FOLIO meets Share-VDE
Collaborative Entity Management Workflows

WOLFCOn 2023 - David Rubenstein Forum, The University of Chicago, August 24th 2023

Andrea Gazzarini, Share-VDE Lead Architect

www.svde.org
info@svde.org
I, Andrea Gazzarini

Software Engineer (1999-)
“Hermit” Software Engineer (2010-)

Programming Passionate
Information Retrieval Passionate

Author of “Apache Solr Essentials”
Apache Qpid (past) Committer

Founder of SpazioCodice
Share-VDE Lead Architect

Husband & Father
Bass Player
Chapman Stick (aspiring) Player
Agenda

➢ The Share-VDE Initiative
➢ Sapientia: The Share-VDE Knowledge Base
➢ The Domain Model
➢ The Entity as a “Prism”
➢ Knowledge Base Management API: JCricket
➢ Outbound Connector Framework
➢ FOLIO Integration: Challenges
➢ FOLIO Integration: Use Cases
➢ FOLIO Integration: Available Options
➢ FOLIO Integration: Roadmap
The Share-VDE Initiative
Agenda

➢ The Share-VDE Initiative
  ○ What is Share-VDE?
  ○ History
  ○ The Share Family Linked Data Ecosystem
➢ Sapientia: The Share-VDE Knowledge Base
➢ The Domain Model
➢ The Entity as a “Prism”
➢ Knowledge Base Management API: JCricket
➢ Outbound Connector Framework
➢ FOLIO Integration: Challenges
...
Share-VDE: Share Virtual Discovery Environment

In a Nutshell

Share-Virtual Discovery Environment is a library-driven initiative which brings together, in a shared discovery environment, the bibliographic catalogues and authority files of a growing number of leading academic and national libraries from across North America and Europe.
The Share Family is a global community built on collaboration that brings together libraries, archives, museums, consortia and Library Service Platforms (LSP) and joins their knowledge in an ever-widening network of interconnected bibliographic data.

For further details please refer to [https://www.share-family.org](https://www.share-family.org) and the dedicated Share-VDE wiki section.
Sapientia: The Share-VDE Knowledge Base
Agenda

➢ The Share-VDE Initiative
➢ **Sapientia: The Share-VDE Knowledge Base**
  ○ Tenants and Provenances
  ○ Sapientia: Genesis
  ○ Record-Level Provenance
➢ The Domain Model
➢ The Entity as a “Prism”
➢ Knowledge Base Management API: JCricket
➢ Outbound Connector Framework
➢ FOLIO Integration: Challenges

...
Tenants and Provenances

Share-VDE manages a Knowledge Base which consists of clustered, integrated and enriched entities.

In Share-VDE, a tenant is represented by a set of institutions contributing to the same Knowledge base.

An institution within a tenant is called provenance. We use that term because we want to retain the relationship between Share-VDE entities and data that originally contributed to them.
Sapientia: Genesis

Knowledge Base (Sapientia)
Record-Level Provenance

Each record coming from a provenance contributes in building/enriching one or more Share-VDE entities.

A Share-VDE entity can be seen as a prism where each face represents data coming from a given provenance.

Each Share-VDE cluster maintains a link to the records it originated from.
The Domain Model
Agenda

- The Share-VDE Initiative
- Sapientia: The Share-VDE Knowledge Base
- **The Domain Model**
  - Core Entities
  - Agents, Contributions
  - Subjects
  - Non-core Entities
- The Entity as a “Prism”
- Knowledge Base Management API: JCricket
- Outbound Connector Framework

...
The highest level of abstraction in Share-VDE, an Opus is an entity that permits the grouping of works that are considered functional or near equivalents. The Opus is defined by a constellation of elements that form the shared content of works.

A work is a resource reflecting the conceptual essence of a cataloguing resource. A work is defined by a constellation of elements representing the specific intellectual or artistic form that an Opus takes each time it is "realized".

Instances and Items have the exact same meaning of the corresponding entities in BIBFRAME.
Agents, Contributions
Subjects
Non-core Entities

Share-VDE manages as Prisms other non-core entities, too. Some example:

- Places
- Formats
- Languages
- Availability
The Entity as a “Prism”
Agenda

- The Share-VDE Initiative
- Sapientia: The Share-VDE Knowledge Base
- The Domain Model
- The Entity as a “Prism”
  - From Library Data to Sapientia
  - Prism, faces: the Share-VDE Entity
  - Faces (aka Contributions)
  - Properties: Attributes, Relationships, Links
- Knowledge Base Management API: JCricket
- Outbound Connector Framework

...
FOLIO instance data is split across the entities that form the Share-VDE domain model. In this example we focus on the properties that are assigned to a Share-VDE instance (red triangle above).

A Share-VDE member (Stanford, in the example) uses FOLIO for managing its data.
Prism, faces: the Share-VDE Entity
Faces (aka Contributions)

<table>
<thead>
<tr>
<th>Library</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIVERSITY OF ALBERTA LIBRARY</td>
<td></td>
</tr>
<tr>
<td><strong>title</strong></td>
<td>Alice in wonderland</td>
</tr>
<tr>
<td><strong>titleAlternative</strong></td>
<td>Alice’s adventures under ground</td>
</tr>
<tr>
<td><strong>author</strong></td>
<td><a href="https://svde.org/people/201">https://svde.org/people/201</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stanford</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>title</strong></td>
</tr>
<tr>
<td><strong>titleAlternative</strong></td>
</tr>
<tr>
<td><strong>author</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>National Library of Norway</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>title</strong></td>
</tr>
<tr>
<td><strong>titleAlternative</strong></td>
</tr>
<tr>
<td><strong>author</strong></td>
</tr>
</tbody>
</table>

**links**

<table>
<thead>
<tr>
<th>sameAs</th>
<th><a href="http://dbpedia.org/resource/Alice%2527s_Adventures_in_Wonderland">http://dbpedia.org/resource/Alice%2527s_Adventures_in_Wonderland</a></th>
<th>Dbpedia</th>
</tr>
</thead>
<tbody>
<tr>
<td>sameAs</td>
<td><a href="https://www.wikidata.org/wiki/Q189875">https://www.wikidata.org/wiki/Q189875</a></td>
<td>Wikidata</td>
</tr>
<tr>
<td>sameAs</td>
<td><a href="https://data.bnf.fr/ark:/12148/cb358500385#about">https://data.bnf.fr/ark:/12148/cb358500385#about</a></td>
<td>bnf</td>
</tr>
</tbody>
</table>
Properties: Attributes, Relationships, Links

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Provenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>Alice in wonderland</td>
<td></td>
</tr>
<tr>
<td>titleAlternative</td>
<td>Alice's adventures under ground</td>
<td></td>
</tr>
<tr>
<td>titleAlternative</td>
<td>Journeys in Wonderland</td>
<td></td>
</tr>
</tbody>
</table>

An attribute is a data property, having a literal as value

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Provenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>sameAs</td>
<td><a href="http://dbpedia.org/resource/Alice%27s_Adventures_in_Wonderland">http://dbpedia.org/resource/Alice%27s_Adventures_in_Wonderland</a></td>
<td>Dbpedia</td>
</tr>
<tr>
<td>sameAs</td>
<td><a href="https://www.wikidata.org/wiki/Q189875">https://www.wikidata.org/wiki/Q189875</a></td>
<td>Wikidata</td>
</tr>
<tr>
<td>sameAs</td>
<td><a href="https://data.bnf.fr/ark:/12148/cb358500385#about">https://data.bnf.fr/ark:/12148/cb358500385#about</a></td>
<td>bnf</td>
</tr>
</tbody>
</table>

A link is a connection between a Share-VDE Prism and an external reference

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Provenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>author</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A relationship is a connection between two Share-VDE Prisms
Knowledge Base Management API: JCricket
Agenda

➢ The Share-VDE Initiative
➢ Sapientia: The Share-VDE Knowledge Base
➢ The Domain Model
➢ The Entity as a “Prism”
➢ Knowledge Base Management API: JCricket
  ○ The Big Picture: Genesis, Search, Edit
  ○ JCricket: Available Operations
  ○ JCricket UI
➢ Outbound Connector Framework

...
The Big Picture: Genesis, Search, Edit

Data flows into Share-VDE from libraries, institutions and third-party sources (e.g. VIAF, ISNI, FAST)

The Share-VDE knowledge base (*Sapientia*) contains the integrated/clustered/enriched entities.

Data is searchable through the entity discovery portal.

Data is mainly edited through JCricket, the Share-VDE entity editor.

Any third-party, authorized application can be a Share-VDE client

Third-party Applications

but...

but...

Edit API (GraphQL)

Search API (GraphQL / REST)
JCricket: Available Operations

**Edit:** a property is added/updated/deleted

Lewis Carroll → Lewiss Carroll

is author of


**Invalidate**

**Merge:** multiple prisms are merged into one

For example, two prisms, “Mark Twain” and “Samuel Clemens”, should be actually part of the same entity.

**Split:** a prism is split into multiple prisms

A prism (wrongly) contains information belonging to multiple entities (e.g., “Wallace David” and “David Wallace”)

29
Step 1: Search on Share-VDE Main Portal

Step 2: Enter the entity details page

Step 3: Click on the edit action and enter the JCricket UI
Outbound Connectors Framework
Agenda

- The Share-VDE Initiative
- Sapientia: The Share-VDE Knowledge Base
- The Domain Model
- The Entity as a “Prism”
- Knowledge Base Management API: JCricket
- Outbound Connector Framework
  - The Architecture
- FOLIO Integration: Challenges
- FOLIO Integration: Use Cases
- FOLIO Integration: Available Options
- FOLIO Integration in Share-VDE: Roadmap
The Architecture

Data flows into Share-VDE from libraries, institutions and third-party sources (e.g. VIAF, ISNI, FAST)

The Share-VDE knowledge base (Sapientia) contains the integrated/clustered/enriched entities.

Data is mainly edited through JCricket, the Share-VDE entity editor.
FOLIO Integration: Challenges
Agenda

➢ The Share-VDE Initiative
➢ Sapientia: The Share-VDE Knowledge Base
➢ The Domain Model
➢ The Entity as a “Prism”
➢ Knowledge Base Management API: JCricket
➢ Outbound Connector Framework
➢ **FOLIO Integration: Challenges**
  ○ #1: Different Domain Model, Entities, Properties
  ○ #2: Use Cases
  ○ #3: Data Consistency
➢ **FOLIO Integration: Use Cases**
...

...
Micro-services based, FOLIO has a flexible, modular and extensible domain model.

More than one domain model, it’s like a set of interconnected micro-domain-models, therefore able to capture a vast domain space.

The Domain Model includes both bibliographic (e.g. Instances, Items) and management entities (e.g. Loan, Orders, Invoices).

Share-VDE inherits from, and extends the BIBFRAME Domain Model.

Everything is an entity (e.g., Works, Agents, Places, Formats, even properties).

Entities are called Prisms composed by multiple faces, each one representing the data contributed by a given library/institution.

Entities are mainly created through a clustering process where deduplication, integration, enrichment play a crucial role.

The core domain model doesn’t not contains anything outside the bibliographic sphere.
Challenge #1: Different Entities

- Instance
  - has
  - Contributor
    - name: string
  - Publication
    - place: string
    - date: date
    - role: string

- ContributorType
  - has
  - Relator code (e.g. aut, ill)

- Corporate Name
- Meeting Name

- Enum
  - Name Type
    - Personal Name
    - Corporate Name
    - Meeting Name

- Contribution
  - has
  - Role
    - https://svde.org/rc/aut

- Agent
- Person
- Family
- Meeting
- Organization
Challenge #1: Different Entities
Challenge #1: Different Properties
Challenge #1: How can we address that?

Share-VDE configuration allows to associate a provenance (i.e., a member) to one or more (external) data provider systems (e.g. ILS). In the simplest scenario there's just one system; however, a member could be in a context where data is managed by multiple systems.

Other than the association, the configuration stores the coordinates of each target system (e.g., protocol, endpoints, credential access)
Challenge #1: Reverse Mapping
Challenge #2: “Unusual” Use Cases

Split

Merge

Invalidate

icons from vecteezy.com
Challenge #3: Data Consistency

The **async nature** of the communication between the two systems makes **very hard** to avoid at all the consistency issue. Things get even worse if multiple users/apps insist on the same entities, on both sides.
Challenge #3: How can we address?

Real-time inbound connectors + Stream Clustering

Synchronous communication?

Winner defined in config

icons from vecteezy.com
Use Cases
Agenda

➢ The Domain Model
➢ The Entity as a “Prism”
➢ Knowledge Base Management API: JCricket
➢ Outbound Connector Framework
➢ FOLIO Integration: Challenges
➢ **FOLIO Integration: Use Cases**
  ○ Edit
  ○ Invalidate
  ○ Merge
  ○ Split

...
“Edit” Property (Value)

A property value is added(updated/deleted according to its formal definition in the Share-VDE dictionary (e.g., type, datatype, cardinality, mandatory or not)

A property can be an attribute (a literal value), a relationship (a connection between two prisms) or a link (a connection to an external URL/URI)
**Edit: How does it work in Share-VDE?**

Share-VDE is able to **detect** and **signal** a change that occurs at **property-value level**.

When such an event occurs, a notification is sent to **Chronos**, an internal **service bus** component that implements a **publish-subscribe** messaging model.
The integration with a system that does not have a public “identity” for each property value can be challenging because a given property value instance in Share-VDE cannot be easily correlated with the corresponding value instance in the target system.

Share-VDE provides one way to mitigate the issue by means of a Memento pattern: change signals include the old and the new value for the affected property; it helps in correlating the value instances between the two systems.

Even using the above, the consistency issue is mitigated, not completely solved (e.g., parallel editing on both sides).

Actually, the async nature of the communication between the two systems makes very hard to avoid at all the consistency issue.
Edit Demo: API Requests

Postman Collection
https://tinyurl.com/folio-requests

Postman Collection
https://tinyurl.com/sharevde-requests
Edit Demo: FOLIO Instances
Edit Demo: From FOLIO To Share-VDE
Edit Demo: Share-VDE Entities

Alice’s adventures in Wonderland

Novel by Lewis Carroll. Created in 1865.

Alice’s Adventures in Wonderland is an 1865 English children’s novel by Lewis Carroll, a mathematics don at Oxford University. It details the story of a young girl named Alice who falls through a rabbit hole into a fantasy world of anthropomorphic creatures. It is seen as an example of the literary nonsense genre. The artist John Tenniel provided 42 wood-engraved illustrations for the book. – Wikipedia

More options

Publications of this

5 results

<table>
<thead>
<tr>
<th>No.</th>
<th>Title</th>
<th>Language</th>
<th>Publication year</th>
<th>Format</th>
<th>Contributors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lewis Carroll’s Alice’s adventures in Wonderland</td>
<td>English</td>
<td>1982</td>
<td>Volume</td>
<td>Stephen Dodgson (Illustrator)</td>
</tr>
<tr>
<td>2</td>
<td>Aventures d’Alice au pays des merveilles</td>
<td>French</td>
<td>1972</td>
<td>Volume</td>
<td>Alfred Ludlow Carroll (contributor)</td>
</tr>
<tr>
<td>3</td>
<td>Aventures d’Alice au pays des merveilles</td>
<td>French</td>
<td>1869</td>
<td>Volume</td>
<td>Alfred Ludlow Carroll (contributor)</td>
</tr>
</tbody>
</table>
Edit Demo: Enter in JCricket

The editor changes the publication title to “Alice nel Paese delle meraviglie”
Edit Demo: Enter in JCricket
The Knowledge Base does not perform deletes by removing data. Instead, prisms are invalidated, meaning they are marked with tombstones. Invalidated Prisms are no longer searchable in the public portal; however, they can be still managed through JCricket API.

A prism can be invalidated explicitly, or implicitly/indirectly, when it’s part of a merge context.

Explicit Invalidation

Implicit Invalidation (Merge)
**Merge** *n* Prisms into One

Multiple Prisms are merged into one. The “merge” result is called “destination” prism. The editor can choose which properties move between sources and destination. After merging, source prisms are implicitly invalidated.

Merge operates at Prism level, meaning that it affects all prism “faces” (i.e. data belonging to all provenances that contributed to that prism). For that reason the user must have special grants (advanced editing); in addition, the whole operation must be reviewed/approved by another advanced editor.

Sources → One prism is marked as destination. After merging, it will be the only survivor → After merging, sources are invalidated.
**Split One Prism into n Prisms**

One prism is split into multiple entities (new or existing). In the Split context, the source is called giver, the splits are called receivers.

Split operates at **Prism level**, meaning that it affects all prism “faces” (i.e. data belonging to all provenances that contributed to that prism). For that reason the user must have special grants (advanced editing); in addition, the whole operation must be reviewed/approved by another advanced editor.

Giver → Receivers can be existing or new prisms.
Challenge: What Should We Send Back?

- **Split**
- **Merge**
- **Invalidate**
Share-VDE<->FOLIO Integration: Scenarios
Agenda

➢ Knowledge Base Management API: JCricket
➢ Outbound Connector Framework
➢ Third Parties Integration Challenges
➢ FOLIO Integration: Use Cases
➢ FOLIO Integration: Options/Scenarios
  ○ JCricket UI + FOLIO Connector
  ○ JCricket UI in FOLIO
  ○ Another JCricket UI in FOLIO
  ○ Online Stream Clustering
  ○ Offline Batch Clustering

...
The **Share-VDE editor** accesses the **JCricket UI** and starts changing Prisms. As far as changes are related to a **provenance** connected to **FOLIO**, the **outbound connector** asynchronously **propagates** the change.

![Diagram showing the integration of JCricket and FOLIO](image-url)
A JCricket native FOLIO App that communicates with Share-VDE GraphQL API and propagates (back) the changes to FOLIO through the Share-VDE outbound connector.
A JCricket native FOLIO App that synchronously updates the FOLIO Inventory and then uses the Share-VDE GraphQL API to propagate the changes into the Share-VDE Knowledge Base.
FOLIO (Inventory) feeds Share-VDE through a GraphQL (or a REST) channel which triggers an online Stream Clustering that updates the Share-VDE Knowledge Base.
FOLIO feeds Share-VDE periodically with data pertaining to a given period. That could be the whole system dataset (e.g., very beginning, disaster recovery) or the delta that captures the changes occurred in a given period (e.g., a week, a day, a month).
FOLIO Integration: Roadmap
Agenda

➢ The Share-VDE Initiative
➢ Sapientia: The Share-VDE Knowledge Base
➢ The Domain Model
➢ The Entity as a “Prism”
➢ Knowledge Base Management API: JCricket
➢ Outbound Connector Framework
➢ FOLIO Integration: Challenges
➢ FOLIO Integration: Use Cases
➢ FOLIO Integration: Options/Scenarios
➢ **FOLIO Integration: Roadmap**
**JIRA Timeline (in progress)**

- **Description**: The epic captures the work needed for integrating Folio with Share-VDE (most probably only the JCriblet module).

- **Child issues**:
  - [4047-570] Create a Test Dataset with FOLIO instances
  - [4047-572] Choose changes
  - [4047-573] Mapping between Folio & CKB Domain Models
  - [4047-574] Folio Instance shape depending on the CKB change scenario
  - [4047-575] Mapping between library (member) in Share-VDE and FOLIO instance
  - [4047-576] Update Workflow between J.Criblet and FOLIO (WRITE path)
  - [4047-577] Update Workflow between FOLIO and J.Criblet (READ path)
  - [4047-578] Async Module: FOLIO Update Processor
  - [4047-452] create parser in the shareware-robos module for Folio mapping
  - [4047-579] Create new FOLIO connector module
  - [4047-580] configuration for mapping a Share-VDE provenance with FOLIO external system
  - [4047-581] Create an adapter that implements the mapping defined in the configuration for mapping a Share-V
  - [4047-582] update to the existing signaling framework
  - [4047-583] Create a connector component that rebuilds FOLIO instances
  - [4047-584] Install & Drive into FOLIO
  - [4047-585] Move/Port the sync components in core api to Chronos
  - [4047-586] FOLIO Schema (i.e. properties metadata)
Get involved!

Becoming part of the global Share Family means sharing data and cooperating with the greater international library community.

The family continues to expand as more and more libraries worldwide embrace the opportunity to be involved in an international network of information, creating dialogue, participation and partnership.

Get in touch with us to find out more about how the Share family can help your library:

- info@svde.org
- https://wiki.svde.org/
- https://www.svde.org/about/about-share-vde
FOLIO meets Share-VDE
Collaborative Entity Management Workflows

WOLFCOn 2023 - David Rubenstein Forum, The University of Chicago, August 24th 2023

Andrea Gazzarini, Share-VDE Lead Architect

www.svde.org
info@svde.org