Expectations for management of user permissions

Syria

Proposal A. Using the AWS infrastructure to set up a new PostgreSQL database that will hold a copycat version of the EAMENA DB that can be accessed via QGIS. This copy is what we are already using for Syria. Currently this DB is served from France using a free service called AlwaysData. This option requires the migration of the AlwaysData DB to the AWS server. This process should be relatively straightforward.

Advantages:

1. No need for new Arches set up
2. New data will be available for Durham Staff immediately.
3. Main EAMENA database is not modified nor touched whatsoever
4. Data available in QGIS
5. Upgrades not needed in the long term
6. The use of schemas within the same database solves effectively the problem with the different privacy issues with our Syrian partners and is the easiest way to create memberships.
7. Syrian partners do not work with Arches (yes, this is an advantage)

Disadvantages:

1. It could be slow, or it could throw errors in QGIS if connection is unstable
2. No direct connection to Arches. For future implementations/migration/appending with the main EAMENA DB, data needs to be extracted first, and then Bulk uploaded or Appended in Arches.
3. Each group works separately and feed different databases. The migration to a potential national database could require a little bit more effort than just extracting and bulk uploading.
4. Syrian partners do not work with Arches (yes, this is also a disadvantage)

WD: I think this is the best idea and allows us to sidestep ringfencing altogether by hosting EAMENA compatible data that is sensitive and giving access to particular tables or schemas to whoever we like. Similar 'databases' could be used for other sensitive datasets such as Gaza or Nagorno Karabakh. I wonder if we could pay Reuben (instead of ringfencing) to develop a way of exporting heritage places from the main instance into an even better postgres 'copycat' relational database hosted on the fast EAMENA AWS but kept in a separate PG database for security reasons. This would allow us to query the EAMENA database in a much more useful way than is possible via the frontend, and would provide a schema we could reuse in separate postgres databases for sensitive data such as Syria.

Proposal B. Forking EAMENA DB within same server and create a new Arches Syrian DB within the AWS server. In this case, we could fork the EAMENA DB as it is, filtering out everything but the Syrian data. Within this new instance of Arches group memberships are set and every Syrian "region"/group will have access to its own data. Here they will feed the Syrian DB independently and separately based on the the already uploaded data, potentially avoiding data duplication.

- This will create a new domain (e.g. syria.eamena.org)
- This could improve performance without paying more money and could reduce redundancy and complexity
- The Syrian database will contain data of already known sites and therefore data duplication might be avoided.

Advantages:

1. The DB and webpage runs within the same server of the EAMENA main DB without interfering with it.
2. The migration process to a potential national integrated database would be smoother.
3. Each group will feed the same database but can work separately.
4. Its deployment does not interfere with the EAMENA DB.
5. Membership rules are set up from the beginning but can be easily overrun if needed without worrying about data integrity of the main EAMENA DB or privacy issues.
6. Syrian subdomains should be easy to set up. Technically, this should not be an issue for the AWS server unless each subdomain is accessed more than the main EAMENA DB (e.g. more than 1000 users interactions per day with transactions of more than 5MB).

Disadvantages:

1. Requires a new Arches set up
2. Could require further updates to comply with software updates
3. Membership rules should apply usage rules as well (what one group can see, and what not) If this is a burden, in practice three arches DBs should be deployed. This problem is the same problem with group membership cards and usage, and/or access to information in the first proposal

WD: Even if technically possible, I do not think running an extra database (let alone three!) on the main EAMENA instance is a good idea – I am fairly sure that the AWS instance simply doesn't have the resources needed, and we would not want to compromise an already slow database.

Proposal C. User membership, cards and level access (ring fencing)

Ideally, we would like to have two different levels of access. The highest level of access will allow the EAMENA Staff team access to all the information produced by our Syrian partners. In this level we should be able to access, monitor, edit, delete, extract, manage the information. This data should not be accessible by anyone but the EAMENA Staff and the Syrian partners in their respective groups. This means that not only information cards are hidden as user level, but even locations should be hidden to users out from this group.

Now, the second level of access should be divided in three groups. The information generated from each group should be available to the specific group that produce it, plus the EAMENA Durham Team group, but it should be hidden to the other two groups.

Advantages:

1. No need for new Arches set up

2. New data will be available for Durham Staff immediately.
3. Same database

Disadvantages:

1. Added complexity to an already complex model.
2. Data duplication is very likely.
3. Creation of very precise membership rules might be a burden. Membership rules should apply usage rules as well (what one group can see, and what not)
4. The access restriction to some data might cause feeling of exclusion to some users.

WD: This does not lay out the technical details – in some cases there will be duplicate cards, but how do we choose which one to show to EAMENA Staff by default? What about for sites where there is no equivalent in the main EAMENA dataset? Will these records be hidden to others by default or are 'open' skeleton records needed? I feel like this will add another layer of complexity onto an already complex arches instance.