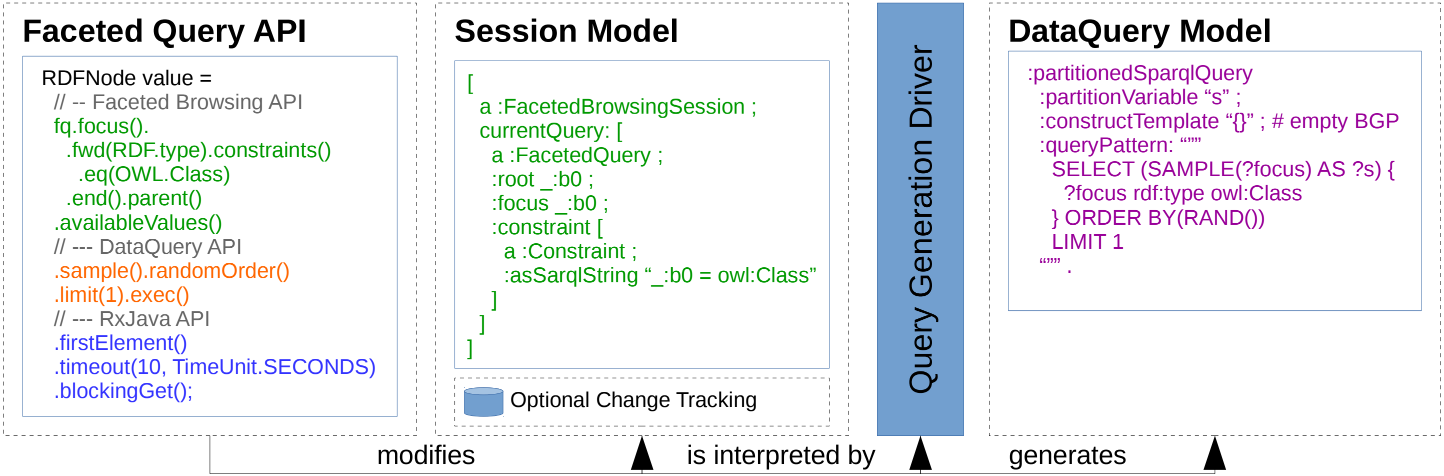

Faceted Query API

```
RDFNode value =
  // -- Faceted Browsing API
  fq.focus().
    .fwd(RDF.type).constraints()
      .eq(OWL.Class)
    .end().parent()
  .availableValues()
  // --- DataQuery API
  .sample().randomOrder()
  .limit(1).exec()
  // --- RxJava API
  .firstElement()
  .timeout(10, TimeUnit.SECONDS)
  .blockingGet();
```

Session Model

```
[
  a :FacetedBrowsingSession ;
  currentQuery: [
    a :FacetedQuery ;
    :root _:b0 ;
    :focus _:b0 ;
    :constraint [
      a :Constraint ;
      :asSarqlString “_:b0 = owl:Class”
    ]
  ]
]
```

Optional Change Tracking

DataQuery Model

```
:partitionedSparqlQuery
  :partitionVariable “s” ;
  :constructTemplate “{}” ; # empty BGP
  :queryPattern: “””
    SELECT (SAMPLE(?focus) AS ?s) {
      ?focus rdf:type owl:Class
    } ORDER BY(RAND())
    LIMIT 1
  “”” .
```