



Upload



Extract



Code



Calculate



Results



Certificate

Covenant

AI-extracted co

EBITDA An

BASE
Con
Claus

Final

COVENANT

Leverage Rat

Axiom: The Covenant Logic Compiler

Bridging the "Legal-to-Code" Gap in Syndicated Lending.

LMA EDGE Hackathon

LMA

@anujxforge - X



The \$100M Problem

Manual "Stare & Compare"

Translating 300-page legal PDFs into Excel formulas.

49-Hour Redundancy

Weeks of duplicated work repeated across 15+ banks.

Human Error Risk

Manual "English-to-Math" translation leads to \$100M errors.

Problem Statement



www.axiom-complier.vercel.app



The AI-First Compliance Standard

Our Vision



Bridge the Gap

Automated translation of legal text to math.



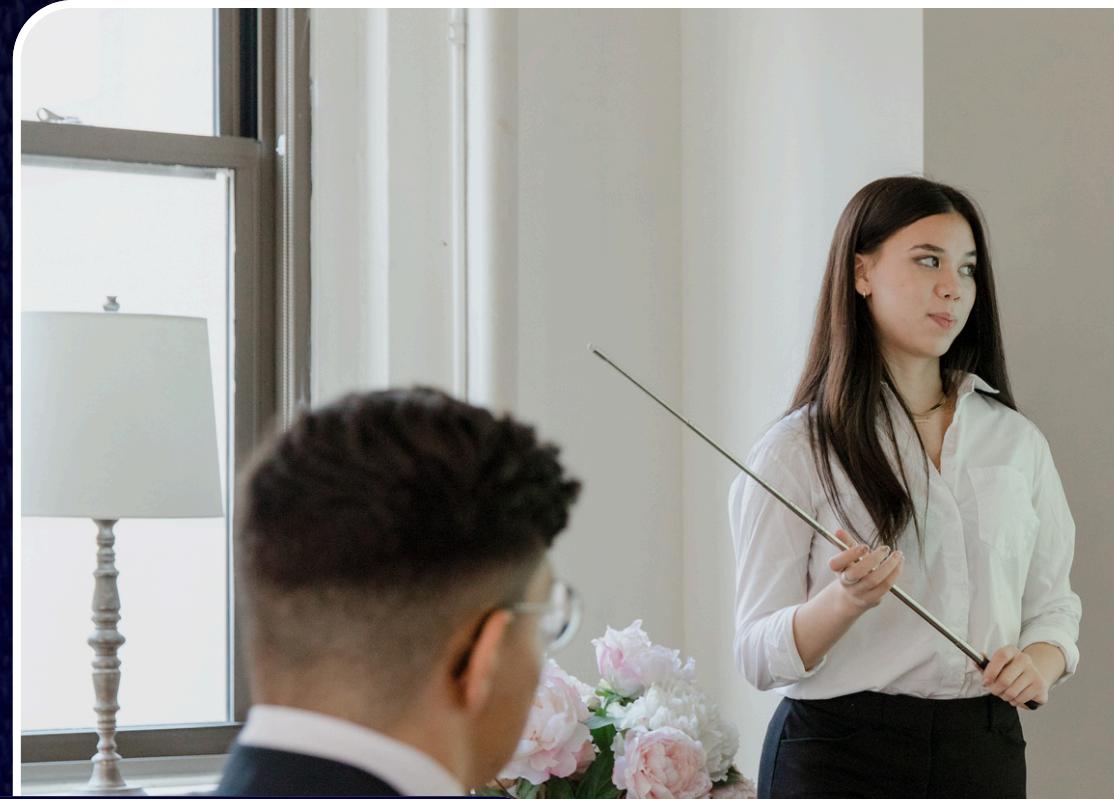
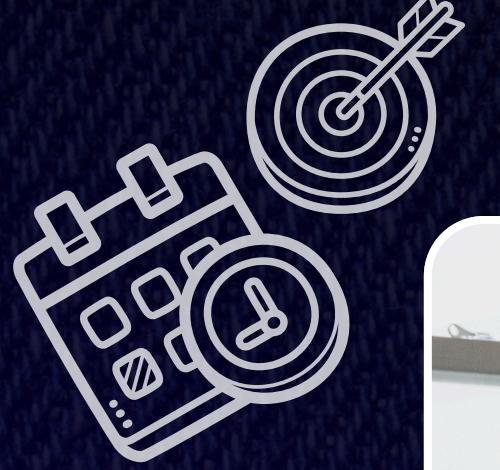
One Source of Truth

Shared calculation logic for every lender.

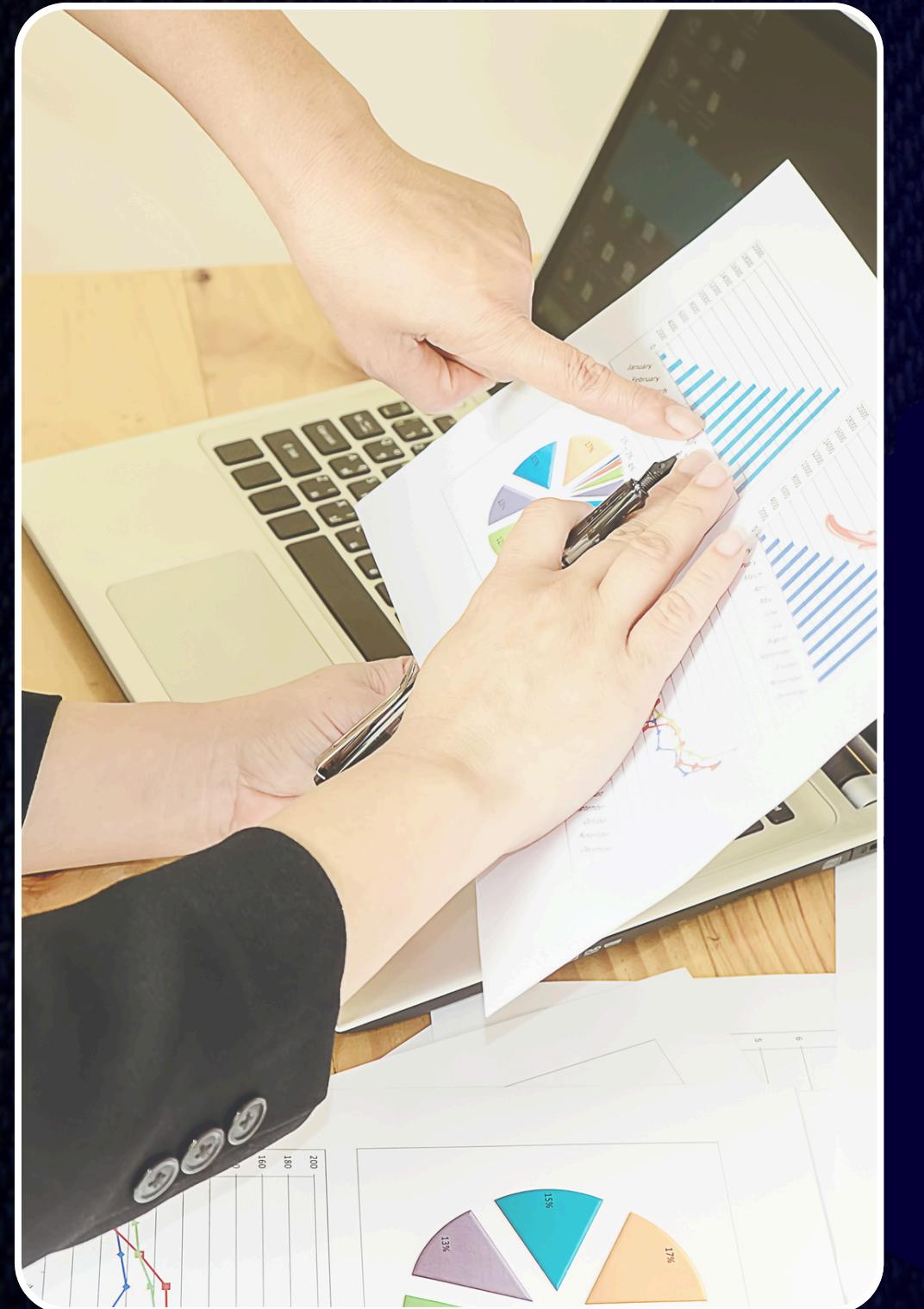


Audit-Ready

Every compliance check linked to its source clause



Survey of consumption from all over the world



Our Solution

From PDF to Verifiable Code

Smart Extraction: RAG-based AI reads 300+ page contracts.

Logic Compiler: Generates auditable Python code for every rule.

Instant Verification: Real-time compliance testing against financials.



Business Model



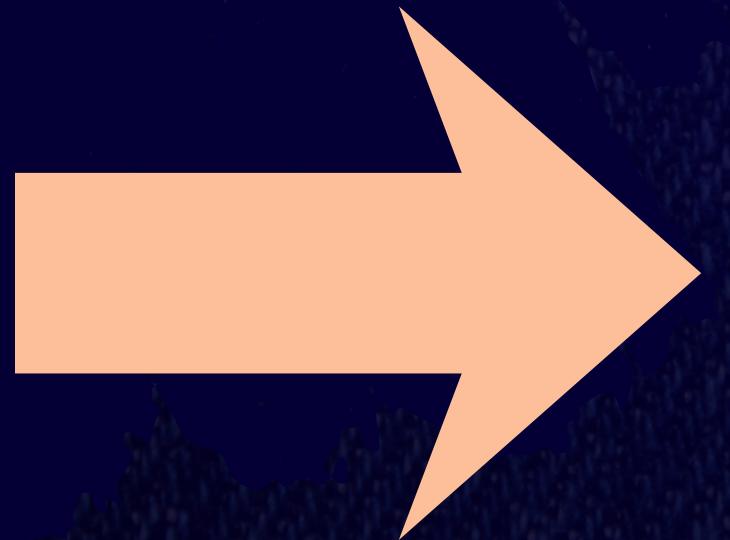
Our revenue streams are driven primarily by platform subscriptions, serving as the main recurring income from agency banks and lenders. This is complemented by certification fees applied to each compliance report, covering automated verification and calculation costs. Together, these create a scalable Software-as-a-Service model that balances loan volume with precise, automated risk management, ensuring both growth and operational excellence.

Competitive Advantage

- Beyond Field Extraction: We don't just "find" text; we write the math.
- Syndicate Synchronization: One source of truth for every lender in the pool.
- Full Auditability: Every percentage and dollar linked directly to the contract clause.
- Built for LMA Standards: Precise handling of complex "Bucket" caps and EBITDA add-backs.



Let's move on to Demo



```
    iter = self._key_value_iterator()
    statuses = {}
    for data in resp_iter:
        status = Status(
            status_id=data.id, name=data.name)
        statuses[status.name] = status
    return statuses
```

Thank You Very Much!

Contact me

LMA



www.anuj846k.dev



India

