Journey

The Trusted Identity Platform

Deepfakes in Authentication

September 16th, 2025 | Alex Shockley & Keziah Gopalla

Table of Contents

Introduction **2**

1. Authentication fundamentals (mapped to risk) **3**

1.1 The identity onion: from immutable self to practical representations 3

1.2 The three questions that actually drive assurance 4

1.3 Metrics that matter: FAR, FRR, EER—and PAD for liveness 5

1.4 Passkeys: what they are, how they work, and why they change the baseline 6

1.5 Voice-first realities (and a note on video) 7

1.6 Inclusivity and exception paths (without opening a back door) 8

1.7 Summary table: methods, assurance role, and what to measure 8

2. Deepfakes **9**

2.1 Working definition and scope 9

2.2 A brief history: from GANs to diffusion and neural codecs 9

2.3 What “audio deepfakes” really are (and how they’re built) 10

2.4 What “video deepfakes” really are (and how they’re built) 11

2.5 From training loop to live call: a minimal production pipeline 12

2.6 Why detection is hard (and keeps getting harder) 13

2.7 Presentation vs. injection: two fundamentally different problems 13

2.8 The economics: capability is cheap, accessible, and scalable 14

2.9 Watermarking and provenance: helpful, but not sufficient on their own 14

2.10 Where capability is heading (and why it matters for auth) 15

2.11 A note on language, risk, and measurement 16

3. Synthetic Identity at the Gate: Deepfakes and Auth **16**

3.1 A quick reality check on prevalence and capability 16

3.2 Two distinct bypass modes: presentation vs. injection 17

3.3 Method-by-method: how deepfakes change the game 17

1. Knowledge factors (passwords, KBA) 17

2. SMS/TOTP one-time codes 18

3. OTP with telephone-network validation (ANI-anchored verification) 18

4. Push-based MFA (including number matching) 19

5. Passkeys (device-bound WebAuthn), benchmarked with on-device Face ID unlock 19

6. Voice biometrics (active) 19

7. Voice biometrics (passive) 20

8. Facial biometrics (video escalations) 20

9. Agent exception paths and overrides 20

3.4 STIR/SHAKEN: useful signal, not person authentication 21

3.5 Why detection alone won’t save you (and what to measure instead) 21

3.6 The shifting goal posts for authorization risk 22

3.7 Human-in-the-loop signals: helpful heuristics, not gates 22

3.8 Compliance note: server-side biometrics vs. on-device unlock 23
3.9 Takeaways 23
3.10 Deepfakes × Auth in the contact center: what changes, where to defend 23
3.11 Closing: a preview of the policy implications 25
4. What to do now: manual policy changes with outsized impact 25
4.1 Institute a “no phone-only overrides” rule for high-risk actions 25
4.2 Replace static KBA and SMS-only gates in recovery and resets 25
4.3 Treat STIR/SHAKEN as call integrity—not identity 26
4.4 Set acceptance rules for biometrics that assume a hostile channel 26
4.5 Engineer exception paths with equivalent assurance 26
4.6 Harden the agent environment against media injection 26
4.7 Train for conversational anomalies—use them to trigger step-ups 27
4.8 Publish standardized KPIs and hold quarterly policy reviews 27
4.9 Align biometrics governance to jurisdictional obligations 27
4.10 Red-team the channel you actually run 27
4.11 What “good” looks like: a simple policy-maturity model 27
4.12 Planning for the near future: AI-governed policy without the hype 28
4.13 Example Scenarios 29
5.0 From Voice-Only to Voice+Digital: Journey’s Approach to Trust 29
5.1 Spectrum coverage: the right step-up, complete in seconds 30
5.2 Policy partnership: moving authorization milestones—on purpose 31
5.3 MCP: connecting trust controls to your AI decisioning 32
5.4 What this looks like in practice 32
Conclusion 33

Introduction

Deepfakes—AI-generated voices, faces, and behaviors that convincingly mimic real people—have moved from novelty to material enterprise risk. This paper examines how synthetic media is eroding the reliability of identity verification, with a pragmatic focus on authentication in contact centers. Our aim is to equip engineering, security, and fraud teams with a clear threat model and a set of controls that still work when an attacker can sound and look exactly like a trusted customer or executive. Accordingly, this paper treats deepfake detection as a risk signal that can trigger step-up controls—not as a binary gate.

Contact centers are now the frontline. As more sensitive transactions shift to remote channels (voice, video, chat), adversaries gain two decisive advantages: scale, thousands of calls or sessions can be launched from anywhere, and anonymity, which hides preparation and rehearsal behind polished synthetic personas. The result is a rapidly escalating problem in which deepfakes amplify both social engineering and technical bypass, turning routine service interactions into high-impact fraud opportunities.

Recent incidents illustrate the pattern: in 2019, a voice clone of a German executive convinced a UK energy firm to authorize a transfer; in 2020, a cloned “director” voice persuaded a UAE bank manager to move $35M across accounts; and in 2024, a deepfaked video conference led a Hong Kong finance worker to approve $25M in payments. In each case, the attacker succeeded not by guessing a secret, but by presenting high-assurance signals (voice, face, presence) that felt authentic—and by exploiting exception paths and time pressure.

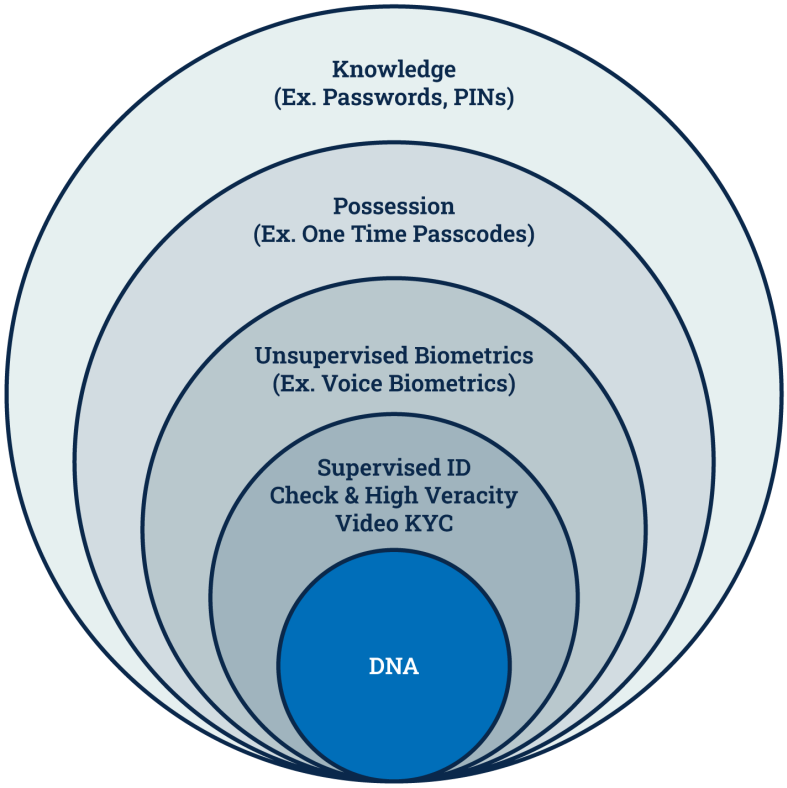
This is the core shift for defenders: traditional frameworks judge whether an input matches a stored reference (a password, a voiceprint, a face), often assuming the input comes from a live, legitimate source. Deepfakes break that assumption. Effective defenses must therefore move beyond content matching to verify origin (cryptographic binding and secure input), liveness (robust presentation-attack detection), and context (behavioral consistency, transaction plausibility, and policy)—especially in voice channels where social engineering pressure is highest.

In this paper, we (1) map the contact-center threat model and where deepfakes enter the call flow, (2) analyze common authentication methods through a deepfake lens—how each is attacked and which controls measurably reduce risk—and (3) outline a risk-based, AI-governed approach to orchestration so organizations can raise assurance dynamically without overwhelming customers or agents.

1. Authentication fundamentals

1.1 The identity onion: from immutable self to practical representations

A useful way to reason about authentication is as a set of concentric layers radiating out from an immutable core—proof of self—to progressively more convenient but less definitive representations of self. At the conceptual center (Layer 0) is infallible identity—for example, a live, proctored DNA test. This anchor is rarely practical for operations, but it grounds the idea that everything else we use in production is a proxy: evidence that is *good enough* for the channel, risk, and user experience at hand.



Moving one layer outward are high-assurance representations: supervised, in-person checks of a government ID; monitored, real-time biometric capture; video-based KYC with strong liveness. These have higher operational cost and friction, but when executed correctly provide strong confidence that the person present is the account owner. Beyond that, we encounter *unsupervised* biometrics (e.g., passive voice during a call, selfie upload), then possession factors (codes, pushes, hardware keys, and device-bound cryptography), and finally knowledge factors (passwords, PINs, KBA). As you travel outward, convenience rises, but so does the attack surface and the likelihood that the signal can be replayed, relayed, or misattributed. The practical art of authentication is deciding which layer(s) are appropriate for which risks, and how to combine them without overwhelming customers or agents.

Two cross-cutting realities shape this onion in contact centers. First, channel constraints: inbound telephony introduces compression, noise, and limited ability to verify device integrity; WebRTC adds its own artifacts. Second, exception paths: accommodations for accessibility, device loss, or biometric opt-outs are essential, but each one must be engineered so it doesn't become the path of least resistance.

An authentication design that looks strong on paper can be undone by a single permissive fallback.

1.2 The three questions that actually drive assurance

Most systems historically asked only: Does the content match? (Does the password equal the stored hash? Does this voice or face look like the enrollment template?) In this paper, 'origin' explicitly includes device and sensor provenance on the endpoint (for example, attested camera/microphone and a 'no virtual devices' policy)—not merely transport-layer protections like TLS. In modern voice and video channels, defenders must answer three independent questions for any factor:

1. Content: Does this input match our reference?
2. Origin: Did it originate from a trustworthy capture path? (e.g., real microphone/camera on a known device, not a virtual device or injected stream)
3. Liveness/Interactivity: Was it produced live in response to an unpredictable prompt?

A strong match score without trustworthy origin and liveness can produce a false sense of security; conversely, a cryptographic origin with weak policy around fallbacks can still be social-engineered. Treat these as orthogonal axes you can combine to reach the assurance the use case demands.

Implementation note (biometric acceptance): For biometrics used in contact centers—especially voice during calls and face over video—acceptance should depend on both robust liveness and a trustworthy capture path. Streams from virtual audio devices or virtual cameras should be flagged or blocked for any action above low risk, because liveness cannot be trusted if the sensor path itself is untrusted.

1.3 Metrics that matter: FAR, FRR, EER—and PAD for liveness

Biometric outcomes are probabilistic, not binary. Instead of a single "accuracy" number, teams should tune and report:

- FAR (False Acceptance Rate): Probability an impostor is accepted.
- FRR (False Rejection Rate): Probability a genuine user is rejected.
- EER (Equal Error Rate): The operating point where FAR = FRR; useful for comparison, not a deployment target.

Treat EER as a benchmarking statistic; in production you'll select an operating point above or below EER to balance fraud and friction budgets. Operationally, you pick a threshold on the model's ROC/DET curve that trades off user friction (FRR) against fraud (FAR) for your channel and audience. A system that claims "one-in-a-million" false accepts at a lab setting may behave very differently over telephony or consumer webcams. Publish FAR/FRR at your deployed threshold and revisit as channel conditions or customer mix change.

For liveness, use the ISO/IEC 30107 Presentation Attack Detection (PAD) metrics:

- APCER: % of attack presentations (spoofs) misclassified as bona fide.
- BPCER: % of bona fide presentations misclassified as attacks.
- ACER: The average of APCER and BPCER.

Critically, measure PAD in the same conditions you operate (telephony codecs, WebRTC, typical background noise) and against attacks representative of your risk. Liveness numbers measured on crisp lab audio or studio video will overstate protection in the field.

1.4 Passkeys: what they are, how they work, and why they change the baseline

Passkeys are the user-friendly name for FIDO2/WebAuthn public-key authentication on the web and in apps. Instead of sharing a reusable secret (password), the client creates a unique key pair per service: the server (relying party) stores the public key, and the private key stays on the user's authenticator. At sign-in, the server sends a challenge bound to its domain (RP ID). The authenticator signs that challenge with the private key only after local user verification (biometric or PIN), and the server verifies it with the stored public key. Because the key pair is cryptographically tied to the origin, passkeys are phishing-resistant and immune to server-side credential replay.

Where the keys live. Users can satisfy passkey flows in several ways:

- Platform authenticators (device-bound): The key lives in the device's secure hardware (e.g., Secure Enclave/TPM) and is unlocked locally via a biometric or device PIN (e.g., Face ID, Windows Hello).
- Roaming authenticators (cross-platform): Hardware security keys (e.g., FIDO2 keys) carried by the user; typically unlocked with a PIN and optional fingerprint sensor.

- Synced multi-device passkeys: Many ecosystems now offer passkeys that are synchronized across a user's devices via an encrypted cloud service (e.g., the platform's keychain or a modern password manager). Synced passkeys preserve relying-party key isolation and phishing resistance; the trade-off is shifting some residual risk to account recovery and ecosystem integrity (for example, cloud-account access policies and device trust posture), which your policy must explicitly govern. This improves portability while keeping per-service key separation and phishing resistance. Importantly, 'synced' does not mean a shared secret—the credential remains RP-bound and origin-verified, so phishing resistance is preserved.

Presence and phishing resistance. WebAuthn distinguishes user presence (the user is physically there to consent) and user verification (the user proved it's them—typically with a biometric or local PIN). In practice, well-configured passkeys require user verification and are bound to the service's origin, so tricking a user into approving on a look-alike website will not validate—the RP ID doesn't match.

Recovery and fallbacks. The residual risk with passkeys is rarely the cryptography—it's recovery and exception flows. SMS or email fallbacks, device-change helpdesk overrides, and poorly governed account recovery can re-introduce weaker factors. Treat these flows as first-class surfaces in threat modeling and orchestration; they deserve the same rigor as primary authentication.

Passkeys via Face ID offer a practical benchmark. When a passkey is unlocked with a local, on-device biometric (e.g., Face ID/TouchID/Windows Hello), the organization gains cryptographic possession-and-presence without storing or processing biometric templates centrally. The biometric never leaves the device; it is used only to unlock access to the private key. This design delivers phishing resistance and strong user verification while simplifying regulatory exposure relative to cloud-processed biometrics (e.g., BIPA obligations are different when no biometric data is captured or stored server-side). From a metrics perspective, the WebAuthn ceremony itself is cryptographically sound; the biometric step inherits the device's local FAR/FRR profile (commonly on the order of 10^{-5}–10^{-6} FAR under typical conditions, with FRR driven by environment and user variability).

1.5 Voice-first realities (and a note on video)

Voice is the predominant interaction in contact centers, and it brings unique constraints. Telephony codecs compress and smooth speech; background noise and handset variety reduce signal quality; and agent tooling often runs in desktop environments where virtual audio devices can be present. These conditions affect both matching performance (raising FRR unless thresholds are tuned for channel quality) and liveness confidence (PAD must be evaluated in-channel, not just on

pristine audio). Where voice is used as a factor—active passphrases or passive voice biometrics—insist on PAD metrics measured over your network conditions, and pair biometric signals with additional evidence when the transaction risk warrants it. In practice, inbound paths traverse codecs like G.711 (narrowband) and AMR-WB/HD Voice (wideband), each imposing spectral and temporal changes that affect model thresholds

Video increasingly appears in escalations (e.g., high-value transactions, remote proofing). Here the capture path matters as much as the model: a browser's real camera under attested conditions and active liveness (randomized prompts, motion parallax) provide more assurance than a generic "show your face" check. As with voice, do not accept liveness at face value unless you also trust the signal's origin (e.g., block or flag virtual cameras for high-risk actions).

In both channels, the cleanest way to raise assurance without abandoning the call is to introduce a cryptographic, device-bound step—for example, sending a transaction-bound approval to the customer's registered app or presenting a WebAuthn challenge. Done well, this evolves the interaction from voice-only to voice+digital without handing the session to a different team or channel.

1.6 Inclusivity and exception paths (without opening a back door)

Not every customer can or will use biometrics or security keys. Accessibility matters, and regulations in some jurisdictions restrict certain biometric uses. The answer isn't to abandon strong factors; it's to engineer exception paths with equivalent assurance. Examples include: offering passkeys unlocked by a device PIN (for users who opt out of biometrics) or enabling assisted workflows that escalate to supervised checks when self-service fails—without dropping to insecure factors for high-risk actions. Publish what your exception paths allow and audit them; in practice, this is where many real-world breaches originate. Because raising thresholds to curb FAR often increases FRR for certain populations or acoustic conditions, Journey advises favoring device-bound cryptography for high assurance to maintain accessibility.

1.7 Summary table: methods, assurance role, and what to measure

Method	Assurance role (onion layer)	FAR/FRR or Residual Attack Surface	PAD metrics (if applicable)	UX burden	Operational complexity	Compliance posture
Passkey (device-bound; unlocked by on-device biometric like Face ID)	Possession + local user verification; phishing-resistant; cryptographic origin	Cryptographically strong. Residual risk dominated by recovery/fallbacks, device theft with weak local unlock	N/A (biometric stays on device; treat local FRR as part of UX)	Low	Low-Med (device registration, recovery design)	Favorable (no server-side biometrics)

Password + Manager	Knowledge (outer layer) with improved entropy/uniqueness	High residual risk: phishing, server compromise of password database, master-password social engineering	N/A	Med (manager UX)	Low-Med	Favorable (no biometrics), but centralized vault risk
Password + Manager + SMS/TOTP	Knowledge + weak possession	Residual risk: SIM-swap/OTP intercept, MFA fatigue, helpdesk resets	N/A	Med	Med (OTP delivery, rate limiting)	Favorable (no biometrics)
Push-based MFA (number matching)	Possession with improved intent confirmation	Residual risk: social engineering to approve out-of-context prompts; device posture	N/A	Low	Low-Med	Favorable
Voice biometric — Active (prompted phrase)	Inherence; needs PAD + trusted capture	Report FAR/FRR at threshold; driven by channel conditions	Report APCER/BPCER in telephony; require randomized prompts	Low-Med	Med (enrollment, tuning, fraud ops)	Regulated in some jurisdictions (biometric privacy laws)
Voice biometric — Passive (free speech)	Inherence; unobtrusive; needs PAD + trusted capture	Report FAR/FRR at threshold; channel-dependent	Report APCER/BPCERin telephony; evaluate continuous PAD	Low	Med-High (continuous scoring, monitoring)	Regulated; continuous capture considerations
Facial biometric — Cloud-verified	Inherence; often used in supervised proofing/escalations	Report FAR/FRR at threshold under WebRTC conditions	Report APCER/BPCER with active liveness; require camera attestation	Med-High	Med-High (SDKs, capture QA)	Heavier compliance (biometric governance)

2. Deepfakes

2.1 Working definition and scope

"Deepfakes" are synthetic audio, video, or combined audiovisual signals generated or manipulated by machine-learning systems to imitate real people's appearance, voice, and behavior. In the authentication context, deepfakes matter for two reasons: (1) they simulate high-assurance cues (voice timbre, facial motion, gaze) well enough to pass as genuine; and (2) they arrive through channels (phone, WebRTC) where the capture path and liveness are hard to police. This primer explains how modern systems create these signals, how they run in real time, and why "detecting fakes" is fundamentally different from verifying origin and liveness.

The case studies introduced in this paper—voice-only impersonation (2019 UK energy firm), voice-led authorization (2020 UAE bank), and multi-participant video

reenactment (2024 Hong Kong)—illustrate how capability has progressed from offline spoofing to live, interactive impersonation.

2.2 A brief history: from GANs to diffusion and neural codecs

The public story of deepfakes often begins with Generative Adversarial Networks (GANs)—two neural nets (a generator and a discriminator) trained in a minimax game so that the generator learns to produce samples the discriminator cannot distinguish from real data. The original formulation by Goodfellow et al. (2014) catalyzed the first wave of compelling synthetic images and faces and later inspired video face-swap systems.

Since then, the field has diversified:

- Diffusion models (and related flow-matching methods) generate images and video by iteratively denoising random noise; they've become state of the art for general image/video synthesis.
- Neural codec language models (NCLMs) for audio—popularized by VALL-E—tokenize speech with a neural codec and learn to "speak" those discrete tokens as a conditional language-modeling task, enabling zero-shot TTS that clones a speaker's timbre from a few-second prompt.
- Voice conversion (VC) systems transform one speaker's voice into another's without rewriting the linguistic content. Modern retrieval-based voice conversion (RVC) is widely available as open-source, trainable on consumer hardware with minutes of audio.
- Face reenactment and lip-sync methods such as First Order Motion Model and Wav2Lip drive a face to match target motion or speech, yielding convincing avatar puppeteering and lip-sync that align tightly with the audio stream.

Pragmatically: today's attackers don't need to train cutting-edge diffusion models from scratch. They compose off-the-shelf components (TTS/VC + reenactment + streaming) or rent them via APIs. That shift—from research novelty to routineized tooling—explains the sharp rise in real-world incidents.

2.3 What "audio deepfakes" really are (and how they're built)

In contact-center scenarios, two families dominate:

1. Text-to-speech cloning (TTS).

- Inputs: A short “enrollment” prompt from the target speaker (seconds to a few minutes), plus the attack script (plain text).
- Model: A neural codec LM (e.g., VALL-E-style) or diffusion-TTS that conditions on the target voice prompt and the text to produce speaker-matched tokens/audio.
- Output: Arbitrary speech in the target’s voice, with controllable prosody and style. Lab systems demonstrate 3-second enrollment prompts for zero-shot cloning.

2. Voice conversion (VC).
    - Inputs: Live or recorded source speech (attacker) + a trained model for the target voice.
    - Model: RVC-class models map content features (phonetic units, F0/intonation) from the source onto the target’s spectral/timbre representation.
    - Output: Near-real-time speech-to-speech transformation that preserves the linguistic content and timing of the attacker while sounding like the target. Open-source RVC tooling emphasizes short training times and small data (order-of-minutes) to get acceptable clones.

Streaming constraints. In live calls, latency is king: perceived conversational quality degrades once one-way delay exceeds roughly 150–200 ms end-to-end. TTS pipelines often run above that; VC pipelines are commonly engineered for interactive latency by caching speaker features and running lightweight vocoders. Practical attackers blend techniques—pre-recorded TTS for long monologues, VC for back-and-forth—to mask delays. (For latency guidance, see ITU-T G.114.)

Codec and channel effects. Telephony codecs (e.g., narrowband/HD Voice) and WebRTC processing alter spectral cues. Modern systems are trained or fine-tuned on those degradations so the synthetic voice survives compression, which also blurs artifacts detectors might otherwise exploit. This is one reason detection scores measured on pristine audio often over-predict performance in the field.

2.4 What “video deepfakes” really are (and how they’re built)

Three capabilities matter most for authentication contexts:

- Face swap. Replacing a face in a target video with a synthesized face. Face-swap toolchains emerged early from GAN variants and remain powerful, especially when camera perspective and lighting are constrained.

- Face reenactment. Driving a target face from a source motion (head pose, expressions). The First Order Motion Model family learns keypoints and affine transformations to retarget motion without 3D face rigs, yielding controllable, real-time "puppeteering."

- Lip-sync / audio-driven animation. Methods like Wav2Lip align mouth shapes (visemes) to a given audio stream, enabling tight audio-video consistency—critical for video calls where moving lips must match the cloned voice.

When combined with virtual cameras, these systems can inject a rendered avatar into a conferencing app as though it were a live webcam feed. Off the shelf, they can also translate or dub speech while preserving the target's voice timbre, making cross-language impersonation feasible with modest preparation.

2.5 From training loop to live call: a minimal production pipeline

A modern deepfake pipeline that targets a conversation typically looks like this:

1. Data acquisition. Public talks, interviews, podcasts, earnings calls, and short social clips yield clean speech and multiple camera angles. For some systems (e.g., VALL-E-style TTS), even 3–10 seconds can bootstrap cloning; more data improves style control and robustness.

2. Model prep.

    - Audio: Train or fine-tune a VC model (minutes to hours) and/or set up a zero-shot TTS.

    - Video: Prepare an avatar with reenactment or lip-sync; calibrate to the conferencing background and lighting.

3. Real-time stack. Run voice capture → conversion → vocoding with a latency budget under ~200 ms; route the output to a virtual audio device. For video, render the avatar at 25–30 fps and publish via a virtual camera.

4. Operator console. A human attacker or scripted agent drives the interaction, mixes pre-baked phrases with live VC, and mirrors behavioral cues (typing sounds, talking while "screen-sharing") to appear authentic.

Publicly reported incidents feature the same components: a voice clone persuading a manager to approve transfers (UAE, 2020), and an entire deepfaked video meeting that looked routine to a finance employee (Hong Kong, 2024).

2.6 Why detection is hard (and keeps getting harder)

1. Cross-model generalization is weak. Detectors trained on a specific synthesis family (say, a GAN or a particular lip-sync model) often fail when tested on different generators, domains, or post-processing pipelines. A large body of work documents performance drops when moving across datasets and generation methods; tuning helps, but zero-shot robustness remains an open problem.
2. Channel effects erase artifacts. In the wild, content is compressed, resampled, and mixed (telephony codecs, noise suppression, AEC). Those operations erase or mask the subtle artifacts detectors look for, while also increasing false alarms on bona fide signals. Contact-center pipelines—AGC, AEC, denoising, and transcoding—both mask synthesis artifacts and raise false positives on bona fide audio; this lab-to-field mismatch is why reported AUCs often overstate deployed protection.
3. Human perception is calibrated for meaning, not fingerprints. People reason about identity, authority, and context, not bit-level cues. If the story is plausible and the cadence feels right, confidence rises quickly—even among trained staff.
4. Adversarial adaptation is rapid. Open-source communities iterate on artifacts detectors publicize (blink rate, teeth glitches, lip-sync offsets). Work like Wav2Lip emerged precisely to close obvious gaps; reenactment methods tackle head-pose and gaze. Each generation narrows the behavioral distance to real signals.

The implication for authentication is straightforward: content matching alone is not a control. Verification must also ask where the signal came from (origin) and whether it's live (liveness), which we operationalize later via trustworthy capture and PAD.

2.7 Presentation vs. injection: two fundamentally different problems

When deepfakes meet authentication, there are two high-level threat surfaces:

- Presentation: A synthetic or replayed signal is presented to the sensor (microphone/camera). Here, Presentation Attack Detection (PAD) applies—challenge-response, motion parallax, or audio-liveness designed to differentiate live, in-scene capture from a screen/speaker replay.

- Injection: The attacker bypasses the sensor entirely by delivering a crafted media stream directly into the application stack (e.g., a virtual camera for WebRTC; a virtual audio device or SIP/RTP manipulation for telephony/softphones). PAD that relies on the physical sensor cannot see the attack if the app trusts a virtual device as its source.

Academic and standards communities have begun to treat injection explicitly, showing that many liveness tests assume presentation and can be bypassed if the signal enters the post-sensor. NIST workshops, USENIX, and other studies discuss how camera/voice injection evades biometric verification; enterprise blogs echo the same trend as these attacks move from theory to practice.

Why it matters here: contact centers often terminate media into desktop software where virtual devices are commonplace (loopback audio, OBS virtual cams). Unless endpoints or browsers attest the capture path (i.e., "this stream originates from a real sensor") and disallow virtual devices for high-risk actions, even the best PAD cannot establish liveness. All liveness claims presuppose a trusted capture path; if origin is untrusted (virtual devices), liveness is unprovable.

2.8 The economics: capability is cheap, accessible, and scalable

Two trends have put deepfake capability into ordinary attackers' hands:

- Data cost collapsed. For audio cloning, datasets shrank from hours to seconds for initial timbre capture (with reduced quality at the extreme). Microsoft's VALL-E demonstrates 3-second zero-shot cloning; open-source RVC workflows routinely train usable voices with ≈10 minutes of clean audio.

- Compute/tooling democratized. Consumer GPUs and hosts can run near-real-time voice conversion; turnkey WebUIs let non-experts train and infer models. Cloud APIs commoditize higher-quality TTS and avatar puppeteering, and commercial SaaS wraps it in accessible UX.

At the same time, attack volume in voice channels is measurable. Pindrop reports that fraud now occurs in 1 of every 599 incoming calls on average (a 26% increase over 2023 and 100% over 2021), and that attempts occur approximately every 46 seconds across contact centers. Independent identity-verification vendors report order-of-magnitude increases in detected deepfakes between 2022 and 2023, with North America showing a 1,740% rise in deepfake use over that period—consistent with what security teams are experiencing.

The high-impact incidents align with this accessibility trend. The UAE $35M voice-clone case in 2020 and the Hong Kong ~$25M deepfake video-meeting case in 2024—both widely reported—show that these attacks aren't just about

authenticating logins; they reshape trust in how decisions are made over phone and video.

2.9 Watermarking and provenance: helpful, but not sufficient on their own

Two families of "trust tech" are often proposed as silver bullets; both help, neither replaces authentication controls:

- Content provenance (C2PA/Content Credentials). C2PA provides a way to sign manifests that record how a piece of content was captured or edited, enabling downstream tamper-evidence and verification of who created it. For assets that carry these credentials end-to-end (e.g., news photos), provenance builds accountability. But live streams and ad-hoc calls rarely carry such manifests; adoption is uneven; and an attacker can simply use non-credentialed tools. Treat C2PA as a positive signal when present, not as a required precondition for trust in authentication.

- Watermarking/labeling. Watermarks (visible/invisible) can help with ecosystem transparency, but robustness remains a challenge—watermarks can be stripped by re-encoding, cropping, paraphrasing (for text), or model-to-model "translation." NIST's recent overview and independent analyses emphasize these limitations; researchers have shown current schemes are easily defeated or yield false signals outside narrow conditions. In authentication, you should not assume a lack of watermark implies "real," nor that a watermark implies "fake."

Bottom line: Provenance and watermarking are promising complements for media ecosystems, not standalone defenses for identity proof.

2.10 Where capability is heading (and why it matters for auth)

The near-term trajectory:

- Lower-data, higher-fidelity voice. NCLMs and diffusion TTS will get better at style control (emotion, speaking rate) and cross-language voice transfer, letting an attacker talk naturally in a victim's own voice across languages.

- More stable real-time reenactment. Face puppeteering will reduce temporal jitter, improve eye-gaze correction, and better track fine mouth dynamics, shrinking the perceptual gap in video calls.

- Tooling integration. Expect one-click stacks that combine number spoofing, voice conversion, avatar streaming, and scripted call flows—all run by non-experts.

- Defender tech will chase, but generalization remains hard. Even as detectors improve, cross-model robustness is the binding constraint; therefore origin and liveness will remain the security fulcrum.

2.11 A note on language, risk, and measurement

In security terms, deepfakes don't just add "another bypass." They shift the identity problem from matching (does this voice/face match a template?) to verifying conditions of capture (did this originate from a trusted device/channel? was it produced live in response to an unpredictable challenge?). That is why the provenance/liveness questions introduced earlier are essential, and why the contact-center channel—voice first, increasingly video-assisted—requires controls that bind origin and liveness before high-risk actions.

3. Synthetic Identity at the Gate: Deepfakes and Authentication

Deepfakes don't merely "fool people." In authentication, they reshape what must be proven. Matching content (a familiar voice, a face that aligns with a template) is no longer enough; we also have to prove where the signal came from and whether it was produced live. The result is a moving target for risk, new bypasses on old factors, and a visible split between controls that rely on human perception and those that rely on cryptographic provenance. This section translates the deepfake primer into the authentication domain—especially for contact centers—so engineering and security teams can see where the goal posts have moved, and which controls still stand.

3.1 A quick reality check on prevalence and capability

Fraud pressure in voice channels is measurable and rising. Independent analyses report that 1 in every 599 incoming calls to contact centers is now fraudulent on average (a 26% YoY increase over 2023 and 100% over 2021), with attempts occurring roughly every 46 seconds across centers—figures that align with what operators observe on the ground. At the same time, deepfake use exploded in verification contexts between 2022 and 2023, with one study citing a 1,740% increase in North America alone. On the capability side, modern systems can clone a voice from seconds of audio (e.g., ~15 seconds in recent public previews), and open-source

stacks make real-time voice conversion and avatar streaming accessible to non-experts.

High-impact incidents show the arc clearly: in the 2019 UK energy case an executive voice clone prompted an urgent transfer; in 2020 UAE a cloned director voice persuaded a bank manager to authorize $35M in wires; and in 2024 Hong Kong, a finance employee attended a video meeting full of deepfaked colleagues and authorized ~$25M in payments.

3.2 Two distinct bypass modes: presentation vs. injection

In the primer we distinguished presentation attacks (a synthetic or replayed signal presented to a real sensor) from injection attacks (a synthetic signal injected after the sensor, directly into the app's media path). That distinction becomes pivotal in contact centers:

- Presentation is what classic PAD (ISO/IEC 30107) is designed to catch: randomized prompts, motion parallax, challenge/response that separates live, in-scene capture from replay. PAD outcomes are measured with APCER/BPCER/ACER and must be tuned to the channel you actually use (telephony/WebRTC).
- Injection bypasses the sensor entirely: virtual audio devices route synthetic speech straight into the softphone; virtual cameras publish an avatar as if it were a webcam feed. PAD that depends on the physical sensor cannot see this. Commodity tooling (e.g., OBS Virtual Camera) is designed for legitimate scenarios but doubles as a media-injection vehicle in fraud.

In desktop agent environments, virtual devices are common (screen casting, QA recording, accessibility). Unless your endpoint or browser attests that media originates from a real sensor—and you block or quarantine virtual devices for medium/high-risk flows—even strong PAD becomes a paper tiger. We'll return to policy later; here, it's enough to say: origin and liveness are separate problems, and injection attacks target origin.

3.3 Method-by-method: how deepfakes change the game

Below we walk the major authentication methods used in or around contact centers, layer them onto the identity "onion," and explain what deepfakes amplify, where injection appears, and what still works. Where we use quantitative claims or public cases, we cite them; otherwise, the recommendations are grounded in current standards and channel realities.

1. Knowledge factors (passwords, KBA)

What changed: Deepfakes supercharged social engineering. A cloned voice in a convincing scenario ("IT here—we detected suspicious activity…") elicits secrets and bypasses with higher success than text or email alone. The core technical weaknesses—reuse, phishability, server leaks—were already known; deepfakes primarily accelerate compromise by making the ruse feel trusted and urgent.

Where injection appears: If the attack's goal is to harvest credentials, the audio/video path doesn't need injection; a normal call suffices. Injection becomes relevant if the attacker wants to feed audio into IVR systems that capture short passphrases.

What still works: Move away from knowledge as a gate for anything meaningful. If it must exist, never let it unlock high-risk actions without a device-bound, cryptographic step-up (discussed later).

2. SMS/TOTP one-time codes

What changed: Deepfakes increased the success of OTP coaching ("read me the code I just sent you") and MFA fatigue(relentless prompts until users accept). The codes themselves aren't altered by deepfakes; the human around them is. SIM-swap and port-out fraud remain core risks whenever the phone number is the recovery backbone.

Where injection appears: Attackers can inject synthetic voice into softphones to drive IVR OTP reset trees or use bots to automate callback loops.

What still works: Treat OTP as possession-lite, not high assurance. Rate-limit retries, bind codes to short windows and transaction context, and never accept OTP as the sole factor for high-risk authorizations. Consider stronger phone checks (next item).

3. OTP with telephone-network validation (ANI-anchored verification)

This pattern pairs one-time codes with carrier-side signals to confirm the calling number and line status (e.g., whether a number is valid and in service, has recent SIM-swap/porting activity, or aligns with customer-provided data). Automatic Number Identification (ANI)—delivered through signaling for toll-free and enterprise call flows—is generally harder to block or mask than simple caller ID, giving the enterprise a more authoritative view of the number behind the call.

What changed: Deepfakes don't alter ANI or carrier checks directly, but they raise the bar on the human step (talking a user into giving the OTP anyway). Adding network-level risk checks (recent SIM swaps, number mismatches) before or

alongside the OTP reduces the attack surface for account recovery and contact-center resets.

Caveats: STIR/SHAKEN attestation, when available, helps with number authenticity but does not validate the person, and it's not universal (e.g., non-IP legs, some international routes). Treat it as a supplemental signal, not a gate. Signal depth and coverage vary by route and region; treat phone-number intelligence as an advisory risk input—not a trust stamp—and, where feasible, constrain to allow-listed ingress (for example, domestic toll-free).

Used this way, OTP plus network-level number intelligence is a pragmatic low-friction first move that reliably deflects the most common synthetic-voice incursions without disrupting legitimate callers.

4. Push-based MFA (including number matching)

What changed: Deepfakes increased the success of guided approvals ("I'm your admin; I'll send a verification—read me the number on the screen"). Number matching raises friction for blind "tap Yes," and federal guidance recommends it to dampen MFA fatigue, but a convincing voice can still walk a victim through the approval process.

Where injection appears: Injection isn't the main bypass here; social engineering on the call is.

What still works: Prefer phishing-resistant flows (passkeys) where possible; if push remains, bind to context(transaction details), require presence (biometric/PIN on device), and detect improbable approval sequences.

5. Passkeys (device-bound WebAuthn), benchmarked with on-device Face ID unlock

What changed: Deepfakes don't break the cryptography. Passkeys are origin-bound and phishing-resistant; the local user-verification step is enforced on the user's device (e.g., Face ID/TouchID/Windows Hello). Apple documents a ~1 in 1,000,000 FAR class for Face ID (device-local), with failure to match triggering a fallback to the device PIN. In practice, deepfakes can only target people and process around passkeys (e.g., bad recovery policies, coerced approvals, device theft with weak local unlock). Even when sync is enabled, passkeys remain RP-scoped and phishing-resistant; the residual risks concentrate in recovery flows and local device unlock, not in the WebAuthn ceremony.

Where injection appears: Not applicable to the auth step itself; the WebAuthn ceremony is cryptographically bound to the relying party's origin.

What still works: Use passkeys wherever feasible in contact-center journeys (e.g., voice→digital step-ups). Keep attention on the recovery surface: don't let an attacker downgrade to OTP via persuasion.

6. Voice biometrics (active)

Active systems prompt the caller to speak a passphrase. In telephony, they live or die by model thresholding, channel quality, and PAD.

What changed: High-quality text-to-speech and voice conversion systems can now mimic target voices with seconds to minutes of data, and many have been tuned to survive telephony compression. Detectors trained on one synthesis family often fail to generalize to others, especially over compressed audio. That means "accuracy" from lab reports over pristine audio will overstate protection on live calls.

Where injection appears: Attackers can inject synthetic speech via virtual audio devices; challenge-response (e.g., randomized phrases) helps, but only if you trust the capture path.

What still works: Enforce trusted-origin capture (no virtual audio devices) for anything beyond low risk. Require active, randomized prompts; measure and publish FAR/FRR at deployed thresholds and PAD metrics (APCER/BPCER) in your telephony conditions. ISO/IEC 30107 gives the vocabulary; use it.

7. Voice biometrics (passive)

Passive (text-independent) systems authenticate from free speech. They're attractive because they're invisible to customers and map well onto agent flows.

What changed: Everything above still applies, but with an extra wrinkle: passive systems are often continuous or opportunistic, which means an injected stream can blend into a normal call without an explicit prompt boundary. Deepfakes improve at prosody and coarticulation, narrowing perceptual gaps.

Where injection appears: The passive case is especially vulnerable to post-sensor injection (virtual audio devices) because there may be no randomized prompt to constrain the synthesizer.

What still works: If you keep passive, gate its acceptance on trusted capture and in-channel PAD tuned to your codec. Consider running passive as a signal to risk scoring, not as the final authorization gate for anything high-value. Deployed this way, continuous passive voiceprints are best treated as post-auth telemetry to spot session drift or takeover signals—not as the final gate for high-value actions.

8. Facial biometrics (video escalations)

What changed: Face reenactment and lip-sync have improved; many detectors overfit to specific artifacts and degrade across generators or after re-encoding. The basic lesson mirrors voice: lab PAD isn't field PAD.

Where injection appears: Virtual cameras are the default injection path in desktop video. If your app accepts a generic webcam source with no attestation, a rendered avatar can pass through your liveness check if the liveness relies on on-screen motion alone.

What still works: Prefer active liveness with unpredictable prompts, verify the camera origin (no virtual cams for high-risk), and record and review APCER/BPCER measured over WebRTC conditions, not studio video.

9. Agent exception paths and overrides

What changed: Deepfakes expand the social bandwidth of impostors during edge cases: "I lost my phone," "I'm traveling," "biometrics don't work on me." The more exceptions you allow without equivalent assurance, the more attractive they become.

Where injection appears: Calling from a spoofed number (discussed next), injecting an avatar into a video escalation, or importing pre-recorded snippets alongside live conversion.

What still works: Treat exception flows as first-class surfaces. If a path exists for accessibility or rare cases, bind it to carrier-network checks (e.g., recent SIM-swap/port-out flags), device-bound cryptography, or supervised proofing—not to a single phone conversation, no matter how convincing. (We'll formalize policy lines later.)

3.4 STIR/SHAKEN: useful signal, not person authentication

STIR/SHAKEN digitally signs the calling number across IP voice networks, providing a way for terminating carriers to verify that the originating provider vouches for the caller's right to use that number. Attestation levels A/B/C express that provider's confidence (A = knows the customer and number; C = gateway unknown). It helps fight number spoofing and improve traceback.

But three caveats matter for authentication:

1. Scope: It only covers IP legs; non-IP segments and some cross-border routes weaken or strip the signal.

2. Semantics: An attested call says “this number is legitimate for this customer”, not “this person is who they claim to be.” You still don’t know who is on the line, nor whether the audio is synthetic.

3. Abuse: Enterprises and CPaaS with legitimate numbers can still carry attacker traffic; A-attestation doesn’t sanctify the content of the call.

Use STIR/SHAKEN as input to call reputation and routing, but do not equate it with identity verification. It’s a network-integrity signal, not an authenticator.

3.5 Why detection alone won’t save you (and what to measure instead)

Detection models (for fake speech or video) are improving, but generalization remains the binding constraint: models trained on one set of generators often underperform on new ones, especially after compression and noise from real channels. Results from ASVspoof and subsequent studies repeatedly show dramatic performance drops out-of-domain. That doesn’t mean “do nothing”; it means to treat detectors as signals in a broader risk engine, not as binary gates for high-risk authorizations.

What should you publish internally? For biometrics, report FAR/FRR and PAD (APCER/BPCER) at your deployed thresholds and channels (telephony/WebRTC). For passkeys, focus on the residual attack surface (recovery, fallbacks, device posture), not on “accuracy.”

3.6 The shifting goal posts for authorization risk

When deepfakes enter the picture, the assurance that used to come “for free” from a human conversation evaporates. Two practical effects follow:

- Threshold drift. Teams raise biometric thresholds (to curb false accepts) and see FRR rise in noisy channels, increasing escalations.

- Policy friction. Exception paths balloon unless they’re engineered with equivalent assurance, not lower bars.

The pragmatic answer—especially for contact centers—is not to abandon calls; it’s to add a cryptographic, device-bound step at decision points. In practice, that means evolving voice-only interactions to voice+digital for the final “approve” or “release funds” step (details later in policy/orchestration).

3.7 Human-in-the-loop signals: helpful heuristics, not gates

Agents and supervisors can learn to notice anomalies that correlate with synthetic media. None is dispositive, and all have exceptions; use them to raise suspicion and trigger a step-up, not to deny service outright.

- Turn-taking/latency: Consistently delayed replies (~200–400 ms more than typical), a refusal to barge-in during overlaps, or unnaturally smooth back-and-forth may indicate a conversion pipeline.
- Prosody & disfluency: Unusual consistency in pitch (F0), limited micro-variations, sterile breath/noise profiles, or oddly placed filler words.
- Channel congruence: A “mobile caller” with studio-clean audio; a “noisy office” with no room impulse cues; lip movements perfectly in sync all the time on video (real human lips drift).
- Context pressure: Excessive urgency and preemption of normal controls (“we don’t have time—just read me the six-digit code”).

Train agents to flag and escalate such patterns. Then give them safe exits: “I’m moving us to a secure approval on your device,” not “I think you’re a deepfake.” (You’ll detail the scripts in policy.)

3.8 Compliance note: server-side biometrics vs. on-device unlock

Regulatory exposure differs meaningfully between server-side biometric processing and on-device user verification used only to unlock a local private key (as with passkeys). Laws like Illinois BIPA impose stringent notice, consent, retention, and private right of action requirements on biometric identifiers in possession of a private entity; that calculus changes if you never collect or store biometric templates centrally. This is another reason many enterprises prefer passkeys with on-device biometrics for authentication, and reserve server-processed biometrics for supervised or exception contexts.

3.9 Takeaways

In the era of deepfakes, matching is table stakes; origin and liveness are the security levers. In practice:

- Treat voice/face as signals that demand trusted capture plus PAD, measured in your channel.

- Move high-risk authorizations to device-bound cryptography (passkeys, in-app signing), keeping the call but shifting the final proof off the human conversation.
- Harden recovery and exception paths, and add carrier-network phone checks when a phone number becomes the pivot for identity.
- Use STIR/SHAKEN and detectors as inputs, not as proof of personhood.

The table below summarizes the core shifts per method and points to what still works.

3.10 Deepfakes × Auth in the contact center: what changes, where to defend

Method	Primary deepfake-amplified threats	Injection vector (if any)	What still works (control pattern)
Passwords / KBA	High-success vishing to elicit secrets; persuasive impostors to reset creds	Not required (presentation suffices)	De-scope from high-risk; never sole gate; route to device-bound step-ups for authorization.
SMS/TOTP	Coaching to read codes; MFA fatigue via relentless prompts; SIM-swap/port-out risk	Optional bot-driven IVR abuse	Bind codes to narrow windows and transaction context; rate-limit; check SIM-swap/port-out risk before allowing OTP-based recovery.
OTP + phone-network validation (ANI-anchored)	Social engineering still targets users; however number/line risk is harder to fake	N/A	Validate number ownership/status with authoritative signals; disallow risky numbers for recovery; treat as stronger OTP, not high-assurance auth.
Push MFA (number matching)	Guided approvals ("read me the number")	N/A	Enforce number matching + local user verification; add context (transaction details); monitor improbable approval sequences; prefer passkeys where possible.
Passkeys (device-bound WebAuthn; on-device biometric unlock)	Human-layer attacks: recovery downgrades, coercion, device theft with weak local unlock	None on auth path	Rely on phishing-resistant cryptography; harden recovery; keep biometric on-device; use for voice→digital step-ups during the call.

Voice biometric (active)	Zero-/few-shot TTS mimics prompts; detectors lose generalization in the wild	Virtual audio devices feed synthetic speech	Require trusted origin (block virtual audio devices) + randomized prompts; publish FAR/FRR and APCER/BPCER at telephony conditions.
Voice biometric (passive)	Continuous cloned speechblended into calls; no prompt boundary	Virtual audio devices	Same as active plus treat as risk signal, not sole authorization gate for high value.
Face (video escalation)	Reenactment + lip-sync reduce visual tells; detectors overfit	Virtual cameras publish avatars	Use active liveness and camera attestation; require no virtual cams for high-risk; measure APCER/BPCER under WebRTC capture.
Agent overrides / exceptions	Deepfaked execs request bypasses; "lost device" narratives	All of the above	Engineer equivalent-assurance paths (e.g., carrier checks + device-bound signing or supervised proofing); audit relentlessly.
Network trust (STIR/SHAKEN)	Misconception that it authenticates a person; it doesn't	N/A	Use as call-integrity signal; don't equate A-attestation with identity; IP-only scope; still require origin+liveproofs for auth.

3.11 Closing: a preview of the policy implications

Two themes thread through every method above:

1. Do not let a phone conversation be the only authorization for high-risk actions. Keep the call, but shift final authorization to a cryptographic, device-bound step or a supervised flow that establishes origin and liveness under your control. (We'll formalize this in the policy chapter.)

2. Accept that detectors are signals, not gates. In voice-first channels, resilience comes from trusted capture, in-channel PAD, and binding authorizations to keys you control—not from guessing whether a sample "sounds real." For example, if an audio detector's score crosses a tuned threshold or you detect a virtual device, route to a passkey step-up; if the step-up fails or risk remains high, escalate to supervised proofing (selfie + ID) before allowing funds movement.

With these principles, contact-center authentication can keep pace with synthetic media: not by trying to out-guess every generator, but by changing what must be proven before money moves or privileges are granted.

4. What to do now: manual policy changes with outsized impact

The following changes can be designed, approved, and rolled out in weeks—not quarters. They deliberately favor policy and process over "buy another tool," and they assume a voice-first contact center with optional video escalation.

4.1 Institute a "no phone-only overrides" rule for high-risk actions

If a request could move money (wires, new beneficiaries, high-value refunds), change entitlements (payroll, limits, privileged access), or liquidate obligations (insurance payouts), the phone call cannot be the sole authorization—no matter who is on the line or how convincing they sound. Keep the conversation, but finalize with device-bound, cryptographic approvals (e.g., passkeys/WebAuthn or in-app transaction signing) or supervised proofing. Make this a written policy, train agents on the phrasing, and audit exceptions. This one step invalidates a large class of deepfake-enabled social engineering.

4.2 Replace static KBA and SMS-only gates in recovery and resets

Retire static KBA for anything beyond low risk. Where OTPs remain, bind them to narrow windows and explicit context (the specific transaction or change), and prohibit agents from asking callers to read a code aloud. For phone-anchored recovery, add network-level checks—valid line status, recent SIM-swap/port-out signals, and number ownership consistency—before trusting a number as a recovery factor. These carrier-side signals materially reduce abuse of OTP flows without naming or tying to any vendor.

4.3 Treat STIR/SHAKEN as call integrity—not identity

When present, STIR/SHAKEN attestation is a useful signal that a number wasn't trivially spoofed across IP networks. But it does not tell you who is speaking, nor whether the content is synthetic. Use it to inform routing and reputation, not to satisfy identity checks. Document this in policy so teams stop treating attested calls as "trusted identities."

4.4 Set acceptance rules for biometrics that assume a hostile channel

If you use voice (active or passive) or face (video escalations) as factors, require two things in policy:

- Trusted capture: block or quarantine virtual audio devices and virtual cameras for any medium/high-risk flow. If the app can't attest origin, do not treat the sample as proof of presence.

- Measured liveness: publish PAD expectations (APCER/BPCER/ACER) and FAR/FRR at deployed thresholds in your actual channels (telephony/WebRTC), not lab conditions.

Make these preconditions explicit. If they aren't met, fall back to device-bound cryptography rather than relaxing thresholds.

4.5 Engineer exception paths with equivalent assurance

Accessibility and edge cases (no smartphone, biometric opt-outs, device loss) are realities. The mistake is allowing exceptions that lower assurance. For example, replace "I lost my phone—let me reset with KBA" with "we'll continue the call, and you'll complete an in-branch or video-supervised check, or we'll send a one-time, origin-bound link to complete passkey registration on a new device." Publish the menu of acceptable exceptions and enforce dual-controlwhere needed (e.g., finance and ops both approve high-risk exceptions).

4.6 Harden the agent environment against media injection

Most injection attacks exploit desktop realities: virtual devices, screen-share loopholes, and softphone settings. Policy should mandate:

- Application-level blocks on virtual audio/cam sources for medium/high-risk workflows.
- Per-session recording and screen capture rules that protect sensitive codes and keys from appearing on agent desktops.
- VDI or kiosk profiles for agents handling high-risk queues, limiting local installs (including "virtual cable" drivers).
- Safe-exit scripts: when risk spikes, agents shift the session to a secure step-up ("I'm sending a secure approval to your registered device now") rather than debating authenticity.

4.7 Train for conversational anomalies—use them to trigger step-ups

Teach agents to recognize turn-taking delays, unnaturally consistent prosody, and over-rehearsed urgency as risk triggers. These are not denial conditions; they justify moving the authorization milestone to stronger evidence. Pair training with clean UI cues in the agent desktop so the "next step" is obvious and non-confrontational.

4.8 Publish standardized KPIs and hold quarterly policy reviews

Policy only improves if it's measured. Track: Step-Up Rate (and abandonment impact), False Escalation Rate, Recovery-Flow Fraud Rate, Loss Avoided per 1,000 Calls, Deepfake Detection Precision/Recall (if you run detectors), and Mean Time to

Contain suspicious sessions. Review quarterly with fraud, security, ops, and compliance. Tie objectives to loss-per-interaction and CSAT where appropriate.

4.9 Align biometrics governance to jurisdictional obligations

If you process biometrics server-side (voiceprints, face templates), adopt notice, consent, retention, and deletion practices consistent with jurisdictions like Illinois (BIPA). Where possible, prefer on-device user verification (e.g., passkeys unlocked by Face ID/Touch ID/Windows Hello) to reduce regulatory footprint while raising assurance. Document the difference in your policy and procurement criteria.

4.10 Red-team the channel you actually run

Run telephony/WebRTC red-teaming with compressed audio, background noise, and desktop agents. Test presentation and injection paths, not just replay. Use the findings to tune thresholds, PAD, and agent scripts. Put a date on the calendar now; repeating it transforms policy from a document into a living control.

4.11 What "good" looks like: a simple policy-maturity model

Level	Name	What it looks like	Risk posture
1	**Static**	KBA, passwords, SMS OTP; ad-hoc exceptions; human trust in voices/video; minimal measurement.	High loss volatility; exception abuse.
2	**Contextual**	Basic risk scoring (velocity, device reputation); STIR/SHAKEN used for routing only; phone-number risk checks (line status, SIM-swap/port-out) added to recovery; initial padlocks on virtual devices.	Reduced abuse of phone-anchored resets; social engineering still lands.
3	**Dynamic**	No phone-only overrides for high-risk actions; routine voice→digitalstep-ups with passkeys; biometric use gated by trusted capture and PAD in telephony/WebRTC; agents trained on anomaly cues.	Fraud materially drops; escalations increase but are predictable and measured.
4	**AI-Governed**	Continuous risk evaluation across the session; detectors as signals (not gates); model-informed balancing of friction budgets and loss budgets; red-team/feedback loops; policy rolls forward automatically with governance oversight.	Loss per interaction stabilizes; friction targets enforced without manual firefighting.

Define simple exit criteria per level—for example, at Level 3: '≥95% of high-risk actions are finalized using device-bound cryptography with human identity out-of-band only for exceptions.

4.12 Planning for the near future: AI-governed policy without the hype

AI-governed does not mean "let a model decide everything." It means the policy engine continuously updates its view of trust using many small signals (voice/face liveness outcomes, device posture, phone-number risk, behavioral consistency, transaction context), and adjusts the assurance required to proceed. Practically:

- Continuous evaluation, not one-and-done: Risk is recalculated as the interaction unfolds. A conversation that starts low risk can become high risk when the customer asks to add a new international beneficiary. The system doesn't start over; it steps up—seamlessly, during the call.
- Signals from different planes: Combine cryptographic provenance (passkeys; in-app signing), channel integrity (STIR/SHAKEN as a call-integrity input), carrier data (line status, SIM-swap/port-out recency), biometric evidence (only with trusted capture + PAD), and conversation/behavior (latency and prosody anomalies) into a single risk view.
- Detectors as inputs, not vetoes: Deepfake detectors—audio or video—inform the risk score but rarely make the final call. They are valuable when fused with provenance and context, and dangerous when used as binary gates.
- Explainable decisions and auditability: Every step-up or deny should be explainable after the fact: "Risk increased because (a) recent SIM swap; (b) virtual camera detected; (c) transaction value exceeded threshold; therefore we required passkey authorization and supervisor co-approval." That's what regulators, auditors, and boards will ask for.
- Human-centered fail-safes: Agents need clean transitions ("I'm sending a secure approval to your device now"), no-drama exits, and scripts that avoid accusing customers while still protecting the firm.

This is less about a novel architecture and more about operationalizing the three questions we introduced earlier—content, origin, liveness—as policy levers. The future arrives when the system can continuously balance these levers against what the user is asking to do and the risk tolerance of the business.

4.13 Example Scenarios

Financial Services (wire transfer):
A caller sounding like a long-time CFO asks to push a same-day wire to a new

overseas beneficiary. The agent's desktop flags low call-content risk but medium origin risk (no STIR/SHAKEN on the route), with recent SIM-swap noted on the callback number. Policy triggers an in-call step-up: the agent keeps the caller on the line while the authorized signatory receives an in-app, transaction-bound approval protected by a passkey unlocked via on-device biometric. The voice never authorizes the transaction; the device and cryptography do. If the approval fails, the agent moves to a supervised video escalation with camera-origin attestation—no virtual cameras allowed.

Insurance (claim payout):
A claimant requests a claim advance after a catastrophic loss. Policy routes to a video verification because the payout exceeds a threshold. The verification app refuses virtual cameras, runs active liveness with motion parallax, and captures device posture. Despite passing liveness, the system notes behavioral divergence (unusual response latency) and nudges the flow to a device-bound cryptographic approval tied to the claim ID. If the claimant cannot complete it, policy offers a branch or notarized path—an exception with equivalent assurance, not a lower bar.

5.0 From Voice-Only to Voice+Digital: Journey's Approach to Trust

Journey's approach is simple: make it easy for enterprises to raise assurance during the conversation, not after it—so that policy can always stay one step ahead of deepfake-driven risk. In practice, that means (1) offering fast, low-frictionways to verify identity at multiple points along the assurance spectrum, (2) advising leaders on policy so authorization milestones move earlier or later based on risk, and (3) exposing our verification methods through Model Context Protocol (MCP) so your AI risk and workflow engines can orchestrate the right step-up at the right moment across CCaaS and UCaaS platforms.

5.1 Spectrum coverage: the right step-up, complete in seconds

Our philosophy is that coverage along the spectrum beats any single "strongest" method. When your risk engine or an agent needs to step up assurance, the method must be immediately available, comprehensible to the customer, and complete in seconds.

- OTP with phone-number validation (ANI-anchored checks).
  When you need a quick improvement over KBA, one-time passcodes remain valuable—if they are paired with authoritative phone-number signals. Behind the scenes, we incorporate network-level checks (e.g., number validity and recent port-out/SIM-swap indicators) so OTP isn't blindly trusted just because

a code is read back. This is an easy uplift that often deploys in days, and it cuts the “45–90 second KBA ritual” down to a single prompt that customers recognize.

- Passkeys (WebAuthn) as the default step-up.
  For most contact-center use cases, passkeys offer the cleanest blend of phishing-resistant cryptography and low friction. A risk engine or agent can send a transaction-bound approval that the customer completes with Face ID/Touch ID/Windows Hello on a registered device. It typically clears in seconds and proves possession-and-presence without exposing the enterprise to server-side biometric obligations. In our experience, this becomes the workhorse step-up for authorizing money movement, changing entitlements, or releasing sensitive data.

- Cloud-verified face + photo ID match for high-risk events.
  When the stakes are highest—e.g., new international beneficiaries, large claim payouts, high-value refunds—policy can call for a video selfie with active liveness and a photo ID scan (license or passport). This combines human-familiar proofing with machine verification. It’s more effort than a passkey, but that’s the point: you reserve it for moments where assurance must be indisputable.

Where does voice biometrics fit? We are pragmatic: in many contact-center environments, cost, operational complexity, FAR/FRR tuning, and biometric-privacy obligations make voice a less attractive primary factor. If a client already has voice in their stack, we help set the right guardrails and position it as a signal rather than the final authorization gate for high-risk actions. The spectrum remains covered without depending on a factor that synthetic speech is explicitly designed to imitate.

5.2 Policy partnership: moving authorization milestones—on purpose

The strongest technology fails under weak policy. Journey works with clients to clarify policy first: which actions require which level of proof, where exceptions live, and how to move milestones along the spectrum when risk rises. The core moves we recommend:

- Make “no phone-only overrides” explicit for high-risk actions.
  The conversation can initiate and coordinate, but final authorization for wires, beneficiary changes, high-value refunds, claim payouts, and privileged support actions should shift to device-bound cryptography (passkeys) or a supervised verification (e.g., video selfie + ID). This single change cuts off the

most lucrative deepfake pathways while keeping the agent-led experience intact.

- Redesign recovery and exception paths with equivalent assurance.
  Policy should never let a user drop from “strong” to “weak” just because they’ve lost a device or opted out of biometrics. We help clients replace static KBA with OTP+number validation and make it easy to bootstrap a new passkey—all in-call, with agent guidance if needed. For customers who can’t complete digital steps, policy escalates to supervised options rather than lower bars.

- Measure and adjust.
  Leaders publish the same KPI pack we recommend elsewhere—Step-Up Rate, False Escalation Rate, Recovery-Flow Fraud Rate, Loss Avoided per 1,000 calls, and end-to-end time-to-clear—and review them quarterly with security, fraud, and operations. Our role is to ensure each policy decision has a fast, customer-proven path on the spectrum and a clean handoff for agents.

We’re advisors here—not arbiters. You set policy and risk appetite. Our job is to make compliance easy: when your policy says “step up now,” the customer completes the step in seconds, and the agent stays in control of the conversation.

5.3 MCP: connecting trust controls to your AI decisioning

Model Context Protocol (MCP) is the middleware layer that lets your AI risk/workflow engines call Journey’s verification steps on demand in your CCaaS/UCaaS environment. Think of MCP as a small set of standards-based endpoints that expose composable primitives—send OTP with number checks, initiate passkey challenge, launch video selfie + ID—and return signed, auditable outcomes your systems can trust.

- Orchestration without lock-in.
  MCP does not replace your policy engine; it answers it. Your agentic AI evaluates context (transaction value, session history, device posture, phone-number risk, conversational signals) and asks MCP for the next proof. MCP executes the step-up, then hands back a verifiable result and minimal metadata (e.g., which factor, timestamp, any risk flags) for your records.

- Built for the contact center.
  MCP integrates with CCaaS/UCaaS platforms, presenting customers with the right in-call or co-browsedexperience: a one-time link for OTP+number validation, a passkey approval the customer completes on their device, or a

video-selfie + ID flow for high-assurance proofing. Agents don't leave the desktop; customers don't get bounced across channels.

- Audit and governance from day one.
  Each outcome is signed and timestamped with minimal metadata for audit, enabling downstream systems and regulators to verify exactly 'what was proven' and when. Because outcomes are signed and standardized, compliance teams gain a clear audit trail: what was requested, what was proven, and by which method. That makes it straightforward to show regulators and auditors that policy wasn't just well-intentioned—it was enforced.

The result is a clean separation of concerns. Your AI decides the what and when; MCP provides the how and returns proof. That's how organizations scale from static rules to AI-governed trust without rewriting their contact-center stack.

5.4 What this looks like in practice

- Financial services: A caller requests a same-day transfer to a new beneficiary. The agent stays on the line while your risk engine triggers a passkey-based, transaction-bound approval through MCP. The customer approves with Face ID in seconds; if they can't, policy escalates to video selfie + ID. Either way, the final "yes" is cryptographic or supervised, not just a convincing voice.

- Retail (card-not-present refund): A customer asks to credit a new card. Policy forbids reading codes on calls; MCP fires an OTP+number-validation step that completes faster than KBA and returns a signed outcome. For higher values, the flow upgrades to passkey.

- Insurance/healthcare: A claimant requests an advance on a payout or access to sensitive records. MCP launches a video selfie + ID with active challenges. If the context is lower risk (e.g., appointment or address updates), your engine selects passkey instead. Same interface, appropriate assurance.

Across all three, the agent keeps the relationship; the proof shifts to a stronger plane at the precise moment policy demands it.

Conclusion

Deepfakes have forced a simple but transformative realization: identity in remote channels can no longer rest on what sounds familiar or looks right. Voices and faces—once treated as high-assurance cues—are now synthesizable on demand, and contact centers, where decisions are made under time pressure and at scale, have become the proving ground for this shift. The practical consequence is that

authentication has to evolve from matching content to proving origin and liveness, and authorization has to move from voice-only trust to voice+digital certainty.

This paper opened by reframing authentication as an “identity onion,” with immutable self at the core and progressively more convenient representations radiating outward. That model remains useful, but deepfakes collapse our old intuitions about where the strong layers live. A convincing, real-time clone can sit inside a voice conversation and still be entirely untrustworthy. The way forward is to require evidence that synthetic media cannot provide: cryptographic bindings to known devices and unpredictable, verified liveness in channels we control.

For contact centers, the most important operational pivot is also the most pragmatic: keep the call, but shift the final proof off the conversation. When money moves, entitlements change, or sensitive data is released, the last step should be a device-bound, phishing-resistant ceremony or a supervised verification—not a persuasive voice. That is what “voice-only to voice+digital” means in practice. It does not break the agent experience; it anchors it to something the attacker cannot spoof.

At the same time, leaders should expect the goal posts to keep moving. Synthetic speech and video will continue to gain fidelity and reduce latency; model-specific detectors will keep improving but will remain brittle out of domain; and adversaries will seek whatever exception path remains open. The durable defenses are the ones we can measure and govern: published FAR/FRR and PAD outcomes in the actual channels we run; clear thresholds for when step-ups happen; explicit rules that phone-only overrides end at defined risk levels; and recovery processes that preserve assurance rather than silently downgrading it.

Policy is the mechanism that turns these principles into practice. In the near term, that means manual changes any team can roll out: move authorization milestones earlier along the spectrum where loss concentrates; pair OTP with authoritative phone-number checks for recovery; gate biometrics behind trustworthy capture and liveness; train agents to treat conversational anomalies as triggers for step-up rather than points of confrontation; and harden desktops against media injection. These are not theoretical controls—they are the difference between a believable story and a completed fraud.

Over the medium term, policy becomes continuous and model-informed. The same way fraud analytics learned to weigh many small signals into a single decision, identity assurance will be recalculated throughout a session. Signals from cryptographic provenance (passkeys and transaction signing), channel integrity (caller-ID authentication as a network signal, not personhood), carrier data (line

status and SIM-swap/port-out recency), biometric liveness (measured under telephony/WebRTC constraints), and conversational behavior will be fused to decide whether to proceed, step up, contain, or deny—without forcing agents or customers to change channels. This is what we mean by AI-governed trust: not black-box vetoes, but continuously balanced evidence that can be explained and audited after the fact.

Journey's role in this landscape is deliberately practical. The company focuses on making high-assurance step-ups available in seconds—from quick lifts like OTP backed by phone-number validation, to passkeys that bring phishing-resistant cryptography and on-device biometrics to the call, to cloud-verified face-and-ID checks reserved for the highest-risk events. Journey also works with leaders to align policy—clarifying which actions require which proof and how to move those proof points along the spectrum when risk rises—then ensures the corresponding step-ups are easy for agents to invoke and customers to complete. Behind the scenes, Journey exposes these capabilities in ways that connect cleanly to AI-driven risk and workflow engines, so orchestration becomes a policy choice rather than an integration project.

There is a regulatory dividend in this approach. Anchoring the most common step-ups in passkeys unlocked by on-device biometrics keeps sensitive templates on the customer's hardware, reducing exposure under biometric privacy laws while delivering strong user verification. Where server-side biometrics are appropriate (e.g., supervised escalations), publishing liveness metrics and capture-origin controls helps satisfy emerging expectations for explainability and testing. Boards and regulators are asking the same question security teams are asking: what was proven, by which method, at what point in the journey, and why was that good enough for this risk? The controls outlined here make that answer specific, repeatable, and auditable.

The incidents and statistics cited throughout this paper make clear that synthetic identity is a present-tense problem, not a future concern. But they also show that the path forward is not mysterious. Recenter the system on proofs that deepfakes can't fake. Move the decisive moment from voice-only belief to voice+digital proof. Treat detectors as signals, not gates. Design exception paths with equivalent assurance, not lower bars. And measure everything, because measured controls get tuned—and tuned controls hold. Organizations that move the decisive moment from voice-only belief to voice+digital proof will see fraud decline and escalations become measurable and predictable—two properties boards and regulators actually reward

If the last decade was about building better matches, the next one will be about establishing trustworthy conditions of capture and authorization. Organizations that make these changes now will still have persuasive conversations with their customers—but the approvals will come from places only the real customer can reach.

Sources:

- Generative Adversarial Networks (original paper): https://arxiv.org/abs/1406.2661
- Neural Codec Language Models (VALL-E): https://arxiv.org/abs/2301.02111
- VALL-E 2 (human-parity zero-shot TTS): https://arxiv.org/abs/2406.05370
- Retrieval-based Voice Conversion (RVC) — project repo: https://github.com/RVC-Project/Retrieval-based-Voice-Conversion-WebUI
- First Order Motion Model for Image Animation (FOMM): https://arxiv.org/abs/2003.00196
- Wav2Lip (lip-sync to arbitrary speech): https://arxiv.org/abs/2008.10010
- ITU-T G.114 (voice one-way delay recommendations): https://www.itu.int/rec/T-REC-G.114
- ITU-T G.711 (narrowband telephony PCM): https://www.itu.int/rec/T-REC-G.711/
- 3GPP TS 26.190 (AMR-WB speech codec): https://www.3gpp.org/dynareport/26190.htm
- ETSI copy of 3GPP TS 26.190 (PDF): https://www.etsi.org/deliver/etsi_ts/126100_126199/126190/06.01.01_60/ts_126190v060101p.pdf
- ASVspoof challenge (home): https://www.asvspoof.org/
- ASVspoof 2021 evaluation plan (tracks LA/PA/DF): https://www.asvspoof.org/asvspoof2021/asvspoof2021_evaluation_plan.pdf
- ASVspoof 2021 summary paper: https://arxiv.org/pdf/2210.02437
- C2PA specification (Content Credentials, v2.2): https://c2pa.org/specifications/specifications/2.2/specs/C2PA_Specification.html
- C2PA explainer (durable content credentials & soft binding): https://c2pa.org/specifications/specifications/2.2/explainer/Explainer.html
- C2PA soft binding (manifest recovery): https://c2pa.org/specifications/specifications/2.2/softbinding/Decoupled.html
- Google DeepMind SynthID (overview): https://deepmind.google/science/synthid/

- Meta AudioSeal (audio watermarking): https://ai.meta.com/research/publications/proactive-detection-of-voice-cloning-with-localized-watermarking/
- W3C WebAuthn Level 3 (spec): https://www.w3.org/TR/webauthn-3/
- FIDO Alliance — Passkeys overview: https://fidoalliance.org/passkeys/
- Apple Platform Security (includes Face ID device-bound security): https://help.apple.com/pdf/security/en_US/apple-platform-security-guide.pdf
- Apple Face ID Security Guide (standalone): https://www.apple.com/business-docs/FaceID_Security_Guide.pdf
- ISO/IEC 30107-3:2023 (PAD test methodology): https://www.iso.org/standard/79520.html
- iBeta (PAD testing program page): https://www.ibeta.com/biometric-testing/
- FCC STIR/SHAKEN call authentication overview: https://www.fcc.gov/call-authentication
- ATIS / STI-GA: “Improper Attestation” (A/B/C definitions in practice): https://sti-ga.atis.org/wp-content/uploads/2023/05/Improper-Attestation-Final.pdf
- SIP Forum (STIR/SHAKEN overview & attestation concept): https://www.sipforum.org/
- 47 CFR §64.1600 — ANI definition (authoritative): https://www.law.cornell.edu/cfr/text/47/64.1600
- CISA fact sheet — Implement number matching in MFA apps: https://www.cisa.gov/sites/default/files/publications/fact-sheet-implement-number-matching-in-mfa-applications-508c.pdf
- Microsoft Learn — Number matching in Authenticator: https://learn.microsoft.com/en-us/entra/identity/authentication/how-to-mfa-number-match
- W3C WebAuthn Level 2 (for historical compatibility): https://www.w3.org/TR/webauthn-2/
- OBS Studio “Virtual Camera” (typical desktop injection path): https://obsproject.com/kb/virtual-camera-guide
- 2019 UK energy firm voice-clone heist (~£220k): https://www.theguardian.com/technology/2019/sep/03/fraudsters-use-ai-to-mimic-bosss-voice-in-220000-scam
- 2020 UAE $35M voice-clone heist (Forbes): https://www.forbes.com/sites/thomasbrewster/2021/10/14/huge-bank-fraud-uses-deep-fake-voice-tech-to-steal-millions/
- 2024 Hong Kong multimillion video-conference deepfake (news explainer): https://www.cfo.com/news/deepfake-cfo-hong-kong-25-million-fraud-cyber-crime/

- Sumsub research—deepfake incidents surge 10× (2022→2023): https://sumsub.com/newsroom/sumsub-research-global-deepfake-incidents-surge-tenfold-from-2022-to-2023/
- Pindrop Voice Intelligence & Security Report (landing page): https://www.pindrop.com/research/report/voice-intelligence-security-report/
- Resemble AI (Q1 2025 fraud trends incl. deepfake fraud): https://www.resemble.ai/blog/voice-fraud-q1-2025-report