Take-home Assignment 2

Quynh Trang Do

2025-05-28

# 1. Objective

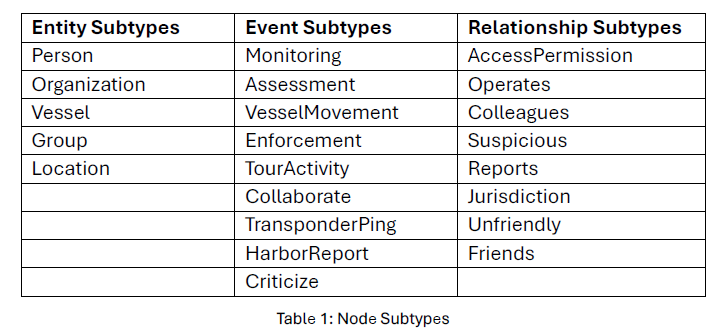
In this exercise, I will tackle Question 2 from [Mini-Challenge 3](https://vast-challenge.github.io/2025/MC3.html) of the [VAST Challenge 2025](https://vast-challenge.github.io/2025/index.html).

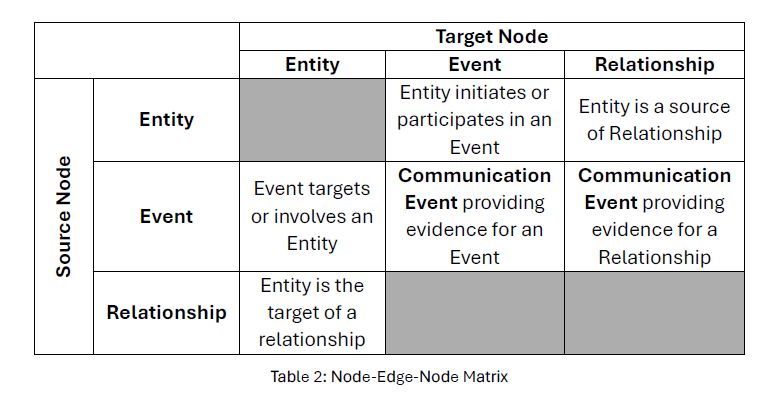
**Question 2**: Clepper has noticed that people often communicate with (or about) the same people or vessels, and that grouping them together may help with the investigation. Use visual analytics to help Clepper understand and explore the interactions and relationships between vessels and people in the knowledge graph. Are there groups that are more closely associated? If so, what are the topic areas that are predominant for each group?

# 2. Data Overview

The dataset, MC3\_graph.json, is a directed knowledge graph constructed from transcripts of boat radio communications. It aims to capture interactions between people, their roles, and the events and locations they discuss, ultimately supporting the prediction of when and where future events may occur.

The graph contains 1,159 nodes and 3,226 edges, and is stored in JSON format. Full description of node attributes and edge attributes is shown below.





# 3. Set up

## 3.1. Load necessary packages

pacman::p\_load(tidyverse, jsonlite, dplyr, igraph,  
 tidygraph, ggraph, scales, visNetwork,  
 SmartEDA, widyr, ggforce, tidytext, htmlwidgets,  
 webshot2, knitr)

## 3.2. Import data

### Import data

MC3 <- fromJSON("data/MC3\_graph.json")  
MC3\_schema <- fromJSON("data/MC3\_schema.json")

### Quick check on the imported data

glimpse(MC3)

List of 5  
 $ directed : logi TRUE  
 $ multigraph: logi FALSE  
 $ graph :List of 4  
 ..$ mode : chr "static"  
 ..$ edge\_default: Named list()  
 ..$ node\_default: Named list()  
 ..$ name : chr "VAST\_MC3\_Knowledge\_Graph"  
 $ nodes :'data.frame': 1159 obs. of 31 variables:  
 ..$ type : chr [1:1159] "Entity" "Entity" "Entity" "Entity" ...  
 ..$ label : chr [1:1159] "Sam" "Kelly" "Nadia Conti" "Elise" ...  
 ..$ name : chr [1:1159] "Sam" "Kelly" "Nadia Conti" "Elise" ...  
 ..$ sub\_type : chr [1:1159] "Person" "Person" "Person" "Person" ...  
 ..$ id : chr [1:1159] "Sam" "Kelly" "Nadia Conti" "Elise" ...  
 ..$ timestamp : chr [1:1159] NA NA NA NA ...  
 ..$ monitoring\_type : chr [1:1159] NA NA NA NA ...  
 ..$ findings : chr [1:1159] NA NA NA NA ...  
 ..$ content : chr [1:1159] NA NA NA NA ...  
 ..$ assessment\_type : chr [1:1159] NA NA NA NA ...  
 ..$ results : chr [1:1159] NA NA NA NA ...  
 ..$ movement\_type : chr [1:1159] NA NA NA NA ...  
 ..$ destination : chr [1:1159] NA NA NA NA ...  
 ..$ enforcement\_type : chr [1:1159] NA NA NA NA ...  
 ..$ outcome : chr [1:1159] NA NA NA NA ...  
 ..$ activity\_type : chr [1:1159] NA NA NA NA ...  
 ..$ participants : int [1:1159] NA NA NA NA NA NA NA NA NA NA ...  
 ..$ thing\_collected :'data.frame': 1159 obs. of 2 variables:  
 .. ..$ type: chr [1:1159] NA NA NA NA ...  
 .. ..$ name: chr [1:1159] NA NA NA NA ...  
 ..$ reference : chr [1:1159] NA NA NA NA ...  
 ..$ date : chr [1:1159] NA NA NA NA ...  
 ..$ time : chr [1:1159] NA NA NA NA ...  
 ..$ friendship\_type : chr [1:1159] NA NA NA NA ...  
 ..$ permission\_type : chr [1:1159] NA NA NA NA ...  
 ..$ start\_date : chr [1:1159] NA NA NA NA ...  
 ..$ end\_date : chr [1:1159] NA NA NA NA ...  
 ..$ report\_type : chr [1:1159] NA NA NA NA ...  
 ..$ submission\_date : chr [1:1159] NA NA NA NA ...  
 ..$ jurisdiction\_type: chr [1:1159] NA NA NA NA ...  
 ..$ authority\_level : chr [1:1159] NA NA NA NA ...  
 ..$ coordination\_type: chr [1:1159] NA NA NA NA ...  
 ..$ operational\_role : chr [1:1159] NA NA NA NA ...  
 $ edges :'data.frame': 3226 obs. of 5 variables:  
 ..$ id : chr [1:3226] "2" "3" "5" "3013" ...  
 ..$ is\_inferred: logi [1:3226] TRUE FALSE TRUE TRUE TRUE TRUE ...  
 ..$ source : chr [1:3226] "Sam" "Sam" "Sam" "Sam" ...  
 ..$ target : chr [1:3226] "Relationship\_Suspicious\_217" "Event\_Communication\_370" "Event\_Assessment\_600" "Relationship\_Colleagues\_430" ...  
 ..$ type : chr [1:3226] NA "sent" NA NA ...

### Extract the node and edge tables

mc3\_nodes <- as\_tibble(MC3$nodes)  
mc3\_edges <- as\_tibble(MC3$edges)

## Conversion for htmp outputs

VisNetwork produces interactive HTML widgets, which cannot be embedded directly in Word (.docx). Thus, a helper function is created to render the html outputs, take a screenshot of each html output, and insert them into the docx.

render\_visnetwork\_as\_image <- function(vis\_obj, name = "network", width = 1000, height = 1000) {  
 dir.create("images", showWarnings = FALSE)  
  
 html\_file <- file.path("images", paste0(name, ".html"))  
 img\_file <- file.path("images", paste0(name, ".png"))  
  
 saveWidget(vis\_obj, file = html\_file, selfcontained = TRUE)  
 webshot2::webshot(  
 url = html\_file,  
 file = img\_file,  
 vwidth = width,  
 vheight = height,  
 cliprect = "viewport"  
 )  
  
 return(img\_file)  
}

# 4. Data Cleaning and Wrangling

## 4.1. For nodes

The code chunk below is used to convert values in id field into character data type, exclude records with ID value are NA, exclude records with similar ID values, exclude thing\_collected field, and save the cleaned tibble dataframe into a new tibble datatable called mc3\_nodes\_cleaned.

mc3\_nodes\_cleaned <- mc3\_nodes %>%  
 mutate(id = as.character(id)) %>%  
 filter(!is.na(id)) %>%  
 distinct(id, .keep\_all = TRUE) %>%  
 select(-thing\_collected)

## 4.2. For edges

The code chunk below is used to rename source and target fields to from\_id and to\_id respectively, convert values in from\_id and to\_id fields to character data type, exclude values in from\_id and to\_id which not found in the id field of mc3\_nodes\_cleaned, exclude records whereby from\_id and/or to\_id values are missing, and save the cleaned tibble dataframe and called it mc3\_edges\_cleaned.

mc3\_edges\_cleaned <- mc3\_edges %>%  
 rename(from\_id = source,   
 to\_id = target) %>%  
 mutate(across(c(from\_id, to\_id),   
 as.character)) %>%  
 filter(from\_id %in% mc3\_nodes\_cleaned$id,   
 to\_id %in% mc3\_nodes\_cleaned$id) %>%  
 filter(!is.na(from\_id), !is.na(to\_id))

Next, create mapping of character id in mc3\_nodes\_cleaned to row index.

node\_index\_lookup <- mc3\_nodes\_cleaned %>%  
 mutate(.row\_id = row\_number()) %>%  
 select(id, .row\_id)

Next, join and convert from\_id and to\_id to integer indices. At the same time, drop rows with unmatched nodes.

mc3\_edges\_indexed <- mc3\_edges\_cleaned %>%  
 left\_join(node\_index\_lookup,   
 by = c("from\_id" = "id")) %>%  
 rename(from = .row\_id) %>%  
 left\_join(node\_index\_lookup,   
 by = c("to\_id" = "id")) %>%  
 rename(to = .row\_id) %>%  
 select(from, to, is\_inferred, type) %>%  
 filter(!is.na(from) & !is.na(to))

Next, subset nodes to only those referenced by edges.

used\_node\_indices <- sort(  
 unique(c(mc3\_edges\_indexed$from,   
 mc3\_edges\_indexed$to)))  
  
mc3\_nodes\_final <- mc3\_nodes\_cleaned %>%  
 slice(used\_node\_indices) %>%  
 mutate(new\_index = row\_number())

Then, rebuild lookup from old index to new index.

old\_to\_new\_index <- tibble(  
 old\_index = used\_node\_indices,  
 new\_index = seq\_along(  
 used\_node\_indices))

Finally, update edge indices to match new node table.

mc3\_edges\_final <- mc3\_edges\_indexed %>%  
 left\_join(old\_to\_new\_index,   
 by = c("from" = "old\_index")) %>%  
 rename(from\_new = new\_index) %>%  
 left\_join(old\_to\_new\_index,   
 by = c("to" = "old\_index")) %>%  
 rename(to\_new = new\_index) %>%  
 select(from = from\_new, to = to\_new,   
 is\_inferred, type)

# 5. Exploratory Data Analysis

## 5.1. Node type distribution

mc3\_nodes\_final %>%  
 count(type, sort = TRUE) %>%  
 ggplot(aes(x = reorder(type, n), y = n)) +  
 geom\_col(fill = "forestgreen") +  
 geom\_text(aes(label = n), hjust = -0.2) +  
 coord\_flip() +  
 labs(title = "Node Type Distribution",  
 x = "Node Type", y = "Count") +  
 theme\_minimal()

## 5.2. Sub-type Distribution for Entities

mc3\_nodes\_final %>%  
 filter(type == "Entity") %>%  
 count(sub\_type, sort = TRUE) %>%  
 ggplot(aes(x = reorder(sub\_type, n), y = n)) +  
 geom\_col(fill = "skyblue") +  
 geom\_text(aes(label = n), hjust = -0.2) +  
 coord\_flip() +  
 labs(title = "Entity Subtype Distribution",  
 x = "Entity Subtype", y = "Count") +  
 theme\_minimal()

## 5.3. Sub-type Distribution for Event

mc3\_nodes\_final %>%  
 filter(type == "Event") %>%  
 count(sub\_type, sort = TRUE) %>%  
 ggplot(aes(x = reorder(sub\_type, n), y = n)) +  
 geom\_col(fill = "grey") +  
 geom\_text(aes(label = n), hjust = -0.2) +  
 coord\_flip() +  
 labs(title = "Event Subtype Distribution",  
 x = "Event Subtype", y = "Count") +  
 theme\_minimal()

# 6. Visual Analytics

## 6.1. Relationship Analysis

This section focuses on exploring the relationships among involved individuals, organizations, and vessels.

### 6.1.1. Person-to-Person Relationships

To explore social and operational connections among individuals in the Oceanus community, a network visualization is constructed to represent person-to-person relationships mediated by relationship-type nodes. The process involves the following steps:

**Identifying Relationship Nodes**

Nodes labeled as “Relationship” are extracted from the cleaned graph data and split into two categories:

* Undirected relationships: Types like “Friends” and “Colleagues” which imply mutual connection.
* Directed relationships: All other types (e.g., “Manager”, “Employee”, “Superior”), which imply directional links.

**Constructing Undirected and Directed Person–Person Edges**

* For undirected relationships, a table is built to capture all people connected to the same relationship node. These are paired using self-joins to create mutual (bidirectional) person-to-person links.
* For directed relationships, the graph schema supports a two-step link: a person connects to a relationship node, which then connects to another person. These are joined to create directed edges between people, preserving the direction implied by the intermediate relationship node.

node\_lookup <- mc3\_nodes\_final %>% select(new\_index, sub\_type, name)  
edges\_labeled <- mc3\_edges\_final %>%  
 left\_join(node\_lookup, by = c("from" = "new\_index")) %>%  
 rename(from\_type = sub\_type, from\_name = name) %>%  
 left\_join(node\_lookup, by = c("to" = "new\_index")) %>%  
 rename(to\_type = sub\_type, to\_name = name)  
  
# Create non-directional relationships  
nondir\_rel\_ids <- mc3\_nodes\_final %>%  
 filter(type == "Relationship", sub\_type %in% c("Friends", "Colleagues")) %>%  
 pull(new\_index)  
  
# Get all Entity → Relationship edges (where Entity is a Person)  
nondir\_links <- edges\_labeled %>%  
 filter(from\_type == "Person", to %in% nondir\_rel\_ids) %>%  
 transmute(person = from, rel = to) %>%  
 distinct()  
  
# Pair up people connected to the same relationship  
pp\_rel\_pairs\_nondir <- nondir\_links %>%  
 inner\_join(nondir\_links, by = "rel") %>%  
 filter(person.x != person.y) %>%  
 distinct(rel, from = person.x, to = person.y)  
  
# Create directional relationships  
dir\_rel\_ids <- mc3\_nodes\_final %>%  
 filter(type == "Relationship", !sub\_type %in% c("Friends", "Colleagues")) %>%  
 pull(new\_index)  
  
# From person to relationship  
pr\_links <- edges\_labeled %>%  
 filter(from\_type == "Person", to %in% dir\_rel\_ids) %>%  
 transmute(person = from, rel = to)  
  
# From relationship to another person  
rp\_links <- edges\_labeled %>%  
 filter(from %in% dir\_rel\_ids, to\_type == "Person") %>%  
 transmute(rel = from, person = to)  
  
pp\_rel\_pairs\_dir <- inner\_join(pr\_links, rp\_links, by = "rel") %>%  
 filter(person.x != person.y) %>%  
 distinct(rel, from = person.x, to = person.y)

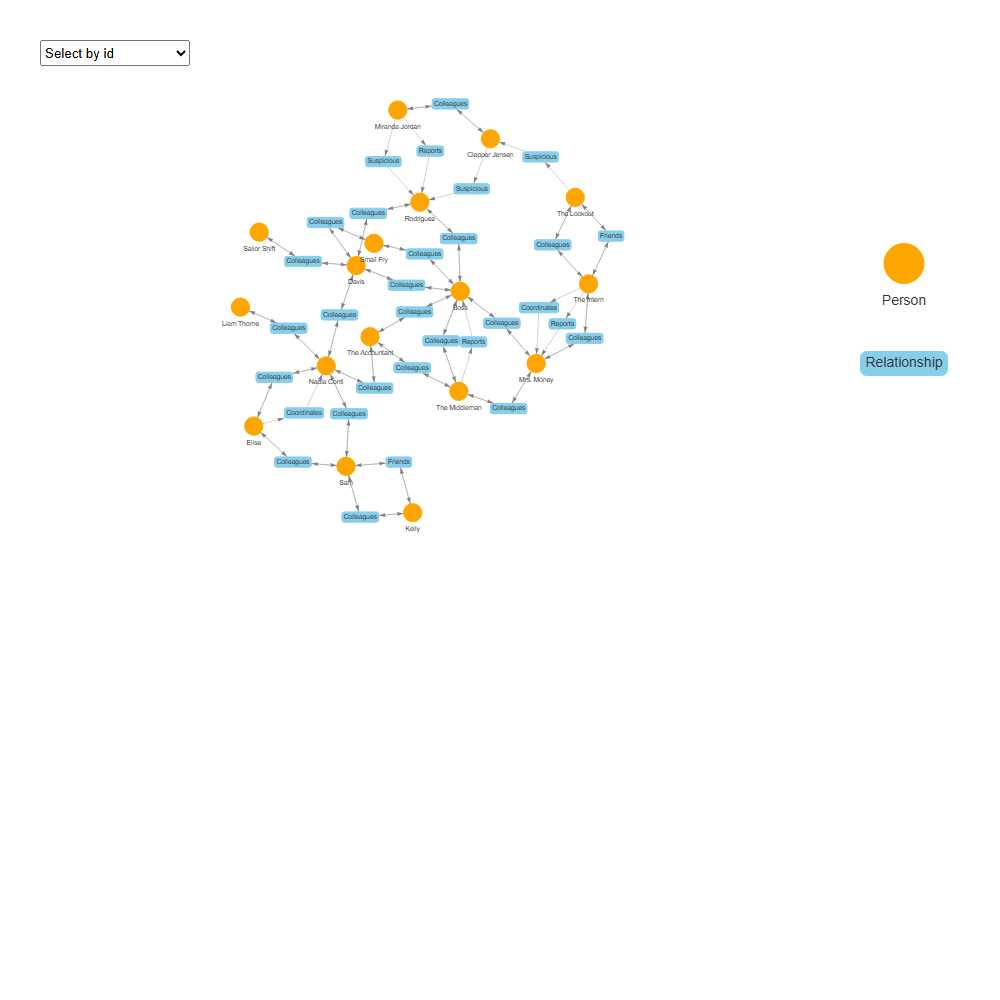
**Combining and Preparing Network Data**

* Both undirected and directed edges are combined into a unified edge list, annotated with a direction label.
* The relevant subset of nodes (people and relationships involved in these edges) is extracted and labeled with types and visual styles:
  + Persons were represented as orange dots.
  + Relationship nodes were represented as sky-blue boxes.

**Network Visualization**

* The graph’s rendered using the visNetwork package in R. Arrows indicate direction of relationships (where applicable).
* Node interaction features are added:
  + Hovering highlights connected nodes.
  + A drop-down menu allows selection of specific person nodes.
  + A legend was added to clarify node types and styling.

rel\_nodes <- mc3\_nodes\_final %>%  
 filter(type == "Relationship") %>%   
 select(new\_index, rel\_subtype = sub\_type)  
  
rel\_ids <- rel\_nodes$new\_index  
  
# Combine both types of relationship  
pp\_rel\_pairs\_nondir <- pp\_rel\_pairs\_nondir %>%  
 mutate(direction = "undirected")  
  
pp\_rel\_pairs\_dir <- pp\_rel\_pairs\_dir %>%  
 mutate(direction = "directed")  
  
pp\_rel\_pairs <- bind\_rows(pp\_rel\_pairs\_nondir, pp\_rel\_pairs\_dir)  
  
# Prepare network data  
used\_ids <- unique(c(pp\_rel\_pairs$from, pp\_rel\_pairs$to, pp\_rel\_pairs$rel))  
  
nodes\_tbl <- mc3\_nodes\_final %>%  
 filter(new\_index %in% used\_ids) %>%  
 mutate(type = if\_else(new\_index %in% rel\_ids, "Relationship", "Person")) %>%  
 select(node\_id = new\_index, label, type) %>%  
 mutate(row\_id = row\_number())  
  
id\_map <- nodes\_tbl %>% select(node\_id, row\_id)  
  
edges\_vis <- bind\_rows(  
 # Person → Relationship  
 pp\_rel\_pairs %>%  
 left\_join(id\_map, by = c("from" = "node\_id")) %>% rename(from\_id = row\_id) %>%  
 left\_join(id\_map, by = c("rel" = "node\_id")) %>% rename(to\_id = row\_id) %>%  
 transmute(from = from\_id, to = to\_id),  
  
 # Relationship → Person  
 pp\_rel\_pairs %>%  
 left\_join(id\_map, by = c("rel" = "node\_id")) %>% rename(from\_id = row\_id) %>%  
 left\_join(id\_map, by = c("to" = "node\_id")) %>% rename(to\_id = row\_id) %>%  
 transmute(from = from\_id, to = to\_id)  
)  
  
# Visualize  
vis\_nodes <- nodes\_tbl %>%  
 transmute(  
 id = row\_id,  
 label = label,  
 group = type,  
 shape = if\_else(type == "Person", "dot", "box"),  
 color = if\_else(type == "Person", "orange", "skyblue")  
 )  
  
vis\_edges <- edges\_vis %>%  
 distinct() %>%  
 mutate(arrows = "to", color = "gray")  
  
network1 <- visNetwork(vis\_nodes, vis\_edges) %>%  
 visNodes(font = list(size = 18)) %>%  
 visEdges(smooth = FALSE) %>%  
 visOptions(  
 highlightNearest = list(enabled = TRUE, degree = 2, hover = TRUE),  
   
 # restrict ID selection to Person nodes only  
 nodesIdSelection = list(  
 enabled = TRUE,  
 values = vis\_nodes %>% filter(group == "Person") %>% pull(id)  
 )  
 ) %>%  
 visLayout(randomSeed = 1234) %>%  
 visLegend(  
 addNodes = data.frame(  
 label = c("Person", "Relationship"),  
 shape = c("dot", "box"),  
 color = c("orange", "skyblue")  
 ),  
 useGroups = FALSE,  
 position = "right"  
 )  
  
img1 <- render\_visnetwork\_as\_image(network1, name = "network1")  
knitr::include\_graphics(img1)



To identify key individuals within the communication network, I compute four standard network centrality metrics for all Person nodes:

* **Degree centrality**: Measures the number of direct connections a node has, indicating activity level.
* **Betweenness centrality**: Quantifies how often a node lies on the shortest paths between others, identifying information brokers or gatekeepers.
* **Closeness centrality**: Reflects how quickly a node can reach others in the network, showing communication efficiency.
* **Eigenvector centrality**: Captures influence based on being connected to other highly connected nodes.

The analysis is performed using the igraph package in R, with the full graph including both Person and Relationship nodes. Centrality metrics are calculated on the full graph structure, but filtered to show results only for Person nodes.

# Step 1: Create igraph with all nodes and edges  
igraph\_pp <- graph\_from\_data\_frame(  
 d = edges\_vis,  
 vertices = nodes\_tbl %>% select(name = row\_id, label, type),  
 directed = TRUE  
)  
  
# Step 2: Compute centrality metrics  
V(igraph\_pp)$degree <- degree(igraph\_pp, mode = "all")  
V(igraph\_pp)$betweenness <- betweenness(igraph\_pp, directed = TRUE)  
V(igraph\_pp)$closeness <- closeness(igraph\_pp, normalized = TRUE)  
V(igraph\_pp)$eigenvector <- eigen\_centrality(igraph\_pp, directed = TRUE)$vector  
  
# Step 3: Create centrality table and filter for only Person nodes  
centrality\_tbl <- tibble(  
 id = as.integer(V(igraph\_pp)$name), # row\_id  
 name = V(igraph\_pp)$label,  
 type = V(igraph\_pp)$type,  
 degree = V(igraph\_pp)$degree,  
 betweenness = V(igraph\_pp)$betweenness,  
 closeness = V(igraph\_pp)$closeness,  
 eigenvector = V(igraph\_pp)$eigenvector  
) %>%  
 filter(type == "Person") %>%  
 arrange(desc(degree))  
  
print(centrality\_tbl)

# A tibble: 17 × 7  
 id name type degree betweenness closeness eigenvector  
 <int> <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
 1 16 Boss Person 13 938. 0.217 1   
 2 3 Nadia Conti Person 11 876. 0.181 0.378   
 3 6 Davis Person 10 692. 0.202 0.596   
 4 1 Sam Person 8 301. 0.143 0.125   
 5 11 The Intern Person 8 580. 0.166 0.0599   
 6 14 Mrs. Money Person 8 776. 0.195 0.279   
 7 7 Rodriguez Person 7 362. 0.175 0.254   
 8 15 The Middleman Person 7 271. 0.198 0.301   
 9 13 The Accountant Person 6 418. 0.193 0.317   
10 4 Elise Person 5 98 0.141 0.0799   
11 12 The Lookout Person 5 312. 0.157 0.0190   
12 2 Kelly Person 4 1 0.113 0.0397   
13 9 Clepper Jensen Person 4 310. 0.160 0.00265   
14 10 Miranda Jordan Person 4 120 0.162 0.000363  
15 17 Small Fry Person 4 34.7 0.175 0.253   
16 5 Liam Thorne Person 2 0 0.135 0.0518   
17 8 Sailor Shift Person 2 0 0.146 0.0816

**Insights**: The centrality analysis and network visualization reveal that Boss is the most influential individual in the network, with the highest degree, betweenness, and eigenvector centrality, indicating strong connectivity and a central role in information flow. Nadia Conti and Davis also emerge as key connectors, acting as bridges between subgroups. Despite fewer direct links, The Intern and Mrs. Money exhibit high betweenness, suggesting they serve as strategic intermediaries.

### 6.1.2. Person-Vessel Relationships

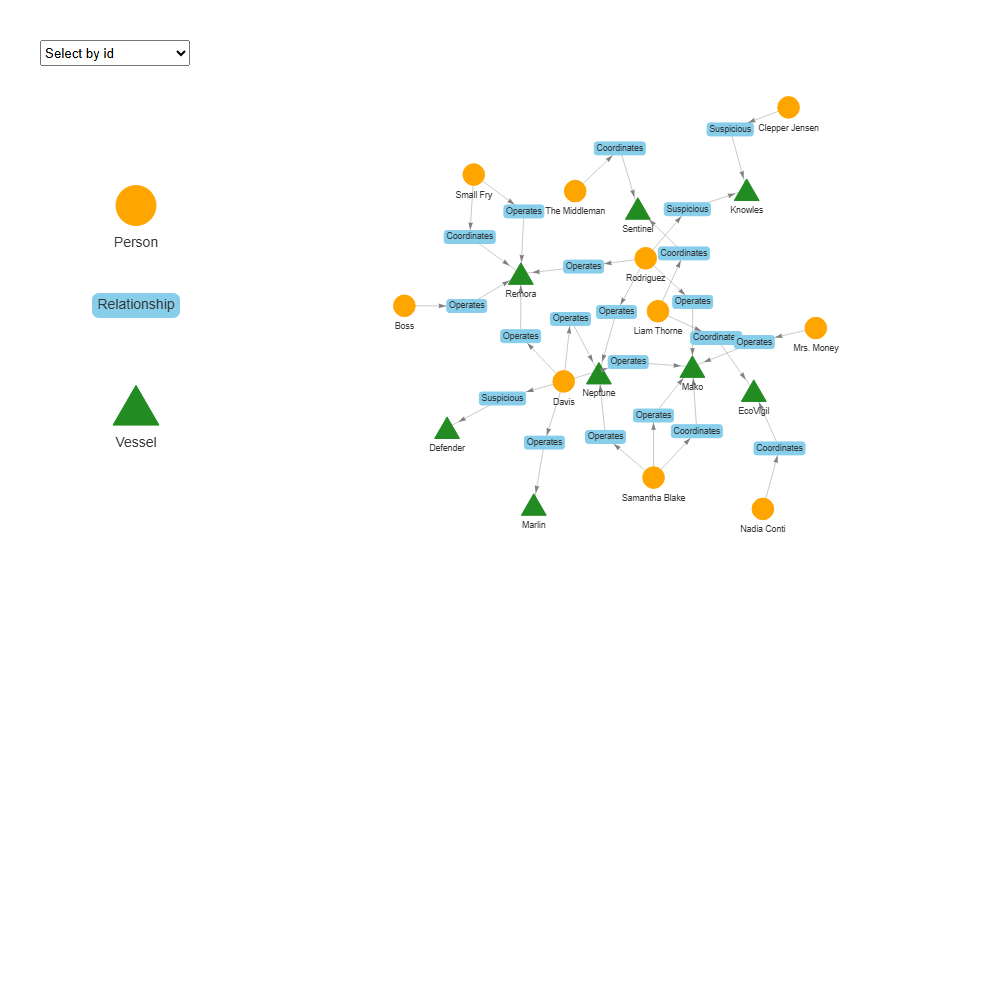
To explore how individuals are connected to vessels, a tripartite network is constructed. The process began by identifying all relationship nodes and linking them to person nodes (Person → Relationship) and vessel nodes (Relationship → Vessel) using the edge table. These two-step connections are joined to form person–relationship–vessel triplets, capturing associations between people and vessels.

All involved nodes are extracted and categorized as Person, Relationship, or Vessel, with each assigned a unique row ID. These are used to build a node table (vis\_nodes) with customized shapes and colors, and an edge table (vis\_edges) to represent directional links.

The resulting network is visualized using the visNetwork package in R, where:

* Persons appear as orange dots,
* Relationships as sky-blue boxes,
* Vessels as green triangles.
* Interactive features allow for node highlighting and filtering, enabling intuitive exploration of how individuals may influence or interact with vessels via intermediary relationship types.

# Construct person–relationship–vessel triplet  
pr\_edges <- edges\_labeled %>%  
 filter(from\_type == "Person", to %in% rel\_ids) %>%  
 transmute(person\_id = from,   
 relationship\_id = to)  
  
vessel\_ids <- mc3\_nodes\_final %>%  
 filter(sub\_type == "Vessel") %>% pull(new\_index)  
  
rv\_edges <- edges\_labeled %>%  
 filter(from %in% rel\_ids, to %in% vessel\_ids) %>%  
 transmute(relationship\_id = from,  
 vessel\_id = to)  
  
triplets <- inner\_join(pr\_edges, rv\_edges, by = "relationship\_id")  
  
all\_ids <- unique(c(triplets$person\_id,  
 triplets$relationship\_id,  
 triplets$vessel\_id))  
  
# Node table  
rel\_tbl <- mc3\_nodes\_final %>%  
 filter(new\_index %in% all\_ids) %>%  
 mutate(type = case\_when(  
 new\_index %in% vessel\_ids ~ "Vessel",  
 new\_index %in% rel\_ids ~ "Relationship",  
 TRUE ~ "Person"  
 )) %>%  
 select(index = new\_index, label, type) %>%  
 distinct() %>%  
 arrange(index) %>%  
 mutate(row\_id = row\_number())   
  
id\_map <- rel\_tbl %>% select(index, row\_id)  
  
# Edge table  
edges\_tbl <- bind\_rows(  
 triplets %>%  
 select(from = person\_id, to = relationship\_id),  
 triplets %>%  
 select(from = relationship\_id, to = vessel\_id)  
) %>%  
 left\_join(id\_map, by = c("from" = "index")) %>%  
 rename(.from = row\_id) %>%  
 left\_join(id\_map, by = c("to" = "index")) %>%  
 rename(.to = row\_id) %>%  
 filter(!is.na(.from) & !is.na(.to)) %>%  
 transmute(from = .from, to = .to)  
  
# Visualize  
vis\_nodes <- rel\_tbl %>%  
 transmute(  
 id = row\_id,  
 label = label,  
 group = type,  
 shape = case\_when(  
 type == "Person" ~ "dot",  
 type == "Relationship"~ "box",  
 type == "Vessel" ~ "triangle"  
 ),  
 color = case\_when(  
 type == "Person" ~ "orange",  
 type == "Relationship"~ "skyblue",  
 type == "Vessel" ~ "forestgreen"  
 )  
 )  
  
vis\_edges <- edges\_tbl %>%  
 transmute(  
 from = from,  
 to = to,  
 arrows = "to",  
 color = "gray"  
 )  
  
network2 <- visNetwork(vis\_nodes, vis\_edges) %>%  
 visNodes(font = list(size = 20)) %>%  
 visEdges(smooth = FALSE) %>%  
 visOptions(highlightNearest = list(enabled = TRUE, degree = 2, hover = TRUE), nodesIdSelection = TRUE) %>%  
 visLayout(randomSeed = 123) %>%  
 visLegend(addNodes = data.frame(  
 label = c("Person", "Relationship", "Vessel"),  
 shape = c("dot", "box", "triangle"),  
 color = c("orange", "skyblue", "forestgreen")  
 ), useGroups = FALSE)  
  
img2 <- render\_visnetwork\_as\_image(network2, name = "network2")  
knitr::include\_graphics(img2)



**Insights**: The Person–Relationship–Vessel network visualization reveals a tightly interconnected operational structure centered around key individuals. Davis and Rodriguez emerge as central figures, each linked to multiple vessels through “Operates” and “Coordinates” relationships, suggesting they play pivotal roles in managing or overseeing maritime activities. Vessels such as Remora, Sentinel, Neptune, and Mako form the core of this network, with multiple people connected to each—indicative of shared usage and coordinated operations. While most links are operational, a few “Suspicious” relationships—such as those involving Defender and Knowles—highlight potential vessels of concern. Notably, individuals like Boss, Clepper Jensen, and Nadia Conti are connected to only one vessel or relationship, indicating they may be responsible for more high-level or strategic roles. Overall, the structure suggests a coordinated network with central command figures, key vessels, and isolated links that may warrant further investigation.

## 6.2. Interaction Analysis

This analysis focuses solely on communication events between entities to identify patterns of frequent interaction.

### 6.2.1. Person-Person Communication

The process begins by identifying all edges linking Persons and Communication events. From this, all possible pairs of people who appear in the same communication event are extracted. To avoid duplication and self-pairs, only distinct pairs with different individuals are retained.

The number of times each person-pair co-occurs in communication events is then counted, forming a weighted edge list (num\_comm\_events). To standardize the order of pairs, names are alphabetically sorted within each row.

These person pairs are used to build a graph where:

* Nodes represent individuals,
* Edges represent the frequency of shared communication events,
* Edge width is scaled by the number of co-occurrences.

The resulting network is visualized using ggraph, providing insight into the structure of interpersonal communication patterns.

pp\_comm <- edges\_labeled %>%  
 filter((from\_type == "Person" & to\_type == "Communication") |  
 (from\_type == "Communication" & to\_type == "Person")) %>%  
 transmute(person = if\_else(from\_type == "Person", from\_name, to\_name),  
 comm\_id = if\_else(from\_type == "Communication", from, to))   
  
person\_pairs <- pp\_comm %>%  
 inner\_join(pp\_comm, by = "comm\_id") %>%  
 filter(person.x != person.y) %>%  
 count(person.x, person.y, name = "num\_comm\_events") %>% # ← new column name  
 rowwise() %>%  
 mutate(p1 = min(person.x, person.y),  
 p2 = max(person.x, person.y)) %>%  
 ungroup() %>%  
 select(from = p1, to = p2, num\_comm\_events) %>%  
 distinct()  
  
person\_pairs %>%  
 arrange(desc(num\_comm\_events)) %>%  
 print(n = 10)

# A tibble: 22 × 3  
 from to num\_comm\_events  
 <chr> <chr> <int>  
 1 Clepper Jensen Miranda Jordan 38  
 2 Mrs. Money The Intern 20  
 3 The Intern The Lookout 20  
 4 Boss Mrs. Money 10  
 5 Sam The Lookout 8  
 6 Boss The Middleman 7  
 7 Mrs. Money The Middleman 6  
 8 Davis Nadia Conti 5  
 9 Liam Thorne Nadia Conti 4  
10 The Accountant The Middleman 4  
# ℹ 12 more rows

pp\_nodes <- tibble(name = unique(c(person\_pairs$from, person\_pairs$to)))  
  
id\_lkp <- pp\_nodes %>% mutate(row\_id = row\_number())  
  
pp\_edges\_final <- person\_pairs %>%  
 left\_join(id\_lkp %>% rename(from\_id = row\_id), by = c("from" = "name")) %>%  
 left\_join(id\_lkp %>% rename(to\_id = row\_id), by = c("to" = "name")) %>%  
 select(from = from\_id, to = to\_id, num\_comm\_events)  
  
set.seed(42)  
ggraph(tbl\_graph(nodes = pp\_nodes, edges = pp\_edges\_final, directed = FALSE),  
 layout = "fr") +  
 geom\_edge\_link(aes(width = num\_comm\_events), alpha = 0.5, color = "darkgrey") +  
 geom\_node\_point(color = "orange", size = 4) +  
 geom\_node\_text(aes(label = name), repel = TRUE, size = 3) +  
 scale\_edge\_width(range = c(0.3, 2.5)) +  
 theme\_void() +  
 labs(title = "Person–Person Interactions")

**Insights**: The person-to-person communication network reveals several high-frequency interaction pairs, with Clepper Jensen and Miranda Jordan—both mentioned in the case as investigators—sharing the most communication events (38). A secondary cluster features Mrs. Money, The Intern, and The Lookout, who are tightly linked through repeated communications, potentially indicating a shared operational or covert role. Boss emerges as a central figure, engaging with both Mrs. Money and The Middleman, and likely playing a coordinating or supervisory role. Additional patterns show Nadia Conti frequently contacted by both Davis and Liam Thorne, while The Middleman serves as a common contact for multiple individuals.

### 6.2.2. Person-Organization-Group Interactions

This analysis constructs a network to explore how people, organizations, and groups interact based on their co-occurrence in communication events.

The process begins by identifying all edges that connect an entity (either a Person, Organization, or Group) to a Communication event. Both directions are considered to capture all valid interactions.

Next, for each communication event, all unique pairs of co-occurring entities are generated. The number of shared communication events between each pair is counted, forming a weighted edge list.

Each unique entity is recorded as a node and labeled with its type (Person, Organization, or Group). A numeric ID is assigned to each node for graph construction. The entity pairs and node list are joined to create a graph-compatible edge table with numeric references.

The resulting network is visualized using ggraph, where:

* Edge width reflects the frequency of shared communications.
* Node color and shape represent entity type: orange circles for Persons, purple diamonds for Organizations, and brown squares for Groups.
* The Fruchterman–Reingold layout emphasizes structural proximity between frequently co-occurring entities.

# Step 1: Get all Entity–Communication edges (Person, Organization, Group)  
entity\_comm <- edges\_labeled %>%  
 filter((from\_type %in% c("Person", "Organization", "Group") & to\_type == "Communication") |  
 (from\_type == "Communication" & to\_type %in% c("Person", "Organization", "Group"))) %>%  
 transmute(  
 entity = if\_else(from\_type %in% c("Person", "Organization", "Group"), from\_name, to\_name),  
 comm\_id = if\_else(from\_type == "Communication", from, to),  
 entity\_type = if\_else(from\_type %in% c("Person", "Organization", "Group"), from\_type, to\_type)  
 )  
  
# Step 2: Create all entity–entity pairs that co-occur in the same communication event  
entity\_pairs <- entity\_comm %>%  
 inner\_join(entity\_comm, by = "comm\_id") %>%  
 filter(entity.x != entity.y) %>%  
 count(entity.x, entity.y, name = "num\_comm\_events") %>%  
 rowwise() %>%  
 mutate(p1 = min(entity.x, entity.y),  
 p2 = max(entity.x, entity.y)) %>%  
 ungroup() %>%  
 select(from = p1, to = p2, num\_comm\_events) %>%  
 distinct()  
  
# Step 3: Create node table with type info  
node\_types <- entity\_comm %>%  
 select(name = entity, sub\_type = entity\_type) %>%  
 distinct()  
  
network\_nodes <- entity\_comm %>%   
 distinct(name = entity, sub\_type = entity\_type) %>%  
 arrange(name) %>%   
 mutate(row\_id = row\_number())   
  
id\_lkp <- network\_nodes %>% select(name, row\_id)  
  
# Step 4: Build edge table with numeric IDs  
network\_edges <- entity\_pairs %>%  
 left\_join(id\_lkp %>% rename(from\_id = row\_id), by = c("from" = "name")) %>%  
 left\_join(id\_lkp %>% rename(to\_id = row\_id), by = c("to" = "name")) %>%  
 select(from = from\_id, to = to\_id, num\_comm\_events)  
  
# Step 5: Plot the network with ggraph  
set.seed(42)  
ggraph(tbl\_graph(nodes = network\_nodes, edges = network\_edges, directed = FALSE),  
 layout = "fr") +  
 geom\_edge\_link(aes(width = num\_comm\_events), alpha = 0.5, color = "grey60") +  
 geom\_node\_point(aes(color = sub\_type, shape = sub\_type), size = 4) +  
 geom\_node\_text(aes(label = name), repel = TRUE, size = 3) +  
 scale\_edge\_width(range = c(0.3, 3)) +  
 scale\_color\_manual(values = c(  
 Person = "orange",  
 Organization = "pink",  
 Group = "brown"  
 )) +  
 scale\_shape\_manual(values = c(  
 Person = 19,  
 Organization = 18,  
 Group = 15  
 )) +  
 theme\_void() +  
 labs(title = "Person-Organization-Group Interactions")

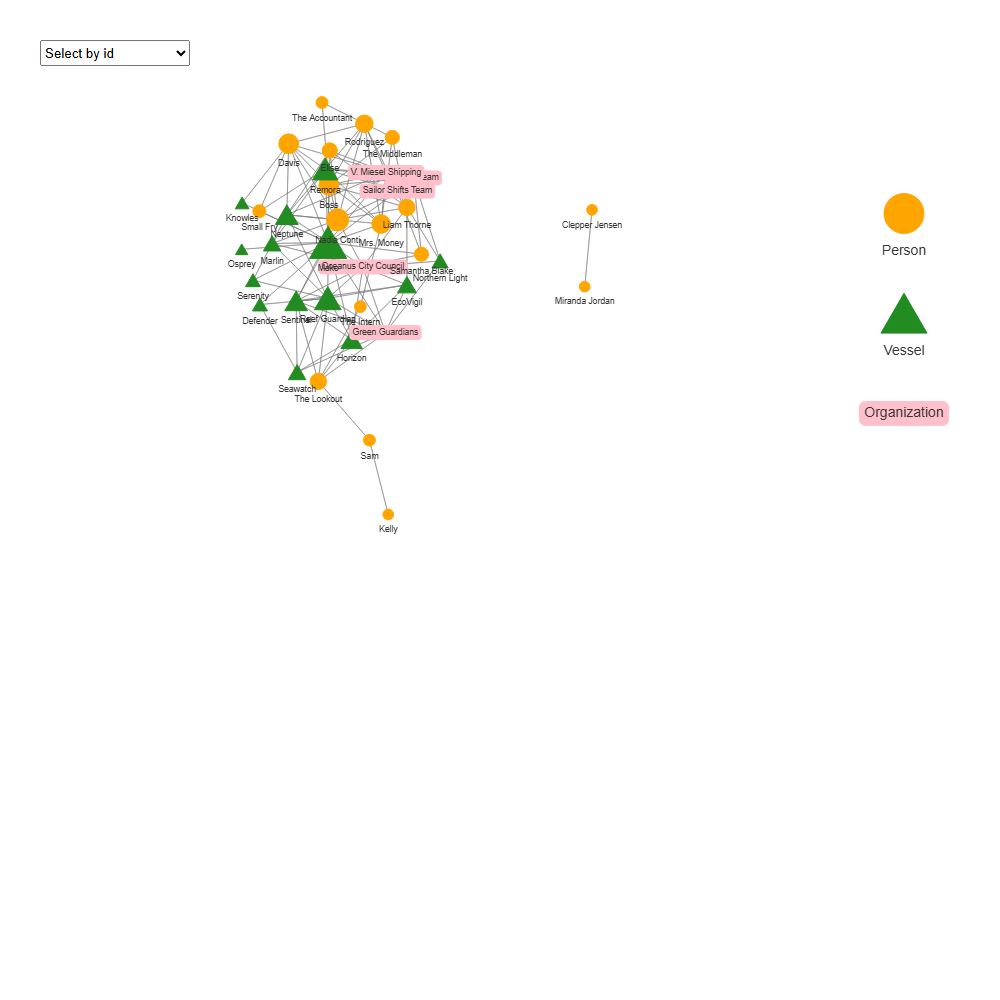
**Insights**: The network reveals a densely connected core of individuals and organizations, suggesting coordinated activity across multiple entities. V-Miesel Shipping appears central, linked to individuals such as Boss, Mrs. Money, Rodriguez, and The Middleman, indicating a potential hub for operational or logistical coordination. Oceanus City Council and Green Guardians also serve as prominent intermediaries, connecting civic actors like Samantha Blake, Liam Thorne, and The Lookout. Nadia Conti stands out as a central bridge across these clusters, linking both activist-aligned and logistics-aligned individuals.

A notable observation is the presence of only one Group node — “City Officials” — which is completely disconnected from the rest of the network. This suggests that, although the entity exists in the data, it did not participate in any communication event involving the individuals or organizations present in this chart.

### 6.2.3. Person-Organzation-Vessel Interactions

In this stage, an interactive network visualization is constructed to explore communication-based relationships among people, vessels, and organizations. The process begins with extracting communication links between each entity type and communication events. From these, co-occurrence pairs are formed—entities that appeared together in the same communication event—counted, and converted into a weighted edge list. A node table is created with entity names and types. Using tidygraph, a network graph object is built, and centrality metrics (degree, betweenness, closeness, eigenvector) are computed to assess influence and positioning within the network. Finally, visNetwork is used to generate an interactive plot where node size reflects degree centrality, tooltip text displays all metrics, and nodes are colored and shaped based on their type (person, vessel, or organization). This allows for intuitive exploration of key actors and communication patterns.

# Build Communication tables  
pc <- edges\_labeled %>%  
 filter((from\_type == "Person" & to\_type == "Communication") |  
 (from\_type == "Communication" & to\_type == "Person")) %>%  
 transmute(entity = if\_else(from\_type == "Person", from\_name, to\_name),  
 comm = if\_else(from\_type == "Person", to, from),  
 type = "Person")  
  
vc <- edges\_labeled %>%  
 filter((from\_type == "Vessel" & to\_type == "Communication") |  
 (from\_type == "Communication" & to\_type == "Vessel")) %>%  
 transmute(entity = if\_else(from\_type == "Vessel", from\_name, to\_name),  
 comm = if\_else(from\_type == "Vessel", to, from),  
 type = "Vessel")  
  
oc <- edges\_labeled %>%  
 filter((from\_type == "Organization" & to\_type == "Communication") |  
 (from\_type == "Communication" & to\_type == "Organization")) %>%  
 transmute(entity = if\_else(from\_type == "Organization", from\_name, to\_name),  
 comm = if\_else(from\_type == "Organization", to, from),  
 type = "Organization")  
  
entity\_comm <- bind\_rows(pc, vc, oc)  
  
# Pairwise co-occurrences  
edges\_raw <- entity\_comm %>%  
 inner\_join(entity\_comm, by = "comm", suffix = c("\_a", "\_b")) %>%  
 filter(entity\_a != entity\_b) %>%  
 mutate(pair\_id = map2\_chr(entity\_a, entity\_b, ~ paste(sort(c(.x, .y)), collapse = "||"))) %>%  
 count(pair\_id, entity\_a, entity\_b, name = "weight")  
  
# Node table  
nodes\_tbl <- entity\_comm %>%  
 distinct(name = entity, sub\_type = type) %>%  
 mutate(row\_id = row\_number())  
  
# Edge table  
id\_lkp <- nodes\_tbl %>% select(name, row\_id)  
edges\_final <- edges\_raw %>%  
 left\_join(id\_lkp, by = c("entity\_a" = "name")) %>% rename(from = row\_id) %>%  
 left\_join(id\_lkp, by = c("entity\_b" = "name")) %>% rename(to = row\_id) %>%  
 select(from, to, weight)  
  
full\_graph <- tbl\_graph(nodes = nodes\_tbl, edges = edges\_final, directed = FALSE) %>%  
 mutate(  
 degree = centrality\_degree(),  
 betweenness = centrality\_betweenness(),  
 closeness = centrality\_closeness(),  
 eigen = centrality\_eigen()  
 )  
  
# Visualize  
vis\_nodes <- full\_graph %>%  
 as\_tibble() %>%  
 mutate(  
 id = row\_number(),  
 label = name,  
 value = scales::rescale(degree, to = c(5, 25)), # Size by degree  
 group = sub\_type,  
 color = case\_when(  
 sub\_type == "Person" ~ "orange",  
 sub\_type == "Vessel" ~ "forestgreen",  
 sub\_type == "Organization" ~ "pink",  
 TRUE ~ "lightgray"  
 ),  
 shape = case\_when(  
 sub\_type == "Person" ~ "dot",  
 sub\_type == "Vessel" ~ "triangle",  
 sub\_type == "Organization" ~ "box",  
 TRUE ~ "ellipse"  
 ),  
 title = paste0(  
 "<b>", name, "</b><br>",  
 "Degree: ", degree, "<br>",  
 "Betweenness: ", round(betweenness, 2), "<br>",  
 "Closeness: ", round(closeness, 3), "<br>",  
 "Eigenvector: ", round(eigen, 3)  
 )  
 ) %>%  
 select(id, label, group, shape, color, value, title)  
  
vis\_edges <- edges\_final %>%  
 mutate(arrows = "none", color = "gray") %>%  
 distinct()  
  
network3 <- visNetwork(vis\_nodes, vis\_edges) %>%  
 visNodes(font = list(size = 16)) %>%  
 visEdges(smooth = FALSE) %>%  
 visOptions(  
 highlightNearest = list(enabled = TRUE, degree = 1, hover = TRUE),  
 nodesIdSelection = TRUE  
 ) %>%  
 visLayout(randomSeed = 42) %>%  
 visLegend(addNodes = data.frame(  
 label = c("Person", "Vessel", "Organization"),  
 shape = c("dot", "triangle", "box"),  
 color = c("orange", "forestgreen", "pink")  
 ), useGroups = FALSE, position = "right")  
  
img3 <- render\_visnetwork\_as\_image(network3, name = "network3")  
knitr::include\_graphics(img3)



**Insights**: This network map illustrates the complex web of communication and co-occurrence among individuals, vessels, and organizations. At the center is a densely connected cluster revolving around vessels such as Mako, Remora, Reef Guardian, and Neptune, suggesting these are frequent topics of discussion.

Highly connected individuals like Nadia Conti, Boss, Davis, and Mrs. Money, along with key organizations such as the Oceanus City Council and Green Guardians, play prominent roles—consistent with their high degree and betweenness centrality observed earlier. In contrast, individuals like Sam, Kelly, and The Lookout appear on the network’s periphery, linked to the core through only one or two connections, indicating more isolated or limited involvement.

## 6.3. Clustering Analysis

To better understand the structure of interactions, I conducted two clustering analyses based on communication frequency. The first focused on Person–Organization communication, revealing how individuals cluster around formal entities they frequently communicate with. The second included Vessels to capture broader patterns, highlighting how people and organizations group around frequently discussed or engaged vessels. By comparing both, we can distinguish between clusters formed through institutional communication and those shaped by shared operational or topical focus.

### 6.3.1. Person-Organization Clustering

This analysis aims to identify clusters of people and organizations that frequently co-occur in communication events. The process begins by filtering the edges to retain only those pairs that have interacted at least twice, ensuring meaningful connections. A graph is then constructed and restricted to its largest connected component to focus on the primary network structure. Centrality metrics (degree, betweenness, eigenvector) are computed to assess node influence. Louvain clustering is applied using edge weights (number of shared communication events) to detect densely connected communities. The resulting network is visualized with node size indicating degree centrality and color representing cluster membership.

# Step 1: Filter edges with sufficient communication events  
network\_filtered <- network\_edges %>%  
 filter(num\_comm\_events >= 2)  
  
# Step 2: Create graph  
pp\_graph <- tbl\_graph(nodes = network\_nodes,  
 edges = network\_filtered,  
 directed = FALSE) %>%  
 activate(edges) %>%  
 mutate(weight = num\_comm\_events)  
  
# Step 3: Filter to largest connected component  
pp\_graph <- pp\_graph %>%  
 activate(nodes) %>%  
 filter(!node\_is\_isolated()) %>%  
 mutate(comp = group\_components()) %>%  
 filter(comp == which.max(table(comp)))  
  
# Step 4: Compute centrality + clustering  
pp\_graph <- pp\_graph %>%  
 activate(nodes) %>%  
 mutate(  
 cluster = as\_factor(group\_louvain(weights = weight)),  
 deg = centrality\_degree(),  
 betw = centrality\_betweenness(),  
 eig = centrality\_eigen()  
 )  
  
# Step 5: Plot with different shapes for Person vs Organization  
set.seed(1234)  
ggraph(pp\_graph, layout = "fr") +  
 geom\_edge\_link(aes(width = weight), colour = "grey70", alpha = 0.4) +  
 geom\_node\_point(aes(colour = cluster, size = deg, shape = sub\_type)) +  
 geom\_node\_text(aes(label = name, colour = cluster), repel = TRUE, size = 3) +  
 scale\_shape\_manual(values = c(  
 Person = 16, # circle  
 Organization = 15 # square  
 )) +  
 scale\_edge\_width(range = c(0.2, 2.5)) +  
 theme\_void() +  
 labs(title = "People–Organization Clustering (Louvain method)")

**Insights**: The Louvain clustering reveals four distinct communities of people and organizations based on shared communication events:

* Cluster 1 (Red): Comprising Nadia Conti, Davis, Elise, and V. Miesel Shipping, this group appears to be closely connected, potentially involved in operational or logistical discussions.
* Cluster 2 (Green): Includes Oceanus City Council, Green Guardians, Samantha Blake, and Sailor Shifts Team, suggesting a coalition focused on environmental or public affairs.
* Cluster 3 (Blue): Formed by The Lookout, The Intern, Sam, and Kelly, this cluster is more peripheral but tightly knit—possibly playing an observational or low-influence role.
* Cluster 4 (Purple): Consisting of Boss, Mrs. Money, The Middleman, and The Accountant, this group may represent organizational or administrative coordination within the broader network.

Notably, Liam Thorne and Mrs. Money serve as key connectors, bridging multiple clusters and facilitating cross-group communication. In terms of centrality, The Intern, Mrs. Money, and Nadia Conti stand out as the most well-connected individuals, indicating their critical roles in the network’s information flow.

### 6.3.2. Person-Organization-Vessel Clustering

The second network graph is constructed to uncover clusters of people, organizations, and vessels that frequently co-occur in communication events. First, only strong connections—those with five or more shared communication events are retained to ensure meaningful interactions. A graph is then built and filtered to include only the largest connected component, focusing the analysis on the most relevant part of the network. Louvain clustering is applied using edge weights to detect community structures, while centrality metrics (degree, betweenness, eigenvector) are computed to identify influential entities. The resulting network is visualized with node size representing degree centrality and color indicating cluster membership, revealing key actors and tightly-knit groups in the communication ecosystem.

# Step 1: Filter edges with sufficient communication events  
edges\_filtered <- edges\_final %>%  
 filter(weight >= 5)  
  
# Step 2: Create graph  
entity\_graph <- tbl\_graph(nodes = nodes\_tbl,  
 edges = edges\_filtered,  
 directed = FALSE) %>%  
 activate(edges) %>%  
 mutate(weight = weight)  
  
# Step 3: Filter to largest connected component  
entity\_graph <- entity\_graph %>%  
 activate(nodes) %>%  
 filter(!node\_is\_isolated()) %>%  
 mutate(comp = group\_components()) %>%  
 filter(comp == which.max(table(comp)))  
  
# Step 4: Compute centrality + clustering  
entity\_graph <- entity\_graph %>%  
 activate(nodes) %>%  
 mutate(  
 cluster = as\_factor(group\_louvain(weights = weight)),  
 deg = centrality\_degree(),  
 betw = centrality\_betweenness(),  
 eig = centrality\_eigen()  
 )  
  
# Step 5: Plot with shape by sub\_type  
set.seed(1234)  
ggraph(entity\_graph, layout = "fr") +  
 geom\_edge\_link(aes(width = weight), colour = "grey70", alpha = 0.4) +  
 geom\_node\_point(aes(colour = cluster, size = deg, shape = sub\_type)) +  
 geom\_node\_text(aes(label = name, colour = cluster), repel = TRUE, size = 3) +  
 scale\_shape\_manual(values = c(  
 Person = 16, # circle  
 Organization = 15, # square  
 Vessel = 17 # triangle  
 )) +  
 scale\_edge\_width(range = c(0.2, 2.5)) +  
 theme\_void() +  
 labs(title = "People–Organization–Vessel Clustering (Louvain method)")

**Insights**: The inclusion of vessels reveals distinct communication clusters centered around frequently mentioned ships and their associated entities.

* Cluster 1 (Red) is dominated by vessels like Mako, Remora, Neptune, and V. Miesel Shipping, alongside individuals like Davis and Nadia Conti. This suggests a group engaged in operational or shipping-related discussions.
* Cluster 2 (Green) consists entirely of individuals including Mrs. Money, The Intern, The Middleman, and Boss—likely representing a coordinated team, possibly administrative or managerial.
* Cluster 3 (Blue) centers on Green Guardians, Oceanus City Council, and vessels like Reef Guardian, indicating an environmental or civic coalition involving both institutions and ships.
* Cluster 4 (Purple) consists entirely of 3 vessels Sentinel, Horizon, and EcoVigil. This suggests the cluster represents a group of vessels that are frequently discussed together or are involved in similar communication contexts.