Open Data XBRL Repository
filingindex.arelle.org
reportindex.arelle.org

17th Eurofiling Workshop, London

19 June 2013

Herm Fischer

Why another, Why open?

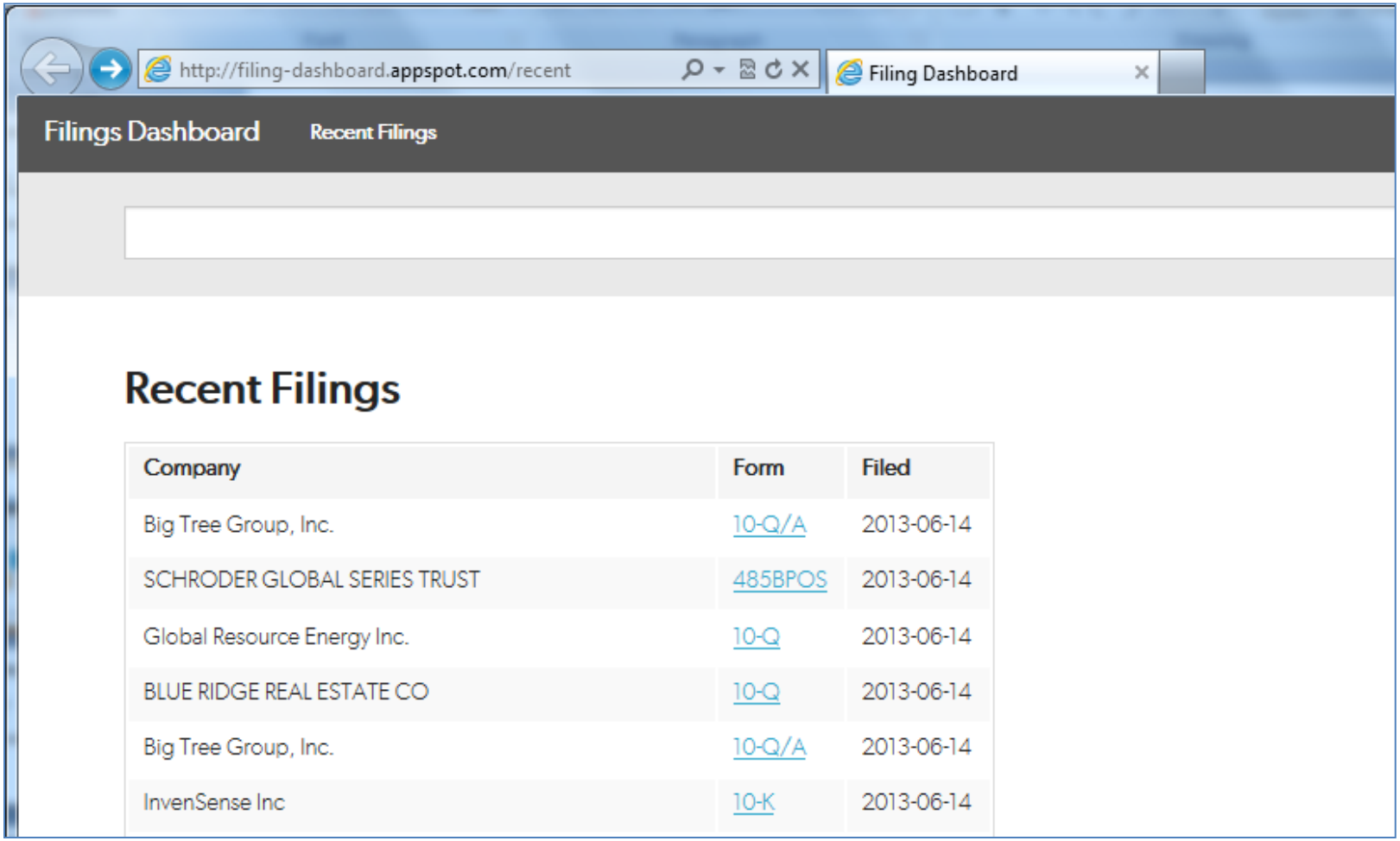
- SEC filings can be accessed:
  - On SEC website (what you see is what you get)
  - XBRL Cloud with quantitative assessment
    - Filings validated against “strict” criteria
    - No transparency, clicks for details -> request for $$
  - Other private (non-open) efforts under way
  - XBRL-US Challenge database
    - Postgres database of XBRL syntax
  - Clamor for transparency

Arelle user-based contribution

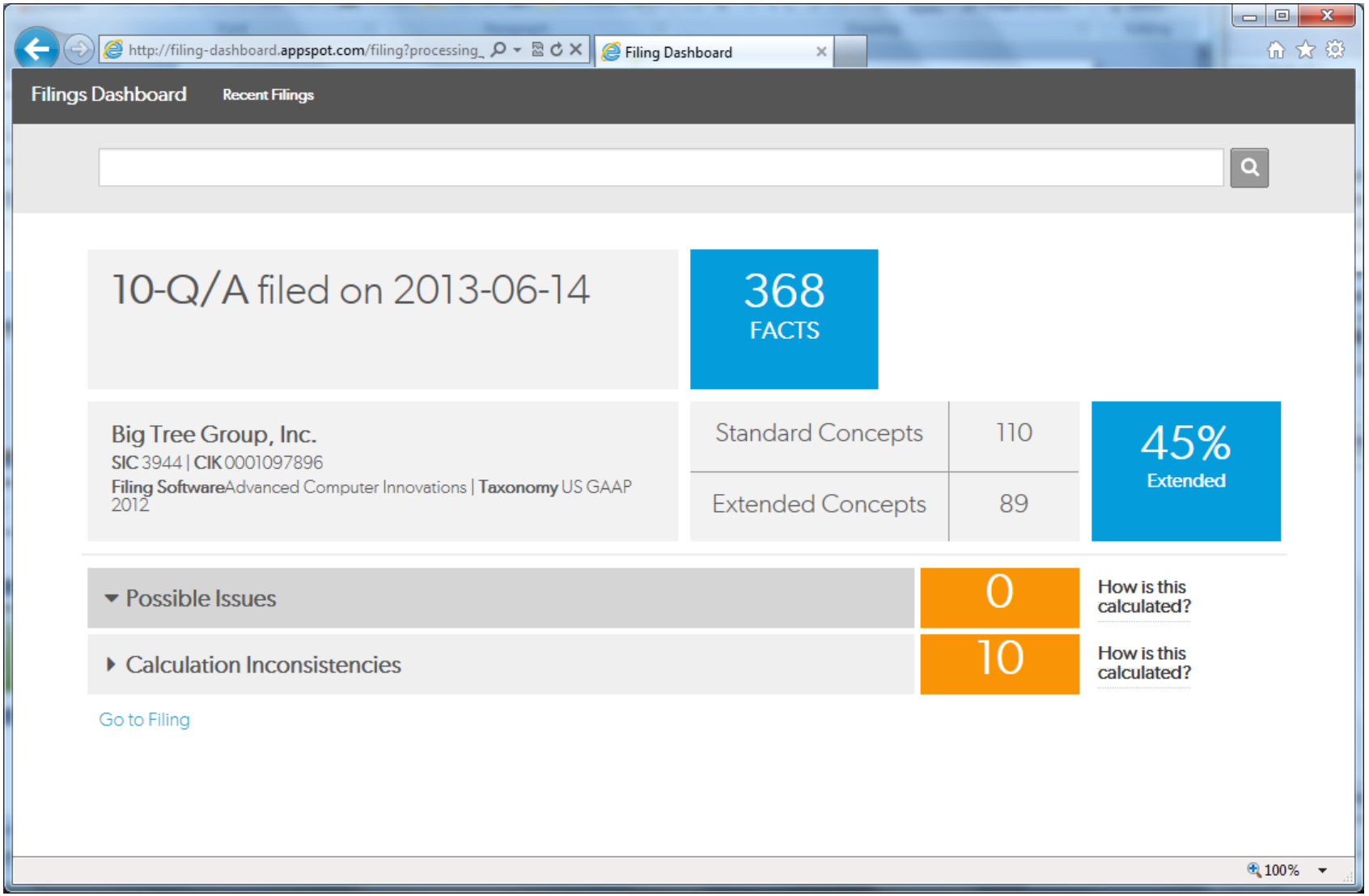
- Full validation of filings by Arelle
  - Criteria are transparent
  - Excessively picky criteria pre-filtered out
- Goal of single and multi-instance querying
  - Quality assessments
  - Data, value, and KPI assessments
  - (like the goal of XBRL-US challenge contest)

Initial contribution step

- Post all validation results by Arelle



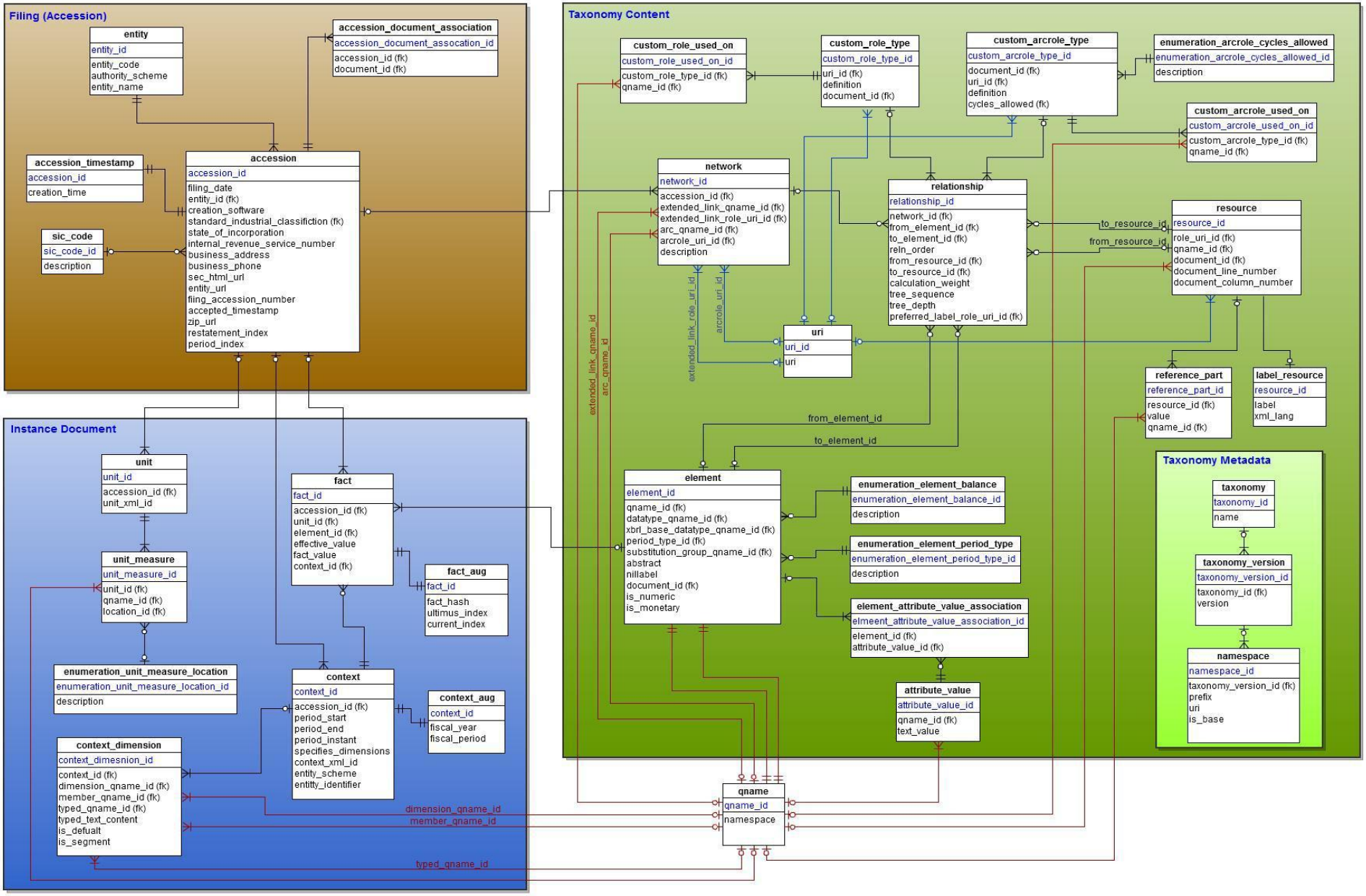
An SEC Filing



Graph Model Research Stage

- Arelle xbrlDB plugin
  - Load XBRL-US Postgres Public Database
  - Load Abstract Model Graph Database
- Arelle Viewer (Browser-based App)
  - Replicate desktop views from graph database
  - Develop cross-instance queries
- Next
  - KPI and cross-instance queries
  - Rendering models and browser-based viewing

XBRL-US Postgres Public Database



User feedback

- SQL queries are painful and slow
- XBRL modeled at the syntax level
- SEC filing focus
- Impractical to do:
  - KPI
  - Cross-instance queries
  - Rendering LB views

Graph Database

- Open-source social network database technology
  - Cassandra (facebook)
  - Titan-hbase (twitter, gitHub)
  - HADOOP (map-reduce)
- New software
  - XPath unsuitable
  - Gremlin hosted on Groovy (tinkerpop)
    - Pipeline & closures architecture
    - Python features on Java VM
  - Criticality of map-reduce

Timing and Sizing

- Timing
  - Load time less than desktop GUI (few seconds)
  - Render time good with in-memory data
- Sizing
  - Concerned here
  - 165GB+ for XBRL Public Postgres DB
  - Worry terabytes per year for SEC filings
    - Rearchitect, refactor, compress, etc.

2008 Japan project for Comparison

Filing	Form	Added Vert.	Added Edge	Cumul. Vert	Cumul. Edge	2008 NTT	2013 Graph DB
E00056-2008	EDInet	17,280	47,978	17,280	47,978	714KB	14.8MB
E02513-2008	EDInet	18,358	46,922	35,641	94,902	766KB	31.4MB
E02529-2008	EDInet	18,256	47,152	53,897	142,054	726KB	48.4MB

2008 more compact because…

- Custom interning of strings
- Only pres LB (vs. all LBs)
- Only standard & presented labels (vs. all)
- C# *binary* serialization
  - List of nested lists in pres LB nesting (≈ JSON)
  - No graph modeling

First Graph DB results

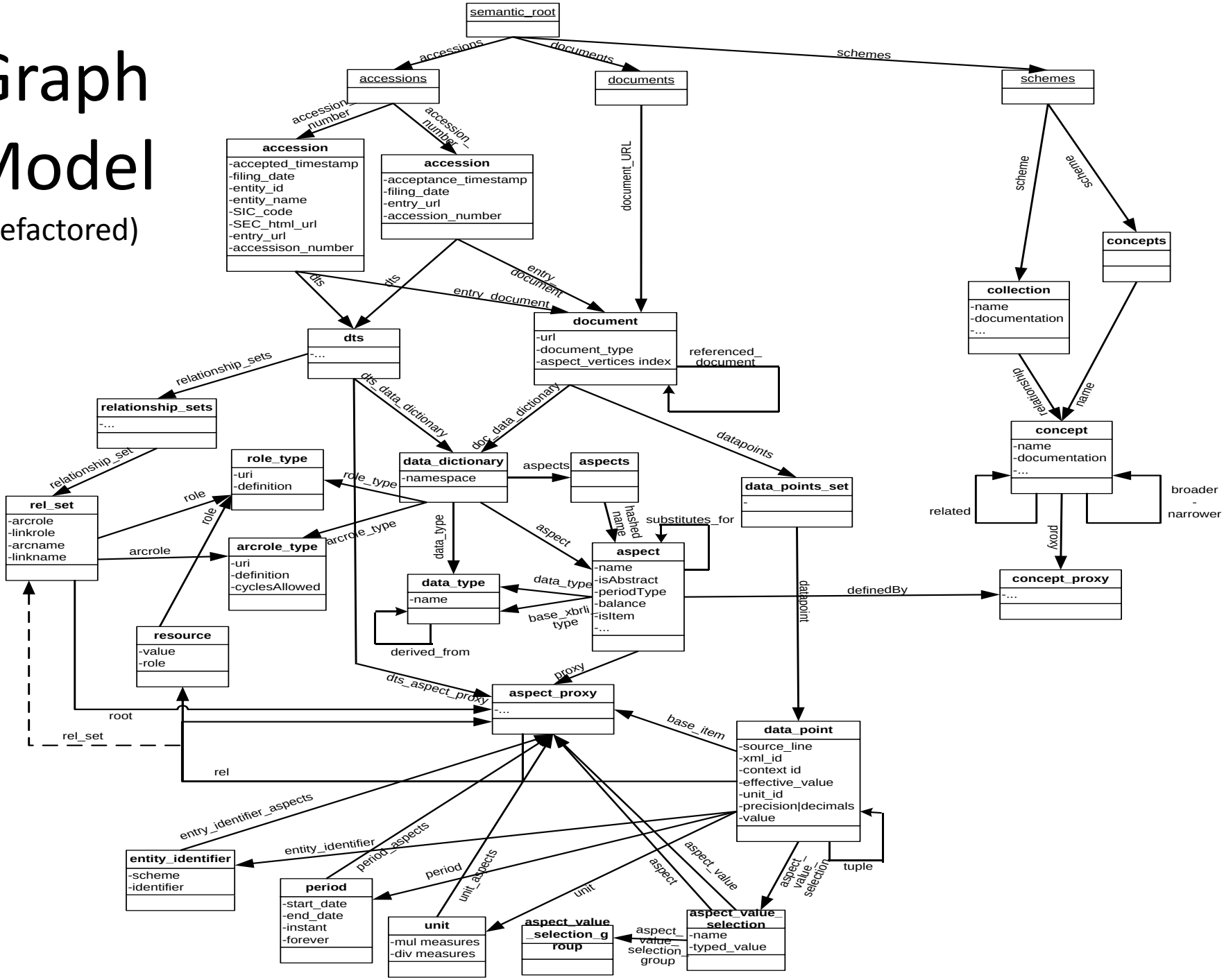
Filing	Form	Added Vert.	Added Edge	Cumul. Vert.	Cumul. Edge	Titan Graph DB on disk	Tinkerpop in memory
startup						0MB	221.9MB
krft-20130330	10-Q	33,705	98,055	33,705	98,055	47.4MB	399.5MB
jci-20130330	10-Q	15.334	34,896	49,042	132,953	65.9MB	403.9MB
el-20130330	10-Q	14,166	38,619	63,208	171,572	81.7MB	464.1MB
txn-20130503	8-K	12,688	38,292	75,896	209,864	100MB	489.8MB
tpc-20130202 (Company.II)	10-K	6,570	21,499	82,466	231,363	109MB	528.4MB
tpc-20130202 (Company.I)	10-K	12,955	46,833	95,421	278,196	132MB	552.8MB
Extrapolation guesstimate:							
…per filing		12,500	36,000			17MB	30.5MB
…15k x 5 per year = 75k/yr				937,500,000	2,700,000,000	1,275GB	2,288GB

After refactoring

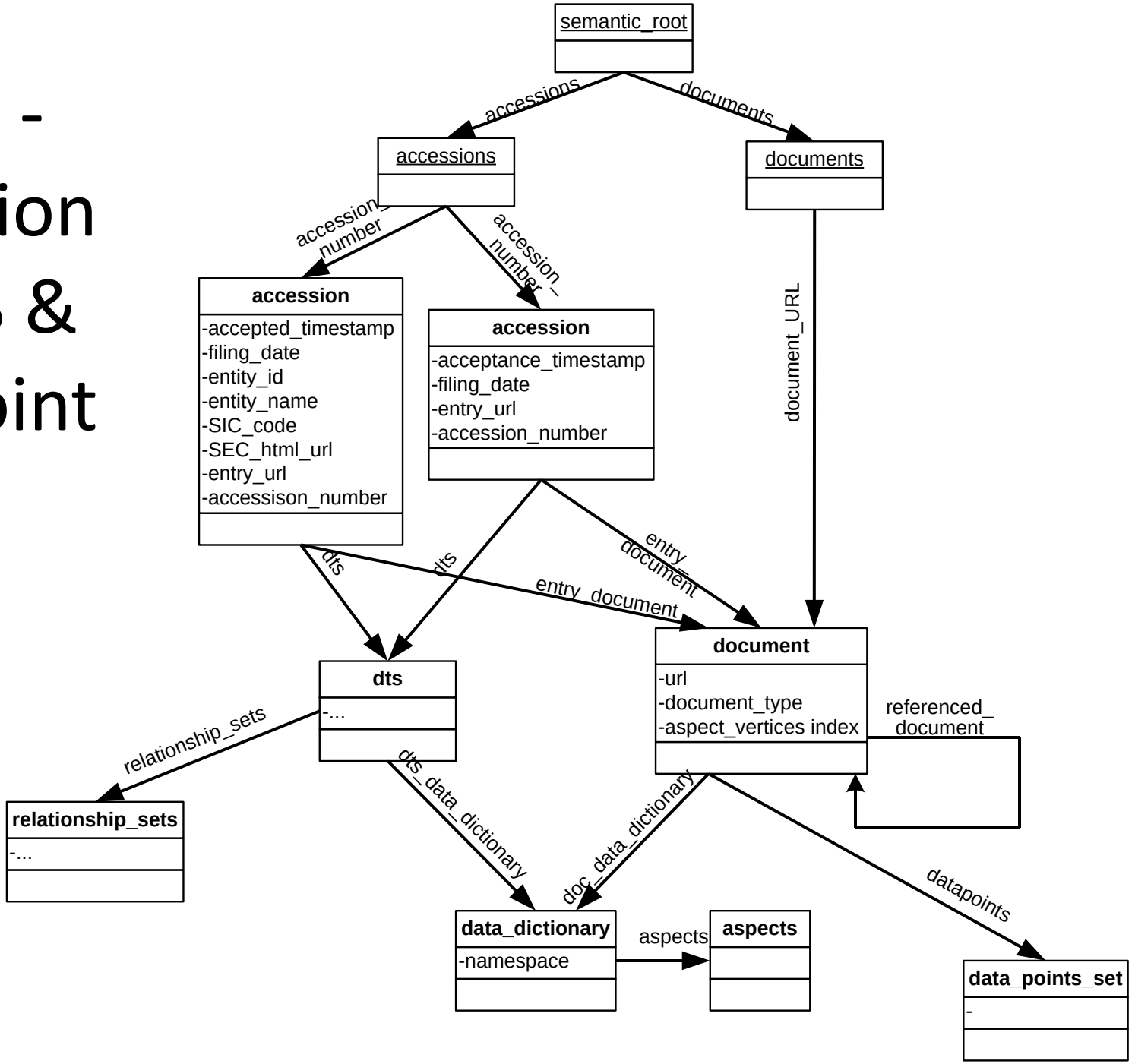
Filing	Form	Added Vert.	Added Edge	Cumul. Vert.	Cumul. Edge	Cassandra on disk
startup						0MB
krft-20130330	10-Q	30,094	88,854	30,094	88,854	36MB (9 min)
jci-20130330	10-Q	11,482	25,186	41,579	114,042	43MB(1.85m)
el-20130330	10-Q	6,341	18,408	47,920	132,450	51MB(.6m)
txn-20130503	8-K	6,333	21,939	54,253	154,389	65M(.4m)
Extrapolation guesstimate:						
…per extension		8,052	21,844			11MB
…15k x 5 per year = 75k/yr						825GB

Graph Model

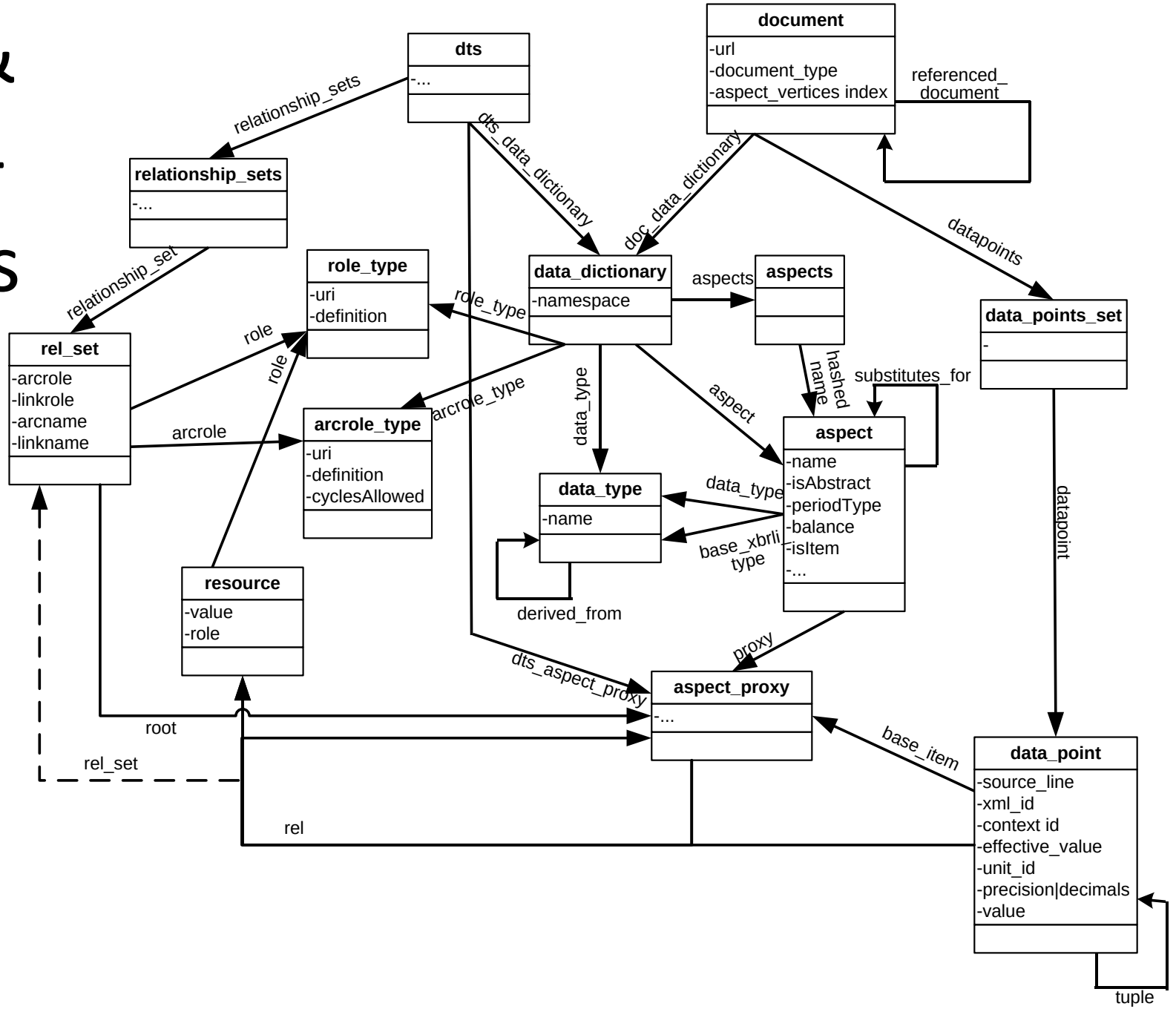
(refactored)



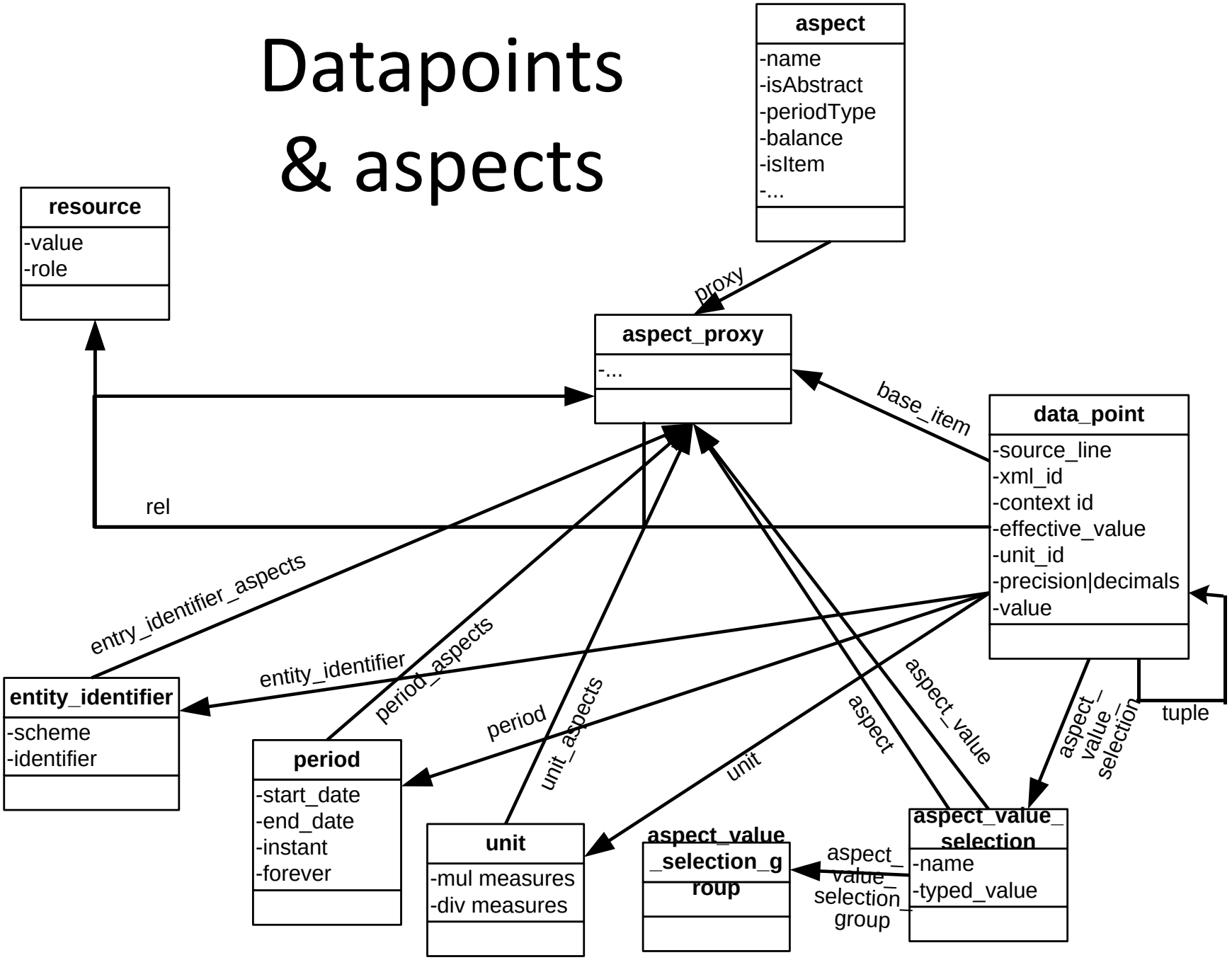
Filing - Accession to DTS & DataPoint



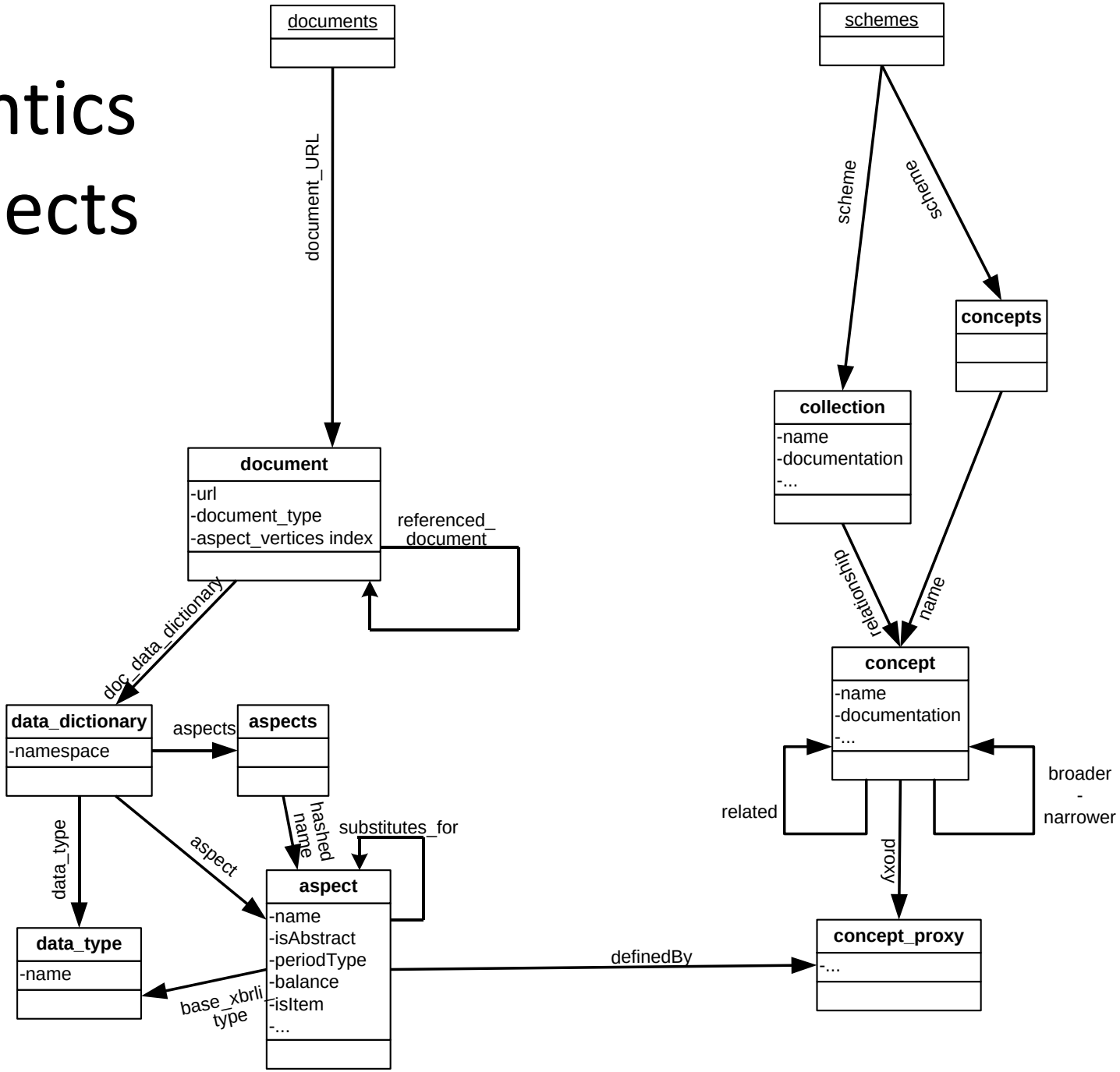
DTS & Data-points



Datapoints & aspects

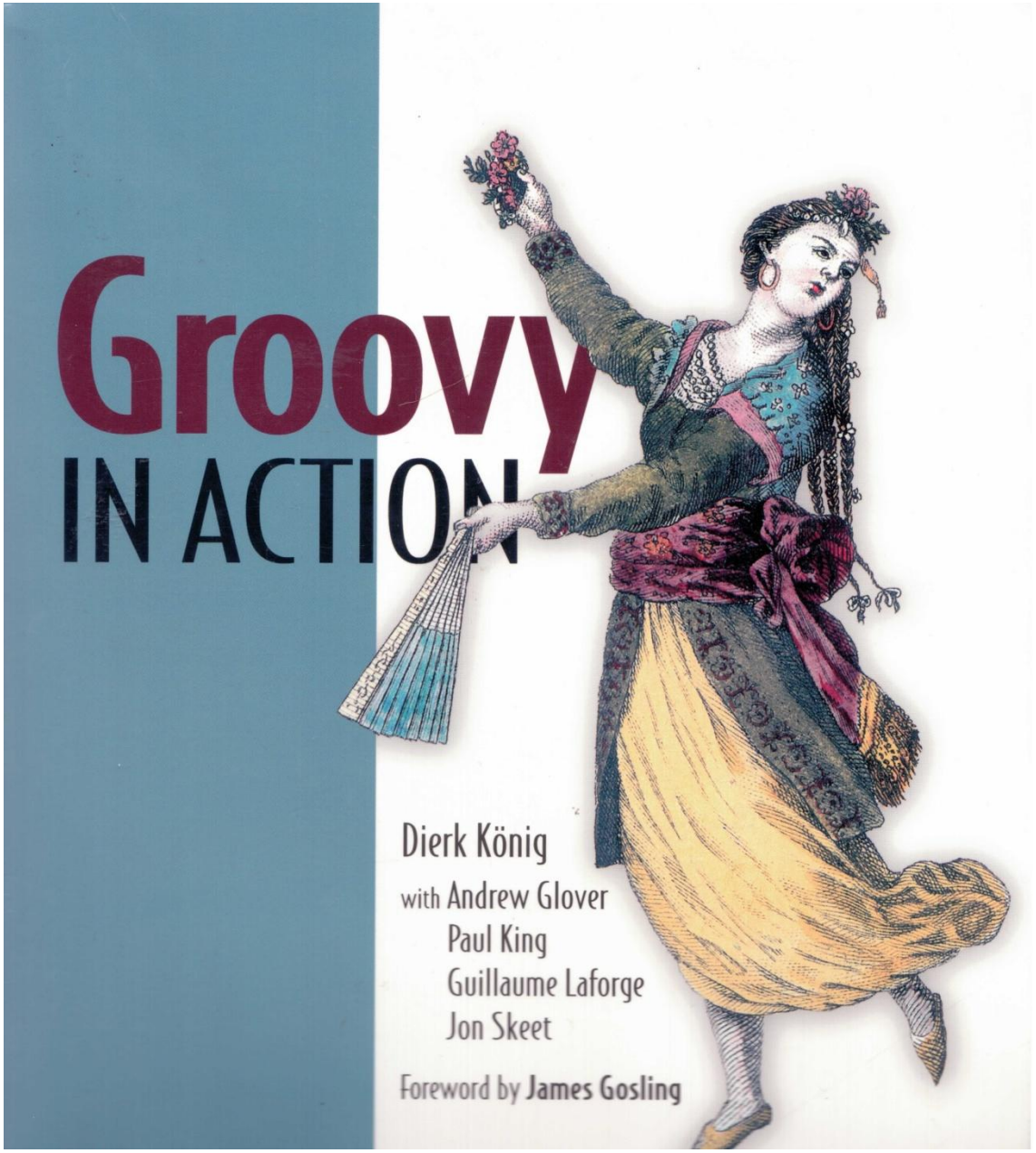


Semantics of Aspects

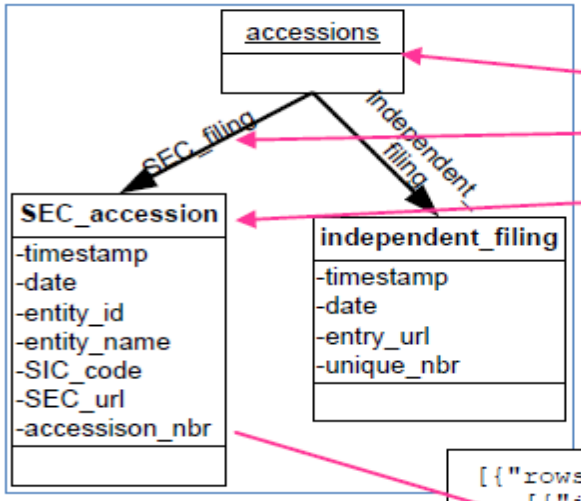


Graph traversal, set based, closures –

- query
- view (edit?)
- validation



Graph to Filing Accession

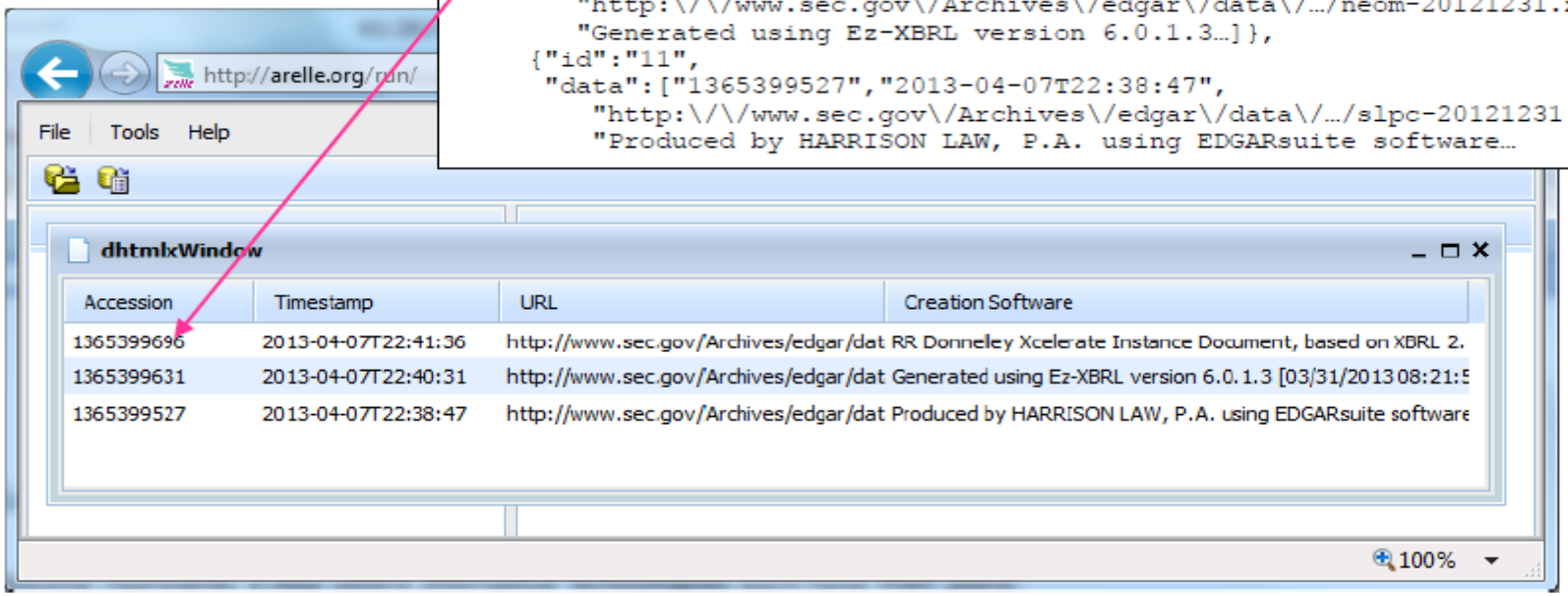


Gremlin graph query

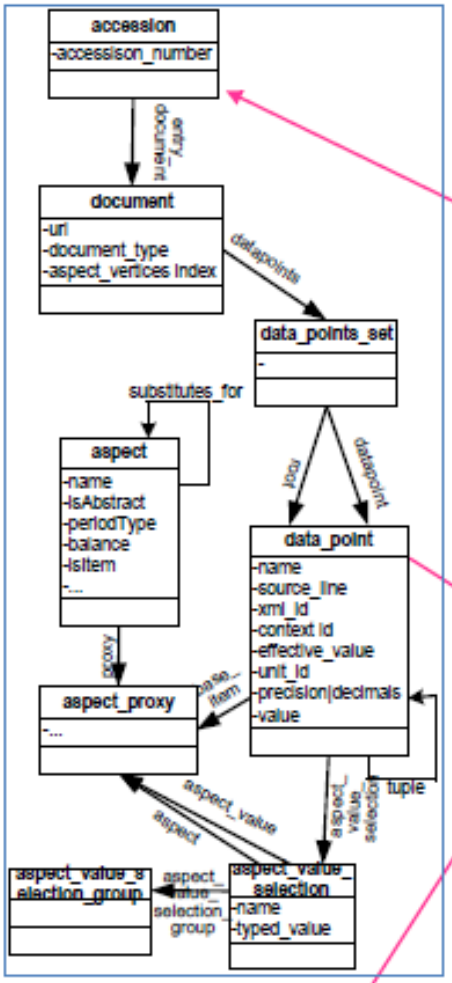
```
['rows':g.V('class','accessions').out
 .as('data').as('id').select{it.id}
 {[it.filing_accession_number,
   it.accepted_timestamp,
   it.entry_url,
   it.creation_software?:'']}]
```

Json results


```
[{"rows":
   [{"id":"132005",
    "data":["1365399696","2013-04-07T22:41:36",
      "http:\/\/www.sec.gov\/Archives\/edgar\/data…/penx-20130228.xml",
      " RR Donnelley Xcelerate Instance Document …  6.20.1 },
   {"id":"88311",
    "data":["1365399631","2013-04-07T22:40:31",
      "http:\/\/www.sec.gov\/Archives\/edgar\/data\/…/neom-20121231.xml",
      "Generated using Ez-XBRL version 6.0.1.3…]},
   {"id":"11",
    "data":["1365399527","2013-04-07T22:38:47",
      "http:\/\/www.sec.gov\/Archives\/edgar\/data\/…/slpc-20121231.xml",
      "Produced by HARRISON LAW, P.A. using EDGARsuite software…
```

Graph to Data Points



Gremlin graph query

```
['rows':g.v(132005).out('entry_document')
  .out('data_points').out('data_point')
  .order{it.a.source_line <=> it.b.source_line}
  .as('data').as('id').select{it.id}{
  [it.name,
   it.source_line,
   it.context?:'',
   it.unit?:'',
   it.effective_value?:it.value?:'']}]
```

Json results

```
{"rows":
 [{"id":"134195",
   "data":["dei:EntityCommonStockSharesOutstanding",9,
           "eol_PE9760----1310-Q0002_STD_0_20130402_0",
           "shares",
           "12,418,428"]},
  {"id":"134197",
   "data":["us-gaap:BusinessAcquisitionCostOfAcquired…",
           10,
           "eol_PE9760----1310-Q0002_STD_0_20120131…",
           "iso4217_USD",
           "8,500,000"]},
```