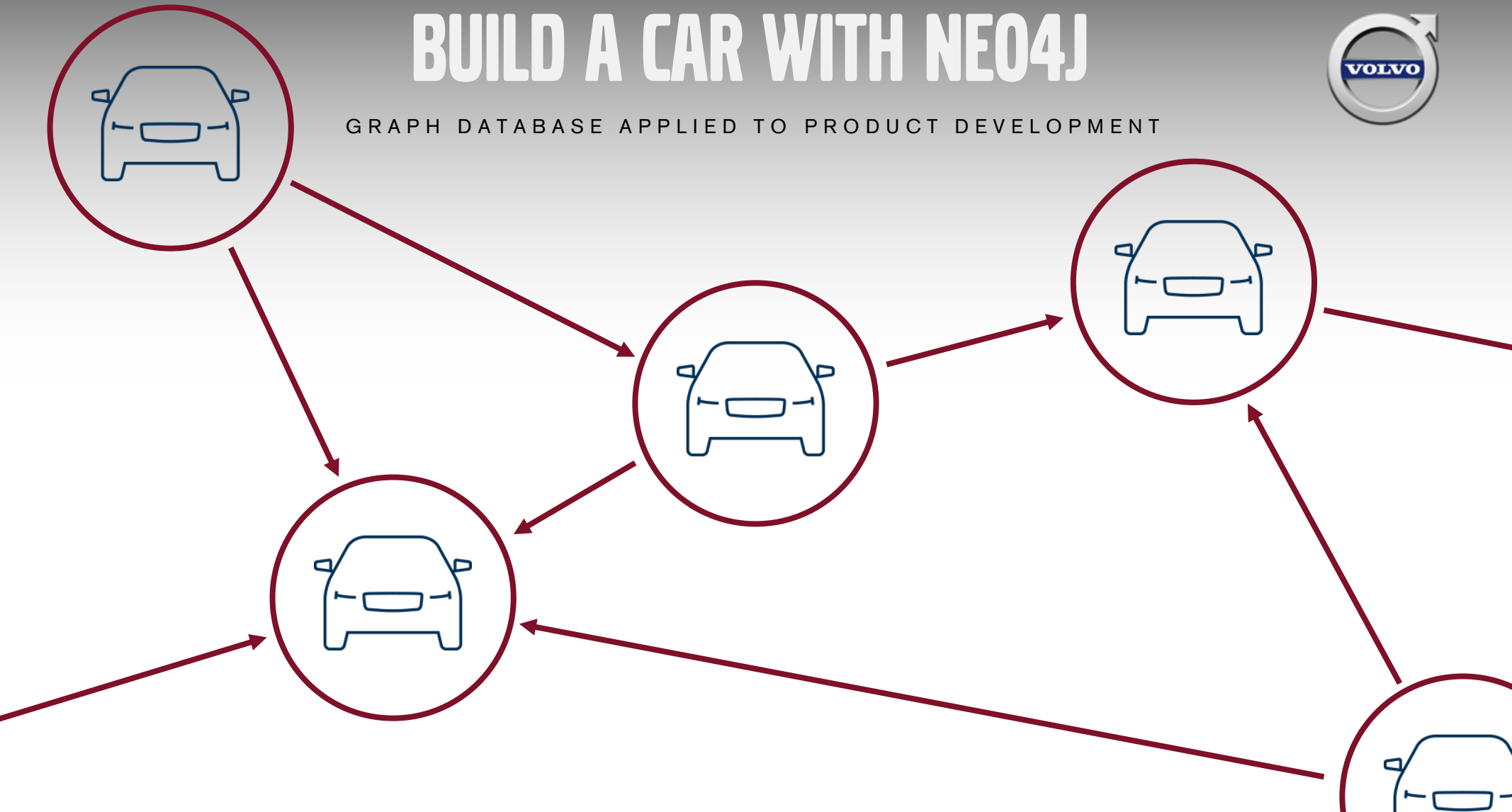
# BUILD A CAR WITH NEO4J

GRAPH DATABASE APPLIED TO PRODUCT DEVELOPMENT

# BUILD A CAR WITH NEO4J

GRAPH DATABASE APPLIED TO PRODUCT DEVELOPMENT



# OUTLINE

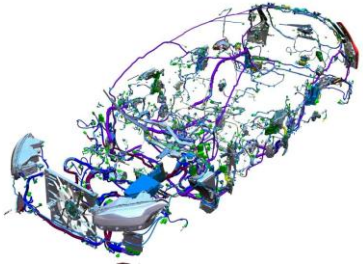
- The challenges of complex product development
- How can Neo4j help?
- System Engineering at Volvo Cars with Neo4j

# THE CHALLENGES OF COMPLEX PRODUCT DEVELOPMENT

# IS A CAR A COMPLEX PRODUCT?

- 30,000 hardware components
- 200 ECU
- 100 communication busses
- 2,000 software components
- 10,000 signals
- 400 functions
- >100,000 configurations
- No place for critical bugs or safety issues

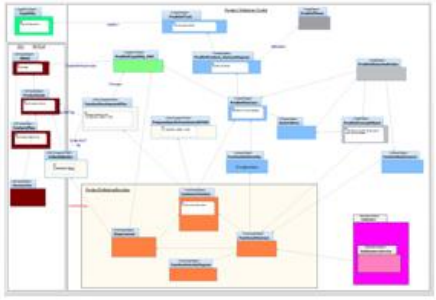


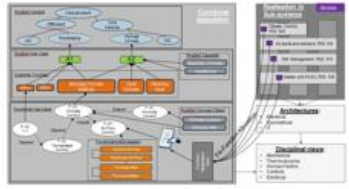


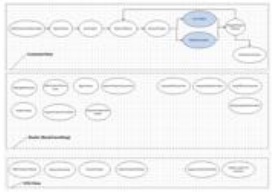


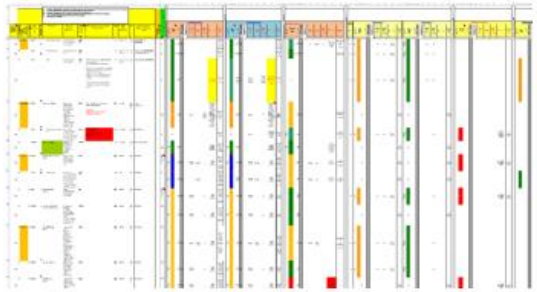
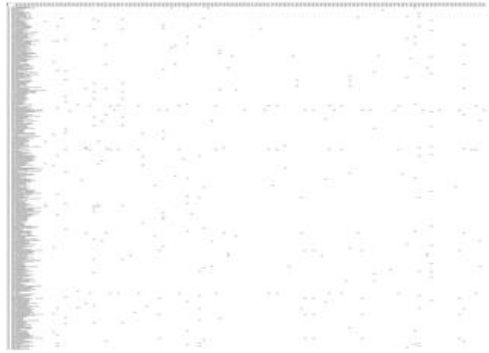


# HOW DO WE DO TODAY?



Rhapsody, Visio, PowerPoint, Excel, SystemWeaver, Elektra, TeamCenter, ...

# WHAT SHOULD WE DO?

- Let's **KILL COMPLEXITY**


or?

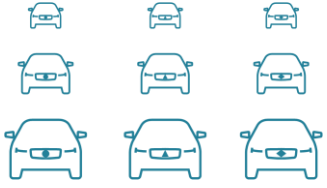
# ON THE HORIZON - FOCUS AREAS

Autonomy

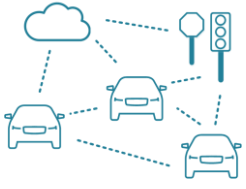
Electrification

Connectivity

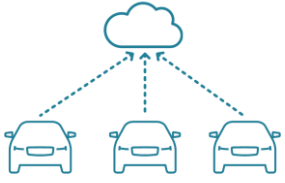
Sharing



Multiple brands
multiple segments



System of systems



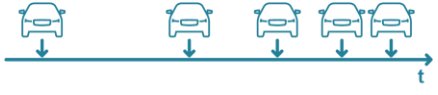
Crowd sourced data

Machine learning

Third party access


Personalization


Product evolution
after original sale




Decreasing time to market

. . .

Unknowns

# WHAT SHOULD WE DO?

- Let's **KILL ACCIDENTAL COMPLEXITY**
- Let's **MANAGE ESSENTIAL COMPLEXITY**

**Accidental Complexity**
Arises purely from mismatches in the particular choice of tools and methods applied in the solution.

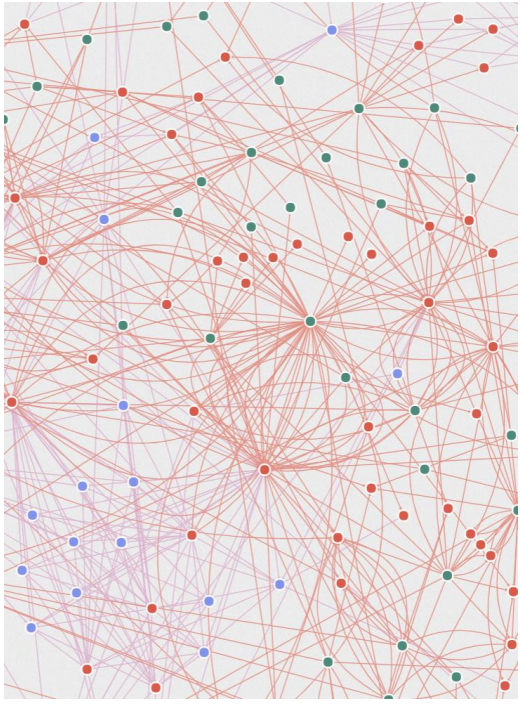
**Essential Complexity**
The absolute minimal necessary complexity arising from adequately developing the solution to a problem.

We need to **UNDERSTAND COMPLEXITY !!!**

# MAIN HINDERS TO SECURE PRODUCT QUALITY

- Huge amount of data:
  - Organized among many different specialized tools
  - Handled in silos
  - Difficult/impossible to understand the big picture
  - Difficult/impossible to navigate through the silos

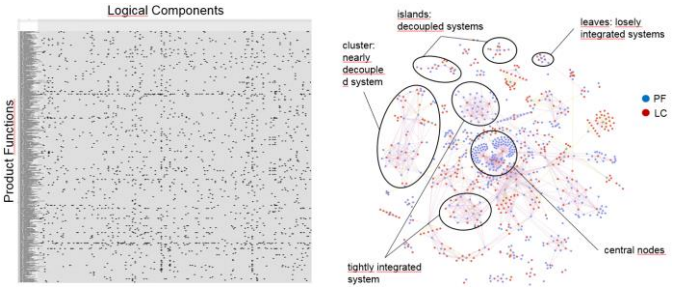
- Highly integrated systems with many dependencies:
  - Dependencies between data objects are often stored in brains
  - Difficult/impossible to comprehend a system design
  - Difficult/impossible to evaluate complexity



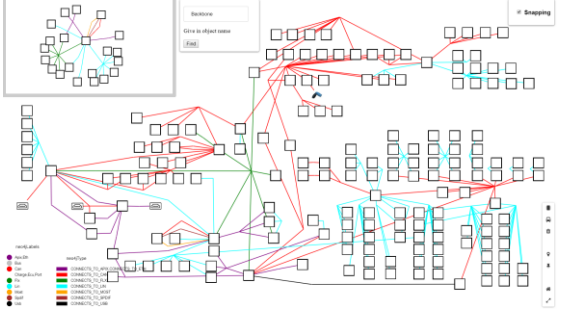
# HOW CAN NEO4J HELP?

# THE WAYS TO USE NEO4J

- Interface Database between different systems
  - Reveal modelled dependencies
  - Allows Knowledge Mining

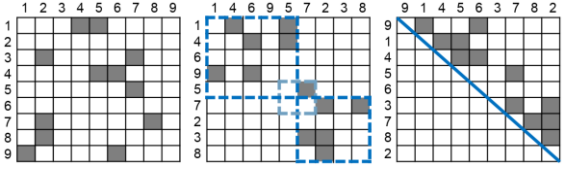



- System modelling platform
  (Neo4j + Visualization Tool)



- DSM (Design Structure Matrix) Analytics
  - System Design visualization
  - System Design optimization

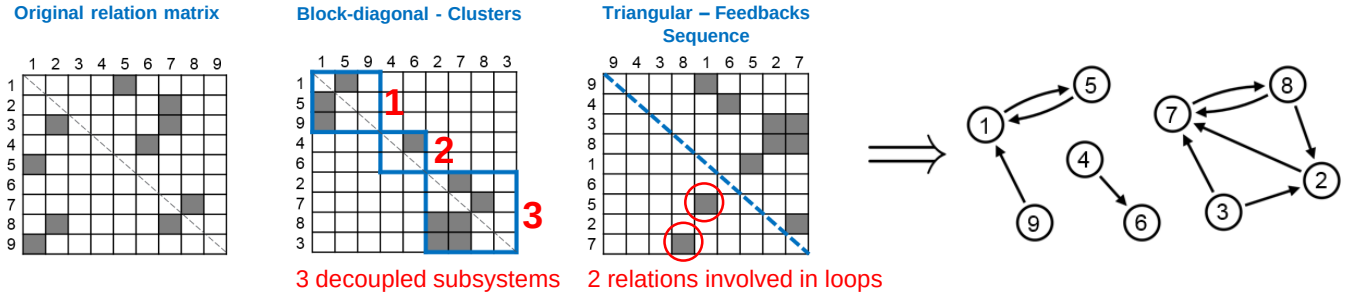
  → Complexity Management



# DSM (DESIGN STRUCTURE MATRIX) ANALYTICS

- "(The DSM) is the equivalent of an **adjacency matrix in graph theory**, and is used in systems engineering and project management to model the structure of **complex systems or processes**, in order to perform **system analysis**, project planning and organization design" (Wikipedia).

- Analytics:
  - Clusters
  - Feedbacks
  - Sequencing




- DSM algorithms are generally sub-optimally applied to the matrix-plan itself
  - Heavy calculation that can take hours to deliver an answer for "large" systems
  - Results are often approximative

- **DSM algorithms are actually Graph Theory algorithms!**

# SYSTEM ENGINEERING AT VOLVO CARS WITH NEO4J

# VOLVO CARS USE-CASE

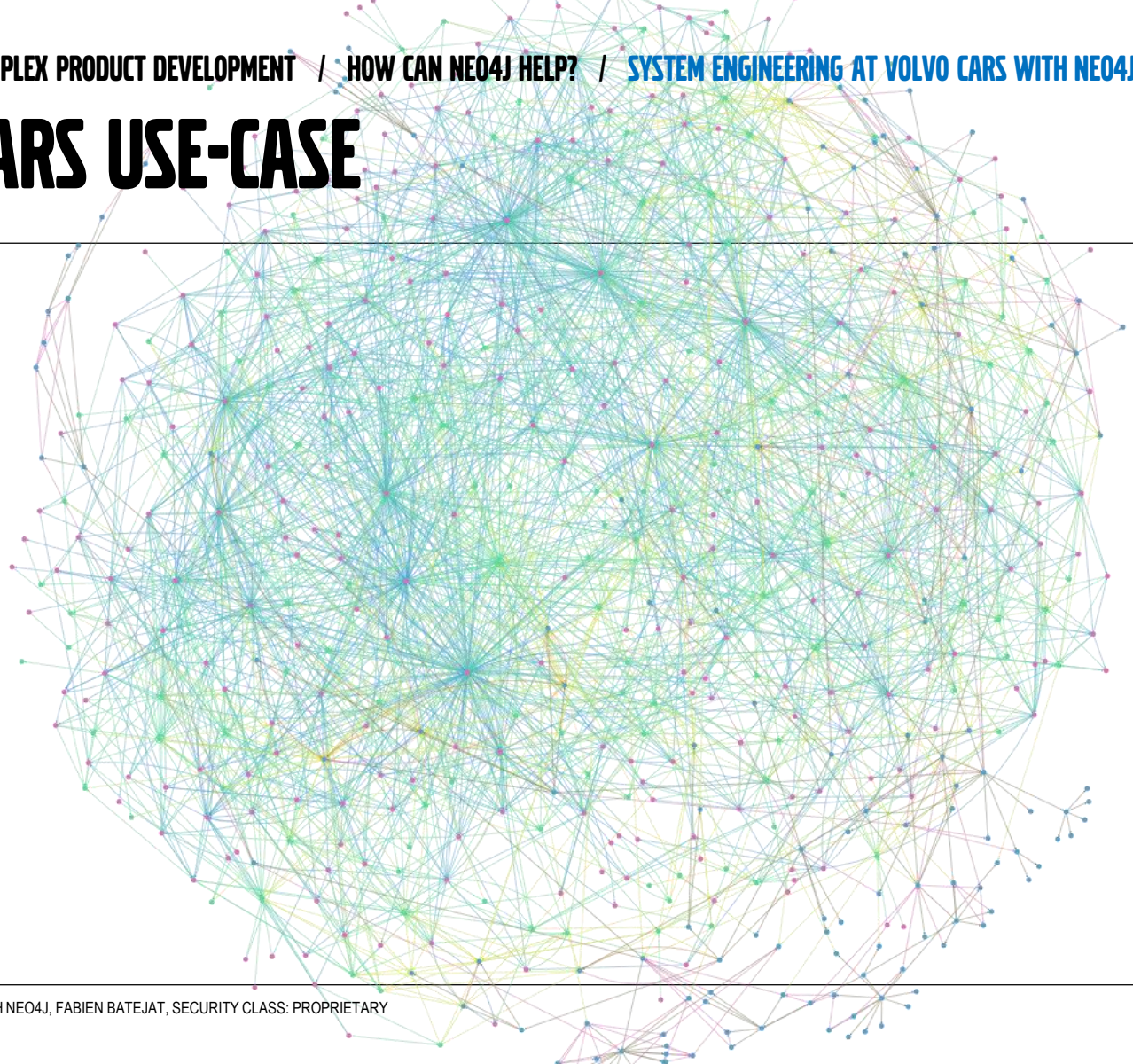
- *Connecting Systems Engineering and Human Factors through MBSE*

  Nilsson, Robert - Volvo Cars Corporation

  Bligård, Lars-Ola - Chalmers University of Technology

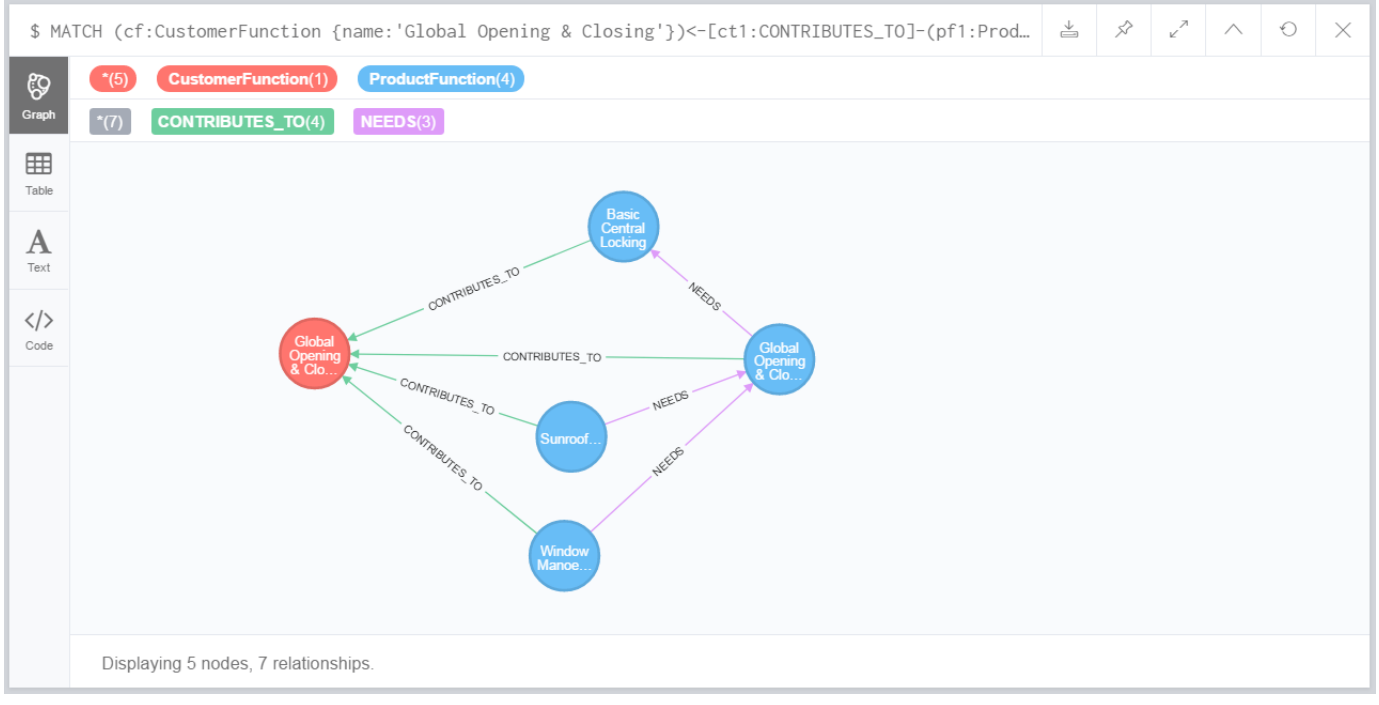
- Functional Disposition
  - Every system component is connected to the **customer needs** instead of mere engineering performance cursors.

# VOLVO CARS USE-CASE



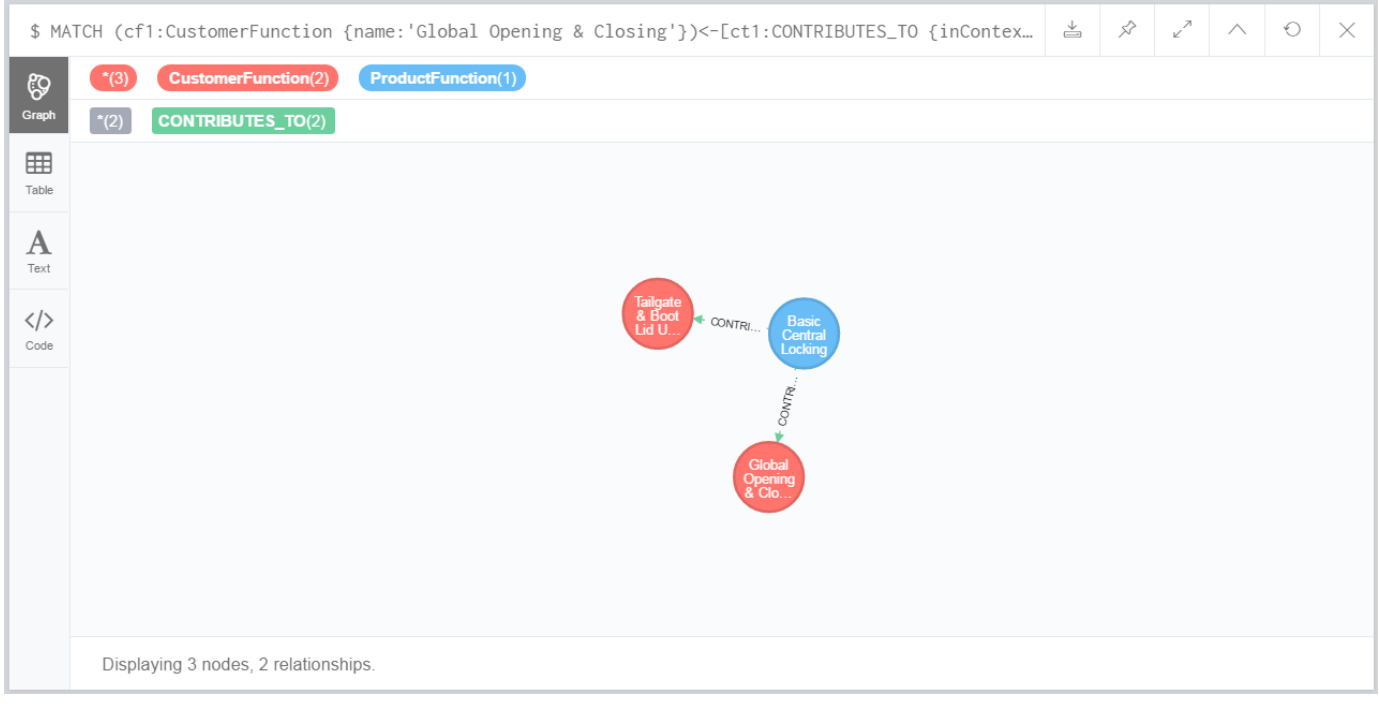
# VOLVO CARS USE-CASE

What PFs realize my CF? and how?



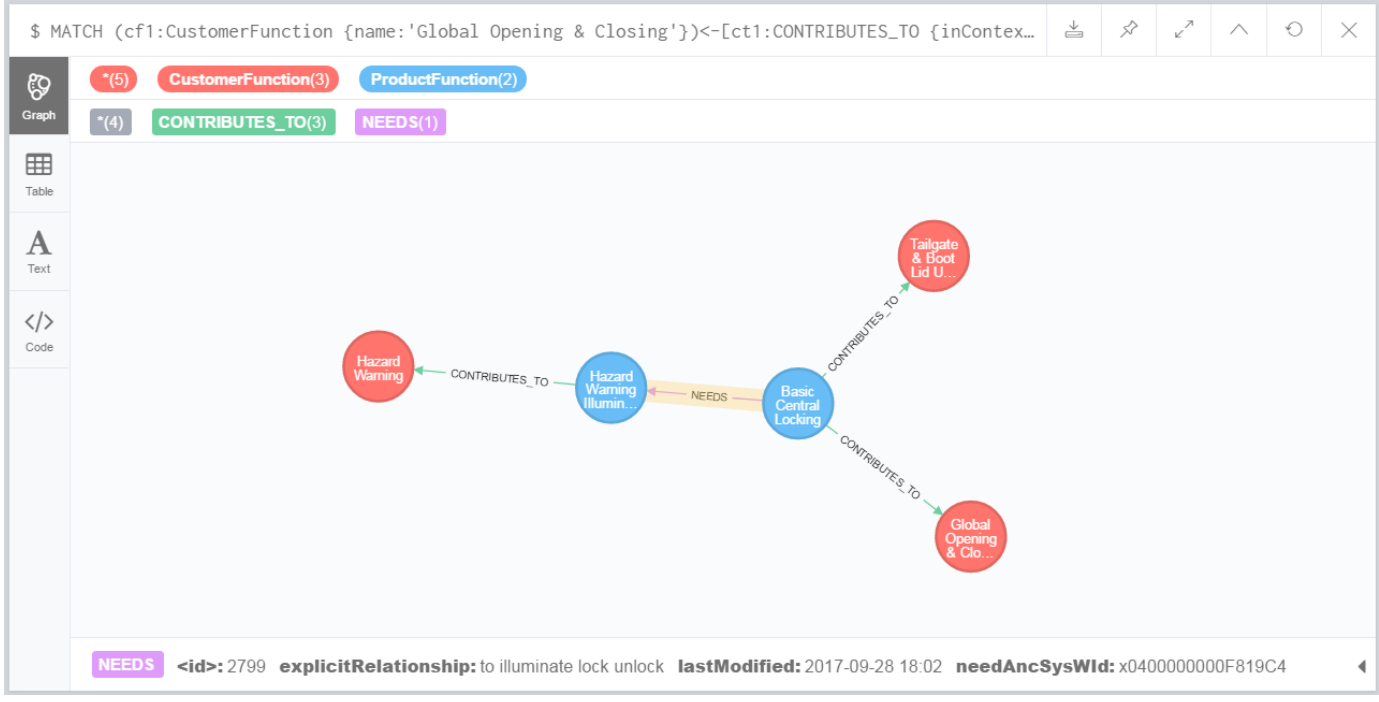
# VOLVO CARS USE-CASE

What CFs does my PF contribute to?



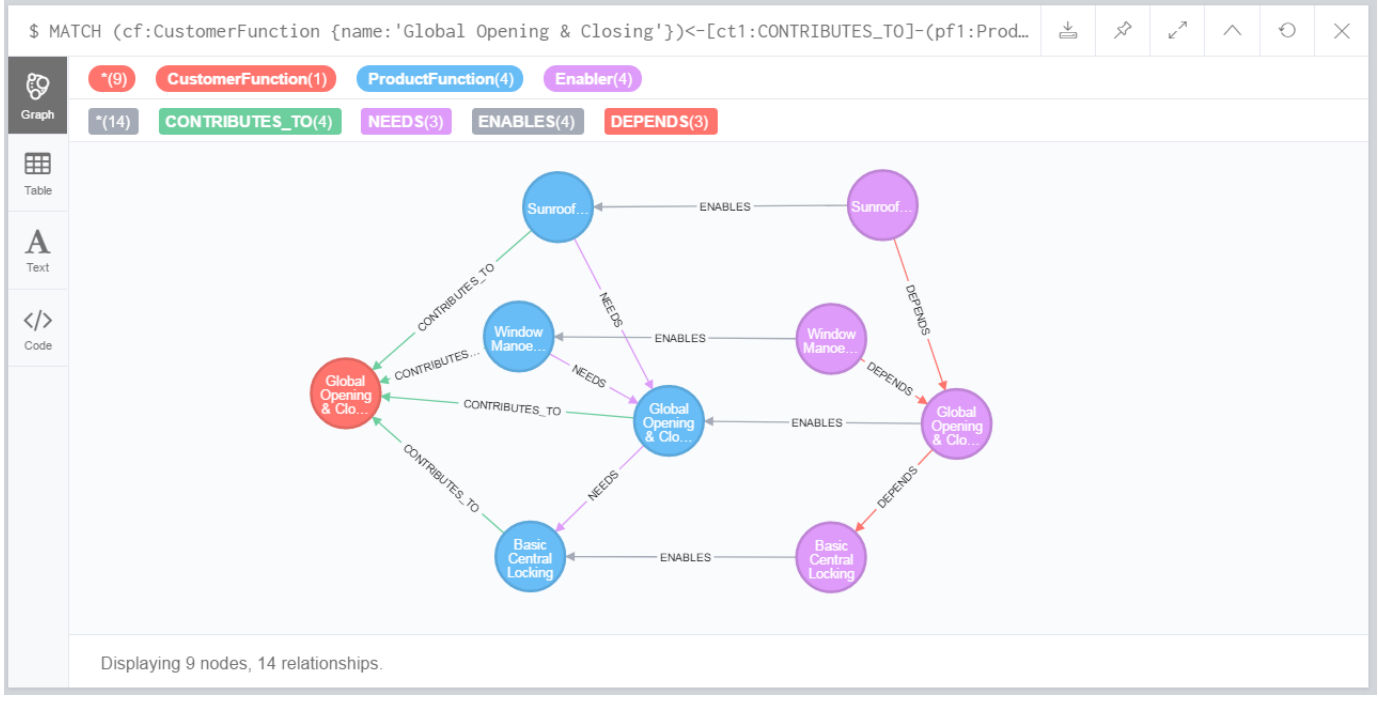
# VOLVO CARS USE-CASE

How does that PF relate to CF 'Hazard Warning'?



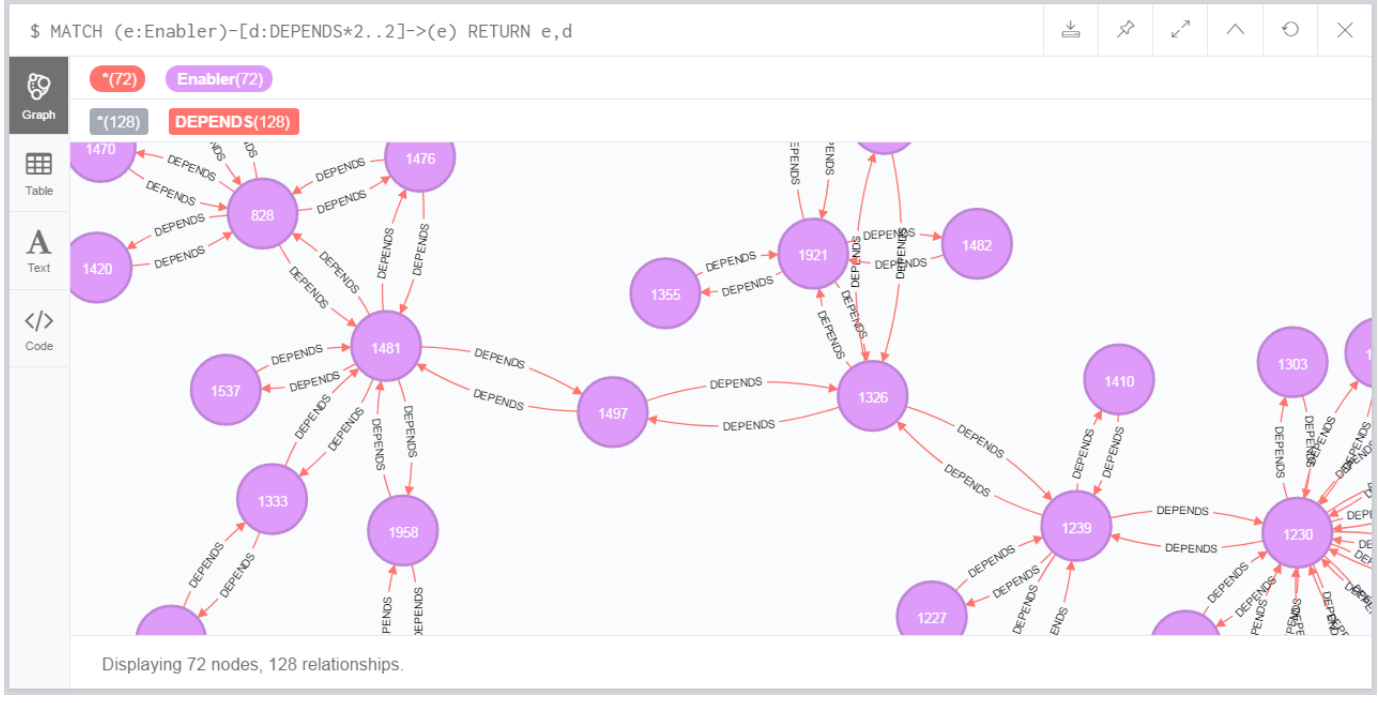
# VOLVO CARS USE-CASE

## Show Enablers layer

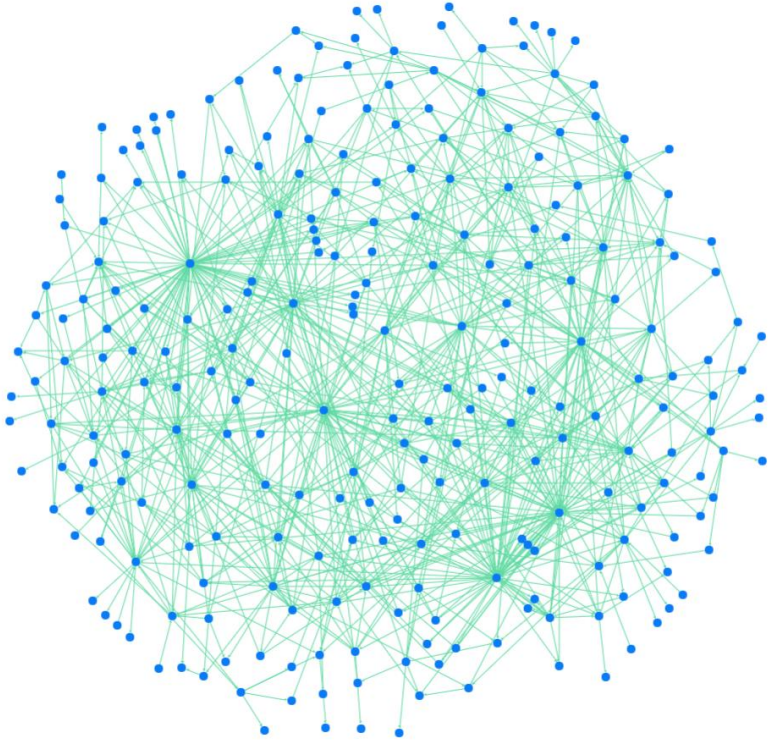


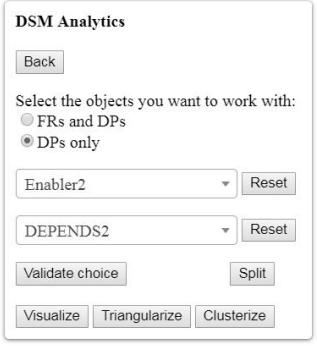
# VOLVO CARS USE-CASE

Create Enablers and find / handle loops

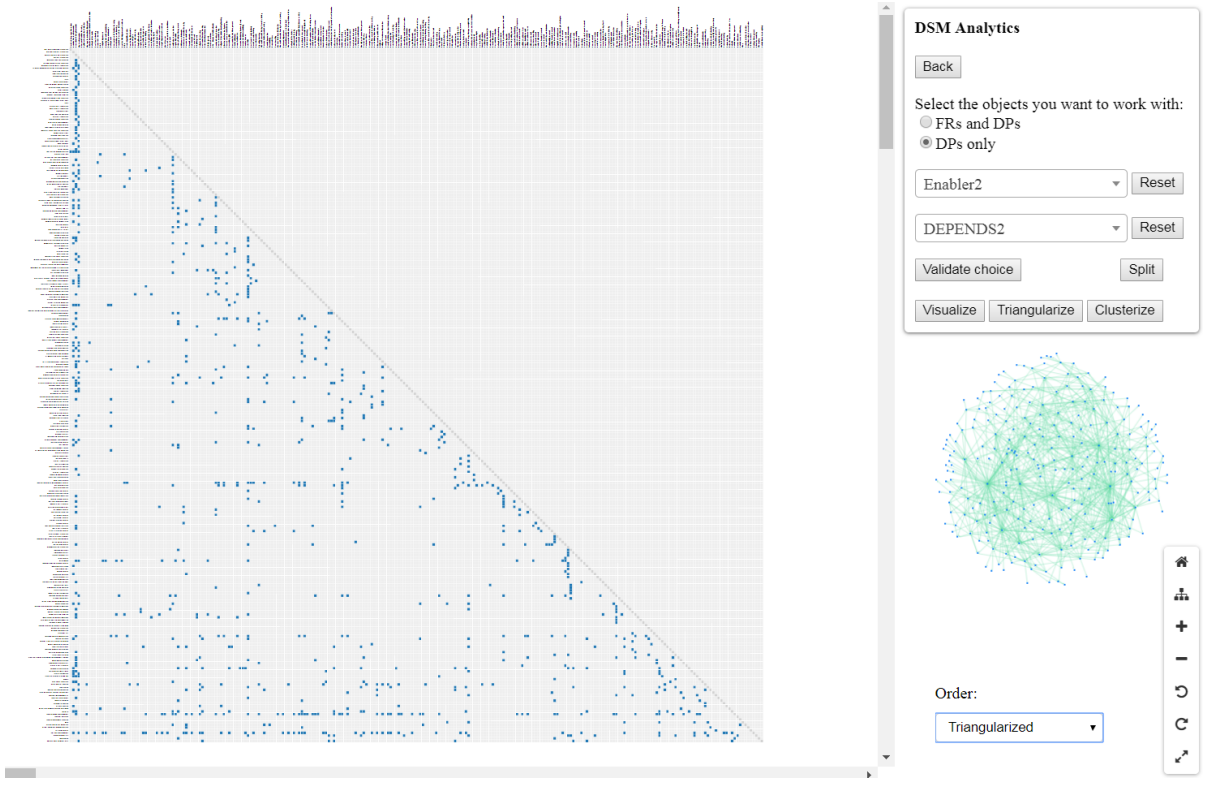


# VOLVO CARS USE-CASE

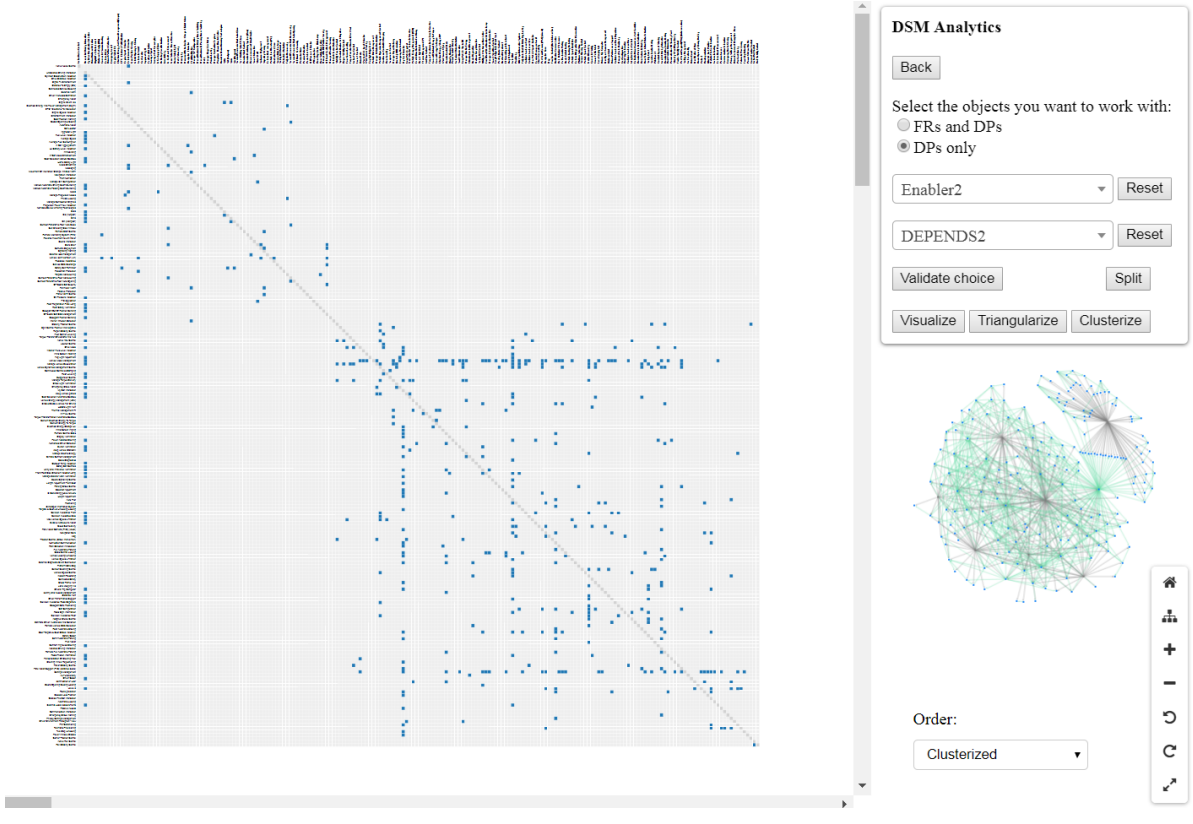




# VOLVO CARS USE-CASE

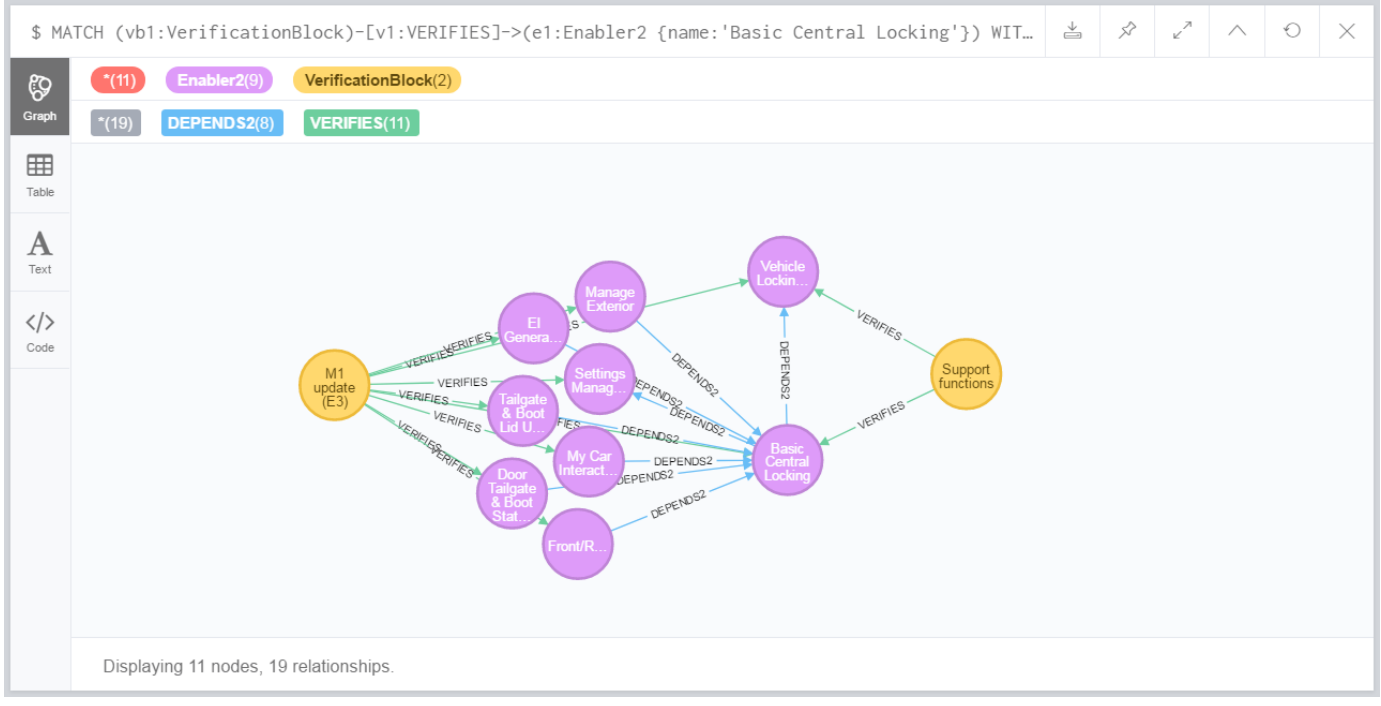


# VOLVO CARS USE-CASE

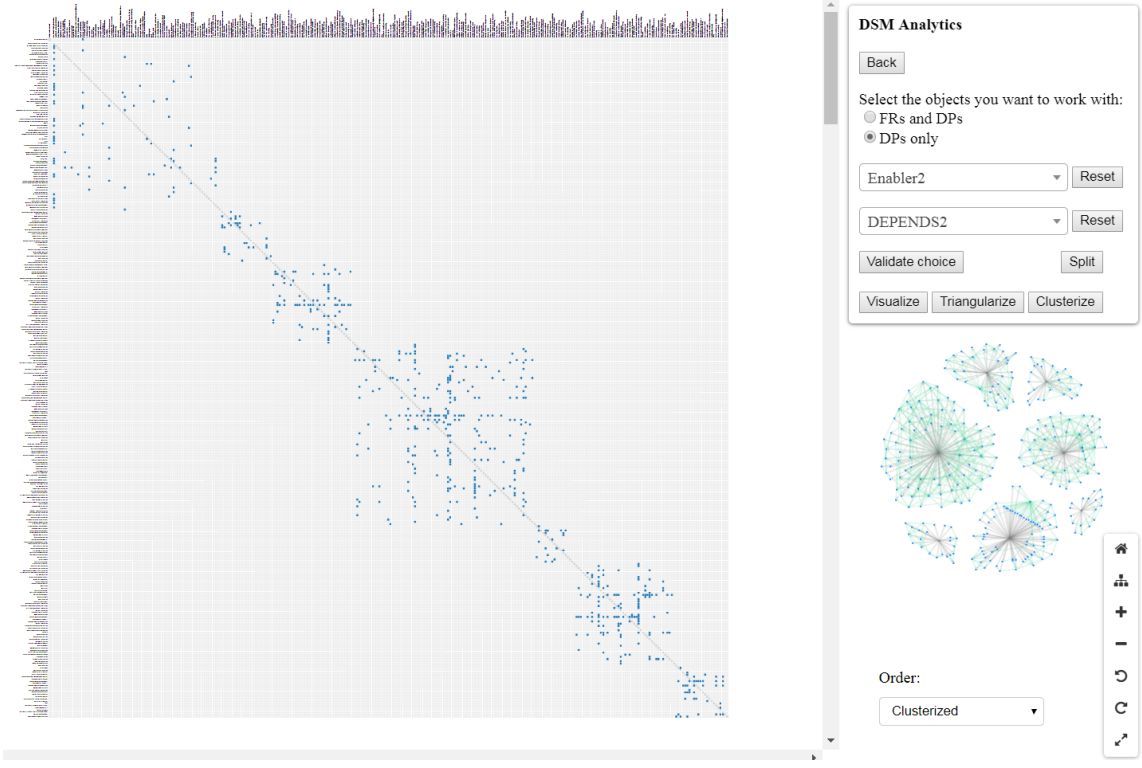


# VOLVO CARS USE-CASE

## Show 'Basic Central Locking' before Split



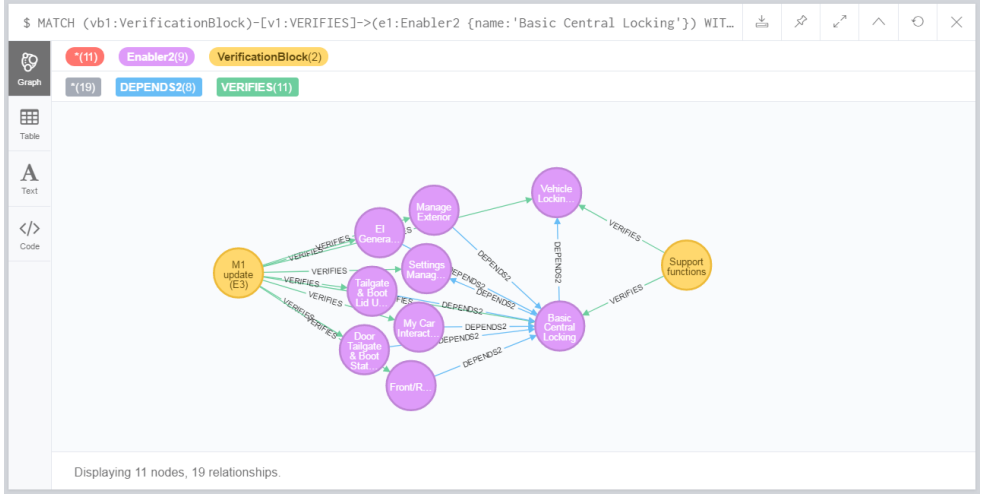
# VOLVO CARS USE-CASE



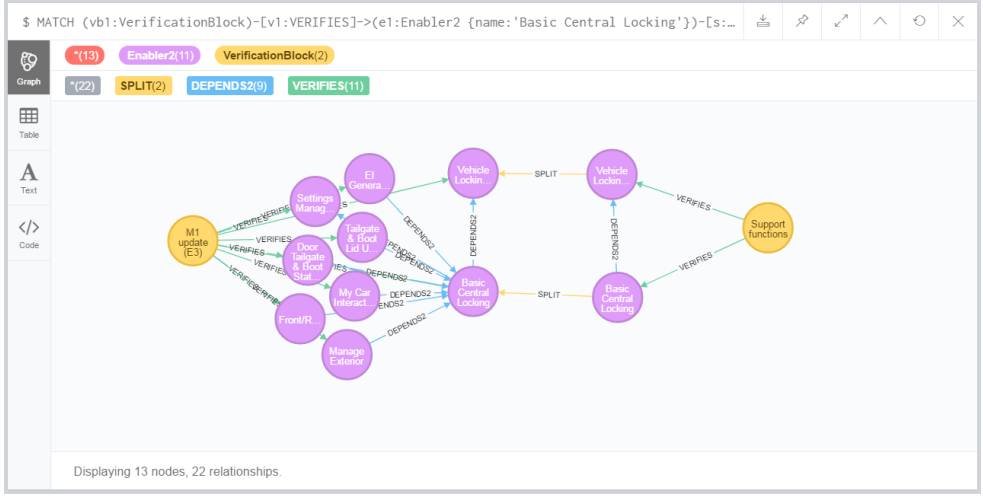
# VOLVO CARS USE-CASE

Show 'Basic Central Locking' before and after Split

## Before split




## After split



# TO YOU NEO4J COMMUNITY

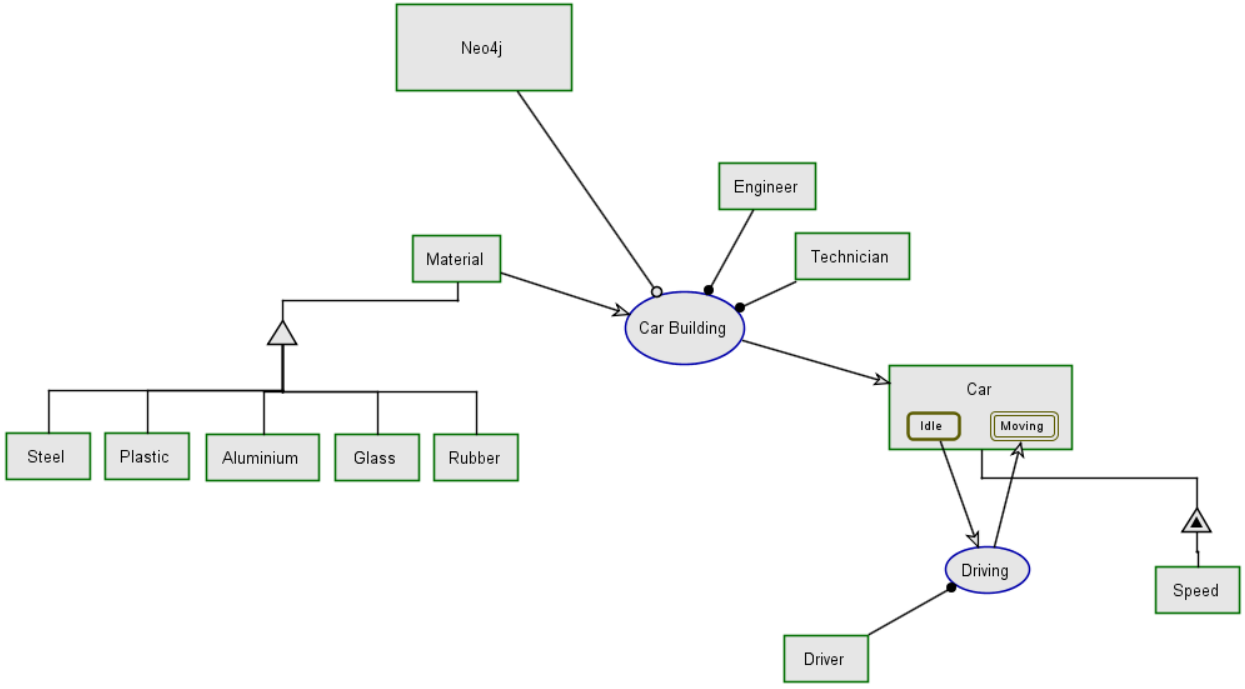
- Graph Database potential for Product Development is huge
  - Systems are always more connected, complex and integrated
  - At the same time, we expect them to be intuitive and robust

- Fully understanding a system's design is a game changer
  - Graph Database allows just that

- My call:
  - Let us contribute more to System Design use-case

    Some hints:
    - DSM Analytics
    - Axiomatic Design
    - Analysis of System Engineering models with OPM (Object Process Methodology)
    - 3D graph visualization
    - ...

# OBJECT PROCESS METHODOLOGY



# 3D GRAPH VISUALIZATION