Global Logistics

Routing deliveries through coordination of shipping companies

Agents and Multi-Agent Systems

FEUP-M.EIC009-2022/2023-2S

Group B
Diogo Costa - up201906731
Francisco Colino - up201905405
Henrique Sousa - up201906681

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Problem

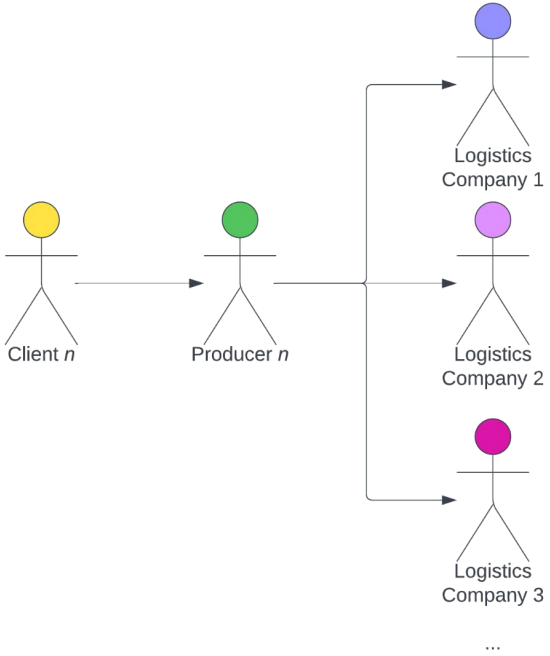
In the globalized world we live in, logistics is a central part of it. There are **global container ports** and **logistic companies** that operate them and use them with their cargo ships. In parallel, there are also **long land** routes and **short** ones following the same logic. Some logistics companies operate at several routes, and others just at some.

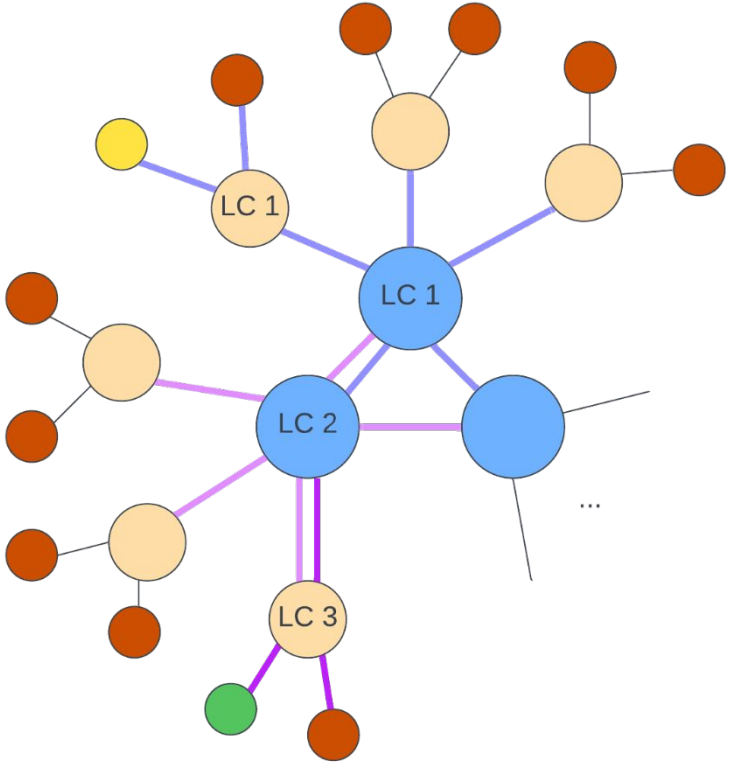
There are **producer companies** that want to transport their products from an origin to a destination. These companies will sell at **auction** to the logistics companies the transportation of their products, which may involve several types of routes (**international**, **regional**, **neighborhood**).

Variables

- Dependent:
  - Price
  - Request time
  - Route length
  - Holding time
  - Idle time
  - Travel time
  - Unique companies
- Independent:
  - No. Clients
  - No. Companies
  - No. Cities
  - No. Houses
  - No. Regional Hubs
  - No. Global Hubs
  - No. Ships
  - No. Semi Trucks
  - No. Vans

Agents

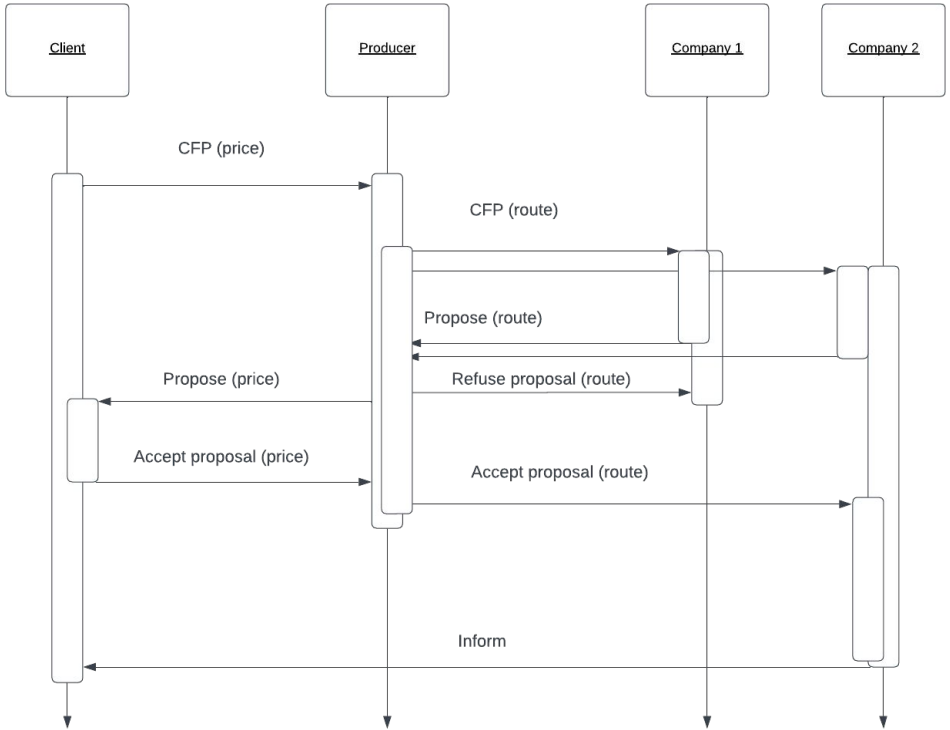

Client *n*
Producer *n*
Logistics Company 1
Logistics Company 2
Logistics Company 3
...


LC 1
LC 1
LC 2
LC 3
...

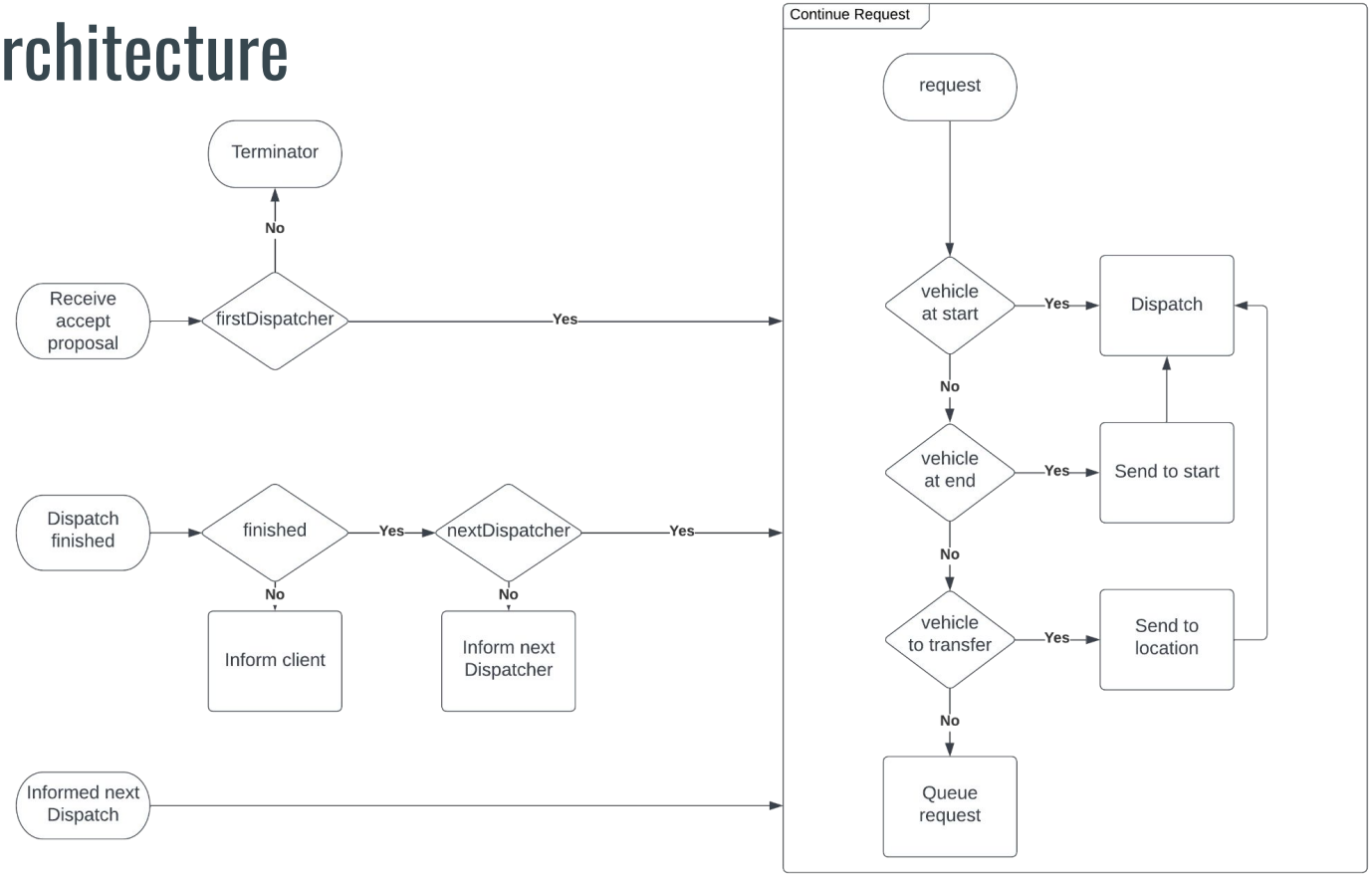
Communication

The agents have a set of behaviors to handle the high-level interactions

- Client:
  - Contract Net Initiator
- Producer:
  - SSResponderDispatcher
  - Contract Net Initiatior
- Company
  - SSResponderDispatcher



Agent architecture

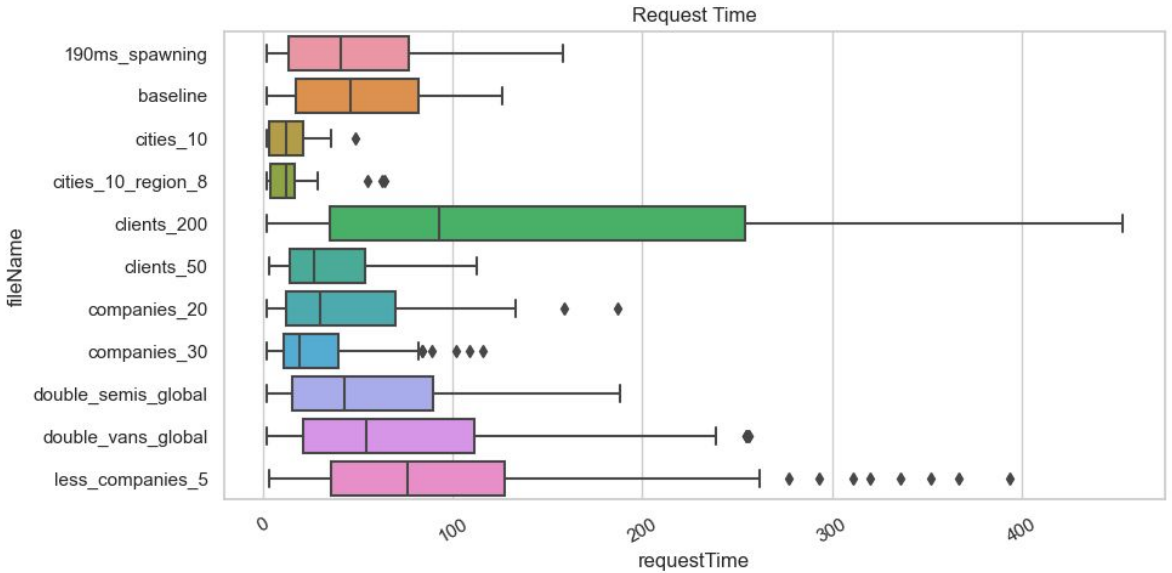


Other Mechanisms

- Client:
  - Look in the DF for every producer to get every available product.
  - Chooses a product.
  - Initiates a contract net with the respective producers to ask for a price
- Producer:
  - Usage of yellow pages to announce what products they produce.
  - Look in the DF for every company.
  - Initiates a contract net with every company to ask for their pricing for a route
- Company:
  - Usage of yellow pages to announce that they exist

Experiments

- Baseline:
  - Clients: 100
  - Cities: 5
  - Regional hubs: 5
  - Companies: 10
  - Client spawn ratio: 20-200 ms
- Varying clients:
  - 50, 150
- Varying cities:
  - 10
- Varying cities (10) and reg. hubs (8)
- Varying companies:
  - 20, 30



- Double semi trucks in global hubs.
- Double vans.
- Varying client spawn ratio:
  - 190-200 ms

Analysis

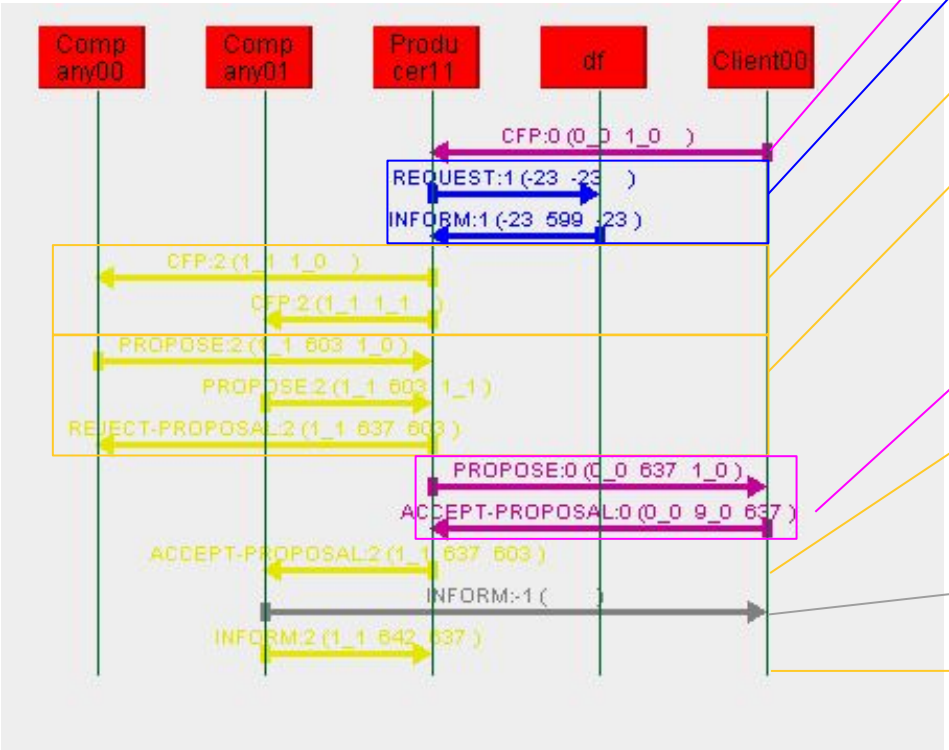
- Successful requests:
  - Increase the number of cities reduces the number of successful requests.
  - Increasing both the number of cities and of regional hubs reduces the number of successful requests but not by as much.
- Price:
  - The price is inversely proportional to the amount of companies (not trivial since proposed prices do not take into account other companies).
- Request time:
  - More clients increased the request time.
  - More companies decrease the request time.
  - More vehicles increased the request time (computational power?).

Conclusions and Future Work

- Conclusions:
  - The dimension of the proposed theme was already big and broad without much exploration of strategies:
    - Several spatial-temporal agents.
  - Logistics is a complicated business.
  - A more dense network is better overall.
- Future Work:
  - Implement more logistics companies pricing strategies.
  - Implement more logistics companies vehicle management strategies.
  - Usage of real-life data.
  - Implement more utility functions for the producers to decide which logistics companies use.

Annexes

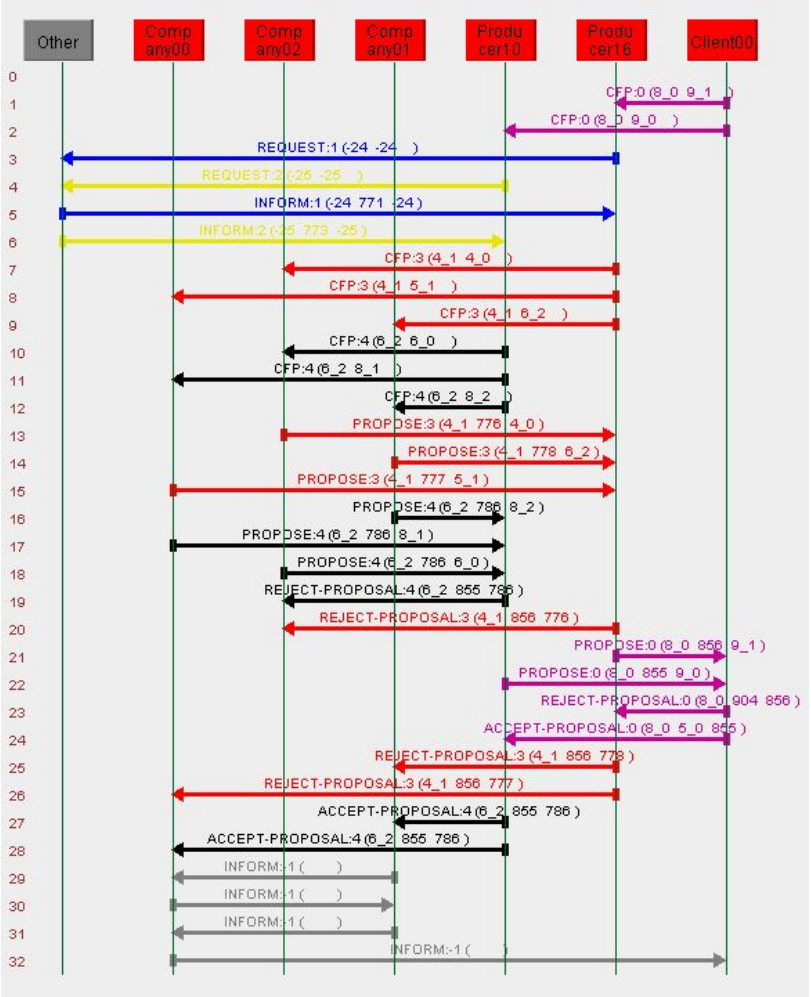
Execution Examples - Bidding



- Client is spawned and emits a CFP.
- The producer, since it does not yet know the available companies, asks the DF and gets a response.
- The producer emits a CFP to each of the companies and gets a response.
- It handles the proposals:
  - It immediately rejects Company00's proposal.
  - Company01's proposal might be accept, so will withhold a response.
- The producer sends a proposal to the Client which is accepted (there are not other producers).
- The proposal by Company00 is accepted.
- Company00, receiving the acceptance, does its job as part of the shipment of the product (in this case just this one takes part).
- After completing the request (in this case, before doing job), it informs the client (grey inform).
- After doing its job, it informs the producer (yellow inform).

Execution Examples - Complete

- Here is a full execution of a client in a sequence diagram that completes the one that only displayed the bidding phase.
- Bidding:
  - Here, the bidding is to two producers but happens analogously
    - Companies bid to both producers
    - Both producers bid to client
    - Producer that is accepted - accepts the companies' proposals
    - Producer that is rejected - rejects the companies' proposals
- Shipping:
  - Each inform between companies represents the moment when the product changes hands to another company
  - The final inform represents the conclusion of the request



Execution Examples - Bidding

```
Client00 is requesting Pencil from 2 producers.
Producer10 received a product request for Pencil from Client00
Producer10 sending route request to 3 companies
Producer16 received a product request for Pencil from Client00
Producer16 sending route request to 3 companies
Producer10 received 3 bids for routes
[2181.818181818182,3000.0,2833.333333333333,4000.0,1.7976931348623157E308] from Company01
[1.7976931348623157E308,3000.0,2000.0,1.7976931348623157E308,1250.0000000000002] from Company00
[1.7976931348623157E308,4000.0,2833.333333333333,1.7976931348623157E308,1333.3333333333335] from Company02
Producer16 received 3 bids for routes
[1.7976931348623157E308,3000.0,2833.333333333333,4000.0,1.7976931348623157E308] from Company01
[1.7976931348623157E308,4000.0,2833.333333333333,1.7976931348623157E308,1333.3333333333335] from Company02
[2285.714285714286,3000.0,2000.0,1.7976931348623157E308,1250.0000000000002] from Company00
Producer10 holding proposal from Company01
Producer10 holding proposal from Company00
Producer10 rejected proposal from Company02
Producer16 holding proposal from Company01
Producer16 rejected proposal from Company02
Producer16 holding proposal from Company00
Client00 received 2 price proposals
Client00 received a proposal of 12441.818181818182 from Producer10
Client00 received a proposal of 12545.714285714286 from Producer16
Client00 accepted proposal of 12441.818181818182 from Producer10
Producer16 finally rejected proposal from Company01
Producer16 finally rejected proposal from Company00
Producer10 finally accepted proposal from Company00
Producer10 finally accepted proposal from Company01
```

Each element of the array is a price for a given route of the total path that the product must travel.

Each company gives a price to all, and an infinite price (1.7976931348623157E308) if they cannot provide that route.

The producer will agglomerate the prices for every route and assign a route to each company. A company can get multiple routes assignments.

Since we have two producers, only one of the proposals will be accepted.

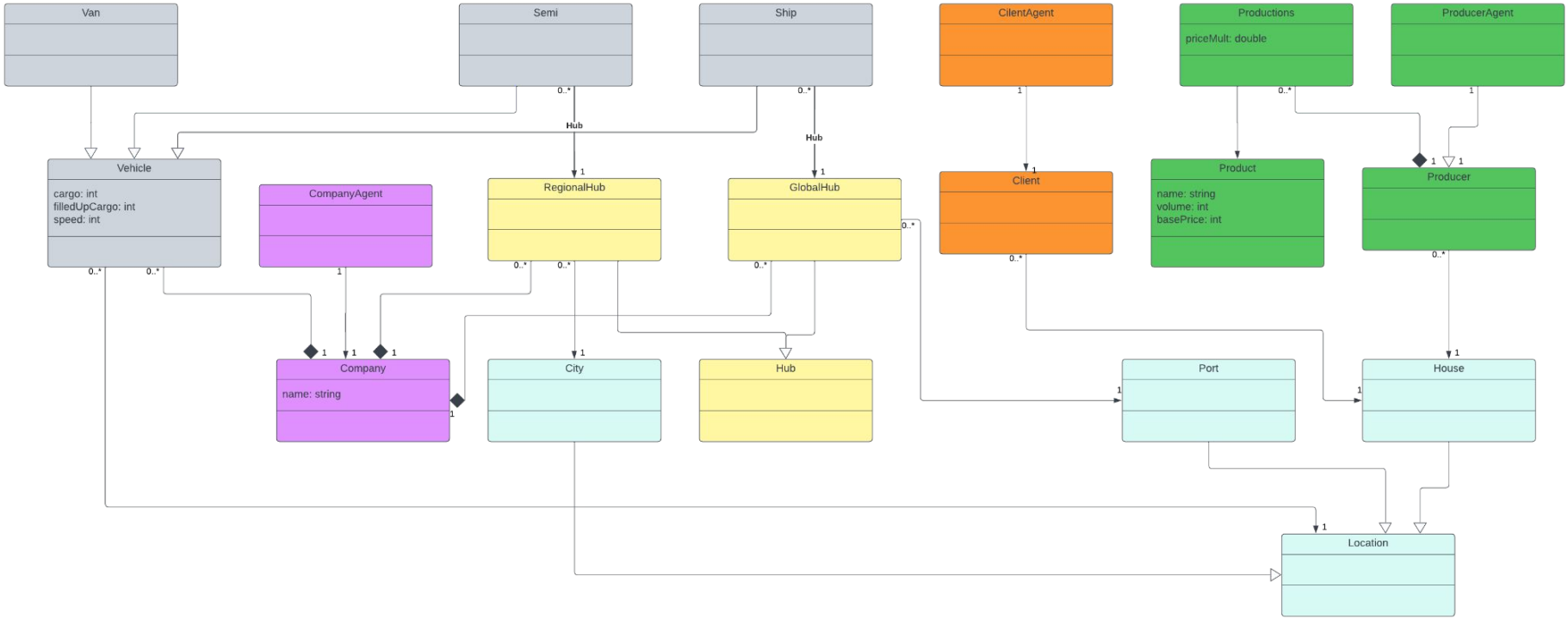
Each producer will relay the response to their proposal to the companies they had on hold.

Execution Examples - Shipping

- First, we can see an overview of the path, including each of the routes (1-4-0 -> 1-4 e.g.) to take and the company that will be taking care of the route dispatch
- Company1-Van68 and Company1-Semi59 are initially stationed at the hubs. Therefore, the vehicle first needs to go to the House and City, respectively, to pick up the product and then return
- The first two routes are dealt by Company1. As such, company 1 deals with the two “in house”. Only for the third route does Company1 inform Company0 that the dispatch has arrived. In this execution, such information is traded 3 times, one for each of the company switches.
- When the request is finished, the client is the one that is informed.

```
Request route
1-4-0 -> 1-4 by Company01
1-4 -> 1 by Company01
1 -> 2 by Company00
2 -> 2-3 by Company01
2-3 -> 2-3-8 by Company00
Company00 got an accept fromProducer10
Company01 got an accept fromProducer10
Company01 is first dispatcher of request
Company01's Company1-Van68 finished dispatch from 1-4 to 1-4-0
Company01's Company1-Van68 finished dispatch from 1-4-0 to 1-4
Company01 completed route from 1-4-0 to 1-4
Company01 continuing request
Company01's Company1-Semi59 finished dispatch from 1 to 1-4
Company01's Company1-Semi59 finished dispatch from 1-4 to 1
Company01 completed route from 1-4 to 1
Company01 informing Company00 about completion
Company00's Company0-Ship0 finished dispatch from 1 to 2
Company00 completed route from 1 to 2
Company00 informing Company01 about completion
Company01's Company1-Semi61 finished dispatch from 2 to 2-3
Company01 completed route from 2 to 2-3
Company01 informing Company00 about completion
Company00's Company0-Van6 finished dispatch from 2-3 to 2-3-8
Company00 sending requestFinished to Client00!
SUCCESS Client-agent Client00 received a product: Pencil
```

Implemented Classes

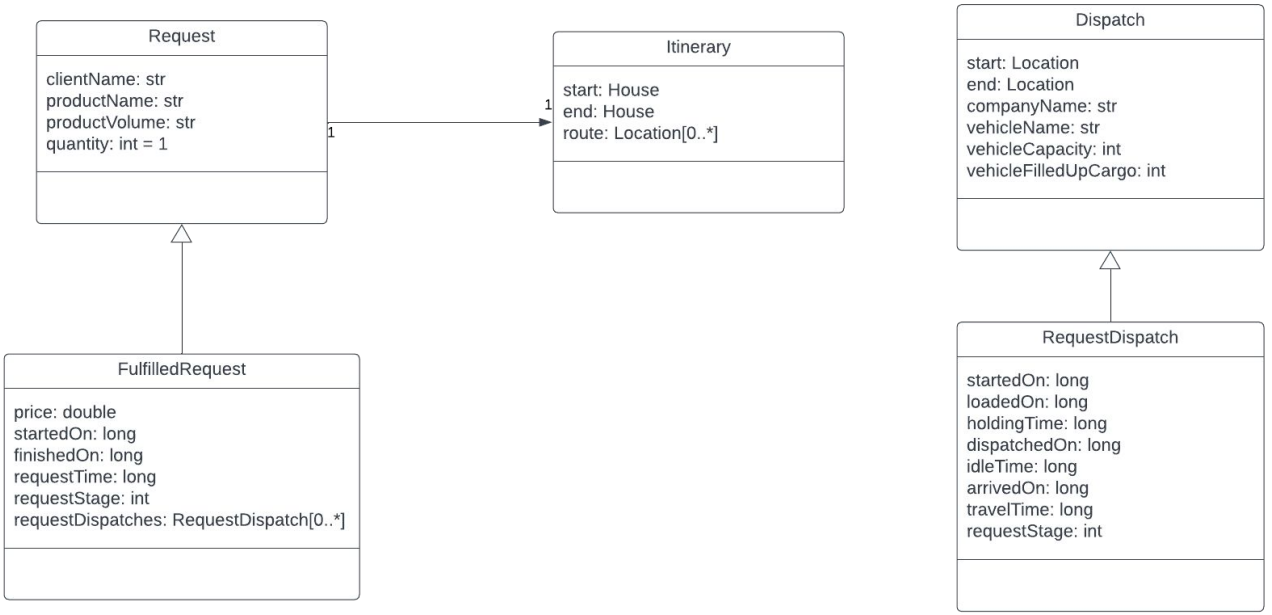


Implementation Classes - Details

In the bidding phase, a Request object will be passed to the CompanyAgent so that they can calculate their price.

In the shipping phase, for the communication between the companies a FulfilledRequest is passed in the message.

In the very end, this object will arrive at the client for confirmation of shipment and it will also be logged.



Experiments Configurations

	Clients	Cities	Regional Hubs	Companies	Company's Semitrucks	Company's Vans	Client Spawn ratio (ms)
baseline	100	5	5	10	3	50	20-200
companies_5	100	5	5	50	3	50	20-200
companies_20	100	5	5	20	3	50	20-200
companies_30	100	5	5	30	3	50	20-200
clients_50	50	5	5	10	3	50	20-200
clients_200	200	5	5	10	3	50	20-200
cities_10	100	10	5	10	3	50	20-200
cities_10_regional_8	100	10	8	10	3	80	20-200
semis_6	100	5	5	10	6	50	20-200
vans_100	100	5	5	10	3	100	20-200
190ms_spawning	100	5	5	10	3	50	190-200

Experiments Configurations - Raw data

	price							
	count	mean	std	min	25%	50%	75%	max
fileName								
00_baseline	99	11019	2234	3273	9484	11814	12394	13417
01_companies_5	99	11316	2233	3414	10498	11909	13034	13418
02_companies_20	99	10288	1950	3273	8892	10738	11605	12306
03_companies_30	99	9696	2007	3273	8845	10513	10864	12238
04_clients_50	50	10717	2216	3273	9182	11385	12453	13247
05_clients_200	199	10903	2529	3273	8988	11923	12314	13361
06_cities_10	22	10206	2839	3273	8833	11233	12393	13382
07_cities_10_region_8	57	10919	2265	3322	10388	11308	12331	14483
08_semis_6	99	9318	1911	3396	8134	10077	10649	11871
09_vans_100	92	10171	2210	3210	8743	11060	11395	12586
10_190ms_spawning	99	11175	2217	3282	10328	11723	12781	13858

	requestTime							
	count	mean	std	min	25%	50%	75%	max
fileName								
00_baseline	99.00	50.75	36.04	2.00	17.50	46.00	82.00	126.00
01_companies_5	99.00	100.31	89.91	3.00	36.00	76.00	127.50	394.00
02_companies_20	99.00	43.66	39.79	2.00	12.00	30.00	70.00	187.00
03_companies_30	99.00	29.26	26.28	2.00	11.00	19.00	39.50	116.00
04_clients_50	50.00	38.32	33.13	3.00	14.25	27.00	54.00	113.00
05_clients_200	199.00	145.28	133.56	2.00	35.00	93.00	254.00	453.00
06_cities_10	22.00	14.50	12.67	2.00	3.25	12.00	21.50	49.00
07_cities_10_region_8	57.00	13.58	13.31	2.00	4.00	12.00	17.00	64.00
08_semis_6	99.00	58.02	52.01	2.00	15.50	43.00	89.50	188.00
09_vans_100	92.00	81.48	75.04	2.00	21.00	54.50	111.75	256.00
10_190ms_spawning	99.00	47.27	39.28	2.00	13.50	41.00	77.00	158.00

	route_len							
	count	mean	std	min	25%	50%	75%	max
fileName								
00_baseline	99.0	5.6	0.7	3.0	5.0	6.0	6.0	6.0
01_companies_5	99.0	5.6	0.7	3.0	5.0	6.0	6.0	6.0
02_companies_20	99.0	5.6	0.7	3.0	5.0	6.0	6.0	6.0
03_companies_30	99.0	5.5	0.8	3.0	5.0	6.0	6.0	6.0
04_clients_50	50.0	5.6	0.7	3.0	5.0	6.0	6.0	6.0
05_clients_200	199.0	5.5	0.8	3.0	5.0	6.0	6.0	6.0
06_cities_10	22.0	5.4	0.9	3.0	5.0	6.0	6.0	6.0
07_cities_10_region_8	57.0	5.4	0.8	3.0	5.0	6.0	6.0	6.0
08_semis_6	99.0	5.5	0.8	3.0	5.0	6.0	6.0	6.0
09_vans_100	92.0	5.5	0.8	3.0	5.0	6.0	6.0	6.0
10_190ms_spawning	99.0	5.6	0.7	3.0	5.0	6.0	6.0	6.0

	holdingTime							
	count	mean	std	min	25%	50%	75%	max
fileName								
00_baseline	99.00	3.14	4.35	0.00	1.00	1.00	2.00	16.00
01_companies_5	99.00	4.32	5.12	0.00	1.00	1.00	9.00	17.00
02_companies_20	99.00	3.63	4.83	0.00	0.00	1.00	9.00	15.00
03_companies_30	99.00	2.91	4.21	0.00	1.00	1.00	2.00	14.00
04_clients_50	50.00	1.92	3.08	0.00	0.25	1.00	1.00	14.00
05_clients_200	199.00	4.27	5.30	0.00	1.00	1.00	9.00	15.00
06_cities_10	22.00	3.05	4.28	1.00	1.00	1.00	1.75	14.00
07_cities_10_region_8	57.00	0.95	1.75	0.00	0.00	1.00	1.00	13.00
08_semis_6	99.00	3.40	4.75	0.00	1.00	1.00	5.50	23.00
09_vans_100	92.00	4.23	5.00	0.00	1.00	1.00	9.00	15.00
10_190ms_spawning	99.00	3.48	4.45	0.00	1.00	1.00	8.00	17.00

	travelTime							
	count	mean	std	min	25%	50%	75%	max
fileName								
00_baseline	99.00	8.18	5.27	1.00	2.00	10.00	11.00	16.00
01_companies_5	99.00	8.24	5.27	1.00	2.00	10.00	11.50	16.00
02_companies_20	99.00	8.29	5.29	1.00	2.00	10.00	12.00	16.00
03_companies_30	99.00	8.27	5.32	1.00	2.00	10.00	11.00	16.00
04_clients_50	50.00	8.36	5.39	1.00	2.00	10.00	13.25	16.00
05_clients_200	199.00	8.38	5.46	0.00	2.00	10.00	14.00	16.00
06_cities_10	22.00	7.27	5.14	1.00	2.00	9.00	10.00	15.00
07_cities_10_region_8	57.00	6.79	5.06	0.00	2.00	9.00	11.00	15.00
08_semis_6	99.00	8.05	5.16	1.00	2.00	10.00	11.00	16.00
09_vans_100	92.00	8.50	5.16	1.00	2.00	10.00	11.75	16.00
10_190ms_spawning	99.00	8.25	5.15	1.00	2.00	10.00	12.00	16.00

	unique_companies							
	count	mean	std	min	25%	50%	75%	max
fileName								
00_baseline	99.0	3.5	1.0	1.0	3.0	4.0	4.0	5.0
01_companies_5	99.0	2.9	0.9	1.0	2.0	3.0	4.0	5.0
02_companies_20	99.0	4.0	1.0	1.0	3.0	4.0	5.0	5.0
03_companies_30	99.0	3.9	1.2	1.0	3.0	4.0	5.0	5.0
04_clients_50	50.0	3.5	1.1	1.0	3.0	4.0	4.0	5.0
05_clients_200	199.0	3.3	1.1	1.0	3.0	3.0	4.0	5.0
06_cities_10	22.0	3.2	1.1	1.0	3.0	3.0	4.0	5.0
07_cities_10_region_8	57.0	3.7	1.2	1.0	3.0	4.0	5.0	5.0
08_semis_6	99.0	3.6	1.2	1.0	3.0	4.0	4.0	5.0
09_vans_100	92.0	3.6	1.1	1.0	3.0	4.0	4.0	5.0
10_190ms_spawning	99.0	3.5	1.1	1.0	3.0	4.0	4.0	5.0

Experiments Configurations - Analysis