A Card is the smallest data unit that represents information about an entity. These cards can be saved into decks that are owned by users. Thus, A deck is a collection of cards.

```
// deck list endpoint, lists all decks for a user with minimal metadata
// paginated using pageToken params
// pageToken retrieves specific page of results
// GET users/{username}/decks
{
  "decks": [
    {
      "id": string,
      "desc": string,
    },
    ...
  ],
  "nextPageToken": string,
  "resultSizeEstimate": integer
}

// deck endpoint, gets the specified deck
// GET /decks/{id}
{
  "id": string,
  "desc": string,
  "cards": [
    {
      "id": string,
      "title": string,
      "payload": {
        ...
      }
    },
    ...
  ]
}
```

Implement a decks detail endpoint that uses the above two endpoints and prepares a combined response with deck detailed info resolved. You can use language of your choice.

- This endpoint should implement pageToken param as supported via /users/{username}/decks endpoint.
- How would you implement this endpoint without significantly increasing the response time?
- How would you handle timeouts from //decks/{id} endpoint?

```
//decks combined response would look like
{
  "decks": [
   {
     "id": string,
     "desc": string,
     "cards": [
      {
        "id": string,
        "title": string,
        "payload": {
          ...
        }
      },
      ...
    ]
  }
  ...
  ],
  "nextPageToken": string,
  "resultSizeEstimate": integer
}
```