## What role types do we need, with permissions?

* System Administrator
  + manage AWS services, start/stop services, reboot the system and initialize the Client(s)
  + Needs IAM access to the services AWS account
  + Just have wild card for both and full iam for resources associated with the system
* Client Account Administrator
  + Client Account Administrator will register IoT devices, create content, approve submitted content, publish content, and add section administrators & other Client Account Administrators
  + Add a tag for root ( a boolean) if they’re tagged as root they arent deletable)
  + A client account administrator who is the root “owner” of the Client Account; the first use for that account. There is only one. Can be transferred.
  + Create user add them to client acc group
    - Populate the client id but for section its a wild card
    - Essentially view all sections with associated client id
* Section Administrator
  + Section Administrator manages a subset of devices in a section within a Client, but does not have access to other sections in that Client
  + Create a tag for the client and section on the user just lambda access
  + Create user add them to section admin group
    - Populate the client id and section id fields with the specific section and client
* ~~End User~~
  + ~~End User will be able to view content and request content to be published~~
  + ~~End users dont need accounts~~

### Action

We will create a group for each of these user types on the cognito with section and client id tags.

Section \* wild card

Client and section cards

## How will we restrict access on the api?

Using Groups to Control Permission with Amazon API Gateway

You can use groups in a user pool to control permission with Amazon API Gateway. The groups that a user is a member of are included in the ID token provided by a user pool when a user signs in. You can submit those ID tokens with requests to Amazon API Gateway, use a custom authorizer Lambda function to verify the token, and then inspect which groups a user belongs to. See this blog post for an example of using user pool tokens with an Amazon API Gateway custom authorizer.

If it’s a valid ID Token for a user of your User Pool, you can then access all the claims of ID Token in your API using ‘$context.authorizer.claims’. For example ‘$context.authorizer.claims.email’ will return user’s email address and ‘$context.authorizer.claims.sub’ will return you user’s unique identifier. If the ID token is expired or is invalid, Cognito User Pool Authorizer will send Unauthorized (401) response to the caller.

Use claims to get client id and section id

Register users with client id and section id

## How will we create accounts in the system?

All accounts will be done by invite. We will create an admin page in the application to register users, when a user is created, they will be created with a client id and section id. The user will be created by using a form which send the info to cognito either via the sdk or via lambda.

## ACTION ITEMS

Needed changes

* Register user (maybe just cognito sdk)
* Get user cred ( maybe by token)
* Configure pre-existing lambdas to use the tags