# Documentation

##### Name: Krystal Distribution Group

##### Author: Mykhailo Kruts

##### Description: Krystal Distribution Group (KdG) specializes in the efficient and seamless distribution of essential raw materials, including gypsum, iron ore, cement, petcoke, and slag. They want to renew their logistics system, as it is outdated and hard to change. They need a new system that can be adapted easily and makes data available for other systems.

Table of contents:

[**Documentation 1**](#_xxjzg84gxycd)

[Subdomains 2](#_u4z3fluu1a0s)

[Warehouse context: 3](#_2bjl44hq7l3q)

[Landside context: 4](#_j7in9f9x2uka)

[Waterside context: 6](#_vd23p1nv46j8)

[Invoicing context: 8](#_yglrkbatfvyq)

[Context mapping: 9](#_876fdotld5er)

## 

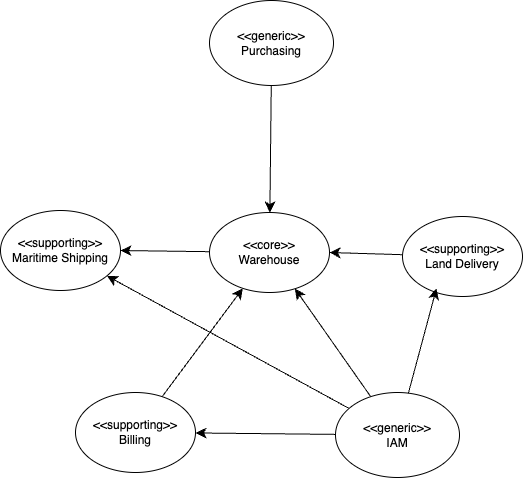
## 

## 

## 

## Subdomains

* Warehouse :core
* Land Delivery :supporting
* Maritime Shipping :supporting
* Billing :supporting
* Purchasing :generic
* IAM :generic

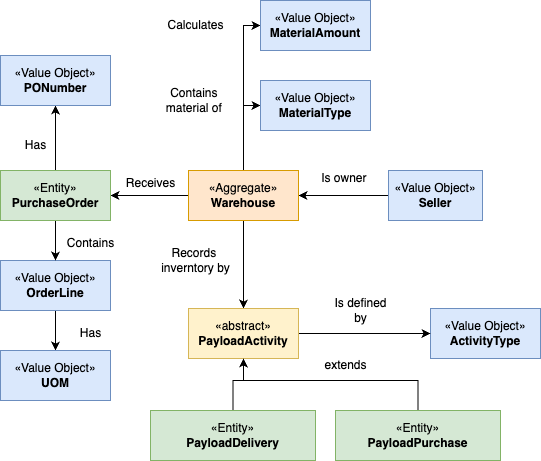


## Context mapping:

* **Published Language** between 4 contexts:
  + Landside
  + Warehouse
  + Waterside
  + Invoicing

## Warehouse context:

#### Domain model:



#### Purpose:

* Manage warehouse information, including storage capacity and raw material stock levels.
* Maintain records of mineral types and quantities stored.

#### Entities:

* PurchaseOrder
* PayloadActivity
  + PayloadDelivery
  + PayloadPurchase

#### Value Objects:

* ActivityType
* MaterialType
* MaterialAmount
* OrderLine
* UOM

#### Aggregates:

* Warehouse
  + Seller
  + MaterialType
  + PayloadActitvity[]

#### Commands:

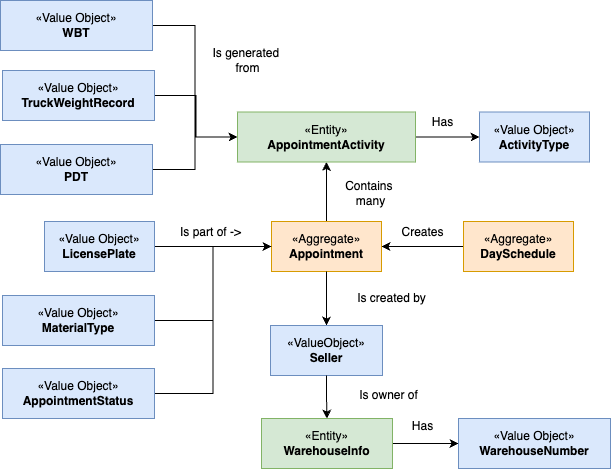
* SnapshotCommand
  + Description: Command to trigger snapshotting specific warehouse

#### Events:

* WarehouseCapacityChangeEvent
  + Description: Project warehouse capacity change to landside
* AdjustInventoryEvent
  + Description: Register delivery of payload
* ChangePOStatusEvent
  + Description: Event to trigger matching of Purchase Order with Shipment Order or initiating deduction of material
* CalculateCommissionForPurchaseOrderEvent
  + Description: Event sent to invoicing to calculate the commission fee for purchase order after successfully deducting the material.
* PurchaseOrderCreatedEvent:
  + Description: Receive Purchase Order from external system and process it.
* StorageChangeEvent:
  + Description: Send arrived payload information to invoicing context to calculate storage fee.

## Landside context:

#### Domain model:



#### Purpose:

* Schedule truck arrivals to ensure efficient processing and avoid congestion.
* Maintain a log of scheduled and actual arrival times.
* Maintain a log of departures.
* Record the weight of trucks arriving and leaving the warehouse.
* Calculate the net weight of minerals delivered.

#### Entities:

* WarehouseInfo
* AppointmentActivity

#### Value Objects:

* ActivityType
* AppointmentStatus
* LicensePlate
* MaterialType
* PDT
* WBT
* TruckWeightRecord
* WarehouseNumber
* SellerId

#### Aggregates:

* Appointment
  + LicensePlate
  + MaterialType
  + AppointmentStatus
  + AppointmentActivity[]
* DaySchedule
  + Appointment[]

#### Commands:

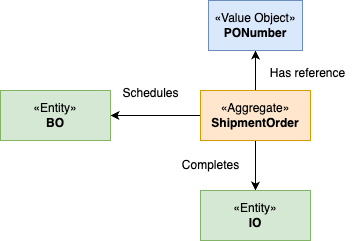
* CreateAppointmentCommand
  + Description: Seller schedules an appointment for truck arrival.
* PassBridgeCommand
  + Description: Truck passes weighing bridge to record his weight. Used to either store his weight on entry or calculate net weight on leaving.
* TruckArrivalCommand
  + Description: Truck arrives to facility and gate recognizes license plate;
* DumpPayloadCommand
  + Description: Truck dumps payload on conveyor belt and receives copy of payload delivery ticket

#### Events:

* WarehouseCapacityChangeEvent
  + Description: Receive the changed projection value from warehousing context.
* PayloadDeliveryEvent
  + Description: Send calculated amount of payload that was delivered.

## Waterside context:

#### Domain model:



#### Purpose:

* Track outgoing shipments.
* Track Inspection Operations, Bunkering Operation
* Match Shipping Orders with Purchase Orders

#### Entities:

* BO
* IO

#### Value Objects:

* PONumber

#### Aggregates:

* ShipmentOrder
  + BO
  + IO
  + PONumber (reference)

#### Commands:

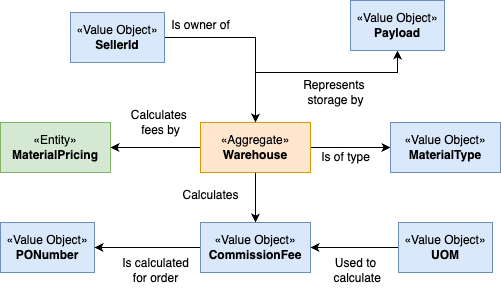
* CheckVesselStatusCommand
  + Description: Ship captain checks status of ship order and vessel operations.
* VesselInspectionCommand
  + Description: Worker completes inspection operation.
* InputVesselInfoCommand
  + Description: Ship captain inputs shipment order with vessel information
* MatchSOAndPOCommand
  + Description: Match purchase order with shipment order
* PlanBunkeringOperationCommand
  + Description: Dock worker plans bunkering operation for vessel for specific date

#### Events:

* ChangePOStatusEvent
  + Description: Send PO number to state that either Shipment Order and Purchase Order were Matched or all operations of vessel have finished

## Invoicing context:

#### Domain model:



#### Purpose:

* Calculate storage fee each day per customer
* Calculate commission fee each day per customer
* At request invoicing to customers (invoice outstanding credit)

#### Entities:

* MaterialPricing

#### Value Objects:

* SellerId
* Payload
* PONumber
* CommissionFee
* UOM

#### Aggregates:

* Warehouse
  + MaterialPricing
  + MaterialType
  + SellerId
  + Payload[]

#### Commands:

* CalculateStorageFeeCommand
  + Description: Calculate storage fee for a specific seller.

#### Events:

* CommissionEvent
  + Description: Event received to invoicing to calculate the commission fee for purchase order after successfully deducting the material.
* StorageChangeEvent:
  + Description: Receive arrived payload information to invoicing context to calculate storage fee.