Share Family tenant infrastructure

Core principles

- Redundancy of data is complex to manage
- Linking entities is easier than duplicate data
- Cooperate and maintain autonomy at the same time
- Homogeneity of datasets and possible services to be shared
- Centralize core data through a lightweight method
- Distribute the technologic load to achieve long-term sustainability
- Profile levels of cooperation among systems and initiatives

Solution in the Share Family architecture

Creation of more branches in the Share Family, named tenants

Consistent groups of institutions gathered by similar scope or from the same domain:

Share-VDE
Share-Catalogue
Kubikat-LOD
PCC data pool
Parsifal project (network of ecclesiastical university libraries in Rome)
National bibliographies Group

What is a tenant

Definition of tenant from Wikipedia:

- "The term software multitenancy refers to a software architecture in which a single instance of software runs on a server and serves multiple tenants".

- "A tenant is a group of users who share a common access with specific privileges to the software instance. With a multitenant architecture, a software application is designed to provide every tenant a dedicated share of the instance - including its data, configuration, user management, tenant individual functionality" etc.

Benefits

More efficient data management

Technological sustainability

Dedicated applications and services tailored to the institutions members of the various branches

From the users perspective this enables richer and specialized sets of resources to be consulted

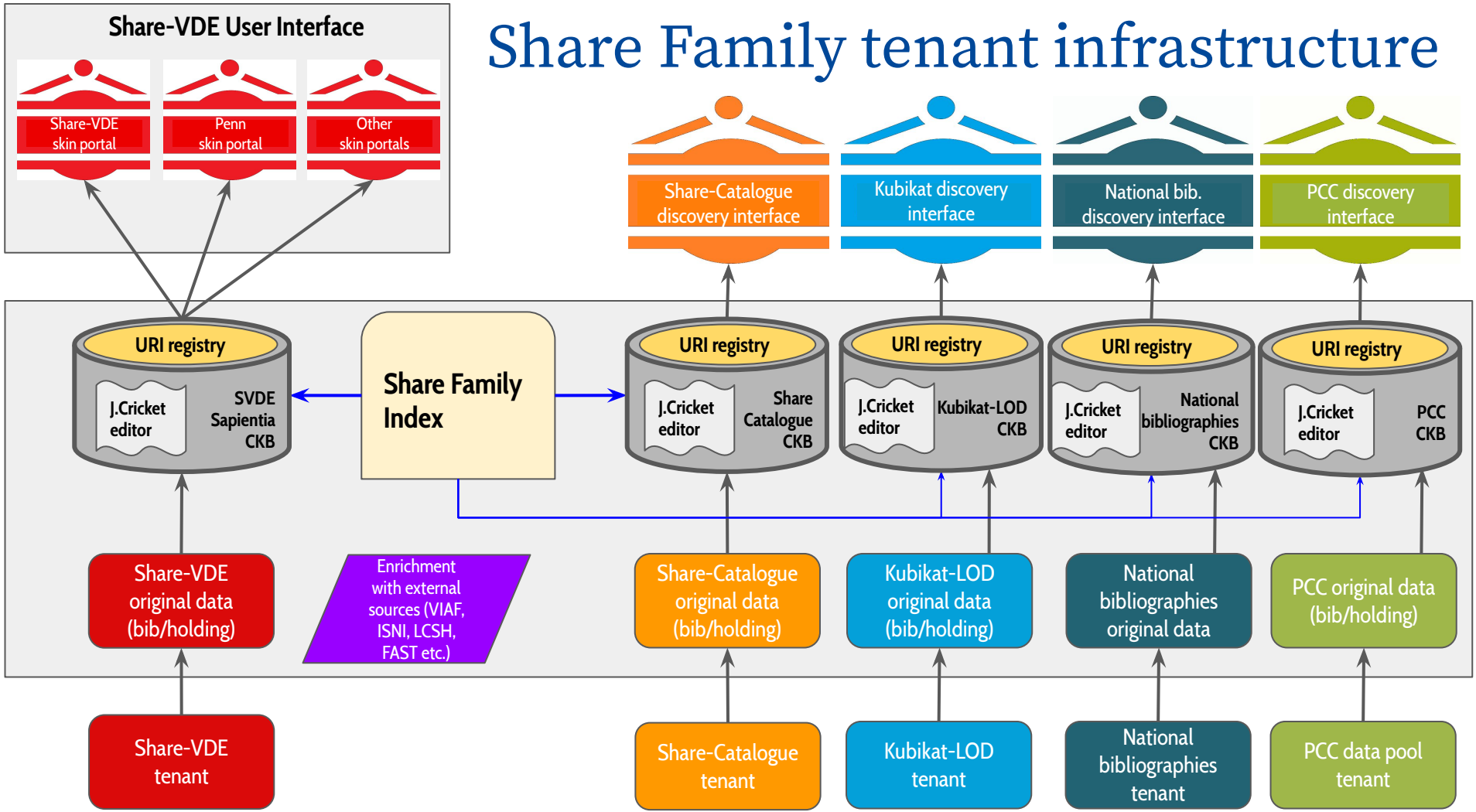
Main components of the Share Family tenants

Each tenant of the Share Family will have its own components and data will live in autonomous storages.

Each tenant has:

- its own CKB (e.g. Sapientia CKB, PCC CKB, Kubikat-LOD CKB, URBE CKB etc.)
- its own Cluster (URI) Registry with specific namespace
- its own J.Cricket CKB editor (= entity editor to manually update linked data entities)
- its own datastores (i.e. RDBMS, Search Engine and RDF Store)
- its own discovery portal with one or more skins (skin = filter on a part of tenant whole data)
    - e.g. general SVDE discovery portal skin https://svde.org + other ad hoc skins such as https://penn.svde.org/)

Share Family tenant infrastructure



Share Family Index and Share Family Identifier

SFI - Share Family Index

- index that centralizes core entity data from each CKB (e.g. entities URIs and very few data only for search and redirection purposes)
- central index able to point to all the URIs in all the CKBs of the different tenants

SFId - Share Family Identifier

- each entity has a unique URI within the tenant's CKB namespaces
- all the URIs for the same entity in different namespaces are grouped under a unique Share Family Identifier in the Share Family Index
- the SFId is a unique identifier linking to URIs that identify the entities in each CKB
- the SFId carries the minimum amount of data needed to identify the entities

Share Family Index and Share Family Identifier

E.g. Ernest Hemingway URIs in different CKB namespaces and the corresponding Share Family Identifier in the Index (the following URIs are for simulation purposes):

```
http://sfi/agents/456789 [Share Family Identifier]
    sameAs
https://svde.org/agents/101631288986955
    sameAs
https://svde.org/pcc/agents/7890123
    sameAs
http://kubikat-lod.org/agents/456789
```

Central orchestration: the SFI - Share Family Index

The Share Family Index is:

- a registry which is in charge to create and assign a unique **Share Family Identifier (SFId)**
- a central index that aggregates URIs for entities that are stored in different CKBs
- a metasearch engine able to run queries in all the Share Family tenants
- an orchestrator of queries and messages between tenants according to service user agreements/profiles, functioning similarly to the ESB - Enterprise Service Bus
- a pointer to entities URIs in the individual CKBs of the different tenants

The Share Family Index is not:

- a database/storage
- a CKB
- a Search Engine

Interaction through the Share Family Index

The SFI orchestrates among all tenants the communication of the changes done within an individual CKB

CKBs are decoupled: they never interact directly each other, they are always intermediated by SFI

The SFI acts as an Enterprise Service Bus (ESB) - it provides routing, transformation, propagation, policies, search services across the interconnected CKB instances.

Examples of interaction use cases

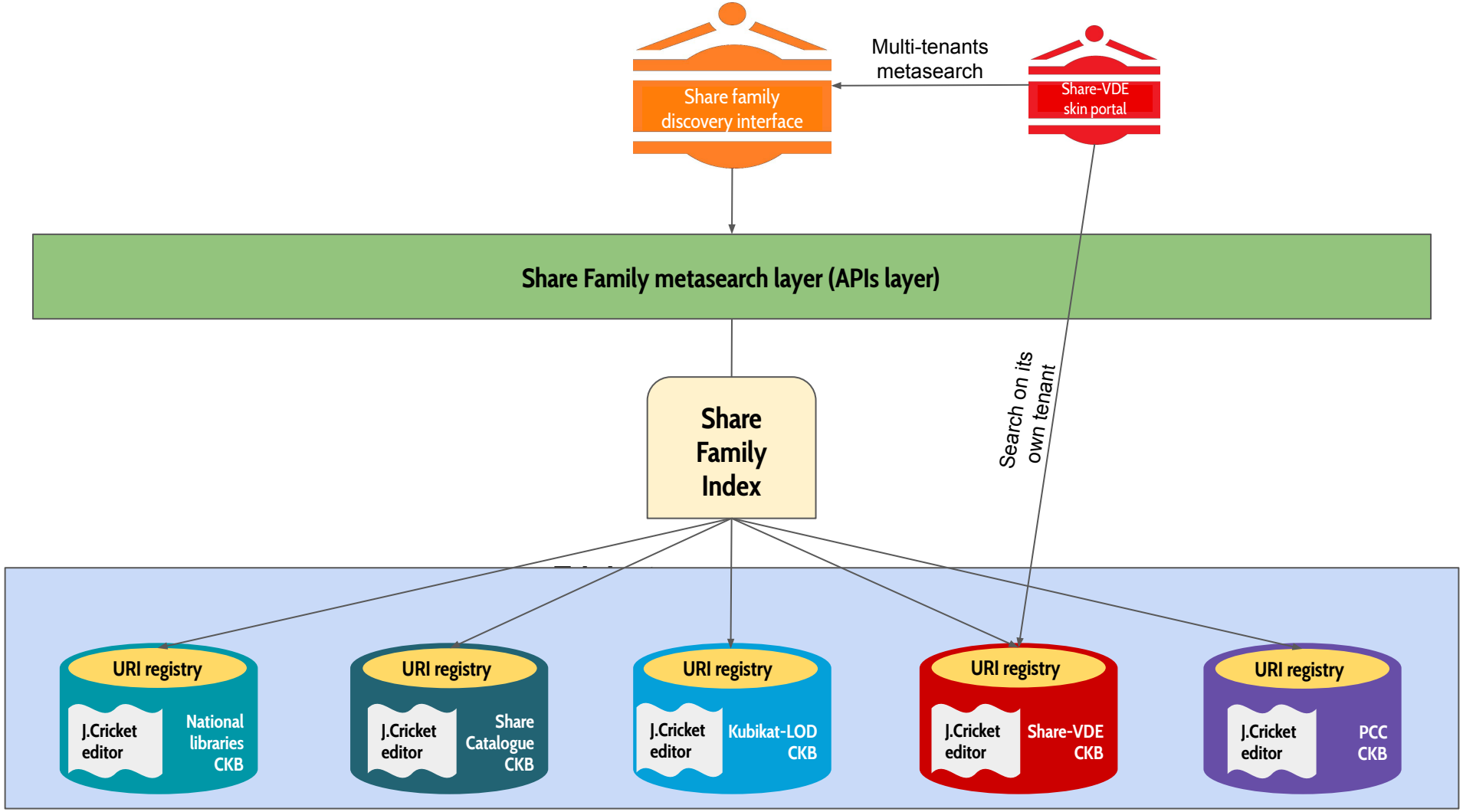
1. Change alert only: two (or more) tenants hold the same entity; *tenant 1* receives a notification that some data of the entity has been changed in *tenant 2*.

2. Propagate changes done to entity data: two (or more) tenants hold the same entity; *tenant 1* changes some data of the entity and propagates the changes to *tenant 2*.
   - Example: A library using Sinopia creates an Instance that already exists in the PCC data pool of SVDE
     - Scenario 1: every time an entity undergoes changes, the SFI receives push notifications from Sinopia and vice versa
     - Scenario 2: periodic triggers unsolicited to collect and register changes

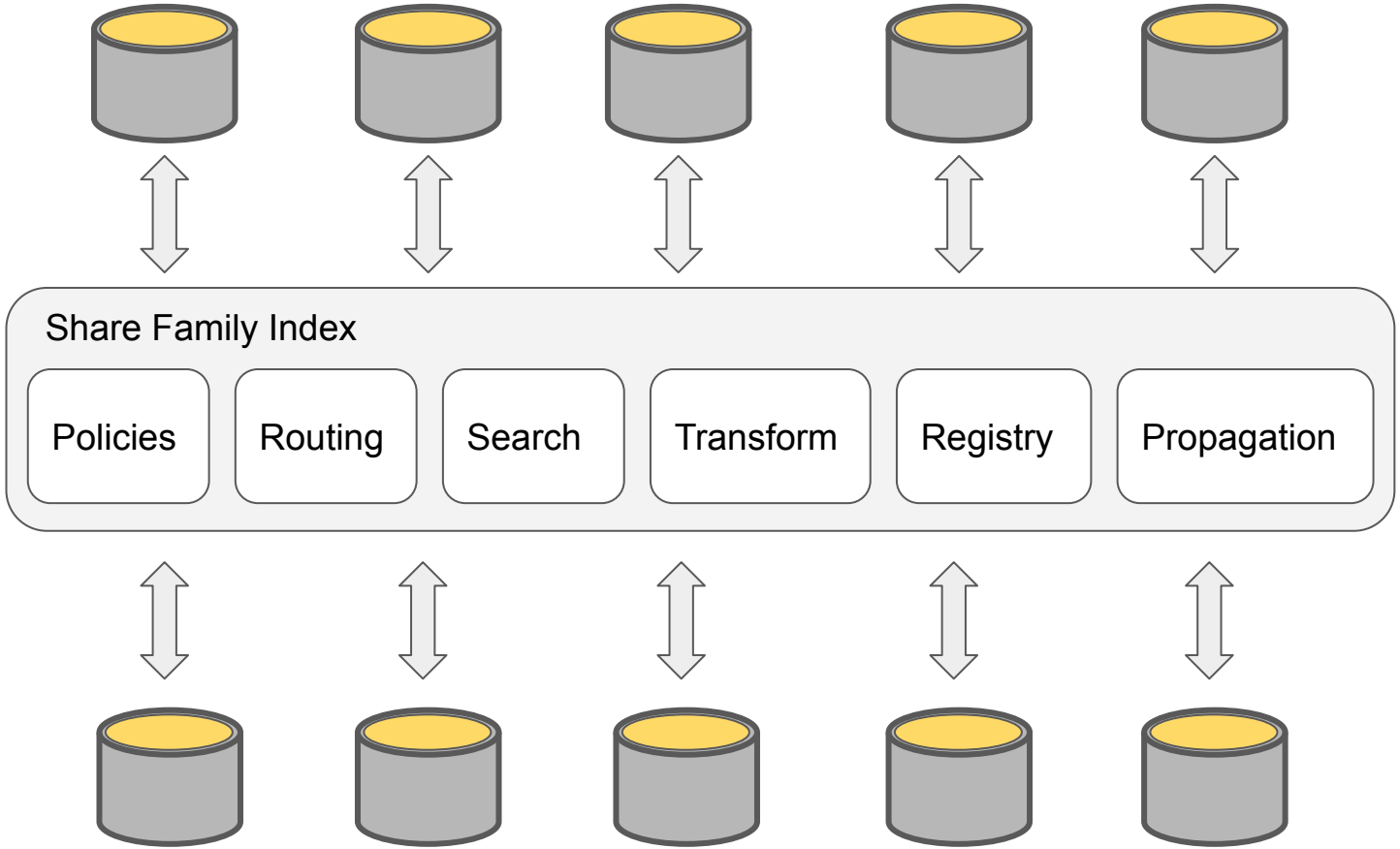
Search mechanism

The SFI search layer sits on top of all tenants; it allows simultaneous searches across them (metasearch or federated search).

This function is the opposite of a traditional skin, which allows to configure searches on subsets of data (example of a skin: https://penn.svde.org/ is a skin that filters Penn data from the whole Sapientia CKB where Penn data is included).

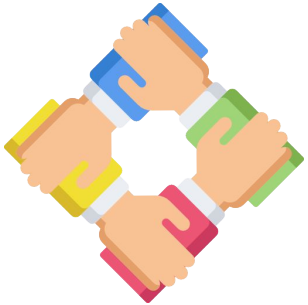

Share family discovery interface
Multi-tenants metasearch
Share-VDE skin portal
Share Family metasearch layer (APIs layer)
Share Family Index
Search on its own tenant
URI registry
J.Cricket editor
National libraries CKB
URI registry
J.Cricket editor
Share Catalogue CKB
URI registry
J.Cricket editor
Kubikat-LOD CKB
URI registry
J.Cricket editor
Share-VDE CKB
URI registry
J.Cricket editor
PCC CKB

Share Family Index: an ESB++ for the tenants



Levels of cooperation

Consensus from the groups of Institutions involved



Ad hoc agreements to set up among the communities/groups of institutions to establish Service User Profiles