



SLOVENSKI STANDARD
oSIST prEN IEC 62541-11:2024
01-marec-2024

Enotna arhitektura OPC - 11. del: Zgodovinski dostop

OPC Unified Architecture - Part 11: Historical Access

OPC Unified Architecture - Teil 11: Zugang zu historischen Daten

Architecture unifiée OPC - Partie 11: Accès à l'historique

Ta slovenski standard je istoveten z: prEN IEC 62541-11:2024

ICS:

25.040.40	Merjenje in krmiljenje industrijskih postopkov	Industrial process measurement and control
35.240.50	Uporabniške rešitve IT v industriji	IT applications in industry

oSIST prEN IEC 62541-11:2024

en,fr,de



PROJECT NUMBER:

IEC 62541-11 ED3

DATE OF CIRCULATION:

2024-01-26

CLOSING DATE FOR VOTING:

2024-04-19

SUPERSEDES DOCUMENTS:

65E/983/RR

IEC SC 65E : DEVICES AND INTEGRATION IN ENTERPRISE SYSTEMS	
SECRETARIAT: United States of America	SECRETARY: Mr Donald (Bob) Lattimer
OF INTEREST TO THE FOLLOWING COMMITTEES:	PROPOSED HORIZONTAL STANDARD: <input type="checkbox"/> Other TC/SCs are requested to indicate their interest, if any, in this CDV to the secretary.
FUNCTIONS CONCERNED: <input type="checkbox"/> EMC <input type="checkbox"/> ENVIRONMENT <input type="checkbox"/> QUALITY ASSURANCE <input type="checkbox"/> SAFETY	
<input checked="" type="checkbox"/> SUBMITTED FOR CENELEC PARALLEL VOTING Attention IEC-CENELEC parallel voting The attention of IEC National Committees, members of CENELEC, is drawn to the fact that this Committee Draft for Vote (CDV) is submitted for parallel voting. The CENELEC members are invited to vote through the CENELEC online voting system.	<input type="checkbox"/> NOT SUBMITTED FOR CENELEC PARALLEL VOTING

oSIST prEN IEC 62541-11:2024

<https://standards.iteh.ai/> This document is still under study and subject to change. It should not be used for reference purposes. [pren-iec-62541-11-2024](https://standards.iteh.ai/pren-iec-62541-11-2024)

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Recipients of this document are invited to submit, with their comments, notification of any relevant "In Some Countries" clauses to be included should this proposal proceed. Recipients are reminded that the CDV stage is the final stage for submitting ISC clauses. (SEE [AC/22/2007](#) OR [NEW GUIDANCE DOC](#)).

TITLE:

OPC Unified Architecture - Part 11: Historical Access

PROPOSED STABILITY DATE: 2026

NOTE FROM TC/SC OFFICERS:

Copyright © 2023 International Electrotechnical Commission, IEC. All rights reserved. It is permitted to download this electronic file, to make a copy and to print out the content for the sole purpose of preparing National Committee positions. You may not copy or "mirror" the file or printed version of the document, or any part of it, for any other purpose without permission in writing from IEC.

CONTENTS

1		
2		
3		
4	FOREWORD	vi
5	1 Scope	1
6	2 Normative references	1
7	3 Terms, definitions, and abbreviations	1
8	3.1 Terms and definitions	1
9	3.2 Abbreviated terms	3
10	4 Concepts	3
11	4.1 General	3
12	4.2 Data architecture	3
13	4.3 Historians and interruption of data collection	4
14	4.4 Modification of Historical Data/Events	5
15	4.5 Timestamps	5
16	4.6 Bounding Values and time domain	6
17	4.7 Changes in AddressSpace over time	7
18	5 Historical Information Model	8
19	5.1 HistoricalNodes	8
20	5.1.1 General	8
21	5.1.2 Annotations Property	8
22	5.2 HistoricalDataNodes	9
23	5.2.1 General	9
24	5.2.2 HistoricalDataConfigurationType	9
25	5.2.3 Attributes	11
26	5.2.4 Historical Data Configuration Object	11
27	5.3 References	11
28	5.3.1 Overview	11
29	5.3.2 HasHistoricalConfiguration ReferenceType	12
30	5.3.3 HasCurrentData ReferenceType	12
31	5.3.4 HasCurrentEvent ReferenceType	13
32	5.4 HistoricalEventNodes	13
33	5.4.1 General	13
34	5.4.2 HistoricalEventFilter Property	14
35	5.4.3 HistoricalEventConfigurationType	14
36	5.4.4 HistoricalEventNode Attributes	15
37	5.5 External History sources	15
38	5.5.1 Introduction	15
39	5.5.2 External Historical Event Node	15
40	5.6 Example Object Models in Historian Servers (informative)	16
41	5.6.1 Overview	16
42	5.6.2 HistoricalDataNodes Address Space Model	16
43	5.6.3 Historical data	17
44	5.6.4 HistoricalEventNodes Address Space Model	17
45	5.6.5 Historical Events	18
46	5.7 Exposing supported functions and capabilities	19
47	5.7.1 General	19
48	5.7.2 HistoryServerCapabilitiesType	20
49	5.7.3 Default Configuration	22

50	5.8	Historical Audit Events	23
51	5.8.1	General	23
52	5.8.2	AuditHistoryEventUpdateEventType	23
53	5.8.3	AuditHistoryValueUpdateEventType	24
54	5.8.4	AuditHistoryAnnotationUpdateEventType	25
55	5.8.5	AuditHistoryDeleteEventType	25
56	5.8.6	AuditHistoryRawModifyDeleteEventType	26
57	5.8.7	AuditHistoryAtTimeDeleteEventType	27
58	5.8.8	AuditHistoryEventDeleteEventType	27
59	5.8.9	AuditHistoryConfigurationChangeEvent	28
60	5.8.10	AuditHistoryBulkInsertEventType	28
61	6	Historical Access specific usage of Services	29
62	6.1	General	29
63	6.2	Historical Nodes StatusCodes	29
64	6.2.1	Overview	29
65	6.2.2	Operation level result codes	29
66	6.2.3	Semantics changed	31
67	6.3	Continuation Points	31
68	6.4	Arrays, index ranges and substrings	32
69	6.5	HistoryReadDetails parameters	33
70	6.5.1	Overview	33
71	6.5.2	ReadEventDetails structure	34
72	6.5.3	ReadRawModifiedDetails structure	37
73	6.5.4	ReadProcessedDetails structure	39
74	6.5.5	ReadAtTimeDetails structure	41
75	6.5.6	ReadAnnotationDataDetails structure	42
76	6.6	HistoryData parameters returned	42
77	6.6.1	Overview	42
78	6.6.2	HistoryData type	42
79	6.6.3	HistoryModifiedData type	43
80	6.6.4	HistoryEvent DataType	43
81	6.6.5	HistoryModifiedEvent DataType	44
82	6.6.6	Annotation DataType	44
83	6.7	HistoryUpdateType Enumeration	45
84	6.8	PerformUpdateType Enumeration	45
85	6.9	HistoryUpdateDetails parameter	46
86	6.9.1	Overview	46
87	6.9.2	UpdateDataDetails structure	47
88	6.9.3	UpdateStructureDataDetails structure	49
89	6.9.4	UpdateEventDetails structure	50
90	6.9.5	DeleteRawModifiedDetails structure	52
91	6.9.6	DeleteAtTimeDetails structure	53
92	6.9.7	DeleteEventDetails structure	53
93	Annex A (informative)	Client conventions	55
94	A.1	How clients may request timestamps	55
95	A.2	Determining the first or last historical data point	56
96		Bibliography	57
97			

98
99**FIGURES**

100	Figure 1 – Possible OPC UA Server supporting Historical Access	4
101	Figure 2 – ReferenceType hierarchy	12
102	Figure 3 – Historical Variable with Historical Data Configuration and Annotations	16
103	Figure 5 – Representation of an Event with History in the AddressSpace	18
104	Figure 7 – Server and HistoryServer Capabilities	20

105

TABLES106
107

108	Table 1 – Bounding Value examples	6
109	Table 2 – Annotations Property	8
110	Table 3 – HistoricalDataConfigurationType definition	9
111	Table 4 – ExceptionDeviationFormat Items	10
112	Table 5 – ExceptionDeviationFormat definition	11
113	Table 6 – Historical Access configuration definition	11
114	Table 7 – HasHistoricalConfiguration ReferenceType	12
115	Table 8 – HasCurrentData ReferenceType	13
116	Table 9 – HasCurrentEvent ReferenceType	13
117	Table 10 – Standard Historical Events Properties	14
118	Table 11 – HistoricalEventConfigurationType definition	14
119	Table 12 – HistoricalExternalEventSourceType definition	15
120	Table 13 – HistoryServerCapabilitiesType Definition	21
121	Table 16 – AuditHistoryEventUpdateEventType definition	24
122	Table 17 – AuditHistoryValueUpdateEventType definition	24
123	Table 18 – AuditHistoryAnnotationUpdateEventType definition	25
124	Table 19 – AuditHistoryDeleteEventType definition	26
125	Table 20 – AuditHistoryRawModifyDeleteEventType definition	26
126	Table 21 – AuditHistoryAtTimeDeleteEventType definition	27
127	Table 22 – AuditHistoryEventDeleteEventType definition	27
128	Table 23 – AuditHistoryConfigurationChangeEvent definition	28
129	Table 24 – AuditHistoryBulkInsertEventType definition	28
130	Table 25 – Bad operation level result codes	30
131	Table 26 – Good operation level result codes	30
132	Table 27 – HistoryReadDetails parameter Symbolic Names	33
133	Table 28 – HistoryReadDetails definition	34
134	Table 29 – ReadEventDetails Structure	34
135	Table 30 – ReadEventDetails definition	35
136	Table 31 – ReadEventDetails2 Structure	36
137	Table 32 – ReadEventDetails2 definition	36
138	Table 33 – ReadRawModifiedDetails structure	37
139	Table 34 – ReadRawModifiedDetails definition	37
140	Table 35 – ReadProcessedDetails structure	39

141	Table 36 – ReadProcessedDetails definition	40
142	Table 37 – ReadAtTimeDetails structure	41
143	Table 38 – ReadAtTimeDetails definition	41
144	Table 39 – ReadAnnotationDataDetails Structure	42
145	Table 40 – ReadAnnotationDataDetails definition	42
146	Table 41 – HistoryData structure	42
147	Table 42 – HistoryData definition	43
148	Table 43 – HistoryModifiedData structure	43
149	Table 44 – HistoryModifiedData definition	43
150	Table 45 – HistoryEvent structure	43
151	Table 46 – HistoryEvent definition	44
152	Table 47 – HistoryModifiedEvent structure	44
153	Table 48 – HistoryModifiedEvent definition	44
154	Table 49 – Annotation Structure	44
155	Table 50 – <i>Annotation</i> definition	45
156	Table 51 – HistoryUpdateType Items	45
157	Table 52 – HistoryUpdateType definition	45
158	Table 53 – PerformUpdateType Items	45
159	Table 54 – PerformUpdateType definition	46
160	Table 55 – HistoryUpdateDetails parameter Symbolic Names	46
161	Table 56 – HistoryUpdateDetails Structure	47
162	Table 57 – HistoryUpdateDetails definition	47
163	Table 58 – UpdateDataDetails Structure	48
164	Table 59 – UpdateDataDetails definition	48
165	Table 60 – UpdateStructureDataDetails Structure	49
166	Table 61 – UpdateStructureDataDetails definition	49
167	Table 62 – UpdateEventDetails Structure	50
168	Table 63 – UpdateEventDetails definition	51
169	Table 64 – DeleteRawModifiedDetails Structure	52
170	Table 65 – DeleteRawModifiedDetails definition	52
171	Table 66 – DeleteAtTimeDetails Structure	53
172	Table 67 – DeleteAtTimeDetails definition	53
173	Table 68 – DeleteEventDetails Structure	53
174	Table 69 – DeleteEventDetails definition	54
175	Table A.1 – Time keyword definitions	56
176	Table A.2 – Time offset definitions	56
177		
178		

INTERNATIONAL ELECTROTECHNICAL COMMISSION

OPC UNIFIED ARCHITECTURE –

Part 11: Historical Access

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.

2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.

3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.

8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62541-11 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

This fourth edition cancels and replaces the third edition published in 2020. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

a) Added functionality to support retrieving of modified events.

b) Added Event to indicate when a backfill occurred.

c) Defined a new ReferenceType that can be used to indicate an external node.

d) Improved text to better explain the concept of annotation and remove conflicting explanations.

e) Defined a default historian configuration (and where to find it).

f) Added HistoricalEventConfigurationType, which provides general configuration information about the historical Event storage.

g) Updated text and added optional fields to HA configuration object to allow configuration to be defined for periodic data collection, not just for exception-based collection.

232 h) Provided ObjectType that can be used for external event collection and also provided example
233 how historians can be configured.

234

235 The text of this International Standard is based on the following documents:

CDV	Report on voting
65E/XX/CDV	65E/XX/RVC

236

237 Full information on the voting for the approval of this International Standard can be found in the report
238 on voting indicated in the above table.

239 This document has been drafted in accordance with the ISO/IEC Directives, Part 2.

240 Throughout this document and the other parts of the IEC 62541 series, certain document conventions
241 are used:

242 *Italics* are used to denote a defined term or definition that appears in the "Terms and definition" clause
243 in one of the parts of the IEC 62541 series.

244 *Italics* are also used to denote the name of a service input or output parameter or the name of a
245 structure or element of a structure that are usually defined in tables.

246 The *italicized terms and names* are, with a few exceptions, written in camel-case (the practice of
247 writing compound words or phrases in which the elements are joined without spaces, with each
248 element's initial letter capitalized within the compound). For example, the defined term is
249 *AddressSpace* instead of Address Space. This makes it easier to understand that there is a single
250 definition for *AddressSpace*, not separate definitions for Address and Space.

251 A list of all parts of the IEC 62541 series, published under the general title *OPC Unified Architecture*,
252 can be found on the IEC website.

253 The committee has decided that the contents of this document will remain unchanged until the stability
254 date indicated on the IEC website under "<http://webstore.iec.ch>" in the data related to the specific
255 document. At this date, the document will be

- 256 • reconfirmed,
- 257 • withdrawn,
- 258 • replaced by a revised edition, or
- 259 • amended.

260

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

261

OPC Unified Architecture

Part 11: Historical Access

262
263
264
265
266
267
268

1 Scope

270 This document is part of the OPC Unified Architecture standard series and defines the
271 *Information Model* associated with Historical Access (HA). It particularly includes additional and
272 complementary descriptions of the *NodeClasses* and *Attributes* needed for Historical Access,
273 additional standard *Properties*, and other information and behaviour.

274 The complete *AddressSpace* Model including all *NodeClasses* and *Attributes* is specified in IEC
275 62541-3. The predefined *Information Model* is defined in IEC 62541-5. The *Services* to detect
276 and access historical data and events, and description of the *ExtensibleParameter* types are
277 specified in IEC 62541-4.

278 This standard includes functionality to compute and return *Aggregates* like minimum, maximum,
279 average etc. The *Information Model* and the concrete working of *Aggregates* are defined in IEC
280 62541-13.

2 Normative references

282 The following documents, in whole or in part, are normatively referenced in this document and
283 are indispensable for its application. For dated references, only the edition cited applies. For
284 undated references, the latest edition of the referenced document (including any amendments
285 and errata) applies. (<https://standards.iteh.ai/>)

286 IEC 62541-1, OPC Unified Architecture - Part 1: Overview and Concepts

287 IEC 62541-3, OPC Unified Architecture - Part 3: Address Space Model

288 IEC 62541-4, OPC Unified Architecture - Part 4: Services

289 IEC 62541-5, OPC Unified Architecture - Part 5: Information Model

290 IEC 62541-7, OPC Unified Architecture - Part 7: Profiles

291 IEC 62541-8, OPC Unified Architecture - Part 8: Data Access

292 IEC 62541-13, OPC Unified Architecture - Part 13: Aggregates

3 Terms, definitions, and abbreviations

3.1 Terms and definitions

295 For the purposes of this document, the terms and definitions given in IEC 62541-1, IEC 62541-
296 3, IEC 62541-4, IEC 62541-13 and the following apply.

3.1.1

Annotation

299 metadata associated with an item at a given instance in time

300 Note 1 to entry: An *Annotation* is metadata that is associated with an item at a given instance in time.

3.1.2

BoundingValues

303 values associated with the starting and ending time

304 Note 1 to entry: *BoundingValues* are the values that are associated with the starting and ending time of a
305 *ProcessingInterval* specified when reading from the historian. *BoundingValues* may be required by *Clients* to
306 determine the starting and ending values when requesting *RawData* over a time range. If a *RawData* value exists at
307 the start or end point, it is considered the bounding value even though it is part of the data request. If no *RawData*

308 value exists at the start or end point, then the *Server* will determine the boundary value, which may require data from
309 a data point outside of the requested range. See 4.6 for details on using *BoundingValues*.

310 3.1.3

311 Historian

312 application storing time series data and/or time series events

313 3.1.4

314 HistoricalNode

315 *Object, Variable, Property or View* in the *AddressSpace* where a *Client* can access historical
316 data or *Events*

317 Note 1 to entry: A *HistoricalNode* is a term used in this document to represent any *Object, Variable, Property or*
318 *View* in the *AddressSpace* for which a *Client* may read and/or update historical data or *Events*. The terms
319 "*HistoricalNodes's* history" or "history of a *HistoricalNodes*" will refer to the time series data or *Events* stored for this
320 *HistoricalNode*. The term *HistoricalNode* refers to both *HistoricalDataNodes* and *HistoricalEventNodes*.

321 3.1.5

322 HistoricalDataNode

323 *Variable or Property* in the *AddressSpace* where a *Client* can access historical data

324 Note 1 to entry: A *HistoricalDataNode* represents any *Variable or Property* in the *AddressSpace* for which a *Client*
325 may read and/or update historical data. "*HistoricalDataNode* history" or "history of a *HistoricalDataNode*" refers to
326 the time series data stored for this *HistoricalNode*. Examples of such data are:

- 327 • device data (like temperature sensors)
- 328 • calculated data
- 329 • status information (open/closed, moving)
- 330 • dynamically changing system data (like stock quotes)
- 331 • diagnostic data

332 The term *HistoricalDataNodes* is used when referencing aspects of the standard that apply to accessing historical
333 data only.

334 3.1.6

335 HistoricalEventNode

336 *Object or View* in the *AddressSpace* for which a *Client* can access historical *Events*

337 Note 1 to entry: "*HistoricalEventNode's* history" or "history of a *HistoricalEventNode*" refers to the time series
338 *Events* stored in some historical system. Examples of such data are:

- 339 • *Notifications*
- 340 • system *Alarms*
- 341 • operator action *Events*
- 342 • system triggers (such as new orders to be processed)

343 The term *HistoricalEventNode* is used when referencing aspects of the standard that apply to accessing historical
344 *Events* only.

345 3.1.7

346 ModifiedValues

347 *HistoricalDataNode's* value that has been changed (or manually inserted or deleted) after it was
348 stored in the historian

349 Note 1 to entry: For some *Servers*, a lab data entry value is not a *modified value*, but if a user corrects a lab value,
350 the original value would be considered a *modified value*, and would be returned during a request for *ModifiedValues*.
351 Also manually inserting a value that was missed by a standard collection system may be considered a *modified value*.
352 Unless specified otherwise, all historical *Services* operate on the current, or most recent, value for the specified
353 *HistoricalDataNode* at the specified timestamp. Requests for *ModifiedValues* are used to access values that have
354 been superseded, deleted or inserted. It is up to a system to determine what is considered a *modified value*.
355 Whenever a *Server* has modified data available for an entry in the historical collection it shall set the *ExtraData* bit
356 in the *StatusCode*.

357 3.1.8

358 RawData

359 data that is stored within the historian for a *HistoricalDataNode*

360 Note 1 to entry: The data may be all data collected for the *DataValue* or it may be some subset of the data depending
361 on the historian and the storage rules invoked when the item's values were saved.

362 **3.1.9**
 363 **StartTime/EndTime**
 364 bounds of a history request which define the time domain

365 Note 1 to entry: For all requests, a value falling at the end time of the time domain is not included in the domain,
 366 so that requests made for successive, contiguous time domains will include every value in the historical collection
 367 exactly once.

368 **3.1.10**
 369 **TimeDomain**
 370 interval of time covered by a particular request, or response

371 Note 1 to entry: In general, if the start time is earlier than or the same as the end time, the time domain is considered
 372 to begin at the start time and end just before the end time; if the end time is earlier than the start time, the time
 373 domain still begins at the start time and ends just before the end time, with time "running backward" for the particular
 374 request and response. In both cases, any value which falls exactly at the end time of the *TimeDomain* is not included
 375 in the *TimeDomain*. See the examples in 4.6. *BoundingValues* effect the time domain as described in 4.6.

376 All timestamps which can legally be represented in a *UtcTime DataType* are valid timestamps, and the *Server* may
 377 not return an invalid argument result code due to the timestamp being outside of the range for which the *Server* has
 378 data. See IEC 62541-3 for a description of the range and granularity of this *DataType*. *Servers* are expected to handle
 379 out-of-bounds timestamps gracefully, and return the proper *StatusCodes* to the *Client*.

380 **3.1.11**
 381 **StructuredHistoryData**
 382 structured data stored in a history collection where parts of the structure are used to uniquely
 383 identify the data within the data collection

384 Note 1 to entry: Most historical data applications assume only one current value per timestamp. Therefore, the
 385 timestamp of the data is considered the unique identifier for that value. Some data or meta data such as *Annotations*
 386 may permit multiple values to exist at a single timestamp. In such cases the *Server* would use one or more parameters
 387 of the *StructuredHistoryData* entry to uniquely identify each element within the history collection. *Annotations* are
 388 examples of *StructuredHistoryData*.

389 **3.2 Abbreviated terms**

390 DA Data Access
 391 HA Historical Access
 392 HDA Historical Data Access
 393 UA Unified Architecture

394 **4 Concepts**

395 **4.1 General**

396 This document defines the handling of historical time series data and historical *Event* data in
 397 the OPC Unified Architecture (in a Historian). Included is the specification of the representation
 398 of historical data and *Events* in the *AddressSpace*.

399 **4.2 Data architecture**

400 A *Server* supporting Historical Access provides *Clients* with transparent access to different
 401 historical data and/or historical *Event* sources (e.g., process *Historians*, event *Historians*, etc.).

402 The historical data or *Events* may be in a proprietary data collection, database or a short-term
 403 buffer within memory. A *Server* supporting Historical Access will provide historical data and
 404 *Events* for all or a subset of the available *Variables*, *Objects*, *Properties* or *Views* within the
 405 *Server AddressSpace*.

406 Figure 1 illustrates how the *AddressSpace* of a UA *Server* might consist of a broad range of
 407 different historical data and/or historical *Event* sources.

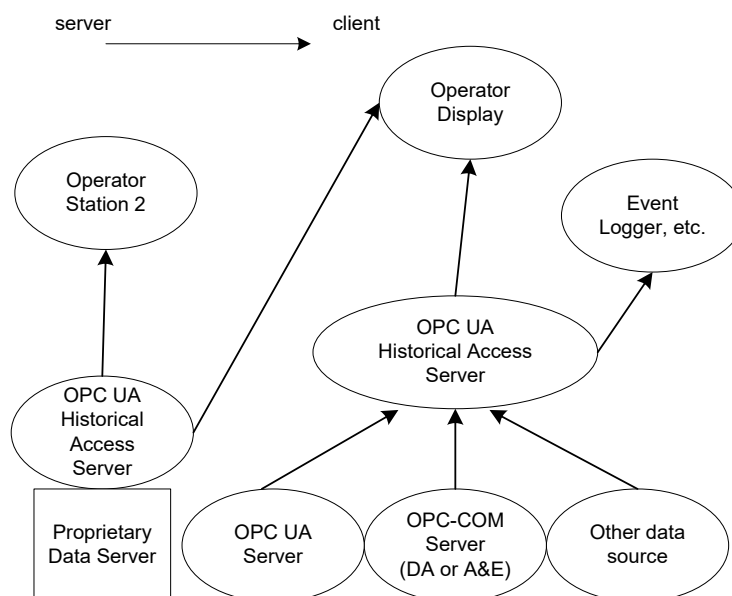


Figure 1 – Possible OPC UA Server supporting Historical Access

The *Historian* may be implemented as a standalone OPC UA Server that collects data from another OPC UA Server or another data source. The *Historian* may also just aggregate historical data from underlying *Historians*. The *Client* that references the OPC UA Server supporting Historical Access for historical data may be simple trending packages that desire values over a given time frame or they may be complex reports that require data in multiple formats.

There are general requirements for *Historians*, but *Historians* can vary in functionality. A consistent requirement for all *Historians* is that they store Historical data including a timestamp. All historical data should include status information for each value, but a *Historian* may compress this to only storing status information that indicates a problem (Bad status) and/or status change, instead of storing a status for every time series data item. The status of historical data can be complex. What is required is that the values returned as part of the timeseries raw data match the data that would have been observed if the *Value* was subscribed to at that point in time.

Historical *Events* are more complicated. In a stream of *Events* each Event can have a different list of fields. *EventTypes* are defined in a hierarchical manner, where each *EventType* inherits fields from its parent type and can add additional fields. Some of these additional fields may be mandatory and are required to understand or process the given *EventType*. A *Historian* that stores *Events*, shall be configurable to store all mandatory fields for any *EventTypes* that it historizes. If it receives for storage an *EventType* that it does not support all mandatory fields for, it may store it as one of its supertype *EventTypes* (one that it does support all mandatory fields for), but then it shall not claim that it supports historizing of that *EventType* (see 5.4.3). The *Historian* shall also provide information about the fields that are currently being historized (see 5.4.3).

4.3 Historians and interruption of data collection

When an *Historian* is collecting and storing data, the data collection may be interrupted. The interruption may have been for collecting the current values of data or for an event stream. The interruption may have been due to an interruption in the source of a value or an interruption of the forwarding of historical data from an underlying *Historian*. The interruption might also have been due to an action that stopped the collection of *HistoricalData* or historical *Events*. Some of these interruptions may recover with no loss of data, others may result in data gaps. The *Historian* shall report any gaps when a client is accessing the stored historical data with an error code of *Bad_DataLost*.