Linking the (un)linked data through backlinks

Michalis Stefanidakis[1] and Ioannis Papadakis[2]
[1]Department of Computer Science, Ionian University, Greece
[2]Department of Archives and Library Science, Ionian University, Greece

Abstract

This paper proposes a framework capable of enhancing the interlinking between Linked Open Data – LOD nodes. Such interlinking is achieved through the realization of **backlinking**. Backlinking refers to the **consistent management** of the **references** that are made to an entity within a LOD node from other nodes through their triples.

Introduction

Every LOD node stores triples that contain its own **"defined"** entities, as well as **"referenced"** entities, defined in remote repositories. A triple containing a "referenced" entity constitutes a **"forward link"** to the remote node that defines the corresponding entity.

Symmetrically to the above, LOD nodes should be able to consistently store external references to the entities they define, as **"backlinks"**. Backlinks should ideally be created and maintained in a fully automated manner. This requires a framework for the reception and storage of backlinks.

Some research efforts that are targeted towards aggregative LOD services employ the notion of a **catalog of remote endpoints**, in order to broadcast a SPARQL query about a given resource (e.g. [1]).

In an effort to mimic the online community sites' **pingback mechanisms**, the authors of [3] suggest that whenever a reference to a remote resource is made from a local LOD repository, the corresponding remote server should be notified and update its underlying repository with accordingly produced triples.

The authors of [2] rely on a **crawling service** in order to discover backlinks within the linked data of the UK Public Sector Information domain. This reproduces the traditional large-scale search engine Web paradigm.

The proposed framework

The proposed framework achieves bidirectional linking betwen LOD nodes in a fully automated manner.

- For each "defined" entity within a LOD node that is referenced in other remote nodes, a **table of backlinks** is created in an adequate indexing structure, called thereafter ***the registry***.
- The rows of the table store the **SPARQL endpoints** of all remote nodes that reference the entity.
  - This way, each LOD node knows the number of references of its defined entities and the way to access such references.
- Upon creation of a triple containing a referenced entity in a LOD node, the remote LOD node owning the entity is notified with a **backlinking notification** that a reference is made to one of its entities.
  - The proposed framework dutifully **records the backlink** but **does not interpret its semantics**: this is a task of the yet-to-come applications that will be using the framework.

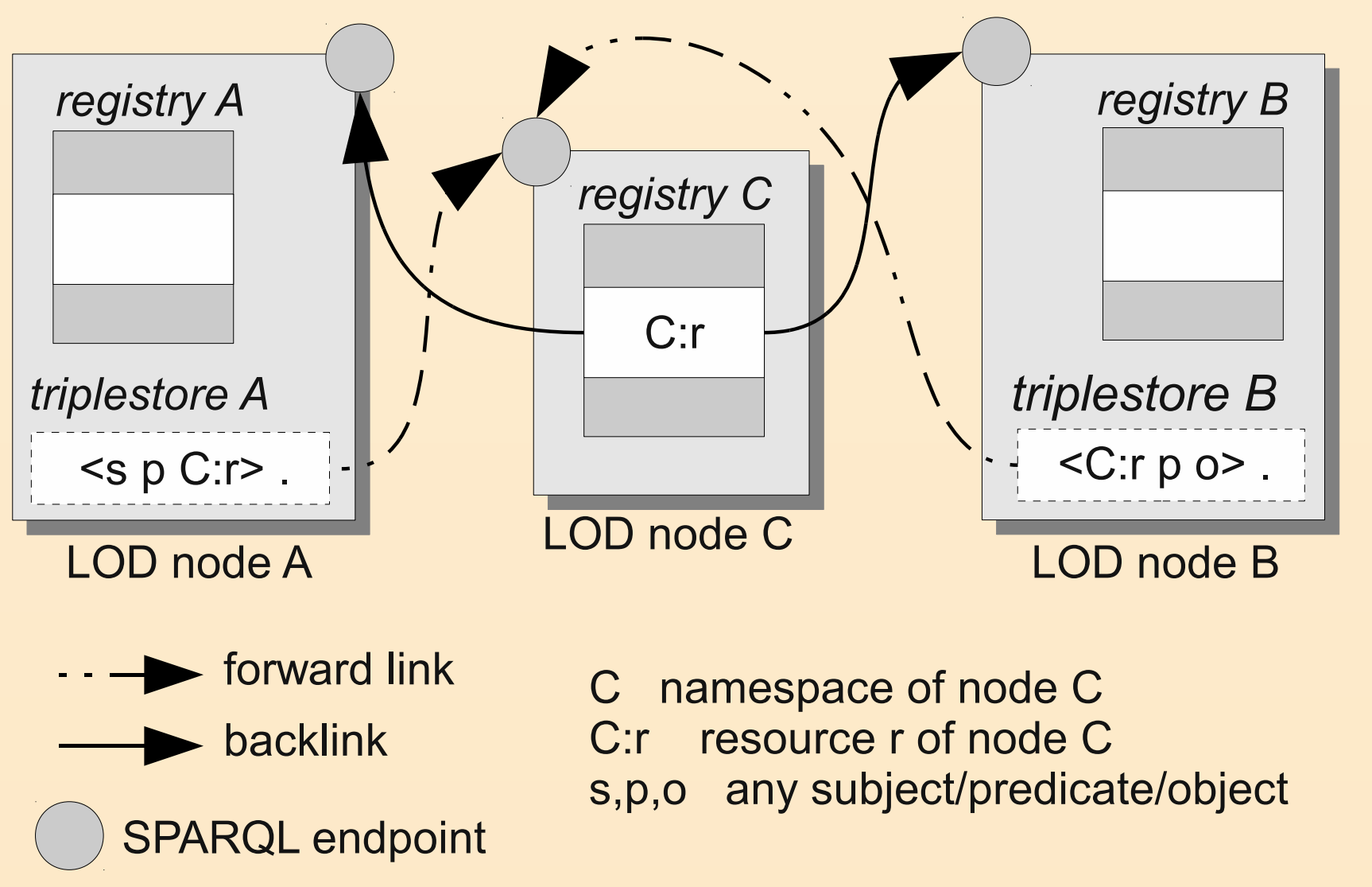

Figure 1: The backlinking scheme.

In **figure 1**, the typical interlinking between LOD repositories is illustrated as forward links, whereas the proposed enhanced interlinking is illustrated as backlinks.

The LOD nodes of study

A part of the **LOD-cloud** was studied with respect to the **interlinking** between nodes, depicted in **figure 2**.

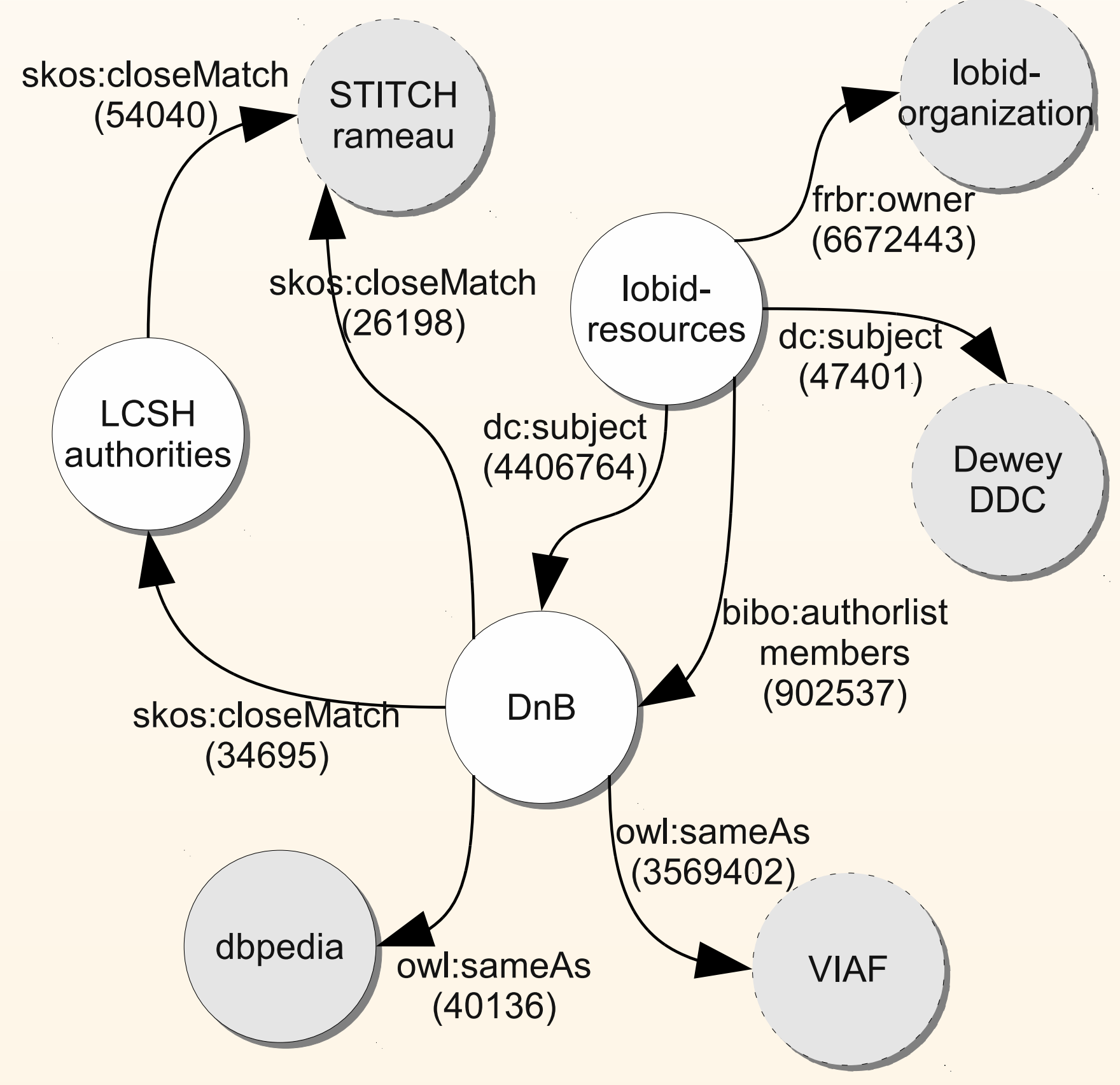

Figure 2: Interlinking predicates between studied nodes.

Each node constitutes a repository in a loose definition: some nodes provide **SPARQL endpoints**, other **data dumps** only, while a third category provides the bare minimum service of **dereferenceable URIs**.

What is clearly visible in **figure 2**?

- Links between LOD nodes present a significant **semantic richness**!
  - far beyond the the widely employed *owl:sameAs* predicate
- **Scalable** backlinking mechanisms should be **predicate-agnostic**.
  - based solely on the *owl:sameAs* or *rdf:seeAlso* constructs will fail to capture the evolving LOD interlinking

Example 1: Creating a backlink

The DnB node declares a *skos:closeMatch* **forward-linking** triple between its entity "Künstler" and the LCSH's "Artists" entity (**fig. 3a**).

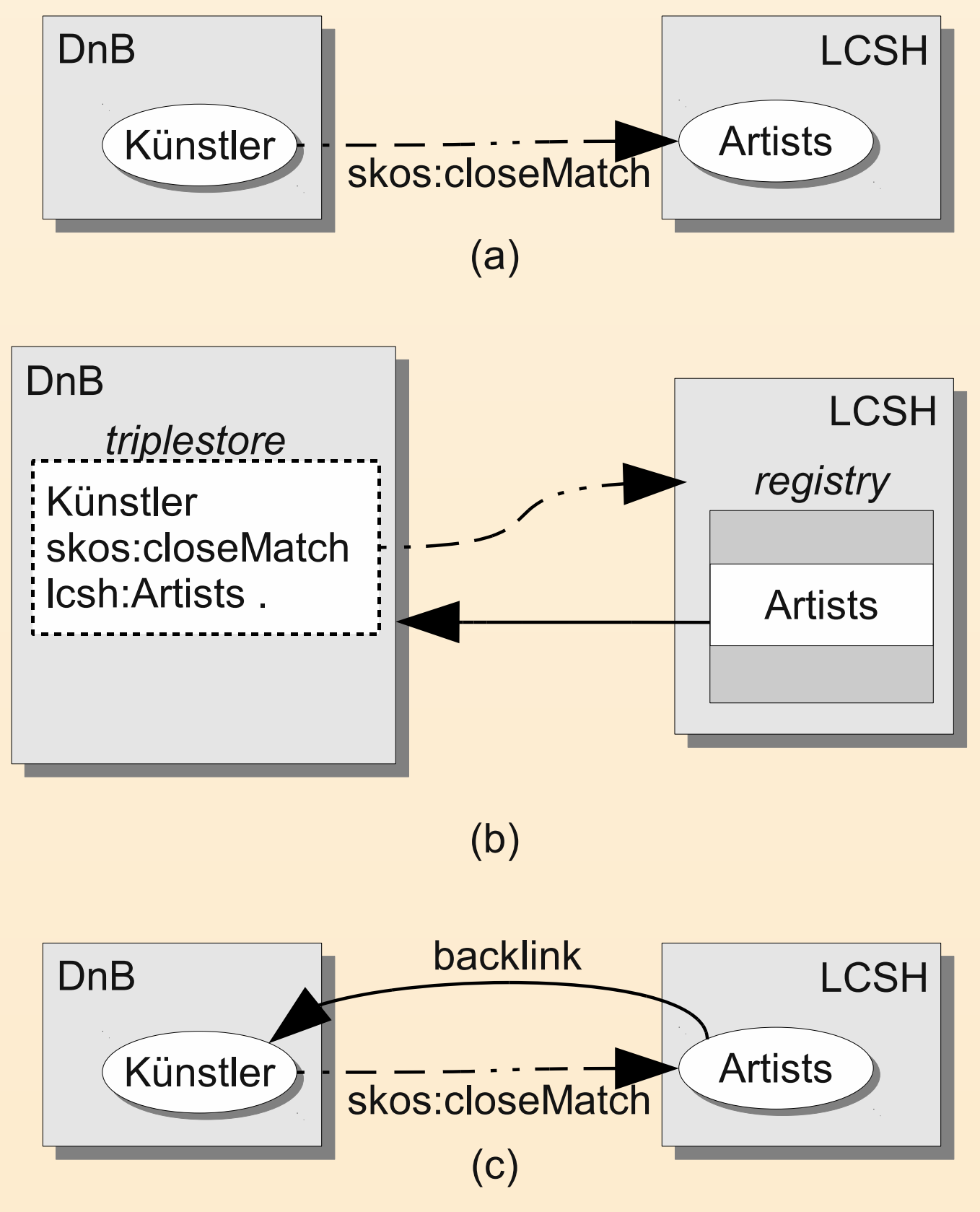

Figure 3: Backlink creation process.

1. The triple is entered in the DnB node with a tool aware of the backlinking framework.
2. The **backlinking web service** of the LCSH node is notified via an **HTTP request** for the newly created reference. The HTTP request carries as parameters a) the referenced item *lcsh:Artists* and b) the SPARQL endpoint of DnB node.
3. If accepted, the request will lead to the addition in the LCSH's registry of an entry **keyed by** ***lcsh:Artists***, having as **value** the SPARQL endpoint of the DnB node (**fig. 3b**).

LCSH registry can be queried in order to discover locations of external references to *lcsh:Artists*, creating in effect a **backlink** for *lcsh:Artists* pointing to the DnB node (**fig. 3c**).

Example 2: Backlink navigation

The **backlinking framework** presented herein can be used to support the process of **distributed data gathering**.

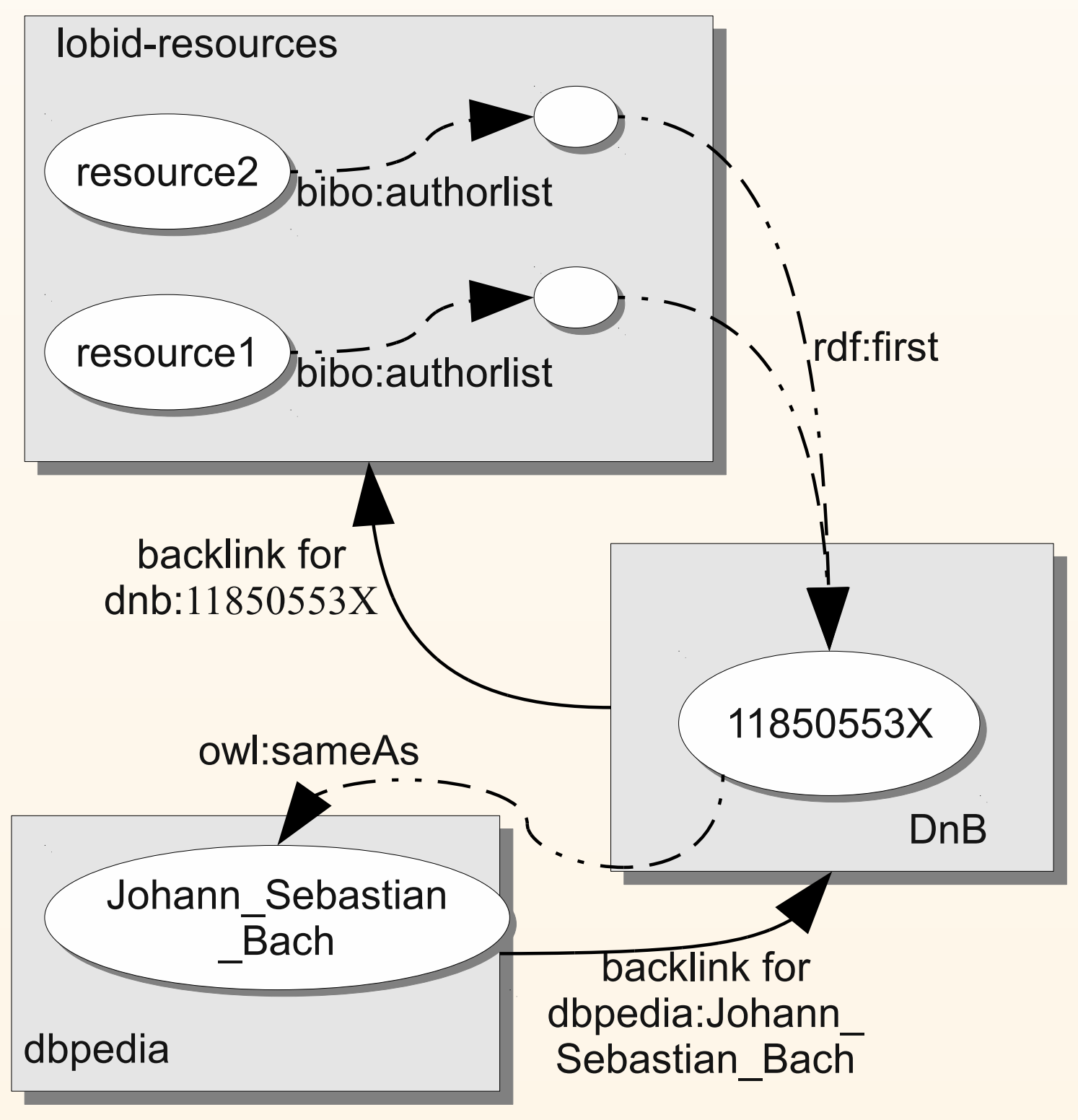

Figure 4: Navigation via backlinks.

The conceptual graph of **figure 4** between three of the studied nodes contains information about the composer Johann Sebastian Bach.

When dbpedia is used as a starting point for navigation or querying, **only backlinks** can provide a path for discovering entities in lobid-resources node that are related to *dbpedia: Johann_Sebastian_Bach* entity.

Example 3: Facilitating maintenance

In **figure 5**, another possible use of backlinking is demonstrated.

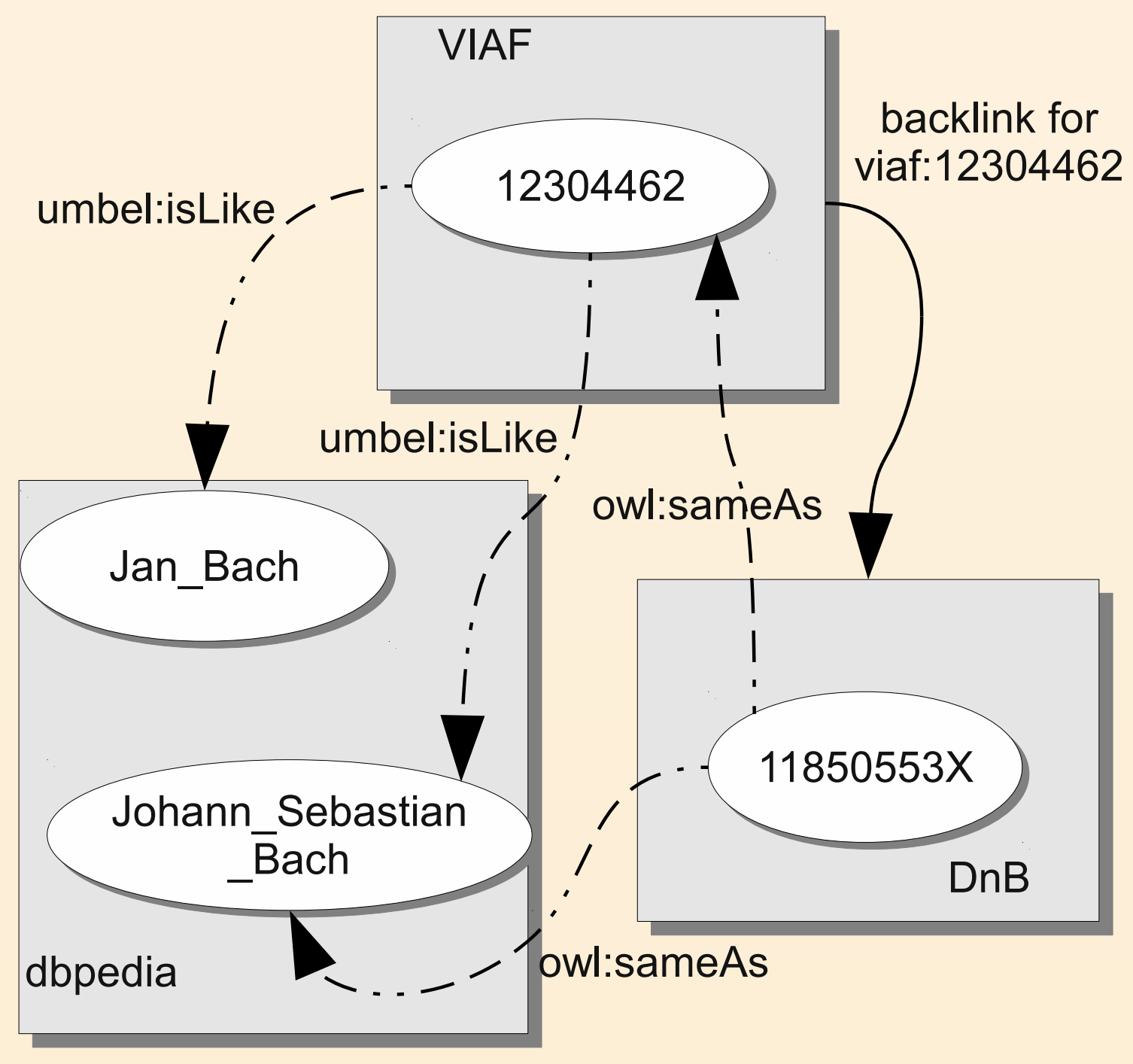

Figure 5: Checking identities via backlinks.

The VIAF node has **two possible mappings** towards dbpedia for its defined entity *viaf:12304462*, representing the composer Johann Sebastian Bach.

This dubious linking situation can be fixed: DnB node vouches that the entity under question is the same as *dbpedia:Johann _Sebastian_Bach*. This fact can be discovered **only through backlinking** to DnB.

References

[1] A. Langegger, W. Wöß, and M. Blöchl. A semantic web middleware for virtual data integration on the web. In S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, editors, *ESWC*, volume 5021 of *Lecture Notes in Computer Science*, pages 493–507. Springer, 2008.

[2] T. Omitola, C. L. Koumenides, I. O. Popov, Y. Yang, M. Salvadores, M. Szomszor, T. Berners-Lee, N. Gibbins, W. Hall, schraefel, and N. R. Shadbolt. Put in your postcode, out comes the data: A case study. In *The Semantic Web: Research and Applications, 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 - June 3, 2010, Proceedings, Part I*, pages 318–332, 2010.

[3] S. Tramp, P. Frischmuth, T. Ermilov, and S. Auer. Weaving a Social Data Web with Semantic Pingback. In P. Cimiano and H. Pinto, editors, *Proceedings of the EKAW 2010 - Knowledge Engineering and Knowledge Management by the Masses; 11th October-15th October 2010 - Lisbon, Portugal*, volume 6317 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 135–149, Berlin / Heidelberg, October 2010. Springer.