New process - draft

Registration and whitelisting flow, initial premises

- Done in completely automatic fashion.
- SP needs to be listed in one of the metadata that SA is consuming, at the moment: eduGAIN, OpenAthens, SWAMID, InCommon
- Technical and Administrative contact from the SP metadata are taken as contacts that SA is recognising
- Advanced (and potentially Standard) implementors will need to register the API keys in order to call the persistence service API
- For API key registration domain ownership needs to be proved by inserting a defined record in their DNS?
- Once an API key is registered, there needs to be a process for renewal. It can be an automatic job, and the old key is left functioning if there is a job error.
- During the registration process, SPs need to accept the terms of use:
  - Advanced - registration flow in the website, part of click-through, policed through API key registration process
  - Standard - registration flow in the website, part of click-through, policed through API key registration process if mandatory for standard
  - Limited - registration flow in the website, part of click-through, no way to police

Which entity categories we need:

1. ToS - assert accept from from incoming feeds and also assert ourselves as part of pixiedusting - think also about accepting it from
2. Authorisation for using the advanced (they need to read and write to)

Filtering ideas:

- have a special or vanity URL that has special fitelring configured ... do they need to have a separate
- this may become

Option 3 (start process) - Registration of the SP and acceptance of ToS

This is the process we will start with, as eventual method for Option 3 that is made more automatic and needs some amount of coding in both front and back end. This process assumes that there is a semi-manual registration process that SP owner and SA team need to follow.

- SP owner sends email from the admin or technical contact published in its metadata. Email needs to state:
  - The integration SP wishes to use: Limited/Standard/Advanced
  - entityID of the SP they wish to register
  - Acceptance of the ToS
  - (Whether SP opts-in to be published in SA website as using the SA) - we can just require this but needs to be added to ToS-we also want to publish this to metadata
  - Whether SP wishes to be added to the SA communication channels - Slack SA general channel, the users mailing list, SA status notification...
- ??? some form for populating airtable - populate table from some kind of web form ... and how to extract information ... also if there are any risks with using airtable without payed licence
- Which email and who is looking to that and on which schedule? What is the response time we want to establish for this? For the sake of this process, lets call this the job of the Level 1 support.

- L1 support records the registration, that includes:
  - Record the request in Airtable or something alike
  - L1 support checks if the SP is published in any of the metadata that SA consumes
  - If the requested integration is advanced, then the request is forwarded to the SA xx team. Wait until the SA xx team has approved the integration and then continue...Update the Airtable
  - SA xx team validates that advanced integration is approved: proof that they are following ToS, UX validation
  - Record the integration in the website, if opted in (can we use airtable automatically for this?)
  - Add them to the communication channels if opted-in

Option 3 (start process) - Registration of the API key

- ... TO BE DESCRIBED BY LEIF
- SP admin uses curl call that is described in documentation. Shibboleth SP key is used to sign a message that is then sent to the API server.
- API server checks if the SP is registered by checking a list of the entity-IDs that is prepared based on the Airtable (check if this is possible)? - use api to fetch data from airtable and pixiedust the MD
- API server responds with JSON and key?
- SP should place the key in the environment variable or in the file - documentation for this to be provided.

Option 3 (eventual process) - Registration of the SP and acceptance of ToS

This is eventual method for Option 3 that is as automatic as possible. It assumes that there is a UI registration process for the SP owner. The SP owner needs to prove the control over the entity and the SA site marks this in metadata that is being aggregated by SA.

- Person that wants to register SP, needs to do the registration over the SA website, that looks something like this:
  - he chooses SP from the list which is being populated from metadata SA is consuming
  - UI presents the email addresses of the administrative/technical contacts that are registered with that SPs metadata.
  - he needs to choose one of the email addresses to prove s/he has access to it, and then clicks "send email"
  - he receives email with a link containing a long string. Click on that link takes s/he to the SP specific registration page on seamlessacces
s.org
  - (instead of everything above, we can have login, but I am not sure whether person that administers SP also has an ldP to login with)
This page shows some of the data about the SP that is parsed from the metadata, with message to correct this data through SPs published metadata if needed.

There are checkboxes to:
- accept ToU (mandatory)
- choose which SP contact email to add to the users mail list (optional)
- choose which SP contact email to add to status notifications (?) (optional)
- opt-in to being listed in the SA site as one of the users
- choose if it is limited, advanced or standard integration

For advanced integration, there is an approval process as well as the implementation needs to be approved by the SA xx team.

After registration there is a log created - to be defined what information and in which format, and adding to the lists?

Limited and Standard integrations gets immediately added to an internal list, and Advanced after approval (can we do it over UI as well?). This list need only to contain entityID of the SP.

Based on this list, and EC is added to the SA metadata.

* Based on SA metadata, we can have an internal and external view on who are SPs that use SA.
  * internal view would be for the use by the SA team and would have basic info about SP, contacts from metadata and which integration they use
  * external for start number of SPs using advanced and standard.

**Option 3 (eventual process) - Registration of the API key**

* ... TO BE DESCRIBED BY LEIF *
  * SP admin uses curl call that is described in documentation. Shibboleth SP key is used to sign a message that is then sent to the API server.
  * API server checks if the SP has entity category set in SA metadata and for advanced also if it is in the list of the approved ones.
  * API server responds with JSON and key ?
  * SP should place the key in the environment variable or in the file - documentation for this to be provided.

**Option 2 - Registration of the SP and acceptance of ToS via entity category**

* SP expresses its acceptance of ToS by adding entity category (which is TBD) to its metadata. Different EC are used for advanced and standard integration.
  * If SP’s federation allows, SP adds EC themselves to the metadata
  * If SP’s federation has strict policy and doesn’t allow SP to tag metadata with SA EC, then SP needs to send email from administrative contact expressed in the metadata to SA contact to request to tag its metadata with SA EC. Then SA will in its metadata aggregate tag the SP’s metadata with the EC.
* Seamless Access scans at the metadata periodically (hourly?) and records a list of SPs that has the entity category tag. Each SP seen to use this entity category, is recorded together with timestamp, noting all the changes (add/remove EC). This data is needed to be able to keep the history. This will be done by pushing the MD aggregate to github.
  * Every change is being notified by email to ?
  * For the advanced integration, the SP needs also to contact the SA team to approve its implementation.
  * SA team maintains a list of SP’s entityIDs that have been approved for advanced implementation.
* Based on SA metadata, we can have an internal and external view on who are SPs that use SA.
  * internal view would be for the use by the SA team and would have basic info about SP, contacts from metadata and which integration they use
  * external for start number of SPs using advanced and standard.

**Option 2 - Registration of the API key**

* ... TO BE DESCRIBED BY LEIF *
  * SP admin uses curl call that is described in documentation. Shibboleth SP key is used to sign a message that is then sent to the API server.
  * API server checks if the SP has entity category set in SA metadata and for advanced also if it is in the list of the approved ones.
  * API server responds with JSON and key ?
  * SP should place the key in the environment variable or in the file - documentation for this to be provided.

**Option 1 - Registration of the SP and acceptance of ToS**

* Registration is done via seamlessaccess.org website.
  * Person that wants to register SP, chooses SP from the list which is being populated from metadata SA is consuming.
  * UI presents the email addresses of the administrative/technical contacts that are registered with that SPs metadata.
  * Person needs to choose one of the email addresses to prove s/he has access to it, and then clicks “send email”
* Person receives email with a link containing a long string. Click on that link takes s/he to the SP specific registration page on seamlessaccess.org
* This page shows some of the data about the SP that is parsed from the metadata, with message to correct this data through SPs published metadata if needed.
  * There are checkboxes to:
    - accept ToU (mandatory)
    - choose which SP contact email to add to the users mail list (optional)
    - choose which SP contact email to add to status notifications (?) (optional)
    - choose if it is advanced or standard integration
* For advanced integration, there is an approval process as well as the implementation needs to be approved by the SA.
* After registration there is a log created - to be defined what information and in which format, and adding to the lists?
Need to check if SP exists in the log created in the registration flow in the website
... TO BE DESCRIBED BY LEIF
SP admin uses curl call that is described in documentation. Shibboleth SP key is used to sign a message that is then sent to the API server.
API server responds with JSON and key?
SP should place the key in the environment variable or in the file - documentation for this to be provided.

**Option 2 - Registration of the SP and acceptance of ToS via entity category**

- SP expresses its acceptance of ToS by adding entity category (which is TBD) to its metadata. Different EC are used for advanced and standard integration.
  - If SP’s federation allows, SP adds EC themselves to the metadata
  - If SP’s federation has strict policy and doesn’t allow SP to tag metadata with SA EC, then SP needs to send email from administrative contact expressed in the metadata to SA contact to request to tag its metadata with SA EC. Then SA will in its metadata aggregate tag the SP’s metadata with the EC.
- Seamless Access scans at the metadata periodically (hourly?) and records a list of SPs that has the entity category tag. Each SP seen to use this entity category, is recorded together with timestamp, noting all the changes (add/remove EC). This data is needed to be able to keep the history. This will be done by pushing the MD aggregate to github.
- Every change is being notified by email to?
- For the advanced integration, the SP needs also to contact the SA team to approve its implementation.
- SA team maintains a list of SP’s entityIDs that have been approved for advanced implementation.
- Based on SA metadata, we can have an internal and external view on who are SPs that use SA.
  - internal view would be for the use by the SA team and would have basic info about SP, contacts from metadata and which integration they use
  - external for count number of SPs using advanced and standard.

**Option 2 - Registration of the API key**

... TO BE DESCRIBED BY LEIF

- SP admin uses curl call that is described in documentation. Shibboleth SP key is used to sign a message that is then sent to the API server.
- API server checks if the SP has entity category set in SA metadata and for advanced also if it is in the list of the approved ones.
- API server responds with JSON and key?
- SP should place the key in the environment variable or in the file - documentation for this to be provided.