OPENSHIPPING.ORG

Workshop

TRADELENS BACKGROUND

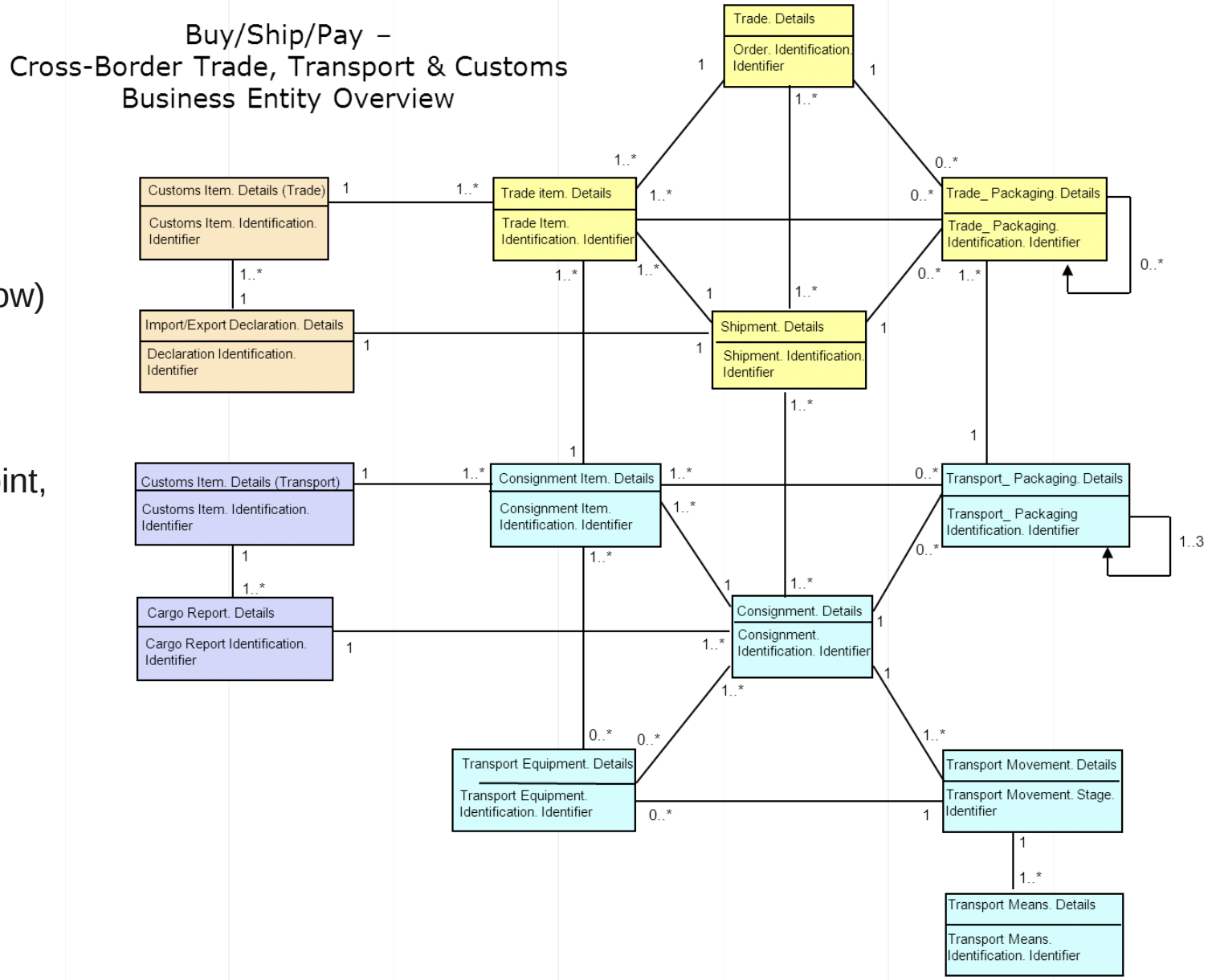
- TradeLens is a joint effort, collaboratively headed by IBM and Maersk Line. Its aim is to digitize the documentation and processes involved in global trade.
- This is achieved through a foundation of modern API data interfaces, backed by blockchain technologies.
- Fitting into a global network of trade and logistics systems, TradeLens fully recognizes the necessity to base interfaces on open standards – no one today wants to invest in integrating to proprietary protocols.
- Where standards are missing, TradeLens is investing to help defined and drive new ones forward.

TARGETED PROCESSES AND TECHNOLOGY

- TradeLens's scope is the post-booking logistical and operational processes.
- This is the natural extension to the pre-booking scope, discussed in Hamburg on June 7th 2018.
- The aim is to facilitate the interoperability and data flows between the significant parties involved in Supply Chain and Logistics.
- Technically, the ambition is to move from narrowly defined, use cases specific protocols towards generic, self-explanatory APIs of resources.
- Data should be shared "event-driven", as they get generated – rather than in bulk, when they are assumed to be of interest to others.
- Embracing modern RESTful standards facilitates industry new-comers and innovation.

UN/CEFACT FOUNDATION

- The proposed API is an adoption of the UN/CEFACT abstract class model.
- It differentiates between Trade Data (yellow) and Transport Data (blue). Particularly, it distinguishes between Shipment and Consignment.
- The UN/CEFACT model is the starting point, not the solution.

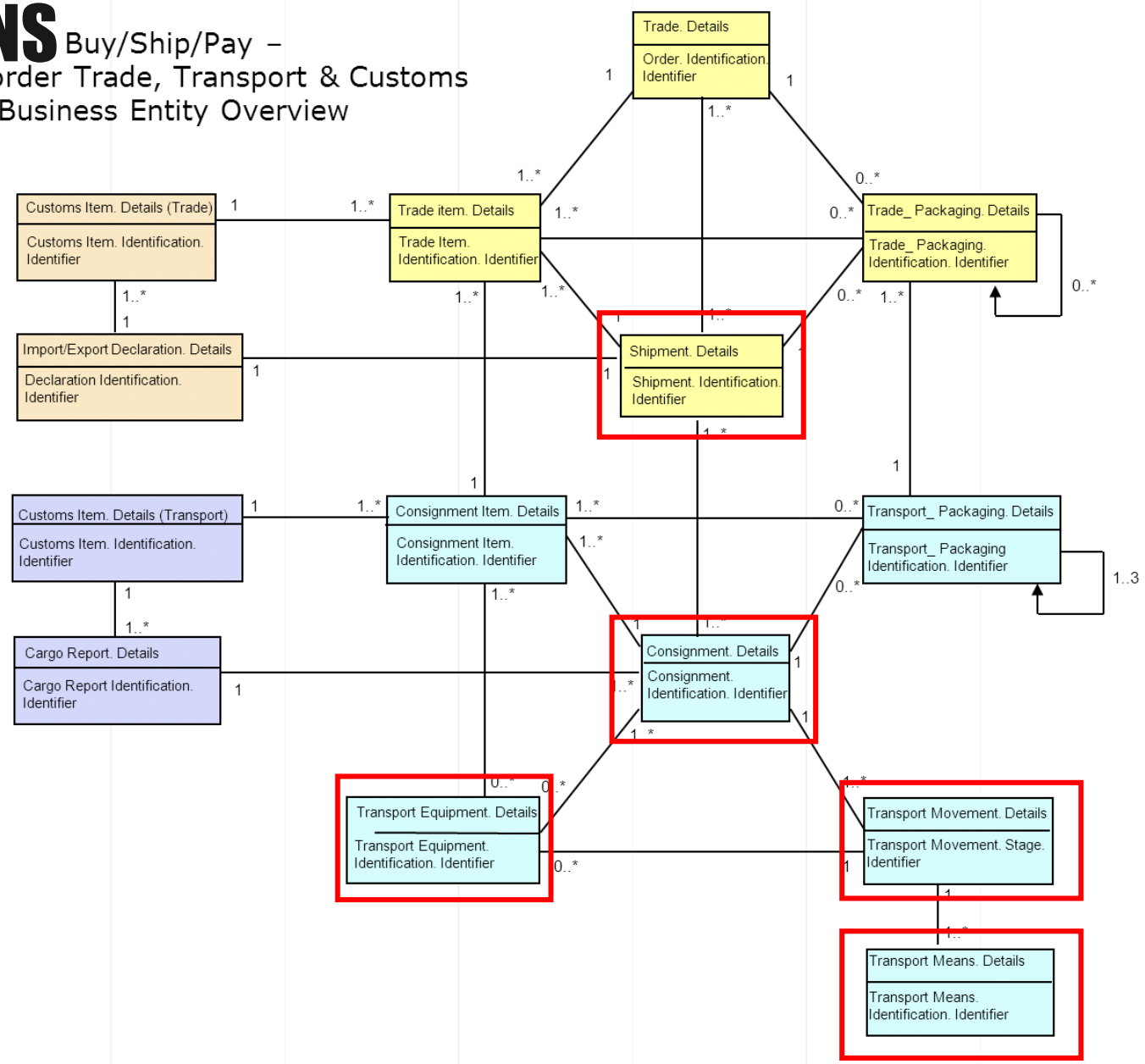


MAIN UN/CEFACT INTERPRETATIONS

Buy/Ship/Pay –
Cross-Border Trade, Transport & Customs
Business Entity Overview

Targeted resources:

- Shipment
- Consignment
- Equipment
- Movement
- Means



SHIPMENT-CONSIGNMENT-CONSIGNMENTITEM

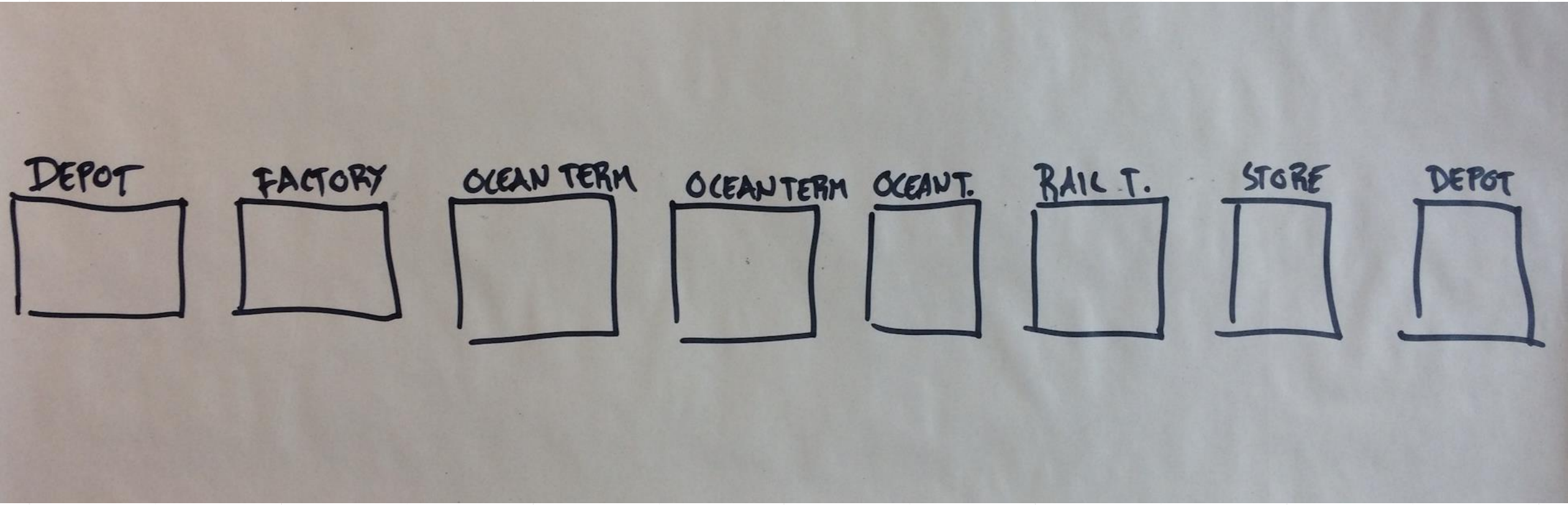
- From the UN/CEFACT documentation:
  - **Shipment (Trade Delivery)**
    A shipment is an identifiable collection of one or more Trade Items (available to be) transported together from the Seller (Original Consignor/Shipper) to the Buyer (Final/Ultimate Consignee):
    - A Shipment can only be destined for one BuyerTrade Items from one or more Sales
    - A Shipment can be made up of some or all Orders
    - A Shipment can have only one Customs UCR
    - A shipment may form part or all of a Consignment or may be transported in different Consignments.
  - **Consignment**
    A consignment is a separately identifiable collection of Consignment Items (available to be) transported from one Consignor to one Consignee via one or more modes of transport as specified in one single transport service contractual document:
    - A Consignment can only have one Transport Service Buyer
    - A Consignment can only have one Transport Service Provider
    - A Consignment can only have one Consignor
    - A Consignment can only have one Consignee
    - The Transport Service Buyer can be either the Consignor or the Consignee
    - A Consignment is made up of one or more Consignment Items
    - A Consignment can be made up of some or all Trade Items (aggregated into Consignment Items) from one or more Shipments

- Equipment represented by its ISO6346 code (physically printed on the container) as an attribute on ConsignmentItem. The “asset management” aspects of the container are not of particular importance to the targeted scope, but if Equipments were to be created as resources, the ISO6346 would still be an essential identifier.

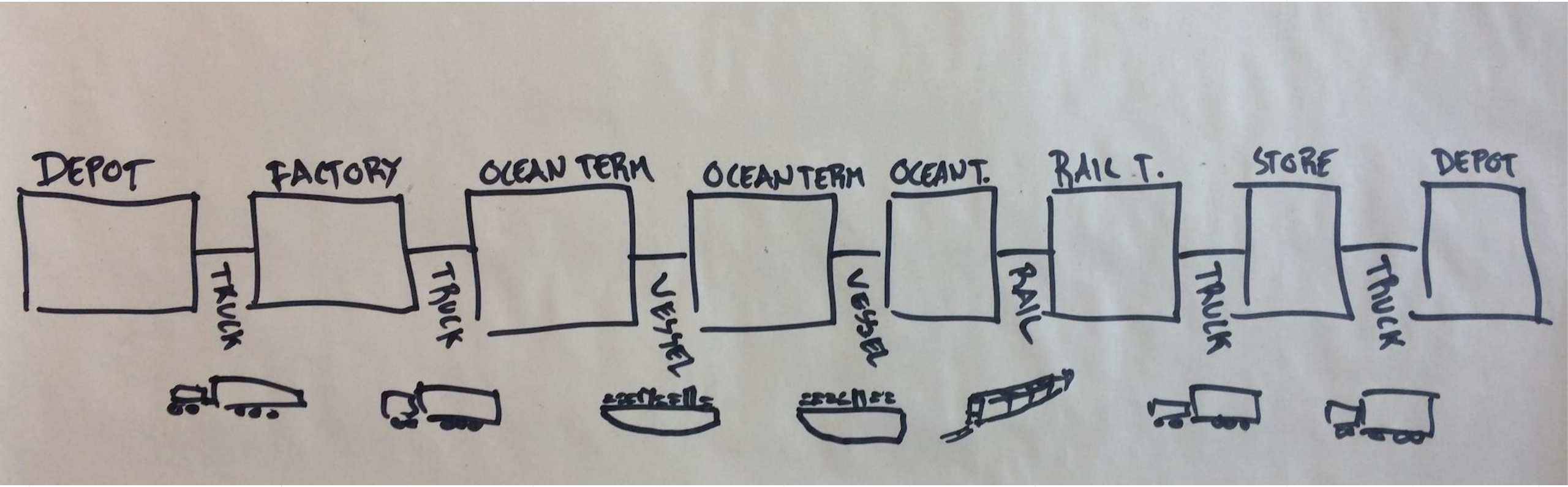
RESOURCE MODEL

POST	`/shipments`	Registers a new Shipment.
PATCH	`/shipments/{shipmentId}`	Update Shipment
POST	`/consignments`	Register a new Consignment.
PATCH	`/consignments/{bookingNumber}`	Updates Consignment details (Shipment, Container Transports, Parties, Consignment References)
POST	`/transportEquipments`	Creates a Container Transport. TransportEquipments can be created by the following roles Carrier (CA), Consignor (CZ), Rail carrier (GT ORIGIN or GT DESTINATION), Road carrier (GA ORIGIN or GA DESTINATION)
PATCH	`/transportEquipments/{transportEquipmentId}`	Updates TransportEquipment's attributes (consignment, equipmentNumber, commodityCode)

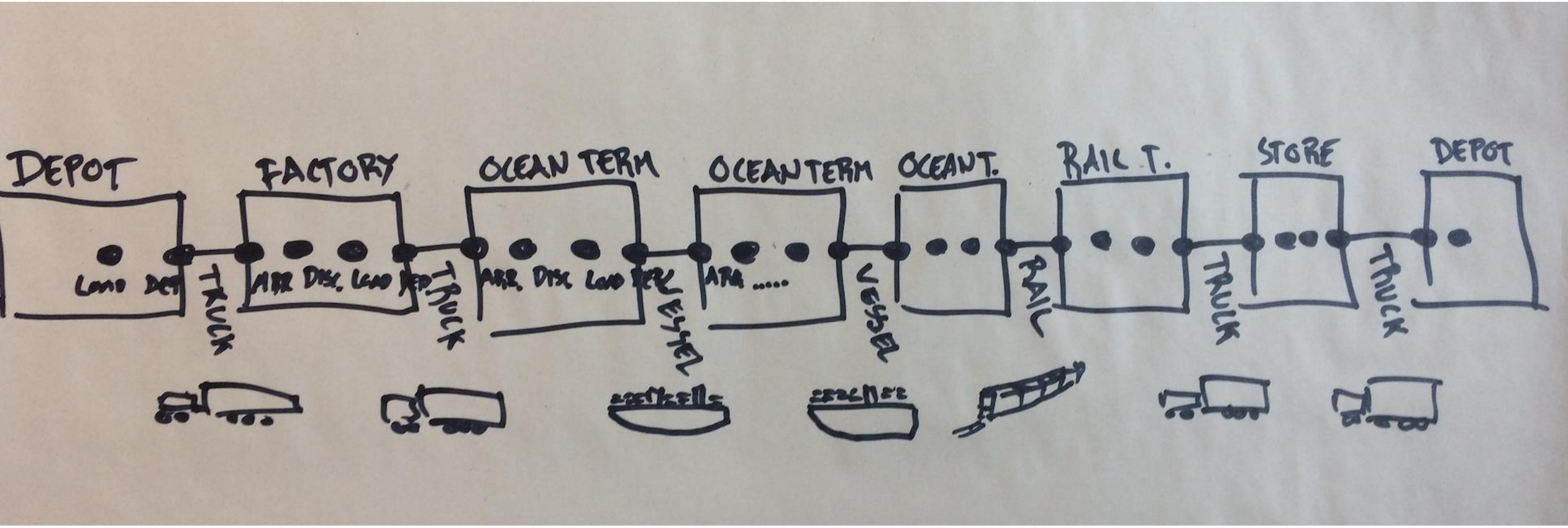
EVENT MODEL – LOCATIONS



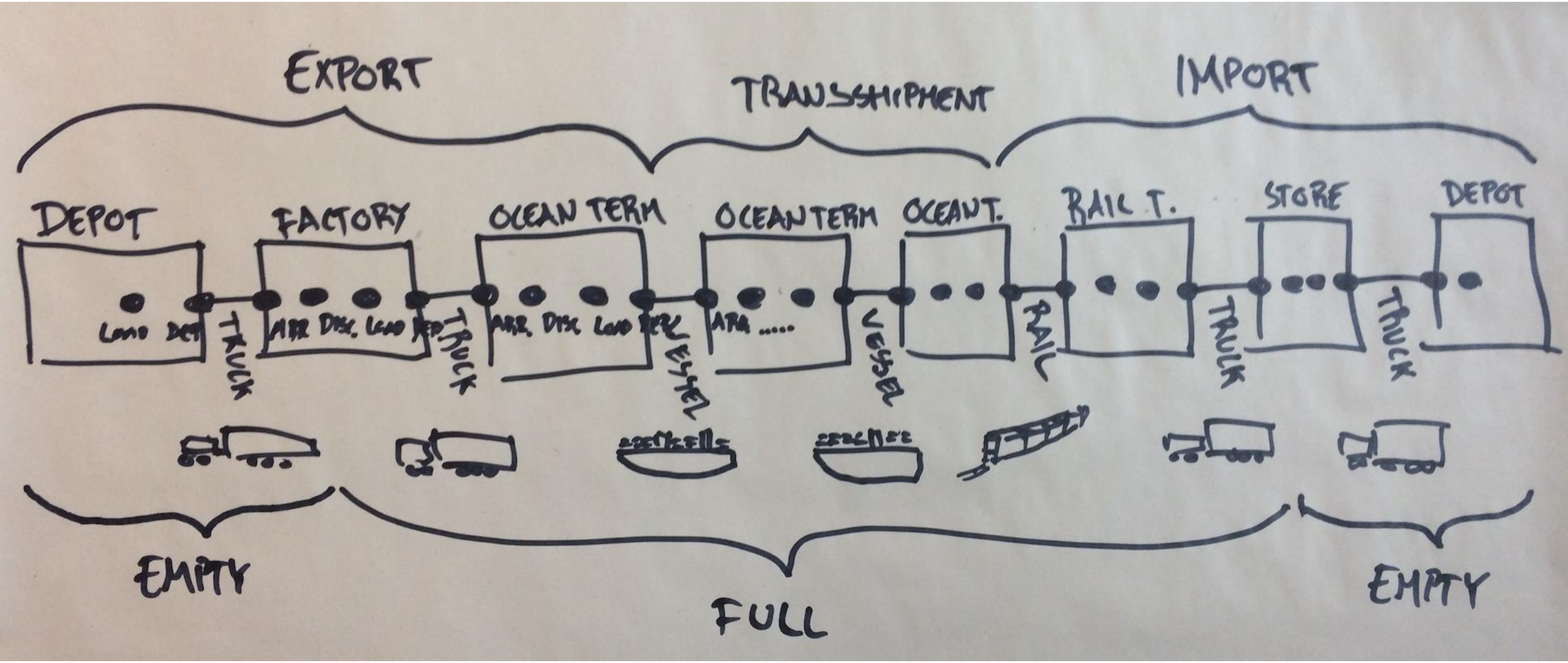
EVENT MODEL – LEGS



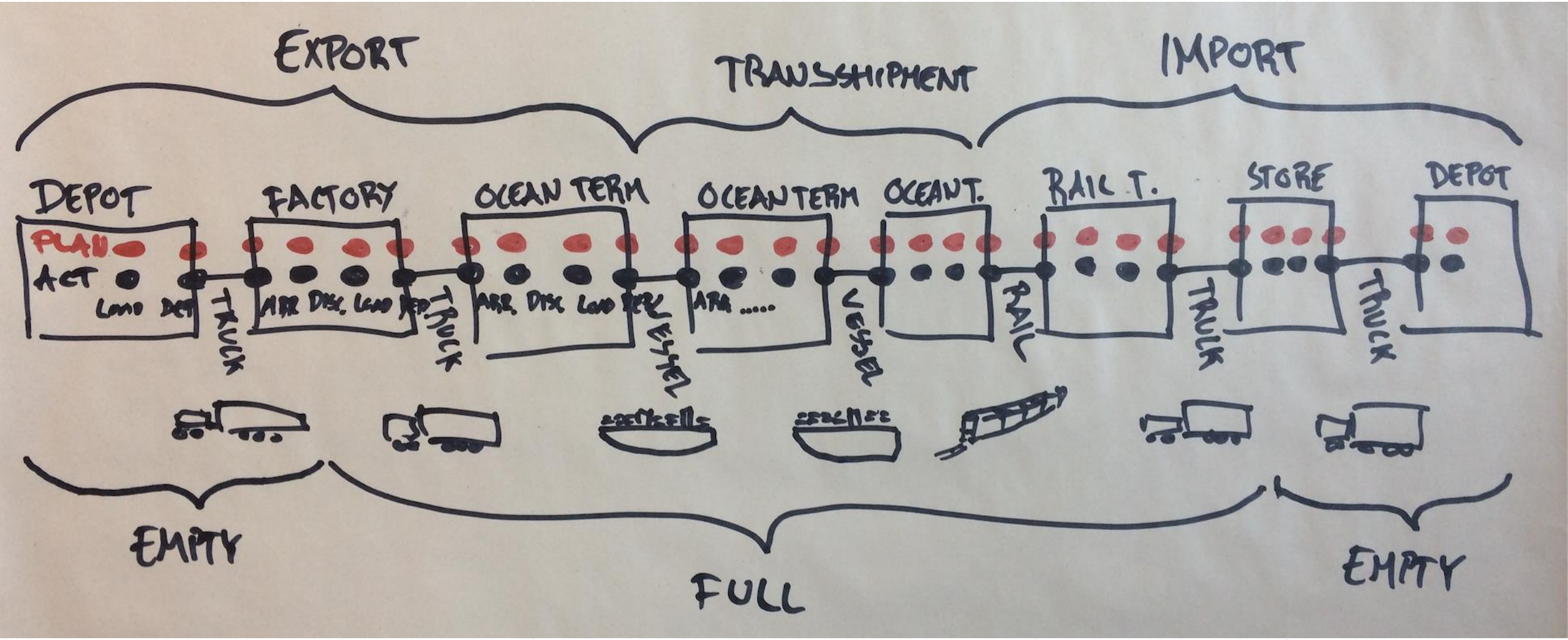
EVENT MODEL – FOUR EVENTS PER LEG



EVENT MODEL – PHASE AND FULLSTATUS ATTRIBUTES

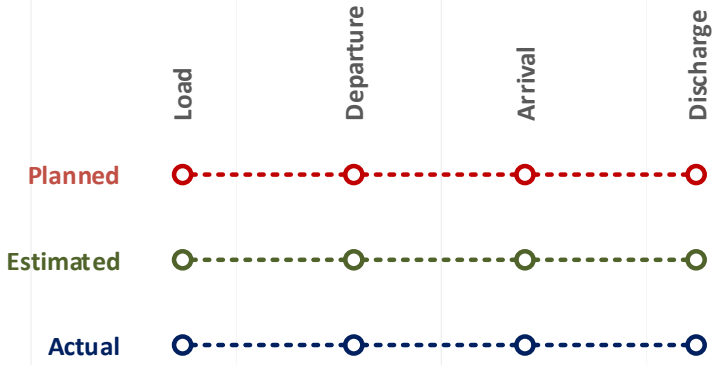


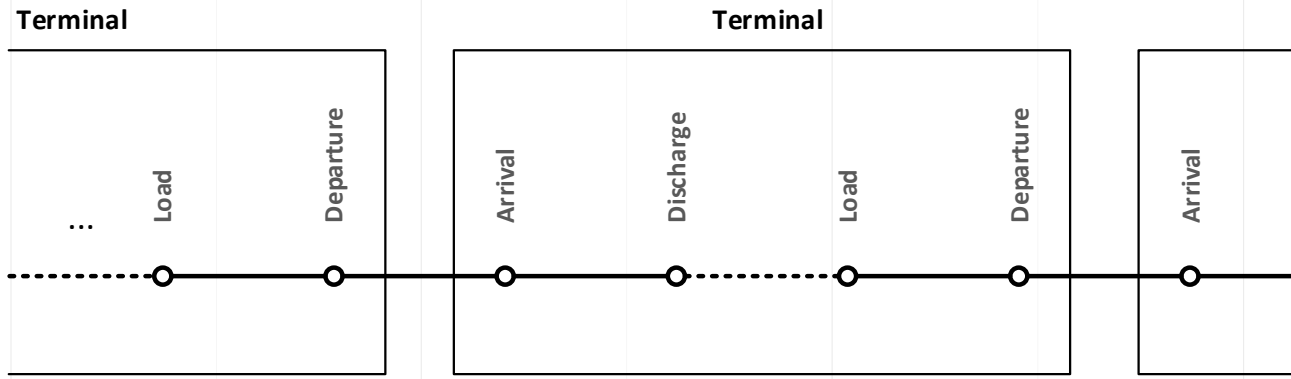
EVENT MODEL – PLANNED, ESTIMATED, ACTUAL



EVENT MODEL

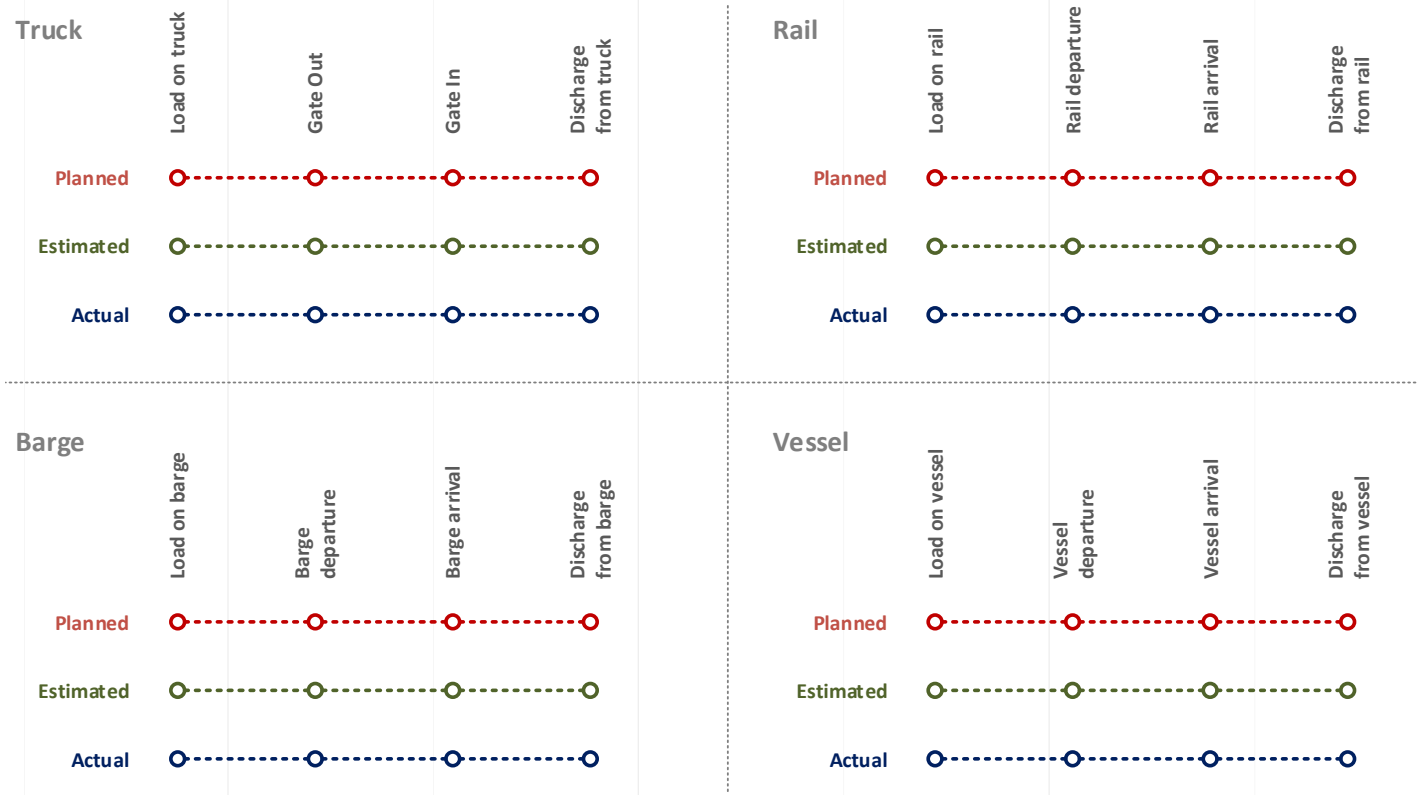
- A transport plan Leg is described by four events:
  - Load
  - Departure
  - Arrival
  - Discharge
- This, in turn, come in three flavors:
  - **Planned** – what is supposed to happen
  - **Estimated** – what and when events are expected to happen
  - **Actual** – what has indeed happened
- Legs are joined together to form a full Consignment transport plan. Leg-joining happens at terminals.



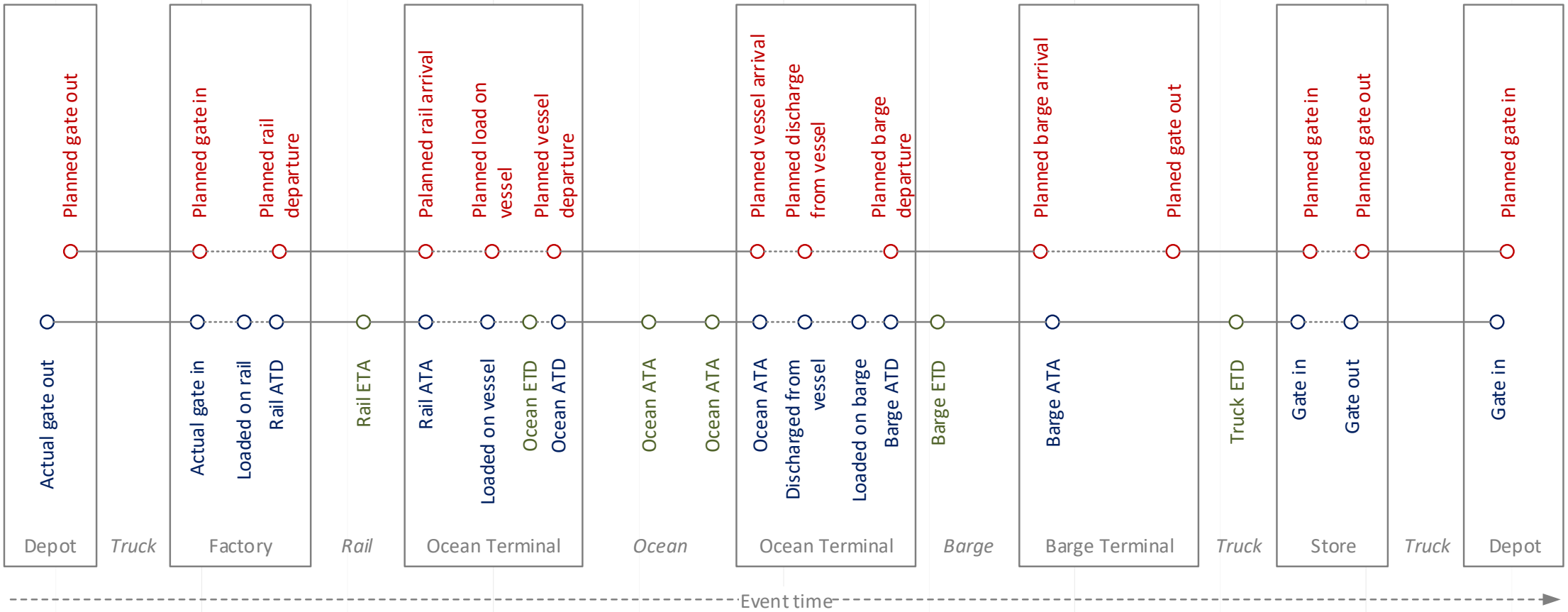


EVENT MODEL

- There are different events for four types of legs:
  - Truck
  - Rail
  - Barge
  - Vessel
- The event model thus consists of a total of 48 events – i.e. endpoints.
- This model is easily extensible (e.g. adding an “Air” leg type).
- While the event model is strictly logical, the naming of individual events are made to embrace common industry lingo. E.g. “Gate out” instead of ”Truck departure”.



EVENT MODEL EXAMPLE



LINKING OF CONSIGNMENTITEM AND EVENTS

Planning is often done for all the consignment's containers at once, so it makes sense to allow to submit them all in one go.

`POST` **`/consignments/{bookingNumber}/plannedVesselLoads`** ↵

Planning on individual container level can also make sense and should be allowed as well.

Further, only the carrier, who is responsible for delivering the consignment service should be allowed to post planned events.

By contrast, actuals and estimates are always handled individually per container:

`POST` **`/transportEquipments/{equipmentNumber}/estimatedGateIns`** Creates an estimated gate in event ↵

`POST` **`/transportEquipments/{equipmentNumber}/actualGateOut`** Creates and actual gate out event, representing the departure of a truck from a terminal ↵

… Let's go have a look at https://app.swaggerhub.com/apis/OpenShippingDotOrg/OpenShipping

OTHER ITEMS

- How does this apply to the LCL scenarios?
- Is the endpoint-attribute split appropriate?
- Are we missing additional physical moves/events in between the 4-per-leg model?
- SMDG: “RULED-PLP” vs “RULEDPLP”?