

Collaborative Carrier Network in Python

Lorenz Ledding, Max Kretschmann, Shachar Dan
Institute for Software Systems – TU Hamburg – Germany

Agenda

- Task
- Agent Communication Protocol
- Auction Phases
- Bundles
- Scrum Review
- Code Structure
- Gitlab CI and Automated Testing
- Quality of Solution
- Web-App Implementation
- Live Demo

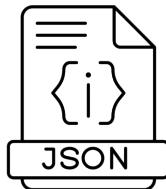
Task

- **General Task:** „Develop an agent based auctioneering solution to allow carriers to sell unprofitable shipments to other carriers.“
- 8 Stories -> **All resolved**
- Most **challenging** Stories:
 - **ST02:** Agent Infrastructure
 - **ST05:** Simple Auctions
 - **ST07:** Auction with Bundles



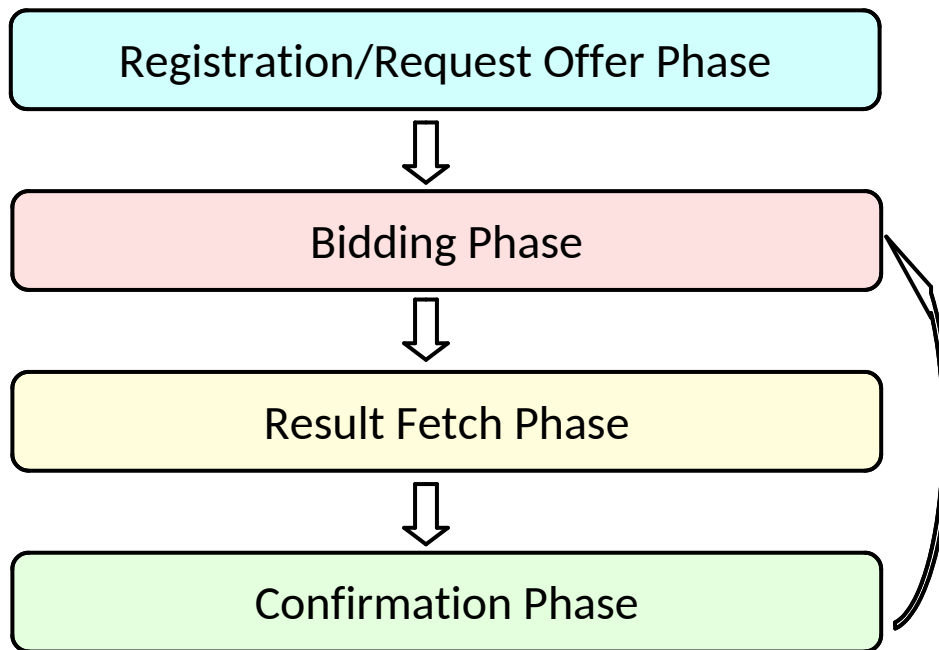
Agent Communication Protocol

```
{
  "carrier_id": "carrier1",
  "action": "request_offer",
  "time": "1719492196",
  "timeout": 1719492199,
  "payload": {
    "status": "OK",
    "offer": {
      "offer_id": "bundle_11fc633f-d812-4fb9-965d-f0e6d3bfe117",
      "loc_pickup": [
        {
          "pos_x": 91.99215240497111,
          "pos_y": -19.126706287206517
        }...
      ],
      "loc_dropoff": [
        {
          "pos_x": -0.21269604171901335,
          "pos_y": 11.017038302798326
        }...
      ],
    },
  },
  "revenue": 21473.575514810247
}
```



- *Pull System* for Carrier
- Protocol developed by **us**
- Uses *JSON* Notation
- Very flexible due to *payload* system
- Implemented using *Python Websockets*

Auction Phases



- **4** Stages for Auction
- Every Offer is auctioned off individually
- Terminates after **r** Rounds or when nothing **has been auctioned off**
- Starts with *Bundles*, ends with *Single Offers*

Bundles

Generation of Bundles

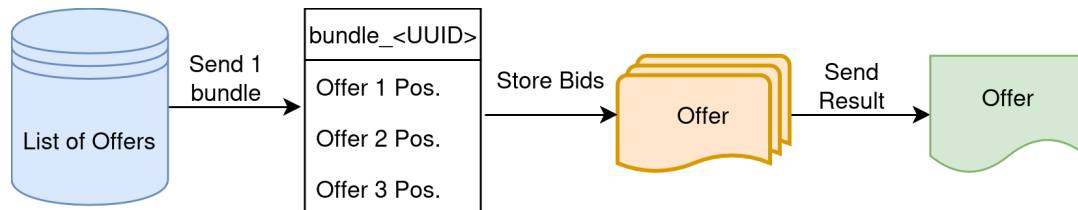
```
for i in range(0, int(n/2)):
    if i % bundle_size == 0 and i != 0:
        bundle_iterator += 1
        self.bundles[bundle_iterator] = []

    if i == 0:
        self.bundles[bundle_iterator] = []

    self.bundles[bundle_iterator].append(self.offers[offers_sorted_indices[i]].offer_id)
    self.bundles[bundle_iterator].append(self.offers[offers_sorted_indices[n-i-1]].offer_id)
```

- **Goal:** Average out the Revenue on Offers
- Stored using **logical abstraction layer** of Offer
- Very **Low storage overhead**

Level of abstraction



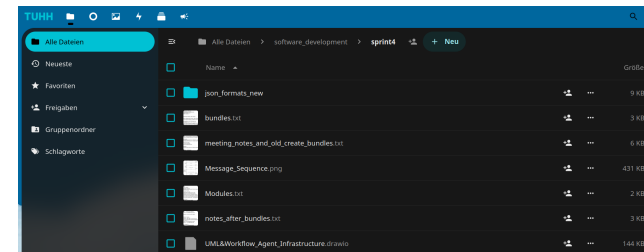
SCRUM Review

- Time planning in Backlog was **realistic**
- Task were **evenly** distributed
- Workload was as **expected**
- Collaboration through several *tools and methods*



```
# FIXME: ACCESS first element of bundle
# (Shachar:) ???
```

	A	B	C	D	E	F	G	H
1	Task	est. Time	Assigned Workers			Worker1	Worker2	Worker3
2	Interagent Communication: Optimize Meta Data Format		1 Worker1			18	18	18
3	Interagent Communication: Implement Optimized Meta Data FC		2 Worker2				18	18
4	Interagent Communication: Implement Data as nested Object		3 Worker2		Req. min		18	18
5	Interagent Communication: Define Fields for Registration		2 Worker2					
6	Interagent Communication: Implement Fields for Registration		3 Worker3					
7	Interagent Communication: Define Fields for Routes Request		2 Worker2					
8	Interagent Communication: Implement Fields for Routes Request		3 Worker3					
9	Interagent Communication: Define Fields for Routes Offer		2 Worker3					
10	Interagent Communication: Implement Fields for Routes Offer		4 Worker3					
11	Interagent Communication: Define Fields for Betting		2 Worker1					
12	Interagent Communication: Implement Fields for Betting		3 Worker3					
13	Interagent Communication: Define Fields for Fetching Result		2 Worker1					
14	Interagent Communication: Implement Fields for Fetching Result		3 Worker3					
15								
16	Define: Data Model for Storing Results		4 Worker1					
17	Implement: Data Model for Storing Results		3 Worker2					
18								
19	Define: Automata for Program Flow (Registration, Auction, etc)		6 Worker1					
20								
21	Define: Routing-Carrier Interface		3 Worker1					
22	Implement: Agent Interface in Routing Software		3 Worker2					
23	Implement: Agent Interface in Carrier Software		3 Worker2					
24								
25								
26								
27								
28			54	54	0			



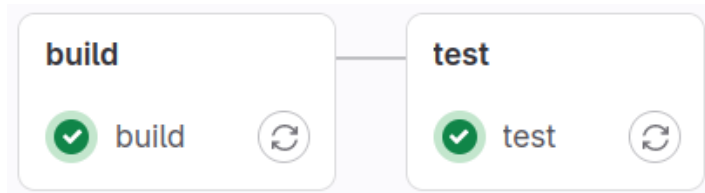
Code Structure

Code Example

```
@send_response("offer")
def receive_offer(self, data):
    """
    Receive an offer from a carrier.
    """
    carrier_id = data['carrier_id']
    if self.auctioneer.phase != "REGIST":
        return {"response": "OFFER_SUBMISSION_TIMEOUT"}
    if carrier_id not in self.auctioneer.registered_carriers:
        return {"response": "NOT_REGISTERED"}
    self.auctioneer.add_offer(carrier_id, data['payload'])
    payload = {
        "offer_id": data['payload']['offer_id'],
        "response": "OK"
    }
    return payload
```

- OOP Approach
- **Multiple Threads:**
 - **1** *Launch* Thread
 - **1** *Phase* Handler Thread
 - **1** *Client* Handler Thread
 - **GIL**
- Route planning calculated with *OR-Tools*
- **Syntatic Sugar:** *Function Decorators*

Gitlab CI and Automated Testing



```
def test_generate_bundles(self):
    self.auctioneer.generate_bundles(bundle_size=2)
    # ensure all bundles have size bundle_size, except potentially the last one
    bundles = list(self.auctioneer.bundles.values())
    bundle_size = len(bundles[0])
    for i in range(len(bundles) - 1):
        self.assertEqual(len(bundles[i]), bundle_size)
    # check the last bundle
    if len(bundles) > 0:
        last_bundle_size = len(bundles[-1])
        self.assertTrue(last_bundle_size <= bundle_size, f"Last bundle size is {last_bundle_size} which is greater than {bundle_size}")
    # check that correct bundles order
    expected_bundles = [['offer5', 'offer2', 'offer4', 'offer1'], ['offer6', 'offer3']]
    self.assertEqual(bundles, expected_bundles)
```

- **2** Unit-Test classes
 - *Auctioneer*
 - *Routing*
- Static Assert Tests
 - Checks **Correctness** of Functions and Workflows
- **CI:** Direct Feedback after Push
 - Fully **automated** with *gitlab-ci*

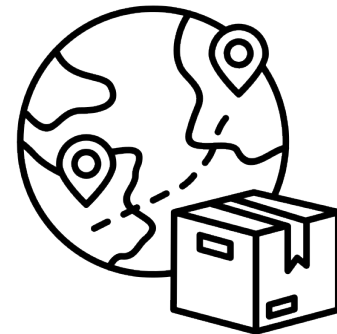
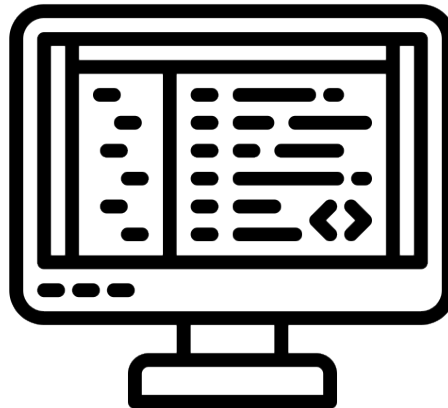
Quality of Solution

- **Significant** Increase of Profit (>2,000% possible!)
- **Profit** does **not** decrease

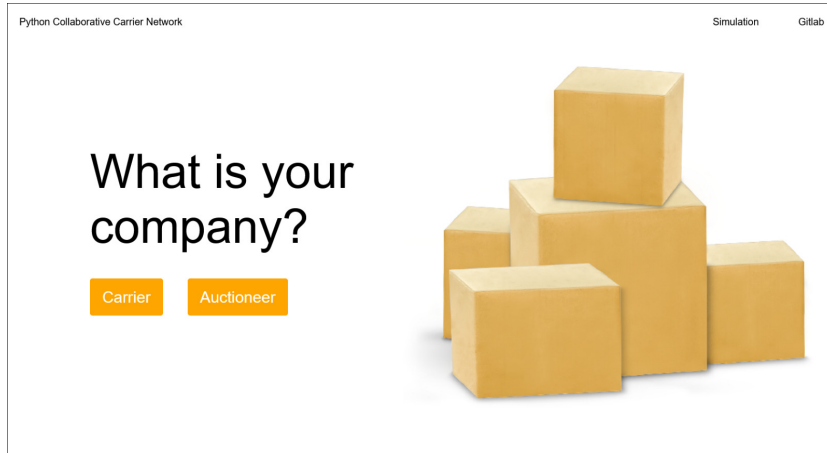
```
Before auction day: {'revenue': 10669.12, 'cost': 10968.12, 'profit': -299.0}  
After auction day: {'new_revenue': 16560.74, 'new_cost': 9960.0, 'new_profit': 6600.74}  
Increase: 2307.58%
```

-
- **Benchmark:**
 - Most **expensive** Operation: *Or-Tools Route Calculation*
 - » Performed **only** by Carrier
 - Auctioneer: Can handle >10.000+ Carriers (*theoretically*)
 - **Well-Structured** and **maintainable** Code-Base
 - **Standardized Communication Protocol** for easy adaptation

LIVE DEMO



Web-App Implementation



WebUI Start Page

- Written in *Flask*
- **Simple** and **Clean** UI
- Features:
 - Upload / generation of *shipment files*
 - Startup of *Auctioneering Server*

LIVE DEMO

