Project Motivation

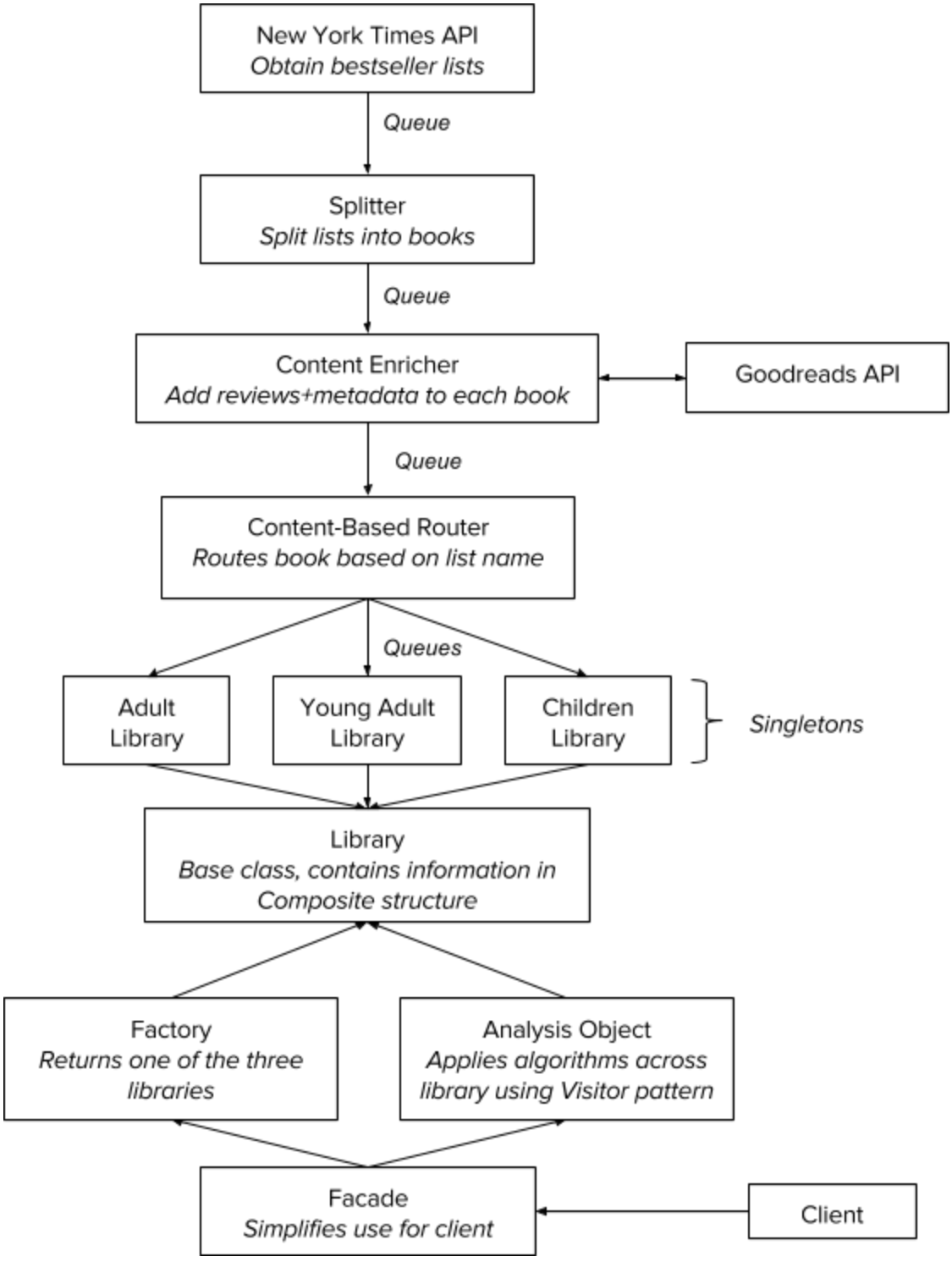
The goal of this system is to gather a rich set of information about bestselling books over time, thereby allowing various parties—such as libraries, book publishers, and academic researchers—to analyze reading trends for their own purposes. For example, a book publisher may want to extract the most popular keywords for adult fiction bestsellers to shape its acquisition decisions, or a sociologist may want to study the correlation between authors' social backgrounds and literary success.

Project Concept

My project will pull the various bestseller lists from the New York Times Books API when they are updated once a week, and convert each bestseller list to a MESSAGE. This MESSAGE gets sent to a SPLITTER where it is split into individual messages; each message corresponds to a single book and maintains a reference to the original list that it was derived from (for the purposes of the content-based router later on—see below). Then, each message gets sent through a QUEUE to a CONTENT ENRICHER that runs the ISBN of the book through the Goodreads API to acquire additional information about each book such as a collection of its most popular reviews, as well as richer metadata (e.g. number of pages, gender and country of author, average rating, etc).

After passing through the CONTENT ENRICHER, each message gets sent to a CONTENT-BASED ROUTER via another QUEUE, where it gets routed to one of three possible Library subclasses based on the name of the list from which the book was derived. E.g. a book that appeared on the list "Children's Chapter Books" gets routed to ChildrenLibrary, "Hardcover Fiction" gets routed to AdultLibrary, "Young Adult E-Book" gets routed to YoungAdultLibrary, etc. Each Library subclass will be implemented as a SINGLETON. The information about each book will be stored in a directory structure using the COMPOSITE pattern.

Because the three Library objects share the same superclass, we can use a FACTORY to determine which Library we want to use at run-time (since we expect users of the system to be interested in a specific age range for the books they want to analyze). Then, using the VISITOR pattern, we can apply various analysis algorithms across a library depending on the kind of information that we are interested in. The client will interact with the library through a FACADE to make the system easier to use.


New York Times API
Obtain bestseller lists
Queue
Splitter
Split lists into books
Queue
Content Enricher
Add reviews+metadata to each book
Goodreads API
Queue
Content-Based Router
Routes book based on list name
Queues
Adult Library
Young Adult Library
Children Library
Singletons
Library
Base class, contains information in Composite structure
Factory
Returns one of the three libraries
Analysis Object
Applies algorithms across library using Visitor pattern
Facade
Simplifies use for client
Client