ITUPublications
Recommendations

International Telecommunication Union
Standardization Sector

Recommendation

ITU-T Y.3602 (09/2022)

SERIES Y: Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities

Big Data

Big data – Functional requirements for data provenance

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Computing power networks	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	Y.3000–Y.3499
CLOUD COMPUTING	Y.3500–Y.3599
BIG DATA	**Y.3600–Y.3799**
QUANTUM KEY DISTRIBUTION NETWORKS	Y.3800–Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	Y.4400–Y.4549
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.3602

Big data – Functional requirements for data provenance

Summary

Recommendation ITU-T Y.3602 describes a model and operations for big data provenance. This Recommendation also provides the functional requirements for a big data service provider (BDSP) to manage big data provenance. The reliability of data is an important factor in determining the reliability of the analysis result. Data provenance aims to ensure the reliability of data by providing transparency of the historical path of the data. In a big data environment, complex data processing and migration due to the big data lifecycle and data distribution cause various difficulties in managing data provenance.

History

Edition	Recommendation	Approval	Study Group	Unique ID[*]
1.0	ITU-T Y.3602	2018-12-14	13	11.1002/1000/13817
2.0	ITU-T Y.3602	2022-09-29	13	11.1002/1000/15074

Keywords

Big data, data provenance, provenance model, provenance operation, provenance requirements, use case.

[*] To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T Y.3602

Big data – Functional requirements for data provenance

1 Scope

This Recommendation specifies the functional requirements for data provenance in a big data ecosystem as defined in [ITU-T Y.3600]. This Recommendation introduces data provenance as well as data provenance in a big data ecosystem, and provides a conceptual model, operations, logical components, and functional requirements for big data provenance. The functional requirements provided in this Recommendation are derived from use cases.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.3600]	Recommendation ITU-T Y.3600 (2015), *Big data – Cloud computing based requirements and capabilities*.
[ITU-T Y.3603]	Recommendation ITU-T Y.3603 (2019), *Big data – Requirements and conceptual model of metadata for data catalogue*.
[ITU-T X.680]	Recommendation ITU-T X.680 (2021), *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 big data [ITU-T Y.3600]: A paradigm for enabling the collection, storage, management, analysis and visualization, potentially under real-time constraints, of extensive datasets with heterogeneous characteristics.

NOTE – Examples of datasets characteristics include high-volume, high-velocity, high-variety, etc.

3.1.2 provenance [b-ITU-T X.1255]: Information pertaining to any source of information including the party or parties involved in generating it, introducing it and/or vouching for it.

3.2 Terms defined in this Recommendation

This Recommendation defines the following term:

3.2.1 big data provenance: Information that records the historical path of data according to the data lifecycle operations in a big data ecosystem.

NOTE 1 – Data lifecycle operations include data generation, transmission, storage, use, and deletion.

NOTE 2 – Data provenance information provides details about the source of data, such as responsible party for the provision of data, functions applied to data, and information about the computing environment for data processing (e.g., operating system, description of the hardware, locale settings and time zone).

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

BD	Big Data
BDC	Big Data service Customer
BDSP	Big Data Service Provider
DB	Data Broker
DP	Data Provider
DS	Data Supplier
H/W	Hardware
OS	Operating System
PI	Provenance Information
PII	Personally Identifiable Information
URI	Uniform Resource Identifier

5 Conventions

In this Recommendation:

The keywords "**is required to**" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this document is to be claimed.

The keywords "**is recommended**" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

The keywords "**can optionally**" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

In the body of this document and its annexes, the words shall, shall not, should, and may sometimes appear, in which case they are to be interpreted, respectively, as is required to, is prohibited from, is recommended, and can optionally. The appearance of such phrases or keywords in an appendix or in material explicitly marked as informative are, to be interpreted as having no normative intent.

6 Introduction to data provenance

6.1 General concept of data provenance

The reliability of data used is an important factor to determine the trustworthiness of a data analysis outcome. Indeed, data can be manipulated and transformed according to the intent of the analyst and distorted in order to extract the desired result. In this sense, data provenance aims to ensure the reliability of data and analysis results by providing transparency of the historical path of data.

Provenance is information pertaining to a responsible party for the provision of data , functions applied to data, and information about the computing environment for data processing (e.g., operating system, description of the hardware, locale settings and time zone) when the source data is changed. From a data management point of view, data continues to change until it is used, and data provenance keeps the history of these changes within:

– data product;

NOTE 1 – A data product is the output data production for distribution (open or sell) purposes.

- a process that enables the creation of data;

NOTE 2 – A process is described by the applied functions on the data source, intermediate outputs and their order.

- a metadata recording process of workflow, annotations, notes about processes; and,
- information that helps determine the derivation history of a data product, starting from its sources.

Data provenance is useful for:

- managing derivation history of a data product starting from its sources;
- ascertaining the quality of data based on ancestral data and derivation;
- tracking back sources of errors;
- allowing automated re-enactment of derivations to update data;
- providing attribution of data sources.

The concept for data provenance in this Recommendation does not include the identification of data subjects or the collection of information to identify them to ensure the use of this Recommendation complies with national legal and regulatory requirements.

Provenance information (PI) is composed of a set of data flows, and each flow contains information of processes (*f*), data sources (*d*) and responsible parties (*p*). In this sense, *PI* is notated as:

$$PI = \{(f, p), (d, p)\}$$

A data flow is divided into a directly associated flow and subordinately associated flow. For example, in Figure 6-1, the PI about *Data d* is composed by a set of:

- directly associated flow: *PI* = {(*f2, pC*), (*Data c, pC*)};
- subordinately associated flow: *PI* (*Data c*) = {(*f1, pC*), ((*Data a, pA*), (*Data b, pB*))}

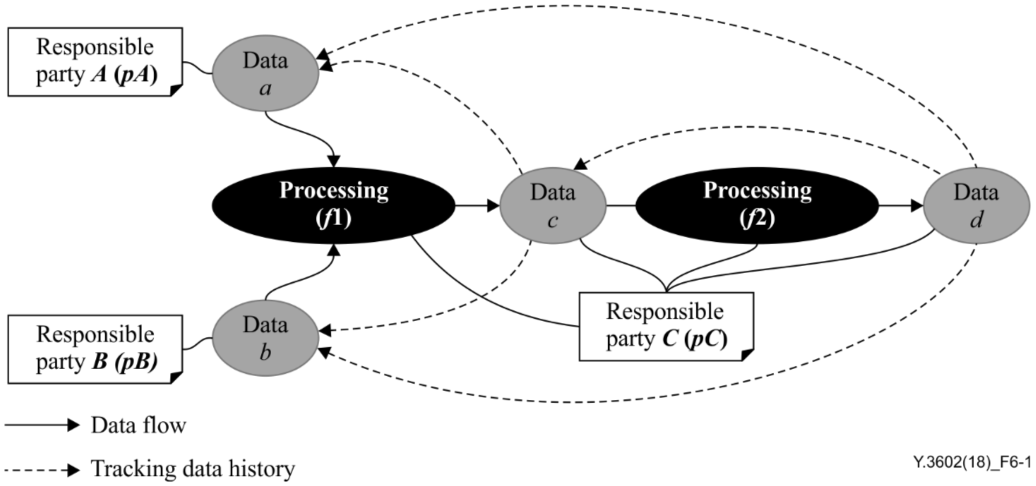


Figure 6-1 – An example of data PI

6.2 Data provenance in a big data environment

In a big data environment, complex data processing and migration due to the big data lifecycle operations (e.g., data generation, transmission, storage, use, and deletion) and data distribution cause various difficulties in managing data provenance. According to the big data ecosystem described in [ITU-T Y.3600], big data provenance needs to treat:

- huge volumes of non-structured, semi-structured, and structured data;
- processing functions description for various types and formats of data;

– data history across multi-application domains.

NOTE 1 – An application domain is an area of knowledge or activity applied for one specific economic, commercial, social or administrative scope [b-ITU-T Y.4100].

NOTE 2 – Transport application domain, health application domain, and government application domain are examples of application domains.

In addition, the big data environment causes several computing challenges for data provenance such as:

– **an efficient storing mechanism for provenance data**: The size of provenance data can be larger than the original data, which may cause storage overhead;

– **minimize provenance collection overhead**: In a distributed system environment, recording provenance and provenance computation cost are important aspects to be considered together;

– **execution's reproduction from PI**: For some big data applications, the environment information (e.g., hardware (H/W) information and parameter configuration of big data engines) is an important factor to be taken into account when aiming to reproduce the execution from data provenance.

The application area of big data provenance and its benefits are:

– **collaborative big data analysis**: Big data provenance allows collaborative big data analysis related to multiple and different domains or applications using data sources information and their process steps;

– **reuse of big data analysis process**: Generally, a big data analysis has complex process steps. Thus, a well-defined analysis model which can be derived from provenance information is helpful to be reused in other cases of big data processing;

NOTE 3 – In a data processing system, the data processing means a course of events occurring according to an intended purpose of effect.

– **automating big data analysis process**: Data provenance gives a context regarding the use of data and allows automated validation and revision of derived data when the base data is updated;

– **audit and protect intellectual property**: Data provenance gives a lineage of data and allows auditing and tracing of digital rights on mash-up data.

7 Overview of big data provenance

This clause presents an overview of big data provenance. This clause describes data provenance in a big data ecosystem, a conceptual model, provenance operations, and logical components for big data provenance.

7.1 Big data ecosystem and data provenance

According to [ITU-T Y.3600], a big data service provider (BDSP) supports data provenance as a part of data management by managing information about the origin and generation process methods of data, including the party or parties involved in the generation, introduction and/or mash-up processes for data.

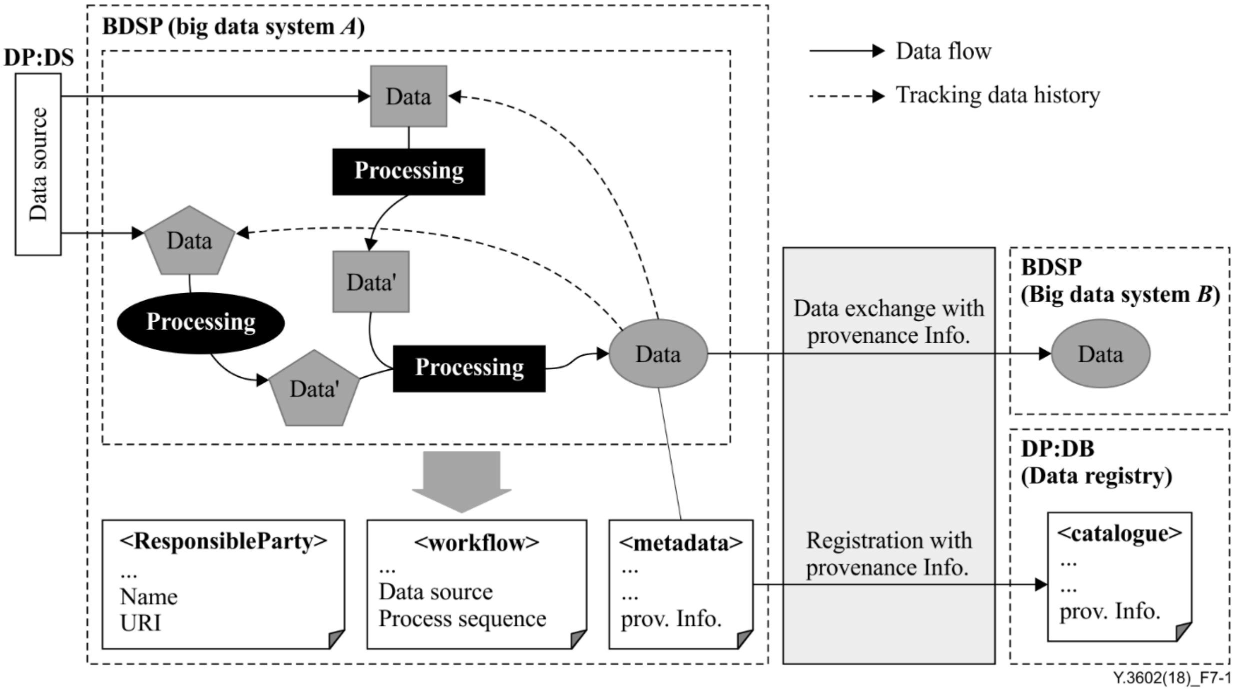

BDSP (big data system A)
DP:DS
Data source
Data flow
Tracking data history
Data
Processing
Data
Data'
Processing
Processing
Data'
Data
Data exchange with provenance Info.
BDSP (Big data system B)
Data
DP:DB (Data registry)
<ResponsibleParty> ... Name URI
<workflow> ... Data source Process sequence
<metadata> ... ... prov. Info.
Registration with provenance Info.
<catalogue> ... ... prov. Info.
Y.3602(18)_F7-1

Figure 7-1 – Using data provenance in a big data ecosystem

Figure 7-1 shows the use of data provenance in a big data ecosystem:

– when data is imported from an outside data source (data provider (DP):data supplier (DS)) and stored, BDSP (big data system *A*) generates metadata based on the importing context (e.g., responsible party information, time, size) and these metadata elements are used for constructing data PI;

– BDSP *A* monitors and stores information about data analysis processes as a form of PI to ensure the reliability of data quality and reproducibility of the analysis result;

– when BDSP *A* exports data to BDSP *B* or registers data catalogue information to a data registry in a data market (DP:data broker (DB)), BDSP *A* delivers the corresponding data PI.

NOTE – Concerning data exported or registered by the BDSP, the level of detail of data PI provided by the BDSP depends on the data or service policy applied by the BDSP.

7.2 Conceptual model of PI

Big data PI described in this clause is based on an extension of the general data provenance concept described in clause 6.1.

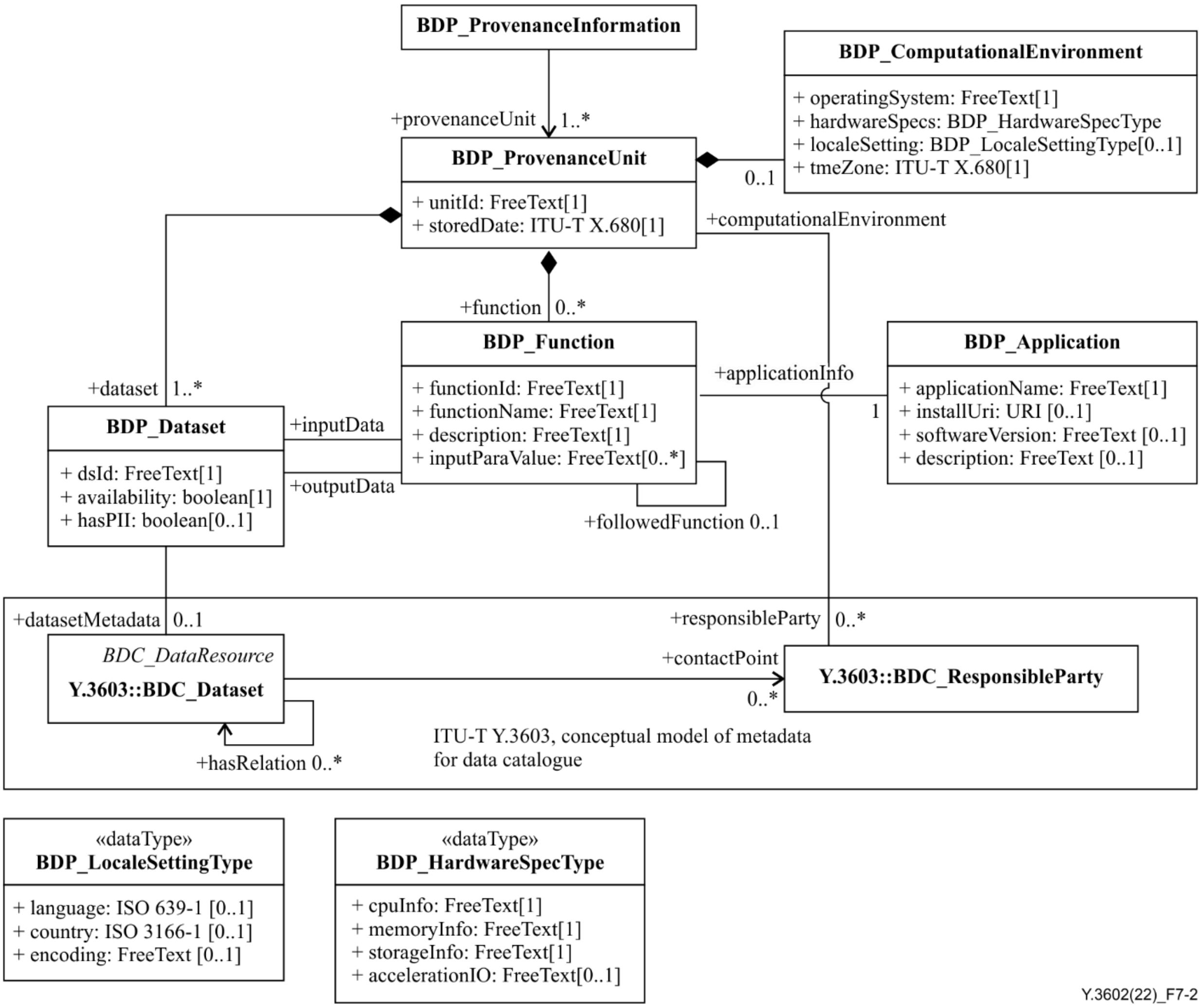

BDP_ProvenanceInformation
+provenanceUnit
1..*
BDP_ComputationalEnvironment
+ operatingSystem: FreeText[1]
+ hardwareSpecs: BDP_HardwareSpecType
+ localeSetting: BDP_LocaleSettingType[0..1]
+ tmeZone: ITU-T X.680[1]
BDP_ProvenanceUnit
+ unitId: FreeText[1]
+ storedDate: ITU-T X.680[1]
0..1
+computationalEnvironment
+function
0..*
BDP_Function
+ functionId: FreeText[1]
+ functionName: FreeText[1]
+ description: FreeText[1]
+ inputParaValue: FreeText[0..*]
+applicationInfo
1
BDP_Application
+ applicationName: FreeText[1]
+ installUri: URI [0..1]
+ softwareVersion: FreeText [0..1]
+ description: FreeText [0..1]
+dataset
1..*
BDP_Dataset
+ dsId: FreeText[1]
+ availability: boolean[1]
+ hasPII: boolean[0..1]
+inputData
+outputData
+followedFunction 0..1
+datasetMetadata
0..1
+responsibleParty
0..*
BDC_DataResource
Y.3603::BDC_Dataset
+contactPoint
0..*
Y.3603::BDC_ResponsibleParty
+hasRelation 0..*
ITU-T Y.3603, conceptual model of metadata for data catalogue
«dataType»
BDP_LocaleSettingType
+ language: ISO 639-1 [0..1]
+ country: ISO 3166-1 [0..1]
+ encoding: FreeText [0..1]
«dataType»
BDP_HardwareSpecType
+ cpuInfo: FreeText[1]
+ memoryInfo: FreeText[1]
+ storageInfo: FreeText[1]
+ accelerationIO: FreeText[0..1]
Y.3602(22)_F7-2

Figure 7-2 – Conceptual model for big data PI

Figure 7-2 shows a conceptual model for big data PI. Big data PI (*BDP_ProvenanceInformation*) is a set of big data provenance units (*BDP_ProvenanceUnit*).

A big data provenance unit corresponds to the minimum set of big data PI. It provides information about data ownership or authority (*BDC_ResponsibleParty*), the data processing environment (*BDP_ComputationalEnvironment*), and a sequence of functions (*BDP_Function*) with input and output data (*BDP_Dataset*) which are involved in a data cleansing, preparation, mash-up, or analysis.

The scope of *hasPII* attribute in *BDP_Dataset* for data provenance in this Recommendation does not include the identification of data subjects or the collection of information to identify them to ensure the use of this Recommendation complies with national legal and regulatory requirements.

NOTE 1 – A workflow depicts the actual sequence of the involved functions in a data processing sequence. In the big data PI model, a workflow can be derived through the use of the association *(+followedFunction)* which describes the sequence of functions *(BDP_function)* involved in the workflow.

NOTE 2 – *BDP_Dataset* stores whether the dataset exists in the storage and whether personally identifiable information (PII) is included. This class also refers *BDC_Dataset* metadata that is composed of identifiable information (e.g., access information, type and format of data, data size, date) for a dataset.

NOTE 3 – *BDC_ResponsibleParty* is defined in clause 8 of [ITU-T Y.3603].

7.2.2 Classes and data type

The following descriptors are used to explain the data dictionaries in the clauses that follow:

- Name/Role name: title of class (grey coloured row in the tables), association (noted as role) and attribute;
- Description: a short description about 'Name/Role name';
- M/R/O: the mandatory/recommended/optional (M/R/O) necessity of each class, association, and attribute;
- Cardinality: the maximum number of instances that the metadata entity or metadata element may have. Single occurrences are shown by "1"; repeating occurrences are represented by "N".
- Domain: the values allowed, or the use of free text placed on the contents of the field. "Free text" indicates that no restrictions are placed on the contents of the field.
- Domain code: designated code used in place of the concept.

7.2.2.1 Class: BDP_ProvenanceInformation

Table 7-1 shows the data dictionary of *BDP_ProvenanceInformation*.

Table 7-1 – PI overview

	Name/Role name	Description	M/R/O	Cardinality	Domain
1	BDP_ProvenanceInformation	root entity which defines PI	M	1	Line 2
2	role: provenanceUnit	provenance unit for the PI	M	N	BDP_ProvenanceUnit

7.2.2.2 Class: BDP_ProvenanceUnit

Table 7-2 shows the data dictionary of *BDP_ProvenanceUnit*.

Table 7-2 – Provenance unit

	Name/Role name	Description	M/R/O	Cardinality	Domain
3	BDP_ProvenanceUnit	a record of the most recent changes to the data	M	N	Line 4 to 9
4	role: computationalEnvironment	computational environment information for the provenance unit	O	1	BDP_Computational Environment
5	role: responsibilityParty	responsible party information for the data catalogue	O	N	BDC_Responsibility Party
6	role: dataset	dataset information for the data catalogue	M	1	BDP_Dataset
7	role: function	function information for the data catalogue	R	N	BDP_Function
8	unitId	a unique provenance unit name	M	1	free text

Table 7-2 – Provenance unit

	Name/Role name	Description	M/R/O	Cardinality	Domain
9	storedDate	the date of storing the provenance unit	R	N	[ITU-T X.680]

NOTE - [ITU-T X.680] provides support for the full range of time types specified in ISO 8601:2004 [b-ISO8601].

7.2.2.3 Class: BDP_ComputationalEnvironment

Table 7-3 shows the data dictionary of *BDP_ComputationalEnvironment*.

Table 7-3 – Computational environment

	Name/Role name	Description	M/R/O	Cardinality	Domain
10	BDP_ComputationalEnvironment	information about the computing environment	O	1	Line 11 to 14
11	operatingSystem	information about the operating system	M	1	free text
12	hardwareSpecs	information about the hardware specification	M	N	BDP_HardwareSpecType
13	localeSetting	information about language, display formats of time, date, currency, etc.	O	1	BDP_LocaleSettingType
14	timeZone	information of time zone	M	N	ITU-T X.680
15	BDP_HardwareSpecType	information required to describe the hardware environment	M	1	Line 16 to 19
16	cpuInfo	information about speed and number of units of CPU	M	1	free text
17	memoryInfo	information about size and speed of memory	M	1	free text
18	storageInfo	information about size and speed of storage	M	1	free text
19	accelerationIO	Information about hardware acceleration unit (e.g., GPU)	O	1	free text
20	BDP_LocaleSettingType	information required to describe locale setting value	O	1	Line 21 to 23

Table 7-3 – Computational environment

	Name/Role name	Description	M/R/O	Cardinality	Domain
21	language	a natural language used for hardware	O	1	ISO 639-1 [b-ISO639]
22	country	a country code	O	1	ISO 3166-1 [b-ISO3166]
23	encoding	the value for character encoding setting	O	1	free text (e.g., UTF-8)

7.2.2.4 Class: BDC_ResponsibilityParty

Table 7-4 shows the data dictionary of *BDC_ResponsibleParty*. The detailed information is described in clause 8.2.2 of [ITU-T Y.3603].

Table 7-4 – Responsible party information

	Name/Role name	Description	M/R/O	Cardinality	Domain
24	BDC_ResponsibleParty	name and position information for an individual or organization that is responsible for the data catalogue or dataset	O	N	ITU-T Y.3603

7.2.2.5 Class: BDP_Dataset

Table 7-5 shows the data dictionary of *BDP_Dataset*.

Table 7-5 – Dataset

	Name/Role name	Description	M/R/O	Cardinality	Domain
25	BDP_Dataset	a collection of data which is recorded in a provenance unit	M	N	Line 26 to 30
26	role: datasetMetadata	dataset metadata for the dataset	O	1	BDC_Dataset
27	dsId	A unique identifier of the dataset	M	1	free text
28	availability	information about whether the data exists in the storage or has been deleted	M	1	boolean
29	hasPII	information about whether the data includes PII or not	R	1	boolean
30	BDC_Dataset	a collection of data that is available for access and/or download	O	1	[ITU-T Y.3603]

7.2.2.6 Class: BDP_Function

Table 7-6 shows the data dictionary of *BDP_Function*.

Table 7-6 – Function

	Name/Role name	Description	M/R/O	Cardinality	Domain
20	BDP_Function	information about the used function	O	N	Line 21 to 30
21	role: inputData	information of input dataset	O	N	BDP_Dataset
22	role: outputData	information of output dataset	O	N	BDP_Dataset
23	role: followedFunction	information about the function used after	O	1	BDP_Function
24	role: applicationInfo	information about the software providing the function used	M	1	BDP_Application
25	functionId	a unique function name	M	N	free text
26	functionName	The alternative name of the function	O	1	free text
27	description	a short introduction to the function	M	1	free text
28	inputParaValue	the input parameter value	O	N	free text

7.2.2.7 Class: BDP_Application

Table 7-7 shows the data dictionary of *BDP_Application*.

Table 7-7 – Application

29	BDP_Application	information about the software providing the function used	M	1	Line 30 to 33
30	applicationName	name of software application	M	1	free text
31	installUri	URI at which the app may be installed	R	1	URI
32	softwareVersion	version of software	R	1	free text (major.minor.patch)
33	description	additional information about the application	O	1	free text

7.2.3 Examples of PI

Figure 7-3 illustrates an example of capturing the provenance unit on Data *C*.

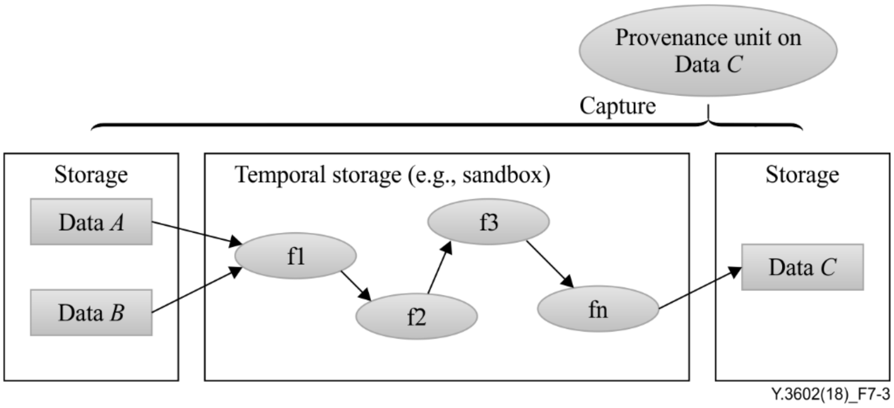


Figure 7-3 – An example of capturing a provenance unit on a data instance

The capturing of the provenance unit occurs simultaneously when Data *C* is stored in the data storage. Even though all functions have input and output data, the information about the first input data and output data are captured in a provenance unit with process steps described by a sequence of functions.

NOTE – In Figure 7-3, Data *A* and *B* are input data, and Data *C* is output data.

Figure 7-4 shows an example of big data PI illustrated with a graph model. The PI of *Data d* corresponds to the aggregation of the provenance unit of *Data c* and the provenance unit of *Data d*.

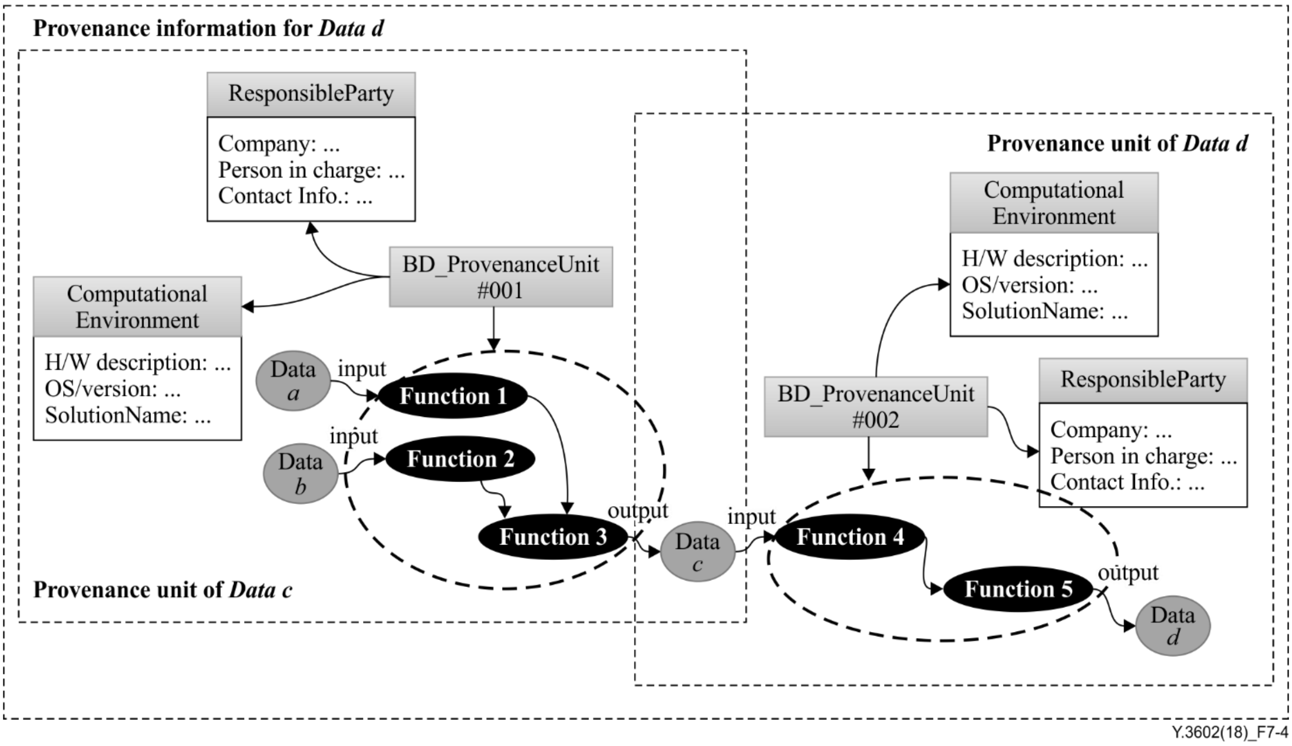


Figure 7-4 – An example of big data PI

7.3 Operations on PI

According to the change of data state such as storing data, updating data and deleting data, a provenance unit is recorded, kept, combined or deleted. Figure 7-5 shows the relationship between data state change and provenance operations.

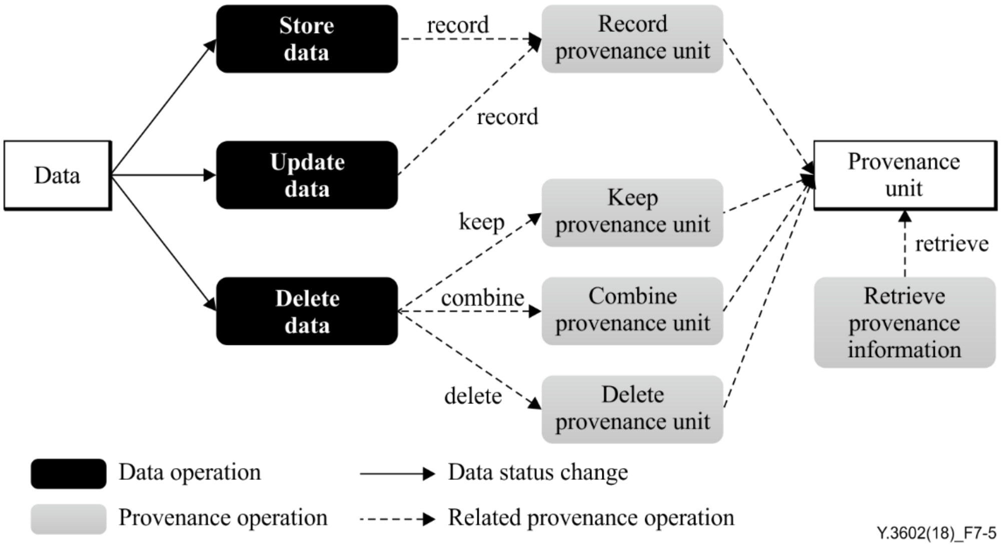


Figure 7-5 – Provenance operations according to data state change

Operations to manage PI are:

- record provenance unit (see clause 7.3.1);
- keep provenance unit (see clause 7.3.2.1);
- combine provenance unit (see clause 7.3.2.2);
- delete provenance unit (see clause 7.3.2.3);
- retrieve PI (see clause 7.3.3).

7.3.1 Record provenance unit

Figure 7-6 shows an example of provenance units recording when the data is stored and updated.

- **provenance unit recording**: When Data 1 is stored, BDSP records Provenance unit 1. Data 1 is updated to Data 2 then to Data 3, BDSP records Provenance unit 2 then Provenance unit 3 sequentially.

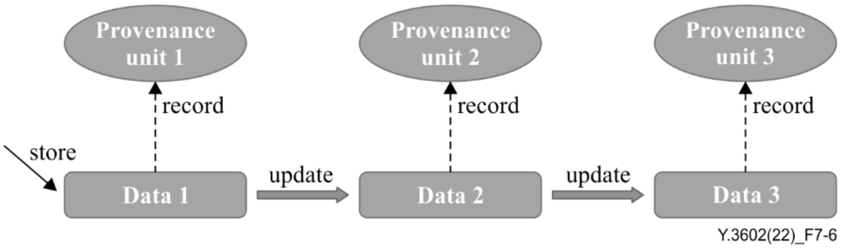


Figure 7-6 – Record provenance units

7.3.2 PI in case of deleting data

In case of deleting data, the BDSP provenance management system acts in the three ways described in clauses 7.3.2.1 to 7.3.2.3.

7.3.2.1 Keep provenance unit

When Data 2 is deleted from storage, the provenance management system keeps its provenance unit (Provenance unit 2) to support the provenance of data within process steps. Figure 7-7 shows an example where provenance units are kept after the deletion of data.

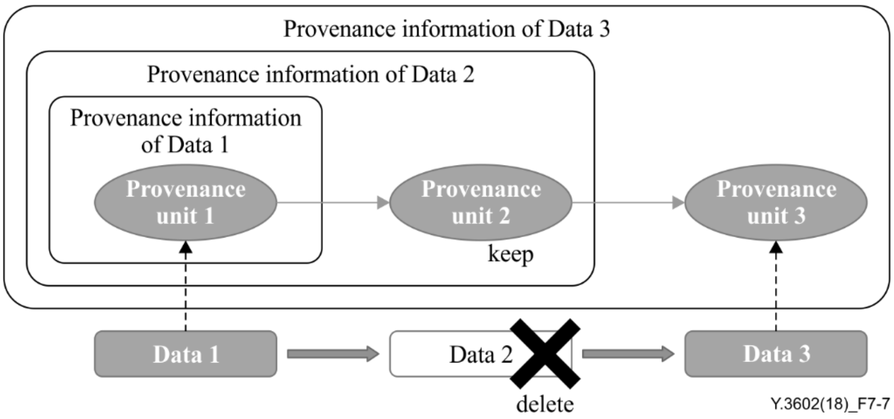

Provenance information of Data 3
Provenance information of Data 2
Provenance information of Data 1
Provenance unit 1
Provenance unit 2
keep
Provenance unit 3
Data 1
Data 2
delete
Data 3
Y.3602(18)_F7-7

Figure 7-7 – Keep a provenance unit

7.3.2.2 Combine provenance units

When Data 2 is deleted from storage, the provenance management system combines its provenance unit (Provenance unit 2) with the forward nearest provenance unit (Provenance unit 3 as shown in the example) within process steps. This is described in the example given in Figure 7-8.

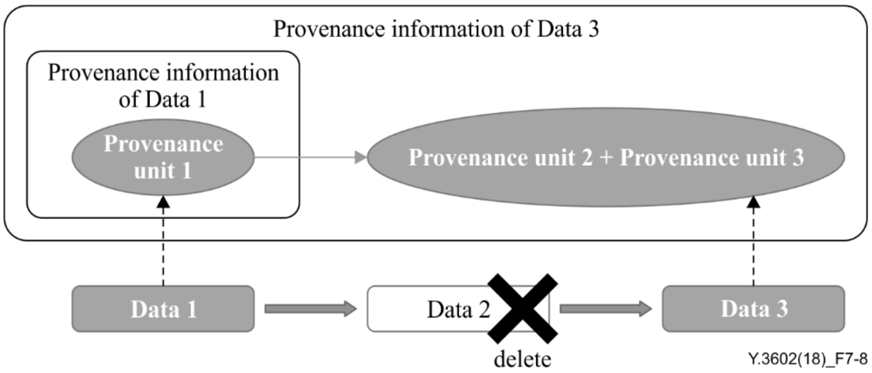

Provenance information of Data 3
Provenance information of Data 1
Provenance unit 1
Provenance unit 2 + Provenance unit 3
Data 1
Data 2
delete
Data 3
Y.3602(18)_F7-8

Figure 7-8 – Combine provenance units

7.3.2.3 Delete provenance unit

When data 3 is deleted from storage, the provenance management system deletes the provenance unit (Provenance unit 3) together with the data instance (Data 3) since placed at the right end node of process steps. This is illustrated in the example given in Figure 7-9.

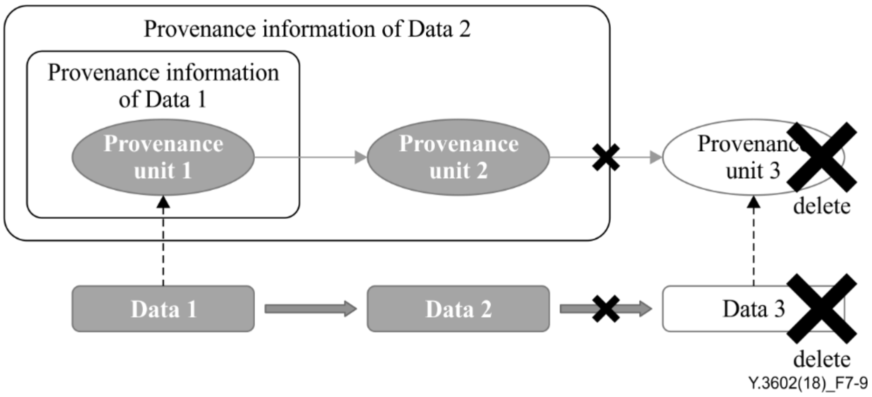


Figure 7-9 – Delete provenance units

7.3.3 Retrieve PI

Figure 7-10 shows an example where PI is retrieved. As shown in the figure, when the PI of Data 3 is requested from an application (step 1), the BDSP traces the history of data based on each provenance unit (step 2). The BDSP aggregates the identified provenance units (step 3) and provides the aggregated PI of Data 3 back to the requesting application (step 4).

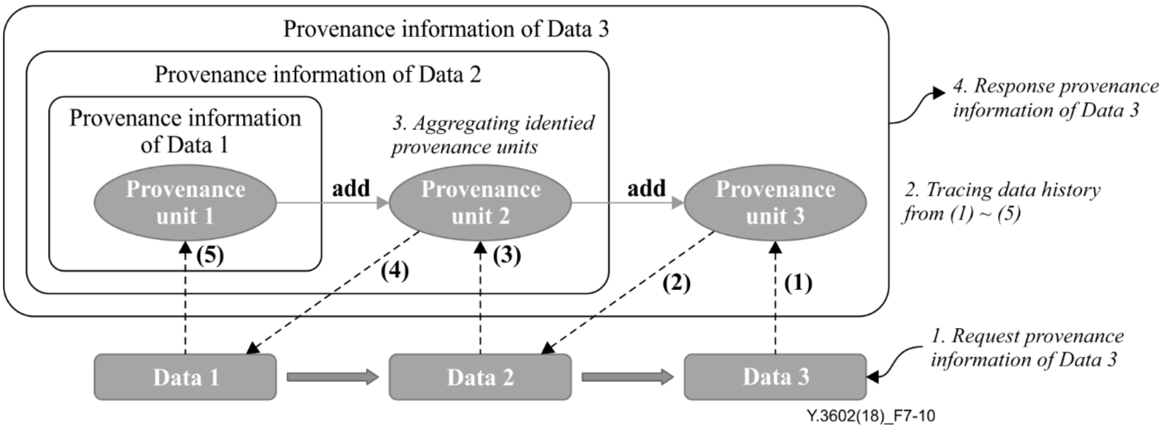

Provenance information of Data 3
Provenance information of Data 2
Provenance information of Data 1
3. Aggregating identied provenance units
4. Response provenance information of Data 3
Provenance unit 1
add
Provenance unit 2
add
Provenance unit 3
2. Tracing data history from (1) ~ (5)
(5)
(4)
(3)
(2)
(1)
Data 1
Data 2
Data 3
1. Request provenance information of Data 3
Y.3602(18)_F7-10

Figure 7-10 – Retrieval of PI

7.4 Logical components for big data provenance management

Figure 7-11 illustrates the logical components for managing big data provenance.

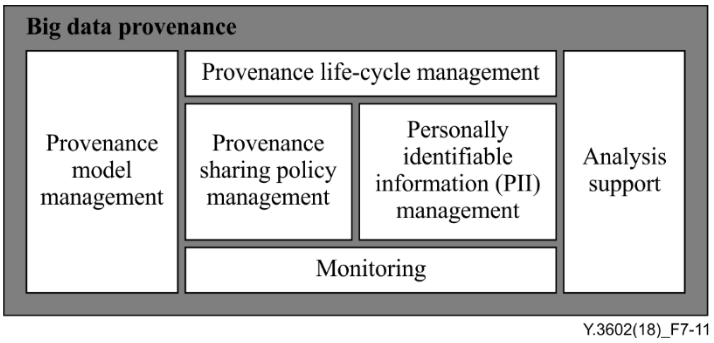

Big data provenance
Provenance model management
Provenance life-cycle management
Provenance sharing policy management
Personally identifiable information (PII) management
Monitoring
Analysis support
Y.3602(18)_F7-11

Figure 7-11 – Configuration of logical components for big data provenance

The logical components shown in Figure 7-11 are as follows:

– **provenance model management**: This logical component manages PI compatibility among different BDSPs. This logical component validates the PI received from outside based on the big data provenance model (see clause 7.2). The valid PI is then encoded as a common model and delivered to the provenance lifecycle management logical component to store it;

– **provenance lifecycle management**: This logical component performs the recording and deletion of PI according to store, update and delete data (see clauses 7.3.1 and 7.3.2). This logical component supports retrieving PI (see clause 7.3.3);

– **analysis support**: This logical component extracts the workflows from PI, and stores them. From the stored workflows, this logical component retrieves the candidate analysis workflows based on the information of BDSP's data analysis functions and data. For the request of PI or workflow from the different system (e.g., external BDSP), this logical component may check the adaptability of the computational environment, and map to an equivalent function for that system. This logical component also supports automating the data analysis process based on the update of data, adding user annotation on PI, and managing the relationship between BDSP's functions and data;

NOTE 1 – Based on the relationship between functions and data in PI, it is possible to query the list of available data with functions and the list of functions applicable to the data.

– **provenance sharing policy management**: This logical component manages multiple sharing policies on PI. When exporting PI, BDSP checks the sharing policy and may simplify the PI before sending it to another BDSP;

NOTE 2 – PI may contain confidential information (e. g. key-idea about a specific analysis process). In this reason, a BDSP formulates a policy and processes to allow appropriate and responsible sharing of PI. A sharing policy can be realized by abstracting the details of PI on several levels.

– **personally identifiable information (PII) management**: This logical component checks whether the data instance contains PII when recording a provenance unit. This logical component also requests to BDSP a protection mechanism to be applied on PI that includes PII;

– **monitoring**: This logical component monitors changes in value about the computational environment and responsible party in PI. When changes are detected, this logical component updates them.

8 Functional requirements of big data provenance

8.1 Provenance lifecycle requirements

Provenance lifecycle requirements include:

– **(provenance model description)** It is required that BDSP supports an information model for big data provenance;

NOTE 1 – Big data PI model includes function name and its uses, computational environment, data type and format of input and output data, input parameters, responsible party information, etc.

NOTE 2 – Example of computational environment information is OS, H/W description, locale settings, time zone, etc.

– **(a common format for exchange)** It is recommended that BDSP supports encoding and decoding PI in a common format for use on different systems;

NOTE 3 –In this Recommendation, the meaning of encoding is the process of converting PI into a specialized format. Decoding is the opposite process.

– **(provenance recording initiation)** It is required that BDSP records provenance unit when data is stored;

NOTE 4 – The information contained in the metadata (from DP:DB or generated by BDSP) can be used for recording the provenance unit.

– **(storing provenance unit)** It is required that BDSP supports a cost-efficient storing mechanism for provenance units;

NOTE 5 – In the case of recording PI for streaming data, an efficient storage mechanism is to define a predetermined period of time to record a provenance unit rather than recording a provenance unit every time data are stored. Data compression techniques can also be considered.

– **(storing PI)** BDSP can optionally support storing PI in advance to satisfy user's retrieval requests on time;

– **(searching provenance unit)** It is required that BDSP supports searching a provenance unit;

– **(combining provenance units)** It is required that BDSP supports the combining of provenance units;

NOTE 6 – The combination of provenance units is needed in case of data deletion (see clause 7.3.2.2).

– **(aggregating PI)** It is required that BDSP supports the aggregation of provenance units;

- **(deleting provenance unit)** It is required that BDSP supports deleting provenance units.

  NOTE 7 – In the case of deleting data, BDSP acts according to one of three mechanisms (keep, combine, delete provenance unit) based on the context (see clause 7.3.2).

  NOTE 8 – The BDSP can maintain the provenance unit when associated data are deleted. This is subject to management policy.

8.2 Analysis support requirements

Analysis support requirements include:

- **(extracting workflow)** It is required that BDSP provides extraction of workflow information from PI;
- **(storing workflow)** It is recommended that BDSP supports storing workflow;

  NOTE 1 – The workflow is stored in the form of the graph, which is organized with the usage frequency of the analysis functions and sequential relationships among them.
- **(retrieving workflow)** It is recommended that BDSP supports workflow retrieval;
- **(providing data list on function)** It is recommended that BDSP provides a list of data related to a given function recorded in a given workflow;
- **(providing function list on data)** It is recommended that BDSP provides a list of functions related to a given data recorded in a given workflow;
- **(data analysis automation)** It is recommended that BDSP supports analysis automation based on workflow;
- **(user annotation)** BDSP can optionally support annotation on PI;
- **(equivalent function for process steps)** It is recommended that BDSP identifies a function for process steps based on PI coming from a different system;

  NOTE 2 – For the identification of such an equivalent function, the name of the function, the format and structure of input and output data of this function, the frequency of analysis functions and the relationship among them can be used.

  NOTE 3 – The results of the equivalent function mapping can be the same function with different names or a combination of functions that provide the same output.
- **(adaptability of the computational environment)** It is recommended that BDSP provides identification of the computational environment based on PI coming from a different system.

8.3 Monitoring requirements

Monitoring requirements include:

- **(monitoring computational environment)** It is required that BDSP monitors the changes regarding the computational environment;
- **(monitoring responsible party)** It is required that BDSP monitors the changes regarding responsible party information;
- **(applying the monitoring result)** It is required that BDSP updates the monitoring results to record provenance units.

  NOTE – The monitoring results include the change of computational environment and responsible party.

8.4 Policy management requirements

Policy management requirements include:

- **(verifying PII)** It is required that BDSP provides verifying PII in a data instance when recording the corresponding provenance unit;

  NOTE 1 – Verification of PII follows BDSP's policy on PII.

NOTE 2 – In a provenance unit, data instance information (*BD_DataInstance*) includes information about whether PII is contained or not (see clause 7.2).

- **(protecting PII)** It is required that BDSP provides a protection mechanism for PII in data sources;

  NOTE 3 – When PII is included in data sources, BDSP decides to omit it or not based on the user's access authority.

- **(simplifying PI)** It is recommended that BDSP supports simplifying PI based on a sharing policy;

  NOTE 4 – Methods of PI simplification include multiple levels of detail and encoding formats, etc.

  NOTE 5 – About "sharing policy", see NOTE 2 in clause 7.4.

- **(sharing level of provenance)** It is required that BDSP supports sharing policy according to the different levels of provenance.

  NOTE 6 – The provenance level determines the level of provenance, and it is determined by the sharing policy. PI contains process steps with the applied functions, intermediate data, and responsible party information. For transferring the PI, the PI can be simplified according to the sharing policy.

  NOTE 7 – About "level of provenance", see NOTE 2 in clause 7.4.

9 Security considerations

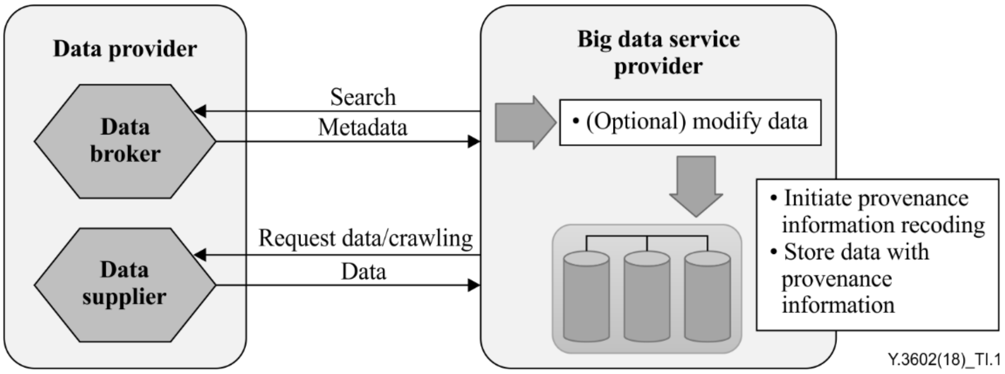
Relevant security requirements of [b-ITU-T Y.2201], [b-ITU-T Y.2701] and applicable X, Y and M series of ITU-T Recommendations need to be taken into consideration, including access control, authentication, data confidentiality, data retention policy, network security, data integrity, availability and protection of personal information.

Appendix I

Use cases of big data provenance

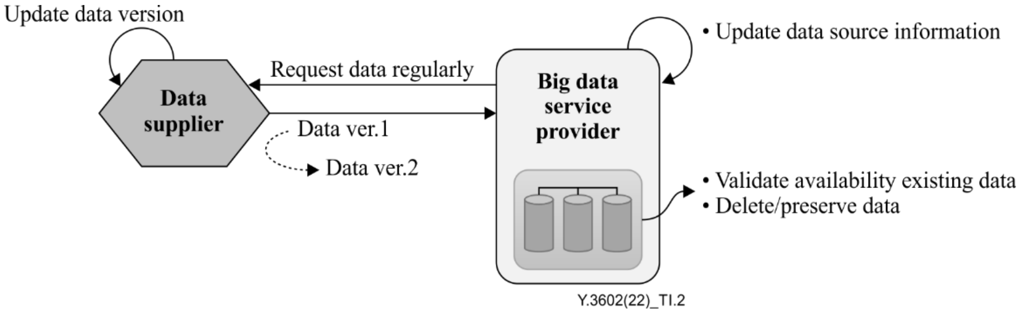
(This appendix does not form an integral part of this Recommendation.)

Table I.1 – Use case – Initiating PI

Title	Initiating PI record
Description	A BDSP requests data from a data provider (DP), and gets the data. At this time, BDSP stores data as it is or modifies the original data to fit its own database. During this process, BDSP starts to record PI which includes the origin of data described in metadata, functional processes which were applied to modify data, and stores them to a distributed database.
Roles/sub-roles	DP:DS DP:DB BDSP
Figure (optional)	 Data provider Data broker Data supplier Search Metadata Request data/crawling Data Big data service provider • (Optional) modify data • Initiate provenance information recoding • Store data with provenance information Y.3602(18)_TI.1
Pre-conditions (optional)	DP:DS published metadata to DP:DB BDSP searches data from DP:DB and requests data to DP:DS or crawls data from DP:DS
Post-conditions (optional)	

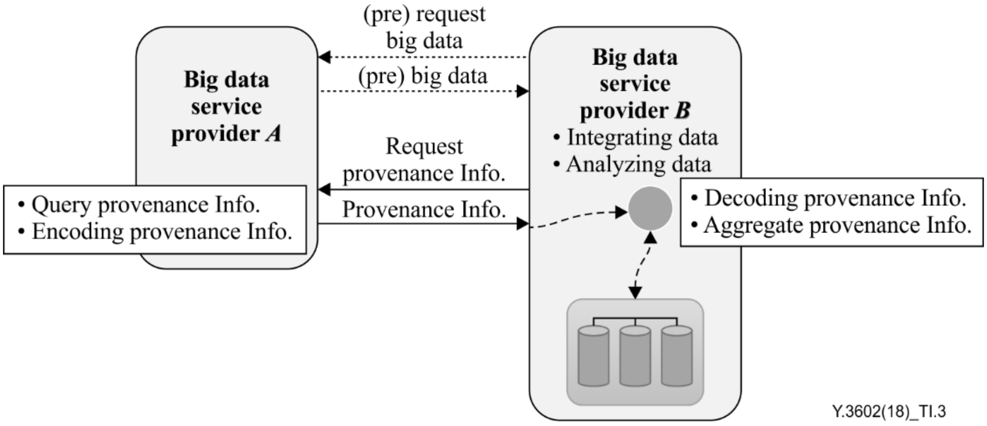
Derived requirement	– Provenance recording initiation (see clause 8.1);
	– Provenance model description (see clause 8.1);
	– Storing provenance unit (see clause 8.1);
	– User annotation (see clause 8.2);
	– Sharing level of provenance (see clause 8.4).

Table I.2 – Use case – Updating data and managing PI

Title	Updating data and managing PI in big data system
Description	<Adding PI based on the change of data source status> – A BDSP uses data from DP:DS regularly. When DP:DS updates the status of data (e.g., schema version upgrade, change of responsible person's information), BDSP updates the change of source information; <Managing PI caused by deleting or preserving data from local data storage> – BDSP deletes or archives the stored data for storage efficiency as well as for other management reasons. BDSP monitors the PI and decides whether to delete it or not.
Roles/Sub roles	DP:DS BDSP
Figure (optional)	 Update data version Data supplier Request data regularly Data ver.1 Data ver.2 Big data service provider • Update data source information • Validate availability existing data • Delete/preserve data Y.3602(22)_TI.2
Pre-conditions (optional)	BDSP stored extensible markup language (XML) data with an external uniform resource identifier (URI) from DP:DS.
Post-conditions (optional)	

Derived requirement	– Searching provenance unit (see clause 8.1);
	– Combining provenance units (see clause 8.1);
	– Deleting provenance unit (see clause 8.1);
	– Monitoring computational environment (see clause 8.3);
	– Monitoring responsible party (see clause 8.3);
	– Applying monitoring results (see clause 8.3).

Table I.3 – Use case – Sharing and aggregating PI

Title	Sharing and aggregating PI
Description	Two collaborating BDSPs share PI with each other. When BDSP B uses data from BDSP A: – BDSP B requests and receives data from BDSP A; – BDSP B initiates PI recording with storing data from BDSP A; If BDSP B needs more information (e.g., data history) on received data from: – BDSP A then requests the PI about received data to BDSP A.; – BDSP B aggregates the delivered PI with the local one that was created when the data from BDSP A was stored.
Roles/sub-roles	– BDSP
	 (pre) request big data (pre) big data Big data service provider *A* Big data service provider *B* • Integrating data • Analyzing data Request provenance Info. Provenance Info. • Query provenance Info. • Encoding provenance Info. • Decoding provenance Info. • Aggregate provenance Info. Y.3602(18)_TI.3
Pre-conditions (optional)	BDSP *B* requested big data to BDSP A and received the data.
Post-conditions (optional)	BDSP *B* stored the aggregated PI.

Derived requirement	– Searching provenance unit (see clause 8.1);
	– Common format for exchange (see clause 8.1);
	– Aggregating PI (see clause 8.1);
	– Simplifying PI (see clause 8.4).

Table I.4 – Use case – Reuse data processing methods

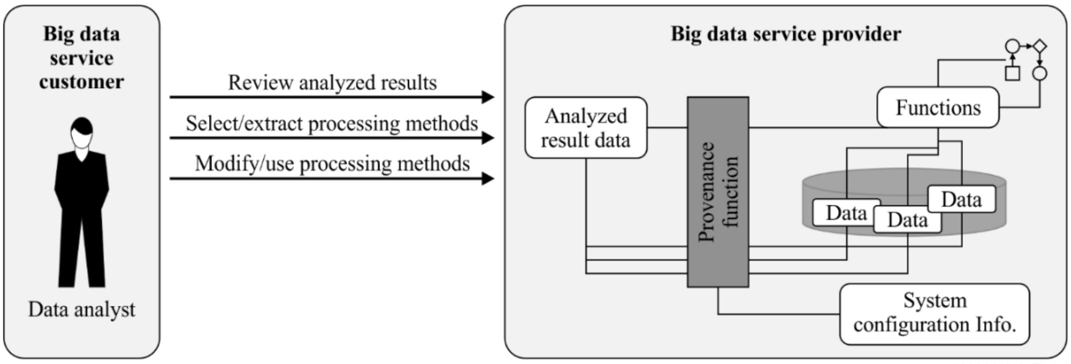
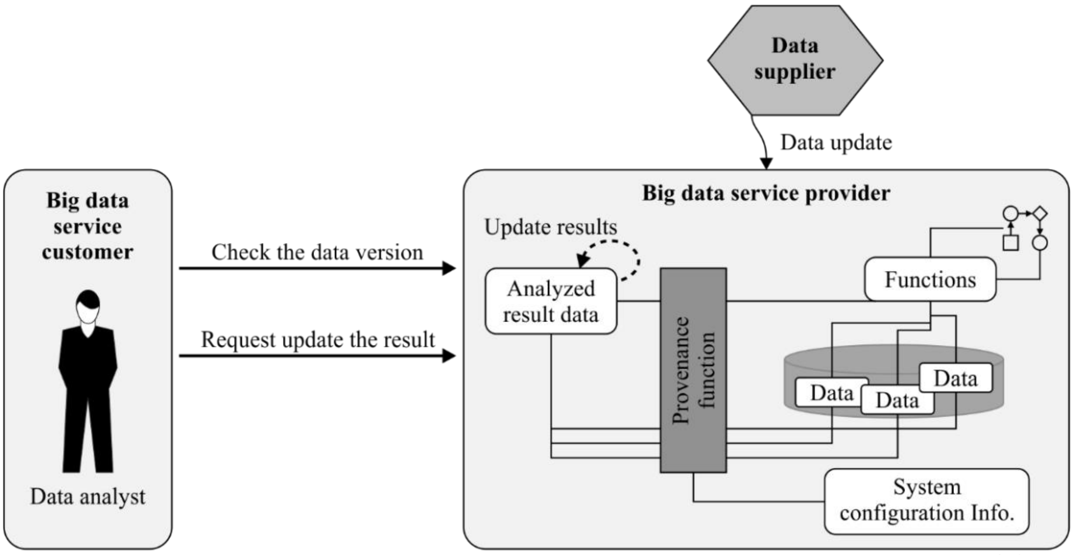
Title	Reusing and automating data processing methods with big data PI
Description	A data analyst (e.g., a BDC playing the analyse data activity [ITU-T Y.3600]) is preparing an experiment based on existing big data analysis results with a different data source. To do this, the data analyst uses the provenance functions provided by the BDSP to extract the data analysis process and apply it. – data analyst reviews analysis results; – data analyst selects an analysis result to reuse its processing methods; – data analyst extracts the data processing methods and related data from the PI of the analysis result and modifies them to fit for the new experiment; – data analyst applies the data processing method. Data analyst is using analysis automation based on data updates by using the PI. Data analyst sets up the periodic analysis based on the updated data according to the update period of the data.
Roles/Sub roles	– DP:DS – BDSP – big data service customer (BDC)
Figure (optional)	 Big data service customer Data analyst Review analyzed results Select/extract processing methods Modify/use processing methods Big data service provider Analyzed result data Provenance function Functions Data Data Data System configuration Info. Y.3602(18)_TI.4-1 <Reusing data processing methods>  Data supplier Data update Big data service customer Data analyst Check the data version Request update the result Big data service provider Update results Analyzed result data Provenance function Functions Data Data Data System configuration Info. Y.3602(18)_TI.4-2 <Automating big data analysis process>

Table I.4 – Use case – Reuse data processing methods

Pre-conditions (optional)	BDSP stores PI about the analysed result data.
Post-conditions (optional)	

Derived requirement	– Extracting workflow (see clause 8.2);
	– Retrieving workflow (see clause 8.2);
	– Data analysis automation (see clause 8.2);
	– Providing data list on function (see clause 8.2);
	– Providing function list on data (see clause 8.2).

Table I.5 – Use case – Managing personal information

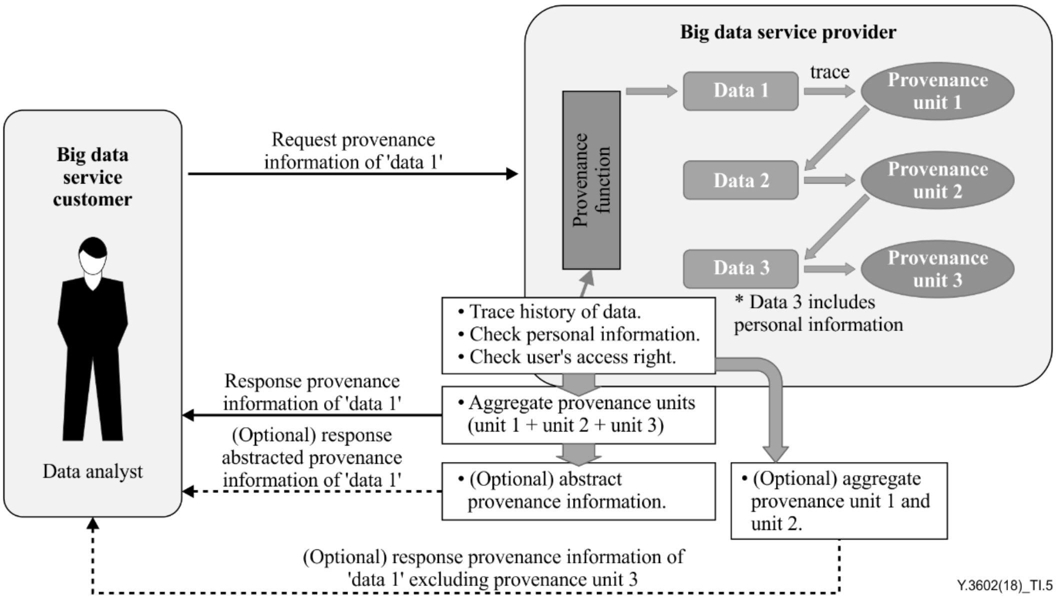
Title	Managing personal information
Description	A data analyst (e.g., a BDC playing the analyse data activity [ITU-T Y.3600] requests the PI on 'data 1' to BDSP. At this time, BDSP: – traces the history of 'data 1'; – checks whether each data includes personal information and find out 'data 3' includes personal information; – checks BDC's access right on 'data 3'. If BDC has the access right on 'data 3', BDSP aggregates provenance units and responds it. If not, BDSP responds the PI excluding provenance unit 3 or abstracts the PI based on a predefined policy and responds it.
Roles/Sub roles	– BDC – BDSP
Figure (optional)	 Big data service provider trace Data 1 Provenance unit 1 Data 2 Provenance unit 2 Data 3 Provenance unit 3 Provenance function * Data 3 includes personal information Big data service customer Data analyst Request provenance information of 'data 1' • Trace history of data. • Check personal information. • Check user's access right. Response provenance information of 'data 1' • Aggregate provenance units (unit 1 + unit 2 + unit 3) (Optional) response abstracted provenance information of 'data 1' • (Optional) abstract provenance information. • (Optional) aggregate provenance unit 1 and unit 2. (Optional) response provenance information of 'data 1' excluding provenance unit 3 Y.3602(18)_TI.5
Pre-conditions (optional)	

Table I.5 – Use case – Managing personal information

Post-conditions (optional)	
Derived requirement	– Aggregating PI (see clause 8.1) – Verifying PII (see clause 8.4) – Protecting PII (see clause 8.4) – Sharing level for provenance (see clause 8.4) – Simplifying PI (see clause 8.4)

Table I.6 – Use case – Reuse PI from the different analysis system

Title	Reusing PI from the different analysis system
Description	A data analyst (e.g., a BDC playing the analyse data activity [ITU-T Y.3600]) is preparing an experiment using the PI received from BDSP *A* with his/her own data. To this end, a data analyst uses the provenance functions (provided by BDSP *A*) to extract the data analysis workflow and reconfigure it to fit BDSP *B*'s analysis environment. – data analyst requests the PI to BDSP *B*; – BDSP *B* decodes the PI; – BDSP *B* extracts the workflow from the PI; – BDSP *B* checks the adaptability of workflow and converts it to be available; A. BDSP *B* checks the adaptability of the computational environment of BDSP *A*. B. BDSP *B* maps the process steps extracted from the PI and the functions supported by the BDSP *B*. C. When the functions are not mapped correctly, BDSP *B* examines for the alternate functions, and the data analyst selects the functions from them. – data analyst applies the process steps to his/her own data.
Roles/sub roles	– BDSP – BDC
Figure (optional)	

Table I.6 – Use case – Reuse PI from the different analysis system

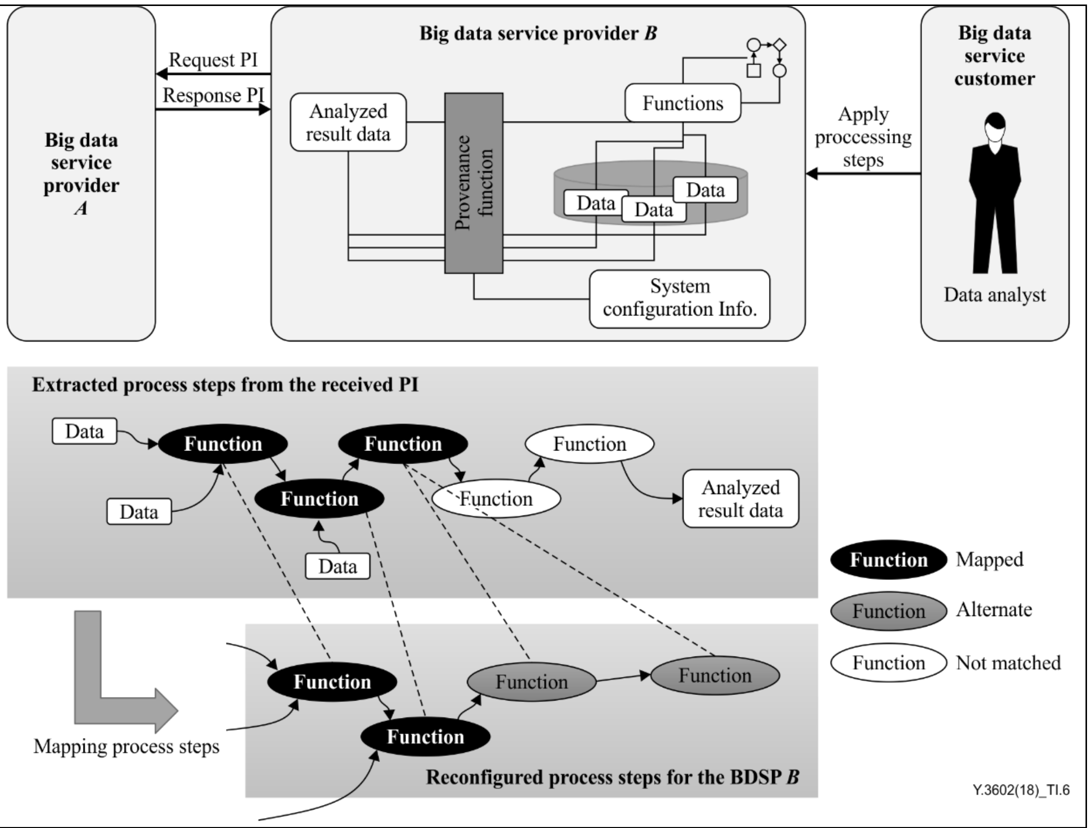
	 Big data service provider A Request PI Response PI Big data service provider B Analyzed result data Provenance function Functions Data Data Data System configuration Info. Apply proccessing steps Big data service customer Data analyst Extracted process steps from the received PI Data Function Function Function Data Function Function Analyzed result data Data Function Mapped Function Alternate Function Not matched Mapping process steps Function Function Function Function Reconfigured process steps for the BDSP B Y.3602(18)_TI.6
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirement	– Provenance model description (see clause 8.1); – Common format for exchange (see clause 8.1); – Extracting workflow (see clause 8.2); – Adaptability of the computational environment (see clause 8.2); – Equivalent function for process steps (see clause 8.2).

Table I.7 – Use case –Scientific workflow provenance

Title	PI collection and query of scientific workflow
Description	Scientific workflow is a typical application system, facilitating e-Science. Scientists model, design, execute, debug, re-configure and re-run their analysis. PI in the scientific workflow system is very useful for a scientist to interpret their workflow results and for other scientists to establish trust in the experimental result. The BDSP (scientific workflow system) initiates PI automatically, and stores the PI in the database. BDC (scientists) (e.g., a BDC playing the analyse data activity [ITU-T Y.3600]) need to retrieve the PI to confirm the source of the scientific data in an experiment or analysis process. The BDSP stores PI automatically when data status is changed to support frequent retrieving.

Table I.7 – Use case –Scientific workflow provenance

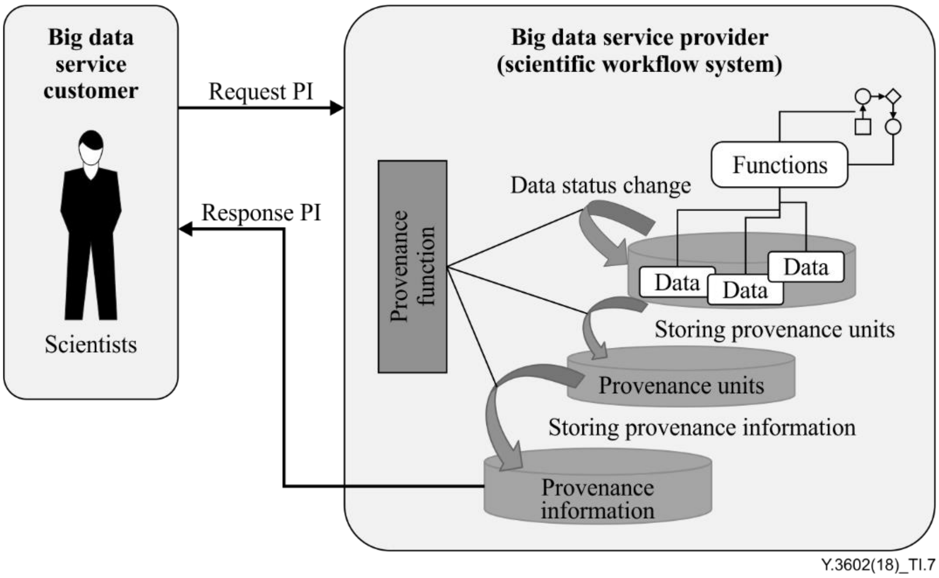
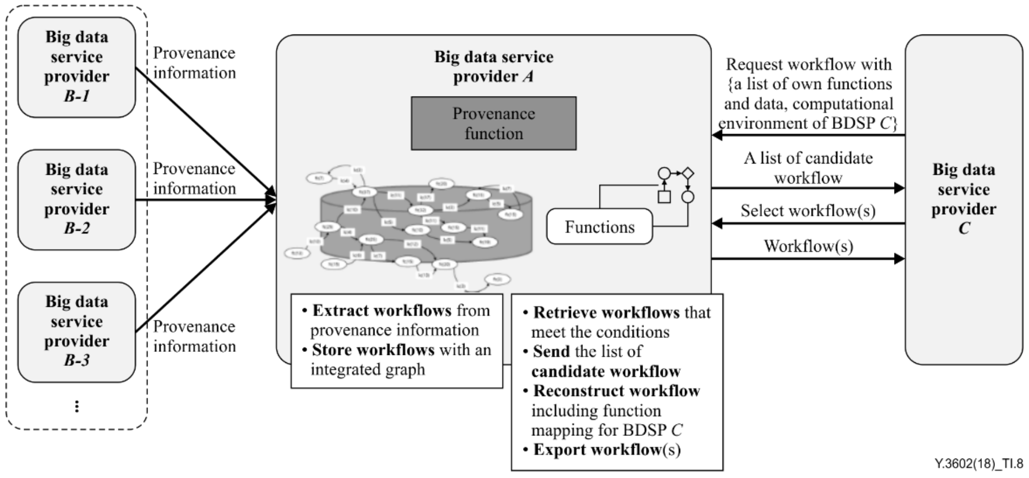
Roles/sub-roles	– BDSP – BDC
Figure (optional)	 Big data service customer Scientists Request PI Response PI Big data service provider (scientific workflow system) Provenance function Functions Data status change Data Data Data Storing provenance units Provenance units Storing provenance information Provenance information Y.3602(18)_TI.7
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirement	– Aggregating PI (see clause 8.1) – Storing PI (see clause 8.1)

Table I.8 – Use case – Extracting analysis workflow from the accumulated PI

Title	Extracting analysis workflow from the accumulated PI
Description	BDSP A collects the PI from the different BDSPs to accumulate the analysis workflows and reuse them. BDSP C wants to find analysis methods that can be applied to its system functions and data through BDSP A. Accordingly: – BDSP A collects PI from BDSP B; – BDSP A extracts the workflows from PI; – BDSP A stores the workflows with an integrated graph organized by the usage frequency of the analysis functions and sequential relationship among them; – BDSP C requests workflow with a list of its own functions and data as well as the information of OS, H/W description and on, locale settings; – BDSP A retrieves workflow based on the information that came from BDSP C; – BDSP A sends the list of candidate workflows to BDSP C; – BDSP C selects the workflow that satisfies the analysis purpose; – BDSP A reconstructs the workflow in a form that it can run on the BDSP C and send it to BDSP C; – BDSP C uses the workflow.

Table I.8 – Use case – Extracting analysis workflow from the accumulated PI

Roles/sub-roles	– BDSP
Figure (optional)	
Pre-conditions (optional)	
Post-conditions (optional)	
Derived requirement	– Provenance model description (see clause 8.1) – Extracting workflow (see clause 8.2) – Retrieving workflow (see clause 8.2) – Storing workflow (see clause 8.2) – Equivalent function for process steps (see clause 8.2) – Adaptability of the computational environment (see clause 8.2)

Big data service provider B-1
Big data service provider B-2
Big data service provider B-3
⋮
Provenance information
Provenance information
Provenance information
Big data service provider A
Provenance function
Functions
• Extract workflows from provenance information
• Store workflows with an integrated graph
• Retrieve workflows that meet the conditions
• Send the list of candidate workflow
• Reconstruct workflow including function mapping for BDSP C
• Export workflow(s)
Request workflow with {a list of own functions and data, computational environment of BDSP C}
A list of candidate workflow
Select workflow(s)
Workflow(s)
Big data service provider C
Y.3602(18)_TI.8

Appendix II

Examples of provenance information profile

(This appendix does not form an integral part of this Recommendation.)

This appendix uses ITU-T Y.3603 metadata profile for data catalogue, the W3C data catalog vocabulary (DCAT) in the XML profile, terse resource description framework triple language (Turtle) profile (see [b-W3C DCAT]) and a JSON-based Serialization for Linked Data (JSON-LD). The namespace for DCAT is http://www.w3.org/ns/dcat#, but DCAT makes extensive use of terms from other vocabularies. The extended prefix and its namespace for this appendix are "bdp" and "http://www.itu.int/xml-namespace/itu-t/Y.3602/bigdataprovenance#".

II.1 RDF profile

This clause provides the XML schema with resource description framework (RDF) format corresponding to the UML model in clause 7.

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:bdp="http://www.itu.int/xml-namespace/itu-t/Y.3602/bigdataprovenance#"
        xmlns:bdc="http://www.itu.int/xml-namespace/itu-t/Y.3603/bigdatacatalogue#"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:dct="http://purl.org/dc/terms/"
        xmlns:owl="http://www.w3.org/2002/07/owl#"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema#">

  <rdfs:Class rdf:about="bdp:Application">
    <rdfs:comment>information about the software providing the function used</rdfs:comment>
    <rdfs:subClassOf rdf:resource="owl:Thing"/>
  </rdfs:Class>

  <rdfs:Class rdf:about="bdp:ComputationalEnvironment">
    <rdfs:comment>information about the computing environment</rdfs:comment>
    <rdfs:subClassOf rdf:resource="owl:Thing"/>
  </rdfs:Class>

  <rdfs:Class rdf:about="bdp:Dataset">
    <rdfs:comment>a collection of data which is recorded in a provenance unit</rdfs:comment>
    <rdfs:subClassOf rdf:resource="owl:Thing"/>
  </rdfs:Class>

  <rdfs:Class rdf:about="bdp:Function">
    <rdfs:comment>information about the used function </rdfs:comment>
    <rdfs:subClassOf rdf:resource="owl:Thing"/>
  </rdfs:Class>

  <rdfs:Class rdf:about="bdp:HardwareSpecType">
    <rdfs:comment>information required to describe the hardware environment</rdfs:comment>
    <rdfs:subClassOf rdf:resource="owl:Thing"/>
  </rdfs:Class>

  <rdfs:Class rdf:about="bdp:LocaleSettingType">
    <rdfs:comment>information required to describe locale setting value </rdfs:comment>
    <rdfs:subClassOf rdf:resource="owl:Thing"/>
  </rdfs:Class>
```

```
<rdfs:Class rdf:about="bdp:ProvenanceInformation">
  <rdfs:comment>root entity which defines PI</rdfs:comment>
  <rdfs:subClassOf rdf:resource="owl:Thing"/>
</rdfs:Class>

<rdfs:Class rdf:about="bdp:ProvenanceUnit">
  <rdfs:comment>a record of the most recent changes to the data</rdfs:comment>
  <rdfs:subClassOf rdf:resource="owl:Thing"/>
</rdfs:Class>

<rdf:Property rdf:about="bdp:accelerationIO">
  <rdfs:comment>information about hardware acceleration unit (e.g., GPU)</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:HardwareSpecType"/>
  <rdfs:range rdf:resource="xsd:string"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:applicationInfo">
  <rdfs:comment>information about the software providing the function used</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:Function"/>
  <rdfs:range rdf:resource="bdp:Application"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:applicationName">
  <rdfs:comment>name of software application</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:Application"/>
  <rdfs:range rdf:resource="xsd:string"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:availability">
  <rdfs:comment>information about whether the data exists in the storage or has been deleted</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:Dataset"/>
  <rdfs:range rdf:resource="xsd:boolean"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:computationalEnvironment">
  <rdfs:comment>computational environment information for the provenance unit</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:ProvenanceUnit"/>
  <rdfs:range rdf:resource="bdp:ComputationalEnvironment"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:country">
  <rdfs:comment>a country code</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:LocaleSettingType"/>
  <rdfs:range rdf:resource="dct:ISO3166"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:cpuInfo">
  <rdfs:comment>information about speed and number of units of CPU</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:HardwareSpecType"/>
  <rdfs:range rdf:resource="xsd:string"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:dataset">
  <rdfs:comment>dataset information for the data catalogue</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:ProvenanceUnit"/>
  <rdfs:range rdf:resource="bdp:Dataset"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:datasetMetadata">
  <rdfs:comment>dataset metadata for the dataset</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:Dataset"/>
  <rdfs:range rdf:resource="bdc:Dataset"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:dsId">
  <rdfs:comment>a unique identifier of the dataset</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:Dataset"/>
  <rdfs:range rdf:resource="xsd:string"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:encoding">
  <rdfs:comment>the value for character encoding setting</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:LocaleSettingType"/>
  <rdfs:range rdf:resource="xsd:string"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:followedFunction">
  <rdfs:comment>information about the function used after</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:Function"/>
  <rdfs:range rdf:resource="bdp:Function"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:function">
  <rdfs:comment>function information for the data catalogue</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:ProvenanceUnit"/>
  <rdfs:range rdf:resource="bdp:Function"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:functionId">
  <rdfs:comment>a unique function name</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:Function"/>
  <rdfs:range rdf:resource="xsd:string"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:functionName">
  <rdfs:comment>the alternative name of the function</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:Function"/>
  <rdfs:range rdf:resource="xsd:string"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:hardwareSpecs">
  <rdfs:comment>information about the hardware specification</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:ComputationalEnvironment"/>
  <rdfs:range rdf:resource="bdp:HardwareSpecType"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:hasPII">
  <rdfs:comment>information about whether the data includes PII or not</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:Dataset"/>
  <rdfs:range rdf:resource="xsd:boolean"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:inputParaValue">
  <rdfs:comment>the input parameter value</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:Function"/>
  <rdfs:range rdf:resource="xsd:string"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:installUri">
  <rdfs:comment>URI at which the app may be installed</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:Application"/>
  <rdfs:range rdf:resource="xsd:anyURI"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:langauge">
  <rdfs:comment>a natural language used for hardware</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:LocaleSettingType"/>
  <rdfs:range rdf:resource="dct:RFC4646"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:localeSetting">
  <rdfs:comment>information about language, display formats of time, date, currency, etc.</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:ComputationalEnvironment"/>
  <rdfs:range rdf:resource="bdp:LocaleSettingType"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:memoryInfo">
  <rdfs:comment>information about size and speed of memory</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:HardwareSpecType"/>
  <rdfs:range rdf:resource="xsd:string"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:operatingSystem">
  <rdfs:comment>information about the operating system</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:ComputationalEnvironment"/>
  <rdfs:range rdf:resource="xsd:string"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:provenanceUnit">
  <rdfs:comment>provenance unit for the PI</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:ProvenanceInformation"/>
  <rdfs:range rdf:resource="bdp:ProvenanceUnit"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:responsibleParty">
  <rdfs:comment>responsible party information for the data catalogue</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:ProvenanceUnit"/>
  <rdfs:range rdf:resource="bdc:ResponsibleParty"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:softwareVersion">
  <rdfs:comment>version of software</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:Application"/>
  <rdfs:range rdf:resource="xsd:string"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:storageInfo">
  <rdfs:comment>information about size and speed of storage</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:HardwareSpecType"/>
  <rdfs:range rdf:resource="xsd:string"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:storedDate">
  <rdfs:comment>the date of storing the provenance unit</rdfs:comment>
  <rdfs:domain rdf:resource="bdp:ProvenanceUnit"/>
  <rdfs:range rdf:resource="dct:W3CDTF"/>
</rdf:Property>

<rdf:Property rdf:about="bdp:timeZone">
  <rdfs:comment>information of time zone</rdfs:comment>
```

```
    <rdfs:domain rdf:resource="bdp:ComputationalEnvironment"/>
    <rdfs:range rdf:resource="dct:W3CDTF"/>
  </rdf:Property>

  <rdf:Property rdf:about="bdp:unitId">
    <rdfs:comment>a unique provenance unit name</rdfs:comment>
    <rdfs:domain rdf:resource="bdp:ProvenanceUnit"/>
    <rdfs:range rdf:resource="xsd:string"/>
  </rdf:Property>

  <rdf:Description rdf:about="dc:description">
    <rdfs:comment>a short introduction to the function or the application </rdfs:comment>
    <rdfs:domain rdf:resource="bdp:Application"/>
    <rdfs:domain rdf:resource="bdp:Function"/>
    <rdfs:range rdf:resource="xsd:string"/>
  </rdf:Description>

</rdf:RDF>
```

II.2 Turtle profile

This clause provides the Turtle format schema corresponding to the UML model in clause 7.

```
# baseURI: http://www.itu.int/xml-namespace/itu-t/Y.3602/bigdataprovenance
# imports: http://purl.org/dc/elements/1.1/
# imports: http://purl.org/dc/terms/
# imports: https://www.w3.org/2002/07/owl#
# imports: http://www.itu.int/xml-namespace/itu-t/Y.3603/bigdatacatalogue
# prefix: bdp

@prefix bdp: <http://www.itu.int/xml-namespace/itu-t/Y.3602/bigdataprovenance#> .
@prefix bdc: <http://www.itu.int/xml-namespace/itu-t/Y.3603/bigdatacatalogue#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

bdp:Application
  rdf:type rdfs:Class ;
  rdfs:comment "information about the software providing the function used" ;
  rdfs:subClassOf owl:Thing ;
.
bdp:ComputationalEnvironment
  rdf:type rdfs:Class ;
  rdfs:comment "information about the computing environment" ;
  rdfs:subClassOf owl:Thing ;
.
bdp:Dataset
  rdf:type rdfs:Class ;
  rdfs:comment "a collection of data which is recorded in a provenance unit" ;
  rdfs:subClassOf owl:Thing ;
.
bdp:Function
  rdf:type rdfs:Class ;
  rdfs:comment "information about the used function " ;
  rdfs:subClassOf owl:Thing ;
.
```

```
bdp:HardwareSpecType
  rdf:type rdfs:Class ;
  rdfs:comment "information required to describe the hardware environment" ;
  rdfs:subClassOf owl:Thing ;
.
bdp:LocaleSettingType
  rdf:type rdfs:Class ;
  rdfs:comment "information required to describe locale setting value " ;
  rdfs:subClassOf owl:Thing ;
.
bdp:ProvenanceInformation
  rdf:type rdfs:Class ;
  rdfs:comment "root entity which defines PI" ;
  rdfs:subClassOf owl:Thing ;
.
bdp:ProvenanceUnit
  rdf:type rdfs:Class ;
  rdfs:comment "a record of the most recent changes to the data" ;
  rdfs:subClassOf owl:Thing ;
.
bdp:accelerationIO
  rdf:type rdf:Property ;
  rdfs:comment "information about hardware acceleration unit (e.g., GPU)" ;
  rdfs:domain bdp:HardwareSpecType ;
  rdfs:range xsd:string ;
.
bdp:applicationInfo
  rdf:type rdf:Property ;
  rdfs:comment "information about the software providing the function used" ;
  rdfs:domain bdp:Function ;
  rdfs:range bdp:Application ;
.
bdp:applicationName
  rdf:type rdf:Property ;
  rdfs:comment "name of software application" ;
  rdfs:domain bdp:Application ;
  rdfs:range xsd:string ;
.
bdp:availability
  rdf:type rdf:Property ;
  rdfs:comment "information about whether the data exists in the storage or has been deleted" ;
  rdfs:domain bdp:Dataset ;
  rdfs:range xsd:boolean ;
.
bdp:computationalEnvironment
  rdf:type rdf:Property ;
  rdfs:comment "computational environment information for the provenance unit" ;
  rdfs:domain bdp:ProvenanceUnit ;
  rdfs:range bdp:ComputationalEnvironment ;
.
bdp:country
  rdf:type rdf:Property ;
  rdfs:comment "a country code" ;
  rdfs:domain bdp:LocaleSettingType ;
  rdfs:range dct:ISO3166 ;
.
bdp:cpuInfo
  rdf:type rdf:Property ;
  rdfs:comment "information about speed and number of units of CPU" ;
  rdfs:domain bdp:HardwareSpecType ;
  rdfs:range xsd:string ;
.
bdp:dataset
  rdf:type rdf:Property ;
  rdfs:comment "dataset information for the data catalogue" ;
  rdfs:domain bdp:ProvenanceUnit ;
  rdfs:range bdp:Dataset ;
.
bdp:datasetMetadata
  rdf:type rdf:Property ;
  rdfs:comment "dataset metadata for the dataset" ;
  rdfs:domain bdp:Dataset ;
  rdfs:range bdc:Dataset ;
.
bdp:dsId
  rdf:type rdf:Property ;
  rdfs:comment "a unique identifier of the dataset" ;
  rdfs:domain bdp:Dataset ;
  rdfs:range xsd:string ;
.
bdp:encoding
  rdf:type rdf:Property ;
  rdfs:comment "the value for character encoding setting" ;
  rdfs:domain bdp:LocaleSettingType ;
  rdfs:range xsd:string ;
.
bdp:followedFunction
  rdf:type rdf:Property ;
  rdfs:comment "information about the function used after" ;
  rdfs:domain bdp:Function ;
  rdfs:range bdp:Function ;
.
bdp:function
  rdf:type rdf:Property ;
  rdfs:comment "function information for the data catalogue" ;
  rdfs:domain bdp:ProvenanceUnit ;
  rdfs:range bdp:Function ;
.
bdp:functionId
  rdf:type rdf:Property ;
  rdfs:comment "a unique function name" ;
  rdfs:domain bdp:Function ;
  rdfs:range xsd:string ;
.
bdp:functionName
  rdf:type rdf:Property ;
  rdfs:comment "the alternative name of the function" ;
  rdfs:domain bdp:Function ;
  rdfs:range xsd:string ;
.
bdp:hardwareSpecs
  rdf:type rdf:Property ;
  rdfs:comment "information about the hardware specification" ;
  rdfs:domain bdp:ComputationalEnvironment ;
  rdfs:range bdp:HardwareSpecType ;
.
bdp:hasPII
  rdf:type rdf:Property ;
  rdfs:comment "information about whether the data includes PII or not" ;
  rdfs:domain bdp:Dataset ;
  rdfs:range xsd:boolean ;
.
bdp:inputParaValue
  rdf:type rdf:Property ;
  rdfs:comment "the input parameter value" ;
  rdfs:domain bdp:Function ;
  rdfs:range xsd:string ;
.
bdp:installUri
  rdf:type rdf:Property ;
  rdfs:comment "URI at which the app may be installed" ;
  rdfs:domain bdp:Application ;
  rdfs:range xsd:anyURI
.
bdp:langauge
  rdf:type rdf:Property ;
  rdfs:comment "a natural language used for hardware" ;
  rdfs:domain bdp:LocaleSettingType ;
  rdfs:range dct:RFC4646 ;
.
bdp:localeSetting
  rdf:type rdf:Property ;
  rdfs:comment "information about language, display formats of time, date, currency, etc." ;
  rdfs:domain bdp:ComputationalEnvironment ;
  rdfs:range bdp:LocaleSettingType ;
.
bdp:memoryInfo
  rdf:type rdf:Property ;
  rdfs:comment "information about size and speed of memory" ;
  rdfs:domain bdp:HardwareSpecType ;
  rdfs:range xsd:string ;
.
bdp:operatingSystem
  rdf:type rdf:Property ;
  rdfs:comment "information about the operating system" ;
  rdfs:domain bdp:ComputationalEnvironment ;
  rdfs:range xsd:string ;
.
bdp:provenanceUnit
  rdf:type rdf:Property ;
  rdfs:comment "provenance unit for the PI" ;
  rdfs:domain bdp:ProvenanceInformation ;
  rdfs:range bdp:ProvenanceUnit ;
.
bdp:responsibleParty
  rdf:type rdf:Property ;
  rdfs:comment "responsible party information for the data catalogue" ;
  rdfs:domain bdp:ProvenanceUnit ;
  rdfs:range bdc:ResponsibleParty ;
.
bdp:softwareVersion
  rdf:type rdf:Property ;
  rdfs:comment "version of software" ;
  rdfs:domain bdp:Application ;
  rdfs:range xsd:string ;
.
bdp:storageInfo
  rdf:type rdf:Property ;
  rdfs:comment "information about size and speed of storage" ;
  rdfs:domain bdp:HardwareSpecType ;
  rdfs:range xsd:string ;
.
bdp:storedDate
  rdf:type rdf:Property ;
  rdfs:comment "the date of storing the provenance unit" ;
  rdfs:domain bdp:ProvenanceUnit ;
  rdfs:range dct:W3CDTF ;
.
bdp:timeZone
  rdf:type rdf:Property ;
  rdfs:comment "information of time zone" ;
  rdfs:domain bdp:ComputationalEnvironment ;
  rdfs:range dct:W3CDTF ;
.
bdp:unitId
  rdf:type rdf:Property ;
  rdfs:comment "a unique provenance unit name" ;
  rdfs:domain bdp:ProvenanceUnit ;
  rdfs:range xsd:string ;
.
dc:description
  rdfs:comment "a short introduction to the function or the application " ;
  rdfs:domain bdp:Application ;
  rdfs:domain bdp:Function ;
  rdfs:range xsd:string ;
.
```

II.3 JSON-LD profile

This clause provides the JSON-LD format schema corresponding to the UML model in clause 7.

```
{
  "@context": {
    "bdp": "http://www.itu.int/xml-namespace/itu-t/Y.3602/bigdataprovenance#",
    "bdc": "http://www.itu.int/xml-namespace/itu-t/Y.3603/bigdatacatalogue#",
    "dc": "http://purl.org/dc/elements/1.1/",
    "dct": "http://purl.org/dc/terms/",
    "owl": "http://www.w3.org/2002/07/owl#",
    "rdf": "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
    "rdfs": "http://www.w3.org/2000/01/rdf-schema#",
    "xsd": "http://www.w3.org/2001/XMLSchema#"
  },
  "@graph": [
    {
      "@id": "bdp:Application",
      "rdf:type": {
        "@id": "rdfs:Class"
      },
      "rdfs:comment": "information about the software providing the function used",
      "rdfs:subClassOf": {
        "@id": "owl:Thing"
      }
    },
    {
      "@id": "bdp:ComputationalEnvironment",
      "rdf:type": {
        "@id": "rdfs:Class"
      },
      "rdfs:comment": "information about the computing environment",
      "rdfs:subClassOf": {
        "@id": "owl:Thing"
      }
    },
    {
      "@id": "bdp:Dataset",
      "rdf:type": {
        "@id": "rdfs:Class"
      },
      "rdfs:comment": "a collection of data which is recorded in a provenance unit",
      "rdfs:subClassOf": {
        "@id": "owl:Thing"
      }
    },
    {
      "@id": "bdp:Function",
      "rdf:type": {
        "@id": "rdfs:Class"
      },
      "rdfs:comment": "information about the used function ",
      "rdfs:subClassOf": {
        "@id": "owl:Thing"
      }
    },
    {
      "@id": "bdp:HardwareSpecType",
      "rdf:type": {
        "@id": "rdfs:Class"
      },
      "rdfs:comment": "information required to describe the hardware environment",
      "rdfs:subClassOf": {
        "@id": "owl:Thing"
      }
    },
    {
      "@id": "bdp:LocaleSettingType",
      "rdf:type": {
        "@id": "rdfs:Class"
      },
      "rdfs:comment": "information required to describe locale setting value ",
      "rdfs:subClassOf": {
        "@id": "owl:Thing"
      }
    },
    {
      "@id": "bdp:ProvenanceInformation",
      "rdf:type": {
        "@id": "rdfs:Class"
      },
      "rdfs:comment": "root entity which defines PI",
      "rdfs:subClassOf": {
        "@id": "owl:Thing"
      }
    },
    {
      "@id": "bdp:ProvenanceUnit",
      "rdf:type": {
        "@id": "rdfs:Class"
      },
      "rdfs:comment": "a record of the most recent changes to the data",
      "rdfs:subClassOf": {
        "@id": "owl:Thing"
      }
    },
    {
      "@id": "bdp:accelerationIO",
      "rdf:type": {
        "@id": "rdf:Property"
      },
      "rdfs:comment": "information about hardware acceleration unit (e.g., GPU)",
      "rdfs:domain": {
        "@id": "bdp:HardwareSpecType"
      },
      "rdfs:range": {
        "@id": "xsd:string"
      }
    },
    {
      "@id": "bdp:applicationInfo",
      "rdf:type": {
        "@id": "rdf:Property"
      },
      "rdfs:comment": "information about the software providing the function used",
      "rdfs:domain": {
        "@id": "bdp:Function"
      },
      "rdfs:range": {
        "@id": "bdp:Application"
      }
    },
    {
      "@id": "bdp:applicationName",
      "rdf:type": {
        "@id": "rdf:Property"
      },
      "rdfs:comment": "name of software application",
      "rdfs:domain": {
        "@id": "bdp:Application"
      },
      "rdfs:range": {
        "@id": "xsd:string"
      }
    },
    {
      "@id": "bdp:availability",
      "rdf:type": {
        "@id": "rdf:Property"
      },
      "rdfs:comment": "information about whether the data exists in the storage or has been deleted",
      "rdfs:domain": {
        "@id": "bdp:Dataset"
      },
      "rdfs:range": {
        "@id": "xsd:boolean"
      }
    },
    {
      "@id": "bdp:computationalEnvironment",
      "rdf:type": {
        "@id": "rdf:Property"
      },
  "rdfs:comment": "computational environment information for the provenance unit",
  "rdfs:domain": {
    "@id": "bdp:ProvenanceUnit"
  },
  "rdfs:range": {
    "@id": "bdp:ComputationalEnvironment"
  }
},
{
  "@id": "bdp:country",
  "rdf:type": {
    "@id": "rdf:Property"
  },
  "rdfs:comment": "a country code",
  "rdfs:domain": {
    "@id": "bdp:LocaleSettingType"
  },
  "rdfs:range": {
    "@id": "dct:ISO3166"
  }
},
{
  "@id": "bdp:cpuInfo",
  "rdf:type": {
    "@id": "rdf:Property"
  },
  "rdfs:comment": "information about speed and number of units of CPU",
  "rdfs:domain": {
    "@id": "bdp:HardwareSpecType"
  },
  "rdfs:range": {
    "@id": "xsd:string"
  }
},
{
  "@id": "bdp:dataset",
  "rdf:type": {
    "@id": "rdf:Property"
  },
  "rdfs:comment": "dataset information for the data catalogue",
  "rdfs:domain": {
    "@id": "bdp:ProvenanceUnit"
  },
  "rdfs:range": {
    "@id": "bdp:Dataset"
  }
},
{
  "@id": "bdp:datasetMetadata",
  "rdf:type": {
    "@id": "rdf:Property"
  },
  "rdfs:comment": "dataset metadata for the dataset",
  "rdfs:domain": {
    "@id": "bdp:Dataset"
  },
  "rdfs:range": {
    "@id": "bdc:Dataset"
  }
},
{
 "@id": "bdp:dsId",
 "rdf:type": {
  "@id": "rdf:Property"
 },
 "rdfs:comment": "a unique identifier of the dataset",
 "rdfs:domain": {
  "@id": "bdp:Dataset"
 },
 "rdfs:range": {
  "@id": "xsd:string"
 }
},
{
 "@id": "bdp:encoding",
 "rdf:type": {
  "@id": "rdf:Property"
 },
 "rdfs:comment": "the value for character encoding setting",
 "rdfs:domain": {
  "@id": "bdp:LocaleSettingType"
 },
 "rdfs:range": {
  "@id": "xsd:string"
 }
},
{
 "@id": "bdp:followedFunction",
 "rdf:type": {
  "@id": "rdf:Property"
 },
 "rdfs:comment": "information about the function used after",
 "rdfs:domain": {
  "@id": "bdp:Function"
 },
 "rdfs:range": {
  "@id": "bdp:Function"
 }
},
{
 "@id": "bdp:function",
 "rdf:type": {
  "@id": "rdf:Property"
 },
 "rdfs:comment": "function information for the data catalogue",
 "rdfs:domain": {
  "@id": "bdp:ProvenanceUnit"
 },
 "rdfs:range": {
  "@id": "bdp:Function"
 }
},
{
 "@id": "bdp:functionId",
 "rdf:type": {
  "@id": "rdf:Property"
 },
 "rdfs:comment": "a unique function name",
 "rdfs:domain": {
  "@id": "bdp:Function"
```

```
      },
      "rdfs:range": {
        "@id": "xsd:string"
      }
    },
    {
      "@id": "bdp:functionName",
      "rdf:type": {
        "@id": "rdf:Property"
      },
      "rdfs:comment": "the alternative name of the function",
      "rdfs:domain": {
        "@id": "bdp:Function"
      },
      "rdfs:range": {
        "@id": "xsd:string"
      }
    },
    {
      "@id": "bdp:hardwareSpecs",
      "rdf:type": {
        "@id": "rdf:Property"
      },
      "rdfs:comment": "information about the hardware specification",
      "rdfs:domain": {
        "@id": "bdp:ComputationalEnvironment"
      },
      "rdfs:range": {
        "@id": "bdp:HardwareSpecType"
      }
    },
    {
      "@id": "bdp:hasPII",
      "rdf:type": {
        "@id": "rdf:Property"
      },
      "rdfs:comment": "information about whether the data includes PII or not",
      "rdfs:domain": {
        "@id": "bdp:Dataset"
      },
      "rdfs:range": {
        "@id": "xsd:boolean"
      }
    },
    {
      "@id": "bdp:inputParaValue",
      "rdf:type": {
        "@id": "rdf:Property"
      },
      "rdfs:comment": "the input parameter value",
      "rdfs:domain": {
        "@id": "bdp:Function"
      },
      "rdfs:range": {
        "@id": "xsd:string"
      }
    },
    {
      "@id": "bdp:installUri",
      "rdf:type": {
```

```
    "@id": "rdf:Property"
  },
  "rdfs:comment": "URI at which the app may be installed",
  "rdfs:domain": {
    "@id": "bdp:Application"
  },
  "rdfs:range": {
    "@id": "xsd:anyURI"
  }
},
{
  "@id": "bdp:langauge",
  "rdf:type": {
    "@id": "rdf:Property"
  },
  "rdfs:comment": "a natural language used for hardware",
  "rdfs:domain": {
    "@id": "bdp:LocaleSettingType"
  },
  "rdfs:range": {
    "@id": "dct:RFC4646"
  }
},
{
  "@id": "bdp:localeSetting",
  "rdf:type": {
    "@id": "rdf:Property"
  },
  "rdfs:comment": "information about language, display formats of time, date, currency, etc.",
  "rdfs:domain": {
    "@id": "bdp:ComputationalEnvironment"
  },
  "rdfs:range": {
    "@id": "bdp:LocaleSettingType"
  }
},
{
  "@id": "bdp:memoryInfo",
  "rdf:type": {
    "@id": "rdf:Property"
  },
  "rdfs:comment": "information about size and speed of memory",
  "rdfs:domain": {
    "@id": "bdp:HardwareSpecType"
  },
  "rdfs:range": {
    "@id": "xsd:string"
  }
},
{
  "@id": "bdp:operatingSystem",
  "rdf:type": {
    "@id": "rdf:Property"
  },
  "rdfs:comment": "information about the operating system",
  "rdfs:domain": {
    "@id": "bdp:ComputationalEnvironment"
  },
  "rdfs:range": {
    "@id": "xsd:string"
  }
 },
 {
  "@id": "bdp:provenanceUnit",
  "rdf:type": {
   "@id": "rdf:Property"
  },
  "rdfs:comment": "provenance unit for the PI",
  "rdfs:domain": {
   "@id": "bdp:ProvenanceInformation"
  },
  "rdfs:range": {
   "@id": "bdp:ProvenanceUnit"
  }
 },
 {
  "@id": "bdp:responsibleParty",
  "rdf:type": {
   "@id": "rdf:Property"
  },
  "rdfs:comment": "responsible party information for the data catalogue",
  "rdfs:domain": {
   "@id": "bdp:ProvenanceUnit"
  },
  "rdfs:range": {
   "@id": "bdc:ResponsibleParty"
  }
 },
 {
  "@id": "bdp:softwareVersion",
  "rdf:type": {
   "@id": "rdf:Property"
  },
  "rdfs:comment": "version of software",
  "rdfs:domain": {
   "@id": "bdp:Application"
  },
  "rdfs:range": {
   "@id": "xsd:string"
  }
 },
 {
  "@id": "bdp:storageInfo",
  "rdf:type": {
   "@id": "rdf:Property"
  },
  "rdfs:comment": "information about size and speed of storage",
  "rdfs:domain": {
   "@id": "bdp:HardwareSpecType"
  },
  "rdfs:range": {
   "@id": "xsd:string"
  }
 },
 {
  "@id": "bdp:storedDate",
  "rdf:type": {
   "@id": "rdf:Property"
  },
  "rdfs:comment": "the date of storing the provenance unit",
      "rdfs:domain": {
        "@id": "bdp:ProvenanceUnit"
      },
      "rdfs:range": {
        "@id": "dct:W3CDTF"
      }
    },
    {
      "@id": "bdp:timeZone",
      "rdf:type": {
        "@id": "rdf:Property"
      },
      "rdfs:comment": "information of time zone",
      "rdfs:domain": {
        "@id": "bdp:ComputationalEnvironment"
      },
      "rdfs:range": {
        "@id": "dct:W3CDTF"
      }
    },
    {
      "@id": "bdp:unitId",
      "rdf:type": {
        "@id": "rdf:Property"
      },
      "rdfs:comment": "a unique provenance unit name",
      "rdfs:domain": {
        "@id": "bdp:ProvenanceUnit"
      },
      "rdfs:range": {
        "@id": "xsd:string"
      }
    },
    {
      "@id": "dc:description",
      "rdfs:comment": "a short introduction to the function or the application ",
      "rdfs:domain": [
        {
          "@id": "bdp:Application"
        },
        {
          "@id": "bdp:Function"
        }
      ],
      "rdfs:range": {
        "@id": "xsd:string"
      }
    }
  ]
}
```

Bibliography

[b-ITU-T X.1255] Recommendation ITU-T X.1255 (2013), *Framework for discovery of identity management information*.

[b-ITU-T Y.2201] Recommendation ITU-T Y.2201 (2009), *Requirements and capabilities for ITU-T NGN*.

[b-ITU-T Y.2701] Recommendation ITU-T Y.2701 (2007), *Security requirements for NGN release 1*.

[b-ITU-T Y.4100] Recommendation ITU-T Y.4100/Y.2066 (2014), *Common requirements of the Internet of things*.

[b-ISO8601] ISO 8601-2 (2019), *Date and time – Representations for information interchange – Part 2: Extensions.*

[b-ISO639] ISO 639-1 (2002), *Codes for the representation of names of languages – Part 2: Alpha-3 code.*

[b-ISO3166] ISO 3166-1 (2020), *Codes for the representation of names of countries and their subdivisions – Part 1: Country code.*

[b-W3C DCAT] W3C Recommendation (202014), *Data Catalog Vocabulary (DCAT) version 2.*

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Published in Switzerland
Geneva, 2022