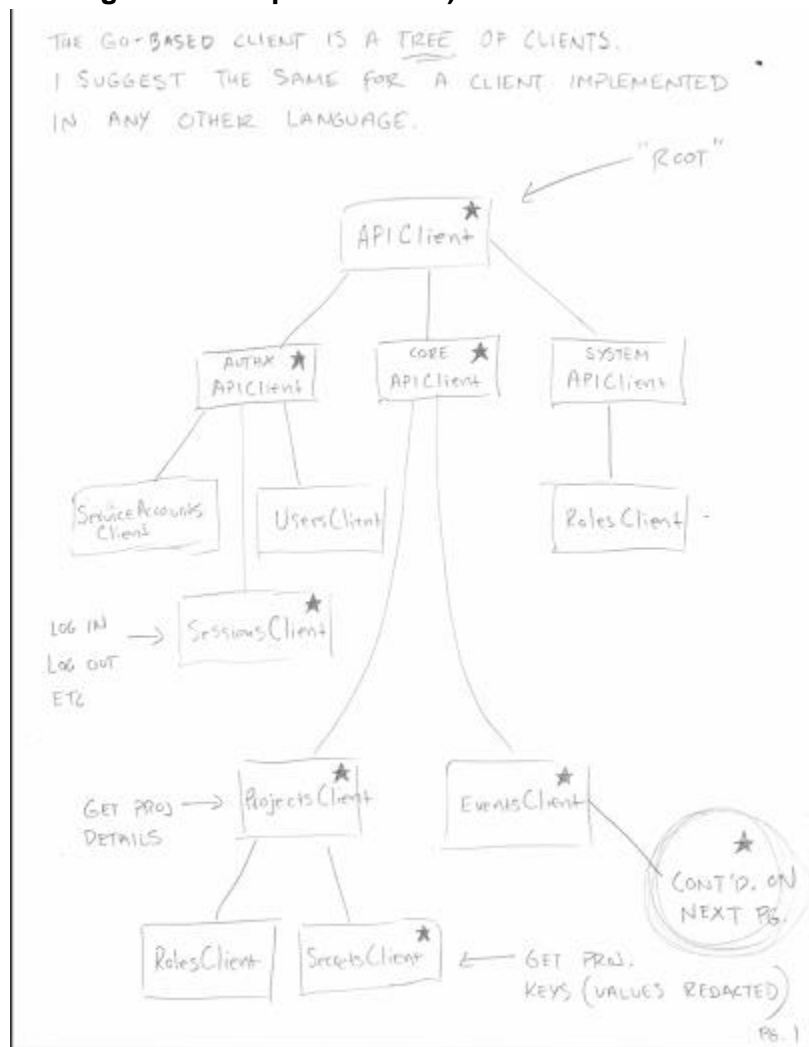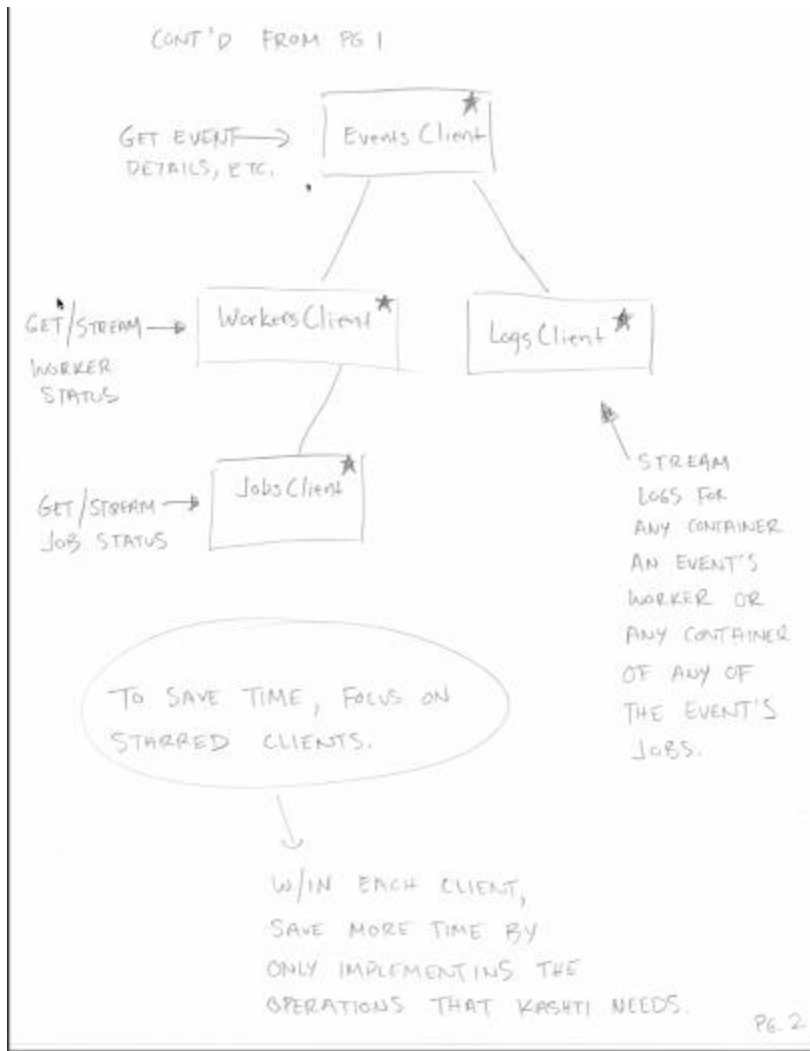- Files with "test" in the name something we want to keep moving forward?
    - Any important features in files with "_test" in the name?
- How will the "internal/restmachinery" directory fit in to the typescript SDK? Will they need to be converted or will they be separate from it?
- **Answered:**
    - Understanding how the colons work (EX: Lines 40-42 of client.go)
    - internal/restmachinery/client.go 131-134: Understanding this function declaration and its structure
    - (For the first question): Test files will be really important to make sure that functionality is working properly. Though the V 1.0 files may not transfer over, we should consider having new ones in V 2.0

- **(Notes on diagram visual presentation):**

-
- Authx is authentication (authx) and authorization (authc)
    - Sessions client (starred items are the most important!)
    - ***UNSTARRED ITEMS DO NOT NEED TO BE PORTED OVER (FOR NOW)

CONT'D FROM PG 1

GET EVENT → | Events Client |
DETAILS, ETC.

GET/STREAM → | Workers Client |
WORKER
STATUS

GET/STREAM → | Jobs Client |
JOB STATUS

| Logs Client |

STREAM
LOGS FOR
ANY CONTAINER
AN EVENT'S
WORKER OR
ANY CONTAINER
OF ANY OF
THE EVENT'S
JOBS.

TO SAVE TIME, FOCUS ON
STARRED CLIENTS.

W/IN EACH CLIENT,
SAVE MORE TIME BY
ONLY IMPLEMENTING THE
OPERATIONS THAT KASHTI NEEDS.

PG. 2

- Again, follow the starred sections for the most part

(Understanding the most relevant things without having to understand go):

```
248
249   func (e *eventsClient) Get(
250       ctx context.Context,
251       id string,
252   ) (Event, error) {
253       event := Event{}
254       return event, e.ExecuteRequest(
255           ctx,
256           rm.OutboundRequest{
257               Method:      http.MethodGet,
258               Path:        fmt.Sprintf("v2/events/%s", id),
259               AuthHeaders: e.BearerTokenAuthHeaders(),
260               SuccessCode: http.StatusOK,
261               RespObj:     &event,
262           },
263       )
264   }
265
```

-
- This is in the API client.go