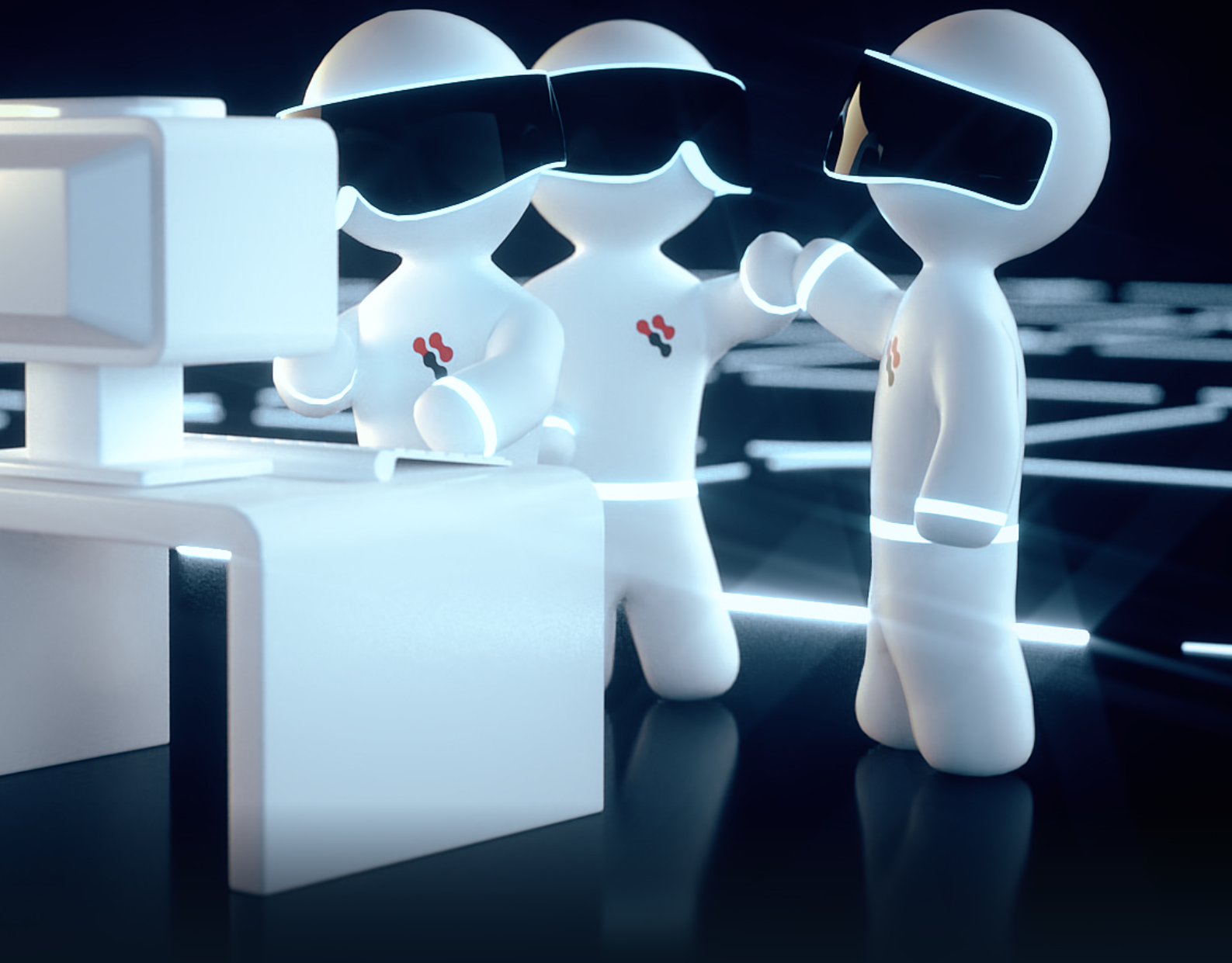
dotfive™



Salt

Next-gen data analysis platform, for identifying relationships and effects to give insight into performance and projection.

Next-gen data analysis platform, for identifying relationships and effects to give insight into performance and projection.

Salt.io is a technology and data analysis company, offering tailored solutions in data strategy, system integration, business intelligence, and similar areas.

As part of their continual work in data modelling and prediction, they identified an opportunity to develop a new product in the market, and approached us due to our extensive experience in these areas of technology.

BACKGROUND

Salt.io deal with a wide range of data relating to their clients' interests. This includes data on financial products and assets like property, to help them gain insights into trends and performance. Salt wanted a way of analysing this information that would go beyond simply mining the data itself and would identify relationships across a number of data attributes, the strength of the relationships, and the effect of those factors when trying to answer certain key questions for the purpose of establishing trends and performing predictive modelling.

A critical, fundamental requirement was that this new system should be extremely flexible and adaptable for different purposes, so that it could be quickly and easily configured for different data types and situations, and able to still perform the same analysis. This creation of a generic yet powerful tool would allow them to carry out data interrogations for any type of client or scenario, and therefore be a major boon for their business.

An example goal of this project was to determine which parties and properties have the most influence on the property market, especially when valuing property and buildings. If a party takes a certain strategy on a property and has a lot of influence, it would be safe to assume that the same strategy is going to be copied and re-used throughout the market for other similar properties, according to certain related factors such as locality. So the system should be able to narrow down which properties should be watched closely for the strategy to emerge, because that will signal a future trend — something which is incredibly valuable for investors.

Salt

CHALLENGES

The primary challenge here was creating an appropriate way to describe, model, and present the data. We were provided with the data by Salt.io, but we needed a method to flexibly project that data after loading it into the system, so that meaningful questions could be asked of it, and answered. If the system had a single purpose this would be complex, yet straightforward enough — the fact that this system had to be adaptable for any given situation made it much harder, as we had to invent a way in which to describe and configure the environment in order to appropriately import, store, and interrogate the various datasets.

In addition, the way in which the data changes over time is also very important to the nature of the questions being asked. The system needed to be able to allow analysis to be carried out any any point in the history of the dataset as well as against the current state. This aspect is critical for training machine-learning algorithms and comparing new approaches against existing ones, and for spotting trends. Being able to fully track the journey of the data, including movement and transformations along the way, is known as data lineage, and was therefore a key requirement of the system. This adds another layer of complexity, and the difficulty here was handling this correctly in addition to the other requirements, whilst not over-complicating the interaction with the system.

We reduced a task that previously would have taken several days down to just 15 minutes.

SOLUTION

We decided the best style of database was a graph database. We put together a system that helped map the raw data onto this graph using projections, and we provided data enrichment through calculated and aggregated metrics. We also developed a method to associate product entity data that was otherwise siloed, to ensure a broad relationship context for the intelligent retrieval of relevant data.

We created a command model, with its own small language, which reflected the business logic of actions taken with financial products. The commands could be processed either in real-time or in batches, and every event happened to data entities was captured, enabling a history to be built up. This history could then be used to run projections.

In addition to the command language, we also created a configuration structure which allows almost any dataset to be modelled onto the data fabric system. This means that new scenarios can be quickly sketched out and configured, and then data loaded in and queried without needing any custom development.

For many projections, time is an important factor, as many entities (such as the balance of a loan) will have different values across a range of dates. The system allows such fluctuations over time to be analysed in addition to the other data events, and the effect of these on the relationship to, and impact on, other entities can emerge. A good example here is that if a company has rising debt, that will affect the actions of other companies dealing with it, and indeed its abilities to take certain actions in the market.

The reliability and scalability of the system was very important, and so we used a messaging queue to ensure that if a projection was ever down, all the data changes would still be queued, thus improving resilience. It also reduced the continuous load on projections, helping manage performance.

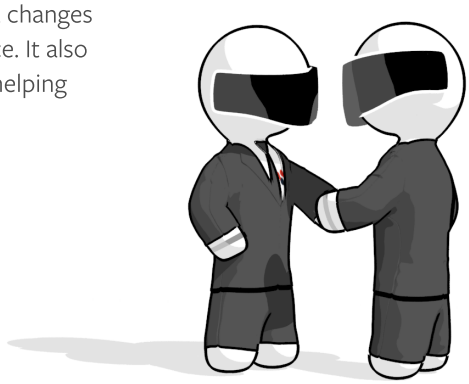
RESULTS

We worked closely with Salt.io to streamline the projection process, so they simply have to provide a configuration and will then be presented with the relevant results. We ensured that only the projection data directly relevant to the properties in the configuration would be shown, to mitigate against the projections becoming overloaded.

This has had a phenomenal effect on their business, as they are now able to create new models and scenarios very quickly and easily, and immediately start gaining insight into customer datasets.

Extending the command processes is also just a case of quick configuration. This allows analysts to rapidly create customised commands for specific datasets, and in providing this, we reduced a task that previously would have taken several days down to just 15 minutes.

Although the product is not yet complete, and we are still continuing to work with Salt.io in further developing it, our efforts to date have already resulted in significant and measurable success. The system delivers the desired results and also already allows our client to rely on it to generate income and advise others on making investments and critical commercial decisions with a range of assets.



dotfive™

www.dotfive.co.uk
hello@dotfive.co.uk
+44 (0) 8456 808 805
+1 (307) 222 4842

dotfive limited | frontell house, west coker hill, west coker, yeovil, somerset, ba22 9dg
dotfive inc. | 1908 thomas avenue, cheyenne, wyoming, 82001-3527

Technologies used:

NODEJS TYPESCRIPT PYTHON RABBITMQ NEO4J

NodeJS | Typescript | Python | Postgres | TimescaleDB | Neo4J | RabbitMQ | Amundsen | Airflow | JSON
AWS | CI/CD | CLI

dotfive™

www.dotfive.co.uk
hello@dotfive.co.uk
+44 (0) 8456 808 805
+1 (307) 222 4842

dotfive limited | frontell house, west coker hill, west coker, yeovil, somerset, ba22 9dg
dotfive inc. | 1908 thomas avenue, cheyenne, wyoming, 82001-3527