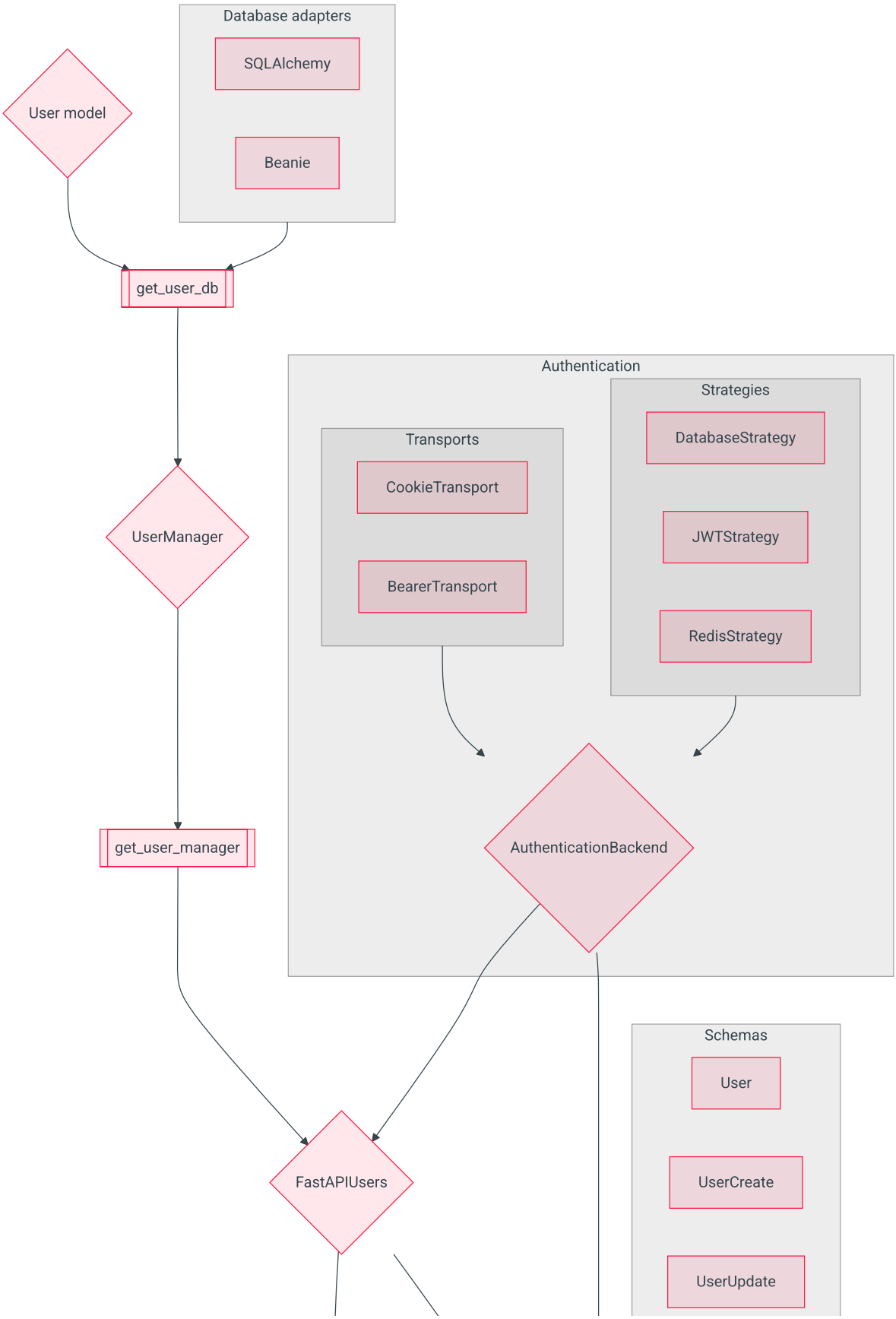
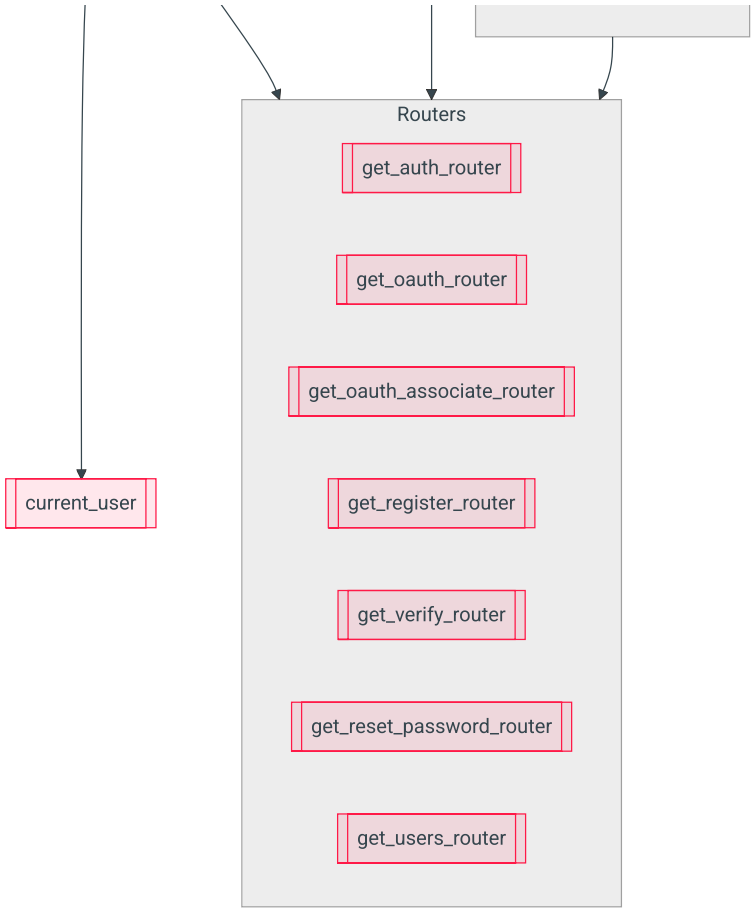
Overview

The schema below shows you how the library is structured and how each part fit together.


Database adapters
SQLAlchemy
Beanie
User model
get_user_db
Authentication
Strategies
DatabaseStrategy
Transports
CookieTransport
JWTStrategy
BearerTransport
RedisStrategy
UserManager
AuthenticationBackend
get_user_manager
Schemas
User
FastAPIUsers
UserCreate
UserUpdate



User model and database adapters

FastAPI Users is compatible with various **databases and ORM**. To build the interface between those database tools and the library, we provide database adapters classes that you need to instantiate and configure.

➡️ I'm using SQLAlchemy

➡️ I'm using Beanie

Authentication backends

Authentication backends define the way users sessions are managed in your app, like access tokens or cookies.

They are composed of two parts: a **transport**, which is how the token will be carried over the requests (e.g. cookies, headers...) and a **strategy**, which is how the token will be generated and secured (e.g. a JWT, a token in database...).

➡️ Configure the authentication backends

`UserManager`

The `UserManager` object bears most of the logic of FastAPI Users: registration, verification, password reset... We provide a `BaseUserManager` with this common logic; which you should overload to define how to validate passwords or handle events.

This `UserManager` object should be provided through a FastAPI dependency, `get_user_manager`.

➡️ Configure `UserManager`

Schemas

FastAPI is heavily using Pydantic models to validate request payloads and serialize responses. **FastAPI Users** is no exception and will expect you to provide Pydantic schemas representing a user when it's read, created and updated.

➡️ Configure schemas

`FastAPIUsers` and routers

Finally, `FastAPIUsers` object is the main class from which you'll be able to generate routers for classic routes like registration or login, but also get the `current_user` dependency factory to inject the authenticated user in your own routes.

➡️ Configure `FastAPIUsers` and routers