

Functional Analysis - Attachments

Proposal for HL7 Belgium WG Infrastructure & Security

Version management

Version	Date	Description	Author
1.0	21/09/2023	Initial analysis	Maxime Caucheteur (Smals - Vitalink)

1. Introduction

Not all useful information can be shared textually as part of standardized FHIR resources. Attachments like images, videos, PDF's can be useful to visualize or add details to the medical resource or Communication resources. Therefore, we need to find a way to store and serve back attachments that are sent along with FHIR resources.

2. Solutions in FHIR

The FHIR specification provides 1 data type and 3 resource types to handle this use case:

- Data Type:
 - o **Attachment:** *"This type is for containing or referencing attachments - additional data content defined in other formats. The most common use of this type is to include images or reports in some report format such as PDF. However, it can be used for any data that has a MIME type."*
- Resource Type:
 - o **Binary:** *"A resource that represents the data of a single raw artifact as digital content accessible in its native format. A Binary resource can contain any content, whether text, image, pdf, zip archive, etc."*
 - o **Media:** *"A photo, video, or audio recording acquired or used in healthcare. The actual content may be inline or provided by direct reference."*
 - o **DocumentReference:** *"A reference to a document of any kind for any purpose. Provides metadata about the document so that the document can be discovered and managed. The scope of a document is any serialized object with a mime-type, so includes formal patient centric documents (CDA), clinical notes, scanned paper, and non-patient specific documents like policy text."*

As the clients are interacting with us through a REST API, and as the FHIR specification recommends it, the file attachments will be sent to us as an embedded base64 string in the resource data itself.

3. Pros and Cons

Pros and Cons of Binary

PROS	CONS
<ul style="list-style-type: none">- Uniform way of using attachments/files: client always needs to use the resource type Binary. The Binary will be linked in the medical resource.- File is loosely bounded to the medical resource: an update of the medical resource does not require a new upload of the file (only if there is a new file of course).- A Binary resource can contain any content which allows for more flexibility.- Easier to plugin on the attachments coming in to offload it to external file storage/service/db ... → abstraction layer in the vault.- The FHIR Binary resource is designed to efficiently store and manage binary data within a FHIR server. It does not impose strict size limitations on the data it can represent, which means it can handle files of substantial size without causing performance issues or requiring complex workarounds.- FHIR servers and applications that support the Binary resource can implement chunking and streaming mechanisms to transmit, store, or retrieve large binary files.	<ul style="list-style-type: none">- Specific implementation will be required to cover the security on the Binary resource: a Binary can only be requested if the access is granted on the medical resource (securityContext).- We need to avoid that the vault will be used as file storage: the endpoint of Binary can only be used if there is a medical resource linked to it where the client has create/update access for.- Large binary files may need to be retained and archived for extended periods for compliance and legal reasons. Implementing long-term storage and archival solutions is essential.- When retrieving the medical resource the Binary should be retrieved in a separate call. At least for a GET by ID (read). For searches we could allow the include search to include the attachments in the search result, BUT we need to be careful with the size the Bundle will become.- The Binary resource does not provide specific semantics or metadata related to the content of the binary data.

Pros and Cons of Media

PROS	CONS
<ul style="list-style-type: none">- It provides a standardized way to capture rich metadata associated with media content. This metadata can include information about the patient, the context in which the media was created, and technical details.- The Media resource makes it easier to search for and retrieve media content within a FHIR	<ul style="list-style-type: none">- Narrow focus on media content. It may not be suitable for handling binary data that doesn't fit the typical definition of "media" such as pdf, zip, proprietary formats, ...- For simple use cases where basic binary data storage is sufficient, the Media resource may

<p>ecosystem. You can query and filter media records based on attributes like modality, patient, or date, facilitating efficient content retrieval.</p> <ul style="list-style-type: none"> - It aligns with FHIR's principles of standardization, making it easier for healthcare organizations to adopt and implement a consistent approach to managing media data. 	<p>introduce unnecessary complexity due to its rich metadata and specialized structure.</p> <ul style="list-style-type: none"> - Media resource can handle binary data, but it may not be as efficient as the FHIR Binary resource when dealing with large files.
---	--

Pros and Cons of DocumentReference

PROS	CONS
<ul style="list-style-type: none"> - DocumentReference resource is designed specifically for structured documents, such as clinical notes, reports, and other healthcare documents. It provides a standardized way to represent document content, including information about who/what/when, ... - It allows to link to external documents or binary resources using references, making it suitable for referencing media files or binary data associated with documents. - The DocumentReference resource can capture clinical context, making it valuable when documents need to be associated with specific patient encounters or episodes of care. 	<ul style="list-style-type: none"> - While DocumentReference can be used to reference media files or binary data associated with documents, it is not as specialized for media content as the Media resource.

4. Proposal

The choice between Binary, DocumentReference and Media depends on business needs. Seeing the properties of the three different resources types: Binary is the most simple and straight forward, DocumentReference is more focused on documents and Media on images/videos.

As we will need to support different file formats such as MP4, MOV, MKV, Dicom, g3fax, gif, jpeg, png, tiff or pdf, Binary seems more appropriate as Media and DocumentReference will not be flexible enough.

FHIR Specification: <https://hl7.org/fhir/R4/binary.html>

There are different options to send the attachments to the server: in a Transaction, as a contained resource or as a 'standalone' resource.

4.1 Transaction

Below are presented the different flows for each CRUD operation when the Binary is sent in a Transaction:

CRUD Operations Flow

Create:

If a Binary is created, it has to have a reference to the medical resource (in securityContext). The medical resource must also have a reference to the binary.

1. Medical resource & binary are created in a Transaction: We check that the resource has a reference to the Binary and vice-versa. Internal references can be used.
2. Medical resource already exists: If there is a create of a Binary, it has to be in a Transaction and contain an update of the medical resource adding a reference to the Binary.

All other scenarii will fail.

Read:

Basic behavior, the access will be checked based on the securityContext of the Binary.

Update:

We won't support the update of Binary.

Delete:

We need to preserve referential integrity.

1. The Binary & the medical resource are deleted in a Transaction: We check that the two resources are linked and delete both resources.
2. The Binary is deleted: Check that the transaction contains an update of the medical resource where we remove the reference to the binary.
3. The medical resource is deleted: If there is a Binary linked to this resource (need to check properties looking for a reference to a Binary), the resource cannot be deleted if there is no delete of that binary in the Transaction.

Search:

No Search supported on Binary in the FHIR specifications.

4.2 Contained resources

For Contained resources, there are two different options. We can either keep the contained resource inside the resource and store the resource as is in the database, or we extract the Binary from the contained resource and store it in a separate database as a resource with its own ID (The medical resource would have a reference to this created Binary).

Below are presented the different flows for each CRUD operation when the Binary is sent as a contained resource and is extracted:

CRUD Operations Flow – Contained resource is extracted

Create:

The Binary is added as a contained resource inside the FHIR resource. The server checks for the presence of a Binary in contained resource. The Binary is extracted and stored in a separate database. The medical resource holds a reference to the attachment.

Read:

When reading the medical resource, the response is smaller as we only have a reference to the Binary and if we want to get that Binary, we can do a GET on the reference.

Update:

In case of an update on a medical resource (without changing the Binary), the reference to the Binary stays.

In case of an update with a change to the Binary (new contained resource), the server will need to detect there is a new contained resource, extract it and create a new Binary and store it. Then, **what happens to the previous Binary ? Keep it as if we read the history of the resource, we might still need it ?**

Delete:

Delete of the Binary as it is created as a resource on its own. But if we want to ensure referential integrity, it can only be deleted if the medical resource referring to it, is deleted or has a new version without a reference to the Binary.

Search:

No Search supported on Binary in the FHIR specifications.

Search is done for the medical resource and then we can do a GET on the Binary referenced.

CRUD Operations Flow – Contained resource kept as is

Create:

The medical resource (with the contained resource) is stored as is in the database.

Read:

As the contained resource is in the medical resource, the server always returns the attachment with the resource (in case of large files, this can be a problem).

Update:

As the contained resource is in the medical resource, updates can be done in a classic way.

Delete:

Delete the medical resource and by this the attachment is deleted together with it.

Search:

Search is not supported on Binary as it does not have any searchable properties, the search happens on the medical resource and the attachment is retrieved in the response.

4.3 'Standalone' resource

By 'Standalone' resource is meant the idea of having an endpoint for Binary.

Below are presented the different flows for each CRUD operation when the Binary is sent on the Binary endpoint:

CRUD Operations Flow

Create:

The Binary is sent to the /Binary endpoint with a reference to the medical resource in the securityContext (on the condition the medical resource ID is created first). There is one thing the server can't enforce and that is that the resource does not reference back to the Binary. If a client is not updating the resource we have an orphan Binary resource. In case the client does send the update it is ok. The disadvantage of this setup is 3 separate requests need to take place for the creation of the resource and the attachment.

Read:

Read by ID. Behavior like other FHIR resources.

Update:

- We decide to support updates so the content can be changed (you don't need to change the reference in the medical resource if you upload a new version of a certain document).
- We decide not to support updates which enforces the clients to create a new Binary and update the reference to the Binary.

Delete:

If we delete the Binary, the medical resource it was linked to will reference a deleted resource. The referential integrity will be broken.

Search:

No Search supported on Binary in the FHIR specifications.

5. Examples

5.1 Creation of a Binary in a Transaction

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

5.2 Creation of a Binary in a contained resource

5.3 Creation of a Binary as a 'Standalone' resource

```
{
  "resourceType": "Communication",
  "contained": [
    { }, //Practitioner
    { } //PractitionerRole
  ]
  "subject": {
    "identifier": {
      "system": "https://www.ehealth.fgov.be/standards/fhir/core/NamingSystem/ssin",
      "value": "{{patient1_ssin}}"
    }
  }
  "recipient": [ { } ],
  "partOf": [ { } ]
}
```

First POST request – Communication:

Second POST request - Binary:

```
{
  "resourceType": "Binary",
  "securityContext": {
    "reference": "Communication/communication_id"
  },
  "contentType": "image/jpeg",
  "data": Base64 String of the content
}
```

PUT request – Communication:

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```