Design - Lower effort to deposit

Overview

The process of adding content to DSpace is very "manual data entry" oriented, with numerous opportunities to auto-populate metadata from external sources and systems.

Use Cases / Needs

- Integrations - Integrations that significantly lower the effort to fill DSpace with content, both from in house systems and third party content
- Depending on the types of integrations selected, this project may also be dependent on Design - Module Framework and Registry (in that some integrations may be best implemented as optional "modules")
- Loosely related: End User - Easy and Intuitive Deposit Interface (in that these integrations may need to touch the Deposit UI in general)

Planning Phase

- Based on the above use cases (and others gathered by the team), select a one or two integrations to prototype or build "beta versions" from
  - Determine whether such integrations would be worthwhile to prototype as "modules" as defined by the "Module Framework" project
  - If not, all should minimally be based on the same "interface"
- Build the prototype(s), determining the model or interface for additional integrations
- Scope the project, based on which integrations seem to provide the most immediate benefit to DSpace users (worldwide or regionally). Any integrations that fall outside of the project scope can be detailed as "to-do" and simply logged in JIRA for a future developer to "claim".

Development Phase

to be defined