Share Family Workshop

Jill Morris, PALCI
Sebastian Hammer, Index Data
Tiziana Possemato, Share Family
Anna Lionetti, Share Family

29th June 2026, ALA Conference, Chicago

https://share-family.org
https://wiki.share-family.org
info@share-family.org

PALCI pilot on e-book ILL

Resource Sharing: Licensed Digital Content

Collaborators

- **Consortium:** PALCI
- **Academic Libraries:** Lehigh University, New York University
- **Technology Partners:** Share Family, Index Data

Project Overview

A multi-phase, vendor-neutral, standards-based proof-of-concept exploring Linked Open Data to enable e-first interlibrary loan for licensed digital content. The project assesses the metadata, architecture, and workflows necessary for secure, scalable, and interoperable resource sharing.

Project Scope & Structure

Phase 1: Discovery Infrastructure

- Leverage Linked Open Data for interoperable discovery.
- Improve resource identification/matching across collections.
- Build metadata foundations for e-first fulfillment.

Phase 2: Fulfillment Infrastructure

- Design secure ILL workflows for licensed digital content.
- Integrate metadata, rights management, and secure delivery.
- Evaluate technical architecture for cross-institutional e-first fulfillment.

Why This Matters for Consortia

Potential Benefits for Library Consortia

The pilot explores how an interoperable, Linked Open Data ecosystem can help consortia:

- Enable e-first fulfillment workflows for licensed digital content.
- Improve discovery and matching across member institutions.
- Reduce metadata fragmentation through shared, entity-based metadata.
- Support secure, rights-managed access to licensed resources.
- Build a scalable, vendor-neutral infrastructure for resource sharing.
- Increase interoperability and reduce duplication of effort across institutions.

Integration into ReShare

What is ReShare

Launched in 2018 as a community-led resource sharing (ILL) initiative

In production in the US, Australia, and Germany

Governed by a steering committee, product team, Subject Matter Expert Group, and Index Data as platform stewards

The current data flow

1. Catalog and holdings data is retrieved and aggregated in Reservoir, and open-source tool; holdings are normalized
2. An algorithm derived from GoldRush is used to produce a match-key which in turns allows records and holdings to be clustered
3. the consolidated stream is fed into VuFind for discovery, and consumed by ReShare for request routing

So what problems are we trying to solve?

Our GoldRush-based clustering is *fine*... but it has some weaknesses

- It tends to be very picky about what records it matches, which is frustrating for users and leads to poor fulfillment rates
- By design, it keeps electronic and print resources completely separate, which is "correct", but not the effect we're looking for in our pilot
- There is interest in seeing whether e.g. entity browsing might aid ILL workflows

The data flow we're testing

1. ReShare's Reservoir will continue to aggregate and normalize catalog and holdings across consortial members
2. A feed of unclustered records is made available to the Share platform
3. The Share platform creates instance and work clusters
4. Reservoir consumes a feed back from Share-VDE including instance and work URIs
5. Reservoir builds clusters based on those URIs and feeds clusters to VuFind

Beyond basic validation (why we're doing this)

- Compare the Instance-level clustering from Share with GoldRush
  - But how, you ask!!
- Implement Digital requesting as an option from VuFind
- Implement digital fulfillment
  - Phase one will leverage existing nonreturnable copy delivery workflows

Future phases will explore full eBook ILL fulfillment within an ISO ILL context

PALCI as Share Family member

PALCI ENRICHMENT PROJECT

PALCI E-Book Enrichment Pilot

Exploring the LOD Platform used in the Share Family initiative to improve metadata indexing in ReShare for E-Book Discovery & Interlibrary Loan

This pilot explores the feasibility and effectiveness of using Share Family Entities to improve metadata indexing in ReShare, with a primary focus on reconciling multiple instances of a work — especially format differences (print vs. digital) — to facilitate e-first fulfillment.

The Problem

Current indexing relies on text strings, leading to split instances and manual work. This is expected to worsen with the introduction of e-books.

The Approach

The Share Family uses entity modeling and a hierarchical structure applying BIBFRAME with Share extensions (Opus → Work → Instance → Item) to cluster records meaningfully.

The Opportunity

Clustering digital and physical versions of the same book is particularly valuable for e-first ILL fulfillment.

PALCI Pilot Plan

The Palci pilot project is structured into three distinct phases, designed to systematically evaluate the E-Book Enrichment process and its impact on metadata quality and e-first fulfillment.

01

Phase 1: Planning & Test Case Definition

Define the specific scope and initial test cases for the pilot, identify key participants, and establish foundational steps for data sharing and collaboration.

02

Phase 2: Metadata Ingestion & Clustering

Ingest the selected pilot metadata into the LOD Platform environment, leveraging the Share Family's capabilities to perform entity clustering and create enriched metadata.

03

Phase 3: Results Analysis & Comparison

Evaluate the quality and effectiveness of the Share Family clustering results, comparing them with the current methods and analyzing digital entitlement representation.

Pilot Data Set

For our pilot, we analyzed a comprehensive dataset comprising approximately **5.7 million records** from three PALCI member libraries, including both print and electronic resources. This dataset was carefully selected to represent a diverse range of collection types and sizes.

New York University (US-NNU)

- Ebooks: 631,114
- All Resources: ~2,702,159

The New School (US-NNNS)

- Ebooks: 24,815
- All Resources: ~182,344

Lehigh University (US-PBL)

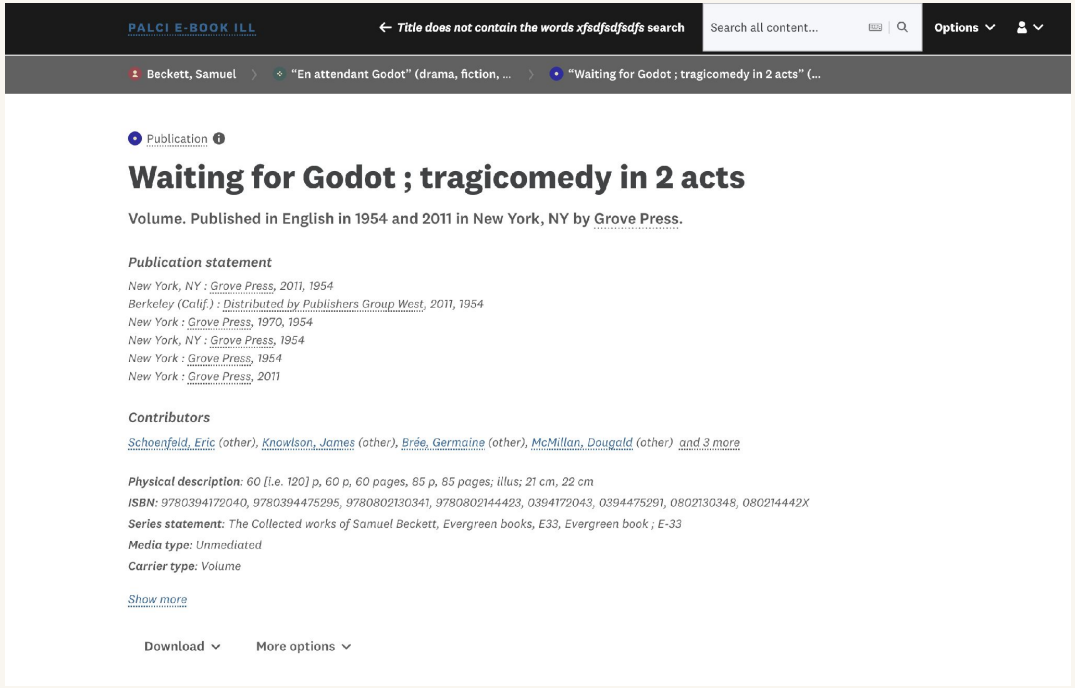
- All Resources (Ebooks & Print): ~2,085,620

This diverse collection allowed us to rigorously test the LOD Platform's ability to identify, cluster, and enrich metadata across various formats and institutional contexts.

The PALCI Share Portal

The PALCI consortium's Share portal was built using approximately **5.7 million bibliographic records** contributed by the member libraries. This massive dataset includes both print and electronic resources, forming a comprehensive foundation for the platform.

The resulting platform serves both end users and cataloguers, enhancing resource discovery while supporting a more effective and integrated user experience for accessing library services. It streamlines how users find and access resources and improves metadata management for librarians.



https://palci.preview.share-family.org/waiting-for-godot-tragicomedy-in-2-acts-p831781602233617/copies

A Closer Look on the Analysis

During Phase 1, a pre-analysis was conducted to confirm that the Share Family reconciliation logics would lead to the desired outcome. The analysis used a matching system to identify each print resource with its corresponding digital resource and viceversa. The results were published on a web application for data visualisation.

Only a small subset of records has been loaded to the matching system for demonstration purposes.

The Matching System

The web application visualizes the effort to reconcile records via tag 776.
A file of bibliographic records is processed, and records with a 776 tag pointing to a record in another format are extracted. Based on the data in the 776 tag, the system searches all uploaded bibliographic records for possible matches. Its interface displays three sections:

Input (Original)

The source record containing a 776 field.

In the processed records, sometimes 776 fields are incomplete or point to external IDs.

Enriched (776)

It isolates the 776 tag of the input record. The $w subfield is enriched with the record IDs of the corresponding matching records. Any $w enrichments are highlighted in red.

Matching

List of Records with a possible match to the 776 data — not all of which result in enrichment.

Enrichment criteria take into account a range of subfields and are currently being refined.

Case 1: Match with Desired Format

Record

11158249_US-PBL

"Ordinary people, extraordinary lives : political and economic change in a Tōhoku village" — Virginia R. Domínguez.
Its tag 776 points to a **Print version** with LCCN **91015781**

3 Records Found

The matching system found 3 records, one from each of the 3 institutions in the test dataset. All three have tag 010 with the matching LCCN identifier.

Enrichment Applied

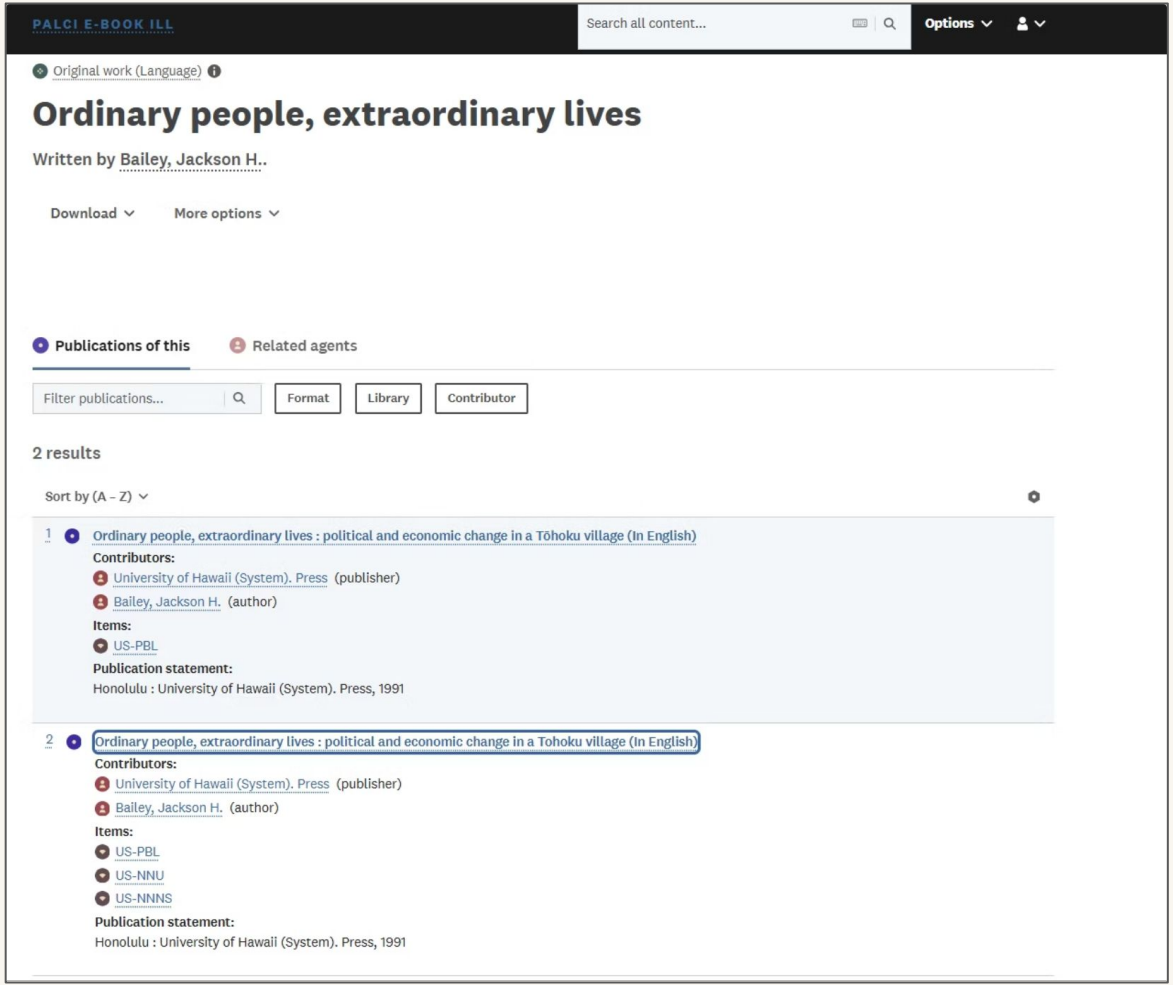
The system enriches the $w subfield of field 776 with data retrieved from field 999 of each institution's record; specifically, it derives the original record identifier from subfield $l and the owning institution from subfield $s.

Result

Three new $w subfields added:

=**776** 08**$i**Print version:**$a**Bailey, Jackson H.**$t**Ordinary people, extraordinary lives.**$d**Honolulu : University of Hawaii Press, ©1991**$z**0824812999**$w**(DLC) 91015781**$w**(OCoLC)23694026**$w(US-NNNS)990019806010107875$w(US-NNU)990019806010107876$w(US-PBL)444804**

Case 1: Match with Desired Format in Share enviroment



Case 2: Match in the Wrong Format

Record

000118589323 US-PBL
"People as subject, people as object" — Virginia R. Domínguez

Its tag 776 points to an **online version** of the resource.

Outcome: No Enrichment

The system found 2 matching records, but both are **print** — not the desired online format. Since the format does not match the $i subfield in 776, no $w enrichment occurs.
Format is determined by inspecting tags **300** / **337** / **338** in the matched records.

Case 3: Match Blocked by Volume Mismatch

Record

00011409609_US-PBL
"We the People. Volume 2, Transformations" — Bruce Ackerman

Its tag 776 points to a **Print version,** with a title specifying that the edition refers only to volume two.

Outcome: No Enrichment

The system finds a match with a record *"We the People"* by ISBN; however, the match is discarded because the title of the retrieved record refers to the complete work composed of three volumes, confirmed by the tag 300 \\ **$a**volumes <1-3>

Other cases 4–6

Case 4: Failed Match

10799788 US-PBL — *"The Grove Symposium."* A general match is found, but the distance between the 776 $a data and the candidate record is too great. No enrichment occurs.

Case 5: Original ($i = "Original")

00011996901 US-PBL — *"The rise and fall of freedom of contract."* There is a match with two records thanks to the LCCN identifier of tag 776. The enrichment is applied even though the $i Original in tag 776 suggests that the resource is a copy.

Case 6: Misleading ISBN

00012019444 US-PBL — *"Amplify" — Adam Met.* There is a match with two records via ISBN, but the records found refer to a print version, while tag 776 specifies, through the use of $i, that it is an online version. The match fails.

Key Takeaways & Next Steps

PALCI
Partnership for Academic Library Collaboration and Innovation

Refining Entity Matching

- Extend the matching logic to consider additional MARC subfields.
- Improve the interpretation of free-text cataloguing information through defined matching criteria.

Managing Data Quality

- Support the identification of cataloguing inconsistencies and conflicting metadata.
- Increase the robustness of entity matching in the presence of incomplete records relevant record due to a failure to handle missing information.

Improving the Matching Pipeline

- Address the remaining edge cases that may prevent relevant records from being clustered correctly.

Integration into the Share Family Ecosystem

- Incorporate the enrichment logic into the Share Family clustering pipeline.
- Expose enriched relationships and matching results through the PALCI discovery portal, enhancing both user discovery and resource-sharing workflows.

Share FAMILY

Thank You!

Learn more about JCricket, SHARE-VDE, and the broader SHARE Family of projects through the resources below. We welcome questions, contributions, and collaboration from catalogers and data stewards across the community.

SHARE Family

share-family.org

SHARE Wiki

wiki.share-family.org

Contact Us

info@share-family.org

SVDE Portal

svde.org

JCricket best practices

JCRICKET
Entity Editor

Best Practices for Using an Entity Editor in a Shared Environment

A practical guide for catalogers and data stewards working across multi-institution, entity-based cataloguing systems.

Tiziana Possemato, Share Family

Modernizing Cataloguing: A Shared Vision

Inspired by the 2008 Library of Congress Working Group on the Future of Bibliographic Control report *On the Record*, the JCricket Entity Editor embraces web standards and Linked Open Data to support a global, interoperable knowledge environment. This shift from traditional record-based practices to entity-based data management enhances clarity, reusability, and seamless integration across systems.

The Share Family's tenant-based architecture enables institutions to manage their data independently while contributing to a collaborative ecosystem. By moving bibliographic data beyond institutional silos, it increases visibility, strengthens interoperability, and supports data exchange at an international scale.

What is JCricket?

JCricket
Entity Editor

JCricket is a BIBFRAME-based entity editor for managing and enriching Linked Open Data. It enables the creation, editing, merging, and splitting of entities derived from bibliographic and authority records, as well as the creation of new entities from scratch.

Streamlined Entity Management

Enables member libraries to manually correct errors in entities generated from MARC record conversion within the Share portal.

Enhanced Data Quality

Applies merge and split functions to improve the accuracy and consistency of automated clustering processes in the Cluster Knowledge Base (CKB).

Interoperability & Notifications

JCricket APIs communicate entity changes back to member libraries, supporting informed decisions and optional entity updates.

Integrated into the Share discovery portal, JCricket empowers authenticated users to contribute to a collaborative and high-quality data environment.

Use of JCricket in a Shared Environment

Operating within a multi-institution, shared, entity-based catalog, JCricket's inherent complexity necessitates agreed-upon guidelines. These high-level practices, developed in collaboration with the Share Family's SEI working group, ensure effective and consistent use across the community of catalogers.

Establish Practices

For working on a multi-institution, shared, entity-based catalogue.

Consolidate Workflows

To optimize cataloguing processes within JCricket.

Ensure Data Integration

For streamlined, consistent data flow with external systems.

Collaboration Scenarios and Best Practices

Collaboration Scenarios

The scenarios cited in this presentation highlight key areas of collaboration within JCricket.

Each scenario involves cataloguing actions that support data quality and interoperability. Some actions rely on existing procedures, while others require the development of shared community best practices.

JCricket Practices Cover

- Core set of properties for entity description
- Selection of preferred forms (e.g., single-valued properties, controlled vocabularies)
- Constraints on editing data from other Provenances
- Criteria for entity cancellation
- Criteria for selecting authoritative external sources

Policy 1: Adding an Additional Property Value

A foundational best practice in JCricket concerns how catalogers should handle the addition of a new value to a property that already exists in an entity's description. Understanding the type of value being added is critical to applying the correct workflow.

1 — Text String Attribute

A literal text value entered directly by the cataloger, such as a title or note field.

2 — Controlled Vocabulary Value

A value drawn from a predefined, managed vocabulary — either open or closed within SVDE.

3 — Relationship to Another Entity

A link establishing a typed relationship between the current entity and another entity in the shared catalogue.

Policy 1: Adding a Property Value as a Text String

A **text string** is a property with a literal value entered directly by a cataloger. The behavior of that string — and how it appears in the user portal — depends on whether the property is defined as single-valued or multi-valued.

Single-Value

Only **one value** is displayed on the entity page for a given property, regardless of how many institutions have contributed different strings. A single value must be designated as **IsLeader** to appear in the portal interface.

Multi-Value

One **or more values** for the same property can be displayed on the entity page. Catalogers use the **IsLeader** flag to indicate which values — given the same language — should be surfaced to users in the portal.

Mono-Value Property in JCricket

The **has Title** property is a classic example of a single-value property. Multiple institutions have contributed slightly different title strings for the same instance — all in English in this example — reflecting minor variations in transcription practice.

Stanford

One Hundred Years of Solitude and Other Novels (eng)

National Library of Norway (NLN)

One Hundred Years of Solitude and Other Novels (eng)

New York University (NYU)

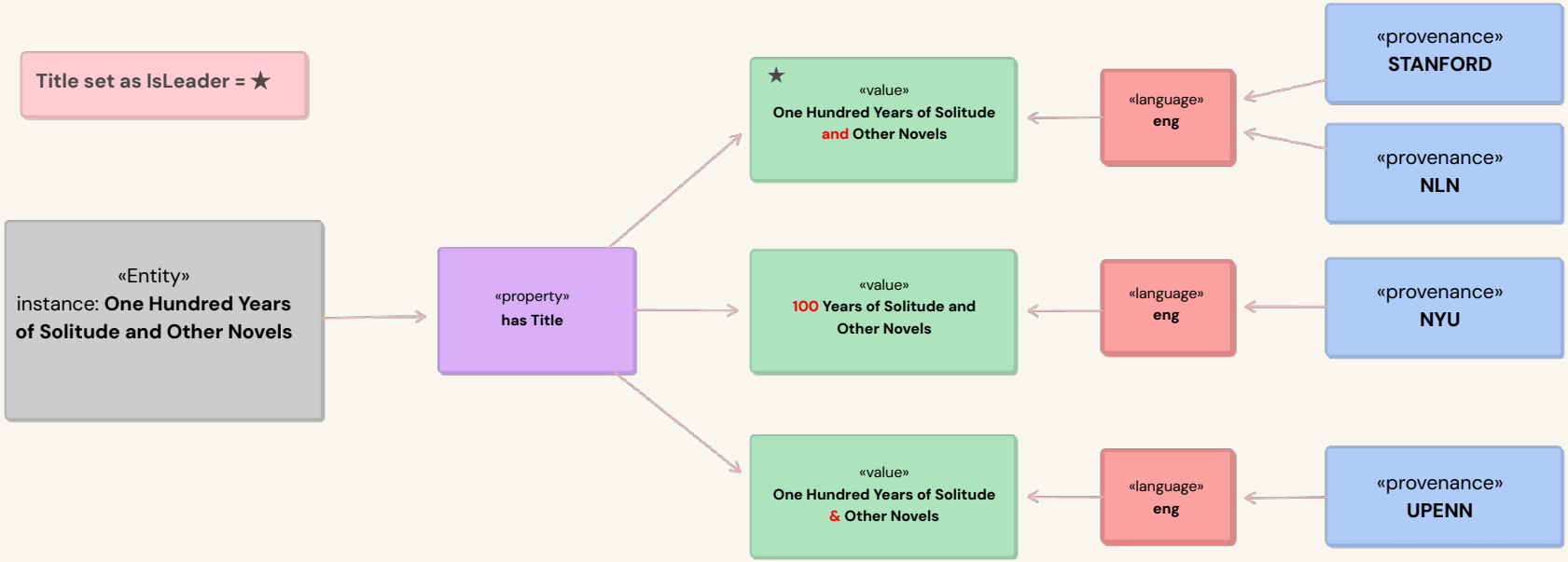
100 Years of Solitude and Other Novels (eng) — uses numeral

University of Pennsylvania (UPENN)

One Hundred Years of Solitude & Other Novels (eng) — uses ampersand

Because **has Title** is single-valued, only one string can be displayed in the user portal. Catalogers must evaluate the variants and set **IsLeader** on the most authoritative or widely accepted form.

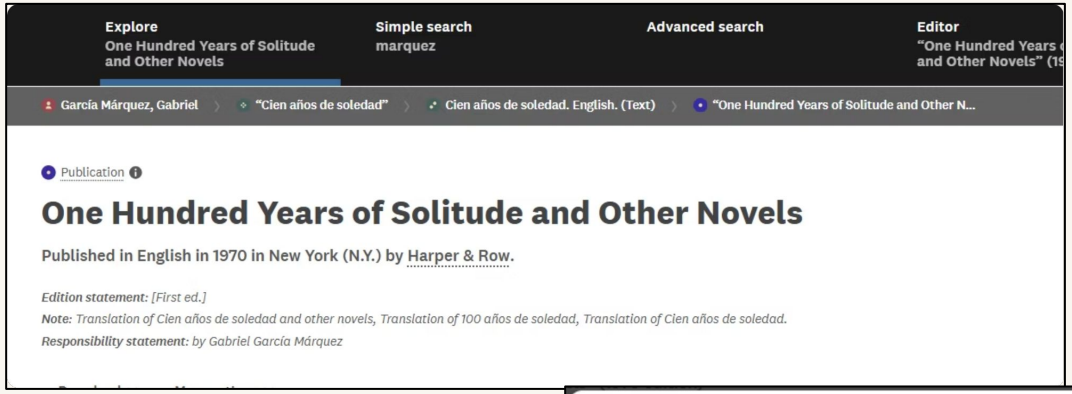
Mono-Value Property in JCricket

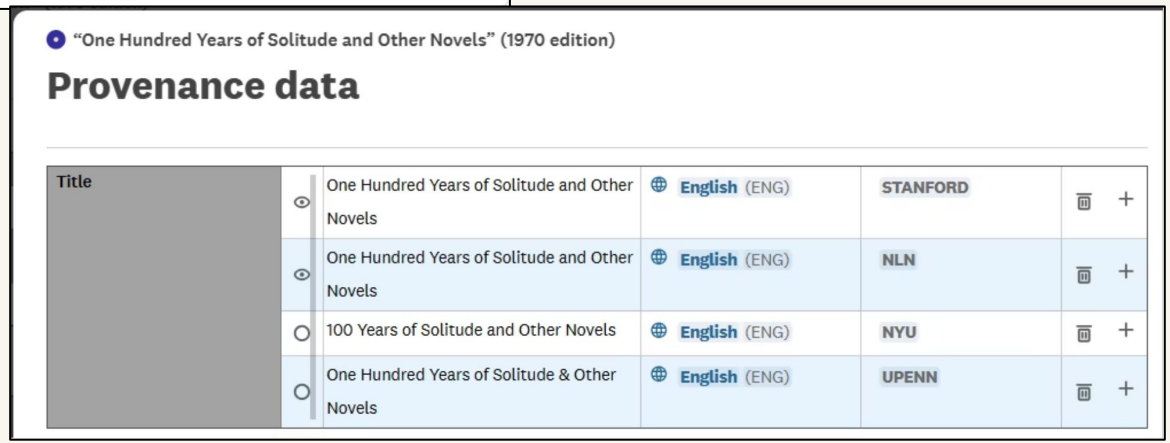


The above property statements, **has Title**, is considered single-value property, in the sense that only one value is displayed on the instance page. Hence, since STANFORD *(and NLN too)*, NYU and UPENN have typed different titles in the same language *(as the use of & or number in the title)*, one must be chosen and set as IsLeader for the display in the user portal interface.

Mono-Value Property — Portal View

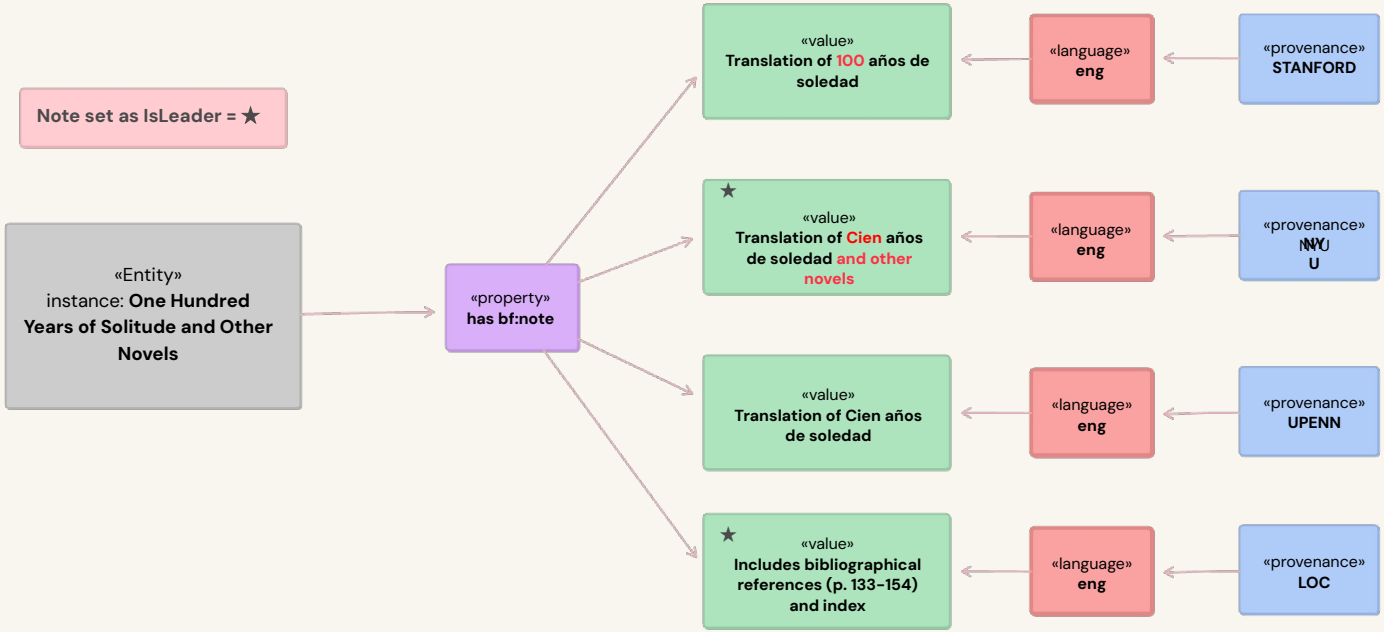
The screenshots below illustrate how a mono-value property appears in the JCricket interface. The public entity page (top) displays only the IsLeader title, while the Provenance data panel (bottom) reveals all contributed values and their institutional sources, allowing catalogers to review and reassign the IsLeader designation as needed.





Multi-Value Property in JCricket

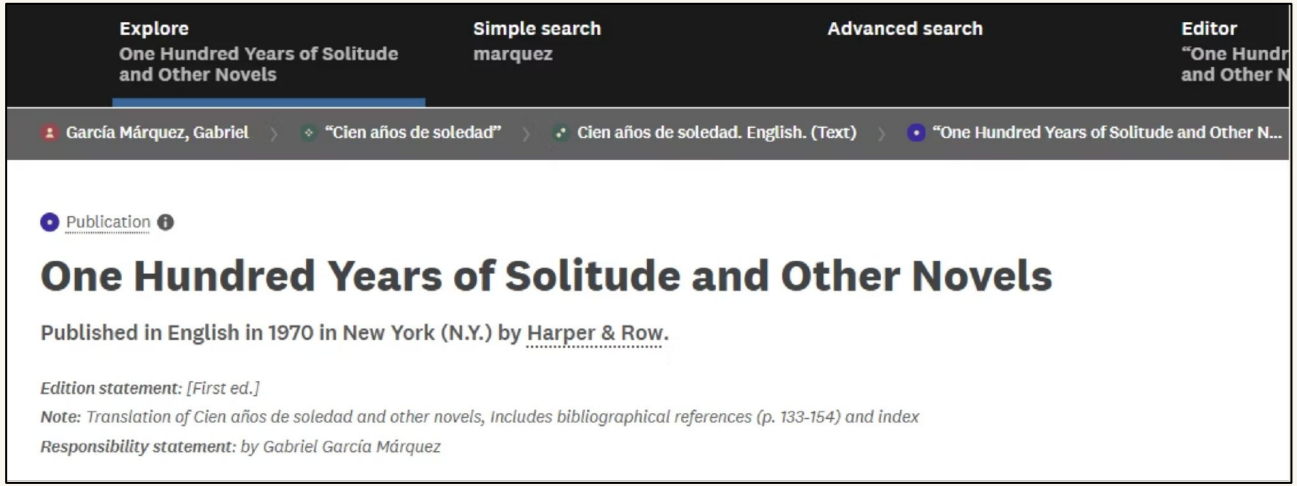
The **has bf:note** property illustrates a multi-value scenario. Four institutions have contributed notes to the same instance — three with nearly identical content (translation notes) and one with entirely different content (bibliographical references). The **IsLeader** flag helps catalogers control which notes are surfaced per language in the portal.

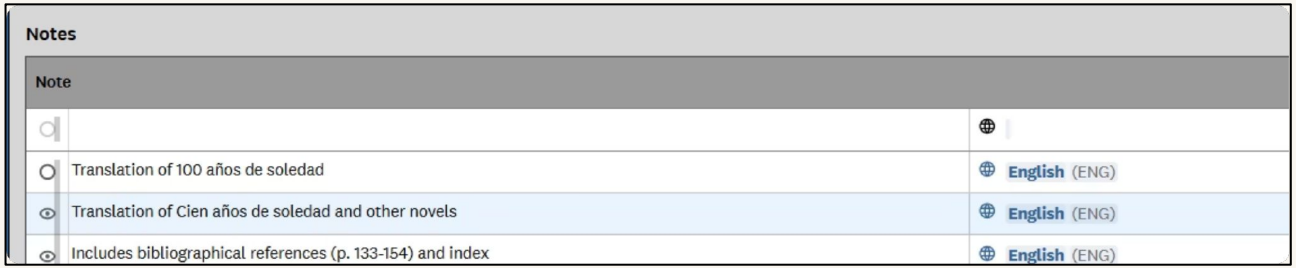


The above property statements, **has bf:note**, is considered multi-value property, in the sense that one or more values, for the same property, can be displayed on the entity page. Hence, since STANFORD, NLN, NYU and UPENN have associated three notes that have nearly identical meanings and one note with completely different content, the user, by setting the IsLeader value, indicates which notes, given the same language, they want to display on the portal.

Multi-Value Property — Portal View

The screenshots below show how the **has bf:note** multi-value property appears in JCricket. The public entity page (top) displays the notes selected as IsLeader, while the detailed edition statement and notes panel (bottom) reveals all contributed values across provenances, enabling catalogers to make informed decisions about display priority.





Policy 2: Adding a Property Value from a Controlled Vocabulary

When a property's value comes from a controlled vocabulary within SVDE, catalogers must understand both the vocabulary type and the display rules that govern how multiple values are rendered in the user portal.

Open Vocabulary

Catalogers **may add a new term** to the vocabulary if no existing term adequately represents the content. This requires care to avoid redundancy and maintain consistency across institutions.

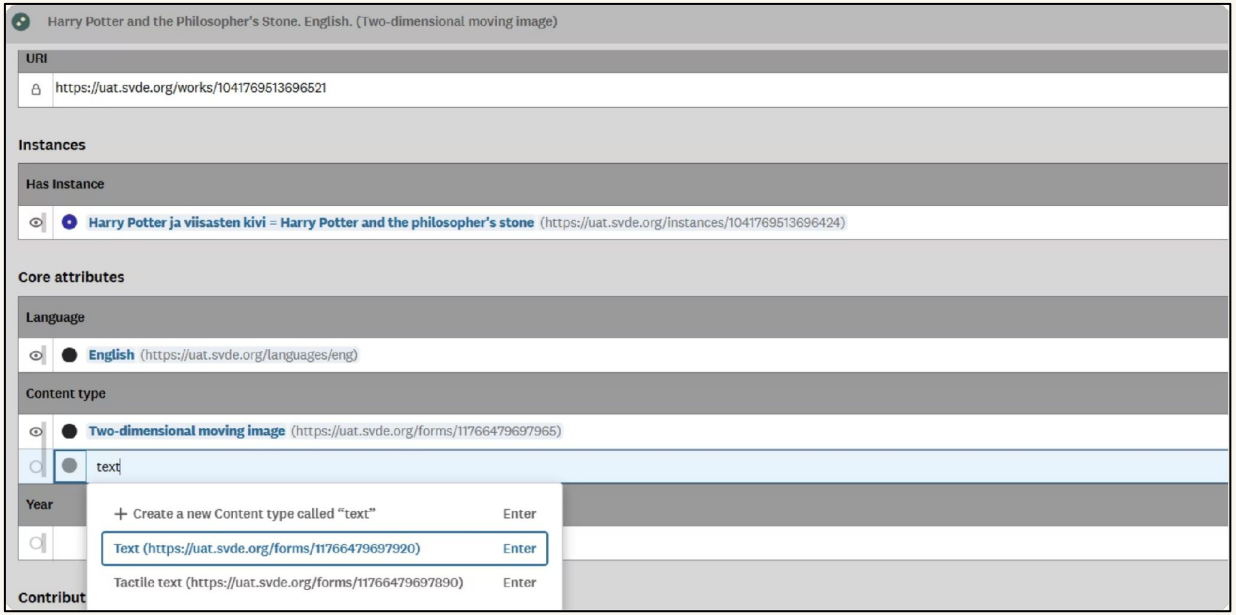
Closed Vocabulary

Catalogers **cannot add new terms** — only existing, pre-approved values may be selected. This enforces strict consistency but limits flexibility for edge cases.

Most controlled vocabulary properties support **multiple values**. For example, an instance with a primary content type of "text" may also carry "still image" if illustrations are substantial. Consortia must define clear policies determining which values appear in the portal — the most widely shared value, all values, or a curated selection based on community consensus.

Policy 2: Controlled Vocabulary in JCricket — System View

The screenshot below demonstrates how JCricket handles controlled vocabulary input. When a cataloger types a partial term (e.g., "text"), the system presents matching options from the SVDE vocabulary — including the option to create a new term if the vocabulary is open — along with linked URIs for each candidate value. This interface supports accurate, consistent value selection across contributing institutions.



The "Content type" field shown here is a controlled vocabulary property. Selecting the correct URI-linked value — rather than a free-text entry — ensures interoperability across all member institutions.

And More Policies...

Beyond the policies detailed above, the JCricket Best Practices framework addresses a broader set of shared cataloguing governance topics essential for maintaining data integrity across institutions.

01

Policy 3 — Constraints on Other Provenances

Define constraints for advanced users on attributes contributed by other Provenances, preventing unintended overwriting of authoritative data.

02

Policy 4 — Closed and Open Controlled Vocabularies

Establish which vocabularies are open (extensible) and which are closed (fixed), and govern how new terms may be proposed and approved.

03

Policy 5 — Criteria for Removing Entities

Set clear, community-agreed criteria for when an entity may be cancelled or merged, minimizing data loss and orphaned relationships.

04

Policy 6 — Criteria for Removing Entity Relationships

Define conditions under which typed relationships between entities may be safely removed without compromising catalogue integrity.

05

Policy 7 — Authoritative External Sources

Specify which external sources are considered authoritative for enriching entities or for lookup, ensuring consistent provenance and data quality.

Thank You!

Learn more about JCricket, SHARE-VDE, and the broader SHARE Family of projects through the resources below. We welcome questions, contributions, and collaboration from catalogers and data stewards across the community.

SHARE Family

share-family.org

SHARE Wiki

wiki.share-family.org

Contact Us

info@share-family.org

SVDE Portal

svde.org

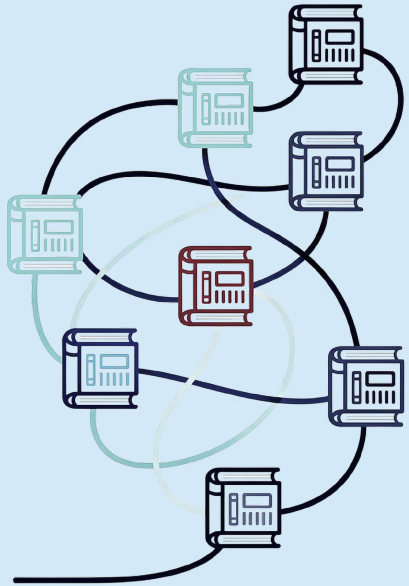
LOD Platform Distributed Development Community · Framework Agreement

LOD Platform DDC - Distributed Development Community

Goal of the DDC: enable member institutions and external interested parties to contribute to the development of extensions and integrations of the LOD Platform software, which serves as the technological platform for the Share Family initiative and is created under the guidance of the library community.

Benefits:

- encourage broader participation in the LOD Platform development process;
- disseminate the know-how within a wider community;
- support more sustainable use and long-term management of a more robust and consolidated LOD Platform.



Governance model

Starting Point

The LOD Platform is governed under a library-driven governance model, combining the long-term stewardship of Casalini Libri and @Cult with the collective insight and guidance of the international library community. This governance structure ensures that the platform evolves in alignment with the mission, values, and needs of research libraries while preserving the integrity, coherence, and sustainability of the core.

Components of the governance structure

- **Casalini/@Cult**, as owner and steward of the core platform;
- The **Share Family Advisory Council (AC)**, representing the institutional voice of the library community;
- The **Technical Governance Committee (TGC)**, responsible for architectural oversight and technical direction;
- The **Distributed Development Community (DDC)**, participating through structured contribution workflows.

Roles of the governance structure

The governance structure guarantees meaningful involvement of libraries in shaping the platform's evolution:

- **the Advisory Council** articulates institutional priorities and ensures platform alignment with the community vision.
- **the Technical Governance Committee** transforms those priorities into technical decisions and architectural guidelines.
- **Casalini/@Cult** implement and maintain the core components, guaranteeing long-term stability and coherence.

This engagement model ensures:

- broad **participation** without fragmentation,
- **community guidance** without dilution of technical responsibility,
- **platform sustainability** through shared governance and dedicated stewardship.

Contributor Workflow Simulation · External Contribution Analysis

Context & Objective

Why This Simulation?

Starting Point

The Framework Agreement for the Distributed Development Community (DDC) defines a governance model and contribution workflow: **RFC → Development → Review → Release**. The agreement sets the rules — this document proposes scenarios to apply those rules in practice.

Scope of Analysis

- Currently available resources: wiki, APIs, tooling
- Tools in the Casalini/@Cult ecosystem: Jira, Bitbucket, Keycloak
- People involved and their areas of responsibility
- Gaps that must be closed to make the process operational

The analysis is conducted by simulating a real partner institution attempting to contribute to the platform from scratch.

Chapter 1

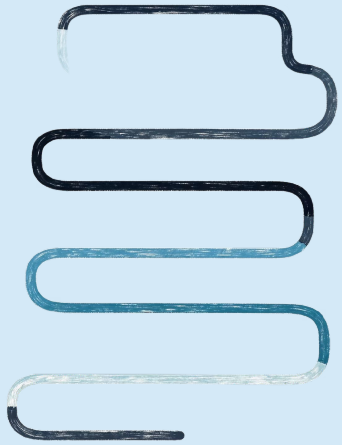
Contribution Scenarios: Three Maturity Levels

Contribution scenarios were identified across three maturity levels, spanning from fully autonomous integrations to aspirational contributions requiring the complete DDC infrastructure.

Level A — Feasible now
Achievable with existing resources and public documentation, with high contributor autonomy.

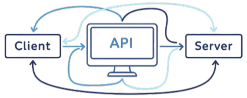
Level B — Feasible with fixes
Requires bridging identifiable gaps, eg. missing format specs and validation rules.

Level C — Aspirational
Requires the full DDC infrastructure: TGC operational, pluggable interfaces, core component access.



Level A and B Scenarios

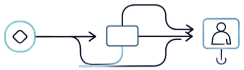
Achievable with Existing Resources



A1 · External API Consumer

Who: IT dept. of a Share Family institution
What: Tool/service querying public REST/GraphQL APIs to integrate linked data locally (harvester, catalog widget, RDF export, Sinopia integration)
RFC: Type A · **Autonomy:** High

⚠ Gaps: Credentials not self-service, base URL unpublished, no rate limiting, single-tenant token constraint



B1 · Format Transformer

Who: Institution with non-MARC data
What: Standalone tool converting records from the institution's format to the LOD Platform pipeline input format
RFC: Type A · **Autonomy:** Medium

⚠ Gaps: Input format spec, file delivery method, no validator

B2 · Pre-Ingest Validator

Who: IT dept. of a consortium (e.g. PALCI, PCC)
What: Standalone tool validating MARC records before ingest — detects missing fields, encoding errors, probable duplicates
RFC: Type A · **Autonomy:** Medium

⚠ Gaps: Pipeline-specific validation rules, acceptance thresholds, sample datasets

Level C Scenarios

Aspirational: Full DDC Infrastructure Required

C1 · Custom Authority Source Connector

Who: National library (e.g. BnF, DNB) · **What:** Add a new authority source to the Authify enrichment system

Authify is fully internal with no pluggable interfaces. Integration requires modifications to **3 core components**. This scenario best illustrates the gap between Framework Agreement outline and architectural reality.

Contributor autonomy: low.

C2 · Archival Ontology Extension

Who: University library with archival collections (e.g. Yale, Stanford, LoC) · **What:** Propose an SVDE ontology extension for archival material

RFC-B with impact across data model, ontology, RDFizer, and clustering. Requires a constituted and operational TGC and a formal impact analysis framework.

Contributor autonomy: low.

C3 · Domain-Specific ClusterKey (Music)

Who: National library with specialized music corpus · **What:** Variant of ClusterKey generation for musical entities, where standard rules produce false positives

RFC-B with direct impact on the clustering engine. Requires TGC, access to unexposed core logic, and precision/recall test datasets. **Contributor autonomy: none.**

Comparative Matrix

Scenarios at a Glance

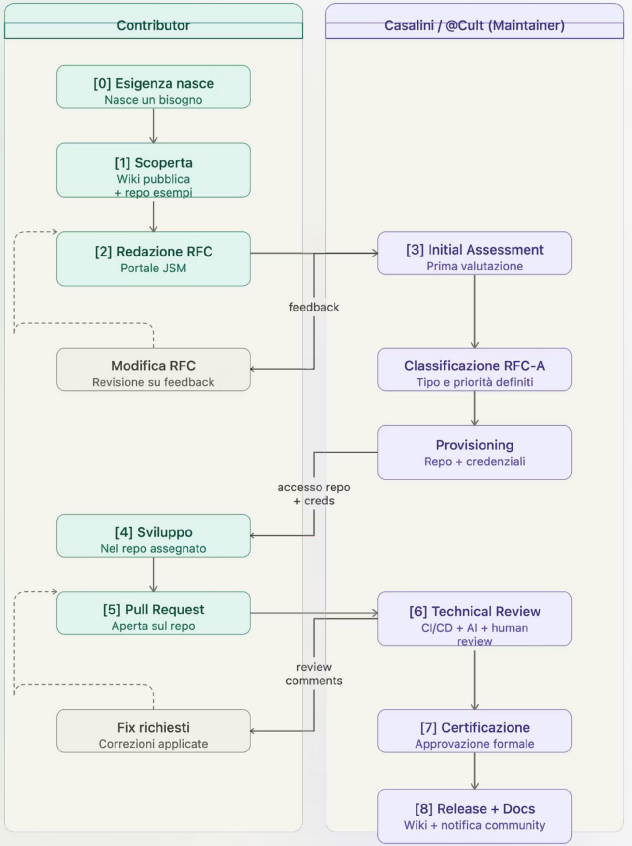
Scenario	Formal RFC	Effective RFC	Autonomy	Main Gaps
A1 — API Consumer	A	**A**	High	Credentials, base URL, rate limiting
B1 — Format Transformer	A	**A**	Medium	Input format specification
B2 — Pre-Ingest Validator	A	**A**	Medium	Validation rules, sample datasets
C1 — Authority Connector	A	B (de facto)	Low	Modifications to 3 core components
C2 — Ontology Extension	B	B	Low	TGC, ontology docs, process
C3 — ClusterKey Music	B	B	None	TGC, core access, test metrics

Key insight: The only truly autonomous scenarios (effective RFC-A) are A1, B1 and B2. All three produce artifacts running in the contributor's own infrastructure, interacting with the platform only via documented APIs or agreed file formats.

Chapter 2

End-to-End Contributor Flow

The workflow spans 9 phases (0–8) divided between the **Contributor** and the **Casalini/@Cult Maintainer**. The left lane shows contributor actions; the right lane shows maintainer responses, with handoffs at RFC submission, provisioning, PR creation, and release.



Flow Detail

Phase-by-Phase Breakdown

Phase 0 · Need Arises

Institution identifies a technical need (integrate Share-VDE data or feed the pipeline with a non-standard format). The DDC Framework Agreement provides the process entry URL.

Phase 1 · Discovery

Contributor accesses the public wiki: API docs, DDC onboarding process, approved examples, and submission portal link.

Phase 2 · RFC Submission

Contributor logs into the JSM portal on casalinigroup.atlassian.net. Two forms: Members use "Submit RFC"; non-members use "Request to enable DDC" (Advisory Council approval required).

Phase 3 · Assessment + Provisioning

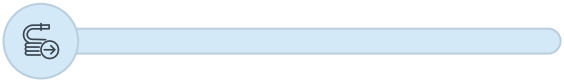
Maintainer team evaluates the RFC (RFC-A or RFC-B), communicates outcome. If approved: dedicated Bitbucket repository created, test API credentials released. One repo per contribution.

Phases 4–5 · Development & PR

Contributor develops in the assigned repo, tests autonomously (A1: against APIs; B1: against format spec), prepares documentation, then opens a Pull Request linked to the Jira RFC issue.

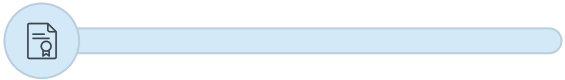
Flow Detail — continued

Review, Certification & Release



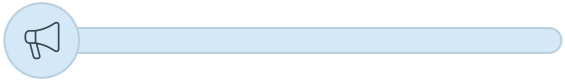
Phase 6 · Technical Review

Multi-layer review: **automated CI/CD** (build, test, security scan) + **AI-assisted review** + **human review with final decision**. External contribution CI/CD pipelines must be configured case-by-case.



Phase 7 · Certification

Formal approval. The contribution is included in the DDC approved tool catalog. For scenario A1, production API credentials are released at this stage.



Phase 8 · Release

DDC catalog updated on the wiki. Community notified via the appropriate communication channels. **The contributor is credited as author.** Approved repos are published as community examples.

Thank you!

https://share-family.org
https://wiki.share-family.org
info@share-family.org
https://svde.org