Entity Identifiers for Lineage Preservation

Julien Gaugaz and Gianluca Demartini
L3S Research Center
Leibniz Universität Hannover
– OKKAM Goals:

• Foster the re-use of entity identifiers to ease information integration
• To create and manage a large collection of entity identifiers (EID)
• Not to create a complete knowledge base
  – Only discriminative information is stored
Operations during the Entity Lifecycle:

- Creation (ID Issuing)
- Split
- Merge

As a result of revising entity identity decisions.
– Goal: Resolve locally the lineage of entities
– By providing Lineage Preserving EIDs
– EIDs that include its history
  • Content changes, history doesn’t!
  • Lineage Preserving EIDs allow to detect deprecated EIDs locally
  • No need for querying the OKKAM node
  • Definitive advantage in a fast evolving environment and as long-term solution
Prime Numbers Labelling Scheme for DAGs

- DAG: G(V,E)

- Algorithm:
  - Assign a unique prime number \( p \) to each \( v \) in \( V \)
  - Label each \( v \) with \( (p \ast \text{lcm of its ancestors' label}) \)

Adapting Prime Number Labeling Scheme for Directed Acyclic Graphs
G Wu, K Zhang, C Liu, J Li - Database Systems for Advanced Applications, 2006 Springer
The history of an entity can be represented as a DAG

Prime number labeling can be used as a basis for creating the entity Ids

For each entity $e$ we assign $i = (e_{self}, e_{ancestor})$
Application of LPID

Sender

Local Entity Repository

Sender

Local Entity Repository

Receiver

isAncestor(7^{120}, 3) = \text{rem}(120, 3)
Using DNS for resolving EIDs

- We can encode LPID in a DNS node for associating user-friendly name to entities
- The max number of unique ancestors we can encode is, at least, 483 (estimation)

[RFC1035 (DNS standard)]
Thanks
Merge vs Split

- If the system knows A and receives C
- It can replace A with C

Merge

Split

- If the system knows A and receives C
- It only knows that A is deprecated