Conceptual Overview for Developers using SecondBrain

Audience and Purpose

This conceptual overview is written for developers who want to understand the design, system architecture, and data flow behind SecondBrain.

This document explains what SecondBrain is, how it conceptually integrates with an existing mobile operating system, and how its internal components communicate.

What SecondBrain Is

SecondBrain is a system level interface layer that sits on top of an existing mobile operating system and presents itself as the primary way a user interacts with their phone.

A system level interface layer operates above the operating system UI while relying on native OS services for execution, security, and hardware access.

SecondBrain does not replace the operating system. Instead, it:

- Uses system permissions and UI placement
- Collects metadata from approved system APIs and applications
- Translates that data into a unified, distraction free interface

While the underlying mobile operating system continues to:

- Run applications
- Manage memory and processes
- Enforce security and permissions

Core Interface Components

SecondBrain replaces traditional application navigation with three primary interface components:

The Now Page (Home Page) Displays:

Tasks, Reminders, Events, Time sensitive information etc. What appears on the Now Page depends on:

- User created Context Pages
- System data from calendar, notifications, reminders etc.
- Internal prioritization logic

Context Pages:

Act as functional replacements for applications. Instead of opening individual apps, users interact with Context Pages that can represent work, school, goals, etc. Each Context Page:

- Pulls in relevant data from permitted applications
- Removes algorithmic feeds and unnecessary content
- Communicates directly with the Now Page and Daily Digest

Daily Digest

The Daily Digest is a summary generated at the end of the day. It surfaces, emails, messages, posts, system events etc. The Daily Digest is populated using:

- Metadata from granted permissions
- Internal prioritization and relevance scoring

How SecondBrain Overlays the Existing Operating System

SecondBrain operates through:

- Elevated UI visibility
- System permissions
- OS supported services such as notification listeners
- Displays its own pages
- Hides app icons and algorithmic feeds from the user interface
- Observes changes in system state

Meanwhile, the operating system:

- Continues to execute apps
- Handles memory and background processes

- Manages security and permissions

Data Statement

SecondBrain relies on system APIs that the user explicitly grants access to, such as calendar, notifications, contacts, reminders etc. This data is then processed:

- Converted into a shared internal schema
- Stripped to remove redundancy
- Standardize into fields such as:
    - Type
    - Priority
    - Title
    - Start and end time
    - Associated context

How SecondBrain interacts/communicates with Applications

SecondBrain interacts/communicates with applications in one of two ways:

Applications with Public APIs

For apps that provide public APIs like email services:

- SecondBrain reads metadata
- User actions are translated into structured API requests
    - Example: If a user deletes emails from the Daily Digest page, SecondBrain formats that action and sends it to the email provider's API which updates within the internal application.

Applications Without Public APIs

For apps without public APIs like social media platforms:

- SecondBrain stores user generated content locally as drafts
- Uses deep linking to send the user directly to the intended action screen
- Avoids exposing the user to algorithmic feeds

- Example: If a user wants to create a social media post, they can draft the content within SecondBrain and when the user chooses to publish it, SecondBrain opens the social media application directly to the post creation screen and automatically pre fills the content using the saved draft. The user can then review and submit the post within the application without interacting with the platform’s feed.

Tools & Environment

- Relies on native OS APIs such as IOS or Android
- Uses permission frameworks for calendar, notifications, contacts etc.
- Integrates with third party services via public APIs when available
- Does not require modification of third party applications