Distributed Provenance Chain for a Digital Pathology Use Case

Authors: Rudolf Wittner, Cecilia Mascia, Francesca Frexia, Markus Plass, Matej Gallo, Heimo Müller, Jörg Geiger, Petr Holub

This document demonstrates usage of the provenance backbone concept to document a complex real world use case. The main purpose of the provenance backbone is to enable creation of distributed provenance chains in a heterogeneous environment. In particular, the backbone consists harmonized derivation paths between inputs and outputs of a documented process in provenance, and enables traversal of the distributed provenance chain using a common algorithm, independently of the particular process being documented.

In the following sections, each step of the use case is described, and application of the underlying provenance model is shown and explained. The use case can be divided into two parts: 1) clinical part dealing mainly with biological material, diagnostics and data generation; 2) computational part dealing with processing of the data that are generated from the biological material.

The provenance graphs shown in the figures adopt the convention defined by the W3C PROV standard. For clarity, the figures in are simplified. More specifically, the figures in this document do not contain prefixes of identifiers and destination bundle identifier and service URI attributes of connectors. Corresponding figures including the technical details are uploaded to the repository[1] together with this document, and are referenced from captions using corresponding file name in parentheses[2]. The detailed information is also included in generated provenance that is part of the implementation part. Since expression of domain specific semantics is not in scope of this work, we implemented domain specific provenance generation only for the computational steps of the use case.

1 General description of the use case

The example considers a use case from the Digital pathology domain specialised in the detection of prostate cancer[3]. Its goal is to train AI model to detect presence of carcinogenic cells in Whole Slide Images of human prostate, which have been acquired in clinical environment.

The whole process consists of several steps, each of which is carried out by different organizations, or different organizational units of the same organization. General schema of resulting provenance information documenting the process is depicted in Figure 1.

1. **Biological material acquisition** is done in a clinical environment. Primary samples are taken as part of a medical treatment – prostate biopsy – and sent to a pathological department for examination.

2. **Biological material processing, WSI generation and examination** are done as part of the diagnostic process. This consists of the gross evaluation, generation of tissue blocks, cutting of tissue block into slices to be placed on glass slides, staining and scanning. Resulting scans (whole slide images (WSIs) of the slides) are consecutively examined and annotated by a pathologist. The annotations depict tumor areas and other morphological features. The annotated scans, resulting diagnosis and the slides are then sent to a biobank to be stored. The annotated scans are provided to an AI based computational workflow.

[1] `https://github.com/RationAI/crc_ml-provenance/tree/new-auto-provenance`

[2] The figures containing the technical details were implemented using the PROV library, and could be easily translated to PROV-O, PROV-XML or PROV-JSON serializations.

[3] Digital pathology is a research field applying achievements in imaging technologies to develop systems capable of diagnosing or supporting diagnosis of patients based on their clinical data and large scans of histopathological biological material (so called Whole Slide Images, WSIs), also enabling the application of AI methodologies to improve the evaluation results.

3. **Biological material storage**. The samples are stored in a controlled environment and provided through a searchable database for future use. Thereby a specific cohort or a set of samples can be collected to answer designated research questions.
4. **WSI preprocessing** prepares the annotated scans to be processed by an AI model. Since the AI model is not capable of processing high resolution images, the images are split into smaller pieces – patches – and corresponding annotations transformed into appropriate format. The resulting dataset is divided into two parts - training and testing sets.
5. **AI model training**. Training part of the AI workflow consists of consuming the annotated images (training set) and training an AI model to identify tumor on the WSIs. Result of this step is a trained AI model and a summary file containing the AI model predictions about tumor presence.
6. **AI model testing** The trained AI model is used to identify carcinogenic cells at whole slide images, present in the testing set (WSIs in the testing set has not been used during training). The result is compared with pathologist's annotations. In this step, it is possible to estimate the quality of the predictions made by the pipeline either to modify it to get better results, or to declare that is satisfies given requirements and can therefore potentially be used in clinical use or research.

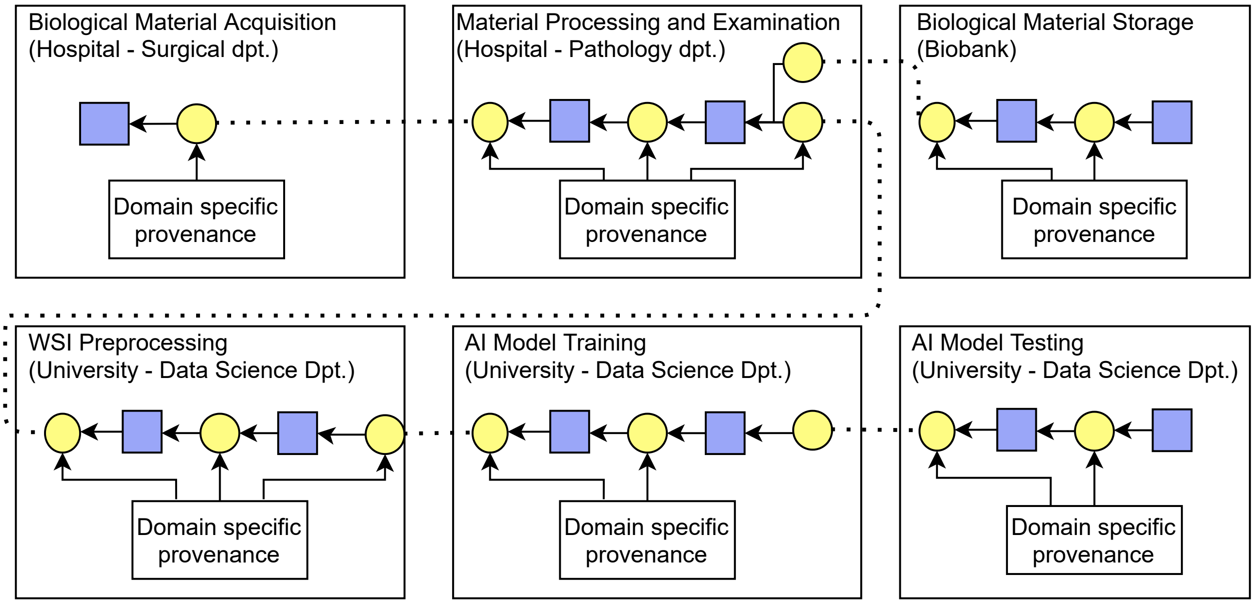

Biological Material Acquisition
(Hospital - Surgical dpt.)
Domain specific provenance
Material Processing and Examination
(Hospital - Pathology dpt.)
Domain specific provenance
Biological Material Storage
(Biobank)
Domain specific provenance
WSI Preprocessing
(University - Data Science Dpt.)
Domain specific provenance
AI Model Training
(University - Data Science Dpt.)
Domain specific provenance
AI Model Testing
(University - Data Science Dpt.)
Domain specific provenance

Figure 1: Simplified schema of the distributed provenance chain for the digital pathology research pipeline.

2 Biological material handling and digitization

2.1 Biological material acquisition

Beginning of the pipeline starts with a patient in a surgical department of a hospital from whom a biological sample is extracted. Patients are associated with an identifier, which is unique across given country (in countries like Czech Republic or nordic countries). A bioptic request is generated for the patient, which serves as a prescription for the acquisition of biological material from the patient and its consecutive examination. Result of the acquisition process is biological material which is sent to a pathology department of the same institute.

Provenance bundle depicted in Figure 2 represents the beginning of a distributed provenance chain. The provenance backbone does not contain receiver connector, receipt activity nor an external input, since there is no previous step in this example, which would be documented by finalized provenance information. For

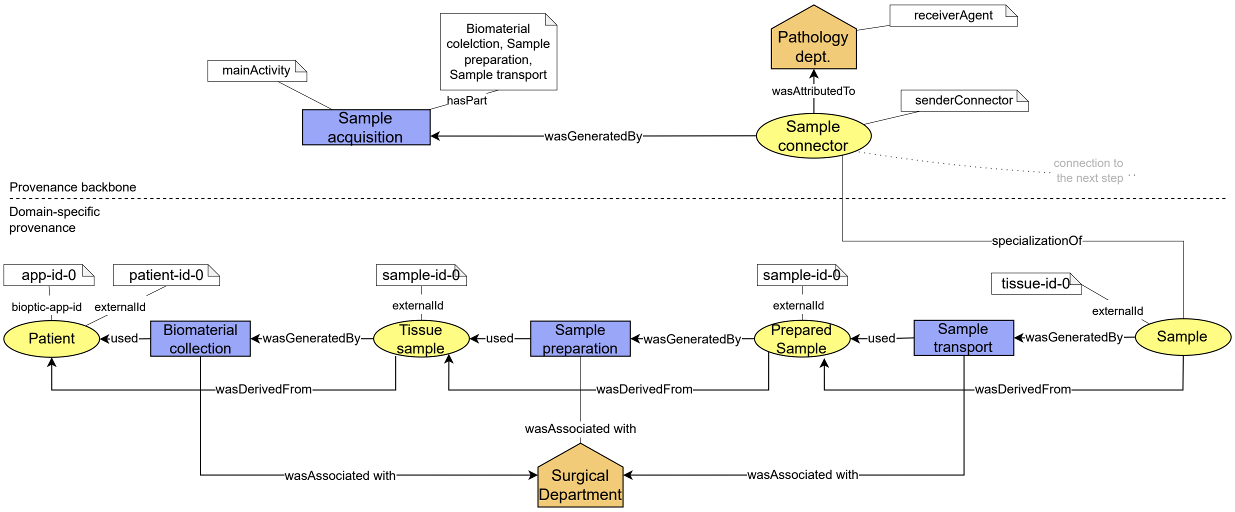


Figure 2: Finalized provenance information documenting sample acquisition (DP1 - Samples acquisition.png).

that reason, start of the chain is expressed on the backbone as a main activity, which generates the sender connector, representing the acquired sample.

If there would be a previous step documenting the sample source – e.g., finalized provenance information documenting patient related clinical data using HL7 FHIR[4] – this would be linked using the receiver connector, receipt activity and external input provenance structures as defined in the proposed provenance model.

The sender connector entity is present on the provenance backbone, despite finalized provenance information documenting the consecutive step does not necessarily exists during the finalization event for the current step. Semantics of presence of the sender connector is that it explicitly says that there was an output of the documented process, which was provided to an external consecutive process – this says nothing about whether finalization event for the consecutive step has already happened, so that it says nothing about whether provenance information about the next step exists, or whether the next step was already performed. In such a case, the attributes of the sender connector (destination bundle identifier and provenance service URI) used for the forward navigation would stay empty. In case when the finalization event for the consecutive step is executed later, the technical attributes of the connector to locate the consecutive provenance bundle can be added using the proposed versioning mechanism. This is the same for the all parts of the chain. Further description of relation between connector and finalization event is present in Section 5.

2.2 Biological material processing, WSI generation and examination

Based on the same bioptic request, the biological material goes to a pathology department, to perform the biological material preparation, scanning and examination. This consists of the following steps:

1. Macroscopy: pathologist decides which parts of the material will be further processed. These parts are split, fixed in formalin, and embedded into paraffin. Results are blocks with fixed samples (small tissue parts).

2. Slicing: the blocks are cut into slices, which are put onto a glass slides.

3. Staining: the slides are stained according to a standardized staining protocol. Different slides can be stained according to different protocols (depending on purpose), possibly at different places, these can be later collected together to continue the process

4. Digitization process/scanning: using a slide scanner, whole slide images of the slides are generated.

[4] https://www.hl7.org/fhir/provenance.html

5. Digital pathology examination: pathologist takes WSIs and examines them on his/her computer, trying to identify cancer structures, producing annotations of those structures.

Physical slides along the WSIs and pathological diagnosis are then sent to and stored in the biobank. WSIs with annotation are sent to for integration into dataset used in an AI pipeline.

Finalized provenance information for this step, depicted in Figure 3, contains single receiver connector and derived external input entity. These two entities represent the sample received from the previous step. The sample is then used by the main activity (physical slide preparation), which generates three different types of outputs – pathological diagnoses, resulting slides collection and WSI data (scans and annotations). All of these are sent to a biobank for storage in the consecutive step. A copy of the WSI data is sent to a high-performance computing center to train the AI models, which is expressed on the backbone as a standalone entity.

2.3 Biological material storage

The slides, pathological diagnosis and WSI annotations from the digital pathology step are received and stored in biobank, and can be later requested for further use. For the purpose of the example, processes related to biological material handling and processing ends here. Finalized provenance information for the samples storage step is depicted in Figure 4.

The provenance backbone in the finalized provenance information contains three receiver connectors, each of them corresponding to the outputs of the previous process. The pathological diagnosis and WSI data are merged into a single entity during the receipt activity, since there was no reason to express those as standalone entities[5].

The main activity – storage process – does not generate any outputs on the provenance backbone. This is because the outputs of this process (stored samples) have not been sent to any consecutive process yet.

The reason for generating finalized provenance example, despite it has no traceable outputs, is that the stored samples might be requested after long time (generally in years) and there is a need to finalize (and consequently digitally sign and archive) particular provenance information without further waiting (not signing provenance information would leave a space for potential malicious tampering with generated provenance). If the biological material would be requested for further use later, the generated provenance bundle could be updated using the versioning mechanism (described in Section 5).

[5] the reason for splitting the entities in the previous step was that the WSI data are sent to AI pipeline **without** the pathological diagnosis

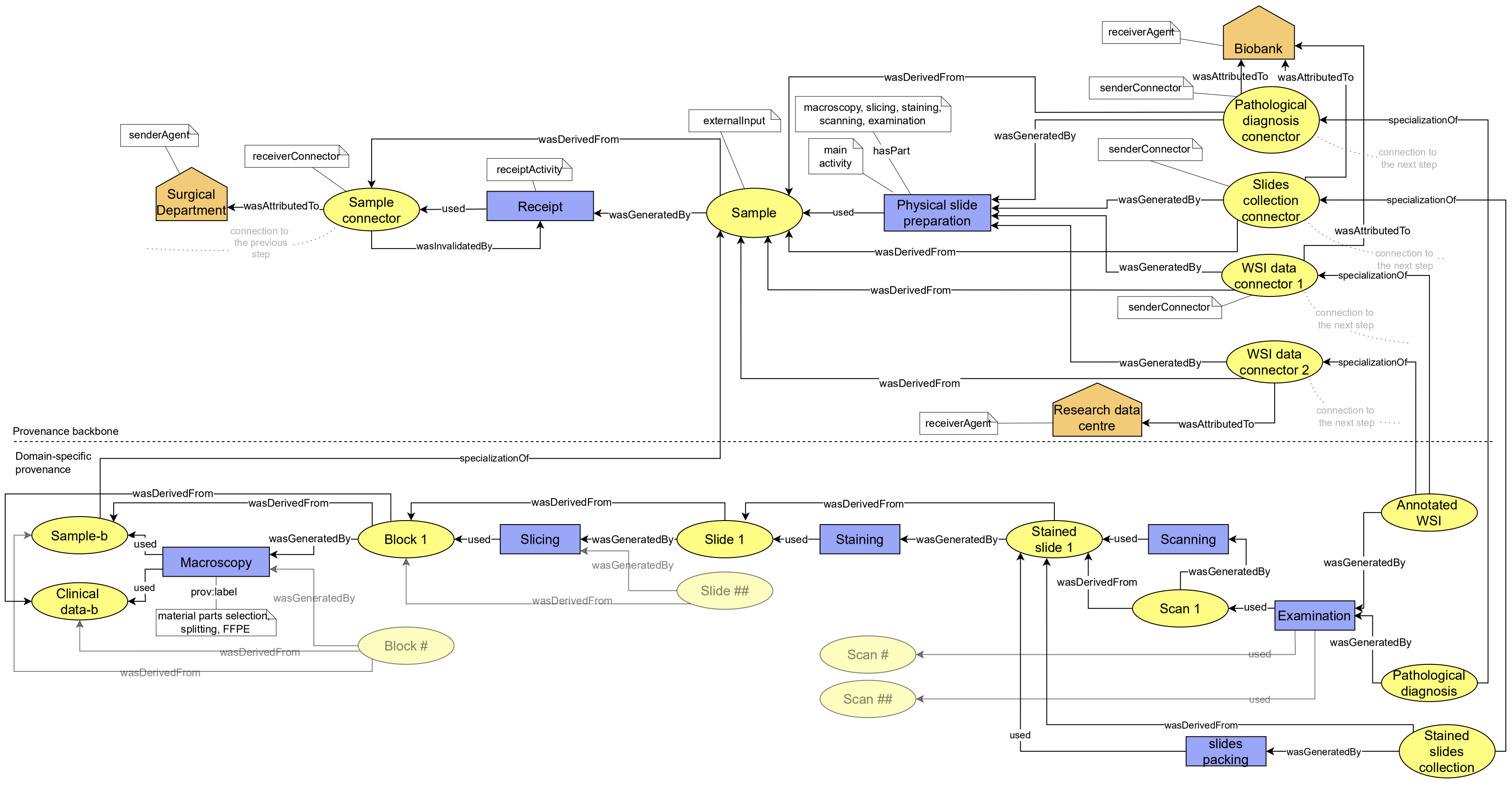


Figure 3: Finalized provenance information documenting sample processing, whole slide images generation and their examination (DP2 - Sample processing and scanning.png).

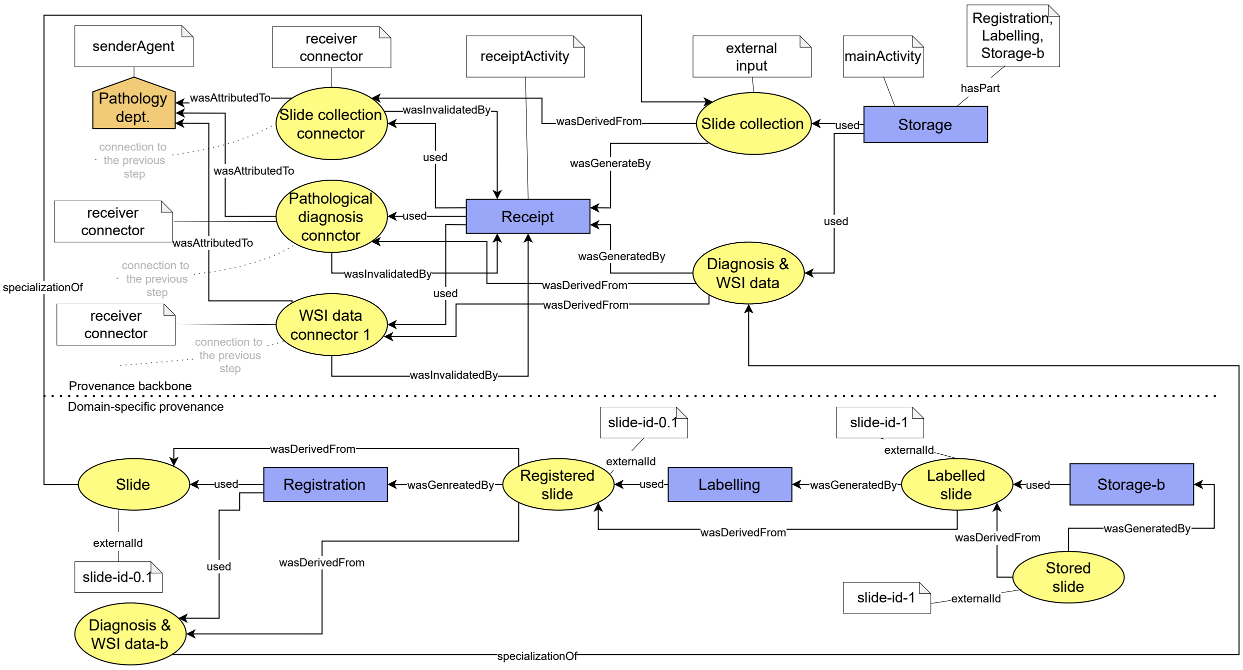


Figure 4: Finalized provenance information documenting sample storage (DP3 - Sample storage and biobanking.png).

3 Data processing and machine learning pipeline

The machine learning pipeline (MLP) processes annotated images of slides and learns to detect the presence of carcinoma. The pipeline is implemented as a set of python scripts and is executed on a server at a high-performance computing center. The input dataset is stored on a filesystem, where it is accessed by the scripts.

3.1 WSI dataset preprocessing

In this step of the research pipeline, the WSIs coming from the previous step are collected and preprocessed. Purpose of the preprocessing step is to split the WSIs into smaller regions – patches – to make them appropriate for the consecutive AI model training. The patches are also divided into two datasets: training and testing. Patches, which form WSIs in the testing dataset can never be included in the validation or training dataset, since this would disrupt the whole AI model training process.

Finalized provenance information documenting the WSIs preprocessing is depicted in Figure 5

WSIs are expressed on provenance backbone as an external input received by a receipt activity and derived from a connector, linking the current bundle to the previous step. External input is then used by the main activity, which generates two entities representing training and testing sets represented as connectors.

3.2 AI Model Training

In this step of the research pipeline, the train set from the previous step is used to train the AI model. The set consists of two subsets - training and validation subset. Patches sets forming validation and training sets must be strictly disjunct, but can originate from the same WSI. The training process consists of two parts: training of the model itself and validation. The training part is performed in iterations using the training set - in each training iterations, the model uses more patches to learn, so that is expected to provide more precise estimations about presence of carcinogenic cells in WSI patches. The validation part evaluates the trained model after each training iteration using the validation dataset, to get continuous results about the model performance. Both validation and training is performed on patches level (not on WSI level). Number

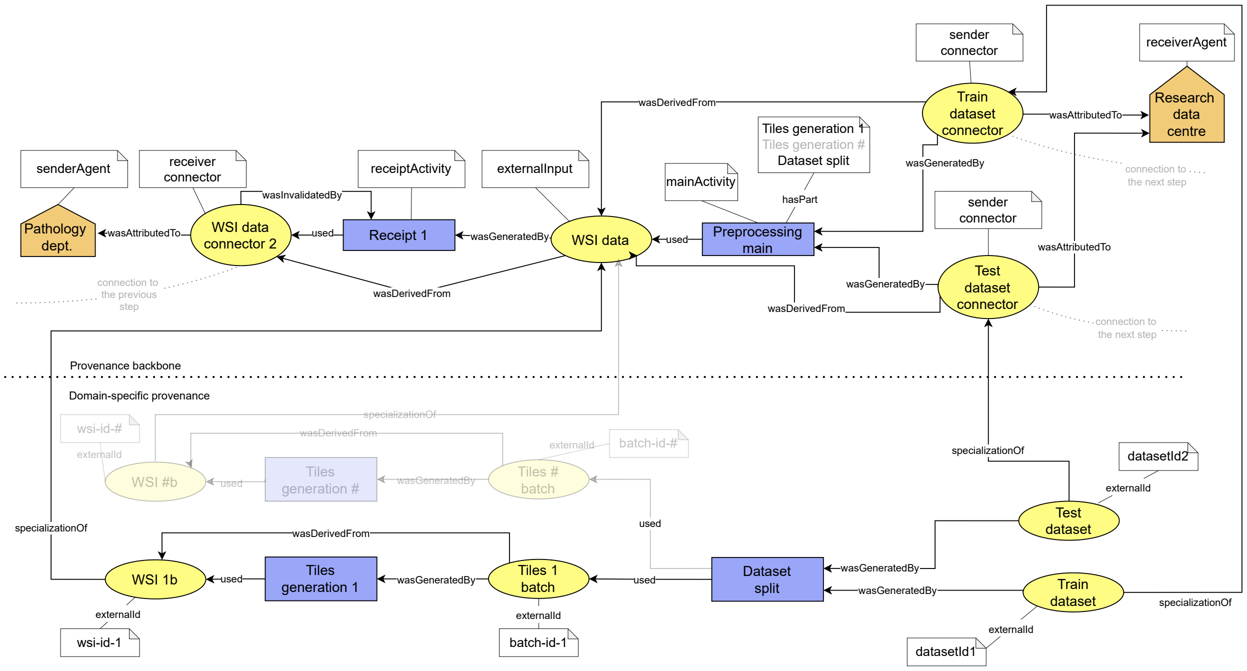


Figure 5: Finalized provenance information documenting WSI data preprocessing (DP4 - WSI dataset preprocessing.png).

of iterations is determined by the training configuration. As a result, the model with best results (based on validations) is selected. This is depicted in Figure 6.

Input dataset is coming from the previous step, which is expressed using a sender agent responsible for the receiver connector. The trained model is expressed as a trained model connector entity, since it serves as an input for the consecutive step – AI model testing.

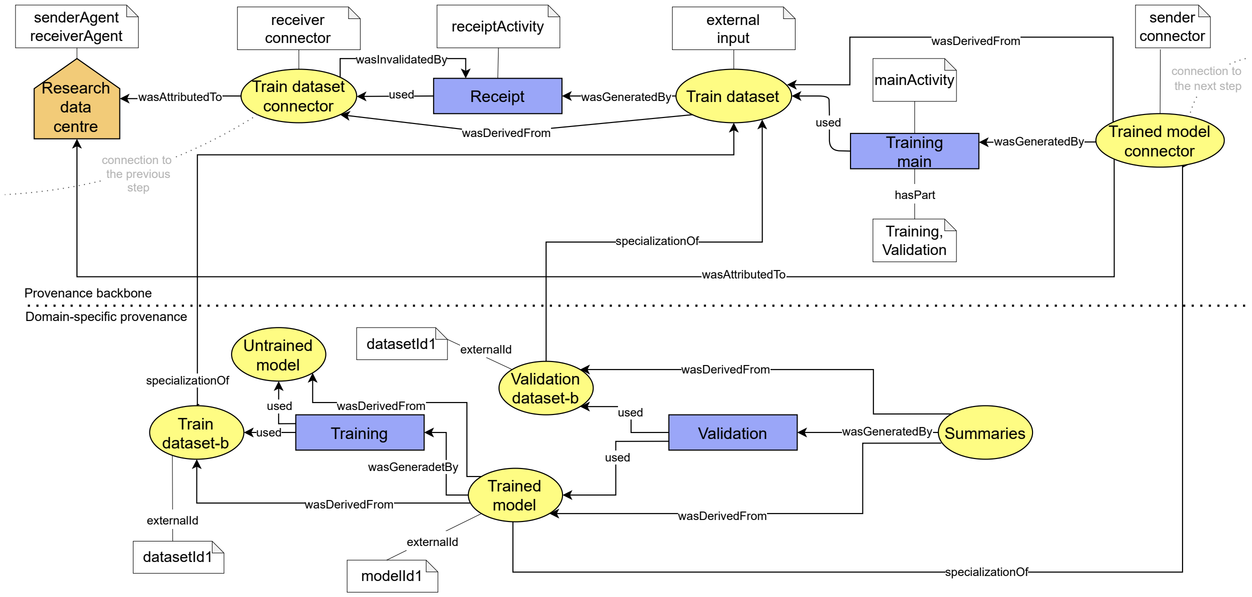


Figure 6: Finalized provenance information documenting AI model training (DP5 - Model training.png).

3.3 AI Model Testing

Goal of the AI model testing step is to estimate how good the model is with predictions of tumor presence in given WSIs. In contrast with previous step, where the validation steps were evaluated according to patches annotations, the testing part is evaluated according to the whole slide images. This is done by running the the model on the testing dataset, and its results are then compared with the original annotations from the pathologists. In this part of the research pipeline, the AI model does not have the original annotations as part of the input - is it the task of the trained model to determine whether and where the carcinogenic parts of slides are present. Result of this step is a statistics providing qualitative assessment of the trained model results.

The resulting finalized provenance information is depicted in Figure 7.

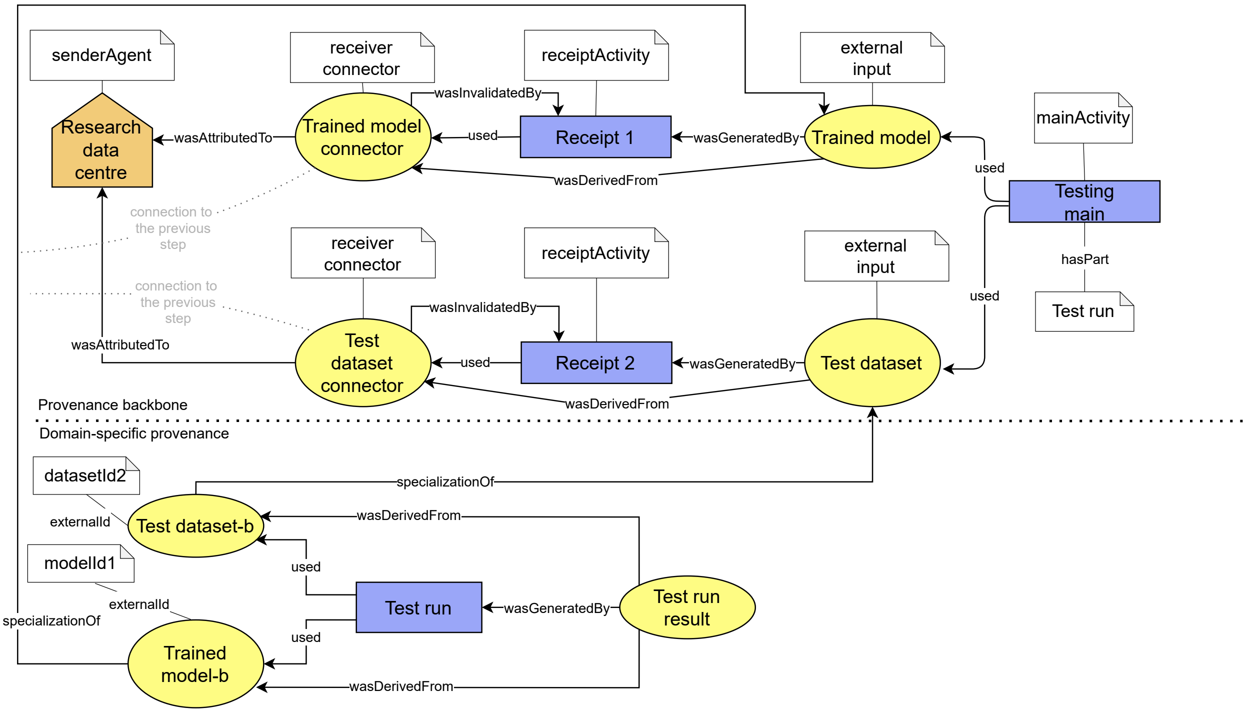


Figure 7: Finalized provenance information documenting AI model testing (DP6 - Model testing.png).

This is the last step of the pipeline described in this example. As a result, the main activity does not generate any outputs – sender connectors – which would enable navigation to finalized provenance information documenting a consecutive step (simply because there is no consecutive step). For that reason, every outputs of this process are expressed only in a domain specific part of finalized provenance information.

4 Missing provenance components and jump connectors

The basic structure of a distributed provenance chain was shown (Figure 1) in the previous sections. In this section, we will use the existing chain and the example to demonstrate usage of jump connectors — a method to address missing provenance components. The methods for missing provenance components can be used when a provenance part in a chain cease to exists (e.g., because particular responsible organization cease to exists), when part of a chain is not generated at all, or to simplify traversal of the chain in cases, when some steps of the are not in primary interest.

Since attachment of domain specific provenance is not affected by including jump connectors, we omit domain specific provenance from figures for simplicity, and depict relevant parts of provenance backbone only.

4.1 Adding jump backward connectors

The first bundles in the provenance chain containing a jump backward connector are documenting biological material storage and WSI data preprocessing steps. This is because earlier steps in the chain has no preceding steps, to which they could refer using the jump backward connectors. Inputs for both the storage and preprocessing processes are slides and WSI images, whose origin reach to a biological sample acquired at a hospital. This is expressed in provenance information as a jump backward connector. The connector is included in both the storage and preprocessing provenance (with distinct identifiers), from which particular external input entities were derived. Attributes of the both jump connectors refer to the sender connector in the biological material acquisition bundle, from which the WSI data and slides were later obtained. The resulting provenance is depicted in figures Figure 8 and Figure 9.

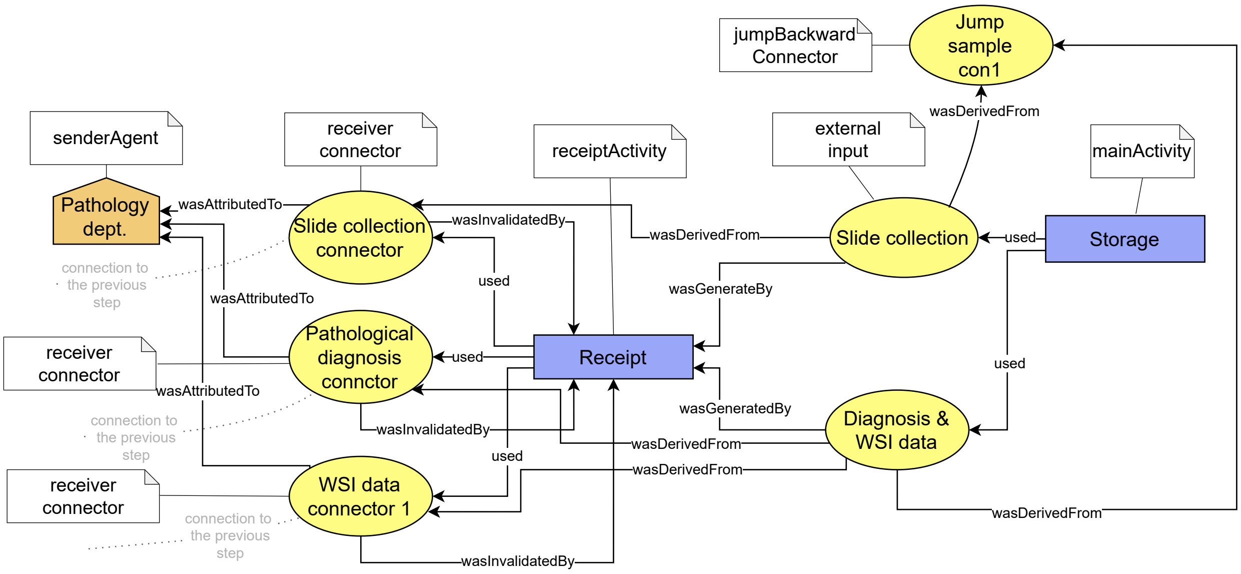


Figure 8: Finalized provenance information documenting samples storage with single jump backward connector included (DP3 - Sample storage and biobanking - BackConnectors.png).

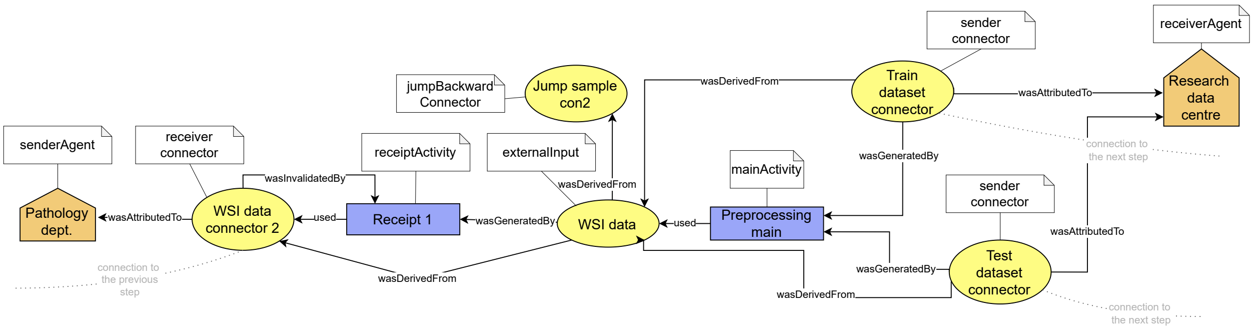


Figure 9: Finalized provenance information documenting WSI data preprocessing with single jump backward connector included (DP4 - WSI preprocessing - BackConnectors.png).

Jump backward connectors are added to consecutive parts of the chain in an analogous way. Provenance bundle documenting the AI model training step uses single input expressed as connector, which is the trained dataset coming from the data preprocessing step, but which was previously affected by sample acquisition and data generation steps. This is expressed in the AI training bundle as two distinct jump connectors - one referring to acquired sample in biological material acquisition step, and the second referring to the raw WSI data generated in the data generation step. This is depicted in Figure 10.

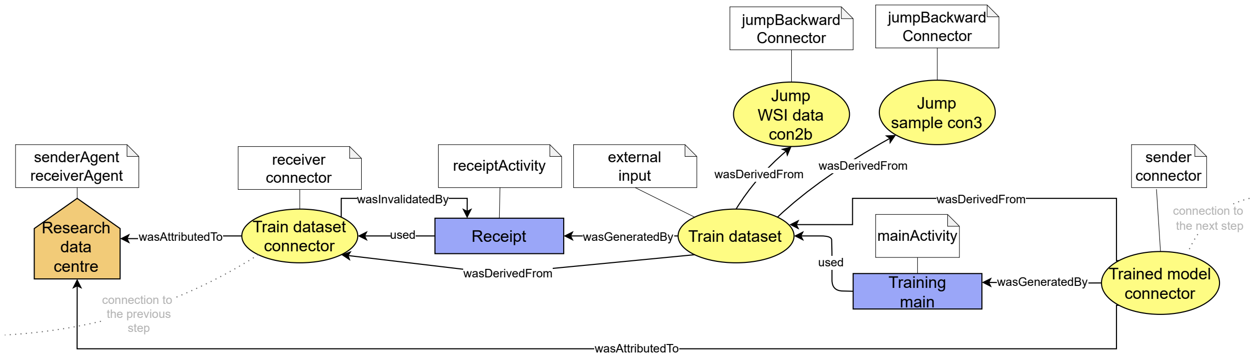


Figure 10: Provenance backbone for AI model training including two jump backward connectors (DP5 - Model training - BackConnectors.png).

The last bundle in the provenance chain containing a jump backward connector is documenting the AI model testing process. This bundle contains two external inputs: the trained model coming from the AI model training step, and a test dataset coming from the data preprocessing step. Both of the inputs have the same predcestors, so they are derived from the same jump backward connectors: one representing acquired sample in the biological material step, and second representing WSI data generated in the data generation step. In addition, external input representing the trained model is derived from the jumpBackward connector representing the trained dataset, which was generated in the data preprocessing step. The resulting bakcbone is depicted in Figure 11.

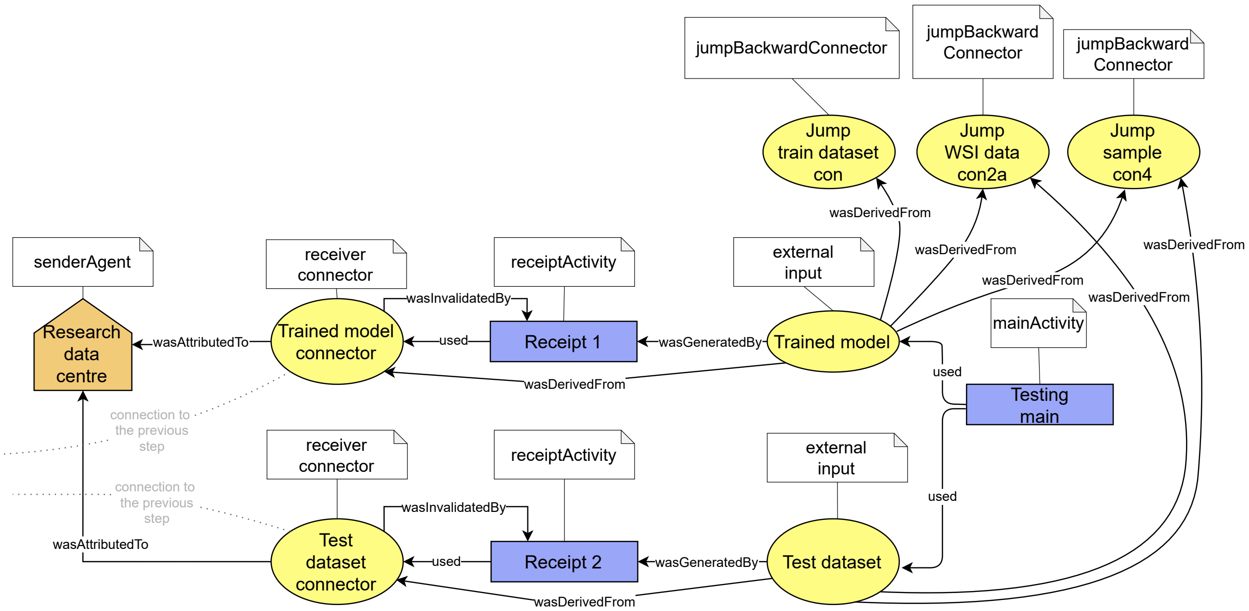


Figure 11: Provenance backbone for AI model testing including three jump backward connectors (DP6 - Model testing - BackConnectors.png).

4.2 Adding jump forward connectors

Jump connectors are entities with shared identifiers. For that reason, similarly to sender and receiver connectors, jump backward connectors should have corresponding jump forward connectors in the referenced bundles. Jump forward connectors are included in four bundles in our example. Biological material acquisition bundle contains four jump forward connectors, each referring to a bundle, where derivates of acquired biological material are used. In particular, the bundle contains a jump connector to refer to biological material storage step, WSI data preprocessing step, AI model training step and AI model testing step. Analogously, the material processing and data generation bundle contain jump forward connectors to refer to AI model training and testing steps. The WSI data preprocessing bundle contains a jump forward connector to refer to AI model testing step. The resulting bundles with jump forward connectors included are depicted in Figure 12, Figure 13 and Figure 14.

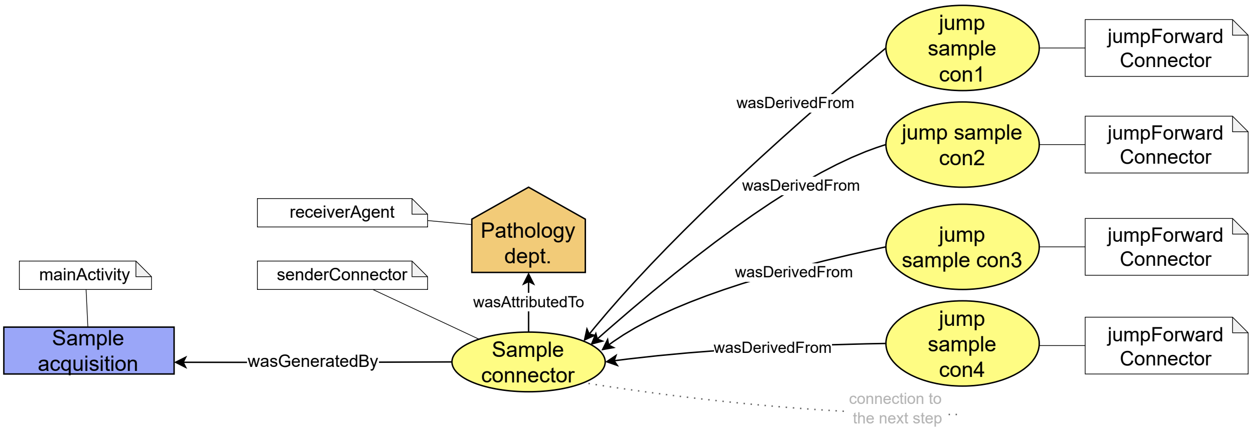


Figure 12: Provenance backbone for biological sample acquisition including four jump forward connectors (DP1 - Samples acquisition - ForwardConnectors.png).

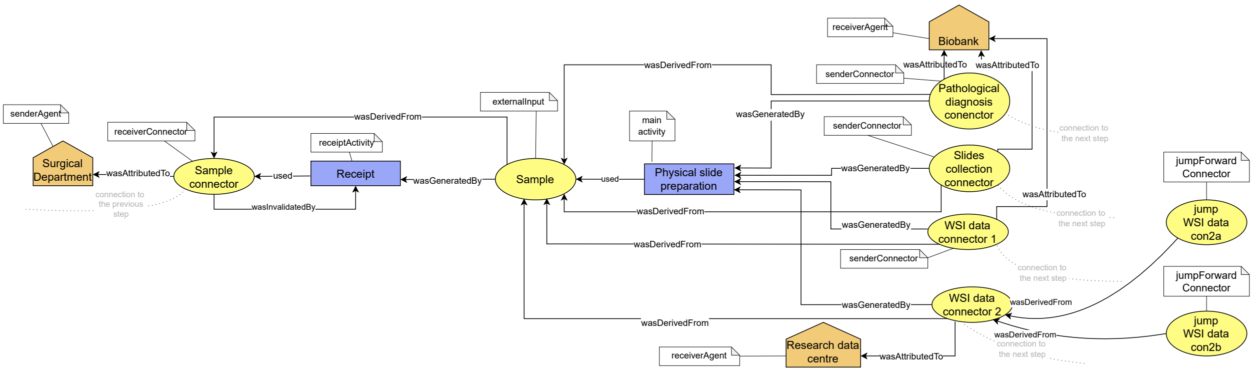


Figure 13: Provenance backbone for samples processing and data generation step including two jump forward connectors (DP2 - Sample processing and scanning - ForwardConnectors.png).

5 Provenance versioning

In this section, we show usage of the proposed provenance versioning mechanism with respect to the provenance finalization event. Provenance finalization event is a time instance, when available information about executed processes is transformed into *finalized provenance information*, which is expressed in terms of PROV and in accordance to the proposed provenance model.

For the purpose of demonstration, consider the first bundle presented in this supplementary material documenting biological material acquisition process, which was depicted in Figure 2 (the same mechanism applies for all provenance bundles in the chain). The exact content of provenance backbone in this bundle – especially values of attributes of the sender connector entity – is directly dependent on when the finalization event for this bundle occurs. In particular, if the finalization event occurs before generation of identifier of provenance bundle documenting the consecutive step – biological material processing and WSI generation

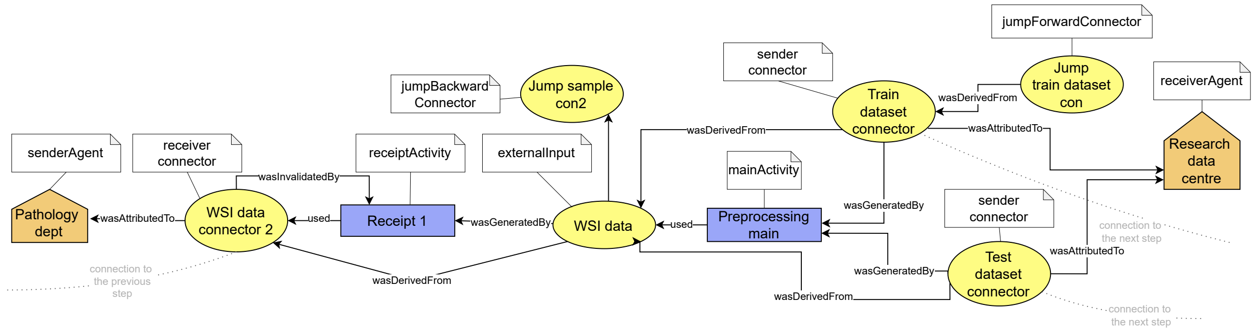
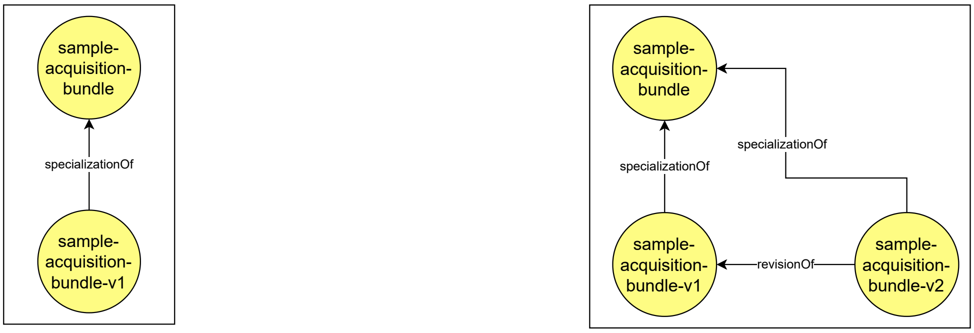


Figure 14: Provenance backbone for WSI data preprocessing step including both jump forward connectors and jump backward connector (DP4 - WSI preprocessing - BackNForwardConnectors.png).

(step 2) – then the sender connector entity can not contain the identifier (simply because it has not been generated yet). For that reason, if finalization event for biological material acquisition provenance occurs before, then the bundle documenting the biological material acquisition process should be later updated in order to be complete – to include the consecutive bundle identifier.

According to the proposed model, this can be accomplished by a meta-bundle. The meta bundle contains meta-provenance, which is provenance information about provenance information[6]. During the finalization event, an entity representing the finalized bundle is generated and included in the meta-bundle according to the revision pattern[7], which is depicted in Figure 15a.

In order to update the content of the finalized bundle – e.g., to include identifier of a consecutive bundle in the attributes of connector – new copy of the bundle is created. The copy is perceived as a replacement of the original bundle and includes any modification of the original bundle - additional, removed or updated provenance information. The original bundle is not deleted! The new bundle can be then seen as a new version of the original bundle, which is expressed in the meta-bundle by applying the revision pattern. Result is depicted in Figure 15b.


sample-acquisition-bundle
specializationOf
sample-acquisition-bundle-v1
sample-acquisition-bundle
specializationOf
specializationOf
sample-acquisition-bundle-v1
revisionOf
sample-acquisition-bundle-v2

(a) Meta-bundle documenting single version of a finalized provenance bundle.

(b) Meta-bundle documenting two versions of a finalized provenance bundle

Figure 15: Provenance versioning – meta-bundle example

The proposed mechanism preserves all earlier versions of any bundle, so it does not disrupt integrity of the whole chain, and enables to add or correct an information in finalized provenance. During resolution of particular budnle URI, the resolving mechanism can always check for the presents of updates in the meta-bundle.

[6] Provenance of provenance concept: `https://www.w3.org/TR/prov-links/`
[7] DOI: 10.2200/S00528ED1V01Y201308WBE007