**Entitlement Subsystem Requirements**

Level A—Overview of Requirements

The purpose of an entitlement subsystem is to restrict access to data only to those for which such access is granted. To that end every data entity must have some description of access associated with it.

End users are only allowed to perform actions on objects to which they are entitled. This means for \*everything\* in the system. This has to be taken to the level that one may not even be allowed to know of the existence of an object unless they have an entitlement that allows that knowledge. This is especially important for the seismic brokers/service companies and to some extent the NDRs where even the knowledge of the existence of a data object might compromise its commercial or proprietary value.

Thus the entitlements subsystem is orthogonal to the workflows/actions users of the system perform. Those workflows/actions must be executed under the control of the entitlements subsystem. That is why it would be easier to implement if every end user action was driven by a specific workflow. In that case, the entitlements defined for current user define everything they can do and hence could even be used to drive the UI. If the ESRI-based interface is independent/parallel to the Fuse engine, then both modules will have to apply the "data screening" at a relatively fine grained level, probably in different ways.  
  
The entitlements subsystem itself does have workflows associated with it, and those were the ones I attempted to document (that is what Ken asked for). But it is the application of the entitlements so-defined that have to be applied everywhere else in the system where non-internal-admin users will interact.

Level B—Requirements

1. The system must have a data structure that defines restrictions on all data entities:
   1. geographic area or
   2. other person (or company) having been granted access,
   3. Time or date limits,
   4. etc. (list other)
2. The system must have a data structure that defines access to all data for all users. This may possibly require definitions of groups of users and/or companies.
3. The system data structures should to the extent possible conform to the PPDM data model for entitlements. This is important for marketing issues as well as giving a starting point for satisfying the first two requirements.
4. The system must have a mechanism whereby users are assembled into groups all of which have the same access capabilities. Groups should be within companies defined in the system.
5. The system must have an administrator interface to manage user access (and group access) to data, i.e., interact with and provide an administrator capability to manage the user data structure. This subsystem must provide capability to add users and establish their company and group, change a user’s company or group, and delete users. It should also have the capability to add and delete companies and groups with the capability to ensure all users remain in a company and group.
6. The system must have a user interface to define any restrictions to access to each data item, i.e., interact with and provide an administrator capability to manage the data item data base.
7. The ISDS must restrict access to data for those who do not have access. This includes even viewing or discovering that the data exists in the system.
8. The ISDS must verify that any data a user may want to acquire is actually available to him/her before delivering it to him/her.
9. The ISDS should provide a mechanism to define the access restrictions to a data item upon its creation and loading into ISDS.
10. The ISDS and its connection with the ESRI SDI map display must provide for enforcement of requirement 6 in the map display.

Level C—Detailed Requirements

# Background

This approach to entitlements is based on the PPDM Entitlements Module as represented in PPDM 3.8 data model. Basically this model has six main components:

* A Company which provides a mechanism for organizing Users,
* A User which is the entity that is granted rights under the entitlement,
* An *Entitlement\_Group* which groups set of *Users* which will be granted certain rights under a given *Entitlement*,
* An *Entitlement\_Data* object which provides a reference to the entity whose access is controlled by the *Entitlement*,
* An Access\_Feature which defines the kinds of operation the members of the *Entitlement\_Groups* can perform on the *Entitlement\_Data* objects, and
* The *Entitlement* itself which associates one or more *Entitlement\_Groups* with one or more *Entitlement\_Data* objects. The *Entitlement* also specifies the *Access\_Features* associated with the Entitlement.

Each of these components has a set of required and optional metadata which include a GUID, Name, Description and a set of human readable attributes which completely define the object and provide human readable information to assist the Entitlement Administrators manage the Entitlement subsystem.

# Note

Depending on the kinds of access a customer may want to include in their instantiation of ISDS and how we decide to implement the functionality, a given *Access\_Feature* may wind up being identical to a particular workflow. The decision depends on a deeper understanding on my part of how far into the system workflows go. For example, if there is only light coupling or none at all between the ESRI interface and the Fuse workflow engine, then the *Access\_Features* must remain a separate entity. But if everything a User can do in the map interface is driven by a Fuse workflow, then we may want to combine the concepts. For the purposes of this discussion, I am assuming that (end) *Users* who access the system will never touch the FileTek components directly nor will they use the Troika utilities; rather these tools will only be used by data managers and administrators and not the end Users who must work under Entitlement controls.

# Workflow Actions

# Create *<Company>*

Creates a valid entry in the company table

1. Get company name
2. Capture additional required metadata (TBD)
3. Check to see if that name or a similar one already exists
4. If so
   1. Exit with Error
5. If not,
   1. Create *Company*
   2. Exit with Success

# Deactivate *<Company>*

In order to maintain a persistent audit trail, validly created companies which have participated in any transactions must not be deleted.

1. Get company name
2. Check to see if the *Company* exists in the catalog
3. If so
   1. If transactions exist for the *Company*
      1. Ask to confirm or cancel the deactivation
      2. If confirmed,
         1. Mark *Company* deactivated
         2. Exit with success
      3. If cancelled
         1. Exit with Error
   2. If transactions do not exist for *Company*
      1. Ask whether to delete or deactivate
      2. If delete
         1. Ask to confirm
         2. If confirmed
            1. Delete *Company*
            2. Exit with success
         3. If cancelled
            1. Exit with error
      3. If deactivate
         1. Ask to confirm or cancel the deactivation
         2. If confirmed
            1. Mark *Company* deactivated
            2. Exit with success
         3. If cancelled
            1. Exit with Error

# Create *<User>*

Users only have identities with respect to a particular company. If an individual working for one company which is registered with the system, leaves and joins another company, he/she must be created as a new user with a different identity.

1. Get individual’s name, company and other required metadata (TBD)
2. Check to see if the *User* already exists
3. If so
   1. Ask whether any information should be changed to handle a true new *User*
   2. If yes
      1. Get updated info
      2. Loop back to 2
   3. If no
      1. Exit with error
4. If not
   1. Create *User*
   2. Exit with success

# Deactivate *<User>*

In order to maintain a persistent audit trail, validly created users which have participated in any transactions must not be deleted.

1. Get *User* name (may need additional metadata)
2. Check to see if the *User* exists in the catalog
3. If so
   1. If transactions exist for the *User*
      1. Ask to confirm or cancel the deactivation
      2. If confirmed,
         1. Mark *User* deactivated
         2. Exit with success
      3. If cancelled
         1. Exit with Error
   2. If transactions do not exist for *User*
      1. Ask whether to delete or deactivate
      2. If delete
         1. Ask to confirm
         2. If confirmed
            1. Delete *User*
            2. Exit with success
         3. If cancelled
            1. Exit with error
      3. If deactivate
         1. Ask to confirm or cancel the deactivation
         2. If confirmed
            1. Mark *User* deactivated
            2. Exit with success
         3. If cancelled
            1. Exit with Error

# Create *<Entitlement\_Group>*

*Entitlement\_Groups* are the entities that are directly associated with a particular *Entitlement*. They must be created before any *Users* can be granted access. *Entitlement\_Groups* are defined independently of any particular *Entitlement* so that the group membership can be changed without having to modify each the *Entitlements* assigned to a particular set of *Users*. A given *Entitlement\_Group* can only contain *Users* from a single *Company*.

1. Present a list of registered *Companies*
2. Select *Company* for which the *Entitlement\_Group* is being created
3. Provide a name and the required metadata (TBD) for the *Entitlement\_Group*.
4. Ask for confirmation to create
5. If confirmed
   1. Create the *Entitlement\_Group*
   2. Exit with Success
6. If cancelled
   1. Exit with error

# Add *<User>* to *<Entitlement\_Group>*

In order to grant any kind of access to a particular *User*, it must be added to an *Entitlement\_Group*. Select *Entitlement\_Group*

1. Present a list of registered *Companies*
2. Select the *Company* for the desired *Entitlement\_Group*
3. Present a list of defined *Entitlement\_Groups*
4. Select the desired *Entitlement\_Group*
5. Display the current *Users* who are members of the chosen Entitlement\_Group, if any
6. Confirm that this is the correct *Entitlement\_Group*
7. If not,
   1. Ask whether to cancel the operation
      1. If cancelled
         1. Exit with error
      2. If not
         1. Go to 3
8. Present a list of *Users* associated with the *Company* who are not current members of the *Entitlement\_Group*
9. Select one or more *User(s)* to add to the *Entitlement\_Group*. Multi-item selection should be supported
10. Ask to confirm the addition of the *User(s)*to the *Entitlement\_Group*
11. If confirmed
    1. Add the *User(s)* to the *Entitlement\_Group*
    2. Exit with success
12. If not
    1. Ask whether to cancel the operation
       1. If cancelled
          1. Exit with error
       2. If not
          1. Go to 8

# Remove *<User>* from *<Entitlement\_Group>*

*Users* may need to be removed from an *Entitlement\_Group* for a number of reasons – reassignment to a new role within the Company that no longer requires access, leaving the Company, etc.

1. Present a list of registered *Companies*
2. Select the *Company* for the desired *Entitlement\_Group*
3. Present a list of defined *Entitlement\_Groups*
4. Select the desired *Entitlement\_Group*
5. Display the User of members of the chosen Entitlement\_Group
6. Confirm that this is the correct *Entitlement\_Group*
7. If not,
   1. Ask whether to cancel the operation
      1. If cancelled
         1. Exit with error
      2. If not
         1. Go to 3
8. Present a list of *Users* who are current members of the *Entitlement\_Group*
9. Select one or more *User(s)* to removed from the *Entitlement\_Group*. Multi-item selection should be supported
10. Confirm the removal of the *User(s)*
11. If confirmed
    1. Remove the *User(s)* from the *Entitlement\_Group*
    2. Exit with success
12. If not
    1. Ask whether to cancel the operation
       1. If cancelled
          1. Exit with error
       2. If not
          1. Go to 8

# Create *<Entitlement>*

The *Entitlement* object is the entity that associates one or more *Entitlement\_Groups* with an *Entitlement\_Data* object and defines the permitted operations that may be performed on the *Entitlement\_Data* object.

1. Present a list of registered *Companies*
2. Select the *Company* for which the *Entitlement* is being created
3. Provide a name for the *Entitlement* and other required metadata (TBD)
4. Ask for confirmation to create *Entitlement*
5. If confirmed
   1. Create *Entitlement*
   2. Exit with success
6. If not
   1. Ask whether to cancel the operation
      1. If cancelled
         1. Exit with error
      2. If not
         1. Go to 3

# Add *<Entitlement\_Group>* to *<Entitlement>*

Once an *Entitlement* is created, one or more *Entitlement\_Groups* can be associated with it. This allows a general *Entitlement* (say, for map view of all surveys) can be shared among *Users* from multiple *Companies*.

1. Present a list of currently defined *Entitlements*. Some filtering of this list may be desirable
2. Select an *Entitlement*
3. Present the Entitlement information: metadata, *Entitlement\_Groups*, *Entitlement\_Data, Access\_Features*. This may be necessary to make sure the right *Entitlement* was selected.
4. If this is not the right *Entitlement*
   1. Ask whether to cancel the operation
      1. If cancelled
         1. Exit with error
      2. If not
         1. Go to 1
5. If this is the right *Entitlement*
6. Present a list of the currently defined *Entitlement\_Groups* that are not already associated with this *Entitlement*. Some filtering of the list may be desirable (e.g., restrict to a given *Company*’s *Entitlement\_Groups*)
7. Select an *Entitlement\_Group*
   1. Display the *Users* and the relevant metadata
   2. If this is the right *Entitlement\_Group*
      1. Confirm addition of the *Entitlement\_Group* to the *Entitlement*
         1. If confirmed
            1. Add *Entitlement\_Group* to *Entitlement*
            2. Exit with success
         2. If not confirmed
            1. Ask whether to cancel the operation

If cancelled

Exit with error

Go to 6

* 1. If this is not the right *Entitlement\_Group*
     1. Ask whether to cancel the operation
        1. If cancelled
           1. Exit with error
        2. If not cancelled
           1. Go to 6

# Remove *<Entitlement\_Group>* from *<Entitlement>*

One may need to remove an entire *Entitlement\_Group* from an existing *Entitlement* because of a change in status of the members of the *Entitlement\_Group* or their *Company* (e.g., termination of a contract).

1. Present a list of currently defined Entitlements. Some filtering of this list may be desirable
2. Select an Entitlement
3. Present the Entitlement information: Metadata, *Entitlement\_Groups*, *Entitlement\_Data, Access\_Features*. This may be necessary to make sure the right Entitlement was selected.
4. If this is not the right Entitlement
   1. Ask whether to cancel the operation
      1. If cancelled
         1. Exit with error
      2. If not
         1. Go to 1
5. If this is the right Entitlement
   1. Select an Entitlement\_Group
   2. Display the Users and the relevant metadata
   3. If this is the right Entitlement\_Group
      1. Confirm removal of Entitlement\_Group
         1. If confirmed
            1. Remove Entitlement\_Group from Entitlement
            2. Exit with success
         2. If not confirmed
            1. Ask whether to cancel the operation

If cancelled

Exit with error

Go to 5

* 1. If this is not the right Entitlement\_Group
     1. Ask whether to cancel the operation
        1. If cancelled
           1. Exit with error
        2. If not cancelled
           1. Go to 5

# Create *<Entitlement\_Data>*

This is the key component of the whole access control system. The *Data\_Objects* necessarily must be defined and maintained outside of the *Entitlements* subsystem itself, but the *Entitlements* subsystem will be used by the rest of the system to control and manage access.

The specific requirements for what a given deployment will require will need to be completely spelled out for each contract, but at a minimum, the following *Data\_Objects* will need to be supported:

* 3-D Survey (all products)
* Individual 3-D survey products (e.g., pre-stack data, post stack time migration, etc.)
* 2-D survey (all products)
* 2-D line (all products)
* 3-D survey blocks
* 3-D survey polygons
* 2-D survey line segments
* Well data
* Etc.

The steps in the workflow are as follows:

1. Present a list of the *Data\_Objects* known to the system (surveys, lines, products, subset, etc.) Some filtering will be necessary
2. Select the *Data\_*Objects which the *Entitlement\_Data* will control
3. Provide a name for the *Entitlement\_Data* and other required metadata (TBD)
4. Ask for confirmation to create *Entitlement\_Data*
5. If confirmed
   1. Create *Entitlement\_Data*
   2. Exit with success
6. If not
   1. Ask whether to cancel the operation
      1. If cancelled
         1. Exit with error
      2. If not
         1. Go to 1

# Add *<Entitlement\_Data>* to *<Entitlement>*

An *Entitlement* can control access to one or more *Entitlement\_Data* objects.

1. Present a list of currently defined *Entitlements*. Some filtering of this list may be desirable
2. Select an *Entitlement*
3. Present the *Entitlement* information: Metadata, *Entitlement\_Groups*, *Entitlement\_Data, Access\_Features*. This may be necessary to make sure the right *Entitlement* was selected.
4. If this is not the right *Entitlement*
   1. Ask whether to cancel the operation
      1. If cancelled
         1. Exit with error
      2. If not
         1. Go to 1
5. If this is the right *Entitlement*
   1. Present a list of the currently defined *Entitlement\_Data* objects. Some filtering of this list may be desirable
   2. Select an *Entitlement\_Data* object
   3. Present the *Entitlement\_Data* information: Metadata, *Data\_Objects*. This may be necessary to make sure the right *Entitlement\_Data* object was selected.
   4. Ask for confirmation to add the *Entitlement\_Data* to the *Entitlement*
   5. If confirmed
      1. Add the *Entitlement\_Data* to the *Entitlement*
      2. Exit with success
   6. If not
      1. Ask whether to cancel the operation
         1. If cancelled
            1. Exit with error
         2. If not
            1. Go to 5

# Remove *<Entitlement\_Data>* from *<Entitlement>*

This workflow may be required for a variety of reasons, e.g., the need for finer granularity in the set of *Entitlement\_Data* objects, etc.

1. Present a list of currently defined *Entitlements*. Some filtering of this list may be desirable
2. Select an *Entitlement*
3. Present the *Entitlement* information: Metadata, *Entitlement\_Groups*, *Entitlement\_Data, Access\_Features*. This may be necessary to make sure the right *Entitlement* was selected.
4. If this is not the right *Entitlement*
   1. Ask whether to cancel the operation
      1. If cancelled
         1. Exit with error
      2. If not
         1. Go to 1
5. If this is the right *Entitlement*
   1. Present a list of the currently defined *Entitlement\_Data* objects. Some filtering of this list may be desirable
   2. Select an *Entitlement\_Data* object
   3. Present the *Entitlement\_Data* information: Metadata, *Data\_Objects*. This may be necessary to make sure the right *Entitlement\_Data* object was selected.
   4. Ask for confirmation to remove *the Entitlement\_Data* from the *Entitlement*
   5. If confirmed
      1. Remove the *Entitlement\_Data* from the *Entitlement*
      2. Exit with success
   6. If not
      1. Ask whether to cancel the operation
         1. If cancelled
            1. Exit with error
         2. If not
            1. Go to 5

# Add *<Access\_Feature>* to *<Entitlement>*

To complete the definition of an Entitlement, one or more Access\_Features must be added to the Entitlement object. The specific requirements for what a given deployment will require will need to be completely spelled out for each contract, but at a minimum, the following *Access\_Features* will need to be supported:

* View location (allows the associated *Entitlement\_Data* to be displayed on a map to the *Users* in an associated *Entitlement\_Group*)
* View metadata (allows the associated *Entitlement\_Data*’s metadata to be displayed to the *Users* in an associated *Entitlement\_Group*)
* View Data (allows the associated *Entitlement\_Data*’s *Data\_Objects* to be displayed to the *Users* in an associated *Entitlement\_Group*)
* Select (allows an associated *Entitlement\_Data*’s *Data\_Object* to be chosen the *Users* in an associated *Entitlement\_Group* for additional operations)
* Move to Shopping Cart (allows the associated *Entitlement\_Data*’s selected data to be queued for extraction delivery the *Users* in an associated *Entitlement\_Group*)
* Subset (allows the associated *Entitlement\_Data* selected *Data\_Object* to be sliced-and-diced by the *Users* in an associated *Entitlement\_Group*)

The steps in the workflow are as follows:

1. Present a list of currently defined *Entitlements*. Some filtering of this list may be desirable
2. Select an *Entitlement*
3. Present the *Entitlement* information: Metadata, *Entitlement\_Groups*, *Entitlement\_Data, Access\_Features*. This may be necessary to make sure the right *Entitlement* was selected.
4. If this is not the right *Entitlement*
   1. Ask whether to cancel the operation
      1. If cancelled
         1. Exit with error
      2. If not
         1. Go to 1
5. If this is the right *Entitlement*
   1. Present a list of the possible *Access\_Features* that can be enabled.
   2. Select one or more *Access\_Features* to add to the Entitlement. Multi-item selection should be supported
   3. Ask to confirm the addition of the *Access\_Feature(s)* to the *Entitlement*
   4. If confirmed
      1. Add the *Access\_Feature(s)* to the *Entitlement*
      2. Exit with success
   5. If not
      1. Ask whether to cancel the operation
         1. If cancelled
            1. Exit with error
         2. If not
            1. Go to 5

# Remove *<Access\_Feature>* from *<Entitlement>*

From time to time it may be necessary to remove an Access\_Feature from a given Entitlement.

1. Present a list of currently defined *Entitlements*. Some filtering of this list may be desirable
2. Select an *Entitlement*
3. Present the *Entitlement* information: Metadata, *Entitlement\_Groups*, *Entitlement\_Data, Access\_Features*. This may be necessary to make sure the right *Entitlement* was selected.
4. If this is not the right *Entitlement*
   1. Ask whether to cancel the operation
      1. If cancelled
         1. Exit with error
      2. If not
         1. Go to 1
5. If this is the right *Entitlement*
   1. Present a list of the currently defined *Access\_Features*
   2. Select one or more *Access\_Features* to remove from the Entitlement. Multi-item selection should be supported
   3. Ask to confirm the removal of the *Access\_Feature(s)* from the *Entitlement*
   4. If confirmed
      1. Remove the *Access\_Feature(s)* from the *Entitlement*
      2. Exit with success
   5. If not
      1. Ask whether to cancel the operation
         1. If cancelled
            1. Exit with error
         2. If not
            1. Go to 5