Accordingly, no numerical $d_{ij}^{*}(t)$ value is assigned to those records.

D.2 Observable implications

The model accommodates (i) cross-conversation picker persistence, (ii) path-dependent media behavior, (iii) an application lifecycle termination, and (iv) separately recorded interface, display, and diagnostic views. It does not assert that these incidents share a defect. A cross-vendor causal claim additionally requires a defined transfer, identified receiving component, synchronized trace, and repeatable downstream effect. [P-1]

E Mechanism Discrimination Matrix

Five candidate mechanisms are compared against discriminating evidence. The scene-create watchdog directly identifies an application lifecycle termination. Vehicle-local and interface explanations require separate controls. [P-1]

Candidate	Existing support	Discriminating evidence
A. App lifecycle stall	FCA scene-create watchdog termination (recorded IPS fields); blocking operation unknown.	Original IPS, stacks, signposts, launch dependencies, app-only repeat trials.
B. Independent infotainment rendering	Jeep display complaint; independence from phone not demonstrated.	Head-unit traces, source switching, phone-disconnected baseline.
C. CarPlay session mismatch	Provisional timestamp pair; insufficient session metadata.	Original fault reports; synchronized connection/pairing and head-unit traces.
D. Vehicle module/network issue	Stored U0422-00, U148D-00, B1498-19, B1570-19 descriptions; source and timing unknown.	Full scan/freeze-frame, module topology, code timestamps, no-CarPlay controls.
E. Shared cross-vendor defect	Proposed source-to-receiver propagation; no demonstrated transfer or downstream causal effect.	Captured transfer, receiving parser/state, synchronized reproduction and negative control.

E.1 Explanatory utility versus causal attribution

The state-divergence model is useful for reasoning about why independently observed layers may disagree. That utility does not elevate candidate E above vehicle-local alternatives or convert provisional CarPlay timing into a synchronized chain. A negative test under incompletely specified conditions is not universal falsification; positive correlation without an identified transfer is not causal proof. [P-1]

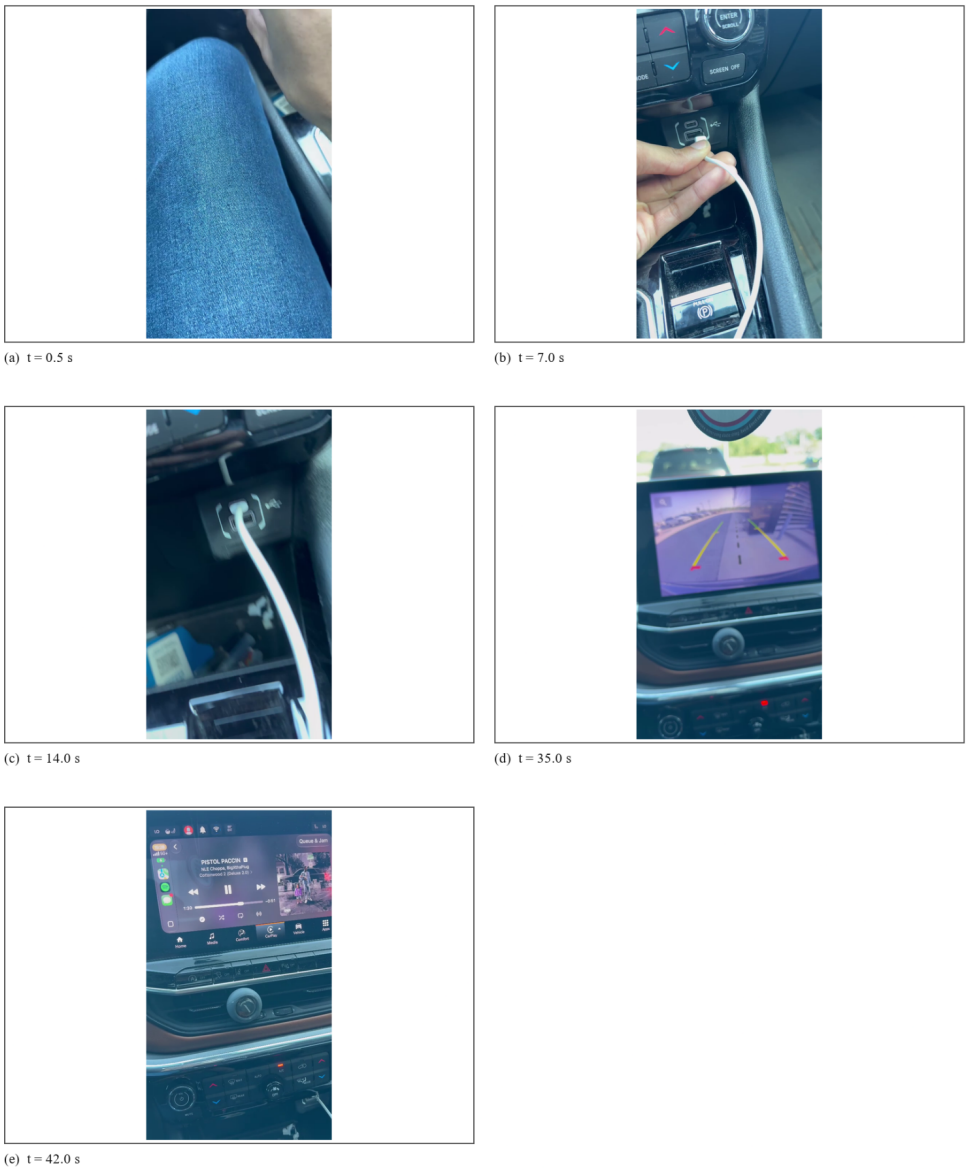
F Experimental Evidence Matrix

Artifact / experiment	Established proposition	Unresolved edge
`make_extremes.sh` and G2 JPEG	Extreme aspect ratios; baseline 4:4:4 retained header; SHA-256 identified.	Device-side decode linkage, precise path, and incident-specific outcome.
`wm_robust.py`	Repeated marker, DWT/DCT/SVD embedding and extraction attempts across in-memory, PNG, and JPEG representations.	Execution logs, extraction results, any G2 run linkage.

Artifact / experiment	Established proposition	Unresolved edge
`bug_test.py`	Subsequent G2 watermark probe using the retained `G2_444_RGB_baseline_4096x1.jpg` as its configured input.	The displayed excerpt establishes artifact paths; embedding/extraction calls and run-level results are not reproduced in that excerpt.
PixInsight project and inventory	Fourier/inverse-Fourier history and recorded watermark-contribution settings; retained named outputs and hashes.	Exact `Image12` operand/order; coefficient-to-output mapping; RGB/LRGB sequence; measured extraction after blending.
Fourier and PixelMath notes	Retained spectral workflow and separate SHO, star, and luminance expressions.	Which operations were executed on a particular watermarked output; no assumed shared pipeline.
Apple diagnostic records	Reported MTE, accessory, RTKit, and JPEG-related observations in distinct records.	Exact cross-device source/target lineage and incident-level fault linkage.
Restore comparison (runs 10–12)	Two modified-cache runs failed; the original-cache run succeeded.	Network equivalence across runs and causal isolation of the cache perturbation; relationship to JPEG/RTKit records.
Jeep/CarPlay documentary corpus	Independently documented interface, presentation, service, and diagnostic observations.	Any causal Apple-to-vehicle data transfer or shared defect.

The matrix distinguishes demonstrated test construction and reported restore interventions from diagnostic observations and untested transfer relationships.

G Documentary Figures



(a) t = 0.5 s

(b) t = 7.0 s

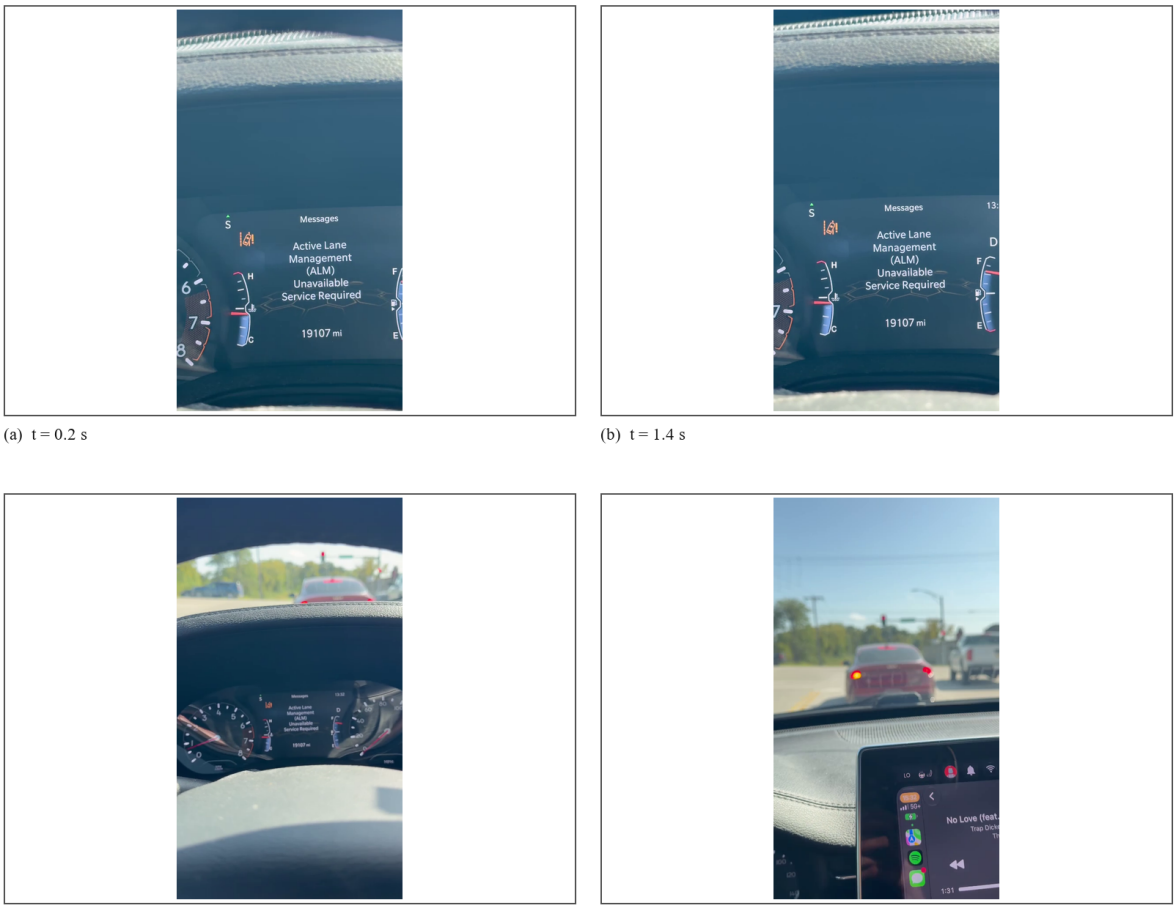
(c) t = 14.0 s

(d) t = 35.0 s

(e) t = 42.0 s

Source: sep-16-dealership-display.mp4; panel timestamps are elapsed video time.

Figure 1: Dealership reproduction. Representative frames from the September 16 recording documenting the USB connection, head-unit interaction, and CarPlay presentation.



(a) t = 0.2 s

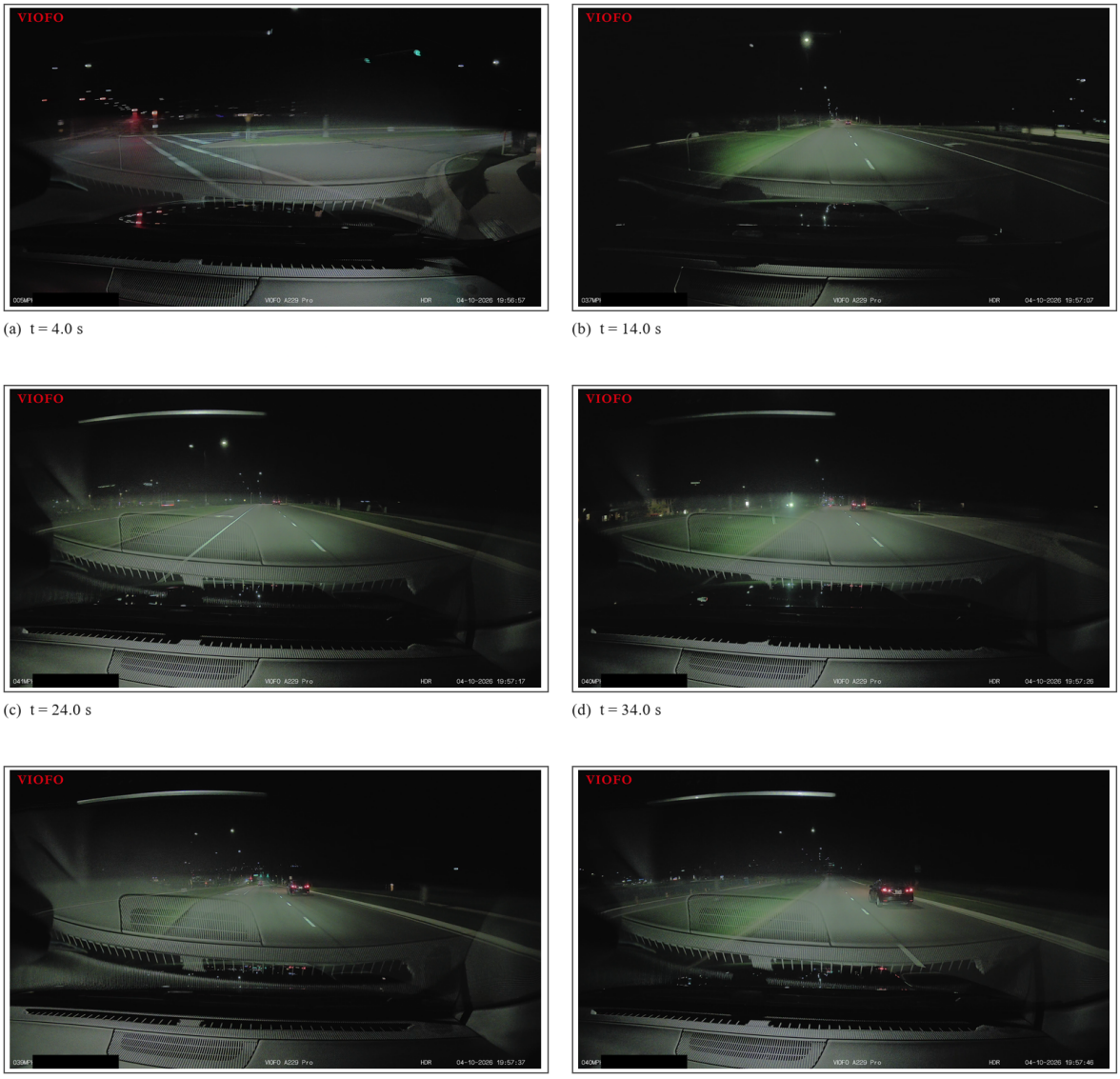
(b) t = 1.4 s

(c) t = 2.6 s

(d) t = 5.1 s

Source: IMG_4566.mp4; panel timestamps are elapsed video time.

Figure 2: Post-departure follow-up. Selected frames recorded after leaving the dealership, including the instrument-cluster warning: "Active Lane Management (ALM) Unavailable—Service Required."



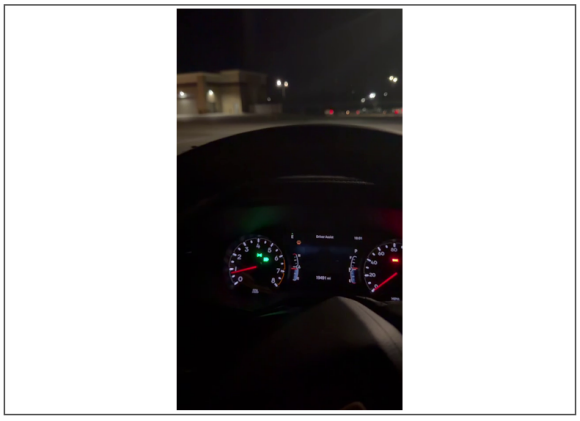
(a) t = 4.0 s (b) t = 14.0 s

(c) t = 24.0 s (d) t = 34.0 s

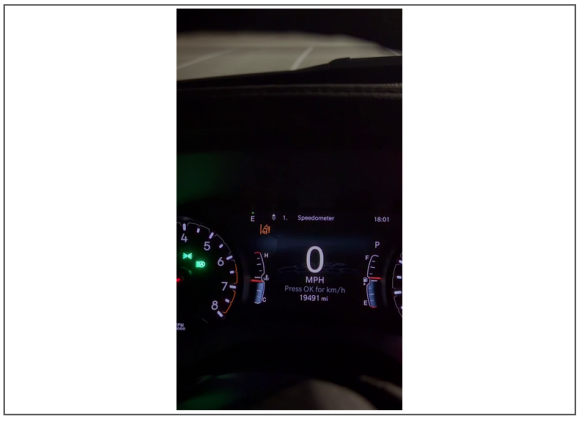
(e) t = 44.0 s (f) t = 54.0 s

Source: oct-04-dashcam-gps-removed.mp4; coordinate overlay concealed in stills; timestamps are elapsed video time.

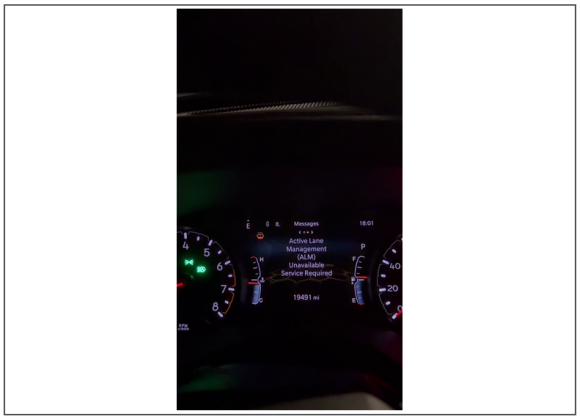
Figure 3: October 4 dashcam sequence. Representative frames from the 4K publication copy, preserving roadway context and elapsed recording time. Coordinate overlays are concealed.



(a) t = 1.0 s



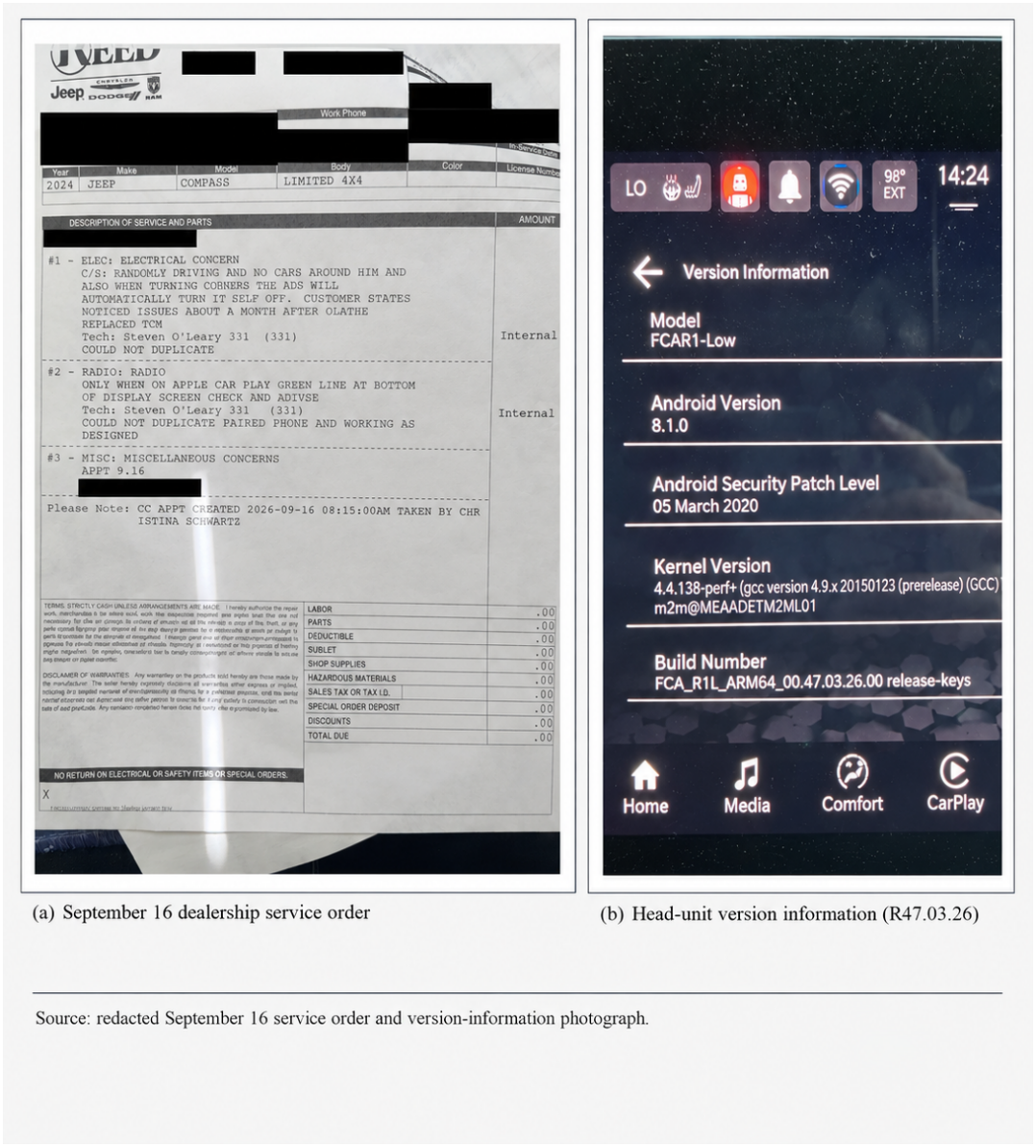
(b) t = 16.0 s



(c) t = 22.0 s

Source: IMG_5189.mp4; panel timestamps are elapsed video time.

Figure 4: October 4 phone follow-up. Selected instrument-cluster states from the recorded vehicle observations, including the Active Lane Management warning.



(a) September 16 dealership service order

(b) Head-unit version information (R47.03.26)

Source: redacted September 16 service order and version-information photograph.

Figure 5: Firmware and service documentation. (a) Redacted September 16 dealership service order identifying the 2024 Jeep Compass and recording the reported CarPlay display complaint. (b) Head-unit version information showing model FCAR1-Low, Android 8.1.0, and build `FCA_R1L_ARM64_00.47.03.26.00 release-keys (R47.03.26)`.