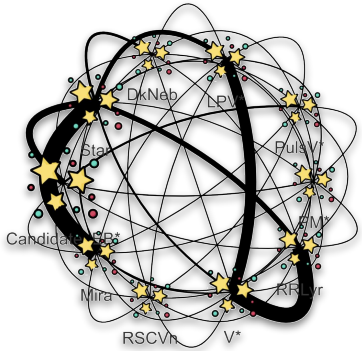
Visualisation de Bases de Données Des Graphs
Un Service Web Générique

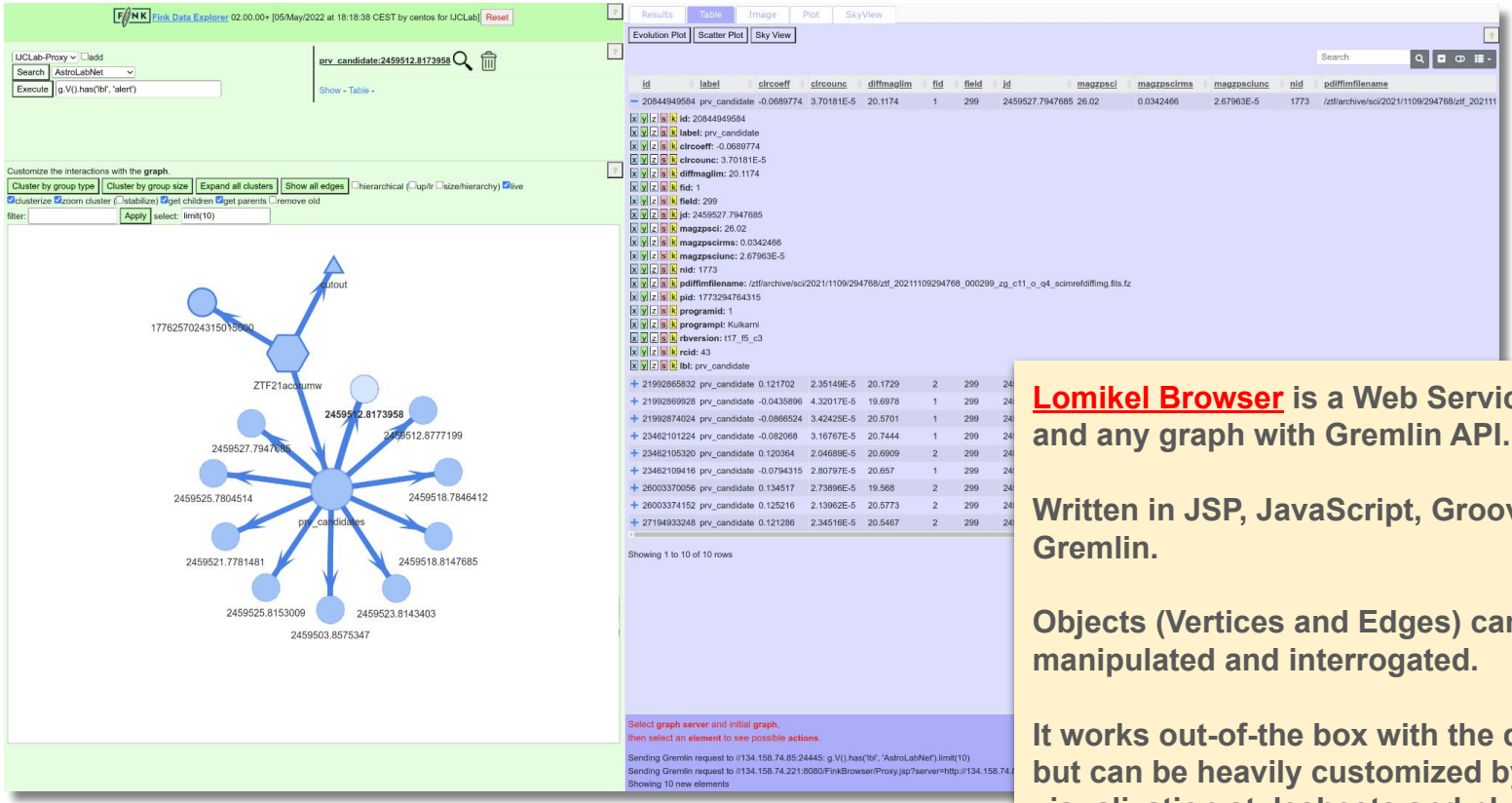
Actuellement, il n'existe pas d'outils publics disponibles pour la visualisation générique des bases de données des graphes. Les solutions existantes sont souvent propriétaires, coûteuses ou spécifiques à des domaines d'application particuliers. Cette présentation introduit un service web interactif, dynamique et hautement configurable, conçu pour la visualisation du contenu des bases de données des graphes, des requêtes et des résultats d'analyses. Les capacités de cet outil seront démontrées en utilisant les données graphiques du broker Fink de l'Observatoire Vera Rubin.



Julius Hrivnac, IJCLab

JI 2024
Saint-Nicolas-la-Chapelle
23-26/Sep/2024

Lomikel Browser



Lomikel Browser is a Web Service to visualize and any graph with Gremlin API.

Written in JSP, JavaScript, Groovy and Gremlin.

Objects (Vertices and Edges) can be manipulated and interrogated.

It works out-of-the box with the default style, but can be heavily customized by visualization stylesheets and plugins.

Lomikel Browser

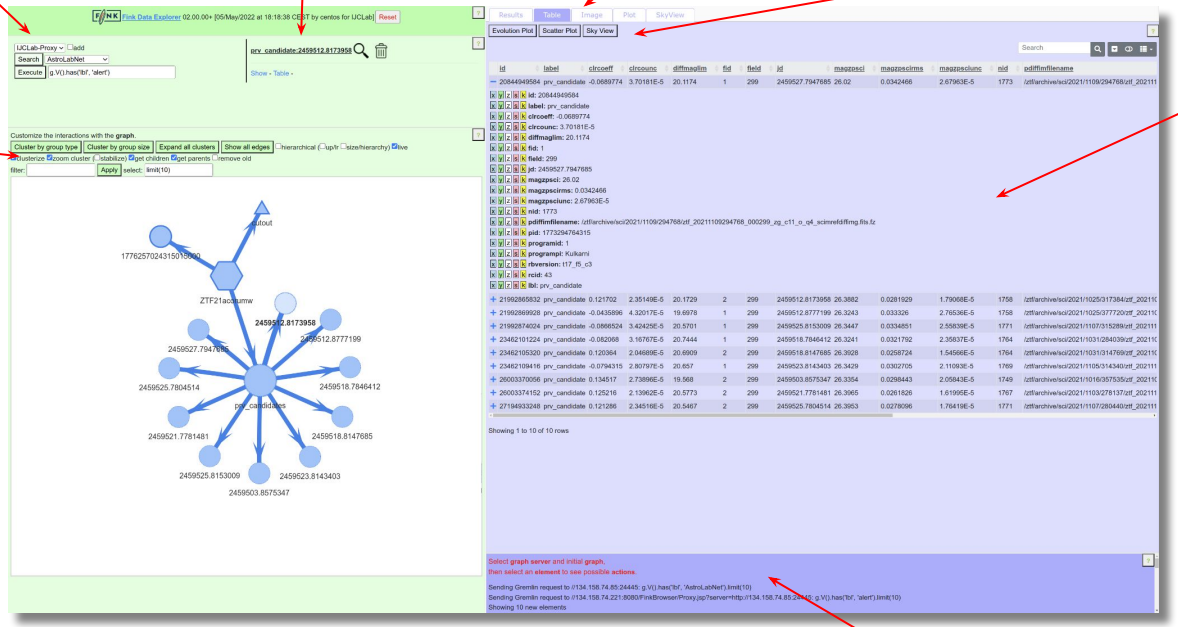
Graph context-sensitive operations
And introspection

Other ways of graph analyses
(as plugins)

Selection of the Graph server
And initial request

Other plugins can be called
For more detailed analyses

Global operations
on Graph



Graph shown as
an interactive table

Interactive view of a Graph
(customisable by a stylesheet)

Customisation

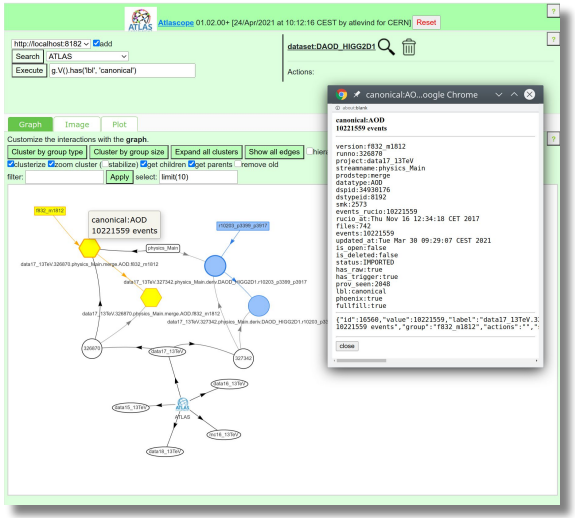
Stylesheet

```
stylesheet.nodes.datalink = {
  properties:{gremlin:"valueMap('name', 'technology').toList()[0]"},
  graphics: {
    label:"name",
    title:"name",
    subtitle:"technology",
    group:" ",
    shape:"dot",
    image:"",
    borderRadius:"0",
    borderWidth:"1",
    borderDashes:[1,0],
    value:"0"
    },
  actions:[
    {name:"Link",    url:{gremlin:"id().next().toString().replaceFirst(\"^\", \"DataLink.jsp?id=\")"    }, target:"result"  },
    {name:"Fits",    url:{gremlin:"id().next().toString().replaceFirst(\"^\", \"DataLinkFits.jsp?id=\")"}, target:"result"  },
    {name:"Show",    url:{gremlin:"id().next().toString().replaceFirst(\"^\", \"Node.jsp?id=\")"        }, target:"result"  },
    {name:"Table",   url:{gremlin:"id().next().toString().replaceFirst(\"^\", \"Nodes.jsp?id=\")"       }, target:"table"   }
    ]
  }
stylesheet.nodes.alert = {
  properties:{gremlin:"valu
  graphics: {
    label:"lbl",
    title:"lbl",
    subtitle:" ",
    group:{gremlin:"values(
    shape:"hexagon",
    image:"",
    borderRadius:"0",
    borderWidth:"2",
    borderDashes:[1,1],
    value:{gremlin:"out().o
    },
  actions:[
    {name:"Show",    url:{g
    {name:"Table",   url:{g
    ]
  }
```

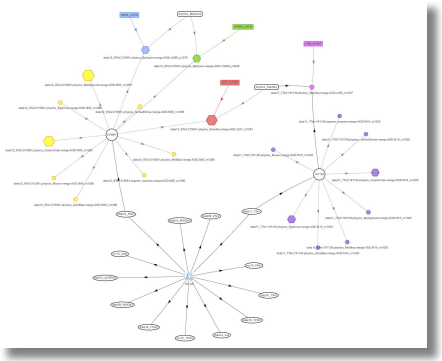
Stylesheet is a JSON document, describing possible Vertices and Edges, containing scripts in Gremlin or JavaScript.

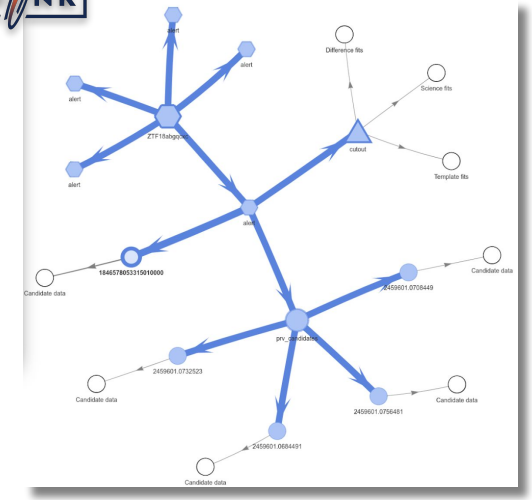
Plugins can specify
- how Vertices/Edges are shown
- their context-sensitive operations as either call to internal plugins or external services.

Many **standard plugins** exist:
- correlations/overlaps (i.e. properties of Edges between Vertices) as table and Venn diagrams
- scatterplots for Vertex/Edge properties
- time dependence of Vertex/Edge properties (if time property defined)
- visualization of embedded data (pictures,...)
- navigation to connected databases (SQL, NoSQL or Graph)

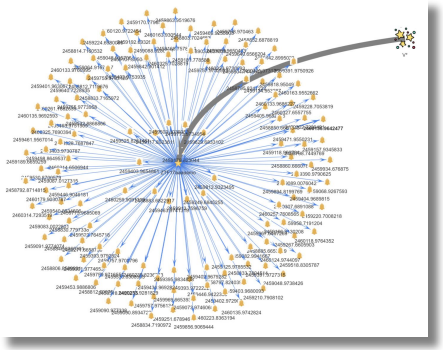


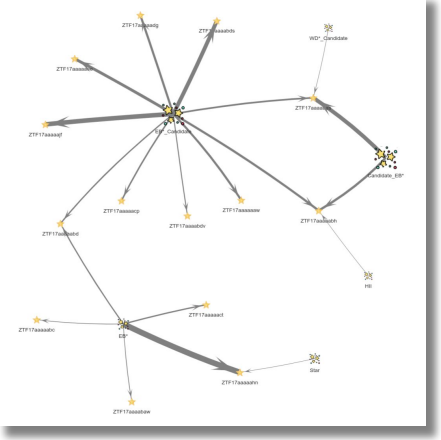
Examples



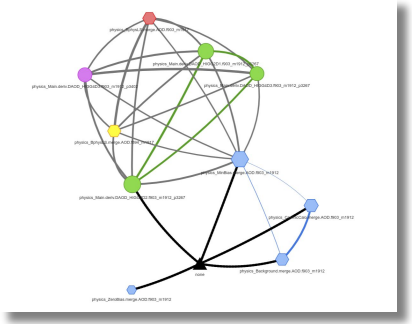


Vertex introspection

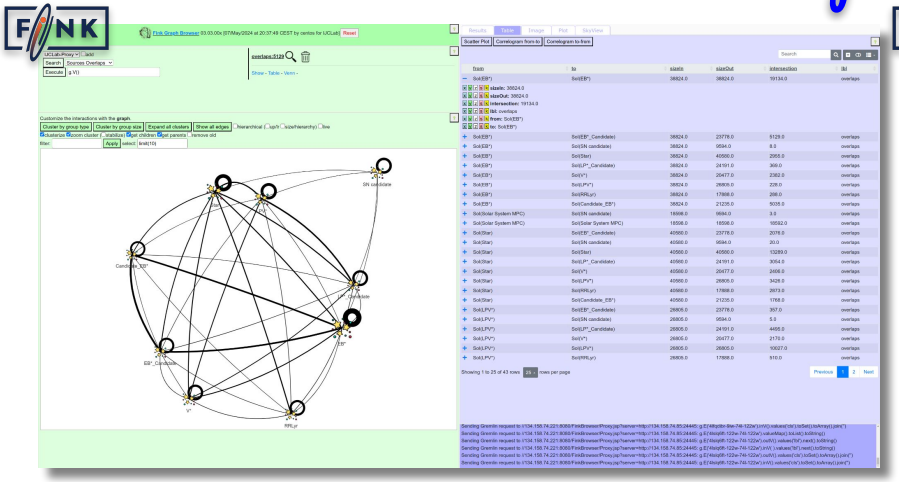




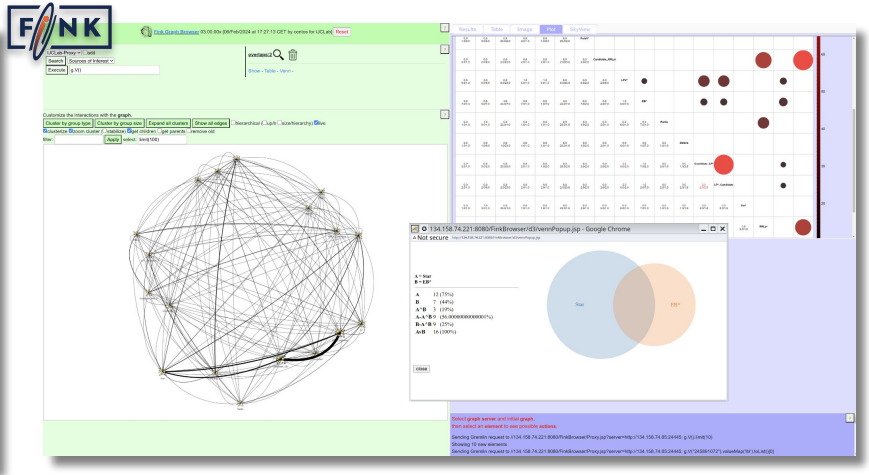
Graph with relations to data in external database (HBase in this case)



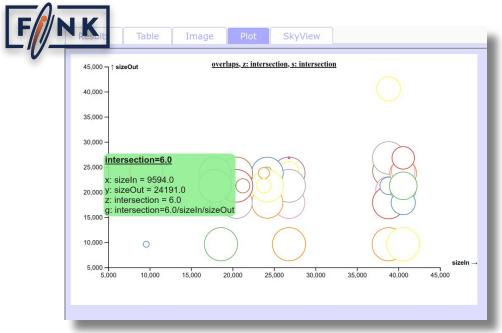
Overlaps/Correlations



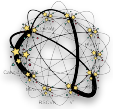
Vertex: Group of Fink Alerts of certain type
Edge: Number of common Alerts to both Vertices
(=overlap/correlation)



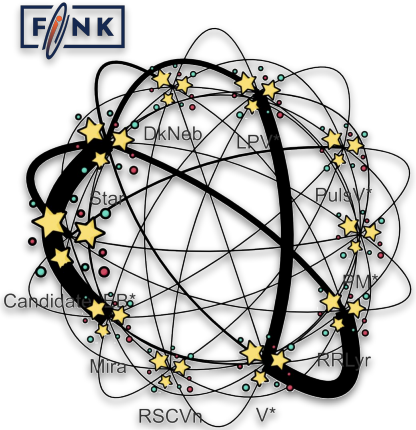
Overlaps shown as a matrix
and a Venn diagram (for one overlap)



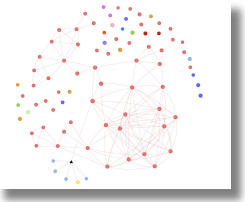
Overlaps shown
as a scatterplot

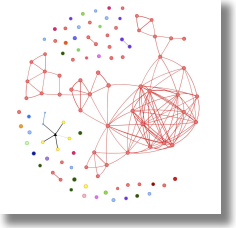


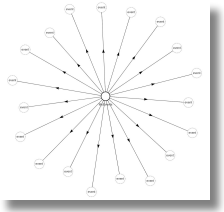
Visualisation de Bases de Données Des Graphs
Un Service Web Générique

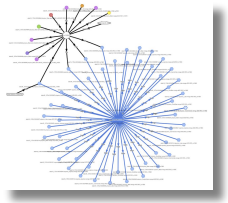


Home: https://cern.ch/hrivnac/Activities/Packages/Lomikel
Git: https://github.com/hrivnac/Lomikel.git









Future development:
- **light version to allow visualization of vast amount of data (but with less interactivity)**
- **standalone application (not a Web Service)**
- **possibility to modify and create objects in the database**