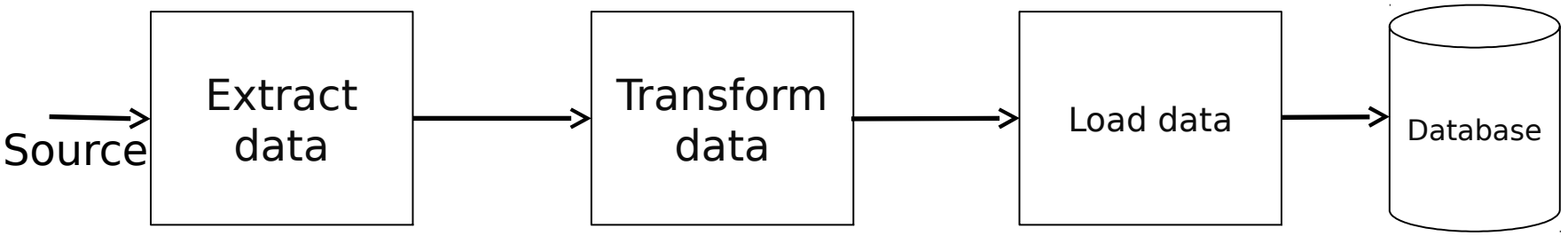
# ETL (Extract-Transfer-Load)

Devdatta kulkarni

# ETL

- ETL (Extract, Transform, Load)
  - Extract
    - Read the data from one or more sources
  - Transform
    - Transform the data
      - There could be series of transformations
  - Load
    - Load the data into the target system
      - Database
      - Workflow system

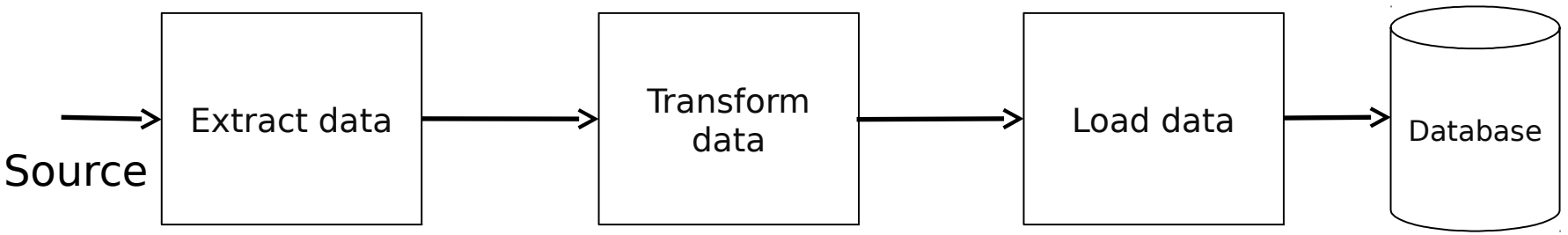
# ETL system: Conceptual architecture




(XML Feeds (Atom, RSS))
REST APIs
Databases
Websites

# Example

Search emails from particular user on openstack-dev mailing list

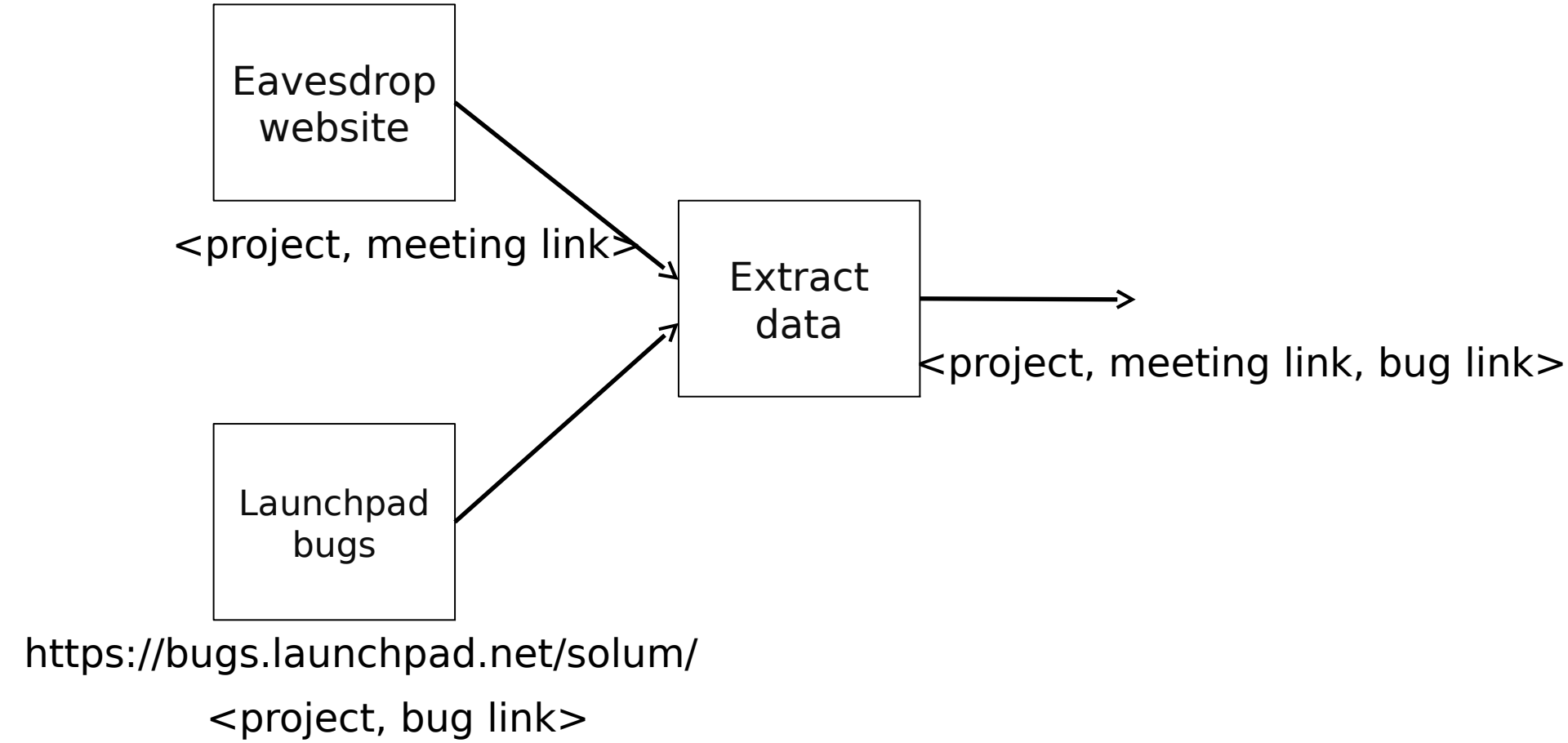



Source

openstack-dev mailing list archieve

# Loading from different sources

http://eavesdrop.openstack.org/meetings/solum_team_meeting/



# Issues in designing ETL systems

- Continually updating source
  - Need to periodically poll; maintain “last-queried-pointer”
- Load from different sources
- Merge data
  - Concurrency
    - Upsert
      - Insert or Update
- Target data representation
- Transformations
- High throughput
  - Using JDBC vs Hibernate (Object/Relational Mapping)

# Target data representation

- Single row, multiple columns

| Project | Meeting link | Bug link |
|---|---|---|
| Solum | <meeting link> | <bug link> |

- Multiple rows, multiple columns

| Project | Meeting link | Bug link |
|---|---|---|
| Solum | <meeting link> | - |
| Solum | - | <bug link> |

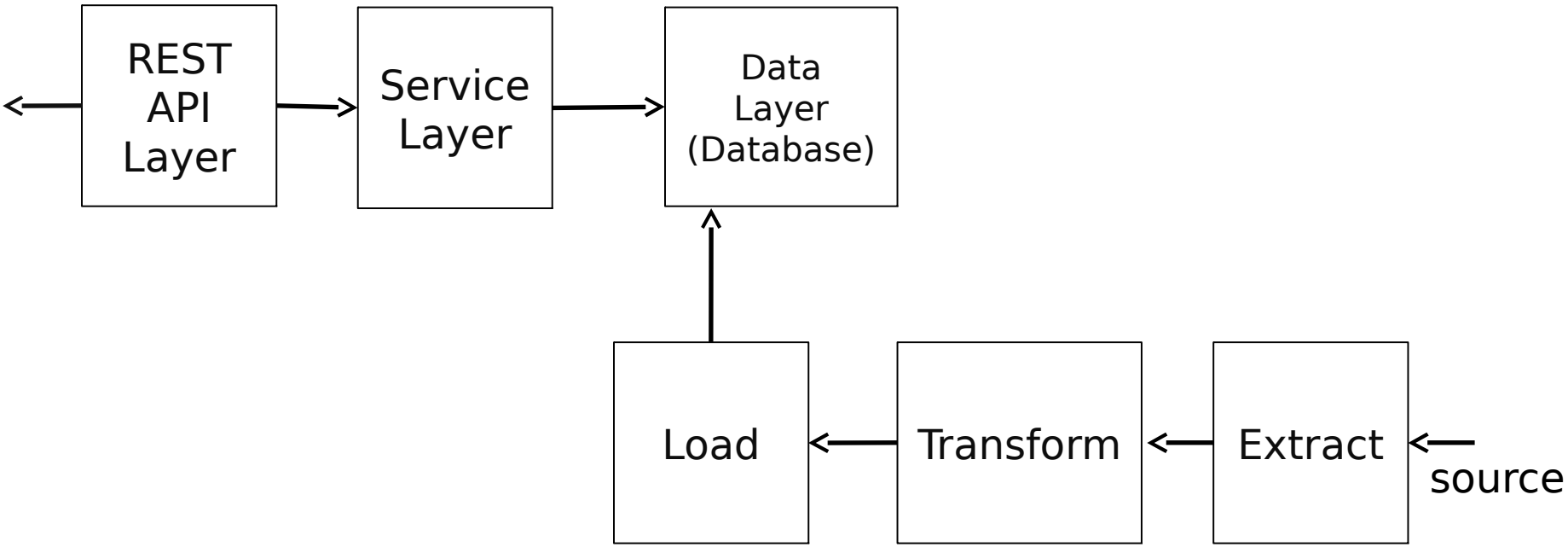
# Target data representation

- Single row, multiple columns
  - Pro:
    - Logic on the retrieval side is easier
      - E.g.: GET /projects/solum needs to query only single row from the target table
    - Data storage requirements proportional to number of entities (projects) in the system
  - Con:
    - Logic on the insert side is complex
      - Need to use ‘Upserts’ (Insert or Update)
        - » Why?
      - Prone to race condition issues

# Target data representation

- Multiple rows, multiple columns
  - Pro:
    - Logic on the insert side is easier
      - Every external representation of an entity is *inserted* as a separate row into the table
  - Con:
    - Logic on the retrieval side is complex
      - Need to write complex joins
      - Storage is proportional to product of number of entities and number of updates to them
        - » Could be huge

# ETL + REST


REST
API
Layer
Service
Layer
Data
Layer
(Database)
Load
Transform
Extract
source