

# WildTrace Application Architecture: End-to-End Data Flow

UI Thread  
(Android Main Looper - synchronous)

- Composables (SearchView, SightingDetailView, MapView, etc.)
- Collects state from ViewModels
  - searchResults
  - isLoading
  - animalName
  - wikiSummary
  - error
- Handles navigation
  - Map → Search
  - Map → SightingDetail
  - Back stack

BACKGROUND THREADS  
(Kotlin Coroutines / Dispatchers)

```
ViewModelScope.launch { ... }
```

- SearchViewModel
  - performSearch()
  - simulate API delay
  - generate mock search results
- SightingDetailViewModel
  - loadSightingDetails()
  - delay(700) simulate network
  - prepare mock data (animal name, wiki summary, date, location)
- Future real tasks (Final Project):
  - iNaturalist REST API calls
  - Wikipedia summary API
  - Firestore reads/writes
  - Image upload/download

