

Referential Integrity

Grahame Grieve, Health Intersections Pty Ltd



HL7 FHIR DevDays 2021, Virtual Edition, June 7–10, 2021 | @HL7 | @FirelyTeam | #fhirdevdays | www.devdays.com

ORGANIZED BY



PARTNER



Who am I?

- Grahame Grieve
- FHIR Product Director
- Health Interoperability Consultant



Learning Objectives

- What the FHIR Specification says about referential integrity
- How you can maintain referential integrity
- Practical Challenges maintaining it

Referential Integrity

- “a property of data such that all references are valid” (“broken links”)
- a property of a system such that it ensures that all references are to data that actually exists

```
{  
  "resourceType": "Observation",  
  "id": "f001",  
  "subject": {  
    "reference": "Patient/f001",  
    "display": "P. van de Heuvel"  
  }  
}
```

GET {base}/Patient/f001
--> 200 OK

Referential Integrity in the FHIR Specification

- Update: “Servers are permitted to reject update interactions because of integrity concerns or other business rules”
- Delete: “If the server refuses to delete a resource because of reasons specific to that resource, such as referential integrity, it should return the 409 Conflict status code. Note that the servers MAY choose to enforce business rules regarding deletion of resources that are being referenced by other resources, but they also MAY NOT do so.”

Referential Integrity in the FHIR Specification

- “Both client and server systems SHOULD clearly document how transaction integrity is handled, in the documentation in their CapabilityStatement.”

Why not enforce Referential Integrity?

- Enforcing Referential integrity is a natural and necessary feature of Enterprise Information Systems
 - Primary sources of information
- But:
 - Security / Access Control: client may not have access
 - Secondary sinks (CDR) may get partial records
 - Asynchronous processes – things may arrive out of order
 - Legacy Data
 - Broken business processes (paper)
 - Archive trails

Kinds of references in FHIR #1

```
"subject": {  
  "reference" : "Patient/1234"  
}  
"subject": {  
  "reference" : "http://someserver/some-path"  
}  
"subject": {  
  "reference" : "Patient/1234",  
  "type" : "Patient"  
}  
"subject": {  
  "reference" : "#pat"  
}
```


Kinds of references in FHIR #2

```
"subject": {  
  "identifier" : {  
    "system": "http://example.org/x",  
    "value" : "1234"  
  }  
}
```

```
"subject": {  
  "identifier" : {  
    "system": "http://example.org/x",  
    "value" : "1234"  
  },  
  "type" : "Patient"  
}
```

Reference Data types

- Reference(X)
- canonical(X)
- uri | url | oid | uuid

- Also, references may be found embedded inside
 - markdown
 - xhtml

Versioning and reference integrity

- FHIR is actually an insert only database at the version level
- Deleted resources still have a history
- Version specific references are not broken by deletion (maybe)
- But actually making this work – very complicated
 - I Haven't seen it
- Versioning + referential integrity – a permutational explosion of complexity

Maintaining Referential Integrity

- POST to Observation with unknown patient.
- 422 Business Rule Violation

```
{
  "resourceType": "OperationOutcome",
  "text": {
  },
  "issue": [
    {
      "severity": "error",
      "code": "not-found",
      "details": {
        "text": "The subject was not known"
      },
      "expression": [
        "Observation.subject.reference"
      ]
    }
  ]
}
```

Maintaining Referential integrity

- POST/PUT dependencies first
 - Id management issues
 - If they don't exist – consider conditional create
- What if a dependent commit fails?
 - Might be a problem
- Use transaction → make it all the server's problem

Maintaining Referential Integrity

- Delete to Patient/1234
- 422 Business Rule Violation

```
{
  "resourceType": "OperationOutcome",
  "text": {
  },
  "issue": [
    {
      "severity": "error",
      "code": "business-rule",
      "details": {
        "text": "Other records refer to this patient"
      },
      "expression": [
        "Patient"
      ]
    }
  ]
}
```

Maintaining Referential Integrity

- Server should enforce some referential integrity
- The exact rules depend on context
- Some things don't need integrity
 - AuditEvent / Provenance
 - DiagnosticReports with stated order, but no details
- Some things really should have integrity
 - All subject references
 - MedicationAdministration → Medication

Maintaining Referential Integrity

- History doesn't reflect transaction boundaries:

“Note, however, that the original transaction boundaries might not be represented in a history list, and a resource may occur more than once in a history list, so servers processing history bundles must have some strategy to manage this”

- This is a general issue with replication / subscription / secondary feeds

Circular References

- A particularly tricky issue is circular references:

```
{
  "resourceType": "Patient",
  "id": "A",
  "link": [
    {
      "other": {
        "reference": "Patient/B"
      }
      "type": "seealso"
    }
  ]
}
```

```
{
  "resourceType": "Patient",
  "id": "B",
  "link": [
    {
      "other": {
        "reference": "Patient/A"
      }
      "type": "seealso"
    }
  ]
}
```

Server Control

- HAPI has an interceptor for control
 - Controls referential deletes
 - More control needed though

Summary

- Its up the server how much referential integrity to enforce
 - Almost all servers need some, almost no servers need everything
- Clients and servers need to document their expectations
- The API allows servers to enforce integrity
 - Clear error messages are useful
- The corner cases are very messy

Q&A

- During DevDays, you can find / reach me here:
 - Via Whova App – Speaker’s Gallery
 - Email: grahameg@gmail.com
 - Twitter: [@GrahameGrieve](https://twitter.com/GrahameGrieve)

ORGANIZED BY



PARTNER

