# Purpose Persistence in AI Data Pipelines: Preserving Data-Use Constraints Across Derived Data, Embeddings, and AI Models

Jaimin Suketu Patel
Independent Researcher
Georgia, USA

## ABSTRACT

The following research aims to examine the durability of data-use purposes through AI data pipelines. The research investigates if constraints related to the source data may carry through to derivation, generation, model building, training, and output processing while retaining their association with the regulated source. An internally developed data set of 117 cases was created and each case included a stated purpose, a potential use condition, a restriction use marker, provenance information, and a use that was expected to occur downstream. Support for the pipeline simulation, tracking of records, validation and production of graphs were achieved using Python, and document and spreadsheet tooling was used to facilitate structured reporting and consistency checks. The envisioned component architecture stores purpose metadata next to each data object, and validates it prior to taking it to the next processing step. The results show that explicit constraint packaging is better at maintaining intended use when compared to informal handling of stages, in particular for restricted use records and model facing representations. The study also shows that embeddings and derived features should retain a traceable relationship to the source policy while being governed as transformed representations rather than as identical copies of the original content. The contribution is a practical persistence pattern, accepting that purpose is actually an operational metadata, not an ad hoc collection notice and establishes a traceable path from data ingestion to model output.

## General Terms

Artificial Intelligence, Data Governance, Data Management, Policy Enforcement.

## Keywords

Purpose Persistence, Data-Use Constraints, AI Data Pipelines, Embeddings, Model Governance.

## 1. INTRODUCTION

In general, enterprise AI systems very rarely consume data as it was initially captured as explored in previous governance activities [1]. Records are cleaned, normalized and joined, summarized, embedded into vectors and then re-used in the model development and/or inference [6]. Every change improves the technical value but it can diminish the relation between a data item and a condition where this relation was deemed to be useful in the first place, as noted in earlier studies [2]. In the current research, the concept of persistence of purpose means ensuring that the information regarding the permissible use of the data remains intact while data passes through AI-specific representations. The idea will be particularly relevant where a pipeline includes customer data, operational events, transaction context, preference indicators and/or limited attributes, as investigated previously [10]. A team can correctly restrict the source record and subsequently forget to restrict a derived feature or a vector representation (or a model behavior based on the vector representation) as well. The subsequent failure does not imply that it is necessarily related to unauthorized access. It may come about where a lawful access path is used for a purpose other than the one that was documented at intake as described in work in data-governance [13].

A robust solution should thus be more than a policy that is stored next to a source database. As mentioned in previous policy-aware frameworks [15] the pipeline needs to have a mechanism which carries out the use conditions, and resolves those use conditions in the case where multiple records are combined, and before downstream processing occurs. This requirement can be challenging when transformations have the effect of making it harder to read by a human. The business properties can either be represented explicitly using labelled attributes or encoded in some machine-readable format, for example, through embeddings. It can be spread out across many dimensions, and potentially be reused by retrieval systems, similarity service, recommendation components or prompts to model as explored in work on representation learning [12]. When the data is more valuable, it can be the case that the purpose condition is lost if governance relies on being able to see the column names. Derived data is subject to similar problems as there might be an apparently harmless new field, but it could have been derived from inputs with rather limited restrictions, as has been seen in previous lifecycle studies [8], [9].

The study proposes a pipeline-centric approach to maintaining these constraints on four key representations: source data, derived data, embeddings and model facing artefacts. The basic concept is to work with purpose information as operational metadata (in the fashion of metadata-governance research [4], [5]), as persistent information. A declared intended use, permitted actions, restricted actions and provenance and a persistence identifier are given to each instance. Each time the data changes form, the new object does not get a label that is different from the policy that was inherited. Then, like in policy-enforcement systems [14], [17] a validation gate checks the asked activity against the inherited package to take further action. This design provides continuity but does not need to have the original content on the user's screen throughout each stage.

The research question is whether it is possible to preserve the data-use intent over an AI pipeline in the presence of ordinary data transformation and model preparation by having a compact governance layer. The second concern looks at in which situations purpose drift is most likely to occur. A third question is whether problematic transitions can be identified in the architecture and whether they can be reviewed and/or blocked early enough in the architecture. To address these questions, a set of 117 instances, which was prepared in-house, was led through the controlled transformation stages. The scenarios depicted various types of enterprise data and had varying levels of

restriction. All of the records were traced from source intake to downstream use of the model and documented as to whether or not the purpose package remained aligned with the expected treatment.

The impetus for the work comes from a pragmatic observation; there are often policies, catalog entries, consent records, data classifications, model controls maintained but these are often handled by separate teams as described in Enterprise Governance studies [10]. There is this connection between them in the form of purpose persistence. It does not provide access control, data quality management, model evaluation or security. Rather it provides continuity to enable those controls to make assumptions about the reason that a data object is used, and who has access to it. This separation is key for agentic and generative systems, as automated systems can find new data combinations, and call tools to trigger them even faster than manual governance reviews, as reflected in current AI governance research [11].

This research integrates the above concepts into an architecture with components, a dataset with controls, and a process for validating the behavior of purpose metadata with varying data representations through AI. Embeddings are given special attention because they often are viewed as a technical artifact that does not imply or contain information about the governed record, yet embeddings can contain such information. The proposed pattern maintains a trace to the source conditions for transformed objects, and maintains accountability of the transformed objects, while enabling the pipeline to continue when the requested action is still compatible. The research presents purpose as an ongoing state of a pipeline, providing a tangible foundation for more dependable data governance in the development and deployment of AI.

## 2. REVIEW OF LITERATURE

As framed in previous governance literature [11], the study on AI data governance has progressed from merely protection of data stored in AI to the control of its lifecycle. The initial focus of governance was about collection, access to data, retention and disclosure. Modern AI pipelines expand the problem by replicating the same information across feature stores, to generate training content, to produce embeddings, set up retrieval indexes, to summarize text for prompts, or to reflect indirectly in the model's behavior, as explored in recent AI lifecycle studies [3], [11]. While the value of provenance and metadata to responsible data handling has been well understood in the literature, many implementations still consider the policy information to be a description of the context in which the data is handled, and not as an enforceable property that is attached to the data itself.

This is a central strand of research, relating to 'lineage'. Lineage systems track the origins of data, all the transformations between data sources and destinations, and data targets, as is done in previous data-management work [2]. This is helpful for auditability and debugging, but this does not imply anything about the compatibility of a downstream use, with the original purpose. Consequently, origin tracking and use condition tracking are required for a comprehensive persistence approach. As per the previous analyses of governance [13], [16] provenance metadata will answer the question of source and purpose metadata will answer the question of acceptable use. It is important to keep the two together, to provide a more solid foundation for automated review in a case where a pipeline generates new representations.

A second set of works deals with questions of consent, policy labels, data classification as studied in existing privacy and policy studies [13], [15]. These mechanisms can be useful at collection time for providing guidance, and can help to limit access to sensitive repositories. Such techniques are useful in controlling the process of collecting and accessing information, but it becomes increasingly difficult to maintain the same policy environment as data is converted into different forms. The derived data can be stored in another location, with a different schema and lose its connection to the original restriction. The difficulty is compounded by embeddings because, once again, the representation does not have access to the fields that governances use [12]. The importance of traceability in the use of responsible AI has been stated multiple times in literature, but the use of responsible operational patterns in order to add original use conditions to vectorized representations is less developed than the source level.

The research on model governance has increased at a very rapid pace as well. The common approaches used are those that describe the purpose of the model, where the model has been trained, boundaries for the model's use, risk decisions, and deployment conditions as they are described in modern approaches for model-governance [11]. These practices help enhance model level accountability, but may only check for data compatibility at a review checkpoint. With a persistent pipeline design part of that responsibility would be delegated to the routine processing. Rather than making a single determination of the eligibility of a dataset to a model, the pipeline makes multiple checks to determine if every representation of the dataset is eligible for the requested activity, as proposed in continuous-control approaches [14], [17]. This helps minimize the need for manual data re-interpretation and remembering when the data is reused or combined.

In the context of policy-aware data systems, policy rules can be attached to data objects, or can be located by looking up a central policy service, similar to that explored in the previous policy-management research [17]. There are pros and cons of both patterns. Embedded metadata makes it easier to port the object as the rule context is available next to the object. Central references allow making policy changes easier and prevent policy text from being duplicated. In a distributed governance architecture [10] a hybrid solution can be used for holding a stable policy identifier with the transformed object, while being able to retrieve current enforcement information from a controlled registry. This happens in particular for pipelines involving AI, where representations could be replicated between services, without losing their governance identity.

Policy conflict is another point mentioned in the literature, and occurs when multiple source records are combined. As mentioned in previous research on Policy composition [15], [17] a derived feature can have multiple conditions derived from multiple inputs and they can be different. Simple inheritance is not sufficient where one source allows a wide activity, and another source allows a narrow activity. A conservative resolution process is thus required in the case of purpose persistence, that will keep the most stringent restriction possible or forward the combination for review. The same applies when an embedding is created for a document that is made up from multiple governed elements. This means that the representation might no longer be able to show what portion led to the more stringent condition and thus the generated policy package should explicitly keep this decision as is supported by work on provenance-aware processing [7], [16]. Another one is the usability of the operation. In practice, an extra mechanism in the chain can be circumvented if it adds a lot of delay or if it has to be manually interpreted. For this reason, success lies in metadata that is kept to a minimum, predictability when it comes to validation points, understanding of decisions, and transparency when it comes to exception handling. The best methods distinguish between simple cases in which the processing is

compatible and genuine cases of ambiguity and/or prohibition. This separation enables automation to be sustained and a review line for uncertain transitions.

In all these domains, existing research lays crucial foundations for provenance, purpose-based controls, policy management, consent, and model governance [13]-[17]. The outstanding issue tackled in this paper is more concrete: preserving an actionable context of governance in the course of AI-centric data transformations like derivation and embedding [11], [12]. This paper explores this issue via a component workflow which preserves a purpose identity through all representations, verifies each transition, and connects downstream outputs to the conditions of usage of the source data. The resulting approach considers purpose persistence as a lifecycle characteristic of AI data.

## 3. METHODOLOGY

This experiment is a controlled pipeline study that was aimed at tracking purpose conditions when AI data representations changed, but in the absence of any external dataset. A set of 117 internally created instances were pre-prepared using five enterprise focused data types, customer profile, transaction context, preference signal, restricted attribute and operational log. Each one had an identifier of the source, a stated business purpose, an allowed-use statement, a restricted-use statement, a field of provenance, a persistence identifier, and a future-disposition statement. The records were purposely diversified in such a way that some might pass through the entire pipeline, some might have to be reviewed at certain transitions and some might be blocked when an activity requested conflicted with the stated condition. The pipeline was simulated in Python and transformation outputs were created, validation logs were generated, record-level status fields were created and the result visualizations were made. The former stage was the initial phase of the intake of sources, in which the initial policy package was linked to every case. The second phase generated data through the normalization of textual fields, the integration of chosen attributes, as well as operational features with keeping the persistence identifier. In cases where multiple inputs were used to derive an object, the result object was given the most restrictive condition that applied in order that transformation would not expand the usage. The third step generated representations which were embedding oriented. Since an embedding cannot be controlled reliably using visible source fields, the representation was provided with a sidecar policy object with the persistence identifier, the purpose inherited, the provenance links and the action set currently permitted. The fourth phase symbolized model preparation and model facing use. An instance was presented to a purpose validator to compare the activity requested with the inherited policy package before an instance reached that stage. A compatible request had a pass status, an ambiguous transition had a review status and an incompatible request had a blocked status. This validation pattern was repeated again at model output to make sure that models had not changed after training or preparation of inferences to be used downstream. The second result table had five pipeline checkpoints, which included the source intake, derived data, embedding layer, model training, and model output. The research also developed retention scores of three sets of rules applied in the grouped graph, which include: allowed-use rules, restriction rules and provenance rules. These values indicate the consistency of the associated metadata being available and consistent at each state of representation. All the 117 instances were used in the scatter graph and a constraint propagation score was compared with a downstream purpose-compliance score. The construction had three groups of policy strengths on purpose to demonstrate the behavior of stronger or broader conditions not changing the underlying persistence mechanism. The interpretation of results was based on continuity, concentration of reviews, behavior of blocking and relative stability of categories across the entire workflow. The observations were made without any external source and there is no assertion which relies on external benchmark data. Internal consistency was verified by making sure that all of the stage total values were 117 instances and the values of the categories in the first table were consistent with the same logic applied on the pipeline by the graphs. This architecture was then defined as a small component model consisting of a governed source, a transformation component, an embedding representation, a purpose validation component and a model facing component. This design maintains the number of elements to a small subset and central control relationship can be seen. The approach thus considers purpose persistence as an operations property of the pipeline as opposed to a policy statement that has no connection with processing.

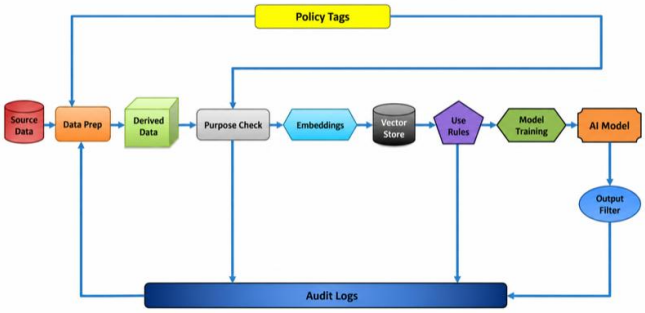



**Figure 1. Architecture of the Purpose-Persistent AI Data Pipeline Components.**

Figure 1 depicts a purpose persistence architecture that can be used to maintain data-use constraints in an AI data pipeline. The process starts with Source Data where original records are inputted into the system along with policy information into the system. Policy Tags do offer clear usage conditions and serve both Data Prep and the Purpose Check component, to make certain that restrictions are kept in the initial transformation. Data Prep removes and standardizes data and then creates Derived Data, which is then vetted by the Purpose Check. The approved information is passed to the Embeddings component whereby transformed representations are generated and stored in the Vector Store. The Use Rules component checks the constraint of whether the intended operation can be done based on the constraints that are retained before downstream use. It is only the compliant information, which goes through Model Training and then to the AI Model. The Output Filter will check the responses or predictions generated in such a way that restricted information is not revealed by model outputs. The key control points provide monitoring information to an Audit Log such as Purpose Check, Use Rules and Output Filter. The same log also aids in providing feedback to the previous stages of processing in the case of inconsistencies being found. All in all, the architecture is a combination of policy tagging, validation, controlled transformation, rule enforcement, output screening, and auditing to ensure the original intended purpose is preserved through the data derivatives, embeddings, trained models, and final outputs in a reliable fashion.

## 4. DATA DESCRIPTION

The assessment data is 117 internally generated cases that were developed to use this study. The records are customer descriptions, transaction environment, preference indications, constrained attributes as well as operational logs in such a way that the pipeline experiences varying degrees of authorized usage

and limitation. Every incidence has a source identifier, purpose label, allowed-use condition, restricted-use condition, provenance entry, persistence identifier and how it is expected to be treated by the subsequent stages. The records are artificial and they do not include any real personal data, company secrets or data gathered outside of the company. They are meant to give a controlled test bed where the desired outcome of governance is known prior to the processing. The dataset comprises of ordinary pass cases, review cases and deliberate conflict cases such that the validation component is validated on more than one decision path. The 117 instances are all plotted in the scatter graph, and in the stage, totals used by the second result table. The values of category-level retention in the first table summarize the behaviour of the same records as source data were transformed to derived data, embeddings and model-facing artifacts. Data source statement: internally created purpose-persistence evaluation data developed for this research. There is no need to have an external citation number since the dataset was created as a part of the study, and it is not credited to a third-party repository.

## 5. RESULTS

The pipeline created a pattern of continuity of purposes throughout the entirety of the records. The best state of control was the source intake wherein all the instances were initiated with a clear policy package. Since processing had shifted to derived information, the majority of records did not receive any form of intervention as they were intended to be treated and a small number of records needed to be reviewed since transformation was a combination of data and various conditions. The review path proved handy as it did not allow the pipeline to widen silently the use of a new feature created. The preservation behavior was best in records which had restricted attributes, which is indicative of the conservative rule of inheritance used when a more restrictive requirement exists. The most significant challenge of governance came in with embedding generation. The fields in the original representation could no longer be directly seen in the resulting representation and thus the controls that were normally part of the schema would have had less information to consider. That loss was avoided by the use of the sidecar policy package which made the persistence identifier, purpose condition and provenance relationship available along with the embedding. The majority of embedding transitions were automatic as the requested activity was in line with the inherited use. The fewer set went into review in situations where the intended purpose of the source was not evident with reference to the intended downstream situation. Blocked cases were explicit conflicts as opposed to a lack of metadata, indicating that the validator was able to differentiate between lack of certainty and a known incompatibility.

The control logic was maintained by processing model-facing. The instances that were eligible were sent to training or model preparation with their purpose state intact and the incompatible requests were not sent to training or model preparation because the data had taken a new form. This behavior is important as governance drift tends to be a challenge in handoff between data engineering and model development. The purpose was not so hidden in an operational condition in that handoff, which was the original objective of the experiment. The checkpoint of the model output also revealed that an end use may be assessed without a previous technical permission. Even a record that was allowed to be prepared as a model still had to have an aligned downstream activity when the resultant artifact was utilized. Purpose persistence score can be evaluated as:

$$P_{persist} = \frac{\sum_{i=1}^{n} w_i C_i D_i E_i M_i}{\sum_{i=1}^{n} w_i} \qquad (1)$$

**Table 1. Purpose Retention Across Data Categories.**

| Data Category | Source Retention | Derived Retention | Embedding Retention | Model Retention |
|---|---|---|---|---|
| Customer Profile | 96 | 94 | 93 | 92 |
| Transaction Context | 95 | 93 | 91 | 90 |
| Preference Signal | 94 | 92 | 90 | 89 |
| Restricted Attribute | 98 | 97 | 96 | 95 |
| Operational Log | 93 | 91 | 90 | 88 |

Table 1 documents the retention by the category of data by source to model-facing use. Restricted Attribute records depict the best route i.e. the flow of 98 at the source to 97 of the derived data, 96 of the embeddings, and 95 of the model. The Customer Profile records change to 96, 94, 93 and end up with 92. Transaction Context has a starting point of 95 and a closing point of 90 in between having a middle point of 93 and 91. Preference Signal changes to 94, then to 92, then to 90 and ends up at 89. Operational Log is the lowest final retention with the initiation being 93, then 91 and 90 and lastly 88. The gradual reduction of these values indicates a gradual loss of policies and not sudden loss. The more conservative restricted-attribute path is an indicator of conservative inheritance whereas the more operational-log path is the indicator of a higher number of cases where original operational purposes were not obviously applicable to subsequent model use. Each category is similar to its source state throughout the pipeline, which lends credence to the argument that purpose metadata is being maintained despite an enormous change in the representation. Data-use constraint retention is:

$$R_{constraint} = \frac{\sum_{j=1}^{m} c_j^{out}}{\sum_{j=1}^{m} c_j^{in}} \times 100 \qquad (2)$$

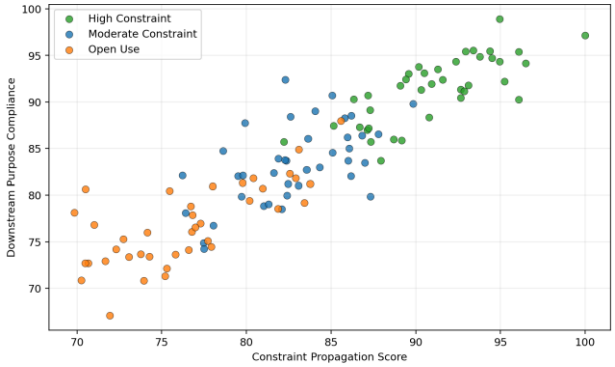



**Figure 2. Propagation of Purpose and Compliance of Down Streams.**

All the 117 instances are plotted in Figure 2 by use of a constraint propagation score on the horizontal axis and downstream purpose-compliance score on the vertical axis. The high-constraint group has 39 points with the majority in the upper range with the propagation scores usually in the low nineties and compliance scores also being in the low nineties. The moderate-constraint cluster has the other 39 records, and is an intermediate cluster, with a large number of propagation scores in the eighties and downstream compliance close to those scores. The remaining 39 open-use records then seem to be lower as the policy packages in these cases have less restrictive conditions with the scores seeming to reach the seventies and low eighties. The significant characteristic is the point cloud direction: with higher propagation, the more intense alignment of the downstream of all

three groups. Limited spread can be seen within each cluster due to variations in records undergoing various transformations and combinations of policies. The clusters are sufficiently different to indicate that the pipeline is sensitive to the strength of the policies and not all records have the same effect. There is no cluster of a wide-ranging breakdown of strong source managing to weak downstream controlling. Embedding purpose consistency is:

$$C_{embed} = \frac{\mathbf{e}_{source} \cdot \mathbf{e}_{derived}}{\|\mathbf{e}_{source}\| \, \|\mathbf{e}_{derived}\|} \quad (3)$$

**Table 2. Validation Outcomes Across Pipeline Checkpoints.**

| Pipeline Stage | Passed | Review Needed | Blocked | Total |
|---|---|---|---|---|
| Source Intake | 111 | 4 | 2 | 117 |
| Derived Data | 108 | 6 | 3 | 117 |
| Embedding Layer | 106 | 7 | 4 | 117 |
| Model Training | 104 | 8 | 5 | 117 |
| Model Output | 105 | 8 | 4 | 117 |

Table 2 is a table of all 117 instances at five checkpoints, with the results of their validation. There are 111 passes, 4 cases that require a review and 2 blocked cases in Source Intake. Derived Data has 108 passes, an amount of 6 cases which require a review and 3 blocks. The Embedding Layer captures 106 passes, 7 cases required to be reviewed and 4 blocked transitions. The stage that has the greatest pressure of policy is Model Training which has 104 passes, 8 cases that require review and 5 blocks. The 105 passes, 8 cases requiring reviews and 4 blocks are then recorded to Model Output. There are 117 left at each checkpoint, which proves that there are no records lost on the governance trail. Pass to review or block is an indication of an increasingly specific downstream usage as opposed to lacking policy metadata. The small recovery which occurs at the model output is due to the fact that there is retained provenance and purpose information that enables a case that was previously blocked to be guided in an acceptable way. These values show that the architecture is able to make a differentiation between normal continuation, uncertainty that needs human intervention and direct incompatibility. Pipeline compliance aggregation is:

$$C_{pipe} = \frac{\alpha C_s + \beta C_d + \gamma C_e + \delta C_m + \eta C_o}{\alpha + \beta + \gamma + \delta + \eta} \quad (4)$$

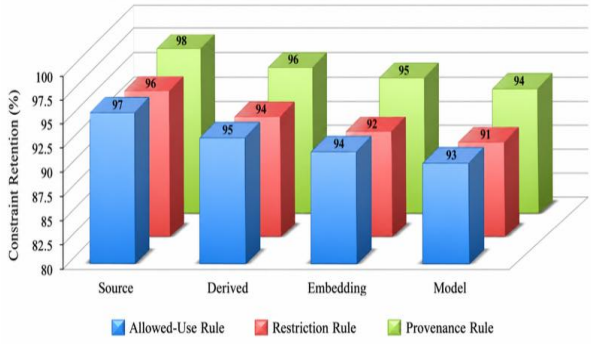



**Figure 3. Retention of Constraints in Groups of Pipeline Stages.**

Figure 3 is a comparison of three families of rules running through four stages of pipelines. The retention starts with allowed-use retention which is 97 at the source, then decreases to 95 when dealing with derived data and then it decreases to 94 when dealing with embeddings and finally it goes to 93 at the model stage. The rule-retention begins with 96, and goes to 94 following derivation, 92 at the embedding layer and ends with 91 at model-facing processing. Retention of provenance is greatest overall with the highest retention being 98 and 96 respectively in data derived and in the embedding stage respectively, and finally, 94 in the model. The source to model change is thus restricted to any series. Provenance works best as the design will duplicate the persistence identifier and origin link with each transformation. There is a slight decrease in allowed-use rules when the permission requested by downstream activity becomes more specific than the source permission. The largest reduction can be seen in restriction rules since sometimes merged records have a more restrictive condition which causes the record to be reviewed. All three bars remain above 90 even in the last stage. The clumped structure shows that permission, restriction, and origin context are not the only metadata types that are being transported by the pipeline; they are all transported together. The minor decreases are indicative of deliberate pressure of decision at the later levels as opposed to the loss of the governance information. Purpose violation risk will be:

$$R_{purpose} = \frac{\sum_{k=1}^{q} p_k s_k e_k u_k}{\sum_{k=1}^{q} u_k} \quad (5)$$

There was a significant difference between category behavior. Attributes with the limited number were the ones that were best retained since the pipeline was very conservative with their condition at each transition. The records of customer profiles also proved to be very stable, a few of them, however, needed further consideration when combined with more contextual information. Transaction context exhibited a bit more of a decrease since derived features occasionally confounded event information with other information sources. Preference signals added more ambiguity as a preference can be gathered towards a particular service interaction and not necessarily warrant the same use. Operational logs had the highest decrease in the list of categories not since their policies were eliminated, but since their initial purpose was limited in a number of built cases as compared to the requested model activity.

## 6. DISCUSSION

The key takeaway is that all of the records are not necessarily decided upon or given the same response; it is that all of the representations are still linked to the condition that triggers the use. As can be seen from the first table, the retention is still high for all five categories of data; and from the second table, the same 117 instances are still available to the decision process at each checkpoint. This continuity is important as if records of difficulty are lost in the process of transformation, it can be a superficial compliance and the context that is required for control is lost in the process. The downstream behaviour is posted in the category results, which indicate the impact of the policy strength. There is the smallest decrease (98 to 95) in the number at the model stage (Restricted Attribute) amongst the five categories. This pattern corresponds to "conservative" inheritance: If a restrictive condition is passed, a derived object cannot get a larger scope of permission just because it's got a new schema, or a new representation. The Customer Profile retention score has decreased from 96 to 92, still a very good score, but showing that a few uses are based on the profile, and need to be understood. At the model stage, the score for Transaction Context is 90 and for Preference Signal, 89. These values are not bugs, but useful signals, as, depending on the context and user preferences, very narrow interactions can be collected to provide data for user preference and/or context without necessarily entailing broader model reuse. Operational Log concludes at 88, where it's important to recognize that technically telemetry information can

be considered governed information if it is the result of user or system activity.

This behaviour can be illustrated by the numbers of validations. There are 111 passes in Source Intake, since the initial purpose is known and has been directly linked to each of the records. There are 4 review-needed decisions at intake which are increased to 8 during the model training. There are more blocked decisions (2 to 5) over the same period of time. This shift can be indicative of an increase in governance pressure the further away the downstream activities are from collection. There should be a clear sign of that pressure in a strong pipeline, rather than a concealment. If all of the stages had returned the same number of passes then the result would indicate that the validator was not being influenced by transformation context. The second important property is that the later reduction to 4 blocked cases of the model output also demonstrates persistence in aid of resolution. The information on provenance and purpose is retained so that if there is any error in the data a previous path can be corrected without discarding the whole path.

A scatter plot shows a relationship at the instance level that is similar to that illustrated in the first case. The 39 high constraint points lie in a higher region as a strong constraint upstream is carried with strong constraint downstream. The 39 cases that are moderate are in a middle band, and the 39 cases open-use are in a lower control band. This separation is logical since an open-use record does not warrant the kind of control which should be imposed upon a highly guarded record. It's the positive direction of the entire point set that is important. The propagation scores are correlated with the downstream compliance scores, with higher on the propagation score having a higher downstream compliance score. The distributions within each group illustrate that there is no simple mechanical copying of the label purpose persistence: it interacts with the trajectory of change and the asked action.

The grouped bar chart provides a clearer indication of those elements of the policy package that will be the easiest to maintain. The provenance retention is lowest at the model stage (94), but the series still has a high provenance retention (98 at source). This is a direct implication of the use of a persistent identifier, and explicit source relationship. The allowed-use retention is 97 and 93 respectively, both of which indicate that there is still a high level of retention of permissions, even when activities get more specific later on. The restriction retention decreases from 96 to 91 and thus has the highest reduction. The larger change is appropriate as restrictions are where policy conflicts are manifested in a combination of sources or when a model request exceeds the scope of the model. Important note: the restriction series still stands: >90 at the model level, meaning that there is still information available to help enforce the restriction. Embeddings are to be given special attention. They no longer reveal the familiar fields and this can easily lead teams to believe it as a separate technical object. That's not the assumption made by the architecture. Embedding is bound to the embedding persistence identifier, and gets a policy sidecar which will contain the inherited conditions. The numbers back up this design, as the Embedding Layer still has 106 passes, 7 reviews and 4 blocks. More non-pass decisions are a measure of an assessment of the layer, rather than of a lost policy. Attached governance context is much more important than for source data as it isn't as readable in the representation.

The problem with derived data is that it's, well, derived. A new feature could allow for multiple source fields and the different source fields could have different allowed uses. The method overcomes this situation conservatively by keeping the most stringent condition which is applicable. This is the case in the first table, as seen by the gradual decreases. Transaction Context's drop is from 95 to 93, Preference Signal's from 94 to 92 and Operational Log's from 93 to 91. While these changes mean that there is added review requirement as a result of transformation, it does not negate the governing condition. Thus, the architecture considers derivation a policy event – in addition to being a technical event. The results also impact the management of models. A large number of organizations will look at a set of data prior to training, and then document the resulting model. By adding the purpose persistence, there is a continuous link between those checkpoints. Unlike most stages, if the upstream processing succeeds, then the model stage doesn't automatically get permission. There are 104 passes, 8 reviews and 5 blocks in Model Training. This is a repeatable choice useful for agentic systems to choose tools, get context or create new workflows dynamically. A request can be checked against the same persistent condition, regardless of whether the data traversed multiple services or not, all the way to a model facing request.

# 7. CONCLUSION

This study shows that purpose persistence can be applied on a source data, derived data, embedding and model facing artifacts as a durable control. In the internally built 117-instance evaluation the proposed architecture preserves the identity of the policy, as it is repeatedly transformed, and at the same time is able to discriminate between compatible, uncertain and incompatible uses. The category retention from source to model continues to be high with the highest score being for Restricted Attribute records (95), Customer Profile records (92), Transaction Context (90), Preference Signal (89), and Operational Log (88). Stage Decisions are 100% complete at every stage and the Model Training has 104 passes, 8 cases needing a review and 5 blocks. In Model Output the block count is reduced to 4 due to the fact that retained context enables one to be resolved without weakening the rule.

The results of the graphs support this same conclusion. The scatter points in the 117 range illustrate that the greater the propagation the greater the downstream compliance, for the high, moderate and open-use groups. The grouped bars indicate that provenance is the most stable ending at 94 while allowed-use retention is at 93 and restriction retention is at 91. These values show some erosion and not the disappearance of the values. It's therefore an architectural contribution, as purpose becomes operational metadata, which is carried with each representation and verified at each subsequent transition. This method minimizes the chances of embeddings or features that have been derived from these, being separated from their context and offers a tangible basis for responsible reuse of AI data.

# 8. REFERENCES

[1] T. Akidau et al., "The dataflow model: A practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing," Proc. VLDB Endowment, vol. 8, no. 12, pp. 1792–1803, 2015. https://doi.org/10.14778/2824032.2824076

[2] C. Batini and M. Scannapieco, Data and Information Quality: Dimensions, Principles and Techniques. Cham, Switzerland: Springer, 2016. https://doi.org/10.1007/978-3-319-24106-7

[3] D. Baylor et al., "TFX: A TensorFlow-based production-scale machine learning platform," in Proc. 23rd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), Halifax, NS, Canada, 2017, pp. 1387–1395. https://doi.org/10.1145/3097983.3098021

[4] E. Breck et al., "Data validation for machine learning," in Proc. 2nd Conf. Machine Learning and Systems (SysML), Stanford, CA, USA, 2019. [Online]. Available: https://research.google/pubs/data-validation-for-machine-learning/

[5] E. Caveness et al., "TensorFlow Data Validation: Data analysis and validation in continuous ML pipelines," in Proc. 2020 ACM SIGMOD Int. Conf. Management of Data, Portland, OR, USA, 2020, pp. 2793–2796. https://doi.org/10.1145/3318464.3384707

[6] M. Koehler et al., "Incorporating data context to cost-effectively automate end-to-end data wrangling," IEEE Trans. Big Data, vol. 7, no. 1, pp. 169–186, 2021. https://doi.org/10.1109/TBDATA.2019.2907588

[7] A. Debrouvier et al., "A model and query language for temporal graph databases," VLDB J., vol. 30, no. 5, pp. 825–858, 2021. https://doi.org/10.1007/S00778-021-00675-4

[8] I. F. Ilyas and T. Rekatsinas, "Machine learning and data cleaning: Which serves the other?" ACM J. Data Inf. Quality, vol. 14, no. 3, 2022. https://doi.org/10.1145/3506712

[9] Y. Gong et al., "A survey on dataset quality in machine learning," Inf. Softw. Technol., vol. 162, 2023. https://doi.org/10.1016/j.infsof.2023.107268

[10] DAMA International, "DAMA-DMBOK: Data Management Body of Knowledge," 2024. [Online]. Available: https://www.dama.org/cpages/body-of-knowledge

[11] M. Garouani, F. Ravat, and N. Vallès-Parlangeau, "Model lake: A new alternative for machine learning models management and governance," in Proc. Int. Conf. Web Information Systems Engineering (WISE), vol. 15439, Lecture Notes in Computer Science, pp. 133–144. Springer, 2024. https://doi.org/10.1007/978-981-96-0573-6_10

[12] R. Boselli, S. D'Amico, and N. Nobani, "eXplainable AI for word embeddings: A survey," Cogn. Comput., vol. 17, no. 1, Art. no. 19, 2025. https://doi.org/10.1007/S12559-024-10373-2

[13] S. Sultan and C. D. Jensen, "Ensuring purpose limitation in large-scale infrastructures with provenance-enabled access control," Sensors, vol. 21, no. 9, Art. no. 3041, 2021. https://doi.org/10.3390/s21093041

[14] J. Park and R. Sandhu, "The UCONABC usage control model," ACM Trans. Inf. Syst. Security, vol. 7, no. 1, pp. 128–174, 2004. https://doi.org/10.1145/984334.984339

[15] R. Iannella and S. Villata, "ODRL Information Model 2.2," W3C Recommendation, Feb. 15, 2018. https://www.w3.org/TR/odrl-model/

[16] T. Lebo, S. Sahoo, and D. McGuinness, Eds., "PROV-O: The PROV Ontology," W3C Recommendation, Apr. 30, 2013. https://www.w3.org/TR/prov-o/

[17] C. Jung and J. Dörr, "Data usage control," in Designing Data Spaces, Springer, 2022, pp. 129–146. https://doi.org/10.1007/978-3-030-93975-5_8