Auto DML v1

Author - Aakash Khatri

Version - 1

Last update - 20th April 2023

Vision

The end state vision of the auto DML is to build an intelligent layer which will encompass not just LaaP orders but also any supply chain orders including e2e. This layer should be able to understand key parameters about the supplier, customer and also have an intelligent network mapping built basis which the most efficient supply chain can be chosen. In the end state, the auto DML layer will include features like weight, category, pricing, TAT, customer & seller characteristics, 3PL capacity, node level TAT & pricing etc. which will enable the layer to formulate the decision of choosing the appropriate supply chain.

Design Principles

1. Separation of intent - Solutioning of Auto DML needs to be defined across multiple layers with each having a clear goal mutually exclusive of the preceding or succeeding layer. This is done to avoid conflicts/ human errors in file uploads for manual data entry, keep each layer individually scalable and diagnose system faults more accurately.
2. Computation optimization - With multiple layers and intrinsic rule based system built into the Auto DML, the final output on every order basis will be extremely computation heavy. Hence, to avoid just in time decisions and large response times, a decision optimization layer will need to be built especially across the network mapping layer. (To be further scrutinised during tech solutioning)
3. Designing for scale - All decision, logics and rules that are to be made, will be defined such as to keep 100% LaaP scale with end state vision in focus. The core brain and layers will not prove to be 100% iterative in design (rather rigid in architectural construct) hence it would be paramount to design Auto DML for 100% LaaP scalability while having room for variable scalability. This would mean solving for most not-iterative long term use cases in the v1 build itself.
4. Reducing manual intervention - The goal for Auto DML is to build intelligence into the system without the need for non-binary manual inputs. This is intended to reduce human errors, build higher system let computation for faster turn around on changes and reduce scale up time on new lanes.
5. Centralised business inputs: In terms of hard filters and optimisations/objectives at certain dimensions (e.g. prioritise experience over cost for new users in lane X, optimise for RTO in lane Y etc.)

Versioning

Owing to the fact that building the end state vision of Auto DML will be a large term effort and as we grow LaaP, capturing more data and automating process for data transfer are key, the Auto DML will be phased out in different version. Each version will have a milestone attached to it with a clear business use case, which should help us measure progress and success.

v1

In the v1, we intend to solve majorly for the below mentioned problem statements with a pre-diwali timeline -

- Establishing a clear PDD infrastructure for LaaP to reduce breach from **x% to y%.**
- Establishing capacity level validation and checks at lane x node level to reduce capacity exhaustion based manifestation errors from **x% to y%**
- Removing manual file upload mechanism for supply chain lane x supplier level configurations to increase OC ceiling from **20% to 100%**
- Establish an intelligent layer of network x ops tech level map to increase number of supply chains being run on LaaP from **3 to Y.**

Outlining the solution

Acronym Index

Acronym	Expanded form
DML	Decision making layer
Ops tech	Operations technology software
Orch	Orchestration layer
FM	First mile
MM	Mid mile
LM	Last mile
FMSC	First mile sort centre
MMSC	Mid mile sort centre
LMSC	Last mile sort centre
LMDC	Last mile delivery centre
LMCD	Last mile cross docks
e2e	End to end
OAE	Order allocation engine

Layers

- Seller x FM hub mapping layer

  Each seller will be mapped to a FM hub. This mapping will be at a seller ID x FM hub code level which can be a many to many mapping as well. This will be a 1x1 matrix with a maximum row threshold of 10 lakhs(At peak, each seller can be mapped to at most 2 hubs in the worst case scenario). Here's how it will look like -

Seller ID	FMH code
ABC123	GZS
ABC123	JAI
XYZ321	GZS
FGH789	GZS

- LM pincode x LMDC mapping layer

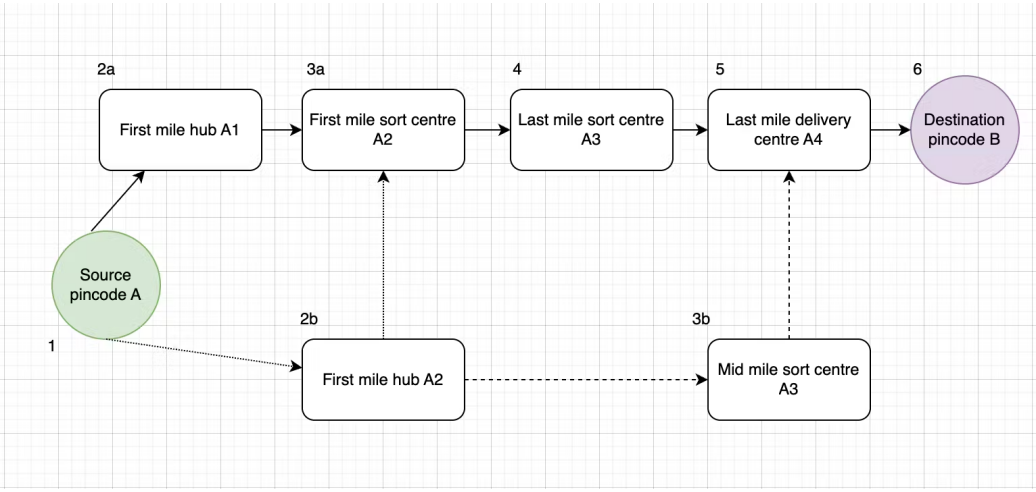
  Each LM pincode will be mapped to a LMDC. This mapping will be at a LM pincode x LMDC hub code level. This will be a 1x1 matrix with a maxim threshold of 32k rows (At peak, each LM pincode can be mapped to 2 LMDCs). Here's how it will look like -

LM Pincode	LMDC code
400706	GZS12
560048	BLR12
400604	BHI123

- Network Map layer

  This is the core layer which governs the connection of every node in the LaaP supply chain. It represents the mapping and node level attributes while mentioning the connectivity of one node to another. A node in the LaaP supply chain can be anything from the First Mile hub to the last mile delivery centre. This layer will essentially governs the route of the supply chainl. For example, a shipment from source pincode A to destination pincode B can take multiple routes. From the below diagram, it can take 3 supply chain routes -

  1. 1 - 2a - 3a - 4 - 5 - 6
  2. 1 - 2b - 3a - 4 -5 - 6
  3. 1 - 2b - 3b -4 - 5 - 6



To store the above information, A nxn matrix will be made which will represent the node connectivity with each value in the matrix storing information like TAT, Price etc. Value in the matrix represent that a connection exists and null value represents that the 2 nodes are not connected. For instance, in the below table, FMH-1 is connected to FMSC - 2 having TAT and price values as 2 days and Rs. 3 respectively. Whereas, FMH-1 is not connected to MMSC-1.

Nodes	FMH-1	FMH-2	FMSC-1	FMSC-2	MMSC-1	MMSC-2	LMSC-1	LMSC-2
FMH-1	Null	Null	null	{tat:2,price:3}	null	null	null	null
FMH-2	Null	Null	Null	{tat:2,price:3}	null	null	null	null
FMSC-1	Null	{tat:2,price:3}	Null	null	null	null	{tat:2,price:3}	null
FMSC-2	Null	{tat:2,price:3}	null	null	null	{tat:2,price:3}	null	{tat:2,p

- Hub property map layer

  The hub x property map layer will consist of a mapping between each hub(node) and the ops tech layer it is operating on. It is a 1x1 matrix having a one-to-one mapping. Here is how it will look like -

Hub code	Ops tech	FM capacity	LM capacity
FMH-1	Loadshare	100	1000
FMSC-2	Elasticrun	100	1000
LMDC-3	Fareye	100	1000
LMDC-1	Loadshare	100	1000

- Ranking engine

  The ranking engine consists of Meesho inherent logics which will help us define the priority order of the supply chain to be taken for a particular order. This will constitute input from features like pricing, TAT, capacity, weight etc. The feature values which will become an input for the ranking engine will be taken from the network mapping layer where each price, TAT, capacity key has a value which represent the connection between two nodes. For instance, the row at the intersection of FMH-1 and FMSC-2 with not Null values, represents the price and TAT for the connection beween FMH-1 and FMSC-2. Eventually, the supply chain will have many such connection and the Price, TAT will be a combination of each node values. The ranking engine will take the final output for calculating the priority.

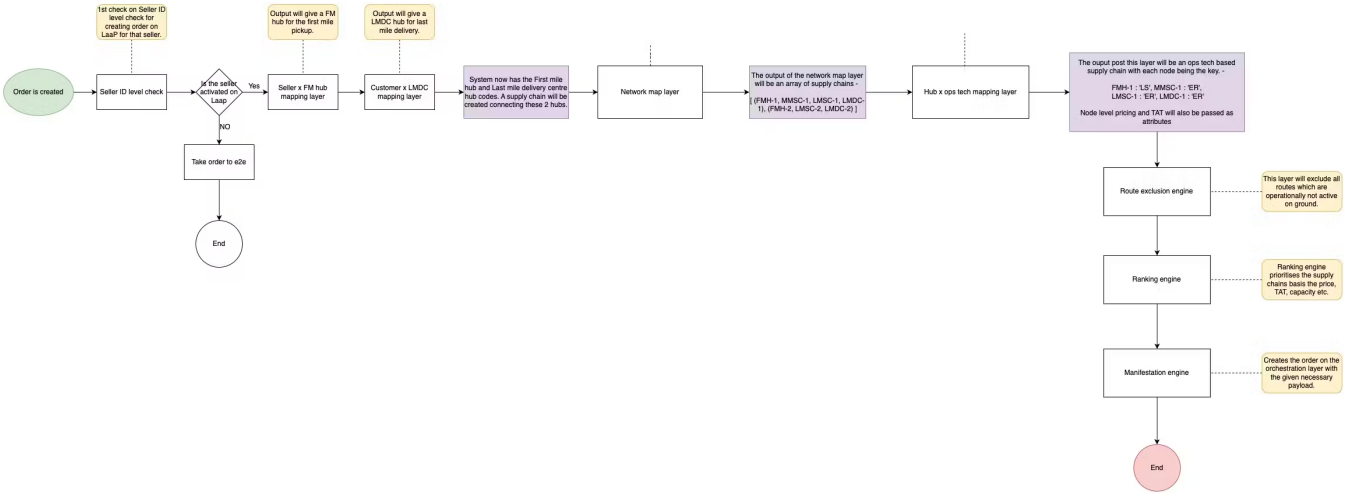
- Route exclusion layer

  A final layer which will have all the lanes which cannot be an outcome of the ranking engine if though a connection exists. This will be a hard check, manually uploaded.

- Manifestation engine

  This layer is only responsible for creating orders on the orchestration layers and catching exceptions wherever necessary to send them back to our e2e system so as to avoid any order loss.

Stitching the layers together



Key pointers

1. The property of a node will depend on the lane it has to operate on. For instance a particular hub may operate as a cross-dock for one lane and a sort centre for the other. These logics since manually driven currently, will be logically coded in the ranking engine.
2. Any given lane **will not have more than 2 sorts in a route**. This can be changes as per ops inputs but for v1, it'll be kept as such.
3. Nodes and link will have properties like capacity, price, TAT etc which will enable the ranking engine to select the best possible route.
4. Route config with node level information will be sent to all ops tech partners so that they can configure the relevant actions to be done in the given lane at a ground level. This will help us to standardize the logics being built on the ops tech by collating them on the DML layer itself.
5. A final layer which will have all the lanes which cannot be an outcome of the ranking engine if though a connection exists, will be present as a manual hard check to protect selection of non-operational lanes in real world.