

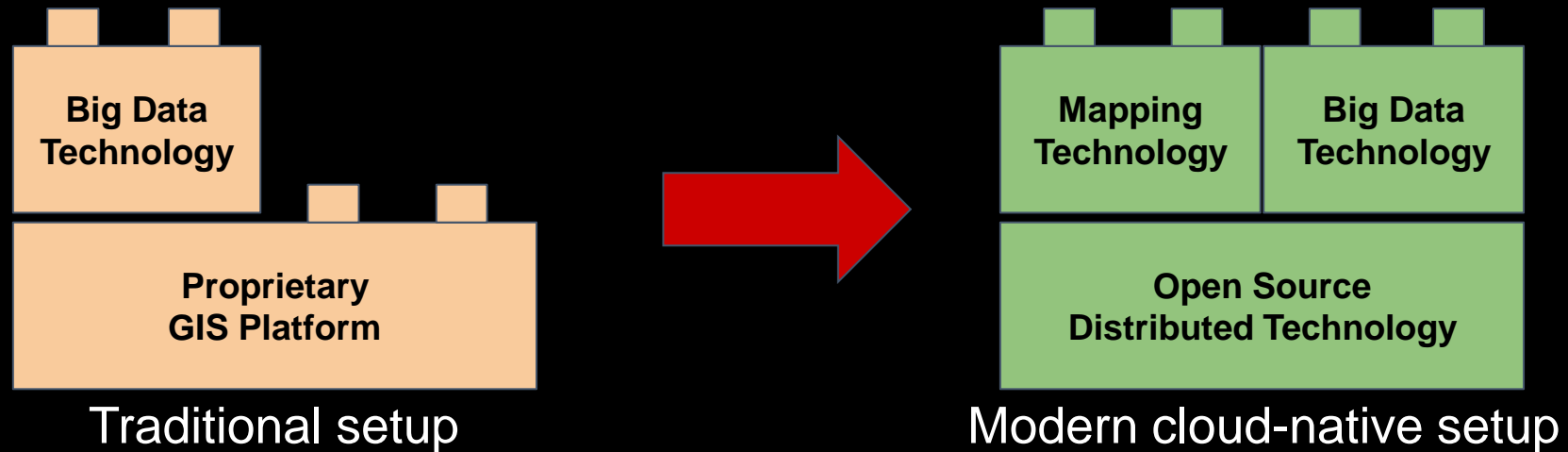
Open Source Trench-based Network Modelling

Jesper Ladegaard, DAX ApS
jesper@dax.dk



Agenda:

- An inspiring history from Bangkok
- Trench/Route-concept as a bridge between geographical data and complex network modelling data
- Examples of how the trench-concept in combination with native cloud technology can be used to optimize utility processes



Inspiring history



Bill Gates

Page Liked · 19 hrs · 🌐

LIVE WIRES

Due to faulty infrastructure, many urban areas suffer from frequent blackouts and power cuts, and the electrical grid often doesn't serve the people who need it most.

I've visited many cities filled with tangled wires such as those in this photo from Thailand, where people have illegally tapped into the grid on their own to get the power they need—at great personal risk.

👍 Like 💬 Comment ➦ Share



Overhead-to-underground cable conversion project in Bangkok



Asawin Kwanmuang, Bangkok Governor promotes underground cable project. Source: [Bangkok post](https://www.bangkokpost.com).

All power and telecom lines are "moved" into the ground



> 250 km tunnel network holding power transmission and telecom-backbone



Pipe-jacking / micro-tunnelling

30-50 meters deep below metro lines

Price around 15 mill euro per kilometer

A piece of "bad ass" tunnel route seen from the inside

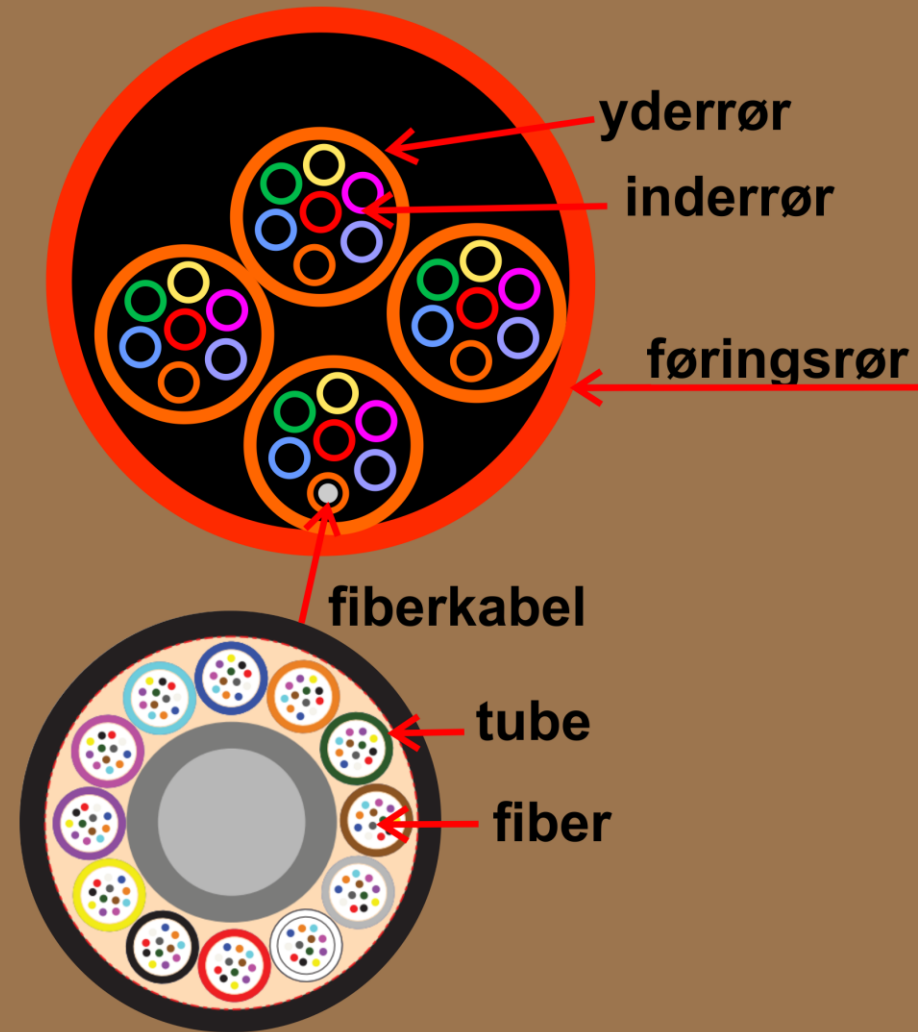


Street- food
arriving 😊

Telecom-
backbone
fibercables
In multi-
conduits

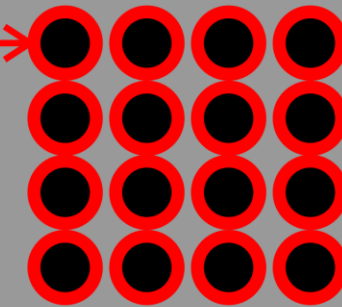
230 kV
water
cooled
power
lines

2.450 km access-network as we know from Denmark (multi conduit technology)

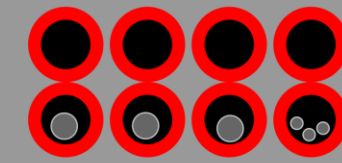


tracé

tele rørblok



el rørblok

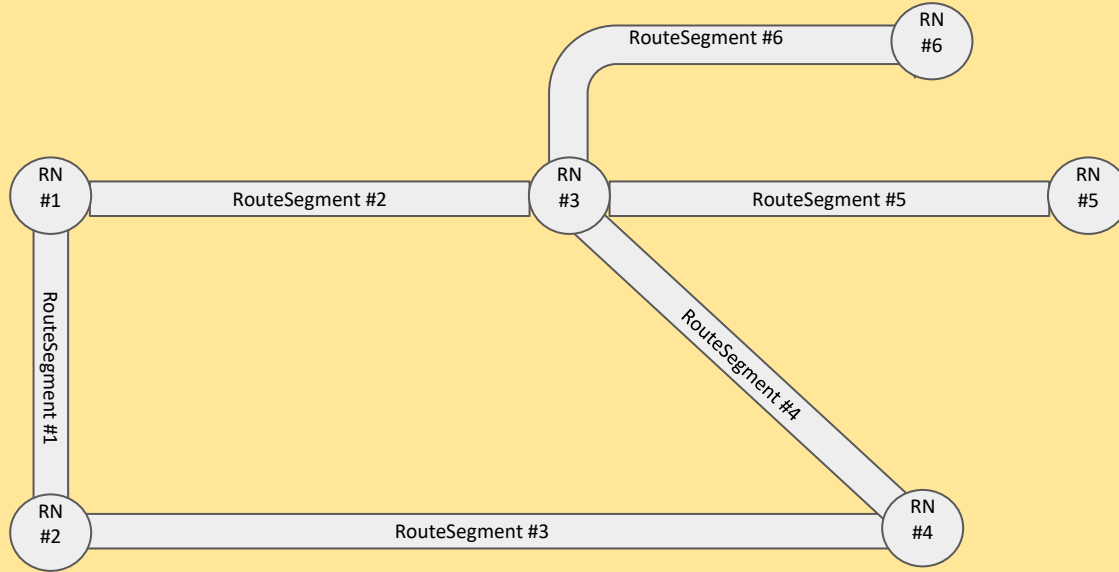


80 cm.

200 cm.

Graph-based modelling is a blessing dealing with complex multi-level topology

Geodata part as route nodes and route segments



Non-geographical part, modelled as a multi-level graph:

- Duct banks
- Multi conduits (outer and inner ducts)
- Power and fiber cables
- Electrical wires / Tubes and fibres in fiber cables
- Physical connectivity
- Logical/topological connectivity
- Network models inside nodes

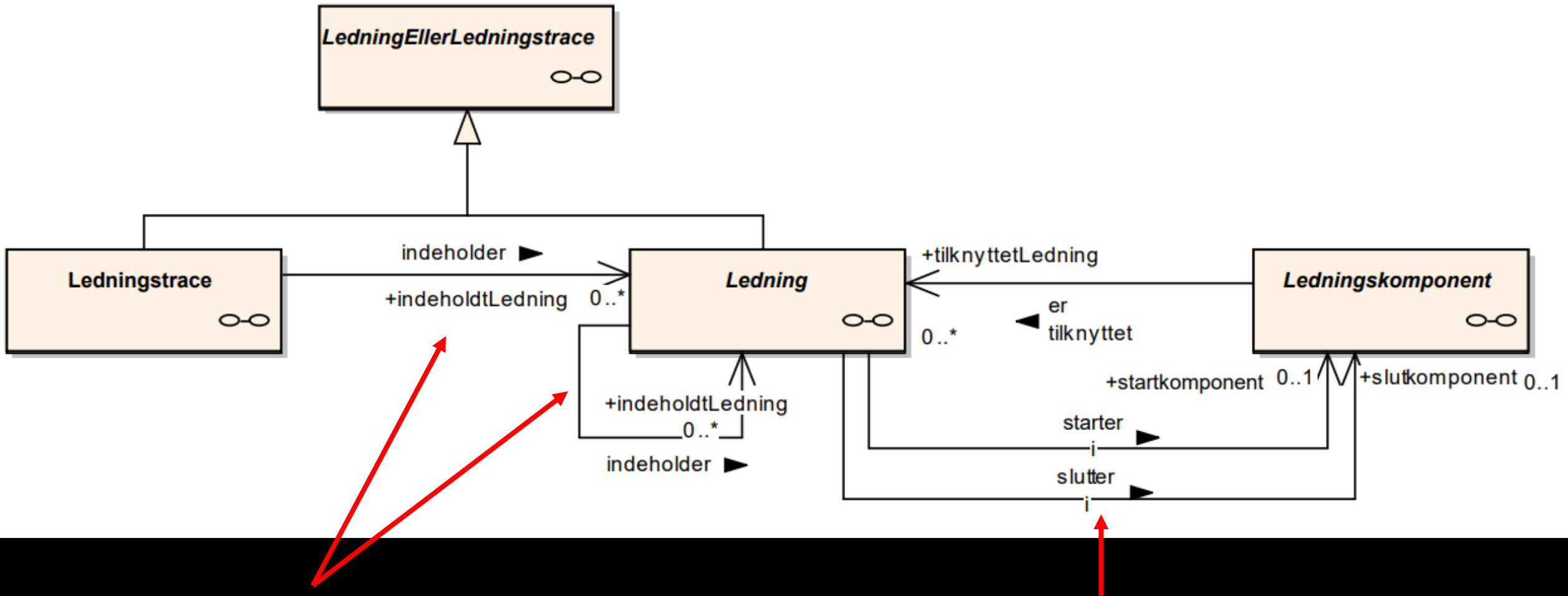
Example geographical view of the route network

Example view of the equipments related to a specific route element

The screenshot shows a mobile application interface. The top part displays a map with a blue line representing a route. Labels on the map include 'Bredalkærvej', 'J-1010', 'SP-1010', 'Flatliner 12x12/8', and 'I100003'. The bottom part shows a detailed view of the route element 'SP-1010'. It includes tabs for 'general.conduits', 'general.equipment', and 'general.circuits'. Under 'general.equipment', there are two entries for 'Flatliner 12x12/8' with '+' and '-' signs. At the bottom, there is a table with columns 'FROM' and 'TO'.

FROM	TO	(+)
I100003	SP-1010	
Lærkelunden	Lærkelunden	(+)

Danish LER 2.0 support



Composite parent-child structure

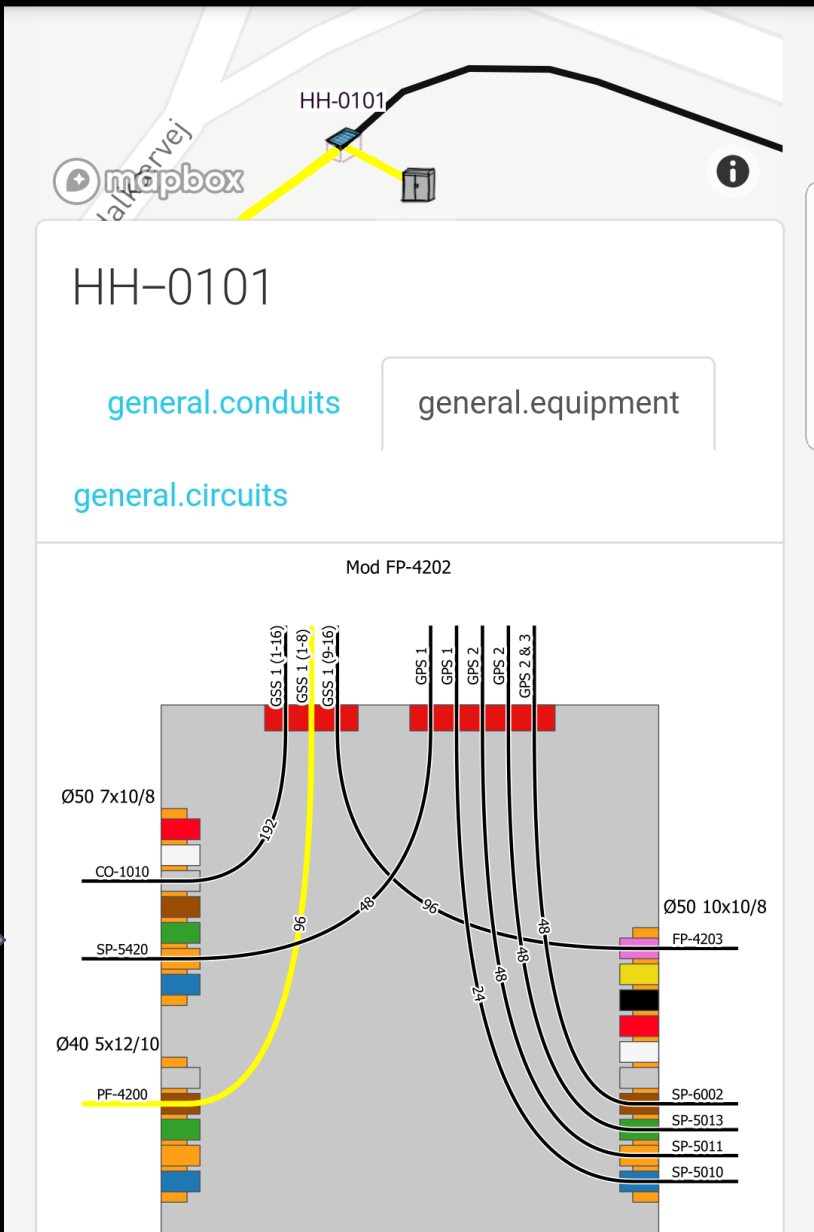
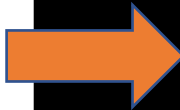
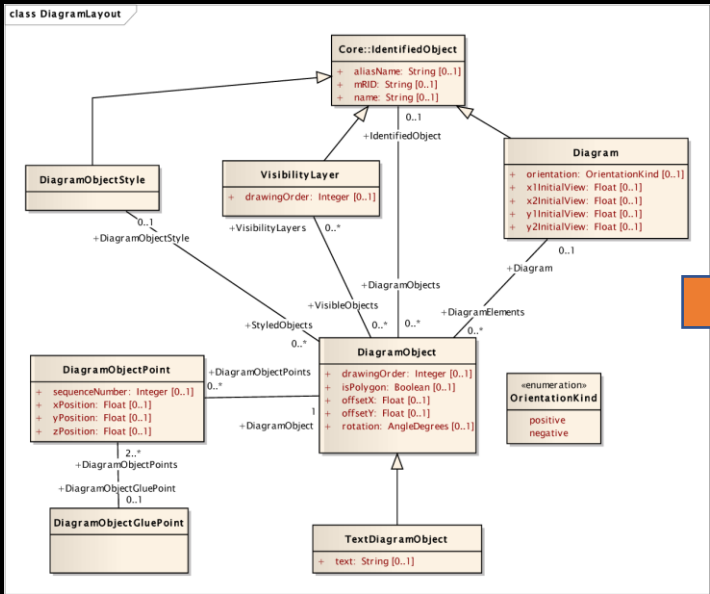
One route segment can hold segments (i.e. Conduits), that again can hold segments (i.e. telco or power lines),

Maps 1:1 to graph-strukture

Ledning (line segment) = graph link.

Ledningskomponent (komponent) = graph vertex

CIM Diagram Layout to facilitate flexible and automated schematic functionality



Flexible query API using GraphQL and distributed database technology

Q closure

M reset

Q routeNode

Q routeNode

Q diagramService

Q routeNode X

+

PRETTIFY HISTORY <https://localhost:44345/graphql> COPY CURL

```
1 {
2   routeNode(id: "0b2168f2-d9be-455c-a4de-e9169f000022") {
3     name
4     relatedConduits {
5       relationType
6       conduit {
7         id
8         kind
9         name
10        position
11        color
12        colorMarking
13        parent {
14          id
15          colorMarking
16        }
17        children {
18          id
19          position
20          color
21        }
22      }
23      conduitSegment {
24        line {
25          startRouteNode {name}
26          endRouteNode {name id}
27          allConduitSegments { id }
28        }
29      }
30    }
31  }
32 }
```

▶

```
{
  "data": {
    "routeNode": {
      "name": "SP-1010",
      "relatedConduits": [
        {
          "relationType": "OUTGOING",
          "conduit": {
            "id": "ece18b83-e79f-e82e-184f-65bbe12cc24e",
            "kind": "MULTI_CONDUIT",
            "name": "R000067",
            "position": "0",
            "color": "CLEAR",
            "colorMarking": "RED",
            "parent": null,
            "children": [
              {
                "id": "2f47c8d4-8c79-49e3-b726-f2876f258230",
                "position": "3",
                "color": "GREEN"
              },
              {
                "id": "2bbfae82-a6ea-40a7-9ce5-67640ca9e183",
                "position": "1",
                "color": "BLUE"
              },
              {
                "id": "8aeaf874-b8e7-4a2d-8e3a-ff56c21b3dda",
                "position": "12",
                "color": "AQUA"
              }
            ]
          }
        }
      ]
    }
  }
}
```

DOCS SCHEMA

QUERY VARIABLES HTTP HEADERS TRACING

**Network
Traversal
part**



```
23      conduitSegment {
24        line {
25          startRouteNode {name}
26          endRouteNode {name id}
27          allConduitSegments { id }
28        }
29      }
```


Fast traversal queries on graph structures

closure reset routeNode routeNode diagramService routeNode

PRETTIFY HISTORY <https://localhost:44345/graphql> COPY CURL

```
1 {
2   routeNode(id: "0b2168f2-d9be-455c-a4de-e9169f000022") {
3     name
4     relatedConduits {
5       relationType
6       conduit {
7         id
8         kind
9         name
10        position
11        color
12        colorMarking
13        parent {
14          id
15          colorMarking
16        }
17        children {
18          id
19          position
20          color
21        }
22      }
23      conduitSegment {
24        line {
25          startRouteNode {name}
26          endRouteNode {name id}
27          allConduitSegments { id }
28        }
29      }
30    }
31  }
32 }
```

Log

Get Started Statistics Inspectors AutoResponder

Request Count: 1
Bytes Sent: 2.002 (headers:534; body:1.468)
Bytes Received: 72.357 (headers:324; body:72.033)

ACTUAL PERFORMANCE

ClientConnected:	23:57:46.452
ClientBeginRequest:	00:00:23.005
GotRequestHeaders:	00:00:23.005
ClientDoneRequest:	00:00:23.005
Determine Gateway:	Oms
DNS Lookup:	Oms
TCP/IP Connect:	Oms
HTTPS Handshake:	Oms
ServerConnected:	23:57:37.573
FiddlerBeginRequest:	00:00:23.005
ServerGotRequest:	00:00:23.006
ServerBeginResponse:	00:00:23.044
GotResponseHeaders:	00:00:23.044
ServerDoneResponse:	00:00:23.048
ClientBeginResponse:	00:00:23.048
ClientDoneResponse:	00:00:23.050

Overall Elapsed: 0:00:00.045

45 millisekunder !

position: "12",
color: "AQUA"

QUERY VARIABLES HTTP HEADERS

TRACING

This kind of query normally takes minutes on a traditional relational-database-driven GIS platform