The Evolution of Metadata: LinkedIn’s Journey

Shirshanka Das, Principal Staff Engineer, LinkedIn
Mars Lan, Staff Engineer, LinkedIn

Sept 25, 2019
LinkedIn’s Data Ecosystem

Members, Customers → Desktop, Mobile

Internal Apps

Employees

Services

Reads, Writes

Events

Reads

Databases (Espresso, MySQL, Oracle)

Derived Data Stores, Indexes (Pinot, Search, Voldemort, Venice, Graph, MySQL)

Streaming Ingest

Batch loads

Data standardization, Reporting, ML

Snapshots incremental dumps
What is Metadata?

What datasets do we have in our data warehouse (Hadoop, Teradata)
How do we easily find them
What does their schema look like
What datasets derive from these datasets
Attempt 1: Crawl Phase
Attempt 1: Crawl Phase

Crawl all catalogs you can
Parse all the logs you can
ETL into an opinionated data model
Build a search index + a lookup store
Build an app to serve this info

This is a useful product!
We gave it a clever name: **WhereHows**
And we open sourced it in 2016
Attempt 1: A few things we observed

- Pull-based integrations
- Central team’s burden
- Pipeline fragility, freshness
- Rigidity of model
- Hard to iterate quickly

"Grimaces" by Fouquier (CC BY-NC-ND 2.0)
What is Metadata?

What datasets do we have in our entire data ecosystem?
- Espresso, Kafka, Hadoop, Teradata, Search, Pinot …
- 20+ data systems

What does their schema look like?
What datasets derive from these datasets?
What business types are contained in these schemas?
Who owns these datasets?
Where are datasets being copied?
…

"[Katsiaryna Lenets] © 123RF.com"
Attempt 2: Walk Phase
Attempt 2: Walk Phase

Separate REST-ful service to support this diversity
Dataset Naming
Generic Schema model for all kinds of data stores and formats

New extensions to the model
Business metadata
Ownership

Scalable integration patterns
Pub-Sub using Kafka
Support REST + Kafka ingest route

We called it **TMS: THE Metadata Store :)**
We couldn’t open source it, too coupled with our internal business concepts :(
Attempt 2: A few things we observed

The Good
Teams were now accountable for the quality of their metadata and their custom ETL
Aggregate base metadata from all data platforms, overlay new metadata “aspects” on top easily

The Not So Good
New kind of metadata needed —> get in line behind central TMS team
Source of truth versus “Reflection of truth” debates —> no one wants to take a dependency
Standardized Event as interface: requires data model adapters everywhere, low incentive for producer to excel at their job
What is Metadata?

Dataset

- Name
- Schema
- Owners

"Thinking" by Elvin (CC BY-NC 2.0)
What is Metadata?
What is Metadata?

Models, Features, ...

Dataset

People

Name
Department
Dashboards

Name
Owners

Query
Name

Schema

Metric

Dashboard

Chart
Name
Metrics

Time Grain
Some observations about the problem

There is value in local sub-graphs, but the real value lies in the global model. Cannot execute this with a single monolith data model + service. Different teams care about different sub-graphs. Central team bottleneck. Micro-services can help? Back to silo-ed metadata problem.
Attempt 3: Run Together

Distributed but collaborative authorship of model

Single ORM-like layer to auto-generate integrations for

  CRUD operations

  Search queries

  Graph traversal

Distributed deployment and ownership of services possible

We’re calling this, the Generalized Metadata Architecture (GMA)
An Example Metadata Graph

**USER**
- urn
- firstName: John
- lastName: Doe
- Ldap: jdoe

**DATASET**
- urn
- platform name: fabric

**GROUP**
- urn
- size

**PROFILE**
{.firstName: John, lastName: Doe, Ldap: jdoe}

**OWNERSHIP**
{owners: [{type: SRE, user: jdoe}, ...]}

**MEMBERSHIP**
{admin: jdoe, members: [{jdoe, ...}]}

**SCHEMA**
{fields: [{type: integer, name: id}, ...]}

**Relationships**
- Owned By
- Has Admin
- Has Member
Metadata Serving

Metadata Service

- API Endpoints
- CRUD DAO
- Search DAO
- Graph DAO

Document Store
Search Index
Graph DB
Metadata Indexing

Metadata Service

- API Endpoints
- CRUD DAO

Metadata Audit Event

- Search processor
- Graph processor

- Search Index
- Graph DB
Putting it all together

GIT

Web App

Service

CRUD Event

Event Processor

Dataset Service

Search

SoT DB

User, Groups Service

Change Event

Event Processor

Change Event

SoT DB

Analytics + Relevance

Hadoop Data Lake
Design Decision Cheat-Sheet

Generic versus Specific Types:

- Support strong-types layered over generic storage, model-first development
- Expose strongly-typed REST API + Graph APIs for metadata traversal

Integration Strategy: Crawling versus Pub-Sub versus RESTful API:

- Prefer unified “Pub-Sub + REST-ful” API, build crawlers that publish

Single Source of Truth versus Replicated versus Federated:

- For new metadata systems, integrate as ORM layer
- For existing metadata systems, use Kafka to adapt
- Federation strategy only supports lookup cases, hard to support search and graph well
What about X?

A very brief survey of other similar systems in this space

Hive Metastore: limited to Hadoop Dataset, focused on query planning

Atlas: generic model, missing strongly-typed API, dynamic types supported

Marquez: strongly opinionated model, focused on data pipeline metadata

Ground: generic model, missing strongly-typed API, research prototype

Amundsen: Web-app backed by Hive Metastore, strongly opinionated model
Metadata Platform by the numbers

Datasets: ~5M
Dashboards: ~2K
Features: ~3K
Metrics: ~30K
Schemas: ~100K
People: 10K+

CRUD Events: ~2M / day
Relationship Events: ~3M / day
A Web App: Data Hub

Search and Discover Data Constructs
Find relationships, lineage, data quality, ...
Enrich metadata

Some interesting parallels to the metadata platform in terms of extensibility
  How do we add new pages to this app in a multi-team environment?
**Data Hub: a Dataset’s page**

---

**Metrics.MetadataChangeEvent_v4**

*Fabric: Prod*  

![Health Score: 100%](image)

<table>
<thead>
<tr>
<th>Column</th>
<th>Data Type</th>
<th>Default Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>auditHeader[time]</td>
<td>Long</td>
<td>The time at which the event was emitted into kafka.</td>
</tr>
<tr>
<td>auditHeader[server]</td>
<td>String</td>
<td>The fully qualified name of the host from which the event is being emitted.</td>
</tr>
<tr>
<td>auditHeader[instance]</td>
<td>String</td>
<td>The instance on the server from which the event is being emitted. e.g. i001</td>
</tr>
<tr>
<td>auditHeader[appName]</td>
<td>String</td>
<td>The name of the application from which the event is being emitted.</td>
</tr>
<tr>
<td>auditHeader[messageId]</td>
<td>Fixed</td>
<td>A unique identifier for the message</td>
</tr>
</tbody>
</table>

Last modified: 06/22/2019, 1:21:57 am
Data Hub: a Metric’s page

**Metric Details**

**Metric**: cpu_cost

- **Display Name**: -
- **Display Group**: -
- **Description**: sum(cpu_cost)
- **Tier**: Daily, 2

- **Frequency**: Daily
- **PII**: No
- **Good Direction**: Up
- **Derived**: No
- **Time Series**: Yes

**Dimensions**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname</td>
<td>Name of the host</td>
</tr>
<tr>
<td>data_center</td>
<td>Name of the data center</td>
</tr>
</tbody>
</table>
Data Management with Compliance

Data Management (Purge, Export, ...)

Data Access Layer

Application Code

Std Frameworks

Operations

Metadata Platform

Physical Data
### Data Pipeline Operations

**Daily**

- Time Period: August 20, 2019 - August 27, 2019
- Dataset: test_spark

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Flow Date</th>
<th>Execution History</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2019/08/22 00:00</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Id</th>
<th>Status</th>
<th>Updated</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>5785550</td>
<td>DONE</td>
<td>2019/08/22 20:21</td>
<td>metricsV2</td>
</tr>
<tr>
<td>5782201</td>
<td>FAILED</td>
<td>2019/08/22 15:51</td>
<td>metricsV2</td>
</tr>
</tbody>
</table>

[Refresh]
Powered by Metadata

Search and Discovery beyond just Datasets

Data Management, Access with Compliance

Operational Monitoring, Incremental Compute

AI: Model, Feature reproducibility, explainability

... and we’re just getting started

Metadata Platform
It’s open source!
**Open Source : the details!**

Alpha release out currently at [https://lnkd.in/datahub-alpha](https://lnkd.in/datahub-alpha)

Check it out at: Github project wherehows, branch: datahub  ([https://lnkd.in/datahub-github](https://lnkd.in/datahub-github))

**Capabilities:**
- Generic Modeling Layer with CRUD on MySQL
- Search on Elastic
- Graph on Neo4j*

**Interfaces:**
- The DataHub Web App
- REST-ful service from model files
- Event Processors for metadata events

**Metadata Model:**
- Datasets
- People

**Integrations**
- Data Catalogs (Hive, Kafka, JDBC*)
- People (LDAP)

* coming real soon
Open sourcing the Data Model

LinkedIn Internal Model

Open Source Model
Open Source: what’s coming next!

More Entities: [Jobs, Flows]
More Aspects within existing entities: [e.g. ReplicationPolicy, HiveSpecification]
More Integrations: [Calcite-compatible systems for fine-grain lineage]
App Features
   User pages
   Social interactions

Get engaged and contribute to the global model
Build integrations with systems
Thank You!