

Podcast Search Edge Function Interview Script

Intro (0:00-0:10)

"Our podcast search runs entirely on a Supabase Edge Function, so we can aggregate Apple, PodcastIndex, and Spotify data in one call."

Planner (0:10-0:30)

"Inside `supabase/functions/podcast-search/planner.ts` we normalize the query whether it's an Apple URL, RSS feed, or free text and build a prioritized plan list. Flags determine if we need episode payloads, and we pass along placeholders like `idFrom` that later steps resolve dynamically."

Executor (0:30-0:55)

"`executor.ts` walks those plans in order, resolving dependencies if Apple finds the collection ID, we feed it into PodcastIndex to recover the canonical feed. Every provider call runs behind a timeout wrapper so a flaky API can't stall the Lambda, and we track per-provider timing plus errors in a shared results map."

Resolver (0:55-1:15)

"The resolver merges Apple search hits with PodcastIndex metadata, applies fuzzy scoring on title and publisher, and injects domain sanity checks so we don't return shady feeds. Depending on confidence we auto-select or disambiguate and include alternatives and episode lists when the plan requested them."

Edge Wrapper & Client (1:15-1:30)

"`index.ts` wires everything to Deno's `serve`, adds CORS headers, and builds telemetry calls, confidence, even how many episodes had transcripts. On the frontend, `src/integrations/podcast-search/client.ts` invokes the function via `supabase.functions.invoke`, keeping API keys server-side."

Outro (1:30-1:35)

"It gives us a resilient, serverless aggregation layer that we can extend just by adding new planner rules or provider adapters."