

Secure Graph

Joe Ferner

Altamira

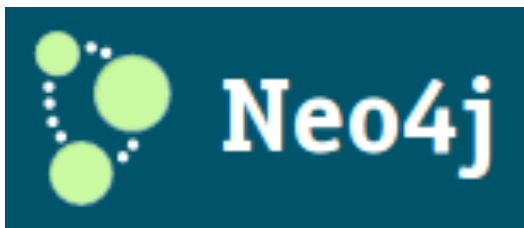
About Me

- Lead software developer for Lumify
- Author of Secure Graph

What is Secure Graph?

Secure Graph is a graph database API that allows security constraints to be applied to vertices, edges, and properties.

What about existing graph DBs?



Neo4j and Titan

- Security not built into the API
- Hard to integrate security

Secure Graph API - Insert

```
Visibility visA = new Visibility("a");
Visibility visAandB = new Visibility("a&b");
Authorizations authA = new AccumuloAuthorizations("a");

graph.prepareVertex("v1", visA, authA)
    .setProperty("name", "Joe Ferner", visA)
    .setProperty("dateOfBirth", "1977/10/30", visAandB)
    .save();
```

Secure Graph API - Query

```
Authorizations authA = new AccumuloAuthorizations("a");  
Vertex v1 = graph.getVertex("v1", authA);  
// dataOfBirth not returned
```

```
Authorizations authAandB = new AccumuloAuthorizations("a", "b");  
Vertex v1 = graph.getVertex("v1", authAandB);  
// dataOfBirth returned
```

```
Iterable<Edge> edges = v1.getEdges(Direction.BOTH, authA)
```

API Demo

Blueprints API

```
AccumuloGraph graph = createGraph();
```

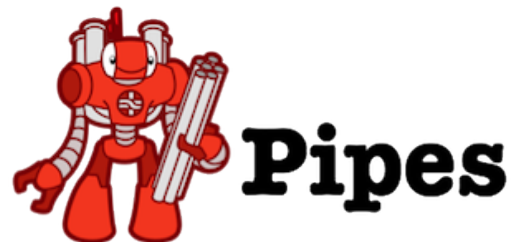
```
VisibilityProvider vp = new DefaultVisibilityProvider();
```

```
config.put("authorizationsProvider.auths", "a,b");
```

```
AuthorizationsProvider ap = new AccumuloAuthorizationsProvider(config);
```

```
com.tinkerpop.blueprints.Graph blueprintsGraph  
    = new AccumuloSecureGraphBlueprintsGraph(graph, vp, ap);
```

Blueprints API



Blueprints Demo

Other Features

- Multi-valued properties
- Property Metadata
- Streaming Values
- Exact match indexing

Current Storage Providers



elasticsearch.

Accumulo

Distributed key/value store similar to HBase with security.

Row ID	Column Family	Column Qualifier	Column Visibility	Timestamp	Value
row1	props	name	a		Charlie
row2	props	name			Joe
row2	props	name	a		Joe Ferner

Secure Graph Tables

- Vertices
- Edges
- Data

Vertices Table Format

Row ID	Column Family	Column Qualifier	Value	Description
V[vertex id]	V	-	-	Vertex existence and visibility
V[vertex id]	EOUT	[edge id]	[label]	Out edges
V[vertex id]	EIN	[edge id]	[label]	In edges
V[vertex id]	VOUT	[vertex id]	[edge label]	Out vertex
V[vertex id]	VIN	[vertex id]	[edge label]	In vertex
V[vertex id]	PROP	[prop name + key]	[prop value]	Property
V[vertex id]	PROPMETA	[prop name + key]	[prop meta]	Property Metadata

Edges Table Format

Row ID	Column Family	Column Qualifier	Value	Description
E[edge id]	E	-	-	Edge existence and visibility
E[edge id]	VOUT	[vertex id]	-	Out vertex
E[edge id]	VIN	[vertex id]	-	In vertex
E[edge id]	PROP	[prop name + key]	[prop value]	Property
E[edge id]	PROPMETA	[prop name + key]	[prop meta]	Property Metadata

Data Table Format

Row ID	Column Family	Column Qualifier	Value	Description
V/E[vertex/edge id + prop name + prop key]	-	-	[data]	Data

Large Data Overflow

Data that is larger than the default 10Meg gets overflowed to HDFS if needed.

Elasticsearch

Data is indexed but not stored in Elasticsearch

We are working on a solution to apply visibility to documents but this is not done yet.

Find out more...

<https://github.com/altamiracorp/secure-graph>