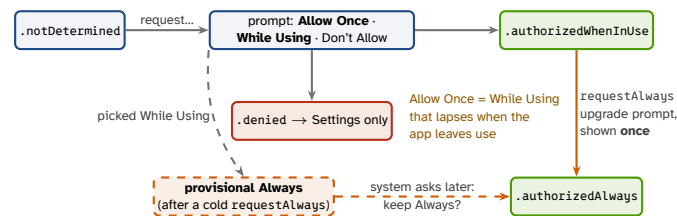


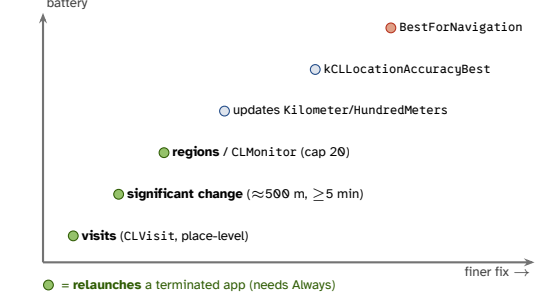
In one line: Core Location fuses GPS/Wi-Fi/cell into fixes under a permission the user can **shrink any time** (While Using/Always and Precise/Approximate); the senior skill is picking the **cheapest service that answers the question**. MapKit draws it: **annotations** = points (reused views), **overlays** = shapes (renderers).

Authorization (iOS 13+)



orthogonal (iOS 14): `accuracyAuthorization` / `fullAccuracy` / `reducedAccuracy` (km-scale)
 → `requestTemporaryFullAccuracy(withPurposeKey:)` — key in `CLLocationTemporaryUsage`.

Which service? — power vs precision



How it works — Core Location

- `CLLocationManager`: strong ref, created on main (callbacks use the creating thread's run loop). `locationManagerDidChangeAuthorization` (iOS 14) fires at creation too — read `authorizationStatus` there.
- Info.plist `NSLocationWhenInUseUsageDescription` (+ `...AlwaysAndWhenInUse...` for Always); missing → no prompt.
- `requestLocation()` = one fix, then stops (needs `didFailWithError`). `startUpdatingLocation()` = stream, tuned by `desiredAccuracy`, `distanceFilter`, `activityType`.
- iOS 17**: `CLLocationUpdate.liveUpdates()` = `AsyncSequence`, no delegate; a held `CLBackgroundActivitySession` keeps it alive in the background. `CLMonitor` (actor) replaces `startMonitoring(for:): add(condition, identifier:)`, then iterate events (`.satisfied/.unsatisfied`).
- Background**: `mode location + allowsBackgroundLocationUpdates = true`, started in the foreground.
- Quality: `horizontalAccuracy < 0` = invalid; first fix often cached (check timestamp). `CLGeocoder`: network, rate-limited.

How it works — MapKit

- Annotation** (`MKAnnotation` point) → `viewFor:` → **reused** `MKAnnotationView` (register + `dequeueReusable`; `MKMarkerAnnotationView`). **Overlay** (`MKPolyline`/`Polygon`/`Circle`/`MKTileOverlay`) → `rendererFor:` → `MKOverlayRenderer`, scales with zoom.
- Clustering** (iOS 11): shared `clusteringIdentifier` → `MKClusterAnnotation`; `displayPriority`.
- Search**: `MKLocalSearchCompleter` (as you type) → `MKLocalSearch` → `[MKMapItem]`. **Directions**: `MKDirections.Request` → `calculate` → `MKRoute` (`polyline`, `expectedTravelTime`); or hand off: `openInMaps(launchOptions:)`.
- SwiftUI Map (iOS 17)**: content builder (`Marker`, `Annotation`, `MapPolyline`, `MapCircle`, `UserAnnotation`), `MapCameraPosition(.automatic, .region, .userLocation(fallback:))`, `.onMapCameraChange`, `.mapControls`. iOS 16 `Map(coordinateRegion:)` is deprecated.

Example — iOS 17 async + SwiftUI Map

```

let bg = CLBackgroundActivitySession() // hold: bg updates
for try await u in CLLocationUpdate.liveUpdates() {
  guard let loc = u.location else { continue }
  track(loc) // + u.isStationary
  let m = await CLMonitor("stores") // actor, 20 max
  await m.add(CLMonitor.CircularGeographicCondition(
    center: shop, radius: 150), identifier: "shop-7")
  for try await e in await m.events {
    if case .satisfied = e.state { arrived(e.identifier) } }
  Map(position: $camera, selection: $picked) {
    ForEach(shops) { s in
      Marker(s.name, coordinate: s.coord).tag(s.id)
    }
    MapPolyline(route.polyline).stroke(.blue, lineWidth: 4)
    UserAnnotation()
  }
}

```

Interview traps

- ! Reading `authorizationStatus` right after `request...` — still `.notDetermined`; the answer comes in the delegate.
- ! No first prompt offers "Always". A cold `requestAlways` gives *provisional* Always; the upgrade prompt shows **once** — after that, and after `Don't Allow`, only Settings.
- ! **Blue pill**: mandatory for `While-Using` background use; *off* by default for Always (opt in via `showsBackgroundLocationIndicator`).
- ! `allowsBackgroundLocationUpdates` without the `location mode` crashes; `pausesLocationUpdatesAutomatically` silently stops a background tracker.
- ! **20 regions** per app: monitor the nearest 19 + one big region round the user whose *exit* re-plans. Geofences lag by minutes; radii under ≈ 100 m are unreliable.
- ! Returning a view for `MKUserLocation` hides the blue dot — return `nil`. `CLGeocoder` per keystroke is throttled.

Remember

Cheapest service that answers it · permission can **shrink** any time · **points** = reused views, **shapes** = renderers.

Likely questions

1. Track a run, screen off? — `While Using` + `location mode`, start in foreground, `.fitness`; blue pill.
2. Remind at 300 stores? — nearest-19 rotation + re-plan region; Always.
3. Log places all day cheaply? — visits or significant change.
4. 10 000 pins? — clustering + reuse, or aggregate server-side.