



OKKAM-based instance level integration

Paolo Bouquet

W3C RDF2RDB



This work is co-funded by the European Commission in the context of the Large-scale Integrated project OKKAM (GA 215032)

Using the Entity Naming System for Managing and Interlinking Entities in Networks of Data

- What is the benefit of using it?
 - Motivation
- ENS Architectural Overview
- OKKAM based data integration
- Conclusions: OKKAM in RDB2RDF

- The Entity Name System (ENS) provides a collection of **services** to support the systematic **re-use of identifiers**
 - Identifier **search**
 - Identifier **creation**
 - Alternative Identifier **management**
 - Create + update of entity **profiles**
- The ENS is built on a **scalable** architecture
- Services are provided via **SOAP**
- Web-frontends available
- Integration in **annotation pipelines** possible

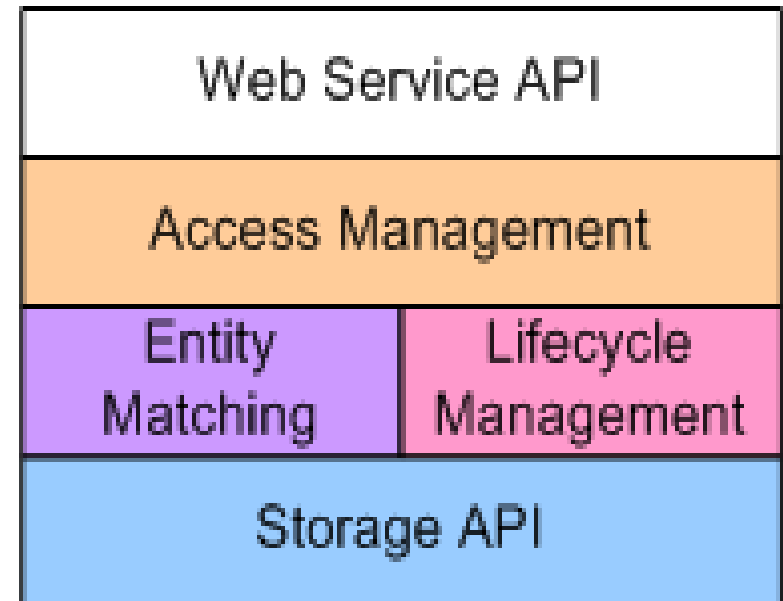
- Enhanced search facilities
 - Using OKKAM Ids and profiles, data (both internal and external) referring to a certain entity can be easily retrieved

- Maintaining up-to-date and representative metadata
 - Entity profiles can be updated based on the “popularity” of the attributes
 - Useless entries from the profiles will be in time removed
 - Useful entries will be promoted and ranked higher in the profile

- Integration
 - Search + aggregation of the information of interest
 - Analysis of corporate-internal unstructured data (BI)

- Access to the outside world
 - OKKAM public nodes can be provided with (restricted) profiles of SAP entities
 - External OKKAMized sources can be searched
 - Interesting BI analyses using external sources referring to SAP products, components, etc.

- Storage API
 - Data and search index management („recall“)
- Entity Matching
 - Ranking („precision“)
- Lifecycle Management
 - Data quality
- Access Management
 - Security, Privacy
- Web Service Layer
 - Accessibility

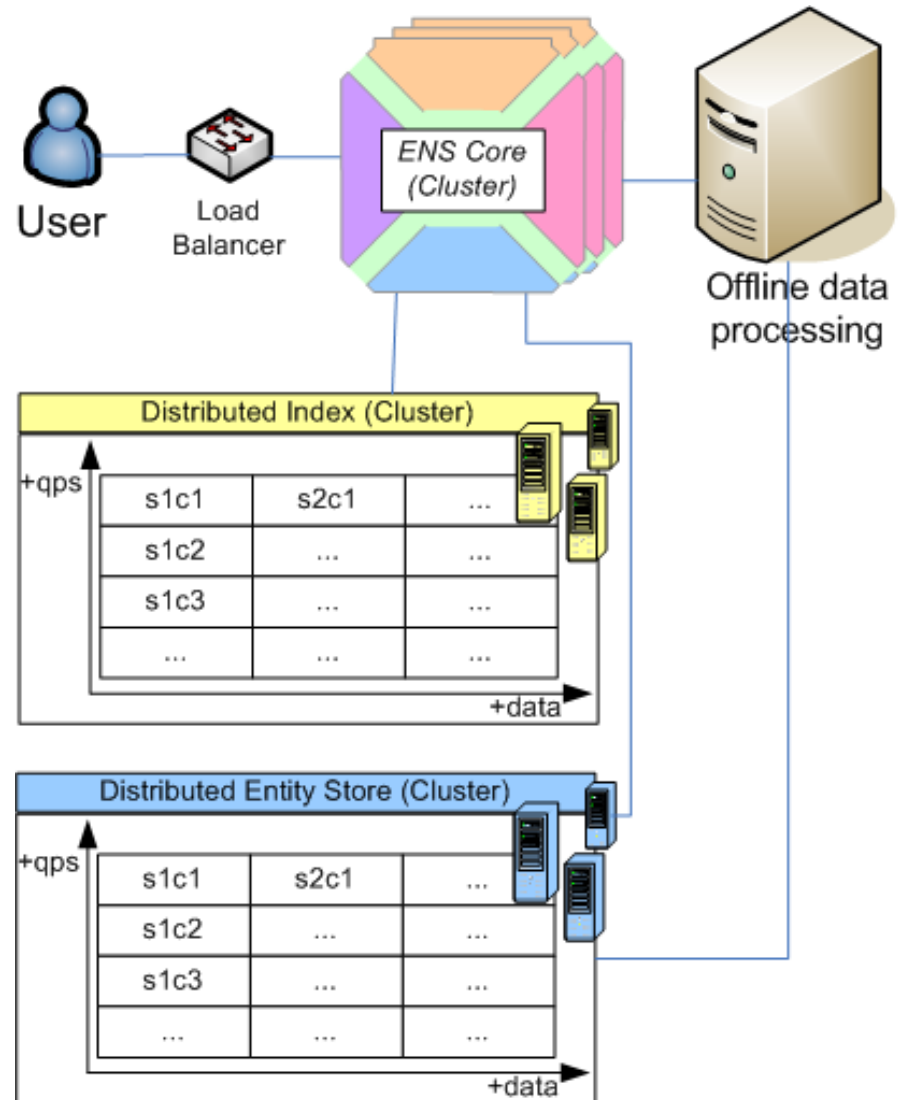


Scalability

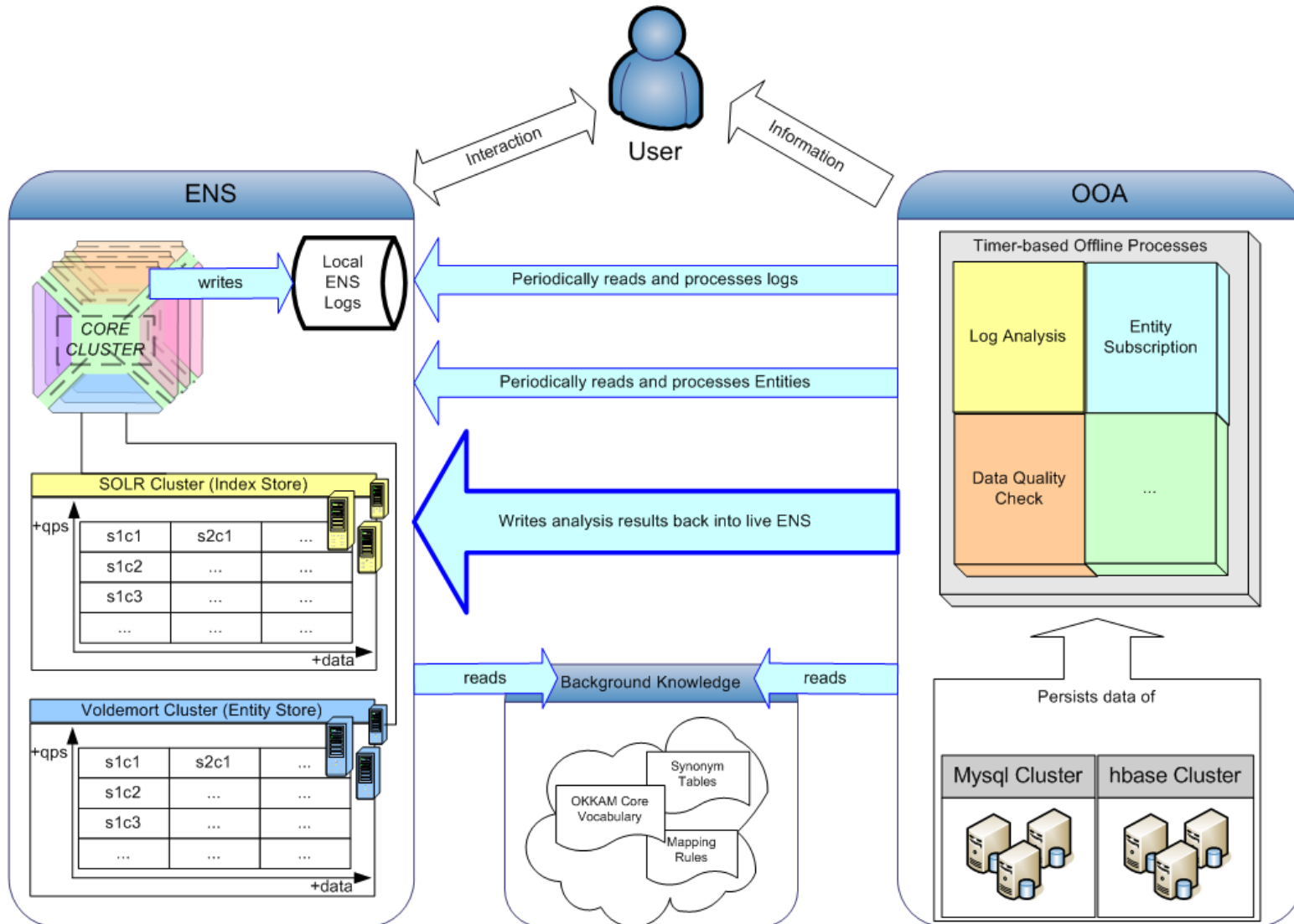


3 levels of clustering

- ENS Core
- Entity profile store
- Search index

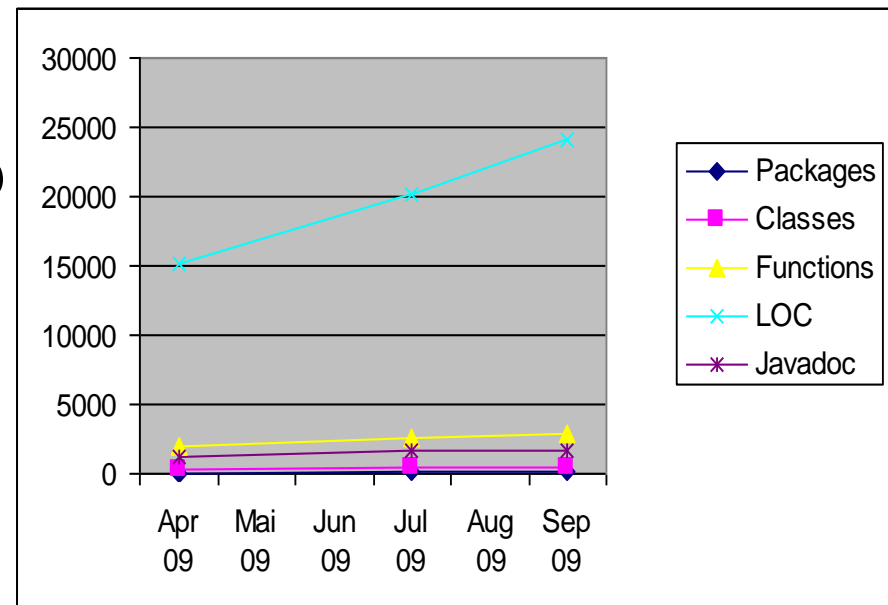


Offline Processing



	V1.0 (12/2008)	V1.1 (02/2009)	V2.0 (11/2009)	V3.0 (06/2010)
Repository capacity	500k	1Mio	50Mio	500Mio
Repository population	500k	1.03Mio	7.4Mio	50Mio
Avg. response time	2000msec	800msec	750msec	400msec
Queries per second	5	5	7	50
Number of CPU Cores	4	4	32	32
Clustered components	none	index	all	all

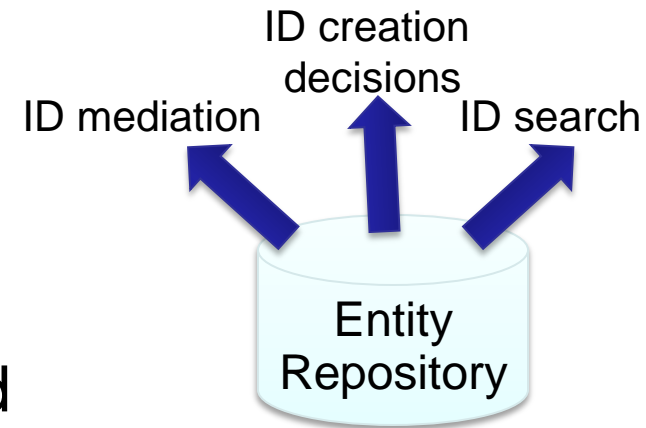
- The ENS codebase is steadily **growing**
- Entity base currently @ 7.5Mio



Entity Repository: Role in the ENS



- Core of the ENS
- Storage and management of
 - issued identifiers
 - information about the entities for which an identifier has been issued
 - (statistics + meta-information)
- focus on
 - discriminative information (helps in finding IDs)
 - alternative IDs (for ID mediation)
- basis for finding existing entity identifiers

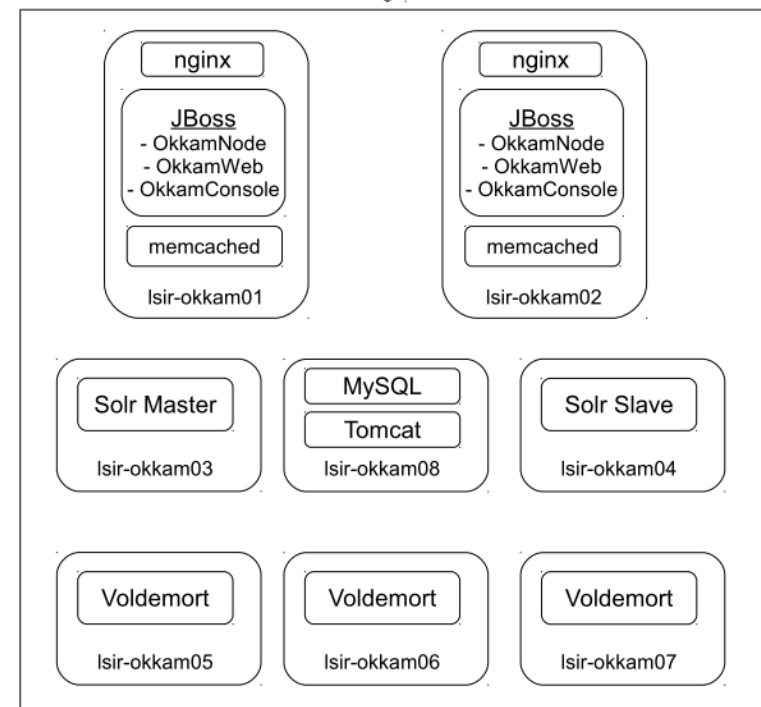


- management of
 - very heterogeneous entity descriptions (no fixed schema)
 - entities of very different types
 - potentially very large entity sets

The Entity Repository: Technology



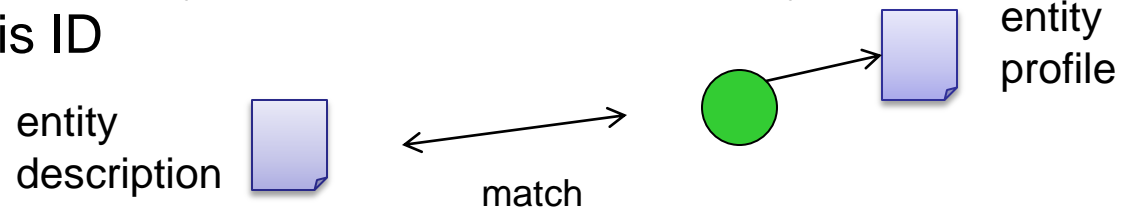
- XML representation of entity descriptions (entity profiles)
 - attribute value pairs describing entity properties
 - alternative IDs
 - links where further info can be found
- information management based on leading edge technology for large heterogeneous data sets:
key value stores +
IR indexing (Voldemort, Solr, Lucene)
- provision for redundancy and load balancing
- Specific technology extensions for entity repository (e.g. indexing with attribute annotations)



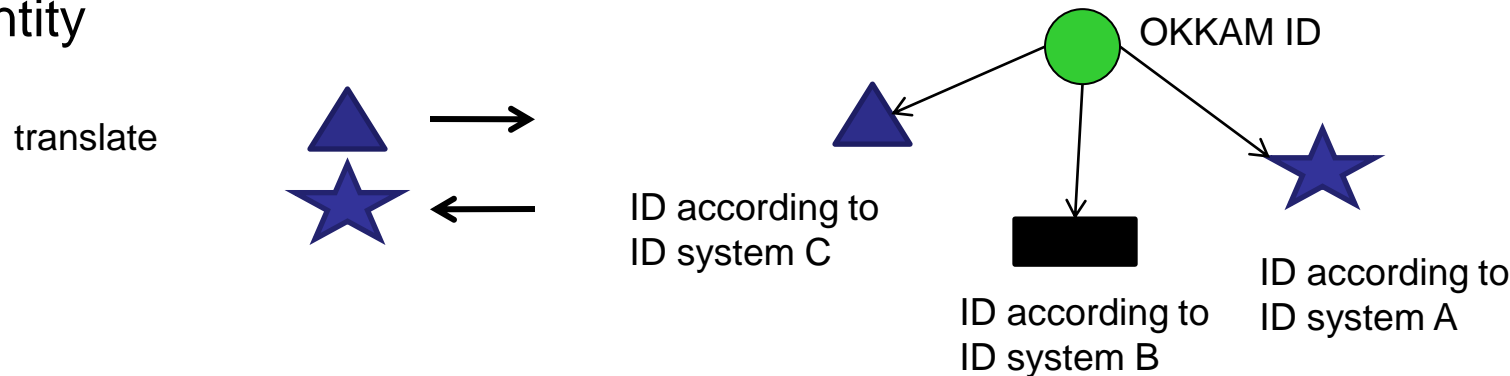
- current content: about 7.5 Mio Entities
- mix of mass import (Wikipedia, geonames) and manual creation
- focus on: persons, locations, organizations
- in addition: products (SAP), proteins
- in principle: no restriction with respect to types of entities that can be managed

Entity ID Search: Role in the ENS

- Core functionality of the ENS
- purpose: find existing entity identifiers
 - Motto: “If there is an identifier for an entity, it should be possible to find it”
- in more detail:
 - given a description of an entity, check, if there is already an ID for it, and if so, find this ID



- given an ID for an entity, check if there are other IDs in use for this entity



- Heterogeneity of entity ID requests:
 - requests from variety of sources (extractors, databases, human input)
 - keyword vs. structured requests
- Heterogeneity of stored entity profiles
- Possibility of over-specification of requests: User of ENS knows more or other things about an entity than the ENS
- Search in very large sets of entity profiles

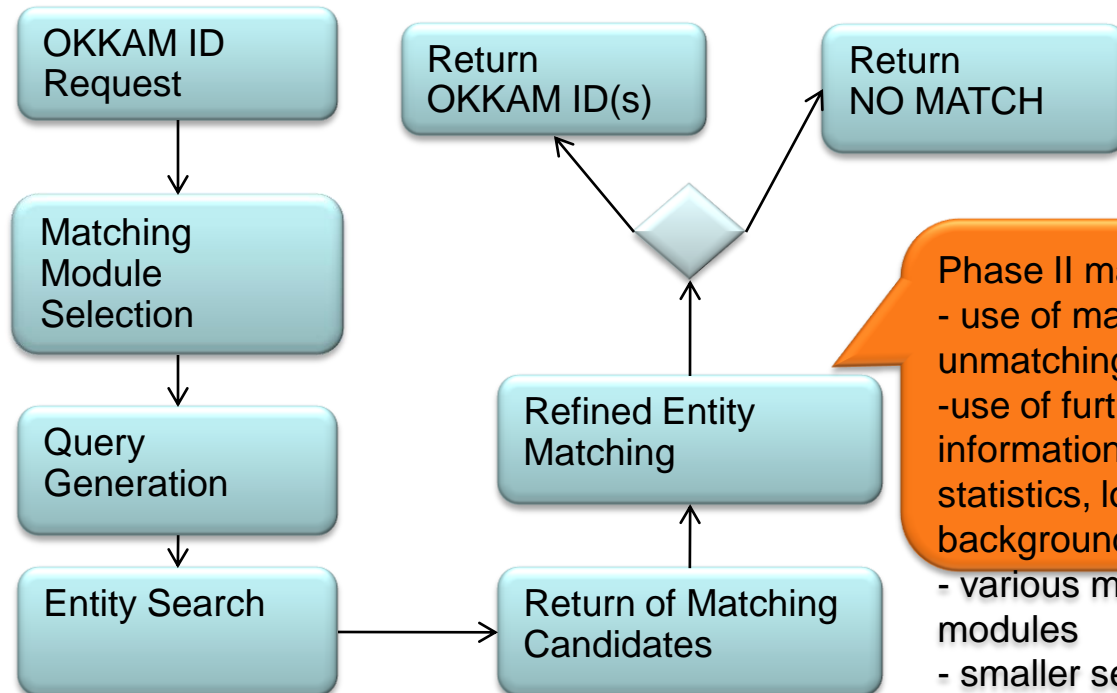
Entity ID Search: Functionality



- entity description as an input: keywords, attribute value pairs or combination of both
- Two phase process for ID request processing:

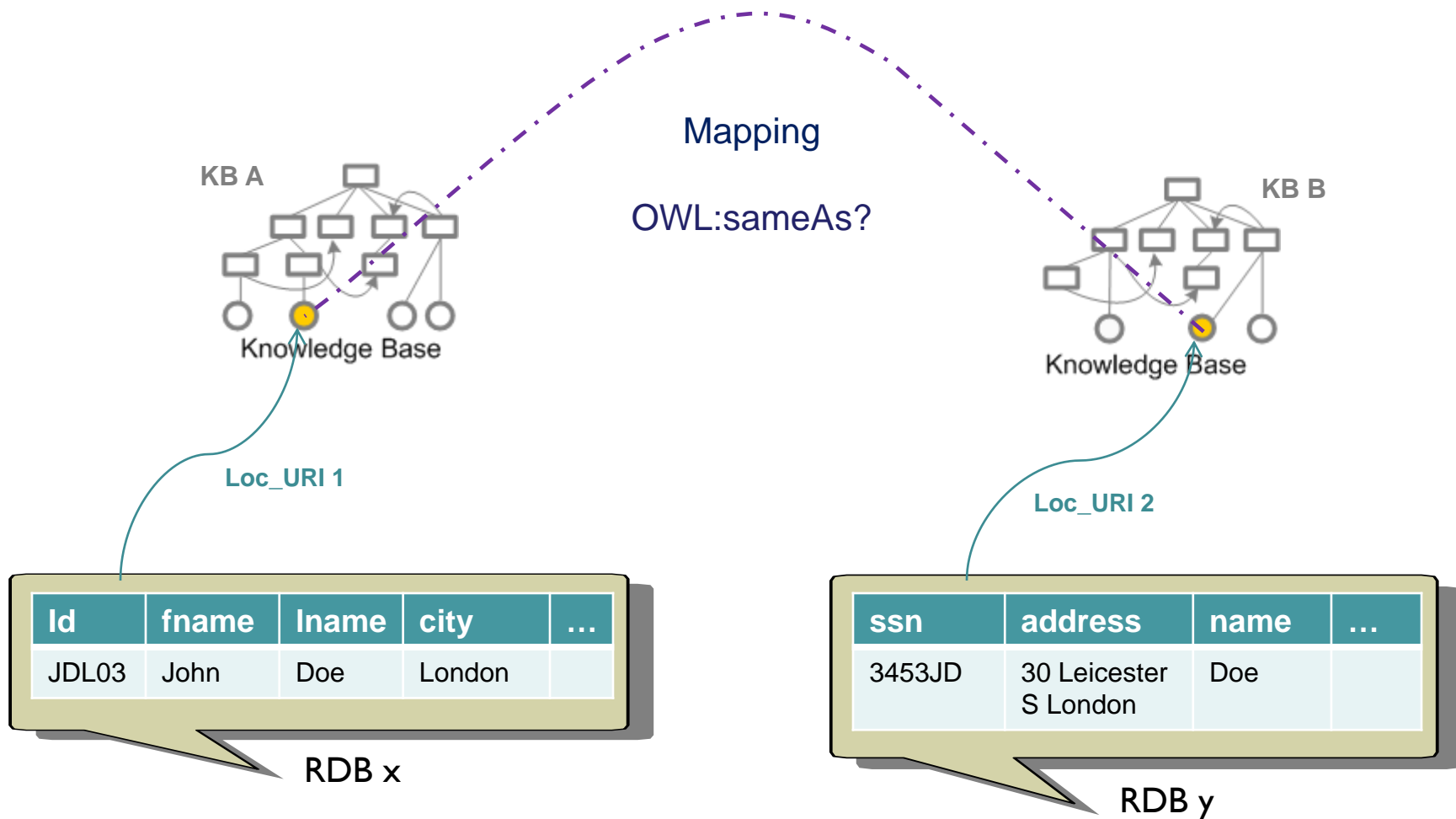
- heterogeneous requests
- no fixed schema
- keywords and/or attribute value pairs

Phase I matching:
large entity sets
focus on recall
mainly based on
search technology

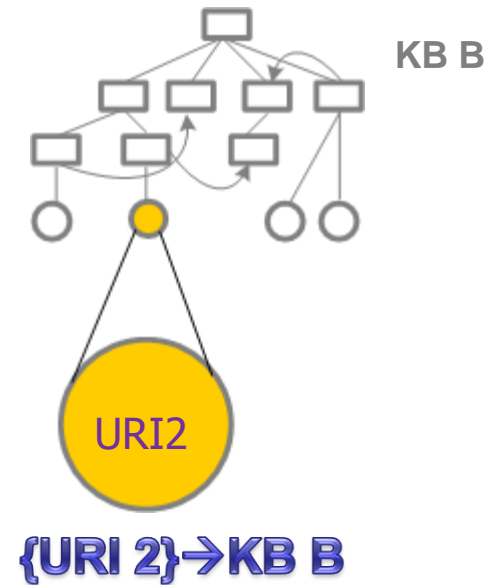
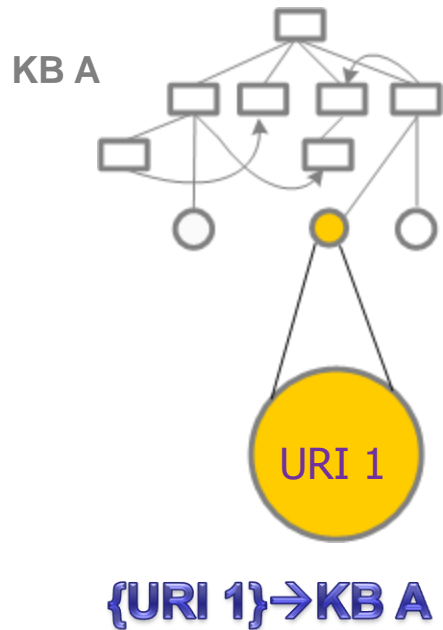


Phase II matching:
- use of matching + unmatching probabilities
- use of further information (e.g. statistics, logs, background info)
- various matching modules
- smaller set of matching candidates

Graph based Integration Solution



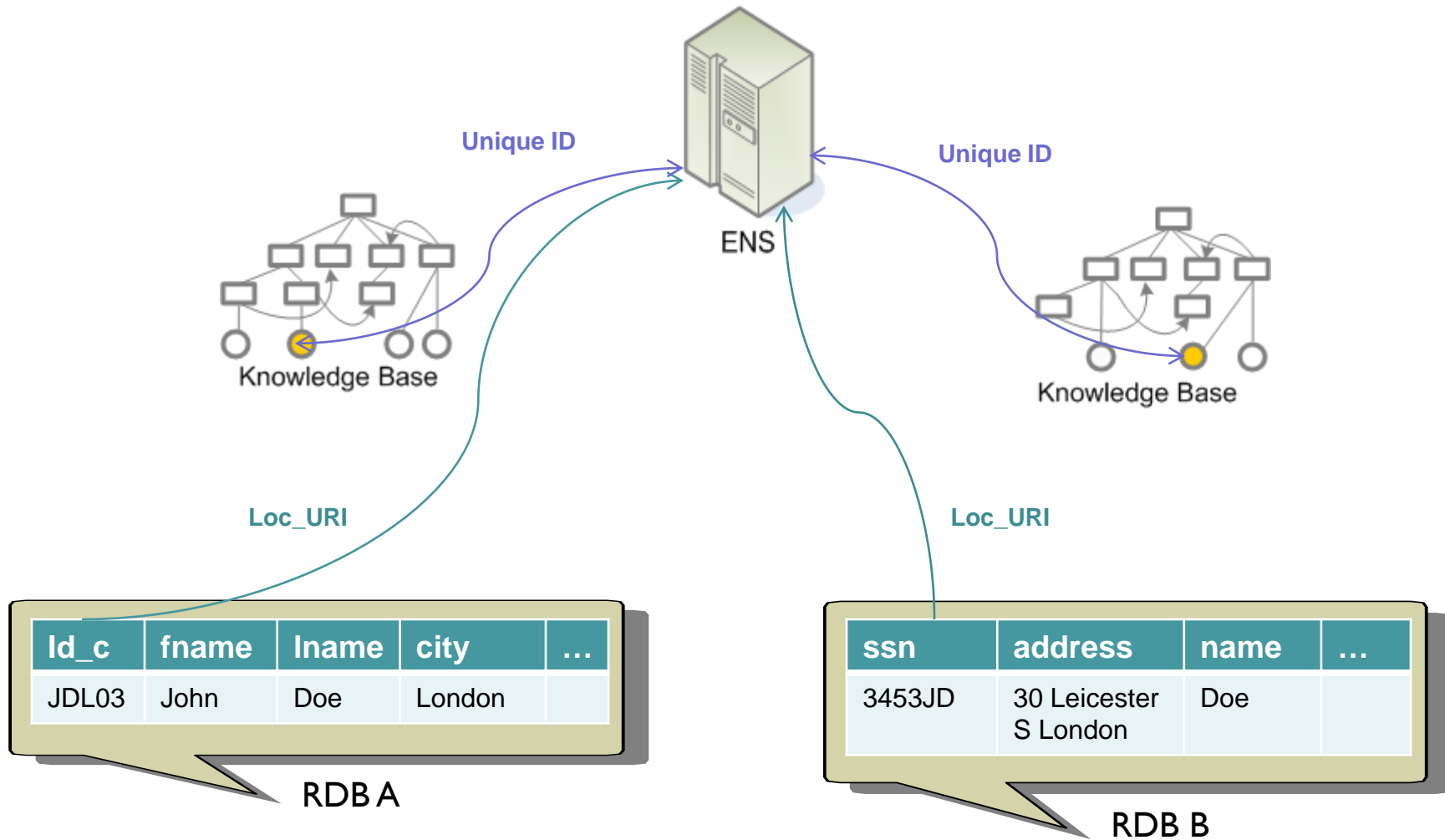
OWL:sameAs problem



| - UR1 1 sameAs URI 2 → {URI 1} U {URI 2}

The KB descriptions could be incompatible OR we do not want to accept the union of the two descriptions

OKKAM based Integration Solution

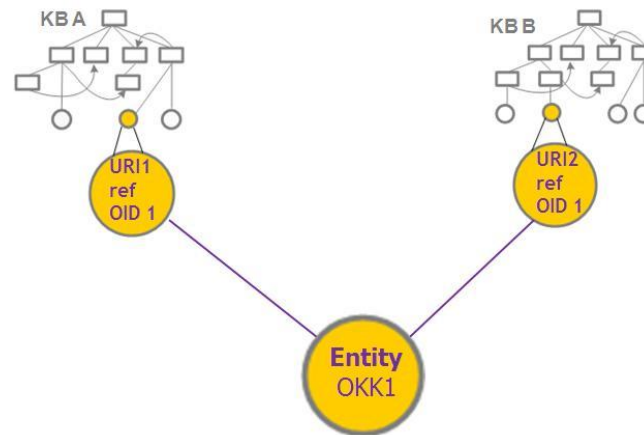


1. Databases are mapped into KB
2. OKKAM IDs are used for identifying an entity in different KBs
3. An entity profile is associate with an OKKAM ID, the aim of the entity profile is to making the entity recognizable.
4. KBs are queried using local identifiers and then the results are merged based on the OKKAM ID and the desired semantic rules.

Tax Project Use Case: Requirements & Solutions



- Instance level integration among many RDBMS;
→ OKKAMization of entities through databases
OKKAM ID as a single entry point

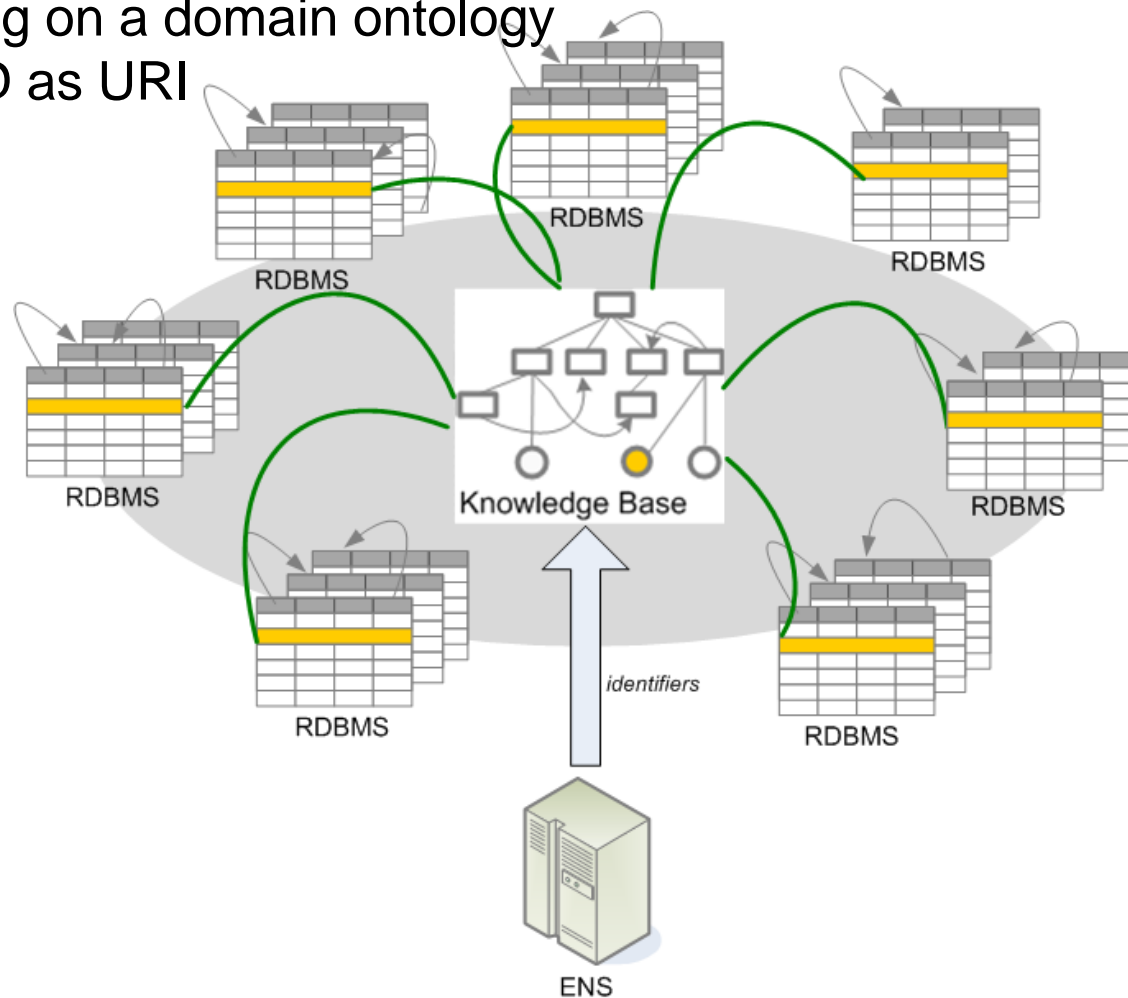


- Easy plug & play of additional RDBMS;
→ Flexible architecture based on a KB

Example Integration: Regional Tax Project

Entity level integration of regional administrative databases for tax control through:

- db mapping on a domain ontology
- OKKAM ID as URI



- Motivation:
 - A URI is indispensable to identify an element in RDF but not in RDB.
 - The ENS provides a mechanism to support the systematic re-use of identifiers.
- Proposed Plan:
 - Generate a mapping between OkkamIDs and local IDs.
- Benefits:
 - Easy to be indexed through OkkamIDs (sig.ma; sindice)
 - Easy to be integrated

Thank you!