

Accounting Service Redesign and Replacement

Current "Accounting" Use Cases Mapped to Services

XRAS (Allocations)

- Submit, review, approve allocation transactions
- Enforce allocation rules, policies, relationships.
- Create, update project metadata
- Communicate project updates and allocation transactions to SPs (via Accounting Service? **ms: AMIE**)
- Create, update project roles (PI, co-PIs, Allocation Managers)
- Communicate role changes to SPs & Accounting Service
- Post Allocation awards to ORCID profiles (if users have granted permissions)
- Provide XSEDE allocations information to XDMoD
- Notify users on results of allocation decisions, need for final reports
- Dealing with allocations associated with merged users
- Provide default allocation resource reviewers

XACCT (Accounting)

Projects & Users

- Create, update projects and allocations based on allocation transactions, including New, Renewal, Supplement, Extension, Transfer, Appeal, Advance, and Adjustment.
- Communicate project updates and transactions to SPs
- Communicate project & allocation status changes to SPs (active, overspent, expired, closed, pending, etc.)
- Notifications for Role users when defined set of usage-related situations arise (nearing expiration, nearing allocation limits, overspent, etc.)
- Support auditing of status for actions completed/pending from SPs
- Set user limits on allocations (XCI use case)
- Send automated project inactivation and notifications to PIs, et al
- XCI: Information Services retrieves via SQL: allocation <-> resource mappings in the acct schema resource information in the acct schema

People (Users & Identity)

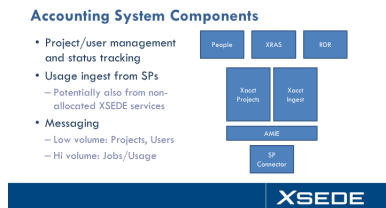
User-Managed

- Users create, manage their own user accounts & profiles
- Mapping ORCID IDs to user identities, collect permissions
- Distinguished Name (DN) generation and management

Admin/System-Managed

- Admins update, manage, deactivate, suspend user accounts
- Create, update, reconcile Organizations data
- Mapping between XSEDE user identities and SP user identities
- Support SSO Hub service
- Get/update DNs for users of site resources
- Merging duplicate users
- Emergency user suspension
- Username validation - not sure where this belongs or if we need to worry about it
- XCI: Globus and MyProxy OAuth use a user (portal) identity API that obtains user information from XDCDB

Initial XACCT Components



- Update project roles based on allocations input
- Add, remove users from projects & allocations
- Re-verify & retain users upon allocation renewal
- Support auditing of status for actions completed/pending from SPs
- Dealing with merged users

Ingest

- Accept resource usage, apply to allocations, including grid/multi-resource shared allocations, storage usage, cloud usage, HPC/compute usage
- Accept usage refunds, credits, debits from SPs, track and apply to project usage totals.
- Allow gateway and other services to "decorate" job records with additional job attributes (e.g., gateway user info associated with job run by "community user" account.
 - XCI: gateway-submit-attributes adds accounting attributes to jobs using an XDCDB API
- Provide XSEDE-collected usage information to XDMoD ?
- Provide usage information to users (xdusage and via XUP)
 - XCI: xdusage uses an XDCDB API to access allocations usage information

- Dealing with usage associated with merged users
- Auditing of status of usage from SPs
- XCI: XCI usage tracking services could have functional overlap with accounting usage

AMIE

- Transfer information from XACCT to SPs and from SPs to XACCT
- Define the protocol for information exchange between XACCT and SPs


Metrics/Reporting

- Support user and program reporting and metrics needs with necessary access, summaries
- Support Normalized Units, resource conversion factors

SP Connector

- Provide default SP-side mechanism to receive/send AMIE communications from/to XACCT

Documentation for XSEDE Accounting System (XDCDB)

File	Modified 
PDF File OSG GRACC.pdf	Feb 07, 2019 by David Hart
PDF File SweGrid SGAS.pdf	Feb 07, 2019 by David Hart
PDF File EGI DGAS and HLRmon.pdf	Feb 07, 2019 by David Hart
Microsoft Powerpoint Presentation accounting info.pptx	Mar 04, 2019 by David Hart
PNG File initial XACCT components.png	Apr 26, 2019 by David Hart

 [Download All](#)

Common/Shared/Managed Vocabularies

- Fields of Science
- Resource Types
- Project Statuses
- User Statuses
- Role Types
- Resource Unit Types
- Organizations

Current "Pain Points"

- Performance of processing usage packets (NPU) from HPC sites
 - Inability to handle the volume of jobs from HTC sites (e.g. OSG). Is this a requirement?
- Installation and testing of SP-side accounting components
- Processing of allocation transactions on current allocations when future allocations exist
- The concept of a project (charge_number, resource) is not modeled explicitly, It has to be inferred from the collection of allocations for a charge_number(account_id) and resource (resource_id).

The concept of a user account (charge_number, resource, person) is not modeled explicitly. It has to be inferred from the allocation_breakdown records associated with allocations.

Also, the concept of multiple allocations for a "project" has presented problems. Under POPS and now under XRAS this has continually created difficulties deciding if an allocation needs to be created vs finding one to be modified. It would have been a lot cleaner to just have a single allocation that gets renewed, supplemented, extended, etc.

project state is for a charge_number, resource; user account state is for a charge_number, resource, person. They don't have FKs into any table since projects and user accounts are not explicitly modeled. Also project state is based on the most recent allocation (ie, the allocation with the latest start date) even if that allocation starts in the future. This has generally worked well, but the latest allocation will "hide" the current allocation (ie, the allocation that has today's date between its start and end dates).

Since POPS did not send an identifier (like the XRAS action_id or request_id) in the AMIE NPC, allocations had to be created/located based on the start date sent by POPS in the NPC. That meant that POPS had to figure out what date to put into the NPC since POPS (like XRAS) did not have a start date for its requests.

- The PI role is handled differently than the other roles. The PI is attached (via the principal_investigators table) to the acct.requests table (and there are multiple requests over time because allocations have to have a request that has the same start date). The other roles are not "date based", but they are associated with resources. I think the original intention may have been to allow different roles for different resources, but now the roles are associated with the portal.teragrid resource only.

The AMIE RPCs need to send the PI in the RPC so they have to decide which request to use to find the PI. This meant that if an RPC was sent for an earlier request, the PI on that request might not be the current PI. AMIE was not modified to get the PI from the most recent request, but the PI for the request that was associated with the allocation. When the XDCDB gets a PI change (via POPS NPC or XRAS accounting service) it change the PI on all requests. We lose the history of the PI, but tracking PI history via requests is not a good way to do it since the PI can change for the current request as well.

- The proposal_number (XRAS request_number) and board_type (XRAS allocation_type?) are stored in the acct.requests table, instead of in the acct.accounts table (which only has the charge_number). This makes it tricky to match an XRAS request with a charge_number – the XDCDB solves this with triggers that force all acct.request entries for a given account_id to have the same proposal_number and board_type.
- The fact that allocations have a FK into the requests table based on the start date means that it is difficult to change the start date of an allocation – there has to be a request in place with the start date and if there isn't one, creating one is not a simple matter since abstracts, PIs, etc are attached to the requests.
- While user accounts can be marked 'dead' to prevent them from being reactivated during project reactivation, projects can only be marked as active or inactive, but not dead. Sometimes it is important to not reactivate inactive projects – for example if the project was explicitly inactivated by a DB admin, instead of due to out of funding or expiration. projects that are inactive for reasons other than funding or expiration should not get reactivated if extensions or supplements occur. Current the reactivation code looks for the reason that the project is inactive to determine if an extension or supplement can reactivate the project. It would have been better to use an explicit state (ie dead) to indicate this.
- The people table contains both real people and non-person (eg Community Users). There is no explicit indication of if the entry is a real person or not. This has created problems with DN generation since non-people do not get DNs. The DN generation process looks for the last name 'Community User' to determine if the "person" should get a DN or not.

I think SDSC has other "non-person" entries in the acct.people table which should not get DNs, but did.

- The jobs table is huge and only continues to grow. The impact is not only storage, but query performance.

It is not presently possible to eliminate "old" jobs (for example to only have jobs from the last 3 years), since deletes adjust the allocation the job was charged against.

One solution may be materialized views (as a kind of job data warehouse) for specific sets of queries.

Another may be a archived jobs table, where "old" jobs are moved (inserts adjust the allocation so that no usage is lost).

- Another issue is tracking people information. We do not require people to update their information – email, organization, nsf status (student, researcher, etc), address, etc, so this data is unreliable.
- There are resources with the same normalization factor, but different dollar values. The normalization factors are stored in the acct schema. The dollar values are in both the rdr2 and xras schemas. Moreover, rdr2 seems to have a different model for dollar values than xras does: rdr2.allocable_resource has a single value for a resource with no set of dates to assign different values over time; xras assigns dollar values based on the resource and the allocation_type (XDCDB board type), as well as having dates to provide different values over time (xras.allocation_type_resource_numbers)