P W D A T A S Y S T E M S v 2 . 3 x

Forensic Reconstruction Platform

Subsystem Architecture and Tri-Axis Visual Grammar Specification

---

Technical Assessment and Reference

Document classification: Operational reference

Prepared: April 2026

1. Executive Summary

This document provides a comprehensive technical assessment of the chrono-core subsystem within the PW Data Systems forensic reconstruction platform, hosted at parrotwisdom.co.uk. The assessment covers the subsystem's architecture, data lifecycle, navigation engine, and the three-tier analytical framework that constitutes its intellectual core: the Institutional Chain Marker Taxonomy, the six-stage propagation lifecycle, and the tri-axis visual grammar (CHIP/ID system) that provides real-time structural visibility into active institutional failure chains.

The subsystem is not a retrospective analytical tool. It is a live detection and tracking instrument. Its elapsed-time counter, seeded at 4 November 2005, marks the origin of an active chain that has been tracked forward through institutional machinery in real time for over twenty years. The platform's architecture is engineered to make that chain's propagation visible, navigable, and structurally legible to a reviewer encountering the material for the first time.

2. System Architecture

2.1 Technology Stack

The platform is built on vanilla JavaScript with no framework dependencies, no build pipeline, and no package management. It runs entirely from a static file directory. This is a deliberate architectural decision: the system must function offline, remain portable, and resist dependency on infrastructure that could become unavailable. The technology choices are:

- **Runtime:** Browser-native JavaScript (ES5/ES6 mixed), no transpilation.
- **Data bus:** localStorage as the sole persistence and inter-module communication layer.
- **View layer:** DOM construction via document.createElement, with cloning for view multiplexing.
- **Navigation:** Hash-based routing with delegated event interception for internal links.
- **Styling:** Four CSS files using CSS custom properties for the document-type colour vocabulary.
- **Data portability:** Export/import cycle via JS backup files wrapping JSON payloads in var assignments.

2.2 File Inventory

The subsystem comprises thirteen files across three directories:

File	Type	Role
chronology.html	HTML shell	Primary chronology interface
search.html	HTML shell	Chronology search hub
appendices.html	HTML shell	Appendices viewer
append-search.html	HTML shell	Appendices search

File	Type	Role
chrono-core.js	JS module	Tab/pane system, marker taxonomy, chip renderer
chrono-jump-builder.js	JS module	Navigation engine, index builder, hash router
legend.js	JS module	Flyout panel management (chronology)
search-legend.js	JS module	Panel management + chain guide (search)
chrono-ancillary.js	JS module	Data export utility
chrono-common.css	Stylesheet	Root variables, document-type colour coding
chrono-shell.css	Stylesheet	Page layout, tab bar, flash-highlight animation
chrono-search.css	Stylesheet	Search-specific layout
pw-drawers.css	Stylesheet	Flyout panel geometry and transitions

2.3 Data Lifecycle

The chronology data follows a controlled lifecycle with a single canonical source:

2.3.1 Ingestion

On page load, chronology.html executes a three-step sequence. First, it removes any existing chronology-state entry from localStorage. Second, it loads the chronology-backup.js script, which declares a global chronologyBackup variable containing a timestamped JSON payload of all chronology entries. Third, it writes that payload into localStorage under the key chronology-state. This delete-on-load pattern ensures the backup file is always canonical. The browser state is transient.

2.3.2 Construction

The jump builder reads entries from localStorage, sorts them by dateKey, and constructs a temporary in-memory list with year-break separators. It then walks every list item, assigning deterministic IDs based on date strings, extracting document-type metadata from element class names, and building three parallel indexes: chronoIndex (dated entries with marker glyphs), chronoDocs (linked evidence items per entry), and chronoJumps (cross-reference links with chain metadata). These indexes are written to localStorage under the key chrono-indexing-data.

2.3.3 Export

chrono-ancillary.js provides an exportDom() function that serialises the current localStorage state back into a downloadable chronology-backup.js file. The JSON is wrapped in a var chronologyBackup = ...; assignment, completing the round trip: file to localStorage to runtime DOM to file.

2.3.4 Staleness Detection

A 24-hour heartbeat runs hourly, checking the saved timestamp in chronology-state against the current time. If the data is older than 24 hours, it attempts to load a fresh backup script. If that script carries a newer timestamp, it clears localStorage and reloads the page. This handles the case where a browser tab remains open for extended periods without manual refresh.

3. Navigation Engine

3.1 Jump Menu Construction

The jump builder constructs a scrollable navigation panel from the chronology entries. Each entry appears with a document-type icon, colour-coded by class (custody, crime report, CAD record, evidence, oversight, office, open-source, adjudicated), and truncated to eight words for readability. The menu supports both HTML select elements and div-based scrollable lists, adapting to whichever element the host page provides.

Evidence items nested within chronology entries appear as indented sub-items with arrow prefixes, preserving the parent-child relationship between a dated event and its supporting documentation.

3.2 Tab and Pane System

chrono-core.js provides two viewing modes. In contiguous view (the default), all entries appear in a single scrollable timeline with year headers. In tabbed view, entries are partitioned by year into separate panes with a tab bar. A toggle button switches between modes.

The system creates clones of source elements rather than moving them, stripping the id attribute from each clone and preserving the original in a data-source-id attribute. This allows the navigation engine to resolve a source ID to a clone in whichever pane is currently active, without collisions between the original DOM and the constructed views.

3.3 Hash Routing

Two independent hash-navigation handlers serve distinct purposes:

- **Persistent highlight (IIFE1):** Handles internal navigation via the jump menu and in-page clicks. Resolves the target through ChronoTabs, scrolls it into view, and applies a flash-highlight class that persists until the next navigation event. Also closes the target panel after scrolling.
- **Flash highlight (IIFE2):** Handles incoming external hash navigation and browser hashchange events. Same resolution logic, but the highlight self-clears after 4000 milliseconds. This is the entry point for links arriving from search results, or external references.

A date format normaliser (resolveChronoId) converts ISO-format date hashes (date-YYYY-MM-DD) to the platform's internal format (date-DD-MM-YYYY), ensuring inbound links resolve correctly regardless of the linking page's date convention.

Internal chronology links carrying target="chronology" are intercepted by a delegated click handler that prevents page reload and routes through hashchange instead, keeping navigation within the single-page architecture.

3.4 Flyout Panel System

Three slide-out panels provide reference and navigation tools: the target panel (jump menu and document-type legend), the marker panel (institutional chain marker taxonomy), and the guide panel (CHIP/ID grammar reference with sub-sections). The panels are mutually exclusive: opening one closes the others. A delegated click listener on the document body closes all panels when the user clicks outside.

4. The Four-Tier Analytical Framework

The platform's analytical architecture operates on four hierarchical tiers. Each tier depends on the one above it for its structural meaning.

4.1 Tier 0: Live Detection Instrument

The platform is not a retrospective analytical tool applied to a completed sequence. It is a live detection instrument that identified an institutional failure chain at its ignition point and has tracked it forward in real time as events occurred. The elapsed-time counter, seeded at 4 November 2005 and ticking in the chronology header, is a running clock on an active chain.

The ignition point marker — designated ◆ L — identifies the single chain that succeeded in permanently embedding itself into state machinery without being caught at the point of entry. Other ignition points exist throughout the continuum, but those chains were brought to a halt by local conditions: witnesses, knowledge of the individuals involved, or contact with verifiable reality. The ◆ L chain entered the system at a level where no such local knowledge was present to challenge it. It was a clean injection into a system that does not self-correct.

Twenty years later, that marker has not been removed. Every subsequent institution has read from it and acted on it without verification. The platform documents this propagation in real time.

4.2 Tier 1: The General Model

The platform asserts that institutional failure chains are structurally predictable. They are not unique events requiring bespoke explanation. They follow a recurring architecture that can be identified, classified, and tracked using a formal grammar.

This assertion is validated against a public corpus of five independently documented UK institutional failure chains: Post Office Horizon, Hillsborough, Grenfell Tower, Windrush, and Mid Staffordshire NHS. Each of these chains, spanning different decades, different institutional domains, and different actors, conforms to the same propagation sequence. The platform's own chronology is presented as the sixth instance.

4.3 Tier 2: The Six-Stage Propagation Lifecycle

The second tier defines the sequence through which institutional failure chains propagate:

Stage	Name	Function
1	Origin	The chain is seeded by a condition, event, or actor. Often appears minor at the time.
2	Distortion	The record is altered, deflected, captured, suppressed, or jurisdictionally displaced.
3	Linkage	The chain connects across institutional domains, actors recur, or independent chains converge.
4	Structural	The chain acquires structural characteristics: pivots, recycled patterns, containment, temporal compression, or drift.
5	Suppression	Oversight fails, regulators are captured, escalation is blocked, or whistleblowers face retaliation.
6	Outcome	The chain reaches a terminal state: partial accountability, systemic acknowledgement, or chain termination.

These stages are not merely descriptive categories. They are a propagation grammar: a predictive sequence in which the output of each stage becomes the input condition for the next. A chain that has reached the Structural stage will exhibit Suppression characteristics. A chain in Suppression will produce Outcome events. The sequence is directional.

4.4 Tier 3: The 23 Institutional Chain Markers

The third tier provides the node-level classification vocabulary. Twenty-three markers, distributed across the six lifecycle stages, allow each event in the chronology to be classified by its function within the propagation sequence. Each marker has a Unicode symbol for visual scanning, a formal alphanumeric code, a definition written as a universal (not case-specific), and public-corpus examples demonstrating the marker in at least two independent chains.

4.4.1 Origin Markers (OR1–OR3)

Sym	Code	Name	Definition
✱	OR1	Seed event	The initiating condition from which the chain proceeds. Often appears minor at the time. Identifiable only in retrospect as the structural origin point.
◎	OR2	Predisposing condition	Pre-existing structural vulnerability that does not itself initiate the chain but determines how the seed event propagates.
⊕	OR3	Trigger event	An external event that activates a dormant predisposing condition or accelerates an existing chain. Not the origin — the accelerant.

4.4.2 Distortion Markers (DI1–DI5)

Sym	Code	Name	Definition
⬬	DI1	Record distortion	Active alteration, omission, or misrepresentation of the documentary record. May be deliberate or systemic.
⊘	DI2	Institutional deflection	Redirection of complaint or challenge away from substantive engagement. The system routes around the problem rather than addressing it.
≋	DI3	Narrative capture	The complained-of characterisation becomes the established frame through which subsequent events are interpreted. The distortion becomes load-bearing.
⌀	DI4	Evidence suppression	Active withholding of material that would alter the trajectory. Distinguished from record distortion by absence rather than alteration.
⥊	DI5	Jurisdictional displacement	Transfer of a matter to a body without jurisdiction, authority, or access to material necessary to resolve it. Appearance of process without substance.

4.4.3 Linkage Markers (LK1–LK4)

Sym	Code	Name	Definition
⇆	LK1	Bi-domain link	A single event or actor connecting two institutional domains, causing patterns in one to affect the other.
◈	LK2	Tri-domain link	A node connecting three or more institutional domains simultaneously. High structural significance.
⟳	LK3	Actor recurrence	The same institutional actor appears at multiple nodes in the chain across time. May indicate systemic role or deliberate pattern.
⋈	LK4	Network convergence	Multiple independent chains converging on a shared institutional node. The node is a structural concentration point.

4.4.4 Structural Markers (ST1–ST5)

Sym	Code	Name	Definition
◆	ST1	Pivot	A node at which the chain’s trajectory changes direction. May mark the moment a cover-up becomes active or a legal threshold is crossed.
⤤	ST2	Recycled pattern	A precursor sequence that has appeared before, now re-expressing. The system is running the same play again.
▣	ST3	Containment structure	A formal mechanism whose function is to absorb pressure without producing systemic change. Distinguished from genuine inquiry by outcome.
⧖	ST4	Temporal compression	A period in which multiple distortion or suppression

Sym	Code	Name	Definition
			events cluster unusually tightly. Active crisis management rather than systemic drift.
∿	ST5	Drift period	Extended period of low-marker-density in which predisposing conditions accumulate without acute events. The chain progresses invisibly.

4.4.5 Suppression Markers (SU1–SU4)

Sym	Code	Name	Definition
⊟	SU1	Oversight failure	A body with formal oversight responsibility fails to identify or act on a known pattern. May be structural or active.
⊡	SU2	Regulatory capture	The regulating body's decision-making becomes structurally aligned with the interests of the body it regulates.
↯	SU3	Escalation block	A point at which information that should have propagated upward through an institutional hierarchy was stopped.
⊛	SU4	Institutional retaliation	Action taken against an individual who has identified or publicised the pattern. Distinguishable from legitimate process by timing and disproportionality.

4.4.6 Outcome Markers (OU1–OU3)

Sym	Code	Name	Definition
◉	OU1	Partial accountability	A formal finding that acknowledges some element of the chain without addressing its structural origin or full extent. Closure without resolution.
⊞	OU2	Systemic acknowledgement	Formal recognition that the failure was structural rather than individual. Rare. Prerequisite for genuine remediation.
⊠	OU3	Chain termination	The chain reaches a documented end-state, whether through resolution, exhaustion, or external intervention. May be genuine or nominal.

5. The Tri-Axis Visual Grammar (CHIP/ID System)

The visual grammar is the platform's most distinctive analytical layer. It operates on a different axis from the marker taxonomy and provides a fundamentally different kind of information. Where the markers answer "what is the state of this node?", the CHIP/ID system answers three simultaneous questions about the structural provenance of each cross-reference link in the chronology.

5.1 The Three Axes

Each chip displayed on a chronology cross-reference link encodes three values in a compact visual notation: [colour | number | letter].

5.1.1 Colour: Domain at the Jump Target

The colour of the chip identifies the institutional domain at the destination of the cross-reference link. It tells the reviewer what kind of terrain they are about to enter.

Colour	Domain	Scope
	Origin	The jump target sits at an initiating condition of the chain.
	Process	The jump target sits within a procedural, legal, or administrative mechanism.
	Record	The jump target involves the state of a primary document or evidence block.
	Institution	The jump target involves an agency's posture toward the information it received.
	Suppression	The jump target involves a method used to prevent correction of the record.
	Consequence	The jump target is a downstream outcome produced by the preceding chain.

5.1.2 Number: Class Within the Domain

The number on the chip identifies the specific chain class within that domain. Each domain contains multiple numbered chains, each representing a distinct sequence of related events.

Domain	Class	Chain Label
Origin	**1**	PRS Mechanism Seed Sequence
	2	Medical Record Origin Sequence
	3	Public Domain Identity Sequence
Process	**1**	Civil Process Fraud Sequence
	2	Criminal Process Sequence
	3	Legal Representation Denial Sequence
Record	**1**	Record Falsification Sequence
	2	Data Breach Sequence
Institution	**1**	Sharples-Benton Actor Recurrence
	2	Multi-Agency Adoption Sequence
Suppression	**1**	Narrative Suppression Sequence
	2	Oversight Deflection Sequence
Consequence	**1**	Displacement and Loss Sequence

Domain	Class	Chain Label
	2	Malicious Prosecution Sequence

5.1.3 Letter: The Originating Actor (Seed Constant)

The letter on the chip identifies the originating actor who seeded the chain. This is the constant: it never changes as a chain propagates through different domains and classes. If you follow any jump backwards, the letter remains the same. Different chains carry different letters. When multiple letters appear at the same node, independent chains — seeded by different actors — converge at that event.

The letter is the provenance axis. It answers: "Regardless of which domain this chain is currently passing through, and regardless of which class within that domain it occupies, who started it?"

5.2 Reading the Chips

A chip is read as a declaration about the link itself, not a forecast about its destination. It identifies the cross-reference as a node in a specific chain. When reading a chip such as [process|2|C], the logic is:

- **Colour (process):** The jump target sits in the procedural domain.
- **Number (2):** The target is a node in the Criminal Process Sequence.
- **Letter (C):** The chain was seeded by a specific originating actor thread.

Two chips of the same colour on a single link indicate two chains within the same domain converging at one node. Two chips of different colours indicate a cross-domain convergence: two institutional terrains meeting at a single event. Both readings expose structural relationships that are invisible in the chronological text alone.

5.3 The Ignition Point Marker: ◆ L

The ignition point marker is a special designation within the seed-letter system. It marks a chain that was identified at its origin and tracked forward through the institutional machinery in real time. The ◆ glyph appears as a banner at the top of year-tab panes whose entries contain it, providing immediate visual identification that events within that year are part of the live-tracked chain.

The distinction between ◆ L and other ignition points in the continuum is not that it was the only one. It is that it was the only one that succeeded in permanently embedding itself into state systems without being intercepted at the point of entry. Other ignition points were caught by local conditions: witnesses who knew the individuals, contact with verifiable reality, or institutional actors who checked the record before acting. The ◆ L chain entered a system that does not self-correct. It has been propagating ever since.

5.4 Relationship Between the Grammar Axes

The three axes of the CHIP/ID system operate independently:

- **Markers** (Tier 3) answer: "What is the state of this node?" They describe the local condition — distortion, actor recurrence, recycled pattern, oversight failure.

- **Chips** (colour + number) answer: "What arrived here, through what domain?" They describe the structural position of the cross-reference within the chain network.
- **Seed letters** answer: "Who started it?" They provide provenance that persists regardless of how far the chain has propagated or how many domains it has crossed.

This separation is the analytical core of the platform. A single chronology event can simultaneously be classified as a Narrative Capture node (DI3) on the marker axis, sitting in the Process domain class 1 on the chip axis, and carrying seed letter L on the provenance axis. Each axis gives the reviewer a different kind of information about the same event. Together, they make the structural architecture of the chain visible at every point of contact.

6. Document-Type Colour Vocabulary

Independently of the chain/domain colour system, the platform uses a separate colour vocabulary for document types. Each evidence item in the chronology is classified by its institutional origin and rendered with a colour-coded left border. The vocabulary is defined in CSS custom properties and applied via class names:

Class	Icon	Colour	Document Type
custody	⚿	#FF1717	Custody record
report	🖿	#fb923c	Crime report
cad	🖚	#0682BD	CAD record (Computer Aided Dispatch)
evidence	⚖	#450578	Evidence item
oversight	🛠	#380696	Oversight body document
office	♔	#3B0675	Official/office document
judged	□	#F527E0	Adjudicated item
open	🗺	#3B0675	Open-source material
claim		#FF5C3D	Claim document
append		#0C961E	Appendix cross-reference

7. Search Subsystem

The search pages (search.html and append-search.html) provide full-text search across the chronology and appendices corpora respectively. search.html loads chronology entries from localStorage into a hidden shadow list, then performs regex-based matching against the text content of each list item. Results link back to the chronology via hash navigation, enabling the reviewer to move directly from a search hit to the corresponding event in context.

The search variant loads search-legend.js instead of the standard legend.js and chrono-core.js combination, providing an integrated panel that includes both the marker taxonomy and the chain domain guide. The search-legend variant adds a "Chains" default tab to the marker panel, displaying the full domain-chain mapping and chip-reading instructions as the first view the reviewer encounters.

8. Structural Observations

8.1 Deliberate Architectural Choices

The platform makes several architectural choices that are non-obvious but purposeful:

- **No framework dependency.** The system must survive independently of any external infrastructure. No CDN, no npm registry, no build server. It runs from a folder.
- **localStorage as data bus.** Decouples the data source (backup file) from the rendering engine (jump builder) without requiring a server. The bus is the browser. The canonical source is always the file.
- **DOM cloning with ID stripping.** Allows the same source data to appear in multiple views (contiguous, tabbed, search results) without ID collisions, while maintaining navigability via data attributes.
- **Two independent hash handlers.** Persistent highlight for internal navigation (reviewer stays oriented) and flash highlight for external arrival (reviewer locates the entry and moves on). Deliberately separate IIFEs, not a shared handler with a flag.
- **Delete-on-load for localStorage.** The backup file is always canonical. The browser state is always transient. This prevents stale data from accumulating across sessions and eliminates an entire class of synchronisation bugs.

8.2 The Grammar as Operational Constraint

The tri-axis visual grammar is not a design space to be interpreted or extended. It is an operational constraint. The colour is the domain. The number is the class. The letter is the seed. These are fixed readings, not metaphors. The grammar's value is that it cannot be misread once understood: a chip declares a structural relationship that is either present in the data or it is not. There is no interpretive latitude.

This is the platform's strongest design principle. The visual grammar is a formal language, not a visualisation style. It constrains the reviewer's reading to structural facts about chain provenance, domain crossing, and actor persistence. It does not suggest, imply, or illustrate. It declares.