

# Smart Search Pipeline Interview Script

## Opening (0:00-0:10)

"Let me walk you through our Smart Search pipeline, which powers an LLM-guided multi-provider media search."

## Planning Stage (0:10-0:35)

"We start in `src/integrations/smart-search/planning.ts`, where we normalize the user query, detect tokens like `S1E3` or `chapter`, and predict a ranked list of media types. That lets us build a cost-aware set of query plans, so we can bias toward TMDB for TV tokens, fall back to OpenLibrary for bookish hints, and cap the provider fan-out based on the user's cost budget."

## Execution Stage (0:35-0:55)

"Those plans flow into `providers.ts`: each provider adapter wraps a Supabase Edge Function or external API. We fire them with `Promise.race` timeouts so a slow provider can't block the pipeline, and we capture telemetry like latency and API errors for debugging."

## Resolution Stage (0:55-1:10)

"Once we have raw results, we normalize each provider's shape into a shared schema, merge duplicates, and run a fuzzy scoring pass in `resolution.ts`. That score blends title similarity, predicted media type, popularity, and penalties for stale or adult content; thresholds decide whether we auto-select a match or ask the user to disambiguate."

## Post-processing & Testing (1:10-1:25)

"Before returning, we annotate the response with episode or chapter hints, echo the query plans for observability, and emit the telemetry bundle. We have unit tests for token detection and planning heuristics plus integration tests that mock providers to validate the full loop."

## Closing (1:25-1:30)

"The result is a resilient search flow that can expand to new providers just by adding adapters and heuristics."