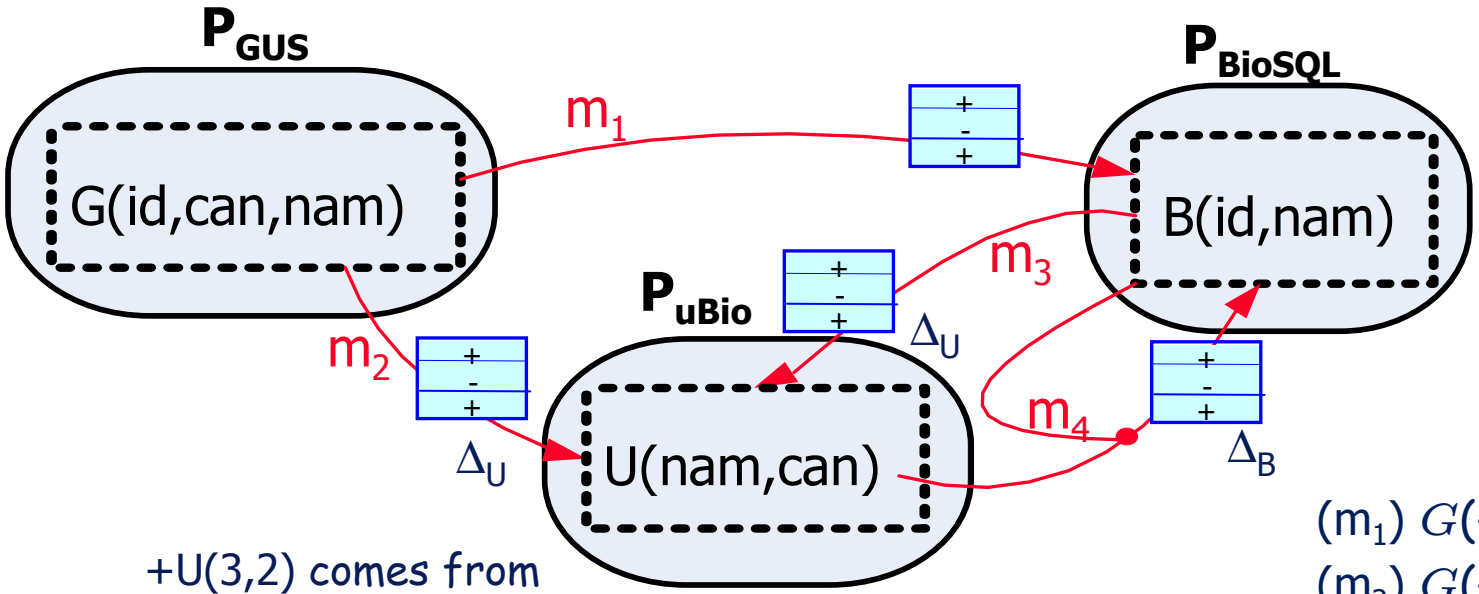
Provenance in ORCHESTRA

T.J. Green, G. Karvounarakis, Z. Ives, V. Tannen
University of Pennsylvania

Principles of Provenance (PrOPr)

Philadelphia, PA June 26, 2007

Collaborative Data Sharing [Ives+ CIDR 05]



(m1) $G(i,c,n) \rightarrow B(i,n)$
(m2) $G(i,c,n) \rightarrow U(n,c)$
(m3) $B(i,n) \rightarrow \exists c\, U(n,c)$
(m4) $B(i,c) \wedge U(n,c) \rightarrow B(i,n)$

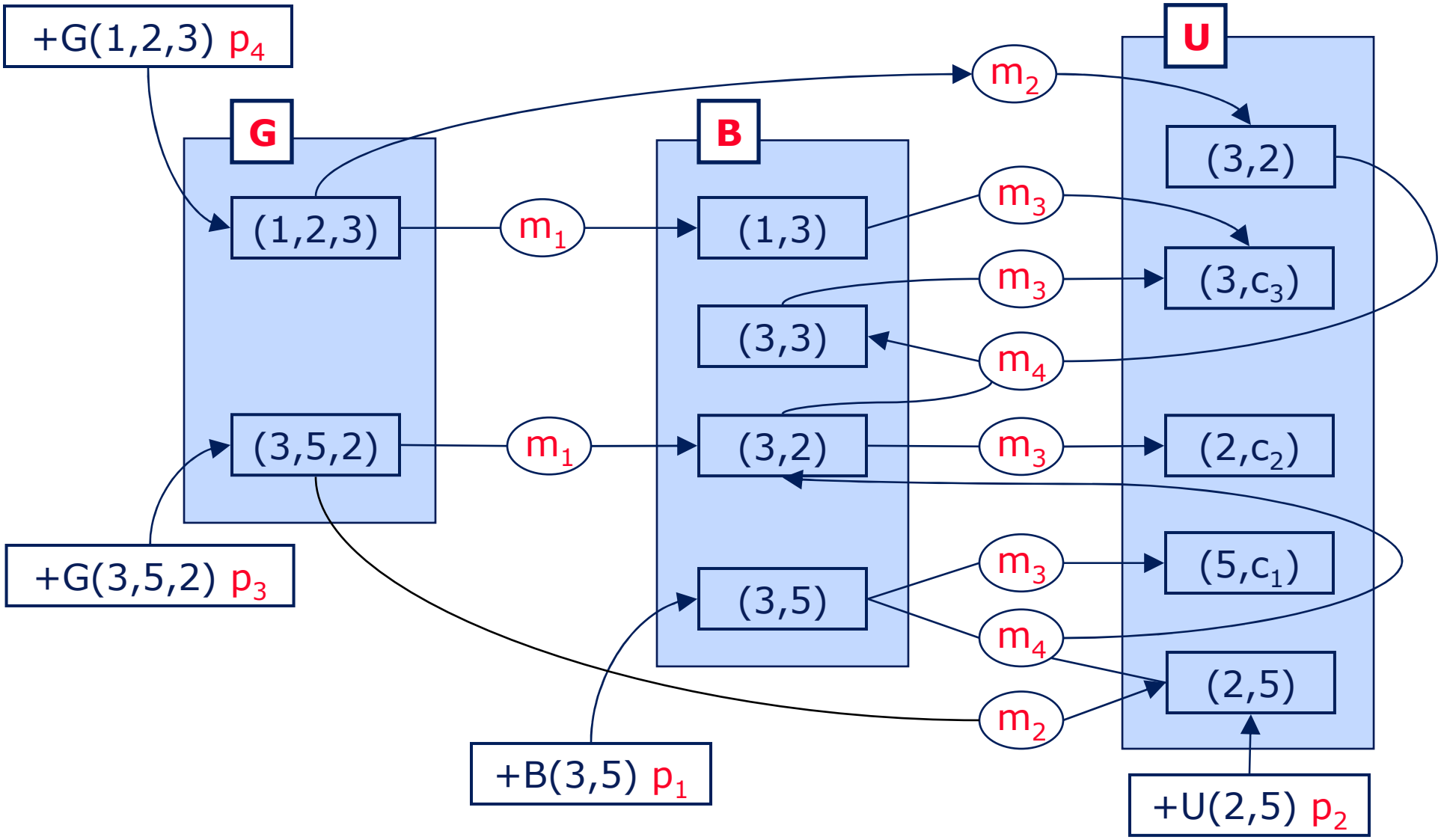
Schema mappings specify how data is logically related

Update exchange propagates updates and records provenance information

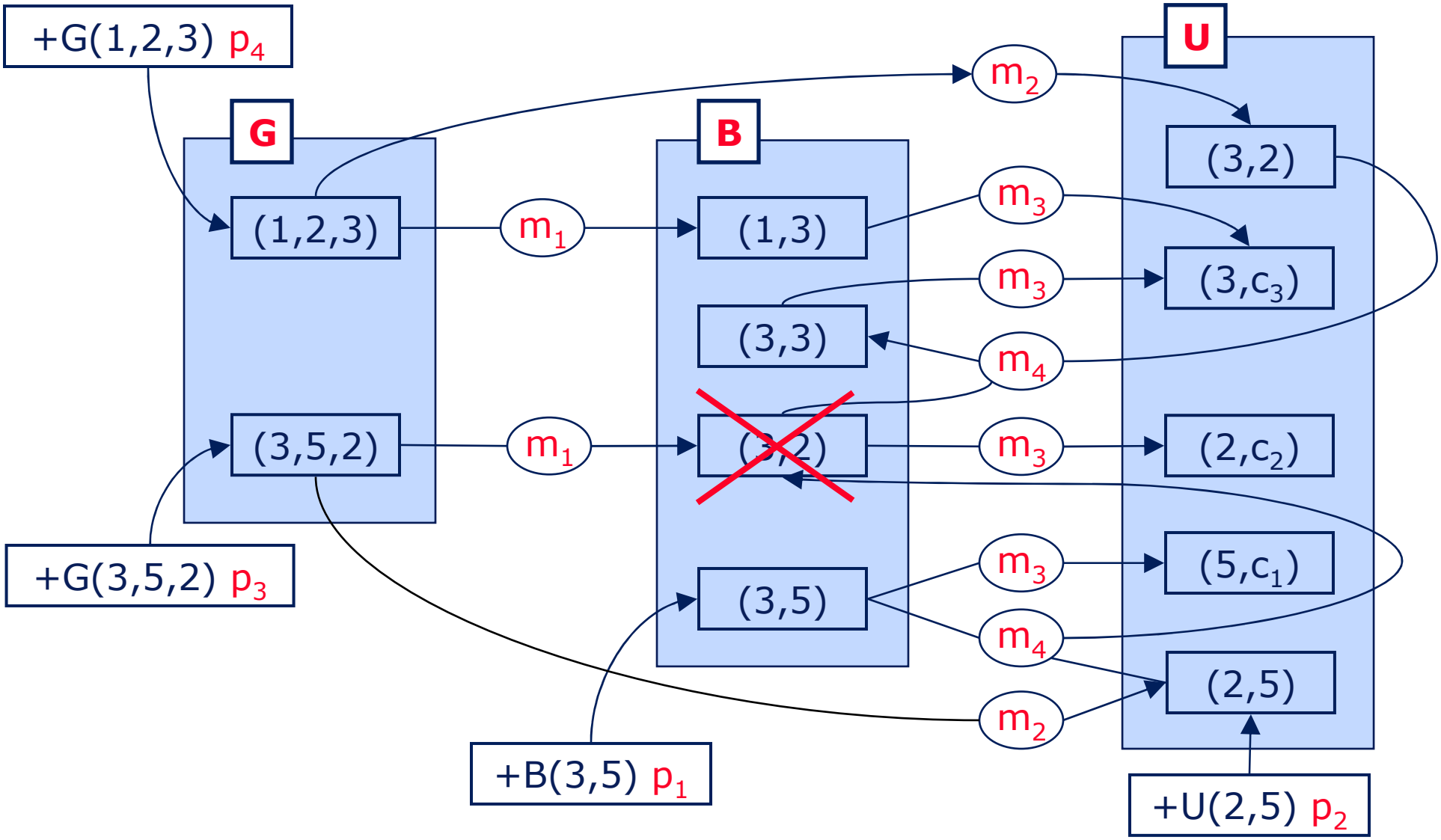
(1) to assess trust conditions

(2) to facilitate incremental maintenance

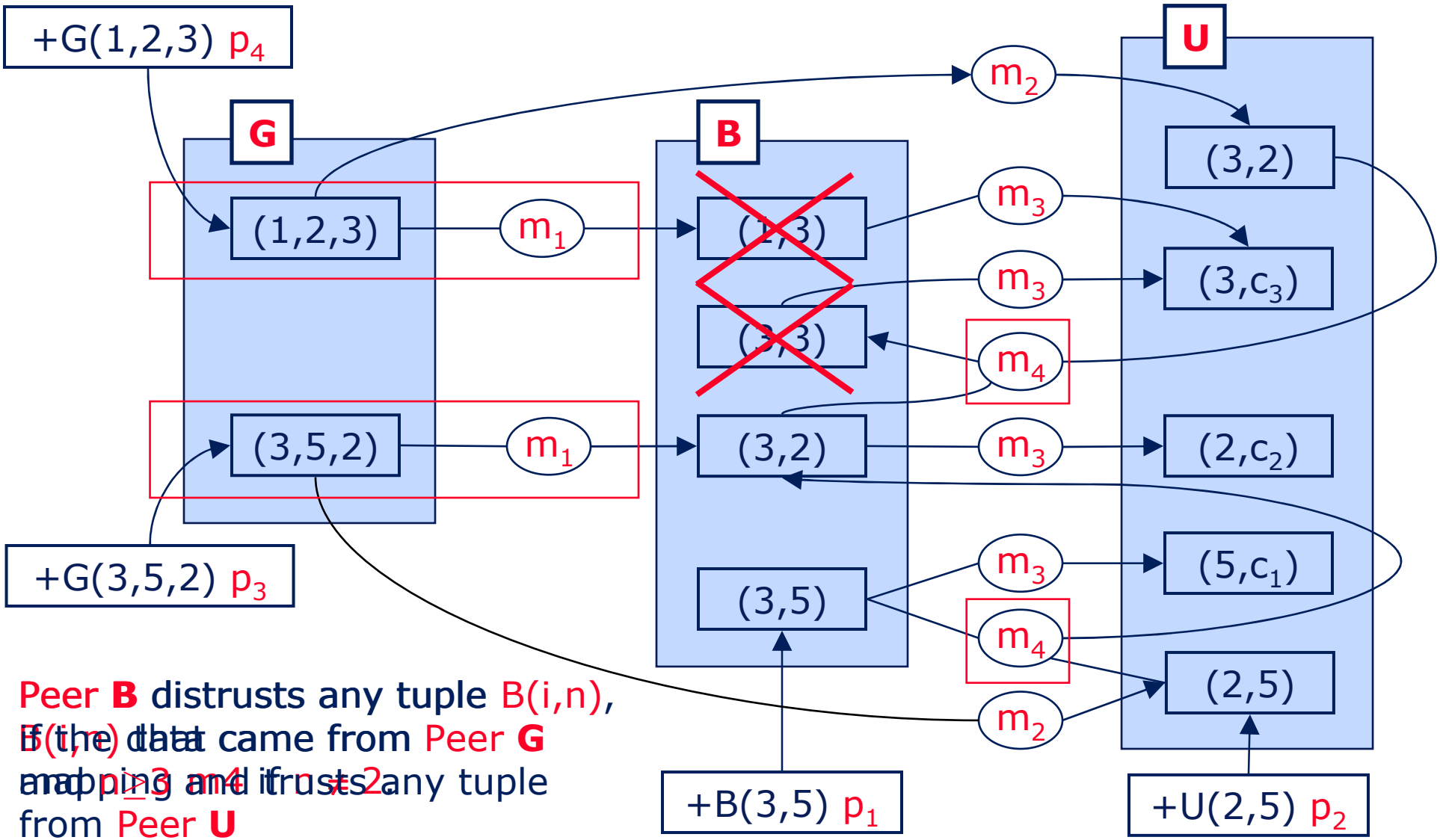
Insertions and provenance



Deletions and provenance



Trust and provenance



Peer **B** distrusts any tuple B(i,n), if the data came from Peer **G** and $p_3 \neq 2$ and it trusts any tuple from Peer **U**

Further aspects of ORCHESTRA

- Semantics: insertions and deletions with idbs
- Handling conflicts among trusted updates (Taylor+Ives SIGMOD 06)
- Prototype implementation (demo SIGMOD 07, technical details VLDB 07):
  - Java middleware layer using database as subcomponent
  - Provenance expressions stored as tables
  - tgds become datalog rules with Skolem functions
  - Update exchange using relational query engine (recursion!)
  - Feasibility experiments
- Future work / topics for discussion
  - What else to do with rich provenance information? (ranked trust models, bag semantics, querying provenance, ...)