END-TO-END DATA GOVERNANCE WITH APACHE AVRO AND ATLAS

Barbara Eckman, Ph.D.

Principal Data Architect

Comcast

COMCAST

Mission

Gather, organize, make sense of Comcast data, and make it universally accessible through Platforms, Solutions, Products.

- Dozens of tenants and stakeholders
- Millions of messages/second captured
- Tens of PB of long term data storage
- Thousands of cores of distributed compute

Motivating example

- Record representing an IP Video player download session.
  - Time download session created
  - List of timestamps when video download started
  - Endtime, status (failure, success)
  - Device info
  - Customer Account info
  - Asset info (ie movie, tv show, etc being downloaded)
  - Fragments: total, completed
  - Failure events
- Suppose I want to integrate this data with :
  - Customer experience data -- join on customer account
  - IP player analytics session (latency, buffering) -- join on device, asset id
  - Comcast network traffic data -- join on device

COMCAST

Data integration shop of horrors!

- Customer account ids:
  - Xcal
  - Xbo
  - Billing
  - “Service”
- Device ids:
  - Physical
  - DeviceId
  - Xcal
  - Xbo
- Asset ids:
  - ProviderId, assetId
  - streamId
  - recordingId
  - EAS_URI
  - mediaGuid
  - mediaId
  - assetContentId
  - programId
  - platformId

Data and Schema Governance to the Rescue!

COMCAST

Questions: Schema Governance

- I want to join your data with my data—but what does your data mean???
- Is it safe to join my data with yours (Avoid “Frankendata”)?
- What attributes in your data match attributes in mine? (ie potential join fields)
- Can I change my schema without breaking systems of others that rely on it?

Questions: Data Governance

- Where can I find data about X?
  - How is this data structured?
  - Who produced it?
- How has the data changed in its journey from ingest to where I'm viewing it?
- Where are the derivatives of my original data to be found?

Outline

- Avro for Schema Governance
  - What is Apache Avro
  - How Comcast BPs make Avro even stronger
- Atlas for Data Governance
  - Metadata Browser and Schema registry
  - Comcast extensions to Atlas
  - Platforms and Atlas work together for lineage

Apache Avro

- A data serialization system
  - **A JSON-based schema language**
  - A compact serialized format
- APIs in a bunch of languages
- Benefits:
  - Cross-language support for dynamic data access
  - Simple but expressive schema definition and evolution

Avro Types

- *primitive*
  - string
  - bytes
  - int & long
  - float & double
  - boolean
  - Null (enables optional fields)
- *complex*
  - record
  - array
  - map: string -> Type
  - union (aka “choice”)
  - fixed<N> (byte arrays, eg uuid)
  - enum: [“larry”|“moe”|“curly”]

Simple Avro Example—Generic Application Log Event

```
 {  "name": "loggerName",
    "type": "string“,
    "default": “unknown”,
    "doc": "The name of the logger class that
       generated the message.”},
{  "name": "message",
   "type": ["null", "string“],
   "default": null,
   "doc": "The formatted message of the log event."}
```

I want to change my schema…will it mess others up??

- AVRO serialized data is never used without its schema.
- We have empirically determined a complete set of rules for schema evolution and backward compatibility, for example:
  - You may add an optional field to your schema
  - You may not change the type of any field

I want to change my schema…will it mess others up??

- Some chains of compatible changes produce incompatible results
  - – remove optional string field X
  - – add new optional int field X
  - – This sequence has changed the type of field X from string to int!
- We can use this data in reasoning about schema evolution, e.g. "is this schema compatible with the schema 3 versions back?"

Avro Schema Creation Best Practices

- Each schema is checked for compliance with Comcast conventions on compile
- Each schema is reviewed and approved by at least one human being
- Comcast conventions:
  - doc comments required to document every attribute
  - All attributes must have default values
  - Unnecessary complexity is discouraged (YAGNI principle)

Avro Schema Creation Best Practices

Data governance policy on updates:

- Data must always match a schema in the schema registry or be traceable to such a schema
- Updates to schemas of data “in flight” or “at rest” are not permitted, though re-publication of enriched data is permitted.
- Schemas in the registry may be evolved at will, but non-compatible changes should be kept to a bare minimum

AND MOST IMPORTANTLY…

Philosophy of modular core subschema reuse

- Github repo of core subschemas
  - application (running on a device)
  - customerAccount
  - device
  - error
  - geolocation
  - header (timestamp, uuid, hostname)
  - logEvent
  - moneyTrace (ie distributed message tracing)
  - monitoringEvent
  - networkInterface
  - more added as needed

Recap: Avro benefits

- Benefits to the Business
  - Data integration to answer business questions
- Benefits to data and schema governance
  - Data standardization
  - Data integrity
  - Standardized documentation
  - Data lineage
  - Clear compatibility criteria

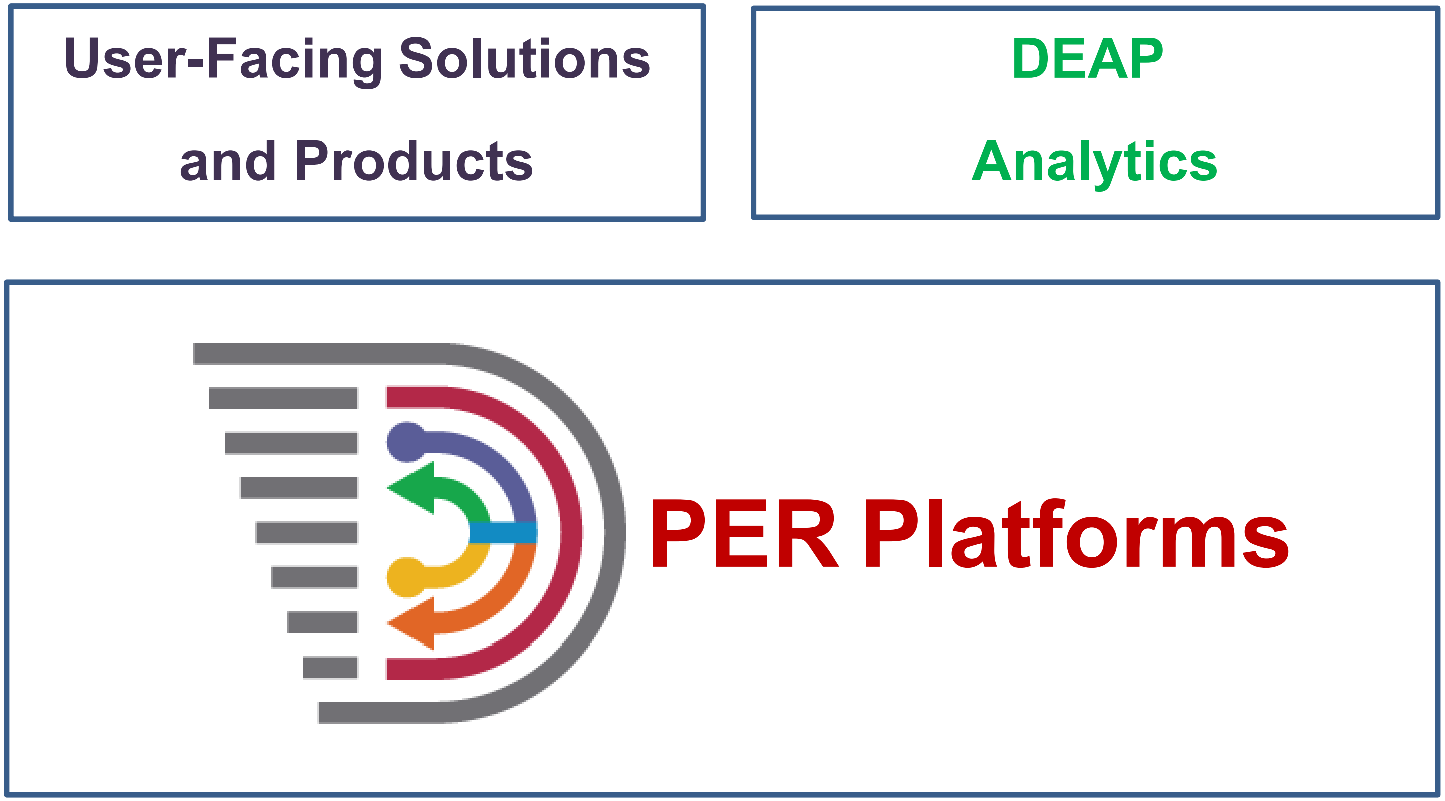
Avro in the context of data governance:
Apache Atlas

Hortonworks

COMCAST

Data Engineering Analytics Platform

DEAP



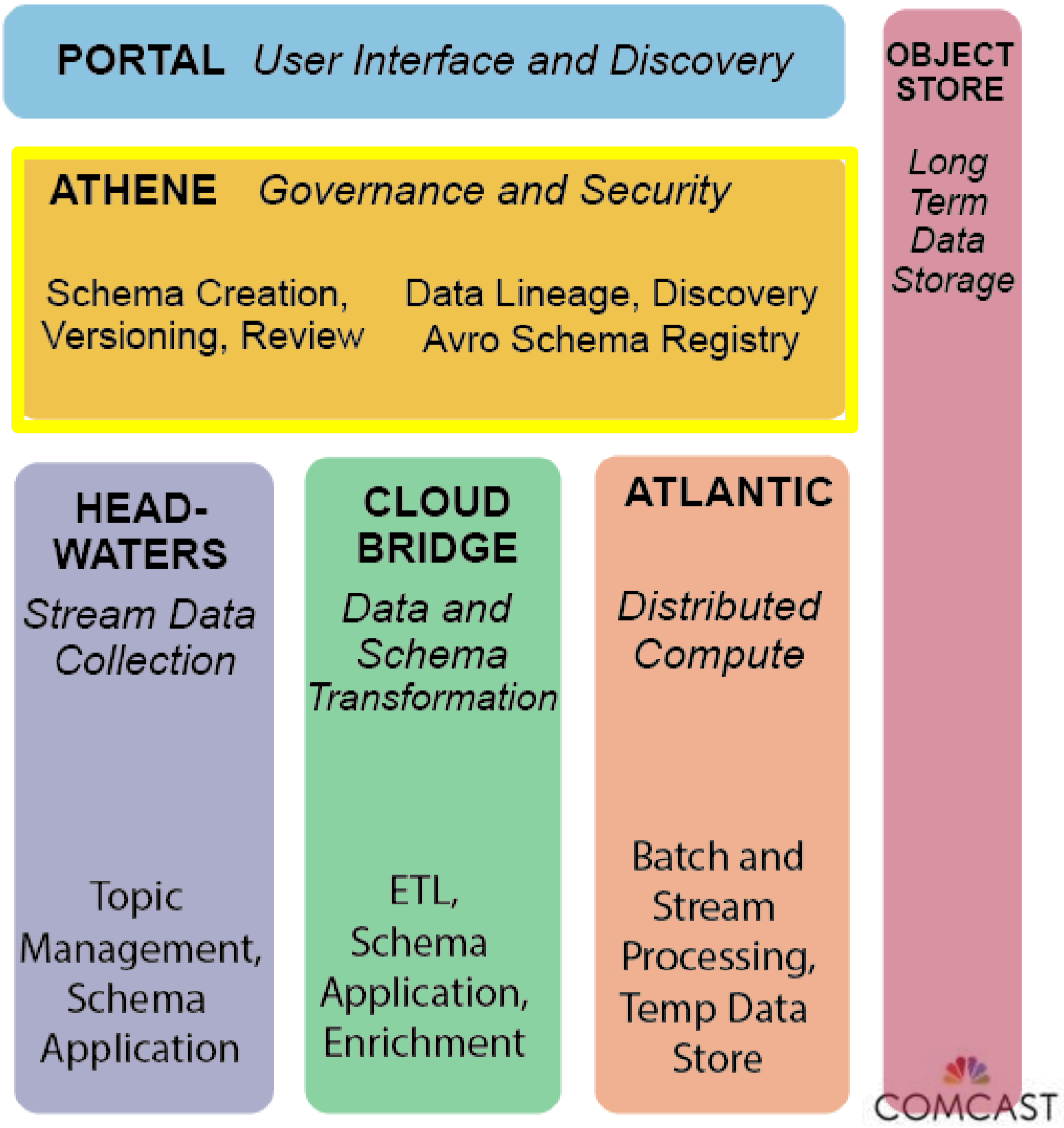
User-Facing Solutions and Products

DEAP Analytics

PER Platforms

DPER

Architecture



PORTAL *User Interface and Discovery*

ATHENE *Governance and Security*

Schema Creation, Versioning, Review

Data Lineage, Discovery Avro Schema Registry

HEAD-WATERS

Stream Data Collection

Topic Management, Schema Application

CLOUD BRIDGE

Data and Schema Transformation

ETL, Schema Application, Enrichment

ATLANTIC

Distributed Compute

Batch and Stream Processing, Temp Data Store

OBJECT STORE

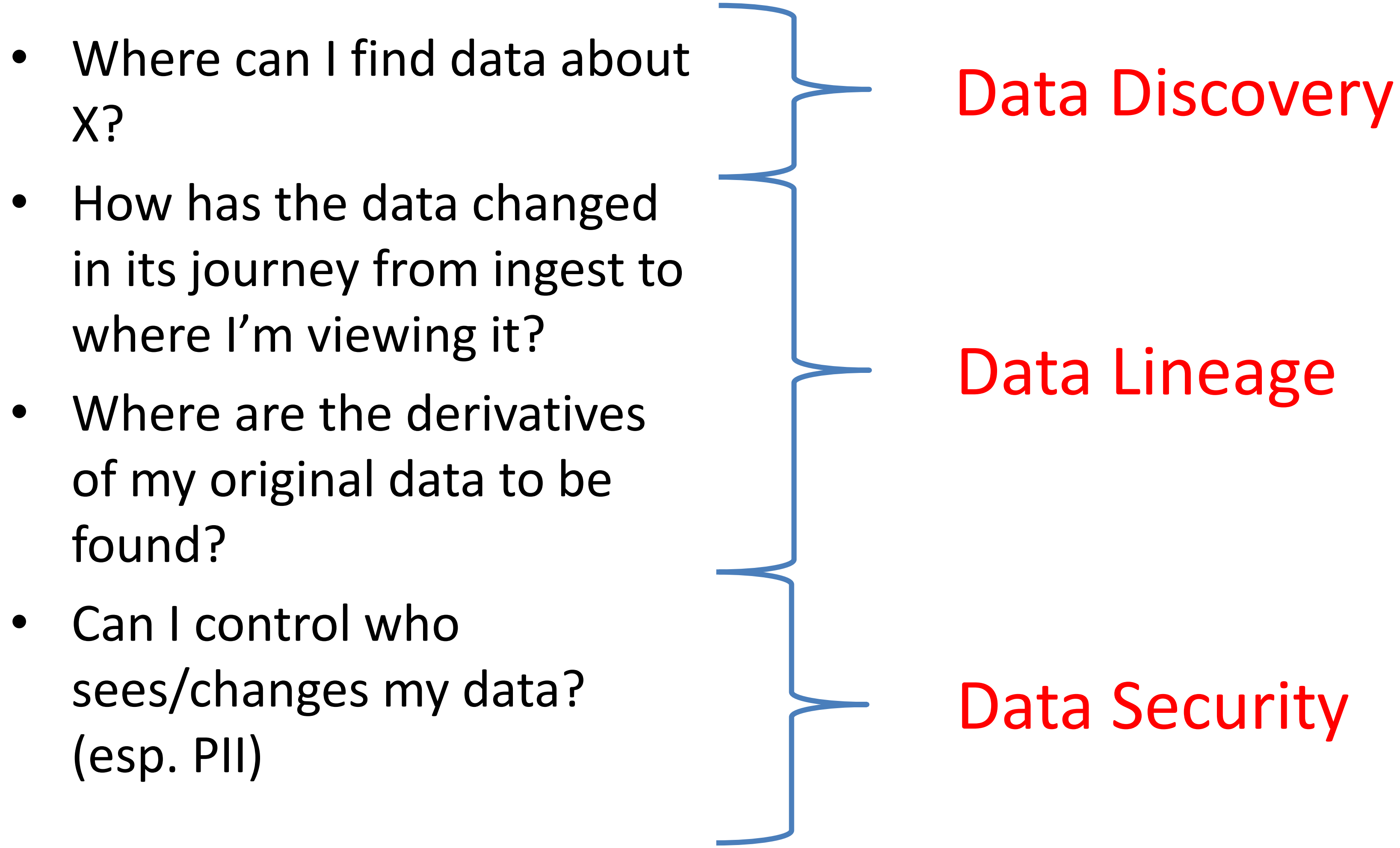
Long Term Data Storage

COMCAST

Athene noctua, “little owl”

Companion of Athena, Greek goddess of Wisdom, Justice, and Good Governance.

Questions: Data Governance



- Where can I find data about X?
- How has the data changed in its journey from ingest to where I’m viewing it?
- Where are the derivatives of my original data to be found?
- Can I control who sees/changes my data? (esp. PII)

Data Discovery

Data Lineage

Data Security

Apache Atlas

- Data Discovery, Lineages
  - Browser UI
  - Rest and Java APIs
  - Synchronous and Asynchonous messaging
- Integrated Security (Apache Ranger)
- Schema Registry as well as Metadata Repo

Open Source
Extensible

Atlas Extensions for Schema Registry

- New typedefs: avro_schema, avro_record, avro_enum, etc
- Extensions to Kafka topic type
  - sizing parameters
- Reciprocal links between topics and schemas
- Schema evolution
  - Versioning
  - schema lineage process
  - B/F/Full compatibility
- Better handling of JSON-valued attributes in UI

Avro Schema Registry for Data at Ingest

- Browsable hierarchical schemas
- Avro schemas corresponding to kafka topics (or no kafka topics)
- Schema lineage using Atlas processes
- Schema evolution and compatibility
- Kafka topic search capabilities
- Contact person for each kafka topic/schema
- API for fully-expanded avro schemas for serde

Data Lineage for Data at Rest

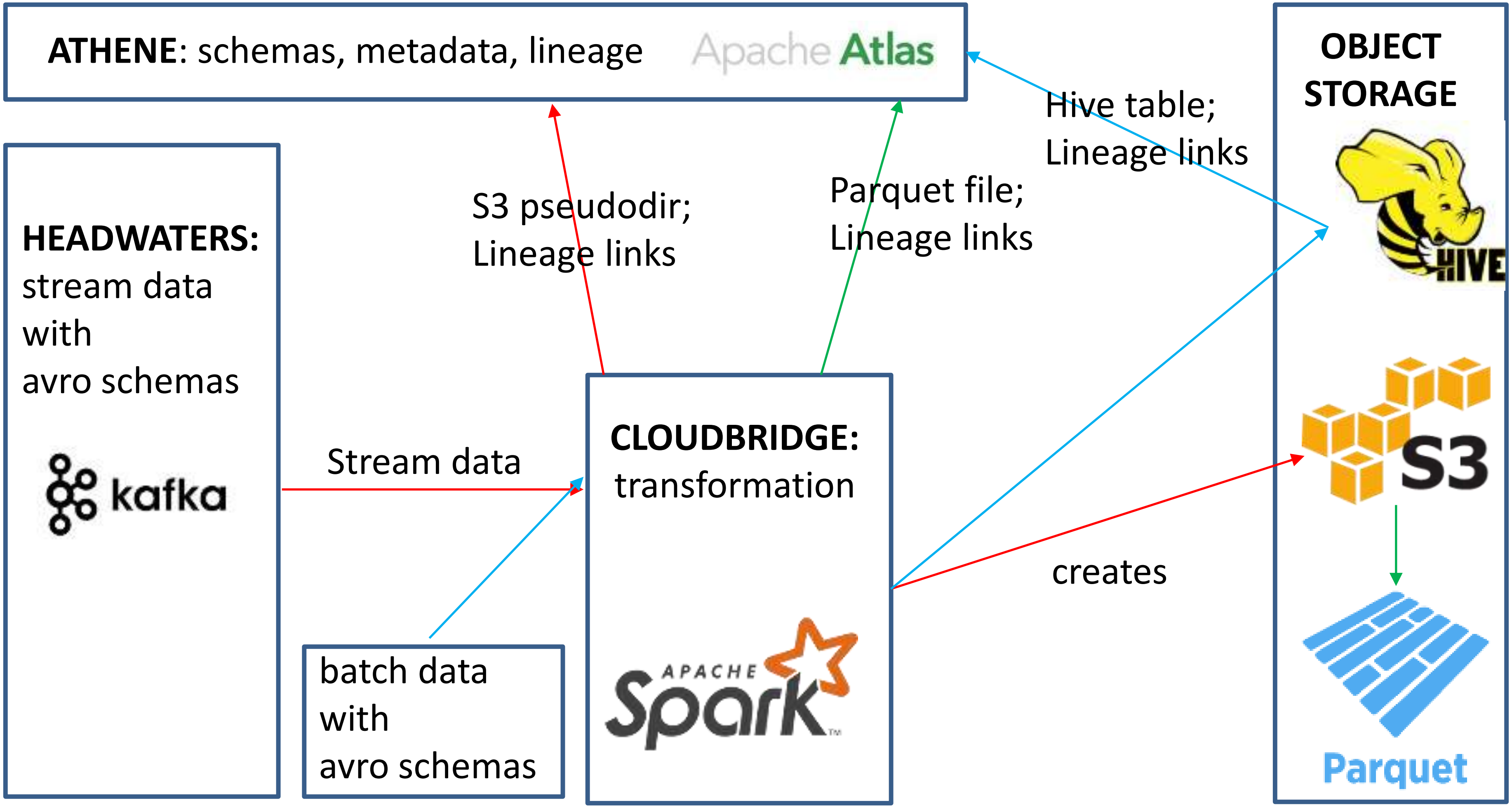
For any data item in any **community** data store, we must be able to identify the corresponding avro schema in the registry

- Compressed avro binary files in long-term object storage
- Hive or parquet tables derived directly from avro data
- Data derived/aggregated from hive tables
- Avro-serialized JSON data in NoSQL
- etc

Atlas Extensions for Data Lineage

- S3 objects (also linked to schemas)
- Hive tables and avro schemas linked
- Link to slack help channel in UI
- Java library to facilitate creation and linking of entities, using asynchronous messaging
  - create S3pseudo, link it to avro
  - Update kafka sizing

Platforms work together for data governance



ATHENE: schemas, metadata, lineage

HEADWATERS: stream data with avro schemas

batch data with avro schemas

CLOUDBRIDGE: transformation

OBJECT STORAGE

Stream data

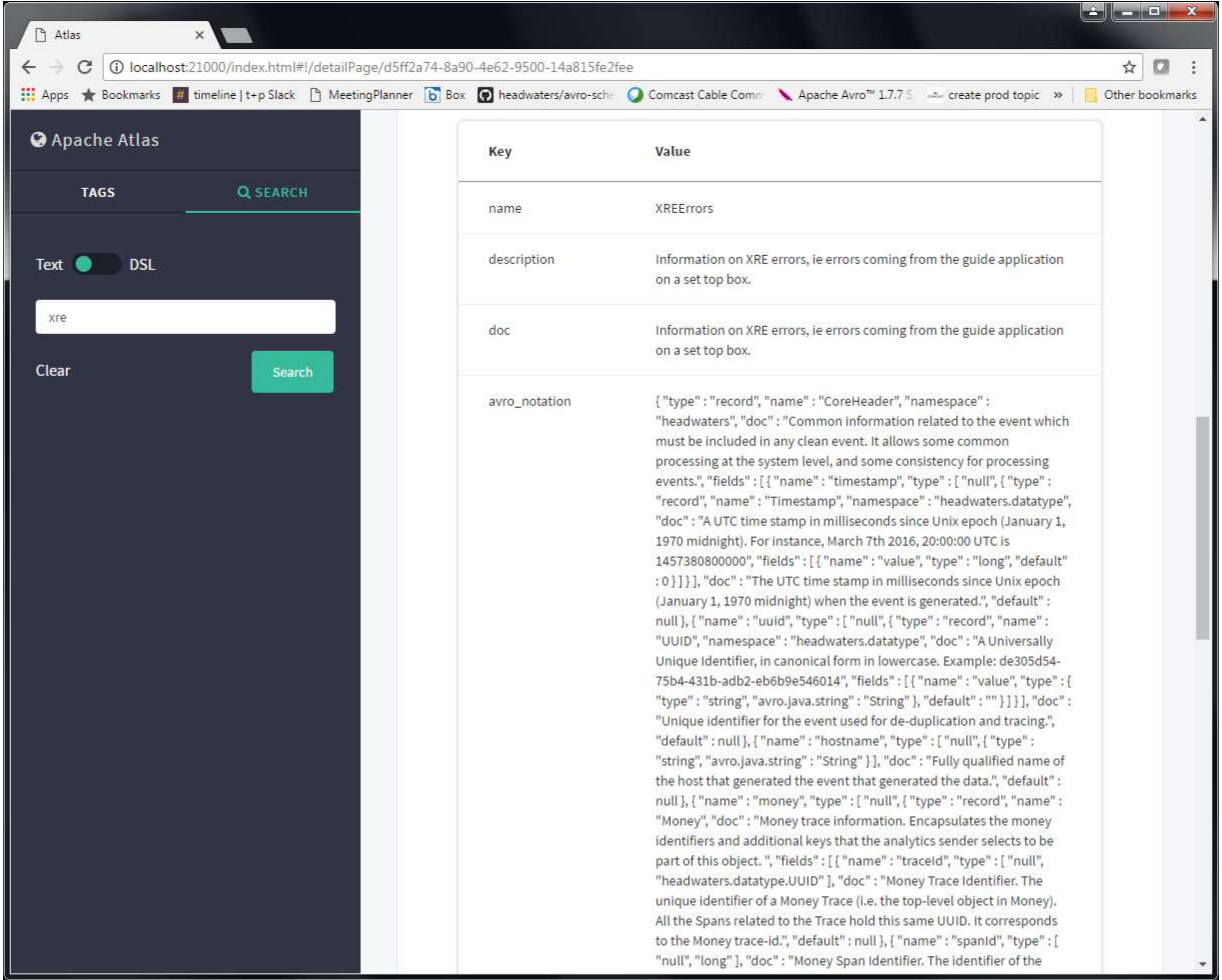
S3 pseudodir; Lineage links

Parquet file; Lineage links

Hive table; Lineage links

creates

Demo here

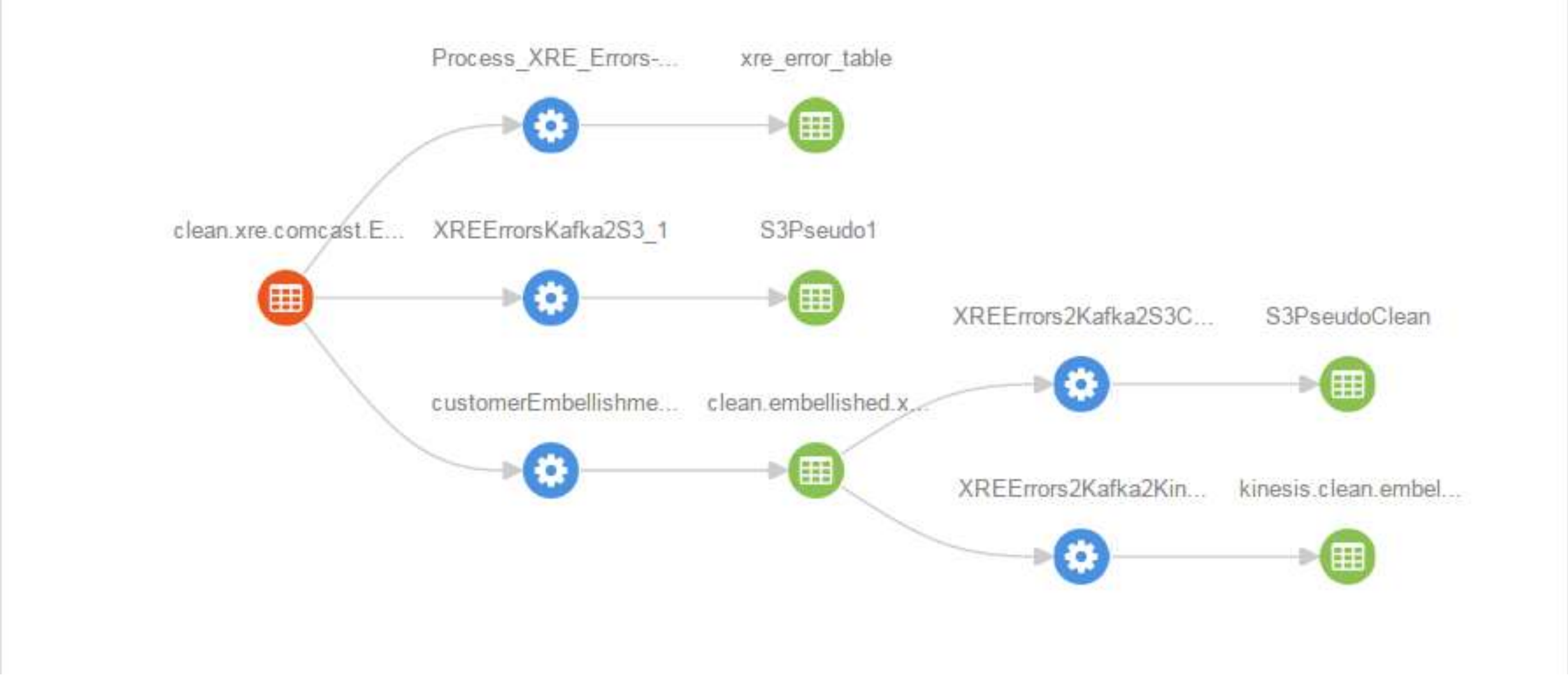


Key	Value
name	XREErrors
description	Information on XRE errors, ie errors coming from the guide application on a set top box.
doc	Information on XRE errors, ie errors coming from the guide application on a set top box.
avro_notation	{ "type" : "record", "name" : "CoreHeader", "namespace" : "headwaters", "doc" : "Common information related to the event which must be included in any clean event. It allows some common processing at the system level, and some consistency for processing events.", "fields" : [{ "name" : "timestamp", "type" : ["null", { "type" : "record", "name" : "Timestamp", "namespace" : "headwaters.datatype", "doc" : "A UTC time stamp in milliseconds since Unix epoch (January 1, 1970 midnight). For instance, March 7th 2016, 20:00:00 UTC is 1457380800000", "fields" : [{ "name" : "value", "type" : "long", "default" : 0 }] }], "doc" : "The UTC time stamp in milliseconds since Unix epoch (January 1, 1970 midnight) when the event is generated.", "default" : null }, { "name" : "uuid", "type" : ["null", { "type" : "record", "name" : "UUID", "namespace" : "headwaters.datatype", "doc" : "A Universally Unique Identifier, in canonical form in lowercase. Example: de305d54-75b4-431b-adb2-eb6b9e546014", "fields" : [{ "name" : "value", "type" : { "type" : "string", "avro.java.string" : "String" }, "default" : "" }] }], "doc" : "Unique identifier for the event used for de-duplication and tracing.", "default" : null }, { "name" : "hostname", "type" : ["null", { "type" : "string", "avro.java.string" : "String" }], "doc" : "Fully qualified name of the host that generated the event that generated the data.", "default" : null }, { "name" : "money", "type" : ["null", { "type" : "record", "name" : "Money", "doc" : "Money trace information. Encapsulates the money identifiers and additional keys that the analytics sender selects to be part of this object. ", "fields" : [{ "name" : "traceId", "type" : ["null", "headwaters.datatype.UUID"], "doc" : "Money Trace Identifier. The unique identifier of a Money Trace (i.e. the top-level object in Money). All the Spans related to the Trace hold this same UUID. It corresponds to the Money trace-id.", "default" : null }, { "name" : "spanId", "type" : ["null", "long"], "doc" : "Money Span Identifier. The identifier of the

Visualizing Data Lineage

Apache Atlas

Hortonworks



COMCAST

What's Next

- More types
  - Independent kafka2S3 service that messages Atlas
  - More transforms as first-class processes
    - Generic transform library with avro schemas expressing inputs and outputs
  - Zeppelin and other notebooks
  - Other data sources in our hybrid environment
    - Circonus checks

Summary

- Avro for Schema Governance
  - Our lingua franca for end-to-end metadata
  - Comcast BPs make Avro even stronger
- Atlas for Data Governance
  - Metadata Browser and Schema registry
  - Comcast typedef extensions to Atlas
    - Avro schemas, records, enums, maps, arrays, etc
    - Add attributes to already existing types (eg kafka topics)
    - Data sources outside Hadoop ecosystem
    - Transforms, schema versioning, lineage, compatibility
  - Platforms and Atlas work together for lineage

Challenges and Solutions

Challenge	Solution
Creating conformant avro schemas is not trivial	Detailed documentation, sample code in Java, Python, C#, GO, etc; team of reviewers
Avro schemas are annoying to create in a text editor	Avro schema builder UI—in beta now
Avro manual schema review process was originally too slow	Trained more reviewers, streamlined processes; Avro schema reviewer UI—in beta now
Not everyone can publish data in Avro, but they can in JSON	Use Apache NiFi to convert from JSON to JSON-serialized avro
No one likes to be “governed”; Some developers saw creating Avro schemas as a drag on their productivity	Top-down mandate; came to recognize the benefits of a schema in data handling: excitement from analysts, end-user applications
Finding data governance tooling to handle metadata and lineage for our hybrid environment	Apache Atlas: Open source, extensible solution!
Need a schema registry that allows browsing, highlights our BPs, also serves up avro for serde	Atlas: Create avro_schema type, extend kafka topic type, represent schema evolution
Informing Atlas of changes where there aren’t built-in “hooks”	Extensibility!! Java library hides Atlas-specific syntax, makes it easy for other platforms

My collaborators

Hortonworks

Vadim Vaks
Sr. Solutions Architect
Hortonworks

Thank You!

COMCAST

Barbara_Eckman@Cable.Comcast.com