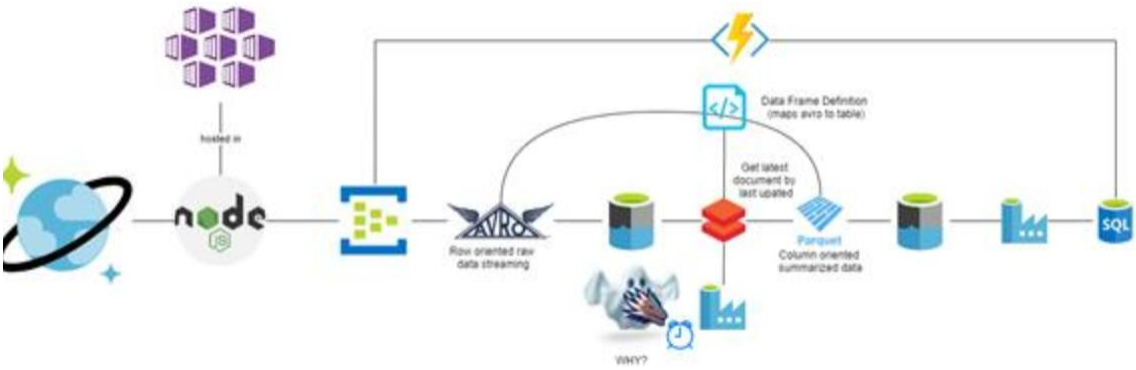
Albero: Halloween Stories

Once upon a time there was a high-throughput HTAP system. It was based on some well-known technology. Engineers decided to bring this system to the cloud and Azure Synapse was chosen as the destination technology. The PoC, MVP and main migration went smoothly (almost 😉). Suddenly technical teams started experiencing degrading performance of the entire system. Needless to say, that the users were not very happy with this. After careful consideration they discovered that growing commit queue was causing performance degradation. Having reviewed of the code, they found out that individual commits were the main root cause for the issue. So, they understood that MPP system is not the best option for individual transactions. Luckily, nobody got hurt in the process.

You know what happens when some people start architecting? They invent new things. Sometimes they are good, sometime useless. And sometimes these things could be implemented even without writing the code. The trick is to know what is available and which integrations exist. Look at the picture below. Did you know that you can implement it without coding just using Azure Cosmos DB Link for Synapse?



Storing all the data “forever” is one of my favorite business requirements. One company just took this approach too directly with Cosmos DB. They never did cleanups so their main collection was growing organically over the years. More data – more partitions. More partitions – more RUs required to support the same performance level. More RUs – more cost. “What for are you using all this stockpiled data?” – they were asked. “We don’t” – was the reply. Cool, right? At least they were able to refer to their collection as, probably, one of the largest in the world.

I had a good friend of mine who was trying to find a ~~job~~ database of his dreams. He had no other choice but to try all of them one by one in production. After trying 16 databases over several years they found something pretty close to what they wanted. The bad thing was that due to the cluster-side replication of the technology selected (which apparently was discovered) they could not use more than 3 nodes. So

they scaled the VMs up. And up. And once again. Until they maxed out on RAM. No, they didn't buy the mainframe. They discovered that there is Azure Data Explorer and that is not a standalone application. Happy end!

Kafka is a new and shiny thing. It is everywhere. It can solve all our problems with data ingestion! Right? Almost. At-least-once delivery is still there. There was a nice story when one company decided to be very innovative and bring their Kafka to a new level. Actually, even two levels. They have implemented (at least tried to) a multi-regional Kafka cluster with cross-reginal replication. To make it more fun they put Kafka into containers. Now they are very busy every day 😉 Curiosity killed the cat (and in this situation the cluster) so now they are going to the other extremity PaaSing everything. Scary.

If there were no problems, we would have to create them. And what is the best way to do so? Of course, to define some cool business requirements. One organization did exactly that. Their business division just decided to set an NFR to have processing, analysis and down streaming of incoming data within 100 ms. Do not ask me why. The coolest thing is that one of our teams helped them and they reached number which is a bit higher than this "business" requirement but at the same time way less than a second. Of course, that was done with Databricks. Everything can be done with Databricks, right, folks 😉 The small problem was the price. And not only for the compute itself but also for extraordinary amount of requests to Storage accounts. I bet that we will need a separate AI model just for cost predictions for running the first model 😉