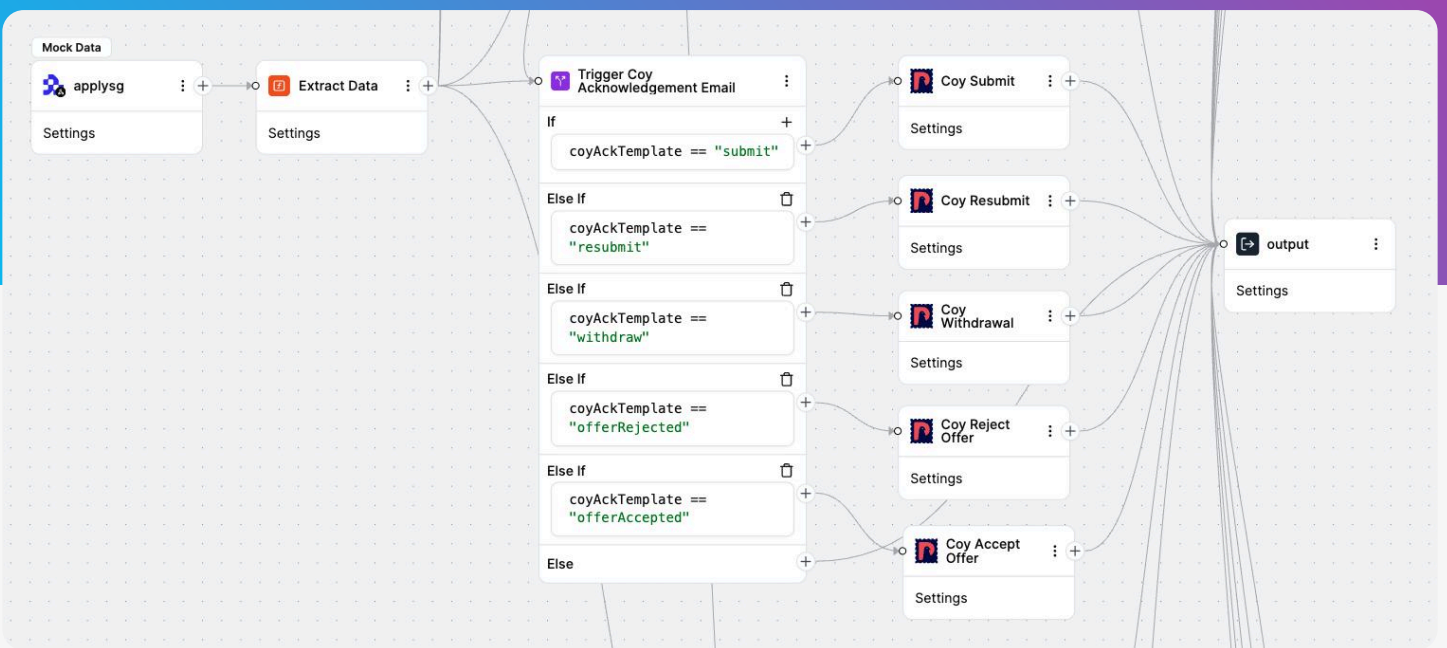


# How WSG Simplified Interest Form Submissions Across Multiple Programs



## What was Opus used for?

WSG was looking to solve a key challenge: ***How might we facilitate a tell-me-once approach and streamline the way companies submit their interest across different programs?***

In the existing setup, each program operated in silos, with its own form and process. As WSG rolled out a new programme, there was also a need to ensure that cases were distributed fairly amongst partners, adding a layer of complexity through the need for load balancing.

With Opus, WSG brought its redesigned process to life. Instead of maintaining multiple forms, the team moved toward a more unified submission flow. Users can now submit their information through a single entry point, while Opus handles the creation and routing of the required cases to the appropriate teams. At the same time, the ability to handle complex load balancing logic ensures that the way cases are distributed addresses the need for fairness in the new operating model.

*“Don’t be afraid to try. It doesn’t have to be a large-scale solution; even simple workflows can help streamline your processes.”*

Benedict Ang Digital Technology Division (WSG)



Beyond a faster time-to-market, Opus has also enabled greater flexibility in how workflows are designed. By leveraging reusable components and child workflows, the team can build processes in a more modular way, without having to rebuild from scratch each time. This has allowed them to continuously iterate and expand their workflows, starting with load balancing and gradually incorporating additional programs and use cases.

Overall, Opus has enabled WSG to move away from fragmented, manual processes toward a more streamlined and responsive way of working. Instead of rebuilding systems for every new requirement, the team can now adapt quickly, scale across multiple programs, and continuously improve their workflows.



## Tips for Teams Getting Started with OPUS

For teams who are interested but unsure where to begin, the advice is simple: ***start small and try it hands-on.***

A practical approach is to experiment with small setups first by stringing up individual nodes you want to learn, starting with an input, the node you want to try, and an output, so you can quickly understand what each node can do.

