

ROTOM Corporation

Architecture Design Document

For

Plotline

Version 1.0
October 20, 2025

Authors
Patrick Hwang, CEO
Julian Cabello
Julia Chaidez
Julianna Arzola
Emily Panfilo

Purpose

The purpose of this architecture diagram is to provide a clear, visual blueprint of how the Plotline system in structure and its components interact. It maps frontend, backend, and infrastructure elements, showing where each core feature lives and how data flows between them using REST, realtime SSE, and asynchronous queues. By outlining services, connections, and communication patterns, this diagram becomes the single source of truth for the team, helping us align on scope, divide responsibilities, plan development tasks, and ensure the system remains scalable, maintainable and consistent as we continue building the platform.

Diagram

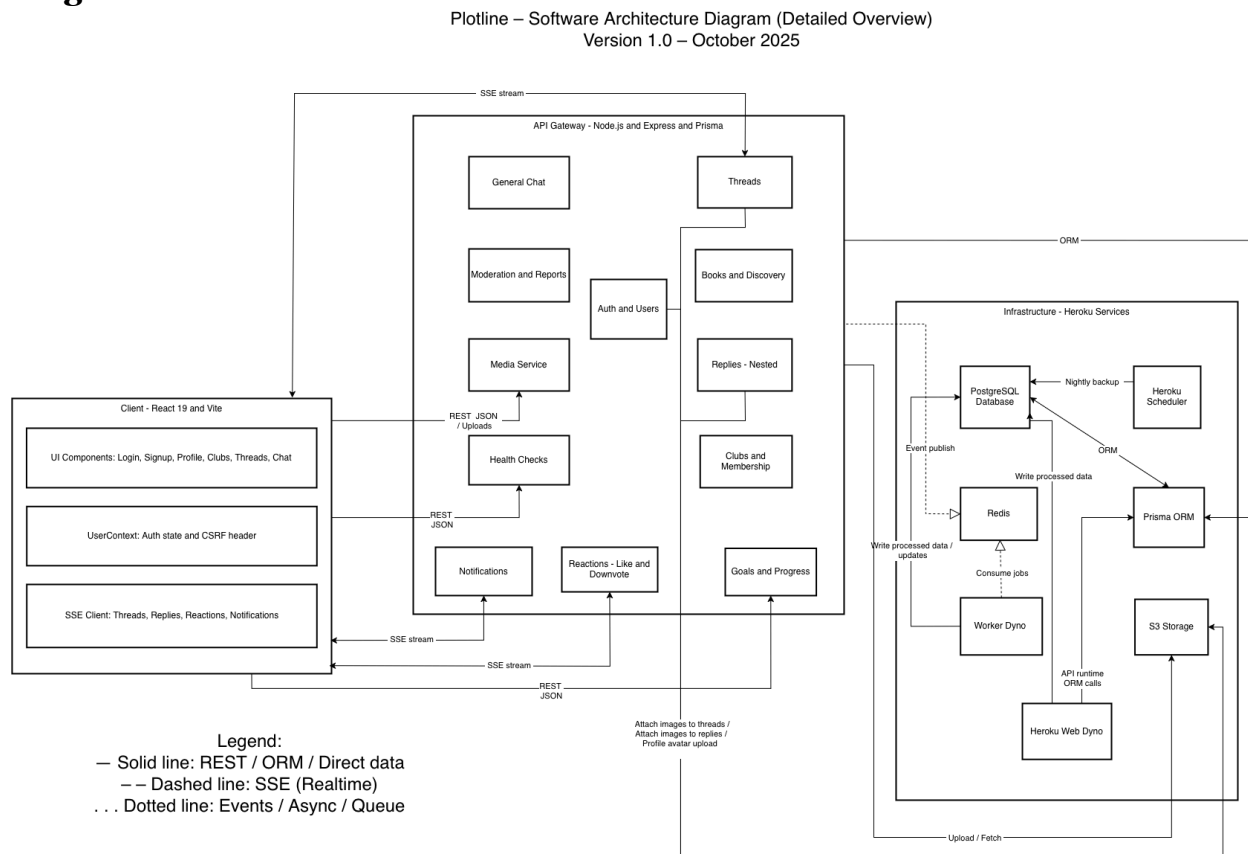


Fig. 1 The Architecture design diagram for Plotline

Summary

The current design for Plotline consists of six primary modules, supported by two affiliated classes, as shown in Fig. 1. The modules include:

- **User Information**
- **Club Management**
- **User Generated Content**
- **Reading & Progress Tracking**
- **Private User Content**
- **User Interface**

User Information, located at the center of the diagram, contains core user profile information and authentication data.

Club Management, located to the upper left of the diagram, manages club membership, member roles, and book selections.

User Generated Content, located to the left of the diagram, contains all public communication created by users, including chat messages, threads, nested replies, reactions and associated media.

Reading & Progress Tracking, located to the upper left, stores book catalog information, goal assignments, and user reading progress.

Private User Content, located at the top, stores notifications, and other user specific private information.

User Interface, located at the bottom of the diagram, contains all major components of plotline's web interface and outlines valid navigation pathways between pages.

Module Interactions

- **User Registration/ Login:** The User Interface module(Login, Sign up) interacts with the **User Information** module (Users, Sessions) to create or verify user accounts.
- **Club Participation and Discussion:** Users create chat messages, threads, replies, and reactions in the User Generated Content module, which are then displayed via the User Interface (Clubs Page, Threads Page, Chat Page).
- **Notifications:** When actions such as mentions, reactions, or replies occur, the **Private User Content** module generates notifications, which are then retrieved and displayed via the User Interface (Notification Page).

Modules

User Information

The user information module contains the following components:

- **Users**
- **Sessions**

The **Users** component contains information specific to individual user accounts, including display names, email addresses, avatars, and bios. It provides an interface to all other modules, including **Club Management**, **User Generated Content**, **Reading & Progress Tracking** and **Private User Content**. The user's component does not require another component to function.

The **Sessions** component manages user authentication and session tracking. It provides an interface to the **User Interface** module, specifically the Login and Sign up

components, and requires the **Users** component in order to store **Session** identifiers for authenticated **Users**.

Club Management

The club management module contains the following components:

- **BookClub**
- **UserBookClubMember**
- **BookClubBook**

The **BookClub** component stores information about individual clubs, including their description and hosts, The **UserBookClubMember** component manages membership and role based permissions for each club, The **BookClubBook** component tracks the books associated with each club, The module requires the **Users** and **Book** components to function and provides an interface to both the **User generated Content** and **Reading & Progress Tracking** modules.

User Generated Content

The user generated content module contains the following components:

- **General Chat**
- **Threads**
- **Thread Replies**
- **Reactions**
- **Media**

The **General Chat** component stores information about real time club level messages.

The **Threads** component, located at the top of this module, contains primary

discussion posts created within clubs. The **Thread Replies** component, located to the left of the module, stores nested replies associated with individual thread posts. The **Reactions** component stores likes and downvotes on posts and replies, providing an interface to the **Private User Content** module. The **Media** component stores images and attachments associated with threads and **cat** messages. All the components in this module require the **Users** and **BookClub** components to function.

Reading & Progress Tracking

The reading and progress tracking modules contains the following components:

- **Book**
- **UserBook**
- **BookClubGoals**
- **UserBookClubBookProgress**

The **Book** component stores catalog information. The **UserBook** component stores **Users** personal reading history. The **BookClubGoals** component stores shared reading milestones for each club, and the **UserBookClubBookProgress** component stores individual progress logs. These components require the **Users** and **BookClub** modules to function and provide interfaces to the **User Interface** module, particularly the Book Page, Clubs Page, and Profile Page.

Private User Content

The private user content module contains the following component;

- **Notifications**

The **Notifications** component stores event driven alerts, such as reaction, replies, role changes, or goal reminders. It provides an interface to the **User Interface** module, specifically the Notification Page, and requires the **Users**, **Threads**, and **Replies**, and **Reactions** component to function.

User Interface

The user interface module contains the following components:

- **Landing Page**
- **Sign Up**
- **Login**
- **Home**
- **Notification Page**
- **Chat Page**
- **Threads Page**
- **Clubs Page**
- **Book Page**
- **Profile Page**

The **Landing Page**, located to the right of the module, is the page that **Users** begin on when they first access the application. It links to the **Sign up**, **Login** and **Home** components within the same module, it provides an interface to the **Browser** component. It does not require any other components to function. The **Sign Up Page**, located at the top center right of the module, is the component that **Users** use to register a new account. It links to the **Login** and **Landing Page** components within the same module. It does not provide an interface to any other component. It requires the

Users and **Sessions** components to function. The **Login Page**, located at the top center left of the module, is the component that users use to log in to an existing account. It links to the **Sign up**, **Landing Page**, and **Home** components within the same module. It does not provide an interface to any other component. It requires the **Users** and **Sessions** components to function. The **Profile Page**, located at the top center left of the module, is the component that users use to view and edit their profile and activity. It links to the **Home** component within the same module. It does not provide an interface to any other component. It requires the **Users** and **User Generated Content** components to function. The **Clubs Page**, located to the left of the module, is the component that users use to view the clubs they belong to and the discussions within them. It links to the **Home** component within the same module. It does not provide an interface to any other component. It requires the **BookClub** and **UserBookClubMember** components to function. The **Threads Page**, located at the left center of the module, is the component that users use to view and participate in structured club discussion threads. It links to the **Clubs Page** within the same module. It does not provide an interface to any other component. It requires the **Threads** and **Thread Replies** components to function. It also receives realtime updates through an SSE stream, allowing new posts and replies to appear immediately on the page. The **Chat Page**, located at the bottom center of the module, is the component that users use to participate in live club conversations. It links to the **Clubs Page** and **Home** components within the same module. It does not provide an interface to any other component. It requires the **General Chat** and **Message** components to function. It also receives real-time updates through an SSE stream, allowing new chat messages to appear instantly without refreshing. The **Book Page**, located at the bottom left of the

module, is the component that users use to view book information, reading goals, and their current progress. It links to the **Clubs Page** and **Home** components within the same module. It does not provide an interface to any other components. It requires the **Book**, **BookClubGoals**, and **UserBookClubBookProgress** components to function. The **Notification Page**, located at the top right of the module, is the component that users use to view alerts about reactions, replies, mentions, goal updates, and club activity. It links to the **Home** component within the same modules. It does not provide an interface to any other component. It requires the **Notifications** component to function. It also receives real-time alerts through an SSE stream, allowing users to see notifications the moment new activity occurs.

Unaffiliated Classes

There are two components unaffiliated with any module, as listed below:

- **Media Storage**
- **Search/ Discovery Index**

Media Storage, located near the bottom left of the application, contains uploaded files such as images used for chat messages, thread posts, and profile avatars. It provides an interface to the **User Generated Content** and **User Interface** modules, specifically with the **Media**, **Chat Page**, and **Threads Page** components. It does not require any other components to function. **Search/ Discovery Index**, located near the bottom left of the application, contains searchable data for books, clubs, and discussion content. It provides an interface to the User Interface module, specifically with the **Book Page** and **Clubs Page** components. It does not require any other components to function.

External Services

There are five external components that communicates with Plotline, as listed below:

- **Browser**
- **PostgreSQL Database**
- **Redis**
- **S3 Storage**
- **Heroku Scheduler**

Browser, located at the bottom right of the application, references the client browser used to access Plotline. It does not provide an interface to any other component. It requires access to the User Interface module in order to load and render the application.

PostgreSQL Database, located to the right of the application, references the relational data store used to persist Plotline's core records. It provides an interface to the API Gateway through Prisma ORM. It does not require any other components to function.

Redis, located to the right of the application, references the in-memory storage and event streaming storage used to support real time behaviour and asynchronous processing. It provides an interface to the **Notifications**, **Chat**,

Threads, and **Worder Dyno** components. It does not require any other components to function.

S3 Storage, located underneath the application, references the outside service used to store uploaded media files such as images. It provides an interface to the

User Interface module, specifically with media upload components. It does not

require any other components to function. **Heroku Scheduler** located to the upper right of the application, references the outside service used to run scheduled

maintenance tasks, such as backups and cleanup operations. It provides an interface to

the **PostgreSQL Database** and **Worker Dyno**. It does not require any other components to function.