Camera System HLD:

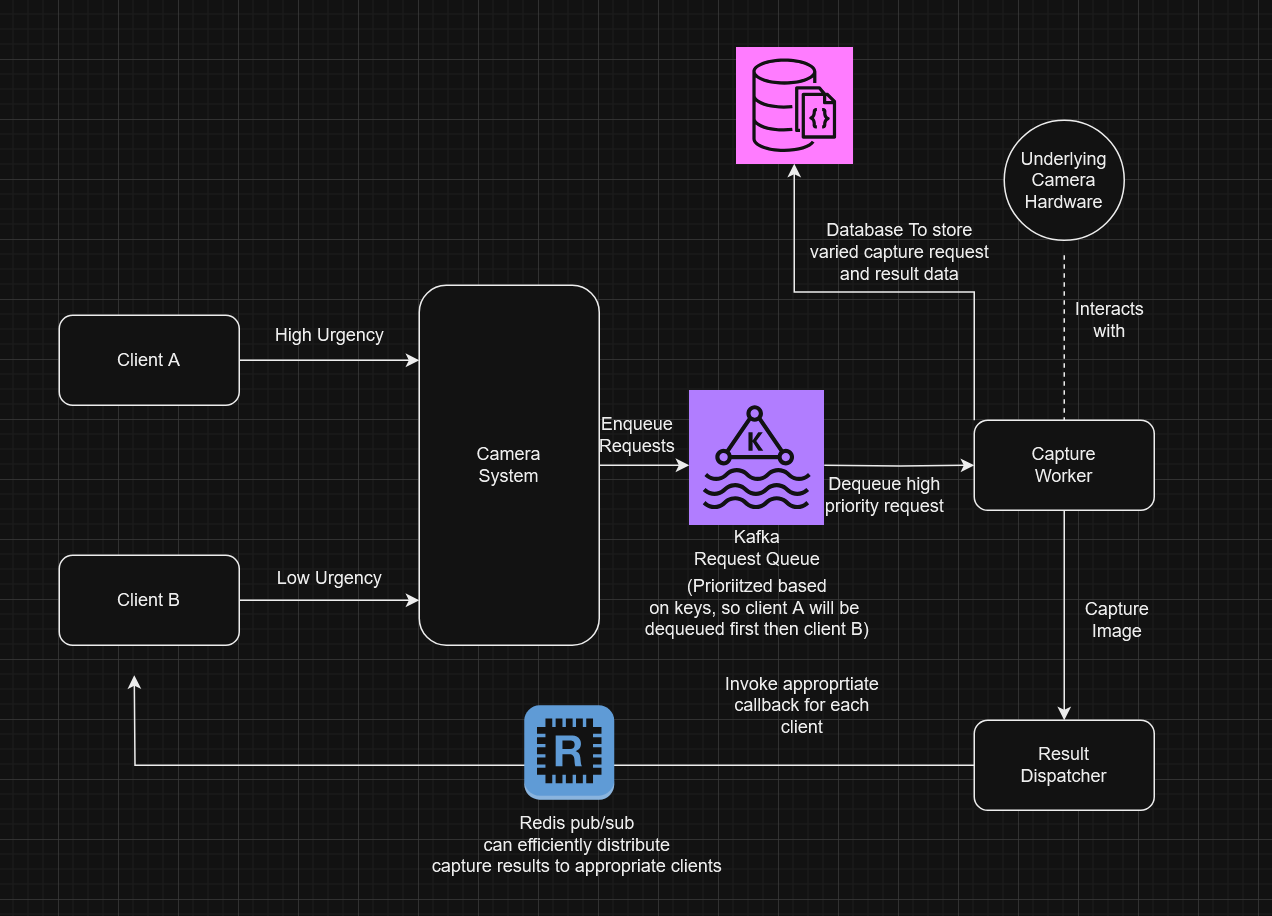
Components:

1. CameraSystem: The main component that handles capture requests and manages the interaction with the underlying camera hardware.
2. RequestQueue: A priority queue that stores and prioritizes capture requests based on their urgency (as defined in self-driving car example)
3. CaptureWorker: A worker component that processes capture requests from the queue and interacts with the camera hardware.
4. ResultDispatcher: Handles the distribution of capture results back to the clients.
5. Client: Represents the various clients that can submit capture requests to the CameraSystem.

Responsibilities:

1. CameraSystem:
   1. Expose RESTful API endpoints for clients to submit capture requests
   2. Accepts capture requests from client.
   3. Manages the request queue.
   4. Coordinates CaptureWorkers.
   5. Handles client registration for callbacks.
2. RequestQueue:
   1. Stores capture requests.
   2. Prioritizes requests based on urgency.
   3. Provides thread safe access to requests.
   4. Can be a kafka based streaming service where keys can be use to prioritize requests.
3. CaptureWorker:
   1. Retrieves request from request queue.
   2. Interacts with camera hardware to capture images.
   3. Passes results to the ResultDispatcher.
4. ResultDispatcher:
   1. Receives capture results from CaptureWorker.
   2. Invokes appropriate callbacks (success or failure) for each client.
5. Client:
   1. Submits capture requests to CameraSystem.
   2. Provides success and failure callbacks.
   3. Specifies urgency for each request.

Design Diagram:



This provides a robust, scalable, and extensible architecture for the camera system. It separates concerns effectively, allowing each component to be developed, tested, and scaled independently.