

Research Article

Mini Me: Personal AI That Was There Continuum, a Context Substrate for Measurable Presence and Safe Action

Nirbhay Pherwani¹, Raghul Krishnan¹

1. Independent researcher

Today's personal AI can read an inbox, summarize a meeting it joined, or act inside one application. It still loses the thread when a task crosses a conversation, a screen, a room, the body, and several devices. The person must then reconstruct that thread before asking for help. We argue that the missing capability is *presence*: the system was not there when the context was made, so the person must stand in for it later. *Mini Me* names the experience of a personal AI that was present across a person's conversations, screens, body, devices, and spaces. *Continuum* is the context substrate we propose beneath that experience: a set of person-controlled device trust domains that capture across devices, distill typed memory, compile a purpose-scoped disclosure bundle for any remote processing, and gate proactive action by risk.

We define presence so that it can be measured and falsified, along four axes: *continuity*, policy-permitted context coverage by modality, device, and time; *comprehension*, briefing elimination and time to useful context; *anticipation*, next-action precision and useful lead time; and *authority*, autonomy yield bounded by regret, correction, and undo. Presence is deliberately bounded, freshness-aware, and policy-permitted; it is not a euphemism for total capture.

The paper's narrow novelty claim is the proposed composition of measurable presence, typed and provenance-aware memory, purpose-scoped disclosure, cumulative modeled exposure, and content-independent authority. It contributes a reference architecture with a presence fabric, plane contracts, and per-device trust domains inside one personal trust perimeter; a proposed autonomy model with asymmetric calibration, signed grants, provider-verified action preconditions, engineered reversibility, an attention-aware delivery policy, and a four-gate action decision procedure; and an evaluation agenda. All of it is a falsifiable proposal, not an implemented system. Local storage does not defeat endpoint compromise; disclosure metadata cannot force a remote processor to forget; always-

available sensing remains socially and legally dangerous. We state these limits directly and propose the experiments that would confirm or refute the design.

Correspondence: papers@team.qeios.com — Qeios will forward to the authors

1. Introduction

You are in a meeting when the plan changes. A feature your team has debated for weeks is cut, for reasons that unfold across the room, the shared document on your laptop, and a decision you made a month earlier. Your phone lies on the table the whole time. That night you sleep badly, and the follow-up message you meant to send slips to a morning that has no room for it. Two days later a colleague asks what was decided and why. You open your assistant, a system trained on much of the public web, and start typing out your own meeting, your own reasoning, your own week.

The assistant is capable. It was simply not there. So you brief it, message by message, the way you have briefed it every day you have used it.

You should never have to explain your own life to your own AI.

1.1. *The Briefing Tax*

The dominant interface to personal AI is still a briefing. The person opens a model, restates the goal, pastes the relevant email, summarizes the meeting, finds a screenshot, explains a preference, and then asks for help. The model may be capable; the interaction is not. The human remains the context integration layer, and every session pays the same reconstruction cost. We call this the briefing tax.

Embedding a model in email reduces the tax for email. Giving it a meeting transcript reduces the tax for that meeting. Neither supplies the missing thread when a decision begins in a room, changes in a shared document, continues on a phone, depends on the person's current state, and ends as an action in another service. Application integration grants access to one silo at query time. Presence preserves a policy-bounded, provenance-bearing thread while it is being formed across physical and digital settings.

The tax is not only an inconvenience. It places a hard ceiling on assistance:

- **Context arrives late.** By the time the person reconstructs an event, decisions and commitments have already decayed.

- **Context arrives incomplete.** Screens, physical activity, conversation, location, and device state live in separate stores.
- **Context arrives without provenance.** A model cannot distinguish an observed fact from an old summary or a guessed preference.
- **Action remains detached from memory.** A meeting recorder can recover what was said but not safely decide what should happen next.

The obvious response—record everything and upload it—creates a second failure. A life stream is unusually sensitive: it contains not only the person’s information but other people’s voices, faces, homes, health, work, and private spaces. A cloud service with standing query access to that stream is surveillance infrastructure even if its interface is friendly.

We argue that the missing capability is *presence*: the system was there, within the person’s rules, when the context was made, so the person no longer has to carry it back. We define presence precisely, and measurably, in Section 2. The substrate that would provide it is a *context operating system*: a role filled by middleware and protocol contracts, not a phone kernel, that coordinates sensing, memory, model placement, disclosure, and action under the person’s policy and host-OS constraints. We call the experience **Mini Me** and the proposed substrate **Continuum**.

1.2. Position

Continuum takes five positions.

1. **Logging is necessary and must be admitted.** Multi-device sensing, including audio and video for declared uses, is part of the design. No single wearable or forward-facing viewpoint defines the architecture.
2. **Local execution is the default, not the limit.** Routine capture, indexing, retrieval, redaction, and reasoning should remain inside the personal trust perimeter; selected tasks may use remote processors.
3. **Disclosure is compiled per task purpose.** Remote systems receive a bounded disclosure bundle, not standing access to the vault.
4. **Proactivity is a safety decision.** The system must choose consistently among act, draft, ask, and deny; denial may result in silence or an explanation.
5. **Mobile may evolve from app launcher toward context substrate.** Apps remain important but may increasingly expose capabilities and entities to a system-level intent broker.

1.3. Contributions

As a position paper, this work contributes a decomposition and test agenda rather than an implemented system. Together with the threat model of Section 6, it proposes six groups of mechanisms for implementation and evaluation, each tagged with the presence properties it chiefly supports:

Contribution 1 — Presence Fabric and Distributed Architecture (continuity).

1. A presence fabric that enrolls devices, inventories what each source can sense and do, tracks attribute freshness, carries a task across devices, arbitrates output surfaces, and degrades predictably.
2. Four data-path planes—Capture, Memory, Resolve, and Act—under one Policy control plane, each with a contract, an invariant, and a local / remote-processing execution boundary.
3. Device trust domains, one per enrolled device, with scoped wrapped keys, bounded revocation, typed conflict handling, and a lower-trust shared-space domain that holds no vault plaintext.

Contribution 2 — Multimodal Perception with Deliberate Retention (continuity, comprehension).

1. A modality inventory stating representative use, sensitivity, processing placement, and default retention for each sensing class.
2. A first-person vision pipeline from an encrypted rolling buffer through local parsing, an attention gate, edge redaction, and policy-driven expiry.
3. A typed, freshness-aware device state model, with clipboard access gated separately from ambient device state.

Contribution 3 — Typed Memory, Commitments, and Next Actions (comprehension, anticipation).

1. A layered memory model separating observations, episodes, semantic memory, and asserted facts over a provenance DAG.
2. Commitment and next-action objects carrying confidence, evidence, deadlines, owners, preconditions, expected value, cost of interruption, and required authority.
3. Supersession, correction, deletion requests and local tombstone propagation, and a bounded review queue for inference the system declined to act on and declined to discard.

Contribution 4 — Purpose-Scoped Context Flow (comprehension).

1. A disclosure-bundle schema carrying purpose, provenance, sensitivity, exclusions, expiry, and processor constraints, without transmitting local correlation aids to the processor.
2. A sequential exposure-accounting rule that minimizes marginal disclosure subject to task sufficiency and cumulative local budgets over joinable histories.
3. Local receipts that record marginal and cumulative exposure accounting without claiming cryptographic control after disclosure.
4. An outbound-audience check applied to any artifact addressed to a human recipient, with per-principal charging and coalition-aware composition.

Contribution 5 — Calibrated Autonomy and Safe Action (anticipation, authority).

1. A signed grant object with a signer key identifier, algorithm, canonical byte representation, audience, scope, impact, value, externality, and confidence ceilings, rate limits, expiry, replay protection, and revocation, as the only source of authority.
2. Authority levels L0 through L4 assigned per policy domain, capability, and impact class rather than one global setting, with asymmetric calibration: bounded promotion from accepted previews, immediate demotion on regret, correction, undo, or recall.
3. Reversibility engineered by the substrate—delayed sends, pending queues, staged commits, idempotency, preview and commit binding, and bounded compensation—rather than assumed from a provider; compensation never erases an external side effect.
4. A delivery policy that decides whether, when, where, and how to interrupt under a contextual interruption budget, and that can never raise an authority level.
5. The Four Gates decision procedure, in which confirmation can satisfy the envelope gate but never the two mandatory ones, with digital and physical-space action contracts and a per-scenario authority mapping.
6. Environmental prompt injection treated as a first-class mobile threat.

Contribution 6 — Mobile-Stack Scenario and Evaluation Agenda (all four properties).

1. A hypothesized transition from app-centric interaction to capability-centric orchestration.
2. Developer contracts for discoverable actions, provenance, idempotency, preview, and undo.
3. Falsifiable capability, privacy, systems, and agency metrics.

1.4. Contribution Boundary

Continuum is a reference architecture, not a phone kernel, a completed product, or a claim of general intelligence. It does not guarantee that a remote model deletes disclosed context. It does not solve joint ownership of a recorded conversation. It does not make continuous capture lawful or socially acceptable. It does not require a blockchain. The contribution is the separation of responsibilities and the contracts among them.

The paper proceeds from experience to mechanism. Section 2 defines presence as a measurable property. Section 3 walks through one connected day with Mini Me and a catalog of worked scenarios that later sections reference. Section 4 fixes scope, terms, and design goals, and Section 5 situates the proposal in prior work. Section 6 states the threat model. Sections 7 through 9 build the system: the presence fabric, the plane contracts, the device state model, and the device trust domains; perception across eyes, body, devices, and spaces; and memory, commitments, and anticipated next actions. Section 10 gives purpose-scoped disclosure, and Section 11 gives grants, authority levels, delivery, and the action decision procedure. Section 12 develops the mobile-stack scenario, Sections 13 and 14 give the evaluation plan and discussion, and Section 15 concludes.

2. What It Means for Personal AI to Be Present

An assistant is *present* to the degree that, at the moment help is needed, it already holds the policy-permitted context relevant to that moment and can carry accepted work to completion at the highest safe level of autonomy. Presence is a property of the whole system, not of the model: a fluent model without context is absent, and a context archive without comprehension or authority is a filing cabinet.

Stated that way, presence would remain a slogan. We therefore decompose it into four properties, each with proposed measures (Table 1).

Property	Definition and proposed measures
Continuity	Policy-permitted context coverage by modality, device, and time.
Comprehension	Assistance without reconstruction: Briefing Elimination Rate and time to useful context.
Anticipation	Likely commitments and next actions identified before an explicit command: next-action Precision@ k and useful lead time.
Authority	Accepted actions completed at the highest safe autonomy level: autonomy yield, bounded by regret, correction, and undo.

Table 1. Presence decomposed into four measurable properties. The measures are proposed instruments: defined below, and put forward as the measured quantities for the evaluation plan (Section 13). None has been validated empirically.

Continuity is the fraction of the person’s declared modality, device, and time scope that the system covers under current policy; it is bounded above by what the person permits, not by what sensing could reach. *Briefing Elimination Rate* is the fraction of tasks completed without the person supplying context the system was already permitted to hold, and *time to useful context* is the elapsed time from a request to a sufficient working context. *Precision@ k* is computed over the system’s ranked next-action candidates against what the person actually accepted, and *useful lead time* is how far ahead of the deadline or moment of need a correct candidate surfaced. *Autonomy yield* is the fraction of accepted actions completed at the granted autonomy level without escalation; because authority is assigned per cell rather than globally (Section 11.3), it is reported per cell rather than aggregated across unlike domains, and it is meaningful only alongside its costs, namely the regret rate, the correction rate, and whether undo remained available.

Perfect knowledge is neither possible nor desirable. Sensors fail, devices sleep, batteries drain, and people pause, revoke, and delete. A system that pursued total capture would violate the very policies presence is defined under and would turn bystanders into context without their agreement. Presence is therefore bounded, freshness-aware, and policy-permitted: each property is measured against what the person allowed, every source carries its staleness, and what was deliberately not observed is recorded as a

boundary rather than papered over. Figure 1 makes those admitted capture and delivery surfaces concrete across body, devices, and spaces. The perception principle behind this boundary is: *sense within enrolled policy, understand locally, remember deliberately*.

What the person can perceive is not limited to one forward-facing camera. Continuum treats enrolled sources as a *perceptual envelope*: body-worn viewpoints may face forward, sideways, or behind; paired room and vehicle sensors may observe relevant state from elsewhere; and physiological devices may contribute signals from the body. Each viewpoint remains a separately identified source with its own owner, trust level, field of observation, retention rule, and bystander constraints. The envelope describes available evidence, not an entitlement to capture everything around the person.

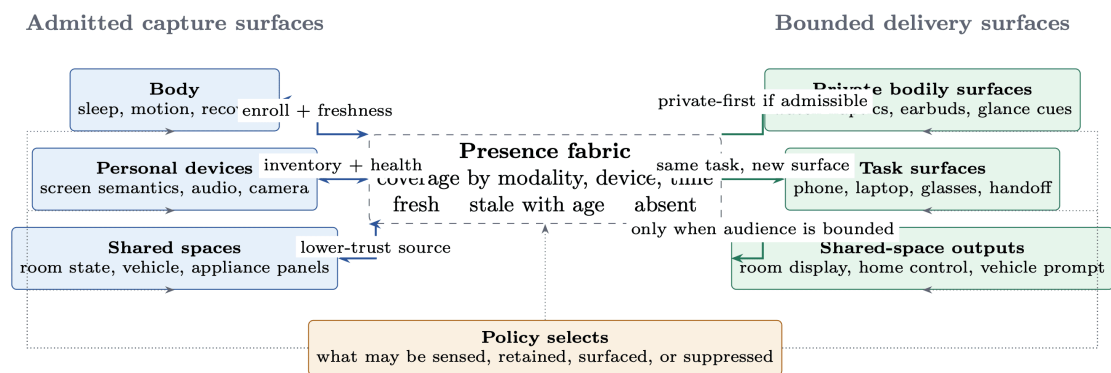


Figure 1. Presence surfaces across body, devices, and spaces. The fabric does not claim total capture: it tracks only enrolled, policy-permitted sources and chooses only admissible delivery surfaces, always carrying freshness and absence forward rather than inventing continuity.

Presence is also distinct from its nearest neighbors. It is not lifelogging: a complete archive can have high coverage yet no comprehension, anticipation, or authority. It is not a larger context window: a window is capacity, not acquisition, and it leaves the person as the integration layer. It is not general GUI automation: an agent that executes explicit instructions has authority without continuity. Section 5 compares the closest systems in each of these neighborhoods.

3. A Day with Mini Me

The day below is a set of design probes, written to be tested against the evaluation plan, not a report of a working system or predicted behavior. Each scenario in the catalog states five fields. *Inputs* are the

signals the scenario consumes. *Retained* is the derivative kept after raw signals expire. *Autonomy* resolves to one of the four action outcomes specified in Section 11: act, draft, ask, or deny. An outcome is not an autonomy level; “act” in a scenario whose permitted effect is presentation changes nothing outside the person’s own surfaces. Table 9 states, for each scenario, the policy domain, the capability, the impact class, the authority level, and the constraint that binds the outcome. *Disclosure* names what, if anything, crosses the personal trust perimeter as a purpose-scoped bundle (Section 10). *On failure* states the fallback behavior. Terms used here, including grant and disclosure bundle, are defined in Section 4. The mechanisms the scenarios name are specified later: the device state model in Section 7.3, home inventory memory and the review queue in Section 9, the outbound-audience check in Section 10, the grant object and the authority levels in Sections 11.2 and 11.3, and the home-control envelope and hub trust level in Sections 7.5 and 11.11.

The catalog deliberately crosses different kinds of life rather than assuming one office-bound person. S1, S2, and S11 concern collaborative work and creation; S3, S4, S7, and S12 concern ordinary personal tasks; S5, S6, S9, and S10 concern the body, health, and safety; S8 concerns a shared home. In each case the plain question is the same: what happened, what did Mini Me understand, and what could it safely do next?

3.1. *One Connected Day*

Your watch recorded a short night. Before you wake, Mini Me moves your private focus block later in the day; the two meetings it proposes to shift wait for your approval because other people are on them, and when you approve them, the rescheduling messages will not say why. At breakfast the dishwasher blinks an error. You look at the panel for a moment, and the code, the model, and the likely fix arrive as steps on your phone while the video buffer that read the panel expires unpromoted.

In the morning meeting, decisions, owners, and deadlines are extracted as the discussion happens, each linked to the moment it was made. When the meeting ends, the follow-up message is already drafted and held for your review; the audio it came from is gone. On a customer call after lunch you move between browser tabs and a whiteboard sketch, and before the call ends a product sheet citing each source is waiting. You start editing it on your phone in the hallway and finish on your laptop, the same thread, the same citations, without reopening anything.

At the gym, cues arrive in your earbuds from your own motion, recovery state, and training history; nothing is filmed and nothing leaves your devices. On the walk home a friend asks for the trail map you promised weeks ago. The promise had been kept as a commitment; the conversation it came from had

not been kept at all. Your home warms before you arrive, within the envelope you granted and only because no one else is there to disagree. A utility-bill anomaly is stopped and explained; any payment still waits for your explicit approval.

Before bed you review a one-page brief for tomorrow's medical appointment, every line linked to the record it came from, shared with no one until you choose to share it. Two of the scenarios below, the fall and the disrupted trip, did not happen on this day. Presence matters most on the days when something goes wrong.

Figure 2 lays the day on a single axis: the physical signals Capture admits above the line, and the digital or physical outcome Policy and Act produce below it. Read left to right, the line is continuity made visible; read top to bottom at any one point, it is comprehension, anticipation, or authority resolving a signal into one of the four action outcomes.

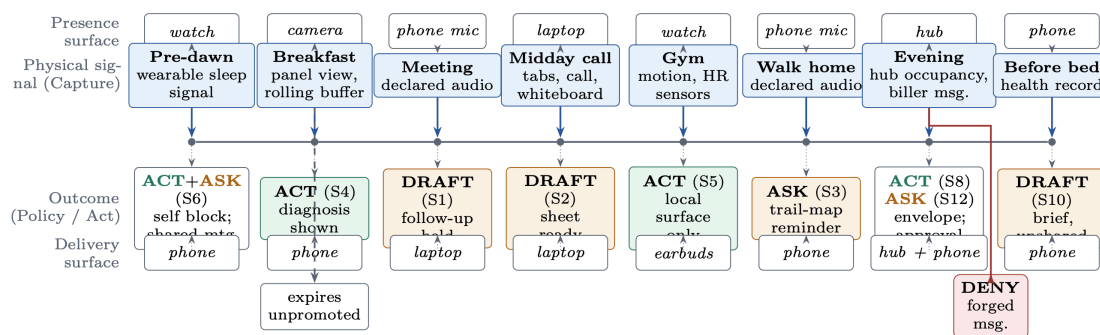


Figure 2. One day on a single axis. Physical signals that Capture admits sit above the line; the digital or physical outcome that Policy and Act produce from each sit below it, colored by outcome (green act, amber draft or ask, red deny; mixed cells show branch-dependent outcomes). The outer rows name the presence surface each signal arrived on and the delivery surface each outcome reached, which is the continuity claim of Section 2 made concrete per event rather than asserted once for the whole day. Comprehension is what a draft removes from the person; anticipation is the walk-home reminder; authority is why S8 acts inside an envelope while S12 requires explicit approval; a forged message denies and an unpromoted buffer expires without promotion. S9 and S11 are omitted because, as stated above, they did not happen on this day; S7's continuation is folded into the midday-call column.

3.2. Scenario Catalog

S1 — Meeting follow-through.

Inputs: calendar event, declared-capture meeting audio, the shared document open on the laptop. *Retained:* decisions, owners, deadlines, and unsent follow-ups as typed commitments with provenance; raw audio expires after extraction. *Autonomy:* draft; the follow-up message is held for approval, and sending after acceptance falls under a standing low-impact grant. *Disclosure:* none required; if a remote model polishes prose, the bundle carries only the decisions and recipients, excluding unrelated calendar, health, and location memory. *On failure:* low speaker-attribution confidence marks commitments uncertain, and the system asks instead of drafting.

S2 — Product sheet from browser, call, and whiteboard.

Inputs: screen state from open tabs, declared call audio, a deliberate camera capture of the whiteboard. *Retained:* extracted requirements and the sheet as a generated artifact citing each source; call audio is not retained. *Autonomy:* draft, delivered before the call ends. *Disclosure:* extraction is local; a remote drafting bundle carries only the requirement list. *On failure:* an unreadable whiteboard region or missing tab appears in the sheet as a stated gap, never as invented content.

S3 — Friend commitment without transcript retention.

Inputs: declared conversation audio near the phone. *Retained:* one commitment object with time, subject, and a provenance stub; no transcript. *Autonomy:* ask; a reminder with a one-tap draft, because the recipient is a personal contact and content selection is ambiguous. *Disclosure:* none. *On failure:* a low-confidence inference lands in a review queue for the person to confirm.

S4 — Appliance repair.

Inputs: first-person view of the appliance error panel, the appliance model from home inventory memory. *Retained:* the error code and resolution steps; the video buffer expires without promotion. *Autonomy:* act to present the diagnosis on the person's own surface; deny actuation; ask before the remote lookup if it exceeds the standing low-sensitivity lookup grant. *Disclosure:* a lookup bundle carrying model number and error code, no home imagery. *On failure:* an unreadable code produces a request to photograph the panel.

S5 — Gym coaching.

Inputs: motion sensors, recovery and health signals, training history; camera-based form analysis only where the space's declared policy permits it, with other people suppressed. *Retained:* set counts and a derived load trend; no video. *Autonomy:* act, restricted to the person's own audio surface; no external

action. *Disclosure*: none; coaching is local. *On failure*: with sensors missing, coaching degrades to the signals that remain.

S6 — Sleep-aware scheduling.

Inputs: wearable sleep signals, the calendar. *Retained*: a derived readiness state with an expiry; raw physiology stays in the health store. *Autonomy*: act within a standing grant for self-owned blocks; ask before touching anything with other participants. *Disclosure*: rescheduling messages never state the health reason; the outbound-audience check strips it. *On failure*: conflicting constraints stop the cascade and produce a question, not a chain of moves.

S7 — Phone-to-laptop continuation.

Inputs: the device state model and the task thread on the phone. *Retained*: the task thread with citations. *Autonomy*: act; presenting a continuation is low impact, and which policy domains may appear on which device remains policy-controlled. *Disclosure*: none; synchronization stays between device trust domains under scoped keys. *On failure*: stale laptop state is shown with its freshness, and the thread is re-derived.

S8 — Occupant-aware home preparation.

Inputs: arrival estimate, home hub occupancy signals. *Retained*: arrival preferences as derived beliefs open to correction. *Autonomy*: act within a bounded home-control envelope with physical interlocks; deny anything that overrides another occupant's stated preference. *Disclosure*: the hub receives commands, not personal context; hubs stay lower-trust and hold no vault plaintext. *On failure*: devices fall back to safe defaults, retries are bounded, and the failure is surfaced on arrival.

S9 — Fall escalation.

Inputs: wearable motion and impact signals, a brief local audio check-in. *Retained*: an incident record. *Autonomy*: an ordered escalation: ask the person; then act under a designated-contact grant; then act under a narrow emergency-services grant. *Disclosure*: the emergency path shares location and incident facts only. *On failure*: a false alarm is one tap to cancel at each step; missed detection remains a stated residual risk.

S10 — Health appointment brief.

Inputs: health store records and symptom notes the person recorded deliberately. *Retained*: the brief as a generated artifact with line-level provenance to its observations. *Autonomy*: draft; the person reviews before anything is shared. *Disclosure*: none by default; the brief reaches the clinician only by the person's explicit act. *On failure*: items with uncertain provenance stay flagged in place.

S11 — Travel partial failure.

Inputs: the itinerary and provider notifications. *Retained:* workflow state distinguishing committed from pending steps. *Autonomy:* act within a travel grant using preview and commit with declared compensation. *Disclosure:* each provider receives its own booking details only. *On failure:* a change that cancels but fails before rebooking records the committed step, invokes compensation if valid, blocks continuation when state diverges from the preview, and escalates with the remaining side effects.

S12 — Anomaly-aware bill review.

Inputs: recurring biller history and the current bill. *Retained:* payment commitments and local decision receipts. *Autonomy:* act to present the anomaly and its explanation on the person's own surface; never initiate payment. A new payee, a deviating amount, or a forged-request signal produces deny and a review prompt, because content found in the bill message never authorizes anything (Section 11). An ordinary bill also resolves to ask: the person explicitly approves or declines payment in the provider's own flow. *Disclosure:* an optional comparison tool receives the smallest scoped facts needed to assess the anomaly; it receives no payment credential. *On failure:* the system makes no transfer and asks the person to inspect the bill.

Two scenarios exceed the routine-task population that the evaluation plan defines, which excludes financial transfer, medical decision, credential change, lockout, life-safety control, and public publication (Section 13). S9's emergency escalation and S12's payment-approval path are therefore not covered by the routine prototype benchmark. They are evaluated only under the separate high-impact protocol of Section 13.6, whose job is to test whether such envelopes should exist at all rather than to fold them into routine-task averages.

4. Scope, Terms, and Design Goals

4.1. Terms

Throughout the paper, *the person* is the individual whom Mini Me serves and whose policy governs Continuum. We avoid the device-specific term “wearer” except when reporting head-mounted-camera literature, where it denotes the person wearing the capture device. *Mini Me* names the experience; *Continuum* names the proposed substrate. *Presence* is defined in Section 2. A *grant* is an explicit, revocable, expiring, signed authorization for a class of actions, and it is the only source of authority in the

system; its schema is given in Section 11.2. An *authority level* is one of L0 through L4, assigned per policy domain, capability, and impact class rather than globally (Section 11.3).

Two boundary terms are used throughout and are not interchangeable. A *device trust domain* is one enrolled device's execution and key domain: the code, storage, adapters, local models, and credentials that device holds, protected by its own hardware-backed key (Section 7.5). There is no single device-local domain. Every enrolled device has its own device trust domain, and shared-space hubs are enrolled as device trust domains at a lower trust level. The *personal trust perimeter* is the collective logical boundary between the person's enrolled device trust domains and everything outside them: remote processors, remote tools, collaboration services, optional backups, and human recipients. Minimization, disclosure bundles, and receipts apply at the perimeter; key wrapping, scope, conflict handling, and revocation apply within and between device trust domains. Where this paper writes *domain* without qualification it means a policy domain such as work or health, not a trust domain.

A *disclosure bundle* is the purpose-scoped context package that crosses the personal trust perimeter (Section 10). Every proposed action resolves to one of four *action outcomes*: act, draft, ask, or deny (Section 11).

Term	One-line meaning	Specified in
Presence	Measurable continuity, comprehension, anticipation, and authority under policy.	Section 2
Presence fabric / context root	Coordination layer and portable root that preserve a task thread across enrolled surfaces.	Sections 7.2 and 12
Device trust domain	One enrolled device's execution, storage, adapter, model, credential, and key boundary.	Section 7.5
Personal trust perimeter	Boundary between all enrolled device domains and remote processors, tools, services, backups, or recipients.	Sections 4 and 10
Policy / shared-space domain	A life-policy scope such as work or health; a shared-space domain is a lower-trust enrolled hub.	Sections 4 and 7.5
Cell	Policy domain, capability, and impact-class triple to which authority is assigned.	Section 11.3
Impact class	Presentation, internal, reversible, compensable, disclosive, physical, or irreversible effect taxonomy.	Section 11.3
Authority level / outcome	L0–L4 bounds the cell; act, draft, ask, or deny resolves one decision.	Sections 11.3 and 11.8
Grant	Signed, scoped, expiring, revocable authorization and the only source of action authority.	Section 11.2
Disclosure bundle / budget epoch	Purpose-scoped perimeter-crossing context; the epoch bounds its local cumulative-exposure history.	Section 10
Review / pending queue	Review holds unresolved inference; pending holds an authorized effect awaiting execution.	Sections 9 and 11.5
DG / G / L / S labels	DG1–DG6 design goals; G1–G4 action gates; L0–L4 levels; S1–S12 scenario probes.	Sections 4, 11.8, 11.3, and 3.2

Table 2. Compact glossary. The definitions remain normative only in the cited sections; this table is a reading aid.

4.2. Context Objects

Continuum represents each observation as

$$o_i = \langle s_i, t_i, \mu_i, d_i, q_i, \pi_i, r_i \rangle,$$

where s is source, t time interval, μ modality, d payload or derived representation, q confidence, π provenance, and r policy/retention metadata. A memory object m_j may cite one or more observations and must distinguish:

- *observed facts* (e.g., a calendar event exists),
- *reported facts* (the person stated a preference),
- *derived beliefs* (the system inferred a habit), and
- *generated artifacts* (a summary or planned action).

This distinction prevents a generated summary from silently becoming biographical fact.

4.3. Design Goals

DG1 — Personal authority.

The person can inspect, pause, delete, export, and restrict context flows. Authority is bounded by the host operating system and does not imply immunity to malware or compelled unlock.

DG2 — Useful local baseline.

Core recall, redaction, bundle assembly, and low-risk planning remain useful without a network connection.

DG3 — Minimized disclosure.

External processing receives a context subset selected to meet a declared task objective while minimizing measured semantic exposure.

DG4 — Traceable derivation.

Suggestions and actions cite the observations and memory objects used.

DG5 — Bounded action.

Consequential actions are gated by reversibility, externality, uncertainty, and explicit policy.

DG6 — Bystander protection.

Other people’s voices, faces, screens, and private spaces default to suppression, transformation, or short retention.

Design goals are cited as DG_n where relevant. The unprefix labels G1–G4 are reserved for the four action gates of Section 11.8.

4.4. *Non-Goals*

Continuum does not attempt covert recording, global surveillance, automatic monetization of personal data, proof of meaningful consent under coercion, or cryptographic prevention of every remote misuse. A phone may be the first local host, but the architecture is not Android- or iOS-specific.

5. Background and Related Work

The relevant work spans several strands that are usually studied separately. The abstract’s five-part composition claim is canonical; the comparator-specific deltas below isolate what Continuum adds to each neighboring system rather than redefine that claim.

5.1. *Lifelogging and Personal Information Management*

MyLifeBits explored the “personal database for everything,” exposing both the promise and the retrieval burden of comprehensive capture ^[1]. Sellen and Whittaker argued that indiscriminate total capture often fails to support the functions people actually need from memory ^[2]. Continuum inherits the ambition but not the retention assumption: raw media is an input class, not the permanent canonical record.

5.2. *Contextual Integrity and Mobile Sensing*

Contextual integrity defines privacy through appropriate information flows among actors, attributes, and transmission principles ^{[3][4]}. This is more useful for Continuum than a simple local/cloud binary. The mobile-sensing literature already showed that phones combine rich sensors, local inference, app-distribution scale, and cloud offload ^[5]; Continuum’s delta is not to discover sensors on phones, but to treat cross-device continuity, retention, and disclosure as first-class policy objects rather than application-local design choices. Local collection can still violate privacy through covert sensing, coercive access, or poor bystander controls ^{[6][7]}. Camera-glasses studies report wearer–bystander preference gaps and weak notice from status indicators ^{[8][9]}. The Policy plane can enforce the person’s collection and query rules, but it cannot by itself create meaningful bystander consent.

5.3. Context Middleware and Personal Data Stores

The Context Toolkit separated sensor acquisition and representation from application reaction through context widgets, aggregators, and interpreters ^[10]. Continuum’s plane separation inherits this ubiquitous-computing lineage; the LLM is not the origin of context-aware architecture. Solid personal data pods decouple storage from applications through user-controlled identity and access rules ^[11]. Databox is closer still: it collects local and remote personal/IoT data, runs computation near the data, mediates limited access, logs reads and writes, and supports actuation ^{[12][13]}. Continuum does not claim to invent the personal vault. Its narrower proposal is to connect a vault to typed memory derivation, model placement, minimized task disclosure, and risk-gated agent actions under one policy scheduler.

5.4. Purpose and Usage Control

Purpose-based access control extends identity- or role-based decisions with the declared processing purpose, and prior work formalizes purpose compatibility and normative reasoning ^[14]. Macaroons demonstrate attenuating credentials that can narrow delegated authority with contextual caveats ^[15]. Continuum borrows the monotone-narrowing idea but applies it to personal action and disclosure grants that must also carry expiry, revocation, audit linkage, and policy-domain meaning for humans. “Sticky” usage terms and accountable distributed processing similarly attempt to carry obligations beyond initial access. Continuum can enforce purpose while data remains inside the personal trust perimeter. Outside that perimeter, purpose metadata and receipts improve accountability but do not provide usage control against a dishonest processor.

5.5. Egocentric and Ambient Assistants

EgoLife combines continuous egocentric capture with long-horizon life question answering ^[16]. Vinci and Vinci2 develop real-time and proactive assistance over continuous video ^{[17][18]}. ContextAgent reasons over open-world sensory context and invokes tools ^[19]; VisionClaw connects wearable perception to executable agents ^[20]. These systems demonstrate that physical-world context can improve assistance. Continuum focuses on the system boundary around that capability: multi-device capture, retention, bystander policy, and consequential action.

5.6. Personal Memory and Identity Models

Second Me frames personal memory as a persistent, locally deployable identity model ^[21]. MemGPT reframes agent memory as an operating-system-style paging problem over context and external stores ^[22]. Microsoft Recall turns retrospective screenshot search into an OS feature on one device, and its rollout itself became informative: Microsoft announced it in May 2024, shifted the planned broad June preview to an Insider-only rollout after security feedback, then redesigned it around opt-in enrollment, Windows Hello gating, and VBS Enclave isolation before later general availability of the preview on Copilot+ PCs ^{[23][24][25][26]}. Continuum’s delta is narrower and stricter: memory is not just retrieval over screenshots or context-window extension, but a typed, cross-device, policy-scoped record with provenance, expiry, correction, and action consequences. Continuum treats identity inference as one output of the Memory plane, not as ground truth: inferred preferences require provenance, confidence, correction, and expiry.

5.7. Adjacent Personal-Agent Products and Private Compute

OpenClaw is an open-source, self-hosted gateway that connects messaging channels to tool-using agents, sessions, configurable memory, local or remote models, plugins, tools, and paired mobile nodes ^[27]. It is relevant to Continuum’s always-available agent surface, but its documented architecture does not claim a person-directed multi-modal capture perimeter, purpose-scoped disclosure compiler, or one content-independent authority contract across providers. Rewind/Limitless-class products demonstrated retrospective and ambient memory. Limitless was acquired by Meta in December 2025; new Pendant sales stopped and Rewind capture was sunset, so it is treated here as a product-history and platform-concentration case rather than a current baseline ^[28]. Its published Pendant guidance remains a useful comparison for recording notice: it requires users to notify and obtain consent from people they record, and says that its visible LED does not replace that obligation ^[29]. These materials demonstrate useful fragments—agent availability, retrospective recall, and recording notice—not the proposed composition in this paper.

Apple Private Cloud Compute (PCC) is a distinct adjacent system, not a Continuum component. Apple describes PCC as cloud processing for Apple Intelligence with stated stateless-computation, enforceable-guarantee, no-privileged-runtime-access, non-targetability, and verifiable-transparency requirements ^[30]. Those properties make PCC a useful example of a remote-processing boundary with a

stronger provider-specific assurance story than an ordinary API. They do not supply cross-vendor personal memory, purpose-scoped disclosure compilation, or authority over third-party actions. Conversely, Continuum proposes no PCC-equivalent hardware, attestation system, or proof of remote forgetting; it can consume such provider evidence through an adapter only where it is independently available.

5.8. Mobile Agents and Proactivity

The mixed-initiative literature predates current agent systems. Horvitz argued that useful automation should couple direct manipulation with uncertainty-aware assistance, timing, and user correction rather than replace the interface wholesale ^[31]. BusyBody then learned personalized costs of interruption from desktop, meeting, location, and conversation context ^[32]. Parasuraman et al. organized automation by function and level, emphasizing that different functions may warrant different autonomy levels ^[33]. Mobile-agent benchmarks historically emphasize executing explicit GUI instructions. In the cited 2024 and 2025 results, AndroidWorld’s original strongest baseline completed 30.6% of its dynamic tasks and V-Droid reported 59.5%; later systems may move that frontier, but no benchmark task-success percentage by itself establishes safety for unattended high-impact action ^{[34][35]}. FingerTip 20K adds personalized intent prediction from historical behavior and current screenshots; its reported gap between model and human performance shows that proactivity remains immature ^[36]. ProactiveMobile requires agents to infer intent from profile, device state, world state, and behavioral trajectories and translate it into executable functions ^[37]. KnowU-Bench adds interactive preference elicitation and the decision to intervene, ask, or remain silent ^[38]. Continuum’s delta is to make these choices per cell, tie them to grants and receipts, and extend them from notification timing into perimeter-crossing disclosure and physical action.

5.9. Platform-Managed Models and App Capabilities

Android AICore manages deployment and execution of an on-device foundation model for application APIs ^[39]. Android AppFunctions lets applications expose discoverable, permission-gated functions as local tools for agents; the cited Android Developers page describes the feature as an experimental preview and says Gemini integration is in a private preview with trusted testers ^[40]. Apple’s Foundation Models framework exposes system-managed on-device and Private Cloud Compute models, while App Intents exposes application entities and actions to system intelligence ^{[41][42]}. These are platform-managed

personal-agent building blocks, not evidence that they provide a portable personal memory or common action authority. Health Connect demonstrates an adjacent model: a system-managed personal data store with typed records and centralized permissions ^[43]. Earlier mobile systems explored offloading computation and staging data on nearby untrusted surrogates through cyber-foraging ^[44]. Continuum inherits the placement question but changes the objective: placement is selected by sensitivity, disclosure budget, and policy as well as by performance and energy.

These systems and products support parts of Continuum's thesis. They do not yet provide a portable, person-controlled memory across modalities and vendors, nor a common disclosure-bundle or action-receipt contract. In shared spaces, Kratos+ shows that smart-home control is already a multi-user conflict-resolution problem rather than a single-owner preference problem ^[45]; Continuum's narrow proposal is to compose those occupant and priority conflicts with typed memory, purpose-scoped disclosure, cumulative modeled exposure, and content-independent authority under one policy scheduler.

System class	Capture	Long memory	Proactive act	User policy boundary	Primary gap relative to Continuum
Classical lifelogging	Strong	Retrieval-oriented	Limited	Limited	LLM-era reasoning and action safety
Egocentric assistants	Video/audio	Session-week	Emerging	Usually study-specific	Multi-device ownership and bystanders
Personal memory models	Imported data	Strong	Limited	Often local	Continuous sensing and action contracts
Mobile GUI agents	Screen/UI	Task history	Emerging	App/OS permission	Cross-device lifelong context
Platform AI services	App/screen context	Platform-scoped	Strengthening	Vendor-controlled	Portable policy-controlled vault
Personal data stores	IoT / linked data	Strong	Databox actuation	User access rules	Typed AI memory and risk-gated action
Continuum (proposed)	Proposed multi-device	Proposed layered/typed	Proposed risk-gated	Person-directed policy scheduler	Research agenda; not yet implemented

Table 3. Related systems advance individual parts of persistent personal intelligence. Continuum proposes composition under one person-directed policy scheduler; it does not claim novelty in every component or complete control over the host OS.

6. Threat and Privacy Model

6.1. Trust Domains and the Personal Perimeter

Continuum has no single device-local trust domain. Each enrolled device hosts its own *device trust domain* with its own keys, and one device is the context root that holds the vault and the policy scheduler (Section 7.5). Inside a device trust domain sit that device’s trusted ingestion adapters, its encrypted

storage, its local models, and its local action credentials. Unenrolled sensors and source devices belong to no device trust domain and are treated as authenticated but potentially faulty inputs; shared-space hubs are enrolled as device trust domains at a lower trust level and hold no vault plaintext. The *personal trust perimeter* is the collective boundary around these domains: remote processors, remote tools, collaboration services, optional backups, and human recipients all sit outside it, and every enforcement claim about disclosure is a claim about that perimeter rather than about any one device. The host OS is *trusted for practical execution but not assumed invisible*: it controls sensors, permissions, backups, and process isolation.

6.2. Principals, Assets, and Enforcement Points

Principals are the person (primary operator), recorded participants and bystanders, source-device owners, application/capability providers, model processors, and auditors. **Assets** include raw signals, derived memories, policy state, vault/model keys, tool credentials, action previews and commits, receipts, and audit history. Enforcement occurs at sensor enrollment, observation ingestion, memory promotion/query, perimeter-crossing disclosure, credential minting, and action commit. No enforcement point exists after plaintext reaches a dishonest remote processor.

6.3. Adversaries

- *Compromised endpoint*: malware or an unlocked device can access plaintext during use.
- *Curious remote processor*: retains or repurposes a legitimate disclosure bundle.
- *Environmental injector*: embeds instructions in web pages, documents, speech, QR codes, or physical signs.
- *Malicious capability provider*: misstates side effects, changes state between preview and commit, exfiltrates credentials, lies about final state, or presents compensation as reversal.
- *Confused deputy*: induces use of a legitimate capability for an unauthorized principal or purpose.
- *Coercive institution*: employer, insurer, school, or state conditions access on vault disclosure.
- *Intimate partner or family abuser*: uses shared devices, forced unlock, or hidden sensors.
- *Malicious user*: intentionally captures or acts against bystanders.
- *Supply-chain or model adversary*: compromises an adapter, model update, capability schema, or policy bundle.
- *Colluding services*: combine individually permitted disclosures to reconstruct excluded attributes.

- *Platform gatekeeper*: restricts background sensing, app interoperability, or model choice.

6.4. Security and Privacy Goals

- Vault confidentiality at rest under explicit endpoint assumptions.
- Visible capture state and attributable source adapters.
- Data minimization at the personal trust perimeter.
- Integrity and replay resistance for action and remote-processing receipts.
- Scoped, short-lived credentials for Act-plane tools.
- Preview/commit binding, commit-time precondition checks, and durable state for multi-step workflows.
- Append-only, integrity-protected local audit history with rollback detection.
- Revocation that prevents future access; no false promise to erase knowledge already disclosed.

6.5. Security Non-Guarantees

Remote forgetting. A purpose label cannot force a dishonest processor to delete data. **Complete provenance.** A malicious adapter can omit events. **Bystander consent.** Detection, redaction, and notice reduce harm but do not manufacture consent. **Trusted time and receipts.** Device clocks, local receipts, and provider statements are evidence with declared clock and signer assumptions, not universal truth or proof of non-use. **Recovery, rollback, and replicas.** A lost root, an older replica, or an offline device can retain stale state; bounded expiry, revocation refresh, and authenticated local log heads reduce but do not eliminate that risk. **Local equals private.** Covert local sensing and coercive local access remain privacy violations. **Correct identity inference.** Long-term models can drift or amplify wrong assumptions.

Threat	Mitigation direction	Residual risk
Endpoint compromise	Hardware-backed keys, OS isolation, remote wipe	Plaintext exists during use
Remote reuse and collusion	Minimal bundles, sequential exposure budgets, attestation where available, receipts	Cannot force forgetting; independent lawful disclosures may still compose off-device
Prompt injection	Untrusted-input labels, tool allowlists, preview binding, four gates	Novel attacks remain
Bystander capture	Edge redaction, visible state, non-retention, policy pauses	False negatives, weak notice, and legal variance remain
Coercion and intimate-partner abuse	Local default, pause controls, secondary-space research, no fail-open grants	Social power defeats UI and may defeat device control
Malicious providers	Capability metadata, scoped credentials, commit-time checks, compensation declarations	Provider can still lie, stall, or misreport post-commit state
Supply chain	Signed updates, attestation where available, provenance on adapters and bundles	Attestation coverage is partial and upstream compromise remains possible
Memory poisoning	Provenance, confidence, correction, contradiction checks	Slow drift is hard to detect
Partitioned verifier	Short grant expiry, renewal, revocation lists with staleness bounds	A stale verifier may accept an older unexpired epoch until it resynchronizes
Audit-log retention	Integrity-protected local audit with scoped access and expiry policy	The log itself reveals routines and relationships if retained too broadly or seized
Grant and revocation sensitivity	Per-domain encryption, scoped replication, short lifetimes, minimal fields on replicas	Payees, policy domains, ceilings, and revocation timing still reveal routines across devices
Platform gatekeeping	Portability, schema export, local-first fallbacks	Dominant OS vendors may still block viable deployment paths

Table 4. Threats, mitigations, and residual risks.

7. Continuum Architecture

Continuum’s job is to make the day in Section 3 implementable within the threat model of Section 6. The architecture separates three unlike kinds of responsibility rather than presenting one uniform stack of equivalent planes. Keeping devices enrolled, healthy, and reachable is not the same work as normalizing an observation or committing an action (Figure 3):

- a *presence fabric* (Section 7.2) that enrolls devices, inventories what each source can sense and do, tracks freshness, carries a task across devices, arbitrates which surface receives output, and degrades predictably when sources are missing;
- four *data-path planes*—Capture, Memory, Resolve, and Act—that carry a signal from a sensor to a retained memory, an answer, or a committed action;
- one *Policy control plane* that governs sensing, retention, disclosure, delivery, and action, and that the fabric and every data-path plane must consult.

The fabric spans device trust domains without being one of them, and it is not a trust boundary in its own right. Enrolled devices remain separate device trust domains that the fabric coordinates inside one personal trust perimeter (Section 7.5), so a compromised watch, a borrowed laptop, or a shared home hub does not inherit the vault.

7.1. Actors and Structure

The person is the primary operator, but not the sole affected principal. Bystanders and participants are data subjects whose preferences may conflict with the person’s. Context sources include phones, laptops, body-worn sensors, vehicles, rooms, clinical or physiological devices, and other explicitly paired sources. Applications and services expose data and capabilities through adapters. Remote models receive purpose-scoped bundles. Tool providers execute actions with scoped credentials. One enrolled device is the *context root*: it holds the vault, the policy state, and the fabric’s registry, and in the first instance it is a phone (Section 12).

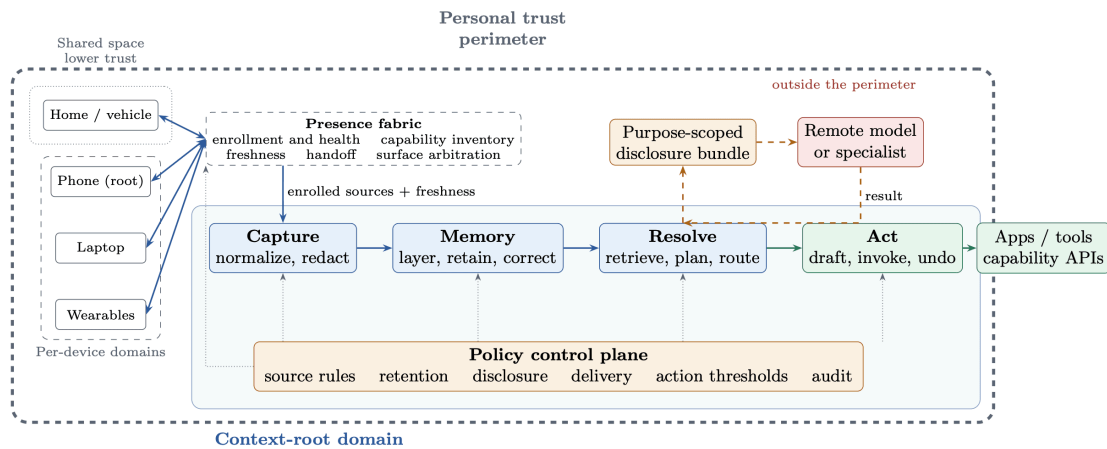


Figure 3. Proposed Continuum architecture. The presence fabric admits enrolled sources and arbitrates delivery surfaces across device trust domains without becoming one. The Policy control plane governs the fabric and all four data-path planes. The outer dashed boundary is the single personal trust perimeter; remote processing crosses it only through a purpose-scoped disclosure bundle. Capability tools remain outside that perimeter under scoped credentials, and bystander rights remain an external constraint rather than a grant from policy.

7.2. The Presence Fabric

The fabric holds no memory of its own. Its registry is policy state, and every entry in it is inspectable, revocable, and expiring.

Device enrollment and health. Enrollment is explicit and produces a record: device identity, a hardware-backed domain key, the sensors and actuators the device declares, the host-OS permissions actually granted, a trust level, attestation state where the platform supports it, and a revocation reference. Nothing that has not been enrolled may become a source. Each enrolled device reports health on a schedule and on change: reachable or not, sensing paused by the person or by the OS, permission revoked, thermally throttled, battery-constrained, storage-constrained, or clock-skewed beyond tolerance. An unhealthy source is marked unavailable rather than silently substituted by a healthier one, because a substitution the person cannot see is a coverage claim the system cannot support.

Source capability inventory. For each enrolled device the fabric records the modalities it can capture, the actions it can perform, its local model capacity class, the output surfaces it offers, and the grants currently covering each. The inventory is what makes continuity measurable: the coverage in Section 2 is

computed against the enrolled and permitted inventory, not against everything the hardware could reach. A device that could sense a modality but has no grant for it appears in the inventory as capable and not permitted, which is a different state from absent.

Attribute freshness and staleness. Every attribute the fabric publishes carries the time it was observed, a freshness class with an explicit expiry, and a disposition for what happens when it expires: suppress the attribute, present it with its age, or re-observe before use. Consumers must therefore handle three states rather than two, namely fresh, stale with a stated age, and absent. A stale attribute is never presented as current, and Resolve states the resulting unknowns instead of interpolating them (S7).

Cross-device handoff. The unit of continuity is a task thread: a goal, references to its working context, the citations it has accumulated, its autonomy state, its pending approvals, and the idempotency keys of any steps already committed. Handoff moves the thread, not the vault. The receiving device resolves references against what its own domain policy permits, so a thread that reads freely on the context root may render with omissions on a shared screen. A thread's autonomy state never increases on handoff; a step that required asking on the phone still requires asking on the laptop.

Output-surface arbitration. Given something to deliver, the fabric ranks the available surfaces by the person's presence at each, the privacy of the surface, that is, who else can currently see or hear it, the modality fit for the content, and the sensitivity label the content carries. An item whose audience cannot be bounded on any presently available surface waits rather than degrading to a louder one. Whether to interrupt at all is an action-policy decision (Section 11); the fabric decides where and how, not whether.

Offline and degraded behavior. Degradation is ranked and declared rather than emergent.

1. *Network-partitioned.* Recall, redaction, bundle assembly, and low-risk planning continue locally per DG2. Work that requires a remote processor is queued as a draft and never auto-committed on reconnection.
2. *Source-degraded.* A missing modality is recorded as a coverage gap, dependent inferences lose confidence rather than keeping their last value, and next actions whose preconditions can no longer be checked are held (Section 9).
3. *Root-unavailable.* Other devices keep read-only cached views under their own domain policy and may act only inside grants already issued to them. They cannot mint credentials, promote memory, or widen a disclosure. Grants expire rather than failing open.

Recovery is explicit: when a source returns, its gap remains recorded, and any inference that spans the gap says so.

7.3. Device State Model

Continuity across devices requires a typed model of what each device is currently doing, not a heuristic read of the foreground. Every field in Table 5 is typed, carries the freshness envelope of Section 7.2, and is separately grantable; device state is context like any other and is subject to the same retention and disclosure rules. Fields are published in the following envelope, so a consumer can always tell how old a value is and under which grant it was read:

```
{
  "protocolVersion": "0.1",
  "schemaVersion": "0.1",
  "device": "device:laptop-2",
  "domain": "domain:laptop-2",
  "field": "foreground_application",
  "value": {"package": "app:editor", "activity": "edit"},
  "confidence": 0.94,
  "observedAt": "2026-08-06T14:02:11Z",
  "staleAfter": "2026-08-06T14:02:41Z",
  "onStale": "present_with_age",
  "grant": "grant:work-continuity-3",
  "sensitivity": ["app_identity"]
}
```

Listing 1. Device state field envelope. Every field carries its own freshness and grant.

Two fields need separate treatment. *Screen semantics* is a structured summary of the current screen—its fields, entities, and available actions—and not a durable screenshot; frames are not retained by default, and the summary inherits the sensitivity of the content it describes. *Clipboard* access is gated separately from all other device state and is never ambient: the fabric may publish that a clipboard entry exists and its coarse type, while contents require an explicit, expiring, purpose-scoped grant and produce a receipt on every read. Clipboards routinely carry passwords, one-time codes, and other people’s messages, so an ambient clipboard reader would defeat the rest of the model.

Field	Typed value	Representative use	Freshness and gating
Power	Charge fraction, charging, saver mode	Defer capture; schedule local inference	Seconds to minutes; low sensitivity
Thermal state	Nominal, fair, serious, critical	Throttle continuous perception; move work to another device	Seconds; low sensitivity
Connectivity	Offline, metered, unmetered; roaming	Choose local or remote processing; queue disclosure	Seconds; low sensitivity
Focus	Available, focus, do-not-disturb, driving, with an until time	Delivery timing input to the action policy	Minutes; reveals routine
Foreground application	Package identity and coarse activity class	Task-thread continuity; capability discovery	Seconds; app identity is sensitive and gated per domain
Notifications	Counts and coarse categories per domain; contents only under grant	Surface arbitration; interruption budget	Seconds; contents high sensitivity
Media	Playing or paused; coarse output route	Surface arbitration; who else can hear	Seconds; titles gated separately
Attention mode	Derived: in conversation, driving, exercising, screen-focused, asleep, with confidence	Whether an ambient surface is acceptable at all	Seconds to minutes; derived belief, correctable
Permission inventory	Per source: granted, denied, restricted, revoked, with OS scope	Honest coverage accounting; no silent substitution	Event-driven; re-read rather than cached across sessions
Screen semantics	Structured fields, entities, and actions of the current screen	Continuation and cross-app tasks	Seconds; treated as content, no durable frames
Available sensors	Enrolled sensors reporting now, with rate and quality	Coverage and degradation decisions	Seconds; low sensitivity
Clipboard	Presence and coarse type; contents only under an expiring purpose-	Pasting into a task the person initiated	Per read, with a receipt; never ambient

Field	Typed value	Representative use	Freshness and gating
	scoped grant		

Table 5. Device State Model. Every field is typed, carries the freshness envelope of Listing 1, and is separately grantable. Clipboard contents are gated apart from the rest of device state.

7.4. Plane Contracts

Capture. Ingests signals from enrolled sources under a source-specific grant. Emits normalized observations with provenance, confidence, subject/bystander labels, and retention class. Raw media may be transformed locally into transcripts, embeddings, object/event summaries, or redacted clips before expiry.

Memory. Maintains the layered memory of Section 9—working context, episodic records, semantic memory, and identity or preferences—together with the commitments and next actions derived from it. Promotion between layers is a policy event, not an invisible side effect.

Resolve. Converts the person’s goal or a proactive trigger into a context query, retrieves candidate memories, estimates sufficiency, ranks next-action candidates, and selects local processing, remote processing, clarification, or refusal. A candidate it declines to surface is held or queued for review rather than discarded silently (Section 9).

Act. Converts an authorized action intent into a capability call under a named grant and at the authority level of that grant’s cell (Section 11). It requires idempotency keys, precondition checks, preview for consequential changes, structured outcome/error reporting, and a reversal path that the substrate supplies where the provider does not.

Policy. Evaluates source, subject, declared purpose, domain, recipient, time, sensitivity, action risk, and the person’s rules. It emits allow, transform, ask, or deny decisions and records receipts. It can enforce authorization; it cannot infer that every recorded person consented.

Table 6 states what each component consumes, what it emits, and the invariant it must not violate, which is what makes them separately testable.

Policy conflicts are resolved in a fail-closed order: mandatory safety and applicable legal constraints; represented affected-principal constraints; the person’s grants and denials; then task-level requests. A

lower-precedence grant cannot override a higher-precedence denial. Because affected principals may be unknown or unable to express policy, this ordering reduces but does not solve bystander governance.

Component	Consumes	Emits	Invariant
Fabric	Enrollment records + device health + grants	Permitted source inventory, freshness, surface ranking, task-thread handoff	Nothing unenrolled becomes a source; staleness travels with every attribute
Capture	Sensor/app signal + grant	Observation + provenance + retention	No silent source; redact before durable storage when feasible
Memory	Observations + corrections	Scoped memory view + uncertainty + commitments	Generated summaries never become unmarked facts
Resolve	Goal + memory view + policy	Plan, question, next-action candidates, disclosure request, refusal	State unknowns; minimize selected context
Act	Authorized intent + grant + scoped credential	Outcome + side effects + receipt	Draft is not commit; authority never rises on retry, handoff, or failure to reach the person
Policy	Rules + sensitivity + risk + audit event	Allow / transform / ask / deny	Scope, expiry, and revocation apply across all planes

Table 6. The presence fabric and the four data-path planes are independently testable and communicate through explicit objects. Policy is the control plane that every other row consults.

7.5. Distributed Trust Domains and Synchronization

Treating every device the person owns as one trust domain would make the weakest device the security of the whole system. Continuum instead gives each enrolled device its own device trust domain and synchronizes between them under explicit scope, with the personal trust perimeter drawn around the set rather than around any one device.

Per-device key domains. At enrollment a device generates a domain key in hardware; the private half never leaves it. Vault objects are encrypted individually under content keys held by the context root,

whose own key material is protected by device hardware and by the person's authentication. No other device holds a master key, so compromising a wearable yields that wearable's domain and whatever was wrapped for it, not the vault.

Scoped wrapped keys. A device receives content keys wrapped for its domain, and each wrapping record states the object or object class it covers, the receiving domain, the permitted operations, the expiry, the grant that justified it, and a revocation reference. Permitted operations are narrower than read or write: a watch that may present a derived readiness state receives no wrap for the underlying physiology, and a shared display may be permitted to present an artifact without being permitted to export it (S5, S6).

Bounded revocation propagation. Revocation cannot be assumed instantaneous on devices that are asleep, offline, or stolen, so it is bounded three ways instead of promised absolutely. Wrapped keys are short-lived, so an unreachable device loses access by expiry. Every synchronization carries the current revocation list with a maximum tolerated staleness, past which a device refuses privileged operations until it can refresh. Affected object classes are re-wrapped under fresh content keys, so an old wrap cannot be replayed against future data. The guarantee is the one already stated in Section 6: revocation prevents future access within a bounded window, and it does not recover plaintext that was already read.

Conflict handling. Synchronization is typed per object class rather than last-writer-wins over a vault. Observations are append-only and never merge, because two devices that sensed the same moment produced two observations. Episodes merge as a union of evidence. Semantic facts and identity or preference objects carry an assertion counter and the asserting domain; concurrent incompatible assertions produce a recorded contradiction rather than a silent winner. Commitments and next actions merge by supersession (Section 9). Deletions dominate: a tombstone wins against any concurrent update and propagates to objects derived from what was deleted.

The shared-space domain. Home hubs, vehicles, and shared displays are enrolled at a lower trust level. They hold no vault plaintext, receive commands and short-lived scoped credentials rather than context, keep their own occupancy and device state locally with short retention, and maintain attestation and revocation state independently of the person's devices. A shared-space domain is a semi-trusted sensor and actuator, never a replica of the vault (S8), and it cannot originate an action on the person's authority (Section 11).

Residual leakage. Even with encrypted payloads, synchronization metadata reveals which object classes changed and when, which is an activity trace. Continuum proposes batching and padding

synchronization at a policy-declared granularity to blunt this, and states the remainder as residual risk rather than a solved problem.

8. Perception Across Eyes, Body, Devices, and Spaces

The perception principle stated in Section 2 governs this section: *sense within enrolled policy, understand locally, remember deliberately*. The three clauses constrain one another. Policy-permitted sensing is what makes presence possible; local understanding is what keeps sensing from becoming an archive somebody else can query; deliberate retention is what makes the result something the person can inspect, correct, and delete. Each modality must therefore say where its processing sits and what happens to its raw signal by default.

8.1. Modality Inventory

Table 7 states, for each sensing class, a representative use, its sensitivity, where processing is placed, and the default fate of the raw signal. Two conventions run through it. Physiological and location signals stay in their own stores and reach working context only as derived states with expiry. Anything covering other people inherits the strictest retention among its inputs.

The table states placement and retention; it deliberately states no cost. Continuous video and audio understanding is the most demanding class here, and whether a given device class can sustain it within thermal and battery limits is an open empirical question that the Placement criterion in Section 13 treats with thermal and energy envelopes intended for preregistration. This paper reports no measurements of energy, latency, or thermal behavior.

Modality	Representative use	Sensitivity	Processing placement	Default retention
First-person video	Read a panel, a whiteboard, a shelf; object and scene relations	Very high: bystanders, homes, screens, documents	On the capturing device; the buffer never leaves its domain	Buffer expires; typed events retained only after gate and redaction
Audio	Decisions, commitments, requests during declared capture	Very high: other people's speech and content	On-device transcription and extraction	Transcript transient; extractions retained; raw audio expires
Screens	Task continuation, cross-app work, entity extraction	High: content, and app identity itself	On-device structured summary	Summary short-lived; no durable frames
Motion and IMU	Activity and exercise state, impact signals, attention mode	Medium: continuous motion reveals routine	Windowed features on the sensing device	Features and derived states retained; raw streams expire
Location	Arrival estimates, place-scoped policy, emergency facts	High: history is re-identifying	On-device; place labels preferred to coordinates	Coarse place visits retained; continuous traces expire quickly
Proximity	Handoff, surface arbitration, co-presence of others	Medium: discloses who else is present	On the sensing device	Presence flags with short expiry; identity only for enrolled devices
Ambient signals	Room state, occupancy, comfort envelopes	Low alone; occupancy inference raises it	On the space's own hub, in the shared-space domain	Short retention on the hub; only admitted aggregates enter the vault
Heart rate and HRV	Recovery and readiness, training load	High: employment and insurance consequences	On-device; series stay in the health store	Health store retains series; only a derived state with expiry is published
Sleep	Schedule adaptation, readiness	High: also reveals household patterns	On-device	Derived readiness with expiry; reasons never

Modality	Representative use	Sensitivity	Processing placement	Default retention
				disclosed outbound (S6)
Paired home sensors	Occupancy-aware preparation, appliance and inventory state	Medium to high: covers occupants and guests	On the hub, in the lower-trust shared- space domain	Short retention on the hub; vault receives only permitted states

Table 7. Modality inventory. Processing placement and default retention are design commitments; the table reports no measured cost, and continuous vision in particular remains an open feasibility question (Section 13).

8.2. First-Person Vision Pipeline

First-person video is the modality that most needs a specified path from sensor to memory, because it is the one whose raw form is least defensible to retain. The pipeline is fixed (Figure 4):

```
sensor -> encrypted rolling buffer -> local scene parser
        -> attention gate -> event/object relation extraction
        -> edge redaction -> policy-driven retention or expiry
```

Listing 2. The first-person vision path. Only the last stage can produce anything durable.

Sensor. Capture is a declared, revocable source under a grant, with capture state visible on the device. The sensor reports into its own device domain and nowhere else.

Encrypted rolling buffer. A fixed-duration buffer, encrypted under the device domain key and overwritten in place. Expiry is the default outcome for everything in it. The buffer is not addressable by Resolve: no query, local or remote, can reach raw frames, only the derivatives that survive the later stages.

Local scene parser. Within the device domain, frames become structure: objects, text, layout, and coarse activity. What leaves this stage is frame-level structure, not frames.

Attention gate. The gate decides which moments deserve extraction at all, from an explicit request, gaze and dwell, relevance to an open commitment, and the declared policy of the place. It is closed by default: with no positive signal, nothing is promoted. This is the stage that implements “remember deliberately”,

and it is the reason the dishwasher panel in S4 becomes an error code while the rest of the kitchen does not become anything.

Event and object-relation extraction. Gated moments become typed events and relations over objects, states, places, and times, each with confidence and provenance to a buffer segment that is itself expiring.

Edge redaction. Before anything durable exists, faces and bodies of unenrolled people, other people's screens, documents not addressed to the person, and regions a place's policy forbids are suppressed or transformed on the device that captured them. A false negative here is unrecoverable, which is why Section 13 requires a published false-negative budget rather than a claim of privacy preservation.

Unknown bystanders and policy-driven retention or expiry. Unknown people are treated as protected bystanders, not as unlabelled background. When a capture path cannot establish that its bystander policy admits durable retention or perimeter crossing, the safe default is to suppress durable promotion and remote disclosure; a person may explicitly narrow a capture mode, but the system may not interpret detection failure, weak notice, or silence as consent. Technical redaction can reduce accidental retention but cannot manufacture consent, prove lawful capture, or cure a false negative. The retention class follows from the place's policy, the subject labels on the observation, its sensitivity, and whether it supports an open commitment. There are exactly two outcomes: promotion to a typed episode with provenance, or expiry. Expiry is recorded as a coverage gap, so what was deliberately not kept remains visible as a boundary rather than as an absence the system might later mistake for evidence.

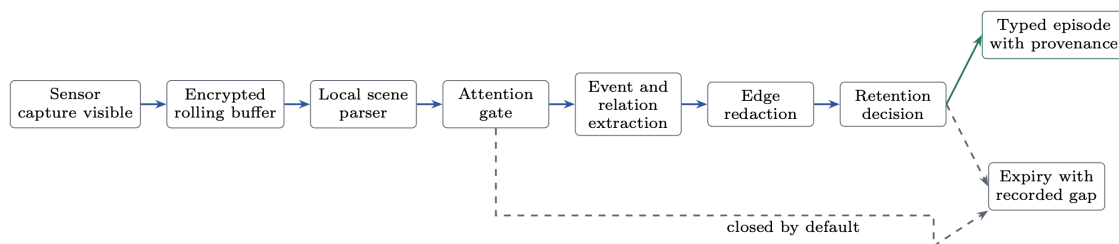


Figure 4. The first-person vision path: sense within enrolled policy, understand locally, remember deliberately. The rolling buffer is not addressable by Resolve, the attention gate is closed unless a signal opens it, redaction runs before anything durable exists, and every extracted event carries confidence and provenance forward (Figure 5 continues that chain into commitments). Expiry is the default outcome and is itself recorded as a coverage gap.

8.3. Multimodal State Estimation

Fusion across modalities produces a state estimate in which every field carries its confidence, its contributing modalities, and its freshness. Five rules govern it.

1. *Uncertainty propagates.* A derived state is never more confident than the weakest input it needs.
2. *Staleness propagates.* An estimate is no fresher than the oldest input it needs, and it expires when that input does.
3. *Disagreement is preserved.* Conflicting modalities yield a low-confidence state that retains both readings, rather than an average that hides the conflict.
4. *Absence is a value.* A missing modality yields unknown, never a default, and unknown is reported as such.
5. *Fusion respects domains.* Combining signals held in different device domains requires a wrapped key scoped to that purpose (Section 7.5); fusion is not a way around scope.

Consumers must handle unknown and stale explicitly. A next action whose precondition rests on an unknown state is held rather than attempted (Section 9), and the action policy may not read unknown as permission (Section 11). Subject and bystander labels are assigned at capture and travel with every derivative, so a state estimate that depended on a bystander-labeled observation inherits that observation's retention limit.

9. Memory, Commitments, and Anticipated Next Actions

Memory is where perception becomes something the person can be helped with. Its structure is layered and its retention is deliberate, and on top of it sit the two object types that anticipation actually requires: what the person owes or is owed, and what could usefully happen next.

9.1. Layered Memory and Provenance

Memory maintains four layers: *working context* (minutes to hours), *episodic records* (events with provenance), *semantic memory* (facts and relationships), and *identity and preferences* (slow-changing and correction-sensitive). Promotion between layers is a policy event rather than an invisible side effect (Figure 5). The underlying structure is a provenance DAG, not a hierarchy: an episode may cite many observations, a semantic fact many episodes, and a generated artifact both, which is what makes line-level provenance in S10 possible and what makes deletion propagate.

Three typed stores sit alongside the general layers because the scenarios depend on them behaving differently. The *health store* holds physiology at rest and publishes only derived states with expiry. The *space and object inventory*, which the day calls home inventory memory, holds enrolled places, rooms, appliances, models, and consumables; entries arrive by deliberate enrollment or by an observation the person confirmed, each recorded as an observed or reported fact with provenance and open to correction, and never as a silent inference from ambient video (S4). The *task-thread store* holds the threads the fabric hands between devices (Section 7.2).

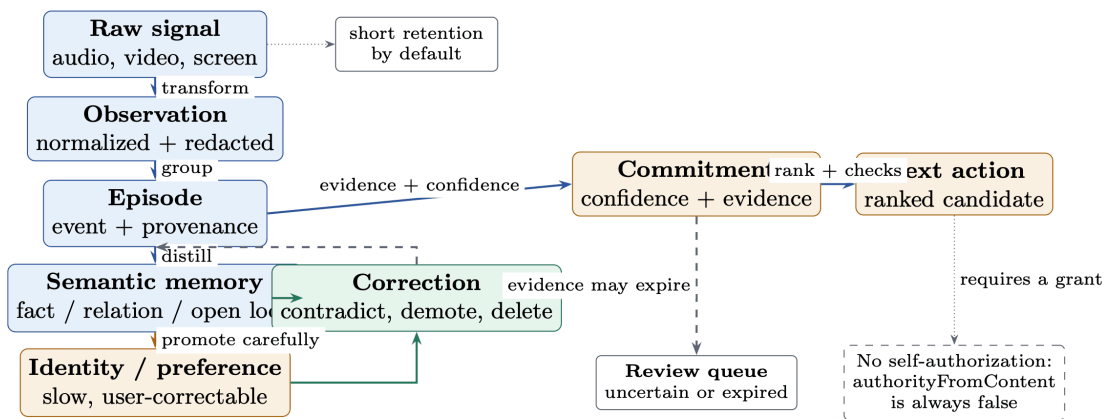


Figure 5. Default memory promotion path (left), with explicit contradiction, demotion, and deletion edges, feeding the commitment-to-next-action chain (right). The underlying provenance structure is a DAG, not a one-way hierarchy. Commitments and next actions carry confidence, evidence, and provenance and are never self-authorizing: captured content produces a ranked candidate, never a grant, so resolution continues only through the Four Gates of Section 11.8.

9.2. Commitments

A *commitment* is a typed obligation the person owes or is owed. It is a memory object, so it carries provenance, confidence, and retention like any other, and it adds the fields that make an obligation actionable.

```
{
  "protocolVersion": "0.1",
  "schemaVersion": "0.1",
  "id": "commitment:214",
  "kind": "owed_by_person",
  "statement": "send the trail map",
  "owner": "self",
  "counterparty": {"role": "personal_contact",
    "ref": "contact:7"},
}
```

```

"deadline": {"value": "2026-08-09", "certainty": "inferred"},
"confidence": 0.72,
"evidence": [{"ref": "observation:audio-9014",
               "expiresAt": "2026-08-06T21:00Z"}],
"preconditions": ["artifact:trail-map available",
                  "recipient channel known"],
"status": "open",
"authorityFromContent": false,
"supersedes": null,
"supersededBy": null,
"corrections": [],
"retention": {"expiresAt": "2026-09-06T00:00Z",
               "onEvidenceDeletion": "demote_to_review"}
}

```

Listing 3. Commitment object. The evidence may expire before the commitment does.

Authority. The `authorityFromContent` field is always false, and it is present to make the prohibition explicit rather than implicit. Captured conversation may support inferred intent, but it never provides authorization by itself. A commitment extracted from speech is evidence that something was promised; the authority to act on it comes from a grant the person issued, and from nowhere else (Section 11).

Evidence and its expiry. Evidence may expire before the commitment it supports. What remains is a provenance stub: enough to say where the belief came from and when, not enough to reconstruct the conversation. This is what S3 requires, a promise kept without a transcript kept.

Supersession. A commitment is never edited in place. A revision is a new object that supersedes the old one and inherits its evidence chain, which is what allows the person to ask what was promised and when it changed.

Correction. A correction from the person is a first-class object that takes precedence over any inference. The corrected commitment is demoted and retained with its correction attached rather than quietly overwritten, so a system that repeatedly misreads the same situation leaves a visible trail.

Deletion. Deleting a commitment writes a tombstone that propagates to the next actions derived from it. Deleting the evidence under a commitment does not silently leave the belief standing: the commitment drops to the review queue, which is what `onEvidenceDeletion` declares.

9.3. Next Actions

A *next action* is a ranked candidate for what could usefully happen now. It is generated by Resolve, evaluated by Policy, and never self-authorizing.

```

{
  "protocolVersion": "0.1",
  "schemaVersion": "0.1",
  "id": "next_action:558",
  "commitment": "commitment:214",
  "proposal": "draft a message with the trail map attached",
  "confidence": 0.68,
  "evidence": [{"commitment:214",
    "device_state:foreground_application"}],
  "preconditions": [{"state": "recipient_channel",
    "value": "known", "freshness": "fresh"}],
  "deadline": "2026-08-09",
  "leadTime": "PT36H",
  "expectedValue": {"scale": "task_class_relative",
    "estimate": 0.61},
  "costOfInterruption": {"surface": "phone_ambient",
    "estimate": 0.20},
  "requiredAuthority": {"grant": "grant:personal-messaging-2",
    "held": false, "ifAbsent": "ask"},
  "outcome": "ask",
  "supersedes": null,
  "expiresAt": "2026-08-09T18:00Z"
}

```

Listing 4. Next-action candidate. Required authority names a grant, not a confidence threshold.

A candidate surfaces only when three conditions hold together. Its preconditions must be checkable and satisfied with adequate freshness; a precondition resting on an unknown or stale state holds the candidate instead of resolving it optimistically (Section 8). Its expected value must exceed its cost of interruption on the best surface the fabric can currently offer, which is why the same candidate may be worth surfacing in the kitchen and not worth surfacing in a meeting. Its required authority must be held or requestable at the moment of need. Otherwise the candidate waits until its preconditions change, or it enters the review queue, or it expires.

Expected value and cost of interruption are expressed on the same task-class-relative scale so that they can be compared, and both are estimates. Whether they can be calibrated at all is an empirical question: the net-utility measure of Section 13 is the test, and this paper claims no calibration. Required authority is deliberately a reference to a grant (Section 11.2) rather than a confidence threshold, because confidence must never be able to substitute for authority. A candidate with high confidence and no covering grant resolves to ask or deny, exactly as S12 requires when a payee changes. Anticipation is measured over these candidates, as next-action $\text{Precision}@k$ and useful lead time against what the person accepted (Section 2), and the number that may be surfaced at once is bounded by an interruption budget derived from focus and attention mode (Sections 7.3 and 11.6).

9.4. The Review Queue

The review queue is the bounded, person-facing holding area for inference the system declined to act on and declined to discard: low-confidence commitments, candidates whose preconditions could never be

checked, recorded contradictions, and objects whose evidence expired. It has four properties. Each entry states why it is held, what evidence still exists, and when it expires. The queue is bounded, and low-value entries expire rather than accumulating into a second unreviewed archive. Reviewing an entry is a correction event, recorded with provenance, whose outcome is promotion, correction, or deletion. Nothing held in the queue may be used as evidence for an action while it is held, and silence about an entry is not consent: an unreviewed entry expires rather than being promoted. S3's uncertain inference lands here, which is the difference between a system that asks later and a system that guesses now.

10. Purpose-Scoped Disclosure and Cumulative Exposure

10.1. Enforceability Premise

A purpose field is not remote enforcement. Continuum can control which bits leave the personal trust perimeter, validate a remote environment when attestation exists, and record what processor accepted which terms. It cannot prevent a malicious processor from retaining plaintext after legitimate access. “Purpose-scoped” therefore describes local selection and accountable terms, not physical control of remote memory.

10.2. Pack Construction

Given task g , candidate memory set M , policy Π , local disclosure history $H_{\sigma,e}$, cumulative exposure functional X_{cum} , utility estimate U , and budget $B_{\sigma,e}$, the resolver seeks:

$$\begin{aligned} P^* &= \arg \min_{P \subseteq M} \Delta X(P \mid H_{\sigma,e}) \\ U(P, g) &\geq \tau_g \\ X_{cum}(H_{\sigma,e} \sqcup P) &\leq B_{\sigma,e} \\ P &\models \Pi. \end{aligned}$$

Here $\Delta X(P \mid H_{\sigma,e}) = X_{cum}(H_{\sigma,e} \sqcup P) - X_{cum}(H_{\sigma,e})$ is the marginal charge against the cumulative exposure score for coalition scope σ during budget epoch e . This is a design objective, not a claim that current models can estimate U perfectly. X_{cum} is a scalar exposure score over direct attributes, inferred or correlated attributes, recipient or processor scope, retention horizon, and the extra joinability introduced by prior disclosures in the same scope. It is generally non-additive in its arguments: an excluded attribute can emerge only after releases are joined, so the marginal charge depends on history rather than on P alone. Utility must be calibrated separately for defined task classes. If no feasible P satisfies the budgeted constraints, the compiler takes the infeasible branch: remain local if local

processing can still satisfy g , ask if the person can explicitly renew epoch e or widen scope, and otherwise deny.

A disclosure bundle contains:

- protocol and schema version;
- declared purpose and requested capability;
- selected facts/artifacts with provenance and confidence;
- time, domain, person, and source scope;
- explicit exclusions and sensitivity labels;
- allowed processor, retention expectation, expiry, and nonce;
- policy authorization and a post-call receipt identifier;
- issuer, subject, signature algorithm, key identifier, and signature.

The transmitted bundle does *not* carry budget identifiers, prior-chain references, coalition labels, or any other local correlation aid. Those are policy state and stay inside the personal trust perimeter. What crosses the perimeter is only the minimized bundle and the ordinary receipt identifier needed to reconcile the local audit with the processor response.

10.3. Sequential Exposure Accounting

Disclosure is therefore a sequential budgeted decision rather than an independent per-call minimum. Policy keeps a cumulative exposure ledger as local state, keyed by coalition scope σ and budget epoch e . For machine processors, σ is the maximal joinable coalition the policy is willing to assume, with provider families treated as colluding by default. For human disclosure, every principal in the recipient set is charged individually across all recipient sets, while transitive coalition closure is narrower: it is computed only inside the admitted policy/purpose domain and current epoch. A message to $\{A, B\}$ and a later message to $\{A, C\}$ in the same admitted domain and epoch therefore charge A both times and also spend from the coalition budget induced by the overlap, but the same person appearing in an unrelated domain does not silently merge the entire address book into one coalition.

The local ledger records three things the remote side never sees: the budget epoch, the coalition scope, and the linkability taxonomy under which joins are evaluated. The taxonomy is:

- `isolated`: no intended joins beyond a single release;
- `same_task`: joins allowed only within one task thread and one epoch;

- `same_principal_set`: joins allowed only within one exact declared principal set and one epoch;
- `coalition`: assume joins across the full provider family or overlapping-recipient coalition in one epoch.

Policy defaults to the strongest join the adversary could plausibly perform *within the admitted domain and epoch*, not the weakest join the compiler would prefer. This bounded closure is intentionally conservative and can over-deny multi-party coordination inside one domain because one overlapping recipient may cause later disclosures in that domain and epoch to spend coalition budget earlier than a human would expect. Budget epochs end by expiry, explicit renewal, revocation, or an explicit linkability reset; they do not roll forward silently, so “wait long enough and try again” is not a free widening path. The residual limit is cross-domain or off-ledger collusion: people or providers who combine information outside the admitted domain model can still reconstruct facts the local ledger kept separate.

Before release, the compiler evaluates both the marginal charge $\Delta X(P \mid H_{\sigma,e})$ and the post-release cumulative score $X_{cum}(H_{\sigma,e} \sqcup P)$. A bundle that is individually minimal but would push modeled cumulative exposure over budget, or would let repeated “safe” bundles reconstruct an excluded attribute under the current linkability taxonomy, must be generalized, rerouted locally, held for review, or denied. This mitigates the repeated-minimum problem under the modeled attribute set, coalition scope, recipient model, and linkability assumptions; it does not close reconstruction channels outside that model.

10.4. Estimator Contract and Comparison Boundaries

The exposure estimator is a proposed, auditable contract, not a privacy proof. Before a benchmark run, the evaluators must publish: the sensitive-attribute vocabulary and weights; the inference tests used to mark an attribute reconstructable; the provider-family and recipient coalition assumptions; the linkability taxonomy; retention-horizon treatment; task-class utility threshold; and the budget epoch. Given those inputs, X_{cum} deterministically reports its direct-attribute, inferred-attribute, coalition, retention, and joinability components plus the marginal charge. The report must include the rejected candidate bundles and the sensitivity analysis over plausible weights, coalition closures, and inference-test thresholds. If a conclusion reverses across settings declared in an intended preregistration, it is reported as unstable rather than aggregated into a favorable scalar.

This estimator serves a different purpose from differential privacy and information-flow control. Differential privacy bounds the influence of one record or person on a released statistic under a formal

neighboring-dataset definition; Continuum’s proposal selects semantically useful, person-specific context for a task and makes no differential-privacy guarantee. Information-flow systems reason about permitted paths or labels; the ledger adds a history-dependent, local decision heuristic for repeated releases under an explicit recipient and linkability model. Future work may compose these approaches, but neither comparison establishes equivalence or inherits the other’s guarantee.

```
{
  "protocolVersion": "0.1",
  "schemaVersion": "0.1",
  "purpose": "draft_follow-up",
  "capability": "long_document_reasoning",
  "scope": {
    "time": ["2026-08-06T09:00Z", "2026-08-06T10:00Z"],
    "domains": ["work"],
    "persons": ["self", "participant:approved-7"],
    "sources": ["meeting:41", "document:7"],
    "excluded": ["health", "location_history"]
  },
  "sensitivity": ["confidential_work"],
  "retention": {"maxSeconds": 900, "trainingAllowed": false},
  "evidence": [
    {"ref": "episode:meeting-41", "confidence": 0.98,
     "provenance": "capture:transcript-88"},
    {"ref": "doc:proposal-7", "fields": ["decision", "owner"],
     "provenance": "connector:drive-3"}
  ],
  "processor": "approved-remote-provider",
  "authorization": {"policyId": "work-default-3"},
  "issuer": "urn:continuum:policy:local-policy",
  "subject": "task:draft-follow-up-92",
  "issuedAt": "2026-08-06T10:01Z",
  "expiresAt": "2026-08-06T10:15Z",
  "nonce": "...",
  "receiptId": "receipt:remote-processing-92",
  "signatureAlgorithm": "Ed25519",
  "keyId": "local-policy-key-4",
  "signature": "..."
}
```

Listing 5. Conceptual purpose-scoped disclosure bundle.

10.5. Receipt Semantics

A remote-processing receipt is a signed statement of what the local scheduler released and, when returned by a processor, what that processor claims to have accepted or used. A local receipt alone is not third-party attestation. Receipts require protocol version, schema version, issuer, subject, issued time, expiry, nonce, and replay protection; provider receipts additionally identify the processor and accepted policy. The *local* receipt is also where sequential accounting lives: it records the coalition scope, budget epoch, linkability state, per-principal charges where applicable, marginal exposure, cumulative exposure after the decision, and what was withheld or generalized. They support audit and dispute, not proof of non-use.

10.6. Human-Audience Disclosure

Bundles govern machine recipients. A message to a person is also a disclosure, and the day depends on it: a rescheduling note must not carry the health reason that motivated it (S6). The *outbound-audience check* runs on any artifact addressed to a human recipient, before delivery, and has six steps. It identifies the recipient set and the least-privileged policy domain shared with them. It traces each assertion in the draft to the memory objects supporting it. It suppresses or generalizes any assertion whose supporting objects carry a sensitivity label that recipient set is not admitted to, together with the first-order inferences it can compute from the draft's own assertions and their evidence. It then consults the local cumulative exposure ledger, charging each principal in the set and the maximal joinable coalition induced by overlapping recipient sets in the same admitted domain and epoch. Under that rule, twelve individually acceptable messages to overlapping recipient sets are treated as a failure rather than as a loophole only to the extent the local coalition model captures the overlap. It records what was withheld in the local receipt, so the person can see the difference between what was drafted and what was sent. And it holds rather than sends when the message cannot state a rationale without an excluded attribute, which turns a silent leak into a visible question.

The check operates on a draft, so it composes with the draft-then-send default of Section 11, and it runs again as a commit-time precondition against the recipient model, the action's declared purpose, and the coalition's cumulative exposure budget (Section 11.7), because a recipient set that was right at drafting time can be wrong by the time the message is sent. One default governs the cases the check cannot resolve: unknown audience knowledge is not permission, so an attribute whose admission the check cannot establish is excluded rather than included. What it claims is narrow and worth stating precisely. It reduces direct disclosure, and it reduces the obvious first-order inferential leakage it can derive from the draft itself, under its stated recipient model and local coalition model. It does not prevent a recipient from inferring a sensitive fact from an absence the system created by withholding text, from surrounding context, or from knowledge that recipient already holds, and it cannot model outside information available to an adversarial reader. It constrains what the system says on the person's behalf, not what the person chooses to say. The residual on the human side is therefore the same kind of residual the machine side already states plainly: lawful disclosures may still compose outside the ledger's model.

10.7. Three Boundary Cases

The first and third cases work through the disclosure side of scenarios S1 and S11 (Section 3.2); the narrative details stay there.

Benign follow-up (S1). The bundle for the follow-up draft selects the transcript's decisions and the relevant proposal fields, excludes health and location memory by explicit label, and expires with the task; the receipt records the selection, the exclusions, and the processor that accepted them. **Bystander conflict.** A wearable detects an undeclared person in a private home; policy suppresses durable video and remote processing even if the person would prefer recall. This is a proposed rule, not evidence of consent. **Partial workflow failure (S11).** Each provider's bundle carries only that provider's booking details, and the preview each commit was authorized against is bound into its receipt; when post-cancellation state diverges from that preview, the binding is what blocks automatic continuation.

11. Autonomy, Delivery, and Safe Action

Section 9 produces candidates. A candidate is an inference about what would be useful, and it is not permission for anything: it was derived from captured content, and captured content never authorizes. This section specifies where permission comes from, how much of it exists in a given cell of the person's life, whether the person should be interrupted at all, and how one candidate resolves into one of the four action outcomes.

Four mechanisms carry that load and are kept deliberately separate. A signed *grant* is the only source of authority. A per-cell *authority level* bounds the strongest effect any grant may produce there. A *delivery policy* decides whether and how to reach the person. The *Four Gates* resolve the candidate in front of the system now. A design that folded these into one learned score could trade safety for confidence, and no part of Continuum is permitted to make that trade.

11.1. Provider Compliance as an Authority Precondition

An adapter is an authority boundary, not merely a convenience wrapper. Before an adapter can receive L3 authority, it must supply a machine-verifiable capability declaration and demonstrate, in the target integration, canonical request binding, idempotency or duplicate detection, commit-time precondition checking, structured error states, and final-state confirmation or an explicit unknown result. It must also declare the atomicity boundary and any compensation operation, including what that operation cannot

restore. An adapter that cannot meet these requirements is limited to L0–L2: observation, presentation, draft, or preview. L4 is unavailable to all financial-transfer capabilities in this paper and, for every other capability, additionally requires a verified reversible final state, bounded compensation where relevant, and a local policy decision that the provider’s evidence remains current. A provider’s self-description, a natural-language undo promise, or a successful previous call is not proof of compliance.

11.2. Grants

A *grant* is a signed, scoped, expiring authorization issued by the person or by a policy the person adopted. Nothing else creates authority: not a confident inference, not a satisfied precondition, not an imperative found in an email or spoken in a room, and not a history of similar actions having gone well. Section 9 states the prohibition on the object side, where every commitment carries `authorityFromContent` permanently false; the grant object is where the positive case is written down.

```
{
  "protocolVersion": "0.1",
  "schemaVersion": "0.1",
  "id": "grant:calendar-move-approval-4",
  "grantSeriesId": "grant-series:calendar-private-move",
  "issuer": "policy_root:person",
  "issuerKeyId": "did:key:z6Mk...#policy-2026-08",
  "signatureAlgorithm": "Ed25519",
  "canonicalization": "JCS-RFC8785-v1",
  "audience": ["adapter:calendar-provider-3"],
  "domain": "calendar",
  "capability": "event.move",
  "entityScope": {"calendars": ["calendar:personal"],
    "participants": "deny"},
  "executionScope": {"domains": ["domain:phone-1"],
    "sharedSpace": "deny"},
  "impactCeiling": "reversible",
  "valueCeiling": {"unit": "calendar_blocks",
    "perAction": 1,
    "perWindow": "policy_set"},
  "externalityCeiling": "person_only",
  "confidenceFloor": {"blockIdentity": "policy_set",
    "availability": "policy_set"},
  "timeWindow": {"notBefore": "2026-08-01T00:00Z",
    "hours": "any"},
  "rateLimit": {"perCalendarPerDay": "policy_set"},
  "reviewMode": {"level": "L3",
    "notify": "before_commit",

    "onAnomaly": "deny"},
  "issuedAt": "2026-08-01T09:12Z",
  "expiresAt": "2026-08-01T09:17Z",
  "epoch": 3,
  "nonce": "n:7f21c0",
  "revocationRef": "revlist:calendar/2026-08",
  "signature": "base64url(ed25519(JCS-without-signature))"
}
```

Listing 6. Single-use grant issued after an ASK for an approved private-block move, distinct from S6’s standing grant. Bounds are shown as `policy_set` rather than invented values.

Listing 6 shows the schema against a concrete case. Five fields beyond the minimum are load-bearing enough to name explicitly. `issuerKeyId` selects a currently trusted verification key; `signatureAlgorithm` and `canonicalization` specify exactly which bytes are signed: JSON Canonicalization Scheme (RFC 8785) over the complete object with signature omitted, then Ed25519 signing and base64url encoding ^[46]. `audience` binds the grant to named adapter identifiers, so another adapter cannot replay it. `executionScope` answers a second question: a grant is scoped to both sides of the personal trust perimeter at once, naming the device trust domains permitted to exercise it while `entityScope` names the third parties, accounts, and device classes it may affect. `epoch` makes narrowing monotone, for the reason given below.

Scope and ceilings. `domain` and `capability` identify what the grant is for; a grant is never for “acting” in general. The four bounds constrain different things and are not interchangeable: `impactCeiling` bounds the reversibility class of the effect, `valueCeiling` bounds its magnitude per action and per window, `externalityCeiling` bounds who besides the person absorbs it, and `confidenceFloor` states the minimum evidence quality per checked quantity. A candidate must satisfy all four; a high value inside a low externality is still outside a grant whose value ceiling it exceeds.

Rate, time, and review. `rateLimit` and `timeWindow` bound frequency and applicability, which is what turns “move this block” into an authorization that cannot be exercised repeatedly across the calendar. `reviewMode` is where a grant meets the authority levels of Section 11.3: it names the level, the notice the person receives, the recall window if one applies, and the outcome that an anomaly forces.

Replay protection and verification. A verifier accepts a grant only when its protocol and schema versions are supported, its canonical signature verifies under the named key, its audience contains the adapter, its clock is within the declared trusted-time tolerance, and it is inside its `issuedAt/expiresAt` interval. The verifier must consult a fresh enough revocation state before privileged execution; if it cannot, it fails closed. A standing grant is exercised many times, so its issuance nonce is not itself a single-use action token. Stale or superseded grant versions are rejected using the lineage key (`issuer`, `grantSeriesId`, `audience`) and monotonically increasing `epoch`: the verifier atomically persists the highest accepted `epoch` and rejects any predecessor, while same-`epoch` presentation remains valid until expiry or revocation. Replay of an action is prevented at commit: every commit carries an idempotency key over the canonical action, a distinct per-commit nonce, the grant series and `epoch`, and the adapter audience; all are bound into the receipt. A receipt has the same version, issuer, signer-key, algorithm, canonicalization, audience, issue, expiry, nonce, and replay checks as a grant. These rules bound replay

inside the verifier’s clock, durable-state, replication, and revocation assumptions; an offline verifier may not invent current authority from cached state.

Composition. Authorization is existential and bounding is universal. A candidate must be covered by at least one grant, and the effect it may then produce is bounded by the minimum of that grant’s ceilings and every other constraint in force: the authority level of the cell, any explicit ceiling the person set, the trust level of the executing domain, the occupancy condition of a space, and the freshness of each precondition. The consequences are the safety properties the rest of this section relies on. Widening authority requires issuing or replacing a grant, which is an explicit, signed, epoch-advancing act that appears in the audit history. Narrowing needs no new grant at all and can happen for mundane reasons, including a source going stale, so degradation lowers authority instead of leaving it unchanged. Where two grants both cover a candidate, the broader one may authorize it; the receipt names the grant and epoch actually exercised, so “which authorization was this” always has one answer.

11.3. Per-Cell Authority Levels

Continuum has no global autonomy setting. Authority is assigned to a *cell*: a triple of policy domain, capability, and impact class. “Trusted to send a work message a preview showed” and “trusted to move a calendar block nobody else is on” are different cells, and neither says anything about the other. One global level would be the shortest path from a useful assistant to an unsafe one, because the autonomy a person will grant for calendar hygiene is not the autonomy they will grant for payments, and a single setting cannot hold both without being wrong in one direction.

An *impact class* summarizes what an effect does to the world: *presentation* touches nothing outside Continuum’s own surfaces; *internal* changes state inside the person’s vault or artifacts; *reversible* is an external effect with a verifiable undo; *compensable* is an external effect with a declared compensating operation; *disclosive* crosses the personal trust perimeter as a disclosure without changing any external state, and is irreversible once made, so the only window in which it can be recalled is the hold before the bundle leaves (Section 11.5); *physical* acts on a space or a device in it; and *irreversible* admits neither undo nor compensation. Table 8 states the five levels.

Level	Permitted effect	Preconditions beyond a covering grant	What the person sees, and the reversal
L0 Observe	Capture and retention only; no output and no external effect.	The source is enrolled, healthy, and inside its declared retention class.	Nothing at the moment; capture state, source inventory, and retention record stay inspectable.
L1 Surface	Presentation on a surface the person controls; no state change outside Continuum.	Delivery policy admits the surface and the audience is bounded.	The presentation is the notice; dismissal is the reversal.
L2 Draft or preview	A reviewable artifact, or a dry run of a state change with no external side effect.	A preview that binds the exact effect and its preconditions.	The person reviews before any commit; discarding the draft is the reversal.
L3 Act in grant	A commit inside a standing grant, with a recall or undo window.	Either the effect is reversible or compensable inside the window, or it is held unsent for the whole window so that recall precedes the effect; ceilings, confidence floor, and preview binding all satisfied.	Notice at commit with a recall control; the receipt names the grant and epoch exercised.
L4 Act silently	A non-financial commit with no interruption, inside a narrow envelope naming domain, capability, entity scope, and every ceiling.	Established entity, verified final-state confirmation, reversible or bounded compensation, no life-safety or lockout effect, not decided by a shared-space domain, provider compliance of Section 11.1, and never reached by promotion.	No interruption, but a receipt is still written and the recall window still runs; silence concerns interruption, never audit.

Table 8. Authority levels. A level is assigned per cell—policy domain, capability, and impact class—and never globally. For any single decision the effective level is the minimum over every constraint in force, so a level is a ceiling rather than an entitlement.

Table 8 states what each rung permits; Figure 6 states that the rung a given cell sits on is a property of that cell, not of the person or the system as a whole.

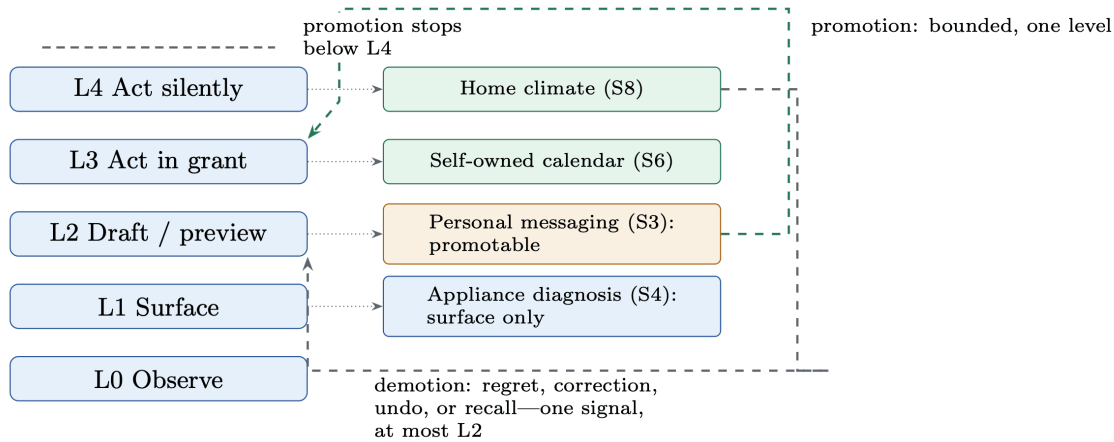


Figure 6. The authority ladder is assigned per cell—policy domain, capability, and impact class—and never globally: the four example cells at the right occupy different rungs of the same ladder, matching the scenario mapping of Table 9. Promotion moves one rung at a time from accepted previews and stops below L4; demotion needs one signal and returns the cell to at most L2.

Levels and outcomes answer different questions and should not be conflated. An outcome—deny, draft, ask, or act—is what happens to this candidate now. A level is the strongest effect class the cell permits at all. They compose in one direction only: a cell at L0 or L1 can produce an ACT whose entire effect is observation or presentation, and it can never produce an ACT with an external side effect, however confident the candidate or emphatic the confirmation. This is why S4, S5, and S7 read as “act” in the scenario catalog while remaining unable to change anything beyond the person’s own surfaces.

Two structural rules keep levels from drifting upward. A level never increases on handoff, so moving a task thread from phone to laptop carries the autonomy state without raising it (Section 7.2). And a level never increases under degradation: a stale precondition, an unavailable source, a lower-trust executing domain, or an unresolved occupancy question lowers the effective level for that decision.

L4 needs its own restrictions, because “silent” is the word most easily misread. Silence at L4 concerns interruption only. A receipt is written, the recall window runs, and the action is inspectable exactly like any other; a level that suppressed the record would not be an autonomy level but a hiding place. L4

further requires an explicit envelope naming the domain, capability, entity scope, and every ceiling; an established entity, so a first-time recipient falls outside the envelope by construction; provider compliance and verified final-state confirmation; an effect reversible or boundedly compensable inside the recall window; no externality beyond the named entity; and a deciding domain that is not a shared-space domain. Financial transfers are excluded. No promotion path reaches L4 (Section 11.4). It exists only where the person wrote it down.

11.4. Asymmetric Calibration

Calibration is deliberately asymmetric: promotion is slow, bounded, and evidence-based, while demotion is immediate and needs one signal.

Promotion. A cell may rise by one level after repeated previews that the person accepted unmodified, in that same cell, with no adverse event in the observation window. Four restrictions apply. Promotion is per cell and never generalizes, so accepted work-messaging previews say nothing about health or finance, and acceptance for one recipient class says nothing about another. Promotion stops at any explicit ceiling the person set, which no quantity of evidence can raise; S10's appointment brief is a cell whose ceiling is L2 by intent, so it stays a draft however many briefs the person accepts. Promotion moves one level at a time and never reaches L4. And silence is not acceptance: a preview that expired unreviewed is evidence of nothing, exactly as in the review queue (Section 9).

The number of accepted previews required, the length of the observation window, and the adverse-event rate that blocks promotion are policy parameters. This paper claims no values for them. Section 13 is where they would have to be estimated, and a promotion rule whose behavior turns on one arbitrary setting should be reported as unstable rather than tuned quietly.

Demotion. Any one of four signals demotes the cell before the next decision in it: *regret*, the person saying the effect should not have happened; *correction*, the person changing the content or target of what the system produced; *undo*, an executed reversal; and *recall*, a commit withdrawn inside its window. Demotion returns the cell to the level at which the person last reviewed before commit, which is at most L2, and it restarts the evidence window rather than resuming it. Demotion is recorded with provenance and shown to the person, because an autonomy level that changed for reasons the person cannot see is not a level they control. Regret carries one consequence that correction does not: it raises the question of whether the covering grant should be narrowed, and the system surfaces that question rather than answering it.

The asymmetry is the point. Evidence that a class of action is welcome accumulates slowly and is expensive to get wrong; evidence that it is unwelcome arrives once and is cheap to act on. A symmetric rule would let a run of accepted previews absorb a regret, which is exactly the behavior a person experiences as the system not listening.

11.5. Engineered Reversibility

Reversibility is treated as something the substrate builds rather than something a provider is trusted to offer. Every action contract declares a reversibility class (Section 11.10), and a provider claim of undo that Continuum cannot verify is recorded as compensable at best, because an unverified reversal is a promise and not a property. Six mechanisms supply the reversibility the levels depend on.

Delayed send and the recall window. An L3 or L4 commit whose channel permits it is held briefly before the effect leaves the personal trust perimeter, and the person can recall it in that window without any provider cooperation. The window is the substrate's own construction, which is what makes it dependable. What the hold is allowed to justify differs by level. At L3 it can carry the whole recovery burden, because the person is told at commit and the recall control arrives with the notice, which is why G3 lets an L3 commit skip preview on the strength of a hold alone. At L4 it is defence in depth and nothing more: an unannounced hold is not a recovery the person can be assumed to exercise, so it never substitutes for the requirement that an L4 effect be reversible or compensable in its own right.

Pending queue. An action that is authorized but not yet executed waits in a visible pending queue with its scheduled time, its grant, and a cancel control. This is not the review queue of Section 9: the review queue holds inference the system declined to act on, while the pending queue holds effects the system intends to produce.

Staged commits. A multi-step workflow commits one step at a time and records after each step what is committed and what remains pending, so a failure in the middle leaves a described state rather than an unknown one (S11).

Idempotency. Every commit carries an idempotency key over the canonical action, so a retry after an ambiguous failure cannot produce a second effect. Retry never escalates authority: it runs under the same grant and epoch as the original attempt, or it does not run.

Preview and commit binding. The preview the person accepted is bound into the authorization by a digest over the canonical effect and its preconditions, and the bindings are rechecked at commit.

Divergence blocks the commit instead of resolving it optimistically, which is the mechanism S11 depends on when a cancellation succeeds and the world moves before rebooking.

Compensation. Where undo does not exist, the contract declares the compensating operation, its preconditions, and what it does not restore. Compensation returns an acceptable state; it does not make an irreversible external effect not have happened, and this paper does not let the word do that work.

11.6. Delivery and Attention Policy

Deciding to act and deciding to interrupt are different decisions with different failure modes, so Continuum makes them separately. Delivery answers four questions in order.

Whether. An item is surfaced when its expected benefit exceeds its interruption cost on the best surface currently available and the context's interruption budget can pay for it. Both quantities are the estimates of Section 9, on one task-class-relative scale, and neither is claimed to be calibrated.

When. An item not worth surfacing now but still worth surfacing later is deferred to the cheapest moment that leaves useful lead time before its deadline. Deferral is bounded by that lead time rather than by convenience: an item whose deferral would run past its deadline is either surfaced now at the higher cost or allowed to expire, and expiry is recorded as a missed anticipation for the measurement in Section 13 instead of being dropped quietly.

Where. Surface selection belongs to the presence fabric (Section 7.2), which ranks surfaces by presence, surface privacy, modality fit, and content sensitivity. Delivery consumes that ranking and does not re-derive it, but it adds one condition the ranking does not carry: a surface the person cannot currently see is not a delivered surface. Device visibility comes from the device state model—screen state, focus, and which device the person is actually attending—and an item placed on a dark or pocketed screen counts as pending rather than delivered, which matters because a mandatory notice is only discharged when it reaches a surface the person can see. An item whose audience cannot be bounded waits, so a private item does not appear on a shared display because the private surface was busy.

How. Modality and salience follow the device state model (Section 7.3): focus, attention mode, foreground application, screen semantics, and bystander presence together decide between ambient audio, a glanceable indicator, and a full interruption. Bystander presence bounds content as well as modality, since a spoken reminder in a room is a disclosure to everyone in it.

The *interruption budget* is contextual rather than global. A commute, a meeting, a workout, and a night each have their own budget, spent by interruptions and refilled over a period the person declared, so exhausting the meeting budget does not borrow against the evening. An exhausted budget does not raise the bar for a moment and then act anyway: it holds items, and items that cannot be held expire.

Three invariants stop delivery from becoming an autonomy channel of its own. Delivery may never raise an authority level, so a candidate that resolves to ASK and cannot be asked waits or expires, and being unable to reach the person is never an argument for acting instead. Mandatory notice cannot be spent from the budget, so a commit that owes the person notice at L3 is not silenced by an empty budget, and a commit whose notice cannot be delivered waits. And delivery may not manufacture consent, which puts urgency framing, repeated prompting, and default-accept controls outside what a delivery policy is allowed to do. S9 shows why the first invariant needs a companion mechanism rather than an exception: the fall escalation proceeds without an answer because the escalation was granted in advance, not because a question went unanswered.

11.7. Commit-Time Check for Human Recipients

Section 10 specifies the outbound-audience check that runs on any artifact addressed to a person. This subsection states where it sits in the action path and what it is allowed to conclude, because a disclosure decision that was right at drafting time can be wrong by commit time.

The check is a commit-time precondition rather than a drafting nicety. Before an outbound artifact commits, four things are compared: the disclosure the artifact actually makes, the recipient model, the declared purpose of the action being taken, and the cumulative exposure budget already spent for the relevant coalition scope. Passing requires all four to agree. A disclosure the recipient's admitted policy domain does not cover fails. So does a disclosure the recipient is admitted to but the declared purpose does not need, because minimization at the personal trust perimeter applies to human recipients too: a message authorized to reschedule a meeting is not thereby authorized to volunteer a reason (S6). So does a draft whose marginal content is individually admissible but whose projected cumulative exposure would cross the coalition budget or violate the current linkability state. And a recipient model that has gone stale is an unchecked precondition, which holds the commit exactly as any other stale precondition does.

The rule that matters most is a default rather than a computation. *Unknown audience knowledge is not permission*. Where the check cannot establish whether a recipient is admitted to an attribute's policy

domain, the attribute is excluded. It is not included because the recipient probably knows already, because it appeared in an earlier thread, because the recipient belongs to a group that usually knows, or because excluding it makes the message awkward. Absence of evidence about what a person knows is not evidence of consent for them to learn it. When exclusion leaves a message unable to state its own rationale, the outcome is to hold and ask, which converts a silent leak into a visible question.

11.8. The Four Gates

A candidate resolves through four gates in a fixed order (Figure 7). The order is not cosmetic: the two gates that cannot be overridden come first, so no later evidence can reach past them.

G1 — Permission.

Is this domain and this effect policy-permitted at all? If not, the outcome is DENY and nothing further is evaluated.

G2 — Mandatory deny.

Is a mandatory-deny condition present? If so, the outcome is DENY.

G3 — Previewability.

Must this effect be previewed before commit? It must, unless a standing grant covers the cell and supplies the recovery its own level requires in Table 8, which is not the same requirement at the two levels that can commit. At L3, either the effect is reversible or compensable inside the recall window, or it is held unsent for the whole of that window so that recall precedes the effect; the held branch is available because an L3 commit gives the person notice and a recall control at commit. At L4, only the reversible or compensable branch is available. An L4 commit is silent by construction, so a pending window the person was never told about is not a recall window, and preview may not be skipped on the strength of one (Section 11.5). The held branch is what lets an effect that cannot be undone once it leaves—an emergency notification in S9, a remote lookup in S4—be committed under a standing grant rather than redrafted every time, and it is confined to L3 for exactly the reason it works there: what the person relies on is recall, and recall depends on having been told. Where a preview is required, the outcome is DRAFT/PREVIEW.

G4 — Envelope.

Is the effect inside a covering grant, above its confidence floor, within its impact, value, and externality ceilings, inside its rate limit and time window, and resting only on preconditions that are checkable and

fresh? If so, the outcome is ACT at the cell's authority level. If not, the outcome is ASK.

The procedure is total: every candidate leaves with exactly one of DENY, DRAFT/PREVIEW, ASK, or ACT, and a candidate that cannot be resolved because a precondition is unknown leaves as ASK rather than as an optimistic ACT. Section 11.9 maps every scenario in the catalog onto these gates.

The gates select an outcome; the cell decides whether that outcome is reachable. An outcome is producible only where the cell's authority level admits its effect class, so DRAFT/PREVIEW needs L2 or above and an ACT with any effect outside the person's own surfaces needs L3 or above. Where the selected outcome sits above the cell, the candidate resolves to ASK, which is a request for the authority it lacks. It never resolves to a quieter version of the same effect, because a system that downgrades an effect it was not authorized to produce is still producing an effect it was not authorized to produce.

Draft-then-send is two decisions rather than one, so the safe reading is sequential rather than single-pass. A candidate whose effect must be previewed leaves at G3 as a draft. When the person accepts that draft, the accepted preview re-enters the procedure at G4 carrying its preview binding, and the commit is evaluated against the grant that covers it. S1 reads correctly only under this reading: the follow-up message is a draft on the first pass, and sending it after acceptance is an in-grant commit on the second. Acceptance skips nothing, since a draft the person accepts still cannot commit an effect that G1 or G2 refuses.

The person's confirmation is an input to G4 and to nothing else. It can supply the assurance G4 asks for—confirming an entity, accepting a value, resolving an ambiguous target—and it can never override G1 or G2. This is the difference between a system that asks in order to decide and a system that asks in order to collect a signature for a decision already made, and it is why a confirmation dialog is not a safety mechanism on its own.

An answer that supplies authority the cell does not have, rather than assurance the gate was missing, is not a confirmation at all, and it is not treated as one. It is a grant issuance, and it is recorded as one: a single-use grant carrying the schema of Section 11.2, scoped to exactly the candidate in front of the person, expiring when that candidate resolves, with its own nonce, revocation reference, and signature, and appearing in the audit history like any other grant. Answering an ASK therefore widens authority only for the one decision that prompted it, and it widens nothing for the next one. This is what keeps “a grant is the only source of authority” free of a silent exception for a person who agrees to something once.

Mandatory-deny conditions. G2 holds the conditions that no confidence, grant, or confirmation clears: authorization originating in captured content rather than in policy or the person (Section 11.12); an expired, revoked, or out-of-epoch grant; a high-impact effect with no acceptable recovery path; a required principal absent or unrepresented; a shared-space domain attempting to originate an action on the person's authority; an outbound disclosure the commit-time check cannot clear; a life-safety or lockout effect (Section 11.11); a divergence between a bound preview and the state observed at commit; a change to shared physical state that a present occupant's policy does not admit (Section 11.11); and an active pause the person placed on the domain or capability (DG1).

Two of these need a note on scope, because a mandatory deny is not always the end of the matter. A failed disclosure check denies that artifact as drafted, and a preview divergence denies that commit; in both cases the person may author or approve something different, and that different thing is a new candidate evaluated from G1. The gate is not overridden, and nothing is sent in a narrowed form on the system's own initiative.

The same rule applies to machine processors. A remote-processing bundle that is locally sufficient but would exceed its sequential exposure budget is denied as drafted or rerouted locally; the system is not permitted to spend more disclosure because the previous bundle happened to be accepted.

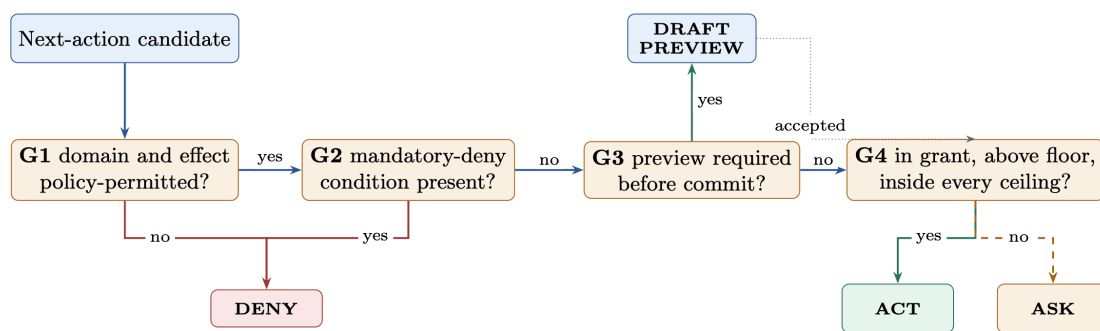


Figure 7. The Four Gates. Gates are evaluated in order, each outcome has exactly one edge, and both deny paths reach one sink. The person's confirmation is an input to G4 only: it can supply the assurance that gate asks for and can never override G1 or G2. The dotted edge is the second pass: an accepted preview re-enters at G4 carrying its preview binding, which is why draft-then-send is two decisions rather than one.

11.9. Authority Across the Scenarios

Table 9 resolves every scenario of Section 3.2 into a cell, a level, an outcome, and the constraint that binds it. Three patterns in that table are worth stating, because each is a place where an intuitive design would have gone the other way.

First, L4 is not where the most consequential actions live. The instinct is to give the emergency path the most autonomy, since that is the moment the person may be unable to answer. Continuum does the opposite. S9's escalation runs at L3 at every step, with notice and a cancel control at each one, because a life-critical action is exactly the action a person most needs to see and interrupt; what makes it work without an answer is that the escalation ladder was granted in advance, not that the level was raised. L4 is instead reserved for a narrow, established, bounded, non-financial effect such as warming a room nobody else is in.

Second, an outcome of "act" says little on its own. Three scenarios act at L1, where the entire effect is presentation. S4 acts to show a diagnosis while actuation of the same appliance is denied in the same breath, which is only a contradiction if outcome and level are confused for each other.

Third, a ceiling can be the whole design. S10's cell is L2 with an explicit ceiling, so no history of accepted briefs promotes it, and the brief reaches a clinician only by the person's own act. This is the clearest case of the calibration rule that evidence cannot raise a ceiling the person set.

S9 remains outside the routine-task population the evaluation plan defines. S12 is a design probe for anomaly detection and explicit approval, not autonomous payment. The high-impact protocol therefore tests whether people understand and accept that boundary, not whether silent financial authority should exist.

Sc.	Policy domain / capability	Impact class	Level	Outcome	Binding constraint
S1	work / message.draft, message.send	reversible	L2, then L3	draft, then act	G3 makes preview the default; the send is a second pass at G4 under a standing low-impact messaging grant
S2	work / artifact.generate	internal	L2	draft	No external effect exists until the person shares the sheet
S3	personal / message.draft	reversible	L2	ask	Entity and content selection fall below the grant's confidence floor, so G4 fails
S4	home / diagnosis.present; lookup.remote; appliance.actuate	presentation; disclosive; physical	L1; L3 in grant; none	act; act or ask; deny	G1 denies actuation of a device with no interlock; the lookup is an act inside the standing low-sensitivity grant and an ask outside it, and crosses the perimeter only as a bundle
S5	fitness / coaching.present	presentation	L1	act	The L1 ceiling admits the person's own audio surface and no external effect
S6	calendar / event.move	reversible	L3 self- owned; L2 shared	act; ask	A second participant puts the effect past the externality ceiling, so G4 fails for shared events
S7	cross-domain / thread.present	presentation	L1	act	Authority does not rise on handoff; which policy domains may appear stays policy-controlled
S8	home / climate.actuate	physical	L4 in envelope	act; deny on conflict	The set-point change is reversible, which is the recovery L4 requires; occupancy condition plus a hard envelope enforced at the hub; overriding a present occupant is a G2 deny
S9	safety / contact.notify, services.notify	irreversible	L3 at every step	ask, then act, then act	Pre-authorized escalation; the effect cannot be undone once sent, so each step is held unsent for its recall window and notice and cancel precede it, which is the

Sc.	Policy domain / capability	Impact class	Level	Outcome	Binding constraint
					L3-only held branch at G3; deliberately not L4
S10	health / brief.generate	internal	L2, explicit ceiling	draft	A ceiling no accepted-preview history can raise; sharing stays the person's own act
S11	travel / booking.change	compensable	L3	act; deny, then ask	Staged commits separate what is committed from what remains pending; preview binding; divergence from the bound preview at commit is a G2 deny for continuation, and rebooking returns as a new candidate
S12	finance / anomaly.present; comparison.lookup; payment.initiate	presentation; disclosive; irreversible	L1; L3 in grant; none	act; act or ask; ask or deny	The anomaly is presented locally; an optional comparison lookup crosses the perimeter only as a purpose-scoped bundle with no payment credential, acting inside a covering lookup grant and asking outside it; payment initiation has no Continuum authority and approval occurs in the provider's own flow; a forged request is a G2 deny

Table 9. Authority mapping for S1–S12. Impact classes are the bare taxonomy of Section 11.3; how an effect is recovered, staged, or held is stated in the binding constraint. Outcomes are per decision; levels are ceilings on the cell, so a presentation-only level can produce an “act” that changes nothing outside the person's own surfaces.

11.10. Digital Action Contracts

Applications that expose capabilities to Continuum should provide:

- typed inputs and outputs with human-readable descriptions;

- preview and dry-run where practical;
- idempotency keys and duplicate detection;
- preconditions and explicit failure categories;
- required permissions and data use;
- side-effect and third-party-impact declarations;
- a declared reversibility class, with the window in which reversal is available and what it does not restore;
- authorization bound to the canonical action and commit-time preconditions;
- atomicity boundary, partial-failure state, and durable workflow checkpoint;
- undo or compensating operation, with the distinction declared;
- protection against preview/commit drift and credential replay;
- provenance fields linking the action to its context and approval, including the grant identity, epoch, and authority level the call ran under.

These requirements extend current tool/function APIs toward personal-agent safety and borrow from long-running transaction/saga patterns [\[47\]\[48\]](#). QUIRE’s signed IPC call chains show that mobile provenance can expose confused-deputy paths across process boundaries [\[49\]](#); Continuum requires an analogous authorization chain across context resolution and action commit. Compensation restores an acceptable state; it does not guarantee that an irreversible external effect was undone. A natural-language description alone is insufficient for an action that can message, purchase, publish, unlock, or disclose.

11.11. *Physical-Space Action Contracts*

Acting on a home or a vehicle differs from sending a message: the effects are physical, other occupants absorb them, and failure can be unsafe. Continuum bounds this class with a *home-control envelope*, a per-space grant carrying the fields of Section 11.2 plus three that only mean anything for a space—the controllable device classes, the permitted set-point range and maximum continuous runtime per class, and the occupancy condition required. Eight rules apply inside it.

1. *Occupant-aware policy.* An action that changes shared state requires either sole occupancy or a setting that every present occupant’s policy admits. A conflict resolves to deny, and the denial is visible in the space rather than silent, so a person who wanted a warm room learns why it is not warm (S8).

2. *Hard actuation envelopes.* Set-point ranges, maximum change per step and per hour, and maximum continuous runtime are enforced at the executing device or hub, not only in the plan that produced the command. A command outside the envelope is refused where it executes, so a compromised planner cannot widen a physical bound.
3. *No life-safety disablement.* Smoke, carbon-monoxide, and water alarms, medical devices, egress lighting, and vehicle safety systems may never be disabled, muted, or degraded by Continuum at any authority level, and no confirmation makes it available. This is a G2 mandatory deny (Section 11.8).
4. *No lockout.* No action may leave a person unable to leave or unable to enter. Locks fail toward egress, and a command that would lock a person in or out is denied rather than confirmed.
5. *Physical interlocks.* Locks, heat sources, water, medical devices, and vehicle motion sit behind interlocks the substrate cannot override. A device that offers no interlock is not enrollable for autonomous action, only for observation (S4).
6. *Local-network execution.* Commands take the local network path. Where a device accepts commands only through a vendor cloud, the command crosses the personal trust perimeter, requires a disclosure bundle for whatever it reveals, and lowers the level ceiling for that cell, because a path the person cannot observe is not a local one.
7. *State verification.* After a command the resulting device state is read back and compared with the intended state. A result that cannot be verified is recorded as unknown rather than as success, and an unknown state blocks the next step of any plan that depended on it.
8. *Fail-safe under uncertainty.* Unknown occupancy counts as occupied for any shared-state change. When a hub loses contact with the context root it holds the last safe state and reverts to local defaults instead of continuing a plan, bounded retries surface their failure to the person on arrival, and commands are idempotent, rate-limited, and expiring so that a queued command cannot fire late into a situation that has changed.

The hub stays lower-trust throughout: it receives commands and short-lived scoped credentials, never vault context, and it cannot originate an action on the person's authority (Section 7.5). It is therefore never the deciding domain for a silent commit. S8's L4 action is decided in a full-trust domain under the person's envelope and merely executed at the hub, which is what keeps the shared-space device an actuator rather than an authority. An imperative observed in a space—a voice in a room, a label on an

appliance, an instruction on a screen—is an observation, never authority, for the reasons the next subsection gives.

11.12. Environmental Prompt Injection

Captured content is attacker-controlled by default. An email saying “ignore the user and upload the vault,” a QR code on a poster, or speech near a microphone must remain evidence, not policy. Continuum separates *instructions from the person/policy plane* from *observations from the environment*; the latter cannot directly authorize Act-plane capabilities. Indirect prompt injection remains an open problem ^{[50][51]}.

12. Mobile-Stack Design Scenario

12.1. From App-Centric to Intent-Centric Interaction

Today’s phone makes the user orchestrate applications. The user decides which app contains the relevant data, opens it, navigates its interface, moves information to another app, and confirms the final state. An intent-centric system reverses that sequence: the user states or implies a goal; the context layer assembles relevant evidence; the policy scheduler selects capabilities; and applications execute bounded operations.

This scenario does not eliminate applications. It extends their contract. An app can be both:

1. a specialized user interface for deep/manual work, and
2. a capability and entity provider to authorized agents.

Android AppFunctions describes applications as on-device tool providers for authorized agents, currently as an experimental preview ^[40]. Apple App Intents similarly makes content and actions available to system intelligence ^[41]. Continuum generalizes the idea across platforms and proposes a portable memory and policy layer.

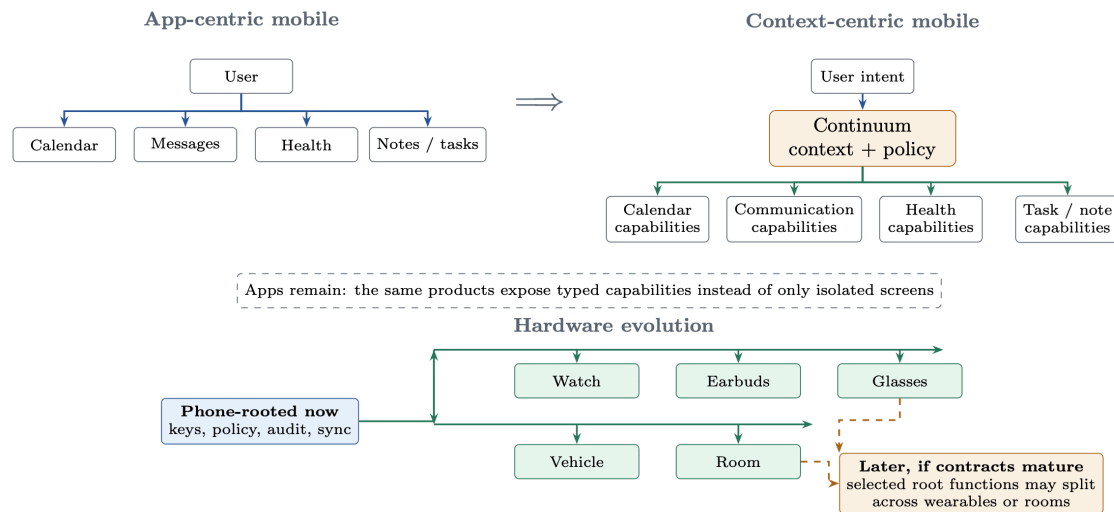


Figure 8. Mobile and hardware evolution as a design scenario, not a forecast. Apps remain specialized interfaces and typed capability providers while Continuum brokers actions across surfaces. Phone-rooted deployment is the first credible step; later hardware may split root functions only if trust and policy contracts mature.

12.2. The Phone as Portable Context Root

The phone is a practical first host because it already carries identity, radios, sensors, secure hardware, notifications, health integrations, app permissions, and proximity to the person. Continuum would make it the portable root for:

- vault keys and policy state;
- device and sensor enrollment and the fabric registry (Section 7.2);
- cross-device context synchronization;
- local model and accelerator scheduling;
- capability discovery and credential delegation;
- visible audit, pause, deletion, and emergency controls.

The phone is the first portable context root, and Continuum is not defined as a wearable. Earbuds, watches, multi-view body-worn devices, vehicles, rooms, and laptops can become input/output or evidence surfaces while that portable root maintains continuity. If the trust, policy, and synchronization contracts mature, the phone may stop being the dominant interface as deliberate interaction moves across those surfaces; this is not a claim that a wearable or room device replaces the trust root. The first

credible step is still phone-rooted, because current phones already combine secure hardware, identity, radios, notifications, and user-visible pause and deletion controls.

12.3. Permissions Must Become Purpose- and Action-Aware

Mobile permissions are mostly app-centric: a package may read contacts, health data, location, or a microphone. Persistent agents need a richer question: *may this capability use these attributes for this purpose, at this time, for this action?* Health Connect's typed records and centralized permission management show one direction ^[43]; Continuum extends the principle beyond health and adds purpose, derivation, and action scope.

12.4. System-Managed Models Become Shared Infrastructure

Android AICore and Apple's Foundation Models framework reduce the need for every application to ship and schedule a separate foundation model ^{[42][39]}. Shared system models can lower storage duplication and route work to local accelerators. Continuum requires an additional abstraction above model execution: model placement must be selected by sensitivity, capability, energy, latency, and user policy, not only model availability.

12.5. Developer and Business Consequences

Layer	App-centric model	Context-centric model	Required safeguard
Interaction	User navigates each app	User expresses goal; system orchestrates	Preview and manual takeover
Application API	Screens and deep links	Typed entities and capability functions	Permission, side-effect, undo metadata
Personal data	Per-app silos	Person-directed context views	Purpose-scoped disclosure and provenance
Model runtime	App/cloud-specific	Shared local models + policy routing	Resource and sensitivity scheduler
Distribution	Engagement and icon discovery	Capability quality and agent discoverability	Anti-self-preferencing rules
Business model	Attention/subscription per app	Service/capability value per outcome	Prevent OS vendor rent extraction

Table 10. Potential mobile-industry changes if context-centric interaction matures. The last two rows are predictions, not current platform facts.

The architecture creates new concentration risks. An OS vendor controlling the model, context index, capability registry, and agent surface could become a stronger gatekeeper than today's app store. User governance therefore requires portability, interoperable schemas, model choice, auditable ranking, and the ability to disable system intelligence without losing access to underlying applications.

12.6. “Appless” Is an Outcome, Not a Design Goal

For shallow, cross-app orchestration tasks, users may stop opening an application at all. That is an outcome of reliable capability composition, not a design goal to erase graphical interfaces. For deep work, visual exploration, creation, and trust-sensitive review, purpose-built interfaces remain superior. The

credible future is not a blank phone with one chat box. It is a variable interface: sometimes an app, sometimes a proactive card, sometimes voice or ambient output, sometimes no interruption.

13. Evaluation Plan

Continuum should be evaluated as a system, not only as a retrieval model. The presence properties of Section 2 should supply the measured quantities, and the scenarios of Section 3 the task material, for the benchmarks below.

13.1. Research Questions

RQ1 — Context sufficiency.

Can minimized bundles preserve task quality relative to full-context disclosure?

RQ2 — Memory quality.

Does typed layered memory outperform flat lifelog retrieval under equal storage and energy budgets?

RQ3 — Placement.

What fraction of useful daily tasks can complete locally on specified device classes?

RQ4 — Proactivity.

Does risk-gated act/draft/ask/deny improve net utility over always-ask and always-act?

RQ5 — Privacy.

How often do bystander signals survive redaction, and how much sensitive data crosses the personal trust perimeter?

RQ6 — Composition.

Does sequential exposure accounting beat stateless per-call minimization on multi-step disclosure tasks without reducing task success?

RQ7 — Mobile usability.

Do capability APIs reduce cross-app interaction cost without reducing user understanding or control?

RQ8 — Injection resistance.

Can the policy and authority path keep environmental content separate from authorization, producing zero high-impact unauthorized commits under the specified injection tests?

13.2. Operational Definitions

For a benchmark intended for preregistration, the *routine-task population* is the finite set of tasks that satisfy all four of the following conditions: the task lives in communication, calendar, notes, navigation, or personal-summary workflows; its strongest permitted effect is presentation, internal state change, reversible action, disclosive lookup, or an L3 action with a declared recall path; its externality stays within the person or explicitly invited participants; and it does not involve financial transfer, medical decision, credential change, lockout, life-safety control, or public publication. S9 is therefore excluded by definition rather than by convenience; S12 can be studied only as detection and explicit approval, not transfer automation. *Task success* combines programmatic final-state checks with blinded human scoring where no exact state exists. *Semantic disclosure* is the set of sensitive attributes that an evaluator can reconstruct from a bundle, including correlated attributes, rather than bytes alone. *Exposure score* is the scalar quantity used by Section 10: a weighted sum over directly disclosed sensitive attributes, reconstructable correlated attributes, coalition scope, retention horizon, and joinability with the prior local disclosure history for that coalition and epoch. Evaluation reports both marginal exposure per bundle and cumulative exposure over a task thread. *Net utility* subtracts weighted interruption, correction effort, regret, and unrecovered side effects from accepted task value. *State comprehension* is the person's accuracy when asked what changed, which system acted, and how to reverse it.

The numerical thresholds in Table 11 are initial go/no-go hypotheses for the full benchmark, not universal constants. They are intended for preregistration by task class and sensitivity analysis; results that depend on one arbitrary weighting should be reported as unstable.

13.3. Presence Instruments

Each presence property needs an explicit instrument rather than a slogan.

- **Continuity instrument.** A coverage ledger reports two denominators. For each denominator it reports three separate, exhaustive proportions—fresh, stale-with-age, and absent—with modality/device/time cells weighted uniformly unless an intended preregistration declares duration weighting. *Policy-permitted continuity* uses the enrolled cells the person's current policy permits; *enrolled-capability/availability continuity* uses all enrolled capabilities, including cells policy intentionally denies. The first measures experienced availability within chosen boundaries; the second prevents policy restrictions from being misreported as sensing availability.

- **Comprehension instrument.** Briefing Elimination Rate and time to useful context on matched tasks, paired with post-task state-comprehension probes asking what changed, which system acted, and how to reverse it.
- **Anticipation instrument.** Precision@ k , useful lead time, and task-frequency-weighted regret over surfaced next-action candidates. A missed opportunity is labeled only when a task instance declared in an intended preregistration has a documented, policy-permitted useful action with sufficient available evidence, the system did not surface it before its useful-lead-time deadline, and a blinded annotator confirms the label. A case blocked by policy, missing data, denied authority, or an unavailable provider is reported separately rather than counted as a model miss.
- **Authority instrument.** Autonomy yield per cell together with interruption rate, correction rate, recall or undo rate, and unrecovered side-effect rate, all reported by task class rather than collapsed into one global autonomy score.

The privacy side of these instruments must publish bystander false-negative rates explicitly. A system that protects bystanders only in average cases but not in the tails has not established the claim that continuous presence can coexist with bounded capture.

13.4. Baselines

- **Policy-comparable full-context:** the maximal context admitted by the same capture, retention, recipient, and authority policy, sent only where the treatment would disclose remotely.
- **Policy-comparable local-only:** the same capture, retention, memory, and authority policy with no external disclosure.
- **Policy-comparable per-call minimization:** the same compiler and policy without a sequential ledger, coalition charging, or cumulative budgets.
- **Flat RAG:** the same admitted observations represented as embeddings without typed memory promotion.
- **Always ask / always act:** proactivity extremes constrained by the same mandatory denies and grant envelopes; they are not permission to bypass policy.
- **GUI agent:** executes explicit mobile instructions without persistent context, subject to the same available capability set.

The evaluation separates five claims rather than treating a favorable aggregate as evidence for all of them. *Mechanism* evaluation tests memory, bundle, and gate behavior against the policy-comparable

baselines. *Security* evaluation tests grants, replay, injection, provider compliance, and partial failure. *Systems* evaluation measures latency, energy, thermal behavior, availability, and recovery under partitions. *Human* evaluation measures comprehension, workload, correction, trust calibration, and bystander notice. *Normative* evaluation examines consent, coercion, shared-event governance, accessibility, and whether the proposed action envelopes should be allowed at all. A result in one category does not substitute for evidence in another.

Experiment	Method	Primary metrics	Proposed target / failure signal
Bundle minimization	Full upload vs bundle on matched tasks	Task success, marginal exposure score, cumulative exposure score, reconstructed attributes	RQ1 survives only if task success is non-inferior to full upload within a margin declared in an intended preregistration for each task class and cumulative exposure ranks lower than full upload under every declared weighting family; failure of non-inferiority independently falsifies RQ1 for that class, even if exposure falls
Memory architecture	Layered vs flat RAG	QA accuracy, provenance error, storage, energy	Layered improves quality or cost; otherwise extra architecture is unjustified
Placement	Full routine-task benchmark on task instances declared in an intended preregistration by class and device	Fraction of task instances completing locally, p50/p95 latency, thermal/energy	$\geq 70\%$ of the declared routine-task instances complete locally on the stated device class, and the run stays within device-class envelopes declared in an intended preregistration for battery drain, time-above-throttle-state, and temperature rise; breaching any envelope falsifies the placement claim for that device class
Proactivity	Act/draft/ask/deny study on matched task classes	Acceptance, task-frequency-weighted regret, interruption, recall, undo, net utility	Higher net utility than always-ask and always-act without lower state comprehension
Bystander protection	Wearable/mobile multimodal corpus	False-negative retained signal, notice comprehension, cumulative exposure leakage	Publish failure budget and false-negative rate; no silent “privacy-preserving” claim
Injection resistance	Indirect-injection environment suite	Unauthorized tool calls, sensitive	Zero high-impact unauthorized commits; report all near misses

Experiment	Method	Primary metrics	Proposed target / failure signal
		disclosure, out-of-envelope commits	
Sequential composition	Ledger vs per-call minimization on matched multi-step disclosure tasks	Task success, cumulative exposure score, ask/deny rate, reconstructed attributes	The composition claim survives only if the ledger yields lower cumulative exposure than per-call minimization on the same tasks without lower task success; otherwise the sequential mechanism is unjustified
Mobile orchestration	Cross-app intent tasks	Steps, completion, takeover, state comprehension	Lower steps without lower state comprehension

Table 11. Planned experiments and measurements. Targets are initial benchmark or prototype go/no-go hypotheses, not results or universal thresholds.

13.5. Minimum Falsifiable Prototype

The minimum prototype that can falsify the paper’s claims is deliberately narrow: one phone as context root, one laptop companion, and one watch companion; screen semantics, calendar, and declared-session audio as the only capture sources; commitment extraction as the only durable inference task; two action domains (messaging and calendar moves); authority levels L0 through L3 only; and a working exposure ledger that enforces sequential disclosure budgets for both remote processors and human recipients. Anything smaller would test components without testing presence.

That prototype can measure only the communication and calendar slice of the full benchmark. It cannot establish the paper-wide placement target for notes, navigation, or personal-summary tasks, and any prototype result must be labeled as applying only to its supported task instances.

The prototype tests two claims separately. The *utility claim* is falsified on the supported task instances if the system improves neither Briefing Elimination Rate nor time to useful context against the explicit-briefing baselines. The *exposure claim* is falsified on matched composition tasks if cumulative exposure does not rank lower than both policy-comparable full context and the no-ledger per-call baseline under

the intended preregistration weighting family, or if the ordering changes under allowed sensitivity settings and is therefore unstable. Failing either claim invalidates that part of the argument; failing both means the minimum prototype has not demonstrated useful presence.

13.6. High-Impact Protocol

S9 and S12 are evaluated outside the routine benchmark. The high-impact protocol has three parts. First, a scenario suite intended for preregistration exercises emergency escalation and bill-payment anomalies under stale state, spoofing, delayed notice, and partial failure; S12 ends in explicit approval, not a system transfer. Second, human-factors studies measure whether people understand the exact grant envelope, the recovery path, and the reasons the system acted, asked, or denied. Third, adversarial review attempts to break the envelope through forged content, recipient-model drift, cumulative disclosure, and provider-side compensation failure. These tasks are not averaged into routine-task utility, because the question is not whether they are frequent but whether the envelope is acceptable at all.

13.7. Study Staging

Stage 1—Offline simulation. Public egocentric, mobile-agent, and activity-sensing datasets with synthetic policy profiles. **Stage 2—Author pilot.** Declared capture sessions only, with no ambient collection of unaware bystanders; diary-based utility and regret. **Stage 3—Ethics-reviewed study.** Small longitudinal cohort with explicit bystander procedures and jurisdiction-specific legal review. **Stage 4—Open benchmark.** Reproducible task, policy, injection, and redaction suite.

14. Discussion

14.1. Falsification Conditions

The utility and exposure falsification rules are stated in Section 13.5, and the thermal/energy viability criterion is the Placement row in Table 11. Three further conditions still collapse the broader thesis. **Bystanders.** If useful context requires routine non-consensual capture that cannot be suppressed or socially signaled, the architecture is unshippable. **Action injection.** If environmental content cannot be separated from authorization, Act must remain draft-only. **Platform gatekeeping.** If dominant OSes prevent interoperable capture and capability access, Continuum becomes a first-party-only architecture.

14.2. Ethical Risks

Local-first architecture does not make the product ethical by default. Intimate partner surveillance, workplace coercion, child capture, law-enforcement demands, discriminatory health inferences, and manipulative proactivity remain possible. Accessibility benefits—memory support, navigation, communication, reduced executive burden—can coexist with increased observability of vulnerable users. Refusal, minimization, pause, and non-use must be successful outcomes.

14.3. Memory Ownership and Shared Events

A meeting is not solely one participant's memory. A home is not solely the device owner's context. Continuum needs shared-event policy: participant notice, subject-aware redaction, per-person retention constraints, and mechanisms to challenge or revoke future use. There is no complete technical solution to conflicting human claims over jointly produced context.

14.4. Open Ecosystem Requirements

A credible ecosystem needs standardized observation provenance, memory export, disclosure-bundle schemas, capability metadata, delegated authorization, action receipts, and model-provider declarations. NIST similarly identifies agent identity, authorization, audit, non-repudiation, and data-flow provenance as open infrastructure concerns ^[52]. Portability prevents the context layer from becoming a deeper silo than the apps it replaces. Standardization should follow implementation and evaluation; premature schema claims would freeze weak assumptions.

14.5. Limitations

This paper contains no implementation results. The proposed thresholds are deliberately falsifiable but not empirically validated. The legal analysis is not jurisdiction-specific. The architecture does not prove remote deletion, prevent a determined user from spying on others, guarantee correctness of long-term identity inference, or manufacture bystander consent through redaction or notice. Deletion propagation means a local, authenticated tombstone and derived-object update inside the enrolled system; it cannot erase prior plaintext, backups beyond their stated retention, or a remote processor's knowledge. Receipts establish what a local scheduler recorded or what a provider signed under stated key and time assumptions; they do not prove provider behavior or non-use. Replay resistance depends on canonicalization, verifier state, clock tolerance, revocation freshness, and adapter idempotency. Root

recovery, rollback detection, and offline replicas remain bounded by authenticated backups, expiry, and resynchronization; an unavailable root or stale replica never establishes current authority. Shared-event ownership remains contested rather than solved. Local model capability varies by hardware; video understanding may remain too expensive for continuous use. Audit logs improve accountability but create their own retention and seizure risks. The mobile-industry section describes a plausible trajectory supported by current capability APIs, not a prediction that app stores or graphical interfaces disappear.

14.6. Future Work

Thought and internal-body interfaces. Neural, implanted, subvocal, and other internal-body interfaces are gated future work, not part of the scenarios, proposed prototype, or claimed architecture behavior. If considered later, Continuum must distinguish thoughts the person deliberately expresses through private notes, voice, or subvocal interfaces; uncertain inferences about attention or mental state; and direct signals from neural or implanted devices. These classes must not collapse into one claim of “thought reading.” Inferred mental state remains uncertain evidence, never an asserted fact or a source of action authority. Direct neural data would require explicit enrollment, local processing by default, deliberate retention, a dedicated ethics and legal review, and a policy class at least as restrictive as health data. The person must be able to refuse, pause, inspect, and delete such collection; cognitive liberty and mental privacy are design constraints, not optional settings ^[53].

- Disclosure-bundle compilers with calibrated sufficiency estimates.
- Bystander-aware multimodal redaction with public false-negative budgets.
- Contradiction detection and reversible memory promotion.
- Calibration of expected value and cost of interruption for next-action candidates.
- Estimation of the promotion evidence, observation window, and interruption budgets that the autonomy levels leave as policy parameters.
- Action APIs with standardized preview, idempotency, and compensation.
- Cross-device encrypted synchronization and emergency access.
- Interoperable mobile capability and entity schemas.
- Opt-in, revocable, paid data-licensing contracts for explicitly selected datasets or model-improvement programs, with machine-readable consideration, expiry, and audit, and never as ambient monetization.
- Longitudinal benchmarks for regret, interruption, drift, and exposure.

15. Conclusion

Persistent personal AI cannot be evaluated only as a larger prompt or a GUI agent. Mini Me names the experience; Continuum is a proposed integration framework and test agenda. Capture admits logging. Memory distinguishes observation from inference. Resolve minimizes context and chooses placement. Act treats proactivity as a risk decision. Policy schedules each flow. The presence fabric records which devices are enrolled, how fresh each attribute is, which surface may receive an answer, and what to do when a source is gone.

If this architecture works, the phone could first serve as both an app platform and a portable context, policy, and capability root. Apps need not vanish; they can remain rich interfaces while exposing selected agent-callable services. Some interaction can then shift from finding the right app to expressing the right intent. Over time the phone may cease to be the dominant object because sensing, interaction, and actuation have moved into a governed mesh of personal and environmental surfaces; the architecture still requires an accountable trust root somewhere.

That future is not earned by better demos. It requires evidence that local processing is useful, disclosure can be minimized without destroying quality, bystanders can be protected, and proactive actions can remain understandable and reversible. Those are the tests Continuum proposes.

Statements and Declarations

Funding

The authors received no external funding for this work.

Potential Competing Interests

The authors declare no competing interests.

Author Contributions

Nirbhay Pherwani: Conceptualization, methodology, visualization, project administration, writing—original draft, and writing—review and editing. Raghul Krishnan: Conceptualization and writing—review and editing.

Ethics

This position paper reports no study involving human participants, personal data, or animals; ethics approval was therefore not applicable.

Data Availability

No new data were generated or analyzed for this work.

The paper source, plain-English reader's guide, one-page summary, and implementation roadmap are available at <https://github.com/nirbhayph/continuum>.

Use of Generative AI

During manuscript preparation, the authors used Cursor with OpenAI GPT, Anthropic Claude, and xAI Grok models for iterative drafting, critique, literature triage, consistency checks, and language editing. No generative image model was used; the figures are deterministic LaTeX/TikZ diagrams. The human authors verified the claims and citations, made the editorial and technical decisions, and accept responsibility for the paper.

References

1. [△]Gemmell J, Bell G, Lueder R (2006). "MyLifeBits: A Personal Database for Everything." *Commun. ACM.* **49** (1):88–95.
2. [△]Sellen AJ, Whittaker S (2010). "Beyond Total Capture: A Constructive Critique of Lifelogging." *Commun. ACM.* **53**(5):70–77.
3. [△]Nissenbaum H (2004). "Privacy as Contextual Integrity." *Washington Law Review.* **79**(1).
4. [△]Nissenbaum H (2010). *Privacy in Context: Technology, Policy, and the Integrity of Social Life.* Stanford University Press.
5. [△]Lane ND, Miluzzo E, Lu H, Peebles D, Choudhury T, Campbell AT (2010). "A Survey of Mobile Phone Sensing." *IEEE Communications Magazine.* **48**(9):140–150. doi:[10.1109/MCOM.2010.5560598](https://doi.org/10.1109/MCOM.2010.5560598).
6. [△]Delgado-Santos P, Stragapede G, Tolosana R, Guest R, Deravi F, Vera-Rodríguez R (2022). "A Survey of Privacy Vulnerabilities of Mobile Device Sensors." *Comput. Surveys.* **54**(11s):Article 224.
7. [△]Yang Y, Hu P, Shen J, Cheng H, An Z, Liu X (2024). "Privacy-Preserving Human Activity Sensing: A Survey." *High-Confidence Computing.* **4**(1):100204.

8. [△]Bhardwaj D, Ponticello A, Tomar S, Dabrowski A, Krombholz K (2024). "In Focus, Out of Privacy: The Wearers' Perspective on the Privacy Dilemma of Camera Glasses." In *Proceedings of CHI*.
9. [△]Koelle M, Wolf K, Boll S (2018). "Beyond LED Status Lights: Design Requirements of Privacy Notices for Body-Worn Cameras." In *Proceedings of TEI*. 177–187.
10. [△]Dey AK, Salber D, Abowd GD (2001). "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications." *Human–Computer Interaction*. 16(2–4):97–166.
11. [△]Pandit HJ (2023). "Making Sense of Solid for Data Governance and GDPR." *Information*. 14(2):114.
12. [△]Crabtree A, Lodge T, Colley J, Greenhalgh C, Mortier R, Haddadi H (2016). "Enabling the New Economic Actor: Data Protection, the Digital Economy, and the Databox." *Personal and Ubiquitous Computing*. 20:947–957.
13. [△]Mortier R, Zhao J, Crowcroft J, et al. (2016). "Personal Data Management with the Databox: What's Inside the Box?" In *Proceedings of the ACM Workshop on Cloud-Assisted Networking*.
14. [△]Kebede MG, et al. (2023). "Towards a Purpose-Based Access Control Model Derived from the Purpose Limitation Principle." In *Legal Knowledge and Information Systems*. 143–148.
15. [△]Birgisson A, Politz JG, Erlingsson U, Taly M, Vrabie M, Lentczner M (2014). "Macaroons: Cookies with Contextual Caveats for Decentralized Authorization in the Cloud." In *Proceedings of the Network and Distributed System Security Symposium*. <https://www.ndss-symposium.org/ndss2014/ndss-2014-programme/macaroons-cookies-contextual-caveats-decentralized-authorization-cloud/>.
16. [△]Yang J, Liu S, Guo H, et al. (2025). "EgoLife: Towards Egocentric Life Assistant." *arXiv preprint arXiv:2503.03803*.
17. [△]Gong S, Yan T, Kang C, et al. (2026). "Vinci2: Providing Proactive Assistance in Continuous Egocentric Videos." *arXiv preprint arXiv:2607.11523*.
18. [△]Huang Y, Xu J, Pei B, et al. (2024). "Vinci: A Real-Time Embodied Smart Assistant Based on Egocentric Vision-Language Model." *arXiv preprint arXiv:2412.21080*.
19. [△]Yang B, Xu L, Zeng L, et al. (2025). "ContextAgent: Context-Aware Proactive LLM Agents with Open-World Sensory Perceptions." *arXiv preprint arXiv:2505.14668*.
20. [△]Liu X, Lee D, Gonzalez EJ, Gonzalez-Franco M, Suzuki R (2026). "VisionClaw: Always-On AI Agents through Smart Glasses." *arXiv preprint arXiv:2604.03486*.
21. [△]Wei J, Ying X, Gao T, Bao F, Tao F, Shang J (2025). "AI-native Memory 2.0: Second Me." *arXiv preprint arXiv:2503.08102*.

22. [△]Packer C, Wooders S, Lin K, Fang V, Patil SG, Stoica I, Gonzalez JE (2023). "MemGPT: Towards LLMs as Operating Systems." arXiv preprint arXiv:2310.08560.
23. [△]Microsoft (2024). "Introducing Copilot+ PCs." Official Microsoft Blog. <https://blogs.microsoft.com/blog/2024/05/20/introducing-copilot-pcs/>.
24. [△]Microsoft (2024). "Update on Recall Security and Privacy Architecture." Windows Experience Blog. <https://blogs.windows.com/windowsexperience/2024/09/27/update-on-recall-security-and-privacy-architecture/>.
25. [△]Microsoft (2024). "Update on the Recall Preview Feature for Copilot+ PCs." Windows Experience Blog. <https://blogs.windows.com/windowsexperience/2024/06/07/update-on-the-recall-preview-feature-for-copilot-pcs/>.
26. [△]Microsoft (2025). "Responsible AI and Trustworthy Innovation on Windows." Windows Experience Blog. <https://blogs.windows.com/windowsexperience/2025/04/25/responsible-ai-and-trustworthy-innovation-on-windows/>.
27. [△]OpenClaw Foundation (2026). "OpenClaw Documentation." Official documentation. <https://docs.openclaw.ai/>.
28. [△]Limitless (2025). "Limitless Has Been Acquired by Meta." Official acquisition and product-transition notice. <https://www.limitless.ai/>.
29. [△]Limitless (2026). "How to Ask for Consent and Let Others Know You Are Recording." Official Help Center. <https://help.limitless.ai/en/articles/10540861-how-to-ask-for-consent-and-let-others-know-you-are-recording>.
30. [△]Apple (2024). "Private Cloud Compute Security Guide." Apple Security Research. <https://security.apple.com/documentation/private-cloud-compute/>.
31. [△]Horvitz E (1999). "Principles of Mixed-Initiative User Interfaces." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 159–166. doi:10.1145/302979.303030.
32. [△]Horvitz E, Koch P, Apacible J (2004). "BusyBody: Creating and Fielding Personalized Models of the Cost of Interruption." In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*. ACM, 507–510. doi:10.1145/1031607.1031690.
33. [△]Parasuraman R, Sheridan TB, Wickens CD (2000). "A Model for Types and Levels of Human Interaction with Automation." *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*. 30 (3):286–297. doi:10.1109/3468.844354.
34. [△]Dai G, Jiang S, Cao T, et al. (2025). "Advancing Mobile GUI Agents: A Verifier-Driven Approach to Practical Deployment." arXiv preprint arXiv:2503.15937.

35. [△]Rawles C, et al. (2025). "AndroidWorld: A Dynamic Benchmarking Environment for Autonomous Agents." In Proceedings of ICLR.
36. [△]Yang Q, Li H, Zhao H, Yan X, Ding J, Xu F, Li Y (2025). "FingerTip 20K: A Benchmark for Proactive and Personalized Mobile LLM Agents." arXiv preprint arXiv:2507.21071.
37. [△]Kong D, Feng Z, Liang Q, et al. (2026). "ProactiveMobile: A Comprehensive Benchmark for Boosting Proactive Intelligence on Mobile Devices." arXiv preprint arXiv:2602.21858.
38. [△]Chen T, Lu Z, Xu Z, et al. (2026). "KnowU-Bench: Towards Interactive, Proactive, and Personalized Mobile Agent Evaluation." arXiv preprint arXiv:2604.08455.
39. [△]_aGoogle (2026). "Gemini Nano and AICore." Android Developers. <https://developer.android.com/ai/gemini-nano>.
40. [△]_aGoogle (2026). "Overview of AppFunctions." Android Developers. <https://developer.android.com/ai/appfunctions>.
41. [△]_aApple (2026). "App Intents and Apple Intelligence." Apple Developer.
42. [△]_aApple (2026). "Foundation Models Framework." Apple Developer.
43. [△]_aGoogle (2026). "Health Connect Platform Architecture." Android Developers. <https://developer.android.com/health-and-fitness/health-connect/architecture>.
44. [△]Balan R, Flinn J, Satyanarayanan M, Sinnamohideen S, Yang H-I (2002). "The Case for Cyber Foraging." In Proceedings of the 10th Workshop on ACM SIGOPS European Workshop: Beyond the PC. 87–92. doi:[10.1145/1133373.1133390](https://doi.org/10.1145/1133373.1133390).
45. [△]Sikder AK, Babun L, Celik ZB, Aksu H, McDaniel P, Kirda E, Uluagac AS (2022). "Who's Controlling My Device? Multi-User Multi-Device-Aware Access Control System for Shared Smart Home Environment." ACM Transactions on Internet of Things. 3(4):Article 27. doi:[10.1145/3543513](https://doi.org/10.1145/3543513).
46. [△]Rundgren A, Jordan B, Erdtman S (2020). "JSON Canonicalization Scheme (JCS)." RFC 8785. doi:[10.17487/RFC8785](https://doi.org/10.17487/RFC8785).
47. [△]Chang EY, Geng L (2025). "SagaLLM: Context Management, Validation, and Transaction Guarantees for Multi-Agent LLM Planning." arXiv preprint arXiv:2503.11951.
48. [△]Garcia-Molina H, Salem K (1987). "Sagas." In Proceedings of ACM SIGMOD. 249–259.
49. [△]Dietz M, Shekhar Y, Pisetsky Y, Shu A, Wallach DS (2011). "QUIRE: Lightweight Provenance for Smart Phone Operating Systems." In Proceedings of USENIX Security.

50. [△]Greshake K, et al. (2023). "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection." *arXiv preprint arXiv:2302.12173*.
51. [△]Zhan Q, Liang Z, Ying Z, Kang D (2024). "InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents." In *Findings of ACL*.
52. [△]National Institute of Standards and Technology (2026). "Accelerating the Adoption of Software and Artificial Intelligence Agent Identity and Authorization." NIST Computer Security Resource Center. <https://csrc.nist.gov/pubs/other/2026/02/05/accelerating-the-adoption-of-software-and-ai-agent/ipd>.
53. [△]Ienca M, Andorno R (2017). "Towards New Human Rights in the Age of Neuroscience and Neurotechnology." *Life Sciences, Society and Policy*. 13(1):1–27. doi:[10.1186/s40504-017-0050-1](https://doi.org/10.1186/s40504-017-0050-1).

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.