Last week we had session with Clara, Raghav and Brendon. In the session, it was alluded that we are going into technical live, then we publish the outcomes to the dashboards and then Clara and Brendon can give us the feedback then we finetune the model. Once they are comfortable with the results, then we can go business live.

For some reason, Finance now has a different process on tagging PC(01:23) code. When you apply for a new PC code, they will give you a standalone PC code. You need to go into AAS approval system, appoint your approvers, and you need to link it to all the systems like CUPA(1:47), DCON.

**Eddie is now solving with Finance to raise a DCON**. The PC code comes later. Raise DCON so we can get an estimate for the work, we only need to put in a PC code after we get an estimate.

NLP:

There are 2 aspects:

1. Prepare the data pipeline. We are discussing with DST(04:12) the source of Bloomberg chats. They are sending these files to ACSS(04:20) team, which is the NCS(04:22) side. We are working out with NCS team whether DST team can send it directly to us and we do the parsing and push it to ADA, then they can consume from ADA. We are working out a plan with them. Otherwise, they would need to make a fair bit of changes because DST is unable to send the same set of files to different locations due to bandwidth considerations.
2. Prepare the model pipeline. We have started retrofitting the codes to be compatible with ADA’s structure. This is done is parallel to the data pipeline. We have taken this 94% accuracy as good for now, which we are going to implement. Once we deploy and get the manual feedback, we will finetune the model to get a better accuracy.

We are targeting to go live by end August, provided we don’t certain roadblocks on ADA. For the FX Alive, we are also hitting into some issues, in terms of some dependencies with the ADA team, mainly due to change in processes as a result of new cluster migration. We hope to resolve all of these by July and continue with the work.

This target date is for processing of chats of a daily basis. At some point, we also need to think with ACSS as to how they will bring historical data into ADA as well. If we need to retrain, for example, we need all that data in ADA. Whatever they have in the ACSS, we will get the number of snapshots and decide together as to how much we need to load in. So far, the model is being trained by 15 days’ worth of data.

We have a mockup dashboard that we presented to Brendon last time. He has given some feedback. Around that, we will kick start some discussion with Brendon and get some clarity and we will continue to work on the BI portion. On Andrew Nadi (07:32)’s side, he is also developing some dashboard on the FIX market using data from FIX backend database. We will see whether it can be an extension to that dashboard or we should have a standalone one. But since Clara is already using the FIX dashboard, it would probably make sense to extend that.

Kelvin is also testing Google and Amazon’s voice-to-text NLP. After we finish the text-to-text, and we are thinking about broadening it, then we need to connect with Kelvin.

There is a piece that OPS (10:40) is also working on, it is on Reuter’s chat data. At some point, we will sit down to see how we can create a framework around all these things. Even say voice-to-text, how we process it in the analytical core engine that we want to build. There’s some bit of POC that we might have to do, in terms of how to create a single framework for the general landscape of all these chats. As of now, we are only operationalizing what we have done for Bloomberg.

For NLP, we are also covering NLU, but we haven’t reach there yet, because as of now we are just looking at missed deal and done deal objectives, which didn’t need NLU concepts.

We have chosen to build NLP internally, because even for ACOE asset terms (12:05 – 12:06), they have also created an NLP utility. That’s the same utility that we have used. The only hard part for us is that we need to train the model according to the structure / verbatims that are used in T&M. We have not really built from scratch. We need to make sure we understand every lingo of the conversation from end-to-end. As of now, we have not used BERT. BERT is the framework that we want to use when we are thinking about the core engine for all the chats. As of now, we have used LSTM which is a neural network model. The moment we start to look at, for example, shopping behaviors from context of these conversations, we would need to build a linguistic model, otherwise our score is not going to go up more than 80% for sure.

Real Time

On Monday, we presented the near real time, based on one user use case right now on credit trading, which Ben is trying to build. But for him to achieve his business requirements, near real time is something that we need to achieve, especially since his business requirement would require data from all different sources. Eddie is very keen to find out the end-to-end time / latency from point when all the data is ready and when is it projected to the visualization / BI tool. We then explore what kind of infrastructure we will take in order to reduce the latency. We also talked about Tableau issue, because going real time now is near-impossible as a lot of things are manually triggered for the server to the Tableau visualization tool.

When the RFQ comes in and overlaps, we have a problem of matching the displayed data with the RFQ coming in on AXE Trading. **How do we ensure that, especially when they ignore some of the RFQs?** The workflow on when Ben wants to select which ISIN / bond he is interested in quoting, **Eddie has to go a little further on how he does that**. Eddie suspects that this will be a regular change on when he wants quote certain bonds and when he doesn’t want to quote those bonds.

We have to be flexible for fit-for-purpose, first, we have the analytics dashboarding, the MIS that he wants to do, our current route fits his purpose. We must treat his trading requirements on data coming up as a completely separate use case. And if data is duplicated in ELK, then so be it, because we need it to deliver.

This discussion on credit trading tool is different from the dashboard that we are building. Credit dashboard is the old tableau inline with the rest of the trading desks, that one will continue as it is. But credit trading tool is a world of its own. So we are treating them as 2 separate projects. Now we are thinking about the broad infrastructure to achieve. Tableau may not work since it requires a manual trigger to refresh all the data. Last time, we also discussed when there is a decision that needs to be done, we cannot use Tableau or any other visualization tool.

There were many solutions through, KDB, KX, ELK, etc. Whatever data Thiam Hock is scrapping to drive this, like BPipe logs, will be pumped to ELK straightaway and from ELK, we will have to build services that look into it. When it receives the signal of an incoming RFQ, it will look out for that field for customer name, then based on what traders want to see, info about customer, info about the bond, etc. This is a pre-fixed view of what are the backend data to retrieve, and is to be built on Digi markets because that’s the easiest and fastest way to extend the content to the users.

In Digi markets, they run the customer db, where we take the sales workbench information from the customer perspective, and merge together with GSIN data, the CIS data, so when user searches the customer name, over a million records is searched under half a second. From that perspective, ELK will be fastest system to pull out. We cannot use MariaDB because indexing will be extremely long.

Eventually traders’ response to RFQs would be assisted with algorithms when it comes to pricing, as we will crunch the data that comes in very quickly around all of the past transaction data, ISIN-related activities, and we will come back with some analytics.

Now, we either go with Digi markets as a frontend, to let them do the trade execution, or it will be something they are building off AXE. There is a set of architecture called UniPlotter, that they are considering, and which the other desks also can use. **By next Wednesday, there will be a demo to align everyone as to what can be done**. Based on UniPlotter architecture, how can we expand that for credit desk or even other desks to do real-time analysis.

Between the ability to stream real time charting views, versus the ability to crunch a lot of data quicker, we would choose the latter. In the T&M world, even WeeKiat don’t look at streaming data and they are not using KDB to stream tick data to that particular point. Even though Vincent has got the KDB dashboards, so far the views that ha has built are not looking at continuous streaming like a front manager, he is looking at MIS data about his TCA, where the hits at, etc.

The KDB DB is something that we must use for our future because it will come when we will take streaming information, tie them up together, for all events, data points, price ticks, orders, PnL stuff in KDB. And to be able to stream them together in a timeseries manner to paint the picture of series of events. We haven’t come to this point yet, but it will come. So for KDB’s use cases, we need to get in all these data in there and store them as time series.

From a Hadoop perspective, while we can store a lot of data and time series’ value, it is the regurgitation part that is not up to par with KDB. The use cases are different.

We are definitely underutilizing KX, as only Vincent’s desk is using it, and he’s using it like how Tableau can also be used. So internally, we are not creating a use case when we are building our own real time streaming data. From Vincent’s UB, it is supposed to go into his price queuing algo to drive the price queuing, and he’s looking at the result through the trader’s price rate (50:34). He is not tracking the skew through a chart.

In the customer’s perspective, we got sales views and customer views where there are recommendation, etc, look at aggregating costs and funding, look at contextual marketing. For recommendation, we need market data to back test or to derive some results of contextual marketing.

**Need to give Elene (54:20) an idea of how much it costs to store certain data.**

We have already done the testing of data streaming from Kafka to Druid, which is using Spark as well. We stream the data from Kafka, and put it in a Druid store, and Tableau is connected to Druid. If we want to tap on trades which are not fast moving data, then we can go that way as well.

**We need to get Andrew Tan to prioritise ingestion of Bloomberg chats into ADA.**