Ushahidi Pointers:

- Problem?:

\* Given that different organizations run of the same Ushahidi platform, how can a particular data set that is shared by all organizations be verified, validated, visualized, and managed given the disjointedness of separate geographic regions pertaining to individual organizations?

- Solution = Georole:

-a geometric extension of a generic role in which the user is constrained to operate on those reports that are included in the geometric property defined for that, so as exploiting a spatial constraint of inclusion. It is more likely that it is always a polygon that can be declared simply from a geometric element of a shapefile or a GeoJSON. This solution also generalizes the case where the geometry is not exactly a nation but may represent a geographic shape of a generic type.

\*Georole is a way to break up the different uses each organization has with the platforms data set into georoles, thereby separating the access to information and tasks that a particular organization carries out to be viewed by only that organization. (allows inclusion of data and processes for each separate organization that uses the same, but different parts of a data set).

\* a Georole allows separate organizations to better verify, validate and coordinate data from within their region that is added to the data set, as each organization is simply responsible for their data set and not all data as a whole (instead of having a SuperAdmin and a group of admins try and organize reports from the whole set).

\* Georoles also allow easier management of reports coming in from added regions, as separate admins are responsible for each region (simply a SuperAdmin would just have to assign a admin to be responsible for the new region, instead of giving permissions to all other admins to be able to work with the new workload added from this region, cuts down on work).

- The roles within the Ushahidi API are hierarchical:

-SuperAdmin - Access to everything

-Admin - Access to almost everything

-Member - Regular user (only access to UI)

\* Given these already defined roles, a hierarchical system can be implemented to give each administrator access to data from a data that only pertain to the region in which they oversea (ie their georole). Each different administrator would be responsible for the reports and information verification/validation for their region, as well as the users who operate within their region (ie each users georole will be defined by their administrator). This allows the division of the workload to be more efficiently processed individually as opposed to having all the admins responsible for all the data in the data set. As well, adding and managing users is more efficient if separate as opposed to having all admins deal with all users. All the admins georoles are managed and defined by a SuperAdmin, who has access and permissions all data for that API.

-The Ushahidi API already has a way to specify permission given a user (only admin can set up the persmisisons) under DashBoard->Users->Add/Manage Role and Permissions.

Requirements:

- User requirements:

- The SuperAdmin will be able to separate and define what data is available to each administrator based on their perceived geographical region, and set their georoles appropriately.

- The SuperAdmin will be able to add, delete, and modify the roles and access to data that each administrator has.

- Administrators of different organizations will be able to define the "georoles" of specific users, and the amount of access they have to reports in their given georoles.

- Members will be able to interact (view, submit, comment on, validate... etc) with reports only in their given georole region.

\* Administrators and Members under each administrator will not be able to interact (view, submit, modify... etc) with report data from georoles outside their own, even if the data is from the same data set.

-Functional Requirements:

The system Georole based system supports the:

- management/permissions of defined Georoles in terms of their modification, (expansion/shrinking of georole region, addition of roles to added administrators, deletion of roles, etc..)

\* NOTE: management done in a hierarchical fashion where the SuperAdmin can change the georole of any administrator or member of any region, administrators can change the role of members within their own region, members have no permission to change their or other members roles

- graphical depiction of Users Georole on the implemented map system

\* Users will be able to so their defined georole and may only interact with reports within its bounds

\* Georole system must coincide with already implemented map system

-Non-Functional Requirements:

- Visualization of physical Georole on the map implementation shouldn't affect the clarity of the visualizations for the report data (should be a lined border, transparent, etc..)

- Management of Georoles (regardless of SuperAdmin or Admin) should be straightforward and be implemented generally through a simple graphical form with fields to fill out or something to that effect.

\*frequent tasks such as georole management should be optimized and the easiest and least time consuming to carry out

- Business Requirements:

\*\*\* Due to the disjointedness of separate geographical regions and the different organizations that specifically work with data (mostly exclusively) specific to each one, and who share the same Ushahidi platform, the verification, validation, and visualization of the data set requires a great deal of work and management by all organizations. To simplify this, the distinction and overseeing of data pertaining to a specific region can be given to those organization(s) that use that data only, thus dividing the work more evenly across organizations and reducing the amount of the data management of the whole data set on all the users, effectively reducing the costs for each organization to maintain the system.

\* Essential reason to implement Georole Interface

Risks:

- Business Risk -- Are we building the right thing?

- Social Risk -- Can our team implement this given the constraints imposed?

- Technical Risks -- Implementation constrained by available technology and resources

(ex will it work on a given platform, will it scale)

- Cost/Time Limits -- Can stories and sprint goals be completed given the resources and time limit imposed on the project