CASE STUDY

When Legal Documentation Becomes the Operating System for Debt Investment

Sector: Debt Investment / Lending Operations / FinTech Infrastructure

CASE BACKGROUND

For most borrowers, a loan agreement is something they sign at the end of the process.

For a debt investment fintech, that agreement is the process.

It decides who is borrowing, who is guaranteeing, what has been pledged, what happens if repayment fails, which clauses apply, which signatures are valid, and whether the document can stand up legally if challenged later.

That was the context behind this project.

The organisation sat between investors looking for higher-yield debt opportunities and companies looking to raise capital through structures such as non-convertible debentures, equipment purchase finance, equipment lease finance, loan against property, joint development agreements, and other structured products.

Each deal depended on legal precision.

A borrower's name had to match everywhere. A co-borrower's exposure percentage had to map correctly. A guarantor clause had to appear only when relevant. Signature boxes had to sit on the right page. A clause that applied to one product could not appear accidentally in another.
In this business, documentation was not paperwork after the transaction.

It was the transaction's legal backbone.

THE PROBLEM

Why the Earlier Process Could Not Support Scale

Before the platform was built, legal and operations teams were working with large sets of Word documents and Excel files. For every product, there were multiple document sets. Each set could include 30 to 40 documents. Each document could run from a few dozen pages to well over a hundred pages depending on the structure of the deal.

Only a portion of each document changed from one deal to another, but that portion was critical. The clauses usually remained stable. The data did not.

Borrower names changed. Co-borrowers changed. Promoters changed. Collateral details changed. Equipment details changed. Exposure percentages changed. Product structures changed. In some cases, the number of borrowers itself changed, which meant tables, clauses, signatures, and references had to adjust dynamically across the entire document pack.

That is where the risk increased.

A missed name change in one section could create ambiguity. A signature block moving to the next page could create room for misuse. A conditional clause appearing where it should not could weaken the document. A template copied from an older transaction could carry forward language that no longer applied.

For an organisation operating in a regulated debt environment, these were not formatting mistakes.
They were legal and operational risks.

The Real Problem Was Contract Variability

The organisation did not need a basic document generator. It needed a system that understood how debt products behaved.

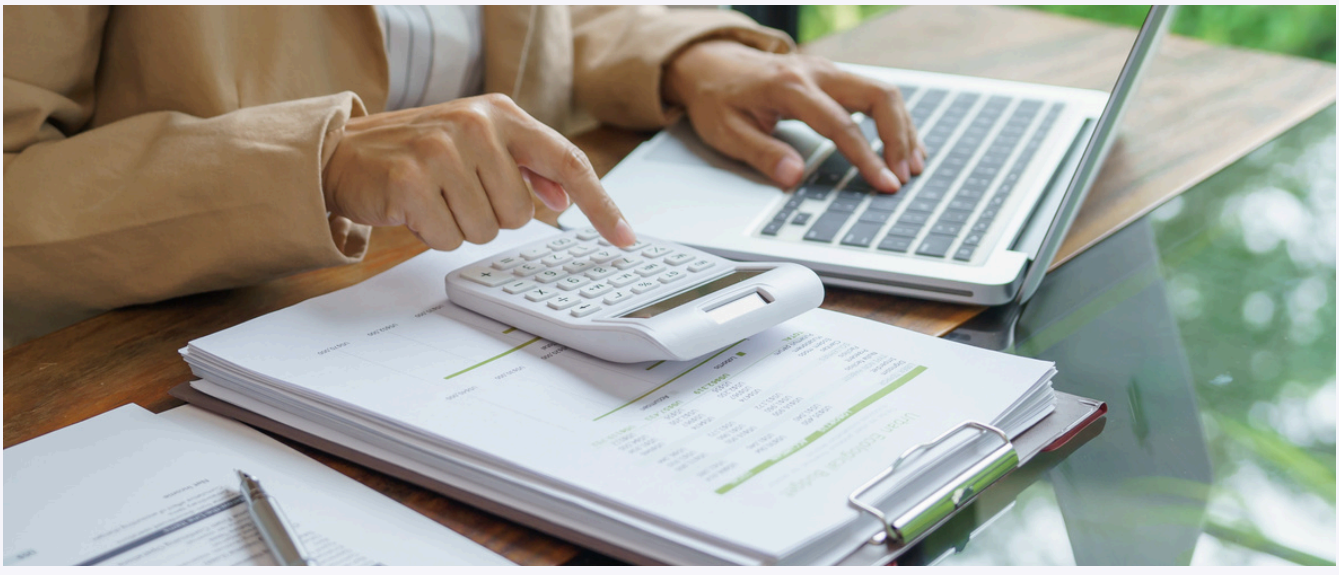
Each product carried its own documentation logic.

An NCD structure required one set of legal documents. Equipment purchase finance required another. Equipment lease finance followed a different structure. Loan against property involved collateral-linked documentation. Joint development agreements introduced their own legal and commercial variations.

Even within the same product, no two deals were identical.

One borrower could become three. One guarantor could become five. A single entity could become a group of entities. A clause could become conditional. A table could expand based on the number of parties involved. A signature section could require strict alignment to preserve document enforceability.

The challenge was to convert this variability into a controlled workflow without flattening the legal nuance that each product required.



THE APPROACH

Building an Intelligent Template Infrastructure

We helped build a platform that converted legal document preparation into a structured, template-driven system.

> *The first step was separating static content from dynamic content across product-specific document packs. Legal clauses that remained constant were preserved. Deal-specific fields were converted into controlled placeholders. Conditional sections were mapped based on product logic and borrower structure.*

This made the templates intelligent.

If a deal involved multiple borrowers, the system could populate the relevant names, exposure percentages, tables, references, and signature sections correctly. If a clause applied only under specific conditions, it could be included or excluded based on the underlying data. If a document required a particular paper size, format, watermark, or PDF output, the generation flow accounted for it.

The objective was never to remove legal oversight, but to remove repetitive legal drafting.

The organisation wanted a system that could automatically generate nearly the entire document pack while preserving human judgement where it mattered. In practical terms, the target was straightforward: **automate roughly 99% of the documentation process and leave the final 1% to legal reviewers responsible for validation, exceptions, clause-level refinements, and approval.**

Once generated, documents could be routed to authorised reviewers through role-based access controls. Reviewers could make necessary edits, decide whether a change applied only to a specific contract or to the underlying master template, and then lock the document for final PDF generation and external circulation.

This ensured that automation improved speed without compromising legal accountability.

THE SOLUTION

Why Access Control Was as Important as Automation

The documentation involved sensitive financial and legal templates. Allowing these files to sit on individual laptops or circulate as editable Word documents created a serious information security risk.

A template library of this nature carries business value. If copied or leaked, it could expose proprietary legal structures, product logic, and operating methods.

The platform therefore had to control not only document generation, but also document access.

Documents and templates were moved into an online, access-controlled environment. Users could work only within defined permissions. Reviewers could access editable versions where required. Final documents could be locked and converted into PDFs before being shared externally.

Auditability became central to the design.

The system needed to answer questions such as: who changed a template, who viewed a document, who edited a clause, who generated a PDF, and whether the change applied to one contract or the master template itself.

That distinction mattered because changing a live contract is different from changing the template that future contracts will use.

THE IMPLEMENTATION

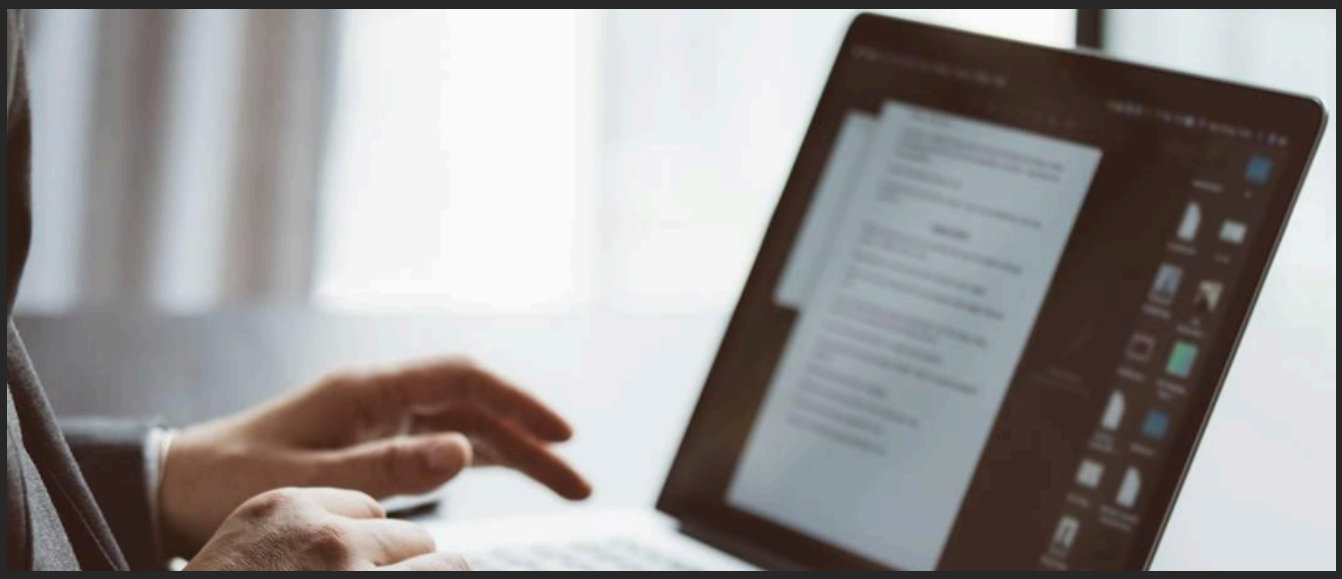
Why Google Docs Became Part of the Architecture

The team chose Google Docs and Google APIs to manage document templates because the platform required online collaboration, access control, and controlled document manipulation. But the implementation was not a simple case of storing documents on Drive.

The system had to manipulate templates programmatically. Placeholder replacement had to be precise enough to avoid disturbing legal language. Dynamic sections had to expand cleanly. Tables had to respond to borrower structures. Signature placement had to remain enforceable.

Generated documents had to convert into locked PDFs with the required formatting.

This required working directly with Google Docs APIs and building logic around how legal-financial documents behave in real conditions, not only how templates behave in theory.



THE TRANSFORMATION

What Changed in Practice

The platform fundamentally changed where legal and operational teams spent their time.

Previously, lawyers and operations staff were spending significant effort recreating documents, copying information across templates, checking borrower details, updating clauses, validating signatures, and reviewing multiple versions of largely similar contracts.

With the new system, most of that repetitive work moved into the platform itself.

Product-specific document packs could be generated automatically using structured inputs, predefined business rules, conditional clauses, dynamic borrower structures, and controlled templates. What previously involved weeks of drafting, checking, correction, and back-and-forth review could now be completed in minutes.

More importantly, the operating model shifted from manual preparation to exception-based review.

> *Approximately **99% of the documentation workflow** could be **automated** through templatisation and document intelligence, allowing legal teams to focus on the final **1%: reviewing** deal-specific nuances, validating contractual intent, approving exceptions, and ensuring enforceability.*

The platform also gave the organisation a foundation it could own internally. After the initial implementation, we helped transfer knowledge and guide the creation of an internal capability capable of maintaining and extending the platform independently.

The result was not simply faster document generation. It was a documentation process that became scalable, auditable, and significantly less dependent on manual effort.

KEY TAKEAWAY

The Larger Lesson

For a debt investment fintech, documentation is not a back-office activity. It is the legal infrastructure that protects investors, borrowers, and the business itself.

As the organisation moved from a smaller investor base toward larger, more regulated debt-market operations, the old model of manually editing document packs could no longer support the level of precision, auditability, and speed required.

We helped convert that process into a controlled documentation platform: one that could handle multiple debt products, dynamic borrower structures, conditional clauses, access-controlled reviews, audit trails, and final PDF generation without losing legal discipline.

The result was a system that allowed the organisation to scale its debt documentation process without scaling legal risk at the same pace.

JOSH

That is what this project really solved.

A centralised, governed, searchable system – built to assist rather than obstruct.

www.joshsoftware.com | contact@joshsoftware.com | +91 7887889902