Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A multilingual large language model agent for wildfire evacuation guidance in limited English proficiency communities

Shruti Kulkarni
University of San Francisco
Lynn Tong
University of San Francisco
Aditi Namboodiripad
University of San Francisco
Chelyah Miller
University of San Francisco
Helen Lin
University of San Francisco
Peeyush Patel
University of San Francisco
Bogdan Bistriceanu
Accenture (United States)
Diane Woodbridge
dwoodbridge@usfca.edu

University of San Francisco

Research Article

Keywords: wildfire evacuation, agentic AI, large language models, multilingual emergency communication, disaster risk reduction, human-AI interaction

Posted Date: September 8th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10730069/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A multilingual large language model agent for wildfire evacuation guidance in limited English proficiency communities

Shruti Kulkarni[1*], Lynn Tong[1], Aditi Namboodiripad[1], Chelyah Miller[1], Helen Lin[1], Peeyush Patel[1], Bogdan Bistriceanu[2], Diane Myung-kyung Woodbridge[1*]

[1*]MS in Data Science and Artificial Intelligence Program, University of San Francisco.
[2]Accenture.

*Corresponding author(s). E-mail(s): skulkarni3@usfca.edu; dwoodbridge@usfca.edu;
Contributing authors: latong@usfca.edu; anamboodiripad@usfca.edu; cmiller9@usfca.edu; hlin65@usfca.edu; ppatel19@usfca.edu; bogdan.bistriceanu@accenture.com;

Abstract

Wildfire seasons in the United States are now an average of 84 days longer than in the 1970s, and wildfire smoke contributes to an estimated 40,000 deaths annually. Roughly 26 million U.S. residents have limited English proficiency (LEP), of whom 9.5 million speak neither English nor Spanish, yet over 80% of emergency alerts are still issued in English only. We present BEACON (Broadscale Evacuation Agentic Coordination, Outreach & Navigation), an open-source mobile application that provides comprehensive and personalized evacuation guidance in language. BEACON combines (i) WatchDuty real-time fire-perimeter polygons, (ii) NOAA HRRR wind, temperature, and moisture fields projected onto an Hourly Wildfire Potential (HWP) score, and (iii) polygon-avoidant routing through OpenRouteService whose returned GeoJSON paths are filtered to avoid active fire perimeters and high-HWP corridors. A Claude-driven agent auto-detects the user's language from each turn, injects the danger bin and household profile at session start, computes a safe route on the first route request, and degrades gracefully to "call 911" when no safe route exists. We describe BEACON's architecture, the danger-assessment algorithm, the multilingual prompting strategy, and the graceful-degradation protocols. BEACON was

demonstrated as a live application at the WiDS AI Forum 2026 and is, to our knowledge, the first publicly documented integration of polygon-avoidant routing with turn-level multilingual agentic dialog in an evacuation context.

Keywords: wildfire evacuation, agentic AI, large language models, multilingual emergency communication, disaster risk reduction, human-AI interaction

1 Introduction

Climate change has stretched the U.S. wildfire season by an average of 84 days compared with the 1970s, and the destructive footprint of recent fires — Camp (2018), Marshall (2021), Lahaina (2023), Palisades & Eaton (2025) — has shown that timely evacuation is no longer guaranteed. In each of these events, fast-moving fire behavior driven by strong winds and dry fuels overwhelmed warning and response systems that were designed for a slower-moving threat. The period between ignition and the moment residents must decide whether and where to evacuate often approaches the limits of what warning and evacuation systems can operationally deliver Synolakis and Karagiannis (2024). Wildfire smoke now contributes to an estimated 40,000 excess deaths per year in the contiguous United States, and the number of structures destroyed annually has more than doubled since 2010 Qiu et al. (2025); National Interagency Fire Center (2024).

According to the American Community Survey, approximately 25.7 million U.S. residents have limited English proficiency (LEP), of whom 9.5 million speak neither English nor Spanish Medicaid and CHIP Payment and Access Commission (2024). Despite this, over 80% of Wireless Emergency Alerts and state/county-level evacuation orders are issued exclusively in English. Federal mandates requiring multilingual emergency alerting including English and 14 additional languages do not take effect until June 2028 U.S. Government Accountability Office (2026). In a fast-onset wildfire, an evacuation order that the resident cannot read or understand in time is effectively no order at all.

In addition to the language barrier, we found two further operational gaps in evacuation support. First, commercial GPS services repeatedly returned evacuation routes that crossed active fire perimeters in our controlled tests against WatchDuty perimeter and evacuation-polygon data Watch Duty (2026), even after the perimeters and points-to-avoid were explicitly provided. This is a known limitation of optimization objectives that minimize travel time without polygon-level exclusion constraints. Second, most preparedness guidance published by agencies (e.g. ready.gov, and state- or county-level evacuation checklists) is static text, available only in English (and occasionally Spanish), and does not adapt to household situation — children, pets, livestock, mobility needs, medical needs, or vehicle availability.

This paper presents BEACON (Broadscale Evacuation Agentic Coordination, Outreach & Navigation), an open-source mobile application designed to close these gaps. BEACON is best characterized as an end-to-end agentic system, a streamlined pipeline of data ingestion, danger-assessment modeling, and polygon- and point-based routing,

fronted by a multilingual large language model (LLM) agent with structured prompt construction, tool-calling against the map and checklist, an embedding-backed cross-session memory, explicit graceful-degradation protocols, and cloud-native deployment infrastructure.

The major contributions of this paper include the following.

- Design and development of a personalized multilingual wildfire evacuation assistant agent that combines real-time fire-perimeter data, weather data, wind-adjusted danger scoring, polygon-avoidant routing, and an LLM agent that automatically detects and switches language (Section 6).
- Implementation and incorporation of National Oceanic and Atmospheric Administration (NOAA) Hourly Wildfire Potential (HWP) wind-adjustment layer that takes the standard distance-only danger score and re-bins it upward when the High-Resolution Rapid Refresh (HRRR) fields indicate hot, dry, gusty conditions blowing toward the user (Section 5) Dowell et al. (2022).
- Development of **polygon-avoidant routing pipeline** on OpenRouteService and WatchDuty perimeters. In controlled tests, we observed commercial routing services to be unreliable and often return a path crossing an active fire polygon. To overcome this, we implemented our own routing system by calculating routes and layering them on MapBox OpenRouteService (2026); Mapbox (2026).
- Implementation of **agent's systematic primitives** which includes structured multi-section system prompt re-assembled at each session start; three state-mutating tools that bind the chat surface to the map, the checklist, and the user-preference store (`get_evacuation_route`, `toggle_checklist_item`, `update_user_preferences`); a two-layer intent classifier that conditions tool exposure on each turn; and a Qdrant-backed embedding memory that carries durable household facts across sessions (Section 6).
- Articulation of **graceful-degradation protocols** for common real-world failure modes: no safe route, missing household context, degraded semantic intent classification, and cross-session memory unreachable. (Section 7).

The rest of the paper is organized as follows. Section 2 situates BEACON against prior work in wildfire evacuation decision-support, multilingual emergency communication, polygon-aware routing, and LLM-based agents for safety-critical domains. Section 3 gives a system-level overview of the data flow from a user's GPS ping to a localized conversational response. Section 4 catalogs the four primary data sources. Section 5 formalizes the danger-assessment and routing algorithms, including the adaptive HWP refresh classifier. Section 6 details the agentic layer — structured prompt construction, language detection, intent-conditioned tool exposure, history compression, and the Qdrant-backed embedding memory. Section 7 reports the behavioral results from the live demonstration sessions. Section 8 summarizes the contribution and sketches future work.

2 Related Work

The wildland–urban interface (WUI) evacuation literature has matured substantially over the past two decades, moving from trigger-based decision models integrated with fire behavior, road-network geometry, and community context. The Wildland-Urban Interface Evacuation (WUIVAC) model and its successors Cova et al. (2007); Ronchi et al. (2023) formalized this lineage by establishing spatial trigger buffers around at-risk communities. When a wildfire crosses one of these buffers, the model issues an evacuation order on the basis that the fire will otherwise reach the community before a safe evacuation can complete. Building on this trigger logic, directional-vulnerability analyzes Beverly and Forbes (2023) combine the physical geometry of the road network with the dynamic, wind-driven behavior of the fire.Town-scale work in the Pacific Northwest by Dye et al. Dye et al. (2021) shows that infrastructure-only metrics systematically underestimate real-world evacuation difficulty because they omit the social and logistical complexities such as vehicle availability, household composition, language access that determine whether a planned evacuation actually unfolds as modeled. However, most of these tools are designed for planners and emergency managers, not individual residents during a fire.

Uekusa and Matthewman's recent study shows that language barriers in emergency systems are not merely technical hurdles but are rooted in structural and ideological power imbalances that disproportionately endanger marginalized populations. The authors claim that it is often not logistically possible for a government to provide official services in every language or dialect Uekusa and Matthewman (2023). The Federal Communications Commission (FCC) has required wireless providers that participate in Wireless Emergency Alerts (WEA) to support alerts in 13 languages, including English, Arabic, Chinese, French, German, Haitian, Hindi, Italian, Korean, etc. which are commonly spoken in the United States, with implementation required by June 2028. Federal Communications Commission (2026). Unfortunately, those are limited to template-based alert messages, not covering broader communications.

The widespread deployment of LLMs has revolutionised multilingual communication by enabling real-time, context-aware translation across many languages and dialects. Modern frontier models produce nuanced, natural-sounding conversation even in under-resourced and less-spoken languages, though coverage and fluency remain uneven Shu et al. (2024). Two recurring risks in safety-critical deployments are (i) hallucination of operational specifics — shelter addresses, road closures, contact numbers — and (ii) drift across multilingual sessions, where a user's first message may be in one language and subsequent messages in another (for example, an English-speaking household member opens the app and a Spanish-speaking parent then takes the phone). BEACON addresses (i) by limiting agent authority to a small, fixed set of tools (onboarding, route, checklist) and instructing the agent to refuse to surface shelter information that did not come from a verified source. It addresses (ii) by anchoring language detection to the most recent user message rather than to a session-level setting, so the agent follows the user's latest language choice each turn (Section 6). The related but orthogonal problem of degraded performance on low-resource languages

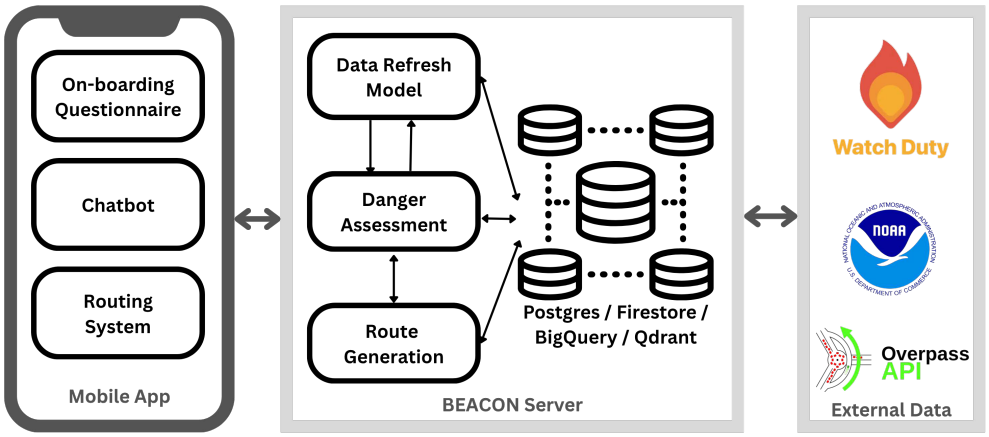


Fig. 1 BEACON end-to-end data flow from a user's GPS ping through danger assessment, polygon-avoidant routing, and a multilingual agent that returns turn-by-turn evacuation steps and a personalized checklist. The server-side container image is built using Docker and deployed to Google Cloud Run.

or code-switched input is a limitation that BEACON may inherit from the underlying model, which can be partially mitigated through a human-reviewed corpus of emergency strings.

Polygon-avoidant routing dynamically blocks specific geographic areas from a calculated path during mapping and navigation Kim and Atkins (2022). OpenRouteService (ORS) provides a polygon exclusion API that accepts an arbitrary GeoJSON polygon and returns paths guaranteed not to cross it OpenRouteService (2026). Commercial routing engines do not generally expose this primitive at the public API level. Recent work on emergency-vehicle routing under dynamic hazards has explored similar constraints, but typically for fleet dispatch rather than individual-resident way-finding Cheung et al. (2008). BEACON's end-to-end pipeline fetches WatchDuty polygons, merges overlapping perimeters, calls ORS, filtering candidate destinations by an HWP-weighted hazard score, and returns a GeoJSON collection that the mobile client can render without re-projection.

3 System Overview

BEACON is a client–server application. The client is a native iOS app built with Swift, while the server is a FastAPI service deployed on Google Cloud Run, packaged as a Docker container. The application is structured around a single session abstraction — one "session" per app open — with all state, including location, danger assessment, computed routes, chat history, household profile, and retrieved memories persisted under the user's identifier in a hybrid Postgres, Firestore, BigQuery, and Qdrant stores in the server (Figure 1, Table 1).

We distinguish two scopes that will recur throughout the paper. A *session* begins when the user opens the iOS client and the client calls `POST /chat/session/start`; it ends when the user closes the app or an idle timeout elapses. Each session has a GPS

snapshot, computed danger bin, safe route (or a `no_routes` fallback), household profile loaded from Postgres, and retrieved memories from Qdrant. These are computed once at session start and reused across the rest of the session. A *turn* is one user message and the agent's reply within a session. Per-turn operations are purely agent-layer: the intent classifier runs, tool exposure is recomputed, language detection is anchored to the latest message, and history is compressed if it exceeds the summarization threshold. Operations that are computationally expensive or whose inputs are stable within an evacuation are session-scoped, whereas operations that depend on what the user just said are turn-scoped.

3.1 Session lifecycle

On app open, the iOS client issues `POST /chat/session/start` with the user's GPS coordinates and an authentication token. The server performs the following operations.

1. *Resolves location.* The location service geocodes the coordinates to a human-readable display string for use in the agent's responses (Section 6).
2. *Checks for an onboarding record.* For logged-in users the server reads a single `household` record from Cloud SQL Postgres containing twelve boolean answers — e.g. `owns_home`, `has_car`, `has_pets`, `has_livestock`, `has_children`, `has_seniors`, `has_disabled`. New users receive `has_household=false` and are routed into a one-minute conversational onboarding flow.
3. *Assesses danger.* The evac-zone service queries a spatial BigQuery table of active fire perimeters within a radius (default 50 km, configurable per session) using `distance` against an indexed `geopoint`. Each hit is filtered on `is_active = true` and `evacuation_status IN {advisories, warnings, orders}`.
4. *Computes routes.* The maps service draws the candidate shelters from BigQuery (curated lodging and public shelter information from Overpass API and shelters from WatchDuty), candidate "drop-by" or "avoid" facilities, and dispatches them to the routing engine (Section 5) OpenStreetMap (2026).
5. *Builds the system prompt and retrieves memories.* The chat service composes a system prompt (Figure 2) containing the danger bin, the loaded safe route, language rules, and personalized checklist item IDs. It runs the Qdrant-backed memory store in parallel, retrieving data from prior sessions.
6. *Persists session state.* The session is written to Firestore under `/users/{user_id}/sessions/{session_id}`.

3.2 Per-turn flow

Each subsequent user turn triggers a two-layer intent classifier (Section 6). Layer 1 is a keyword-and-phrase matcher covering five intents including Route, Urgent, Checklist, Onboarding, Language, and TaskMention. Layer 2 is a semantic-fallback call to a lightweight model that fires only when Layer 1 returns None. The matched intents determine which tools the agent is allowed to call on that turn. For

Table 1 Persistence tiers in the BEACON server layer. Each store is chosen for the shape of the data and its access pattern.

Store	Type	What it holds	Why this store
Postgres	Relational	Users, household onboarding answers (12 booleans), checklist progress	Durable per-user state with relational integrity
Firestore	NoSQL document	Per-session location, evac data, language, history (last 20 turns), checklist item IDs	Schema-flexible, fast per-user document reads/writes
BigQuery	Columnar	Fire perimeters, HWP grid, curated and event-tagged shelters, Overpass facilities	Spatial filtering at scale
Qdrant Cloud	Vector index	Cross-session data, including composition, property, concerns, preferences, language, etc.	Semantic retrieval over free-text memories

instance, on a LANGUAGE-only turn — where the user has asked only to switch language — the route tool is withheld so the agent cannot trigger a fresh routing pipeline, which may run over the already-loaded path.

After the system prompt and tool envelope are assembled, the agent returns a reply through three runtime mechanisms.

1. *Streaming.* The reply is streamed token-by-token from Claude to the iOS client over Server-Sent Events (SSE), so the user sees the response appear as it is generated rather than waiting for the full reply to assemble. Markdown formatting in the reply — lists, bold, links — is rendered natively by the chat view.
2. *Tool-use loop.* When the agent invokes one of its three tools (`get_evacuation_route`, `toggle_checklist_item`, `update_user_preferences`), it emits a tool-use block instead of a reply. The server executes the corresponding function on the server and writes the result back into the conversation as a synthetic `tool_result` message, which the agent consumes in its next round before producing the user-facing reply.
3. *History compression.* When the running transcript exceeds a configured threshold (20 turns by default) the chat service calls Claude API once with the entire history and the instruction to "summarize this emergency chat concisely, keeping all safety details, locations, and decisions made". The summary replaces all prior turns as a single synthetic exchange, and the subsequent turns proceed from this compressed state, which keeps the agent inside its context window without forcing it to lose the safety-critical state of the evacuation.

3.3 Service map

The server is organized into FastAPI routes and services that encapsulate I/O against the four backing stores. The key services and their routes are:

1. `check_evac` - evacuation-zone retrieval against BigQuery using `distance` (Section 4).
2. `wildfire_potential` & `hwp_pipeline` — HRRR ingest via the Herbie library, HWP computation and 9×9 spatial smoothing (Section 5). 9×9 spatial smoothing is a 2D spatial-average filter to suppress potential sharp swings between neighboring HWP cells in its 3 km resolution. BEACON replaces each cell's HWP value with the average of itself and its 80 nearest neighbors in a 9×9 box.
3. `ors_route` - polygon-avoidant routing through OpenRouteService, drop-by or avoid point, and per-language instruction rendering (Section 5).
4. `chatbot`, `chat_intent`, `memory_service`, `chat_store` - the agentic layer described in Section 6.
5. `onboarding`, `user_preferences`, `checklist` - household profile, language preference, and personalized checklist generation.
6. `facilities_search` - Overpass API client calls or WatchDuty data search for potential shelters and facilities.
7. `fcm` - Firebase Cloud Messaging for push alerts in case of no safe route available, fire danger, or evacuation order detected.

4 Data Sources

BEACON draws on four primary public or semi-public data sources. We summarize each below, along with its role in the pipeline and the key freshness/quality assumptions we make.

4.1 Watch Duty data

WatchDuty is a nonprofit emergency information service operated by active or retired firefighters, dispatchers, first responders, and reporters. They utilize official sources and radio communications to provide timely information. While artificial intelligence is employed for tasks such as smoke detection from images, transcribing radio communications, and parsing official documents, human oversight remains essential, and every alert is issued by a human WatchDuty (2026). As WatchDuty does not offer public APIs, we utilized data collected between January 21, 2025 and September 5, 2025, used for the 2026 Women in Data Science Worldwide (WiDS) Datathon Women in Data Science Worldwide (2026). This data includes active fire perimeters, evacuation zones, and shelter information.

In our pipeline, fire perimeters, evacuation zones, and shelters are stored in BigQuery with `geo_json` columns of a flattened table, where each row carries an active flag, status, and timestamps. The danger-assessment query uses these columns to filter to currently active polygons within the configured search radius.

4.2 NOAA HRRR meteorological data

The NOAA HRRR (High-Resolution Rapid Refresh) model produces hourly analysis on a 3 km continental United States (CONUS) grid. BEACON uses the surface analysis fields including wind gust (`GUST`), 2-m temperature (`TMP`), 2-m dewpoint (`DPT`), soil moisture availability (`MSTAV`), and snow water-equivalent (`WEASD`). Data are retrieved via the Herbie library, where the most recent analysis with an upload buffer of approximately one hour is used to compute the HWP grid Blaylock (2026); Global System Laboratories, National Oceanic & Atmospheric Administration, U.S. Department of Commerce (2026).

4.3 User GPS

At session start, the iOS client transmits the user's most recent GPS value, including `(latitude, longitude)` with a timestamp. The location is used three ways: (i) to compute the Haversine distance from the user to the nearest active fire perimeter, (ii) as the origin for the routing engine, and (iii) to identify the nearest HWP grid cell. For privacy, BEACON does not persist raw GPS traces; only the session's snapshot location is stored in Firestore.

4.4 Shelters and drop-by facilities

Candidate evacuation destinations come from two sources: an event-tagged shelter table derived from WatchDuty geo-events data, and potential shelters including schools, fairgrounds, and community centers typically used as evacuation facilities DiGrande et al. (2019). The two are concatenated and de-duplicated before being passed to the routing engine. A point of interest or avoidance, such as pharmacies, supermarkets, and convenience stores, is pulled from OpenStreetMap via the Overpass API and cached in BigQuery to keep query cost bounded and enhance search speed.

4.5 Privacy and identifier handling

BEACON collects only the minimum personal data needed to render a useful evacuation response. Authentication is delegated to Firebase Authentication, which issues an opaque, persistent user identifier (user_id) that is the key in all server-side stores. Household-onboarding answers are twelve booleans without any personally identifiable information (PII). For privacy, GPS coordinates are kept at the granularity sent by the device and are not derived into long-term mobility profiles. Instead, a single snapshot per session is stored in Firestore and ages out with the session. The retrieval-memory store (Qdrant) keeps user-scoped memories at the collection level, retrieved only under the `user_id` that produced them, encrypted at rest by the managed-Qdrant provider.

5 Danger Assessment and Polygon-Avoidant Routing

BEACON's analytical layer has two responsibilities: (i) translate the user's location, the active fire perimeters, and the local meteorology into a danger bin, and (ii) compute one or more polygon-avoidant evacuation routes ranked by both distance and hazard levels.

5.1 Danger assessment

5.1.1 Distance-based base score

The base danger assessment is a function of the Haversine distance from the user's GPS point to the boundary of the nearest active fire perimeter, $d_{\min}$. Active perimeters are retrieved from the WatchDuty data using the spatial filter

$$\texttt{distance}(\texttt{geo_json}, \texttt{geopoint}(\lambda, \varphi)) \leq R, \tag{1}$$

where (λ, φ) is the user's longitude/latitude and R is the configurable search radius (default 50 km). Polygons are then filtered on `is_active = true` and `status` $\in$ {`advisories, warnings, orders`} before the nearest-perimeter distance is computed.

Then it maps the result to one of three ordinal urgency levels, conditioned on whether a safe route could be computed and on the count n of active zones returned:

$$b_0 \;=\; \begin{cases} \textsc{Critical}, & \text{no safe route exists,} \\ \textsc{High}, & n > 0 \text{ active zones within } R, \\ \textsc{Low}, & n = 0 \text{ active zones within } R. \end{cases} \tag{2}$$

Critical therefore signals a routing-pipeline failure and forces the agent to direct the user to call 911. High signals proximity to active zones. Low is the default all-clear.

5.1.2 Hourly Wildfire Potential (HWP)

The fire-weather companion to the urgency assignment is an Hourly Wildfire Potential (HWP) score, computed on the HRRR grid from five surface fields: wind gust (`GUST`), 2-m temperature (`TMP`), 2-m dewpoint (`DPT`), soil moisture availability (`MSTAV`), and snow water equivalent (`WEASD`). We adopt a simplified HWP formulation derived from the operational HRRR fire-weather literature James et al. (2025); National Oceanic and Atmospheric Administration (2026):

$$HWP = 0.213 \cdot G^{1.50} \cdot \mathrm{VPD}^{0.73} \cdot (1 - M)^{5.10} \cdot S, \tag{3}$$

where

- G is the surface wind gust in $\mathrm{m\,s^{-1}}$, lower-bounded at 3 $\mathrm{m\,s^{-1}}$ to avoid pathological scores in near-calm conditions;
- VPD is the 2-m vapor pressure deficit in hPa, computed from `TMP` and `DPT`;
- $M \in [0, 1]$ is the soil moisture availability fraction derived from `MSTAV`/100;
- $S = \exp(-\texttt{WEASD}/10)$ is a snow suppression term.

The raw grid HWP is combined with a 9×9 uniform box kernel (approximately 27 km × 27 km at the 3 km HRRR resolution) to suppress single-cell noise, and the user's HWP value is taken from the nearest grid point by Euclidean distance in

latitude and longitude space.

$$\widetilde{\mathrm{hwp}}(i,j) = \frac{1}{81} \sum_{\Delta i=-4}^{4} \sum_{\Delta j=-4}^{4} \mathrm{hwp}(i+\Delta i, j+\Delta j) \tag{4}$$

The HWP score is consumed by the routing pipeline, where candidate shelter routes whose length exceeds a configurable fraction $\phi_{\max}$ in cells with $hwp > \tau_{hwp}$ are discarded.

5.1.3 Adaptive HWP refresh model

The hourly HWP grid described in Section 5.1.2 is costly to recompute — each refresh pulls five HRRR surface fields via Herbie, evaluates the HWP formula on the whole Colorado bounding box, applies the 9×9 smoothing convolution, and writes the result to BigQuery. A strict hourly cadence is wasteful during calm periods and arguably too slow during rapid weather escalations. We therefore prototyped an *adaptive refresh-cadence model* that classifies the forthcoming twelve-hour HWP regime at each grid cell and schedules the next refresh accordingly.

We pose the refresh decision as a per-cell four-class classification problem with target

$$y \;=\; \max_{t \in [t_0+1,\, t_0+12]} HWP(t), \tag{5}$$

binned into Safe ($y < 10$), Elevated ($10 \le y < 20$), High ($20 \le y < 50$), and Extreme ($y \ge 50$), where the thresholds are those used by the routing-stage discard rule (Section 5.2.2).

For each grid cell at time t_0 we compute ten predictors, materialized as a BigQuery view.

- *Temporal:* `hour_of_day`, `month_of_year`;
- *Current and lag values:* $HWP(t_0)$, $HWP(t_0{-}1)$, $HWP(t_0{-}3)$, $HWP(t_0{-}12)$, $HWP(t_0{-}24)$;
- *Momentum:* $HWP(t_0{-}1) - HWP(t_0{-}3)$ and $HWP(t_0{-}1) - HWP(t_0{-}12)$;
- *Rolling mean:* the trailing 24-hour mean of HWP at the cell.

The view is partitioned by (λ, φ) so that lag and rolling-mean windows are computed per cell.

The classifier is a BigQuery ML `BOOSTED_TREE_CLASSIFIER`, gradient-boosted decision trees using the underlying XGBoost library trained with `auto_class_weights=TRUE` on HWP history data.

Given a predicted class $\hat{c} \in \{0, 1, 2, 3\}$ corresponding to $\{\text{Safe}, \text{Elevated}, \text{High}, \text{Extreme}\}$, the scheduling helper sets the next refresh interval to

$$\Delta t(\hat{c}) \;=\; \begin{cases} 24 \text{ h}, & \hat{c} = \text{Safe}, \\ 6 \text{ h}, & \hat{c} = \text{Elevated}, \\ 3 \text{ h}, & \hat{c} = \text{High}, \\ 1 \text{ h}, & \hat{c} = \text{Extreme}. \end{cases} \tag{6}$$

Cells without danger defer their next compute by up to a day while cells on a worsening trajectory refresh as often as the underlying HRRR cadence permits. Cells in the same Colorado bounding box are scheduled independently, so an escalation in one corner does not force a full-grid refresh.

5.1.4 Agent tone calibration

The three-level bin is consumed by the agentic layer (Section 6) in two ways: it is rendered into the system prompt as "fire urgency level: LOW/HIGH/CRITICAL", and it selects from a small library of *tone snippets* that shape the agent's voice from calm-preparedness in LOW to "no safe route available, call 911" in CRITICAL.

5.2 Polygon-avoidant routing

Commercial mapping platforms are attractive baselines because of their reach and the familiarity of their UI to non-expert users. However, in our controlled tests, they exposed two limitations for our use case. First, many commercial platforms we evaluated offer no first-class mechanism for rendering external fire-perimeter and evacuation zone polygons as overlays on the displayed map, so the user sees the route but not the hazard the route is supposed to avoid. Second, the routes those platforms returned repeatedly crossed active fire polygons even when the polygons had been supplied through their avoid-areas APIs. The avoid-area parameters they expose are *soft* hints, not hard exclusion constraints. We therefore switched to OpenRouteService (ORS), whose `directions` endpoint accepts a GeoJSON object as a *hard* constraint and returns a path guaranteed not to cross any supplied polygon.

5.2.1 Candidate destinations and avoidance geometry

The routing pipeline accepts a list of candidate shelters and, optionally, two distinct kinds of points of interest pulled from the Overpass API and cached in BigQuery: *waypoints* the user wants to visit en route, such as pharmacies or grocery stores; and *points of avoidance* the user must avoid, such as reported accidents, road closures, or downed power lines. For each candidate shelter, we issue an ORS driving request from the user's GPS origin, supplying two geometries that jointly constrain the returned path.

Avoidance geometry (A)

The set of regions the returned route must not cross is the union of all active fire perimeter polygons returned by the danger-assessment service and point hazards lifted into small polygons by buffering each point with a fixed radius r. Formally,

$$A \;=\; \Big(\bigcup_i p_i\Big) \;\cup\; \Big(\bigcup_j B_r(q_j)\Big), \tag{7}$$

where each p_i is an active perimeter polygon and $B_r(q_j)$ is the disk of radius r around a point of avoidance q_j. Buffering each point into a small polygon lets us treat polygon hazards and point hazards through a single ORS primitive (`avoid_polygons`).

Waypoint sequence (W)

The route is computed as a multi-stop path rather than a single origin-to-destination segment. The waypoint sequence is

$$W = \big(o,\ w_1,\ w_2,\ \ldots,\ w_k,\ s\big), \tag{8}$$

where o is the user's GPS origin, s is the candidate shelter, and $w_1, \ldots, w_k$ are zero or more drop-by waypoints selected by the user (e.g. a pharmacy to collect medication, a grocery store to refill supplies, a relative's address en route). When no drop-by is requested, the sequence collapses to the two-stop case $W = (o, s)$.

The pair (A, W) is the full instruction handed to ORS: the returned geometric line object starts at o, visits each intermediate waypoint w_l in order, ends at s, and is guaranteed to lie entirely outside A.

5.2.2 HWP-aware shelter ranking

A polygon-avoidant route is necessary but not sufficient: a route that skirts the fire only to spend most of its length in a high-HWP corridor is also unsafe. Each candidate route returned by ORS is therefore scored on two axes:

1. Geodesic length ℓ;
2. The fraction ϕ of the route's length that lies in cells whose HWP exceeds an operational threshold τ_{hwp}.

Candidates with $\phi > \phi_{\max}$ (i.e. 10% of route length in high-HWP cells) are discarded. Among the remaining candidates, one with the shortest ℓ is returned. If the discard list exhausts all candidates, the pipeline returns a `no_routes` status, which the agent renders as an emergency instruction and which triggers a push alert via Firebase Cloud Messaging.

5.2.3 Output

The routing service emits a single GeoJSON object containing the line object of the selected route, the union avoidance geometry A rendered as fill polygons, and point features for any drop-by waypoints in W. The mobile client renders all three layers in Mapbox without further reprojection.

5.3 Multilingual instruction rendering

ORS natively supports a fixed set of languages for turn-by-turn instruction text (English, Spanish, French, German, Italian, Japanese, Korean, Portuguese, Russian, Chinese, and a long tail of European languages, totalling 23 at the time of writing). When the user's preferred language is supported, instructions are returned in that language directly. When not, the route is requested in English and the instruction strings are passed through a Claude translation step keyed to the user's language code. This fallback permits us to expose the full Claude-supported language envelope to the user without depending on the routing provider for low-resource-language coverage.

Table 2 Agent primitives and the persistent stores they bind.

Primitive	Backing store	Role
`get_evacuation_route`	BigQuery + ORS	Map (GeoJSON to client)
`toggle_checklist_item`	Postgres	Checklist (per-item state)
`update_user_preferences`	Postgres	Language preference
Session context injection	Firestore	Per-session state
Embedding memory	Qdrant Cloud	Cross-session facts

Table 3 LLM and embedding models used.

Model	Provider	Role
`claude-sonnet-4-6`	Anthropic	Main conversational agent: streaming response, tool calling, governance refusal patterns.
`claude-haiku-4-5`	Anthropic	Lightweight Claude tasks: history summarization, `mem0` fact extraction, and ORS instruction translation for languages outside the 23-language ORS envelope.
`gpt-4o-mini`	OpenAI	Layer-2 semantic intent fallback when the Layer-1 keyword classifier returns `Intent.NONE`.
`text-embedding-3-small`	OpenAI	Embedding model for the Qdrant cross-session memory store, populated via `mem0`.

6 The Multilingual Agentic Layer

BEACON's conversational surface is an LLM-driven *agent*, not a chatbot. Concretely the agent has four systematic primitives: (i) a structured, six-block system prompt re-assembled each session; (ii) three tools that mutate the persistent state of `get_evacuation_route`, `toggle_checklist_item`, and `update_user_preferences`; (iii) a two-layer intent classifier that gates which tools are exposed on each turn; and (iv) a Qdrant-backed embedding memory that carries durable household facts across sessions. Together, these primitives bind the chat surface tightly to the map for route generation and to the checklist for preparedness-task toggling, so the agent is not a free-floating chatbot but the conversational projection of a single underlying state.

Table 3 presents the LLM and embedding models used across BEACON's agentic layer, with their call scope. The split between a higher-capability model (`claude-sonnet-4-6`) for the user-facing conversational agent and lighter-weight models for auxiliary tasks (semantic classification, summarization, and fact extraction) keeps the per-turn cost bounded while preserving response quality on the critical user-facing path.

6.1 Context injection

The system prompt is constructed once per session at the session start by the `build_system_prompt` routine (Figure 2):

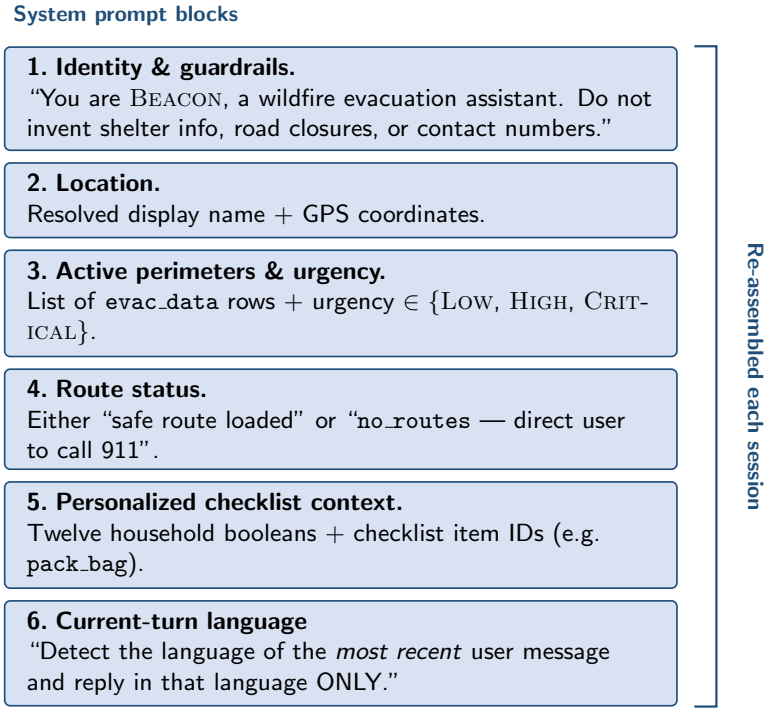


Fig. 2 Skeleton of the system prompt assembled at session start. The "Current-turn language" block is the highest-priority instruction; the agent is required to detect language from the most recent user message rather than from a session-level setting.

1. **Identity**: a one-paragraph statement of who the assistant is ("BEACON, a wildfire evacuation assistant") and what it must and must not do (i.e, no clinical advice, no unverified information).
2. **Location**: the resolved display name and raw coordinates of the user.
3. **Active perimeters and urgency**: the active evacuation zones returned by `check_evac`, with the urgency level pre-computed (LOW/HIGH/CRITICAL).
4. **Route status**: either "a safe route has been loaded and is visible on the map" or "no safe route could be computed — direct the user to call 911".
5. **Personalized checklist context**: the household's twelve boolean answers and the list of checklist item IDs (e.g. `pack_bag`). The agent is instructed to refer to checklist items by their human name, but to use the IDs internally when calling the `toggle_checklist_item` tool.
6. **Language rules**: the most-recent-message anchoring instruction.

When the user has not completed household onboarding, the personalized checklist context is omitted from the system prompt, and a short *onboarding-notice block* is appended in its place. The notice instructs the agent to encourage the user to complete the onboarding questions in the Checklist tab, and then explain the evacuation checklist purpose.

6.2 Language detection

A common pitfall in multilingual LLM deployment is to rely on a single, session-level language setting. This fails as soon as the user's first message is in their second language — for instance, a Spanish-speaking household member opens the app in a panic and types in English, then their parent takes the phone and types in Spanish. BEACON addresses this by inserting a single highest-priority instruction into the system prompt:

> *== CURRENT TURN LANGUAGE — HIGHEST PRIORITY ==*
> *Detect the language of THE MOST RECENT USER MESSAGE and reply in that language ONLY. Do not preemptively switch languages; switch only when the user does. Persist language switches to the user's profile only on non-emergency turns.*

When the agent detects that the user has switched language, it can invoke the `update_user_preferences` tool to persist the change.

6.3 Two-layer intent classification

Per-turn intent classification is the mechanism by which we shape the tool envelope. The classifier returns a set of one or more intents from the enumeration of {ROUTE, URGENT, CHECKLIST, ONBOARDING, LANGUAGE, TASKMENTION}, combined as bit-flags so that multiple can co-occur (e.g. a typical URGENT OR ROUTE message is "show me my evacuation route — flames are close", which contains both an URGENT keyword and a ROUTE keyword).

Layer 1 — keyword/phrase matching.

Each intent has its own phrase set, and the user's message is tested for substring membership.

URGENT covers "evacuate", "flames", "trapped", "smoke outside" and a short list of related English phrases. TASKMENTION matches past-tense completion phrases ("packed the go-bag", "fueled my car") but is suppressed when the same turn either contains a forward-looking phrase ("want to", "need to", "going to") or also fires the ONBOARDING intent. For instance, "I got a new dog" updates the household profile rather than toggling a pet-carrier item on the checklist.

Layer 2 — semantic fallback.

When Layer 1 returns NONE, we invoke a lightweight classifier over a fixed five-category taxonomy: *onboarding*, *checklist*, *route*, *urgent*, *general*. This fallback adds approximately 100–150 ms of latency at the median and costs roughly $0.00015 per call.

6.4 Tool exposure conditioned on intent

The agent has three tools:

`get_evacuation_route` returns the GeoJSON route computed in Section 5.

toggle_checklist_item toggles a single preparedness item by ID and persists the change to the `checklist_progress` Postgres table.
update_user_preferences updates the user's language and other settings.

These are exposed conditionally to suppress spurious calls. The route tool is suppressed on a LANGUAGE-only turn; the checklist tool is suppressed on an URGENT + ROUTE emergency turn; the language tool is suppressed on emergency turns to avoid mid-evacuation persistence of an unrelated preference change.

6.5 History summarization

The agent's context window is finite; for long sessions, we compress the prior history using the agent itself. When the number of turns exceeds a configured threshold, the chat service calls Claude once with the entire history and the instruction to "summarize this emergency chat concisely, keeping all safety details, and decisions made".

The returned summary replaces the prior turns as a single synthetic exchange:

> `user`: *Summary of our conversation so far: ...*
> `assistant`: *Understood. I have the context from our previous conversation.*

This preserves the safety context, including where the user is, what danger bin we last assessed, and what they have already packed, without incurring the token cost of replaying the full transcript on every turn.

6.6 Embedding-based cross-session memory

Beyond the per-session prompt, BEACON maintains a durable, embedding-indexed memory. The store is a Qdrant Cloud collection (`beacon_user_memories`), populated via the `mem0` library with Anthropic Claude as the fact-extraction LLM and OpenAI `text-embedding-3-small` as the embedding model. On each post-stream hook, we call `add_memory(user_id, last_exchange)` and let `mem0` decide whether the exchange contains a memorable fact.

The extraction prompt is constrained to a closed taxonomy — household composition, property facts, stated concerns, evacuation preferences and language. On session start, we retrieve the top-k=5 memories matching the seed query *"user household facts: pets, children, seniors, mobility needs, property features, stated concerns, evacuation preferences, language"* and inject them into the system prompt under a header titled *"what I know about this user"*.

This design separates the three persistence tiers cleanly: **operational state** (current session, current location, current route) lives in Firestore; **structured profile** (twelve onboarding booleans, checklist progress) lives in Postgres; **semantic memory** (free-text household facts learned from conversation) lives in Qdrant. The three are joined only inside the system prompt, never in the data layer — a separation that keeps the operational path fast and the memory layer purely additive.

6.7 Governance and refusal patterns

The agent operates inside three guardrails:

- **No invented operational specifics.** The agent is instructed to refuse to invent shelter addresses, road closures, phone numbers or other operational facts that did not come from a verified source.
- **Escalation at system limits.** When no safe route can be computed, the agent is instructed to explicitly tell the user to call 911.
- **Restricted scope on emergency turns.** On URGENT or ROUTE turns, the agent is instructed to suppress non-essential topics (onboarding questions, language-preference updates, checklist progress celebrations).

7 Results from the Live Demonstration

Because this is an application paper rather than a benchmark study, the results below are qualitative observations of end-to-end system behavior during demonstration sessions, focused on the four properties we set out to deliver: (i) automatic language detection without configuration; (ii) polygon-avoidant routing under realistic fire perimeters; (iii) full situational context from the first turn of a session; and (iv) graceful degradation when one or more inputs are unavailable.

Due to the availability of WatchDuty data, we concentrated on Colorado data from July to September 2025, where there were nine fires, each burning more than 1,000 acres.

7.1 Automatic language detection without configuration

Across demonstration sessions, the agent reliably identified the user's language from the first message and replied in the same language without manual selection. Furthermore, the entire user interface on the application appeared in the user's language. We also verified that the agent honors mid-session switches, where the agent followed each switch from the most-recent message without preserving the previous-turn language. Figure 3 shows a danger assessment for Montrose County, Colorado; Figure 4 shows the personalized checklist rendered in Spanish under the heading "Lista de verificación de evacuación" after using Spanish.

7.2 Polygon-avoidant routing under realistic perimeters

With WatchDuty polygons of an active perimeter near Montrose County, Colorado, the polygon-avoidant routing pipeline consistently returned paths that avoided every active fire polygon (Figure 5). For the same input geometry, the baseline commercial routing APIs in our preliminary testing returned paths that crossed at least one active perimeter, which is described in Section 5.

7.3 Full situational awareness from the first turn

Because the system prompt assembled at session start contains the danger bin, the loaded safe route, the household profile, and the top-k retrieved memories, the agent responds with full context on the very first user turn. In demonstration sessions, a user's opening message "Should I leave?" produced a tailored response that referenced



Fig. 3 Agent responding with a danger assessment and evacuation prompts.

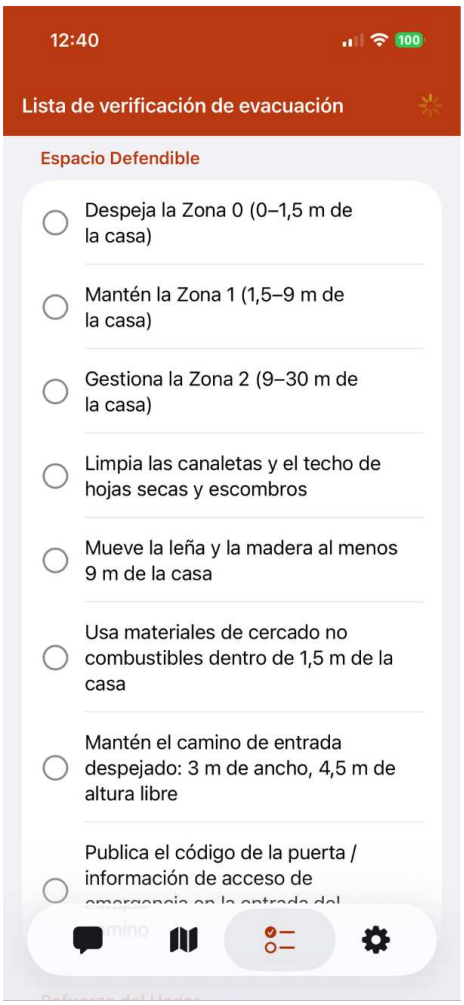


Fig. 4 Evacuation checklist rendered in Spanish.

the user's distance to the nearest perimeter, their pre-loaded route, and household-specific items (e.g. pet carriers, livestock evacuation) without the user having to ask. This stands in contrast to a more typical chatbot architecture in which the user has to volunteer their context across several turns.

7.4 Graceful degradation under operational failures

A core design principle of BEACON is that no single integration failure should cause the system to fall silent in front of a user who may be in immediate danger. Each external service — the WatchDuty perimeter feed, the OpenRouteService routing engine, the Anthropic and OpenAI LLM endpoints, the Qdrant memory store, the Firestore

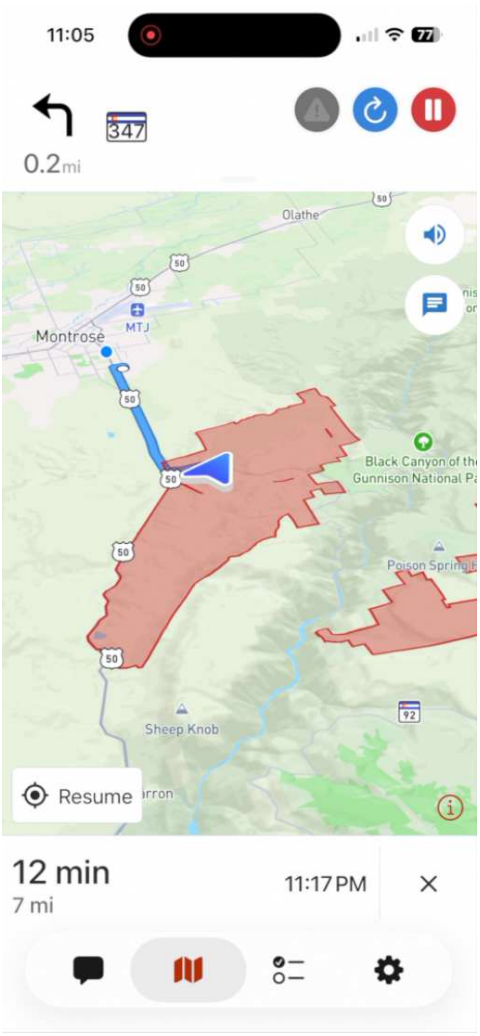


Fig. 5 Polygon-avoidant evacuation route from the user's GPS origin in Montrose County to the nearest ORS-computed shelter, rendered in Mapbox over the union of active fire perimeters. In this example, the user's origin was within the escape order zone.

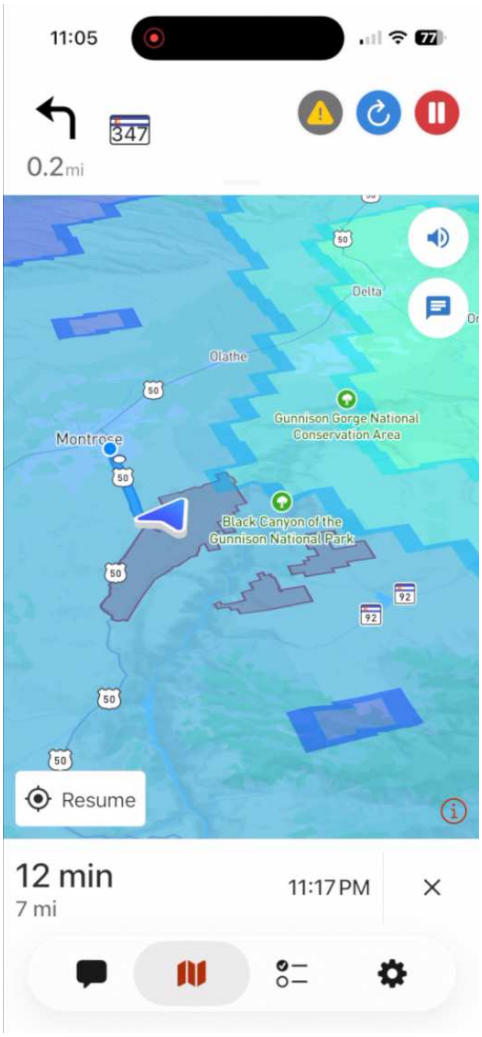


Fig. 6 Polygon-avoidant evacuation route with HWP overlay over Figure 5.

session store, the Cloud SQL Postgres checklist store — is wrapped in code paths that either continue with a degraded but still-useful response or escalate explicitly to emergency services. We summarize the patterns exercised during pre-deployment testing.

- **No safe route.** When the routing pipeline exhausts all shelter candidates and returns `no_routes`, the agent's response switches to a "call 911 now" instruction,

the iOS client surfaces the dial-911 action, and a push notification is dispatched via Firebase Cloud Messaging as a redundant channel.

- **Drop-by constraint infeasible.** If a user-requested drop-by waypoint rules out every shelter — for example because every candidate route that includes the waypoint exceeds the HWP budget — the pipeline retries without the drop-by and tells the user that the stop was not feasible, rather than returning no route.
- **Routing language unsupported.** If the user's preferred language falls outside the 23-language envelope natively supported by ORS, the route is requested in English and the turn-by-turn instructions are translated via a Claude call keyed to the user's language code, so the user still receives instructions in their own language.
- **Semantic classifier failure.** If the Layer-2 GPT call fails or times out, the per-turn intent defaults to `Intent.NONE` and the agent responds from the system prompt alone, without tool exposure. Latency on these turns is bounded by the keyword pass (Layer 1) plus the failed-API timeout.
- **Cross-session memory unreachable.** If the Qdrant retrieval errors or returns no results, the *What I Know About This User* block is omitted from the system prompt; the agent operates from the per-session context only and continues to serve the user.
- **Session-state save failure.** Each call to Firestore `save_session` is wrapped in a fire-and-forget try/except: a failed save is logged but does not block the response stream, so a Firestore outage does not prevent the user from getting their current-turn answer.
- **Missing household onboarding.** When the user has no `household` row in Postgres, the personalized checklist context is omitted from the system prompt and a new-user notice block is appended instructing the agent to introduce itself and encourage the user to complete the twelve household questions in the iOS Checklist tab.

In every observed degraded session, the agent continued to provide some useful response, and the escalation paths to emergency services were preserved.

7.5 Adaptive HWP refresh

We evaluate the adaptive refresh policy (Section 5.1.3) by replaying the September 2025 holdout window against the trained classifier and comparing the implied refresh schedule to the strict hourly baseline. Treating each cell-hour as contributing $1/\Delta t(\hat{c})$ refreshes under the adaptive policy and one refresh under the baseline, the mean predicted refresh interval across the held-out month is approximately 8.6 hours, corresponding to a reduction in HRRR fetches and HWP recomputes of roughly 88% relative to the hourly baseline. The 1-hour cadence is preserved for cells predicted in the EXTREME class.

As a safety check, we measure the fraction of cell-hours predicted SAFE (deferred 24 h) whose true 12-hour-ahead HWP maximum exceeded the ELEVATED threshold of 10. From the per-class recall on the held-out window (SAFE $= 0.65$), the resulting miss rate is non-trivial; the production scheduler therefore augments the model with an

unconditional refresh on receipt of an external NOAA Red Flag Warning or equivalent official fire-weather alert for the bounding box.

7.6 Operational latency profile

We repeated end-to-end walk-throughs of the cold user path including account signup, household onboarding, session start, asking the agent for an evacuation route completed in under three minutes consistently. The following per-stage observations situate where the time is spent:

- **Session start.** Cold session-start latency (GPS → first streamed token of the agent's opening turn) is dominated by two server-side calls run in parallel: the danger-assessment BigQuery query (`distance` over the active-perimeter table) and the Qdrant memory retrieval. The two are awaited together so the session-start wall-clock latency is the *slower* of the two rather than their sum.
- **Route generation.** The evacuation route is not computed at session start; it is generated on demand when the user asks for it, invoking `get_evacuation_route` tool. Route generation is the long pole of any route turn because it issues one ORS directions request per candidate shelter, plus an HWP lookup per cell along each candidate route, before selecting the best path.
- **Non-interactive cost: HRRR ingest.** The hourly HRRR fetch and HWP computation run on a scheduled job outside the user-facing path, and the resulting grid is amortized across all users. The dominant costs are the GRIB downloads through Herbie and the 9×9 smoothing convolution.

These observations are consistent with the under-three-minute end-to-end cold path: the highest user-visible costs are the session-start parallel-await on BigQuery and Qdrant, the on-demand ORS round-trip per candidate shelter when the user first requests a route, and the LLM streaming time during the conversation itself.

8 Conclusion and Future Work

We presented BEACON, a personalized multilingual wildfire evacuation agent that combines WatchDuty real-time fire-perimeter polygons, NOAA HRRR wind-aware hazard modeling, polygon-avoidant routing, and a Claude-driven agent that auto-detects the user's language at each turn and degrades gracefully to "call 911" when no safe route exists. The contribution is in the integrated system — the data flow, the danger model, the multilingual prompt strategy, the guardrails, the graceful-degradation protocols, and the deployment infrastructure.

Federal mandates for multilingual emergency alerting do not take effect in the United States until June 2028. Between now and then, every additional wildfire season is a season in which LEP residents are systematically underserved. The technical primitives to close that gap — real-time perimeter data, public polygon-avoidant routing, capable multilingual language models — exist today, and an end-to-end mobile application built on those primitives can be built by a small team in months.

The immediate next steps are: (i) a live shelter availability feed, ideally co-developed with state and county emergency managers; (ii) extending the iOS client to

Android; (iii) a precomputed, expert-reviewed corpus of canonical emergency strings that the agent prefers for high-stakes translations; and (iv) speech-to-text input so that a user can operate the application hands-free while driving.

The authors believe the design patterns described in this paper — language anchored to the latest message, scripted graceful degradation paths, and intent-conditioned tool exposure — transfer beyond wildfire evacuation to any safety-critical conversational application serving linguistically diverse populations.

9 Statements and Declarations

Acknowledgments. The authors thank the Women in Data Science Worldwide (WiDS) Apprenticeship program for providing data, feedback, and mentorship.

Competing Interests. The authors have no relevant financial or non-financial interests to disclose.

Funding Declaration. There was no external or internal funding provided

Clinical Trial and Consent to Participate, Consent to Publish. Not applicable

Ethics. Not applicable. This article does not contain any studies with human participants or animals performed by any of the authors.

Data availability. The datasets used during and/or analyzed during the current study are publicly available or can be re-generated using the methods in the paper (Table 4).

Table 4 Data sources used in this study.

Dataset	Source / Access
Watch Duty fire perimeters	https://www.kaggle.com/competitions/WiDSWorldWide_GlobalDathon26/data (Requires registration).
NOAA HRRR meteorological fields	https://registry.opendata.aws/noaa-hrrr-pds/ and https://github.com/blaylockbk/Herbie
Shelter and drop-by facility locations	https://wiki.openstreetmap.org/wiki/Overpass_API

BEACON draws on a combination of publicly available data, data derived from public sources and user-generated session data that cannot be shared for privacy reasons.

1. **Publicly available data.** The Watch Duty fire-perimeter and evacuation-zone data used for training and evaluation of BEACON's danger-assessment and routing pipelines were obtained from the 2026 Women in Data Science (WiDS) University Datathon dataset, publicly hosted on Kaggle. NOAA High-Resolution Rapid Refresh (HRRR) meteorological fields are publicly available through the NOAA Registry of Open Data on AWS and can also be retrieved through the

open-source Herbie library. Shelter and drop-by facility locations are drawn from OpenStreetMap via the Overpass API. (Table 4)

2. **Derived data.** The Hourly Wildfire Potential (HWP) grid computed from HRRR fields and the classifier used for adaptive refresh scheduling described in the paper, can be reproduced from the publicly available inputs above using the provided equations and algorithms in the paper.
3. **Restricted data.** User GPS coordinates, household onboarding answers, chat-session transcripts, and cross-session semantic memories collected are personal data and cannot be shared. These data are not required to reproduce the methodological results reported in this paper.

References

Beverly JL, Forbes AM. Assessing directional vulnerability to wildfire. Natural Hazards. 2023;117(1):831–849.

Blaylock BK.: Herbie: Download Weather Forecast Model Data in Python; 2026. https://herbie.readthedocs.io.

Cheung BKS, Choy K, Li CL, Shi W, Tang J. Dynamic routing model and solution methods for fleet management with mobile technologies. International Journal of Production Economics. 2008;113(2):694–705.

Cova TJ, Dennison PE, Kim TH, Moritz MA. WUIVAC: a wildland-urban interface evacuation trigger model applied in strategic wildfire scenarios. Natural Hazards. 2007;41(1):181–199.

DiGrande L, Carley-Baxter L, Bevc C, Triplett S, William C, Williams J, et al. Pilot study on the experiences of hurricane shelter evacuees. RTI Press; 2019.

Dowell DC, Alexander CR, James EP, Weygandt SS, Benjamin SG, Manikin GS, et al. The High-Resolution Rapid Refresh (HRRR): An hourly updating convection-allowing forecast model. Part I: Motivation and system description. Weather and Forecasting. 2022;37(8):1371–1395.

Dye AW, Kim JB, McEvoy A, Fang F, Riley KL. Evaluating rural Pacific Northwest towns for wildfire evacuation vulnerability. Natural hazards. 2021;107(1):911–935.

Federal Communications Commission.: Multilingual Wireless Emergency Alerts; 2026. https://www.fcc.gov/multilingual-wireless-emergency-alerts.

Global System Laboratories, National Oceanic & Atmospheric Administration, U S Department of Commerce.: High-Resolution Rapid Refresh (HRRR); 2026. https://rapidrefresh.noaa.gov/hrrr/.

James E, Ahmadov R, Romero-Alvarez J, Grell G, Csiszar I, Anderson LD, et al. An Hourly Wildfire Potential Index for Predicting Subdaily Fire Activity Based on

Rapidly Updating Convection-Allowing Model Forecasts. Weather and Forecasting. 2025;40(9):1805–1822.

Kim J, Atkins E. Airspace geofencing and flight planning for low-altitude, urban, small unmanned aircraft systems. Applied Sciences. 2022;12(2):576.

Mapbox.: Mapbox — Maps, Navigation, Search, and Data; 2026. https://www.mapbox.com.

Medicaid and CHIP Payment and Access Commission.: Enrollment and Access Barriers for People with Limited English Proficiency; 2024. Accessed: 2026-08-17. https://www.macpac.gov/publication/enrollment-and-access-barriers-for-people-with-limited-english-proficiency/.

National Interagency Fire Center.: Wildland Fire Statistics; 2024. Accessed: 2026-05-01. https://www.nifc.gov/fire-information/statistics.

National Oceanic and Atmospheric Administration.: NOAA High-Resolution Rapid Refresh (HRRR) Model - Registry of Open Data on AWS; 2026. https://registry.opendata.aws/noaa-hrrr-pds/.

OpenRouteService.: OpenRouteService; 2026. https://openrouteservice.org.

OpenStreetMap.: Overpass API - OpenStreetMap Wiki; 2026. https://wiki.openstreetmap.org/wiki/Overpass_API.

Qiu M, Li J, Gould CF, Jing R, Kelp M, Childs ML, et al.: Wildfire smoke exposure and mortality burden in the USA under climate change. Nature Publishing Group UK London; 2025.

Ronchi E, Wahlqvist J, Ardinge A, Rohaert A, Gwynne S, Rein G, et al. The verification of wildland–urban interface fire evacuation models. Natural hazards. 2023;117(2):1493–1519.

Shu P, Chen J, Liu Z, Wang H, Wu Z, Zhong T, et al. Transcending language boundaries: Harnessing llms for low-resource language translation. arXiv preprint arXiv:241111295. 2024;.

Synolakis CE, Karagiannis GM. Wildfire risk management in the era of climate change. PNAS nexus. 2024;3(5):pgae151.

Uekusa S, Matthewman S. Preparing multilingual disaster communication for the crises of tomorrow: A conceptual discussion. International journal of disaster risk reduction. 2023;87:103589.

U S Government Accountability Office. Weather Safety: Agencies Face Challenges Related to Multilingual Weather Alerts, and AI Project Needs Better Planning. U.S. Government Accountability Office; 2026.

Watch Duty.: Watch Duty - Wildfire Maps & Alerts; 2026. https://www.watchduty.org.

WatchDuty.: 2025 Annual Report - Watch Duty; 2026. https://www.watchduty.org/blog/2025-annual-report.

Women in Data Science Worldwide.: WiDS University Datathon 2026 — Kaggle; 2026. https://www.kaggle.com/competitions/WiDSWorldWide_GlobalDathon26/data.