Dependence-Aware Turn-Taking for Voice Agents with Reversible Audio Commitment

An implemented and computationally verified controller for reversible response preparation and atomic audio commitment in interactive voice systems

AUTHORS

Eros Marcello Iuliano · Thea Peterson
ethereal computing inc.
Moffett Field, California, USA

CORRESPONDING AUTHOR

info@etherealcomputing.com

EDITION

Version 4.2 · August 2026
Public Research Edition
Supersedes version 4.1 (August 2026)

ARTIFACT

Reference controller, verifier, stress tests
Available from the corresponding author

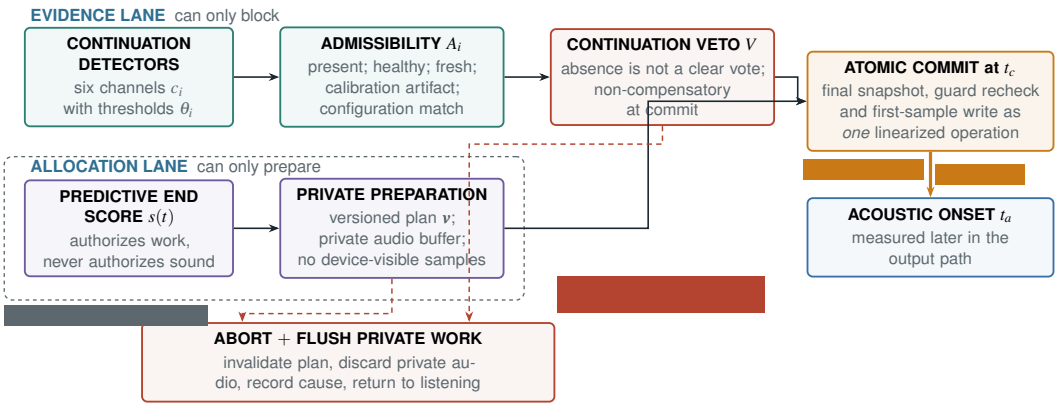
Status of this document. This public research edition reports an implemented reference controller and computational verification. It has not completed peer review and should not be represented as a published journal article. The reported results verify controller logic and declared model sensitivity; they do not establish detector accuracy, real-world audio latency, interruption reduction, or user preference. Corpus replay, hardware-in-the-loop timing, and participant evaluation remain required.

Abstract

Voice agents can prepare responses before a user yields, but early audio release can interrupt speech and speculative content can become stale when incremental recognition revises. This paper presents Predictive Conversational Patience (PCP), a reference control architecture that separates reversible response preparation from an atomic first-sample commitment. A continuation veto gated by detector presence, health, freshness, calibration, and configuration identity remains non-compensatory at commit; versioned plans and private audio are discarded whenever their producing state changes. The controller was implemented and evaluated over all 40,960 combinations of five states and 13 Boolean guard conditions, followed by 1,000,000 randomized transitions containing 80,478 valid commits and 21,552 aborts. No contract violation occurred, and five seeded safety mutations were all detected. Model-based beta–binomial and Markov stress tests, numerically cross-checked with 1,000,000 Monte Carlo trials per point, show that positive detector dependence can raise a simultaneous six-detector miss probability from 10^{-6} at independence to 3.23×10^{-2}, while temporal persistence can raise mean false-veto delay from 18.0 to 120.1 ms. These results verify controller logic and model sensitivity, not conversational performance; corpus, hardware, and user evaluation remain required.

Keywords: Conversational interfaces, Dialogue systems, Interactive audio, Speech processing, Streaming systems

System at a glance



Read the two lanes as concurrent, not sequential. A rising end score may start planning and synthesis while the veto is still asserted, because nothing is audible yet. The two lanes meet only at t_c, where the veto is non-compensatory and no weighted score can override it.

REFERENCE CONTROLLER: COMPUTATIONALLY VERIFIED

40,960	1,000,000	0	5 / 5
exhaustive state/guard cases	randomized transitions	checked contract violations	seeded safety mutations detected

Evidence at a glance

Verified in the reference controller. 40,960 exhaustive state/guard evaluations; 1,000,000 randomized transitions; zero checked contract violations; five of five seeded safety mutations detected.

Established by model sensitivity analysis. Positive cross-detector dependence can sharply increase simultaneous misses; temporal persistence can thicken false-veto delay tails. Monte Carlo cross-checks are informative only above the estimator's resolution floor (Sec. 7).

Not yet established. Corpus-level detector performance, real audio-path timing, interruption reduction, or user preference.

Suggested citation

E. M. Iuliano and T. Peterson, "Dependence-Aware Turn-Taking for Voice Agents with Reversible Audio Commitment," ethereal computing inc., Public Research Edition, version 4.2, Aug. 2026.

Revision history

4.2 (Aug. 2026). Editorial and diagrammatic revision. Notation disambiguated (Y label values; active-channel set $\mathscr{D}$). Veto polarity of G and W stated explicitly; the nonempty-channel condition folded into Eq. (5) rather than carried separately. Admissibility predicates renamed for one-to-one correspondence with the figures. Guard dimensions enumerated (Table 2); abort ledger closed to its total. Monte Carlo resolution floor disclosed. Figures redrawn. No claim, count, or model result was changed.

4.1 (Aug. 2026). First public research edition.

Contents

1 INTRODUCTION

Human turn exchange is fast enough that listeners must project likely completion before a speaker stops rather than react only after a long silence. Ordinary conversation shows tightly timed transitions across languages, although overlap and gaps vary with task and culture [1–3]. A voice agent faces the same temporal problem while incremental automatic speech recognition (ASR) revises words, dialogue state changes, speech synthesis incurs startup time, and audio becomes difficult or impossible to recall after it enters a device queue.

The two principal errors need not have equal deployment cost. Beginning playback while a user continues can mask speech, force repetition, and make the system appear to seize the floor. Waiting after a true yield adds latency. In some settings delay is itself costly, so the ratio cannot be universal. The system should expose an unsafe-commit–delay frontier and let a deployment choose a point subject to explicit constraints instead of hiding the trade inside a balanced classification score.

This paper presents Predictive Conversational Patience (PCP) as an implemented control plane around, rather than a replacement for, turn-end predictors. Its central distinction is between private preparation and irreversible audio commitment. A predictive score may start response planning and synthesis, but a separate continuation veto remains non-compensatory at the instant the first sample is transferred to the output path. Missing, stale, unhealthy, uncalibrated, or configuration-mismatched detector output is not interpreted as evidence that the floor is clear.

This paper makes four contributions. First, it specifies a five-state controller with version-bound plans, private audio buffers, and a linearized final recheck at first-sample commitment. Second, it provides an auditable Python reference implementation and computational verification: 40,960 exhaustive abstract evaluations, 1,000,000 stateful randomized transitions, and five executable mutation tests. Third, it separates dependence-free safety properties from independence-only probability formulas and quantifies sensitivity to cross-detector dependence and temporal persistence. Fourth, it locks the corpus, hardware, and user-evaluation protocol needed to test conversational performance without gaming nonresponse or latency tails.

The reported verification establishes conformance of the reference controller to its stated transition contract. The stress tests establish consequences of declared probabilistic models. Neither is evidence that the detector stack is accurate, that the audio path meets a real-time budget, or that users prefer the system. Those questions remain empirical. Sec. 2 reviews relevant work; Sections 3 to 5 specify the design; Sections 6 and 7 report computational results; Sections 8 and 9 define the remaining validation and limitations; and Sec. 10 concludes.

2 PRIOR WORK AND SCOPE

2.1 Turn-Taking Prediction

Classic interaction research describes turn exchange as rule-governed and identifies prosodic, syntactic, pragmatic, and visual cues associated with yielding and continuation [1,4,5]. Prosody-based endpoint detectors can reduce reliance on fixed silence thresholds [6]. Finite-state dialogue controllers and continuous recurrent predictors subsequently moved spoken systems from endpoint reaction toward turn projection [7,8]. Voice Activity Projection (VAP) predicts future speaker activity from conversational audio, and later work conditions predictions on linguistic context, response content, or prompts [9–12]. Recent full-duplex and speculative end-turn systems reinforce the practical need to prepare work before the boundary is certain [13,14].

PCP does not claim a superior turn-taking predictor. Any calibrated detector or learned model may supply its end score or continuation channels. The contribution is the authorization boundary between uncertain inference and audible output: predictor confidence may allocate reversible computation, but it cannot compensate for a valid continuation veto at commit.

2.2 Incremental and Speculative Execution

Incremental dialogue systems can plan or generate before a complete utterance is available [15]. Speculative computation similarly performs work that may later be discarded [16]. In a voice system, however, generated text, privately synthesized audio, transfer to an operating-system queue, and acoustic emission are different commitments. Treating them as one event obscures races in which the user resumes or ASR revises after synthesis begins but before sound becomes audible.

The final queue transfer is modeled as a linearizable operation: the decisive guard snapshot and the first-sample write have one externally meaningful ordering point [17]. Finite-state exploration, property-oriented randomized testing, and mutation testing are used to test the implementation against this contract [18–20]. These techniques do not prove the behavior of an unmodeled audio stack, but they can expose control-path omissions before corpus or participant experiments.

2.3 Positioning

The architecture is narrower than a complete dialogue agent. It does not define the ASR model, language model, text-to-speech (TTS) synthesizer, loudspeaker, or conversational policy. It defines the data required from these components, the conditions under which private work may progress, and the conditions under which the first sample may become non-recallable. This scope follows from the implemented control-plane contribution: an auditable audio-commitment controller rather than a speculative claim about end-to-end conversational quality.

3 SYSTEM MODEL AND RISK OBJECTIVE

At each candidate boundary, let the reference label be $Y \in \{\text{CONT}, \text{YIELD}\}$ for continuation or yield. Let t_{yield} denote the annotated true-yield time. The policy commitment time t_c is the linearization point at which the first response sample is transferred to a queue from which recall is no longer guaranteed. The acoustic onset time t_a is the first response energy observed at a declared reference microphone or coupler. Their difference,

$$L_{\text{out}} = t_a - t_c, \tag{1}$$

is output-path latency. Electrical loopback can estimate internal timing but is not evidence that sound reached the user.

An unsafe commit U occurs when t_c falls inside a reference continuation interval. An audible interruption I occurs when response audio overlaps reference speech at the acoustic point. U is attributable to the decision layer and measurable in corpus replay; I also depends on L_{out}, device buffering, echo cancellation, and the room. After a true yield, the policy and acoustic delays are

$$D_c = t_c - t_{\text{yield}}, \qquad D_a = t_a - t_{\text{yield}} = D_c + L_{\text{out}}. \tag{2}$$

A response-required opportunity is a labeled yield for which the dialogue policy should produce output. When no commit occurs by a preregistered horizon H, the observation is not assigned a favorable finite delay. It is represented as $D_c = +\infty$ for extended-real quantiles, right-censored at H for time-to-commit curves, and reported separately as nonresponse. This prevents a conservative policy from appearing fast by suppressing difficult opportunities.

All thresholds, freshness limits, persistence rules, and admissible degraded configurations are selected jointly. For parameter vector ψ, a development procedure may minimize a declared delay quantile subject to one-sided safety and tail constraints:

$$\begin{aligned} \min_{\psi} \quad & Q_\tau\big(D_{c,\psi} \mid Y = \text{YIELD}\big) \\ \text{s.t.} \quad & \text{UCB}_{1-\gamma}\big[\Pr(U_\psi = 1 \mid Y = \text{CONT})\big] \le \varepsilon, \\ & \text{UCB}_{1-\gamma}\big[\Pr(D_{c,\psi} > D_{\max} \mid Y = \text{YIELD})\big] \le \eta. \end{aligned} \tag{3}$$

The risk limit ε, delay limit $D_{\max}$, tail probability η, quantile τ, confidence level $1-\gamma$, clustering unit, and candidate family are fixed before test evaluation. Selection must use a nested split or a simultaneous bound over the locked family; a pointwise interval computed after sweeping candidates is invalid. If no candidate is feasible, the correct result is failure to meet the deployment requirement.

Equation (3) constrains U and D_c rather than I and D_a, because the decision layer controls only the former pair. The acoustic quantities depend on L_{out} and are therefore deferred to the hardware stage of Sec. 8.

4 HEALTH-AWARE VETO ARBITRATION

Fig. 1 separates preparation evidence from commit authorization. The reference interface uses six continuation-detector families: fundamental-frequency contour, energy and phonation, speaking-rate and pause context, boundary tone, final lengthening, and semantic completion. This inventory reflects established acoustic and pragmatic cue classes [4–6]; it is not a claim that six channels are universally optimal. An implementation may add, merge, or remove channels, but every active channel must satisfy the same admissibility contract.

Let $\mathscr{D}$ denote the set of active continuation channels. For detector $i \in \mathscr{D}$, let $c_i \in [0,1]$ be continuation confidence, θ_i its veto threshold, and

$$A_i = P_i \wedge H_i \wedge F_i \wedge K_i \wedge M_i \tag{4}$$

indicate that a score is present (P_i), healthy (H_i), fresh (F_i), produced under an approved calibration artifact (K_i), and matched to the active detector configuration (M_i). Let $G = 1$ indicate that a hard acoustic or cancellation guard is asserted, and let $W = 1$ indicate that the observed health pattern matches a separately calibrated and validated configuration. Both are stated as indicators so that polarity is unambiguous: G blocks when true, W blocks when false. The commit veto is

$$V = \neg W \vee G \vee [\mathscr{D} = \varnothing] \vee \bigvee_{i \in \mathscr{D}} (\neg A_i \vee [c_i > \theta_i]). \tag{5}$$

Thus absence is not a clear vote. A detector fault, expired score, calibration mismatch, or unseen health combination blocks commit. The explicit $[\mathscr{D} = \varnothing]$ term matters: without it an empty disjunction evaluates to false, so a configuration that lost every channel would present as unvetoed. A deployment may switch only to an explicitly whitelisted degraded configuration with its own thresholds and validation; it may not silently delete a failed channel from the gate.

The veto is non-compensatory only at irreversible commit. A downstream end score $s(t)$ may rise while $V = 1$ and may start private preparation because no audio is exposed. At commit, however, no weighted score can override V. This division retains the latency benefit of prediction without allowing several weak completion cues to outvote one strong or invalid continuation channel.

No independence assumption is needed for the structural property: adding an active veto condition cannot enlarge the set of states in which commit is authorized. Probability formulas require stronger assumptions and are treated separately in Sec. 7. Detector scores also require calibration on speaker- or dyad-disjoint development data. Platt scaling and isotonic regression are examples, not mandated methods [21,22]. Calibration artifacts, domains, and freshness limits are versioned so that a score cannot be interpreted under a threshold learned for another configuration.

5 REVERSIBLE AUDIO-COMMITMENT CONTROLLER

5.1 States and Version Binding

The controller has five states, shown in Fig. 2: LISTENING, PREPARING, ARMED, COMMITTED, and ABORTED. LISTENING performs no response work. PREPARING may generate text and synthesize audio into a private buffer. ARMED means the end score and private-audio conditions are sufficient to attempt the atomic commit, but no sample has entered the device queue. COMMITTED begins at the successful first-sample write and is absorbing for the current response; the controller returns to LISTENING when that response completes or is cancelled downstream. ABORTED records the cause, invalidates the plan, flushes private audio, and then returns to LISTENING.

Every plan and buffer carries a version tuple

$$\boldsymbol{v} = (v_{\text{ASR}}, v_{\text{dialogue}}, v_{\text{policy}}, v_{\text{TTS}}, v_{\text{detector}}). \tag{6}$$

Any incompatible increment makes the material stale. This is stricter than checking only the latest transcript: a dialogue-state revision, policy/generator change, synthesizer reset, or detector-configuration change can also invalidate the intended response or its timing authorization.

5.2 Atomic Commit

The ordinary frame evaluation may move the controller to ARMED. The actual commit then obtains a final snapshot and tests all guards immediately before the first-sample write. Commit succeeds only when: cancellation is absent; the active detector vector is nonempty and fully admissible; $V = 0$; the final acoustic guard is clear; the output queue is ready; private audio is present; the plan version equals the current version; and $s(t)$ meets the commit threshold. The final snapshot and queue transfer are one linearized operation. If any condition changes between ordinary evaluation and this point, the attempt aborts rather than reusing an earlier clear decision.

This definition deliberately separates t_c from t_a. The controller can guarantee that it writes no sample before authorization, but it cannot infer acoustic onset from the write alone. Hardware evaluation must measure the output path and state whether its reference is a software timestamp, electrical tap, coupler, or microphone.

6 REFERENCE IMPLEMENTATION AND COMPUTATIONAL VERIFICATION

The reference controller is a compact Python implementation intended for inspection. It models the control plane and a boolean device-write event; it is not a real-time audio engine. A detector reading contains score, threshold, health, freshness, calibration, and configuration-match fields. The controller stores state, plan version, private-buffer presence, and an event log. The implementation never substitutes a low score for an inadmissible detector. Each step evaluates a fixed set of guards plus one pass over the active channel vector, so per-step cost is linear in $|\mathscr{D}|$.

6.1 Exhaustive Finite Abstraction

The verifier enumerates the five controller states crossed with the 13 Boolean guard dimensions listed in Table 2. This yields $5 \times 2^{13} = 40{,}960$ transition evaluations. The enumeration deliberately includes logically inconsistent combinations to test defensive behavior rather than assuming upstream nesting. Each result is checked against the local clauses in Table 1. No violation was found.

This is exhaustive only for the declared finite abstraction and one controller step. It does not prove floating-point calibration, asynchronous queue semantics, detector correctness, or a production scheduler. The value is narrower: every abstract combination that reaches the reference transition function satisfies the checked first-sample contract.

6.2 Stateful Randomized Traces

A second verifier executes 25,000 traces of 40 steps with fixed seed 20260801, for 1,000,000 transitions. Eighty percent use a structured phase schedule that reaches ordinary prepare–arm–commit cycles; 20% are adversarial. Low-rate perturbations inject detector faults, cancellations, and version changes during ordinary evaluation. Additional races alter detector state, plan version, final acoustic guard, queue readiness, cancellation, score, or audio readiness between arming and the final snapshot.

All five states were reached. The run produced 80,478 valid first-sample commits and 21,552 abort transitions, comprising 9,575 final-guard aborts, 6,551 veto-or-health aborts, 3,227 stale-plan aborts, and 2,199 aborts recorded under the remaining causes. No checked violation occurred. The counts establish exercise of both positive and negative paths under a reproducible generator; they are not estimates of operational event rates because the generator is intentionally synthetic.

6.3 Mutation Tests

Five seeded mutants remove one safety-relevant behavior each: fail open on an inadmissible detector, omit the final veto recheck, omit the final plan-version comparison, enqueue on the prepare-to-arm transition, and treat a stale final score as clear. Each mutant has a concrete counterexample and all five were killed. Table 3 summarizes the verification results. Mutation adequacy is limited to these five faults; it does not imply completeness over all possible implementation defects.

7 DEPENDENCE AND PERSISTENCE STRESS TESTS

The veto topology couples detector errors. Independence provides a useful reference but can be dangerously optimistic for simultaneous misses because acoustic cues often co-vary. Two model-based stress tests quantify the direction and scale of this sensitivity. They are analytical calculations with Monte Carlo cross-checks, not measurements from a speech corpus.

7.1 Exchangeable Cross-Detector Dependence

Let X_i be a detector event with marginal probability p across N channels. Under an exchangeable beta–binomial model, a latent event probability is distributed as $\Theta \sim \mathrm{Beta}(a,b)$ and $X_i \mid \Theta$ are conditionally independent, where

$$a = p\,(\rho^{-1} - 1), \qquad b = (1-p)\,(\rho^{-1} - 1). \tag{7}$$

The intraclass dependence is ρ, since $\mathrm{Corr}(X_i, X_j) = 1/(a+b+1) = \rho$; the case $\rho = 0$ is handled by the independent limit. The probability that at least one detector fires and the probability that every detector fires are

$$\Pr(\textstyle\bigcup_i X_i) = 1 - \dfrac{B(a, b+N)}{B(a,b)}, \tag{8}$$

$$\Pr(\textstyle\bigcap_i X_i) = \dfrac{B(a+N, b)}{B(a,b)}. \tag{9}$$

For a six-channel illustration, per-detector false-veto probability is set to 0.05 and miss probability to 0.10. At independence, the false-veto union is 0.2649 and the all-channel miss probability is 10^{-6}. At $\rho = 0.60$, the union falls to 0.1008 because false vetoes cluster, while simultaneous misses rise to 0.03232. Fig. 3 shows the full sweep. Positive dependence therefore reduces one cost while sharply weakening the apparent geometric protection against the costly error. The independent product is not a safety guarantee. Note also that a single ρ is applied to both event families here; real false-veto and miss dependence need not share a value.

7.2 Temporal Persistence of False Vetoes

False vetoes can also persist across adjacent re-evaluations. Let q be the false-veto probability at a true yield and let $\phi \in$

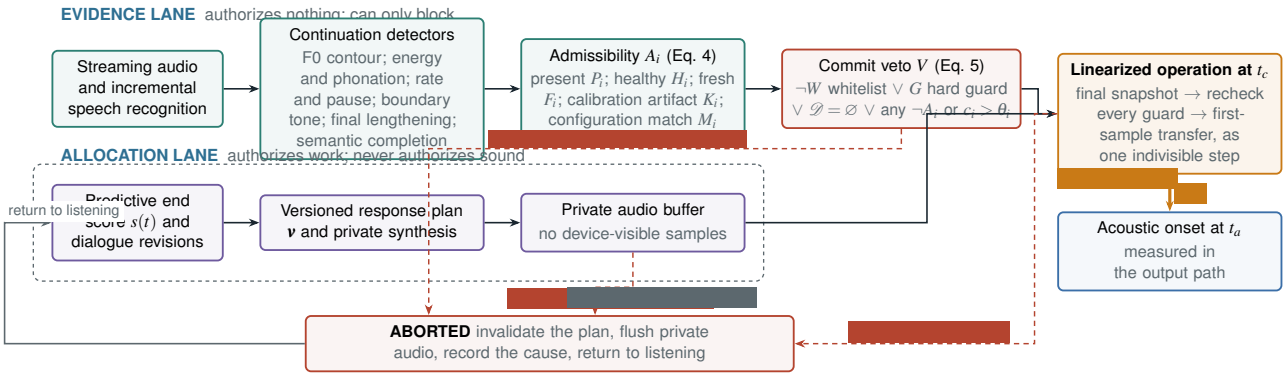


Figure 1. PCP system contract. The two lanes run concurrently. The allocation lane may start planning and synthesis on a rising end score while the evidence lane still asserts a veto, because nothing has been exposed. Calibrated continuation channels pass presence, health, freshness, calibration-artifact, and configuration-match checks before entering a non-compensatory commit veto. Versioned material stays off the device queue until the final recheck and the first-sample write execute as one linearized operation at t_c; acoustic onset occurs later at t_a. Dashed paths abort and flush private work.

Table 1. Controller contract and its executable checks. "Final snapshot" means the snapshot sampled inside the linearized first-sample operation, not an earlier frame evaluation.

Contract clause	Required behavior	Verification hook
Commit origin	A first-sample device write occurs only on ARMED→COMMITTED; entering COMMITTED without that write is invalid.	Exhaustive transition predicate; early-enqueue mutant
Final admissibility	The active channel set is nonempty and every active detector is present, healthy, fresh, calibration-artifact approved, configuration-matched, and nonvetoing in the final snapshot.	Exhaustive guards; fail-open and stale-score mutants
Version consistency	The prepared plan version equals the final ASR, dialogue, policy, TTS, and detector version tuple.	Exhaustive guards; stale-plan and omitted-version mutants
Race closure	Veto, cancellation, guard, queue, score, audio-ready, and version conditions are rechecked at the linearization point.	Randomized injected races; omitted-final-veto mutant
Precommit reversibility	No device write occurs in LISTENING, PREPARING, ARMED, or ABORTED; every abort flushes private material.	Transition predicates across all states

Table 2. The 13 Boolean guard dimensions of the finite abstraction. Crossed with the five controller states these give $5 \times 2^{13} = 40{,}960$ evaluations. Dimensions 8–12 are sampled inside the linearized first-sample operation, not at the earlier frame evaluation, which is what makes race closure testable.

#	Guard dimension	Sampled at
1	Initial veto V	frame evaluation
2	Initial admissibility of $\mathscr{D}$	frame evaluation
3	Preparation score region $s \geq \theta_{\text{prep}}$	frame evaluation
4	Arm score region $s \geq \theta_{\text{arm}}$	frame evaluation
5	Commit score region $s \geq \theta_{\text{commit}}$	frame evaluation
6	Private-audio readiness	frame evaluation
7	Plan currency	frame evaluation
8	Final acoustic guard G	final snapshot
9	Final veto V	final snapshot
10	Final admissibility of $\mathscr{D}$	final snapshot
11	Final plan currency	final snapshot
12	Queue readiness	final snapshot
13	Cancellation	either point

$[0,1)$ parameterize lag-one persistence. The probability of remaining in the false-veto state is

$$r = q + \phi\,(1-q). \tag{10}$$

With re-evaluation interval Δ, the added-delay mean and tail are

$$\mathbb{E}[D] = \Delta\,\frac{q}{1-r}, \qquad \Pr(D \geq k\Delta) = q\,r^{k-1}, \quad k \geq 1. \tag{11}$$

Using $q = 0.2649$ from the independent six-channel false-veto example and $\Delta = 50$ ms, mean added delay rises from 18.0 ms at $\phi = 0$ to 120.1 ms at $\phi = 0.85$. The probability

Table 3. Computational verification results. Counts describe synthetic controller execution, not dialogue performance.

Measure	Result
Abstract states	5
Boolean guard dimensions	13
Exhaustive evaluations	40,960
Exhaustive violations	0
Randomized traces	25,000
Transitions	1,000,000
Valid commits	80,478
Abort transitions	21,552
Randomized violations	0
Seeded mutants detected	5 of 5

of at least 100 ms rises from 0.0702 to 0.2357, and the probability of at least 250 ms rises from 0.00130 to 0.1660. Fig. 4 visualizes both the mean and the two tails. A one-frame false-veto rate is therefore insufficient; evaluation must report run lengths and delay tails.

7.3 Resolution of the Monte Carlo Cross-Check

For each dependence and persistence point, the implementation also ran $n = 1{,}000{,}000$ Monte Carlo trials with seed 20260801. Maximum absolute discrepancies from the formulas were 0.000399 for union probability, 0.0000378 for simultaneous-miss probability, 0.318 ms for mean delay, and 0.000447 for the reported tails.

These checks primarily guard the calculation and plotting code; they do not validate the model assumptions, and they are informative only above the estimator's resolution. The standard error of a Monte Carlo estimate near p is

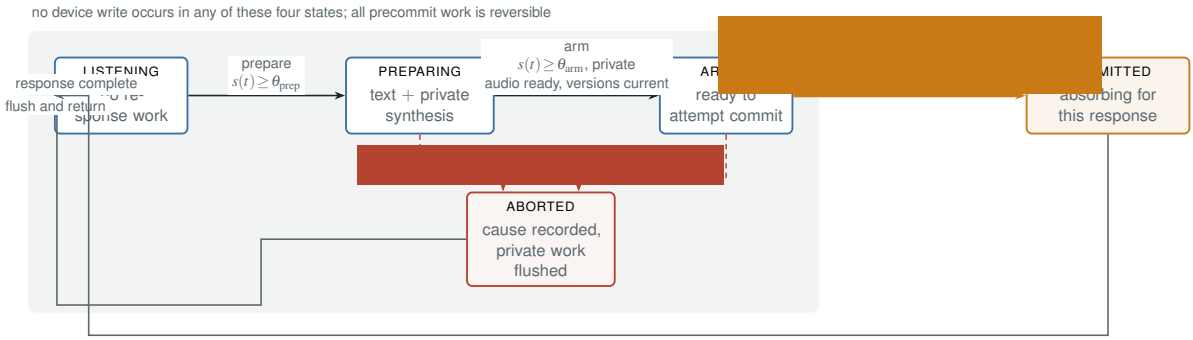


Figure 2. Five-state controller. Preparation and arming are silent and reversible: no sample reaches the device queue in LISTENING, PREPARING, ARMED, or ABORTED. A valid veto, an inadmissible or empty detector set, a cancellation, a version mismatch, or a guard change before t_c aborts precommit work, flushes private material, and returns the controller to LISTENING. The only path to audible output is the linearized ARMED→COMMITTED operation, drawn in bold; COMMITTED is absorbing for the current response and releases back to LISTENING on completion.

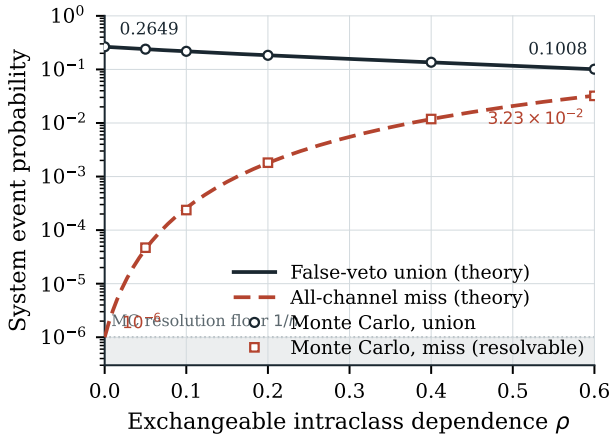


Figure 3. Model-based dependence sensitivity for $N = 6$ exchangeable detector events. The solid curve is the union probability for marginal false-veto probability 0.05; the dashed curve is the simultaneous-miss probability for marginal miss probability 0.10. Markers are Monte Carlo cross-checks with 10^6 trials per point, plotted for the miss curve only where the expected count exceeds 30. The shaded band is the $1/n$ resolution floor below which a 10^6-trial estimate carries no usable precision. These are stress-model outputs, not measured detector statistics.

$\sqrt{p(1-p)/n}$, so the independent all-channel miss probability of 10^{-6} has an expected count near one and a relative standard error near 100%. Small absolute agreement at that point is arithmetic, not evidence. The cross-check is treated as binding only where the expected count is at least 30, that is $p \gtrsim 3 \times 10^{-5}$, which on the miss curve begins near $\rho = 0.05$. Fig. 3 marks the floor at $1/n$ and plots miss-curve Monte Carlo points only where they are resolvable.

8 EMPIRICAL VALIDATION PROTOCOL

The remaining work is organized as three locked stages so that each claim is tested at the appropriate layer. Each stage has a distinct reference event, measurement target, and failure criterion.

8.1 Corpus Replay

Stage 1 replays a turn-taking corpus through the detector interfaces and controller. A condition-blind candidate-boundary set is created once and shared by all systems. Speaker or dyad identities are disjoint across training, tuning, feasibility, and test splits. Detector scores are calibrated on development data, and all policy parameters in Eq. (3) are selected jointly. Baselines include a silence/endpoint policy, VAP or a comparable future-activity model, and weighted late fusion. They share the ASR alignment, candidate set, response-required labels, output-path assumptions, and tuning budget.

Primary policy outcomes are unsafe-commit rate over every continuation opportunity and extended-real D_c over every response-required yield. Reports include one-sided clustered safety bounds, the full unsafe-commit–delay frontier, nonresponse, false-veto run lengths, timeout or fail-closed stalls, subgroup results, and sensitivity to candidate definitions. An error analysis must distinguish detector misses, inadmissible readings, stale plans, final-snapshot races, and downstream policy failures.

8.2 Hardware-in-the-Loop Audio Timing

Stage 2 places the selected policy on declared hardware and measures t_c, t_a, and L_{out}. The procedure records audio interface, sample rate, block size, operating system, driver, queue depth, TTS first-sample latency, timestamp clock, and coupler or microphone reference. A loopback-only result is labeled internal latency. Repeated measurements report median, 95th and 99th percentiles, cold-start behavior, and worst observed queue excursions. Races are reinjected while a physical output trace confirms that aborted precommit buffers remain inaudible.

8.3 Participant Evaluation

Stage 3 compares the locked systems within subjects after an a priori power analysis. Scenarios include read dialogue, task-oriented exchange, and open conversation, counterbalanced across conditions. Objective outcomes include acoustic interruption, overlap duration, response delay, repair requests, and nonresponse; subjective outcomes include perceived interruption, responsiveness, control, and trust. Analysis uses participant and item clustering, reports effect sizes and uncertainty, and treats brief backchannels separately from floor-taking responses. No participant number or favorable effect direction is asserted before the power analysis and preregistration.

9 LIMITATIONS, ETHICS, AND REPRODUCIBILITY

The reference implementation verifies a control contract, not end-to-end speech behavior. It contains no production ASR, prosody extractor, semantic completion model, TTS engine, or audio-driver integration. The finite abstraction

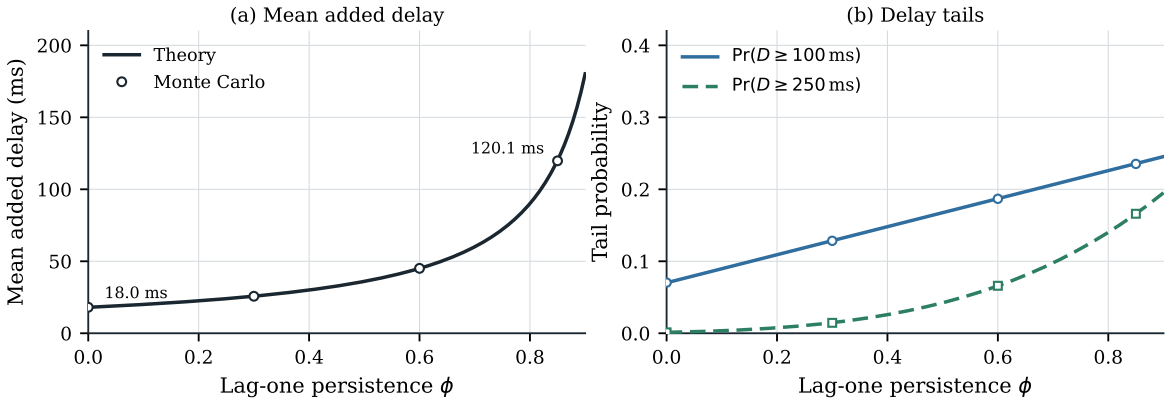


Figure 4. Model-based effect of lag-one false-veto persistence for $q = 0.2649$ and $\Delta = 50$ ms. (a) Mean added delay. (b) Tail probabilities at 100 ms and 250 ms. Open markers are Monte Carlo cross-checks at the four reported operating points. Temporal correlation thickens the delay tail even when the marginal false-veto rate is unchanged, which is why a single-frame rate is not a sufficient latency report.

evaluates one-step guard combinations and the randomized traces use a designed generator; neither substitutes for model checking of a deployed concurrent system. The five mutation operators target known hazards but cannot establish fault completeness.

The beta–binomial model assumes exchangeability and one nonnegative dependence parameter. Real detector dependence can be heterogeneous, conditional on phonetic context, and nonexchangeable. The first-order persistence model omits session drift and higher-order runs. Their purpose is to invalidate casual independence reasoning and identify quantities to measure, not to forecast field performance.

A fail-closed policy can produce stalls or nonresponse, which may be unacceptable in time-critical contexts. Equation (3) therefore constrains both safety and latency, and evaluation counts every response-required opportunity. No timeout forces speech through an invalid configuration. A deployment that requires bounded response must establish a separately validated fallback rather than silently weakening the guard.

Per-speaker cadence, voice features, and transcripts can be sensitive. The controller does not require persistent speaker identification. Evaluation should default to session-local identifiers, minimize raw-audio retention, document consent and withdrawal, and separate performance by language, accent, speech impairment, speaking style, and acoustic condition. A detector configuration validated in one population or language is not automatically admissible in another.

The accompanying artifact package contains the reference controller, verifier, mutation tests, stress-test code, raw comma-separated-value outputs, figure source, fixed package versions, and checksums. The computational environment used Python 3.13.5, NumPy 2.2.6, and Matplotlib 3.10.3. A versioned public archive with a persistent identifier is planned; until that deposit is finalized, the artifact package is available from the corresponding author.

9.1 Use of Generative Tools

Anthropic Claude (Anthropic PBC, accessed 2026 Aug. 1–2) was used for drafting assistance, structural editing, numerical re-derivation, and figure regeneration. OpenAI ChatGPT (OpenAI, accessed 2026 Aug. 1) was used for technical red-teaming, source verification, implementation support, calculation checking, figure generation, and editing. The human authors critically reviewed and verified the manuscript, code, calculations, figures, and references and accept full responsibility for the work.

10 CONCLUSION

This paper converts asymmetric conversational risk into an auditable audio-commitment boundary. Predictive models may allocate reversible response work, but audible authorization remains gated by detector admissibility, a noncompensatory continuation veto, current version identity, private-audio readiness, and a final race-closing snapshot. The first-sample queue transfer is defined separately from acoustic onset so that policy safety and output-path timing can be measured rather than conflated.

The implemented reference controller satisfied its transition contract in all 40,960 abstract guard combinations and 1,000,000 randomized transitions, while five safety mutations produced detectable counterexamples. The dependence stress test shows why an independent all-channel miss product can be severely optimistic, and the persistence stress test shows why marginal false-veto rate understates latency tails.

These are genuine implementation and model-sensitivity results, but they do not establish conversational superiority. Corpus replay, hardware-in-the-loop timing, and participant evaluation remain the necessary tests. The architecture is useful only if those stages find a feasible safety–latency frontier under real detector dependence and real audio-path behavior.

REFERENCES

[1] H. Sacks, E. A. Schegloff, and G. Jefferson, “A Simplest Systematics for the Organization of Turn-Taking for Conversation,” *Language*, vol. 50, no. 4, pp. 696–735 (1974), doi.org/10.2307/412243.

[2] T. Stivers, N. J. Enfield, P. Brown, et al., “Universals and Cultural Variation in Turn-Taking in Conversation,” *Proc. Natl. Acad. Sci. USA*, vol. 106, no. 26, pp. 10587–10592 (2009 Jun.), doi.org/10.1073/pnas.0903616106.

[3] S. C. Levinson and F. Torreira, “Timing in Turn-Taking and Its Implications for Processing Models of Language,” *Front. Psychol.*, vol. 6, article 731 (2015 May), doi.org/10.3389/fpsyg.2015.00731.

[4] S. Duncan, “Some Signals and Rules for Taking Speaking Turns in Conversations,” *J. Pers. Soc. Psychol.*, vol. 23, no. 2, pp. 283–292

(1972 Aug.), doi.org/10.1037/h0033031.

[5] A. Gravano and J. Hirschberg, “Turn-Taking Cues in Task-Oriented Dialogue,” *Comput. Linguist.*, vol. 37, no. 4, pp. 779–803 (2011 Dec.), doi.org/10.1162/COLI_a_00078.

[6] L. Ferrer, E. Shriberg, and A. Stolcke, “A Prosody-Based Approach to End-of-Utterance Detection That Does Not Require Speech Recognition,” in *Proc. Intl. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, pp. I-608–I-611, Hong Kong (2003 Apr.), doi.org/10.1109/ICASSP.2003.1198854.

[7] A. Raux and M. Eskenazi, “A Finite-State Turn-Taking Model for Spoken Dialog Systems,” in *Proc. Human Language Technologies: NAACL*, pp. 629–637, Boulder, CO (2009 Jun.), aclanthology.org/N09-1071 (accessed 2026 Aug. 1).

[8] G. Skantze, “Towards a General, Continuous Model of Turn-Taking in Spoken Dialogue Using LSTM Recurrent Neural Networks,” in *Proc. SIGDIAL*, pp. 220–230, Saarbrücken, Germany (2017 Aug.), doi.org/10.18653/v1/W17-5527.

[9] E. Ekstedt and G. Skantze, “Voice Activity Projection: Self-Supervised Learning of Turn-Taking Events,” in *Proc. Interspeech*, pp. 5190–5194, Incheon, Republic of Korea (2022 Sep.), doi.org/10.21437/Interspeech.2022-10955.

[10] E. Ekstedt and G. Skantze, “TurnGPT: A Transformer-Based Language Model for Predicting Turn-Taking in Spoken Dialog,” in *Findings of EMNLP*, pp. 2981–2990, online (2020 Nov.), doi.org/10.18653/v1/2020.findings-emnlp.268.

[11] B. Jiang, E. Ekstedt, and G. Skantze, “Response-Conditioned Turn-Taking Prediction,” in *Findings of ACL*, pp. 12241–12248, Toronto, Canada (2023 Jul.), doi.org/10.18653/v1/2023.findings-acl.776.

[12] K. Inoue, M. Elmers, Y. Fu, et al., “Prompt-Guided Turn-Taking Prediction,” in *Proc. SIGDIAL*, pp. 146–151, Avignon, France (2025 Aug.), doi.org/10.18653/v1/2025.sigdial-1.9.

[13] T.-E. Lin, Y. Wu, F. Huang, et al., “Duplex Conversation: Towards Human-Like Interaction in Spoken Dialogue Systems,” in *Proc. ACM SIGKDD*, pp. 3299–3308, Washington, DC (2022 Aug.), doi.org/10.1145/3534678.3539209.

[14] H. Ok, S. Yoo, and J. Lee, “Speculative End-Turn Detector for Efficient Speech Chatbot Assistant,” in *Proc. ACL* (2026); preprint arXiv:2503.23439 (accessed 2026 Aug. 1).

[15] G. Skantze and A. Hjalmarsson, “Towards Incremental Speech Generation in Dialogue Systems,” in *Proc. SIGDIAL*, pp. 1–8, Tokyo, Japan (2010 Sep.), aclanthology.org/W10-4301 (accessed 2026 Aug. 1).

[16] F. W. Burton, “Speculative Computation, Parallelism, and Functional Programming,” *IEEE Trans. Comput.*, vol. C-34, no. 12, pp. 1190–1193 (1985 Dec.), doi.org/10.1109/TC.1985.6312193.

[17] M. P. Herlihy and J. M. Wing, “Linearizability: A Correctness Condition for Concurrent Objects,” *ACM Trans. Program. Lang. Syst.*, vol. 12, no. 3, pp. 463–492 (1990 Jul.), doi.org/10.1145/78969.78972.

[18] E. M. Clarke, E. A. Emerson, and A. P. Sistla, “Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications,” *ACM Trans. Program. Lang. Syst.*, vol. 8, no. 2, pp. 244–263 (1986 Apr.), doi.org/10.1145/5397.5399.

[19] K. Claessen and J. Hughes, “QuickCheck: A Lightweight Tool for Random Testing of Haskell Programs,” in *Proc. ACM Intl. Conf. Functional Programming*, pp. 268–279, Montreal, Canada (2000 Sep.), doi.org/10.1145/351240.351266.

[20] Y. Jia and M. Harman, “An Analysis and Survey of the Development of Mutation Testing,” *IEEE Trans. Softw. Eng.*, vol. 37, no. 5, pp. 649–678 (2011 Sep./Oct.), doi.org/10.1109/TSE.2010.62.

[21] J. C. Platt, “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods,” in A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, Eds., *Advances in Large Margin Classifiers*, pp. 61–74 (MIT Press, Cambridge, MA, 2000).

[22] B. Zadrozny and C. Elkan, “Transforming Classifier Scores into Accurate Multiclass Probability Estimates,” in *Proc. ACM SIGKDD*, pp. 694–699, Edmonton, Canada (2002 Jul.), doi.org/10.1145/775047.775151.

About the authors

Eros Marcello Iuliano

Eros Marcello Iuliano is with ethereal computing inc., Moffett Field, California, USA. His work on this study concerns conversational-agent architecture, real-time voice interaction, software implementation, and system verification.

Thea Peterson

Thea Peterson is with ethereal computing inc., Moffett Field, California, USA. Her work on this study concerns analytical verification, technical editing, and reproducibility.

Public release note

This document is a public, pre-peer-review technical disclosure. It is provided for research and engineering discussion and should not be represented as an accepted or published peer-reviewed journal article. Public release of technical material may affect intellectual-property strategy; the authors should confirm filing and disclosure posture with qualified counsel before publication. This notice is procedural and not legal advice.

REFERENCE CONTROLLER: COMPUTATIONALLY VERIFIED

40,960	1,000,000	0	5 / 5
exhaustive state/guard cases	randomized transitions	checked contract violations	seeded safety mutations detected

MODEL SENSITIVITY

Dependence and persistence can dominate the tails

CROSS-DETECTOR DEPENDENCE

Simultaneous six-detector miss
$10^{-6} \rightarrow 3.23 \times 10^{-2}$
at intraclass dependence $\rho = 0.60$
beta–binomial model

TEMPORAL PERSISTENCE

Mean false-veto delay
18.0 ms $\rightarrow$ 120.1 ms
at lag-one persistence $\phi = 0.85$
first-order Markov model

EVIDENCE BOUNDARY

Established here: controller-contract conformance under an exhaustive finite abstraction and a seeded randomized generator, plus probabilistic sensitivity under two declared models with Monte Carlo cross-checks above the estimator's resolution floor.

Still required: corpus replay, hardware-in-the-loop timing, and participant evaluation.

ethereal computing inc.

www.etherealcomputing.com/research · info@etherealcomputing.com