AI / ML ENGINEER UTICA, MICHIGAN

Joseph DeMasse

I build what people assume needs a whole team.

Alone, and in production. AI / ML engineer in Michigan — real-time voice agents on live customer calls, retrieval systems that answer regulatory questions with citations, and the Salesforce and AWS platform underneath all of it.

18
systems designed and shipped — voice AI, retrieval, underwriting, platform, customer-facing products, internal tooling

24/7
an autonomous voice agent on live borrower calls, unattended, on infrastructure I built and run

3
Salesforce production orgs I own end to end, sandbox through QA to release

203,606
calls transcribed into a private training corpus without a second of audio leaving our own hardware

How I got here.

I came up inside the mortgage industry, where I watched smart people lose hours every day to systems that didn't talk to each other, to manual work that should have been automated, and to *that's just how it's done*. So I learned to build. Not slideware — real systems that move real revenue and run every day.

Since then I've shipped **real-time AI voice agents** that make outbound calls on their own: low-latency SIP pipelines, intent routing, and a compliance engine that clears do-not-call, TCPA, calling-hour windows and state licensing before a single call connects. And **retrieval systems** that read dense regulatory documents and answer with citations — vector search, OCR ingestion, and a won't-answer-unless-it's-grounded firewall so the model never makes things up. One of them assists on loan underwriting by citing the actual guideline thresholds.

Underneath sits **production cloud and CRM architecture**: Salesforce across live production, dev and QA orgs, wired into AWS with JWT auth, webhooks and multi-tenant routing, plus

integrations into enterprise lending APIs, dialers and internal tooling that hundreds of people use daily. On the front end, 3D web interfaces and React apps packaged from one codebase for web, desktop and mobile.

The throughline: I take a messy, high-stakes business problem and turn it into a shipped, compliant, reliable system — usually faster than anyone expects, and usually solo. I think like an engineer, but I learned the business first, so I build things people actually use instead of things that demo well.

Experience.

Roles and what I owned in them, newest first.

Chief Executive Officer / President

October 2025 - Present

AquaBots Inc. · Shelby, Michigan

Water-technology company building AI-powered, solar-driven filtration robots that detect, purify and preserve water sources in industrial and humanitarian settings.

- Autonomous water filtration and purification systems.
- IoT and AI-driven water-quality analytics, with onboard sensors adapting in real time.
- Solar-powered, energy-efficient design for municipal, industrial and humanitarian use.

Vice Chairman of the Board of Directors

October 2025 - Present

AquaAid Foundation · Michigan

Nonprofit humanitarian NGO delivering clean, safe and sustainable water to communities in need — pairing filtration technology with community-led programs.

- Deployment of AI-enhanced filtration and solar-powered purification in the field.
- Disaster relief and emergency water aid.
- Partnerships with engineers, environmental scientists, nonprofits and local leaders.

AI / ML Engineer

June 2025 - Present

Ur Mortgage · Troy, Michigan

I design and ship production AI/ML systems that move real revenue in mortgage lending, with compliance built in (TCPA, DNC, FCRA, ECOA / Reg-B) — because in this industry “smart” isn’t enough, it has to be defensible.

- Built an internal AI “brain” platform: retrieval-augmented generation over a vector store spanning mortgage, sales and engineering domains, with a compliance firewall that abstains rather than hallucinates and self-train/test loops that measurably lift accuracy.
- Built an AI underwriting assistant that maps real loan data to a structured file, cites specific guideline thresholds, and produces findings memos as decision support — abstaining when a claim can’t be cited.
- Built a real-time AI voice dialer on Twilio and OpenAI Realtime, re-architected to a pre-generated playbook plus an embedding-based intent router, with a Whisper call-study pipeline that learns technique from real recordings.
- Enforced compliance gating end to end: do-not-call, calling-hour windows, state licensing and consent, all cleared before a call connects.
- Built ML and data pipelines: local OCR ingestion of scanned guidelines with no data egress, lead-scoring models, and an autoencoder plus K-Means anomaly-detection security layer.
- Worked the full stack — vector search and LLM orchestration on the AI side; Salesforce (Apex, LWC, multi-org lender integrations) and AWS serverless (Lambda, API Gateway, DynamoDB, Amplify) on the platform side; React, Next.js and Three.js on the front end.
- Engineered for reliability: idempotency, race-condition handling, deferred-retry queues and replay tooling.

Full Stack Developer

June 2025 - Present

Ur Mortgage · Troy, Michigan

Platform and integration work alongside the AI engineering role.

- Salesforce setup and administration.
- Experian cloud and CRM integration.
- Web application development.

Mortgage Loan Officer

June 2025 - Present

Ur Mortgage · Troy, Michigan

Licensed originator alongside the engineering work — the reason the systems I build match how the business actually runs.

Salesforce Administrator

June 2025 - Present

NEXA Mortgage · Troy, Michigan

Manage, develop and own all Salesforce work.

Loan Officer

June 2025 - Present

NEXA Mortgage · Troy, Michigan

Originating residential mortgages alongside the platform and engineering work.

Mortgage Loan Officer

January 2025 - June 2025

Lend-A-Loan LLC · Michigan

Provided individuals and families with affordable, secure mortgage options — refinance, equity, purchase and DSCR.

Artificial Intelligence Engineer

May 2023 - October 2023

RadicalX · Remote

Built onboarding and personal-manager bots for the RadicalX platform using Botpress.

Marketing Associate

July 2023 - November 2023

Innovative Client Connections, Inc. · Southfield, Michigan

Marketing for clean-energy companies inside retailers.

Web Developer

May 2023 - June 2023

Mother Earth Natural Health · United States

Built the site and managed the backend using WordPress and Elementor.

Full Stack Engineer

November 2018 - October 2023

Freelance · Michigan

Five years of independent full-stack work across web and application projects.

Licensed Health and Accident Producer

February 2021 - June 2023

American Health Plans · United States

Helped individuals and families find major medical, supplemental and dental/vision coverage on the marketplace.

Life Insurance Agent

July 2021 - April 2022

World Financial Group (WFG) · Mount Clemens, Michigan

Indexed universal life insurance alongside financial planning and advising.

Education.

MIT xPRO

start - May 2026

Certificate, Designing AI Products & Services — Artificial Intelligence / Machine Learning

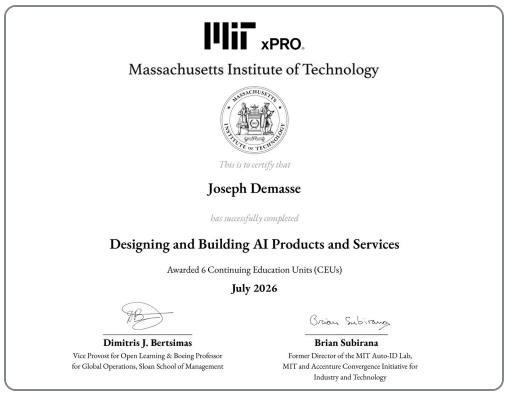
---

Michigan State University

May 2023 - November 2023

Bachelor’s degree, Computer Science

Certifications.



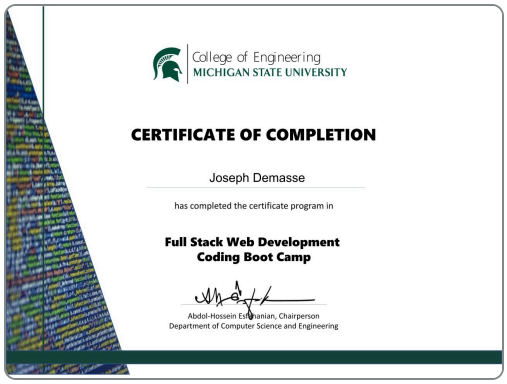
MIT XPRO — MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Designing and Building AI Products and Service

17 July 2026

Awarded 6 Continuing Education Units. Signed by Dimitris J. Bertsima Learning and Boeing Professor for Global Operations at the MIT Sloan and Brian Subirana, former Director of the MIT Auto-ID Lab. Delivered independently verifiable via Accredible. Capstone: an autonomous out designed through the double-diamond framework with defined intellig extraction accuracy, ≤300 ms barge-in latency, ≥0.90 intent-classifica audience, and a governed model-update path with offline validation ar

Verify this credential → (https://certificates.emeritus.org/eb3c79de-c



MICHIGAN STATE UNIVERSITY — COLLEGE OF ENGINEERING

Full Stack Web Development Coding Boot Camp

date

Certificate of completion issued by the Department of Computer Science and Engineering, signed by chairperson Abdol-Hossein Esfahanian.

NMLS

Residential Mortgage Loan Originator License

date

Licensed originator — the reason the compliance layer in everything above is built from the regulation outward rather than bolted on afterwards.

FREECODECAMP

Front End Development Libraries

date

what it covered

FREECODECAMP

Responsive Web Design

date

what it covered

Awards & honours.

MIT XPRO — MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Exemplary Assignment — Final Capstone Projec

22 July 2026

Recognised for exceptional performance on Workbook 9.1, the program designing an AI product using the double-diamond framework and sc maximises short-term success while avoiding the prototype trap. The autonomous outbound voice agent now running in production. Assess management, entrepreneurial mindset, AI design thinking, and produc deployment.

Verify this credential → (https://certificates.emeritus.org/29a5ce03-f

MIT XPRO — MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Exemplary Assignment — Sustainability, Feasibi Responsibility of GANs

16 July 2026

Awarded for exemplary performance on the required GANs workbook: adversarial networks to a real societal challenge, then holding the des feasibility and responsible-AI standards rather than capability alone. A decision-making, creative problem-solving, feasibility analysis and sus

Verify this credential → (https://certificates.emeritus.org/3c28fbff-e

Selected work.

Eighteen systems, grouped by what they do. Built at Ur Mortgage unless noted otherwise; most are in production right now.

AI & MACHINE LEARNING

Avery

Autonomous outbound voice agent

2026 · Node · Telnyx · OpenAI Realtime · ElevenLabs / Cartesia · Salesforce · AWS Lightsail · in production

Speed-to-lead is the largest controllable driver of funded volume in this business, and it is decided in the first ninety seconds. Avery owns those ninety seconds at unlimited concurrency: it calls the moment a lead arrives, greets the borrower by name, qualifies intent, captures the opening Form 1003 through ordinary conversation, and warm-transfers only interested borrowers to a licensed officer with the application already filled in. It never quotes a rate and never makes a decision. That boundary is enforced in code, not written in a prompt and hoped for.

The half that matters is underneath. A fail-closed gate clears do-not-call status, calling window, state licensing and recording disclosure before a digit is dialed. Carrier webhooks are verified as Ed25519 signatures over the raw body and rejected outright when they don't check out — which meant learning that a raw 32-byte key needs an SPKI DER prefix prepended before the runtime will accept it, and that the vendor's own documentation names the wrong field on the key endpoint.

The call log is the piece I'm most careful about, because it is the artifact that answers a subpoena. Version one wrote its state file in place and swallowed its own errors, so a crash mid-write could destroy the entire history and quietly reseed an empty one. Version two writes atomically and treats an append-only, SHA-256 hash-chained journal as the source of truth: any edit or deletion breaks the chain, and verification names the exact sequence number where it broke. Searchable by phone number, IP and transcript.

It runs unattended on its own cloud host behind automatic TLS, works a live pool of 15,784 leads, and both speech engines draw from that one pool so neither starves.

Crucible

Adversarial, transparent AI underwriting

2026 · Python · FastAPI · vLLM / ROCm · BM25 retrieval · Docker · public repository

Underwriting today is a black box that says no. Crucible makes it a glass box that shows the path to yes. Three agents argue over one live loan file: an **Adversary** that tries to kill it and cites every finding, an **Advocate** that finds the guideline-valid route around each kill shot, and an **Adjudicator** that rules on the cited record — abstaining where the evidence is missing — then emits a ranked remediation path alongside the Regulation B adverse-action reasons the borrower is owed either way.

Citations are retrieval-grounded against a Fannie Mae and HUD corpus rather than generated, so a claim that can't be traced doesn't ship. A hybrid router keeps cheap and sensitive work on a local GPU and escalates only genuinely hard reasoning to a frontier model, which turns out to be both the cost story and the privacy story. Uploaded documents are classified and auto-clear the conditions they satisfy.

Built for the AMD Developer Hackathon Act II, Unicorn track, and submitted July 2026.
Source → (https://github.com/UR-MORTGAGE/crucible)

Astravyx

Local-first agentic retrieval

2025-2026 · Python · Haystack · Ollama · Chroma · BM25 · private

A knowledge engine that runs on hardware we own, because the documents are not allowed to leave it. Astravyx indexes underwriting guidelines, compliance material, internal runbooks, Salesforce metadata and vendor API docs, then answers with citations — and when the answer isn't in the index it says so instead of inventing one. Retrieval reconfigures across vector, keyword, hybrid and agentic modes depending on the question.

No model reaches users on reputation. A promotion gate scores every candidate against held-out golden sets and blocks the swap if it regresses; groundedness and citation accuracy are measured, not assumed. Anything touching a credit decision passes an ECOA / Regulation B firewall first, and the audit trail records how each answer was produced.

The penetration teardown

Why a worse dialer was beating ours

2026 · Python · faster-whisper · AWS S3 · Telnyx number API · **engine deployed**

The question was uncomfortable: our incumbent vendor's dialer was less configured than ours, had no AI, and still got answered far more often. The assumption in the room was cadence, or dial volume, or something in the model. I went looking in our own archive instead — 984,281 stored objects, of which 252,690 turned out to encode the outbound caller ID and agent right there in the filename for a ten-month window before the naming scheme changed to opaque identifiers. That accident is the whole reason the question was answerable at all.

It was none of the things we assumed. They were dialing from **6,406 distinct numbers across 310 area codes**. We had **23**. Eighty-six percent of theirs were used on a single day and then retired; four hundred to eleven hundred fresh ones were injected every month. Spam labelling is reputation-based, so a number that is one day old has no reputation to have ruined yet. They weren't avoiding the spam flags — they were outrunning them.

Which inverts the intuition completely: with 23 numbers, dialing *more* makes penetration *worse*, because every extra call deepens the reputation damage on the same tiny set. Every sophisticated thing we had built — the cadence engine, the AI, the fail-closed compliance gate — was sitting behind a twenty-three-number-wide front door.

So I built the rotation engine. It holds a target pool of fresh local numbers, plans purchases spread across the least-covered area codes nearest our actual lead geography, and retires numbers the health checker has flagged once they're past a minimum age. Because it spends real money, it is dry-run unless explicitly confirmed, capped on pool size and purchases per run, and structurally incapable of releasing the aged, branded numbers that carry our inbound and callback traffic — it can only ever retire what it bought itself.

PLATFORM & INTEGRATION

Salesforce ↔ wholesale-lender platform

The lending stack's spine

2025-2026 · Apex · Lightning Web Components · AWS Lambda · DynamoDB · **3 orgs in production**

Loan files move from Salesforce into the wholesale lender's API, and milestone events come back through webhooks into ten tracked date fields. Idempotency and monotonic rules mean a replayed or out-of-order event cannot corrupt a timeline — which sounds academic until a race condition strands eleven live loans. I built the buffering retry that stopped it recurring and the replay script that healed the eleven without anyone re-keying a date.

Conditions, document requirements, automated underwriting and credit pulls all ride the same path. Everything promotes sequentially through sandbox, QA and production with a validation-only deploy before each one. In a two-person Salesforce team the discipline is the only thing standing between you and a Friday outage.

UR Document Hub

Loan documents, inside the CRM

2026 · TypeScript · Next.js · AWS Amplify · Salesforce Named Credentials · **live in 3 orgs**

A document hub embedded directly into Salesforce through a Named Credential, so it reads as part of the CRM rather than another tab someone has to remember. It pulls the wholesale lender's outstanding conditions and reconciles them into a six-bucket working view — what's needed, what's submitted, what's cleared — so processors stop cross-referencing two systems by hand.

Live across all three Salesforce orgs and as a standalone web app on Amplify.

Lead Command Panel

The screen a loan officer works from

2026 · React · Vite · AWS Amplify · Salesforce · **in production**

A React app embedded in Salesforce over `postMessage` through a host Lightning component. It renders the live loan application fields as the voice agent fills them in mid-call, so an officer picking up a warm transfer sees the borrower's answers appear rather than starting from a blank record.

Lead Loader & drip queue

Sales-ops plumbing

2026 · Python · Salesforce · Five9 · internal

Getting leads where they need to be, without breaking the dialer. A GUI-driven loader writes campaign lists straight into the dialer's list configuration, builds rate-drop lists off pipeline criteria, and paces uploads through a throttled drip queue — per list, rather than dumping one enormous bulk job and watching the whole thing back up.

CUSTOMER-FACING PRODUCTS

urmortgages.com

Public site + online mortgage application

2026 · JavaScript · AWS Amplify · Route 53 · live

The company's public site, rebuilt to the brand rather than to a template, with the online mortgage application wired directly into it instead of bouncing applicants to a third-party portal. Consent is gated for TCPA before anything is captured, and an AI assistant backed by our own retrieval engine answers questions on the page. Deployed on Amplify with DNS moved onto Route 53.

Client Portal

Where the borrower sees their loan

2026 · TypeScript · Next.js · AWS Amplify · Salesforce · in production

A borrower-facing portal talking to Salesforce through a hub API, so document requests, outstanding conditions and loan status all come from the same source of truth the loan officers work in — no second copy to drift.

When it went down, everyone assumed credentials. It was a stale CORS origin on the Lambda after an environment moved. Worth saying because the debugging lesson — check what actually changed before you start resetting things — has saved more time since than the fix itself.

Sales leaderboard

Pipeline as a game the floor actually plays

2026 · Apex · Lightning Web Components · React · live in 3 orgs + web

Milestones score points, standings update live in Salesforce and on the web. The part that mattered was integrity: points are voided when a loan cancels or withdraws, so the board can't be farmed by writing business that never funds. Extended with a 3D avatar and emote store built on a shared skeleton so new characters drop in without new animation work.

SECURITY & INTERNAL TOOLING

Peacock

Authorized penetration-testing agent

2026 · TypeScript · Electron · Ed25519 · signed Windows installer

A kill-chain runner for infrastructure we own. Scope is an Ed25519-signed authorization the agent verifies before it will touch anything — a target outside the signed scope is refused, not warned about. Reconnaissance, vulnerability lookup, CVE correlation and reporting all run behind that gate, which is the difference between a security tool and a liability.

Raquazya

Codebase intrusion detection

2026 · PowerShell · Salesforce · AWS · GitHub

Watches for third-party changes across our Salesforce orgs, AWS accounts and GitHub repositories, flags anything originating outside a configured trust list, and catches device mismatches on owner logins. The rules are written against behaviour and never against a named person — a distinction worth designing in deliberately rather than defending later.

JIRANAMO

Desktop issue-tracker client

2026 · Electron · Python · signed auto-update

A native client for the tracker the team lives in. Its first auto-updater was precisely the vulnerability class it had no business being — unverified update payloads, CWE-494 — so I rebuilt the release path around Ed25519 signatures over both shipped binaries. The app now refuses any update it cannot verify, which is the only acceptable behaviour for software that replaces itself.

Nero

The engineering ledger

2026 · TypeScript · MCP · internal

Every project, repository, Salesforce org and Lambda under one roof, plus the exact place each hand — human or AI — left off. Built because with this many moving systems and two people, the expensive thing isn't writing the code, it's re-deriving where you were three weeks ago.

PERSONAL PROJECTS

Reliquary

Digital artifact exchange

2026 · TypeScript · Capacitor — iOS / Android · in development

A marketplace where every artifact is its own ticker. Fungible shares trade against a real central-limit order book, while specific serials and grades trade as individual objects. Royalties fund a published buyback floor, crate odds are provably fair rather than promised fair, and a geolocation hunt puts region-locked artifacts on a real map. Web first, wrapped for both mobile platforms from the same codebase.

Lati

3D social and streaming platform

2026 · TypeScript · Three.js · WebGL · in development

A consumer platform built around a real-time 3D interface rather than a feed — Three.js throughout, with social, streaming and cloud-gaming surfaces sharing one spatial shell.

NeuralForge

Machine-learning fundamentals, as a game

2026 · TypeScript · Next.js · Electron

An interactive study app covering 127 machine-learning terms with spaced repetition and boss-battle checkpoints, built on the MIT xPRO AI/ML glossary while I was working through the programme. The fastest way to learn the vocabulary turned out to be building the thing that tests you on it.

Capabilities.

Things I have shipped with, not things I have read about.

LANGUAGES	TypeScript · JavaScript · Python · **Apex** · PowerShell · SQL
AI & RETRIEVAL	OpenAI Realtime · Anthropic · Haystack · Ollama · Chroma · BM25 hybrid retrieval · faster-whisper · vLLM on ROCm · voice cloning (ElevenLabs, Cartesia) · evaluation harnesses and golden sets
CLOUD	AWS — Lambda, Amplify, S3, DynamoDB, Lightsail, IAM · Docker · Caddy · Cloudflare tunnels
SALESFORCE	Apex · Lightning Web Components · Flows · Named Credentials · custom metadata · permission sets · multi-org promotion and validation-first deploys
TELEPHONY	Telnyx · Twilio · SIP · Five9 · signed carrier webhooks · caller-ID rotation and reputation management · predictive pacing
INTERFACES	React · Next.js · Vite · Three.js · Electron · Capacitor · embedded Salesforce surfaces over postMessage

SECURITY	Ed25519 signing and verification · hash-chained tamper-evident logs · fail-closed authorization gates · intrusion and drift detection · authorized penetration testing
REGULATORY	ECOA / Regulation B · TCPA and do-not-call · two-party recording consent · state licensing boundaries · adverse-action reasoning · PII handling

One problem, in detail.

Because a list of systems tells you what someone shipped, and only one story tells you how they think when the thing refuses to work.

Our AI agent could already dial. What we needed next was a seat at the same table for a *human* — a loan officer's assistant wearing a headset, working a lead list through the browser, with the calls bridged, recorded and screen-popped exactly like the agent's. Every telephony vendor sells this as a solved problem. You register a softphone and you're done.

It did not work. It kept not working, in five different ways, and each failure was convincing enough to look like the answer.

01 **Telnyx WebRTC softphone** DEAD END

Intermittent `D11 · Destination Number is invalid` on roughly two-thirds of nail-ups. The credential connection reported `registration_status: Not Registered` with `updated_at: null` — it had never registered, not once.

02 **Fix the registration** WRONG PREMISE

Reordered the media permissions, surfaced the real errors, added onboarding. The browser reached `ready` and registration still never happened — because WebRTC clients connect over a WebSocket and never issue a SIP REGISTER at all. Dialing *into* a SIP address was unreachable by construction. The whole approach was wrong, not broken.

03 **Invert it — let the browser place its own leg** REJECTED BY CARRIER

The server would catch that leg inbound and use it as the agent side. This also surfaced a genuine bug worth having: the connection carried an empty `ani_override` set to *always*, so it blanked the caller ID on every browser call and the carrier dropped legs with no ANI. Fixed that, and the leg still died `INCOMPATIBLE_DESTINATION` — the carrier refuses to route a WebRTC connection to our own numbers on-net.

04 **Server dials the browser through a dedicated number** SAME WALL

Moved a DID onto the softphone connection so the call would ring the browser instead. Inbound was accepted by the carrier and never delivered to the client.

05

Browser media relay — reuse the AI's own audio path SHIPPED, VERIFIED

No SIP. No registration. No on-net routing. The browser only ever talks to our own server.

Before building the fifth, I made myself prove the first four were dead rather than merely unlucky. Two separate connections, a third created clean, a SIP subdomain, a native SIP client over UDP, TCP and TLS, and a full reboot — every path failed identically. That is not a bug in my code. That is an account-level provisioning defect, and the useful output of a day's debugging was a sentence: *on this account, only Call Control and PSTN work; inbound client registration never will.* I wrote it down where the next person would find it, because the expensive part wasn't the fix, it was the eight hours proving which door was locked.

Why can't I just do what the AI does?

THE QUESTION THAT ENDED IT

The AI agent had never needed a softphone. Its synthesized voice was already being injected straight into the live call's media stream. There was nothing special about that audio being synthetic — the pipeline just wanted frames.

So the headset became a source of frames. The browser captures the microphone, encodes to μ-law at 8 kHz, and streams it over a WebSocket to our server, which injects it into the call at precisely the point the AI's voice would have gone. The lead's audio returns the same way. The human sits inside the machine's plumbing, and the carrier never has to know a human is there.

It went live on 20 July: headset connects, the dialer runs multiple simultaneous lines, the first human answer wins and the losing lines are dropped and requeued, the call bridges, the Salesforce lead pops, a wrap timer runs, the disposition advances to the next lead. Automated machine detection was deliberately turned off in this mode — it was mis-flagging live humans as voicemail and discarding real leads, and a person wearing a headset can identify a voicemail perfectly well on their own.

The lesson I actually took from it is about sequence. Four architectures failed because I was trying to make our stack behave like the vendor's documentation said it should. The fifth worked because I stopped asking what the vendor supported and started asking what we already had running.

Get in touch.

I'm extremely happy where I am and I'm not looking to leave — but I am **100% open to contract work**. If you have something AI, platform or integration shaped that needs building rather than demoing, I'd like to hear about it.

This page also exists as a record. Anyone curious about how a two-person team runs autonomous voice, retrieval and underwriting inside a regulated industry has somewhere to look. Happy to talk shop either way.

Download résumé (PDF) ↓

EMAIL

demasse1125@gmail.com →

LINKEDIN

in/devdemasse → (https://www.linkedin.com/in/devdemasse/)

GITHUB

UR-MORTGAGE → (https://github.com/UR-MORTGAGE)

BASED IN

Utica, Michigan
