LeaseSync: AI-Powered Lease Management & Document Processing

Bravyto Takwa Pangukir, Markus Kevin, Pascal Christian

Abstract. LeaseSync is an AI-powered lease management system that automates document extraction, classification, and tracking for businesses handling large volumes of lease agreements and related documents. Traditional lease review processes are manual, costly, and error-prone. Our solution uses Large Language Models (LLMs) with expanded context windows that can parse entire multi-page documents in one go, alleviating many of the limitations inherent in older approaches like basic OCR or chunk-based retrieval. By merging robust OCR techniques with LLM reasoning and prompting, LeaseSync provides fast, accurate, and cost-effective extraction of critical data, enabling businesses to automate previously manual document workflows. This white paper outlines the foundation of LeaseSync, including its advantages over OCR and general Retrieval-Augmented Generation (RAG) systems, a cost analysis illustrating up to 99% savings, and its potential for broader applications beyond lease management.

1. Introduction

The Growing Challenge of Document Processing

Businesses that manage many leases, financial contracts, or legal documents face an escalating challenge in reviewing and tracking these documents. Many companies still maintain documents in paper form or unstructured PDFs, making data extraction a cumbersome bottleneck.

Lease agreements often span dozens of pages with dense legal language and varied non-standardized formatting, requiring careful extraction of key terms. Manual processing is time-consuming and prone to human error, leading to missed details or inconsistencies. In addition, for each lease, there exists multiple of related or derivative documents, such as lease amendments, payment invoices, tax withholding stubs, real estate agency agreements, etc. These need to be manually reviewed and matched to the "parent" agreement. After certain volume, we personally struggle to keep up with this complexity, resulting in slow turnaround times and high error rates.

LeaseSync as a Solution

LeaseSync addresses these challenges by automating the analysis of lease agreements and related documents using AI. It can ingest large sets of documents and instantly extract key details such as lease duration, rental rates, payment schedules, and involved parties. Unlike basic OCR that only converts images to text, LeaseSync uses a powerful language model to understand context across long documents, preserving the meaning of clauses and definitions. By automatically classifying document types (distinguishing a lease from an amendment or an invoice) and grouping/linking related documents (payment invoices, tax withholding stubs, real estate agency contracts), LeaseSync provides a cohesive view of all lease-related records. This significantly cuts down manual work, reduces errors, and frees up staff for higher-value tasks.

Scope of This Paper

This white paper overviews the historical evolution of document parsing, from early OCR to modern LLMs, and provides details on how LeaseSync integrates these techniques to address the limitations of methods such as chunk-based RAG. We also explore cost and performance implications for real-world deployments, emphasizing the Indonesian market and context. Finally, we consider security, deployment, and future directions, illustrating how the LeaseSync framework can be extended to other domains.

2. Historical Context: Document Parsing & OCR

Early Document Processing Methods

Early document processing solutions relied heavily on Optical Character Recognition (OCR) to transform scanned pages into text. While OCR was a breakthrough, it was prone to errors if the document quality was poor—faint print, smudges, or distortions could easily result in misreads. Moreover, extracting meaningful, structured data from raw text typically depended on rule-based heuristics, which are fragile when dealing with diverse non-standardized layouts and wordings. Lease contracts in Indonesia can differ significantly in format, language style, and clause ordering, making purely rule-based approaches unwieldy. This meant that manual review remained the norm for accuracy.

Large Language Models as a Game Changer

A pivotal shift occurred with the introduction of Large Language Models (LLMs). LLMs can read and comprehend long passages of text, enabling structured data retrieval from unstructured documents in a way previously not possible. Instead of relying on a fixed set of rules, an AI model can be prompted to “find the lease start date” or “identify all parties in this agreement,” and it will interpret the entire document to provide an answer. This capability is transformative for contracts and leases: the AI can consider definitions, context, and nuance, rather than just doing pattern matching.

Early versions of LLMs (e.g. GPT-3, Llama prior to 3.1) allowed for this nuanced understanding and summarization but were limited to context windows equivalent to just a few pages of text. (In LLMs, “context windows” refer to the processing “memory” for a particular task. Early versions only have a “memory” of few pages of text, limiting their usefulness.)

Recent LLMs boast substantially expanded context windows, often tens or even hundreds of thousands of tokens—enough to handle hundreds of pages. This evolution is transformative for document processing. Rather than chunking, the model can read an entire lease, preserving continuity. When combined with robust OCR, these models can even overcome transcription errors or spelling mistakes. They generate coherent outputs—extract of key fields—despite potential text noise from low-quality scans. Such resilience is particularly valuable in Indonesia, where inconsistent document scanning (sometimes captured from mobile phone cameras) remain prevalent.

3. Limitations of RAG in Long-Form Document Processing

How Retrieval-Augmented Generation (RAG) Works

Retrieval-Augmented Generation (RAG) is a technique that allows AI and LLMs to access "external" data outside of what was available during their training. It is the most popular method of enabling AI and LLMs to process external documents. Generally, this is done by splitting large external documents into smaller "chunks" and using a retrieval mechanism (often a vector database) to pull only the relevant chunks for the language model to process. It originated as an effort to harness large corpora—think entire knowledge bases—without overloading the LLM with text.

In a typical RAG workflow, each chunk might be a few hundred words of text. These chunks are stored in a vector database, so when you need information (like answering a question about the document), the system finds which chunks are likely relevant and supplies them to the AI. This allows the language model to operate without exceeding its maximum context length and can speed up responses by narrowing the scope of text it considers. RAG combines strengths of information retrieval and generation: it can handle large knowledge bases by focusing only on pertinent snippets. In theory, this seems ideal for lengthy leases or collections of contracts, because you could query details without reading the entire text into the model.

Core Limitations of RAG

While RAG is powerful, it has critical limitations when applied to long-form legal documents like leases. The act of chunking text inherently risks losing global context. Important details can be "stranded" in a different chunk that isn't retrieved, leading to answers that lack completeness. For example, a lease agreement's first page might define the parties (landlord and tenant names). If a later clause on page 5 refers to obligations of the parties, the RAG system that is tasked to retrieve "obligations" will obtain the text of that clause only, preventing the AI model from understanding who "the Party" refers to. Hence, a RAG system might correctly pull a clause about renewal terms when prompted but fail to mention to whom those terms apply, because the definition of the "Lessor" or "Tenant" was in a different chunk.

(Over time, researchers added improvements—like re-ranking or dynamic chunk grouping—to preserve context, but these add complexity and still can't replicate the continuity you get by reading a document as a whole.)

Another limitation is handling related documents. Leases often come with amendments and associated invoices or payment schedules. A traditional RAG setup treats each document independently in the retrieval index. It might retrieve text from the main lease or from an amendment, but not easily realize that a particular amendment corresponds to a specific lease (unless the user's query explicitly links them). This can lead to confusion or missing links—amendments may be overlooked or an invoice might not be recognized as part of a lease's data.

In short, RAG can struggle with the relational aspect of lease portfolios, where multiple documents collectively define the full picture.

LeaseSync's Solution

LeaseSync avoids the pitfalls of RAG by leveraging an LLM with an extended context window and a purpose-built relational pipeline designed for end-to-end understanding. Rather than splitting a single lease into many small pieces, LeaseSync feeds as much of the lease document as possible into the AI model at once. This requires a modern model that can handle extremely large inputs, something that is only possible relatively recently.

Our chosen model (Amazon Nova Pro) can handle extremely large inputs – up to 300,000 tokens in a single context (equivalent of 500 pages of texts). This means even very lengthy agreements or a batch of related documents can be processed together without chopping them into disjointed chunks. The result is that the AI has the full context when extracting information. Definitions from page one are "seen" by the model when it's analyzing page five, so it never loses track of who "the Tenant" or "the Lessor" is. These semantic extracts are then stored in a specialized relational database, rather than general vector database, enabling more easier query and retrieval.

For linking leases with their amendments and invoices, LeaseSync's workflow intelligently groups documents, and combine RAG-style search with relational database queries. If a user uploads a lease contract along with one or more addendums or subsequent invoices, the system will query the relational database to identify their relationships (for example, matching a landlord ID or property address across files). All related documents are processed in one go or in a coordinated manner, so the model understands the hierarchy: parent lease and its child documents. This ensures that extracted data like "Lease Expiration Date" or "Monthly Rent" is attributed to the correct base lease, and any changes from amendments are noted in context. In essence, LeaseSync replicates the workflow of how a human reviewer would read through an entire folder of lease paperwork. By maintaining global context and document linkages, LeaseSync delivers more accurate and complete outputs for than chunk-based RAG methods.

4. LeaseSync: AI-Powered Lease Management Workflow

End-to-End Workflow

LeaseSync is designed as an end-to-end solution that takes raw documents and produces structured, actionable data. The workflow consists of several automated steps:

- **Batch Document Upload:** Users start by uploading files (PDFs, images, Word documents, etc.) to the LeaseSync system. They can upload a mix of document types in one batch – for example, a main lease agreement plus several related documents like amendments, termination letters, or rent invoices. Upon upload, each file is stored securely (e.g. in an object bucket such as AWS S3) and queued for analysis. The system can handle even dozens of files in a batch, making it scalable for portfolio-level processing.

- **AI-Driven Processing:** Once uploaded, the AI engine kicks in. First, an OCR component (Tesseract in the current stack) converts any scanned pages or images into text. Then the Amazon Nova Pro model (via AWS Bedrock) analyzes the text to determine the document type and extract key information. It classifies each document as a lease, an amendment, an invoice, etc., by looking at telltale signs (for example, a document titled "Lease Agreement" or an invoice layout with tables). Documents that are related (like an amendment referencing "Lease #12345") are automatically linked in the system's data model. The Nova Pro LLM reads each document (or related group of documents) in full and pulls out critical fields: parties involved, lease start and end dates, rent amount and escalation terms, security deposit, renewal options, payment due dates, and so on. The AI essentially creates a relational summary of each lease package. Importantly, because the model retains context, it correctly associates information across pages – for instance, if an amendment extends the lease term, the new end date is captured and tied to the original lease entry.

- **Structured Data Output:** The extracted data is then output in a structured format and saved to a database. LeaseSync's backend organizes this information into a consistent schema (for example, a table for Leases). Each record is indexed and linked: an amendment is flagged as belonging to Lease X, an invoice is linked to Lease X and to a specific month/period of rent. The system populates fields such as Lease ID, Premises Address, Tenant Name, Landlord Name, Lease Term (start/end dates), Rent Details (amount, frequency, escalation), Deposit Amount, Key Clauses (e.g., termination notice period), etc. This structured data can then be reviewed in the app's dashboard or exported. Essentially, what used to be an unstructured contract becomes a row in a database, making it easy to query and analyze across all leases.

Dashboard & Integration

On the front-end, LeaseSync provides a dashboard that gives users a consolidated view of all their processed lease documents. Each lease (with its linked documents) appears as a record that can be clicked for details. The dashboard presents highlights and key terms for each lease in a human-friendly format, so managers can quickly scan important info without reading the original document. For example, the dashboard might show a card for "Lease – Property ABC" with fields like "Tenant: XYZ Corp", "Rent: $5,000/month", "Next Payment Due: May 1, 2025", "Expiration: Dec 31, 2030", etc., all pulled from the documents. Amendments and invoices attached to that lease would be accessible in the same view.

LeaseSync can be integrated with other business systems, recognizing that lease data is often needed in accounting, scheduling, or CRM platforms. The application can be extended to export the structured lease data to common formats (CSV, JSON) or even directly through APIs. This means a company could automatically push rent schedules into their accounting software or feed lease milestones into a calendar/reminder system. For instance, an integration can be set up so that when LeaseSync extracts a lease expiration date, it can create a reminder event 90 days before expiry to start renewal discussions. Similarly, monthly rent payment information could be sent to an accounting system to generate invoices or receipts. Given the clean structured data, such integrations are straightforward.

Overall, the end-to-end LeaseSync workflow turns what was once a pile of static documents into a living database of lease information. The combination of batch processing, AI extraction, a user-friendly dashboard, and integration hooks provides a seamless experience from document ingestion to actionable insights.

5. Model Selection, Cost, and Speed

Choice of Amazon Nova Pro on AWS Bedrock

We chose Amazon Nova Pro, a state-of-the-art foundation model available through AWS Bedrock, for its balanced mix of capability and efficiency. Nova Pro is a multimodal LLM known for having the best combination of accuracy, speed, and cost across a wide range of tasks. In the context of lease processing, accuracy means correctly interpreting legal language; speed means processing documents quickly; and cost refers to the compute expense per document. Nova Pro excels on all three fronts. It can handle extremely large inputs (up to 300k tokens), is optimized for fast inference, and affordably priced. Running on AWS Bedrock also means we get a fully managed, scalable infrastructure with low-latency API calls. Since our target users include businesses in Indonesia, we deploy LeaseSync in the AWS Asia Pacific (Singapore) region (ap-southeast-1). This ensures that the model inference happens on servers close to our users, minimizing network latency and providing quick responses.

Use with Other Models

LeaseSync's architecture itself is LLM-agnostic: it can integrate with a variety of models such as Llama, Claude, Qwen, or DeepSeek. Considerations are tradeoffs in speed, token context window, and cost. For example, OpenAI-o1 or DeepSeek R1 might handle large context windows, but can be more expensive or have slightly slower inference speeds due to its extensive reasoning capabilities. Conversely, a smaller model might reduce cost but lacks robust understanding—particularly for non-English/Indonesian domain-specific texts.

Models with smaller parameter counts (or those specialized only for short answers) typically have difficulty handling entire Indonesian leases, which can be 20+ pages long in a mix of Indonesian and English legal phrasing. Our testing shows that larger LLMs consistently outperform limited-parameter models for non-English/Indonesian prompts, especially for nuanced legal language.

Cost Efficiency

A key advantage of LeaseSync is the drastic reduction in document processing cost compared to manual labor. Using AWS Bedrock's pay-as-you-go model, we only incur charges based on the number of tokens the model processes. Amazon Nova Pro's pricing on Bedrock is approximately $0.0008 per 1,000 input tokens and about $0.0032 per 1,000 output tokens. To put this in perspective, consider a typical lease document: an average page might contain 600–800 tokens of text (roughly 300-400 words). So a 20-page lease could be around 12,000–16,000 tokens. At $0.0008 per 1k tokens, processing the entire 20-page lease costs on the order of $0.01 to $0.013 USD. Even if we double that to account for output tokens or any overhead, we're still talking about only a few cents per document.

Now compare that to human effort: a person reviewing a 20-page lease might easily take an hour or more to thoroughly read and summarize it. In Jakarta, the minimum wage equates to roughly $2.14 per hour for labor. LeaseSync performing the same task for about two cents represents 99% cost reduction. Even outside of minimum wage contexts, for skilled paralegal or analyst time, the savings are massive. This efficiency means companies can scale up their lease processing without proportional increases in staff or overtime. Whether it's 10 leases or 10,000 leases, the cost scales linearly and remains very low, allowing resources to be reallocated to tasks that truly require human judgment.

Processing Speed

LeaseSync dramatically improves processing speed as well. The AI can analyze a 20-page lease in around 30–40 seconds, whereas manual review could take hours when you include reading, note-taking, and data entry. This speed difference means what used to require a full day of work by a person can be done by the system in under a minute. In practical usage, this speedy processing opens up new workflows: for instance, a batch of 100 new leases could be processed overnight and the results ready by the time staff come into the office the next morning. During tests, we observed that even larger documents on the order of 50-100 pages typically complete within minutes.

6. Implementation & Deployment

Technical Stack

LeaseSync is built with a modern, open-source tech stack that prioritizes reliability and scalability. On the backend, we use Python with FastAPI (a high-performance web framework) to handle API requests and orchestrate the workflow. Document data and extracted fields are stored in a relational database, accessed via SQLAlchemy (an ORM for database operations) – this makes it easy to query and manage the structured lease data. For AI processing, the backend integrates with AWS Bedrock to call the Amazon Nova Pro model. We send the document text to Bedrock's API and receive the model's analysis as a response. If the document is not already in text form (for example, a scanned PDF), we utilize Tesseract OCR to extract text from images/PDFs as a preprocessing step. All uploaded files and any intermediate outputs are stored in Amazon S3, which provides secure and durable storage. This setup allows LeaseSync to be stateless in terms of compute – files and results persist in S3/Database, and multiple API servers can run in parallel to process documents from an SQS queue or similar, enabling easy scaling.

On the frontend, we have a web application built with Next.js (a React framework) and styled with Tailwind CSS for a clean, responsive user interface. The frontend communicates with the backend API to upload documents, display processing status, and fetch results for the dashboard. Because LeaseSync is open-source, an emphasis was placed on a stack that many developers are familiar with and that is easy to deploy on cloud infrastructure. The entire application can be containerized with Docker, for instance, and deployed to AWS (using ECS, EKS, or even a simple EC2) or other cloud providers. The use of Bedrock means the heavy AI computation is handled by AWS's managed service, so our backend mainly coordinates tasks and handles presentation – keeping the server logic lightweight.

Integration and Deployment Capabilities

Deploying LeaseSync in an enterprise environment is straightforward. The solution can be packaged and launched in a private cloud environment ensuring that all data (documents and extracted information) stays within the company's control. This is a big plus for data security and compliance, since sensitive documents don't need to leave your infrastructure. For integration, LeaseSync's FastAPI backend can be extended to external systems which allow them to programmatically interact with LeaseSync. For example, an ERP system could automatically send newly signed leases to LeaseSync via API and get back structured data to insert into its own database.

The design is modular – if a company has an existing OCR service or a different LLM they prefer, those components can be swapped out with some configuration changes or minor code adjustments. LeaseSync's open-source nature allows customization: businesses can extend the data model (maybe to extract custom clauses specific to their needs), or integrate additional notification channels (perhaps pushing reminders to Slack or Microsoft Teams).

For deployment, the system can run in a serverless fashion as well. Since Bedrock calls are external API calls, one could host the orchestration on AWS Lambda functions triggered by S3 upload events for a fully serverless pipeline. Alternatively, a more traditional deployment using Docker containers ensures long-running tasks and easier debugging. In summary, LeaseSync is engineered to be easily deployable on AWS and integratable with other systems, making it practical to roll out in real-world IT environments. Companies can start with a pilot and then scale up usage once they are confident.

7. Security, Privacy & Open-Source Availability

Open-Source Model

We decide to release LeaseSync as an open-source project, encouraging both transparency and community collaboration. It uses a permissive Apache 2.0 license, which allows companies and developers to use the software freely in their own environments and to modify it to suit their needs. By open-sourcing the platform, we aim to build trust – users can inspect the source code to ensure it meets their security and quality standards. It also fosters innovation: developers can contribute enhancements, fix bugs, or add new features (for example, support for additional document types or new AI models). Enterprise adoption is also facilitated by open source; many organizations prefer not to be locked into proprietary software. With LeaseSync, they have full ownership of the solution and can maintain it on their own terms.

Data Security & Compliance

From the ground up, LeaseSync is designed to be deployed in a way that protects sensitive data. Lease agreements and related documents often contain confidential financial and personal information, so security is paramount. In a typical deployment on AWS, all data resides in the company's AWS account – documents in S3 (which can be encrypted with AWS KMS), and the database with extracted data can be an encrypted RDS instance. During processing, data is sent to AWS Bedrock, which is a secure AWS service; no data is sent to external third-party APIs. All actions (uploads, data view, exports) can also be logged for audit purposes.

Continuity & Reliability

LeaseSync is intended for enterprise-grade reliability. The system is stateless and can be easily set up with high availability – multiple instances of processing workers that run together so that if one fails, others carry on. Using managed services such as AWS Bedrock, availability can be scaled almost infinitely. Database can be a multi-AZ deployment, object storage (S3) can be multi-AZ replicated. Regular backups of database and versioning of S3 buckets can be set up so original documents are never lost and can be recovered if accidentally deleted.

Because it's open-source, there's also no dependency on a single vendor for support. Companies can opt for community support (GitHub issues) or use internal teams to troubleshoot issues given they have full access to code. The permissive license means that even if the original maintainers stop active development, any other party can fork and continue the project – providing continuity of the solution. For enterprises, this mitigates the risk often associated with proprietary SaaS: with LeaseSync, you won't suddenly lose support or be forced into an upgrade; you have control.

8. Future Directions

Expanding to Other Document Workflows

While LeaseSync currently focuses on lease agreements and their related documents, the core process can be applied to a broad range of document-heavy workflows. In the future, we plan to extend or create variants of LeaseSync for loan agreements, sales contracts, insurance policies, tax documents, etc. Each of these documents has structured data hidden in unstructured text – and a similar challenge of extraction and tracking. For example, a LoanSync could extract interest rates, collateral details, and covenants from loan contracts; or an InsuranceSync could parse policy documents for coverage terms and exclusions. The design of LeaseSync means much of the pipeline (OCR, LLM extractiton, data storage) can be reused, with only the prompts or fine-tuning adjusted for the new document type. By expanding into these areas, businesses could deploy a suite of AI tools to handle all their document workflows in a unified way.

Model Fine-Tuning for Document Processing AI

The pace of AI development is rapid, and one promising direction is fine-tuning large language models for specific legal or financial domains. LeaseSync currently uses a general foundation model (Nova Pro) with prompt engineering to extract information. In the future, we may train or fine-tune a model specifically on lease agreements and real estate legal language. A fine-tuned model could achieve even higher accuracy and reduce the token count (hence cost) for extraction. We're exploring assembling a dataset of annotated leases for this purpose. Additionally, as models like Nova evolve (or new ones appear), we will evaluate their larger context windows and inference improvements. Another aspect is incorporating multi-modal inputs: Nova Pro is multimodal, so in the future LeaseSync could even handle images or floor plan diagrams attached with leases, extracting text from them or analyzing them if needed. We will continue to leverage the latest AI advancements to improve LeaseSync's capabilities. The aim is to remain at the cutting edge of document processing AI, ensuring that LeaseSync users always have the most powerful tools at their disposal for document processing.

9. Conclusion

LeaseSync delivers a compelling solution for organizations drowning in lease paperwork. By automating the extraction and organization of lease terms, it significantly cuts down processing time and cost. What once took hours of manual review can now be done in seconds with greater accuracy. The AI-driven approach means fewer human errors and more consistent outcomes – every lease is analyzed with the same thoroughness, regardless of volume. We demonstrated how LeaseSync outperforms traditional OCR and RAG-based methods by preserving full context and linking related documents, which is crucial for long-form documents such as lease agreements. The cost analysis shows an astonishing 99% reduction in expense compared to manual labor, effectively allowing companies to do more with less and reallocate budget and staff to more strategic tasks. From a technical standpoint, LeaseSync is built with proven open-source tools and cloud services, making it easy to deploy and integrate while keeping data secure and private. This combination of efficiency, accuracy, and security addresses the core pain points in lease management.

Equally important, LeaseSync provides businesses with peace of mind regarding compliance and control. Because it's deployed in your own cloud environment and is open-source, there is no black box – you have full visibility into how your sensitive lease data is handled.

In summary, LeaseSync converts the tedious chore of lease administration into a streamlined, intelligence-driven process. It empowers organizations to manage their leases proactively rather than reactively, with a comprehensive view of their lease portfolio at their fingertips.