# The Chain of Intent

*Tracing Execution Across Humans and Agents.*

*GrayBrain Labs · hello@graybrain.dev*

GPL-TR-2026-02 · 9 July 2026

**Abstract.** As automated agents begin to act inside human interfaces, services increasingly want to answer two different questions from one interaction trace: *how* was this action executed, and *was it authorized*. We argue these are categorically distinct, and we separate them empirically and formally. First, we give an agency ontology that decomposes "human vs. bot" into six axes (executor, actuation, assistance, goal origin, supervision, and authorization) and show that only the execution-facing axes are candidates for trace-based inference. Second, using a public corpus of real human mouse dynamics (the Balabit Mouse Dynamics Challenge) and six openly specified automation generators spanning a naive-to-humanized spectrum, we show that a transparent human-vs-automation detector separates classes almost perfectly in-distribution (session AUC 0.98), that this performance is essentially attributable to implementation artifacts (an artifact-only detector retains 99.3% of the full AUC), and that it transfers poorly: detectors built on timing/precision artifacts *anti-transfer* across automation families (off-diagonal AUC as low as 0.00), and no feature regime detects a family that replays recorded human motion (leave-one-family-out AUC 0.44–0.60). Third, on the same real traces we show that a strong automation detector is *blind* to human sessions with an account/operator identity mismatch; it flags 0 of 685 labelled sessions and orders mismatch versus owner sessions weakly inversely (AUC 0.42, 95% CI $[0.375, 0.460]$), while a per-user identity detector only *partially* separates them (AUC 0.86) and still does not observe authorization. Fourth, we prove that authorization is not identifiable from the trace: whenever authorization can vary while execution is held fixed, there exist observationally identical traces with opposite authorization, so any trace-only authorization test is at chance on that fiber. We close with a delegation-evidence requirements framework and a coverage analysis against public standards, arguing that trustworthy delegation needs provenance infrastructure, not better behavioral classifiers.

**Keywords:** agent detection; delegation; provenance; behavioral biometrics; observational equivalence; authorization; mouse dynamics

---

## 1 Introduction

The operative security question on the web is shifting. For two decades the dominant framing was *human versus bot*: is there a person behind this request, or a script? That framing is becoming obsolete. A growing share of legitimate activity is executed by software acting for a person: autonomous agents, human-authored plans run by agents, agent-proposed actions approved by humans, shared-control sessions, accessibility tooling, and remote humans whose input is actuated through automation-like channels. In this world the interesting questions are no longer binary, and they are not all the same question: *who or what executed the action, who chose the goal, who authorized the scope, who supervised the process, which tool actuated the interface, and was the action permitted?* A single classifier that outputs "bot" collapses all of these.

This report makes a deliberately narrow and, we think, clarifying claim. Interaction traces can carry real information about the *mode of execution*: the motor and actuation signature of how bytes reached the interface. They do not provide a universal way to determine or certify *authorization*: whether the principal permitted this action, within what scope, under what supervision. Deployment-specific statistical information may nevertheless

exist when authorization is correlated with execution. The first is a measurement problem with contingent, implementation-dependent answers. The second is not a measurement problem at all: it is a fact about provenance and policy that is simply not a function of the trace. Conflating the two produces two characteristic failures: treating a detectable automation signature as evidence of wrongdoing, and treating an undetectable one as evidence of legitimacy.

We support this claim along four lines.

1. **An agency ontology** (Section 2) that separates executor, actuation, assistance, goal origin, supervision, and authorization, and marks which axes are even candidates for trace-based inference.
2. **An empirical artifact-dependence and cross-family transfer study** (Section 5, Section 6) built entirely from public human data and openly specified automation generators. We find that strong in-distribution automation detection is dominated by implementation artifacts and does not transfer: artifact-based detectors anti-transfer across families, and a mimicry family that reuses human motion is undetectable by any regime under family holdout.
3. **A demonstration of authorization blindness on real traces** (Section 7): an automation detector is blind to human sessions with an account/operator identity mismatch (Balabit owner vs. `is_illegal` labels), while a per-user identity detector partially separates them; crucially, *neither observes authorization*.
4. **A formal observational-equivalence result** (Section 8) showing that authorization is not identifiable from the trace, distinguishing this epistemic limit from a computational one, and bounding what any trace-only test can do.

We then translate the negative result into a positive one: a delegation-evidence requirements framework and a coverage analysis against public standards (Section 9). The message is constructive. If legitimacy is not in the trace, it must travel *with* the action as verifiable provenance. Behavior is for anomaly detection; provenance is for authorization.

All code, generators, cached results, and figures are deterministic (seed 42) and reproducible from the repository; no proprietary detector, feature list, trace, or attack result from any GrayBrain system enters the analysis.

## 2 An agency ontology

The phrase “human vs. bot” names a one-dimensional projection of a higher-dimensional state. We find it useful to decompose an interaction into the axes in Figure 1 and Table 1. The axes are not independent, but they *dissociate*: real configurations populate combinations that the binary framing cannot name (an autonomous agent executing a human-authored, human-authorized plan; a human physically executing an action their employer has not authorized; a remote human whose motion is actuated through a scripted channel).

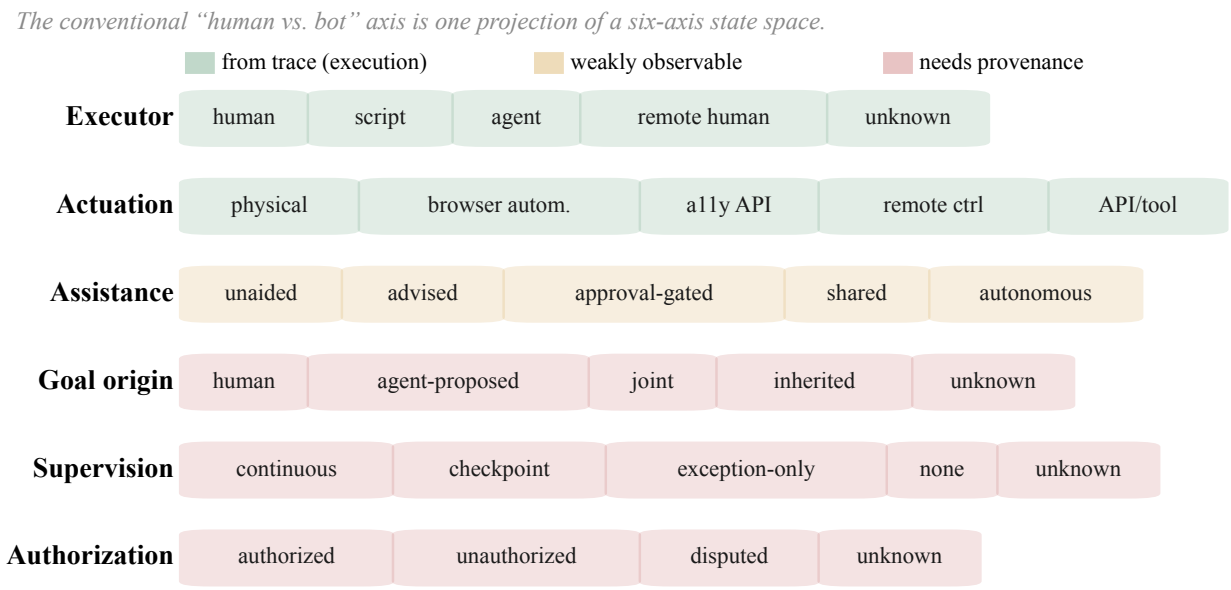



Figure 1: The agency and delegation state space. The conventional human/bot axis is a projection of six axes. Green axes (executor, actuation) leave a signature in the trace; amber (assistance) is weakly and partially observable; red (goal origin, supervision, authorization) are provenance variables that are not functions of the trace. "Unknown" is a first-class value on every axis.

The ontology enforces one discipline: *use only source-provided provenance, and do not infer missing labels.* Behavior may support a guess about the executor or the actuation channel. It does not license a claim about supervision or authorization, which are properties of the delegation relationship, not of the motion. The empirical sections quantify how far the observable axes can be pushed; the formal section shows why the provenance axes cannot be reached at all.

| Axis | Example values | Trace-identifiable? |
|---|---|---|
| Executor | human · script · agent · remote human · unknown | partly |
| Actuation | physical · browser automation · a11y API · remote control · API/tool | partly |
| Assistance | unaided · advised · approval-gated · shared · autonomous | weakly |
| Goal origin | human-defined · agent-proposed · joint · inherited | no |
| Supervision | continuous · checkpoint · exception-only · none | no |
| Authorization | authorized · unauthorized · disputed · unknown | no |

Table 1: The six agency axes and whether each is a candidate for inference from an interaction trace. "Partly" means detectable in-distribution but implementation-dependent and poorly transferable (Section 5, Section 6); "no" means not a function of the trace (Section 8).

## 3 Observation model and the observable/provenance split

We fix notation used throughout. A *world* $w$ comprises a principal $p$, an authorization/scope state, a goal origin, a supervision mode, an executor $e$, an actuation channel $c$, an environment/task $\eta$, and a realized motor/timing signal $u$. A destination service does not see $w$; it sees an *observable trace* $O(w)$ (here, a stream of pointer samples, buttons, and timestamps). Figure 2 draws the dependencies. The trace is produced by the *execution* subsystem: the executor and actuation channel, shaped by the environment, emit $u$, which the capture channel records as $O$. The provenance variables (principal, authorization, goal origin, and supervision) sit *upstream* and influence $O$, if at all, only *through* the executor's choices. This structural fact, made precise in Section 8, is the reason

authorization is unidentifiable: two worlds that differ only in provenance can drive the executor to emit the same $u$, hence the same $O$.

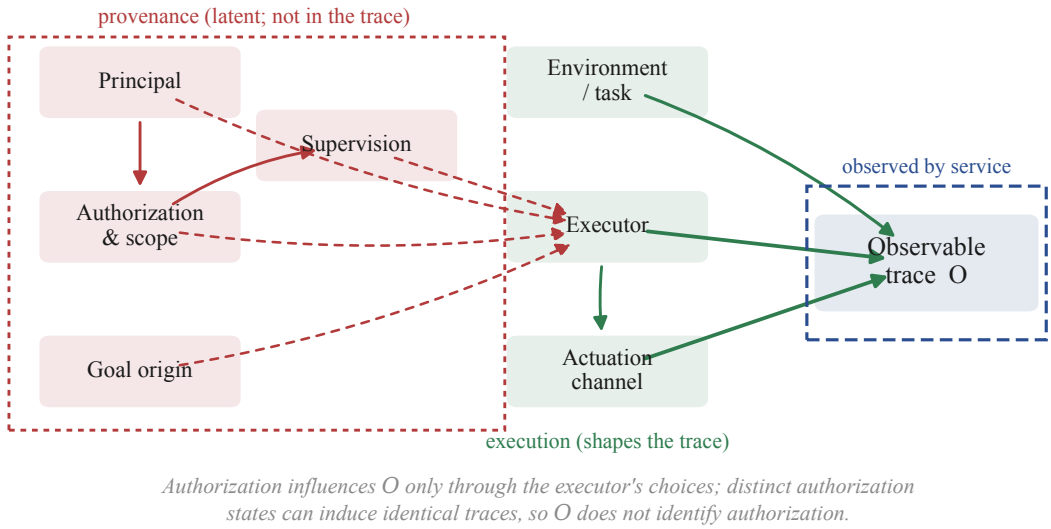



Figure 2: Observable versus provenance. Execution variables (green) generate the trace $O$; provenance variables (red) are latent and reach $O$ only through the executor. A service observes only the dashed box. Authorization is not an edge into $O$, so distinct authorization states can induce identical traces.

This picture also delimits our own claims. It does *not* say the trace is uninformative: the green edges are real, and Section 5 shows they can be read with high in-distribution accuracy. It says the informative content concerns *execution*, and that even there the signal is implementation-specific (Section 6). The red variables are a different kind of thing.

## 4 Data and openly specified automation generators

### 4.1 Human traces: the Balabit corpus

We use the public Balabit Mouse Dynamics Challenge data set [1], which records mouse activity from ten user accounts during remote-desktop (RDP) sessions. The training split contains 65 sessions (5–7 per account) known to be executed by the legitimate owners. The public test split contains sessions recorded against an account whose true operator is unknown and labelled `is_illegal`; those sessions are genuine human motion from a *different* person injected into the account, so *every* test session (owner or mismatch) is human-executed. This is exactly the structure we need for Section 7: it lets us ask what an automation detector and an identity detector do when the observed operator differs from the account owner. It does not observe an authorization violation.

We read the client-clock timestamps and pixel coordinates, drop the rare uint16 sentinel `65535` (an out-of-range/no-position marker; a handful of rows per session, after which coordinates lie within real screen extents), and never compute kinematics across the long idle gaps that punctuate RDP sessions.

### 4.2 Automation traces: six openly specified generators

Because no license-clean, richly-labelled public corpus of *automation* mouse traces in this modality exists, we generate automation with six transparent, seeded programs that emit the same event schema as Balabit and

pass through the identical feature extractor. They span a deliberate *naive-to-humanized* gradient (Table 2); full specifications are in Appendix B.

- **naive**: `linear` (constant-velocity straight interpolation on a fixed timing tick, a WebDriver-style `moveTo`) and `grid` (axis-aligned Manhattan routing). Continuous, perfectly regular in time and space.
- **humanized**: `bezier` (cubic-Bézier path with ease-in-out and jitter, a "humanized cursor" library), `minjerk` (the Flash–Hogan minimum-jerk trajectory [2]), and `windmouse` (the WindMouse gravity+wind algorithm). Curved, smooth, with jittered timing and inter-action pauses.
- **mimicry**: `replay`, an automation that *reuses recorded human motion*. It stitches contiguous full-event blocks of real human sessions (preserving their timing, drags, scrolls, and pauses) at fresh integer screen offsets. Within-block behavior is human by construction; only the block sequence and placement are generated. `replay` is a positive control for the limiting case of humanization and marks the behavioral-detection floor.

| Source | Sessions | Windows | Class | Notes |
|---|---|---|---|---|
| Balabit human (train) | 65 | 1261 | human | 10 accounts, genuine owners |
| `linear` | 60 | 891 | naive | fixed-tick straight interpolation |
| `grid` | 60 | 944 | naive | axis-aligned Manhattan routing |
| `bezier` | 60 | 712 | humanized | cubic Bézier + jitter |
| `minjerk` | 60 | 695 | humanized | Flash–Hogan minimum jerk |
| `windmouse` | 60 | 948 | humanized | gravity + wind path |
| `replay` | 60 | 732 | mimicry | stitched real human motion |
| Balabit test (labelled) | 685 | 1253 | human | 366 legal · 319 illegal |

Table 2: Data used in the study. A *window* is 800 consecutive move-samples (capped at 20/session); each window yields one 23-feature vector. Test sessions are used only in Section 7 and never train any detector.

### 4.3 Features and regimes

From each window we compute 23 features in three groups (Appendix A): **artifact** (timing/precision quantization: modal-interval fraction, interval CV and entropy, step-length regularity, axis-alignment, collinearity, direction concentration) plus **cadence** (events/s, pause fraction, click/scroll rates); **kinematic** (speed, acceleration, jerk statistics; straightness; curvature; submovement rate); and two **shape-sequence** features (drag fraction, reversal rate). We define four feature *regimes*:

| | |
|---|---|
| **all** (23) | every feature; the in-distribution ceiling |
| **artifact** (11) | timing/precision/cadence markers of the capture channel only |
| **shape** (12) | channel-robust motion geometry + action-mix (no timing/precision) |
| **kinematic** (10) | the pure motion-geometry subset of `shape` |

The artifact/shape split operationalizes the paper's question: how much of "automation detection" is reading the *channel* (timing tick, coordinate precision, RDP batching) versus channel-robust *behavior*?

## 5 Experiment 1: artifact dependence

**Setup.** A transparent detector (standardized $L_2$ logistic regression with balanced class weights) separates human windows from automation windows. We use `StratifiedGroupKFold` (5 folds) with the group set to the source session, so no session is split across folds; out-of-fold window probabilities are averaged to a session score. We report, per regime, the pooled session AUC and the AUC restricted to each automation class (naive/humanized/mimicry). Uncertainty is a 2000-fold bootstrap over sessions.

**Result 1: strong in-distribution detection.** With all features the detector reaches pooled session AUC 0.98 (95% CI $[0.97, 0.99]$). The naive and humanized families are separated *perfectly* in every regime (AUC 1.00); the entire difficulty is the mimicry family (Figure 3, Table 3).

**Result 2: the signal is artifact.** An *artifact-only* detector retains 99.3% of the full-feature pooled AUC (0.974 vs. 0.981). In other words, the timing, precision, and cadence of the capture channel are, on their own, almost sufficient to explain the in-distribution result. The strongest single features separating humans from machine-motion automation are cadence features (`seq_pause_frac` and `seq_events_per_s`, each with $|2 \cdot \mathrm{AUC} - 1| \approx 1.0$), followed by kinematic shape (`kin_curv_mean` 0.93, `kin_v_mean` 0.91, `kin_straightness` 0.89). The very same features separate the *mimicry* family from humans at $\leq 0.26$; they read the channel and gross motion of unsophisticated motion models, not "automation-ness."

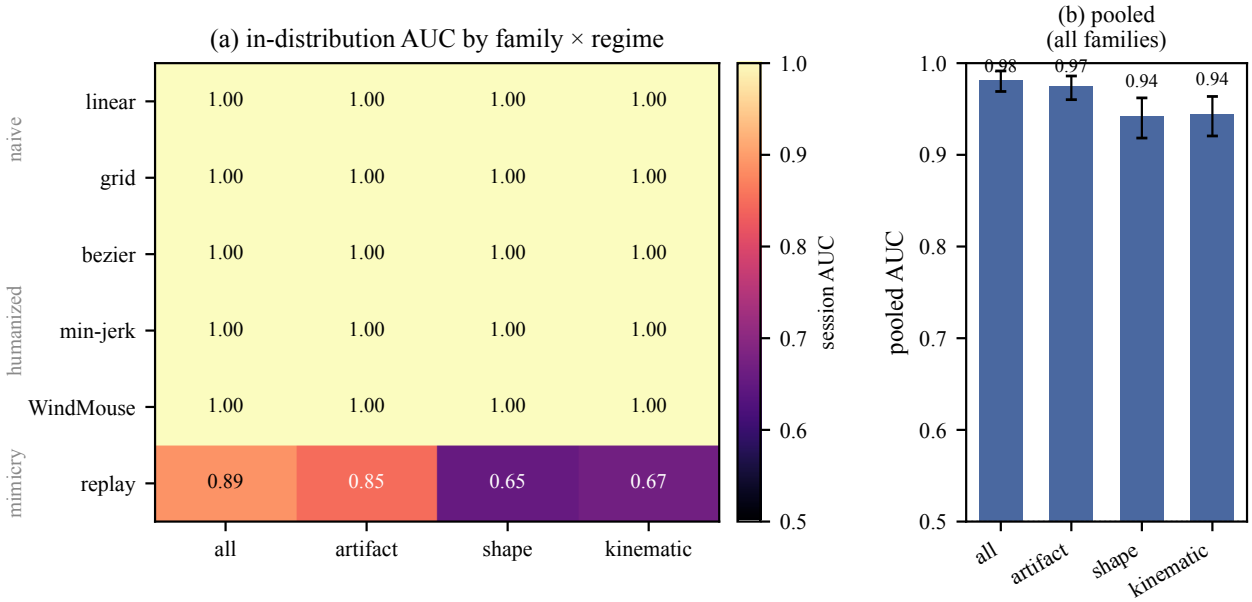



Figure 3: In-distribution automation detection. (a) Session AUC by family × regime: naive and humanized families are perfectly separable in every regime; only the mimicry (`replay`) family is hard, and it gets harder as timing/precision artifacts are removed ($0.89 \rightarrow 0.65$). (b) Pooled AUC per regime with bootstrap CIs; artifact-only nearly matches all-feature.

**Result 3: artifact removal affects the humanized limit.** Dropping the artifact/cadence features (regime `shape`) leaves naive and humanized detection at 1.00 but drives mimicry detection from 0.85 (artifact) to 0.65 (shape). The pooled AUC falls modestly ($0.981 \rightarrow 0.942$) precisely because five of six families are trivially separable by geometry alone; the drop is concentrated where behavior actually has to do the work. This previews Section 6: the in-distribution numbers are not a good guide to what transfers.

| Regime | n | Pooled AUC (95% CI) | naive | humanized | mimicry |
|---|---|---|---|---|---|
| all | 23 | 0.981 (0.969–0.991) | 1.000 | 1.000 | 0.887 |
| artifact | 11 | 0.974 (0.960–0.986) | 1.000 | 1.000 | 0.847 |
| shape | 12 | 0.942 (0.918–0.962) | 1.000 | 1.000 | 0.654 |
| kinematic | 10 | 0.944 (0.921–0.964) | 1.000 | 1.000 | 0.666 |

Table 3: E1 in-distribution session AUC. Artifact-only detection is 99.3% as strong as all-feature detection; the humanized-limit (mimicry) family is the only one that behavioral features must earn, and it is the only one artifact removal affects.

## 6 Experiment 2: cross-family transfer

**Setup.** In-distribution accuracy overstates deployable detection because production traffic contains *unfamiliar* tools. We test transfer with disjoint human negatives (a fixed, user-stratified A/B session split) and per-family session halves. For every ordered pair (train family $i$, test family $j$) we fit a detector on human split A vs. family $i$ (train half) and evaluate it on human split B vs. family $j$ (test half); the diagonal ($i = j$) is in-distribution (disjoint sessions), the off-diagonal is transfer. We also run leave-one-family-out (LOFO): fit against all families but one, evaluate on the held-out family.

**Result 1: artifact detectors anti-transfer.** On the `artifact` regime the transfer matrix (Figure 4 a) collapses: mean off-diagonal AUC is 0.63 versus diagonal 0.96, and the minimum off-diagonal is 0.00. A detector trained on the naive families scores the humanized families *below chance* (e.g., `bezier` evaluated by a `grid`-trained detector: AUC 0.01; `linear` by a `replay`-trained detector: 0.00), because the families' timing signatures are not merely different but *opposite*: naive tools have hyper-regular ticks, the humanized generators are more irregular than humans. An artifact detector thus does not learn "automation"; it memorizes one tool's fingerprint and mislabels the next.

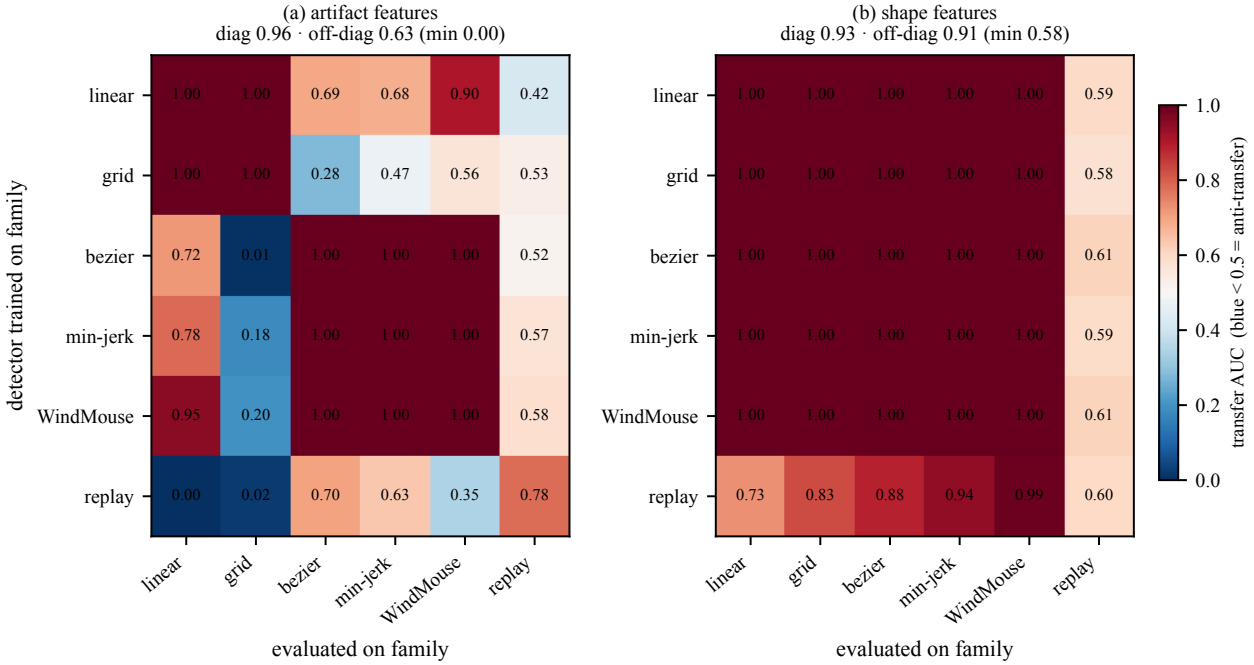
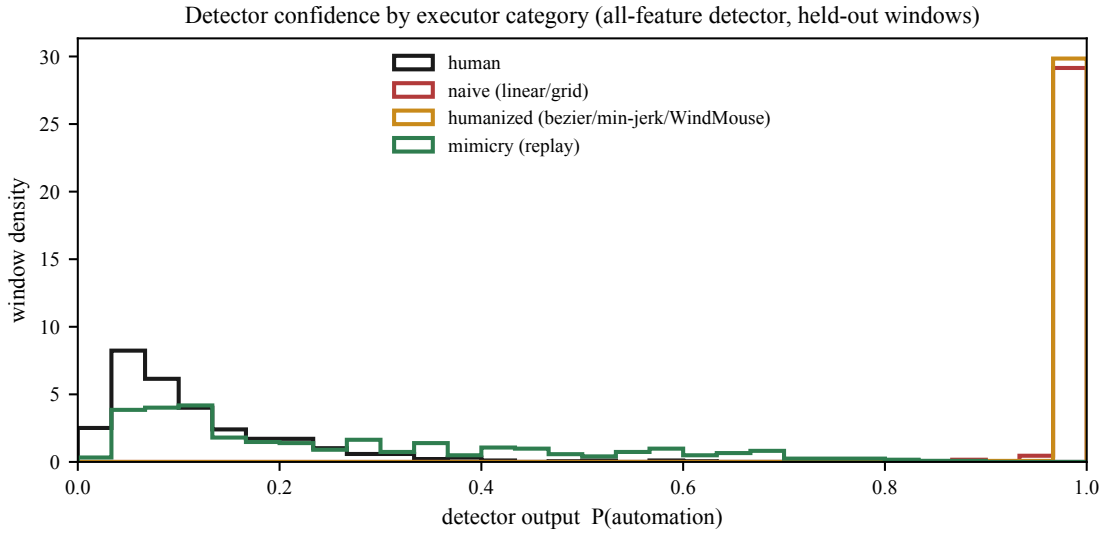



Figure 4: Cross-family transfer AUC (rows: training family; columns: evaluation family). (a) Artifact features anti-transfer across the naive/humanized boundary (blue $< 0.5$). (b) Shape features transfer among the five machine-motion families ($\approx 1.0$) but every detector scores the mimicry family at $\approx 0.6$ (right column). No regime detects `replay` from motion it did not train on.

**Result 2: shape transfers among machine motion, but not to human replay.** On the `shape` regime, the five machine-motion families transfer almost perfectly (off-diagonal 0.91, LOFO 1.00): they share a common non-human signature (too smooth, too continuous, no human pauses or tremor), so training on four generalizes to the fifth. This is the residual, transferable structure the program anticipated (H4), but it is structure that distinguishes *machine motion models* from humans, not automation from humans in general.

**Result 3: mimicry challenges family holdout.** The `replay` family, which reuses human motion, is difficult to distinguish under family holdout: LOFO AUC is 0.51 (all), 0.44 (artifact), 0.57 (shape), 0.60 (kinematic). The artifact result is below chance, while the shape and kinematic point estimates are above chance and are not called undetectable without intervals (Table 4). A detector that has never seen this family, but has seen five others and thousands of human windows, cannot flag automation that emits recorded human motion. Figure 5 shows

why: the detector's confidence on `replay` windows overlaps its confidence on human windows, while naive and humanized synthetic motion separate cleanly.




*Naive and humanized synthetic motion separate cleanly; replayed human motion is indistinguishable from human.*

Figure 5: Detector confidence P(automation) by executor category on held-out windows. Naive and humanized synthetic motion separate from human; replayed human motion is indistinguishable from human. Confidence, not just the top-1 label, reflects the collapse.

| Regime | Diagonal | Off-diagonal | Min off-diag | LOFO mimicry |
|---|---|---|---|---|
| all | 0.958 | 0.841 | 0.062 | 0.507 |
| artifact | 0.963 | 0.625 | 0.000 | 0.439 |
| shape | 0.933 | 0.912 | 0.577 | 0.571 |
| kinematic | 0.933 | 0.908 | 0.558 | 0.598 |

Table 4: E2 transfer summary. Artifact detection anti-transfers (min off-diagonal 0.00). Shape detection transfers among machine-motion families but not to the mimicry family, whose leave-one-family-out AUC is at chance in every regime.

**Reading.** Together E1 and E2 say: the impressive in-distribution numbers are carried by implementation artifacts that are *family-specific*, so they do not generalize to new tools; the portion that does transfer separates machine-motion models from humans, which any tool with access to human motion can erase. There is no behavioral floor guaranteeing automation is detectable.

## 7 Experiment 3: authorization blindness on real traces

We now hold execution fixed at "human" and vary only whether the human is the authorized owner. On the Balabit public test split (685 sessions; 366 legal, 319 illegal), we compare two frozen detectors.

**The automation detector is blind.** We take the best-case all-feature human-vs-automation detector from E1 (it separates human from automation at train AUC 0.96) and apply it, unchanged, to the test sessions. It calls *every* test session human: the maximum session-level P(automation) over all 685 sessions is 0.50, and at a 0.5 threshold it flags 0 legal and 0 illegal sessions. Its ability to order owner vs. identity-mismatch sessions is weakly inverse: AUC 0.42 (95% CI $[0.375, 0.460]$); if anything, identity-mismatch sessions receive marginally *lower* automation scores (mean P 0.083 vs. 0.101), which is spurious with respect to authorization. A detector purpose-

built to find automation contributes nothing to finding unauthorized *human* use, because there is no automation to find.

**The identity detector partially separates, but does not observe authorization.** A transparent per-account template (scaled-Manhattan distance on channel-robust `shape` features, a standard mouse-dynamics identification baseline [3], [4]) reproduces the Balabit task and separates illegal from legal at pooled AUC 0.86 (95% CI $[0.836, 0.892]$; mean per-user AUC 0.85, $[0.80, 0.91]$; per-session EER 0.205). This is genuine signal, but it is a signal about *identity*, not authorization, and it is far from clean: 8.5% of identity-mismatch sessions (27 of 319) look at least as genuine as the median owner session (Figure 6).

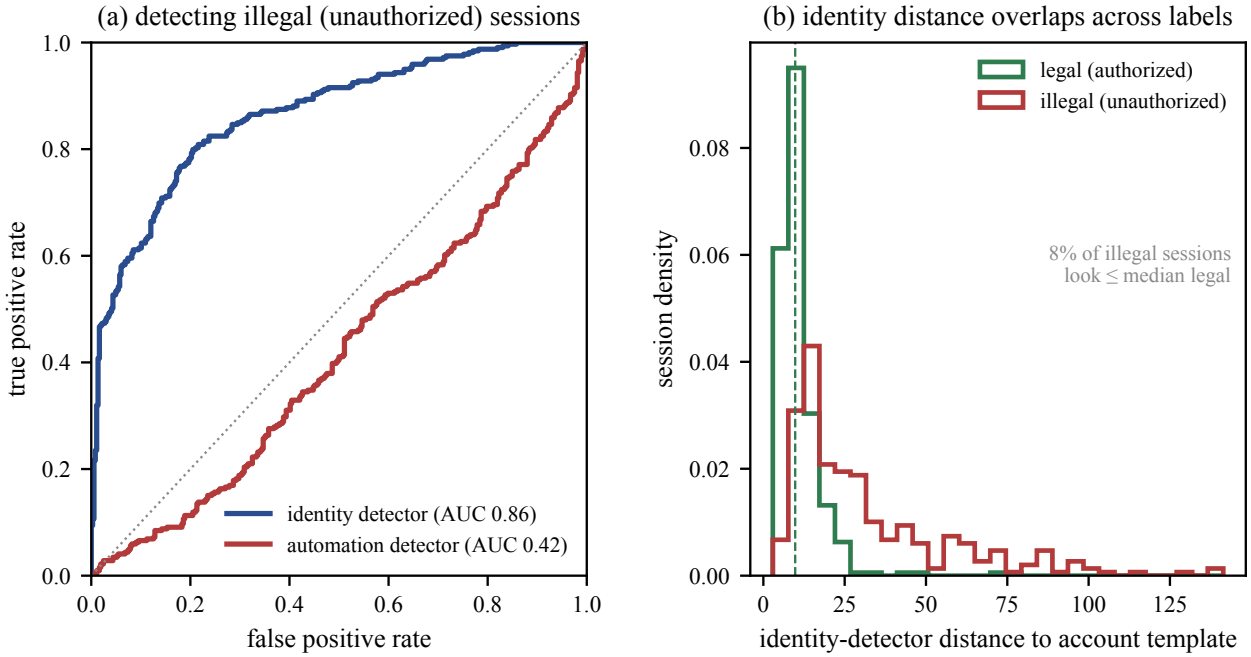



Figure 6: Evidence class: empirical reanalysis of public data. (a) ROC for separating Balabit identity-mismatch sessions from owner sessions: the identity detector is informative (AUC 0.86); the automation detector has weak inverse ordering (AUC 0.42, 95% CI $[0.375, 0.460]$). (b) The identity-distance distributions for owner and identity-mismatch sessions overlap substantially; 8.5% of illegal sessions fall at or below the median legal distance.

**Identity is neither necessary nor sufficient for authorization.** Even a *perfect* identity detector would not yield authorization. A different operator can be authorized (a delegated assistant, shared administrative account, or a legitimately handed-off session); the account owner can be unauthorized (acting outside granted scope, after revocation, or in violation of policy). The Balabit "illegal" label is defined as *identity mismatch*, so identity detection scores well against it, but the quantity a service actually needs, *authorization*, is a different variable that identity only proxies under environment-specific assumptions.

**Observable equivalence is visible in the data.** In the observable shape space, owner and identity-mismatch sessions interleave (Figure 7 a). The nearest cross-label pair (an identity-mismatch session on account `user35` and an owner session on account `user12`) sits at standardized distance 0.28, *closer* than the median within-label nearest-neighbor distance (0.69 for legal, 0.92 for illegal). Two sessions with opposite authorization labels are more alike, in everything the service can measure, than typical same-label sessions are. This is the finite-sample shadow of the formal result we now state.

# 8 The observational equivalence of authorization

We make precise the sense in which authorization is unavailable to a trace-only test, and delimit the claim so it does not overreach.

## 8.1 Setup

Let $w \in W$ be a world with execution summary $X(w) = (e, c, \eta, u)$ (executor, actuation channel, environment, and realized motor/timing signal) and provenance summary $R(w) = (p, \text{scope}, \text{supervision}, \text{revocation})$. A service observes $O(w)$, the interaction trace. Authorization is a predicate $A : W \to \{0, 1\}$ determined by policy applied to provenance, $A(w) = \Pi(R(w), \text{action}, \text{context})$.

**Assumption A1 (execution-mediated observation).** The trace is a function of the execution summary alone: if $X(w_1) = X(w_2)$ then $O(w_1) = O(w_2)$. Equivalently, provenance influences $O$ only through $X$ (Figure 2). This is the formal content of "the destination sees the interface, not the delegation relationship."

**Assumption A2 (authorization autonomy).** There exist worlds that share an execution summary but differ in authorization: $\exists\, w_1, w_2$ with $X(w_1) = X(w_2)$ and $A(w_1) \neq A(w_2)$. This holds whenever the same motions can be performed both within and outside granted authority: for example, an agent issues the identical purchase before and after the principal revokes the mandate, or an operator performs the identical action with and without a policy that permits it.

## 8.2 Result

**Theorem 1 (non-identifiability of authorization).** Under A1–A2 there exist $w_1, w_2$ with $O(w_1) = O(w_2)$ but $A(w_1) \neq A(w_2)$. Consequently no function $h$ of the observable can equal $A$ on all worlds: for any $h : \text{traces} \to \{0, 1\}$ there is a world on which $h(O(w)) \neq A(w)$. On the equivalence class $\{w_1, w_2\}$ any such $h$ has balanced accuracy exactly $1/2$.

*Proof. By A2 pick $w_1, w_2$ with $X(w_1) = X(w_2)$ and $A(w_1) \neq A(w_2)$. By A1, $O(w_1) = O(w_2) =: o^*$. Any $h$ returns one value $h(o^*)$, which differs from $A$ on at least one of $w_1, w_2$; since the two worlds carry opposite labels, $h$ is right on exactly one, giving balanced accuracy $1/2$.* ∎

**Corollary 1 (all trace statistics inherit the blindness).** Any score $s = g(O(w))$ (an automation probability, an identity distance, or an anomaly score) is constant on an $O$-equivalence class and therefore cannot separate the authorized from the unauthorized members of that class. The automation detector of Section 7 (constant "human" across legal and illegal) is the empirical instance.

**Corollary 2 (base rates, not traces, govern the posterior).** If authorized and unauthorized worlds can produce the same execution with positive probability, the posterior authorization probability given an observed trace equals the provenance-induced prior on that fiber; the trace contributes likelihood ratio 1. Trace-only authorization is not universally certifiable; deployment-specific statistical information may exist to the extent authorization is *statistically dependent* on execution in a particular deployment. This dependence is a contingent correlation (e.g., "impostors in this corpus happen to move differently"), never a guarantee, and exactly the kind of correlation E2 shows an adversary can remove.

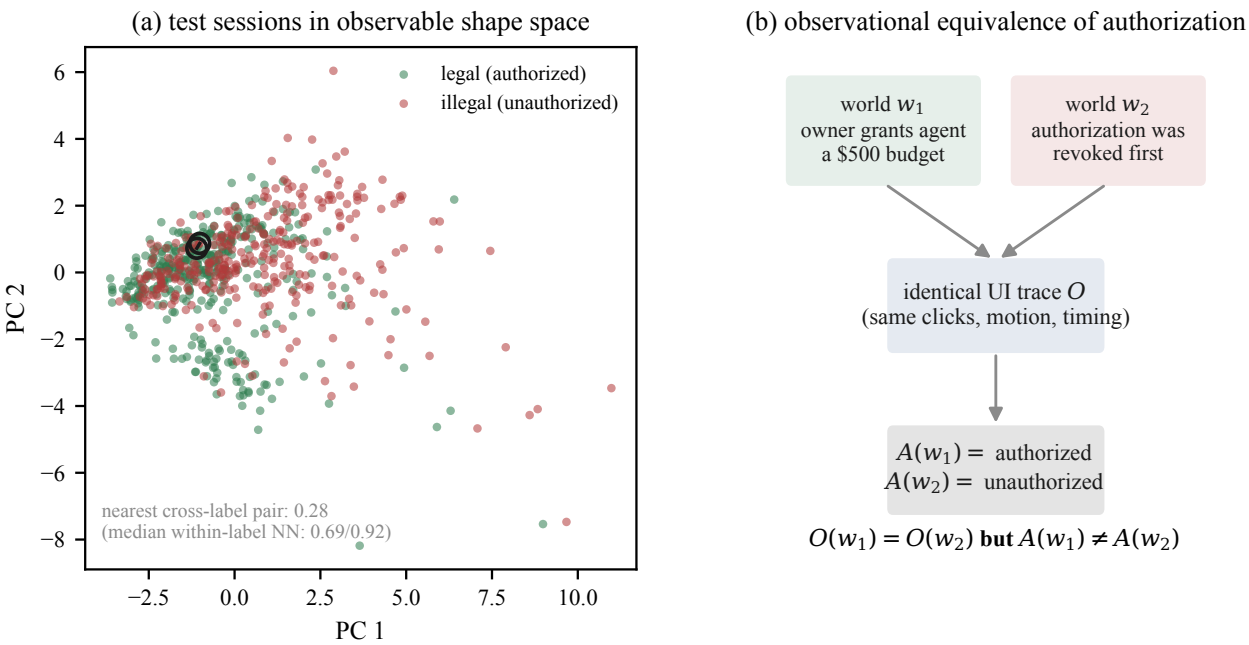



Figure 7: (a) Test sessions in observable shape space (PCA); the circled nearest cross-label pair is closer than typical within-label neighbors. (b) The two-world construction: identical observable trace $O$, opposite authorization $A$. The missing variable is provenance, not behavior.

### 8.3 What the result does not say

The limit is *epistemic*, not computational: it is about the information in $O$, so more data, more compute, or a better model cannot close it (contrast a complexity barrier, which better algorithms might). It is scoped to the provenance axes. It does *not* claim that execution properties are unobservable; A1 leaves the executor/actuation signature in $O$, and Section 5 confirms it is often readable in-distribution. It does not claim detection is useless: behavioral anomaly signals remain valuable for triage. It claims only that no trace-only rule can universally determine or certify *authorization, supervision, and legitimacy*, and therefore that a system which needs them must obtain them elsewhere.

## 9 From negative result to requirements

If legitimacy is not in the trace, it must accompany the action as verifiable provenance. We state this as a *delegation-evidence requirements* framework (Table 5), deliberately treating it as a requirements artifact rather than a new wire protocol. We then assess how far existing public mechanisms already cover each requirement (Figure 8). The lifecycle in Figure 9 shows how the pieces compose: a principal issues a scoped, time-boxed delegation; each hop *attenuates* rather than broadens authority (as in capability systems [5], [6], [7]); consequential actions are checked and logged at an enforcement boundary; signed receipts are emitted; and revocation is verifiable. The trace $O$ is one input to anomaly detection, not the basis for legitimacy.

| Requirement | Minimum content |
|---|---|
| Principal | human or organization on whose behalf the action is taken |
| Agent identity | software/model/agent identifier, version, execution environment |
| Delegated scope | permitted sites, actions, data categories, time window, transaction limits |
| Supervision mode | autonomous · approval-gated · advised · shared · human-executed |
| Action chain | stable references linking plan, delegation, approvals, tool calls, outcome |
| Narrowing rule | downstream delegation cannot broaden authority (attenuation only) |
| Revocation | verifiable status and revocation path |
| Receipts | signed evidence at consequential policy-enforcement boundaries |
| Disclosure | what the destination may inspect vs. what remains private |
| Separation | cryptographic integrity, policy validity, and real-world truth are distinct claims |

Table 5: Delegation-evidence requirements. Each is a property of the delegation relationship, precisely the provenance that Section 8 shows the trace omits.

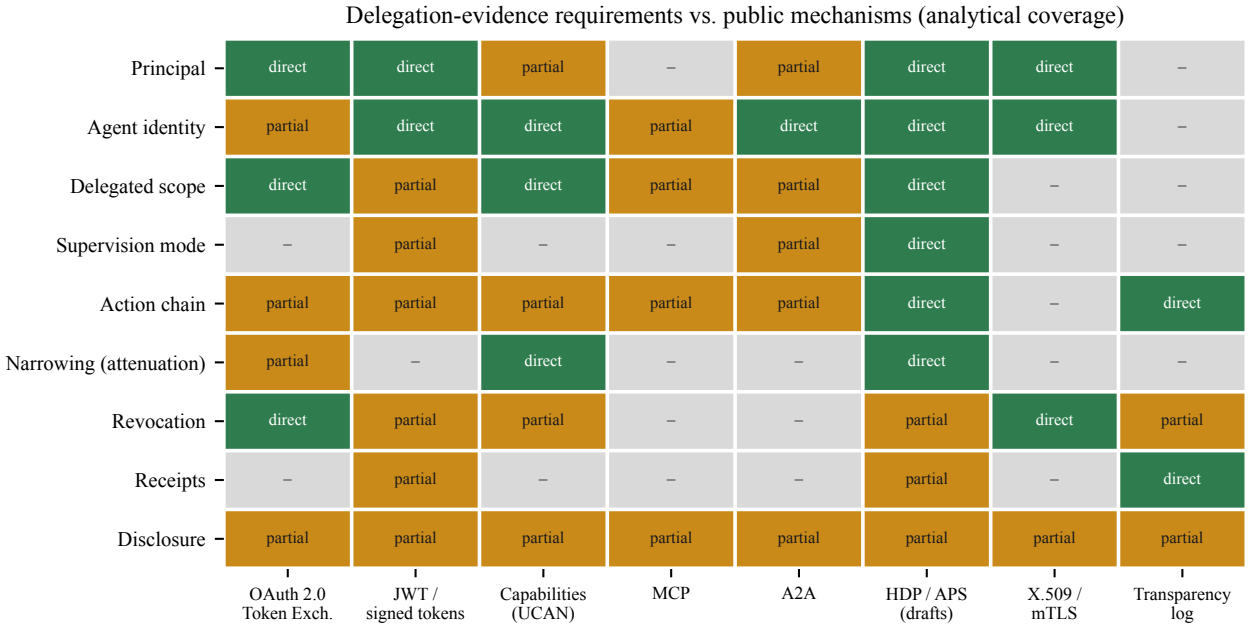



Figure 8: Analytical coverage of the requirements by public mechanisms: OAuth 2.0 token exchange [8], signed tokens/JWT [9], capability systems/UCAN [5], [7], MCP [10], A2A [11], delegation- provenance Internet-Drafts (HDP [12]), X.509/mTLS, and transparency logs [13]. No single mechanism is complete; supervision mode, narrowing, and receipts are the least covered.

Two points follow. First, much of the machinery already exists: scoped tokens, token exchange, capability attenuation, verifiable credentials [14], and transparency logs cover principal, agent identity, scope, and revocation to a useful degree. The gaps concentrate on *supervision mode*, *attenuation guarantees* across delegation hops, and *signed receipts* at enforcement boundaries. These are relationship-level facts, not cryptographic primitives. Second, these mechanisms make claims of different kinds, and conflating them repeats the trace mistake at a higher level: a signature attests *integrity*, a policy check attests *validity*, and neither attests *real-world truth*. A delegation-evidence layer should keep these separate and should present any concrete schema as an experimental research artifact, not an adopted standard.

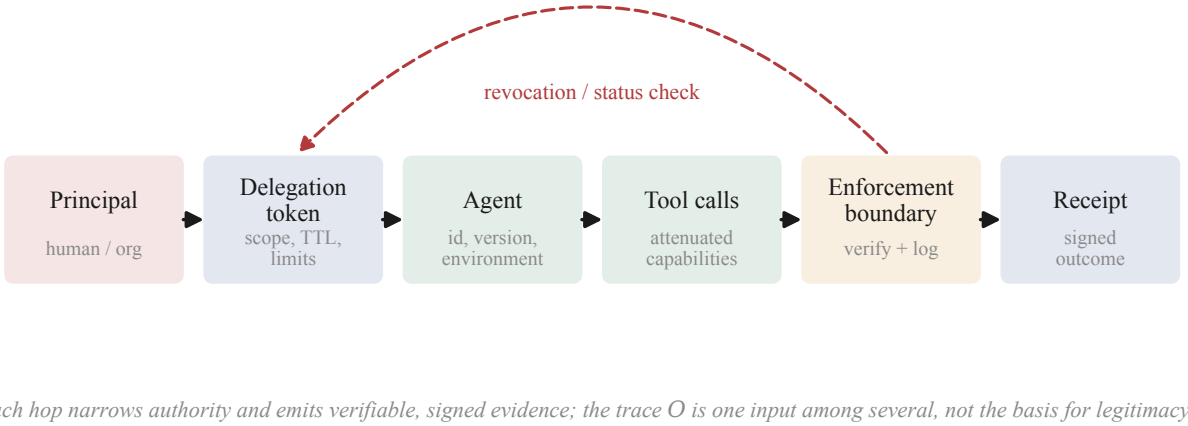



Figure 9: A delegation lifecycle with a signed evidence chain. Authority narrows at each hop and every step emits verifiable evidence; the interaction trace is one input among several, not the source of legitimacy.

## 10 Related work

**Behavioral biometrics and mouse dynamics.** Mouse and pointer dynamics are an established behavioral biometric for identification and continuous authentication [3], [4], [15]. Our identity detector is a deliberately transparent instance of this line, used not to advance identification but to contrast identity with authorization. The imitation/spoofing literature [16] already cautions that such biometrics can be mimicked; our `replay` control makes the point sharply in the delegation setting.

**Motor models.** The humanized generators draw on classical motor control: Fitts' law [17], the minimum-jerk model [2], and the submovement structure of aimed movements [18], which is also why kinematic features separate *naive* automation so easily and humanized automation less so.

**Agent and bot detection.** Recent work fingerprints LLM browser agents from UI traces and asks what minimal features suffice [19]. Our contribution is orthogonal and cautionary: we show, on an independent modality, that such detection is dominated by implementation artifacts and transfers poorly, and we separate the (partly solvable) execution question from the (unsolvable-from-traces) authorization question.

**Authorization and delegation.** The provenance side builds on OAuth and token exchange [8], [20], signed tokens [9], capability security and attenuation [5], [6], [7], [21], verifiable credentials [14], transparency logs [13], and emerging agent-interaction protocols [10], [11] and a delegation-provenance proposal (currently an individual IETF Internet-Draft, not an adopted standard) [12]. We do not propose a new standard; we specify what evidence any solution must carry and assess coverage. The observational-equivalence argument is a causal-identifiability statement in the sense of [22].

## 11 Limitations, dual use, and ethics

**Modality and generators.** Our execution study is one modality (RDP mouse dynamics) and uses *synthetic* automation. The generators are a controlled model of a naive-to-humanized spectrum, not a census of deployed tools; the specific AUCs are properties of these families. We therefore state the transfer findings *conditionally* ("detection built on these artifacts does not transfer to these families") and anchor the authorization result on *real* human data (Section 7), where no synthetic realism is required. The `replay` control is intentionally a best case for the adversary; it establishes a floor (no behavioral guarantee), not a typical case.

**The Balabit label.** "Illegal" denotes injected motion from another human, i.e., identity mismatch, not a genuine authorization violation. This is a strength for our purpose because it isolates the identity/authorization distinction, but it means our identity-detector AUC speaks to identification, and the authorization claim rests on the formal result plus the observation that identity is neither necessary nor sufficient for authorization.

**Construct discipline.** We do not equate automation with malice, agent identity with principal identity, detectability with disclosure, or classification with authorization. Detectors remain useful for anomaly triage; our claim is about what they can *certify*.

**Dual use.** The mimicry result could be read as an evasion recipe. We deliberately publish only openly specified, well-known motion models and a replay control that requires access to recorded human motion, add no novel evasion technique, and release no detector that a defender would rely on for authorization, precisely because our thesis is that no such detector should exist. The constructive artifact is the provenance requirements framework, which improves defense.

## 12 Conclusion

Interaction traces answer an execution question and, even then, mostly by reading implementation artifacts that differ across tools and vanish against an adversary who reuses human motion. They cannot universally determine or certify authorization: identical observable traces can carry opposite authorization, so no trace-only test can provide a deployment-independent separation of authorized from unauthorized action. Deployment-specific statistical associations may still exist. The empirical and formal results point the same way. Trustworthy delegation on an agentic web will not come from better behavioral classifiers; it will come from scoped, attenuating, revocable, and receipted provenance that travels with the action, while behavior is kept in its proper role as a signal for anomaly, not a verdict on legitimacy.

Reproducibility. All results are deterministic (seed 42) and regenerated by `analysis/run_all.py` using the repository Python environment; numbers are read from `results/results.json` and figures from `figures/`. The study uses only the public Balabit corpus [1] and the openly specified generators of Appendix B.

## Appendix A: Feature definitions

All features are computed per window of 800 consecutive move-samples, using only within-stroke steps (consecutive samples separated by $\leq 0.20$ s); idle gaps are excluded from kinematics.

**Artifact (timing/precision).** `art_dt_modal_frac` (fraction of inter-sample intervals at the modal millisecond), `art_dt_cv` (coefficient of variation of intervals), `art_dt_entropy` (Shannon entropy of the millisecond-binned interval histogram), `art_step_modal_frac` (fraction of steps at the modal integer step length), `art_frac_axis_aligned` (fraction of steps with zero horizontal or vertical component), `art_frac_collinear` (fraction of successive step pairs that are collinear), `art_dir_concentration` (1 minus the normalized entropy of 5°-binned step directions).

**Cadence (capture channel rate).** `seq_events_per_s`, `seq_pause_frac` (share of inter-sample gaps exceeding the stroke threshold), `seq_click_rate`, `seq_scroll_rate`.

**Kinematic (channel-robust geometry/dynamics).** `kin_v_mean`, `kin_v_std`, `kin_v_p95`, `kin_v_max` (speed statistics, px/s); `kin_acc_absmean`, `kin_acc_std` (acceleration); `kin_jerk_absmean` (jerk); `kin_straightness` (net displacement over path length, per stroke); `kin_curv_mean` (mean absolute turning angle per step); `kin_submov_rate` (interior speed maxima per 100 steps, a submovement proxy [18]).

**Shape-sequence.** `seq_drag_frac` (share of Drag samples), `seq_reversal_rate` (share of successive step vectors with negative dot product).

## Appendix B: Automation generator specifications

Each generator is seeded (a fixed base per family plus the session index), samples a screen resolution from those observed in the human data, and emits waypoint-to- waypoint moves with occasional clicks; humanized families add inter-action pauses.

- **`linear`.** Constant velocity $v$; fixed tick $\Delta t \in \{8, 10, 16\}$ ms; straight interpolation with integer rounding. Perfectly regular timing and direction.
- **`grid`.** As `linear` but each move is an axis-aligned two-leg (horizontal then vertical) path; produces right-angle turns and high axis-alignment.
- **`bezier`.** Cubic Bézier between waypoints with laterally offset control points, ease-in-out ($3u^2 - 2u^3$) sampling, sub-pixel Gaussian jitter, and a tick jittered around 12–20 ms; pauses $\sim \mathrm{Exp}(0.4\ \mathrm{s})$ half the time.
- **`minjerk`.** Flash–Hogan minimum-jerk profile $s(u) = 10u^3 - 15u^4 + 6u^5$ [2] with jittered tick and small positional noise; single-peaked, bell-shaped speed.
- **`windmouse`.** The WindMouse algorithm (gravity toward the target plus a random-walk wind term, with velocity clamping); curved, humanized paths with reversals.
- **`replay`.** Concatenates contiguous full-event blocks (1200 events) sampled from the real human training sessions, translated to fresh integer offsets and joined by human-sampled pauses. Within-block behavior and timing are human by construction.

## Appendix C: Reproducibility

`analysis/common.py` (loading, cleaning, windowing, 23-feature extraction, metrics), `analysis/generators.py` (the six families), `analysis/build_dataset.py` (feature tables), `analysis/e1_artifact_dependence.py`, `analysis/e2_cross_family_transfer.py`, `analysis/e3_authorization_blindness.py`, and `analysis/make_figures.py`. Detectors are standardized $L_2$ logistic regression; grouping is by source session; uncertainty

is a 2000-fold bootstrap. Running `analysis/run_all.py` regenerates `results/results.json`, the cached CSVs, and all figures deterministically.

## References


[1] Á. Fülöp, L. Kovács, T. Kurics, and E. Windhager-Pokol, "Balabit Mouse Dynamics Challenge Data Set." 2016.

[2] T. Flash and N. Hogan, "The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model," *Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985, doi: 10.1523/JNEUROSCI.05-07-01688.1985.

[3] C. Shen, Z. Cai, X. Guan, Y. Du, and R. A. Maxion, "User Authentication Through Mouse Dynamics," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 16–30, 2013, doi: 10.1109/TIFS.2012.2223677.

[4] N. Zheng, A. Paloski, and H. Wang, "An Efficient User Verification System via Mouse Movements," in *Proc. ACM Conf. on Computer and Communications Security (CCS)*, 2011, pp. 139–150. doi: 10.1145/2046707.2046725.

[5] M. S. Miller, "Robust Composition: Towards a Unified Approach to Access Control and Concurrency Control," Doctoral dissertation, 2006.

[6] A. Birgisson, J. G. Politz, Ú. Erlingsson, A. Taly, M. Vrable, and M. Lentczner, "Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud," in *Proc. Network and Distributed System Security Symp. (NDSS)*, 2014.

[7] UCAN Working Group, "User Controlled Authorization Network (UCAN) Specification, Version 1.0.0."

[8] M. B. Jones, A. Nadalin, B. Campbell, J. Bradley, and C. Mortimore, "OAuth 2.0 Token Exchange," RFC 8693, 2020.

[9] M. B. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, 2015.

[10] Anthropic, "Model Context Protocol (MCP) Specification." 2024.

[11] A2A Project, "Agent2Agent (A2A) Protocol Specification." 2025.

[12] Helixar Limited, "Human Delegation Provenance Protocol (HDP)." 2026.

[13] B. Laurie, A. Langley, and E. Kasper, "Certificate Transparency," RFC 6962, 2013.

[14] W3C, "Verifiable Credentials Data Model v1.1." 2022.

[15] A. A. E. Ahmed and I. Traore, "A New Biometric Technology Based on Mouse Dynamics," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 3, pp. 165–179, 2007, doi: 10.1109/TDSC.2007.70207.

[16] Z. Jorgensen and T. Yu, "On Mouse Dynamics as a Behavioral Biometric for Authentication," in *Proc. ACM Symp. on Information, Computer and Communications Security (ASIACCS)*, 2011, pp. 476–482. doi: 10.1145/1966913.1966983.

[17] P. M. Fitts, "The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement," *Journal of Experimental Psychology*, vol. 47, no. 6, pp. 381–391, 1954, doi: 10.1037/h0055392.

[18] D. E. Meyer, R. A. Abrams, S. Kornblum, C. E. Wright, and J. E. K. Smith, "Optimality in Human Motor Performance: Ideal Control of Rapid Aimed Movements," *Psychological Review*, vol. 95, no. 3, pp. 340–370, 1988, doi: 10.1037/0033-295X.95.3.340.

[19] W. Lugoloobi, S. Marro, J. Magomere, J. Wright, and C. Russell, "Known By Their Actions: Fingerprinting LLM Browser Agents via UI Traces." 2026.

[20] D. Hardt, "The OAuth 2.0 Authorization Framework," RFC 6749, 2012.

[21] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, "SPKI Certificate Theory," RFC 2693, 1999.

[22] J. Pearl, *Causality: Models, Reasoning, and Inference*, 2nd ed. Cambridge University Press, 2009.