People and Roles

Stakeholders

A Stakeholder in API Extension Architecture effort (API-X) is any individual who has potential business interest in using API-X in support a use case in production, and wishes to play a part in determining the direction of the API-X design, development, or implementation effort.

A stakeholder in API-X potentially has two areas of interest: The API-X framework as a whole, or specific use cases. These interests may be disjoint in time, and may evolve in time. The set of stakeholders involved at API-X at any given moment determines the direction of the collective effort of the whole group.   This effort may be directed towards the API-X framework itself, or specific extensions that have a shared interest among several stakeholders.

Responsibilities:

1. Prioritize work, determine overall goals for development sprints
2. Ratify API-X framework requirements implied by your individual use cases, or areas of interest
    a. Use case evaluations should have a list of potential requirements to choose from.
3. Collectively decide on milestones and deliverables for API-X sprints
    a. This may include development on specific extensions, the framework as a whole, or interactions between the two
4. Review the deliverables produced from development sprints
5. Communication with the technical/developer communities. Informing potential interested parties as to the API-X progress.
    a. Summarize sprints and communicate results to relevant communities

Current List of Stakeholders

This is the current list of stakeholders in the API-X, as well as a brief description of their use cases or area of interest.

- Joshua Westgard (University of Maryland): Package Deposit, Async External Preservation, and Binary Derivative Generation.
- Ruth Duerr (Ronin Institute for Independent Scholarship): actually there is a longer list of use cases I am interested in; but these all go together and are my highest priority
    - Package Deposit - https://wiki.duraspace.org/x/9JQpB
    - Deposit Workflow State - https://wiki.duraspace.org/x/9pQpB
    - Recover from Failed Package Deposit - https://wiki.duraspace.org/x/_5QpB
- Elliot Metsger (Johns Hopkins Univ):
    - Generic Use Case: Distributing API Extensions
    - Package Deposit
    - Provenance Stream
- Randall Floyd and William G. Cowan (Indiana University): Design - Asynchronous and Pluggable Storage
- Stefano Cossu (AIC): Content and structural validation – de-duplication – Access control (?)
  – Content model
- Unknown User (acoburn) (Amherst College): asynchronous workflows, ETL patterns, Distribution, Service discovery, Clustering (horiz. scaling) of services

Please add yourself to this list at any time if you wish to self-identify as a stakeholder